CIU32M011/M031

集成预驱型 32 位电机系列微控制器

参考手册

RM1004

北京中电华大电子设计有限责任公司
CEC Huada Electronic Design Co.,Ltd

声　明

北京中电华大电子设计有限责任公司保留未经通知用户对本手册内容进行修改的权利。虽然我们已经核对本手册的内容，但是差错有时候难以完全避免，所以，我们会对手册的内容进行定期的审查，并在下一版的文件中作必要的修改。建议您在最终设计前从华大电子获取本文档的最新版本。

目录

1. 简介

CIU32M010、CIU32M030 是基于 ARM Cortex-M0 内核的集成预驱型电机控制系列 MCU 产品，支持 SSOP24、LQFP32、LQFP48 封装，最高频率可达 72MHz，内部集成运算放大器、高速比较器、ADC、1 个 16 位高级定时器、6 个通用定时器、1 个基本定时器、1 个独立看门狗、1 个窗口看门狗、1 个 SysTick 定时器，还包含 2 个 SPI/I²C 接口、2 个 UART 接口。集成栅极驱动模块，可直接驱动三路 P/NMOS 功率器件。

CIU32M011、CIU32M031 电机控制系列 MCU 应用场景：

- 电动工具
- 高速风筒
- 风扇
- 风机和水泵等

2. 引脚分配与功能描述

2.1. 引脚分配图

图 2-1 SSOP24（CIU32M011H3HB、CIU32M031H5HB）

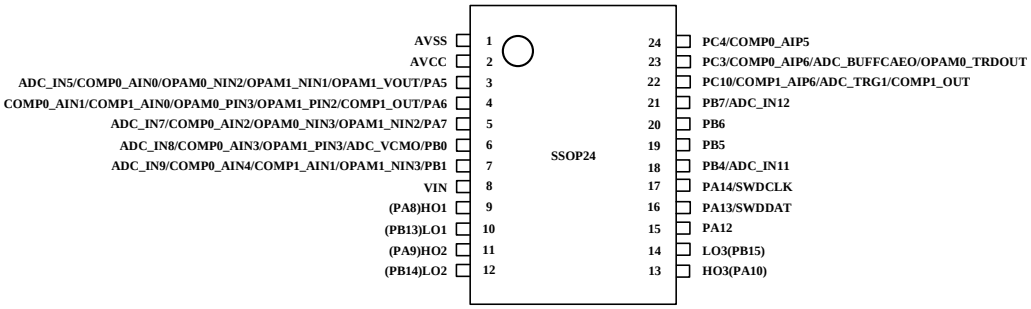
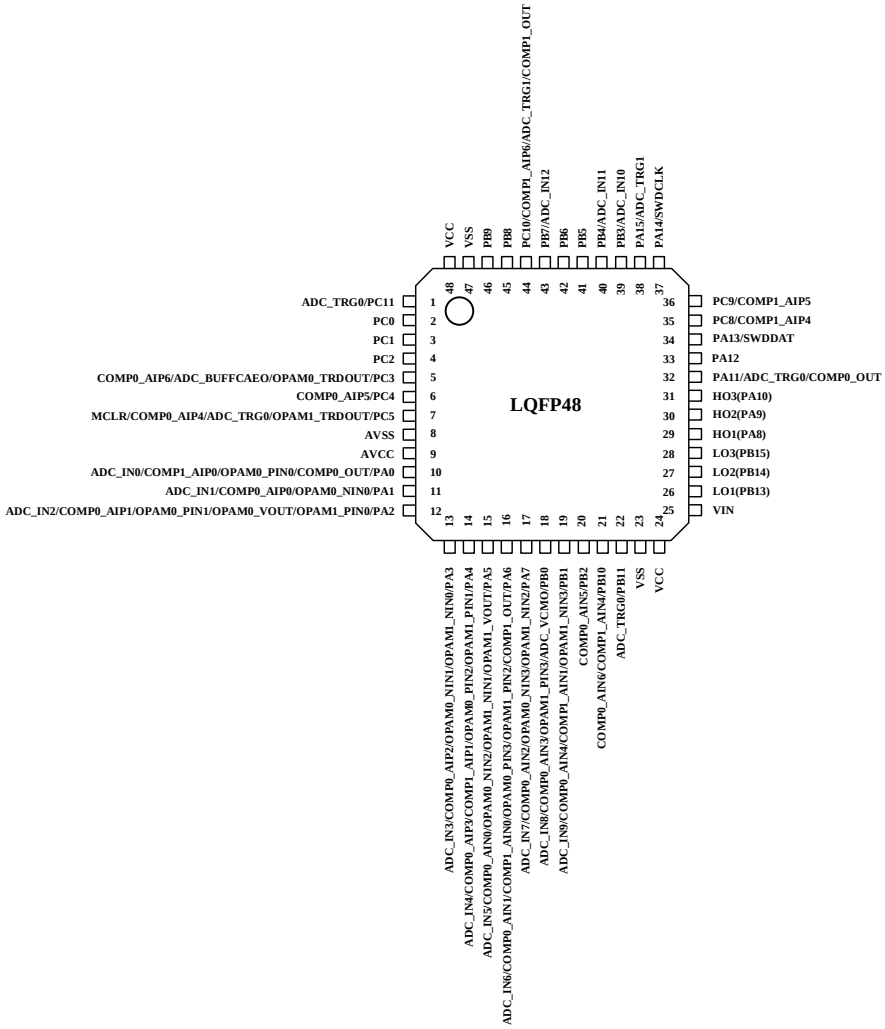


图 2-2 LQFP48（CIU32M011K3PB、CIU32M031K5PB）



2.2. 引脚功能描述

表 2-1 GPIO 引脚定义

LQFP48	SSOP24	引脚	引脚类型	I/O 电路类型	附加功能	复用功能
1	-	PC11	I/O	IOG1	-	ADC_TRG0 TIM4_BKIN TIM5_BKIN TIM6_BKIN
2	-	PC0	I/O	IOG1	-	PORT_WKUP_IN1 TIM6_CH1N TIM2_CH1
3	-	PC1	I/O	IOG1	-	TIM6_CH1 UART1_TX TIM5_CH2 TIM2_CH2
4	-	PC2	I/O	IOG1	-	SPI1_CS UART1_RX TM6_CH2 TIM2_CH3
5	23	PC3	I/O	IOG1	COMP0_AIP6	UART0_TX SPI1_CLK SPI1_IO0 TIM5_CH1 SYS_RXEV_IN ADC_BUFFCAEO OPAM0_TRDOUT
6	24	PC4	I/O	IOG1	COMP0_AIP5	UART0_RX SPI1_IO0 SPI1_CLK TIM5_CH1N SYS_NMI_IN TIM4_CH2 TIM5_CH2 TIM6_CH2
7	-	PC5	I/O	IOG1	COMP0_AIP4 MCLR	SPI1_IO1 TIM6_CH1N ADC_TRG0 OPAM1_TRDOUT
8	1	AVSS	-	-	-	-
9	2	AVCC	-	-	-	-
10	-	PA0	I/O	IOG1	ADC_IN0 COMP1_AIP0 OPAM0_PIN0	PORT_WKUP_IN0 TIM2_ETR TIM2_CH1 SPI1_CS TIM2_CH3

						TIM6_CH1 COMP0_OUT
11	-	PA1	I/O	IOG1	ADC_IN1 COMP0_AIP0 OPAM0_NIN0	UART0_TX TIM2_CH2 TIM4_CH1N TIM6_CH1N SPI1_IO0
12	-	PA2	I/O	IOG1	ADC_IN2 COMP0_AIP1 OPAM0_PIN1 OPAM0_VOUT OPAM1_PIN0	UART1_TX UART0_RX TIM2_CH3 SPI1_CS TIM4_CH1 TIM6_CH1 SPI1_IO1
13	-	PA3	I/O	IOG1	ADC_IN3 COMP0_AIP2 OPAM0_NIN1 OPAM1_NIN0	UART1_RX UART0_TX TIM2_CH4 SPI1_CS TIM4_CH2 TIM6_CH1N SPI1_CLK
14	-	PA4	I/O	IOG1	ADC_IN4 COMP0_AIP3 COMP1_AIP1 OPAM0_PIN2 OPAM1_PIN1	SPI0_CS UART1_TX TIM2_ETR TIM1_BKIN TIM3_CH1 SPI1_IO0 TIM1_CH2N TIM4_CH1
15	3	PA5	I/O	IOG1	ADC_IN5 COMP0_AIN0 OPAM0_NIN2 OPAM1_NIN1 OPAM1_VOUT	SPI0_CLK TIM2_ETR TIM2_CH1 TIM1_ETR SPI1_CLK TIM1_CH3N TIM4_CH1N
16	4	PA6	I/O	IOG1	ADC_IN6 COMP0_AIN1 COMP1_AIN0 OPAM0_PIN3 OPAM1_PIN2	SPI0_IO1 TIM3_CH1 TIM1_BKIN UART1_RX TIM1_ETR TIM5_CH1 TIM1_CH3 COMP1_OUT
17	5	PA7	I/O	IOG1	ADC_IN7 COMP0_AIN2 OPAM0_NIN3	SPI0_IO0 TIM3_CH2 TIM1_CH1N

					OPAM1_NIN2	TIM3_CH1 TIM6_CH1 TIM1_CH2N TIM1_CH3N
18	6	PB0	I/O	IOG1	ADC_IN8 COMP0_AIN3 OPAM1_PIN3 ADC_VCMO	TIM3_CH3 TIM1_CH2N TIM1_CH1N TIM1_CH3
19	7	PB1	I/O	IOG1	ADC_IN9 COMP0_AIN4 COMP1_AIN1 OPAM1_NIN3	TIM3_CH1 TIM3_CH4 TIM1_CH3N TIM1_CH4 TIM1_CH2N CLK_TO_IO TIM1_CH2 TIM1_CH1N
20	-	PB2	I/O	IOG1	COMP0_AIN5	TIM4_CH1 TIM1_BKIN
21	-	PB10	I/O	IOG1	COMP0_AIN6 COMP1_AIN4	SPI1_CLK TIM2_CH3 TIM5_CH1 SPI0_CLK
22	-	PB11	I/O	IOG1	-	ADC_TRG0 SPI1_IO0 TIM2_CH4 TIM5_CH1N
23	-	VSS	-	-	-	-
24	-	VCC	-	-	-	-
25	8	VIN(1)	-	-	-	-
26	10	LO1(2)	-	-	-	-
27	12	LO2	-	-	-	-
28	14	LO3	-	-	-	-
29	9	HO1(3)	-	-	-	-
30	11	HO2	-	-	-	-
31	13	HO3	-	-	-	-
32	-	PA11	I/O	IOG1	-	ADC_TRG0 TIM1_CH4 SPI1_IO1 SPI1_CLK COMP0_OUT
33	15	PA12	I/O	IOG1	-	TIM1_ETR SPI1_CS SPI1_IO1 SPI1_IO0 TIM1_CH2
34	16	PA13	I/O	IOG1	SWDDAT	UART0_RX TIM3_CH1

						TIM3_ETR SPI1_IO1 TIM1_CH2 TIM1_BKIN
35	-	PC8	I/O	IOG1	COMP1_AIP4	SPI1_CLK UART1_TX
36	-	PC9	I/O	IOG1	COMP1_AIP5	SPI1_IO0 UART1_RX
37	17	PA14	I/O	IOG1	SWDCLK	UART0_TX TIM4_CH1 SPI0_CS UART1_RX
38	-	PA15	I/O	IOG1	-	SPI0_CS UART0_RX TIM2_CH1 TIM2_ETR UART1_TX TIM1_ETR TIM6_CH1 ADC_TRG1
39	-	PB3	I/O	IOG1	ADC_IN10	SPI0_CLK TIM2_CH2 UART0_TX TIM2_CH3 TIM1_CH1 TIM2_CH1
40	18	PB4	I/O	IOG1	ADC_IN11	SPI0_IO1 TIM3_CH1 UART0_RX TIM6_BKIN TIM1_CH2 TIM2_CH2
41	19	PB5	I/O	IOG1	-	SPI0_IO0 TIM3_CH2 TIM5_BKIN TIM1_CH3 TIM2_CH3
42	20	PB6	I/O	IOG1	-	UART0_TX SPI1_CLK TIM5_CH1N UART1_RX TIM2_CH1
43	21	PB7	I/O	IOG1	ADC_IN12	UART0_RX SPI1_IO0 TIM6_CH1N UART1_TX
44	22	PC10	I/O	IOG1	COMP1_AIP6	PORT_WKUP_IN3

						TIM3_ETR ADC_TRG1 COMP1_OUT
45	-	PB8	I/O	IOG1	-	SPI1_CLK TIM5_CH1 UART0_TX UART1_RX
46	-	PB9	I/O	IOG1	-	TIM3_CH1 SPI1_IO0 TIM6_CH1 UART0_RX TIM1_CH4 SPI0_CS
47	-	VSS	-	-	-	-
48	-	VCC	-	-	-	-

注:

1. VIN：内置预驱电源输入引脚。当 VIN 接外部电源供电，则 VCC、AVCC 由内部驱动芯片供电，只需 VCC 外接滤波电容（1~10uf），AVCC 外接去耦电容（0.01~1uf）即可。
2. LO1、LO2、LO3：分别为相 1、相 2、相 3 低边栅极驱动输出引脚，PWM 控制上分别对应 PB13、PB14、PB15。
3. HO1、HO2、HO3：分别为相 1、相 2、相 3 高边栅极驱动输出引脚，PWM 控制上分别对应 PA8、PA9、PA10。

表 2-2 GPIO 复用功能 AF0~AF7

Port	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PA0	PORT_WKUP_IN0	TIM2_ETR	TIM2_CH1	SPI1_CS	TIM2_CH3	TIM6_CH1	-	COMP0_OUT
PA1	-	UART0_TX	TIM2_CH2	-	TIM4_CH1N	TIM6_CH1N	SPI1_IO0	-
PA2	UART1_TX	UART0_RX	TIM2_CH3	SPI1_CS	TIM4_CH1	TIM6_CH1	SPI1_IO1	-
PA3	UART1_RX	UART0_TX	TIM2_CH4	SPI1_CS	TIM4_CH2	TIM6_CH1N	SPI1_CLK	-
PA4	SPI0_CS	UART1_TX	TIM2_ETR	TIM1_BKIN	TIM3_CH1	SPI1_IO0	TIM1_CH2N	TIM4_CH1
PA5	SPI0_CLK	TIM2_ETR	TIM2_CH1	TIM1_ETR	-	SPI1_CLK	TIM1_CH3N	TIM4_CH1N
PA6	SPI0_IO1	TIM3_CH1	TIM1_BKIN	UART1_RX	TIM1_ETR	TIM5_CH1	TIM1_CH3	COMP1_OUT
PA7	SPI0_IO0	TIM3_CH2	TIM1_CH1N	-	TIM3_CH1	TIM6_CH1	TIM1_CH2N	TIM1_CH3N
PA8	-	UART0_TX	TIM1_CH1	TIM1_BKIN	-	-	TIM1_CH2	TIM1_CH3
PA9	TIM4_BKIN	UART0_TX	TIM1_CH2	UART0_RX	SPI1_CLK	CLK_TO_IO	TIM1_CH1N	TIM1_CH4
PA10	TIM6_BKIN	UART0_RX	TIM1_CH3	UART0_TX	SPI1_IO0	TIM1_BKIN	TIM1_CH1	SPI1_CLK
PA11	ADC_TRG0	-	TIM1_CH4	-	SPI1_IO1	SPI1_CLK	-	COMP0_OUT
PA12	-	-	TIM1_ETR	SPI1_CS	SPI1_IO1	SPI1_IO0	-	TIM1_CH2

PA13	UART0_RX	TIM3_CH1	TIM3_ETR	-	SPI1_IO1	-	TIM1_CH2	TIM1_BKIN
PA14	-	UART0_TX	TIM4_CH1	SPI0_CS	UART1_RX	-	-	-
PA15	SPI0_CS	UART0_RX	TIM2_CH1	TIM2_ETR	UART1_TX	TIM1_ETR	TIM6_CH1	ADC_TRG1
PB0	-	TIM3_CH3	TIM1_CH2N	TIM1_CH1N	TIM1_CH3	-	-	-
PB1	TIM3_CH1	TIM3_CH4	TIM1_CH3N	TIM1_CH4	TIM1_CH2N	CLK_TO_IO	TIM1_CH2	TIM1_CH1N
PB2	TIM4_CH1	TIM1_BKIN	-	-	-	-	-	-
PB3	SPI0_CLK	-	TIM2_CH2	UART0_TX	TIM2_CH3	-	TIM1_CH1	TIM2_CH1
PB4	SPI0_IO1	TIM3_CH1	-	UART0_RX	-	TIM6_BKIN	TIM1_CH2	TIM2_CH2
PB5	SPI0_IO0	TIM3_CH2	TIM5_BKIN	-	-	-	TIM1_CH3	TIM2_CH3
PB6	UART0_TX	SPI1_CLK	TIM5_CH1N	UART1_RX	TIM2_CH1	-	-	-
PB7	UART0_RX	SPI1_IO0	TIM6_CH1N	-	UART1_TX	-	-	-
PB8	-	SPI1_CLK	TIM5_CH1	UART0_TX	UART1_RX	-	-	-
PB9	TIM3_CH1	SPI1_IO0	TIM6_CH1	UART0_RX	TIM1_CH4	SPI0_CS	-	-
PB10	-	SPI1_CLK	TIM2_CH3	TIM5_CH1	-	SPI0_CLK	-	-
PB11	ADC_TRG0	SPI1_IO0	TIM2_CH4	TIM5_CH1N	-	-	-	-
PB13	SPI0_CLK	SPI0_IO1	TIM1_CH1N	SPI1_CS	SPI1_IO0	SPI1_CLK	TIM1_CH3N	TIM2_CH1
PB14	SPI0_IO1	SPI0_IO0	TIM1_CH2N	SPI1_CLK	SPI1_CS	SPI1_IO0	TIM1_CH3	TIM1_CH1
PB15	SPI0_IO0	SPI0_CS	TIM1_CH3N	SPI1_IO1	SPI1_CLK	ADC_TRG1	TIM1_CH2N	TIM1_CH2
PC0	PORT_WKUP_IN1	TIM6_CH1N	-	-	-	-	TIM2_CH1	-
PC1	-	TIM6_CH1	UART1_TX	TIM5_CH2	-	-	TIM2_CH2	-
PC2	-	SPI1_CS	UART1_RX	TM6_CH2	-	-	TIM2_CH3	-
PC3	UART0_TX	SPI1_CLK	SPI1_IO0	TIM5_CH1	SYS_RXEV_IN	ADC_BUFFCAEO	-	OPAM0_TRDOUT
PC4	UART0_RX	SPI1_IO0	SPI1_CLK	TIM5_CH1N	SYS_NMI_IN	TIM4_CH2	TIM5_CH2	TIM6_CH2
PC5	-	SPI1_IO1	-	TIM6_CH1N	-	ADC_TRG0	-	OPAM1_TRDOUT
PC8	SPI1_CLK	UART1_TX	-	-	-	-	-	-
PC9	SPI1_IO0	UART1_RX	-	-	-	-	-	-

PC10	PORT_WKUP_IN3	-	TIM3_ETR	-	-	ADC_TRG1	-	COMP1_OUT
PC11	ADC_TRG0	-	-	-	TIM4_BKIN	TIM5_BKIN	TIM6_BKIN	-

注：UART0_RX/UART1_RX 单线工作时作收发引脚，双线工作时作接收引脚。

表 2-3 SSP 引脚选择

	SPI 功能	**I2C 功能**
SSP_CLK	SPI0_CLK	I2C_CLK
SSP_CS	SPI0_CS	-
SSP_IO0	SPI0_IO0	I2C_DATA
SSP_IO1	SPI0_IO1	-

3. 系统及存储器架构

CIU32M011、CIU32M031 器件是基于 ARM Cortex M0 处理器的 32 位通用微控制器存储器芯片。采用了哈佛结构，具有低中断延迟时间和低成本调试特性，而且高集成度和增强的特性使这颗处理器适合于那些需要高性能和低功耗微控制器的市场领域。预先定义的存储器映射和高达 4GB 的存储空间，充分保证了系统的灵活性和可扩展性。

3.1. 系统架构

CIU32M011、CIU32M031 器件采用 32 位多层总线结构，该结构可使系统中的多个主机和从机之间的并行通信成为可能。多层总线结构包括一个 AHB 互联矩阵、两个 AHB 总线和两个 APB 总线。AHB 互联矩阵的互联关系接下来将进行说明。

CIU32M011、CIU32M031 主系统由以下两部分构成：

- 2 个驱动单元
  - CPU 内核系统总线(S-bus)
  - DMA 总线
- 2 个存储单元
  - 内部闪存存储器
  - 内部 SRAM

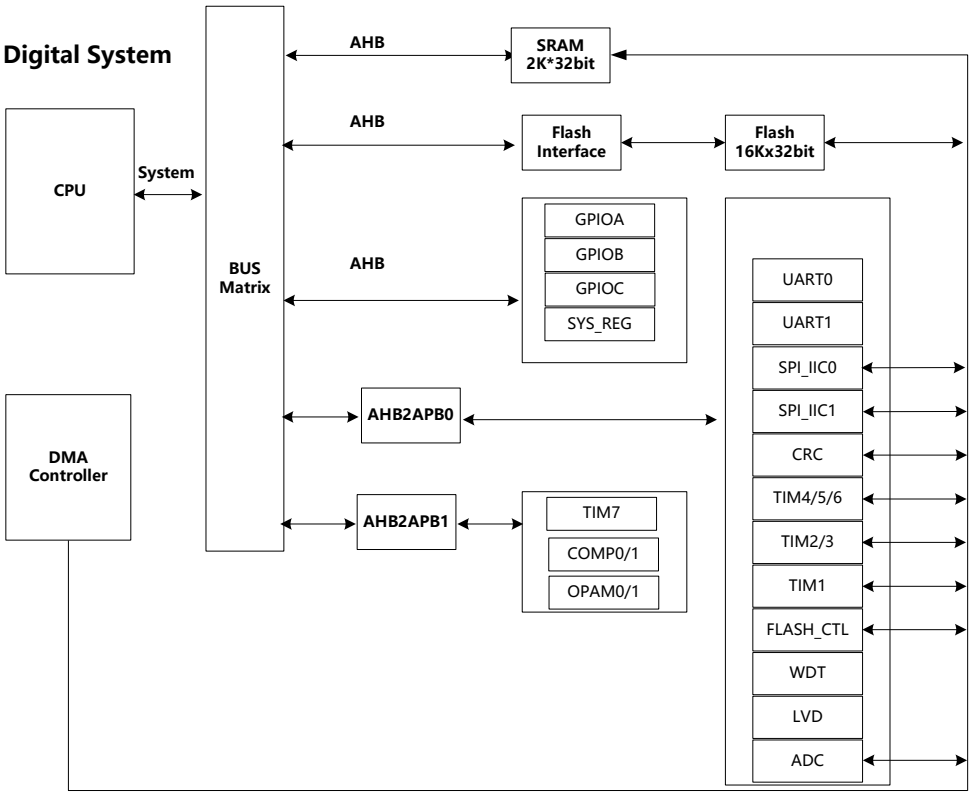


图 3-1 总线系统架构图

系统总线

此总线连接 CPU 内核的系统总线(外设总线)到总线矩阵，总线矩阵协调着内核和各个高速部件间的访问。

总线矩阵(Bus Matrix)

- 总线矩阵管理着内核系统总线与各外设模块的访问仲裁，总线矩阵由主模块总线及从模块总线组成。
- AHB 外设通过总线矩阵与系统总线相连。
- AHB 到 APB 桥(AHB2APB bridges-APB)。
- AHB 到 APB 桥在 AHB 与 APB 总线间提供同步连接。

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

3.2. 存储器映射

此 32 位处理器采用同一套总线来读取指令和加载/存储数据。指令代码和数据都位于相同的存储器地址空间，但在不同的地址范围。程序存储器，数据存储器，寄存器和 IO 端口都在同一个线性的 4GB 的地址空间之内。这是 32 位处理器的最大地址范围，因为它的地址总线宽度是 32 位。此外，为了降低不同客户在相同应用时的软件复杂度，存储映射是按 32 位处理器提供的规则预先定义的。在存储器映射表中，一部分地址空间由 32 位处理器的系统外设所占用，且不可更改。此外，其余部分地址空间可由芯片供应商定义使用。CIU32M011、CIU32M031 器件的存储器映射表显示了 CIU32M011、CIU32M031 器件的存储器映射，包括代码、SRAM、外设和其他预先定义的区域。简化了每个外设的地址译码。

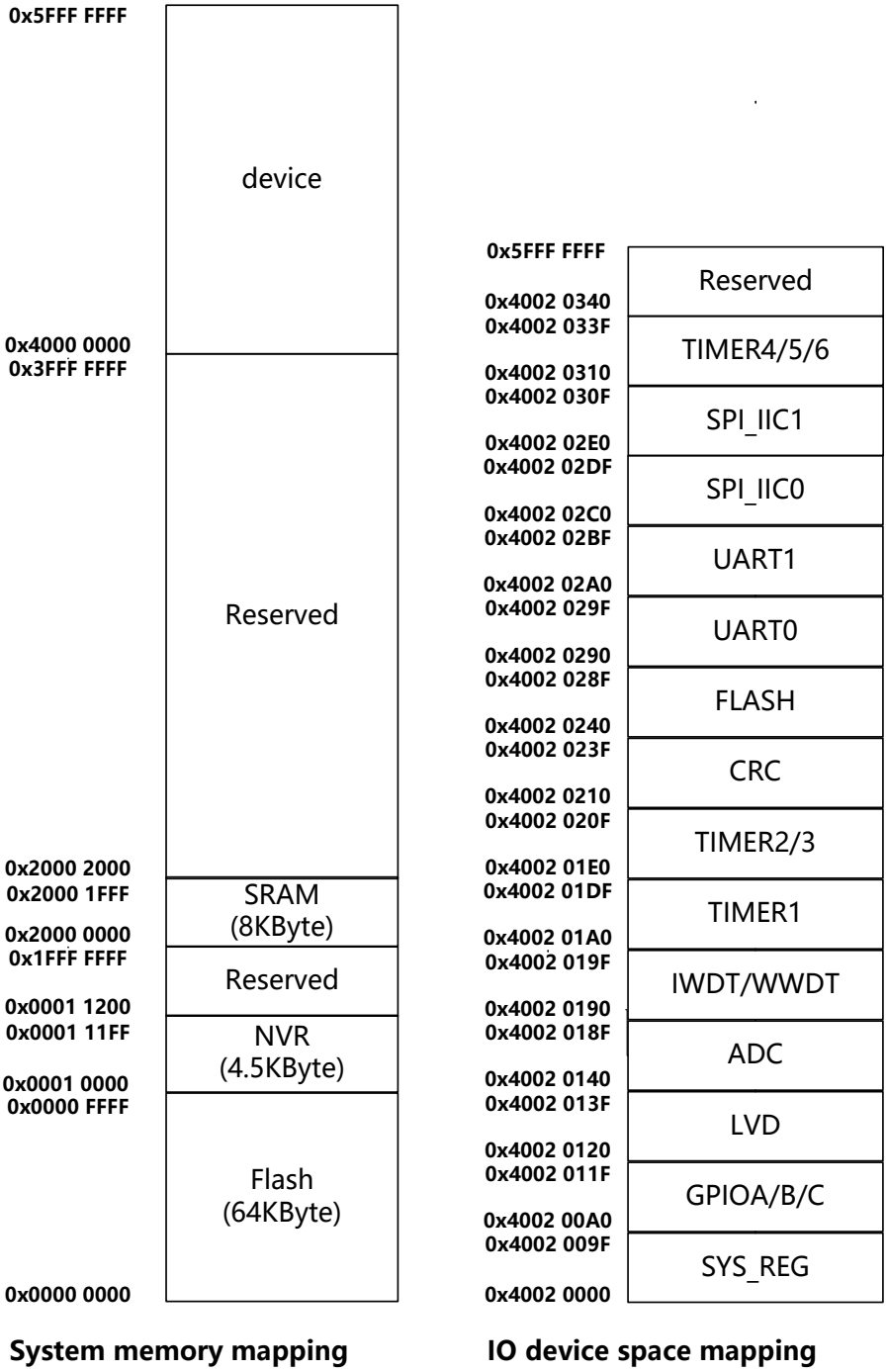


图 3-2 存储器映射表

3.2.1. 片上 SRAM

芯片内置 8K 字节的 SRAM。它可以按字节(8 位)、半字(16 位)或字(32 位)进行访问。 SRAM 起始地址为 0x2000_0000。

3.2.2. 片上 FLASH 概述

闪存存储器有两个不同存储区域

- 主闪存存储块，它包括应用程序和用户数据区(若需要时)
- 副闪存存储块，也叫信息块，其包含两个部分
  - 选项字节(Option bytes) - 内含硬件及存储保护用户配置选项。
  - 系统存储器(System memory) - 闪存接口基于 AHB 协议执行指令和数据存取。

3.2.3. 引导配置

芯片复位后，通过客户自己在副闪存的配置，选择 SWDCLK 默认工作是上拉还是下拉。正常启动后，CPU 从地址 0x0000_0000 获取堆栈顶的地址，并从存储器的 0x0000_0004 位置指示的地址开始执行代码。

3.2.4. 位带操作

为了减少“读-改-写”操作的次数，32 位 RISC 处理器提供了一个可以执行单原子比特操作的位带功能。存储器映射包含了两个支持位带操作的区域。其中一个是 SRAM 区的最低 1MB 范围，第二个是片内外设区的最低 1MB 范围。这两个区域中的地址除了普通应用外，还有自己的“位带别名区”。位带别名区把每个比特扩展成一个 32 位的字。当用户访问位带别名区时，就可以达到访问原始比特的目的。

下面的公式表明了位带别名区中的每个字如何对应位带区的相应比特或目标比特。

bit_word_addr =bit_band_base +(byte_offset×32)+(bit_number×4)

其中：

- bit_word_addr 指的是位带区目标比特对应在位带别名区的地址；
- bit_band_base 指的是位带别名区的起始地址；
- byte_offset 指的是位带区目标比特所在的字节的字节地址偏移量；
- bit_number 指的是目标比特在对应字节中的位置(0-7)。

例如，要想访问 0x2000 0200 地址的第 7 位，可访问的位带别名区地址是：

bit_word_addr = 0x2200 0000 + (0x200 * 32)+ (7 * 4)= 0x2200 401C

如果对 0x2200 401C 进行写操作，那么 0x2000 0200 的第 7 位将会相应变化；如果对 0x2200 401C 进行读操作，那么视 0x2000 0200 的第 7 位状态而返回 0x01 或 0x00。

4. 嵌入式闪存（FLASH）

4.1. 模块介绍

CIU32M011、CIU32M031 集成了嵌入式 FLASH 控制模块，该模块控制 FLASH 的擦除、编程以及读取数据。上电时会从 FLASH 中读取相关数据进行校验以及初始化配置，保证芯片程序在正确且安全的情况下运行。

4.2. 功能特点

- 支持高达 64K 主闪存空间的 FLASH
- 存储器结构
  - 主闪存空间 64K 字节
  - 副闪存空间 4.5K 字节
- 指出对闪存空间的擦写、编程和读操作
- 支持对闪存空间访问限制和擦写保护
- 支持低功耗模式

4.3. 功能说明

4.3.1. 闪存结构

闪存空间由 32 位宽的存储单元组成，既可以存代码又可以存数据。主闪存块按 64 页（每页 1K 字节、2 个扇区）分块，以页为单位设置写保护（参见存储保护相关内容），擦写以扇区为单位。

表 4-1 闪存结构

闪存空间	名称	地址	大小(字节)
主闪存空间	Page 0	0x0000_0000 – 0x0000_03FF	1K
	Page 1	0x0000_0400 – 0x0000_07FF	1K
	Page 2	0x0000_0800 – 0x0000_0BFF	1K
	Page 3	0x0000_0C00 – 0x0000_0FFF	1K
	…	…	1K
	Page 62	0x0000_F800 – 0x0000_FBFF	1K
	Page 63	0x0000_FC00 – 0x0000_FFFF	1K
副闪存空间	Sector 0	0x0001_0000 – 0x0001_01FF	512
	Sector 1	0x0001_0200 – 0x0001_03FF	512
	…	…	512
	Sector 8	0x0001_1000 – 0x0001_11FF	512

注：当主闪存空间 64KB 不够存放用户程序时，可把副闪存空间的扇区 0 和扇区 7 扩展为程序存放空间，即支持最大 68KB 的程序存放空间。

4.3.2. 闪存读保护

读操作在整个芯片工作电压范围内都可以完成，用于存放指令或者数据。

当 NVR8 用户配置区经过自定义的保护配置后,SWD 连接时会对 FLASH 的代码数据执行保护机制。

注：FLASH 运行在 24MHz 工作频率，当系统时钟超过 30MHz 时，需要配置 TIMER_REG0 的 RC 参数，增加时钟周期数再把 FLASH 接口的数据写到寄存器。

4.3.3. 闪存擦除和烧写操作

烧写和擦除操作在整个芯片工作电压范围内都可以完成。

烧写和擦除操作由下列 6 个寄存器完成，先根据烧写的时钟配置好烧写时序(TIME_REG1)，再配置烧写密码，配置好编程地址，最后配置好编程数据，即可开始执行烧写，然后等待操作结束。

烧写操作相关寄存器

- 时序寄存器 1 ：TIME_REG1
- 密码寄存器 ：NVR_PASSWORD/MAIN_PASSWORD
- 编程地址寄存器：PROG_ADDR
- 编程数据寄存器：PROG_DATA
- 状态寄存器 ：DONE

擦除操作相关寄存器：

- 擦除控制寄存器：ERASE_CTRL

注：需要注意的是，FLASH 在擦除/烧写的同时不可以从 FLASH 取数据，所以 FLASH 在擦除/烧写过程中会让总线停顿，直到完成后才能继续运行。

4.4. 模块框图

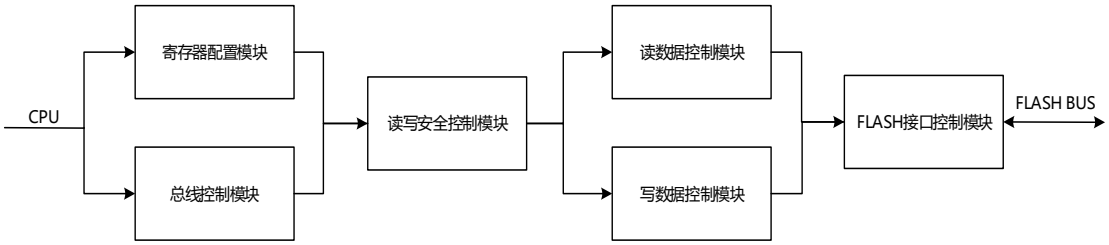


图 4-1 FLASH 模块架构图

4.5. 寄存器描述

4.5.1. 寄存器列表

Base address：0x4002_0000

Name	Offset	Reset	Description
EFLASH_CTRL0	0x0240	32'h1000	FLASH 控制寄存器 0
EFLASH_KST	0x0244	32'h0	触发寄存器
EFLASH_DONE	0x0248	32'h1c53	完成状态寄存器
EFLASH_PROG_ADDR	0x024C	32'h80000000	编程地址寄存器
EFLASH_PROG_DATA	0x0250	32'h0	编程数据寄存器
EFLASH_ERASE_CTRL	0x0254	32'h0	擦除控制寄存器
EFLASH_TIME_REG0	0x0258	32'h7efdfbf2	时序控制寄存器 0
EFLASH_TIME_REG1	0x025C	32'h3e810	时序控制寄存器 1
EFLASH_NVR_PWD	0x0260	32'h0	NVR 秘钥寄存器
EFLASH_MAIN_PWD	0x0264	32'h0	MAIN 秘钥寄存器
EFLASH_CRC_ADDR	0x0268	32'h0	CRC 校验地址
EFLASH_CRC_LEN	0x026C	32'h0	CRC 校验长度
EFLASH_CRC_OUT	0x0270	32'h5a5a55aa	CRC 校验结果
EFLASH_CFG_SECTOR	0x0280	32'h380	MAIN 区域扇区数量寄存器

4.5.2. 寄存器详细说明

4.5.2.1. 控制寄存器（EFLASH_CTRL0）

Width	Name	Reset	Property	Description
31:21	Reserved	-	-	-

20	ERR_MUL_IE	1'b0	RW	ecc 出错多 bit 中断使能
19	ERR_DET_IE	1'b0	RW	ecc 出错 1bit 中断使能
18	FWUP	1'b0	RW	EFLASH 快速唤醒使能 0:10us 唤醒时间 1:5us 唤醒时间
17	Reserved	-	-	-
16	PROG_CLK_SEL	1'b0	RW	FLASH 烧写时钟源选择，推荐使用 RC 时钟 0: HIRC 时钟 2 分频 1: 保留
15:13	Reserved	-	-	-
12	PROG_RAM_CRC_EN	1'b1	RW	把 RAM 的数据搬移到 FLASH 时在最后添加 CRC 值 0: 不使能 1: 使能
11	LVD_PROG_STOP	1'b0	RW	在 LVD 断电时，是否允许打断 FLASH 编程和擦除，如果断电时，需要立刻把重要数据保存在 FLASH 上，就打开此功能，用于快速让 FLASH 处于空闲状态 0: 不允许打断 1: 允许打断
10	PROG_RAM_MODE	1'b0	RW	直接把 RAM 的数据写到 FLASH，根据 PROG_RAM_CRC_EN 为 1 时在 FLASH 最后自动添加 CRC 校验值，具体影响到几个寄存器:PROG_ADDR、CRC_ADDR、CRC_LEN 0: FLASH CRC 模式 1: RAM to FLASH 模式
9:0	Reserved	-	-	-

4.5.2.2. 触发寄存器（EFLASH_KST）

Width	Name	Reset	Property	Description
31:27	Reserved	-	-	-
26	CRC_KST_EN	1'b0	WO	FLASH 的 CRC 触发使能 0: 不使能 1: 使能
25:11	Reserved	-	-	-
10	CRC_KST	1'b0	WO	FLASH 的 CRC 校验触发 与第 26 位同时写‘1’触发
9:0	Reserved	-	-	-

4.5.2.3. 状态寄存器（EFLASH_DONE）

Width	Name	Reset	Property	Description
31:13	Reserved	-	-	-
12	CHIP_ERASE_OK_FLAG	1'b1	RO	Chip 擦除结果 0: 擦除失败 1: 擦除成功
11	PROG_OK_FLAG	1'b1	RO	编程结果，写入后读出数据检查是否与写入数据

				一致 0: 烧写失败 1: 烧写成功
10	CRC_DONE	1'b1	RO	FLASH 的 CRC 完成标志 0: 进行中 1: 空闲状态
9:7	Reserved	-	-	-
6	PROG_DONE	1'b1	RO	编程结束标志 0: 进行中 1: 空闲状态
5:2	Reserved	-	-	-
1	CHIP_ERASE_DONE	1'b1	RO	Main 区域全擦除标志 0: 正在运行 1: 空闲状态
0	SECTOR_ERASE_DONE	1'b1	RO	扇区擦除标志 0: 正在运行 1: 空闲状态

4.5.2.4. 编程地址寄存器（EFLASH_PROG_ADDR)

Width	Name	Reset	Property	Description
31:30	Reserved	-	-	-
29	PROG_NVR_SEL	1'b0	RW	编程的地址选择 MAIN/NVR 0: MAIN 区域 1: NVR 区域
28:16	Reserved	-	-	-
15:0	ADDR	16'h0	RW	FLASH 编程的逻辑地址（即单位是字节），编程需要以 word 为单位 (即地址低两位需为 0)。 注: 在 CTRL0 寄存器里的第 10 位有效时，此寄存器作为 FLASH 编程的开始地址.

4.5.2.5. 编程数据寄存器（EFLASH_PROG_DATA)

Width	Name	Reset	Property	Description
31:0	DATA	32'h0	RW	PROG_RAM_MODE 为 0 时: 配置好 PROG_ADDR 和 PASSWORD 后，配置 PROG_DATA 启动编程; PROG_RAM_MODE 为 1 时: 该寄存器显示当前往 FLASH 编程的数据。

4.5.2.6. 擦除控制寄存器（EFLASH_ERASE_CTRL)

Width	Name	Reset	Property	Description
31	CHIP_ERASE_KST	1'b0	WO	Chip 擦除的触发，写“1”触发，需要先配置密码
30	SECTOR_ERASE_KST	1'b0	WO	扇区擦除的触发，写“1”触发，需要先配置密码
29	NVR_SECTOR_EN	1'b0	RW	NVR 的扇区使能位 0: MAIN 区域

				1: NVR 区域
28:7	Reserved	-	-	-
6:0	SECTOR_ERASE_ADDR	7'h0	RW	擦除的扇区选择

4.5.2.7. 时序 0 寄存器（EFLASH_TIME_REG0）

Width	Name	Reset	Property	Description
31:25	PGH	7'h3f	RW	WE high to PROG2 high hold min time is 500ns（以系统时钟周期为单位来计算）
24:18	ADS	7'h3f	RW	BYTE/Address/data setup min time is 500ns（以系统时钟周期为单位来计算）
17:11	ADH	7'h3f	RW	BYTE/Address/data hold min time is 500ns（以系统时钟周期为单位来计算）
10:4	RW	7'h3f	RW	Latency to next operation after PROG/sector ERASE low min time is 500ns（以系统时钟周期为单位来计算）
3:0	RC	4'h2	RW	读 flash 周期 当不开启 ECC 校验功能时： 0: 1 周期（用于系统时钟 0~24MHz） 1: 2 周期（用于系统时钟 24~48MHz） 2: 3 周期（用于系统时钟 48~72MHz） 3: 4 周期（用于系统时钟 72~96MHz） 其它：保留 当开启 ECC 校验功能时： 0: 1 周期（用于系统时钟 0~12MHz） 1: 2 周期（用于系统时钟 12~24MHz） 2: 3 周期（用于系统时钟 24~48MHz） 3: 4 周期（用于系统时钟 48~72MHz） 4: 5 周期（用于系统时钟 72~96MHz） 其它：保留

4.5.2.8. 时序 1 寄存器（EFLASH_TIME_REG1）

Width	Name	Reset	Property	Description
31:20	Reserved	-	-	-
18:8	MS_CNT_SET	11'h3e8	RO	1ms 的时间配置值，以 1us 为单位
7:0	US_CNT_SET	8'h10	RW	1us 的时间配置值，时钟频率默认是 32MHz 的 2 分频

4.5.2.9. 密码 0 寄存器（EFLASH_NVR_PWD）

Width	Name	Reset	Property	Description
31:0	NVR_PASS_WORD	32'h0	RW	NVR 区域密码，配置后才能对 NVR 区域编程/擦除. NVR0~8 密码为: 0x20150931

4.5.2.10. 密码 1 寄存器（EFLASH_MAIN_PWD）

Width	Name	Reset	Property	Description

31:0	MAIN_PASS_WORD	32'h0	RW	MAIN 区域密码，配置后才能对 MAIN 区域编程/擦除，密码为: 0x20170230

4.5.2.11. CRC 地址配置寄存器（EFLASH_CRC_ADDR）

Width	Name	Reset	Property	Description
31:30	Reserved	-	-	-
29	NVR_SEL	1'b0	RW	地址是否 NVR 区域 0: MAIN 区域 1: NVR 区域
28:2	CRC_ADDR	27'h0	RW	PROG_RAM_MODE 为 0 时: 配置成 FLASH 中需要校验 CRC 的开始地址，且是 FLASH 的物理地址。 注意: 地址需要按 word 地址配置
1:0	Reserved	-	-	-

4.5.2.12. CRC 长度配置寄存器（EFLASH_CRC_LEN）

Width	Name	Reset	Property	Description
31:17	Reserved	-	-	-
16:2	CRC_LEN	15'h0	RW	FLASH 校验 CRC 的数据长度 注意: 长度需要按 word 长度配置
1:0	Reserved	-	-	-

4.5.2.13. CRC 结果配置寄存器（EFLASH_CRC_OUT）

Width	Name	Reset	Property	Description
31:0	CRC_OUT	32'h5a5a55aa	RO	CRC 的结果，多项式 CRC-32 如下: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

4.5.2.14. 主扇区信息寄存器（EFLASH_CFG_SECTOR）

Width	Name	Reset	Property	Description
31:18	Reserved	-	-	-
17:16	RAM_SIZE	2'b11	RO	b00: 4k b01: 6k b1x: 8k
15:0	MAIN_SECTOR_NUM	32'h80	RO	EFLASH MAIN 的 sector 数

4.5.3. NVR8 闪存用户配置区

副闪存 NVR8 最后 64byte 为用户配置区，逻辑地址区间：0x0001_11C0-0x0001_11FF,该区域的配置数据需要在程序烧写阶段烧写，芯片正常上电后这些信息为只读。

4.5.3.1. 系统配置 0（0x0001_11C0）

Width	Name	Reset	Property	Description
31	Reserved	-	-	-
30	Reserved	-	-	-
29	Reserved	-	-	-

28:11	Reserved	-	-	-
10	CODE_PROTECT_DIS	1'b1	RO	代码保护使能 0: 使能 1: 不使能 （使能后 SWD 连接时，FLASH 的读、写、擦权限都会关闭，且部分模块寄存器的读写权限也关闭）
9:8	Reserved	-	-	-
7	NVR7_WRITE_EN	1'b1	RO	NVR7 擦写权限 0: 关闭 1: 打开
6	NVR6_WRITE_EN	1'b1	RO	NVR6 擦写权限 0: 关闭 1: 打开
5	NVR5_WRITE_EN	1'b1	RO	NVR5 擦写权限 0: 关闭 1: 打开
4	NVR4_WRITE_EN	1'b1	RO	NVR4 擦写权限 0: 关闭 1: 打开
3	NVR3_WRITE_EN	1'b1	RO	NVR3 擦写权限 0: 关闭 1: 打开
2	NVR2_WRITE_EN	1'b1	RO	NVR2 擦写权限 0: 关闭 1: 打开
1	NVR1_WRITE_EN	1'b1	RO	NVR1 擦写权限 0: 关闭 1: 打开
0	NVR0_WRITE_EN	1'b1	RO	NVR0 擦写权限 0: 关闭 1: 打开

4.5.3.2. 系统配置 1（0x0001_11C4）

Width	Name	Reset	Property	Description
31:0	MAIN_WRITE_EN	32'hffffffff	RO	MAIN 擦写权限，每 1bit 代表 1Kbyte 0: 关闭 1: 打开

4.5.3.3. 系统配置 2（0x0001_11C8）

Width	Name	Reset	Property	Description
31:0	MAIN_WRITE_EN1	32'hffffffff	RO	MAIN 后 32K 擦写权限，每 1bit 代表 1Kbyte 0: 关闭 1: 打开

4.5.3.4. 系统配置 5（0x0001_11D4）

Width	Name	Reset	Property	Description
31	CODE_CHECK_EN	1'b1	RO	CODE CRC 校验使能 0: 不使能 1: 使能
30:16	Reserved	-	-	-
15:0	CODE_LEN	16'hffff	RO	需要进行 CRC 校验的 CODE 长度，以 word 为单位，超过该长度的区域为 DATA 区。

4.5.3.5. 功能配置 0（0x0001_11E0）

Width	Name	Reset	Property	Description
31:15	Reserved	-	-	-
14	SWD_EN	1'b1	RO	SWD 接口是否使能 0: 关闭 1: 使能 该位和 SYS_CON1[8]同时为 0 时才关闭
13	Reserved	-	-	-
12	Reserved	-	-	-
11	Reserved	-	-	-
10:9	SWD_PULL_MODE	2'b11	RO	SWD 接口的 TCK 复位状态下上拉或下拉 10: 下拉 11: 上拉 Others: 上下拉关闭，SWD 也关闭
8:0	Reserved	-	-	-

4.5.4. NVR9 闪存芯片信息区

4.5.4.1. 芯片信息 0(0x0001_1200)

Width	Name	Reset	Property	Description
31:0	UID0	32'hffffffff	RO	UID 信息 0

4.5.4.2. 芯片信息 1(0x0001_1204)

Width	Name	Reset	Property	Description
31:0	UID1	32'hffffffff	RO	UID 信息 1

4.5.4.3. 芯片信息 2(0x0001_1208)

Width	Name	Reset	Property	Description
31:0	UID2	32'hffffffff	RO	UID 信息 2

4.5.4.4. 芯片信息 3(0x0001_120C)

Width	Name	Reset	Property	Description
31:2	Reserved	32'hffffffff	RO	-
1	NVR_ECC_EN	1'b1	RO	nvr 区 ecc 功能使能
0	MAIN_ECC_EN	1'b1	RO	main 区 ecc 功能使能

4.5.4.5. 芯片信息 6(0x0001_1218)

Width	Name	Reset	Property	Description
31:0	CUSTOMER_ID	32'b0	RO	客户信息

5. 中断和事件（INT/EVT）

5.1. 嵌套向量中断控制器

- 中断都可屏蔽(除了 NMI)
- 4 个可编程的优先等级
- 低延迟的异常和中断处理
- 电源管理控制
- 系统控制寄存器的实现

嵌套向量中断控制器(NVIC)和处理器核的接口紧密相连，可以实现低延迟的中断处理和高效地处理晚到的中断。

嵌套向量中断控制器管理着包括核异常等中断。关于更多的异常和 NVIC 编程的说明请参考 CPU 技术参考手册。

5.2. 系统滴答(SysTick)校准值寄存器

本芯片支持系统滴答计时。

5.3. 中断功能描述

处理器和嵌套式矢量型中断控制器(NVIC)在处理(Handler)模式下对所有异常进行优先级区分以及处理。当异常发生时，系统自动将当前处理器工作状态压栈，在执行完中断服务子程序（ISR)后自动将其出栈。

取向量是和当前工作态压栈并行进行的，从而提高了中断入口效率。处理器支持咬尾中断，可实现背靠背中断，大大削减了反复切换工作态所带来的开销。

表 5-1 NVIC 异常类型

异常类型	向量编号	优先级	向量地址	描述
-	0	--	0x0000_0000	Reserved
复位	1	-3（最高）	0x0000_0004	复位
NMI	2	-2	0x0000_0008	不可屏蔽中断
硬件故障(HardFault)	3	-1	0x0000_000C	各种硬件级别故障
-	4-10	--	0x0000_0010 0x0000_002B	Reserved
SVcall 服务调用	11	--	0x0000_002C	通过 SWI 指令实现系统服务调用
-	12-13	--	0x0000_0030 0x0000_0037	Reserved
PendSV 挂起服务	14	可编程设置	0x0000_0038	可挂起的系统服务请求
系统节拍	15	可编程设置	0x0000_003C	系统节拍定时器

表 5-2 中断向量表

中断编号	向量编号	外设中断描述	向量地址
IRQ0	16	LVD_IRQn	0x0000_0040
IRQ1	17	TIM1_IRQn	0x0000_0044
IRQ2	18	TIM2_IRQn	0x0000_0048
IRQ3	19	TIM3_IRQn	0x0000_004C
IRQ4	20	ADC_IRQn	0x0000_0050
IRQ5	21	TIM4_IRQn	0x0000_0054
IRQ6	22	TIM5_IRQn	0x0000_0058
IRQ7	23	TIM6_IRQn	0x0000_005C

IRQ8	24	TIM7_IRQn	0x0000_0060
IRQ9	25	WKPND_IRQn	0x0000_0064
IRQ10	26	WDT_IRQn	0x0000_0068
IRQ11	27	WWDG_IRQn	0x0000_006C
IRQ12	28	UART0_IRQn	0x0000_0070
IRQ13	29	UART1_IRQn	0x0000_0074
IRQ14	30	SPI0_IRQn	0x0000_0078
IRQ15	31	SPI1_IRQn	0x0000_007C
IRQ16	32	GPIOA_IRQn	0x0000_0080
IRQ17	33	GPIOB_IRQn	0x0000_0084
IRQ18	34	GPIOC_IRQn	0x0000_0088
IRQ19	35	DMA0_IRQn	0x0000_008C
IRQ20	36	DMA1_IRQn	0x0000_0090
IRQ21	37	DMA2_IRQn	0x0000_0094
IRQ22	38	DMA3_IRQn	0x0000_0098
IRQ23	39	DMA4_IRQn	0x0000_009C
IRQ24	40	DMA5_IRQn	0x0000_00A0
IRQ25	41	DMA6_IRQn	0x0000_00A4
IRQ26	42	DMA7_IRQn	0x0000_00A8
IRQ27	43	HWDIV_IRQn	0x0000_00AC
IRQ28	44	COMP_IRQn	0x0000_00B0
IRQ29	45	-	0x0000_00B4
IRQ30	46	-	0x0000_00B8
IRQ31	47	-	0x0000_00BC

5.4. 外部中断/事件控制器(EXTI)

外部中断/事件控制器包含 44 个产生中断/事件触发的边沿检测电路，每条输入线可以独立地配置触发事件类型（上升沿或下降沿或者双边沿都触发）。每条输入线都可以独立地被屏蔽，挂起寄存器保持着状态线的中断请求，可通过对挂起的寄存器对应位写“1”清除中断请求。

5.4.1. 主要特征

EXTI 控制器的主要特性如下

- 每个中断/事件都有独立的触发和屏蔽
- 每个中断线都有专用的状态位
- 支持多达 44 个软件中断/事件请求
- 支持上升沿、下降沿和双边沿 3 种触发事件类型

5.4.2. 唤醒事件管理

CIU32M011、CIU32M031 可以处理外部或内部事件来唤醒内核(WFE)。唤醒事件可以通过下述配置产生：

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

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

6. 循环冗余校验计算单元（CRC)

6.1. 模块介绍

循环冗余校验(CRC)计算单元是根据自定义的生成多项式得到任意一个 32 位全字的 CRC 计算结果。在其他的应用中，CRC 技术主要应用于核实数据传输或者数据存储的正确性和完整性。CRC 计算单元可以在程序运行时计算出软件的标识，之后与在连接时生成的参考标识比较，然后存放在指定的存储器空间。

6.2. 功能特点

- 支持 16/32 位不同长度的多项式
- 支持自定义的多项式
- 默认是 32 位多项式:
  $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$
- 一个 32 位初始值配置寄存器用于输入，一个 32 位结果寄存器用于输出结果

6.3. 功能说明

该模块用于计算 SRAM 中指定数据段的 CRC 校验值，软件配置计算初始值、校验多项式、起始地址、数据长度，启动 CRC 计算后等待硬件完成标志有效时读取 CRC_OUT 寄存器可获得 CRC 校验值。

6.4. 模块框图

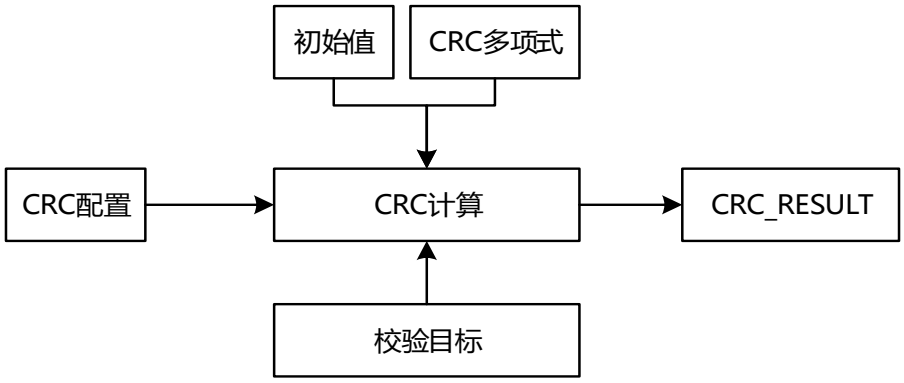


图 6-1 CRC 计算单元框图

6.5. 时钟与复位

6.5.1. 时钟介绍

该模块时钟源为系统时钟，低功耗模式下到达模块的时钟会被自动停止。

6.5.2. 复位介绍

该模块的复位源有两个，分别是系统复位和软件复位，软件复位可通过配置系统寄存器触发。

6.6. 寄存器描述

6.6.1. 寄存器列表

Base address：0x4002 0000

Name	Offset	Reset	Description
CRC_CFG	0x0210	32'h100	CRC 控制寄存器
CRC_INIT	0x0214	32'hffffffff	CRC 初始值寄存器

CRC_INV	0x0218	32'hffffffff	CRC 输出结果取反控制寄存器
CRC_POLY	0x021C	32'hedb88320	CRC 多项式配置寄存器
CRC_STA	0x0220	32'h0	CRC 状态标志寄存器
CRC_DATA	0x0224	32'h0	CRC 数据寄存器
CRC_LEN	0x0228	32'h0	CRC 长度寄存器

6.6.2. 寄存器详细说明

6.6.2.1. 控制配置寄存器（CRC_CFG）

Width	Name	Reset	Property	Description
31:9	Reserved	-	-	-
8	POLY_WIDTH	1'b1	RW	多项式宽度 0: 16 位 1: 32 位
7:2	Reserved	-	-	-
1	BIT_ORDER	1'b0	RW	输入数据的校验顺序 0: 高位到低位(7:0) 1: 低位到高位(0:7)
0	EN	1'b0	RW	CRC 使能 0: 关闭 1: 使能

注：POLY_WIDTH=1 时，只支持 BIT_ORDER_EN=0。

6.6.2.2. 初始值配置寄存器（CRC_INIT）

Width	Name	Reset	Property	Description
31:0	INIT_VALUE	32'hffffffff	RW	CRC 初始值

6.6.2.3. 取反寄存器（CRC_INV）

Width	Name	Reset	Property	Description
31:0	INV_VALUE	32'hffffffff	RW	CRC 输出结果取反控制 0: 不取反 1: 取反

6.6.2.4. 多项式配置寄存器（CRC_POLY）

Width	Name	Reset	Property	Description
31:0	POLY_VALUE	32'hedb88320	RW	CRC 多项式值，默认为 32 位多项式 注: 写入的值是多项式二进制表示值的反向值，比如多项式 X16+X15+X2+1 的多项式二进制表示是 0x8005，则高低位反转为 0xa001 写入 CRC_POLY 寄存器。

6.6.2.5. 状态寄存器（CRC_STA）

Width	Name	Reset	Property	Description
31:1	Reserved	-	-	-
0	BUSY	1'b0	RO	CRC 忙状态

				0: crc 空闲状态 1: crc 正在计算中

6.6.2.6. 数据寄存器（CRC_DATA）

Width	Name	Reset	Property	Description
31:0	DATA	32'h0	RW	CRC 数据寄存器 写时触发对写入数据的 crc 计算； 读时返回的是 CRC 计算结果

6.6.2.7. 长度寄存器（CRC_LEN）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	CRC_LENGTH	16'h0	RW	CRC 校验数据 byte 长度 （写此寄存器会将 CRC 计算结果清除，并初始化为 CRC_INIT 寄存器的值）

7. 硬件除法运算单元（HWDIV）

7.1. 模块介绍

本硬件除法运算单元能自动执行有符号或无符号的 32 位的整数除法运算。

7.2. 功能特点

- 支持 32 位无符号除法
- 支持 32 位有符号除法

7.3. 功能说明

硬件除法单元包括了 4 个 32 位的数据寄存器，分别为被除数，除数，商和余数。可做有符号或无符号的 32 位除法运算，通过硬件除法控制寄存器的 SIGN 位来进行选择。每一次写入除数寄存器，会自动触发除法运算，可以软件等待 8 个系统时钟周期后去读取结果或者等待状态寄存器的完成标志为 1 后去读取结果。如果除数为零，会产生溢出中断标志位。

7.4. 时钟与复位

7.4.1. 时钟介绍

该模块时钟源为系统时钟，低功耗模式下到达模块的时钟会被自动停止。

7.4.2. 复位介绍

该模块的复位源有两个，分别是系统复位和软件复位，软件复位可通过配置系统寄存器触发。

7.5. 寄存器描述

7.5.1. 寄存器列表

Base address：0x4002 0000

Name	Offset	Reset	Description
HWDIV_DR	0x03A0	32'h0	被除数数据寄存器
HWDIV_SR	0x03A4	32'h0	除数数据寄存器
HWDIV_QUO	0x03A8	32'h0	商结果寄存器
HWDIV_REM	0x03AC	32'h0	余数结果寄存器
HWDIV_CR	0x03B0	32'h0	控制寄存器
HWDIV_STA	0x03B4	32'h0	状态寄存器

7.5.2. 寄存器详细说明

7.5.2.1. 被除数数据寄存器（HWDIV_DR）

Width	Name	Reset	Property	Description
31:0	DIVDR	32'h0	RW	被除数数据

7.5.2.2. 除数数据寄存器（HWDIV_SR）

Width	Name	Reset	Property	Description
31:0	DIVSR	32'h0	RW	除数数据 写入该寄存器后自动触发除法运算。

7.5.2.3. 商结果寄存器（HWDIV_QUO）

Width	Name	Reset	Property	Description
31:0	DIVQUO	32'h0	RO	商结果

7.5.2.4. 余数结果寄存器（HWDIV_REM）

Width	Name	Reset	Property	Description
31:0	DIVREM	32'h0	RO	余数结果

7.5.2.5. 控制寄存器（HWDIV_CR）

Width	Name	Reset	Property	Description
31:3	Reserved	-	-	-
2	UNSIGN	1'b0	RW	运算符号位控制 0: 有符号除法 1: 无符号除法
1	DONE_IE	1'b0	RW	运算结束中断控制位 0: 不使能中断 1: 使能中断
0	DIVBY0_IE	1'b0	RW	除数为 0 中断控制位 0: 不使能中断 1: 使能中断

7.5.2.6. 状态寄存器（HWDIV_STA）

Width	Name	Reset	Property	Description
31:2	Reserved	-	-	-
1	DONE_PEND	1'b0	RW	运算结束标志位 0: 运算未结束 1: 运算结束 此位写 1 或写 DIV_SR 寄存器可清除此标志
0	DIVBY0_PEND	1'b0	RW	除数为 0 标志位 0: 除数不为 0 1: 除数为 0 此位写 1 或写 DIV_SR 寄存器可清除此标志

8. 电源管理（POWER MANAGEMENT）

8.1. 电源

芯片的工作电压为 3.6V~5.5V。本芯片采用 Cap-Less 设计，无需在内置 LDO 输出上外挂电容。内置 LDO 具有 2 挡位下拉电流使能 PMUCON0[18:17]，同时也有低功耗参考电流寄存器 PMUCON0[16:14]。

8.1.1. 电压调节器

复位后调节器总是使能的。在需要低功耗的场合，可以使能低功耗工作模式。

8.2. 电源管理器

8.2.1. 上电复位(POR) 和掉电复位(PDR)

CIU32M011、CIU32M031 内部有一个完整的上电复位(POR)和掉电复位(PDR) 电路，当供电电压达到 3.6V 时系统能正常工作。当 VDD 低于指定的限位电压 VPOR/VPDR 时，系统保持为复位状态，而无需外部复位电路。

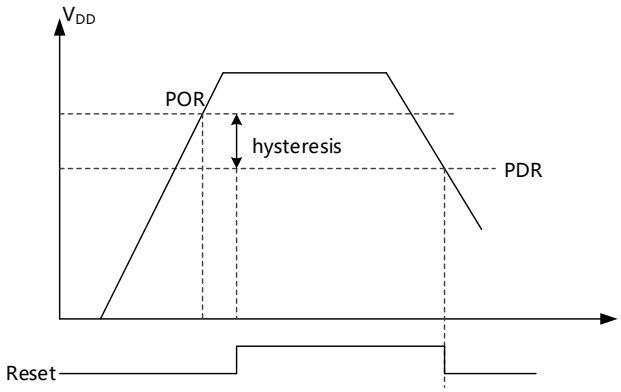


图 8-1 上下电复位波形图

8.2.2. 可编程电压监测器(PVD)

CIU32M011、CIU32M031 内部集成一个外部供电 VCC 电压检测器，检测电压均阈值可选。当系统监测到 VCC 电压低于配置电压值时，可以选择触发系统复位或通过使能 PVD 中断进入中断子函数。这一特性可用于执行紧急关闭任务。检测信号可以选择经过毛刺滤波电路或直接检测，由 LVDCON 的 LVD_VCC_BPS_EN 来控制。

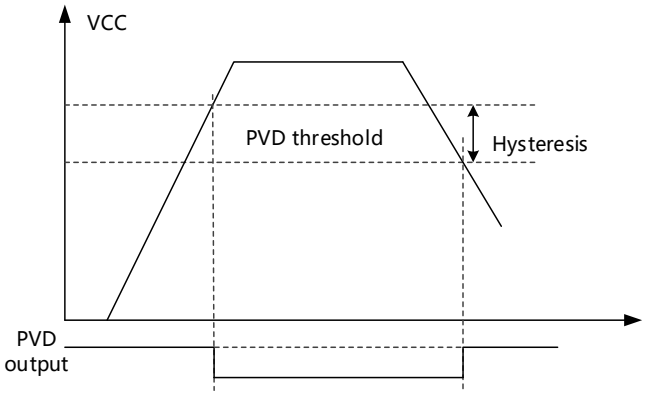


图 8-2 掉电检测波形图

8.3. 电源控制寄存器

8.3.1. 寄存器列表

Base address：0x4002 0000

Name	Offset	Reset	Description
LVDCON	0x0120	32'he51	LVD 控制寄存器。（需要写 SYS_KEY 寄存器才能操作）
LVDCON1	0x0124	32'h0	LVD 控制寄存器 1(需要写 SYS_KEY 寄存器才能操作)

8.3.2. 寄存器详细说明

8.3.2.1. LVD 控制寄存器（LVDCON）

Width	Name	Reset	Property	Description
31	VDD_PEND	1'b0	RC	VDD 电压低于设定阈值触发 PEND，写 1 清除
30	VCC_PEND	1'b0	RC	VCC 电压低于设定阈值触发 PEND，写 1 清除
29	VDD_LVD_ANA_OUTPUT_DIS	1'b0	RW	VDD_LVD 模拟信号输出使能 0: 使能 1: 不使能 先使能 VDD_LVD_EN，再使能 VDD_LVD_ANA_OUTPUT_DIS
28:22	DBS_LO_LIMIT_SET	7'h0	RW	对 PMU_LVD_VDD 信号的低电平毛刺滤波宽度，单位为 lvd_dbs_clk 时钟周期。
21:15	DBS_HI_LIMIT_SET	7'h0	RW	对 PMU_LVD_VDD 信号的高电平毛刺滤波宽度，单位为 lvd_dbs_clk 时钟周期
14	VCC_SYNC_DIS	1'b0	RW	VCC_LVD 在中断模式下唤醒 CPU 的信号是否需要先经过系统时钟同步 0: 同步 1: 不同步
13	VDD_SHIELD_DEBOUNCE_EN	1'b0	RW	屏蔽对 PMU_VDD_LVD 信号的滤波处理 0: 不屏蔽 1: 屏蔽
12	VCC_SHIELD_DEBOUNCE_EN	1'b0	RW	屏蔽对 PMU_VCC_LVD 信号的滤波处理 0: 不屏蔽 1: 屏蔽
11	LVD_DIG_EN	1'b1	RW	LVD 数字开关，使能后才会对 LVD 事件产生复位或者中断 0: 不使能 1: 使能
10	VDD_RST_EN	1'b1	RW	VDD 阈值判断触发后复位系统使能 0: 中断，不复位 1: 复位，不中断
9	VCC_RST_EN	1'b1	RW	VCC 阈值判断触发后复位系统使能 0: 中断，不复位 1: 复位，不中断
8	VCC_LVD_ANA_OUTPUT_DIS	1'b0	RW	VCC_LVD 模拟信号输出使能

				0: 使能 1: 不使能 先使能 VCC_LVD_EN，再使能 VCC_LVD_ANA_OUTPUT_DIS
7:5	VDD_LVD_SET	3'b010	RW	VDD_LVD 电压检测设置(单位 V) b000: 1.15 b001: 1.20 b010: 1.25 b011: 1.30 b100: 1.35 b101: 1.40 b110: 1.45 b111: 1.50
4	VDD_LVD_EN	1'b1	RW	VDD_LVD 模拟开关，使能后才会检测 VDD_LVD 电压 0: 不使能 1: 使能
3:1	VCC_LVD_SET	3'b000	RW	VCC_LVD 电压检测设置(单位 V): b000: 1.80 b001: 2.00 b010: 2.20 b011: 2.40 b100: 2.70 b101: 3.00 b110: 3.60 b111: 4.20
0	VCC_LVD_EN	1'b1	RW	VCC_LVD 模拟开关，使能后才会检测 VCC_LVD 电压 0: 不使能 1: 使能

8.3.2.2. LVD 控制寄存器（LVDCON1）

Width	Name	Reset	Property	Description
31:15	Reserved	-	-	-
14:8	VCC_DBS_LO_LIMIT	7'h0	RW	对 PMU_LVD_VCC 信号的低电平毛刺滤波宽度，单位为 lvd_dbs_clk 时钟周期。
7	Reserved	-	-	-
6:0	VCC_DBS_HI_LIMIT	7'h0	RW	对 PMU_LVD_VCC 信号的高电平毛刺滤波宽度，单位为 lvd_dbs_clk 时钟周期

9. 低功耗（LOW POWER）

9.1. 低功耗模式

在系统或电源复位以后，微控制器处于正常模式运行状态，系统所用时钟为 256KHz 内部 RC 振荡器输出。当 CPU 不需继续运行时，可以利用进入多种低功耗模式来节省功耗。例如等待某个外部事件时，用户需要根据最低电源消耗、最快启动时间和可用的唤醒源等条件，选定一个最佳的低功耗模式。

三种低功耗模式

- 待机模式(Idle Mode)
- 停止模式(Stop Mode)
- 睡眠模式(Sleep Mode)

表 9-1 低功耗模式说明

待机模式(Idle Mode)	停止模式(Stop Mode)	睡眠模式(Sleep Mode)
-	数字模块时钟停止	数字模块时钟停止
-	可选: 片内 HIRC 时钟源关闭(设置 LPCON[3])	可选: 片内 HIRC 时钟源关闭(设置 LPCON[3])
-	可选: SRAM 关闭(设置 LPCON[2])	可选: SRAM 关闭(设置 LPCON[2])
-	-	可选: 片内 256K 时钟源关闭(设置 LPCON[4])
-	-	可选: PMU 进入低功耗模式(设置 LPCON[7])
-	FLASH 进入普通睡眠模式	FLASH 进入深度睡眠模式(功耗更低)
-	-	可选: 唤醒后复位(设置 SYSCON0[30])
CPU 停止	CPU 停止	CPU 停止
__ASM("WFI")指令进入该模式	配置 LPCON[1]进入该模式	配置 LPCON[0]进入该模式

注：可以通过关闭未使用的外设、时钟源使功耗降到最低。

此外，在运行模式下，可以通过以下方式中的一种降低功耗

- 降低系统时钟
- 关闭 APB 和 AHB 总线上未被使用的外设时钟。
- 合理配置 APB 与 AHB 的频率关系

9.2. 进入低功耗

进入低功耗模式(Sleep)步骤：

- Step1: 关闭低功耗下不需要工作的模拟模块
- Step2: 系统时钟切换到 LIRC_256K，并且关闭除 LIRC_256K 以外的时钟源
- Step3: 配置 PMU_LPDOS 使其电压档位比 PMU_HPLDOS 电压档位高
- Step4: 把 PMU_HPXCP, PMU_HPPDI, PMU_HPPDLI 配置为 0
- Step5（可选）：配置 LPCON[7]为 1，并且配置 PMUBK 寄存器设置低功耗下 LPLDO 电压
- Step6: 配置 PMU_HPV2I 为 0（如果执行了 step5，则无需执行 step6）
- Step7: 配置空闲的 IO 为模拟模式。
- Step8: 进入 Sleep 前的准备工作：包括配置唤醒源，初始化唤醒中断服务函数，关闭看门狗(可选)，使能进入 Sleep 时自动关闭 256K 内部 RC 振荡器（可选），使能 SLEEP_GOON_EN 唤醒时不复位（可选）
- Step9: 配置 LP_CON0[0]进入 Sleep

9.3. 低功耗唤醒

支持多种唤醒方式

• 端口唤醒

总共有 4 个 IO 唤醒源（由 WKUP_CON[3:0]控制使能）

这些唤醒源唤醒之后是会产生中断并且有对应的中断状态位，中断是不可屏蔽的（即有中断状态就一定有中断）。另外，由 SLEEP_GOON_EN（SYS_CON0[30]）决定用端口唤醒时是产生系统复位还是继续运行。

IO 唤醒初始化步骤：配置 IO 模式->配置唤醒边沿->清除唤醒标志位->中断初始化使能（根据需要）->使能唤醒位->清除唤醒标志位->配置 LP_CON 进入低功耗模式。后续只需要在进入低功耗模式之前，切换 IO 模式->清除唤醒标志位->进入低功耗模式。

注意：

如果选择了上升沿/下降沿唤醒，而此时 IO 为高电平/低电平，则唤醒标志位马上会置 1。使能后 IO 保持为高电平/低电平不会重复触发，唤醒标志位保持为 1，上升沿/下降沿才会触发。

只要唤醒标志位为 1，则无法进入低功耗模式。

对于唤醒 IO 的电平翻转时间无法确定的应用场景，如果在主程序中使能 wkup_en/int_en，在唤醒中断函数中关闭 wkup_en/int_en，会存在以下风险：使能 wkup_en/int_en 时马上触发 wkup_pend 导致进入中断，然后在中断中关闭 wkup_en/int_en，退出中断后进入低功耗模式，导致无法唤醒/唤醒后不进入唤醒中断。因此唤醒中断要慎重使用。

• 内部源唤醒

支持看门狗（WDT）唤醒，TIMER7_TRGO 唤醒，比较器（COMP）唤醒，flash_wkup(STOP 模式下 prog_ram_done)，LVDVCC 唤醒。

注:

1. 如果不使能 WKUP_CON[3:0]，即只使用内部唤醒源时，进入 sleep 之前，系统时钟要配置成选择内部 LIRC(256KHz)，LIRC 在 sleep 期间不能关闭，且把系统时钟分频配置成大于等于 2.
2. 使能 WKUP_CON[3:1]时，除了对应的 GPIO 可以唤醒外，还复用了内部唤醒源，其中 WKUP_CON[3]复用 LVDVCC 唤醒，WKUP_CON[2]复用比较器唤醒，WKUP_CON[1]复用 TIMER7_TRGO 唤醒。即，当使能了 WKUP_CON[3:1]，但对应的 GPIO 没有配置为相应的 function 模式时，复用的内部唤醒源可以代替 GPIO 触发唤醒。

10. 复位和时钟系统（RESET/CLOCK)

10.1. 复位

CIU32M011、CIU32M031 支持系统复位、电源复位和主复位。

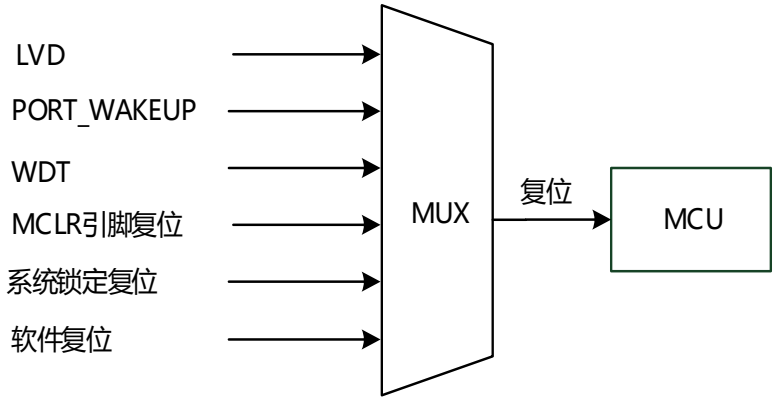


图 10-1 复位源框图

10.1.1. 系统复位

系统复位将复位除某些复位状态寄存器和特殊功能寄存器之外的所有寄存器。

当以下事件中的一件发生时，产生一个系统复位

- SLEEP 模式下外部 IO 口唤醒
- WDT 计数溢出复位
- 系统锁定复位

10.1.2. 主复位

主复位能将部分系统复位无法复位的寄存器复位。

以下事件可以触发一个主复位

- 软件复位
- PVD 检测到电压低事件，且控制器处于复位功能模式

10.1.3. 电源复位

上电/掉电复位(POR/PDR 复位)都属于电源复位。电源复位将复位所有的逻辑和模拟模块。复位入口矢量被固定在地址 0x0000_0004。

10.2. 时钟

10.2.1. 模块框图

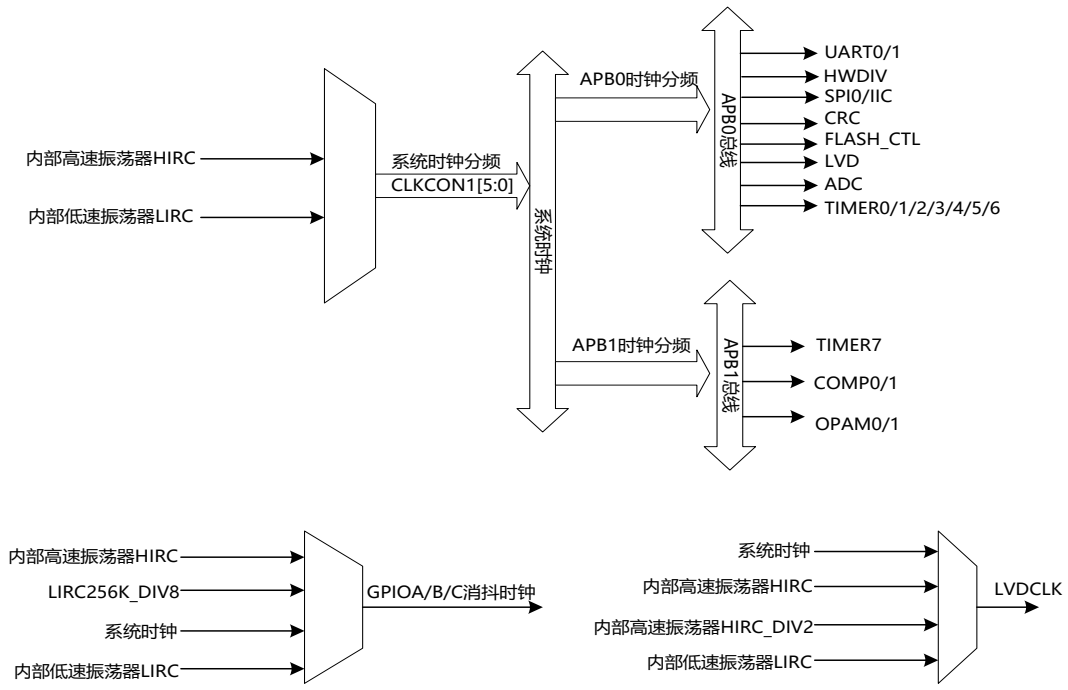


图 10-2 时钟模块

10.2.2. HIRC 时钟

HIRC 时钟信号由内部 72MHz 的振荡器产生，HIRC 振荡器能够在不需要任何外部器件的条件下提供系统时钟。HIRC 需要频率校准，校准值写在 FLASH 系统存储区域。在程序使用这个时钟前，可以读取并配置出高精度的 HIRC 时钟。经过出厂校验后，正常温度范围下 HIRC 精度为 72MHz(±1.5%)，具体请参考 *电气特性参数* 。

10.2.3. LIRC 时钟

LIRC 振荡器担当一个低功耗时钟源的角色，它作为系统启动时钟为其他单元提供时钟。LIRC 时钟频率大约 256KHz。

10.2.4. 系统时钟(SYSCLK)

两种不同的时钟源可被用来驱动系统时钟(SYSCLK)

- 内部低速 256KHz LIRC
- 内部高速 72MHz(±1.5%) 高速振荡器

10.2.5. 毛刺滤波时钟源选择

三种不同的时钟源可被用来驱动 GPIO 的毛刺滤波时钟

- 内部高速 HIRC 的分频时钟
- 内部低速 LIRC_256K 8 分频时钟
- 系统时钟
- 内部低速 LIRC_256K

当不被使用时，任一个时钟源都可被独立地启动或关闭，由此优化系统功耗。

11. 通用输入输出（GPIO）

11.1. 模块介绍

每组 GPIO 端口有四个 32 位配置寄存器(GPIOx_MODE,GPIOx_OTYPE, GPIOx_OSPEED and GPIOx_PUPD)，两个 32 位数据寄存器(GPIOx_IDAT and GPIOx_ODAT)，一个 32 位置位/复位寄存器(GPIOx_BSR)和一个 32 位翻转寄存器(GPIOx_TGL)。另外，所有 GPIO 有两个复用功能选择寄存器(GPIOx_AFRH and GPIOx_AFRL)。

注：GPIOx 中的 x 表示 GPIO 组数。

11.2. 功能特点

- 输出状态：推挽或开漏(上下拉)
- 输出寄存器状态值(GPIOx_ODAT) 或者复用功能输出
- 输入状态：浮空、上下拉、模拟
- 输入数据到数据寄存器(GPIOx_IDAT) 或复用功能输入
- 独立置位/复位/翻转 IO 状态(GPIOx_ BSR、GPIOx_TGL)
- 模拟功能
- 复用功能(开漏或推挽、上拉或下拉)

11.3. 功能说明

GPIO 的每一个端口可以通过软件独立配置成下面状态

- 输入浮空
- 输入上拉
- 输入下拉
- 模拟功能
- 开漏输出(上拉或下拉)
- 推挽输出
- 复用功能(开漏或推挽、上拉或下拉)

表 11-1 GPIO 端口位配置表

MODE/模式	OTYPE/输出类型	OSPEED/驱动能力		PUPD/上下拉		IO configuration/端口配置	
01	0	SPEED		0	0	GP output	PP
	0			0	1	Reserved	
	0			1	0		
	0			1	1		
	1			0	0	GP output	OD
	1			0	1	GP output	OD+PU
	1			1	0	Reserved	
	1			1	1		
10	0	SPEED		0	0	AF	PP
	0			0	1	Reserved	
	0			1	0		
	0			1	1		
	1			0	0	AF	OD
	1			0	1	AF	OD+PU
	1			1	0	Reserved	
	1			1	1		
00	x	x	x	0	0	Input	PP

	x	x	x	0	1	Input	PP+PU
	x	x	x	1	0	Input	PP+PD
	x	x	x	1	1	Reserved	
11	x	x	x	0	0	Input/Output	Analog
	x	x	x	0	1	Reserved	
	x	x	x	1	0		
	x	x	x	1	1		

注： GP = Generate purpose（通用），PP = Push pull(推挽)，PU = Pull up(上拉)，PD = Pull down(下拉)，OD = Open drain(开漏)，AF = Alternate function(复用)，SPEED = 驱动能力。

11.3.1. 通用 IO（GPIO）

复位期间和刚复位后，复用功能未开启，IO 端口被配置成浮空输入模式。

当作为输出配置时，写到输出数据寄存器上的值（GPIOx_ ODAT）输出到相应的 IO 引脚。可以以推挽或开漏模式使用输出驱动器。

输入数据寄存器（GPIOx_ IDAT）在每个 APB 时钟周期捕捉 IO 引脚上的数据。

所有 GPIO 引脚有一个内部弱上拉，当配置为输入时，它们可以被激活也可以被断开。

11.3.2. 单独的位操作

当对 GPIOx_ ODAT 的个别位编程时，软件不需要禁止中断：在单次 APB 写操作里，可以只更改一个或多个位。只需要通过对“置位/复位寄存器”（GPIOx_BSR）或“取反寄存器”（GPIOx_TGL）中想要更改的位写“1”来实现。没被选择的位将不被更改。

11.3.3. 复用功能（AF）

芯片 IO 引脚通过多路选择器连接到片内外设，每个 IO 上同一时刻只能选通一个复用功能。每个 IO 引脚有一个 2 输入的多路选择器连接到复用功能（AF0~AF1），通过配置 GPIOx_AFRH/L 选择功能。如果把端口配置成复用输出功能，则引脚和输出寄存器断开，并和片上外设的输出信号连接。如果软件把一个 GPIO 脚配置成复用输出功能，但是外设没有被激活，它的输出将不确定。

11.3.4. GPIO 锁定机制

锁定机制允许在 GPIO 控制寄存器 GPIOx_LCK 上执行一串锁定程序，然后把 GPIO 的状态锁定，一旦 GPIO 状态被锁定，将不可改变，直到复位。被锁定的寄存器有（GPIOx_MODE, GPIOx_OTYPE,GPIOx_OSPEED, GPIOx_PUPD, GPIOx_AFRL and GPIOx_AFRH）。

锁定序列参考 *GPIOx_LCK 寄存器描述* 。

11.3.5. 输入配置

当 IO 端口配置为输入时

- 输出缓存器被禁止
- 施密特触发输入被激活
- 根据输入配置（上拉、下拉或浮空）的不同，弱上拉和下拉电阻被连接
- 出现在 IO 脚上的数据在每个 APB 时钟被采样到输入数据寄存器
- 对输入数据寄存器的读访问可得到 IO 状态

11.3.6. 输出配置

当 IO 端口被配置为输出时

- 输出缓冲器被激活

开漏模式：输出寄存器上的“0”激活 N-MOS，而输出寄存器上的“1”将端口置于高阻态（P-MOS 从不被激活）

推挽模式：输出寄存器上的“0”激活 N-MOS，而输出寄存器上的“1”将激活 P-MOS。

- 施密特触发输入被激活
- 弱上拉和下拉电阻被禁止

- 出现在 IO 脚上的数据在每个 APB 时钟被采样到输入数据寄存器
- 在开漏模式时，对输入数据寄存器的读访问可得到 IO 状态
- 在推挽模式时，对输出数据寄存器的读访问得到最后一次写的值

11.3.7. 模拟输入配置

当 IO 端口被配置为模拟输入配置时

- 输出缓存器被禁止
- 禁止施密特触发输入，实现了每个模拟 IO 引脚上的零消耗。施密特触发输出值被强制为“0”
- 弱上拉和下拉电阻被禁止
- 读取输入数据寄存器时数值为“0”

11.3.8. 复用功能配置

对 IO 端口进行编程作为复用功能时

- 在开漏或推挽式配置中，输出缓冲器被打开
- 内置外设的信号驱动输出缓冲器（复用功能输出）
- 施密特触发输入被激活
- 弱上拉和下拉电阻被禁止
- 在每个 APB 时钟周期，出现在 IO 脚上的数据被采样到输入数据寄存器
- 开漏模式时，读输入数据寄存器时可得到 IO 口状态
- 在推挽模式时，读输出数据寄存器时可得到最后一次写的值

11.4. 模块框图

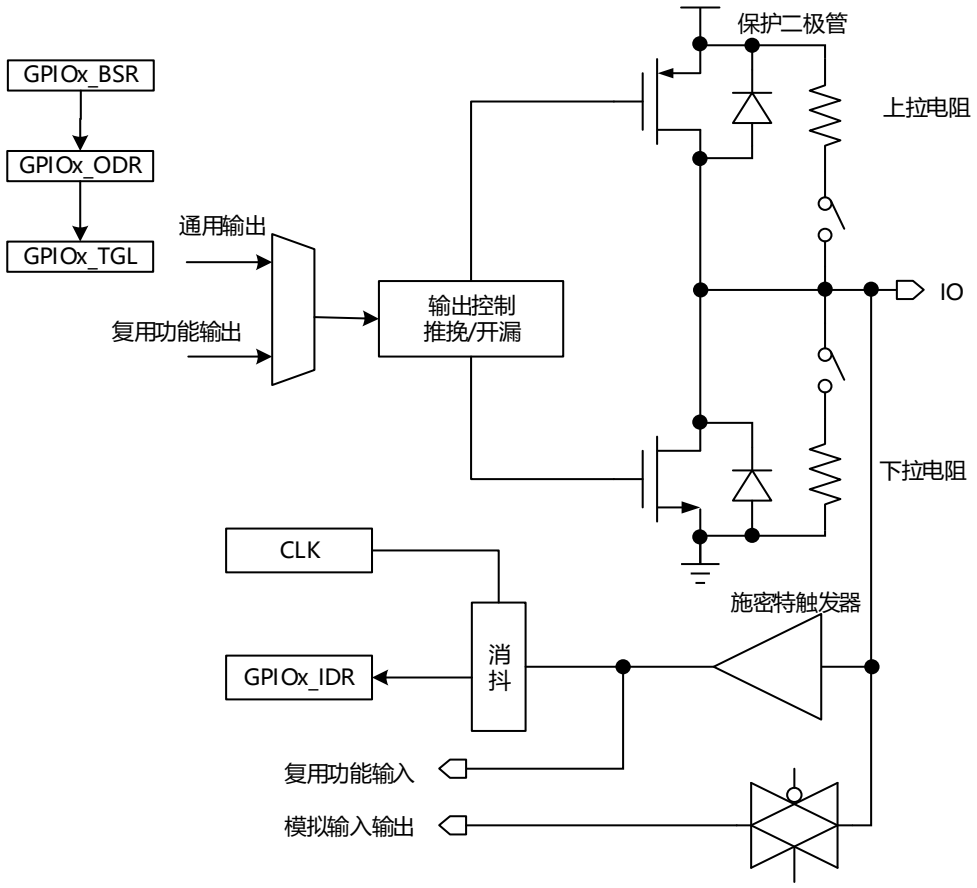


图 11-1 GPIO 模块框图

11.5. 寄存器描述

11.5.1. 寄存器列表

Base address：0x4002 0000

Name	Offset（A/B/C)	Reset	Description
GPIOx_MODE	0x00A0/0x00E0/0x120	32'h0	GPIO 端口模式寄存器
GPIOx_OTYPE	0x00A4/0x00E4/0x124	32'h0	GPIO 端口输出类型寄存器
GPIOx_OSPEED	0x00A8/0x00E8/0x128	32'h0	GPIO 端口驱动能力档位寄存器
Reserved	0x00AC/0x00EC/0x12C	-	-
GPIOx_PUPD	0x00B0/0x00F0/0x130	32'h0	GPIO 端口上拉/下拉寄存器
GPIOx_IDAT	0x00B4/0x00F4/0x134	32'h0	GPIO 端口输入数据寄存器
GPIOx_ODAT	0x00B8/0x00F8/0x138	32'h0	GPIO 端口输出数据寄存器
GPIOx_BSR	0x00BC/0x00FC/0x13C	32'h0	GPIO 端口置位/复位寄存器
GPIOx_LCK	0x00C0/0x0100/0x140	32'h0	GPIO 端口锁定控制寄存器
GPIOx_AFRL	0x00C4/0x0104/0x144	32'h0	GPIO 端口复用功能低位寄存器
GPIOx_AFRH	0x00C8/0x0108/0x148	32'h0	GPIO 端口复用功能高位寄存器
GPIOx_TGL	0x00CC/0x010C/0x14C	32'h0	GPIO 端口取反寄存器
GPIOx_IMK	0x00D0/0x0110/0x150	32'h0	GPIO 端口中断寄存器
GPIOx_TGPEND	0x00D4/0x0114/0x154	32'h0	GPIO 端口标记寄存器
GPIOx_IE_EN	0x00D8/0x0118/0x158	32'h0	GPIO 端口输入控制寄存器
GPIOx_TG_EDGE	0x00DC/0x011C/0x15C	32'h0	GPIO 端口翻转沿控制寄存器

11.5.2. 寄存器详细说明

11.5.2.1. GPIO 端口模式寄存器（GPIOx_MODE） (X=A..C)

Width	Name	Reset	Property	Description
31:30	MODER15	2'b0	RW	MODERy[1:0]: GPIOx 的模式位（y=0..15） b00: 输入模式（复位状态） b01: 通用输出模式 b10: 复用功能模式 b11: 模拟模式
29:28	MODER14	2'b0	RW	
27:26	MODER13	2'b0	RW	
25:24	MODER12	2'b0	RW	
23:22	MODER11	2'b0	RW	
21:20	MODER10	2'b0	RW	
19:18	MODER9	2'b0	RW	
17:16	MODER8	2'b0	RW	
15:14	MODER7	2'b0	RW	
13:12	MODER6	2'b0	RW	
11:10	MODER5	2'b0	RW	
9:8	MODER4	2'b0	RW	
7:6	MODER3	2'b0	RW	
5:4	MODER2	2'b0	RW	
3:2	MODER1	2'b0	RW	
1:0	MODER0	2'b0	RW	

11.5.2.2. GPIO 端口输出类型寄存器（GPIOx_OTYPE） (X=A..C)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	OT15	1'b0	RW	OTy: GPIOx 的输出类型选择位（y = 0..15) 0: 推挽输出（复位状态） 1: 开漏输出
14	OT14	1'b0	RW	
13	OT13	1'b0	RW	
12	OT12	1'b0	RW	
11	OT11	1'b0	RW	
10	OT10	1'b0	RW	
9	OT9	1'b0	RW	
8	OT8	1'b0	RW	
7	OT7	1'b0	RW	
6	OT6	1'b0	RW	
5	OT5	1'b0	RW	
4	OT4	1'b0	RW	
3	OT3	1'b0	RW	
2	OT2	1'b0	RW	
1	OT1	1'b0	RW	
0	OT0	1'b0	RW	

11.5.2.3. GPIO 端口驱动能力档位寄存器（GPIOx_OSPEED） (X=A..C)

Width	Name	Reset	Property	Description
31:30	OSPEED15	1'b0	RW	OSPEEDy: GPIOx 驱动能力档位选择位（ 对于 GPIOA，y=0..15； 对于 GPIOB，y=0..15； 对于 GPIOC，y=0..11） b00: 最低档 b01: 中低档 b10: 中高档 b11: 最高档
29:28	OSPEED14	1'b0	RW	
27:26	OSPEED13	1'b0	RW	
25:24	OSPEED12	1'b0	RW	
23:22	OSPEED11	1'b0	RW	
21:20	OSPEED10	1'b0	RW	
19:18	OSPEED9	1'b0	RW	
17:16	OSPEED8	1'b0	RW	
15:14	OSPEED7	1'b0	RW	
13:12	OSPEED6	1'b0	RW	
11:10	OSPEED5	1'b0	RW	
9:8	OSPEED4	1'b0	RW	
7:6	OSPEED3	1'b0	RW	
5:4	OSPEED2	1'b0	RW	
3:2	OSPEED1	1'b0	RW	
1:0	OSPEED0	1'b0	RW	

11.5.2.4. GPIO 端口上拉/下拉寄存器（GPIOx_PUPD） (X=A..C)

Width	Name	Reset	Property	Description
31	PD15	1'b0	RW	PDy: GPIOx 的下拉使能位（y = 0..15） 0: 不使能 1: 使能 （输入模式下使用）
30	PD14	1'b0	RW	
29	PD13	1'b0	RW	
28	PD12	1'b0	RW	
27	PD11	1'b0	RW	
26	PD10	1'b0	RW	
25	PD9	1'b0	RW	
24	PD8	1'b0	RW	
23	PD7	1'b0	RW	
22	PD6	1'b0	RW	
21	PD5	1'b0	RW	
20	PD4	1'b0	RW	
19	PD3	1'b0	RW	
18	PD2	1'b0	RW	
17	PD1	1'b0	RW	
16	PD0	1'b0	RW	
15	PU15	1'b0	RW	PUy: GPIOx 的上拉使能位（y = 0..15） 0: 不使能 1: 使能 （输入模式或开漏输出模式下使用）
14	PU14	1'b0	RW	
13	PU13	1'b0	RW	
12	PU12	1'b0	RW	
11	PU11	1'b0	RW	
10	PU10	1'b0	RW	
9	PU9	1'b0	RW	
8	PU8	1'b0	RW	
7	PU7	1'b0	RW	
6	PU6	1'b0	RW	
5	PU5	1'b0	RW	
4	PU4	1'b0	RW	
3	PU3	1'b0	RW	
2	PU2	1'b0	RW	
1	PU1	1'b0	RW	
0	PU0	1'b0	RW	

11.5.2.5. GPIO 端口输入数据寄存器（GPIOx_IDAT） (X=A..C)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	IDAT15	1'b0	RO	IDATy: GPIOx 的输入数据位(y = 0..15) 这些位为只读，它们代表相应 IO 端口的输入值
14	IDAT14	1'b0	RO	
13	IDAT13	1'b0	RO	
12	IDAT12	1'b0	RO	
11	IDAT11	1'b0	RO	
10	IDAT10	1'b0	RO	
9	IDAT9	1'b0	RO	

8	IDAT8	1'b0	RO	
7	IDAT7	1'b0	RO	
6	IDAT6	1'b0	RO	
5	IDAT5	1'b0	RO	
4	IDAT4	1'b0	RO	
3	IDAT3	1'b0	RO	
2	IDAT2	1'b0	RO	
1	IDAT1	1'b0	RO	
0	IDAT0	1'b0	RO	

11.5.2.6. GPIO 端口输出数据寄存器(GPIOx_ODAT) (X=A..C)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	ODAT15	1'b0	RW	
14	ODAT14	1'b0	RW	
13	ODAT13	1'b0	RW	
12	ODAT12	1'b0	RW	
11	ODAT11	1'b0	RW	
10	ODAT10	1'b0	RW	
9	ODAT9	1'b0	RW	ODATy: GPIOx 的输出数据位(y = 0..15)
8	ODAT8	1'b0	RW	ODAT 寄存器的这些位可以单独设置或者通过写
7	ODAT7	1'b0	RW	入 GPIOx_BSR 中的 BS/BR 对这组寄存器进行置位
6	ODAT6	1'b0	RW	/复位。
5	ODAT5	1'b0	RW	
4	ODAT4	1'b0	RW	
3	ODAT3	1'b0	RW	
2	ODAT2	1'b0	RW	
1	ODAT1	1'b0	RW	
0	ODAT0	1'b0	RW	

11.5.2.7. GPIO 端口置位/复位寄存器(GPIOx_BSR) (X=A..C)

Width	Name	Reset	Property	Description
31	BR15	1'b0	WO	
30	BR14	1'b0	WO	
29	BR13	1'b0	WO	
28	BR12	1'b0	WO	BRy: GPIOx 的清除位(y = 0..15)
27	BR11	1'b0	WO	这些位只写，对这些位进行读操作将返回值
26	BR10	1'b0	WO	0x0000。
25	BR9	1'b0	WO	0: 对应的 ODATx 位没有动作
24	BR8	1'b0	WO	1: 复位相应的 ODATx 位
23	BR7	1'b0	WO	Note: 如果设置了 BSx 和 BRx，则 BRx 具有优先
22	BR6	1'b0	WO	级。
21	BR5	1'b0	WO	
20	BR4	1'b0	WO	
19	BR3	1'b0	WO	

18	BR2	1'b0	WO	
17	BR1	1'b0	WO	
16	BR0	1'b0	WO	
15	BS15	1'b0	WO	BSy: GPIOx 的设置位(y = 0..15) 这些位只写，对这些位进行读操作将返回值 0x0000。 0: 对应的 ODATx 位没有动作 1: 置位相应的 ODATx 位
14	BS14	1'b0	WO	
13	BS13	1'b0	WO	
12	BS12	1'b0	WO	
11	BS11	1'b0	WO	
10	BS10	1'b0	WO	
9	BS9	1'b0	WO	
8	BS8	1'b0	WO	
7	BS7	1'b0	WO	
6	BS6	1'b0	WO	
5	BS5	1'b0	WO	
4	BS4	1'b0	WO	
3	BS3	1'b0	WO	
2	BS2	1'b0	WO	
1	BS1	1'b0	WO	
0	BS0	1'b0	WO	

11.5.2.8. GPIO 端口锁定控制寄存器(GPIOx_LCK)（X=A..C)

该寄存器用于锁定 GPIO 配置寄存器，受影响的寄存器有(GPIOx_MODE, GPIOx_OTYPE,GPIOx_OSPEED, GPIOx_PUPD, GPIOx_AFRL and GPIOx_AFRH)。必须按照正确的序列操作 GPIOx_LCK 寄存器才能使锁定机制生效，其中 GPIOx_LCK[15:0]每 bit 对应一个 IO，在锁定序列中，GPIOx_LCK[15:0]的值不能改变，锁定机制生效后，GPIOx_LCK 寄存器也不能再被修改，直到系统复位或者 GPIO 复位。

Width	Name	Reset	Property	Description
31:17	Reserved	-	-	-
16	LCKK	0	RW	LCKK: (Lock key) 该 bit 任意时刻可读，但是只有按照正确锁定序列写入时才能使其变 1. 0: GPIO 锁定机制无效 1: GPIO 锁定机制生效，GPIOx_LCK 寄存器本身也被锁定，直到系统复位或者 GPIO 复位。 锁定序列如下: Write LCKK = ‘1’ + LCK[15:0] Write LCKK = ‘0’ + LCK[15:0] Write LCKK = ‘1’ + LCK[15:0] Read GPIOx_LCK Read LCKK = ‘1’ (该步骤可选，但是读到 LCKK 为 1 可以确认锁定机制是否生效) 注意：在锁定序列期间 LCK[15:0]的值不可改变，在序列期间任意步骤不满足，则终止锁定。锁定机制生效后，读 LCKK 将返回“1”，直到系统复位或者 GPIO 复位。
15	LCK15	0	RW	每 bit 对应一个 IO 的锁定使能，这些位只有在
14	LCK14	0	RW	

13	LCK13	0	RW	LCKK 为“0”时可写，任意时刻可读。
12	LCK12	0	RW	0: 不锁定
11	LCK11	0	RW	1: 锁定
10	LCK10	0	RW	
9	LCK9	0	RW	
8	LCK8	0	RW	
7	LCK7	0	RW	
6	LCK6	0	RW	
5	LCK5	0	RW	
4	LCK4	0	RW	
3	LCK3	0	RW	
2	LCK2	0	RW	
1	LCK1	0	RW	
0	LCK0	0	RW	

11.5.2.9. GPIO 端口复用功能低位寄存器(GPIOx_AFRL) (X=A..C)

Width	Name	Reset	Property	Description
31:28	AFR7	4'b0	RW	AFRy: GPIOx[y]的复用通道选择位(
27:24	AFR6	4'b0	RW	对于 GPIOA，y=0..7；
23:20	AFR5	4'b0	RW	对于 GPIOB，y=0..7；
19:16	AFR4	4'b0	RW	对于 GPIOC，y=0..7)
15:12	AFR3	4'b0	RW	bx000: AF0
11:8	AFR2	4'b0	RW	bx001: AF1
7:4	AFR1	4'b0	RW	.....
3:0	AFR0	4'b0	RW	bx111: AF7

11.5.2.10. GPIO 端口复用功能高位寄存器(GPIOx_AFRH) (X=A..C)

Width	Name	Reset	Property	Description
31:28	AFR15	4'b0	RW	AFRy: GPIOx[y]的复用通道选择位(
27:24	AFR14	4'b0	RW	对于 GPIOA，y=8..15；
23:20	AFR13	4'b0	RW	对于 GPIOB，y=8..15；
19:16	AFR12	4'b0	RW	对于 GPIOC，y=8..11)
15:12	AFR11	4'b0	RW	bx000: AF0
11:8	AFR10	4'b0	RW	bx001: AF1
7:4	AFR9	4'b0	RW	.....
3:0	AFR8	4'b0	RW	bx111: AF7

11.5.2.11. GPIO 端口取反寄存器(GPIOx_TGL) (X=A..C)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	TG15	1'b0	WO	TGy: GPIOx 的翻转位(y = 0..15)
14	TG14	1'b0	WO	这些位是只写的。对这些位进行读操作将返回值
13	TG13	1'b0	WO	0x0000
12	TG12	1'b0	WO	0: 对应的 ODATx 位没有动作
11	TG11	1'b0	WO	1: 翻转相应的 ODATx 位的输出电平

10	TG10	1'b0	WO	Note: 如果设置了 BSx BRx 和 TGx，则 BRx 具有第一优先级，BSx 具有第二优先级。
9	TG9	1'b0	WO	
8	TG8	1'b0	WO	
7	TG7	1'b0	WO	
6	TG6	1'b0	WO	
5	TG5	1'b0	WO	
4	TG4	1'b0	WO	
3	TG3	1'b0	WO	
2	TG2	1'b0	WO	
1	TG1	1'b0	WO	
0	TG0	1'b0	WO	

11.5.2.12. GPIO 端口中断寄存器(GPIOx_IMK ）(X=A..C)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	IMK15	1'b0	RW	IMKy: GPIOx 的中断使能位(y = 0..15) 如果 IMKx 使能，那么当对应的 TGPEND 位为 1 时，会产生中断 0: 中断不使能 1: 中断使能
14	IMK14	1'b0	RW	
13	IMK13	1'b0	RW	
12	IMK12	1'b0	RW	
11	IMK11	1'b0	RW	
10	IMK10	1'b0	RW	
9	IMK9	1'b0	RW	
8	IMK8	1'b0	RW	
7	IMK7	1'b0	RW	
6	IMK6	1'b0	RW	
5	IMK5	1'b0	RW	
4	IMK4	1'b0	RW	
3	IMK3	1'b0	RW	
2	IMK2	1'b0	RW	
1	IMK1	1'b0	RW	
0	IMK0	1'b0	RW	

11.5.2.13. GPIO 端口标志寄存器(GPIOx_TGPEND ）(X=A..C)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	TGPEND15	1'b0	RC	TGPENDy: GPIOx 的输入边沿检测标志位(y = 0..15) 根据 GPIOx_TG_EDGE 的配置决定检测 IO 输入的哪个边沿。 0: 检测 IO 输入边沿无效 1: 检测 IO 输入边沿有效 软件可写 1 清 0
14	TGPEND14	1'b0	RC	
13	TGPEND13	1'b0	RC	
12	TGPEND12	1'b0	RC	
11	TGPEND11	1'b0	RC	
10	TGPEND10	1'b0	RC	
9	TGPEND9	1'b0	RC	
8	TGPEND8	1'b0	RC	
7	TGPEND7	1'b0	RC	
6	TGPEND6	1'b0	RC	

5	TGPEND5	1'b0	RC	
4	TGPEND4	1'b0	RC	
3	TGPEND3	1'b0	RC	
2	TGPEND2	1'b0	RC	
1	TGPEND1	1'b0	RC	
0	TGPEND0	1'b0	RC	

11.5.2.14. GPIO 端口输入使能寄存器(GPIOx_IE_EN) (X=A..C)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	GPIOx15_IE_EN	1'b0	RW	GPIOx_IE_ENy: GPIOx 的输入（IE）强制使能控制位 (y = 0..15) 0: 不强制使能 IE 1: 强制使能 IE
14	GPIOx14_IE_EN	1'b0	RW	
13	GPIOx13_IE_EN	1'b0	RW	
12	GPIOx12_IE_EN	1'b0	RW	
11	GPIOx11_IE_EN	1'b0	RW	
10	GPIOx10_IE_EN	1'b0	RW	
9	GPIOx9_IE_EN	1'b0	RW	
8	GPIOx8_IE_EN	1'b0	RW	
7	GPIOx7_IE_EN	1'b0	RW	
6	GPIOx6_IE_EN	1'b0	RW	
5	GPIOx5_IE_EN	1'b0	RW	
4	GPIOx4_IE_EN	1'b0	RW	
3	GPIOx3_IE_EN	1'b0	RW	
2	GPIOx2_IE_EN	1'b0	RW	
1	GPIOx1_IE_EN	1'b0	RW	
0	GPIOx0_IE_EN	1'b0	RW	

11.5.2.15. GPIO 端口边沿检测寄存器(GPIOx_TG_EDGE) (X=A..C)

Width	Name	Reset	Property	Description
31:30	TG_DEGE15	2'b0	RW	TG_EDGEy: GPIOx 的输入边沿的检测控制位(y = 0..15) b00: 不检测边沿 b01: 检测上升沿 b10: 检测下降沿 b11: 检测上升沿与下降沿
29:28	TG_DEGE14	2'b0	RW	
27:26	TG_DEGE13	2'b0	RW	
25:24	TG_DEGE12	2'b0	RW	
23:22	TG_DEGE11	2'b0	RW	
21:20	TG_DEGE10	2'b0	RW	
19:18	TG_DEGE9	2'b0	RW	
17:16	TG_DEGE8	2'b0	RW	
15:14	TG_DEGE7	2'b0	RW	
13:12	TG_DEGE6	2'b0	RW	
11:10	TG_DEGE5	2'b0	RW	
9:8	TG_DEGE4	2'b0	RW	
7:6	TG_DEGE3	2'b0	RW	
5:4	TG_DEGE2	2'b0	RW	
3:2	TG_DEGE1	2'b0	RW	
1:0	TG_DEGE0	2'b0	RW	

12. 同步串行接口（SSP）

12.1. 模块介绍

SPI_IIC 模块可用作 SPI 接口通信和 IIC 接口通信，两种功能同一时间只能选择其中一种使用。该模块集成两种接口协议，节省资源的同时又能满足不同的应用需求。

12.2. 功能特点

12.2.1. SPI 功能

- 支持主模式和从模式工作
- 支持全双工收发
- 可编程时钟极性，采样相位，支持 4 种模式
- 支持 1~32bit 传输
- 支持 5byte 发送/接收数据缓冲
- 传输数据顺序 MSB 和 LSB
- 支持标准模式，三线模式
- 可触发中断的专用发送和接收标志

12.2.2. IIC 功能

- 支持主模式和从模式
- 主模式支持时钟同步和总线仲裁
- 从模式支持在发送数据没有准备好或者接收缓冲器满时候拉低 SCL
- 从模式支持 7bit 地址或者 10bit 地址
- 从模式支持接收广播地址
- 支持 5byte 发送/接收数据缓冲

12.3. 功能说明

12.3.1. SPI 工作模式

- 模式 0：时钟空闲为 0，上升沿采样，下降沿出数据
- 模式 1：时钟空闲为 0，下降沿采样，上升沿出数据
- 模式 2：时钟空闲为 1，下降沿采样，上升沿出数据
- 模式 3：时钟空闲为 1，上升沿采样，下降沿出数据

12.3.2. SPI 接口模式

- 标准模式：通信线有 CLK,CS,IO0(MOSI),IO1(MISO),一个 CLK 传输 1bit 数据
- 三线模式：通信线有 CLK,CS,IO0,接收和发送都通过 IO0，一个 CLK 传输 1bit 数据

12.3.3. SPI 数据帧与内部缓存

SPI 可支持 1~32bit 帧数据传输，内部集成了一个 40bit 的缓冲区，根据配置的数据帧长度不同，缓冲区能缓存的帧数也不一样。数据帧长度<=8bit 时，缓冲区可容纳 5 帧数据，8bit<数据帧长度<=16bit 时，缓冲区可容纳两帧数据，数据帧长度>16bit 时，缓冲区可容纳 1 帧数据，当缓冲区无法再容下一帧数据时，缓冲区满标志会置 1。

12.3.4. IIC 主机时钟同步和总线仲裁

IIC 主机模式时，在多主机的应用场景下，支持时钟同步和总线仲裁。当总线上连接了不止一个主机时，就会存在同时发起通信的情况，这时候需要时钟同步以及总线仲裁机制决定由哪个主机占用总线完成数据传输。

时钟同步的原理：IIC 总线上的不同主机可能发起传输时的时钟频率不一样，通过时钟同步机制，可

以让所有主机的时钟同步，才能进行逐位仲裁。所有主机的 SCL 在总线上是线与的关系，当总线上的 SCL 由高切换到低电平时，所有主机从 0 开始计算低电平周期时间。当电平时间达到时，如果总线上的其它主机的 SCL 低电平仍然保持，那么其它主机进入高电平等待状态，等低电平时间最长的主机的 SCL 拉高时再统一拉高。因此，总线上同步后的 SCL 的低电平时间由低电平周期最长的主机决定，而高电平时间由高电平周期最短的主机决定。

总线仲裁原理：IIC 总线上不同主机的 SDA 线也是线与的关系，各主机在 SCL 线为高电平时，检查 SDA 线的电平是否和自己发送的 SDA 信号一致，如果检测到 SDA 线为低电平时，自己要发送的 SDA 信号为高电平，那么该主机仲裁失败，停止总线上的传输动作。

12.3.5. IIC 从机拉低 SCL

IIC 从机在发送状态下如果缓冲区中没有可发送的数据，或者在接收状态下缓冲区已满时，将会在 SCL 端口输出低电平，拉低总线上的 SCL 信号使主机暂停发送时钟。当从机准备好发送或者接收后，将在 SCL 输出高电平，主机又可以重新控制总线上的 SCL 线，恢复数据传输。

12.3.6. IIC 从机支持 7bit/10bit 寻址

IIC 从机支持 7bit 或者 10bit 寻址模式，由寄存器 CON0 的 IIC_SLAVE_ADDR_WIDTH 位决定。

7bit 模式下，主机需要在 TX 模式下发送带 START 位的 7bit 地址，最后 1bit 为 R/W 标志。当主机写从机时，寻址完成即可进行数据发送。当主机读从机时，寻址完成后，主机需要改成 RX 模式，然后配置准备接收的数据长度（寄存器 DMA_LEN），然后对寄存器 CMD_DATA 写任意值启动接收。

10bit 模式下，主机需要在 TX 模式下发送带 START 位的第 1byte 地址（此时 R/W 位为 1），接着发送第 2byte 地址，此时如果收到从机的 ACK 信号，则为寻址成功。接下来如果是主机写从机，就可以直接进行数据发送。如果是主机读从机，那么需要主机再次发送带 START 为的第 1byte 地址（此时 R/W 为 1），然后切换成 RX 模式，配置寄存器 DMA_LEN，并且对寄存器 CMD_DATA 写任意值启动读数据。

主机读从机时，从机被寻址成功且收到读标志时，需要切换成 TX 模式，并且往缓冲区中写入要发送的数据。

12.4. IO 映射

详细参考 3.2. 管脚定义

12.5. 模块框图与接口时序

12.5.1. 模块架构图

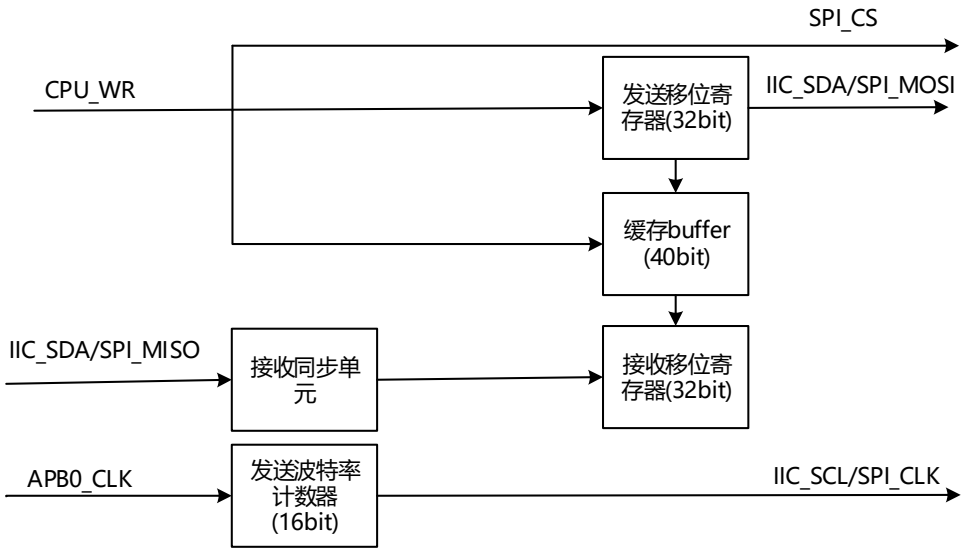


图 12-1 SPI/IIC 模块架构图

12.5.2. SPI 时序图

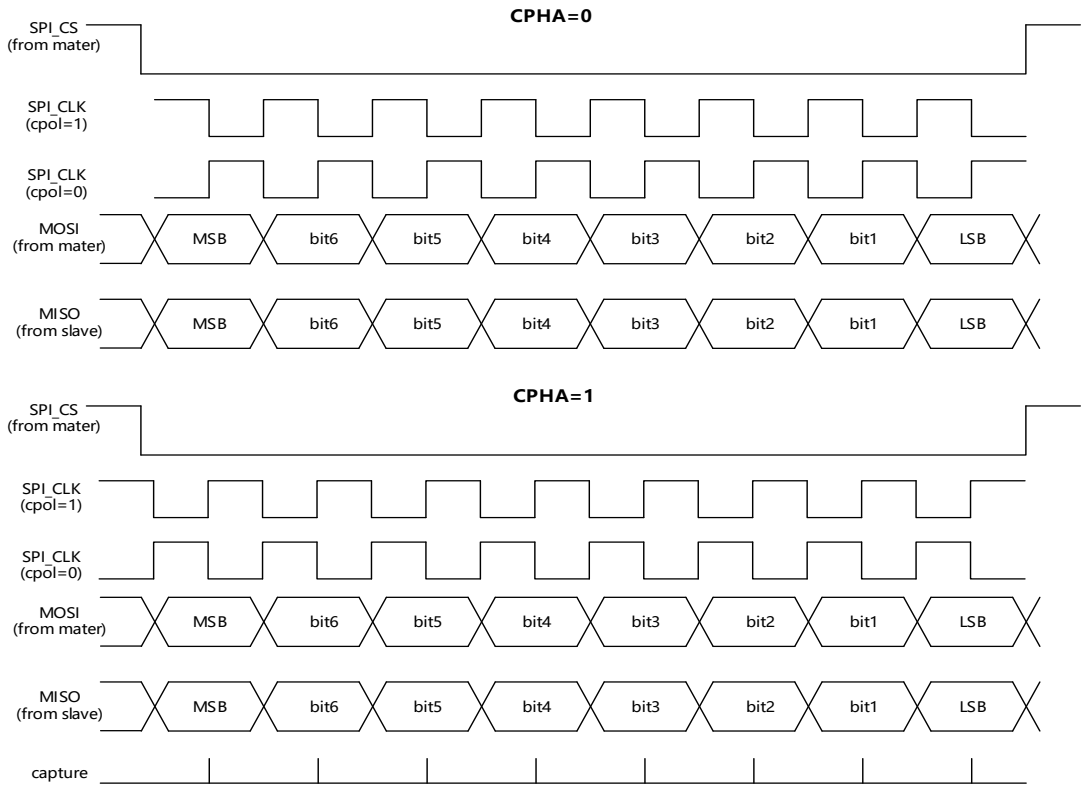


图 12-2 SPI 时序图

12.5.3. IIC 时序图

IIC 数据传输中，每帧数据为 8bit，传送数据时先传高位(MSB)，每帧数据传输完之后必须跟随一位应答位（低电平为应答，高电平为非应答）。

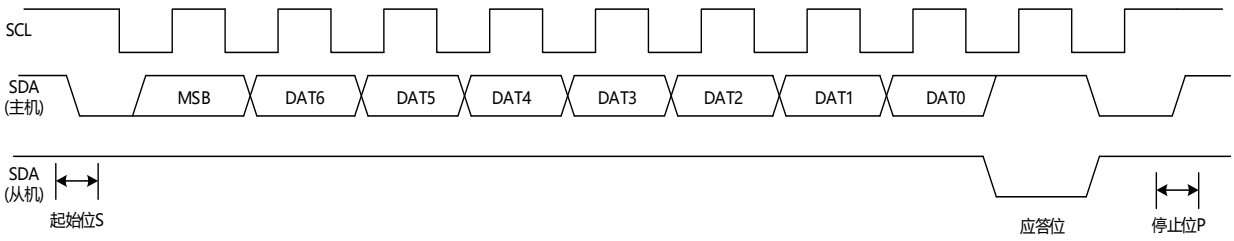


图 12-3 IIC 时序

12.5.4. IIC 数据传输

7bit 寻址，主机向从机写数据：

S	从机地址	W	A	数据	A	数据	A/$\bar{A}$	P

图 12-4 IIC 主机向从机写数据（7bit 寻址）

7bit 寻址，主机向从机读数据：

S	从机地址	R	A	数据	A	数据	$\bar{A}$	P

图 12-5 IIC 主机向从机读数据（7bit 寻址）

7bit 寻址，主机向从机先写数据后读数据：

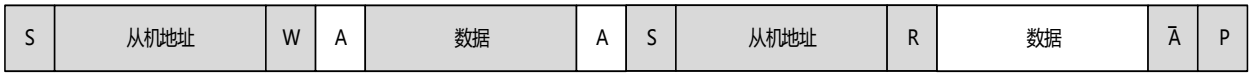


图 12-6 IIC 主机向从机先写数据后读数据（7bit 寻址）

10bit 寻址主机向从机写数据：

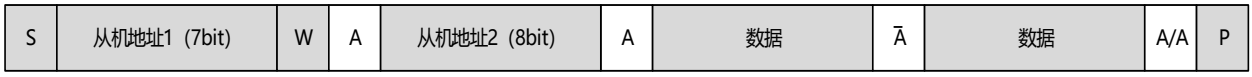


图 12-7 IIC 主机向从机写数据（10bit 寻址）

10bit 寻址主机向从机读数据：

图 12-8 IIC 主机向从机读数据（10bit 寻址）

12.6. 时钟与复位

12.6.1. 时钟介绍

该模块时钟来源于系统时钟，可通过配置系统寄存中 CLKCON2 来使能时钟。

12.6.2. 复位介绍

该模块的复位源有两个，分别是系统复位和软件复位，软件复位可通过配置系统寄存器触发。

12.7. 寄存器描述

12.7.1. 寄存器列表

Base address：0x4002_0000

Name	Offset	Reset	Description
SPIx_CON0/ IICx_CON0	0x02C0/0x02E0	32'h0	SPI_IIC 控制寄存器 0
SPIx_CON1 IICx_CON1	0x02C4/0x02E4	32'h0	SPI_IIC 控制寄存器 1
SPIx_CMD_DATA IICx_CMD_DATA	0x02C8/0x02E8	32'h0	SPI_IIC 数据寄存器
SPIx_BAUD IICx_BAUD	0x02CC/0x02EC	32'h0	SPI_IIC 波特率寄存器
SPIx_STA IICx_STA	0x02D0/0x0F0	32'h814	SPI_IIC 状态寄存器
SPIx_RONLY_CNT IICx_RONLY_CNT	0x02D4/0x02F4	32'h0	SPI_IIC 只读长度寄存器

12.7.2. 寄存器详细说明

12.7.2.1. SPI 控制寄存器（SPIx_CON0）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:14	FRAME_SIZE	2'h0	RW	SPI 数据帧长度

				0: 每帧传输 8bit 1: 每帧传输 16bit 2: 每帧传输 24bit 3: 每帧传输 32bit 注意: 当数据帧小于 32bit 时，数据寄存器的低位有效，比如 FRAME_SIZE=0 那么 CMD_DATA 的低 8 位为有效数据。
13	CS_POS_IE	1'b0	RW	SPI 从机模式 CS 上升沿中断使能 0: 不使能 1: 使能（检测到 CS 上升沿产生中断）
12	SPI_CS_EN	1'b0	RW	SPI_CS 引脚使能，此位仅在 SPI 从模式下有效 0: 不使能,SPI 总线没有 CS 引脚 1: 使能, SPI 总线有 CS 引脚
11	SPI_CS	1'b0	RW	SPI CS 引脚控制输出，此位仅在 SPI 主模式下有效 0: 输出低电平 1: 输出高电平
10:8	MASTER_CAP_DLY	3'b0	RW	SPI 主机模式延时采样数据 b000: 不延时 b001: 延时 1 个周期 b010: 延时 2 个周期 ... b111: 延时 7 个周期
7	SLAVE_SYNC_EN	1'b0	RW	SPI 从机模式同步输入数据使能 0: 不使能 1: 使能
6	MASTER_SYNC_EN	1'b0	RW	SPI 主机模式同步输入数据使能 0: 不使能 1: 使能
5	Reserved	-	-	-
4	LSBFE	1'b0	RW	SPI 先传输低位使能 0: 先传输高位 1: 先传输低位
3	Reserved	-	-	-
2	WIRE_MODE	1'b0	RW	SPI 通信模式: 0: 普通模式（CS,MOSI,MISO） 1: 3 线模式（CS,CLK,IO0）
1:0	SPI_MODE	2'b0	RW	SPI 接口时钟极性与采样相位 CPOL,CPHA: b00: 时钟初始为 0，上升沿采样，下降沿出数据 b01: 时钟初始为 0，下降沿采样，上升沿出数据 b10: 时钟初始为 1，下降沿采样，上升沿出数据 b11: 时钟初始为 1，上升沿采样，下降沿出数据

12.7.2.2. IIC 控制寄存器（IICx_CON0）

Width	Name	Reset	Property	Description
31:23	Reserved	-	-	-
22	RX_NACK_IE	1'b0	RW	接收到 NACK 中断使能 0: 不使能 1: 使能
21	AL_IE	1'b0	RW	主机仲裁丢失中断使能 0: 不使能 1: 使能
20	STOP_IE	1'b0	RW	检测到线上有 STOP 信号中断使能(主机和从机) 0: 不使能 1: 使能
19	ADDR_MATCH_IE	1'b0	RW	从机地址匹配中断使能 0: 不使能 1: 使能
18:14	FILTER_MAX	5'h0	RW	IIC 每 IIC_FILTER_CNT 个 IIC 模块时钟，采样一次 SCL 和 SDA，用于滤除一定宽度的线路毛刺。IIC_FILTER_CNT 配置越大虑掉的毛刺宽度越大。
13	BROADCAST_IE	1'b0	RW	IIC 从机接收到广播地址中断使能 0: 不使能 1: 使能
12	BROADCAST_EN	1'b0	RW	IIC 从机接收到广播地址使能 0: 忽略广播地址 1: 接收到广播地址时，回应 ACK，并产生 BROADCAST_PEND 标志
11:2	SLV_ADDR	10'h0	RW	IIC 从机地址
1	TX_NACK	1'b0	RW	IIC 在接收数据的时候，回应 NACK 还是 ACK 选择位 0: ACK 1: NACK
0	ADDR_WIDTH	1'b0	RW	IIC 从机地址宽度 0: 7bit 1: 10bit

12.7.2.3. 控制寄存器（SPIx_CON1/IICx_CON1）

Width	Name	Reset	Property	Description
31:13	Reserved	-	-	-
12	RXDMA_EN	1'b0	RW	接收 DMA 使能控制 0: 不使能 1: 使能
11	TXDMA_EN	1'b0	RW	发送 DMA 使能控制 0: 不使能 1: 使能
10	Reserved	-	-	-
9	Reserved	-	-	-

8	BUF_OV_IE	1'b0	RW	接收缓冲区溢出中断使能 0: 不使能 1: 使能
7	RBUF_NEMPTY_IE	1'b0	RW	接收缓冲区不空中断使能 0: 不使能 1: 使能
6	TBUF_NFULL_IE	1'b0	RW	发送缓冲区不满中断使能 0: 不使能 1: 使能
5	DONE_IE	1'b0	RW	发送/接收完一帧数据中断使能 0: 不使能 1: 使能
4	SPI_FULL_DUPLEX	1'b0	RW	全双工模式选择 0: 半双工 1: 全双工
3	TX_RX	1'b0	RW	接口发送/接收使能（仅 HALF_DUPLEX 为 1 时有效） 0: RX_EN 1: TX_EN
2	MASTER_SLAVE_SEL	1'b0	RW	接口主从模式设置 0: 主模式 1: 从模式
1	SPI_IIC_SEL	1'b0	RW	SPI/IIC 模式选择 0: SPI 模式 1: IIC 模式
0	SSP_EN	1'b0	RW	SPI 和 IIC 模块使能 0: 不使能 1: 使能

12.7.2.4. 数据寄存器（SPIx_CMD_DATA/IICx_CMD_DATA)

Width	Name	Reset	Property	Description
31:0	CMD_DATA	32'h0	RW	SPI 接口 写: 将要发送的数据写入该寄存器，触发 SPI 发送 读: 读该寄存器，获取收到的数据 IIC 接口 写: [31:10] 无效位 [9] STOP 使能位，在发送完 1byte 数据后，发送一个 STOP 位。(仅在主机模式有效) [8] START 使能位，使能后，在发送 1byte 数据前会先发 START 位。(仅在主机模式有效) [7:0] 将要发送的 8bit 数据 读: [31:8] 无效位 [7:0] 获取收到的数据

12.7.2.5. 波特率寄存器（SPIx_BAUD/IICx_BAUD）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	BAUD	16'h0	RW	主机模式 SPI 波特率=SYSCLK/(2*(BAUD+1)) IIC 波特率=SYSCLK/(4*(BAUD+1)) 从机模式 SPI: 无效 IIC: 从机准备好发送数据再延时（BAUD+1）个 SYSCLK 时钟周期再释放 SCL

12.7.2.6. 状态寄存器（SPIx_STA/IICx_STA）

Width	Name	Reset	Property	Description		
31	Reserved	-	-	-		
30:28	STATE	3'h0	RO	IIC 状态	主	从
				b000: IDLE	空闲	空闲
				b001:START	发送 start	已经接收到 start，等待 SCL 变 0
				b010: TX	发送 1byte 数据	发送 1byte 数据
				b011: RX	接收 1byte 数据	接收 1byte 数据
				b100:STOP	发送 stop	保留
				b101:ADDR0	保留	等待接收 1byte 地址
				b110:ADDR1	保留	等待接收 2byte 地址
				b111: 保留	保留	保留
27	SLAVE_ADDRED	1'b0	RO	IIC 从机模式，被寻址标志 0: 没有被寻址 1: 已经被寻址		
26:24	Reserved	-	-	-		
23	CLR_BUF_CNT	1'b0	WO	软件对该 bit 写 1 使 TBUF_CNT 和 RBUF_CNT 清零		
22	RBUF_OV	1'b0	RC	接收缓冲区溢出，有数据丢失标志 0: 缓冲区没有溢出 1: 缓冲区已经满，又有新数据要写入，丢失最新收到的数据 注: 缓冲区容量是 5byte，但不一定是装满 5byte 数据才是满，而是装不下一帧新的数据时就认为是满。		
21:19	TBUF_CNT	3'h0	RO	表示发送缓冲区里面有多少 byte 有效数据。帧宽度为 1~8/9~16/16~32bit 时，CPU 每写一次 CMD_DATA，计数器加 1/2/4，控制器每发一帧数据，计数器减 1/2/4。		

18:16	RBUF_CNT	3'h0	RO	表示接收缓冲区里面有多少 byte 有效数据。帧宽度为 1~8/9~16/16~32bit 时，控制器每收一帧数据，计数器加 1/2/4，CPU 每读一次 CMD_DATA，计数器减 1/2/4。
15	MASTER_RX_BUSY	1'b0	RO	主机模式接收数据时，指示是否读完需要读取的数据长度 0: 已读完 1: 未读完
14	IIC_RX_NACK	1'b0	RC	IIC 主从模式发送数据的时，在第 9bit 收到的应答信号 0: ACK 1: NACK
13	IIC_SLAVE_RW	1'b0	RO	IIC 从机在接收地址阶段，接收到的读写标志 0: 主机写从机 1: 主机读从机
12	IIC_BUS_BUSY	1'b0	RO	IIC 检测到线路 busy 0: 线路空闲 1: 线路繁忙 该位在检测到 start 位时置 1，检测到 stop 位时清零.
11	SPI_SLAVE_CS	1'b1	RO	SPI 从机模式接收到的 CS 电平状态
10	SSP_BUSY	1'b0	RO	SPI/IIC 作为主机或从机发送模式时的状态 0: 空闲 1: 主机正在发送/接收一帧数据 从机正在发送一帧数据 注: 从机接收模式下该 bit 无效
9	AL_PEND	1'b0	RC	IIC 主机模式检测到仲裁丢失 0: 没有仲裁丢失 1: 仲裁丢失 注: 只能软件清零，且必须清掉 IIC 才能正常工作
8	STOP_PEND	1'b0	RC	IIC 检测到线路上产生了 STOP 位 0: 没有检测到 STOP 位 1: 检测到 STOP 位 注: 只能软件清零
7	ADDR_MATCH_PEND	1'b0	RC	IIC 从机模式下，接收到主机发送过来的正确从机地址 0: 从机地址不匹配 1: 从机地址匹配 注: 只能软件清零
6	IIC_BROADCAST_PEND	1'b0	RC	IIC 从机模式下，检测到广播地址标志位 0: 没有检测到广播地址 1: 检测到广播地址 注: 只能软件清零
5	SPI_CS_POS_PEND	1'b0	RC	SPI 从机模式检测到 CS 引脚上升沿 0: 没有检测到上升沿 1: 检测到上升沿

				注: 只能软件清零
4	TBUF_EMPTY	1'b1	RO	发送缓冲区空标志 0: 非空 1: 空
3	TBUF_FULL	1'b0	RO	发送缓冲区满标志 0: 非满 1: 满
2	RBUF_EMPTY	1'b1	RO	接收缓冲区空标志 0: 非空 1: 空
1	RBUF_FULL	1'b0	RO	接收缓冲区满标志 0: 非满 1: 满
0	SSP_DONE	1'b0	RC	SPI 模式 0: 未完成数据收发 1: 已经完成一帧数据的发送/接收 IIC 模式 0: 未完成数据收发 1: 已经完成一帧数据发送/接收 注: IIC 主机模式的 数据帧（START(可选)+WRITE/READ(8bit+ACK)+STOP(可选)） IIC 从机模式的数据帧 (8bit+ACK)/ADDR0/ADDR1

12.7.2.7. 控制寄存器（SPIx_RONLY_CNT/IICx_RONLY_CNT)

Width	Name	Reset	Property	Description
31:12	Reserved	-	-	-
11:0	RONLY_CNT	12'h0	RW	半双工只读模式下 SPI 接收的数据长度。 当配置为半双工只读模式时，写此寄存器触发接收动作。 读此寄存器返回的是剩余未接收的数据量。

13. 通用异步收发器（UARTx）

13.1. 模块介绍

CIU32M011、CIU32M031 集成的通用异步收发器(UART)提供了一种灵活的方法与使用工业标准 NRZ 异步串行数据格式的外部设备之间进行全双工数据交换。

13.2. 功能特点

- 支持 8bit 数据和 9bit 数据模式
- 支持 18bit 波特率寄存器灵活配置
- 支持全双工异步通信、单工发送、单工接收、单线通信模式
- 支持硬件奇偶校验可选
- 具有 4 帧数据的接收缓存，1 帧数据的发送缓冲
- 硬件检测接收超时，超时长度可以配置，配置范围：1~16384 比特率时间
- 支持帧出错检测
- 可选 STOP 位为 1 位或 2 位

13.3. 功能说明

13.3.1. 波特率生成功能

波特率计算公式 = SYSCLK/(UART_BAUD+1) （UART_BAUD>=6)

UART 模块的工作时钟为系统时钟，因此，应当根据当前系统时钟频率以及需要的波特率计算 UART_BAUD 的配置值。一旦系统时钟频率改变，应当相应修改波特率配置。

13.3.2. UART 发送器

发送器可发送 8/9bit 数据，由寄存器 UART_CON 中的 BIT9_EN 位的配置值决定。软件向寄存器 UART_DATA 写入数据时，发送器将通过 IO 管脚 UART_TX 发送数据帧，UART_TX 的极性可以通过寄存器 UART_CON 中的 TX_INV 位来配置。

发送数据流程：发射器输出管脚（UART_TX）闲置状态时，默认为高电平。模块使能后，软件向寄存器 UART_DATA 写入要发送的数据启动发送。发送器有一帧的发送缓冲，寄存器 UART_STA 中 TX_BUF_EMPTY 为高电平时，软件可向寄存器 UART_DATA 再写入一帧数据，它将被存入发送缓冲区，TX_BUF_EMPTY 会变零，在当前帧发送完成时，会接着发送缓冲区中的数据。

数据发送完成且缓冲区为空时，寄存器 UART_STA 中 TC_PEND 位置 1，此时如果寄存器 UART_CON 中 TCIE 位使能，则产生中断。

发送器发送序列：起始位->数据位(LSB)->停止位

13.3.3. UART 接收器

接收器可接收 8/9bit 数据，由寄存器 UART_CON 中的 BIT9_EN 位的配置值决定。

数据字符由逻辑 0 的起始位、8/9bit 数据位，奇偶校验位和逻辑 1 的停止位组成。

接收器有 4 个 8/9bit 的数据缓冲。在缓冲区接收到 4 个数据之后，且又有一帧数据接收完成时，寄存器 UART_STA 中 RX_BUF_OV 会置 1，新的数据将不会存储在数据缓冲区中，即新数据丢失。

在接收一个数据帧的过程中可使能奇偶错误检测、帧错误检测和超时检测，通过配置寄存器 UART_CON 打开相应的错误检测使能位和相应的错误中断使能位。

帧错误检测机制是指在 stop 位检测到 IO 引脚 UART_RX 是低电平时,为帧错误。

超时检测机制是指在接收到 1byte 之后，检测是否超过了设置的时间，如果没有则继续接收数据，超时时间可以通过寄存器 UART_CON 中 TO_BIT_LEN 进行配置。

接收缓冲区有数据，即寄存器 UART_STA 中的 RX_BUF_NOT_EMPTY=1 时，软件可以通过读 UART_DATA 寄存器的方式来获取收到的数据。

13.4. IO 映射

详细参考 3.2. 管脚定义

13.5. 模块框图与接口时序

13.5.1. UART 模块架构图

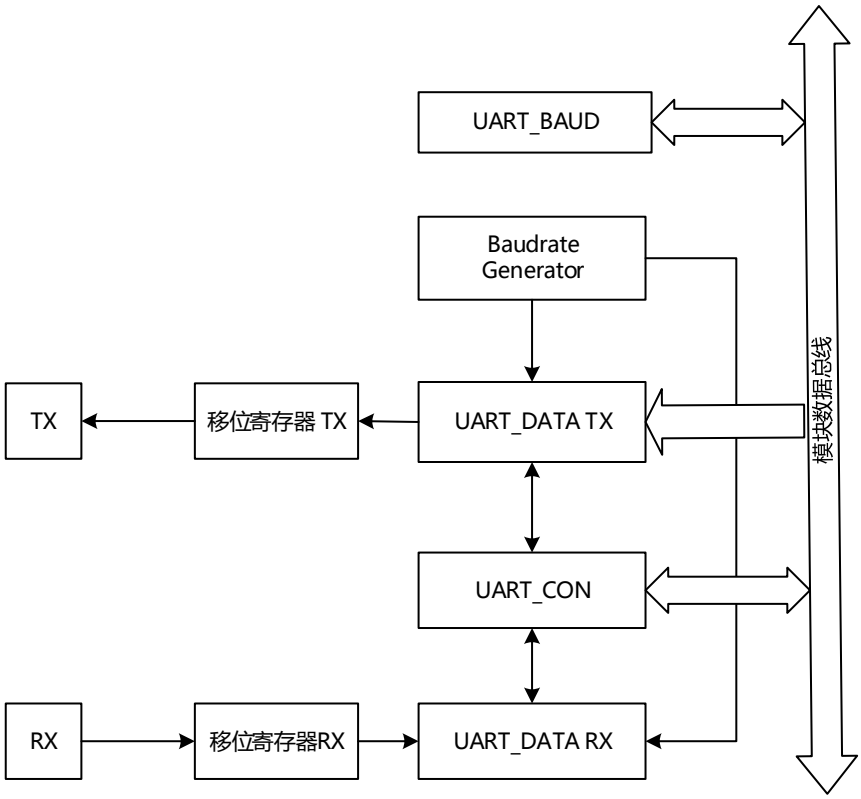


图 13-1 UART 模块架构图

13.5.2. UART 协议图

UART 数据发送/接收的数据格式如下所示，其中数据位可选择 8bit 或 9bit，需要注意的是，选择 9bit 数据时，就不可以使能奇偶校验位。相反，使能奇偶校验位时，不能选择 9bit 数据。

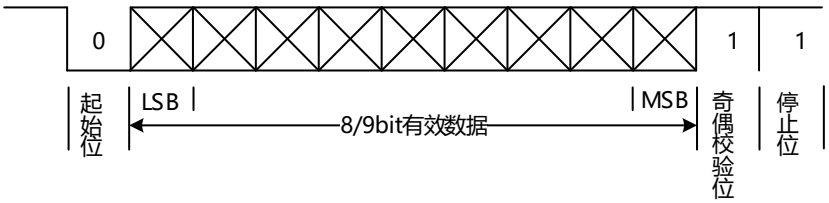


图 13-2 UART 时序图

13.6. 时钟与复位

13.6.1. 时钟介绍

该模块时钟来源于系统时钟，可通过配置系统寄存中 CLKCON2 来使能时钟。

13.6.2. 复位介绍

该模块的复位源有两个，分别是系统复位和软件复位，软件复位可通过配置系统寄存器触发。

13.7. 寄存器描述

13.7.1. 寄存器列表

Base address：0x4002_0000

Name	Offset	Reset	Description
UARTx_CON	0x0290/0x02A0	32'h360000	UART 模式控制寄存器
UARTx_BAUD	0x0294/0x02A4	18'h9c3	UART 波特率控制寄存器
UARTx_DATA	0x0298/0x02A8	xxxx	UART 数据寄存器
UARTx_STA	0x029C/0x02AC	17'h1	UART 状态寄存器

13.7.2. 寄存器详细说明

13.7.2.1. 控制寄存器（UARTx_CON）

Width	Name	Reset	Property	Description
31	RXDMA_EN	1'b0	RW	DMA 接收请求使能配置 0: 不使能 1: 使能
30	TXDMA_EN	1'b0	RW	DMA 发送请求使能配置 0: 不使能 1: 使能
29:16	TO_BIT_LEN	14'h36	RW	超时配置.单位是比特率时间 TIME OUT TIME =(TO_BIT_LEN+1)* BIT RATE TIME 默认值设置的 TIME OUT TIME = 5*(start_bit+data_bit(8\9)+parity_bit+stop_bit(1\2))=55
15	TC_IE	1'b0	RW	UART 发送完成中断使能 0: 不使能 1: 使能（UART_STA 寄存器中 TC=1 时产生中断
14	TMR_PWM_EN	1'b0	RW	UART 的 TX 带上 TMR PWM 载波 （UART0 对应 TIMER0 的 PWM，UART1 对应 TIMER1 的 PWM） 0: 不使能 1: 使能
13	TO_IE	1'b0	RW	TIME OUT 中断使能位 0: 不使能 1: 使能
12	TO_EN	1'b0	RW	TIME OUT 检测使能位 检测是否经历了（TO_BIT_LEN+1）*BIT_RATE_TIME 时间，没有接收到数据。 0: 不使能 1: 使能 注:每次 TO_EN 后，收到 1byte 数据后会产生 rx_one_byte 标志，该标志为 1 才会开始检测。该标志会在每次清 TO_PEND 的时候清 0，在 TO_EN 等于 0 时也会清 0。
11	FERR_IE	1'b0	RW	帧出错中断使能位 0: 不使能 1: 使能

10	TX_BUF_EMPTY_IE	1'b0	RW	发送缓冲空中断使能位 0: 不使能 1: 使能
9	RX_BUF_NEMPTY_IE	1'b0	RW	接收缓冲非空中断使能位 0: 不使能 1: 使能
8	TX_INV	1'b0	RW	发送输出信号取反选择位 0: 不使能 1: 使能
7	RX_INV	1'b0	RW	接收输入信号取反选择位 0: 不取反 1: 取反
6	ODD_EN	1'b0	RW	奇偶校验选择 0: 偶校验 1: 奇校验
5	PARITY_EN	1'b0	RW	奇偶校验使能 0: 不使能 1: 使能 注: 奇偶检验和 BIT9_EN 只能二选一
4	BIT9_EN	1'b0	RW	UART 传输 9bit 数据模式选择位 0: UART 传输 8bit 数据 1: UART 传输 9bit 数据 注: 奇偶检验和 BIT9_EN 只能二选一
3	STOP_BIT	1'b0	RW	UART 停止位选择 0: 一个停止位 1: 两个停止位
2	RX_CLOSE_EN	1'b0	RW	关闭 UART 接收功能 0: 不关闭 1: 关闭
1	WORK_MODE	1'b0	RW	UART 工作模式选择 0: 双线全双工模式 1: 单线半双工模式（软件启动发送时 IO 为输出状态，不发送时 IO 为输入状态）
0	UART_EN	1'b0	RW	UART 模块使能位 0: 不使能 1: 使能

13.7.2.2. 波特率寄存器（UARTx_BAUD）

Width	Name	Reset	Property	Description
31:18	Reserved	-	-	-
17:0	UART_BAUD	18'hd04	RW	UART 波特率控制寄存器 波特率=SYSCLK/(UART_BAUD+1) 注: 需配置 UART_BAUD >=6，否则输入信号会被内部滤波器滤掉。

13.7.2.3. 数据寄存器（UARTx_DATA）

Width	Name	Reset	Property	Description
31:9	Reserved	-	-	-
8:0	UART_DATA	xxx	RW	UART 数据寄存器 写: 写 8/9bit 数据到 UARTDATA 读: 获取接收到的低 8/9bit 的数据 注: 读之前，先读取 PERR[0]获取奇偶校验信息

13.7.2.4. 状态寄存器（UARTx_STA）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	TC_PEND	1'b0	RC	UART 发送完成标志（该标志由硬件置 1，软件清零） 硬件置 1: 当 UART 发送完一帧数据且缓冲区为空时，TC 置 1，此时如果 TCIE=1，则产生中断。 软件清 0: 软件通过对 TC 所在位写 1，或者向 UARTDATA 寄存器写数据均可对 TC 标志清零。 软件读: 0: UART 发送未完成 1: UART 发送完成
14:12	Reserved	-	-	-
11	TO_PEND	1'b0	RC	TIME OUT 标志位 按照 TO_BIT_LEN 默认值配置的时间，没有收到数据 0: 无效 1: 有效 （软件对该位写 1 清零）
10:7	PERR	4'b0	RO	UART 奇偶检验标志位 4bit 分别对应 BUF 里面的 4 个数据。硬件会自动将 PERR[0]对应的数据存放在 UARTDATA，在读 UARTDATA 之前，需要先读这个寄存器。每读走一个 UARTDATA 的数据，硬件会自动将这 4bit 右移 1 位。
6:4	RX_CNT	3'b0	RO	UART 接收缓存数据的数量 b000: 0 个数据 b001: 1 个数据 b010: 2 个数据 b011: 3 个数据 b100: 4 个数据 其它: 无效
3	FERR	1'b0	RC	帧错误，表示在 STOP bit 期间检测到 IO 引脚 UART_RX 出现了低电平。

				0: 没有帧错误 1: 出现了帧错误 （软件对该位写 1 清零）
2	RX_BUF_OV	1'b0	RC	UART 接收缓存满标志 接收 BUF 最多可以容纳 4 个 8/9bit 数据，如果软件还没有来得及读走，又收到一个数据，标志会起来。 0: 接收了（<=4byte）数据 1: 接收了（>4byte）的数据，只保存最开始的 4byte，其他丢弃 （软件对该位写 1 清零）
1	RX_BUF_NO EMPTY	1'b0	RO	接收缓冲区非空标志 0: 接收缓冲区空 1: 接收缓冲区非空
0	TX_BUF_EMPTY	1'b1	RO	UART 发送缓存空标志 UART 内部处理发送移位寄存器外，还有一帧（9bit）发送缓存，如果发送缓存为空标志会起来。 0: 发送缓存非空 1: 发送缓存为空

14. 高级定时器（TIMER1）

表 14-1 TIMERx 功能分类

类型	名称	位数	DMA	刹车	霍尔接口	正交编码	外部时钟	通道数	互补对数
高级	TIMER1	16	有	有	有	有	有	4	3
通用 1	TIMER2	16	有	无	有	有	有	4	无
	TIMER3	16	有	无	有	有	有	4	无
通用 2	TIMER4	16	有	有	无	无	无	2	1
	TIMER5	16	有	有	无	无	无	2	1
	TIMER6	16	有	有	无	无	无	2	1
基础	TIMER7	16	有	无	无	无	无	无	无

表 14-2 TIMERx 内部触发链接

从定时器	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIMER1	TIMER2	TIMER3	TIMER4	TIMER5
TIMER2	TIMER1	TIMER3	TIMER4	TIMER5
TIMER3	TIMER1	TIMER2	TIMER4	TIMER6
TIMER4	TIMER1	TIMER2	TIMER5	TIMER6
TIMER5	TIMER1	TIMER3	TIMER4	TIMER6
TIMER6	TIMER1	TIMER2	TIMER3	TIMER4

14.1. 模块介绍

高级控制定时器（TIMER1）由一个 16 位的自动装载计数器组成，它由一个可编程的预分频器驱动。它适合多种用途，包含测量输入信号的脉冲宽度(输入捕获)，或者产生输出波形(输出比较、PWM、嵌入死区时间的互补 PWM 等)。

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

高级控制定时器(TIMER1)和通用定时器(TIMERx)是完全独立的，它们不共享任何资源。

14.2. 功能特点

- 16 位向上、向下、向上/下自动装载计数器
- 16 位可编程（可以实时修改）预分频器，计数器时钟频率的分频系数为 1～65535 之间的任意数值
- 多达 4 个独立通道：
  - 输入捕获
  - 输出比较
  - PWM 生成（边缘或中间对齐模式）
  - 单脉冲模式输出
- 死区时间可编程的互补输出
- 使用外部信号控制定时器和定时器互联的同步电路
- 允许在指定数目的计数器周期之后更新定时器寄存器的重复计数器
- 刹车输入信号可以将定时器输出信号置于复位状态或者一个已知状态
- 如下事件发生时产生中断/DMA：
  - 更新：计数器向上溢出/向下溢出，计数器初始化(通过软件或者内部/外部触发)
  - 触发事件(计数器启动、停止、初始化或者由内部/外部触发计数)
  - 输入捕获
  - 输出比较
  - 刹车信号输入
- 支持针对定位的增量(正交)编码器和霍尔传感器电路
- 触发输入作为外部时钟或者按周期的电流管理

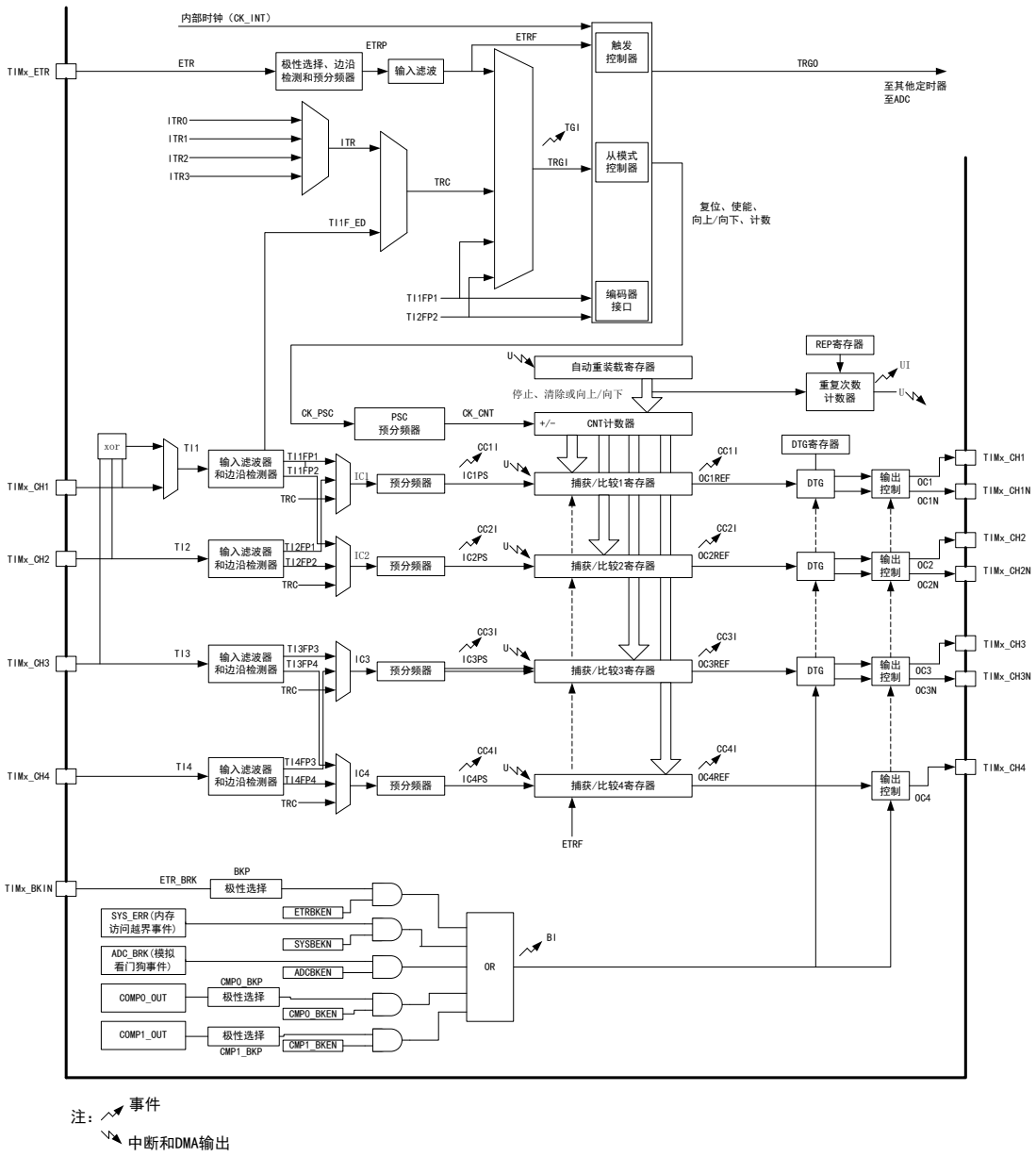


图 14-1

14.3. 功能说明

14.3.1. 时基单元

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

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

时基单元包含：

- 计数器寄存器(TIMERx_CNT)
- 预分频器寄存器 (TIMERx_PSC)
- 自动装载寄存器 (TIMERx_ARR)
- 重复次数寄存器 (TIMERx_RCR)

自动装载寄存器是预先装载的，写或读自动重装载寄存器将访问预装载寄存器。根据在 TIMERx_CR1 寄存器中的自动装载预装载使能位(ARPE)的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件(向下计数时的下溢条件)并当 TIMERx_CR1 寄

存器中的 UDIS 位等于 0 时，产生更新事件。更新事件也可以由软件产生。随后会详细描述每一种配置下更新事件的产生。

计数器由预分频器的时钟输出 CK_CNT 驱动，仅当设置了计数器 TIMERx_CR1 寄存器中的计数器使能位(CEN)时，CK_CNT 才有效。(更多有关使能计数器的细节，请参见控制器的从模式描述)。

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

预分频器描述

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

以下两图给出了在预分频器运行时，更改计数器参数的例子。

当预分频器的参数从 1 变到 2 时，计数器的时序图：

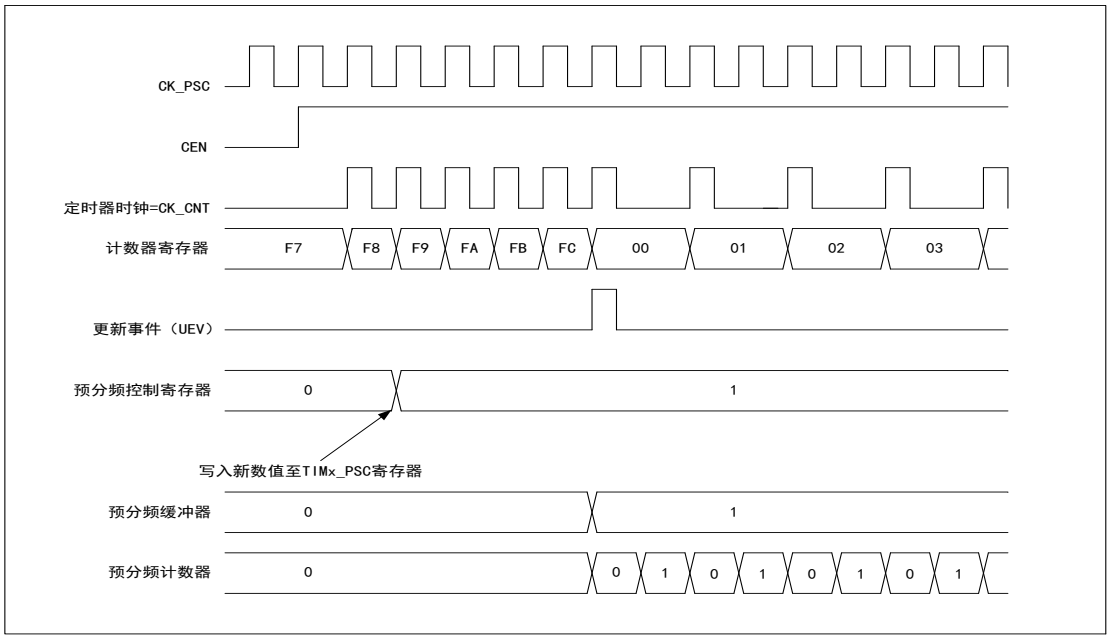


图 14-2

当预分频器的参数从 1 变到 4 时，计数器的时序图：

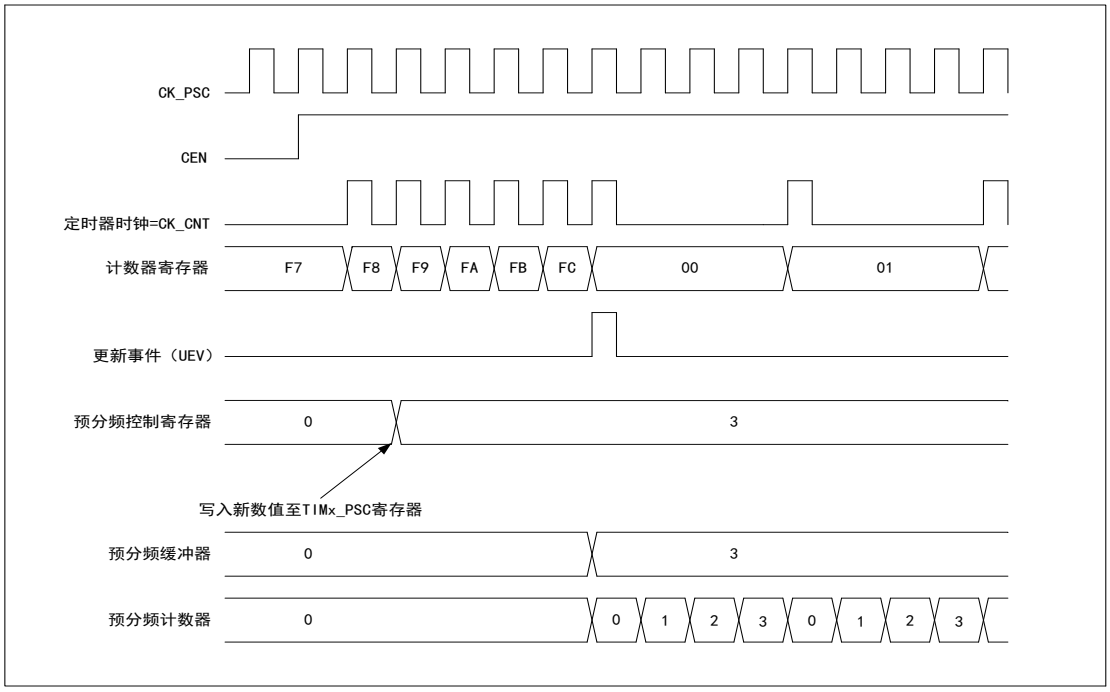


图 14-3

14.3.2. 计数器模式

向上计数模式

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

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

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

设置 TIMERx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清‘0’之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清‘0’，同时预分频器的计数也被清 0(但预分频器的数值不变)。此外，如果设置了 TIMERx_CR1 寄存器中的 URS 位(选择更新请求)，设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志(即不产生中断或 DMA 请求)。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

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

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

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

计数器时序图，预分频参数为 1

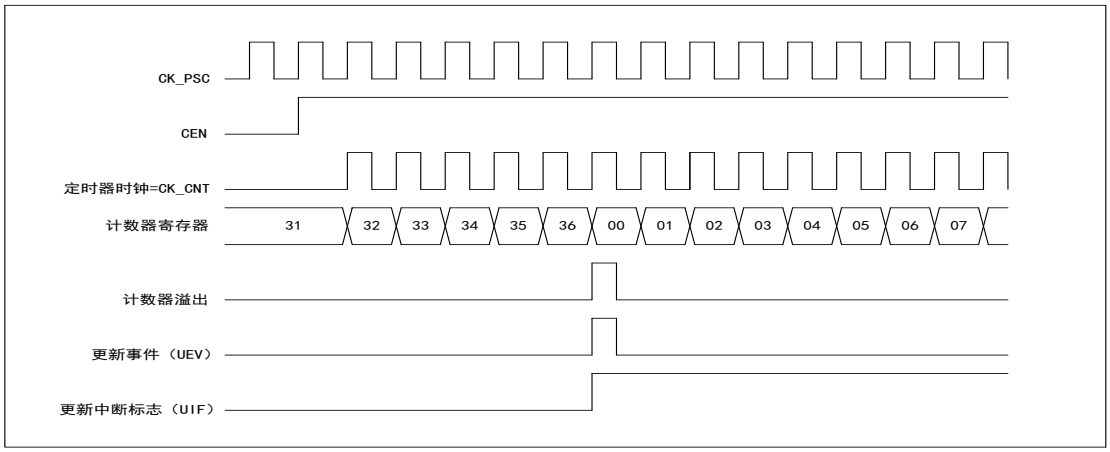


图 14-4

计数器时序图，预分频参数为 2

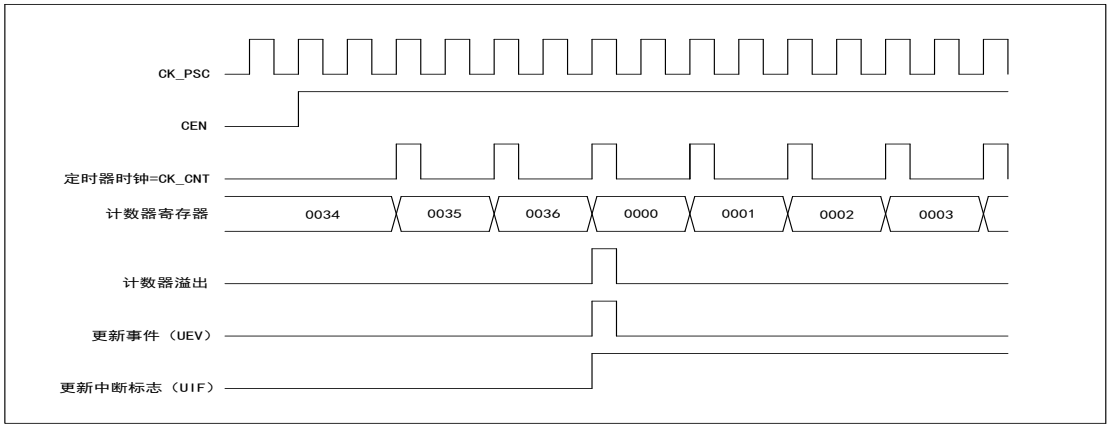


图 14-5

计数器时序图，预分频参数为 4

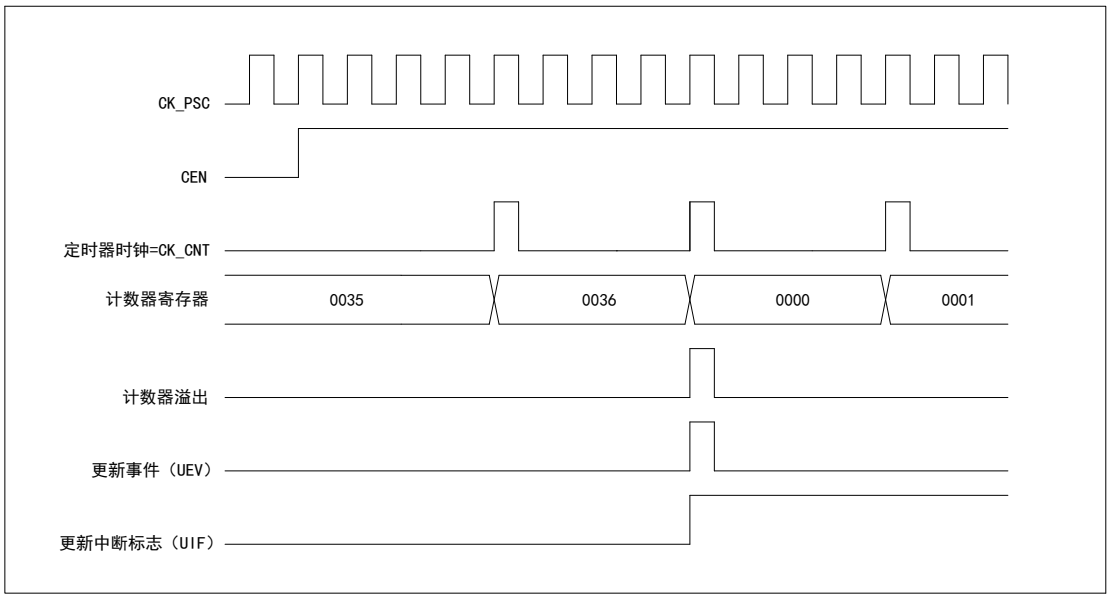


图 14-6

计数器时序图，预分频参数为 N

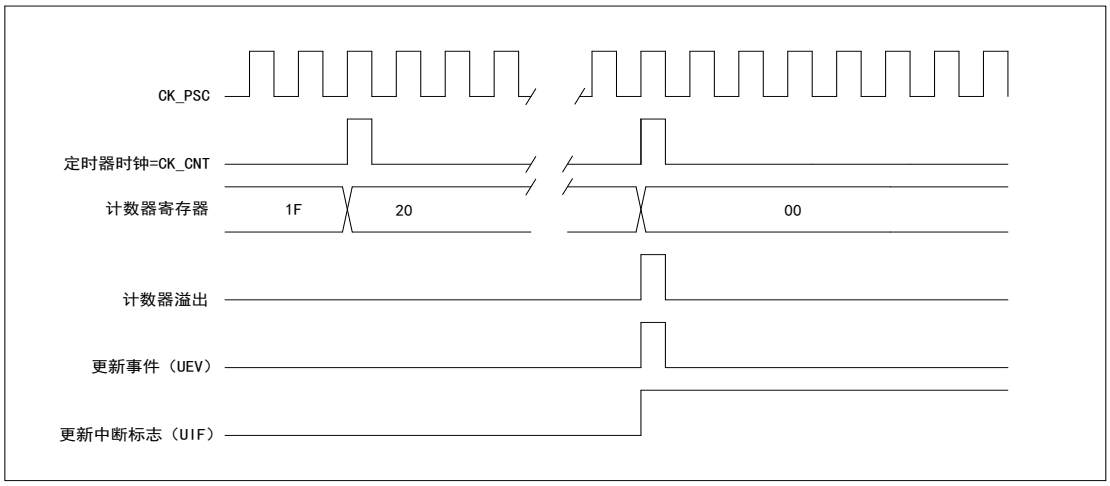


图 14-7

计数器时序图，当 ARPE=0 时的更新事件(TIMERx_ARR 没有预装入)

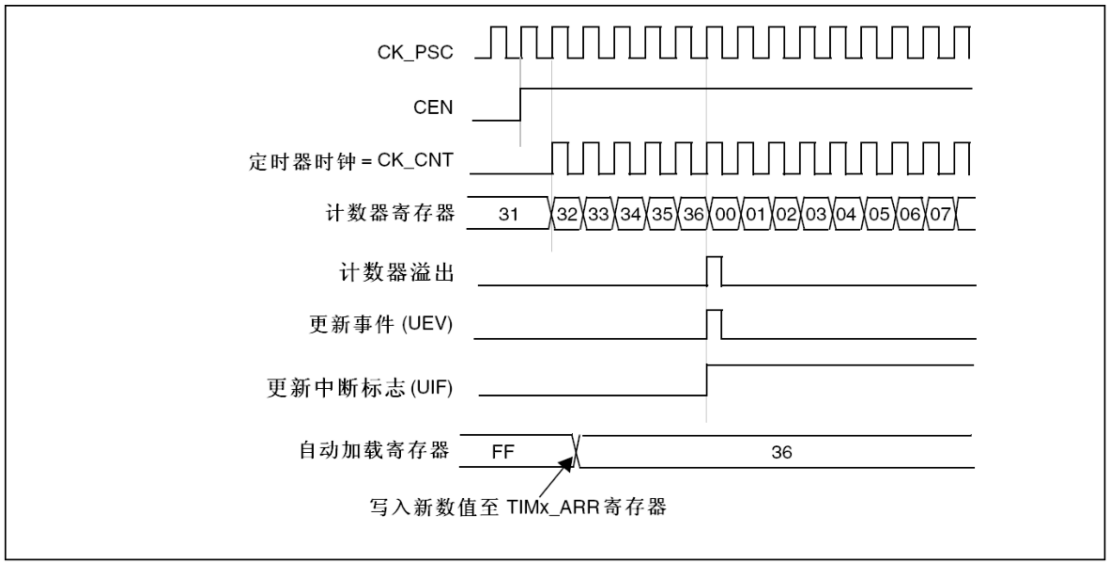


图 14-8

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

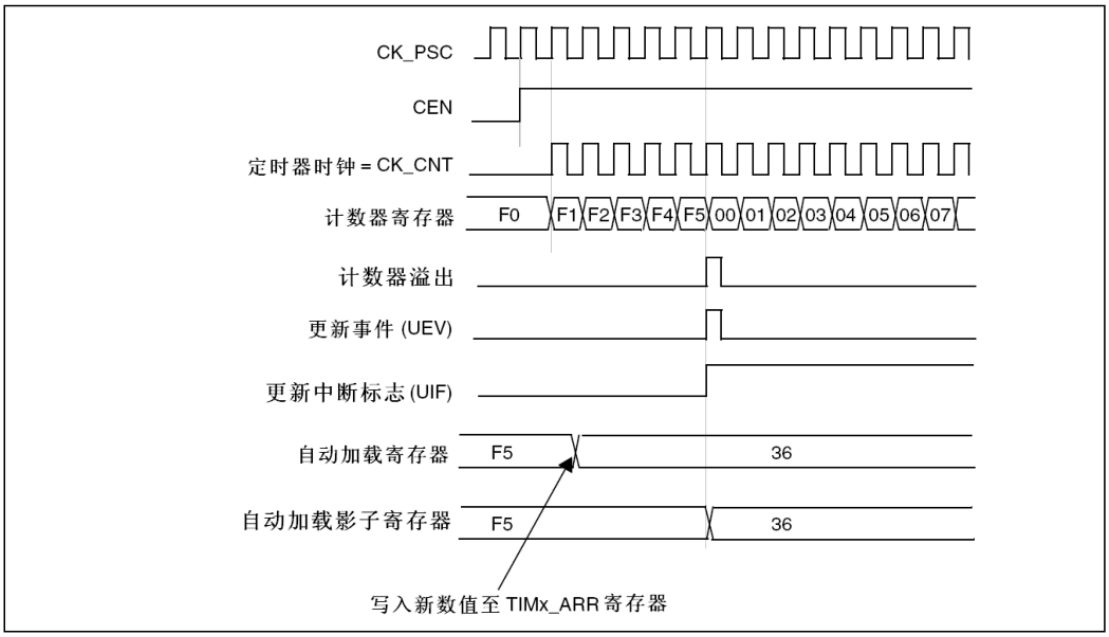


图 14-9

向下计数模式

在向下模式中，计数器从自动装入的值(TIMERx_ARR 计数器的值)开始向下计数到 0，然后从自动装入的值重新开始并且产生一个计数器向下溢出事件。

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

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

设置 TIMERx_CR1 寄存器的 UDIS 位可以禁止 UEV 事件。这样可以避免向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，并且预分频器的计数器重新从 0 开始(但预分频系数不变)。

此外，如果设置了 TIMERx_CR1 寄存器中的 URS 位(选择更新请求) ，设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志(因此不产生中断和 DMA 请求),这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

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

- 重复计数器被重新加载为 TIMERx_RCR 寄存器的内容。
- 预分频器的缓冲区被置入预装载寄存器的值(TIMERx_PSC 寄存器的内容)。
- 当前的自动加载寄存器被更新为预装载值(TIMERx_ARR 寄存器中的内容)。注：自动装载在计数器重载入之前被更新，因此下一个周期将是预期的值。

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

计数器时序图，预分频参数为 1

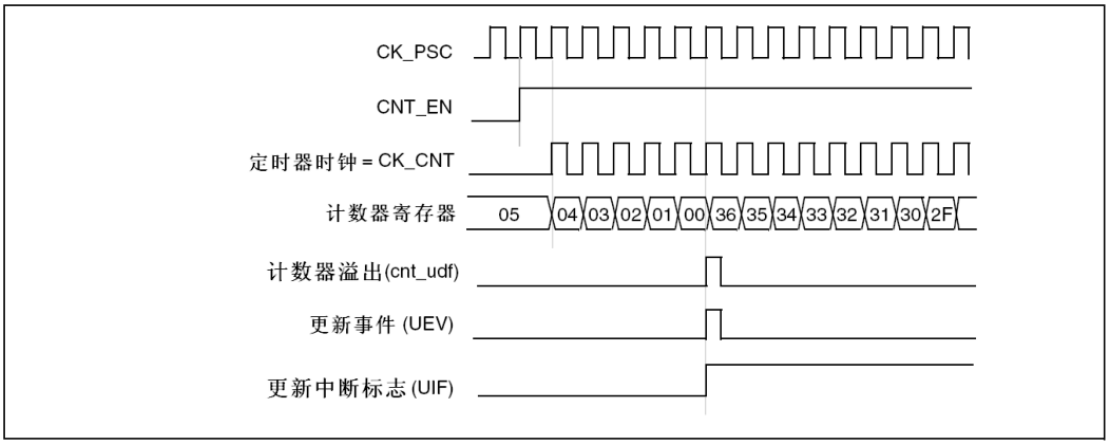


图 14-10

计数器时序图，预分频参数为 2

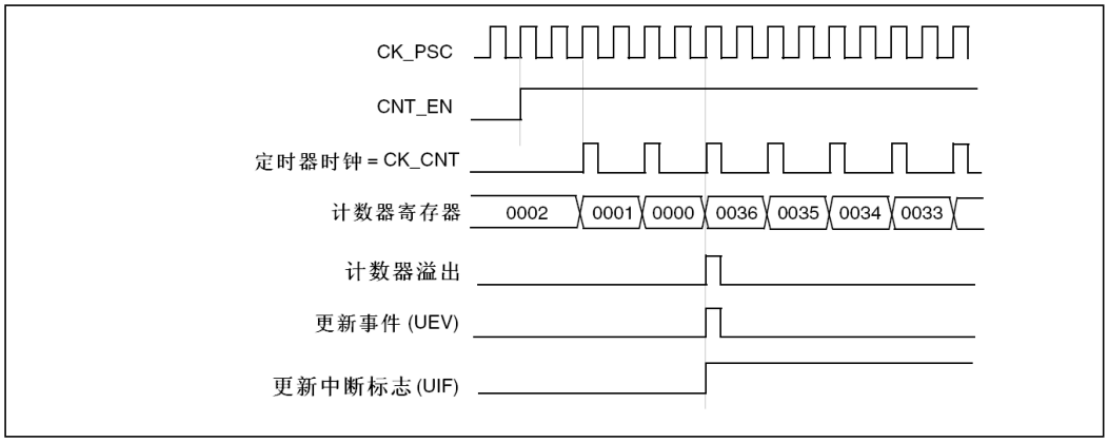


图 14-11

计数器时序图，预分频参数为 4

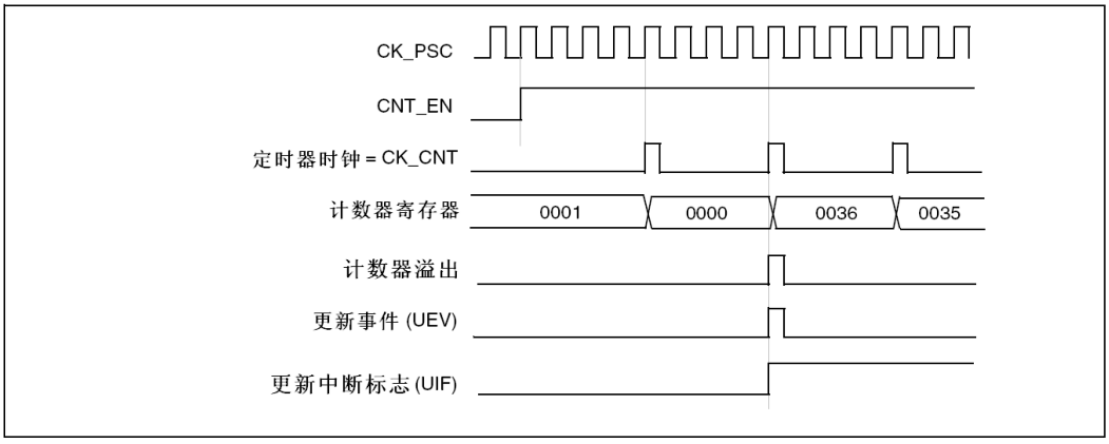


图 14-12

计数器时序图，预分频参数为 N

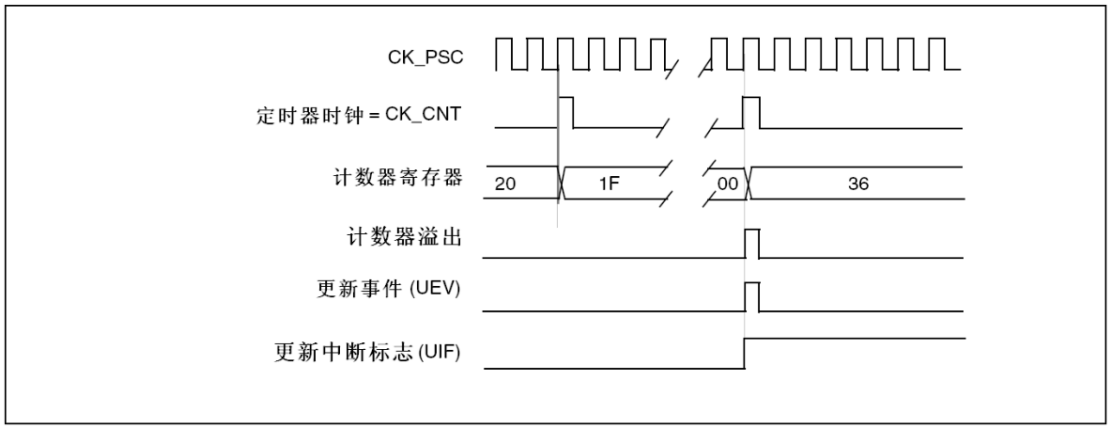


图 14-13

计数器时序图，当没有使用重复计数器时的更新事件

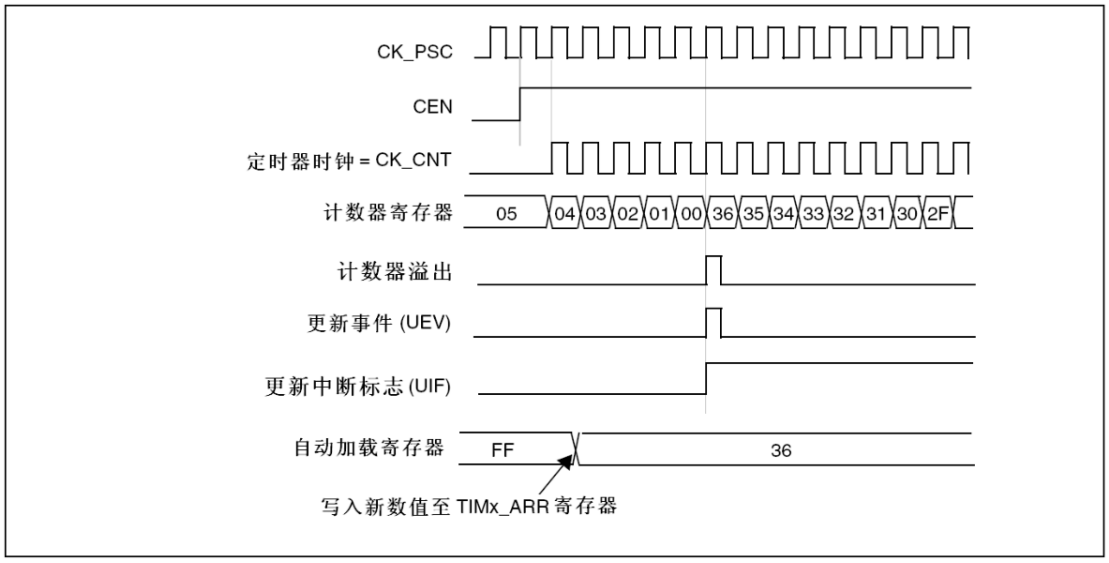


图 14-14

中央对齐模式（向上/向下计数）

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

在此模式下，不能写入 TIMERx_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。

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

设置 TIMERx_CR1 寄存器中的 UDIS 位可以禁止 UEV 事件。这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。

此外，如果设置了 TIMERx_CR1 寄存器中的 URS 位(选择更新请求) ，设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志(因此不产生中断和 DMA 请求),这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

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

- 重复计数器被重新加载为 TIMERx_RCR 寄存器的内容。
- 预分频器的缓冲区被置入预装载寄存器的值(TIMERx_PSC 寄存器的内容)。
- 当前的自动加载寄存器被更新为预装载值(TIMERx_ARR 寄存器中的内容)。注：如果因为计数器溢出而产生更新，自动重装载将在计数器重载入之前被更新，因此下一个周期将是预期的值(计数器被装载

为新的值)。

以下是一些计数器在不同时钟频率下的操作的例子：

计数器时序图，内部时钟分频因子为 1，TIMERx_ARR=0x6

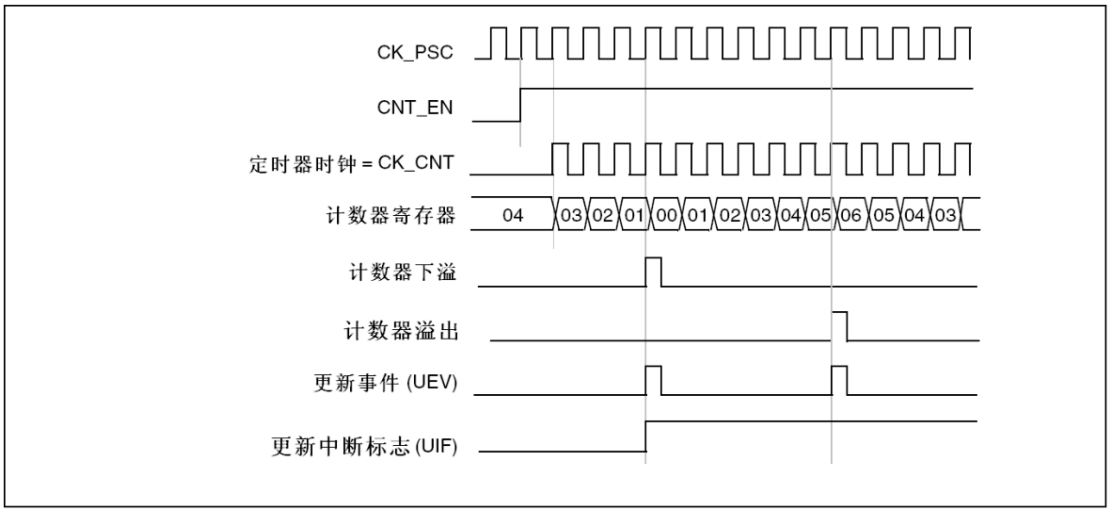


图 14-15

计数器时序图，预分频参数为 2

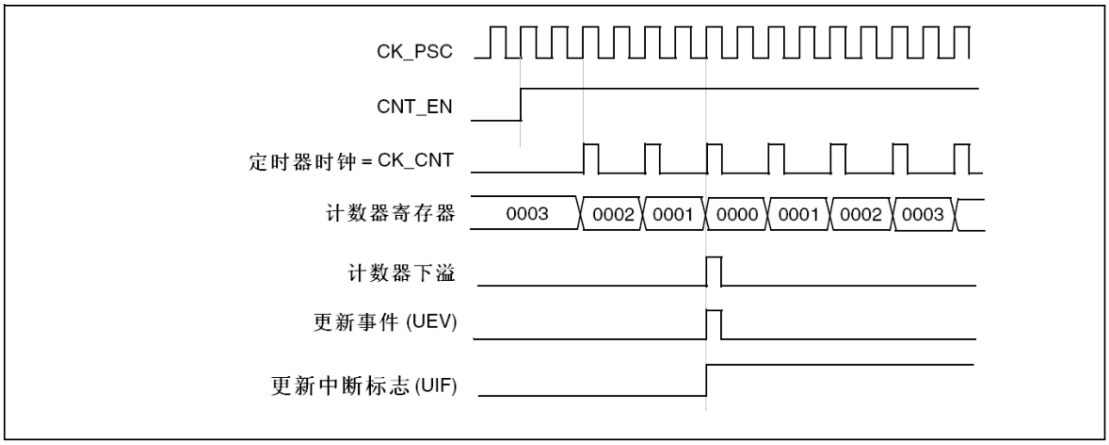


图 14-16

计数器时序图，预分频参数为 4，TIMERx_ARR=0x36

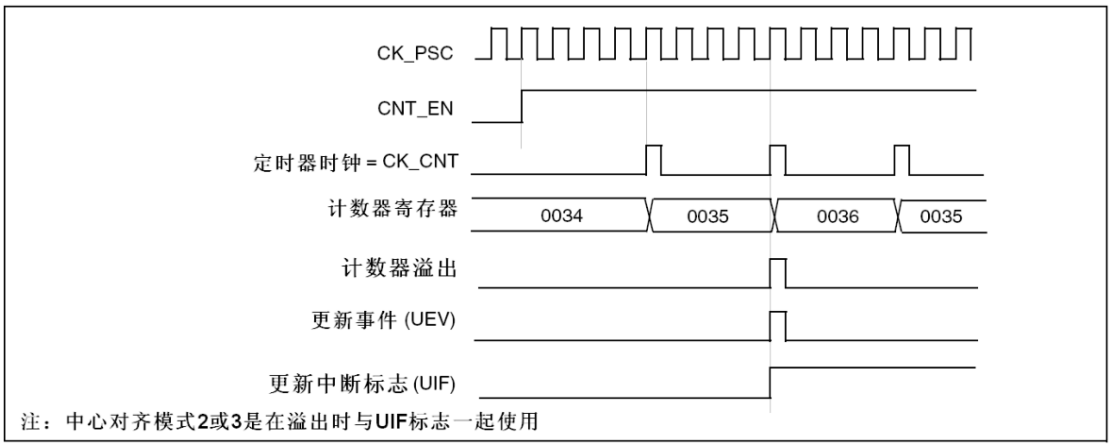


图 14-17

计数器时序图，预分频参数为 N

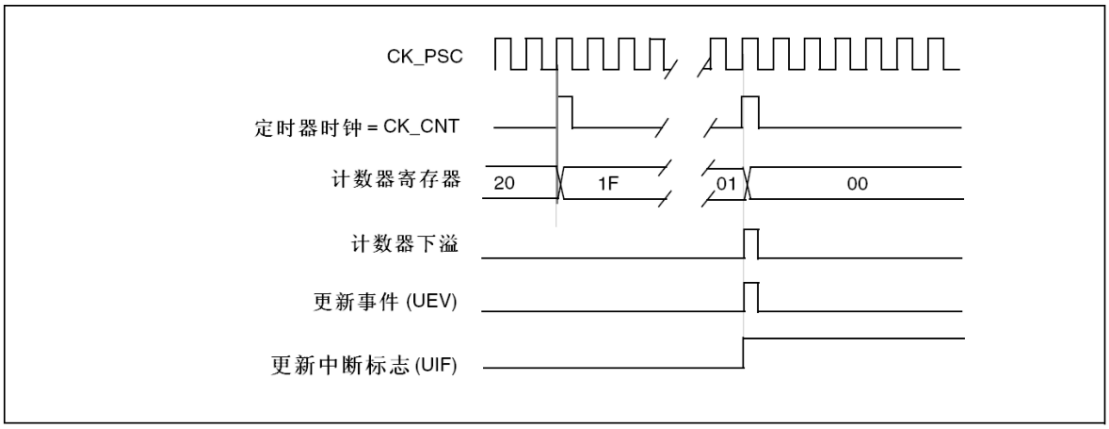


图 14-18

计数器时序图，ARPE=1 时的更新事件(计数器下溢)

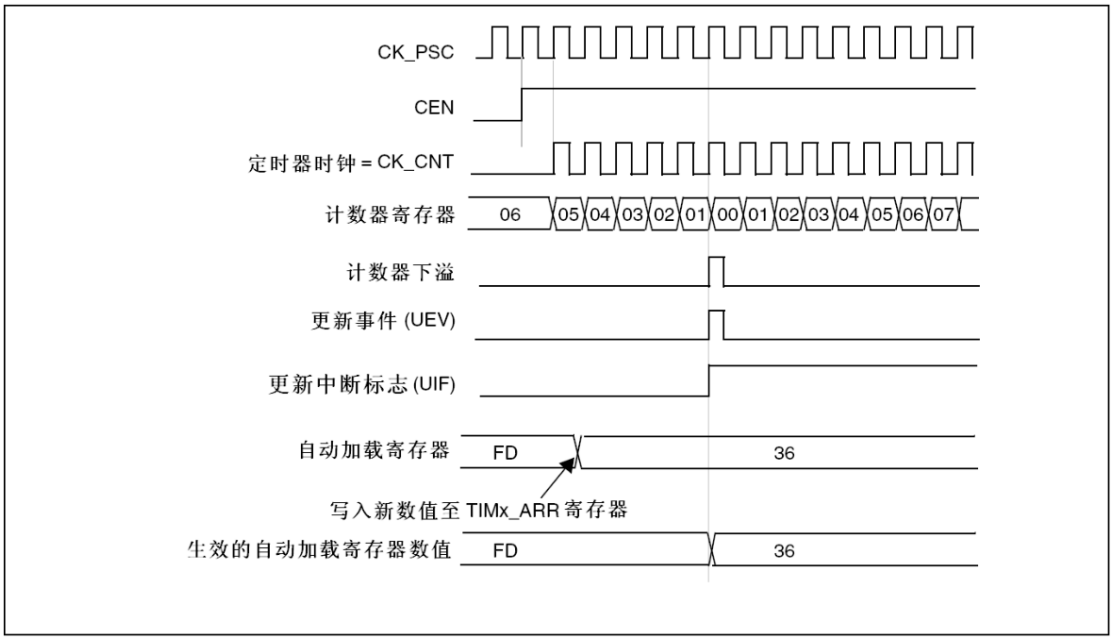


图 14-19

计数器时序图，ARPE=1 时的更新事件(计数器溢出)

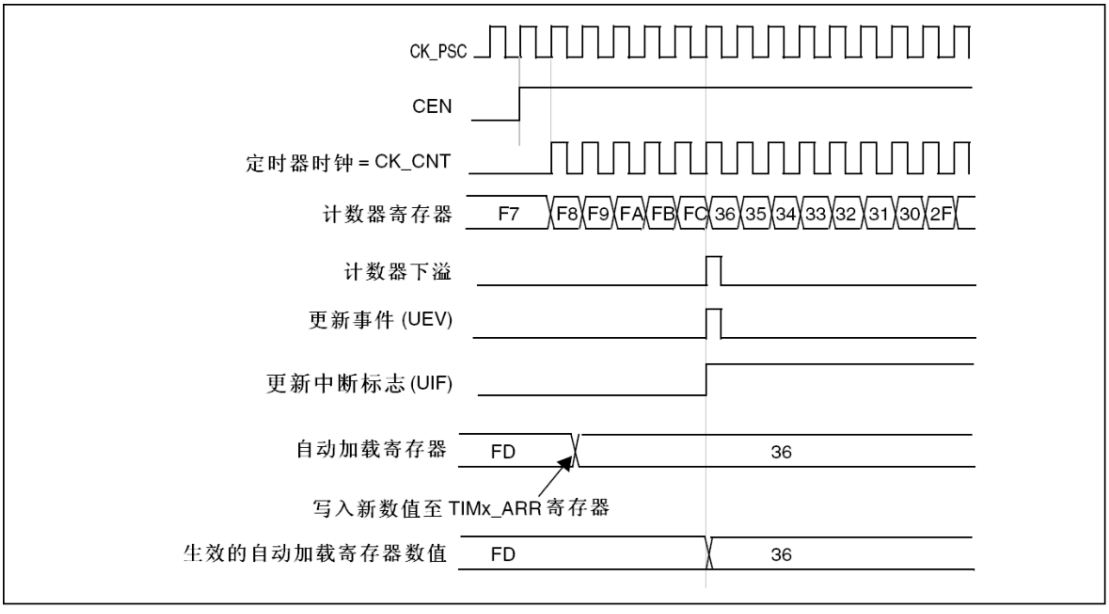


图 14-20

14.3.3. 重复计数器

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

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

重复计数器在下述任一条件成立时递减：

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

重复计数器是自动加载的，重复速率是由 TIMERx_RCR 寄存器的值定义(参考下图 15-21)。当更新事件由软件产生(通过设置 TIMERx_EGR 中的 UG 位)或者通过硬件的从模式控制器产生，则无论重复计数器的值是多少，立即发生更新事件，并且 TIMERx_RCR 寄存器中的内容被重载入到重复计数器。

不同模式下更新速率的例子，及 TIMERx_RCR 的寄存器设置

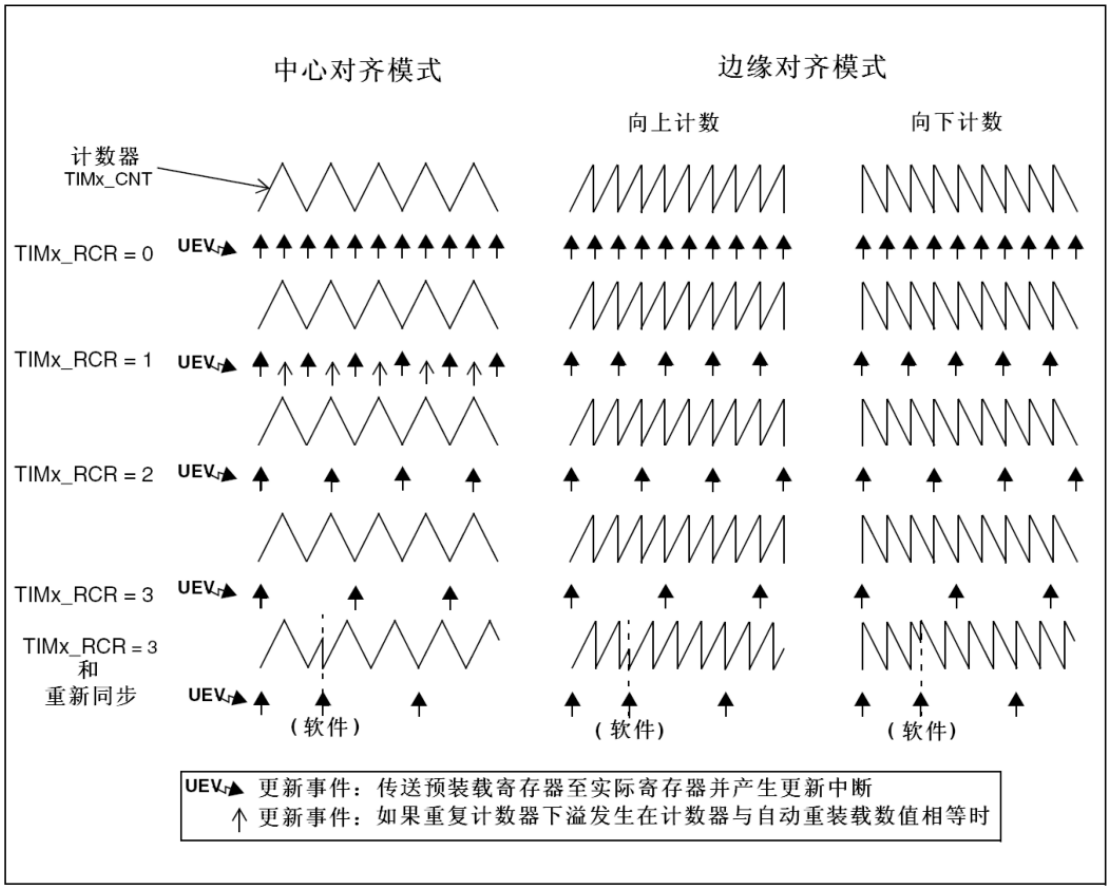


图 14-21

14.3.4. 时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟（CK_INT）
- 外部时钟模式 1：外部输入引脚
- 外部时钟模式 2：外部触发输入 ETR
- 内部触发输入(ITRx)：使用一个定时器作为另一个定时器的预分频器。如可以配置一个定时器 Timer1 而作为另一个定时器 Timer2 的预分频器。详见下一章。

内部时钟源 内部时钟源(CK_INT)

如果禁止了从模式控制器(SMS=000)，则 CEN、DIR(TIMERx_CR1 寄存器)和 UG 位(TIMERx_EGR 寄存器)是事实上的控制位，并且只能被软件修改(UG 位仍被自动清除)。只要 CEN 位被写成“1”,预分频器的时钟就由内部时钟 CK_INT 提供。

下图显示控制电路和向上计数器在一般模式下，不带预分频器时的操作。

一般模式下的控制电路，预分频参数为 1

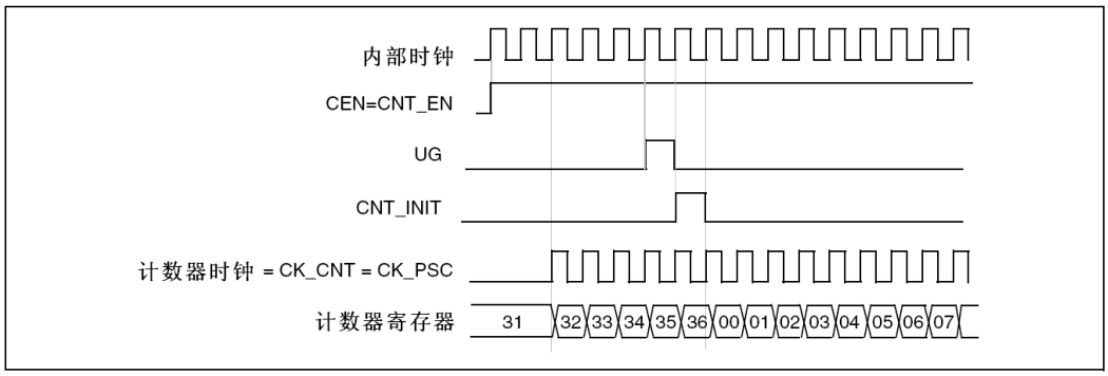


图 14-22

外部时钟源模式 1

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

TI2 外部时钟连接例子

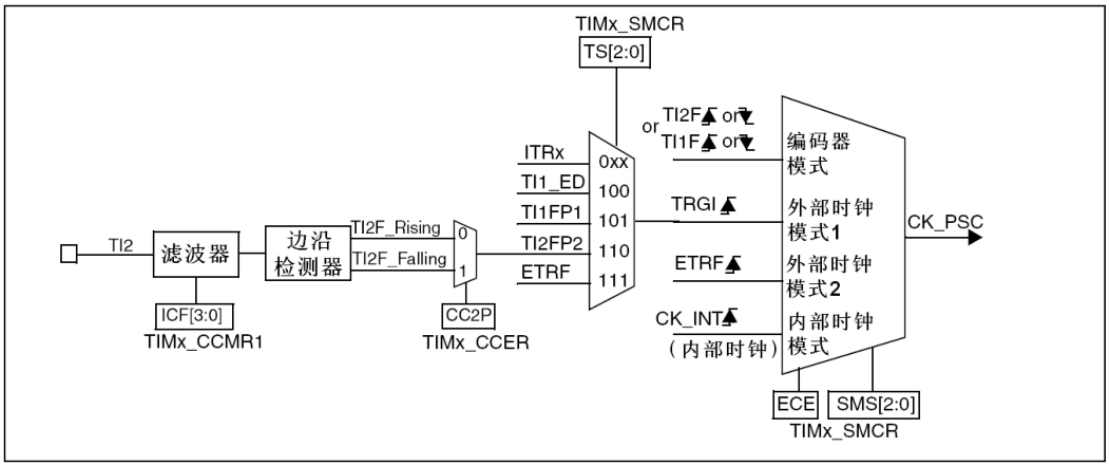


图 14-23

例如，要配置向上计数器在 T12 输入端的上升沿计数，使用下列步骤：

配置 TIMERx_CCMR1 寄存器 CC2S=01，配置通道 2 检测 TI2 输入的上升沿

配置 TIMERx_CCMR1 寄存器的 IC2F[3:0]，选择输入滤波器带宽(如果不需要滤波器，保持 IC2F=0000)

配置 TIMERx_CCER 寄存器的 CC2P=0，选定上升沿极性

配置 TIMERx_SMCR 寄存器的 SMS=111，选择定时器外部时钟模式 1

配置 TIMERx_SMCR 寄存器中的 TS=110，选定 TI2 作为触发输入源

设置 TIMERx_CR1 寄存器的 CEN=1，启动计数器

注： 捕获预分频器不用作触发，所以不需要对它进行配置

当上升沿出现在 TI2，计数器计数一次，且 TIF 标志被设置。

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

外部时钟模式 1 下的控制电路

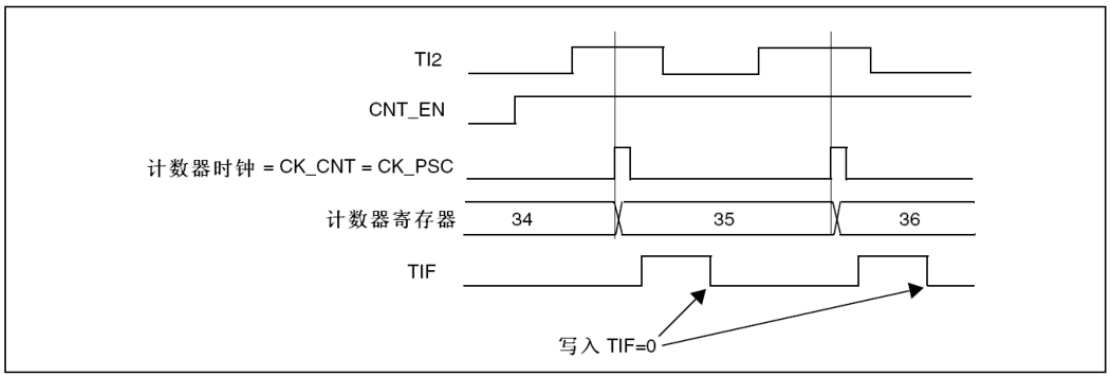


图 14-24

外部时钟源模式 2

选定此模式的方法为：令 TIMERx_SMCR 寄存器中的 ECE=1,计数器能够在外部触发 ETR 的每一个上升沿或下降沿计数。

下图是外部触发输入的框图

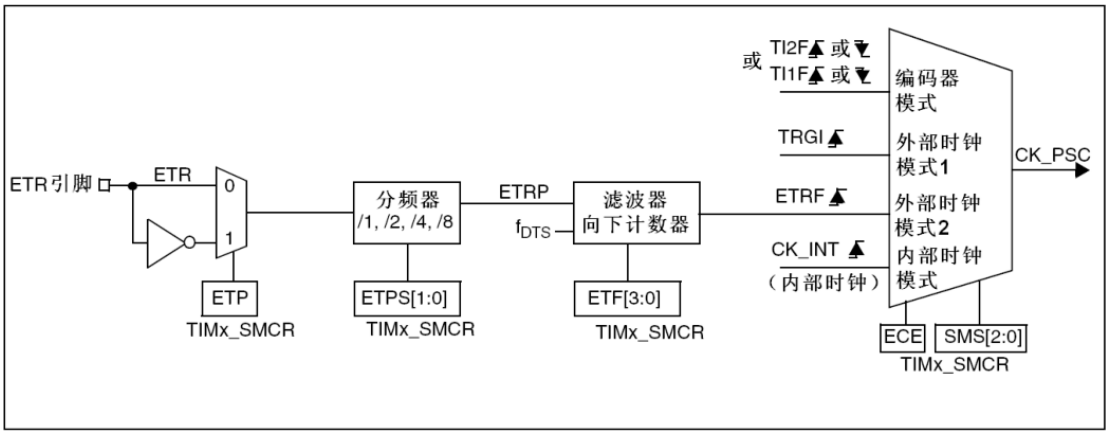


图 14-25 外部触发输入框图

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

本例中不需要滤波器，置 TIMERx_SMCR 寄存器中的 ETF[3:0]=0000

设置预分频器，置 TIMERx_SMCR 寄存器中的 ETPS[1:0]=01

选择 ETR 的上升沿检测，置 TIMERx_SMCR 寄存器中的 ETP=0

开启外部时钟模式 2，写 TIMERx_SMCR 寄存器中的 ECE=1

启动计数器，写 TIMERx_CR1 寄存器中的 CEN=1

计数器在每 2 个 ETR 上升沿计数一次。

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

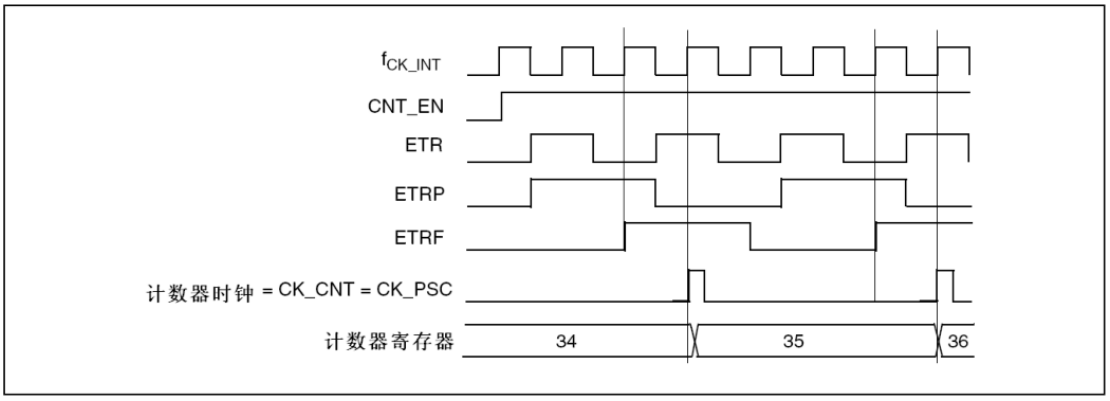


图 14-26 外部时钟模式 2 下的控制电路

14.3.5. 捕获/比较通道

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

以下三图是一个捕获/比较通道概览。

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

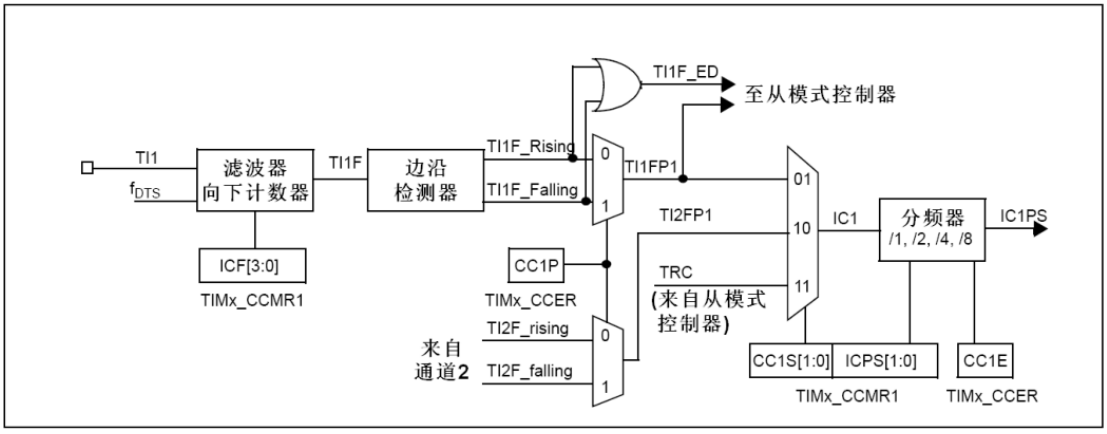


图 14-27 捕获/比较通道(如：通道 1 输入部分)

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

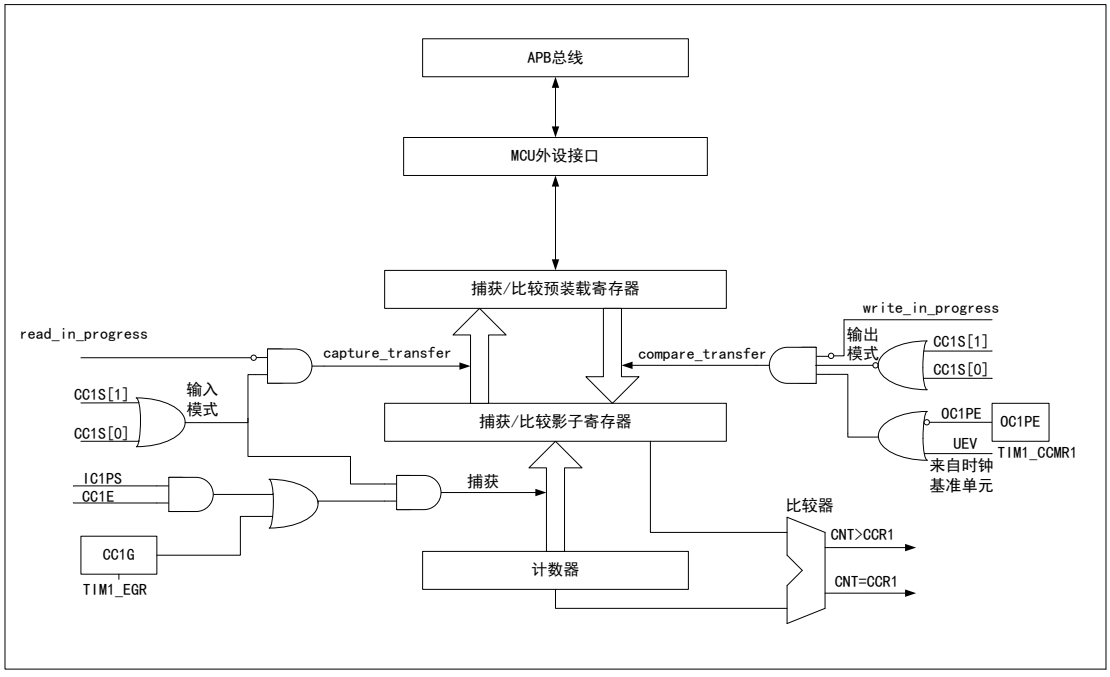


图 14-28 捕获/比较通道 1 的主电路

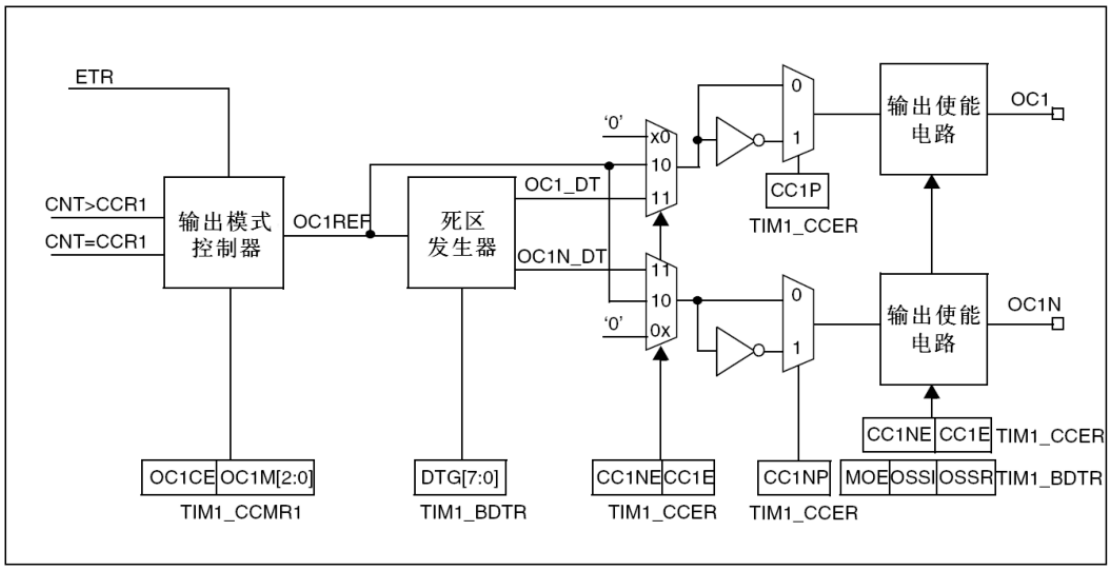


图 14-29 捕获/比较通道的输出部分(通道 1 至 3)

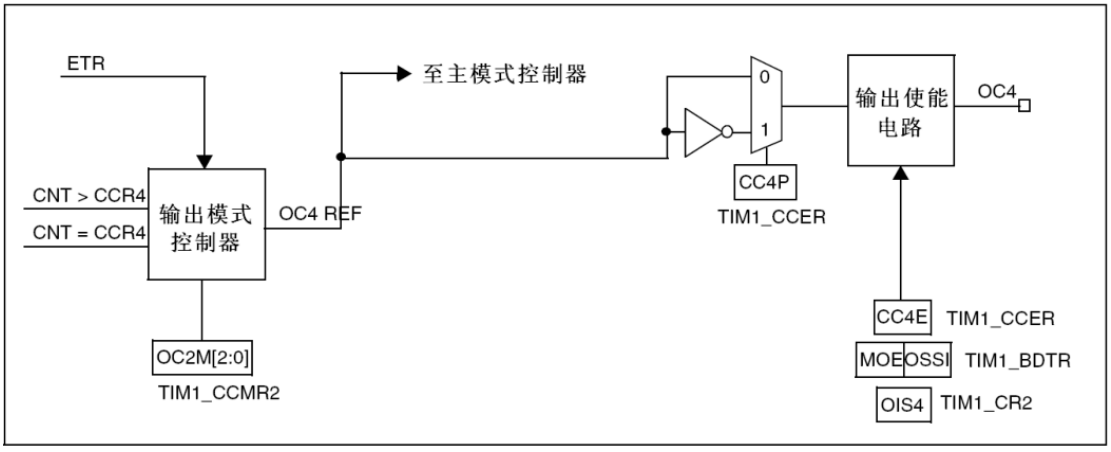


图 14-30 捕获/比较通道的输出部分(通道 4)

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

在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。

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

14.3.6. 输入捕获模式

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

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIMERx_CCR1 寄存器中，步骤如下：

- 选择有效输入端：TIMERx_CCR1 必须连接到 TI1 输入，所以写入 TIMERx_CCMR1 寄存器中的 CC1S=01，只要 CC1S 不为“00”，通道被配置为输入，并且 TIMERx_CCR1 寄存器变为只读。
- 根据输入信号的特点，配置输入滤波器为所需的带宽(即输入为 TIx 时，输入滤波器控制位是 TIMERx_CCMRx 寄存器中的 ICxF 位)。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期；因此我们可以(以 f DTS 频率)连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIMERx_CCMR1 寄存器中写入 IC1F=0011。
- 选择 TI1 通道的有效转换边沿，在 TIMERx_CCER 寄存器中写入 CC1P=0(上升沿)。
- 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止(写 TIMERx_CCMR1 寄存器的 IC1PS=00)。
- 设置 TIMERx_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。

- 如果需要，通过设置 TIMERx_DIER 寄存器中的 CC1IE 位允许相关中断请求，通过设置 TIMERx_DIER 寄存器中的 CC1DE 位允许 DMA 请求。
- 当发生一个输入捕获时：
- 产生有效的电平转换时，计数器的值被传送到 TIMERx_CCR1 寄存器。
- CC1IF 标志被设置(中断标志)。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置 1。
- 如设置了 CC1IE 位，则会产生一个中断。
- 如设置了 CC1DE 位，则还会产生一个 DMA 请求。
- 为了处理捕获溢出，建议在读出捕获溢出标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

注： 设置 TIMERx_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断和/或 DMA 请求。

14.3.7. PWM 输入模式

该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

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

例如,你需要测量输入到 TI1 上的 PWM 信号的长度(TIMERx_CCR1 寄存器)和占空比(TIMERx_CCR2 寄存器)，具体步骤如下(取决于 CK_INT 的频率和预分频器的值)

- 选择 TIMERx_CCR1 的有效输入：置 TIMERx_CCMR1 寄存器的 CC1S=01(选中 TI1)。
- 选择 TI1FP1 的有效极性(用来捕获数据到 TIMERx_CCR1 中和清除计数器)：置 CC1P=0(上升沿有效)。
- 选择 TIMERx_CCR2 的有效输入：置 TIMERx_CCMR1 寄存器的 CC2S=10(选中 TI1)。
- 选择 TI1FP2 的有效极性(捕获数据到 TIMERx_CCR2)：置 CC2P=1(下降沿有效)。
- 选择有效的触发输入信号：置 TIMERx_SMCR 寄存器中的 TS=101(选择 TI1FP1)。
- 配置从模式控制器为复位模式：置 TIMERx_SMCR 中的 SMS=100。
- 使能捕获：置 TIMERx_CCER 寄存器中 CC1E=1 且 CC2E=1。

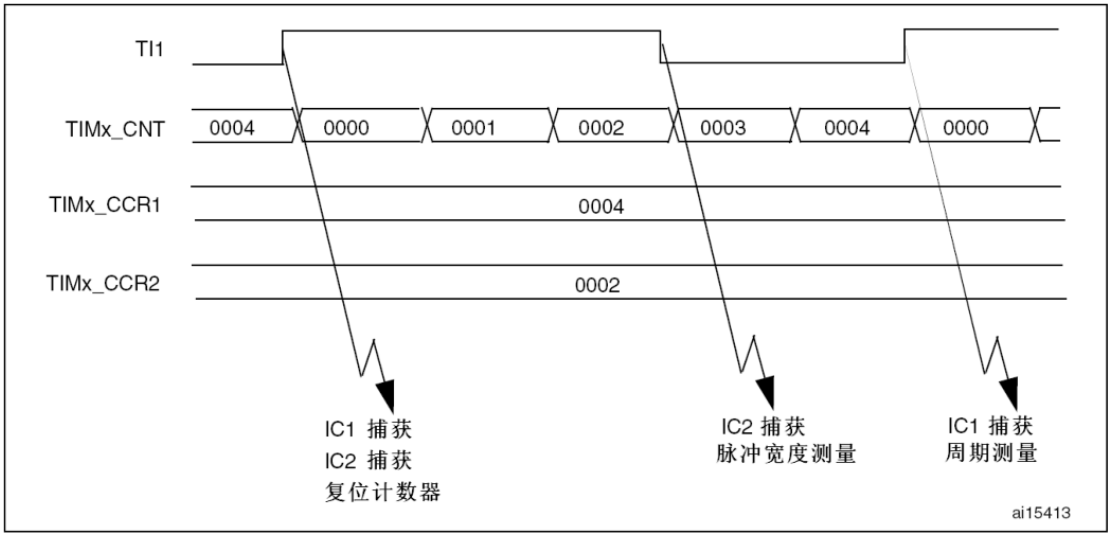


图 14-31 PWM 输入模式时序

因为只有 TI1FP1 和 TI2FP2 连到了从模式控制器，所以 PWM 输入模式只能使用 TIMERx_CH1/TIMERx_CH2 信号。

14.3.8. 强制输出模式

在输出模式(TIMERx_CCMRx 寄存器中 CCxS=00)下，输出比较信号(OCxREF 和相应的 OCx/OCxN)能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。

置 TIMERx_CCMRx 寄存器中相应的 OCxM=101，即可强置输出比较信号(OCxREF/OCx)为有效状态。这样 OCxREF 被强置为高电平(OCxREF 始终为高电平有效)，同时 OCx 得到 CCxP 极性相反的信号。

例如：CCxP=0(OCx 高电平有效)，则 OCx 被强置为高电平。

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

该模式下，在 TIMERx_CCRx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断和 DMA 请求。这将会在下面的输出比较模式一节中介绍。

14.3.9. 输出比较模式

此项功能是用来控制一个输出波形，或者指示一段给定的的时间已经到时。

当计数器与捕获/比较寄存器的内容相同时，输出比较功能做如下操作：

- 将输出比较模式(TIMERx_CCMRx 寄存器中的 OCxM 位)和输出极性(TIMERx_CCER 寄存器中的 CCxP 位)定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平(OCxM=000)、被设置成有效电平(OCxM=001)、被设置成无效电平(OCxM=010)或进行翻转(OCxM=011)。
- 设置中断状态寄存器中的标志位(TIMERx_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽(TIMERx_DIER 寄存器中的 CCxIE 位)，则产生一个中断。
- 若设置了相应的使能位(TIMERx_DIER 寄存器中的 CCxDE 位，TIMERx_CR2 寄存器中的 CCDS 位选择 DMA 请求功能)，则产生一个 DMA 请求。

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

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

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

输出比较模式的配置步骤：

1. 选择计数器时钟(内部，外部，预分频器)。
2. 将相应的数据写入 TIMERx_ARR 和 TIMERx_CCRx 寄存器中。
3. 如果要产生一个中断请求，设置 CCxIE 位。
4. 选择输出模式，例如：
   - 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚，设置 OCxM=011
   - 置 OCxPE = 0 禁用预装载寄存器
   - 置 CCxP = 0 选择极性为高电平有效
   - 置 CCxE = 1 使能输出
5. 设置 TIMERx_CR1 寄存器的 CEN 位启动计数器

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

输出比较模式，翻转 OC1

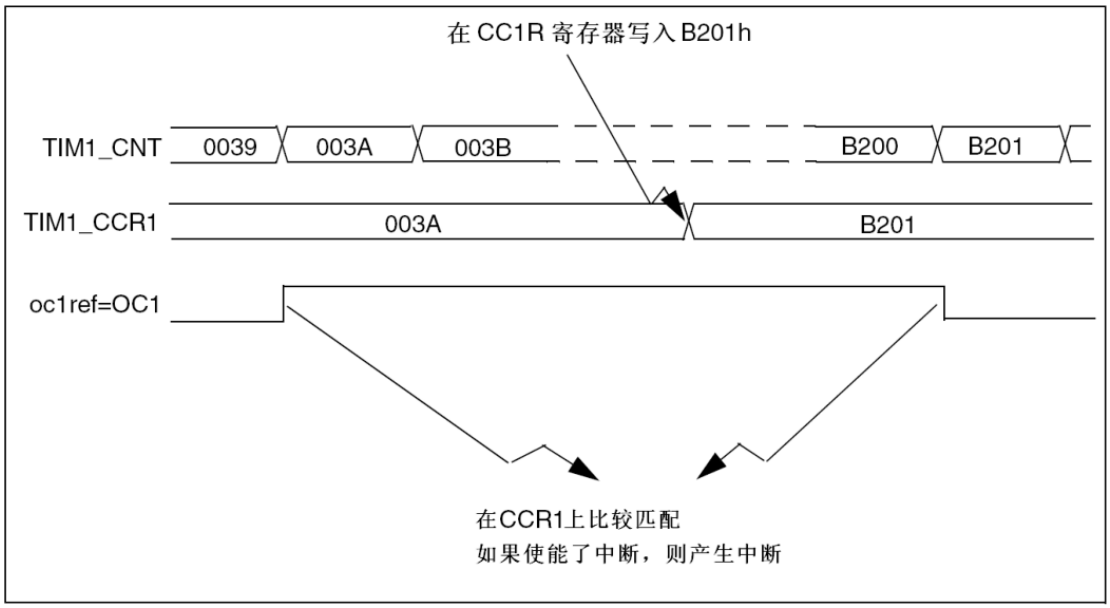


图 14-32

14.3.10. PWM 模式

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

在 TIMERx_CCMRx 寄存器中的 OCxM 位写入“110”(PWM 模式 1)或“111”(PWM 模式 2)，能够独立地设置每个 OCx 输出通道产生一路 PWM。必须通过设置 TIMERx_CCMRx 寄存器的 OCxPE 位使能相应的预装载寄存器，最后还要设置 TIMERx_CR1 寄存器的 ARPE 位，(在向上计数或中心对称模式中)使能自动重装载的预装载寄存器。

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

OCx 的极性可以通过软件在 TIMERx_CCER 寄存器中的 CCxP 位设置，它可以设置为高电平有效或低电平有效。OCx 的输出使能通过(TIMERx_CCER 和 TIMERx_BDTR 寄存器中)CCxE、CCxNE、MOE、OSSI 和 OSSR 位的组合控制。详见 TIMERx_CCER 寄存器 的描述。

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

根据 TIMERx_CR1 寄存器中 CMS 位的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

PWM 边沿对齐模式

- 向上计数的配置

当 TIMERx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。参看下面是一个 PWM 模式 1 的例子。当 TIMERx_CNT<TIMERx_CCRx 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIMERx_CCRx 中的比较值大于自动重装载值(TIMERx_ARR)，则 OCxREF 保持为“1”。如果比较值为 0，则 OCxREF 保持为“0”。下图为 TIMERx_ARR=8 时边沿对齐的 PWM 波形实例。

边沿对齐的 PWM 波形(ARR=8)

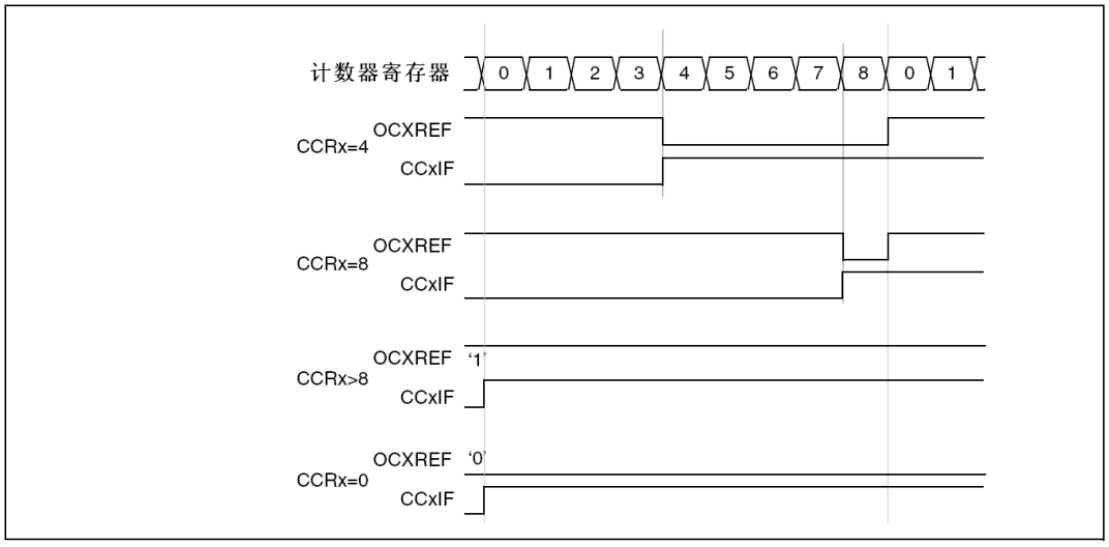


图 14-33

- 向下计数的配置

当 TIMERx_CR1 寄存器的 DIR 位为高时执行向下计数。参看在 PWM 模式 1，当 TIMERx_CNT>TIMERx_CCRx 时参考信号 OCxREF 为低，否则为高。如果 TIMERx_CCRx 中的比较值大于 TIMERx_ARR 中的自动重装载值，则 OCxREF 保持为’1’。该模式下不能产生 0%的 PWM 波形。

PWM 中央对齐模式

当 TIMERx_CR1 寄存器中的 CMS 位不为“00”时为中央对齐模式(所有其他的配置对 OCxREF/OCx 信号都有相同的作用)。根据不同的 CMS 位设置，比较标志可以在计数器向上计数时被置 1、在计数器向下计数时被置 1、或在计数器向上和向下计数时被置 1。TIMERx_CR1 寄存器中的计数方向位(DIR)由硬件更新，不要用软件修改它。下图给出了一些中央对齐的 PWM 波形的例子：

- TIMERx_ARR=8
- PWM 模式 1

- TIMERx_CR1 寄存器的 CMS=01，在中央对齐模式 1 下，当计数器向下计数时设置比较标志。
- 中央对齐的 PWM 波形(APR=8)

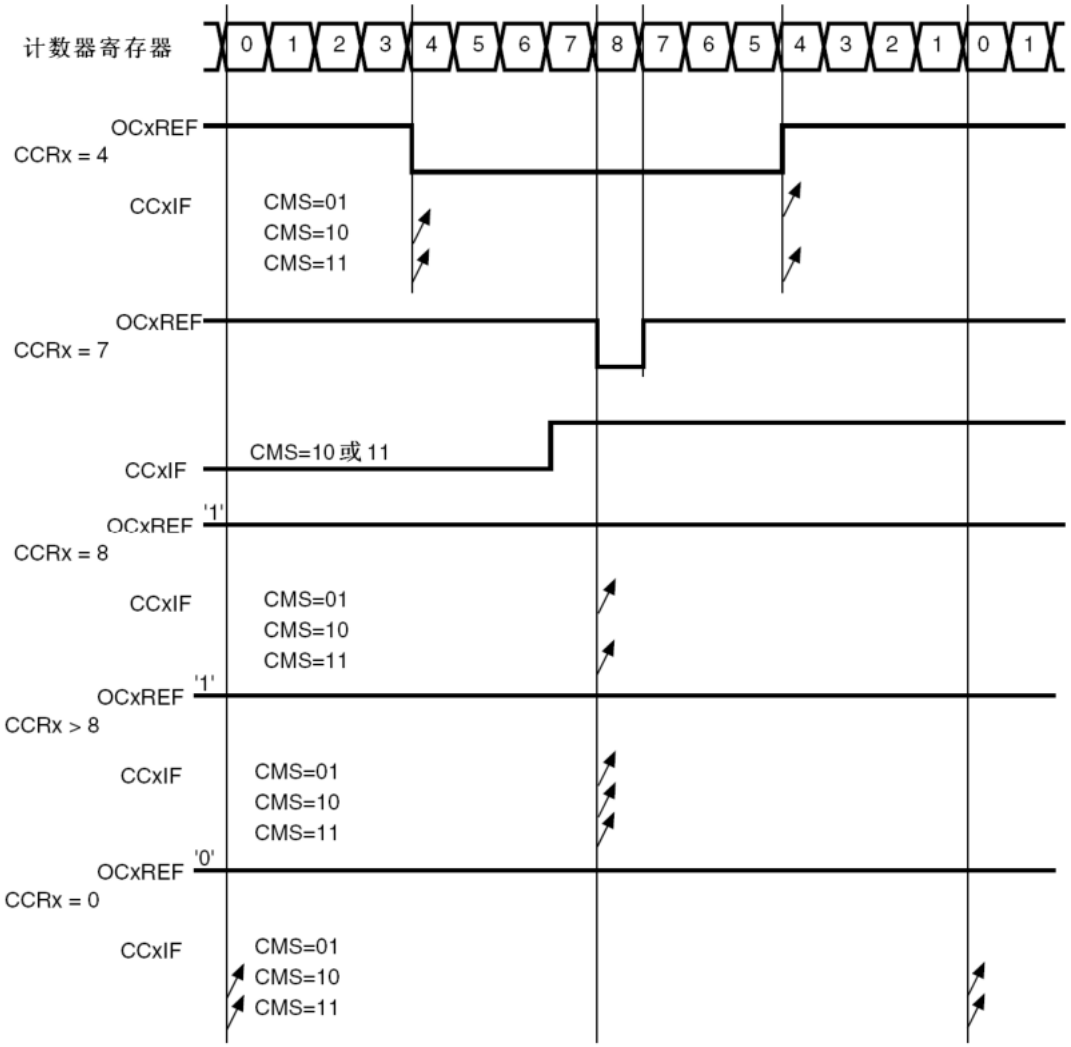


图 14-34

使用中央对齐模式的提示：

- 进入中央对齐模式时，使用当前的向上/向下计数配置；这就意味着计数器向上还是向下计数取决于 TIMERx_CR1 寄存器中 DIR 位的当前值。此外，软件不能同时修改 DIR 和 CMS 位。
- 不推荐当运行在中央对齐模式时改写计数器，因为这会产生不可预知的结果。特别地：
  - 如果写入计数器的值大于自动重加载的值(TIMERx_CNT>TIMERx_ARR)，则方向不会被更新。例如，如果计数器正在向上计数，它就会继续向上计数。
  - 如果将 0 或者 TIMERx_ARR 的值写入计数器，方向被更新，但不产生更新事件 UEV。
- 使用中央对齐模式最保险的方法，就是在启动计数器之前产生一个软件更新(设置 TIMERx_EGR 位中的 UG 位)，并且不要在计数进行过程中修改计数器的值。

14.3.11. 互补输出和死区插入

高级控制定时器能够输出两路互补信号，并且能够管理输出的瞬时关断和接通。

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

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

互补信号 OCx 和 OCxN 通过下列控制位的组合进行控制：TIMERx_CCER 寄存器的 CCxE 和 CCxNE 位，TIMERx_BDTR 和 TIMERx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位，详见 表 15-4 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位 。特别的是，在转换到 IDLE 状态时(MOE 下降到 0)死区被激活。

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

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

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

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

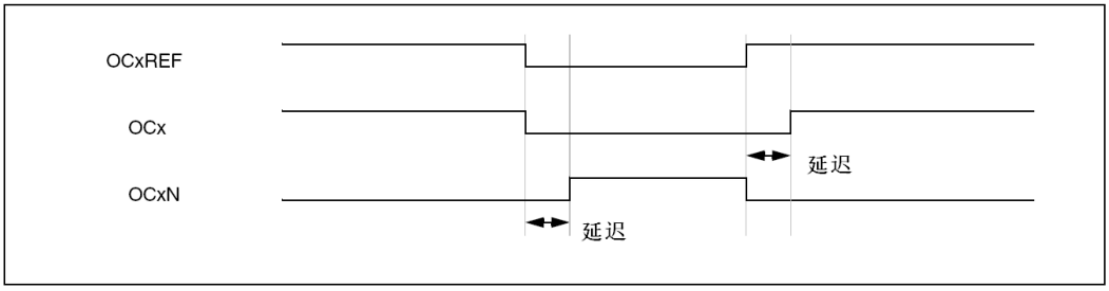


图 14-35 带死区插入的互补输出

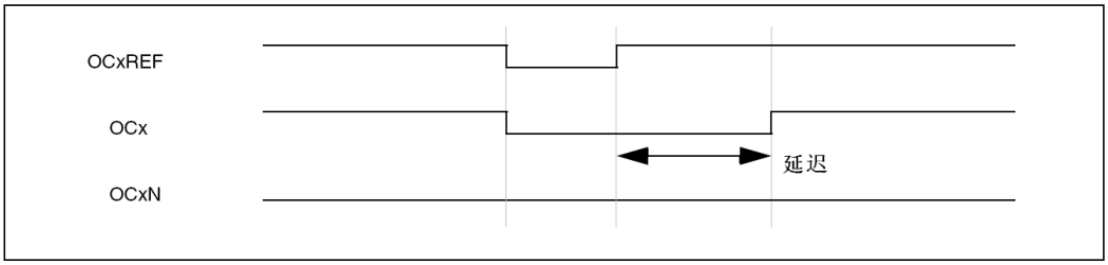


图 14-36 死区波形延迟大于负脉冲

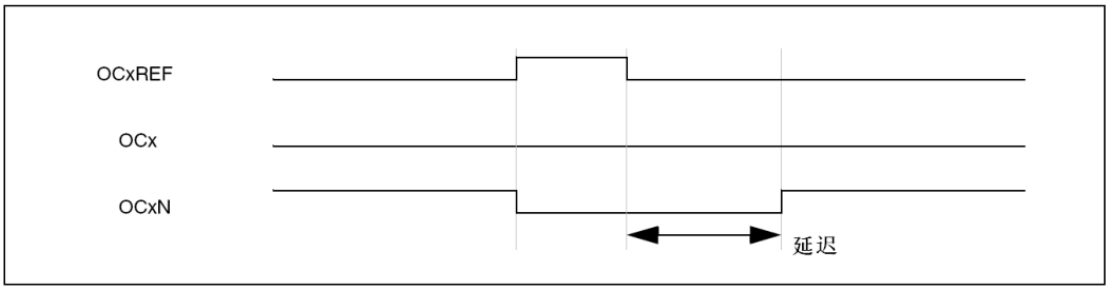


图 14-37 死区波形延迟大于正脉冲

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

重定向 OCxREF 到 OCx 或 OCxN

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

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

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

14.3.12. 使用刹车功能

当使用刹车功能时，依据相应的控制位(TIMERx_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位，TIMERx_CR2 寄存器中的 OISx 和 OISxN 位)，输出使能信号和无效电平都会被修改。但无论何时，OCx 和 OCxN 输出不能在同一时间同时处于有效电平上。详见 表 15-4 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位 。

刹车源既可以是刹车输入引脚，又可以是一个 sram 访问越界事件。

系统复位后，刹车电路被禁止，MOE 位为低。设置 TIMERx_BDTR 寄存器中的 BKE 位可以使能刹车功能，刹车输入信号的极性可以通过配置同一个寄存器中的 BKP 位选择。BKE 和 BKP 可以同时被修改。

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

当发生刹车时(在刹车输入端出现选定的电平)，有下述动作：

- MOE 位被异步地清除，将输出置于无效状态、空闲状态或者复位状态(由 OSSI 位选择)。这个特性在 MCU 的振荡器关闭时依然有效。
- 一旦 MOE=0，每一个输出通道输出由 TIMERx_CR2 寄存器中的 OISx 位设定的电平。如果 OSSI=0，则定时器释放使能输出，否则使能输出始终为高。
- 当使用互补输出时：
  - 输出首先被置于复位状态即无效的状态(取决于极性)。这是异步操作，即使定时器没有时钟时，此功能也有效。
  - 如果定时器的时钟依然存在，死区生成器将会重新生效，在死区之后根据 OISx 和 OISxN 位指示的电平驱动输出端口。即使在这种情况下，OCx 和 OCxN 也不能被同时驱动到有效的电平。注，因为重新同步 MOE，死区时间比通常情况下长一些(大约 2 个 ck_tim 的时钟周期)。
  - 如果 OSSI=0，定时器释放使能输出，否则保持使能输出；或一旦 CCxE 与 CCxNE 之一变高时，使能输出变为高。
- 如果设置了 TIMERx_DIER 寄存器中的 BIE 位，当刹车状态标志(TIMERx_SR 寄存器中的 BIF 位)为“1”时，则产生一个中断。如果设置了 TIMERx_DIER 寄存器中的 BDE 位，则产生一个 DMA 请求。
- 如果设置了 TIMERx_BDTR 寄存器中的 AOE 位，在下一个更新事件 UEV 时 MOE 位被自动置位；例如，这可以用来进行整形。否则，MOE 始终保持低直到被再次置“1”；此时，这个特性可以被用在安全方面，你可以把刹车输入连到电源驱动的报警输出、热敏传感器或者其他安全器件上。

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

刹车由 BRK 输入产生，它的有效极性是可编程的，且由 TIMERx_BDTR 寄存器中的 BKE 位开启。除了刹车输入和输出管理，刹车电路中还实现了写保护以保证应用程序的安全。它允许用户冻结几个配置参数(死区长度，OCx/OCxN 极性和被禁止的状态，OCxM 配置，刹车使能和极性)。

用户可以通过 TIMERx_BDTR 寄存器中的 LOCK 位，从三级保护中选择一种，参看 刹车和死区寄存器(TIMERx_BDTR) 。在 MCU 复位后 LOCK 位只能被修改一次。下图显示响应刹车的输出实例。

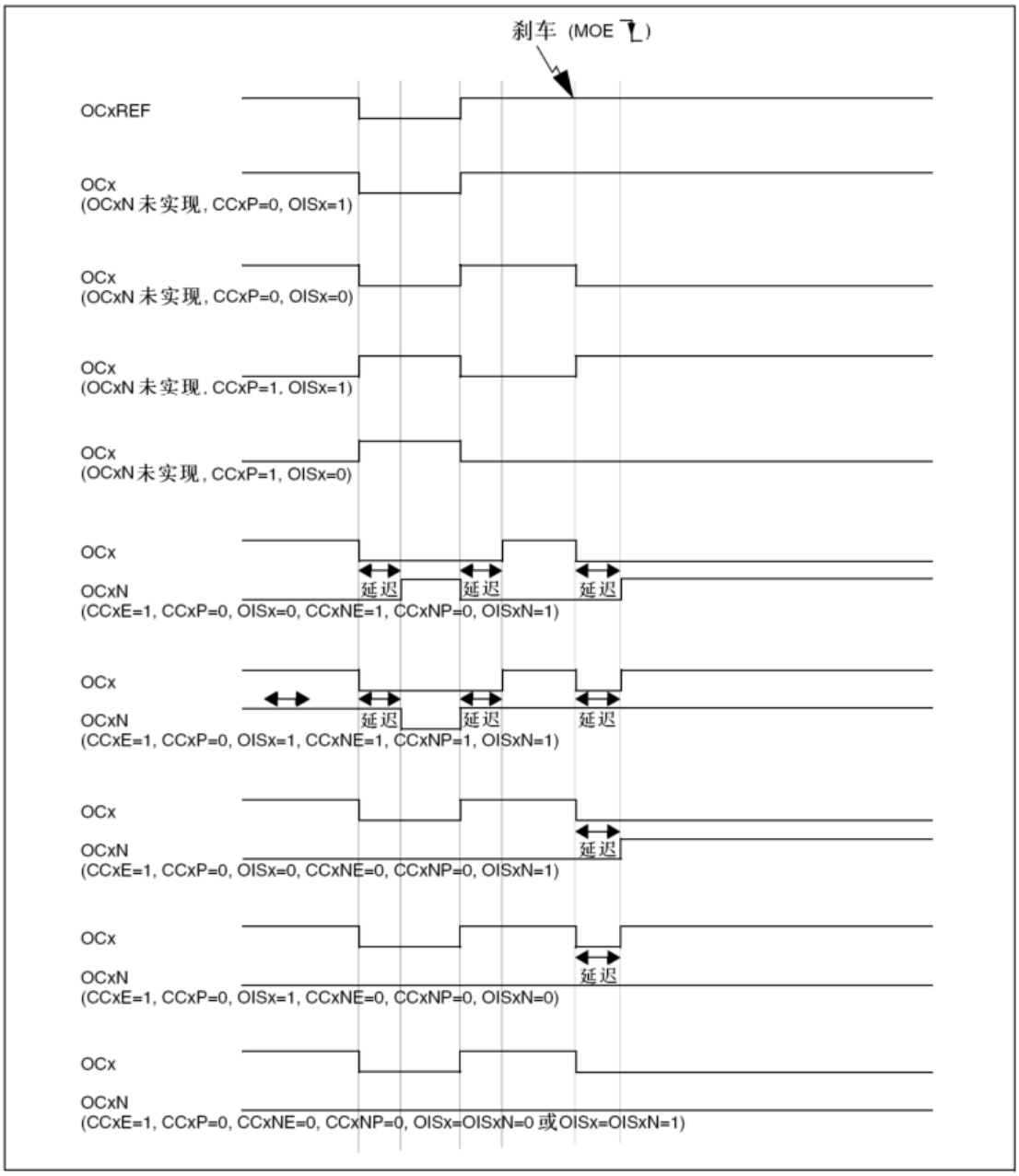


图 14-38 响应刹车的输出

14.3.13. 在外部事件时清除 OCxREF 信号

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

该功能只能用于输出比较和 PWM 模式，而不能用于强置模式。

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

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

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

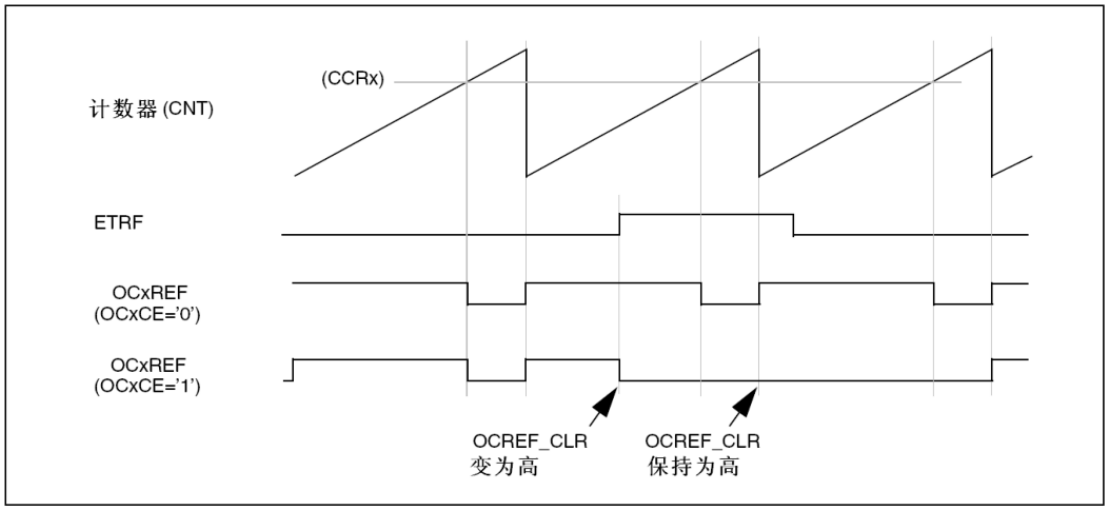


图 14-39 清除 TIMERx 的 OCxREF

14.3.14. 产生六步 PWM 输出

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

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

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

产生六步 PWM，使用 COM 的例子(OSSR=1)

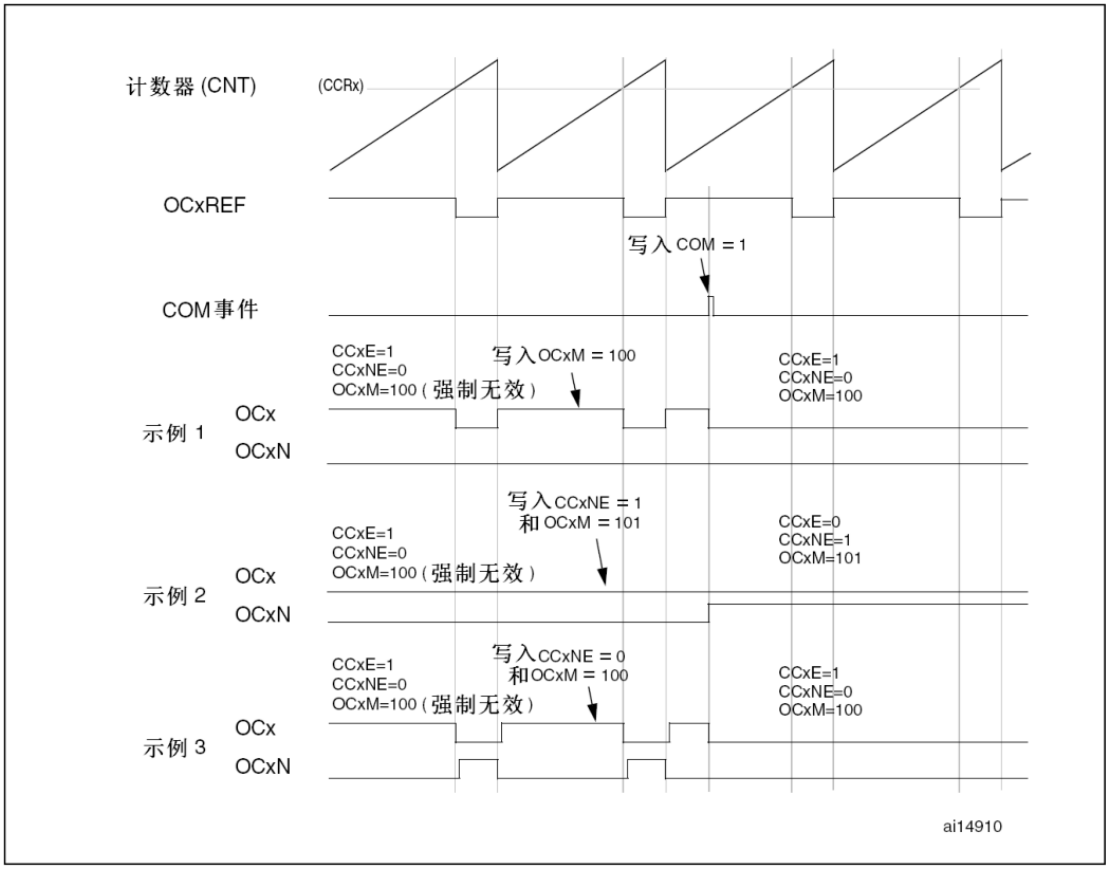


图 14-40

14.3.15. 单脉冲模式

单脉冲模式(OPM)是前述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后产生一个脉宽可程序控制的脉冲。

可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIMERx_CR1 寄存器中的 OPM 位将选择单脉冲模式，这样可以让计数器自动地在产生下一个更新事件 UEV 时停止。

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

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

单脉冲模式的例子

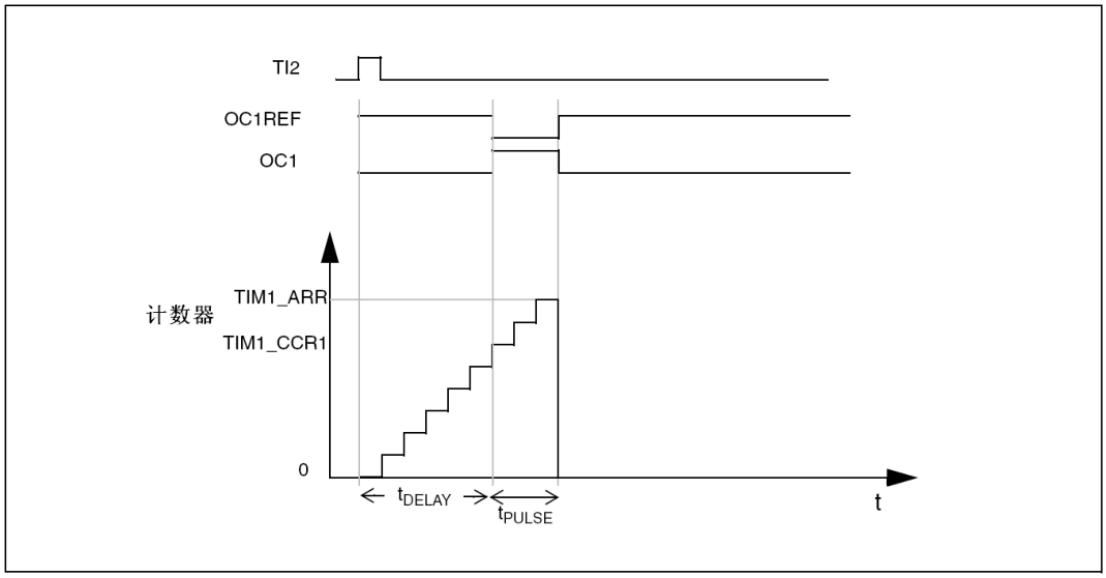


图 14-41

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

假定 TI2FP2 作为触发 1:

- 置 TIMERx_CCMR1 寄存器中的 CC2S=01，把 TI2FP2 映像到 TI2。
- 置 TIMERx_CCER 寄存器中的 CC2P=0，使 TI2FP2 能够检测上升沿。
- 置 TIMERx_SMCR 寄存器中的 TS=110，TI2FP2 作为从模式控制器的触发(TRGI)。
- 置 TIMERx_SMCR 寄存器中的 SMS=110(触发模式)，TI2FP2 被用来启动计数器。

OPM 的波形由写入比较寄存器的数值决定(要考虑时钟频率和计数器预分频器)

- tDELAY 由 TIMERx_CCR1 寄存器中的值定义。
- tPULSE 由自动装载值和比较值之间的差值定义(TIMERx_ARR - TIMERx_CCR1)。
- 假定当发生比较匹配时要产生从 0 到 1 的波形，当计数器达到预装载值时要产生一个从 1 到 0 的波形；首先要置 TIMERx_CCMR1 寄存器的 OC1M=111，进入 PWM 模式 2；根据需要有选择地使能预装载寄存器：置 TIMERx_CCMR1 中的 OC1PE=1 和 TIMERx_CR1 寄存器中的 ARPE；然后在 TIMERx_CCR1 寄存器中填写比较值，在 TIMERx_ARR 寄存器中填写自动装载值，设置 UG 位来产生一个更新事件，然后等待在 TI2 上的一个外部触发事件。本例中，CC1P=0。

  在这个例子中，TIMERx_CR1 寄存器中的 DIR 和 CMS 位应该置低。

  因为只需要一个脉冲，所以必须设置 TIMERx_CR1 寄存器中的 OPM=1，在下一个更新事件(当计数器从自动装载值翻转到 0)时停止计数。

特殊情况： 特殊情况：OCx 快速使能：

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

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

PWM1 和 PWM2 模式时起作用。

14.3.16. 编码器接口模式

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

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

两个输入 TI1 和 TI2 被用来作为增量编码器的接口。参看 表 15-3 ，假定计数器已经启动(TIMERx_CR1 寄存器中的 CEN=1)，则计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和变相，则 TI1FP1=TI1，TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数，同时硬件对 TIMERx_CR1 寄存器的 DIR 位进行相应的设置。

不管计数器是依靠 TI1 计数、依靠 TI2 计数或者同时依靠 TI1 和 TI2 计数，在任一输入端(TI1 或者 TI2)的跳变都会重新计算 DIR 位。

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

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

表 14-3 计数方向与编码器信号的关系

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

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

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

- CC1S=“01”(TIMERx_CCMR1 寄存器，IC1FP1 映射到 TI1)
- CC2S=“01”(TIMERx_CCMR2 寄存器，IC2FP2 映射到 TI2)
- CC1P=“0” (TIMERx_CCER 寄存器，IC1FP1 不反相，IC1FP1=TI1)
- CC2P=“0” (TIMERx_CCER 寄存器，IC2FP2 不反相，IC2FP2=TI2)
- SMS=“011” (TIMERx_SMCR 寄存器，所有的输入均在上升沿和下降沿有效)
- CEN=“1” (TIMERx_CR1 寄存器，计数器使能)

编码器模式下的计数器操作实例

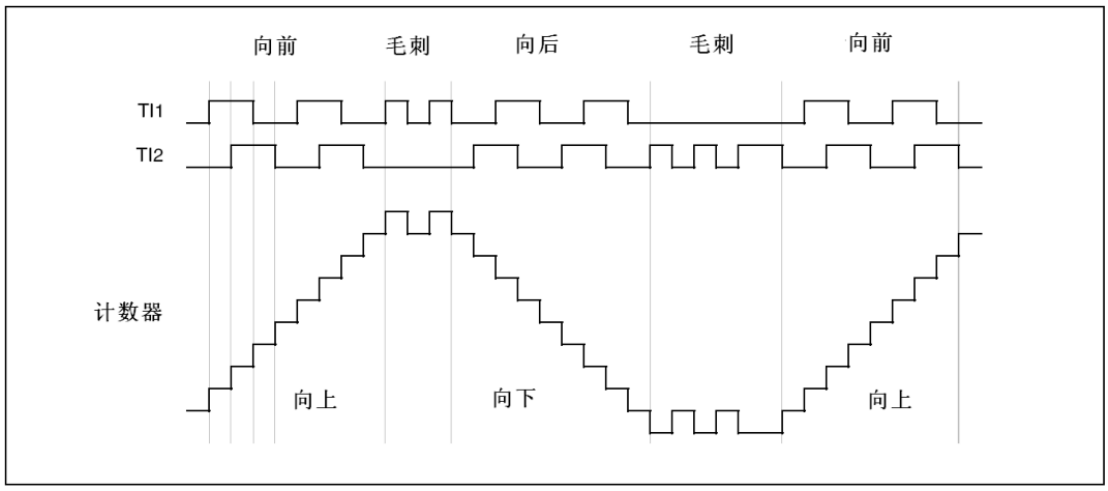


图 14-42

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

IC1FP1 反相的编码器接口模式实例

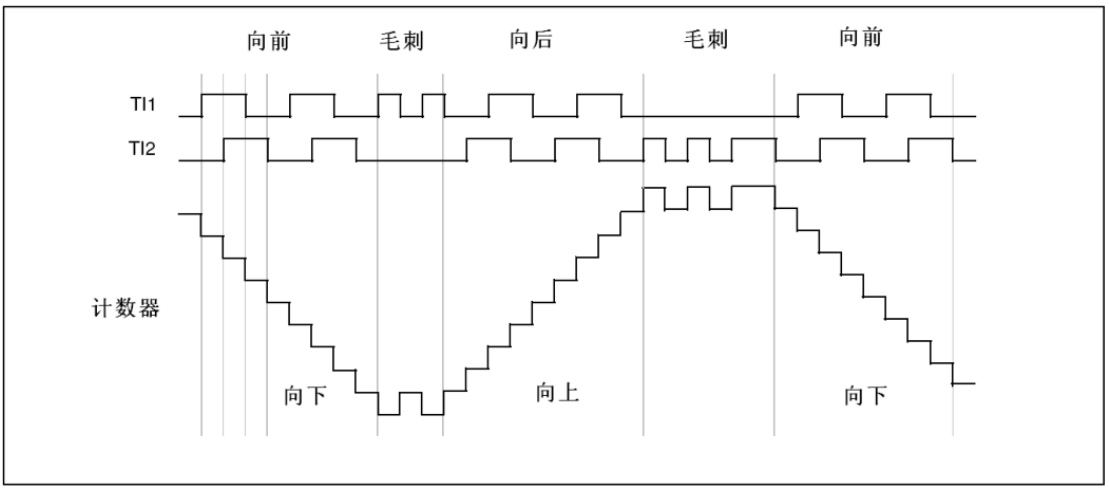


图 14-43

当定时器配置成编码器接口模式时，提供传感器当前位置的信息。使用第二个配置在捕获模式的定时器，可以测量两个编码器事件的间隔，获得动态的信息(速度，加速度，减速度)。指示机械零点的编码器输出可被用做此目的。根据两个事件间的间隔，可以按照固定的时间读出计数器。如果可能的话，你可以把计数器的值锁存到第三个输入捕获寄存器(捕获信号必须是周期的并且可以由另一个定时器产生)；也可以通过一个由实时时钟产生的 DMA 请求来读取它的值。

14.3.17. 定时器输入异或功能

TIMERx_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIMERx_CH1、TIMERx_CH2 和 TIMERx_CH3。

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

14.3.18. 与霍尔传感器的接口

使用高级控制定时器(TIMER1)产生 PWM 信号驱动马达时，可以用另一个通用 TIMERx(TIM2 或 TIM3)定时器作为“接口定时器”来连接霍尔传感器，3 个定时器输入脚(CC1、CC2、CC3)通过一个异或门连接到 TI1 输入通道(通过设置 TIMERx_CR2 寄存器中的 TI1S 位来选择)，“接口定时器”捕获这个信号。

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

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

“接口定时器”可以用来在输出模式产生一个脉冲，这个脉冲可以(通过触发一个 COM 事件)用于改

变高级定时器各个通道的属性，而高级控制定时器产生 PWM 信号驱动马达。
因此“接口定时器”通道必须编程为在一个指定的延时(输出比较或 PWM 模式)之后产生一个正脉冲，这个脉冲通过 TRGO 输出被送到高级控制定时器。
举例：霍尔输入连接到 TIMERx 定时器，要求每次任一霍尔输入上发生变化之后的一个指定的时刻，改变高级控制定时器 TIMERx 的 PWM 配置。

- 置 TIMERx_CR2 寄存器的 TI1S 位为“1”，配置三个定时器输入逻辑或到 TI1 输入
- 时基编程：置 TIMERx_ARR 为其最大值(计数器必须通过 TI1 的变化清零)。设置预分频器得到一个最大的计数器周期，它长于传感器上的两次变化的时间间隔。
- 设置通道 1 为捕获模式(选中 TRC)：置 TIMERx_CCMR1 寄存器中 CC1S=01，如果需要，还可以设置数字滤波器。
- 设置通道 2 为 PWM2 模式，并具有要求的延时：置 TIMERx_CCMR1 寄存器中的 OC2M=111 和 CC2S=00。
- 选择 OC2REF 作为 TRGO 上的触发输出：置 TIMERx_CR2 寄存器中的 MMS=101。

在高级控制寄存器 TIMER1 中，正确的 ITR 输入必须是触发器输入，定时器被编程为产生 PWM 信号，捕获/比较控制信号为预装载的(TIMERx_CR2 寄存器中 CCPC=1)，同时触发输入控制 COM 事件(TIMERx_CR2 寄存器中 CCUS=1)。在一次 COM 事件后，写入下一步的 PWM 控制位(CCxE、OCxM)，这可以在处理 OC2REF 上升沿的中断子程序里实现。
下图显示了这个实例

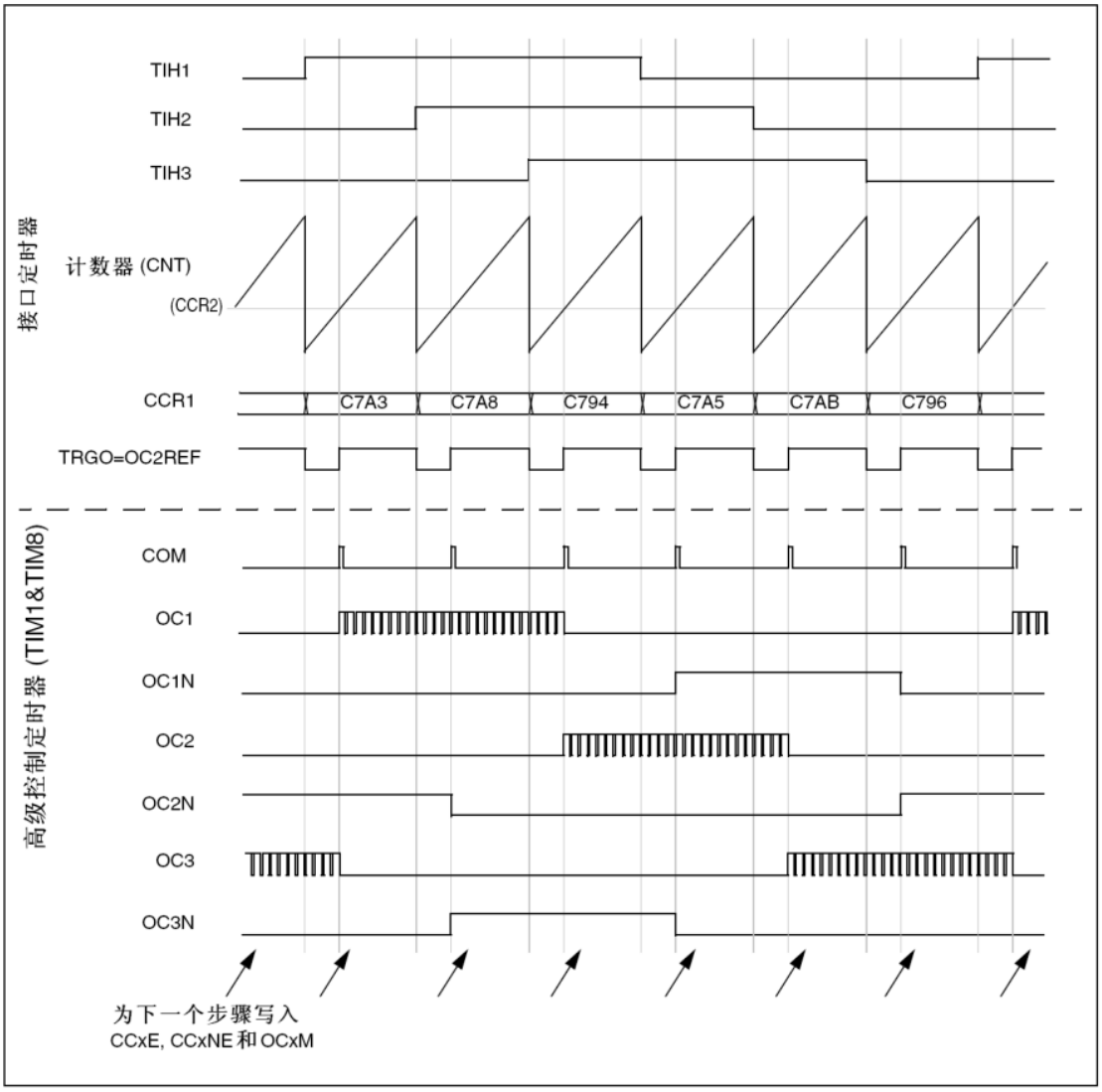


图 14-44 霍尔传感器接口的实例

14.3.19. TIMERx 定时器和外部触发的同步

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

从模式：复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMERx_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器(TIMERx_ARR，TIMERx_CCRx)都被更新了。

在以下的例子中，TI1 输入端的上升沿导致向上计数器被清零：

- 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽(在本例中，不需要任何滤波器，因此保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位只选择输入捕获源，即 TIMERx_CCMR1 寄存器中 CC1S=01。置 TIMERx_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)。
- 置 TIMERx_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIMERx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMERx_CR1 寄存器中 CEN=1，启动计数器。

计数器开始依据内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志(TIMERx_SR 寄存器中的 TIF 位)被设置，根据 TIMERx_DIER 寄存器中 TIE(中断使能)位和 TDE(DMA 使能)位的设置，产生一个中断请求或一个 DMA 请求。

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

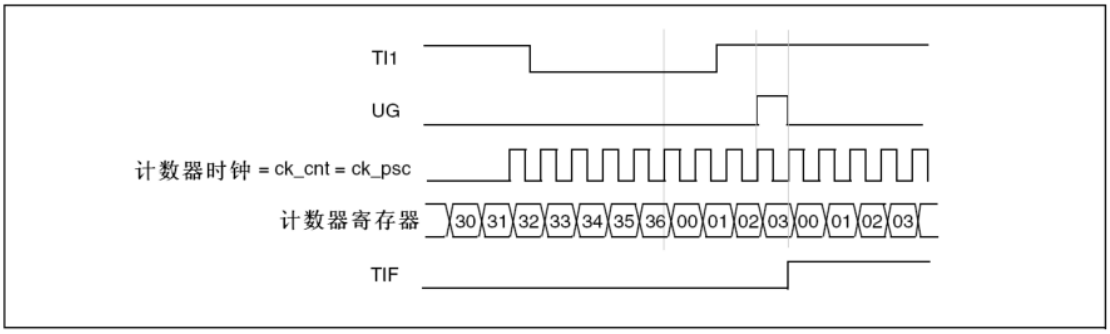


图 14-45

从模式：门控模式

按照选中的输入端电平使能计数器。

在如下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽(本例中，不需要滤波，所以保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输入捕获源，置 TIMERx_CCMR1 寄存器中 CC1S=01。置 TIMERx_CCER 寄存器中 CC1P=1 以确定极性(只检测低电平)。
- 置 TIMERx_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIMERx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMERx_CR1 寄存器中 CEN=1，启动计数器。在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何。

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

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

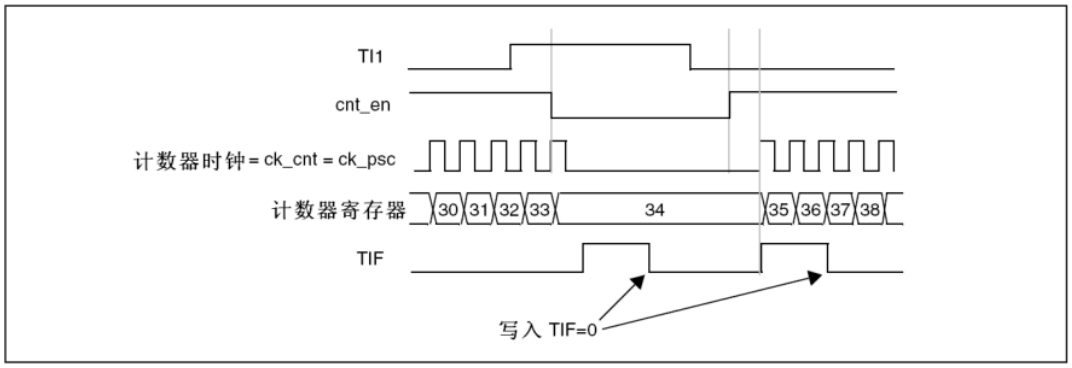


图 14-46

从模式：触发模式

输入端上选中的事件使能计数器。

在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽(本例中，不需要任何滤波器，保持 IC2F=0000)。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIMERx_CCMR1 寄存器中 CC2S=01。置 TIMERx_CCER 寄存器中 CC2P=1 以确定极性(只检测低电平)。
- 置 TIMERx_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIMERx_SMCR 寄存器中 TS=110，选择 TI2 作为输入源。

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

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

触发器模式下的控制电路

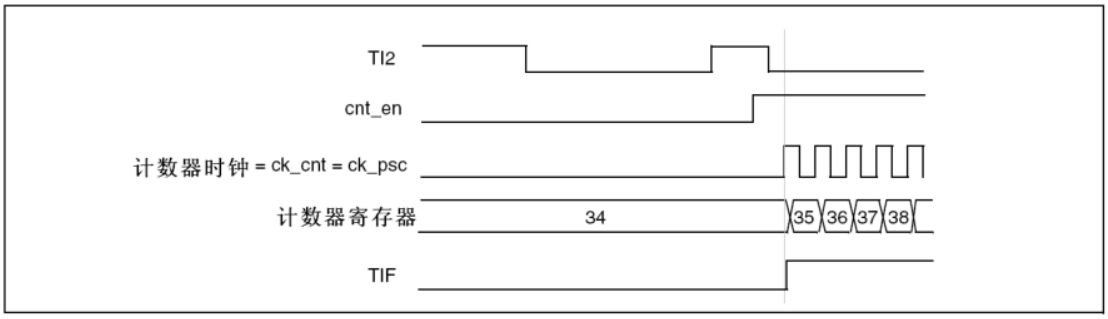


图 14-47

从模式：外部时钟模式 2 + 触触发模式

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

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

1. 通过 TIMERx_SMCR 寄存器配置外部触发输入电路：
   - ETF=0000：没有滤波
   - ETPS=00：不用预分频器
   - ETP=0：检测 ETR 的上升沿，置 ECE=1 使能外部时钟模式 2。
2. 按如下配置通道 1，检测 TI 的上升沿：
   - IC1F=0000：没有滤波
   - 触发操作中不使用捕获预分频器，不需要配置
   - 置 TIMERx_CCMR1 寄存器中 CC1S=01，选择输入捕获源
   - 置 TIMERx_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)
3. 置 TIMERx_SMCR 寄存器中 SMS=110，配置定时器为触发模式。置 TIMERx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。

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

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

外部时钟模式 2 + 触发模式下的控制电路

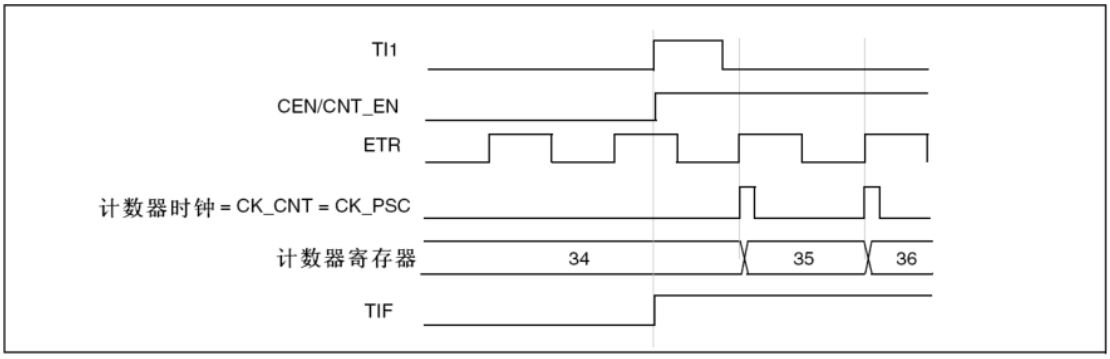


图 14-48

14.3.20. 定时器同步

所有 TIM 定时器在内部相连，用于定时器同步或连接。

14.4. 寄存器描述

14.4.1. 寄存器列表

Base address：0x4002 0000

Name	Offset	Reset	Description
TIMER1_CR1	0x400	32'h0	TIMER1 控制寄存器 1
TIMER1_CR2	0x404	32'h0	TIMER1 控制寄存器 2
TIMER1_SMCR	0x408	32'h0	TIMER1 从模式控制寄存器
TIMER1_DIER	0x40C	32'h0	TIMER1 DMA/中断使能寄存器
TIMER1_SR	0x410	32'h0	TIMER1 状态寄存器
TIMER1_EGR	0x414	32'h0	TIMER1 事件产生寄存器
TIMER1_CCMR1	0x418	32'h0	TIMER1 捕获/比较模式寄存器 1
TIMER1_CCMR2	0x41C	32'h0	TIMER1 捕获/比较模式寄存器 2
TIMER1_CCER	0x420	32'h0	TIMER1 捕获/比较使能寄存器
TIMER1_CNT	0x424	32'h0	TIMER1 计数器
TIMER1_PSC	0x428	32'h0	TIMER1 预分频器
TIMER1_ARR	0x42C	32'hFFFF	TIMER1 自动重载寄存器
TIMER1_RCR	0x430	32'h0	TIMER1 重复计数寄存器
TIMER1_CCR1	0x434	32'hFFFF	TIMER1 捕获/比较寄存器 1
TIMER1_CCR2	0x438	32'hFFFF	TIMER1 捕获/比较寄存器 2
TIMER1_CCR3	0x43C	32'hFFFF	TIMER1 捕获/比较寄存器 3
TIMER1_CCR4	0x440	32'hFFFF	TIMER1 捕获/比较寄存器 4
TIMER1_BDTR	0x444	32'h0	TIMER1 刹车和死区寄存器
TIMER_ALLCON	0x45C	32'h0	TIM1~TIM6 使能/清除控制寄存器

14.4.2. 寄存器详细说明

14.4.2.1. 控制寄存器 1（TIMER1_CR1）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	ETRBKEN	1'b0	RW	外部 IO 刹车输入开关 0: 禁止外部 IO 输入刹车 1: 开启外部 IO 输入刹车 注：当设置了 LOCK 级别 1 时(TIMERx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。
14	SYSBKEN	1'b0	RW	系统错误刹车开关 0: 禁止系统错误刹车 1: 开启系统错误刹车（越界访问 sram） 注：当设置了 LOCK 级别 1 时(TIMERx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。
13	CMP1BKP	1'b0	RW	比较器 1 刹车极性 0: 比较器 1 刹车低电平有效 1: 比较器 1 刹车高电平有效 注：当设置了 LOCK 级别 1 时(TIMERx_BDTR

				寄存器中的 LOCK 位)，该位不能被修改。
12	CMP1BKEN	1'b0	RW	比较器 1 刹车开关 0: 禁止比较器 1 刹车 1: 开启比较器 1 刹车 注：当设置了 LOCK 级别 1 时(TIMERx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。
11	CMP0BKP	1'b0	RW	比较器 0 刹车极性 0: 比较器 0 刹车低电平有效 1: 比较器 0 刹车高电平有效 注：当设置了 LOCK 级别 1 时(TIMERx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。
10	CMP0BKEN	1'b0	RW	比较器 0 刹车开关 0: 禁止比较器 0 刹车 1: 开启比较器 0 刹车 注：当设置了 LOCK 级别 1 时(TIMERx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。
9:8	CKD	2'b0	RW	时钟分频因子 这 2 位定义在定时器时钟(CK_INT)频率、死区时间和由死区发生器与数字滤波器(ETR,TIx)所用的采样时钟之间的分频比例。 b00: tDTS = t CK_INT b01: tDTS = 2 x t CK_INT b10: tDTS = 4 x t CK_INT b11: 保留，不要使用这个配置
7	ARPE	1'b0	RW	自动重装载预装载允许位 0: TIMERx_ARR 寄存器没有缓冲 1: TIMERx_ARR 寄存器被装入缓冲器。
6:5	CMS	2'b0	RW	选择中央对齐模式 b00: 边沿对齐模式。计数器依据方向位(DIR)向上或向下计数。 b01: 中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道(TIMERx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器向下计数时被设置。 b10: 中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道(TIMERx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器向上计数时被设置。 b11: 中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道(TIMERx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，在计数器向上和向下计数时均被设置。 注：在计数器开启时(CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。
4	DIR	1'b0	RW	计数方向 0: 计数器向上计数 1: 计数器向下计数

				注：当计数器配置为中央对齐模式或编码器模式时，该位为只读
3	OPM	1'b0	RW	单脉冲模式 0：在发生更新事件时，计数器不停止 1：在发生下一次更新事件(清除 CEN 位)时，计数器停止
2	URS	1'b0	RW	更新请求源 0：如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求： – 计数器溢出/下溢 – 设置 UG 位 – 从模式控制器产生的更新 1：如果使能了更新中断或 DMA 请求，则只有计数器溢出/下溢才产生更新中断或 DMA 请求。
1	UDIS	1'b0	RW	禁止更新 软件通过该位允许/禁止 UEV 事件的产生 0：允许 UEV。更新(UEV)事件由下述任一事件产生： – 计数器溢出/下溢 – 设置 UG 位 – 从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。(译注：更新影子寄存器) 1：禁止 UEV。不产生更新事件，影子寄存器(ARR、PSC、CCRx)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。
0	CEN	1'b0	RW	使能计数器 0：禁止计数器 1：使能计数器 注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

14.4.2.2. 控制寄存器 2（TIMER1_CR2）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	ADCBKEN	1'b0	RW	ADC 刹车开关 0：禁止 ADC 刹车 1：开启 ADC 刹车 注：当设置了 LOCK 级别 1 时(TIMERx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。
14	OIS4	1'b0	RW	输出空闲状态 4(OC4 输出)。 参见 OIS1 位。
13	OIS3N	1'b0	RW	输出空闲状态 3(OC3N 输出)。 参见 OIS1N 位。
12	OIS3	1'b0	RW	输出空闲状态 3(OC3 输出)。

				参见 OIS1 位。
11	OIS2N	1'b0	RW	输出空闲状态 2(OC2N 输出)。 参见 OIS1N 位。
10	OIS2	1'b0	RW	输出空闲状态 2(OC2 输出)。 参见 OIS1 位。
9	OIS1N	1'b0	RW	输出空闲状态 1(OC1N 输出) 0: 当 MOE=0 时，死区后 OC1N=0 1: 当 MOE=0 时，死区后 OC1N=1 注：已经设置了 LOCK(TIMERx_BKR 寄存器)级别 1、2 或 3 后，该位不能被修改。
8	OIS1	1'b0	RW	输出空闲状态 1(OC1 输出) 0: 当 MOE=0 时，如果实现了 OC1N，则死区后 OC1=0 1: 当 MOE=0 时，如果实现了 OC1N，则死区后 OC1=1 注：已经设置了 LOCK(TIMERx_BKR 寄存器)级别 1、2 或 3 后，该位不能被修改。
7	TI1S	1'b0	RW	TI1 选择 0: TIMERx_CH1 引脚连到 TI1 输入 1: TIMERx_CH1、TIMERx_CH2 和 TIMERx_CH3 引脚经异或后连到 TI1 输入
6:4	MMS	3'b0	RW	主模式选择 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下： b000: 复位 – TIMERx_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式)，则 TRGO 上的信号相对实际的复位会有一个延迟。 b001: 使能 – 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式(见 TIMERx_SMCR 寄存器中 MSM 位的描述)。 b010: 更新 – 更新事件被选为触发输入(TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 b011: 比较脉冲 – 在发生一次捕获或一次比较成功时，当要设置 CC1IF 标志时(即使它已经为高)，触发输出送出一个正脉冲(TRGO)。 b100:比较 – OC1REF 信号被用于作为触发输出(TRGO)。 b101:比较 – OC2REF 信号被用于作为触发输出(TRGO)。 b110:比较 – OC3REF 信号被用于作为触发输

				出(TRGO)。 b111:比较 – OC4REF 信号被用于作为触发输出(TRGO)。
3	CCDS	1'b0	RW	捕获/比较的 DMA 选择 0: 当发生 CCx 事件时，送出 CCx 的 DMA 请求 1: 当发生更新事件时，送出 CCx 的 DMA 请求
2	CCUS	1'b0	RW	捕获/比较控制更新选择 0: 如果捕获/比较控制位是预装载的(CCPC=1)，只能通过设置 COM 位更新它们 1: 如果捕获/比较控制位是预装载的(CCPC=1)，可以通过设置 COM 位或 TRGI 上的一个上升沿更新它们 注：该位只对具有互补输出的通道起作用
1	Reserved	-	-	-
0	CCPC	1'b0	RW	捕获/比较预装载控制位 0: CCxE，CCxNE 和 OCxM 位不是预装载的 1: CCxE，CCxNE 和 OCxM 位是预装载的；设置该位后，它们只在设置了 COM 位后被更新。 注：该位只对具有互补输出的通道起作用。

14.4.2.3. 从模式控制寄存器（TIMER1_SMCR）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	ETP	1'b0	RW	外部触发极性 该位选择是用 ETR 还是 ETR 的反相来作为触发操作 0: ETR 不反相，高电平或上升沿有效 1: ETR 被反相，低电平或下降沿有效
14	ECE	1'b0	RW	外部时钟使能位 该位启用外部时钟模式 2 0: 禁止外部时钟模式 2 1: 使能外部时钟模式 2。计数器由 ETRF 信号上的任意有效边沿驱动 注 1: 设置 ECE 位与选择外部时钟模式 1 并将 TRGI 连到 ETRF(SMS=111 和 TS=111)具有相同功效。 注 2: 下述从模式可以与外部时钟模式 2 同时使用：复位模式，门控模式和触发模式；但是，这时 TRGI 不能连到 ETRF(TS 位不能是“111”)。 注 3: 外部时钟模式 1 和外部时钟模式 2 同时被使能时，外部时钟的输入是 ETRF。
13:12	ETPS	2'b0	RW	外部触发预分频 b00: 关闭预分频 b01: ETRP 频率除以 2 b10: ETRP 频率除以 4 b11: ETRP 频率除以 8

11:8	ETF	4'b0	RW	外部触发滤波 这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。 h0: 无滤波器，以 fDTS 采样 h1: 采样频率 fSAMPLING=fCK_INT，N=2 h2: 采样频率 fSAMPLING=fCK_INT，N=4 h3: 采样频率 fSAMPLING=fCK_INT，N=8 h4: 采样频率 fSAMPLING=fDTS/2，N=6 h5: 采样频率 fSAMPLING=fDTS/2，N=8 h6: 采样频率 fSAMPLING=fDTS/4，N=6 h7: 采样频率 fSAMPLING=fDTS/4，N=8 h8: 采样频率 fSAMPLING=fDTS/8，N=6 h9: 采样频率 fSAMPLING=fDTS/8，N=8 ha: 采样频率 fSAMPLING=fDTS/16，N=5 hb: 采样频率 fSAMPLING=fDTS/16，N=6 hc: 采样频率 fSAMPLING=fDTS/16，N=8 hd: 采样频率 fSAMPLING=fDTS/32，N=5 he: 采样频率 fSAMPLING=fDTS/32，N=6 hf: 采样频率 fSAMPLING=fDTS/32，N=8
7	MSM	1'b0	RW	主/从模式 0: 无作用 1: 触发输入(TRGI)上的事件被延迟了，以允许在当前定时器(通过 TRGO)与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的
6:4	TS	3'b0	RW	触发选择 这 3 位选择用于同步计数器的触发输入。 b000: 内部触发 0(ITR0) b001: 内部触发 1(ITR1) b010: 内部触发 2(ITR2) b011: 内部触发 3(ITR3) b100: TI1 的边沿检测器(TI1F_ED) b101: 滤波后的定时器输入 1(TI1FP1) b110: 滤波后的定时器输入 2(TI2FP2) b111: 外部触发输入(ETRF)
3	Reserved	-	-	-
2:0	SMS	3'b0	RW	从模式选择 当选择了外部信号，触发信号(TRGI)的有效边沿与选中的外部输入极性相关(见输入控制寄存器和控制寄存器的说明) b000: 关闭从模式 – 如果 CEN=1，则预分频器直接由内部时钟驱动。 b001: 编码器模式 1 – 根据 TI1FP1 的电平，计数器在 TI2FP2 的边沿向上/下计数。 b010: 编码器模式 2 – 根据 TI2FP2 的电平，

				计数器在 TI1FP1 的边沿向上/下计数。 b011: 编码器模式 3 – 根据另一个信号的输入电平，计数器在 TI1FP1 和 TI2FP2 的边沿向上/下计数。 b100: 复位模式 – 选中的触发输入(TRGI)的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。 b101: 门控模式 – 当触发输入(TRGI)为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止(但不复位)。计数器的启动和停止都是受控的。 b110: 触发模式 – 计数器在触发输入 TRGI 的上升沿启动(但不复位)，只有计数器的启动是受控的。 b111: 外部时钟模式 1 – 选中的触发输入(TRGI)的上升沿驱动计数器。 注： 1. 如果 TI1F_ED 被选为触发输入(TS=100)时，不要使用门控模式。这是因为，TI1F_ED 在每次 TI1F 变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。 2. 如果配置成编码器模式，且 ECE=0，TIMx_ETR 的上升沿会对计数器清零。

TIMER1 内部触发链接

从定时器	ITR0 (TS=000)	ITR1 (TS=001)	ITR2 (TS=010)	ITR3 (TS=011)
TIMER1	TIMER2	TIMER3	TIMER4	TIMER5

14.4.2.4. DMA/中断使能寄存器（TIMER1_DIER）

Width	Name	Reset	Property	Description
31:15	Reserved	-	-	-
14	TDE	1'b0	RW	允许触发 DMA 请求 0: 禁止触发 DMA 请求 1: 允许触发 DMA 请求
13	COMDE	1'b0	RW	允许 COM 的 DMA 请求 0: 禁止 COM 的 DMA 请求 1: 允许 COM 的 DMA 请求
12	CC4DE	1'b0	RW	允许捕获/比较 4 的 DMA 请求 0: 禁止捕获/比较 4 的 DMA 请求 1: 允许捕获/比较 4 的 DMA 请求
11	CC3DE	1'b0	RW	允许捕获/比较 3 的 DMA 请求 0: 禁止捕获/比较 3 的 DMA 请求 1: 允许捕获/比较 3 的 DMA 请求
10	CC2DE	1'b0	RW	允许捕获/比较 2 的 DMA 请求 0: 禁止捕获/比较 2 的 DMA 请求 1: 允许捕获/比较 2 的 DMA 请求
9	CC1DE	1'b0	RW	允许捕获/比较 1 的 DMA 请求

				0: 禁止捕获/比较 1 的 DMA 请求 1: 允许捕获/比较 1 的 DMA 请求
8	UDE	1'b0	RW	允许更新的 DMA 请求 0: 禁止更新的 DMA 请求 1: 允许更新的 DMA 请求
7	BIE	1'b0	RW	允许刹车中断 0: 禁止刹车中断 1: 允许刹车中断
6	TIE	1'b0	RW	触发中断使能 0: 禁止触发中断 1: 使能触发中断
5	COMIE	1'b0	RW	允许 COM 中断 0: 禁止 COM 中断 1: 允许 COM 中断
4	CC4IE	1'b0	RW	允许捕获/比较 4 中断 0: 禁止捕获/比较 4 中断 1: 允许捕获/比较 4 中断
3	CC3IE	1'b0	RW	允许捕获/比较 3 中断 0: 禁止捕获/比较 3 中断 1: 允许捕获/比较 3 中断
2	CC2IE	1'b0	RW	允许捕获/比较 2 中断 0: 禁止捕获/比较 2 中断 1: 允许捕获/比较 2 中断
1	CC1IE	1'b0	RW	允许捕获/比较 1 中断 0: 禁止捕获/比较 1 中断 1: 允许捕获/比较 1 中断
0	UIE	1'b0	RW	允许更新中断 0: 禁止更新中断 1: 允许更新中断

14.4.2.5. 状态寄存器（TIMER1_SR)

Width	Name	Reset	Property	Description
31:13	Reserved	-	-	-
12	CC4OF	1'b0	RC	捕获 4 重复捕获标记 参见 CC1OF 描述。
11	CC3OF	1'b0	RC	捕获 3 重复捕获标记 参见 CC1OF 描述。
10	CC2OF	1'b0	RC	捕获 2 重复捕获标记 参见 CC1OF 描述。
9	CC1OF	1'b0	RC	捕获 1 重复捕获标记 0: 无重复捕获产生 1: 计数器的值被捕获到 TIMERx_CCR1 寄存器时，CC1IF 的状态已经为‘1’。
8	Reserved	-	-	-
7	BIF	1'b0	RC	刹车中断标记 一旦刹车输入有效，由硬件对该位置 1，如果刹车输入无效，则该位可由软件清 0。

				0: 无刹车事件产生 1: 刹车输入上检测到有效电平
6	TIF	1'b0	RC	触发器中断标记 当发生触发事件(当从模式控制器处于除门控模式外的其它模式时，在 TRGI 输入端检测到有效边沿，或门控模式下的任一边沿)时由硬件对该位置 1，它由软件清 0。 0: 无触发器事件产生 1: 触发中断等待响应
5	COMIF	1'b0	RC	COM 中断标记 一旦产生 COM 事件(当捕获/比较控制位: CCxE、CCxNE、OCxM 已被更新)该位由硬件置 1，它由软件清 0。 0: 无 COM 事件产生 1: COM 中断等待响应
4	CC4IF	1'b0	RC	捕获/比较 4 中断标记 参考 CC1IF 描述
3	CC3IF	1'b0	RC	捕获/比较 3 中断标记 参考 CC1IF 描述
2	CC2IF	1'b0	RC	捕获/比较 2 中断标记 参考 CC1IF 描述
1	CC1IF	1'b0	RC	捕获/比较 1 中断标记 **如果通道 如果通道 CC1 配置为输出模式:** 当计数器值与比较值匹配时该位由硬件置 1，但在中心对称模式下除外(参考 TIMERx_CR1 寄存器的 CMS 位)。它由软件清 0。 0: 无匹配发生; 1: TIMERx_CNT 的值与 TIMERx_CCR1 的值匹配。 当 TIMERx_CCR1 的内容大于 TIMERx_ARR 的内容时，在向上或向上/下计数模式时计数器溢出，或向下计数模式时的计数器下溢条件下，CC1IF 位变高 **如果通道 CC1 配置为输入模式:** 当捕获事件发生时该位由硬件置 1，它由软件清 0 或通过读 TIMERx_CCR1 清 0。 0: 无输入捕获产生 1: 计数器值已被捕获(拷贝)至 TIMERx_CCR1(在 IC1 上检测到与所选极性相同的边沿)
0	UIF	1'b0	RC	更新中断标记 当产生更新事件时该位由硬件置 1。它由软件清 0。 0: 无更新事件产生; 1: 更新中断等待响应。当寄存器被更新时该位由硬件置 1: – 若 TIMERx_CR1 寄存器的 UDIS=0，当重复

				计数器数值上溢或下溢时(重复计数器=0 时产生更新事件)。 – 若 TIMERx_CR1 寄存器的 URS=0、UDIS=0，当设置 TIMERx_EGR 寄存器的 UG=1 时产生更新事件，通过软件对计数器 CNT 重新初始化时。 – 若 TIMERx_CR1 寄存器的 URS=0、UDIS=0，当计数器 CNT 被触发事件重新初始化时。(参考从模式控制寄存器 TIMERx_SMCR)。

14.4.2.6. 事件产生寄存器（TIMER1_EGR)

Width	Name	Reset	Property	Description
31:8	Reserved	-	-	-
7	BG	1'b0	WO	产生刹车事件 该位由软件置 1，用于产生一个刹车事件，由硬件自动清 0。 0: 无动作 1: 产生一个刹车事件。此时 MOE=0、BIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA
6	TG	1'b0	WO	产生触发事件 0: 无动作 1: TIMERx_SR 寄存器的 TIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA
5	COMG	1'b0	WO	捕获/比较事件，产生控制更新 该位由软件置 1，由硬件自动清 0 0: 无动作 1: 当 CCPC=1，允许更新 CCxE、CCxNE、OCxM 位 注：该位只对拥有互补输出的通道有效。
4	CC4G	1'b0	WO	产生捕获/比较 4 事件 参考 CC1G 描述。
3	CC3G	1'b0	WO	产生捕获/比较 3 事件 参考 CC1G 描述。
2	CC2G	1'b0	WO	产生捕获/比较 2 事件 参考 CC1G 描述。
1	CC1G	1'b0	WO	产生捕获/比较 1 事件 该位由软件置 1，用于产生一个捕获/比较事件，由硬件自动清 0。 0: 无动作 1: 在通道 CC1 上产生一个捕获/比较事件 **若通道 CC1 配置为输出：** 设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。 **若通道 CC1 配置为输入：** 当前的计数器值被捕获至 TIMERx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，

				则设置 CC1OF=1。
0	UG	1'b0	WO	产生更新事件 该位由软件置 1，由硬件自动清 0。 0: 无动作; 1: 重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清 0(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清 0;若 DIR=1(向下计数)则计数器取 TIMERx_ARR 的值。

14.4.2.7. 捕获/比较模式寄存器 1（TIMER1_CCMR1）

通道可用于输入(捕获模式)或输出(比较模式)，通道的方向由相应的 CCxS 位定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

输出比较模式：

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	OC2CE	1'b0	RW	输出比较 2 清 0 使能 参考 CC1CE
14:12	OC2M	3'b0	RW	输出比较 2 模式 参考 OC1M
11	OC2PE	1'b0	RW	输出比较 2 预装载使能 参考 OC1PE
10	OC2FE	1'b0	RW	输出比较 2 快速使能 参考 OC1FE
9:8	CC2S	2'b0	RW	捕获/比较 2 选择 该位定义通道的方向(输入/输出)，及输入脚的选择: b00: CC2 通道被配置为输出 b01: CC2 通道被配置为输入，IC2 映射在 TI2 上 b10: CC2 通道被配置为输入，IC2 映射在 TI1 上 b11: CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择) 注：CC2S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC2E=0)才是可写的
7	OC1CE	1'b0	RW	输出比较 1 清 0 使能 0: OC1REF 不受 ETRF 输入的影响 1: 一旦检测到 ETRF 输入高电平，清除 OC1REF=0
6:4	OC1M	3'b0	RW	输出比较 1 模式 该 3 位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效，而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。

				b000: 冻结。输出比较寄存器 TIMERx_CCR1 与计数器 TIMERx_CNT 间的比较对 OC1REF 不起作用 b001: 匹配时设置通道 1 为有效电平。当计数器 TIMERx_CNT 的值与捕获/比较寄存器 1(TIMERx_CCR1)相同时，强制 OC1REF 为高 b010: 匹配时设置通道 1 为无效电平。当计数器 TIMERx_CNT 的值与捕获/比较寄存器 1(TIMERx_CCR1)相同时，强制 OC1REF 为低 b011: 翻转。当 TIMERx_CCR1=TIMERx_CNT 时，翻转 OC1REF 的电平 b100: 强制为无效电平。强制 OC1REF 为低 b101: 强制为有效电平。强制 OC1REF 为高 b110: PWM 模式 1 - 在向上计数时，一旦 TIMERx_CNT<TIMERx_CCR1 时通道 1 为有效电平，否则为无效电平；在向下计数时，一旦 TIMERx_CNT>TIMERx_CCR1 时通道 1 为无效电平(OC1REF=0)，否则为有效电平(OC1REF=1) b111: PWM 模式 2 - 在向上计数时，一旦 TIMERx_CNT<TIMERx_CCR1 时通道 1 为无效电平，否则为有效电平；在向下计数时，一旦 TIMERx_CNT>TIMERx_CCR1 时通道 1 为有效电平，否则为无效电平 注 1: 一旦 LOCK 级别设为 3(TIMERx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。
3	OC1PE	1'b0	RW	输出比较 1 预装载使能 0: 禁止 TIMERx_CCR1 寄存器的预装载功能，可随时写入 TIMERx_CCR1 寄存器，并且新写入的数值立即起作用。 1: 开启 TIMERx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMERx_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。 注 1: 一旦 LOCK 级别设为 3(TIMERx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 仅在单脉冲模式下(TIMERx_CR1 寄存器的 OPM=1)，可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定。
2	OC1FE	1'b0	RW	输出比较 1 快速使能 该位用于加快 CC 输出对触发输入事件的响应。 0: 根据计数器与 CCR1 的值，CC1 正常操作，

				即使触发器是打开的。当触发器的输入有一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期。 1：输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。 OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。
1:0	CC1S	2'b0	RW	捕获/比较 1 选择 这 2 位定义通道的方向(输入/输出),及输入脚的选择： b00：CC1 通道被配置为输出； b01：CC1 通道被配置为输入，IC1 映射在 TI1 上； b10：CC1 通道被配置为输入，IC1 映射在 TI2 上； b11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择)。 注：CC1S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC1E=0)才是可写的。

输入捕获模式：

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:12	IC2F	4'b0	RW	输入捕获 2 滤波器 参考 IC1F
11:10	IC2PSC	2'b0	RW	输入/捕获 2 预分频器 参考 IC1PSC
9:8	CC2S	2'b0	RW	捕获/比较 2 选择 这 2 位定义通道的方向(输入/输出),及输入脚的选择： b00: CC2 通道被配置为输出 b01: CC2 通道被配置为输入，IC2 映射在 TI2 上 b10: CC2 通道被配置为输入，IC2 映射在 TI1 上 b11: CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择) 注：CC2S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC2E=0)才是可写的。
7:4	IC1F	4'b0	RW	输入捕获 1 滤波器 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它

				记录到 N 个事件后会产生一个输出的跳变: h0: 无滤波器，以 fDTS 采样 h1: 采样频率 fSAMPLING=fCK_INT，N=2 h2: 采样频率 fSAMPLING=fCK_INT，N=4 h3: 采样频率 fSAMPLING=fCK_INT，N=8 h4: 采样频率 fSAMPLING=fDTS/2，N=6 h5: 采样频率 fSAMPLING=fDTS/2，N=8 h6: 采样频率 fSAMPLING=fDTS/4，N=6 h7: 采样频率 fSAMPLING=fDTS/4，N=8 h8: 采样频率 fSAMPLING=fDTS/8，N=6 h9: 采样频率 fSAMPLING=fDTS/8，N=8 ha: 采样频率 fSAMPLING=fDTS/16，N=5 hb: 采样频率 fSAMPLING=fDTS/16，N=6 hc: 采样频率 fSAMPLING=fDTS/16，N=8 hd: 采样频率 fSAMPLING=fDTS/32，N=5 he: 采样频率 fSAMPLING=fDTS/32，N=6 hf: 采样频率 fSAMPLING=fDTS/32，N=8
3:2	IC1PSC	2'b0	RW	输入/捕获 1 预分频器 这 2 位定义了 CC1 输入(IC1)的预分频系数。一旦 CC1E=0(TIMERx_CCER 寄存器中)，则预分频器复位。 b00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 b01: 每 2 个事件触发一次捕获 b10: 每 4 个事件触发一次捕获 b11: 每 8 个事件触发一次捕获
1:0	CC1S	2'b0	RW	捕获/比较 1 选择 这 2 位定义通道的方向(输入/输出),及输入脚的选择: b00: CC1 通道被配置为输出; b01: CC1 通道被配置为输入，IC1 映射在 TI1 上; b10: CC1 通道被配置为输入，IC1 映射在 TI2 上; b11: CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择)。 注: CC1S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC1E=0)才是可写的。

14.4.2.8. 捕获/比较模式寄存器 2（TIMER1_CCMR2）

输出比较模式:

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	OC4CE	1'b0	RW	输出比较 4 清 0 使能 参考 CC1CE
14:12	OC4M	3'b0	RW	输出比较 4 模式

				参考 OC1M
11	OC4PE	1'b0	RW	输出比较 4 预装载使能 参考 OC1PE
10	OC4FE	1'b0	RW	输出比较 4 快速使能 参考 OC1FE
9:8	CC4S	2'b0	RW	捕获/比较 4 选择 该位定义通道的方向(输入/输出)，及输入脚的选择： b00: CC4 通道被配置为输出 b01: CC4 通道被配置为输入，IC4 映射在 TI4 上 b10: CC4 通道被配置为输入，IC4 映射在 TI3 上 b11: CC4 通道被配置为输入，IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择) 注：CC4S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC4E=0)才是可写的
7	OC3CE	1'b0	RW	输出比较 1 清 0 使能 参考 CC1CE
6:4	OC3M	3'b0	RW	输出比较 3 模式 参考 OC1M
3	OC3PE	1'b0	RW	输出比较 3 预装载使能 参考 OC1PE
2	OC3FE	1'b0	RW	输出比较 3 快速使能 参考 OC1FE
1:0	CC3S	2'b0	RW	捕获/比较 3 选择 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： b00：CC3 通道被配置为输出； b01：CC3 通道被配置为输入，IC3 映射在 TI3 上； b10：CC3 通道被配置为输入，IC3 映射在 TI4 上； b11：CC3 通道被配置为输入，IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择)。 注：CC3S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC3E=0)才是可写的。

输入捕获模式：

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:12	IC4F	4'b0	RW	输入捕获 4 滤波器 参考 IC1F
11:10	IC4PSC	2'b0	RW	输入/捕获 4 预分频器 参考 IC1PSC

9:8	CC4S	2'b0	RW	捕获/比较 4 选择 这 2 位定义通道的方向(输入/输出),及输入脚的选择: b00: CC4 通道被配置为输出 b01: CC4 通道被配置为输入，IC4 映射在 TI4 上 b10: CC4 通道被配置为输入，IC4 映射在 TI3 上 b11: CC4 通道被配置为输入，IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择) 注：CC4S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC4E=0)才是可写的。
7:4	IC3F	4'b0	RW	输入捕获 3 滤波器 参考 IC1F
3:2	IC3PSC	2'b0	RW	输入/捕获 3 预分频器 参考 IC1PSC
1:0	CC3S	2'b0	RW	捕获/比较 3 选择 这 2 位定义通道的方向(输入/输出),及输入脚的选择: b00: CC3 通道被配置为输出; b01: CC3 通道被配置为输入，IC3 映射在 TI3 上; b10: CC3 通道被配置为输入，IC3 映射在 TI4 上; b11: CC3 通道被配置为输入，IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择)。 注：CC3S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC3E=0)才是可写的。

14.4.2.9. 捕获/比较使能寄存器（TIMER1_CCER)

Width	Name	Reset	Property	Description
31:14	Reserved	-	-	-
13	CC4P	1'b0	RW	输入/捕获 4 输出极性 参考 CC1P 的描述。
12	CC4E	1'b0	RW	输入/捕获 4 输出使能 参考 CC1E 的描述。
11	CC3NP	1'b0	RW	输入/捕获 3 互补输出极性 参考 CC1NP 的描述。
10	CC3NE	1'b0	RW	输入/捕获 3 互补输出使能 参考 CC1NE 的描述。
9	CC3P	1'b0	RW	输入/捕获 3 输出极性 参考 CC1P 的描述。
8	CC3E	1'b0	RW	输入/捕获 3 输出使能 参考 CC1E 的描述。
7	CC2NP	1'b0	RW	输入/捕获 2 互补输出极性

				参考 CC1NP 的描述。
6	CC2NE	1'b0	RW	输入/捕获 2 互补输出使能 参考 CC1NE 的描述。
5	CC2P	1'b0	RW	输入/捕获 2 输出极性 参考 CC1P 的描述。
4	CC2E	1'b0	RW	输入/捕获 2 输出使能 参考 CC1E 的描述。
3	CC1NP	1'b0	RW	输入/捕获 1 互补输出极性 **CC1 配置为输出：** 0: OC1N 高电平有效 1: OC1N 低电平有效 **CC1 配置为输入：** 用于与 CC1P 一起决定 TI1FP1/TI2FP1 注：一旦 LOCK 级别(TIMERx_BDTR 寄存器中的 LOCK 位)设为 3 或 2 且 CC1S=00(通道配置为输出)则该位不能被修改。
2	CC1NE	1'b0	RW	输入/捕获 1 互补输出使能 0: 关闭 - OC1N 禁止输出，因此 OC1N 的电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。 1: 开启 - OC1N 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。
1	CC1P	1'b0	RW	输入/捕获 1 输出极性 **CC1 通道配置为输出：** 0: OC1 高电平有效; 1: OC1 低电平有效。 **CC1 通道配置为输入：** CC1NP 和 CC1P 一起决定 TI1FP1 和 TI2FP1 极性 b00: 不反相，上升沿。 TIxFP1 上升沿有效（捕获，复位模式，外部时钟，触发模式），TIxFP1 不反向（门控模式，正交编码模式） b01: 反相，下降沿。 TIxFP1 下降沿有效（捕获，复位模式，外部时钟，触发模式），TIxFP1 反向（门控模式，正交编码模式） b10: reserved b11:不反向，上升沿和下降沿都有效。该配置不能用于正交编码模式。 注：一旦 LOCK 级别(TIMERx_BDTR 寄存器中的 LOCK 位)设为 3 或 2，则该位不能被修改。
0	CC1E	1'b0	RW	输入/捕获 1 输出使能 **CC1 通道配置为输出：** 0: 关闭 - OC1 禁止输出，因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N

				和 CC1NE 位的值。 1: 开启 - OC1 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 **CC1 通道配置为输入：** 该位决定了计数器的值是否能捕获入 TIMERx_CCR1 寄存器。 0: 捕获禁止； 0: 捕获使能。

表 14-4 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位

MOE	OSSI	OSSR	CCxE	CCxNE	OCx 输出状态	OCxN 输出状态
1	x	0	0	0	输出禁止(与定时器断开) OCx=0，OCx_EN=0	输出禁止(与定时器断开) OCxN=0，OCxN_EN=0
		0	0	1	输出禁止(与定时器断开) OCx=0，OCx_EN=0	OCxREF + 极性， OCxN= OCxREF xor CCxNP， OCxN_EN=1
		0	1	0	OCxREF + 极性， OCx= OCxREF xor CCxP， OCx_EN=1	输出禁止(与定时器断开) OCxN=0，OCxN_EN=0
		0	1	1	OCxREF + 极性 + 死区， OCx_EN=1	OCxREF 反相+极性+死区， OCxN_EN=1
		1	0	0	输出禁止(与定时器断开) OCx=CCxP，OCx_EN=0	输出禁止(与定时器断开) OCxN=CCxNP，OCxN_EN=0
		1	0	1	关闭状态(输出使能且为无效电平) OCx=CCxP，OCx_EN=1	OCxREF + 极性， OCxN= OCxREF xor CCxNP， OCxN_EN=1
		1	1	0	OCxREF + 极性， OCx= OCxREF xor CCxP， OCx_EN=1	关闭状态(输出使能且为无效电平) OCxN=CCxNP，OCxN_EN=1
		1	1	1	OCxREF + 极性 + 死区， OCx_EN=1	OCxREF 反相+极性+死区， OCxN_EN=1
0	0	x	0	0	输出禁止(与定时器断开) 异步地: OCx=CCxP，OCx_EN=0，OCxN=CCxNP，OCxN_EN=0; 若时钟存在：经过一个死区时间后 OCx=OISx，OCxN=OISxN，假设 OISx 与 OISxN 并不都对应 OCx 和 OCxN 的有效电平。	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0		
	1		0	1	关闭状态(输出使能且为无效电平) 异步地: OCx=CCxP，OCx_EN=1，OCxN=CCxNP，OCxN_EN=1; 若时钟存在: 经过一个死区时间后 OCx=OISx，OCxN=OISxN，假设 OISx 与 OISxN 并不都对应 OCx 和 OCxN 的有效电平。	
	1		1	0		
	1		1	1		

注：

1. 如果一个通道的 2 个输出都没有使用(CCxE = CCxNE = 0)，那么 OISx, OISxN, CCxP 和 CCxNP 都必须清零。

2.引脚连接到互补的 OCx 和 OCxN 通道的外部 I/O 引脚的状态，取决于 OCx 和 OCxN 通道状态和 GPIO 控制寄存

器。

3.当 SYSCON1[31]为 1 时，只要通道配置成输出模式，则 IO 一直由 timer 驱动。

当 SYSCON1[31]为 0 时，上表中 OCx_EN/OCxN_EN=1 时，IO 由 timer 驱动，OCx_EN/OCxN_EN=0 时，IO 为浮空态。

14.4.2.10. 计数器（TIMER1_CNT)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	CNT	16'b0	RW	计数器的值

14.4.2.11. 预分频器（TIMER1_PSC)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	PSC	16'b0	RW	预分频器的值 计数器的时钟频率(CK_CNT)等于 fCK_PSC /(PSC[15:0]+1)。 PSC 包含了每次当更新事件产生时，装入当前预分频器寄存器的值；更新事件包括计数器被 TIM_EGR 的 UG 位清 0 或被工作在复位模式的从控制器清 0。

14.4.2.12. 自动重装载寄存器（TIMER1_ARR)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	ARR	16'hffff	RW	自动重装载的值 ARR 包含了将要装载入实际的自动重装载寄存器的值。 当自动重装载的值为空时，计数器不工作。

14.4.2.13. 重复计数寄存器（TIMER1_RCR)

Width	Name	Reset	Property	Description
31:8	Reserved	-	-	-
7:0	REP	8'b0	RW	重复计数器的值 开启了预装载功能后，这些位允许用户设置比较寄存器的更新速率(即周期性地从预装载寄存器传输到当前寄存器)；如果允许产生更新中断，则会同时影响产生更新中断的速率。 每次向下计数器 REP_CNT 达到 0，会产生一个更新事件并且计数器 REP_CNT 重新从 REP 值开始计数。由于 REP_CNT 只有在周期更新事件 U_RC 发生时才重载 REP 值，因此对 TIMERx_RCR 寄存器写入的新值只在下次周期更新事件发生时才起作用。 这意味着在 PWM 模式中，(REP+1)对应着： - 在边沿对齐模式下，PWM 周期的数目； - 在中心对称模式下，PWM 半周期的数目；

14.4.2.14. 捕获/比较寄存器 1（TIMER1_CCR1）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	CCR1	16'hffff	RW	捕获/比较通道 1 的值 **若 CC1 通道配置为输出：** CCR1 包含了装入当前捕获/比较 1 寄存器的值(预装载值)。 如果在 TIMERx_CCMR1 寄存器(OC1PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 1 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMERx_CNT 的比较，并在 OC1 端口上产生输出信号。 **若 CC1 通道配置为输入：** CCR1 包含了由上一次输入捕获 1 事件(IC1)传输的计数器值。 注：当配置为捕获模式时，该寄存器变成只读。

14.4.2.15. 捕获/比较寄存器 2（TIMER1_CCR2）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	CCR2	16'hffff	RW	捕获/比较通道 2 的值 **若 CC2 通道配置为输出：** CCR2 包含了装入当前捕获/比较 2 寄存器的值(预装载值)。 如果在 TIMERx_CCMR2 寄存器(OC2PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 2 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMERx_CNT 的比较，并在 OC2 端口上产生输出信号。 **若 CC2 通道配置为输入：** CCR2 包含了由上一次输入捕获 2 事件(IC2)传输的计数器值。 注：当配置为捕获模式时，该寄存器变成只读。

14.4.2.16. 捕获/比较寄存器 3（TIMER1_CCR3）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	CCR3	16'hffff	RW	捕获/比较通道 3 的值 **若 CC3 通道配置为输出：** CCR3 包含了装入当前捕获/比较 3 寄存器的值(预装载值)。

				如果在 TIMERx_CCMR3 寄存器(OC3PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 3 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMERx_CNT 的比较，并在 OC3 端口上产生输出信号。 **若 CC3 通道配置为输入：** CCR3 包含了由上一次输入捕获 3 事件(IC3)传输的计数器值。 注：当配置为捕获模式时，该寄存器变成只读。

14.4.2.17. 捕获/比较寄存器 4（TIMER1_CCR4）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	CCR4	16'hffff	RW	捕获/比较通道 4 的值 **若 CC4 通道配置为输出：** CCR4 包含了装入当前捕获/比较 4 寄存器的值(预装载值)。 如果在 TIMERx_CCMR4 寄存器(OC4PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 4 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMERx_CNT 的比较，并在 OC4 端口上产生输出信号。 **若 CC4 通道配置为输入：** CCR4 包含了由上一次输入捕获 4 事件(IC4)传输的计数器值。 注：当配置为捕获模式时，该寄存器变成只读。

14.4.2.18. 刹车和死区寄存器（TIMER1_BDTR）

注: 根据锁定设置，AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0]位均可被写保护，有必要在第一次写入 TIMERx_BDTR 寄存器时对它们进行配置。

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	MOE	1'b0	RW	主输出使能 一旦刹车输入有效，该位被硬件异步清 0。根据 AOE 位的设置值，该位可以由软件清 0 或被自动置 1。它仅对配置为输出的通道有效。 0: 禁止 OC 和 OCN 输出或强制为空闲状态； 1: 如果设置了相应的使能位(TIMERx_CCER 寄存器的 CCxE、CCxNE 位)，则开启 OC 和 OCN 输出。 有关 OC/OCN 使能的细节，参见捕获/比较使能寄存器(TIMERx_CCER)。

14	AOE	1'b0	RW	自动输出使能 0: MOE 只能被软件置 1； 1: MOE 能被软件置 1 或在下一个更新事件被自动置 1(如果刹车输入无效)。 注：一旦 LOCK 级别(TIMERx_BDTR 寄存器中的 LOCK 位)设为 1，则该位不能被修改。
13	BKP	1'b0	RW	外部 IO 刹车输入极性 0: 刹车输入低电平有效 1: 刹车输入高电平有效 注：一旦 LOCK 级别(TIMERx_BDTR 寄存器中的 LOCK 位)设为 1，则该位不能被修改。
12	BKE	1'b0	RW	刹车功能使能 0: 禁止刹车功能(BRK 及 CCS 时钟失效事件)； 1：开启刹车功能(需要配置 TIM1_CR1 选择刹车信号源)。 注：当设置了 LOCK 级别 1 时(TIMERx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。
11	OSSR	1'b0	RW	运行模式下“关闭状态”选择该位用于当 MOE=1 且通道为互补输出时。没有互补输出的定时器中不存在 OSSR 位。 参考 OC/OCN 使能的详细说明(捕获/比较使能寄存器(TIMERx_CCER))。 0: 当定时器不工作时，禁止 OC/OCN 输出(OC/OCN 使能输出信号=0)； 1: 当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，首先开启 OC/OCN 并输出无效电平，然后置 OC/OCN 使能输出信号=1。 注: 一旦 LOCK 级别(TIMERx_BDTR 寄存器中的 LOCK 位)设为 2，则该位不能被修改。
10	OSSI	1'b0	RW	空闲模式下“关闭状态”选择该位用于当 MOE=0 且通道设为输出时。 参考 OC/OCN 使能的详细说明(捕获/比较使能寄存器(TIMERx_CCER))。 0: 当定时器不工作时，禁止 OC/OCN 输出(OC/OCN 使能输出信号=0)； 1: 当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，OC/OCN 首先输出其空闲电平，然后 OC/OCN 使能输出信号=1。 注：一旦 LOCK 级别(TIMERx_BDTR 寄存器中的 LOCK 位)设为 2，则该位不能被修改。
9:8	LOCK	2'b0	RW	锁定设置 该位为防止软件错误而提供写保护。 b00: 锁定关闭，寄存器无写保护； b01: 锁定级别 1，不能写入 TIMERx_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 TIMERx_CR2 寄存器的 OISx/OISxN 位； b10: 锁定级别 2，不能写入锁定级别 1 中的各

				位，也不能写入 CC 极性位(一旦相关通道通过 CCxS 位设为输出，CC 极性位是 TIMERx_CCER 寄存器的 CxP/CCNxP 位)以及 OSSR/OSSI 位； b11：锁定级别 3，不能写入锁定级别 2 中的各位，也不能写入 CC 控制位(一旦相关通道通过 CCxS 位设为输出，CC 控制位是 TIMERx_CCMRx 寄存器的 OCxM/OCxPE 位)； 注：在系统复位后，只能写一次 LOCK 位，一旦写入 TIMERx_BDTR 寄存器，则其内容冻结直至复位。
7:0	DTG	8'b0	RW	死区发生器设置 这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间： DTG[7:5]=0xx=>DT=DTG[7:0]×T dtg，Tdtg=TDTS ； DTG[7:5]=10x=>DT=(64+DTG[5:0]) × Tdtg，Tdtg=2×TDTS； DTG[7:5]=110=>DT=(32+DTG[4:0]) × Tdtg，Tdtg=8×TDTS； DTG[7:5]=111=> DT=(32+DTG[4:0]) ×Tdtg，Tdtg=16×TDTS； 例：若 TDTS=125ns(8MHZ)，可能的死区时间为： 0 到 15875ns，若步长时间为 125ns； 16us 到 31750ns，若步长时间为 250ns； 32us 到 63us，若步长时间为 1us； 64us 到 126us，若步长时间为 2us； 注：一旦 LOCK 级别(TIMERx_BDTR 寄存器中的 LOCK 位)设为 1、2 或 3，则不能修改这些位。

14.4.2.19. TIMER 同步控制寄存器（TIMER_ALLCON）

Width	Name	Reset	Property	Description
31:25	Reserved	-	-	-
24	TIM456_ETRBIN_SEL	1'b0	RW	TIM4/5/6 外部 IO 刹车信号选择 0: TIM4/5/6 的外部 IO 刹车信号独立 1: TIM4/5/6 共用外部 IO 刹车信号 当配置为 1 时，TIM4/5/6 使用的外部 IO 刹车信号是各自外部 IO 刹车信号或的关系: TIM456_ETR= (TIM4_ETR | TIM5_ETR | TIM6_ETR)
23:22	Reserved	-	-	-
21:16	TIMERx_STOP	6'h0	WO	TIM1~TIM6 停止计数 0: 无效 1: 计数器停止（CEN 清零）
15:14	Reserved	-	-	-
13:8	TIMERx_CLR	6'h0	WO	TIM1~TIM6 计数器清零 0: 无效

				1: 计数器清零
7:6	Reserved	-	-	-
5:0	TIMERx_KST	6'h0	WO	TIM1~TIM6 计数器使能 0: 无效 1: 计数器使能

15. 通用定时器（TIMER2/3）

15.1. 模块介绍

通用定时器是一个通过可编程预分频器驱动的 16 位自动装载计数器构成。

它适用于多种场合，包括测量输入信号的脉冲长度(输入捕获)或者产生输出波形(输出比较和 PWM)。

使用定时器预分频器和 CMU 时钟控制器预分频器，脉冲长度和波形周期可以在几个微秒到几个毫秒间调整。

每个定时器都是完全独立的，没有互相共享任何资源。它们可以一起同步操作。

15.2. 功能特点

通用 TIMERx (TIMER2/3)定时器功能包括：

- 16 位向上、向下、向上/下自动装载计数器
- 16 位可编程（可以实时修改）预分频器，计数器时钟频率的分频系数为 1～65535 之间的任意数值
- 多达 4 个独立通道：
  - 输入捕获
  - 输出比较
  - PWM 生成（边缘或中间对齐模式）
  - 单脉冲模式输出
- 使用外部信号控制定时器和定时器互联的同步电路
- 如下事件发生时产生中断/DMA：
  - 更新：计数器向上溢出/向下溢出，计数器初始化(通过软件或者内部/外部触发)
  - 触发事件(计数器启动、停止、初始化或者由内部/外部触发计数)
  - 输入捕获
  - 输出比较
- 支持针对定位的增量(正交)编码器和霍尔传感器电路
- 触发输入作为外部时钟或者按周期的电流管理

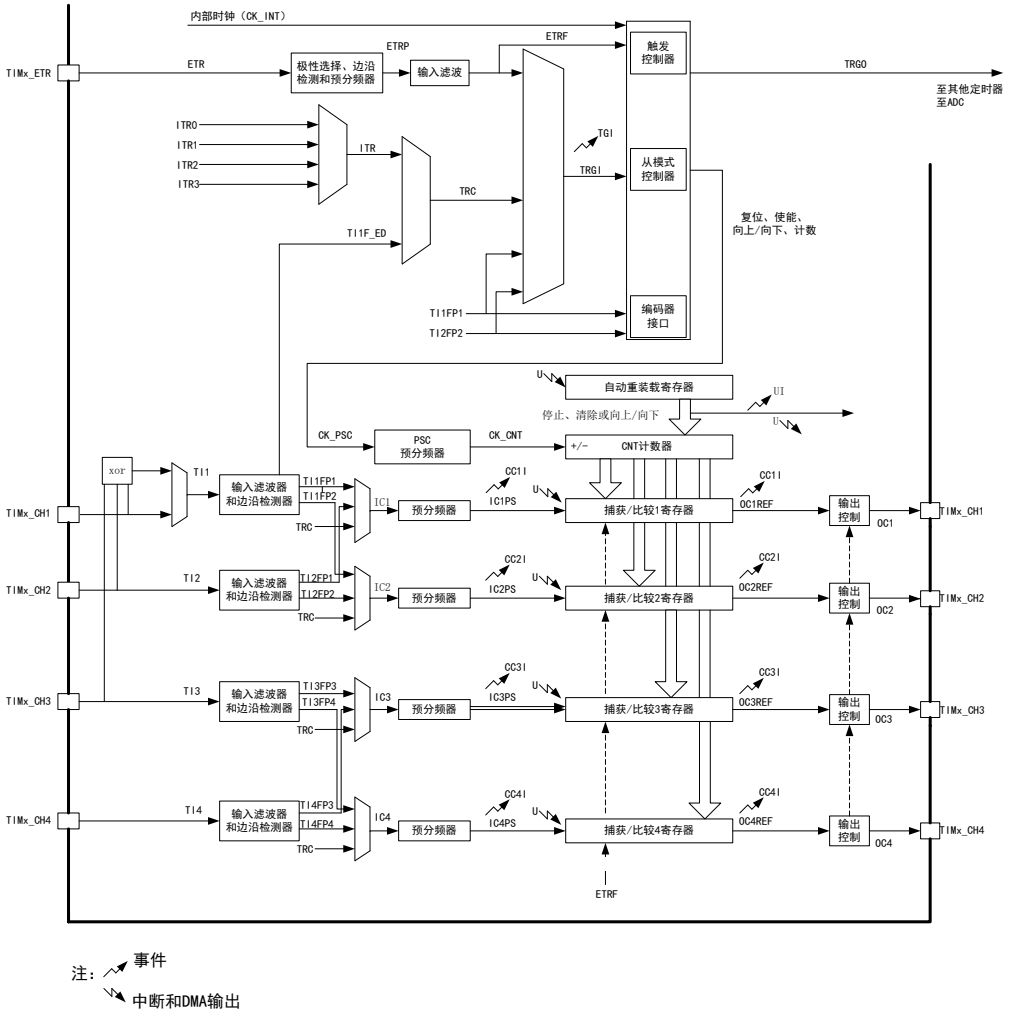


图 15-1

15.3. 功能说明

15.3.1. 时基单元

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

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

时基单元包含：

- 计数器寄存器(TIMERx_CNT)
- 预分频器寄存器 (TIMERx_PSC)
- 自动装载寄存器 (TIMERx_ARR)
- 重复次数寄存器 (TIMERx_RCR)

自动装载寄存器是预先装载的，写或读自动重装载寄存器将访问预装载寄存器。根据在 TIMERx_CR1 寄存器中的自动装载预装载使能位(ARPE)的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件(向下计数时的下溢条件)并当 TIMERx_CR1 寄存器中的 UDIS 位等于 0 时，产生更新事件。更新事件也可以由软件产生。随后会详细描述每一种配置下更新事件的产生。

计数器由预分频器的时钟输出 CK_CNT 驱动，仅当设置了计数器 TIMERx_CR1 寄存器中的计数器使能位(CEN)时，CK_CNT 才有效。(更多有关使能计数器的细节，请参见控制器的从模式描述)。

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

预分频器描述

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是基于一个(在 TIMERx_PSC 寄存器中的)16 位寄存器控制的 16 位计数器。因为这个控制寄存器带有缓冲器，它能够在运行时被改变，

新的预分频器的参数在下一次更新事件到来时被采用。

以下两图给出了在预分频器运行时，更改计数器参数的例子。

当预分频器的参数从 1 变到 2 时，计数器的时序图：

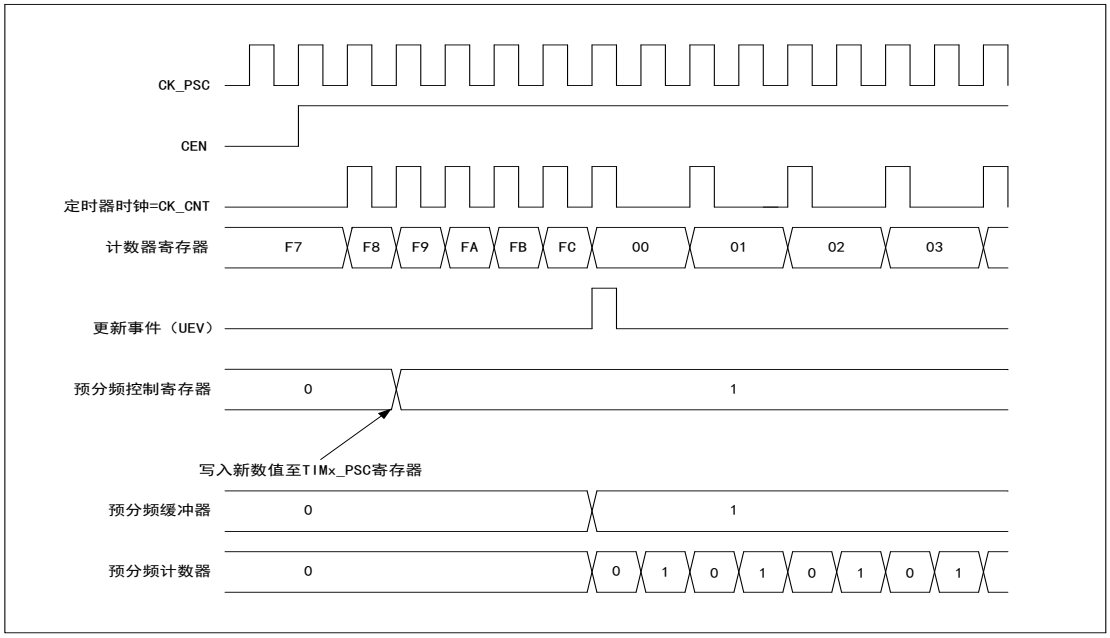


图 15-2

当预分频器的参数从 1 变到 4 时，计数器的时序图：

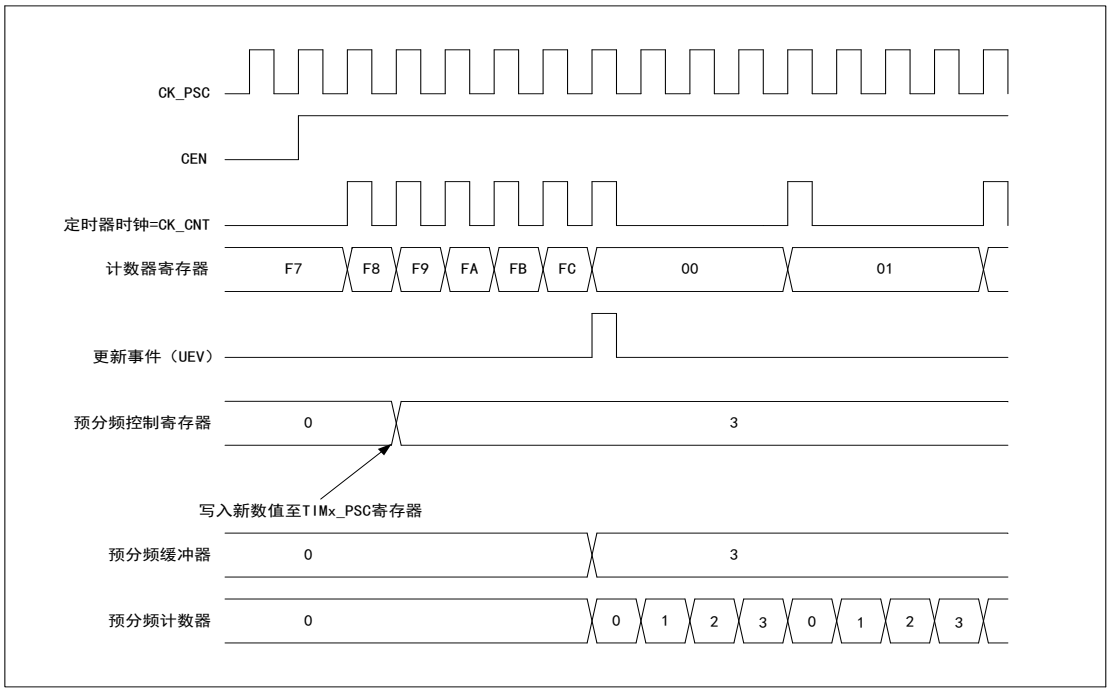


图 15-3

15.3.2. 计数器模式

向上计数模式

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

每次计数器溢出时可以产生更新事件，在 TIMERx_EGR 寄存器中(通过软件方式或者使用从模式控制器)设置 UG 位也同样可以产生一个更新事件。

设置 TIMERx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清‘0’之前，将不产生更新事件。但是在应该产生更新事件时，

计数器仍会被清‘0’，同时预分频器的计数也被清 0(但预分频器的数值不变)。此外，如果设置了 TIMERx_CR1 寄存器中的 URS 位(选择更新请求)，设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志(即不产生中断或 DMA 请求)。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

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

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

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

计数器时序图，预分频参数为 1

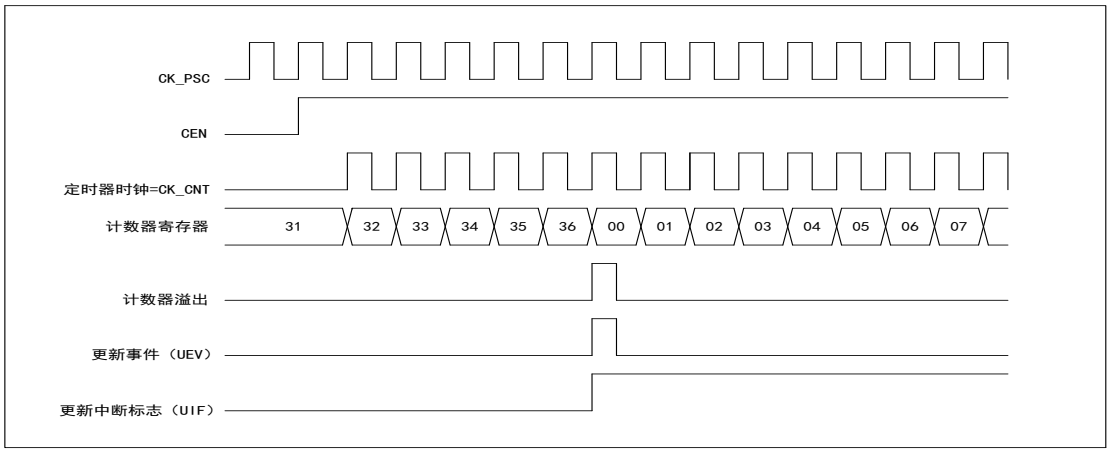


图 15-4

计数器时序图，预分频参数为 2

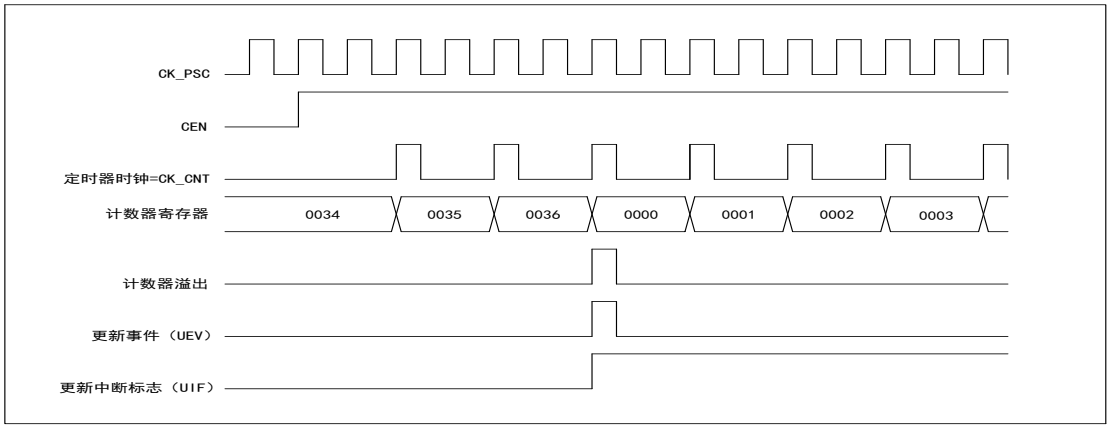


图 15-5

计数器时序图，预分频参数为 4

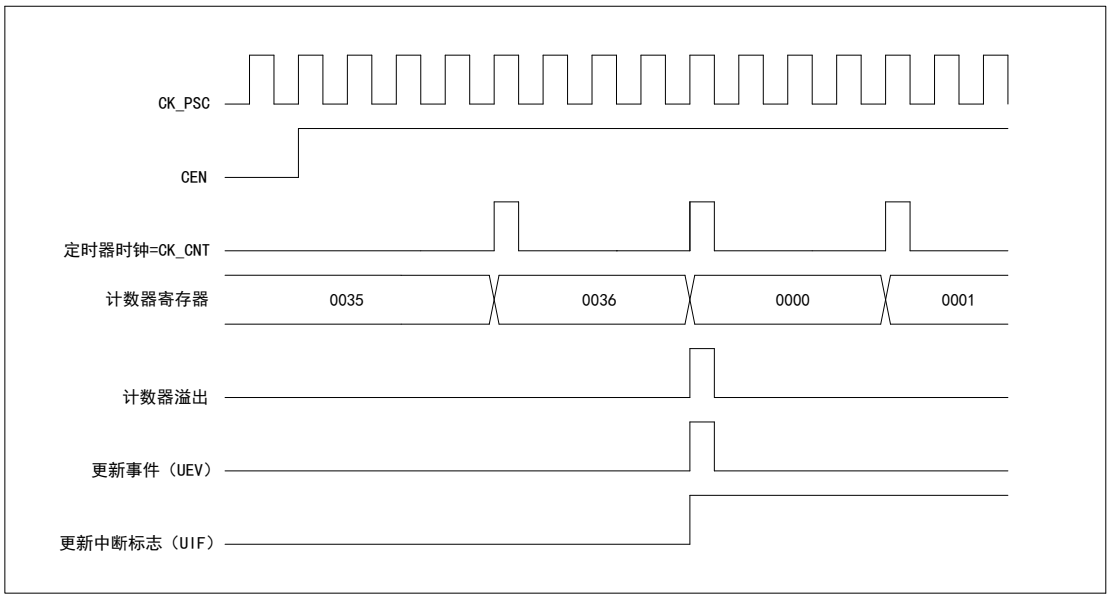


图 15-6

计数器时序图，预分频参数为 N

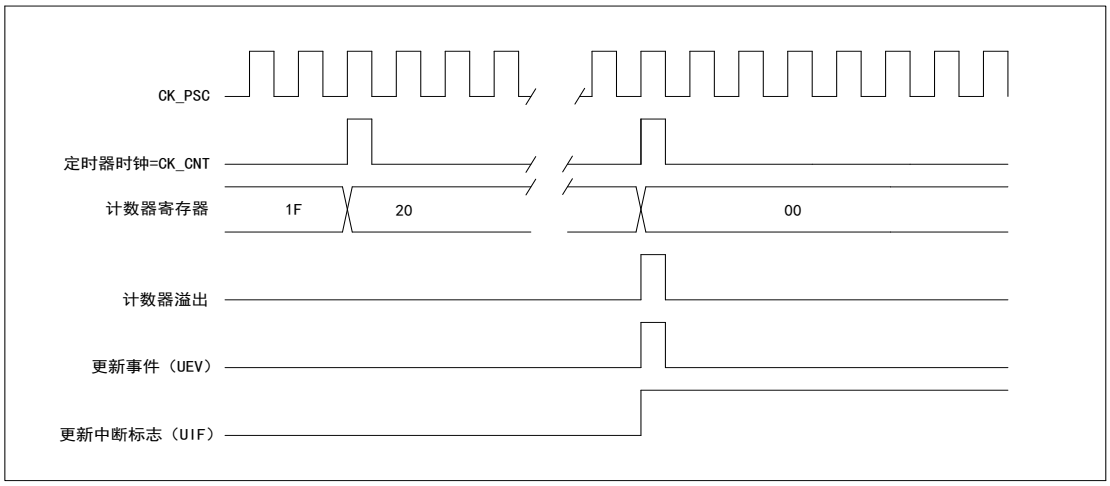


图 15-7

计数器时序图，当 ARPE=0 时的更新事件(TIMERx_ARR 没有预装入)

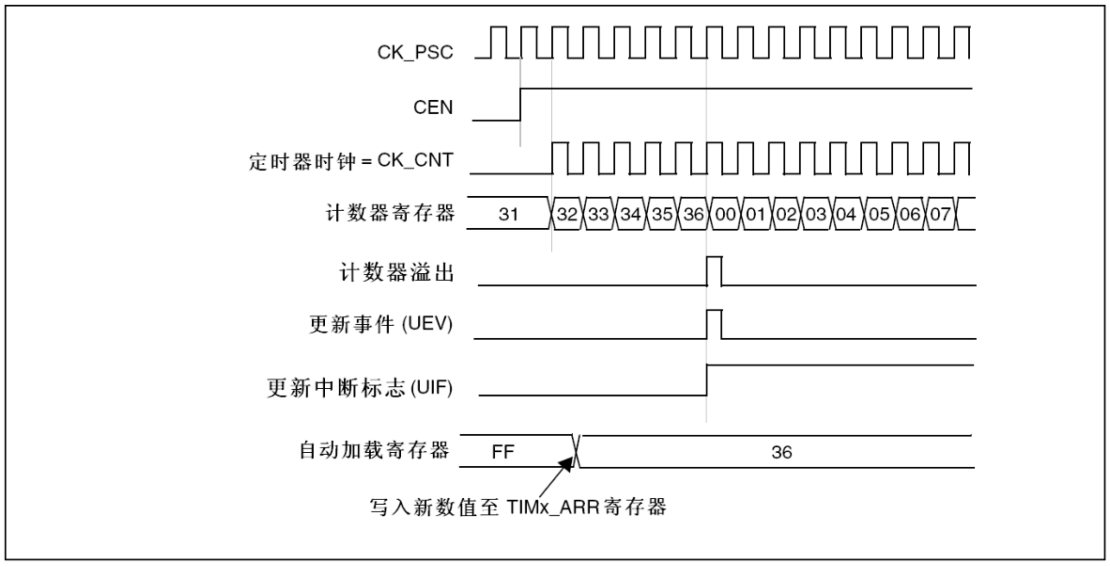


图 15-8

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

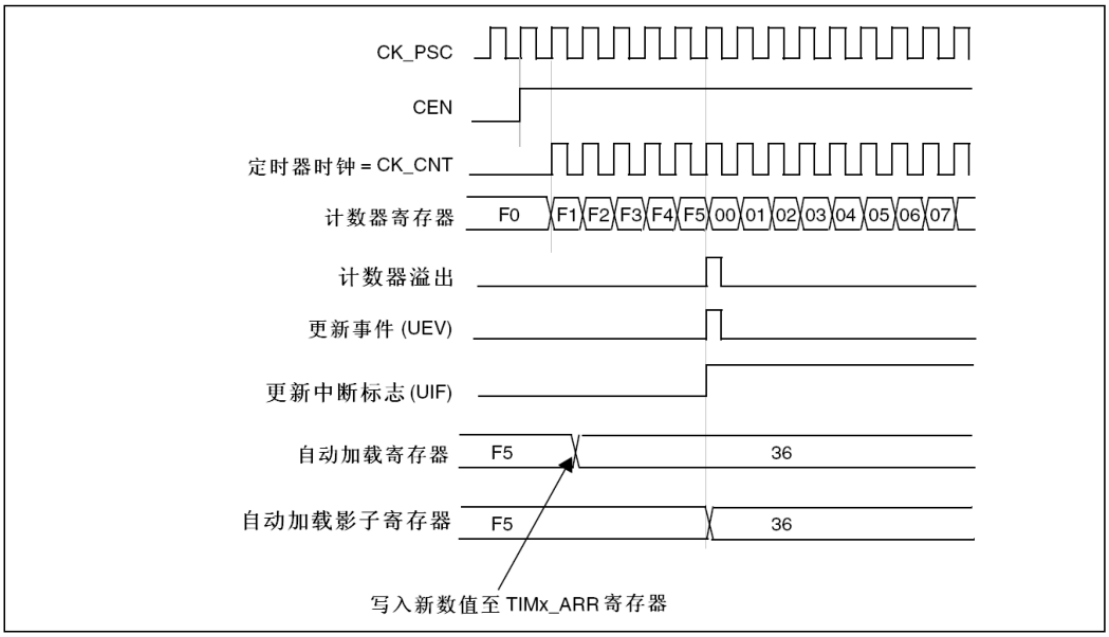


图 15-9

向下计数模式

在向下模式中，计数器从自动装入的值(TIMERx_ARR 计数器的值)开始向下计数到 0，然后从自动装入的值重新开始并且产生一个计数器向下溢出事件。

每次计数器溢出时可以产生更新事件，在 TIMERx_EGR 寄存制器)设置 UG 位，也同样可以产生一个更新事件。

设置 TIMERx_CR1 寄存器的 UDIS 位可以禁止 UEV 事件。这样可以避免向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，并且预分频器的计数器重新从 0 开始(但预分频系数不变)。

此外，如果设置了 TIMERx_CR1 寄存器中的 URS 位(选择更新请求) ，设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志(因此不产生中断和 DMA 请求),这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

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

- 预分频器的缓冲区被置入预装载寄存器的值(TIMERx_PSC 寄存器的内容)。
- 当前的自动加载寄存器被更新为预装载值(TIMERx_ARR 寄存器中的内容)。

注：自动装载在计数器重载入之前被更新，因此下一个周期将是预期的值。

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

计数器时序图，预分频参数为 1

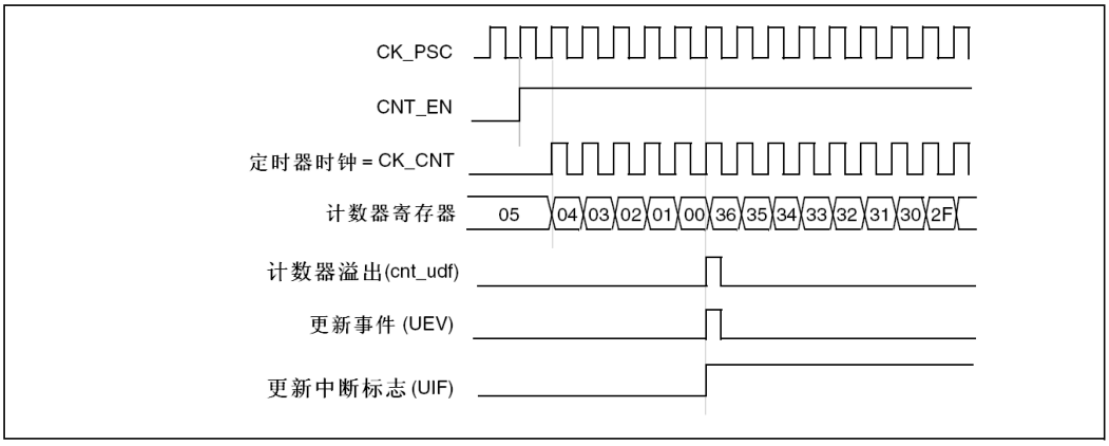


图 15-10

计数器时序图，预分频参数为 2

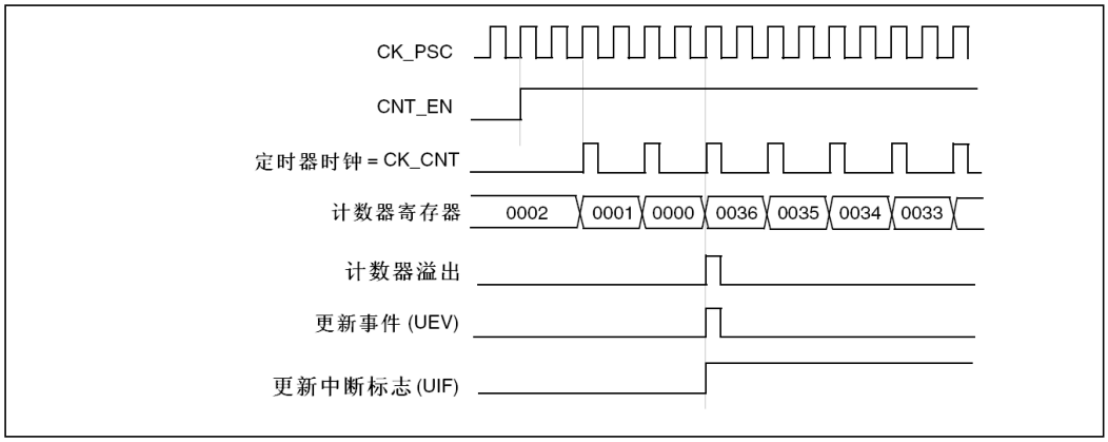


图 15-11

计数器时序图，预分频参数为 4

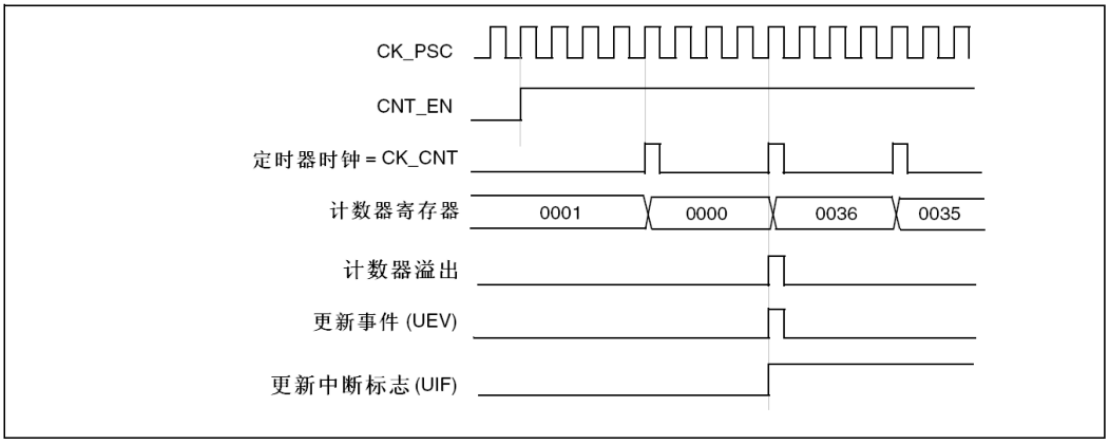


图 15-12

计数器时序图，预分频参数为 N

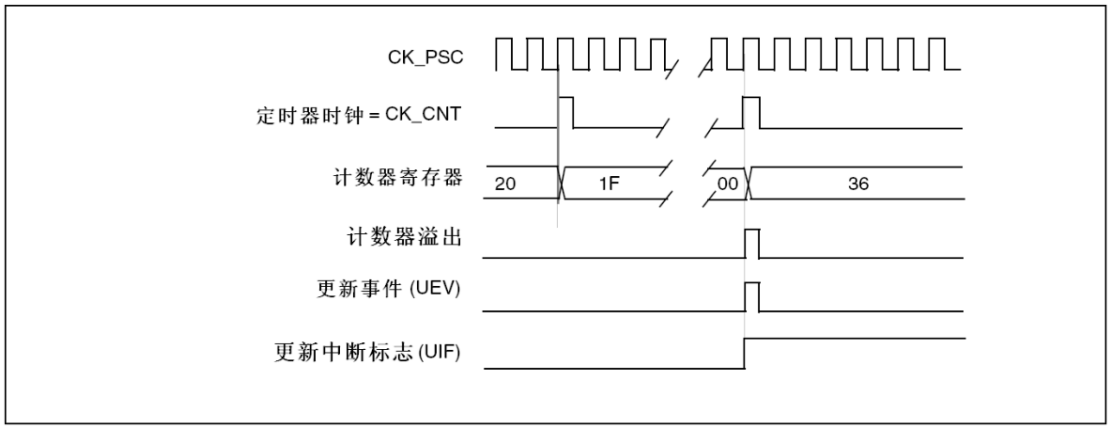


图 15-13

中央对齐模式（向上/向下计数）

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

在此模式下，不能写入 TIMERx_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。

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

设置 TIMERx_CR1 寄存器中的 UDIS 位可以禁止 UEV 事件。这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会根据当前自动

重加载的值，继续向上或向下计数。

此外，如果设置了 TIMERx_CR1 寄存器中的 URS 位(选择更新请求) ，设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志(因此不产生中断和 DMA 请求),这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

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

- 预分频器的缓冲区被置入预装载寄存器的值(TIMERx_PSC 寄存器的内容)。
- 当前的自动加载寄存器被更新为预装载值(TIMERx_ARR 寄存器中的内容)。

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

以下是一些计数器在不同时钟频率下的操作的例子：

计数器时序图，预分频参数为 1，TIMERx_ARR=0x6

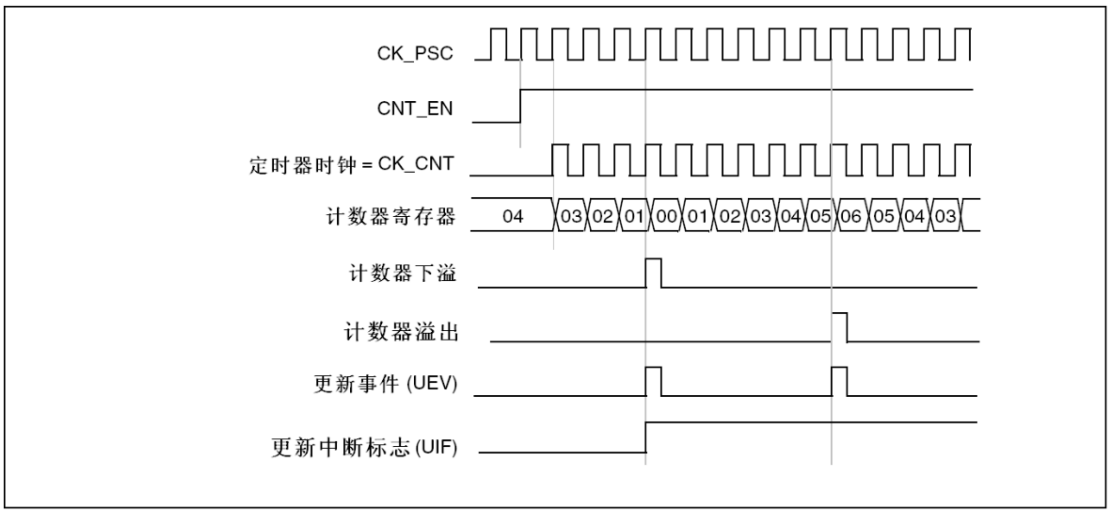


图 15-14

计数器时序图，预分频参数为 2

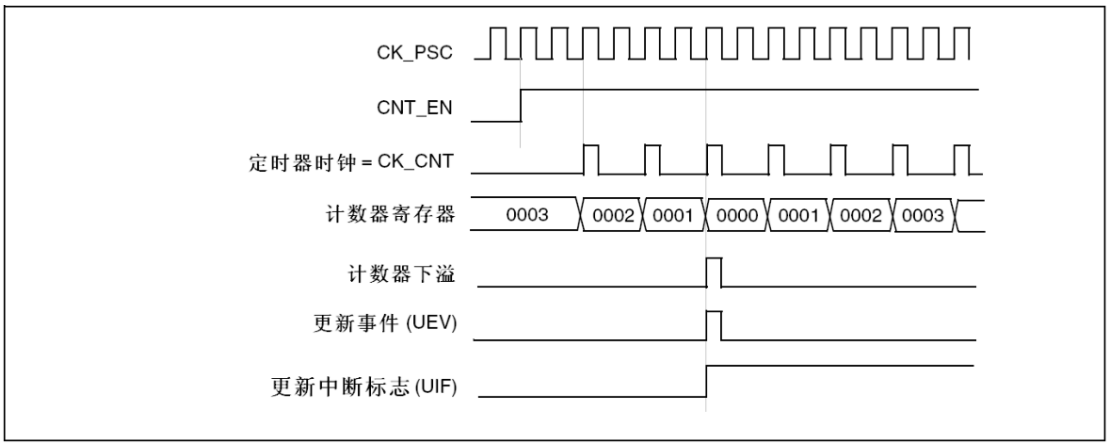


图 15-15

计数器时序图，预分频参数为 4，TIMERx_ARR=0x36

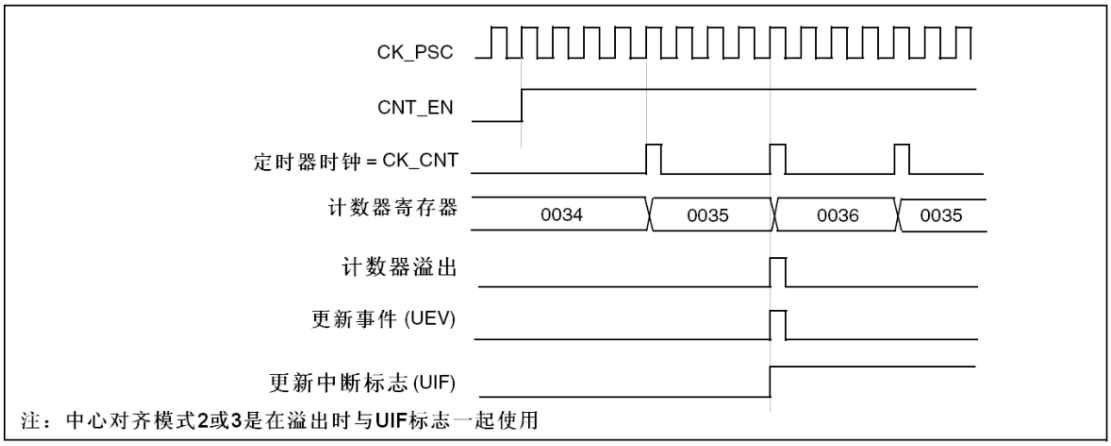


图 15-16

计数器时序图，预分频参数为 N

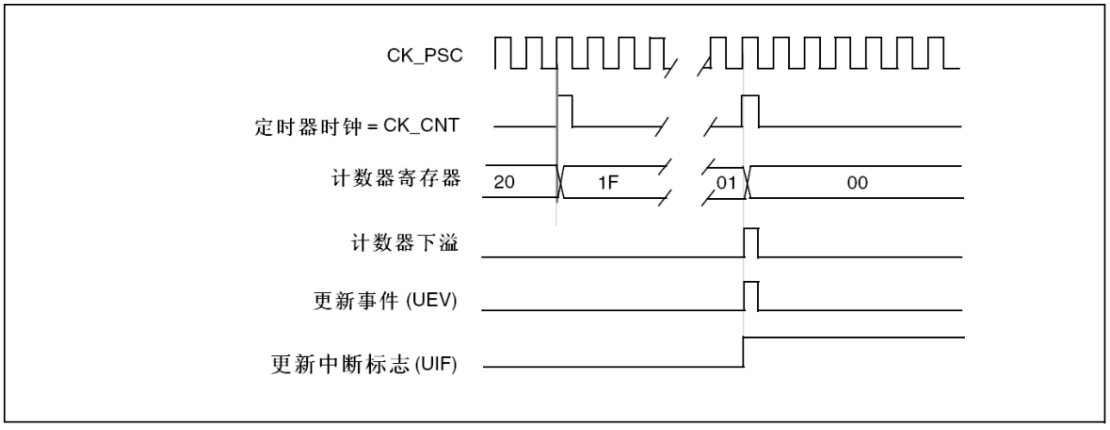


图 15-17

计数器时序图，ARPE=1 时的更新事件(计数器下溢)

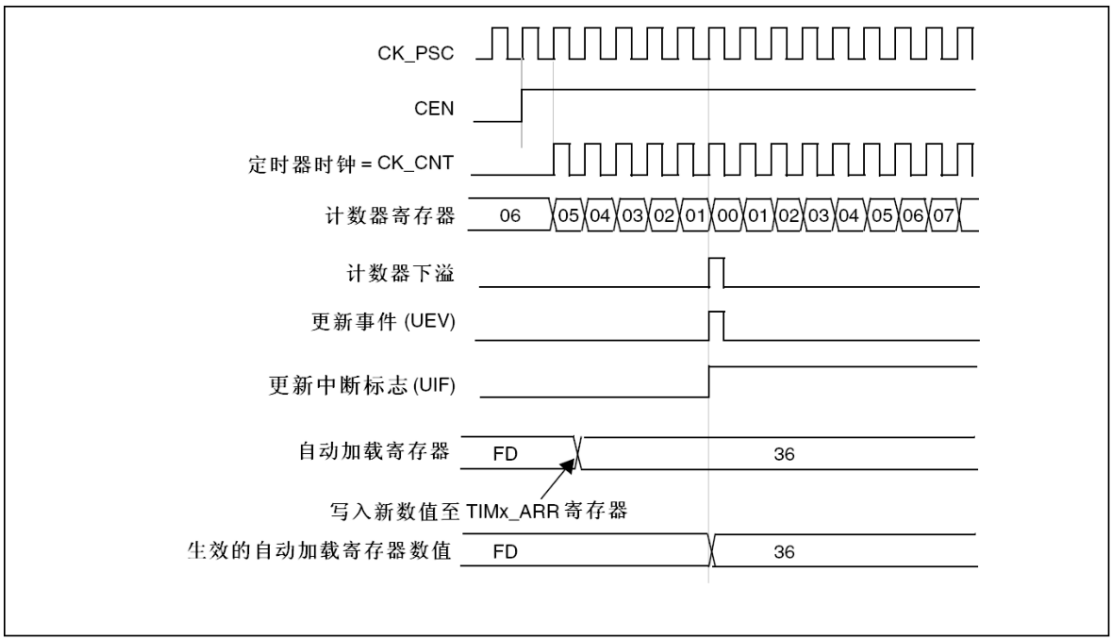


图 15-18

计数器时序图，ARPE=1 时的更新事件(计数器溢出)

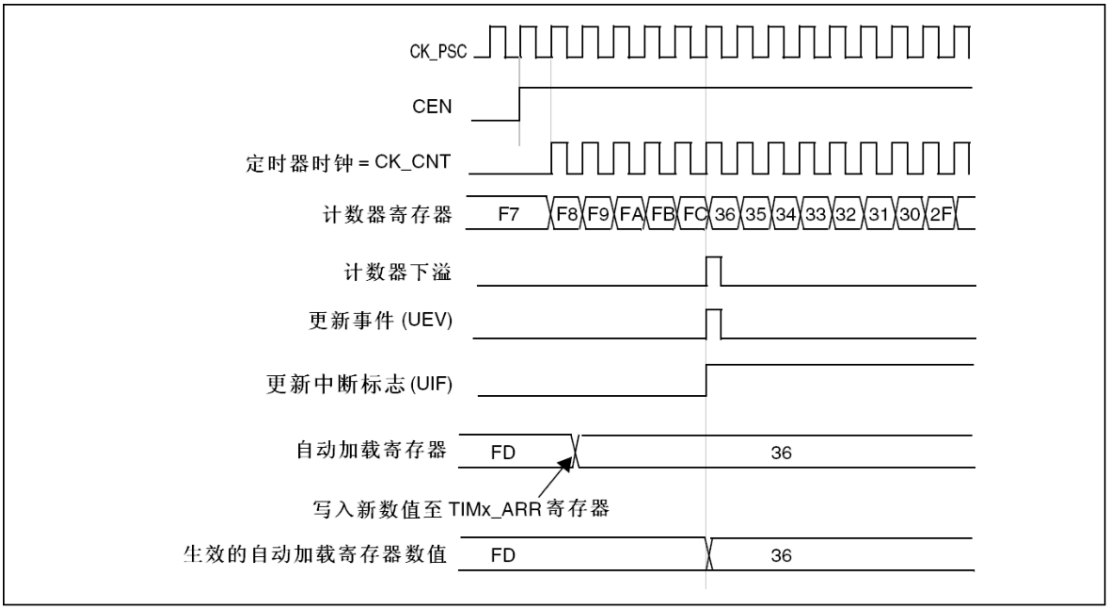


图 15-19

15.3.3. 时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟（CK_INT）
- 外部时钟模式 1：外部输入引脚
- 外部时钟模式 2：外部触发输入 ETR
- 内部触发输入(ITRx)：使用一个定时器作为另一个定时器的预分频器。如可以配置一个定时器 Timer1 而作为另一个定时器 Timer2 的预分频器。详见下一章。

内部时钟源 内部时钟源(CK_INT)

如果禁止了从模式控制器(SMS=000)，则 CEN、DIR(TIMERx_CR1 寄存器)和 UG 位(TIMERx_EGR 寄存器)是事实上的控制位，并且只能被软件修改(UG 位仍被自动清除)。只要 CEN 位被写成“1”,预分频器的时钟就由内部时钟 CK_INT 提供。

下图显示控制电路和向上计数器在一般模式下，不带预分频器时的操作。

一般模式下的控制电路，预分频参数为 1

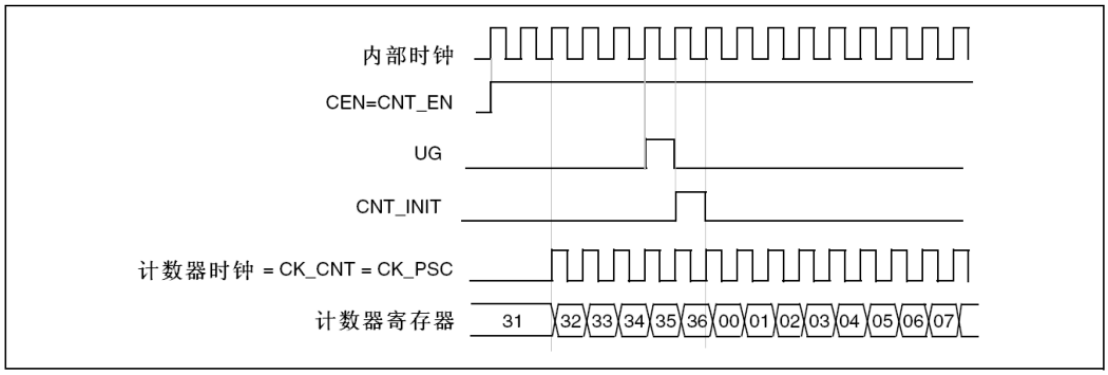


图 15-20

外部时钟源模式 1

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

TI2 外部时钟连接例子

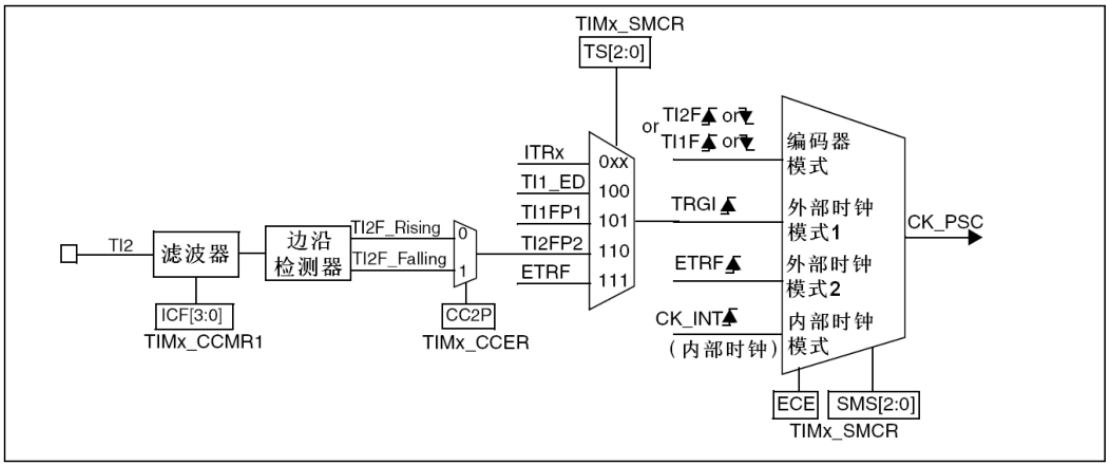


图 15-21

例如，要配置向上计数器在 T12 输入端的上升沿计数，使用下列步骤：

配置 TIMERx_CCMR1 寄存器 CC2S=01，配置通道 2 检测 TI2 输入的上升沿

配置 TIMERx_CCMR1 寄存器的 IC2F[3:0]，选择输入滤波器带宽(如果不需要滤波器，保持 IC2F=0000)

配置 TIMERx_CCER 寄存器的 CC2P=0，选定上升沿极性

配置 TIMERx_SMCR 寄存器的 SMS=111，选择定时器外部时钟模式 1

配置 TIMERx_SMCR 寄存器中的 TS=110，选定 TI2 作为触发输入源

设置 TIMERx_CR1 寄存器的 CEN=1，启动计数器

注： 捕获预分频器不用作触发，所以不需要对它进行配置

当上升沿出现在 TI2，计数器计数一次，且 TIF 标志被设置。

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

外部时钟模式 1 下的控制电路

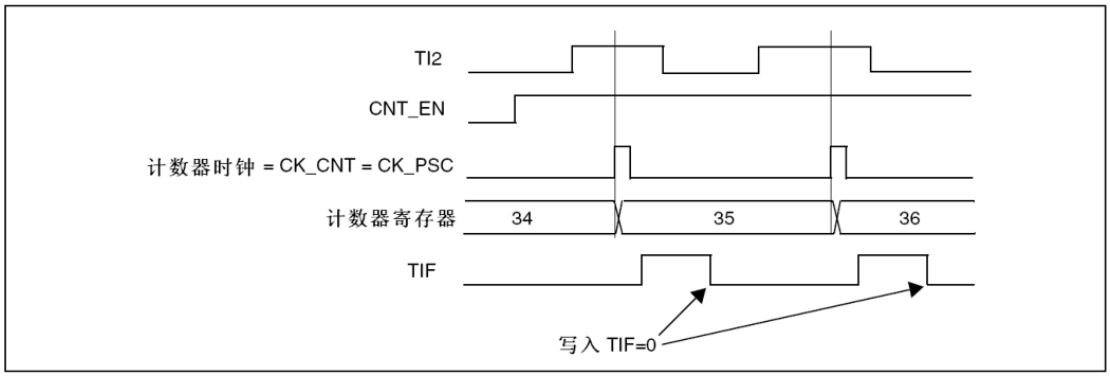


图 15-22

外部时钟源模式 2

选定此模式的方法为：令 TIMERx_SMCR 寄存器中的 ECE=1，

计数器能够在外部触发 ETR 的每一个上升沿或下降沿计数。

下图是外部触发输入的框图

外部触发输入框图

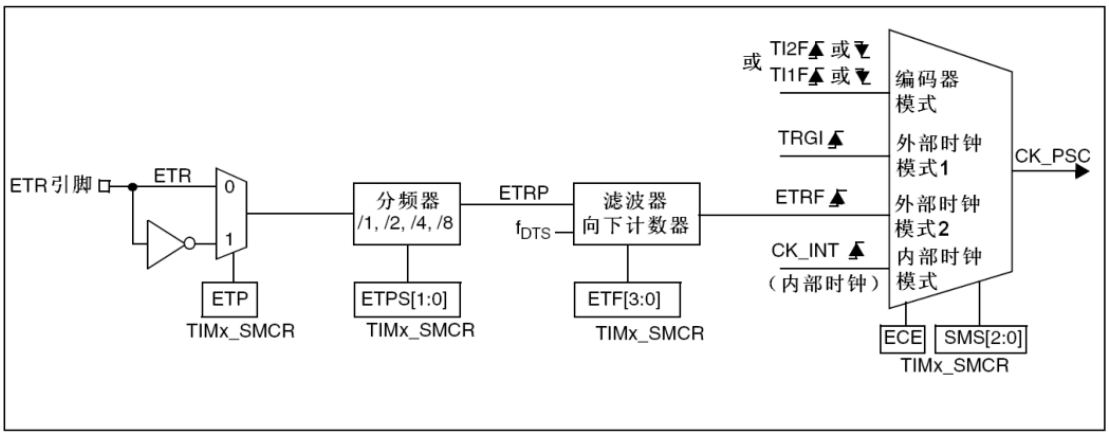


图 15-23

例如，要配置在 ETR 下每 2 个上升沿计数一次的向上计数器，使用下列步骤：
本例中不需要滤波器，置 TIMERx_SMCR 寄存器中的 ETF[3:0]=0000
设置预分频器，置 TIMERx_SMCR 寄存器中的 ETPS[1:0]=01
选择 ETR 的上升沿检测，置 TIMERx_SMCR 寄存器中的 ETP=0
开启外部时钟模式 2，写 TIMERx_SMCR 寄存器中的 ECE=1
启动计数器，写 TIMERx_CR1 寄存器中的 CEN=1
计数器在每 2 个 ETR 上升沿计数一次。
在 ETR 的上升沿和计数器实际时钟之间的延时取决于在 ETRP 信号端的重新同步电路。
外部时钟模式 2 下的控制电路

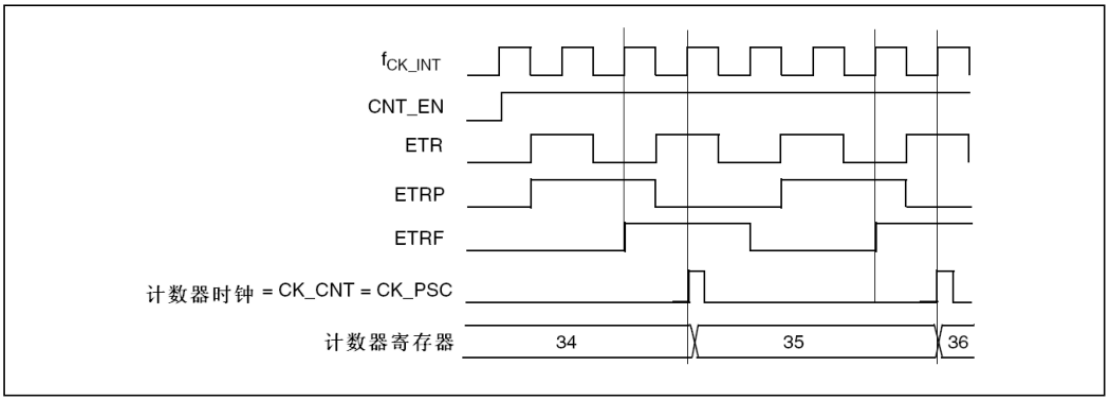


图 15-24

15.3.4. 捕获/比较通道

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

以下三图是一个捕获/比较通道概览。

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

捕获/比较通道(如：通道 1 输入部分)

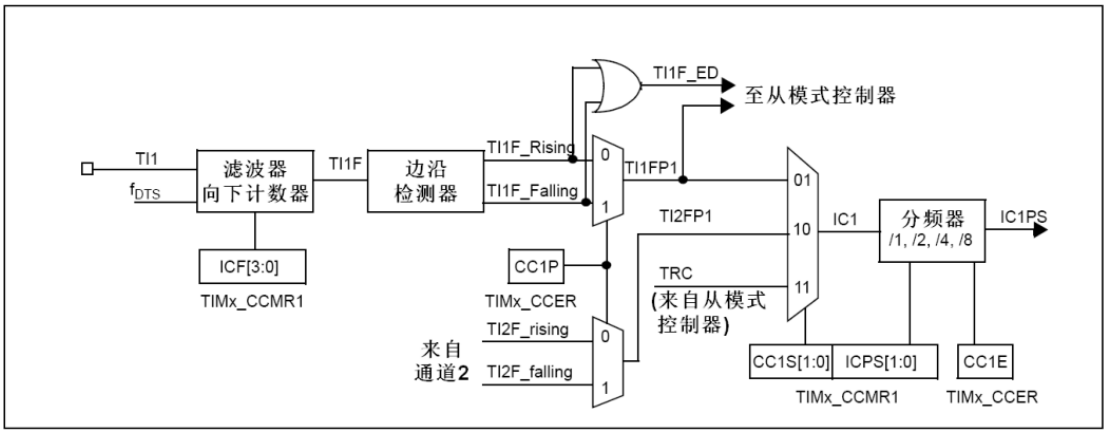


图 15-25

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

捕获/比较通道 1 的主电路

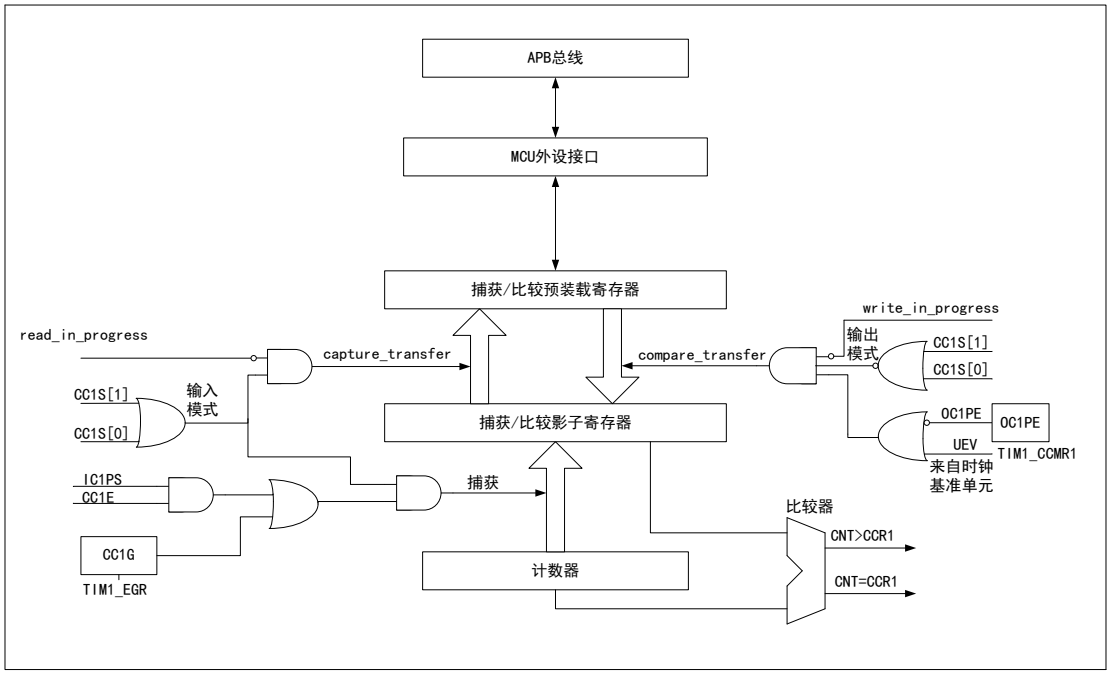


图 15-26

捕获/比较通道的输出部分(通道 1)

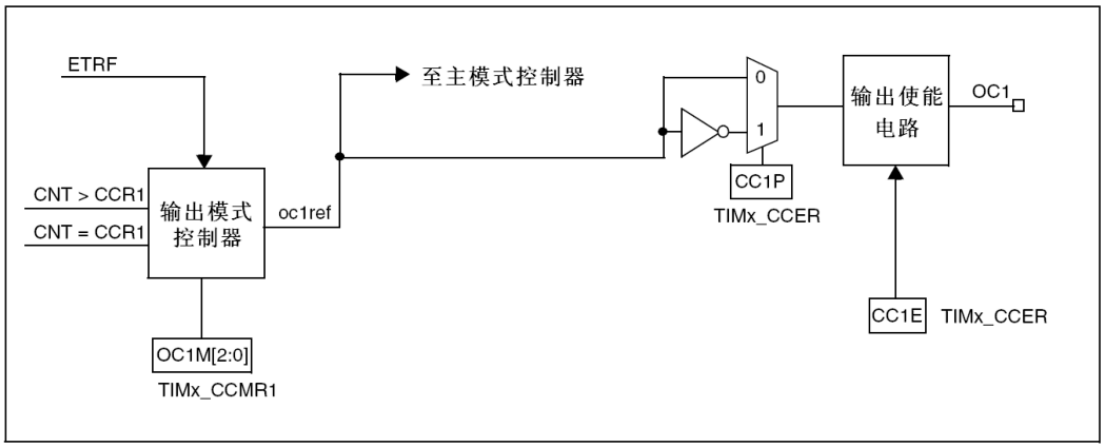


图 15-27

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

在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。

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

15.3.5. 输入捕获模式

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

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIMERx_CCR1 寄存器中，步骤如下：

- 选择有效输入端：TIMERx_CCR1 必须连接到 TI1 输入，所以写入 TIMERx_CCMR1 寄存器中的 CC1S=01，只要 CC1S 不为“00”，通道被配置为输入，并且 TIMERx_CCR1 寄存器变为只读。
- 根据输入信号的特点，配置输入滤波器为所需的带宽(即输入为 TIx 时，输入滤波器控制位是 TIMERx_CCMRx 寄存器中的 ICxF 位)。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期；因此我们可以(以 f DTS 频率)连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIMERx_CCMR1 寄存器中写入 IC1F=0011。
- 选择 TI1 通道的有效转换边沿，在 TIMERx_CCER 寄存器中写入 CC1P=0(上升沿)。
- 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止(写 TIMERx_CCMR1 寄存器的 IC1PS=00)。
- 设置 TIMERx_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
- 如果需要，通过设置 TIMERx_DIER 寄存器中的 CC1IE 位允许相关中断请求，通过设置 TIMERx_DIER 寄存器中的 CC1DE 位允许 DMA 请求。

当发生一个输入捕获时：

- 产生有效的电平转换时，计数器的值被传送到 TIMERx_CCR1 寄存器。
- CC1IF 标志被设置(中断标志)。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置 1。
- 如设置了 CC1IE 位，则会产生一个中断。
- 如设置了 CC1DE 位，则还会产生一个 DMA 请求。

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

注： 设置 TIMERx_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断和 / 或 DMA 请求。

15.3.6. PWM 输入模式

该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

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

例如，你需要测量输入到 TI1 上的 PWM 信号的长度(TIMERx_CCR1 寄存器)和占空比(TIMERx_CCR2 寄存器)，具体步骤如下(取决于 CK_INT 的频率和预分频器的值)

- 选择 TIMERx_CCR1 的有效输入：置 TIMERx_CCMR1 寄存器的 CC1S=01(选中 TI1)。
- 选择 TI1FP1 的有效极性(用来捕获数据到 TIMERx_CCR1 中和清除计数器)：置 CC1P=0(上升沿有效)。
- 选择 TIMERx_CCR2 的有效输入：置 TIMERx_CCMR1 寄存器的 CC2S=10(选中 TI1)。
- 选择 TI1FP2 的有效极性(捕获数据到 TIMERx_CCR2)：置 CC2P=1(下降沿有效)。
- 选择有效的触发输入信号：置 TIMERx_SMCR 寄存器中的 TS=101(选择 TI1FP1)。
- 配置从模式控制器为复位模式：置 TIMERx_SMCR 中的 SMS=100。
- 使能捕获：置 TIMERx_CCER 寄存器中 CC1E=1 且 CC2E=1。

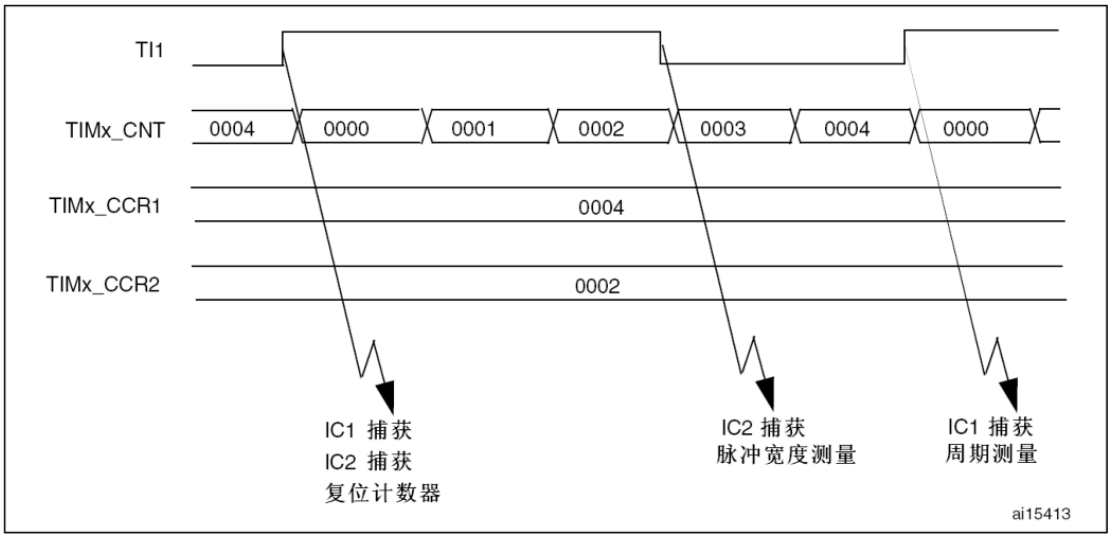


图 15-28 PWM 输入模式时序

因为只有 TI1FP1 和 TI2FP2 连到了从模式控制器，所以 PWM 输入模式只能使用 TIMERx_CH1 /TIMERx_CH2 信号。

15.3.7. 强制输出模式

在输出模式(TIMERx_CCMRx 寄存器中 CCxS=00)下，输出比较信号(OCxREF 和相应的 OCx/OCxN)能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。

置 TIMERx_CCMRx 寄存器中相应的 OCxM=101，即可强置输出比较信号(OCxREF/OCx)为有效状态。这样 OCxREF 被强置为高电平(OCxREF 始终为高电平有效)，同时 OCx 得到 CCxP 极性相反的信号。

例如：CCxP=0(OCx 高电平有效)，则 OCx 被强置为高电平。

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

该模式下，在 TIMERx_CCRx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断和 DMA 请求。这将会在下面的输出比较模式一节中介绍。

15.3.8. 输出比较模式

此项功能是用来控制一个输出波形，或者指示一段给定的的时间已经到时。

当计数器与捕获/比较寄存器的内容相同时，输出比较功能做如下操作：

- 将输出比较模式(TIMERx_CCMRx 寄存器中的 OCxM 位)和输出极性(TIMERx_CCER 寄存器中的 CCxP 位)定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平(OCxM=000)、被设置成有效电平(OCxM=001)、被设置成无效电平(OCxM=010)或进行翻转(OCxM=011)。
- 设置中断状态寄存器中的标志位(TIMERx_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽(TIMERx_DIER 寄存器中的 CCxIE 位)，则产生一个中断。
- 若设置了相应的使能位(TIMERx_DIER 寄存器中的 CCxDE 位，TIMERx_CR2 寄存器中的 CCDS 位选择 DMA 请求功能)，则产生一个 DMA 请求。

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

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

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

输出比较模式的配置步骤：

1. 选择计数器时钟(内部，外部，预分频器)。
2. 将相应的数据写入 TIMERx_ARR 和 TIMERx_CCRx 寄存器中。
3. 如果要产生一个中断请求，设置 CCxIE 位。
4. 选择输出模式，例如：
   - 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚，设置 OCxM=011
   - 置 OCxPE = 0 禁用预装载寄存器

– 置 CCxP = 0 选择极性为高电平有效
– 置 CCxE = 1 使能输出

5. 设置 TIMERx_CR1 寄存器的 CEN 位启动计数器

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

输出比较模式，翻转 OC1

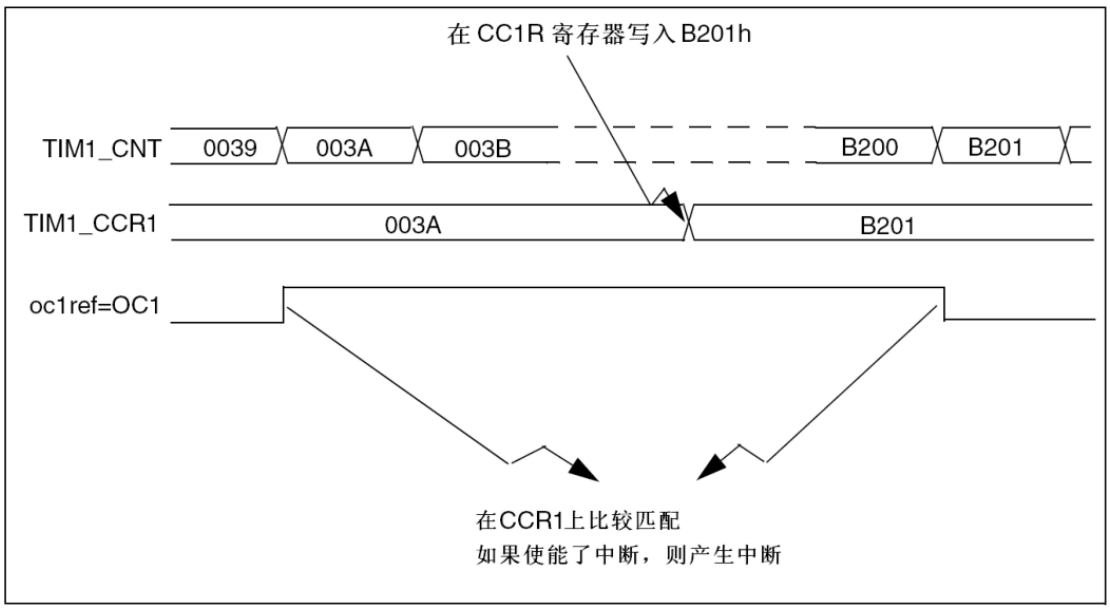


图 15-29

15.3.9. PWM 模式

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

在 TIMERx_CCMRx 寄存器中的 OCxM 位写入“110”(PWM 模式 1)或“111”(PWM 模式 2)，能够独立地设置每个 OCx 输出通道产生一路 PWM。必须通过设置 TIMERx_CCMRx 寄存器的 OCxPE 位使能相应的预装载寄存器，最后还要设置 TIMERx_CR1 寄存器的 ARPE 位，(在向上计数或中心对称模式中)使能自动重装载的预装载寄存器。

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

OCx 的极性可以通过软件在 TIMERx_CCER 寄存器中的 CCxP 位设置，它可以设置为高电平有效或低电平有效。OCx 的输出使能通过(TIMERx_CCER 和 TIMERx_BDTR 寄存器中)CCxE、CCxNE、MOE、OSSI 和 OSSR 位的组合控制。详见 TIMERx_CCER 寄存器的描述 。

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

然而为了与 OCREF_CLR 的功能(在下一个 PWM 周期之前，ETR 信号上的一个外部事件能够清除 OCxREF)一致，OCxREF 信号只能在下述条件下产生：

- 当比较的结果改变，或
- 当输出比较模式(TIMERx_CCMRx 寄存器中的 OCxM 位)从“冻结”(无比较，OCxM=‘000’)切换到某个 PWM 模式(OCxM=‘110’或‘111’)。

这样在运行中可以通过软件强置 PWM 输出。

根据 TIMERx_CR1 寄存器中 CMS 位的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

PWM 边沿对齐模式

- 向上计数的配置

当 TIMERx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。参看下面是一个 PWM 模式 1 的例子。

当 TIMERx_CNT<TIMERx_CCRx 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIMERx_CCRx 中的比较值大于自动重装载值(TIMERx_ARR)，则 OCxREF 保持为“1”。如果比较值为 0，则 OCxREF 保持为“0”。下图为 TIMERx_ARR=8 时边沿对齐的 PWM 波形实例。

边沿对齐的 PWM 波形(ARR=8)

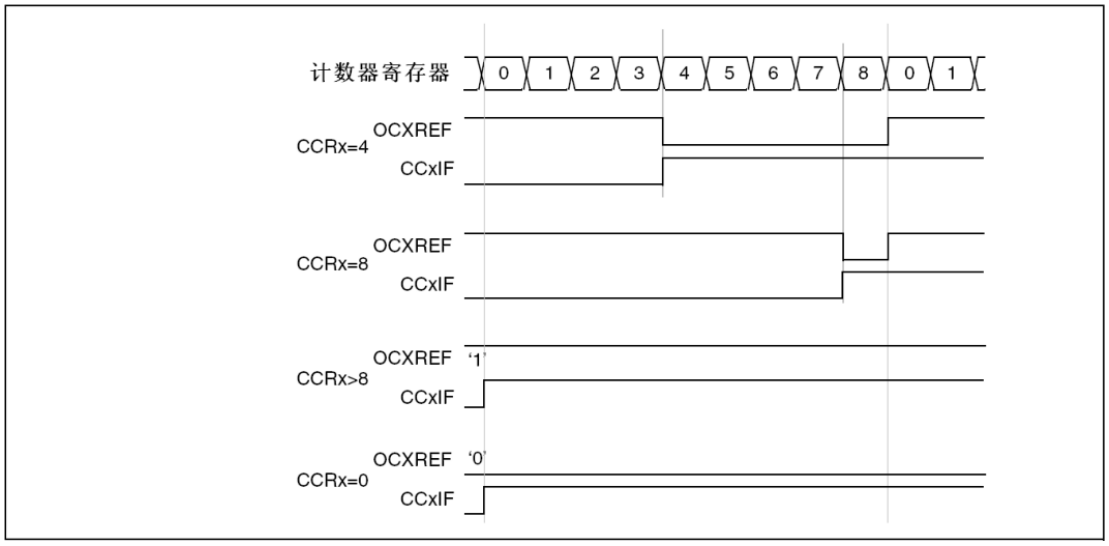


图 15-30

- 向下计数的配置

当 TIMERx_CR1 寄存器的 DIR 位为高时执行向下计数。参看在 PWM 模式 1，当 TIMERx_CNT>TIMERx_CCRx 时参考信号 OCxREF 为低，否则为高。如果 TIMERx_CCRx 中的比较值大于 TIMERx_ARR 中的自动重装载值，则 OCxREF 保持为’1’。该模式下不能产生 0%的 PWM 波形。

PWM 中央对齐模式

当 TIMERx_CR1 寄存器中的 CMS 位不为“00”时为中央对齐模式(所有其他的配置对 OCxREF/OCx 信号都有相同的作用)。根据不同的 CMS 位设置，比较标志可以在计数器向上计数时被置 1、在计数器向下计数时被置 1、或在计数器向上和向下计数时被置 1。TIMERx_CR1 寄存器中的计数方向位(DIR)由硬件更新，不要用软件修改它。下图给出了一些中央对齐的 PWM 波形的例子：

- TIMERx_ARR=8
- PWM 模式 1
- TIMERx_CR1 寄存器的 CMS=01，在中央对齐模式 1 下，当计数器向下计数时设置比较标志。中央对齐的 PWM 波形(APR=8)

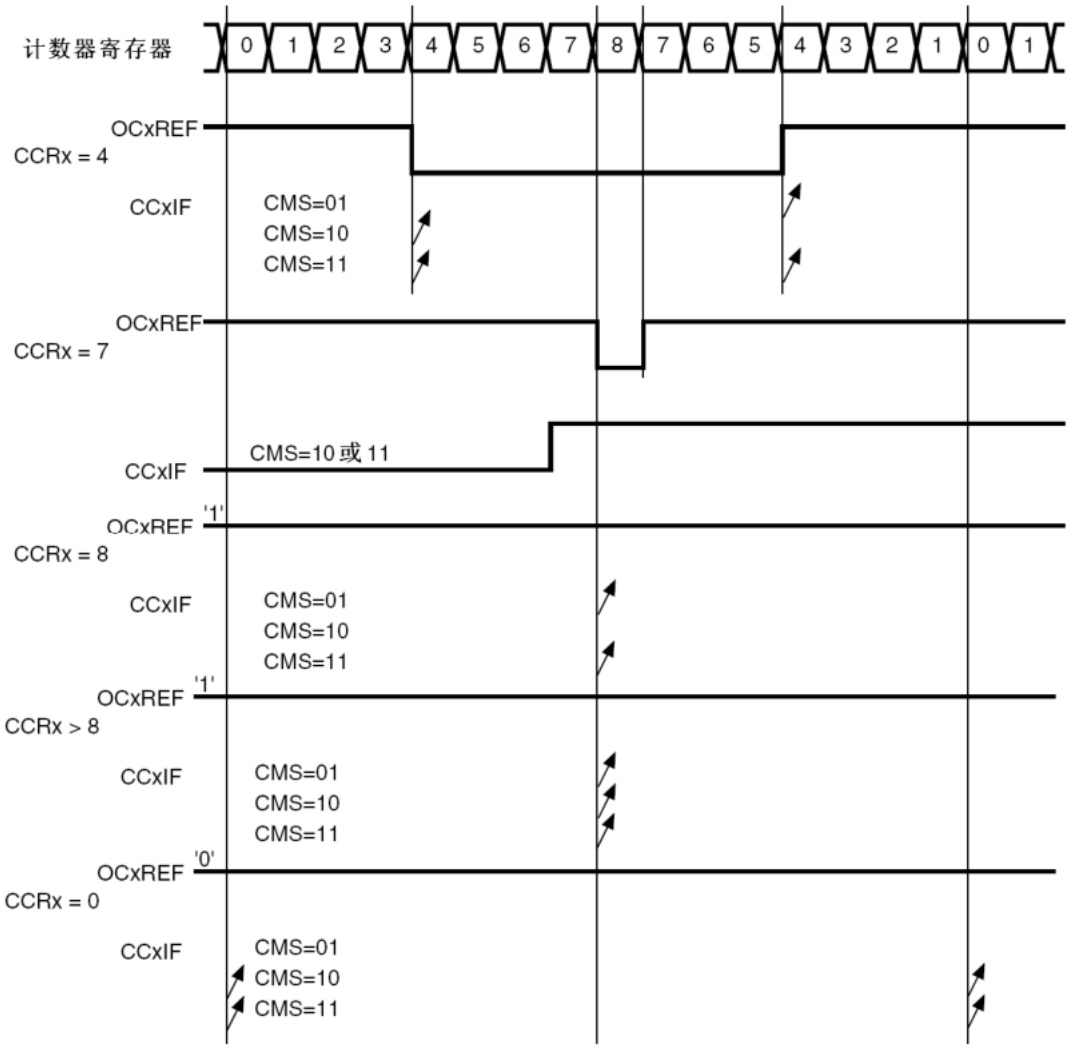


图 15-31

使用中央对齐模式的提示：

- 进入中央对齐模式时，使用当前的向上/向下计数配置；这就意味着计数器向上还是向下计数取决于 TIMERx_CR1 寄存器中 DIR 位的当前值。此外，软件不能同时修改 DIR 和 CMS 位。
- 不推荐当运行在中央对齐模式时改写计数器，因为这会产生不可预知的结果。特别地：
  - 如果写入计数器的值大于自动重加载的值(TIMERx_CNT>TIMERx_ARR)，则方向不会被更新。例如，如果计数器正在向上计数，它就会继续向上计数。
  - 如果将 0 或者 TIMERx_ARR 的值写入计数器，方向被更新，但不产生更新事件 UEV。
- 使用中央对齐模式最保险的方法，就是在启动计数器之前产生一个软件更新(设置 TIMERx_EGR 位中的 UG 位)，并且不要在计数进行过程中修改计数器的值。

15.3.10. 单脉冲模式

单脉冲模式(OPM)是前述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后产生一个脉宽可程序控制的脉冲。

可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIMERx_CR1 寄存器中的 OPM 位将选择单脉冲模式，这样可以让计数器自动地在产生下一个更新事件 UEV 时停止。

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

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

单脉冲模式的例子

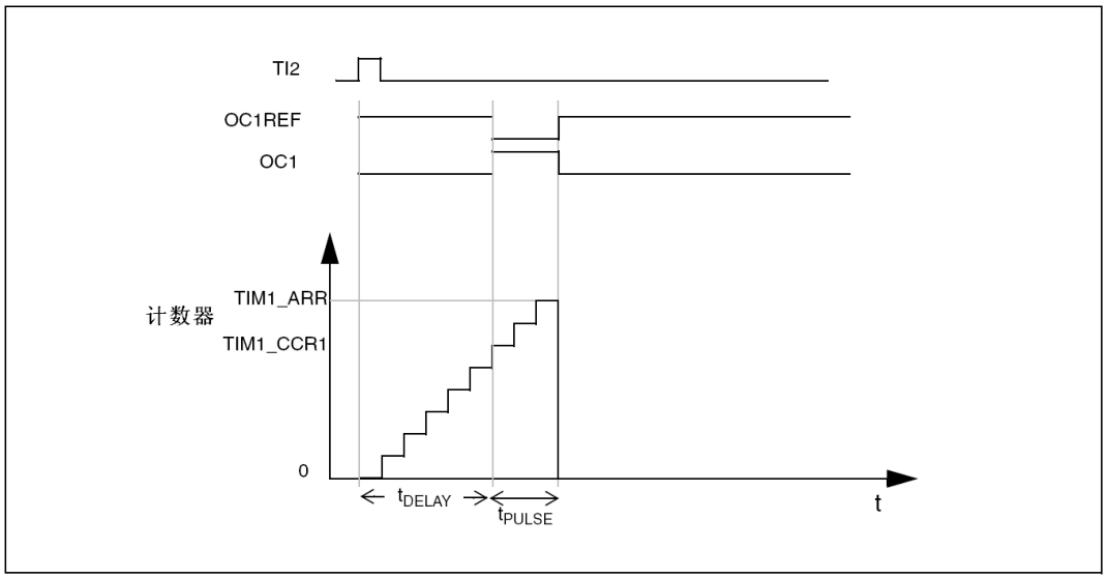


图 15-32

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

假定 TI2FP2 作为触发 1:

- 置 TIMERx_CCMR1 寄存器中的 CC2S=01，把 TI2FP2 映像到 TI2。
- 置 TIMERx_CCER 寄存器中的 CC2P=0，使 TI2FP2 能够检测上升沿。
- 置 TIMERx_SMCR 寄存器中的 TS=110，TI2FP2 作为从模式控制器的触发(TRGI)。
- 置 TIMERx_SMCR 寄存器中的 SMS=110(触发模式)，TI2FP2 被用来启动计数器。

OPM 的波形由写入比较寄存器的数值决定(要考虑时钟频率和计数器预分频器)

- tDELAY 由 TIMERx_CCR1 寄存器中的值定义。
- tPULSE 由自动装载值和比较值之间的差值定义(TIMERx_ARR - TIMERx_CCR1)。
- 假定当发生比较匹配时要产生从 0 到 1 的波形，当计数器达到预装载值时要产生一个从 1 到 0 的波形；

首先要置 TIMERx_CCMR1 寄存器的 OC1M=111，进入 PWM 模式 2；根据需要有选择地使能预装载寄存器：置 TIMERx_CCMR1 中的 OC1PE=1 和 TIMERx_CR1 寄存器中的 ARPE；然后在 TIMERx_CCR1 寄存器中填写比较值，在 TIMERx_ARR 寄存器中填写自动装载值，设置 UG 位来产生一个更新事件，然后等待在 TI2 上的一个外部触发事件。本例中，CC1P=0。

在这个例子中，TIMERx_CR1 寄存器中的 DIR 和 CMS 位应该置低。

因为只需要一个脉冲，所以必须设置 TIMERx_CR1 寄存器中的 OPM=1，在下一个更新事件(当计数器从自动装载值翻转到 0)时停止计数。

特殊情况： 特殊情况：OCx 快速使能：

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

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

15.3.11. 在外部事件时清除 OCxREF 信号

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

该功能只能用于输出比较和 PWM 模式，而不能用于强置模式。

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

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

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

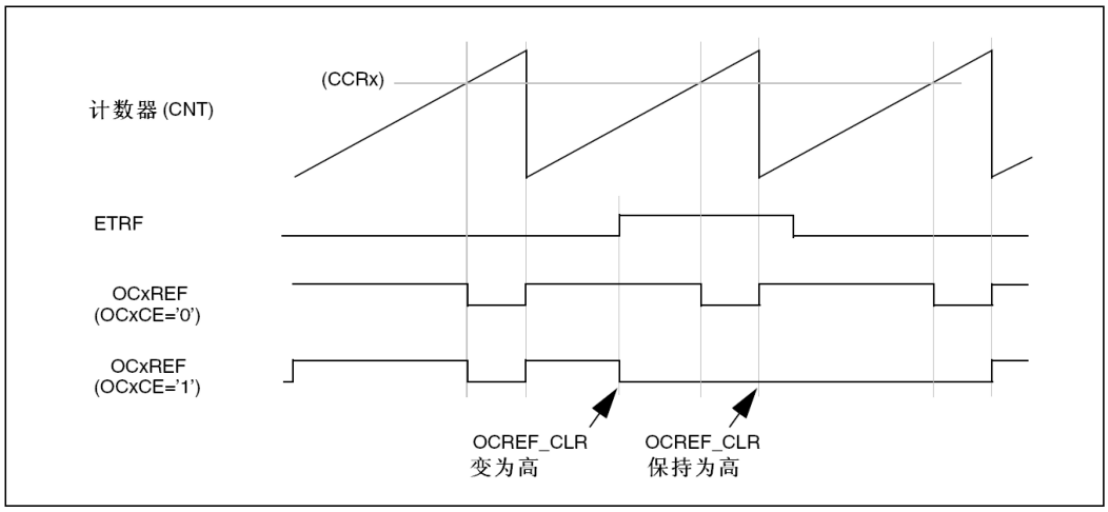


图 15-33

15.3.12. 定时器输入异或功能

TIMERx_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIMERx_CH1、TIMERx_CH2 和 TIMERx_CH3。

异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。

15.3.13. 与霍尔传感器的接口

参考 TIMER1 霍尔传感器接口 章节的描述。

15.3.14. 编码器接口模式

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

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

两个输入 TI1 和 TI2 被用来作为增量编码器的接口。参看 表 15-3 ，假定计数器已经启动 (TIMERx_CR1 寄存器中的 CEN=1)，则计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和变相，则 TI1FP1=TI1，TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数，同时硬件对 TIMERx_CR1 寄存器的 DIR 位进行相应的设置。

不管计数器是依靠 TI1 计数、依靠 TI2 计数或者同时依靠 TI1 和 TI2 计数，在任一输入端(TI1 或者 TI2)的跳变都会重新计算 DIR 位。

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

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

表 15-1 计数方向与编码器信号的关系

有效边沿	相对信号的电平 (TI1FP1 对应 TI2,	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降

	TI2FP2 对应 TI1)				
仅在 TI1 计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在 TI2 计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在 TI1 和 TI2 上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

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

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

- CC1S=“01”(TIMERx_CCMR1 寄存器，IC1FP1 映射到 TI1)
- CC2S=“01”(TIMERx_CCMR2 寄存器，IC2FP2 映射到 TI2)
- CC1P=“0” (TIMERx_CCER 寄存器，IC1FP1 不反相，IC1FP1=TI1)
- CC2P=“0” (TIMERx_CCER 寄存器，IC2FP2 不反相，IC2FP2=TI2)
- SMS=“011” (TIMERx_SMCR 寄存器，所有的输入均在上升沿和下降沿有效)
- CEN=“1” (TIMERx_CR1 寄存器，计数器使能)

编码器模式下的计数器操作实例

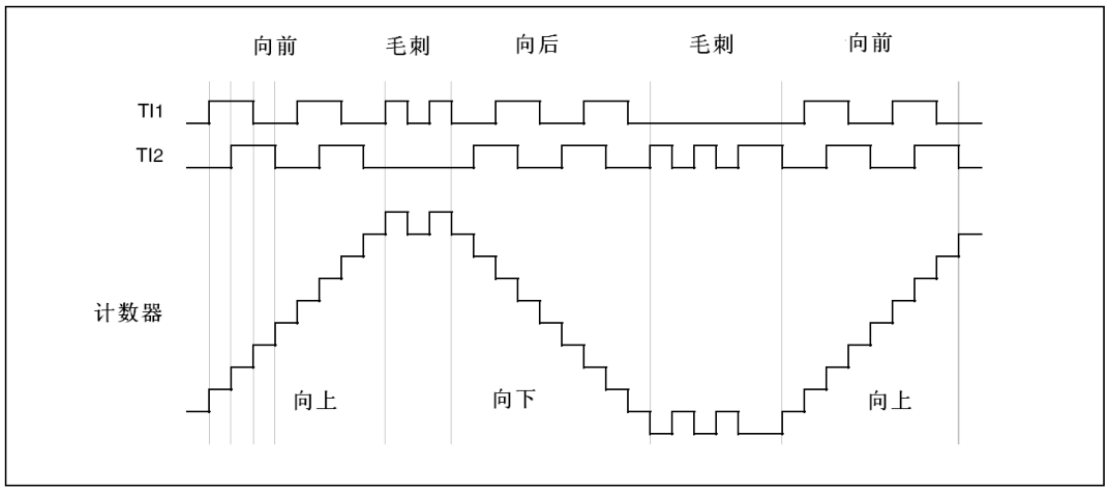


图 15-34

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

IC1FP1 反相的编码器接口模式实例

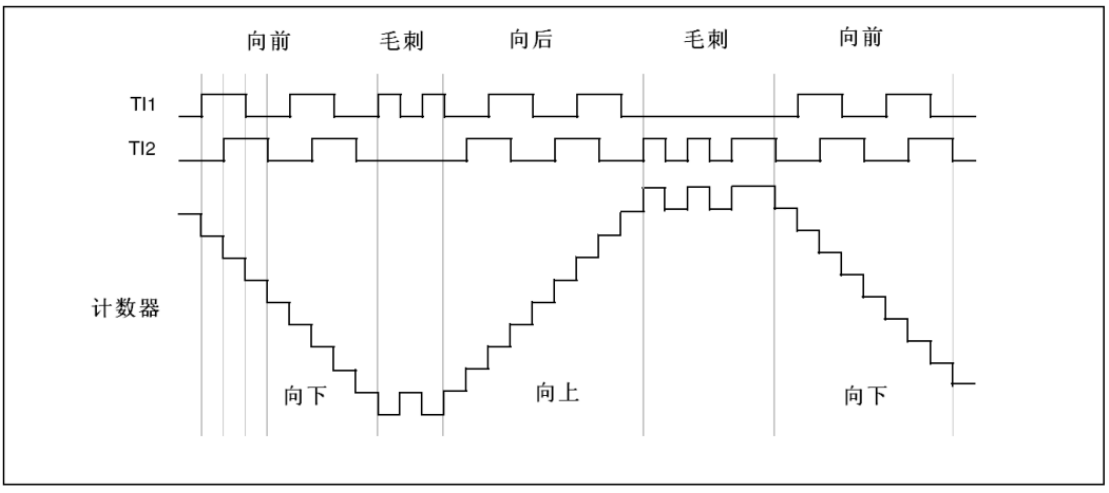


图 15-35

当定时器配置成编码器接口模式时，提供传感器当前位置的信息。使用第二个配置在捕获模式的定时器，可以测量两个编码器事件的间隔，获得动态的信息(速度，加速度，减速度)。指示机械零点的编码器输出可被用做此目的。根据两个事件间的间隔，可以按照固定的时间读出计数器。如果可能的话，你可以把计数器的值锁存到第三个输入捕获寄存器(捕获信号必须是周期的并且可以由另一个定时器产生)；也可以通过一个由实时时钟产生的 DMA 请求来读取它的值。

15.3.15. 定时器输入异或功能

TIMERx_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIMERx_CH1、TIMERx_CH2 和 TIMERx_CH3。

异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。

15.3.16. TIMERx 定时器和外部触发的同步

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

从模式：复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMERx_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器(TIMERx_ARR，TIMERx_CCRx)都被更新了。

在以下的例子中，TI1 输入端的上升沿导致向上计数器被清零：

- 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽(在本例中，不需要任何滤波器，因此保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位只选择输入捕获源，即 TIMERx_CCMR1 寄存器中 CC1S=01。置 TIMERx_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)。
- 置 TIMERx_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIMERx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMERx_CR1 寄存器中 CEN=1，启动计数器。

计数器开始依据内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志(TIMERx_SR 寄存器中的 TIF 位)被设置，根据 TIMERx_DIER 寄存器中 TIE(中断使能)位和 TDE(DMA 使能)位的设置，产生一个中断请求或一个 DMA 请求。

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

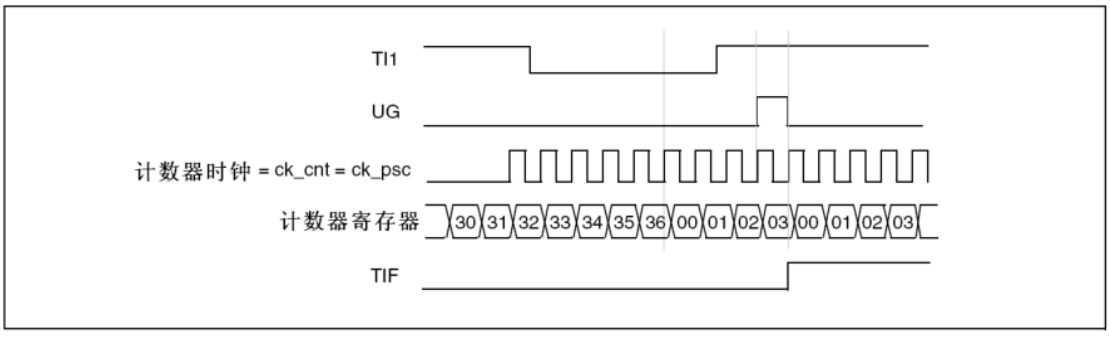


图 15-36

从模式：门控模式

按照选中的输入端电平使能计数器。

在如下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽(本例中，不需要滤波，所以保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输入捕获源，置 TIMERx_CCMR1 寄存器中 CC1S=01。置 TIMERx_CCER 寄存器中 CC1P=1 以确定极性(只检测低电平)。
- 置 TIMERx_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIMERx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMERx_CR1 寄存器中 CEN=1，启动计数器。在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何。

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

设置 TIMERx_SR 中的 TIF 标置。

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

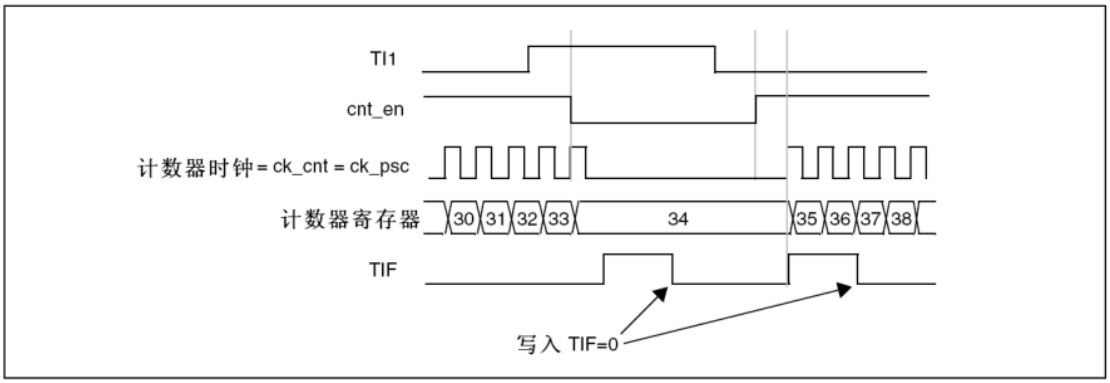


图 15-37

从模式：触发模式

输入端上选中的事件使能计数器。

在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽(本例中，不需要任何滤波器，保持 IC2F=0000)。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIMERx_CCMR1 寄存器中 CC2S=01。置 TIMERx_CCER 寄存器中 CC2P=1 以确定极性(只检测低电平)。
- 置 TIMERx_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIMERx_SMCR 寄存器中 TS=110，选择 TI2 作为输入源。

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

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

触发器模式下的控制电路

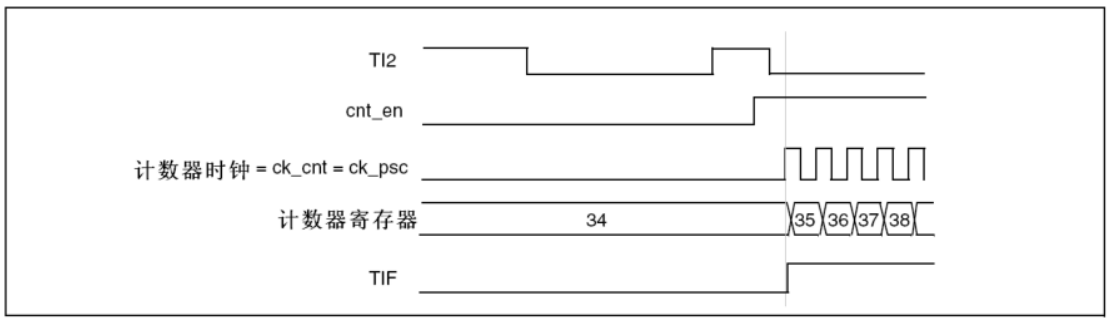


图 15-38

从模式：外部时钟模式 2 + 触发模式

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

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

1. 通过 TIMERx_SMCR 寄存器配置外部触发输入电路：
   - ETF=0000：没有滤波
   - ETPS=00：不用预分频器
   - ETP=0：检测 ETR 的上升沿，置 ECE=1 使能外部时钟模式 2。
2. 按如下配置通道 1，检测 TI 的上升沿：
   - IC1F=0000：没有滤波
   - 触发操作中不使用捕获预分频器，不需要配置
   - 置 TIMERx_CCMR1 寄存器中 CC1S=01，选择输入捕获源
   - 置 TIMERx_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)
3. 置 TIMERx_SMCR 寄存器中 SMS=110，配置定时器为触发模式。置 TIMERx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。

当 TI1 上出现一个上升沿时，TIF 标志被设置，计数器开始在 ETR 的上升沿计数。
ETR 信号的上升沿和计数器实际复位间的延时，取决于 ETRP 输入端的重同步电路。
外部时钟模式 2 + 触发模式下的控制电路

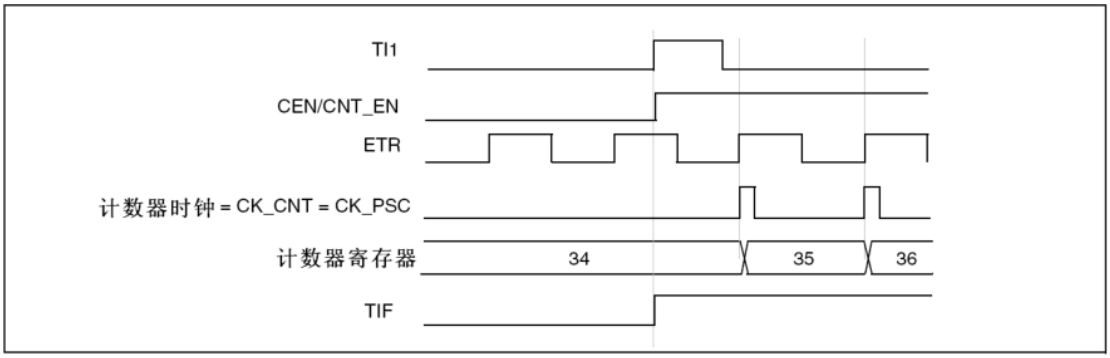


图 15-39

15.3.17. 定时器同步

所有 TIM 定时器在内部相连，用于定时器同步或链接。当一个定时器处于主模式时，它可以对另一个处于从模式的定时器的计数器进行复位、启动、停止或提供时钟等操作。
下图显示了触发选择和主模式选择模块的概况。
使用一个定时器作为另一个定时器的预分频器
主/从定时器的例子

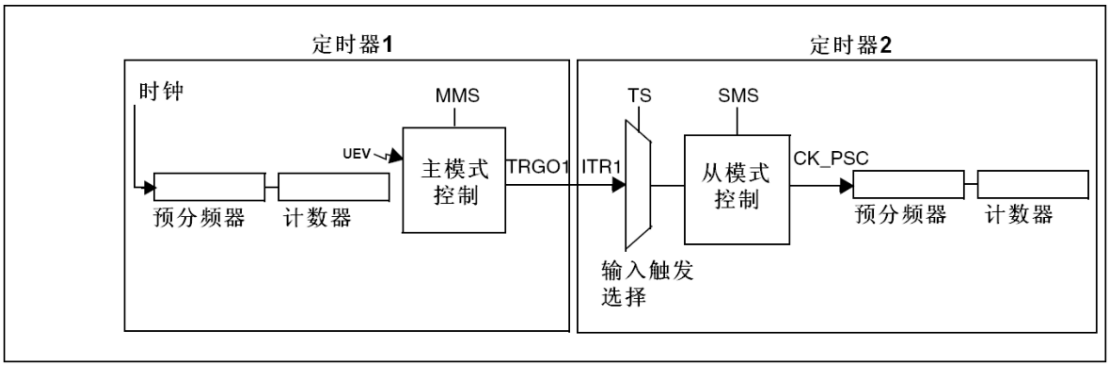


图 15-40

如：可以配置定时器 1 作为定时器 2 的预分频器。参考 图 16-40 ，进行下述操作：

- 配置定时器 1 为主模式，它可以在每一个更新事件 UEV 时输出一个周期性的触发信号。在 TIM1_CR2 寄存器的 MMS= “010” 时，每当产生一个更新事件时在 TRGO1 上输出一个上升沿信号。
- 连接定时器 1 的 TRGO1 输出至定时器 2，设置 TIM2_SMCR 寄存器的 TS= “000” ，配置定时器 2 为使用 ITR1 作为内部触发的从模式。
- 然后把从模式控制器置于外部时钟模式 1(TIM2_SMCR 寄存器的 SMS=111)；这样定时器 2 即可由定时器 1 周期性的上升沿(即定时器 1 的计数器溢出)信号驱动。
- 最后，必须设置相应(TIMERx_CR1 寄存器)的 CEN 位分别启动两个定时器。

注：如果 OCx 已被选中为定时器 1 的触发输出 (MMS=1xx) ，它的上升沿用于驱动定时器 2 的计数器。

使用一个定时器使能另一个定时器 使用一个定时器使能另一个定时器

在这个例子中，定时器 2 的使能由定时器 1 的输出比较控制。参考 图 16-40 的连接。只当定时器 1 的 OC1REF 为高时，定时器 2 才对分频后的内部时钟计数。两个定时器的时钟频率都是由预分频器对 CK_INT 除以 3(fCK_CNT=fCK_INT/3)得到。

- 配置定时器 1 为主模式，送出它的输出比较参考信号(OC1REF)为触发输出(TIM1_CR2 寄存器的 MMS=100)
- 配置定时器 1 的 OC1REF 波形(TIM1_CCMR1 寄存器)
- 配置定时器 2 从定时器 1 获得输入触发(TIM2_SMCR 寄存器的 TS=000)
- 配置定时器 2 为门控模式(TIM2_SMCR 寄存器的 SMS=101)
- 置 TIM2_CR1 寄存器的 CEN=1 以使能定时器 2

• 置 TIM1_CR1 寄存器的 CEN=1 以启动定时器 1

注： 定时器 2 的时钟不与定时器 1 的时钟同步，这个模式只影响定时器 2 计数器的使能信号。

定时器 1 的 OC1REF 控制定时器 2

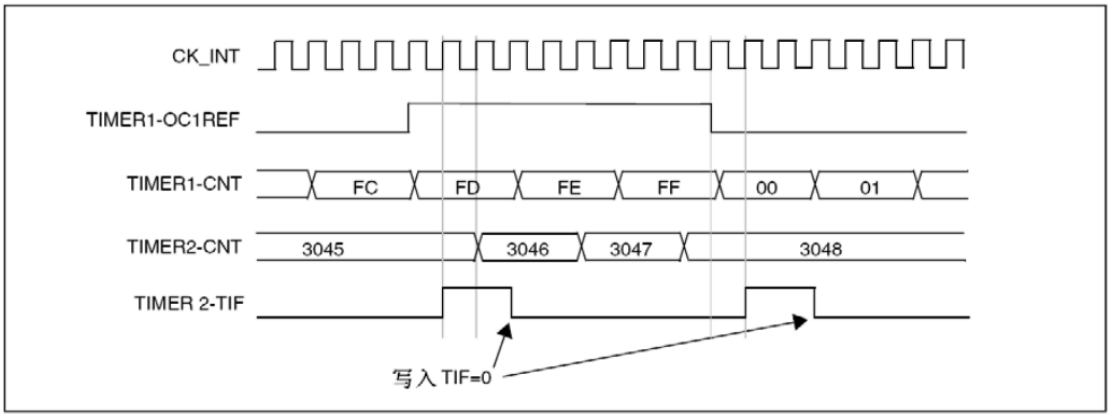


图 15-41

在上图的例子中，在定时器 2 启动之前，它们的计数器和预分频器未被初始化，因此它们从当前的数值开始计数。可以在启动定时器 1 之前复位 2 个定时器，使它们从给定的数值开始，即在定时器计数器中写入需要的任意数值。写 TIMERx_EGR 寄存器的 UG 位即可复位定时器。

在下一个例子中，需要同步定时器 1 和定时器 2。定时器 1 是主模式并从 0 开始，定时器 2 是从模式并从 0xE7 开始；2 个定时器的预分频器系数相同。写’0’到 TIM1_CR1 的 CEN 位将禁止定时器 1，定时器 2 随即停止。

• 配置定时器 1 为主模式，送出输出比较 1 参考信号(OC1REF)做为触发输出(TIM1_CR2 寄存器的 MMS=100)。
• 配置定时器 1 的 OC1REF 波形(TIM1_CCMR1 寄存器)。
• 配置定时器 2 从定时器 1 获得输入触发(TIM2_SMCR 寄存器的 TS=000)
• 配置定时器 2 为门控模式(TIM2_SMCR 寄存器的 SMS=101)
• 置 TIM1_EGR 寄存器的 UG=1，复位定时器 1。
• 置 TIM2_EGR 寄存器的 UG=1，复位定时器 2。
• 写’0xE7’至定时器 2 的计数器(TIM2_CNTL)，初始化它为 0xE7。
• 置 TIM2_CR1 寄存器的 CEN=1,以使能定时器 2。
• 置 TIM1_CR1 寄存器的 CEN=1,以启动定时器 1。
• 置 TIM1_CR1 寄存器的 CEN=0,以停止定时器 1。

通过使能定时器 1 可以控制定时器 2

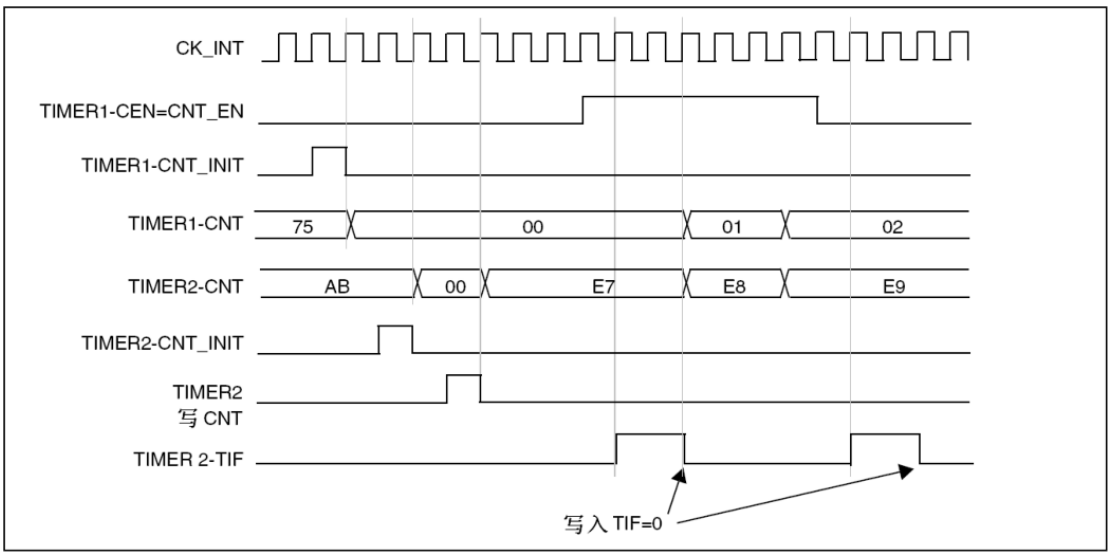


图 15-42

使用一个定时器去启动另一个定时器

在这个例子中，使用定时器 1 的更新事件使能定时器 2。参考 图 16-40 的连接。一旦定时器 1 产生

更新事件，定时器 2 即从它当前的数值(可以是非 0)按照分频的内部时钟开始计数。在收到触发信号时，定时器 2 的 CEN 位被自动地置’1’，同时计数器开始计数直到写’0’到 TIM2_CR1 寄存器的 CEN 位。两个定时器的时钟频率都是由预分频器对 CK_INT 除以 3(fCK_CNT=fCK_INT/3)。

- 配置定时器 1 为主模式，送出它的更新事件(UEV)做为触发输出(TIM1_CR2 寄存器的 MMS=010)。
- 配置定时器 1 的周期(TIM1_ARR 寄存器)。
- 配置定时器 2 从定时器 1 获得输入触发(TIM2_SMCR 寄存器的 TS=000)
- 配置定时器 2 为触发模式(TIM2_SMCR 寄存器的 SMS=110)
- 置 TIM1_CR1 寄存器的 CEN=1 以启动定时器 1。

使用定时器 1 的更新触发定时器 2

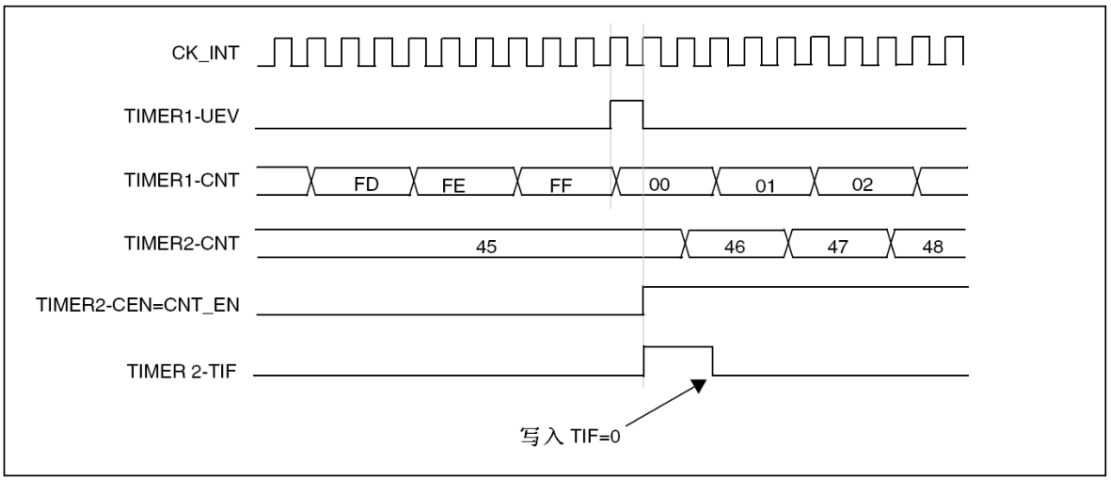


图 15-43

在上一个例子中，可以在启动计数之前初始化两个计数器。 显示在与 0 相同配置情况下，使用触发模式而不是门控模式(TIM2_SMCR 寄存器的 SMS=110)的动作。

利用定时器 1 的使能触发定时器 2

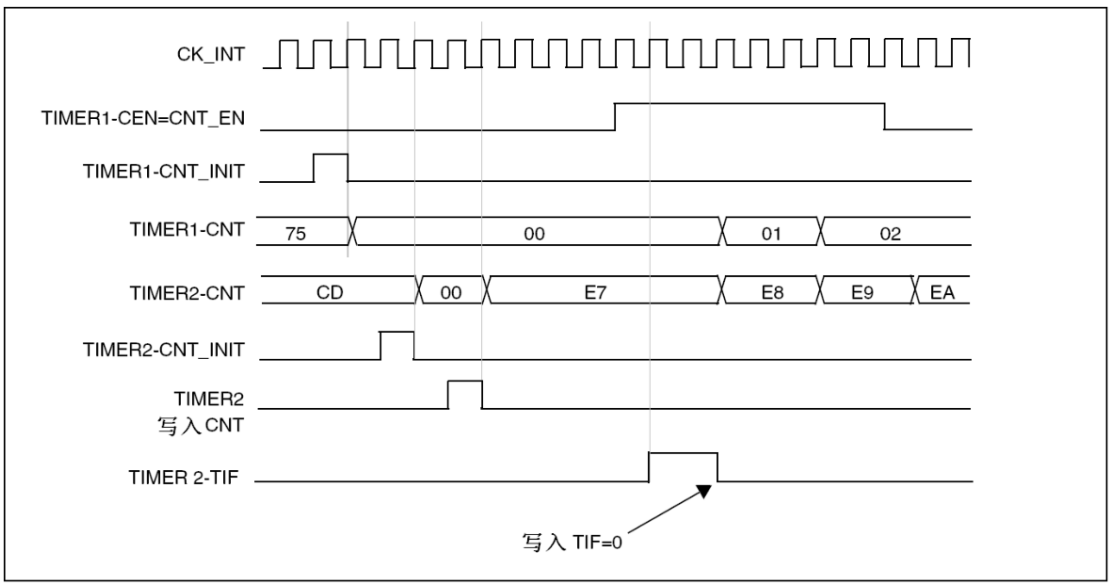


图 15-44

使用一个定时器作为另一个的预分频器

这个例子使用定时器 1 作为定时器 2 的预分频器。配置如下：

- 配置定时器 1 为主模式，送出它的更新事件 UEV 做为触发输出(TIM1_CR2 寄存器的 MMS=’010’)。然后每次计数器溢出时输出一个周期信号。
- 配置定时器 1 的周期(TIM1_ARR 寄存器)。
- 配置定时器 2 从定时器 1 获得输入触发(TIM2_SMCR 寄存器的 TS=000)
- 配置定时器 2 使用外部时钟模式(TIM2_SMCR 寄存器的 SMS=111)
- 置 TIM1_CR2 寄存器的 CEN=1 以启动定时器 2。
- 置 TIM1_CR1 寄存器的 CEN=1 以启动定时器 1。

使用一个外部触发同步地启动 使用一个外部触发同步地启动 2 个定时器

这个例子中当定时器 1 的 TI1 输入上升时使能定时器 1，使能定时器 1 的同时使能定时器 2。为保证计数器的对齐，定时器 1 必须配置为主/从模式(对应 TI1 为从，对应定时器 2 为主)：

- 配置定时器 1 为主模式，送出它的使能做为触发输出(TIM1_CR2 寄存器的 MMS=’ 001’)。
- 配置定时器 1 为从模式，从 TI1 获得输入触发(TIM1_SMCR 寄存器的 TS=’ 100’)。
- 配置定时器 1 为触发模式(TIM1_SMCR 寄存器的 SMS=’ 110’)。
- 配置定时器 1 为主/从模式，TIM1_SMCR 寄存器的 MSM=’ 1’ 。
- 配置定时器 2 从定时器 1 获得输入触发(TIM2_SMCR 寄存器的 TS=000)
- 配置定时器 2 为触发模式(TIM2_SMCR 寄存器的 SMS=’ 110’)。

当定时器 1 的 TI1 上出现一个上升沿时，两个定时器同步地按照内部时钟开始计数，两个 TIF 标志也同时被设置。

注： 在这个例子中，在启动之前两个定时器都被初始化（ 设置相应的 UG 位 ），两个计数器都从 0 开始，但可以通过写入任意一个计数器寄存器（TIMERx_CNT）在定时器间插入一个偏移。下图中能看到主/从模式下在定时器 1 的 CNT_EN 和 CK_PSC 之间有个延迟。

使用定时器 1 的 TI1 输入触发定时器 1 和定时器 2

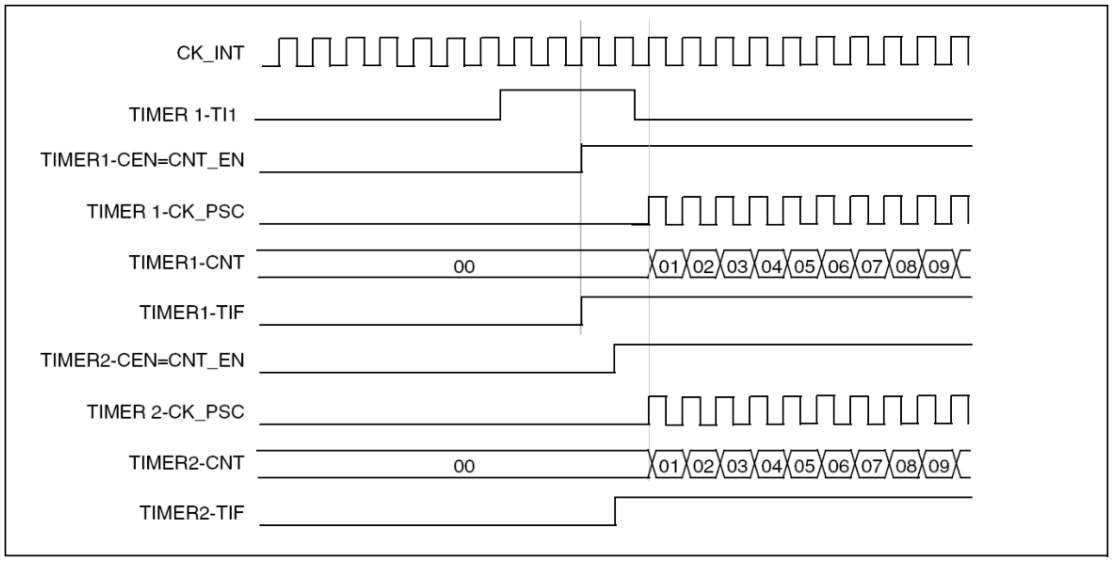


图 15-45

15.4. 寄存器描述

15.4.1. 寄存器列表

Base address：0x4002 0000

Name	Offset	Reset	Description
TIMERx_CR1	0x460/0x4C0	32'h0	TIMERx 控制寄存器 1
TIMERx_CR2	0x464/0x4C4	32'h0	TIMERx 控制寄存器 2
TIMERx_SMCR	0x468/0x4C8	32'h0	TIMERx 从模式控制寄存器
TIMERx_DIER	0x46C/0x4CC	32'h0	TIMERx DMA/中断使能寄存器
TIMERx_SR	0x470/0x4D0	32'h0	TIMERx 状态寄存器
TIMERx_EGR	0x474/0x4D4	32'h0	TIMERx 事件产生寄存器
TIMERx_CCMR1	0x478/0x4D8	32'h0	TIMERx 捕获/比较模式寄存器 1
TIMERx_CCMR2	0x47C/0x4DC	32'h0	TIMERx 捕获/比较模式寄存器 2
TIMERx_CCER	0x480/0x4E0	32'h0	TIMERx 捕获/比较使能寄存器
TIMERx_CNT	0x484/0x4E4	32'h0	TIMERx 计数器
TIMERx_PSC	0x488/0x4E8	32'h0	TIMERx 预分频器
TIMERx_ARR	0x48C/0x4EC	32'hFFFF	TIMERx 自动重载寄存器

Reserved	-	-	-
TIMERx_CCR1	0x494/0x4F4	32'hFFFF	TIMERx 捕获/比较寄存器 1
TIMERx_CCR2	0x498/0x4F8	32'hFFFF	TIMERx 捕获/比较寄存器 2
TIMERx_CCR3	0x49C/0x4FC	32'hFFFF	TIMERx 捕获/比较寄存器 3
TIMERx_CCR4	0x4A0/0x500	32'hFFFF	TIMERx 捕获/比较寄存器 4

15.4.2. 寄存器详细说明

15.4.2.1. 控制寄存器 1（TIMERx_CR1）

Width	Name	Reset	Property	Description
31:10	Reserved	-	-	-
9:8	CKD	2'b0	RW	时钟分频因子 这 2 位定义在定时器时钟(CK_INT)频率、死区时间和由死区发生器与数字滤波器(ETR,TIx)所用的采样时钟之间的分频比例。 b00: tDTS = t CK_INT b01: tDTS = 2 x t CK_INT b10: tDTS = 4 x t CK_INT b11: 保留，不要使用这个配置
7	ARPE	1'b0	RW	自动重装载预装载允许位 0: TIMERx_ARR 寄存器没有缓冲 1: TIMERx_ARR 寄存器被装入缓冲器。
6:5	CMS	2'b0	RW	选择中央对齐模式 b00: 边沿对齐模式。计数器依据方向位(DIR)向上或向下计数。 b01: 中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道(TIMERx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器向下计数时被设置。 b10: 中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道(TIMERx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器向上计数时被设置。 b11: 中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道(TIMERx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，在计数器向上和向下计数时均被设置。 注：在计数器开启时(CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。
4	DIR	1'b0	RW	计数方向 0: 计数器向上计数 1: 计数器向下计数 注：当计数器配置为中央对齐模式或编码器模式时，该位为只读
3	OPM	1'b0	RW	单脉冲模式 0: 在发生更新事件时，计数器不停止 1: 在发生下一次更新事件(清除 CEN 位)时，计

				数器停止
2	URS	1'b0	RW	更新请求源 0: 如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求： – 计数器溢出/下溢 – 设置 UG 位 – 从模式控制器产生的更新 1: 如果使能了更新中断或 DMA 请求，则只有计数器溢出/下溢才产生更新中断或 DMA 请求。
1	UDIS	1'b0	RW	禁止更新 软件通过该位允许/禁止 UEV 事件的产生 0: 允许 UEV。更新(UEV)事件由下述任一事件产生： – 计数器溢出/下溢 – 设置 UG 位 – 从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。(译注：更新影子寄存器) 1: 禁止 UEV。不产生更新事件，影子寄存器(ARR、PSC、CCRx)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。
0	CEN	1'b0	RW	使能计数器 0: 禁止计数器 1: 使能计数器 注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

15.4.2.2. 控制寄存器 2（TIMERx_CR2）

Width	Name	Reset	Property	Description
31:8	Reserved	-	-	-
7	TI1S	1'b0	RW	TI1 选择 0: TIMERx_CH1 引脚连到 TI1 输入 1: TIMERx_CH1、TIMERx_CH2 和 TIMERx_CH3 引脚经异或后连到 TI1 输入
6:4	MMS	3'b0	RW	主模式选择 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下： b000: 复位 – TIMERx_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式)，则 TRGO 上的信号相对实际的复位会有一个延迟。 b001: 使能 – 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门

				控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式(见 TIMERx_SMCR 寄存器中 MSM 位的描述)。 b010: 更新 – 更新事件被选为触发输入(TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 b011: 比较脉冲 – 在发生一次捕获或一次比较成功时，当要设置 CC1IF 标志时(即使它已经为高)，触发输出送出一个正脉冲(TRGO)。 b100: 比较 – OC1REF 信号被用于作为触发输出(TRGO)。 b101: 比较 – OC2REF 信号被用于作为触发输出(TRGO)。 b110: 比较 – OC3REF 信号被用于作为触发输出(TRGO)。 b111: 比较 – OC4REF 信号被用于作为触发输出(TRGO)。
3	CCDS	1'b0	RW	捕获/比较的 DMA 选择 0: 当发生 CCx 事件时，送出 CCx 的 DMA 请求 1: 当发生更新事件时，送出 CCx 的 DMA 请求
2:0	Reserved	-	-	-

15.4.2.3. 从模式控制寄存器（TIMERx_SMCR)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	ETP	1'b0	RW	外部触发极性 该位选择是用 ETR 还是 ETR 的反相来作为触发操作 0: ETR 不反相，高电平或上升沿有效 1: ETR 被反相，低电平或下降沿有效
14	ECE	1'b0	RW	外部时钟使能位 该位启用外部时钟模式 2 0: 禁止外部时钟模式 2 1: 使能外部时钟模式 2。计数器由 ETRF 信号上的任意有效边沿驱动 注 1: 设置 ECE 位与选择外部时钟模式 1 并将 TRGI 连到 ETRF(SMS=111 和 TS=111)具有相同功效。 注 2: 下述从模式可以与外部时钟模式 2 同时使用：复位模式，门控模式和触发模式；但是，这时 TRGI 不能连到 ETRF(TS 位不能是“111”)。 注 3: 外部时钟模式 1 和外部时钟模式 2 同时被使能时，外部时钟的输入是 ETRF。
13:12	ETPS	2'b0	RW	外部触发预分频 b00: 关闭预分频 b01: ETRP 频率除以 2

				b10: ETRP 频率除以 4 b11: ETRP 频率除以 8
11:8	ETF	4'b0	RW	外部触发滤波 这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。 h0: 无滤波器，以 fDTS 采样 h1: 采样频率 fSAMPLING=fCK_INT，N=2 h2: 采样频率 fSAMPLING=fCK_INT，N=4 h3: 采样频率 fSAMPLING=fCK_INT，N=8 h4: 采样频率 fSAMPLING=fDTS/2，N=6 h5: 采样频率 fSAMPLING=fDTS/2，N=8 h6: 采样频率 fSAMPLING=fDTS/4，N=6 h7: 采样频率 fSAMPLING=fDTS/4，N=8 h8: 采样频率 fSAMPLING=fDTS/8，N=6 h9: 采样频率 fSAMPLING=fDTS/8，N=8 ha: 采样频率 fSAMPLING=fDTS/16，N=5 hb: 采样频率 fSAMPLING=fDTS/16，N=6 hc: 采样频率 fSAMPLING=fDTS/16，N=8 hd: 采样频率 fSAMPLING=fDTS/32，N=5 he: 采样频率 fSAMPLING=fDTS/32，N=6 hf: 采样频率 fSAMPLING=fDTS/32，N=8
7	MSM	1'b0	RW	主/从模式 0: 无作用 1: 触发输入(TRGI)上的事件被延迟了，以允许在当前定时器(通过 TRGO)与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的
6:4	TS	3'b0	RW	触发选择 这 3 位选择用于同步计数器的触发输入。 b000: 内部触发 0(ITR0) b001: 内部触发 1(ITR1) b010: 内部触发 2(ITR2) b011: 内部触发 3(ITR3) b100: TI1 的边沿检测器(TI1F_ED) b101: 滤波后的定时器输入 1(TI1FP1) b110: 滤波后的定时器输入 2(TI2FP2) b111: 外部触发输入(ETRF)
3	Reserved	-	-	-
2:0	SMS	3'b0	RW	从模式选择 当选择了外部信号，触发信号(TRGI)的有效边沿与选中的外部输入极性相关(见输入控制寄存器和控制寄存器的说明) b000: 关闭从模式 – 如果 CEN=1，则预分频器直接由内部时钟驱动。 b001: 编码器模式 1 – 根据 TI1FP1 的电平，

				计数器在 TI2FP2 的边沿向上/下计数。 b010: 编码器模式 2 – 根据 TI2FP2 的电平，计数器在 TI1FP1 的边沿向上/下计数。 b011: 编码器模式 3 – 根据另一个信号的输入电平，计数器在 TI1FP1 和 TI2FP2 的边沿向上/下计数。 b100: 复位模式 – 选中的触发输入(TRGI)的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。 b101: 门控模式 – 当触发输入(TRGI)为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止(但不复位)。计数器的启动和停止都是受控的。 b110: 触发模式 – 计数器在触发输入 TRGI 的上升沿启动(但不复位)，只有计数器的启动是受控的。 b111: 外部时钟模式 1 – 选中的触发输入(TRGI)的上升沿驱动计数器。 注： 1. 如果 TI1F_ED 被选为触发输入(TS=100)时，不要使用门控模式。这是因为，TI1F_ED 在每次 TI1F 变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。 2. 如果配置成编码器模式，且 ECE=0，TIMx_ETR 的上升沿会对计数器清零。

TIMER1 内部触发链接

从定时器	ITR0 (TS=000)	ITR1 (TS=001)	ITR2 (TS=010)	ITR3 (TS=011)
TIMER2	TIMER1	TIMER3	TIMER4	TIMER5
TIMER3	TIMER1	TIMER2	TIMER4	TIMER6
TIMER4	TIMER1	TIMER2	TIMER5	TIMER6
TIMER5	TIMER1	TIMER3	TIMER4	TIMER6
TIMER6	TIMER1	TIMER2	TIMER3	TIMER4

15.4.2.4. DMA/中断使能寄存器（TIMERx_DIER)

Width	Name	Reset	Property	Description
31:15	Reserved	-	-	-
14	TDE	1'b0	RW	允许触发 DMA 请求 0: 禁止触发 DMA 请求 1: 允许触发 DMA 请求
13	Reserved	-	-	-
12	CC4DE	1'b0	RW	允许捕获/比较 4 的 DMA 请求 0: 禁止捕获/比较 4 的 DMA 请求 1: 允许捕获/比较 4 的 DMA 请求
11	CC3DE	1'b0	RW	允许捕获/比较 3 的 DMA 请求 0: 禁止捕获/比较 3 的 DMA 请求 1: 允许捕获/比较 3 的 DMA 请求

10	CC2DE	1'b0	RW	允许捕获/比较 2 的 DMA 请求 0: 禁止捕获/比较 2 的 DMA 请求 1: 允许捕获/比较 2 的 DMA 请求
9	CC1DE	1'b0	RW	允许捕获/比较 1 的 DMA 请求 0: 禁止捕获/比较 1 的 DMA 请求 1: 允许捕获/比较 1 的 DMA 请求
8	UDE	1'b0	RW	允许更新的 DMA 请求 0: 禁止更新的 DMA 请求 1: 允许更新的 DMA 请求
7	Reserved	-	-	-
6	TIE	1'b0	RW	触发中断使能 0: 禁止触发中断 1: 使能触发中断
5	Reserved	-	-	-
4	CC4IE	1'b0	RW	允许捕获/比较 4 中断 0: 禁止捕获/比较 4 中断 1: 允许捕获/比较 4 中断
3	CC3IE	1'b0	RW	允许捕获/比较 3 中断 0: 禁止捕获/比较 3 中断 1: 允许捕获/比较 3 中断
2	CC2IE	1'b0	RW	允许捕获/比较 2 中断 0: 禁止捕获/比较 2 中断 1: 允许捕获/比较 2 中断
1	CC1IE	1'b0	RW	允许捕获/比较 1 中断 0: 禁止捕获/比较 1 中断 1: 允许捕获/比较 1 中断
0	UIE	1'b0	RW	允许更新中断 0: 禁止更新中断 1: 允许更新中断

15.4.2.5. 状态寄存器（TIMERx_SR)

Width	Name	Reset	Property	Description
31:13	Reserved	-	-	-
12	CC4OF	1'b0	RC	捕获 4 重复捕获标记 参见 CC1OF 描述。
11	CC3OF	1'b0	RC	捕获 3 重复捕获标记 参见 CC1OF 描述。
10	CC2OF	1'b0	RC	捕获 2 重复捕获标记 参见 CC1OF 描述。
9	CC1OF	1'b0	RC	捕获 1 重复捕获标记 0: 无重复捕获产生 1: 计数器的值被捕获到 TIMERx_CCR1 寄存器时，CC1IF 的状态已经为‘1’。
8:7	Reserved	-	-	-
6	TIF	1'b0	RC	触发器中断标记 当发生触发事件(当从模式控制器处于除门控模式外的其它模式时，在 TRGI 输入端检测到有效

				边沿，或门控模式下的任一边沿)时由硬件对该位置 1，它由软件清 0。 0: 无触发器事件产生 1: 触发中断等待响应
5	Reserved	-	-	-
4	CC4IF	1'b0	RC	捕获/比较 4 中断标记 参考 CC1IF 描述
3	CC3IF	1'b0	RC	捕获/比较 3 中断标记 参考 CC1IF 描述
2	CC2IF	1'b0	RC	捕获/比较 2 中断标记 参考 CC1IF 描述
1	CC1IF	1'b0	RC	捕获/比较 1 中断标记 **如果通道 如果通道 CC1 配置为输出模式:** 当计数器值与比较值匹配时该位由硬件置 1，但在中心对称模式下除外(参考 TIMERx_CR1 寄存器的 CMS 位)。它由软件清 0。 0: 无匹配发生; 1: TIMERx_CNT 的值与 TIMERx_CCR1 的值匹配。 当 TIMERx_CCR1 的内容大于 TIMERx_ARR 的内容时，在向上或向上/下计数模式时计数器溢出，或向下计数模式时的计数器下溢条件下，CC1IF 位变高 **如果通道 CC1 配置为输入模式:** 当捕获事件发生时该位由硬件置 1，它由软件清 0 或通过读 TIMERx_CCR1 清 0。 0: 无输入捕获产生 1: 计数器值已被捕获(拷贝)至 TIMERx_CCR1(在 IC1 上检测到与所选极性相同的边沿)
0	UIF	1'b0	RC	更新中断标记 当产生更新事件时该位由硬件置 1。它由软件清 0。 0: 无更新事件产生; 1: 更新中断等待响应。当寄存器被更新时该位由硬件置 1: – 若 TIMERx_CR1 寄存器的 UDIS=0，当重复计数器数值上溢或下溢时(重复计数器=0 时产生更新事件)。 – 若 TIMERx_CR1 寄存器的 URS=0、UDIS=0，当设置 TIMERx_EGR 寄存器的 UG=1 时产生更新事件，通过软件对计数器 CNT 重新初始化时。 – 若 TIMERx_CR1 寄存器的 URS=0、UDIS=0，当计数器 CNT 被触发事件重新初始化时。(参考从模式控制寄存器 TIMERx_SMCR)。

15.4.2.6. 事件产生寄存器（TIMERx_EGR）

Width	Name	Reset	Property	Description
31:7	Reserved	-	-	-
6	TG	1'b0	WO	产生触发事件 0: 无动作 1: TIMERx_SR 寄存器的 TIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA
5	Reserved	-	-	-
4	CC4G	1'b0	WO	产生捕获/比较 4 事件 参考 CC1G 描述。
3	CC3G	1'b0	WO	产生捕获/比较 3 事件 参考 CC1G 描述。
2	CC2G	1'b0	WO	产生捕获/比较 2 事件 参考 CC1G 描述。
1	CC1G	1'b0	WO	产生捕获/比较 1 事件 该位由软件置 1，用于产生一个捕获/比较事件，由硬件自动清 0。 0: 无动作 1: 在通道 CC1 上产生一个捕获/比较事件 **若通道 CC1 配置为输出：** 设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。 **若通道 CC1 配置为输入：** 当前的计数器值被捕获至 TIMERx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF=1。
0	UG	1'b0	WO	产生更新事件 该位由软件置 1，由硬件自动清 0。 0: 无动作； 1: 重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清 0(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清 0；若 DIR=1(向下计数)则计数器取 TIMERx_ARR 的值。

15.4.2.7. 捕获/比较模式寄存器 1（TIMERx_CCMR1）

通道可用于输入(捕获模式)或输出(比较模式)，通道的方向由相应的 CCxS 位定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

输出比较模式：

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	OC2CE	1'b0	RW	输出比较 2 清 0 使能 参考 CC1CE
14:12	OC2M	3'b0	RW	输出比较 2 模式

				参考 OC1M
11	OC2PE	1'b0	RW	输出比较 2 预装载使能 参考 OC1PE
10	OC2FE	1'b0	RW	输出比较 2 快速使能 参考 OC1FE
9:8	CC2S	2'b0	RW	捕获/比较 2 选择 该位定义通道的方向(输入/输出)，及输入脚的选择： b00: CC2 通道被配置为输出 b01: CC2 通道被配置为输入，IC2 映射在 TI2 上 b10: CC2 通道被配置为输入，IC2 映射在 TI1 上 b11: CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择) 注：CC2S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC2E=0)才是可写的
7	OC1CE	1'b0	RW	输出比较 1 清 0 使能 0: OC1REF 不受 ETRF 输入的影响 1：一旦检测到 ETRF 输入高电平，清除 OC1REF=0
6:4	OC1M	3'b0	RW	输出比较 1 模式 该 3 位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效，而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。 b000: 冻结。输出比较寄存器 TIMERx_CCR1 与计数器 TIMERx_CNT 间的比较对 OC1REF 不起作用 b001: 匹配时设置通道 1 为有效电平。当计数器 TIMERx_CNT 的值与捕获/比较寄存器 1(TIMERx_CCR1)相同时，强制 OC1REF 为高 b010: 匹配时设置通道 1 为无效电平。当计数器 TIMERx_CNT 的值与捕获/比较寄存器 1(TIMERx_CCR1)相同时，强制 OC1REF 为低 b011: 翻转。当 TIMERx_CCR1=TIMERx_CNT 时，翻转 OC1REF 的电平 b100: 强制为无效电平。强制 OC1REF 为低 b101: 强制为有效电平。强制 OC1REF 为高 b110: PWM 模式 1 - 在向上计数时，一旦 TIMERx_CNT<TIMERx_CCR1 时通道 1 为有效电平，否则为无效电平；在向下计数时，一旦 TIMERx_CNT>TIMERx_CCR1 时通道 1 为无效电平(OC1REF=0)，否则为有效电平(OC1REF=1) b111: PWM 模式 2 - 在向上计数时，一旦

				TIMERx_CNT<TIMERx_CCR1 时通道 1 为无效电平，否则为有效电平；在向下计数时，一旦 TIMERx_CNT>TIMERx_CCR1 时通道 1 为有效电平，否则为无效电平 注 1: 一旦 LOCK 级别设为 3(TIMERx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。
3	OC1PE	1'b0	RW	输出比较 1 预装载使能 0: 禁止 TIMERx_CCR1 寄存器的预装载功能，可随时写入 TIMERx_CCR1 寄存器，并且新写入的数值立即起作用。 1: 开启 TIMERx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作， TIMERx_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。 注 1: 一旦 LOCK 级别设为 3(TIMERx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 仅在单脉冲模式下(TIMERx_CR1 寄存器的 OPM=1)，可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定。
2	OC1FE	1'b0	RW	输出比较 1 快速使能 该位用于加快 CC 输出对触发输入事件的响应。 0: 根据计数器与 CCR1 的值，CC1 正常操作，即使触发器是打开的。当触发器的输入有一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期。 1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。 OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。
1:0	CC1S	2'b0	RW	捕获/比较 1 选择 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： b00：CC1 通道被配置为输出； b01：CC1 通道被配置为输入，IC1 映射在 TI1 上； b10：CC1 通道被配置为输入，IC1 映射在 TI2 上； b11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由

				TIMERx_SMCR 寄存器的 TS 位选择)。 注：CC1S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC1E=0)才是可写的。

输入捕获模式：

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:12	IC2F	4'b0	RW	输入捕获 2 滤波器 参考 IC1F
11:10	IC2PSC	2'b0	RW	输入/捕获 2 预分频器 参考 IC1PSC
9:8	CC2S	2'b0	RW	捕获/比较 2 选择 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： b00: CC2 通道被配置为输出 b01: CC2 通道被配置为输入，IC2 映射在 TI2 上 b10: CC2 通道被配置为输入，IC2 映射在 TI1 上 b11: CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择) 注：CC2S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC2E=0)才是可写的。
7:4	IC1F	4'b0	RW	输入捕获 1 滤波器 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变： h0: 无滤波器，以 fDTS 采样 h1: 采样频率 fSAMPLING=fCK_INT，N=2 h2: 采样频率 fSAMPLING=fCK_INT，N=4 h3: 采样频率 fSAMPLING=fCK_INT，N=8 h4: 采样频率 fSAMPLING=fDTS/2，N=6 h5: 采样频率 fSAMPLING=fDTS/2，N=8 h6: 采样频率 fSAMPLING=fDTS/4，N=6 h7: 采样频率 fSAMPLING=fDTS/4，N=8 h8: 采样频率 fSAMPLING=fDTS/8，N=6 h9: 采样频率 fSAMPLING=fDTS/8，N=8 ha: 采样频率 fSAMPLING=fDTS/16，N=5 hb: 采样频率 fSAMPLING=fDTS/16，N=6 hc: 采样频率 fSAMPLING=fDTS/16，N=8 hd: 采样频率 fSAMPLING=fDTS/32，N=5 he: 采样频率 fSAMPLING=fDTS/32，N=6 hf: 采样频率 fSAMPLING=fDTS/32，N=8
3:2	IC1PSC	2'b0	RW	输入/捕获 1 预分频器 这 2 位定义了 CC1 输入(IC1)的预分频系数。一旦 CC1E=0(TIMERx_CCER 寄存器中)，则预分

				频器复位。 b00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 b01: 每 2 个事件触发一次捕获 b10: 每 4 个事件触发一次捕获 b11: 每 8 个事件触发一次捕获
1:0	CC1S	2'b0	RW	捕获/比较 1 选择 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： b00: CC1 通道被配置为输出； b01: CC1 通道被配置为输入，IC1 映射在 TI1 上； b10: CC1 通道被配置为输入，IC1 映射在 TI2 上； b11: CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择)。 注：CC1S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC1E=0)才是可写的。

15.4.2.8. 捕获/比较模式寄存器 2（TIMERx_CCMR2）

输出比较模式：

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	OC4CE	1'b0	RW	输出比较 4 清 0 使能 参考 CC1CE
14:12	OC4M	3'b0	RW	输出比较 4 模式 参考 OC1M
11	OC4PE	1'b0	RW	输出比较 4 预装载使能 参考 OC1PE
10	OC4FE	1'b0	RW	输出比较 4 快速使能 参考 OC1FE
9:8	CC4S	2'b0	RW	捕获/比较 4 选择 该位定义通道的方向(输入/输出)，及输入脚的选择： b00: CC4 通道被配置为输出 b01: CC4 通道被配置为输入，IC4 映射在 TI4 上 b10: CC4 通道被配置为输入，IC4 映射在 TI3 上 b11: CC4 通道被配置为输入，IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择) 注：CC4S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC4E=0)才是可写的
7	OC3CE	1'b0	RW	输出比较 1 清 0 使能 参考 CC1CE

6:4	OC3M	3'b0	RW	输出比较 3 模式 参考 OC1M
3	OC3PE	1'b0	RW	输出比较 3 预装载使能 参考 OC1PE
2	OC3FE	1'b0	RW	输出比较 3 快速使能 参考 OC1FE
1:0	CC3S	2'b0	RW	捕获/比较 3 选择 这 2 位定义通道的方向(输入/输出),及输入脚的选择: b00：CC3 通道被配置为输出; b01：CC3 通道被配置为输入，IC3 映射在 TI3 上; b10：CC3 通道被配置为输入，IC3 映射在 TI4 上; b11：CC3 通道被配置为输入，IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择)。 注：CC3S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC3E=0)才是可写的。

输入捕获模式：

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:12	IC4F	4'b0	RW	输入捕获 4 滤波器 参考 IC1F
11:10	IC4PSC	2'b0	RW	输入/捕获 4 预分频器 参考 IC1PSC
9:8	CC4S	2'b0	RW	捕获/比较 4 选择 这 2 位定义通道的方向(输入/输出),及输入脚的选择: b00: CC4 通道被配置为输出 b01: CC4 通道被配置为输入，IC4 映射在 TI4 上 b10: CC4 通道被配置为输入，IC4 映射在 TI3 上 b11: CC4 通道被配置为输入，IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择) 注：CC4S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC4E=0)才是可写的。
7:4	IC3F	4'b0	RW	输入捕获 3 滤波器 参考 IC1F
3:2	IC3PSC	2'b0	RW	输入/捕获 3 预分频器 参考 IC1PSC
1:0	CC3S	2'b0	RW	捕获/比较 3 选择 这 2 位定义通道的方向(输入/输出),及输入脚的

				选择: b00: CC3 通道被配置为输出; b01: CC3 通道被配置为输入，IC3 映射在 TI3 上; b10: CC3 通道被配置为输入，IC3 映射在 TI4 上; b11: CC3 通道被配置为输入，IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择)。 注：CC3S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC3E=0)才是可写的。

15.4.2.9. 捕获/比较使能寄存器（TIMERx_CCER)

Width	Name	Reset	Property	Description
31:14	Reserved	-	-	-
13	CC4P	1'b0	RW	输入/捕获 4 输出极性 参考 CC1P 的描述。
12	CC4E	1'b0	RW	输入/捕获 4 输出使能 参考 CC1E 的描述。
11	CC3NP	1'b0	RW	输入/捕获 3 输出极性 参考 CC1NP 的描述。
10	Reserved	-	-	-
9	CC3P	1'b0	RW	输入/捕获 3 输出极性 参考 CC1P 的描述。
8	CC3E	1'b0	RW	输入/捕获 3 输出使能 参考 CC1E 的描述。
7	CC2NP	1'b0	RW	输入/捕获 2 输出极性 参考 CC1NP 的描述。
6	Reserved	-	-	-
5	CC2P	1'b0	RW	输入/捕获 2 输出极性 参考 CC1P 的描述。
4	CC2E	1'b0	RW	输入/捕获 2 输出使能 参考 CC1E 的描述。
3	CC1NP	1'b0	RW	输入/捕获 1 输出极性 CC1 配置为输出: CC1NP 需要配置为 0. CC1 配置为输入: 用于与 CC1P 一起决定 TI1FP1/TI2FP1 极性，参考 CC1P 的描述。
2	Reserved	-	-	-
1	CC1P	1'b0	RW	输入/捕获 1 输出极性 **CC1 通道配置为输出:** 0: OC1 高电平有效; 1: OC1 低电平有效。 **CC1 通道配置为输入:** CC1NP 和 CC1P 一起决定 TI1FP1 和 TI2FP1 极性

				b00: 不反相，上升沿。 TIxFP1 上升沿有效（捕获，复位模式，外部时钟，触发模式），TIxFP1 不反向（门控模式，正交编码模式） b01: 反相，下降沿。 TIxFP1 下降沿有效（捕获，复位模式，外部时钟，触发模式），TIxFP1 反向（门控模式，正交编码模式） b10: reserved b11:不反向，上升沿和下降沿都有效。该配置不能用于正交编码模式。
0	CC1E	1'b0	RW	输入/捕获 1 输出使能 **CC1 通道配置为输出：** 0: 关闭 - OC1 禁止输出，因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 1: 开启 - OC1 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 **CC1 通道配置为输入：** 该位决定了计数器的值是否能捕获入 TIMERx_CCR1 寄存器。 0: 捕获禁止； 0: 捕获使能。

标准 OCx 通道的输出控制位

CCxE 位	OCx 输出状态
0	禁止输出(OCx=0，OCx_EN=0)
1	OCx = OCxREF + 极性，OCx_EN=1

注:

1.引脚连接到互补的 OCx 和 OCxN 通道的外部 I/O 引脚的状态，取决于 OCx 和 OCxN 通道状态和 GPIO 控制寄存器。

2.当 SYSCON1[31]为 1 时，只要通道配置成输出模式，则 IO 一直由 timer 驱动。

当 SYSCON1[31]为 0 时，上表中 OCx_EN/OCxN_EN=1 时，IO 由 timer 驱动，OCx_EN/OCxN_EN=0 时，IO 为浮空态。

15.4.2.10. 计数器（TIMERx_CNT)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	CNT	16'b0	RW	计数器的值

15.4.2.11. 预分频器（TIMERx_PSC)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	PSC	16'b0	RW	预分频器的值 计数器的时钟频率(CK_CNT)等于 fCK_PSC /(PSC[15:0]+1)。 PSC 包含了每次当更新事件产生时，装入当前预分频器寄存器的值；更新事件包括计数器被

				TIM_EGR 的 UG 位清 0 或被工作在复位模式的从控制器清 0。

15.4.2.12. 自动重装载寄存器（TIMERx_ARR）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	ARR	16'hffff	RW	自动重装载的值 ARR 包含了将要装载入实际的自动重装载寄存器的值。 当自动重装载的值为空时，计数器不工作。

15.4.2.13. 捕获/比较寄存器 1（TIMERx_CCR1）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	CCR1	16'hffff	RW	捕获/比较通道 1 的值 **若 CC1 通道配置为输出：** CCR1 包含了装入当前捕获/比较 1 寄存器的值(预装载值)。 如果在 TIMERx_CCMR1 寄存器(OC1PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 1 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMERx_CNT 的比较，并在 OC1 端口上产生输出信号。 **若 CC1 通道配置为输入：** CCR1 包含了由上一次输入捕获 1 事件(IC1)传输的计数器值。 注：当配置为捕获模式时，该寄存器变成只读。

15.4.2.14. 捕获/比较寄存器 2（TIMERx_CCR2）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	CCR2	16'hffff	RW	捕获/比较通道 2 的值 **若 CC2 通道配置为输出：** CCR2 包含了装入当前捕获/比较 2 寄存器的值(预装载值)。 如果在 TIMERx_CCMR2 寄存器(OC2PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 2 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMERx_CNT 的比较，并在 OC2 端口上产生输出信号。 **若 CC2 通道配置为输入：** CCR2 包含了由上一次输入捕获 2 事件(IC2)传

				输的计数器值。 注：当配置为捕获模式时，该寄存器变成只读。

15.4.2.15. 捕获/比较寄存器 3（TIMERx_CCR3）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	CCR3	16'hffff	RW	捕获/比较通道 3 的值 **若 CC3 通道配置为输出：** CCR3 包含了装入当前捕获/比较 3 寄存器的值(预装载值)。 如果在 TIMERx_CCMR3 寄存器(OC3PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 3 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMERx_CNT 的比较，并在 OC3 端口上产生输出信号。 **若 CC3 通道配置为输入：** CCR3 包含了由上一次输入捕获 3 事件(IC3)传输的计数器值。 注：当配置为捕获模式时，该寄存器变成只读。

15.4.2.16. 捕获/比较寄存器 4（TIMERx_CCR4）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	CCR4	16'hffff	RW	捕获/比较通道 4 的值 **若 CC4 通道配置为输出：** CCR4 包含了装入当前捕获/比较 4 寄存器的值(预装载值)。 如果在 TIMERx_CCMR4 寄存器(OC4PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 4 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMERx_CNT 的比较，并在 OC4 端口上产生输出信号。 **若 CC4 通道配置为输入：** CCR4 包含了由上一次输入捕获 4 事件(IC4)传输的计数器值。 注：当配置为捕获模式时，该寄存器变成只读。

16. 通用定时器（TIMER4/5/6）

16.1. 模块介绍

通用定时器定时器（TIMER4/5/6）由一个 16 位的自动装载计数器组成，它由一个可编程的预分频器驱动。它适合多种用途，包含测量输入信号的脉冲宽度(输入捕获)，或者产生输出波形(输出比较、PWM、嵌入死区时间的互补 PWM 等)。

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

通用定时器(TIMER4/5/6)是完全独立的，它们不共享任何资源。

16.2. 功能特点

- 16 位向上、向下、向上/下自动装载计数器
- 16 位可编程（可以实时修改）预分频器，计数器时钟频率的分频系数为 1～65535 之间的任意数值
- 多达 2 个独立通道：
  - 输入捕获
  - 输出比较
  - PWM 生成（边缘或中间对齐模式）
  - 单脉冲模式输出
- 死区时间可编程的互补输出（只有 channel 1 支持）
- 使用外部信号控制定时器和定时器互联的同步电路
- 允许在指定数目的计数器周期之后更新定时器寄存器的重复计数器
- 刹车输入信号可以将定时器输出信号置于复位状态或者一个已知状态
- 如下事件发生时产生中断/DMA：
  - 更新：计数器向上溢出/向下溢出，计数器初始化(通过软件或者内部/外部触发)
  - 触发事件(计数器启动、停止、初始化或者由内部/外部触发计数)
  - 输入捕获
  - 输出比较
  - 刹车信号输入

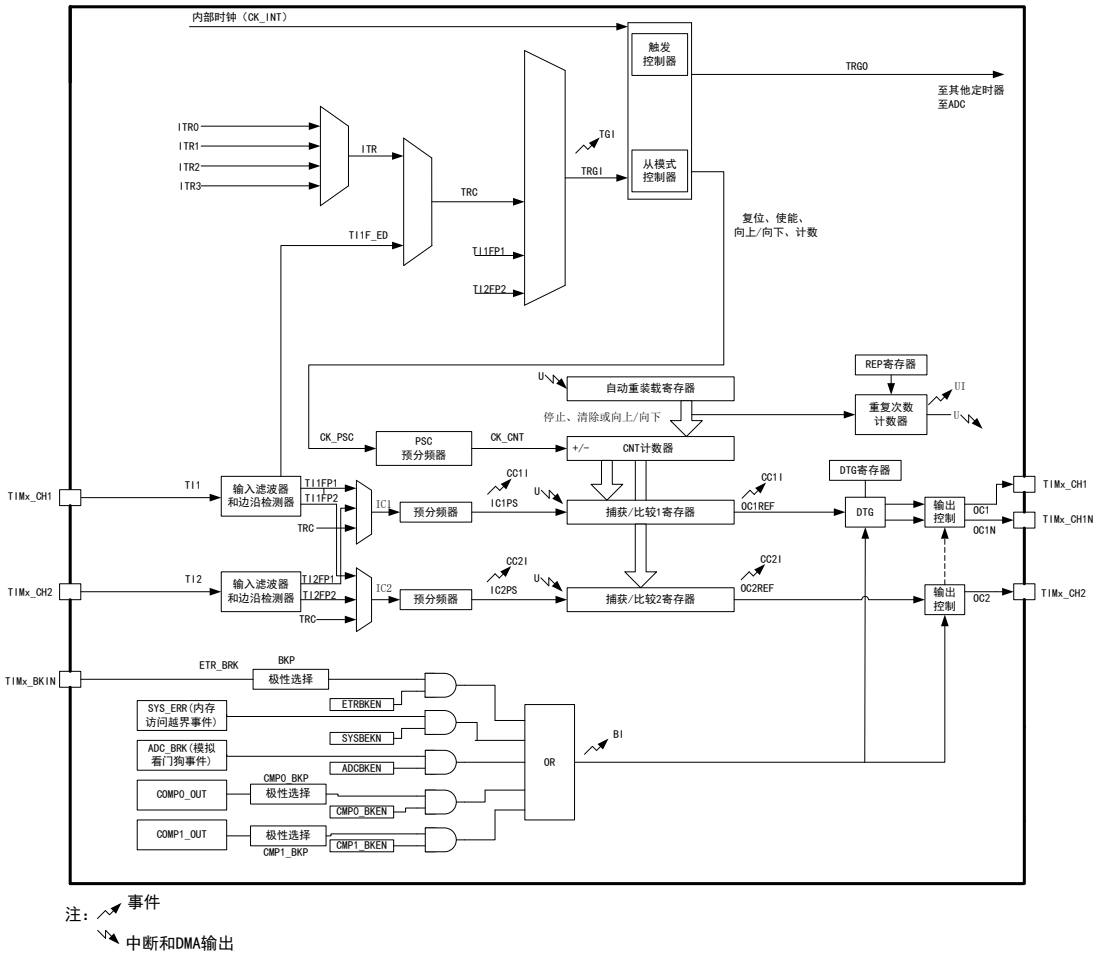


图 16-1

16.3. 功能说明

16.3.1. 时基单元

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

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

时基单元包含：

- 计数器寄存器(TIMERx_CNT)
- 预分频器寄存器 (TIMERx_PSC)
- 自动装载寄存器 (TIMERx_ARR)
- 重复次数寄存器 (TIMERx_RCR)

自动装载寄存器是预先装载的，写或读自动重装载寄存器将访问预装载寄存器。根据在 TIMERx_CR1 寄存器中的自动装载预装载使能位(ARPE)的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件(向下计数时的下溢条件)并当 TIMERx_CR1 寄存器中的 UDIS 位等于 0 时，产生更新事件。更新事件也可以由软件产生。随后会详细描述每一种配置下更新事件的产生。

计数器由预分频器的时钟输出 CK_CNT 驱动，仅当设置了计数器 TIMERx_CR1 寄存器中的计数器使能位(CEN)时，CK_CNT 才有效。(更多有关使能计数器的细节，请参见控制器的从模式描述)。

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

预分频器描述

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是基于一个(在 TIMERx_PSC 寄存器中的)16 位寄存器控制的 16 位计数器。因为这个控制寄存器带有缓冲器，它能够在运行时被改变，

新的预分频器的参数在下一次更新事件到来时被采用。

以下两图给出了在预分频器运行时，更改计数器参数的例子。

当预分频器的参数从 1 变到 2 时，计数器的时序图：

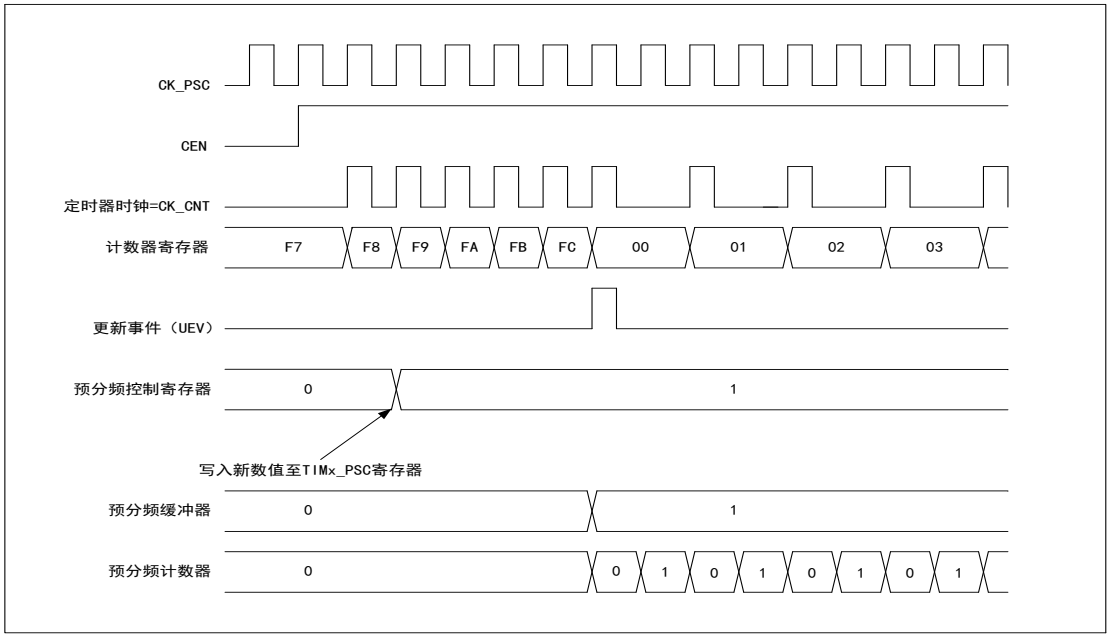


图 16-2

当预分频器的参数从 1 变到 4 时，计数器的时序图：

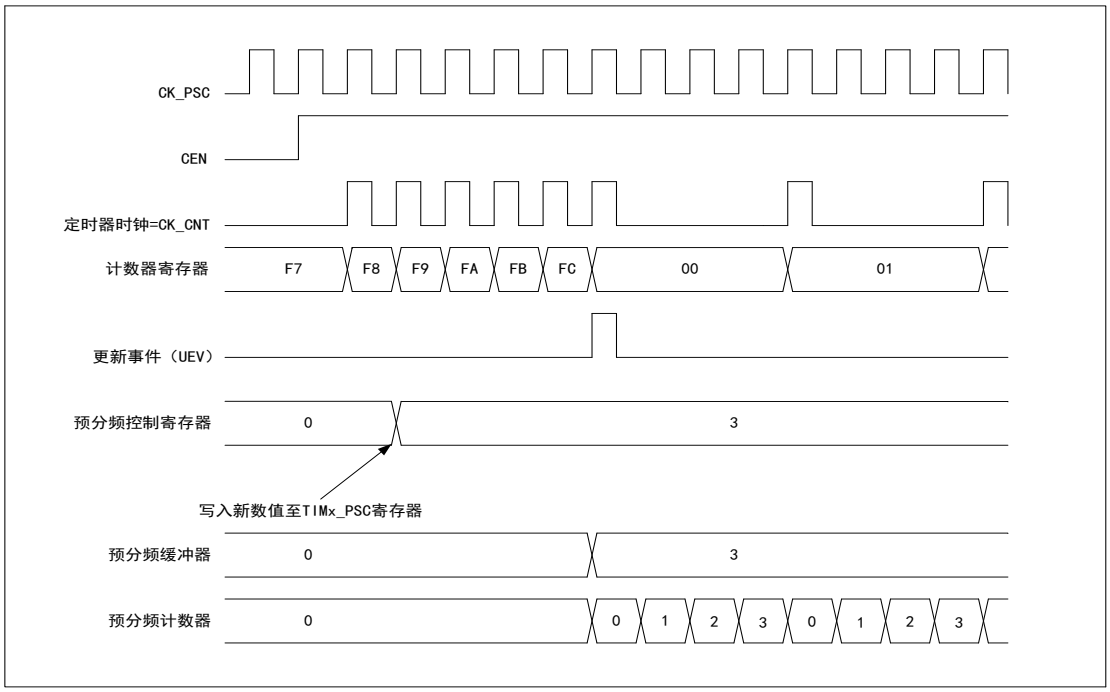


图 16-3

16.3.2. 计数器模式

向上计数模式

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

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

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

设置 TIMERx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清‘0’之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清‘0’，同时预分频器的计数也被清 0(但预分频器的数值不变)。此外，如果设置了 TIMERx_CR1 寄存器中的 URS 位(选择更新请求)，设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志(即不产生中断或 DMA 请求)。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

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

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

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

计数器时序图，预分频参数为 1

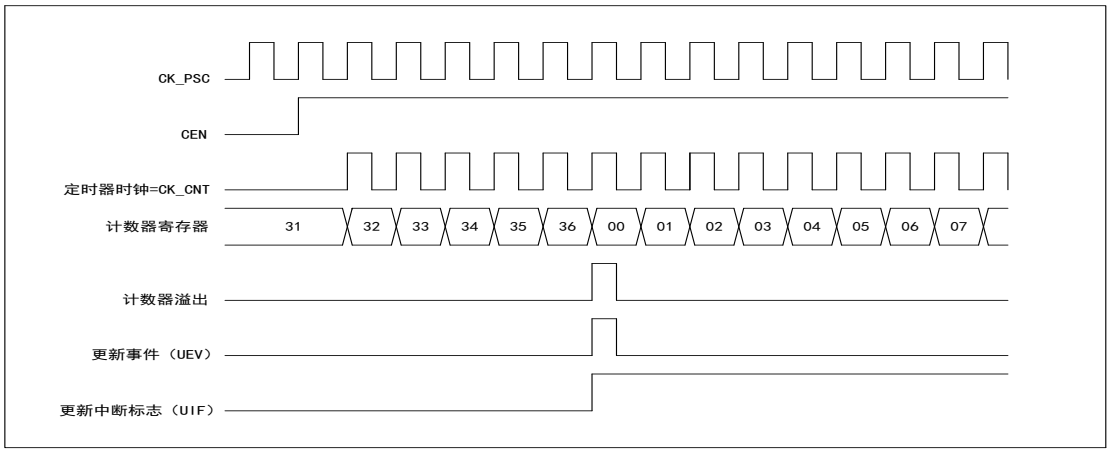


图 16-4

计数器时序图，预分频参数为 2

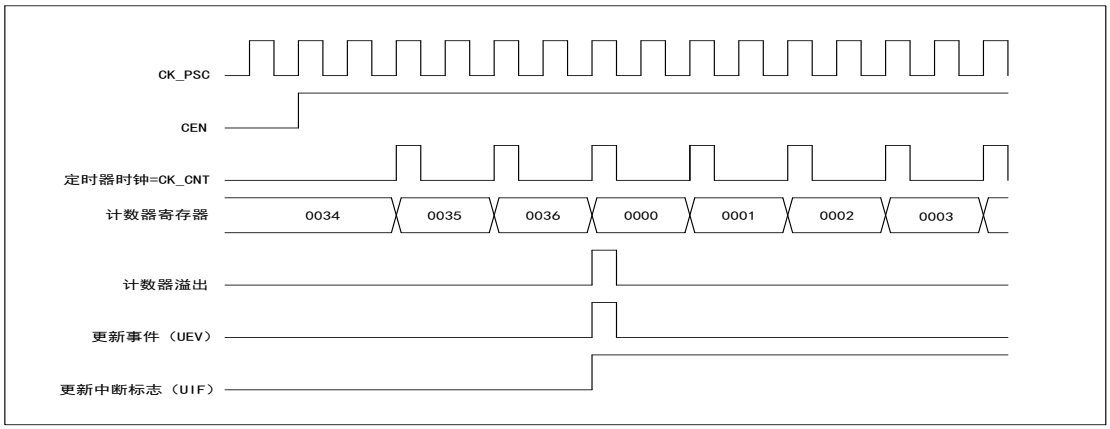


图 16-5

计数器时序图，预分频参数为 4

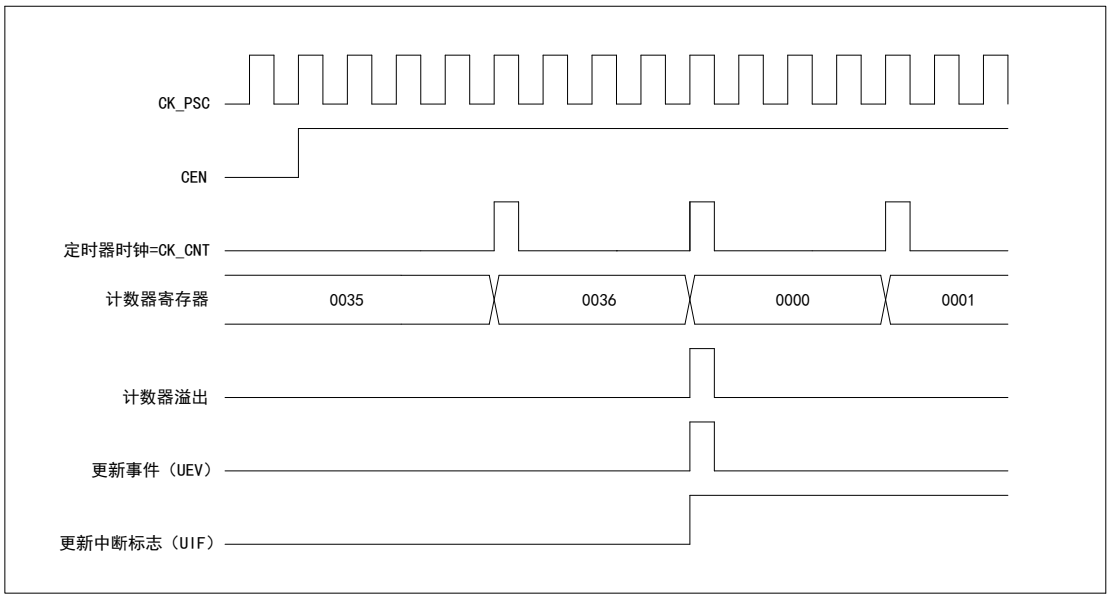


图 16-6

计数器时序图，预分频参数为 N

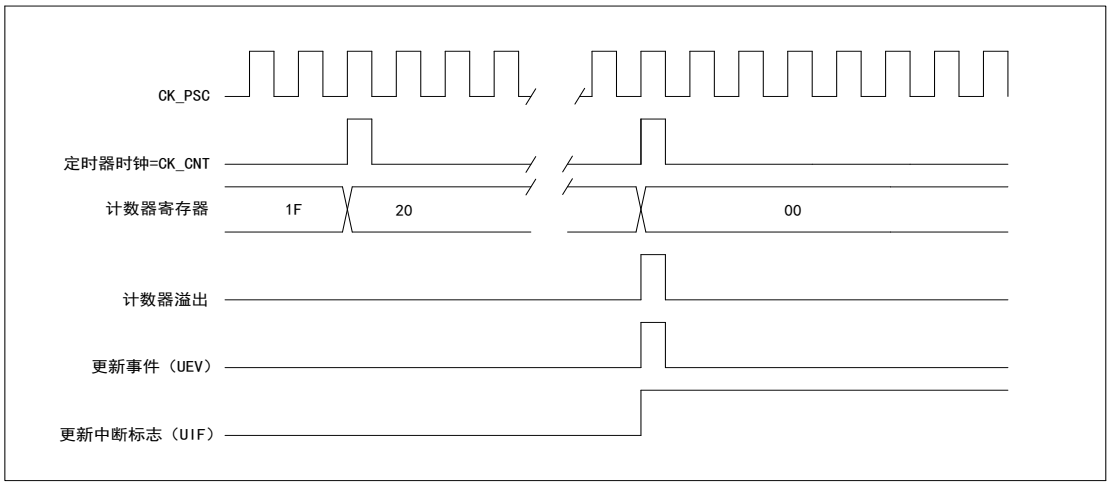


图 16-7

计数器时序图，当 ARPE=0 时的更新事件(TIMERx_ARR 没有预装入)

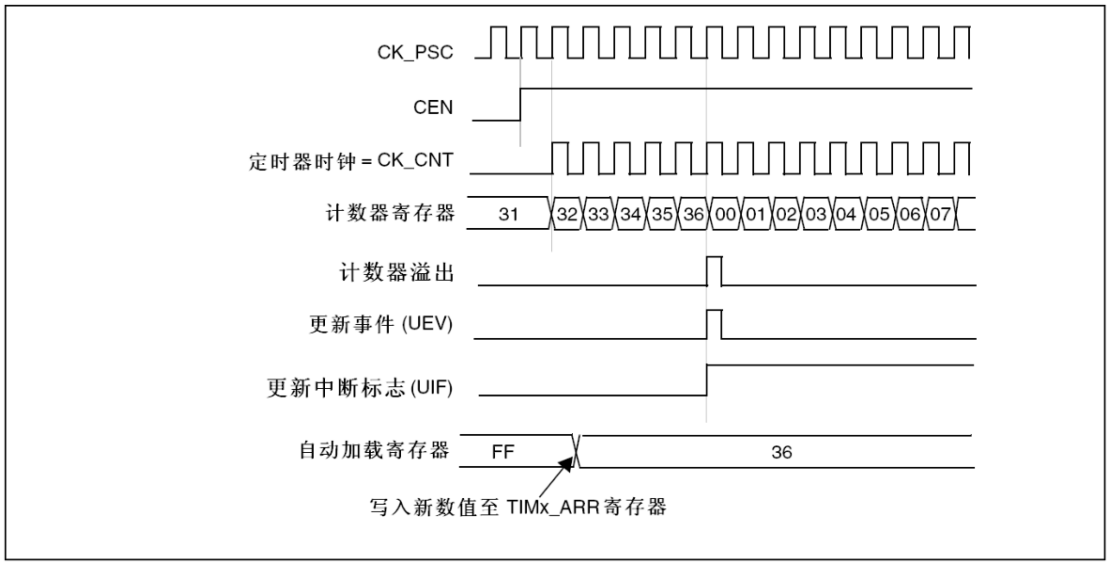


图 16-8

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

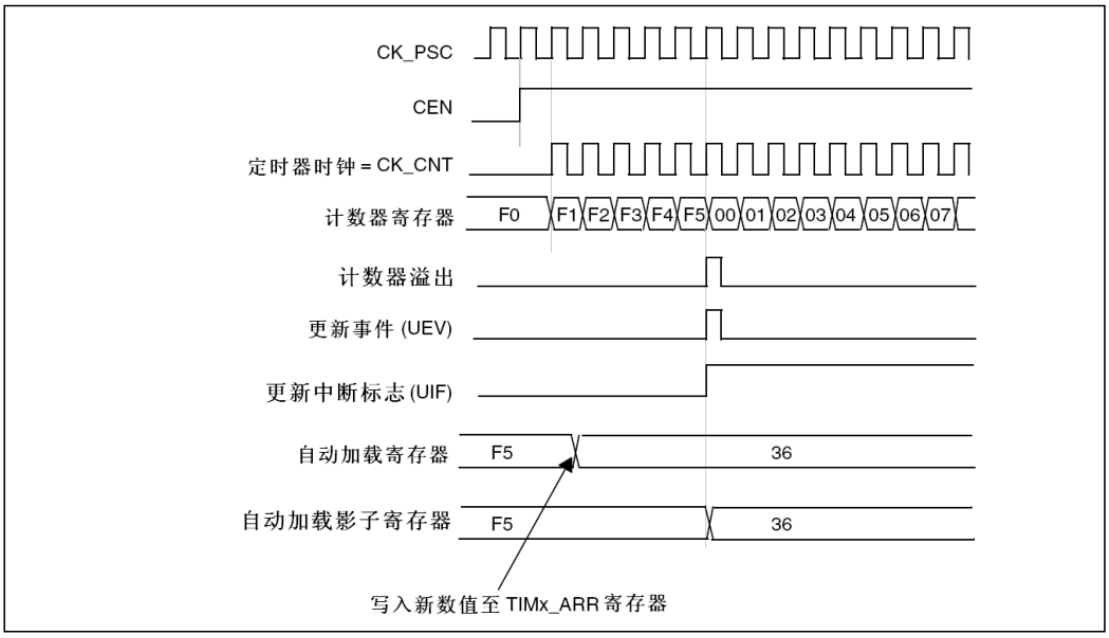


图 16-9

向下计数模式

在向下模式中，计数器从自动装入的值(TIMERx_ARR 计数器的值)开始向下计数到 0，然后从自动装入的值重新开始并且产生一个计数器向下溢出事件。

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

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

设置 TIMERx_CR1 寄存器的 UDIS 位可以禁止 UEV 事件。这样可以避免向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，并且预分频器的计数器重新从 0 开始(但预分频系数不变)。

此外，如果设置了 TIMERx_CR1 寄存器中的 URS 位(选择更新请求) ，设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志(因此不产生中断和 DMA 请求),这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

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

- 重复计数器被重新加载为 TIMERx_RCR 寄存器的内容。
- 预分频器的缓冲区被置入预装载寄存器的值(TIMERx_PSC 寄存器的内容)。
- 当前的自动加载寄存器被更新为预装载值(TIMERx_ARR 寄存器中的内容)。

注：自动装载在计数器重载入之前被更新，因此下一个周期将是预期的值。

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

计数器时序图，预分频参数为 1

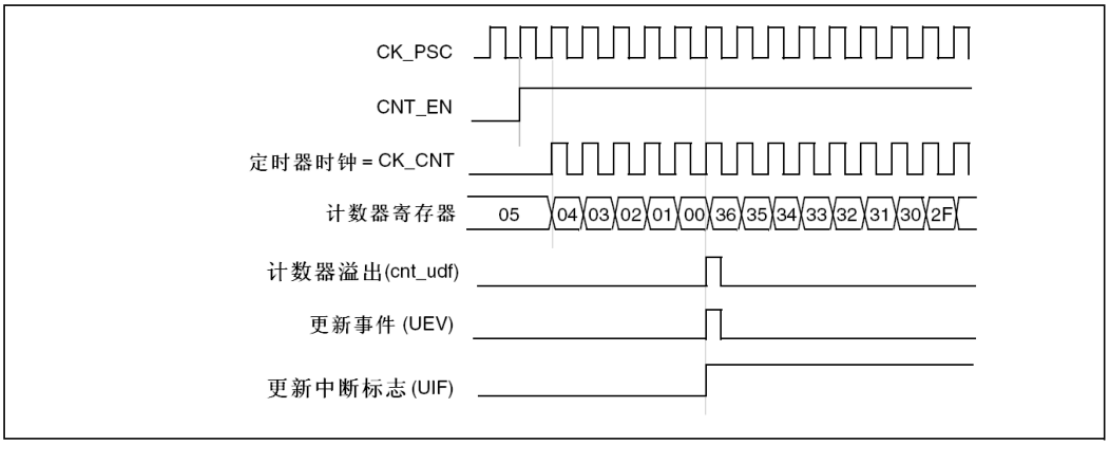


图 16-10

计数器时序图，预分频参数为 2

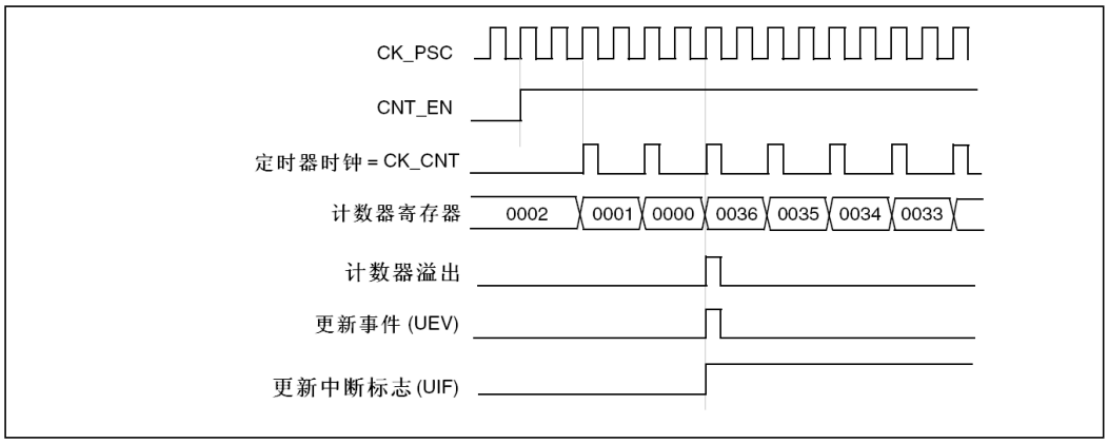


图 16-11

计数器时序图，预分频参数为 4

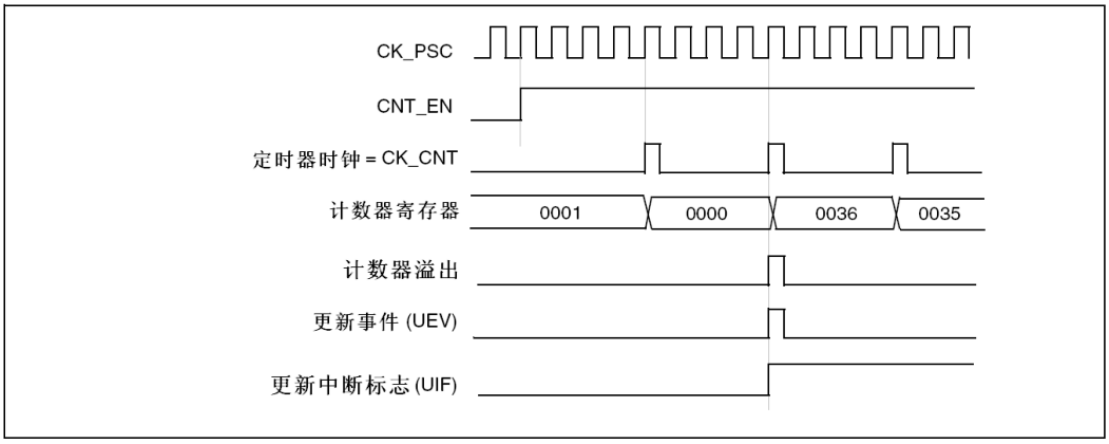


图 16-12

计数器时序图，预分频参数为 N

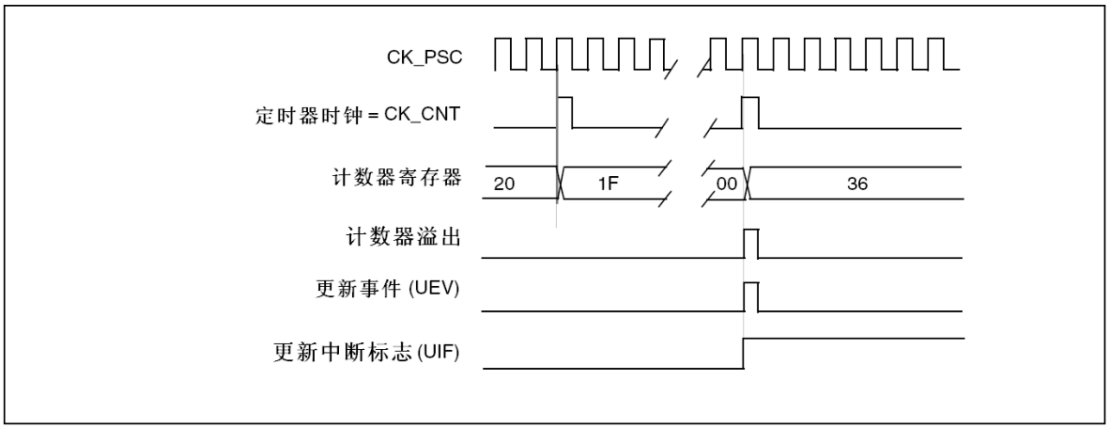


图 16-13

计数器时序图，当没有使用重复计数器时的更新事件

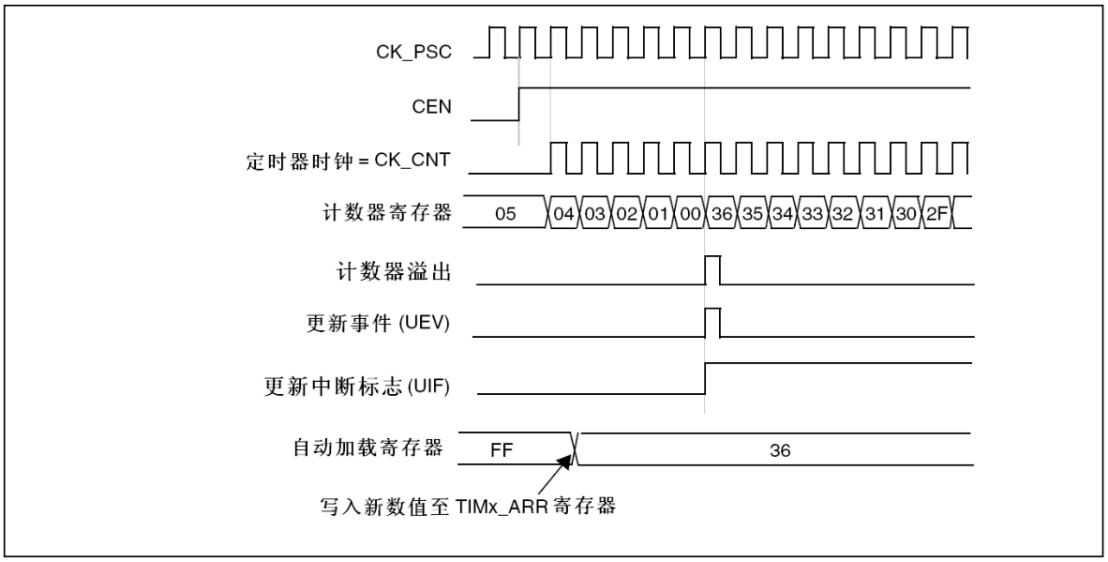


图 16-14

中央对齐模式（向上/向下计数）

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

在此模式下，不能写入 TIMERx_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。

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

设置 TIMERx_CR1 寄存器中的 UDIS 位可以禁止 UEV 事件。这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。

此外，如果设置了 TIMERx_CR1 寄存器中的 URS 位(选择更新请求) ，设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志(因此不产生中断和 DMA 请求),这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

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

- 重复计数器被重新加载为 TIMERx_RCR 寄存器的内容。
- 预分频器的缓冲区被置入预装载寄存器的值(TIMERx_PSC 寄存器的内容)。
- 当前的自动加载寄存器被更新为预装载值(TIMERx_ARR 寄存器中的内容)。注：如果因为计数器溢出而产生更新，自动重装载将在计数器重载入之前被更新，因此下一个周期将是预期的值(计数器被装载为新的值)。

以下是一些计数器在不同时钟频率下的操作的例子：

计数器时序图，预分频参数为 1，TIMERx_ARR=0x6

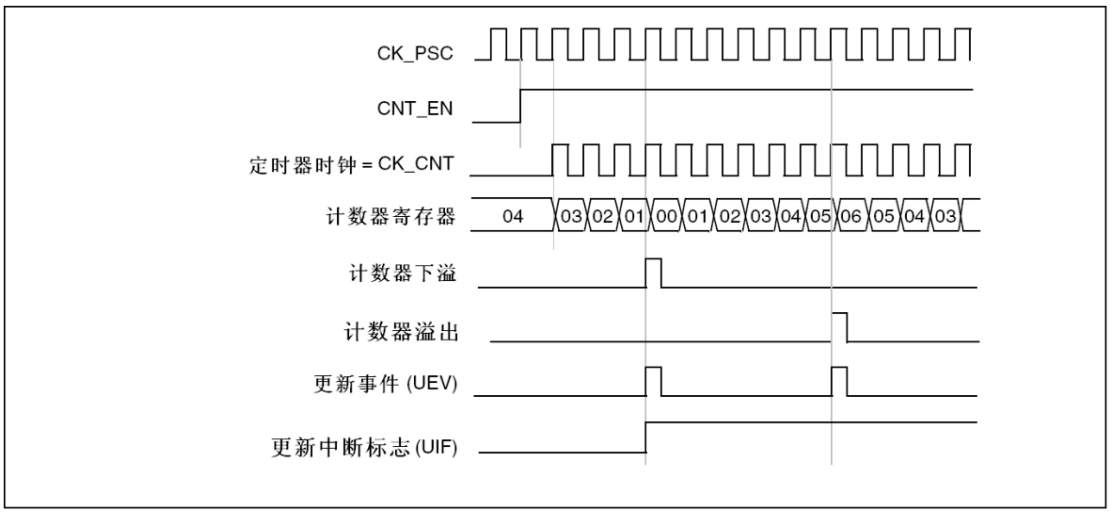


图 16-15

计数器时序图，预分频参数为 2

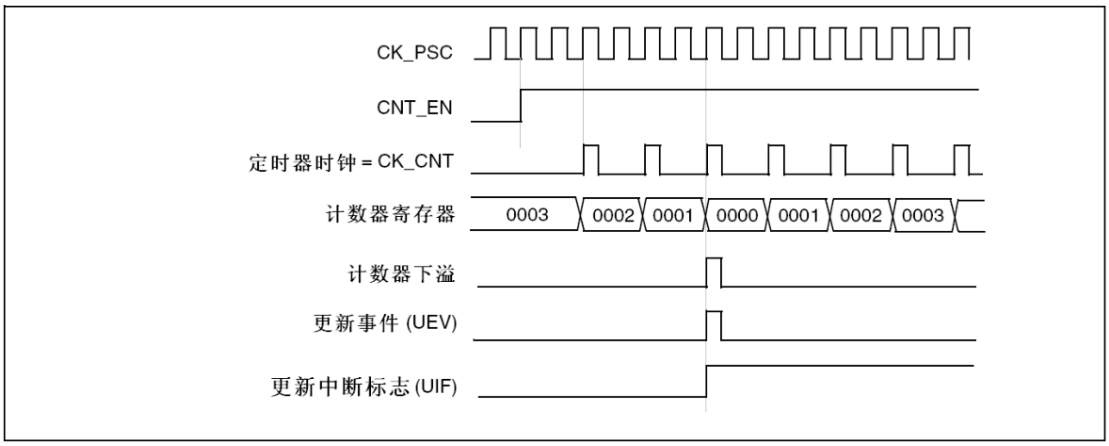


图 16-16

计数器时序图，预分频参数为 4，TIMERx_ARR=0x36

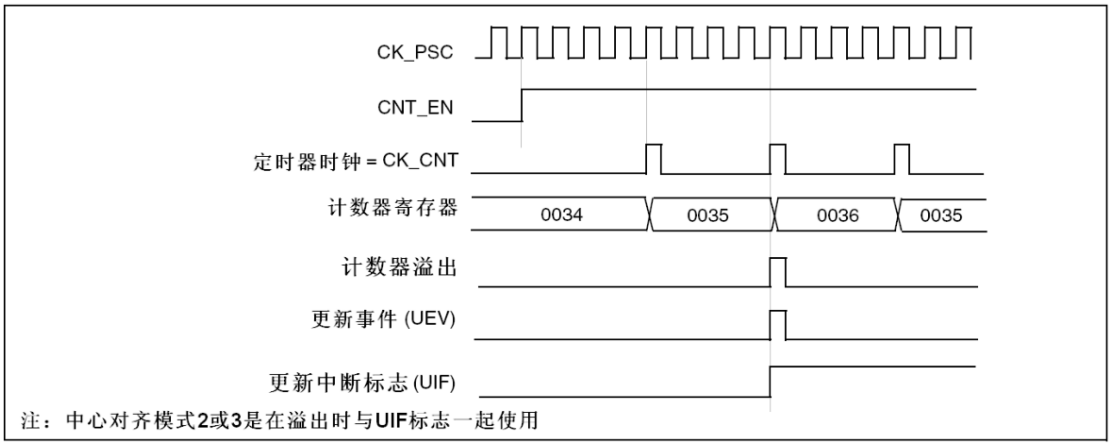


图 16-17

计数器时序图，预分频参数为 N

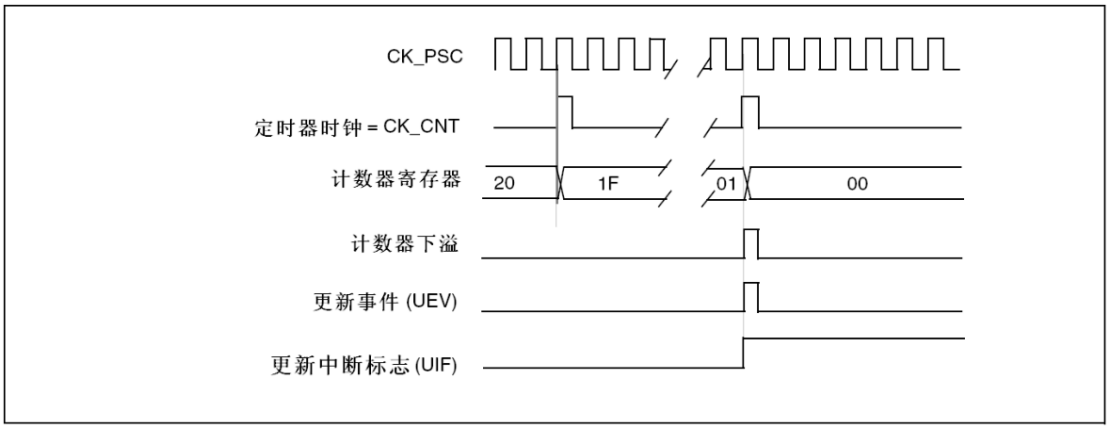


图 16-18

计数器时序图，ARPE=1 时的更新事件(计数器下溢)

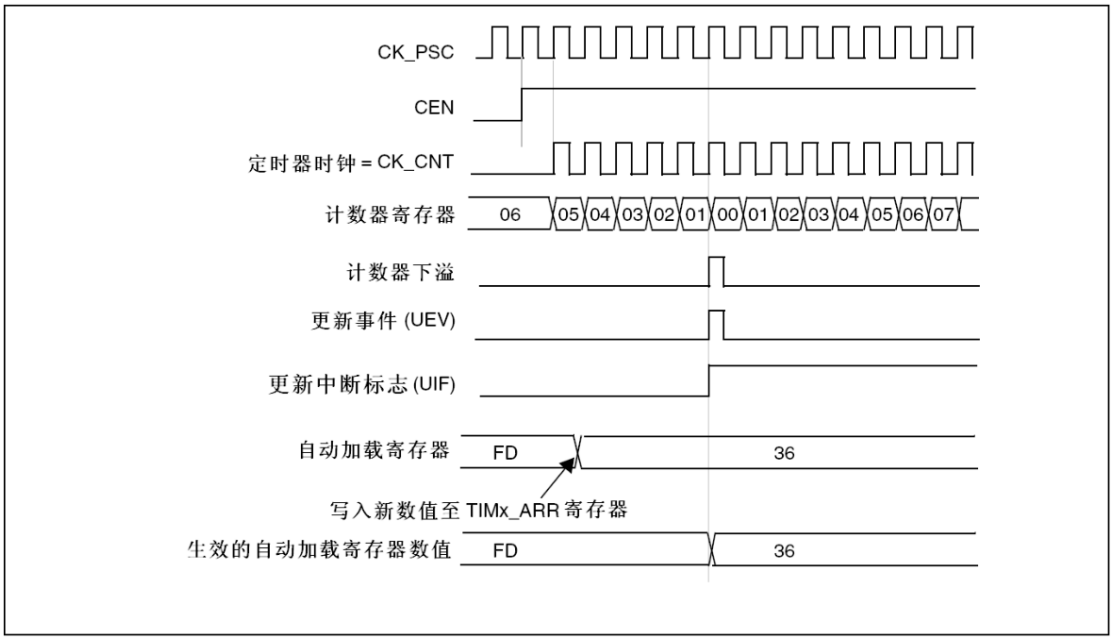


图 16-19

计数器时序图，ARPE=1 时的更新事件(计数器溢出)

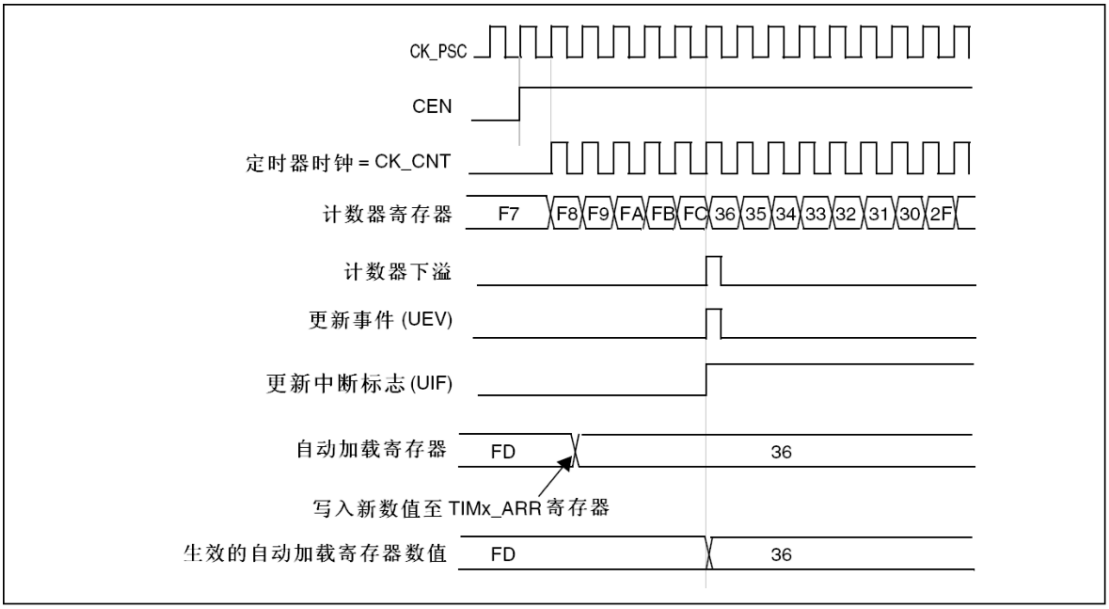


图 16-20

16.3.3. 重复计数器

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

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

重复计数器在下述任一条件成立时递减：

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

重复计数器是自动加载的，重复速率是由 TIMERx_RCR 寄存器的值定义(参看下图 17-21)。当更新事件由软件产生(通过设置 TIMERx_EGR 中的 UG 位)或者通过硬件的从模式控制器产生，则无论重复计数器的值是多少，立即发生更新事件，并且 TIMERx_RCR 寄存器中的内容被重载入到重复计数器。

不同模式下更新速率的例子，及 TIMERx_RCR 的寄存器设置

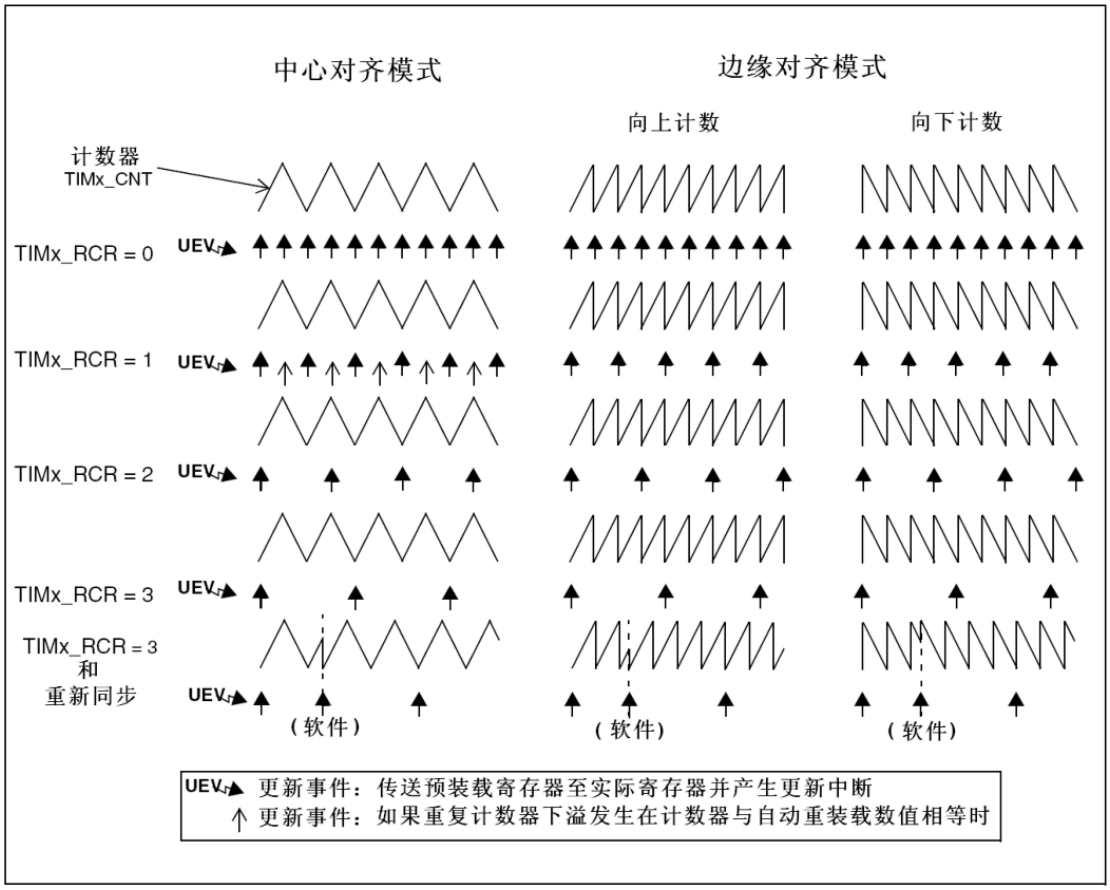


图 16-21

16.3.4. 时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟（CK_INT）
- 外部时钟模式 1：外部输入引脚
- 内部触发输入(ITRx)：使用一个定时器作为另一个定时器的预分频器。

内部时钟源 内部时钟源(CK_INT)

如果禁止了从模式控制器(SMS=000)，则 CEN、DIR(TIMERx_CR1 寄存器)和 UG 位(TIMERx_EGR

寄存器)是事实上的控制位，并且只能被软件修改(UG 位仍被自动清除)。只要 CEN 位被写成“1”,预分频器的时钟就由内部时钟 CK_INT 提供。
下图显示控制电路和向上计数器在一般模式下，不带预分频器时的操作。
一般模式下的控制电路，预分频参数为 1

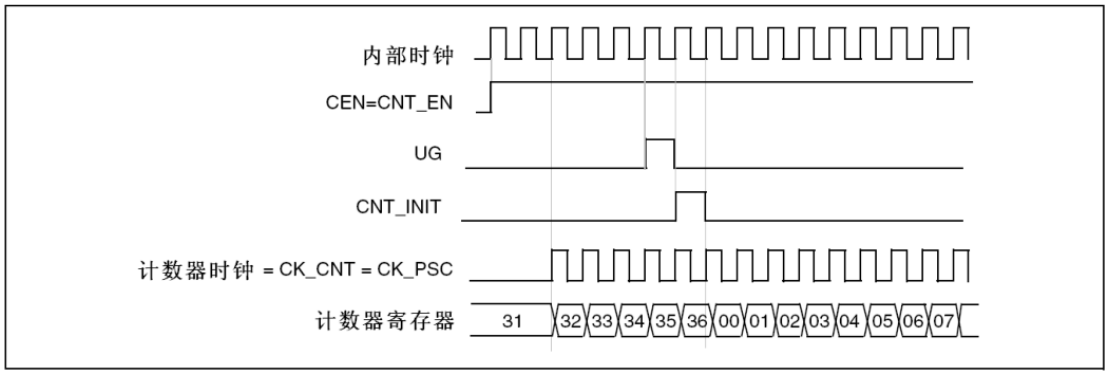


图 16-22

外部时钟源模式 1

当 TIMERx_SMCR 寄存器的 SMS=111 时，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。
TI2 外部时钟连接例子

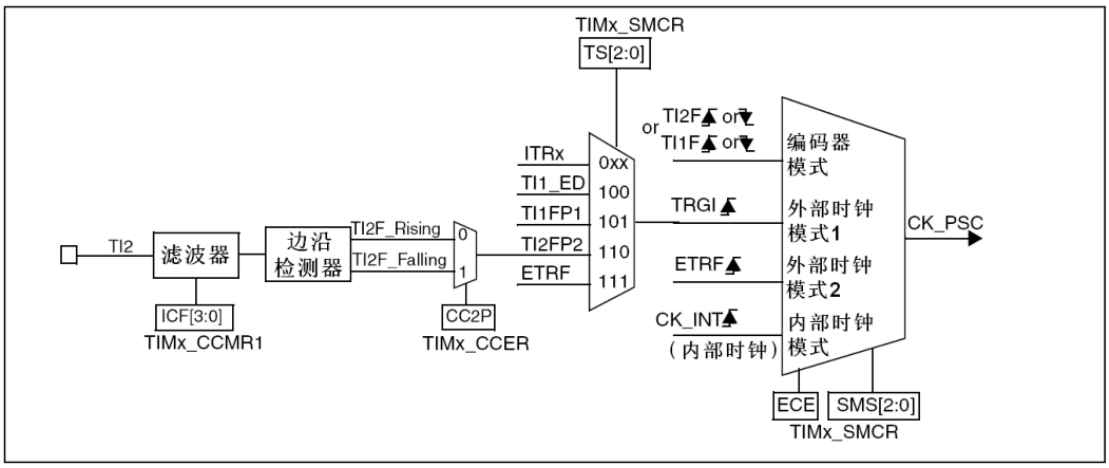


图 16-23

例如，要配置向上计数器在 T12 输入端的上升沿计数，使用下列步骤:
配置 TIMERx_CCMR1 寄存器 CC2S=01，配置通道 2 检测 TI2 输入的上升沿
配置 TIMERx_CCMR1 寄存器的 IC2F[3:0]，选择输入滤波器带宽(如果不需要滤波器，保持 IC2F=0000)
配置 TIMERx_CCER 寄存器的 CC2P=0，选定上升沿极性
配置 TIMERx_SMCR 寄存器的 SMS=111，选择定时器外部时钟模式 1
配置 TIMERx_SMCR 寄存器中的 TS=110，选定 TI2 作为触发输入源
设置 TIMERx_CR1 寄存器的 CEN=1，启动计数器
注： 捕获预分频器不用作触发，所以不需要对它进行配置
当上升沿出现在 TI2，计数器计数一次，且 TIF 标志被设置。
在 TI2 的上升沿和计数器实际时钟之间的延时，取决于在 TI2 输入端的重新同步电路。
外部时钟模式 1 下的控制电路

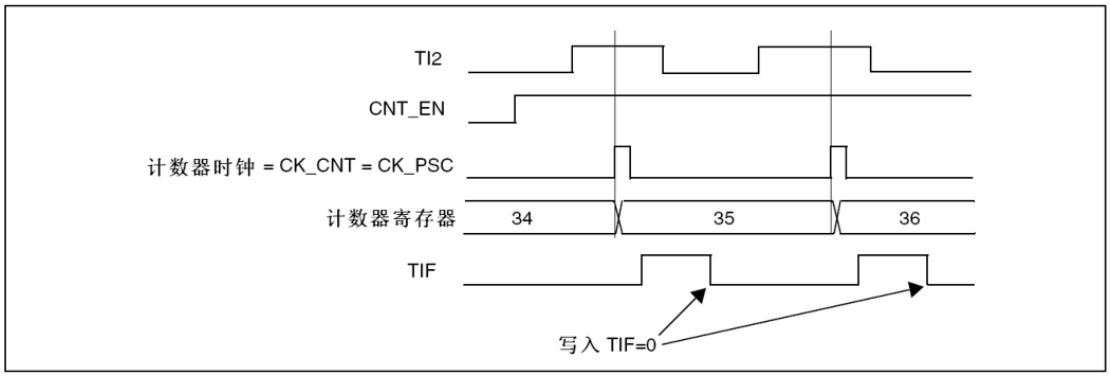


图 16-24

16.3.5. 捕获/比较通道

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

以下三图是一个捕获/比较通道概览。

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

捕获/比较通道(如：通道 1 输入部分)

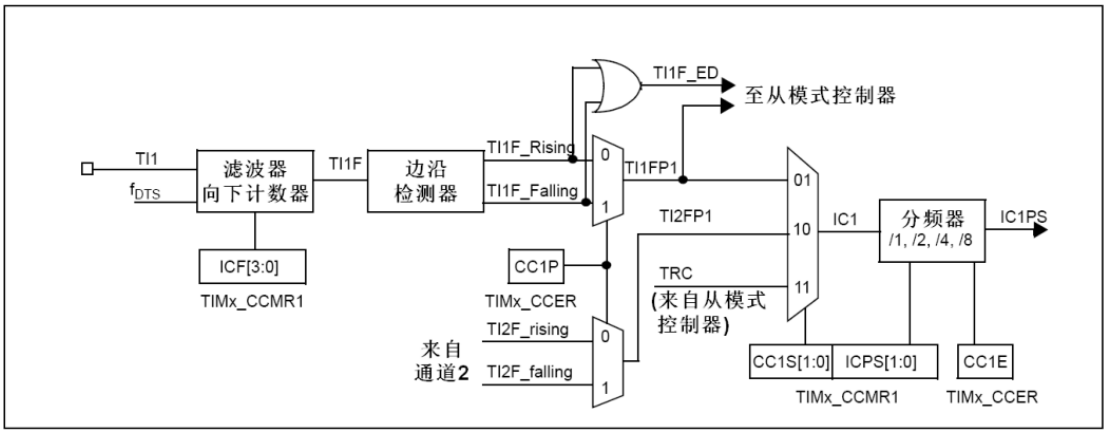


图 16-25

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

捕获/比较通道 1 的主电路

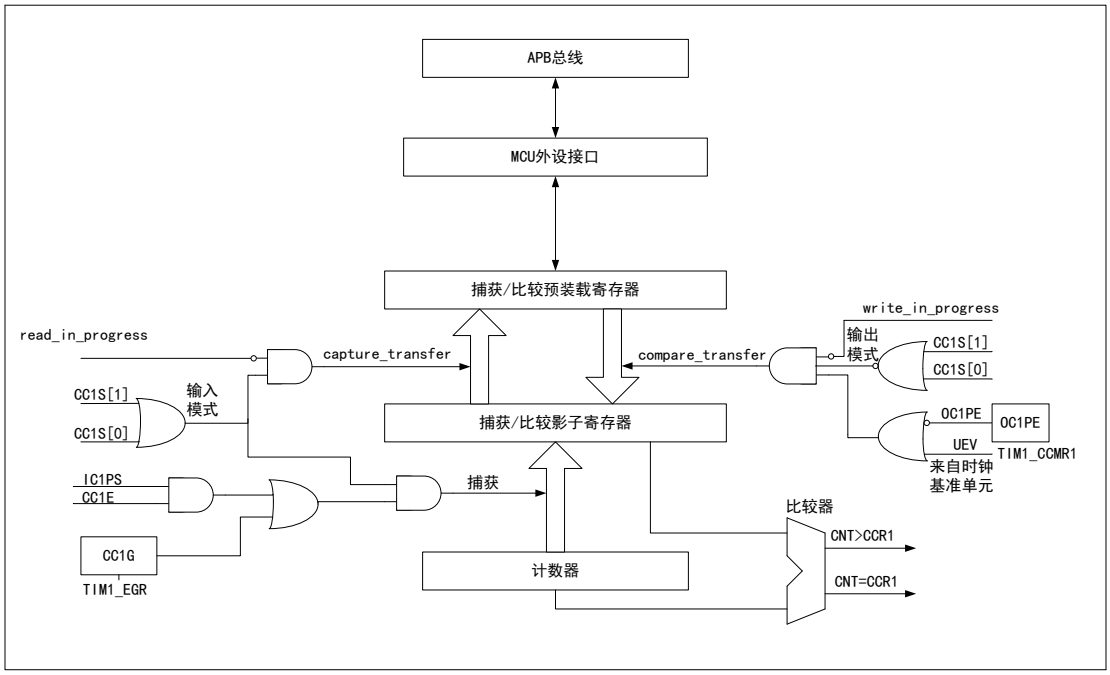


图 16-26

捕获/比较通道的输出部分(通道 1)

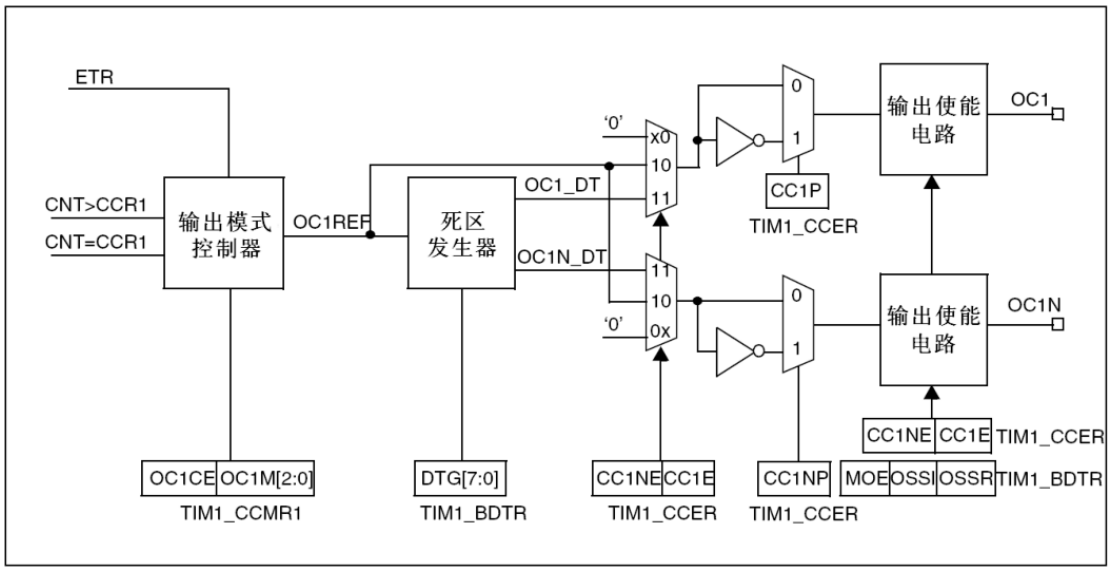


图 16-27

捕获/比较通道的输出部分(通道 2)

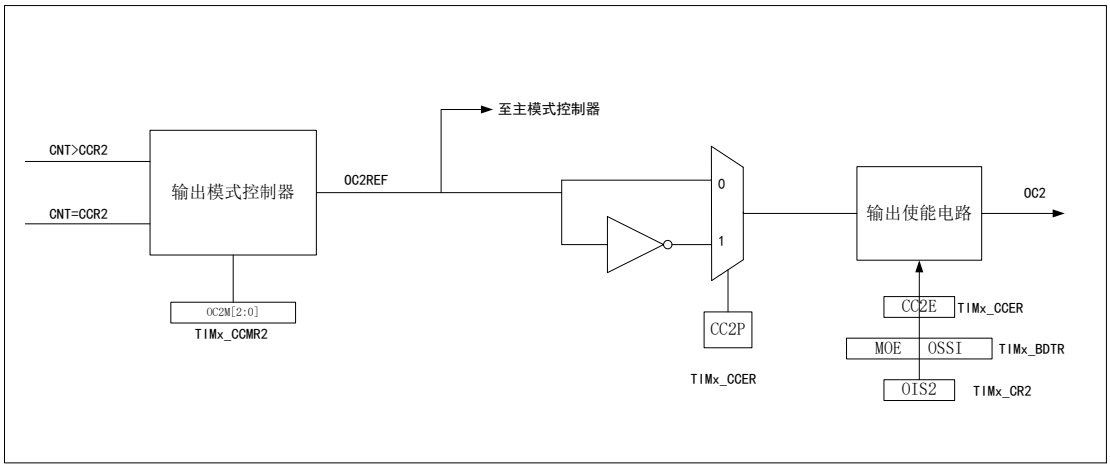


图 16-28

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。读写过程仅操作预装载寄存器。
在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。
在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

16.3.6. 输入捕获模式

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

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIMERx_CCR1 寄存器中，步骤如下：

- 选择有效输入端：TIMERx_CCR1 必须连接到 TI1 输入，所以写入 TIMERx_CCMR1 寄存器中的 CC1S=01，只要 CC1S 不为“00”，通道被配置为输入，并且 TIMERx_CCR1 寄存器变为只读。
- 根据输入信号的特点，配置输入滤波器为所需的带宽(即输入为 TIx 时，输入滤波器控制位是 TIMERx_CCMRx 寄存器中的 ICxF 位)。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期；因此我们可以(以 f DTS 频率)连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIMERx_CCMR1 寄存器中写入 IC1F=0011。
- 选择 TI1 通道的有效转换边沿，在 TIMERx_CCER 寄存器中写入 CC1P=0(上升沿)。
- 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止(写 TIMERx_CCMR1 寄存器的 IC1PS=00)。
- 设置 TIMERx_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
- 如果需要，通过设置 TIMERx_DIER 寄存器中的 CC1IE 位允许相关中断请求，通过设置 TIMERx_DIER 寄存器中的 CC1DE 位允许 DMA 请求。

当发生一个输入捕获时：

- 产生有效的电平转换时，计数器的值被传送到 TIMERx_CCR1 寄存器。
- CC1IF 标志被设置(中断标志)。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置 1。
- 如设置了 CC1IE 位，则会产生一个中断。
- 如设置了 CC1DE 位，则还会产生一个 DMA 请求。

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

注： 设置 TIMERx_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断和 / 或 DMA 请求。

16.3.7. PWM 输入模式

该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

- 两个 ICx 信号被映射至同一个 TIx 输入。

- 这 2 个 ICx 信号为边沿有效，但是极性相反。
- 其中一个 TIxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。
- 例如，你需要测量输入到 TI1 上的 PWM 信号的长度(TIMERx_CCR1 寄存器)和占空比(TIMERx_CCR2 寄存器)，具体步骤如下(取决于 CK_INT 的频率和预分频器的值)
- 选择 TIMERx_CCR1 的有效输入：置 TIMERx_CCMR1 寄存器的 CC1S=01(选中 TI1)。
- 选择 TI1FP1 的有效极性(用来捕获数据到 TIMERx_CCR1 中和清除计数器)：置 CC1P=0(上升沿有效)。
- 选择 TIMERx_CCR2 的有效输入：置 TIMERx_CCMR1 寄存器的 CC2S=10(选中 TI1)。
- 选择 TI1FP2 的有效极性(捕获数据到 TIMERx_CCR2)：置 CC2P=1(下降沿有效)。
- 选择有效的触发输入信号：置 TIMERx_SMCR 寄存器中的 TS=101(选择 TI1FP1)。
- 配置从模式控制器为复位模式：置 TIMERx_SMCR 中的 SMS=100。
- 使能捕获：置 TIMERx_CCER 寄存器中 CC1E=1 且 CC2E=1。

PWM 输入模式时序

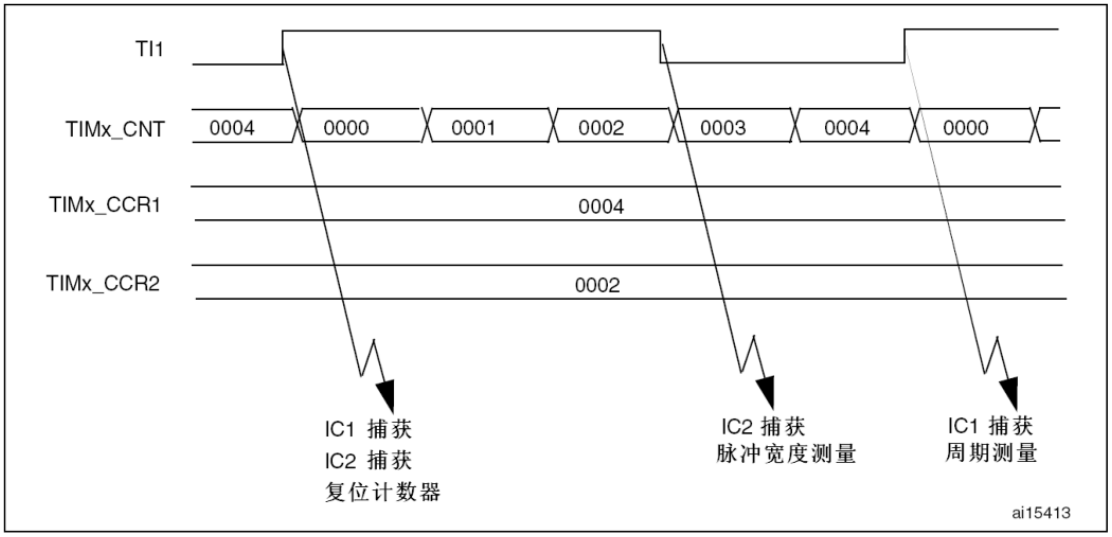


图 16-29

因为只有 TI1FP1 和 TI2FP2 连到了从模式控制器，所以 PWM 输入模式只能使用 TIMERx_CH1 /TIMERx_CH2 信号。

16.3.8. 强制输出模式

在输出模式(TIMERx_CCMRx 寄存器中 CCxS=00)下，输出比较信号(OCxREF 和相应的 OCx/OCxN)能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。

置 TIMERx_CCMRx 寄存器中相应的 OCxM=101，即可强置输出比较信号(OCxREF/OCx)为有效状态。这样 OCxREF 被强置为高电平(OCxREF 始终为高电平有效)，同时 OCx 得到 CCxP 极性相反的信号。

例如：CCxP=0(OCx 高电平有效)，则 OCx 被强置为高电平。

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

该模式下，在 TIMERx_CCRx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断和 DMA 请求。这将会在下面的输出比较模式一节中介绍。

16.3.9. 输出比较模式

此项功能是用来控制一个输出波形，或者指示一段给定的的时间已经到时。

当计数器与捕获/比较寄存器的内容相同时，输出比较功能做如下操作：

- 将输出比较模式(TIMERx_CCMRx 寄存器中的 OCxM 位)和输出极性(TIMERx_CCER 寄存器中的 CCxP 位)定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平(OCxM=000)、被设置成有效电平(OCxM=001)、被设置成无效电平(OCxM=010)或进行翻转(OCxM=011)。
- 设置中断状态寄存器中的标志位(TIMERx_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽(TIMERx_DIER 寄存器中的 CCxIE 位)，则产生一个中断。
- 若设置了相应的使能位(TIMERx_DIER 寄存器中的 CCxDE 位，TIMERx_CR2 寄存器中的 CCDS 位选择 DMA 请求功能)，则产生一个 DMA 请求。

TIMERx_CCMRx 中的 OCxPE 位选择 TIMERx_CCRx 寄存器是否需要使用预装载寄存器。
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。
同步的精度可以达到计数器的一个计数周期。输出比较模式(在单脉冲模式下)也能用来输出一个单脉冲。
输出比较模式的配置步骤：

1. 选择计数器时钟(内部，外部，预分频器)。
2. 将相应的数据写入 TIMERx_ARR 和 TIMERx_CCRx 寄存器中。
3. 如果要产生一个中断请求，设置 CCxIE 位。
4. 选择输出模式，例如：
   - 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚，设置 OCxM=011
   - 置 OCxPE = 0 禁用预装载寄存器
   - 置 CCxP = 0 选择极性为高电平有效
   - 置 CCxE = 1 使能输出
5. 设置 TIMERx_CR1 寄存器的 CEN 位启动计数器

TIMERx_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器(OCxPE= ‘0’ ，否则 TIMERx_CCRx 的影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。
输出比较模式，翻转 OC1

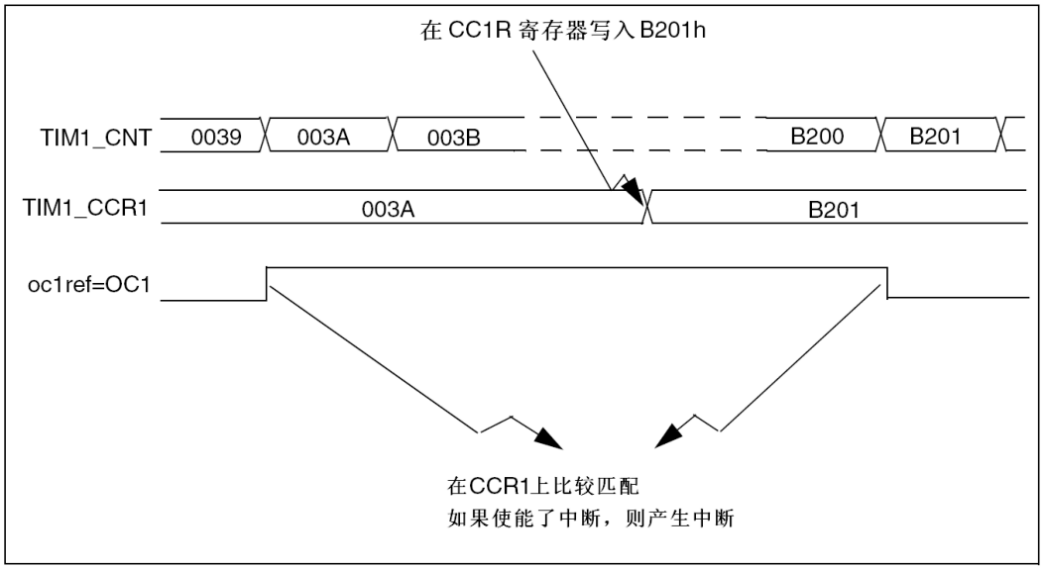


图 16-30

16.3.10. PWM 模式

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

在 TIMERx_CCMRx 寄存器中的 OCxM 位写入“110”(PWM 模式 1)或“111”(PWM 模式 2)，能够独立地设置每个 OCx 输出通道产生一路 PWM。必须通过设置 TIMERx_CCMRx 寄存器的 OCxPE 位使能相应的预装载寄存器，最后还要设置 TIMERx_CR1 寄存器的 ARPE 位，(在向上计数或中心对称模式中)使能自动重装载的预装载寄存器。

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

OCx 的极性可以通过软件在 TIMERx_CCER 寄存器中的 CCxP 位设置，它可以设置为高电平有效或低电平有效。OCx 的输出使能通过(TIMERx_CCER 和 TIMERx_BDTR 寄存器中)CCxE、CCxNE、MOE、OSSI 和 OSSR 位的组合控制。详见 TIMERx_CCER 寄存器的描述 。

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

根据 TIMERx_CR1 寄存器中 CMS 位的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的

PWM 信号。

PWM 边沿对齐模式

• 向上计数的配置

当 TIMERx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。参看下面是一个 PWM 模式 1 的例子。当 TIMERx_CNT<TIMERx_CCRx 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIMERx_CCRx 中的比较值大于自动重装载值(TIMERx_ARR)，则 OCxREF 保持为“1”。如果比较值为 0，则 OCxREF 保持为“0”。下图为 TIMERx_ARR=8 时边沿对齐的 PWM 波形实例。

边沿对齐的 PWM 波形(ARR=8)

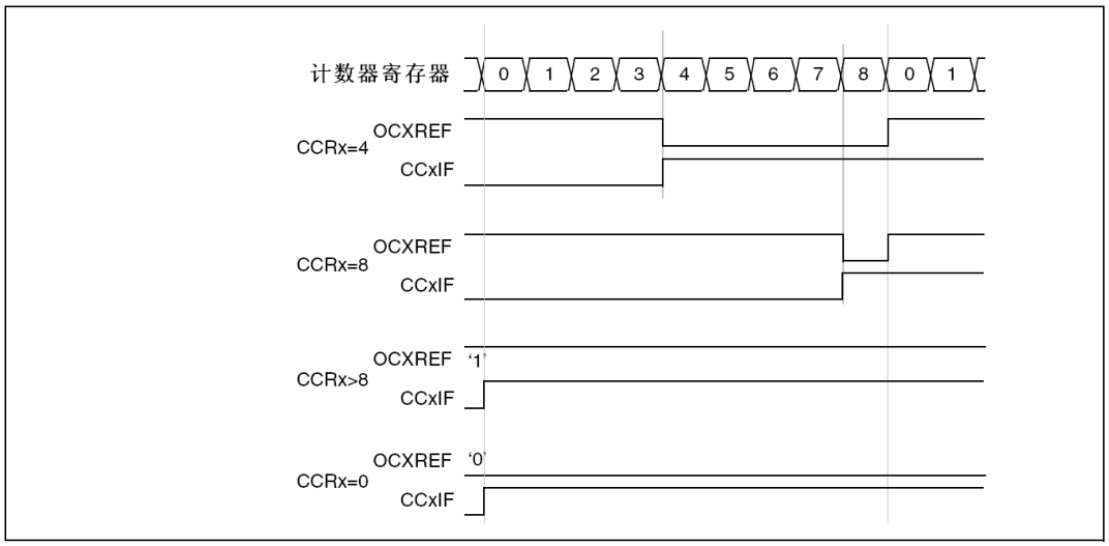


图 16-31

• 向下计数的配置

当 TIMERx_CR1 寄存器的 DIR 位为高时执行向下计数。参看在 PWM 模式 1，当 TIMERx_CNT>TIMERx_CCRx 时参考信号 OCxREF 为低，否则为高。如果 TIMERx_CCRx 中的比较值大于 TIMERx_ARR 中的自动重装载值，则 OCxREF 保持为’1’。该模式下不能产生 0%的 PWM 波形。

PWM 中央对齐模式

当 TIMERx_CR1 寄存器中的 CMS 位不为“00”时为中央对齐模式(所有其他的配置对 OCxREF/OCx 信号都有相同的作用)。根据不同的 CMS 位设置，比较标志可以在计数器向上计数时被置 1、在计数器向下计数时被置 1、或在计数器向上和向下计数时被置 1。TIMERx_CR1 寄存器中的计数方向位(DIR)由硬件更新，不要用软件修改它。下图给出了一些中央对齐的 PWM 波形的例子：

• TIMERx_ARR=8
• PWM 模式 1
• TIMERx_CR1 寄存器的 CMS=01，在中央对齐模式 1 下，当计数器向下计数时设置比较标志。
• 中央对齐的 PWM 波形(APR=8)

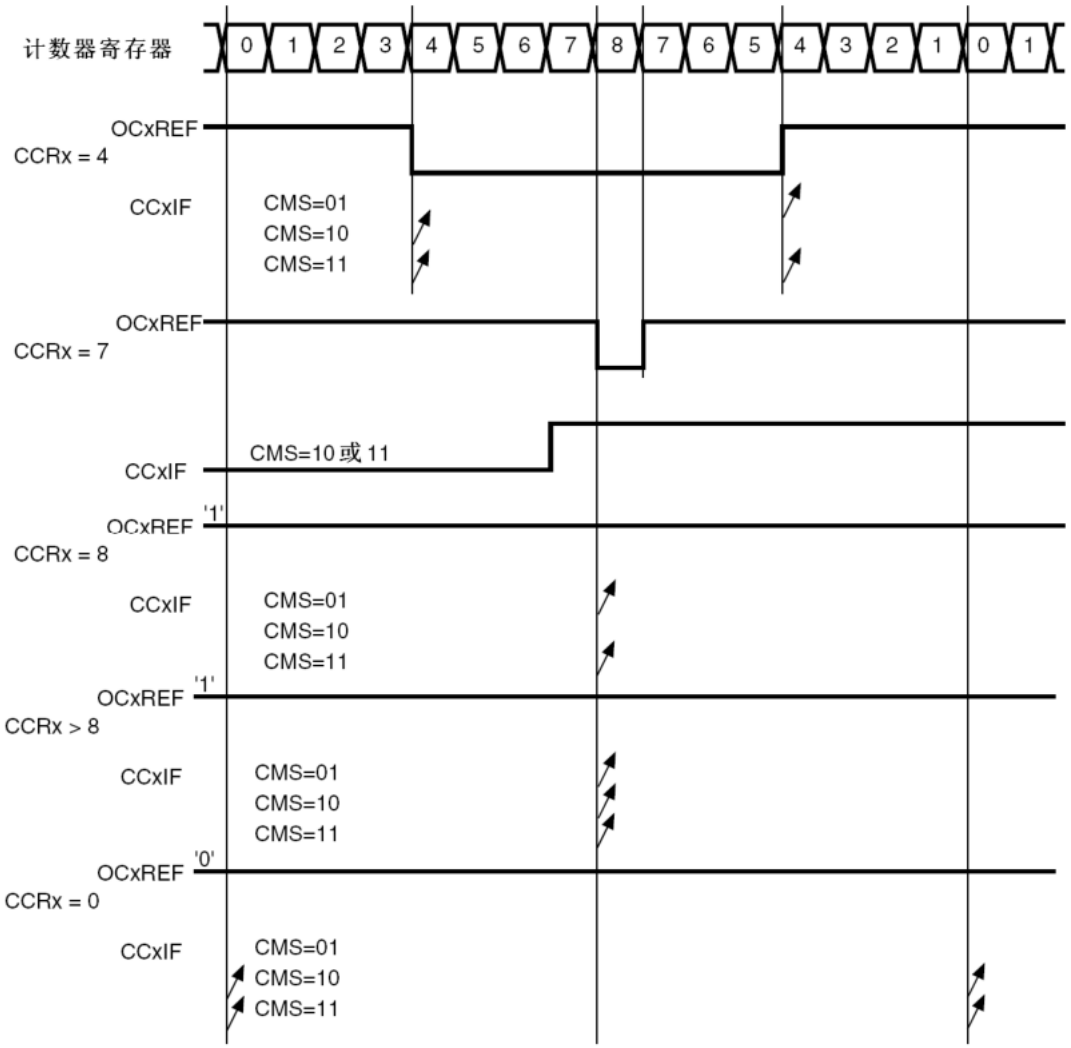


图 16-32

使用中央对齐模式的提示：

- 进入中央对齐模式时，使用当前的向上/向下计数配置；这就意味着计数器向上还是向下计数取决于 TIMERx_CR1 寄存器中 DIR 位的当前值。此外，软件不能同时修改 DIR 和 CMS 位。
- 不推荐当运行在中央对齐模式时改写计数器，因为这会产生不可预知的结果。特别地：
  - 如果写入计数器的值大于自动重加载的值(TIMERx_CNT>TIMERx_ARR)，则方向不会被更新。例如，如果计数器正在向上计数，它就会继续向上计数。
  - 如果将 0 或者 TIMERx_ARR 的值写入计数器，方向被更新，但不产生更新事件 UEV。
- 使用中央对齐模式最保险的方法，就是在启动计数器之前产生一个软件更新(设置 TIMERx_EGR 位中的 UG 位)，并且不要在计数进行过程中修改计数器的值。

16.3.11. 互补输出和死区插入

高级控制定时器能够输出两路互补信号，并且能够管理输出的瞬时关断和接通。

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

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

互补信号 OCx 和 OCxN 通过下列控制位的组合进行控制：TIMERx_CCER 寄存器的 CCxE 和 CCxNE 位，TIMERx_BDTR 和 TIMERx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位，详见 表 15-4 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位 。特别的是，在转换到 IDLE 状态时(MOE 下降到 0)死区被激活。

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

有效：

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

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

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

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

带死区插入的互补输出

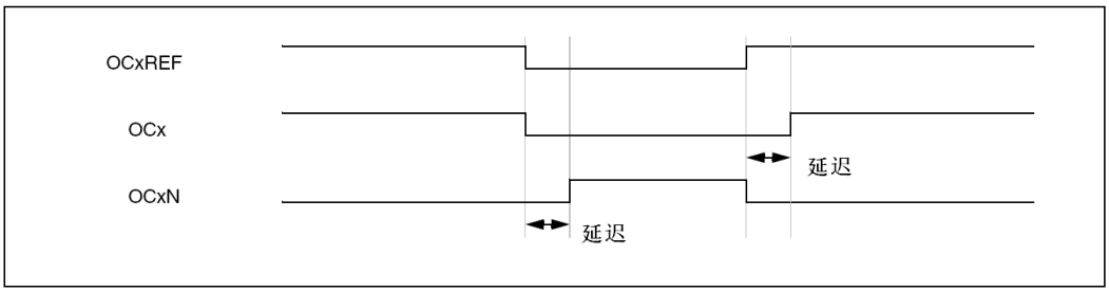


图 16-33

死区波形延迟大于负脉冲

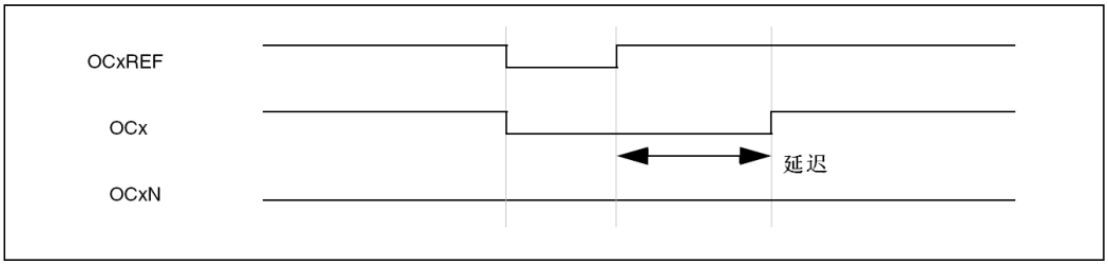


图 16-34

死区波形延迟大于正脉冲

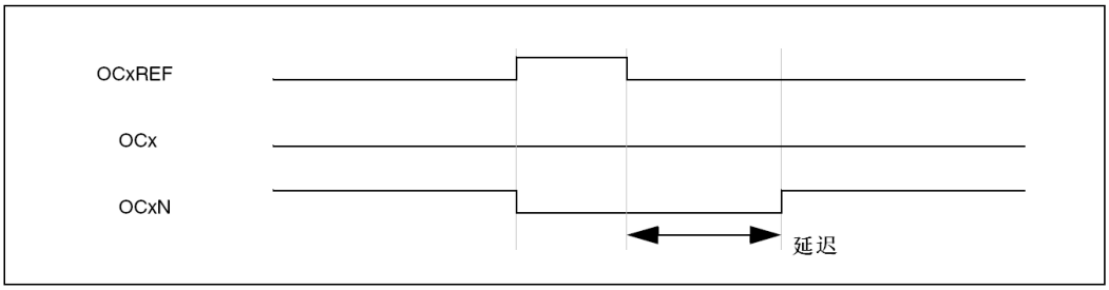


图 16-35

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

重定向 OCxREF 到 OCx 或 OCxN

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

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

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

16.3.12. 使用刹车功能

当使用刹车功能时，依据相应的控制位(TIMERx_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位，TIMERx_CR2 寄存器中的 OISx 和 OISxN 位)，输出使能信号和无效电平都会被修改。但无论何时，OCx 和 OCxN 输出不能在同一时间同时处于有效电平上。详见 表 15-4 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位 。

刹车源既可以是刹车输入引脚，又可以是一个 sram 访问越界事件。

系统复位后，刹车电路被禁止，MOE 位为低。设置 TIMERx_BDTR 寄存器中的 BKE 位可以使能刹车功能，刹车输入信号的极性可以通过配置同一个寄存器中的 BKP 位选择。BKE 和 BKP 可以同时被修改。

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

当发生刹车时(在刹车输入端出现选定的电平)，有下述动作：

- MOE 位被异步地清除，将输出置于无效状态、空闲状态或者复位状态(由 OSSI 位选择)。这个特性在 MCU 的振荡器关闭时依然有效。
- 一旦 MOE=0，每一个输出通道输出由 TIMERx_CR2 寄存器中的 OISx 位设定的电平。如果 OSSI=0，则定时器释放使能输出，否则使能输出始终为高。
- 当使用互补输出时：
  - 输出首先被置于复位状态即无效的状态(取决于极性)。这是异步操作，即使定时器没有时钟时，此功能也有效。
  - 如果定时器的时钟依然存在，死区生成器将会重新生效，在死区之后根据 OISx 和 OISxN 位指示的电平驱动输出端口。即使在这种情况下，OCx 和 OCxN 也不能被同时驱动到有效的电平。注，因为重新同步 MOE，死区时间比通常情况下长一些(大约 2 个 ck_tim 的时钟周期)。
  - 如果 OSSI=0，定时器释放使能输出，否则保持使能输出；或一旦 CCxE 与 CCxNE 之一变高时，使能输出变为高。
- 如果设置了 TIMERx_DIER 寄存器中的 BIE 位，当刹车状态标志(TIMERx_SR 寄存器中的 BIF 位)为“1”时，则产生一个中断。如果设置了 TIMERx_DIER 寄存器中的 BDE 位，则产生一个 DMA 请求。
- 如果设置了 TIMERx_BDTR 寄存器中的 AOE 位，在下一个更新事件 UEV 时 MOE 位被自动置位；例如，这可以用来进行整形。否则，MOE 始终保持低直到被再次置“1”；此时，这个特性可以被用在安全方面，你可以把刹车输入连到电源驱动的报警输出、热敏传感器或者其他安全器件上。

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

刹车由 BRK 输入产生，它的有效极性是可编程的，且由 TIMERx_BDTR 寄存器中的 BKE 位开启。除了刹车输入和输出管理，刹车电路中还实现了写保护以保证应用程序的安全。它允许用户冻结几个配置参数(死区长度，OCx/OCxN 极性和被禁止的状态，OCxM 配置，刹车使能和极性)。

用户可以通过 TIMERx_BDTR 寄存器中的 LOCK 位，从三级保护中选择一种，参看 刹车和死区寄存器(TIMERx_BDTR) 。在 MCU 复位后 LOCK 位只能被修改一次。

下图显示响应刹车的输出实例。

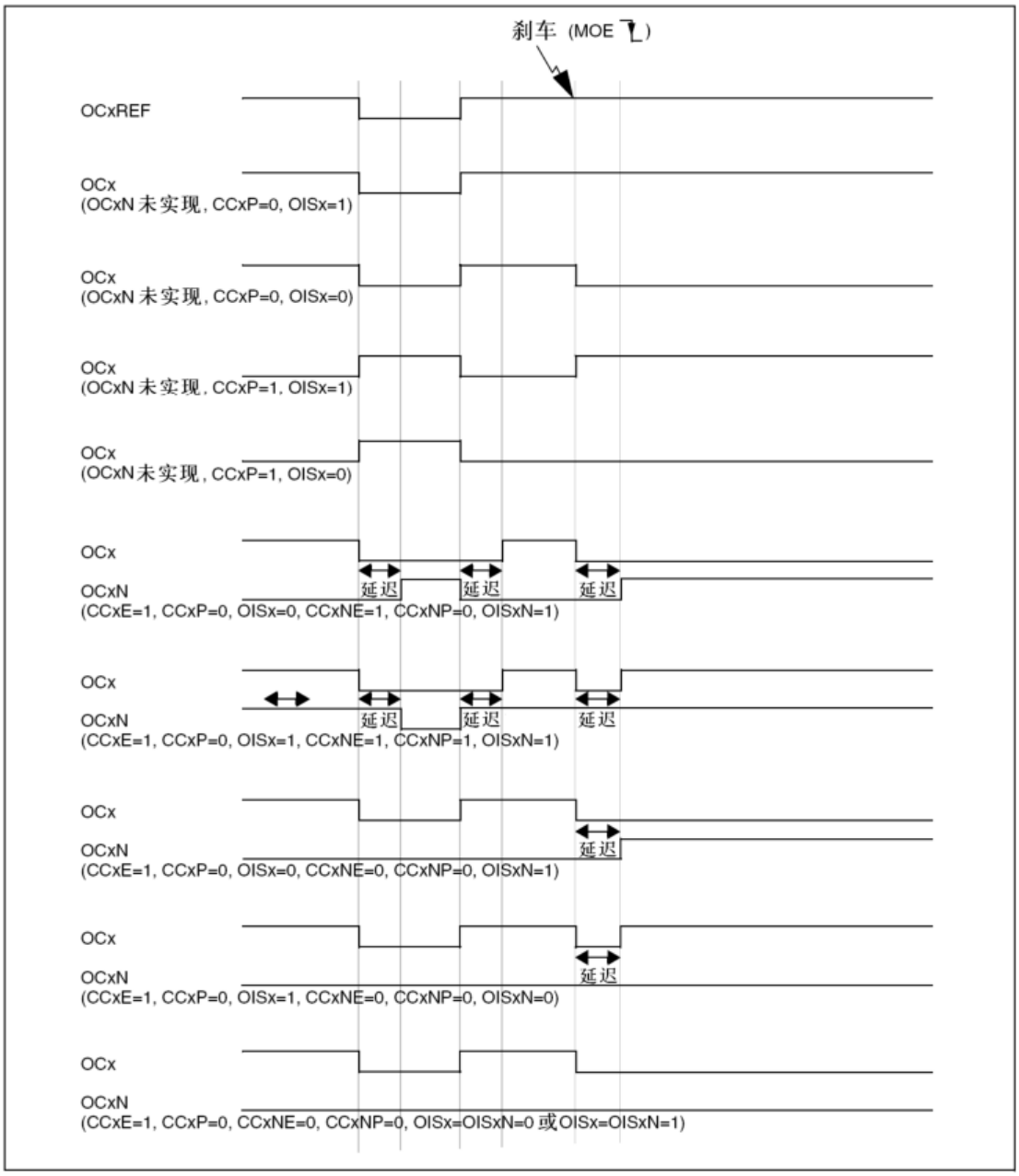


图 16-36 响应刹车的输出

16.3.13. 单脉冲模式

单脉冲模式(OPM)是前述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后产生一个脉宽可程序控制的脉冲。

可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIMERx_CR1 寄存器中的 OPM 位将选择单脉冲模式，这样可以让计数器自动地在产生下一个更新事件 UEV 时停止。

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

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

单脉冲模式的例子

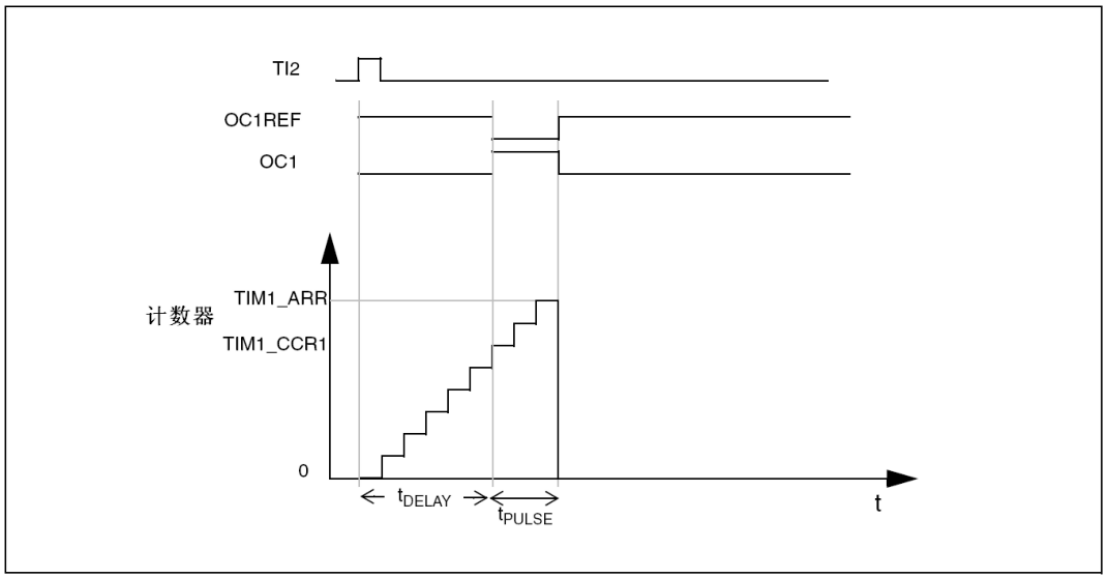


图 16-37

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

假定 TI2FP2 作为触发 1:

- 置 TIMERx_CCMR1 寄存器中的 CC2S=01，把 TI2FP2 映像到 TI2。
- 置 TIMERx_CCER 寄存器中的 CC2P=0，使 TI2FP2 能够检测上升沿。
- 置 TIMERx_SMCR 寄存器中的 TS=110，TI2FP2 作为从模式控制器的触发(TRGI)。
- 置 TIMERx_SMCR 寄存器中的 SMS=110(触发模式)，TI2FP2 被用来启动计数器。
  OPM 的波形由写入比较寄存器的数值决定(要考虑时钟频率和计数器预分频器)
- tDELAY 由 TIMERx_CCR1 寄存器中的值定义。
- tPULSE 由自动装载值和比较值之间的差值定义(TIMERx_ARR - TIMERx_CCR1)。
- 假定当发生比较匹配时要产生从 0 到 1 的波形，当计数器达到预装载值时要产生一个从 1 到 0 的波形;

首先要置 TIMERx_CCMR1 寄存器的 OC1M=111，进入 PWM 模式 2；根据需要有选择地使能预装载寄存器:置 TIMERx_CCMR1 中的 OC1PE=1 和 TIMERx_CR1 寄存器中的 ARPE;然后在 TIMERx_CCR1 寄存器中填写比较值，在 TIMERx_ARR 寄存器中填写自动装载值，设置 UG 位来产生一个更新事件，然后等待在 TI2 上的一个外部触发事件。本例中，CC1P=0。

在这个例子中，TIMERx_CR1 寄存器中的 DIR 和 CMS 位应该置低。

因为只需要一个脉冲，所以必须设置 TIMERx_CR1 寄存器中的 OPM=1，在下一个更新事件(当计数器从自动装载值翻转到 0)时停止计数。

特殊情况： 特殊情况：OCx 快速使能：

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

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

16.3.14. TIMERx 定时器和外部触发的同步

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

从模式：复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMERx_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV;然后所有的预装载寄存器(TIMERx_ARR，TIMERx_CCRx)都被更新了。

在以下的例子中，TI1 输入端的上升沿导致向上计数器被清零:

- 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽(在本例中，不需要任何滤波器，因此保持

IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位只选择输入捕获源，即 TIMERx_CCMR1 寄存器中 CC1S=01。置 TIMERx_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)。

- 置 TIMERx_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIMERx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMERx_CR1 寄存器中 CEN=1，启动计数器。

计数器开始依据内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志(TIMERx_SR 寄存器中的 TIF 位)被设置，根据 TIMERx_DIER 寄存器中 TIE(中断使能)位和 TDE(DMA 使能)位的设置，产生一个中断请求或一个 DMA 请求。

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

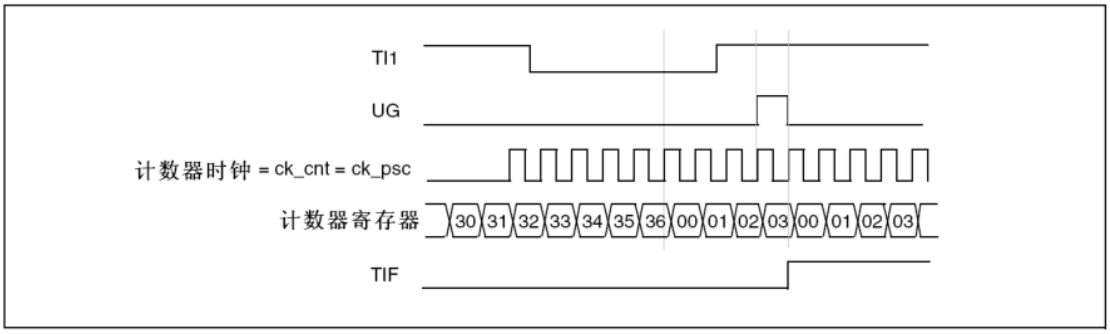


图 16-38

从模式：门控模式

按照选中的输入端电平使能计数器。

在如下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽(本例中，不需要滤波，所以保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输入捕获源，置 TIMERx_CCMR1 寄存器中 CC1S=01。置 TIMERx_CCER 寄存器中 CC1P=1 以确定极性(只检测低电平)。
- 置 TIMERx_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIMERx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMERx_CR1 寄存器中 CEN=1，启动计数器。在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何。

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

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

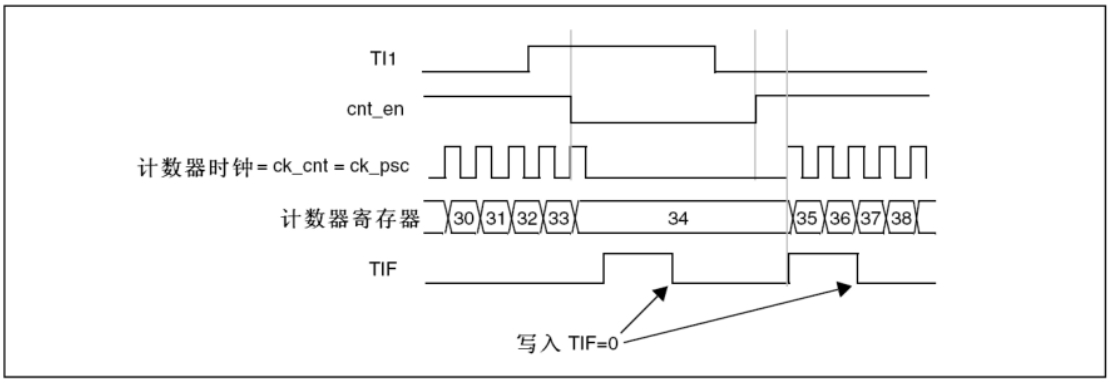


图 16-39

从模式：触发模式

输入端上选中的事件使能计数器。

在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽(本例中，不需要任何滤波器，保持 IC2F=0000)。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIMERx_CCMR1

寄存器中 CC2S=01。置 TIMERx_CCER 寄存器中 CC2P=1 以确定极性(只检测低电平)。

- 置 TIMERx_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIMERx_SMCR 寄存器中 TS=110，选择 TI2 作为输入源。

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

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

触发器模式下的控制电路

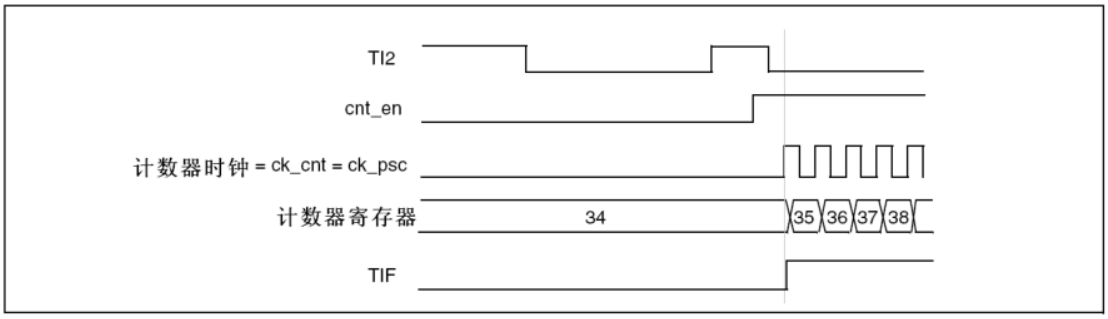


图 16-40

16.3.15. 定时器同步

所有 TIM 定时器在内部相连，用于定时器同步或链接。

16.4. 寄存器描述

16.4.1. 寄存器列表

Base address：0x4002 0000

Name	Offset	Reset	Description
TIMERx_CR1	0x520/0x580/ 0x5E0	32'h0	TIMERx 控制寄存器 1
TIMERx_CR2	0x524/0x584/ 0xE4	32'h0	TIMERx 控制寄存器 2
TIMERx_SMCR	0x528/0x588/ 0xE8	32'h0	TIMERx 从模式控制寄存器
TIMERx_DIER	0x52C/0x58C/ 0x5EC	32'h0	TIMERx DMA/中断使能寄存器
TIMERx_SR	0x530/0x590/ 0x5F0	32'h0	TIMERx 状态寄存器
TIMERx_EGR	0x534/0x594/ 0x5F4	32'h0	TIMERx 事件产生寄存器
TIMERx_CCMR1	0x538/0x598/ 0x5F8	32'h0	TIMERx 捕获/比较模式寄存器 1
Reserved	0x53C/0x59C/ 0x5FC	-	-
TIMERx_CCER	0x540/0x5A0/ 0x600	32'h0	TIMERx 捕获/比较使能寄存器
TIMERx_CNT	0x544/0x5A4/ 0x604	32'h0	TIMERx 计数器
TIMERx_PSC	0x548/0x5A8/ 0x608	32'h0	TIMERx 预分频器
TIMERx_ARR	0x54C/0x5AC/ 0x60C	32'hFFFF	TIMERx 自动重载寄存器

TIMERx_RCR	0x550/0x5B0/ 0x610	32'h0	TIMERx 重复计数寄存器
TIMERx_CCR1	0x554/0x5B4/ 0x614	32'hFFFF	TIMERx 捕获/比较寄存器 1
TIMERx_CCR2	0x558/0x5B8/ 0x618	32'hFFFF	TIMERx 捕获/比较寄存器 2
Reserved	0x55C/0x5BC/ 0x61C	-	-
Reserved	0x560/0x5C0/ 0x620	-	-
TIMERx_BDTR	0x564/0x5C4/ 0x624	32'h0	TIMERx 刹车和死区寄存器

16.4.2. 寄存器详细说明

16.4.2.1. 控制寄存器 1（TIMERx_CR1）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	ETRBKEN	1'b0	RW	外部 IO 刹车输入开关 0: 禁止外部 IO 输入刹车 1: 开启外部 IO 输入刹车 注：当设置了 LOCK 级别 1 时(TIMERx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。
14	SYSBKEN	1'b0	RW	系统错误刹车开关 0: 禁止系统错误刹车 1: 开启系统错误刹车（越界访问 sram） 注：当设置了 LOCK 级别 1 时(TIMERx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。
13	CMP1BKP	1'b0	RW	比较器 1 刹车极性 0: 比较器 1 刹车低电平有效 1: 比较器 1 刹车高电平有效 注：当设置了 LOCK 级别 1 时(TIMERx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。
12	CMP1BKEN	1'b0	RW	比较器 1 刹车开关 0: 禁止比较器 1 刹车 1: 开启比较器 1 刹车 注：当设置了 LOCK 级别 1 时(TIMERx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。
11	CMP0BKP	1'b0	RW	比较器 0 刹车极性 0: 比较器 0 刹车低电平有效 1: 比较器 0 刹车高电平有效 注：当设置了 LOCK 级别 1 时(TIMERx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。
10	CMP0BKEN	1'b0	RW	比较器 0 刹车开关 0: 禁止比较器 0 刹车 1: 开启比较器 0 刹车 注：当设置了 LOCK 级别 1 时(TIMERx_BDTR

				寄存器中的 LOCK 位），该位不能被修改。
9:8	CKD	2'b0	RW	时钟分频因子 这 2 位定义在定时器时钟(CK_INT)频率、死区时间和由死区发生器与数字滤波器(ETR,TIx)所用的采样时钟之间的分频比例。 b00: tDTS = t CK_INT b01: tDTS = 2 x t CK_INT b10: tDTS = 4 x t CK_INT b11: 保留，不要使用这个配置
7	ARPE	1'b0	RW	自动重装载预装载允许位 0: TIMERx_ARR 寄存器没有缓冲 1: TIMERx_ARR 寄存器被装入缓冲器。
6:5	CMS	2'b0	RW	选择中央对齐模式 b00: 边沿对齐模式。计数器依据方向位(DIR)向上或向下计数。 b01: 中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道(TIMERx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器向下计数时被设置。 b10: 中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道(TIMERx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器向上计数时被设置。 b11: 中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道(TIMERx_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位，在计数器向上和向下计数时均被设置。 注：在计数器开启时(CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。
4	DIR	1'b0	RW	计数方向 0: 计数器向上计数 1: 计数器向下计数 注：当计数器配置为中央对齐模式或编码器模式时，该位为只读
3	OPM	1'b0	RW	单脉冲模式 0: 在发生更新事件时，计数器不停止 1: 在发生下一次更新事件(清除 CEN 位)时，计数器停止
2	URS	1'b0	RW	更新请求源 0: 如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求： – 计数器溢出/下溢 – 设置 UG 位 – 从模式控制器产生的更新 1: 如果使能了更新中断或 DMA 请求，则只有计数器溢出/下溢才产生更新中断或 DMA 请求。
1	UDIS	1'b0	RW	禁止更新

				软件通过该位允许/禁止 UEV 事件的产生 0: 允许 UEV。更新(UEV)事件由下述任一事件产生: – 计数器溢出/下溢 – 设置 UG 位 – 从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。(译注：更新影子寄存器) 1: 禁止 UEV。不产生更新事件，影子寄存器(ARR、PSC、CCRx)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。
0	CEN	1'b0	RW	使能计数器 0: 禁止计数器 1: 使能计数器 注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

16.4.2.2. 控制寄存器 2（TIMERx_CR2）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	ADCBKEN	1'b0	RW	ADC 刹车开关 0: 禁止 ADC 刹车 1: 开启 ADC 刹车 注：当设置了 LOCK 级别 1 时(TIMERx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。
14:11	Reserved	-	-	-
10	OIS2	1'b0	RW	输出空闲状态 2(OC2 输出)。 参见 OIS1 位。
9	OIS1N	1'b0	RW	输出空闲状态 1(OC1N 输出) 0: 当 MOE=0 时，死区后 OC1N=0 1: 当 MOE=0 时，死区后 OC1N=1 注：已经设置了 LOCK(TIMERx_BKR 寄存器)级别 1、2 或 3 后，该位不能被修改。
8	OIS1	1'b0	RW	输出空闲状态 1(OC1 输出) 0: 当 MOE=0 时，如果实现了 OC1N，则死区后 OC1=0 1: 当 MOE=0 时，如果实现了 OC1N，则死区后 OC1=1 注：已经设置了 LOCK(TIMERx_BKR 寄存器)级别 1、2 或 3 后，该位不能被修改。
7	Reserved	-	-	-
6:4	MMS	3'b0	RW	主模式选择 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下: b000: 复位 – TIMERx_EGR 寄存器的 UG 位被

				用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式)，则TRGO 上的信号相对实际的复位会有一个延迟。 b001: 使能 – 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式(见TIMERx_SMCR 寄存器中 MSM 位的描述)。 b010: 更新 – 更新事件被选为触发输入(TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 b011: 比较脉冲 – 在发生一次捕获或一次比较成功时，当要设置 CC1IF 标志时(即使它已经为高)，触发输出送出一个正脉冲(TRGO)。 b100: 比较 – OC1REF 信号被用于作为触发输出(TRGO)。 b101: 比较 – OC2REF 信号被用于作为触发输出(TRGO)。 b110: 比较 – OC3REF 信号被用于作为触发输出(TRGO)。 b111: 比较 – OC4REF 信号被用于作为触发输出(TRGO)。
3	CCDS	1'b0	RW	捕获/比较的 DMA 选择 0: 当发生 CCx 事件时，送出 CCx 的 DMA 请求 1: 当发生更新事件时，送出 CCx 的 DMA 请求
2	CCUS	1'b0	RW	捕获/比较控制更新选择 0: 如果捕获/比较控制位是预装载的(CCPC=1)，只能通过设置 COM 位更新它们 1: 如果捕获/比较控制位是预装载的(CCPC=1)，可以通过设置 COM 位或 TRGI 上的一个上升沿 更新它们 注：该位只对具有互补输出的通道起作用
1	Reserved	-	-	-
0	CCPC	1'b0	RW	捕获/比较预装载控制位 0: CCxE，CCxNE 和 OCxM 位不是预装载的 1: CCxE，CCxNE 和 OCxM 位是预装载的；设置该位后，它们只在设置了 COM 位后被更新。 注：该位只对具有互补输出的通道起作用。

16.4.2.3. 从模式控制寄存器（TIMERx_SMCR)

Width	Name	Reset	Property	Description
31:8	Reserved	-	-	-
7	MSM	1'b0	RW	主/从模式

				0: 无作用 1: 触发输入(TRGI)上的事件被延迟了，以允许在当前定时器(通过 TRGO)与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的
6:4	TS	3'b0	RW	触发选择 这 3 位选择用于同步计数器的触发输入。 b000: 内部触发 0(ITR0) b001: 内部触发 1(ITR1) b010: 内部触发 2(ITR2) b011: 内部触发 3(ITR3) b100: TI1 的边沿检测器(TI1F_ED) b101: 滤波后的定时器输入 1(TI1FP1) b110: 滤波后的定时器输入 2(TI2FP2) b111: 外部触发输入(ETRF)
3	Reserved	-	-	-
2:0	SMS	3'b0	RW	从模式选择 当选择了外部信号，触发信号(TRGI)的有效边沿与选中的外部输入极性相关(见输入控制寄存器和控制寄存器的说明) b000: 关闭从模式 – 如果 CEN=1，则预分频器直接由内部时钟驱动。 b001: Reserved b010: Reserved b011: Reserved b100: 复位模式 – 选中的触发输入(TRGI)的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。 b101: 门控模式 – 当触发输入(TRGI)为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止(但不复位)。计数器的启动和停止都是受控的。 b110: 触发模式 – 计数器在触发输入 TRGI 的上升沿启动(但不复位)，只有计数器的启动是受控的。 b111: 外部时钟模式 1 – 选中的触发输入(TRGI)的上升沿驱动计数器。 注: 1. 如果 TI1F_ED 被选为触发输入(TS=100)时，不要使用门控模式。这是因为，TI1F_ED 在每次 TI1F 变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。 2. 如果配置成编码器模式，且 ECE=0，TIMx_ETR 的上升沿会对计数器清零。

16.4.2.4. DMA/中断使能寄存器（TIMERx_DIER)

Width	Name	Reset	Property	Description

31:15	Reserved	-	-	-
14	TDE	1'b0	RW	允许触发 DMA 请求 0: 禁止触发 DMA 请求 1: 允许触发 DMA 请求
13	COMDE	1'b0	RW	允许 COM 的 DMA 请求 0: 禁止 COM 的 DMA 请求 1: 允许 COM 的 DMA 请求
12:11	Reserved	-	-	-
10	CC2DE	1'b0	RW	允许捕获/比较 2 的 DMA 请求 0: 禁止捕获/比较 2 的 DMA 请求 1: 允许捕获/比较 2 的 DMA 请求
9	CC1DE	1'b0	RW	允许捕获/比较 1 的 DMA 请求 0: 禁止捕获/比较 1 的 DMA 请求 1: 允许捕获/比较 1 的 DMA 请求
8	UDE	1'b0	RW	允许更新的 DMA 请求 0: 禁止更新的 DMA 请求 1: 允许更新的 DMA 请求
7	BIE	1'b0	RW	允许刹车中断 0: 禁止刹车中断 1: 允许刹车中断
6	TIE	1'b0	RW	触发中断使能 0: 禁止触发中断 1: 使能触发中断
5	COMIE	1'b0	RW	允许 COM 中断 0: 禁止 COM 中断 1: 允许 COM 中断
4:3	Reserved	-	-	-
2	CC2IE	1'b0	RW	允许捕获/比较 2 中断 0: 禁止捕获/比较 2 中断 1: 允许捕获/比较 2 中断
1	CC1IE	1'b0	RW	允许捕获/比较 1 中断 0: 禁止捕获/比较 1 中断 1: 允许捕获/比较 1 中断
0	UIE	1'b0	RW	允许更新中断 0: 禁止更新中断 1: 允许更新中断

16.4.2.5. 状态寄存器（TIMERx_SR）

Width	Name	Reset	Property	Description
31:11	Reserved	-	-	-
10	CC2OF	1'b0	RC	捕获 2 重复捕获标记 参见 CC1OF 描述。
9	CC1OF	1'b0	RC	捕获 1 重复捕获标记 0: 无重复捕获产生 1: 计数器的值被捕获到 TIMERx_CCR1 寄存器时，CC1IF 的状态已经为‘1’。
8	Reserved	-	-	-

7	BIF	1'b0	RC	刹车中断标记 一旦刹车输入有效，由硬件对该位置 1，如果刹车输入无效，则该位可由软件清 0。 0: 无刹车事件产生 1: 刹车输入上检测到有效电平
6	TIF	1'b0	RC	触发器中断标记 当发生触发事件(当从模式控制器处于除门控模式外的其它模式时，在 TRGI 输入端检测到有效边沿，或门控模式下的任一边沿)时由硬件对该位置 1，它由软件清 0。 0: 无触发器事件产生 1: 触发中断等待响应
5	COMIF	1'b0	RC	COM 中断标记 一旦产生 COM 事件(当捕获/比较控制位: CCxE、CCxNE、OCxM 已被更新)该位由硬件置 1，它由软件清 0。 0: 无 COM 事件产生 1: COM 中断等待响应
4:3	Reserved	-	-	-
2	CC2IF	1'b0	RC	捕获/比较 2 中断标记 参考 CC1IF 描述
1	CC1IF	1'b0	RC	捕获/比较 1 中断标记 **如果通道 如果通道 CC1 配置为输出模式:** 当计数器值与比较值匹配时该位由硬件置 1，但在中心对称模式下除外(参考 TIMERx_CR1 寄存器的 CMS 位)。它由软件清 0。 0: 无匹配发生; 1: TIMERx_CNT 的值与 TIMERx_CCR1 的值匹配。 当 TIMERx_CCR1 的内容大于 TIMERx_ARR 的内容时，在向上或向上/下计数模式时计数器溢出，或向下计数模式时的计数器下溢条件下，CC1IF 位变高 **如果通道 CC1 配置为输入模式:** 当捕获事件发生时该位由硬件置 1，它由软件清 0 或通过读 TIMERx_CCR1 清 0。 0: 无输入捕获产生 1: 计数器值已被捕获(拷贝)至 TIMERx_CCR1(在 IC1 上检测到与所选极性相同的边沿)
0	UIF	1'b0	RC	更新中断标记 当产生更新事件时该位由硬件置 1。它由软件清 0。 0: 无更新事件产生; 1: 更新中断等待响应。当寄存器被更新时该位由硬件置 1: – 若 TIMERx_CR1 寄存器的 UDIS=0，当重复

				计数器数值上溢或下溢时(重复计数器=0 时产生更新事件)。 – 若 TIMERx_CR1 寄存器的 URS=0、UDIS=0，当设置 TIMERx_EGR 寄存器的 UG=1 时产生更新事件，通过软件对计数器 CNT 重新初始化时。 – 若 TIMERx_CR1 寄存器的 URS=0、UDIS=0，当计数器 CNT 被触发事件重新初始化时。(参考从模式控制寄存器 TIMERx_SMCR)。

16.4.2.6. 事件产生寄存器（TIMERx_EGR）

Width	Name	Reset	Property	Description
31:8	Reserved	-	-	-
7	BG	1'b0	WO	产生刹车事件 该位由软件置 1，用于产生一个刹车事件，由硬件自动清 0。 0: 无动作 1: 产生一个刹车事件。此时 MOE=0、BIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA
6	TG	1'b0	WO	产生触发事件 0: 无动作 1: TIMERx_SR 寄存器的 TIF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA
5	COMG	1'b0	WO	捕获/比较事件，产生控制更新 该位由软件置 1，由硬件自动清 0 0: 无动作 1: 当 CCPC=1，允许更新 CCxE、CCxNE、OCxM 位 注：该位只对拥有互补输出的通道有效。
4:3	Reserved	-	-	-
2	CC2G	1'b0	WO	产生捕获/比较 2 事件 参考 CC1G 描述。
1	CC1G	1'b0	WO	产生捕获/比较 1 事件 该位由软件置 1，用于产生一个捕获/比较事件，由硬件自动清 0。 0: 无动作 1: 在通道 CC1 上产生一个捕获/比较事件 **若通道 CC1 配置为输出：** 设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。 **若通道 CC1 配置为输入：** 当前的计数器值被捕获至 TIMERx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经为 1，则设置 CC1OF=1。
0	UG	1'b0	WO	产生更新事件 该位由软件置 1，由硬件自动清 0。

				0: 无动作; 1: 重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清 0(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清 0;若 DIR=1(向下计数)则计数器取 TIMERx_ARR 的值。

16.4.2.7. 捕获/比较模式寄存器 1（TIMERx_CCMR1）

通道可用于输入(捕获模式)或输出(比较模式)，通道的方向由相应的 CCxS 位定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

输出比较模式：

Width	Name	Reset	Property	Description
31:15	Reserved	-	-	-
14:12	OC2M	3'b0	RW	输出比较 2 模式 参考 OC1M
11	OC2PE	1'b0	RW	输出比较 2 预装载使能 参考 OC1PE
10	OC2FE	1'b0	RW	输出比较 2 快速使能 参考 OC1FE
9:8	CC2S	2'b0	RW	捕获/比较 2 选择 该位定义通道的方向(输入/输出)，及输入脚的选择: b00: CC2 通道被配置为输出 b01: CC2 通道被配置为输入，IC2 映射在 TI2 上 b10: CC2 通道被配置为输入，IC2 映射在 TI1 上 b11: CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择) 注：CC2S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC2E=0)才是可写的
7	Reserved	-	-	-
6:4	OC1M	3'b0	RW	输出比较 1 模式 该 3 位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效，而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。 b000: 冻结。输出比较寄存器 TIMERx_CCR1 与计数器 TIMERx_CNT 间的比较对 OC1REF 不起作用 b001: 匹配时设置通道 1 为有效电平。当计数器 TIMERx_CNT 的值与捕获/比较寄存器 1(TIMERx_CCR1)相同时，强制 OC1REF 为高 b010: 匹配时设置通道 1 为无效电平。当计数器 TIMERx_CNT 的值与捕获/比较寄存器

				1(TIMERx_CCR1)相同时，强制 OC1REF 为低 b011: 翻转。当 TIMERx_CCR1=TIMERx_CNT 时，翻转 OC1REF 的电平 b100: 强制为无效电平。强制 OC1REF 为低 b101: 强制为有效电平。强制 OC1REF 为高 b110: PWM 模式 1 - 在向上计数时，一旦 TIMERx_CNT<TIMERx_CCR1 时通道 1 为有效电平，否则为无效电平；在向下计数时，一旦 TIMERx_CNT>TIMERx_CCR1 时通道 1 为无效电平(OC1REF=0)，否则为有效电平(OC1REF=1) b111: PWM 模式 2 - 在向上计数时，一旦 TIMERx_CNT<TIMERx_CCR1 时通道 1 为无效电平，否则为有效电平；在向下计数时，一旦 TIMERx_CNT>TIMERx_CCR1 时通道 1 为有效电平，否则为无效电平 注 1: 一旦 LOCK 级别设为 3(TIMERx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。
3	OC1PE	1'b0	RW	输出比较 1 预装载使能 0: 禁止 TIMERx_CCR1 寄存器的预装载功能，可随时写入 TIMERx_CCR1 寄存器，并且新写入的数值立即起作用。 1: 开启 TIMERx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMERx_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。 注 1: 一旦 LOCK 级别设为 3(TIMERx_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出)则该位不能被修改。 注 2: 仅在单脉冲模式下(TIMERx_CR1 寄存器的 OPM=1)，可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定。
2	OC1FE	1'b0	RW	输出比较 1 快速使能 该位用于加快 CC 输出对触发输入事件的响应。 0: 根据计数器与 CCR1 的值，CC1 正常操作，即使触发器是打开的。当触发器的输入有一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期。 1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。 OCFE 只在通道被配置成 PWM1 或 PWM2 模

				式时起作用。
1:0	CC1S	2'b0	RW	捕获/比较 1 选择 这 2 位定义通道的方向(输入/输出),及输入脚的选择: b00：CC1 通道被配置为输出; b01：CC1 通道被配置为输入，IC1 映射在 TI1 上; b10：CC1 通道被配置为输入，IC1 映射在 TI2 上; b11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择)。 注：CC1S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC1E=0)才是可写的。

输入捕获模式：

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:12	IC2F	4'b0	RW	输入捕获 2 滤波器 参考 IC1F
11:10	IC2PSC	2'b0	RW	输入/捕获 2 预分频器 参考 IC1PSC
9:8	CC2S	2'b0	RW	捕获/比较 2 选择 这 2 位定义通道的方向(输入/输出),及输入脚的选择: b00: CC2 通道被配置为输出 b01: CC2 通道被配置为输入，IC2 映射在 TI2 上 b10: CC2 通道被配置为输入，IC2 映射在 TI1 上 b11: CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择) 注：CC2S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC2E=0)才是可写的。
7:4	IC1F	4'b0	RW	输入捕获 1 滤波器 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变: h0: 无滤波器，以 fDTS 采样 h1: 采样频率 fSAMPLING=fCK_INT，N=2 h2: 采样频率 fSAMPLING=fCK_INT，N=4 h3: 采样频率 fSAMPLING=fCK_INT，N=8 h4: 采样频率 fSAMPLING=fDTS/2，N=6 h5: 采样频率 fSAMPLING=fDTS/2，N=8 h6: 采样频率 fSAMPLING=fDTS/4，N=6

				h7: 采样频率 fSAMPLING=fDTS/4，N=8 h8: 采样频率 fSAMPLING=fDTS/8，N=6 h9: 采样频率 fSAMPLING=fDTS/8，N=8 ha: 采样频率 fSAMPLING=fDTS/16，N=5 hb: 采样频率 fSAMPLING=fDTS/16，N=6 hc: 采样频率 fSAMPLING=fDTS/16，N=8 hd: 采样频率 fSAMPLING=fDTS/32，N=5 he: 采样频率 fSAMPLING=fDTS/32，N=6 hf: 采样频率 fSAMPLING=fDTS/32，N=8
3:2	IC1PSC	2'b0	RW	输入/捕获 1 预分频器 这 2 位定义了 CC1 输入(IC1)的预分频系数。一旦 CC1E=0(TIMERx_CCER 寄存器中)，则预分频器复位。 b00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 b01: 每 2 个事件触发一次捕获 b10: 每 4 个事件触发一次捕获 b11: 每 8 个事件触发一次捕获
1:0	CC1S	2'b0	RW	捕获/比较 1 选择 这 2 位定义通道的方向(输入/输出),及输入脚的选择: b00: CC1 通道被配置为输出; b01: CC1 通道被配置为输入，IC1 映射在 TI1 上; b10: CC1 通道被配置为输入，IC1 映射在 TI2 上; b11: CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCR 寄存器的 TS 位选择)。 注：CC1S 仅在通道关闭时(TIMERx_CCER 寄存器的 CC1E=0)才是可写的。

16.4.2.8. 捕获/比较使能寄存器（TIMERx_CCER)

Width	Name	Reset	Property	Description
31:8	Reserved	-	-	-
7	CC2NP	1'b0	RW	输入/捕获 2 互补输出极性 参考 CC1NP 的描述。
6	Reserved	-	-	-
5	CC2P	1'b0	RW	输入/捕获 2 输出极性 参考 CC1P 的描述。
4	CC2E	1'b0	RW	输入/捕获 2 输出使能 参考 CC1E 的描述。
3	CC1NP	1'b0	RW	输入/捕获 1 互补输出极性 **CC1 配置为输出:** 0: OC1N 高电平有效 1: OC1N 低电平有效 **CC1 配置为输入:**

				用于与 CC1P 一起决定 TI1FP1/TI2FP1 注：一旦 LOCK 级别(TIMERx_BDTR 寄存器中的 LOCK 位)设为 3 或 2 且 CC1S=00(通道配置为输出)则该位不能被修改。
2	CC1NE	1'b0	RW	输入/捕获 1 互补输出使能 0: 关闭 - OC1N 禁止输出，因此 OC1N 的电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。 1: 开启 - OC1N 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。
1	CC1P	1'b0	RW	输入/捕获 1 输出极性 **CC1 通道配置为输出：** 0: OC1 高电平有效; 1: OC1 低电平有效。 **CC1 通道配置为输入：** CC1NP 和 CC1P 一起决定 TI1FP1 和 TI2FP1 极性 b00: 不反相，上升沿。 TIxFP1 上升沿有效（捕获，复位模式，外部时钟，触发模式），TIxFP1 不反向（门控模式，正交编码模式） b01: 反相，下降沿。 TIxFP1 下降沿有效（捕获，复位模式，外部时钟，触发模式），TIxFP1 反向（门控模式，正交编码模式） b10: reserved b11:不反向，上升沿和下降沿都有效。该配置不能用于正交编码模式。 注：一旦 LOCK 级别(TIMERx_BDTR 寄存器中的 LOCK 位)设为 3 或 2，则该位不能被修改。
0	CC1E	1'b0	RW	输入/捕获 1 输出使能 **CC1 通道配置为输出：** 0: 关闭 - OC1 禁止输出，因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 1: 开启 - OC1 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 **CC1 通道配置为输入：** 该位决定了计数器的值是否能捕获入 TIMERx_CCR1 寄存器。 0: 捕获禁止; 0: 捕获使能。

带刹车功能的互补输出通道 OCx 和 OCxN 的控制位

MOE	OSSI	OSSR	CCxE	CCxNE	OCx 输出状态	OCxN 输出状态
1	x	0	0	0	输出禁止(与定时器断开)	输出禁止(与定时器断开)

					OCx=0，OCx_EN=0	OCxN=0，OCxN_EN=0
		0	0	1	输出禁止(与定时器断开) OCx=0，OCx_EN=0	OCxREF + 极性， OCxN= OCxREF xor CCxNP， OCxN_EN=1
		0	1	0	OCxREF + 极性， OCx= OCxREF xor CCxP， OCx_EN=1	输出禁止(与定时器断开) OCxN=0，OCxN_EN=0
		0	1	1	OCxREF + 极性 + 死区， OCx_EN=1	OCxREF 反相+极性+死区， OCxN_EN=1
		1	0	0	输出禁止(与定时器断开) OCx=CCxP，OCx_EN=0	输出禁止(与定时器断开) OCxN=CCxNP，OCxN_EN=0
		1	0	1	关闭状态(输出使能且为无效电平) OCx=CCxP，OCx_EN=1	OCxREF + 极性， OCxN= OCxREF xor CCxNP， OCxN_EN=1
		1	1	0	OCxREF + 极性， OCx= OCxREF xor CCxP， OCx_EN=1	关闭状态(输出使能且为无效电平) OCxN=CCxNP，OCxN_EN=1
		1	1	1	OCxREF + 极性 + 死区， OCx_EN=1	OCxREF 反相+极性+死区， OCxN_EN=1
0	0	x	0	0	输出禁止(与定时器断开) 异步地: OCx=CCxP，OCx_EN=0，OCxN=CCxNP，OCxN_EN=0; 若时钟存在：经过一个死区时间后 OCx=OISx，OCxN=OISxN，假设 OISx 与 OISxN 并不都对应 OCx 和 OCxN 的有效电平。	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0		
	1		0	1	关闭状态(输出使能且为无效电平) 异步地: OCx=CCxP，OCx_EN=1，OCxN=CCxNP，OCxN_EN=1; 若时钟存在: 经过一个死区时间后 OCx=OISx，OCxN=OISxN，假设 OISx 与 OISxN 并不都对应 OCx 和 OCxN 的有效电平。	
	1		1	0		
	1		1	1		

1. 如果一个通道的 2 个输出都没有使用(CCxE = CCxNE = 0)，那么 OISx, OISxN, CCxP 和 CCxNP 都必须清零。

2.引脚连接到互补的 OCx 和 OCxN 通道的外部 I/O 引脚的状态，取决于 OCx 和 OCxN 通道状态和 GPIO 控制寄存器。

3.当 SYSCON1[31]为 1 时，只要通道配置成输出模式，则 IO 一直由 timer 驱动。

当 SYSCON1[31]为 0 时，上表中 OCx_EN/OCxN_EN=1 时，IO 由 timer 驱动，OCx_EN/OCxN_EN=0 时，IO 为浮空态。

16.4.2.9. 计数器（TIMERx_CNT)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	CNT	16'b0	RW	计数器的值

16.4.2.10. 预分频器（TIMERx_PSC）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	PSC	16'b0	RW	预分频器的值 计数器的时钟频率(CK_CNT)等于 fCK_PSC /(PSC[15:0]+1)。 PSC 包含了每次当更新事件产生时，装入当前预分频器寄存器的值；更新事件包括计数器被 TIM_EGR 的 UG 位清 0 或被工作在复位模式的从控制器清 0。

16.4.2.11. 自动重装载寄存器（TIMERx_ARR）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	ARR	16'hffff	RW	自动重装载的值 ARR 包含了将要装载入实际的自动重装载寄存器的值。 当自动重装载的值为空时，计数器不工作。

16.4.2.12. 重复计数寄存器（TIMERx_RCR）

Width	Name	Reset	Property	Description
31:8	Reserved	-	-	-
7:0	REP	8'b0	RW	重复计数器的值 开启了预装载功能后，这些位允许用户设置比较寄存器的更新速率(即周期性地从预装载寄存器传输到当前寄存器)；如果允许产生更新中断，则会同时影响产生更新中断的速率。 每次向下计数器 REP_CNT 达到 0，会产生一个更新事件并且计数器 REP_CNT 重新从 REP 值开始计数。由于 REP_CNT 只有在周期更新事件 U_RC 发生时才重载 REP 值，因此对 TIMERx_RCR 寄存器写入的新值只在下次周期更新事件发生时才起作用。 这意味着在 PWM 模式中，(REP+1)对应着： - 在边沿对齐模式下，PWM 周期的数目； - 在中心对称模式下，PWM 半周期的数目；

16.4.2.13. 捕获/比较寄存器 1（TIMERx_CCR1）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	CCR1	16'hffff	RW	捕获/比较通道 1 的值 **若 CC1 通道配置为输出：** CCR1 包含了装入当前捕获/比较 1 寄存器的值(预装载值)。 如果在 TIMERx_CCMR1 寄存器(OC1PE 位)中

				未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 1 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMERx_CNT 的比较，并在 OC1 端口上产生输出信号。 **若 CC1 通道配置为输入：** CCR1 包含了由上一次输入捕获 1 事件(IC1)传输的计数器值。 注：当配置为捕获模式时，该寄存器变成只读。

16.4.2.14. 捕获/比较寄存器 2（TIMERx_CCR2）

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	CCR2	16'hffff	RW	捕获/比较通道 2 的值 **若 CC2 通道配置为输出：** CCR2 包含了装入当前捕获/比较 2 寄存器的值(预装载值)。 如果在 TIMERx_CCMR2 寄存器(OC2PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 2 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMERx_CNT 的比较，并在 OC2 端口上产生输出信号。 **若 CC2 通道配置为输入：** CCR2 包含了由上一次输入捕获 2 事件(IC2)传输的计数器值。 注：当配置为捕获模式时，该寄存器变成只读。

16.4.2.15. 刹车和死区寄存器（TIMERx_BDTR）

注：根据锁定设置，AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0]位均可被写保护，有必要在第一次写入 TIMERx_BDTR 寄存器时对它们进行配置。

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	MOE	1'b0	RW	主输出使能 一旦刹车输入有效，该位被硬件异步清 0。根据 AOE 位的设置值，该位可以由软件清 0 或被自动置 1。它仅对配置为输出的通道有效。 0: 禁止 OC 和 OCN 输出或强制为空闲状态； 1: 如果设置了相应的使能位(TIMERx_CCER 寄存器的 CCxE、CCxNE 位)，则开启 OC 和 OCN 输出。 有关 OC/OCN 使能的细节，参见捕获/比较使能寄存器(TIMERx_CCER)。
14	AOE	1'b0	RW	自动输出使能

				0: MOE 只能被软件置 1； 1: MOE 能被软件置 1 或在下一个更新事件被自动置 1(如果刹车输入无效)。 注：一旦 LOCK 级别(TIMERx_BDTR 寄存器中的 LOCK 位)设为 1，则该位不能被修改。
13	BKP	1'b0	RW	外部 IO 刹车输入极性 0: 刹车输入低电平有效 1: 刹车输入高电平有效 注：一旦 LOCK 级别(TIMERx_BDTR 寄存器中的 LOCK 位)设为 1，则该位不能被修改。
12	BKE	1'b0	RW	刹车功能使能 0: 禁止刹车输入 1：开启刹车输入(需要配置 TIMERx_CR1 选择刹车信号源)。 注：当设置了 LOCK 级别 1 时(TIMERx_BDTR 寄存器中的 LOCK 位)，该位不能被修改。
11	OSSR	1'b0	RW	运行模式下“关闭状态”选择该位用于当 MOE=1 且通道为互补输出时。没有互补输出的定时器中不存在 OSSR 位。 参考 OC/OCN 使能的详细说明(捕获/比较使能寄存器(TIMERx_CCER))。 0: 当定时器不工作时，禁止 OC/OCN 输出(OC/OCN 使能输出信号=0)； 1: 当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，首先开启 OC/OCN 并输出无效电平，然后置 OC/OCN 使能输出信号=1。 注: 一旦 LOCK 级别(TIMERx_BDTR 寄存器中的 LOCK 位)设为 2，则该位不能被修改。
10	OSSI	1'b0	RW	空闲模式下“关闭状态”选择该位用于当 MOE=0 且通道设为输出时。 参考 OC/OCN 使能的详细说明(捕获/比较使能寄存器(TIMERx_CCER))。 0: 当定时器不工作时，禁止 OC/OCN 输出(OC/OCN 使能输出信号=0)； 1: 当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，OC/OCN 首先输出其空闲电平，然后 OC/OCN 使能输出信号=1。 注：一旦 LOCK 级别(TIMERx_BDTR 寄存器中的 LOCK 位)设为 2，则该位不能被修改。
9:8	LOCK	2'b0	RW	锁定设置 该位为防止软件错误而提供写保护。 b00: 锁定关闭，寄存器无写保护； b01: 锁定级别 1，不能写入 TIMERx_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 TIMERx_CR2 寄存器的 OISx/OISxN 位； b10: 锁定级别 2，不能写入锁定级别 1 中的各位，也不能写入 CC 极性位(一旦相关通道通过

				CCxS 位设为输出，CC 极性位是 TIMERx_CCER 寄存器的 CxP/CCNxP 位)以及 OSSR/OSSI 位； b11：锁定级别 3，不能写入锁定级别 2 中的各位，也不能写入 CC 控制位(一旦相关通道通过 CCxS 位设为输出，CC 控制位是 TIMERx_CCMRx 寄存器的 OCxM/OCxPE 位)； 注：在系统复位后，只能写一次 LOCK 位，一旦写入 TIMERx_BDTR 寄存器，则其内容冻结直至复位。
7:0	DTG	8'b0	RW	死区发生器设置 这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间： DTG[7:5]=0xx=>DT=DTG[7:0]×T dtg，Tdtg=TDTS ； DTG[7:5]=10x=>DT=(64+DTG[5:0]) × Tdtg，Tdtg=2×TDTS； DTG[7:5]=110=>DT=(32+DTG[4:0]) × Tdtg，Tdtg=8×TDTS； DTG[7:5]=111=> DT=(32+DTG[4:0]) ×Tdtg，Tdtg=16×TDTS； 例：若 TDTS=125ns(8MHZ)，可能的死区时间为： 0 到 15875ns，若步长时间为 125ns； 16us 到 31750ns，若步长时间为 250ns； 32us 到 63us，若步长时间为 1us； 64us 到 126us，若步长时间为 2us； 注：一旦 LOCK 级别(TIMERx_BDTR 寄存器中的 LOCK 位)设为 1、2 或 3，则不能修改这些位。

17. 基本定时器（TIMER7）

17.1. 模块介绍

基本定时器 TIMER7 包含一个 16 位自动装载计数器，由可编程预分频器驱动，它们可以作为通用定时器提供时间基准。

17.2. 功能特点

- 16 位自动重装载累加计数器
- 16 位可编程(可实时修改)预分频器，用于对输入的时钟按系数为 1～65536 之间的任意数值分频
- 在更新事件(计数器溢出)时产生中断/DMA 请求

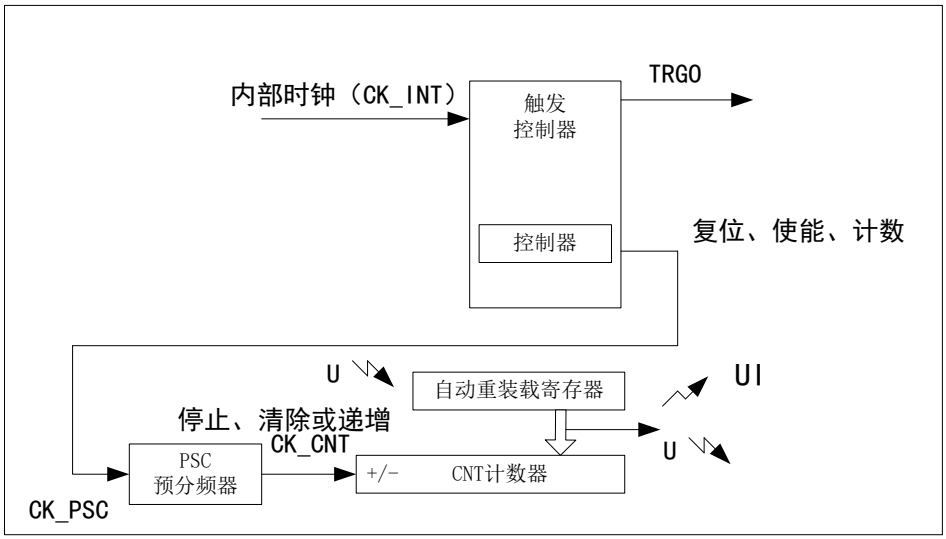


图 17-1 基本定时器

17.3. 功能说明

17.3.1. 时基单元

这个可编程定时器的主要部分是一个带有自动重装载的 16 位累加计数器，计数器的时钟通过一个预分频器得到。

软件可以读写计数器、自动重装载寄存器和预分频寄存器，即使计数器运行时也可以操作。

时基单元包含：

- 计数器寄存器(TIMERx_CNT)
- 预分频寄存器(TIMERx_PSC)
- 自动重装载寄存器(TIMERx_ARR)

自动重装载寄存器是预加载的，每次读写自动重装载寄存器时，实际上是通过读写预加载寄存器实现。根据 TIMERx_CR1 寄存器中的自动重装载预加载使能位(ARPE)，写入预加载寄存器的内容能够立即或在每次更新事件时，传送到它的影子寄存器。当 TIMERx_CR1 寄存器的 UDIS 位为”0“，则每当计数器达到溢出值时，硬件发出更新事件；软件也可以产生更新事件；关于更新事件的产生，随后会有详细的介绍。

计数器由预分频输出 CK_CNT 驱动，设置 TIMERx_CR1 寄存器中的计数器使能位(CEN)使能计数器计数。

注：实际的设置计数器使能信号 CNT_EN 相对于 CEN 滞后一个时钟周期。

预分频器

预分频可以以系数介于 1 至 65536 之间的任意数值对计数器时钟分频。它是通过一个 16 位寄存器(TIMERx_PSC)的计数实现分频。因为 TIMERx_PSC 控制寄存器具有缓冲，可以在运行过程中改变它的数值，新的预分频数值将在下一个更新事件时起作用。

以下两图是在运行过程中改变预分频系数的例子。
预分频系数从 1 变到 2 的计数器时序图

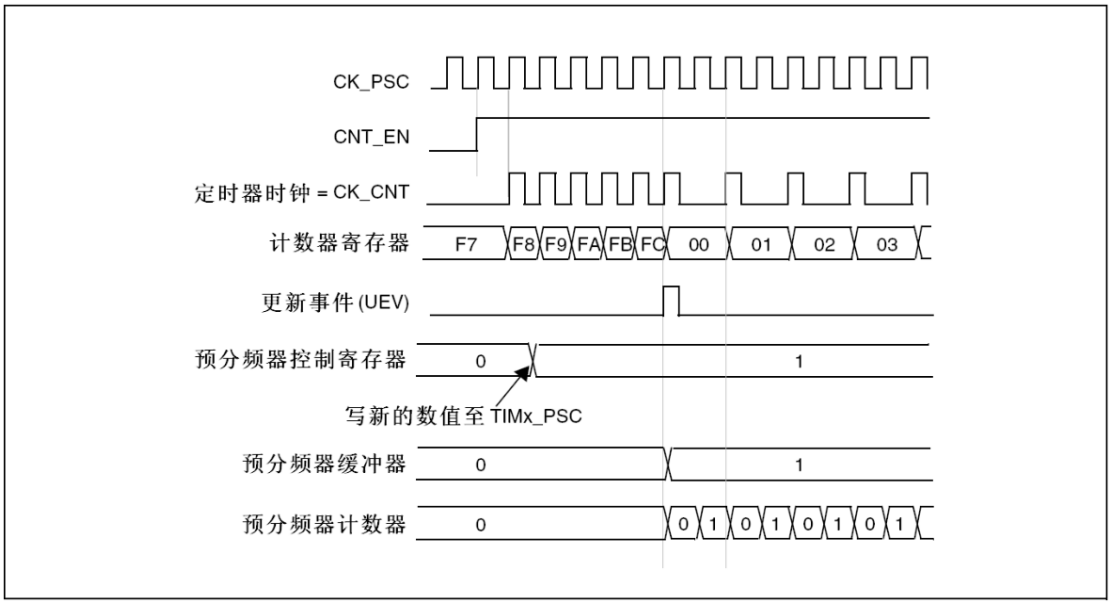


图 17-2

预分频系数从 1 变到 4 的计数器时序图

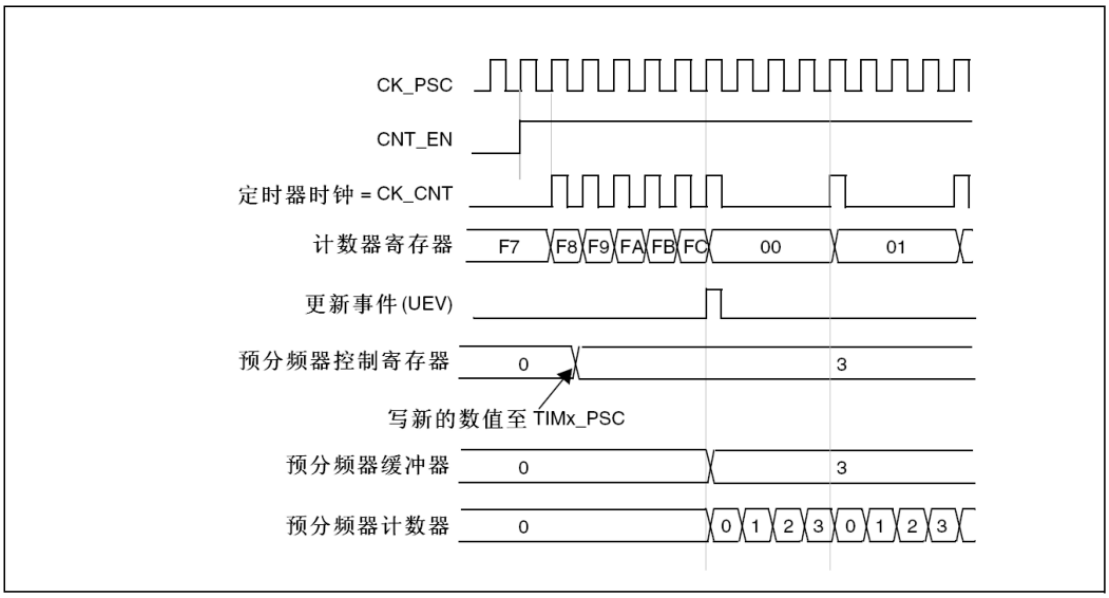


图 17-3

17.3.2. 计数模式

计数器从 0 累加计数到自动重装载数值(TIMERx_ARR 寄存器)，然后重新从 0 开始计数并产生一个计数器溢出事件。

每次计数器溢出时可以产生更新事件；(通过软件或使用从模式控制器)设置 TIMERx_EGR 寄存器的 UG 位也可以产生更新事件。

设置 TIMERx_CR1 中的 UDIS 位可以禁止产生 UEV 事件，这可以避免在写入预加载寄存器时更改影子寄存器。在清除 UDIS 位为’0’之前，将不再产生更新事件，但计数器和预分频器依然会在应产生更新事件时重新从 0 开始计数(但预分频系数不变)。另外，如果设置了 TIMERx_CR1 寄存器中的 URS(选择更新请求)，设置 UG 位可以产生一次更新事件 UEV，但不设置 UIF 标志(即没有中断或 DMA 请求)。

当发生一次更新事件时，所有寄存器会被更新并(根据 URS 位)设置更新标志(TIMERx_SR 寄存器的 UIF 位)：

- 传送预装载值(TIMERx_PSC 寄存器的内容)至预分频器的缓冲区。
- 自动重装载影子寄存器被更新为预装载值(TIMERx_ARR)。

以下是一些在 TIMERx_ARR=0x36 时不同时钟频率下计数器工作的图示例子。

计数器时序图，内部时钟分频系数为 1

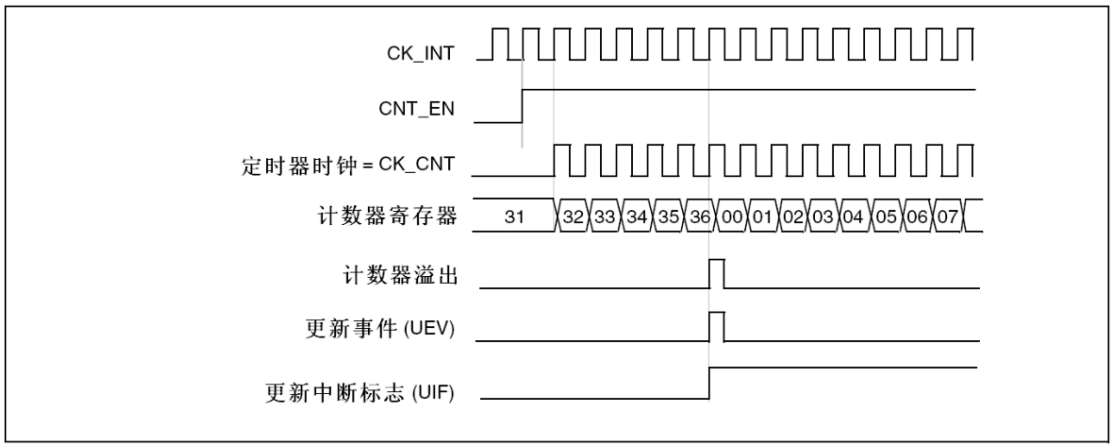


图 17-4

计数器时序图，内部时钟分频系数为 2

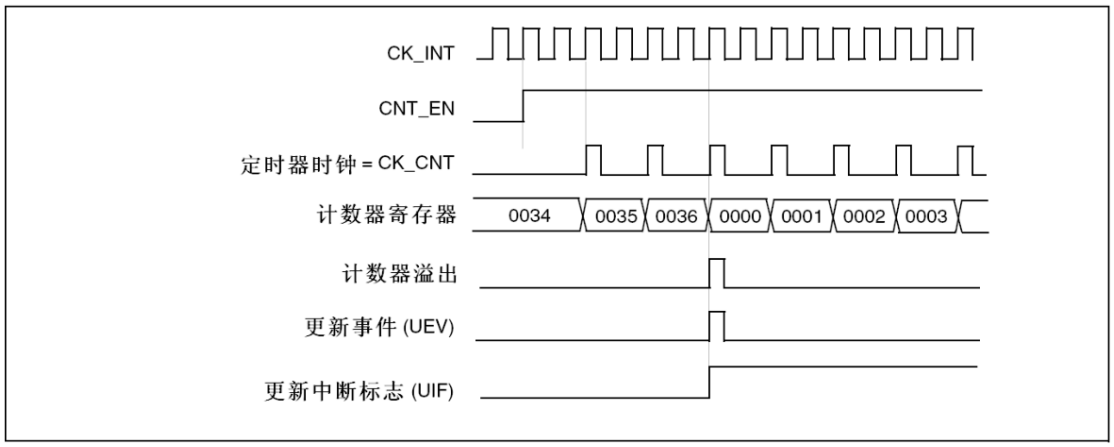


图 17-5

计数器时序图，内部时钟分频系数为 4

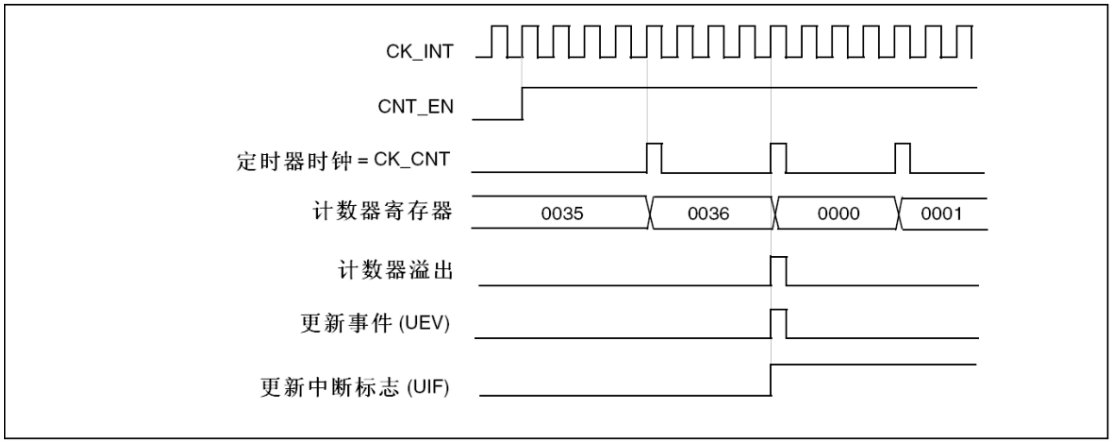


图 17-6

计数器时序图，内部时钟分频系数为 N

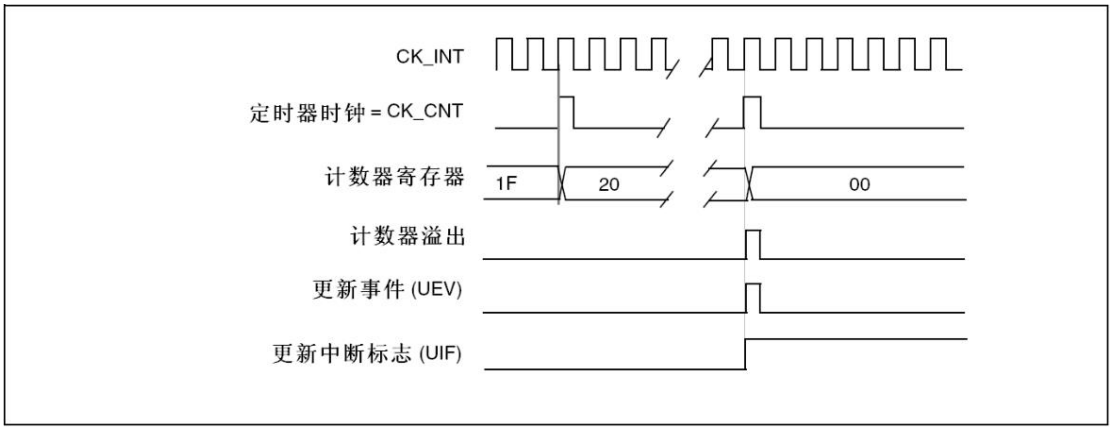


图 17-7

计数器时序图，当 ARPE=0 时的更新事件(TIMERx_ARR 没有预装载)

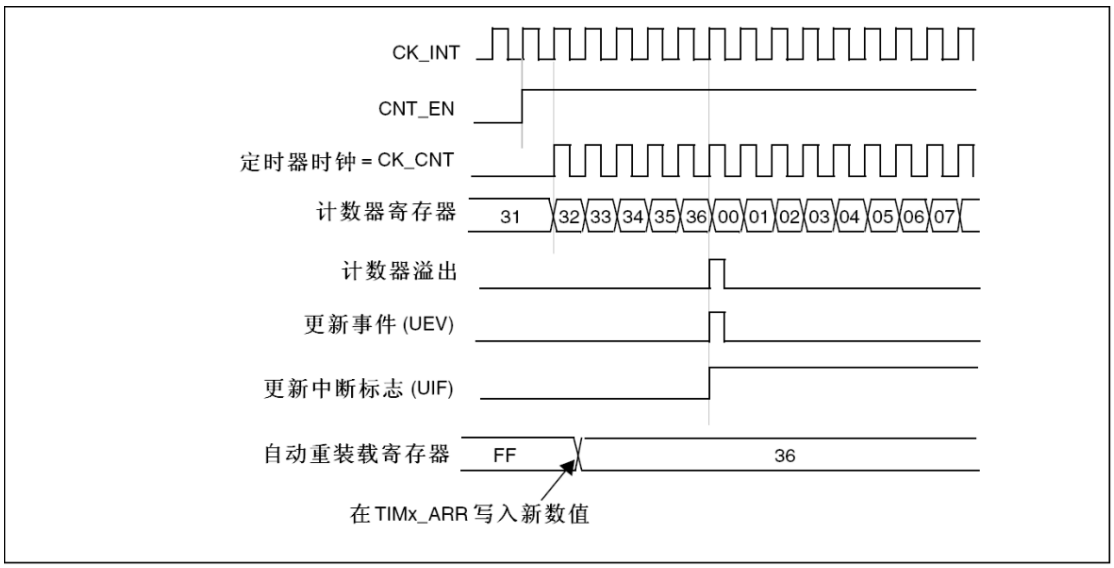


图 17-8

计数器时序图，当 ARPE=1 时的更新事件(预装载 TIMERx_ARR)

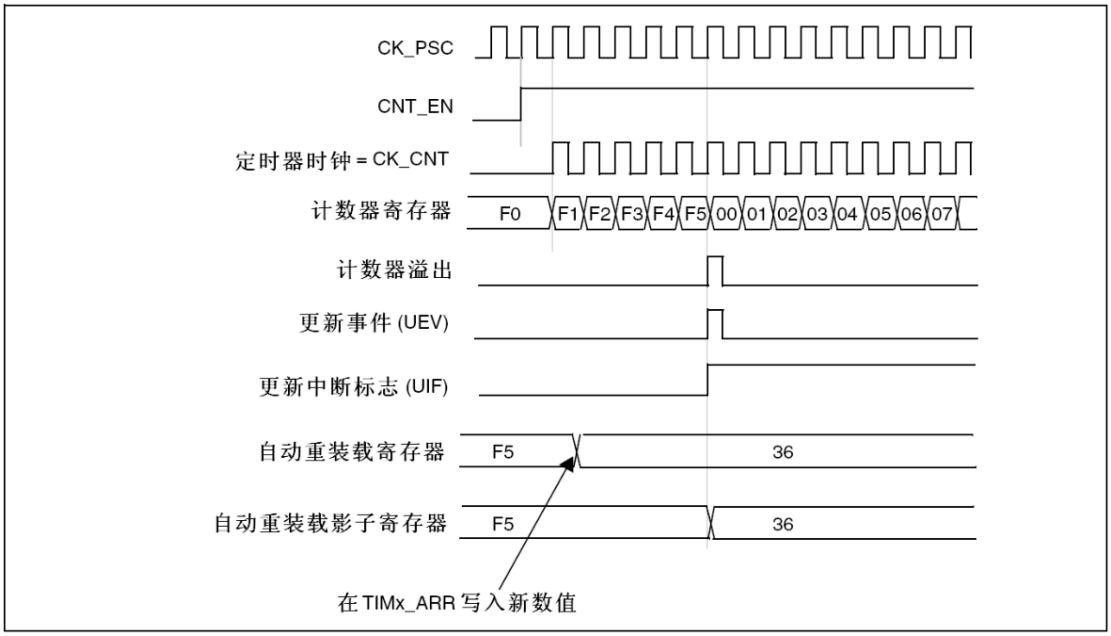


图 17-9

17.3.3. 时钟源

计数器的时钟由内部时钟(CK_INT)提供。

TIMERx_CR1 寄存器的 CEN 位和 TIMERx_EGR 寄存器的 UG 位是实际的控制位，(除了 UG 位被自动清除外)只能通过软件改变它们。一旦置 CEN 位为’1’，内部时钟即向预分频器提供时钟。

下图示出控制电路和向上计数器在普通模式下，没有预分频器时的操作。

普通模式时序图，内部时钟分频系数为 1

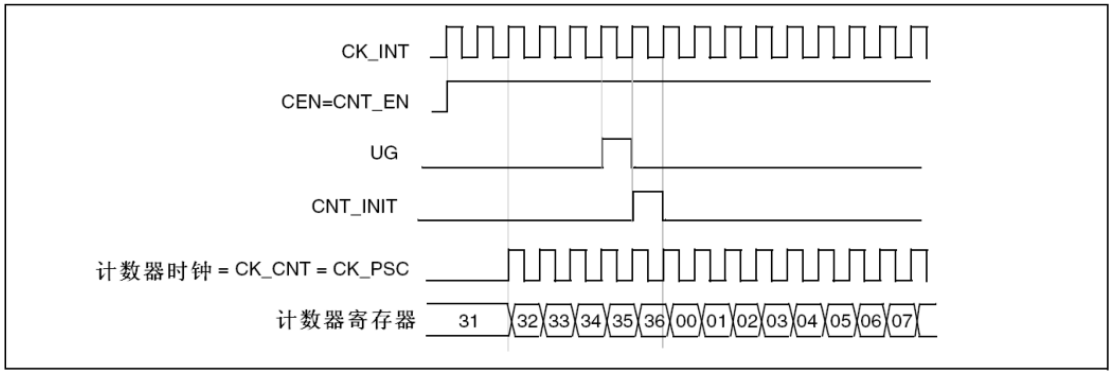


图 17-10

17.4. 寄存器描述

17.4.1. 寄存器列表

Base address：0x4002 0000

Name	Offset	Reset	Description
TIMER7_CR1	0x310	32'h0	TIMER1 控制寄存器 1
TIMER7_CR2	0x314	32'h0	TIMER1 控制寄存器 2
TIMER7_DIER	0x31C	32'h0	TIMER1 DMA/中断使能寄存器
TIMER7_SR	0x320	32'h0	TIMER1 状态寄存器
TIMER7_EGR	0x324	32'h0	TIMER1 事件产生寄存器
TIMER7_CNT	0x328	32'h0	TIMER1 计数器
TIMER7_PSC	0x32C	32'h0	TIMER1 预分频器
TIMER7_ARR	0x330	32'hFFFF	TIMER1 自动重载寄存器

17.4.2. 寄存器详细说明

17.4.2.1. 控制寄存器 1（TIMER7_CR1）

Width	Name	Reset	Property	Description
31:8	Reserved	-	-	-
7	ARPE	1'b0	RW	自动重装载预装载允许位 0: TIMERx_ARR 寄存器没有缓冲 1: TIMERx_ARR 寄存器被装入缓冲器。
6:4	Reserved	-	-	-
3	OPM	1'b0	RW	单脉冲模式 0: 在发生更新事件时，计数器不停止 1: 在发生下一次更新事件(清除 CEN 位)时，计数器停止
2	URS	1'b0	RW	更新请求源 0: 如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求:

				– 计数器溢出 – 设置 UG 位 1: 如果使能了更新中断或 DMA 请求，则只有计数器溢出才产生更新中断或 DMA 请求。
1	UDIS	1'b0	RW	禁止更新 软件通过该位允许/禁止 UEV 事件的产生 0: 允许 UEV。更新(UEV)事件由下述任一事件产生: – 计数器溢出 – 设置 UG 位 产生更新事件后，带缓冲的寄存器被加载为预加载数值。 1: 禁止 UEV。不产生更新事件，影子寄存器(ARR、PSC)保持它们的值。如果设置了 UG 位发出了一个硬件复位，则计数器和预分频器被重新初始化。
0	CEN	1'b0	RW	使能计数器 0: 禁止计数器 1: 使能计数器

17.4.2.2. 控制寄存器 2（TIMER7_CR2）

Width	Name	Reset	Property	Description
31:7	Reserved	-	-	-
6:4	MMS	3'b0	RW	主模式选择 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下: b000: 复位 – TIMERx_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式)，则 TRGO 上的信号相对实际的复位会有一个延迟。 b001: 使能 – 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。 b010: 更新 – 更新事件被选为触发输入(TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。
3:0	Reserved	-	-	-

17.4.2.3. DMA/中断使能寄存器（TIMER7_DIER）

Width	Name	Reset	Property	Description
31:9	Reserved	-	-	-
8	UDE	1'b0	RW	允许更新的 DMA 请求 0: 禁止更新的 DMA 请求 1: 允许更新的 DMA 请求
7:1	Reserved	-	-	-

0	UIE	1'b0	RW	允许更新中断 0: 禁止更新中断 1: 允许更新中断

17.4.2.4. 状态寄存器（TIMER7_SR)

Width	Name	Reset	Property	Description
31:1	Reserved	-	-	-
0	UIF	1'b0	RC	更新中断标记 当产生更新事件时该位由硬件置 1。它由软件清 0。 0: 无更新事件产生； 1: 产生了更新中断。下述情况下由硬件设置该位： – 计数器产生上溢或下溢并且 TIMERx_CR1 中的 UDIS=0； – 如果 TIMERx_CR1 中的 URS=0 并且 UDIS=0，当使用 TIMERx_EGR 寄存器的 UG 位重新初始化计数器 CNT 时。

17.4.2.5. 事件产生寄存器（TIMER7_EGR)

Width	Name	Reset	Property	Description
31:1	Reserved	-	-	-
0	UG	1'b0	WO	产生更新事件 该位由软件置 1，由硬件自动清 0。 0: 无动作； 1: 重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清 0(但是预分频系数不变)。

17.4.2.6. 计数器（TIMER7_CNT)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	CNT	16'b0	RW	计数器的值

17.4.2.7. 预分频器（TIMER7_PSC)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	PSC	16'b0	RW	预分频器的值 计数器的时钟频率(CK_CNT)等于 fCK_PSC /(PSC[15:0]+1)。 在每一次更新事件时，PSC 的数值被传送到实际的预分频寄存器中。

17.4.2.8. 自动重装载寄存器（TIMER7_ARR)

Width	Name	Reset	Property	Description

31:16	Reserved	-	-	-
15:0	ARR	16'hffff	RW	自动重装载的值 ARR 包含了将要装载入实际的自动重装载寄存器的值。 当自动重装载的值为空时，计数器不工作。

18. 数模转换器模块（ADC）

18.1. 模块介绍

该模块是一个 12 位的逐次逼近式的 ADC 控制器，ADC 支持单通道扫描和多通道扫描。转换的结果可以左对齐或右对齐的方式存储在 16 位寄存器中。ADC 的启动包括软件启动以及其他片内外设启动（TIMER 触发启动）。

18.2. 功能特点

- 支持 DMA 保存采样数据
- 最高 12 位可编程分辨率的 SARADC，多达 13 路外部输入通道通道采样时间，分辨率可软件配置
- 支持最高采样率为 1.2Msps
- 转换结束、注入转换结束和发生阈值看门狗事件时产生中断
- 支持 DMA 传输数据
- 可选多个 A/D 转换开始条件
  - 软件启动
  - 外部触发启动（TIMER/GPIO）

18.3. 功能说明

18.3.1. ADC 开关控制

通过设置寄存器 ADC_CFG 的 ADEN 和 ADC_LDOEN 位可给 ADC 上电。当第一次设置 ADEN 和 ADC_LDOEN 位时，它将 ADC 从断电状态下唤醒。ADC 上电延迟一段时间后（>=10us），设置采样通道之后，设置寄存器 ADC_CFG 的 ADST 位开始进行转换。软件可通过清除 ADST 位停止转换，或等所有通道转换结束自动清除。设置 ADEN 和 ADC_LDOEN 为 0 可置于断电模式。

18.3.2. 通道选择

包含 13 个外部通道。

18.3.3. 单通道采样模式

在单通道采样模式下，A/D 转换器只对使能的通道采样一次，并把结果保存在寄存器 ADC_DATA 中，操作步骤如下

- 根据需要采样的通道配置寄存器 ADC_RSEQCFGx/ADC_JSEQCFGx
- 转换完成后，寄存器 ADC_STA 中的 ADIF 标志位置 1，如果使能了中断，该标志置 1 时会进入中断函数。
- 软件可轮询 ADIF 标志位为 1 或者在中断函数中读取寄存器 ADC_DATA 获得采样值

18.3.4. 多通道采样模式

在多通道采样模式下，A/D 转换器对寄存器 ADC_RSEQCFGx/ADC_JSEQCFGx 中使能的通道依次扫描，操作步骤如下:

- 配置存储采样数据的 SRAM 起始地址。
- 使能寄存器 ADC_CR 的 DMAEN 使 ADC 工作在 DMA 模式下。
- 软件或外部触发寄存器 ADCFG 的 ADST 开始转换，外部触发可软件配置触发延时。
- 每路 A/D 转换完成后，A/D 转换数值将有序装载到相应的 SRAM 地址中。
- 当所有通道转换都完成后产生 DMA 完成标志位，如果使能了中断，则产生中断。
- 只要 ADST 位保持为 1，则转换未结束，当 ADST 位被清 0，A/D 转换停止，A/D 转换器进入空闲状态。

18.3.5. 采样频率设置

ADC 的时钟 ADC_CLK 由系统时钟分频得到，分频系数可通过设置寄存器 ADC_CFG 的 ADCPRE 位

来确定。

连续采样时，每次转换固定需要 14+(SMPCYC+1)个 ADC_CLK 周期，则 ADC 采样率计算如下：

ADC 采样率= F_{pclk} /(ADCPRE+1)/(14+(SMPCYC+1))，理论上 ADC 支持最快采样率为 1MSPS。

18.3.6. 外部触发转换

ADC 转换可以由定时器触发，如果设置了寄存器 ADC_CR 的 TRGEN 位，就可以使用定时器触发转换。通过设置 TRGSEL 位可以选择触发源。

外部触发可设置延时控制，具体参考 ADC_CR[14:12]的 JSEQ_TRG_DLY 和 ADC_CR[6:4]的 RSEQ_TRG_DLY 的描述。

18.3.7. 阈值看门狗

如果被 ADC 转换的模拟电压低于低阈值或高于高阈值，阈值看门狗状态位被设置。阈值位于 ADC_WDGCFG0 和 ADC_WDGCFG0 寄存器的最低 12 个有效位中。通过设置 ADC_CR 寄存器的 WDG_IE 位以允许产生相应中断。

阈值独立于由 ADC_CR 寄存器上的 ALIGN 位选择的数据对齐模式。比较是在对齐之前完成的。通过配置 ADC_WDGCFG0 寄存器，阈值看门狗可以作用于 1 个或多个通道，如表 2 所示。

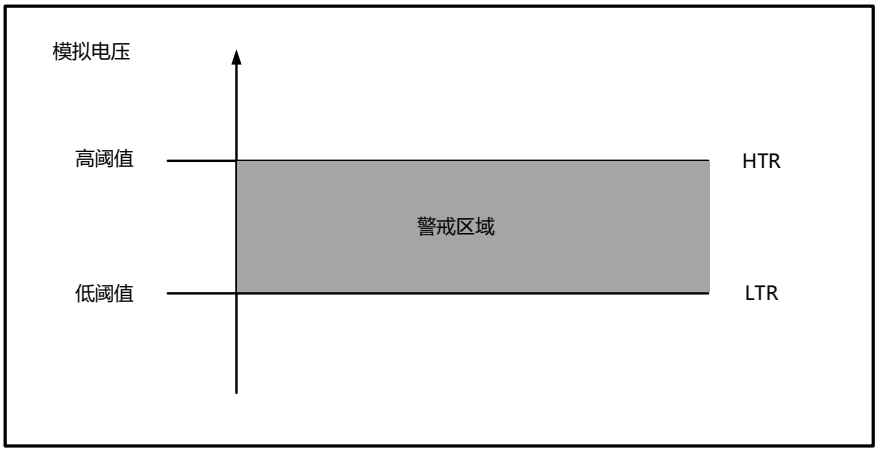


图 18-1 阈值看门狗警戒区

表 18-3

阈值看门狗警戒的通道	ADC_WDGCFG0 寄存器控制位		
	CHMODE 位	RCH_EN 位	JCH_EN 位
无	任意值	0	0
所有注入通道	0	0	1
所有规则通道	0	1	0
所有注入和规则通道	0	1	1
单一的注入通道	1	0	1
单一的规则通道	1	1	0
单一的注入或规则通道	1	1	1

18.4. IO 映射

详细参考 3.2. 管脚定义 。

18.5. 模块框图与接口时序

18.5.1. ADC 模块接口结构图

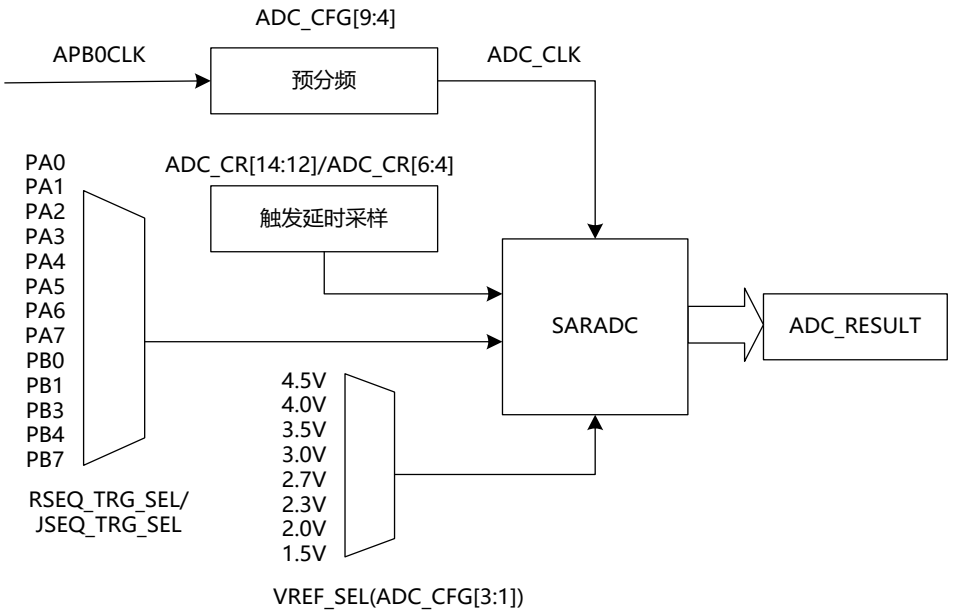


图 18-2 ADC 模块接口架构图

18.6. 时钟与复位

18.6.1. 时钟介绍

本模块的时钟来源只有系统时钟，在 sleep 或 stop 模式下，本模块时钟停止。

18.6.2. 复位介绍

该模块的复位源有两个，分别是系统复位和软件复位，其中软件复位由系统寄存器 SYSCON0[19]控制。

18.7. 寄存器描述

18.7.1. 寄存器列表

Base address：0x4002_0000

Name	Offset	Reset	Description
ADC_CFG	0x0140	32'h40fe	ADC 配置寄存器
ADC_CR	0x0144	32'h00400000	ADC 控制寄存器
ADC_STA	0x0148	32'h0	ADC 状态寄存器
ADC_WDGCFG0	0x014C	32'h0	ADC 看门狗配置寄存器 0
ADC_WDGCFG1	0x0150	32'h0	ADC 看门狗配置寄存器 1
ADC_RSEQCFG0	0x0154	32'h0	ADC 规则序列配置寄存器 0
ADC_RSEQCFG1	0x0158	32'h0	ADC 规则序列配置寄存器 1
ADC_RSEQCFG2	0x015C	32'h0	ADC 规则序列配置寄存器 2
ADC_JSEQCFG	0x0160	32'h0	ADC 注入序列配置寄存器
ADC_JDATA0	0x0164	32'h0	ADC 注入数据寄存器 0
ADC_JDATA1	0x0168	32'h0	ADC 注入数据寄存器 1

ADC_JDATA2	0x016C	32'h0	ADC 注入数据寄存器 2
ADC_JDATA3	0x0170	32'h0	ADC 注入数据寄存器 3
ADC_DATA	0x0174	32'h0	ADC 规则数据寄存器
ADC_OFFSET	0x0178	32'h0	ADC 补偿值寄存器
ADC_BOUND	0x017C	32'h0	ADC 补偿选择寄存器
ADC_BUFCAL	0x018C	32'h0	ADC 偏移校准控制寄存器

18.7.2. 寄存器详细说明

18.7.2.1. 配置寄存器（ADC_CFG)

Width	Name	Reset	Property	Description
31:29	Reserved	-	-	-
28:27	Reserved	-	-	-
26	Reserved	-	-	-
25	Reserved	-	-	-
24	Reserved	-	-	-
23	BUFSE	1'b0	RW	ADC rail to rail 输入控制 0: 不使能 1: 使能
22	VREF_AVCC_EN	1'b0	RW	AVCC 参考源使能控制 0: 不使能 1: 使能
21	VCM_OEN	1'b0	RW	内部 VCM 输出到 IO 使能控制(PB0) 0: 不使能 1: 使能
20	VCM_EN	1'b0	RW	内部 VCM 产生使能控制 0: 不使能 1: 使能
19	LDO_EN	1'b0	RW	ADC 模块内部 LDO 0: 不使能 1: 使能
18:17	Reserved	-	-	-
16:13	SMPCYC	4'b2	RW	两次转换之间的间隔（需配置 1 或以上） （n+1）* ADCCLK
12:10	Reserved	-	-	-
9:4	PSC	6'hf	RW	ADC 预分频 0: 2 分频 其它: (n+1)分频
3:1	VREF_SEL	3'b111	RW	ADC 参考电压选择 b000: 1.5V b001: 2.0V b010: 2.3V b011: 2.7V b100: 3.0V b101: 3.5V b110: 4.0V b111: 4.5V

0	EN	1'b0	RW	ADC 使能 0: 不使能 1: 使能

18.7.2.2. 控制寄存器（ADC_CR）

Width	Name	Reset	Property	Description
31	SW_RST	1'b0	WO	软件复位模块内部状态机 0: 无效 1: 有效
30:28	Reserved	-	-	-
27	OVR_IE	1'b0	RW	Overrun 中断使能位 0: 不使能 1: 使能 ADC_DATA 和 ADC_JDATAx 里任一 OVR_PEND 位为 1 且 OVR_IE 为 1 时产生中断
26	WDG_IE	1'h0	RW	阈值看门狗中断使能位。 0: 中断不使能 1: 中断使能 WDG_PEND 为 1 且 WDG_IE 为 1 时产生中断
25	IE_MODE	1'b0	RW	A/D 中断模式位 0: RSEQ_ENDIF 置位且 RSEQ_IE 置位时产生中断，或 JSEQ_ENDIF 置位且 JSEQ_IE 置位时产生中断； 1: RSEQ_PEND 任一位置位且 RSEQ_IE 置位时产生中断，或 JSEQ_PEND 任一位置位且 JSEQ_IE 置位时产生中断。
24	JSEQ_IE	1'b0	RW	注入通道 A/D 转换中断使能位 0: 不使能 1: 使能
23	RSEQ_IE	1'b0	RW	规则通道 A/D 转换中断使能位 0: 不使能 1: 使能
22	CAL_EN	1'b1	RW	硬件数据校准使能位 0: 无效 1: 有效
21	JDMA_EN	1'b0	RW	注入通道 DMA 使能 0: 不使能 1: 使能
20	RDMA_EN	1'b0	RW	规则通道 DMA 使能 0: 不使能 1: 使能
19	CTU	1'b0	RW	连续转换（Continuous conversion） 0: 单次转换模式，触发后只转换一遍 1: 连续转换模式，触发后一直转换，直到软件把 CTU 配置为 0
18	AUTOJC	1'b0	RW	转换完规则通道后，自动启动注入通道组转换。 0: 关闭自动的注入通道组转换

				1: 开启自动的注入通道组转换
17	DATASIGN	1'b0	RW	数据扩张位选择，影响 ADC_DATA 的最高 4 位或最低 4 位
16	ALIGN	1'b0	RW	数据对齐 0: 左对齐 1: 右对齐
15	JSEQ_KST	1'b0	RW	注入通道的转换开始。 软件触发模式下对该位写 1 或硬件触发模式下有触发信号发生，该位都会被置 1，表示 A/D 转换开始。转换完成后该位会被硬件自动清零。 读: 0: A/D 空闲中 1: A/D 转换中
14:12	JSEQ_TRG_DLY	3'b0	RW	注入通道外部触发延时采样 在触发信号产生后，延时 N 个 PCLK 的时钟周期再开始采样 0: 不延时 1: 4 个周期 2: 16 个周期 3: 32 个周期 4: 64 个周期 5: 128 个周期 6: 256 个周期 7: 512 个周期
11:8	JSEQ_TRG_SEL	3'b0	RW	注入通道的触发源选择: 0: 软件触发 1: TIMER1 CC1 触发 2: TIMER1 CC2 触发 3: TIMER1 CC3 触发 4: TIMER1 CC4 触发 5: TIMER1 TRGO 触发 6: TIMER2 TRGO 触发 7: TIMER7 TRGO 触发 8: TIMER4 CC1 和 TIMER4_CC2 触发 9: TIMER5 CC1 触发 10: TIMER6 CC2 触发 11: TIMER4 TRGO 触发 12: TIMER5 TRGO 触发 13: TIMER6 TRGO 触发 14: GPIO 触发 0 15: GPIO 触发 1
7	RSEQ_KST	1'b0	RW	规则通道的转换开始。 软件触发模式下对该位写 1 或硬件触发模式下有触发信号发生，该位都会被置 1，表示 A/D 转换开始。转换完成后该位会被硬件自动清零。 读: 0: A/D 空闲中

				1: A/D 转换中
6:4	RSEQ_TRG_DLY	3'b0	RW	规则通道外部触发延时采样 在触发信号产生后，延时 N 个 PCLK 的时钟周期再开始采样 0: 不延时 1: 4 个周期 2: 16 个周期 3: 32 个周期 4: 64 个周期 5: 128 个周期 6: 256 个周期 7: 512 个周期
3:0	RSEQ_TRG_SEL	3'b0	RW	规则通道的触发源选择: 0: 软件触发 1: TIMER1 CC1 触发 2: TIMER1 CC2 触发 3: TIMER1 CC3 触发 4: TIMER1 CC4 触发 5: TIMER1 TRGO 触发 6: TIMER2 TRGO 触发 7: TIMER7 TRGO 触发 8: TIMER4 CC1 和 TIMER4_CC2 触发 9: TIMER5 CC1 触发 10: TIMER6 CC2 触发 11: TIMER4 TRGO 触发 12: TIMER5 TRGO 触发 13: TIMER6 TRGO 触发 14: GPIO 触发 0 15: GPIO 触发 1

18.7.2.3. 状态寄存器（ADC_STA）

Width	Name	Reset	Property	Description
31:28	Reserved	-	-	-
27:24	CHANNEL	4'h0	RO	当前转换通道 读: BUSY = 1 时，表示进行转换中的通道
23	BUSY	1'b0	RO	忙/空闲（Busy）标志 0: A/D 转换器空闲 1: A/D 转换器忙碌
22	WDG_PEND	1'h0	RC	阈值看门狗标志位。 0: 没有发生阈值看门狗事件 1: 发生阈值看门狗事件 软件也可以对该位写 1 清 0
21	JSEQ_ENDIF	1'h0	RC	注入通道 A/D 转换结束标志位，该位由硬件在所有注入通道采样结束时设置 1 0: 注入通道 A/D 转换未全完成 1: 所有注入通道 A/D 转换完成

				软件也可以对该位写 1 清 0
20	RSEQ_ENDIF	1'h0	RC	规则通道 A/D 转换结束标志位，该位由硬件在所有规则通道采样结束时设置 1 0: 规则通道 A/D 转换未全完成 1: 所有规则通道 A/D 转换完成 软件也可以对该位写 1 清 0
19:16	JSEQ_PEND	4'b0	RC	注入通道转换完成标志位。 每个比特对应注入通道序列的 1 个转换，如 JSEQ_PEND[0]表示 JSEQ0 转换完成， JSEQ_PEND[1]表示 JSEQ1 转换完成， JSEQ_PEND[3]表示 JSEQ3 转换完成。 0: 转换未完成 1: 转换完成 软件对该位写 1 清 0
15:0	RSEQ_PEND	16'b0	RC	规则通道转换完成标志位。 每个比特对应规则通道序列的 1 个转换，如 RSEQ_PEND[0]表示 RSEQ0 转换完成， RSEQ_PEND[1]表示 RSEQ1 转换完成， RSEQ_PEND[15]表示 RSEQ15 转换完成。 0: 转换未完成 1: 转换完成 软件对该位写 1 清 0

18.7.2.4. 看门狗配置寄存器 0（ADC_WDGCFG0）

Width	Name	Reset	Property	Description
31:19	Reserved	-	-	-
18:15	CHSEL	4'h0	RW	看门狗通道选择位 h0: ADC 模拟输入通道 0 h1: ADC 模拟输入通道 1 …… hE: ADC 模拟输入通道 14 hF: ADC 模拟输入通道 15
14	CHMODE	1'b0	RW	看门狗通道选择模式 0: 在所有的通道上使用看门狗 1: 在单一通道上使用看门狗
13	RCH_EN	1'b0	RW	在规则通道上开启看门狗。 0: 在规则通道上禁用看门狗 1: 在规则通道上使用看门狗
12	JCH_EN	1'b0	RW	在注入通道上开启看门狗。 0: 在注入通道上禁用看门狗 1: 在注入通道上使用看门狗
11:0	HTH	12'h0	RW	看门狗高阈值。 这些位定义了看门狗的阈值高限。

18.7.2.5. 看门狗配置寄存器 1（ADC_WDGCFG1）

Width	Name	Reset	Property	Description

31:12	Reserved	-	-	-
11:0	LTH	12'h0	RW	看门狗低阈值。 这些位定义了看门狗的阈值低限。

18.7.2.6. 规则序列配置寄存器 0（ADC_RSEQCFG0）

Width	Name	Reset	Property	Description
31:28	RSEQ7	4'h0	RW	规则序列索引 7 的通道配置
27:24	RSEQ6	4'h0	RW	规则序列索引 6 的通道配置
23:20	RSEQ5	4'h0	RW	规则序列索引 5 的通道配置
19:16	RSEQ4	4'h0	RW	规则序列索引 4 的通道配置
15:12	RSEQ3	4'h0	RW	规则序列索引 3 的通道配置
11:8	RSEQ2	4'h0	RW	规则序列索引 2 的通道配置
7:4	RSEQ1	4'h0	RW	规则序列索引 1 的通道配置
3:0	RSEQ0	4'h0	RW	规则序列索引 0 的通道配置 配置值是转换通道号 n（n=0~15），表示转换索引 0 时是转换通道 n。

18.7.2.7. 规则序列配置寄存器 1（ADC_RSEQCFG1）

Width	Name	Reset	Property	Description
31:28	RSEQ15	4'h0	RW	规则序列索引 15 的通道配置
27:24	RSEQ14	4'h0	RW	规则序列索引 14 的通道配置
23:20	RSEQ13	4'h0	RW	规则序列索引 13 的通道配置
19:16	RSEQ12	4'h0	RW	规则序列索引 12 的通道配置
15:12	RSEQ11	4'h0	RW	规则序列索引 11 的通道配置
11:8	RSEQ10	4'h0	RW	规则序列索引 10 的通道配置
7:4	RSEQ9	4'h0	RW	规则序列索引 9 的通道配置
3:0	RSEQ8	4'h0	RW	规则序列索引 8 的通道配置

18.7.2.8. 规则序列配置寄存器 2（ADC_RSEQCFG2）

Width	Name	Reset	Property	Description
31:12	Reserved	-	-	-
11:8	RSEQ_TNUM	4'h0	RW	规则序列每次触发转换的索引数目（总的索引数量是 RSEQ_EIDX-RSEQ_SIDX+1，这里配置的每次转换的索引数目不能大于总的索引数量） 0: 1 个转换 1: 2 个转换 …… 15: 16 个转换
7:4	RSEQ_EIDX	4'h0	RW	规则序列转换结束索引
3:0	RSEQ_SIDX	4'h0	RW	规则序列转换起始索引

18.7.2.9. 注入序列配置寄存器（ADC_JSEQCFG）

Width	Name	Reset	Property	Description
31:22	Reserved	-	-	-
21:20	JSEQ_TNUM	2'h0	RW	注入序列每次触发转换的索引数目（总的索引数

				量是 JSEQ_EIDX-JSEQ_SIDX+1，这里配置的每次转换的索引数目不能大于总的索引数量） 0: 1 个转换 1: 2 个转换 2: 3 个转换 3: 4 个转换
19:18	JSEQ_EIDX	2'h0	RW	注入序列转换结束索引
17:16	JSEQ_SIDX	2'h0	RW	注入序列转换起始索引
15:12	JSEQ3	4'h0	RW	注入序列索引 3 的通道配置
11:8	JSEQ2	4'h0	RW	注入序列索引 2 的通道配置
7:4	JSEQ1	4'h0	RW	注入序列索引 1 的通道配置
3:0	JSEQ0	4'h0	RW	注入序列索引 0 的通道配置

18.7.2.10. 注入数据寄存器（ADC_JDATAx）(x=0~3)

Width	Name	Reset	Property	Description
31:24	Reserved	-	-	-
23	VALID	1'b0	RO	DATA 有效标志位 0: DATA 有效 1: DATA 无效 启动采样或读取 ADC_DATA 寄存器时该位清零。
22	OVR_PEND	1'b0	RO	数据覆盖标志位 0: DATA 为最近一次转换结果 1: DATA 上一次数据被覆盖 新的转换结果转载至寄存器之前，若 DATA[15:0]数据没有被读取，OVR_PEND 将置“1”， 软件读 ADC_DATA 寄存器后，该位自动清 0.
21:20	Reserved	-	-	-
19:16	DATA_CHANNEL	4'h0	RO	显示当前数据所对应的通道 n=通道 n 的转换数据
15:0	DATA	16'h0	RO	12 位 A/D 转换结果 根据设置左对齐或者右对齐

18.7.2.11. 规则数据寄存器（ADC_DATA)

Width	Name	Reset	Property	Description
31:24	Reserved	-	-	-
23	VALID	1'b0	RO	DATA 有效标志位 0: DATA 有效 1: DATA 无效 启动采样或读取 ADC_DATA 寄存器时该位清零。
22	OVR_PEND	1'b0	RO	数据覆盖标志位 0: DATA 为最近一次转换结果 1: DATA 上一次数据被覆盖 新的转换结果转载至寄存器之前，若

				DATA[15:0]数据没有被读取，OVR_PEND 将置“1”， 软件读 ADC_DATA 寄存器后，该位自动清 0.
21:20	Reserved	-	-	-
19:16	DATA_CHANNEL	4'h0	RO	显示当前数据所对应的通道 n=通道 n 的转换数据
15:0	DATA	16'h0	RO	12 位 A/D 转换结果 根据设置左对齐或者右对齐

18.7.2.12. 偏移校准寄存器（ADC_BUFCAL）

Width	Name	Reset	Property	Description
31:8	Reserved	-	-	-
7:0	BUFCAL	8'h0	RW	模拟内部 buffer 偏移校准控制

19. 比较器（COMP0/1）

19.1. 模块介绍

比较器模块用于比较两个输入模拟电压的大小，并根据比较结果输出高/低电平。当正极输入端电压高于负极输入端电压时，比较器输出为高电平，当正极输入端电压低于负极输入端电压时，比较器输出为低电平。

19.2. 功能特点

- 比较结果可输出到 IO 端口或触发定时器
- 有 4 档迟滞电压可配置
- 具备唤醒能力
- 具有寄存器锁定机制

19.3. 功能说明

19.3.1. 比较器功能

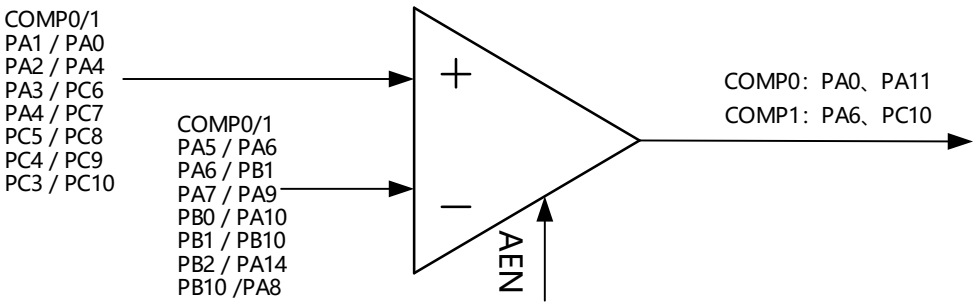


图 19-1 比较器

2 个比较器可独立配置，完成比较器功能。通过配置可以选择比较器的输出，连接到相应的端口。运用这种模式时需配置寄存器如下

- 通过配置 MUXP 选择一个正级输入 IO，并设置为模拟模式
- 通过配置 MUXN 选择一个负极输入 IO，并设置为模拟模式
- 如需要输出到 IO，选择一个输出 IO，配置为复用功能模式
- 根据需要配置迟滞 HYST 等级
- 根据需要配置输出极性是否取反 INVEN
- 打开比较器使能 AEN

19.3.2. 6bitDAC 比较输出

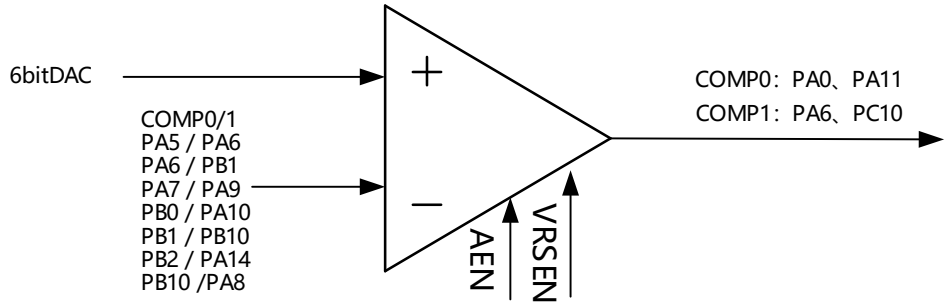


图 19-2 DAC 比较输出

- 通过配置 MUXP 选择内部 DAC 输入（MUXP=7）
- 通过配置 MUXN 选择一个负极输入 IO，并设置为模拟模式
- 如需要输出到 IO，选择一个输出 IO，配置为复用功能模式
- 选择内部 DAC 的参考源 VRSEL
- 打开内部 DAC 开关（即 VRSEN=1）
- 使能 AEN，打开比较器

19.4. 时钟与复位

19.4.1. 时钟介绍

本模块的时钟来源只有系统时钟，在 sleep 或 stop 模式下，本模块时钟仍然有效。

19.4.2. 复位介绍

该模块的复位源有两个，分别是系统复位和软件复位，其中软件复位由系统寄存器 SYSCON0[12]控制。

19.5. 寄存器描述

19.5.1. 寄存器列表

Base address：0x4002_0000

Name	Offset	Reset	Description
COMPx_CON	0x0370/ 0x0380	32'h0	COMPx 配置寄存器
COMPx_STA	0x0374/ 0x0384	32'h0	COMPx 状态寄存器

19.5.2. 寄存器详细说明

19.5.2.1. 配置寄存器（COMPx_CON）(x=0/1)

Width	Name	Reset	Property	Description
31	LOCK	1'b0	RW	寄存器锁定，该位写 1 后，寄存器所有位都被锁定为只读，包括 LOCK 位，直到产生系统复位或触发模块软件复位。 0: 不锁定 1: 锁定
30	Reserved	-	-	-
29	WKUPEN	1'b0	RW	比较器唤醒 0: 不使能 1: 使能
28:27	EDGSEL	2'b0	RW	边沿检测选择，检测到边沿 pend 位置 1 b00: 上升沿 b01: 下降沿 b1x: 上升沿与下降沿
26	INTEN	1'b0	RW	中断开关 0: 不使能 1: 使能
25:21	FILTNUM	5'b0	RW	滤波器系数 0: 不滤波

				其它: 滤波 N+1 个时钟周期
20	INVEN	1'b0	RW	比较器输出结果取反 0: 不取反 1: 取反
19:18	HYST	2'b0	RW	迟滞电压选择 b00: 0mV b01: 20mV b10: 30mV b11: 40mV
17:16	ILPS	2'b0	RW	低功耗控制 b00: 1/1 b01: 1/4 b10: 1/10 b11: 1/13
15:13	MUXN	3'b0	RW	负极输入 COMP0/1: b000: PA5/PA6 b001: PA6/PB1 b010: PA7/PA9 b011: PB0/PA10 b100: PB1/PB10 b101: PB2/PB14 b110: PB10/PA8 b111: Reserved
12:10	MUXP	3'b0	RW	正极输入 COMP0/1: b000: PA1/PA0 b001: PA2/PA4 b010: PA3/PC6 b011: PA4/PC7 b100: PC5/PC8 b101: PC4/PC9 b110: PC3/PC10 b111: 6bitDAC
17:10	Reserved	-	-	-
9:4	VSRS	6'b0	RW	内部 6bit DAC 电压挡位
3	VRSAO	1'b0	RW	内部 6bit DAC 通过 VRSAOUT 输出 0: 不使能 1: 使能
2	VRSEL	1'b0	RW	内部 6bit DAC 参考源 0: VCC 1: VCMINN
1	VRSEN	1'b0	RW	内部 6bit DAC 开关 0: 不使能 1: 使能
0	AEN	1'b0	RW	比较器模块开关 0: 不使能

				1: 使能

19.5.2.2. 状态寄存器（COMPx_STA）(x=0/1)

Width	Name	Reset	Property	Description
31:1	Reserved	-	-	-
0	PEND	1'b0	RW	比较器 PEND 位 0: 未检测到有效边沿 1: 检测到有效边沿 软件对该位写 1 可清除 PEND

20. 运算放大器（OPAM0/1）

20.1. 模块介绍

CIU32M011、CIU32M031 内置 OPAM 模块，可以灵活配置，适用于独立运算放大器、可编程增益放大器(PGA)，和跟随器等模式应用。内部的运算放大器可以配置为具有不同增益的组合放大器，也可以使用外部电阻进行级联。OPAM 的输入范围是 0V 到 VCC-1.5V，输出范围是 0.5V 到 VCC-0.5V。OPAM 的输出内部连接到 ADC 的输入通道用于模拟信号测量。

注：OPAMx 中的 x 表示第几个 OPAM。

20.2. 功能特点

- 可编程增益，有 8 个增益放大倍数可选
- 正负输入端有四种输入模式可供选择
- 多种工作模式可配置
  - 电压跟随器
  - 单端可编程增益模式
  - 差分可编程增益模式
  - 独立运放模式
- 可输入 VCM

20.3. 功能说明

OPAM 可以通过寄存器选择配置为各种不同的 PGA 模式，也可以配置为用户使用外部元件的 OPAM 功能。OPAM 的输出可以作为 ADC 的通道输入。

20.3.1. 电压跟随器

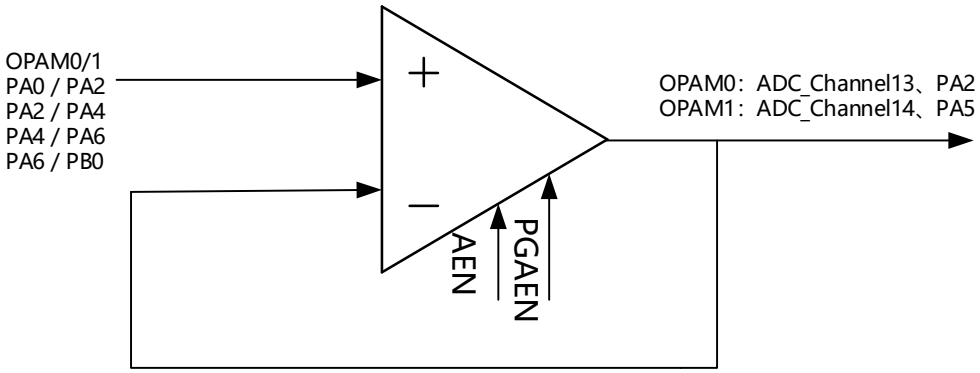


图 20-1 电压跟随器

电压跟随器是输出电压直接跟随输入电压，运用这种模式时需配置寄存器如下

- 使能正级输入（即 POSEN=1），并通过 POSSEL 选择一个正级输入 IO，IO 设置为模拟模式
- 使能负极输入（即 NEGEN=1），并选择固定输入（NEGSEL=3）
- 选择 OP2VAEN 使得输出连接到 ADC，选择 OP2VOEN 使得输出到 IO（模拟模式），两个输出通道可以独立设置，互不影响
- 设置为 PGA 功能（即 PGAEN=1）
- 使能 AEN，打开运算放大器

20.3.2. 单端可编程增益（PGA ）模式

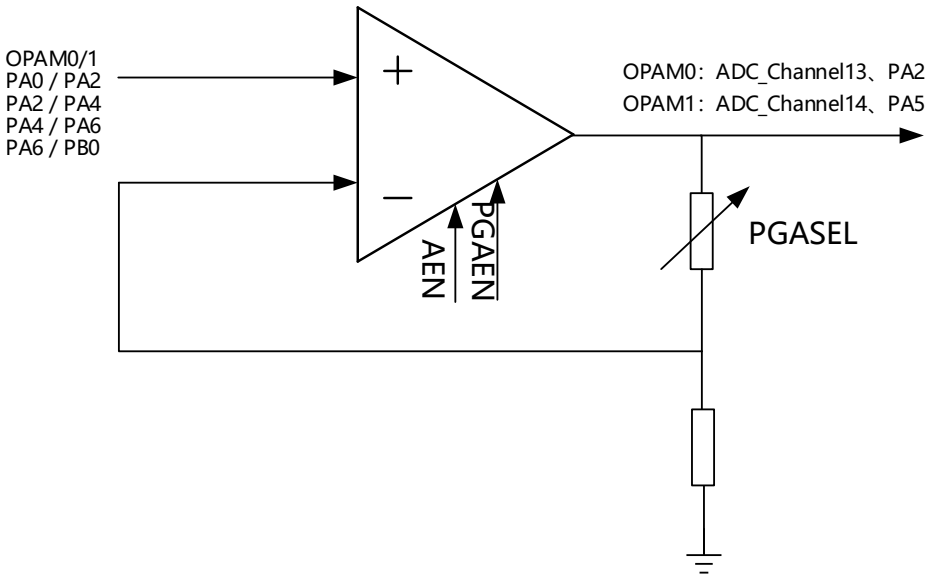


图 20-2 单端可编程增益模式

单端 PGA 模式，通过选择 PGASEL 的档位改变可变电阻的阻值，从而实现对输入电压的放大。支持 1/2/4/8/12/16/24/32 共 8 种放大倍数，可通过设置 PGASEL 进行增益选择。需配置寄存器如下

- 使能正级输入（即 POSEN=1），并通过 POSSEL 选择一个正级输入 IO，IO 设置为模拟模式
- 使能负极输入（即 NEGEN=1），并选择固定输入（NEGSEL=3）
- 选择 OP2VAEN 使得输出连接到 ADC，选择 OP2VOEN 使得输出到 IO（模拟模式），两个输出通道可以独立设置，互不影响
- 设置为 PGA 功能（即 PGAEN=1），通过 PGASEL 配置放大倍数
- 使能 AEN，打开运算放大器

20.3.3. 差分可编程增益（PGA ）模式

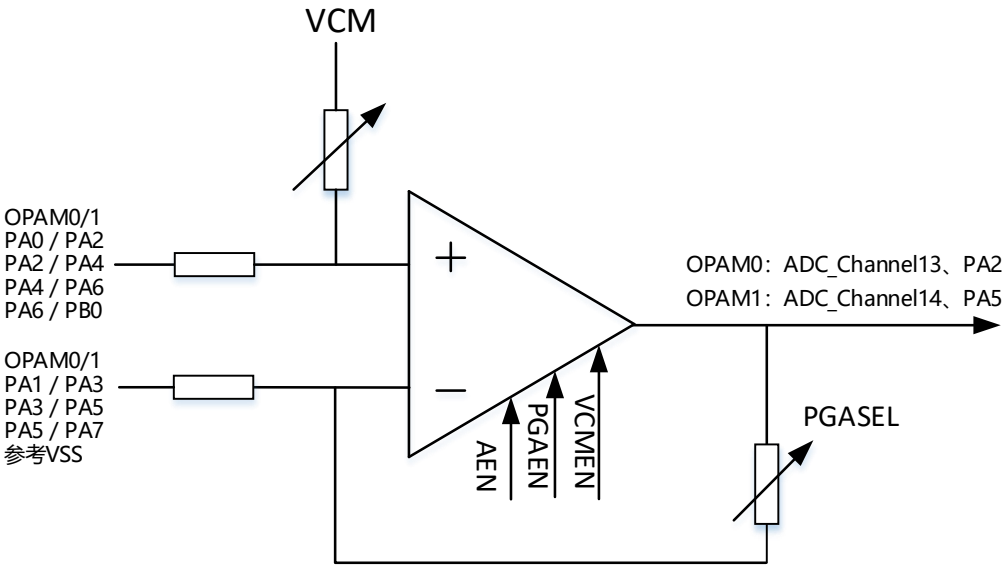


图 20-3 差分可编程增益模式

差分输入，单端输出的可编程增益模式，需要打开输入 VCM 电压的开关，即配置 VCMEN=1，同样通过调节可变电阻的阻值改变增益放大倍数。输入端内部等效电阻约 50K~160K，增益放大倍数越大内部

电阻越大。运用这种模式时需配置寄存器如下

- 使能正级输入（即 POSEN=1），并通过 POSSEL 选择一个正级输入 IO，IO 设置为模拟模式
- 使能负极输入（即 NEGEN=1），并通过 NEGSEL 选择一个负级输入 IO，IO 设置为模拟模式。注意当 NEGSEL=3 时，输入参考 VSS
- 选择 OP2VAEN 使得输出连接到 ADC，选择 OP2VOEN 使得输出到 IO（模拟模式），两个输出通道可以独立设置，互不影响
- 通过配置 ADC 寄存器使能内部 VCM
- 设置为 PGA 功能（即 PGAEN=1），通过 PGASEL 配置放大倍数
- 打开 VCM 开关（即 VCMEN=1）
- 使能 AEN，打开运算放大器

20.3.4. 独立运放模式

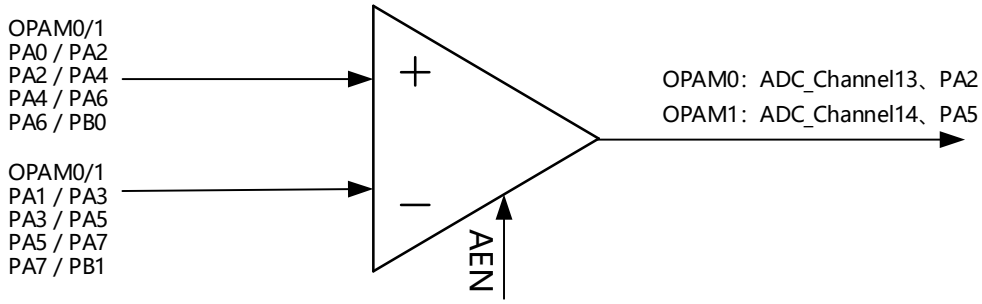


图 20-4 独立运放模式

独立运放模式，所有的端口引出，包括两个正/负极输入和一个输出，用户可通过外部电阻设置放大倍数。

20.3.5. 比较器模式

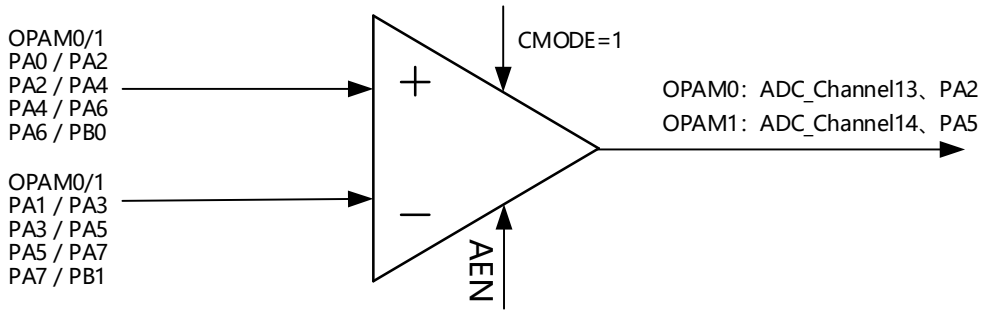


图 20-5 比较器模式

运算放大模块可以设置为比较器模式，运用这种模式时需配置寄存器如下

- 使能正级输入（即 POSEN=1），并通过 POSSEL 选择一个正级输入 IO，IO 设置为模拟模式
- 使能负极输入（即 NEGEN=1），并通过 NEGSEL 选择一个负级输入 IO，IO 设置为模拟模式
- 选择 OP2VAEN 使得输出连接到 ADC，选择 OP2VOEN 使得输出到 IO（模拟模式），两个输出通道可以独立设置，互不影响
- 设置为比较器功能（即 CMODE=1）
- 使能 AEN，打开运算放大器

20.4. 时钟与复位

20.4.1. 时钟介绍

本模块的时钟来源只有系统时钟，在 sleep 或 stop 模式下，本模块时钟仍然有效。

20.4.2. 复位介绍

该模块的复位源有两个，分别是系统复位和软件复位，其中软件复位由系统寄存器 SYSCON0[12]控制。

20.5. 寄存器描述

20.5.1. 寄存器列表

Base address：0x4002_0000

Name	Offset	Reset	Description
OPAMx_CON	0x0390/ 0x0398	32'h0	OPAMx 配置寄存器

20.5.2. 寄存器详细说明

20.5.2.1. 配置寄存器（OPAMx_CON）(x=0/1)

Width	Name	Reset	Property	Description
31	LOCK	1'b0	RW	寄存器锁定，该位写 1 后，寄存器所有位都被锁定为只读，包括 LOCK 位，直到产生系统复位或触发模块软件复位。 0: 不锁定 1: 锁定
29:25	Reserved	-	-	-
24:23	NEGSEL	2'h0	RW	负极输入选择 OPAM0/1: b00: PA1/PA3 b01: PA3/PA5 b10: PA5/PA7 b11: PA7/PB1
22:21	POSSEL	2'h0	RW	正极输入选择 OPAM0/1: b00: PA0/PA2 b01: PA2/PA4 b10: PA4/PA6 b11: PA6/PB0
20:18	PGASEL	3'h0	RW	单端正极/差分内部增益选择 h0: 1 倍/无效 h1: 2 倍/1 倍 h2: 4 倍/3 倍 h3: 8 倍/7 倍 h4: 12 倍/11 倍 h5: 16 倍/15 倍 h6: 24 倍/23 倍

				h7: 32 倍/31 倍
17:10	Reserved	-	-	-
9	Reserved	-	-	-
8	Reserved	-	-	-
7	CMODE	1'b0	RW	工作模式 0: 运放模式 1: 比较器模式
6	NEGEN	1'b0	RW	负极输入开关 0: 不使能 1: 使能
5	OP2VAEN	1'b0	RW	运放输出到 ADC 开关 0: 不使能 1: 使能
4	OP2VOEN	1'b0	RW	运放输出到 IO 开关 0: 不使能 1: 使能
3	PGAEN	1'b0	RW	内部增益开关 0: 不使能 1: 使能
2	POSEN	1'b0	RW	正极输入开关 0: 不使能 1: 使能
1	VCMEN	1'b0	RW	VCM 电压模式 0: 不使能 1: 使能
0	AEN	1'b0	RW	OPAM 模块开关 0: 不使能 1: 使能

21. 独立看门狗（WDT)

21.1. 模块介绍

CIU32M011、CIU32M031 内置看门狗，为系统提供了更高的安全性，时间的精确性和使用的灵活性。看门狗可用来检测和解决软件错误引起的故障，当计数值超出阈值时，产生系统复位或者中断，保证系统的安全性。看门狗由专门的低速时钟驱动，即使主时钟发生故障它也不受影响。支持修改分频系数改变超时时间，发生超时事件的时候可以选择中断或者复位。

21.2. 功能特点

- 分频系数可配置
- 可工作在中断模式或复位模式
- 可定时唤醒系统

21.3. 模块框图

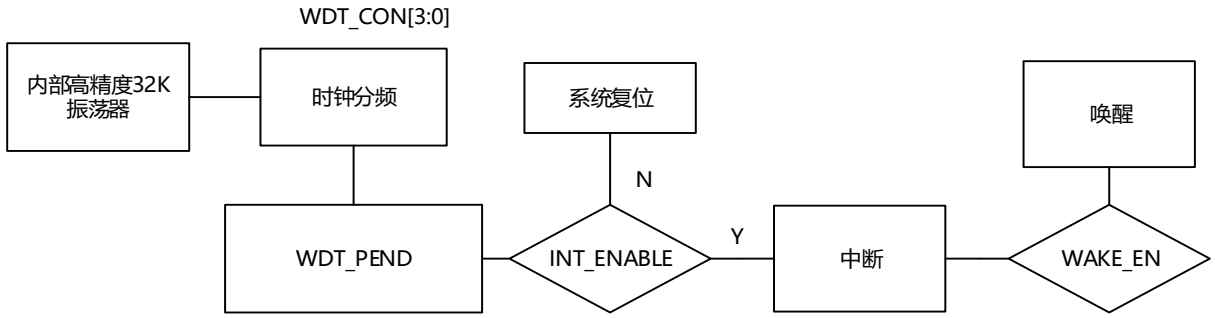


图 21-1 看门狗模块框图

21.4. 时钟与复位

21.4.1. 时钟介绍

该模块有两个时钟源，分别是系统时钟和看门狗时钟 wdt_clk，其中系统时钟用于配置模块寄存器，wdt_clk 为内部 LIRC_256K 分频得到的 32KHz 时钟，用于看门狗计数。

21.4.2. 复位介绍

该模块有两个复位源，分别是系统复位和看门狗模块复位，其中看门狗模块复位由外部 IO 或者掉电复位触发。

21.5. 寄存器描述

21.5.1. 寄存器列表

Base address：0x4002_0000

Name	Offset	Reset	Description
WDT_CON	0x0190	32'h18	WDT 控制寄存器
WDT_KEY	0x0194	32'h0	WDT 秘钥寄存器

21.5.2. 寄存器详细说明

21.5.2.1. WDT_CON

Width	Name	Reset	Property	Description
31:10	Reserved	-	-	-
9	WAKE_EN	1'b0	RO	唤醒使能
8:7	Reserved	-	-	-
6	WDT_PEND	1'b0	RO	看门狗计数满标志 WDG_KEY 写 0xAAAA，清除 WDT_PEND
5	IE	1'b0	RO	中断使能 0: 计数满时复位系统 1: 计数满时产生中断
4	WDT_EN_STA	1'b1	RO	看门狗使能状态 0: 看门狗关闭 1: 看门狗使能
3:0	WDT_PSR	4'b1000	RW	分频系数，每次配置该位域之前必须先写 WDG_KEY=0x5555 b0000: 不分频 b0001: 2 b0010: 4 b0011: 8 b0100: 16 b0101: 32 b0110: 64 b0111: 128 b1000: 256 b1001: 512 b1010: 1024 b1011: 2048 b1100: 4096 b1101: 8192 b1110: 16384 b1111: 32768 看门狗复位时间=1/32K*256*分频系数

21.5.2.2. WDT_KEY

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	KEY	16'h0	WO	KEY[15: 0]: 键值(只写寄存器，读出值为 0x0000) 软件必须以一定的间隔写入 0xAAAA，否则，当计数器为 0 时，看门狗会产生复位。当 WDT_PEND 为 1 的时候，写 0xAAAA 会清除 WDT_PEND。 写入 h5555 表示允许访问 WDT_PSR 写入 hCCCC，启动看门狗工作

				写入 hDDDD，关闭看门狗 写入 hAAAA，清除 WDT_PEND 写入 h55AA，开启中断使能 写入 hAA55，关闭中断使能 写入 h5A5A，开启 WAKE_EN 写入 hA5A5，关闭 WAKE_EN

22. 窗口看门狗（WWDG)

22.1. 模块介绍

窗口看门狗通常被用来监测，由外部干扰或不可预见的逻辑条件造成的应用程序背离正常的运行序列而产生的软件故障。除非递减计数器的值在 T[6]位变成 0 前被刷新，看门狗电路在达到预置的时间周期时，会产生一个 MCU 复位。在递减计数器达到窗口寄存器的数值之前，如果 7 位的递减计数器的数值被刷新，也将产生一个 MCU 复位。这表明递减计数器需要在一个有限的时间窗口中被刷新。

22.2. 功能特点

- 可编程的自由运行递减计数器
- 条件复位
  - 当递减计数器的值小于 0x40，(若看门狗被启动)则产生复位。
  - 当递减计数器在窗口外被重新装载，(若看门狗被启动)则产生复位。
- 如果启动了看门狗并且允许中断，当递减计数器等于 0x40 时产生早期唤醒中断(EWI)，它可以被用于重装载计数器以避免 WWDG 复位。

22.3. 功能说明

如果看门狗被启动，并且当 7 位(T[6:0])递减计数器从 0x40 翻转到 0x3F(T[6]位清零)时，则产生一个复位。如果软件在计数器值大于窗口寄存器中的数值时重新装载计数器，将产生一个复位。

应用程序在正常运行过程中必须定期地写入 WWDG_CR 寄存器以防止 MCU 发生复位。只有当计数器值小于窗口寄存器的值时，才能进行写操作。储存在寄存器 WWDG_CR 中的数值必须在 0xFF 和 0xC0 之间。

22.3.1. 启动看门狗

在系统复位后，看门狗总是处于关闭状态，设置寄存器 WWDG_CR 的 WDGA 位能够开启看门狗，随后它不能再被关闭，除非发生复位。

22.3.2. 递减计数器

递减计数器处于自由运行状态，即使看门狗被禁止，递减计数器仍继续递减计数。当看门狗被启用时，T6 位必须被设置，以防止立即产生一个复位。T[5:0]位包含了看门狗产生复位之前的计时数目；复位前的延时时间在一个最小值和一个最大值之间变化，这是因为写入 WWDG_CR 寄存器时，预分频值是未知的。

寄存器 WWDG_CFR 中包含窗口的上限值：要避免产生复位，递减计数器必须在其值小于窗口寄存器的数值并且大于 0x3F 时被重新装载。

另一个重装载计数器的方法是利用早期唤醒中断(EWI)。设置 WWDG_CFR 寄存器中的 EWI 位开启该中断。当递减计数器到达 0x40 时，则产生此中断，相应的中断服务程序(ISR)可以用来加载计数器以防止 WWDG 复位。在 WWDG_SR 寄存器中写 “1” 可以清除该中断。

注：可以用 T6 位产生一个软件复位(设置 WDGA 位为 “1”，T6 位为 “0”）。

22.3.3. 看门狗超时时间

可以使用下面提供的公式计算窗口看门狗的超时时间。

计算超时的公式如下：

$$T_{WWDDG}=T_{PCLK1}\times 4096\times 2^{WDGTB}\times (T[5:0]+1);\quad (ms)$$

其中：

T_{WWDG}： WWDG 超时时间

T_{PCLK1} ： APB1 以 ms 为单位的时钟间隔

注：当写入 WWDG_CR 寄存器时，始终置 T6 位为 ‘1’ 以避免立即产生一个复位。

在 PCLK1=36MHz 时的最小-最大超时值

WDGTB	最小超时值	最大超时值
0	113μs	7.28ms

1	227μs	14.56ms
2	455μs	29.12ms
3	910μs	58.25ms

22.4. 模块框图

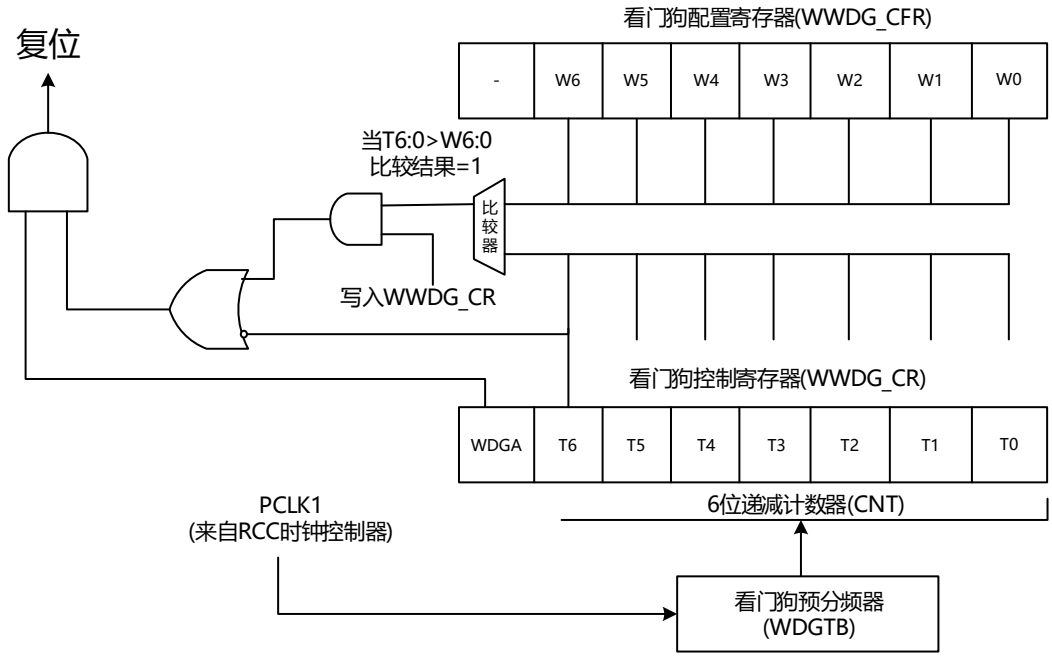


图 22-1 窗口看门狗模块框图

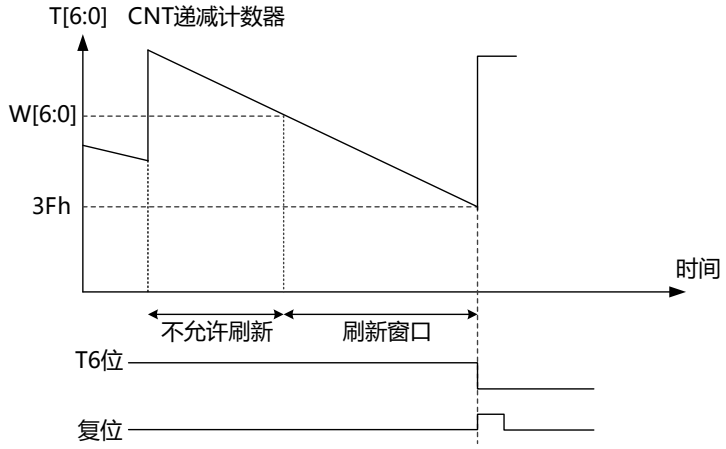


图 22-2 窗口看门狗时序图

22.5. 寄存器描述

22.5.1. 寄存器列表

Name	Offset	Reset	Description
WWDG_CR	0x1A0	32'h7f	WWDG 控制寄存器
WWDG_CFR	0x1A4	32'h7f	WWDG 配置寄存器
WWDG_SR	0x1A8	32'h00	WWDG 状态寄存器

22.5.2. 寄存器详细说明

22.5.2.1. WWDG_CR

Width	Name	Reset	Property	Description
31:8	Reserved	-	-	-
7	WDGA	1'b0	RW	激活位 此位由软件置 1，但仅能由硬件在复位后清 0。当 WDGA=1 时，看门狗可以产生复位。 0: 禁止看门狗 1: 启用看门狗
6:0	T	7'h7f	RW	7 位计数器 这些位用来存储看门狗的计数器值。每(4096x2^WDGTB)个 SYSCLK 周期减 1。当计数器值从 40h 变为 3Fh 时(T6 变成 0)，产生看门狗复位。

22.5.2.2. WWDG_CFR

Width	Name	Reset	Property	Description
31:10	Reserved	-	-	-
9	EWI	1'b0	RW	提前唤醒中断 此位若置 1，则当计数器值达到 40h，即产生中断。 此中断只能由硬件在复位后清除。
8:7	WDGTB	2'b0	RW	时基 预分频器的时基可以设置如下: b00: CK 计时器时钟(SYSCLK 除以 4096)除以 1 b01: CK 计时器时钟(SYSCLK 除以 4096)除以 2 b10: CK 计时器时钟(SYSCLK 除以 4096)除以 4 b11: CK 计时器时钟(SYSCLK 除以 4096)除以 8
6:0	W	7'h7f	RW	7 位窗口值 这些位包含了用来与递减计数器进行比较用的窗口值。

22.5.2.3. WWDG_SR

Width	Name	Reset	Property	Description
31:1	Reserved	-	-	-
0	EWIF	1'b0	RC	提前唤醒中断标志 当计数器值达到 40h 时，此位由硬件置 1。它必须通过软件写 1 来清除。对此位写 0 无效。若中断未被使能，此位也会被置 1。

23. 直接存储访问控制模块（DMA）

23.1. 模块介绍

外设直接存储器访问电路 DMA 提供了 8 个单向通道用于专用外设执行“外设→存储器”和“存储器→外设”的数据传输。应用程序同时也支持和要求如 FLASH→SRAM 或 SRAM→SRAM 类型的“存储器→存储器”数据传输。每个 DMA 通道配置都是独立的。DMA 通道传输分为多个块处理，且每个块的大小等于块的长度乘以数据的宽度

23.2. 功能特点

- 8 个单向 DMA 通道
- 支持存储器→外设、外设→存储器和存储器→存储器类型的数据传输 8-bit、16-bit 和 32-bit 宽度数据传输
- 带有可配置通道优先级的软硬件需求的数据传输
- 线性、环形和非递增地址模式
- 4 个传输事件标志：传输完成、半传输、块结束和传输错误
- 自动重载功能

23.3. 功能说明

23.3.1. AHB 主机

DMA 是一个 AHB 主机，通过总线矩阵与其它 AHB 外设相连，如 FLASH 存储器、SRAM 存储器和 AHB-to-APB 总线桥。CPU 和 DMA 通过总线矩阵可以同时访问不同的 AHB 从机。

23.3.2. DMA 通道

有 8 个单向 DMA 通道用于外设和存储器之间的数据传输。每个 DMA 通道的配置和操作都是独立的。对于一个双向数据传输应用，需要两个 DMA 通道。每个 DMA 通道可支持多个外设设备，但使用同一套寄存器。因此，一个 DMA 通道在同一时间内只能服务一个外设。DMA 通道的相关寄存器被限制只能由 32-bit 操作访问，否则，将发生系统硬故障。

23.3.3. DMA 请求映射

来自众多外设的请求，如 ADC、SPI、IIC 和 USART 等等，在进入 DMA 之前先进行简单的逻辑与。这意味着一个 DMA 通道一次只能使能一个请求。请参考 图 24-1DMA 请求映射架构 。详细的外设 IP 请求映射列表如 表 24-1 DMA 通道配置 所示。外设 DMA 请求可通过相应外设寄存器中的 DMA 控制位独立激活或禁用。

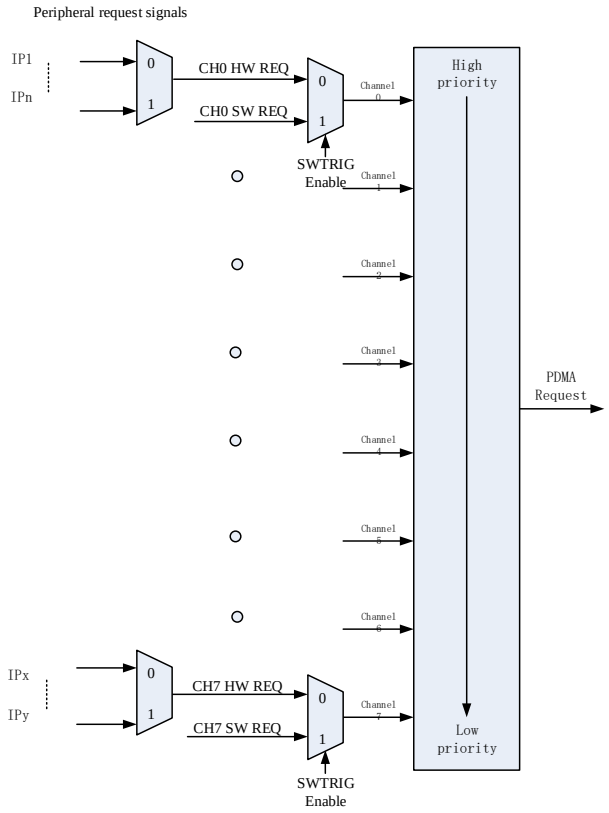


图 23-1 DMA 请求映射架构

表 23-1 DMA 通道配置

TRGSRC	REQ	描述
0	adkey_jdma_req	adc 注入通道 dma 请求
1	adkey_rdma_req	adc 规则通道 dma 请求
2	tim1_tri_dma_req	timer1 触发事件 dma 请求
3	tim1_ud_dma_req	timer1 更新事件 dma 请求
4	tim1_com_dma_req	timer1 的 COM 事件 dma 请求
5	tim1_cc1_dma_req	timer1 通道 1 事件 dma 请求
6	tim1_cc2_dma_req	timer1 通道 2 事件 dma 请求
7	tim1_cc3_dma_req	timer1 通道 3 事件 dma 请求
8	tim1_cc4_dma_req	timer1 通道 4 事件 dma 请求
9	tim2_tri_dma_req	timer2 触发事件 dma 请求
10	tim2_ud_dma_req	timer2 更新事件 dma 请求
11	tim2_cc1_dma_req	timer2 通道 1 事件 dma 请求
12	tim2_cc2_dma_req	timer2 通道 2 事件 dma 请求
13	tim2_cc3_dma_req	timer2 通道 3 事件 dma 请求
14	tim2_cc4_dma_req	timer2 通道 4 事件 dma 请求
15	tim3_tri_dma_req	timer3 触发事件 dma 请求
16	tim3_ud_dma_req	timer3 更新事件 dma 请求
17	tim3_cc1_dma_req	timer3 通道 1 事件 dma 请求
18	tim3_cc2_dma_req	timer3 通道 2 事件 dma 请求
19	tim3_cc3_dma_req	timer3 通道 3 事件 dma 请求
20	tim3_cc4_dma_req	timer3 通道 4 事件 dma 请求
21	tim4_tri_dma_req	timer4 触发事件 dma 请求
22	tim4_ud_dma_req	timer4 更新事件 dma 请求

23	tim4_com_dma_req	timer4 的 COM 事件 dma 请求
24	tim4_cc1_dma_req	timer4 通道 1 事件 dma 请求
25	tim4_cc2_dma_req	timer4 通道 2 事件 dma 请求
26	Reserved	-
27	Reserved	-
28	tim5_tri_dma_req	timer5 触发事件 dma 请求
29	tim5_ud_dma_req	timer5 更新事件 dma 请求
30	tim5_com_dma_req	timer5 的 COM 事件 dma 请求
31	tim5_cc1_dma_req	timer5 通道 1 事件 dma 请求
32	tim5_cc2_dma_req	timer5 通道 2 事件 dma 请求
33	Reserved	-
34	Reserved	-
35	tim6_tri_dma_req	timer6 触发事件 dma 请求
36	tim6_ud_dma_req	timer6 更新事件 dma 请求
37	tim6_com_dma_req	timer6 的 COM 事件 dma 请求
38	tim6_cc1_dma_req	timer6 通道 1 事件 dma 请求
39	tim6_cc2_dma_req	timer6 通道 2 事件 dma 请求
40	Reserved	-
41	Reserved	-
42	crc_dma_req	crc 的 dma 请求
43	eflash_dma_req	eflash 的 dma 请求
44	spi0_dma_tx_req	spi0 发送的 dma 请求
45	spi0_dma_rx_req	spi0 接收的 dma 请求
46	spi1_dma_tx_req	spi1 发送的 dma 请求
47	spi1_dma_rx_req	spi1 接收的 dma 请求
48	uart0_dma_tx_req	uart0 发送的 dma 请求
49	uart0_dma_rx_req	uart0 接收的 dma 请求
50	uart1_dma_tx_req	uart1 发送的 dma 请求
51	uart1_dma_rx_req	uart1 接收的 dma 请求
52~63	保留	保留

23.3.4. 通道传输

DMA 通道传输被分成多个块处理，在每个块处理结束后，DMA 仲裁发生。虽然这些通道传输都可以被激活，但一次只能通过总线进行一个块处理传输。通道传输顺序取决于每个 DMA 通道的优先级设置。总的传输大小由块处理数和块大小计算而得。块大小等于块长度和数据位宽度的乘积。对于一个有效的传输，建议块长度为 4 的倍数。

总传输数据大小可由下列等式算出：

DMA 通道总传输数据大小 = 块处理数 × (块长度 × 数据宽度)

23.3.5. 通道优先级

DMA 提供四个优先级，即非常高、高、中、低，可由应用软件设置。DMA 提供了两种方式来决定通道优先级。一种由应用软件配置选项进行设置，另一种由固定硬件编号决定。DMA 仲裁处理器将首先检查请求 DMA 提供数据传输服务的通道的软件配置优先级。如果多个通道有相同的优先级，则经过仲裁后，编号较小的通道比编号大的通道具有高优先级。

注意，当其它低优先级通道请求挂起时，最高优先级通道不会一直占用 DMA 服务。在一个块处理完成后，最高优先级通道将被跳过一个块处理的时间。接着执行由第二优先级通道请求的块处理。由于第二优先级通道在当前块处理完成时会被排除，所以在第二优先级通道的当前块处理完成时，DMA 仲裁处理器将重新检查除第二优先级通道外的其它请求通道。因此，较高优先级通道的块数据处理将被服务，且在

当前块处理完成时，此通道将从优先级仲裁中排除。DMA 将用上述的方法继续传输数据直到所有请求的通道数据传输完成。下图的例子显示了 DMA 通道仲裁和安排。

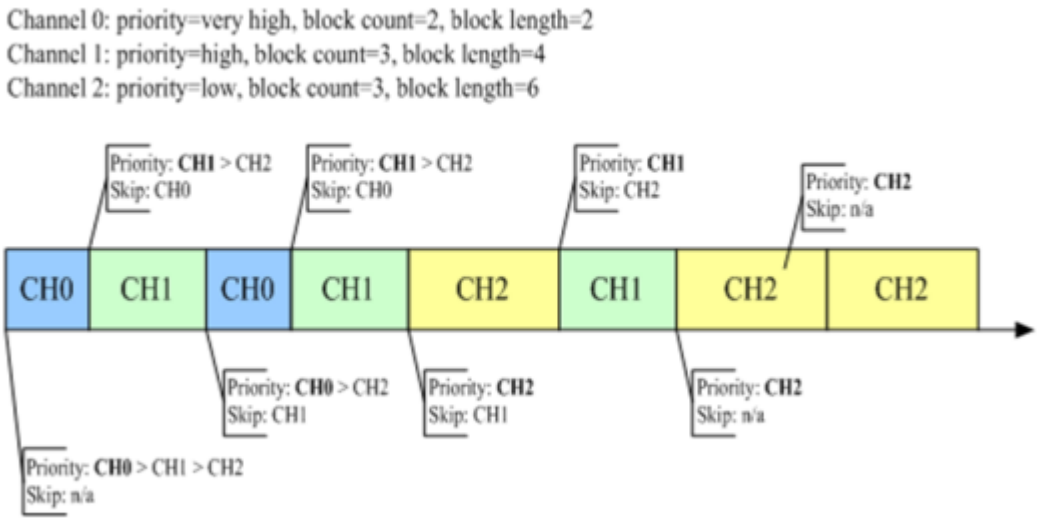


图 23-2 DMA 通道仲裁和传输安排

23.3.6. 传输请求

对于“外设→存储器”或“存储器→外设”传输，一个外设硬件请求将触发一个专用 DMA 通道的块处理。但当软件请求发生时，相关专用 DMA 通道的一个完整数据传输会被触发。应“存储器→存储器”数据复制应用的软件需求，建议 DMA 通道被配置为较低的优先级和较小的块长度。

23.3.7. 地址模式

DMA 提供了三种地址模式，即线性地址、环形地址和固定地址模式。这些不同的地址模式用来支持不同种类的源地址和目标地址配置。下表显示了详细的地址模式组合。

表 23-2 DMA 地址模式

源地址模式	目的地址模式
线性递增/递减地址	线性递增/递减地址
线性递增/递减地址	环形递增/递减地址
线性递增/递减地址	固定地址
环形递增/递减地址	线性递增/递减地址
环形递增/递减地址	环形递增/递减地址
固定地址	线性递增/递减地址
固定地址	固定地址

线性地址模式

在数据被传输后，当前地址将以 1、2 或 4 为间隔递增或递减，取决于数据位的宽度设置。

环形地址模式

在数据被传输后，当前地址将以 1、2 或 4 为间隔递增或递减，取决于数据位的宽度设置。当块处理完成，所设置的起始地址将加载到当前地址中。

固定地址

数据传输后，当前地址仍然不变。

23.3.8. 自动重载

当 DMA 通道 n 控制寄存器 DMA_CHnCR 中的自动重载控制位 AUTORLn 被置位时，在 当前 DMA 通道数据传输完全完成时，通道 n 当前地址和通道 n 当前传输大小将分别自动载入对应的初始值。通道 n 仍然有效且下一个相关的 DMA 请求在不需要软件重新配置的情况下可以被服务。

23.3.9. 传输中断

每个 DMA 通道都有 5 种传输事件，当传输事件发生时可以产生相应中断。这些传输事件分别为块处理结束(BE)、半传输(HT)、传输完成(TC)、传输错误(TE)和总传输事件(GE)。 在 DMA 中断使能寄存器 DMAIER 中设置相关控制位可使能相关中断事件。如果 BE、HT、TC 或 TE 四个中断事件中有任意一个发生时，将产生总中断事件 GE。清零 BE、HT、TC 或 TE 事件标志位也会清零 GE 标志位。清零 GE 标志位将自动清零所有其它事件标志位。当 DMA 访问一个系统预留地址空间或 DMA 接收一个请求但相应传输大小设置为 0 时，将产生 TE 中断事件。

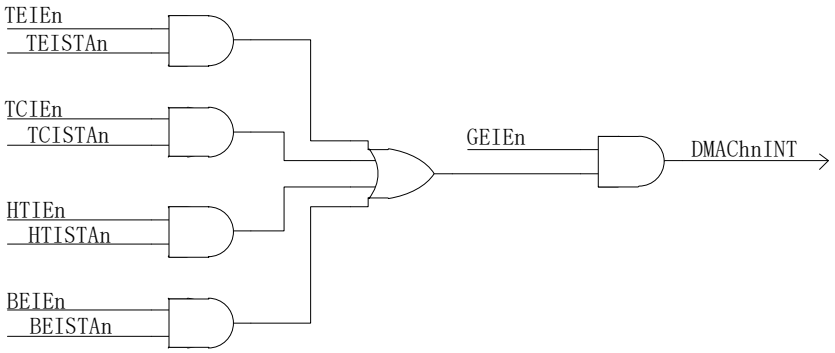


图 23-3

23.4. 模块架构与接口时序

23.4.1. 模块架构图

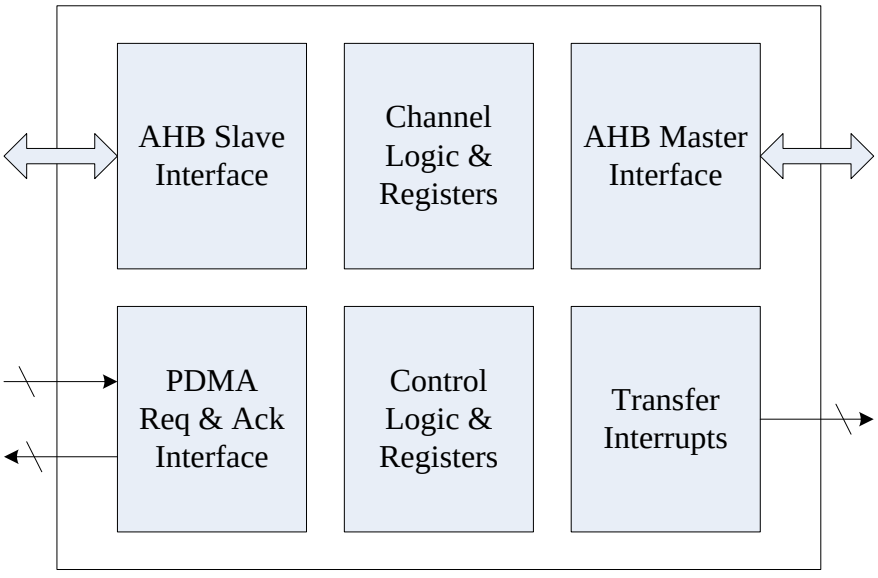


图 23-4 DMA 方框图

23.4.2. 接口时序

寄存器读写接口是 ahb 接口。

dma 访问外设和存储器的接口是 ahb 接口。

23.5. 寄存器描述

23.5.1. 寄存器列表

Base address：0x4003_0000

Name	Offset	Reset	Description

DMA_CHxCR	0x0000+x*0x18(x=0~7)	32'h0	DMA 通道 x 控制寄存器
DMA_CHxSADR	0x0004+x*0x18	32'h0	DMA 通道 x 源地址寄存器
DMA_CHxDADR	0x0008+x*0x18	32'h0	DMA 通道 x 目标地址寄存器
Reserved	0x000C+x*0x18	-	-
DMA_CHxTSR	0x0010+x*0x18	32'h0	DMA 通道 x 传输大小寄存器
DMA_CHxCTSR	0x0014+x*0x18	32'h0	DMA 通道 x 当前传输大小寄存器
DMA_ISR0	0x120	32'h0	DMA 中断状态寄存器 0
DMA_ISR1	0x124	32'h0	DMA 中断状态寄存器 1
DMA_ICLR0	0x128	32'h0	DMA 中断状态清除寄存器 0
DMA_ICLR1	0x12C	32'h0	DMA 中断状态清除寄存器 1
DMA_IER0	0x130	32'h0	DMA 中断使能寄存器 0
DMA_IER1	0x134	32'h0	DMA 中断使能寄存器 1

23.5.2. 寄存器详细说明

23.5.2.1. DMA 通道 n 控制寄存器（DMA_CHnCR），n=0 ~ 7

Width	Name	Reset	Property	Description
31:22	Reserved	-	-	-
21:16	TRGSRC	6'b0	WR	硬件发起源外设 详见 表 24-1 DMA 通道配置
15:12	Reserved	-	-	-
11	AUTORL	1'b0	WR	通道 n 自动重载使能控制位 0: 除能自动重载功能 1: 使能自动重载功能 当传输完成，通道 n 当前地址和当前传输大小将重新载入对应的起始值。 该位设为 1 将使能自动重载功能，DMA 通道 n 使能仍有效。 如果该位设为 0，在传输完成后，通道 n 当前地址和当前传输大小的值将保持不变且 DMA 通道 n 将除能。
10	FIXAEN	1'b0	WR	通道 n 固定地址使能控制位 0: 在环形地址模式下除能固定地址功能 1: 在环形地址模式下使能固定地址功能 注意，该位仅当源地址或目标地址处于环形地址模式下时才有效。例如，源地址模式设置为线性地址模式，目标地址模式设置为环形模式。如果该位置位使能固定地址功能，则源地址模式仍会处于线性地址模式，但是目标地址将会处于固定地址模式而不是环形地址模式。
9:8	CHPRI	2'b0	WR	通道 n 优先级 00: 低 01: 中 10: 高 11: 非常高 CHnPRI 字段通过应用程序来配置通道的优先

				级。如果含有相同的软件配置优先等级的通道超过 1 个，则更小通道编号的通道将在仲裁后优先传输一个数据块。
7	SRCAMOD	1'b0	WR	通道 n 源地址模式选择位 0: 线性地址模式 1: 环形地址模式 在线性地址模式下，在一个完整的数据传输过程中当前源地址的值根据 SRCAINCn 位的值可以递增或递减。在环形地址模式下，当前源地址的值在数据传输过程中也是根据 SRCAINCn 位的值来决定是递增或递减，当一块的数据传输完成时 DMACHnSADR 寄存器的低 17-bit 的值将会被加载为当前源地址的值。
6	Reserved	-	-	-
5	DSTAMOD	1'b0	WR	通道 n 目标地址模式选择位 0: 线性地址模式 1: 环形地址模式 在线性地址模式下，当前目标地址的值在一个完整的数据传输过程中根据 DSTAINCn 位的值可以递增或递减。在环形地址模式下，当前地址在一个数据块传输过程中也是根据 DSTAINCn 位的值来决定是递增或递减，当一块的数据传输完成时所设置的起始地址将被加载到当前目标地址中。
4	Reserved	-	-	-
3:2	DWIDTH	2'b0	WR	数据位宽度选择位 00: 8-bit 01: 16-bit 10: 32-bit 11: 保留 该位用来选择 DMA 通道 n 的数据位宽度。
1	SWTRIG	1'b0	WR	软件触发控制位 0: 无操作 1: 软件触发传输请求 置位该位将会在 DMA 通道 n 产生一个“存储器→存储器”软件传输请求。当传输完成该位会自动清零。
0	CHEN	1'b0	WR	通道 n 使能控制位 0: 除能 DMA 通道 n 1: 使能 DMA 通道 n 置位该位将会在 DMA 通道 n 上使能一个软件或硬件传输请求。当传输完成且自动重载功能除能时，该位将自动由硬件清零。然而，如果 AUTORLn 位设为 1，使能自动重载功能，在传输完成后该位将保持高以使能 DMA 通道 n 功能用于下一个传输请求而不会由硬件自动清零。

23.5.2.2. DMA 通道 n 源地址寄存器（DMA_CHnSADR）

bits	Field	Reset	Property	Description
31:0	SADR	32'b0	RW	通道 n 源地址 该寄存器用于定义 DMA 通道 n 的 32-bit 源地址。

23.5.2.3. DMA 通道 n 目标地址寄存器（DMA_CHnDADR）

bits	Field	Reset	Property	Description
31:0	DADR	32'b0	RW	通道 n 目的地址 该寄存器用于定义 DMA 通道 n 的 32-bit 目的地址。

23.5.2.4. DMA 通道 n 传输大小寄存器（DMA_CHnTSR）

bits	Field	Reset	Property	Description
31:24	Reserved	-	-	-
23:12	BLKCNT	12'b0	RW	通道 n 块处理数量 BLKCNTn 代表通道 n 完整传输的块处理的数量。一个完整传输的容量是由 BLKCNTn 和 BLKLENn 的值计算得到。BLKCNTn 最大值为 0xfff。
11:5	Reserved	-	-	-
4:0	BLKLEN	5'b0	RW	通道 n 块长度 BLKLENn 代表数据块的长度。数据宽度由 DMACHnCR 寄存器中的 DWIDTHn 字段来定义。BLKLENn 最大值为 31。

23.5.2.5. DMA 通道 n 当前传输大小寄存器（DMA_CHnCTSR）

bits	Field	Reset	Property	Description
31:24	Reserved	-	-	-
23:12	CBLKCNT	12'b0	RO	通道 n 当前块数量 CBLKCNTn 是一个 12-bit 的只读值，它定义了待传输的数据块的数量。一个数据块传输完成后，CBLKCNTn 值将会减 1。给 DMACHnTSR 寄存器中的 BLKCNTn 写一个新值将会更新 CBLKCNTn 的值。
11:5	Reserved	-	-	-
4:0	CBLKLEN	5'b0	RO	通道 n 当前块长度 CBLKLENn 代表当前数据块的长度。也就是还没传输完成的数据量。一个数据传输完成后，CBLKLENn 值将会减 1。给 DMACHnTSR 寄存器中的 BLKLENn 写一个新值将会更新 CBLKLENn 的值。

23.5.2.6. DMA 中断状态寄存器 0（DMA_ISR0）

bits	Field	Reset	Property	Description
31:30	Reserved	-	-	-
29, 24, 19, 14, 9, 4	TEISTAn	1'b0	RO	通道 n 传输错误中断状态位(n = 0 ~ 5) 0：传输错误未发生 1：传输错误发生 该位由硬件置位，写 1 到 DMAISCR0 寄存器中的相关中断状态清零位将会清零该位。 当 DMA 访问系统预留地址空间或 DMA 接收到一个请求但是对应的传输容量为 0 时，将会发生传输错误。
28, 23, 18, 13, 8, 3	TCISTAn	1'b0	RO	通道 n 传输完成中断状态位(n = 0 ~ 5) 0：传输完成未发生 1：传输完成发生 该位由硬件置位，写 1 到 DMAISCR0 寄存器的相关中断状态清零位将会清零该位。 当 DMA 完成数据传输任务时将发生传输完成事件。
27, 22, 17, 12, 7, 2	HTISTAn	1'b0	RO	通道 n 半传输中断状态位(n = 0 ~ 5) 0：半传输事件未发生 1：半传输事件发生 该位由硬件置位，写 1 到 DMAISCR0 寄存器的相关中断状态清零位将会清零该位。 当 DMA 完成一半的数据传输任务时将发生半传输完成事件。
26, 21, 16, 11, 6, 1	BEISTAn	1'b0	RO	通道 n 块处理结束中断状态位(n = 0 ~ 5) 0：块处理结束事件未发生 1：块处理结束事件发生 该位由硬件置位，写 1 到 DMAISCR0 寄存器的相关中断状态清零位将会清零该位。 当 DMA 完成数据块处理任务时将发生块处理结束事件。
25, 20, 15, 10, 5, 0	Reserved	-	-	-

23.5.2.7. DMA 中断状态寄存器 1 （DMA_ISR1）

bits	Field	Reset	Property	Description
31:10	Reserved	-	-	-
9, 4	TEISTAn	1'b0	RO	通道 n 传输错误中断状态位(n = 6 ~ 7) 0：传输错误未发生 1：传输错误发生 该位由硬件置位，写 1 到 DMAISCR1 寄存器中的相关中断状态清零位将会清零该位。 当 DMA 访问系统预留地址空间或 DMA 接收到一个请求但是对应的传输容量为 0 时，将会发生传输错误。

8, 3	TCISTAn	1'b0	RO	通道 n 传输完成中断状态位(n = 6 ~ 7) 0：传输完成未发生 1：传输完成发生 该位由硬件置位，写 1 到 DMAISCR1 寄存器的相关中断状态清零位将会清零该位。 当 DMA 完成数据传输任务时将发生传输完成事件。
7, 2	HTISTAn	1'b0	RO	通道 n 半传输中断状态位(n = 6 ~ 7) 0：半传输事件未发生 1：半传输事件发生 该位由硬件置位，写 1 到 DMAISCR1 寄存器的相关中断状态清零位将会清零该位。 当 DMA 完成一半的数据传输任务时将发生半传输完成事件。
6, 1	BEISTAn	1'b0	RO	通道 n 块处理结束中断状态位(n = 6 ~ 7) 0：块处理结束事件未发生 1：块处理结束事件发生 该位由硬件置位，写 1 到 DMAISCR1 寄存器的相关中断状态清零位将会清零该位。 当 DMA 完成数据块处理任务时将发生块处理结束事件。
5, 0	Reserved	-	-	-

23.5.2.8. DMA 中断状态清除寄存器 0 （DMA_ISCR0）

bits	Field	Reset	Property	Description
31:30	Reserved	-	-	-
29, 24, 19, 14, 9, 4	TEICLRn	1'b0	WC	通道 n 传输错误中断状态清除(n = 0 ~ 5) 0：无操作 1：将 DMAISR0 寄存器中对应的 TEISTAn 位清零 写 1 到 TEICLRn 位会将 DMAISR0 寄存器中的 TEISTAn 位清零。写 1 之后该位会自动清零。
28, 23, 18, 13, 8, 3	TCICLRn	1'b0	WC	通道 n 传输完成中断状态清除(n = 0 ~ 5) 0：无操作 1：将 DMAISR0 寄存器中对应的 TCISTAn 位清零 写 1 到 TCICLRn 位会将 DMAISR0 寄存器中的 TCISTAn 位清零。写 1 之后该位会自动清零。
27, 22, 17, 12, 7, 2	HTRICLRn	1'b0	WC	通道 n 半传输中断状态清除(n = 0 ~ 5) 0：无操作 1：将 DMAISR0 寄存器中对应的 HTISTAn 位清零 写 1 到 HTRICLRn 位会将 DMAISR0 寄存器中的 HTISTAn 位清零。写 1 之后该位会自动清零。
26, 21, 16, 11, 6, 1	BEICLRn	1'b0	WC	通道 n 块处理结束中断状态清除(n = 0 ~ 5) 0：无操作 1：将 DMAISR0 寄存器中对应的 BEISTAn 位清

				零 写 1 到 BEICLRn 位会将 DMAISR0 寄存器中的 BEISTAn 位清零。写 1 之后该位会自动清零。
25, 20, 15, 10, 5, 0	Reserved	-	-	-

23.5.2.9. DMA 中断状态清除寄存器 1（DMA_ISCR1）

bits	Field	Reset	Property	Description
31:10	Reserved	-	-	-
9, 4	TEICLRn	1'b0	WC	通道 n 传输错误中断状态清除(n = 6 ~ 7) 0：无操作 1：将 DMAISR1 寄存器中对应的 TEISTAn 位清零 写 1 到 TEICLRn 位会将 DMAISR1 寄存器中的 TEISTAn 位清零。写 1 之后该位会自动清零。
8, 3	TCICLRn	1'b0	WC	通道 n 传输完成中断状态清除(n = 6 ~ 7) 0：无操作 1：将 DMAISR1 寄存器中对应的 TCISTAn 位清零 写 1 到 TCICLRn 位会将 DMAISR1 寄存器中的 TCISTAn 位清零。写 1 之后该位会自动清零。
7, 2	HTRICLRn	1'b0	WC	通道 n 半传输中断状态清除(n = 6 ~ 7) 0：无操作 1：将 DMAISR1 寄存器中对应的 HTISTAn 位清零 写 1 到 HTRICLRn 位会将 DMAISR1 寄存器中的 HTISTAn 位清零。写 1 之后该位会自动清零。
6, 1	BEICLRn	1'b0	WC	通道 n 块处理结束中断状态清除(n = 6 ~ 7) 0：无操作 1：将 DMAISR1 寄存器中对应的 BEISTAn 位清零 写 1 到 BEICLRn 位会将 DMAISR1 寄存器中的 BEISTAn 位清零。写 1 之后该位会自动清零。
5, 0	Reserved	-	-	-

23.5.2.10. DMA 中断使能寄存器 0（DMA_IER0）

bits	Field	Reset	Property	Description
31:30	Reserved	-	-	-
29, 24, 19, 14, 9, 4	TEIEn	1'b0	RW	通道 n 传输错误中断使能控制位(n = 0 ~ 5) 0：传输错误中断除能 1：传输错误中断使能 该位通过软件置位和清零。
28, 23, 18, 13, 8, 3	TCIEn	1'b0	RW	通道 n 传输完成中断使能控制位(n = 0 ~ 5) 0：传输完成中断除能 1：传输完成中断使能 该位通过软件置位和清零。

27, 22, 17, 12, 7, 2	HTIEn	1'b0	RW	通道 n 半传输中断使能控制位(n = 0 ~ 5) 0：半传输中断除能 1：半传输中断使能 该位通过软件置位和清零。
26, 21, 16, 11, 6, 1	BEIEn	1'b0	RW	通道 n 块处理结束中断使能控制位(n = 0 ~ 5) 0：块处理结束中断除能 1：块处理结束中断使能 该位通过软件置位和清零。
25, 20, 15, 10, 5, 0	Reserved	-	-	-

23.5.2.11. DMA 中断使能寄存器 1（DMA_IER1）

bits	Field	Reset	Property	Description
31:10	Reserved	-	-	-
9, 4	TEIEn	1'b0	RW	通道 n 传输错误中断使能控制位(n = 6 ~ 7) 0：传输错误中断除能 1：传输错误中断使能 该位通过软件置位和清零。
8, 3	TCIEn	1'b0	RW	通道 n 传输完成中断使能控制位(n = 6 ~ 7) 0：传输完成中断除能 1：传输完成中断使能 该位通过软件置位和清零。
7, 2	HTIEn	1'b0	RW	通道 n 半传输中断使能控制位(n = 6 ~ 7) 0：半传输中断除能 1：半传输中断使能 该位通过软件置位和清零。
6, 1	BEIEn	1'b0	RW	通道 n 块处理结束中断使能控制位(n = 6 ~ 7) 0：块处理结束中断除能 1：块处理结束中断使能 该位通过软件置位和清零。
5, 0	Reserved	-	-	-

24. 系统寄存器（SYSTEM_REG)

24.1. 模块介绍

该模块用于对系统寄存器配置，部分系统寄存器会在上电过程中根据 FLASH 的系统配置区数据是否通过 CRC 校验，自动把配置参数装载到系统寄存器中。系统寄存器控制整个芯片系统的时钟使能、时钟选择、复位控制、唤醒控制和低功耗控制等。

24.2. 寄存器描述

24.2.1. 寄存器列表

Base address：0x4002_0000

Name	Offset	Reset	Description
SYSKEY	0x0000	32'h1	系统控制寄存器秘钥
SYSCON0	0x0004	32'h69f9040	系统控制寄存器 0
SYSCON1	0x0008	32'h300	系统控制寄存器 1
SYSCON2	0x000C	32'h0	系统控制寄存器 2
SYSCON3	0x0010	32'h0	系统控制寄存器 3
CLKCON0	0x0024	32'h0	时钟控制寄存器 0
CLKCON1	0x0028	32'hf8000000	时钟控制寄存器 1
CLKCON2	0x002C	32'h7878ff4	时钟控制寄存器 2
CLKCON3	0x0030	32'h80020240	时钟控制寄存器 3
SYSERR	0x0048	32'h0	系统错误记录寄存器
WKUPCON	0x004C	32'h0	系统唤醒控制寄存器
LPCON	0x0050	32'h20	低功耗控制寄存器
CHIPID_DCN	0x005C	32'h8	CHIPID 和 DCN 寄存器
MODE	0x0060	32'h0	系统启动模式寄存器
PMUCON0	0x0064	32'h10177800	PMU 控制寄存器
RPCON	0x0068	32'h0	复位控制寄存器
PMU_BK	0x006C	32'h10	PMU 备份寄存器

24.2.2. 寄存器详细说明

24.2.2.1. 控制寄存器（SYSKEY)

Width	Name	Reset	Property	Description
31:0	SYS_KEY	32'h1	RW	系统寄存器写使能 写: 0x3fac87e4: 使能系统寄存器写权限 others：关闭系统寄存器写权限 读: 0: 系统寄存器写权限关闭 1: 系统寄存器写权限开启

24.2.2.2. 控制寄存器（SYSCON0)

Width	Name	Reset	Property	Description
31	FAST_RST_EN	1'b0	RW	快速释放复位 0: 不使能

				1: 使能
30	SLEEP_GOON_EN	1'b0	RW	系统唤醒时复位 0: 复位 1: 不复位
29:27	SLEEP_DLY_CNT	3'b0	RW	IO 唤醒时延时 SLEEP_DLY_CNT 个系统时钟周期，再唤醒系统。 （如果睡眠时系统跑 256K，该配置应>=2）
26	DBS_SOFT_RST_	1'b1	RW	IO 消抖电路软件复位控制 0: 复位 1: 完成复位
25	CRC_SOFT_RST_	1'b1	RW	CRC 计算单元软件复位控制 0: 复位 1: 完成复位
24	Reserved	-	-	-
23	HWDIV_SOFT_RST_	1'b1	RW	HWDIV 软件复位控制 0: 复位 1: 完成复位
22:21	Reserved	-	-	-
20	GPIO_SOFT_RST_	1'b1	RW	GPIO 相关寄存器软件复位控制 0: 复位 1: 完成复位
19	ADC_SOFT_RST_	1'b1	RW	ADC 控制模块软件复位控制 0: 复位 1: 完成复位
18	UART1_SOFT_RST_	1'b1	RW	UART1 软件复位控制 0: 复位 1: 完成复位
17	UART0_SOFT_RST_	1'b1	RW	UART0 软件复位控制 0: 复位 1: 完成复位
16	SPI_IIC1_SOFT_RST_	1'b1	RW	SPI_IIC1 软件复位控制 0: 复位 1: 完成复位
15	SPI_IIC0_SOFT_RST_	1'b1	RW	SPI_IIC0 软件复位控制 0: 复位 1: 完成复位
14:13	Reserved	-	-	-
12	COMP_SOFT_RST_	1'b1	RW	COMP_OPAM 软件复位 0: 复位 1: 完成复位
11:7	Reserved	-	-	-
6	TIMER_SOFT_RST_	1'b1	RW	TIMERx 软件复位控制 0: 复位 1: 完成复位
5:0	Reserved	-	-	-

24.2.2.3. 控制寄存器（SYSCON1）

Width	Name	Reset	Property	Description
31	SYS_TIM_OC_FEN	1'b0	RW	TIMER 输出通道输出使能 0: timer 通道输出使能由内部逻辑决定 1: timer 通道输出使能固定有效
30:18	Reserved	-	-	-
17	Reserved	-	-	-
16	Reserved	-	-	-
15:14	Reserved	-	-	-
13	WDT_LP_GATE_EN	1'b0	RW	看门狗进入 sleep/stop 模式时自动关闭时钟 0: 不使能 1: 使能
12	DEBUG_EN	1'b0	RW	如果配置成 0，在 DEBUG（连接 SWD）模式下停止 sleep 和 stopclk 功能，在超级模式中还会停止 wdt_reset 功能 0: 停止 1: 允许
11	INT_REMAP_EN	1'b0	RW	把中断入口从 FLASH 映射到 SRAM 0-192Bytes 0: 不映射 1: 映射
10	NMI_INV_SEL	1'b0	RW	NMI 引脚极性反向选择 0: 高电平引脚触发 NMI 1: 低电平引脚触发 NMI
9	CP_MODE_SYSCLK_EN	1'b1	RW	在 CP 模式中系统时钟选择 0: 由 CLKCON0[1:0]选择 1: 系统时钟选择 IO 时钟
8	SWD_EN	1'b1	RW	SWD 使能控制 0: 不使能 1: 使能
7	LVDVCC_WKUP_EN	1'b0	RW	LVDVCC 中断模式下，唤醒系统 0: 不使能 1: 使能
6	Reserved	-	-	-
5	SYS_ERR_RESP_EN	1'b0	RW	系统访问超出 SRAM 地址范围时，返回出错信号给 CPU 0: 不使能 1: 使能
4	SYS_ERR_INT_EN	1'b0	RW	系统总线访问越界地址或时钟出错时触发 NMI 中断 0: 不使能 1: 使能
3	Reserved	-	-	-
2	RXEV_EN	1'b0	RW	CPU 接收事件使能 0: 不使能 1: 使能

1	NMI_INT_ENABLE	1'b0	RW	PB5 触发 NMI 中断使能 0: 不使能 1: 使能
0	LOCKUP_ENABLE	1'b0	RW	两次访问越界地址时触发系统复位 0: 不使能 1: 使能（需要 SYS_ERR_RESP_EN 为 1）

24.2.2.4. 控制寄存器（SYSCON2）

Width	Name	Reset	Property	Description
31:16	PB_DEB_EN	16'h0	RW	PB 消抖控制 0: 不使能 1: 使能
15:0	PA_DEB_EN	16'h0	RW	PA 消抖控制 0: 不使能 1: 使能

24.2.2.5. 控制寄存器（SYSCON3）

Width	Name	Reset	Property	Description
31:12	Reserved	-	-	-
11:0	PC_DEB_EN	12'h0	RW	PC 消抖控制 0: 不使能 1: 使能

24.2.2.6. 控制寄存器（CLKCON0）

Width	Name	Reset	Property	Description
31:18	Reserved	-	-	-
17:16	LVD_DEB_CLK_SEL	2'h0	RW	LVD 消抖时钟选择 b00: SYSCLK b01: HIRC_CLK b10: HIRC_DIV2_CLK b11: LIRC_256K
15:8	Reserved	-	-	-
7:6	GPIO_DEB_CLK_SEL	2'h0	RW	GPIO 消抖时钟选择 b00: HIRC_CLK b01: LIRC_DIV_32K b10: SYSCLK b11: LIRC 256K
5:4	Reserved	-	-	-
3:2	Reserved	-	-	-
1:0	SYSCLK_SEL	2'h0	RW	系统时钟选择 b00: LIRC_256K b01: XOSC(IO 时钟) b10: HIRC_CLK b11: HIRC_CLK

24.2.2.7. 控制寄存器（CLKCON1）

Width	Name	Reset	Property	Description
31:27	Reserved	-	-	-
26:6	Reserved	-	-	-
5:0	SYSCLK_DIV	6'h0	RW	系统时钟分频 b000000: 1 分频 b000001: 2 分频 ... b111110: 63 分频 b111111: 时钟停止

24.2.2.8. 控制寄存器（CLKCON2）

Width	Name	Reset	Property	Description
31:30	Reserved	-	-	-
29	Reserved	-	-	-
28	Reserved	-	-	-
27	Reserved	-	-	-
26	COMP_CLK_EN	1'b1	RW	COMP_OPAM 时钟使能 0: 不使能 1: 使能
25	CRC_CLK_EN	1'b1	RW	CRC 时钟使能 0: 不使能 1: 使能
24	FLASH_MEM_CLK_EN	1'b1	RW	FLASH 擦除/烧写时钟使能 0: 不使能 1: 使能
23	HWDIV_CLK_EN	1'b1	RW	HWDIV 时钟使能 0: 不使能 1: 使能
22:19	Reserved	-	-	-
18	UART1_CLK_EN	1'b1	RW	UART1 时钟使能 0: 不使能 1: 使能
17	UART0_CLK_EN	1'b1	RW	UART0 时钟使能 0: 不使能 1: 使能
16	SPI_IIC1_CLK_EN	1'b1	RW	SPI_IIC1 时钟使能 0: 不使能 1: 使能
15	SPI_IIC0_CLK_EN	1'b1	RW	SPI_IIC0 时钟使能 0: 不使能 1: 使能
14:12	Reserved	-	-	-
11	WWDG_CLK_EN	1'b1	RW	WWDG 时钟使能 0: 不使能 1: 使能

10:4	TIMERx_CLK_EN	7'h7f	RW	TIMERx 时钟使能(x=1...7) 0: 不使能 1: 使能
3	Reserved	-	-	-
2	SRAM0_CLK_EN	1'b1	RW	SRAM0 时钟使能 0: 不使能 1: 使能
1:0	Reserved	-	-	-

24.2.2.9. 控制寄存器（CLKCON3）

Width	Name	Reset	Property	Description
31	HIRC_EN_FLAG	1'b1	RO	HIRC_CLK 使能标志位 0: 未使能 1: 已使能
30	Reserved	-	-	-
29:18	HIRC_FSC	12'h0	RW	HIRC_CLK 频率控制(NVR TRIM) h0: 最低频 ..... hfff: 最高频
17	HIRC_EN	1'b1	RW	HIRC_CLK 使能控制 0: 不使能 1: 使能
16:12	Reserved	-	-	
11:10	HIRC_FSEL	2'b0	RW	HIRC 频率选择 b00: 32MHz b01: 48MHz b10: 72MHz b11: reserved
9:7	HIRC_FTS	3'b100	RW	HIRC 温度补偿系数
6:2	HIRC_FFS	5'b10000	RW	HIRC fine tune control (NVR TRIM) b00000: lowest freq b11111: highest freq
1	HIRC_ATSEL	1'b0	RW	HIRC 模拟测试选择 0: LDO 电压 1: 内部测试电压
0	HIRC_ATSEN	1'b0	RW	HIRC 模拟测试使能 0: 不使能 1: 使能

24.2.2.10. 控制寄存器（SYSERR）

Width	Name	Reset	Property	Description
31:2	Reserved	-	-	-
1	CLK_ERR	1'b0	RW	时钟使用错误标志 0: 正确 1: 错误 软件写 0 清除

0	SYS_ERR0	1'b0	RW	系统总线访问越界地址错误标志 0: 正确 1: 错误 软件写 0 清除

24.2.2.11. 控制寄存器（WKUPCON）

Width	Name	Reset	Property	Description
31:28	Reserved	-	-	-
27:24	CLR_WKUP_PEND	4'h0	WO	软件写 1 清除 WKUP_PEND
23:20	Reserved	-	-	-
19:16	WKUP_PEND	4'h0	RO	IO 边沿唤醒标志 0: 未检测到边沿 1: 检测到边沿
15:12	Reserved	-	-	-
11:8	WKUP_EDGE	4'h0	RW	唤醒边沿选择 0: 上升沿 1: 下降沿
7:4	Reserved	-	-	-
3:0	WKUP_EN	4'h0	RW	IO 边沿检测唤醒 0: 不使能 1: 使能 WKUP_EN[1]: 复用为 TIMR7 唤醒使能 WKUP_EN[2]: 复用为 COMP 唤醒使能 WKUP_EN[3]: 复用为 LVDVCC 唤醒使能

24.2.2.12. 控制寄存器（LPCON）

Width	Name	Reset	Property	Description
31:9	Reserved	-	-	-
8	PMU_LP_SW_EN	1'b0	RW	设置为 1 时，PMU_V2IEN 关闭，且 LPLDOS 切换到 PMUBK 寄存器的值 0: 不使能 1: 使能
7	PMU_LP_HW_EN	1'b0	RW	设置为 1 时，当 PMU 进入 sleep 模式，硬件自动关闭 PMU_V2IEN，且 LPLDOS 切换到 PMUBK 寄存器的值 0: 不使能 1: 使能
6	Reserved	-	-	-
5	RC256K_SOFT_EN	1'b1	RW	LIRC_256K 使能 0: 不使能 1: 使能
4	RC256K_AUTO_DIS	1'b0	RW	进入 sleep 模式时自动关闭 LIRC_256K 0: 不使能 1: 使能
3	HIRC_AUTO_DIS	1'b0	RW	进入 sleep/stop 模式时自动关闭 HIRC 0: 不使能

				1: 使能
2	SRAM0_AUTO_DIS	1'b0	RW	进入 sleep/stop 模式自动关闭 SRAM CE 0: 不使能 1: 使能
1	STOP_CLK_MODE	1'b0	RW	STOP 模式 0: 不使能 1: 使能
0	SLEEP	1'b0	RW	SLEEP 模式 0: 不使能 1: 使能

24.2.2.13. 控制寄存器（CHIPID_DCN）

Width	Name	Reset	Property	Description
31:24	Reserved	-	-	-
23:16	CHIP_DCN	8'h0	RO	记录版本修改
15:0	CHIP_ID	16'h8	RO	芯片 ID

24.2.2.14. 控制寄存器（PMUCON0）

Width	Name	Reset	Property	Description
31:29	Reserved	-	-	-
28:25	PMU_HPBGS	4'h8	RW	普通模式 BG VREF 电压控制(EFALSH TRIM) b0000: 最小值 ... b1000: 1.218 ... b1111: 最大值 Step=6mv
24	Reserved	-	-	-
23	PMU_ITSEN	1'b0	RW	内部温度传感器使能 0: 不使能 1: 使能
22	HPXCPFB	1'b0	RW	内部 LDO 配置选项 0: 带反馈 1: 不带反馈
21	Reserved	-	-	-
20	PMU_HPV2I_EN	1'b1	RW	模拟偏置电流使能 0: 不使能 1: 使能
19	Reserved	-	-	-
18	PMU_HPPDLI	1'b1	RW	VDD LDO 弱下拉电阻开关 0: 不使能 1: 使能
17	PMU_HPPDI	1'b1	RW	VDD LDO 下拉电阻开关 0: 不使能 1: 使能
16:14	PMU_LPIS	3'b101	RW	低功耗参考电流控制(EFALSH TRIM)

				b000: 最小电流 b111: 最大电流
13	PMU_HPLDO	1'b1	RW	普通模式 LDO 使能 0: 不使能 1: 使能 进入 SLEEP 模式时硬件自动把该位清 0
12	PMU_HPXCP	1'b1	RW	内部 LDO 使能 0: 不使能 1: 使能
11:8	PMU_LPBGS	4'h8	RW	ULPBG VREF 电压控制(EFALSH TRIM) b0000: 最小值 ... b1000: 0.835 ... b1111: 最大值 Step=10mv
7	PMU_IDEEP	1'b0	RW	深度睡眠电流开关 0: 使能 1: 不使能
6:4	PMU_LPLDOS	3'b0	RW	低功耗模式 LDO 电压选择(FLASH TRIM): b000=1.50V b001=1.45V b010=1.55V b011=1.60V b100=1.65V b101=1.75V b110=1.85V b111=1.25V
3	PMU_LPSSI	1'b0	RW	PMU 低功耗加速 0: 不使能 1: 使能
2:0	PMU_HPLDOS	3'b0	RW	普通模式 LDO 电压选择(FLASH TRIM): b000=1.50V b001=1.45V b010=1.55V b011=1.60V b100=1.65V b101=1.75V b110=1.85V b111=1.25V

24.2.2.15. 控制寄存器（RPCON）

Width	Name	Reset	Property	Description
31:19	Reserved	-	-	-
18	LOCKUP_RESET_CLR	1'b0	WO	软件写 1 清除 LOCK_RESET_PEND
17	SOFT_RESET_CLR	1'b0	WO	软件写 1 清除 SOFT_RESET_PEND
16	SLEEP_STA_CLR	1'b0	WO	软件写 1 清除 SLEEP_PEND

15:3	Reserved	-	-	-
2	LOCK_RESET_PEND	1'b0	RO	CPU LOCK RESET 发生时，该标志位置 1 0: 没有发生 LOCK_RESET 1: 发生过 LOCK_RESET
1	SOFT_RESET_PEND	1'b0	RW	软件对该位写 1 触发系统复位，同时会重新获取 FLASH 的 NVR/MAIN 数据 0: 不使能软件系统复位 1: 软件触发系统复位且该位置 1
0	SLEEP_PEND	1'b0	RO	软件写 LPCON[0]进入 sleep 模式时，该 bit 置 1，CPU 写 SLEEP_STA_CLR 清除。

24.2.2.16. 控制寄存器（PMUBK）

Width	Name	Reset	Property	Description
31:7	Reserved	-	-	-
6:4	PMU_LPDOS	3'h1	RW	低功耗模式 LDO 电压档位选择 b000=1.50V b001=1.45V b010=1.55V b011=1.60V b100=1.65V b101=1.75V b110=1.85V b111=1.25V 当 LPCON[7]为 1，系统进入 sleep 时，PMU_LPLDO 的电压挡位会切到 PMUBK 寄存器配置的档位，以获得更低功耗。
3:0	Reserved	-	-	-

25. 器件电子签名（E-SIGNATURE)

电子签名存放在闪存存储器模块的系统存储区域，可以通过 JTAG、DBG 或者 CPU 读取。它所包含的芯片识别信息在出厂时编写，用户固件或者外部设备可以读取电子签名，用以自动匹配不同配置的 CIU32M011、CIU32M031 系列微控制器。

25.1. 产品唯一身份标识寄存器（96 位)

产品唯一的身份标识非常合适

- 用来作为序列号(例如 USB 字符序列号或者其他的终端应用)
- 用来作为密码，在编写闪存时，将此唯一标识与软件加解密算法结合使用，提高代码在闪存存储器的安全性
- 用来激活带安全机制的自举过程

96 位的产品唯一身份标识所提供的参考号码对任意一个 CIU32M011、CIU32M031 系列微控制器，在任何情况下都是唯一的。用户在何种情况下，都不能修改这个身份标识。

关于这个寄存器的详细描述，请参考 嵌入式闪存 章节。

26. 调试支持（DEBUG)

CIU32M011、CIU32M031 系列内核内含硬件调试模块，支持复杂的调试操作。硬件调试模块允许内核在取指(指令断点)或访问数据(数据断点)时停止。内核停止时，内核的内部状态和系统的外部状态都是可以查询的。完成查询后，内核和外设可以被复原，程序将继续执行。当 CIU32M011、CIU32M031 系列微控制器连接到调试器并开始调试时，调试器将使用内核的硬件调试模块进行调试操作。

支持：串行调试接口

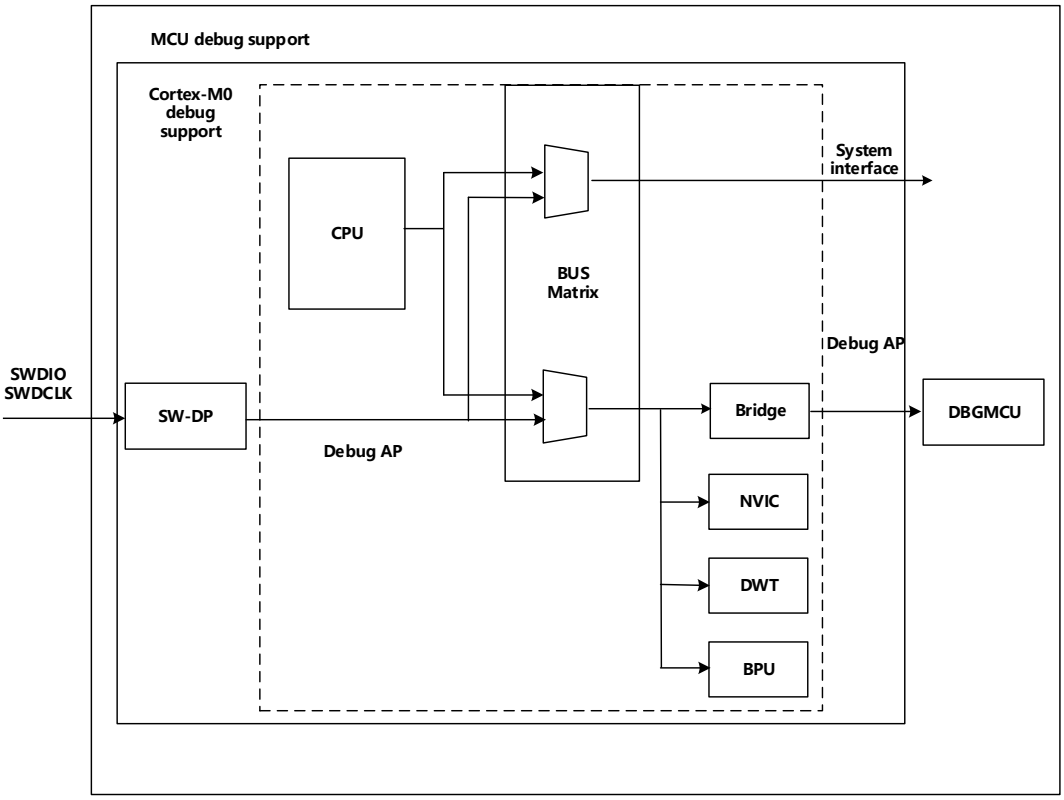


图 26-1 串行接口调试框图

CPU 内核提供集成的片上调试功能。它由以下部分组成

- SW-DP: 串行调试端口
- AHP-AP: AHB 访问端
- ITM: 执行跟踪单元
- FPB: 闪存指令断点
- DWT: 数据触发
- TPUI: 跟踪单元接口

26.1. 引脚分布和调试端口脚

不同封装有不同的有效引脚数。因此，某些与引脚相关的功能可能随封装而不同。

26.2. SWD 调试端口脚

CIU32M011、CIU32M031 的 SW-DP 接口引脚如下所示:

SWJ-DP 端口引脚名称	SW 调试接口		引脚分配
	类型	调试功能	
SWDIO	输入/输出	串行数据输入/输出	PA13
SWCLK	输入	串行时钟	PA14

SW-DP 接口引脚选择在 flash 用户自定义功能区域定义。

26.2.1. SWD 脚上的内部上拉和下拉

保证 SWD 的输入引脚不是悬空的是非常必要的，因为他们直接连接到 D 触发器控制着调试模式。必须特别注意 SWCLK 引脚，因为他们直接连接到一些 D 触发器的时钟端。

为了避免任何未受控制的 IO 电平，CIU32M011、CIU32M031 在 SWD 输入脚上嵌入了内部上拉和下拉。

SWDIO：默认状态是输入悬空

SWCLK：内部上/下拉可选

一旦 SWD IO 被用户代码释放，GPIO 控制器再次取得控制。这些 IO 口的状态将恢复到复位时的状态。

软件可以把这些 IO 口作为普通的 IO 口使用。

26.3. SW 调试端口

26.3.1. SW 协议介绍

此同步串行协议使用 2 个引脚

- SWCLK：从主机到目标的时钟信号
- SWDIO：双向数据信号

协议允许读写 2 个寄存器组(DPACC 和 APACC 寄存器组)。

数据位按 LSB 传输。

由于 SWDIO 为双向口，该引脚需有上拉（默认芯片内部无上下拉，建议使用 100K 电阻）。

按协议每次 SWDIO 方向改变时，需插入一个转换时间。在该期间内主机和目标都不驱动此信号线。转换时间的默认值是 1 个比特，但可以通过配置 SWCLK 频率来调节。

26.3.2. SW 协议序列

每个序列由 3 个阶段组成

- 主机发送包请求（8 位）
- 目标发送确认响应（3 位）
- 主机或目标发送数据（33 位）

比特位	名称	描述
0	起始	必须为 1
1	APnDP	0: 访问 DP　1: 访问 AP
2	RnW	0: 写请求　1: 读请求
4:3	A（3:2）	DP 或 AP 寄存器的地址
5	Parity	前面比特位的校验位
6	Stop	0
7	Park	不能由主机驱动，由于有上拉，目标永远读为 1

有关 DPACC 和 APACC 寄存器描述的详细资料，请参考 CPU 技术参考手册。

包请求后总是跟一个(缺省为 1 位)转换时间，此时主机和目标都不驱动线路。

比特位	名称	描述
0:2	ACK	001: 失败 010: 等待 100: 成功

当 ACK 为失败或等待，或者是一个回复读操作的 ACK，此 ACK 后有一个转换时间。

比特位	名称	描述
0:31	WDATA/RDATA	写或读的数据
32	Parity	32 位数据的奇偶校验位

读操作的数据传输操作后有一个转换时间。

26.4. MCU 调试

为了支持低功耗模式下(STOP/SLEEP)的调试，当 SWD 连接成功后，如果芯片处于 STOP 模式，则会被唤醒，如果芯片处于 SLEEP 模式，则会被复位。此复位不会让 SWD 失连。

27. 版本历史

表 27-1 版本更改履历

日期	版本号	修改范围
2022-08-19	V1.0	初始版本
2023-07-08	V1.1	完善引脚定义描述
2023-08-25	V1.2	1、供电电压范围修改为：3.6~5.5V 2、引脚分配图章节：更新引脚分配图

28. 联系方式

网址：www.hed.com.cn

地址：北京市昌平区北七家未来科技城南区中国电子网络安全和信息化产业基地 C 栋

邮编：102209

如果您在购买与使用过程中有任何意见或建议，请随时与我们联系。