

# HC32F0653

## 用户手册

**32 位 ARM Cortex-M0+ 微控制器**

# 目录

<b>1 文档约定</b>	<b>6</b>
1.1 排版约定	6
1.2 寄存器协议	6
<b>2 系统和存储器概述</b>	<b>7</b>
2.1 系统特性	7
2.2 存储器组织	10
2.3 片上 SRAM 存储器	12
2.4 片上 FLASH 闪存存储器	12
2.5 注意事项	12
<b>3 电源控制 (PWR) 与功耗</b>	<b>13</b>
3.1 概述	13
3.2 电源监控	14
3.3 工作模式	15
3.4 低功耗应用	18
3.5 Cortex®-M0+ 内核系统控制寄存器 (SCB->SCR)	19
<b>4 复位和时钟 (RCC)</b>	<b>20</b>
4.1 复位	20
4.2 时钟及控制	22
4.3 时钟启动、校准与状态检测	25
4.4 SysClk 系统时钟切换	27
4.5 寄存器列表	31
4.6 寄存器描述	32
<b>5 系统控制 (SYSCTRL)</b>	<b>41</b>
5.1 概述	41
5.2 寄存器列表	41
5.3 寄存器描述	41
<b>6 中断 (NVIC)</b>	<b>44</b>
6.1 概述	44
6.2 主要特性	44
6.3 SysTick 校准值寄存器	44
6.4 中断优先级	44
6.5 中断向量表	45
6.6 中断相关寄存器	46
6.7 寄存器列表	47
6.8 寄存器描述	47
<b>7 RAM 存储器 (RAM)</b>	<b>51</b>
7.1 概述	51

7.2 主要特性 .....	51
7.3 RAM 存储器操作 .....	51
7.4 奇校验功能 .....	52
7.5 寄存器列表 .....	52
7.6 寄存器描述 .....	53
<b>8 FLASH 存储器 (FLASH) .....</b>	<b>54</b>
8.1 FLASH 概述 .....	54
8.2 FLASH 主要特性 .....	54
8.3 FLASH 功能说明 .....	55
8.4 FLASH 选项字节 .....	58
8.5 FLASH 保护 .....	61
8.6 FLASH 中断 .....	63
8.7 FLASH 寄存器列表 .....	63
8.8 FLASH 寄存器描述 .....	64
<b>9 通用输入输出端口 (GPIO) .....</b>	<b>70</b>
9.1 概述 .....	70
9.2 主要特性 .....	70
9.3 功能描述 .....	71
9.4 编程示例 .....	77
9.5 寄存器列表 .....	78
9.6 寄存器描述 .....	80
<b>10 循环冗余校验 (CRC) .....</b>	<b>85</b>
10.1 概述 .....	85
10.2 主要特性 .....	85
10.3 功能描述 .....	85
10.4 编程示例 .....	86
10.5 寄存器列表 .....	87
10.6 寄存器描述 .....	87
<b>11 通用计算模块 (ALU) .....</b>	<b>88</b>
11.1 概述 .....	88
11.2 主要特性 .....	88
11.3 功能描述 .....	88
11.4 寄存器列表 .....	88
11.5 寄存器描述 .....	89
<b>12 独立看门狗定时器 (IWDG) .....</b>	<b>91</b>
12.1 概述 .....	91
12.2 主要特性 .....	91
12.3 功能描述 .....	91
12.4 编程示例 .....	94
12.5 寄存器列表 .....	95
12.6 寄存器描述 .....	95
<b>13 高级定时器 (ATIM) .....</b>	<b>98</b>

13.1 概述 .....	98
13.2 主要特性 .....	98
13.3 功能描述 .....	99
13.4 寄存器列表 .....	148
13.5 寄存器描述 .....	149
<b>14 通用定时器 (GTIM) .....</b>	<b>187</b>
14.1 概述 .....	187
14.2 主要特性 .....	187
14.3 功能描述 .....	188
14.4 GTIM 中断 .....	227
14.5 触发 ADC .....	227
14.6 调试支持 .....	227
14.7 编程示例 .....	228
14.8 寄存器列表 .....	232
14.9 寄存器描述 .....	233
<b>15 基本定时器 (BTIM) .....</b>	<b>256</b>
15.1 概述 .....	256
15.2 主要特性 .....	256
15.3 功能描述 .....	256
15.4 编程示例 .....	263
15.5 寄存器列表 .....	265
15.6 寄存器描述 .....	266
<b>16 通用异步收发器 (UART) .....</b>	<b>271</b>
16.1 概述 .....	271
16.2 UART 特性 .....	271
16.3 功能描述 .....	272
16.4 低功耗模式 .....	284
16.5 UART 中断 .....	284
16.6 编程示例 .....	285
16.7 寄存器列表 .....	289
16.8 寄存器描述 .....	290
<b>17 串行外设接口总线 (SPI) .....</b>	<b>297</b>
17.1 概述 .....	297
17.2 主要特性 .....	297
17.3 功能描述 .....	298
17.4 寄存器列表 .....	312
17.5 寄存器描述 .....	312
<b>18 I2C 接口 (I2C) .....</b>	<b>316</b>
18.1 概述 .....	316
18.2 主要特性 .....	316
18.3 协议描述 .....	317
18.4 协议描述 .....	321
18.5 编程示例 .....	339



18.6 寄存器列表 .....	341
18.7 寄存器描述 .....	343
<b>19 模数转换器 (ADC) .....</b>	<b>348</b>
19.1 概述 .....	348
19.2 主要特性 .....	348
19.3 功能框图 .....	349
19.4 功能描述 .....	350
19.5 寄存器列表 .....	357
19.6 寄存器描述 .....	358
<b>20 数模转换器 (DAC) .....</b>	<b>369</b>
20.1 概述 .....	369
20.2 主要特性 .....	369
20.3 功能框图 .....	369
20.4 功能描述 .....	370
20.5 寄存器列表 .....	370
20.6 寄存器描述 .....	371
<b>21 全差分运算放大器 (FDA) .....</b>	<b>372</b>
21.1 概述 .....	372
21.2 主要特性 .....	372
21.3 功能描述 .....	372
21.4 寄存器列表 .....	374
21.5 寄存器描述 .....	375
<b>22 模拟电压比较器 (VC) .....</b>	<b>376</b>
22.1 概述 .....	376
22.2 主要特性 .....	376
22.3 功能描述 .....	377
22.4 VC 中断 .....	380
22.5 编程示例 .....	380
22.6 寄存器列表 .....	380
22.7 寄存器描述 .....	381
<b>23 版本记录 .....</b>	<b>384</b>

# 1 文档约定

## 1.1 排版约定

本文中的排版惯例是：

**粗斜体** 表示特殊术语或引文，或者提示注意项。

**粗体** 表示组件或信号名称，如用于无章节序号的标题说明、图表的标题等。

大写英文 用于具有特定技术含义的术语，并包含在词汇表中，如寄存器或位域名称。

彩色文本 表示链接。包括：

- URL 外部链接，如 <http://www.holychip.cn>
- 交叉引用，本文的内部章节互相引用
- 内部链接，图、表或附录、词汇条目

## 1.2 寄存器协议

符号示例	描述
0x53CA	带‘0x’前缀的数字字符串表示十六进制数，数值为数字 0~9 或 A~F 大写英文
REGNAME[n]	REGNAME 的特定位，REGNAME 可以是寄存器或寄存器字段，例如，CR1[2] 指的是 CR1 寄存器（字段）的第 2 位
REGNAME[m:n]	REGNAME 的特定位，REGNAME 可以是寄存器或寄存器字段，例如，CR1[7:5] 指的是 CR1 寄存器（字段）的第 7 位 ~ 第 5 位
RW	可读可写，软件可以读写这些位域
RO	只读，软件只能读取这些位域
WO	只写，软件只能写入该位域，读取该位域时将返回无效数据
RW0	软件可以读写该位域，只能对该位域写入 0，写入 1 对该位域无影响
RW1	软件可以读写该位域，只能对该位域写入 1，写入 0 对该位域无影响
R0W1	软件读取该位域为 0，只能对该位域写入 1，写入 0 对该位域无影响
R1W0	软件读取该位域为 1，只能对该位域写入 0，写入 1 对该位域无影响
RFU	保留位域，请保持默认值
Word	一个字的数据长度为 32 bit
Half Word	一个半字的数据长度为 16 bit
Byte	一个字节的数据长度为 8 bit

## 2 系统和存储器概述

### 2.1 系统特性

HC32F0653 微控制器系统包含：

- 1 个主设备：
  - ARM Cortex-M0+ 内核
- 多个从设备：
  - 片上 SRAM
  - 片上 FLASH
  - FLASH 控制器
  - CRC 冗余计算单元
  - GPIO 端口
  - AHB 总线上所有设备
  - AHB 到 APB 转换桥及 APB 总线上所有设备

主从设备通过 AHB 总线矩阵进行连接，系统架构如下图所示：

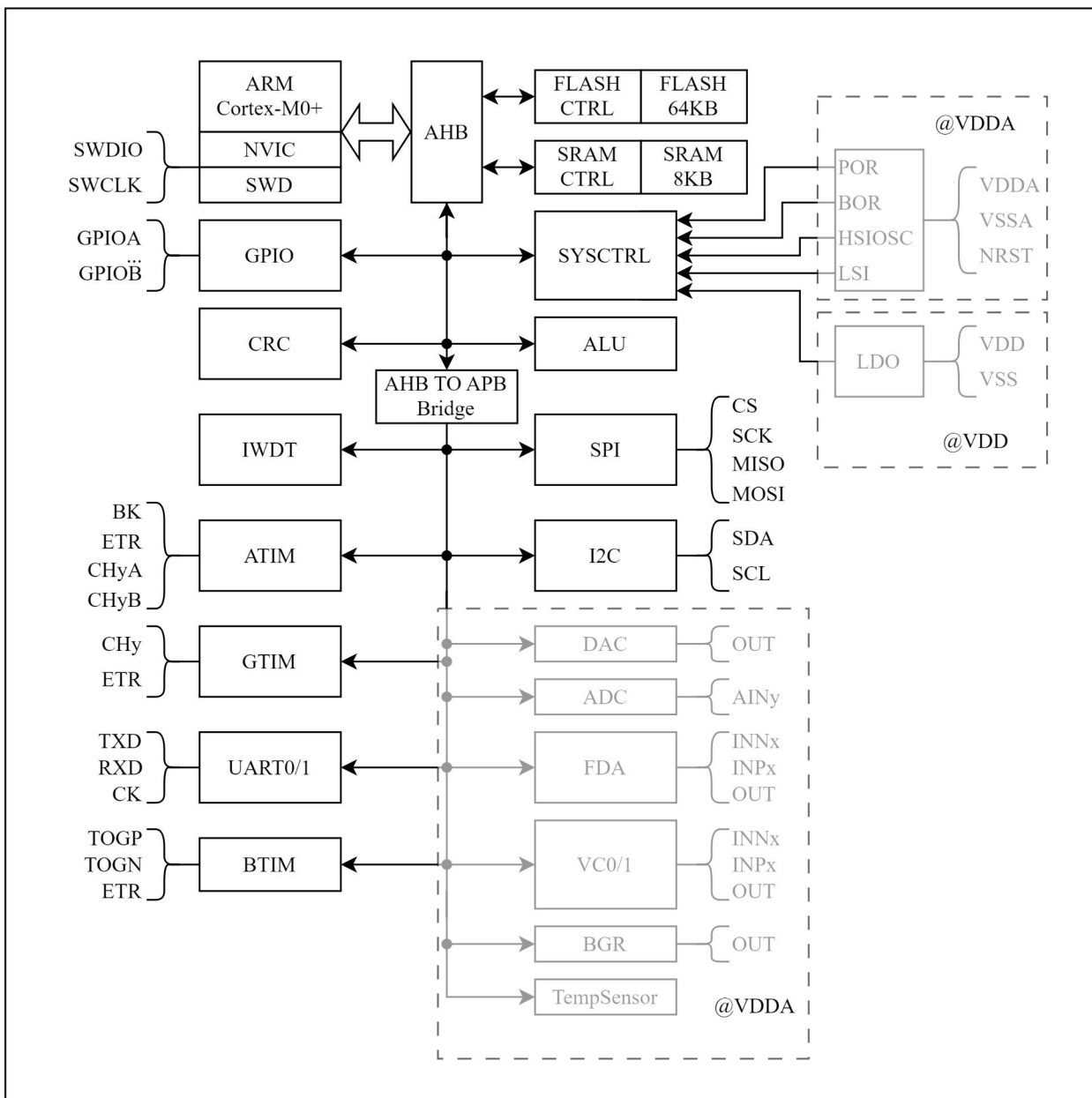


图2-1系统架构

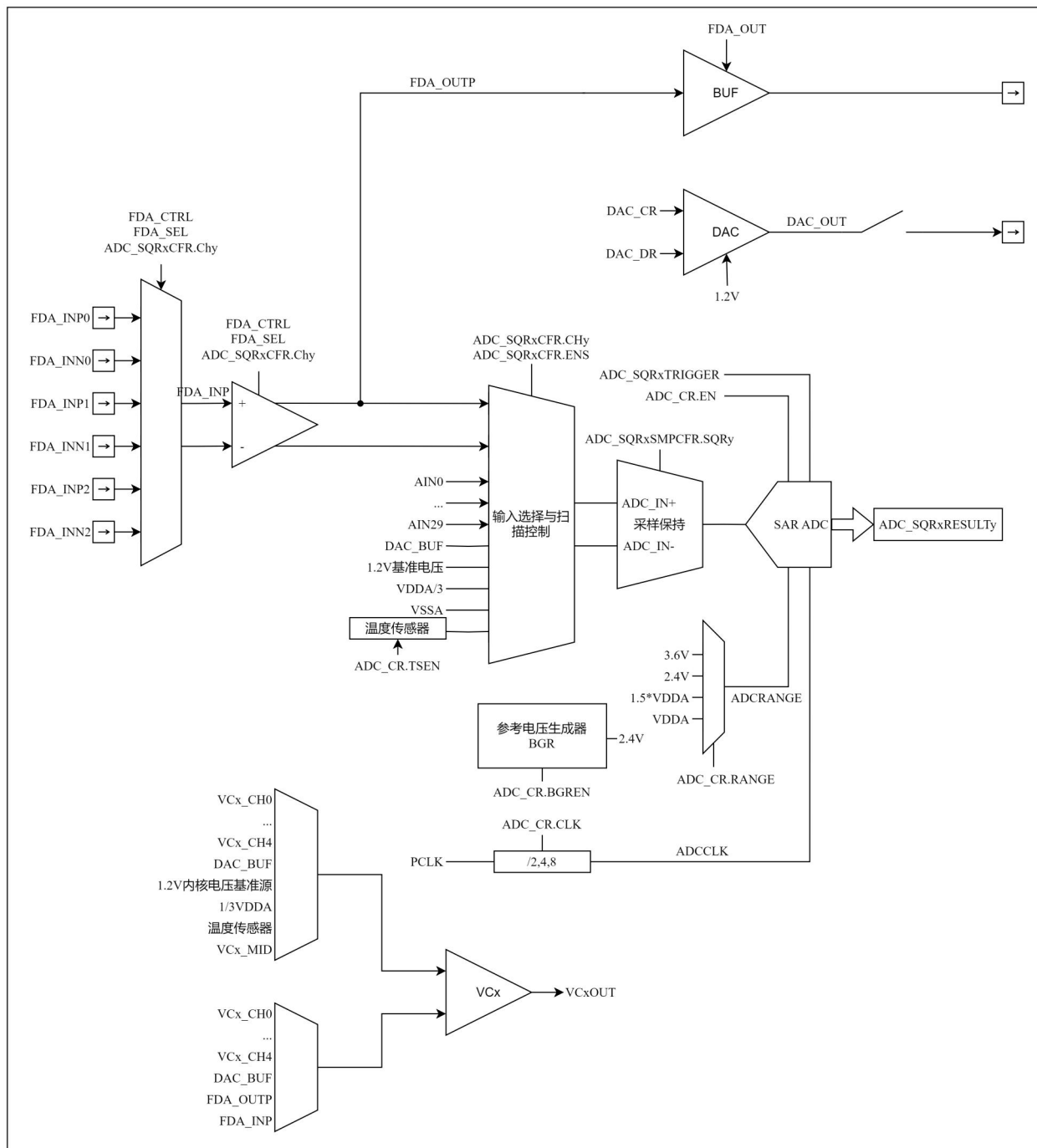
- **系统总线**

实现 M0+ 微处理器的外设总线和总线矩阵的连接。

- **总线矩阵**

管理 M0+ 微处理器对 FLASH、SRAM 以及所有外设的访问存取仲裁。仲裁控制采用轮询调度算法来对负载进行均衡处理，保证总线利用效率。

模拟架构如下图所示:



### 图2-2模拟架构

## 2.2 存储器组织

### 2.2.1 概述

HC32F0653内核为32位的ARM Cortex-M0+微处理器，最大寻址空间为4GB。芯片内置的程序存储器、数据存储器、各外设及端口寄存器被统一编址在同一个4GB的线性地址空间内。

存储器中字节组织为小端格式。一个字存储空间的最低字节数据为字的最低有效位，最高字节数据为最高有效位。

例：将 0x1122 3344存放在地址为 0x0000 0000 的存储器空间中，实际存放结果是：

0x00000000 字节存放 0x44，

0x00000001 字节存放 0x33，

0x00000002 字节存放 0x22，

0x00000003 字节存放 0x11。

## 2.2.2 存储器映射和寄存器边界地址

片上存储器及各外设的详细起始地址空间分配，如下表所示：

表 2-1 存储器和外设地址分配

设备或总线	边界地址	大小	对应外设
主 FLASH 存储器	0x0000 0000 - 0x0000 FFFF	64KB	主 FLASH
用户配置字	0x1FFF F800 - 0x1FFF F9FF	512B	OPTION
FT 测试信息	0x1FFF FA00 - 0x1FFF FBFF	512B	FT
CP 测试信息	0x1FFF FC00 - 0x1FFF FDFE	512B	CP
华虹测试用	0x1FFF FE00 - 0x1FFF FFFF	512B	HH
SRAM 存储器	0x2000 0000 - 0x2000 1FFF	8KB	SRAM
AHB 外设	0x4000 0000 - 0x4000 03FF	1KB	FLASH CTRL
	0x4000 0400 - 0x4000 07FF	1KB	SRAM CTRL
	0x4000 0800 - 0x4000 0BFF	1KB	RCC
	0x4000 0C00 - 0x4000 0FFF	1KB	SYSCTRL
	0x4000 1000 - 0x4000 13FF	1KB	CRC
	0x4000 1400 - 0x4000 17FF	1KB	ALU
	0x4000 2000 - 0x4000 23FF	1KB	GPIOA
	0x4000 2400 - 0x4000 27FF	1KB	GPIOB
	0x4000 2C00 - 0x4000 2FFF	1KB	GPIO MAP
APB 外设	0x4001 0000 - 0x4001 03FF	1KB	ATIM
	0x4001 0400 - 0x4001 07FF	1KB	GTIM
	0x4001 0800 - 0x4001 0BFF	1KB	BTIM0
	0x4001 0C00 - 0x4001 0FFF	1KB	BTIM1
	0x4001 1000 - 0x4001 13FF	1KB	IWDT
	0x4001 1400 - 0x4001 17FF	1KB	UART0
	0x4001 1800 - 0x4001 1BFF	1KB	UART1
	0x4001 1C00 - 0x4001 1FFF	1KB	I2C
	0x4001 2000 - 0x4001 23FF	1KB	SPI
	0x4001 2400 - 0x4001 27FF	1KB	VC0
	0x4001 2800 - 0x4001 2BFF	1KB	VC1
	0x4001 2C00 - 0x4001 2FFF	1KB	FDA
	0x4001 3000 - 0x4001 33FF	1KB	DAC
	0x4001 3400 - 0x4001 37FF	1KB	ADC
	0x4001 3800 - 0x4001 3BFF	1KB	BTIM2
M0+ 外设	0xE000 0000 - 0xE00F FFFF	1MB	M0+ 内核外设

## 2.3 片上 SRAM 存储器

HC32F0653 内部集成 8KB 的片上 SRAM，起始地址为 0x2000 0000。SRAM 支持以字节 (8bit)、半字 (16bit) 或全字 (32bit) 3 种位宽进行访问，能够被 CPU 以最大的系统时钟频率进行访问，零等待延迟。

## 2.4 片上 FLASH 闪存存储器

片上 FLASH 存储器共 64KB，地址空间为 0x0000 0000 - 0x0000 FFFF。该区域主要用于存放应用程序代码和用户数据，用户可编程。

FLASH 控制器实现对 FLASH 的各种操作（擦除、写、读取），内部的预取缓存机制可加速 CPU 代码执行速度。

FLASH 支持以字节 (8bit)、半字 (16bit) 或全字 (32bit) 3 种位宽进行访问，访问的最高频率为 24MHz，如果系统配置的 HCLK 时钟频率高于 24MHz，则必须通过 FLASH 控制寄存器 FLASH\_CR2 的 WAIT 位域配置合理的响应等待时间，才能保证 FLASH 被正确访问。

## 2.5 注意事项

HC32F0653 在使用中需要注意如下事项：

- FLASH、SRAM 以及 GPIOx\_ODR、CRC\_DR 等少数寄存器支持 8bit/16bit/32bit 访问方式，其它外设只支持 32bit 访问方式。对不支持 8bit/16bit 访问方式的地址进行 8bit/16bit 访问会产生 HardFault 硬件错误异常。
- RES 区域均为保留的地址空间，无对应的物理设备，如果对这些保留地址空间进行访问，会产生 HardFault 硬件错误异常。
- 芯片复位后，除了 SYSTICK 和 SRAM 外的所有外设时钟都处于关闭状态，用户在使用这些外设前必须通过外设时钟使能控制寄存器（AHB 外设时钟使能控制寄存器 SYSCTRL\_AHBEN、APB 外设时钟使能控制寄存器 SYSCTRL\_APBEN）打开对应外设的配置时钟和工作时钟。没有打开外设配置时钟会导致访问失败：写入数据失败，且不会产生 HardFault 硬件错误异常，读数据的结果不可用。没有打开外设的工作时钟，则外设主要功能失效。
- 上电后 4 ms 内，SWDIO 和 SWCLK 为默认管脚，可以在初始化程序中将两个脚的上拉使能打开。
- RST 脚上电后不会立即生效，读完用户选项字节以后，再决定是否生效。
- ADC 模拟看门狗，在配置成单序列单通道转换时，需要把 AN0 和所设置的通道同时打开才能生效。
- UART 在间隔段标志位置 1 后需要延时 1/16 波特率周期才能清掉标志位，该标志位和 PE 共用。



## 3 电源控制（PWR）与功耗

### 3.1 概述

HC32F0653 工作时需要一组电源供电：VDD、VSS。

HC32F0653 内嵌一路 1.5V 低压差 LDO 稳压器，为芯片内部数字电源域供电。

芯片内置参考电压生成器（BGR）电路，可为其他模拟模块提供参考电压。

芯片内部电源域的划分与分配，如下图所示：

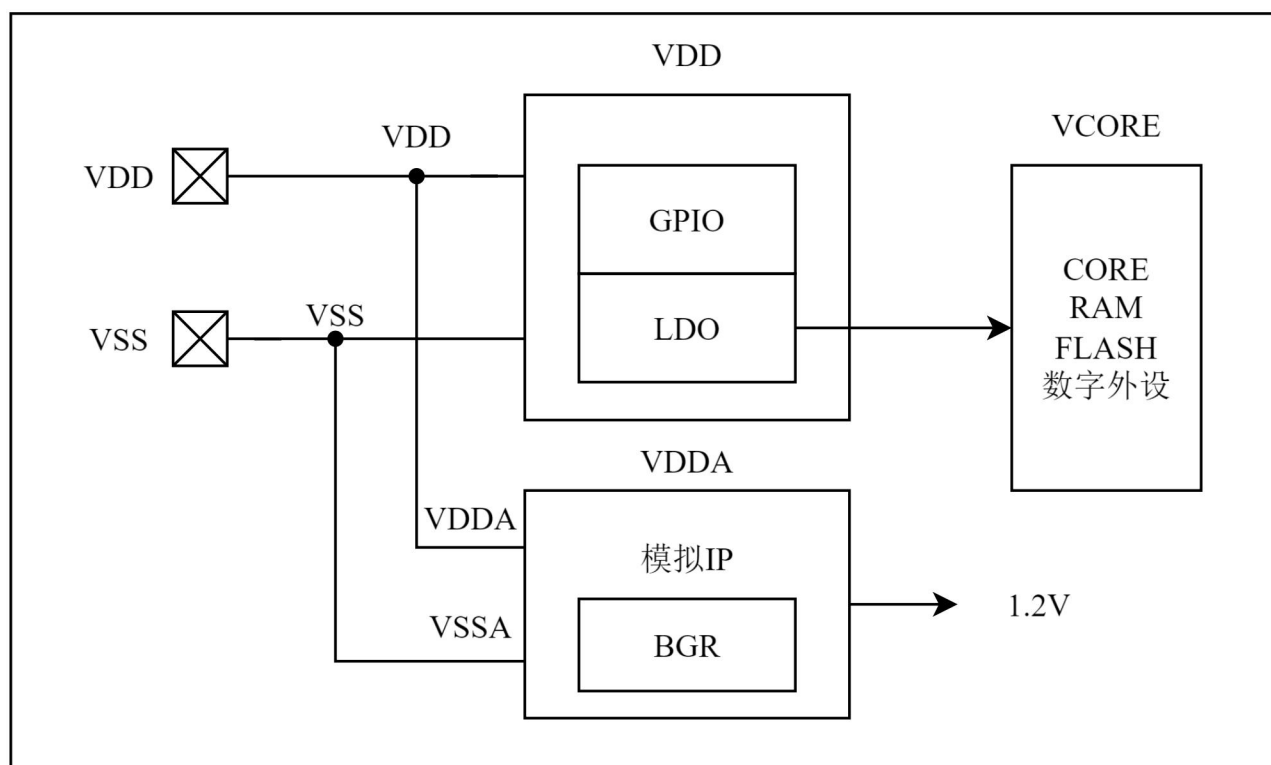


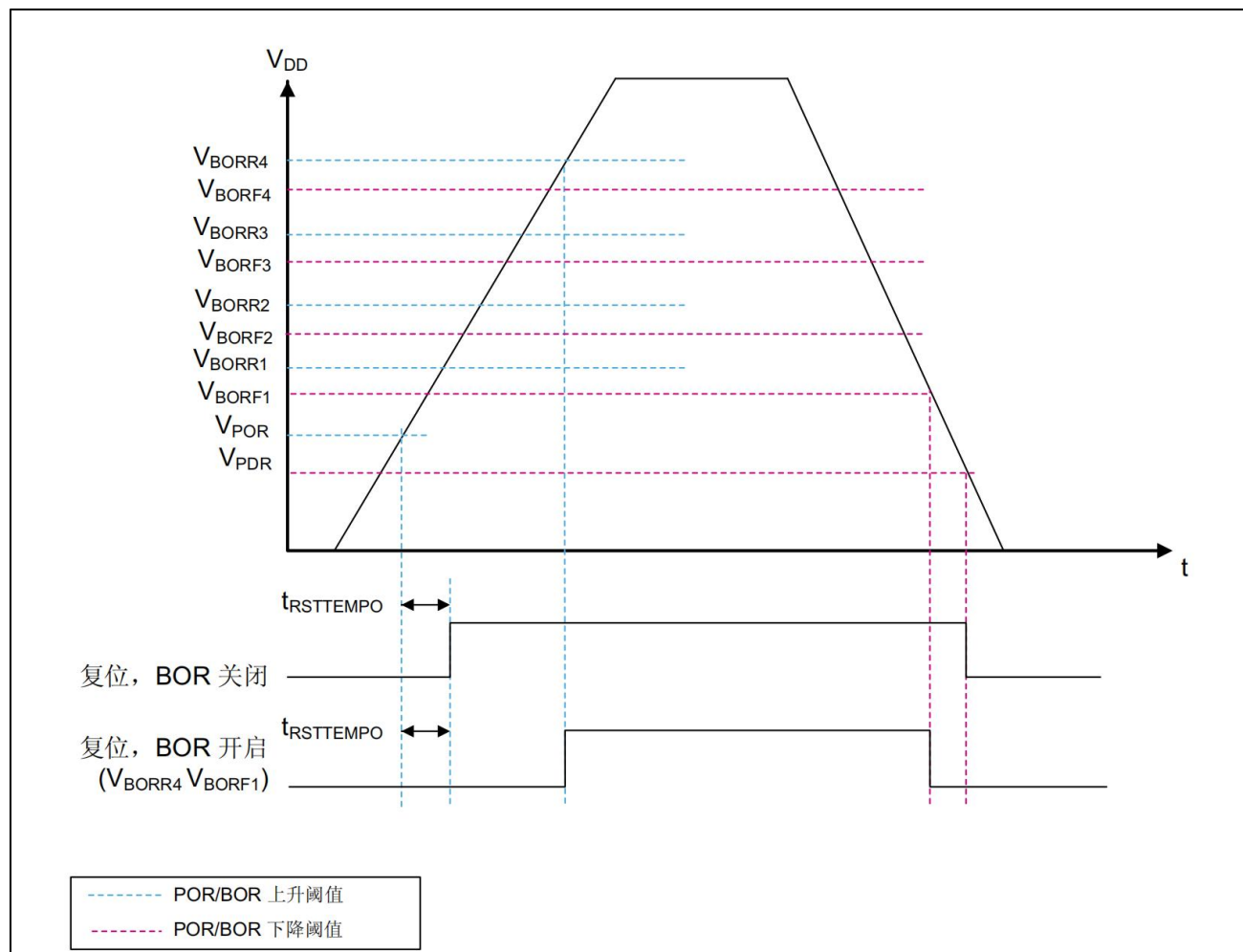
图 3-1 系统电源

### 3.2 电源监控

上电复位 (POR) / 掉电复位 (BOR)

HC32F0653 集成了上电复位（POR）和掉电复位（BOR）电源监控电路，电源上电后始终处于工作状态。POR/BOR 同时监控 VDD 和 VDDA 电源电压，当监测到电源电压低于复位阈值（VBOR）时，系统会进入复位状态。用户无需额外增加外部硬件复位电路。

电源上电启动以及电源跌落阶段的复位信号波形，如下图所示：



### 图3-2上电/掉电复位时序图

### 3.3 工作模式

HC32F0653 支持三种工作模式，由内嵌的电源管理模块自动完成电源的统一管理。三种工作模式是：

- 运行模式（Active mode）
- 休眠模式（Sleep mode）
- 深度休眠模式（Deep Sleep mode）

电源上电后，系统自动进入运行模式。用户可通过软件程序，进入休眠或深度休眠两种低功耗运行状态；在低功耗运行状态时，可通过硬件中断触发唤醒机制，使系统返回到运行模式。

三种工作模式的转换机制，如下图所示：

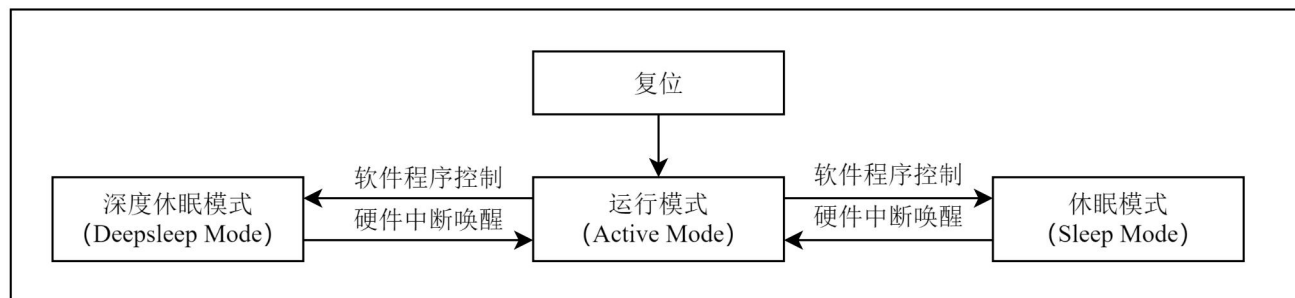


图3-3 工作模式转换机制

运行模式下 CPU 正常运行，所有模块用户均可正常使用。休眠模式下，CPU 停止运行，所有外设不受影响，所有 I/O 引脚保持状态不变。深度休眠模式下，CPU 停止运行，高速时钟（HSIOSC）自动关闭，低速时钟（LSI）保持原状态不变，所有 I/O 引脚保持状态不变。

不同工作模式下 CPU 与时钟状态，如下表所示：

表 3-1 工作模式与 CPU 及时钟状态

工作模式	CPU 状态	HSI	LSI	LDO
运行模式（Active mode）	运行	ON/OFF	ON/OFF	正常
休眠模式（Sleep mode）	停止	ON/OFF	ON/OFF	正常
深度休眠模式（Deep Sleep mode）	停止	OFF	ON/OFF	低功耗

### 3.3.1 进入休眠模式或深度休眠模式

使用 M0+ 内核的 ARM 等待中断专用指令，WFI（Wait for Interrupt），配合 M0+ 内核的系统控制寄存器（SCR，System Control Register）的 SLEEPONEXIT 和 SLEEP DEEP 位编号，可实现立即进入或退出（中断服务程序）时进入休眠模式或深度休眠模式。

- 立即进入

执行 WFI 指令，MCU 将立即进入休眠模式（SLEEP DEEP 为 0 时）或深度休眠模式（SLEEP DEEP 为 1 时）

- 退出时进入

将 SLEEPONEXIT 位置 1，当退出最低优先级的中断服务程序后，MCU 会进入休眠模式（SLEEP DEEP 为 0 时）或深度休眠模式（SLEEP DEEP 为 1 时），而不需执行 WFI 指令。

表 3-2 进入休眠模式或深度休眠模式

进入方式	SLEEPONEXIT 位	进入条件	SLEEP DEEP 位	进入模式
立即进入	0	执行 WFI 指令	0	休眠模式
			1	深度休眠模式
退出时进入	1	退出最低优先级的中断服务程序后	0	休眠模式
			1	深度休眠模式

**注：** 在深度休眠模式下，系统将自动关闭高速时钟。如用户需要在深度休眠模式下使部分外设仍保持运行，则须在进入深度休眠模式前，启动低速时钟并将该外设时钟设置为低速时钟。

### 3.3.2 退出休眠模式或深度休眠模式

在休眠模式或深度休眠模式下，均可通过中断来唤醒 CPU，返回到运行模式。但是，值得注意的是，如果用户在中断服务程序中执行 WFI 命令进入休眠（包括深度休眠），则需要比此中断更高优先级的中断才能唤醒 CPU，因此，我们强烈建议用户在准备进入休眠前，应先处理完所有中断服务程序，并且清除所有中断请求和中断标志。

不同工作模式下，CPU 可响应的中断类型，如下表所示：

表 3-3 工作模式与中断源

中断号	中断源	运行模式	休眠模式	深度休眠模式
0	IWDT	Y	Y	Y
1	FLASH	Y	Y	N
2	SRAM	Y	Y	N
3	RCC	Y	Y	N
4	GPIOA	Y	Y	Y
5	GPIOB	Y	Y	Y
6	ADC	Y	Y	N
7	ATIM_BRK	Y	Y	N
8	ATIM_UP	Y	Y	N
9	ATIM_TRG ATIM_COM ATIM_DIR ATIM_IDX	Y	Y	N
10	ATIM_CC	Y	Y	N
11	VC0	Y	Y	N
12	VC1	Y	Y	N
13	GTIM	Y	Y	N
14	BTIM0	Y	Y	N
15	BTIM1	Y	Y	N
16	BTIM2	Y	Y	N
17	I2C	Y	Y	N
18	SPI	Y	Y	N
19	UART0	Y	Y	N
20	UART1	Y	Y	N

使用中断退出休眠模式，用户必须在进入休眠（包括深度休眠）前使能此中断的允许位。

中断唤醒退出休眠模式后，CPU 将立即进入此中断的中断服务程序。如果用户未设置此中断服务程序，且为立即进入休眠时：CPU 将继续执行进入休眠的 WFI 指令的下一条语句；而为退出时进入休眠时：继续执行最后进入的中断服务程序的下一条语句。一般情况下，基于系统可靠性考虑，强烈建议用户设置此中断的服务程序，并在中断服务程序中清除中断请求和中断标志。

中断唤醒退出深度休眠模式时，CPU 运行状态与退出休眠模式相同。

### 3.3.3 工作模式与复位源

即使在休眠模式或深度休眠模式，CPU 亦可响应部分复位源。不同工作模式下 CPU 可响应的硬件复位或软件复位，如下表所示：

表 3-4 工作模式与复位源

复位源	运行模式	休眠模式	深度休眠模式
上电复位 / 掉电复位 (POR/BOR)	Y	Y	Y
引脚输入复位 (RST)	Y	Y	Y
IWDT 复位	Y	Y	Y
内核 LOCKUP 故障	Y	N	N
内核 SYSRESETREQ 复位	Y	N	N

## 3.4 低功耗应用

休眠模式下，CPU 停止运行，所有外设保持运行，包括 ARM® Cortex®-M0+ 内核外设，比如 NVIC、SysTick 等外设。休眠模式的功耗低于运行模式。

深度休眠模式下，CPU 停止运行，高速时钟关闭，低速时钟保持状态不变，部分外设可以配置为继续运行，NVIC 中断处理仍然工作。深度休眠模式的功耗远小于休眠模式。

用户可以通过以下方式降低系统运行功耗：

#### 降低系统时钟频率

- 使用低频率的高速时钟 HSI 或低速时钟 LSI
- 通过编程预分频寄存器，降低 SYSCLK、HCLK、PCLK 的频率
  - 设置 SYSTCTRL\_CR0 寄存器的 SYSCLK 位编号，选择适当的时钟源
  - 设置 SYSTCTRL\_CR0 寄存器的 HCLKPRS 位编号，降低 HCLK 频率
  - 设置 SYSTCTRL\_CR0 寄存器的 PCLKPRS 位编号，降低 PCLK 频率

#### 关闭休眠期间不使用的时钟和外设

- AHB 总线时钟 HCLK 和 APB 总线时钟 PCLK，可以根据需要关闭
- 关闭与唤醒无关的外设的时钟
  - AHB 外设时钟使能控制寄存器，SYSTCTRL\_AHBEN
  - APB 外设时钟使能控制寄存器，SYSTCTRL\_APBEN

### 3.5 Cortex®-M0+ 内核系统控制寄存器 (SCB->SCR)

地址： 0xE000 ED10 复位值： 0x0000 0000

位编号	位符号	权限	说明
31-5	RFU	-	保留位
4	SEVONPEND	RW	设置为 1 时，中断的每次新的挂起都会产生一个事件，如果使用了 WFE 休眠，它可用于唤醒处理器。
3	RFU	-	保留位
2	SLEEP DEEP	RW	1：执行 WFI 或 SLEEPONEXIT 为 1 且退出所有中断服务程序时进入深度休眠模式（DeepSleep mode）； 0：执行 WFI 或 SLEEPONEXIT 为 1 且退出所有中断服务程序时进入休眠模式（Sleep mode）
1	SLEEPONEXIT	RW	1：当退出所有中断服务程序时，处理器自动进入休眠模式（或深度休眠模式）； 0：退出时进入休眠功能被禁止
0	RFU	-	保留位

## 4 复位和时钟（RCC）

### 4.1 复位

HC32F0653 支持以下 3 种复位：

- 电源复位
  - 上电复位（POR）
  - 掉电复位（BOR）
- 系统复位
  - 引脚输入复位（RST）
  - IWDT 复位
  - 内核 SYSRESETREQ 复位
  - 内核 LOCKUP 故障复位
- 外设复位

发生电源复位后，整个芯片都会被复位到默认值

发生系统复位后，CPU 重新运行，除了系统复位标志寄存器 RCC\_RESETFLAG 中的复位标志外，寄存器都被复位到默认值，程序从中断向量表的复位中断入口地址开始执行。

用户可通过系统复位标志寄存器 RCC\_RESETFLAG 来查询本次系统复位的复位源。复位标志由硬件置位，软件清零。建议用户在读取标志后清除该寄存器相关标志位使标志位为 0，以避免在下次复位后发生混淆。

#### 4.1.1 电源复位

##### 4.1.1.1 上电复位（POR） / 掉电复位（BOR）

HC32F0653集成了专门的 POR 和 BOR 电路对电源电压进行监控，在电源电压低于安全范围时将芯片保持在复位状态，防止芯片在上电 / 掉电过程中误动作。为保证系统工作稳定，用户须保持电源电压在安全范围内。

**注：**

为保证芯片解除复位后工作正常，须在电路设计上保证 VDD/VDDA 同时上下电。

#### 4.1.2 系统复位

##### 4.1.2.1 引脚输入复位（RST）

HC32F0653具有专门的复位输入引脚，输入一定宽度的低电平信号会引起系统复位，复位入口地址被固定在 0x0000 0004 处。芯片内部设计有专用防抖电路，短于 20μs 的低电平脉冲信号会被屏蔽。

复位输入引脚内置有上拉电阻，用户如需外接 RC 电路，须考虑内部上拉电阻的影响。

在引脚输入复位有效的情况下，芯片也可以下载程序和进入仿真。

##### 4.1.2.2 IWDT复位

HC32F0653集成有独立看门狗（IWDT），当 IWDT 满足复位条件时，会产生复位信号引起系统复位。

##### 4.1.2.3 内核 SYSRESETREQ复位

内核 SYSRESETREQ 复位是软件复位，通过设置 ARM® Cortex®-M0+ 的应用中断和控制状态寄存



器（AIRCR, Application Interrupt and Reset Control Register）的 SYSRESETREQ 位编号来实现。应用程序设置该位为 1 则会产生内核 SYSRESETREQ 复位，从而实现软件复位。

#### 4.1.2.4 内核LOCKUP故障复位

当 CPU 遇到严重异常（如读取到的指令无效、访问 FLASH 时位宽和目标地址不匹配），会将 PC 指针停在当前地址处锁定，并产生内核 LOCKUP 故障复位信号。

芯片上电后，LOCKUP 复位功能默认处于不使能状态，用户需要手动使能，通过设置系统控制寄存器 RCC\_CR2 的 LOCKUP 位编号为 1，即可使能 LOCKUP 复位功能。

#### 4.1.3 外设复位

用户可使用软件将 HC32F0653 内部的各种外设单独复位。外设复位可以让各外设的寄存器、状态机以及各种控制逻辑等，恢复到上电复位后的默认状态。用户通过设置 RCC\_AHBRSRST 和 RCC\_APBRSRST 寄存器来进行各外设模块的独立复位操作。

对于上电复位 / 掉电复位（POR/BOR），内部各外设模块已处于复位后的默认状态，可直接进行模块初始化配置，不需要再单独对各模块进行独立复位。

对于其它类型系统复位情形，部分外设可能保留复位前的工作状态。用户如需要将外设恢复到上电后的默认状态，应通过对应的复位寄存器执行外设复位操作后再使用。

## 4.2 时钟及控制

### 4.2.1 概述

HC32F0653 内置两路时钟产生电路，通过分频器和多路选择器产生各种不同频率的时钟给 CPU 及各外设使用。

系统内部时钟 SysClk 经过分频为 CPU 内核提供高级高性能总线时钟 HCLK，HCLK 时钟经过分频为数字及模拟外设提供高级外设时钟 PCLK。

系统时钟 SYSCLK 有 2 个时钟源：

- HSI 时钟，由内部高速 RC 振荡器时钟（HSIOSC）经过分频产生
- 内部低速 RC 振荡器时钟（LSI）

系统内部时钟树如下图所示：

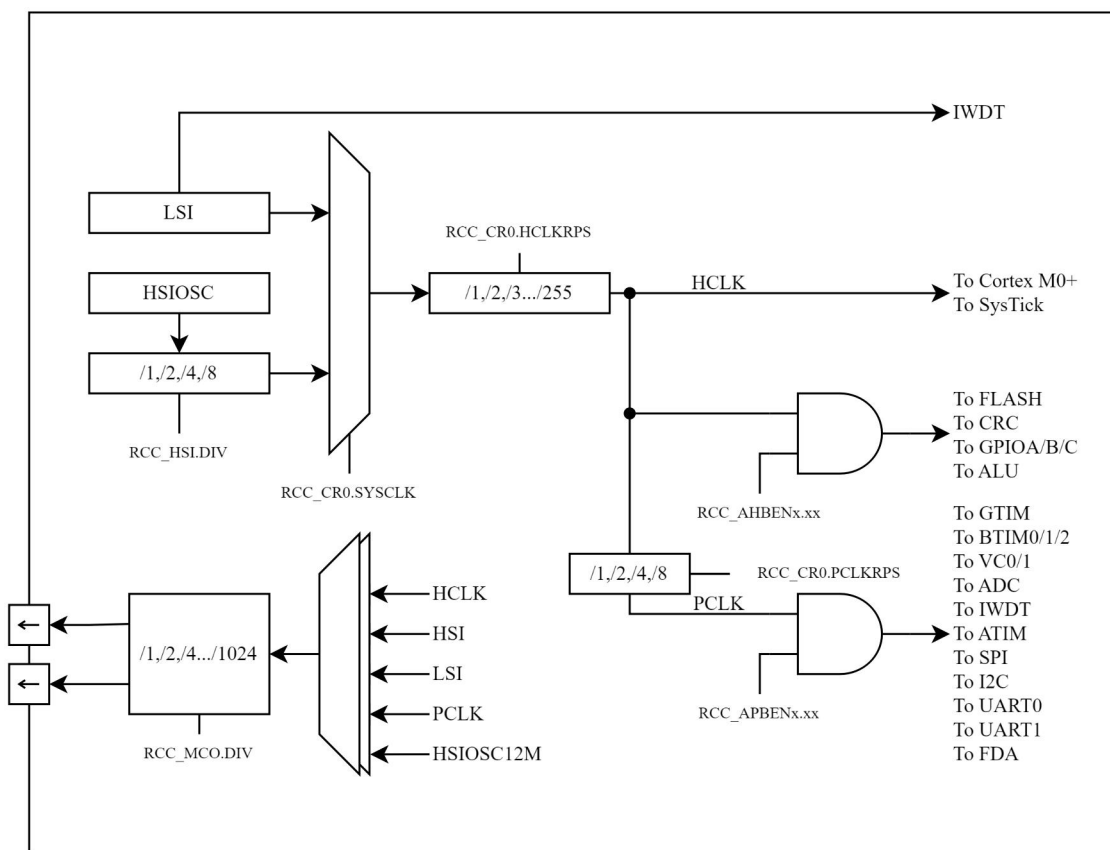


图 4-1 系统内部时钟树

内部高速 RC 振荡器时钟 HSIOSC 经过分频器分频后产生 HSI 时钟，分频系数通过内置高频时钟控制寄存器 RCC\_HSI 的 DIV 位编号进行设置，有效分频系数为 1、2、4、8。

系统时钟 SysClk 可选 2 个时钟源：HSI、LSI，通过系统控制寄存器 RCC\_CR0 的 SYSCLK 位编号进行选择。

高级高性能总线时钟 HCLK，由系统时钟 SysClk 经过分频产生，作为 M0+ 内核、SysTick、FLASH、CRC、GPIO、定时器、SPI、I2C 等模块的配置时钟及工作时钟。分频系数通过系统控制寄存器 RCC\_CR0 的 HCLKPRS 位编号设置。

## 4.2.2 系统时钟与工作模式

HC32F0653 支持三种工作模式：运行模式（Active mode），休眠模式（Sleep mode）和深度休眠模式（Deep Sleep mode）。运行模式下 CPU 正常运行，所有模块用户均可正常使用。休眠模式下，CPU 停止运行，各时钟振荡器及外设保持原状态不变。深度休眠模式下，CPU 停止运行，高速时钟 HSIOSC 的振荡器被自动关闭以节省功耗；低速时钟 LSI 的振荡器保持原状态不变；系统时钟 SysClk 及 HCLK、PCLK 时钟是否有效，取决于 SysClk 系统时钟的时钟源状态。

从深度休眠模式唤醒后，如果系统控制寄存器 RCC\_CR2 的 WAKEUPCLK 位编号配置为 1，则系统会自动使用 HSI 作为系统时钟的时钟源。如果 RCC\_CR2.WAKEUPCLK 为 0，则系统会等待系统进入深度休眠前所使用的系统时钟源稳定后才开始运行。

由于 HSI 启动速度很快，可快速响应用户需求，因此建议在进入深度休眠模式前切换系统时钟的时钟源为 HSI 时钟或者配置 RCC\_CR2.WAKEUPCLK 为 1。

## 4.2.3 HSIOSC 时钟

HSIOSC 时钟由内部 RC 振荡器产生，不需要外部电路，启动速度快。HSIOSC 时钟频率可配置为 48 / 60 MHz，默认为 48 MHz。

RC 振荡器输出时钟的频率受芯片加工过程、工作电压、环境温度等因素影响，HC32F0653 提供了 HSIOSC 时钟频率校准功能，用户可通过设置内置高频时钟控制寄存器 RCC\_HSI 的 TRIM 位编号值来校准 HSIOSC 时钟频率。

HSIOSC 内部高速 RC 振荡器在芯片上电后，默认处于开启状态，用户可通过设置系统控制寄存器的 RCC\_CR1 的 HSIEN 位编号为 0 来关闭。如用户停止并重新启动了 HSIOSC 振荡器，可通过内置高频时钟控制寄存器 RCC\_HSI 的 STABLE 标志位来确定 HSI 时钟是否稳定，STABLE 标志为 1 表示 HSIOSC 时钟已稳定，为 0 则表示 HSIOSC 时钟还未稳定。

**注：HSIOSC 振荡器应在启动前设置好所有参数，启动后禁止修改相关参数。**

HSIOSC 时钟经过分频后输出 HSI 时钟，分频系数通过内置高频时钟控制寄存器 RCC\_HSI 的 DIV 位域设置，有效分频系数为 1、2、4、8，上电后默认值为 8，所以 HSI 时钟默认频率为 6 MHz。

## 4.2.4 LSI 时钟

LSI 时钟由内部低速 RC 振荡器产生，LSI 频率值 FLASH\_LSIFREQ 保存在地址 0x1FFF FA90。内部低速 RC 振荡器不需要外部电路。

LSI 内部低速 RC 振荡器默认处于关闭状态，通过设置系统控制寄存器 RCC\_CR1 的 LSIEN 位编号为 1 启动。LSI 振荡器启动后，芯片内部时钟监控模块检测到一定数量的 LSI 时钟信号，则认为 LSI 时钟已稳定。检测时钟数量可通过内置低频振荡器控制寄存器 RCC\_LSI 的 WAITCYCLE 位编号进行设置，如下表所示：

表 4-1 LSI 稳定检测时钟信号数量

RCC_LSI.WAITCYCLE	LSI 检测时钟数量
00	4（默认值）
01	16
10	64
11	256

通过内置低频时钟控制寄存器 RCC\_LSI 的 STABLE 标志位，可确定 LSI 时钟是否稳定，STABLE 标志为 1 表示 LSI 时钟已稳定，为 0 则表示 LSI 时钟还未稳定。

**注：LSI 振荡器在启动前必须设置好所有参数，启动后禁止修改相关参数。**

### 4.2.5 SysClk 系统时钟

系统时钟 SysClk 可选 2 种时钟源，包括 HSI、LSI。

系统上电复位完成后默认选择 HSI 作为 SysClk 的时钟源，时钟频率默认值是 6MHz。

用户可通过系统控制寄存器 RCC\_CR0 的 SYSCLK 位编号选择系统时钟源，当选择某一时钟作为系统时钟源后，用户无法通过设置该时钟电路的使能位 RCC\_CR1.xxxEN 为 0 来停止该时钟。

### 4.2.6 片内外设时钟控制

片内外设一般都需要有配置时钟和工作时钟，配置时钟用来响应 CPU 对外设寄存器的读写操作。

在使用外设前都必须打开外设的配置时钟和工作时钟，否则外设无法工作。通过设置 AHB 外设时钟使能控制寄存器 RCC\_AHBEN 的对应位为 1，打开对应外设的配置时钟和工作时钟。

当外设不需要使用时，通过关闭外设的配置时钟和工作时钟禁止外设，能有效降低芯片功耗。

## 4.3 时钟启动、校准与状态检测

### 4.3.1 时钟启动

HC32F0653 的时钟源启动过程类似，以 HSI 为例说明时钟启动后的稳定过程，如下图所示：

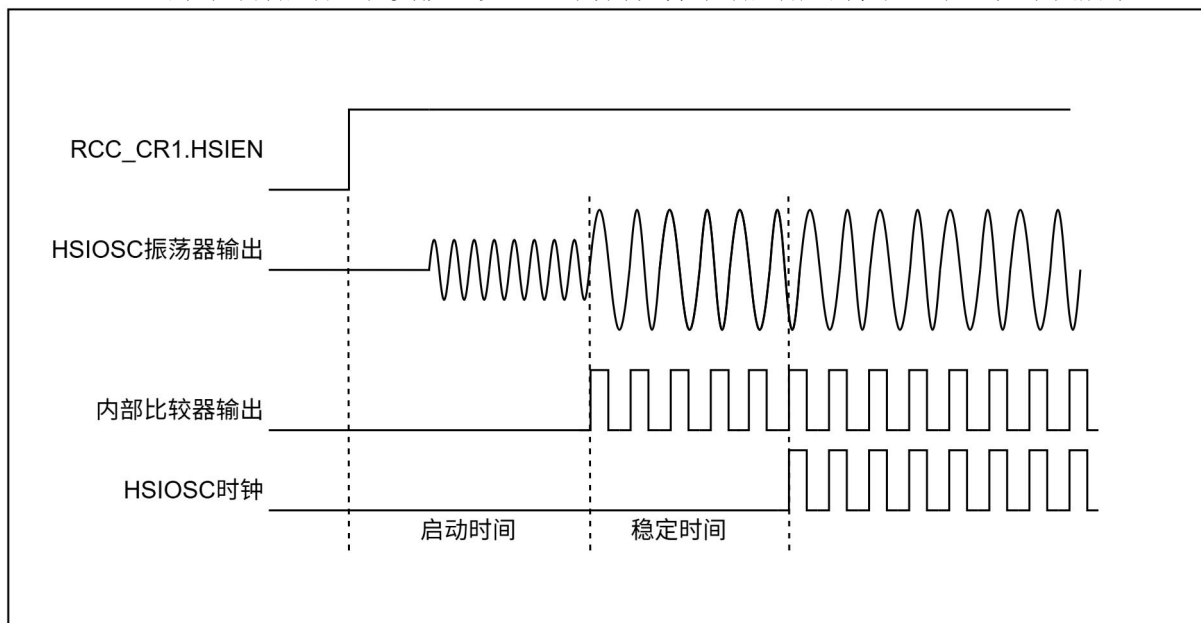


图 4-2 HSI 时钟启动过程

当设置 `RCC_CR1.HSIEN` 为 1 后，HSI 时钟振荡电路开始工作，但此时输出的时钟信号振幅很小。经过启动时间阶段后，输出时钟信号的振幅、占空比等可满足内部采样电路需求，进入稳定时间阶段。在稳定时间阶段，芯片内部时钟监控电路对 HSI 输出的时钟信号进行计数，当计数值达到设定的个数后，认为 HSI 时钟信号已稳定，HSI 时钟稳定标志位 `RCC_HSI.STABLE` 被置 1。

### 4.3.2 时钟校准

时钟校准主要针对 HSIOSC 时钟和 LSI 时钟。通过调整振荡器的 `TRIM` 值来实现时钟频率校准。

#### HSIOSC 时钟校准

芯片出厂时已预调好校准参数，并存放在 FLASH 中。应用程序只需要将 FLASH 内的校准值读出并写入 `RCC_HSI.TRIM` 即可获得精准的 HSI 时钟。HSI 时钟校准值存放地址为：`0x1FFF FA40`。如需其它频率的时钟则需要用户自行调整 `RCC_HSI.TRIM` 的值。

#### LSI 时钟校准

LSI 时钟的频率值 `FLASH_LSIFREQ` 保存地址：`0x1FFF FA90`，读出来就可以得到当前 LSI 的精准频率值（该数值单位为 HZ），来用于计算。

### 4.3.3 时钟稳定检测

HSIOSC、LSI 这两种时钟源都支持时钟稳定检测功能，用户可通过对应时钟源的稳定标志位来确定时钟状态。时钟稳定标志在关闭时钟源时由硬件清 0，在时钟源启动并稳定后由硬件置 1。

注意时钟稳定标志只针对时钟启动过程而言，在时钟稳定运行过程中，检测到时钟运行失效不会影响该时钟稳定标志。

以 HSI 时钟源为例，针对 HSI 时钟稳定标志和时钟稳定中断标志，说明如下：

#### RCC\_HSI.STABLE

硬件在关闭 HSI 时清零，硬件在检测到 HSI 时钟稳定时置位。

#### RCC\_ISR.HSIRDY

当硬件检测到时钟由不稳定状态变为稳定状态时（即 RCC\_HSI.STABLE 标志位由 0 变为 1）置 1，为时钟稳定中断标志，用户可通过设置 RCC\_ICR.HSIRDY 为 0 清除该标志位。

### 4.3.4 时钟验证与输出

HC32F0653 支持将内部各种时钟信号输出到外部引脚。用户可利用该功能对当前系统的 CPU 运行频率、系统总线频率、外设工作频率等进行测量。

#### MCO\_OUT 引脚

输出 HSIOSC/LSI/PCLK/HCLK/HSIOSC12M 时钟信号，时钟输出到 MCO<sub>x</sub>\_OUT 引脚前可通过预分频器进行分频（有效分频系数为 1、2、8、64、128、256、512、1024），以便低带宽仪表能准确测量信号。

时钟输出框图如下图所示：

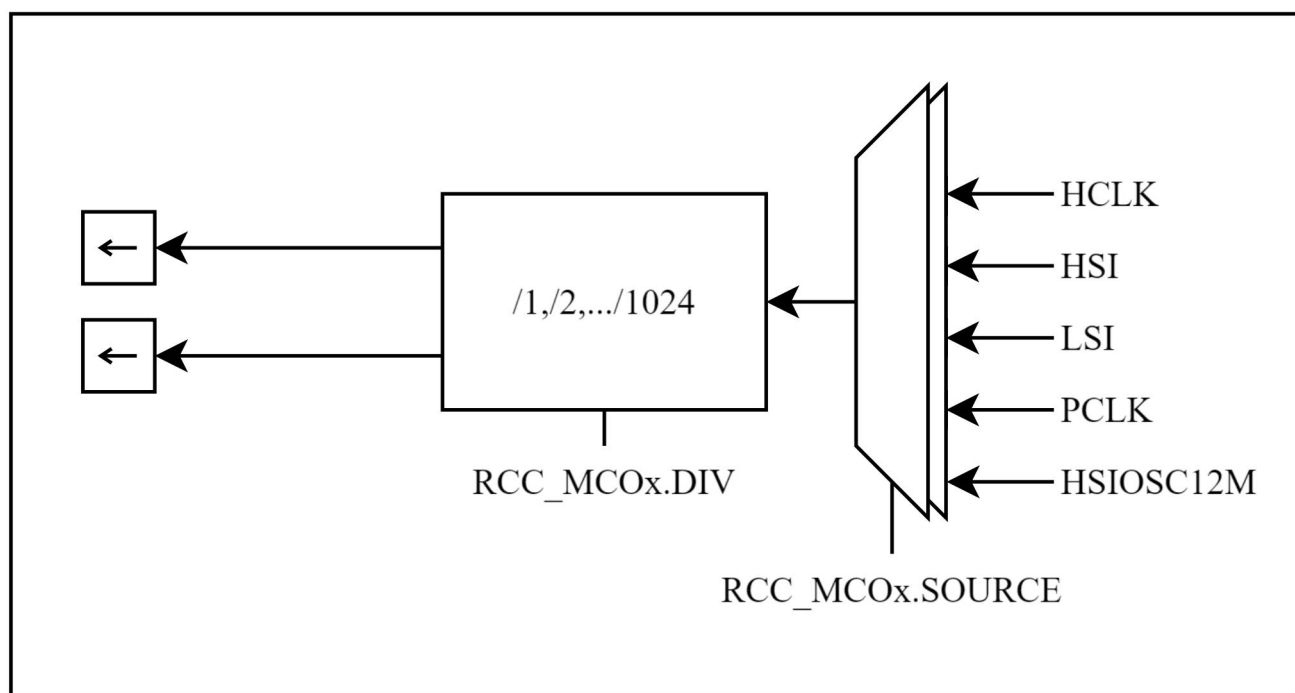


图 4-3 时钟输出

## 4.4 SysClk 系统时钟切换

系统时钟 SysClk 可选择 HSI、LSI 两种时钟源，通过对系统控制寄存器 RCC\_CR0 的 SYSCLK 位域进行设置，可在不同时钟源之间进行切换。

在低功耗模式切换 SysClk 的时钟源为低速时钟降低系统功耗，在正常使用状态切换 SysClk 的时钟源为高速时钟来快速响应用户使用需求。

时钟源安全切换规则如下：

HSI、LSI 两个时钟源可以相互切换。

系统时钟切换操作必须按照下文所描述的时钟切换流程进行，否则可能出现异常。

**注：系统时钟切换时需要同步配置 FLASH 控制寄存器 FLASH\_ACR.WAIT 读等待周期参数：系统时钟频率不大于 24MHz 则应设置 FLASH 控制寄存器 FLASH\_ACR.WAIT 为 0；系统时钟频率大于 24MHz 小于等于 48M 则应设置 FLASH 控制寄存器 FLASH\_ACR.WAIT 为 1，系统时钟频率大于 48MHz 则应设置 FLASH 控制寄存器 FLASH\_ACR.WAIT 为 2。**

#### 4.4.1 标准的时钟切换流程

标准时钟切换操作流程如下：

1. 配置新时钟源的稳定周期；
2. 设置新时钟源振荡器使能位为 1；
3. 根据当前时钟源和新时钟源两者中较高的频率，配置 FLASH\_ACR.WAIT 读等待周期参数；
4. 等待新时钟源输出稳定的频率（新时钟源的 STABLE 信号变为 1）；
5. 配置系统控制寄存器 RCC\_CR0.SYSCLK，选择系统时钟 SysClk 的时钟源为新时钟源；
6. 根据新时钟源的频率，配置 FLASH\_CR2.WAIT 读等待周期参数；
7. 关闭不再使用的时钟源。

以系统时钟从 HSI 切换为 LSI 为例，新旧时钟的切换过程如下图所示：



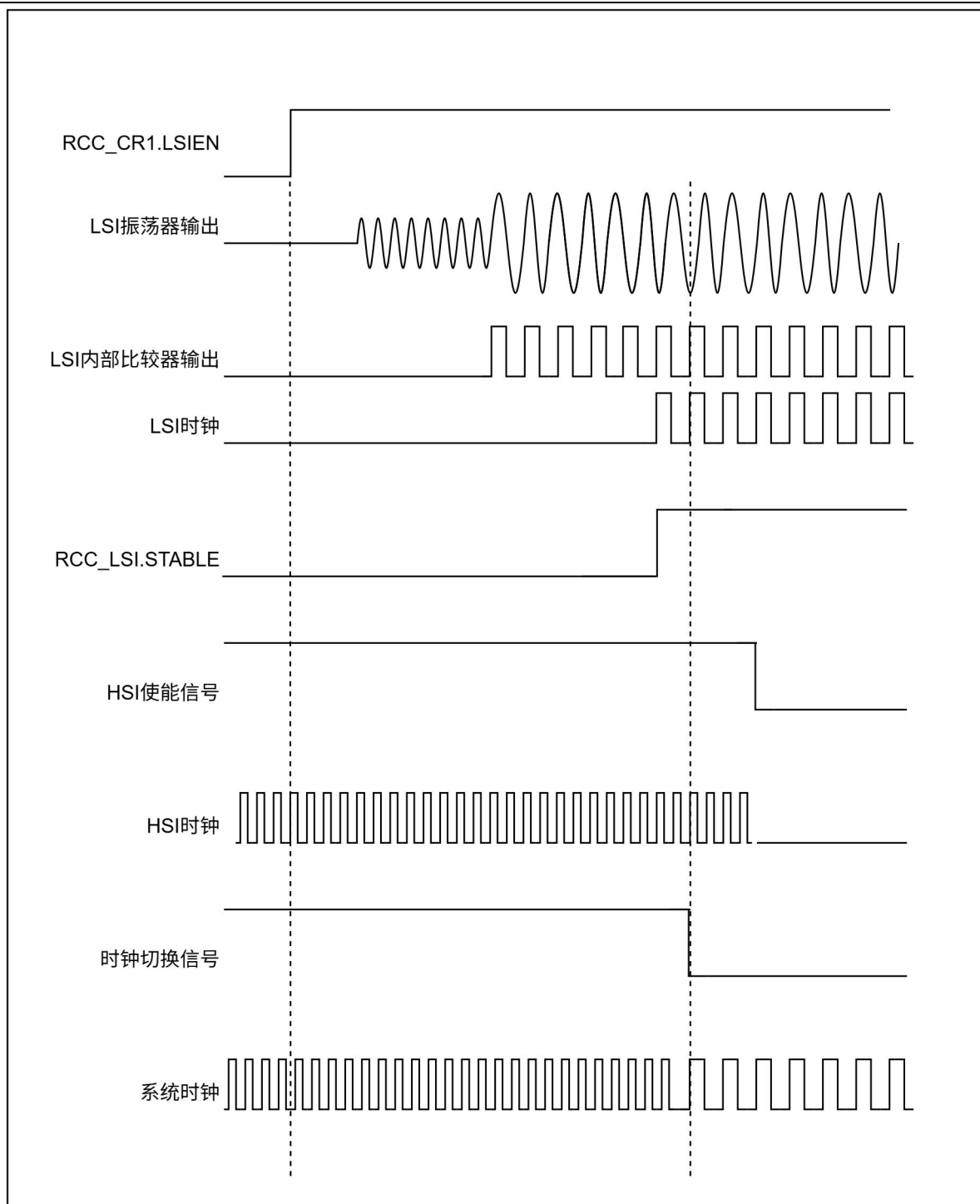


图 4-4 HSI 时钟切换为 LSI 时钟

#### 4.4.2 HSI 时钟不同频率间切换流程

HSI 时钟不同频率切换不需要启动新时钟源，只需要改变 HSI 时钟预分频器的分频比即可，切换速度快，切换前后无需等待时钟稳定时间。切换操作流程如下：

1. 根据当前时钟源和新时钟源两者中较高的频率，按要求配置 FLASH\_ACR.WAIT 读等待周期参数；
2. 读取内置高频时钟控制寄存器 RCC\_HSI 寄存器的值为 Value；
3. 根据需要的目标频率，设置内置高频时钟控制寄存器 RCC\_HSI.DIV 为对应的分频系数（注意

RCC\_HSI.TRIM 域要保持不变);

4. 根据新时钟源的频率, 按要求配置 FLASH\_CR2.WAIT 读等待周期参数。

#### 4.4.3 从其它时钟到切换到 LSI 示例

切换操作流程如下:

1. 读取内置低频时钟的时钟频率 FLASH\_LSIFREQ, FLASH\_LSIFREQ 地址: 0x1FFF FA90, 然后设置内置低频时钟控制寄存器 RCC\_LSI.WAITCYCLE 为合适的值;
2. 设置系统控制寄存器 RCC\_CR1.LSIEN 为 1, 使能 LSIRC 振荡电路;  
**注: RCC\_CR1 寄存器具有 KEY 保护特性, 写入的数据高 16bit 数据必须是 0x5A5A, 否则无法写入。**
3. 循环查询并等待内置低频时钟控制寄存器 RCC\_LSI.STABLE 稳定标志变为 1, 即等待 LSI 输出稳定时钟;
4. 设置系统控制寄存器 RCC\_CR0.SYSCLK 为 1, 将 SysClk 时钟来源切换为 LSI;  
**注: RCC\_CR0 寄存器具有 KEY 保护特性, 写入的数据高 16bit 数据必须是 0x5A5A, 否则无法写入。**
5. 配置 FLASH\_CR2.WAIT 读等待周期参数为 0;
6. 设置系统控制寄存器 RCC\_CR1.xxxEN 为 0, 关闭原时钟。

#### 4.4.4 从其它时钟到切换到 HSI 示例

切换操作流程如下:

1. 配置内置高频时钟控制寄存器 RCC\_HSI.TRIM、RCC\_HSI.WAITCYCLE 以及 RCC\_HSI.DIV 为合适的值;
2. 设置系统控制寄存器 RCC\_CR1.HSIEN 为 1, 使能 HSIOSC 时钟振荡电路;  
**注: RCC\_CR1 寄存器具有 KEY 保护特性, 写入的数据高 16bit 数据必须是 0x5A5A, 否则无法写入。**
3. 循环查询并等待内置高频时钟控制寄存器 RCC\_HSI.STABLE 稳定标志变为 1, 即等待 HSIOSC 输出稳定时钟;
4. 根据当前时钟和 HSI 两者中较高的频率, 按要求配置 FLASH\_CR2.WAIT 读等待周期参数;
5. 设置系统控制寄存器 RCC\_CR0.SYSCLK 为 0, 将 SysClk 时钟来源切换为 HSI;  
**注: RCC\_CR0 寄存器具有 KEY 保护特性, 写入的数据高 16bit 数据必须是 0x5A5A, 否则无法写入。**
6. 根据 HSI 的时钟频率, 按要求配置 FLASH\_CR2.WAIT 读等待周期参数;
7. 设置系统控制寄存器 RCC\_CR1.xxxEN 为 0, 关闭原时钟。

## 4.5 寄存器列表

RCC 基地址:  $\text{RCC\_BASE} = 0x4000\ 0800$

表 4-2 RCC 寄存器列表

寄存器名称	寄存器地址	寄存器描述
RCC_CR0	$\text{RCC\_BASE} + 0x00$	系统控制寄存器 0
RCC_CR1	$\text{RCC\_BASE} + 0x04$	系统控制寄存器 1
RCC_HSI	$\text{RCC\_BASE} + 0x0C$	内置高频时钟控制寄存器
RCC_LSI	$\text{RCC\_BASE} + 0x10$	内置低频时钟控制寄存器
RCC_IER	$\text{RCC\_BASE} + 0x18$	系统中断使能控制寄存器
RCC_ISR	$\text{RCC\_BASE} + 0x20$	系统中断标志寄存器
RCC_ICR	$\text{RCC\_BASE} + 0x28$	系统中断标志清除寄存器
RCC_AHBEN	$\text{RCC\_BASE} + 0x2C$	AHB 外设时钟使能控制寄存器
RCC_AHBRSRST	$\text{RCC\_BASE} + 0x30$	AHB 外设复位控制寄存器
RCC_APBEN	$\text{RCC\_BASE} + 0x34$	APB 外设时钟使能控制寄存器
RCC_APBRSRST	$\text{RCC\_BASE} + 0x38$	APB 外设复位控制寄存器
RCC_RESETFLAG	$\text{RCC\_BASE} + 0x4C$	系统复位标志寄存器
RCC_MCO0	$\text{RCC\_BASE} + 0x54$	系统时钟输出控制寄存器 0
RCC_MCO1	$\text{RCC\_BASE} + 0x58$	系统时钟输出控制寄存器 1

## 4.6 寄存器描述

有关寄存器描述里所使用的缩写，请参见 1 文档约定章节。

### 4.6.1 RCC\_CR0 系统控制寄存器 0

地址：请参见表 4-2 RCC 寄存器列表      复位值：0x0000 0000

位编号	位符号	权限	功能描述
31-16	KEY	WO	仅当 KEY 为 0x5A5A 时，对该寄存器的写操作有效
15-12	RFU	-	保留位，请保持默认值
11-4	HCLKDIV	RW	配置 HCLK 的时钟分频系数 0000 0000：设置 HCLK 的时钟源为 SysClk / 1 0000 0001：设置 HCLK 的时钟源为 SysClk / 1 0000 0010：设置 HCLK 的时钟源为 SysClk / 2 0000 0011：设置 HCLK 的时钟源为 SysClk / 3 0000 0100：设置 HCLK 的时钟源为 SysClk / 4 ... 1111 1110：设置 HCLK 的时钟源为 SysClk / 254 1111 1111：设置 HCLK 的时钟源为 SysClk / 255
3-2	PCLKDIV	RW	配置 PCLK 的时钟分频系数 00：设置 PCLK 的时钟源为 HCLK 01：设置 PCLK 的时钟源为 HCLK / 2 10：设置 PCLK 的时钟源为 HCLK / 4 11：设置 PCLK 的时钟源为 HCLK / 8
1	RFU	-	保留位，请保持默认值
0	SYSCLK	RW	配置 SysClk 的时钟来源 0：设置 SysClk 的时钟源为 HSI 1：设置 SysClk 的时钟源为 LSI

### 4.6.2 RCC\_CR1 系统控制寄存器 1

地址：请参见表 4-2 RCC 寄存器列表      复位值：0x0000 0001

位编号	位符号	权限	功能描述
31-16	KEY	WO	仅当 KEY 为 0x5A5A 时，对该寄存器的写操作有效
15-4	RFU	-	保留位，请保持默认值
3	LSIEN	RW	内部低速时钟 LSI 使能控制 0：关闭 1：使能 <i>注：当系统进入 Deep Sleep，此低速时钟不会自动关闭。</i> <i>注：当 IWDG 使能时，LSI 会被强制打开，且无法关闭。</i>
2-1	RFU	-	保留位，请保持默认值
0	HSIEN	RW	内部高速时钟 HSIOSC 使能控制 0：关闭 1：使能 <i>注：当系统进入 DeepSleep，此高速时钟会自动关闭。</i>

### 4.6.3 RCC\_HSI 内置高频时钟控制寄存器

地址：请参见表 4-2 RCC 寄存器列表 复位值：0x0000 0C00

位编号	位符号	权限	功能描述
31-16	KEY	WO	仅当 KEY 为 0x5A5A 时，对该寄存器的写操作有效
15	STABLE	RO	HSIOSC 时钟稳定状态位 0：HSIOSC 时钟尚未稳定 1：HSIOSC 时钟已经稳定
14-12	RFU	-	保留位，请保持默认值
11-10	DIV	RW	HSI 时钟与 HSIOSC 时钟分频系数配置 00：HSI = HSIOSC / 1 01：HSI = HSIOSC / 2 10：HSI = HSIOSC / 4 11：HSI = HSIOSC / 8
9	FREQ	RW	HSIOSC 频率配置 0：HSIOSC = 48M 1：HSIOSC = 60M 注：频率切换，需要等待 10us，系统才会再次运行。
8-7	RFU	-	保留位，请保持默认值
6-0	TRIM	RW	时钟频率调整，更改该寄存器位的数值即可调整 HSIOSC 的振荡频率。 TRIM 值每增加 1 则 HSIOSC 的振荡频率增加约 0.2%。FLASH 中已保存了 HSI 的校准值，上电后，芯片自动将 FLASH 内的校准值读出并写入 RCC_HSI.TRIM，获得精准的频率。 该寄存器也可以软件调整。 校准值地址：0x1FFF FA40

### 4.6.4 RCC\_LSI 内置低频时钟控制寄存器

地址：请参见表 4-2 RCC 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	功能描述
31-16	RFU	-	保留位，请保持默认值
15	STABLE	RO	LSI 时钟稳定状态位 0：LSI 时钟尚未稳定 1：LSI 时钟已经稳定
14-12	RFU	-	保留位，请保持默认值
11-10	WAITCYCLE	RW	内部低速时钟 LSI 稳定时间选择 00：4 个周期 01：16 个周期 10：64 个周期 11：256 个周期
9-0	RFU	-	保留位，请保持默认值

#### 4.6.5 RCC\_IER 系统中断使能控制寄存器

地址：请参见表 4-2 RCC 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	功能描述
31-16	KEY	WO	仅当 KEY 为 0x5A5A 时，对该寄存器的写操作有效
15-9	RFU	-	保留位，请保持默认值
3	LSIRDY	RW	LSI 稳定中断使能控制 0：禁止 LSI 稳定中断 1：使能 LSI 稳定中断
2-1	RFU	-	保留位，请保持默认值
0	HSIRDY	RW	HSIOSC 稳定中断使能控制 0：禁止 HSIOSC 稳定中断 1：使能 HSIOSC 稳定中断

#### 4.6.6 RCC\_ISR 系统中断标志寄存器

地址：请参见表 4-2 RCC 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	功能描述
31-4	RFU	-	保留位，请保持默认值
3	LSIRDY	RO	LSI 时钟稳定标志位 0：LSI 时钟尚未稳定 1：LSI 时钟已经稳定 <i>注：当打开 LSI 时钟稳定后且 RCC_IER.LSIRDY 置 1 时由硬件置 1。</i>
2-1	RFU	-	保留位，请保持默认值
0	HSIRDY	RO	HSIOSC 时钟稳定标志位 0：HSIOSC 时钟尚未稳定 1：HSIOSC 时钟已经稳定 <i>注：当打开 HSI 时钟稳定后且 RCC_IER.HSIRDY 置 1 时由硬件置 1。</i>

#### 4.6.7 RCC\_ICR 系统中断标志清除寄存器

地址：请参见表 4-2 RCC 寄存器列表 复位值：0x0000 0009

位编号	位符号	权限	功能描述
31-4	RFU	-	保留位，请保持默认值
3	LSIRDY	R1W0	LSI 时钟稳定标志位清零控制 W0：清除 ISR 寄存器中的相应标志 W1：无功能
2-1	RFU	-	保留位，请保持默认值
0	HSIRDY	R1W0	HSIOSC 时钟稳定标志位清零控制 W0：清除 ISR 寄存器中的相应标志 W1：无功能

#### 4.6.8 RCC\_AHBEN AHB 外设时钟使能控制寄存器

地址：请参见表 4-2 RCC 寄存器列表      复位值：0x0000 0000

位编号	位符号	权限	功能描述
31-11	RFU	-	保留位，请保持默认值
10	MAP	RW	MAP 端口映射配置时钟及工作时钟使能控制 0：关闭 1：使能
9-6	RFU	-	保留位，请保持默认值
5	GPIOB	RW	GPIOB 端口配置时钟及工作时钟使能控制 0：关闭 1：使能
4	GPIOA	RW	GPIOA 端口配置时钟及工作时钟使能控制 0：关闭 1：使能
3	ALU	RW	ALU 配置时钟及工作时钟使能控制 0：关闭 1：使能
2	CRC	RW	CRC 模块配置时钟及工作时钟使能控制 0：关闭 1：使能
1-0	RFU	-	保留位，请保持默认值

#### 4.6.9 RCC\_APBEN APB 外设时钟使能控制寄存器

地址：请参见表 4-2 RCC 寄存器列表      复位值：0x0000 0000

位编号	位符号	权限	功能描述
31-23	RFU	-	保留位，请保持默认值
22	FDA	RW	FDA 模块配置时钟使能控制 0：关闭 1：使能
21	VC	RW	VC 模块配置时钟使能控制 0：关闭 1：使能
20	DAC	RW	DAC 模块配置时钟及工作时钟使能控制 0：关闭 1：使能
19	ADC	RW	ADC 模块配置时钟及工作时钟使能控制 0：关闭 1：使能
18	RFU	-	保留位，请保持默认值

17	I2C	RW	I2C 模块配置时钟及工作时钟使能控制 0: 关闭 1: 使能
16	RFU	-	保留位, 请保持默认值
15	SPI	RW	SPI 模块配置时钟及工作时钟使能控制 0: 关闭 1: 使能
14-13	RFU	-	保留位, 请保持默认值
12	UART1	RW	UART1 模块配置时钟及工作时钟使能控制 0: 关闭 1: 使能
11	UART0	RW	UART0 模块配置时钟及工作时钟使能控制 0: 关闭 1: 使能
10	RFU	-	保留位, 请保持默认值
9	IWDT	RW	IWDT 模块配置时钟使能控制 0: 关闭 1: 使能
8	BTIM0/1/2	RW	BTIM0/1/2 模块配置时钟及工作时钟使能控制 0: 关闭 1: 使能
7	RFU	-	保留位, 请保持默认值
6	GTIM	RW	GTIM 模块配置时钟及工作时钟使能控制 0: 关闭 1: 使能
5	ATIM	RW	ATIM 模块配置时钟及工作时钟使能控制 0: 关闭 1: 使能
4-0	RFU	-	保留位, 请保持默认值



#### 4.6.10 RCC\_AHBRST AHB 外设复位控制寄存器

地址：请参见表 4-2 RCC 寄存器列表      复位值：0x0000 063C

位编号	位符号	权限	功能描述
31-11	RFU	-	保留位，请保持默认值
10	MAP	RW	MAP 模块复位控制 0：模块处于复位状态 1：模块正常工作
9-6	RFU	-	保留位，请保持默认值
5	GPIOB	RW	GPIOB 模块复位控制 0：模块处于复位状态 1：模块正常工作
4	GPIOA	RW	GPIOA 模块复位控制 0：模块处于复位状态 1：模块正常工作
3	ALU	RW	ALU 模块复位控制 0：模块处于复位状态 1：模块正常工作
2	CRC	RW	CRC 模块复位控制 0：模块处于复位状态 1：模块正常工作
1-0	RFU	-	保留位，请保持默认值

#### 4.6.11 RCC\_APBRST APB 外设复位控制寄存器

地址：请参见表 4-2 RCC 寄存器列表      复位值：0x007A 9BE0

位编号	位符号	权限	功能描述
31-23	RFU	-	保留位，请保持默认值
22	FDA	RW	FDA 模块复位控制 0：模块处于复位状态 1：模块正常工作
21	VC	RW	VC 模块复位控制 0：模块处于复位状态 1：模块正常工作
20	DAC	RW	DAC 模块复位控制 0：模块处于复位状态 1：模块正常工作
19	ADC	RW	ADC 模块复位控制 0：模块处于复位状态 1：模块正常工作
18	RFU	-	保留位，请保持默认值

17	I2C	RW	I2C 模块复位控制 0: 模块处于复位状态 1: 模块正常工作
16	RFU	-	保留位, 请保持默认值
15	SPI	RW	SPI 模块复位控制 0: 模块处于复位状态 1: 模块正常工作
14-13	RFU	-	保留位, 请保持默认值
12	UART1	RW	UART1 模块复位控制 0: 模块处于复位状态 1: 模块正常工作
11	UART0	RW	UART0 模块复位控制 0: 模块处于复位状态 1: 模块正常工作
10	RFU	-	保留位, 请保持默认值
9	IWDT	RW	IWDT 模块复位控制 0: 模块处于复位状态 1: 模块正常工作
8	BTIM0/1/2	RW	BTIM0/1/2 模块复位控制 0: 模块处于复位状态 1: 模块正常工作
7	RFU	-	保留位, 请保持默认值
6	GTIM	RW	GTIM 模块复位控制 0: 模块处于复位状态 1: 模块正常工作
5	ATIM	RW	ATIM 模块复位控制 0: 模块处于复位状态 1: 模块正常工作
4-0	RFU	-	保留位, 请保持默认值

## 4.6.12 RCC\_RESETFLAG 系统复位标志寄存器

地址：请参见表 4-2 RCC 寄存器列表      复位值：0x---- ----

位编号	位符号	权限	功能描述
31-10	RFU	-	保留位，请保持默认值
9	SYSRESETREQ	RW0	Cortex-M0+ CPU SYSRESETREQ 软复位标志 0: 未发生 CPU SYSRESETREQ 软复位 1: 已发生 CPU SYSRESETREQ 软复位 写 0 清除，写 1 无效
8	LOCKUP	RW0	Cortex-M0+ CPU Lockup 复位标志 0: 未发生 Lockup 复位 1: 已发生 Lockup 复位 写 0 清除，写 1 无效
7	RFU	-	保留位，请保持默认值
6	RSTB	RW0	NRST 引脚复位标志 0: 未发生引脚复位 1: 已发生引脚复位 写 0 清除，写 1 无效
5	RFU	-	保留位，请保持默认值
4	IWDT	RW0	IWDT 复位标志 0: 未发生 IWDT 复位 1: 已发生 IWDT 复位 写 0 清除，写 1 无效
3-2	RFU	-	保留位，请保持默认值
1	BOR	RW0	BOR 复位标志 0: 未发生 BOR 复位 1: 已发生 BOR 复位 写 0 清除，写 1 无效
0	POR	RW0	POR 复位标志 0: 未发生 POR 复位 1: 已发生 POR 复位 写 0 清除，写 1 无效

#### 4.6.13 RCC\_MCOx 系统时钟输出控制寄存器(x=0,1)

地址：请参见表 4-2 RCC 寄存器列表      复位值：0x0000 0000

位编号	位符号	权限	功能描述
31-7	RFU	-	保留位，请保持默认值
6-4	DIV	RW	MCO 输出分频控制 000: 1 分频 001: 2 分频 010: 8 分频 011: 64 分频 100: 128 分频 101: 256 分频 110: 512 分频 111: 1024 分频
3	RFU	-	保留位，请保持默认值
2-0	SOURCE	RW	MCO 输出信号来源配置 000: 无输出 001: HCLK 011: HSIOSC 100: LSI 101: PCLK 110: HSIOSC/4=12M

## 5 系统控制（SYSCTRL）

### 5.1 概述

HC32F0653 有一组系统控制寄存器。系统控制寄存器的主要用途如下：

- 控制安全性功能
- DEBUG 模式下定时器计数配置
- DEBUG\_ID

### 5.2 寄存器列表

SYSCTRL 基地址：SYSCTRL\_BASE = 0x4000 0400

表 5-1 SYSCTRL 寄存器列表

寄存器名称	寄存器地址	寄存器描述
SYSCTRL_CR	SYSCTRL_BASE + 0x00	系统控制寄存器
SYSCTRL_DEBUG	SYSCTRL_BASE + 0x04	调试状态定时器控制寄存器
SYSCTRL_DEBUG_ID	SYSCTRL_BASE + 0x08	DEBUG_ID 寄存器

### 5.3 寄存器描述

有关寄存器描述里所使用的缩写，请参见 1 文档约定章节。

#### 5.3.1 SYSCTRL\_CR 系统控制寄存器

地址：请参见表 5-1 SYSCTRL 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	功能描述
31-16	KEY	WO	仅当 KEY 为 0x5A5A 时，对该寄存器的写操作有效
15	SRAM	RW	SRAM 奇偶校验错误触发 ATIM 刹车使能控制 0：关闭 1：使能
14	RFU	-	保留位，请保持默认值
13	DSBRKEN	RW	Deep Sleep 触发 ATIM 刹车使能控制 0：关闭 1：使能
12	CLLBRKEN	RW	产生 HardFault 或 Cortex-M0+LockUp 标志触发 ATIM 刹车使能控制 0：关闭 1：使能
11-4	RFU	-	保留位，请保持默认值
3	WAKEUPCLK	RW	Deep Sleep 唤醒时，系统时钟的来源配置 0：保持原系统时钟来源 1：切换系统时钟来源为 HSI 6 MHz，原系统时钟来源保持使能

2	LOCKUP	RW	Cortex-M0+ LockUp 功能配置 0: 关闭 1: 使能 注: 使能该功能, 则 CPU 读到无效指令时会复位 MCU。
1	SWDIO_N	RW	SWD 端口功能配置 0: 使能 SWD 端口, GPIO 功能不可用 1: 使能 GPIO 端口, SWD 功能不可用 注: 在调试模式下, 该位写 1 无效
0	RFU	-	保留位, 请保持默认值

### 5.3.2 SYSCTRL\_DEBUG 调试状态定时器控制寄存器

地址: 请参见表 5-1 SYSCTRL 寄存器列表      复位值: 0x0000 0227

位编号	位符号	权限	功能描述
31-11	RFU	-	保留位, 请保持默认值
9	IWDT	RW	调试状态下, 断点期间, IWDT 计数功能配置 0: 正常计数 1: 暂停计数
8-6	RFU	-	保留位, 请保持默认值
5	BTIM0/1/2	RW	调试状态下, 断点期间, BTIM0/1/2 计数功能配置 0: 正常计数 1: 暂停计数
4-2	RFU	-	保留位, 请保持默认值
1	GTIM0	RW	调试状态下, 断点期间, GTIM 计数功能配置 0: 正常计数 1: 暂停计数
0	ATIM	RW	调试状态下, 断点期间, ATIM 计数功能配置 0: 正常计数 1: 暂停计数

### 5.3.3 SYSCTRL\_DEBUGID\_DEBUG\_ID 寄存器

地址：请参见表 5-1 SYSCTRL 寄存器列表      复位值：0XXXXX XXXX

位编号	位符号	权限	功能描述
31-28	CM	RW	内核代号 0000: M0 0001: M0+ 0010: 保留 0011: M3 0100: M4 ...
27-20	DEV_ID	RW	芯片类型代号 0000 0000: 小容量 0000 0001: 中容量 0000 0010: 大容量 0000 0011: 互联网 0000 0100: 电机 ...
19-16	CM	RW	内核代号 0000: M0 0001: M0+ 0010: 保留 0011: M3 0100: M4 ...
15-14	CUSTOM	RW	00: 公司定制代号 01: 合作商定制代号 ...
13-4	DEV_NUM	RW	同系列芯片累加值
3-0	REV_ID	RW	晶圆版本号

## 6 中断 (NVIC)

### 6.1 概述

ARM®Cortex®-M0+内核的嵌套向量中断控制器 (NVIC)，用于管理中断和异常。NVIC和处理器内核紧密相连，可以实现低延迟的异常和中断处理。

处理器支持最多32个中断请求(IRQ)输入，支持多个内部异常。

本章节只介绍了处理器的21个外部中断请求（IRQ0 ~ IRQ20），处理器内部异常的具体情况请参考“ARM® Cortex®-M0+Technical Reference Manual”与“ARM® v6-MArchitecture Reference Manual”。

### 6.2 主要特性

- 16 个内部异常
- 21 个可屏蔽外部中断
- 4 个可编程的优先级
- 低延时的异常和中断处理
- 支持中断嵌套
- 中断向量表重映射

### 6.3 SysTick 校准值寄存器

SysTick 校准值设置为 6000。当 SysTick 时钟设置为 6 MHz（最大  $f_{HCLK}/8$ ）时，会产生 1 ms 时间基准。

### 6.4 中断优先级

外部中断可设置 4 级优先级，最高优先级为“0”，最低优先级为“3”，默认值为“0”。

当处理器正在执行一个中断处理程序时，如果出现一个更高优先级的中断，那么这个中断就被抢占。如果出现的中断的优先级和正在处理的中断的优先级相同或更低，这个中断就不会被抢占，但是新中断的状态就变为挂起。

如果多个挂起的中断具有相同的优先级，中断编号越小的挂起中断优先处理。例如，如果IRQ[0]和IRQ[1]均挂起时，并且两者的优先级相同，那么先处理 IRQ[0]。



## 6.5 中断向量表

ARM®Cortex®-M0+ 响应中断时，处理器自动从存储器的中断向量表中取出中断服务程序（ISR）的起始地址。中断向量表包括主栈指针（MSP）的初始值，内部异常和外部中断的服务程序入口地址。每个中断向量占用1个字（4字节），中断向量的存储地址为向量编号乘以4，如下表所示：

表 6-1中断向量表

向量编号	外部中断 (IRQ#)	优先级	中断源	简介	地址
0	-	-	-	MSP 初始值	0x0000 0000
1	-	-3	Reset	复位向量	0x0000 0004
2	-	-2	NMI	不可屏蔽中断 2	0x0000 0008
3	-	-1	HardFault	硬件错误异常（fault）	0x0000 000C
4-10	-	-	-	保留	0x00000010~002B
11	-	可配置	SVCall	通过 SWI 指令调用的管理程序	0x0000 002C
12-13	-	-	-	保留	0x00000030~0037
14	-	可配置	PendSV	系统服务的可挂起请求	0x0000 0038
15	-	可配置	SysTick	系统滴答定时器	0x0000 003C
16	0	可配置	IWDT	独立看门狗中断	0x0000 0040
17	1	可配置	FLASH	FLASH 全局中断	0x0000 0044
18	2	可配置	SRAM	SRAM 全局中断	0x0000 0048
19	3	可配置	RCC	RCC 全局中断	0x0000 004C
20	4	可配置	GPIOA	GPIOA 全局中断	0x0000 0050
21	5	可配置	GPIOB	GPIOB 全局中断	0x0000 0054
22	6	可配置	ADC	ADC 全局中断	0x0000 0058
23	7	可配置	ATIM_BRK	ADC 刹车中断	0x0000 005C
24	8	可配置	ATIM_UP	ATIM 更新中断	0x0000 0060
25	9	可配置	ATIM_TRG ATIM_COM ATIM_DIR ATIM_IDX	ATIM 触发中断 ATIM COM 中断 ATIM 编码器方向改变中断 ATIM 编码器索引中断	0x0000 0064
26	10	可配置	ATIM_CC	ATIM 捕获比较中断	0x0000 0068
27	11	可配置	VC0	VC0 全局中断	0x0000 006C
28	12	可配置	VC1	VC1 全局中断	0x0000 0070
29	13	可配置	GTIM	GTIM 全局中断	0x0000 0074
30	14	可配置	BTIM0	BTIM0 全局中断	0x0000 0078
31	15	可配置	BTIM1	BTIM1 全局中断	0x0000 007C
32	16	可配置	BTIM2	BTIM2 全局中断	0x0000 0080
33	17	可配置	I2C	I2C全局中断	0x0000 0084
34	18	可配置	SPI	SPI 全局中断	0x0000 0088
35	19	可配置	UART0	UART0 全局中断	0x0000 008C
36	20	可配置	UART1	UART1 全局中断	0x0000 0090

**注：**由于部分外设的中断复用了一个IRQ中断源，用户在中断服务程序中应先检查中断标志位，以确定产生中断的外设。

## 6.6 中断相关寄存器

### 6.6.1 NVIC 中断使能和禁止使能

ARM® Cortex-M0+处理器支持最多32个外部中断源，分别对应中断使能设置寄存器 NVIC\_ISER 的32个使能位，和中断使能清除寄存器 NVIC\_ICER 的32个禁止位。将使能位置1，允许中断；将禁止位置1，禁止中断。

上文中NVIC中断使能仅针对处理器NVIC而言，外设的中断是否使能，还受相应外设的中断控制寄存器控制。

### 6.6.2 NVIC 中断挂起和清除挂起

在中断发生时，如果系统正在处理与之相同优先级或更高优先级的中断，系统将不会立即处理此中断，而是将中断的状态设置为挂起，保存在中断挂起状态寄存器中；在处理器未进入此中断处理之前，如没有手动清除挂起状态，该状态将会一直保持有效。当处理器开始进入中断处理时，硬件会自动清除相应的中断挂起状态。

用户可通过设置中断挂起设置寄存器 NVIC\_ISPR 的对应位，将此中断的状态设置为挂起状态，如果系统没有正在处理与之相同优先级或更高优先级的中断，此中断将被立即响应并处理。

用户可以通过设置中断挂起清除寄存器 NVIC\_ICPR 的对应位，将此中断的状态设置为挂起清除状态。

### 6.6.3 NVIC 中断优先级

中断优先级控制寄存器 NVIC\_IPR0~NVIC\_IPR7，用于设置 IRQ0~IRQ17 的中断优先级，每个中断源使用8位，在 HC32F0653 中仅使用了高两位，最多可设置4个中断优先级。

**注：**ARM® Cortex-M0+ 的中断优先级寄存器的设置应在中断使能之前，用户不可在中断使能之后改变中断优先级，这将导致不可预知的结果。

### 6.6.4 NVIC 中断屏蔽

在某些特殊场合，需要禁止所有中断，可以使用中断屏蔽寄存器 PRIMASK 实现。PRIMASK 只有最低1位有效，将此位置1，除了 NMI 和硬件错误异常之外的所有外部中断和异常都被禁止；清0后，允许响应中断和异常。该位复位后默认为0。

ARM® Cortex-M0+ 有专用的 ARM 指令用于修改 PRIMASK 寄存器，CPSIE i 和 CPSID i，详细请参考《ARM® v6-M Architecture Reference Manual》。

汇编指令示例参考：

CPSIE i；清除 PRIMASK （使能中断）

CPSID i；设置 PRIMASK （禁止中断）

C 语言（调用 CMSIS 设备驱动库）示例参考：

void \_\_enable\_irq(void); // 清除 PRIMASK

void \_\_disable\_irq(void); // 设置 PRIMASK

## 6.6.5 外设中断使能

外设模块一般都有各自的中断使能寄存器，在使用中断时，必须首先打开外设中断使能，同时参见 表 6-1中断向量表 打开该中断源的 NVIC 中断使能。具体外设的中断使能，请参阅相关外设模块章节描述。

## 6.7 寄存器列表

NVIC 基地址：NVIC\_BASE = 0xE000 E000

表 6-2 NVIC 寄存器列表

寄存器名称	寄存器地址	寄存器描述
NVIC_ISER	NVIC_BASE + 0x100	IRQ0~IRQ17 中断使能设置寄存器
NVIC_ICER	NVIC_BASE + 0x180	IRQ0~IRQ17 中断使能清除寄存器
NVIC_ISPR	NVIC_BASE + 0x200	IRQ0~IRQ17 中断挂起设置寄存器
NVIC_ICPR	NVIC_BASE + 0x280	IRQ0~IRQ17 中断挂起清除寄存器
NVIC_IPR0	NVIC_BASE + 0x400	IRQ0~IRQ3 中断优先级控制寄存器 0
NVIC_IPR1	NVIC_BASE + 0x404	IRQ4~IRQ7 中断优先级控制寄存器 1
NVIC_IPR2	NVIC_BASE + 0x408	IRQ8~IRQ11 中断优先级控制寄存器 2
NVIC_IPR3	NVIC_BASE + 0x40C	IRQ12~IRQ15 中断优先级控制寄存器 3
NVIC_IPR4	NVIC_BASE + 0x410	IRQ16~IRQ19 中断优先级控制寄存器 4
NVIC_IPR5	NVIC_BASE + 0x414	IRQ20 中断优先级控制寄存器 5

## 6.8寄存器描述

有关寄存器描述里所使用的缩写，请参见 1 文档约定章节。

### 6.8.1 NVIC\_ISER 中断使能设置寄存器

地址：请参见表 6-2 NVIC 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
20-0	SETIRQ	RW	<p>设置使能外部中断 IRQ0 ~ IRQ20；写“1”置位，写“0”无效。</p> <p>[0]: IRQ0</p> <p>[1]: IRQ1</p> <p>[2]: IRQ2</p> <p>.....</p> <p>[20]: IRQ20</p> <p>读出值表示当前中断使能状态</p>

## 6.8.2 NVIC\_ICER 中断使能清除寄存器

地址：请参见表 6-2 NVIC 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-21	RFU	-	保留位，请保持默认值
20-0	CLRIRQ	RW	<p>设置禁止外部中断 IRQ0 ~ IRQ20；写“1”置位，写“0”无效。</p> <p>[0]: IRQ0</p> <p>[1]: IRQ1</p> <p>[2]: IRQ2</p> <p>.....</p> <p>[17]: IRQ20</p> <p>读出值表示当前中断使能状态</p>

## 6.8.3 NVIC\_ISPR 中断挂起设置寄存器

地址：请参见表 6-2 NVIC 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-21	RFU	-	保留位，请保持默认值
20:0	SETPEND	RW	<p>设置外部中断 IRQ0 ~ IRQ20 的挂起状态；写“1”置位，写“0”无效。</p> <p>[0]: IRQ0</p> <p>[1]: IRQ1</p> <p>[2]: IRQ2</p> <p>.....</p> <p>[14]: IRQ20</p> <p>读出值表示当前中断挂起状态</p>

## 6.8.4 NVIC\_ICPR 中断挂起清除寄存器

地址：请参见表 6-2 NVIC 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-18	RFU	-	保留位，请保持默认值
17:0	CLRPEND	RW	<p>清除外部中断 IRQ0 ~ IRQ17 的挂起状态；写“1”清除，写“0”无效。</p> <p>[0]: IRQ0</p> <p>[1]: IRQ1</p> <p>[2]: IRQ2</p> <p>.....</p> <p>[17]: IRQ17</p> <p>读出值表示当前中断挂起状态</p>

## 6.8.5 NVIC\_IPR0 中断优先级控制寄存器 0

地址：请参见表 6-2 NVIC 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
14:30	IPRIRQ3	RW	中断 IRQ3 的优先级，00 优先级最高，11 优先级最低
29:24	RFU	-	保留位，请保持默认值
23:22	IPRIRQ2	RW	中断 IRQ2 的优先级，00 优先级最高，11 优先级最低
21:16	RFU	-	保留位，请保持默认值
15:14	IPRIRQ2	RW	中断 IRQ1 的优先级，00 优先级最高，11 优先级最低
13:8	RFU	-	保留位，请保持默认值
7:6	IPRIRQ0	RW	中断 IRQ0 的优先级，00 优先级最高，11 优先级最低
5:0	RFU	-	保留位，请保持默认值

## 6.8.6 NVIC\_IPR1 中断优先级控制寄存器 1

地址：请参见表 6-2 NVIC 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
14:30	IPRIRQ7	RW	中断 IRQ7 的优先级，00 优先级最高，11 优先级最低
29:24	RFU	-	保留位，请保持默认值
23:22	IPRIRQ6	RW	中断 IRQ6 的优先级，00 优先级最高，11 优先级最低
21:16	RFU	-	保留位，请保持默认值
15:14	IPRIRQ5	RW	中断 IRQ5 的优先级，00 优先级最高，11 优先级最低
13:8	RFU	-	保留位，请保持默认值
7:6	IPRIRQ4	RW	中断 IRQ4 的优先级，00 优先级最高，11 优先级最低
5:0	RFU	-	保留位，请保持默认值

## 6.8.7 NVIC\_IPR2 中断优先级控制寄存器 2

地址：请参见表 6-2 NVIC 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31:30	IPRIRQ11	RW	中断 IRQ11 的优先级，00 优先级最高，11 优先级最低
29:24	RFU	-	保留位，请保持默认值
23:22	IPRIRQ10	RW	中断 IRQ10 的优先级，00 优先级最高，11 优先级最低
21:16	RFU	-	保留位，请保持默认值
15:14	IPRIRQ9	RW	中断 IRQ9 的优先级，00 优先级最高，11 优先级最低
13:8	RFU	-	保留位，请保持默认值
7:6	IPRIRQ8	RW	中断 IRQ8 的优先级，00 优先级最高，11 优先级最低
5:0	RFU	-	保留位，请保持默认值

## 6.8.8 NVIC\_IPR3 中断优先级控制寄存器 3

地址：请参见表 6-2 NVIC 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31:30	IPRIRQ15	RW	中断 IRQ15 的优先级, 00 优先级最高, 11 优先级最低
29:24	RFU	-	保留位, 请保持默认值
23:22	IPRIRQ14	RW	中断 IRQ14 的优先级, 00 优先级最高, 11 优先级最低
21:16	RFU	-	保留位, 请保持默认值
15:14	IPRIRQ13	RW	中断 IRQ13 的优先级, 00 优先级最高, 11 优先级最低
13:8	RFU	-	保留位, 请保持默认值
7:6	IPRIRQ12	RW	中断 IRQ12 的优先级, 00 优先级最高, 11 优先级最低
5:0	RFU	-	保留位, 请保持默认值

## 6.8.9 NVIC\_IPR3 中断优先级控制寄存器 4

地址：请参见表 6-2 NVIC 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31:30	IPRIRQ19	RW	中断 IRQ19 的优先级, 00 优先级最高, 11 优先级最低
29:24	RFU	-	保留位, 请保持默认值
23:22	IPRIRQ18	RW	中断 IRQ18 的优先级, 00 优先级最高, 11 优先级最低
21:16	RFU	-	保留位, 请保持默认值
15:14	IPRIRQ17	RW	中断 IRQ17 的优先级, 00 优先级最高, 11 优先级最低
13:8	RFU	-	保留位, 请保持默认值
7:6	IPRIRQ16	RW	中断 IRQ16 的优先级, 00 优先级最高, 11 优先级最低
5:0	RFU	-	保留位, 请保持默认值

## 6.8.10 NVIC\_IPR3 中断优先级控制寄存器 5

地址：请参见表 6-2 NVIC 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31:8	RFU	-	保留位, 请保持默认值
7:6	IPRIRQ20	RW	中断 IRQ20 的优先级, 00 优先级最高, 11 优先级最低
5:0	RFU	-	保留位, 请保持默认值

## 7 RAM 存储器（RAM）

### 7.1 概述

HC32F0653 内部集成 8KB 嵌入式 RAM 供用户使用,用来存放程序执行过程中的各种数据。RAM 的起始地址为 0x2000 0000, 数据在 RAM 中以小端模式存储, 即最低字节地址空间存放数据的最低有效字节数据。

### 7.2 主要特性

- 支持以字节（8bit）、半字（16bit）或全字（32 bit）3 种位宽进行访问
- 零等待延迟, 能够被 CPU 以最大的系统时钟频率进行访问
- 支持奇校验功能

### 7.3 RAM 存储器操作

用户可执行的 RAM 存储器操作包括: 读操作、写操作。

对RAM的读写操作支持8bit、16bit 和 32bit三种位宽, 用户程序可以通过直接访问绝对地址的方式完成读写, 但要注意读写的数据位宽必须和对应地址边界对齐, 否则读写操作无效, 并会导致 HardFault硬件错误异常。

#### 7.3.1 读操作

读操作支持 3种不同位宽, 可采用直接访问绝对地址方式读取, 但要注意读取的数据位宽必须和对应地址边界对齐。

代码示例:

8bit 读:

```
tempdata = * ( ( uint8_t * ) 0x2000 0001 );
```

16bit 读:

```
tempdata = * ( ( uint16_t * ) 0x2000 0002 );
```

32bit 读:

```
tempdata = * ( ( uint32_t * ) 0x2000 0004 )
```

#### 7.3.2 写操作

写操作支持 3 种不同位宽, 可采用直接访问绝对地址的方式写入数据, 但要注意写入的数据位宽必须和对应地址 边界对齐。

代码示例:

8bit 写:

```
tempdata = ( ( uint8_t * ) 0x2000 0001 ) = 0x12;
```

16bit 写:

```
tempdata = ( ( uint16_t * ) 0x2000 0002 ) = 0x1234;
```

32bit 写:

```
tempdata = ( ( uint32_t * ) 0x2000 0004 ) = 0x1234 5678。
```

## 7.4 奇校验功能

HC32F0653 支持 RAM 的奇校验功能，上电后奇校验功能默认打开，用户不可配置。

每字节 RAM 数据实际存放在 9bit 的物理空间中，包括 8bit 数据位和 1bit 奇校验位。

CPU 在对 RAM 进行写入时，RAM 的奇校验单元会计算校验位并写入对应的校验位空间。在读取 RAM 数据时，数据连同校验位一起被读取，CPU 对数据进行每字节的奇校验，并将计算的校验位和读取的校验位进行比较，如果一致则说明 RAM 中数据正确，如果不一致则奇校验错误标志 RAM\_ISR.PARITY 被置位，如果设置 RAM 奇校验出错中断使能控制位 RAM\_IER.PARITY 为 1，CPU 会响应中断服务。用户程序可设置 RAM\_ICR.PARITY 为 0 来清除奇校验错误标志。

用户可通过读取奇校验出错地址寄存器 RAM\_ADDR，以获取发生奇校验错误的 RAM 地址。

## 7.5 寄存器列表

RAM 基地址：RAM\_BASE = 0x4000 0400

表 7-1 RAM 寄存器列表

寄存器名称	寄存器地址	寄存器描述
RAM_IER	RAM_BASE + 0x00	中断使能控制寄存器
RAM_ADDR	RAM_BASE + 0x04	奇校验出错地址寄存器
RAM_ISR	RAM_BASE + 0x08	中断标志寄存器
RAM_ICR	RAM_BASE + 0x0C	中断标志清除寄存器



## 7.6 寄存器描述

有关寄存器描述里所使用的缩写，请参见 1 文档约定章节。

### 7.6.1 RAM\_IER 中断使能控制寄存器

地址：请参见表 7-1 RAM 寄存器列表 复位值：0x0000 0001

位编号	位符号	权限	说明
31:2	RFU	-	保留位，请保持默认值
1	PARITY	RW	RAM 奇校验出错中断使能控制 0：禁止 1：使能
0	EN	RO	RAM 奇校验使能标志 0：禁止 1：使能

### 7.6.2 RAM\_ADDR 奇校验出错地址寄存器

地址：请参见表 7-1 RAM 寄存器列表 复位值：0x2000 0000

位编号	位符号	权限	说明
31:0	ADDR	RO	奇校验出错的 RAM 所在的地址 注：如果读一次 RAM 产生多个校验错误，则返回第一个字节出错地址

### 7.6.3 RAM\_ISR 中断标志寄存器

地址：请参见表 7-1 RAM 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31:1	RFU	-	保留位，请保持默认值
0	PARITY	RO	奇校验错误标志 0：未发生奇校验错误 1：已发生奇校验错误

### 7.6.4 RAM\_ICR 中断标志清除寄存器

地址：请参见表 7-1 RAM 寄存器列表 复位值：0x0000 0001

位编号	位符号	权限	说明
31:1	RFU	-	保留位，请保持默认值
0	PARITY	R1W0	奇校验错误标志清除控制 W0：清除奇校验错误标志 W1：无功能

## 8 FLASH存储器（FLASH）

### 8.1 FLASH 概述

HC32F0653 内部集成 64KB 嵌入式 FLASH 供用户使用，可用来存储应用程序和用户数据。芯片支持对 FLASH 存储器的读、擦除和写操作，支持擦写保护和读保护。

### 8.2 FLASH 主要特性

- 高达 64 KB 的 FLASH 单存储区域架构
- 存储器构成：2 个存储区域
  - 主存储器
  - 信息存储器
- 扇区大小：512 字节
- 数据读取宽度为 32 位
- 数据写入宽度为 8, 16, 32 位
- 扇区擦除 (512B) 和整片擦除

FLASH 接口特性：

- FLASH 读操作
- FLASH 编程/擦除操作
- FLASH 擦写保护
- 由选项字节 (RDP) 激活的读保护
- 选项字节加载器

## 8.3 FLASH 功能说明

### 8.3.1 FLASH 构成

HC32F0653内部集成 64KB 用户可访问的 FLASH 存储器，按每扇区 512 字节进行分扇区管理，共 128 扇区，用户可以对 FLASH 进行整扇区擦除和逐字节编程操作。

FLASH 构成如下表所示。

表 8-1 FLASH 模块构成

区域	地址	大小（字节）	名称
主存储器	0x0000 0000 - 0x0000 0FFF	512B	第 0 扇区
	0x0000 0000 - 0x0000 1FFF	512B	第 1 扇区
	0x0000 0000 - 0x0000 2FFF	512B	第 2 扇区
	...	...	...
	0x0000 0000 - 0x0000 FFFF	512B	第 127 扇区
信息块	0x1FFF F800 - 0x1FFF F9FF	512B	用户配置信息
	0x1FFF FA00 - 0x1FFF FBFF	512B	FT 测试信息
	0x1FFF FC00 - 0x1FFF FDFD	512B	CP 测试信息
	0x1FFF FE00 - 0x1FFF FFFF	512B	华虹测试用

### 8.3.2 FLASH 读访问延迟

为了准确读取 FLASH 数据，必须根据 FLASH (HCLK) 时钟频率在FLASH 访问控制寄存器 (FLASH\_ACR) 中正确地编程等待状态数 (LATENCY)。

表 8-2 FLASH 时钟 (HCLK) 频率对应的等待状态数

等待状态 (WS)(LATENCY)	频率范围
0 WS (1 个 HCLK 周期)	≤ 24
1 WS (2 个 HCLK 周期)	≤ 48
2 WS (3 个 HCLK 周期)	≤ 64

上电复位后，HCLK 时钟频率为 6 MHz（范围 1），FLASH\_ACR 寄存器中的等待状态(WS) 为 0。

### 8.3.3 FLASH 加速

#### 指令预取

每个 FLASH 读操作可读取 32 位，可以是 1 行 32 位指令，也可以是 2 行 16 位指令，具体取决于烧写在 FLASH 中的程序。这一 32 位当前指令行保存在当前缓冲区中。因此，对于顺序执行的代码，至少需要 1 个 CPU 周期来执行前一次读取的指令行。在 CPU 请求当前指令行时，可使用 CPU S 总线的预取操作读取 FLASH 中的下一个连续存放的指令行。

可将 FLASH 访问控制寄存器 (FLASH\_ACR) 中的 PRFTEN 位置 1，来使能预取功能。当访问 FLASH 至少需要一个等待状态时，此功能非常有用。

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

如果当前缓冲区中存在循环，则不会执行新的访问。

### 8.3.4 FLASH 编程和擦除操作

嵌入式 FLASH 可采用在线编程或在应用中编程两种方式。

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

与 ICP 方法相比,在**应用中编程 (IAP)** 可通过微控制器支持的任何通信接口(I/O、UART、I2C 和 SPI 等)将编程数据下载到存储器。IAP 允许用户在应用程序运行时重新编程 FLASH。但是,部分应用程序必须事先通过 ICP 方式编程到 FLASH。

如果在 FLASH 操作期间发生器件复位,无法保证 FLASH 中的内容。

在对 FLASH 执行编程/擦除操作期间,如果尝试读取 FLASH,则会使总线停止工作。在完成编程/擦除操作后,会正确执行读操作。

#### 解锁 FLASH

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

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

如果操作顺序不正确,会锁定 FLASH\_CR 寄存器,下次系统复位才**允许**解锁。

也可通过软件将这些寄存器中的一个 LOCK 位置 1 来锁定 FLASH\_CR 寄存器。

**注:** FLASH 状态寄存器 (FLASH\_SR) 中的 BSY 位置 1 时,FLASH\_CR 寄存器无法写入。BSY 位为 1 时,对该寄存器进行写操作会被忽略。

### 8.3.5 FLASH 主存储器擦除顺序

FLASH 擦除操作可在扇区级（扇区擦除）或在整个存储器（整片擦除）上执行。整片擦除不影响选项字节。

#### FLASH 扇区擦除

当某扇区受 WRP 保护时，该扇区不会被擦除且 WRPERR 位置 1。

扇区擦除的具体步骤如下：

1. 查询 FLASH 忙标志位 FLASH\_ISR.BUSY，如果 FLASH\_ISR.BUSY 为 0，则当前未执行任何 FLASH 操作，可以执行下一步骤；
2. 检查并清零之前的编程所导致的全部错误编程标志，否则，FLASH\_SR.PGSERR 将置 1。
3. 将 FLASH\_CR.SER 位置 1 并设置要擦除的地址 FLASH\_AR。
4. 将 FLASH\_CR.STRT 位置 1。
5. 等待 FLASH\_SR.BSY 位清零。

#### FLASH 整片擦除

当 WRP 使能时，FLASH 整片擦除中止，不会开始擦除，并且 FLASH\_ISR.WRPERR 位置 1。

整片擦除的具体步骤如下：

1. 查询 FLASH 忙标志位 FLASH\_ISR.BUSY，如果 FLASH\_ISR.BUSY 为 0，则当前未执行任何 FLASH 操作，可以执行下一步骤；
2. 检查并清零之前的编程所导致的全部错误编程标志，否则，FLASH\_SR.PGSERR 将置 1；
3. 将 FLASH 整片擦除使能位 FLASH\_ISR.MER 位置 1；
4. 将 FLASH 擦除启动位 FLASH\_ISR.STRT 位置 1，触发擦除操作；
5. 等待 FLASH 忙标志位 FLASH\_ISR.BSY 清零。

### 8.3.6 FLASH 主存储器编程顺序

#### 标准编程

标准模式下，FLASH 编程顺序如下：

1. 查询 FLASH 忙标志位 FLASH\_ISR.BUSY，如果 FLASH\_ISR.BUSY 为 0，则当前未执行任何 FLASH 操作，可以执行下一步骤；
2. 检查并清零之前的编程所导致的全部错误编程标志。否则，PGSERR 将置 1；
3. 将 FLASH 编程使能位 FLASH\_CR.PG 位置 1，使能 FLASH 编程功能；
4. 针对所需存储器地址执行数据写入操作；
  - a. 配置 FLASH\_CR.PSIZE 为 8bit 时按字节写入
  - b. 配置 FLASH\_CR.PSIZE 为 16bit 时按半字写入
  - c. 配置 FLASH\_CR.PSIZE 为 32bit 时按全字写入
5. 等待 FLASH 忙标志位 FLASH\_ISR.BSY 清零；
6. 查询等待 FLASH 操作结束位 FLASH\_ISR.EOP 标志置 1（编程操作已成功），并通过软件将其清零；
7. 如果不再有编程操作，则将 FLASH 编程控制位 FLASH\_CR.PG 位清零。

## 编程错误

可检测到多种错误。若发生错误，FLASH 操作（编程或擦除）会中止。

### ● FLASH\_ISR.PGSERR: 编程顺序错误

如果发生以下事件，FLASH\_ISR.PGSERR 位会置 1:

- 错误标志未清零，置位 FLASH\_CR.PG, FLASH\_ISR.MER 或 FLASH\_CR.SER 位
- FLASH\_CR.PG 清零时写入数据。
- FLASH\_CR.PG 置 1 后，FLASH\_CR.MER 和 FLASH\_CR.SER 不清零。
- 对于整片擦除顺序: FLASH\_CR.MER 置 1 后，FLASH\_CR.PG 和 FLASH\_CR.SER 不清零。
- 对于扇区擦除顺序: FLASH\_CR.SER 置 1 后，FLASH\_CR.PG 和 FLASH\_CR.MER 不清零。

### ● FLASH\_ISR.WRPERR: 写保护错误

如果发生以下事件，FLASH\_ISR.WRPERR 位会置 1:

- 在写保护区域 (WRP) 执行编程或擦除操作;
- 在有扇区受 WRP 保护时，执行整片擦除操作;
- 读保护(RDP)设为级别 1 时，连接调试功能进行擦写操作;
- 在读保护 (RDP) 设置为级别 2 时修改选项字节。

### ● SIZERR: (大小错误)

- 在编程过程中，写入数据和 FLASH\_CR.PSIZE 大小不一致时，SIZERR 置 1。

### ● PGAERR: (编程对齐错误)

- 在标准编程过程中: 要编程的第一个字与字地址不对齐。

如果在擦写操作期间出现错误，则 FLASH 状态寄存器 (FLASH\_SR) 中的以下错误标志之一将置 1:

- SIZERR: (大小错误)
- PGAERR: (编程对齐错误)
- PGSERR: (编程顺序错误)
- WRPERR: (写保护错误标志)

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

## 8.4FLASH 选项字节

### 8.4.1 FLASH 用户选项字说明

用户选项字由用户根据具体的应用要求进行配置。

用户选项字基地址: OPT\_BASE = 0x1FFF F800

表 8-3 选项字列表

地址	名称	说明
OPT_BASE + 0x00	FLASH_OPTR	用户选项字及补码
OPT_BASE + 0x04	FLASH_OPTWRPR	用户擦写保护选项字及补码

#### 8.4.1.1 FLASH\_OPTR用户选项字及补码

地址: 请参见表 8-3 选项字列表 出厂默认值: 0x0000 FFAA

位编号	位符号	权限	说明
31-16	COMPLEMENT	RW	低 16 位的补码
15	RFU	-	保留位，请保持默认值

14	RSTIO_EN	RW	RST 端口功能配置 0: 使能 GPIO 端口, RST 功能不可用 1: 使能 RST 端口, GPIO 功能不可用
13	IWDG_SW	RW	独立看门狗选择 0: 硬件独立看门狗 1: 软件独立看门狗
12-9	RFU	-	保留位, 请保持默认值
8	BOR_EN	RW	欠压复位使能 0: 禁止可配置欠压复位 1: 使能可配置欠压复位
7-0	RDP	RW	读保护级别 (Read protection level) <b>注:</b> 0xAA: 级别 0, 未激活读保护 0xCC: 级别 2, 激活芯片读保护 其他: 级别 1, 激活存储器读保护

#### 8.4.1.2 FLASH\_OPTWRPR 用户擦写保护选项字及补码

地址: 请参见表 8-3 选项字列表 出厂默认值: 0xFFFF 0000

位编号	位符号	权限	说明
31-16	COMPLEMENT	RW	低 16 位补码
15	WRP16	RW	扇区 120–扇区 127 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
14	WRP15	RW	扇区 112–扇区 119 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
13	WRP14	RW	扇区 104–扇区 111 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
12	WRP13	RW	扇区 96–扇区 103 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
11	WRP12	RW	扇区 88–扇区 95 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
10	WRP11	RW	扇区 80–扇区 87 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
9	WRP10	RW	扇区 72–扇区 79 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
8	WRP9	RW	扇区 64–扇区 71 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护



7	WRP8	RW	扇区 56–扇区 63 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
6	WRP7	RW	扇区 48–扇区 55 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
5	WRP6	RW	扇区 40–扇区 47 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
4	WRP5	RW	扇区 32–扇区 39 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
3	WRP3	RW	扇区 24 – 扇区 31 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
2	WRP2	RW	扇区 16 – 扇区 23 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
1	WRP1	RW	扇区 8 – 扇区 15 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
0	WRP0	RW	扇区 0 – 扇区 7 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护

## 8.4.2 FLASH 用户选项字编程

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

1. 使用 LOCK 清零顺序解锁 FLASH\_CR。
2. 在 FLASH 选项密钥寄存器 (FLASH\_OPTKEYR) 中写入 OPTKEY1=0x08192A3B。
3. 在 FLASH 选项密钥寄存器 (FLASH\_OPTKEYR) 中写入 OPTKEY2=0x4C5D6E7F。

如果操作或者密钥顺序不正确，会锁定 FLASH 选项寄存器，下次系统复位才会解锁。

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

注：如果 LOCK 由软件置 1，则 OPTLOCK 也自动置 1。

### 修改用户选项

选项字节与主存储器用户地址的编程方式不同。

要修改用户选项的值，请按照以下步骤操作：

1. 按照上述清零顺序将 OPTLOCK 选项锁定位清零；
2. 在 FLASH 选项寄存器中写入所需的值，除 RDP 和 IWDG\_SW 外，其他用户选项立即生效，但不会更新到 FLASH；
3. 检查 FLASH 状态寄存器 (FLASH\_SR) 中的 BSY 位，以确认当前未执行任何 FLASH 操作。
4. 将 FLASH 控制寄存器 (FLASH\_CR) 中的选项启动位 OPTSTRT 置 1，此时所有新选项都将更



新到 FLASH。

5. 等待 BSY 位清零，此时修改后的 RDP 生效。
  6. 如果希望 IWDG\_SW 生效，复位 MCU，等待复位结束，此时修改后的 IWDG\_SW 生效。OPTSTRT 位置 1 后，会自动计算补码值并写入补码选项字节。
- 注：降级 RDP 选项字节为禁止所有级别保护 (0xAA) 时，会自动整片擦除 FLASH 主存储器。**

### 用户选项字节加载

用户选项字节擦写完成，并且 BSY 位清零后，除了 IWDG\_SW 其他所有新选项都将更新到 FLASH，应用到系统，IWDG\_SW 位只有在任意复位结束后才会执行选项字加载并应用到系统。

用户选项加载期间，芯片会自动验证用户选项位及其补码可检查是否已正确进行加载。

如果字与其补码匹配，则选项字将复制到选项寄存器中。

如果字与其补码失配，则 OPTVERR 状态位置 1。失配的结果值被强制放入选项寄存器：

- 对于 FLASH\_OPT 选项，所有选项位的失配结果值均为 1，但 BORR\_LEV 和 BORF\_LEV 位域的位除外，这些位的失配结果值为 00（最低阈值），BOR\_EN 位也除外，该位的失配结果值为 0（禁止 BOR）。
- 对于 FLASH\_WRP 选项，失配结果值为默认值“保护”。
- 对于 FLASH\_RDP 选项，失配结果值为默认值“级别 1”。

系统复位后，选项字将加载到 FLASH 选项寄存器 (FLASH\_OPTR) 中。

## 8.5 FLASH 保护

可对主 FLASH 进行保护，使其具有读保护 (RDP)，以防其遭受外部访问。连续多扇区 FLASH 还具备防止因程序指针错乱而发生意外的写操作 (WRP) 保护功能。

### 8.5.1 FLASH 读保护 (RDP)

修改用户选项区域的 RDP 后应用系统复位来重载新的 RDP 选项字节，可激活读保护。读保护功能可保护主 FLASH。

**注：有三种读保护级别：无保护（级别 0）到最大保护或禁止调试（级别 2）。**

RDP 选项字节及其补码包含成对值时，FLASH 受保护，如下表所示。

**表 8-4 FLASH 读保护状态**

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

#### 级别 0：无保护

可对主 FLASH 区域执行读取、编程和擦除操作。也可对选项字进行所有操作。

#### 级别 1：读保护

当 RDP 字节和 RDP 补码字节包含 [0xAA, 0x55] 和 [0xCC, 0x33] 以外的任何值组合时，设置级别 1 读保护。级别 1 是擦除 RDP 选项字节时的默认保护级别。

- 用户模式：在用户模式下执行的代码可对主 FLASH、选项字节执行所有操作。
- 调试模式：在调试模式下，主 FLASH 完全不可访问，对 FLASH 进行读访问会返回全 1，擦写操作会生成写保护错误。

#### 级别 2：禁止调试/芯片读保护

在此级别下，可保持保护级别 1 的模式。此外，不再支持 CPU 调试端口。在用户执行模式下，允许对主 FLASH 执行所有操作。

**注：**复位时会禁止 CPU 调试端口。

**注：**普通用户无法对设为保护级别 2 的器件做失效分析。

#### 更改读保护级别

可以更改读保护级别：

- 通过将 RDP 字节的值更改为 0xCC 以外的任意值，从级别 0 更改为级别 1
- 通过将 RDP 字节的值更改为 0xCC，从级别 0 或级别 1 更改为级别 2
- 通过将 RDP 字节的值更改为 0xAA，从级别 1 更改为级别 0

更改为级别 2 后，普通用户无法再更改读保护级别。

表 8-5 访问状态 vs 保护级别和执行模式

区域	保护级别	用户模式			调试模式		
		读	写	擦除	读	写	擦除
主 FLASH	1	Y	Y	Y	N	N	N
	2	Y	Y	Y	N	N	N
选项字节	1	Y	Y	Y	Y	Y	Y
	2	Y	N	N	N	N	N

1. 保护级别 2 激活后，将禁止调试端口。
2. 修改 RDP 选项字节为禁止所有级别保护 (0xAA) 时，会无视掉 WRP，自动整片擦除 FLASH 主存储器

## 8.5.2 FLASH 写保护 (WRP)

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

如果尝试对 FLASH 的写保护区域执行擦除/编程操作，则 FLASH\_SR 寄存器中的写保护错误标志 (WRPERR) 将置 1。

**注：**选择 FLASH 读保护级别 (RDP 级别 = 1) 后，如果已连接 CPU 调试功能，则即使 WRP 未激活，也无法对主存储器执行编程或擦除操作。

## 8.6 FLASH 中断

表 8-6 FLASH 中断请求

中断事件	事件标志	事件标志/中断清除方法	中断使能控制位
操作结束	EOP <sup>(1)</sup>	写入 EOP=1	EOPIE
操作错误	OPERR <sup>(2)</sup>	写入 OPERR=1	ERRIE
写保护错误	WRPERR	写入 WRPERR=1	N/A
大小错误	SIZERR	写入 SIZERR=1	N/A
编程顺序错误	PGSERR	写入 PGSERR=1	N/A

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

## 8.7 FLASH 寄存器列表

FLASH 基地址: FLASH\_BASE = 0x4000 0000

表 8-7 FLASH 寄存器列表

寄存器名称	寄存器地址	寄存器描述
FLASH_ACR	FLASH_BASE + 0x00	FLASH 访问控制寄存器
FLASH_KEYR	FLASH_BASE + 0x04	FLASH 密钥寄存器
FLASH_OPTKEYR	FLASH_BASE + 0x08	FLASH 选项密钥寄存器
FLASH_ISR	FLASH_BASE + 0x0C	FLASH 状态寄存器
FLASH_CR	FLASH_BASE + 0x10	FLASH 控制寄存器
FLASH_OPTCR	FLASH_BASE + 0x14	FLASH 选项控制寄存器
FLASH_AR	FLASH_BASE + 0x18	FLASH 地址寄存器
FLASH_IER	FLASH_BASE + 0x1C	FLASH 中断使能寄存器

## 8.8 FLASH 寄存器描述

### 8.8.1 FLASH 访问控制寄存器 (FLASH\_ACR)

地址：请参见表 8-7 FLASH 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-9	RFU	-	保留位，请保持默认值
8	FETCH	RW	CPU 预取使能 0：禁止 CPU 预取 1：使能 CPU 预取
7-2	RFU	-	保留位，请保持默认值
1-0	WAIT	RW	FLASH 取指周期配置 00：1 个 HCLK 周期，适用于 HCLK ≤ 24MHz 01：2 个 HCLK 周期，适用于 24MHz < HCLK ≤ 48MHz 10：3 个 HCLK 周期，适用于 48MHz < HCLK ≤ 60MHz

### 8.8.2 FLASH 密钥寄存器 (FLASH\_KEYR)

地址：请参见表 8-7 FLASH 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-0	KEY	W	FLASH 密钥 要将 FLASH 控制寄存器 (FLASH_CR) 解锁以允许执行编程 / 擦除操作，必须顺序写入以下值： KEY1：0x4567 0123 KEY2：0xCDEF 89AB

### 8.8.3 FLASH 选项密钥寄存器 (FLASH\_OPTKEYR)

地址：请参见表 8-7 FLASH 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-0	OPTKEYR	W	选项字节密钥 要将 FLASH 选项寄存器解锁以允许执行选项字节编程/擦除操作，必须顺序写入以下值： KEY1：0x0819 2A3B KEY2：0x4C5D 6E7F

### 8.8.4 FLASH 中断状态寄存器 (FLASH\_ISR)

地址：请参见表 8-7 FLASH 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31	DEBUG	RO	该标志指示芯片是否处于调试状态下
30	OPT_CFG	RO	该标志指示芯片配置字是否生效
29-17	RFU	-	保留位，请保持默认值

16	BUSY	RO	繁忙 该标志指示 FLASH 控制寄存器 (FLASH_CR) 请求的 FLASH 操作正在进行。该位在 FLASH 操作开始时置 1，在操作结束或出现错误时复位。
15	OPTVERR	RO	用户选项字加载有效性错误 读取的用户选项字可能不是由用户或生产配置时，由硬件置 1。如果未正确加载选项位，则 OPTVERR 在每次系统复位后再次置 1。加载失败的用户选项将被强制为安全值。 客户上电后可以读取该位，判断是否加载成功，不成功，可以执行系统复位，重新加载。
14	FTVERR	RO	FT 测试信息加载有效性错误 读取的测试信息可能不是由用户或生产配置时，由硬件置 1。如果未正确加载校准字，则 FTVERR 在每次系统复位后再次置 1。加载失败的测试信息将被强制为安全值。 客户上电后可以读取该位，判断是否加载成功，不成功，可以执行系统复位，重新加载。
13	FLASHVERR	RO	FLASH 校准字加载有效性错误 读取的 FLASH 校准字可能不是由用户或生产配置时，由硬件置 1。如果未正确加载校准字，则 FLASHVERR 在每次系统复位后再次置 1。加载失败的校准字将被强制为安全值。 客户上电后可以读取该位，判断是否加载成功，不成功，可以执行系统复位，重新加载。
12-8	RFU	-	保留位，请保持默认值
7	PGSERR	RW1	编程顺序错误 如果代码在 PG 前未置 1 的情况下对 FLASH 执行写访问，则该位由硬件置 1。当先前的编程错误导致 SIZERR、PGAERR、WRPERR 置 1 时，该位也会由硬件置 1。 写入 1 即可将该位清零。
6	SIZERR	RW1	大小错误 如果在编程期间数据访问类型（字节、半字、字）与配置的并行位数 PSIZE (x8, x16, x32) 不符，将由硬件为该位置 1。 写入 1 即可将该位清零。
5	PGAERR	RW1	编程对齐错误 如果在标准编程期间要编程的数据无法包含在同一字（32 位）FLASH 中时该位由硬件置 1。 写入 1 即可将该位清零。
4	WRPERR	RW1	写保护错误 如果要擦除/编程的地址属于 FLASH 中受写保护（受 WRP 或 RDP 级别 1 保护）的区域，则该位由硬件置 1。 写入 1 即可将该位清零。
3-2	RFU	-	保留位，请保持默认值
1	OPERR	RW1	操作错误 当 FLASH 操作（编程/擦除）失败时，该位由硬件置 1。

			只有在使能错误中断 (ERRIE = 1) 后, 该位才会置 1。 写入“1”即可将该位清零。
0	EOP	RW1	操作结束 当成功完成一个或多个 FLASH 操作 (编程/擦除) 时, 该位由硬件置 1。 只有在使能操作结束中断 (EOPIE = 1) 后, 该位才会置 1。 写入 1 即可将该位清零。

## 8.8.5 FLASH 控制寄存器 (FLASH\_CR)

地址: 请参见表 8-7 FLASH 寄存器列表 复位值: 0x8000 0000

访问: 当前未执行任何 FLASH 操作时无等待状态, 按字、半字和字节访问

位编号	位符号	权限	说明
31	LOCK	RW1	FLASH_CR 锁定 该位只置 1。置 1 后, FLASH_CR 寄存器被锁定。当检测到解锁序列时, 由硬件将该位清零。 如果解锁操作失败, 该位仍保持置 1, 直到下一次系统复位。
30	OPTLOCK	RW1	锁定选项 该位只能写入 1。该位置 1 时, 表示 FLASH_OPTCR 寄存器已锁定。 当检测到解锁序列时, 由硬件将该位清零。 如果解锁操作失败, 该位仍保持置 1, 直到下一次复位。
29-18	RFU	-	保留位, 请保持默认值
17	OPTSTART	RW1	开始修改用户选项字 该位置 1 后可触发选项操作。 该位只能通过软件置 1, 并在 FLASH_SR 中的 BSY 位清零后随之清零。
16	STRT	RW1	启动 该位置 1 后可触发擦除操作。如果 MER 和 PGSER 位均复位并且 STRT 位置 1, 可能会发生无法预知的行为而不会生成任何错误标志。 应禁止此情况发生。 该位只能通过软件置 1, 并在 FLASH_SR 中的 BSY 位清零后硬件清零。
15-13	RFU	-	保留位, 请保持默认值
12-11	PSIZE	RW	PSIZE: 编程大小 这些位用于选择编程位数。 00: 字节 (8bit) 01: 半节 (16bit) 10: 全节 (32bit) 11: 保留位
10-3	RFU	-	保留位, 请保持默认值
2	MER	RW1	整片擦除 该位置 1 后可触发整片擦除 (所有用户扇区)。

1	SER	RW	扇区擦除 0: 禁止扇区擦除 1: 使能扇区擦除
0	PG	RW	编程 0: 禁止 FLASH 编程 1: 使能 FLASH 编程

### 8.8.6 FLASH 用户选项控制寄存器 (FLASH\_OPTCR)

地址: 请参见表 8-7 FLASH 寄存器列表 复位值: 0xXXXX XXXX

复位信号释放时, 硬件将 FLASH 中的值载入这些选项位。

注: 修改 FLASH\_OPTCR 后, WRP, BOR 位立即生效。

位编号	位符号	权限	说明
31	WRP16	RW	扇区 120–扇区 127 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
30	WRP15	RW	扇区 112–扇区 119 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
29	WRP14	RW	扇区 104–扇区 111 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
28	WRP13	RW	扇区 96–扇区 103 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
27	WRP12	RW	扇区 88–扇区 95 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
26	WRP11	RW	扇区 80–扇区 87 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
25	WRP10	RW	扇区 72–扇区 79 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
24	WRP9	RW	扇区 64–扇区 71 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
23	WRP8	RW	扇区 56–扇区 63 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
22	WRP7	RW	扇区 48–扇区 55 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护



21	WRP6	RW	扇区 40–扇区 47 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
20	WRP5	RW	扇区 32–扇区 39 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
19	WRP3	RW	扇区 24 – 扇区 31 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
18	WRP2	RW	扇区 16 – 扇区 23 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
17	WRP1	RW	扇区 8 – 扇区 15 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
16	WRP0	RW	扇区 0 – 扇区 7 写保护配置 0: 开启所选扇区的写保护 1: 关闭所选扇区的写保护
15	RFU	-	保留位, 请保持默认值
14	RSTIO_EN	RW	RST 端口功能配置 0: 使能 GPIO 端口, RST 功能不可用 1: 使能 RST 端口, GPIO 功能不可用
13	IDWG_SW	RW	独立看门狗选择 0: 硬件独立看门狗 1: 软件独立看门狗
12-9	RFU	-	保留位, 请保持默认值
8	BOR_EN	RW	欠压复位使能 0: 禁止可配置欠压复位 1: 使能可配置欠压复位
7-0	RDP	RW	读保护级别 (Read protection level) <b>注:</b> 0xAA: 级别 0, 未激活读保护 0xCC: 级别 2, 激活芯片读保护 其他: 级别 1, 激活存储器读保护

### 8.8.7 FLASH 地址寄存器 (FLASH\_AR)

地址: 请参见表 8-7 FLASH 寄存器列表 复位值: 0x0000 0000

位编号	位符号	权限	说明
31-0	ADDR	WO	FLASH 扇区擦除地址 当进行页擦除时选择要擦除的页。 注意: 当 FLASH_SR 中的 BSY 位为 1 时, 不能写这个寄存器。



### 8.8.8 FLASH 中断使能寄存器 (FLASH\_IER)

地址：请参见表 8-7 FLASH 寄存器列表      复位值： 0x0000 0000

位编号	位符号	权限	说明
31-3	RFU	-	保留位，请保持默认值
1	ERRIE	RW	错误中断使能 当 FLASH_SR 寄存器中的 OPERR 位置 1 后，可通过该位使能中断产生功能。 0：禁止 OPERR 错误中断 1：使能 OPERR 错误中断
0	EOPIE	RW	操作结束中断使能 当 FLASH_SR 寄存器中的 EOP 位置 1 后，可通过该位使能中断产生功能。 0：禁止 EOP 中断 1：使能 EOP 中断

## 9 通用输入输出端口（GPIO）

### 9.1 概述

GPIO 控制器实现芯片内部各类数字和模拟电路与物理引脚之间的联系。

GPIO 可配置为数字输入输出和模拟功能，支持外设功能复用，支持高电平、低电平、上升沿和下降沿 4 种中断源，可在深度休眠模式下通过外部中断唤醒 MCU 回到运行模式。

### 9.2 主要特性

- 所有寄存器通过 AHB 总线接口读写
- 具有数字输入输出和模拟功能
- 数字输入输出支持普通 GPIO 和全映射功能复用
- 模拟功能可作为 ADC、VC、FDA 的输入信号和 FDA 的输出信号
- 支持内部多种时钟信号输出
- 数字输入支持内部上拉、下拉和高阻三种模式
- 数字输出支持推挽和开漏模式
- GPIO 配置支持锁定功能
- 数字输出支持位置位，位清零，位翻转的原子位操作
- 中断功能支持高电平、低电平、上升沿、下降沿触发方式
- GPIO 具有数字滤波功能，可选择多种时钟源
- 支持在深度休眠模式下通过外部中断唤醒 MCU

## 9.3 功能描述

### 9.3.1 功能框图

GPIO 控制器的功能框图如下图所示：

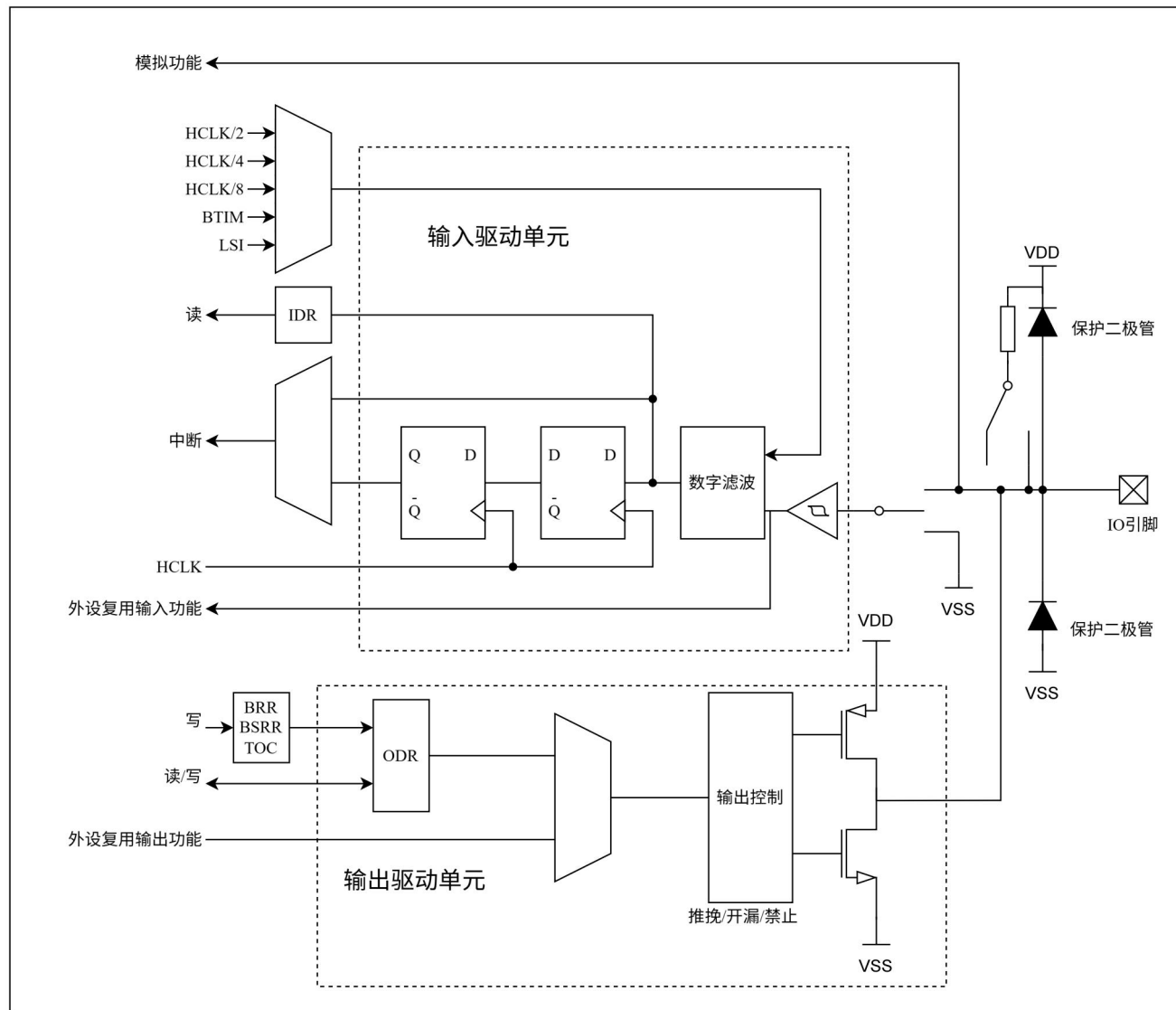


图 9-1 GPIO 功能框图

## 9.3.2 数字输出

将模拟数字配置寄存器 `GPIOx_ANALOG[y]` ( $y$  是引脚号,  $y=0\sim15$ ; 下同) 清零, 配置相应的 GPIO 端口为数字功能; 将输入输出方向寄存器 `GPIOx_DIR[y]` 清零, 配置 GPIOx 端口为输出模式。

数字输出信号来源可以是:

- 输出数据寄存器 `GPIOx_ODR`
- 片内数字外设

通过输出模式寄存器 `GPIOx_OPENDRAIN` 配置输出模式, 可选择推挽输出或开漏输出。

## 9.3.3 数字输入

将模拟数字配置寄存器 `GPIOx_ANALOG[y]` 清零, 配置 GPIO 端口为数字功能; 将输入输出方向寄存器 `GPIOx_DIR[y]` 置位, 配置相应的 GPIO 端口为输入模式。数字输入信号可配置:

到达输入数据寄存器 `GPIOx_IDR`

- 到达片内数字外设
- 触发中断

在该模式下, 数字输入信号通过 ADS 开关导入内部数字输入电路。

经施密特触发器确认电平状态后, 可以直接被送往片内复用功能所指向的数字外设的输入, 或者通过一个基于 HCLK 的同步器后, 在输入数据寄存器 `GPIOx_IDR[y]` 上呈现。

`GPIOx_IDR` 寄存器的各位与其前面的锁存器组成了一个同步器, 可以避免系统时钟变化的时间内引脚电平跳化而造成的信号不稳定, 但是会产生一定的读取延迟。读端口引脚的同步时序如下图所示:

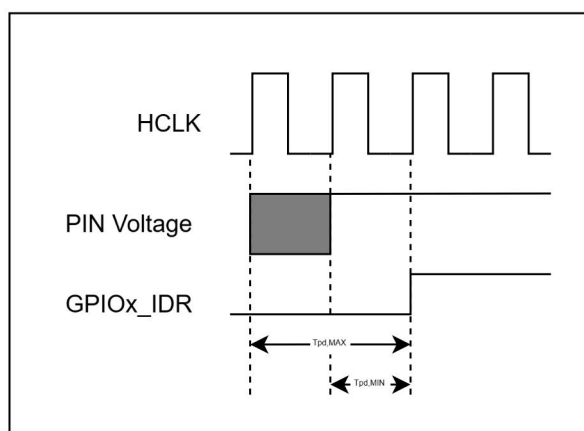


图 9-2 读端口同步时序

在系统时钟上升沿之后的时钟周期, 引脚电平信号会锁存在内部寄存器, 如图中阴影部分所示, 在下次系统时钟上升沿之后, 稳定的引脚电平信号被读取, 再一个系统时钟上升沿时, 数据被锁存到 `GPIO_IDR` 寄存器中。信号延迟  $T_{pd}$  为 1~2 个系统时钟。

如果考虑将该输入信号用于触发中断, 还可以启用内置的硬件滤波器电路。该滤波器电路是基于双 D 触发器同步器实现的, 该同步器的时钟来源有 7 种, 其中部分时钟源是低功耗模式特有的。例如, 可以轻易的实现无软件干预的按键消抖操作。

在该模式下, 通过上拉电阻寄存器 `GPIOx_PUR` 和下拉电阻寄存器 `GPIOx_PDR` 可以单独选择打开或者关闭内部上拉和下拉功能。

### 9.3.4 模拟功能

对于配置有模拟功能的 GPIO 端口，可通过设置模拟数字配置寄存器 GPIOx\_ANALOG[y] 为 1，打开 GPIO 模拟信号通道。在打开 GPIO 模拟信号通道时，端口的数字功能关闭，内部上拉、下拉均被断开，内部数字输入信号通过 ADS 开关被短接到 VSS，内部数字输出功能被禁止。

### 9.3.5 复用功能

HC32F0653 内置外设功能引脚全映射模块，可通过用户软件操作将绝大多数的外设功能引脚配置在任意一个非电源口（VDD、GND）上。

#### 9.3.5.1 复用功能模块特性

- 外设引脚为输入功能时，系统将允许其多对一映射，即将多种输入外设功能引脚分配到同一 IO 口上，此举可使用户系统得到更好优化。
- 外设引脚为输出功能时，如果将多个输出外设功能引脚分配到同一 IO 口，遵循固定的优先级，只能有一个输出有效。
- 软件操作，使用灵活。用户在应用系统设计时，不用考虑外设功能引脚布局问题，从而降低开发成本。
- 用户在开发过程中遇到如 PCB 上外设功能引脚布局错误情况时，可使用该模块对外设功能引脚进行重新分配，从而缩短开发周期。
- 用户应用系统升级如更换外围器件或 MCU 时，改动项可减至最少，从而降低系统维护成本。

### 9.3.5.2 可全映射外设功能引脚

外设	名称	类型	说明
SYSCTRL	MCO0	O	时钟输出口 0
	MCO1	O	时钟输出口 1
ATIM	ATIM_CH0A	I/O	ATIM 通道
	ATIM_CH0B	O	ATIM 通道
	ATIM_CH1A	I/O	ATIM 通道
	ATIM_CH1B	O	ATIM 通道
	ATIM_CH2A	I/O	ATIM 通道
	ATIM_CH2B	O	ATIM 通道
	ATIM_CH3A	I/O	ATIM 通道
	ATIM_CH3B	O	ATIM 通道
	ATIM_CH4A	I/O	ATIM 通道
	ATIM_CH4B	O	ATIM 通道
	ATIM_CH5A	I/O	ATIM 通道
	ATIM_CH5B	O	ATIM 通道
	ATIM_ETR	I	ATIM 外部触发输入
	ATIM_BKIN0	I	ATIM 刹车输入 0
	ATIM_BKIN1	I	ATIM 刹车输入 1
GTIM	GTIM_CH0	O	GTIM 通道
	GTIM_CH1	O	GTIM 通道
	GTIM_CH2	O	GTIM 通道
	GTIM_CH3	O	GTIM 通道
	GTIM_ETR	I	GTIM 外部触发输入
BTIM0	BTIM0_ETR	I	BTIM0 外部触发输入
	BTIM0_TOGP	O	BTIM0 翻转通道
	BTIM0_TOGN	O	BTIM0 翻转通道
BTIM1	BTIM1_ETR	I	BTIM1 外部触发输入
	BTIM1_TOGP	O	BTIM1 翻转通道
	BTIM1_TOGN	O	BTIM1 翻转通道
BTIM2	BTIM2_ETR	I	BTIM2 外部触发输入
	BTIM2_TOGP	O	BTIM2 翻转通道
	BTIM2_TOGN	O	BTIM2 翻转通道
UART	TXD0	O	UART0 数据传输口
	RXD0	I	UART0 接收脚
	CK0	O	UART0 时钟输出
	TXD1	O	UART1 数据传输口
	RXD1	I	UART1 接收脚
	CK1	O	UART1 时钟输出
SPI	MOSI	I/O	SPI 的数据口，主机的输出和从机的输入

	MISO	I/O	SPI 的数据口，主机的输入和从机的输出
	SCK	I/O	SPI 的时钟口
	CS	I/O	SPI 的片选口
I2C	SCL	I/O	I2C 时钟口
	SDA	I/O	I2C 数据口
VC	OUT	O	VC0 输出脚
	OUT	O	VC1 输出脚
ADC	START	O	ADC 转换信号输出脚
	SQR0TRGO	I	ADC 序列 0 GPIO 触发信号输入
	SQR1TRGO	I	ADC 序列 1 GPIO 触发信号输入
	SQR2TRGO	I	ADC 序列 2 GPIO 触发信号输入
	SQR3TRGO	I	ADC 序列 3 GPIO 触发信号输入

### 9.3.5.3 不可全映射外设功能引脚

不可全映射外设功能引脚包括电源口（VDD、GND）、ADC 输入、Vref 引脚、VC、FDA、外部复位端口（RST）这些功能口是固定的，不能任意映射。

### 9.3.6 中断功能

每个 GPIO 在设置为数字输入模式时，可作为外部中断信号源，产生中断的信号源可以设置为高电平、低电平、上升沿、下降沿 4 种。中断触发方式可组合使用，但共用同一个中断标志位。

中断触发后，中断标志寄存器 GPIOx\_ISR 的对应位会被硬件置位，程序可通过查询 GPIO\_ISR 来确认产生中断的端口。通过中断标志清除寄存器 GPIOx\_ICR[y]，可以清除对应的中断标志位。

内部的数字滤波器可对引脚上的输入信号进行数字滤波，提供了 5 种滤波时钟选择，如下表所示：

GPIO_FILTER.FLTCLK	数字滤波时钟频率
000	HCLK / 2
001	HCLK / 4
010	HCLK / 8
011	BTIM 溢出
101	LSI (约 32.8 kHz)

由于选择的滤波时钟周期范围宽广，用户可以轻易实现灵活的输入中断防抖功能。输入电平的变化如果未保持超过一个完整的滤波时钟周期，将不会通过硬件滤波器传达到内部中断触发电路。输入电平的变化如果保持超过两个完整的滤波时钟周期，则一定会通过硬件滤波器。

对于边沿触发类型，考虑到对触发沿的时间的敏感性，建议在GPIO数字滤波器配置寄存器 GPIOx\_FILTER[y] 中关闭硬件滤波器功能，因为硬件滤波器在提升信号稳定性的同时，也会插入一定延迟。

当 HC32F0653 工作于休眠模式 (Sleep mode) 或深度休眠模式 (DeepSleep mode) 时，仍可使用 GPIO 的外部中断功能，当产生外部中断后，可将芯片从休眠模式或深度休眠模式唤醒回到运行模式。

## 9.3.7 其他功能

### 1. 原子位操作

GPIOx 控制器支持位置位、位清零和位翻转功能。

向 GPIOx 位置位清零寄存器 GPIOx\_BSRR[y] 或位清零寄存器 GPIOx\_BRR[y] 写入 1，将直接改变输出数据寄存器 GPIOx\_ODR 的对应位的状态，从而间接影响最终的输出电平，但不会影响该寄存器其它位的状态。

向 GPIOx 位翻转寄存器 GPIOx\_TOG[y] 写入 1，将使输出端口的电平状态发生翻转。

### 2. 端口配置锁定

当配置锁定寄存器 GPIOx\_LCKR 的对应位被设置为 1 后，配置寄存器的相应比特不可修改，包括如下配置寄存器：GPIOx\_ANALOG、GPIOx\_DIR、GPIOx\_OPENDRAIN、GPIOx\_PUR、GPIOx\_PDR、GPIOx\_DRIVER、GPIOx\_RISEIE、GPIOx\_FALLIE、GPIOx\_HIGHIE、GPIOx\_LOWIE。

用户可在 GPIOx 初始化完成后，对重要端口的配置锁定寄存器相应位进行锁定，防止程序跑飞而对端口的异常操作。

例：向 GPIOA\_LCKR 写入 0x5A5A0201，解锁 GPIOA 除 PA09、PA00 之外的端口相关配置寄存器，同时，锁定 PA09 和 PA00 端口相关配置寄存器。

### 3. 端口复位状态

上电或复位后，SWCLK，SWDIO 默认为数字输入，RST 默认为数字上拉输入。其他端口默认为模拟高阻输入（high resistance input），上拉或下拉均默认不打开。



## 9.4 编程示例

在配置 GPIOx 端口时，必须先设置 SYSCTRL\_AHBEN.GPIOx 为 1，使能对应的 GPIOx 配置时钟及工作时钟，并向 GPIOx\_LCKR 锁定寄存器写入‘0x5A5Aiiii’，以解锁 GPIO 相关配置寄存器，配置完成后，如有必要，可设置 GPIOx\_LCKR 锁定寄存器，以保护设置内容不会被意外改写。

### 9.4.1 数字输出编程示例

1. 设置 GPIOx\_ANALOG.PINy 为 0，将端口配置为数字功能；
2. 设置 GPIOx\_DIR.PINy 为 0，将端口配置成输出；
3. 配置 GPIOx\_OPENDRAIN 寄存器，设置端口输出模式；
4. 配置 GPIOx\_ODR 寄存器，设置端口输出电平。

### 9.4.2 数字输入编程示例

1. 设置 GPIOx\_ANALOG.PINy 为 0，将端口配置为数字功能；
2. 配置 GPIOx\_FILTCLR.CLK，选择端口滤波时钟源和时钟个数；
3. 设置 GPIOx\_FILTER.PINy 为 1，使能相应端口滤波时钟；
4. 设置 GPIOx\_DIR.PINy 为 1，将端口配置成输入；
5. 配置 GPIOx\_PUR 寄存器，选择是否使能内部上拉电阻；
6. 配置 GPIOx\_PDR 寄存器，选择是否使能内部下拉电阻；
7. 读取 GPIOx\_IDR 寄存器，读出端口输入电平。

### 9.4.3 模拟功能编程示例

1. 设置 GPIOx\_ANALOG.PINy 为 1，将端口配置为模拟功能。

### 9.4.4 复用功能编程实例

1. 根据应用需求将端口配置成数字输出或数字输入；
2. 配置 GPIOx\_XXX\_MAP 寄存器，设置端口复用功能。

### 9.4.5 中断功能编程示例

1. 将端口配置成数字输入；
2. 配置 GPIOx\_FILTCLR.CLK，选择端口滤波时钟源和时钟个数；
3. 设置 GPIOx\_FILTER.PINy 为 1，使能相应端口滤波功能；
4. 配置 NVIC 控制器；
5. 根据应用需求，配置 GPIOx\_RISEIE、GPIOx\_FALLIE、GPIOx\_HIGHIE、GPIOx\_LOWIE 寄存器，选择 GPIOx 中断触发方式；
6. 端口中断输入信号触发 GPIOx 中断，执行中断服务函数

## 9.5 寄存器列表

GPIOA 基地址: GPIOA\_BASE = 0x4000 2000

GPIOB 基地址: GPIOA\_BASE = 0x4000 2400

GPIO\_XXX\_MAP 基地址: GPIO\_MAP\_BASE = 0x4000 2C00

表 9-1GPIO 寄存器列表

寄存器名称	寄存器地址	寄存器描述
GPIOx_DIR	GPIOx_BASE+0x00	GPIO 输入输出方向寄存器
GPIOx_OPENDRAIN	GPIOx_BASE+0x04	GPIO 输出模式寄存器
GPIOx_PDR	GPIOx_BASE+0x0C	GPIO 下拉电阻寄存器
GPIOx_PUR	GPIOx_BASE+0x10	GPIO 上拉电阻寄存器
GPIOx_ANALOG	GPIOx_BASE+0x1C	GPIO 模拟数字配置寄存器
GPIOx_RISEIE	GPIOx_BASE+0x24	GPIO 上升沿中断使能寄存器
GPIOx_FALLIE	GPIOx_BASE+0x28	GPIO 下降沿中断使能寄存器
GPIOx_HIGHIE	GPIOx_BASE+0x2C	GPIO 高电平中断使能寄存器
GPIOx_LOWIE	GPIOx_BASE+0x30	GPIO 低电平中断使能寄存器
GPIOx_ISR	GPIOx_BASE+0x34	GPIO 中断标志寄存器
GPIOx_ICR	GPIOx_BASE+0x38	GPIO 中断标志清除寄存器
GPIOx_LCKR	GPIOx_BASE+0x3C	GPIO 端口配置锁定键寄存器
GPIOx_FILTER	GPIOx_BASE+0x44	GPIO 中断数字滤波器配置寄存器
GPIOx_FILTCR	GPIOx_BASE+0x48	GPIO 数字滤波器时钟配置寄存器
GPIOx_IDR	GPIOx_BASE+0x4C	GPIO 输入数据寄存器
GPIOx_ODR	GPIOx_BASE+0x50	GPIO 输出数据寄存器
GPIOx_BRR	GPIOx_BASE+0x54	GPIO 位清零寄存器
GPIOx_BSR	GPIOx_BASE+0x58	GPIO 位置位寄存器
GPIOx_TOG	GPIOx_BASE+0x5C	GPIO 位翻转寄存器

表 9-2GPIO 映射寄存器列表

寄存器名称	寄存器地址	寄存器描述
GPIO_MCO0_MAP	GPIO_MAP_BASE+0x00	时钟输出口 0
GPIO_MCO1_MAP	GPIO_MAP_BASE+0x04	时钟输出口 1
GPIO_ATIM_CH0A_MAP	GPIO_MAP_BASE+0x08	ATIM_CH0A 通道
GPIO_ATIM_CH0B_MAP	GPIO_MAP_BASE+0x0C	ATIM_CH0B 通道
GPIO_ATIM_CH1A_MAP	GPIO_MAP_BASE+0x10	ATIM_CH1A 通道
GPIO_ATIM_CH1B_MAP	GPIO_MAP_BASE+0x14	ATIM_CH1B 通道
GPIO_ATIM_CH2A_MAP	GPIO_MAP_BASE+0x18	ATIM_CH2A 通道
GPIO_ATIM_CH2B_MAP	GPIO_MAP_BASE+0x1C	ATIM_CH2B 通道
GPIO_ATIM_CH3A_MAP	GPIO_MAP_BASE+0x20	ATIM_CH3A 通道
GPIO_ATIM_CH3B_MAP	GPIO_MAP_BASE+0x24	ATIM_CH3B 通道
GPIO_ATIM_CH4A_MAP	GPIO_MAP_BASE+0x28	ATIM_CH4A 通道

GPIO_ATIM_CH4B_MAP	GPIO_MAP_BASE+0x2C	ATIM_CH4B 通道
GPIO_ATIM_CH5A_MAP	GPIO_MAP_BASE+0x30	ATIM_CH5A 通道
GPIO_ATIM_CH5B_MAP	GPIO_MAP_BASE+0x34	ATIM_CH5B 通道
GPIO_ATIM_ETR_MAP	GPIO_MAP_BASE+0x38	ATIM 外部触发输入
GPIO_ATIM_BKIN0_MAP	GPIO_MAP_BASE+0x3C	ATIM 刹车 0 输入
GPIO_ATIM_BKIN1_MAP	GPIO_MAP_BASE+0x40	ATIM 刹车 1 输入
GPIO_GTIM_CH0_MAP	GPIO_MAP_BASE+0x44	GTIM_CH0 通道
GPIO_GTIM_CH1_MAP	GPIO_MAP_BASE+0x48	GTIM_CH1 通道
GPIO_GTIM_CH2_MAP	GPIO_MAP_BASE+0x4C	GTIM_CH2 通道
GPIO_GTIM_CH3_MAP	GPIO_MAP_BASE+0x50	GTIM_CH3 通道
GPIO_GTIM_ETR_MAP	GPIO_MAP_BASE+0x54	GTIM_ETR 外部触发输入
GPIO_BTIM0_TOGP_MAP	GPIO_MAP_BASE+0x58	BTIM0_TOGP 输出
GPIO_BTIM0_TOGN_MAP	GPIO_MAP_BASE+0x5C	BTIM0_TOGN 输出
GPIO_BTIM0_ETR_MAP	GPIO_MAP_BASE+0x60	BTIM0_ETR 外部触发输入
GPIO_BTIM1_TOGP_MAP	GPIO_MAP_BASE+0x64	BTIM1_TOGP 输出
GPIO_BTIM1_TOGN_MAP	GPIO_MAP_BASE+0x68	BTIM1_TOGN 输出
GPIO_BTIM1_ETR_MAP	GPIO_MAP_BASE+0x6C	BTIM1_ETR 外部触发输入
GPIO_BTIM2_TOGP_MAP	GPIO_MAP_BASE+0x70	BTIM2_TOGP 输出
GPIO_BTIM2_TOGN_MAP	GPIO_MAP_BASE+0x74	BTIM2_TOGN 输出
GPIO_BTIM2_ETR_MAP	GPIO_MAP_BASE+0x78	BTIM2_ETR 外部触发输入
GPIO_UART0_TXD_MAP	GPIO_MAP_BASE+0x7C	UART0 数据传输口
GPIO_UART0_RXD_MAP	GPIO_MAP_BASE+0x80	UART0 接收脚
GPIO_UART0_CK_MAP	GPIO_MAP_BASE+0x84	UART0 时钟输出/输入
GPIO_UART1_TXD_MAP	GPIO_MAP_BASE+0x88	UART1 数据传输口
GPIO_UART1_RXD_MAP	GPIO_MAP_BASE+0x8C	UART1 接收脚
GPIO_UART1_CK_MAP	GPIO_MAP_BASE+0x90	UART1 时钟输出/输入
GPIO_SPI_MOSI_MAP	GPIO_MAP_BASE+0x94	SPI 的数据口, 主机的输出和从机的输入
GPIO_SPI_MISO_MAP	GPIO_MAP_BASE+0x98	SPI 的数据口, 主机的输入和从机的输出
GPIO_SPI_SCK_MAP	GPIO_MAP_BASE+0x9C	SPI 的时钟口
GPIO_SPI_CS_MAP	GPIO_MAP_BASE+0xA0	SPI 的片选口
GPIO_I2C_SCL_MAP	GPIO_MAP_BASE+0xA4	IIC 时钟口
GPIO_I2C_SDA_MAP	GPIO_MAP_BASE+0xA8	IIC 数据口
GPIO_VC0_OUT_MAP	GPIO_MAP_BASE+0xAC	VC0 输出脚
GPIO_VC1_OUT_MAP	GPIO_MAP_BASE+0xB0	VC1 输出脚
GPIO_ADC_START_MAP	GPIO_MAP_BASE+0xB4	ADC 启动转换信号输出
GPIO_ADC_SQR0TRGO_MAP	GPIO_MAP_BASE+0xB8	ADC 序列 0 GPIO 触发信号输入
GPIO_ADC_SQR1TRGO_MAP	GPIO_MAP_BASE+0xBC	ADC 序列 1 GPIO 触发信号输入
GPIO_ADC_SQR2TRGO_MAP	GPIO_MAP_BASE+0xC0	ADC 序列 2 GPIO 触发信号输入
GPIO_ADC_SQR3TRGO_MAP	GPIO_MAP_BASE+0xC4	ADC 序列 3 GPIO 触发信号输入

## 9.6 寄存器描述

### 9.6.1 GPIOx\_OPENDRAIN GPIO 输出模式寄存器( x=A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 0000(GPIOA)  
0x0000 0000(GPIOB)

位编号	位符号	权限	说明
31-30	RFU	-	保留位，请保持默认值
15-0	PINy y=0 ~ 15	RW	端口输入输出方向控制 0：推挽输出 1：开漏输出

### 9.6.2 GPIOx\_PDR GPIO 下拉电阻寄存器( x=A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 0000(GPIOA)  
0x0000 0000(GPIOB)

位编号	位符号	权限	说明
31-30	RFU	-	保留位，请保持默认值
15-0	PINy y=0 ~ 15	RW	端口下拉电阻使能控制 0：禁止下拉电阻 1：使能下拉电阻

### 9.6.3 GPIOx\_DIR GPIO 输入输出方向寄存器( x=A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 FFFF(GPIOA)  
0x0000 3FFF(GPIOB)

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	PINy y=0 ~ 15	RW	端口输入输出方向控制 0：将端口配置成输出 1：将端口配置成输入 <i>注：输入在数字模式下常开，该位实际上是输出控制位</i>

### 9.6.4 GPIOx\_PUR GPIO 上拉电阻寄存器( x=A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 0000(GPIOA)  
0x0000 0000(GPIOB)

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	PINy y=0 ~ 15	RW	端口上拉电阻使能控制 0：禁止上拉电阻 1：使能上拉电阻

### 9.6.5 GPIOx\_ANALOG GPIO 模拟数字配置寄存器( x=A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 FFFF(GPIOA)  
0x0000 3FFF(GPIOB)

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	PINy y=0 ~ 15	RW	端口模拟 / 数字功能配置 0：将端口配置为数字功能 1：将端口配置为模拟功能

### 9.6.6 GPIOx\_RISEIE GPIO 上升沿中断使能寄存器( x=A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 0000(GPIOA)  
0x0000 0000(GPIOB)

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	PINy y=0 ~ 15	RW	端口上升沿中断使能控制 0：禁止相应端口的上升沿中断 1：使能相应端口的上升沿中断

### 9.6.7 GPIOx\_FALLIE GPIO 下降沿中断使能寄存器( x=A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 0000(GPIOA)  
0x0000 0000(GPIOB)

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	PINy y=0 ~ 15	RW	端口下降沿中断使能控制 0：禁止相应端口的下降沿中断 1：使能相应端口的下降沿中断

### 9.6.8 GPIOx\_HIGHIE GPIO 高电平中断使能寄存器( x=A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 0000(GPIOA)  
0x0000 0000(GPIOB)

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	PINy y=0 ~ 15	RW	端口高电平中断使能控制 0：禁止相应端口的高电平中断 1：使能相应端口的高电平中断

### 9.6.9 GPIOx\_LOWIE GPIO 低电平中断使能寄存器( x=A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 0000(GPIOA)  
0x0000 0000(GPIOB)

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	PINy y=0 ~ 15	RW	端口低电平中断使能控制 0：禁止相应端口的低电平中断 1：使能相应端口的低电平中断

### 9.6.10 GPIOx\_ISR GPIO 中断标志寄存器( x=A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 0000(GPIOA)  
0x0000 0000(GPIOB)

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	PINy y=0 ~ 15	RO	端口中断状态标志 0：未检测到已使能的中断 1：已检测到已使能的中断

### 9.6.11 GPIOx\_FILTER GPIO 数字滤波器使能寄存器 ( x=A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 0000(GPIOA)  
0x0000 0000(GPIOB)

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	PINy y=0 ~ 15	RW	端口滤波时钟使能控制 0：禁止相应端口滤波时钟 1：使能相应端口滤波时钟 注：滤除宽度小于 GPIO_FLTCLK.CT 时钟宽度的脉冲

### 9.6.12 GPIOx\_FILTCR GPIO 数字滤波器时钟配置寄存器( x=A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 0000(GPIOA)  
0x0000 0000(GPIOB)

位编号	位符号	权限	说明
31-3	RFU	-	保留位，请保持默认值
2-0	CLK	RW	端口滤波时钟选择 000：HCLK / 2 001：HCLK / 4 010：HCLK / 8 011：LSI（约 32.8kHz） 101：BTIM0 溢出

### 9.6.13 GPIOx\_ICR GPIO 中断标志清除寄存器( x =A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 0000(GPIOA)  
0x0000 0000(GPIOB)

位编号	位符号	权限	说明
31-30	RFU	-	保留位，请保持默认值
29-0	PINy y=0 ~ 15	R1W0	端口中断状态标志清除 W0：清除相应的中断标志位 W1：无功能

### 9.6.14 GPIOx\_LCKR GPIO 端口配置锁定寄存器( x =A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 0000(GPIOA)  
0x0000 0000(GPIOB)

位编号	位符号	权限	说明
31-16	KEY	WO	仅当 KEY 为 0x5A5A 时，对 GPIO_LCKER 寄存器的写操作有效
15-0	PINy y=0 ~ 15	RW	端口配置锁定 0：配置寄存器的相应比特可以修改 1：配置寄存器的相应比特不可修改 注：配置寄存器如下所示： GPIOx_ANALOG、GPIOx_DIR、GPIOx_OPENDRAIN、 GPIOx_PDR、GPIOx_PUR、 GPIOx_RISEIE、GPIOx_FALLIE、GPIOx_HIGHIE、 GPIOx_LOWIE

### 9.6.15 GPIOx\_IDR GPIO 输入数据寄存器( x =A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 0000(GPIOA)  
0x0000 0000(GPIOB)

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	PINy y=0 ~ 15	RO	读出端口输入电平状态 0：端口为低电平 1：端口为高电平

### 9.6.16 GPIOx\_ODR GPIO 输出数据寄存器( x =A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x---- ----(GPIOA)  
0x---- ----(GPIOB)

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	PINy y=0 ~ 15	RW	设置端口输出电平 0：设置端口输出低电平 1：设置端口输出高电平



### 9.6.17 GPIOx\_BRR GPIO 端口位清零寄存器( x =A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 0000(GPIOA)  
0x0000 0000(GPIOB)

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	PINy y=0 ~ 15	R0W1	端口位清零控制 0：不影响 GPIO_ODR 寄存器相应的比特 1：设置 GPIO_ODR 寄存器相应的比特为 0

### 9.6.18 GPIOx\_BSR GPIO 端口位置位寄存器( x =A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 0000(GPIOA)  
0x0000 0000(GPIOB)

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	PINy y=0 ~ 15	R0W1	端口位置位控制 0：不影响 GPIO_ODR 寄存器相应的比特 1：设置 GPIO_ODR 寄存器相应的比特为 1

### 9.6.19 GPIOx\_TOG GPIO 端口位翻转寄存器( x =A, B )

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 0000(GPIOA)  
0x0000 0000(GPIOB)

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	PINy y=0 ~ 15	R0W1	端口位翻转控制 0：不影响 GPIO_ODR 寄存器相应的比特 1：对 GPIO_ODR 寄存器相应的比特取反

### 9.6.20 GPIO\_xxx\_MAP GPIO xxx 复用功能映射配置寄存器

地址：请参见表 9-1GPIO 寄存器列表      复位值：0x0000 003F

位编号	位符号	权限	说明
31-16	KEY	WO	仅当 KEY 为 0x5A5A 时，对 GPIO_xxx_MAP 寄存器的写操作有效
15-6	RFU	-	保留位，请保持默认值
5-4	PORT	RW	复用功能映射到 PORT 00：GPIOA 01：GPIOB 其他位：保留
3-0	PIN	RW	复用功能映射到 PIN



## 10 循环冗余校验（CRC）

### 10.1 概述

循环冗余校验 (CRC) 主要应用于核实数据传输或数据存储的正确性和完整性。本模块支持计算 CRC 值和检验 CRC 值。

### 10.2 主要特性

- 8bit 输入数据位宽
- 2 种多项式
  - CRC-16 多项式 1:  $x^{16} + x^{15} + x^2 + 1$
  - CRC-16 多项式 2:  $x^{16} + x^{12} + x^2 + 1$
- 8 种常用的算法
  - 基于多项式, 初始值, 结果异或值, 输入/输出反转的组合

### 10.3 功能描述

CRC 单元通过对输入数据（或输入数据的反转）和选定的多项式值进行‘除’运算，得到的余数再进行反转或者不反转，以及异或处理，得到 CRC 计算结果。

CRC 单元在使用之前，需要设置 SYSCTRL\_AHBEN.CRC 为 1，打开 CRC 单元的配置时钟及工作时钟，一般在系统初始化时进行设置。

#### 10.3.1 算法模式

表 10-1CRC 算法模式

算法名称	多项式值	初始值	输入反转	输出反转	结果异或值
CRC16_IBM	0x8005	0x0000	True	True	0x0000
CRC16_MAXIM	0x8005	0x0000	True	True	0xFFFF
CRC16_USB	0x8005	0xFFFF	True	True	0xFFFF
CRC16_MODBUS	0x8005	0xFFFF	True	True	0x0000
CRC16_CCITT	0x1021	0x0000	True	True	0x0000
CRC16_CCITT_False	0x1021	0xFFFF	False	False	0x0000
CRC16_X25	0x1021	0xFFFF	True	True	0xFFFF
CRC16_XMODEM	0x1021	0x0000	False	False	0x0000

各参数含义如下：

#### ● 多项式值

多项式是码组的描述，如 CRC-16 多项式 2:  $x^{16} + x^{12} + x^2 + 1$ ，对应的码组是 1 0001 0000 0010 0001。因为多项式码组的最高位固定为 1，且最高位的位置已知，因此一般将最高位 1 去掉后的码组称为多项式值，如CRC-16 多项式 2 的值为 0001 0000 0010 0001，即 0x1021。

#### ● 初始值

在计算 CRC 校验值之前，CRC 寄存器的初始值。

- **输入数据反转**

即在计算开始前,将需要计算 CRC 校验值的数据进行高低序位反转,如数据位 1011,反转后为 1101。

- **输出数据反转**

在 CRC 计算结束后,与结果异或值进行异或之前,将计算值进行高低序位反转,如计算结果为 1011,反转后为 1101。

- **结果异或值**

在 CRC 计算结束后,得到的 CRC 计算值与结果异或值进行异或操作,就得到了最终的 CRC 校验值。

### 10.3.2 输入数据位宽

CRC 计算单元支持 8bit 输入数据位宽,输入数据时,先输入低字节再输入高字节。

例如,用户需要计算 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77 这一组数据的 CRC 校验值,写入顺序为: 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77。

代码示例:

```
HC_CRC -> DR = 0x00 ;  
HC_CRC -> DR = 0x11 ;  
HC_CRC -> DR = 0x22 ;  
HC_CRC -> DR = 0x33 ;  
HC_CRC -> DR = 0x44 ;  
HC_CRC -> DR = 0x55 ;  
HC_CRC -> DR = 0x66 ;  
HC_CRC -> DR = 0x77 ;
```

## 10.4 编程示例

### 10.4.1 CRC16\_CCITT 算法模式

**步骤 1:** 设置 CRC\_CR.MODE 为 0x04, 选择 CRC16\_CCITT 算法模式, 硬件自动配置 CRC 寄存器初始值为 0x0000;

**步骤 2:** 将待编码的原始数据依次写入数据寄存器 CRC\_DR;

**步骤 3:** 读取 CRC\_RESULT 获取 CRC 校验值。

代码示例:

```
tempdata = HC_CRC -> RESULT ;
```

## 10.5 寄存器列表

CRC 基地址: CRC\_BASE = 0x4000 1000

表 10-2 CRC 寄存器列表

寄存器名称	寄存器地址	寄存器描述
CRC_CR	CRC_BASE + 0x00	控制寄存器
CRC_DR	CRC_BASE + 0x04	数据寄存器
CRC_RESULT	CRC_BASE + 0x08	结果寄存器

## 10.6 寄存器描述

### 10.6.1 CRC\_CR 控制寄存器

地址: 请参见表 10-2 CRC 寄存器列表 复位值: 0x0000 0000

位编号	位符号	权限	说明
31-3	RFU	-	保留位, 请保持默认值
2-0	MODE	RW	CRC 算法模式配置 000: CRC16_IBM 001: CRC16_MAXIM 010: CRC16_USB 011: CRC16_MODBUS 100: CRC16_CCITT 101: CRC16_CCITT_False 110: CRC16_X25 111: CRC16_XMODEM

### 10.6.2 CRC\_DR 数据寄存器

地址: 请参见表 10-2 CRC 寄存器列表 复位值: 0x0000 0000

位编号	位符号	权限	说明
31-8	RFU	-	保留位, 请保持默认值
7-0	DR8	RW	数据写入寄存器, 位宽 8bit 时使用

### 10.6.3 CRC\_RESULT 结果寄存器

地址: 请参见表 10-2 CRC 寄存器列表 复位值: 0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位, 请保持默认值
15-0	RESULT	RO	CRC16 计算结果

# 11 通用计算模块（ALU）

## 11.1 概述

HC32F0653 的通用计算模块提供 2 种运算模式，可用于快速运算。

## 11.2 主要特性

- 除法运算模式：用于 32bit 的除法运算，32 个 PCLK 周期
- 开平方运算模式：用于 32bit 无符号数的开平方运算，16 个 PCLK 周期

## 11.3 功能描述

### 11.3.1 除法运算模式

ALU 模块可以用于 32bit \* 32bit 的有符号数除法运算。除数和被除数分别存放于 DSP\_DIS 和 DSP\_DID 寄存器中。当 ALU 模块用于除法运算时，可按如下步骤配置：

1. 写入被除数 DSP\_DID
2. 写入除数 DSP\_DIS 的同时，触发一次除法操作
3. 等待计算完成
4. 读取 ALU\_QUO，ALU\_REM，得到商和余数

**注意：**除法的结果向零取整。

### 11.3.2 开方运算模式

ALU 模块开方模式被开方数为 32 位无符号数，结果为 16 位无符号数，16 个总线周期（最高 48/60MHz）完成计算。当 ALU 模块用于开方运算时，可按如下步骤配置：

1. 写入被开方数 ALU\_RAD，触发开方器开始运算
2. 等待计算完成
3. 读取 ALU\_SQRT 得到开方结果

**注意：**开方运算的结果经过四舍五入。

## 11.4 寄存器列表

ALU 基地址：ALU\_BASE = 0x4000 1400

表 11-1ALU 寄存器列表

寄存器名称	寄存器地址	寄存器描述
ALU_DID	ALU_BASE+0x20	ALU 除法操作被除数
ALU_DIS	ALU_BASE+0x24	ALU 除法操作除数
ALU_QUO	ALU_BASE+0x28	ALU 除法操作商
ALU_REM	ALU_BASE+0x2C	ALU 除法操作余数
ALU_RAD	ALU_BASE+0x30	ALU 开方操作被开方数
ALU_SQRT	ALU_BASE+0x34	ALU 开方操作平方根
ALU_SR	ALU_BASE+0x38	ALU 状态寄存器

## 11.5 寄存器描述

有关寄存器描述里所使用的缩写，请参见 1 文档约定章节。

### 11.5.1 ALU\_DID ALU 除法操作被除数

地址：请参见表 11-1ALU 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-0	DR	RW	除法被除数

### 11.5.2 ALU\_DIS ALU 除法操作除数

地址：请参见表 11-1ALU 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-0	DR	RW	除法操作除数

### 11.5.3 ALU\_QUO ALU 除法商寄存器

地址：请参见表 11-1ALU 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-0	DR	R	除法商

### 11.5.4 ALU\_REM ALU 除法余数寄存器

地址：请参见表 11-1ALU 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-0	DR	R	除法余数

### 11.5.5 ALU\_RAD ALU 被开方数寄存器

地址：请参见表 11-1ALU 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-0	DR	RW	被开方数

### 11.5.6 ALU\_SQRT ALU 平方根寄存器

复地址：请参见表 11-1ALU 寄存器列表 位值：0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	DR	R	平方根数

### 11.5.7 ALU\_SR ALU 状态寄存器

地址：请参见表 11-1 ALU 寄存器列表    复位值：0x0000 0000

位编号	位符号	权限	说明
31-2	RFU	-	保留位，请保持默认值
1	SQRTOVER	R	平方根计算完成状态位 0：空闲状态 1：计算状态
0	DIVOV	R	除法计算完成状态位 0：空闲状态 1：计算状态

## 12 独立看门狗定时器 (IWDT)

### 12.1 概述

HC32F0653 内部集成独立看门狗定时器 (IWDT)。一旦启动 IWDT，用户需要在规定时间间隔内对 IWDT 的计数器 进行重载，否则计数器溢出会触发复位或产生中断信号。IWDT 启动后，可停止计数。可选择在深度休眠模式下 IWDT 保持运行或暂停计数。

专门设置的键值寄存器，可以锁定 IWDT 的关键寄存器，防止寄存器被意外修改。

### 12.2 主要特性

- 12bit 的向下计数器
- 可编程时钟预分频周期
- 溢出可触发中断或复位
- 寄存器保护锁功能
- 深度休眠模式下可暂停计数
- 窗口比较匹配触发中断
- 支持软硬件看门狗

### 12.3 功能描述

#### 12.3.1 功能框图

IWDT 功能框图如下图所示：

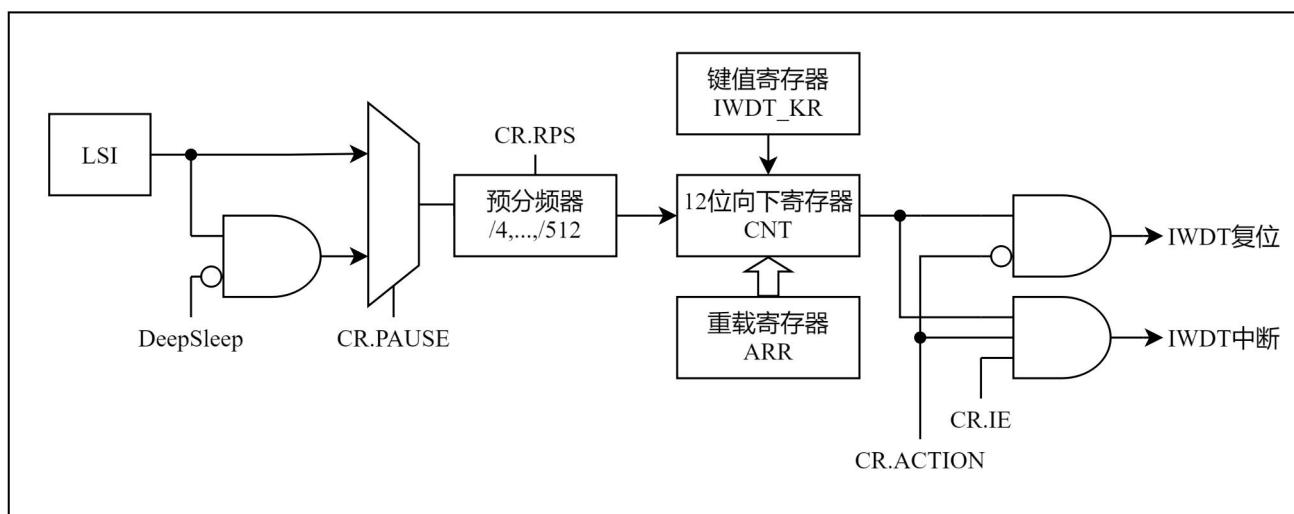


图 12-1 IWDT 功能框图

IWDT 由一个 12 位可重载的向下计数器实现，其计数时钟源为内部低速 LSI 时钟，通过控制寄存器 IWDT\_CR 的 PRS 位域可对其时钟源 LSI 信号进行 4 ~ 512 的预分频。IWDT 计数器发生溢出时可选择产生中断和复位信号。

## 12.3.2 工作方式

启动 IWDT 的计数器，需要向键值寄存器 IWDT\_KR 写入 0xCCCC，计数器开始从 0xFFFF 向下计数。

在计数器减到 0 之前，向 IWDT\_KR 寄存器写入 0xAAAA，会触发计数器重载，将 ARR 寄存器值加载到计数器。

当 IWDT 计数器值递减到 0，会产生溢出事件，溢出事件可触发 MCU 复位或产生 IWDT 中断信号，同时触发计数器重载。控制寄存器 IWDT\_CR 的 ACTION 和 IE 位域，用于控制看门狗溢出时是否产生中断和复位，如下表所示：

表 12-1 IWDT 工作方式

IWDT_CR.ACTION	IWDT_CR.IE	IWDT 溢出后动作
0	x	复位
1	1	不复位，只产生中断
1	0	不复位，不产生中断

IWDT 在 MCU 进入深度休眠模式时，可选择暂停 IWDT 计数，从而达到更低的系统整体功耗：控制寄存器 IWDT\_CR 的 PAUSE 位域为 0，深度休眠模式时保持 IWDT 定时器运行；为 1 时暂停计数，当 MCU 退出深度休眠模式时 IWDT 自动恢复计数。

## 12.3.3 窗口选项

向窗口寄存器 IWDT\_WINR 写入一个小于重载寄存器 IWDT\_ARR 的值，可使 IWDT 工作于窗口看门狗模式。

IWDT\_WINR 寄存器的默认值是 0x0FFF，即窗口选项默认是关闭的。

修改 IWDT\_WINR 寄存器的值，会触发重载操作，将 ARR 寄存器值加载到计数器。

使用 IWDT 窗口功能，用户应在计数器值小于等于 IWDT\_WINR 窗口值，并且递减到 0 之前进行重载操作，以避免产生复位或看门狗定时器溢出。在看门狗计数器的值大于窗口值时进行重载操作，将触发系统复位。

可以通过设置 IWDT\_CR.WINCIE 来决定当 IWDT\_CNT 等于 IWDT\_WINR 的时候，是否产生匹配中断。

## 12.3.4 寄存器锁定功能

通过 IWDT 的键值寄存器 IWDT\_KR，可配置 IWDT 的重要寄存器 IWDT\_CR、IWDT\_ARR、IWDT\_WINR 为锁定状态或解除锁定，锁定状态下不可对寄存器进行修改操作。

向 IWDT\_KR 寄存器写入 0x5555，解除对 IWDT\_CR、IWDT\_ARR、IWDT\_WINR 寄存器的锁定；向 IWDT\_KR 寄存器写入其他任何值，启动锁定保护。

芯片在上电复位后，IWDT\_CR、IWDT\_ARR、IWDT\_WINR 寄存器默认处于锁定状态，用户需先解除锁定，才可对其进行修改操作。

## 12.3.5 启动刷新与停止

配置 IWDT\_KR 寄存器，可实现 IWDT 的启动、刷新和停止操作：

- 写入 0xCCCC，启动 IWDT。
- 写入 0xAAAA，重载计数器，即刷新 IWDT。
- 顺序写入 0x5A5A、0xA5A5，停止 IWDT。
- 依次写入 0x6969、0x9696，启动 IWDT 计数器，并且无法停止

注：上述操作会同时启动 IWDT 寄存器锁定保护。



### 12.3.6 状态寄存器

状态寄存器 IWDT\_SR，指示 IWDT 当前运行状态或寄存器更新状态，如下表所示：

表 12-2 IWDT 状态指示

IWDT_SR 位	名称	描述	1	0
6	WINC	WINR 寄存器匹配标志	产生匹配	未匹配
5	RELOAD	计数器重载标志	正在重载	重载完成
4	RUN	运行标志	运行	未运行
3	OV	溢出标志	产生溢出	未溢出
2	WINRF	WINR 寄存器更新标志	正在更新	更新完成
1	ARRF	ARR 寄存器更新标志	正在更新	更新完成
0	CRF	CR 寄存器更新标志	正在更新	更新完成

为确保对 IWDT 寄存器的操作正确性，用户在更新 IWDT\_WINR、IWDT\_ARR、IWDT\_CR 之后，需要检查 WINRF、ARRF、CRF 标志位是否为0，以确认操作是否完成。

为确保对 IWDT 的重载操作，用户在进行重载操作后，应当检查 RELOAD 标志位是否为 0。

### 12.3.7 定时时长设定

IWDT 的计数时钟源为内部低速 LSI 时钟（时钟频率约为 32.8kHz，具体请参阅数据手册），通过控制寄存器 IWDT\_CR 的 PRS 位域，可对其时钟源 LSI 信号进行分频，如下表所示：

表 12-3 IWDT 分频系数表

IWDT_CR.PRS	预分频值
000	4
001	8
010	16
011	32
100	64
101	128
110	256
111	512

看门狗定时时长计算公式：

$$T = (4 \times 2^{\text{PRS}} \div f) \times (\text{ARR} + 1)$$

其中，f 为时钟源 LSI 的频率，PRS 为预分频系数，ARR 为重载值。

所以当时钟源 LSI 的频率为 32800Hz 时，IWDT 的最长和最短定时范围：

$$\text{IWDT最短定时} = (4 \times 2^0 \div 32800) \times (0x000 + 1) \approx 122 \mu\text{S}$$

$$\text{IWDT最长定时} = (4 \times 2^7 \div 32800) \times (0xFFF + 1) \approx 63.9 \text{ S}$$

例：当时钟源 LSI 的频率为 32800Hz 时，设置预分频值为 64，重载值为 512，则：

$$\text{IWDT定时时长} = (4 \times 2^4 \div 32800) \times (512 + 1) \approx 1 \text{ S}$$

## 12.3.8 调试支持

IWDT 支持在调试模式下停止或继续计数，通过调试状态定时器控制寄存器 SYSCTRL\_DEBUG 的 IWDT 位域来设置。

- 设置 SYSCTRL\_DEBUG.IWDT 为 1，则在调试状态时暂停 IWDT 的计数器计数。
- 设置 SYSCTRL\_DEBUG.IWDT 为 0，则在调试状态时 IWDT 的计数器继续计数。

## 12.4 编程示例

### 12.4.1 配置 IWDT 为独立看门狗

1. 设置 SYSCTRL\_APBEN.IWDT 为 1，使能 IWDT 的配置时钟；
2. 向 IWDT\_KR 寄存器写入 0xCCCC，启动 IWDT；  
**注：需要启动 IWDT 后，才可对相关寄存器进行修改。**
3. 向 IWDT\_KR 寄存器写入 0x5555，解除 IWDT 寄存器锁定功能；
4. 配置 IWDT\_CR，配置看门狗计数时钟与 LSI 振荡器的预分频值、溢出后动作、深度休眠模式下是否自动暂停；
5. 配置 IWDT\_ARR，配置看门狗的溢出周期；
6. 等待 IWDT\_SR.ARRF 和 IWDT\_SR.CRF 变为 0，等待重载值和 CR 寄存器更新完成；
7. 向 IWDT\_KR 寄存器写入 0xAAAA，加载 ARR 到 IWDT 计数器。

### 12.4.2 配置 IWDT 为窗口看门狗

1. 设置 SYSCTRL\_APBEN.IWDT 为 1，使能 IWDT 的配置时钟；
2. 向 IWDT\_KR 寄存器写入 0xCCCC，启动 IWDT；  
**注：需要启动 IWDT 后，才可对相关寄存器进行修改。**
3. 向 IWDT\_KR 寄存器写入 0x5555，解除 IWDT 寄存器锁定功能；
4. 配置 IWDT\_CR，配置看门狗计数时钟与 LSI 振荡器的预分频值、溢出后动作、深度休眠模式下是否自动暂停；
5. 配置 IWDT\_ARR，配置看门狗的重载值；
6. 配置 IWDT\_WINR，配置窗口大小，注意 IWDT\_WINR 必须小于 IWDT\_ARR 重载值；
7. 等待 IWDT\_SR.ARRF、IWDT\_SR.WINRF 和 IWDT\_SR.CRF 变为 0，等待重载值、窗口寄存器和 CR 寄存器更新完成
8. 向 IWDT\_KR 寄存器写入 0xAAAA，加载 ARR 到 IWDT 计数器。

### 12.4.3 刷新 IWDT（喂狗操作）

1. 向 IWDT\_KR 寄存器写入 0xAAAA，加载 ARR 到计数器；
2. 等待 IWDT\_SR.RELOAD 变为 0，等待重载操作完成。

## 12.5 寄存器列表

IWDT 基地址: IWDT\_BASE = 0x4001 1000

表 12-4 IWDT 寄存器列表

寄存器名称	寄存器地址	寄存器描述
IWDT_KR	IWDT_BASE + 0x00	键值寄存器
IWDT_CR	IWDT_BASE + 0x04	控制寄存器
IWDT_ARR	IWDT_BASE + 0x08	重载寄存器
IWDT_SR	IWDT_BASE + 0x0C	状态寄存器
IWDT_WINR	IWDT_BASE + 0x10	窗口寄存器
IWDT_CNT	IWDT_BASE + 0x24	计数值寄存器

## 12.6 寄存器描述

### 12.6.1 IWDT\_KR 键值寄存器

地址: 请参见表 12-4 IWDT 寄存器列表 复位值: 0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位, 请保持默认值
15-0	KR	WO	<p>写入 0xCCCC: 启动 IWDT 计数器</p> <p>写入 0xAAAA: 重载 IWDT 计数值</p> <p>依次写入 0x5A5A、0xA5A5: 停止看门狗</p> <p>依次写入 0x6969、0x9696, 启动 IWDT 计数器, 并且无法停止, 锁定 LSI 使能位, LSI 无法关闭</p> <p>写入 0x5555: 解除 IWDT_ARR、IWDT_WINR、IWDT_CR 的写保护</p> <p>写入非 0x5555: 使能 IWDT_ARR、IWDT_WINR、IWDT_CR 的写保护</p> <p>注: 假如 FLASH_OPTCR.IWDT 配置为 0, 相当于写入 0xCCCC, 芯片复位结束后立刻启动计数。</p>

### 12.6.2 IWDT\_CR 控制寄存器

地址: 请参见表 12-4 IWDT 寄存器列表 复位值: 0x0000 0000

位编号	位符号	权限	说明
31-7	RFU	-	保留位, 请保持默认值
6	WINCIE	RW	<p>IWDT 窗口匹配中断</p> <p>0: 禁止看门狗窗口匹配中断</p> <p>1: 使能看门狗窗口匹配中断</p>
5	PAUSE	RW	<p>DeepSleep 模式下 IWDT 暂停控制</p> <p>0: DeepSleep 模式下 IWDT 继续运行</p> <p>1: DeepSleep 模式下 IWDT 自动暂停</p>
4	IE	RW	<p>IWDT 中断使能控制</p> <p>0: 禁止看门狗中断</p> <p>1: 使能看门狗中断</p>

3	ACTION	RW	IWDT 溢出后动作配置 0: 看门狗溢出后产生复位 1: 看门狗溢出后产生中断
2-0	PRS	RW	配置看门狗计数时钟与 LSI 振荡器的分频比 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: 512 分频

注：向 IWDT\_KR 寄存器写入 0x5555 后，才可修改本寄存器；IWDT\_SR.CRF 为 0 时，才可修改 IWDT\_CR。

### 12.6.3 IWDT\_ARR 重载寄存器

地址：请参见表 12-4 IWDT 寄存器列表 复位值：0x0000 0FFF

位编号	位符号	权限	说明
31-12	RFU	-	保留位，请保持默认值
11-0	ARR	RW	IWDT 重载值

注：向 IWDT\_KR 寄存器写入 0x5555 后，才可修改本寄存器；IWDT\_SR.ARRF 为 0，才可修改本寄存器。

### 12.6.4 IWDT\_CNT 计数值寄存器

地址：请参见表 12-4 IWDT 寄存器列表 复位值：0x0000 0FFF

位编号	位符号	权限	说明
31-12	RFU	-	保留位，请保持默认值
11-0	CNT	RO	计数值寄存器

### 12.6.5 IWDT\_WINR 窗口寄存器

地址：请参见表 12-4 IWDT 寄存器列表 复位值：0x0000 0FFF

位编号	位符号	权限	说明
31-12	RFU	-	保留位，请保持默认值
11-0	WINR	RW	窗口比较器比较值

### 12.6.6 IWDT\_SR 状态寄存器

地址：请参见表 12-4 IWDT 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-7	RFU	-	保留位，请保持默认值
6	WINC	RW0	WDT 窗口匹配中断标志 R0: WDT 窗口未匹配

			R1: WDT 窗口已经匹配 W0: 清除 WDT 窗口匹配标志
5	RELOAD	RO	WDT 计数器重载标志 0: WDT 计数器已完成重载操作 1: WDT 计数器正在进行重载操作 <b>注: 窗口模式下, 进行重载操作后需要等待该标志变为 0</b>
4	RUN	RO	WDT 运行标志 0: WDT 没有运行 1: WDT 正在运行
3	OV	RW0	WDT 溢出中断标志 R0: WDT 没有溢出 R1: WDT 溢出 W0: 清除 WDT 溢出标志
2	WINRF	RO	WINR 寄存器更新标志 0: WINR 寄存器更新完成 1: WINR 寄存器正在更新 <b>注: 写 WINR 寄存器后需要等待该标志变为 0</b>
1	ARRF	RO	ARR 寄存器更新标志 0: ARR 寄存器更新完成 1: ARR 寄存器正在更新 <b>注: 写 ARR 寄存器后需要等待该标志变为 0</b>
0	CRF	RO	CR 寄存器更新标志 0: CR 寄存器更新完成 1: CR 寄存器正在更新 <b>注: 写 CR 寄存器后需要等待该标志变为 0</b>

## 13 高级定时器（ATIM）

### 13.1 概述

高级定时器 (ATIM) 包含一个 16bit 自动重载计数器，并由一个可编程的预分频器驱动。ATIM 支持定时、计数、复位、门控、触发和编码器等多种工作模式，带 6 路独立的捕获 / 比较通道，可实现 6 路独立 PWM 输出或 6 对带死区互补 PWM 输出或对 6 路输入进行捕获。可用于基本的定时/计数、测量输入信号的脉冲宽度和周期、产生输出波形（PWM、单脉冲、插入死区时间的互补 PWM 等）。

### 13.2 主要特性

ATIM 定时器主要特性：

- 16 位递增、递减、递增/递减自动重载计数器
- 16 位可编程（可以实时修改）预分频器，计数器时钟频率的分频系数为 1 ~ 65535 之间的任意数值
- 支持单次计数模式和连续计数模式
- 6 路独立输入捕获和输出比较通道
- 死区时间可编程的互补 PWM 输出
- 2 个刹车输入，用于将定时器的输出信号置于用户可选的安全配置中
- 触发输入信号（TRGI）控制定时器实现多种从模式
- 定时器级联 ITR 和片内外设互联 ETR
- 支持针对定位的增量（正交）编码器和霍尔传感器电路
- 发生如下事件时生成中断请求：
  - 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - 输入捕获
  - 输出比较
  - 刹车事件
  - 换相事件
  - 编码器事件

## 13.3 功能描述

### 13.3.1 功能框图

ATIM 的功能框图如下图所示：

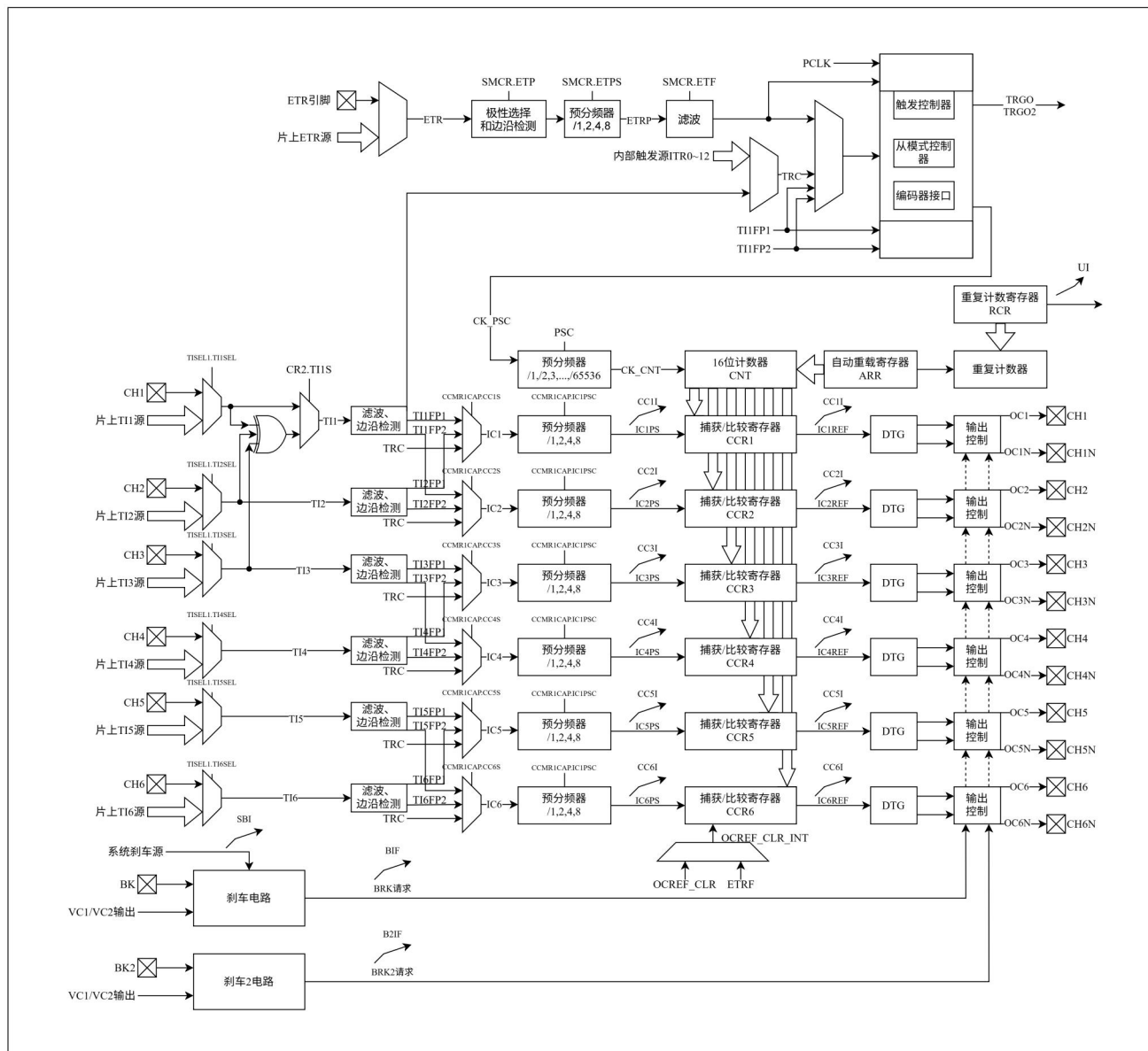


图 13-1ATIM 功能框图

### 13.3.2 时钟源

计数器的计数时钟源可由内部时钟 PCLK、触发信号 TRGI 或外部触发输入信号 ETRF 提供，经预分频器 ATIM\_PSC 分频后驱动计数器进行计数。不同工作模式下具有不同时钟源，具体请参见 1.3.3 工作模式。



### 13.3.3 预分频器

预分频器对 CK\_PSC 时钟进行分频，得到计数时钟 CK\_CNT，以驱动计数器计数。分频系数通过 ATIM\_PSC 寄存器进行设置，支持 1、2、3、4、...、65536 分频。

ATIM\_PSC 寄存器具有缓冲功能，可在运行中修改，新的预分频值将在下一个更新事件发生时生效。下图给出了运行过程中预分频器的分频由 1 变为 2 时的时序图。

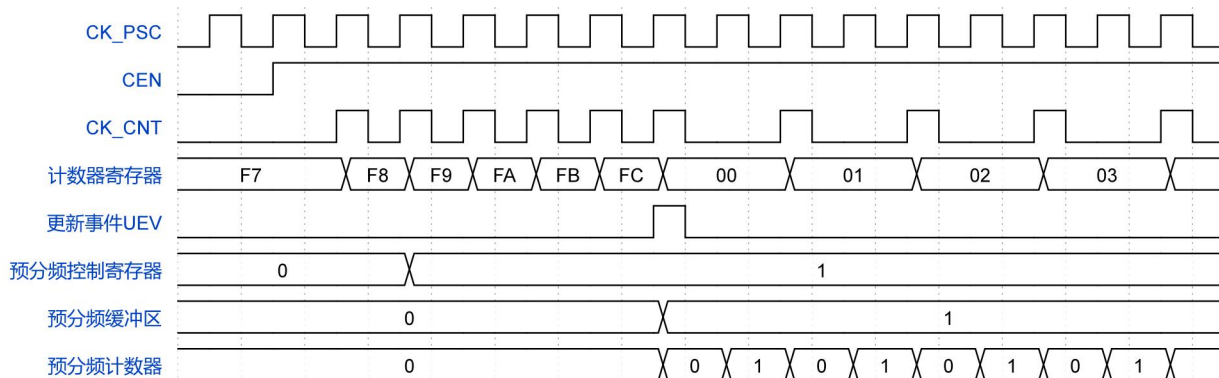


图 13-2 预分频器分频由 1 变为 2 时的时序图

### 13.3.4 计数器与计数模式

计数器可设置为递增计数（边沿对齐模式）、递减计数（边沿对齐模式）或递增 / 递减双向计数（中心对齐模式）。具体通过控制寄存器 ATIM\_CR1 的 CMS 和 DIR 位域进行配置，如下表所示：

表 13-1 计数模式

ATIM_CR1.CMS	ATIM_CR1.DIR	计数模式
0	0	递增计数（边沿对齐模式）
	1	递减计数（边沿对齐模式）
1	-	中心对齐模式 1
10	-	中心对齐模式 2
11	-	中心对齐模式 3

当设置 ATIM\_CR1 寄存器的 CEN 位域为 1 时，计数器开始按设定模式计数，注意计数器将在 CEN 位置 1 时刻的一个时钟周期后开始计数。

#### 递增计数模式

在递增计数模式下，计数器从 0 开始递增计数到重载值 ARR，然后重新从 0 开始递增计数，同时生成计数器上溢出事件并产生上溢出信号 OV（OV 信号会自动清除）。

如果使用了重复计数器，则当溢出次数达到重复值加 1（REP+1）时，将产生更新事件 UEV（UEV 信号会自动清除），否则，将在每次计数器上溢出时产生更新事件 UEV，计数器更新中断标志位 ATIM\_ISR.UIF 被硬件置位，如果允许中断（设置 ATIM\_IER.UIE 为 1），将产生中断请求，设置 ATIM\_ICR.UIF 为 0 清除该标志位。

以下是计数器在不同时钟频率下的操作示例，其中 ATIM\_ARR=0x36、ATIM\_RCR=0x00。



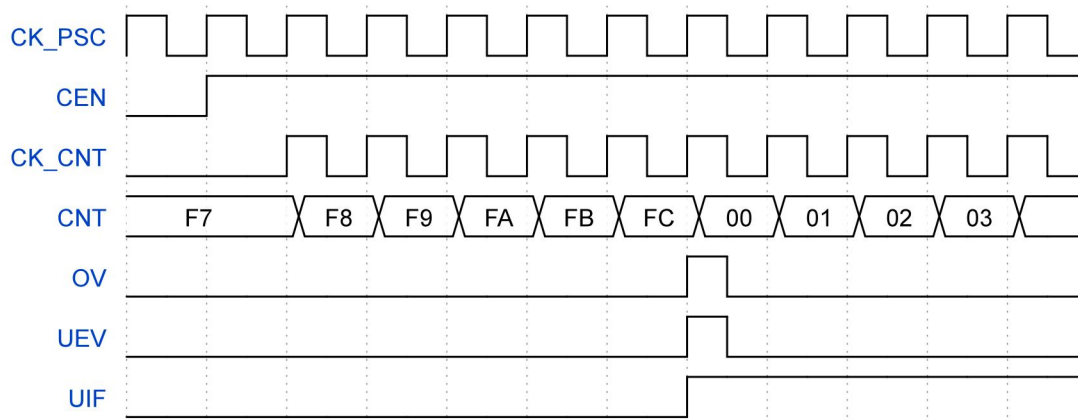


图 13-3 递增计数，内部时钟分频因子为 1

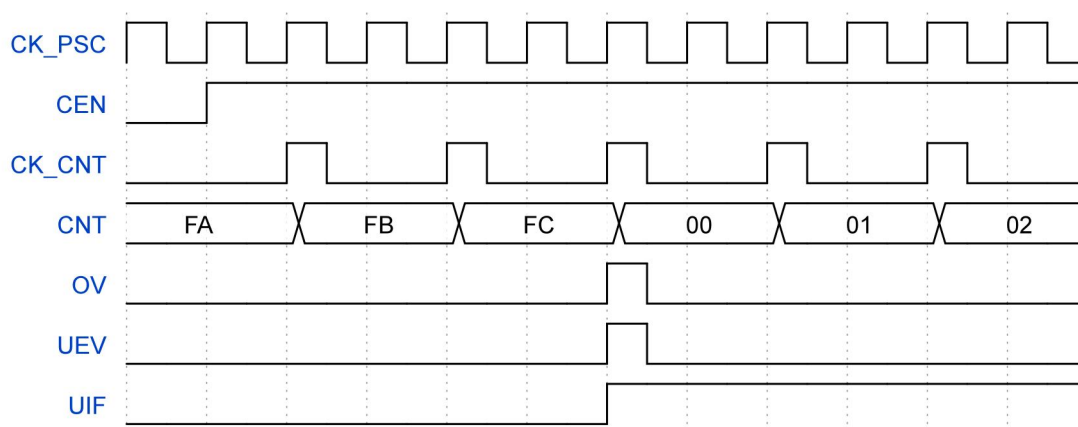


图 13-4 递增计数，内部时钟分频因子为 2

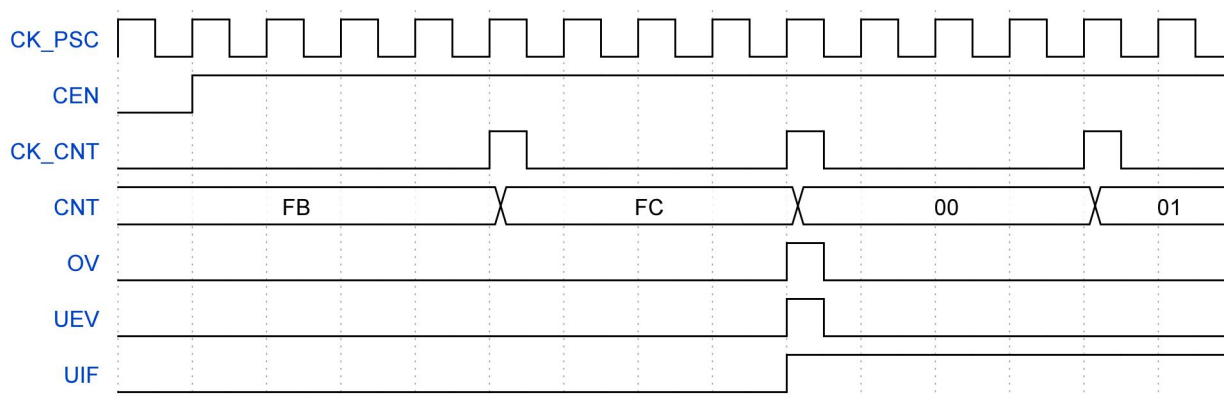


图 13-5 递增计数，内部时钟分频因子为 4

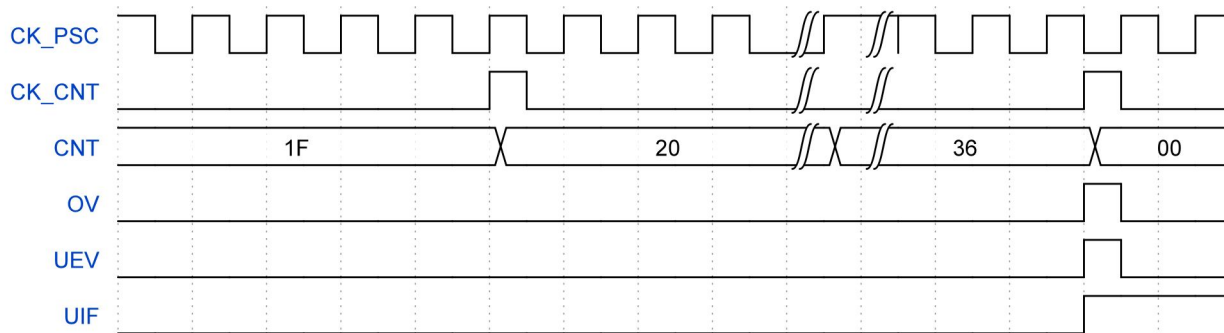


图 13-6 递增计数，内部时钟分频因子为 N

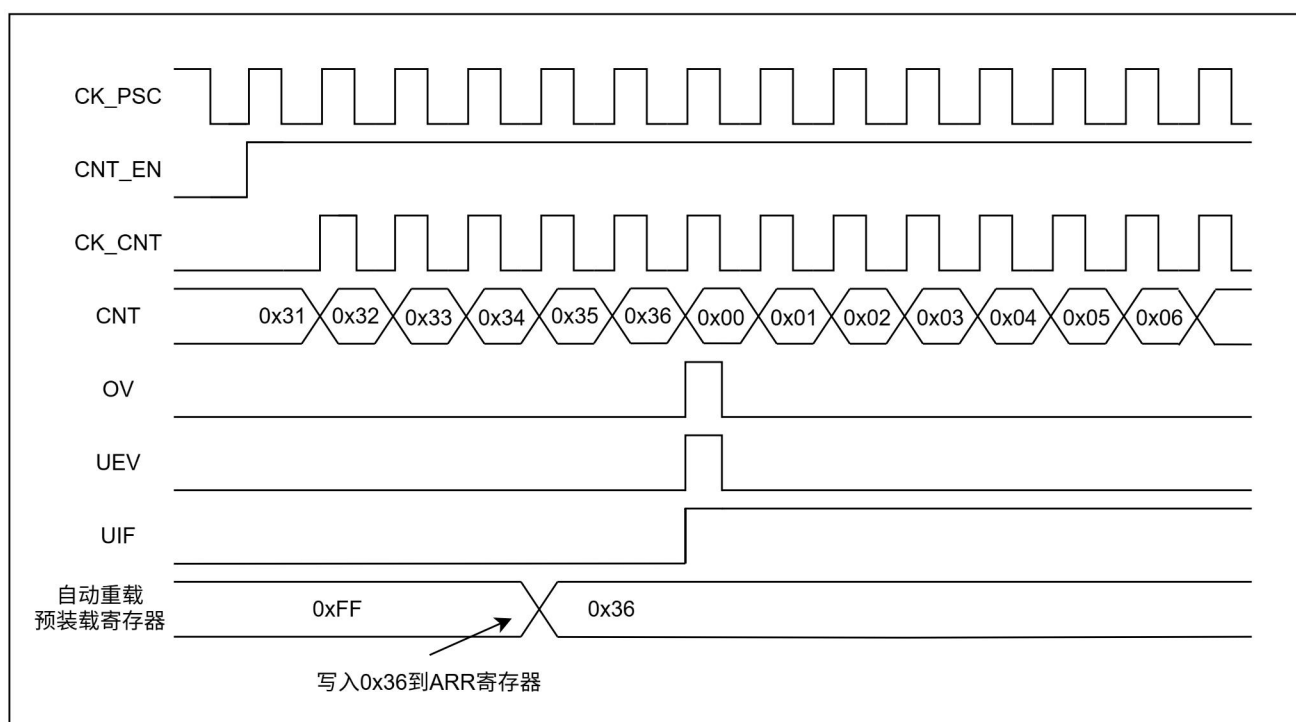


图 13-7 递增计数，当重载缓存禁止时的更新事件 (ATIM\_CR1.ARPE=0)

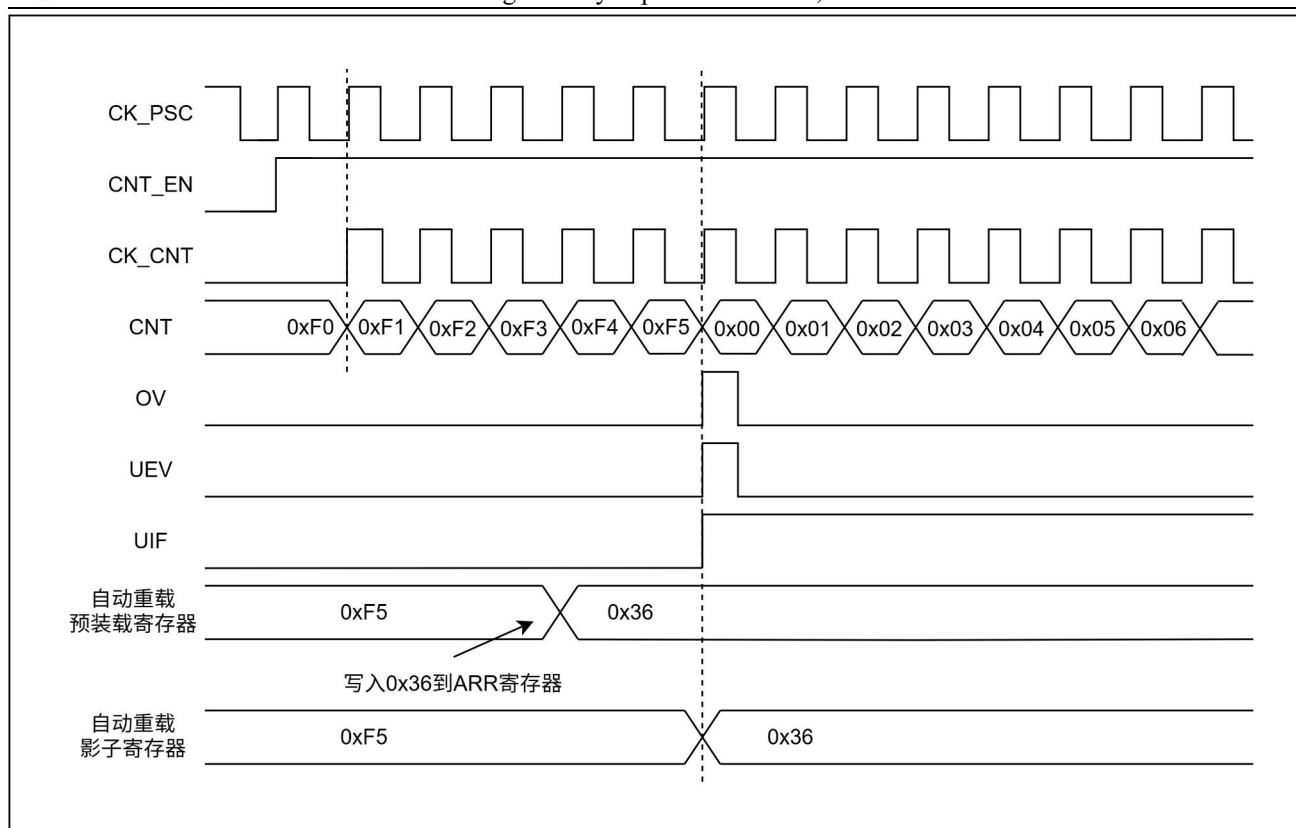


图 13-8 递增计数，当重载缓存使能时的更新事件 (ATIM\_CR1.ARPE=1)

### 递减计数模式

在边沿对齐模式下设置控制寄存器 ATIM\_CR1 的 DIR 位为 1 时，计数器工作在递减计数模式。

在递减计数模式下，硬件自动加载 ARR 值到计数器 CNT，计数器开始递减计数到 0，然后重新装载 ARR 值递减计数，同时生成计数器下溢出事件并产生下溢出信号 UND（UND 信号会自动清除）。

如果使用了重复计数器，则当溢出次数达到重复值加 1 (REP+1) 时，将产生更新事件 UEV (UEV 信号会自动清除)，否则，将在每次计数器下溢时产生更新事件 UEV，计数器更新中断标志位 ATIM\_ISR.UIF 被硬件置位，如果允许中断（设置 ATIM\_IER.UIE 为 1），将产生中断请求，设置 ATIM\_ICR.UIF 为 0 清除该标志位。

以下是计数器在不同时钟频率下的操作示例，其中 ATIM\_ARR = 0x36、ATIM\_RCR = 0x00。

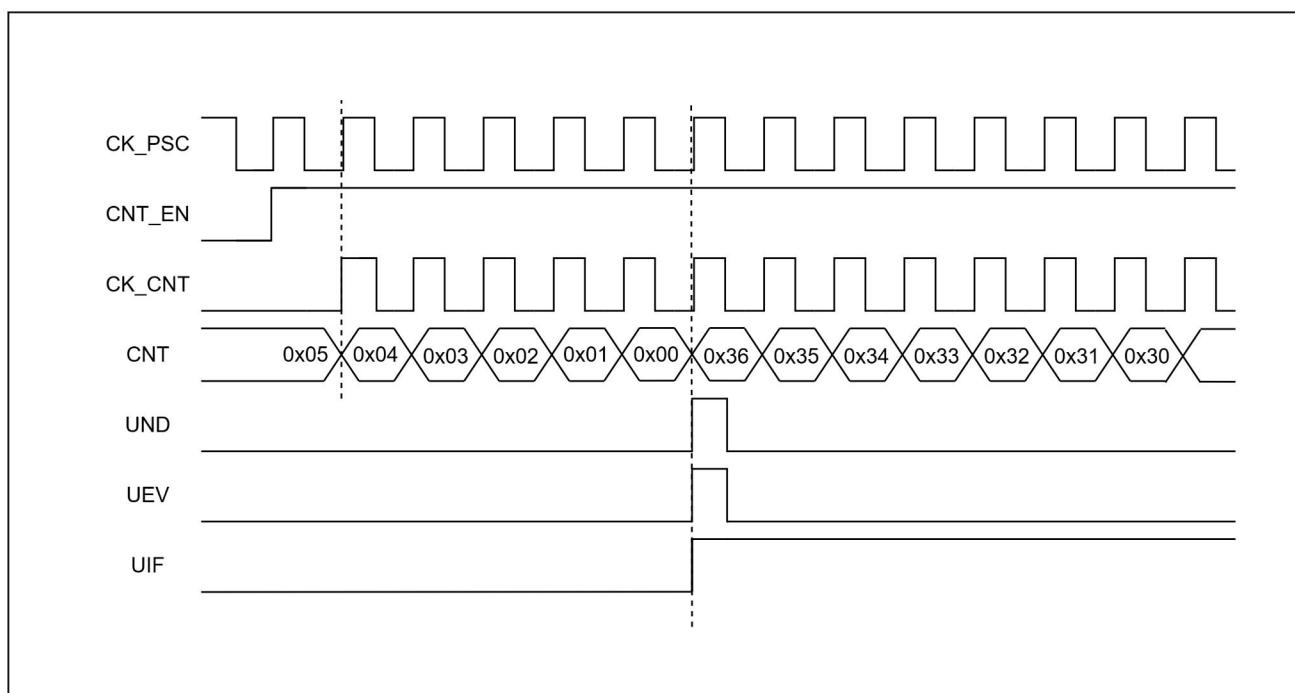


图 13-9 递减计数，内部时钟分频因子为 1

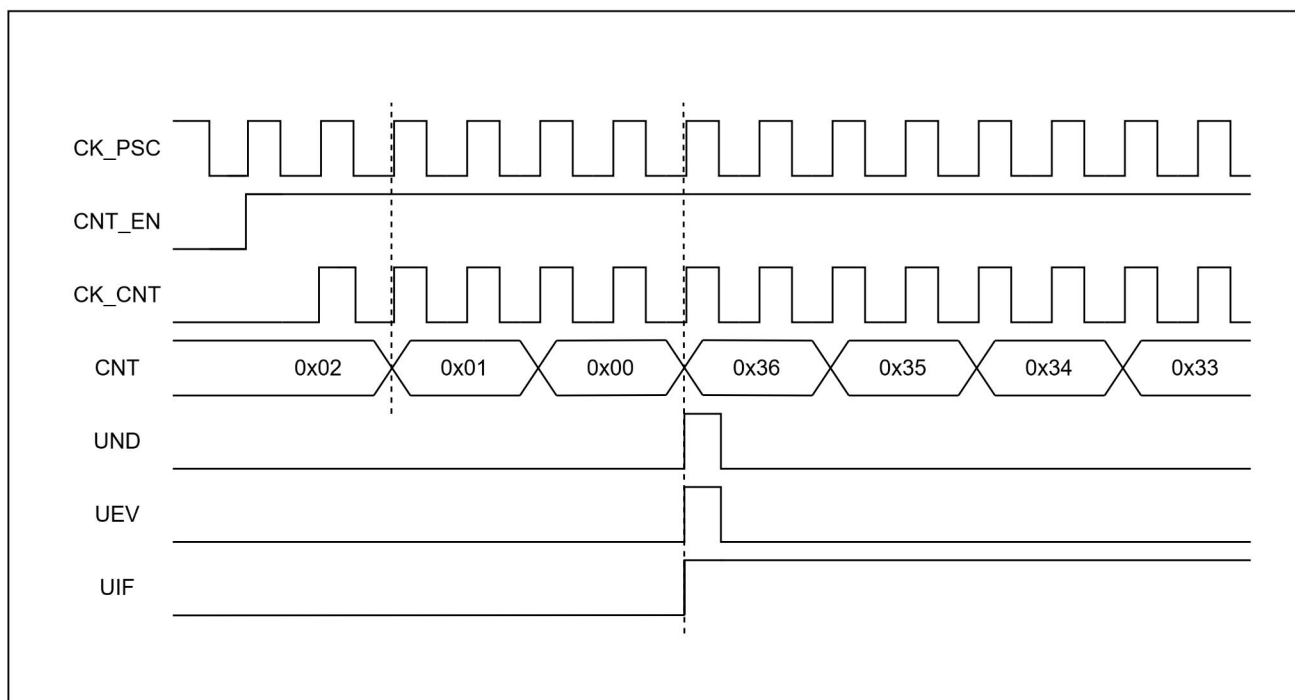


图 13-10 递减计数，内部时钟分频因子为 2

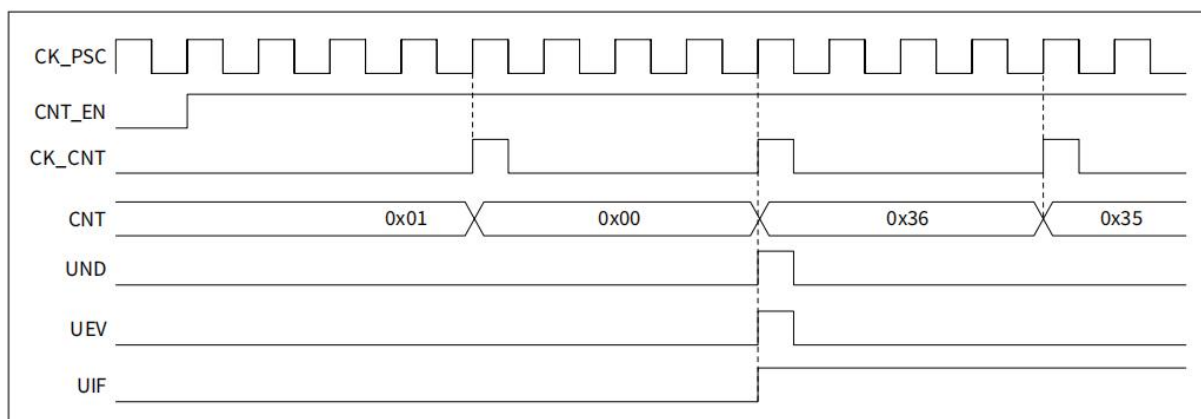


图 13-11 递减计数，内部时钟分频因子为 4

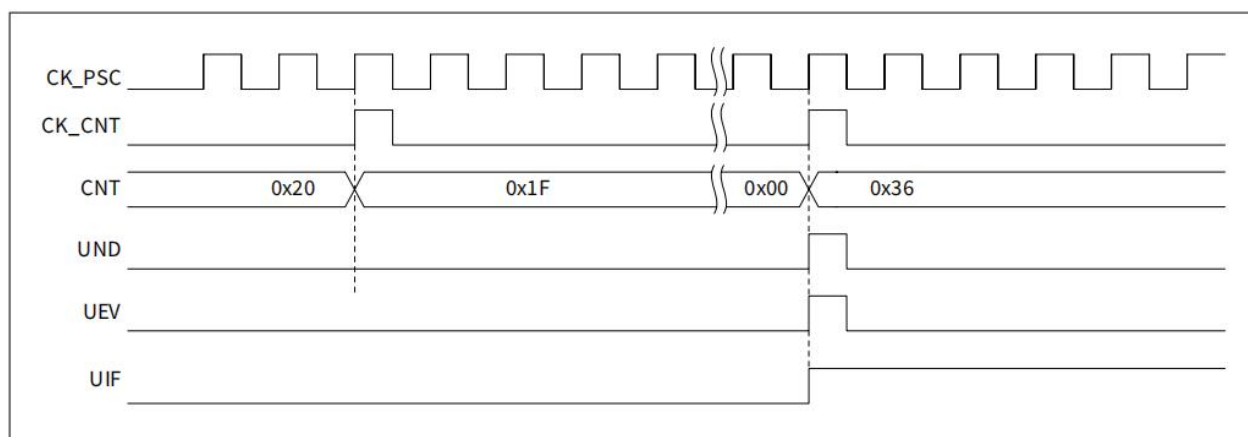


图 13-12 递减计数，内部时钟分频因子为 N

### 中心对齐模式（递增 / 递减计数）

在中心对齐模式下，计数器从 0 开始递增计数到 ARR-1，产生一个计数器上溢出事件，然后从 ARR 值开始递减计数到 1 并产生一个计数器下溢出事件，之后重新从 0 开始递增计数。

中心对齐模式包括中心对齐模式 1、中心对齐模式 2 和中心对齐模式 3，三种模式在计数方式上完全相同，只是输出比较中断标志的置位时机不同，具体请参见 1.9.1 ATIM\_CR1 控制寄存器 1 的 CMS 位域说明。

在中心对齐模式下，控制寄存器 ATIM\_CR1 的 DIR 位不能由软件写入，但可以读出，DIR 由硬件更新并指示当前的计数方向。启动中心对齐模式时，计数器将根据当前的 DIR 位域值进行递增或递减计数。不能同时通过软件修改 DIR 和 CMS 位域。

不建议在运行中心对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤其是：

- 如果在计数器 CNT 中写入大于自动重载值 ARR 的值 (CNT>ARR)，则不会更新方向。例如，如果计数器之前是递增计数，则继续递增计数。
- 如果向计数器 CNT 写入 0 或 ARR 值，计数方向会更新，但不生成更新事件 UEV。

使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新事件 UEV（设置 ATIM\_EGR.UG 为 1），并且不要在计数器运行过程中对其执行写操作。

以下是计数器在不同时钟频率下的操作示例，未使用重复计数器。

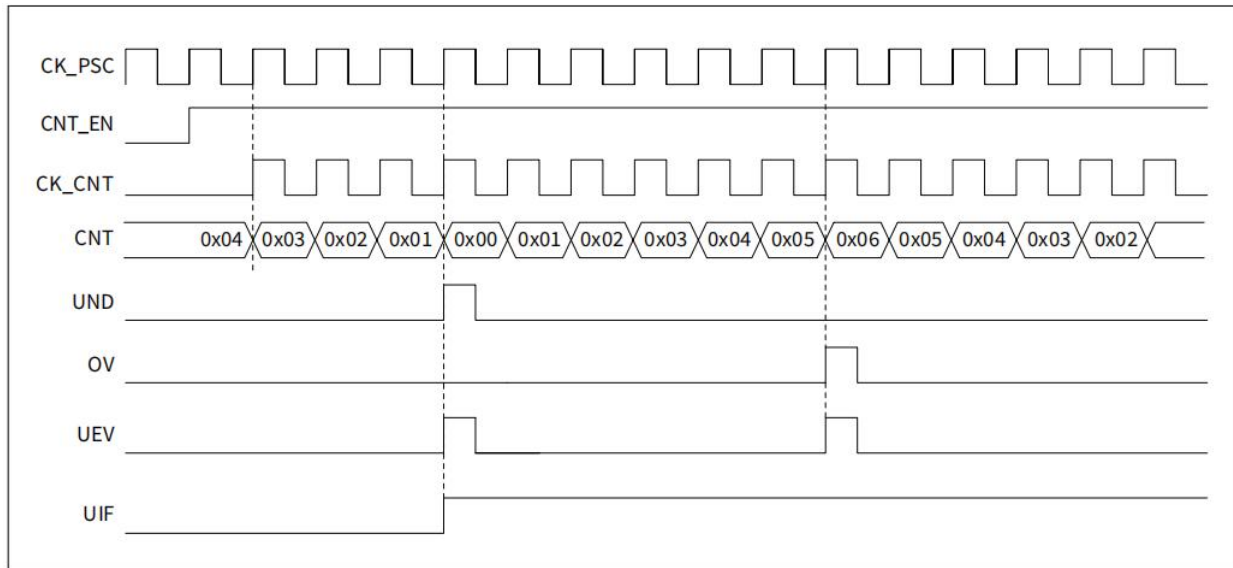


图 13-13 中心对齐模式，内部时钟分频因子为 1，ARR=0x06

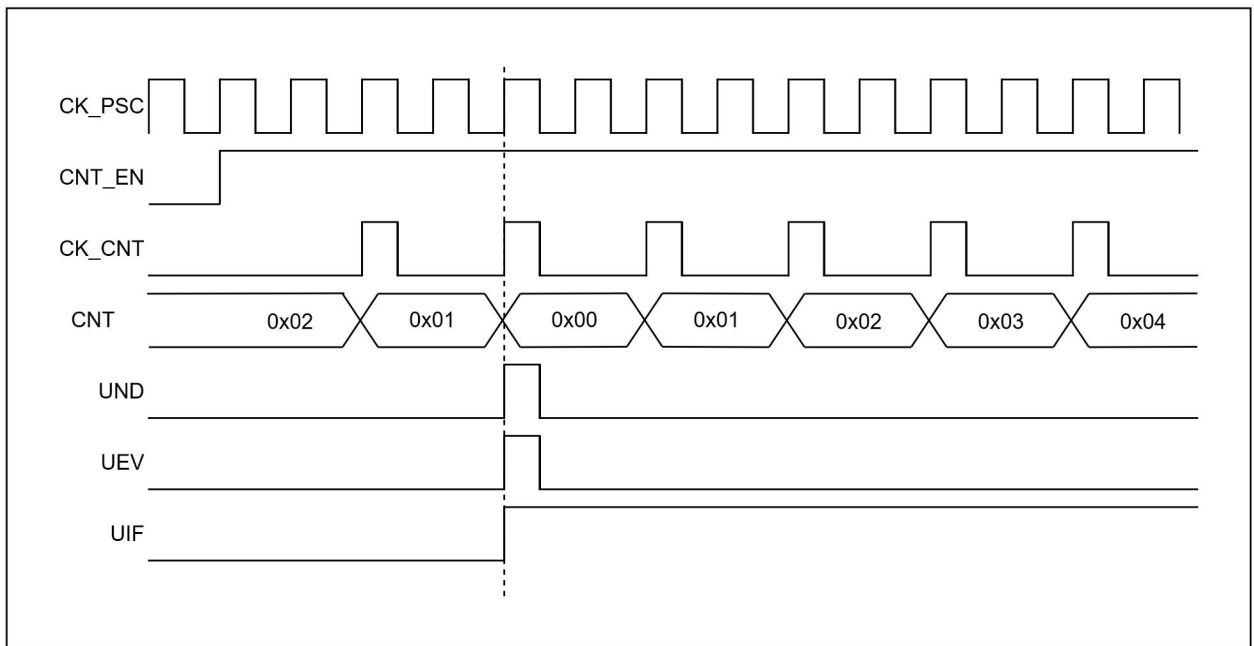


图 13-14 中心对齐模式，内部时钟分频因子为 2

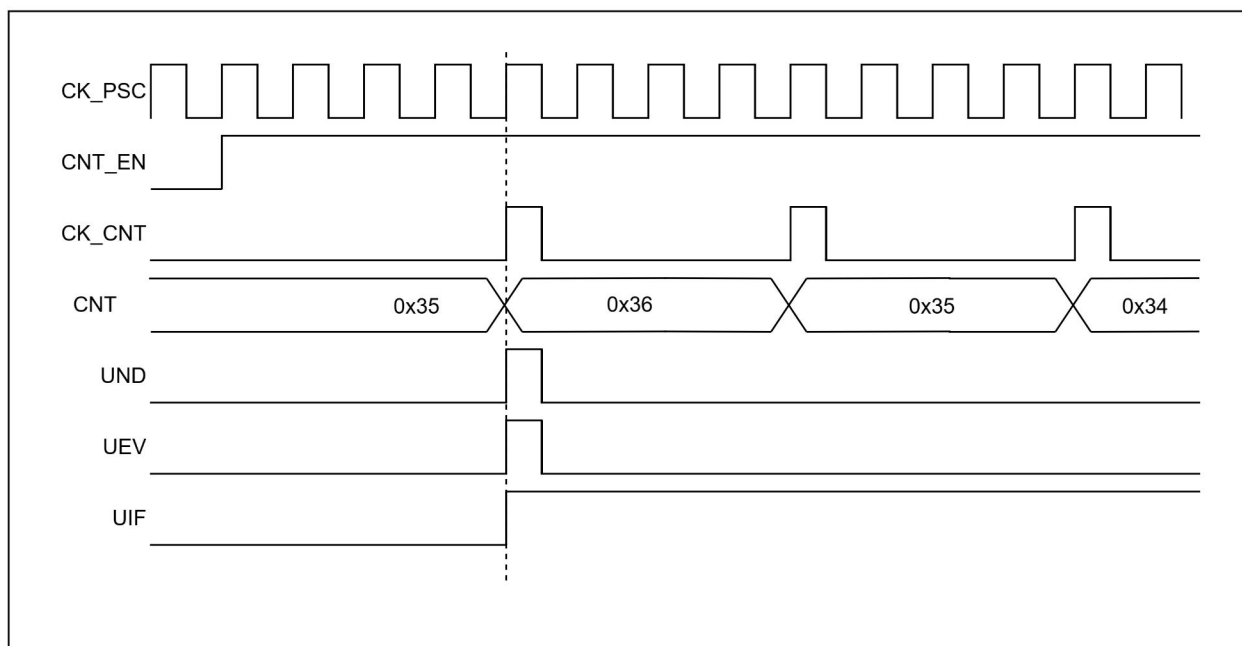


图 13-15 中心对齐模式，内部时钟分频因子为 4，ARR=0x36

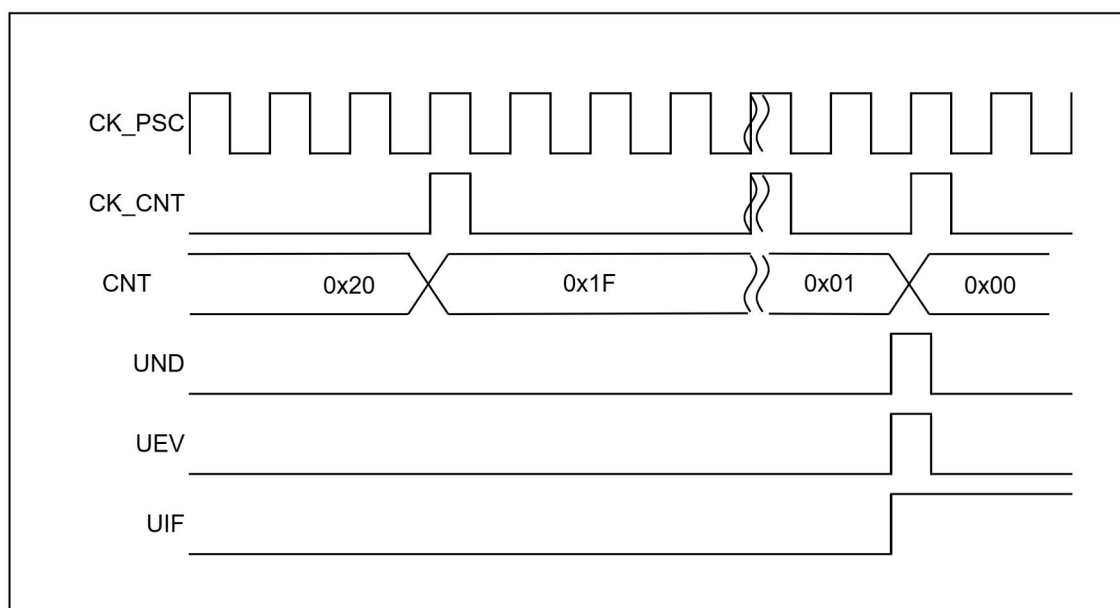


图 13-16 中心对齐模式，内部时钟分频因子为 N

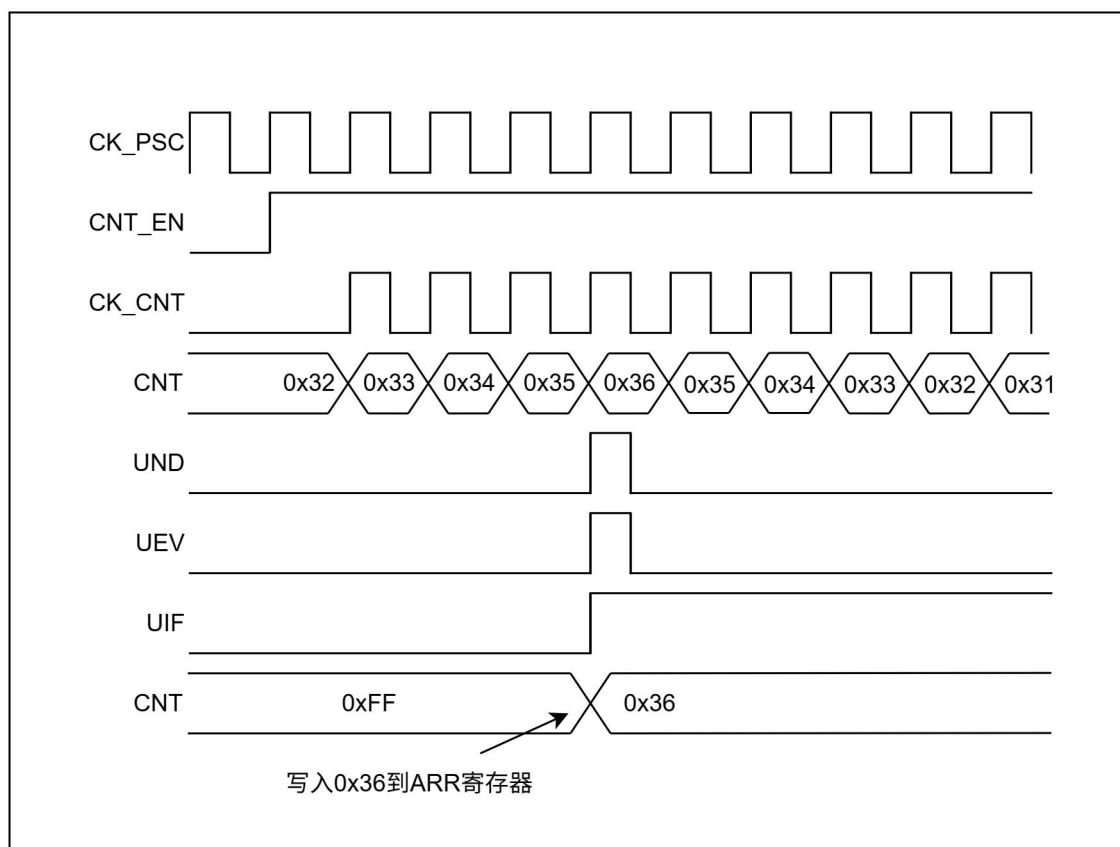


图 13-17 中心对齐模式，当重载缓存禁止时的更新事件（ATIM\_CR1.ARPE=0）



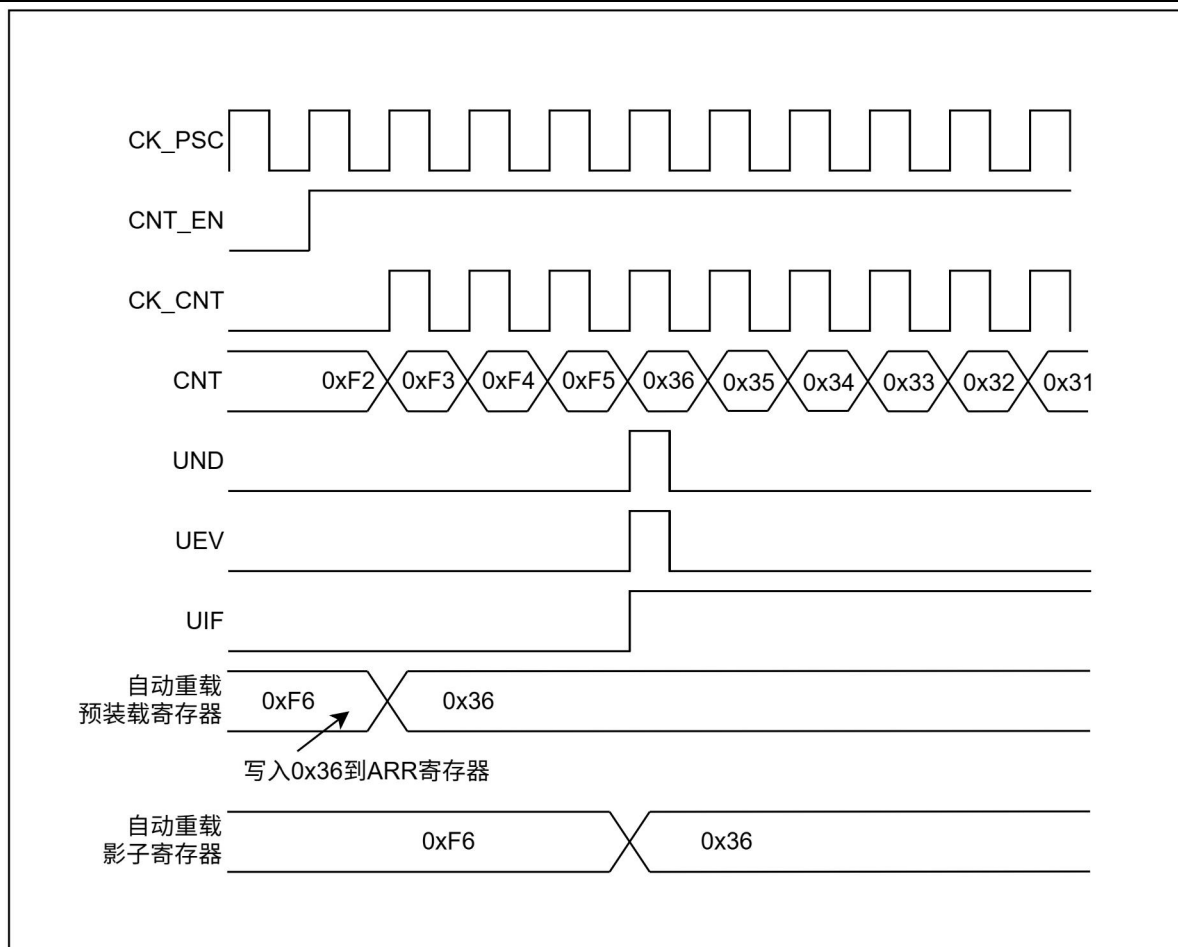


图 13-18 中心对齐模式，当重载缓使能时的更新事件 (ATIM\_CR1LARPE=1)

### 13.3.5 重载寄存器

自动重载寄存器 ATIM\_ARR 具有缓存功能，通过控制寄存器 ATIM\_CR1 的 ARPE 位域开启或关闭。

当计数器处于停止状态或是缓存功能关闭时，更新重载寄存器 ARR 将会立即更新其影子寄存器。当定时器处于运行状态且缓存功能有效时，修改重载寄存器 ARR 将不会立即更新影子寄存器，仅当生成更新事件 UEV 时才会将重载寄存器 ARR 的值更新到影子寄存器中。

### 13.3.6 重复计数器

启动 ATIM 会自动加载重复计数寄存器 ATIM\_RCR 的 REP 值到重复计数器，在主计数器产生溢出时自动减 1。

使用重复计数器功能（ATIM\_RCR 的 REP 设置不为 0）时，只有当重复计数器减为 0 时，主计数器的溢出才会触发更新事件 UEV。

通过软件更新 UG 或从模式复位时，会立即产生更新事件 UEV，不关心当前重复计数器的值，并且，ATIM\_RCR 寄存器中 REP 的内容将立即加载到重复计数器。

在中心对齐模式下，如果重复值 REP 为奇数，根据写入 ATIM\_RCR 寄存器以及启动计数器的时机，更新事件将在上溢或下溢时发生：如果在启动计数器前写入 RCR 寄存器，则更新事件 UEV 在上溢时发生；如果在启动计数器后写入 RCR 寄存器，则更新事件 UEV 在下溢时发生。

下图是不同重复计数值，在不同计数模式下产生更新事件 UEV 的示例：

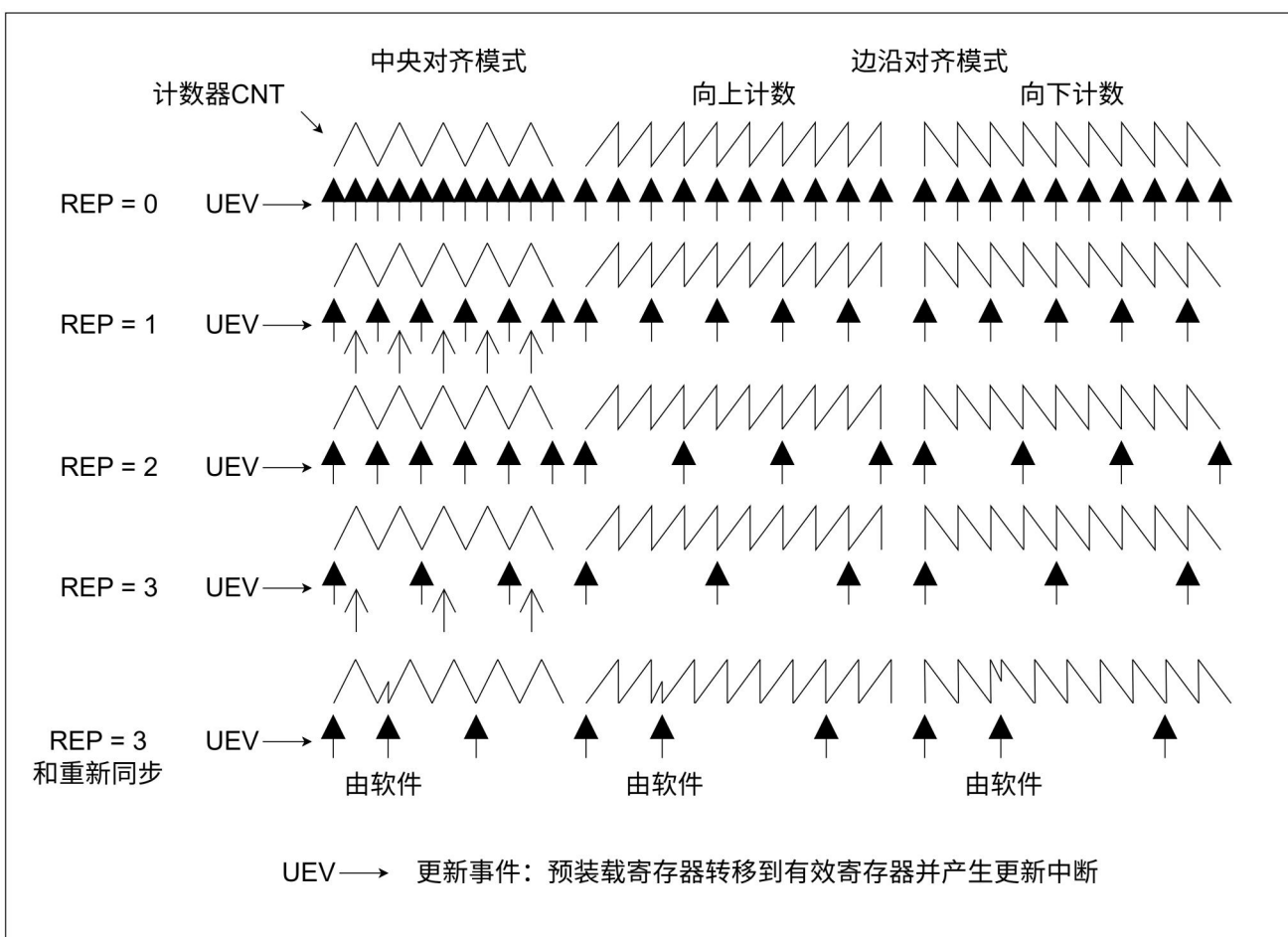


图 13-19 重复计数示例

### 13.3.7更新事件 UEV

允许通过控制寄存器 ATIM\_CR1 的 UDIS 位域来禁止或使能更新。使能 UEV 设置 UDIS 位域为 0 使能 UEV，此时根据 URS 位域可选择更新请求源，如下表所示。

表 13-2 更新源设置

ATIM_CR1.URS	更新源
0	计数器上溢出、下溢出； UG 置位； 通过从模式控制器生成的更新事件
1	计数器上溢出、下溢出

当发生更新事件时，将进行以下动作：

1. 重新初始化计数器：

递减计数（边沿对齐模式）：重新加载自动重载值 ATIM\_ARR； 中心对齐模式或递增计数（边沿对齐模式）：计数器清零。

2. 如果使能了重复计数器，重复计数器中将重新装载 ATIM\_RCR 寄存器的内容。

3. 如果使用了缓存寄存器功能，将更新对应的预装载寄存器到其影子寄存器。

4. 预分频器的计数器被清零，ATIM\_PSC 中新的预分频值生效。

5. 事件更新中断标志位 ATIM\_ISR.UIF 被硬件置位。

当发生一个更新事件 UEV 时，事件更新中断标志位 ATIM\_ISR.UIF 会被硬件置位，如果允许中断（设置 ATIM\_IER.UIE 为 1），将产生一个更新中断请求，设置 ATIM\_ICR.UIF 为 0 可清除该标志位。

注意：

如果使用了重复计数器(ATIM\_RCR.REP 不为 0)，只有在计数次数达到重复计数次数(ATIM\_RCR.REP 达到 0) 时，才会产生更新事件 UEV。

#### 禁止 UEV

设置 UDIS 位域为 1 禁止 UEV，不再生成任何更新事件。

如果使用了缓存寄存器功能，对应的各影子寄存器的值保持不变。但如果 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器的计数器。

### 13.3.8单脉冲模式

计数器可工作在单次计数或连续计数模式下，通过控制寄存器 ATIM\_CR1 的 OPM 位域来选择。

#### 单次计数模式

设置 ATIM\_CR1.OPM 为 1，使定时器工作在单次计数模式下。

启动计数器后，计数器 CNT 在 CK\_CNT 时钟的驱动下计数，在发生下一更新事件时，计数器停止计数，同时 ATIM\_CR1.CEN 被硬件自动清零。

可以通过从模式控制器启动计数器，配合定时器的输出比较模式，使得计数器在一个触发信号的触发下启动，并 在一段可编程的延时后产生一个脉宽可编程的单脉冲。如下图所示，定时器在检测到触发信号的有效边沿时开始 计数，延迟 tDELAY 之后，在比较输出端口上产生一个宽度为 tPULSE 的正脉冲。其中 tDELAY 由计数寄存器 ATIM\_CNT 的初值和捕获 / 比较寄存器 ATIM\_CCRy 的差值确定，tPULSE 由自动重载寄存器 ATIM\_ARR 和捕获 / 比较寄存器 ATIM\_CCRy 的差值来确定。

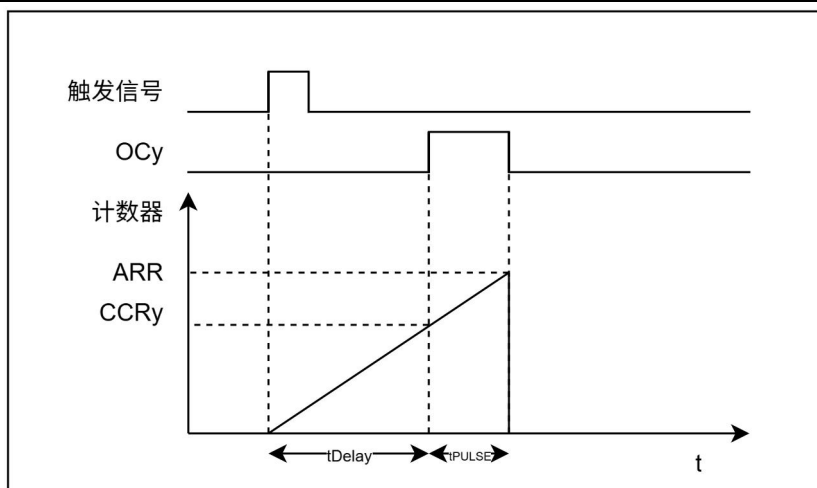


图 13-20 单脉冲输出示例（递增计数模式）

采样触发输入到激活 OC 输出需要多个时钟周期，可设置比较模式寄存器 ATIM\_CCMRxCMP 的 OCyFE 位域为 1，以缩短 OC 输出延迟时间。

#### 连续计数模式

设置 ATIM\_CR1.OPM 为 0，使定时器工作在连续计数模式下，计数器在发生更新事件时不会停止计数。

### 13.3.9 外部触发输入通道

外部触发输入信号 ETR 可用作从模式控制器的触发输入 (TRGI)，也可用作计数器的计数时钟，具体请参见 1.3.2 工作模式。可对 ETR 信号进行输入控制，包括极性选择和边沿检测、预分频和滤波。

#### 极性选择和边沿检测

从模式控制寄存器 ATIM\_SMCR 的 ETP 位域用于选择 ETR 输入信号的触发极性。当设置 ETP 为 0 时，ETR 不反相，高电平或上升沿有效；当 ETP 设置为 1 时，ETR 反相，低电平或下降沿有效。

#### 预分频器

外部触发信号 ETRP 频率不得超过 PCLK 频率的 1/4。当 ETRP 的频率较高时，用户应通过适当的 ETPS 预分频器设置对外部信号进行分频，以降低 ETRP 频率，可设置分频系数为 1、2、4、8。

#### 滤波器

滤波器采用数字滤波方式，以一定频率对输入信号进行采样，当连续采样到 N 个相同电平时信号有效，否则信号无效，以此滤除高频杂波信号。

ETR 输入信号的滤波器由从模式控制寄存器 ATIM\_SMCR 的 ETF 位域进行配置，可设置采样频率和采样点的个数。采样频率 fSAMPLING 由 fDTS 分频后的时钟提供，fDTS 是 fPCLK 分频后得到的频率，分频因子由 ATIM\_CR1 寄存器的 CKD 位域配置，可设置 1、2、4 分频。

### 13.3.10 输入捕获通道

ATIM 支持 6 个输入捕获通道 TI1/2/3/4/5/6，可用作捕获命令。其中 TI1 和 TI2 还可用作从模式控制器的触发输入 (TRGI) 和编码器接口输入，具体请参见 1.3.2 工作模式。

支持对 TIy 信号进行输入控制，包括滤波、边沿检测和预分频。

#### 滤波器

滤波阶段以一定频率对相应的 TIy 输入信号进行采样，生成滤波后信号 TIyF。

TIy 输入信号的滤波器由捕获模式寄存器 ATIM\_CCMRxCAP 的 ICyF 位域进行配置，可设置采样频率和采样点的个数。采样频率 fSAMPLING 由 fPCLK 或 fDTS 分频后的时钟提供，fDTS 是

fPCLK 分频后得到的频率，分频因子由 ATIM\_CR1 寄存器的 CKD 位域配置，可设置 1、2、4 分频。

### 边沿检测

带有极性选择功能的边沿检测器可生成一个 TIXFPx 信号，具体有效极性请参见 14.9.14 ATIM\_CCER 捕获 / 比较 使能寄存器的 CCyNP 和 CCyP 位域。

### 预分频器

TIy 信号用于输入捕获时，可对捕获通道信号 ICy 进行分频，通过捕获模式寄存器 ATIM\_CCMRxCAP 的 ICyPSC 位域进行控制，支持 1、2、4、8 分频。

## 13.3.11 输出比较通道

ATIM 支持 6 个独立输出比较通道 CH1/2/3/4/5/6（所有通道均支持互补输出 CHyN），每个通道均由一个捕获 / 比较寄存器（包括一个影子寄存器）和一个输出阶段（比较器和输出控制）组成。

输出控制单元用于比较匹配时，控制输出端口的波形，可配置输出比较模式、输出极性选择和输出使能等。以比较通道 1 为例，其框图如下图所示：

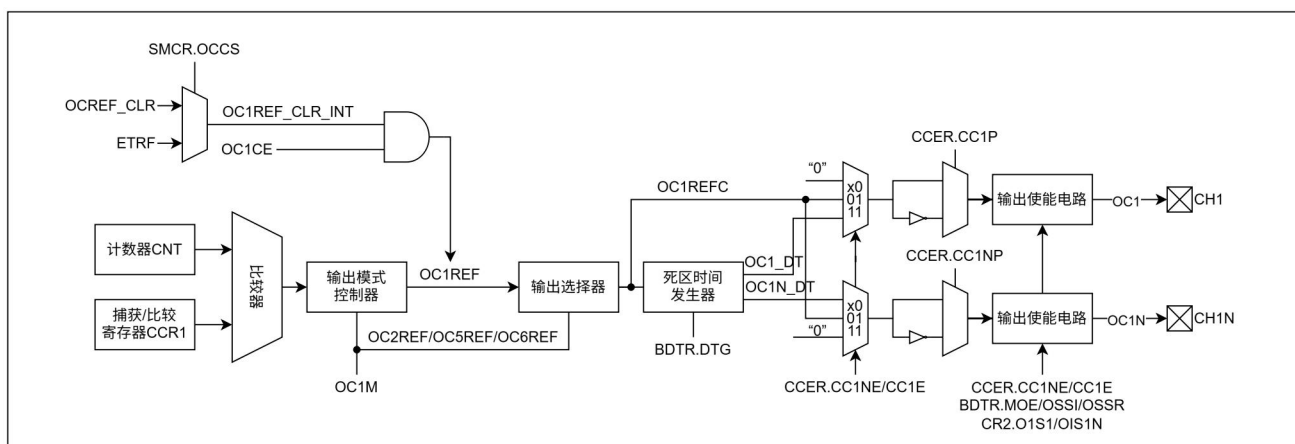


图 13-21 输出比较 1 通道

### 13.3.12 工作模式

ATIM 支持多种工作模式，具体由从模式控制寄存器 ATIM\_SMCR 的 SMS 位域来配置，如表 1-3 所示：

表 13-3 ATIM 工作模式

ATIM_SMCR.SMS	从模式选择
0	禁止从模式，使用内部时钟
111	外部时钟模式 1
100	复位模式
101	门控模式
110	触发模式
1000	组合复位+触发模式
1001	组合门控+复位模式
1110	正交编码器模式，x1 模式
1111	正交编码器模式，x1 模式
1	正交编码器模式，x2 模式
10	正交编码器模式，x2 模式
11	正交编码器模式，x4 模式
1010	编码模式（时钟+方向），x2 模式
1011	编码模式（时钟+方向），x1 模式
1100	编码模式（带方向时钟），x2 模式
1101	编码模式（带方向时钟），x1 模式

注 1：

从模式选择位包括 ATIM\_SMCR[16] 和 ATIM\_SMCR[2:0]，需组合配置。

注 2：

从模式选择位 SMS[3:0] 支持预装载功能，请参见 SMSPE 和 SMSPE 位域说明。

#### 13.3.12.1 内部时钟模式

当从模式控制寄存器 ATIM\_SMCR 的 SMS 位域为 0x0 时，禁止定时器从模式，预分频器时钟直接由内部时钟 PCLK 提供。设置 ATIM\_CR1.CEN 为 1，将使能计数器开始计数。

### 13.3.12.2 外部时钟模式

#### 外部时钟模式 1

当从模式控制寄存器 ATIM\_SMCR 的 SMS 位域为 0x7 时，由所选触发信号 (TRGI) 的上升沿提供计数器时钟。TRGI 信号有多种触发选择，具体通过 ATIM\_SMCR 寄存器的 TS 位域进行选择，如下表所示：

表 13-4 TRGI 信号来源

ATIM_SMCR.TS 位域值	TRGI 信号来源
111	外部触发输入(ETRF)
000xx/01xxx/10000	内部触发 (ITR0~12)
100	TI1 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)

ETR 输入信号来源可以是外部 ATIM\_ETR/GTIM\_ETR 引脚，也可以是片内其它外设，请参见 14.3.7 片内外设互联 ETR。选择 ATIM\_ETR 为 TRGI 信号源时，可通过 ATIM\_SMCR.ETP 选择外部触发极性，通过 ATIM\_SMCR.ETPS 设置预分频，通过 ATIM\_SMCR.ETF 进行滤波控制。

内部触发 ITR 来源为 BTIM 和 GTIM 的触发输出信号 TRGO 以及 UART 的 TXD/RXD 信号，请参见 14.3.6 定时器级联 ITR。

TI1 和 TI2 都具有滤波和边沿检测功能，TI1 和 TI2 分别通过捕获模式寄存器 ATIM\_CCMR1CAP 的 IC1F 和 IC2F 位域进行滤波控制，分别通过捕获 / 比较使能寄存器 ATIM\_CCER 的 CC1P/CC1NP 和 CC2P/CC2NP 位域进行边沿检测。

外部时钟模式 1 连接示意图如下图所示：

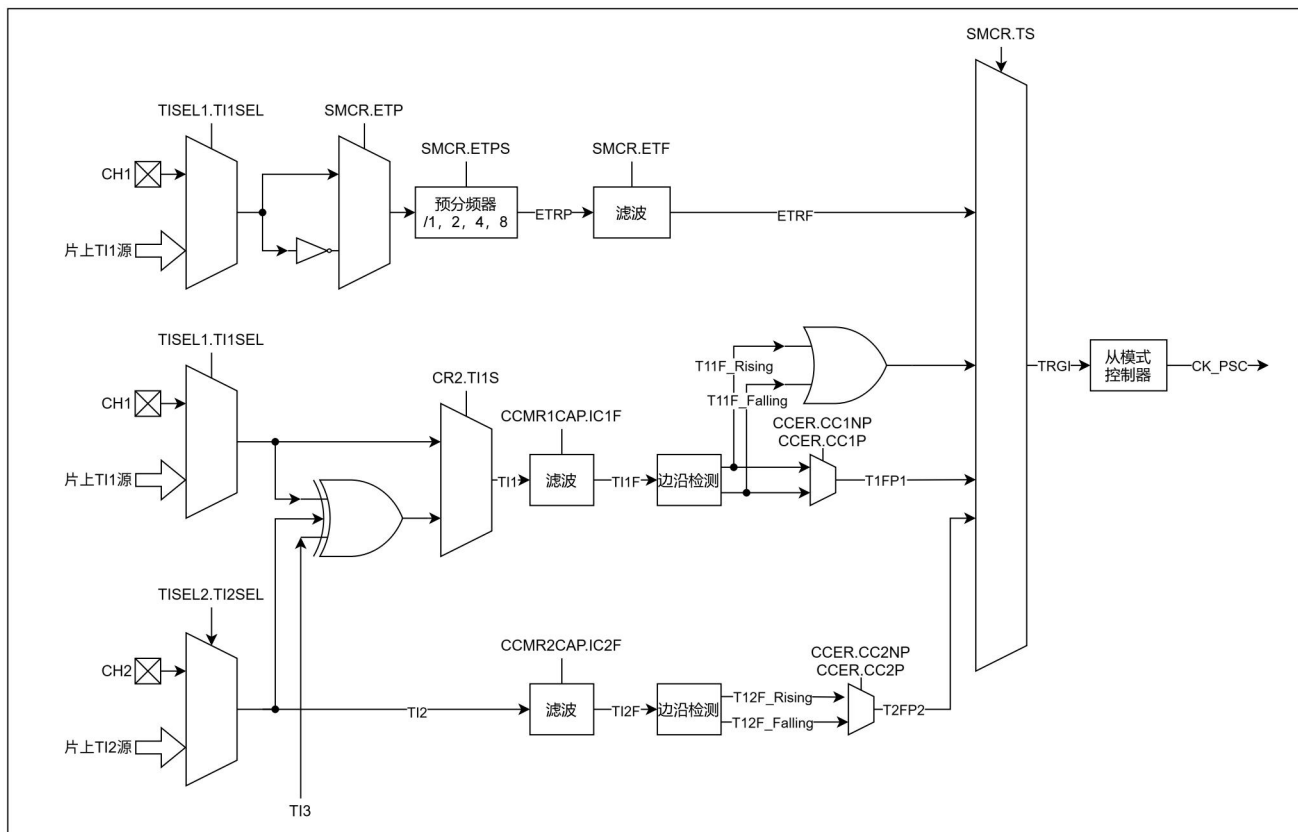


图 13-22 外部时钟模式 1 连接图



当 TRGI 出现有效边沿时，ATIM\_ISR.TIF 标志将置 1，向 ATIM\_ICR.TIF 写 0 可清除该标志。

下图所示为配置 TI1 上升沿的外部时钟模式 1 时序图，TI1 的上升沿与实际计数器时钟之间的延迟是由于 TI1 输入的重新同步电路引起的。

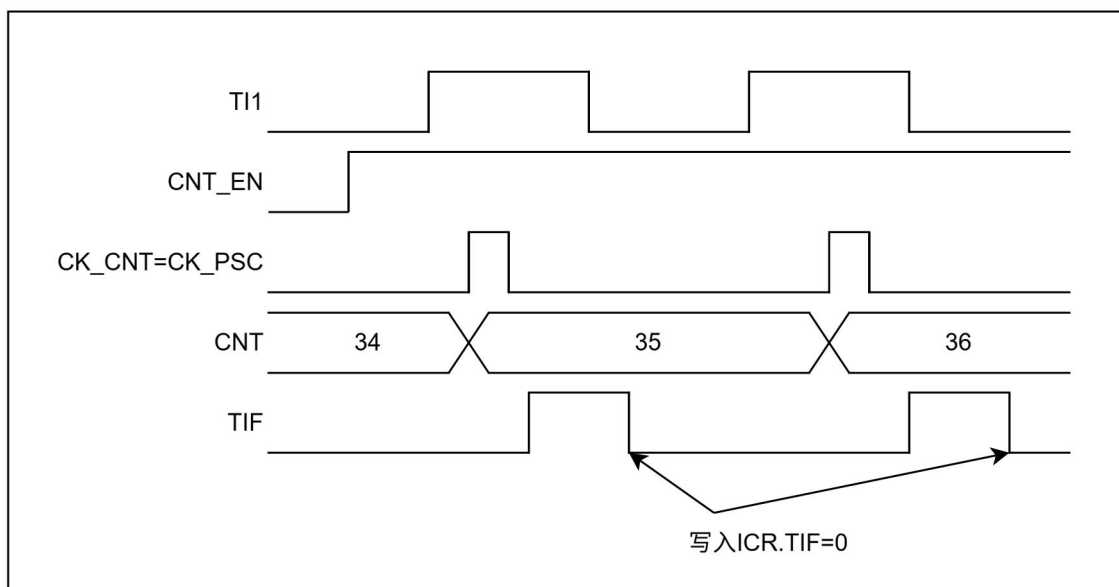


图 13-23 外部时钟模式 1 时序示例

## 外部时钟模式 2

设置从模式控制寄存器 ATIM\_SMCR 的 ECE 位域为 1，使能外部时钟模式 2。该模式下，计数器时钟由 ETRF 信号的任意有效边沿提供，与选择外部时钟模式 1 并将 TRGI 连接到 ETRF (SMS=0111 且 TS=00111) 具有相同效果，连接框图可参见图 14-22 外部时钟模式 1 连接图。

外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式，此时从模式下的 TRGI 不得连接 ETRF (即 TS 位域不能设置为 0x7)。如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 ETRF。外部时钟模式 2 不能与编码器模式同时使用。

下图所示为 ETR 设置为 2 分频时的外部时钟模式 2 的时序示例，ETR 的上升沿与实际计数器时钟之间的延迟是由 ETRP 信号的重新同步电路引起的，因此计数器可以正确捕获的最大频率最高为内部时钟 PCLK 频率的 1/4，当 ETRP 信号变快时，用户应对外部信号进行适当的分频。



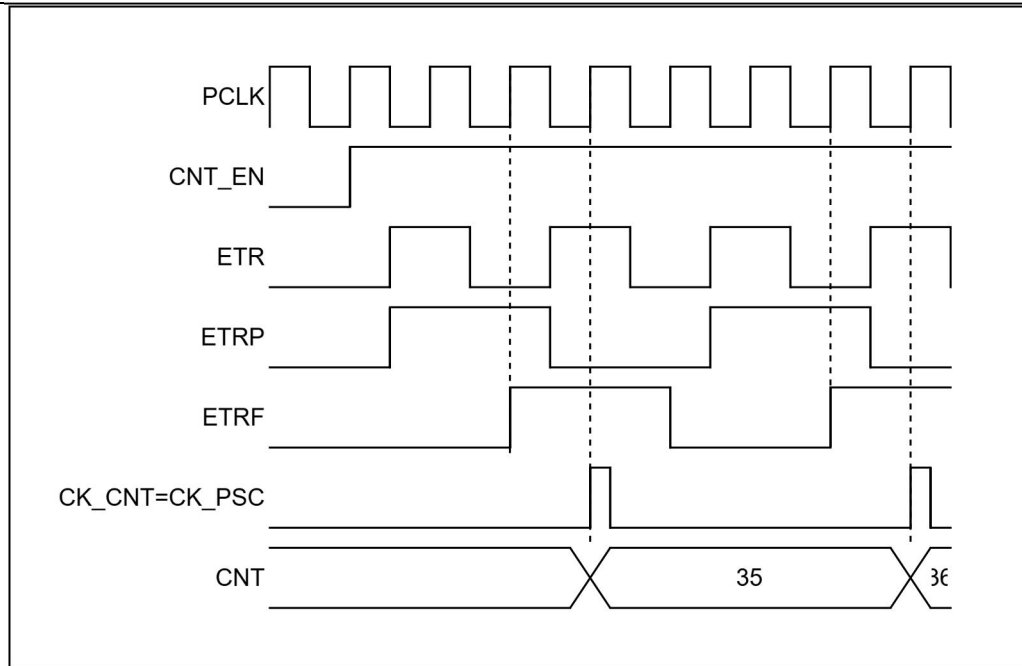


图 13-24 外部时钟模式 2 时序示例

### 13.3.12.3 复位模式

当从模式控制寄存器 ATIM\_SMCR 的 SMS 位域为 0x4 时，ATIM 配置为复位模式，计数器的复位由 TRGI 信号控制。TRGI 信号的来源由从模式控制寄存器 ATIM\_SMCR 的 TS 位域控制，可参见表 13-4 TRGI 信号来源。

当检测到有效的触发输入（TRGI）时，将产生以下影响：

1. 重新初始化计数器和预分频器的计数器；
2. 如果控制寄存器 ATIM\_CR1 的 URS 位域为 0，则会产生更新事件 UEV，事件更新中断标志 ATIM\_ISR.UIF 置 1，可产生中断请求；
3. 触发中断标志 ATIM\_ISR.TIF 置 1，可产生中断请求。

下图所示为复位模式时序图示例。设置 ATIM\_CR1.CEN 为 1 使能计数器，计数器根据计数时钟 CK\_CNT 正常计数，当 TI1 出现上升沿时，计数器清零并重新从 0 开始计数，同时 ATIM\_ISR.TIF 标志位置 1。TI1 的上升沿与实际计数器复位之间的延迟是由 TI1 输入的重新同步电路引起的。

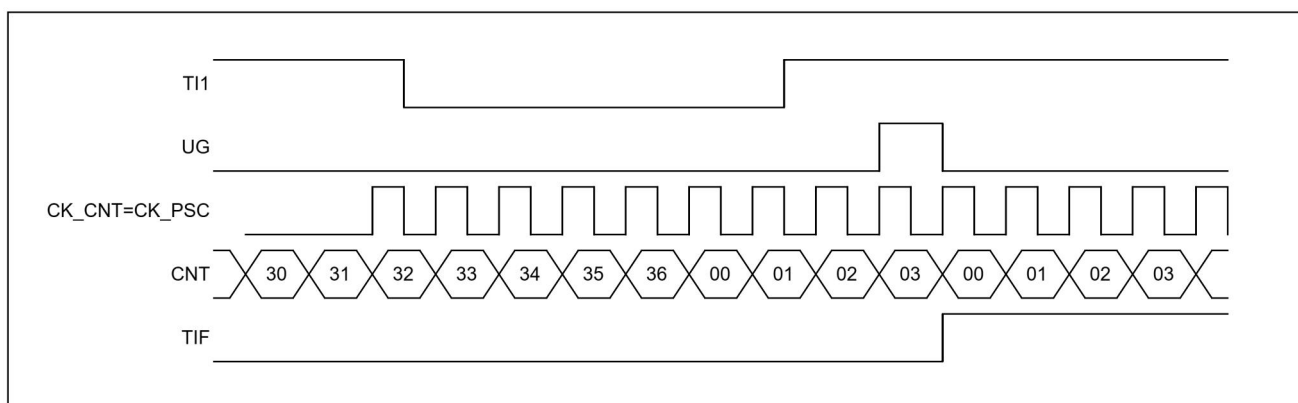


图 13-25 复位模式时序

### 13.3.12.4 门控模式

当从模式控制寄存器 ATIM\_SMCR 的 SMS 位域为 0x5 时，ATIM 配置为门控模式。在该模式下，触发输入 (TRGI) 为高电平且 ATIM\_CR1.CEN 为 1 时，启动计数器计数；触发输入 (TRGI) 为低电平或 ATIM\_CR1.CEN 为 0 时，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。

TRGI 信号的来源由从模式控制寄存器 ATIM\_SMCR 的 TS 位域控制，可参见表 14-4 TRGI 信号来源，但需注意门控模式下不能选择 TI1F\_ED 作为触发输入。

下图所示为门控模式时序图示例。当 TI1 为高电平时，计数器启动计数；当 TI1 为低电平时，计数器暂停计数。计数器启动和停止时，ATIM\_ISR.TIF 标志位都会置 1。TI1 的边沿与实际计数器使能信号 CNT\_EN 之间的延迟是由 TI1 输入的重新同步电路引起的。

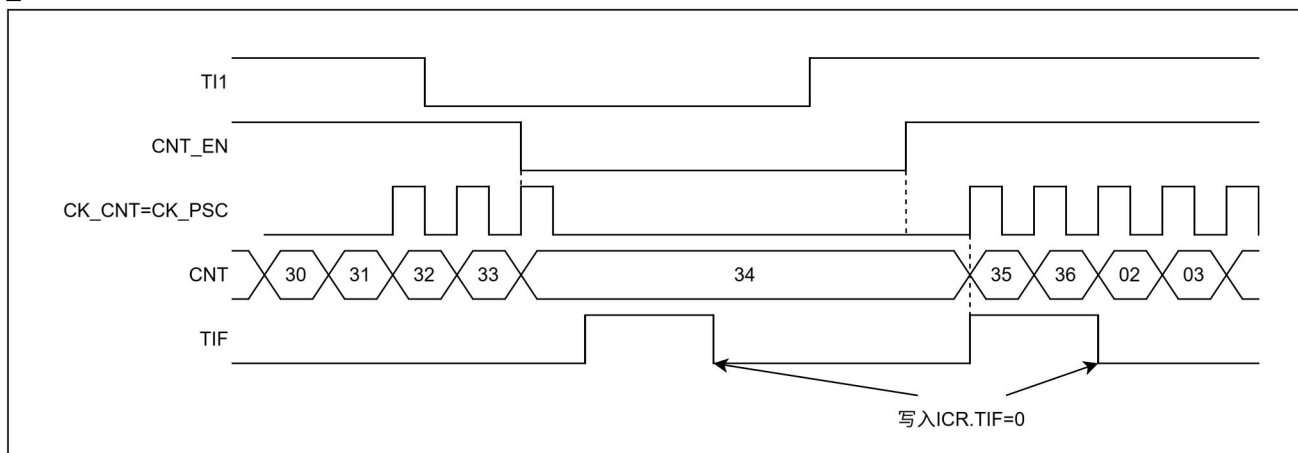


图 13-26 门控模式时序

### 13.3.12.5 触发模式

当从模式控制寄存器 ATIM\_SMCR 的 SMS 位域为 0x6 时，ATIM 配置为触发模式。在该模式下，设置 ATIM\_CR1.CEN 为 1 或触发信号 TRGI 出现上升沿时，触发启动计数器计数（但不复位）。

TRGI 信号的来源由从模式控制寄存器 ATIM\_SMCR 的 TS 位域控制，可参见表 1-4 TRGI 信号来源。当检测到有效的触发信号时，将产生以下影响：

1. ATIM\_CR1.CEN 被硬件置位；
2. 触发中断标志位 ATIM\_ISR.TIF 置 1，可产生中断请求；
3. 计数器启动，开始计数。

下图所示为触发模式时序图示例。当 TI1 出现上升沿时，计数器启动计数，同时 ATIM\_ISR.TIF 标志位置 1。TI1 的上升沿与实际计数器启动之间的延迟是由 TI1 输入的重新同步电路引起的。

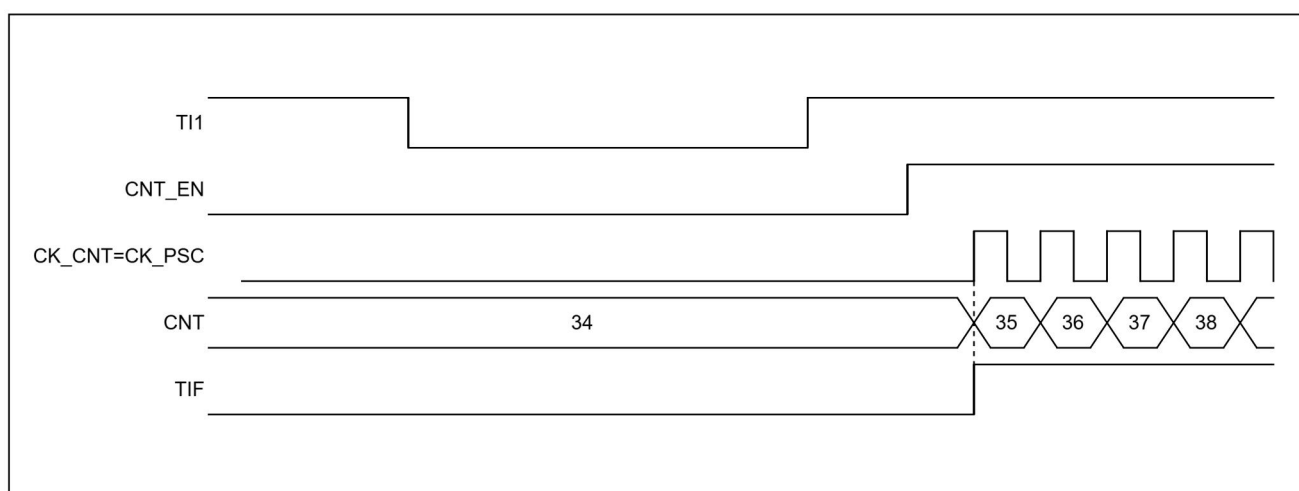


图 13-27 触发模式时序

### 13.3.12.6 组合复位+触发模式

当从模式控制寄存器 ATIM\_SMCR 的 SMS 位域为 0x8 时，ATIM 配置为组合复位 + 触发模式。在该模式下，出现所选触发输入 (TRGI) 上升沿时，将重新初始化计数器并启动计数器，同时触发中断标志 ATIM\_ISR.TIF 置 1。如果控制寄存器 ATIM\_CR1 的 URS 位域为 0，则会产生更新事件 UEV，事件更新中断标志 ATIM\_ISR.UIF 会被硬件置位。

TRGI 信号的来源由从模式控制寄存器 ATIM\_SMCR 的 TS 位域控制，可参见表 1-4 TRGI 信号来源。

### 13.3.12.7 组合门控+复位模式

当从模式控制寄存器 ATIM\_SMCR 的 SMS 位域为 0x9 时，ATIM 配置为组合门控 + 复位模式。在该模式下，计数器的启动和停止都被控制，触发输入 (TRGI) 为高电平且 ATIM\_CR1.CEN 为 1 时，启动计数器计数；触发输入 (TRGI) 为低电平时，计数器立即停止计数，并被复位。计数器启动和停止时，ATIM\_ISR.TIF 标志位都会置 1；计数器复位时，ATIM\_ISR.UIF 标志位置 1。

TRGI 信号的来源由从模式控制寄存器 ATIM\_SMCR 的 TS 位域控制，可参见表 1-4 TRGI 信号来源。

### 13.3.12.8 正交编码器模式

ATIM 支持正交编码器模式，用于接收并解码正交编码器的信号。在该模式下，允许通过 CH1、CH2 引脚与外部的正交编码器直接连接，根据输入信号的跳变顺序，实现计数器自动递增或递减计数，计数器值始终表示编码器 的位置。其功能框图如下图所示：

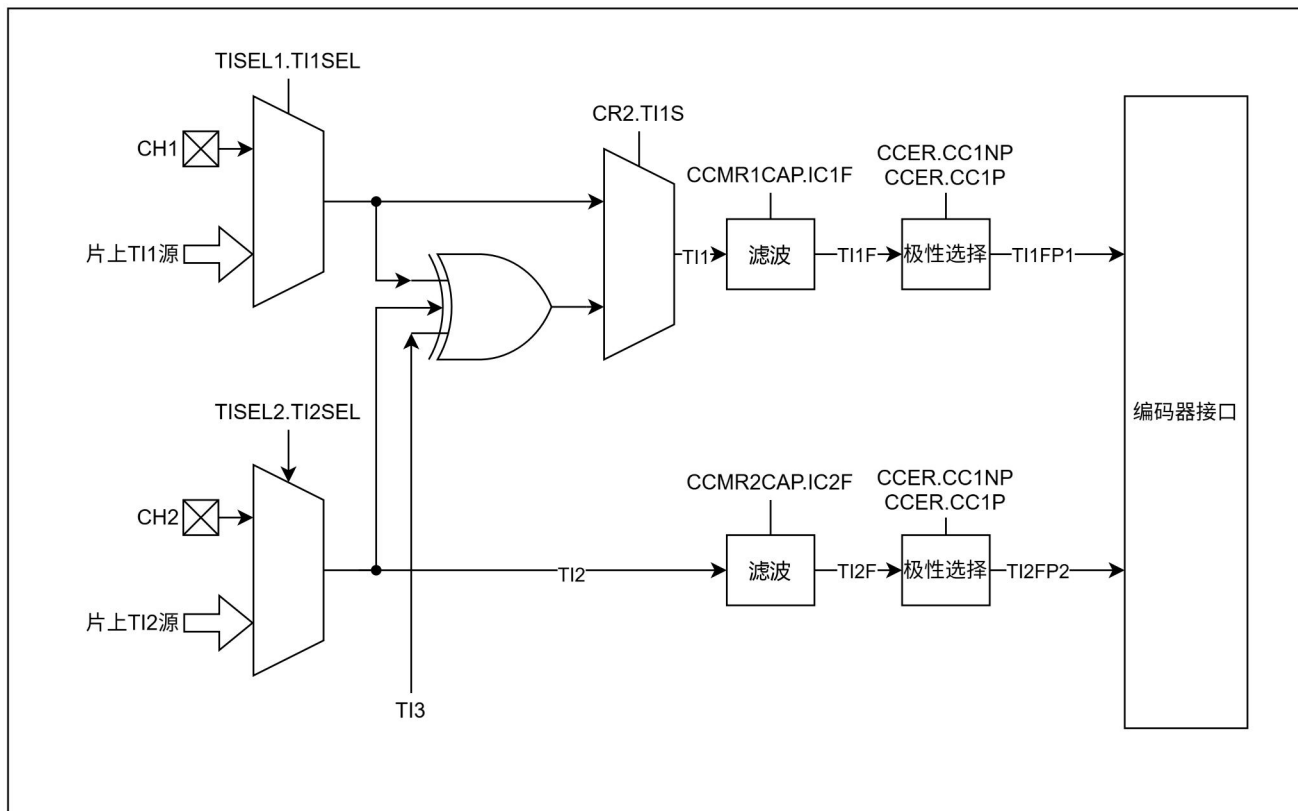


图 13-28 编码器模式框图

CH1 和 CH2 输入信号具有滤波和极性选择功能，分别通过 ATIM\_CCMR1CAP 寄存器的 IC1F 和 IC2F 位域进行滤波控制，通过 ATIM\_CCER 寄存器的 CC1P 和 CC2P 位域选择输入极性，CC1NP 和 CC2NP 位域必须保持清零。

设置 ATIM\_CR1.CEN 为 1 使能计数器，计数器将由通道 CH1 和 CH2 引脚输入信号经滤波和极性选择后的信号 TI1FP1 和 TI2FP2 的每次有效跳变驱动，根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号，参见表 14-5 计数方向和编码器信号的关系。编码器当前计数方向标志 ATIM\_CR1.DIR 由硬件自动设置和清除，且在任何输入（CH1 或 CH2）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 TI1FP1 或 TI2FP2 边沿处计数，还是同时在 TI1FP1 和 TI2FP2 处计数。

ATIM 支持多种正交编码计数模式，通过从模式控制寄存器 ATIM\_SMCR 的 SMS 位域进行设置，不同模式下计数 方向和编码器信号的关系如下表所示：

表 13-5 计数方向和编码器信号的关系

模式	SMS	信号的电平		TI1FP1		TI2FP2	
		TI2FP2	TI1FP1	上升	下降	上升	下降
在 TI1FP1 边沿计数 (x1 模式)	1110	高	-	向下计数	向上计数	不计数	不计数
		低	-	不计数	不计数	不计数	不计数
在 TI2FP2 边沿计数 (x1 模式)	1111	-	高	不计数	不计数	向上计数	向下计数
		-	低	不计数	不计数	不计数	不计数
在 TI1FP1 边沿计数	1	高	-	向下计数	向上计数	不计数	不计数

(x2 模式)		低	-	向上计数	向下计数	不计数	不计数
在 TI2FP2 边沿计数 (x2 模式)	10	-	高	不计数	不计数	向上计数	向下计数
		-	低	不计数	不计数	向下计数	向上计数
在 TI1FP1 和 TI2FP2 边沿计数 (x4 模式)	11	高	高	向下计数	向上计数	向上计数	向下计数
		低	低	向上计数	向下计数	向下计数	向上计数

编码器输出的第三个信号表示机械零点，可以把它连接到外部触发输入引脚上，用以触发计数器复位。

下图是一个正交编码计数模式 -x4 模式的操作实例，显示了计数信号的产生和方向控制。它还显示了当选择了双 边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。

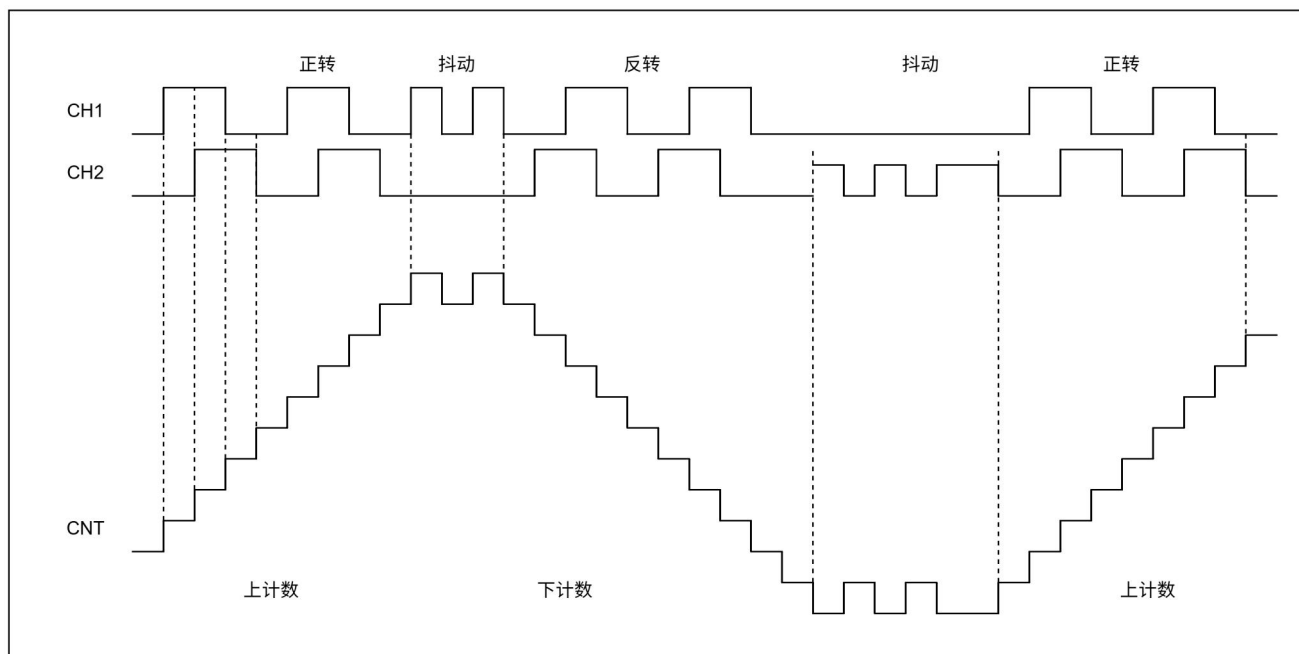


图 13-29 正交编码器模式 -x4 模式操作示例

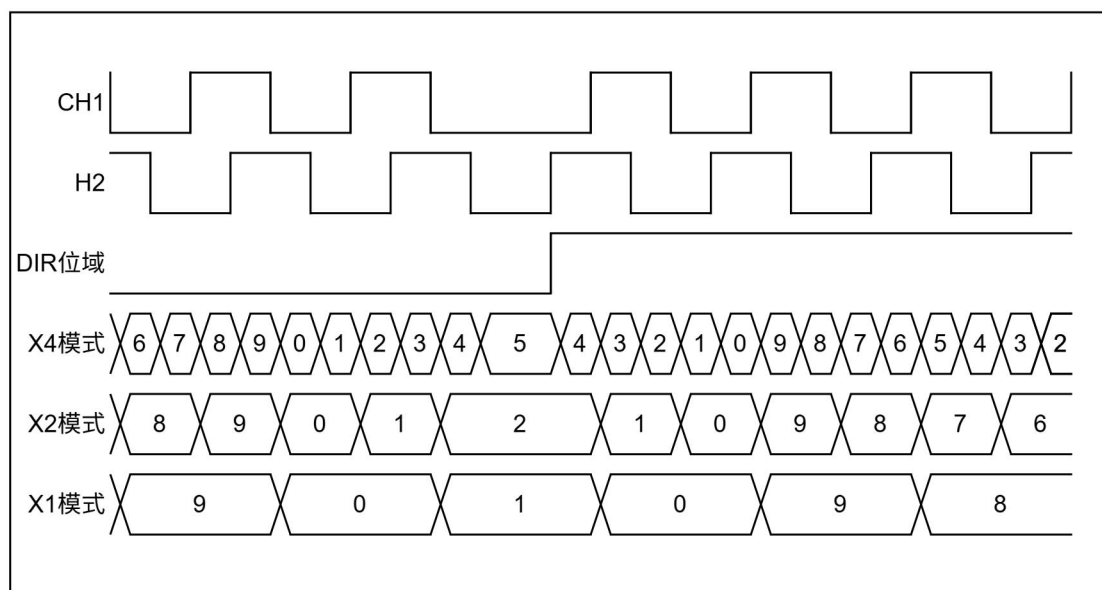


图 13-30 各正交编码器模式计数示例

编码器模式可以提供传感器当前位置的相关信息。计数器根据编码器输出的脉冲自动进行递增或递减计数，当前计数器值即表示编码器当前位置。使用另一个配置为捕获模式的定时器，可以捕获编码器信号的上升沿或下降沿，并记录其时间戳，通过测量两个事件之间的时间间隔，可以计算出相应的动态信息，如速度、加速度和减速度；可以使用指示机械零位的编码器输出来实现此目的，通过测量从机械零位到其他位置的时间间隔，可以计算出相对于零位的位置和速度变化。

### 13.3.12.9 时钟加方向编码器模式

定时器还支持时钟加方向的编码器模式，需配置从模式控制寄存器 ATIM\_SMCR 的 SMS 位域为 0xA 或 0xB，分别对应 x2 模式和 x1 模式。其功能框图可参见图 14-28 编码器模式框图，可对输入信号进行滤波和极性控制。

带方向的编码器通常具有两个输出信号：时钟信号和方向信号，使用时钟信号可以计算出旋转编码器的旋转速度和位置变化，而方向信号可以用来确定旋转的方向。在 x2 模式，计数器在时钟信号的上升沿和下降沿计数；在 x1 模式，计数器根据 CC2P 位域的值在单个时钟边沿计数，CC2P 为 0 对应上升沿敏感，CC2P 为 1 对应下降沿敏感。CH1 通道上方向信号的极性由 CC1P 位设置：CC1P 为 0 对应正极性（当 CH1 为高电平时递增计数，当 CH1 为低电平时递减计数），CC1P 为 1 对应负极性（当 CH1 为低电平时递增计数，当 CH1 为高电平时递减计数）。

下图显示了时钟加方向编码器模式下的计数器计数实例，示例中 CH1 和 CH2 输入均不反相。

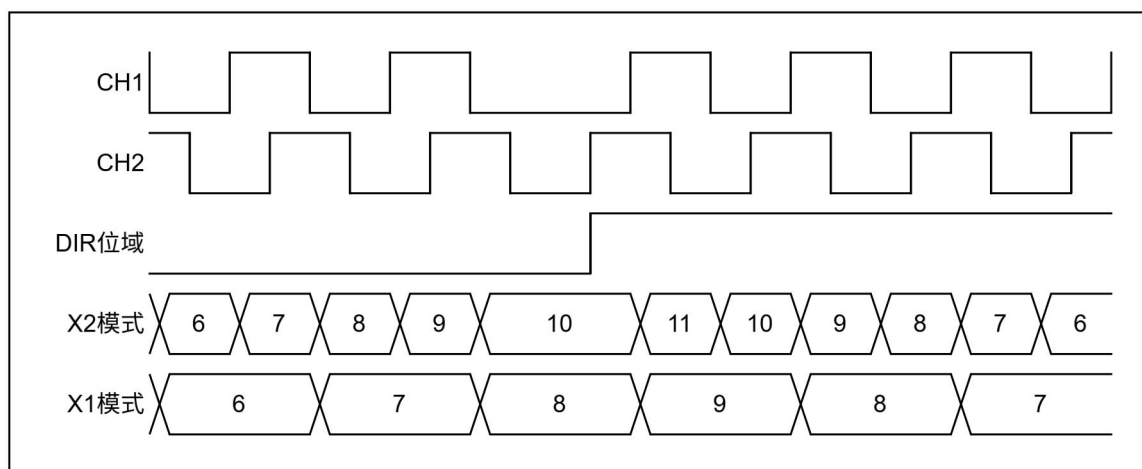


图 13-31 时钟加方向编码器模式计数示例

### 13.3.12.10 定向时钟编码器模式

定时器还支持定向时钟编码器模式，需配置从模式控制寄存器 ATIM\_SMCR 的 SMS 位域为 0xC 或 0xD，分别对应 x2 模式和 x1 模式。其功能框图可参见图 1-28 编码器模式框图，可对输入信号进行滤波和极性控制。

定向时钟编码器会根据旋转方向分别提供一条向上计数时钟线和向下计数时钟线。在 x2 模式，计数器在两个时钟线中的任意一个的上升沿和下降沿计数；CC1P 和 CC2P 位是时钟空闲状态的编码，CCyP 为 0 对应高电平空闲，CCyP 为 1 对应低电平空闲。在 x1 模式，计数器根据 CC1P 和 CC2P 位域的值在单个时钟边沿计数，CCyP 为 0 对应下降沿敏感和高电平空闲，CCyP 为 1 对应上升沿敏感和低电平空闲。

不同模式下计数方向和编码器信号的关系如下表所示：

表 13-6 计数方向和编码器信号的关系

模式	SMS	信号的电平		TI1FP1		TI2FP2	
		TI2FP2	TI1FP1	上升	下降	上升	下降
x2 模式 CCyP=0	1100	高	高	向下计数	向下计数	向上计数	向上计数
		低	低	不计数	不计数	不计数	不计数
x2 模式 CCyP=1		高	高	不计数	不计数	不计数	不计数
		低	低	向下计数	向下计数	向上计数	向上计数
x1 模式 CCyP=0	1101	高	高	不计数	向下计数	不计数	向上计数
		低	低	不计数	不计数	不计数	不计数

x1 模式		高	高	不计数	不计数	不计数	不计数
CCyP=1		低	低	向下计数	不计数	向上计数	不计数

下图是定向时钟编码器模式的计数示例：

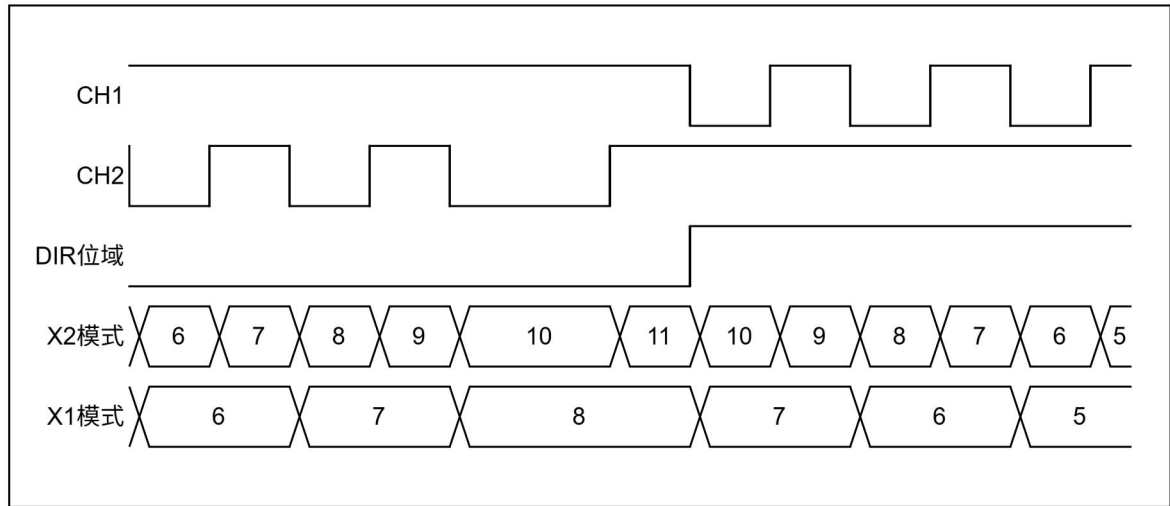


图 13-32 定向时钟编码器模式计数示例 ( $CC1P = CC2P = 0$ )

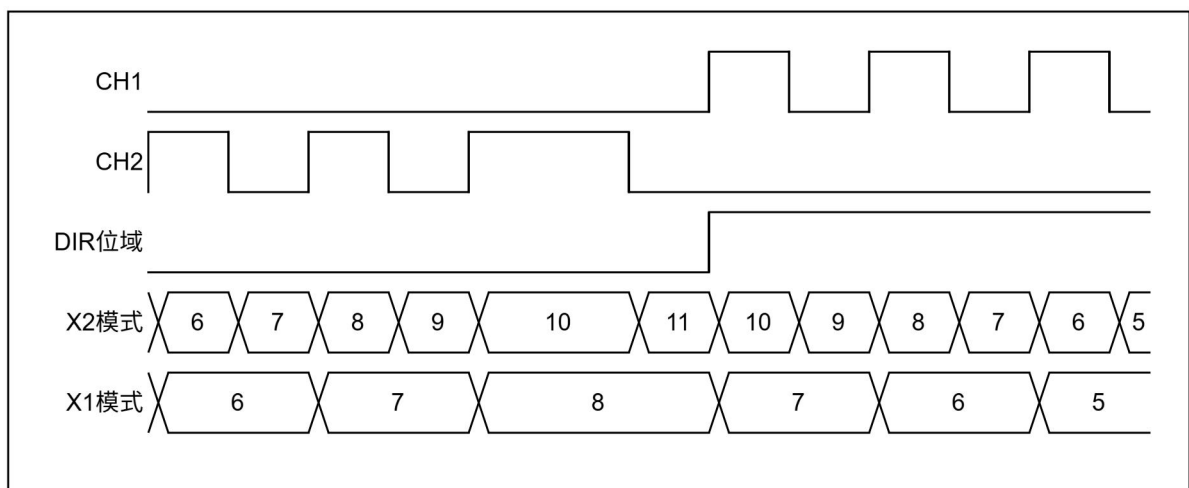


图 13-33 定向时钟编码器模式计数示例 ( $CC1P = CC2P = 1$ )



### 13.3.13 输入捕获功能

ATIM 支持输入捕获功能，设置捕获模式寄存器 ATIM\_CCMRxCAP 的 CCyS 位域为非零值时，CCy 通道配置为输入，同时指定对应捕获通道 ICy 的输入映射。支持通过软件或硬件触发输入捕获，其功能框图如下图所示：

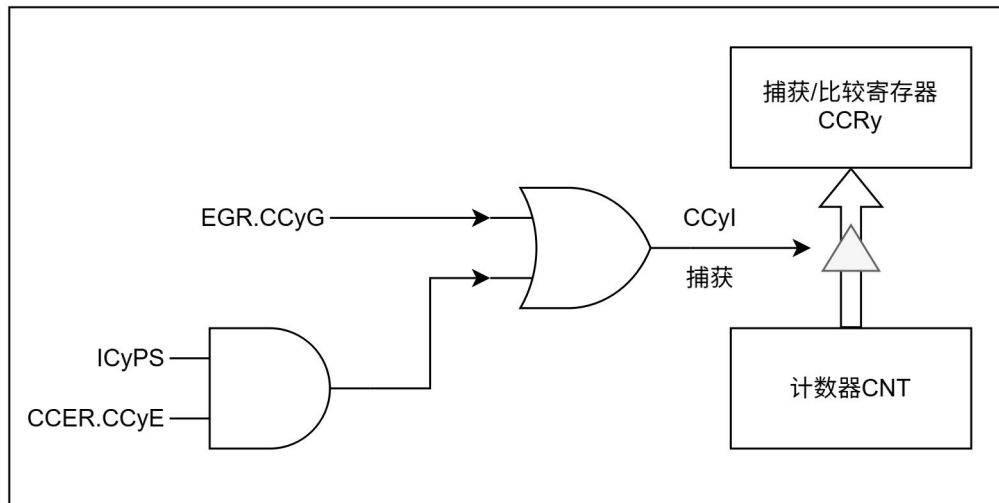


图 13-34 输入捕获模式框图

软件触发：设置 ATIM\_EGR.CCyG 为 1 时，立即软件触发一次捕获，当前计数器 CNT 的值被锁存到对应通道的捕获 / 比较寄存器 CCRy 中，完成一次捕获，捕获完成后 CCyG 位被硬件自动清零。

硬件触发：当检测到输入通道 TIy 上的有效边沿后，当前计数器 CNT 的值被锁存到对应通道的捕获 / 比较寄存器 CCRy 中，完成一次捕获。触发捕获的有效边沿通过捕获 / 比较使能寄存器 ATIM\_CCER 的 CCyNP 和 CCyP 位域 来配置，如下表所示：

表 13-7 输入捕获模式配置

ATIM_CCER 寄存器		捕获触发条件
CCyNP 位域	CCyP 位域	
0	0	上升沿捕获
0	1	下降沿捕获
1	1	上下沿和下降沿均捕获

当发生一次捕获时，对应通道的捕获 / 比较中断标志 ATIM\_ISR.CCyIF 被硬件置 1，同时：

- 如果使能了中断（设置 ATIM\_IER.CCyIE 为 1），将产生中断请求。
- 如果发生捕获事件时，ATIM\_ISR.CCyIF 标志位已经为高，那么重复捕获标志位 ATIM\_ISR.CCyOF 将被硬件置 1。

● 设置 ATIM\_ICR.CCyIF 为 0 或读取捕获寄存器 CCRy 可清除 ATIM\_ISR.CCyIF 标志位， 设置 ATIM\_ICR.CCyOF 为 0 可清除 ATIM\_ISR.CCyOF 标志位。

#### 13.3.13.1 PWM输入模式

PWM 输入模式是输入捕获模式的一个应用。输入捕获功能支持同一通道输入信号被两路捕获寄存器进行捕获， 结合从模式复位功能，可方便测量 PWM 输入信号的周期和脉宽。

PWM 输入模式只能与 T1 和 TI2 信号配合使用，因为只有 TI1FP1 和 TI2FP2 与从模式控制器相连。下图是 PWM 信号从 TI1 通道输入时的时序示例：

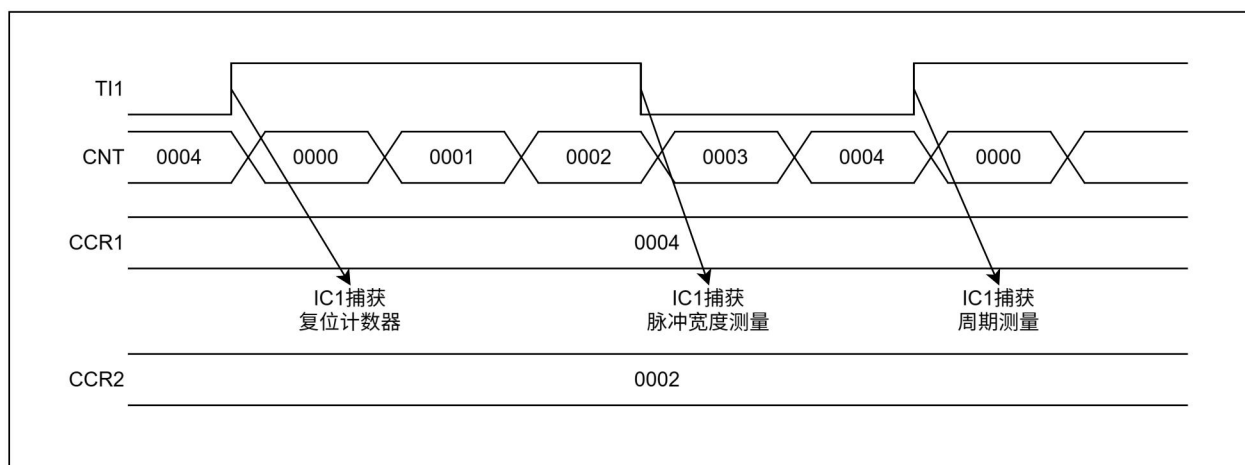


图 13-35 PWM 输入模式时序

### 13.3.13.2 输入捕获来源

ATIM 的输入捕获来源可以是外部 ATIM\_CHy 引脚，也可以是片内其它外设，通过 TI 输入选择寄存器 (ATIM\_TISEL1 和 ATIM\_TISEL2) 的 TIySEL 位域进行配置。

当 ATIM\_TISELx.TIySEL 为 0x0 时，TIy 通道输入捕获信号来源为外部 ATIM\_CHy 引脚，此时需通过 GPIO 复用功能寄存器 (CPIOx\_AFRH 和 CPIOx\_AFRL) 将对应引脚配置为复用功能。

当 ATIM\_TISELx.TIySEL 为 0x1~0xD 时，TIy 通道输入捕获信号来自片内其它外设，如下表所示：

表 13-8 高级定时器输入捕获来源

TIySEL 位域值	TIy 通道的输入捕获来源
0	ATIM_CHy 引脚
1	VC1_OUT
10	VC2_OUT
11	UART1_RXD
100	UART2_RXD
101	MCO_OUT
1001	LSI_OUT
1010	BTIM1_Trigo
1011	BTIM2_Trigo
1100	BTIM3_Trigo
1101	GTIM_Trigo

### 13.3.14 输出比较功能

ATIM 支持输出比较功能，设置比较模式寄存器 ATIM\_CCMRxCMP 的 CCyS 位域为 0 时，CCy 通道配置为输出。在输出比较模式下，当前计数器 CNT 的值与对应通道 CCy 的捕获 / 比较寄存器 CCRy 的值相比较，当两者匹配时，参考信号 OCyREF 输出为可设定的电平状态，同时产生比较中断。参考信号经输出控制单元后由 CHy 和 CHyN 引脚输出，其输出逻辑可参见 14.3.1.10 输出比较通道。

参考信号 OCyREF 的输出动作由比较模式寄存器 ATIM\_CCMRxCMP 的 OCyM 位域设置，如下表所示：

表 13-9 输出比较模式配置

OCyM 位域值	比较模式配置
0	比较匹配时 OCyREF 保持原电平
1	比较匹配时 OCyREF 置 1
10	比较匹配时 OCyREF 置 0
11	比较匹配时 OCyREF 翻转
100	强制 OCyREF 为低电平
101	强制 OCyREF 为高电平
110	PWM 模式 1
111	PWM 模式 2
1000	可再触发 OPM 模式 1
1001	可再触发 OPM 模式 2
1100	组合 PWM 模式 1
1101	组合 PWM 模式 2
1110	不对称 PWM 模式 1
1111	不对称 PWM 模式 2

当发生比较匹配时，对应通道的捕获 / 比较中断标志 ATIM\_ISR.CCyIF 被硬件置 1，同时：

- 如果使能了中断（设置 ATIM\_IER.CCyIE 为 1），将产生中断请求。
- 设置 ATIM\_ICR.CCyIF 为 0 可清除 ATIM\_ISR.CCyIF 标志位。

捕获 / 比较寄存器 ATIM\_CCRy 具有缓存功能，通过 OCyPE 位域选择是否使用缓存功能。当 OCyPE 为 0 时，禁止通道 CCy 的比较缓存功能，可在任意时候通过软件更新 ATIM\_CCRy 寄存器，更新值立即生效并影响输出波形；当 OCyPE 为 1 时，使能通道 CCy 的比较缓存功能，更新 ATIM\_CCRy 寄存器不会立即生效，仅当发生更新事件 UEV 时才会将 ATIM\_CCRy 寄存器的值更新到有效寄存器。

#### 13.3.14.1 匹配输出模式

匹配输出模式用于控制输出波形，或指示已经过某一段时间段。

- 设置 OCyM 位域为 0x0，比较匹配时，参考信号 OCyREF 保持其电平，同时 ATIM\_ISR.CCyIF 标志位置 1。
- 设置 OCyM 位域为 0x1，比较匹配时，参考信号 OCyREF 设置为高电平，同时 ATIM\_ISR.CCyIF 标志位置 1。
- 设置 OCyM 位域为 0x2，比较匹配时，参考信号 OCyREF 设置为低电平，同时 ATIM\_ISR.CCyIF 标志位置 1。
- 设置 OCyM 位域为 0x3，比较匹配时，参考信号 OCyREF 发生翻转，同时 ATIM\_ISR.CCyIF 标志位置 1。下图是各比较匹配输出模式时序示例，示例为边沿对齐递增模式。

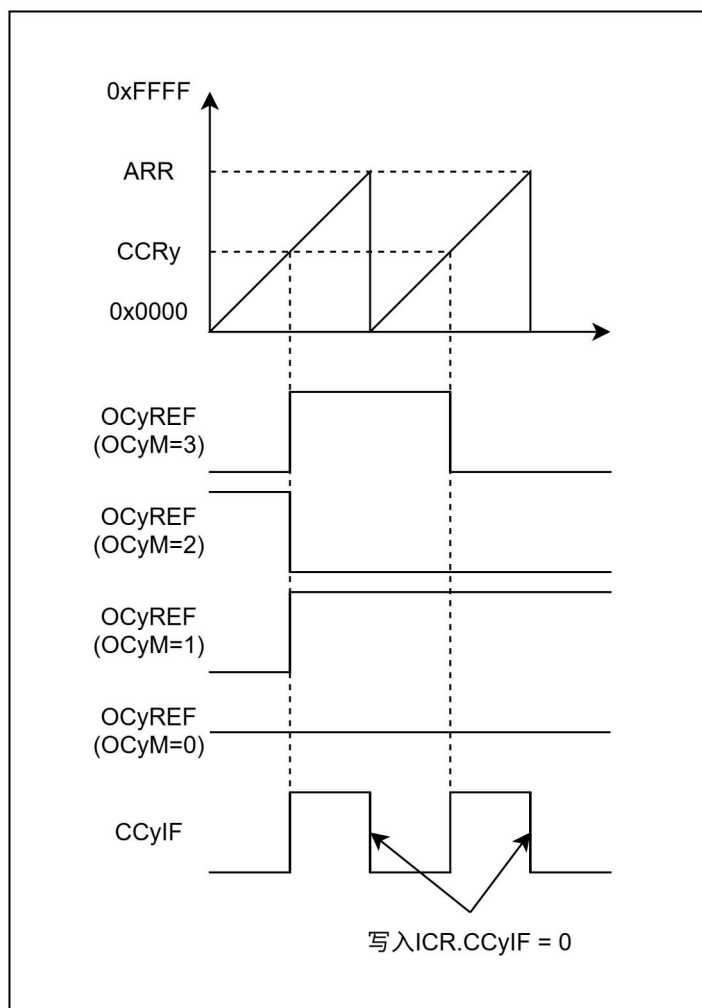


图 13-36 比较匹配输出模式

### 13.3.14.2 强制输出模式

在强制输出模式下，输出比较信号能够直接由软件强置为高或低状态，而不依赖于捕获 / 比较寄存器 ATIM\_CCRy 和计数寄存器 ATIM\_CNT 的比较结果。

设置比较模式寄存器 ATIM\_CCMRxCMP 的 OCyM 位域为 0x4，即可将参考信号 OCyREF 强制变为低电平；设置 OCyM 位域为 0x5，即可将参考信号 OCyREF 强制变为高电平。

强制输出模式下，ATIM\_CCRy 寄存器和计数器 ATIM\_CNT 之间的比较仍然在进行，相应的标志也会置 1，也会产生相应的中断请求。

### 13.3.14.3 PWM模式

脉冲宽度调制（PWM）模式可以产生一个由重载寄存器 ATIM\_ARR 确定频率、由捕获 / 比较寄存器 ATIM\_CCRy 确定占空比的信号。

各通道可以独立选择 PWM 模式，只需向比较模式寄存器 ATIM\_CCMRxCMP 的 OCyM 位域写入 0x6（PWM 模式 1）或 0x7（PWM 模式 2），输出状态如下表所示：

表 13-10 PWM 模式 1/2 输出状态

工作模式	计数方向	PWM 模式 1	PWM 模式 2
边沿对齐模式	向上	CNT < CCRy 时，OCyREF 为高	CNT < CCRy 时，OCyREF 为低
中央对齐模式	向下	CNT > CCRy 时，OCyREF 为低	CNT > CCRy 时，OCyREF 为高

OCy 的极性可以通过 ATIM\_CCER 寄存器的 CCyP 位域设置,可将其设置为高电平有效或低电平有效。通过 CCyE、CCyNE、MOE、OSSI 和 OSSR 位(ATIM\_CCER 和 ATIM\_BDTR 寄存器)的组合使能 OCy 输出。

以下是不同计数模式下各 PWM 模式的波形实例,其中 ATIM\_ARR 为 0x08。

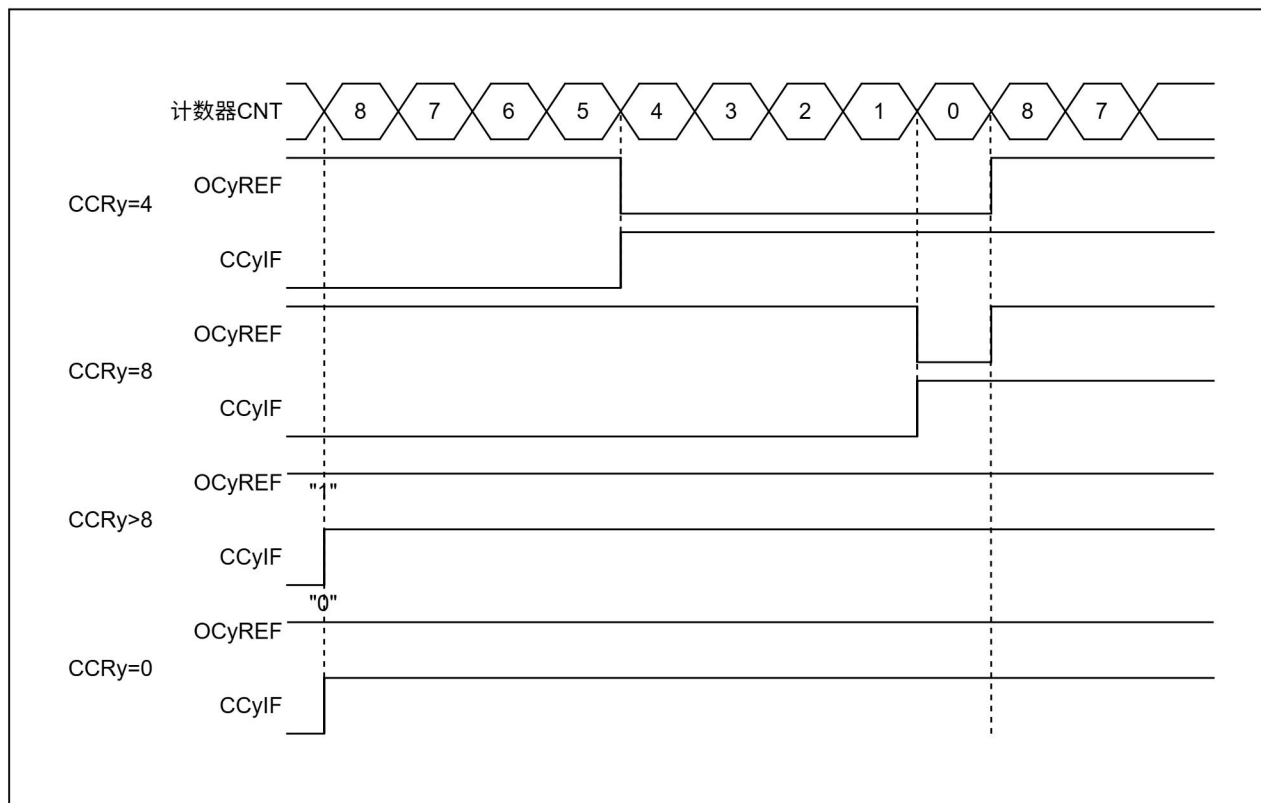


图 13-37PWM 模式 1, 边沿对齐模式, 递增计数

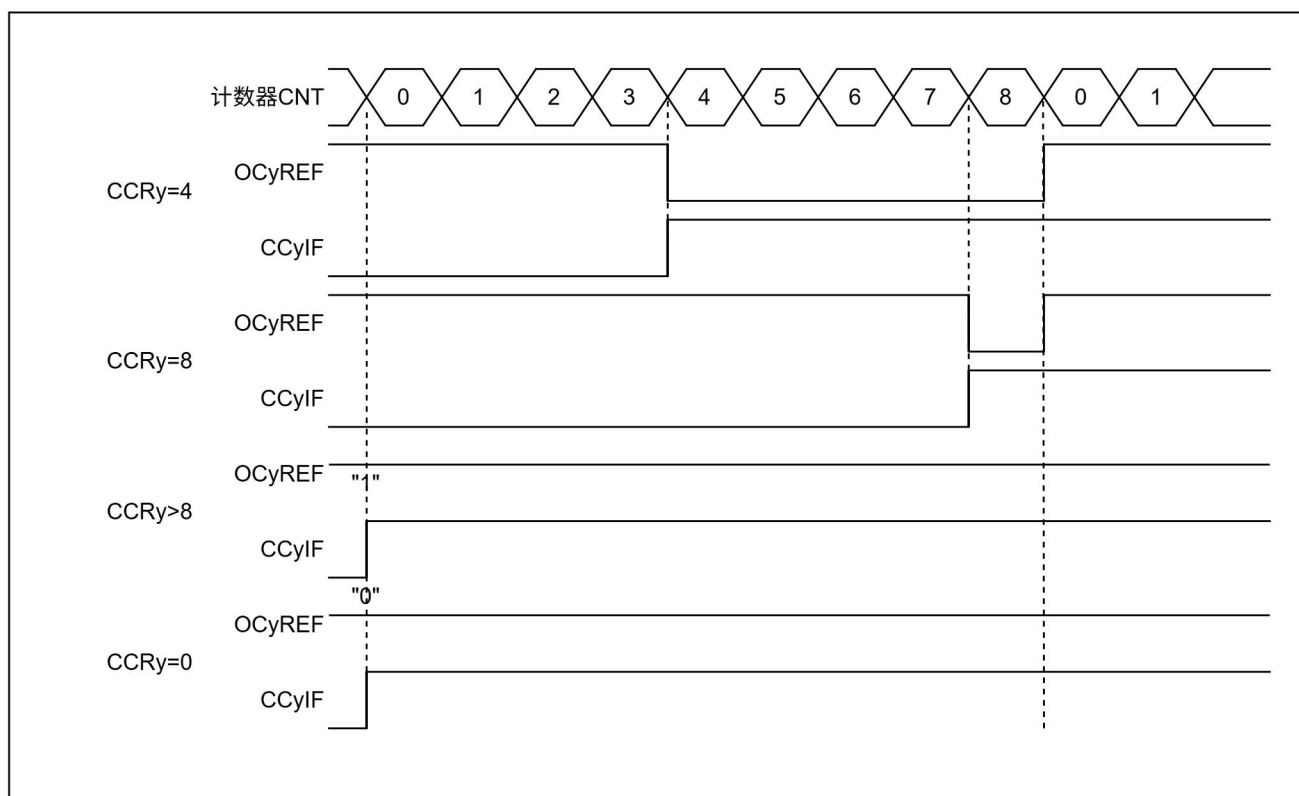


图 13-38 PWM 模式 1，边沿对齐模式，递减计数

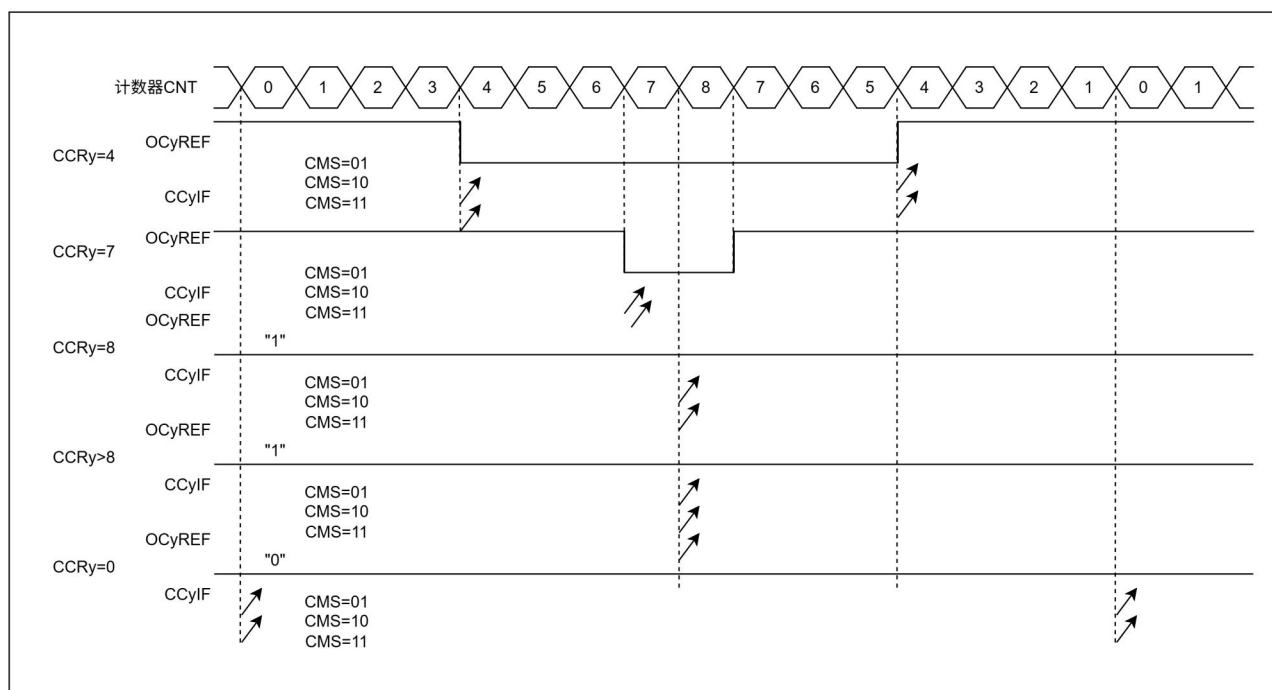


图 13-39 PWM 模式 1，中心对齐模式

#### 13.3.14.4 不对称PWM模式

不对称 PWM 输出是相对于中心对齐模式计数时的对称 PWM 输出而言的。在不对称 PWM 输出模式下，允许在中心对齐模式下生成两个具有可编程相移的 PWM 信号，频率由重载寄存器 ATIM\_ARR 确定，占空比和相移由两个捕获 / 比较寄存器 ATIM\_CCRy 确定，其中一个 CCRy 寄存器控制递增计数时的 PWM，另一个 CCRy 寄存器控制递减计数时的 PWM，具体如下所示：

- OC1REFC 由 CCR1( 递增计数 ) 与 CCR2( 递减计数 ) 控制
- OC2REFC 由 CCR2( 递增计数 ) 与 CCR1( 递减计数 ) 控制
- OC3REFC 由 CCR3( 递增计数 ) 与 CCR4( 递减计数 ) 控制
- OC4REFC 由 CCR4( 递增计数 ) 与 CCR3( 递减计数 ) 控制
- OC5REFC 由 CCR5( 递增计数 ) 与 CCR6( 递减计数 ) 控制
- OC6REFC 由 CCR6( 递增计数 ) 与 CCR5( 递减计数 ) 控制

各通道可以独立选择不对称 PWM 模式，只需向比较模式寄存器 ATIM\_CCMRxCMP 的 OCyM 位域写入 0xE（不对称 PWM 模式 1）或 0xF（不对称 PWM 模式 2）。

当设定通道用作不对称 PWM 输出通道时，另一辅助通道仍可使用。例如，通道 1 配置为不对称 PWM 模式 1 产生 OC1REFC 信号时，通道 2 仍可以配置为 PWM 模式 2 输出 OC2REF 信号，或者配置为不对称 PWM 模式 2 输出 OC2REFC 信号。

以下是不对称 PWM 模式 1/2 的波形实例，其中 ARR=8，CCR1=0，CCR2=8，CCR3=3，CCR4=5。

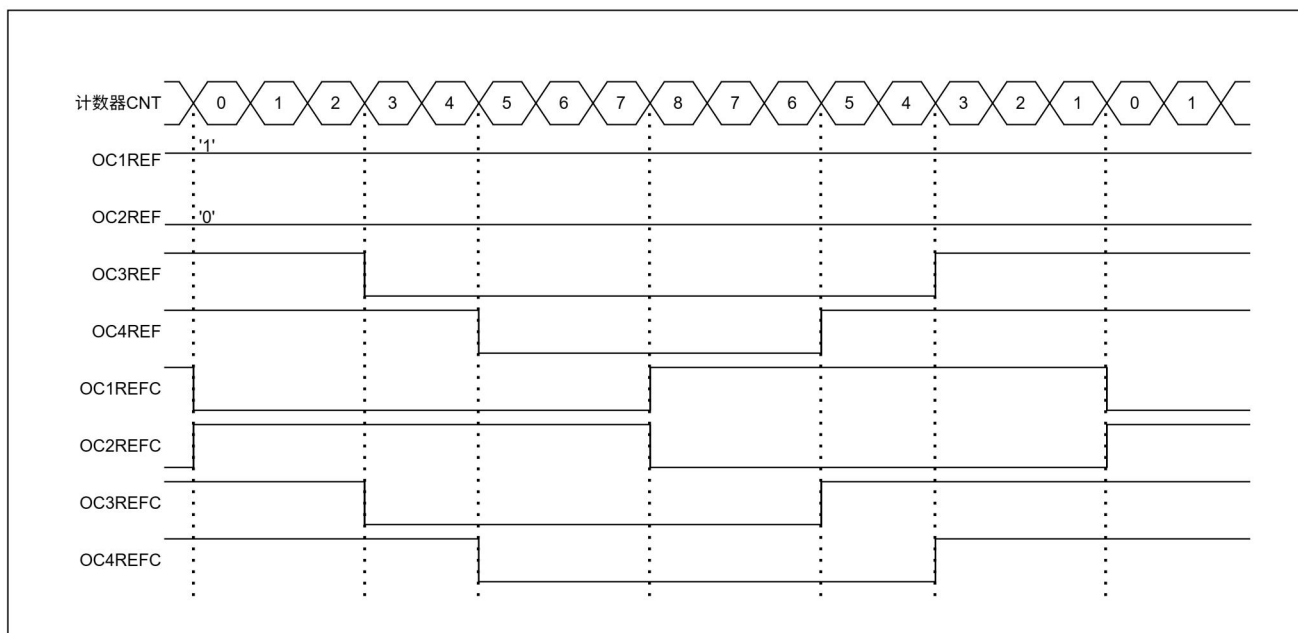


图 13-40 不对称 PWM 模式 1 示例

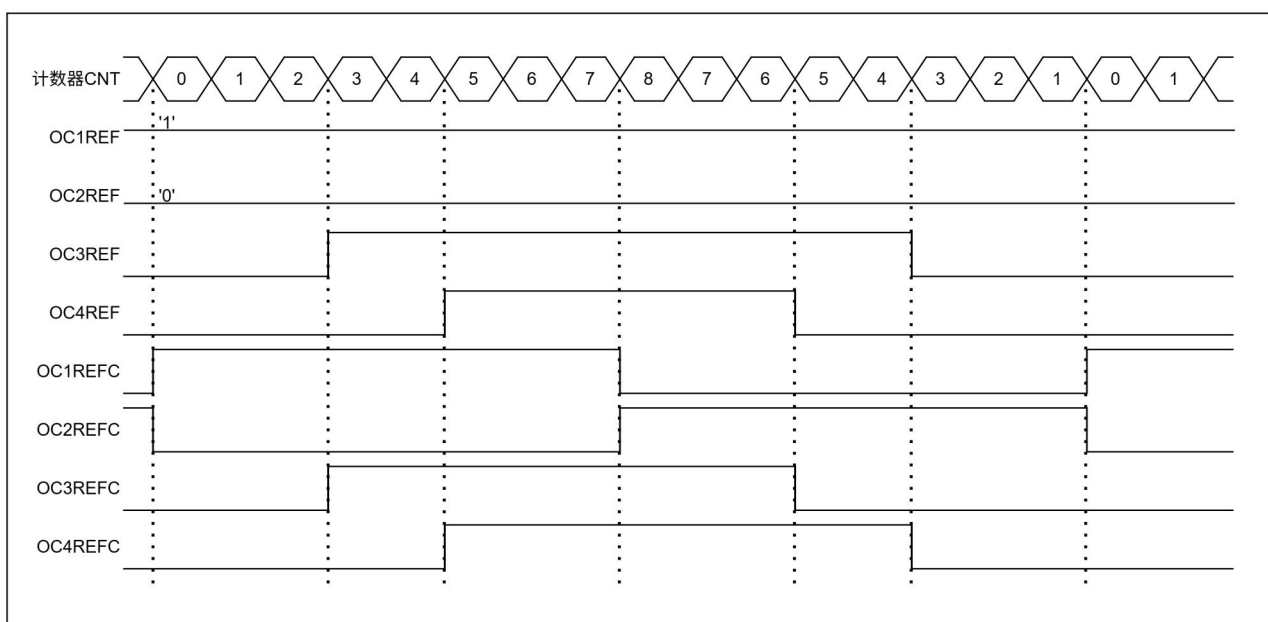


图 13-41 不对称 PWM 模式 2 示例

### 13.3.14.5 组合PWM模式

在组合 PWM 输出模式下，允许在边沿对齐或中心对齐模式下生成两个具有可编程延时和相移的 PWM 脉冲，频率由重载寄存器 ATIM\_ARR 确定，占空比和延时由两个捕获 / 比较寄存器 ATIM\_CCRy 确定。

该模式下，OCyREFC 信号由两个参考信号 OCyREF 的逻辑或运算或者逻辑与运算组合生成，通过比较模式寄存器 ATIM\_CCMRxCMP 的 OCyM 位域可独立选择组合 PWM 模式，具体如下表所示：

表 13-11 组合 PWM 模式配置

OCyM 位域值	输出状态
0xC (组合 PWM 模式 1)	OC1REFC (或 OC2REFC) 由 OC1REF 和 OC2REF 逻辑或运算生成 OC3REFC (或 OC4REFC) 由 OC3REF 和 OC4REF 逻辑或运算生成



	OC5REFC（或 OC6REFC）由 OC5REF 和 OC6REF 逻辑或运算生成
0xD (组合 PWM 模式 2)	OC1REFC（或 OC2REFC）由 OC1REF 和 OC2REF 逻辑与运算生成 OC3REFC（或 OC4REFC）由 OC3REF 和 OC4REF 逻辑与运算生成 OC5REFC（或 OC6REFC）由 OC5REF 和 OC6REF 逻辑与运算生成

当设定通道用作组合 PWM 输出通道时，另一辅助通道仍可使用，但必须设置为相反的 PWM 模式。例如，通道 1 配置为组合 PWM 模式 1 产生 OC1REFC 信号时，通道 2 仍可使用，通道 2 可以配置为 PWM 模式 2 输出 OC2REF 信号。

下图显示了边沿对齐递增模式下，组合 PWM 模式 1 的信号示例，具体配置如下：

- 重载值 ARR 为 8
- 通道 1 配置为组合 PWM 模式 1，CCR1=2
- 通道 2 配置为 PWM 模式 2，CCR2=3
- 通道 3 配置为组合 PWM 模式 1，CCR3=6
- 通道 4 配置为 PWM 模式 2，CCR4=7

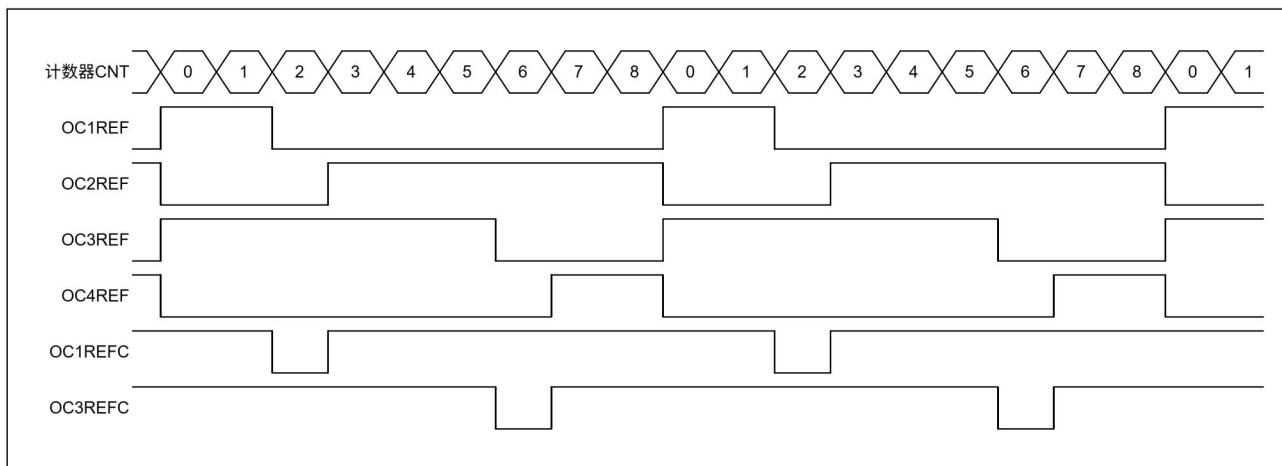


图 13-42 组合 PWM 模式 1 示例

下图显示了边沿对齐递增模式下，组合 PWM 模式 2 的信号示例，具体配置如下：

- 重载值 ARR 为 8
- 通道 1 配置为组合 PWM 模式 1，CCR1=2
- 通道 2 配置为 PWM 模式 1，CCR2=3
- 通道 3 配置为组合 PWM 模式 2，CCR3=6
- 通道 4 配置为 PWM 模式 1，CCR4=7

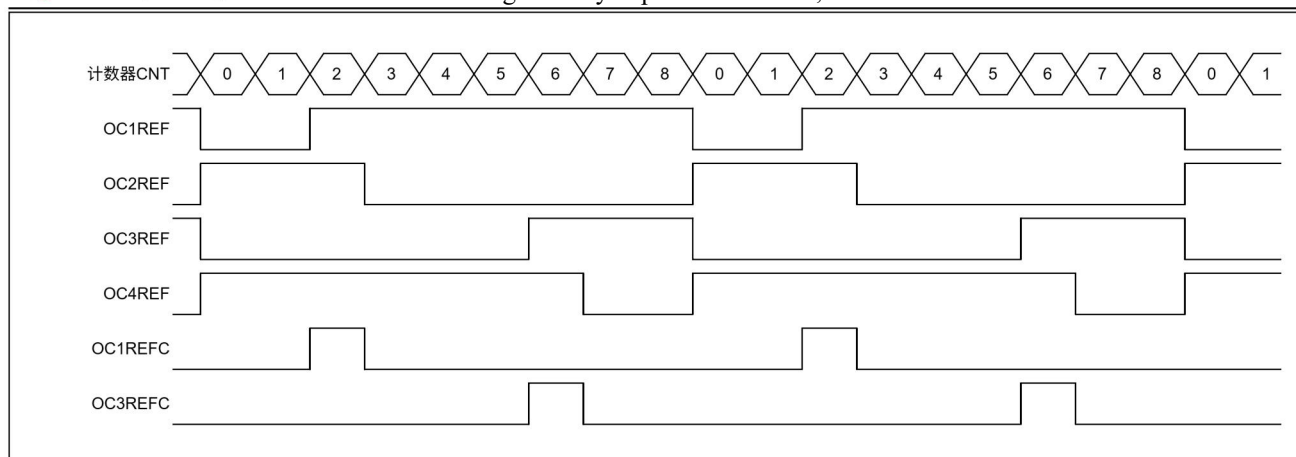


图 13-43 组合 PWM 模式 2 示例

### 13.3.14.6 互补输出和死区输入

ATIM 可以输出 6 对互补 PWM 信号，且支持对称或不对称的死区插入时间。

主输出 OCy 和互补输出 OCyN 可独立设置输出极性，分别通过 ATIM\_CCER 寄存器的 CCyP 和 CCyNP 位域进行控制。

主输出 OCy 和互补输出 OCyN 通过以下多个控制位的组合进行激活：ATIM\_CCER 寄存器的 CCyE 和 CCyNE 位，ATIM\_BDTR 寄存器的 MOE、OSSI、OSSR 位，以及 ATIM\_CR2 寄存器的 OISy、OISyN 位，更多详细信息，请参见表 1-13 有刹车功能的互补通道 OCy 和 OCyN 的输出控制位。应当注意，切换至空闲状态（MOE 变为 0）的时刻，死区仍然有效。

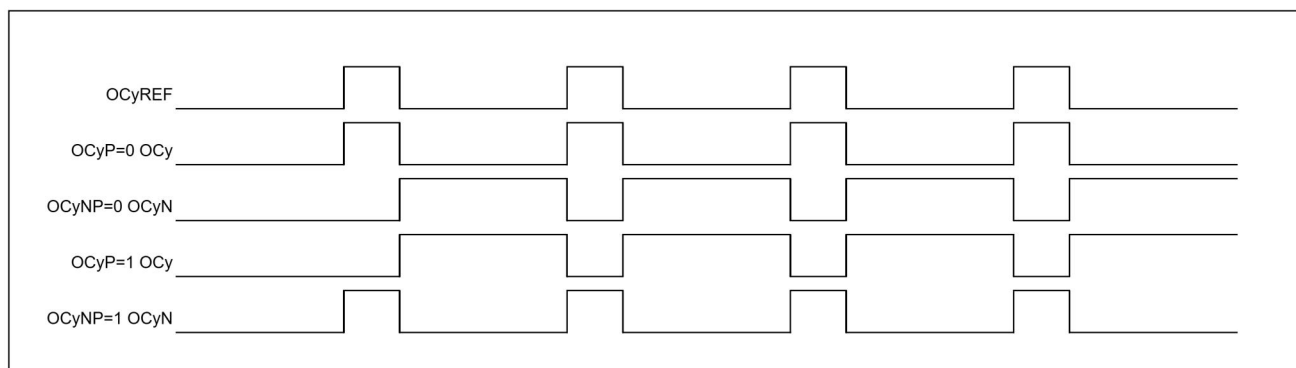


图 13-44 互补输出 PWM 波形

ATIM 支持对称或不对称的死区插入时间，死区时间对所有通道均相同。当 ATIM\_DTR2.DTAE 为 0 时，上升沿和下降沿的死区时间是相同的，由 ATIM\_BDTR 寄存器的 DTG[7:0] 定义；当 ATIM\_DTR2.DTAE 为 1 时，上升沿的死区时间由 DTG[7:0] 定义，下降沿的死区时间由 DTGF[7:0] 定义。DTAE 位必须在使能计数器之前写入，且 CEN 为 1 时不得修改。DTG[7:0] 定义如下表所示，DTGF[7:0] 定义与 DTG[7:0] 相同。

表 13-12 死区时间设置

ATIM_BDTR.DTG	步长	死区时间	当 tDTS=125ns 时，死区时间范围
DTG[7:5]=0xx	1tDTS	DTG[7:0]×tDTS	0~15.875 μs
DTG[7:5]=10x	2tDTS	(64+DTG[5:0])×2×tDTS	16~31.750 μs

DTG[7:5]=110	8tDTS	$(32+DTG[4:0]) \times 8 \times tDTS$	32~63 $\mu s$
DTG[7:5]=111	16tDTS	$(32+DTG[4:0]) \times 16 \times tDTS$	64-126 $\mu s$

下图所示为对称和不对称死区插入时间时，互补 PWM 输出信号与参考信号 OCyREF 之间的关系，示例中 CCyP=0、CCyNP=0。

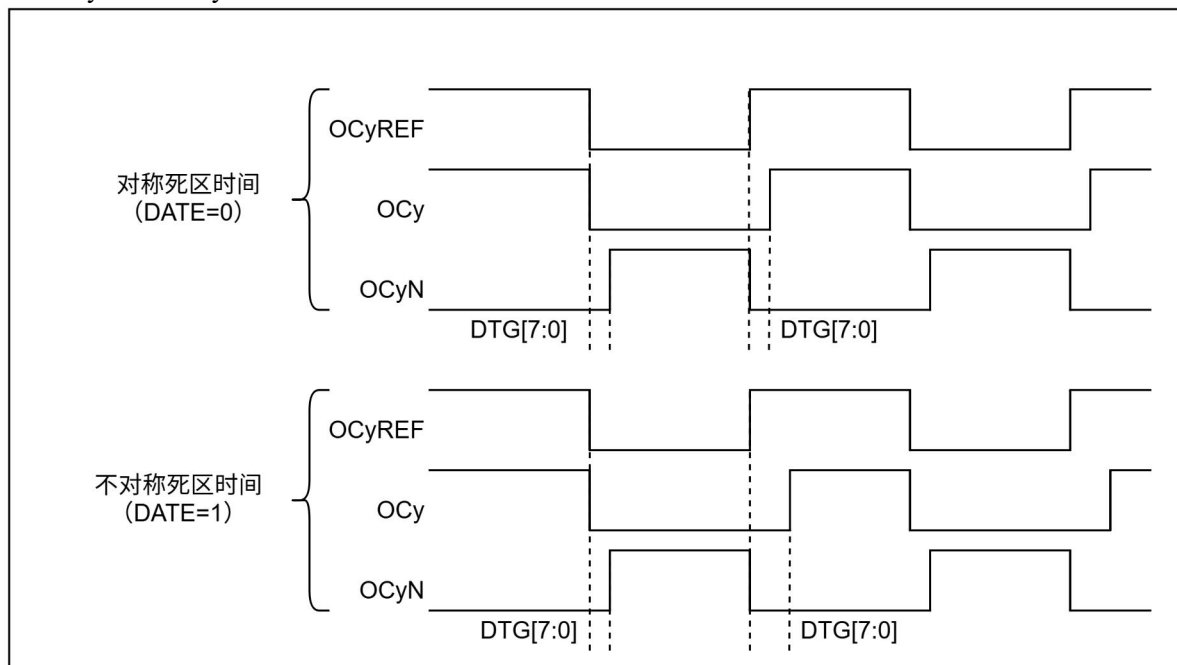


图 13-45 带死区插入的互补 PWM 输出波形

可以使用预加载机制在 PWM 操作期间即时更新死区时间值。当 DTPE 位置 1 时，使能死区时间值预加载，预加载值在下一个更新事件时加载到有效寄存器中。如果在计数器使能时 DTPE 位使能，则自上次更新以来写入的任何新值都将被丢弃，并使用先前的值。

### 13.3.14.7 刹车功能

ATIM 支持两个刹车输入 BRK 和 BRK2，当刹车信号有效时，刹车电路会关闭 PWM 输出，并将其强制为预定义的安全状态。

刹车通道源包括系统级故障（时钟失效和奇偶校验错误等）和应用故障（来自输入引脚和内置比较器），可以在死区持续时间后将输出强制为预定义的电平（有效或无效）。刹车 2 通道源包括应用故障，只能将输出强制为无效状态。BRK 输入的优先级高于 BRK2 输入。

#### 刹车（BRK）触发源：

- 软件刹车，设置 ATIM\_EGR 寄存器的 BG 位域为 1 生成刹车事件，BG 位由硬件自动清零。
- 电压比较器 VC1/2 的比较输出，ATIM\_AF1 寄存器相关位域可设置有效极性和输入使能。
- 系统刹车请求，通过 SYSCTRL\_CR2 寄存器相应位域使能：
  - 产生 HardFault 或 Cortex-M0+ LockUp 标志
  - RAM 奇偶校验错误
  - DeepSleep
- 外部 ATIM\_BK 引脚输入（GPIO 功能复用设置），ATIM\_AF1 寄存器相关位域可设置有效极性和输入使能。

#### 刹车 2（BRK2）触发源：

- 软件刹车，设置 ATIM\_EGR 寄存器的 B2G 位域为 1 生成刹车事件，B2G 位由硬件自动清零。
- 电压比较器 VC1/2 的比较输出，ATIM\_AF2 寄存器相关位域可设置有效极性和输入使能。

- 外部 ATIM\_BK2 引脚输入 (GPIO 功能复用设置), ATIM\_AF2 寄存器相关位域可设置有效极性和输入使能。在所有源进入定时器 BRK 或 BRK2 输入之前, 对其进行逻辑或运算, 如下图所示:

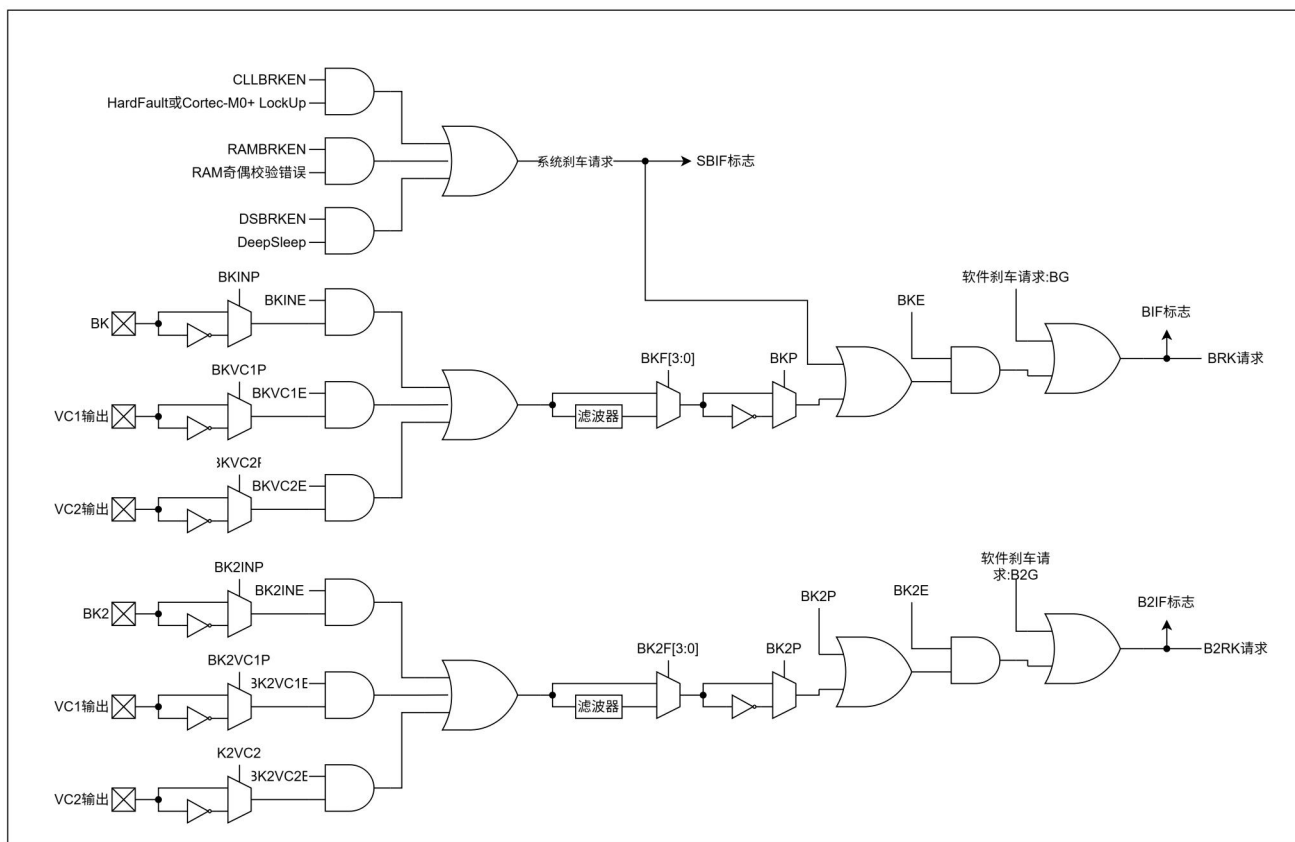


图 13-46 刹车和刹车 2 电路

注意:

只有禁止可编程滤波器时才能保证异步（无时钟）操作。如果使能可编程滤波器，必须使用故障安全时钟模式来保证能够处理刹车事件。

发生刹车事件时，MOE 位异步清零，使输出处于无效状态、空闲状态甚至释放控制权给 GPIO 控制器（通过 OSSR 位进行选择，具体请参见表 1-13 有刹车功能的互补通道 OCy 和 OCyN 的输出控制位）。即使 MCU 振荡器关闭，该功能仍然使能。如果 AOE 位置 1，则 MOE 位会在发生下一更新事件（UEV）时自动再次置 1，否则 MOE 将始终保持低电平，直到应用将其再次置 1。因为刹车输入为电平有效，因此，当刹车输入为有效电平时，MOE 位不能被置 1（自动或通过软件），状态标志 BIF 和 B2IF 也不能被清零。

下图所示为发生刹车事件 BRK 时 OCy 和 OCyN 的输出行为示例:

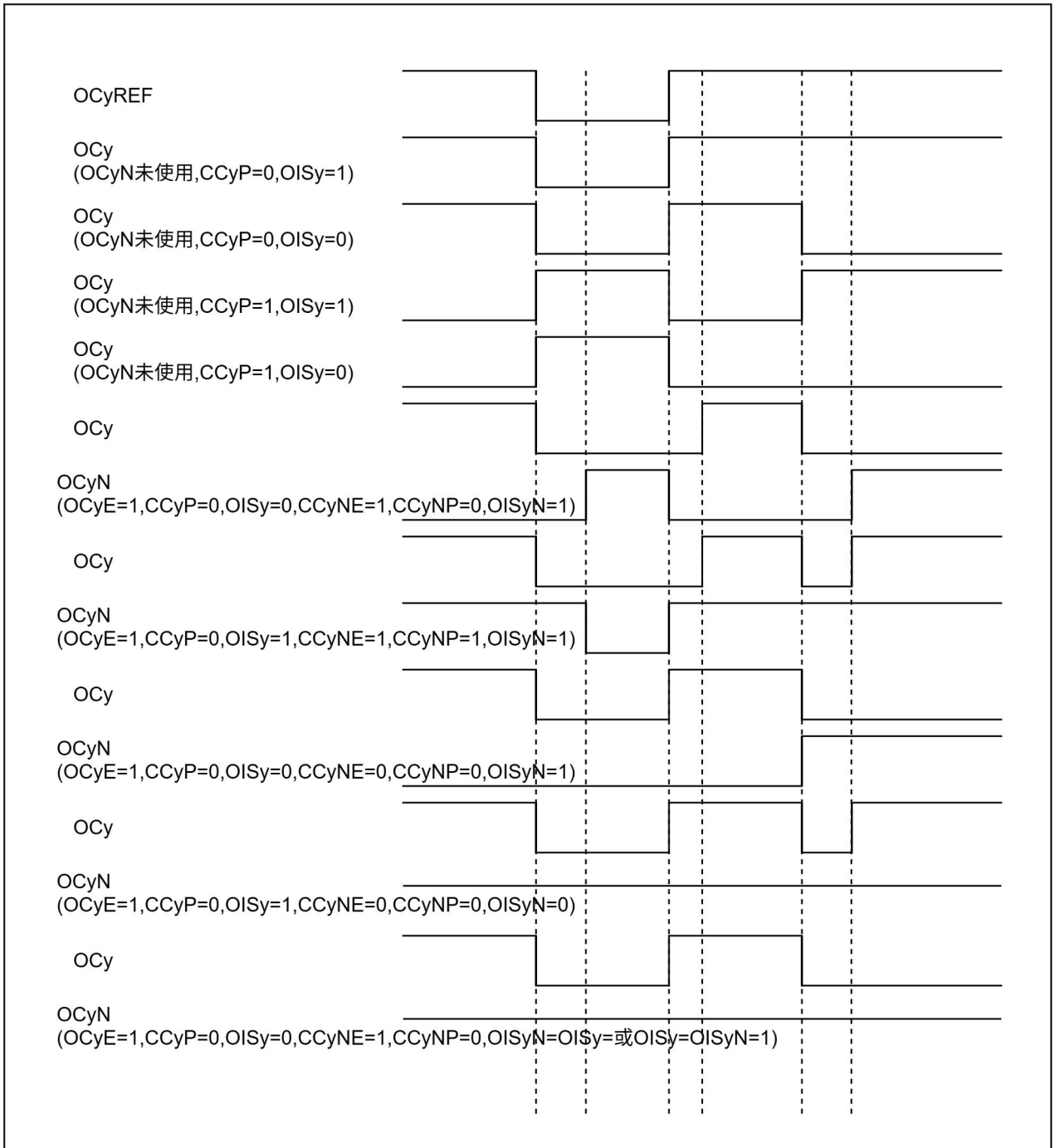


图 13-47 响应刹车事件 BRK 时的输出行为 (OSSI=1)

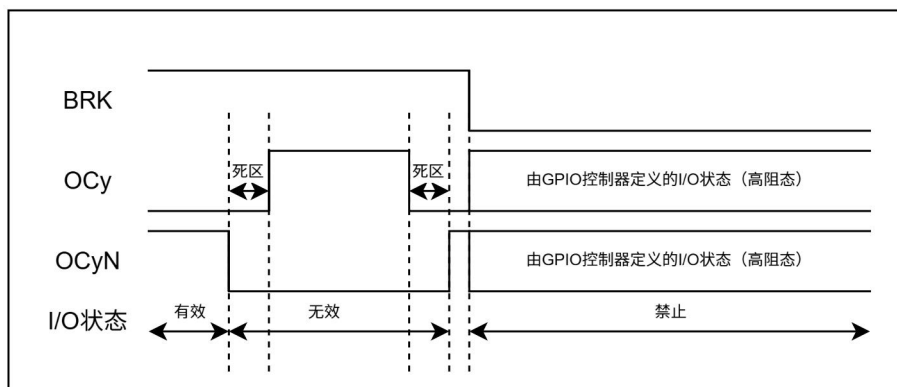


图 13-48 响应刹车事件 BRK 时的输出行为 (OSSI=0)

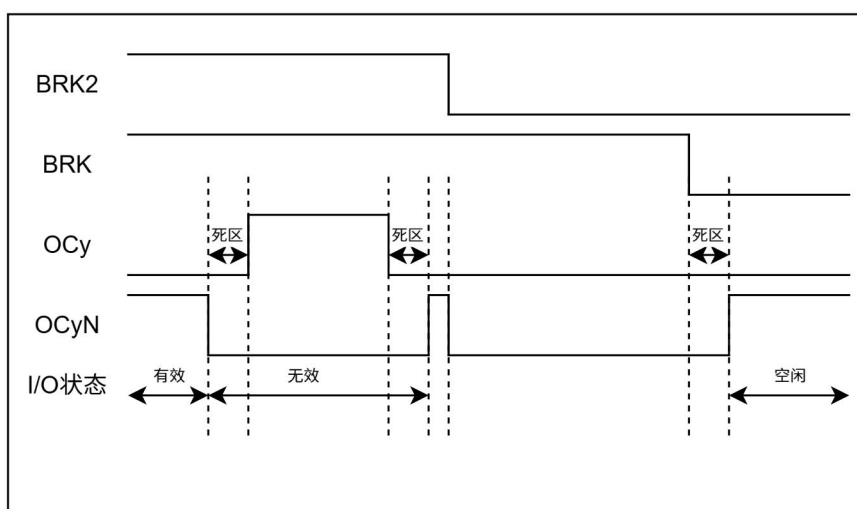


图 13-49 响应刹车 BRK 和 BRK2 时的输出行为 (OSSI=1)

表 13-13 有刹车功能的互补通道 OCy 和 OCyN 的输出控制位

输出控制位					输出状态	
MOE	OSSI	OSSR	CCyE	CCyNE	OCy 输出状态	OCyN 输出状态
1	X	X	0	0	禁止输出（不再由定时器驱动）。 输出状态由 GPIO 逻辑强制为高阻态。 OCy=0、OCyN=0	
		0	0	1	禁止输出（不再由定时器驱动）。 输出状态由 GPIO 逻辑强制为高阻态。 OCy=0	OCyREF+极性 OCyN=OCyREF 异或 CCyNP
		0	1	0	OCyREF+极性 OCy=OCyREF 异或 CCyP	禁止输出（不再由定时器驱动）。 输出状态由 GPIO 逻辑强制为高阻态。 OCyN=0
		X	1	1	OCyREF+极性+死区	OCyREF 互补+极性+死区
		1	0	1	关闭状态（输出使能为无效状态） OCy=CCyP	OCyREF+极性 OCyN=OCyREF 异或 CCyNP

		1	1	0	OCyREF+极性 OCy=OCyREF 异或 CCyP	关闭状态（输出使能为无效状态） OCyN=CCyNP
0	0	X	X	X	禁止输出（不再由定时器驱动）。	
	1		0	0	输出状态由 GPIO 逻辑强制为高阻态。	
			0	1	关闭状态（输出使能为无效状态）	
			1	0	异步：OCy=CCyP、OCyN=CCyNP（如果触发 BRK 或 BRK2）。	
			1	1	随后（仅当触发 BRK 时才有效），如果存在时钟，则在死区时间后 OCy=OISy 且 OCyN=OISyN，假定 OISy 和 OISyN 并没有都设置成 OCy 及 OCyN 的有效电平（否则在半桥配置下驱动开关时可能导致短路）。 注：BRK2 只能在 OSSI=OSSR=1 时使用。	

注：如果一个通道的两个输出均未使用（由 GPIO 接管控制），则 OISy、OISyN、CCyP 和 CCyNP 位必须保持清 零状态。

### 13.3.14.8 可再触发单脉冲模式

可再触发单脉冲模式配合组合复位 + 触发模式 (SMS=8)，允许在触发信号 (TRGI) 的触发下复位并启动计数器，并产生长度可编程的脉冲，脉冲宽度由 ARR 确定。

与单脉冲模式相比，具有如下区别：

表 13-14 可再触发单脉冲模式和单脉冲模式

单脉冲模式	可再触发单脉冲模式
发生触发时，经一段可编程的延时后产生一个脉宽可编程的单脉冲	发生触发时，脉冲立即产生，无可编程延时
新的触发无效	如果上一个触发产生的脉冲未完成，又发生新的触发，脉冲将延长

此模式下，计数器只能设置为边沿对齐模式，不能设置为中心对齐模式。当配置为递增计数模式时，对应 CCRy 必须设置为 0；配置为递减计数模式时，对应 CCRy 必须大于或等于 ARR。

下图所示为可再触发单脉冲模式 2 的示例：

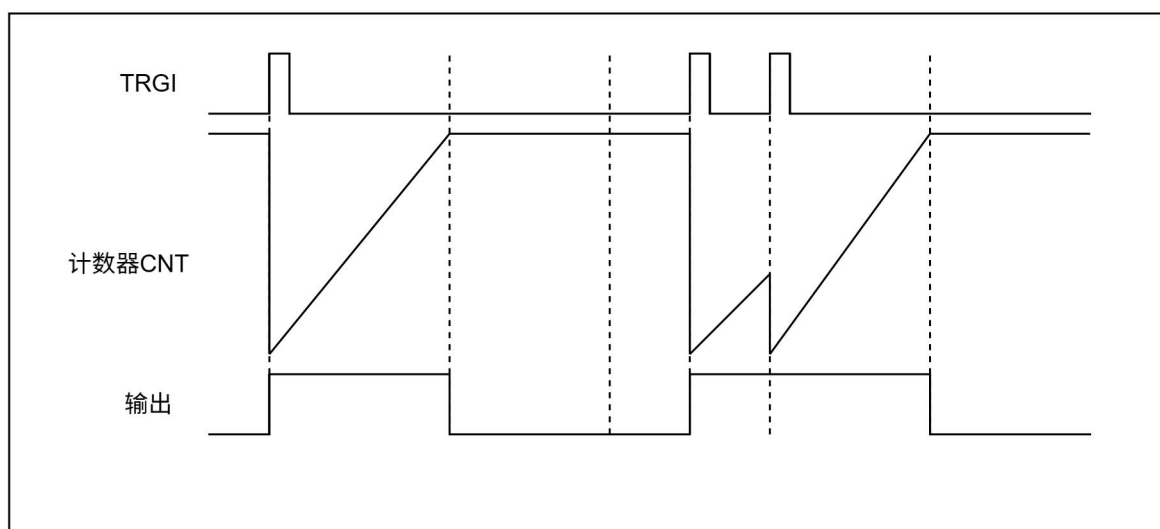


图 13-49 可再触发单脉冲模式 2 示例

### 13.3.14.9 参考信号清零使能

设置 ATIM\_CCMRxCMP 寄存器的 OCyCE 位域为 1，可打开对应通道的 OCyREF 信号清零功能，即当 OCREf\_CLR\_INT 输入上检测到高电平时，可将 OCyREF 信号清零，OCyREF 信号将保持低电平，直到发生下一更新事件 UEV。该功能只在输出比较和 PWM 模式下可用，强制输出模式下不可用。

清零源 OCREf\_CLR\_INT 可选 OCREf\_CLR 或 ETRF，通过从模式控制寄存器 ATIM\_SMCR 的 OCCS 位域进行选择，OCREf\_CLR 也有多种输入源，参见 1.9.31 ATIM\_AF2 复用功能选项寄存器 2 的 OCRSEL 位域说明。

下图所示为 OCREf\_CLR\_INT 输入变为高电平时 OCyREF 信号状态：



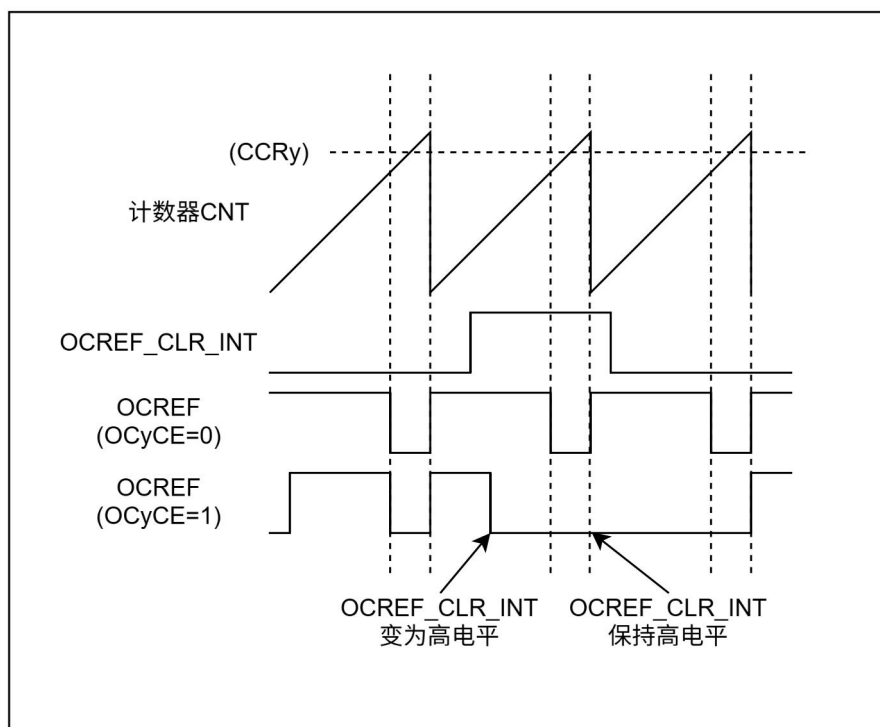


图 13-50OCyREF 清零示例

#### 13.3.14.10 UIF位重映射

设置 ATIM\_CR1 寄存器中的 UIFREMAP 位为 1 可使能 UIF 状态位重映射功能，可强制将更新中断标志 ATIM\_ISR. UIF 连续复制到 ATIM\_CNT 寄存器的 UIFCPY 位域中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位 翻转条件。这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

UIF 和 UIFCPY 标志使能之间没有延迟。

当 ATIM\_CR1.UIFREMAP 为 0 时，UIFCPY 位域保留，读为 0。

### 13.3.14.11 定时器级联ITR

通过 ITR 信号可以实现定时器级联，即将主定时器的触发输出（TRGO）连接到从定时器的 ITR 输入，进而连接到从定时器的触发输入（TRGI）。ITR 级联可实现多种功能，如将一个定时器用作另一个定时器的预分频器、主定时器对从定时器的计数器执行复位、使能或提供时钟等功能。

ATIM 的级联输入 ITR 的来源为 BTIM 和 GTIM 的触发输出信号（TRGO）以及 UART 的 TXD/RXD 信号，可通过从模式控制寄存器 ATIM\_SMCR 的 TS 位域来选择，具体配置如下表所示：

表 13-15ITR 信号来源

ATIM_SMCR.TS 位域值	ITR 信号名称	ITR 信号来源
0	ITR0	-
1	ITR1	-
10	ITR2	GTIM_Trigo
11	ITR3	-
1000	ITR4	-
1001	ITR5	-
1010	ITR6	UART1_TXD
1011	ITR7	UART2_TXD
1100	ITR8	UART1_RXD
1101	ITR9	UART2_RXD
1110	ITR10	BTIM1_Trigo
1111	ITR11	BTIM2_Trigo
10000	ITR12	BTIM3_Trigo

### 13.3.14.12 片内外设互联ETR

ATIM 的 ETR 信号来源可以是外部 ATIM\_ETR/GTIM\_ETR 引脚，也可以是片内其它外设，通过复用功能选项寄存器 ATIM\_AF1 的 ETRSEL 位域进行选择。

当 ETR 信号来自 ETR 引脚输入时，具体外部输入端口可参见 GPIO 复用功能分配表，并需通过 GPIO 复用功能寄存器 (CPIOx\_AFRH 和 CPIOx\_AFRL) 进行复用配置；当 ETR 信号来自片内其它外设时，可实现片内外设互联。

ETR 信号来源如下表所示：

表 13-16ETR 信号来源

AF1.ETRSEL 位域值	ETR 信号来源
0	ATIM_ETR 引脚
1	VC1_OUT
10	VC2_OUT
101	ADC_AWD
1000	LVD_OUT
1001	GTIM_ETR 引脚
1101	UART1_TXD
1110	UART2_TXD

### 13.3.15 ATIM 中断

ATIM 支持 16 个中断源，当 ATIM 中断事件发生时，中断标志位会被硬件置位，如果设置了对应的中断使能控制位，将产生中断请求。

在用户 ATIM 中断服务程序中，应查询相关 ATIM 中断标志位，以进行相应的处理，在退出中断服务程序之前，要清除该中断标志位，以避免重复进入中断服务程序。

各 ATIM 中断源的标志位、中断使能位、中断标志清除位或清除方法，如下表所示

表 13-17 ATIM 中断控制

中断事件	中断标志位	中断使能位	中断标志清除
更新中断	ISR.UIF	IER.UIE	写 0 到 ICR.UIF
捕获/比较 1 中断	ISR.CC1IF	IER.CC1IE	写 0 到 ICR.CC1IF 或读 CCR1 寄存器
捕获/比较 2 中断	ISR.CC2IF	IER.CC2IE	写 0 到 ICR.CC2IF 或读 CCR2 寄存器
捕获/比较 3 中断	ISR.CC3IF	IER.CC3IE	写 0 到 ICR.CC3IF 或读 CCR3 寄存器
捕获/比较 4 中断	ISR.CC4IF	IER.CC4IE	写 0 到 ICR.CC4IF 或读 CCR4 寄存器
捕获/比较 5 中断	ISR.CC5IF	IER.CC5IE	写 0 到 ICR.CC5IF 或读 CCR5 寄存器
捕获/比较 6 中断	ISR.CC6IF	IER.CC6IE	写 0 到 ICR.CC6IF 或读 CCR6 寄存器
COM 中断	ISR.COMIF	IER.COMIE	写 0 到 ICR.COMIF
触发中断	ISR.TIF	IER.TIE	写 0 到 ICR.TIF
刹车中断	ISR.BIF	IER.BIE	写 0 到 ICR.BIF
刹车 2 中断	ISR.B2IF	IER.BIE	写 0 到 ICR.B2IF
系统刹车中断	ISR.SBIF	IER.BIE	写 0 到 ICR.SBIF
编码器索引中断	ISR.IDXF	IER.IDXIE	写 0 到 ICR.IDXF
编码器方向改变中断	ISR.DIRF	IER.DIRIE	写 0 到 ICR.DIRF
编码器索引错误中断	ISR.IERRF	IER.IERRIE	写 0 到 ICR.IERRF
编码器转换错误中断	ISR.TERRF	IER.TERRIE	写 0 到 ICR.TERRF

### 13.3.16 触发 ADC

定时器支持多种内部信号触发启动 ADC，如捕获比较信号、触发输出信号 TRGO 和 TRGO2，触发输出 TRGO 和 TRGO2 有多种可能的事件，分别由控制寄存器 ATIM\_CR2 的 MMS 和 MMS2 位域进行选择。同时，ADC 外设需配置其外部触发启动寄存器，以选择对应触发源。

应注意，必须先使能 ADC 时钟，才能从主定时器接收事件，且从定时器接收触发信号时，不得实时更改 ADC 时钟。

### 13.3.17 调试支持

ATIM 支持在调试模式下停止或继续计数，通过调试状态定时器控制寄存器 SYSCTRL\_DEBUG 的 ATIM 位域来设置。

- 设置 SYSCTRL\_DEBUG.ATIM 为 1，则在调试状态时暂停 ATIM 的计数器计数。
- 设置 SYSCTRL\_DEBUG.ATIM 为 0，则在调试状态时 ATIM 的计数器继续计数。

## 13.3.18 编程示例

### 13.3.18.1 外部时钟模式 1 编程示例

以下示例中，配置递增计数器在 ATIM\_CH1 通道的上升沿进行计数，步骤如下：

步骤 1：设置 SYSCTRL\_AHBEN.GPIOx 为 1，SYSCTRL\_APBEN1.ATIM 为 1，打开 ATIM\_CH1 引脚对应的 GPIO 时钟和 ATIM 配置时钟及工作时钟；

步骤 2：将 ATIM\_CH1 引脚对应的 GPIO 配置成复用输入模式，具体寄存器配置请参见 通用输入输出端口（GPIO）；

步骤 3：设置 ATIM\_TISEL1.TI1SEL 为 0，选择 TI1 来源为 ATIM\_CH1 通道；

步骤 4：设置 ATIM\_CCMR1CAP.IC1F，配置输入滤波带宽；

步骤 5：设置 ATIM\_CCER.CC1NP 为 0、ATIM\_CCER.CC1P 为 0，选择上升沿计数有效；

步骤 6：设置 ATIM\_SMCR.TS 为 0x5，选择 TRGI 输入为 TI1FP1；

步骤 7：设置 ATIM\_SMCR.SMS 为 0x7，使 ATIM 工作于外部时钟模式 1；

步骤 8：设置 ATIM\_IER.UIE 为 1 并配置对应 NVIC，使能更新中断；

步骤 9：配置计数器预分频器 ATIM\_PSC；

步骤 10：设置期望的重载值 ATIM\_ARR；

步骤 11：设置 ATIM\_CR1.CEN 为 1，使能计数器；

步骤 12：当计数器溢出时，ATIM\_ISR.UIF 标志位置 1，进入中断服务程序，设置 ATIM\_ICR.UIF 为 0 清除该中断标志。

### 13.3.18.2 外部时钟模式 2 编程示例

以下是外部时钟模式 2+ 触发模式的编程示例，ATIM\_CH1 输入出现上升沿时触发启动计数器，计数器在 ATIM\_ETR 输入信号的每个上升沿计数，步骤如下：

步骤 1：设置 SYSCTRL\_AHBEN.GPIOx 为 1，SYSCTRL\_APBEN1.ATIM 为 1，打开 ATIM\_ETR、ATIM\_CH1 引脚对应的 GPIO 时钟和 ATIM 配置时钟及工作时钟；

步骤 2：将 ATIM\_ETR、ATIM\_CH1 引脚对应的 GPIO 配置成复用输入模式，具体寄存器配置请参见 8 通用输入输出端口（GPIO）；

步骤 3：设置 ATIM\_AF1.ETRSEL 为 0，选择 ETR 来源为 ATIM\_ETR 引脚；

步骤 4：设置 ATIM\_SMCR.ETP 为 0，选择 ETR 上升沿有效；

步骤 5：设置 ATIM\_SMCR.ETPS 为 0，关闭 ETR 的预分频器；

步骤 6：设置 ATIM\_SMCR.ETF 为 0，关闭 ETR 输入滤波；

步骤 7：设置 ATIM\_SMCR.ECE 为 1，使能外部时钟模式 2；

步骤 8：设置 ATIM\_TISEL1.TI1SEL 为 0，选择 TI1 来源为 ATIM\_CH1 通道；

步骤 9：设置 ATIM\_CCMR1CAP.IC1F 为 0，关闭 TI1 输入滤波；

步骤 10：设置 ATIM\_CCER.CC1NP 为 0、ATIM\_CCER.CC1P 为 0，选择检测 TI1 上升沿；

步骤 11：设置 ATIM\_SMCR.TS 为 0x5，选择 TRGI 输入为 TI1FP1；

步骤 12：设置 ATIM\_SMCR.SMS 为 0x6，使 ATIM 工作于触发模式；

步骤 13：当 ATIM\_CH1 输入出现上升沿时使能，计数器在 ATIM\_ETR 输入信号的每个上升沿计数。

### 13.3.18.3 复位模式编程示例

以下示例中，ATIM\_CH1 输入出现上升沿时重新初始化计数器，步骤如下：

步骤 1：设置 SYSCTRL\_AHBEN.GPIOx 为 1，SYSCTRL\_APBEN1.ATIM 为 1，打开 ATIM\_CH1 引脚对应的 GPIO 时钟和 ATIM 配置时钟及工作时钟；

步骤 2：将 ATIM\_CH1 引脚对应的 GPIO 配置成复用输入模式，具体寄存器配置请参见通用输入输

出端口 (GPIO);

- 步骤 3: 设置 ATIM\_TISEL1.TI1SEL 为 0, 选择 TI1 来源为 ATIM\_CH1 通道;
- 步骤 4: 设置 ATIM\_CCMR1CAP.IC1F, 配置输入滤波带宽;
- 步骤 5: 设置 ATIM\_CCER.CC1NP 为 0、ATIM\_CCER.CC1P 为 0, 选择检测上升沿;
- 步骤 6: 设置 ATIM\_SMCR.TS 为 0x5, 选择 TRGI 输入为 TI1FP1;
- 步骤 7: 设置 ATIM\_SMCR.SMS 为 0x4, 使 ATIM 工作于复位模式;
- 步骤 8: 设置期望的重载值 ATIM\_ARR;
- 步骤 9: 设置 ATIM\_CR1.CEN 为 1, 使能计数器, 开始正常计数;
- 步骤 10: 当 ATIM\_CH1 输入出现上升沿时, 计数器清零, 重新从 0 开始计数。

#### 13.3.18.4 门控模式编程示例

以下示例中, ATIM\_CH1 引脚输入的信号作为门控信号, 控制计数器计数, 步骤如下:

- 步骤 1: 设置 SYSCTRL\_AHBEN.GPIOx 为 1, SYSCTRL\_APBEN1.ATIM 为 1, 打开 ATIM\_CH1 引脚对应的 GPIO 时钟和 ATIM 配置时钟及工作时钟;
- 步骤 2: 将 ATIM\_CH1 引脚对应的 GPIO 配置成复用输入模式, 具体寄存器配置请参见通用输入输出端口 (GPIO);
- 步骤 3: 设置 ATIM\_TISEL1.TI1SEL 为 0, 选择 TI1 来源为 ATIM\_CH1 通道;
- 步骤 4: 设置 ATIM\_CCMR1CAP.IC1F, 配置输入滤波带宽;
- 步骤 5: 设置 ATIM\_CCER.CC1NP 为 0、ATIM\_CCER.CC1P 为 0, 选择检测高电平;
- 步骤 6: 设置 ATIM\_SMCR.TS 为 0x5, 选择 TRGI 输入为 TI1FP1;
- 步骤 7: 设置 ATIM\_SMCR.SMS 为 0x5, 使 ATIM 工作于门控模式;
- 步骤 8: 设置期望的重载值 ATIM\_ARR;
- 步骤 9: 设置 ATIM\_CR1.CEN 为 1, 使能计数器;
- 步骤 10: 当 ATIM\_CH1 输入为高电平时, 计数器开始计数; 当 ATIM\_CH1 输入为低电平时, 计数器停止计数。

#### 13.3.18.5 触发模式编程示例

以下示例中, ATIM\_CH1 输入出现上升沿时触发启动计数器, 步骤如下:

- 步骤 1: 设置 SYSCTRL\_AHBEN.GPIOx 为 1, SYSCTRL\_APBEN1.ATIM 为 1, 打开 ATIM\_CH1 引脚对应的 GPIO 时钟和 ATIM 配置时钟及工作时钟;
- 步骤 2: 将 ATIM\_CH1 引脚对应的 GPIO 配置成复用输入模式, 具体寄存器配置请参见通用输入输出端口 (GPIO);
- 步骤 3: 设置 ATIM\_TISEL1.TI1SEL 为 0, 选择 TI1 来源为 ATIM\_CH1 通道;
- 步骤 4: 设置 ATIM\_CCMR1CAP.IC1F, 配置输入滤波带宽;
- 步骤 5: 设置 ATIM\_CCER.CC1NP 为 0、ATIM\_CCER.CC1P 为 0, 选择检测上升沿;
- 步骤 6: 设置 ATIM\_SMCR.TS 为 0x5, 选择 TRGI 输入为 TI1FP1;
- 步骤 7: 设置 ATIM\_SMCR.SMS 为 0x6, 使 ATIM 工作于触发模式;
- 步骤 8: 设置期望的重载值 ATIM\_ARR;
- 步骤 9: 当 ATIM\_CH1 输入出现上升沿时, 计数器启动计数, 同时 TIF 标志置 1。

#### 13.3.18.6 正交编码器模式编程示例

以下示例中, 配置 ATIM 工作在正交编码器模式 -x4 模式, 步骤如下:

- 步骤 1: 设置 SYSCTRL\_AHBEN.GPIOx 为 1, SYSCTRL\_APBEN1.ATIM 为 1, 打开 ATIM\_CH1、ATIM\_CH2 引脚对应的 GPIO 时钟和 ATIM 配置时钟及工作时钟;
- 步骤 2: 将 ATIM\_CH1 和 ATIM\_CH2 引脚对应的 GPIO 配置成复用输入模式, 具体寄存器配置请参见 8 通用输入输出端口 (GPIO);

- 步骤 3: 设置 ATIM\_TISEL1.TI1SEL 为 0, 选择 TI1 来源为 ATIM\_CH1 通道;
- 步骤 4: 设置 ATIM\_TISEL1.TI2SEL 为 0, 选择 TI2 来源为 ATIM\_CH2 通道;
- 步骤 5: 设置 ATIM\_CCMR1CAP.IC1F 和 ATIM\_CCMR1CAP.IC2F, 配置输入滤波带宽;
- 步骤 6: 设置 ATIM\_CCER.CC1NP、ATIM\_CCER.CC1P 为 0, CH1 输入不反相;
- 步骤 7: 设置 ATIM\_CCER.CC2NP、ATIM\_CCER.CC2P 为 0, CH2 输入不反相;
- 步骤 8: 设置 ATIM\_SMCR.SMS 为 0x3, 使 ATIM 工作于正交编码器模式 -x4 模式;
- 步骤 9: 设置合适的重载值 ATIM\_ARR;
- 步骤 10: 设置 ATIM\_CR1.CEN 为 1, 使能计数器。

### 13.3.18.7 输入捕获编程示例

#### 13.3.18.7.1 基本输入捕获模式

以下示例中, ATIM\_CH1 输入出现上升沿时执行一次输入捕获, 步骤如下:

步骤 1: 设置 SYSCTRL\_AHBEN.GPIOx 为 1, SYSCTRL\_APBEN1.ATIM 为 1, 打开 ATIM\_CH1 引脚对应的 GPIO 时钟和 ATIM 配置时钟及工作时钟;

步骤 2: 将 ATIM\_CH1 引脚对应的 GPIO 配置成复用输入模式, 具体寄存器配置请参见通用输入输出端口 (GPIO);

步骤 3: 设置 ATIM\_TISEL1.TI1SEL 为 0, 选择 TI1 来源为 ATIM\_CH1 通道;

步骤 4: 设置 ATIM\_CCMR1CAP.IC1F, 配置 TI1 输入滤波带宽;

步骤 5: 设置 ATIM\_CCER.CC1NP 为 0、ATIM\_CCER.CC1P 为 0, 选择检测 IC1 上升沿;

步骤 6: 设置 ATIM\_CCMR1CAP.CC1S 为 1, CC1 通道配置为输入, IC1 映射到 TI1 上;

步骤 7: 设置 ATIM\_CCMR1CAP.IC1PSC 为 0, 关闭 IC1 预分频器;

步骤 8: 设置 ATIM\_CCER.CC1E 为 1, 使能输入捕获 1 模式;

步骤 9: 设置 ATIM\_IER.CC1IE 为 1 并配置对应 NVIC, 使能输入捕获 1 中断;

步骤 10: 配置计数器预分频器 ATIM\_PSC;

步骤 11: 设置期望的重载值 ATIM\_ARR;

步骤 12: 设置 ATIM\_CR1.CEN 为 1, 使能计数器, 开始正常计数;

步骤 13: 当发生捕获时, 当前计数器 CNT 的值被锁存到捕获 / 比较寄存器 CCR1 中, 完成一次捕获, 同时

ATIM\_ISR.CC1IF 标志位置 1, 进入中断服务程序, 设置 ATIM\_ICR.CC1IF 为 0 清除该中断标志。

#### 13.3.18.7.2 PWM 输入模式

以下示例测量从 ATIM\_CH1 引脚输入的 PWM 信号的周期和占空比, 步骤如下:

步骤 1: 设置 SYSCTRL\_AHBEN.GPIOx 为 1, SYSCTRL\_APBEN1.ATIM 为 1, 打开 ATIM\_CH1 引脚对应的 GPIO 时钟和 ATIM 配置时钟及工作时钟;

步骤 2: 将 ATIM\_CH1 引脚对应的 GPIO 配置成复用输入模式, 具体寄存器配置请参见通用输入输出端口 (GPIO);

步骤 3: 设置 ATIM\_TISEL1.TI1SEL 为 0, 选择 TI1 来源为 ATIM\_CH1 通道;

步骤 4: 设置 ATIM\_CCMR1CAP.IC1F, 配置 TI1 输入滤波带宽;

步骤 5: 设置 ATIM\_CCER.CC1NP 为 0、ATIM\_CCER.CC1P 为 0, 选择检测 IC1 上升沿;

步骤 6: 设置 ATIM\_CCMR1CAP.CC1S 为 1, CC1 通道配置为输入, IC1 映射到 TI1 上;

步骤 7: 设置 ATIM\_CCMR1CAP.IC1PSC 为 0, 关闭 IC1 预分频器;

步骤 8: 设置 ATIM\_CCER.CC1E 为 1, 使能输入捕获 1 模式;

步骤 9: 设置 ATIM\_IER.CC1IE 为 1 并配置对应 NVIC, 使能输入捕获 1 中断;

步骤 10: 设置 ATIM\_CCER.CC2NP 为 0、ATIM\_CCER.CC2P 为 1, 选择检测 IC2 下降沿;

步骤 11: 设置 ATIM\_CCMR1CAP.CC2S 为 2, CC2 通道配置为输入, IC2 映射到 TI1 上;

步骤 12: 设置 ATIM\_CCMR1CAP.IC2PSC 为 0, 关闭 IC2 预分频器;



- 步骤 13: 设置 ATIM\_CCER.CC2E 为 1, 使能输入捕获 2 模式;
- 步骤 14: 设置 ATIM\_SMCR.TS 为 0x5, 选择 TRGI 输入为 TI1FP1;
- 步骤 15: 设置 ATIM\_SMCR.SMS 为 0x4, 使 ATIM 工作于复位模式;
- 步骤 16: 配置计数器预分频器 ATIM\_PSC; 步骤 17: 设置期望的重载值 ATIM\_ARR;
- 步骤 18: 设置 ATIM\_CR1.CEN 为 1, 使能计数器, 开始正常计数。

### 13.3.18.8 输出比较编程示例

以下示例中, 通道 CH1 对外输出设定的波形, 步骤如下:

步骤 1: 设置 SYSCTRL\_AHBEN.GPIOx 为 1, SYSCTRL\_APBEN1.ATIM 为 1, 打开 ATIM\_CH1 引脚对应的 GPIO 时钟和 ATIM 配置时钟及工作时钟;

步骤 2: 将 ATIM\_CH1 引脚对应的 GPIO 配置成复用输出模式, 具体寄存器配置请参见通用输入输出端口 (GPIO);

步骤 3: 设置 ATIM\_CCER.CC1P 为 0, 选择 OC1 输出有效极性;

步骤 4: 设置 ATIM\_CCMR1CMP.CC1S 为 0, CC1 通道配置为输出;

步骤 5: 设置 ATIM\_CCMR1CMP.OC1M, 选择输出比较 1 模式;

步骤 6: 设置 ATIM\_CCER.CC1E 为 1, 在相应输出引脚上输出 OC1 信号;

步骤 7: 设置 ATIM\_BDTR.MOE 为 1, 使能主输出;

步骤 8: 配置计数器预分频器 ATIM\_PSC; 步骤 9: 设置期望的重载值 ATIM\_ARR;

步骤 10: 设置期望的比较值 ATIM\_CCR1;

步骤 11: 设置 ATIM\_IER.CC1IE 为 1 并配置对应 NVIC, 使能比较 1 中断;

步骤 12: 设置 ATIM\_CR1.CEN 为 1, 使能计数器;

步骤 13: 当发生比较匹配时, ATIM\_ISR.CC1IF 标志位置 1, 进入中断服务程序, 设置 ATIM\_ICR.CC1IF 为 0 清除该中断标志。

## 13.4 寄存器列表

ATIM 基地址: ATIM\_BASE = 0x4001 0000

表 13-18 ATIM 寄存器列表

寄存器名称	寄存器地址	寄存器描述
ATIM_CR1	ATIM_BASE + 0x00	控制寄存器 1
ATIM_CR2	ATIM_BASE + 0x04	控制寄存器 2
ATIM_SMCR	ATIM_BASE + 0x08	从模式控制寄存器
ATIM_IER	ATIM_BASE + 0x0C	中断使能寄存器
ATIM_ISR	ATIM_BASE + 0x10	中断标志寄存器
ATIM_ICR	ATIM_BASE + 0x70	中断标志清除寄存器
ATIM_EGR	ATIM_BASE + 0x14	事件生成寄存器
ATIM_CCMR1CAP	ATIM_BASE + 0x18	捕获模式寄存器 1
ATIM_CCMR1CMP	ATIM_BASE + 0x18	比较模式寄存器 1
ATIM_CCMR2CAP	ATIM_BASE + 0x1C	捕获模式寄存器 2
ATIM_CCMR2CMP	ATIM_BASE + 0x1C	比较模式寄存器 2
ATIM_CCMR3CAP	ATIM_BASE + 0x50	捕获模式寄存器 3
ATIM_CCMR3CMP	ATIM_BASE + 0x50	比较模式寄存器 3
ATIM_CCER	ATIM_BASE + 0x20	捕获 / 比较使能寄存器
ATIM_CNT	ATIM_BASE + 0x24	计数寄存器
ATIM_PSC	ATIM_BASE + 0x28	预分频寄存器
ATIM_ARR	ATIM_BASE + 0x2C	自动重载寄存器
ATIM_RCR	ATIM_BASE + 0x30	重复计数寄存器
ATIM_CCR1	ATIM_BASE + 0x34	捕获 / 比较寄存器 1
ATIM_CCR2	ATIM_BASE + 0x38	捕获 / 比较寄存器 2
ATIM_CCR3	ATIM_BASE + 0x3C	捕获 / 比较寄存器 3
ATIM_CCR4	ATIM_BASE + 0x40	捕获 / 比较寄存器 4
ATIM_CCR5	ATIM_BASE + 0x48	捕获 / 比较寄存器 5
ATIM_CCR6	ATIM_BASE + 0x4C	捕获 / 比较寄存器 6
ATIM_BDTR	ATIM_BASE + 0x44	刹车和死区寄存器
ATIM_DTR2	ATIM_BASE + 0x54	死区时间寄存器 2
ATIM_ECR	ATIM_BASE + 0x58	编码控制寄存器
ATIM_TISEL1	ATIM_BASE + 0x5C	TI 输入选择寄存器 1
ATIM_TISEL2	ATIM_BASE + 0x6C	TI 输入选择寄存器 2
ATIM_AF1	ATIM_BASE + 0x60	复用功能选项寄存器 1
ATIM_AF2	ATIM_BASE + 0x64	复用功能选项寄存器 2



## 13.5 寄存器描述

### 13.5.1 ATIM 控制寄存器 1 (ATIM\_CR1)

偏移地址：0x00 复位值：0x0000 0000

位编号	位符号	权限	说明
31-12	RFU	-	保留位，请保持默认值
11	UIFREMAP	RW	UIF 状态位重映射 (UIF status bit remapping) 0：无重映射。UIF 状态位不复制到 ATIM_CNT 寄存器的位 31。 1：使能重映射。UIF 状态位复制到 ATIM_CNT 寄存器的位 31。
10	RFU	-	保留位，请保持默认值
9-8	CKD	RW	时钟分频 此位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器 (ETR、TlX) 所使用的死区及采样时钟 (tDTS) 之间的分频比 00: $t_{DTS}=t_{CK\_INT}$ 01: $t_{DTS}=2*t_{CK\_INT}$ 10: $t_{DTS}=4*t_{CK\_INT}$ 11: 保留，不要设置这个值
7	ARPE	RW	自动重载预装载使能 0: TIMx_ARR 寄存器不进行缓冲 1: TIMx_ARR 寄存器进行缓冲
6-5	CMS	RW	中心对齐模式选择 00: 边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。 01: 中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道 (TIMx_CCMRx 寄存器中的 CCxS=00) 的输出比较中断标志才置 1。 10: 中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道 (TIMx_CCMRx 寄存器中的 CCxS=00) 的输出比较中断标志才置 1。 11: 中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道 (TIMx_CCMRx 寄存器中的 CCxS=00) 的输出比较中断标志都会置 1。 注：只要计数器处于使能状态 (CEN=1)，就不得从边沿对齐模式切换为中心对齐模式。
4	DIR	RW	方向 0: 计数器递增计数 1: 计数器递减计数 注：当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。
3	OPM	RW	单脉冲模式 0: 计数器在发生更新事件时不会停止计数 1: 计数器在发生下一更新事件时停止计数 (将 CEN 位清零)
2	URS	RW	更新请求源 此位由软件置 1 和清零，用以选择 UEV 事件源。

			<p>0: 使能时, 所有以下事件都会产生更新中断请求。此类事件包括:</p> <ul style="list-style-type: none"> <li>- 计数器上溢/下溢</li> <li>- 将 UG 位置 1</li> <li>- 通过从模式控制器生成的更新事件</li> </ul> <p>1: 使能时, 只有计数器上溢/下溢会生成更新中断请求。</p>
1	UDIS	RW	<p>更新禁止</p> <p>此位由软件置 1 和清零, 用以使能/禁止 UEV 事件生成。</p> <p>0: 使能 UEV。更新 (UEV) 事件可通过以下事件之一生成:</p> <ul style="list-style-type: none"> <li>- 计数器上溢/下溢</li> <li>- 将 UG 位置 1</li> <li>- 通过从模式控制器生成的更新事件</li> </ul> <p>然后更新影子寄存器的值。</p> <p>1: 禁止 UEV。不会生成更新事件, 各影子寄存器的值 (ARR、PSC 和 CCRx) 保持不变。</p> <p>但如果将 UG 位置 1, 或者从从模式控制器接收到硬件复位, 则会重新初始化计数器和预分频器。</p>
0	CEN	RW	<p>计数器使能 (Counter enable)</p> <p>0: 禁止计数器</p> <p>1: 使能计数器</p> <p>注: 只有事先通过软件将 CEN 位置 1, 才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将 CEN 位置 1。</p>

### 13.5.2 ATIM 控制寄存器 2 (ATIM\_CR2)

偏移地址: 0x04      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-26	RFU	-	保留位, 请保持默认值
26-25	MMSH	RW	主模式选择 为 MMS 的最高两位, 需配合 MMS 使用
24-20	MMS2	RW	主模式选择 2 这些位可选择将发送到 ADC 以实现同步的信息 (TRGO2)。这些位的组合请参照 MMS 位域描述。
19	OIS6N	RW	输出空闲状态 6 (OC6N 输出) 请参见 OIS1N 位
18	OIS6	RW	输出空闲状态 6 (OC6 输出) 请参见 OIS1 位
17	OIS5N	RW	输出空闲状态 5 (OC5N 输出) 请参见 OIS1N 位
16	OIS5	RW	输出空闲状态 5 (OC5 输出) 请参见 OIS1 位
15	OIS4N	RW	输出空闲状态 4 (OC4N 输出) 请参见 OIS1N 位
14	OIS4	RW	输出空闲状态 4 (OC4 输出) 请参见 OIS1 位

13	OIS3N	RW	输出空闲状态 3 (OC3N 输出) 请参见 OIS1N 位
12	OIS3	RW	输出空闲状态 3 (OC3 输出) 请参见 OIS1 位
11	OIS2N	RW	输出空闲状态 2 (OC2N 输出) 请参见 OIS1N 位
10	OIS2	RW	输出空闲状态 2 (OC2 输出) 请参见 OIS1 位
9	OIS1N	RW	输出空闲状态 1 (OC1N 输出) 请参见 OIS1N 位
8	OIS1	RW	输出空闲状态 1 (OC1 输出) 请参见 OIS1 位
7	TI1S	RW	TI1 选择 0: TIMx_CH1 引脚连接到 TI1 输入 1: TIMx_CH1、CH2 和 CH3 引脚连接到 TI1 输入 (异或组合)
6-4	MMS	RW	<p>主模式选择</p> <p>这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。配合 MMSH 使用, 这些位的组合如下:</p> <p>00000: 复位——ATIM_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成 (从模式控制器配置为复位模式), 则 TRGO 上的信号相比实际复位会有延迟。</p> <p>00001: 使能——计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器, 或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑与运算组合而成。当计数器使能信号由触发输入控制时, TRGO 上会存在延迟, 选择主 / 从模式时除外 (请参见 14.9.3 ATIM_SMCR 从模式控制寄存器的 MSM 位的说明)。</p> <p>00010: 更新——选择更新事件作为触发输出 (TRGO)。例如, 主定时器可用作从定时器的预分频器。</p> <p>00011: 编码器计数时钟输出</p> <p>00100: CH1 输出比较或输入捕获脉冲 00101: CH2 输出比较或输入捕获脉冲 00110: CH3 输出比较或输入捕获脉冲 00111: CH4 输出比较或输入捕获脉冲 01000: CH5 输出比较或输入捕获脉冲 01001: CH6 输出比较或输入捕获脉冲 01010: OC1REF 信号 01011: OC2REF 信号 01100: OC3REF 信号 01101: OC4REF 信号 01110: OC5REF 信号 01111: OC6REF 信号 10000: OC1REFC 信号 10001: OC2REFC 信号 10010: OC3REFC 信号 10011: OC4REFC 信号 10100: OC5REFC 信号 10101: OC6REFC 信号</p>

			<p>10110: OC3REFC 上升沿或下降沿          10111: OC4REFC 上升沿或下降沿          11000: OC5REFC 上升沿或下降沿          11001: OC6REFC 上升沿或下降沿          11010: OC4REFC 上升沿或 OC5REFC 上升沿          11011: OC4REFC 上升沿或 OC5REFC 下降沿          11100: OC4REFC 上升沿或 OC6REFC 上升沿          11101: OC4REFC 上升沿或 OC6REFC 下降沿          11110: OC5REFC 上升沿或 OC6REFC 上升沿          11111: OC5REFC 上升沿或 OC6REFC 下降沿</p> <p>注: 必须先使能从定时器或 ADC 的时钟, 才能从主定时器 接收事件; 并且从主定时器接收触发信号时, 不得实时 更改从定时器或 ADC 的时钟。</p>
3	RFU	-	保留位, 请保持默认值
2	CCUS	RW	<p>捕获/比较控制更新选择</p> <p>0: 如果捕获/比较控制位进行预装载 (CCPC=1), 仅通过将 COMG 位置 1 来对这些位进行更新</p> <p>1: 如果捕获/比较控制位进行预装载 (CCPC=1), 可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新。</p> <p>注: 此位仅对具有互补输出的通道有效。</p>
1	RFU	-	保留位, 请保持默认值
0	CCPC	RW	<p>捕获/比较预装载控制</p> <p>0: CCyE、CCyNE 和 OCyM 位未进行预装载</p> <p>1: CCyE、CCyNE 和 OCyM 位进行了预装载, 写入这些位后, 仅当发生换向事件 (COM) (COMG 位置 1 或在 TRGI 上检测到上升沿, 取决于 CCUS 位) 时才会对这些位进行更新。</p> <p>注: 此位仅对具有互补输出的通道有效。</p>

### 13.5.3 ATIM 从模式控制寄存器 (ATIM\_SMCR)

偏移地址: 0x08      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位, 请保持默认值
25	SMSPS	RW	预加载源 此位选择 SMS 位字段从预装载寄存器转移到有效寄存器的触发源 0: 传输由计时器的更新事件触发 1: 传输由索引事件触发
24	SMSPE	RW	预加载使能 此位选择 SMS[3:0] 位是否预装载 0: 禁止预装载 1: 使能预装载
23-22	RFU	-	保留位, 请保持默认值
21-20	TSH	RW	触发选择 为 TS 的最高位, 需配合 TS 使用
19-17	RFU	-	保留位, 请保持默认值
15	ETP	RW	外部触发极性 此位可选择将 ETR 还是 ETR 用于触发操作 0: ETR 未反相, 高电平或上升沿有效。 1: ETR 反相, 低电平或下降沿有效。
14	ECE	RW	外部时钟使能 此位可使能外部时钟模式 2。 0: 禁止外部时钟模式 2。 1: 使能外部时钟模式 2。计数器时钟由 ETRF 信号的任意有效边沿提供。 注: 1: 将 ECE 位置 1 与选择外部时钟模式 1 并将 TRGI 连接到 ETRF (SMS=111 且 TS=00111) 具有相同效果。 2: 外部时钟模式 2 可以和以下从模式同时使用: 复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF (TS 位不得为 00111)。 3: 如果同时使能外部时钟模式 1 和外部时钟模式 2, 则外部时钟输入为 ETRF。
13-12	ETPS	RW	外部触发预分频器 外部触发信号 ETRP 频率不得超过 TIMxCLK 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。 00: 预分频器关闭 01: 2 分频 ETRP 频率 10: 4 分频 ETRP 频率 11: 8 分频 ETRP 频率
11-8	ETF	RW	外部触发滤波器

			<p>此位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：</p> <p>0000: 无滤波器，按 fDTS 频率进行采样</p> <p>0001: <math>f_{\text{SAMPLING}}=f_{\text{CK\_INT}}</math>, N=2</p> <p>0010: <math>f_{\text{SAMPLING}}=f_{\text{CK\_INT}}</math>, N=4</p> <p>0011: <math>f_{\text{SAMPLING}}=f_{\text{CK\_INT}}</math>, N=8</p> <p>0100: <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/2</math>, N=6</p> <p>0101: <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/2</math>, N=8</p> <p>0110: <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/4</math>, N=6</p> <p>0111: <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/4</math>, N=8</p> <p>1000: <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/8</math>, N=6</p> <p>1001: <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/8</math>, N=8</p> <p>1010: <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, N=5</p> <p>1011: <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, N=6</p> <p>1100: <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/16</math>, N=8</p> <p>1101: <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, N=5</p> <p>1110: <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, N=6</p> <p>1111: <math>f_{\text{SAMPLING}}=f_{\text{DTS}}/32</math>, N=8</p>
7	MSM	RW	<p>主/从模式</p> <p>0: 不执行任何操作。</p> <p>1: 当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于由单个外部事件对多个定时器进行同步的情况。</p>
6-4	TS	RW	<p>TS[4:0]: 触发选择</p> <p>此位域可选择将要用于同步计数器的触发输入。</p> <p>00000: 内部触发 0 (ITR0)</p> <p>00001: 内部触发 1 (ITR1)</p> <p>00010: 内部触发 2 (ITR2)</p> <p>00011: 内部触发 3 (ITR3)</p> <p>00100: TI1 边沿检测器 (TI1F_ED)</p> <p>00101: 滤波后的定时器输入 1 (TI1FP1)</p> <p>00110: 滤波后的定时器输入 2 (TI2FP2)</p> <p>00111: 外部触发输入 (ETRF)</p> <p>01000: 内部触发 4 (ITR4)</p> <p>01001: 内部触发 5 (ITR5)</p> <p>01010: 内部触发 6 (ITR6)</p> <p>01011: 内部触发 7 (ITR7)</p> <p>01100: 内部触发 8 (ITR8)</p> <p>01101: 内部触发 9 (ITR9)</p> <p>01110: 内部触发 10 (ITR10)</p> <p>01111: 内部触发 11 (ITR11)</p> <p>10000: 内部触发 12 (ITR12)</p> <p>注 1: 这些位只能在未使用的情况下（例如，SMS=000 时）进行更</p>

			<p>改，以避免转换时出现错误的边沿检测。</p> <p>注 2： 其他位位于同一寄存器的位置 20， 21， 需配合 TS 使用</p>
3	OCCS	RW	<p>OCREF 清零选择 (OCREF clear selection)</p> <p>该位用于选择 OCREF 清零源。</p> <p>0: OCREF_CLR_INT 连接到 COMP1 或 COMP2 输出，具体取决于 ATIM_OR1.OCREF_CLR</p> <p>1: OCREF_CLR_INT 连接到 ETRF</p>
2-0	SMS	RW	<p>从模式选择</p> <p>选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入 上所选的极性相关（请参见输入相关控制寄存器说明）。</p> <p>0000: 禁止从模式——如果 CEN=“1”，预分频器时钟直接由内部时钟提供。</p> <p>0001: 正交编码器模式——×2 模式，计数器根据 TI2FP2 电平在 TI1FP1 边沿递增 / 递减计数。</p> <p>0010: 正交编码器模式——×2 模式，计数器根据 TI1FP1 电平在 TI2FP2 边沿递增 / 递减计数。</p> <p>0011: 正交编码器模式——×4 模式，计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个输入的电平。 0100: 复位模式——在出现所选触发输入 (TRGI) 上升沿时， 重新初始化计数器并生成一个寄存器更新事件。</p> <p>0101: 门控模式——触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。</p> <p>0110: 触发模式——触发信号 (TRGI) 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。</p> <p>0111: 外部时钟模式 1——由所选触发信号 (TRGI) 的上升沿提供计数器时钟。</p> <p>1000: 组合复位 + 触发模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。</p> <p>1001: 组合门控 + 复位模式——当触发输入 (TRGI) 为高电平时计数器被使能并开始计数，当触发输入变为低电平时计数器停止计数并被复位，在此模式计数器的启动和停止都被控制。</p> <p>1010: 编码模式——时钟 + 方向，×2 模式。</p> <p>1011: 编码模式——时钟 + 方向，×1 模式，TI2FP2 的边沿极性由 CC2P 设置。</p> <p>1100: 编码模式——带方向时钟，×2 模式。</p> <p>1101: 编码模式——带方向时钟，×1 模式，TI1FP1 和 TI2FP2 的边沿极性由 CC1P 和 CC2P 设置。</p> <p>1110: 正交编码模式——×1 模式，仅计数 TI1FP1 边沿，其边沿极性由 CC1P 设置。</p> <p>1111: 正交编码模式——×1 模式，仅计数 TI2FP2 边沿，其边沿极性由 CC2P 设置。</p> <p>注 1: 如果将 TI1F_ED 选作触发输入(TS=00100)，则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控</p>



			<p>模式检查的则是触发信号的 电平。</p> <p>注 2: 必须先使能接收 TRGO 或 TRGO2 信号的从外设(定 时器、ADC 等)的时钟, 才能从主定时器接收事件; 并且从主定时器接收触发信号时, 不得实时更改时钟 频率 (预分频器)。</p>
--	--	--	--

### 13.5.4 ATIM 中断使能寄存器 (ATIM\_IER)

偏移地址: 0x0C      复位值: 0x0000

位编号	位符号	权限	说明
31-15	RFU	-	保留位, 请保持默认值
23	TERRIE	RW	<p>编码器转换错误中断使能控制</p> <p>0: 禁止转换错误中断</p> <p>1: 使能转换错误中断</p>
22	IERRIE	RW	<p>编码器索引错误中断使能控制</p> <p>0: 禁止索引错误中断</p> <p>1: 使能索引错误中断</p>
21	DIRIE	RW	<p>编码器方向改变中断使能控制</p> <p>0: 禁止方向改变中断</p> <p>1: 使能方向改变中断</p>
20	IDXIE	RW	<p>编码器索引中断使能控制</p> <p>0: 禁止索引中断</p> <p>1: 使能索引中断</p>
19-18	RFU	-	保留位, 请保持默认值
17	CC6IE	RW	<p>捕获/比较 6 中断使能 (Capture/Compare 4 interrupt enable)</p> <p>0: 禁止 CC6 中断</p> <p>1: 使能 CC6 中断</p>
16	CC5IE	RW	<p>捕获/比较 5 中断使能 (Capture/Compare 4 interrupt enable)</p> <p>0: 禁止 CC5 中断</p> <p>1: 使能 CC5 中断</p>
15-8	RFU	-	保留位, 请保持默认值
7	BIE	RW	<p>刹车中断使能 (Break interrupt enable)</p> <p>0: 禁止刹车中断</p> <p>1: 使能刹车中断</p>
6	TIE	RW	<p>触发中断使能 (Trigger interrupt enable)</p> <p>0: 禁止触发中断</p> <p>1: 使能触发中断</p>
5	COMIE	RW	<p>COM 中断使能 (COM interrupt enable)</p> <p>0: 禁止 COM 中断</p> <p>1: 使能 COM 中断</p>
4	CC4IE	RW	<p>捕获/比较 4 中断使能 (Capture/Compare 4 interrupt enable)</p> <p>0: 禁止 CC4 中断</p> <p>1: 使能 CC4 中断</p>



3	CC3IE	RW	捕获/比较 3 中断使能 (Capture/Compare 3 interrupt enable) 0: 禁止 CC3 中断 1: 使能 CC3 中断
2	CC2IE	RW	捕获/比较 2 中断使能 (Capture/Compare 2 interrupt enable) 0: 禁止 CC2 中断 1: 使能 CC2 中断
1	CC1IE	RW	捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable) 0: 禁止 CC1 中断 1: 使能 CC1 中断
0	UIE	RW	更新中断使能 (Update interrupt enable) 0: 禁止更新中断 1: 使能更新中断

### 13.5.5 ATIM 状态寄存器 (ATIM\_ISR)

偏移地址: 0x10      复位值: 0x0000

位编号	位符号	权限	说明
31-24	RFU	-	保留位, 请保持默认值
13	TERRF	RW	编码器转换错误中断标志 0: 未发生转换错误 1: 已发生转换错误
22	IERRF	RW	编码器索引错误中断标志 0: 未发生索引错误 1: 已发生索引错误
21	DIRF	RW	编码器方向改变中断标志 0: 未发生编码方向改变 1: 已发生编码方向改变
20	IDXF	RW	编码器索引中断标志 0: 未发生索引标志 1: 已发生索引标志
19	CC6OF	RW	捕获/比较 6 重复捕获标志 请参见 CC1OF 说明
18	CC5OF	RW	捕获/比较 5 重复捕获标志 请参见 CC1OF 说明
17	CC6IF	RW	捕获/比较 6 中断标志 (Capture/Compare 4 interrupt flag) 请参见 CC1IF 说明
16	CC5IF	RW	捕获/比较 5 中断标志 (Capture/Compare 4 interrupt flag) 请参见 CC1IF 说明
15-14	RFU	-	保留位, 请保持默认值
13	SBIF	RW	系统刹车中断标志 只要系统刹车输入变为有效状态, 此标志便由硬件置 1。系统刹车输入无效后可通过软件对其清零。 此标志必须复位以使 PWM 重新开始工作。

			<p>0: 未发生刹车事件。</p> <p>1: 在系统刹车输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1, 则会生成中断。</p>
12	CC4OF	RW	<p>捕获/比较 4 重复捕获标志</p> <p>请参见 CC1OF 说明</p>
11	CC3OF	RW	<p>捕获/比较 3 重复捕获标志</p> <p>请参见 CC1OF 说明</p>
10	CC2OF	RW	<p>捕获/比较 2 重复捕获标志</p> <p>请参见 CC1OF 说明</p>
9	CC1OF	RW	<p>捕获/比较 1 重复捕获标志</p> <p>仅当将相应通道配置为输入捕获模式时, 此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。</p> <p>0: 未检测到重复捕获。</p> <p>1: TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。</p>
8	B2IF	RW	<p>刹车 2 中断标志 (Break 2 interrupt flag)</p> <p>只要刹车 2 输入变为有效状态, 此标志便由硬件置 1。刹车 2 输入无效后可通过软件对其清零。</p> <p>0: 未发生刹车事件。</p> <p>1: 在刹车 2 输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1, 则会生成中断。</p>
7	BIF	RW	<p>刹车中断标志 (Break interrupt flag)</p> <p>只要刹车输入变为有效状态, 此标志便由硬件置 1。刹车输入无效后可通过软件对其清零。</p> <p>0: 未发生刹车事件。</p> <p>1: 在刹车输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1, 则会生成中断。</p>
6	TIF	RW	<p>触发中断标志 (Trigger interrupt flag)</p> <p>在除门控模式以外的所有模式下, 当使能从模式控制器后在 TRGI 输入上检测到有效边沿时, 该标志将由硬件置 1。选择门控模式时, 该标志将在计数器启动或停止时置 1。但需要通过软件清零。</p> <p>0: 未发生触发事件。</p> <p>1: 触发中断挂起。</p>
5	COMIF	RW	<p>COM 中断标志 (COM interrupt flag)</p> <p>此标志在发生 COM 事件时 (捕获/比较控制位 CCxE、CCxNE 和 OCxM 已更新时) 由硬件置 1。但需要通过软件清零。</p> <p>0: 未发生 COM 事件。</p> <p>1: COM 中断挂起。</p>
4	CC4IF	RW	<p>捕获/比较 4 中断标志 (Capture/Compare 4 interrupt flag)</p> <p>请参见 CC1IF 说明</p>
3	CC3IF	RW	<p>捕获/比较 3 中断标志 (Capture/Compare 3 interrupt flag)</p> <p>请参见 CC1IF 说明</p>
2	CC2IF	RW	<p>捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag)</p> <p>请参见 CC1IF 说明</p>

1	CC1IF	RW	<p>捕获/比较 1 中断标志 (Capture/Compare 1 interrupt flag)</p> <p>如果通道 CC1 配置为输出：当计数器与比较值匹配时，此标志由硬件置 1，中心对齐模式下除外（请参见 TIMx_CR1 寄存器中的 CMS 位说明）。但需要通过软件清零。</p> <p>0：不匹配。</p> <p>1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高。</p> <p>如果通道 CC1 配置为输入：此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将该位清零。</p> <p>0：未发生输入捕获事件</p> <p>1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）</p>
0	UIF	RW	<p>更新中断标志 (Update interrupt flag)</p> <p>该位在发生更新事件时通过硬件置 1。但需要通过软件清零。</p> <p>0：未发生更新。</p> <p>1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：</p> <ul style="list-style-type: none"> <li>- TIMx_CR1 寄存器中的 UDIS=0，并且重复计数器值上溢或下溢时（重复计数器 = 0 时更新）。</li> <li>- TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。</li> <li>- TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时</li> </ul>

### 13.5.6 ATIM 中断标志清除寄存器 (ATIM\_ICR)

偏移地址：0x70      复位值：0x00FF 3FFF

位编号	位符号	权限	说明
31-24	RFU	-	保留位，请保持默认值
13	TERRF	RW	<p>编码器转换错误中断标志清除</p> <p>0：清除编码器转换错误中断标志</p> <p>1：无功能</p>
22	IERRF	RW	<p>编码器索引错误中断标志清除</p> <p>0：清除编码器索引错误中断标志</p> <p>1：无功能</p>
21	DIRF	RW	<p>编码器方向改变中断标志清除</p> <p>0：清除编码器方向改变中断标志</p> <p>1：无功能</p>
20	IDXF	RW	<p>编码器索引中断标志清除</p> <p>0：清除编码器索引中断标志</p> <p>1：无功能</p>
19	CC6OF	RW	捕获 / 比较 6 重复捕获标志清除，请参见 CC1OF 说明
18	CC5OF	RW	捕获 / 比较 5 重复捕获标志清除，请参见 CC1OF 说明

17	CC6IF	RW	捕获 / 比较 6 中断标志清除, 请参见 CC1IF 说明
16	CC5IF	RW	捕获 / 比较 5 中断标志清除, 请参见 CC1IF 说明
15-14	RFU	-	保留位, 请保持默认值
13	SBIF	RW	系统刹车中断标志清除 0: 清除系统刹车中断标志 1: 无功能
12	CC4OF	RW	捕获 / 比较 4 重复捕获标志清除, 请参见 CC1OF 说明
11	CC3OF	RW	捕获 / 比较 3 重复捕获标志清除, 请参见 CC1OF 说明
10	CC2OF	RW	捕获 / 比较 2 重复捕获标志清除, 请参见 CC1OF 说明
9	CC1OF	RW	捕获 / 比较 1 重复捕获标志清除 0: 清除捕获 / 比较 1 的重复捕获标志 1: 无功能
8	B2IF	RW	刹车 2 中断标志清除 0: 清除刹车 2 中断标志 1: 无功能
7	BIF	RW	刹车中断标志清除 0: 清除刹车中断标志 1: 无功能。
6	TIF	RW	触发中断标志清除 0: 清除触发中断标志 1: 无功能
5	COMIF	RW	COM 中断标志清除 0: 清除 COM 中断标志 1: 无功能
4	CC4IF	RW	捕获 / 比较 4 中断标志清除, 请参见 CC1IF 说明
3	CC3IF	RW	捕获 / 比较 3 中断标志清除, 请参见 CC1IF 说明
2	CC2IF	RW	捕获 / 比较 2 中断标志清除, 请参见 CC1IF 说明
1	CC1IF	RW	捕获 / 比较 1 中断标志清除 0: 清除捕获 / 比较 1 中断标志 1: 无功能
0	UIF	RW	更新中断标志清除 0: 清除更新中断标志 1: 无功能

### 13.5.7 ATIM 事件生成寄存器 (ATIM\_EGR)

偏移地址: 0x14 复位值: 0x0000

位编号	位符号	权限	说明
31-9	RFU	-	保留位, 请保持默认值
17	CC6G	RW	捕获/ 比较 6 生成, 请参见 CC1G 说明
16	CC5G	RW	捕获/ 比较 5 生成, 请参见 CC1G 说明
15-9	RFU	-	保留位, 请保持默认值
8	B2G	W	刹车 2 生成 (Break 2 generation) 此位由软件置 1 以生成事件, 并由硬件自动清零。 0: 不执行任何操作。 1: 生成刹车 2 事件。MOE 位清零且 B2IF 标志置 1。使能后可发生相关中断。
7	BG	W	刹车生成 (Break generation) 此位由软件置 1 以生成事件, 并由硬件自动清零。

			<p>0: 不执行任何操作。</p> <p>1: 生成刹车事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。</p>
6	TG	W	<p>触发生成 (Trigger generation)</p> <p>此位由软件置 1 以生成事件，并由硬件自动清零。</p> <p>0: 不执行任何操作。</p> <p>1: TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。</p>
5	COMG	W	<p>捕获/比较控制更新生成 (Capture/Compare control update generation)</p> <p>该位可通过软件置 1，并由硬件自动清零</p> <p>0: 不执行任何操作。</p> <p>1: CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位。</p> <p>注： 此位仅对具有互补输出的通道有效。</p>
4	CC4G	W	<p>捕获/比较 4 生成 (Capture/Compare 4 generation)</p> <p>请参见 CC1G 说明</p>
3	CC3G	W	<p>捕获/比较 3 生成 (Capture/Compare 3 generation)</p> <p>请参见 CC1G 说明</p>
2	CC2G	W	<p>捕获/比较 2 生成 (Capture/compare 2 generation)</p> <p>请参见 CC1G 说明</p>
1	CC1G	W	<p>捕获/比较 1 生成 (Capture/Compare 1 generation)</p> <p>此位由软件置 1 以生成事件，并由硬件自动清零。</p> <p>0: 不执行任何操作</p> <p>1: 通道 1 上生成捕获/比较事件：</p> <p>如果通道 CC1 配置为输出：</p> <p>使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。</p> <p>如果通道 CC1 配置为输入：</p> <p>TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。</p>
0	UG	W	<p>更新生成 (Update generation)</p> <p>该位可通过软件置 1，并由硬件自动清零。</p> <p>0: 不执行任何操作。</p> <p>1: 重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值 (TIMx_ARR)。</p>

### 13.5.8 ATIM 捕获模式寄存器 1 (ATIM\_CCMR1CAP)

偏移地址：0x18      复位值：0x0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获和输出比较模式下的功能均不同。可同时将两条通道分别用于不同模式（例如，通道 1 用于输入捕获模式，通道 2 用于输出比较模式）。

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-12	IC2F	RW	输入捕获 2 滤波器 请参见 IC1F[3:0] 说明。
11-10	IC2PSC	RW	输入捕获 2 预分频器 请参见 OC1PSC[1:0] 说明。
9-8	CC2S	RW	捕获/比较 2 选择 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC2 通道配置为输出 01: CC2 通道配置为输入，IC2 映射到 TI2 上 10: CC2 通道配置为输入，IC2 映射到 TI1 上 11: CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注：仅当通道关闭时（TIMx_CCER 中的 CC2E =“0”），才可向 CC2S 位写入数据。
7-4	IC1F	RW	输入捕获 1 滤波器 此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器的采样长度。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿： 0000: 无滤波器，按 fDTS 频率进行采样 0001: fSAMPLING=fCK_INT, N=2 0010: fSAMPLING=fCK_INT, N=4 0011: fSAMPLING=fCK_INT, N=8 0100: fSAMPLING=fDTS/2, N=6 0101: fSAMPLING=fDTS/2, N=8 0110: fSAMPLING=fDTS/4, N=6 0111: fSAMPLING=fDTS/4, N=8 1000: fSAMPLING=fDTS/8, N=6 1001: fSAMPLING=fDTS/8, N=8 1010: fSAMPLING=fDTS/16, N=5 1011: fSAMPLING=fDTS/16, N=6 1100: fSAMPLING=fDTS/16, N=8 1101: fSAMPLING=fDTS/32, N=5 1110: fSAMPLING=fDTS/32, N=6 1111: fSAMPLING=fDTS/32, N=8
3-2	IC1PSC	RW	输入捕获 1 预分频器 此位域定义 CC1 输入 (IC1) 的预分频比。只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。



			00: 无预分频器, 捕获输入上每检测到一个边沿便执行捕获 01: 每发生 2 个事件便执行一次捕获 10: 每发生 4 个事件便执行一次捕获 11: 每发生 8 个事件便执行一次捕获
1-0	CC1S	RW	捕获/比较 1 选择 此位域定义通道方向 (输入/输出) 以及所使用的输入。 00: CC1 通道配置为输出 01: CC1 通道配置为输入, IC1 映射到 TI1 上 10: CC1 通道配置为输入, IC1 映射到 TI2 上 11: CC1 通道配置为输入, IC1 映射到 TRC 上。此模式仅在通过 TS 位 (TIMx_SMCR 寄存器) 选择内部触发输入时有效 注: 仅当通道关闭时 (TIMx_CCER 中的 CC1E =“0”), 才可向 CC1S 位写入数据。

### 13.5.9 ATIM 比较模式寄存器 1 (ATIM\_CCMR1CMP)

偏移地址: 0x18      复位值: 0x0000 0000

同一寄存器可用于输出比较模式 (本节) 或输入捕获模式 (上一节)。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获和输出比较模式下的功能均不同。可同时将两条通道分别用于不同模式 (例如, 通道 1 用于输入捕获模式, 通道 2 用于输出比较模式)。

输出比较模式:

位编号	位符号	权限	说明
31-25	RFU	-	保留位, 请保持默认值
24	OC2MH	RW	输出比较 2 模式——位 3 (Output Compare 2 mode - bit 3) 请参见 OC2M 说明——位 14:12
23-17	RFU	-	保留位, 请保持默认值
16	OC1MH	RW	输出比较 1 模式——位 3 (Output Compare 1 mode - bit 3) 请参见 OC1M 说明——位 6:4
15	OC2CE	RW	输出比较 2 清零使能 (Output Compare 2 clear enable) 请参见 OC1CE 说明。
14-12	OC2M	RW	输出比较 2 模式 (Output Compare 2 mode) 请参见 OC1M[2:0] 说明。
11	OC2PE	RW	输出比较 2 预装载使能 (Output Compare 2 preload enable) 请参见 OC1PE 说明。
10	OC2FE	RW	输出比较 2 快速使能 (Output Compare 2 fast enable) 请参见 OC1FE 说明。
9-8	CC2S	RW	捕获/比较 2 选择 (Capture/Compare 2 selection) 此位域定义通道方向 (输入/输出) 以及所使用的输入。 00: CC2 通道配置为输出 01: CC2 通道配置为输入, IC2 映射到 TI2 上 10: CC2 通道配置为输入, IC2 映射到 TI1 上 11: CC2 通道配置为输入, IC2 映射到 TRC 上。此模式仅在通过 TS

			<p>位 (TIMx_SMCR 寄存器) 选择内部触发输入时有效</p> <p>注: 仅当通道关闭时 (TIMx_CCER 中的 CC2E =“0”), 才可向 CC2S 位写入数据。</p>
7	OC1CE	RW	<p>输出比较 1 清零使能</p> <p>0: OC1Ref 不受 ocref_clr_int 信号影响</p> <p>1: ocref_clr_int 信号 (OCREF_CLR 输入或 ETRF 输入) 上检测到高电平时, OC1Ref 立即清零</p>
6-4	OC1M	RW	<p>输出比较 1 模式 (Output Compare 1 mode)</p> <p>这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效, 而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。</p> <p>0000: 冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。(该模式用于生成时基)。</p> <p>0001: 将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1(TIMx_CCR1) 匹配时, OC1REF 信号强制变为高电平。</p> <p>0010: 将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1(TIMx_CCR1) 匹配时, OC1REF 信号强制变为低电平。</p> <p>0011: 翻转——TIMx_CNT=TIMx_CCR1 时, OC1REF 发生翻转。</p> <p>0100: 强制变为无效电平——OC1REF 强制变为低电平。</p> <p>0101: 强制变为有效电平——OC1REF 强制变为高电平。</p> <p>0110 : PWM 模式 1——在递增计数模式下, 只要 TIMx_CNT&lt;TIMx_CCR1, 通道 1 便为有效状态, 否则为无效状态。在递减计数模式下, 只要 TIMx_CNT&gt;TIMx_CCR1, 通道 1 便为无效状态 (OC1REF=“0”), 否则为有效状态 (OC1REF=“1”)。</p> <p>0111 : PWM 模式 2——在递增计数模式下, 只要 TIMx_CNT&lt;TIMx_CCR1, 通道 1 便为无效状态, 否则为有效状态。在递减计数模式下, 只要 TIMx_CNT&gt;TIMx_CCR1, 通道 1 便为有效状态, 否则为无效状态。</p> <p>1000: 可再触发 OPM 模式 1——在递增计数模式下, 通道为有效状态, 直至 (在 TRGI 信号上) 检测到触发事件。然后, 在 PWM 模式 1 下进行比较, 通道会在下一次更新时再次变为有效状态。在递减计数模式下, 通道为无效状态, 直至 (在 TRGI 信号上) 检测到触发事件。然后, 在 PWM 模式 1 下进行比较, 通道会在下一次更新时再次变为无效状态。</p> <p>1001: 可再触发 OPM 模式 2——在递增计数模式下, 通道为无效状态, 直至 (在 TRGI 信号上) 检测到触发事件。然后, 在 PWM 模式 2 下进行比较, 通道会在下一次更新时再次变为无效状态。在递减计数模式下, 通道为有效状态, 直至 (在 TRGI 信号上) 检测到触发事件。然后, 在 PWM 模式 1 下进行比较, 通道会在下一次更新时再次变为有效状态。</p> <p>1010: 保留。</p> <p>1011: 保留。</p>



			<p>1100: 组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。</p> <p>1101: 组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。</p> <p>1110: 不对称 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时, OC1REFC 输出 OC1REF; 计数器递减计数时, OC1REFC 输出 OC2REF。</p> <p>1111: 不对称 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时, OC1REFC 输出 OC1REF; 计数器递减计数时, OC1REFC 输出 OC2REF。</p> <p>注: 只要编程了 LOCK (TIMx_BDTR 寄存器中的 LOCK 位) 级别 3 且 CC1S=“00” (通道配置为输出), 这些位即无法修改。</p> <p>注: 在 PWM 模式下, 仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时, OCREF 电平才会发生更改。</p> <p>注: OCxREFC 输出优先级依次是: 组合 PWM 模式&gt;CHx/CH5 组合波形配置&gt;CHx/CH6 组合波形配置</p> <p>注: 此位域将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1, 则仅当生成 COM 事件时, OC1M 有效位才会从预装载位获取新值。</p>
3	OC1PE	RW	<p>输出比较 1 预装载使能 (Output Compare 1 preload enable)</p> <p>0: 禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据, 写入后将立即使用新值。</p> <p>1: 使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到当前寄存器中。</p> <p>注: 1: 只要编程了 LOCK (TIMx_BDTR 寄存器中的 LOCK 位) 级别 3 且 CC1S=“00” (通道配置为输出), 这些位即无法修改。</p> <p>2: 只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式 (TIMx_CR1 寄存器中的 OPM 位置 1)。其他情况下则无法保证该行为。</p>
2	OC1FE	RW	<p>输出比较 1 快速使能 (Output Compare 1 fast enable)</p> <p>此位用于加快触发输入事件对 CC 输出的影响。</p> <p>0: 即使触发开启, CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时, 激活 CC1 输出的最短延迟时间为 5 个时钟周期。</p> <p>1: 触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后, 无论比较结果如何, OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时, OCFE 才会起作用。</p>
1-0	CC1S	RW	<p>捕获/比较 1 选择 (Capture/Compare 1 selection)</p> <p>此位域定义通道方向 (输入/输出) 以及所使用的输入。</p> <p>00: CC1 通道配置为输出</p> <p>01: CC1 通道配置为输入, IC1 映射到 TI1 上</p> <p>10: CC1 通道配置为输入, IC1 映射到 TI2 上</p>

			11: CC1 通道配置为输入, IC1 映射到 TRC 上。此模式仅在通过 TS 位 (TIMx_SMCR 寄存器) 选择内部触发输入时有效 注: 仅当通道关闭时 (TIMx_CCER 中的 CC1E =“0”), 才可向 CC1S 位写入数据。
--	--	--	---

### 13.5.10 ATIM 捕获模式寄存器 2 (ATIM\_CCMR2CAP)

偏移地址: 0x1C      复位值: 0x0000 0000

同一寄存器可用于输入捕获模式 (本节) 或输出比较模式 (下一节)。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获和输出比较模式下的功能均不同。可同时将两条通道分别用于不同模式 (例如, 通道 1 用于输入捕获模式, 通道 2 用于输出比较模式)。

输入捕获模式:

位编号	位符号	权限	说明
31-16	RFU	-	保留位, 请保持默认值
15-12	IC4F	RW	输入捕获 4 滤波器 (Input capture 4 filter) 请参见 IC1F[3:0] 说明。
11-10	IC4PSC	RW	输入捕获 4 预分频器 (Input capture 4 prescaler) 请参见 IC1PSC[1:0] 说明。
9-8	CC4S	RW	捕获/比较 4 选择 (Capture/Compare 4 selection) 此位域定义通道方向 (输入/输出) 以及所使用的输入。 00: CC4 通道配置为输出 01: CC4 通道配置为输入, IC4 映射到 TI4 上 10: CC4 通道配置为输入, IC4 映射到 TI3 上 11: CC4 通道配置为输入, IC4 映射到 TRC 上。此模式仅在通过 TS 位 (TIMx_SMCR 寄存器) 选择内部触发输入时有效 注: 仅当通道关闭时 (TIMx_CCER 中的 CC4E =“0”), 才可向 CC4S 位写入数据。
7-4	IC3F	RW	输入捕获 3 滤波器 (Input capture 3 filter) 请参见 IC1F[3:0] 说明。
3-2	IC3PSC	RW	输入捕获 3 预分频器 (Input capture 3 prescaler) 请参见 IC1PSC[1:0] 说明。
1-0	CC3S	RW	捕获/比较 3 选择 (Capture/compare 3 selection) 此位域定义通道方向 (输入/输出) 以及所使用的输入。 00: CC3 通道配置为输出 01: CC3 通道配置为输入, IC3 映射到 TI3 上 10: CC3 通道配置为输入, IC3 映射到 TI4 上 11: CC3 通道配置为输入, IC3 映射到 TRC 上。此模式仅在通过 TS 位 (TIMx_SMCR 寄存器) 选择内部触发输入时有效 注: 仅当通道关闭时 (TIMx_CCER 中的 CC3E =“0”), 才可向 CC3S 位写入数据。

### 13.5.11 ATIM 比较模式寄存器 2(ATIM\_CCMR2CMP)

偏移地址：0x1C 复位值：0x0000 0000

同一寄存器可用于输入捕获模式（上一节）或输出比较模式（本节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获和输出比较模式下的功能均不同。可同时将两条通道分别用于不同模式（例如，通道 1 用于输入捕获模式，通道 2 用于输出比较模式）。

输入比较模式：

位编号	位符号	权限	说明
31-25	RFU	-	保留位，请保持默认值
24	OC4MH	RW	输出比较 3 模式——位 3 (Output Compare 3 mode - bit 3) 请参见 OC1M 说明。
23-17	RFU	-	保留位，请保持默认值
16	OC3MH	RW	输出比较 3 模式——位 3 (Output Compare 3 mode - bit 3) 请参见 OC1M 说明。
15	OC4CE	RW	输出比较 4 清零使能 (Output compare 4 clear enable) 请参见 OC1CE 说明。
14-12	OC4M	RW	输出比较 4 模式 (Output compare 4 mode) 请参见 OC4M 说明。
11	OC4PE	RW	输出比较 4 预装载使能 (Output compare 4 preload enable) 请参见 OC1PE 说明。
10	OC4FE	RW	输出比较 4 快速使能 (Output compare 4 fast enable) 请参见 OC1FE 说明。
9-8	CC4S	RW	捕获/比较 4 选择 (Capture/Compare 4 selection) 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC4 通道配置为输出 01: CC4 通道配置为输入，IC4 映射到 TI4 上 10: CC4 通道配置为输入，IC4 映射到 TI3 上 11: CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注：仅当通道关闭时（TIMx_CCER 中的 CC4E =“0”），才可向 CC4S 位写入数据。
7	OC3CE	RW	输出比较 3 清零使能 (Output compare 3 clear enable) 请参见 OC1CE 说明。
6-4	OC3M	RW	输出比较 3 模式 (Output compare 3 mode) 请参见 OC1M 说明。
3	OC3PE	RW	输出比较 3 预装载使能 (Output compare 3 preload enable) 请参见 OC1PE 说明。
2	OC3FE	RW	输出比较 3 快速使能 (Output compare 3 fast enable) 请参见 OC1FE 说明。
1-0	CC3S	RW	捕获/比较 3 选择 (Capture/Compare 3 selection) 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC3 通道配置为输出 01: CC3 通道配置为输入，IC3 映射到 TI3 上

			<p>10: CC3 通道配置为输入, IC3 映射到 TI4 上</p> <p>11: CC3 通道配置为输入, IC3 映射到 TRC 上。此模式仅在通过 TS 位 (TIMx_SMCR 寄存器) 选择内部触发输入时有效</p> <p>注: 仅当通道关闭时 (TIMx_CCER 中的 CC3E =“0”), 才可向 CC3S 位写入数据。</p>
--	--	--	--

### 13.5.12 ATIM 捕获模式寄存器 3 (ATIM\_CCMR3CAP)

偏移地址: 0x50      复位值: 0x0000 0000

同一寄存器可用于输入捕获模式 (本节) 或输出比较模式 (下一节)。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获和输出比较模式下的功能均不同。可同时将两条通道分别用于不同模式 (例如, 通道 1 用于输入捕获模式, 通道 2 用于输出比较模式)。

输入捕获模式:

位编号	位符号	权限	说明
31-16	RFU	-	保留位, 请保持默认值
15-12	IC6F	RW	输入捕获 6 滤波器 (Input capture 6 filter) 请参见 IC1F[3:0] 说明。
11-10	IC6PSC	RW	输入捕获 6 预分频器 (Input capture 6 prescaler) 请参见 IC1PSC[1:0] 说明。
9-8	CC6S	RW	捕获/比较 6 选择 (Capture/Compare 6 selection) 此位域定义通道方向 (输入/输出) 以及所使用的输入。 00: CC6 通道配置为输出 01: CC6 通道配置为输入, IC6 映射到 TI6 上 10: CC6 通道配置为输入, IC6 映射到 TI5 上 11: CC6 通道配置为输入, IC6 映射到 TRC 上。此模式仅在通过 TS 位 (TIMx_SMCR 寄存器) 选择内部触发输入时有效 注: 仅当通道关闭时 (TIMx_CCER 中的 CC4E =“0”), 才可向 CC4S 位写入数据。
7-4	IC5F	RW	输入捕获 5 滤波器 (Input capture 5 filter) 请参见 IC1F[3:0] 说明。
3-2	IC5PSC	RW	输入捕获 3 预分频器 (Input capture 3 prescaler) 请参见 IC1PSC[1:0] 说明。
1-0	CC5S	RW	捕获/比较 5 选择 (Capture/compare 5 selection) 此位域定义通道方向 (输入/输出) 以及所使用的输入。 00: CC5 通道配置为输出 01: CC5 通道配置为输入, IC5 映射到 TI5 上 10: CC5 通道配置为输入, IC5 映射到 TI6 上 11: CC5 通道配置为输入, IC3 映射到 TRC 上。此模式仅在通过 TS 位 (TIMx_SMCR 寄存器) 选择内部触发输入时有效 注: 仅当通道关闭时 (TIMx_CCER 中的 CC3E =“0”), 才可向 CC3S 位写入数据。

### 13.5.13 ATIM 比较模式寄存器 3(ATIM\_CCMR3CMP)

偏移地址：0x50 复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获和输出比较模式下的功能均不同。可同时将两条通道分别用于不同模式（例如，通道 1 用于输入捕获模式，通道 2 用于输出比较模式）。

输入比较模式：

位编号	位符号	权限	说明
31-25	RFU	-	保留位，请保持默认值
24	OC6MH	RW	配合 OC6M 使用，为 OC6M 的最高位
23-17	RFU	-	保留位，请保持默认值
16	OC5MH	RW	配合 OC5M 使用，为 OC5M 的最高位
15	OC6CE	RW	输出比较 6 清零使能 (Output compare 6 clear enable) 请参见 OC1CE 说明。
14-12	OC6M	RW	输出比较 6 模式 (Output compare 6 mode) 请参见 OC1M 说明。
11	OC6PE	RW	输出比较 6 预装载使能 (Output compare 6 preload enable) 请参见 OC1PE 说明。
10	OC6FE	RW	输出比较 6 快速使能 (Output compare 6 fast enable) 请参见 OC1FE 说明。
9-8	CC6S	RW	捕获/比较 6 选择 (Capture/Compare 4 selection) 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC6 通道配置为输出 01: CC6 通道配置为输入，IC6 映射到 TI6 上 10: CC6 通道配置为输入，IC6 映射到 TI5 上 11: CC6 通道配置为输入，IC6 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注：仅当通道关闭时（TIMx_CCER 中的 CC6E =“0”），才可向 CC6S 位写入数据。
7	OC5CE	RW	输出比较 5 清零使能 (Output compare 5 clear enable) 请参见 OC1CE 说明。
6-4	OC5M	RW	输出比较 5 模式 (Output compare 5 mode) 请参见 OC1M 说明。
3	OC5PE	RW	输出比较 5 预装载使能 (Output compare 5 preload enable) 请参见 OC1PE 说明。
2	OC5FE	RW	输出比较 5 快速使能 (Output compare 5 fast enable) 请参见 OC1FE 说明。
1-0	CC5S	RW	捕获/比较 5 选择 (Capture/Compare 3 selection) 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC5 通道配置为输出 01: CC5 通道配置为输入，IC5 映射到 TI5 上 10: CC5 通道配置为输入，IC5 映射到 TI6 上 11: CC5 通道配置为输入，IC5 映射到 TRC 上。此模式仅在通过 TS

			位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注：仅当通道关闭时（TIMx_CCER 中的 CC5E =“0”），才可向 CC5S 位写入数据。
--	--	--	--

### 13.5.14 ATIM 捕获/比较使能寄存器 (ATIM\_CCER)

偏移地址：0x20      复位值：0x0000 0000

位编号	位符号	权限	说明
31-24	RFU	-	保留位，请保持默认值
23	CC6NP	RW	捕获/比较 6 互补输出极性 请参见 CC1NP 说明
22	CC6NE	RW	捕获/比较 6 互补输出使能 请参见 CC1NE 说明
21	CC6P	RW	捕获/比较 6 输出极性 请参见 CC1P 说明
20	CC6E	RW	捕获/比较 6 输出使能 请参见 CC1E 说明
19	CC5NP	RW	捕获/比较 5 互补输出极性 请参见 CC1NP 说明
18	CC5NE	RW	捕获/比较 5 互补输出使能 请参见 CC1NE 说明
17	CC5P	RW	捕获/比较 5 输出极性 请参见 CC1P 说明
16	CC5E	RW	捕获/比较 5 输出使能 请参见 CC1E 说明
15	CC4NP	RW	捕获/比较 4 互补输出极性 请参见 CC1NP 说明
14	CC4NE	RW	捕获/比较 4 互补输出使能 请参见 CC1NE 说明
13	CC4P	RW	捕获/比较 4 输出极性 请参见 CC1P 说明
12	CC4E	RW	捕获/比较 4 输出使能 请参见 CC1E 说明
11	CC3NP	RW	捕获/比较 3 互补输出极性 请参见 CC1NP 说明
10	CC3NE	RW	捕获/比较 3 互补输出使能 请参见 CC1NE 说明
9	CC3P	RW	捕获/比较 3 输出极性 请参见 CC1P 说明
8	CC3E	RW	捕获/比较 3 输出使能 请参见 CC1E 说明
7	CC2NP	RW	捕获/比较 2 互补输出极性 请参见 CC1NP 说明



6	CC2NE	RW	捕获/比较 2 互补输出使能 请参见 CC1NE 说明
5	CC2P	RW	捕获/比较 2 输出极性 请参见 CC1P 说明
4	CC2E	RW	捕获/比较 2 输出使能 请参见 CC1E 说明
3	CC1NP	RW	捕获/比较 1 互补输出极性 CC1 通道配置为输出： 0: OC1N 高电平有效。 1: OC1N 低电平有效。 CC1 通道配置为输入： 此位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1P 说明。 注 1：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3 且 CC1S="00"（通道配置为输出），此位立即变为不可写状态。 注 2：此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NP 有效位才会从预装载位获取新值。
2	CC1NE	RW	捕获/比较 1 互补输出使能 0: 关闭——OC1N 未激活。OC1N 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的函数。 1: 开启——在相应输出引脚上输出 OC1N 信号，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位。 注：此位将在具有互补输出的通道上进行预装载。如果 ATIM_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。

1	CC1P	RW	<p>捕获/比较 1 输出极性</p> <p>CC1 通道配置为输出：</p> <p>0：OC1 高电平有效</p> <p>1：OC1 低电平有效</p> <p>CC1 通道配置为输入：CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。</p> <p>00：非反相/上升沿触发。电路作用于 TIxFP1 的上升沿（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。</p> <p>01：反相/下降沿触发。电路作用于 TIxFP1 的下降沿（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。</p> <p>10：保留，不使用此配置。</p> <p>11：非反相/上升沿和下降沿均触发。电路作用于 TIxFP1 的上升沿和下降沿（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。</p> <p>注 1：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3，此位立即变为不可写状态。</p> <p>注 2：此位将在具有互补输出的通道上进行预装载。如果 ATIM_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。</p>
0	CC1E	RW	<p>捕获/比较 1 互补输出使能</p> <p>0：关闭——OC1N 未激活。OC1N 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的函数。</p> <p>1：开启——在相应输出引脚上输出 OC1N 信号，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位。</p> <p>注 1：此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NE 有效位才会从预装载位获取新值。</p> <p>注 2：使用输入捕获功能时，该位必须配 1。</p>



### 13.5.15 计数器 (ATIM\_CNT)

偏移地址: 0x24 复位值: 0x0000

位编号	位符号	权限	说明
31	UIFCPY	RW	根据 ATIM_CR1 寄存器中 UIFREMAP 位域的值, 本位域表示不同的含义: UIFREMAP = 0, 本位域保留, 读为 0 UIFREMAP = 1, 本位域表示 ATIM_ISR 寄存器的 UIF 位的只读副本
30-16	RFU	-	保留位, 请保持默认值
15-0	CNT	RW	计数器值

### 13.5.16 ATIM 预分频器 (ATIM\_PSC)

偏移地址: 0x28 复位值: 0x0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位, 请保持默认值
15-0	PSC	RW	预分频值 (Prescaler value) 计数器时钟频率 (CK_CNT) 等于 $f_{CK\_PSC} / (PSC[15:0] + 1)$ 。 PSC 包含每次发生更新事件时 (包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时, 或在配置为“复位模式”时通过触发控制器清零时) 要装载到活动预分频器寄存器的值。

### 13.5.17 ATIM 自动重载寄存器 (ATIM\_ARR)

偏移地址: 0x2C 复位值: 0xFFFF FFFF

位编号	位符号	权限	说明
31-16	RFU	-	保留位, 请保持默认值
15-0	ARR	RW	自动重载值 (Auto-reload value) ARR 为要装载到实际自动重载寄存器的值。 有关 ARR 更新和行为的更多详细信息, 请参见第 457 页的第 20.3.1 节: 时基单元。当自动重载值为空时, 计数器不工作。

### 13.5.18 ATIM 重复计数器 (ATIM\_RCR)

偏移地址: 0x30 复位值: 0xFFFF FFFF

位编号	位符号	权限	说明
31-16	RFU	-	保留位, 请保持默认值
15-0	REP	RW	重复计数器值 使能预装载寄存器时, 用户可通过这些位设置比较寄存器的更新频率 (即, 从预装载寄存器向活动寄存器周期性传输数据); 使能更新中断时, 也可设置更新中断的生成速率。 与 REP_CNT 相关的减计数器每次计数到 0 时, 都将生成一个更新

			<p>事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时, REP_CNT 才会重载 REP 值, 因此在生成下一重复更新事件之前, 无论向 TIMx_RCR 寄存器写入何值都无影响。</p> <p>这意味着 PWM 模式下 (REP+1) 相当于: 边沿对齐模式下的 PWM 周期数。 中心对齐模式下的 PWM 半周期数。</p>
--	--	--	---

### 13.5.19 ATIM 捕获/比较寄存器 1 (ATIM\_CCR1)

偏移地址: 0x34      复位值: 0xFFFF FFFF

位编号	位符号	权限	说明
31-16	RFU	-	保留位, 请保持默认值
15-0	CCR1	RW	<p>捕获/比较 1 值 (Capture/Compare 1 value)</p> <p>如果通道 CC1 配置为输出: CCR1 为要装载到有效捕获/比较 1 寄存器的值 (预装载值)。</p> <p>如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能, 则该值立刻生效;</p> <p>否则只在发生更新事件时生效 (拷贝到实际起作用的捕获/比较寄存器 1)。</p> <p>实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。</p> <p>如果通道 CC1 配置为输入: CR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。只能读取 TIMx_CCR1 寄存器, 无法对其进行编程。</p>

### 13.5.20 ATIM 捕获/比较寄存器 2 (ATIM\_CCR2)

偏移地址: 0x38      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位, 请保持默认值
15-0	CCR2	RW	<p>捕获/比较 2 值</p> <p>如果通道 CC2 配置为输出: CCR2 为要装载到有效捕获/比较 2 寄存器的值 (预装载值)。</p> <p>如果没有通过 TIMx_CCMR1 寄存器中的 OC2PE 位来使能预装载功能, 则该值立刻生效;</p> <p>否则只在发生更新事件时生效 (拷贝到实际起作用的捕获/比较寄存器 2)。</p> <p>有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。</p> <p>如果通道 CC2 配置为输入: CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。只能读取 TIMx_CCR2 寄存器, 无法对其进行编程。</p>

### 13.5.21 ATIM 捕获/比较寄存器 3 (ATIM\_CCR3)

偏移地址: 0x3C 复位值: 0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位, 请保持默认值
15-0	CCR3	RW	<p>捕获/比较值 (Capture/Compare value)</p> <p>如果通道 CC3 配置为输出: CCR3 为要装载到有效捕获/比较 3 寄存器的值 (预装载值)。</p> <p>如果没有通过 TIMx_CCMR2 寄存器中的 OC3PE 位来使能预装载功能, 则该值立刻生效;</p> <p>否则只在发生更新事件时生效 (拷贝到有效的捕获/比较寄存器 3)。</p> <p>有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC3 输出上发出信号的值。</p> <p>如果通道 CC3 配置为输入: CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。只能读取 TIMx_CCR3 寄存器, 无法对其进行编程。</p>

### 13.5.22 ATIM 捕获/比较寄存器 2 (ATIM\_CCR4)

偏移地址: 0x40 复位值: 0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位, 请保持默认值
15-0	CCR4	RW	<p>捕获/比较值 (Capture/Compare value)</p> <p>如果通道 CC4 配置为输出: CCR4 为要装载到有效捕获/比较 4 寄存器的值 (预装载值)。</p> <p>如果没有通过 TIMx_CCMR2 寄存器中的 OC4PE 位来使能预装载功能, 则该值立刻生效;</p> <p>否则只在发生更新事件时生效 (拷贝到有效的捕获/比较寄存器 4)。</p> <p>有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC4 输出上发出信号的值。</p> <p>如果通道 CC4 配置为输入: CCR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值。只能读取 TIMx_CCR4 寄存器, 无法对其进行编程。</p>

### 13.5.23 ATIM 捕获/比较寄存器 5 (ATIM\_CCR5)

偏移地址: 0x48 复位值: 0x0000 0000

位编号	位符号	权限	说明
31	GC5C6	RW	CH6/CH5 组合波形配置 参见 GC5C1 说明
30	GC5C5	RW	CH5/CH5 组合波形配置 参见 GC5C1 说明
29	GC5C4	RW	CH4/CH5 组合波形配置 参见 GC5C1 说明

28	GC5C3	RW	CH3/CH5 组合波形配置 参见 GC5C1 说明
27	GC5C2	RW	CH2/CH5 组合波形配置 参见 GC5C1 说明
26	GC5C1	RW	CH1/CH5 组合波形配置 0: OC1REFC 与 OC5REF 无关 1: OC1REFC 输出 OC1REF 和 OC5REF 的逻辑与
25-16	RFU	-	保留位, 请保持默认值
15-0	CCR5	RW	捕获/比较值 (Capture/Compare value) 如果通道 CC5 配置为输出: CCR5 为要装载到有效捕获/比较 5 寄存器的值 (预装载值)。 如果没有通过 TIMx_CCMR2 寄存器中的 OC5PE 位来使能预装载功能, 则该值立刻生效; 否则只在发生更新事件时生效 (拷贝到有效的捕获/比较寄存器 5)。 有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC5 输出上发出信号的值。 如果通道 CC5 配置为输入: CCR5 为上一个输入捕获 5 事件 (IC5) 发生时的计数器值。只能读取 TIMx_CCR5 寄存器, 无法对其进行编程。

### 13.5.24 ATIM 捕获/比较寄存器 6 (ATIM\_CCR6)

偏移地址: 0x4C 复位值: 0x0000 0000

31	GC6C6	RW	CH6/CH6 组合波形配置 参见 GC6C1 说明
30	GC6C5	RW	CH5/CH6 组合波形配置 参见 GC6C1 说明
29	GC6C4	RW	CH4/CH6 组合波形配置 参见 GC6C1 说明
28	GC6C3	RW	CH3/CH6 组合波形配置 参见 GC6C1 说明
27	GC6C2	RW	CH2/CH6 组合波形配置 参见 GC6C1 说明
26	GC6C1	RW	CH1/CH6 组合波形配置 0: OC1REFC 与 OC5REF 无关 1: OC1REFC 输出 OC1REF 和 OC6REF 的逻辑与
25-16	RFU	-	保留位, 请保持默认值
15-0	CCR6	RW	捕获 / 比较 6 值, 请参见 CCR1 说明

### 13.5.25 ATIM 刹车和死区寄存器 (ATIM\_BDTR)

偏移地址: 0x44 复位值: 0x0000 0000

注: 由于可以根据 LOCK 配置锁定位 BK2BID、BKBID、BK2DSRM、BKDSRM、BK2P、BK2E、BK2F[3:0]、BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0] 的写操作, 因此必须在第一次对 TIMx\_BDTR 寄存器执行写访问时对这些位进行配置。

位编号	位符号	权限	说明
31-30	RFU	-	保留位, 请保持默认值
29	BK2BID	RW	刹车 2 双向 (Break2 bidirectional) 请参见 BKBID 说明
28	BKBID	RW	刹车双向 0: 刹车输入 BRK 为输入模式 1: 刹车输入 BRK 为双向模式 在双向模式下 (BKBID 位置 1), 刹车输入配置为输入模式和开漏输出模式。任何激活的刹车事件都将使刹车输入上呈逻辑低电平, 以向外部器件指示发生了内部刹车事件。 注: 只要编程了 LOCK (TIMx_BDTR 寄存器中的 LOCK 位) 级别 1, 此位即无法修改。 注: 对该位执行任何写操作后, 都需要经过 1 个 APB 时钟周期的延迟才生效。
27	BK2DSRM	RW	刹车 2 解除 请参见 BKDSRM 说明
26	BKDSRM	RW	刹车解除 0: 启动刹车输入 BRK 1: 解除刹车输入 BRK 当刹车源激活后, 此位由硬件清零。 必须通过软件将 BKDSRM 位置 1 以释放双向输出控制 (开漏输出处于高阻态), 然后不断轮询该位, 直到其由硬件复位, 指示故障条件已消失。 注: 对该位执行任何写操作后, 都需要经过 1 个 APB 时钟周期的延迟才生效。
25	BK2P	RW	刹车 2 极性 (Break 2 polarity) 0: 刹车输入 BRK2 为低电平有效 1: 刹车输入 BRK2 为高电平有效 注: 只要编程了 LOCK (TIMx_BDTR 寄存器中的 LOCK 位) 级别 1, 此位即无法修改。 注: 对该位执行任何写操作后, 都需要经过 1 个 APB 时钟周期的延迟才生效。
24	BK2E	RW	刹车 2 使能 (Break 2 enable) 0: 禁止刹车输入 BRK2 1: 使能刹车输入 BRK2 注: BRK2 必须只在 OSSR = OSSI = 1 时使用。 注: 编程了 LOCK (TIMx_BDTR 寄存器中的 LOCK 位) 级别 1 后, 此位即无法修改。

			注： 对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。
23-20	BK2F	RW	<p>刹车 2 滤波器 (Break 2 filter)</p> <p>此位域可定义 BRK2 输入的采样频率和适用于 BRK2 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：</p> <p>0000: 无滤波器，BRK2 异步工作</p> <p>0001: fSAMPLING=fCK_INT, N=2</p> <p>0010: fSAMPLING=fCK_INT, N=4</p> <p>0011: fSAMPLING=fCK_INT, N=8</p> <p>0100: fSAMPLING=fDTS/2, N=6</p> <p>0101: fSAMPLING=fDTS/2, N=8</p> <p>0110: fSAMPLING=fDTS/4, N=6</p> <p>0111: fSAMPLING=fDTS/4, N=8</p> <p>1000: fSAMPLING=fDTS/8, N=6</p> <p>1001: fSAMPLING=fDTS/8, N=8</p> <p>1010: fSAMPLING=fDTS/16, N=5</p> <p>1011: fSAMPLING=fDTS/16, N=6</p> <p>1100: fSAMPLING=fDTS/16, N=8</p> <p>1101: fSAMPLING=fDTS/32, N=5</p> <p>1110: fSAMPLING=fDTS/32, N=6</p> <p>1111: fSAMPLING=fDTS/32, N=8</p> <p>注：编程了 LOCK(TIMx_BDTR 寄存器中的 LOCK 位)级别 1 后，此位即无法修改。</p>
19-16	BKF	RW	<p>刹车滤波器 (Break filter)</p> <p>此位域可定义 BRK 输入的采样频率和适用于 BRK 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：</p> <p>0000: 无滤波器，BRK 异步工作</p> <p>0001: fSAMPLING=fCK_INT, N=2</p> <p>0010: fSAMPLING=fCK_INT, N=4</p> <p>0011: fSAMPLING=fCK_INT, N=8</p> <p>0100: fSAMPLING=fDTS/2, N=6</p> <p>0101: fSAMPLING=fDTS/2, N=8</p> <p>0110: fSAMPLING=fDTS/4, N=6</p> <p>0111: fSAMPLING=fDTS/4, N=8</p> <p>1000: fSAMPLING=fDTS/8, N=6</p> <p>1001: fSAMPLING=fDTS/8, N=8</p> <p>1010: fSAMPLING=fDTS/16, N=5</p> <p>1011: fSAMPLING=fDTS/16, N=6</p> <p>1100: fSAMPLING=fDTS/16, N=8</p> <p>1101: fSAMPLING=fDTS/32, N=5</p> <p>1110: fSAMPLING=fDTS/32, N=6</p> <p>1111: fSAMPLING=fDTS/32, N=8</p>

			注：编程了 LOCK(TIMx_BDTR 寄存器中的 LOCK 位)级别 1 后，此位即无法修改。
15	MOE	RW	<p>主输出使能 (Main output enable)</p> <p>只要刹车输入 (BRK 或 BRK2) 为有效状态，此位便由硬件异步清零。此位由软件置 1，也可根据 AOE 位状态自动置 1。此位仅对配置为输出的通道有效。</p> <p>0：响应刹车事件 (2 个)。禁止 OC 和 OCN 输出 响应刹车事件或向 MOE 写入 0 时：OC 和 OCN 输出被禁止或被强制为空闲状态，具体取决于 OSSI 位。</p> <p>1：如果 OC 和 OCN 输出的相应使能位 (TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位) 均置 1，则使能 OC 和 OCN 输出。</p>
14	AOE	RW	<p>自动输出使能 (Automatic output enable)</p> <p>0：MOE 只能由软件置 1</p> <p>1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1 (如果刹车输入 BRK 和 BRK2 有效时，此位无效)</p> <p>注：只要编程了 LOCK (TIMx_BDTR 寄存器中的 LOCK 位) 级别 1，此位即无法修改。</p>
13	BKP	RW	<p>刹车极性 (Break polarity)</p> <p>0：刹车输入 BRK 为低电平有效</p> <p>1：刹车输入 BRK 为高电平有效</p> <p>注：只要编程了 LOCK (TIMx_BDTR 寄存器中的 LOCK 位) 级别 1，此位即无法修改。</p> <p>注：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。</p>
12	BKE	RW	<p>刹车使能</p> <p>该位可使能完整的刹车保护 (包括连接到 bk_acth 的所有源和相应的 BKIN 源，如图 146：刹车和刹车 2 电路概述所示)。</p> <p>0：禁止刹车功能</p> <p>1：使能刹车功能</p> <p>注：编程了 LOCK(TIMx_BDTR 寄存器中的 LOCK 位)级别 1 后，此位即无法修改。</p> <p>注：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。</p>
11	OSSR	RW	<p>运行模式下的关闭状态选择 (Off-state selection for Run mode)</p> <p>此位在 MOE=1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。</p> <p>有关详细信息，请参见 OC/OCN 使能说明 (第 20.4.11 节：ATIM 捕获/比较使能寄存器(ATIM_CCER))。</p> <p>0：处于无效状态时，禁止 OC/OCN 输出 (定时器释放输出控制，由强制高阻态的 GPIO 逻辑接管)。</p> <p>1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平 (输出仍由定时器控制)。</p> <p>注：编程了 LOCK(TIMx_BDTR 寄存器中的 LOCK 位)级别 2 后，此位即无法修改。</p>



10	OSSI	RW	<p>空闲模式下的关闭状态选择 (Off-state selection for Idle mode)</p> <p>当由于刹车事件或软件写操作而使 MOE=0 时，此位作用于配置为输出的通道。</p> <p>有关详细信息，请参见 OC/OCN 使能说明（第 20.4.11 节：ATIM 捕获/ 比较使能寄存器(ATIM_CCER)）。</p> <p>0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 GPIO 逻辑接管）。</p> <p>1：处于无效状态时，首先将 OC/OCN 输出强制为其无效电平，然后在死区后将其强制为无效电平。定时器始终控制输出。</p> <p>注：编程了 LOCK(TIMx_BDTR 寄存器中的 LOCK 位)级别 2 后，此位即无法修改。</p>
9-8	LOCK	RW	<p>锁定配置 (Lock configuration)</p> <p>这些位用于针对软件错误提供写保护。</p> <p>00：关闭锁定——不对任何位提供写保护。</p> <p>01：锁定级别 1，此时无法对 TIMx_BDTR 寄存器中的 DTG 位、TIMx_CR2 寄存器中的 OISx 和 OISxN 位以及 TIMx_BDTR 寄存器中的 BK2BID、BKBID、BK2DSRM、BKDSRM、BK2P、BK2E、BK2F[3:0]、BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0] 位执行写操作。</p> <p>10：锁定级别 2，此时无法对锁定级别 1 中适用的各位、CC 极性位（TIMx_CCER 寄存器中的 CCxP/CCxNP 位，只要通过 CCxS 位将相关通道配置为输出）以及 OSSR 和 OSSI 位执行写操作。</p> <p>11：锁定级别 3，此时无法对锁定级别 2 中适用的各位、CC 控制位（TIMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。</p> <p>注：复位后只能对 LOCK 位执行一次写操作。对 TIMx_BDTR 寄存器执行写操作后其中的内容将冻结，直到下一次复位。</p>
7-0	DTG	RW	<p>配置死区发生器 (Dead-time generator setup)</p> <p>此位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。</p> <p><math>DTG[7:5]=0xx \Rightarrow DT=DTG[7:0] \times tdtg</math>，其中 <math>tdtg=tDTS</math>。</p> <p><math>DTG[7:5]=10x \Rightarrow DT=(64+DTG[5:0]) \times tdtg</math>，其中 <math>Tdtg=2 \times tDTS</math>。</p> <p><math>DTG[7:5]=110 \Rightarrow DT=(32+DTG[4:0]) \times tdtg</math>，其中 <math>Tdtg=8 \times tDTS</math>。</p> <p><math>DTG[7:5]=111 \Rightarrow DT=(32+DTG[4:0]) \times tdtg</math>，其中 <math>Tdtg=16 \times tDTS</math>。</p> <p>示例：如果 <math>TDTs=125ns</math> (8MHz)，则可能的死区值为：</p> <p>0 到 15875 ns（步长为 125 ns）</p> <p>16 <math>\mu s</math> 到 31750 ns（步长为 250 ns）</p> <p>32 <math>\mu s</math> 到 63us（步长为 1 <math>\mu s</math>）</p> <p>64 <math>\mu s</math> 到 126 <math>\mu s</math>（步长为 2 <math>\mu s</math>）</p> <p>注：只要编程了 LOCK(TIMx_BDTR 寄存器中的 LOCK 位)级别 1、2 或 3，此位域即无法修改。</p>



注：由于可以根据 LOCK 配置锁定位BK2BID、BKBID、BK2DSRM、BKDSRM、BK2P、BK2E、BK2F[3:0]、BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR和 DTG[7:0]的写操作，因此必须在第一次对ATIM\_BDTR寄存器执行写访问时对这些位进行配置。

### 13.5.26 死区时间寄存器 2 (ATIM\_DTR2)

偏移地址：0x54 复位值：0x0000 0000

位编号	位符号	权限	说明
31-18	RFU	-	保留位，请保持默认值
17	DTPE	RW	死区时间值预加载使能 0：禁止死区时间值预加载 1：使能死区时间值预加载
16	DTAE	RW	死区时间不对称使能 0：上升沿和下降沿的死区时间是相同的，由 DTG[7:0] 定义。 1：上升沿的死区时间由 DTG[7:0] 定义，下降沿的死区时间由 DTGF[7:0] 定义。 注：只要编程了 LOCK（ATIM_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，此位域即无法修改
15-8	RFU	-	保留位，请保持默认值
7-0	DTGF	RW	下降沿的死区时间发生器 此位域定义插入到互补输出下降沿之间的死区持续时间。 $DTGF[7:5]=0xx \Rightarrow DTF=DTGF[7:0] \times tdtg$ ，其中 $tdtg = tDTS$ 。 $DTGF[7:5]=10x \Rightarrow DTF=(64+DTG[5:0]) \times tdtg$ ，其中 $tdtg = 2 \times tDTS$ 。 $DTGF[7:5]=110 \Rightarrow DTF=(32+DTG[4:0]) \times tdtg$ ，其中 $tdtg = 8 \times tDTS$ 。 $DTGF[7:5]=111 \Rightarrow DTF=(32+DTG[4:0]) \times tdtg$ ，其中 $tdtg = 6 \times tDTS$ 。 示例：如果 $tDTS = 125ns$ (8MHz)，则可能的死区值为： - 0 到 15875 ns（步长为 125ns） - 16 $\mu s$ 到 31750ns（步长为 250ns） - 32 $\mu s$ 到 63 $\mu s$ （步长为 1 $\mu s$ ） - 64 $\mu s$ 到 126 $\mu s$ （步长为 2 $\mu s$ ） 注：只要编程了 LOCK（ATIM_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，此位域即无法修改。

### 13.5.27 ATIM 编码控制寄存器(ATIM\_ECR)

偏移地址: 0x58 复位值: 0x0000 0000

31-8	RFU	-	保留位, 请保持默认值
7-6	IPOS	RW	<p>编码索引定位</p> <p>在正交编码器模式 (SMS[3: 0] = 0001、0010、0011、1110、1111) 中, 此位表示索引事件在哪个 AB 输入配置中重置计数器。</p> <p>00: 当 AB = 00 时, 索引会重置计数器</p> <p>01: 当 AB = 01 时, 索引会重置计数器</p> <p>10: 当 AB = 10 时, 索引会重置计数器</p> <p>11: 当 AB = 11 时, 索引会重置计数器</p> <p>在定向时钟编码器模式或时钟加方向编码器模式 (SMS[3: 0] = 1010、1011、1100、1101) 中, 这些位指示索引事件在哪个电平上重置计数器。在定向时钟编码器模式下, 这一点都适用于两个时钟输入。</p> <p>x0: 当时钟信号为低电平时, 索引将重置计数器</p> <p>x1: 当时钟信号为高电平时, 索引将重置计数器</p>
5	FIDX	RW	<p>该位表示是否只考虑第一个索引</p> <p>0: 索引始终处于活动状态</p> <p>1: 仅第一个索引重置计数器</p>
4-3	RFU	-	保留位, 请保持默认值
2-1	IDIR	RW	<p>索引方向</p> <p>此位域表示索引在哪种计数方向上重置计数器</p> <p>00: 在任何计数方向上重置计数器</p> <p>01: 仅在向上计数时重置计数器</p> <p>10: 仅在向下计数时重置计数器</p> <p>11: 保留</p>
0	IE	RW	<p>索引使能</p> <p>0: 禁止索引</p> <p>1: 使能索引</p>

### 13.5.28 输入选择寄存器 1 (ATIM\_TISEL1)

偏移地址: 0x5C 复位值: 0x0000 0000

位编号	位符号	权限	说明
31-28	RFU	-	保留位, 请保持默认值
27-24	TI3SEL	RW	TI4 通道输入捕获信号来源选择, 参见 TI1SEL 说明
23-20	RFU	-	保留位, 请保持默认值
19-16	TI3SEL	RW	TI3 通道输入捕获信号来源选择, 参见 TI1SEL 说明
15-12	RFU	-	保留位, 请保持默认值

11-8	TI2SEL	RW	TI2 通道输入捕获信号来源选择, 参见 TI1SEL 说明
7-4	RFU	-	保留位, 请保持默认值
3-0	TI1SEL	RW	TI1 通道输入捕获信号来源选择 0000: ATIM_CH1 0001: VC1_OUT 0010: VC2_OUT 0011: UART1_RXD 0100: UART2_RXD 0101: MCO_OUT 0110: HSE_FAULT 0111: LSE_FAULT 1000: RTC_OUT 1001: LSI_OUT 1010: BTIM1_Trgo 1011: BTIM2_Trgo 1100: BTIM3_Trgo 1101: GTIM_Trgo

### 13.5.29 输入选择寄存器 1 (ATIM\_TISEL2)

偏移地址: 0x6C 复位值: 0x0000 0000

位编号	位符号	权限	说明
31-12	RFU	-	保留位, 请保持默认值
11-8	TI6SEL	RW	TI6 通道输入捕获信号来源选择, 参见 TI1SEL 说明。
7-4	RFU	-	保留位, 请保持默认值
3-0	TI5SEL	RW	TI5 通道输入捕获信号来源选择, 参见 TI1SEL 说明。

### 13.5.30 复用功能选择寄存器 1 (ATIM\_AF1)

偏移地址: 0x60 复位值: 0x0000 0001

位编号	位符号	权限	说明
31-18	RFU	-	保留位, 请保持默认值
17-14	ETRSRL	RW	ETR 信号来源选择 0000: ATIM_ETR 引脚 0001: VC0_OUT 0010: VC1_OUT 0101: ADC_AWD 1001: GTIM_ETR 引脚 1101: UART0_TXD 1110: UART1_TXD 注: 只要编程了 LOCK (ATIM_BDTR 寄存器中的 LOCK 位) 级别 1, 这些位即无法修改
13-12	RFU	-	保留位, 请保持默认值
11	BKVC2P	RW	BRK VC2 输入极性

			<p>此位选择 VC2 输入电平灵敏度，必须与 BKP 极性位一起编程。</p> <p>0: VC2 输入为高电平有效</p> <p>1: VC2 输入为低电平有效</p> <p>注：只要编程了 LOCK（ATIM_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。</p>
10	BKVC1P	RW	<p>BRK VC1 输入极性</p> <p>此位选择 VC1 输入电平灵敏度，必须与 BKP 极性位一起编程。</p> <p>0: VC1 输入为高电平有效</p> <p>1: VC1 输入为低电平有效</p> <p>注：只要编程了 LOCK（ATIM_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。</p>
9	BKINP	RW	<p>BRK BKIN 输入极性</p> <p>此位选择 BKIN 复用功能输入电平灵敏度，必须与 BKP 极性位一起编程。</p> <p>0: BKIN 输入为高电平有效</p> <p>1: BKIN 输入为低电平有效</p> <p>注：只要编程了 LOCK（ATIM_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。</p>
8-3	RFU	-	保留位，请保持默认值
2	BKVC2E	RW	<p>BRK VC2 使能</p> <p>此位使能定时器 BRK 输入的 VC2。VC2 输出与其他 BRK 源进行“或”运算。</p> <p>0: 禁止 VC2 输入</p> <p>1: 使能 VC2 输入</p> <p>注：只要编程了 LOCK（ATIM_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。</p>
1	BKVC1E	RW	<p>BRK VC1 使能</p> <p>此位使能定时器 BRK 输入的 VC1。VC1 输出与其他 BRK 源进行“或”运算。</p> <p>0: 禁止 VC1 输入</p> <p>1: 使能 VC1 输入</p> <p>注：只要编程了 LOCK（ATIM_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。</p>
0	BKIN	RW	<p>BRK BKIN 输入使能</p> <p>此位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。</p> <p>0: 禁止 BKIN 输入</p> <p>1: 使能 BKIN 输入</p> <p>注：只要编程了 LOCK（ATIM_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。</p>

### 13.5.31 复用功能选择寄存器 2 (ATIM\_AF2)

偏移地址: 0x64 复位值: 0x0000 0001

位编号	位符号	权限	说明
31-19	RFU	-	保留位, 请保持默认值
18-16	OCRSEL	RW	OCREF_CLR 源选择 000: VC1_OUT 001: VC2_OUT 100: ADC_AWD 111: ETRF
15-12	RFU	-	保留位, 请保持默认值
11	BK2VC2P	RW	BRK2 VC2 输入极性 此位选择 VC2 输入电平灵敏度, 必须与 BK2P 极性位一起编程。 0: VC2 输入为低电平有效 1: VC2 输入为高电平有效 注: 只要编程了 LOCK (ATIM_BDTR 寄存器中的 LOCK 位) 级别 1, 此位即无法修改。
10	BK2VC1P	RW	BRK2 VC1 输入极性 此位选择 VC1 输入电平灵敏度, 必须与 BK2P 极性位一起编程。 0: VC1 输入为低电平有效 1: VC1 输入为高电平有效 注: 只要编程了 LOCK (ATIM_BDTR 寄存器中的 LOCK 位) 级别 1, 此位即无法修改。
9	BK2INP	RW	BRK2 BKIN2 输入极性 此位选择 BKIN2 复用功能输入电平灵敏度, 必须与 BK2P 极性位一起编程。 0: BKIN2 输入为低电平有效 1: BKIN2 输入为高电平有效 注: 只要编程了 LOCK (ATIM_BDTR 寄存器中的 LOCK 位) 级别 1, 此位即无法修改。
8-3	RFU	-	保留位, 请保持默认值
2	BK2VC2E	RW	BRK2 VC2 使能 此位使能定时器 BRK2 输入的 VC2。VC2 输出与其他 BRK2 源进行“或”运算。 0: 禁止 VC2 输入 1: 使能 VC2 输入 注: 只要编程了 LOCK (ATIM_BDTR 寄存器中的 LOCK 位) 级别 1, 此位即无法修改。
1	BK2VC1E	RW	BRK2 VC1 使能 此位使能定时器 BRK2 输入的 VC1。VC1 输出与其他 BRK2 源进行“或”运算。 0: 禁止 VC1 输入

			1: 使能 VC1 输入 注: 只要编程了 LOCK (ATIM_BDTR 寄存器中的 LOCK 位) 级别 1, 此位即无法修改。
0	BK2INE	RW	BRK2 BKIN 输入使能 此位使能定时器 BRK2 输入的 BKIN2 复用功能。BKIN2 输入与其他 BRK2 源进行“或”运算。 0: 禁止 BKIN2 输入 1: 使能 BKIN2 输入 注: 只要编程了 LOCK (ATIM_BDTR 寄存器中的 LOCK 位) 级别 1, 此位即无法修改。

## 14 通用定时器 (GTIM)

### 14.1 概述

通用定时器 (GTIM) 包含一个 16bit 自动重载计数器, 并由一个可编程预分频器驱动。GTIM 支持定时、计数、复位、门控、触发和编码器等多种工作模式, 带 4 路独立的捕获 / 比较通道, 可以测量输入信号的脉冲宽度 (输入捕获) 或者产生输出波形 (输出比较和 PWM)。。

### 14.2 主要特性

通用 GTIM 定时器具有以下特性:

- 16bit 递增、递减和递增 / 递减自动重载计数器
- 可编程预分频器支持 1、2、3、4、...、65536 分频
- 支持单次计数模式和连续计数模式
- 4 路独立输入捕获和输出比较通道
- 触发输入信号 (TRGI) 控制定时器实现多种从模式
- 定时器级联 ITR 和片内外设互联 ETR
- 支持针对定位的增量 ( 正交 ) 编码器和霍尔传感器电路
- 多种事件发生时产生中断请求
  - 更新事件
  - 触发事件
  - 输入捕获
  - 输出比较

## 14.3 功能描述

### 14.3.1 功能框图

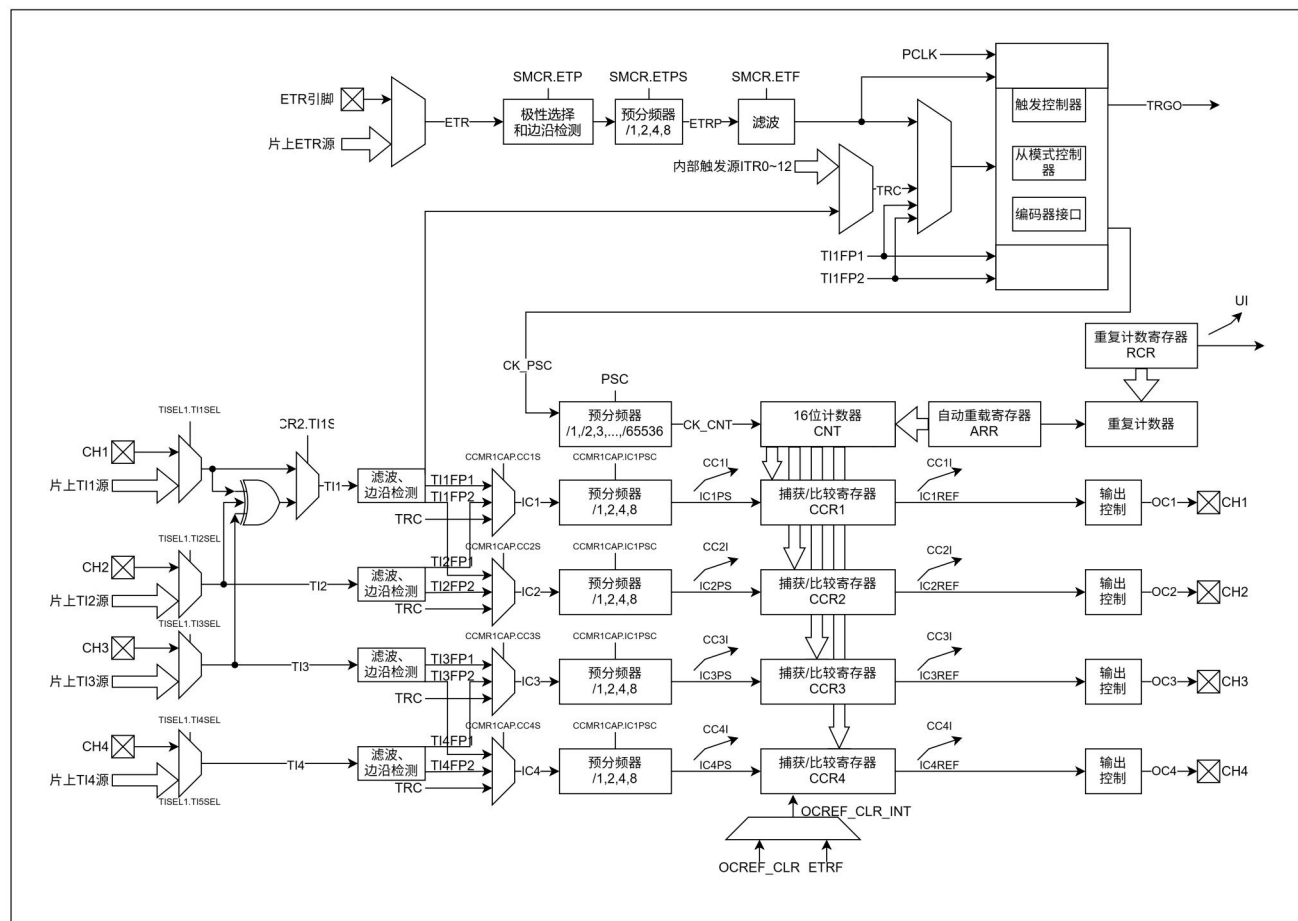


图 14-1 系统框图



### 14.3.1.1 时钟源

计数器的计数时钟源可由内部时钟 PCLK、触发信号 TRGI 或外部触发输入信号 ETRF 提供，经预分频器 GTIM\_PSC 分频后驱动计数器进行计数。不同工作模式下具有不同时钟源。

### 14.3.1.2 预分频器

预分频器对 CK\_PSC 时钟进行分频，得到计数时钟 CK\_CNT，以驱动计数器计数。分频系数通过 GTIM\_PSC 寄存器进行设置，支持 1、2、3、4、...、65536 分频。

GTIM\_PSC 寄存器具有缓冲功能，可在运行中修改，新的预分频值将在下一个更新事件发生时生效。

下图给出了运行过程中预分频器的分频由 1 变为 2 时的时序图：

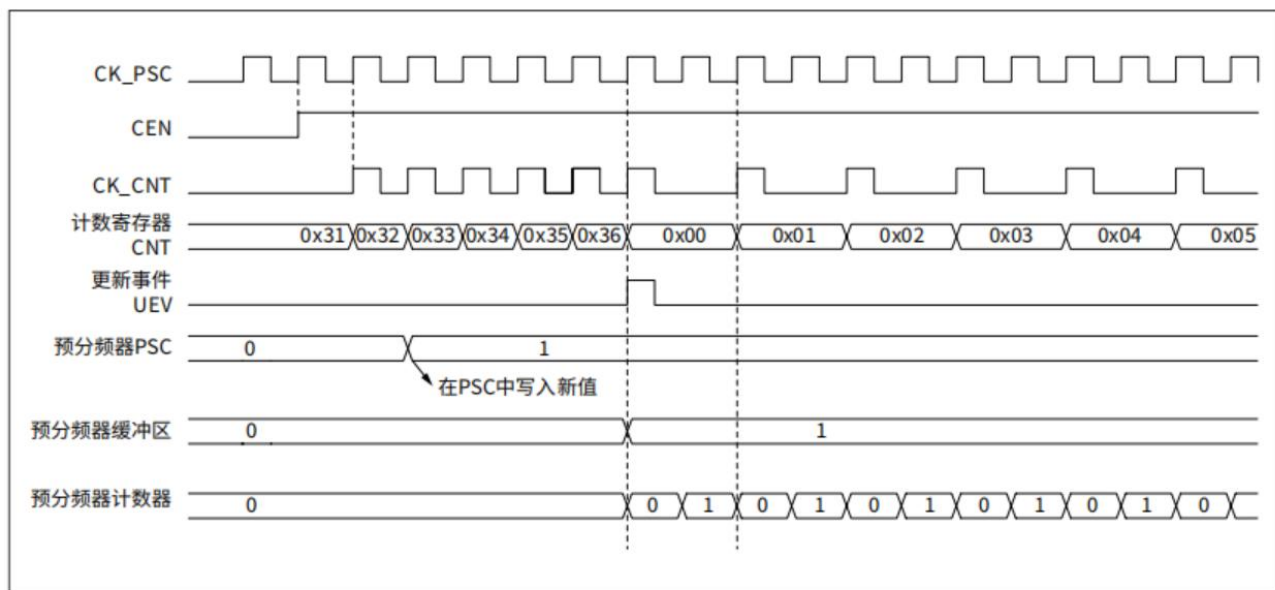


图 14-2 预分频器分频由 1 变为 2 时的时序图

### 14.3.1.3 计数器与计数模式

计数器可设置为递增计数（边沿对齐模式）、递减计数（边沿对齐模式）或递增 / 递减双向计数（中心对齐模式）。

具体通过控制寄存器 GTIM\_CR1 的 CMS 和 DIR 位域进行配置，如下表所示：

表 14-1 计数模式

GTIM_CR1.CMS	GTIM_CR1.DIR	计数模式
0	0	递增计数（边沿对齐模式）
	1	递减计数（边沿对齐模式）
1	-	中心对齐模式 1
10	-	中心对齐模式 2
11	-	中心对齐模式 3

当设置 GTIM\_CR1 寄存器的 CEN 位域为 1 时，计数器开始按设定模式计数，注意实际的计数器使能信号 CNT\_EN 在 CEN 置 1 的一个时钟周期后被置 1。

递增计数模式

在递增计数模式下，计数器从 0 开始递增计数到重载值 ARR，然后重新从 0 开始递增计数，同时生成计数器上溢出事件。

当计数器上溢出时,产生上溢出信号 OV,同时产生更新事件 UEV(OV 信号和 UEV 信号会自动清除),计数器更新中断标志位 GTIM\_ISR.UIF 被硬件置位,如果允许中断(设置 GTIM\_IER.UIE 为 1),将产生中断请求,设置 GTIM\_ICR.UIF 为 0 清除该标志位。

以下是计数器在不同时钟频率下的操作示例,其中 GTIM\_ARR = 0x36:

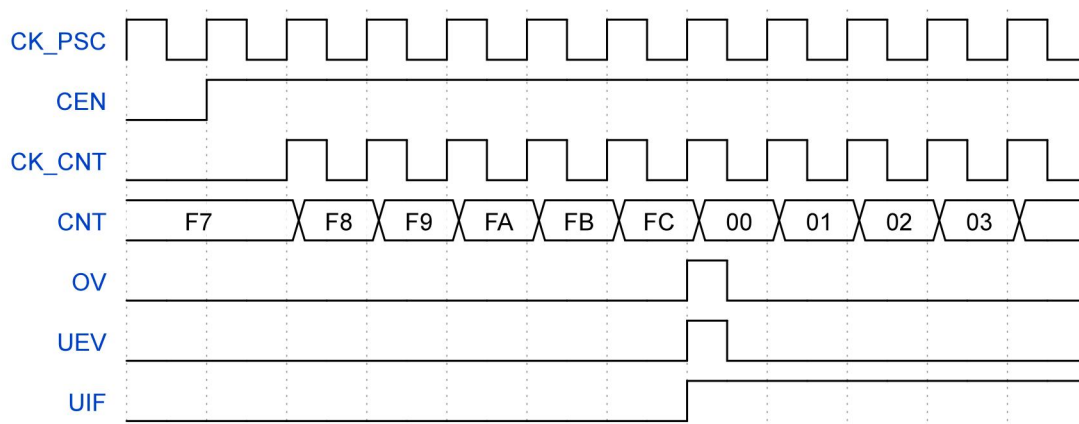


图 14-3 递增计数, 内部时钟分频因子为 1

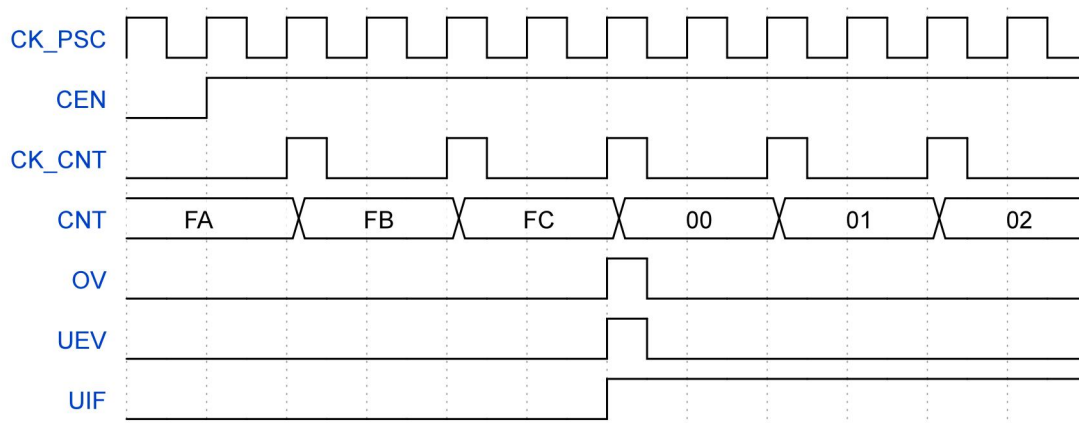


图 14-4 递增计数, 内部时钟分频因子为 2

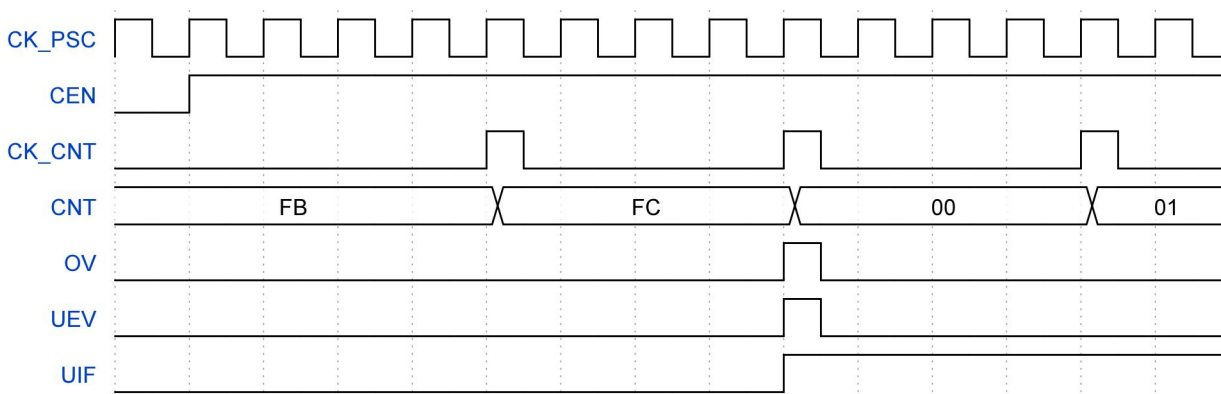


图 14-5 递增计数, 内部时钟分频因子为 4

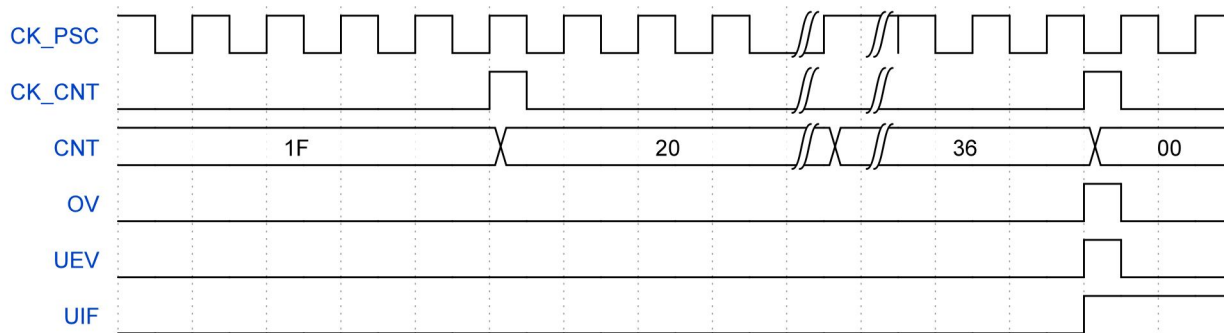


图 14-6 递增计数，内部时钟分频因子为 N

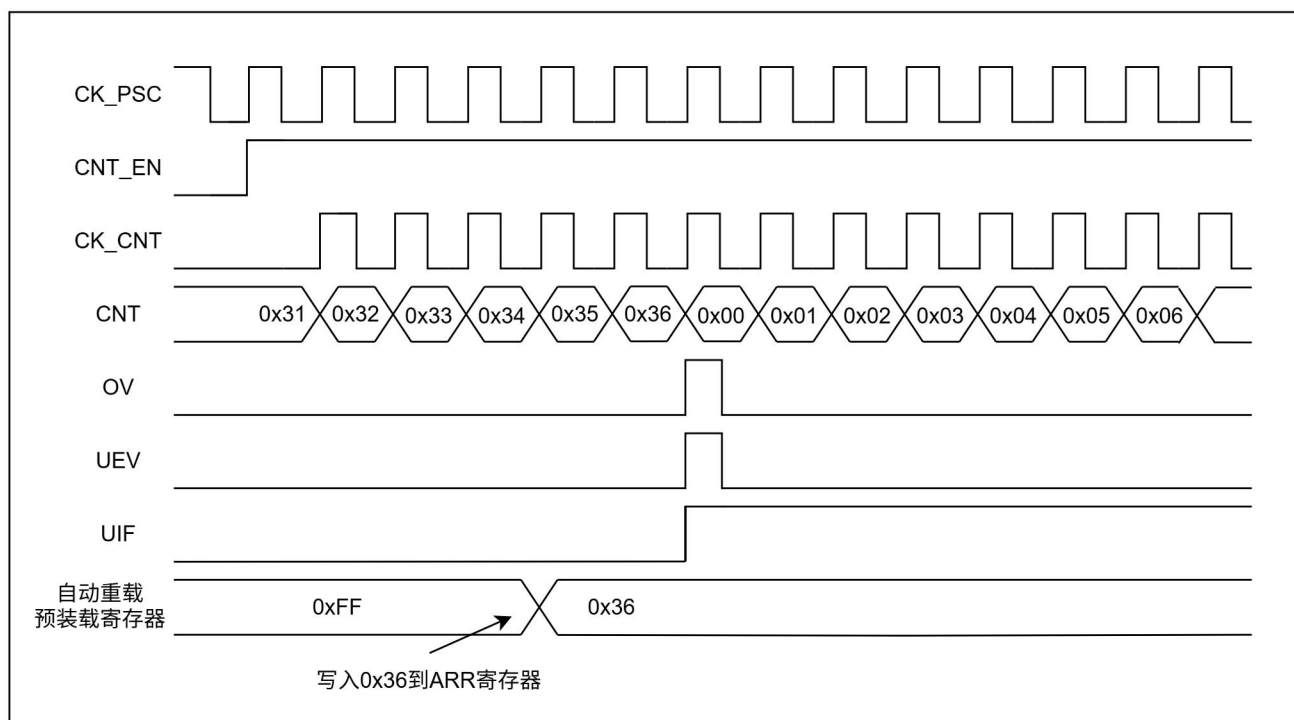


图 14-7 递增计数，当重载缓存禁止时的更新事件 (GTIM\_CR1.ARPE=0)

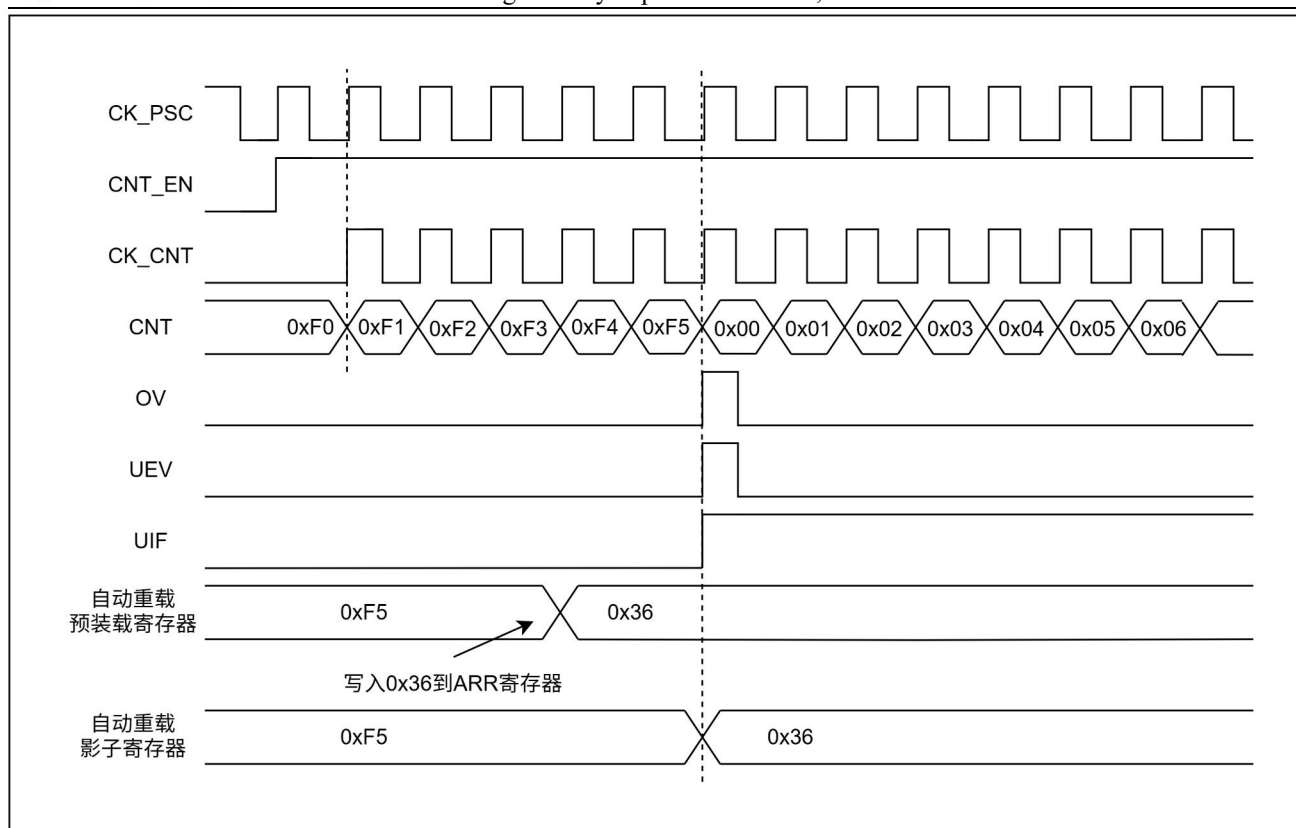


图 14-8 递增计数，当重载缓存使能时的更新事件 (GTIM\_CR1.ARPE=1)

#### 递减计数模式

在边沿对齐模式下设置控制寄存器 GTIM\_CR1 的 DIR 位为 1 时，计数器工作在递减计数模式。

在递减计数模式下，硬件自动加载 ARR 值到计数器 CNT，计数器开始递减计数到 0，然后重新装载 ARR 值递减计数，同时生成计数器下溢出事件。

当计数器下溢出时，产生下溢出信号 UND，同时产生更新事件 UEV (UND 信号和 UEV 信号会自动清除)，计数器更新中断标志位 GTIM\_ISR.UIF 被硬件置位，如果允许中断 (设置 GTIM\_IER.UIE 为 1)，将产生中断请求，设置 GTIM\_ICR.UIF 为 0 清除该标志位。

以下是计数器在不同时钟频率下的操作示例，其中 GTIM\_ARR = 0x36:

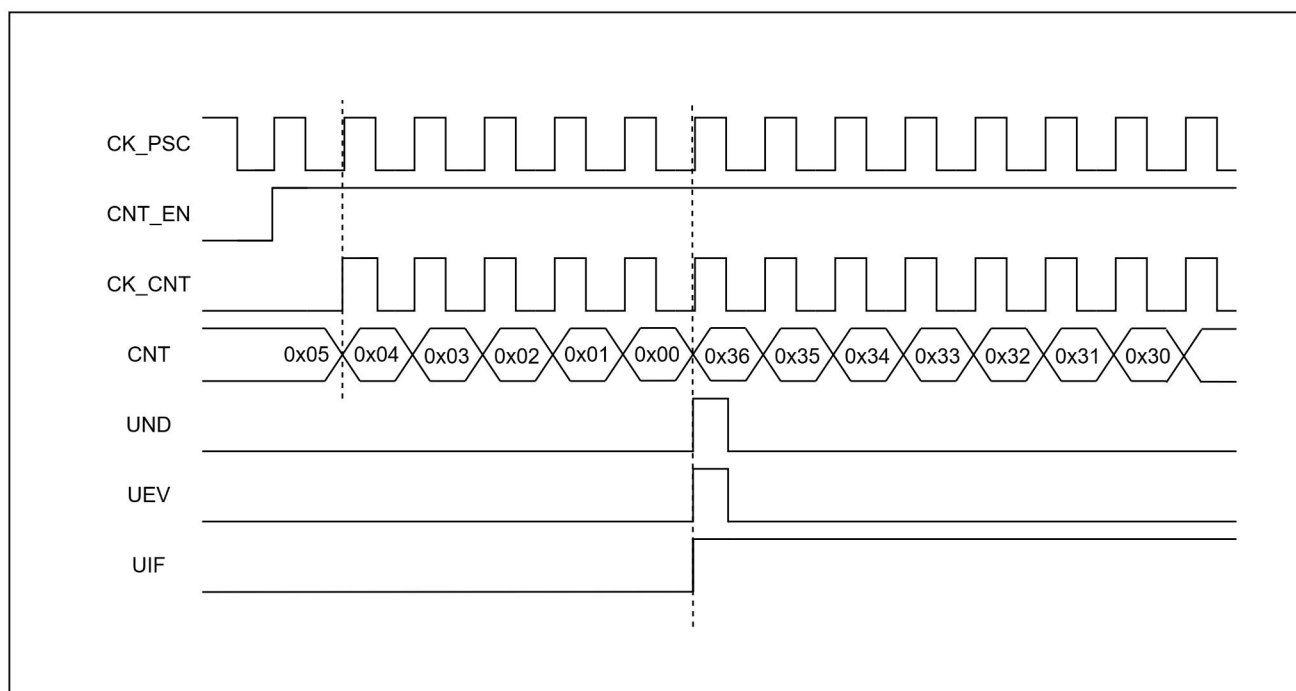


图 14-9 递减计数，内部时钟分频因子为 1

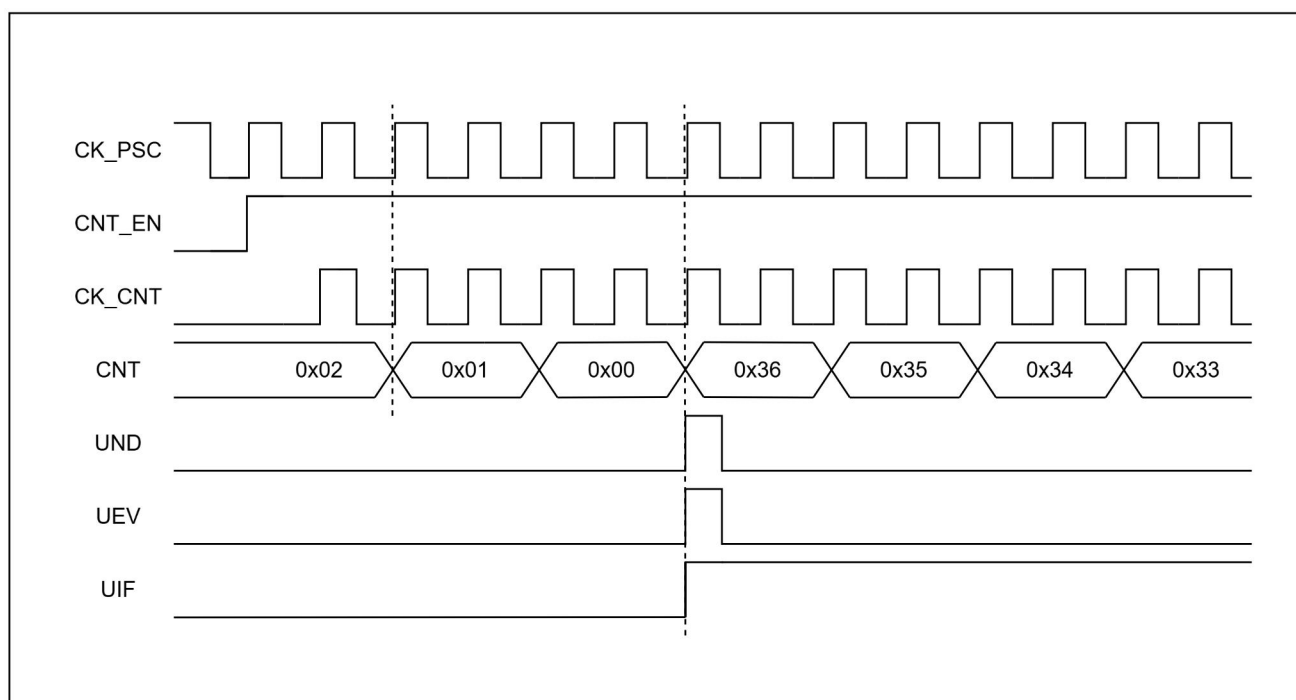


图 14-10 递减计数，内部时钟分频因子为 2

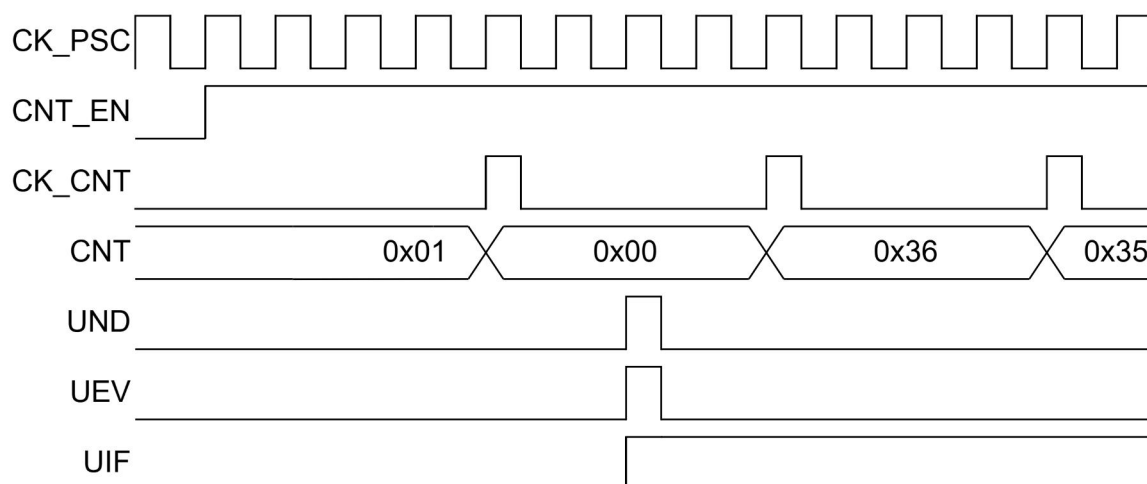


图 14-11 递减计数，内部时钟分频因子为 4

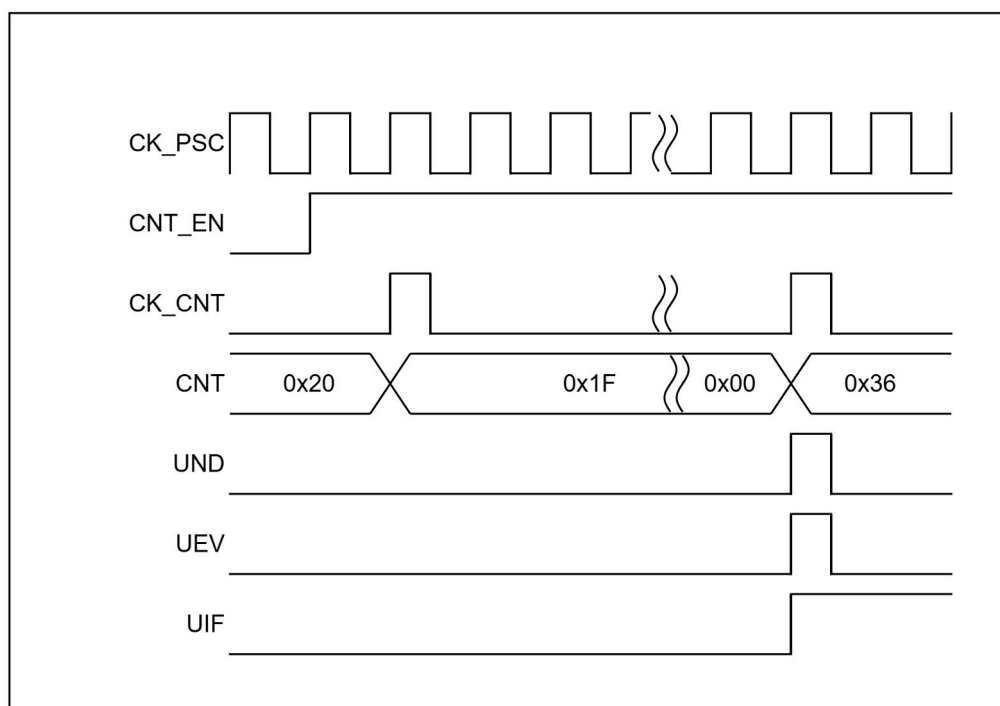


图 14-12 递减计数，内部时钟分频因子为 N

### 中心对齐模式(递增/递减计数)

在中心对齐模式下，计数器从 0 开始递增计数到 ARR-1，产生一个计数器上溢出事件，然后从 ARR 值开始递减计数到 1 并产生一个计数器下溢出事件，之后重新从 0 开始递增计数。

中心对齐模式包括中心对齐模式 1、中心对齐模式 2 和中心对齐模式 3，三种模式在计数方式上完全相同，只是输出比较中断标志的置位时机不同。

在中心对齐模式下，控制寄存器 GTIM CR1 的 DIR 位不能由软件写入，但可以读出，DIR 由硬件更新并指示当前的计数方向。启动中心对齐模式时，计数器将根据当前的 DIR 位域值进行递增或递减计

数。不能同时通过软件修改 DIR 和 CMS 位域。

不建议在运行中心对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤其是：

- 如果在计数器 CNT 中写入大于自动重载值 ARR 的值( $CNT > ARR$ )，则不会更新方向。即，如果计数器之前是递增计数，则继续递增计数。
- 如果向计数器 CNT 写入 0 或 ARR 值，计数方向会更新，但不生成更新事件 UEV。

使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新事件 UEV(设置 GTIM\_EGR.UG 为 1)，并且不要在计数器运行过程中对其执行写操作。

以下是计数器在不同时钟频率下的操作示例：

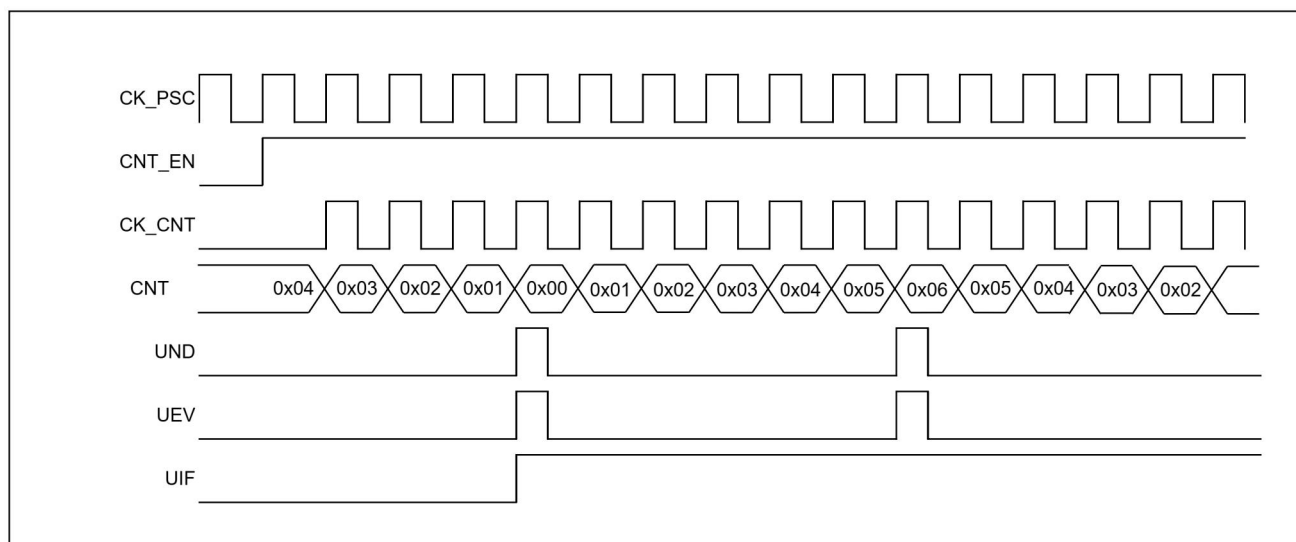


图 14-13 中心对齐模式，内部时钟分频因子为 1，ARR=0x06

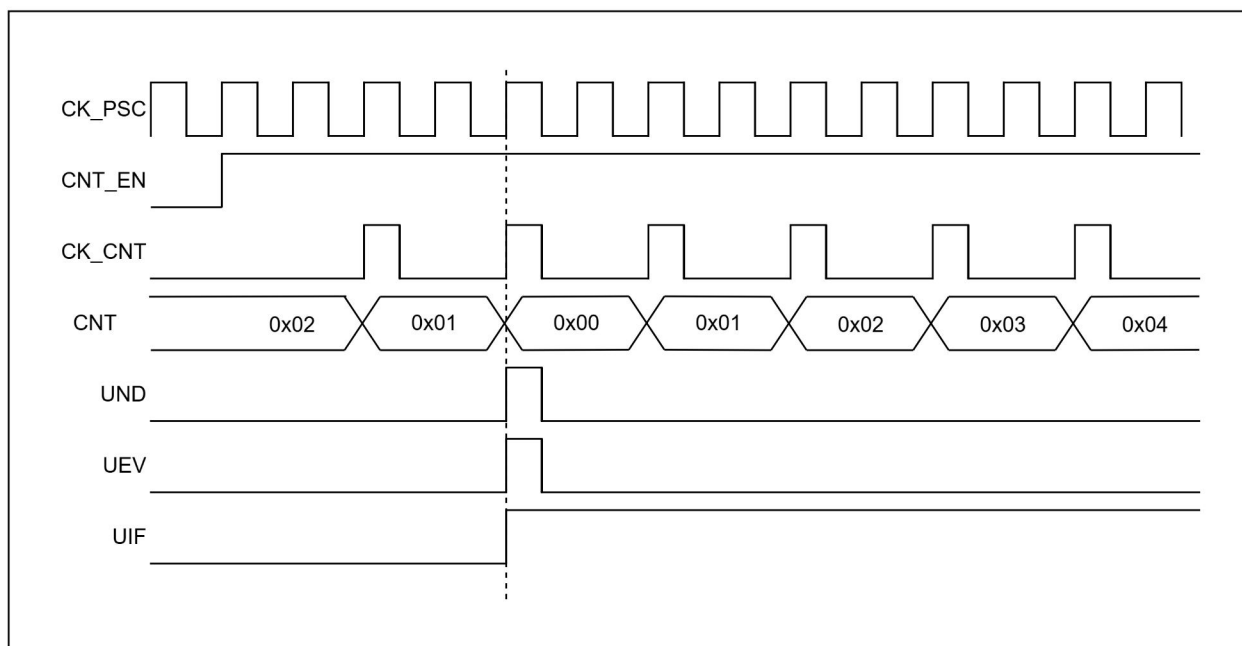


图 14-14 中心对齐模式，内部时钟分频因子为 2



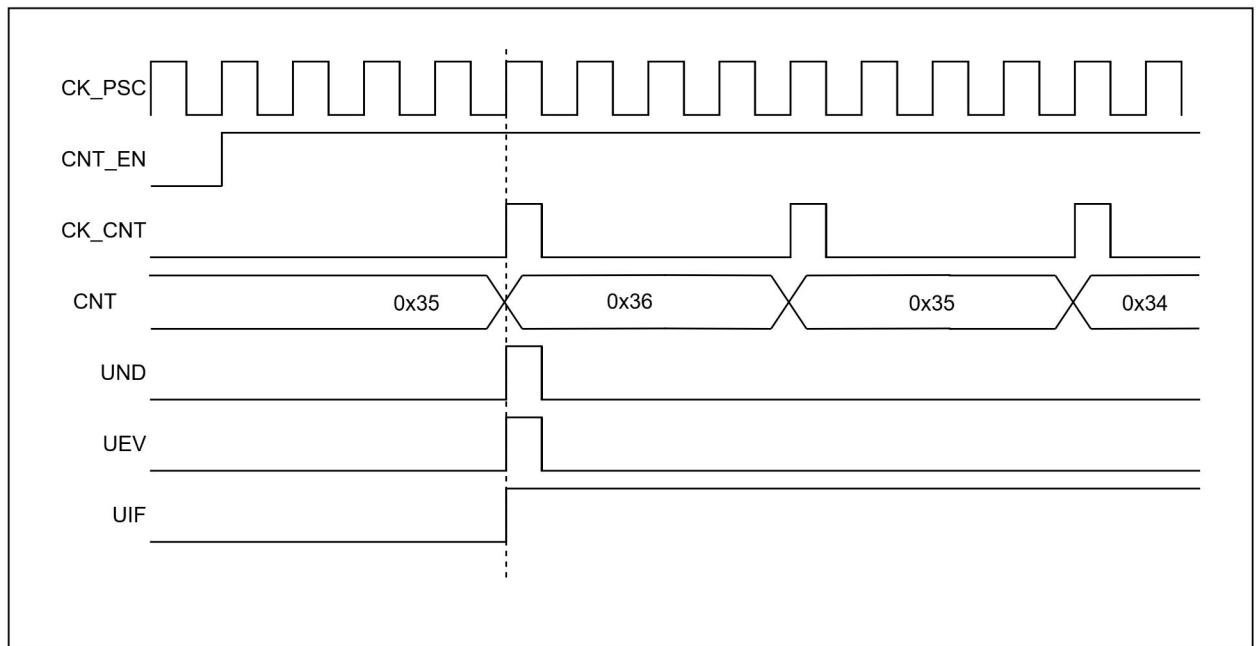


图 14-15 中心对齐模式，内部时钟分频因子为 4，ARR=0x36

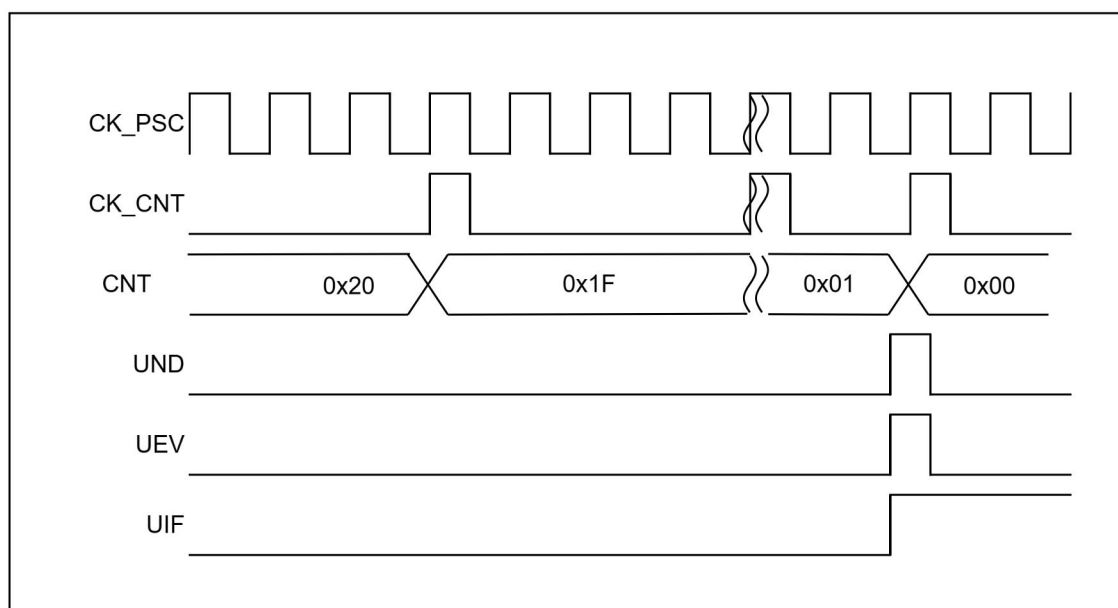


图 14-16 中心对齐模式，内部时钟分频因子为 N

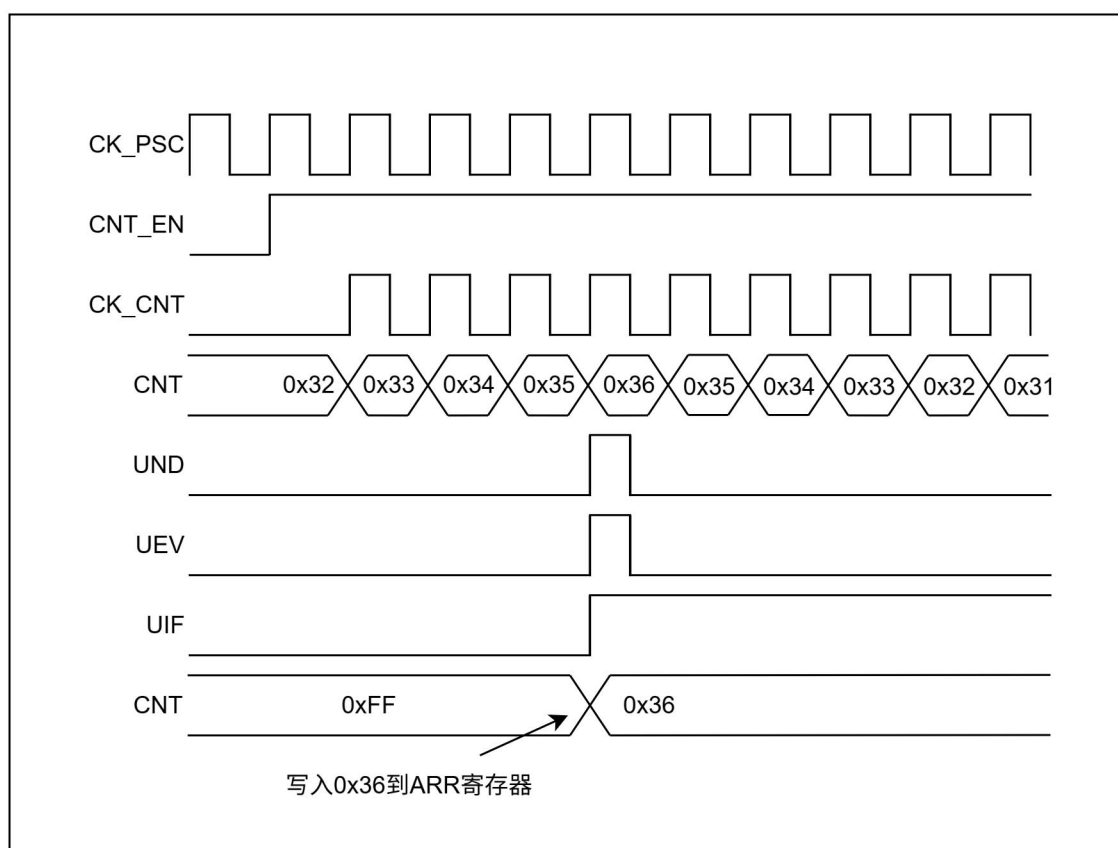


图 14-17 中心对齐模式，当重载缓存禁止时的更新事件 (GTIM\_CR1.ARPE=0)

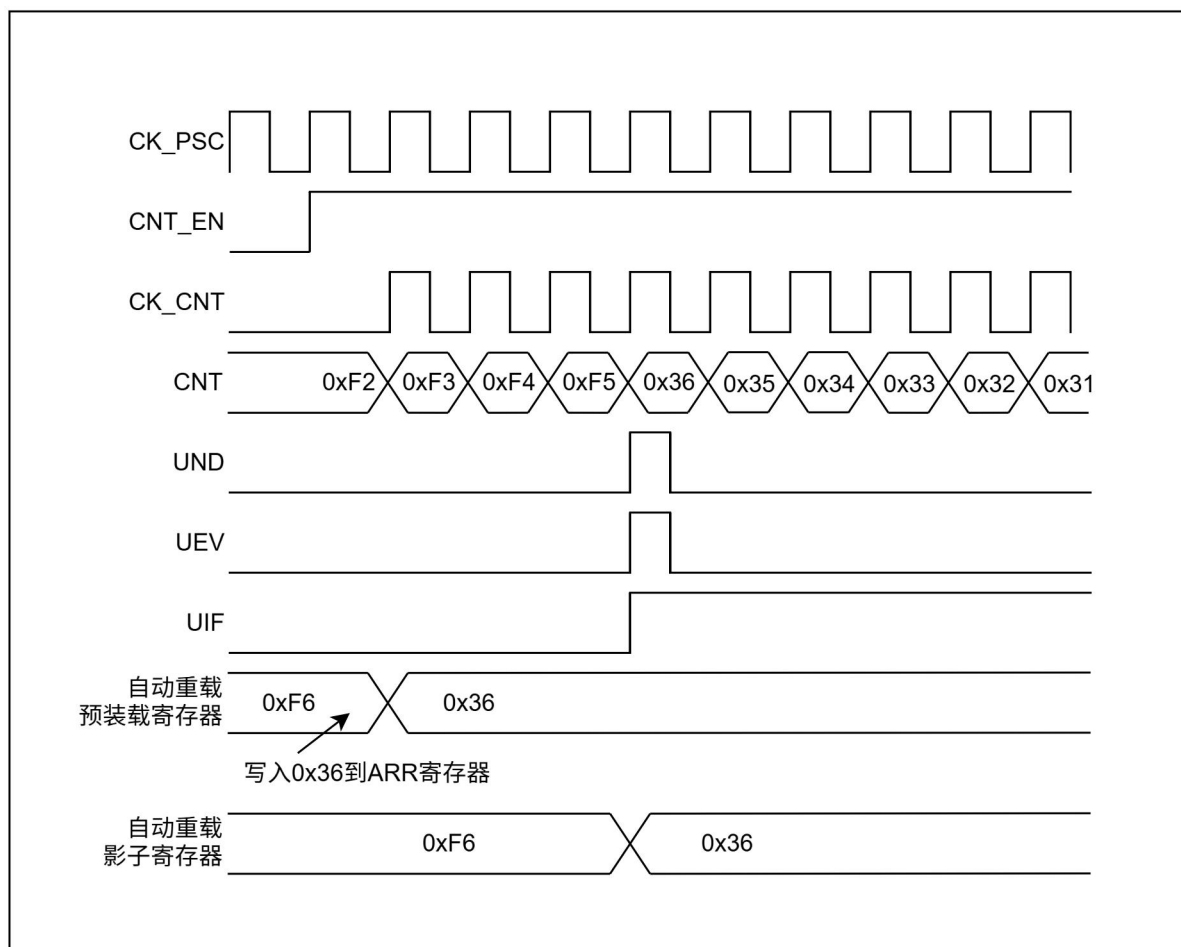


图 14-18 中心对齐模式，当重载缓存使能时的更新事件 (GTIM\_CR1.ARPE=1)

#### 14.3.1.4 重载寄存器

自动重载寄存器 GTIM\_ARR 具有缓存功能,通过控制寄存器 GTIM\_CR1 的 ARPE 位域开启或关闭。

当计数器处于停止状态或是缓存功能关闭时，更新重载寄存器 ARR 将会立即更新其影子寄存器。当定时器处于运行状态且缓存功能有效时，修改重载寄存器 ARR 将不会立即更新影子寄存器，仅当生成更新事件 UEV 时才会将重载寄存器 ARR 的值更新到影子寄存器中。

#### 14.3.1.5 更新事件 UEV

允许通过控制寄存器 GTIM\_CR1 的 UDIS 位域来禁止或使能更新。

##### 使能 UEV

设置 UDIS 位域为 0 使能 UEV，此时根据 URS 位域可选择更新请求源，如下表所示：

表 14-2 更新源设置

GTIM_CR1.URS	更新源
0	计数器上溢出、下溢出； UG 置位； 通过从模式控制器生成的更新事件
1	计数器上溢出、下溢出

当发生更新事件时，将进行以下动作：

1. 重新初始化计数器：  
递减计数(边沿对齐模式):重新加载自动重载值 GTIM\_ARR;  
中心对齐模式或递增计数(边沿对齐模式):计数器清零。
2. 如果使用了缓存寄存器功能，将更新对应的预装载寄存器到其影子寄存器。
3. 预分频器的计数器被清零，GTIM\_PSC 中新的预分频值生效。
4. 事件更新中断标志位 GTIM\_ISR.UIF 被硬件置位。

当发生一个更新事件 UEV 时，事件更新中断标志位 GTIM\_ISR.UIF 会被硬件置位，如果允许中断(设置 GTIMIER.UIE 为 1)，将产生一个更新中断请求，设置 GTIM\_ICR.UIF 为 0 可清除该标志位。

#### 禁止 UEV

设置 UDIS 位域为 1 禁止 UEV，不再生成任何更新事件。

如果使用了缓存寄存器功能，对应的各影子寄存器的值保持不变。但如果 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器的计数器。

### 14.3.1.6 单脉冲模式

计数器可工作在单次计数或连续计数模式下，通过控制寄存器 GTIM\_CR1 的 OPM 位域来选择。

#### 单次计数模式

设置 GTIM\_CR1.OPM 为 1，使定时器工作在单次计数模式下。

启动计数器后，计数器 CNT 在 CK\_CNT 时钟的驱动下计数，在发生下一更新事件时，计数器停止计数，同时 GTIM\_CR1.CEN 被硬件自动清零。

可以通过从模式控制器启动计数器，配合定时器的输出比较模式，使得计数器在一个触发信号的触发下启动，并在一段可编程的延时后产生一个脉宽可编程的单脉冲。如下图所示，定时器在检测到触发信号的有效边沿时开始计数，延迟 t<sub>DELAY</sub> 之后，在比较输出端口上产生一个宽度为 t<sub>PULSE</sub> 的正脉冲。其中 t<sub>DELAY</sub> 由计数寄存器 GTIM\_CNT 的初值和捕获 / 比较寄存器 GTIM\_CCRy 的差值确定，t<sub>PULSE</sub> 由自动重载寄存器 GTIM\_ARR 和捕获 / 比较寄存器 GTIM\_CCRy 的差值来确定。

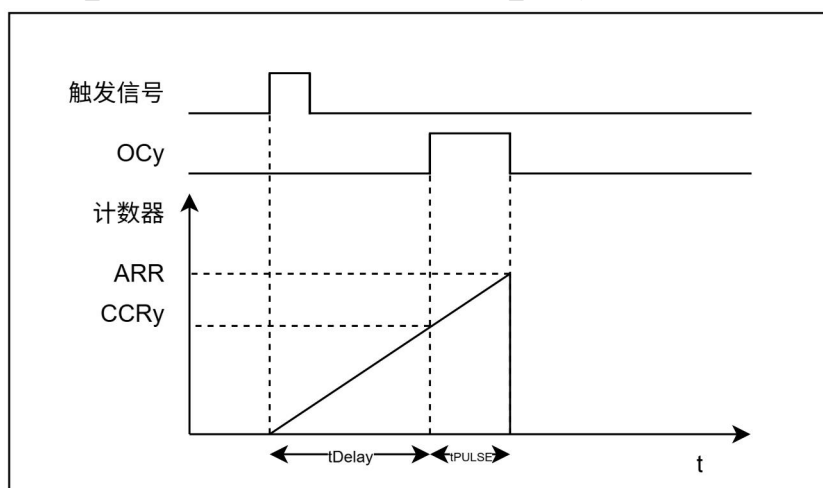


图 14-19 单脉冲输出示例（递增计数模式）

采样触发输入到激活 OC 输出需要多个时钟周期，可设置比较模式寄存器（GTIM\_CCMR1CMP 和 GTIM\_CCMR2CMP）的 OCyFE 位域为 1，以缩短 OC 输出延迟时间。

#### 连续计数模式

设置 GTIM\_CR1.OPM 为 0，使定时器工作在连续计数模式下，计数器在发生更新事件时不会停止计数。

#### 14.3.1.7 外部触发输入通道

外部触发输入信号 ETR 可用作从模式控制器的触发输入 (TRGI)，也可用作计数器的计数时钟。可对 ETR 信号进行输入控制，包括极性选择和边沿检测、预分频和滤波。

##### 极性选择和边沿检测

从模式控制寄存器 GTIM\_SMCR 的 ETP 位域用于选择 ETR 输入信号的触发极性。当设置 ETP 为 0 时，ETR 不反相，高电平或上升沿有效；当 ETP 设置为 1 时，ETR 反相，低电平或下降沿有效。

##### 预分频器

外部触发信号 ETRP 频率不得超过 PCLK 频率的 1/4。当 ETRP 的频率较高时，用户应通过适当的 ETSP 预分频器设置对外部信号进行分频，以降低 ETRP 频率，可设置分频系数为 1、2、4、8。

##### 滤波器

滤波器采用数字滤波方式，以一定频率对输入信号进行采样，当连续采样到 N 个相同电平时信号有效，否则信号无效，以此滤除高频杂波信号。

ETR 输入信号的滤波器由从模式控制寄存器 GTIM\_SMCR 的 ETF 位域进行配置，可设置采样频率和采样点的个数。采样频率  $f_{\text{SAMPLING}}$  由  $f_{\text{DTS}}$  分频后的时钟提供， $f_{\text{DTS}}$  是  $f_{\text{PCLK}}$  分频后得到的频率，分频因子由 GTIM\_CR1 寄存器的 CKD 位域配置，可设置 1、2、4 分频。

#### 14.3.1.8 输入捕获通道

GTIM 支持 4 个输入捕获通道 TI1/2/3/4，可用作捕获命令。其中 TI1 和 TI2 还可用作从模式控制器的触发输入 (TRGI) 和编码器接口输入。

支持对 TIy 信号进行输入控制，包括滤波、边沿检测和预分频。

##### 滤波器

滤波阶段以一定频率对相应的 TIy 输入信号进行采样，生成滤波后信号 TIyF。

TIy 输入信号的滤波器由捕获模式寄存器 GTIM\_CCMR1CAP 和 GTIM\_CCMR2CAP 的 ICyF 位域进行配置，可设置采样频率和采样点的个数。采样频率  $f_{\text{SAMPLING}}$  由  $f_{\text{PCLK}}$  或  $f_{\text{DTS}}$  分频后的时钟提供， $f_{\text{DTS}}$  是  $f_{\text{PCLK}}$  分频后得到的频率，分频因子由 GTIM\_CR1 寄存器的 CKD 位域配置，可设置 1、2、4 分频。

##### 边沿检测

带有极性选择功能的边沿检测器可生成一个 TIxFPx 信号。

##### 预分频器

TIy 信号用于输入捕获时，可对捕获通道信号 ICy 进行分频，通过捕获模式寄存器 GTIM\_CCMR1CAP 和 GTIM\_CCMR2CAP 的 ICyPSC 位域进行控制，支持 1、2、4、8 分频。

#### 14.3.1.9 输出比较通道

GTIM 支持 4 个独立输出比较通道 CH1/2/3/4，每个通道均由一个捕获 / 比较寄存器（包括一个影子寄存器）和一个输出阶段（比较器和输出控制）组成。

输出控制单元用于比较匹配时，控制输出端口的波形，可配置输出比较模式、输出极性选择和输出使能等。以比较通道 1 为例，其框图如下图所示：

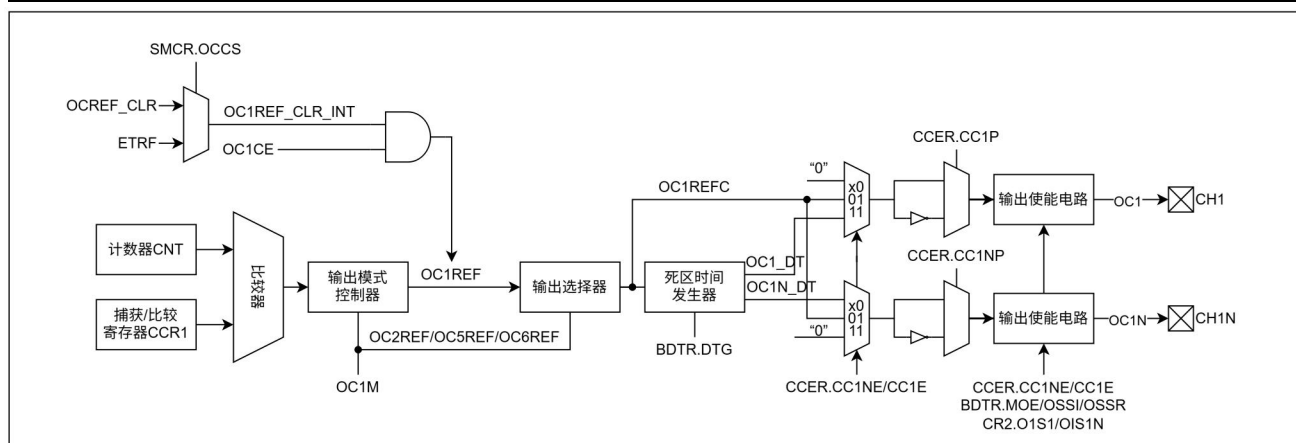


图 14-20 输出比较 1 通道

### 14.3.2 工作模式

GTIM 支持多种工作模式，具体由从模式控制寄存器 GTIM\_SMCR 的 SMS 位域来配置，如下表所示：

表 14-3 GTIM 工作模式

GTIM_SMCR.SMS 值	从模式选择
0000	禁止从模式，使用内部时钟
0111	外部时钟模式 1
0100	复位模式
0101	门控模式
0110	触发模式
1000	组合复位+触发模式
1001	组合门控+复位模式
1110	正交编码器模式，x1 模式
1111	正交编码器模式，x1 模式
0001	正交编码器模式，x2 模式
0010	正交编码器模式，x2 模式
0011	正交编码器模式，x4 模式
1010	编码模式（时钟+方向），x2 模式
1011	编码模式（时钟+方向），x1 模式
1100	编码模式（带方向时钟），x2 模式
1101	编码模式（带方向时钟），x1 模式

注 1：从模式选择位包括 GTIM\_SMCR[16] 和 GTIM\_SMCR[2:0]，需组合配置。

注 2：从模式选择位 SMS[3:0] 支持预装载功能。

#### 14.3.2.1 内部时钟模式

当从模式控制寄存器 GTIM\_SMCR 的 SMS 位域为 0x0 时，禁止定时器从模式，预分频器时钟直接由内部时钟 PCLK 提供。设置 GTIM\_CR1.CEN 为 1，将使能计数器开始计数。

### 14.3.2.2 外部时钟模式

#### 外部时钟模式 1

当从模式控制寄存器 GTIM\_SMCR 的 SMS 位域为 0x7 时,由所选触发信号 (TRGI) 的上升沿提供计数器时钟。

TRGI 信号有多种触发选择,具体通过 GTIM\_SMCR 寄存器的 TS 位域进行选择,如下表所示:

表 14-4 TRGI 信号来源

GTIM_SMCR.TS 位域值	TRGI 信号来源
111	外部触发输入(ETRF)
000xx/01xxx/10000	内部触发 (ITR0~12)
100	TI1 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)

ETR 输入信号来源可以是外部 GTIM ETR 引脚,也可以是片内其它外设,请参见 13.3.7 片内外设互联 ETR。选择 ETR 为 TRGI 信号源时,可通过 GTIM SMCR.ETP 选择外部触发极性,通过 GTIM SMCR.ETPS 设置预分频,通过 GTIM SMCR.ETF 进行滤波控制。

内部触发 ITR 来源为 BTIM 和 GTIM 的触发输出信号 TRGO 以及 UART 的 TXD/RXD 信号。

TI1 和 TI2 都具有滤波和边沿检测功能, TI1 和 TI2 分别通过捕获模式寄存器 GTIM CCMR1CAP 的 IC1F 和 IC2F 位域进行滤波控制,分别通过捕获/比较使能寄存器 GTIM CCER 的 CC1P/CC1NP 和 CC2P/CC2NP 位域进行边沿检测。外部时钟模式 1 连接示意图如下图所示:

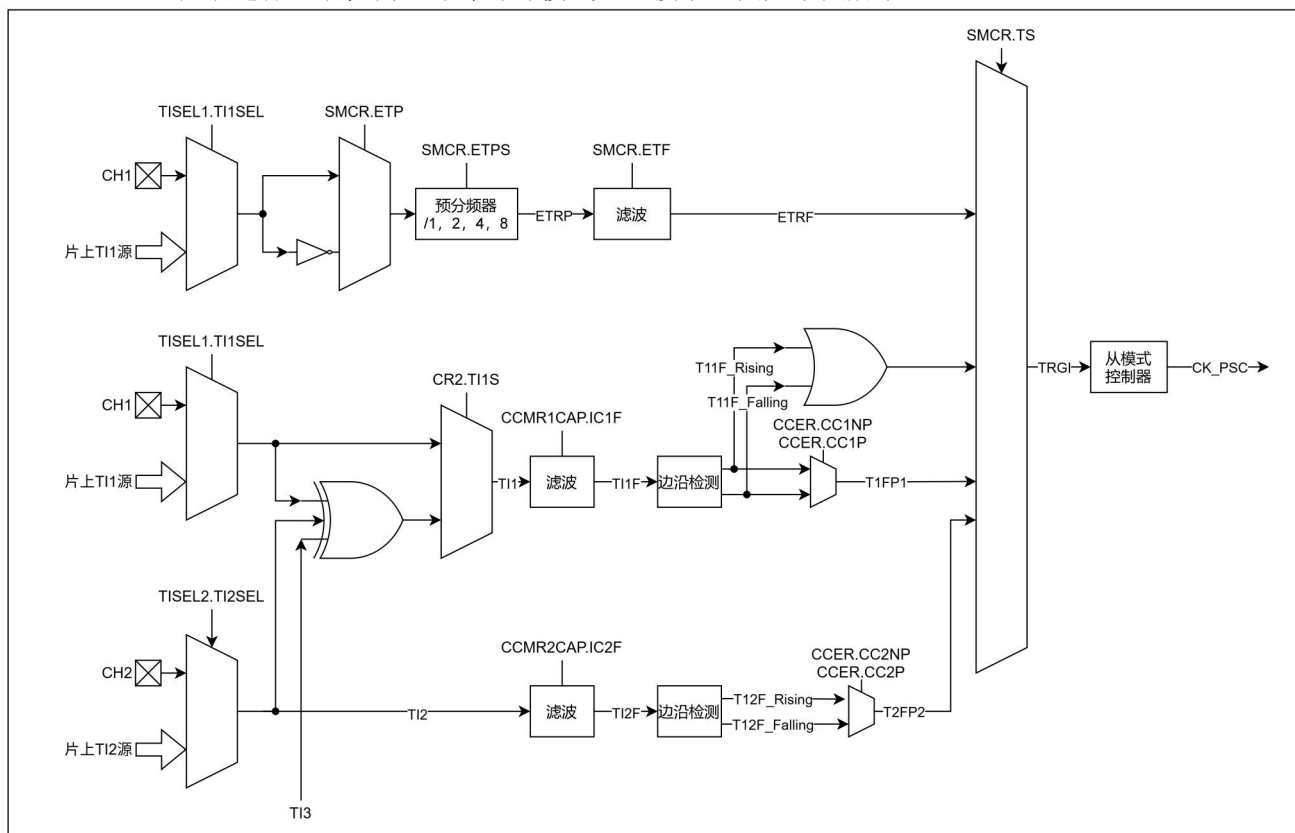


图 14-21 外部时钟模式 1 连接图

当 TRGI 出现有效边沿时, GTIM\_ISR.TIF 标志将置 1, 向 GTIM\_ICR.TIF 写 0 可清除该标志。

下图所示为配置 TI1 上升沿的外部时钟模式 1 时序图，TI1 的上升沿与实际计数器时钟之间的延迟是由于 TI1 输入的重新同步电路引起的。

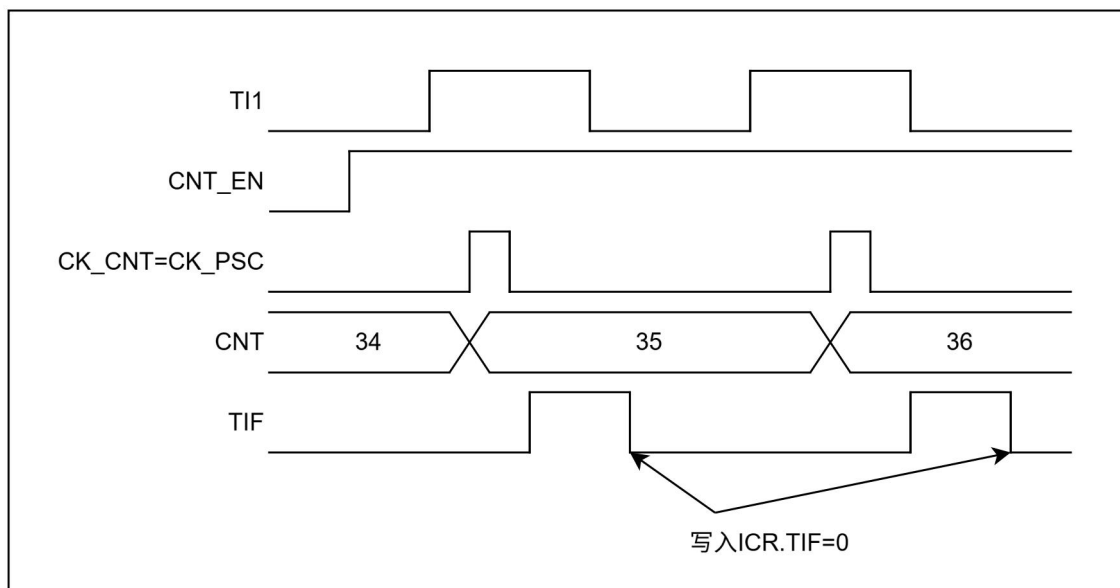


图 14-22 外部时钟模式 1 时序示例



## 外部时钟模式 2

设置从模式控制寄存器 GTIM\_SMCR 的 ECE 位域为 1，使能外部时钟模式 2。该模式下，计数器时钟由 ETRF 信号的任意有效边沿提供，与选择外部时钟模式 1 并将 TRGI 连接到 ETRF（SMS=0111 且 TS=00111）具有相同效果。

外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式，此时从模式下的 TRGI 不得连接 ETRF（即 TS 位域不能设置为 0x7）。如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 ETRF。外部时钟模式 2 不能与编码器模式同时使用。

下图所示为 ETR 设置为 2 分频时的外部时钟模式 2 的时序示例，ETR 的上升沿与实际计数器时钟之间的延迟是由 ETRP 信号的重新同步电路引起的，因此计数器可以正确捕获的最大频率最高为内部时钟 PCLK 频率的 1/4，当 ETRP 信号变快时，用户应对外部信号进行适当的分频。

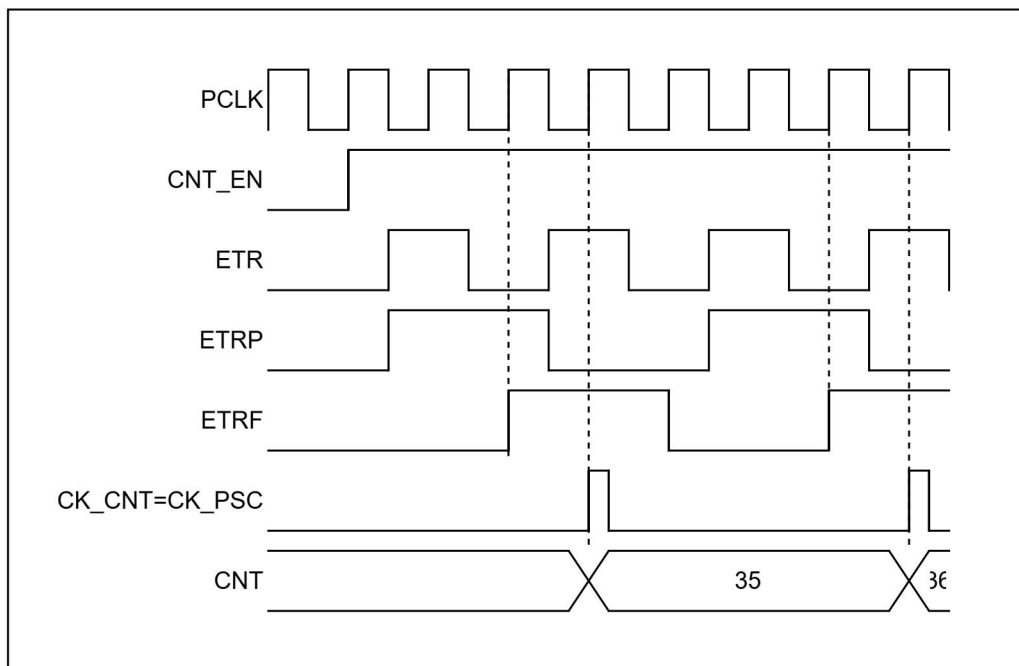


图 14-23 外部时钟模式 2 时序示例

### 14.3.2.3 复位模式

当从模式控制寄存器 GTIM\_SMCR 的 SMS 位域为 0x4 时，GTIM 配置为复位模式，计数器的复位由 TRGI 信号控制。

TRGI 信号的来源由从模式控制寄存器 GTIM\_SMCR 的 TS 位域控制。

当检测到有效的触发输入（TRGI）时，将产生以下影响：

- 重新初始化计数器和预分频器的计数器。
- 如果控制寄存器 GTIM\_CR1 的 URS 位域为 0，则会产生更新事件 UEV，事件更新中断标志 GTIM\_ISR.UIF 置 1，可产生中断请求。
- 触发中断标志 GTIM\_ISR.TIF 置 1，可产生中断请求。

下图所示为复位模式时序图示例。设置 GTIM\_CR1.CEN 为 1 使能计数器，计数器根据计数时钟 CK\_CNT 正常计数，当 TI1 出现上升沿时，计数器清零并重新从 0 开始计数，同时 GTIM\_ISR.TIF 标志位置 1。TI1 的上升沿与实际计数器复位之间的延迟是由 TI1 输入的重新同步电路引起的。

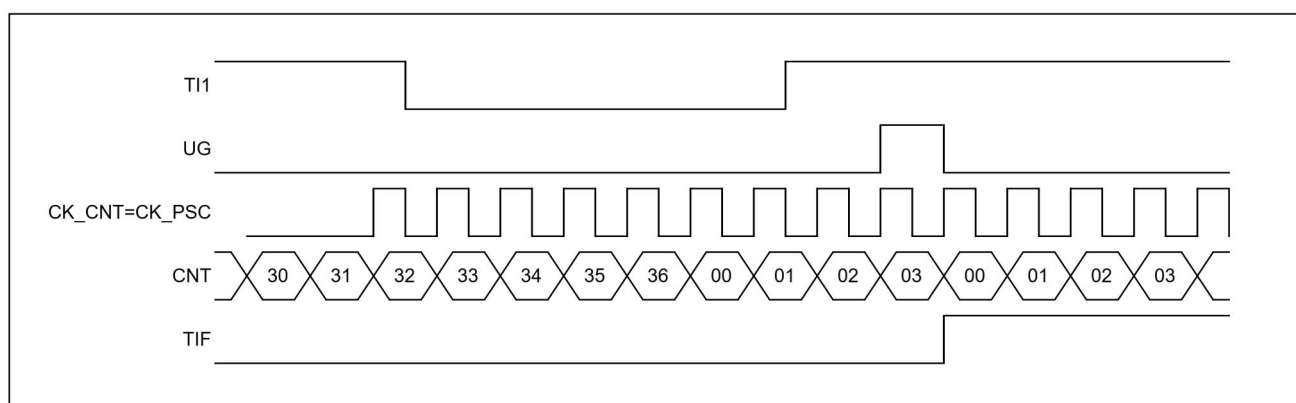


图 14-24 复位模式时序

### 14.3.2.4 门控模式

当从模式控制寄存器 GTIM\_SMCR 的 SMS 位域为 0x5 时，GTIM 配置为门控模式。在该模式下，触发输入 (TRGI) 为高电平且 GTIM\_CR1.CEN 为 1 时，启动计数器计数；触发输入 (TRGI) 为低电平或 GTIM\_CR1.CEN 为 0 时，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。

TRGI 信号的来源由从模式控制寄存器 GTIM\_SMCR 的 TS 位域控制，可参见表 13-4 TRGI 信号来源，但需注意门控模式下不能选择 TI1F\_ED 作为触发输入。

下图所示为门控模式时序图示例。当 TI1 为高电平时，计数器启动计数；当 TI1 为低电平时，计数器暂停计数。

计数器启动和停止时，GTIM\_ISR.TIF 标志位都会置 1。TI1 的边沿与实际计数器使能信号 CNT\_EN 之间的延迟是由 TI1 输入的重同步电路引起的。

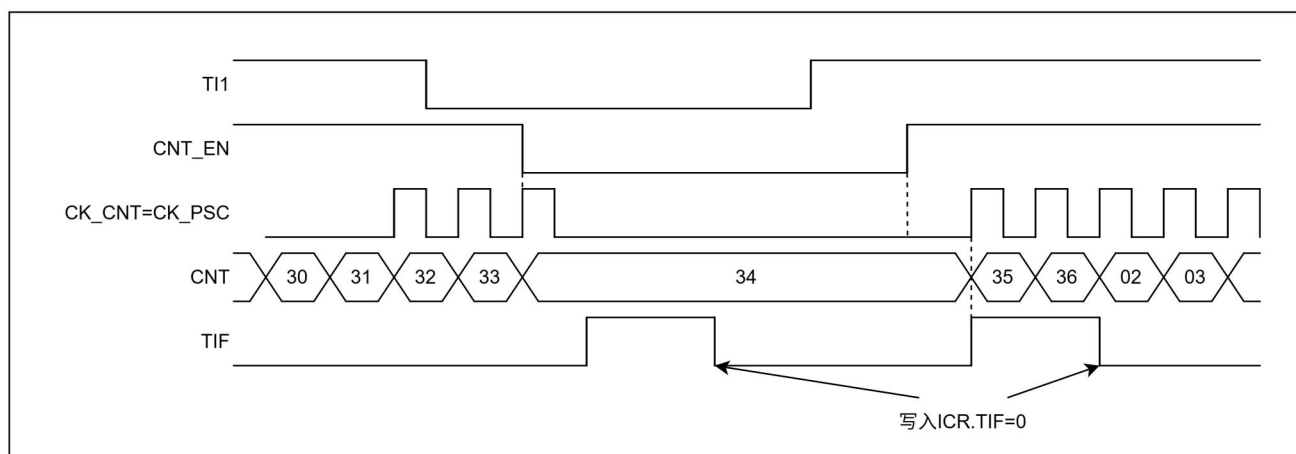


图 14-25 门控模式时序

### 14.3.2.5 触发模式

当从模式控制寄存器 GTIM\_SMCR 的 SMS 位域为 0x6 时，GTIM 配置为触发模式。在该模式下，设置 GTIM\_CR1.CEN 为 1 或触发信号 TRGI 出现上升沿时，触发启动计数器计数（但不复位）。

TRGI 信号的来源由从模式控制寄存器 GTIM\_SMCR 的 TS 位域控制。

当检测到有效的触发信号时，将产生以下影响：

- GTIM\_CR1.CEN 被硬件置位。
- 触发中断标志位 GTIM\_ISR.TIF 置 1，可产生中断请求。

- 计数器启动，开始计数。

下图所示为触发模式时序图示例。当 TI1 出现上升沿时，计数器启动计数，同时 GTIM\_ISR.TIF 标志位置 1。TI1 的上升沿与实际计数器启动之间的延迟是由 TI1 输入的重新同步电路引起的。

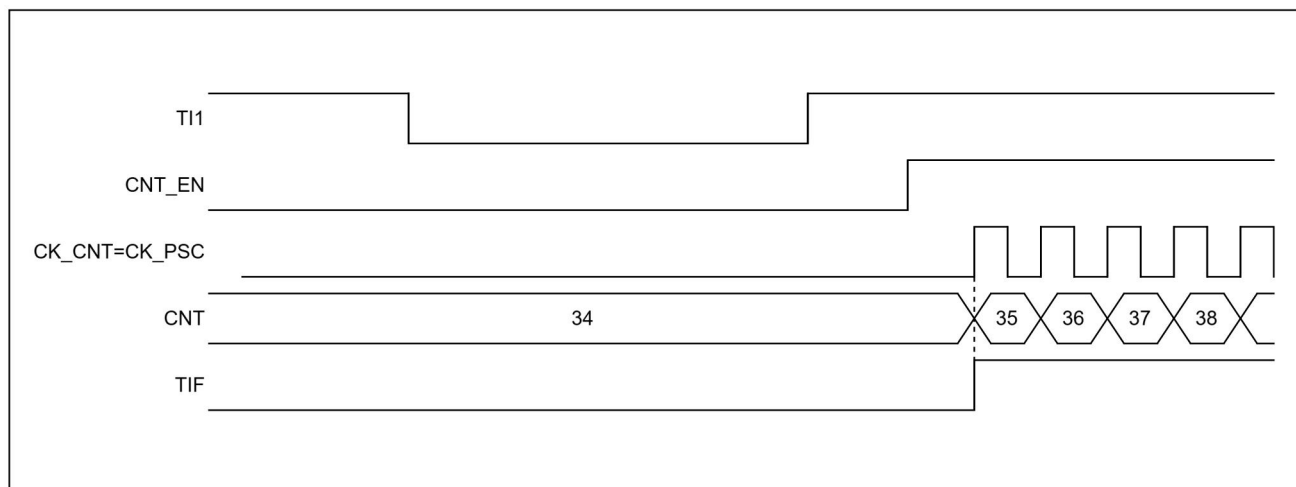


图 14-26 触发模式时序

#### 14.3.2.6 组合复位 + 触发模式

当从模式控制寄存器 GTIM\_SMCR 的 SMS 位域为 0x8 时，GTIM 配置为组合复位 + 触发模式。在该模式下，出现所选触发输入 (TRGI) 上升沿时，将重新初始化计数器并启动计数器，同时触发中断标志 GTIM\_ISR.TIF 置 1。如果控制寄存器 GTIM\_CR1 的 URS 位域为 0，则会产生更新事件 UEV，事件更新中断标志 GTIM\_ISR.UIF 会被硬件置位。

TRGI 信号的来源由从模式控制寄存器 GTIM\_SMCR 的 TS 位域控制。

#### 14.3.2.7 组合门控 + 复位模式

当从模式控制寄存器 GTIM\_SMCR 的 SMS 位域为 0x9 时，GTIM 配置为组合门控 + 复位模式。在该模式下，计数器的启动和停止都被控制，触发输入 (TRGI) 为高电平且 GTIM\_CR1.CEN 为 1 时，启动计数器计数；触发输入 (TRGI) 为低电平时，计数器立即停止计数，并被复位。计数器启动和停止时，GTIM\_ISR.TIF 标志位都会置 1；计数器复位时，GTIM\_ISR.UIF 标志位置 1。

TRGI 信号的来源由从模式控制寄存器 GTIM\_SMCR 的 TS 位域控制。

#### 14.3.2.8 正交编码器模式

GTIM 支持正交编码器模式，用于接收并解码正交编码器的信号。在该模式下，允许通过 CH1、CH2 引脚与外部的正交编码器直接连接，根据输入信号的跳变顺序，实现计数器自动递增或递减计数，计数器值始终表示编码器的位置。其功能框图如下图所示：

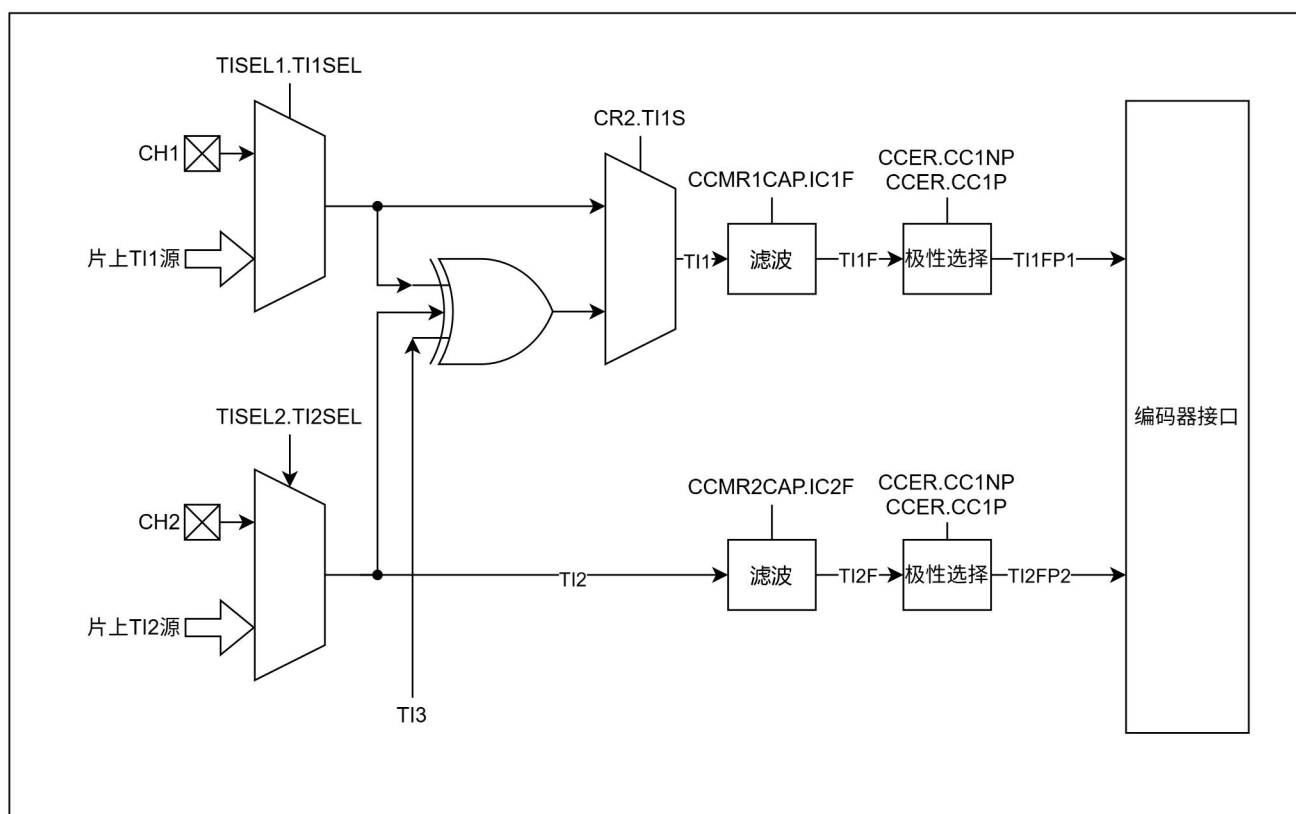


图 14-27 编码器模式框图

CH1 和 CH2 输入信号具有滤波和极性选择功能，分别通过 GTIM\_CCMR1CAP 寄存器的 IC1F 和 IC2F 位域进行滤波控制，通过 GTIM\_CCER 寄存器的 CC1P 和 CC2P 位域选择输入极性，CC1NP 和 CC2NP 位域必须保持清零。

设置 GTIM\_CR1.CEN 为 1 使能计数器，计数器将由通道 CH1 和 CH2 引脚输入信号经滤波和极性选择后的信号 TI1FP1 和 TI2FP2 的每次有效跳变驱动，根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号，参见表 14-5 计数方向和编码器信号的关系。编码器当前计数方向标志 GTIM\_CR1.DIR 由硬件自动设置和清除，且在任何输入（CH1 或 CH2）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 TI1FP1 或 TI2FP2 边沿处计数，还是同时在 TI1FP1 和 TI2FP2 处计数。

GTIM 支持多种正交编码计数模式，通过从模式控制寄存器 GTIM\_SMCR 的 SMS 位域进行设置，不同模式下计数方向和编码器信号的关系如下表所示：

表 14-5 计数方向和编码器信号的关系

模式	SMS	信号的电平		TI1FP1		TI2FP2	
		TI2FP2	TI1FP1	上升	下降	上升	下降
在 TI1FP1 边沿计数 (x1 模式)	1110	高	-	向下计数	向上计数	不计数	不计数
		低	-	不计数	不计数	不计数	不计数
在 TI2FP2 边沿计数 (x1 模式)	1111	-	高	不计数	不计数	向上计数	向下计数
		-	低	不计数	不计数	不计数	不计数
在 TI1FP1 边沿计数 (x2 模式)	1	高	-	向下计数	向上计数	不计数	不计数
		低	-	向上计数	向下计数	不计数	不计数
在 TI2FP2 边沿计数 (x2 模式)	10	-	高	不计数	不计数	向上计数	向下计数
		-	低	不计数	不计数	向下计数	向上计数
在 TI1FP1 和	11	高	高	向下计数	向上计数	向上计数	向下计数

TI2FP2 边沿计数 (x4 模式)		低	低	向上计数	向下计数	向下计数	向上计数
------------------------	--	---	---	------	------	------	------

编码器输出的第三个信号表示机械零点，可以把它连接到外部触发输入引脚上，用以触发计数器复位。

下图是一个正交编码计数模式 -x4 模式的操作实例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。

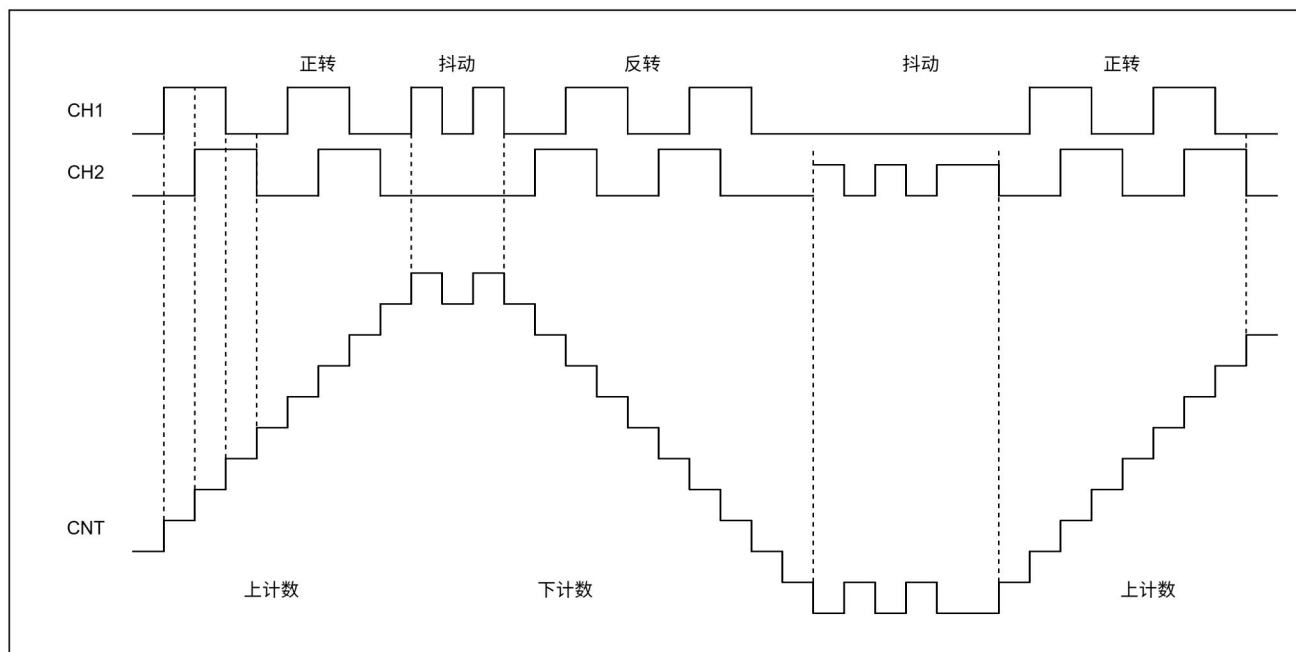


图 14-28 正交编码器模式 -x4 模式操作示例

下图显示了各种正交编码计数模式下编码器反转期间的定时器计数值，示例中 CH1 和 CH2 输入均不反相。

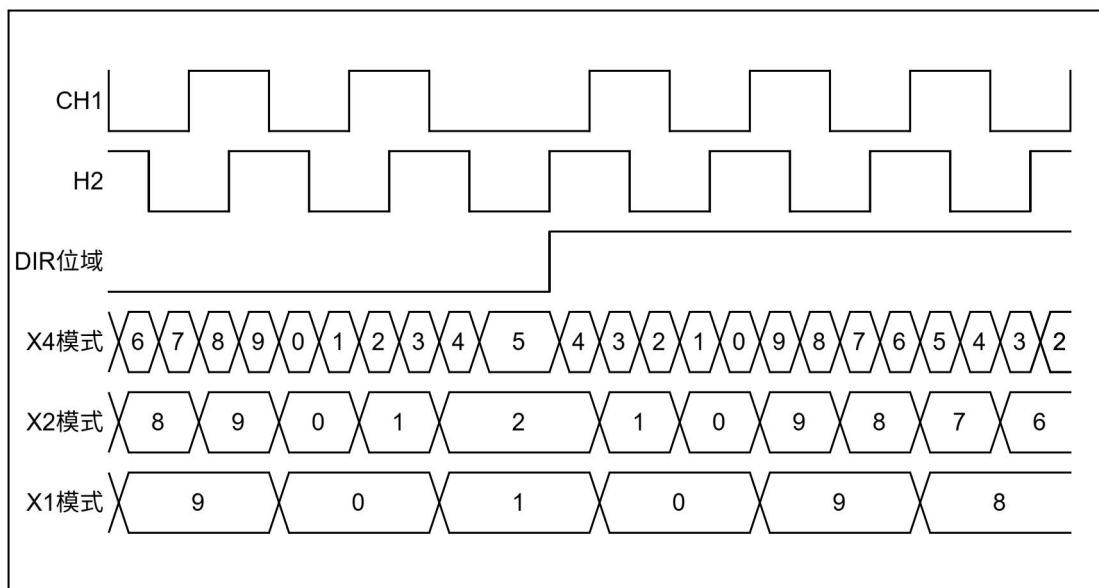


图 14-29 各正交编码器模式计数示例

编码器模式可以提供传感器当前位置的相关信息。计数器根据编码器输出的脉冲自动进行递增或递减计数，当前计数器值即表示编码器当前位置。使用另一个配置为捕获模式的定时器，可以捕获编码器信号的上升沿或下降沿，并记录其时间戳，通过测量两个事件之间的时间间隔，可以计算出相应的动态信息，如速度、加速度和减速度；可以使用指示机械零位的编码器输出来实现此目的，通过测量从机械零位到其他位置的时间间隔，可以计算出相对于零位的位置和速度变化。

#### 14.3.2.9 时钟加方向编码器模式

定时器还支持时钟加方向的编码器模式，需配置从模式控制寄存器 GTIM\_SMCR 的 SMS 位域为 0xA 或 0xB，分别对应 x2 模式和 x1 模式。其功能框图可参见图 13-27 编码器模式框图，可对输入信号进行滤波和极性控制。

带方向的编码器通常具有两个输出信号：时钟信号和方向信号，使用时钟信号可以计算出旋转编码器的旋转速度和位置变化，而方向信号可以用来确定旋转的方向。在 x2 模式，计数器在时钟信号的上升沿和下降沿计数；在 x1 模式，计数器根据 CC2P 位域的值在单个时钟边沿计数，CC2P 为 0 对应上升沿敏感，CC2P 为 1 对应下降沿敏感。

CH1 通道上方向信号的极性由 CC1P 位设置：CC1P 为 0 对应正极性（当 CH1 为高电平时递增计数，当 CH1 为低电平时递减计数），CC1P 为 1 对应负极性（当 CH1 为低电平时递增计数，当 CH1 为高电平时递减计数）。

下图显示了时钟加方向编码器模式下的计数器计数实例，示例中 CH1 和 CH2 输入均不反相：

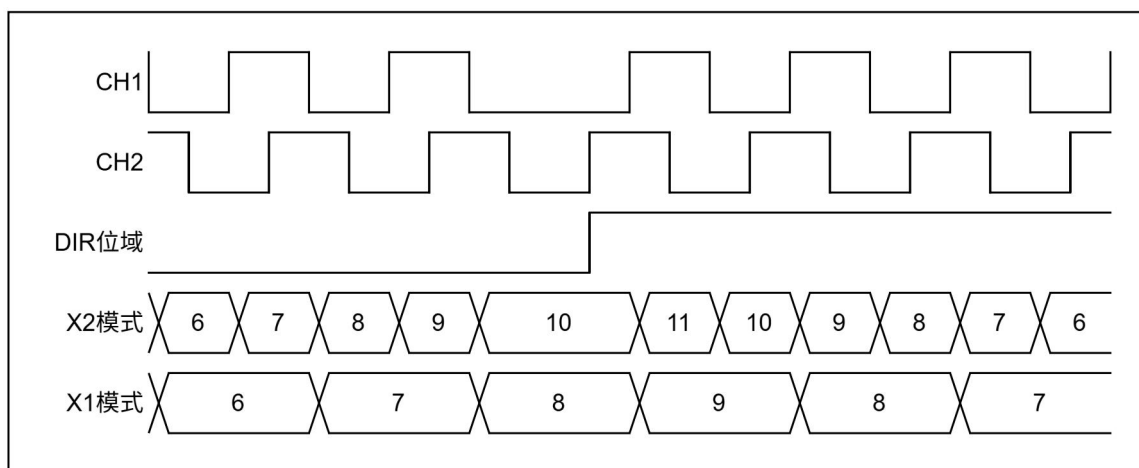


图 14-30 时钟加方向编码器模式计数示例

### 14.3.2.10 定向时钟编码器模式

定时器还支持定向时钟编码器模式，需配置从模式控制寄存器 GTIM\_SMCR 的 SMS 位域为 0xC 或 0xD，分别对应 x2 模式和 x1 模式。其功能框图可参见图 13-27 编码器模式框图，可对输入信号进行滤波和极性控制。

定向时钟编码器会根据旋转方向分别提供一条向上计数时钟线和向下计数时钟线。在 x2 模式，计数器在两个时钟线中的任意一个的上升沿和下降沿计数；CC1P 和 CC2P 位是时钟空闲状态的编码，CCyP 为 0 对应高电平空闲，CCyP 为 1 对应低电平空闲。在 x1 模式，计数器根据 CC1P 和 CC2P 位域的值在单个时钟边沿计数，CCyP 为 0 对应下降沿敏感和高电平空闲，CCyP 为 1 对应上升沿敏感和低电平空闲。

不同模式下计数方向和编码器信号的关系如下表所示：

表 14-6 计数方向和编码器信号的关系

模式	SMS	信号的电平		TI1FP1		TI2FP2	
		TI2FP2	TI1FP1	上升	下降	上升	下降
x2 模式	1100	高	高	向下计数	向下计数	向上计数	向上计数
CCyP=0		低	低	不计数	不计数	不计数	不计数
x2 模式		高	高	不计数	不计数	不计数	不计数
CCyP=1		低	低	向下计数	向下计数	向上计数	向上计数
x1 模式	1101	高	高	不计数	向下计数	不计数	向上计数
CCyP=0		低	低	不计数	不计数	不计数	不计数
x1 模式		高	高	不计数	不计数	不计数	不计数
CCyP=1		低	低	向下计数	不计数	向上计数	不计数

下图是定向时钟编码器模式的计数示例：

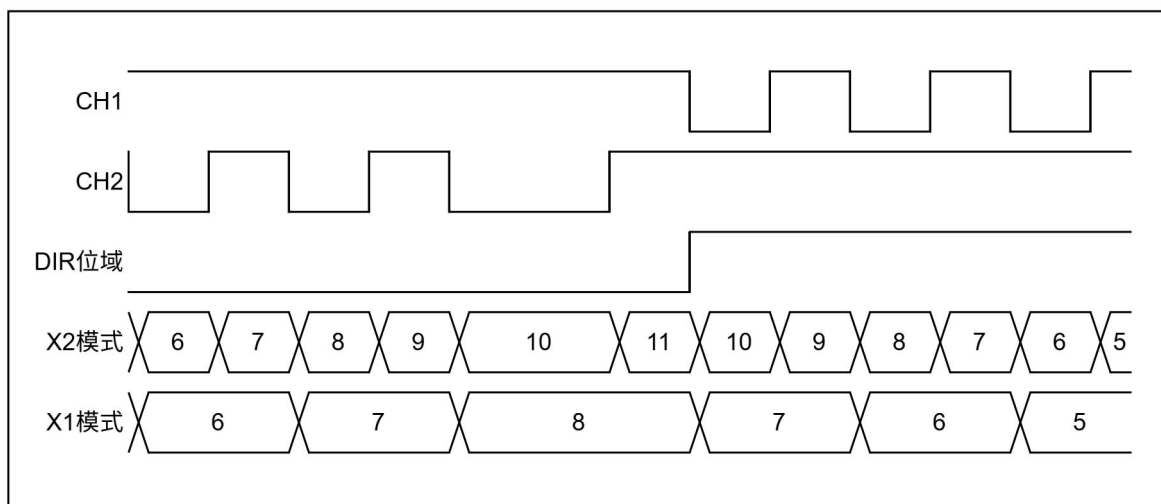


图 14-31 定向时钟编码器模式计数示例 ( $CC1P = CC2P = 0$ )

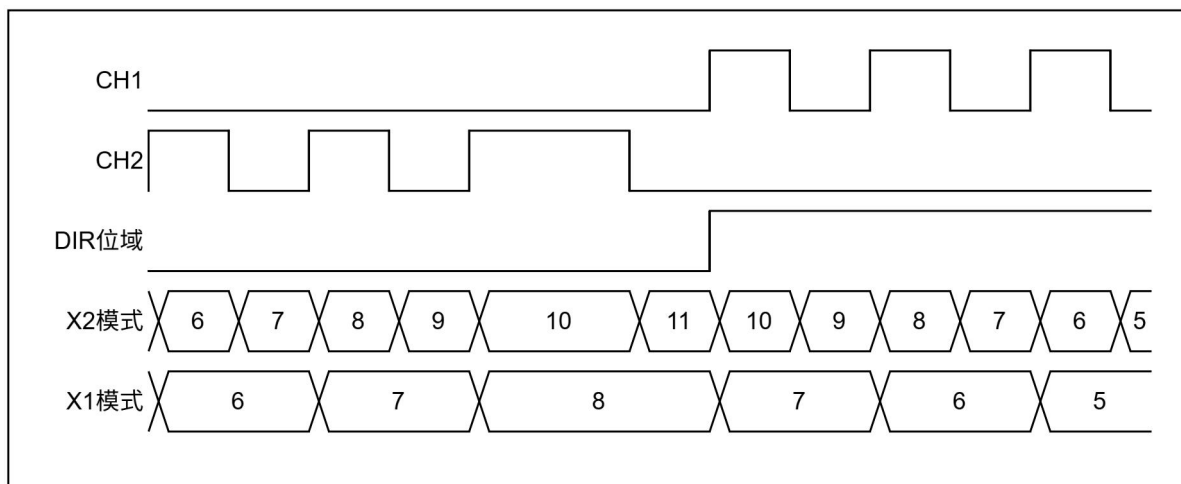


图 14-32 定向时钟编码器模式计数示例 ( $CC1P = CC2P = 1$ )

### 14.3.3 输入捕获功能

GTIM 支持输入捕获功能，设置捕获模式寄存器 (GTIM\_CCMR1CAP 和 GTIM\_CCMR2CAP) 的  $CCyS$  位域为非零值时， $CCy$  通道配置为输入，同时指定对应捕获通道  $ICy$  的输入映射。支持通过软件或硬件触发输入捕获，其功能框图如下图所示：



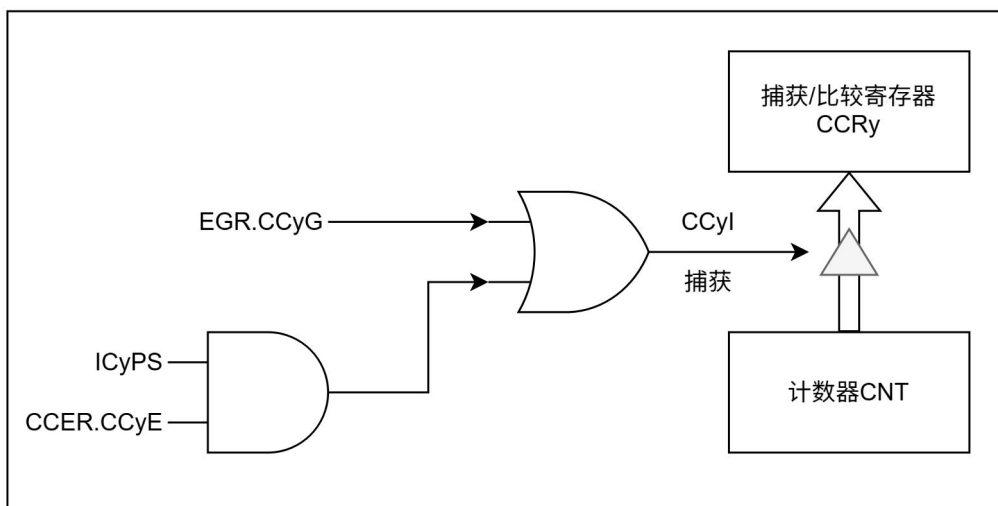


图 14-33 输入捕获模式框图

### 14.3.3.1 PWM 输入模式

PWM 输入模式是输入捕获模式的一个应用。输入捕获功能支持同一通道输入信号被两路捕获寄存器进行捕获，结合从模式复位功能，可方便测量 PWM 输入信号的周期和脉宽。

PWM 输入模式只能与 T1 和 TI2 信号配合使用，因为只有 TI1FP1 和 TI2FP2 与从模式控制器相连。

下图是 PWM 信号从 TI1 通道输入时的时序示例：

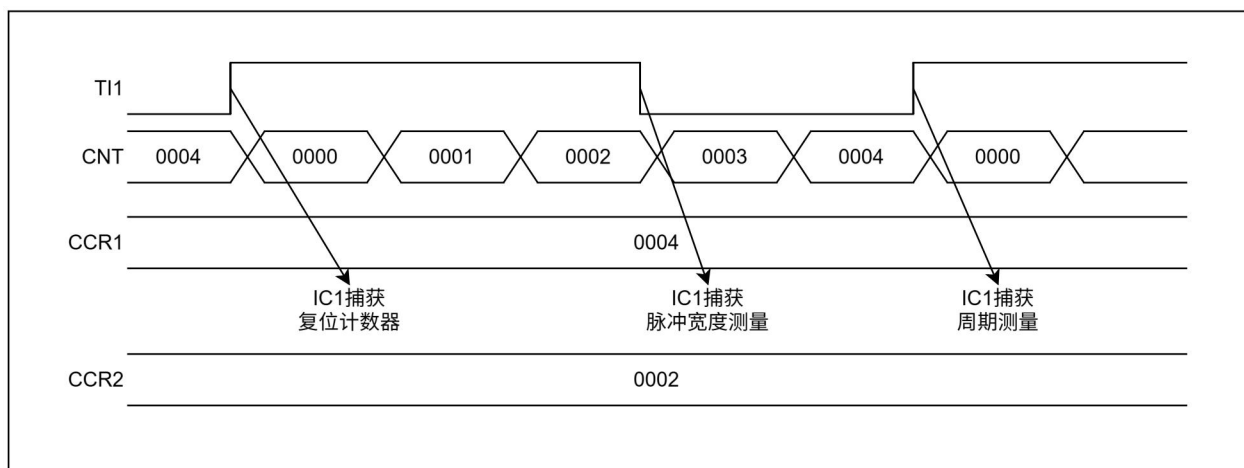


图 14-34 PWM 输入模式时序

### 14.3.3.2 输入捕获来源

GTIM 的输入捕获来源可以是外部 GTIM\_CHy 引脚，也可以是片内其它外设，通过 TI 输入选择寄存器 GTIM\_TISEL 的 TIySEL 位域进行配置。

当 GTIM\_TISEL.TIySEL 为 0x0 时，TIy 通道输入捕获信号来源为外部 GTIM\_CHy 引脚，此时需通过 GPIO 映射寄存器将对应引脚配置为复用功能。

当 GTIM\_TISEL.TIySEL 为 0x1~0xF 时，TIy 通道输入捕获信号来自片内其它外设。

### 14.3.4 输出比较功能

GTIM 支持输出比较功能，设置比较模式寄存器（GTIM\_CCMR1CMP 和 GTIM\_CCMR2CMP）的

CCyS 位域为 0 时，CCy 通道配置为输出。在输出比较模式下，当前计数器 CNT 的值与对应通道 CHy 的捕获 / 比较寄存器 CCRy 的值相比较，当两者匹配时，比较通道 CHy 输出为可设定的电平状态，同时产生比较中断。其功能框图如下图所示：

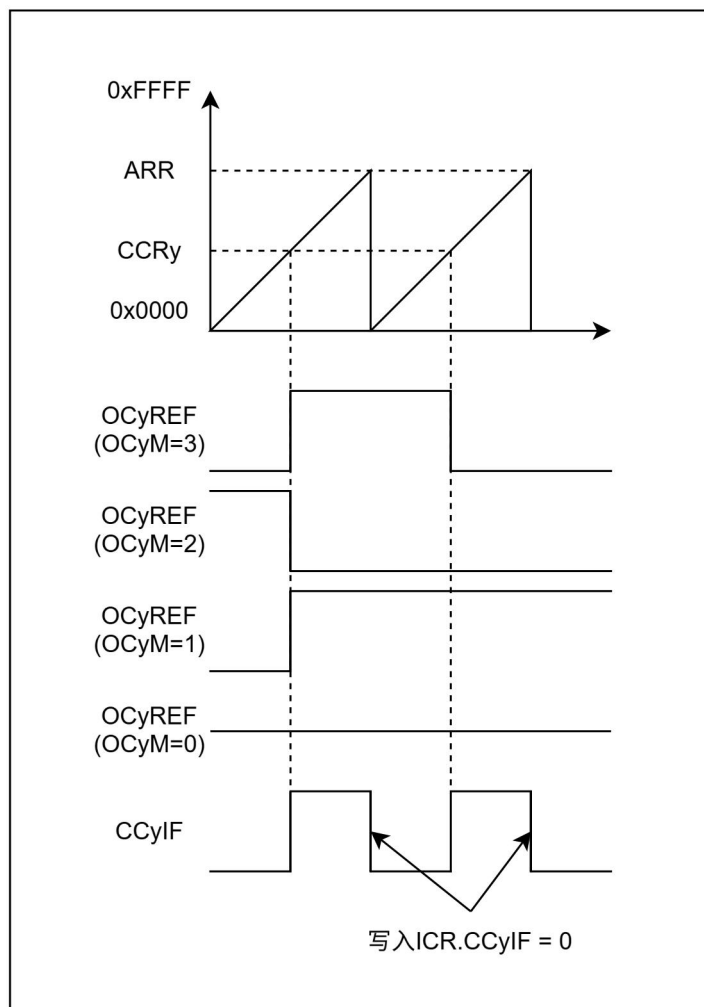


图 14-35 输出比较模式框图

CHy 引脚的输出动作由比较模式寄存器（GTIM\_CCMR1CMP 和 GTIM\_CCMR2CMP）的 OCyM 位域设置，如下表所示：

图 14-36 输出比较模式配置

OCyM 位域值	比较模式配置
0000	比较匹配时 OCyREF 保持原电平
0001	比较匹配时 OCyREF 置 1
0010	比较匹配时 OCyREF 置 0
0011	比较匹配时 OCyREF 翻转
0100	强制 OCyREF 为低电平
0101	强制 OCyREF 为高电平
0110	PWM 模式 1
0111	PWM 模式 2
1000	可再触发 OPM 模式 1

1001	可再触发 OPM 模式 2
1011	计数方向输出
1100	组合 PWM 模式 1
1101	组合 PWM 模式 2
1110	不对称 PWM 模式 1
1111	不对称 PWM 模式 2

当发生比较匹配时，对应通道的捕获 / 比较中断标志 GTIM\_ISR.CCyIF 被硬件置 1，同时：

- 如果使能了中断（设置 GTIM\_IER.CCyIE 为 1），将产生中断请求。
- 设置 GTIM\_ICR.CCyIF 为 0 可清除 GTIM\_ISR.CCyIF 标志位。

捕获 / 比较寄存器 GTIM\_CCRy 具有缓存功能，通过 OCyPE 位域选择是否使用缓存功能。当 OCyPE 为 0 时，禁止通道 CHy 的比较缓存功能，可在任意时候通过软件更新 GTIM\_CCRy 寄存器，更新值立即生效并影响输出波形；

当 OCyPE 为 1 时，使能通道 CHy 的比较缓存功能，更新 GTIM\_CCRy 寄存器不会立即生效，仅当发生更新事件 UEV 时才会将 GTIM\_CCRy 寄存器的值更新到有效寄存器。

#### 14.3.4.1 匹配输出

匹配输出模式用于控制输出波形，或指示已经过某一段时间。

- 设置 OCyM 位域为 0x0，比较匹配时，参考信号 OCyREF 保持其电平，同时 GTIM\_ISR.CCyIF 标志位置 1。
- 设置 OCyM 位域为 0x1，比较匹配时，参考信号 OCyREF 设置为高电平，同时 GTIM\_ISR.CCyIF 标志位置 1。
- 设置 OCyM 位域为 0x2，比较匹配时，参考信号 OCyREF 设置为低电平，同时 GTIM\_ISR.CCyIF 标志位置 1。
- 设置 OCyM 位域为 0x3，比较匹配时，参考信号 OCyREF 发生翻转，同时 GTIM\_ISR.CCyIF 标志位置 1。

下图是各比较匹配输出模式时序示例，示例为边沿对齐递增模式：

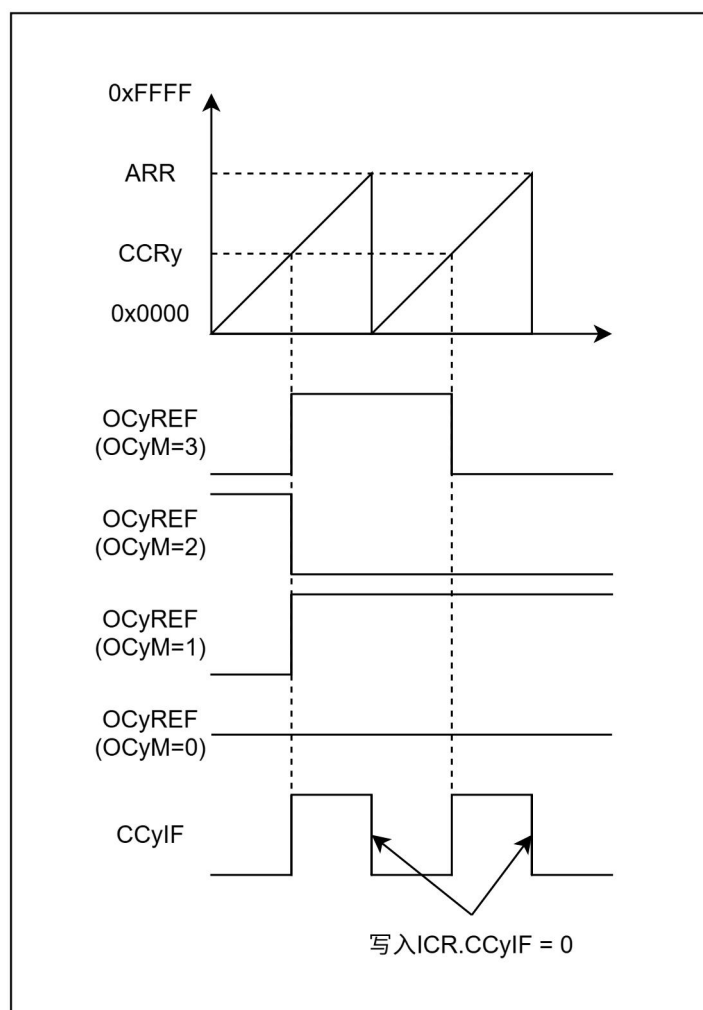


图 14-37 比较匹配输出模式

#### 14.3.4.2 强制输出

在强制输出模式下，输出比较信号能够直接由软件强置为高或低状态，而不依赖于捕获 / 比较寄存器 GTIM\_CCRy 和计数寄存器 GTIM\_CNT 的比较结果。

设置比较模式寄存器 (GTIM\_CCMR1CMP 和 GTIM\_CCMR2CMP) 的 OCyM 位域为 0x4，即可将参考信号 OCyREF 强制变为低电平；设置 OCyM 位域为 0x5，即可将参考信号 OCyREF 强制变为高电

平。

强制输出模式下，GTIM\_CCRy 寄存器和计数器 GTIM\_CNT 之间的比较仍然在进行，相应的标志也会置 1，也会产生相应的中断请求。

### 14.3.4.3 PWM 输出

脉冲宽度调制（PWM）模式可以产生一个由重载寄存器 GTIM\_ARR 确定频率、由捕获 / 比较寄存器 GTIM\_CCRy 确定占空比的信号。

向比较模式寄存器（GTIM\_CCMR1CMP 和 GTIM\_CCMR2CMP）的 OCyM 位域写入 0x6（PWM 模式 1）或 0x7（PWM 模式 2），能够独立地设置每个 CHy 输出通道产生一路 PWM，输出状态如下表所示：

表 14-7 PWM 模式 1/2 输出状态

工作模式	计数方向	PWM 模式 1	PWM 模式 2
边沿对齐模式	向上	CNT < CCRy 时，OCyREF 为高	CNT < CCRy 时，OCyREF 为低
中央对齐模式	向下	CNT > CCRy 时，OCyREF 为低	CNT > CCRy 时，OCyREF 为高

以下是不同计数模式下各 PWM 模式的波形实例，其中 GTIM\_ARR 为 0x08：

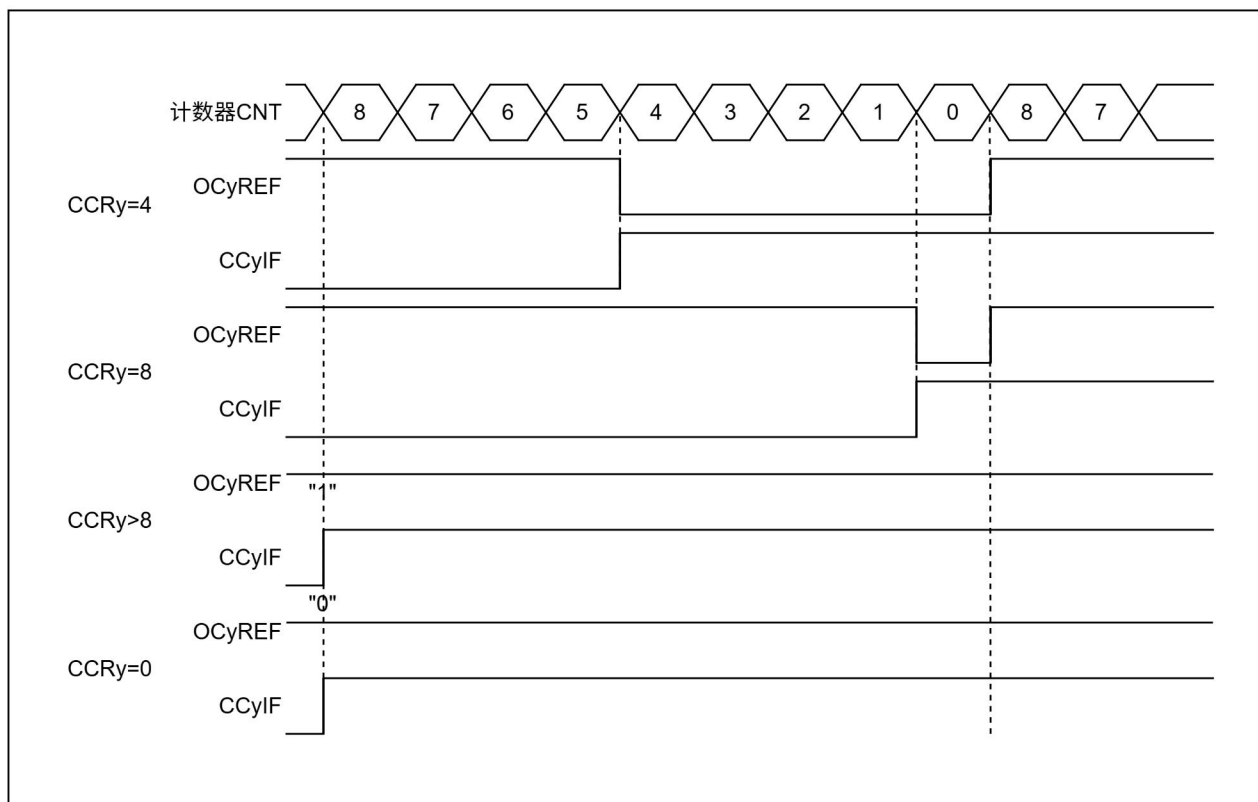


图 14-38 PWM 模式 1，边沿对齐模式，递增计数

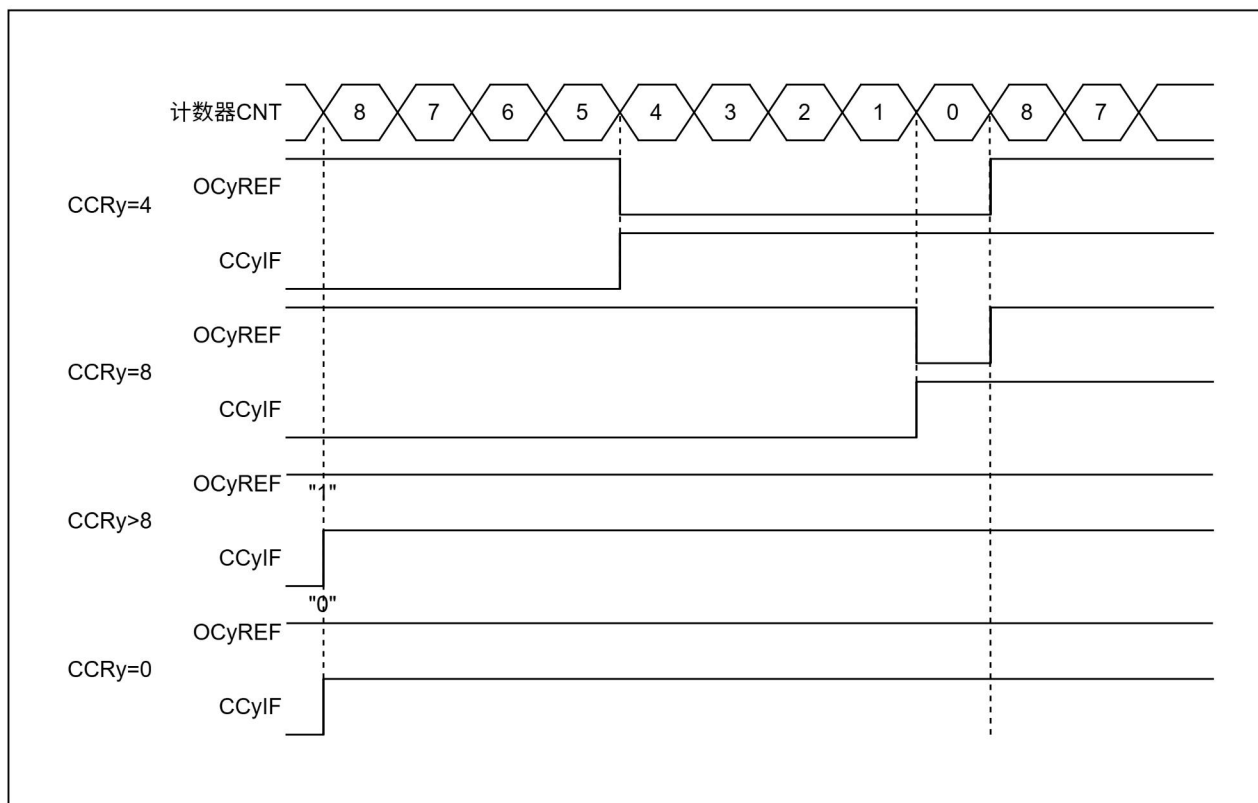


图 14-39 PWM 模式 1，边沿对齐模式，递减计数

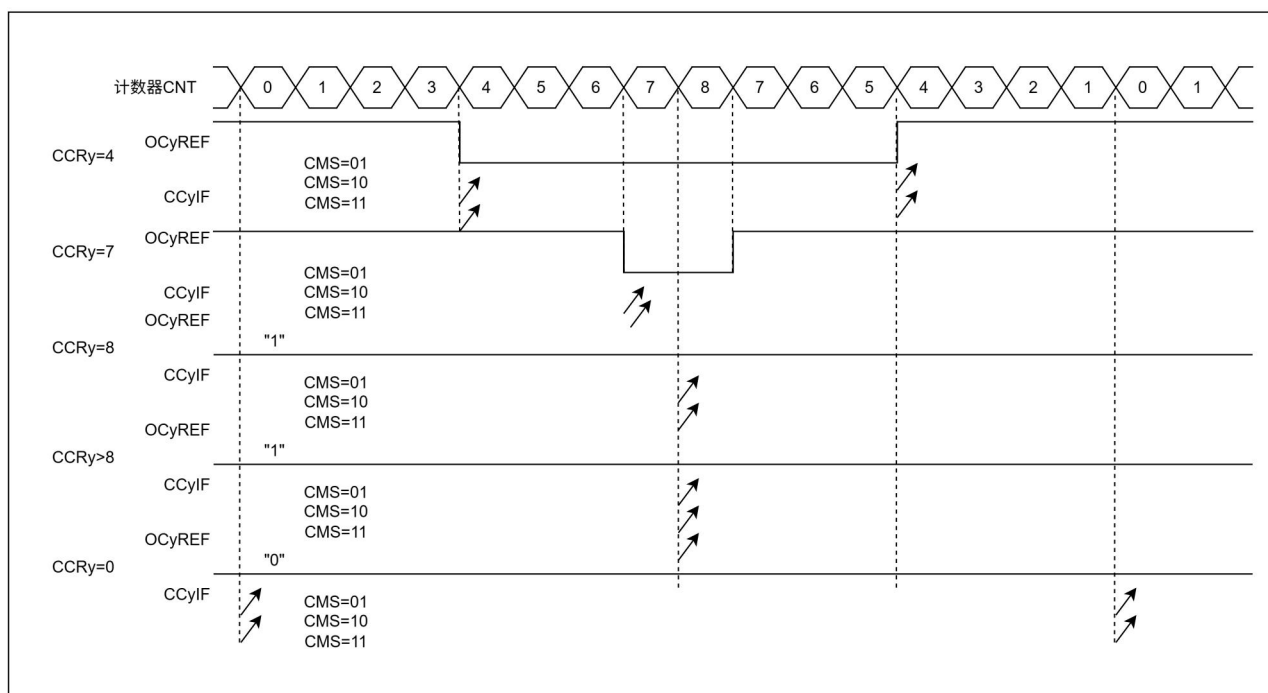


图 14-40 PWM 模式 1，中心对齐模式

#### 14.3.4.4 不对称 PWM 输出

不对称 PWM 输出是相对于中心对齐模式计数时的对称 PWM 输出而言的。在不对称 PWM 输出模式下，允许在中心对齐模式下生成两个具有可编程相移的 PWM 信号，频率由重载寄存器 GTIMARR 确定，占空比和相移由两个捕获/比较寄存器 GTIMCCRv 确定，其中一个 CCR 寄存器控制递增计数时的 PWM，另一个 CCRv 寄存器控制递减计数时的 PWM，具体如下所示：

- OC1REFC 由 CCR1(递增计数)与 CCR2(递减计数)控制
- OC2REFC 由 CCR2(递增计数)与 CCR1(递减计数)控制
- OC3REFC 由 CCR3(递增计数)与 CCR4(递减计数)控制
- OC4REFC 由 CCR4(递增计数)与 CCR3(递减计数)控制

向比较模式寄存器(GTIMCCMR1CMP 和 GTIMCCMR2CMP)的 OCyM 位域写入 0xE(不对称 PWM 模式 1)或 0xF(不对称 PWM 模式 2)，能够独立地设置每个 CHy 输出通道产生不对称 PWM。

当设定通道用作不对称 PWM 输出通道时，另一辅助通道仍可使用。例如，通道 CH1 配置为不对称 PWM 模式 1 产生 OC1REFC 信号时，通道 CH2 仍可以配置为 PWM 模式 2 输出 OC2REF 信号，或者配置为不对称 PWM 模式 2 输出 OC2REFC 信号。

以下是不对称 PWM 模式 1/2 的波形实例，其中 ARR=8，CCR1=0，CCR2=8，CCR3=3，CCR4=5：

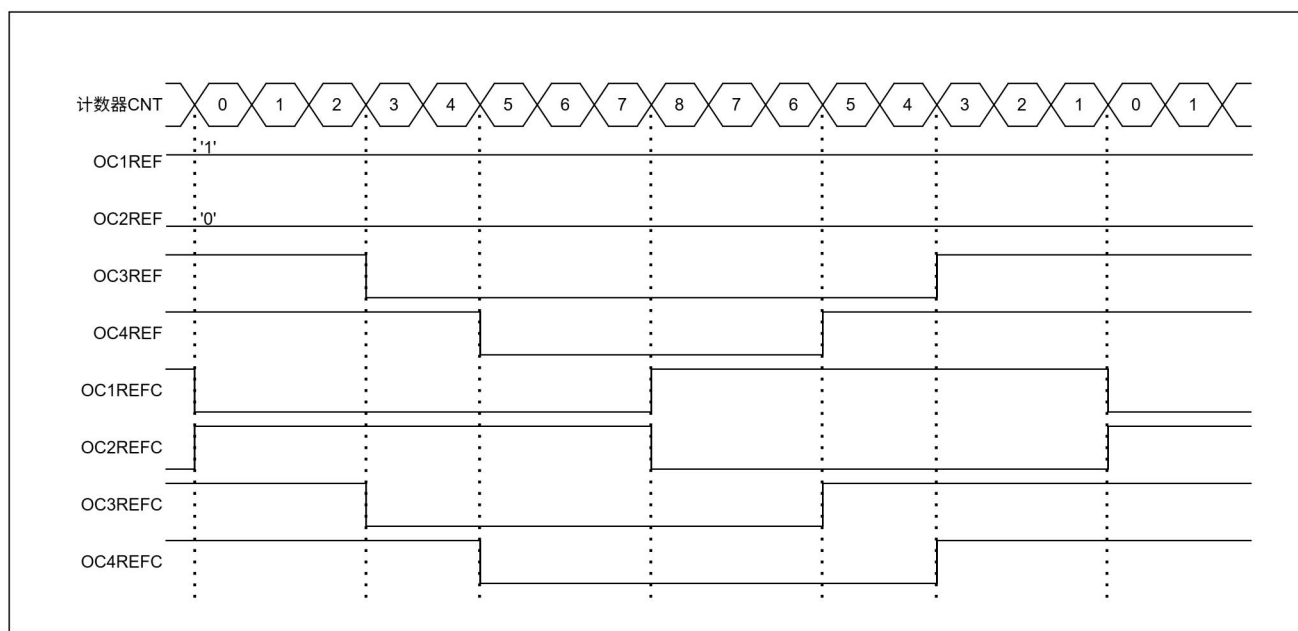


图 14-41 不对称 PWM 模式 1 示例



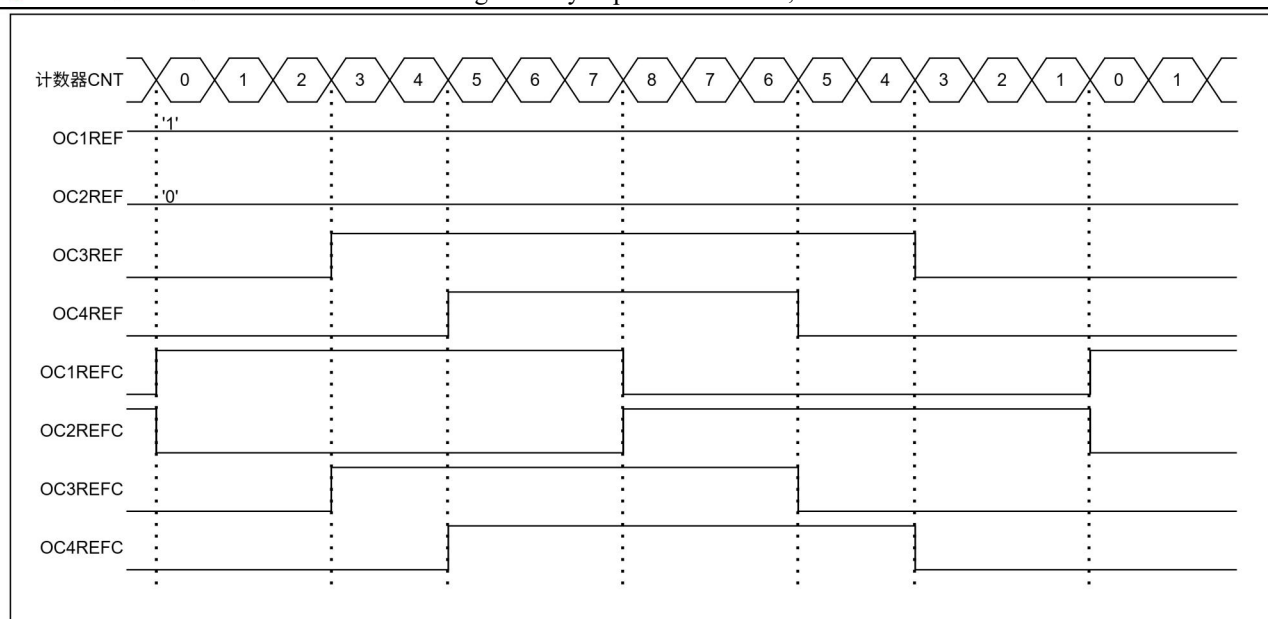


图 14-42 不对称 PWM 模式 2 示例

#### 14.3.4.5 组合 PWM 输出

在组合 PWM 输出模式下，允许在边沿对齐或中心对齐模式下生成两个具有可编程延时和相移的 PWM 脉冲，频率由重载寄存器 GTIM\_ARR 确定，占空比和延时由两个捕获 / 比较寄存器 GTIM\_CCRy 确定。

该模式下，OCyREFC 信号由两个参考信号 OCyREF 的逻辑或运算或者逻辑与运算组合生成，通过比较模式寄存器（GTIM\_CCMR1CMP 和 GTIM\_CCMR2CMP）的 OCyM 位域可独立选择组合 PWM 模式，具体如下表所示：

表 14-8 组合 PWM 模式配置

OCyM 位域值	输出状态
0xC (组合 PWM 模式 1)	OC1REFC (或 OC2REFC) 由 OC1REF 和 OC2REF 逻辑或运算生成
	OC3REFC (或 OC4REFC) 由 OC3REF 和 OC4REF 逻辑或运算生成
0xD (组合 PWM 模式 2)	OC1REFC (或 OC2REFC) 由 OC1REF 和 OC2REF 逻辑与运算生成
	OC3REFC (或 OC4REFC) 由 OC3REF 和 OC4REF 逻辑与运算生成

当设定通道用作组合 PWM 输出通道时，另一辅助通道仍可使用，但必须设置为相反的 PWM 模式。例如，通道 CH1 配置为组合 PWM 模式 1 产生 OC1REFC 信号时，通道 CH2 仍可使用，CH2 可以配置为 PWM 模式 2 输出 OC2REF 信号。

下图显示了边沿对齐递增模式下，组合 PWM 模式 1 的信号示例，具体配置如下：

- 重载值 ARR 为 8。
- 通道 1 配置为组合 PWM 模式 1，CCR1=2。
- 通道 2 配置为 PWM 模式 2，CCR2=3。
- 通道 3 配置为组合 PWM 模式 1，CCR3=6。
- 通道 4 配置为 PWM 模式 2，CCR4=7。

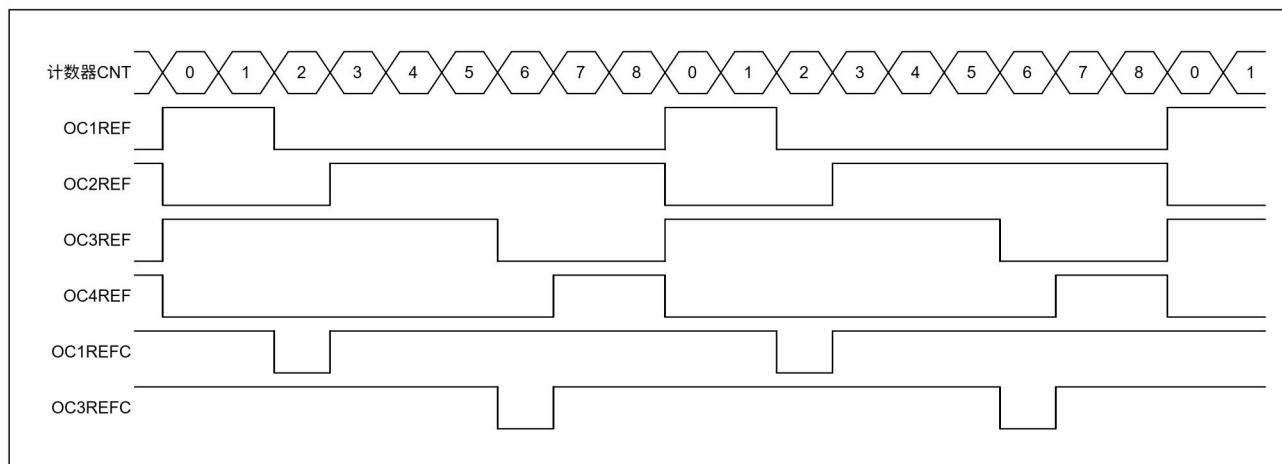


图 14-43 组合 PWM 模式 1 示例

下图显示了边沿对齐递增模式下，组合 PWM 模式 2 的信号示例，具体配置如下：

- 重载值 ARR 为 8。
- 通道 1 配置为组合 PWM 模式 2，CCR1=2。
- 通道 2 配置为 PWM 模式 1，CCR2=3。
- 通道 3 配置为组合 PWM 模式 2，CCR3=6。
- 通道 4 配置为 PWM 模式 1，CCR4=7。

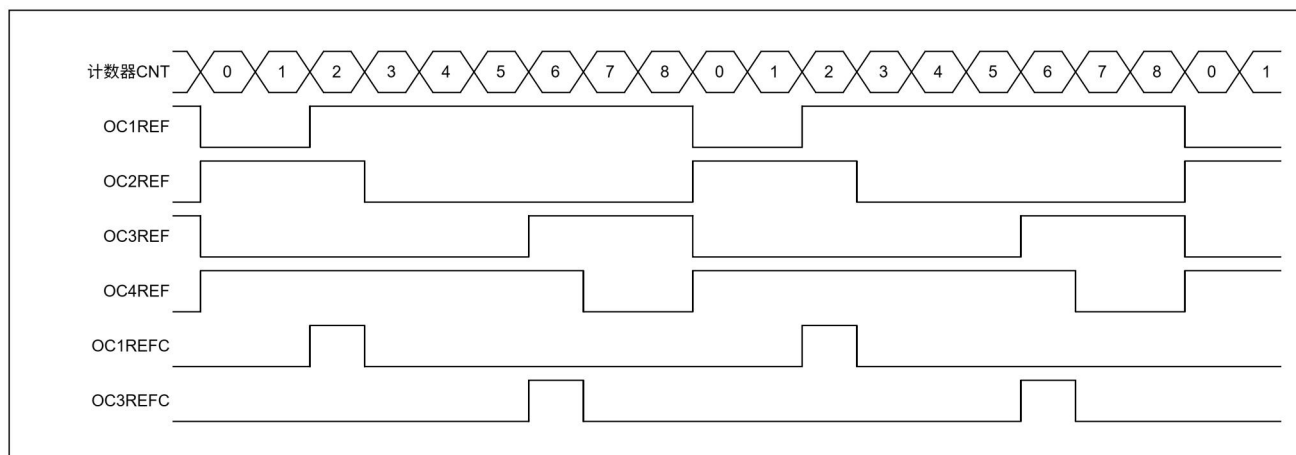


图 14-44 组合 PWM 模式 2 示例

#### 14.3.4.6 可再触发单脉冲模式

可再触发单脉冲模式配合组合复位 + 触发模式 (SMS=8)，允许在触发信号 (TRGI) 的触发下复位并启动计数器，并产生长度可编程的脉冲，脉冲宽度由 ARR 确定。

与单脉冲模式相比，具有如下区别：

表 14-9 可再触发单脉冲模式和单脉冲模式

单脉冲模式	可再触发单脉冲模式
发生触发时，经一段可编程的延时后产生一个脉宽可编程的单脉冲	发生触发时，脉冲立即产生，无可编程延时
新的触发无效	如果上一个触发产生的脉冲未完成，又发生新的触发，脉冲将延长

此模式下，计数器只能设置为边沿对齐模式，不能设置为中心对齐模式。当配置为递增计数模式时，对应 CCRy 必须设置为 0；配置为递减计数模式时，对应 CCRy 必须大于或等于 ARR。

下图所示为可再触发单脉冲模式 2 的示例：

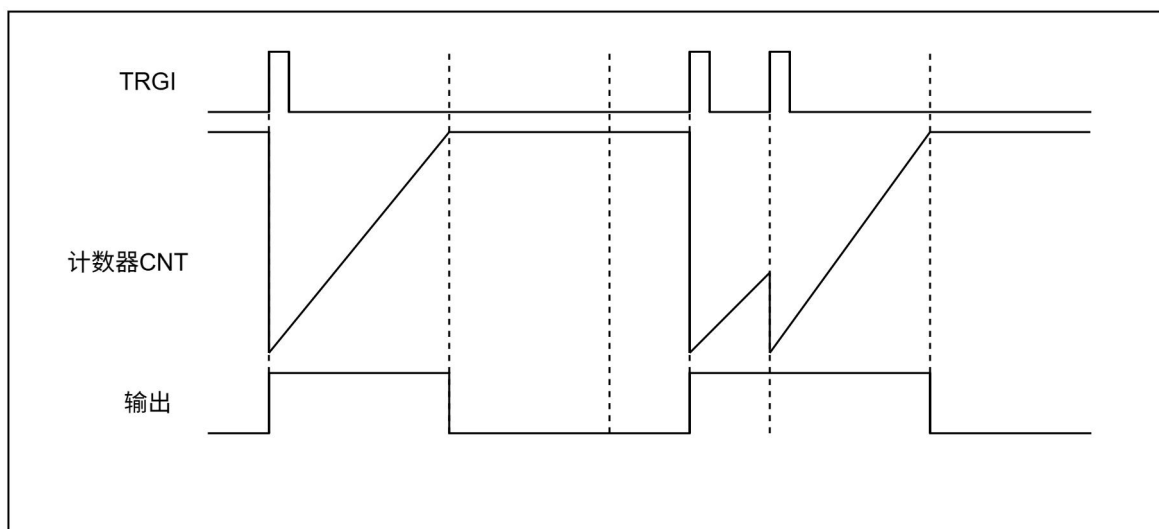


图 14-45 可再触发单脉冲模式 2 示例

#### 14.3.4.7 计数方向输出

将 GTIM\_CCMRxCMP 寄存器的 OCyM 位域设置为 0xB，可在对应的 CHy 通道上输出计数方向信号 (GTIM\_CR1 寄存器中 DIR 位的拷贝)。

此功能可用于在编码器模式下监控计数方向 (或旋转方向)，或在中心对齐 PWM 模式下指示向上 / 向下相位的信号。

#### 14.3.4.8 参考信号清零功能

设置 GTIM\_CCMRxCMP 寄存器的 OCyCE 位域为 1，可打开对应通道的 OCyREF 信号清零功能，即当 OCREF\_CLR\_INT 输入上检测到高电平时，可将 OCyREF 信号清零，OCyREF 信号将保持低电平，直到发生下一更新事件 UEV。该功能只在输出比较和 PWM 模式下可用，强制输出模式下不可用。

清零源 OCREF\_CLR\_INT 可选 OCREF\_CLR 或 ETRF，通过从模式控制寄存器 GTIM\_SMCR 的 OCCS 位域进行选择，OCREF\_CLR 也有多种输入源。

下图所示为 OCREF\_CLR\_INT 输入变为高电平时 OCyREF 信号状态：

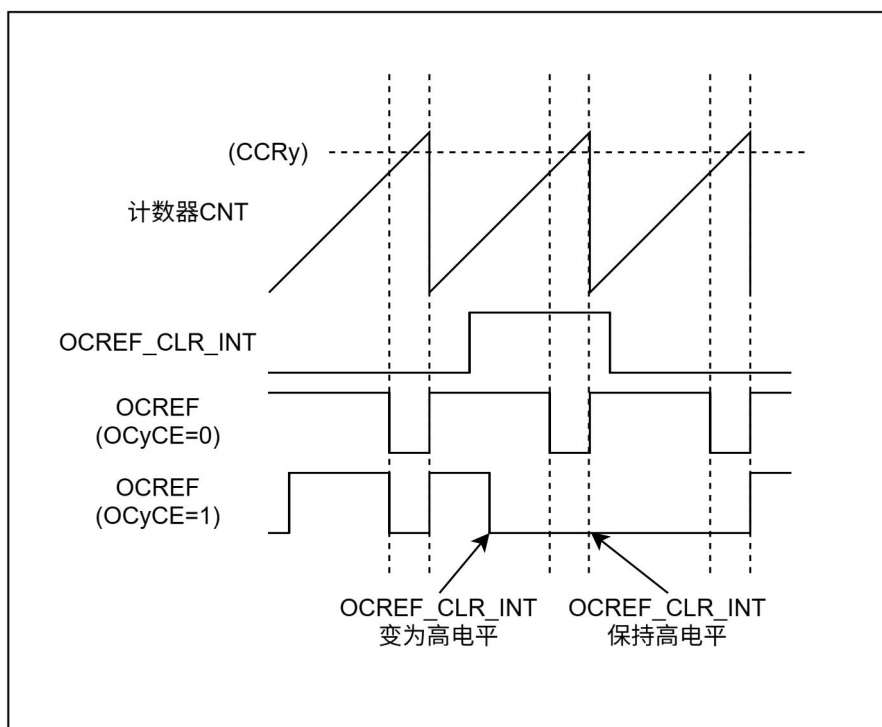


图 14-46 OCyREF 清零示例

### 14.3.5 UIF 位重映射

设置 GTIM\_CR1 寄存器中的 UIFREMAP 位为 1 可使能 UIF 状态位重映射功能，可强制将更新中断标志 GTIM\_ISR.UIF 连续复制到 GTIM\_CNT 寄存器的 UIFCPY 位域中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化角速度的计算。

UIF 和 UIFCPY 标志使能之间没有延迟。

当 GTIM\_CR1.UIFREMAP 为 0 时，UIFCPY 位域保留，读为 0。

### 14.3.6 定时器级联 ITR

通过 ITR 信号可以实现定时器级联，即将主定时器的触发输出（TRGO）连接到从定时器的 ITR 输入，进而连接到从定时器的触发输入（TRGI）。ITR 级联可实现多种功能，如将一个定时器用作另一个定时器的预分频器、主定时器对从定时器的计数器执行复位、使能或提供时钟等功能。

GTIM 的级联输入 ITR 的来源为 BTIM 和 GTIM 的触发输出信号（TRGO）以及 UART 的 TXD/RXD 信号，可通过从模式控制寄存器 GTIM\_SMCR 的 TS 位域来选择，具体配置如下表所示：

表 14-10 ITR 信号来源

GTIM_SMCR.TS 位域值	ITR 信号名称	ITR 信号来源
00000	ITR0	ATIM_Trigo
00001	ITR1	-
00010	ITR2	-
00011	ITR3	-
01000	ITR4	-
01001	ITR5	-
01010	ITR6	UART0_TXD

01011	ITR7	UART1_TXD
01100	ITR8	UART0_RXD
01101	ITR9	UART1_RXD
01110	ITR10	BTIM0_Trigo
01111	ITR11	BTIM1_Trigo
10000	ITR12	BTIM2_Trigo

通过控制寄存器 GTIM\_CR2 的 MMS 位域，可以选择 GTIM 主模式下将要发送到从定时器以实现同步的信息（TRGO），连接到其他定时器的 ITR 输入。如下表所示：

表 14-11 GTIM 触发输出（TRGO）

GTIM_CR2.MMS	TRGO 信号
00000	复位
00001	计数器使能信号 CNT_EN
00010	更新事件
00011	编码器时钟输出
00100	通道 1 比较捕获脉冲
00101	通道 2 比较捕获脉冲
00110	通道 3 比较捕获脉冲
00111	通道 4 比较捕获脉冲
01010	OC1REF 信号
01011	OC2REF 信号
01100	OC3REF 信号
01101	OC4REF 信号
10000	OC1REFC 信号
10001	OC2REFC 信号
10010	OC3REFC 信号
10011	OC4REFC 信号

### 14.3.7 片内外设互联 ETR

GTIM 的 ETR 信号来源可以是外部 GTIM\_ETR 引脚，也可以是片内其它外设，通过复用功能选项寄存器 GTIM\_AF1 的 ETRSEL 位域进行选择。

当 ETR 信号来自 GTIM\_ETR 引脚输入时，具体外部输入端口可参见表 8-2 GPIO 复用功能分配表，并需通过 GPIO 复用功能寄存器 (CPIOx\_AFRH 和 CPIOx\_AFRL) 进行复用配置；当 ETR 信号来自片内其它外设时，可实现片内外设互联。

ETR 信号来源如下表所示：

表 14-12 ETR 信号来源

AF1.ETRSEL 位域值	ETR 信号来源
0000	GTIM_ETR 引脚
0001	VC0_OUT
0010	VC0_OUT

0101	ADC_AWD
1101	UART0_TXD
1110	UART1_TXD

## 14.4 GTIM 中断

GTIM 支持 10 个中断源，当 GTIM 中断事件发生时，中断标志位会被硬件置位，如果设置了对应的中断使能控制位，将产生中断请求。

在用户 GTIM 中断服务程序中，应查询相关 GTIM 中断标志位，以进行相应的处理，在退出中断服务程序之前，要清除该中断标志位，以避免重复进入中断服务程序。

各 GTIM 中断源的标志位、中断使能位、中断标志清除位或清除方法，如下表所示：

表 14-13 GTIM 中断控制

中断事件	中断标志位	中断使能位	中断标志清除
更新中断	ISR.UIF	IER.UIE	写 0 到 ICR.UIF
捕获/比较 1 中断	ISR.CC1IF	IER.CC1IE	写 0 到 ICR.CC1IF 或读 CCR1 寄存器
捕获/比较 2 中断	ISR.CC2IF	IER.CC2IE	写 0 到 ICR.CC2IF 或读 CCR2 寄存器
捕获/比较 3 中断	ISR.CC3IF	IER.CC3IE	写 0 到 ICR.CC3IF 或读 CCR3 寄存器
捕获/比较 4 中断	ISR.CC4IF	IER.CC4IE	写 0 到 ICR.CC4IF 或读 CCR4 寄存器
触发中断	ISR.TIF	IER.TIE	写 0 到 ICR.TIF
编码器索引中断	ISR.IDXF	IER.IDXIE	写 0 到 ICR.IDXF
编码器方向改变中断	ISR.DIRF	IER.DIRIE	写 0 到 ICR.DIRF
编码器索引错误中断	ISR.IERRF	IER.IERRIE	写 0 到 ICR.IERRF
编码器转换错误中断	ISR.TERRF	IER.TERRIE	写 0 到 ICR.TERRF

## 14.5 触发 ADC

定时器支持多种内部信号触发启动 ADC，如捕获比较信号和触发输出信号 TRGO，触发输出 TRGO 有多种可能的事件，具体由控制寄存器 GTIM\_CR2 的 MMS 位域进行选择。同时，ADC 外设需配置其外部触发启动寄存器，以选择对应触发源。

应注意，必须先使能 ADC 时钟，才能从主定时器接收事件，且从定时器接收触发信号时，不得实时更改 ADC 时钟。

## 14.6 调试支持

GTIM 支持在调试模式下停止或继续计数，通过调试状态定时器控制寄存器 SYSCTRL\_DEBUG 的 GTIM 位域来设置：

- 设置 SYSCTRL\_DEBUG.GTIM 为 1，则在调试状态时暂停 GTIM 的计数器计数。
- 设置 SYSCTRL\_DEBUG.GTIM 为 0，则在调试状态时 GTIM 的计数器继续计数。

## 14.7 编程示例

### 14.7.1 外部时钟模式 1 编程示例

以下示例中，配置递增计数器在 GTIM\_CH1 通道的上升沿进行计数，步骤如下：

1. 设置 RCC\_AHBEN.GPIOx 为 1，RCC\_APBEN1.GTIM 为 1，打开 GTIM\_CH1 引脚对应的 GPIO 时钟和 GTIM 配置时钟及工作时钟；
2. 将 GTIM\_CH1 引脚映射到对应的 GPIO；
3. 设置 GTIM\_TISEL.TI1SEL 为 0，选择 TI1 来源为 GTIM\_CH1 通道；
4. 设置 GTIM\_CCMR1CAP.IC1F，配置输入滤波带宽；
5. 设置 GTIM\_CCER.CC1NP 为 0、GTIM\_CCER.CC1P 为 0，选择上升沿计数有效；
6. 设置 GTIM\_SMCR.TS 为 0x5，选择 TRGI 输入为 TI1FP1；
7. 设置 GTIM\_SMCR.SMS 为 0x7，使 GTIM 工作于外部时钟模式 1；
8. 设置 GTIM\_IER.UIE 为 1 并配置对应 NVIC，使能更新中断；
9. 配置计数器预分频器 GTIM\_PSC；
10. 设置期望的重载值 GTIM\_ARR；
11. 设置 GTIM\_CR1.CEN 为 1，使能计数器；
12. 当计数器溢出时，GTIM\_ISR.UIF 标志位置 1，进入中断服务程序，设置 GTIM\_ICR.UIF 为 0 清除该中断标志。

### 14.7.2 外部时钟模式 2 编程示例

以下是外部时钟模式 2+ 触发模式的编程示例，GTIM\_CH1 输入出现上升沿时触发启动计数器，计数器在 GTIM\_ETR 输入信号的每个上升沿计数，步骤如下：

1. 设置 SYSCTRL\_AHBEN.GPIOx 为 1，SYSCTRL\_APBEN1.GTIM 为 1，打开 GTIM\_ETR、GTIM\_CH1 引脚对应的 GPIO 时钟和 GTIM 配置时钟及工作时钟；
2. 将 GTIM\_ETR、GTIM\_CH1 引脚映射到对应的 GPIO；
3. 设置 GTIM\_AF1.ETRSEL 为 0，选择 ETR 来源为 GTIM\_ETR 引脚；
4. 设置 GTIM\_SMCR.ETP 为 0，选择 ETR 上升沿有效；
5. 设置 GTIM\_SMCR.ETPS 为 0，关闭 ETR 的预分频器；
6. 设置 GTIM\_SMCR.ETF 为 0，关闭 ETR 输入滤波；
7. 设置 GTIM\_SMCR.ECE 为 1，使能外部时钟模式 2；
8. 设置 GTIM\_TISEL.TI1SEL 为 0，选择 TI1 来源为 GTIM\_CH1 通道；
9. 设置 GTIM\_CCMR1CAP.IC1F 为 0，关闭 TI1 输入滤波；
10. 设置 GTIM\_CCER.CC1NP 为 0、GTIM\_CCER.CC1P 为 0，选择检测 TI1 上升沿；
11. 设置 GTIM\_SMCR.TS 为 0x5，选择 TRGI 输入为 TI1FP1；
12. 设置 GTIM\_SMCR.SMS 为 0x6，使 GTIM 工作于触发模式；
13. 当 GTIM\_CH1 输入出现上升沿时使能，计数器在 GTIM\_ETR 输入信号的每个上升沿计数。

### 14.7.3 复位模式编程示例

以下示例中，GTIM\_CH1 输入出现上升沿时重新初始化计数器，步骤如下：

1. 设置 SYSCTRL\_AHBEN.GPIOx 为 1，SYSCTRL\_APBEN1.GTIM 为 1，打开 GTIM\_CH1 引脚对应的 GPIO 时钟和 GTIM 配置时钟及工作时钟；
2. 将 GTIM\_CH1 引脚映射到对应的 GPIO；
3. 设置 GTIM\_TISEL.TI1SEL 为 0，选择 TI1 来源为 GTIM\_CH1 通道；



4. 设置 GTIM\_CCMR1CAP.IC1F, 配置输入滤波带宽;
5. 设置 GTIM\_CCER.CC1NP 为 0、GTIM\_CCER.CC1P 为 0, 选择检测上升沿;
6. 设置 GTIM\_SMCR.TS 为 0x5, 选择 TRGI 输入为 TI1FP1;
7. 设置 GTIM\_SMCR.SMS 为 0x4, 使 GTIM 工作于复位模式;
8. 设置期望的重载值 GTIM\_ARR;
9. 设置 GTIM\_CR1.CEN 为 1, 使能计数器, 开始正常计数;
10. 当 GTIM\_CH1 输入出现上升沿时, 计数器清零, 重新从 0 开始计数。

#### 14.7.4 门控模式编程示例

以下示例中, GTIM\_CH1 引脚输入的信号作为门控信号, 控制计数器计数, 步骤如下:

1. 设置 SYSCTRL\_AHBEN.GPIOx 为 1, SYSCTRL\_APBEN1.GTIM 为 1, 打开 GTIM\_CH1 引脚对应的 GPIO 时钟和 GTIM 配置时钟及工作时钟;
2. 将 GTIM\_CH1 引脚映射到对应的 GPIO;
3. 设置 GTIM\_TISEL.TI1SEL 为 0, 选择 TI1 来源为 GTIM\_CH1 通道;
4. 设置 GTIM\_CCMR1CAP.IC1F, 配置输入滤波带宽;
5. 设置 GTIM\_CCER.CC1NP 为 0、GTIM\_CCER.CC1P 为 0, 选择检测高电平;
6. 设置 GTIM\_SMCR.TS 为 0x5, 选择 TRGI 输入为 TI1FP1;
7. 设置 GTIM\_SMCR.SMS 为 0x5, 使 GTIM 工作于门控模式;
8. 设置期望的重载值 GTIM\_ARR;
9. 设置 GTIM\_CR1.CEN 为 1, 使能计数器;
10. 当 GTIM\_CH1 输入为高电平时, 计数器开始计数; 当 GTIM\_CH1 输入为低电平时, 计数器停止计数。

#### 14.7.5 触发模式编程示例

以下示例中, GTIM\_CH1 输入出现上升沿时触发启动计数器, 步骤如下:

1. 设置 SYSCTRL\_AHBEN.GPIOx 为 1, SYSCTRL\_APBEN1.GTIM 为 1, 打开 GTIM\_CH1 引脚对应的 GPIO 时钟和 GTIM 配置时钟及工作时钟;
2. 将 GTIM\_CH1 引脚映射到对应的 GPIO;
3. 设置 GTIM\_TISEL.TI1SEL 为 0, 选择 TI1 来源为 GTIM\_CH1 通道;
4. 设置 GTIM\_CCMR1CAP.IC1F, 配置输入滤波带宽;
5. 设置 GTIM\_CCER.CC1NP 为 0、GTIM\_CCER.CC1P 为 0, 选择检测上升沿;
6. 设置 GTIM\_SMCR.TS 为 0x5, 选择 TRGI 输入为 TI1FP1;
7. 设置 GTIM\_SMCR.SMS 为 0x6, 使 GTIM 工作于触发模式;
8. 设置期望的重载值 GTIM\_ARR;
9. 当 GTIM\_CH1 输入出现上升沿时, 计数器启动计数, 同时 TIF 标志置 1。

#### 14.7.6 正交编码器模式编程示例

以下示例中, 配置 GTIM 工作在正交编码器模式 -x4 模式, 步骤如下:

1. 设置 SYSCTRL\_AHBEN.GPIOx 为 1, SYSCTRL\_APBEN1.GTIM 为 1, 打开 GTIM\_CH1、GTIM\_CH2 引脚对应的 GPIO 时钟和 GTIM 配置时钟及工作时钟;
2. 将 GTIM\_CH1 和 GTIM\_CH2 引脚映射到对应的 GPIO;
3. 设置 GTIM\_TISEL.TI1SEL 为 0, 选择 TI1 来源为 GTIM\_CH1 通道;
4. 设置 GTIM\_TISEL.TI2SEL 为 0, 选择 TI2 来源为 GTIM\_CH2 通道;
5. 设置 GTIM\_CCMR1CAP.IC1F 和 GTIM\_CCMR1CAP.IC2F, 配置输入滤波带宽;

6. 设置 GTIM\_CCER.CC1NP、GTIM\_CCER.CC1P 为 0，CH1 输入不反相；
7. 设置 GTIM\_CCER.CC2NP、GTIM\_CCER.CC2P 为 0，CH2 输入不反相；
8. 设置 GTIM\_SMCR.SMS 为 0x3，使 GTIM 工作于正交编码器模式 -x4 模式；
9. 设置合适的重载值 GTIM\_ARR；
10. 设置 GTIM\_CR1.CEN 为 1，使能计数器。

## 14.7.7 输入捕获编程示例

### 14.7.7.1 基本输入捕获模式

以下示例中，GTIM\_CH1 输入出现上升沿时执行一次输入捕获，步骤如下：

1. 设置 SYSCTRL\_AHBEN,GPIOX 为 1,SYSCTRL\_APBEN1.GTIM 为 1,打开 GTIM CH1 引脚对应的 GPIO 时钟和 GTIM 配置时钟及工作时钟；
2. 将 GTIM CH1 引脚映射到对应的 GPIO；
3. 设置 GTIM\_TISEL.TI1SEL 为 0，选择 TI1 来源为 GTIM\_CH1 通道；
4. 设置 GTIM\_CCMR1CAP.IC1F，配置 TI1 输入滤波带宽；
5. 设置 GTIM\_CCER.CC1NP 为 0、GTIM\_CCER.CC1P 为 0，选择检测 IC1 上升沿；
6. 设置 GTIM\_CCMR1CAP.CC1S 为 1，CC1 通道配置为输入，IC1 映射到 TI1 上；
7. 设置 GTIM\_CCMR1CAP.IC1PSC 为 0，关闭 IC1 预分频器；
8. 设置 GTIM\_CCER.CC1E 为 1，使能输入捕获 1 模式；
9. 设置 GTIM\_IER.CC1IE 为 1 并配置对应 NVIC，使能输入捕获 1 中断；
10. 配置计数器预分频器 GTIM\_PSC；
11. 设置期望的重载值 GTIM\_ARR；
12. 设置 GTIM\_CR1.CEN 为 1，使能计数器，开始正常计数；
13. 当发生捕获时，当前计数器 CNT 的值被锁存到捕获/比较寄存器 CCR1 中，完成一次捕获，同时 GTIM\_ISR.CC1IF 标志位置 1，进入中断服务程序，设置 GTIM\_ICR.CC1IF 为 0 清除该中断标志。

### 14.7.7.2 PWM输入模式

以下示例测量从 GTIM\_CH1 引脚输入的 PWM 信号的周期和占空比，步骤如下：

1. 设置 SYSCTRL\_AHBEN.GPIOx 为 1,SYSCTRL\_APBEN1.GTIM 为 1,打开 GTIM\_CH1 引脚对应的 GPIO 时钟和 GTIM 配置时钟及工作时钟；
2. 将 GTIM\_CH1 映射到对应的 GPIO；
3. 设置 GTIM\_TISEL.TI1SEL 为 0，选择 TI1 来源为 GTIM\_CH1 通道；
4. 设置 GTIM\_CCMR1CAP.IC1F，配置 TI1 输入滤波带宽；
5. 设置 GTIM\_CCER.CC1NP 为 0、GTIM\_CCER.CC1P 为 0，选择检测 IC1 上升沿；
6. 设置 GTIM\_CCMR1CAP.CC1S 为 1，CC1 通道配置为输入，IC1 映射到 TI1 上；
7. 设置 GTIM\_CCMR1CAP.IC1PSC 为 0，关闭 IC1 预分频器；
8. 设置 GTIM\_CCER.CC1E 为 1，使能输入捕获 1 模式；
9. 设置 GTIM\_IER.CC1IE 为 1 并配置对应 NVIC，使能输入捕获 1 中断；
10. 设置 GTIM\_CCER.CC2NP 为 0、GTIM\_CCER.CC2P 为 1，选择检测 IC2 下降沿；
11. 设置 GTIM\_CCMR1CAP.CC2S 为 2，CC2 通道配置为输入，IC2 映射到 TI1 上；
12. 设置 GTIM\_CCMR1CAP.IC2PSC 为 0，关闭 IC2 预分频器；
13. 设置 GTIM\_CCER.CC2E 为 1，使能输入捕获 2 模式；
14. 设置 GTIM\_SMCR.TS 为 0x5，选择 TRGI 输入为 TI1FP1；
15. 设置 GTIM\_SMCR.SMS 为 0x4，使 GTIM 工作于复位模式；

16. 配置计数器预分频器 GTIM\_PSC;
17. 设置期望的重载值 GTIM\_ARR;
18. 设置 GTIM\_CR1.CEN 为 1, 使能计数器, 开始正常计数。

### 14.7.8 输出比较编程示例

以下示例中, 通道 CH1 对外输出设定的波形, 步骤如下:

1. 设置 SYSCTRL\_AHBEN.GPIOx 为 1, SYSCTRL\_APBEN1.GTIM 为 1, 打开 GTIM\_CH1 引脚对应的 GPIO 时钟和 GTIM 配置时钟及工作时钟;
2. 将 GTIM\_CH1 映射到对应的 GPIO;
3. 设置 GTIM\_CCER.CC1P 为 0, 选择输出极性;
4. 设置 GTIM\_CCMR1CMP.CC1S 为 0, CC1 通道配置为输出;
5. 设置 GTIM\_CCMR1CMP.OC1M, 选择输出比较 1 模式;
6. 设置 GTIM\_CCER.CC1E 为 1, 在相应输出引脚上输出 OC1 信号;
7. 配置计数器预分频器 GTIM\_PSC;
8. 设置期望的重载值 GTIM\_ARR;
9. 设置期望的比较值 GTIM\_CCR1;
10. 设置 GTIM\_IER.CC1IE 为 1 并配置对应 NVIC, 使能比较 1 中断;
11. 设置 GTIM\_CR1.CEN 为 1, 使能计数器;
12. 当发生比较匹配时, GTIM\_ISR.CC1IF 标志位置 1, 进入中断服务程序, 设置 GTIM\_ICR.CC1IF 为 0 清除该中断标志。

## 14.8 寄存器列表

GTIM 基地址: GTIM\_BASE = 0x4001 0400

表 14-14GTIM 寄存器列表

寄存器名称	寄存器地址	寄存器描述
GTIM_CR1	GTIM_BASE+0x00	控制寄存器 1
GTIM_CR2	GTIM_BASE+0x04	控制寄存器 2
GTIM_SMCR	GTIM_BASE+0x08	从模式控制寄存器
GTIM_IER	GTIM_BASE+0x0C	中断使能寄存器
GTIM_ISR	GTIM_BASE+0x10	中断标志寄存器
GTIM_ICR	GTIM_BASE+0x70	中断标志清除寄存器
GTIM_EGR	GTIM_BASE+0x14	事件生成寄存器
GTIM_CCMR1CAP	GTIM_BASE+0x18	捕获模式寄存器 1
GTIM_CCMR1CMP	GTIM_BASE+0x18	比较模式寄存器 1
GTIM_CCMR2CAP	GTIM_BASE+0x1C	捕获模式寄存器 2
GTIM_CCMR2CMP	GTIM_BASE+0x1C	比较模式寄存器 2
GTIM_CCER	GTIM_BASE+0x20	捕获/比较使能寄存器
GTIM_CNT	GTIM_BASE+0x24	计数寄存器
GTIM_PSC	GTIM_BASE+0x28	预分频寄存器
GTIM_ARR	GTIM_BASE+0x2C	自动重载寄存器
GTIM_CCR1	GTIM_BASE+0x34	捕获/比较寄存器 1
GTIM_CCR2	GTIM_BASE+0x38	捕获/比较寄存器 2
GTIM_CCR3	GTIM_BASE+0x3C	捕获/比较寄存器 3
GTIM_CCR4	GTIM_BASE+0x40	捕获/比较寄存器 4
GTIM_ECR	GTIM_BASE+0x58	编码控制寄存器
GTIM_TISEL	GTIM_BASE+0x5C	TI 输入选择寄存器
GTIM_AF1	GTIM_BASE+0x60	复用功能选项寄存器 1
GTIM_AF2	GTIM_BASE+0x64	复用功能选项寄存器 2

## 14.9 寄存器描述

### 14.9.1 GTIM\_CR1 控制寄存器 1

地址：请参见表 14-14 GTIM 寄存器列表      复位值：0x0000 0000

位域	名称	权限	功能描述
31-12	RFU	-	保留位，请保持默认值
11	UIFREMAP	RW	UIF 状态位重映射 0：禁止重映射，GTIM_CNT[31]保持为 0 1：使能重映射，GTIM_CNT[31]等效于 GTIM_ISR.UIF
10	RFU	-	保留位，请保持默认值
9-8	CKD	RW	配置 PCLK 时钟与滤波时钟分频比 00：f <sub>DTS</sub> =f <sub>PCLK</sub> /1 01：f <sub>DTS</sub> =f <sub>PCLK</sub> /2 10：f <sub>DTS</sub> =f <sub>PCLK</sub> /4 注：滤波时钟用于 ETR 滤波和 TI 滤波。
7	ARPE	RW	自动重载预装载使能 0：禁止 ARR 寄存器缓冲 1：使能 ARR 寄存器缓冲
6-5	CMS	RW	中心对齐模式选择 00：边沿对齐模式。计数器根据方向位(DIR)递增计数或递减计数。 01：中心对齐模式 1。计数器交替进行递增计数和递减计数。 仅当计数器递减计数时，配置为输出的通道（GTIM_CCMRx 寄存器中的 CCyS=00）的输出比较中断标志才置 1。 10：中心对齐模式 2。计数器交替进行递增计数和递减计数。 仅当计数器递增计数时，配置为输出的通道（GTIM_CCMRx 寄存器中的 CCyS=00）的输出比较中断标志才置 1。 11：中心对齐模式 3。计数器交替进行递增计数和递减计数。 当计数器递增计数或递减计数时，配置为输出的通道（GTIM_CCMRx 寄存器中的 CCyS=00）的输出比较中断标志都会置 1。 <b>注 1：只要计数器处于使能状态(CEN=1)，就不得从边沿对齐模式切换为中心对齐模式。</b> <b>注 2：中心对齐模式的初始计数方向由 DIR 位域决定。</b>
4	DIR	RW/RO	计数方向 0：计数器递增计数 1：计数器递减计数 注：当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。
3	OPM	RW	单脉冲模式 0：计数器在发生更新事件时不会停止计数 1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）
2	URS	RW	更新请求源 此位由软件置 1 和清零，用以选择 UEV 事件源。 0：使能 UEV 时，以下所有事件都会产生更新中断。包括：

			<ul style="list-style-type: none"> <li>● 计数器上溢/下溢</li> <li>● 将 UG 位置 1</li> <li>● 通过从模式控制器生成的更新事件</li> </ul> <p>1: 使能 UEV 时, 只有计数器上溢/下溢会生成更新中断。</p>
1	UDIS	RW	<p>更新禁止</p> <p>此位由软件置 1 和清零, 用以使能/禁止 UEV 事件生成。</p> <p>0: 使能 UEV。更新(UEV)事件可通过以下事件之一生成, 然后更新影子寄存器的值:</p> <ul style="list-style-type: none"> <li>● 计数器上溢/下溢</li> <li>● 将 UG 位置 1</li> <li>● 通过从模式控制器生成的更新事件</li> </ul> <p>1: 禁止 UEV。不会生成更新事件, 各影子寄存器的值 (ARR、PSC 和 CCRy) 保持不变。但如果将 UG 位置 1, 或者从从模式控制器接收到硬件复位, 则会重新初始化计数器和预分频器。</p>
0	CEN	RW	<p>计数器使能</p> <p>0: 禁止计数器</p> <p>1: 使能计数器</p> <p>注 1: 只有事先通过软件将 CEN 位置 1, 才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将 CEN 位置 1。</p> <p>注 2: 在单脉冲模式下, 当发生更新事件时会自动将 CEN 位清零。</p>

## 14.9.2 GTIM\_CR2 控制寄存器 2

地址：请参见表 14-14 GTIM 寄存器列表      复位值：0x0000 0000

位域	名称	权限	功能描述
31-27	RFU	-	保留位，请保持默认值
26-25	MMSH	RW	主模式选择 为 MMS 的最高位，需配合 MMS 使用
24-8	RFU	-	保留位，请保持默认值
7	TI1S	RW	TI1 输入选择 0: GTIM_CH1 引脚连接到 TI1 输入 1: GTIM_CH1、CH2 和 CH3 引脚的异或组合连接到 TI1 输入
6-4	MMS	RW	<p>主模式选择</p> <p>这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。配合 MMSH 使用，这些位的组合如下：</p> <p>00000: 复位——ATIM_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。</p> <p>00001: 使能——计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑与运算组合而成。当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主 / 从模式时除外。</p> <p>00010: 更新——选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器。</p> <p>00011: 编码器计数时钟输出</p> <p>00100: CH1 输出比较或输入捕获脉冲</p> <p>00101: CH2 输出比较或输入捕获脉冲</p> <p>00110: CH3 输出比较或输入捕获脉冲</p> <p>00111: CH4 输出比较或输入捕获脉冲</p> <p>01010: OC1REF 信号</p> <p>01011: OC2REF 信号</p> <p>01100: OC3REF 信号</p> <p>01101: OC4REF 信号</p> <p>10000: OC1REFC 信号</p> <p>10001: OC2REFC 信号</p> <p>10010: OC3REFC 信号</p> <p>10011: OC4REFC 信号</p> <p>注：必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。</p>
3-0	RFU	-	保留位，请保持默认值



### 14.9.3 GTIM\_SMCR 从模式控制寄存器

地址：请参见表 14-14GTIM 寄存器列表      复位值：0x0000 0000

位域	名称	权限	功能描述
31:26	RFU	-	保留位，请保持默认值
25	SMSPS	RW	预加载源 此位选择 SMS[3:0]位字段从预装载寄存器转移到有效寄存器的触发源 0：传输由计时器的更新事件触发 1：传输由索引事件触发
24	SMSPE	RW	预加载使能 此位选择 SMS[3:0]位是否预装载 0：禁止预装载 1：使能预装载
23:22	RFU	-	保留位，请保持默认值
21:20	TSH	RW	触发选择 为 TS 的最高位，需配合 TS 使用
19:17	RFU	-	保留位，请保持默认值
16	SMSH	RW	从模式选择 为 SMS 的最高位，需配合 SMS 使用
15	ETP	RW	外部触发极性 此位可选择将 ETR 还是 ETR 的反相用于触发操作 0：ETR 未反相，高电平或上升沿有效 1：ETR 反相，低电平或下降沿有效
14	ECE	RW	外部时钟使能 此位可使能外部时钟模式 2 0：禁止外部时钟模式 2 1：使能外部时钟模式 2，计数器时钟由 ETRF 信号的任意有效边沿提供。 注 1：将 ECE 位置 1 与选择外部时钟模式 1 并将 TRGI 连接到 ETRF(SMS=111 且 TS=00111) 具有相同效果。 注 2：外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF (TS 位不得为 00111)。 注 3：如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 ETRF。
13:12	ETPS	RW	外部触发预分频器 外部触发信号 ETRP 频率不得超过 PCLK 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。 00：预分频器关闭 01：2 分频 ETRP 频率 10：4 分频 ETRP 频率 11：8 分频 ETRP 频率
11:08	ETF	RW	外部触发滤波器 此位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波器带宽。数



			<p>字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿。</p> <p>0000: 无滤波器，按 <math>f_{DTS}</math> 频率进行采样</p> <p>0001: <math>f_{SAMPLING}=f_{DTS}/1</math>, <math>N=2</math></p> <p>0010: <math>f_{SAMPLING}=f_{DTS}/1</math>, <math>N=4</math></p> <p>0011: <math>f_{SAMPLING}=f_{DTS}/1</math>, <math>N=8</math></p> <p>0100: <math>f_{SAMPLING}=f_{DTS}/2</math>, <math>N=6</math></p> <p>0101: <math>f_{SAMPLING}=f_{DTS}/2</math>, <math>N=8</math></p> <p>0110: <math>f_{SAMPLING}=f_{DTS}/4</math>, <math>N=6</math></p> <p>0111: <math>f_{SAMPLING}=f_{DTS}/4</math>, <math>N=8</math></p> <p>1000: <math>f_{SAMPLING}=f_{DTS}/8</math>, <math>N=6</math></p> <p>1001: <math>f_{SAMPLING}=f_{DTS}/8</math>, <math>N=8</math></p> <p>1010: <math>f_{SAMPLING}=f_{DTS}/16</math>, <math>N=5</math></p> <p>1011: <math>f_{SAMPLING}=f_{DTS}/16</math>, <math>N=6</math></p> <p>1100: <math>f_{SAMPLING}=f_{DTS}/16</math>, <math>N=8</math></p> <p>1101: <math>f_{SAMPLING}=f_{DTS}/32</math>, <math>N=5</math></p> <p>1110: <math>f_{SAMPLING}=f_{DTS}/32</math>, <math>N=6</math></p> <p>1111: <math>f_{SAMPLING}=f_{DTS}/32</math>, <math>N=8</math></p>
7	MSM	RW	<p>主/从模式</p> <p>0: 不执行任何操作</p> <p>1: 当前定时器的触发输入事件(TRGI)的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于由单个外部事件对多个定时器进行同步的情况。</p>
6:04	TS	RW	<p>触发输入 TRGI 选择</p> <p>此位域可选择将要用于同步计数器的触发输入。</p> <p>00000: 内部触发 0(ITR0)</p> <p>00001: 内部触发 1(ITR1)</p> <p>00010: 内部触发 2(ITR2)</p> <p>00011: 内部触发 3(ITR3)</p> <p>00100: TI1 边沿检测器(TI1F_ED)</p> <p>00101: 滤波后的定时器输入 1(TI1FP1)</p> <p>00110: 滤波后的定时器输入 2(TI2FP2)</p> <p>00111: 外部触发输入(ETRF)</p> <p>01000: 内部触发 4(ITR4)</p> <p>01001: 内部触发 5(ITR5)</p> <p>01010: 内部触发 6(ITR6)</p> <p>01011: 内部触发 7(ITR7)</p> <p>01100: 内部触发 8(ITR8)</p> <p>01101: 内部触发 9(ITR9)</p> <p>01110: 内部触发 10(ITR10)</p> <p>01111: 内部触发 11(ITR11)</p> <p>10000: 内部触发 12(ITR12)</p> <p>其他值: 保留</p> <p>注 1: 这些位只能在未使用的情况下（例如，SMS=0000 时）进行更改，以避免转换时出现错误的边沿检测。</p>

			<p>注 2: 其他位位于同一寄存器的位 21、20, 需配合 TS 使用。</p> <p>注 3: 具体 ITRx 来源请参见表 13-13 ITR 信号来源。</p>
3	OCCS	RW	<p>OCREF 清零选择</p> <p>该位用于选择 OCREF 清零源。</p> <p>0: OCREF_CLR_INT 连接到 OCREF_CLR 输入</p> <p>1: OCREF_CLR_INT 连接到 ETRF</p> <p>注: OCREF_CLR 输入源请参见 13.9.23 GTIM_AF2 复用功能选项寄存器 2 的 OCRSEL 位域说明。</p>
2-0	SMS	RW	<p>从模式选择</p> <p>选择外部信号时, 触发信号(TRGI)的有效边沿与外部输入上所选的极性相关(请参见输入相关控制寄存器)。</p> <p>0000: 禁止从模式—如果 CEN=“1”, 预分频器时钟直接由内部时钟提供。</p> <p>0001: 正交编码器模式—x2 模式, 计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。</p> <p>0010: 正交编码器模式—x2 模式, 计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。</p> <p>0011: 正交编码器模式—x4 模式, 计数器在 TI1FP1 和 TI2FP2 的边沿计数, 计数的方向取决于另外一个输入的电平。</p> <p>0100: 复位模式—在出现所选触发输入(TRGI)上升沿时, 重新初始化计数器并生成一个寄存器更新事件。</p> <p>0101: 门控模式—触发输入(TRGI)为高电平时使能计数器时钟。只要触发输入变为低电平, 计数器立即停止计数(但不复位)。计数器的启动和停止都被控制。</p> <p>0110: 触发模式—触发信号(TRGI)出现上升沿时启动计数器(但不复位)。只控制计数器的启动。</p> <p>0111: 外部时钟模式 1—由所选触发信号(TRGI)的上升沿提供计数器时钟。</p> <p>1000: 组合复位+触发模式—在出现所选触发输入(TRGI)上升沿时, 重新初始化计数器, 生成一个寄存器更新事件并启动计数器。</p> <p>1001: 组合门控+复位模式—当触发输入(TRGI)为高电平时计数器被使能并开始计数, 当触发输入变为低电平时计数器停止计数并被复位, 在此模式计数器的启动和停止都被控制。</p> <p>1010: 编码模式—时钟+方向, x2 模式。</p> <p>1011: 编码模式—时钟+方向, x1 模式, TI2FP2 的边沿极性由 CC2P 设置。</p> <p>1100: 编码模式—带方向时钟, x2 模式。</p> <p>1101: 编码模式—带方向时钟, x1 模式, TI1FP1 和 TI2FP2 的边沿极性由 CC1P 和 CC2P 设置。</p>

## 14.9.4GTIM\_IER 中断使能寄存器

地址：请参见表 14-14GTIM 寄存器列表      复位值：0x0000 0000

位编号	位符号	权限	说明
31-15	RFU	-	保留位，请保持默认值
23	TERRIE	RW	编码器转换错误中断使能控制 0：禁止转换错误中断 1：使能转换错误中断
22	IERRIE	RW	编码器索引错误中断使能控制 0：禁止索引错误中断 1：使能索引错误中断
21	DIRIE	RW	编码器方向改变中断使能控制 0：禁止方向改变中断 1：使能方向改变中断
20	IDXIE	RW	编码器索引中断使能控制 0：禁止索引中断 1：使能索引中断
19-7	RFU	-	保留位，请保持默认值
6	TIE	RW	触发中断使能 0：禁止触发中断 1：使能触发中断
5	RFU	-	保留位，请保持默认值
4	CC4IE	RW	捕获/比较 4 中断使能 0：禁止 CC4 中断 1：使能 CC4 中断
3	CC3IE	RW	捕获/比较 3 中断使能 0：禁止 CC3 中断 1：使能 CC3 中断
2	CC2IE	RW	捕获/比较 2 中断使能 0：禁止 CC2 中断 1：使能 CC2 中断
1	CC1IE	RW	捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable) 0：禁止 CC1 中断 1：使能 CC1 中断
0	UIE	RW	更新中断使能 (Update interrupt enable) 0：禁止更新中断 1：使能更新中断

## 14.9.5 GTIM\_ISR 状态寄存器

地址：请参见表 14-14 GTIM 寄存器列表      复位值：0x0000 0000

位编号	位符号	权限	说明
31-24	RFU	-	保留位，请保持默认值
23	TERRF	RW	编码器转换错误中断标志 0：未发生转换错误 1：已发生转换错误
22	IERRF	RW	编码器索引错误中断标志 0：未发生索引错误 1：已发生索引错误
21	DIRF	RW	编码器方向改变中断标志 0：未发生编码方向改变 1：已发生编码方向改变
20	IDXF	RW	编码器索引中断标志 0：未发生索引标志 1：已发生索引标志
19-13	RFU	-	保留位，请保持默认值
12	CC4OF	RW	捕获/比较 4 重复捕获标志 请参见 CC1OF 说明
11	CC3OF	RW	捕获/比较 3 重复捕获标志 请参见 CC1OF 说明
10	CC2OF	RW	捕获/比较 2 重复捕获标志 请参见 CC1OF 说明
9	CC1OF	RW	捕获/比较 1 重复捕获标志 仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。 通过软件写入“0”可将该位清零。 0：未检测到重复捕获。 1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。
8-7	RFU	-	保留位，请保持默认值
6	TIF	RW	触发中断标志 (Trigger interrupt flag) 在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。 0：未发生触发事件。 1：触发中断挂起。
5	RFU	-	保留位，请保持默认值
4	CC4IF	RW	捕获/比较 4 中断标志 (Capture/Compare 4 interrupt flag) 请参见 CC1IF 说明
3	CC3IF	RW	捕获/比较 3 中断标志 (Capture/Compare 3 interrupt flag) 请参见 CC1IF 说明
2	CC2IF	RW	捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag) 请参见 CC1IF 说明

1	CC1IF	RW	<p>捕获/比较 1 中断标志 (Capture/Compare 1 interrupt flag)</p> <p>如果通道 CC1 配置为输出：当计数器与比较值匹配时，此标志由硬件置 1，中心对齐模式下除外（请参见 TIMx_CR1 寄存器中的 CMS 位说明）。但需要通过软件清零。</p> <p>0：不匹配。</p> <p>1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高。</p> <p>如果通道 CC1 配置为输入：此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将该位清零。</p> <p>0：未发生输入捕获事件</p> <p>1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）</p>
0	UIF	RW	<p>更新中断标志 (Update interrupt flag)</p> <p>该位在发生更新事件时通过硬件置 1。但需要通过软件清零。</p> <p>0：未发生更新。</p> <p>1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：</p> <ul style="list-style-type: none"> <li>- TIMx_CR1 寄存器中的 UDIS=0，并且重复计数器值上溢或下溢时（重复计数器 = 0 时更新）。</li> <li>- TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。</li> <li>- TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时</li> </ul>

### 14.9.6GTIM\_ICR 中断标志清除寄存器

地址：请参见表 14-14GTIM 寄存器列表      复位值：0x0000 0000

位编号	位符号	权限	说明
31-24	RFU	-	保留位，请保持默认值
13	TERRF	RW	<p>编码器转换错误中断标志清除</p> <p>0：清除编码器转换错误中断标志 1：无功能</p>
22	IERRF	RW	<p>编码器索引错误中断标志清除</p> <p>0：清除编码器索引错误中断标志 1：无功能</p>
21	DIRF	RW	<p>编码器方向改变中断标志清除</p> <p>0：清除编码器方向改变中断标志</p> <p>1：无功能</p>
20	IDXF	RW	<p>编码器索引中断标志清除</p> <p>0：清除编码器索引中断标志</p> <p>1：无功能</p>
19-13	RFU	-	保留位，请保持默认值
12	CC4OF	RW	捕获 / 比较 4 重复捕获标志清除，请参见 CC1OF 说明
11	CC3OF	RW	捕获 / 比较 3 重复捕获标志清除，请参见 CC1OF 说明
10	CC2OF	RW	捕获 / 比较 2 重复捕获标志清除，请参见 CC1OF 说明

9	CC1OF	RW	捕获 / 比较 1 重复捕获标志清除 0: 清除捕获 / 比较 1 的重复捕获标志 1: 无功能
8-7	RFU	-	保留位, 请保持默认值
6	TIF	RW	触发中断标志清除 0: 清除触发中断标志 1: 无功能
5	RFU	-	保留位, 请保持默认值
4	CC4IF	RW	捕获 / 比较 4 中断标志清除, 请参见 CC1IF 说明
3	CC3IF	RW	捕获 / 比较 3 中断标志清除, 请参见 CC1IF 说明
2	CC2IF	RW	捕获 / 比较 2 中断标志清除, 请参见 CC1IF 说明
1	CC1IF	RW	捕获 / 比较 1 中断标志清除 0: 清除捕获 / 比较 1 中断标志 1: 无功能
0	UIF	RW	更新中断标志清除 0: 清除更新中断标志 1: 无功能

### 14.9.7 GTIM\_EGR 事件生成寄存器

地址: 请参见表 14-14 GTIM 寄存器列表      复位值: 0x0000 0000

位域	名称	权限	功能描述
31-7	RFU	-	保留位, 请保持默认值
6	TG	RW	触发生成 此位由软件置 1 以生成事件, 并由硬件自动清零。 0: 不执行任何操作 1: GTIM_ISR 寄存器中的 TIF 标志置 1。使能后可发生相关中断事件。
5	RFU	-	保留位, 请保持默认值
4	CC4G	RW	捕获/比较 4 生成, 请参见 CC1G 说明
3	CC3G	RW	捕获/比较 3 生成, 请参见 CC1G 说明
2	CC2G	RW	捕获/比较 2 生成, 请参见 CC1G 说明
1	CC1G	RW	捕获/比较 1 生成 此位由软件置 1 以生成事件, 并由硬件自动清零。 0: 不执行任何操作 1: 通道 1 上生成捕获/比较事件, 根据 CC1 的配置分为以下情况: 如果通道 CC1 配置为输出: 使能时, CC1IF 标志置 1 并发送相应的中断请求。 如果通道 CC1 配置为输入: GTIM_CCR1 寄存器中将捕获到计数器当前值。使能时, CC1IF 标志置 1 并发送相应的中断请求。如果 CC1IF 标志已 为高, CC1OF 标志将置 1。 注: CC1E 必须使能, 否则 CCR1 寄存器会捕获计数器当前值, 但不产生中断标志, 无法进入中断。

0	UG	RW	<p>更新生成</p> <p>该位可通过软件置 1，并由硬件自动清零。</p> <p>0：不执行任何操作</p> <p>1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器的计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或递增计数模式，计数器将清零；如果选择递减计数模式，计数器将使用自动重载值 GTIM_ARR。</p>
---	----	----	--

### 14.9.8 GTIM\_CCMR1CAP 捕获模式寄存器 1

地址：请参见表 14-14 GTIM 寄存器列表      复位值：0x0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获和输出比较模式下的功能均不同。可同时将两条通道分别用于不同模式（例如，通道 1 用于输入捕获模式，通道 2 用于输出比较模式）。

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-12	IC2F	RW	<p>输入捕获 2 滤波器</p> <p>请参见 IC1F[3:0] 说明。</p>
11-10	IC2PSC	RW	<p>输入捕获 2 预分频器</p> <p>请参见 OC1PSC[1:0] 说明。</p>
9-8	CC2S	RW	<p>捕获/比较 2 选择</p> <p>此位域定义通道方向（输入/输出）以及所使用的输入。</p> <p>00：CC2 通道配置为输出</p> <p>01：CC2 通道配置为输入，IC2 映射到 TI2 上</p> <p>10：CC2 通道配置为输入，IC2 映射到 TI1 上</p> <p>11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效</p> <p>注：仅当通道关闭时（TIMx_CCER 中的 CC2E =“0”），才可向 CC2S 位写入数据。</p>
7-4	IC1F	RW	<p>输入捕获 1 滤波器</p> <p>此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器的采样长度。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：</p> <p>0000：无滤波器，按 f<sub>DTs</sub> 频率进行采样</p> <p>0001：f<sub>SAMPLING</sub>=f<sub>CK_INT</sub>，N=2</p> <p>0010：f<sub>SAMPLING</sub>=f<sub>CK_INT</sub>，N=4</p> <p>0011：f<sub>SAMPLING</sub>=f<sub>CK_INT</sub>，N=8</p> <p>0100：f<sub>SAMPLING</sub>=f<sub>CK_INT</sub>/2，N=6</p> <p>0101：f<sub>SAMPLING</sub>=f<sub>CK_INT</sub>/2，N=8</p> <p>0110：f<sub>SAMPLING</sub>=f<sub>CK_INT</sub>/4，N=6</p> <p>0111：f<sub>SAMPLING</sub>=f<sub>CK_INT</sub>/4，N=8</p> <p>1000：f<sub>SAMPLING</sub>=f<sub>CK_INT</sub>/8，N=6</p> <p>1001：f<sub>SAMPLING</sub>=f<sub>CK_INT</sub>/8，N=8</p> <p>1010：f<sub>SAMPLING</sub>=f<sub>CK_INT</sub>/16，N=5</p>



			1011: $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}/16$ , N=6 1100: $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}/16$ , N=8 1101: $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}/32$ , N=5 1110: $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}/32$ , N=6 1111: $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}/32$ , N=8
3-2	IC1PSC	RW	输入捕获 1 预分频器 此位域定义 CC1 输入 (IC1) 的预分频比。只要 CC1E="0" (TIMx_CCER 寄存器), 预分频器便立即复位。 00: 无预分频器, 捕获输入上每检测到一个边沿便执行捕获 01: 每发生 2 个事件便执行一次捕获 10: 每发生 4 个事件便执行一次捕获 11: 每发生 8 个事件便执行一次捕获
1-0	CC1S	RW	捕获/比较 1 选择 此位域定义通道方向 (输入/输出) 以及所使用的输入。 00: CC1 通道配置为输出 01: CC1 通道配置为输入, IC1 映射到 TI1 上 10: CC1 通道配置为输入, IC1 映射到 TI2 上 11: CC1 通道配置为输入, IC1 映射到 TRC 上。此模式仅在通过 TS 位 (TIMx_SMCR 寄存器) 选择内部触发输入时有效 注: 仅当通道关闭时 (TIMx_CCER 中的 CC1E = "0"), 才可向 CC1S 位写入数据。

### 14.9.9 GTIM\_CCMR1CMP 比较模式寄存器 1

地址: 请参见表 14-14 GTIM 寄存器列表 复位值: 0x0000 0000

同一寄存器可用于输出比较模式 (本节) 或输入捕获模式 (上一节)。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获和输出比较模式下的功能均不同。可同时将两条通道分别用于不同模式 (例如, 通道 1 用于输入捕获模式, 通道 2 用于输出比较模式)。

输出比较模式:

位编号	位符号	权限	说明
31-25	RFU	-	保留位, 请保持默认值
24	OC2MH	RW	输出比较 2 模式——位 3 (Output Compare 2 mode - bit 3) 请参见 OC2M 说明——位 14:12
23-17	RFU	-	保留位, 请保持默认值
16	OC1MH	RW	输出比较 1 模式——位 3 (Output Compare 1 mode - bit 3) 请参见 OC1M 说明——位 6:4
15	OC2CE	RW	输出比较 2 清零使能 (Output Compare 2 clear enable) 请参见 OC1CE 说明。
14-12	OC2M	RW	输出比较 2 模式 (Output Compare 2 mode) 请参见 OC1M[2:0] 说明。
11	OC2PE	RW	输出比较 2 预装载使能 (Output Compare 2 preload enable) 请参见 OC1PE 说明。



10	OC2FE	RW	输出比较 2 快速使能 (Output Compare 2 fast enable) 请参见 OC1FE 说明。
9-8	CC2S	RW	捕获/比较 2 选择 (Capture/Compare 2 selection) 此位域定义通道方向 (输入/输出) 以及所使用的输入。 00: CC2 通道配置为输出 01: CC2 通道配置为输入, IC2 映射到 TI2 上 10: CC2 通道配置为输入, IC2 映射到 TI1 上 11: CC2 通道配置为输入, IC2 映射到 TRC 上。此模式仅在通过 TS 位 (TIMx_SMCR 寄存器) 选择内部触发输入时有效 注: 仅当通道关闭时 (TIMx_CCER 中的 CC2E =“0”), 才可向 CC2S 位写入数据。
7	OC1CE	RW	输出比较 1 清零使能 0: OC1Ref 不受 ocref_clr_int 信号影响 1: ocref_clr_int 信号 (OCREF_CLR 输入或 ETRF 输入) 上检测到高电平时, OC1Ref 立即清零
6-4	OC1M	RW	输出比较 1 模式 (Output Compare 1 mode) 这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效, 而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。 0000: 冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。(该模式用于生成时基)。 0001: 将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1(TIMx_CCR1) 匹配时, OC1REF 信号强制变为高电平。 0010: 将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1(TIMx_CCR1) 匹配时, OC1REF 信号强制变为低电平。 0011: 翻转——TIMx_CNT=TIMx_CCR1 时, OC1REF 发生翻转。 0100: 强制变为无效电平——OC1REF 强制变为低电平。 0101: 强制变为有效电平——OC1REF 强制变为高电平。 0110 : PWM 模式 1——在递增计数模式下, 只要 TIMx_CNT<TIMx_CCR1, 通道 1 便为有效状态, 否则为无效状态。在递减计数模式下, 只要 TIMx_CNT>TIMx_CCR1, 通道 1 便为无效状态 (OC1REF=“0”), 否则为有效状态 (OC1REF=“1”)。 0111 : PWM 模式 2——在递增计数模式下, 只要 TIMx_CNT<TIMx_CCR1, 通道 1 便为无效状态, 否则为有效状态。在递减计数模式下, 只要 TIMx_CNT>TIMx_CCR1, 通道 1 便为有效状态, 否则为无效状态。 1000: 可再触发 OPM 模式 1——在递增计数模式下, 通道为有效状态, 直至 (在 TRGI 信号上) 检测到触发事件。然后, 在 PWM 模式 1 下进行比较, 通道会在下一次更新时再次变为有效状态。在递减计数模式下, 通道为无效状态, 直至 (在 TRGI 信号上) 检测到触发事件。然后, 在 PWM 模式 1 下进行比较, 通道会在下一次更新时再次变为无效状态。

			<p>1001: 可再触发 OPM 模式 2——在递增计数模式下, 通道为无效状态, 直至 (在 TRGI 信号上) 检测到触发事件。然后, 在 PWM 模式 2 下进行比较, 通道会在下一次更新时再次变为无效状态。在递减计数模式下, 通道为有效状态, 直至 (在 TRGI 信号上) 检测到触发事件。然后, 在 PWM 模式 1 下进行比较, 通道会在下一次更新时再次变为有效状态。</p> <p>1010: 保留。</p> <p>1011: 保留。</p> <p>1100: 组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。</p> <p>1101: 组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。</p> <p>1110: 不对称 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时, OC1REFC 输出 OC1REF; 计数器递减计数时, OC1REFC 输出 OC2REF。</p> <p>1111: 不对称 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时, OC1REFC 输出 OC1REF; 计数器递减计数时, OC1REFC 输出 OC2REF。</p> <p>注: 只要编程了 LOCK (TIMx_BDTR 寄存器中的 LOCK 位) 级别 3 且 CC1S="00" (通道配置为输出), 这些位即无法修改。</p> <p>注: 在 PWM 模式下, 仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时, OCREF 电平才会发生更改。</p> <p>注: OCxREFC 输出优先级依次是:</p> <p>组合 PWM 模式&gt;CHx/CH5 组合波形配置&gt;CHx/CH6 组合波形配置</p> <p>注: 此位域将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1, 则仅当生成 COM 事件时, OC1M 有效位才会从预装载位获取新值。</p>
3	OC1PE	RW	<p>输出比较 1 预装载使能 (Output Compare 1 preload enable)</p> <p>0: 禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据, 写入后将立即使用新值。</p> <p>1: 使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到当前寄存器中。</p> <p>注: 1: 只要编程了 LOCK (TIMx_BDTR 寄存器中的 LOCK 位) 级别 3 且 CC1S="00" (通道配置为输出), 这些位即无法修改。</p> <p>2: 只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式 (TIMx_CR1 寄存器中的 OPM 位置 1)。其他情况下则无法保证该行为。</p>
2	OC1FE	RW	<p>输出比较 1 快速使能 (Output Compare 1 fast enable)</p> <p>此位用于加快触发输入事件对 CC 输出的影响。</p> <p>0: 即使触发开启, CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时, 激活 CC1 输出的最短延迟时间为 5 个时钟周期。</p> <p>1: 触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后,</p>

			无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。
1-0	CC1S	RW	捕获/比较 1 选择 (Capture/Compare 1 selection) 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC1 通道配置为输出 01: CC1 通道配置为输入，IC1 映射到 TI1 上 10: CC1 通道配置为输入，IC1 映射到 TI2 上 11: CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注：仅当通道关闭时（TIMx_CCER 中的 CC1E =“0”），才可向 CC1S 位写入数据。

### 14.9.10 GTIM\_CCMR2CAP 捕获模式寄存器 2

地址：请参见表 14-14GTIM 寄存器列表      复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获和输出比较模式下的功能均不同。可同时将两条通道分别用于不同模式（例如，通道 1 用于输入捕获模式，通道 2 用于输出比较模式）。

输入捕获模式：

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-12	IC4F	RW	输入捕获 4 滤波器 (Input capture 4 filter) 请参见 IC1F[3:0] 说明。
11-10	IC4PSC	RW	输入捕获 4 预分频器 (Input capture 4 prescaler) 请参见 IC1PSC[1:0] 说明。
9-8	CC4S	RW	捕获/比较 4 选择 (Capture/Compare 4 selection) 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC4 通道配置为输出 01: CC4 通道配置为输入，IC4 映射到 TI4 上 10: CC4 通道配置为输入，IC4 映射到 TI3 上 11: CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效 注：仅当通道关闭时（TIMx_CCER 中的 CC4E =“0”），才可向 CC4S 位写入数据。
7-4	IC3F	RW	输入捕获 3 滤波器 (Input capture 3 filter) 请参见 IC1F[3:0] 说明。
3-2	IC3PSC	RW	输入捕获 3 预分频器 (Input capture 3 prescaler) 请参见 IC1PSC[1:0] 说明。
1-0	CC3S	RW	捕获/比较 3 选择 (Capture/compare 3 selection) 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC3 通道配置为输出 01: CC3 通道配置为输入，IC3 映射到 TI3 上 10: CC3 通道配置为输入，IC3 映射到 TI4 上

			<p>11: CC3 通道配置为输入, IC3 映射到 TRC 上。此模式仅在通过 TS 位 (TIMx_SMCR 寄存器) 选择内部触发输入时有效</p> <p>注: 仅当通道关闭时 (TIMx_CCER 中的 CC3E =“0”), 才可向 CC3S 位写入数据。</p>
--	--	--	--

## 14.9.11 GTIM\_CCMR2CMP 比较模式寄存器 2

地址: 请参见表 14-14GTIM 寄存器列表 复位值: 0x0000 0000

同一寄存器可用于输入捕获模式 (上一节) 或输出比较模式 (本节)。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获和输出比较模式下的功能均不同。可同时将两条通道分别用于不同模式 (例如, 通道 1 用于输入捕获模式, 通道 2 用于输出比较模式)。

输入比较模式:

位编号	位符号	权限	说明
31-25	RFU	-	保留位, 请保持默认值
24	OC4MH	RW	输出比较 3 模式——位 3 (Output Compare 3 mode - bit 3) 请参见 OC1M 说明。
23-17	RFU	-	保留位, 请保持默认值
16	OC3MH	RW	输出比较 3 模式——位 3 (Output Compare 3 mode - bit 3) 请参见 OC1M 说明。
15	OC4CE	RW	输出比较 4 清零使能 (Output compare 4 clear enable) 请参见 OC1CE 说明。
14-12	OC4M	RW	输出比较 4 模式 (Output compare 4 mode) 请参见 OC4M 说明。
11	OC4PE	RW	输出比较 4 预装载使能 (Output compare 4 preload enable) 请参见 OC1PE 说明。
10	OC4FE	RW	输出比较 4 快速使能 (Output compare 4 fast enable) 请参见 OC1FE 说明。
9-8	CC4S	RW	捕获/比较 4 选择 (Capture/Compare 4 selection) 此位域定义通道方向 (输入/输出) 以及所使用的输入。 00: CC4 通道配置为输出 01: CC4 通道配置为输入, IC4 映射到 TI4 上 10: CC4 通道配置为输入, IC4 映射到 TI3 上 11: CC4 通道配置为输入, IC4 映射到 TRC 上。此模式仅在通过 TS 位 (TIMx_SMCR 寄存器) 选择内部触发输入时有效 注: 仅当通道关闭时 (TIMx_CCER 中的 CC4E =“0”), 才可向 CC4S 位写入数据。
7	OC3CE	RW	输出比较 3 清零使能 (Output compare 3 clear enable) 请参见 OC1CE 说明。
6-4	OC3M	RW	输出比较 3 模式 (Output compare 3 mode) 请参见 OC1M 说明。
3	OC3PE	RW	输出比较 3 预装载使能 (Output compare 3 preload enable) 请参见 OC1PE 说明。
2	OC3FE	RW	输出比较 3 快速使能 (Output compare 3 fast enable) 请参见 OC1FE 说明。

1-0	CC3S	RW	<p>捕获/比较 3 选择 (Capture/Compare 3 selection) 此位域定义通道方向（输入/输出）以及所使用的输入。 00: CC3 通道配置为输出 01: CC3 通道配置为输入, IC3 映射到 TI3 上 10: CC3 通道配置为输入, IC3 映射到 TI4 上 11: CC3 通道配置为输入, IC3 映射到 TRC 上。此模式仅在通过 TS 位 (TIMx_SMCR 寄存器) 选择内部触发输入时有效 注: 仅当通道关闭时 (TIMx_CCER 中的 CC3E =“0”), 才可向 CC3S 位写入数据。</p>
-----	------	----	--

### 14.9.12 GTIM\_CCER 捕获/比较使能寄存器

地址: 请参见表 14-14GTIM 寄存器列表      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位, 请保持默认值
15	CC4NP	RW	捕获/比较 4 互补输出极性 请参见 CC1NP 说明
14	RFU	-	保留位, 请保持默认值
13	CC4P	RW	捕获/比较 4 输出极性 请参见 CC1P 说明
12	CC4E	RW	捕获/比较 4 输出使能 请参见 CC1E 说明
11	CC3NP	RW	捕获/比较 3 互补输出极性 请参见 CC1NP 说明
10	RFU	-	保留位, 请保持默认值
9	CC3P	RW	捕获/比较 3 输出极性 请参见 CC1P 说明
8	CC3E	RW	捕获/比较 3 输出使能 请参见 CC1E 说明
7	CC2NP	RW	捕获/比较 2 互补输出极性 请参见 CC1NP 说明
6	RFU	-	保留位, 请保持默认值
5	CC2P	RW	捕获/比较 2 输出极性 请参见 CC1P 说明
4	CC2E	RW	捕获/比较 2 输出使能 请参见 CC1E 说明
3	CC1NP	RW	<p>捕获/比较 1 互补输出极性 CC1 通道配置为输出: 0: OC1N 高电平有效。 1: OC1N 低电平有效。 CC1 通道配置为输入: 此位与 CC1P 配合使用, 用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1P 说明。</p>



			<p>注 1: 只要编程了 LOCK (TIMx_BDTR 寄存器中的 LOCK 位) 级别 2 或 3 且 CC1S="00" (通道配置为输出), 此位立即变为不可写状态。</p> <p>注 2: 此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1, 则仅当生成换向事件时, CC1NP 有效位才会从预装载位获取新值。</p>
2	RFU	-	保留位, 请保持默认值
1	CC1P	RW	<p>捕获/比较 1 输出极性</p> <p>CC1 通道配置为输出:</p> <p>0: OC1 高电平有效</p> <p>1: OC1 低电平有效</p> <p>CC1 通道配置为输入: CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。</p> <p>00: 非反相/上升沿触发。电路作用于 TIxFP1 的上升沿 (在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), TIxFP1 未反相 (在门控模式或编码器模式下执行触发操作)。</p> <p>01: 反相/下降沿触发。电路作用于 TIxFP1 的下降沿 (在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), TIxFP1 反相 (在门控模式或编码器模式下执行触发操作)。</p> <p>10: 保留, 不使用此配置。</p> <p>11: 非反相/上升沿和下降沿均触发。电路作用于 TIxFP1 的上升沿和下降沿 (在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), TIxFP1 未反相 (在门控模式下执行触发操作)。编码器模式下不得使用此配置。</p> <p>注 1: 只要编程了 LOCK (TIMx_BDTR 寄存器中的 LOCK 位) 级别 2 或 3, 此位立即变为不可写状态。</p> <p>注 2: 此位将在具有互补输出的通道上进行预装载。如果 GTIM_CR2 寄存器中的 CCPC 位置 1, 则仅当生成换向事件时, CC1P 有效位才会从预装载位获取新值。</p>
0	CC1E	RW	<p>捕获 / 比较 1 输出使能</p> <p>0: 禁止捕获模式 / OC1 未激活</p> <p>1: 使能捕获模式 / 在相应输出引脚上输出 OC1 信号</p> <p>注: 使用输入捕获功能时, 该位必须配置为 1。</p>

### 14.9.13 GTIM\_CNT 计数器

地址: 请参见表 14-14GTIM 寄存器列表      复位值: 0x0000

位编号	位符号	权限	说明
31	UIFCPY	RW	<p>根据 GTIM_CR1 寄存器中 UIFREMAP 位域的值, 本位域表示不同的含义:</p> <p>UIFREMAP = 0, 本位域保留, 读为 0</p> <p>UIFREMAP = 1, 本位域表示 GTIM_ISR 寄存器的 UIF 位的只读副本</p>
30-16	RFU	-	保留位, 请保持默认值
15-0	CNT	RW	计数器值

### 14.9.14 GTIM\_PSC 预分频器

地址：请参见表 14-14GTIM 寄存器列表      复位值：0x0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	PSC	RW	<p>预分频值 (Prescaler value)</p> <p>计数器时钟频率 (CK_CNT) 等于 <math>f_{CK\_PSC} / (PSC[15:0] + 1)</math>。</p> <p>PSC 包含每次发生更新事件时（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）要装载到活动预分频器寄存器的值。</p>

### 14.9.15 GTIM\_ARR 自动重载寄存器

地址：请参见表 14-14GTIM 寄存器列表      复位值：复位值：0x0000 FFFF

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	ARR	RW	<p>自动重载值 (Auto-reload value)</p> <p>ARR 为要装载到实际自动重载寄存器的值。</p> <p>有关 ARR 更新和行为的更多详细信息，请参见第 457 页的第 20.3.1 节：时基单元。当自动重载值为空时，计数器不工作。</p>

### 14.9.16 GTIM\_CCR1 捕获/比较寄存器 1

地址：请参见表 14-14GTIM 寄存器列表      复位值：0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	CCR1	RW	<p>捕获/比较 1 值 (Capture/Compare 1 value)</p> <p>如果通道 CC1 配置为输出：CCR1 为要装载到有效捕获/比较 1 寄存器的值（预装载值）。</p> <p>如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；</p> <p>否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 1）。</p> <p>实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。</p> <p>如果通道 CC1 配置为输入：CR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。只能读取 TIMx_CCR1 寄存器，无法对其进行编程。</p>

### 14.9.17 GTIM\_CCR2 捕获/比较寄存器 2

地址：请参见表 14-14GTIM 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	CCR2	RW	<p>捕获/比较 2 值</p> <p>如果通道 CC2 配置为输出：CCR2 为要装载到有效捕获/比较 2 寄存器的值（预装载值）。</p> <p>如果没有通过 TIMx_CCMR1 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；</p> <p>否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 2）。</p> <p>有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。</p> <p>如果通道 CC2 配置为输入：CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。只能</p> <p>读取 TIMx_CCR2 寄存器，无法对其进行编程。</p>

### 14.9.18 GTIM\_CCR3 捕获/比较寄存器 3

地址：请参见表 14-14GTIM 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	CCR3	RW	<p>捕获/比较值 (Capture/Compare value)</p> <p>如果通道 CC3 配置为输出：CCR3 为要装载到有效捕获/比较 3 寄存器的值（预装载值）。</p> <p>如果没有通过 TIMx_CCMR2 寄存器中的 OC3PE 位来使能预装载功能，则该值立刻生效；</p> <p>否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 3）。</p> <p>有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC3 输出上发出信号的值。</p> <p>如果通道 CC3 配置为输入：CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。只能读取 TIMx_CCR3 寄存器，无法对其进行编程。</p>

### 14.9.19 GTIM\_CCR4 捕获/比较寄存器 4

地址：请参见表 14-14GTIM 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	CCR4	RW	<p>捕获/比较值 (Capture/Compare value)</p> <p>如果通道 CC4 配置为输出：CCR4 为要装载到有效捕获/比较 4 寄存</p>



			<p>器的值（预装载值）。</p> <p>如果没有通过 TIMx_CCMR2 寄存器中的 OC4PE 位来使能预装载功能，则该值立刻生效；</p> <p>否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 4）。</p> <p>有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC4 输出上发出信号的值。</p> <p>如果通道 CC4 配置为输入：CCR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值。只能读取 TIMx_CCR4 寄存器，无法对其进行编程。</p>
--	--	--	---

## 14.9.20 GTIM\_ECR 编码控制寄存器

地址：请参见表 14-14GTIM 寄存器列表      复位值：0x0000 0000

31-8	RFU	-	保留位，请保持默认值
7-6	IPOS	RW	<p>编码索引定位</p> <p>在正交编码器模式（SMS[3: 0] = 0001、0010、0011、1110、1111）中，此位表示索引事件在哪个 AB 输入配置中重置计数器。</p> <p>00：当 AB = 00 时，索引会重置计数器</p> <p>01：当 AB = 01 时，索引会重置计数器</p> <p>10：当 AB = 10 时，索引会重置计数器</p> <p>11：当 AB = 11 时，索引会重置计数器</p> <p>在定向时钟编码器模式或时钟加方向编码器模式（SMS[3: 0] = 1010、1011、1100、1101）中，这些位指示索引事件在哪个电平上重置计数器。在定向时钟编码器模式下，这一点都适用于两个时钟输入。</p> <p>x0：当时钟信号为低电平时，索引将重置计数器</p> <p>x1：当时钟信号为高电平时，索引将重置计数器</p>
5	FIDX	RW	<p>该位表示是否只考虑第一个索引</p> <p>0：索引始终处于活动状态</p> <p>1：仅第一个索引重置计数器</p>
4-3	RFU	-	保留位，请保持默认值
2-1	IDIR	RW	<p>索引方向</p> <p>此位域表示索引在何种计数方向上重置计数器</p> <p>00：在任何计数方向上重置计数器</p> <p>01：仅在向上计数时重置计数器</p> <p>10：仅在向下计数时重置计数器</p> <p>11：保留</p>
0	IE	RW	<p>索引使能</p> <p>0：禁止索引</p> <p>1：使能索引</p>

### 14.9.21 GTIM\_TISEL1 输入选择寄存器 1

地址：请参见表 14-14GTIM 寄存器列表      复位值：0x0000 0000

位编号	位符号	权限	说明
31-28	RFU	-	保留位，请保持默认值
27-24	TI3SEL	RW	TI4 通道输入捕获信号来源选择，参见 TI1SEL 说明
23-20	RFU	-	保留位，请保持默认值
19-16	TI3SEL	RW	TI3 通道输入捕获信号来源选择，参见 TI1SEL 说明
15-12	RFU	-	保留位，请保持默认值
11-8	TI2SEL	RW	TI2 通道输入捕获信号来源选择，参见 TI1SEL 说明
7-4	RFU	-	保留位，请保持默认值
3-0	TI1SEL	RW	TI1 通道输入捕获信号来源选择 0000: GTIM_CH1 0001: VC0_OUT 0010: VC1_OUT 0011: UART0_RXD 0100: UART1_RXD 0101: MCO_OUT0 0110: 保留位 0111: 保留位 1000: 保留位 1001: LSI_OUT 1010: BTIM0_Trigo 1011: BTIM1_Trigo 1100: BTIM2_Trigo 1101: GTIM_Trigo

### 14.9.22 GTIM\_AF1 复用功能选择寄存器 1

地址：请参见表 14-14GTIM 寄存器列表      复位值：0x0000 0000

位编号	位符号	权限	说明
31-18	RFU	-	保留位，请保持默认值
17-14	ETRSRL	RW	ETR 信号来源选择 0000: GTIM_ETR 引脚 0001: VC0_OUT 0010: VC1_OUT 0101: ADC_AWD 0111: 保留位 1000: 保留位 1001: GTIM_ETR 引脚 1101: UART0_TXD 1110: UART1_TXD

			1111: 保留位 注: 只要编程了 LOCK (GTIM_BDTR 寄存器中的 LOCK 位) 级别 1, 这些位即无法修改
13-0	RFU	-	保留位, 请保持默认值

### 14.9.23 GTIM\_AF2 复用功能选择寄存器 2

地址: 请参见表 14-14GTIM 寄存器列表      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-19	RFU	-	保留位, 请保持默认值
18-16	ETRSEL	RW	OCREF_CLR 源选择 000: VC0_OUT 001: VC1_OUT 100: ADC_AWD 111: ETRF
15-0	RFU	-	保留位, 请保持默认值

# 15 基本定时器（BTIM）

## 15.1 概述

HC32F0653 内部集成 3 个基本定时器 (BTIM)，每个 BTIM 完全独立且功能完全相同，各包含一个 16bit 自动重载计数器并由一个可编程预分频器驱动。BTIM 支持内部计数模式、外部计数模式、触发启动模式和门控计数模式 4 种工作模式，支持更新事件和触发事件发生时产生中断。不同工作模式下均可由复位输入信号控制计数器复位。

## 15.2 主要特性

- 16bit 自动重载递增计数器
- 可编程预分频器支持 1、2、3、4、...、65536 分频
- 支持单次计数模式和连续计数模式
- 触发输入信号（TRGI）控制定时器实现多种从模式
- 用于外设间同步的触发输出信号（TRGO）
- 复位信号（RSTI）控制计数器复位
- 更新事件和触发事件发生时产生中断

## 15.3 功能描述

### 15.3.1 功能框图

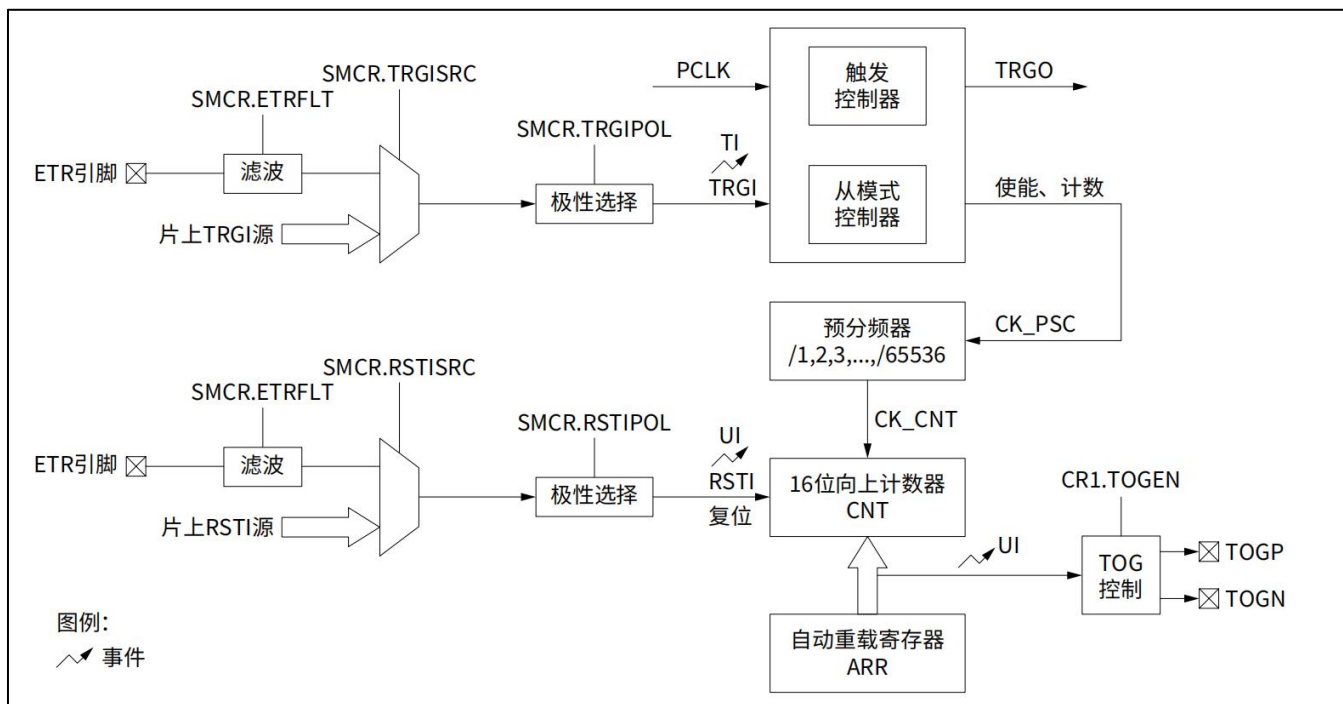


图 15-1BTIM 功能框图

### 15.3.1.1 计数单元

计数单元的核心组件是一个 16bit 递增计数器 CNT 和一个 16bit 自动重载寄存器 ARR，计数器的时钟可通过预分频器进行分频。计数寄存器、自动重载寄存器和预分频寄存器可通过软件进行读写，即使在计数器运行时也可执行读写操作。

预分频器对 CK\_PSC 时钟进行分频，得到计数时钟 CK\_CNT，以驱动计数器计数。分频系数通过 BTIMx\_PSC 寄存器进行设置，支持 1、2、3、4、...、65536 分频。

计数器可工作在单次计数或连续计数模式下，通过控制寄存器 BTIMx\_CR0 的 ONESHOT 位域来选择。当设置 BTIMx\_CR0 寄存器的 EN 位域为 1 时，计数器开始递增计数，注意实际的计数器使能信号 CNT\_EN 在 EN 置 1 的一个时钟周期后被置 1。

#### 单次计数模式

设置 BTIMx\_CR0.ONESHOT 为 1，使定时器工作在单次计数模式下。设置 BTIMx\_CR0.EN 为 1 使能 BTIMx，计数器 CNT 在计数时钟 CK\_CNT 的驱动下累加计数。当计数值到达重载值 ARR 后产生溢出信号 OV 和更新事件 UEV（OV 信号和 UEV 信号会自动清除），计数器更新中断标志 BTIMx\_ISR.UIF 被硬件置位，同时计数器停止计数，BTIMx\_CR0.EN 被硬件自动清零。

下图是单次计数模式示例，其中 BTIMx\_PSC=0x01、BTIMx\_ARR=0xFC。

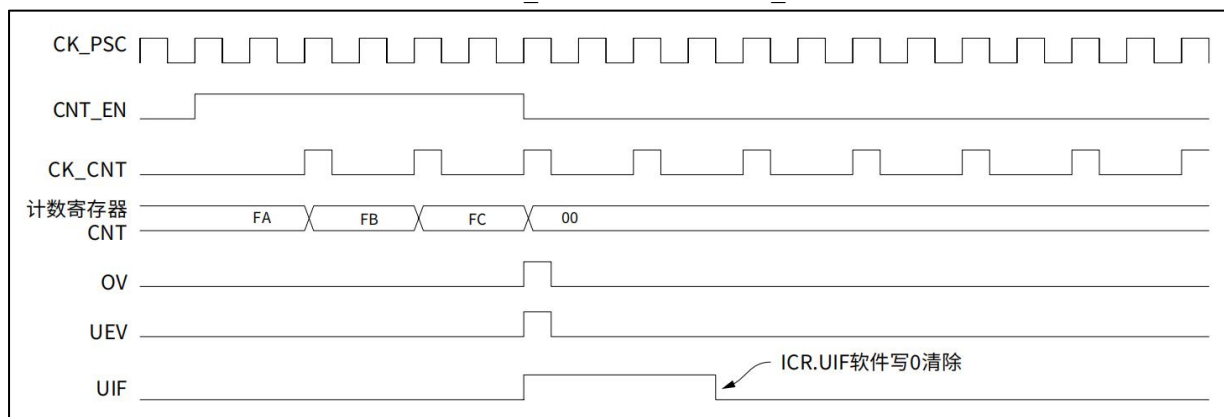


图 15-2 单次计数模式示例

#### 连续计数模式

设置 BTIMx\_CR0.ONESHOT 为 0，使定时器工作在连续计数模式下。设置 BTIMx\_CR0.EN 为 1 使能 BTIMx，计数器 CNT 在计数时钟 CK\_CNT 的驱动下累加计数。当计数值到达重载值 ARR 后，将重新从 0 开始递增计数，同时产生溢出信号 OV 和更新事件 UEV（OV 信号和 UEV 信号会自动清除），计数器更新中断标志位 BTIMx\_ISR.UIF 被硬件置位。

下图是连续计数模式示例，其中 BTIMx\_PSC=0x00、BTIMx\_ARR=0xFC。

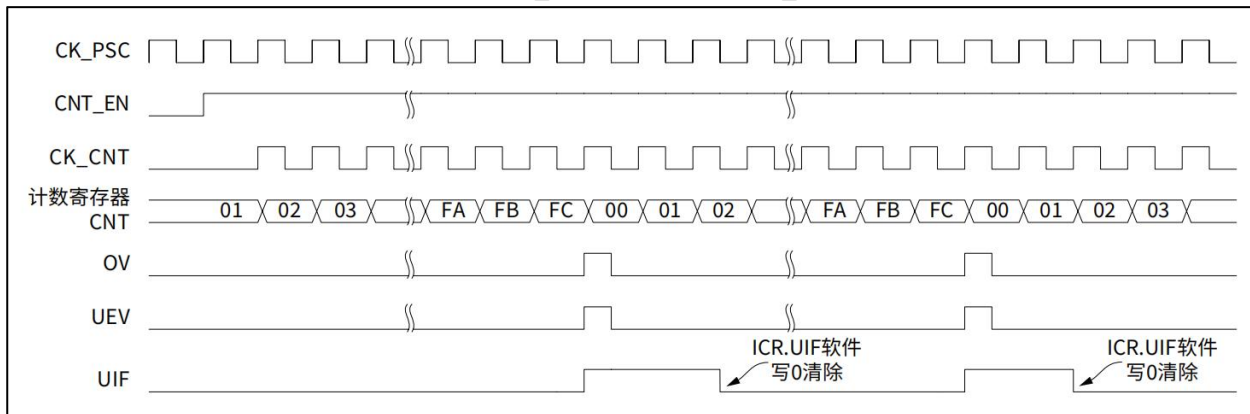


图 15-3 连续计数模式示例

### 15.3.1.2 更新事件UEV

允许通过控制寄存器 BTIMx\_CR0 的 UDIS 位域来禁止或使能更新。

#### 使能 UEV

设置 UDIS 位域为 0 使能 UEV，此时根据 URS 位域可选择更新请求源，如下表所示：

表 15-1 更新源设置

BTIMx_CR0.URS	更新源
0	计数器上溢出； UG 置位； 通过从模式控制器生成的更新事件
1	计数器上溢出

当发生更新事件时，将进行以下动作：

1. 重新初始化计数器，即计数器清零；
2. 预分频器的计数器被清零，但预分频比不受影响；
3. 事件更新中断标志位 BTIMx\_ISR.UIF 被硬件置位。

当发生一个更新事件 UEV 时，事件更新中断标志位 BTIMX\_ISR.UIF 会被硬件置位，如果允许中断(设置 BTIMxIER.UIE 为 1)，将产生一个更新中断请求，设置 BTIMX\_ICR.UIF 为 0 可清除该标志位。

#### 禁止 UEV

设置 UDIS 位域为 1 禁止 UEV，不再生成任何更新事件。

如果 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器的计数器。

### 15.3.1.3 触发输入通道

触发输入信号 (TRGI) 可用作计数器的计数时钟，也可作为计数器的触发启动信号和门控信号。

可对来自于 BTIMx\_ETR 引脚的触发输入信号进行滤波控制，并设置 TRGI 信号的有效极性。

#### 滤波器

滤波器采用数字滤波方式，以一定频率对 ETR 输入信号进行采样，当连续采样到 N 个相同电平时信号有效，否则信号无效，以此滤除高频杂波信号。

滤波单元的采样时钟为 PCLK 或 PCLK 的分频，通过从模式控制寄存器 BTIMx\_SMCR 的 ETRFLT 位域可以选择 PCLK 的分频比及采样点个数 N。

#### 极性选择

从模式控制寄存器 BTIMx\_SMCR 的 TRGIPOL 位域用于选择触发输入信号的有效极性。当设置 TRGIPOL 为 0 时，TRGI 信号不反相，高电平或上升沿有效；当 TRGIPOL 设置为 1 时，TRGI 信号反相，低电平或下降沿有效。

### 15.3.1.4 触发输出通道

通过控制寄存器 BTIMx\_CR0 的 MMS 位域，可以选择 BTIM 主模式下将要发送到其它外设以实现同步的触发输出信号 (TRGO)，如下表所示：

表 15-2 BTIM 触发输出 (TRGO)

BTIMx_CR0.MMS	TRGO 信号
0	复位信号
1	计数器使能信号
10	更新事件
11	溢出信号

### 15.3.1.5 复位输入通道

BTIM 支持在不同工作模式下通过复位输入信号(RSTI)的有效边沿复位计数器，同时生成更新事件。

RSTL 信号的具体来源请参考从模式控制寄存器 BTIMX SMCR 的 RSTISRC 位域说明。可通过从模式控制寄存器 BTIMX SMCR 的 ETRFLT 位域对 BTIMX ETR 引脚的复位信号进行滤波控制;通过 BTIMx SMCR 寄存器的 RSTIPOL 位域设置 RSTI 信号的有效极性，当设置 RSTIPOL 为 0 时，RSTI 信号不反相，上升沿有效;当 RSTIPOL 设置为 1 时，RSTI 信号反相，下降沿有效。

### 15.3.1.6 翻转输出单元

翻转输出单元可通过 ARR 溢出信号 OV 控制外部 BTIMx TOGP 和 BTIMx TOGN 引脚输出翻转信号。

当设置 BTIMX\_CR0.TOGEN 为 0 时，BTIMX\_TOGP 和 BTIMX\_TOGN 引脚均输出低电平。

当设置 BTIMX\_CR0.TOGEN 为 1 时，BTIMX\_TOGP 和 BTIMX\_TOGN 引脚输出电平相反的信号(BTIMx TOGP 默认电平为高电平);当计数器 ARR 溢出时，BTIMx TOGP 和 BTIMX\_TOGN 引脚输出电平将翻转。

下图所示为连续计数模式下，BTIMX TOGP 和 BTIMX TOGN 引脚电平翻转输出示意图:

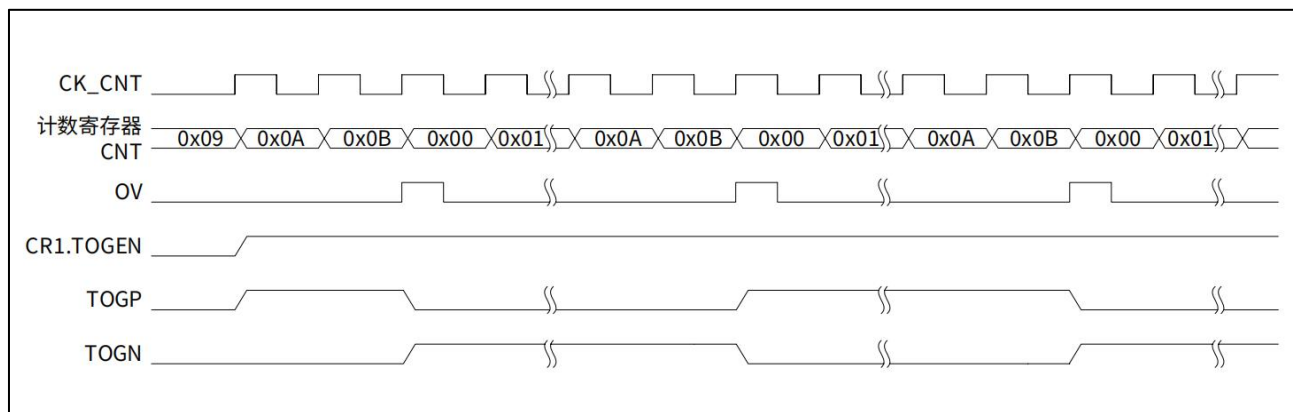


图 15-4 电平翻转输出示意图(ARR=0x0B)

## 15.3.2 工作模式

BTIM 支持 4 种工作模式：内部计数模式、外部计数模式、触发启动模式和门控计数模式。通过从模式控制寄存器 BTIMx\_SMCR 的 SMS 位域来配置，如下表所示：

表 15-3 BTIM 工作模式

BTIMx_SMCR.SMS	工作模式	描述
0	禁止从模式，使用内部时钟	时钟源为 PCLK
1	触发启动模式	时钟源为 PCLK，TRGI 信号触发计数器启动
10	门控计数模式	时钟源为 PCLK，TRGI 信号作为门控信号
11	外部计数模式	时钟源为 TRGI 信号



### 15.3.2.1 内部计数模式

当从模式控制寄存器 BTIMx SMCR 的 SMS 位域为 0x0 时, 禁止定时器从模式, 预分频器时钟直接由内部时钟 PCLK 提供。设置 BTIMXCR0.EN 为 1, 将使能计数器开始计数。内部计数模式框图如下图所示:

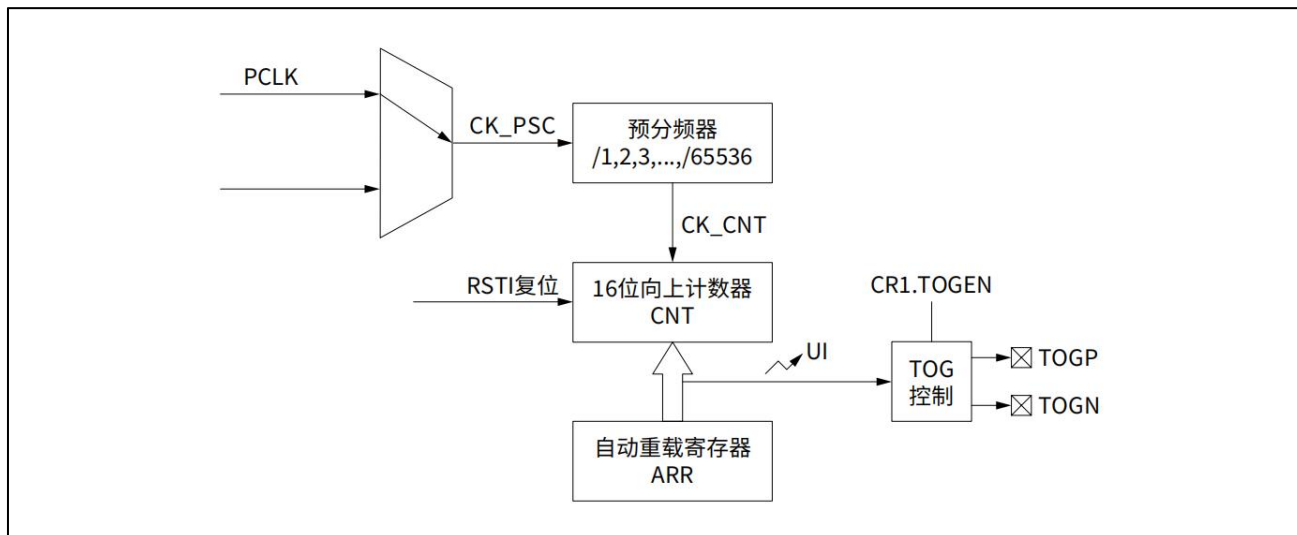


图 15-5 内部计数模式框图

在实际应用中, 系统时钟 PCLK 的频率是已知的, 通过合理设置预分频器系数 PSC 可以对固定时长的时钟进行计数, 配合对重载值 ARR 的设置及计数器事件更新中断标志位的使用, 可以精确获得某一特定时长, 从而达到定时的目地。定时时间工计算公式:

$$T = ((PSC + 1) / PCLK) \times (ARR + 1)$$

其中, PCLK 为计数器时钟源, PSC 为预分频系数, ARR 为重载值。例:

当计数器时钟源 PCLK 的频率为 24MHz 时, 要求定时 100ms。

如果设置预分频系数 PSC 为 0xFF, 计算:

$$T = 100ms = ((255 + 1) / 24MHz) \times (ARR + 1)$$

则

$$ARR = 9374 (0x249E)$$

即需要设置重载值 ARR 为 0x249E。

### 15.3.2.2 外部计数模式

当从模式控制寄存器 BTIMx SMCR 的 SMS 位域为 0x3 时, 由所选触发信号 (TRGI) 的上升沿提供计数器时钟。

TRGI 信号有多种来源, 具体通过 BTIMx SMCR 寄存器的 TRGISRC 位域进行选择, 并可进行滤波控制和极性选择。

外部计数模式框图如下图所示:



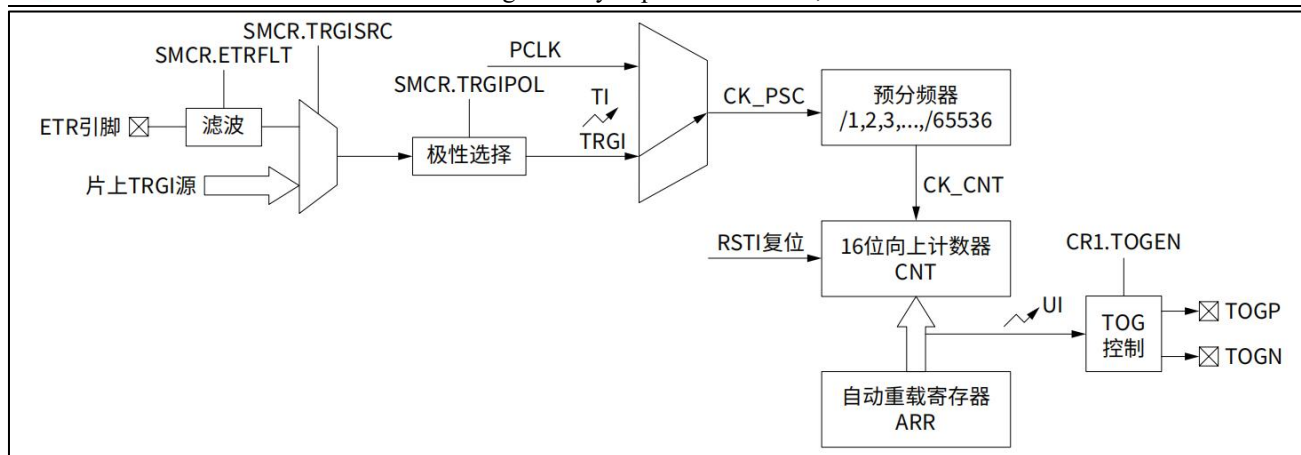


图 15-6 外部计数模式框图

设置 BTIMx\_CR0.EN 为 1 使能计数器，计数器将在 CK\_CNT 时钟信号的驱动下累加计数。当 TRGI 出现有效边沿时，BTIMx\_ISR.TIF 标志将置 1，向 BTIMx\_ICR.TIF 写 0 可清除该标志。

### 15.3.2.3 触发启动模式

当从模式控制寄存器 BTIMx\_SMCR 的 SMS 位域为 0x1 时，BTIM 配置为触发启动模式。在该模式下，设置 BTIMx\_CR0.EN 为 1 或触发信号 TRGI 出现上升沿时，将启动计数器 CNT 对内部时钟 PCLK 经预分频器分频后的 CK\_CNT 信号进行计数。

TRGI 信号的来源由从模式控制寄存器 BTIMx\_SMCR 的 TRGISRC 位域控制，并可通过 ETRFLT 位域设置 ETR 信号滤波，通过 TRGIPOL 位域设置 TRGI 信号的有效极性。

触发启动模式框图如下图所示：

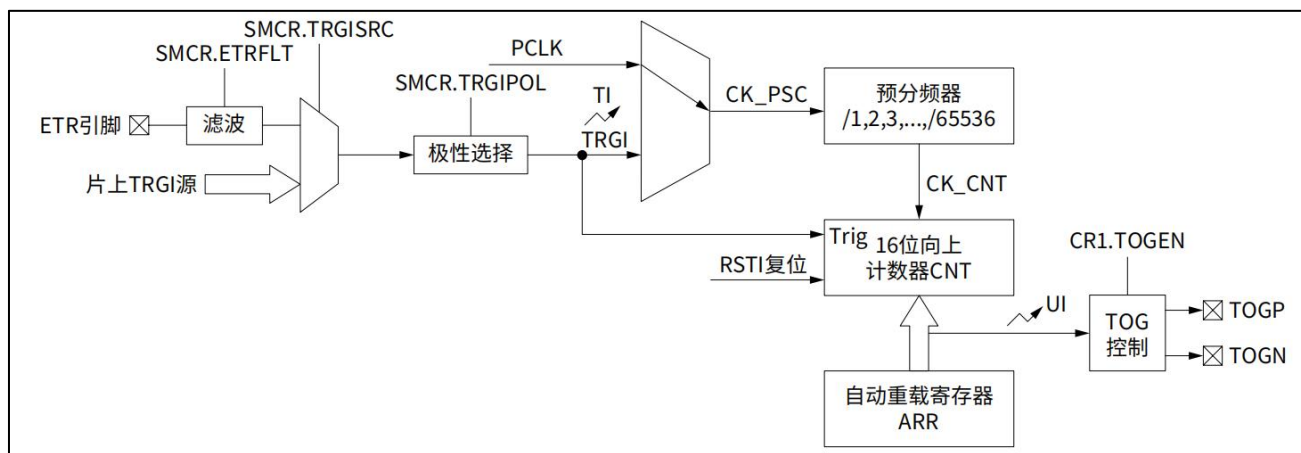


图 15-7 触发启动模式框图

当检测到有效的触发信号时，将产生以下影响：

1. BTIMx\_CR0.EN 被硬件置位；
2. 触发中断标志位 BTIMx\_ISR.TIF 置 1，可产生中断；
3. 计数器启动，开始计数。

计数器启动后，计数器从初始值开始向上计数，当计数值到达重载值 ARR 后产生溢出。在任意时候设置运行控制位 BTIMx\_CR0.EN 为 0 后，计数器立即暂停计数；再次设置运行控制位 BTIMx\_CR0.EN 为 1 或者触发信号有效时，计数器按前一次的设置继续计数。触发启动模式时序图如下图所示，其中 BTIMx\_PSC=0x01：

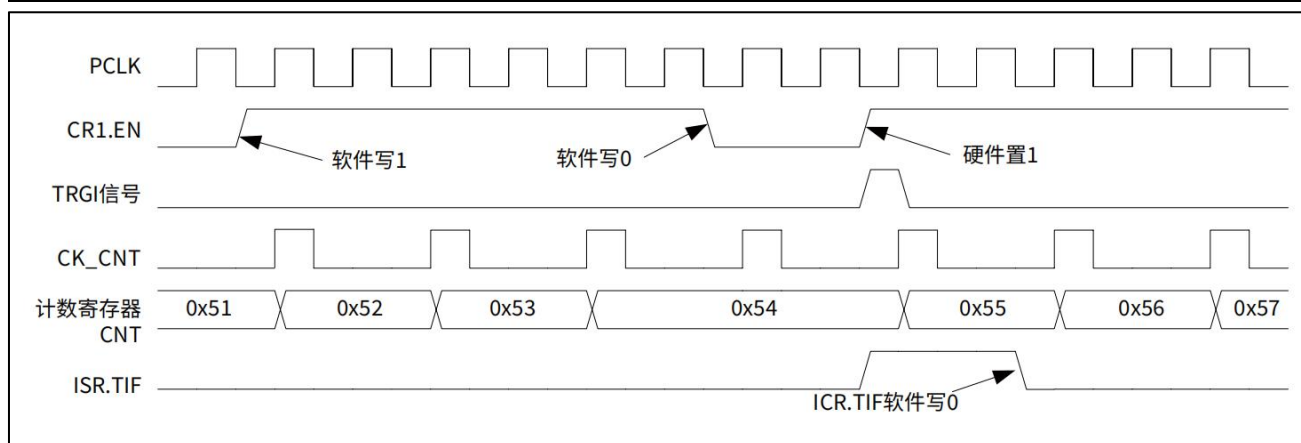


图 15-8 触发启动模式时序图

### 15.3.2.4 门控计数模式

当从模式控制寄存器 BTIMx\_SMCR 的 SMS 位域为 0x2 时，BTIM 配置为门控计数模式。在该模式下，触发输入 (TRGI) 为高电平且 BTIMx\_CR0.EN 为 1 时，将启动计数器对内部时钟 PCLK 经预分频器分频后的 CK\_CNT 信号进行计数；

触发输入 (TRGI) 为低电平或 BTIMx\_CR0.EN 为 0 时，计数器立即停止计数。

TRGI 信号的来源由从模式控制寄存器 BTIMx\_SMCR 的 TRGISRC 位域控制，并可通过 ETRFLT 位域设置 ETR 信号滤波，通过 TRGIPOL 位域设置 TRGI 信号的有效极性。

门控计数模式框图如下图所示：

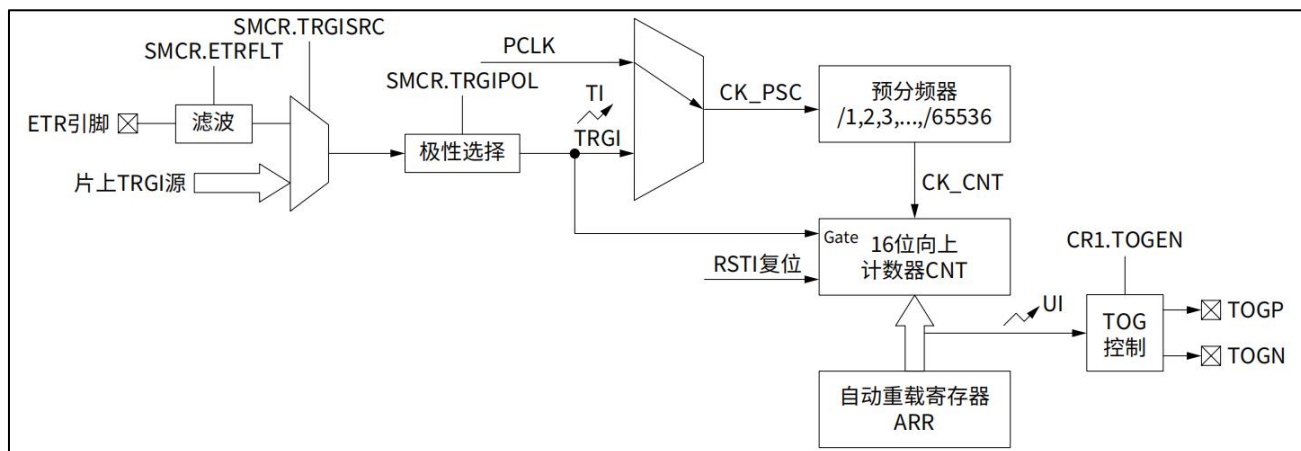


图 15-9 门控计数模式框图

### 15.3.3 UIF 位重映射

设置 BTIMx\_CR0 寄存器中的 UIFREMAP 位为 1 可使能 UIF 状态位重映射功能，可强制将更新中断标志 BTIMx\_ISR.UIF 连续复制到 BTIMx\_CNT 寄存器的 UIFCPY 位域中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

UIF 和 UIFCPY 标志使能之间没有延迟。

当 BTIMx\_CR0.UIFREMAP 为 0 时，UIFCPY 位域保留，读为 0。

### 15.3.4 BTIM 中断

BTIM 支持 2 个中断源，当 BTIM 中断事件发生时，中断标志位会被硬件置位，如果设置了对应的中断使能控制位，将产生中断请求。

在用户 BTIM 中断服务程序中，应查询相关 BTIM 中断标志位，以进行相应的处理，在退出中断服务程序之前，要清除该中断标志位，以避免重复进入中断服务程序。

各 BTIM 中断源的标志位、中断使能位、中断标志清除位或清除方法，如下表所示：

表 15-4 BTIM 中断控制

中断事件	中断标志位	中断使能位	中断标志清除
更新中断	ISR.UIF	IER.UIE	写 0 到 ICR.UIF
触发中断	ISR.TIF	IER.TIE	写 0 到 ICR.TIF

### 15.3.5 触发 ADC

BTIM 的触发输出信号 TRGO 可用于触发 ADC 启动，触发输出 TRGO 有多种可能的事件，具体由控制寄存器 BTIMx\_CR1 的 MMS 位域进行选择。同时，ADC 外设需配置其外部触发启动寄存器，以选择对应触发源。

应注意，必须先使能 ADC 时钟，才能从主定时器接收事件，且从定时器接收触发信号时，不得实时更改 ADC 时钟。

### 15.3.6 调试支持

BTIM 支持在调试模式下停止或继续计数，通过调试状态定时器控制寄存器 SYSCTRL\_DEBUG 的 BTIM123 位域来设置。

- 设置 SYSCTRL\_DEBUG.BTIM012 为 1，则在调试状态时暂停 BTIM0/1/2 的计数器计数。
- 设置 SYSCTRL\_DEBUG.BTIM012 为 0，则在调试状态时 BTIM0/1/2 的计数器继续计数。

## 15.4 编程示例

### 15.4.1 内部计数模式编程示例

1. 设置 SYSCTRL\_APBEN2.BTIM012 为 1，打开 BTIM0、BTIM1 和 BTIM2 的配置时钟及工作时钟；

**注：**BTIM0、BTIM1 和 BTIM2 共用 SYSCTRL\_APBEN.BTIM012 位。

2. 设置 BTIMX SMCR.SMS 为 0x00，使 BTIMx 工作于内部计数模式；
3. 配置 BTIMX CR0.ONESHOT，选择单次或连续计数模式。配置为 0，则为连续计数模式；配置为 1，则为单次计数模式；
4. 配置 BTIMX PSC，设置预分频器的分频比；
5. 配置 BTIMX ARR，设置 BTIMx 计数溢出时间；
6. 设置 BTIMX CNT 为 0x0000，以便计数；
7. 设置 BTIMX IER.UIE 为 1 并配置对应 NVIC，使能更新中断；
8. 设置 BTIMX CR0.EN 为 1，启动 BTIMx；
9. 当计数器溢出时，BTIMX ISR.UIF 标志位置 1，进入中断服务程序，设置 BTIMXICR.UIF 为 0 清除该中断标志。

## 15.4.2 外部计数模式编程示例

以下示例中，配置计数器在 BTIMX ETR 引脚的上升沿进行计数，步骤如下：

1. 设置 SYSCTRL\_APBEN2.BTIM012 为 1，打开 BTIM0、BTIM1 和 BTIM2 的配置时钟及工作时钟；  
**注：BTIM0、BTIM1 和 BTIM2 共用SYSCTRL APBEN.BTIM012 位。**
2. 将 BTIMx\_ETR 引脚对应的 GPIO 配置成复用输入模式；
3. 设置 BTIMX SMCR.TRGISRC 为 0xF，选择 TRG| 信号来源为 BTIMX\_ETR 引脚；
4. 设置 BTIMX SMCR.ETRFLT，配置 ETR 输入滤波带宽；
5. 设置 BTIMX SMCR.TRGIPOL 为 0，选择上升沿有效；
6. 设置 BTIMX SMCR.SMS 为 0x3，使 BTIMx 工作于外部计数模式；
7. 配置 BTIMx\_CR0.ONESHOT，选择单次或连续计数模式。配置为 0，则为连续计数模式，配置为 1，则为单次计数模式；
8. 配置 BTIMX PSC，设置预分频器的分频比；
9. 配置 BTIMX ARR，设置 BTIMx 计数溢出时间；
10. 设置 BTIMX IER.UIE 为 1 并配置对应 NVIC，使能更新中断；
11. 设置 BTIMX CR0.EN 为 1，启动 BTIMx；
12. 当计数器溢出时，BTIMx\_ISR.UIF 标志位置 1，进入中断服务程序，设置 BTIMx\_ICR.UIF 为 0 清除该中断标志。

## 15.4.3 触发启动模式编程示例

以下示例中，BTIMX ETR 输入出现上升沿时触发启动计数器，步骤如下：

1. 设置 SYSCTRL\_APBEN2.BTIM012 为 1，打开 BTIM0、BTIM1 和 BTIM2 的配置时钟及工作时钟；  
**注：BTIM0、BTIM1 和 BTIM2 共用SYSCTRL APBEN.BTIM012 位。**
2. 将 BTIMx\_ETR 引脚对应的 GPIO 配置成复用输入模式；
3. 设置 BTIMX SMCR.TRGISRC 为 0xF，选择 TRG| 信号来源为 BTIMX ETR 引脚；
4. 设置 BTIMX SMCR.ETRFLT，配置 ETR 输入滤波带宽；
5. 设置 BTIMX SMCR.TRGIPOL 为 0，选择上升沿有效；
6. 设置 BTIMX SMCR.SMS 为 0x1，使 BTIMx 工作于触发启动模式；
7. 配置 BTIMx\_CR0.ONESHOT，选择单次或连续计数模式。配置为 0，则为连续计数模式；配置为 1，则为单次计数模式；
8. 配置 BTIMX PSC，设置预分频器的分频比；
9. 配置 BTIMX ARR，设置 BTIMx 计数溢出时间；
10. 当 BTIMX ETR 输入出现上升沿时，计数器启动计数，同时 TIF 标志置 1。

## 15.4.4 门控计数模式编程示例

以下示例中，BTIMX ETR 引脚输入的信号作为门控信号，控制计数器计数，步骤如下：

1. 设置 SYSCTRL\_APBEN2.BTIM012 为 1，打开 BTIM0、BTIM1 和 BTIM2 的配置时钟及工作时钟；  
**注：BTIM0、BTIM1 和 BTIM2 共用SYSCTRL APBEN.BTIM012 位。**
2. 将 BTIMx\_ETR 引脚对应的 GPIO 配置成复用输入模式；设置 BTIMX SMCR.TRGISRC 为 0xF，选择 TRGI 信号来源为 BTIMX ETR 引脚；
3. 设置 BTIMX SMCR.ETRFLT，配置 ETR 输入滤波带宽；
4. 设置 BTIMX SMCR.TRGIPOL 为 0，选择高电平有效；

5. 设置 BTIMX SMCR.SMS 为 0x2, 使 BTIMx 工作于门控计数模式;
6. 配置 BTIMx\_CR0.ONESHOT, 选择单次或连续计数模式。配置为 0, 则为连续计数模式;配置为 1, 则为单次计数模式;
7. 配置 BTIMX PSC, 设置预分频器的分频比;
8. 配置 BTIMX ARR, 设置 BTIMx 计数溢出时间;
9. 设置 BTIMX CR0.EN 为 1, 使能计数器;
10. 当 BTIMX ETR 输入为高电平时, 计数器开始计数;
11. 当 BTIMx ETR 输入为低电平时, 计数器停止计数。

### 15.4.5 计数器复位编程示例

以下示例中, RSTI 输入出现上升沿时复位计数器, 步骤如下:

1. 根据编程示例, 配置 BTIM 的工作模式;
2. 配置 BTIMX SMCR.RSTISRC, 选择计数值复位信号 RSTI 来源;
3. 设置 BTIMX SMCR.RSTIPOL 为 0, 选择上升沿有效;
4. 当 RSTL 输入出现上升沿时, 计数器清零, 重新从 0 开始计数, 同时 UIF 标志置 1。

## 15.5 寄存器列表

BTIM0 基地址: BTIM0\_BASE = 0x4001 0800

BTIM1 基地址: BTIM1\_BASE = 0x4001 0C00

BTIM2 基地址: BTIM2\_BASE = 0x4001 3800

表 15-5 BTIM 寄存器列表

寄存器名称	寄存器地址	寄存器描述
BTIMx_CR0	BTIMx_BASE+0x00	控制寄存器 0
BTIMx_CR1	BTIMx_BASE+0x04	控制寄存器 1
BTIMx_SMCR	BTIMx_BASE+0x08	从模式控制寄存器
BTIMx_IER	BTIMx_BASE+0x0C	中断使能寄存器
BTIMx_ISR	BTIMx_BASE+0x10	中断标志寄存器
BTIMx_ICR	BTIMx_BASE+0x70	中断标志清除寄存器
BTIMx_EGR	BTIMx_BASE+0x14	事件生成寄存器
BTIMx_CNT	BTIMx_BASE+0x24	计数寄存器
BTIMx_PSC	BTIMx_BASE+0x28	预分频寄存器
BTIMx_ARR	BTIMx_BASE+0x2C	自动重载寄存器

## 15.6 寄存器描述

### 15.6.1 BTIMx\_CR0 控制寄存器 0

地址：请参见表 15-5 BTIM 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15	TOGEN	RW	TOG 引脚输出波形使能控制 0: TOGP、TOGN 均输出低电平 1: TOGP、TOGN 输出相反电平
14-12	RFU	-	保留位，请保持默认值
11	UIFREMAP	RW	UIF 状态位重映射使能控制 0: 禁止重映射，BTIMx_CNT[31]保持为 0 1: 使能重映射，BTIMx_CNT[31]等效于 BTIMx_ISR.UIF
10-4	RFU	-	保留位，请保持默认值
3	OPM	RW	单次/连续计数模式控制 0: 连续计数模式 1: 单次计数模式
2	URS	RW	更新请求源配置 0: 使能 UEV 时，所有以下事件都会产生更新中断 <ul style="list-style-type: none"> <li>计数器上溢出</li> <li>将 UG 位置 1</li> <li>通过从模式控制器生成的更新事件</li> </ul> 1: 使能 UEV 时，只有计数器上溢出会生成更新中断
1	UDIS	RW	更新禁止控制 0: 使能 UEV 更新事件可通过以下事件生成： <ul style="list-style-type: none"> <li>计数器上溢出</li> <li>将 UG 位置 1</li> <li>通过从模式控制器生成的更新事件</li> </ul> 1: 禁止 UEV 不会生成更新事件。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。
0	EN	RW	定时器运行控制 0: 定时器停止 1: 定时器运行



## 15.6.2 BTIMx\_CR1 控制寄存器 1

地址：请参见表 15-5 BTIM 寄存器列表 复位值：0x0000 0000

位域	名称	权限	功能描述
31-7	RFU	-	保留位，请保持默认值
6-4	MMS	RW	主模式输出信号（TRGO）配置 000：复位信号，EGR.UG 用作触发输出(TRGO)。如果复位由触发输入生成，则 TRGO 上的信号相比实际复位会有延迟。 001：使能信号，CR0.EN 用作触发输出(TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 EN 控制位与门控模式下的触发输入的逻辑与运算组合而成。当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 SMCR.MSM 的说明）。 010：更新信号，选择更新事件作为触发输出(TRGO)。例如，主定时器可用作从定时器的预分频器。 011：溢出信号，每当 BTIM 溢出时输出一个脉冲。
3-0	RFU	-	保留位，请保持默认值

## 15.6.3 BTIMx\_SMCR 从模式控制寄存器

地址：请参见表 15-5 BTIM 寄存器列表 复位值：0x0000 0000

位域	名称	权限	功能描述
31-18	RFU	-	保留位，请保持默认值
17	TRGIPOL	RW	触发输入信号（TRGI）极性选择 0：正相，高电平或上升沿有效 1：反相，低电平或下降沿有效
16	RSTIPOL	RW	计数值复位信号（RSTI）极性选择 0：正相，上升沿有效 1：反相，下降沿有效
15	RFU	-	保留位，请保持默认值
14-12	ETRFLT	RW	ETR 信号数字滤波电路采样时钟及采样点数配置 000:无滤波 001: $F_{sample}=PCLK, N=2$ 010: $F_{sample}=PCLK, N=4$ 011: $F_{sample}=PCLK, N=6$ 100: $F_{sample}=PCLK/4, N=4$ 101: $F_{sample}=PCLK/4, N=6$ 110: $F_{sample}=PCLK/16, N=4$ 111: $F_{sample}=PCLK/16, N=6$
11	MSM	RW	主/从模式配置 0：不执行任何操作 1：当前定时器的触发输入事件(TRGI)的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。

10-7	TRGISRC	RW	<p>触发输入信号（TRGI）来源配置</p> <p>0000：保留                      0111：ATIM_Trgo          0001：无复位信号            1000：BTIM0_Trgo          0010：VCx_OUT 信号        1001：BTIM1_Trgo          0011：SPI_NCS 信号        1010：BTIM2_Trgo          0100：I2C_SCL 信号        1011：GTIM_Trgo          0101：UARTx_RXD 信号    1100：保留          0110：保留                  1111：BTIMx_ETR 信号</p> <p>注 1：切勿选择 BTIMx 自己输出的 Trgo。</p> <p>注 2：当设置为 0010 时，BTIM0 触发输入信号来源为 VC0_OUT；BTIM1 触发输入信号来源为 VC1_OUT；此时 BTIM2 触发输入信号来源无效。</p>
6-3	RSTISRC	RW	<p>计数值复位信号（RSTI）来源配置</p> <p>0000：无复位信号                      0111：ATIM_Trgo          0001：I2C_SCL                          1000：BTIM0_Trgo          0010：VCx_OUT 信号                  1001：BTIM1_Trgo          0011：SPI_NCS 信号                  1010：BTIM2_Trgo          0100：无复位信号                      1011：GTIM_Trgo          0101：UARTx_RXD 信号              1100：无复位信号          0110：无复位信号                      1111：BTIMx_ETR 信号</p> <p>注 1：在 RSTI 信号的有效边沿复位计数器并生成更新事件。</p> <p>注 2：切勿选择 BTIMx 自己输出的 Trgo。</p> <p>注 3：当设置为 0010 时，BTIM1 复位信号来源为 VC1_OUT；          BTIM2 复位信号来源为 VC2_OUT；此时 BTIM3 复位信号来源无效。</p> <p>注 4：当设置为 0001 时，BTIM0 复位信号来源为 I2C_SCL，BTIM1/2 复位信号来源无效。</p>
2-0	SMS	RW	<p>从模式功能配置</p> <p>000：禁止从模式          001：触发启动模式：在 TRGI 上升沿使能计数器          010：门控计数模式：在 TRGI 高电平对 PCLK 计数          011：外部计数模式：在 TRGI 上升沿计数器自增 1</p>

### 15.6.4BTIMx IER 中断使能寄存器

地址：请参见表 15-5 BTIM 寄存器列表 复位值：0x0000 0000

位域	名称	权限	功能描述
31-7	RFU	-	保留位，请保持默认值
6	TIE	RW	触发中断使能控制 0：禁止 1：使能
5-1	RFU	-	保留位，请保持默认值
0	UIE	RW	更新中断使能控制 0：禁止 1：使能



### 15.6.5 BTIMx\_ISR 中断标志寄存器

地址：请参见表 15-5 BTIM 寄存器列表 复位值：0x0000 0000

位域	名称	权限	功能描述
31-7	RFU	-	保留位，请保持默认值
6	TIF	RO	触发中断标志 0：未发生触发事件 1：已发生触发事件
5-1	RFU	-	保留位，请保持默认值
0	UIF	RO	更新中断标志 0：未发生更新事件 1：已发生更新事件

### 15.6.6 BTIMx\_ICR 中断标志清除寄存器

地址：请参见表 15-5 BTIM 寄存器列表 复位值：0x0000 0041

位域	名称	权限	功能描述
31-7	RFU	-	保留位，请保持默认值
6	TIF	R1W0	触发标志清除 W0：清除触发标志 W1：无功能
5-1	RFU	-	保留位，请保持默认值
0	UIF	R1W0	更新标志清除 W0：清除计数器更新标志 W1：无功能

### 15.6.7 BTIMx\_EGR 事件生成寄存器

地址：请参见表 15-5 BTIM 寄存器列表 复位值：0x0000 0000

位域	名称	权限	功能描述
31-7	RFU	-	保留位，请保持默认值
6	TG	WO	此位由软件置 1 以生成事件，并由硬件自动清零 0：无功能 1：软件生成触发事件
5-1	RFU	-	保留位，请保持默认值
0	UG	WO	该位可通过软件置 1 以生成事件，并由硬件自动清零 0：无功能 1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器的计数器也将清零，但预分频比不受影响。

### 15.6.8 BTIMx\_CNT 计数寄存器

地址：请参见表 15-5 BTIM 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31	UIFCPY	RW	根据 BTIMx_CR0 寄存器中 UIFREMAP 位域的值，本位域表示不同的含义： UIFREMAP=0，本位域保留，读为 0 UIFREMAP=1，本位域表示 BTIMx_ISR 寄存器的 UIF 位的只读副本
30-16	RFU	-	保留位，请保持默认值
15-0	CNT	RW	定时器计数值

### 15.6.9 BTIMx\_PSC 预分频寄存器

地址：请参见表 15-5 BTIM 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	PSC	RW	预分频器的值 计数器时钟频率 CK_CNT 等于 $f_{CK\_PSC} / (PSC[15-0] + 1)$

### 15.6.10 BTIMx\_ARR 自动重载寄存器

地址：请参见表 15-5 BTIM 寄存器列表 复位值：0x0000 FFFF

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-0	ARR	RW	定时器重载值

## 16 通用异步收发器（UART）

### 16.1 概述

HC32F0653 内部集成 2 个通用异步收发器(UART)，支持异步全双工、同步半双工、同步全双工和单线半双工模式，支持多机通信；可编程数据帧结构，可以通过小数波特率发生器提供宽范围的波特率选择。

### 16.2 UART 特性

- 可编程数据帧结构
  - 数据字长：8、9 位，LSB / MSB 可选
  - 校验位：无校验、奇校验、偶校验
  - 停止位长度：1、1.5、2 位
- 16 位整数、4 位小数波特率发生器
- 可配置为 16 倍， 8 倍和 4 倍过采样
- 支持异步全双工、同步半双工、单线半双工、同步全双工
- 支持多机通信，自动地址识别
- 支持 LIN 模式
- 10 个带中断标志的中断源
- 错误检测：奇偶校验错误、帧结构错误

## 16.3 功能描述

### 16.3.1 功能框图

UART 控制器的功能框图如下图所示：

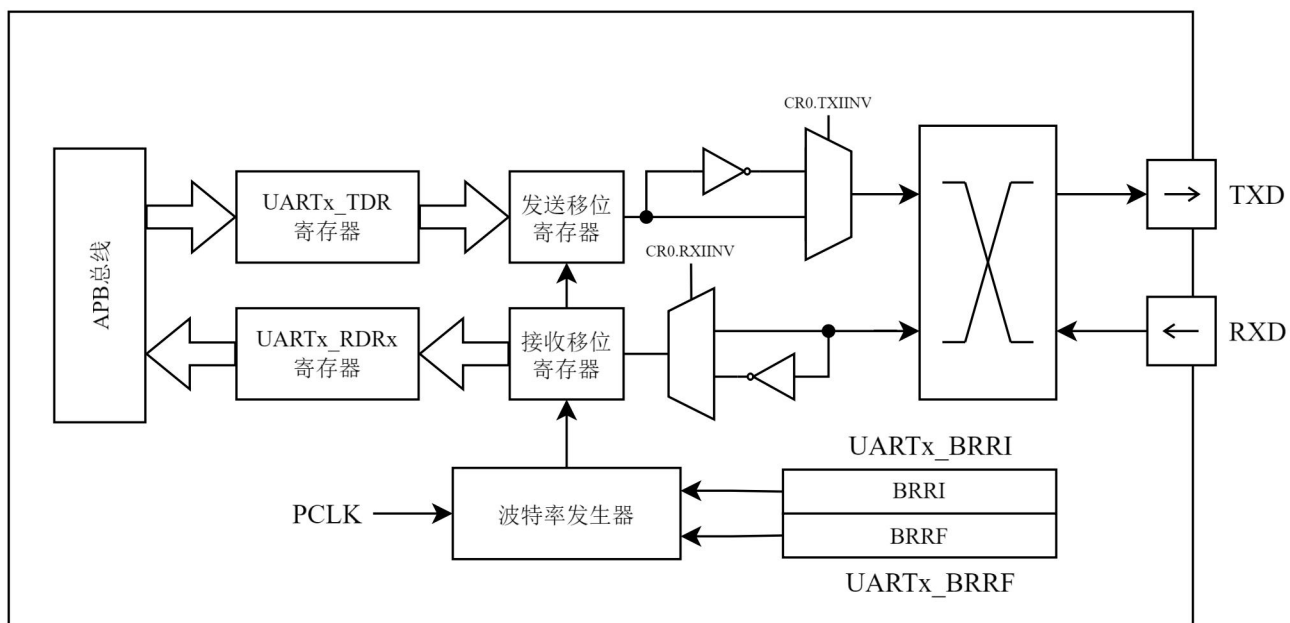


图 16-1 UART 功能框图

UART 控制器支持多种工作模式，在不同的工作模式下，各引脚具有不同的功能，引脚配置也不同，如下表所示（其中  $x=0, 1$ ）：

表 16-1 UART 端口配置

工作模式	UART 引脚	作用	GPIO 配置
异步全双工	UARTx_TXD	数据串行输出	数字，推挽输出 / 开漏输出，映射到 TXD
	UARTx_RXD	数据串行输入	数字，上拉输入，映射到 RXD
同步全双工	UARTx_TXD	数据串行输出	数字，推挽输出 / 开漏输出，映射到 TXD
	UARTx_RXD	数据串行输入	数字，上拉输入，映射到 RXD
	UARTx_CK	时钟信号输出	数字，推挽输出，映射到 CK
单线半双工	UARTx_TXD	数据的发送和接收	数字，开漏输出带上拉
	UARTx_RXD	不使用	可作通用 IO 使用

## 16.3.2 同步模式

UART 支持同步全双工工作模式。在该模式下，UARTx\_CK 引脚输出同步移位时钟信号，UARTx\_RXD 引脚进行数据的接收，UARTx\_TXD 引脚进行数据的发送。

UART 的同步模式可以与 SPI 的三线全双工模式进行通信，SPI 电平模式可选。

设置控制寄存器 UARTx\_CR0 的 SYNC 位域为 1，使 UART 工作于同步全双工模式。此时 UART 只能作为主机，数据字长可选 8/9 位，且 UARTx\_CR1.ADDREN 必须清零。

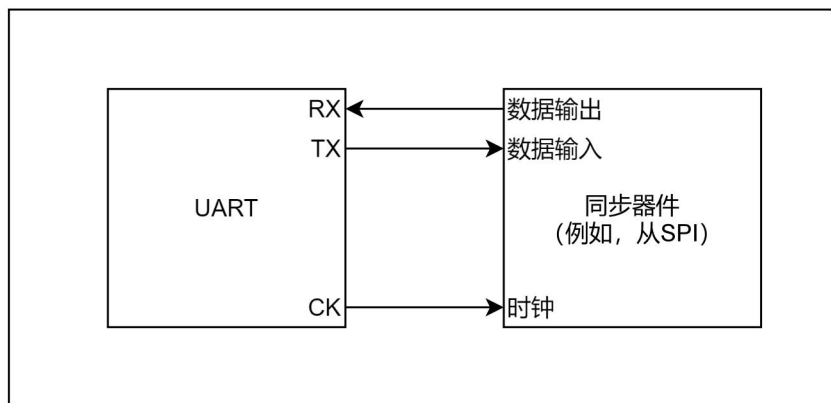


图 16-2 UART 同步全双工发送接收示例

### 16.3.2.1 波特率设置

同步半双工模式下，波特率计算公式见异步模式。

### 16.3.2.2 数据收发

将 UART 发送使能位 `UARTx_CR0.TXEN` 和接收使能位 `UARTx_CR0.RXEN` 置 1 使 UART 工作于全双工状态，将数据写入 `UARTx_TDR` 寄存器后，发送数据将从 `UARTx_TXD` 引脚串行输出，同时 `UARTx_CK` 引脚输出同步移位时钟信号，接收数据从 `UARTx_RXD` 引脚串行输入。当 `UARTx_TDR` 寄存器和移位寄存器中的数据均已发送完成后，发送完成中断标志位 `UARTx_ISR.TC` 和接收完成中断标志位 `UARTx_ISR.RC` 会被硬件置位。此时可以读取 `UARTx_RDR` 寄存器。收发下一个字节之前，将 `UARTx_ISR.RC` 和 `UARTx_ISR.TC` 标志位清零，RC 和 TC 标志清零后，可以立即开始收发下一个字节数据。

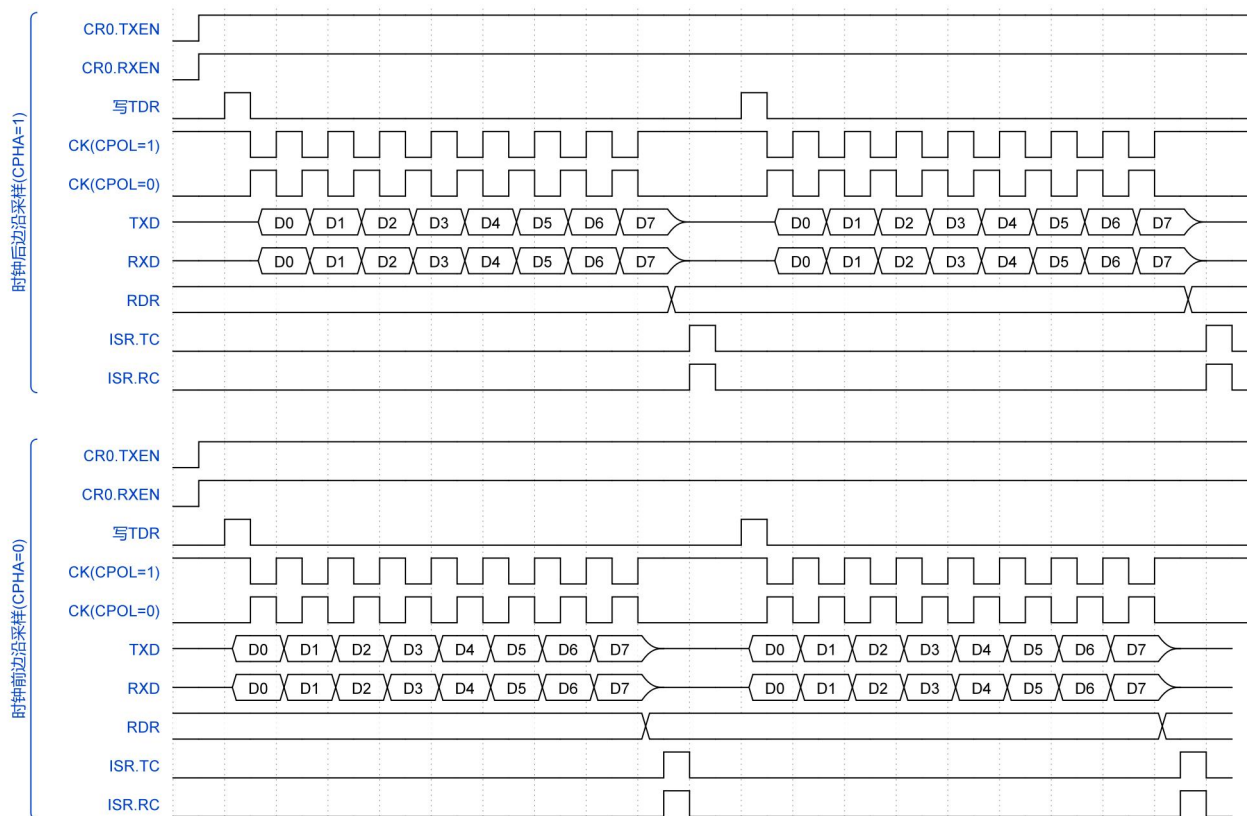


图 16-3 同步全双工模式接收发送时序（发送两个字节数据）

## 16.3.3 异步模式

### 16.3.3.1 数据帧格式

在 UART 异步通信中，数据是以数据帧的形式发送和接收的，一帧数据包括一个起始位、一个数据域、一个可选的校验位和宽度可编程的停止位。

#### 起始位

起始位长度固定为 1 位。

#### 数据域

数据字长可以设置为 8 位或 9 位，通过控制寄存器 UARTx\_CR0 的 CHLEN 位域来配置。当禁止奇偶校验时（设置 UARTx\_CR0.PARITYEN 为 0），数据字长可以设置为 8 位或 9 位；当使能奇偶校验时（设置 UARTx\_CR0.PARITYEN 为 1），数据字长必须设置为 9 位，如果字符长度设置为 8 位，PARITYEN 位域会自动清零。

数据位传输顺序可选择 MSB 或 LSB 在前，通过控制寄存器 UARTx\_CR0 的 MSBF 位域来选择。

#### 校验位

校验方式支持偶校验和奇校验，具体通过控制寄存器 UARTx\_CR0 的 PARITY 位域来选择。

- 奇校验：校验位使一帧数据中数据位和校验位中“1”的总数为奇数。
- 偶校验：校验位使一帧数据中数据位和校验位中“1”的总数为偶数。

#### 停止位

停止位长度可以配置为 1、1.5 或 2 位，具体通过控制寄存器 UARTx\_CR0 的 STOP 位域来配置。

表 16-2 数据帧结构

UARTx_CR0.CHLEN	UARTx_CR0.PARITYEN	数据帧
0	X	起始位+8 位数据+停止位
1	0	起始位+9 位数据+停止位
1	1	起始位+8 位数据+奇偶校验位+停止位

### 16.3.3.2 小数波特率发生器

#### 波特率的产生

UART 的接收和发送波特率是相同的，由同一个波特率发生器产生。

波特率发生器支持 16 倍采样，8 倍采样和 4 倍采样 3 种采样模式，具体的采样模式通过控制寄存器 UARTx\_CR0 的 OVER 位域来选择。

1. OVER = 00，设置 16 倍采样，波特率计算公式：

$$\text{BaudRate} = \text{UCLK} / (16 \times \text{BRRI} + \text{BRRF})$$

UCLK 是 UART 的传输时钟，其来源是 PCLK。

BRRI (UARTx\_BRRI [15:0])，是波特率计数器的整数部分，可设置范围为 1 ~ 65535。

BRRF (UARTx\_BRRF [3:0])，是波特率计数器的小数部分，可设置范围为 0 ~ 15。

例 1：

当传输时钟 UCLK 的频率为 24MHz 时，设置 BRRI = 156 (即 UARTx\_BRRI = 0x9C)，BRRF = 4 (即 UARTx\_BRRF = 0x04)，则：

$$\text{BaudRate} = 24000000 / (16 \times 156 + 4) = 9600 \text{ bps}$$

例 2：

当传输时钟 UCLK 的频率为 24MHz 时，要求配置 BaudRate = 115200 bps，计算

$$16 \times \text{BRRI} + \text{BRRF} = 24000000 / 115200 = 208.33$$

则：

$$\text{BRRI} = 208.33 / 16 = 13.02, \text{ 最接近的整数是: } 13 \text{ (0x0D)}$$

$$\text{BRRF} = 0.02 \times 16 = 0.32, \text{ 最接近的整数是: } 0 \text{ (0x00)}$$

即需要设置 UARTx\_BRRI 为 0x0D，UARTx\_BRRF 为 0x00

此时，实际波特率 BaudRate = 115384.62 bps，误差率为 0.16%

2. OVER = 01，设置 8 倍采样，波特率计算公式：

$$\text{BaudRate} = \text{UCLK} / (8 \times \text{BRRI})$$

#### 波特率设置示例

表 16-3~表 16-8 列举了常用 MCU 频率时，波特率计数寄存器的设定值及波特率误差。

表 16-3 UCLK 为 4MHz 波特率设置示例 (OVER = 0)

波特率 (bps)	BRRI	BRRF	实际波特率	误差率
4800	52	1	4801.92	0.04%
9600	26	1	9592.33	-0.08%
14400	17	6	14388.49	-0.08%
19200	13	0	19230.77	0.16%
38400	6	8	38461.54	0.16%
56000	4	7	56338.03	0.60%
57600	4	5	57971.01	0.64%
115200	2	3	114285.71	-0.79%
256000	1	0	250000	-2.34%



表 16-4 UCLK 为 8MHz 波特率设置示例 (OVER = 0)

波特率 (bps)	BRRI	BRRF	实际波特率	误差率
4800	104	3	4799.04	-0.02%
9600	52	1	9603.84	0.04%
14400	34	12	14388.49	-0.08%
19200	26	1	19184.65	-0.08%
38400	13	0	38461.54	0.16%
56000	8	15	55944.06	-0.10%
57600	8	11	57553.96	-0.08%
115200	4	5	115942.03	0.64%
500000	1	0	500000	0.00%

表 16-5 UCLK 为 16MHz 波特率设置示例 (OVER = 0)

波特率 (bps)	BRRI	BRRF	实际波特率	误差率
4800	208	5	4800.48	0.01%
9600	104	3	9598.08	-0.02%
14400	69	7	14401.44	0.01%
19200	52	1	19207.68	0.04%
38400	26	1	38369.30	-0.08%
56000	17	14	55944.06	-0.10%
57600	17	6	57553.96	-0.08%
115200	8	11	115107.91	-0.08%
1000000	1	0	1000000	0.00%

表 16-6 UCLK 为 24MHz 波特率设置示例 (OVER = 0)

波特率 (bps)	BRRI	BRRF	实际波特率	误差率
4800	312	8	4800.00	0.00%
9600	156	4	9600.00	0.00%
14400	104	3	14397.12	-0.02%
19200	78	2	19200.00	0.00%
38400	39	1	38400.00	0.00%
56000	26	13	55944.06	-0.10%
57600	26	1	57553.96	-0.08%
115200	13	0	115384.62	0.16%
1500000	1	0	1500000	0.00%

表 16-7 UCLK 为 32MHz 波特率设置示例 (OVER = 0)

波特率 (bps)	BRRI	BRRF	实际波特率	误差率
4800	416	11	4799.76	0.00%
9600	208	5	9600.96	0.01%
14400	138	14	14401.44	0.01%
19200	104	3	19196.16	-0.02%
38400	52	1	38415.37	0.04%
56000	35	11	56042.03	0.08%
57600	34	12	57553.96	-0.08%
115200	17	6	115107.91	-0.08%
2000000	1	0	2000000	0.00%

表 16-8 UCLK 为 48MHz 波特率设置示例 (OVER = 0)

波特率 (bps)	BRRI	BRRF	实际波特率	误差率
4800	625	0	4800.00	0.00%
9600	312	8	9600.00	0.00%
14400	208	5	14401.44	0.01%
19200	156	4	19200.00	0.00%
38400	78	2	38400.00	0.00%
56000	53	9	56009.33	0.02%
57600	52	1	57623.05	0.04%
115200	26	1	115107.91	-0.08%
3000000	1	0	3000000	0.00%

### 波特率自动检测

HC32F0653 使用 UART 作为从机进行通信时,可以通过自动波特率检测的方法,自动适应 UART 主机的波特率。可将通用定时器 (GTIM) 的输入捕获来源配置为 UART 的 RXD 信号,或者将 GTIM 的门控信号配置为 UART 的 RXD 信号,配合使用相关软件算法测量 UART 的波特率,以实现波特率自适应。

### 16.3.3.3 发送控制

#### 数据发送

设置 UARTx\_CR0.TXEN 为 1 使能发送电路。将数据写入 UARTx\_TDR 寄存器后,数据会被硬件转移到发送移位寄存器,发送移位寄存器将数据串行移出(数据位传输顺序可选择 MSB 或 LSB 在前)。

当数据被硬件转移到发送移位寄存器后,发送缓冲器空中断标志位 UARTx\_ISR.TXE 会被硬件置位,表示 UARTx\_TDR 寄存器已空,此时允许对 UARTx\_TDR 寄存器写入新的待发送数据。在对 UARTx\_TDR 寄存器写入数据的同时,UARTx\_ISR.TXE 标志位会被自动清零。如果此时发送移位寄存器中尚有未发送完成的数据,那么新写入的数据会在 UARTx\_TDR 寄存器中暂存,在当前数据发送完成后,UARTx\_TDR 寄存器中的数据会被硬件转移到发送移位寄存器中继续发送。

当发送移位寄存器中的数据帧发送完成后,如果 UARTx\_TDR 寄存器中没有待发送的数据(即 UARTx\_ISR.TXE = 1),发送完成中断标志位 UARTx\_ISR.TC 会被硬件置位,同时 UART 发送忙标志位 UARTx\_ISR.TXBUSY 会被硬件清零,表示数据已全部发送完成,UART 发送器空闲。

UART 发送控制环节的时序图如下图所示:

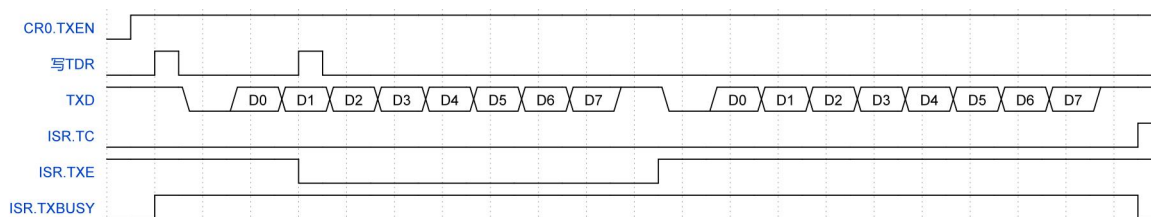


图 16-4 异步工作模式发送控制时序（发送两个字节数据）

### 间隔段帧

间隔段帧是完全由‘0’组成的一个完整的数据帧（停止位期间也是‘0’）。在间隔段帧结束时，发送器再插入 2 个比特的逻辑‘1’，以保证能识别下一帧的起始位。

设置 `UARTx_TDR.BREAK` 为 1 将发送间隔段帧，间隔段帧长度由 UART 数据字长决定。如果写入 `BREAK` 时，存在正在发送的数据，则等待当前数据发送完成后，将在 `TXD` 上发送一个间隔段帧。

### 空闲帧

空闲帧是完全由‘1’组成的一个完整的数据帧（停止位期间也是‘1’），后面跟着包含了数据的下一帧的起始位。

设置 `UARTx_TDR.IDLE` 为 1 将发送空闲帧，空闲帧长度由 UART 数据字长决定。如果写入 `IDLE` 时，存在正在发送的数据，则等待当前数据发送完成后，将在 `TXD` 上发送一个空闲帧。

间隔段帧和空闲帧的时序图如下图所示，其中数据字长为 8 位：

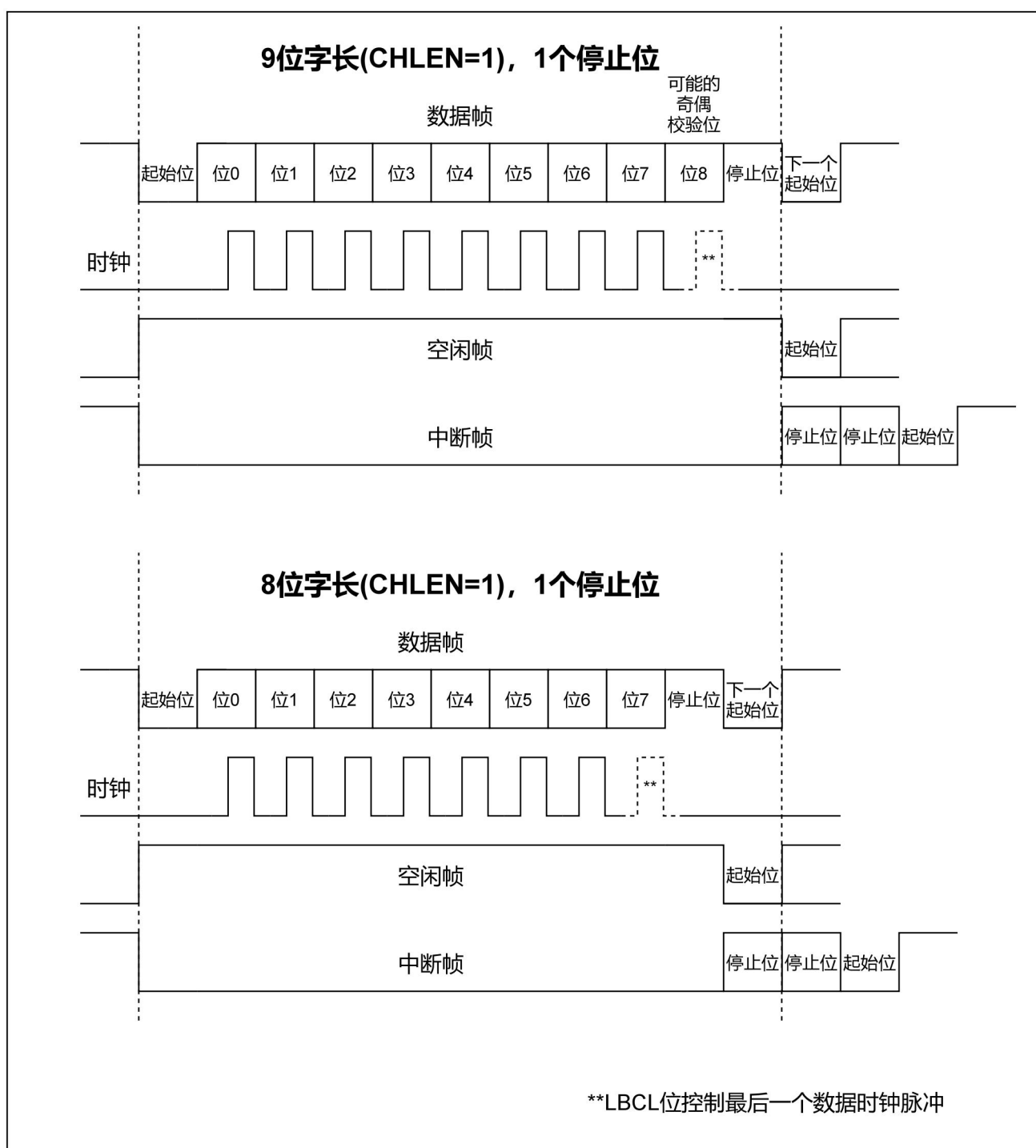


图 16-5 间隔段帧和空闲帧时序

### 16.3.3.4 接收控制

#### 接收数据

清除连续接收次数计数器 `UARTx_RCNT`，设置连续接受次数配置寄存器 `UARTx_RCNTCFG` 为  $x$  ( $1 \leq x \leq 8$ )，设置 `UARTx_CR1.RXEN` 为 1 使能接收电路。当接收器侦测到起始位后，开始接收数据，接收期间，数据从最低位或最高位开始串行移入接收移位寄存器，并行转移到 `UARTx_RDR0` 寄存器，此时数据可以被读出。

当接收数据从移位寄存器转移到 `UARTx_RDR` 寄存器后，连续接收次数计数器 `UARTx_RCNT` 会自动加一，接收完成中断标志位 `UARTx_ISR.RC` 会被硬件置位，表示已经完成一帧数据的接收，假如连续接收次数计数器 `UARTx_RCNT` 等于连续接受次数配置寄存器 `UARTx_RCNTCFG`，连续接收完成中断标志位 `UARTx_ISR.RCONT` 会被硬件置位，表示接收完一组数据。用户程序中，一旦检测到 `UARTx_ISR.RCONT` 标志位为 1，应尽快读取 `UARTx_RDR0` 到 `UARTx_RDRx` 寄存器，并清除 `UARTx_ISR.RCONT` 标志位和 `UARTx_RCNT`。如果未及时读取 `UARTx_RDR0` 到 `UARTx_RDRx` 寄存器，新接收的数据会覆盖 `UARTx_RDRx` 寄存器中的数据，造成数据丢失。

在接收一个数据帧时，会自动进行奇偶校验，若检测到奇偶校验错误，`UARTx_ISR.PE` 标志位会被硬件置位，表示奇偶校验出错。

在接收一个数据帧时，如果在预期时间内未识别到正确的停止位，则判定发生帧结构错误，`UARTx_ISR.FE` 标志位会被硬件置位。

#### 噪声错误

当在接收帧中检测到噪声时，在 `UARTx_ISR.RC` 标志位的上升沿 `UARTx_ISR.NE` 标志位被硬件置位，无效数据从移位寄存器转移到 `UARTx_RDR` 寄存器。如果设置 `UARTx_IER.NE` 为 1，将产生噪声中断，设置 `UARTx_ICR.NE` 为 0，可清除噪声标志。注意噪声错误仅在 16 倍采样或 8 倍采样时有效。

表 16-9 通过采样数据进行噪声检测

采样值	NE 状态	接收的位值
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

#### 溢出错误

如果 `UARTx_ISR.RCNT` 大于等于 `UARTx_ISR.RCNTCFG`，又接收到一个字符，则发生溢出错误，`UARTx_ISR.ORE` 标志位会被硬件置位。

发生溢出错误后，新接收的数据会覆盖 `UARTx_RDR[UARTx_ISR.RCNTCFG-1]` 寄存器中的数据，造成数据丢失。

如果设置 `UARTx_IER.ORE` 为 1，将产生溢出中断，设置 `UARTx_ICR.ORE` 为 0，可清除溢出标志。

#### 间隔段帧

当 `UART` 接收到间隔段帧时，`RXBRK`、`RC` 和 `FE` 标志都会置位，`UARTx_RDR` 寄存器接收到 0，如果使能相应的中断，将产生对应的中断请求。

#### 空闲帧

每当接收完成一个字符时，才会启动空闲字符检测。当检测到空闲字符时，`RXIDLE` 标志置位，如果使能相应的中断，将产生中断请求。

### 16.3.3.5 单线半双工模式

将 UARTx\_TXD 引脚和 UART\_RXD 引脚映射到同一个 GPIO

软件控制 UARTx\_TXD 引脚进入发送状态，写数据到 UARTx\_TDR 寄存器，输出 UARTx\_TDR 寄存器中的数据。数据发送完成后，软件控制 UARTx\_TXD 引脚恢复到常态的接收状态。

没有发送数据时，软件控制 UARTx\_TXD 引脚处于接收状态，数据接收完成后，接收完成标志位 UARTx\_ISR.RC 和 UARTx\_ISR.RCONT 会被硬件置位，此时应尽快读取 UARTx\_RDR 寄存器，并清除 UARTx\_ISR.RC 标志位、UARTx\_ISR.RCONT 标志位和 UARTx\_RCNT 连续接受次数寄存器。

注：

用户应采取适当的应用层保护机制，以确保不会出现多主机同时向总线发送数据。

### 16.3.3.6 多机通信

UART 支持多机通信方式。在该模式下，UART 总线上有一个主机和多台从机，每个从机有唯一的从机地址，通信时主机会先发送地址帧对从机寻址，只有地址匹配的从机才被激活，接收随后主机发送的数据帧。

#### 主机发送

多机通信模式下，主机需将帧数据长度设置为 9 位，并禁止奇偶校验。

UARTx\_TDR.TDR 的最高位决定主机发送地址帧还是数据帧，UARTx\_TDR[8] 为 1 表示主机发送的是地址帧，UARTx\_TDR[8] 为 0 表示主机发送的是数据帧。

#### 从机接收

多机通信模式下，从机需将帧数据长度设置为 9 位，并禁止奇偶校验，同时设置 UARTx\_CR1.ADDREN 为 1，使能从机地址识别，从机硬件自动检测主机发送的地址与本机地址是否匹配。

如果地址匹配，从机会将接收到的地址帧保存到 UARTx\_RDR 寄存器中，UARTx\_ISR.RC 标志位被硬件置位，同时 UARTx\_ISR.SLVMATCH 标志位被硬件置位，从机接收随后主机发送的数据帧。通信过程中，从机需要：

1. 应用程序在接收完成中断 RC 里查询 UARTx\_RDR[8]，以判断接收到的是地址帧还是数据帧。
2. 从机在发送数据帧时，需要将 UARTx\_TDR[8] 设置为 0，以避免被其它从机当作地址帧。

如果地址不匹配，从机不会接收主机发送的数据帧，也不产生接收完成中断，已置位的 UARTx\_ISR.SLVMATCH 标志位将被清零。

#### 从机地址与地址掩码

从机地址由 UARTx\_ADDR 寄存器配置，从机地址应配置为唯一。UARTx\_MASK 寄存器是地址掩码，UARTx\_MASK [7:0] 中为‘1’的位对应的从机地址位参与从机地址匹配运算，为‘0’的位对应的从机地址位 则不参与从机地址匹配运算。

当 UARTx\_MASK [7:0] 中全部位设置为‘1’时，即向 UARTx\_MASK 寄存器写入 0xFF，则从机地址的 8 位地址 位全部参与从机地址匹配，主机能唯一识别到从机。

当 UARTx\_MASK [7:0] 中部分位设置为‘0’时，对应的从机地址位不参与从机地址匹配，用以实现多个从机响 应主机发出的同一地址帧，即主机对多个从机同时寻址。

例 1： 对从机 A 设置：UARTx\_ADDR = 0xA0，UARTx\_MASK = 0xFE

对从机 B 设置：UARTx\_ADDR = 0xA1，UARTx\_MASK = 0xFE

对从机 C 设置：UARTx\_ADDR = 0xA2，UARTx\_MASK = 0xFC

对从机 D 设置：UARTx\_ADDR = 0xA3，UARTx\_MASK = 0xFC

对从机 E 设置：UARTx\_ADDR = 0xA5，UARTx\_MASK = 0xFF

主机发送 0xA0 或 0xA1 地址帧时，可以寻址从机 A、从机 B、从机 C、从机 D

主机发送 0xA2 或 0xA3 地址帧时，可以寻址从机 C、从机 D

主机发送 0xA4 地址帧时，没有从机被寻址

主机发送 0xA5 地址帧时，可以寻址从机 E

#### 广播地址

多机通信中，地址 0xFF 被定义为广播地址，主机发送广播地址帧时，所有地址的从机都被寻址。

### 16.3.4 LIN 模式

通过将 UARTx\_CR2 寄存器中的 LIN 位域设置为 1 来选择 LIN 模式。在 LIN 模式下，必须将 UARTx\_CR0 寄存器中的 SIGNAL、SYNC、STOP 和 PARITY 位清零。

LIN 的一帧 (Frame) 数据包含帧头 (Header) 和应答 (Response) 两部分。主机任务负责发送帧头，从机任务接收帧头并对帧头所包含信息进行解析，并对帧头作出反应 (接收 / 发送 / 忽略应答部分)。LIN 的帧结构如下图所示：

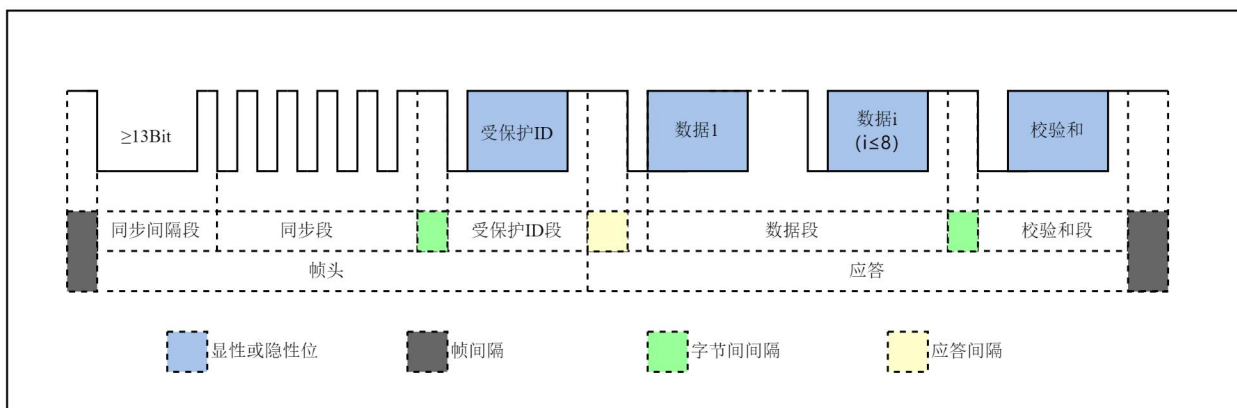


图 16-6 LIN 帧的结构

在 LIN 的一帧当中，除同步间隔段（Break Field），其它各段都是通过字节域的格式传输的。字节域包括 1 位起始位 +8 位数据位 +1 位停止位，数据传输都是最低有效位（LSB）在前。

#### 16.3.4.1 LIN发送

##### 同步间隔段的发送

同步间隔段表示一帧数据的开始，LIN 间隔段的比特数量由 UARTx\_TDR.TDR 设置，注意同步间隔是至少持续 13 位（以主机节点的位速率为准）的显性电平。

在 LIN 模式下，设置 UARTx\_TDR.BREAK 为 1，同时向 UARTx\_TDR.TDR 中写入间隔段的比特数量，将立即发送设定长度的间隔段。

##### 同步段的发送

向 UARTx\_TDR 寄存器中写入 0x55，将发送同步段。

##### 受保护 ID 段的发送

受保护 ID 段的前 6 位是帧 ID，加上两个奇偶校验位后称作受保护 ID。

根据校验公式，计算出帧 ID 对应的奇偶校验位，然后将受保护 ID 写入数据寄存器 UARTx\_TDR，将发送受保护 ID 段。

#### 16.3.4.2 LIN接收

##### 间隔段的检测

使能 LIN 模式后，将激活间隔段检测电路。通过控制寄存器 UARTx\_CR2 的 BRKL 位域设置接收间隔段的长度阈值。

使能接收电路后，硬件开始监测 RXD 引脚输入的起始信号。检测到起始位后，电路会对接下来的位进行采样，当检测到 RXD 通信线低电平宽度大于等于 BRKL 位域设定的间隔段长度阈值，且其后跟随间隔符，则 UARTx\_ISR.RXBRK 标志位会被硬件置位，表示间隔段接收完成。如果 UARTx\_IER.RXBRK



为 1，则会产生中断请求。

如果在设定的间隔段长度阈值之前，已经采样到高电平，则间隔段检测电路会取消当前检测，并重新搜索起始位。

LIN 模式下，当发生帧结构错误时，接收器立即停止，直到间隔段检测电路接收到“1”（间隔段不完整）或接收到间隔符（检测到间隔段）。

间隔段检测时，帧结构错误标志位 UARTx\_ISR.FE 会被硬件置位，清除 UARTx\_ISR.RXBRK 标志位的同时，需要清除 UARTx\_ISR.FE 标志，并读取 UARTx\_RDR 寄存器，以确认接收到的数据为 0x00。

#### 同步段的检测

从机节点通过接收主机节点发出的同步段，计算出主机节点位速率，根据计算结果对自身的位速率重新作调整。

## 16.4 低功耗模式

当系统进入深度休眠模式后，高速时钟会停止运行，UART 不会接收数据。此时，仍可通过 GPIO 中断唤醒 MCU，实现在深度休眠模式下接收数据，参考配置步骤如下：

1. 使能 UARTx\_RXD 对应引脚的 GPIO 下降沿中断；
2. 使能 UART 接收（即设置 UARTx\_CR0.RXEN 为 1）；
3. 进入深度休眠模式；
4. 等待主机发送数据，产生 GPIO 下降沿中断，唤醒 MCU；
5. 关闭 RXD 对应引脚的 GPIO 中断功能，等待 RXD 接收完成。

## 16.5 UART 中断

UART 控制器支持 10 个中断源，当 UART 中断触发事件发生时，中断标志位会被硬件置位，如果设置了对应的中断使能控制位，将产生中断请求。

HC32F0653 的一个 UART 模块使用一个相同的系统 UART 中断，UART 中断是否产生中断跳转由嵌套向量中断控制器 (NVIC) 的中断使能设置寄存器 NVIC\_IUSER 的相应位控制。

在用户 UART 中断服务程序中，应查询相关 UART 中断标志位，以进行相应的处理，在退出中断服务程序之前，要清除该中断标志位，避免重复进入中断程序。

各 UART 中断源的标志位、中断使能位、中断标志清除位或清除方法，如下表所示：

中断事件	中断标志位	中断使能位	标志清除方法
发送缓冲器空	ISR.TXE	IER.TXE	写 UARTx_TDR 寄存器
发送完成	ISR.TC	IER.TC	写 0 到 ICR.TC
接收完成	ISR.RC	IER.RC	写 0 到 ICR.RC
空闲字符接收完成	ISR.RXIDLE	IER.RXIDLE	写 0 到 ICR.RXIDLE
间隔段接收完成	ISR.RXBRK	IER.RXBRK	写 0 到 ICR.RXBRK
帧结构错误	ISR.FE	IER.FE	写 0 到 ICR.FE
奇偶校验错误	ISR.PE	IER.PE	写 0 到 ICR.PE
噪声标志	ISR.NE	IER.NE	写 0 到 ICR.NE
溢出错误	ISR.ORE	IER.ORE	写 0 到 ICR.ORE
连续接收完成标志	ISR.RCONT	IER.RCONT	写 0 到 ICR.RCONT



## 16.6 编程示例

### 16.6.1 异步全双工编程示例

#### 16.6.1.1 查询方式发送数据

1. 设置 `SYSCTRL_AHBEN.GPIOx` 为 1, `SYSCTRL_APBENx.UARTx` 为 1, 使能 UART 引脚对应的 GPIO 时钟和 UART 配置时钟;
2. 将 `UARTx_TXD` 引脚配置成推挽复用输出模式;
3. 设置 `UARTx_CR0.SYNC` 为 0, 配置 `UARTx` 为异步全双工通信模式;
4. 配置数据帧;
5. 校验位: 配置 `UARTx_CR0.PARITY`
6. 停止位: 配置 `UARTx_CR0.STOP`
7. 配置 `UARTx_CR0.OVER`, 选择采样模式;
8. 配置 `UARTx_BRR1` 和 `UARTx_BRRF` 寄存器, 配置波特率;
9. 设置 `UARTx_CR0.TXEN` 为 1 使能发送;
10. 设置 `UARTx_ICR.TC` 为 0, 清除发送完成标志位;
11. 将要发送的一帧数据写入 `UARTx_TDR` 寄存器;
12. 查询等待 `UARTx_ISR.TC` 标志位置 1, 确认一帧数据发送完成;
13. 重复步骤 9 至步骤 12, 发送下一帧数据。

#### 16.6.1.2 查询方式接收数据

1. 设置 `SYSCTRL_AHBEN.GPIOx` 为 1, `SYSCTRL_APBENx.UARTx` 为 1, 使能 UART 引脚对应的 GPIO 时钟和 UART 配置时钟;
2. 将 `RXD` 引脚配置成上拉输入复用模式;
3. 设置 `UARTx_CR0.SYNC` 为 0, 配置 `UARTx` 为异步全双工通信模式;
4. 配置数据帧;
5. 校验位: 配置 `UARTx_CR0.PARITY`
6. 停止位: 配置 `UARTx_CR0.STOP`
7. 配置 `UARTx_CR0.OVER`, 选择采样模式;
8. 配置 `UARTx_RCONTCFG` 配置连续接收次数;
9. 配置 `UARTx_BRR1` 和 `UARTx_BRRF` 寄存器, 配置波特率;
10. 向 `UARTx_ICR` 寄存器写入 0x00, 清除所有标志位;
11. 向 `UARTx_RCNT` 寄存器写入 0x00, 清除连续接受次数;
12. 设置 `UARTx_ICR.RCONT` 为 0, 清除连续接收完成标志位;
13. 设置 `UARTx_CR0.RXEN` 为 1 使能接收;
14. 查询等待 `UARTx_ISR.RC` 标志位置 1, 确认接收完一帧数据;
15. 查询错误标志 `UARTx_ISR.PE` 和 `UARTx_ISR.FE`, 确认数据是否有效, 如果数据无效, 则进行出错处理, 如果数据有效, 则读取对应的 `UARTx_RDRx` 寄存器并保存数据;
16. 设置 `UARTx_ICR.RC` 为 0, 清除接收完成标志位;
17. 查询 `UARTx_ISR.RCONT` 标志位是否置一, 确认是否接收完一组数据, 是则跳转下一步骤, 否则跳转到步骤 14;
18. 设置 `UARTx_ICR.RCONT` 为 0, 清除连续接收完成标志位;
19. 设置 `UARTx_RCNT` 为 0, 清除连续接收次数标志位;
20. 重复步骤 13 至步骤 17, 接收下一组数据。

### 16.6.1.3 中断方式发送数据

1. 设置 SYSCTRL\_AHBEN.GPIOx 为 1, SYSCTRL\_APBENx.UARTx 为 1, 使能 UART 引脚对应的 GPIO 时钟和 UART 配置时钟;
2. 将 TXD 引脚配置成推挽复用输出模式;
3. 设置 UARTx\_CR0.SYNC 为 0, 配置 UARTx 为异步全双工通信模式;
4. 配置数据帧;
5. 校验位: 配置 UARTx\_CR0.PARITY
6. 停止位: 配置 UARTx\_CR0.STOP
7. 配置 UARTx\_CR1.OVER, 选择采样模式;
8. 配置 UARTx\_BRR1 和 UARTx\_BRRF 寄存器, 配置波特率;
9. 配置 NVIC 控制器, 请参见 5 中断章节;
10. 设置 UARTx\_IER.TXE 为 1, 使能发送缓冲器空中断;
11. 设置 UARTx\_CR0.TXEN 为 1 使能发送;
12. 将要发送的一帧数据写入 UARTx\_TDR 寄存器;
13. 数据开始发送, 发送缓冲器空, 进入中断服务函数: 查询判断 UARTx\_ISR.TXE 标志位, 如果标志位为 1, 写一帧新的数据到 UARTx\_TDR 寄存器;
14. 查询等待 UARTx\_ISR.TXBUSY 标志位清零, 关闭 UART。

### 16.6.1.4 中断方式接收数据

1. 设置 SYSCTRL\_AHBEN.GPIOx 为 1, SYSCTRL\_APBENx.UARTx 为 1, 使能 UART 引脚对应的 GPIO 时钟和 UART 配置时钟;
2. 将 RXD 引脚配置成上拉输入复用模式;
3. 设置 UARTx\_CR0.SYNC 为 0, 配置 UARTx 为异步全双工通信模式;
4. 配置数据帧;
5. 校验位: 配置 UARTx\_CR0.PARITY
6. 停止位: 配置 UARTx\_CR0.STOP
7. 配置 UARTx\_CR0.OVER, 选择采样模式;
8. 配置 UARTx\_BRR1 和 UARTx\_BRRF 寄存器, 配置波特率;
21. 配置 UARTx\_RCONTCFG 配置连续接收次数为 1;
9. 配置 NVIC 控制器;
10. 设置 UARTx\_IER.RC 为 1 使能接收完成中断;
11. 设置 UARTx\_CR0.RXEN 为 1 使能接收;
12. 等待一帧数据接收完成, 进入中断服务函数: 查询 UARTx\_ISR.PE 和 UARTx\_ISR.FE 标志位, 确认数据是否有效, 如果无效, 则进行出错处理; 如果有效, 查询判断 UARTx\_ISR.RC 标志位, 如果标志位为 1, 则读取 UARTx\_RDR 寄存器并保存数据, 并清除该标志位。

## 16.6.2 同步全双工编程示例

### 16.6.2.1 数据发送与接收

1. 设置 `SYSCTRL_AHBEN.GPIOx` 为 1, `SYSCTRL_APBENx.UARTx` 为 1, 使能 GPIO 时钟和 UART 配置时钟;
2. 将 `UARTx_TXD`、`UARTx_CK` 引脚配置成推挽复用输出模式, 将 `UARTx_RXD` 引脚配置成数字输入模式;
3. 设置 `UARTx_CR0.SYNC` 为 1, 根据需求配置 `UARTx_CR0.CPHL`、`CPHA`、`LCBL` 时钟极性、相位和脉冲;
4. 配置波特率;
22. 配置 `UARTx_RCONTCFG` 配置连续接收次数为 1;
5. 设置 `UARTx_CR0.TXEN` 为 1 使能发送;
6. 设置 `UARTx_CR0.RXEN` 为 1 使能接收;
7. 设置 `UARTx_ICR.TC` 为 0, 清除发送完成标志位;
8. 设置 `UARTx_ICR.RC` 为 0, 清除接收完成标志位;
9. 将要发送的 8 位数据写入 `UARTx_TDR` 寄存器;
10. 查询等待 `UARTx_ISR.TC` 标志位置 1, 确认数据发送完成; 同时查询等待 `UARTx_ISR.RC` 标志位置 1, 确认数据接收完成, 将 `UARTx_RDR` 寄存器的数据读出。
11. 重复步骤 7 至步骤 10, 发送下一帧数据;

## 16.6.3 单线半双工编程示例

### 16.6.3.1 查询方式发送数据

1. 设置 `SYSCTRL_AHBEN.GPIOx` 为 1, `SYSCTRL_APBENx.UARTx` 为 1, 使能 UART 引脚对应的 GPIO 时钟和 UART 配置时钟;
2. 将 `UARTx_TXD` 和 `RXD` 引脚配置成开漏复用输出模式, 映射至同一个 GPIO 上, 并外接上拉电阻;
3. 配置数据帧;
4. 校验位: 配置 `UARTx_CR0.PARITY`
5. 停止位: 配置 `UARTx_CR0.STOP`
6. 配置 `UARTx_CR0.OVER`, 选择采样模式;
7. 配置 `UARTx_BRR1` 和 `UARTx_BRR2` 寄存器, 配置波特率;
8. 配置 `UARTx_CR0.SIGNAL` 为 1;
9. 设置 `UARTx_CR0.TXEN` 为 1, 使能发送;
10. 设置 `UARTx_ICR.TC` 为 0, 清除发送完成标志位;
11. 将要发送的一帧数据写入 `UARTx_TDR` 寄存器;
12. 查询等待 `UARTx_ISR.TC` 标志位置 1, 确认数据发送完成;
13. 重复步骤 9 至步骤 12, 发送下一帧数据。

### 16.6.3.2 查询方式接收数据

1. 设置 SYSCTRL\_AHBEN.GPIOx 为 1, SYSCTRL\_APBENx.UARTx 为 1, 使能 UART 引脚对应的 GPIO 时钟和 UART 配置时钟;
2. 将 UARTx\_TXD 和 RXD 引脚配置成开漏复用输出模式, 映射至同一个 GPIO 上, 并外接上拉电阻;
3. 配置数据帧;
4. 校验位: 配置 UARTx\_CR0.PARITY
5. 停止位: 配置 UARTx\_CR0.STOP
6. 配置 UARTx\_CR0.OVER, 选择采样模式;
7. 配置 UARTx\_BRR1 和 UARTx\_BRR2 寄存器, 配置波特率;
8. 配置 UARTx\_CR0.SIGNAL 为 0;
9. 向 UARTx\_ICR 写入 0x00, 清除所有标志位;
10. 设置 UARTx\_CR0.RXEN 为 1 使能发送和接收;
11. 查询等待 UARTx\_ISR.RC 标志位置 1, 确认接收完一帧数据;
12. 查询错误标志 UARTx\_ISR.PE 和 UARTx\_ISR.FE, 确认数据是否有效, 如果数据无效, 则进行出错处理; 如果数据有效, 则读取 UARTx\_RDR 寄存器并保存数据;
13. 设置 UARTx\_ICR.RC 为 0, 清除接收完成标志位;
14. 重复步骤 10 至步骤 13, 接收下一帧数据。

## 16.7 寄存器列表

UART0 基地址: UART0\_BASE = 0x4001 1400

UART1 基地址: UART1\_BASE = 0x4001 1800

表 16-10 UART 寄存器列表

寄存器名称	寄存器地址	寄存器描述
UARTx_CR0	UARTx_BASE + 0x00	控制寄存器 0
UARTx_CR1	UARTx_BASE + 0x04	控制寄存器 1
UARTx_CR2	UARTx_BASE + 0x38	控制寄存器 2
UARTx_IER	UARTx_BASE + 0x08	中断使能寄存器
UARTx_BRRI	UARTx_BASE + 0x0C	波特率计数器整数部分寄存器
UARTx_BRRF	UARTx_BASE + 0x10	波特率计数器小数部分寄存器
UARTx_ISR	UARTx_BASE + 0x1C	中断标志寄存器
UARTx_ICR	UARTx_BASE + 0x20	中断标志清除寄存器
UARTx_TDR	UARTx_BASE + 0x28	发送数据寄存器
UARTx_ADDR	UARTx_BASE + 0x30	从机地址寄存器
UARTx_MASK	UARTx_BASE + 0x34	从机地址掩码寄存器
UARTx_RCNTCFG	UARTx_BASE + 0x40	连续接收次数配置寄存器
UARTx_RCNT	UARTx_BASE + 0x44	连续接收次数
UARTx_RDR0	UARTx_BASE + 0x50	连续接收数据寄存器 0
UARTx_RDR1	UARTx_BASE + 0x54	连续接收数据寄存器 1
UARTx_RDR2	UARTx_BASE + 0x58	连续接收数据寄存器 2
UARTx_RDR3	UARTx_BASE + 0x5C	连续接收数据寄存器 3
UARTx_RDR4	UARTx_BASE + 0x60	连续接收数据寄存器 4
UARTx_RDR5	UARTx_BASE + 0x64	连续接收数据寄存器 5
UARTx_RDR6	UARTx_BASE + 0x68	连续接收数据寄存器 6
UARTx_RDR7	UARTx_BASE + 0x6C	连续接收数据寄存器 7

## 16.8 寄存器描述

### 16.8.1 UARTx\_CR0 控制寄存器 0

地址：请参见 表 16-10 UART 寄存器列表     复位值：0x0000 0000

位域	名称	权限	功能描述
31-18	RFU	-	保留位，请保持默认值
17	CPOL	RW	串行时钟极性配置 0：待机时低电平 1：待机时高电平
16	CPHA	RW	串行时钟相位配置 0：前边沿采样 / 后边沿移位 1：前边沿移位 / 后边沿采样
15	LBCL	RW	最后一位时钟脉冲 0：最后一位数据的时钟脉冲不从 CK 输出； 1：最后一位数据的时钟脉冲会从 CK 输出。 注：最后一个数据位就是第 8 或者第 9 个发送的位(根据 UART_CR0 寄存器中的 CHLEN 位所定义的 8 或者 9 位数据帧格式)。
14	SYNC	RW	传输同步异步模式设置 0：工作于异步模式 1：工作于同步模式 注：LIN 模式下，必须保持该控制位为 0。
13-11	RFU	-	保留位，请保持默认值
10-9	OVER	RW	采样方式设置 00：16 倍采样， $Baud=UCLK/(BRRI \times 16 + BRRF)$ 01：8 倍采样， $Baud=UCLK/(BRRI \times 8)$ 10：4 倍采样， $Baud=UCLK/(BRRI \times 4)$ 11：保留 注：当 BRRF 为非零值，自动采用 16 倍采样。
8	RFU	-	保留位，请保持默认值
7	MSBF	RW	数据帧高低位顺序选择 0：最低有效位 LSB 收发在前 1：最高有效位 MSB 收发在前
6	CHLEN	RW	字符长度 0：8bit 1：9bit 注：如果使能奇偶校验，则字符的最后 1Bit 为奇偶校验位。
5-4	STOP	RW	停止位长度设置 00：1 位 01：1.5 位 10：2 位 11：保留 注 1：同步模式下 STOP 位必须清零；

			<b>注2: LIN 模式下STOP 位必须清零。</b>
3	PARITYEN	RW	奇偶校验使能 0: 禁止 1: 使能 <b>注: 当字符长度为8bit 时, 奇偶校验使能自动清零。</b>
2	PARITY	RW	奇偶校验选择 0: 偶校验 1: 奇校验
1	RXEN	RW	接收电路使能控制 0: 禁止 1: 使能
0	TXEN	RW	发送电路使能控制 0: 禁止 1: 使能 注: 使能发送电路后, TXD 将发送一个空闲帧

## 16.8.2 UARTx\_CR1 控制寄存器 1

地址: 请参见 表 16-10 UART 寄存器列表 复位值: 0x0000 0000

位编号	位符号	权限	说明
31-18	RFU	-	保留位, 请保持默认值
17-15	RXSRC	RW	RXD 输入信号来源配置 000: UARTx_RXD 引脚 001: VC0_OUT 010: VC1_OUT 其他: 保留
14-6	RFU	-	保留位, 请保持默认值
5	TXINV	RW	TXD 引脚输出信号反相控制 0: TXD 引脚输出正相信号 1: TXD 引脚输出反相信号
4	RXINV	RW	RXD 引脚输入信号反相控制 0: RXD 引脚信号直接送入接收电路 1: RXD 引脚信号反相送入接收电路
3-1	RFU	-	保留位, 请保持默认值
0	ADDREN	RW	从机地址识别使能控制 0: 禁止 1: 使能

### 16.8.3 UARTx\_CR2 控制寄存器 2

地址：请参见 表 16-10 UART 寄存器列表 复位值：0x0000 0000

位域	名称	权限	功能描述
31-2	RFU	-	保留位，请保持默认值
1	BRKL	RW	LIN 模式，接收间隔段的长度阈值 0：10bit 1：11bit
0	LIN	RW	LIN 工作模式使能 0：工作于 UART 模式 1：工作于 LIN 模式

### 16.8.4 UARTx\_BRRI 波特率计数器整数部分寄存器

地址：请参见 表 16-10 UART 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15-13	BF_LENGTH	RW	LIN 模式间隔段的长度 00：间隔段长度 = 12 01：间隔段长度 = 13 10：间隔段长度 = 10 11：间隔段长度 = 11
12-0	BRRI	RW	波特率计数器整数部分

### 16.8.5 UARTx\_BRRF 波特率计数器小数部分寄存器

地址：请参见 表 16-10 UART 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-4	RFU	-	保留位，请保持默认值
3-0	BRRF	RW	波特率计数器小数部分

### 16.8.6 UARTx\_TDR 发送数据寄存器

地址：请参见 表 16-10 UART 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-11	RFU	-	保留位，请保持默认值
10	BREAK	WO	发送间隔段帧 0：发送正常数据 1：发送间隔段帧 注：UART 模式的间隔段长度由 UART 字长决定。
9	IDLE	WO	发送空闲帧 0：发送正常数据 1：发送空闲帧



			注 1: 空闲帧长度由 UART 字长决定; 注 2: 空闲帧优先级高于间隔帧。
8-0	TDR	WO	待发送的数据

## 16.8.7 UARTx\_IER 中断使能寄存器

地址: 请参见 表 16-10 UART 寄存器列表    复位值: 0x0000 0000

位编号	位符号	权限	说明
31-13	RFU	-	保留位, 请保持默认值
17	RCONTE	RW	连续接收完成中断使能控制 0: 禁止 1: 使能
16-12	RFU	-	保留位, 请保持默认值
11	ORE	RW	溢出中断使能控制 0: 禁止 1: 使能
10	NE	RW	噪声中断使能控制 0: 禁止 1: 使能
9	PE	RW	奇偶校验错误中断使能控制 0: 禁止 1: 使能
8	FE	RW	帧结构错误中断使能控制 0: 禁止 1: 使能
7-5	RFU	-	保留位, 请保持默认值
4	RXBRK	RW	间隔段接收完成中断使能控制 0: 禁止 1: 使能
3	RXIDLE	RW	空闲字符接收完成中断使能控制 0: 禁止 1: 使能
2	RC	RW	接收完成中断使能控制 0: 禁止 1: 使能
1	TC	RW	发送完成中断使能控制 0: 禁止 1: 使能
0	TXE	RW	发送缓冲器空中断使能控制 0: 禁止 1: 使能

## 16.8.8 UARTx\_ISR 中断标志寄存器

地址：请参见 表 16-10 UART 寄存器列表 复位值：0x0000 0001

位编号	位符号	权限	说明
31-18	RFU	-	保留位，请保持默认值
17	RCONT	RO	连续接收完成标志 UARTx_RCNT 等于 UARTx_RCNTCFG 时，硬件将该位置位 0：连续接收未完成 1：连续接收已完成 <b>注：清除该标志位，不会清零 UART_RCNT</b>
16-15	RFU	-	保留位，请保持默认值
14	TXBUSY	RO	TXD 发送状态 0：TDR 寄存器及发送移位寄存器空 1：TDR 寄存器或发送移位寄存器非空
13	SLVMATCH	RO	接收从机地址匹配标志 0：未检测到匹配的地址 1：已检测到匹配的地址
12	RFU	-	保留位，请保持默认值
11	ORE	RO	溢出标志 UARTx_RCNT 等于 UARTx_RCNTCFG 时，移位寄存器中当前正在接收的数据准备好传输至 UARTx_RDRx 寄存器时，硬件将该位置位。 0：无溢出错误 1：检测到溢出错误 <b>注：发生溢出错误后，新接收的数据会覆盖 UARTx_RDRx 寄存器中的数据。</b>
10	NE	RO	噪声标志 0：未检测到噪声 1：检测到噪声 <b>注：OVER 为 2 或 3 时，该位无效，保持为 0。</b>
9	PE	RO	奇偶校验错误中断标志 0：未检测到奇偶校验错误 1：已检测到奇偶校验错误
8	FE	RO	帧结构错误中断标志 0：未检测到帧结构错误 1：已检测到帧结构错误
7-5	RFU	-	保留位，请保持默认值
4	RXBRK	RO	间隔段接收完成标志 0：未接收到间隔段 1：已接收到间隔段
3	RXIDLE	RO	空闲字符接收完成标志 0：未接收到空闲字符 1：已接收到空闲字符 <b>注：每当接收完成一个字符时，才会启动空闲字符检测。</b>

2	RC	RO	接收完成中断标志 0: 尚未完成一帧数据的接收 1: 已经完成一帧数据的接收 注: STOP 为 1bit, RC 收到 1bitSTOP 位置 1; STOP 为 1.5bit, RC 收到 0.5bitSTOP 位置 1; STOP 为 2bit, RC 收到 0.5bitSTOP 位置 1。
1	TC	RO	发送完成中断标志 0: 发送缓冲器或移位寄存器中的数据尚未发送完成 1: 发送缓冲器和移位寄存器中的数据均已发送完成
0	TXE	RO	发送缓冲器空标志 0: 发送缓冲器内有数据 1: 发送缓冲器内无数据 注: 写 UARTx_TDR 寄存器将清除该标志位。

### 16.8.9 UARTx\_ICR 中断标志清除寄存器

地址: 请参见 表 16-10 UART 寄存器列表 复位值: 0x0002 005E

位编号	位符号	权限	说明
31-7	RFU	-	保留位, 请保持默认值
17	RCONT	R1W0	连续接收完成标志 0: 连续接收未完成 1: 连续接收已完成
16-7	RFU	-	保留位, 请保持默认值
6	RO	R1W0	接收溢出标志位清除控制 W0: 清除奇偶校验错误中断标志 W1: 无功能
5	RFU	-	保留位, 请保持默认值
4	PE	R1W0	奇偶校验错误中断标志清除控制 W0: 清除奇偶校验错误中断标志 W1: 无功能
3	FE	R1W0	帧结构错误中断标志清除控制 W0: 清除帧结构错误中断标志 W1: 无功能
2	RC	R1W0	接收完成中断标志清除控制 W0: 清除接收完成中断标志 W1: 无功能
1	TC	R1W0	发送完成中断标志清除控制 W0: 清除发送完成中断标志 W1: 无功能
0	RFU	-	保留位, 请保持默认值

### 16.8.10 UARTx\_ADDR 从机地址寄存器

地址：请参见 表 16-10 UART 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-8	RFU	-	保留位，请保持默认值
7-0	ADDR	RW	从机地址寄存器

### 16.8.11 UARTx\_MASK 从机地址掩码寄存器

地址：请参见 表 16-10 UART 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-8	RFU	-	保留位，请保持默认值
7-0	MASK	RW	从机地址掩码寄存器

### 16.8.12 UARTx\_RCNTCFG 连续接收次数配置寄存器

地址：请参见 表 16-10 UART 寄存器列表 复位值：0x0000 0001

位编号	位符号	权限	说明
31-4	RFU	-	保留位，请保持默认值
3-0	CNT	RW	待接收的数据个数 注：该位写 0 时，硬件会强制修改为 1； 注：该位写超过 8 的数，硬件会强制修改为 8。

### 16.8.13 UARTx\_RCNT 连续接收次数寄存器

地址：请参见 表 16-10 UART 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-4	RFU	-	保留位，请保持默认值
3-0	CNT	RW	已接收的数据个数 可以通过该寄存器修改已接收的数据个数 注：该位写超过 8 的数，硬件会强制修改为 8。

### 16.8.14 UARTx\_RDRx 连续接收数据寄存器 0(x = 0..7)

地址：请参见 表 16-10 UART 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-9	RFU	-	保留位，请保持默认值
8-0	RDR	RO	已接收的数据

## 17 串行外设接口总线 (SPI)

### 17.1 概述

串行外设接口 (SPI) 协议支持与外部器件进行全双工和单工同步串行通信。可配置 MCU 作为主机或从机，作主机时，可为外部从设备提供通信时钟(SCK)。接口还能以多主配置方式工作。

### 17.2 主要特性

主或从操作

- 3 线全双工同步传输
- 双线单工同步传输（带单向数据线）
- 8/16-bit 数据大小选择
- 支持多主模式
- 8 个主模式波特率预分频器，最大为  $f_{CLK}/8$ 。
- 从模式频率最大为  $f_{CLK}/8$ 。
- 主模式和从模式下均可以由软件或硬件进行 NSS 管理：主/从操作模式的动态改变
- 可编程的时钟极性和相位
- 可编程的数据顺序，MSB 在前或 LSB 在前
- 可触发中断的专用发送和接收标志
- SPI 总线忙状态标志
- 支持 SPI Motorola 模式
- 可触发中断的主模式故障和过载标志
- 支持 NSS 脉冲模式

## 17.3 功能描述

### 17.3.1 SPI 功能说明

#### 17.3.1.1 概述

SPI 支持 MCU 与外部器件之间进行同步串行通信。应用软件可通过轮询状态标志或使用 SPI 专用中断对通信进行管理。SPI 的主要组件及其交互方式如下图所示。

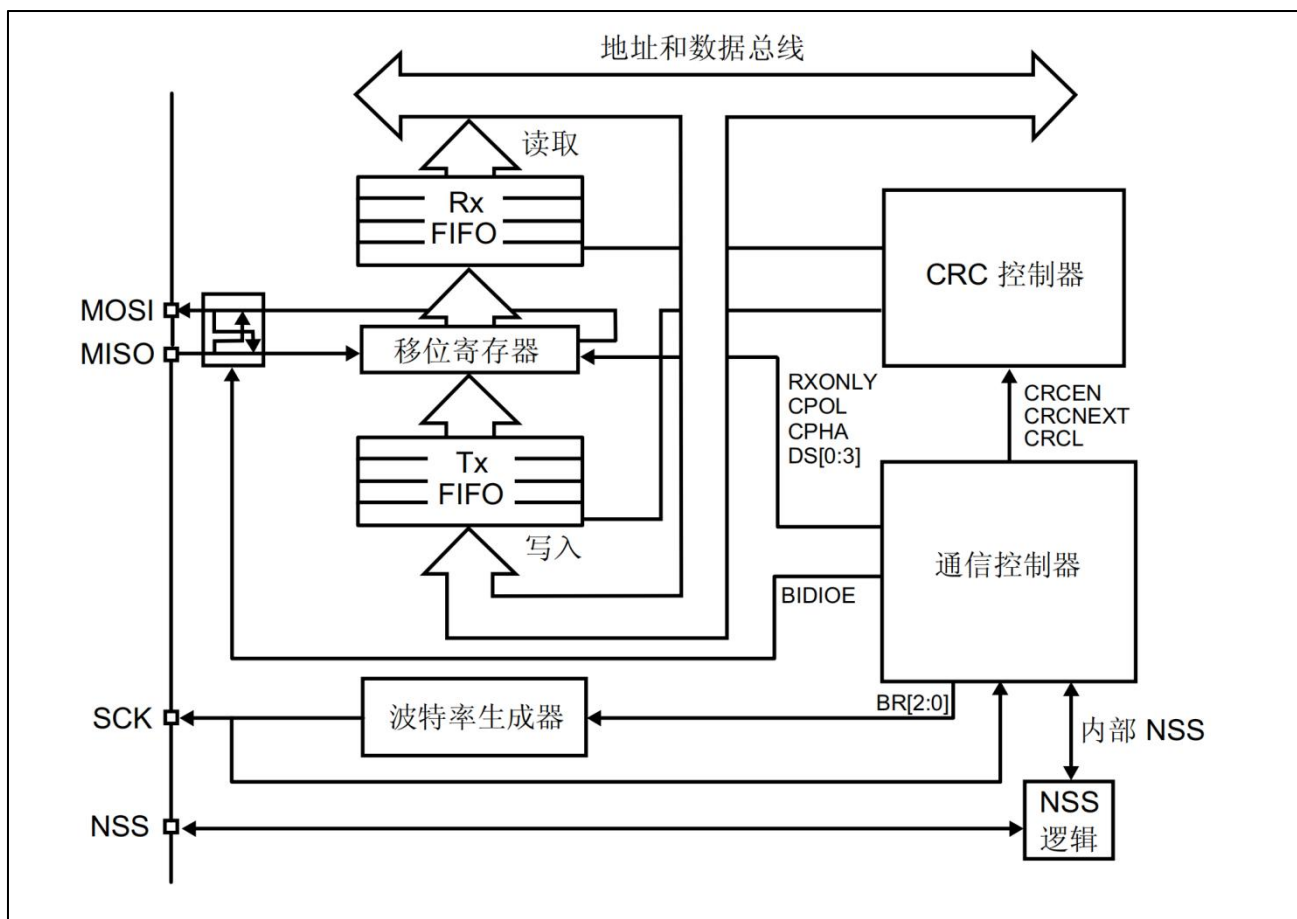


图 17-1 SPI 框图

SPI 通过 4 个专用引脚与外部器件通讯。

- **MISO**：主输入/从输出数据引脚。通常情况下，该引脚在从模式下发送数据，在主模式下接收数据。
- **MOSI**：主输出/从输入数据引脚。通常情况下，该引脚在主模式下发送数据，在从模式下接收数据。
- **SCK**：该引脚在主模式下发送数据，在从模式下接收数据。
- **NSS**：从器件选择引脚。根据 SPI 和 NSS 设置，该引脚可用于：
  - 选择单个从器件进行通信
  - 同步数据帧或
  - 检测多个主器件之间是否存在冲突

SPI 总线支持一个主器件与一个或多个从器件之间进行通信。该总线至少由两条线构成——一条用于

时钟信号，另一条用于同步数据传输。其他信号可以根据 SPI 节点间的数据交换及其从器件选择信号管理进行添加。

### 17.3.1.2 一个主器件和一个从器件之间的通信

SPI 支持 MCU 基于目标器件和应用要求使用不同的配置进行通信。这些配置使用 2 条或 3 条线（通过软件 NSS 管理），也可以使用 3 条或 4 条线（通过硬件 NSS 管理）。通信始终由主器件发起。

#### 全双工通信

默认情况下，SPI 配置为全双工通信。在这种配置下，主器件和从器件的移位寄存器通过 MOSI 和 MISO 引脚之间的两条单向线连接。在 SPI 通信过程中，数据随主器件提供的 SCK 时钟边沿同步移位。主器件通过 MOSI 线将待发送的数据发送给从器件，并通过 MISO 线从从器件接收数据。当数据帧传输完成时（所有位均移出），主器件和从器件之间即完成信息交换。

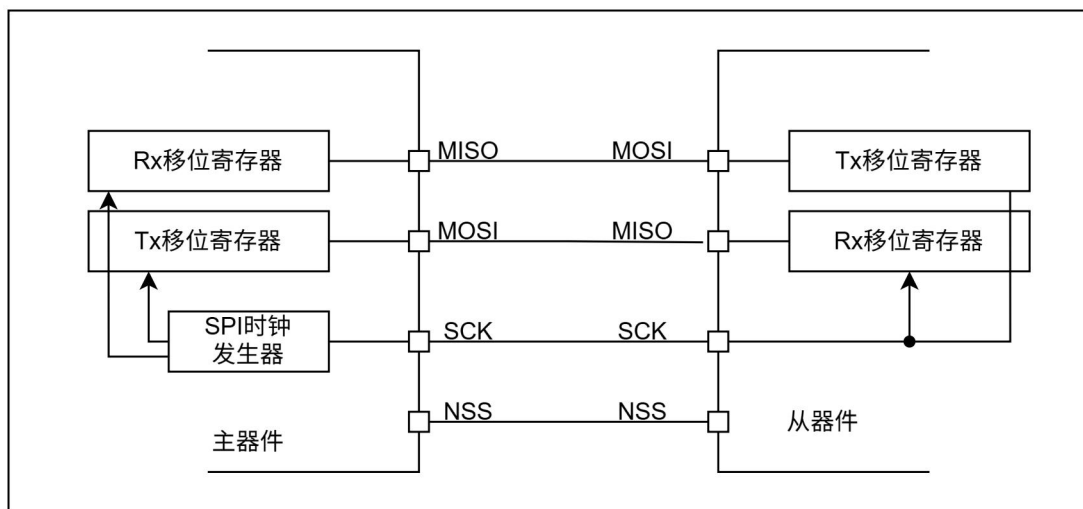


图 17-2 全双工单个主器件/单个从器件应用

1. NSS 引脚可用于在主器件和从器件之间提供硬件控制流。外设也可选择不使用这些引脚。之后，必须在内部为主器件和从器件处理硬件控制流。

#### 单工通信

通过 SPI\_CR2 寄存器中的 RXONLY 位将 SPI 设置为只发送模式或只接收模式，可使 SPI 以单工模式进行通信。在这种配置下，仅使用一条线在主器件和从器件的移位寄存器之间进行传输。MISO 和 MOSI 成对的另一个引脚不用于通信，可用作标准 GPIO。

- 只发送模式 (RXONLY=0): 配置设置与全双工设置相同。应用必须忽略在未使用的输入引脚上捕获的信息。该引脚可以用作标准 GPIO。
- 只接收模式 (RXONLY=1): 应用可通过将 RXONLY 位置 1 来禁止 SPI 输出功能。在从器件配置下，MISO 输出被禁止，该引脚可用作 GPIO。当从器件选择信号有效时，从器件继续从 MOSI 引脚接收数据（请参见 34.5.5: 从器件选择 (NSS) 引脚管理）。基于数据缓冲区的配置产生接收数据事件。在主器件配置下，MOSI 输出被禁止，该引脚可用作 GPIO。只要 SPI 处于使能状态，便不断生成时钟信号。停止时钟的唯一方式是将 RXONLY 位或 SPE 位清零，直至来自 MISO 引脚的传入模式结束，然后基于相应配置填充数据缓冲区结构。

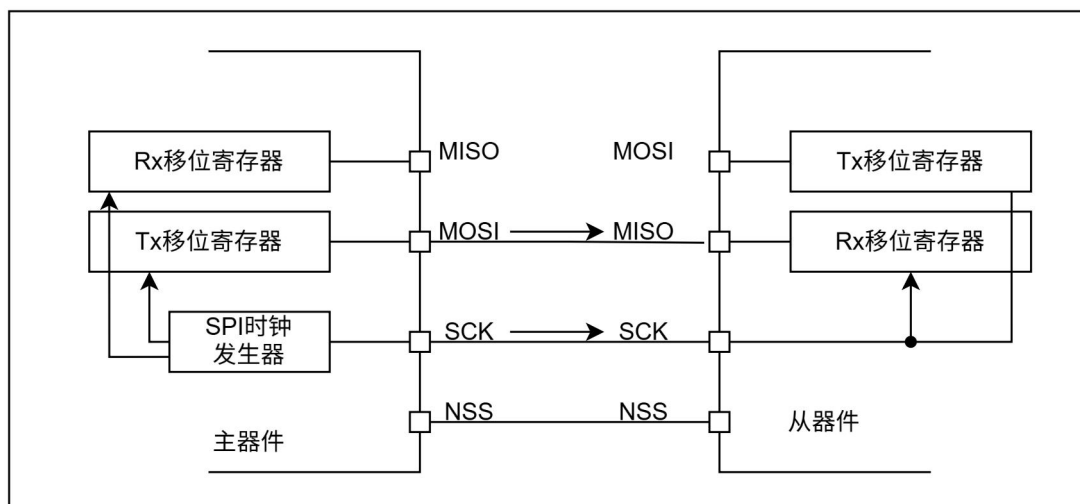


图 17-3 单工单个主器件/单个从器件应用（主器件为只发送模式/从器件为只接收模式）

1. NSS 引脚可用于在主器件和从器件之间提供硬件控制流。外设也可选择不使用这些引脚。之后，必须在内部为主器件和从器件处理硬件控制流。有关更多详细信息，请参见第 34.5.5 节：从器件选择 (NSS) 引脚管理。
2. 在发送器 Rx 移位寄存器的输入上捕获意外输入信息。标准只发送模式下必须忽略与发送器接收流相关的所有事件（例如 OVF 标志）。
3. 在这种配置下，两个 MISO 引脚均可用作 GPIO。



### 17.3.1.3 标准多从器件通信

在具有两个或多个独立从器件的配置下，主器件使用 GPIO 引脚来管理每个从器件的片选线（请参见图 360.）。主器件必须通过拉低与从器件 NSS 输入相连的 GPIO 的电平来单独选择一个从器件。执行该操作后，便建立了标准主器件与专用从器件之间的通信。

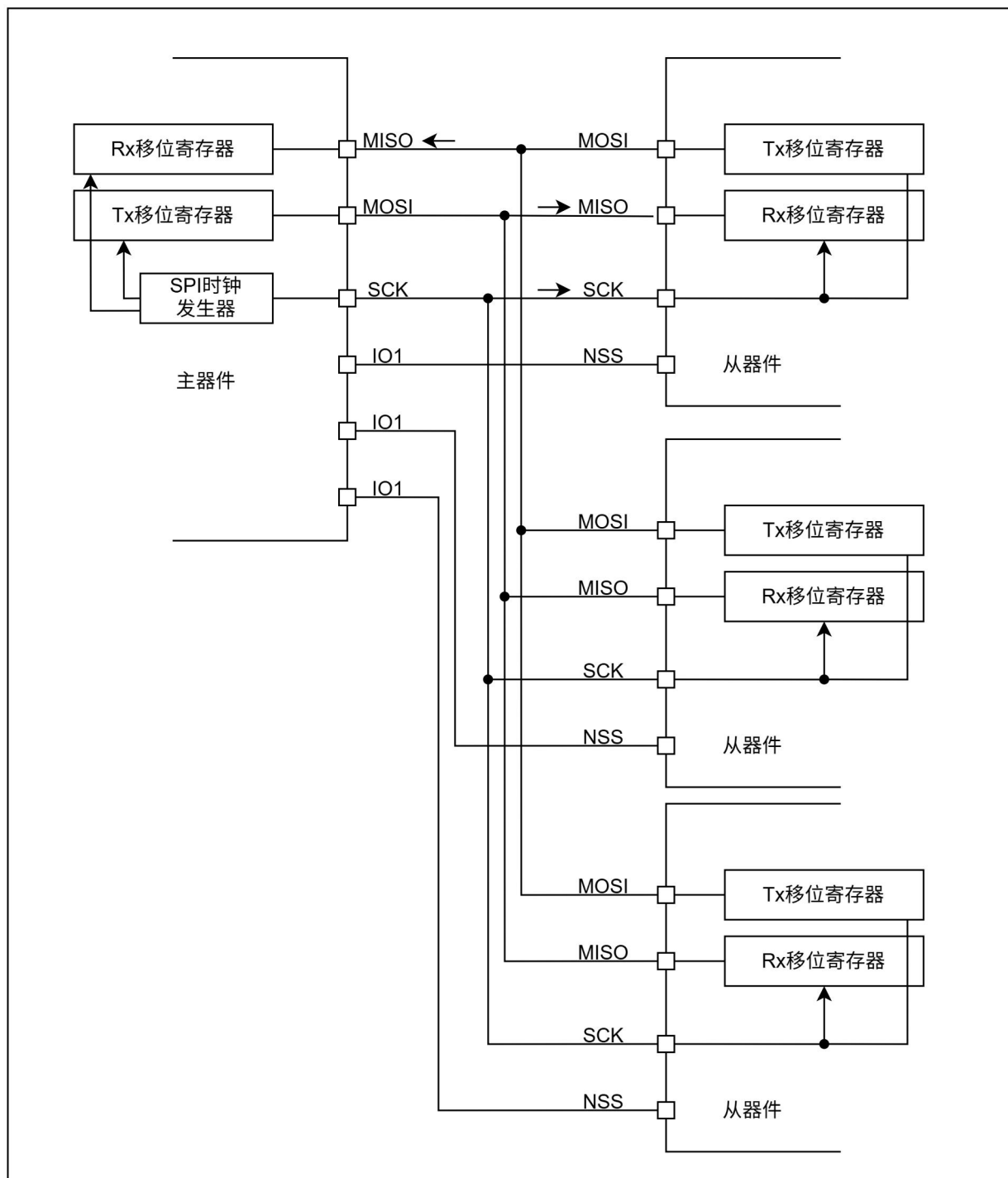


图 17-4 主器件和三个独立的从器件

1. 此配置的主器件侧不使用 NSS 引脚。该引脚必须在内部管理 ( $SSM = 1$ ,  $SSI = 1$ ) 以避免任何 MODF 错误。
2. 由于从器件的 MISO 引脚连在一起，所有从器件 MISO 引脚的 GPIO 配置必须设置为开漏复用

功能。

### 17.3.1.4 多主通信

如果 SPI 总线未用于多主功能,用户可使用内置功能来检测试图同时控制总线的两个节点间是否存在潜在冲突。对于该检测, NSS 引脚配置为硬件输入模式。

由于此时只有一个结点可将其输出施加到公用数据线上, 因此该模式连接的 SPI 节点不能超过两个。

当节点无效时, 默认情况下均保持从模式。一旦一个节点要接管对总线的控制, 它会将自身切换到主模式, 然后通过专用 GPIO 引脚向其他节点的从器件选择输入施加有效电平。会话完成后, 有效的从器件选择信号将被释放, 控制总线的节点会短暂切换回被动从模式, 等待下一个会话开始。

如果两个节点同时发出各自的控制请求, 则会出现总线冲突 (请参见模式故障 MODF 事件)。随后, 用户可应用某个简单的仲裁过程 (例如, 在两个节点上施加不同的预定义超时来推迟下一次尝试)。

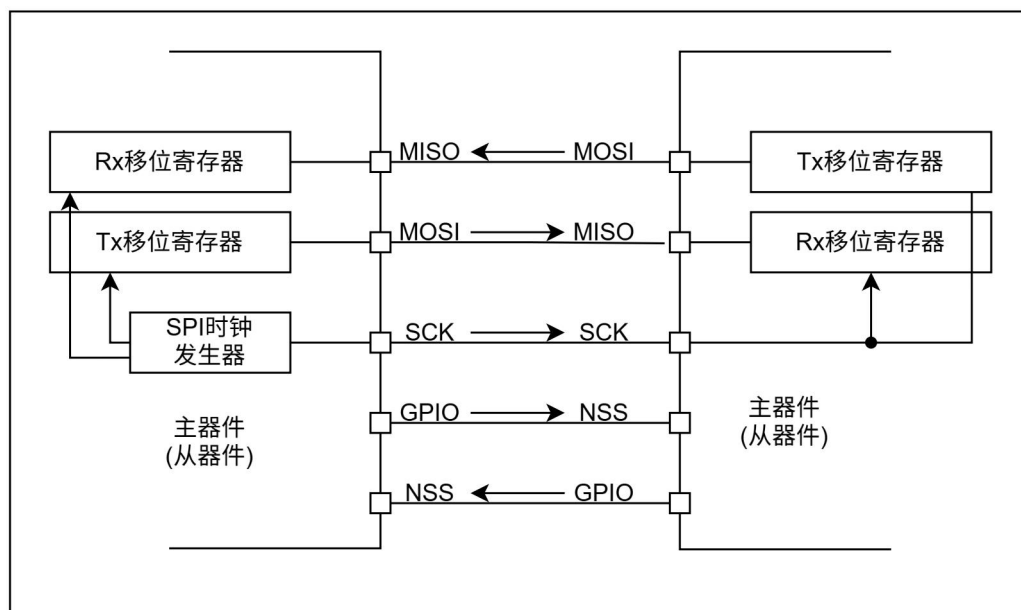


图 17-5 多主应用

1. 在两个节点上, NSS 引脚配置为硬件输入模式。当无效节点配置为从器件时, 其有效电平将使能 MISO 线输出控制。

### 17.3.1.5 从器件选择 (NSS) 引脚管理

在从模式下, NSS 用作标准的“片选”输入, 使从器件与主器件进行通信。在主模式下, NSS 可用作输出或输入。用作输入时, 可防止多主模式总线冲突; 用作输出时, 可驱动单个从器件的从器件选择信号。

可以使用 SPI\_CR1 寄存器中的 SSM 位设置硬件或软件从器件选择管理:

- 软件 NSS 管理 (SSM = 1): 在这种配置下, 由 SPI\_CR1 寄存器中的 SSI 位的值内部驱动从器件选择信息。外部 NSS 引脚空闲, 可供其他应用使用。
- 硬件 NSS 管理 (SSM = 0): 在这种情况下, 可行的配置有两种: 所用配置取决于 NSS 输出配置 (SPI\_CR1 寄存器中的 SSOE 位)。
  - NSS 输出使能 (SSM=0 且 SSOE = 1): 仅在将 MCU 设置为主器件时才使用该配置。NSS 引脚由硬件管理。只要在主模式下使能 SPI (SPE=1), NSS 信号便会被驱动为低电平, 并且会一直保持低电平状态, 直至关闭 SPI (SPE =0)。如果激活 NSS 脉冲模式 (NSSP=1), 连续通信间会生成脉冲。SPI 无法在采用此 NSS 设置的多主模式配置下工作。
  - NSS 输出关闭 (SSM=0 且 SSOE = 0): 如果微控制器在总线上用作主器件, 此配置可实现多主模式功能。如果在该模式下将 NSS 引脚拉至低电平, SPI 将进入主模式故障状态, 器

---

件将在从模式下自动进行重新配置。在从模式下，NSS 引脚用作标准的“片选”输入，当 NSS 线为低电平时将选择从器件。

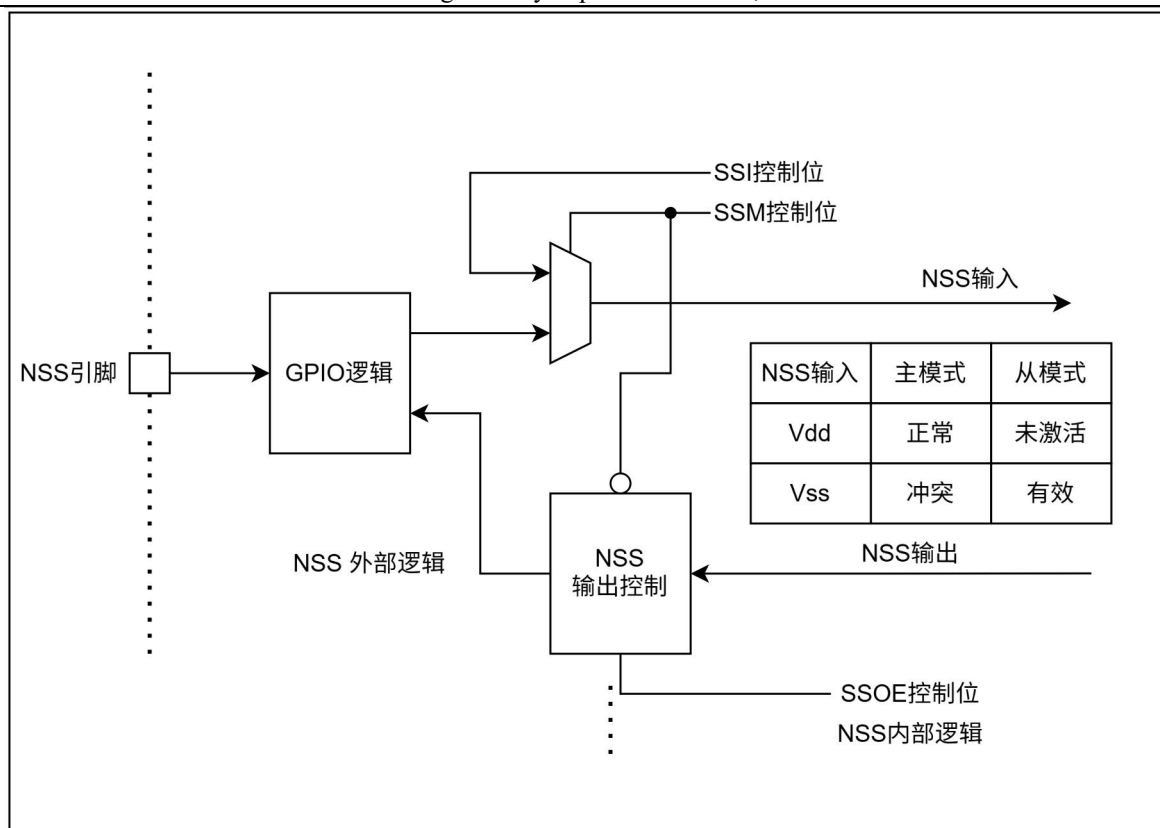


图 17-6 硬件/软件从器件选择管理

### 17.3.1.6 通信格式

SPI 通信过程中，将同时执行接收和发送操作。串行时钟 (SCK) 对数据线上的信息的移位和采样进行同步。通信格式取决于时钟相位、时钟极性和数据帧格式。为了能够在彼此间进行通信，主器件和从器件必须遵循相同的通信格式。

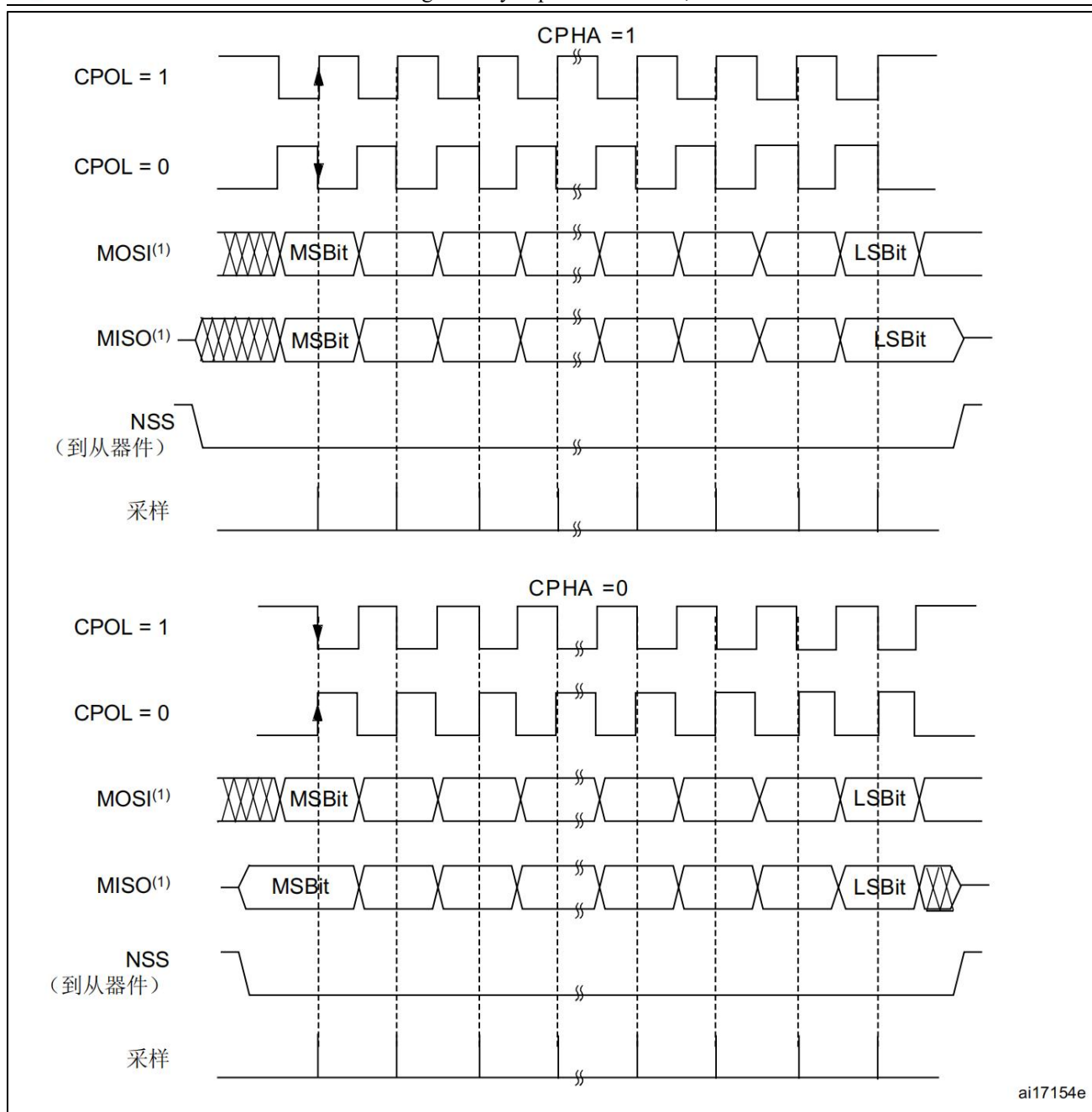
#### 时钟相位和极性控制

通过 SPI\_CR1 寄存器中的 CPOL 和 CPHA 位，可以用软件选择四种可能的时序关系。CPOL（时钟极性）位控制不传输任何数据时的时钟空闲状态值。此位对主器件和从器件都有作用。如果复位 CPOL，SCK 引脚在空闲状态处于低电平。如果将 CPOL 置 1，SCK 引脚在空闲状态处于高电平。

如果将 CPHA 位置 1，则会在 SCK 引脚的第二个边沿捕获传输的第一个数据位（如果复位 CPOL 位，则为下降沿；如果将 CPOL 位置 1，则为上升沿）。即，在每次出现该时钟边沿时锁存数据。如果将 CPHA 位复位，则会在 SCK 引脚的第一个边沿捕获传输的第一个数据位（如果将 CPOL 位置 1，则为下降沿；如果将 CPOL 位复位，则为上升沿）。即，在每次出现该时钟边沿时锁存数据。

CPOL（时钟极性）和 CPHA（时钟相位）位的组合用于选择数据捕获时钟边沿。

下图给出了在 CPHA 和 CPOL 位的四种组合下的 SPI 全双工传输。



ai17154e

图 17-7 数据时钟时序图

1. 数据位的顺序取决于 LSBFIRST 位的设置。

#### 数据帧格式

SPI 移位寄存器可设置为以 MSB 在前或 LSB 在前的方式移出数据，具体取决于 LSBFIRST 位的值。数据帧的长度通过 DS 位进行选择。数据帧的长度可设置为 8 / 16 位，且此设置对于发送和接收均适用。无论所选的数据帧长度为何，必须依照 FRXTH 电平对 FIFO 进行读访问。

### 17.3.1.7 SPI 配置

主器件和从器件的配置步骤几乎相同。对于具体的模式设置，请遵从相应章节的内容。若要对标准通信进行初始化，请执行以下步骤：

1. 写特定的 GPIO 寄存器：配置 MOSI、MISO 和 SCK 信号的 GPIO 引脚。
2. 写 SPI\_CR1 寄存器：
  - a) 通过 BR[2:0] 位配置串行时钟波特率。
  - b) 配置 CPOL 和 CPHA 位的组合，定义数据传输和串行时钟之间的关系（四种关系中的一种）（NSSP 模式下必须将 CPHA 清零）。
  - c) 配置 LSBFIRST 位以定义帧格式。
  - d) 配置 SSM 和 SSI。
  - e) 配置 MSTR 位（在多主模式 NSS 配置下，如果主器件配置为预防 MODF 错误，
  - f) 则应避免 NSS 上出现状态冲突）。
3. 写 SPI\_CR2 寄存器：
  - a) 配置 DS 位，选择传输的数据长度。
  - b) 配置 SSOE。
  - c) 如果在两个数据单元之间需要 NSS 脉冲模式，请将 NSSP 位置 1（NSSP 模式下将 CHPA 位保持清零状态）。
  - d) 配置 FRXTH 位。RXFIFO 阈值必须与 SPI\_DR 寄存器的读访问大小相符。

### 17.3.1.8 使能 SPI 的步骤

建议在主器件发送时钟前使能 SPI 从器件。否则，数据传输可能会不正常。从器件的数据寄存器必须包含待发送的数据才能开始与主器件通信（在通信时钟的第一个边沿；如果时钟信号连续，则是在正在进行的通信结束前）。使能 SPI 从器件前，SCK 信号必须稳定为所选极性对应的空闲状态电平。

当 SPI 处于使能状态，且 TXFIFO 不为空或者对 TXFIFO 执行下一次写操作时，全双工模式（或任何只发送模式）下的主器件开始通信。

在任何主器件只接收模式（RXONLY=1）下，使能 SPI 后，主器件立即开始通信且时钟立即开始运行。

### 17.3.1.9 数据发送和接收过程

#### RXFIFO 和 TXFIFO

所有 SPI 数据交互均经由 32 位内置 FIFO。这使得 SPI 能够以连续流工作，并能防止在数据帧长度较短时发生上溢。每个方向都有其自身的 FIFO，称为 TXFIFO 和 RXFIFO。这些 FIFO 可在所有 SPI 模式下使用。

FIFO 的处理取决于数据交换模式（双工和单工）、数据帧格式（帧中的位数）、FIFO 数据寄存器中的访问大小（8 位或 16 位）以及访问 FIFO 时是否使用数据封包。

对 SPI\_DR 寄存器执行读访问时，会返回尚未读取的 RXFIFO 中存储的最早的值。对 SPI\_DR 执行写访问时，会在发送队列结束时将写入的数据存储到 TXFIFO 中。读访问必须始终与 SPI\_CR2 寄存器中的 FRXTH 位配置的 RXFIFO 阈值保持一致 SPI\_DR 寄存器的读访问必须通过 RXNE 事件进行管理。当数据存储到 RXFIFO 中且达到阈值（由 FRXTH 位定义）时会触发该事件。当 RXNE 清零时，RXFIFO 被视为空。同样地，待发送数据帧的写访问通过 TXE 事件进行管理。当 TXFIFO 占用水平小于或等于其容量的一半时会触发该事件。否则，TXE 清零，TXFIFO 被视为已满。通过这种方式，当数据帧格式不超过 8 位时，RXFIFO 最多可存储四个数据帧，而 TXFIFO 最多只能存储三个数据帧。当软件尝试向 TXFIFO 中写入更多 16 位模式的数据时，这种差异能够防止 TXFIFO 中已存储的 3 个 8 位数据帧出现损坏的情况。TXE 和 RXNE 事件可通过轮询方式或者中断方式处理。请参见图 366 到图 369。

如果在 RXFIFO 已满时收到下一个数据，将发生上溢事件。上溢事件可通过轮询方式或中断方式来



处理。

BSY 位被置 1 表示当前正在处理数据帧。当时钟信号连续运行时，在主器件中，BSY 标志在数据帧之间保持置 1 状态，但在从器件中，BSY 标志在数据帧传输之间变为低电平并持续最短的一段时间（一个 SPI 时钟）。

#### 序列处理

可通过一个序列传送一些数据帧从而完成一条消息。使能发送后，序列即开始，只要主器件的 TXFIFO 中存在数据便一直继续。时钟信号由主器件持续提供，直至 TXFIFO 变为空，之后时钟信号停止，等待其他数据。

在只接收模式或单工模式（RXONLY=1）下，当使能 SPI 并激活只接收模式后，主器件将立即启动序列。时钟信号由主器件提供，且仅当主器件关闭 SPI 或者关闭只接收模式时，时钟信号才会停止。在此之前，主器件会连续接收数据帧。

当主器件能够以连续模式（SCK 信号连续）提供所有交互时，任何时候都必须根据从器件功能来处理数据流及其内容。必要时，主器件必须降低通信速度，提供较慢的时钟或带有足够延时的单独帧或数据段。请注意，SPI 模式下不存在主器件或从器件的下溢错误信号，来自从器件的数据始终由主器件处理，即使从器件无法及时正确地准备数据也是如此。

在多从器件系统中，每个序列必须通过 NSS 脉冲进行控制，从而只选择其中一个从器件进行通信。在单个从器件系统中，无需通过 NSS 来控制从器件，但此时提供此脉冲通常会更好，以在每个数据序列开始时同步从器件。NSS 可通过软件和硬件进行管理（请参见第 34.5.5 节：从器件选择（NSS）引脚管理）。

当 BSY 位置 1 时，表示正在处理数据帧事务。当所进行的帧交互完成时，RXNE 标志将置 1。

最后一位采样后，整个数据帧会存储到 RXFIFO 中。

#### 关闭 SPI 的步骤

当关闭 SPI 时，必须按照本段中介绍的关闭步骤进行操作。当外设时钟停止时，在系统进入低功耗模式前做到这一点是十分重要的。否则会损坏正在进行的交互。在某些模式下，禁止步骤是停止所进行的连续通信的唯一方式。

当处于全双工或只发送模式下的主器件停止提供待发送的数据时，可结束任何事务。在这种情况下，时钟在最后一个数据传输后停止。当处理奇数数量的数据帧时，必须特别注意封装模式以防止交换一些空字节（请参见数据封包部分）。在这些模式下禁止 SPI 之前，用户必须按照标准的禁止步骤进行操作。SPI 在主模式传输时，如果在数据帧处理的过程中，或者 TXFIFO 中有待传输的数据时，此时关闭 SPI，则 SPI 的状态是不可预测的。

只要主器件处于只接收模式，停止连续时钟的唯一方式就是通过 SPE=0 来关闭外设。这必须在最后一个数据帧传输内的特定时间段，即第一位采样与最后一位传输开始之间完成（以便接收全部数量的预期数据帧并防止在最后一个有效数据帧后读取任何其他“空”数据）。在该模式下关闭 SPI 时必须遵循特定步骤。

关闭 SPI 后，已接收但未读取的数据始终存储在 RXFIFO 中，这些数据必须在下次使能 SPI 后进行处理，然后才能启动新序列。为防止存在未读取的数据，需确保关闭 SPI 时 RXFIFO 为空，可通过正确的关闭步骤来关闭 SPI，也可以通过控制外设复位专用的特定寄存器以软件复位的方式来初始化所有 SPI 寄存器从而关闭 SPI（请参见 RCC\_APB1RSTR 寄存器中的 SPI1RST 位）。

标准关闭步骤通过轮询 BSY 状态以及 FTLVL[1:0] 来检查发送会话是否完全结束。还可以在必须识别正在处理的传输是否结束的特定情况下完成这种检查，例如：

- 当 NSS 信号由软件管理且主器件必须为从器件提供 NSS 脉冲结束时，或者
- 最后一个数据帧传输仍在外设总线中处理时。

正确的关闭步骤如下（使用只接收模式时除外）：

1. 等待至 FTLVL[1:0] = 00（无需发送更多数据）。
2. 等待至 BSY=0（最后一个数据帧已处理完）。

3. 关闭 SPI ( $SPE = 0$ )。
4. 读取数据，直至  $FRLVL[1:0] = 00$ （读取接收的所有数据）。

某些只接收模式的正确关闭步骤如下：

1. 当最后一个数据帧正在处理时，通过在特定时间窗口内关闭 SPI ( $SPE=0$ ) 来中断接收流。
2. 等待至  $BSY=0$ （最后一个数据帧已处理完）。
3. 读取数据，直至  $FRLVL[1:0] = 00$ （读取接收的所有数据）。

### 通信图

本部分将介绍一些典型的时序图。无论 SPI 事件是通过轮询还是中断进行处理，这些时序图均有效。为简单起见，此处均假设  $LSBFIRST=0$ 、 $CPOL=0$  和  $CPHA=1$ 。

以下带编号的注释对第 1038 页的图 366 到第 1041 页的图 369 均适用。

1. 激活 NSS 并使能 SPI 后，从器件开始控制 MISO 线，而当其中一个条件不成立时，从器件将与 MISO 线断开。必须为从器件提供充足的时间，以便在传输开始前准备好主器件专用的数据。在主器件上，只有使能 SPI 后，SPI 外设才会控制 MOSI 和 SCK 信号（偶尔还会控制 NSS 信号）。如果关闭 SPI，SPI 外设会与 GPIO 逻辑断开，因此这些线上的电平只取决于 GPIO 设置。
2. 在主器件上，如果通信（时钟信号）连续， $BSY$  在数据帧之间保持有效。在从器件上， $BSY$  信号在数据帧之间始终会保持至少一个时钟周期的低电平状态。
3. 只有  $TXFIFO$  已满时， $TXE$  信号才会清零。

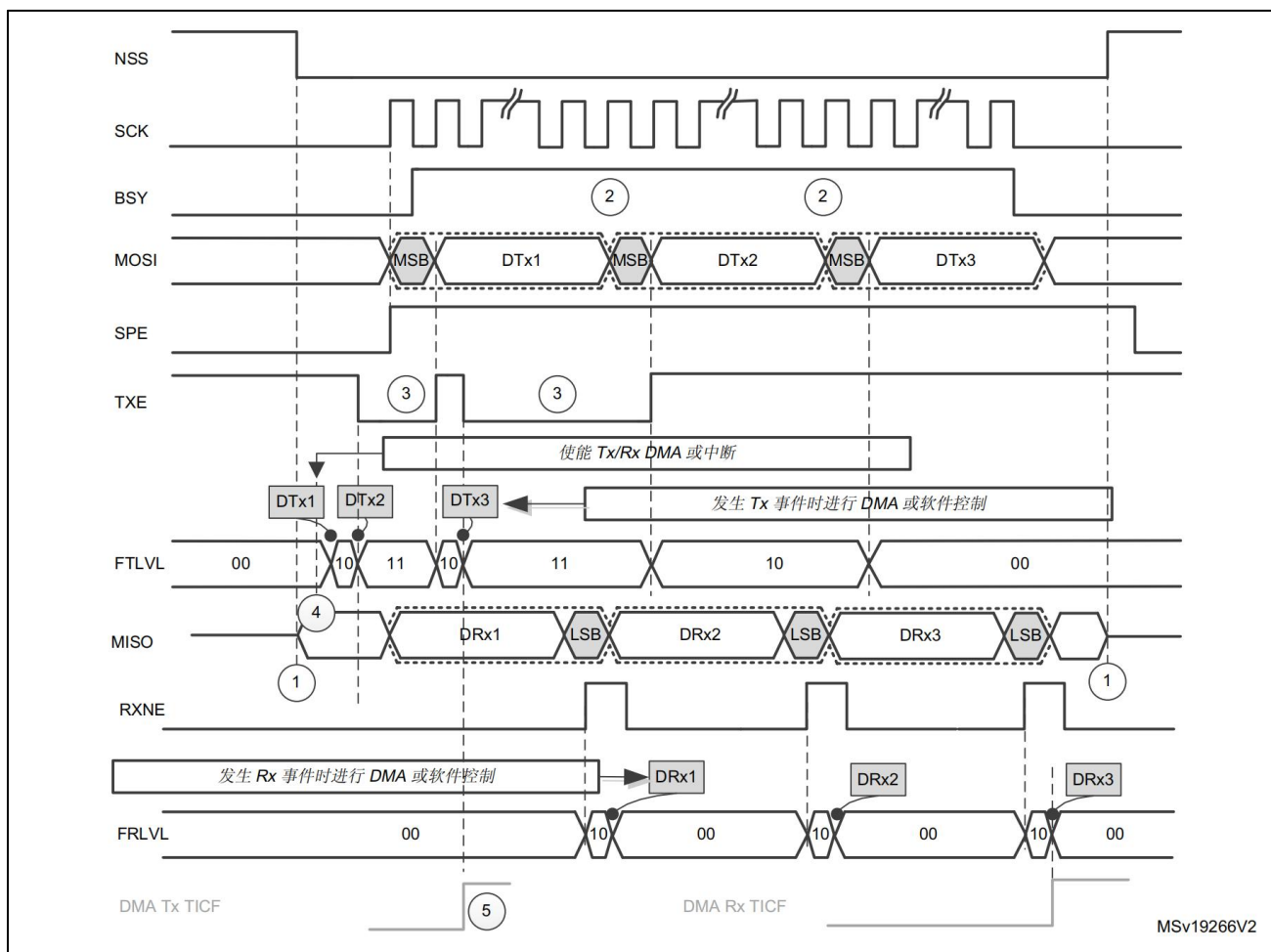


图 17-8 主器件全双工通信

主器件全双工通信示例的假设条件如下：数据大小 > 8 位



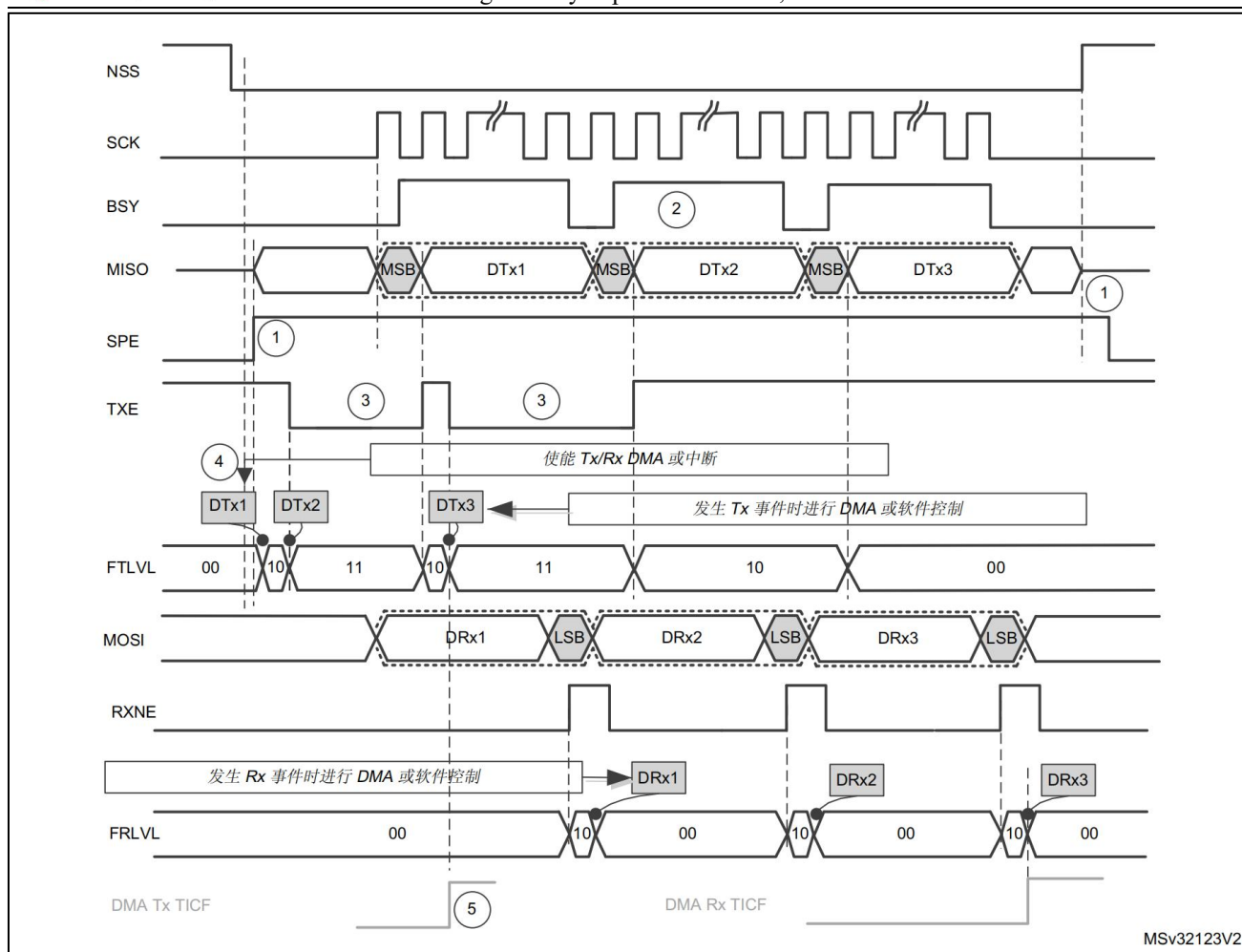


图 17-9 从器件全双工通信

从器件全双工通信示例的假设条件如下：数据大小 > 8 位

### 17.3.1.10 SPI 状态标志

应用可通过三种状态标志监视 SPI 总线的状态。

发送缓冲区为空 (TXE)

当发送 TXFIFO 有足够的空间来存储要发送的数据时，TXE 标志将置 1。TXE 标志与 TXFIFO 占用水平相关。该标志变为高电平后将一直保持高电平状态，直至 TXFIFO 占用水平小于或等于 FIFO 深度的 1/2。如果 SPI\_CR2 寄存器中的 TXEIE 位置 1，可产生中断。

当 TXFIFO 占用水平超过 1/2 时，该位将自动清零。

接收缓冲区非空 (RXNE)

RXNE 标志根据 SPI\_CR2 寄存器中 FRXTH 位的值进行设置：

- 如果 FRXTH 置 1，RXNE 将变为高电平并一直保持高电平状态，直至 RXFIFO 占用水平大于或等于 1/4 (8 位)。
- 如果 FRXTH 清零，RXNE 将变为高电平并一直保持高电平状态，直至 RXFIFO 占用水平大于或等于 1/2 (16 位)。

如果 SPI\_CR2 寄存器中的 RXNEIE 位置 1，可产生中断。

当上述条件不再为真时，RXNE 将由硬件自动清零。

忙标志 (BSY)

BSY 标志由硬件置 1 和清零（写入此标志没有任何作用）。

当 BSY 置 1 时，表示 SPI 上正在进行数据传输（SPI 总线繁忙）。

某些模式下可以使用 BSY 标志来检测传输是否结束，以便软件在进入低功耗模式（该模式下不提供外设时钟）前关闭 SPI 或其外设时钟。这可避免破坏最后一个数据的传输。

BSY 标志还可用于避免在多主模式系统中发生写冲突。

在以下任意一种条件下，BSY 标志将清零：

- 正确关闭 SPI 时
- 在主模式下检测到故障时（MODF 位置 1）
- 在主模式下，完成了数据发送并且不准备发送任何新数据时
- 在从模式下，BSY 标志在各传输之间的至少一个 SPI 时钟周期内置为“0”时。

注：当主器件可以立即处理下一次发送时（例如，如果主器件处于只接收模式或其发送 FIFO 不为空），在主器件侧的传输之间，通信连续且 BSY 标志始终置“1”。尽管从器件并非如此，但建议始终使用 TXE 和 RXNE 标志（而非 BSY 标志）来处理数据发送或接收操作。

### 17.3.1.11 SPI 错误标志

如果以下其中一个错误标志置 1 且已通过将 ERRIE 位置 1 使能了中断，则将生成 SPI 中断。上溢标志 (OVR)

当主器件或从器件接收了数据但 RXFIFO 没有足够的空间来存储接收的数据时，将出现上溢的情况，当出现上溢的情况时，新接收的值不会覆盖 RXFIFO 中之前的值。新接收的值将被丢弃，之后发送的所有数据都将丢失。要将 OVR 位清零，应首先对 SPI\_DR 寄存器执行读访问，然后再对 SPI\_SR 寄存器执行读访问。

模式故障 (MODF)

当主器件的内部 NSS 信号（NSS 硬件模式下为 NSS 引脚，NSS 软件模式下为 SSI 位）被拉低时，将发生模式故障。这会自动将 MODF 位置 1。主模式故障会在以下几方面影响 SPI 接口：

- 如果 ERRIE 位置 1，MODF 位将置 1，并生成 SPI 中断。
- SPE 位清零。这将关闭器件的所有输出，并关闭 SPI 接口。
- MSTR 位清零，从而强制器件进入从模式。

使用以下软件序列将 MODF 位清零：

1. 在 MODF 位置 1 时，对 SPI\_SR 寄存器执行读或写访问。
2. 然后，对 SPI\_CR1 寄存器执行写操作。

为避免包含多个 MCU 的系统中发生多从模式冲突，必须在 MODF 位清零序列期间将 NSS 引脚拉高。在该清零序列后，可以将 SPE 和 MSTR 位恢复到原始状态。安全起见，硬件不允许在 MODF 位置 1 时将 SPE 和 MSTR 位置 1。在从器件中，MODF 位不可置 1，但由前一次多主模式冲突引起时除外。

### 17.3.1.12 NSS 脉冲模式

该模式通过 SPI\_CR2 寄存器中的 NSSP 位来激活，只有将 SPI 接口配置为 Motorola SPI 主模式 (FRF=0) 且在第一个边沿捕捉时，该模式才起作用 (SPI\_CR1 CPHA = 0, CPOL 设置忽略)。激活后，当 NSS 至少保持一个时钟周期的高电平状态时，两个连续的数据帧传输间将生成 NSS 脉冲。该模式下，从器件可以锁存数据。NSSP 脉冲模式旨在用于具有

一个主器件-从器件对的应用。

下图给出了使能 NSSP 脉冲模式后的 NSS 引脚管理情况。

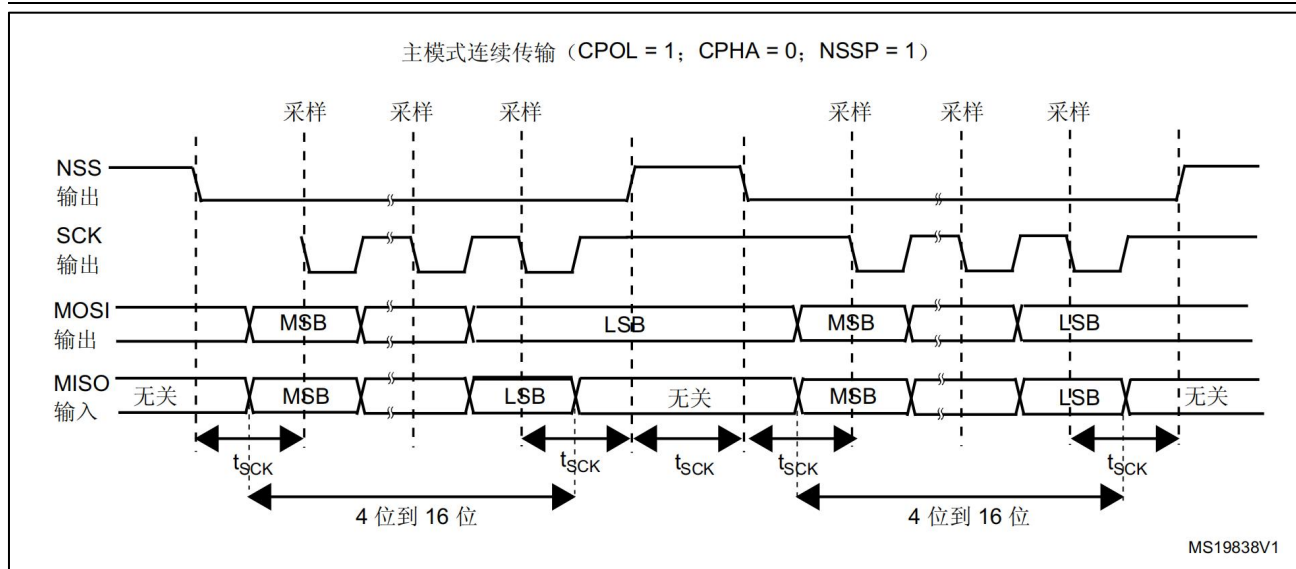


图 17-10 Motorola SPI 主模式下的 NSSP 脉冲生成

注：当 CPOL = 0 时会发生类似行为。在这种情况下，采样边沿为 SCK 的上升沿，NSS 的使能和关闭均参考该采样边沿。

### 17.3.2 SPI 中断

在 SPI 通信过程中，中断可由以下事件产生：

- 发送 TXFIFO 准备被加载
- 接收 RXFIFO 中接收了数据
- 主模式故障
- 上溢错误

中断可分别进行使能和关闭。

表 17-1 SPI 中断请求

中断事件	事件标志	使能控制位
发送 TXFIFO 准备被加载	TXE	TXEIE
RXFIFO 中接收了数据	RXNE	RXNEIE
主模式故障	MODF	ERRIE
上溢错误	OVR	

## 17.4 寄存器列表

SPI 基地址: SPI\_BASE = 0x4001 2000

表 17-2 SPI 寄存器列表

寄存器名称	寄存器地址	寄存器描述
SPI_CR1	0x00	SPI 控制寄存器 1
SPI_CR2	0x04	SPI 控制寄存器 2
SPI_SR	0x08	SPI 状态寄存器
SPI_DR	0x0C	SPI 数据寄存器

## 17.5 寄存器描述

### 17.5.1 SPI 控制寄存器 1 (SPI\_CR1)

偏移地址: 0x00 复位值: 0x0000

位编号	位符号	权限	说明
31-12	RFU	-	保留位, 请保持默认值
11	DS	RW	数据大小 这些位用于配置 SPI 传输的数据长度。 0: 8 位 1: 16 位
10	RXONLY	RW	只接收模式使能 该位用于选择单工模式。此位也适用于多从模式系统, 在此类系统中, 不会访问特定从器件, 也不会损坏访问的从器件的输出。 0: 全双工 (发送和接收) 1: 禁止输出 (只接收模式)
9	SSM	RW	软件从器件管理 当 SSM 位置 1 时, NSS 引脚输入替换为 SSI 位的值。 0: 禁止软件从器件管理 1: 使能软件从器件管理
8	SSI	RW	内部从器件选择 仅当 SSM 位置 1 时, 此位才有效。此位的值将作用到 NSS 引脚上, 并忽略 NSS 引脚的 I/O 值。
7	LSBFIRST	RW	帧格式 0: 发送/接收数据时 MSB 在前 1: 发送/接收数据时 LSB 在前 注: 1. 正在通信时不应更改此位。
6	SPE	RW	SPI 使能 0: 禁止外设 1: 使能外设
5-3	BR	RW	波特率控制 000: $f_{\text{CLK}}/8$

			001: $f_{PCLK}/8$ 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: 正在通信时不应更改这些位。
2	MSTR	RW	主模式选择 0: 从配置 1: 主配置 注: 正在通信时不应更改此位。
1	CPOL	RW	时钟极性 0: 空闲状态时, CK 为 0 1: 空闲状态时, CK 为 1 注: 正在通信时不应更改此位。
0	CPHA	RW	时钟相位 0: 从第一个时钟边沿开始采样数据 1: 从第二个时钟边沿开始采样数据 注: 正在通信时不应更改此位。

## 17.5.2 SPI 控制寄存器 2 (SPI\_CR2)

偏移地址: 0x04      复位值: 0x0000

位编号	位符号	权限	说明
31-15	RFU	-	保留位, 请保持默认值
12	FRXTH	RW	FIFO 接收阈值 该位用于设置触发 RXNE 事件的 RXFIFO 阈值 0: 如果 FIFO 占用水平大于或等于 1/2, 将生成 RXNE 事件。 1: 如果 FIFO 占用水平大于或等于 1/4, 将生成 RXNE 事件。
11-8	RFU	-	保留位, 请保持默认值
7	TXEIE	RW	发送缓冲区空中断使能 0: 屏蔽 TXE 中断 1: 使能 TXE 中断。TXE 标志置 1 时产生中断请求。
6	RXNEIE	RW	接收缓冲区非空中断使能 0: 屏蔽 RXNE 中断 1: 使能 RXNE 中断。RXNE 标志置 1 时产生中断请求。
5	ERRIE	RW	错误中断使能 此位用于在出现错误条件 SPI 模式下的 CRCERR、OVR 和 MODF; 0: 屏蔽错误中断 1: 使能错误中断
4	RFU	-	保留位, 请保持默认值
3	NSSP	RW	NSS 脉冲管理

			<p>此位仅用于主模式。连续传输时，该位允许 SPI 在两个连续数据间生成 NSS 脉冲。单次数据传输时，该位强制 NSS 引脚在传输后变为高电平。如果 CPHA =“1”，该位无意义。</p> <p>0：未生成 NSS 脉冲</p> <p>1：生成 NSS 脉冲</p> <p>注：只有在禁止 SPI (SPE=0) 后才能对此位执行写操作。</p>
2	SSOE	RW	<p>SS 输出使能</p> <p>0：在主模式下禁止 SS 输出，SPI 接口可在多主模式配置下工作</p> <p>1：在主模式下以及使能 SPI 接口后使能 SS 输出。SPI 接口不能在多主模式环境下工作。</p>
1-0	RFU	-	保留位，请保持默认值

### 17.5.3 SPI 状态寄存器 (SPI\_SR)

偏移地址：0x08      复位值：0x0002

位编号	位符号	权限	说明
31-13	RFU	-	保留位，请保持默认值
12-11	FTLVL	RW	<p>FIFP 发送级别</p> <p>这些位将由硬件置 1 和清零。</p> <p>00：FIFO 为空</p> <p>01：1/4 FIFO</p> <p>10：1/2 FIFO</p> <p>11：FIFO 为满（当 FIFO 阈值大于 1/2 时即视为满）</p>
10-9	FRLVL	RW	<p>FIFO 接收级别</p> <p>这些位将由硬件置 1 和清零。</p> <p>00：FIFO 为空</p> <p>01：1/4 FIFO</p> <p>10：1/2 FIFO</p> <p>11：FIFO 已满</p>
8	RFU	-	保留位，请保持默认值
7	BSY	RW	<p>忙标志</p> <p>0：SPI 不繁忙</p> <p>1：SPI 忙于通信或者发送缓冲区不为空</p> <p>此标志由硬件置 1 和清零。</p>
6	OVR	RW	<p>上溢标志</p> <p>0：未发生上溢</p> <p>1：发生上溢</p> <p>此标志由硬件置 1，由软件序列复位。</p>
5	MODF	RW	<p>模式故障</p> <p>0：未发生模式故障</p> <p>1：发生模式故障</p> <p>此标志由硬件置 1，由软件序列复位。</p>
4-2	RFU	-	保留位，请保持默认值
1	TXE	RW	发送缓冲区为空

			0: 发送缓冲区非空 1: 发送缓冲区为空
0	RXNE	RW	接收缓冲区非空 0: 接收缓冲区为空 1: 接收缓冲区非空

### 17.5.4 SPI 数据寄存器 (SPI\_DR)

偏移地址: 0x0C      复位值: 0x0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位, 请保持默认值
15-0	DR	RW	数据寄存器 已接收或者要发送的数据 数据寄存器用于连接 Rx 和 Tx FIFO。读取数据寄存器时, 将访问 Rx FIFO; 而写入数据寄存器时, 将访问 Tx FIFO。 注: 数据始终右对齐。写入寄存器时将忽略未使用位, 读取寄存器时会将 未使用位读为 0。 Rx 阈值设置必须始终与当前使用的读访问相符。

## 18 I2C 接口（I2C）

### 18.1 概述

HC32F0653 内部集成 1 个 I2C 控制器，能按照设定的传输速率（标准，快速，高速）将需要发送的数据按照 I2C 规范串行发送到 I2C 总线上，并对通信过程中的状态进行检测，另外还支持多主机通信中的总线冲突和仲裁处理。

### 18.2 主要特性

- 支持主机发送 / 接收，从机发送 / 接收四种工作模式
- 支持时钟延展（时钟同步）和多主机通信冲突仲裁
- 支持标准 (100Kbps)/ 快速 (400Kbps)/ 高速 (1Mbps) 三种工作速率
- 支持 7bit 寻址功能
- 支持 3 个从机地址
- 支持广播地址
- 支持输入信号噪声过滤功能
- 支持中断状态查询功能



## 18.3 协议描述

I2C 总线使用两根信号线（数据线 SDA 和时钟线 SCL）在设备间传输数据。SCL 为单向时钟线，固定由主机驱动。SDA 为双向数据线，在数据传输过程中由收发两端分时驱动。

I2C 总线上可以连接多个设备，所有设备在没有进行数据传输时都处于空闲状态（未寻址从机接收模式），任一设备都可以作为主机发送 START 起始信号来开始数据传输，在 STOP 停止信号出现在总线上之前，总线一直处于被占用状态。

I2C 通信采用主从结构，并由主机发起和结束通信。主机通过发送 START 起始信号来发起通信，之后发送 SLA+W/R 共 8bit 数据（其中，SLA 为 7bit 从机地址，W/R 为读写位），并在第 9 个 SCL 时钟释放 SDA 总线，对应的从机在第 9 个 SCL 时钟占用 SDA 总线并输出 ACK 应答信号，完成从机寻址。此后根据主机发送的第 1 字节的 W/R 位来决定数据通信的发端和收端，发端每发送 1 个字节数据，收端必须回应 1 个 ACK 应答信号。数据传输完成后，主机发送 STOP 信号结束本次通信。

### 18.3.1 协议帧格式

标准 I2C 传输协议帧包含四个部分：起始信号 (START) 或重复起始信号 (Repeated START) 信号，从机地址及读写位，数据传输，停止信号 (STOP)。如下图所示。

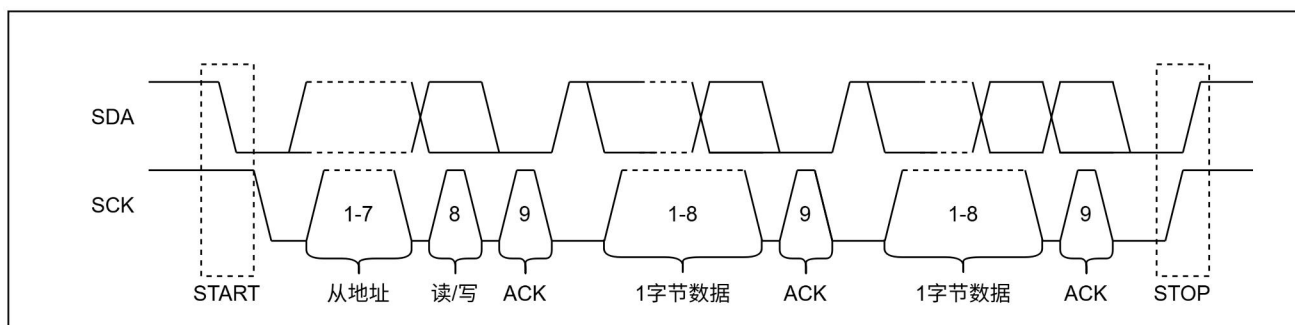


图 18-1 I2C 协议帧

- 起始信号 (START)

当总线处于空闲状态时（SCL 和 SDA 线同时为高），SDA 线上出现由高到低的下降沿信号，则被定义为起始信号。主机向总线发出起始信号后开始数据传输，并占用总线。

- 重复起始信号 (Repeated START)

当一个起始信号后未出现停止信号之前，出现了新的起始信号，新的起始信号被定义为重复起始信号。在主机发送停止信号前，SDA 总线一直处于占用状态，其它主机无法占用总线。

- 停止信号 (STOP)

当 SCL 线为高时，SDA 线上出现由低到高的上升沿信号，则被定义为停止信号。主机向总线发出停止信号以结束数据传输，并释放总线。

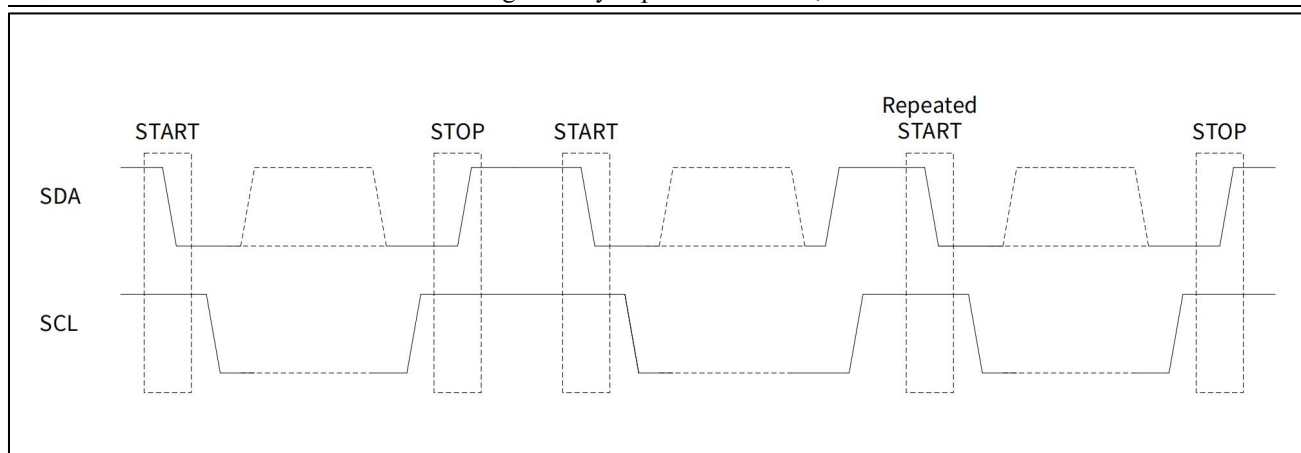


图 18-2 起始和停止信号

- 从机地址及读写位

当起始信号产生后，主机开始传输第 1 字节数据：7 bit 从机地址 + 读写位。读写位（1：读；0：写）控制总线上数据传输方向。被寻址的从机在第 9 个 SCL 时钟周期内占用 SDA 总线，并将 SDA 置为低电平作为 ACK 应答。

- 数据传输

主机在 SCL 线上输出串行时钟信号，主从机通过 SDA 线进行数据传输。数据传输过程中，1 个 SCL 时钟脉冲传输 1 个数据位（最高有效位 MSB 在前），且 SDA 线上的数据只在 SCL 为低时改变，每传输 1 个字节跟随 1 个应答位。如下图所示：

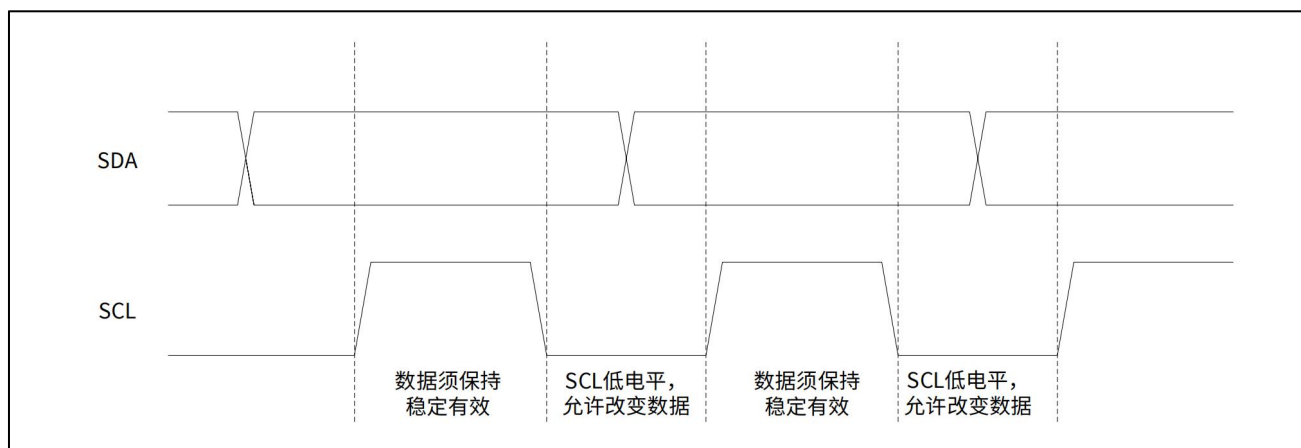


图 18-3 数据传输

### 18.3.2 传输应答

在总线上传输数据时，发端每传输完 1 个字节数据，在第 9 个 SCL 时钟周期发端放弃对 SDA 的控制，收端须在第 9 个 SCL 时钟周期回复 1 个应答位：接收成功，发送 ACK 应答，接收异常发送 NACK 应答。

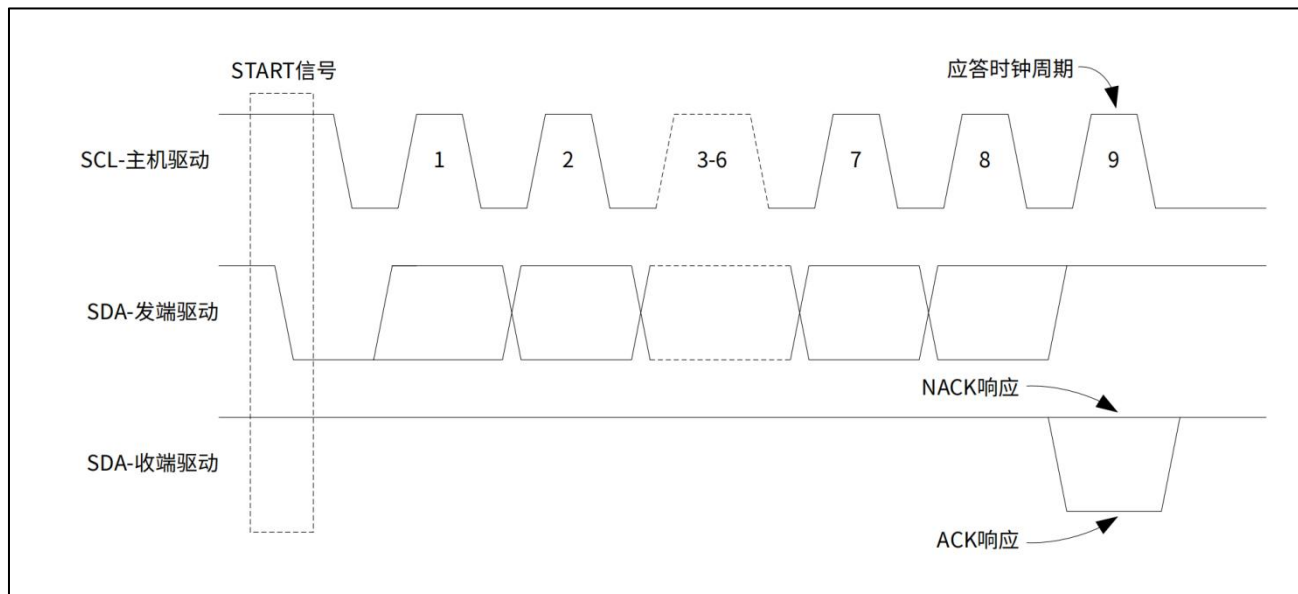


图 18-4 传输应答

### 18.3.3 冲突检测与仲裁

在多主机通信系统中，总线上的每个节点都有从机地址。每个节点可以作为从节点被其它节点访问，也可以作为主节点向其它的节点发送控制字节和传送数据。如果有两个或两个以上的节点同时向总线发出起始信号并开始传输数据，就会造成总线冲突。I2C 控制器内置一个仲裁器，可对 I2C 总线冲突进行检测和仲裁，以保证数据通信的可靠性和完整性。

#### ● 冲突检测原理

在物理实现上，SDA 和 SCL 引脚电路结构相同，引脚的输出驱动与输入缓冲连在一起。输出结构为漏极开路的场效应管、输入结构为高输入阻抗的同相器。基于该结构：

1. 由于 SDA、SCL 为漏极开路结构，借助于外部的上拉电阻实现了信号的“线与”逻辑；
2. 设备向总线写数据的同时读取数据，可用来检测总线冲突，实现“时钟同步”和“总线仲裁”。

根据“线与”逻辑，如果 2 个主机同时发送逻辑 1 或逻辑 0，则 2 个主机都检测不到冲突，需要等到下一位数据发送再继续检测冲突；如果 2 个主机一个发送逻辑 1，一个发送逻辑 0，此时总线上为逻辑 0，发送逻辑 1 的主机检测到冲突，发送逻辑 0 的主机没有检测到冲突。

#### ● 冲突仲裁原理

当主机检测到总线冲突后，该主机丢失仲裁，退出主机发送模式，进入未寻址从机模式，释放 SDA 数据线，并回到地址侦测状态，之后根据接收到的 SLA+W/R 进入相应的从机模式 (SLA 地址匹配进入已寻址从机模式，SLA 地址不匹配则进入未寻址从机模式)。仲裁失败的主机，仍会发送 SCL 串行时钟，直到当前字节传输结束。

当主机没有检测到总线冲突，该主机赢得仲裁，继续主导本次数据传输，直到通信完成。

下图为一个 I2C 总线上 A 主机和 B 主机发送冲突和仲裁示意图（假设 A，B 两个主机的发送时钟同步）：

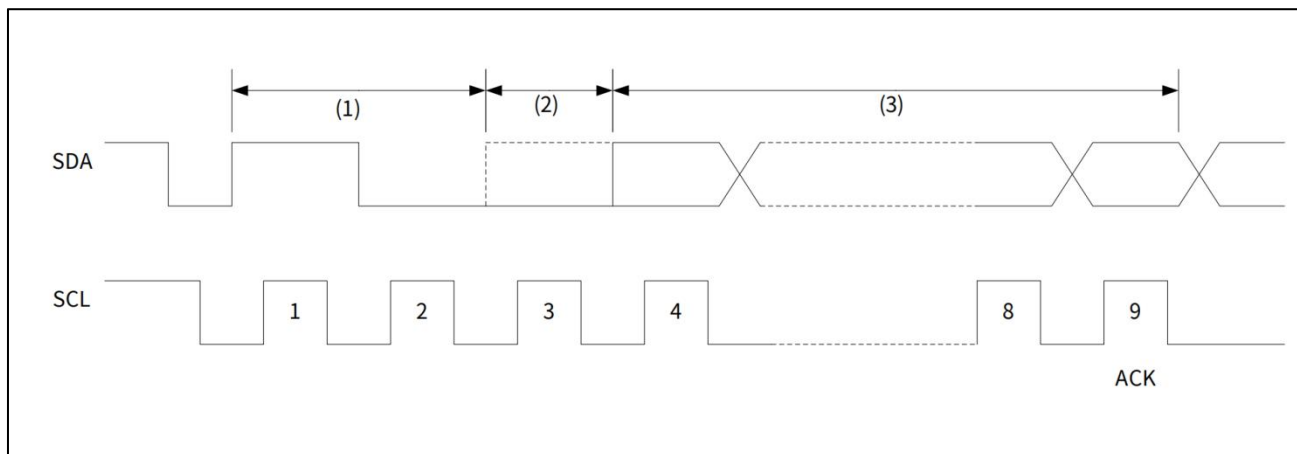


图 18-5 冲突检测及仲裁

1. 在阶段（1），A 主机和 B 主机均先后发送逻辑 1 和逻辑 0，由于数据相同且同步，2 个主机都不会检测到总线冲突。
2. 在阶段（2），A 主机发送逻辑 1，B 主机发送逻辑 0，SDA 总线上数据为逻辑 0。A 主机控制器检测到数据错误，退出发送竞争，即丢失仲裁，A 主机进入未被寻址的从机接收模式。B 主机没有检测到总线冲突，赢得仲裁，继续本次数据传输。
3. 在阶段（3），A 主机处于未被寻址的从机接收模式，但仍产生 SCL 时钟脉冲，直到当前字节传输结束。此后 A 主机 I2C 控制器不再发送时钟信号，B 主机由于赢得仲裁，SCL 和 SDA 都由 B 主机来主导控制传输。

## 18.4 协议描述

### 18.4.1 通信时序和数据格式

I2C 模块主要包括时钟发生器、输入滤波器、地址比较器、协议控制逻辑、仲裁和同步逻辑、以及相关寄存器等。其功能框图如下图所示：

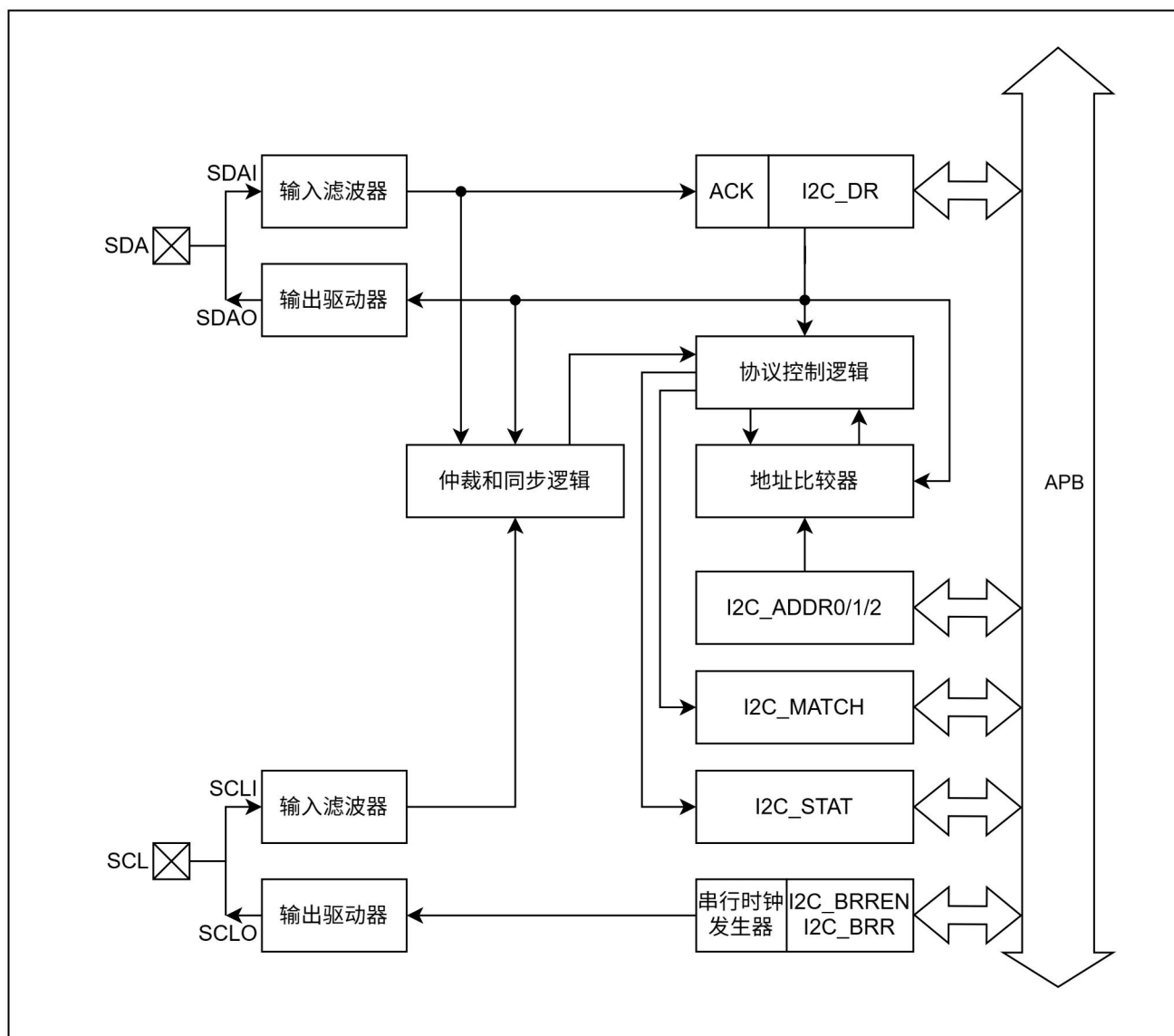


图 18-6 I2C 功能框图

## 18.4.2 通信时序和数据格式

串行时钟发生器用来产生 I2C 通信的波特率时钟 SCL。串行时钟发生器采用 PCLK 作为输入时钟，即  $I2CCLK = PCLK$ ，通过 1 个 8bit 的计数器计数，输出所需波特率的 I2C 时钟信号。

SCL 时钟频率计算公式：

$$f_{SCL} = f_{I2CCLK} / 8 / (BRR + 1)$$

其中，BRR 通过波特率计数器配置寄存器 I2C\_BRR 配置，BRR 有效范围为 1 ~ 255。

串行时钟发生器的计数器计数由 I2C\_BRREN 寄存器的 EN 位域使能，EN 为 1 使能，为 0 禁止。主机时应设置 EN 为 1，从机时该位不影响。

PCLK、BRR 组合和 SCL 时钟频率的对应关系如下表所示：

表 18-1 I2C 传输速率和配置举例

$f_{SCL}(kHz)$ $F_{I2CCLK}(kHz)$	I2C_BRR						
	1	2	3	4	5	6	7
1000	62	41	31	25	20	17	15
2000	125	83	62	50	41	35	31
4000	250	166	125	100	83	71	62
6000	375	250	187	150	124	107	93
8000	500	333	250	200	166	142	125
10000	625	416	312	250	208	178	156
12000	750	500	375	300	250	214	187
14000	875	583	437	350	291	250	218
16000	1000	666	500	400	333	285	250

### 18.4.3 输入滤波器

输入滤波器可对 SDA 和 SCL 输入信号进行滤波处理，小于 1/8/15 个 HCLK 周期的尖峰脉冲信号会被滤除。

### 18.4.4 地址比较器

I2C 总线上各设备都有从机地址，且各从机地址均不同。主机根据从机地址寻址从机，从机通过地址比较器自动检测主机发送的 7bit 寻址地址与本机地址是否匹配，以确定是否与主机通信。

I2C 控制器支持 3 个可编程的从机地址，具体地址信息通过从机地址寄存器 I2C\_ADDR0 / I2C\_ADDR1 / I2C\_ADDR2 进行配置。

从机的地址比较器将接收到的 7bit 寻址地址和 3 个从机地址以及广播地址 (0x00) 相比较。如果符合 4 个地址中的任何一个，则认为地址匹配，同时 I2C 中断标志位 I2C\_CR.SI 会被置 1，并产生一个中断请求。应用程序可通过查询从机地址匹配寄存器 I2C\_MATCH 获取匹配成功的地址序号。

如果地址匹配到 I2C\_ADD0 / 1 / 2，则从机进入相应的已寻址从机接收模式（接收到 SLA+W）或者已寻址从机发送模式（接收到 SLA+R）；如果地址匹配到广播地址 0x00（接收到 SLA+W），则从机进入广播接收模式。

### 18.4.5 仲裁和同步逻辑

#### 18.4.5.1 SCL 同步

I2C 支持时钟同步（时钟延展）功能，SCL 时钟低电平的时间由 SCL 时钟低电平宽度最长的器件决定，而 SCL 时钟高电平时间由时钟高电平宽度最短的器件决定。

如果从机希望主机降低传输速度，可以在收到数据并回应 ACK 后，保持 I2C\_CR.SI 为 1，则从机 I2C 控制器将保持 SCL 为低电平状态，以此来通知主机。当主机准备下一个字节数据传输（发送或者接收）时检测到 SCL 的电平被拉低，则进行等待，直到从机完成操作并将 I2C\_CR.SI 清 0，从机控制器释放 SCL 的拉低控制，主机检测到 SCL 为高电平后继续下一个字节数据的传输。

#### 18.4.5.2 SDA 仲裁

I2C 支持 SDA 冲突检测和仲裁，可以保证在多个主机企图控制 I2C 总线时，I2C 总线上的数据不被破坏。

每个主机发送数据时，都会同时比较总线上的数据与自己发送的数据是否一致，不一致则认为检测到总线冲突，会退出发送竞争，即丢失仲裁。丢失仲裁的主机会立即切换到未被寻址的从机状态，以确保自身能被仲裁成功的主机寻址到。丢失仲裁的主机会继续输出 SCL 串行时钟，直到当前字节传输完成。

SDA 仲裁一般发生在主机发送 SLA+W/R 数据阶段，如果两个主机同时向一个从机发送数据，即两个主机发送的从机地址相同，则仲裁会在第二个字节持续。

## 18.4.6 应答控制

I2C 数据传输的收端必须在每个字节的第 9 个 SCL 时钟周期给发端进行 ACK 或者 NACK 应答，发端通过该应答位来获取收端当前状态：回应 ACK 应答，则表明收端已正确接收该字节数据，可以继续接收下一字节数据；回应 NACK 一般表示收端已不再接收任何数据。

收端发送 ACK 还是 NACK，由收端的 I2C 控制寄存器 I2C\_CR 的 AA 位域来控制。当设置 I2C\_CR.AA 为 1 时，I2C 模块每收到 1 字节数据后会回应 ACK 应答，当设置 I2C\_CR.AA 为 0 时，I2C 模块每收到 1 字节数据后回应 NACK 应答。

在主机接收数据过程中，主机作为通信发起方，控制着收发字节个数，主机（收端）在最后一个字节数据接收完成后回应 NACK 应答给从机（发端），从机收到 NACK 应答后将切换为未寻址从机接收模式，并释放 SDA 总线，以便主机发送 STOP 停止信号或 Repeated START 重复起始信号。

在从机发送数据过程中，如果自身的 I2C\_CR.AA 应答控制位被应用程序清零，则从机在发送完最后 1 字节有效数据后，将自身切换为未寻址从机接收模式，并释放 SDA 总线，主机从总线上读数据将得到 0xFF。此时主机应能判断从机处于无响应状态，并发送 STOP 停止信号或 Repeated START 重复起始信号。

在从机接收数据过程中，如果回应 NACK 给主机（发端），则表示从机（收端）主动结束本次通信，不再接收主机发送的数据，且将自身切换为未寻址从机接收模式，并释放 SDA 总线，此时主机应发送 STOP 停止信号或 Repeated START 重复起始信号。

## 18.4.7 I2C 中断

I2C 控制寄存器 I2C\_CR 的 SI 位域为中断标志位。当 I2C 状态寄存器 I2C\_STAT 的 STAT 位域值发生改变（变成 0xF8 除外）时，I2C\_CR.SI 标志位就会被置位，同时产生中断请求。

在用户 I2C 中断服务程序中，应查询 I2C 状态寄存器 I2C\_STAT 的 STAT 位域值获取 I2C 总线的状态，以确定中断产生原因。设置 I2C\_CR.SI 为 0 清除该标志位。



## 18.4.8 工作模式

I2C 控制器支持 4 种工作模式：主机发送模式、主机接收模式、从机发送模式、从机接收模式。另外还支持广播接收模式，其工作方式和从机接收模式类似。

### 18.4.8.1 主机发送模式

主机发送模式下，主机主动发送多个字节到从机。

SCL 串行时钟由主机控制产生，因此需要先根据传输波特率设置 I2C\_BRR 寄存器并设置 I2C\_BRREN 寄存器的 EN 位域为 1，然后启动传输。

主机设置 I2C\_CR.STA 为 1，通知控制器发送 START 起始信号，控制器收到通知后检测总线是否空闲，当总线空闲时，主机控制器向总线发送一个 START 起始信号。如果发送成功，状态码 I2C\_STAT 变为 0x08，中断标志位 I2C\_CR.SI 被置 1。

主机发送完 START 起始信号后，需要软件设置 I2C\_CR.STA 为 0，然后将从机地址和写标志位 (SLA+W) 写入到 I2C 数据寄存器 I2C\_DR；清零 I2C\_CR.SI 位，主机控制器将发送 SLA+W 到 I2C 总线上；当主机发送完 SLA+W 并收到从机 ACK 应答信号后，状态码 I2C\_STAT 变为 0x18，中断标志位 I2C\_CR.SI 被置 1。

此后主机根据应用需要，发送多个字节的用户自定义数据并检测 ACK 应答信号，每个字节的发送过程和发送 SLA+W 数据帧类似。

主机在发送数据过程中，如果收到 NACK 应答信号，表明从机不再接收主机发送的数据（从机的 I2C\_CR.AA 被清零），主机应发送 STOP 信号结束本次数据传输，或者发送 Repeated START 重复起始信号开启新一轮传输。

当主机发送完成所有数据后，设置 I2C\_CR.STO 为 1，通知控制器数据已发送完成，待发送 STOP 停止信号。清零 I2C\_CR.SI 位，主机控制器将发送 STOP 停止信号到 I2C 总线上，完成本次数据传输，释放总线。

当主机发送完成所有数据后，也可以不发送 STOP 停止信号，而是直接发送 Repeated START 重复起始信号，继续占用总线进行新一轮数据传输。

当主机由于总线冲突丢失仲裁时，会进入未寻址从机接收模式（状态码 I2C\_STAT = 0x38）。

主机发送模式下状态同步图如下图所示：

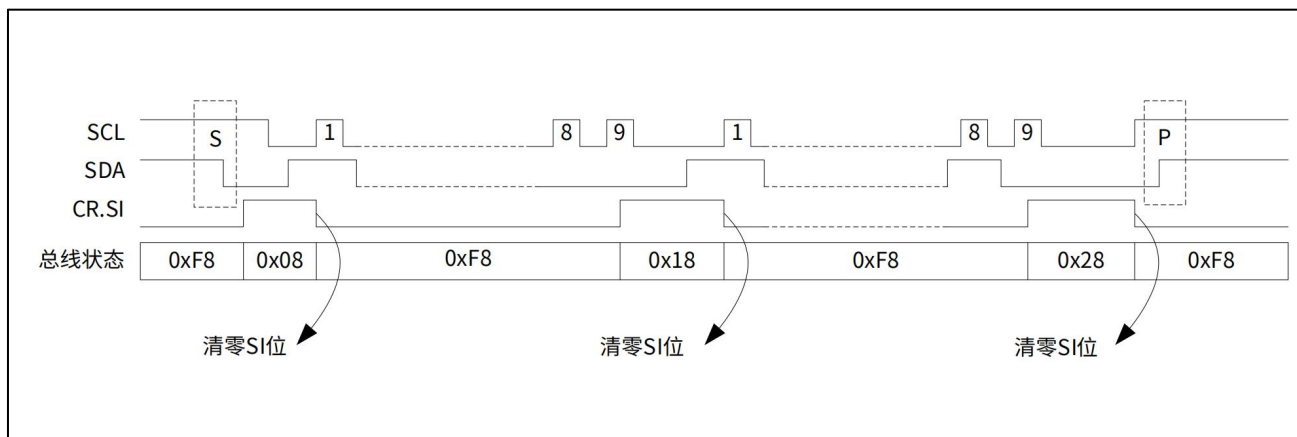


图 18-7 主机发送模式状态同步图

主机发送模式下数据发送流程图及状态寄存器值如下图所示：

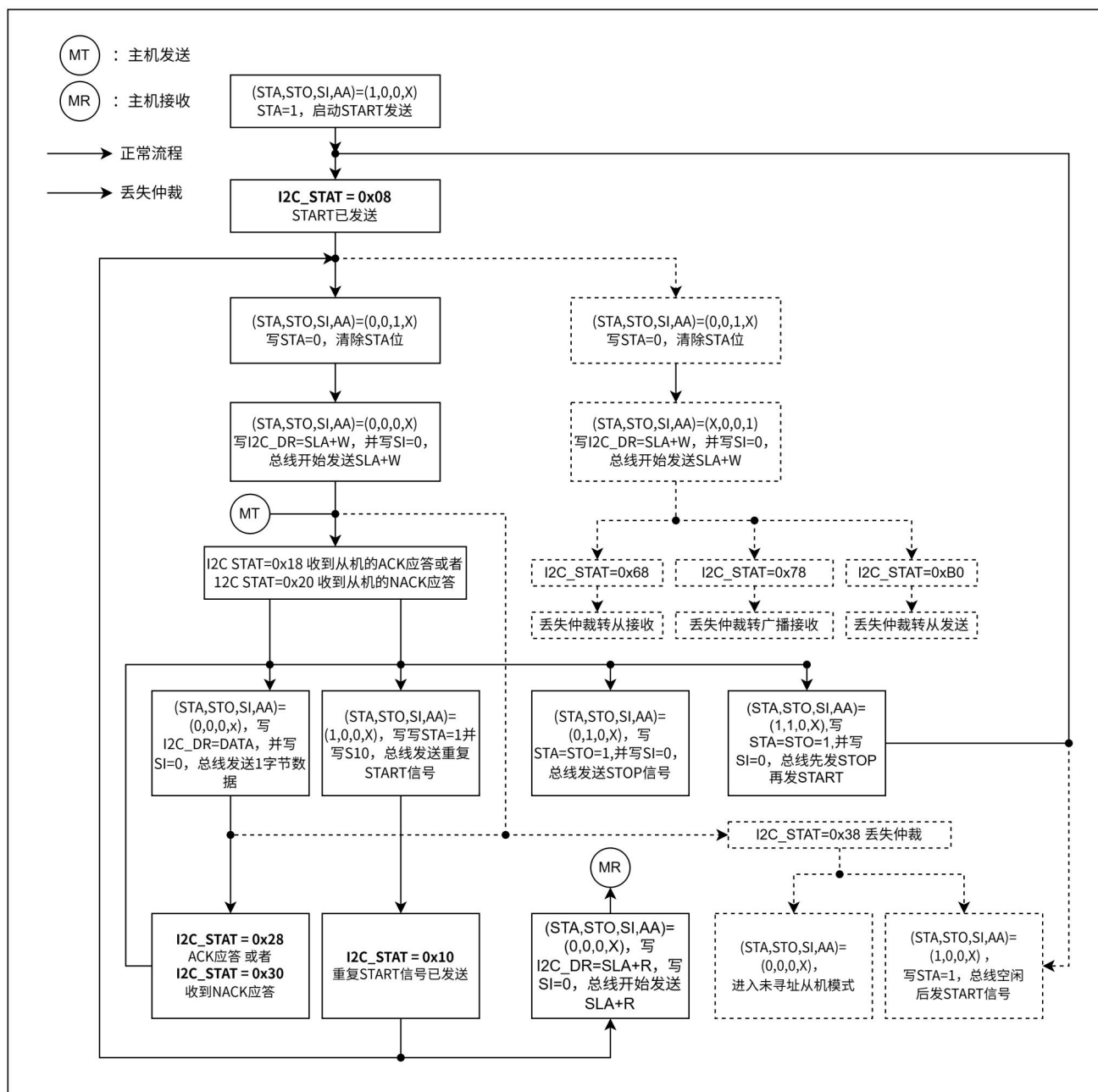


图 18-8 主机发送模式流程和寄存器状态

### 18.4.8.2 主机接收模式

主机接收模式用于接收从机发送的多个数据，主机每接收到 1 字节数据后会回应 ACK 应答信号。SCL 串行时钟由主机控制产生，因此需要先根据传输波特率设置主机的 I2C\_BRR 寄存器并设置 I2C\_BRREN 寄存器的 EN 位域为 1，然后启动传输。

主机设置 I2C\_CR.STA 为 1，通知控制器发送 START 起始信号，控制器收到通知后检测总线是否空闲，当总线空闲时，主机控制器向总线发送一个 START 起始信号。如果发送成功，状态码 I2C\_STAT 变为 0x08，中断标志位 I2C\_CR.SI 被置 1。

主机发送完 START 起始信号后，需要软件设置 I2C\_CR.STA 为 0，然后将从机地址和读标志位 (SLA+R) 写入到 I2C 数据寄存器 I2C\_DR；清零 I2C\_CR.SI 标志位，主机控制器将发送 SLA+R 到 I2C 总线上；当主机发送完 SLA+R 并收到从机 ACK 应答信号后，状态码 I2C\_STAT 变为 0x40，中断标志位 I2C\_CR.SI 被置 1。

此后，主机设置 I2C\_CR.AA 为 1，并清零 I2C\_CR.SI 位，开始接收从机发送的数据，每接收完 1 字节数据后，都要回复一个 ACK 应答信号。在主机接收过程中，应注意：

1. 为保证每接收到 1 字节数据后都能正确产生 I2C\_CR.SI 中断信号，需要在收到 1 字节数据后及时将 I2C\_CR.SI 位清除。
2. 在接收最后一个字节前需要将 I2C\_CR.AA 清零，即主机在接收到最后一个字节时不产生 ACK 应答信号，以此来通知从机停止数据发送。

主机在接收数据过程中，如果从机由于某种原因不再发送主机所需要的数据（从机的 I2C\_CR.AA 被清零），主机后续将收到全 1 信号，此时主机需要在应用层对数据进行判决，判定从机为无响应状态，应发送 STOP 停止信号来结束本次传输，或发送 Repeated START 重复起始信号开启新一轮传输。

当主机接收完成所有数据后，设置 I2C\_CR.STO 为 1，通知控制器数据已接收完成，待发送 STOP 停止信号。清零 I2C\_CR.SI 位，主机控制器发送 STOP 停止信号到 I2C 总线上，完成本次数据传输，释放总线。

当主机接收完成所有数据后，也可以不发送 STOP 停止信号，而是直接发送 Repeated START 重复起始信号，继续占用总线进行新一轮数据传输。

当主机由于总线冲突丢失仲裁时，会进入未寻址从机接收模式（状态码 I2C\_STAT = 0x38）。

主机接收模式下状态同步图如下图所示：

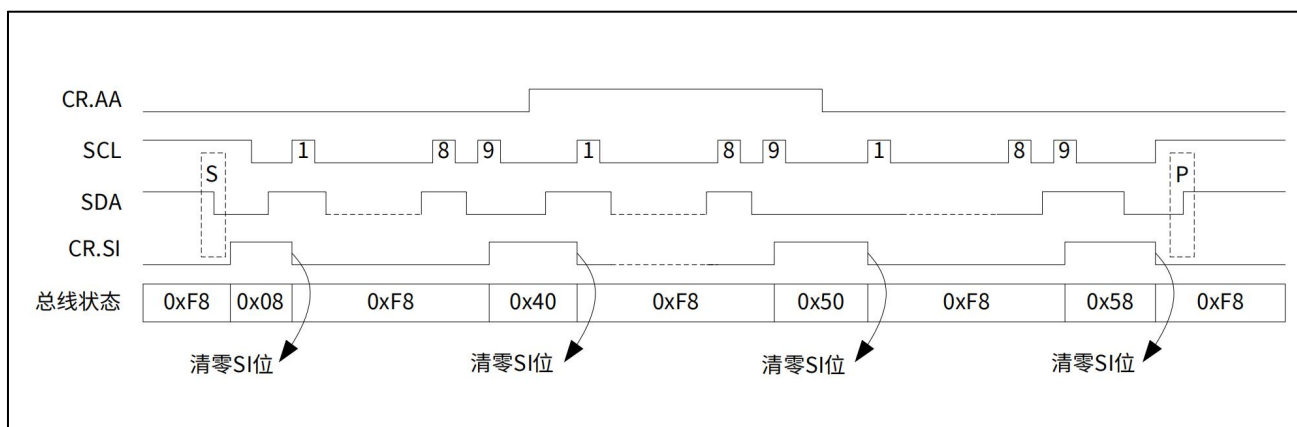


图 18-9 主机接收模式状态同步图

主机接收模式下数据接收流程图及状态寄存器值如下图所示：

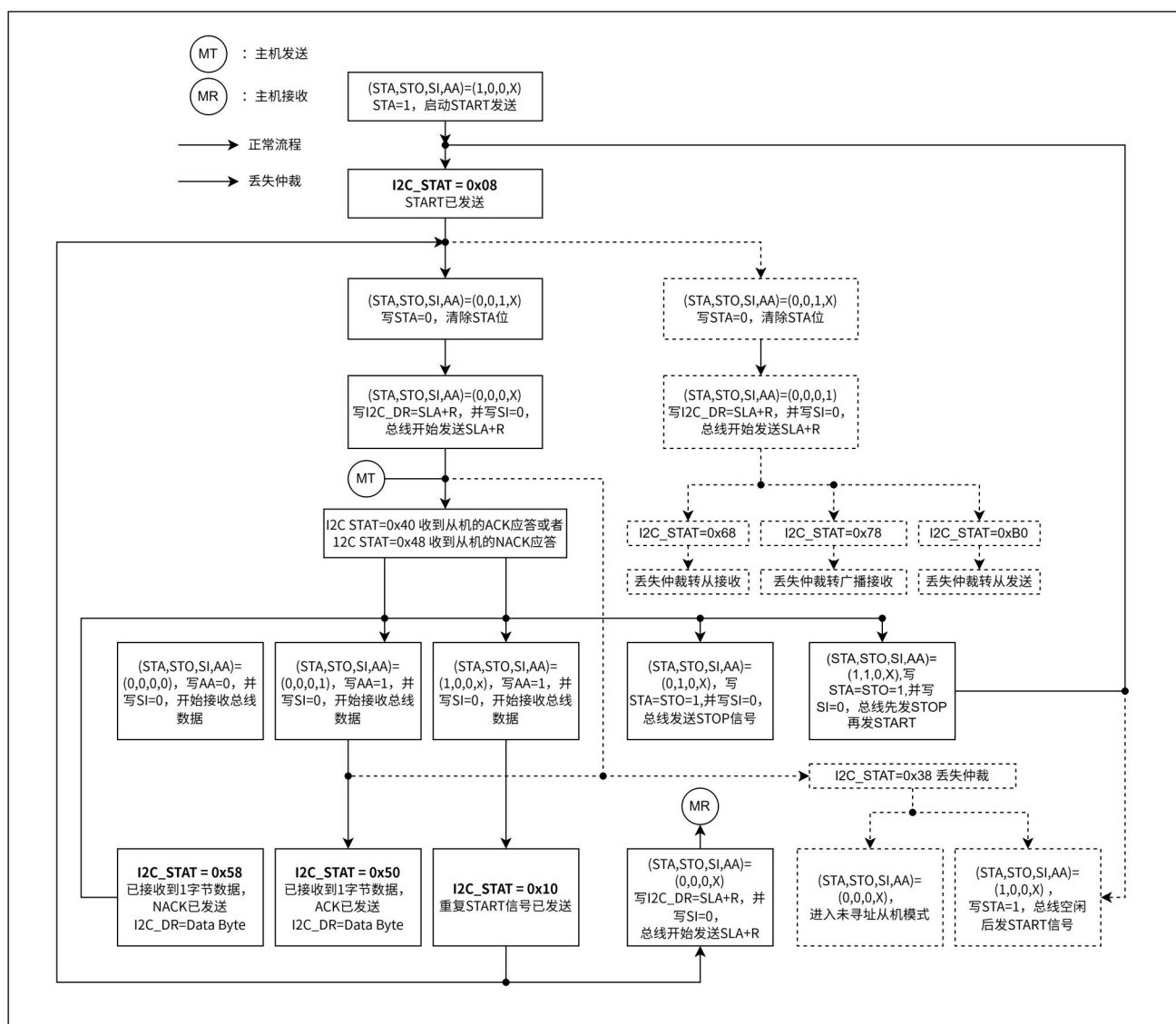


图 18-10 主机接收模式流程和寄存器状态

### 18.4.8.3 从机接收模式

从机接收模式下，从机接收主机发送的数据。

在传输之前，从机需要设定从机地址：将从机地址写入到 3 个从机地址寄存器 I2C\_ADDR0、I2C\_ADDR1、I2C\_ADDR2 任意 1 个中；设置 I2C\_CR.AA 为 1 以响应主机的寻址；I2C\_BRR 寄存器设置值无效，不用设置。

完成上述初始化工作后，从机进入空闲状态（未寻址从机接收模式），等待被主机发送的写信号（SLA+W）寻址。当从机接收到 SLA+W 后，如果地址匹配到 I2C\_ADDR0/1/2，则从机回应 ACK 应答，并进入已寻址从机接收模式，状态码 I2C\_STAT 变为 0x60，中断标志位 I2C\_CR.SI 同时被置 1。此时必须清除 I2C\_CR.SI 位，以便从总线上接收主机发送的数据。

从机每接收到 1 字节数据都要回应 1 个 ACK 应答，应用程序读取完该字节数据后，必须将 I2C\_CR.SI 位清零，为接收下一字节数据做好准备。

从机在接收数据过程中，如果 I2C\_CR.AA 被清零，则从机将在接收到下一字节时返回 NACK 信号，从机自身状态也切换到未寻址从机接收模式，结束与主机的通信，不再接收数据，且 I2C\_DR 寄存器保持之前接收到的数据。由该特性可知，从机应用程序可通过设置 I2C\_CR.AA 为 0 主动将从机从已寻址从机接收模式切换到未寻址从机接收模式。

当主机在 SLA+ 读写阶段由于总线冲突丢失仲裁时会进入未寻址从机接收模式，之后如果接收到符合本机地址的 SLA+W 并回应 ACK 后（状态码 I2C\_STAT = 0x68），则会进入已寻址从机接收模式。

从机接收模式下状态同步图如下图所示：

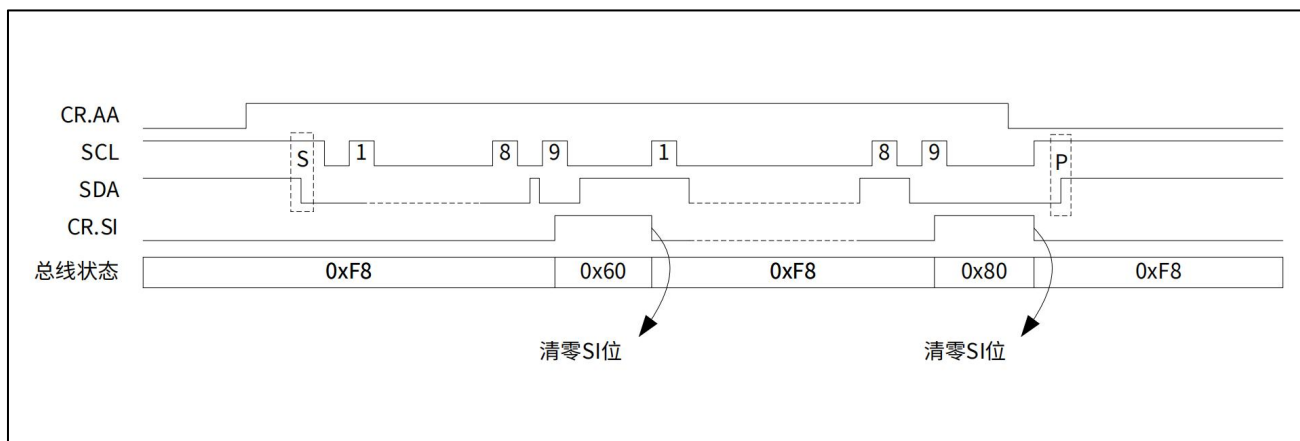


图 18-11 从机接收模式状态同步图

从机接收模式下数据接收流程图及状态寄存器值如下图所示：

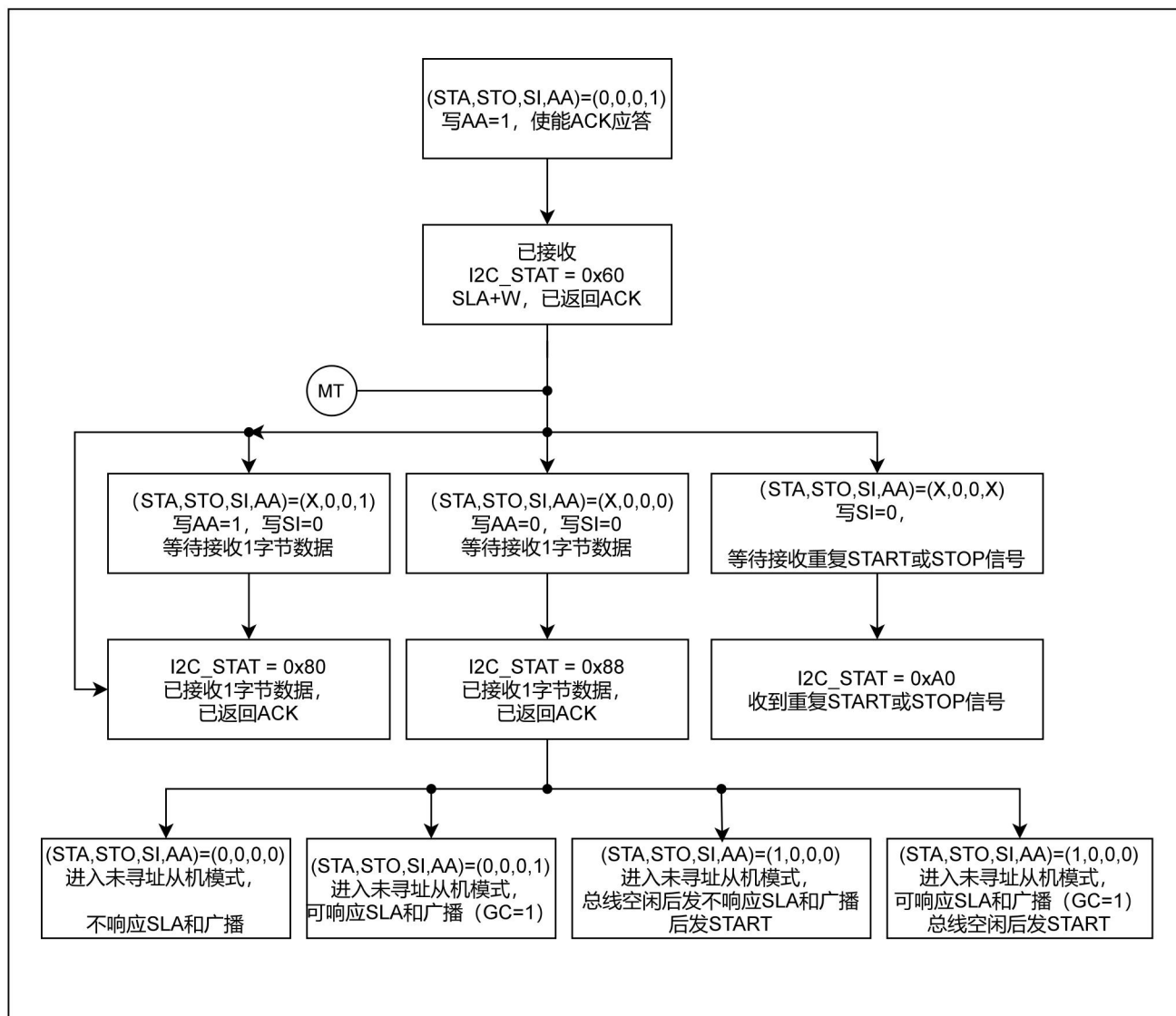


图 18-12 从机接收模式流程和寄存器状态



#### 18.4.8.4 从机发送模式

从机发送模式下，数据由从机发送给主机。

在传输之前，从机需要设定从机地址：将从机地址写入到 3 个从机地址寄存器 I2C\_ADDR0、I2C\_ADDR1、I2C\_ADDR2 任意 1 个中；设置 I2C\_CR.AA 为 1 以响应主机的寻址；I2C\_BRR 寄存器设置值无效，不用设置。

完成上述初始化工作后，从机进入空闲状态（未寻址从机接收模式），等待被主机发送的读信号（SLA+R）寻址。当从机接收到 SLA+R 后，如果地址匹配到 I2C\_ADDR0/1/2，则从机回应 ACK 应答，并进入已寻址从机发送模式，状态码 I2C\_STAT 变为 0xA8，中断标志位 I2C\_CR.SI 同时被置 1。此时应及时将待发送的数据写入 I2C\_DR 寄存器，并清除 I2C\_CR.SI 位，等待发送 1 字节数据，并在每个字节数据发送完成后进行 ACK 应答确认，直到全部数据发送完毕。

如果在传输过程中从机收到 NACK 应答，则从机不再发送数据，并进入未寻址从机接收模式。

如果在传输过程中从机主动将 I2C\_CR.AA 设置为 0，则从机在发送完最后 1 字节有效数据后，将自身切换为未寻址从机接收模式，此后主机从总线上读数据将得到 0xFF。

当主机在 SLA+ 读写阶段由于总线冲突丢失仲裁时会进入未寻址从机接收模式，之后如果接收到符合本机地址的 SLA+R 并回应 ACK 后（状态码 I2C\_STAT = 0xB0），则会进入已寻址从机发送模式。

从机发送模式下状态同步图如下图所示：

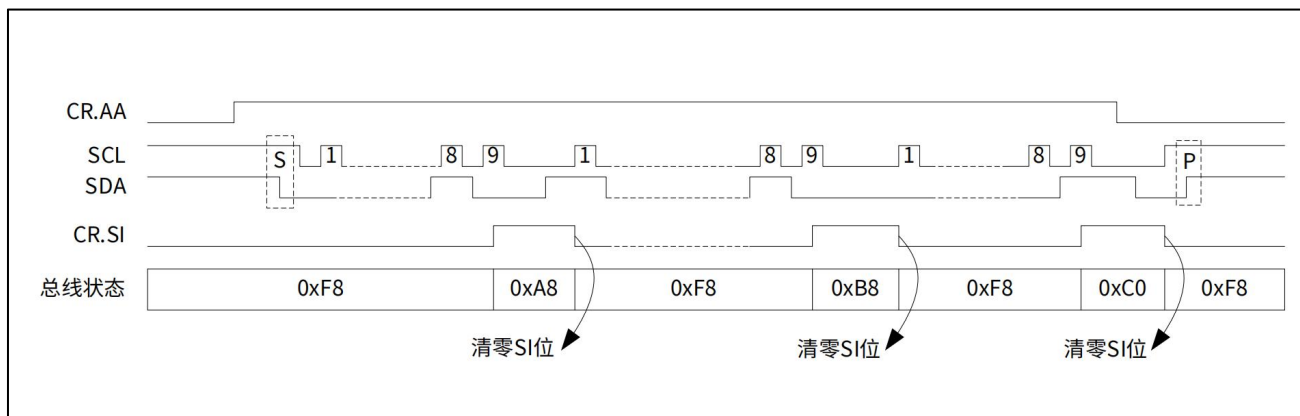


图 18-13 从机发送模式状态同步图

从机发送模式下数据发送流程图及状态寄存器值如下图所示：

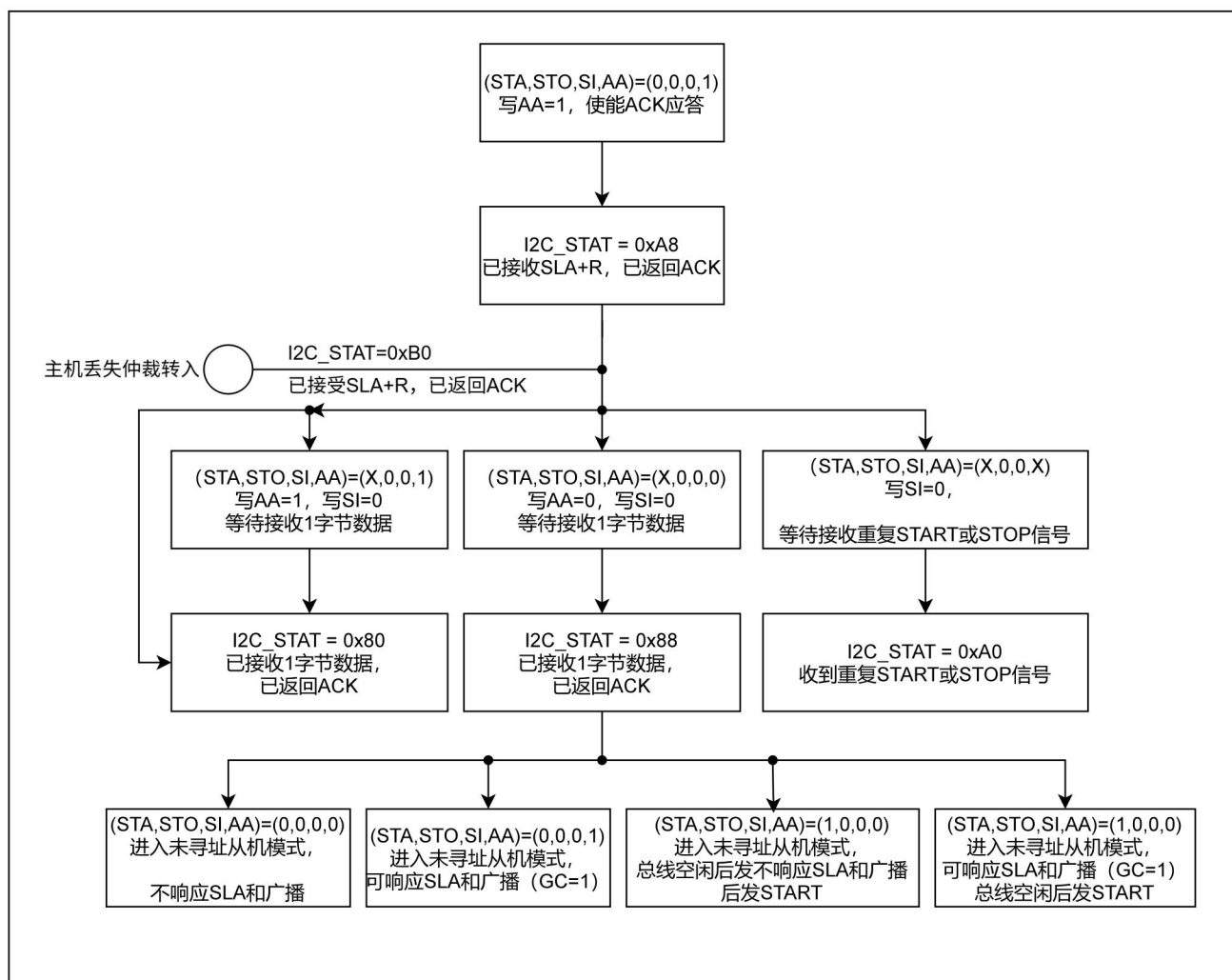


图 18-14 从机发送模式流程和寄存器状态

#### 18.4.8.5 广播接收模式

广播接收模式是一种特殊的从机接收模式，当从机处于空闲状态（未寻址从机接收模式），收到主机发送的 SLA+W，且 SLA 为广播地址 0x00 时进入广播接收模式。该模式下数据接收过程和从机接收模式相同，但 I2C 总线状态码不同。

从机要接收主机发送的广播信息，需设置 I2C\_CR.AA 为 1 以及 I2C\_ADDR0.GC 为 1 以响应主机的广播寻址和广播数据；3 个从机地址寄存器 I2C\_ADDR0/1/2 不用设置；I2C\_BRR 设置值无效，不用设置。

完成上述初始化工作后，从机进入空闲状态（未寻址从机接收模式），等待被主机发送的写信号（SLA+W）寻址。当从机接收到 SLA+W 后，如果地址匹配到广播地址 0x00，则从机回应 ACK 应答，并进入广播接收模式，状态码 I2C\_STAT 变为 0x70，中断标志位 I2C\_CR.SI 同时被置 1。此时必须及时清除 I2C\_CR.SI 位，以便从总线上接收主机发送的数据。

从机每接收到 1 字节数据都要回应 1 个 ACK 应答，应用程序读取完该字节数据后，必须将 I2C\_CR.SI 位清零，为接收下 1 字节数据做好准备。



从机在接收数据过程中，如果 I2C\_CR.AA 被清零，则从机将在接收到下一字节时返回 NACK 信号，从机自身状态也切换为未寻址从机接收模式，结束与主机的通信，不再接收数据，且 I2C\_DR 寄存器保持之前接收到的数据。由该特性可知，从机应用程序可通过设置 I2C\_CR.AA 为 0 主动将从机从已寻址广播接收模式切换未寻址从机接收模式。

当主机在 SLA+ 读写阶段由于总线冲突丢失仲裁时会进入未寻址从机接收模式，之后如果接收到符合广播地址的 SLA+W 并回应 ACK 后（状态码 I2C\_STAT = 0x78），则会进入已寻址广播接收模式。

广播接收模式下数据接收流程图及状态寄存器值如下图所示：

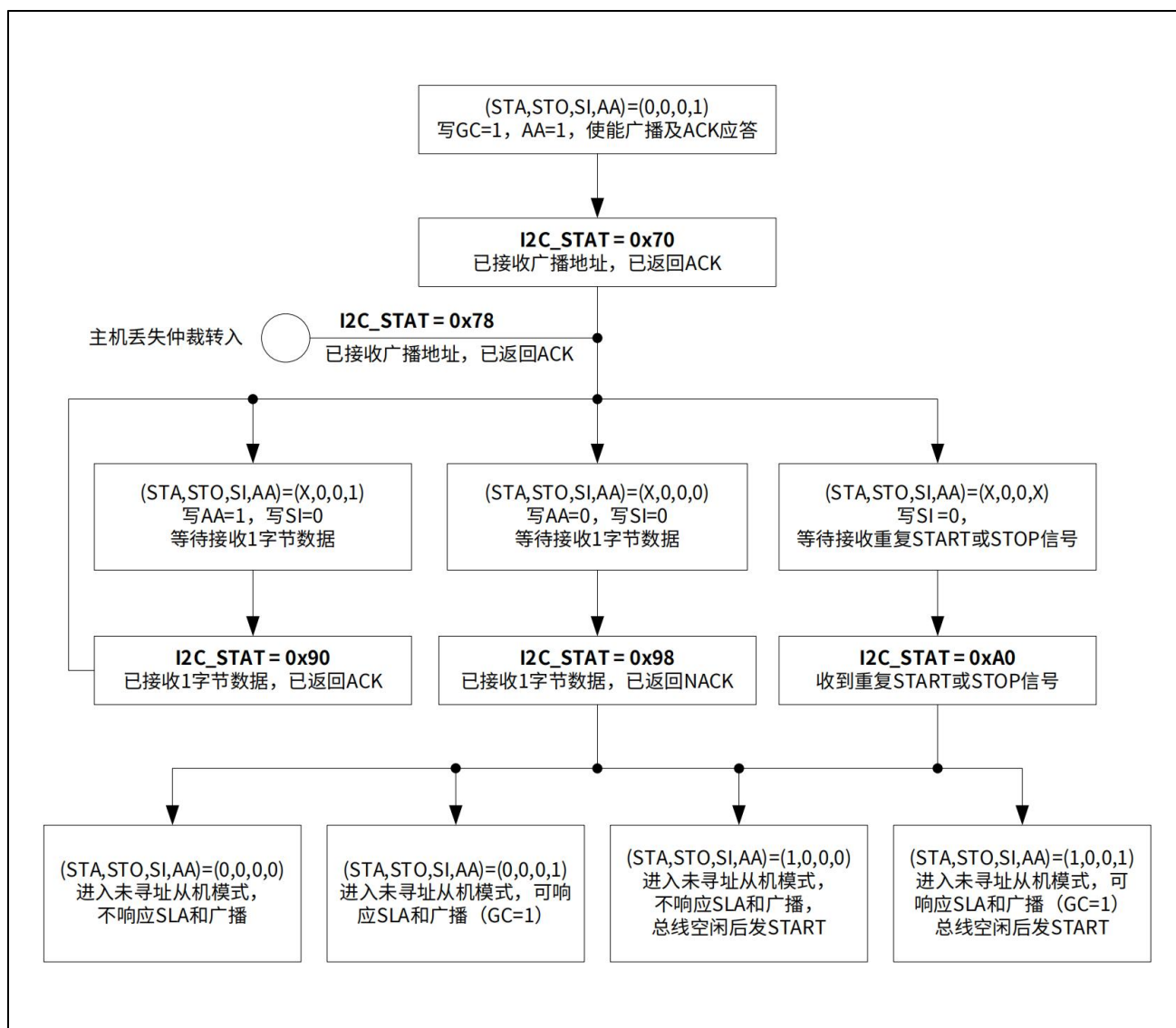


图 18-15 广播接收模式流程和寄存器状态

## 18.4.9 多主机通信

在一些应用中，1 个 I2C 总线上有 2 个或多个主机同时访问从机，并有可能同时在传送数据，此时 SDA 总线上会存在数据冲突。

HC32F0653 的 I2C 能进行 SDA 总线上的数据冲突检测和仲裁，实现多主机应用。如果两个主机同时发送数据，检测到冲突的主机会丢失仲裁并进入未寻址从机模式，未检测到冲突的主机会赢得仲裁并继续主导本次数据通信流程。

### 18.4.10 I2C 状态码

I2C 总线状态通过 I2C 状态寄存器 I2C\_STAT 来标识，共 26 个正常接收或发送状态，和 2 个特殊状态（0xF8:I2C 总线无可可用信息；0x00：总线错误）。

I2C 无论处于主机发送、主机接收、从机接收、从机发送或广播接收模式，当状态寄存器 I2C\_STAT 的内容改变时，都会将 I2C\_CR.SI 置位，且产生 I2C 中断。

I2C 状态码如下表所示：

工作模式	状态码	含义
主机发送模式	08H	已发送起始信号
	10H	已发送重复起始信号
	18H	已发送 SLA+W，已接收 ACK
	20H	已发送 SLA+W，已接收 NACK
	28H	已发送 I2C_DR 中的数据，已接收 ACK
	30H	已发送 I2C_DR 中的数据，已接收 NACK
	38H	主机在发送 SLA+W 阶段或者发送数据阶段丢失仲裁
主机接收模式	08H	已发送起始信号
	10H	已发送重复起始信号
	38H	主机在发送 SLA+R 阶段或者回应 NACK 阶段丢失仲裁
	40H	已发送 SLA+R，已接收 ACK
	48H	已发送 SLA+R，已接收 NACK
	50H	已接收数据字节，ACK 已返回
	58H	已接收数据字节，NACK 已返回
从机接收模式	60H	已接收自身的 SLA+W，已返回 ACK
	68H	当主机时在 SLA+ 读写阶段丢失仲裁，已接收自身的 SLA+W，已返回 ACK
	80H	前一次寻址使用自身从地址，已接收数据字节，已返回 ACK
	88H	前一次寻址使用自身从地址，已接收数据字节，已返回 NACK
	A0H	已寻址从机等待接收数据时，接收到停止条件或重复起始条件
从机发送模式	A8H	已接收自身的 SLA+R，已返回 ACK
	B0H	当主机时在 SLA+ 读写阶段丢失仲裁，已接收自身 SLA+R，已返回 ACK
	B8H	已发送数据字节，已接收 ACK
	C0H	已发送数据字节，已接收 NACK
	C8H	从机最后一个数据字节已被发送，并已接收 ACK

工作模式	状态码	含义
广播接收模式	70H	已接收广播地址 (0x00), 已返回 ACK
	78H	当主机时在 SLA+ 读写阶段丢失仲裁, 已接收广播地址, 已返回 ACK
	90H	前一次寻址使用广播地址, 已接收数据字节, 已返回 ACK
	98H	前一次寻址使用广播地址, 已接收数据字节, 已返回 NACK
	A0H	已寻址从机等待接收数据时, 接收到停止条件或重复起始条件
其它	F8H	无可用的相关状态信息, I2C_CR.SI=0
	00H	传输过程出现总线错误, 或外部干扰使 I2C 进入未定义的状态

特殊状态码 F8H, 表示当前时刻没有任何有用信息, 还不能确定当前总线的状态, 因为 I2C\_CR.SI 还没有被置位, 无中断产生。这种情况在其它状态和 I2C 模块还未开始执行串行传输之前出现。

特殊状态码 00H, 表示 I2C 串行传输过程中出现了总线错误, 如 START 或者 STOP 信号出现在数据帧的错误位置上 (包括在串行传输过程中的地址字节、数据字节或应答位) 或者当外部干扰影响到内部 I2C 模块信号等。总线错误出现时, I2C\_CR.SI 标志位会立即被置位, 且设备立即被切换到未寻址从机接收模式, 释放 SDA 和 SCL, 并将 I2C\_DR 寄存器清零。

当检测到 I2C 总线的 STAT 为总线错误 (状态码为 00H) 时, 由于 I2C 总线为持续使能状态, 并且对 I2C 模块来说错误并没有清除, SI 会持续保持为 1, 即 SI 无法被清除。

总线错误清除方法:

1. 向总线发送 STOP 信号: 置位 STO 位并清除 SI 位 (由于当前处于总线错误状态, 控制器并不会将 STOP 信号实际发送到总线上), STO 位会被硬件自动清 0, 释放总线到正常空闲状态。
2. 如果置 STO 位仍然无法清除 SI 位, 说明时序已被打乱, 需要依次设置 I2C\_CR.EN 为 0 和 1, 即对 I2C 模块进行关闭并重启, 然后设置 SI 为 0 清除 SI 位。

在各工作模式下，I2C 总线状态转换图如下图所示。注意，两种正常状态之间转换时，当未完成动作（如发送 SLA 过程中），即进入新的状态之前，状态码会出现短时的过渡状态，0xF8。用户可不关心，且不会产生中断。

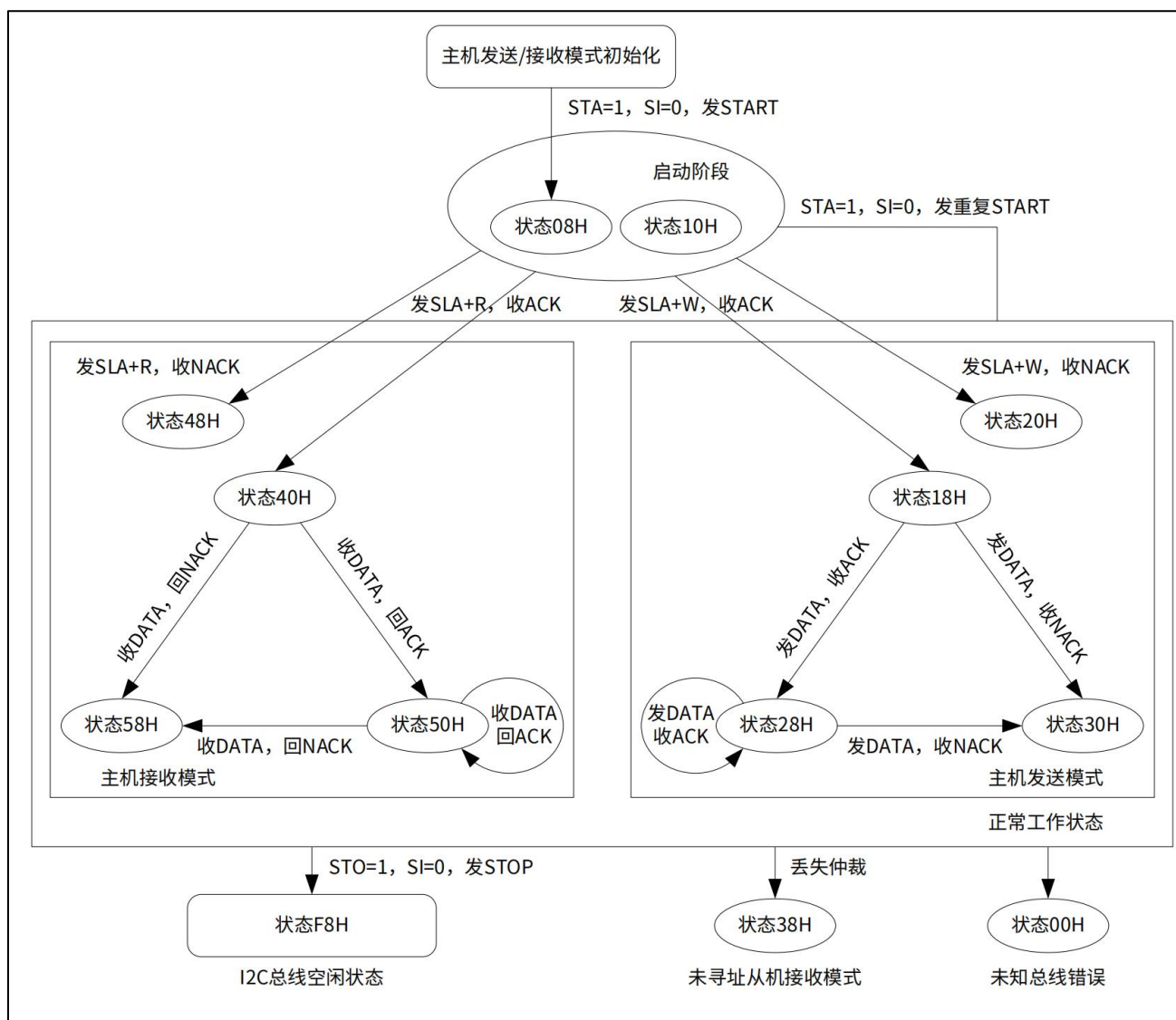


图 18-16 主机发送 / 接收模式状态转换图

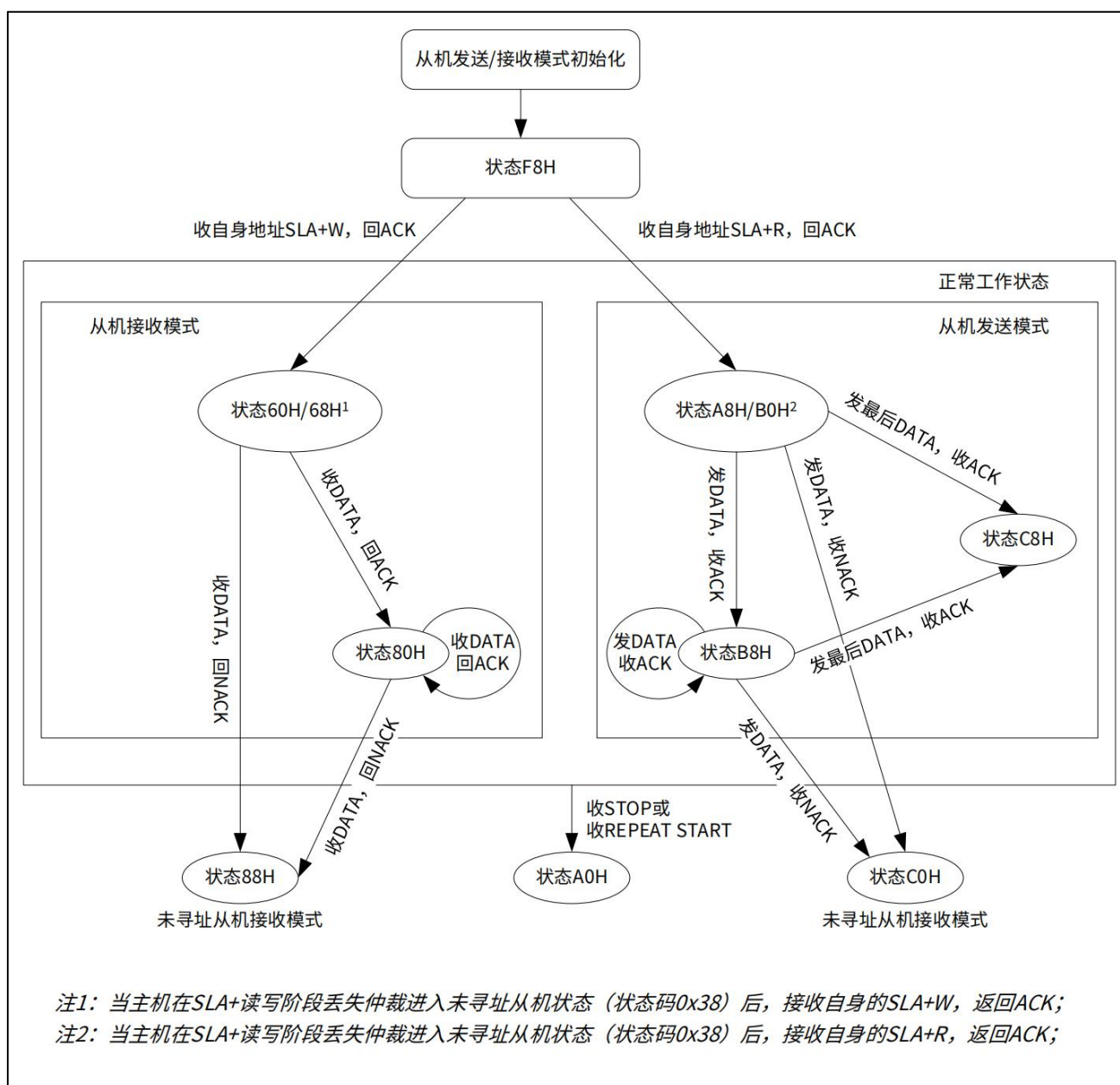


图 18-17 从机发送 / 接收模式状态转换图

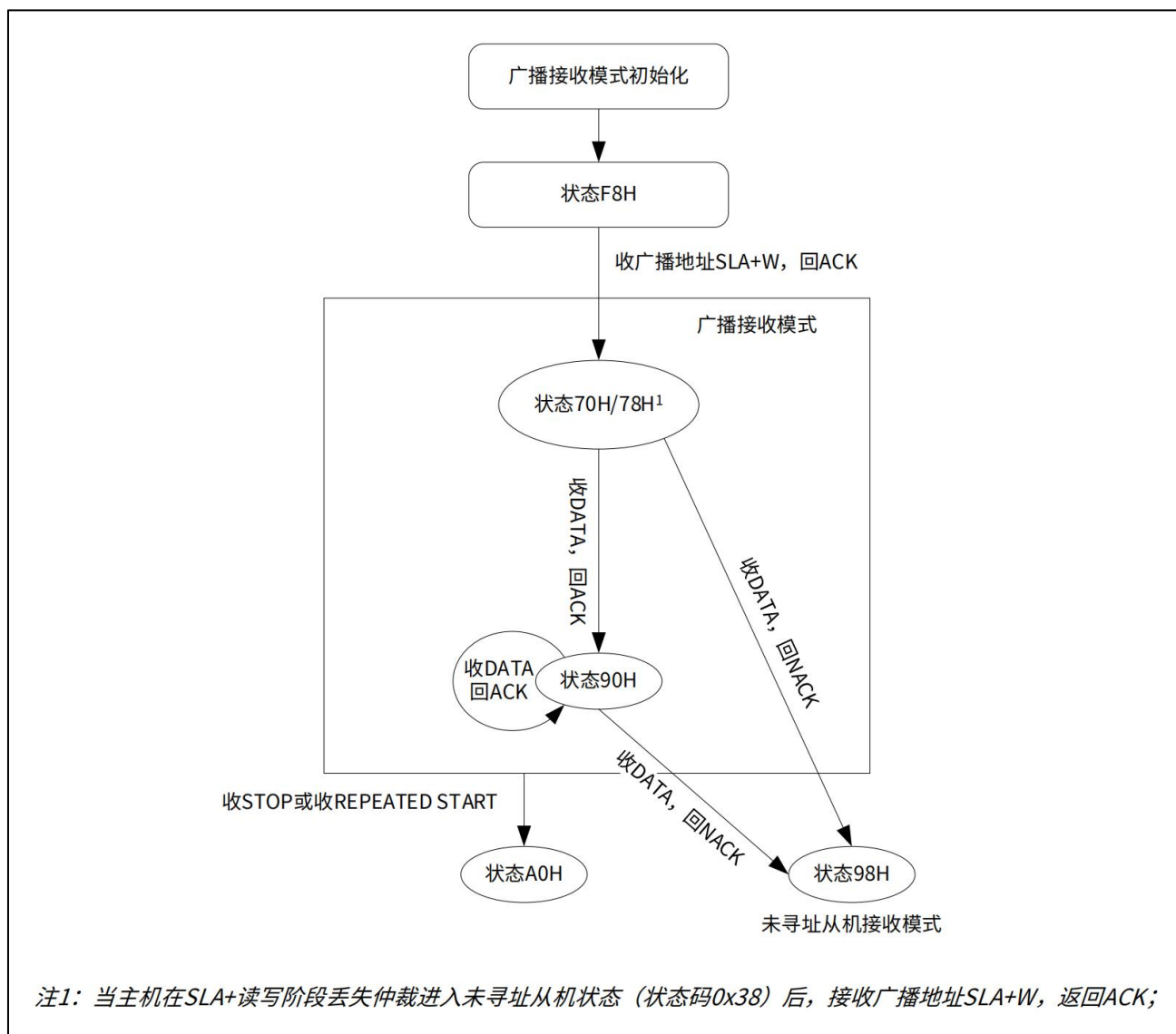


图 18-18 广播接收模式状态转换图

## 18.5 编程示例

### 18.5.1 主机发送示例

- 步骤 1:** 按 GPIO 章节引脚数字复用功能的相关描述，将 SCL、SDA 映射到需要的引脚，并配置 SCL、SDA 引脚为开漏输出模式；
- 步骤 2:** 设置 `SYSCTRL_APBEN1.I2C` 为 1，使能 I2C 模块时钟；
- 步骤 3:** 向 `SYSCTRL_APBRS1.I2C` 依次写入 0、1，复位 I2C 模块；
- 步骤 4:** 配置 `I2C_BRR`，使 SCL 的时钟速率符合应用需求；
- 步骤 5:** 设置 `I2C_BRREN` 为 1，使能 SCL 时钟发生器；
- 步骤 6:** 设置 `I2C_CR.EN` 为 1，使能 I2C 模块；
- 步骤 7:** 设置 `I2C_CR.STA` 为 1，总线尝试发送 START 信号；
- 步骤 8:** 等待 `I2C_CR.SI` 变为 1，START 信号已发送到总线上；
- 步骤 9:** 查询 `I2C_STAT`，如果该寄存器值为 0x08 或 0x10，继续执行下一步骤，否则进行出错处理；
- 步骤 10:** 向 `I2C_DR` 中写入 `SLA+W`，设置 `I2C_CR.STA` 为 0，设置 `I2C_CR.SI` 为 0，发送 `SLA+W`；
- 步骤 11:** 等待 `I2C_CR.SI` 变为 1，`SLA+W` 已发送到总线上；
- 步骤 12:** 查询 `I2C_STAT`，如果该寄存器值为 0x18，继续执行下一步骤，否则进行出错处理；
- 步骤 13:** 向 `I2C_DR` 写入待发送的数据，设置 `I2C_CR.SI` 为 0，发送数据；
- 步骤 14:** 等待 `I2C_CR.SI` 变为 1，数据已发送到总线上；
- 步骤 15:** 查询 `I2C_STAT`，如果该寄存器值为 0x28，继续执行下一步骤，否则进行出错处理；
- 步骤 16:** 如待发送的数据未完成，则跳转到步骤 13 继续执行；
- 步骤 17:** 设置 `I2C_CR.STO` 为 1，设置 `I2C_CR.SI` 为 0，发送 STOP 停止信号，结束本次数据传输。



## 18.5.2 主机接收示例

- 步骤 1:** 按 GPIO 章节引脚数字复用功能的相关描述, 将 SCL、SDA 映射到需要的引脚, 并配置 SCL、SDA 引脚为开漏输出模式;
- 步骤 2:** 设置 SYSCTRL\_APBEN1.I2C 为 1, 使能 I2C 模块时钟;
- 步骤 3:** 向 SYSCTRL\_APBEN1.I2C 依次写入 0、1, 复位 I2C 模块;
- 步骤 4:** 配置 I2C\_BRR, 使 SCL 的时钟速率符合应用需求;
- 步骤 5:** 设置 I2C\_BRREN 为 1, 使能 SCL 时钟发生器;
- 步骤 6:** 设置 I2C\_CR.EN 为 1, 使能 I2C 模块;
- 步骤 7:** 设置 I2C\_CR.STA 为 1, 总线尝试发送 START 信号;
- 步骤 8:** 等待 I2C\_CR.SI 变为 1, START 信号已发送到总线上;
- 步骤 9:** 查询 I2C\_STAT, 如果寄存器值为 0x08 或 0x10, 继续执行下一步骤, 否则进行出错处理;
- 步骤 10:** 向 I2C\_DR 写入 SLA+R, 设置 I2C\_CR.STA 为 0, 设置 I2C\_CR.SI 为 0, 发送 SLA+R;
- 步骤 11:** 等待 I2C\_CR.SI 变为 1, SLA+R 已发送到总线上;
- 步骤 12:** 查询 I2C\_STAT, 如果寄存器值为 0x40 (已收到 ACK), 继续执行下一步骤, 否则进行出错处理;
- 步骤 13:** 设置 I2C\_CR.AA 为 1, 使能应答标志;
- 步骤 14:** 设置 I2C\_CR.SI 为 0, 等待接收 1 字节数据 (主机发送时钟, 从机在时钟作用下发送数据);
- 步骤 15:** 等待 I2C\_CR.SI 变为 1 (主机完成 1 字节数据接收并已回应 ACK 信号), 从 I2C\_DR 读取已接收到的数据;
- 步骤 16:** 查询 I2C\_STAT, 如果该寄存器值为 0x50 或 0x58, 继续执行下一步骤, 否则进行出错处理;
- 步骤 17:** 如果待接收的数据只差最后一个字节, 设置 I2C\_CR.AA 为 0, 使能非应答标志;
- 步骤 18:** 如待接收的数据未完成, 则跳转到步骤 14 继续执行;
- 步骤 19:** 设置 I2C\_CR.STO 为 1, 设置 I2C\_CR.SI 为 0, 发送 STOP 停止信号, 结束本次数据传输。



### 18.5.3 从机接收示例

- 步骤 1:** 按 GPIO 章节引脚数字复用功能的相关描述, 将 SCL、SDA 映射到需要的引脚, 并配置 SCL、SDA 引脚为开漏输出模式;
- 步骤 2:** 设置 SYSCTRL\_APBEN1.I2C 为 1, 使能 I2C 模块时钟;
- 步骤 3:** 向 SYSCTRL\_APBEN1.I2C 依次写入 0、1, 复位 I2C 模块;
- 步骤 4:** 设置 I2C\_CR.EN 为 1, 使能 I2C 模块;
- 步骤 5:** 配置 I2C\_ADDR0 为从机地址;
- 步骤 6:** 设置 I2C\_CR.AA 为 1, 使能应答标志;
- 步骤 7:** 等待 I2C\_CR.SI 变为 1, 被 SLA+W 寻址;
- 步骤 8:** 查询 I2C\_STAT, 如果该寄存器值为 0x60, 继续执行下一步骤, 否则进行出错处理;
- 步骤 9:** 设置 I2C\_CR.SI 为 0, 等待主机发送数据, 并回应 ACK 信号;
- 步骤 10:** 等待 I2C\_CR.SI 变为 1, 从 I2C\_DR 中读取已接收到的数据;
- 步骤 11:** 查询 I2C\_STAT, 如果该寄存器值为 0x80, 继续执行下一步骤, 否则进行出错处理;
- 步骤 12:** 如待接收的数据未完成, 则跳转到步骤 9 继续执行;
- 步骤 13:** 设置 I2C\_CR.AA 为 0, 设置 I2C\_CR.SI 为 0, 从机切换到未寻址从机接收模式, 且不响应主机寻址。

### 18.5.4 从机发送示例

- 步骤 1.** 按 GPIO 章节引脚数字复用功能的相关描述, 将 SCL、SDA 映射到需要的引脚, 并配置 SCL、SDA 引脚为开漏输出模式;
- 步骤 2.** 设置 SYSCTRL\_APBEN1.I2C 为 1, 使能 I2C 模块时钟;
- 步骤 3.** 向 SYSCTRL\_APBEN1.I2C 依次写入 0、1, 复位 I2C 模块;
- 步骤 4.** 设置 I2C\_CR.EN 为 1, 使能 I2C 模块;
- 步骤 5.** 配置 I2C\_ADDR0 为从机地址;
- 步骤 6.** 设置 I2C\_CR.AA 为 1, 使能应答标志;
- 步骤 7.** 等待 I2C\_CR.SI 变为 1, 被 SLA+R 寻址;
- 步骤 8.** 查询 I2C\_STAT, 如果该寄存器的值为 0xA8, 继续执行下一步骤, 否则进行出错处理;
- 步骤 9.** 向 I2C\_DR 写入待发送的数据, 设置 I2C\_CR.SI 为 0, 准备发送数据;
- 步骤 10.** 等待 I2C\_CR.SI 变为 1, 表示数据已发送到总线上, 并收到 ACK 或者 NACK 应答;
- 步骤 11.** 查询 I2C\_STAT, 如果该寄存器的值为 0xB8 时, 继续执行下一步骤, 否则进行出错处理;
- 步骤 12.** 如待发送的数据未完成, 则跳转到步骤 9 继续执行;
- 步骤 13.** 设置 I2C\_CR.AA 为 0, 设置 I2C\_CR.SI 为 0, 从机切换到未寻址从机接收模式, 且不响应主机寻址。

## 18.6 寄存器列表

I2C 基地址: I2C\_BASE = 0x4001 1C00

表 18-2 I2C 寄存器列表

寄存器名称	寄存器地址	寄存器描述
I2C_BRREN	I2C_BASE + 0x00	波特率计数器使能寄存器
I2C_BRR	I2C_BASE + 0x04	波特率计数器配置寄存器
I2C_CR	I2C_BASE + 0x08	控制寄存器
I2C_DR	I2C_BASE + 0x0C	数据寄存器
I2C_ADDR0	I2C_BASE + 0x10	从机地址 0 寄存器
I2C_STAT	I2C_BASE + 0x14	状态寄存器
I2C_ADDR1	I2C_BASE + 0x20	从机地址 1 寄存器
I2C_ADDR2	I2C_BASE + 0x24	从机地址 2 寄存器
I2C_MATCH	I2C_BASE + 0x28	从机地址匹配标志寄存器
I2C_ISR	I2C_BASE + 0x2C	中断标志寄存器

## 18.7 寄存器描述

### 18.7.1 I2C\_BRREN 波特率计数器使能寄存器

地址：请参见 表 18-2 I2C 寄存器列表      复位值：0x0000 0000

位编号	位符号	权限	说明
31-1	RFU	-	保留位，请保持默认值
0	EN	RW	I2C 总线 SCL 波特率计数器使能控制 0：禁止 1：使能 注：主机时应使能 EN，从机时该位不影响

### 18.7.2 I2C\_BRR 波特率计数器配置寄存器

地址：请参见 表 18-2 I2C 寄存器列表      复位值：0x0000 0000

位编号	位符号	权限	说明
31-8	RFU	-	保留位，请保持默认值
7-0	BRR	RW	I2C 总线 SCL 波特率配置 $f_{I2C} = f_{PCLK}$ $f_{SCL} = f_{I2C} / 8 / (BRR + 1)$ ，其中 $BRR > 0$

### 18.7.3 I2C\_CR 控制寄存器

地址：请参见 表 18-2 I2C 寄存器列表      复位值：0x0000 0000

位编号	位符号	权限	说明
31-7	RFU	-	保留位，请保持默认值
6	EN	RW	模块使能控制 0：禁止 1：使能
5	STA	RW	总线状态控制 W0：清零 STA W1：向总线发送 START 起始信号 注 1：设置 STA 为 1 后，如果总线空闲，则发送 START 起始信号，如果总线忙则等待 I2C 停止后，发送 START 信号。 注 2：如果设备已经在主机模式且已发送一个或多个字节，此时再设置 STA，I2C 总线将产生 Repeated START 重复起始信号。 注 3：STA 可在任何时间置 1，包括从机模式。但硬件不会在完成 START 或 repeat START 信号发送后自动清 0，需要用户手动清除 STA。
4	STO	RW	总线状态控制 W0：无功能 W1：向总线发送 STOP 停止信号 注 1：硬件会在完成 STOP 信号发送后自动对 STO 清 0。 注 2：如果在主机模式下 STA 和 STO 同时置 1，I2C 总线在发送 STOP 后马上发送 START。在从机模式下，禁止 STA 及 STO 同时置 1，以避免发出非法 I2C 帧。 注 3：当总线上产生错误状态 (STAT 状态字为 00H) STO 也会置 1，但这种情况下 I2C 总线不会发送 STOP 停止信号。
3	RFU	-	保留位，请保持默认值
2	AA	RW	应答控制 0：在应答阶段发送 NACK 1：在应答阶段发送 ACK 注 1：对于已被寻址的从机，在从机接收模式下未回复 ACK 应答位或在从机发送模式下未接收到 ACK 应答位，该从机将切换为未寻址从机接收模式，无法接收数据直到其 AA 被置 1，且重新被主机寻址。 注 2：特殊情况：从机发送模式时，从机发送最后一个字节给主机之前，清除 AA，发送完最后一个字节的位后，从机将切换为未被寻址的从机模式，和主机断开，状态寄存器 I2C_STAT 为 C8H。主机若再从总线上读数据，将得到 0xFF。
1-0	FLT	RW	I2C 滤波参数配置 00：禁止数字滤波器 01：使能数字滤波器，可滤除的噪声尖峰脉宽= 1 t <sub>I2CCLK</sub> 10：使能数字滤波器，可滤除的噪声尖峰脉宽= 8 t <sub>I2CCLK</sub> 11：使能数字滤波器，可滤除的噪声尖峰脉宽= 15 t <sub>I2CCLK</sub>

## 18.7.4 I2C\_DR 数据寄存器

地址：请参见 表 18-2 I2C 寄存器列表      复位值：0x0000 0000

位编号	位符号	权限	说明
31-8	RFU	-	保留位，请保持默认值
7-0	DR	RW	数据寄存器 在发送模式下，写入待发送的数据 在接收模式下，读出接收到的数据

## 18.7.5 I2C\_ISR 中断标志寄存器

地址：请参见 表 18-2 I2C 寄存器列表      复位值：0x0000 0000

位编号	位符号	权限	说明
31-1	RFU	-	保留位，请保持默认值
0	SI	RW	I2C 中断标志 R0：未发生 I2C 中断 R1：已发生 I2C 中断 W0：清除 I2C 中断标志并使状态机执行下一个动作 W1：无功能 注 1：I2C 所有 26 种状态中出现一种，硬件就会置 1 此位 (F8H 除外)，此时软件通过 I2C_STAT 寄存器值，来确认总线当前状态。 注 2：SI 需要软件清零。 注 3：在 SI 被清 0 之前，SCL 低电平周期延长，传输暂停，该状态对于从机处理接收到的数据非常有用，可以确保准确处理前一数据再接收下一个数据。 注 4：在软件清除 SI 前，软件应该准备好合适的寄存器设置。在 SI 被清除后，I2C 总线将会根据寄存器设置执行相应的操作。

## 18.7.6 I2C\_STAT 状态寄存器

地址：请参见 表 18-2 I2C 寄存器列表      复位值：0x0000 00F8

位编号	位符号	权限	说明
31-8	RFU	-	保留位，请保持默认值
7-0	STAT	RO	I2C 状态寄存器，状态值的具体定义详见 I2C 状态码；STAT = F8H 时，表示无可用的相关状态信息，SI 将保持为 0。其它 26 种状态，都会让 SI 置 1，且产生中断请求。

### 18.7.7 I2C\_ADDR0 从机地址 0 寄存器

地址：请参见 表 18-2 I2C 寄存器列表      复位值： 0x0000 0000

位编号	位符号	权限	说明
31-8	RFU	-	保留位，请保持默认值
7-1	ADDR0	RW	从机模式地址 0 注 1：主机模式无效。 注 2：主机需要寻址该从机，需通过在 START 或 Repeated START 之后的第一个字节值地址信息与此地址相同。如果 AA 为 1，该从机响应主机，成为被寻址从机，否则主机广播寻址信息会被忽略。 注 3：I2C_ADDR0[7:1] 不能写为全 0，因为 0x00 为广播方式寻址专用。
0	GC	RW	广播地址应答使能 0：禁止 1：使能 注 1：主机模式无效。 注 2：使能 GC 后，如果 AA 置 1，则使能广播接收模式，若 AA 清 0，则忽略总线上的广播寻址信息。

### 18.7.8 I2C\_ADDR1 从机地址 1 寄存器

地址：请参见 表 18-2 I2C 寄存器列表      复位值： 0x0000 0000

位编号	位符号	权限	说明
31-8	RFU	-	保留位，请保持默认值
7-1	ADDR1	RW	从机模式地址 1 注：主机模式无效。
0	RFU	-	保留位，请保持默认值

### 18.7.9 I2C\_ADDR2 从机地址 2 寄存器

地址：请参见 表 18-2 I2C 寄存器列表      复位值： 0x0000 0000

位编号	位符号	权限	说明
31-8	RFU	-	保留位，请保持默认值
7-1	ADDR2	RW	从机模式地址 2 注：主机模式无效。
0	RFU	-	保留位，请保持默认值

### 18.7.10 I2C\_MATCH 从机地址匹配寄存器

地址：请参见 表 18-2 I2C 寄存器列表      复位值： 0x0000 0000

位编号	位符号	权限	说明
31-3	RFU	-	保留位，请保持默认值

2	ADDR2	RO	I2C 从机模式地址 2 匹配标志位 0: 从总线收到的设备地址与 ADDR2 不相同 1: 从总线收到的设备地址与 ADDR2 相同
1	ADDR1	RO	I2C 从机模式地址 1 匹配标志位 0: 从总线收到的设备地址与 ADDR1 不相同 1: 从总线收到的设备地址与 ADDR1 相同
0	ADDR0	RO	I2C 从机模式地址 0 匹配标志位 0: 从总线收到的设备地址与 ADDR0 不相同 1: 从总线收到的设备地址与 ADDR0 相同

注：地址匹配标志位在以下情况会清零：

- 模块复位时；
- START/STOP 发送时。

## 19 模数转换器（ADC）

### 19.1 概述

HC32F0653 内部集成一个 12 位精度、最高 1.2M SPS 转换速度的逐次逼近型模数转换器 (SAR ADC)，最多可将 30 +6 路模拟信号转换为数字信号。现实世界中的绝大多数信号都是模拟量，如光、电、声、图像信号等，都要由 ADC 转换成数字信号，才能由 MCU 进行数字化处理。

### 19.2 主要特性

- 12 位精度(包括符号位)
- 可编程转换速度，最高达 1.2M SPS
- 30 +6 路输入转换通道
  - 30 路外部引脚输入
  - 内置温度传感器
  - DAC 电压
  - FDAOUT 电压
  - 1/3 VDDA 电源电压
  - VSSA
  - 1.2V 基准电压
- 2 路参考电压源（Vref）(可 1.5 倍量程扩展)
  - VDDA 电源电压
  - 内置 2.4V 参考电压
- 支持序列扫描功能，支持最多 4 个序列，4 个通道，30+6 个信号来源
- 支持输入通道电压阈值监测
- 支持片内外设自动触发 ADC 转换
- 支持直流偏置校准和增益校准
- 9 个带中断标志的中断源(包含看门狗中断、单次转换结束中断、序列转换结束中断、ADC 触发发生在非空闲状态中断)



## 19.3 功能框图

ADC 功能框图如下图所示：

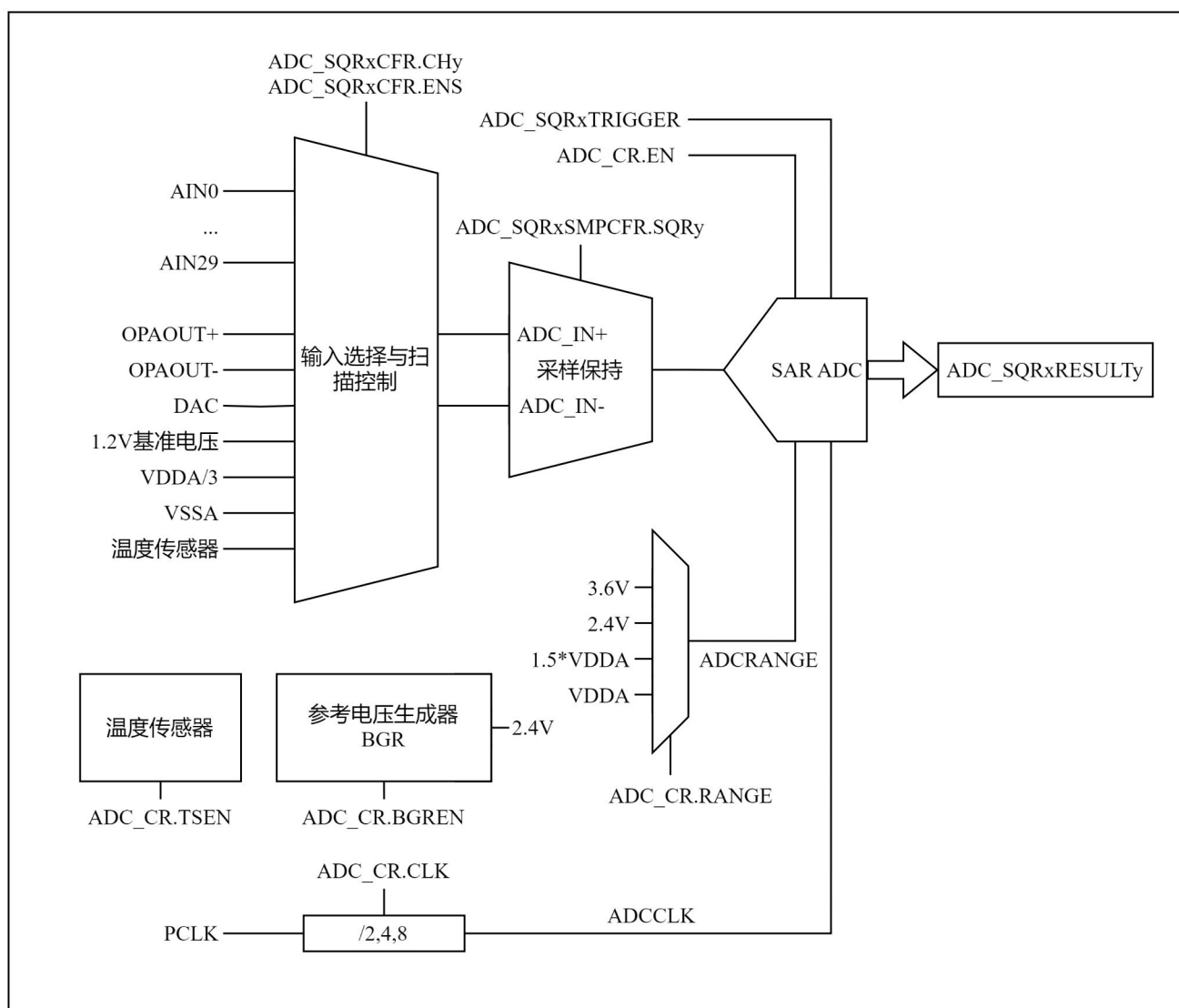


图 19-1 ADC 功能框图

## 19.4 功能描述

### 19.4.1 转换时序

ADC 的转换时序如下图所示：

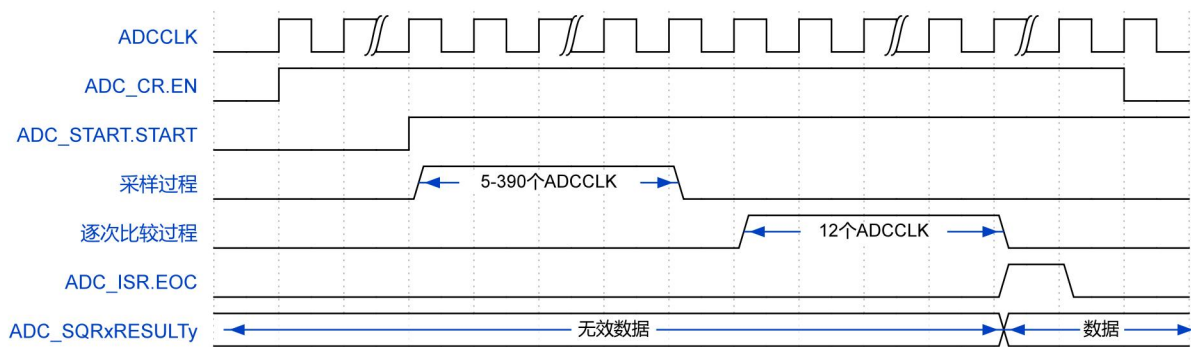


图19-2 ADC转换时序图

向 ADC 控制寄存器 ADC\_CR 的 EN 位域写入 1，使能 ADC 模块。

ADC\_CR.EN 由 0 变为 1 约 1 $\mu$ s 后模拟电路初始化完成，可以开始进行 ADC 转换。

向 ADC 启动寄存器 ADC\_START 的 START 位域写入 1，启动 ADC 转换，转换完成后硬件自动清零。

ADC 工作时钟 ADCCLK，由系统时钟 PCLK 经预分频器分频得到，通过控制寄存器 ADC\_CR 的 CLK 位域可选择 1、2、4、8 分频，如下表所示：

表 19-1 ADC 时钟配置表

ADC_CR.CLK	ADCCLK
000	PCLK
001	PCLK/2
010	PCLK/4
011	PCLK/8

一次完整的 ADC 转换需要 17~402 个 ADCCLK 时钟周期，包括采样阶段和逐次比较两个阶段：

1、采样阶段：需要 5~390 个 ADCCLK 时钟周期。采样周期通过控制寄存器 ADC\_SQRxSMPCFR 的 CHy 位域配置，如下表所示：

表 19-2 ADC 采样周期选择表

ADC_SQRxSMPCFR.CHy	采样周期（ADCCLK 个数）
0	5
1	7
10	9
11	12
100	18
101	24
110	30
111	42

1000	54
1001	70
1010	102
1011	134
1100	166
1101	198
1110	262
1111	390

ADC 采样周期长度由用户对采样的速度要求和采样信号的电气特性决定，用户应选择合适的采样周期，以达到最佳的转换效果。

2、逐次比较阶段：需要 12 个 ADCCLK 时钟周期。

ADC 转换完成之后，转换完成标志位 ADC\_ISR.EOC 会被硬件置 1，ADC 转换结果存储在对应的 ADC 转换结果寄存器 ADC\_SQRxRESULTy (y=0,1,2,3) 中，用户可通过设置 ADC\_ICR.EOC 为 0 清除该标志位。

## 19.4.2 转换速度

ADC 的转换速度的与工作时钟 ADCCLK 的对应关系如下：

ADC 转换速率 =  $f_{ADCCLK} / N_T$

其中， $f_{ADCCLK}$  为 ADCCLK 时钟频率， $N_T$  为一次 ADC 转换所需要的 ADCCLK 个数。

## 19.4.3 转换精度

当 ADC 外部输入信号驱动能力不足，可以调高ADC 采样时间。

## 19.4.4 转换结果

ADC 每次转换完成后，当前转换完成的序列通道 ADC\_SQRxCFR.CHy (y=0、1、2、3) 的转换结果保存在对应的 ADC\_SQRxRESULTy (y=0、1、2、3) 寄存器中。

序列转换结果寄存器是 12 位宽，转换结果右对齐，有效值存储于ADC\_SQRxRESULTy寄存器的低 11 位（位 10:0），最高位（位 11）为符号位。

表19-4 ADC输出数字量数制转换

2.4V 量程输入	3.6V 量程输入	转为有符号数后的数值
2.4	3.6	12'h0111_1111_1111
0	0	12'h0000_0000_0000
-2.4	-3.6	12'h1000_0000_0000

## 19.4.5 工作模式

ADC 支持序列通道转换模式，最多有 4 个转换序列，每个转换序列可对最多 4 个序列通道（ADC\_SQRxCFR.CH0~ADC\_SQRxCFR.CH3）进行轮流转换，序列待转换的通道数量由 ADC\_SQRxCFR 寄存器的 ENS 位域配置。每个序列通道由序列转换通道配置寄存器 ADC\_SQRxCFR 的 CHy 位域进行设置。

在此模式下，无论是通过软件 START 位启动 ADC，还是外部触发启动，启动一次 ADC，将对使能的转换序列全部进行一次转换。

ADC 启动转换信号可以通过 GPIO 送出用于调试。

每次 ADC 转换完成后，ADC\_ISR.EOC 标志位自动置 1，转换结果保存在与序列 ADC\_SQRxCFR.CH0 ~ ADC\_SQRxCFR.CH3 相同序号的转换结果寄存器 ADC\_SQRxRESULT0~ADC\_SQRxRESULT3 中。当所选择的转换序列全部转换完成后，ADC\_ISR.EOSx 标志位变为 1，ADC\_START.START 位自动清 0，ADC 停止转换。

**注1：**假如多个序列使能了同样的触发源，那么触发源来的时候，按照使能开关顺序转换 SQR0, SQR1, SQR2, SQR3，即 SQR0 优先级最高，SQR3 优先级最低。

**注2：**假如在序列转换过程中，有触发源再次触发，那么忽略该次触发。

序列扫描转换模式的时序如下图所示：

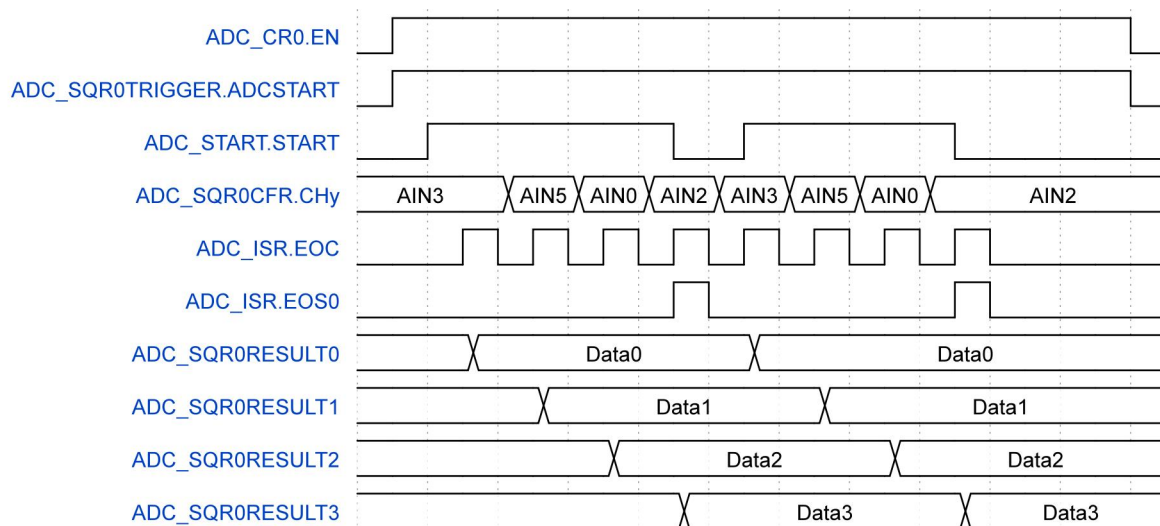


图19-3 序列扫描转换时序图

通过 START 位启动 ADC 序列扫描转换，参考操作流程如下：

1. 设置 RCC\_AHBEN.GPIOx 为 1，RCC\_APBEN.ADC 为 1，使能 ADC 通道对应的 GPIO 时钟和 ADC 工作时钟；
2. 设置 ADC 通道对应的 GPIO 引脚为模拟功能；
3. 设置 ADC\_CR.EN 为 1，使能 ADC 模块；
4. 配置 ADC\_CR.REF，选择 ADC 的参考电压；
5. 配置 ADC\_SQRxSMPCFR 寄存器相应位域及 ADC\_CR.CLK，设置 ADC 的采样速度及时钟选择；
6. 配置 ADC\_SQRxCFR.ENS，选择序列待转换的通道数量，如图 19-3 序列扫描转换时序图所示，设置 ADC\_SQRxCFR.ENS 为 3，序列转换通道为 CH0 ~ CH3；
7. 配置 ADC\_SQRxCFR.CH0，选择序列通道 0 待转换源，如图 19-3 序列扫描转换时序图所示，设置 ADC\_SQRxCFR.SQRCH0 为 3，通道 0 的待转换源为 ADC\_IN3；
8. 配置 ADC\_SQRxCFR.CH1，选择序列通道 1 待转换源，如图 19-3 序列扫描转换时序图所示，设置 ADC\_SQRxCFR.CH1 为 5，通道 1 的待转换源为 ADC\_IN5；

9. 配置 ADC\_SQRxCFR.CH2, 选择序列通道 2 待转换源, 如图 19-3 序列扫描转换时序图所示, 设置 ADC\_SQRxCFR.CH2 为 0, 通道 2 的待转换源为 ADC\_IN0;
10. 配置 ADC\_SQRxCFR.CH3, 选择序列通道 3 待转换源, 如图 19-3 序列扫描转换时序图所示, 设置 ADC\_SQRxCFR.CH3 为 2, 通道 3 的待转换源为 ADC\_IN2;
11. 设置 ADC\_ICR 为 0, 清除 ADC 中断标志;
12. 设置 ADC\_START.START 为 1, 启动 ADC 转换;
13. 等待 ADC\_ISR.EOSx 变为 1, 依次读取 ADC\_SQRxRESULT0 ~ ADC\_SQRxRESULT3 寄存器, 以获取对应通道的 ADC 转换结果。当 ADC\_ISR.EOSx 变为 1 时, 表示一次 4 个通道的序列转换完成, ADC\_START.START 位自动清 0, ADC 转换停止;
14. 如需对其它通道进行转换, 重复执行步骤 6~ 步骤 14;
15. 设置 ADC\_CR.EN 为 0, 关闭 ADC 模块。

## 19.4.6 外部触发源

ADC 转换既可以通过软件启动 (即设置 ADC\_START.START 为 1), 也可通过外部触发启动, 触发源由外部触发寄存器 ADC\_SQRxTRIGGER 选择, 有多种触发 ADC 方式, 详见相关寄存器描述。

## 19.4.7 校准

ADC 硬件接口模块可以进行直流偏置校正与增益校正。

ADC\_AMC 存储的是增益校正系数  $AMP_{correction}$ , 为 10bit 无符号定点数, ADC\_AMC[9] 为整数部分, ADC\_AMC[8:0] 为小数部分。可以表示数值在 1 附近的定点数。

ADC\_DCOFFSET 存储的是 ADC 的直流偏置, 通常在校正阶段通过测量 VSS (内部地) 得到 ADC 直流偏置数值并存入 FLASH 中, 并在系统加载阶段由软件将直流偏置写入 ADC\_DCOFFSET 寄存器中。

记 ADC 输出的数字量为  $D_{ADC}$ ,  $D_{ADC}$  对应的真实值为  $D$ ,  $D_0$  为编码数制的 0, 则

$$D = \text{Saturation}(D_{ADC} - D_0 * AMP_{correction} - DC_{offset})$$

$$AdcValue = (T_0 - 0) * ADC\_AMP - OFFSET$$

最终硬件会将进行校正后的  $D$  存入相应的采样数据寄存器。

假如上溢出为 0x7FF, 下溢出为 0x800

## 19.4.8 模拟看门狗

模拟看门狗功能常用于对模拟量的自动监测, 即将 ADC 转换结果与用户设定的阈值进行比较, 并可产生中断。

模拟看门狗可通过编程 ADC\_AWDCRx 寄存器的位域来保护多条已选的通道, 即 INy 位域置 1 时, 相应通道的看门狗将使能。

模拟看门狗支持上阈值和下阈值比较, 阈值上限和阈值下限分别通过 ADC\_AWDVTH 和 ADC\_AWDVTL 位域来设置。当 ADC 转换的模拟电压低于阈值下限或高于阈值上限时, 模拟看门狗相应标志位会置 1, 如果设置了中断使能寄存器 ADC\_IER 的相应位域 (AWDH、AWDL), 将产生中断请求。上阈值和下阈值比较如下所示:

上阈值比较: 当转换结果位于 [ADC\_AWDVTH, 2047] 区间内时, ADC\_ISR.AWDH 标志位置 1。

下阈值比较: 当转换结果位于 [-2047, ADC\_AWDVTL] 区间内时, ADC\_ISR.AWDL 标志位置 1。

区间值比较: 当转换结果位于 [ADC\_AWDVTL, ADC\_AWDVTH] 区间内时, ADC\_ISR.WDTR 标志位置 1。

模拟看门狗输出的内部硬件信号 ADC\_AWD (上阈值标志位 (ADC\_ISR.WDTH) 和下阈值标志位 (ADC\_ISR.AWDL)) 可直接连接到 GTIM 和 ATIM 的 ETR 输入, 或作为定时器参考信号 OCREF 的清除源, 请参考相应定时器章节。

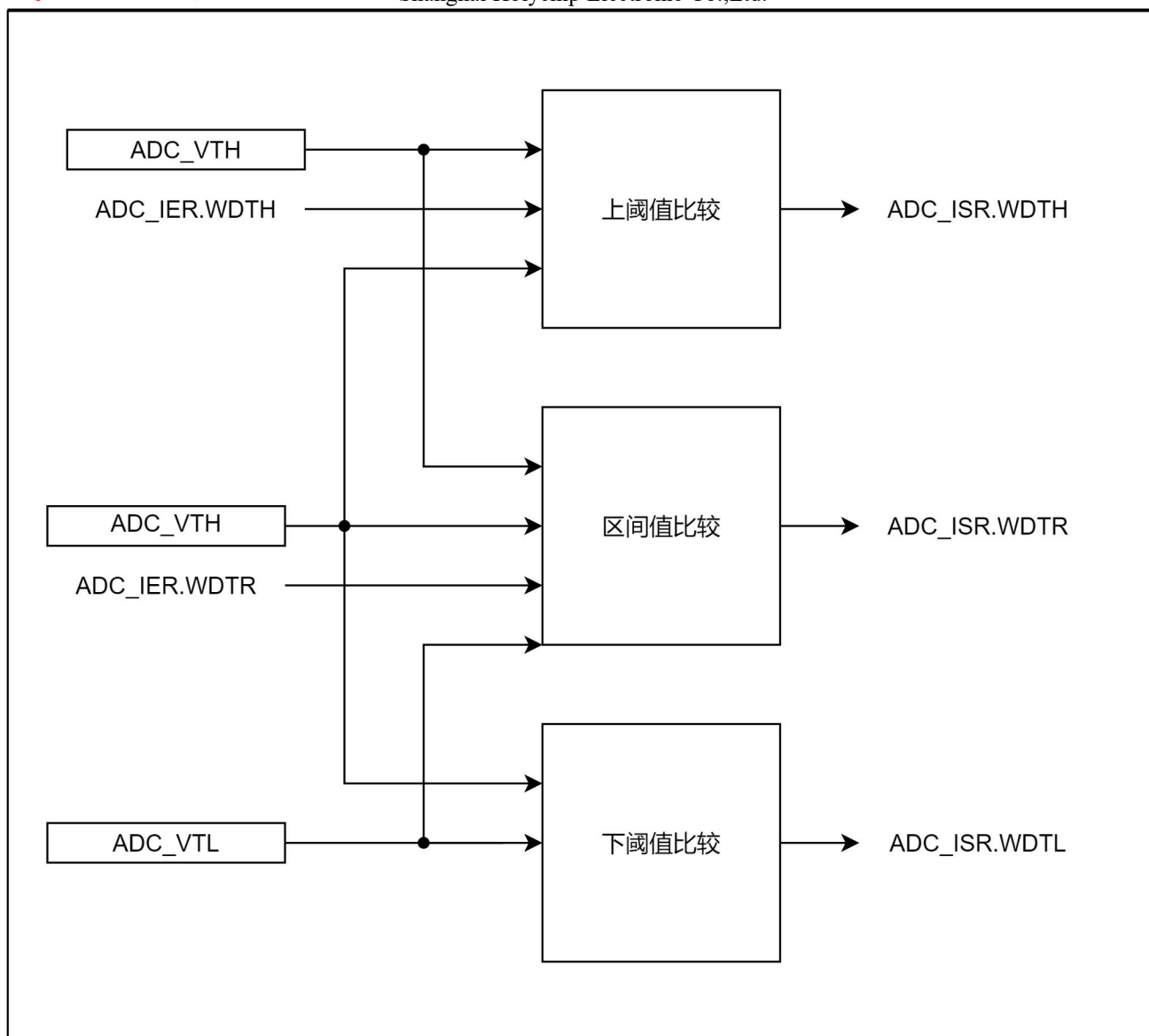


图 19-4 ADC 阈值比较示意图

## 19.4.9 温度传感器

HC32F0653内置温度传感器模块，传感器的输出电压随温度变化，设置 ADC 模块的采样通道为内部温度传感器，

通过 ADC 测量结果可计算得到当前的环境温度。

温度传感器默认处于关闭状态，通过设置控制寄存器 ADC\_CR 的 TSEN 位域为 1，使能温度传感器。

## 19.4.10 ADC 中断

ADC 中断请求，如下表所示：

表 19-6 ADC 中断源与中断标志

中断源	中断标志	中断使能	标志清除
转换结果 $\geq$ ADC_AWDVTL, 且 $<$ ADC_AWDVTH	ADC_ISR.AWDTR	ADC_IER.AWDTR 置 1	ADC_ICR.AWDTR 清 0
转换结果 $\geq$ ADC_AWDVTH	ADC_ISR.AWDTH	ADC_IER.AWDTH 置 1	ADC_ICR.AWDTH 清 0
转换结果 $<$ ADC_AWDVTL	ADC_ISR.AWDTL	ADC_IER.AWDTL 置 1	ADC_ICR.AWDTL 清 0
ADC 序列转换完成	ADC_ISR.EOSx	ADC_IER.EOSx 置 1	ADC_ICR.EOSx 清 0
ADC 转换完成	ADC_ISR.EOC	ADC_IER.EOC 置 1	ADC_ICR.EOC 清 0
ADC 触发发生在非空闲状态	ADC_ISR.TRIGERR	ADC_IER.TRIGERR 置 1	ADC_ICR.TRIGERR 清 0



## 19.5 寄存器列表

ADC 基地址: ADC\_BASE = 0x4001 3400

表 19-7 ADC 寄存器列表

寄存器名称	寄存器地址	寄存器描述
ADC_CR	ADC_BASE + 0x00	ADC 控制寄存器 0
ADC_START	ADC_BASE + 0x04	ADC 启动寄存器
ADC_AWDCR0	ADC_BASE + 0x08	ADC 模拟看门狗配置寄存器 0
ADC_AWDVTH	ADC_BASE + 0x0C	ADC 模拟看门狗高阈值寄存器
ADC_AWDVTL	ADC_BASE + 0x10	ADC 模拟看门狗低阈值寄存器
ADC_SQR0TRIGGER	ADC_BASE + 0x14	ADC 序列 0 触发配置寄存器
ADC_SQR0SMPCFR	ADC_BASE + 0x18	ADC 序列 0 采样周期配置寄存器
ADC_SQR0CFR	ADC_BASE + 0x1C	ADC 序列 0 转换序列配置寄存器
ADC_SQR0RESULT0	ADC_BASE + 0x20	ADC 序列 0 转换结果 0 寄存器
ADC_SQR0RESULT1	ADC_BASE + 0x24	ADC 序列 0 转换结果 1 寄存器
ADC_SQR0RESULT2	ADC_BASE + 0x28	ADC 序列 0 转换结果 2 寄存器
ADC_SQR0RESULT3	ADC_BASE + 0x2C	ADC 序列 0 转换结果 3 寄存器
ADC_SQR1TRIGGER	ADC_BASE + 0x30	ADC 序列 1 触发配置寄存器
ADC_SQR1SMPCFR	ADC_BASE + 0x34	ADC 序列 1 采样周期配置寄存器
ADC_SQR1CFR	ADC_BASE + 0x38	ADC 序列 1 转换序列配置寄存器
ADC_SQR1RESULT0	ADC_BASE + 0x3C	ADC 序列 1 转换结果 0 寄存器
ADC_SQR1RESULT1	ADC_BASE + 0x40	ADC 序列 1 转换结果 1 寄存器
ADC_SQR1RESULT2	ADC_BASE + 0x44	ADC 序列 1 转换结果 2 寄存器
ADC_SQR1RESULT3	ADC_BASE + 0x48	ADC 序列 1 转换结果 3 寄存器
ADC_SQR2TRIGGER	ADC_BASE + 0x4C	ADC 序列 2 触发配置寄存器
ADC_SQR2SMPCFR	ADC_BASE + 0x50	ADC 序列 2 采样周期配置寄存器
ADC_SQR2CFR	ADC_BASE + 0x54	ADC 序列 2 转换序列配置寄存器
ADC_SQR2RESULT0	ADC_BASE + 0x58	ADC 序列 2 转换结果 0 寄存器
ADC_SQR2RESULT1	ADC_BASE + 0x5C	ADC 序列 2 转换结果 1 寄存器
ADC_SQR2RESULT2	ADC_BASE + 0x60	ADC 序列 2 转换结果 2 寄存器
ADC_SQR2RESULT3	ADC_BASE + 0x64	ADC 序列 2 转换结果 3 寄存器
ADC_SQR3TRIGGER	ADC_BASE + 0x68	ADC 序列 3 触发配置寄存器
ADC_SQR3SMPCFR	ADC_BASE + 0x6C	ADC 序列 3 采样周期配置寄存器
ADC_SQR3CFR	ADC_BASE + 0x70	ADC 序列 0 转换序列配置寄存器
ADC_SQR3RESULT0	ADC_BASE + 0x74	ADC 序列 3 转换结果 0 寄存器
ADC_SQR3RESULT1	ADC_BASE + 0x78	ADC 序列 3 转换结果 1 寄存器
ADC_SQR3RESULT2	ADC_BASE + 0x7C	ADC 序列 3 转换结果 2 寄存器
ADC_SQR3RESULT3	ADC_BASE + 0x80	ADC 序列 3 转换结果 3 寄存器
ADC_IER	ADC_BASE + 0x84	ADC 中断使能寄存器
ADC_ICR	ADC_BASE + 0x88	ADC 中断标志清除寄存器
ADC_ISR	ADC_BASE + 0x8C	ADC 中断标志寄存器

ADC_DCOFFSET	ADC_BASE + 0x90	ADC 直流偏置寄存器
ADC_AMC	ADC_BASE + 0x94	ADC 增益校准寄存器
ADC_AWDCR1	ADC_BASE + 0x98	ADC 模拟看门狗配置寄存器 1

## 19.6 寄存器描述

有关寄存器描述里所使用的缩写，请参见 1 文档约定章节。

### 19.6.1 ADC\_CR 控制寄存器

偏移地址: 请参见表 19-7 ADC 寄存器列表      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-8	RFU	-	保留位，请保持默认值
7-6	CLK	RW	ADC 工作时钟配置 00: PCLK / 2 01: PCLK / 2 10: PCLK / 4 11: PCLK / 8
5-4	RANGE	RW	ADC 量程选择 00: 2.4V 01: 3.6V 10: VDDA 11: 1.5 * VDDA
3	RFU	-	保留位，请保持默认值
2	TSEN	RW	内置温度传感器使能控制 0: 禁止内置温度传感器 1: 使能内置温度传感器
1	INREFEN	RW	内置参考电压使能控制 0: 禁止内置参考电压 1: 使能内置参考电压
0	EN	RW	ADC 使能控制 0: 禁止 ADC 1: 使能 ADC

## 19.6.2 ADC\_START 启动寄存器

偏移地址: 请参见表 19-7 ADC 寄存器列表      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-1	RFU	-	保留位, 请保持默认值
0	START	RW1	ADC 启动转换控制 0: 停止 ADC 转换 1: 启动 ADC 转换 注: 启动后硬件自动清零

## 19.6.3 ADC\_SQRxTRIGGER 序列 x 触发配置寄存器(x=0..3)

偏移地址: 请参见表 19-7 ADC 寄存器列表      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-21	RFU	-	保留位, 请保持默认值
20	VC1TRGO	RW	VC1OUT 上升沿触发 ADC 启动 0: 禁止 1: 使能
19	VC0TRGO	RW	VC0OUT 上升沿触发 ADC 启动 0: 禁止 1: 使能
18	GPIORTRGO	RW	GPIO 上升沿触发 ADC 启动 0: 禁止 1: 使能
17	GPIOFTRGO	RW	GPIO 下降沿触发 ADC 启动 0: 禁止 1: 使能
16	BTIM2TRGO	RW	BTIM2TRGO 信号触发 ADC 启动 0: 禁止 1: 使能
15	BTIM1TRGO	RW	BTIM1TRGO 信号触发 ADC 启动 0: 禁止 1: 使能
14	BTIM0TRGO	RW	BTIM0TRGO 信号触发 ADC 启动 0: 禁止 1: 使能
13	GTIMCC4	RW	GTIM 输出比较 CC4 信号触发 ADC 启动 0: 禁止 1: 使能
12	GTIMCC3	RW	GTIM 输出比较 CC3 信号触发 ADC 启动 0: 禁止 1: 使能

11	GTIMCC2	RW	GTIM 输出比较 CC2 信号触发 ADC 启动 0: 禁止 1: 使能
10	GTIMCC1	RW	GTIM 输出比较 CC1 信号触发 ADC 启动 0: 禁止 1: 使能
9	GTIMTRGO	RW	GTIMTRGO 信号触发 ADC 启动 0: 禁止 1: 使能
8	ATIMCC6	RW	ATIM 输出比较 CC6 信号触发 ADC 启动 0: 禁止 1: 使能
7	ATIMCC5	RW	ATIM 输出比较 CC5 信号触发 ADC 启动 0: 禁止 1: 使能
6	ATIMCC4	RW	ATIM 输出比较 CC4 信号触发 ADC 启动 0: 禁止 1: 使能
5	ATIMCC3	RW	ATIM 输出比较 CC3 信号触发 ADC 启动 0: 禁止 1: 使能
4	ATIMCC2	RW	ATIM 输出比较 CC2 信号触发 ADC 启动 0: 禁止 1: 使能
3	ATIMCC1	RW	ATIM 输出比较 CC1 信号触发 ADC 启动 0: 禁止 1: 使能
2	ATIMTRGO2	RW	ATIMTRGO2 信号触发 ADC 启动 0: 禁止 1: 使能
1	ATIMTRGO	RW	ATIMTRGO 信号触发 ADC 启动 0: 禁止 1: 使能
0	ADCSTART	RW	ADCSTART 信号触发 ADC 启动 0: 禁止 1: 使能

## 19.6.4 ADC\_SQRxSMPCFR 序列 x 采样周期配置寄存器(x=0..3)

偏移地址:请参见表 19-7 ADC 寄存器列表      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位, 请保持默认值
15-12	CH3	RW	ADC 序列 x 通道 3 采样时间配置 参见 CH0 位域
11-8	CH2	RW	ADC 序列 x 通道 2 采样时间配置 参见 CH0 位域
7-4	CH1	RW	ADC 序列 x 通道 1 采样时间配置 参见 CH0 位域
3-0	CH0	RW	ADC 序列 x 通道 0 采样时间配置 0000: 5 个 ADCCLK 时钟周期    1000: 54 个 ADCCLK 时钟周期 0001: 7 个 ADCCLK 时钟周期    1001: 70 个 ADCCLK 时钟周期 0010: 9 个 ADCCLK 时钟周期    1010: 102 个 ADCCLK 时钟周期 0011: 12 个 ADCCLK 时钟周期    1011: 134 个 ADCCLK 时钟周期 0100: 18 个 ADCCLK 时钟周期    1100: 166 个 ADCCLK 时钟周期 0101: 24 个 ADCCLK 时钟周期    1101: 198 个 ADCCLK 时钟周期 0110: 30 个 ADCCLK 时钟周期    1110: 262 个 ADCCLK 时钟周期 0111: 42 个 ADCCLK 时钟周期    1111: 390 个 ADCCLK 时钟周期 注 : 转换时间 = 采样时间 + 逐次比较时间(固定 12 个 ADCCLK)。

### 19.6.5 ADC\_SQRxCFR 序列 x 转换序列配置寄存器 (x=0..3)

偏移地址:请参见表 19-7 ADC 寄存器列表

复位值: 0x0000 0000

位编号	位符号	权限	说明
31-27	RFU	-	保留位, 请保持默认值
26-24	ENS	RW	待转换的序列配置 000: 关闭序列 001: 转换 CH0 010: 转换 CH0-CH1 011: 转换 CH0-CH2 100: 转换 CH0-CH3
23-18	CH3	RW	序列 x 待转换通道 3 配置 详见 CH0
17-12	CH2	RW	序列 x 待转换通道 2 配置 详见 CH0
11-6	CH1	RW	序列 x 待转换通道 1 配置 详见 CH0
5-0	CH0	RW	序列 x 待转换通道 0 配置 00 0000: AIN0 00 0001: AIN1 00 0010: AIN2 ... 01 1101: AIN29 10 0001: 1.2V 基准电压 10 0010: VDDA/3 10 0011: 温度传感器 10 0100: DACOUT 10 0101: VSSA 11 0000: FDA0OUT 11 0001: FDA1OUT 11 0010: FDA2OUT 其他: AIN0

### 19.6.6 ADC\_SQRxRESULTy 序列 x 转换结果 y 寄存器 (x=0..3, y=0..3)

偏移地址: 请参见表 19-7 ADC 寄存器列表

复位值: 0x0000 0000

位编号	位符号	权限	说明
31-12	RFU	-	保留位, 请保持默认值
11-0	RESULT	RO	ADC 转换结果寄存器

注: 最高位为补码

## 19.6.7 ADC\_AWDCR0 模拟看门狗配置寄存器

偏移地址: 请参见表 19-7 ADC 寄存器列表      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-25	RFU	-	保留位, 请保持默认值
24	FDA2OUT	RW	模拟看门狗监测 FDA2OUT 使能控制 0: 禁止 1: 使能
23	FDA1OUT	RW	模拟看门狗监测 FDA1OUT 使能控制 0: 禁止 1: 使能
22	FDA0OUT	RW	模拟看门狗监测 FDA0OUT 使能控制 0: 禁止 1: 使能
21-20	RFU	-	保留位, 请保持默认值
19	TS	RW	模拟看门狗监测 温度传感器 使能控制 0: 禁止 1: 使能
18	VDD/3	RW	模拟看门狗监测 VDD/3 使能控制 0: 禁止 1: 使能
17	CORE1V2	RW	模拟看门狗监测 1.2V 基准电压 使能控制 0: 禁止 1: 使能
16	RFU	-	保留位, 请保持默认值
15	IN15	RW	模拟看门狗监测 ADC_IN15 使能控制 0: 禁止 1: 使能
14	IN14	RW	模拟看门狗监测 ADC_IN14 使能控制 0: 禁止 1: 使能
13	IN13	RW	模拟看门狗监测 ADC_IN13 使能控制 0: 禁止 1: 使能
12	IN12	RW	模拟看门狗监测 ADC_IN12 使能控制 0: 禁止 1: 使能
11	IN11	RW	模拟看门狗监测 ADC_IN11 使能控制 0: 禁止 1: 使能
10	IN10	RW	模拟看门狗监测 ADC_IN10 使能控制 0: 禁止 1: 使能

9	IN9	RW	模拟看门狗监测 ADC_IN9 使能控制 0: 禁止 1: 使能
8	IN8	RW	模拟看门狗监测 ADC_IN8 使能控制 0: 禁止 1: 使能
7	IN7	RW	模拟看门狗监测 ADC_IN7 使能控制 0: 禁止 1: 使能
6	IN6	RW	模拟看门狗监测 ADC_IN6 使能控制 0: 禁止 1: 使能
5	IN5	RW	模拟看门狗监测 ADC_IN5 使能控制 0: 禁止 1: 使能
4	IN4	RW	模拟看门狗监测 ADC_IN4 使能控制 0: 禁止 1: 使能
3	IN3	RW	模拟看门狗监测 ADC_IN3 使能控制 0: 禁止 1: 使能
2	IN2	RW	模拟看门狗监测 ADC_IN2 使能控制 0: 禁止 1: 使能
1	IN1	RW	模拟看门狗监测 ADC_IN1 使能控制 0: 禁止 1: 使能
0	IN0	RW	模拟看门狗监测 ADC_IN0 使能控制 0: 禁止 1: 使能

### 19.6.8 ADC\_AWDCR1 模拟看门狗配置寄存器

偏移地址: 请参见表 19-7 ADC 寄存器列表      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-14	RFU	-	保留位, 请保持默认值
13	IN29	RW	模拟看门狗监测 ADC_IN29 使能控制 0: 禁止 1: 使能
12	IN28	RW	模拟看门狗监测 ADC_IN28 使能控制 0: 禁止 1: 使能



11	IN27	RW	模拟看门狗监测 ADC_IN27 使能控制 0: 禁止 1: 使能
10	IN26	RW	模拟看门狗监测 ADC_IN26 使能控制 0: 禁止 1: 使能
9	IN25	RW	模拟看门狗监测 ADC_IN25 使能控制 0: 禁止 1: 使能
8	IN24	RW	模拟看门狗监测 ADC_IN24 使能控制 0: 禁止 1: 使能
7	IN23	RW	模拟看门狗监测 ADC_IN23 使能控制 0: 禁止 1: 使能
6	IN22	RW	模拟看门狗监测 ADC_IN22 使能控制 0: 禁止 1: 使能
5	IN21	RW	模拟看门狗监测 ADC_IN21 使能控制 0: 禁止 1: 使能
4	IN20	RW	模拟看门狗监测 ADC_IN20 使能控制 0: 禁止 1: 使能
3	IN19	RW	模拟看门狗监测 ADC_IN19 使能控制 0: 禁止 1: 使能
2	IN18	RW	模拟看门狗监测 ADC_IN18 使能控制 0: 禁止 1: 使能
1	IN17	RW	模拟看门狗监测 ADC_IN17 使能控制 0: 禁止 1: 使能
0	IN16	RW	模拟看门狗监测 ADC_IN16 使能控制 0: 禁止 1: 使能

### 19.6.9 ADC\_AWDVTH 模拟看门狗高阈值寄存器

偏移地址: 请参见表 19-7 ADC 寄存器列表      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-12	RFU	-	保留位, 请保持默认值
11-0	VTH	RW	模拟看门狗检测高阈值

注: 该寄存器 BIT11 位为补码

## 19.6.10 ADC\_AWDVTL 模拟看门狗低阈值寄存器

偏移地址: 请参见表 19-7 ADC 寄存器列表      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-12	RFU	-	保留位, 请保持默认值
11-0	VTL	RW	模拟看门狗检测低阈值

注: 该寄存器 BIT11 位为补码

## 19.6.11 ADC\_IER 中断使能寄存器

偏移地址: 请参见表 19-7 ADC 寄存器列表      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-8	RFU	-	保留位, 请保持默认值
8	TRIGERR	RW	ADC 触发发生在非空闲状态中断使能控制 0: 禁止 1: 使能
7	AWDTR	RW	模拟看门狗区间中断使能控制 0: 禁止 1: 使能
6	AWDTH	RW	模拟看门上阈值中断使能控制 0: 禁止 1: 使能
5	AWDTL	RW	模拟看门狗下阈值中断使能控制 0: 禁止 1: 使能
4	EOS3	RW	序列 3 转换完成中断使能控制 0: 禁止 1: 使能
3	EOS2	RW	序列 2 转换完成中断使能控制 0: 禁止 1: 使能
2	EOS1	RW	序列 1 转换完成中断使能控制 0: 禁止 1: 使能
1	EOS0	RW	序列 0 转换完成中断使能控制 0: 禁止 1: 使能
0	EOC	RW	单次转换完成中断使能控制 0: 禁止 1: 使能

## 19.6.12 ADC\_ISR 中断标志寄存器

偏移地址: 请参见表 19-7 ADC 寄存器列表 复位值: 0x0000 0000

位编号	位符号	权限	说明
31-8	RFU	-	保留位, 请保持默认值
8	TRIGERR	RO	ADC 触发发生在非空闲状态中断标志 0: ADC 触发未发生在非空闲状态 1: ADC 触发发生在非空闲状态
7	WDTR	RO	模拟看门狗区间标志 0: 转换结果位于 [ADC_VTL , ADC_VTH] 区间外 1: 转换结果位于 [ADC_VTL , ADC_VTH] 区间内
6	WDTH	RO	模拟看门狗上阈值标志 0: 转换结果位于 [ADC_VTH , 2047] 区间外 1: 转换结果位于 [ADC_VTH , 2047] 区间内
5	WDTL	RO	模拟看门狗下阈值标志 0: 转换结果位于 [-2047 , ADC_VTL] 区间外 1: 转换结果位于 [-2047 , ADC_VTL] 区间内
4	EOS3	RO	序列 3 转换完成标志 0: 序列 3 转换未完成 1: 序列 3 转换已完成
3	EOS2	RO	序列 2 转换完成标志 0: 序列 2 转换未完成 1: 序列 2 转换已完成
2	EOS1	RO	序列 1 转换完成标志 0: 序列 1 转换未完成 1: 序列 1 转换已完成
1	EOS0	RO	序列 0 转换完成标志 0: 序列 0 转换未完成 1: 序列 0 转换已完成
0	EOC	RO	转换完成标志 0: 一次 ADC 转换未完成 1: 一次 ADC 转换已完成

## 19.6.13 ADC\_ICR 中断标志清除寄存器

偏移地址: 请参见表 19-7 ADC 寄存器列表 复位值: 0x0000 01FF

位编号	位符号	权限	说明
31-9	RFU	-	保留位, 请保持默认值
8	TRIGERR	R1W0	ADC 触发发生在非空闲状态中断标志清除 0 控制 W0: 清除 ISR 寄存器中的相应标志 W1: 无功能
7	WDTR	R1W0	模拟看门狗区间标志清除 0 控制 W0: 清除 ISR 寄存器中的相应标志 W1: 无功能

6	WDTH	R1W0	模拟看门狗上阈值标志清 0 控制 W0: 清除 ISR 寄存器中的相应标志 W1: 无功能
5	WDTL	R1W0	模拟看门狗下阈值标志清 0 控制 W0: 清除 ISR 寄存器中的相应标志 W1: 无功能
4	EOS3	R1W0	序列 3 转换完成标志清 0 控制 W0: 清除 ISR 寄存器中的相应标志 W1: 无功能
3	EOS2	R1W0	序列 2 转换完成标志清 0 控制 W0: 清除 ISR 寄存器中的相应标志 W1: 无功能
2	EOS1	R1W0	序列 1 转换完成标志清 0 控制 W0: 清除 ISR 寄存器中的相应标志 W1: 无功能
1	EOS0	R1W0	序列 0 转换完成标志清 0 控制 W0: 清除 ISR 寄存器中的相应标志 W1: 无功能
0	EOC	R1W0	转换完成标志清 0 控制 W0: 清除 ISR 寄存器中的相应标志 W1: 无功能

#### 19.6.14 ADC\_DCOFFSET 直流偏置寄存器

偏移地址: 请参见表 19-7 ADC 寄存器列表      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-8	RFU	-	保留位, 请保持默认值
7-0	DCOFFSET	RW	采样电路增益直流偏置

注: 该寄存器 BIT7 位为补码, 8'b 1000 0001 - 8'b 0111 1111

#### 19.6.15 ADC\_AMC 增益校准寄存器

偏移地址: 请参见表 19-7 ADC 寄存器列表      复位值: 0x0000 0200

位编号	位符号	权限	说明
31-10	RFU	-	保留位, 请保持默认值
9-0	AMC	RW	ADC 采样电路增益校正

## 20 数模转换器（DAC）

### 20.1 概述

HC32F0653 有 1 个 8 位的数模转换器 DAC，它们可以输出到模拟电压比较器 VCx 和全差分运算放大器 FDA 内部。

### 20.2 主要特性

- 1 个 DAC 转换器
- 8 位的数模转换分辨率
- 2 路参考电压源（Vref）
  - 内置 1.2V 参考电压
  - 内置 4.8V 参考电压

### 20.3 功能框图

DAC 功能框图如下图所示：

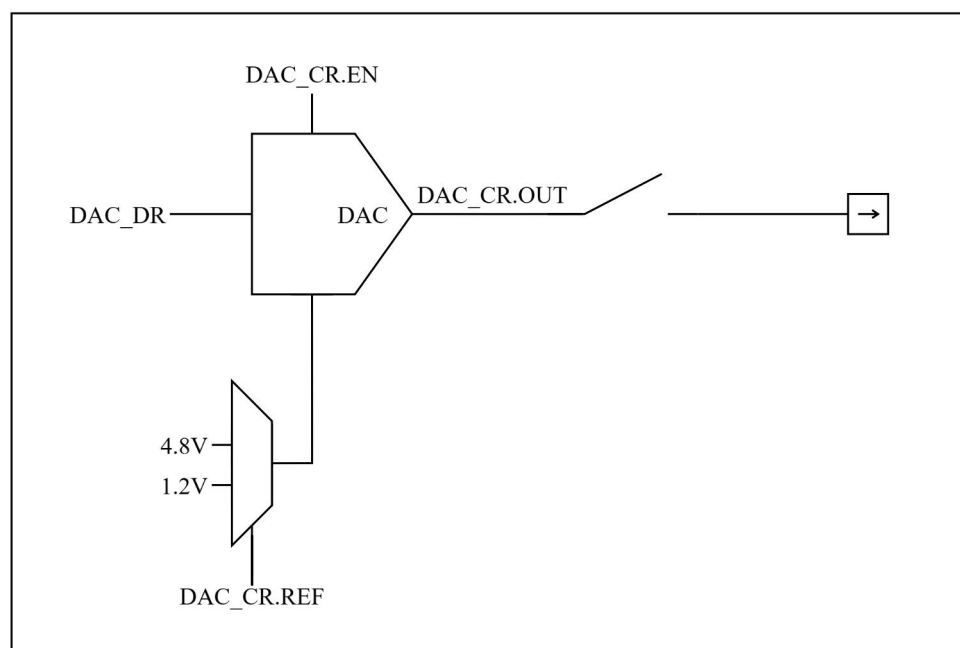


图 20-1 DAC 功能框图

## 20.4 功能描述

### 20.4.1 DAC 使能

一旦 DAC\_CR 寄存器的"EN"位被置 1，DAC 使能。

DAC 上电延迟一段时间后、待参考源处于稳定状态，DAC 将立即转换 DAC\_CR 寄存器的"DATA"位中的数据，经过一段设置时间后，DAC 转换完成。

通过清除 DAC\_CR 寄存器的"EN"位可以停止转换，并将 DAC 置于断电模式，在该模式，DAC 几乎不耗电。

### 20.4.2 DAC 转换

一个 8bit 的符号位"DATA[7:0]"用于配置 DAC 数据，取值范围为 0~255。

$DACOUT = (DAC\_DR / 255) * DAC\_CR.REF$

## 20.5 寄存器列表

DAC 基地址：DAC\_BASE = 0x4001 3000

表 20-1 DAC 寄存器列表

寄存器名称	寄存器地址	寄存器描述
DAC_CR	DAC_BASE + 0x00	DAC 控制寄存器
DAC_DR	DAC_BASE + 0x04	DAC 数据寄存器

## 20.6 寄存器描述

### 20.6.1 DAC\_CR 控制寄存器

偏移地址: 0x00      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-4	RFU	-	保留位, 请保持默认值
3	OUT	RW	DACO 信号是否输出到 IO 0: 失能 DAC 信号输出到 IO 1: 使能 DAC 信号输出到 IO
2	RFU	-	保留位, 请保持默认值
1	REF	RW	DAC 参考电压选择寄存器 0: 1.2V 1: 4.8V
0	EN	RW	DAC 模块电源控制位 0: 关闭 DAC 电源 1: 打开 DAC 电源

### 20.6.2 DAC\_DR 数据寄存器

偏移地址: 0x04      复位值: 0x0000 0000

位编号	位符号	权限	说明
31-8	RFU	-	保留位, 请保持默认值
7-0	DATA	RW	DAC 数据寄存器 $DACOUT = (DAC\_DR / 255) * DAC\_CR.REF$

## 21 全差分运算放大器（FDA）

### 21.1 概述

芯片集成 1 路输入输出轨到轨全差分运算放大器(FDA)，内置反馈电阻，外部引脚上需串联一个电阻到信号源。反馈电阻的阻值可通过寄存器设置，以实现不同的放大倍数。输出可连接到内部 ADC。

### 21.2 主要特性

- 轨到轨输入和输出电压范围
- 低输入偏置电流
- 低输入失调电压
- 20 MHz 增益带宽积
- 高压摆率

### 21.3 功能描述

#### 21.3.1 功能框图

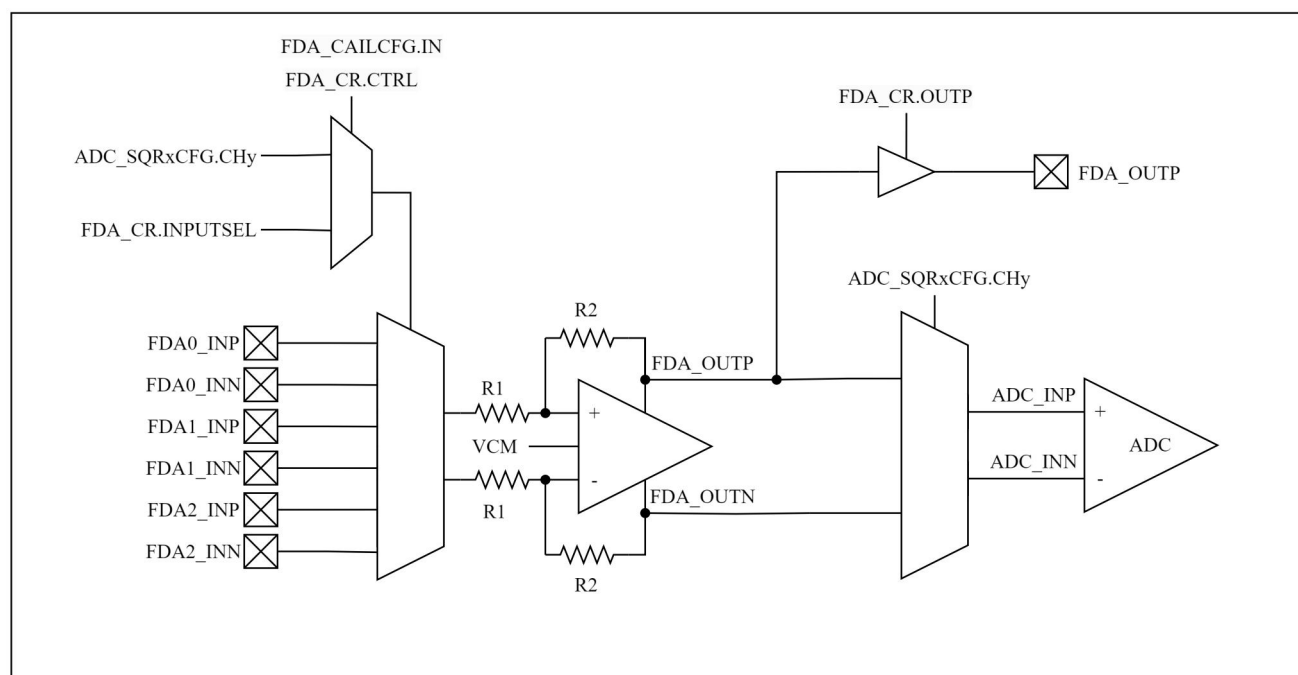


图 21-1FDA 功能框图

#### 21.3.2 外部电阻配置要求

- 输入引脚 FDA<sub>x</sub>\_INP 与 FDA<sub>x</sub>\_INN 需配置外部匹配电阻 (R<sub>0</sub>)，且两电阻阻值必须严格匹配。
- 放大器闭环增益公式：

$$A_u = \frac{R_2}{R_1 + R_0}$$



其中 (R<sub>1</sub>) 为内部反馈电阻, (R<sub>2</sub>) 由寄存器 FDA\_CR.CKD[7:6] 配置。

### 21.3.3 应用场景电阻选型

- MOS 管电阻采样:

当 MOS 管关断时, 输入信号可能瞬态升至数十伏电源电压。为限制输入电流, 外部电阻 (R<sub>0</sub>) 需满足:

- 分流电阻采样:

推荐 (R<sub>0</sub>) 取值范围:

$$100\Omega \leq R_0 \leq 2K\Omega$$

### 21.3.4 输入滤波设计

- 可在 FDA<sub>x</sub>\_INP 与 FDA<sub>x</sub>\_INN 间串联电容 (C<sub>0</sub>), 形成一阶 RC 滤波器:
- 设计建议:
  - 若信号噪声显著, 按目标 (f<sub>c</sub>) 选取 (C<sub>0</sub>);
  - 若需高带宽 (快速响应) 或低噪声环境, 可省略 (C<sub>0</sub>)。

### 21.3.5 输出缓冲与控制

通过寄存器 FDA\_CR.OUT[15] 使能输出缓冲器 (BUFFER), 将由 FDA\_CR.INPUTSEL[14:13] 控制选定放大器通道信号路由至 FDAOUT 引脚。

## 21.4 寄存器列表

FDA 基地址: FDA\_BASE = 0x4001 2C00

表 21-1FDA 寄存器列表

寄存器名称	寄存器地址	寄存器描述
FDA_CR	FDA_BASE + 0x00	FDA 控制寄存器
FDA_CALICFG	FDA_BASE + 0x04	FDA 校准配置寄存器

## 21.5 寄存器描述

### 21.5.1 FDA 控制寄存器 (FDA\_CR)

地址：请参见表 21-1FDA 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-16	RFU	-	保留位，请保持默认值
15	OUT	RW	使能 FDAOUT 端口连接控制。 0：不连通到 FDAOUT 引脚 1：连通到 FDAOUT 引脚
14-13	INPUTSEL	RW	FDA 输入引脚选择 00：FDA0 01：FDA1 10：FDA2 11：FDA2 <b>注：读取该位时是实际设置的 FDA 通道</b>
12	CTRL	RW	FDA 输入通道切换控制选择 0：FDA 1：ADC
11-8	RFU	-	保留位，请保持默认值
7-6	CKD	RW	运放反馈电阻 00: 200k:10k 01: 190k:20k 10: 180k:30k 11: 170k:40k
5-1	RFU	-	保留位，请保持默认值
0	EN	RW	FDA 模块使能位 0：关闭 FDA 1：打开 FDA

### 21.5.2 FDA 校准配置寄存器(FDA\_CALICFG)

地址：请参见表 21-1FDA 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-6	RFU	-	保留位，请保持默认值
5	CALOV	RO	FDA 校准完成标志位
4	FIN	RW	FDA 输入浮空 0：FDA 输入端连接到引脚 1：FDA 输入端不连接到引脚
3-1	TRIM	RW	FDA 输入失调电压校准值
0	TRIMEN	RW	FDA 输入失调电压校准控制位

## 22 模拟电压比较器（VC）

### 22.1 概述

HC32F0653 内部集成 2 个模拟电压比较器（VC），用于比较两路模拟输入电压，并将比较结果从引脚输出。电压比较器的正负端既支持 4 路外部模拟输入，又支持内部 1/3 VDDA、DAC，内部温度传感器等电压参考。比较结果输出具有滤波功能、迟滞窗口功能，BLANK 窗口功能，以及极性选择。支持比较中断，可用于低功耗模式下唤醒 MCU。

### 22.2 主要特性

- 双路的模拟电压比较器 VC0、VC1
- 多达 5 路外部模拟信号输入
- 7 路片内模拟输入信号
  - DAC 输出电压
  - 内置温度传感器输出电压
  - 1/3 VDDA
  - 1.2V 内核电压基准源
  - VCx\_MID
- 可选择输出极性
- 支持迟滞窗口比较功能
- 可编程的滤波器和滤波时间
- BLANK 窗口功能
- 3 种中断触发方式，可组合使用
  - 高电平触发
  - 上升沿触发
  - 下降沿触发
- 支持低功耗模式下运行，中断唤醒 MCU

## 22.3 功能描述

### 22.3.1 功能框图

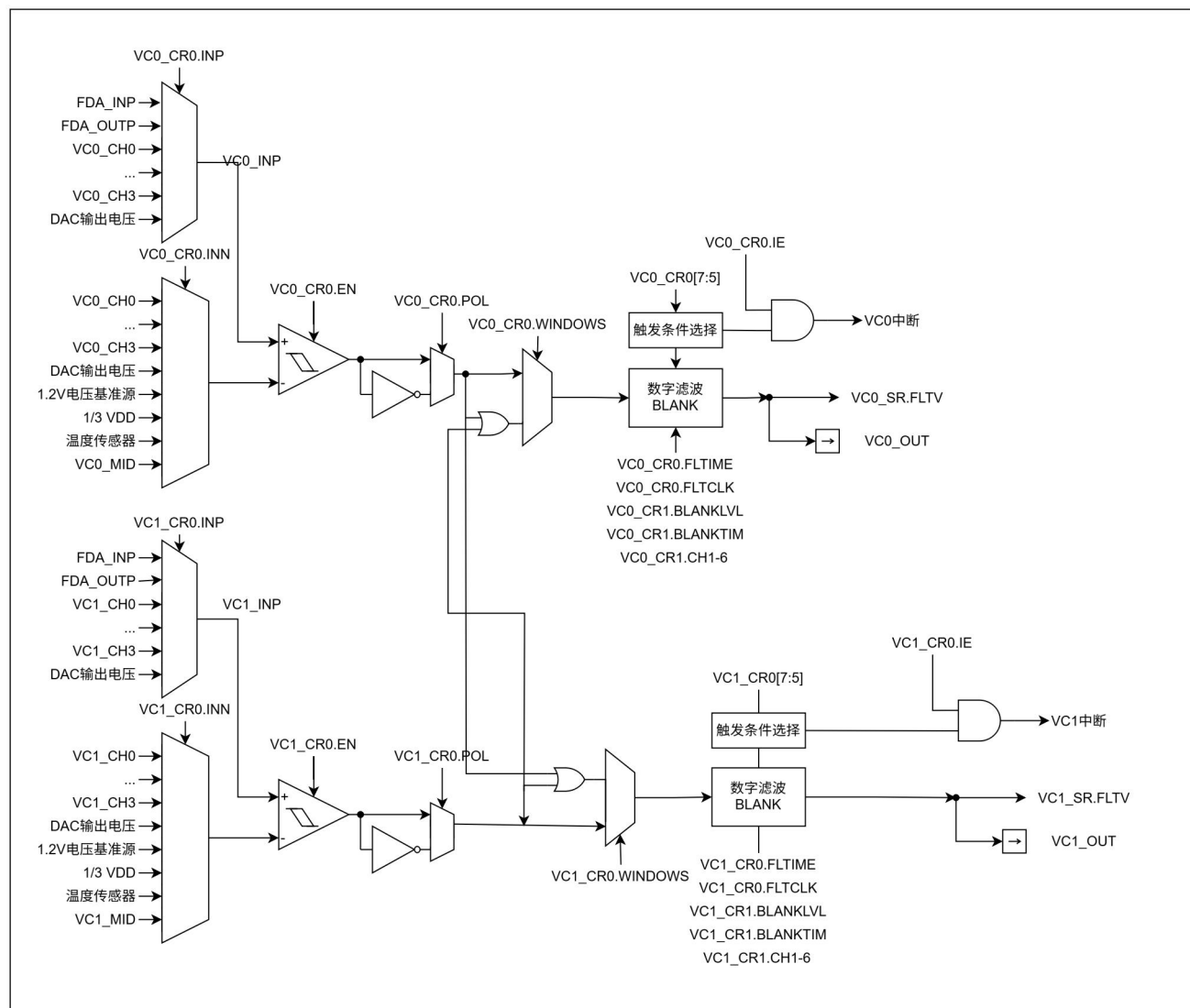


图 22-1VC 功能框图

VC0、VC1 的正负端输入选择，由控制寄存器 VCx\_CR0 的 INP、INN 位域选择，如下表所示：

表 22-1VC 正负端输入信号配置

VCx_CR0.INP	VCx 正端输入信号	VCx_CR0.INN	VCx 负端输入信号
000	VCx_CH0	0000	VCx_CH0
001	VCx_CH1	0001	VCx_CH1
010	VCx_CH2	0010	VCx_CH2
011	VCx_CH3	0011	VCx_CH3
100	VCx_CH4	0100	VCx_CH4
101	DAC 输出电压	0101	DAC 输出电压
110	FDA_INP	0110	1.2V 电压基准源
111	FDA_OUTP	0111	1/3 VDDA 电源电压
-	-	1000	内置温度传感器输出电压
		1001	VCx_MID

## 22.3.2 输入输出引脚

模拟电压比较器支持 5 路外部模拟信号输入，用户必须将对应 GPIO 端口配置为模拟功能（GPIOx\_ANALOG.PINy = 1）。模拟电压比较器支持将比较结果从引脚输出，用户必须将对应 GPIO 端口配置为数字输出，同时选择功能复用。

VC0、VC1 支持的输入输出引脚如下表所示：

表 22-2 VC 输入输出引脚配置

VC0 输入输出	GPIO	配置	VC1 输入输出	GPIO	配置
VC0_CH0	PA0	模拟	VC1_CH0	PA14	模拟
VC0_CH1	PA1	模拟	VC1_CH1	PB3	模拟
VC0_CH2	PA2	模拟	VC1_CH2	PB5	模拟
VC0_CH3	PA3	模拟	VC1_CH3	PB4	模拟
VC0_CH4	PA6	模拟	VC1_CH4	PB0	模拟
VC0_CH5	PA4	模拟	-	-	-

注意：VC0\_CH5 只能作为负端输入。

## 22.3.3 延迟/响应时间

设置控制寄存器 VCx\_CR0 的 EN 位域为 1，使能 VC 模块。

从 VC 使能或 VC 的正负两端输入电压变化，到电压比较器输出正确比较结果的时间，被定义为比较器的延迟响应时间。延迟/响应时间由控制寄存器 VCx\_CR0 的 RESP 位域配置，响应时间越短，VC 模块的功耗越大。

## 22.3.4 极性选择

电压比较器 VC0、VC1 的输出信号的极性，由控制寄存器 VCx\_CR0 的 POL 位域设置：

- POL 为 1，VCx\_OUTP 信号与 VCx\_OUTA 信号极性相反，即正端大于负端时 VCx 输出低电平
- POL 为 0，VCx\_OUTP 信号与 VCx\_OUTA 信号极性相同，即正端大于负端时 VCx 输出高电平

## 22.3.5 数字滤波

电压比较器内置的数字滤波器，用于对电压比较器的输出信号进行数字滤波，用户可使用滤波功能过滤系统噪声，比如马达停止时的大电流噪声等，避免比较器的噪声输出引起系统的误动作。

数字滤波器的时钟由控制寄存器 VCx\_CR1 的 FLTCLK 位域选择：

- FLTCLK 为 1，使用 PCLK 作为滤波时钟
- FLTCLK 为 0，使用 LSI 振荡器时钟作为滤波时钟

数字滤波器的滤波宽度由控制寄存器 VCx\_CR1 的 FLTTIME 位域选择，滤除宽度小于一定时钟周期的信号，有 8 级滤波宽度可选择。

电压比较器经数字滤波后的输出电平，可通过状态寄存器 VCx\_SR 的 FLTV 位域读出。

当用户设置了 VCx\_OUT 引脚为数字输出，且选择 VC 比较输出的复用功能时，VCx\_OUT 引脚将输出电压比较器经数字滤波后的输出电平。

VC 的滤波响应波形，如下图所示：

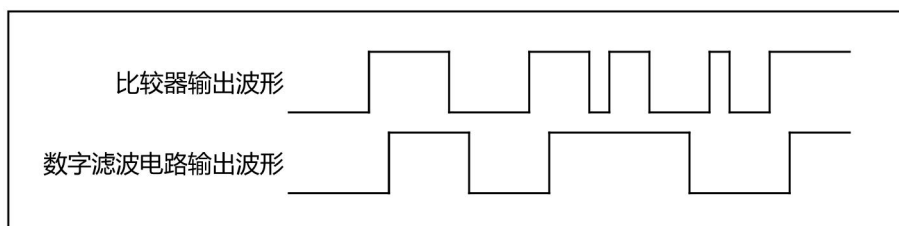


图 22-2 VC 滤波响应时间

电压比较器的输出可以作为高级定时器 (ATIM) 的刹车输入或 OCREF\_CLR 输入。此功能通常应用于电机保护场合。

### 22.3.6 窗口比较功能

模拟电压比较器支持窗口比较功能，可将 VC0 和 VC1 的比较结果进行异或操作后输出，由控制寄存器 VCx\_CR0 的 WINDOW 位域使能。

WINDOW 为 1 时，VCx\_OUTW 信号为 VC0\_OUTP 信号与 VC1\_OUTP 信号的异或值；

WINDOW 为 0 时，VCx\_OUTW 信号与 VCx\_OUTP 信号电平相同。

### 22.3.7 BLANK 窗口功能

在保持 VCx 模块工作的同时，如果想暂时停止电压比较功能，或者为避免某些应用系统（比如电机控制）中，被监测信号短时间的合理波动造成电压比较器的输出电平发生不必要的翻转，本芯片的电压比较器增加了 BLANK 窗口功能，即，当指定的外部触发条件启动 BLANK 窗口时，在设定的 BLANK 窗口期内，不进行电压比较，电压比较器输出设定的高或低电平。BLANK 窗口期之后，电压比较器恢复正常工作。如下表所示：

表 22-3 BLANK 窗口持续时间配置

VCx_CR1.BLANKFLT	BLANK 窗口 PCLK 时钟个数
000	4
001	8
010	16
011	32
100	64
101	128
110	256
111	512

BLANK 窗口持续期间电压比较器的输出电平由控制寄存器 VCx\_CR1 的 BLANKLVL 位域进行设置，BLANKLVL 为 0 输出低电平，BLANKLVL 为 1 则输出高电平。

BLANK 窗口的触发启动条件,由控制寄存器 VCx\_CR1 的 BLANKCH1、BLANKCH2、BLANKCH3、BLANKCH4、BLANKCH5、BLANKCH6 位域配置,分别由 ATIM 的 OC1REFC、OC2REFC、OC3REFC、OC4REFC、OC5REFC、OC6REFC 上升沿触发启动 BLANK 窗口。

## 22.4 VC 中断

HC32F0653 的电压比较器支持在低功耗模式下工作,比较中断可将芯片从低功耗模式下唤醒。

设置控制寄存器 VCx\_CR0 的 IE 位域为 1,使能 VCx 中断,产生中断时状态寄存器 VCx\_SR 的中断标志位 INTF 会被硬件置 1,用户可以向 INTF 位写 0,清除中断标志。

设置控制寄存器 VCx\_CR1 的 HIGHIE、RISEIE、FALLIE 位域,可选择不同的中断触发方式:

- HIGHIE 为 1, VCx\_OUT 输出信号高电平触发中断
- RISEIE 为 1, VCx\_OUT 输出信号上升沿触发中断
- FALLIE 为 1, VCx\_OUT 输出信号下降沿触发中断

## 22.5 编程示例

在此示例中,使用数字滤波功能,并配置比较中断,配置方法如下所示:

1. 配置 VCx\_CR0.INP, 选择正端待监测的电压来源;
2. 配置 VCx\_CR0.INN, 选择负端待监测的电压来源;
3. 配置 VCx\_CR1.FLTTIME, 选择滤波时间;
4. 配置 VCx\_CR1.FLTCLK, 选择滤波时钟;
5. 配置 VCx\_CR1.FLTEN 为 1, 使能 VCx 滤波;
6. 设置 VCx\_CR1 寄存器的 HIGHIE、RISEIE、FALLIE 为 1, 选择中断触发方式;
7. 设置 VCx\_CR0.IE 为 1, 使能 VCx 中断;
8. 设置 VCx\_CR0.EN 为 1, 使能 VCx;
9. 在 VC 模块初始化和中断程序中,对 VCx\_SR 的 INTF 位写入 0 清除中断标志后,允许 VCx 中断的产生。

## 22.6 寄存器列表

VC0 基地址: VC0\_BASE = 0x4001 2400

VC1 基地址: VC1\_BASE = 0x4001 2800

表 22-4VC 寄存器列表

寄存器名称	寄存器地址	寄存器描述
VCx_CR0	VCx_BASE+0x00	控制寄存器 0
VCx_CR1	VCx_BASE+0x04	控制寄存器 1
VCx_SR	VCx_BASE+0x08	状态寄存器



## 22.7 寄存器描述

有关寄存器描述里所使用的缩写，请参见 1 文档约定章节。

### 22.7.1 VCx\_CR0 控制寄存器 0

地址：请参见表 22-4VC 寄存器列表 复位值：0x0000 0000

位编号	位符号	权限	说明
31-13	RFU	-	保留位，请保持默认值
12-9	INN	RW	VCx 负端输入信号设置 0000: VCx_CH0 0001: VCx_CH1 0010: VCx_CH2 0011: VCx_CH3 0100: VCx_CH4 0101: DAC 输出电压 0110: 1.2V 内核电压基准源 0111: 1/3 VDDA 电源电压 1000: 内置温度传感器输出电压 <b>注：需使能 ADC_CR.TSEN</b> 1001: VCx_MID 其他：保留 <b>注：VCx_MID 电压等于 VCx_CH0, VCx_CH1, VCx_CH2 输入电压之和除以三</b> <b>注：配置 ADC_CR 寄存器，需先使能 ADC 外设时钟</b>
8-6	INP	RW	VCx 正端输入信号配置 000: VCx_CH0 001: VCx_CH1 010: VCx_CH2 011: VCx_CH3 100: VCx_CH4 101: DAC 输出电压 110: FDA_INP 111: FDA_OUTP
5	WINDOW	RW	窗口比较功能配置 0: 禁止窗口功能，VCx_OUTW 信号等于 VCx_OUTP 信号 1: 使能窗口功能，VCx_OUTW 信号等于 VC0_OUTP 信号与 VC1_OUTP 信号的异或值
4	POL	RW	窗口比较功能配置 0: 正端大于负端时 VCx 输出高电平 1: 正端大于负端时 VCx 输出低电平
3	IE	RW	VCx 中断使能配置 0: 禁止 1: 使能

2	HYS	RW	VCx 迟滞窗口配置 0: 没有迟滞 1: 迟滞窗口大约 20mV
1	RESP	RW	VCx 响应速度配置 0: 低速 1: 高速 注: 响应速度越快, 功耗越大
0	EN	RW	VCx 使能控制 0: 禁止 1: 使能

## 22.7.2 VCx\_CR1 控制寄存器 1

地址: 请参见表 22-4VC 寄存器列表 复位值: 0x0000 0000

位编号	位符号	权限	说明
31-18	RFU	-	保留位, 请保持默认值
17	BLANKLVL	RW	BLANK 窗口持续期间输出电平配置 0: 输出低电平 1: 输出高电平
16-14	BLANKTIME	RW	BLANK 窗口持续时间配置: $2^{(BLANKTIME+2)}$ 个 PCLK 周期 注: ATIM 的 OC1REFC、OC2REFC、OC3REFC、OC4REFC、OC5REFC、OC6REFC 任意一个上升沿都会触发启动 BLANK 窗口, 在 BLANK 窗口持续时间内, 不进行电压比较。
13	BLANKCH6	RW	ATIM_OC6REFC 上升沿触发 VCx 启动 BLANK 窗口配置 0: 禁止 1: 使能
12	BLANKCH5	RW	ATIM_OC5REFC 上升沿触发 VCx 启动 BLANK 窗口配置 0: 禁止 1: 使能
11	BLANKCH4	RW	ATIM_OC4REFC 上升沿触发 VCx 启动 BLANK 窗口配置 0: 禁止 1: 使能
10	BLANKCH3	RW	ATIM_OC3REFC 上升沿触发 VCx 启动 BLANK 窗口配置 0: 禁止 1: 使能
9	BLANKCH2	RW	ATIM_OC2REFC 上升沿触发 VCx 启动 BLANK 窗口配置 0: 禁止 1: 使能
8	BLANKCH1	RW	ATIM_OC1REFC 上升沿触发 VCx 启动 BLANK 窗口配置 0: 禁止 1: 使能
7	HIGHIE	RW	VCx 输出信号高电平触发中断使能

			0: 禁止 1: 使能
6	RISEIE	RW	VCx 输出信号上升沿触发中断使能 0: 禁止 1: 使能
5	FALLIE	RW	VCx 输出信号下降沿触发中断使能 0: 禁止 1: 使能
4	FLTCLK	RW	数字滤波模块滤波时钟设置 0: LSI 1: PCLK
3-0	FLTTIME	RW	数字滤波模块滤波时间配置 0000: 无滤波, 直接输出 0001: $f_{\text{SAMPLING}}=f_{\text{FLTCLK}}/1$ , $N=2$ 0010: $f_{\text{SAMPLING}}=f_{\text{FLTCLK}}/1$ , $N=4$ 0011: $f_{\text{SAMPLING}}=f_{\text{FLTCLK}}/1$ , $N=8$ 0100: $f_{\text{SAMPLING}}=f_{\text{FLTCLK}}/2$ , $N=6$ 0101: $f_{\text{SAMPLING}}=f_{\text{FLTCLK}}/2$ , $N=8$ 0110: $f_{\text{SAMPLING}}=f_{\text{FLTCLK}}/4$ , $N=6$ 0111: $f_{\text{SAMPLING}}=f_{\text{FLTCLK}}/4$ , $N=8$ 1000: $f_{\text{SAMPLING}}=f_{\text{FLTCLK}}/8$ , $N=6$ 1001: $f_{\text{SAMPLING}}=f_{\text{FLTCLK}}/8$ , $N=8$ 1010: $f_{\text{SAMPLING}}=f_{\text{FLTCLK}}/16$ , $N=5$ 1011: $f_{\text{SAMPLING}}=f_{\text{FLTCLK}}/16$ , $N=6$ 1100: $f_{\text{SAMPLING}}=f_{\text{FLTCLK}}/16$ , $N=8$ 1101: $f_{\text{SAMPLING}}=f_{\text{FLTCLK}}/32$ , $N=5$ 1110: $f_{\text{SAMPLING}}=f_{\text{FLTCLK}}/32$ , $N=6$ 1111: $f_{\text{SAMPLING}}=f_{\text{FLTCLK}}/32$ , $N=8$ 时钟分频, 然后计数 N 次, N 次都是同样的电平, 则输出该电平, 否则保持原样。

### 22.7.3 VCx\_SR 状态寄存器

地址: 请参见表 22-4VC 寄存器列表 复位值: 0x0000 0000

位编号	位符号	权限	说明
31-2	RFU	-	保留位, 请保持默认值
1	FLTV	RO	数字滤波器输出的电平值 0: 数字滤波器输出低电平 1: 数字滤波器输出高电平
0	INTF	RW0	中断标志 R0: 未发生 VC 中断 R1: 已发生 VC 中断 W0: 清除 VC 中断标志 W1: 无功能

## 23 版本记录

版本	日期	描述
Ver1.00	2025-4-7	第一版
Ver1.01	2025-5-15	修改部分错误
Ver1.02	2025-7-17	修改部分错误
Ver1.03	2025-7-29	修改部分错误

HOLYCHIP 公司保留对以下所有产品在可靠性、功能和设计方面的改进作进一步说明的权利。HOLYCHIP 不承担由本手册所涉及的产品或电路的运用和使用所引起的任何责任，HOLYCHIP 的产品不是专门设计来应用于外科植入、生命维持和任何 HOLYCHIP 产品产生的故障会对个体造成伤害甚至死亡的领域。如果将 HOLYCHIP 的产品用于上述领域，即使这些是由 HOLYCHIP 在产品设计和制造上的疏忽引起的，用户应赔偿所有费用、损失、合理的人身伤害或死亡所直接或间接所产生的律师费用，并且用户保证 HOLYCHIP 及其雇员、子公司、分支机构和销售商与上述事宜无关。

芯圣电子

2025 年 3 月