PADAUK 應廣科技

PEC930

RISC-V 电机专用标准产品 32 位 BLDC 微控制器

使用手册

第 0.00 版

2026 年 04 月 01 日

Copyright ©2026 by PADAUK Technology Co., Ltd., all rights reserved.

6F-6, No.1, Sec. 3, Gongdao 5th Rd., Hsinchu City 30069, Taiwan, R.O.C.

TEL: 886-3-572-8688 www.padauk.com.tw

重要声明

应广科技保留权利在任何时候变更或终止产品，建议客户在使用或下单前与应广科技或代理商联系以取得最新、最正确的产品信息。

应广科技不担保本产品适用于保障生命安全或紧急安全的应用，应广科技不为此类应用产品承担任何责任。关键应用产品包括，但不仅限于，可能涉及的潜在风险的死亡，人身伤害，火灾或严重财产损失。

应广科技为服务客户所提供之任何编程软件，皆为服务与参考性质，不具备任何软件漏洞责任，应广科技不承担任何责任来自于因客户的产品设计所造成的任何损失。在应广科技所保障的规格范围内，客户应设计和验证他们的产品。为了尽量减少风险，客户设计产品时，应保留适当的产品工作范围安全保障。

©Copyright 2026, PADAUK Technology Co. Ltd

目录

©Copyright 2026, PADAUK Technology Co. Ltd

©Copyright 2026, PADAUK Technology Co. Ltd

©Copyright 2026, PADAUK Technology Co. Ltd

©Copyright 2026, PADAUK Technology Co. Ltd

©Copyright 2026, PADAUK Technology Co. Ltd

©Copyright 2026, PADAUK Technology Co. Ltd

©Copyright 2026, PADAUK Technology Co. Ltd

©Copyright 2026, PADAUK Technology Co. Ltd

©Copyright 2026, PADAUK Technology Co. Ltd

©Copyright 2026, PADAUK Technology Co. Ltd

图目录

©Copyright 2026, PADAUK Technology Co. Ltd

©Copyright 2026, PADAUK Technology Co. Ltd

©Copyright 2026, PADAUK Technology Co. Ltd

©Copyright 2026, PADAUK Technology Co. Ltd

表目录

©Copyright 2026, PADAUK Technology Co. Ltd

©Copyright 2026, PADAUK Technology Co. Ltd

修订历史

修订	日期	描述
0.00	2026/04/01	初版

使用警告
在使用芯片前，请务必认真阅读 PEC930 相关的 APN（应用注意事项）。
APN 下载地址为: https://www.padauk.com.tw/cn/product/search_list.aspx?kw=PEC

1. 概述

超低功耗 PEC930 采用高性能的 RISC-V 的 32 位微控制器，最高可工作于 60MHz，采用高速的嵌入式 Memory（SRAM 最大 4KB，Flash 程序/数据闪存最大 32KB）。最高可运行在 60MHz，内置高达 4 KB 的 SRAM 支持 0 等待周期用于 code 程序执行使用

集成外设包括:

- 12b SR <1Msps High Resolution ADC (16 通道)
- PGAX2、COMPX2, DAC(8b)X2
- 基础 Timer X2(TIM0,TIM1), 高级 Timer(EPWM,TIM2) , 低功耗 Timer(LPTIM)
- 多路通信串口: UART、SPI、I2C
- EPWM (最多可达 3 个独立出口或 3 个互补式出口).
- DSP 硬件加速模块包含: 32 位有符号除法器，32 位无符号开平方根

以上等丰富的外设接口，具有高整合度、高抗干扰、高可靠性的特点，这些特点使得 PEC930 微控制器系列可广泛适用三相直流无刷电机驱动控制、单相直流无刷电机驱动控制。

©Copyright 2026, PADAUK Technology Co. Ltd

2. 主要特性

2.1. CPU 特性

- 32-bit RISC-V CPU 核，支持 RV32E/M/C 扩充指令集
- 16 个 32 位通用寄存器
- 高效的 2 级执行流水线
- CPU 内置单周期 32 位 x32 位的硬件整数乘法阵列
- CPU 内置 32 位硬件除法器
- CPU 内置 64 位系统定时器
- CJTAG/JTAG 调试接口

2.2. 增强型内核中断控制器(ECLIC)

- 支持 2 个核心指定的内部中断，例如软件中断、定时器中断等。
- 支持 15 外部中断
- 支持软件动态可编程修改中断级别和中断优先级的数值
- 支持基于中断级别的中断嵌套

2.3. 电源管理

- 提供普通休眠(Sleep)、深度休眠(Deep Sleep) 低功耗模式
- 普通休眠(Sleep) 芯片电流: 3.3mA(Typ)，SYSCLK=60MHz 唤醒时间小于 2 us,
- 深度休眠(Deep Sleep)模式下，芯片电流 < 5uA(Typ)@Ta=25°C
- POR,PDR,LVR.
- 低电压检测，可 配置为中断或复位
- 唤醒@Sleep: 所有中断源都可唤醒
- 唤醒@Deep sleep: 所有 GPIO 引脚中断,WDG 与 LPTIM 都可唤醒

2.4. 内置储存器

- 指令存储器: 32KB FLASH ,8KB NVR
- 数据存储器: 4KB SRAM

2.5. 安全机制

- 片上看门狗（WDG），计数/计时的时钟源可配
- 低电压监控，当电压低于安全值时，输出中断或复位

©Copyright 2026, PADAUK Technology Co. Ltd

2.6. UART 通信

- 1 个通道
- 灵活可编程的波特率设定，支持业内标准 9600bps、19200bps、28800bps、38400bps、57600bps、115.2Kbps 等，或其它特殊应用的波特率
- 可编程的数据传输格式: 8 位数据、7 位数据+1 位奇偶校验、8 位数据+1 位奇偶校验、9 位数据
- 可编程奇或偶校验方式
- 可编程设定 0.5 位、1 位、1.5 位或 2 位停止位
- 数据收发全双工
- 硬件错误检测

2.7. SPI 通信

- 1 个通道
- 传输字大小 2bit 可配，支持 8/16/24/32bits datasize
- 模块支持 LSB 和 MSB 两种发送数据配置
- 主机最高传输速率可达 10Mbps（系统时钟> 20MHz 时）

2.8. I2C 通信

- 1 个通道
- 支持 SMBus
- 支持多主机 I2C 总线，支持主机或者从机工作模式。
- 标准模式 100Kbit/s，高速模式可达 400Kbit/s，超高速模式可达 1Mbit/s.

2.9. 高精度 ADC

- 正常工作时功耗<130uA/MHz
- 1Msps 以下采样速率，12 位 SAR 型 ADC
- 16 通道: 12 路的外部引脚,1 路内部温度传感器电压,2 路的 PGA 输出, 1 路的内建参考电压 1.5V
- 外部参考电压: VDD
- 触发条件: software, Hardware:TIM0(ovf), EPWM(ovf,udf,ovf&udf,raising,falling), COMP

2.10. 定时器（Timer）

- TIM0/1: 16 位宽定时器，带预分频。
- LPTIM: 16 位宽定时器，带预分频，计数器时钟源可配，能够在休眠模式下工作
- TIM2: 20 位宽定时器，带预分频，支持递增、递减、中心对齐(对称与不对称输出)，能够输出 4 路 PWM 输出.
- 所有标准外设均具有使能/禁止控制；禁止时，耗电量可忽略不计

2.11. 增强型 EPWM

- 20 位宽定时器，带预分频，支持递增、递减、中心对齐(对称与不对称输出)，能够输出 6 路 3 组互补 PWM.
- 标准外设都配有使能控制，禁止时功耗基本可以忽略
- 支持 2 种对齐模式: 边沿对齐，中心对齐
- 中心对齐模式支持对称计数和非对称计数
- 互补的 PWM 中，支持可编程死区发生器
- PWM 边沿或周期可触发启动 ADC 转换
- 刹车保护源:
  - 外部 BKIN 电平信号(高电平或低电平)
  - 模拟比较器0,1的输出(输出高或输出低)
  - ADC值比较(可选任意两个通道为刹车源)
  - 低电压检测.
  - WDG 溢出事件发生
  - CPU 异常事件发生

2.12. 运算放大器(PGA0/1)

- 增益可选择: 1~16 倍
- PGA 输出/输入:
  - 输出到 ADC 通道
  - 输出到比较器
  - 输出到引脚
  - 放大器输入端（同相）内建箝位二极管，马达相位电流可透过匹配电阻直接馈入放大器输入端，从而简化 MOSFET 的电流取样电路

2.13. DSP 硬件加速模块

- 内置 32 位有符号除法器硬件
- 内置 32 位无符号开平方根硬件

2.14. 比较器(COMP0/1)

- 正端多路可选
- 负端可选端口输入与 DAC
- 支持迟滞电压
- 支持比较器输出触发 EPWM 刹车

©Copyright 2026, PADAUK Technology Co. Ltd

2.15. DAC(8 bit)X2

- 输入参考电压可选: VDD
- 输出电压多级可选

2.16. 时钟源

- 内置高速 RC 振荡器时钟: 60 MHz，精度+5% ~ -2.5% (-40°C ~ 85°C)，精度+3.5% ~ -2.5% (0°C ~ 85°C)
- 内置低速 RC 振荡器时钟: 32KHz，精度±40%，模块功耗 0.5uA，可作为看门狗(WDG)和低功耗定时器（LPTIM）时钟
- 分频器时钟：内部 60M RC 时钟的特定整数分频

2.17. GPIO

- 提供 22 引脚都支持外部中断
- GPIO 支持外设可复用

2.18. 内部温度传感器

- 内建温度传感器: -1.65mV/℃

2.19. 硬件 CRC-16/32 模块

- 支持基于 Byte, Half-word, Word 的写操作
- 可选择的 CRC 多项式

2.20. 12 字节(96 位)UID

- 为每个芯片提供 12 字节的唯一识别符

2.21. 开发工具

- 支持 JTAG 和两线 CJTAG 调试接口

2.22. 工作环境

- 供电电压范围: 2.5 ~ 5.5 V
- 工作环境温度: -40 ~ +85℃

2.23. 封装形式

- PEC930-Y24A: SSOP24(150MIL)
- PEC930-2J24A: QFN24(4*4*0.75MM)

©Copyright 2026, PADAUK Technology Co. Ltd

3. 功能框图

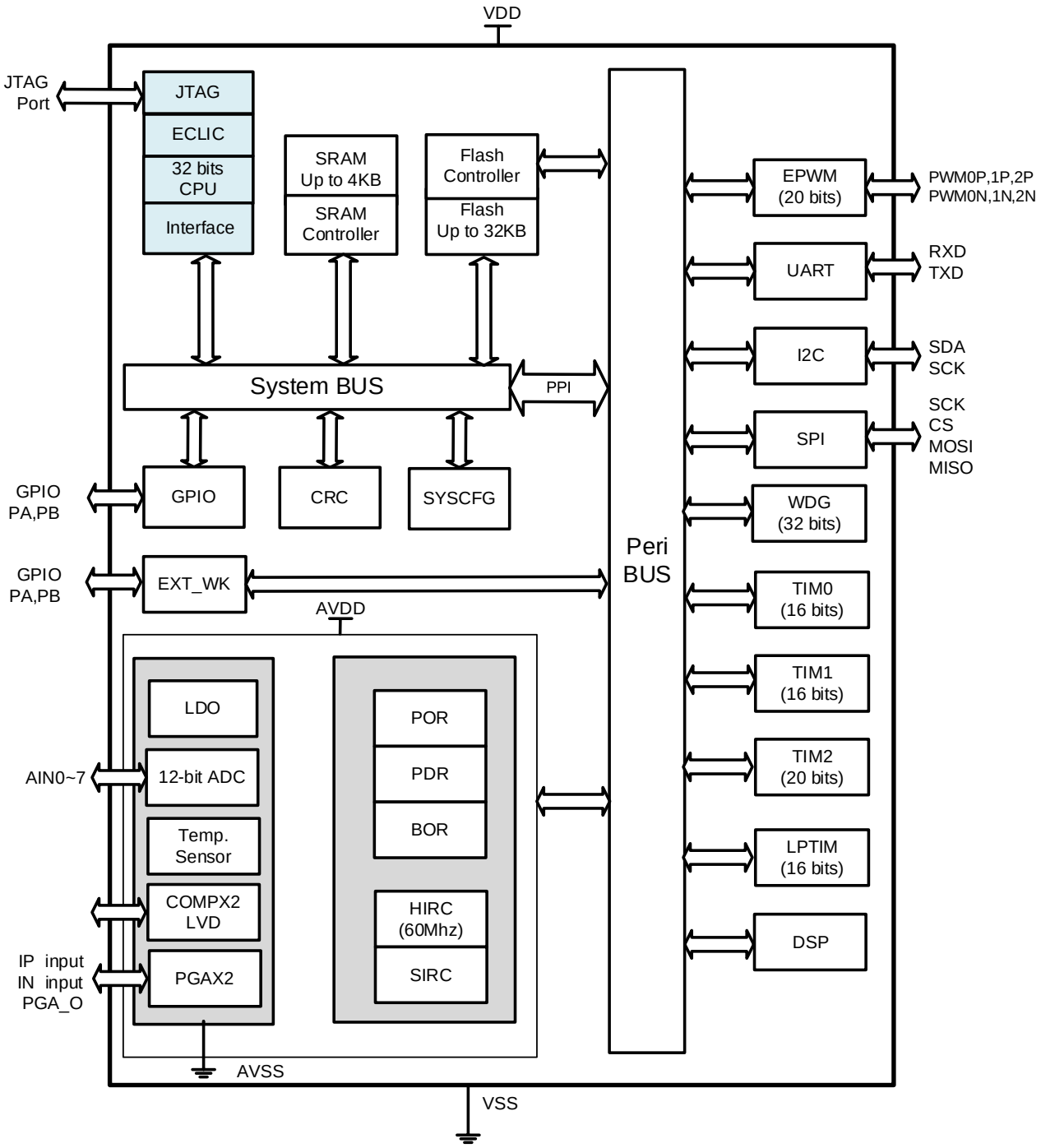


图 3-1 PEC930 功能框图

3.1. CPU

PEC930 集成最新一代的嵌入式高效率、低功耗、小面积的 RISC-V 内核解决方案的 32 位处理器内核带有乘除法器.内核的主要特点有

- 2 级流水线单发射处理器
- 支持两线 CJTAG 与 JTAG 调试接口
- 内核能够通过常见的 WFI（Wait for Interrupt）和 WFE（Wait for Event）机制支持休眠（Sleep/DeepSleep）模式以实现较低的动态和静态功耗

3.2. On-Chip Memory

该设备具有以下特点:

- 内置 FLASH 分为两个数组:
  - 32 KB Main area，用于存放程序和数据.
  - 8 KB NVR area(7 KB 供用户存放数据, 1 KB 保留给系统使用)
- 内置 4KB SRAM ,可操作在 Max. 60MHz.

3.3. CRC 计算单元

CRC(循环冗余校验)计算单元使用一个固定的多项式发生器，产生一个 CRC 码。在众多的应用中，基于 CRC 的技术被用于验证数据传输或存储的一致性。CRC 计算单元可用于在数据传输过程中对讯框或数据计算 CRC 签名，并与发送端在封包生成时附带的签名进行比对，以验证数据在通讯过程中是否遭到破坏或发生错误。

3.4. 低功耗模式

PEC930 系列产品支持两种低功耗模式可以在要求低功耗、短启动时间和多种唤醒事件之间达到最佳的平衡:

- 休眠模式 (Sleep Mode):

  在休眠模式，只有 CPU 停止，所有外设处于工作状态并可在发生中断/事件时唤醒 CPU。
- 待机深度休眠模式 (DeepSleep Mode):

  在保持 SRAM 和寄存器内容不丢失的情况下，深度休眠模式可以达到最低的电能消耗。在深度休眠模式下内部 HIRC 振荡器被关闭，调压器 (LDO) 可以被置于低功耗模式模式，若 LIRC 振荡器被关闭,可以通过任一配置成 IO 中断的信号把微控制器从深度休眠模式中唤醒:若 LIRC 震荡器配置成开启则也可以使用 WDG 或 LPTIM 唤醒。

3.5. 温度传感器

温度传感器产生一个随温度线性变化的电压，转换范围在 2.5V < VDDA < 5.5V 之间。温度传感器在内部被连接到 ADC 输入通道上，用于将传感器的输出转换到数字数值

3.6. ADC

芯片内有一个 12 位的模数转换器 ADC，该 ADC 有 16 个通道，允许 ADC 测量 12 个外部输入引脚电压，及其它内部电压通道,每次采样间隔需要为大于 1us.

3.7. SPI

SPI 串行通讯标准由 Motorola 公司建立，它基于 3 线连接方式实现单片机和外围器件，或者单片机之间的数据通讯。通过从器件的通讯选择控制（片选），可以构成主从方式的 SPI 总线

3.8. 定时器（Timer）

提供如下定时器资源

- 2 个 16 位基本定时器 (TIM0,TIM1)
- 2 个 20 位高级定时器 (TIM2, EPWM)
- 1 个 16 位低功耗唤醒定时器 (LPTIM)
- 1 个 16 位看门狗定时器 (WDG)

3.9. I2C

I2C 是嵌入式系统设计中经常被用到的一种串行通讯总线。它基于 SCL（串行时钟）和 SDA（串行数据）双线联机，以主从方式实现多个互联器件之间的双向数据通讯

3.10. EPWM

该模块为驱动电机提供 3 路 PWM 脉宽调制电路，对应功能引脚位 EPWM0P、EPWM0N、EPWM1P、EPWM1N、EPWM2P、EPWM2N.

互补式 PWM:

(1) EPWM0P=EPWM_CH1, EPWM0N= EPWM_CH1N=~(EPWM_CH1)

(2) EPWM1P=EPWM_CH2, EPWM1N= EPWM_CH2N=~(EPWM_CH2)

(2) EPWM2P=EPWM_CH3, EPWM2N= EPWM_CH3N=~(EPWM_CH3)

3.11. UART

芯片内集成了 1 个 UART 通讯模块，其传输速度最高可达 115.2kbps。

©Copyright 2026, PADAUK Technology Co. Ltd

3.12. ECLIC 中断控制器

ECLIC (增强型内核中断控制器) 可用于多个中断源管理，内核中的所有类型 (除了调试中断之外) 的中断都由 ECLIC 统一进行管理，内核支持的中断类型包括外部中断与内部中断。

外部中断	IRQ 編號	例外或中斷	優先權 (預設)	向量程式地址	類型
	0	保留	（最低）	MTVT +4*0	保留
	1	保留		MTVT +4*1	保留
	2	保留		MTVT +4*2	保留
	3	Machine Software interrupt		MTVT +4*3	电平
	4	保留		MTVT +4*4	保留
	5	保留		MTVT +4*5	保留
	6	保留		MTVT +4*6	保留
	7	Machine Timer interrupt		MTVT +4*7	电平
	8	保留		MTVT +4*8	保留
	9	保留		MTVT +4*9	保留
	10	保留		MTVT +4*10	保留
	11	保留		MTVT +4*11	保留
	12	保留		MTVT +4*12	保留
	13~16	保留		MTVT +4*(13~16)	保留
	17	保留		MTVT +4*17	保留
	18	保留		MTVT +4*18	保留
1	19	TIM0		MTVT +4*19	电平
2	20	TIM1		MTVT +4*20	电平
3	21	TIM2		MTVT +4*21	电平
4	22	LPTIM		MTVT +4*22	电平
5	23	WDG		MTVT +4*23	电平
6	24	SPI		MTVT +4*24	电平
7	25	UART		MTVT +4*25	电平
8	26	I2C-		MTVT +4*26	电平
9	27	GPIOA		MTVT +4*27	电平
10	28	GPIOB		MTVT +4*28	电平
11	29	COMP0		MTVT +4*29	电平
12	30	COMP1		MTVT +4*30	电平
13	31	ADC		MTVT +4*31	电平
14	32	EPWM		MTVT +4*32	电平
15	33	LVD	（最高）	MTVT +4*33	电平

表 3-1 中断向量表

©Copyright 2026, PADAUK Technology Co. Ltd

3.13. DSP

DSP 硬件加速模块包含:

- 32 位有符号除法器
- 32 位无符号开平方根

3.14. CRC

用于核实数据传输或者数据存储的正确性和完整性，协议能选择 CRC-16/32

3.15. GPIO

提供最多 22 个通用 IO(GPIO)可用

四种端口模式

- 高阻抗/关闭输入缓冲器模式
- 上拉/下拉输入模式
- 开漏输出模式
- 推挽输出模式

©Copyright 2026, PADAUK Technology Co. Ltd

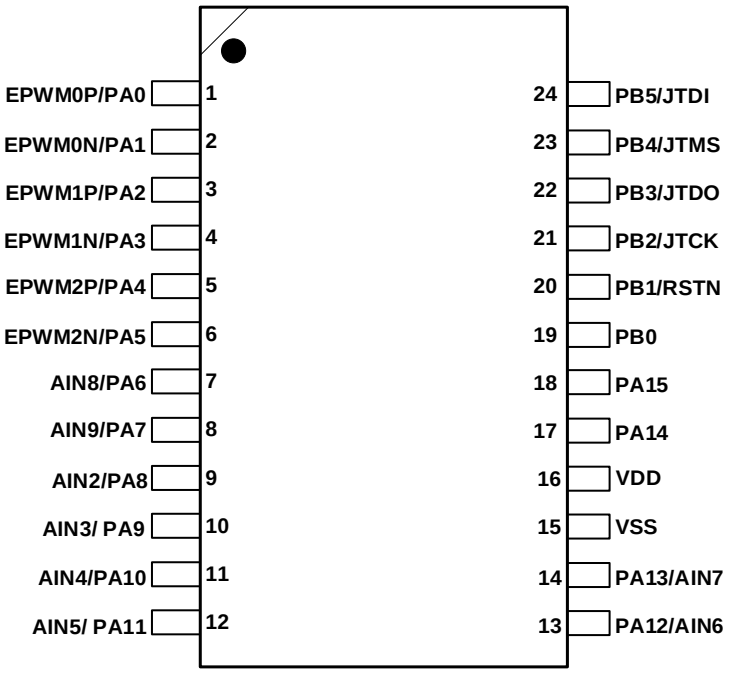
4. 设备概述

外设		PEC930
引脚数		24
通用输入/输出口(GPIO)		22
CPU	内核	RISC-V 32 位核心
	时钟频率	最高 60 MHz
	乘法器	32 位(1T)
	除法器	32 位(17T)
	系统定时器	64 位
FLASH		32 KB
SRAM		4 KB
DSP		32 位有符号除法器，32 位无符号开平方根
定时器	基础(16 位)	TIM0,TIM1
	高级(20 位)	TIM2, EPWM
	省电(16 位)	LPTIM
	看门狗(32 位)	WDG
工作电压范围		2.5V ~ 5.5V
工作温度		-40℃ ~ 85℃
调试功能		CJTAG(二线)/JTAG(四线)
唯一标识符(UID)		12 bytes
通信界面	UART	1
	SPI	1
	I2C	1
CRC 校验		1 (CRC16/32)
内部温度传感器		1
时钟	内部高速晶振(HIRC)	60 MHz
	内部低速晶振(SIRC)	32 KHz
12 位 ADC		1 (16 通道)
比较器(COMP)		2
运算放大器(PGA)		2
封装		SSOP24/QFN24

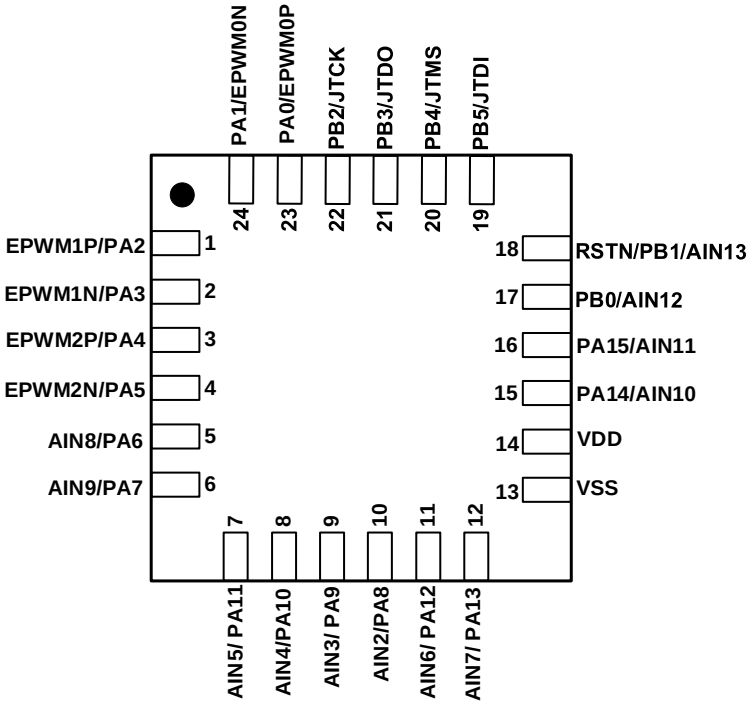
表 4-1 設備概述

5. 引腳配置與功能說明

5.1. 引脚定义



PEC930-Y24A：SSOP24(150MIL)



PEC930-2J24A：QFN24(4*4*0.75MM)

©Copyright 2026, PADAUK Technology Co. Ltd

5.2. 引脚功能配置

SSOP24 引脚编号	引脚名称	引脚类型	配置	预设
1	PA0	I/O		
2	PA1	I/O		
3	PA2	I/O		
4	PA3	I/O		
5	PA4	I/O		
6	PA5	I/O		
7	PA6	I/O		
8	PA7	I/O		
9	PA8	I/O		
10	PA9	I/O		
11	PA10	I/O		
12	PA11	I/O		
13	PA12	I/O		
14	PA13	I/O		
15	VSS	GND		
16	VDD	Power		
17	PA14	I/O		
18	PA15	I/O		
19	PB0	I/O		
20	PB1	I/O	RSTN (POR latch)	RSTN
21	PB2	I/O	JTAG_TCK/CJ_TCK	JTAG_TCK/CJ_TCK
22	PB3	I/O	JTAG_TDO	JTAG_TDO
23	PB4	I/O	JTAG_TMS/CJ_TDIO	JTAG_TMS/CJ_TDIO
24	PB5	I/O	JTAG_TDI	JTAG_TDI

表 5-1 引脚功能配置

©Copyright 2026, PADAUK Technology Co. Ltd

5.3. 引脚复合功能

引脚编号		引脚名称	模拟功能			数字功能						特殊功能管脚
QFN 24	SSOP 24		ADC	PGA	COMP	UART	SPI	I2C	EPWM/ TIM2 in	BKIN	EPWM/ TIM2 out	
23	1	PA0				RXD/ TXD		SDA/ SCL			EPWM0P	
24	2	PA1				TXD/ RXD		SCL/ SDA			EPWM0N	
1	3	PA2					SCK				EPWM1P/ TIM2CH1	
2	4	PA3					MOSI/ MISO				EPWM1N/ TIM2CH2	
3	5	PA4					MISO/ MOSI				EPWM2P/ TIM2CH3	
4	6	PA5					CS				EPWM2N/ TIM2CH4	
5	7	PA6	AIN8		C0_N					BKIN		
6	8	PA7	AIN9	A0_O	C0_O					BKIN		
10	9	PA8	AIN2	A0_P	C1_P0/ C0_P0					BKIN		
9	10	PA9	AIN3	A0_N						BKIN		
8	11	PA10	AIN4	A1_N						BKIN		
7	12	PA11	AIN5	A1_P	C1_P1/ C0_P1					BKIN		
11	13	PA12	AIN6		C1_N					BKIN		
12	14	PA13	AIN7	A1_O	C1_O					BKIN		
13	15	VSS										Ground
14	16	VDD										Power
15	17	PA14	AIN10		C1_P2/ C0_P2		SCK		ECAP1,2,3/ EPETR/ T2ETR/ T2CAP1,2,3			MCO[1*]
16	18	PA15	AIN11		C1_P3/ C0_P3	TXD/ RXD	MOSI/ MISO	SCL	ECAP1,2,3/ EPETR/ T2ETR/ T2CAP1,2,3			
17	19	PB0	AIN12			RXD/ TXD	MISO/ MOSI	SDA	ECAP1,2,3/ EPETR/ T2ETR/ T2CAP1,2,3			
18	20	PB1	AIN13				CS	SCL/ SDA	EPETR/ T2ETR	BKIN	TIM2CH1	RSTN/ MCO[1*]
22	21	PB2					SCK	SCL/ SDA	ECAP1,2,3/ EPETR/ T2ETR/ T2CAP1,2,3		EPWM2N/ TIM2CH1	JTAG_TCK/ CJ_TCK/ OSCIN
21	22	PB3				TXD	MOSI/ MISO	SCL	ECAP1,2,3/ EPETR/ T2ETR/ T2CAP1,2,3	BKIN	EPWM2P	JTAG_TDO
20	23	PB4				TXD/ RXD	MISO/ MOSI		ECAP1,2,3/ EPETR/ T2ETR/ T2CAP1,2,3		EPWM1P	JTAG_TMS/ CJ_TDIO
19	24	PB5				RXD/ TXD	CS	SDA	EPETR/ T2ETR	BKIN	EPWM1N/ TIM2CH1	JTAG_TDI

表 5-2 引脚复合功能

©Copyright 2026, PADAUK Technology Co. Ltd

注:

1. BKIN 为 EPWM 的刹车外部输入端口可选端口。
2. 所有 IO 支持 timer 计数, Capture, 外部中断和唤醒。
3. 所有的 I2C 口内部都支持上拉选择; I2C 引脚内置 10kΩ 上拉电阻，可软件控制开启关闭上拉
4. RSTN 为复位引脚，引脚内置 10kΩ 上拉电阻，固定开启上拉，当 RSTN 功能切换为 GPIO 功能后，上拉可以关闭。
5. [1*]: MCO: 系统时钟输出
6. [2*]: 互补式 PWM
   (1) EPWM0P=PWMA, EPWM0N=PWMD=~(PWMA)
   (2) EPWM1P=PWMB, EPWM1N=PWME=~(PWMB)
   (3) EPWM2P=PWMC, EPWM2N=PWMF=~(PWMC)

©Copyright 2026, PADAUK Technology Co. Ltd

5.4. 引脚复合选择

引脚编号		PAx_AFR[2:0]/PBx_AFR[2:0] (PA0_AFR[2:0]~PA5_AFR[2:0],PA14_AFR[2:0], PA15_AFR[2:0], PB0_AFR[2:0]~PB5_AFR[2:0])							特殊功能	外部输入	AMISC/ ADC 寄存器
QFN 24	SSOP 24	0	1	2	3	4	5	6	配置	IP 功能	AMISC/ ADC 功能
23	1	PA0	RXD	TXD	SDA[1*]	SCL[1*]		EPWM0P			
24	2	PA1	TXD	RXD	SCL[1*]	SDA[1*]		EPWM0N			
1	3	PA2	SCK				TIM2CH1	EPWM1P			
2	4	PA3	MOSI	MISO			TIM2CH2	EPWM1N			
3	5	PA4	MISO	MOSI			TIM2CH3	EPWM2P			
4	6	PA5	CS				TIM2CH4	EPWM2N			
5	7	PA6								BKIN	AIN8/ C0_N
6	8	PA7								BKIN	AIN9/ C0_O/ A0_O
10	9	PA8								BKIN	AIN2/ C1_P0/ C0_P0/ A0_P
9	10	PA9								BKIN	AIN3/ A0_N
8	11	PA10								BKIN	AIN4/ A1_N
7	12	PA11								BKIN	AIN5/ C1_P1/ C0_P1/ A1_P
11	13	PA12								BKIN	AIN6/ C1_N
12	14	PA13								BKIN	AIN7/ C1_O/ A1_O
13	15	VSS									
14	16	VDD									
15	17	PA14			SCK			MCO		ECAP1,2,3/ EPETR/ T2ETR/ T2CAP1,2,3	C1_P2/ C0_P2/ AIN10
16	18	PA15	TXD;	RXD	MOSI	MISO		SCL[1*]		ECAP1,2,3/ EPETR/ T2ETR/ T2CAP1,2,3	C1_P3/ C0_P3/ AIN11

©Copyright 2026, PADAUK Technology Co. Ltd

17	**19**	PB0	RXD	TXD	MISO	MOSI		SDA[1*]		ECAP1,2,3/ EPETR/ T2ETR/ T2CAP1,2,3	AIN12
18	**20**	PB1	SCL [1*]	SDA [1*]	CS	TIM2CH1		MCO	RSTN	EPETR/ T2ETR/ BKIN	AIN13
22	**21**	PB2	SDA [1*]	SCL [1*]	SCK	TIM2CH1		EPWM2N	JTAG_TCK /CJ_TCK	ECAP1,2,3/ EPETR/ T2ETR/ T2CAP1,2,3	OSCIN
21	**22**	PB3	TXD	SCL [1*]	MOSI	MISO		EPWM2P	JTAG_TDO	ECAP1,2,3/ EPETR/ T2ETR/ T2CAP1,2,3/ BKIN	
20	**23**	PB4	TXD	RXD	MISO	MOSI		EPWM1P	JTAG_TMS /CJ_TDIO	ECAP1,2,3/ EPETR/ T2ETR/ T2CAP1,2,3	
19	**24**	PB5	RXD	TXD	CS	TIM2CH1	SDA[1*]	EPWM1N	JTAG_TDI	EPETR/ T2ETR/ BKIN	

表 5-3 引脚复合選擇

注:

(1) 所有的 I2C 口内部都支持上拉选择; I2C 引脚内置 10kΩ 上拉电阻，可软件控制开启关闭上拉

(2) 当 ADC, COMP 和 PGA 通道被使用，需要关闭数字输入缓冲器.

(3) [1*]: 当选择 SDA 或 SCL，设置 I2CEN = 1 & FN2_AFR.I2CPULL = 1，内部上拉电组 10kΩ 使能

©Copyright 2026, PADAUK Technology Co. Ltd

引腳編號		GPIO端口	WK [5:0]	BKIN [3:0]	EPETR [2:0]	ECAP1 [2:0]	ECAP2 [2:0]	ECAP3 [2:0]	T2ETR [2:0]	T2CAP1 [2:0]	T2CAP2 [2:0]	T2CAP3 [2:0]
QFN	SSOP		WK	EPWM (BKIN)	EPWM	EPWM	EPWM	EPWM	TIM2	TIM2	TIM2	TIM2
24	24											
23	1	PA0	WK(1)									
24	2	PA1	WK(2)									
1	3	PA2	WK(3)									
2	4	PA3	WK(4)									
3	5	PA4	WK(5)									
4	6	PA5	WK(6)									
5	7	PA6	WK(7)	BKIN(1)								
6	8	PA7	WK(8)	BKIN(2)								
10	9	PA8	WK(9)	BKIN(3)								
9	10	PA9	WK(10)	BKIN(4)								
8	11	PA10	WK(11)	BKIN(5)								
7	12	PA11	WK(12)	BKIN(6)								
11	13	PA12	WK(13)	BKIN(7)								
12	14	PA13	WK(14)	BKIN(8)								
13	15	VSS										
14	16	VDD										
15	17	PA14	WK(15)		EPETR(1)	ECAP1 (1)	ECAP2 (1)	ECAP3 (1)	T2ETR (1)	T2CAP1 (1)	T2CAP2 (1)	T2CAP3 (1)
16	18	PA15	WK(16)		EPETR(2)	ECAP1 (2)	ECAP2 (2)	ECAP3 (2)	T2ETR (2)	T2CAP1 (2)	T2CAP2 (2)	T2CAP3 (2)
17	19	PB0	WK(17)		EPETR(3)	ECAP1 (3)	ECAP2 (3)	ECAP3 (3)	T2ETR (3)	T2CAP1 (3)	T2CAP2 (3)	T2CAP3 (3)
18	20	RSTN (PB1)	WK(18)	BKIN(9)	EPETR(4)				T2ETR (4)			
22	21	PB2	WK(19)		EPETR(5)	ECAP1 (4)	ECAP2 (4)	ECAP3 (4)	T2ETR (5)	T2CAP1 (4)	T2CAP2 (4)	T2CAP3 (4)
21	22	PB3	WK(20)	BKIN(10)	EPETR(6)	ECAP1 (5)	ECAP2 (5)	ECAP3 (5)	T2ETR (6)	T2CAP1 (5)	T2CAP2 (5)	T2CAP3 (5)
20	23	PB4	WK(21)		EPETR(7)	ECAP1 (6)	ECAP2 (6)	ECAP3 (6)	T2ETR (7)	T2CAP1 (6)	T2CAP2 (6)	T2CAP3 (6)
19	24	PB5	WK(22)	BKIN(11)	EPETR(8)				T2ETR (8)			

表 5-4 引脚复合输入选择

©Copyright 2026, PADAUK Technology Co. Ltd

5.5. 引脚功能说明

SSOP24	引脚名称	类型	描述
1	PA0	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	RXD	IO	UART 数据输入
	TXD	IO	UART 数据输出
	SDA	IO	I2C 数据输入输出
	WK	IO	唤醒
	EPWM0P	IO	EPWM PWM 正相输出通道 0
2	PA1	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	TXD	IO	UART 数据输出
	RXD	IO	UART 数据输入
	SCL	IO	I2C 时钟输入输出
	WK	IO	唤醒
	EPWM0N	IO	EPWM PWM 反相输出通道 0
3	PA2	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	SCK	IO	SPI 时钟输入/输出
	WK	IO	唤醒
	EPWM1P	IO	EPWM PWM 正相输出通道 1
	TIM2CH1	IO	TIM2 PWM 输出通道 1
4	PA3	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	MOSI	IO	SPI Master 输出/Slave 输入数据信号
	WK	IO	唤醒
	MISO	IO	SPI Master 输入/Slave 输出数据信号
	EPWM1N	IO	EPWM PWM 反相输出通道 1
	TIM2CH2	IO	TIM2 PWM 输出通道 2
5	PA4	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	MISO	IO	SPI Master 输入/Slave 输出数据信号
	WK	IO	唤醒
	MOSI	IO	SPI Master 输出/Slave 输入数据信号
	EPWM2P	IO	EPWM PWM 正相输出通道 2
	TIM2CH3	IO	TIM2 PWM 输出通道 3
6	PA5	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能

©Copyright 2026, PADAUK Technology Co. Ltd

SSOP24	引脚名称	类型	描述
	CS	IO	SPI 片选择致能
	WK	IO	唤醒
	EPWM2N	IO	EPWM PWM 反相输出通道 2
	TIM2CH4	IO	TIM2 PWM 输出通道 4
7	PA6	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	AIN8	A	ADC 通道 8 输入
	C0_N	A	COMP0 反相输入通道
	WK	IO	唤醒
	BKIN	IO	EPWM 刹车信号输入
8	PA7	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	AIN9	A	ADC 通道 9 输入
	A0_O	A	PGA0 输出
	C0_O	IO	COMP0 输出
	WK	IO	唤醒
	BKIN	IO	EPWM 刹车信号输入
9	PA8	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	AIN2	A	ADC 通道 2 输入
	A0_P	A	PGA0 正相输入
	C1_P0/C0_P0	A	COMP1 正端输入通道 0/COMP0 正端输入通道 0
	WK	IO	唤醒
	BKIN	IO	EPWM 刹车信号输入
10	PA9	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	AIN3	A	ADC 通道 3 输入
	A0_N	A	PGA0 反相输入
	WK	IO	唤醒
	BKIN	IO	EPWM 刹车信号输入
11	PA10	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	AIN4	A	ADC 通道 4 输入
	A1_N	A	PGA1 反相输入
	WK	IO	唤醒
	BKIN	IO	EPWM 刹车信号输入

©Copyright 2026, PADAUK Technology Co. Ltd

SSOP24	引脚名称	类型	描述
12	PA11	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	AIN5	A	ADC 通道 5 输入
	A1_P	A	PGA1 正相输入
	C1_P1/C0_P1	A	COMP1 正端输入通道 1/COMP0 正端输入通道 1
	WK	IO	唤醒
	BKIN	IO	EPWM 刹车信号输入
13	PA12	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	AIN6	A	ADC 通道 6 输入
	C1_N	A	COMP1 负端输入通道
	WK	IO	唤醒
	BKIN	IO	EPWM 刹车信号输入
14	PA13	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	AIN7	A	ADC 通道 7 输入
	A1_O	A	PGA1 输出
	C1_O	IO	COMP1 输出
	WK	IO	唤醒
	BKIN	IO	EPWM 刹车信号输入
15	VSS	Gound	0V Ground
16	VDD	Power	5V Power
17	PA14	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	AIN10	A	ADC 通道 10 输入
	C1_P2/C0_P2	A	COMP1 正端输入通道 2/COMP0 正端输入通道 2
	SCK	IO	SPI 时钟输入输出
	WK	IO	唤醒
	MCO	IO	系统时钟输出
	EPETR	IO	EPWM 外部触发输入
	ECAP1,2,3	IO	EPWM 捕获输入通道
	T2ETR	IO	TIM2 外部触发输入
	T2CAP1,2,3	IO	TIM2 捕获输入通道
18	PA15	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	AIN11	A	ADC 通道 11 输入

©Copyright 2026, PADAUK Technology Co. Ltd

SSOP24	引脚名称	类型	描述
	C1_P3/C0_P3	A	COMP1 正端输入通道 3/COMP0 正端输入通道 3
	TXD	IO	UART 数据输出
	MOSI	IO	SPI Master 输出/Slave 输入数据信号
	MISO	IO	SPI Master 输入/Slave 输出数据信号
	SCL	IO	I2C 时钟输入输出
	WK	IO	唤醒
	EPETR	IO	EPWM 外部触发输入
	ECAP1,2,3	IO	EPWM 捕获输入通道
	T2ETR	IO	TIM2 外部触发输入
	T2CAP1,2,3	IO	TIM2 捕获输入通道
19	PB0	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	AIN12	A	ADC 通道 12 输入
	RXD	IO	UART 数据输入
	TXD	IO	UART 数据输出
	MISO	IO	SPI Master 输入/Slave 输出数据信号
	MOSI	IO	SPI Master 输出/Slave 输入数据信号
	SDA	IO	I2C 数据输入输出
	WK	IO	唤醒
	EPETR	IO	EPWM 外部触发输入
	ECAP1,2,3	IO	EPWM 捕获输入通道
	T2ETR	IO	TIM2 外部触发输入
	T2CAP1,2,3	IO	TIM2 捕获输入通道
20	PB1	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	AIN13	A	ADC 通道 13 输入
	CS	IO	SPI 片选择致能
	SCL	IO	I2C 时钟输入输出
	WK	IO	唤醒
	BKIN	IO	EPWM 刹车信号输入
	TIM2CH1	IO	TIM2 PWM 输出通道 1
	RSTN	I	预设复位输入端口，低有效，芯片复位
	MCO	IO	系统时钟输出

©Copyright 2026, PADAUK Technology Co. Ltd

SSOP24	引脚名称	类型	描述
	EPETR	IO	EPWM 外部触发输入
	T2ETR	IO	TIM2 外部触发输入
21	PB2	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	SCK	IO	SPI 时钟输入输出
	SCL	IO	I2C 时钟输入输出
	WK	IO	唤醒
	EPWM2N	IO	EPWM PWM 反相输出通道 2
	TIM2CH1	IO	TIM2 PWM 输出通道 1
	JTAG_TCK/CJ_TCK	IO	JTAG 串口
	OSCIN	IO	外部振荡输入
	EPETR	IO	EPWM 外部触发输入
	ECAP1,2,3	IO	EPWM 捕获输入通道
	T2ETR	IO	TIM2 外部触发输入
	T2CAP1,2,3	IO	TIM2 捕获输入通道
22	PB3	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	TXD	IO	UART 数据输出
	MOSI	IO	SPI Master 输出/Slave 输入数据信号
	MISO	IO	SPI Master 输入/Slave 输出数据信号
	SCL	IO	I2C 时钟输入输出
	WK	IO	唤醒
	BKIN	IO	EPWM 刹车信号输入
	EPWM2P	IO	EPWM PWM 正相输出通道 2
	JTAG_TDO	IO	JTAG 串口
	EPETR	IO	EPWM 外部触发输入
	ECAP1,2,3	IO	EPWM 捕获输入通道
	T2ETR	IO	TIM2 外部触发输入
	T2CAP1,2,3	IO	TIM2 捕获输入通道
23	PB4	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	TXD	IO	UART 数据输出

©Copyright 2026, PADAUK Technology Co. Ltd

SSOP24	引脚名称	类型	描述
	RXD	IO	UART 数据输入
	MISO	IO	SPI Master 输入/Slave 输出数据信号
	MOSI	IO	SPI Master 输出/Slave 输入数据信号
	WK	IO	唤醒
	EPWM1P	IO	EPWM PWM 正相输出通道 1
	JTAG_TMS/CJ_TDIO	IO	JTAG 串口
	EPETR	IO	EPWM 外部触发输入
	ECAP1,2,3	IO	EPWM 捕获输入通道
	T2ETR	IO	TIM2 外部触发输入
	T2CAP1,2,3	IO	TIM2 捕获输入通道
24	PB5	IO	GPIO 通过寄存器配置输入输出，上、下拉等功能
	RXD	IO	UART 数据输入
	TXD	IO	UART 数据输出
	CS	IO	SPI 片选择致能
	SDA	IO	I2C 数据输入输出
	WK	IO	唤醒
	BKIN	IO	EPWM 刹车信号输入
	EPWM1N	IO	EPWM PWM 反相输出通道 1
	TIM2CH1	IO	TIM2 PWM 输出通道 1
	JTAG_TDI	IO	JTAG 串口
	EPETR	IO	EPWM 外部触发输入
	T2ETR	IO	TIM2 外部触发输入

表 5-5 引脚功能说明

Note: IO: 逻辑输入/输出； A: 模拟输出

©Copyright 2026, PADAUK Technology Co. Ltd

6. 存储器映像

PEC930 使用统一的内存空间物理编址，下面介绍芯片的具体地址分配。

6.1. AHB 地址分配

内部 AHB 地址分配见下表:

AHB 存储器地址	容量	外设
0x0000 0000 – 0x0000 7FFF	32KB	**FLASH 主程序区**
0x0000 8000 – 0x001F FFFF	-	保留
0x0020 0000 – 0x0020 1FFF	8KB	**NVR 信息配置区**
0x0020 2000 – 0x17FF FFFF	-	保留
0x1800 0000 – 0x1800 0FFF	4KB	**CORE 控制与状态区**
0x1800 1000 – 0x1FFF FFFF	-	保留
0x2000 0000 – 0x2000 0FFF	4KB	**SRAM 数据区**
0x2000 1000 – 0x3FFF FFFF	-	保留
0x4000 0000 – 0x4000 FFFF	64KB	**APB 外设模块区**
0x4001 0000 – 0x4001 0FFF	-	保留
0x4001 1000 – 0x4001 1FFF	4KB	**GPIOA**
0x4001 2000 – 0x4001 2FFF	4KB	**GPIOB**
0x4001 3000 – 0x4001 DFFF	-	保留
0x4001 E000 – 0x4001 EFFF	4KB	**CRC**
0x4001 F000 – 0x4001 FFFF	4KB	**SYSCFG**
0x4002 0000 – 0x5FFF FFFF	-	保留
0x6000 0000 – 0x9FFF FFFF	-	保留
0xA000 0000 – 0xDFFF FFFF	-	保留
0xE000 0000 – 0xFFFF FFFF	-	保留

表 6-1 AHB 地址分配表

©Copyright 2026, PADAUK Technology Co. Ltd

6.2. APB 地址映像

内部 APB 地址分配见下表:

序号	APB 存储器地址	容量	外设
0	0x4000 0000 – 0x4000 07FF	2KB	**TIM0**
1	0x4000 0800 – 0x4000 0FFF	2KB	**TIM1**
2	0x4000 1000 – 0x4000 17FF	2KB	**TIM2**
3	0x4000 1800 – 0x4000 1FFF	2KB	保留
4	0x4000 2000 – 0x4000 27FF	2KB	**UART**
5	0x4000 2800 – 0x4000 2FFF	2KB	保留
6	0x4000 3000 – 0x4000 37FF	2KB	**I2C**
7	0x4000 3800 – 0x4000 3FFF	2KB	**SPI**
8	0x4000 4000 – 0x4000 47FF	2KB	**WDG**
9	0x4000 4800 – 0x4000 4FFF	2KB	**ADC** 控制器
10	0x4000 5000 – 0x4000 57FF	2KB	保留
11	0x4000 5800 – 0x4000 5FFF	2KB	**AMISC(包含 PGA0/PGA1)**
12~15	0x4000 6000 – 0x4000 67FF	8KB	保留
16	0x4000 8000 – 0x4000 87FF	2KB	**DSP**
17	0x4000 8800 – 0x4000 8FFF	2KB	**COMP0**
18	0x4000 9000 – 0x4000 97FF	2KB	保留
19	0x4000 9800 –0x4000 9FFF	2KB	**COMP1**
20~23	0x4000 A000 –0x4000 BFFF	8KB	保留
24	0x4000 C000 –0x4000 C7FF	2KB	**EPWM**
25	0x4000 C800 – 0x4000 CFFF	2KB	**LPTIM**
26~30	0x4000 D000 – 0x4000 F7FF	10KB	保留
31	0x4000 F800 – 0x4000 FFFF	2KB	**FLASH** 控制器

表 6-2 APB 地址分配表

6.3. FLASH NVR 配置

FLASH 的 NVR 区域有 16 个 sector，每个 sector 有 128 个 word，模拟校正参数保存在第 14 和 15 个 sector 位置，对应 AHB 基底地址 0x20_1C00 开始。这些区域保存着模拟模块需要的校正参数，这些参数在量产时写入，芯片上电后自动读入配置模拟模块。NVR 的第 0 ~ 13 个 sector 可供用户存放数据使用，对应 AHB 基底地址从 0x20_0000 开始。

©Copyright 2026, PADAUK Technology Co. Ltd

7. 系统配置区 (SYSCFG)

7.1. 特征

PEC930 系统配置区包括了和系统使用有关的自定义寄存器，系统控制寄存器的基地址是 0x4001_F000，用户可以根据应用需求配置系统有关的设置，比如:

- 时钟配置

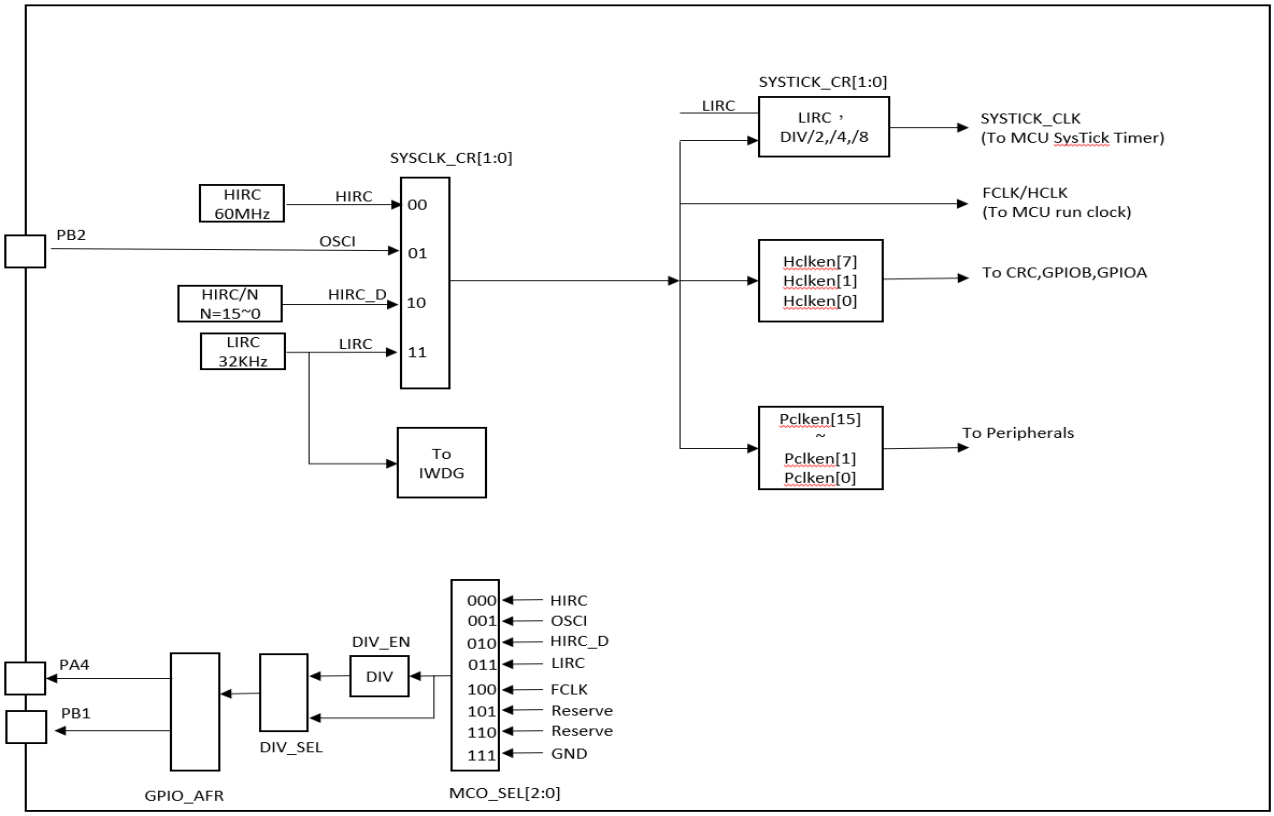


图 7-1 Clock Block Diagram

請參考頻率輸出控制寄存器（SYSCFG_MCOCR），系统时钟配置寄存器（SYSCFG_SYSCLKCR）,外设时钟开关（SYSCFG_PCLKEN），外设时钟 1 开关（SYSCFG_HCLKEN）

- 复位系统配置
  - 請參考复位标识寄存器（SYSCFG_SYSRSTSR）,复位配置寄存器（SYSCFG_SYSRSTCR）,管脚复位使能寄存器（SYSCFG_ICEIOCR）

©Copyright 2026, PADAUK Technology Co. Ltd

- 唤醒配置 (DeepSleep and Sleep)
  - 低功耗有效控制寄存器（SYSCFG_PMUCR）
- TIM2, EPWM 輸入通道配置
  - SYSCFG_TIM2_CON_SEL 寄存器,SYSCFG_EPWM_CON_SEL 寄存器

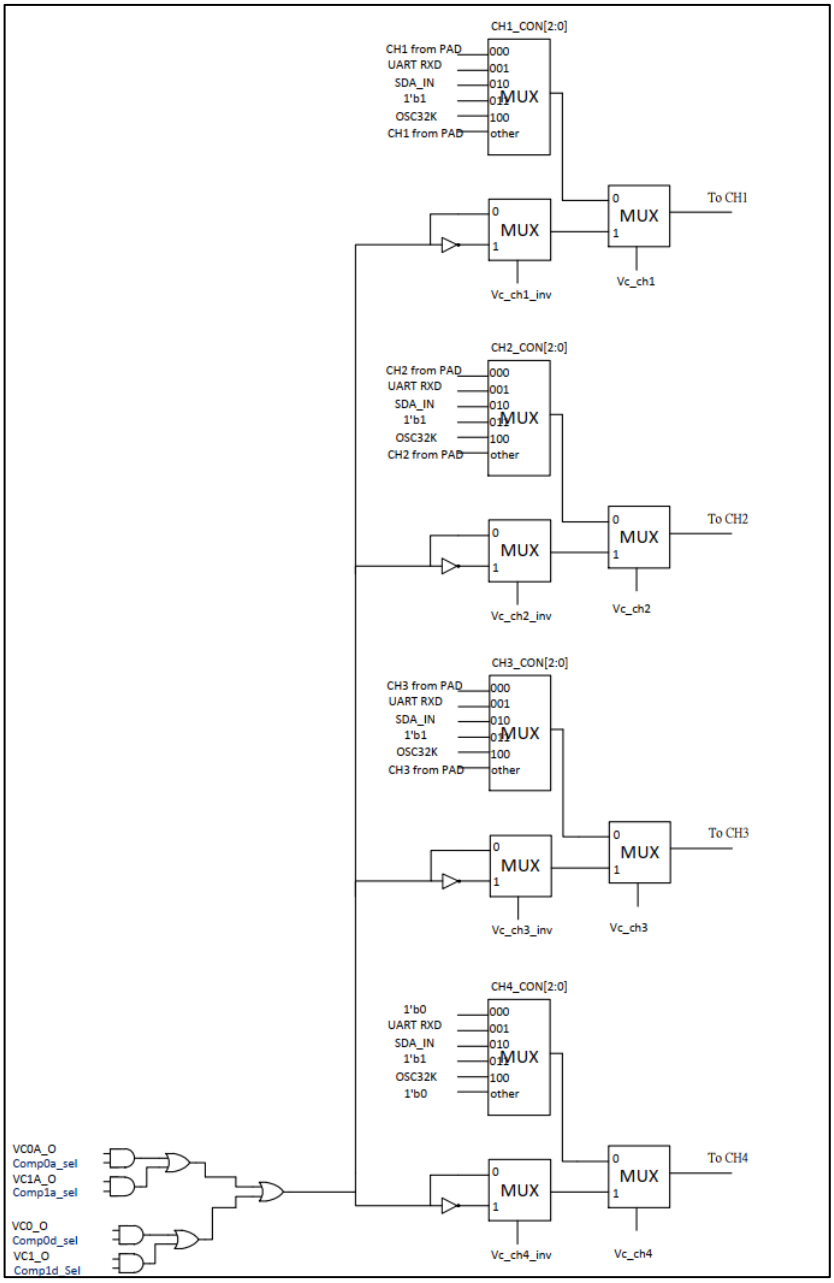


图 7-2 Timer2 and EPWM CONFIG Block Diagram

- EPWM 刹车配置
  - SYSCFG_NMI_BK_SEL 寄存器

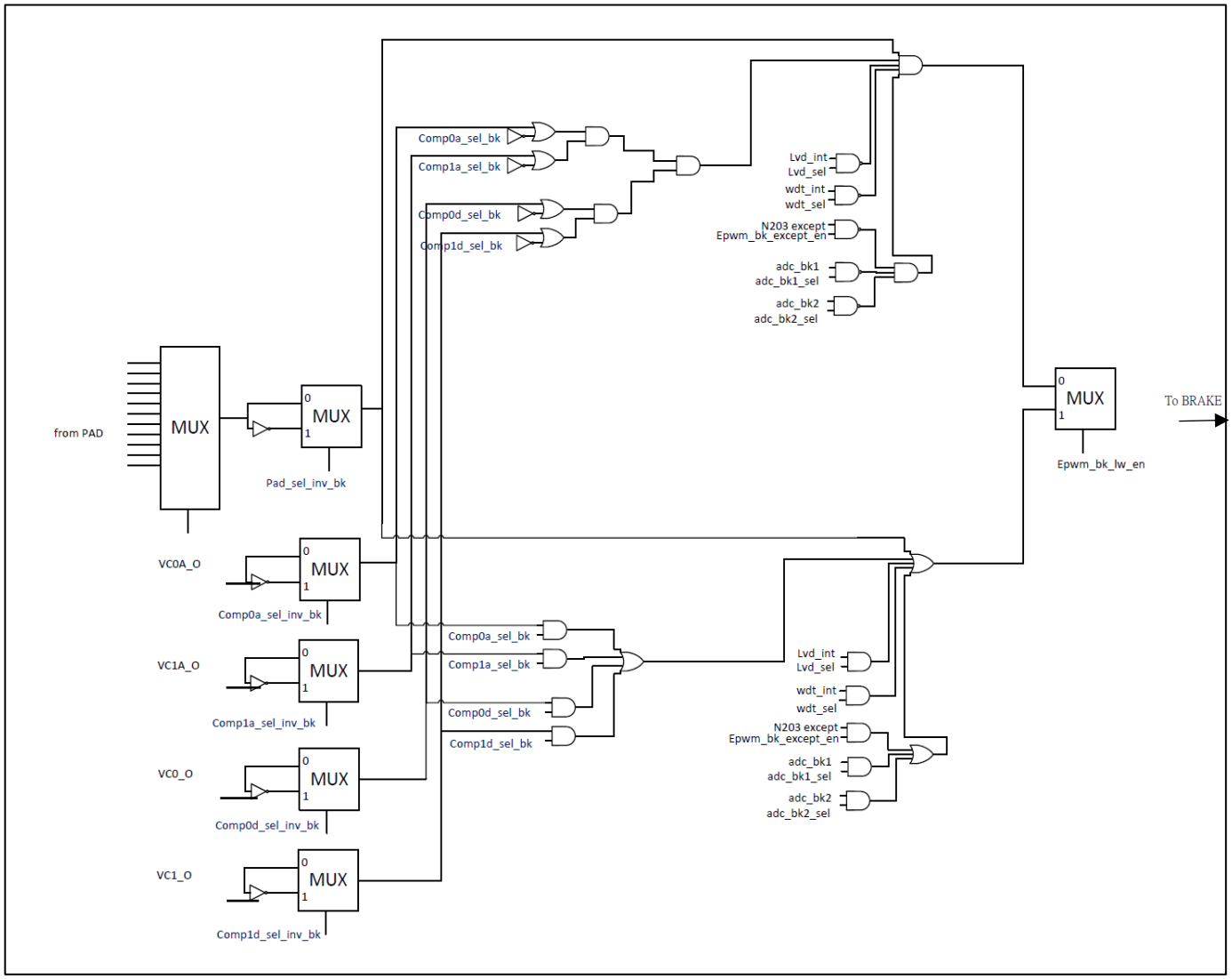


图 7-3 EPWM BRAKE Block Diagram

- Debug 模塊配置
  - Debug 模式控制寄存器（SYSCFG_DEBUGENCR）

7.2. Low Power Mode

PEC930 支持 sleep/deepsleep 省电模式 ，用户可以根据应用需求配置系统有关的设置，流程如下:

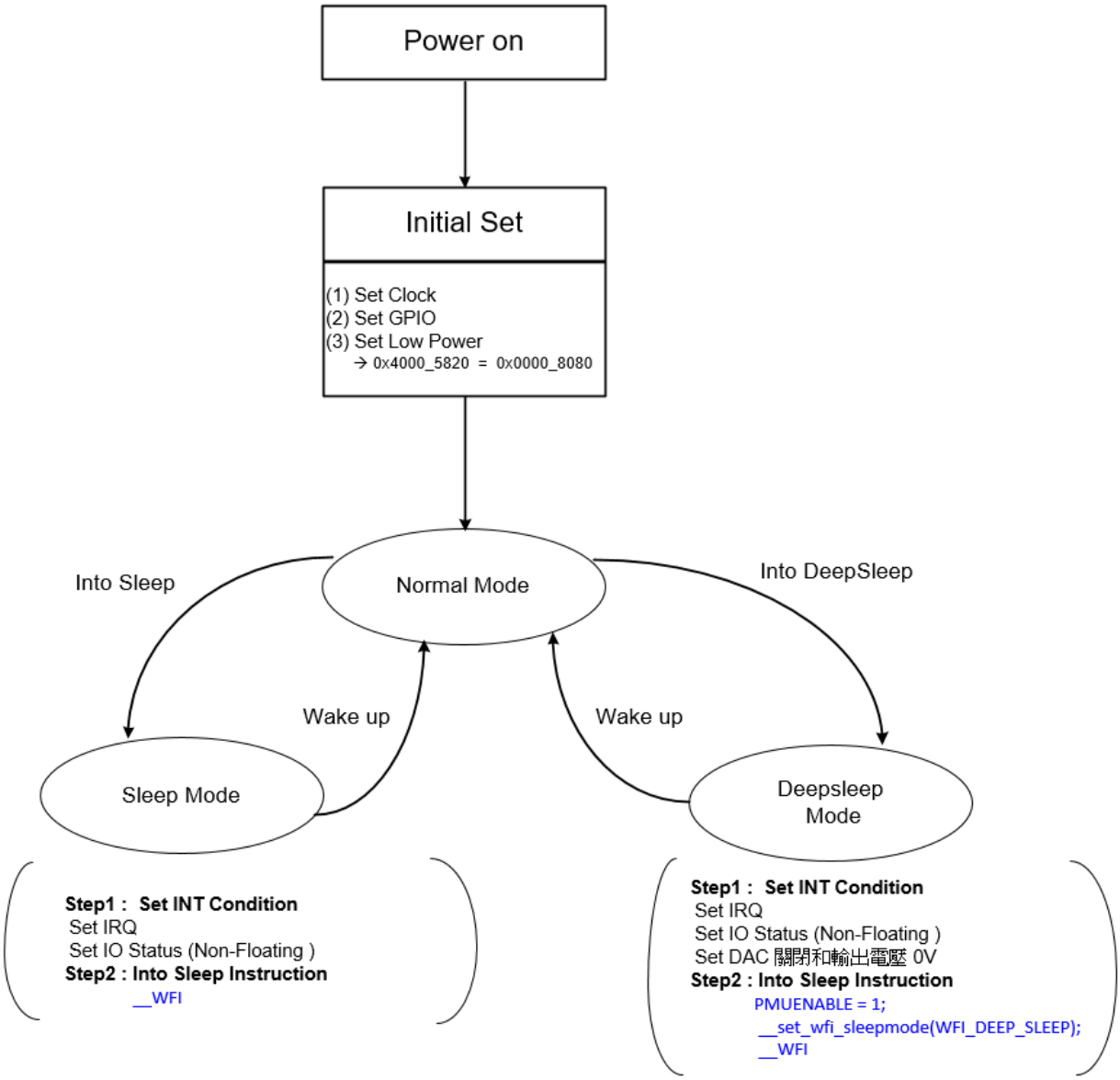


图 7-4 Deepsleep / sleep 設定流程圖

- 設定 IRQ 来源， 参考各自 IP 章节 和相关描述
- Sleep Mode: 参考 9.3 章节，进入 sleep mode
- Deepsleep Mode: 参考 9.3 章节，进入 deepsleep mode 前，需要先设定 0x4001F004 为 1，(需等待 16 個 LIRC 才能真正進入) 参考 7.4.1 章节 wakeup: 参考 2.3 章节

  (a) 唤醒@sleep: 所有中断源都可唤醒

  (b) 唤醒@Deep sleep: 所有外部中断 GPIO PIN ,WDG (低速时钟工作，總是打開) 与功耗唤醒定时器(LPTMR) (低速时钟工作) 参考 9.4 章节，离开 sleep / deepsleep mode。

  (c) Sdk example: 参考 pwr_sleep_mode 範例

©Copyright 2026, PADAUK Technology Co. Ltd

7.3. 寄存器列表

地址	寄存器	描述
0x4001F000	-	保留
0x4001F004	SYSCFG_PMUCR	低功耗有效使能寄存器
0x4001F008	-	保留
0x4001F00C	SYSCFG_MCOCR	MCO 輸出設定寄存器
0x4001F010	SYSCFG_SYSRSTSR	芯片复位状态标识寄存器
0x4001F014	SYSCFG_REBOOT_UNLOCK	RETRIM 密码寄存器
0x4001F018	SYSCFG_SYSRSTCR	芯片复位系统配置寄存器
0x4001F01C	SYSCFG_DEBUGENCR	Debug 模式控制使能寄存器
0x4001F020	SYSCFG_SYSCLKCR	系统时钟配置寄存器
0x4001F024	SYSCFG_PRSTEN	外设复位配置寄存器
0x4001F028	SYSCFG_PCLKEN	外设时钟配置寄存器
0x4001F02C	SYSCFG_ICEIOCR	JTAG 管脚复用使能寄存器
0x4001F030	SYSCFG_RSTPINCR	管脚复位使能寄存器
0x4001F034	SYSCFG_TIM2_CON_SEL	Timer2 Channel 設定寄存器
0x4001F038	SYSCFG_EPWM_CON_SEL	EPWM Channel 設定寄存器
0x4001F03C	-	保留
0x4001F040	SYSCFG_PRSTEN1	外设 1 复位配置寄存器
0x4001F044	SYSCFG_HCLKEN	外设 1 时钟配置寄存器
0x4001F048	SYSCFG_EVT_SEL	EVT 觸發源選擇控制位
0x4001F04C	SYSCFG_NMI_BK_SEL	NMI/BK 觸發源選擇控制位
0x4001F100	SYSCFG_CHIPID	芯片版本号寄存器（只读）

表 7-1 系统配置区寄存器列表

©Copyright 2026, PADAUK Technology Co. Ltd

7.4. 寄存器描述

7.4.1. 低功耗有效控制寄存器 (SYSCFG_PMUCR)

Bit	Name	R/W	Reset	Description
31:1	-	R	0x0	保留
0	PMUENABLE	R/W	0x0	低功耗控制寄存器，配合 SCR 寄存器使用 Example SYSCFG_PMUCR = 1; __set_wfi_sleepmode(WFI_DEEP_SLEEP); __WFI 進入 deepsleep mode

7.4.2. 頻率輸出控制寄存器 (SYSCFG_MCOCR)

Bit	Name	R/W	Reset	Description
31:24	-	-	0x0	保留
23:16	SOFT_KEY	R/W	0x0	寫入 0x5a ,產生 soft reset
15:8	DIV_CNT	R/W	0x0	Frequency divider counter 0: 禁用 1: /2 2~255: /bit[15:8] +1
7	DIV_SEL	R/W	0x0	DIV_SEL 1: 除頻輸出 0: 選定 clock 輸出
6	DIV_EN	R/W	0x0	1: 除頻計數計打開，依據除頻計數值開始計數 0: 除頻計數不計數，並清除頻計數值為 8'h0
5:3	-	-	0x0	保留
2:0	MCO_SEL	R/W	0x0	000: RC OSC 60M 001: Extern clock 010: System frequency division(SYSCFG_SYSCLKCR=3'b010) 011: RC OSC 32K 100: 保留 101: 保留 110: 保留 111: 0

7.4.3. 复位标识寄存器 (SYSCFG_SYSRSTSR)

Bit	Name	R/W	Reset	Description
31:13	-		0x0	保留
12	RES_RB_FG	R/W1c	0x0	1: 执行了 retrim boot 复位 写 1 清零
11	RES_LVD_FG	R/W1c	0x0	1: LVD 电压异常导致复位 写 1 清零
10	-		0x0	保留
9	RES_RESET_FG	R/W1c	0x0	1: RESET 引脚导致复位 写 1 清零
8	RES_POR_FG	R/W1c	0x1	1: POR 导致复位 写 1 清零
7:3	-		0x0	保留
2	RES_LOCK_FG	R/W1c	0x0	1: 复位 LOCKUP 写 1 清零
1	RES_WDG_FG	R/W1c	0x0	1: 看门狗导致复位 写 1 清零
0	RES_SOFT_FG	R/W1c	0x0	1: 系统软复位导致复位 写 1 清零

7.4.4. RETRIM 密码寄存器 (SYSCFG_REBOOT_UNLOCK)

Bit	Name	R/W	Reset	Description
31:16	-		0x0	保留
15:0	RETRIMEN_PWD	R/W	0x0	Reboot 复位密码，当写 RST_CR 寄存器 bit10 触发 Reboot 复位前，需要先将该字段写成 0xAB56，否则不会触发 Reboot 复位

7.4.5. 复位配置寄存器 (SYSCFG_SYSRSTCR)

Bit	Name	R/W	Reset	Description
31:11	-	R	0x0	保留
10	RETRIMING_EN	R/W	0x0	Reboot 使能，高有效，Reboot 使能写 1 会触发系统 Reboot，在复位过程中会重新装载 NVR 中的 trim 值
9:0	-	R	0x0	保留

7.4.6. Debug 模式控制寄存器 (SYSCFG_DEBUGENCR)

Bit	Name	R/W	Reset	Description
31:16	Key	W	0x0	Bit 31~16:只有写 0x8A57 时，才能寫入 Bit 15:0，写其它值时无效
15	-	R	0x0	保留
14	WDG_ICE_EN	R/W	0x1	WDG 模块调试模式停止工作 1: 有效
13:12	-	R	0x0	保留
11	TMR2_ICE_EN	R/W	0x0	TMR2 模块调试模式停止工作 1: 有效
10:7	-	R	0x0	保留
6	EPWM_ICE_EN	R/W	0x0	EPWM 模块调试模式停止工作 1: 有效
5	-	R	0x0	保留
4	LPTMR_ICE_EN	R/W	0x0	LPTMR 模块调试模式停止工作 1:有效
3	TMR1_ICE_EN	R/W	0x0	TMR1 模块调试模式停止工作 1:有效
2	TMR0_ICE_EN	R/W	0x0	TMR0 模块调试模式停止工作 1:有效
1:0	-	R	0x0	保留

©Copyright 2026, PADAUK Technology Co. Ltd

7.4.7. 系统时钟配置寄存器 (SYSCFG_SYSCLKCR)

Bit	Name	R/W	Reset	Description
31:26	-	R	0x0	保留
25:24	SysTick_CR	R/W	0x0	Systick 时钟源选择: 00: 内部 32KHzRC 振荡时钟 01: HCLK/2 10: HCLK/4 11: HCLK/8
23	RC32AON	R/W	0x0	1: RC32K always on (deepsleep mode)
22:12	-	R	0x0	保留
11:8	SYSCLK_DIV	R/W	0x9	Bit11~8：整数分频器分频系数 0000: 表禁用 0001: 表示 2 分频 0010: 表示 3 分频 0011: 表示 4 分频 0100: 表示 5 分频 0101: 表示 6 分频 0110: 表示 7 分频 0111: 表示 8 分频 1000: 表示 9 分频 1001: 表示 10 分频 1010: 表示 11 分频 1011: 表示 12 分频 1100: 表示 13 分频 1101: 表示 14 分频 1110: 表示 15 分频 1111: 表示 16 分频
7:2	-	R	0x0	保留
1:0	SYSCLK_CR	R/W	0x2	系统主时钟选择： 00: 内部 60MHz RC 振荡时钟 01: 外部 PB2 输入时钟 10: 整数分频器输出时钟(分频系数) 11: 内部 32KHzRC 振荡时钟

©Copyright 2026, PADAUK Technology Co. Ltd

7.4.8. 外设复位开关 (SYSCFG_PRSTEN)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15	ANA_RST_EN	R/W	0x0	1: ANA 模块复位开启 0: ANA 模块复位不开启
14	WDG_RST_EN	R/W	0x0	1: WDG 模块复位开启 0: WDG 模块复位不开启
13	COMP1_RST_EN	R/W	0x0	1: COMP1 模块复位开启 0: COMP1 模块复位不开启
12	I2C_RST_EN	R/W	0x0	1: I2C 模块复位开启 0: I2C 模块复位不开启
11	TMR2_RST_EN	R/W	0x0	1: TIMER2 模块复位开启 0: TIMER2 模块复位不开启
10	SPI_RST_EN	R/W	0x0	1: SPI 模块复位开启 0: SPI 模块复位不开启
9	-	R/W	0x0	保留
8	CMP0_RST_EN	R/W	0x0	1: COMP0 模块复位开启 0: COMP0 模块复位不开启
7	DSP_RST_EN	R/W	0x0	1: DSP 模块复位开启 0: DSP 模块复位不开启
6	EPWM_RST_EN	R/W	0x0	1: EPWM 模块复位开启 0: EPWM 模块复位不开启
5	ADC_RST_EN	R/W	0x0	1: ADC 模块复位开启 0: ADC 模块复位不开启
4	LPTMR_RST_EN	R/W	0x0	1: LPTMR 模块复位开启 0: LPTMR 模块复位不开启
3	TMR1_RST_EN	R/W	0x0	1: TIMER1 模块复位开启 0: TIMER1 模块复位不开启
2	TMR0_RST_EN	R/W	0x0	1: TIMER0 模块复位开启 0: TIMER0 模块复位不开启
1	-	R/W	0x0	保留
0	UART_RST_EN	R/W	0x0	1: UART 模块复位开启 0: UART 模块复位不开启

©Copyright 2026, PADAUK Technology Co. Ltd

7.4.9. 外设时钟开关 (SYSCFG_PCLKEN)

Bit	Name	R/W	Reset	Description
31:16		R	0x0	保留
15	ANA_CK_EN	R/W	0x0	1: ANA 模块时钟开启 0: ANA 模块时钟不开启
14	WDG_CK_EN	R/W	0x1	1: WDG 模块时钟开启 0: WDG 模块时钟不开启
13	COMP1_CK_EN	R/W	0x0	1: COMP1 模块时钟开启 0: COMP1 模块时钟不开启
12	I2C_CK_EN	R/W	0x0	1: I2C 模块时钟开启 0: I2C 模块时钟不开启
11	TMR2_CK_EN	R/W	0x0	1: TMR2 模块时钟开启 0: TMR2 模块时钟不开启
10	SPI_CK_EN	R/W	0x0	1: SPI 模块时钟开启 0: SPI 模块时钟不开启
9	-	R/W	0x0	保留
8	CMP0_CK_EN	R/W	0x0	1: CMP0 模块时钟开启 0: CMP0 模块时钟不开启
7	DSP_CK_EN	R/W	0x0	1: DSP 模块时钟开启 0: DSP 模块时钟不开启
6	EPWM_CK_EN	R/W	0x0	1: EPWM 模块时钟开启 0: EPWM 模块时钟不开启
5	ADC_CK_EN	R/W	0x0	1: ADC 模块时钟开启 0: ADC 模块时钟不开启
4	LPTMR_CK_EN	R/W	0x0	1: LPTMR 模块时钟开启 0: LPTMR 模块时钟不开启
3	TMR1_CK_EN	R/W	0x0	1: TMR1 模块时钟开启 0: TMR1 模块时钟不开启
2	TMR0_CK_EN	R/W	0x0	1: TMR0 模块时钟开启 0: TMR0 模块时钟不开启
1	-	R/W	0x0	保留
0	UART_CK_EN	R/W	0x0	1: UART 模块时钟开启 0: UART 模块时钟不开启

©Copyright 2026, PADAUK Technology Co. Ltd

7.4.10. JTAG 管脚复用使能寄存器 (SYSCFG_ICEIOCR)

Bit	Name	R/W	Reset	Description
31:16	KEY	W	0x0	Bit 31~16: 只有写 0xE653 时，才能寫入 Bit 2:0，写其它值时无效
15:3	-	R	0x0	保留
2	DBG_EN	R/W	0x1	CORE debug interface 和内部功能致能 1: Enable 0: Disable
1	DBG_CTRL_EN	R/W	0x1	CORE debug halt 和控制功能致能 1: Enable 0: Disable
0	PIN_JTAG_EN	R/W	0x1	JTAG 管脚复用使能状态 1: Enable 0: Disable

7.4.11. 管脚复位使能寄存器 (SYSCFG_RSTPINCR)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:0	PIN_RST_HOLD	R/W	0x0	PIN 复位端口功能使能 写 0xA563 使能管脚复位；写其它值关闭管脚复位功能； 读该寄存器，bit0 表示管脚复位使能状态，其它 bit 为 0

©Copyright 2026, PADAUK Technology Co. Ltd

7.4.12. TIM2_CON_SEL 寄存器 (SYSCFG_TIM2_CON_SEL)

Bit	Name	R/W	Reset	Description
31:30		R/W	0x0	保留
29:28	Comp1_0d_sel	R/W	0x0	TIM2 输入通道信号来源选择 comp 数位輸出(經過 Filter) 29: Comp1d_sel 28: Comp0d_sel 00: comp1d disable, 0: comp0d disable 01: comp1d disable, 0: comp0d enable 10: comp1d enable, 0: comp0d disable 11: comp1d enable, 0: comp0d enable
27:26	Vc_ch4	R/W	0x0	TIM2_CH4 输入通道信号来源选择 27: Vc_ch4 26: Vc_ch4_inv [27:26] 00: CH4_CON ([14:12]) 設定 01: CH4_CON ([14:12]) 設定 10: COMP 輸入(Comp1_0a_sel 設定) 11: COMP 反向輸入(Comp1_0a_sel 設定)
25:24	Vc_ch3	R/W	0x0	TIM2_CH3 输入通道信号来源选择 25: Vc_ch3 24: Vc_ch3_inv [25:24] 00: CH3_CON ([10:8]) 設定 01: CH3_CON ([10:8]) 設定 10: COMP 輸入(Comp1_0a_sel 設定) 11: COMP 反向輸入(Comp1_0a_sel 設定)
23:22	Vc_ch2	R/W	0x0	TIM2_CH2 输入通道信号来源选择 23: Vc_ch2 22: Vc_ch2_inv [23:22] 00: CH2_CON ([6:4]) 設定 01: CH2_CON ([6:4]) 設定 10: COMP 輸入(Comp1_0a_sel 設定) 11: COMP 反向輸入(Comp1_0a_sel 設定)
21:20	Vc_ch1	R/W	0x0	TIM2_CH1 输入通道信号来源选择

				21: Vc_ch1 20: Vc_ch1_inv [21:20] 00: CH1_CON ([2:0]) 設定 01: CH1_CON ([2:0]) 設定 10: COMP 輸入(Comp1_0a_sel 設定) 11: COMP 反向輸入(Comp1_0a_sel 設定)
19:18	Comp1_0a_sel	R/W	0x0	TIM2 输入通道信号来源选择 comp 類比輸出 19: Comp1a_sel 18: Comp0a_sel 00: comp1a disable, 0: comp0a disable 01: comp1a disable, 0: comp0a enable 10: comp1a enable, 0: comp0a disable 11: comp1a enable, 0: comp0a enable
17:15	-	R/W	0x0	保留
14:12	CH4_CON	R/W	0x0	TIM2_CH4 输入通道信号来源选择 000: 保留 001: uart0_rxd 010: sda_in 011: 保留 100: osc32k 101: 保留 110: 保留 111: 保留
11	-	R/W	0x0	保留
10:8	CH3_CON	R/W	0x0	TIM2_CH3 输入通道信号来源选择 000: from PAD T2CAP3(PA14/PA15/PB0/PB2/PB3/PB4) 001: uart0_rxd 010: sda_in 011: 保留 100: osc32k 101: the same 000 from PAD 110: the same 000 from PAD 111: the same 000 from PAD
7	-	R/W	0x0	保留

6:4	CH2_CON	R/W	0x0	TIM2_CH2 输入通道信号来源选择 000: from PAD T2CAP2(PA14/PA15/PB0/PB2/PB3/PB4) 001: uart0_rxd 010: sda_in 011: 保留 100: osc32k 101: the same 000 from PAD 110: the same 000 from PAD 111: the same 000 from PAD
3	-	R/W	0x0	保留
2:0	CH1_CON	R/W	0x0	TIM2_CH1 输入通道信号来源选择 000: from PAD T2CAP1 (PA14/PA15/PB0/PB2/PB3/PB4) 001: uart0_rxd 010: sda_in 011: 保留 100: osc32k 101: the same 000 from PAD 110: the same 000 from PAD 111: the same 000 from PAD

©Copyright 2026, PADAUK Technology Co. Ltd

7.4.13. EPWM_CON_SEL 寄存器 (SYSCFG_EPWM_CON_SEL)

Bit	Name	R/W	Reset	Description
31	wdg_sel	R/W	0x0	EPWM BK 输入通道信号来源选择 0: wdg bk disable 1: wdg bk enable
30	Lvd_sel	R/W	0x0	EPWM BK 输入通道信号来源选择 0: lvd bk disable 1: lvd bk enable
29:28	Comp1_0d_sel	R/W	0x0	EPWM 输入通道信号来源选择 comp 数位輸出(經過 Filter) 29: Comp1d_sel 28: Comp0d_sel 00: comp1d disable, 0: comp0d disable 01: comp1d disable, 0: comp0d enable 10: comp1d enable, 0: comp0d disable 11: comp1d enable, 0: comp0d enable
27:26	Vc_ch4	R/W	0x0	EPWM_CH4 输入通道信号来源选择 27: Vc_ch4 26: Vc_ch4_inv [27:26] 00: CH4_CON ([14:12]) 設定 01: CH4_CON ([14:12]) 設定 10: COMP 輸入(Comp1_0a_sel 設定) 11: COMP 反向輸入(Comp1_0a_sel 設定)
25:24	Vc_ch3	R/W	0x0	EPWM_CH3 输入通道信号来源选择 25: Vc_ch3 24: Vc_ch3_inv [25:24] 00: CH3_CON ([10:8]) 設定 01: CH3_CON ([10:8]) 設定 10: COMP 輸入(Comp1_0a_sel 設定) 11: COMP 反向輸入(Comp1_0a_sel 設定)
23:22	Vc_ch2	R/W	0x0	EPWM_CH2 输入通道信号来源选择 23: Vc_ch2 22: Vc_ch2_inv [23:22]

©Copyright 2026, PADAUK Technology Co. Ltd

				00: CH2_CON ([6:4]) 設定 01: CH2_CON ([6:4]) 設定 10: COMP 輸入(Comp1_0a_sel 設定) 11: COMP 反向輸入(Comp1_0a_sel 設定)
21:20	Vc_ch1	R/W	0x0	EPWM_CH1 输入通道信号来源选择 21: Vc_ch1 20: Vc_ch1_inv [21:20] 00: CH1_CON ([2:0]) 設定 01: CH1_CON ([2:0]) 設定 10: COMP 輸入(Comp1_0a_sel 設定) 11: COMP 反向輸入(Comp1_0a_sel 設定)
19:18	Comp1_0a_sel	R/W	0x0	EPWM 输入通道信号来源选择 comp 類比輸出 19: Comp1a_sel 18: Comp0a_sel 00: comp1a disable, 0: comp0a disable 01: comp1a disable, 0: comp0a enable 10: comp1a enable, 0: comp0a disable 11: comp1a enable, 0: comp0a enable
17:15	-	R/W	0x0	保留
14:12	CH4_CON	R/W	0x0	EPWM_CH4 输入通道信号来源选择 000: 保留 001: uart0_rxd 010: sda_in 011: 保留 100: osc32k 101: 保留 110: 保留 111: 保留
11	-	R/W	0x0	保留
10:8	CH3_CON	R/W	0x0	EPWM_CH3 输入通道信号来源选择 000: from PAD ECAP3(PA14/PA15/PB0/PB2/PB3/PB4) 001: uart0_rxd 010: sda_in 011: 保留

©Copyright 2026, PADAUK Technology Co. Ltd

				100: osc32k 101: the same 000 from PAD 110: the same 000 from PAD 111: the same 000 from PAD
7	-	R/W	0x0	保留
6:4	CH2_CON	R/W	0x0	EPWM_CH2 输入通道信号来源选择 000: from PAD ECAP2(PA14/PA15/PB0/PB2/PB3/PB4) 001: uart0_rxd 010: sda_in 011: 保留 100: osc32k 101: the same 000 from PAD 110: the same 000 from PAD 111: the same 000 from PAD
3	-	R/W	0x0	保留
2:0	CH1_CON	R/W	0x0	EPWM_CH1 输入通道信号来源选择 000: from PAD ECAP1(PA14/PA15/PB0/PB2/PB3/PB4) 001: uart0_rxd 010: sda_in 011: 保留 100: osc32k 101: the same 000 from PAD 110: the same 000 from PAD 111: the same 000 from PAD

©Copyright 2026, PADAUK Technology Co. Ltd

7.4.14. 外设 1 复位开关 (P1RSTEN_CR)

Bit	Name	R/W	Reset	Description
31:8	-	R	0x0	保留
7	CRC_RES_EN	R/W	0x0	CRC 模块复位开启控制 1: 开启 0: 不开启
6:2	-	R	0x0	Reserved
1	GPIOB_RES_EN	R/W	0x0	GPIO B 模块复位开启控制 1: 开启 0: 不开启
0	GPIOA_RES_EN	R/W	0x0	GPIO A 模块复位开启控制 1: 开启 0: 不开启

7.4.15. 外设 1 时钟开关 (SYSCFG_HCLKEN)

Bit	Name	R/W	Reset	Description
31:8	-	R	保留	保留
7	CRC_CK_EN	R/W	0x0	CRC 模块时钟开启控制 1: 开启 0: 不开启
6:2	-	R	0x0	保留
1	GPIOB_CK_EN	R/W	0x1	GPIO B 模块时钟开启控制 1: 开启 0: 不开启
0	GPIOA_CK_EN	R/W	0x1	GPIO A 模块时钟开启控制 1: 开启 0: 不开启

©Copyright 2026, PADAUK Technology Co. Ltd

7.4.16. EVT_SEL 寄存器 (SYSCFG_EVT_SEL)

Bit	Name	R/W	Reset	Description
31:15	-	R	0x0	保留
14	Evt_LVD_EN	R/W	0x0	evt 來源設定 LVD 1: 設定 0: 關閉
13	Evt_EPWM_EN	R/W	0x0	evt 來源設定 EPWM 1: 設定 0: 關閉
12	Evt_ADC_EN	R/W	0x0	evt 來源設定 ADC 1: 設定 0: 關閉
11	Evt_CMP1_EN	R/W	0x0	evt 來源設定 CMP1 1: 設定 0: 關閉
10	Evt_CMP0_EN	R/W	0x0	evt 來源設定 CMP0 1: 設定 0: 關閉
9	Evt_GPIOB_EN	R/W	0x0	evt 來源設定 GPIOB 1: 設定 0: 關閉
8	Evt_GPIOA_EN	R/W	0x0	evt 來源設定 GPIOA 1: 設定 0: 關閉
7	Evt_I2C_EN	R/W	0x0	evt 來源設定 I2C 1: 設定 0: 關閉
6	Evt_UART_EN	R/W	0x0	evt 來源設定 UART 1: 設定 0: 關閉
5	Evt_SPI_EN	R/W	0x0	evt 來源設定 SPI 1: 設定 0: 關閉

©Copyright 2026, PADAUK Technology Co. Ltd

4	Evt_WDG_EN	R/W	0x0	evt 來源設定 WDG 1: 設定 0: 關閉
3	Evt_LPTMR_EN	R/W	0x0	evt 來源設定 LPTimer 1: 設定 0: 關閉
2	Evt_TMR2_EN	R/W	0x0	evt 來源設定 TMR2 1: 設定 0: 關閉
1	Evt_TMR1_EN	R/W	0x0	evt 來源設定 TMR1 1: 設定 0: 關閉
0	Evt_TMR0_EN	R/W	0x0	evt 來源設定 TMR0 1: 設定 0: 關閉

7.4.17. NMI_BK_SEL 寄存器 (SYSCFG_NMI_BK_SEL)

Bit	Name	R/W	Reset	Description
31:21	-	R/W	0x0	保留
20	Pad_sel_inv_bk	R/W	0x0	EPWM BKP 信号選擇 0: IO 輸入 1: IO 反向輸入
19	Comp1a_sel_inv_bk	R/W	0x0	EPWM BKP 信号選擇 0: comp1 analog 輸入 1: comp1 analog 反向輸入
18	Comp0a_sel_inv_bk	R/W	0x0	EPWM BKP 信号選擇 0: comp0 analog 輸入 1: comp0 analog 反向輸入
17	Comp1d_sel_inv_bk	R/W	0x0	EPWM BKP 信号選擇 0: comp1 digital 輸入 1: comp1 digital 反向輸入
16	Comp0d_sel_inv_bk	R/W	0x0	EPWM BKP 信号選擇

©Copyright 2026, PADAUK Technology Co. Ltd

				0: comp0 digital 輸入 1: comp0 digital 反向輸入
15	Comp1a_sel_bk	R/W	0x0	EPWM BKP 信号選擇 comp1 analog 0: disable 1: enable
14	Comp0a_sel_bk	R/W	0x0	EPWM BKP 信号選擇 comp0 analog 0: disable 1: enable
13	Comp1d_sel_bk	R/W	0x0	EPWM BKP 信号選擇 comp1 digital 0: disable 1: enable
12	Comp0d_sel_bk	R/W	0x0	EPWM BKP 信号選擇 comp0 digital 0: disable 1: enable
11	Epwm_bk_except_EN	R/W	0x0	EPWM BKP 信号選擇 1: enable epem_bk_exception(N203 exception) 0:disable N203 exception list Instruction access fault Load access fault Store/AMO access fault Illegal instruction
10	adc_bk2_sel	R/W	0x0	EPWM BKP 信号選擇 1: adc_bk2 enable 0: adc_bk2 disable
9	adc_bk1_sel	R/W	0x0	EPWM BKP 信号選擇 1: adc_bk1 enable 0: adc_bk1 disable
8	Epwm_bk_low_EN	R/W	0x0	EPWM BKP 信号(Low/High)動作選擇 1: epem_bk_high (平時為0) 0: epem_bk_low (平時為1) **搭配 EPWM (EPWM_BDTR) bit13 BKP 使用**
7:5	NMI_EN	R/W	0x0	NMI 中断使能 (写保护) 3'b101 = 使能 NMI 中断 Other = 禁止 NMI 中断

4:0	NMI_SEL	R/W	0x0	NMI觸發源選擇控制位 通過相對應控制位選擇觸發源的始能 中断源可以从15个外设中断number中选择一个 0x00: timer0 0x01: timer1 0x02: timer2 0x03: lptimer 0x04: wdg 0x05: spi 0x06: uart 0x07: i2c 0x08: gpioa 0x09: apiob 0x0a: cmp0 0x0b: cmp1 0x0c: adc 0x0d: epwm 0x0e: lvd Other: Reserved Referece 3.12 節中断表 (矢量中断控制器)

7.4.18. 芯片版本号寄存器 (SYSCFG_CHIP_ID)

Bit	Name	R/W	Reset	Description
31:0	CHIP_REV_ID	RO	0x00B5_700A	芯片版本控制寄存器（只读） CHIPID [31:8]: 芯片命名，固定为 0x00B570 CHIPID [7:0]: 芯片版本号，例如 0A 代表 A 版，0B 代表 B 版。版本号可以通过金属改版来修改。

8. Flash 控制器 (FMC)

8.1 Flash 特性

片上提供 32KB 的 Flash 存储空间用于存放程序代码。Flash 可以整片一次擦除，也可以按 512 字节为最小单位进行块擦除。通过外部编程工具可将用户代码或数据写入 Flash，程序代码在运行时也可以对 Flash 空间进行自编程。

Flash 数据擦写次数保证>100,000 次，

Flash 基本特性如下:

- Flash 程序区有 32KB 的可用空间
- Flash 快速擦除和编程:

  整片一次性擦除 30-40ms

  512 字节块擦除 2-3ms

  单字编程 5-6.5us
- 100K 擦写次数

8.2 Flash 操作控制

Flash 控制模块封装了对 Flash 程序区进行擦除和编程的接口，使得应用程序无需关注 Flash 区在擦除和数据编程时的物理时序和逻辑控制，只需对相关寄存器进行读写操作，设定 Flash 操作命令、地址和数据，即可实现整 Flash 区的擦除或数据编程工作。

Flash 操作可为:

- 任意逻辑地址处 Flash 数据读取
- 任意逻辑地址处 Flash 数据编程写入，写入前应确保单元数据为 0xFFFFFFFF（擦除状态）
- 以 512 字节为单位进行块擦除
- 整片一次擦除

©Copyright 2026, PADAUK Technology Co. Ltd

8.3 Flash sector erase 流程

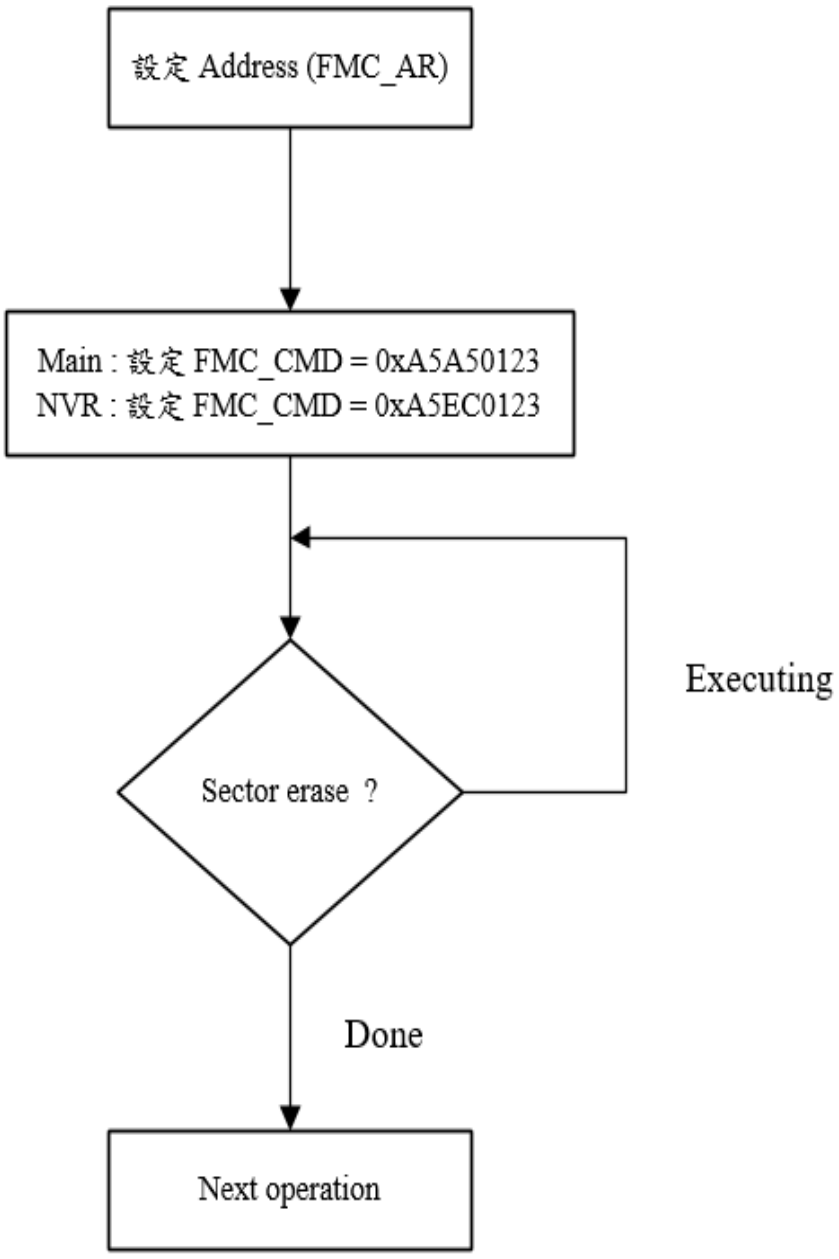


图 8-1 Sector erase 流程圖

8.4 Flash program 流程圖

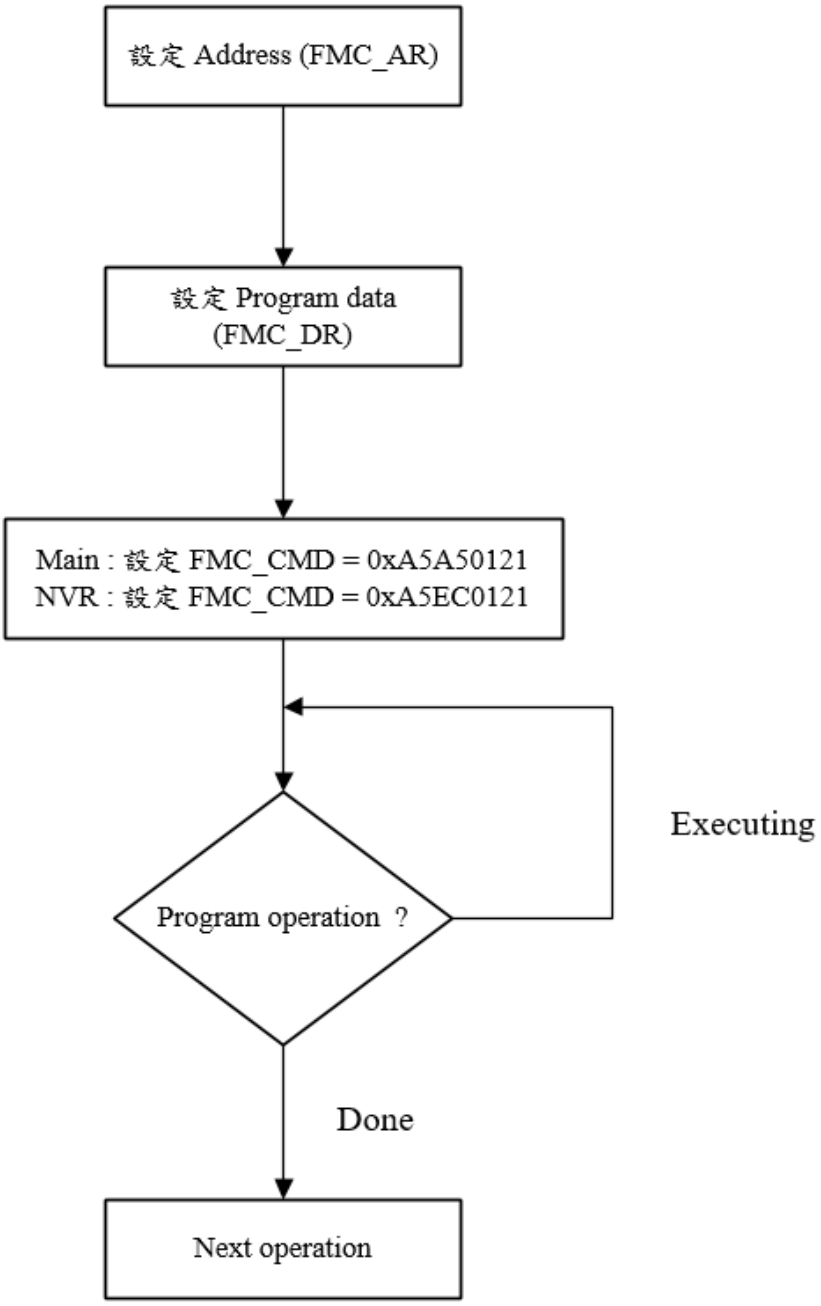


图 8-2 Program 流程圖

8.5 Flash 擦写时钟选择

Flash 擦写时钟为通过 FMC_DIV 寄存器的 DIV 来对系统时钟进行分频产生 1MHz 的擦写时钟。Flash 不支持 0 周期等待默认最小为 1 周期。

8.6 寄存器列表

地址	寄存器	描述
0x4000_F800	FMC_CMD	Flash 命令寄存器
0x4000_F804	FMC_SR	Flash 中断标志寄存器
0x4000_F808	-	保留
0x4000_F80C	FMC_AR	Flash 地址寄存器
0x4000_F810	FMC_DR	Flash 编程数据寄存器
0x4000_F820	FMC_ACM	Flash 访问次数寄存器
0x4000_F824	-	保留
0x4000_F828	FMC_DIV	Flash 擦写时钟分频寄存器

表 8-1 FLASH 寄存器列表

©Copyright 2026, PADAUK Technology Co. Ltd

8.7 寄存器描述

8.7.1. Flash 命令寄存器 (FMC_CMD)

Bit	Name	R/W	Reset	Description
31:16	KEY	W	0x0	寄存器写操作安全密码输入 针对此寄存器低 16 位区域的任何写入操作，必须在此高 16 位区域同步写入操作的安全密码“0xA5A5”，该安全密码对 NVR 区域操作无效，仅对 MAIN CODE 区域操作有效，NVR 区域操作安全密码为 “ 0xA5EC”。
15:11	-	R	0x0	保留
10	ACMR	R/W	0x0	Flash 存取计数器复位 0: Flash 存取计数器正常计数 1: Flash 存取计数器复位清零
9	ACME	R/W	0x0	Flash 存取计数器使能 0: Flash 存取计数器禁止 1: Flash 存取计数器使能
8	UNLOCK	R/W	0x0	LOCK 0: Flash 無法 program / erase。 1: Flash 可以 program /erase。
7:5	NWS	R/W	0x1	Flash 操作等待周期 设定 Flash 命令操作时的等待周期(不支持 0 周期等待) 000: 不支援 0 週期等待 001: 1 周期等待 010: 2 周期等待 011: 3 周期等待 100: 4 周期等待 101: 5 周期等待 110: 6 周期等待 111: 7 周期等待
4	-	R	0x0	保留
3:1	CMD	R/W	0x0	Flash 操作命令 000: 单字（32 位）写入编程 001: 块擦除 010: 整片擦除 011: 保留 1xx: 保留
0	START	R/W	0x0	Flash 操作启动 写操作 0: 无效 1: 启动 Flash 操作命令 读操作 0: Flash 操作命令执行完毕 1: Flash 操作命令执行中（执行完毕后硬件自动清 0）

8.7.2. Flash 中断状态寄存器 (FMC_SR)

Bit	Name	R/W	Reset	Description
31:15	-	R	0x0	保留
14	IOSC60M_TRIM_ERR	R	0x0	0: 寫入的 iosc60m trim 值正確。 1: 寫入的 iosc60m trim 值有誤。
13	IOSC32K_TRIM_ERR	R	0x0	0: 寫入的 iosc32k trim 值正確。 1: 寫入的 iosc32k trim 值有誤。
12	VBUF_TRIM_ERR	R	0x0	0: 寫入的 vbuf trim 值正確。 1: 寫入的 vbuf trim 值有誤。
11	LDO_TRIM_ERR	R	0x0	0: 寫入的 ldo trim 值正確。 1: 寫入的 ldo trim 值有誤。
10	EXT_RSTN_TRIM_ERR	R	0x0	0: 寫入的 ext_rstn trim 值正確。 1: 寫入的 ext_rstn trim 值有誤。
9	PROTECT_R1_TRIM_ERR	R	0x0	0: 寫入的 protect_r1 trim 值正確。 1: 寫入的 protect_r1 trim 值有誤。
8	PROTECT_R2_TRIM_ERR	R	0x0	0: 寫入的 protect_r2 trim 值正確。 1: 寫入的 protect_r2 trim 值有誤。
7	OSCHF_TC_TRIM_ERR	R	0x0	0: 寫入的 oschf_tc trim 值正確。 1: 寫入的 oschf_tc trim 值有誤。
6	HCM	R/W1c	0x0	Flash 访问总次数计数饱和 0: FLACM 计数器尚未饱和（＜0xFF00_0000） 1: FLACM 计数器尚已饱和（≥0xFF00_0000） 软件对该位写 1 将其清 0
5:4	-	R	0x0	保留
3	ADDR_ERR	R/W1c	0x0	Flash 地址错误标志位 0: 地址正确 1: 地址错误（Flash 操作地址超限），软件对该位写 1 将其清 0
2	-	-	0x0	保留
1	KEY_ERR	R/W1c	0x0	KEY 設定错误标志位 1: Main or NVR 執行 erase / program，*KEY* 設定錯誤，软件对该位写 1 将其清 0
0	CMD_END	R/W1c	0x0	Flash 操作命令完成标志位 0: 命令未完成 1: 命令完成，软件对该位写 1 将其清 0

注: bit [14:7] (trim_err_flag) 若有為 1，請設定 SYSCFG_REBOOT_UNLOCK 和 SYSCFG_SYSRSTCR，重新 latch trim 值。

8.7.3. Flash 数据区地址寄存器 (FMC_AR)

Bit	Name	R/W	Reset	Description
31:22	-	R	0x0	保留
21	NVR	R/W	0x0	Flash 数据区操作区域选择 Flash 数据区还有信息配置区，此位决定 Flash 操作针对的具体区域. 0：针对主代码空间进行寻址操作 1：针对信息配置区空间进行寻址操作
20:15	-	R	0x0	保留
14:2	AR	R/W	0x0	Flash 数据区字寻址逻辑地址 总计 32KB 的 Flash 数据区主代码空间等价于 8K 的逻辑字空间（每字 4 字节），每个逻辑地址处对应将寻址 4 字节的 Flash 数据
1:0	-	R	0x0	保留

8.7.4. Flash 编程数据寄存器 (FMC_DR)

Bit	Name	R/W	Reset	Description
31:0	DR	R/W	0x0	Flash 编程数据 内容为 0 时的数据位将被写入 Flash，1 的数据位在编程时被忽略．编程前确保 Flash 单元中的数据已被擦除为 0xFFFFFFFF.

8.7.5. Flash 程序区访问次数寄存器 (FMC_ACM)

Bit	Name	R/W	Reset	Description
31:0	ACM	R	0x0	Flash 程序区访问计数 只要 MCU 内核执行了一次面向 Flash 地址区域的读取访问，计数器即加 1。 ACM 在递增到大于等于 0xFF00_0000 时会置 FLISR 寄存器的 HCM 标志位为 1。ACM 为只读，任何时候都可通过将 FMC_CMD 寄存器的 ACMR 置 1 将 ACM 清 0

©Copyright 2026, PADAUK Technology Co. Ltd

8.7.6. Flash 擦写时钟分频寄存器 (FMC_DIV)

Bit	Name	R/W	Reset	Description
31:9	-	R	0x1	保留
8	CP_DONE	W	0x1	需保持为 1
7:0	DIV	R/W	0x18	Flash 擦写时钟分频系数 DIV 用于对系统时钟进行分频，产生 1MHz 的 Flash 擦写时钟。 DIV 的配置公式为 $freq_{TICK} = \frac{freq_{SYS}}{FLDIV}$ 注: $freq_{TICK}$ 为分频后的 1MHz， $freq_{SYS}$ 为系统时钟频率。

9. 矢量中断控制器

PEC930 RISC-V 集成了改进型内核中断控制器 (ECLIC) 来实现高效的异常和中断处理。ECLIC 旨在为 RISC-V系统提供低延迟、矢量化、抢占式的中断。当 ECLIC 被激活后，它将包含并替换现有的 RISC-V 处理器局部中断控制器。CLIC 设计有一个基本设计，需要最少的硬件，但它支持额外的扩展来提供硬件加速。ECLIC 设计的目标是为各种软件 ABI 和中断模型提供支持，而不需要复杂的硬件影响高性能处理器的实现。它与处理器核心紧密耦合。可以阅读 RISC-V 的技术参考手册，了解有关 ECLIC 的更多详细信息。

9.1 特征

PEC930 中断控制器有以下特征:

- 实际支持硬件中断数量 15 个。
- 3位中断优先级配置位—— 8 个中断优先等级。
- 支持异常抢占和咬尾中断。
- 将系统从省电模式唤醒。

9.2 中断功能说明

RISC-V 处理器和改进型内核中断控制器 (ECLIC) 在机器 (machine) 模式下对所有异常进行优先级区分以及处理。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。下表列出了所有的中断类型。

©Copyright 2026, PADAUK Technology Co. Ltd

	IRQ Number	Exception or Interrupt	Priority	Program Addr Vector	Type
	0	保留	（最低）	Value of MTVT +4*0	保留
	1	保留		Value of MTVT +4*1	保留
	2	保留		Value of MTVT +4*2	保留
	3	Machine Software interrupt		Value of MTVT +4*3	电平正
	4	保留		Value of MTVT +4*4	保留
	5	保留		Value of MTVT +4*5	保留
	6	保留		Value of MTVT +4*6	保留
	7	Machine Timer interrupt		Value of MTVT +4*7	电平正
	8	保留		Value of MTVT +4*8	保留
	9	保留		Value of MTVT +4*9	保留
	10	保留		Value of MTVT +4*10	保留
	11	保留		Value of MTVT +4*11	保留
	12	保留		Value of MTVT +4*12	保留
	13	保留		Value of MTVT +4*13	保留
	14	保留		Value of MTVT +4*14	保留
	15	保留		Value of MTVT +4*15	保留
	16	保留		Value of MTVT +4*16	保留
	17	保留		Value of MTVT +4*17	保留
	18	保留		Value of MTVT +4*18	保留
1	19	Timer0		Value of MTVT +4*19	电平正
2	20	Timer1		Value of MTVT +4*20	电平正
3	21	Timer2		Value of MTVT +4*21	电平正
4	22	LPTIMER		Value of MTVT +4*22	电平正
5	23	WDG		Value of MTVT +4*23	电平正
6	24	SPI		Value of MTVT +4*24	电平正
7	25	UART		Value of MTVT +4*25	电平正
8	26	I2C-		Value of MTVT +4*26	电平正
9	27	GPIOA		Value of MTVT +4*27	电平正
10	28	GPIOB		Value of MTVT +4*28	电平正
11	29	CMP0		Value of MTVT +4*29	电平正
12	30	CMP1		Value of MTVT +4*30	电平正
13	31	ADC		Value of MTVT +4*31	电平正
14	32	EPWM		Value of MTVT +4*32	电平正
15	33	LVD	（最高）	Value of MTVT +4*33	电平/正 or 负缘

表 9-1 中断向量表

注: Reference 3.12

9.3 进入休眠状态

内核可以通过 WFI 指令进入休眠状态。当处理器执行到 WFI 指令之后，将会:

- 立即停止执行当前的指令流；
- 等待处理器内核完成任何尚未完成的滞外操作（Outstanding Transactions），譬如取指令和数据读写操作，以保证发到总线上的操作都完成；
  - 注意：如果在等待总线上的操作完成的过程中发生了存储器访问错误异常，则会进入到异常处理模式，而不会休眠。

当所有的滞外操作（Outstanding Transactions）都完成后，处理器会安全地进入一种空闲状态，这种空闲状态可以被称之为“休眠”状态。

当进入休眠模式后:

- 内核内部的各个主要单元的时钟将会被门控关闭以节省静态功耗;
- 内核的输出信号 core_wfi_mode 会拉高，指示此处理器核处于执行 WFI 指令之后的休眠状态；
- 内核的输出信号 core_sleep_value 会输出 CSR 寄存器 sleepvalue 的值

（注意: 该信号只有在 core_wfi_mode 信号为高电平时生效；core_wfi_mode 信号为低电平时 core_sleep_value 的值一定是 0）。软件可以通过事先配置 CSR 寄存器 sleepvalue 来指示不同的休眠模式（0 或者 1）

注: 当进入深度睡眠模式时，处理器核心将无法再透过 JTAG 接口进行调试

9.4 退出休眠状态

内核处理器退出休眠模式的要点如下:

- 内核的输出信号 core_wfi_mode 会相应拉低。
- 内核处理器可以通过以下几种方式被唤醒:

1. 中断
2. Event

9.4.1 中断唤醒

- 如果 mstatus.MIE 域被配置为 1（表示全局中断被打开），则:
  - 当 ECLIC（通过将外部请求的中断进行仲裁）向处理器内核发送了中断，处理器内核被唤醒，进入到中断服务程序开始执行
- 如果 mstatus.MIE 域被配置为 0（表示全局中断被关闭），则:
  - 如果 CSR 寄存器 wfe.WFE 域被配置为 0，则:

当 ECLIC（通过将外部请求的中断进行仲裁）向处理器内核发送了中断，处理器内核被唤醒，继续顺序执行之前停止的指令流（而不是进入到中断服务程序）。

  - 如果 CSR 寄存器 wfe.WFE 域被配置为 1，则等待 Event 唤醒

9.4.2 Event 唤醒

当满足如下条件时，Event 可以唤醒处理器内核:

- mstatus.MIE 域被配置为 0（表示全局中断被关闭），且 **CSR 寄存器 wfe.WFE 域被配置为 1**
- 当处理器内核检测到输入信号 rx_evt（称之为 Event 信号）为高电平时，处理器内核被唤醒，继续执行之前停止的指令流（而不是进入到中断服务程序）

9.5 Wait for Interrupt 机制

Wait for Interrupt 机制，是指将处理器内核进入休眠模式，然后等待中断唤醒处理器内核，醒来后进入相应中断的处理函数中去。

如第 9.3 节和第 9.4 节所述，Wait for Interrupt 机制可以直接通过 WFI 指令（配合 mstatus.MIE 域被配置为 1）完成。

9.6 Wait for Event 机制

Wait for Event 机制，是指将处理器内核进入休眠模式，然后等待 Event 唤醒处理器内核，醒来后继续先前停止的程序（而不是进入中断的处理函数中去）。

如第 9.3 节和第 9.4 节所述，Wait for Event 机制可以直接通过 WFI 指令，配合如下指令序列完成:

第 1 步: 配置 wfe.WFE 域为 1

第 2 步: 调用 WFI 指令。调用此指令后处理器会进入休眠模式，当 Event 将其唤醒后将会继续向下执行。

第 3 步: 恢复 wfe.WFE 域为 0

NMI 触发源包括来自 I/O 管脚的 22 根线 (SOP24)以及来自内部模块的 2 根线。(包括WDG闹钟、LPTMR唤醒)。通过配置 GPIO 模块的 *GPIOn_IES*, *GPIOn_ITS0* 和 *GPIOn_ITS1* 寄存器，所有的 GPIO 管脚都可以被选作 NMI 的触发源，具体细节请参考GPIO和端口输入中断章节。

RISC-V 内核完全支持等待中断 (WFI)，等待事件 (WFE) 和发送事件 (SEV) 指令。当某些预期的事件发生时，例如一个特定的 I/O 管脚电平翻转或者 WDG/LPTMR 动作，NMI 能唤醒处理器及整个系统。

9.7 相關寄存器列表

PEC930 的中断系统提供的寄存器，各个寄存器的地址空间如下所示:

地址	寄存器	描述
0x4001_F048	EVT_SEL	EVT 觸發源選擇控制位
0x4001_F04C	NMI_BK_SEL	NMI/BK 觸發源選擇控制位

表 9-2 中斷控制器寄存器列表

注:

Reference 7.3.16

Reference 7.3.17

10. GPIO

PEC930 有 2 组 GPIO，总共有 22 个 GPIO 口，包括 PORT A 和 PORT B。其中:

- PORT A: PA_0 ~ PA_15
- PORT B: PB_0 ~ PB_5

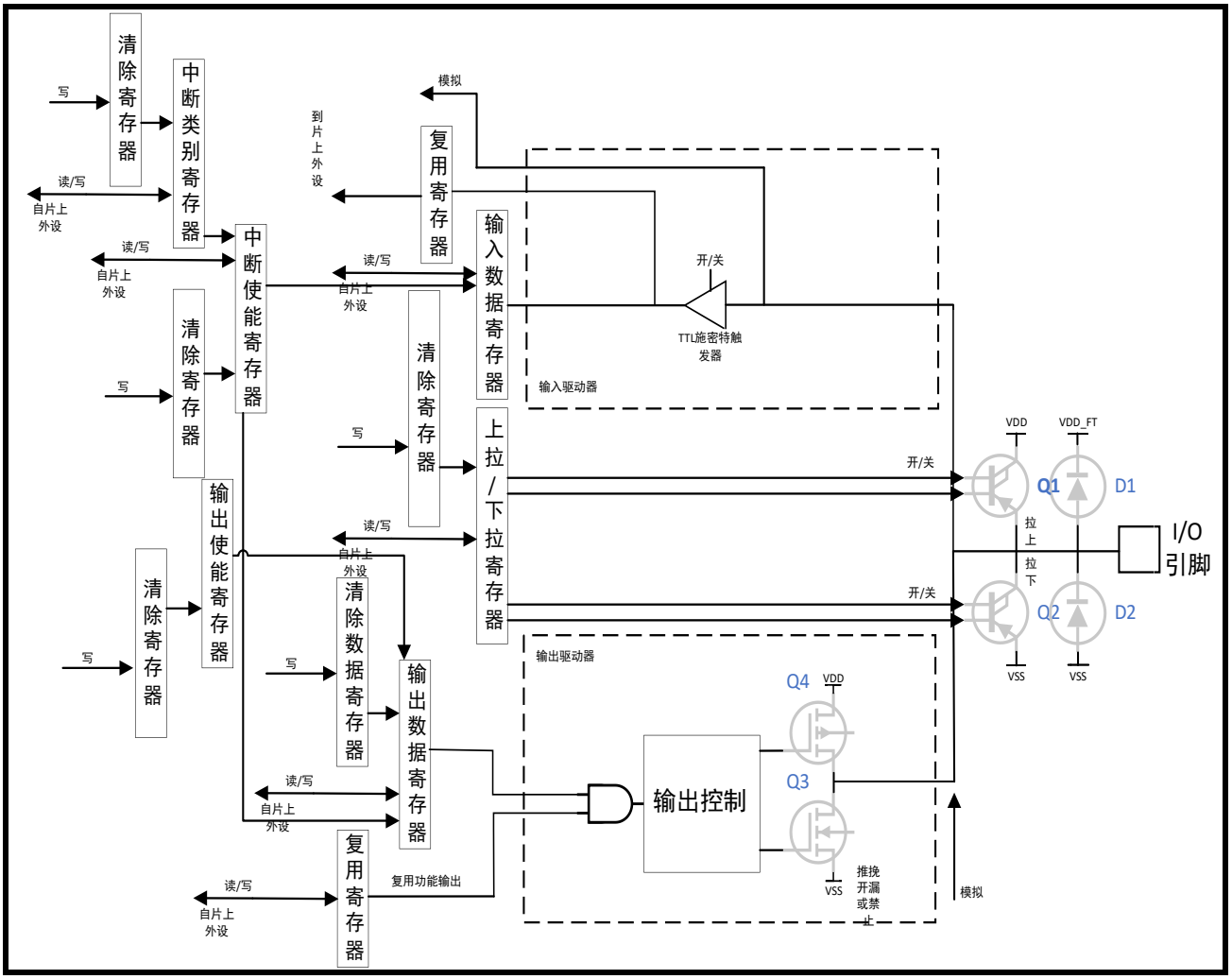


图 10-1 GPIO Block Diagram

10.1 特征

PEC930 的 IO 口配置为 GPIO 时，具有如下特征:

- GPIO 口均可配置为输入或输出。
- 所有 GPIO 都可作为外部 IO 中断。
- GPIO 口在输出模式下上下拉自动禁止
- GPIO 在输入模式下引脚呈高阻抗。
- GPIO 在输入模式下可选用内部上拉或下拉。
- 每个 IO 都可以配置为开漏输出。
- 通过位掩模提供 GPIO 位操作。
- 通过分开的置位和清除操作控制寄存器，提供线程安全的端口操作。
- 通过外设复用功能寄存器配置 IO 口为 GPIO 或者其他功能（如外设功能）。
- 提供端口映射操作，可以只对端口某一位、或某几位进行操作。

10.2 输入输出方向控制

每个 GPIO 都支持输入或输出。输入/输出方向控制由 GPIOn_OES，GPIOn_OEC 寄存器控制，GPIOn_OES 寄存器可以使能 IO 口输出，GPIOn_OEC 寄存器可以清除 IO 的输出使能，设置相应 IO 口为输入。具体控制方法见寄存器说明。

©Copyright 2026, PADAUK Technology Co. Ltd

10.3 端口输入中断

每个 GPIO 都可通过四对寄存器进行配置，包括 GPIOn_IES, GPIOn_IEC, GPIO_nITS0, GPIO_nITC0，GPIO_nITS1, GPIO_nITC1 和 GPIOn_PLS, GPIOn_PLC。其中 GPIOn_IES, GPIOn_IEC 控制端口中断的使能和禁止；GPIOn_ITS0, GPIOn_ITC0，GPIOn_ITS1, GPIOn_ITC1 控制端口中断的类型；GPIOn_PLS, GPIOn_PLC 控制端口中断的极性。

端口中断使能位	端口中断极性位	端口中断类型位 0	端口中断类型位 1	中断响应模式
0	-	-	-	中断禁止
1	0	0	0	低电平
1	0	1	0	下降沿
1	1	0	0	高电平
1	1	1	0	上升沿
1	X	X	1	电平变化

表 10-1 GPIO 中断响应模式列表

引脚中断被触发后，状态寄存器 GPIOn_IST 中的相应位被置 1。所有引脚的中断请求都由同一个中断服务程序得到响应，用户软件在中断服务程序中需要查询 GPIOn_IST 和引脚状态确定具体的引脚中断源，并对 GPIOn_IST 的对应位写 1 来清除中断标志。

10.4 端口输入内部上拉或下拉

当引脚被配置成端口输入模式时，每个引脚都可以独立控制启用片内的弱上拉或下拉电阻（等效阻值约 10K）。在引脚设为端口输出模式时，该上拉或下拉将被自动禁止。GPIOn_PUS 可以使能端口上拉，GPIOn_PUC 可以清除端口上拉，关闭上拉；GPIOn_PDS 可以使能端口下拉，GPIOn_PDC 可以清除端口下拉，关闭下拉。

10.5 端口输出驱动能力

在 5V 供电条件下，所有端口都支持 40mA 拉出的电流驱动能力。

10.6 端口开漏输出功能

任一端口引脚在输出时均可设为开漏模式。此模式下芯片在低电平 0 输出时为低电平驱动；在高电平 1 输出时，芯片引脚为“漏极开路”状态，必须在片外通过电阻上拉到一个特定的电平电压后才能在引脚上得到一个真实的电压值。外部上拉电压不能超过芯片 Vdd+0.6V。通过 GPIOn_ODS, GPIOn_ODC 寄存器可以设置使能或清除 IO 口开漏功能。

10.7 端口施密特功能配置功能

CMOS 特性的端口引脚可以配置使能施密特功能。通过 GPIOn_CTS, GPIOnCTC 寄存器可以对相应 IO 口使能或取消其施密特功能。

©Copyright 2026, PADAUK Technology Co. Ltd

10.8 端口复用功能

IO 口通常默认功能为 GPIO（Debug 口除外以及 PB1 默认 RST）。通过对寄存器 GPIOPn_AFR, FN1_AFR, FN2_AFR 寄存器操作，可以开启或关闭 IO 口的复用功能，如 UART 口等等。

10.9 注意事項

任何情況下須切換端口引脚模式，請先將相關的始能寄存器設定為 0，待寄存器設定完成後，再將使能設定為 1。

©Copyright 2026, PADAUK Technology Co. Ltd

10.10 寄存器列表

地址	寄存器	描述
0x4001_0000 ~ 0x4001_08FC	-	保留
0x4001_1000	GPIOA_DAT	PORTA 端口数据寄存器
0x4001_1004	GPIOA_LAT	PORTA 端口数据输出锁存寄存器
0x4001_1008	GPIOA_ITS1	PORTA 中断类型置位寄存器 1
0x4001_100C	GPIOA_ITC1	PORTA 中断类型清除寄存器 1
0x4001_1010	GPIOA_OES	PORTA 端口输出使能置位寄存器
0x4001_1014	GPIOA_OEC	PORTA 端口输出使能清除寄存器
0x4001_1018	GPIOA_INES	PORTA 端口输入使能置位寄存器
0x4001_101C	GPIOA_INEC	PORTA 端口输入使能清除寄存器
0x4001_1020	GPIOA_IES	PORTA 中断使能置位寄存器
0x4001_1024	GPIOA_IEC	PORTA 中断使能清除寄存器
0x4001_1028	GPIOA_ITS0	PORTA 中断类型置位寄存器 0
0x4001_102C	GPIOA_ITC0	PORTA 中断类型清除寄存器 0
0x4001_1030	GPIOA_PLS	PORTA 中断极性设置寄存器
0x4001_1034	GPIOA_PLC	PORTA 中断极性清除寄存器
0x4001_1038	GPIOA_IST	PORTA 中断标志寄存器
0x4001_103C	GPIOA_PUS	PORTA 内部上拉使能置位寄存器
0x4001_1040	GPIOA_PUC	PORTA 内部上拉使能清除寄存器
0x4001_1044	GPIOA_ODS	PORTA 输出开漏使能置位寄存器
0x4001_1048	GPIOA_ODC	PORTA 输出开漏使能清除寄存器
0x4001_104C	GPIOA_PDS	PORTA 内部下拉使能置位寄存器
0x4001_1050	GPIOA_PDC	PORTA 内部下拉使能清除寄存器
0x4001_1054	GPIOA_PODS	PORTA 输出 PMOS 开漏使能置位寄存器
0x4001_1058	GPIOA_PODC	PORTA 输出 PMOS 开漏使能清除寄存器
0x4001_105C	GPIOA_STE	PORTA 端口施密特功能使能
0x4001_1060	GPIOA_STD	PORTA 端口施密特功能清除
0x4001_1070	GPIOA_AFR	PORTA 外设复用置位寄存器
0x4001_1074	GPIOB_AFR	PORTB 外设复用置位寄存器
0x4001_1078	FN1_AFR	引脚外设复用功能 1 配置寄存器
0x4001_107C	FN2_AFR	引脚外设复用功能 2 配置寄存器

©Copyright 2026, PADAUK Technology Co. Ltd

0x4001_11F8 ~ 0x4001_1BFC	-	保留
0x4001_2000	GPIOB_DAT	PORTB 端口数据寄存器
0x4001_2004	GPIOB_LAT	PORTB 端口数据输出锁存寄存器
0x4001_2008	GPIOB_ITS1	PORTB 中断类型置位寄存器 1
0x4001_200C	GPIOB_ITC1	PORTB 中断类型清除寄存器 1
0x4001_2010	GPIOB_OES	PORTB 端口输出使能置位寄存器
0x4001_2014	GPIOB_OEC	PORTB 端口输出使能清除寄存器
0x4001_2018	GPIOB_INES	PORTB 端口输入使能置位寄存器
0x4001_201C	GPIOB_INEC	PORTB 端口输入使能清除寄存器
0x4001_2020	GPIOB_IES	PORTB 中断使能设置寄存器
0x4001_2024	GPIOB_IEC	PORTB 中断使能清除寄存器
0x4001_2028	GPIOB_ITS0	PORTB 中断类型置位寄存器 0
0x4001_202C	GPIOB_ITC0	PORTB 中断类型清除寄存器 0
0x4001_2030	GPIOB_PLS	PORTB 中断极性设置寄存器
0x4001_2034	GPIOB_PLC	PORTB 中断极性清除寄存器
0x4001_2038	GPIOB_IST	PORTB 中断标志寄存器
0x4001_203C	GPIOB_PUS	PORTB 内部上拉使能置位寄存器
0x4001_2040	GPIOB_PUC	PORTB 内部上拉使能清除寄存器
0x4001_2044	GPIOB_ODS	PORTB 输出开漏使能置位寄存器
0x4001_2048	GPIOB_ODC	PORTB 输出开漏使能清除寄存器
0x4001_204C	GPIOB_PDS	PORTB 内部下拉使能置位寄存器
0x4001_2050	GPIOB_PDC	PORTB 内部下拉使能清除寄存器
0x4001_2054	GPIOB_PODS	PORTB 输出 PMOS 开漏使能置位寄存器
0x4001_2058	GPIOB_PODC	PORTB 输出 PMOS 开漏使能清除寄存器
0x4001_205C	GPIOB_STE	PORTB 端口施密特功能置位
0x4001_2060	GPIOB_STD	PORTB 端口施密特功能清除
0x4001_2064 ~ 0x4001_20FC	-	保留

表 10-2 GPIO 寄存器列表

10.11 寄存器描述

10.11.1 端口数据寄存器 (GPIOn_DAT) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_DAT	R/W	0x0	PORT A 位 [15:0]: PORT A 引脚端口数据位 PORT B 位 [5:0]: PORT B 引脚端口数据位 读寄存器时: 0: 端口引脚为逻辑低电平 1: 端口引脚为逻辑高电平 写寄存器且对应引脚输出使能时: 0: 端口引脚输出逻辑低电平 1: 端口引脚输出逻辑高电平

*1: 当此位为保留位，不使用时，R/W 属性为 R，只读，读出值为 0；当此位已实现，R/W 属性为 R/W，可读可写。

GPIOn_DAT 位端口数据寄存器，分别针对 PORTA, PORTB 端口模块。任何时候读 GPIOn_DAT 时，读回的是端口引脚的实际电平逻辑状态; 任何时候写 GPIOn_DAT 时，实际写入的是端口的数据输出锁存寄存器 GPIOn_LAT，而不是 GPIOn_DAT 寄存器。

10.11.2 端口数据输出锁存寄存器 (GPIOn_LAT) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:10	-	*1	0x0	保留
15:0	PORTn_LAT	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚端口数据输出锁存位 PORTB 位 [5:0]: PORTB 引脚端口数据输出锁存位 读寄存器时，读回此寄存器的锁定值，而不是引脚逻辑状态 : 0: 端口输出锁定为逻辑低电平 1: 端口输出锁定为逻辑高电平 写寄存器时，将数据锁存在寄存器中 : 0: 端口输出锁存为逻辑低电平 1: 端口输出锁存为逻辑高电平

©Copyright 2026, PADAUK Technology Co. Ltd

10.11.3 引脚中断类别设置寄存器 1 (GPIOn_ITS1) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_ITS1	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚中断类别 1 设置设定位 PORTB 位 [5:0]: PORTB 引脚中断类别 1 设置设定位 读寄存器时，读回引脚中断类别状态: 0: 端口引脚输入电平中断或沿跳变中断 1: 端口引脚输入任意电平中断 写寄存器时，设定引脚输入的中断类别: 0: 无效操作 1: 端口引脚输入中断类别为任意电平中断

10.11.4 引脚中断类别清除寄存器 1 (GPIOn_ITC1) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_ITC1	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚中断类别 1 清除设定位 PORTB 位 [5:0]: PORTB 引脚中断类别 1 清除设定位 读寄存器时，读回值始终为 0。 写寄存器时，设定引脚输入的中断类别: 0: 无效操作 1: 端口引脚输入电平中断或沿跳变中断

©Copyright 2026, PADAUK Technology Co. Ltd

10.11.5 端口输出使能寄存器 (GPIOn_OES) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:10	-	*1	0x0	保留
15:0	PORTn_OES	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚输出使能设定位 PORTB 位 [5:0]: PORTB 引脚输出使能设定位 读寄存器时，读回引脚的输入输出方向设置： 0: 端口引脚为输入状态 1: 端口引脚为输出状态 写寄存器时，引脚输出使能设定: 0: 无效操作 1: 端口引脚输出使能设定，引脚为输出模式

10.11.6 引脚输出禁止寄存器 (GPIOn_OEC) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_OEC	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚输出禁止设定位 PORTB 位 [5:0]: PORTB 引脚输出禁止设定位 读寄存器时，读回值始终为 0。 写寄存器时，引脚输出禁止设定 ： 0: 无效操作 1: 端口引脚输出使能禁止

©Copyright 2026, PADAUK Technology Co. Ltd

10.11.7 端口输入使能寄存器 (GPIOn_INES) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_INES	R/W	0xFF	PORTA 位[15:0]: PORTA 引脚输入使能设定位 PORTB 位[5:0]: PORTB 引脚输入使能设定位 读寄存器时，读回引脚的输入输出方向设置: 0: 端口引脚输入使能关闭 1: 端口引脚输入使能打开 写寄存器时，引脚输出使能设定: 0: 无效操作 1: 端口引脚输入使能设定

10.11.8 引脚输入禁止寄存器 (GPIOn_INEC) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_INEC	R/W	0x0	**PORTA 位 [15:0]: PORTA 引脚输入使能禁止设定位** **PORTB 位 [5:0]: PORTB 引脚输入使能禁止设定位** 读寄存器时，读回值始终为 0。 写寄存器时，引脚输出禁止设定: 0: 无效操作 1: 端口引脚输入使能禁止

10.11.9 引脚输入中断使能寄存器 (GPIOn_IES) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_IES	R/W	0x0	PORTA 位 [15:0]: PORTA 端口引脚输入中断使能设定位 PORTB 位 [5:0]:PORTB 端口引脚输入中断使能设定位 读寄存器时，读回引脚中断设定状态: 0: 端口引脚输入中断禁止 1: 端口引脚输入中断使能 写寄存器时，使能引脚的输入中断功能: 0: 无效操作 1: 端口引脚输入中断功能使能

10.11.10 引脚输入中断禁止寄存器 (GPIOn_IEC) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_IEC	R/W	0x0	PORTA 位 [15:0]: PORTA 端口引脚输入中断禁止设定位 PORTB 位 [5:0]: PORTB 端口引脚输入中断禁止设定位 读寄存器时，读回值始终为 0。 写寄存器时，使能引脚的输入中断功能: 0: 无效操作 1: 端口引脚输入中断功能禁止

©Copyright 2026, PADAUK Technology Co. Ltd

10.11.11 引脚中断类别设置寄存器 0 (GPIOn_ITS0) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_ITS0	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚中断类别 0 设置设定位 PORTB 位 [5:0]: PORTB 引脚中断类别 0 设置设定位 读寄存器时，读回引脚中断类别状态 (PAITS1 中断类别 1 对应位状态为 0）: 0: 端口引脚输入电平中断 (需配合 GPIOn_PLS) 1: 端口引脚输入沿跳变中断 (需配合 GPIOn_PLS) 写寄存器时，设定引脚输入的中断类别 (PAITS1 中断类别 1 对应位状态为 0): 0: 无效操作 1: 端口引脚输入中断类别为沿跳变中断

10.11.12 引脚中断类别清除寄存器 0 (GPIOn_ITC0) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_ITC0	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚中断类别 0 清除设定位 PORTB 位 [5:0]: PORTB 引脚中断类别 0 清除设定位 读寄存器时，读回值始终为 0。 写寄存器时，设定引脚输入的中断类别 (PAITS1 中断类别 1 对应位状态为 0): 0: 无效操作 1: 端口引脚输入中断类别为电平中断

©Copyright 2026, PADAUK Technology Co. Ltd

10.11.13 引脚中断极性设置寄存器 (GPIOn_PLS) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_PLS	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚中断极性设置设定位 PORTB 位 [5:0]: PORTB 引脚中断极性设置设定位 读寄存器时，读回引脚中断极性状态: 0: 端口引脚输入中断为低电平或下降沿有效 1: 端口引脚输入中断为高电平或上升沿有效 写寄存器时，设定引脚输入的中断极性: 0: 无效操作 1: 端口引脚输入中断极性为高电平或上升沿有效

10.11.14 引脚中断极性清除寄存器 (GPIOn_PLC) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_PLC	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚中断极性清除设定位 PORTB 位 [5:0]: PORTB 引脚中断极性清除设定位 读寄存器时，读回值始终为 0。 写寄存器时，设定引脚输入的中断极性： 0: 无效操作 1: 端口引脚输入中断极性为低电平或下降沿有效

©Copyright 2026, PADAUK Technology Co. Ltd

10.11.15 引脚中断标志位寄存器 (GPIOn_IST) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_IST	R/W1c	0x0	PORTA 位 [15:0]: PORTA 引脚中断标志位 PORTB 位 [5:0]: PORTB 引脚中断标志位 读寄存器时，读回引脚的输入中断标志： 0: 端口引脚输入无中断发生 1: 端口引脚输入有中断发生 写寄存器时，清除引脚的输入中断标志： 0: 无效操作 1: 端口引脚的输入中断标志清除

©Copyright 2026, PADAUK Technology Co. Ltd

10.11.16 引脚内部上拉使能寄存器 (GPIOn_PUS) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_PUS	R/W	0x0 or 0x6	PORTA 位 [15:0]: PORTA 引脚内部上拉使能设定位，复位值 0x0 PORTB 位 [5:0]: PORTB 引脚内部上拉使能设定位，复位值 0x6 读寄存器时，读回引脚输入上拉设置: 0: 端口引脚输入上拉功能禁止 1: 端口引脚输入上拉功能使能 写寄存器时，引脚输入上拉使能设定: 0: 无效操作 1: 端口引脚输入上拉使能

10.11.17 引脚内部上拉禁止寄存器 (GPIOn_PUC) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_PUC	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚内部上拉禁止设定位 PORTB 位 [5:0]: PORTB 引脚内部上拉禁止设定位 读寄存器时，读回值始终为 0。 写寄存器时，引脚输入上拉使能设定: 0: 无效操作 1: 端口引脚输入上拉禁止

10.11.18 引脚输出开漏使能寄存器 (GPIOn_ODS) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_ODS	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚开漏输出使能设定位 PORTB 位 [5:0]: PORTB 引脚开漏输出使能设定位 读寄存器时，读回引脚输出开漏设置: 0: 端口引脚输出开漏功能禁止 1: 端口引脚输出开漏功能使能 写寄存器时，引脚输出开漏使能设定: 0: 无效操作 1: 端口引脚输出开漏使能

10.11.19 引脚输出开漏禁止寄存器 (GPIOn_ODC) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_ODC	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚开漏输出禁止设定位 PORTB 位 [5:0]: PORTB 引脚开漏输出禁止设定位 读寄存器时，读回值始终为 0。 写寄存器时，引脚输出开漏使能设定: 0: 无效操作 1: 端口引脚输出开漏禁止

©Copyright 2026, PADAUK Technology Co. Ltd

10.11.20 引脚内部下拉使能寄存器 (GPIOn_PDS) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_PDS	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚内部下拉使能设定位 PORTB 位 [5:0]: PORTB 引脚内部下拉使能设定位 读寄存器时，读回引脚输入下拉设置: 0: 端口引脚输入下拉功能禁止 1: 端口引脚输入下拉功能使能 写寄存器时，引脚输入下拉使能设定: 0: 无效操作 1: 端口引脚输入下拉使能

10.11.21 引脚内部下拉禁止寄存器 (GPIOn_PDC) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_PDC	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚内部下拉禁止设定位 PORTB 位 [5:0]: PORTB 引脚内部下拉禁止设定位 读寄存器时，读回值始终为 0。 写寄存器时，引脚输入下拉使能设定: 0: 无效操作 1: 端口引脚输入下拉禁止

©Copyright 2026, PADAUK Technology Co. Ltd

10.11.22 引脚输出 PMOS 开漏输出使能寄存器 (GPIOn_PODS) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0	保留
15:6	-	*1	0	保留
15:0	PORTn_PODS	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚 PMOS 开漏输出使能设定位 PORTB 位 [5:0]: PORTB 引脚 PMOS 开漏输出使能设定位 读寄存器时，读回引脚输出开漏设置: 0: 端口引脚输出 PMOS 开漏功能禁止 1: 端口引脚输出 PMOS 开漏功能使能 写寄存器时，引脚输出开漏使能设定: 0: 无效操作 1: 端口引脚输出 PMOS 开漏使能

10.11.23 引脚输出 PMOS 开漏输出禁止寄存器 (GPIOn_PODC) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_PODC	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚 PMOS 开漏输出禁止设定位 PORTB 位 [5:0]: PORTB 引脚 PMOS 开漏输出禁止设定位 读寄存器时，读回值始终为 0。 写寄存器时，引脚输出开漏使能设定: 0: 无效操作 1: 端口引脚输出开漏禁止。

©Copyright 2026, PADAUK Technology Co. Ltd

10.11.24 端口施密特功能设置寄存器 (GPIOn_STE) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	Portn_STE	R/W	0x1	PORTA 位 [15:0]: PORTA 引脚施密特功能设置位 PORTB 位 [5:0]: PORTB 引脚施密特功能设置位 读寄存器时: 0: 端口的施密特功能设置为 0，不具有 hysteresis 功能。 1: 端口的施密特功能设置为 1，具有 hysteresis 功能。 写寄存器且对应引脚输出使能时: 0: 端口的施密特功能设置不变 1: 端口的施密特功能使能，相应 IO 口具有 hysteresis 功能

10.11.25 端口施密特功能清除寄存器 (GPIOn_STD) (n=A, B)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:6	-	*1	0x0	保留
15:0	PORTn_STD	R/W	0x0	PORTA 位 [15:0]: PORTA 引脚清施密特功能除位 PORTB 位 [5:0]: PORTB 引脚清施密特功能除位 读寄存器时: 该寄存器为只写寄存器，读出始终为 0 写寄存器时: 0: 端口的施密特功能设置不变 1: 端口的施密特功能清除为 0，相应 IO 口不具 hysteresis 功能。

©Copyright 2026, PADAUK Technology Co. Ltd

10.11.26 PORTA 外设复用置位寄存器 (GPIOA_AFR)

GPIOA_AFR 寄存器详细的功能设置说明如下所示。 具体的 PIN 位置参考第 5 章中表格。

Bit	Name	R/W	Reset	Description
31:24	-	R	0x0	Reserved
23:21	PA15_AFR	R/W	0x0	000: GPIO 001: TXD 010: RXD 011: MOSI 100: MISO 101: Reserved 110: SCL 111: Reserved
20:18	PA14_AFR	R/W	0x0	000: GPIO 001: Reserved 010: Reserved 011: SCK 100: Reserved 101: Reserved 110: MCO (Microcontroller Clock Output) 111: Reserved
17:15	PA5_AFR	R/W	0x0	000: GPIO 001: CS 010: Reserved 011: Reserved 100: Reserved 101: TIM2CH4 output 110: EPWM2N output 111: Reserved
14:12	PA4_AFR	R/W	0x0	000: GPIO 001: MISO 010: MOSI 011: Reserved 100: Reserved 101: TIM2CH3 output 110: EPWM2P output 111: Reserved

©Copyright 2026, PADAUK Technology Co. Ltd

11:9	PA3_AFR	R/W	0x0	000: GPIO 001: MOSI 010: MISO 011: Reserved 100: Reserved 101: TIM2CH2 output 110: EPWM2N output 111: Reserved
8:6	PA2_AFR	R/W	0x0	000: GPIO 001: SCK 010: Reserved 011: Reserved 100: Reserved 101: TIM2CH1 output 110: EPWM2P output 111: Reserved
5:3	PA1_AFR	R/W	0x0	000: GPIO 001: TXD 010: RXD 011: SCL 100: SDA 101: Reserved 110: EPWM1N output 111: Reserved
2:0	PA0_AFR	R/W	0x0	000: GPIO 001: RXD 010: TXD 011: SDA 100: SCL 101: Reserved 110: EPWM1P output 111: Reserved

©Copyright 2026, PADAUK Technology Co. Ltd

10.11.27 PORTB 外设复用置位寄存器 (GPIOB_AFR)

PORTB_AFR 寄存器详细的功能设置说明如下所示。具体的 PIN 位置参考第 5 章中表格。

Bit	Name	R/W	Reset	Description
31:18	-	R	0x0	Reserved
17:15	PB5_AFR	R/W	0x0	000: GPIO 001: RXD 010: TXD 011: CS 100: TIM2CH1 output 101: SDA 110: EPWM2N output 111: Reserved
14:12	PB4_AFR	R/W	0x0	000: GPIO 001: TXD 010: RXD 011: MISO 100: MOSI 101: Reserved 110: EPWM2P output 111: Reserved
11:9	PB3_AFR	R/W	0x0	000: GPIO 001: TXD 010: SCL 011: MOSI 100: MISO 101: Reserved 110: EPWM3P output 111: Reserved
8:6	PB2_AFR	R/W	0x0	000: GPIO 001: SDA 010: SCL 011: SCK 100: TIM2CH1 output 101: Reserved 110: EPWM3N output 111: Reserved

©Copyright 2026, PADAUK Technology Co. Ltd

5:3	PB1_AFR	R/W	0x0	000: GPIO 001: SCL 010: SDA 011: CS 100: TIM2CH1 output 101: Reserved 110: MCO (Microcontroller Clock Output) 111: Reserved
2:0	PB0_AFR	R/W	0x0	000: GPIO 001: RXD 010: TXD 011: MISO 100: MOSI 101: Reserved 110: SDA 111: Reserved

©Copyright 2026, PADAUK Technology Co. Ltd

10.11.28 引脚外设复用功能 1 配置寄存器 (FN1_AFR)

FN1_AFR 寄存器详细的功能设置说明如下所示。具体的 PIN 位置参考第 5 章中表格。

Bit	Name	R/W	Reset	Description
31:17	-	R	0x0	Reserved
16:13	EPETR	R/W	0x0	EPWM ETR remapping 0000: Reserved 0001: Remap to PA14 0010: Remap to PA15 0011: Remap to PB0 0100: Remap to PB1 0101: Remap to PB2 0110: Remap to PB3 0111: Remap to PB4 1000: Remap to PB5
12:10	ECAP2	R/W	0x0	ECAP2 remapping 000: Reserved 001: Remap to PA14 010: Remap to PA15 011: Remap to PB0 100: Remap to PB2 101: Remap to PB3 110: Remap to PB4
9:7	ECAP1	R/W	0x0	ECAP1 remapping 000: Reserved 001: Remap to PA14 010: Remap to PA15 011: Remap to PB0 100: Remap to PB2 101: Remap to PB3 110: Remap to PB4

©Copyright 2026, PADAUK Technology Co. Ltd

6:4	ECAP0	R/W	0x0	ECAP0 remapping 000: Reserved 001: Remap to PA14 010: Remap to PA15 011: Remap toPB0 100: Remap to PB2 101: Remap to PB3 110: Remap to PB4
3:0	BKIN	R/W	0x0	BK_IN remapping 0000: Reserved 0001: Remap to PA6 0010: Remap to PA7 0011: Remap to PA8 0100: Remap to PA9 0101: Remap to PA10 0110: Remap to PA11 0111: Remap to PA12 1000: Remap to PA13 1001: Remap to PB1 1010: Remap to PB3 1011: Remap to PB5

©Copyright 2026, PADAUK Technology Co. Ltd

10.11.29 引脚外设复用功能 2 配置寄存器 (FN2_AFR)

FN2_AFR 寄存器详细的功能设置说明如下所示。具体的 PIN 位置参考第 5 章中表格。

Bit	Name	R/W	Reset	Description
31:21	-	R	0x0	保留
20	I2C_PULL7	R/W	0x0	I2c pull High (pull high 10Kohm) 0: Disable PB3 pull high 1: As I2CEN=1,enable PB3 pull high
19	I2C_PULL6	R/W	0x0	I2c pull High (pull high 10Kohm) 0: Disable PB5 pull high 1: As I2CEN=1,enable PB5 pull high
18	I2C_PULL5	R/W	0x0	I2c pull High (pull high 10Kohm) 0: Disable PA0 pull high 1: As I2CEN=1,enable PA0 pull high
17	I2C_PULL4	R/W	0x0	I2c pull High (pull high 10Kohm) 0: Disable PA1 pull high 1: As I2CEN=1,enable PA1 pull high
16	I2C_PULL3	R/W	0x0	I2c pull High (pull high 10Kohm) 0: Disable PA15 pull high 1: As I2CEN=1,enable PA15 pull high
15	I2C_PULL2	R/W	0x0	I2c pull High (pull high 10Kohm) 0: Disable PB0 pull high 1: As I2CEN=1,enable PB0 pull high
14	I2C_PULL1	R/W	0x0	I2c pull High (pull high 10Kohm) 0: Disable PB1 pull high 1: As I2CEN=1,enable PB1 pull high
13	I2C_PULL0	R/W	0x0	I2c pull High (pull high 10Kohm) 0: Disable PB2 pull high 1: As I2CEN=1,enable PB2 pull high

©Copyright 2026, PADAUK Technology Co. Ltd

12:9	T2ETR	R/W	0x0	TIM2 ETR remapping 0000: Reserved 0001: Remap to PA14 0010: Remap to PA15 0011: Remap to PB0 0100: Remap to PB1 0101: Remap to PB2 0110: Remap to PB3 0111: Remap to PB4 1000: Remap to PB5
8:6	TCAP2	R/W	0x0	TIM2 CAP2 remapping 000: Reserved 001: Remap to PA14 010: Remap to PA15 011: Remap to PB0 100: Remap to PB2 101: Remap to PB3 110: Remap to PB4
5:3	TCAP1	R/W	0x0	TIM2 CAP1 remapping 000: Reserved 001: Remap to PA14 010: Remap to PA15 011: Remap to PB0 100: Remap to PB2 101: Remap to PB3 110: Remap to PB4
2:0	TCAP0	R/W	0x0	TIM2 CAP0 remapping 000: Reserved 001: Remap to PA14 010: Remap to PA15 011: Remap to PB0 100: Remap to PB2 101: Remap to PB3 110: Remap to PB4

©Copyright 2026, PADAUK Technology Co. Ltd

11. 基礎定时器 (TIM0, 1, LPTIM)

11.1 定时器 TIM0/1

定时器 TIM0/1 为 16 位宽基礎定时器，工作于递增计数模式。

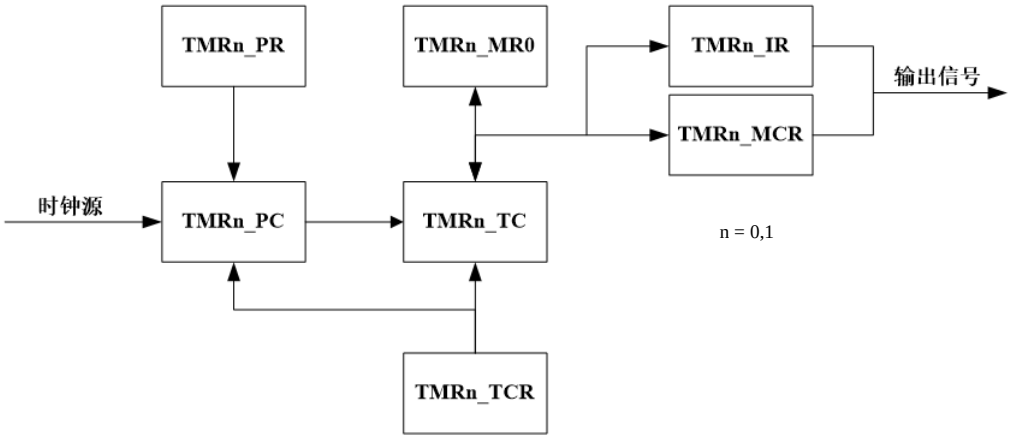


图 11-1 定时器 TIM0 工作原理图

TIM0_PC 寄存器根据 TIM0_PR 寄存器内设置的分频系数对时钟源进行预分频，TIM0_TC 寄存器在 TIM0_PC 寄存器产生的时钟下为定时器 TIM0 计数，当 TIM0_TC 寄存器与 TIM_MR0 寄存器内的值相同时，TIM0_IR 寄存器会产生比较定时中断信号（TIM0_IR 寄存器的 MR0 置 1），根据 TIM0_MCR 寄存器内的参数设置，定时器 TIM0 产生相应的输出信号（输出比较定时中断信号，复位定时器或停止定时器）。TIM0_TCR 用于使能和复位定时器 TIM0。

©Copyright 2026, PADAUK Technology Co. Ltd

11.2 TIM 0/1 寄存器列表

地址	寄存器	描述
0x4000_0000	TIM0_IR	TIM0 中断寄存器
0x4000_0004	TIM0_TCR	TIM0 控制寄存器
0x4000_0008	TIM0_TC	TIM0 计数值寄存器
0x4000_000C	TIM0_PR	TIM0 预分频系数寄存器
0x4000_0010	TIM0_PC	TIM0 预分频计数值寄存器
0x4000_0014	TIM0_MCR	TIM0 匹配控制寄存器
0x4000_0018	TIM0_MR0	TIM0 匹配值寄存器

表 11-1 TIM0 寄存器列表

地址	寄存器	描述
0x4000_0800	TIM1_IR	TIM1 中断寄存器
0x4000_0804	TIM1_TCR	TIM1 控制寄存器
0x4000_0808	TIM1_TC	TIM1 计数值寄存器
0x4000_080C	TIM1_PR	TIM1 预分频系数寄存器
0x4000_0810	TIM1_PC	TIM1 预分频计数值寄存器
0x4000_0814	TIM1_MCR	TIM1 匹配控制寄存器
0x4000_0818	TIM1_MR0	TIM1 匹配值寄存器

表 11-2 TIM1 寄存器列表

©Copyright 2026, PADAUK Technology Co. Ltd

11.3 定时器 LPTIM

定时器 LPTIM 是一个 16 位低功耗定时器，时钟为内部 32 KHz 或系统时钟。定时器 LPTIM 还能够在休眠状态下工作，输出信号可唤醒休眠。

定时器 LPTIM 的两个时钟域间有同步逻辑，当 LPTIM 的控制寄存器由系统时钟操作更新时，随即读这些寄存器时，能立即读到更新值，但该更新值需要等定时器在两个计数时钟域周期同步后才会生效。

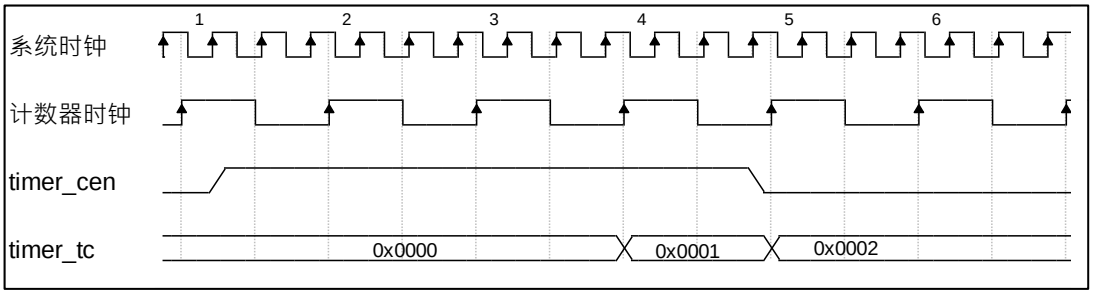


图 11-2 LPTIM 使能信号时序示意图

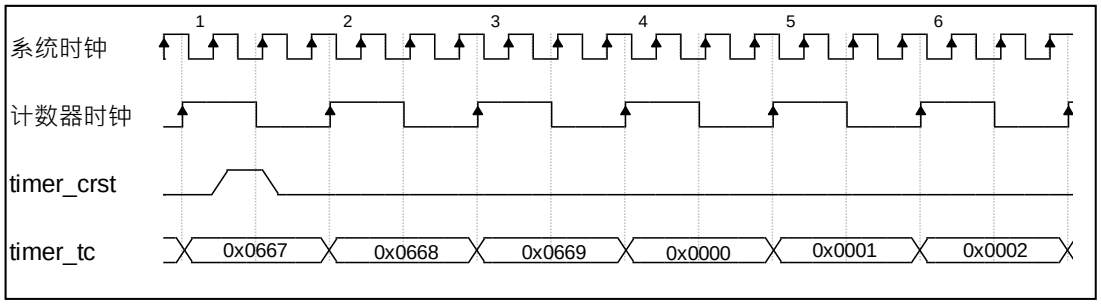


图 11-3 LPTIM 复位信号时序示意图

控制定时器 LPTIM 时，应先配置好各种参数，最后使能计数。

11.4 LPTIM 寄存器列表

地址	寄存器	描述
0x4000_C800	LPTIM_IR	LPTIM 中断寄存器
0x4000_C804	LPTIM_TCR	LPTIM 控制寄存器
0x4000_C808	LPTIM_TC	LPTIM 计数值寄存器
0x4000_C80C	LPTIM_PR	LPTIM 预分频系数寄存器
0x4000_C810	LPTIM_PC	LPTIM 预分频计数值寄存器
0x4000_C814	LPTIM_MCR	LPTIM 匹配控制寄存器
0x4000_C818	LPTIM_MR0	LPTIM 匹配值寄存器

表 11-3 LPTIM 寄存器列表

©Copyright 2026, PADAUK Technology Co. Ltd

11.5 TIM0, 1, LPTIM 寄存器描述 (名稱)

11.5.1 定时器 TIMn 中断寄存器 (TIMn_IR (n = 0,1, LPTIM))

Bit	Name	R/W	Reset	Description
31:1	-	R	0x0	保留
0	MR0	R/W1c	0x0	定时器匹配 0 中断标志位 0: TIMn 的 TIMn_TC 寄存器与 TIMn_MR 寄存器的值未发生过匹配 1: TIMn 的 TIMn_TC 寄存器与 TIMn_MR 寄存器的值发生过匹配，

注: TIM_IR 寄存器包含定时器 TIM 模块中的匹配中断标志位 MR0。只要 TIMn_TC 寄存器和 TIMn_MR0 寄存器匹配就会触发相应的中断标志位 MR0，不受相关中断使能位（TIMn_MCR 寄存器内的参数）设置影响

11.5.2 定时器 TIMn 控制寄存器 (TIMn_TCR (n = 0,1, LPTIM))

Bit	Name	R/W	Reset	Description
31:7	-	R	0x0	保留
6:4	TIMn_TRIG	R/W	0x0	0x0: Normal 启动定时器 n(n=0,1) 定时功能 0x1: EPWM1P(CH1 output) Rising 触发启动定时器 n 定时功能(AS CEn = 1) 0x2: EPWM2P(CH2 output) Rising 触发启动定时器 n 定时功能(AS CEn = 1) 0x3: EPWM3P(CH3 output) Rising 触发启动定时器 n 定时功能(AS CEn = 1) 0x4: EPWM1P(CH1 output) Falling 触发启动定时器 n 定时功能(AS CEn = 1) 0x5: EPWM2P(CH2 output) Falling 触发启动定时器 n 定时功能(AS CEn = 1) 0x6: EPWM3P(CH3 output) Falling 触发启动定时器 n 定时功能(AS CEn = 1) 0x7: Reserved
3:2	CLKS	R	0x0	**定时器 TIM0, 1 时钟选择控制** 0: 系统时钟 其它: 无效
1	CRst	R/W	0x0	**定时器模块软件复位控制位** 当写此位为 1 时，会对定时器计数值寄存器和预分频计数值寄存器执行一次复位操作，LPTIM 由于有同步逻辑，所以从写此寄存器到复位完成有延时，复位同步完成后硬件会自动对此位清 0。 读: 0: 定时器不在复位过程中 1: 定时器计数值寄存器、预分频计数值寄存器复位过程中 写: 0: 无效 1: 复位定时器计数器寄存器和预分频器寄存器，复位完成后硬件自动清 0
0	CEn	R/W	0x0	定时器模块工作使能控制位 0: 定时器定时功能禁止 1: 定时器定时功能启动 (Base on Trigger Source)

LPTIM

Bit	Name	R/W	Reset	Description
31:4	-	R	0x0	保留
3:2	CLKS	R/W	0x0	**定时器 LPTIM 时钟选择控制** 0: 系统时钟 其它: OSC_32KHz 时钟
1	CRst	R/W	0x0	**定时器模块软件复位控制位** 当写此位为 1 时，会对定时器计数值寄存器和预分频计数值寄存器执行一次复位操作，LPTIM 由于有同步逻辑，所以从写此寄存器到复位完成有延时，复位同步完成后硬件会自动对此位清 0。 读: 0: 定时器不在复位过程中 1: 定时器计数值寄存器、预分频计数值寄存器复位过程中 写: 0: 无效 1: 复位定时器计数器寄存器和预分频器寄存器，复位完成后硬件自动清 0
0	CEn	R/W	0x0	定时器模块工作使能控制位 0: 定时器定时功能禁止 1: 定时器定时功能启动 (Base on Trigger Source)

11.5.3 定时器 TIMn 当前计数值寄存器 (TIMn_TC (n = 0, 1, LPTIM))

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:0	TC	R/W	0x0	定时器当前值 该寄存器在 LPTIM 中只读不可写 取值范围 0x0000~0xFFFF

11.5.4 定时器 TIMn 分频计数最大值寄存器 (TIMn_PR (n = 0, 1, LPTIM))

Bit	Name	R/W	Reset	Description
31:8	-	R	0x0	保留
7:0	PR	R/W	0x0	定时器预分频系数 取值范围为 0x00~0xFF。系统时钟的分频计算公式为: fTIM = fSYS / (PR + 1) 其中: fTIM 为定时器工作计数的时钟频率 fSYS 为芯片系统时钟频率 PR 为预分频系数 取值范围 0x00~0xFF

11.5.5 定时器 TIMn 当前分频计数值寄存器 (TIMn_PC (n = 0, 1, LPTIM))

Bit	Name	R/W	Reset	Description
31:8	-	R	0x0	保留
7:0	PC	R/W	0x0	定时器分频计数器当前值 取值范围 0x00~0xFF LPTIM 只读

©Copyright 2026, PADAUK Technology Co. Ltd

11.5.6 定时器 TIMn 匹配控制寄存器 (TIMn_MCR (n = 0, 1, LPTIM))

Bit	Name	R/W	Reset	Description
31:3	-	R	0x0	保留
2	MR0STOP	R/W	0x0	TIMn_TC 和 TIMn_MR0 匹配时计数器停止控制位 0: TIMn_TC 不停止计数 1: TIMn_TC 停止计数，计数器禁止（TIMn_TCR 寄存器的 CEn 清零）
1	MR0RST	R/W	0x0	TIMn_TC 和 TIMn_MR0 匹配时计数器复位控制位 0: 不产生复位 1: 产生复位
0	MR0INT	R/W	0x0	TIMn_TC 和 TIMn_MR0 匹配时产生中断控制位 0: 不产生中断 1: 产生中断

11.5.7 定时器 TIMn 匹配值寄存器 (TIMn_MR0 (n = 0,1, LPTIM))

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:0	MATCH	R/W	0x0	定时器比较值 取值范围 0x0000~0xFFFF。

注: TIMn_MR0 寄存器用于保存定时器 TIMn 匹配值。

12. 高级定时器与通用定时器 (EPWM, TIM2)

12.1 EPWM 简介

高级定时器 (EPWM) 由一个 20 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 RCC 时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

12.2 EPWM 主要特性

EPWM 定时器的功能包括:

- 20 位向上、向下、向上/向下自动装载计数器
- 16 位可编程(可以实时修改)预分频器，计数器时钟频率的分频系数为 1～65535 之间的任意数值
- 多达 4 个独立通道(EPWM_CH1,2,3,4):
  - 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

©Copyright 2026, PADAUK Technology Co. Ltd

- 如下事件发生时产生中断:
  - 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:
    (1) 外部 BKIN 电平信号(高电平或低电平)
    (2) 模拟比较器0,1的输出(输出高或输出低)
    (3) ADC 阈值比转换结果事件發生.(可選兩個通道)
    (4) 低電壓檢測低於選擇閾值.
    (5) WDG 溢出事件發生
    (6) CPU 異常事件發生
- 支持针对定位的增量 (正交) 编码器
- 触发输入作为外部时钟或者按周期的电流管

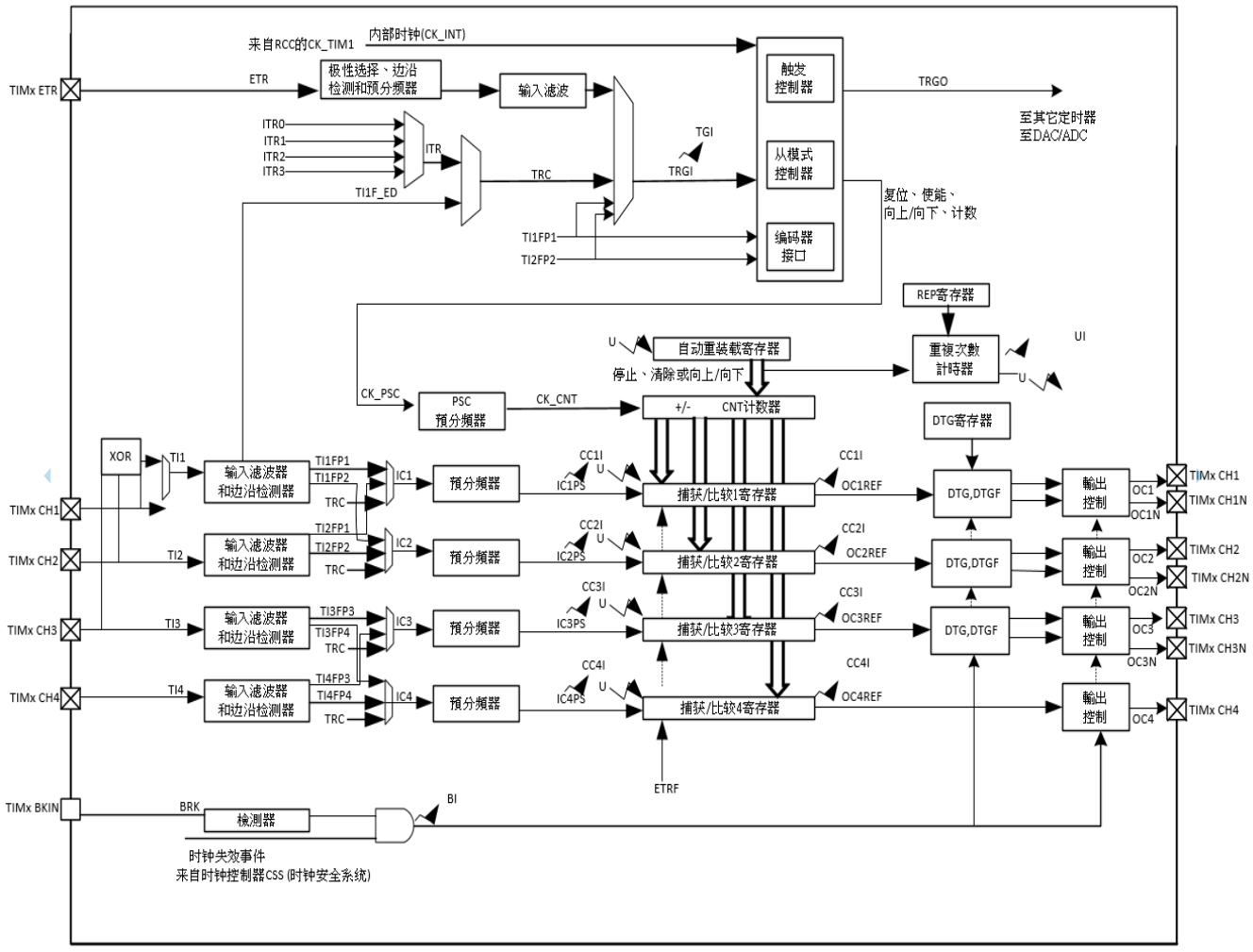


图 12-1 高级控制定时器框图

注:

Reg 根据控制位的设定，在 U(更新)事件时传送预加载寄存器的内容至工作寄存器

事件

中断

12.3 EPWM 功能描述

12.3.1 时基单元

可编程高级控制定时器的主要部分是一个 20 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 (EPWM_CNT)
- 预分频器寄存器 (EPWM_PSC)
- 自动装载寄存器 (EPWM_ARR)
- 重复次数寄存器 (EPWM_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 EPWM_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 EPWM_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 EPWM_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 EPWM_CR1 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 EPWM_PSC 寄存器中的)16 位寄存器控制的 20 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。图 12-2 和图 12-3 给出了在预分频器运行时，更改计数器参数的例子。

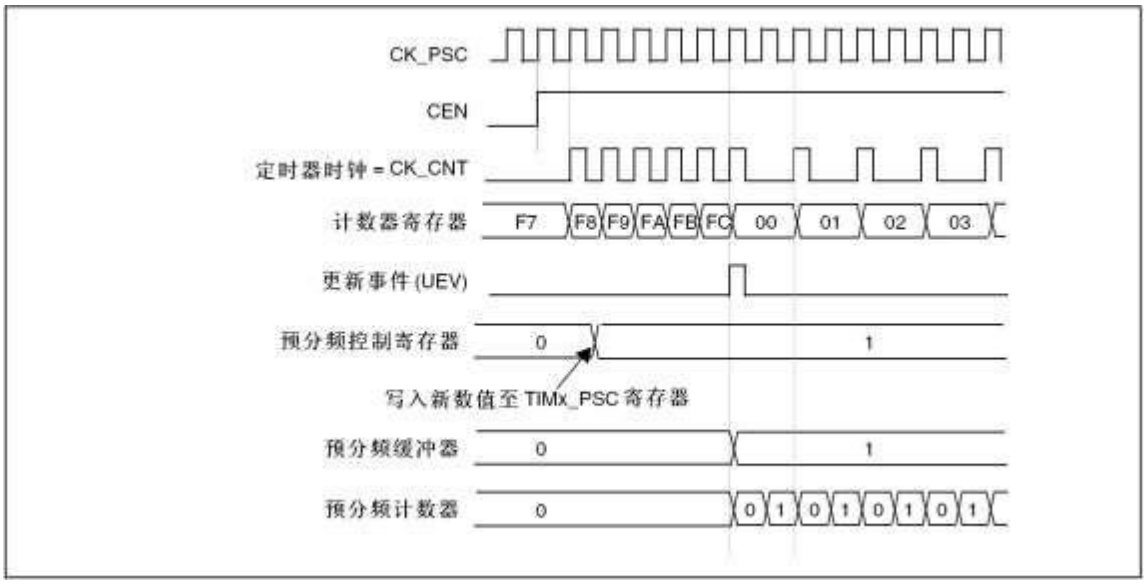


图 12-2 当预分频器的参数从 1 变到 2 时，计数器的时序图

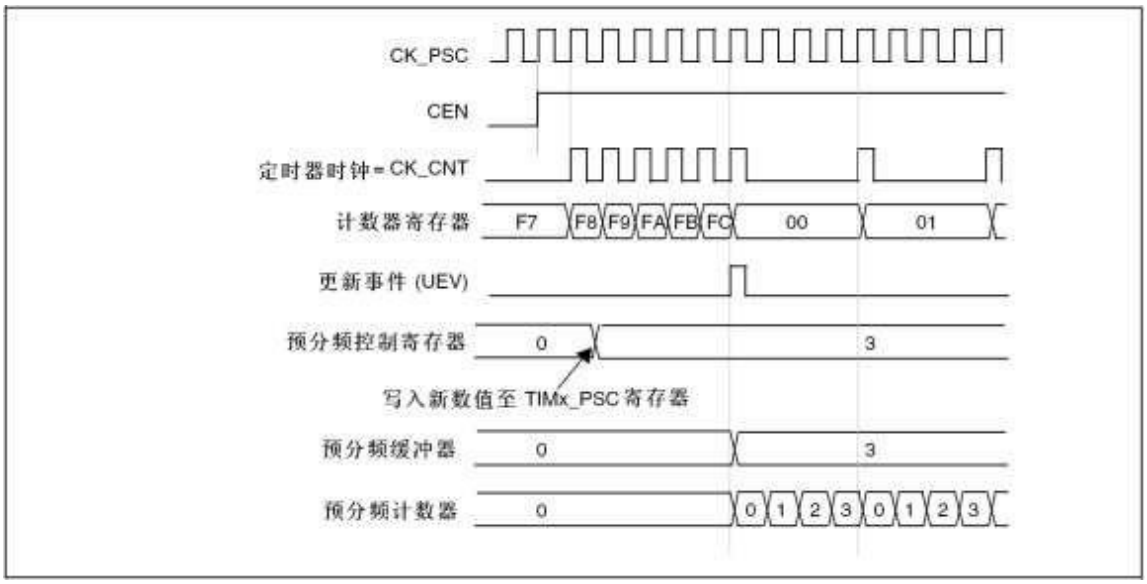


图 12-3 当预分频器的参数从 1 变到 4 时，计数器的时序图

12.3.2 计数器模式

12.3.2.1 向上计数模式

在向上计数模式中，计数器从 0 计数到自动加载值 (EPWM_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 (EPWM_RCR) 时，产生更新事件(UEV)；否则每次计数器溢出时产生更新事件。在 EPWM_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 EPWM_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 EPWM_CR1 寄存器中的 URS 位(选择更新请求)，设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志(即不产生中断)。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

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

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

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

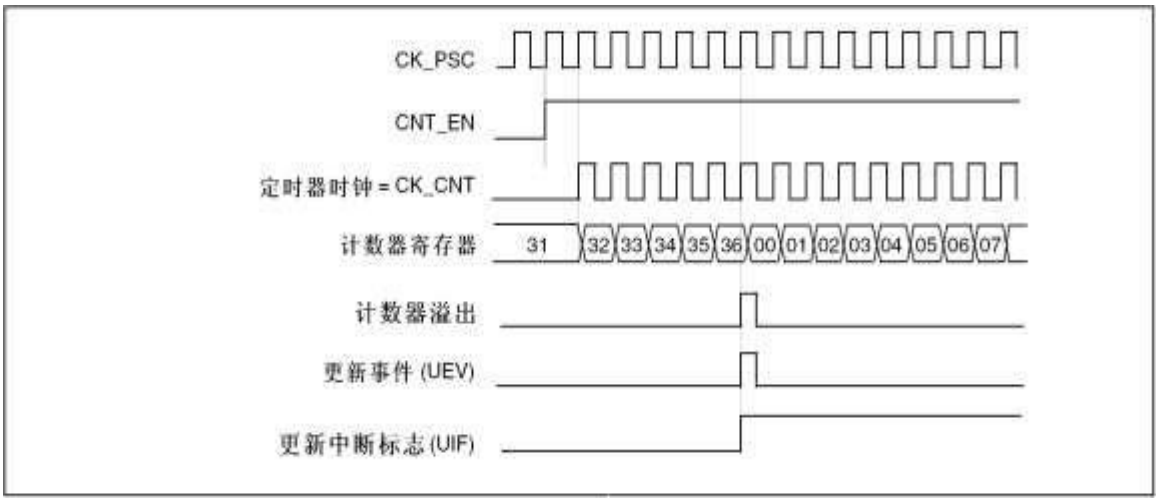


图 12-4 计数器时序图: 内部时钟分频因子为 1

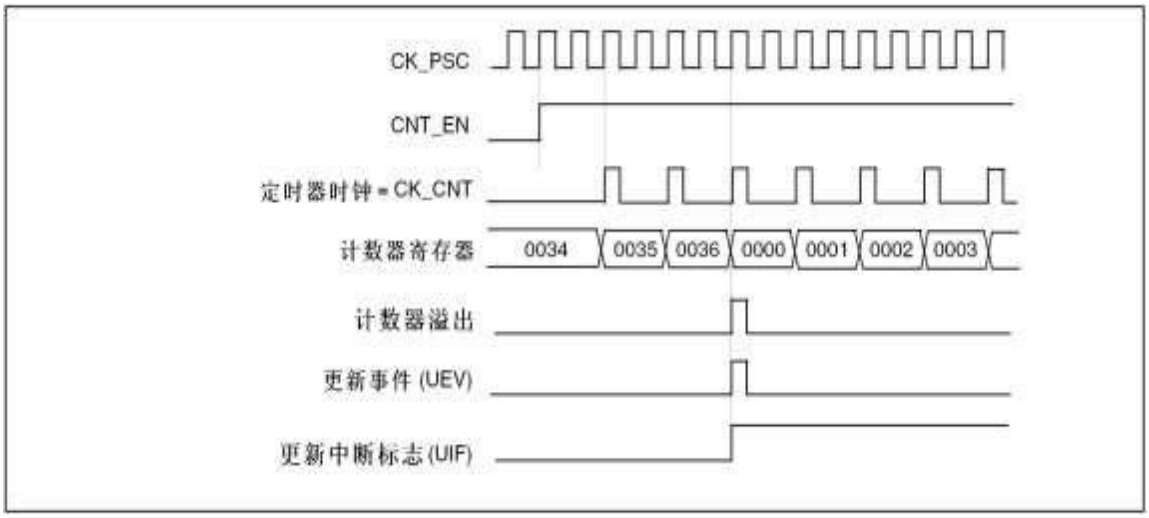


图 12-5 计数器时序图: 内部时钟分频因子为 2

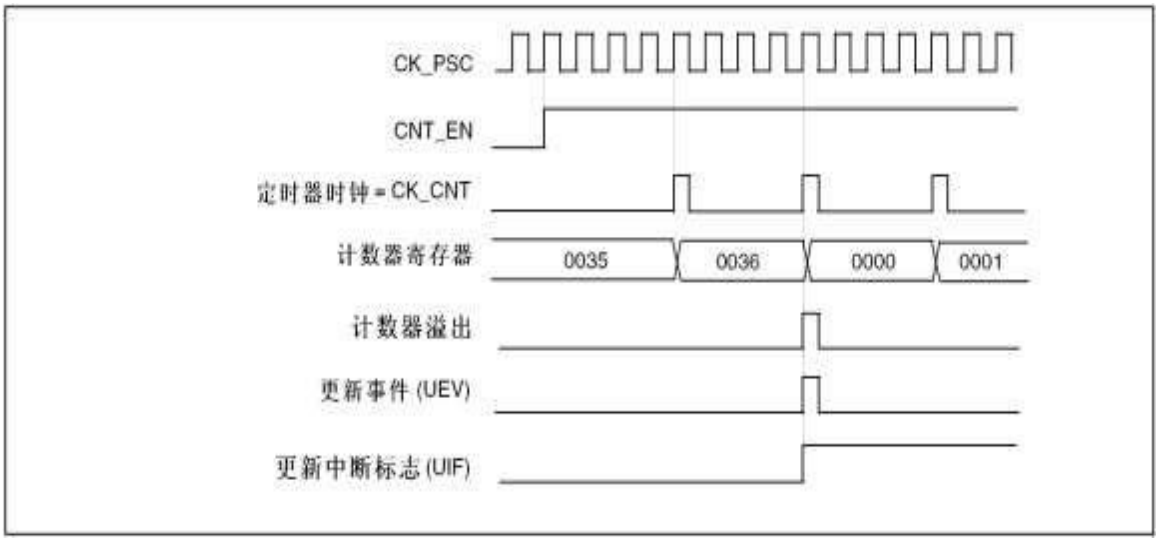


图 12-6 计数器时序图: 内部时钟分频因子为 4

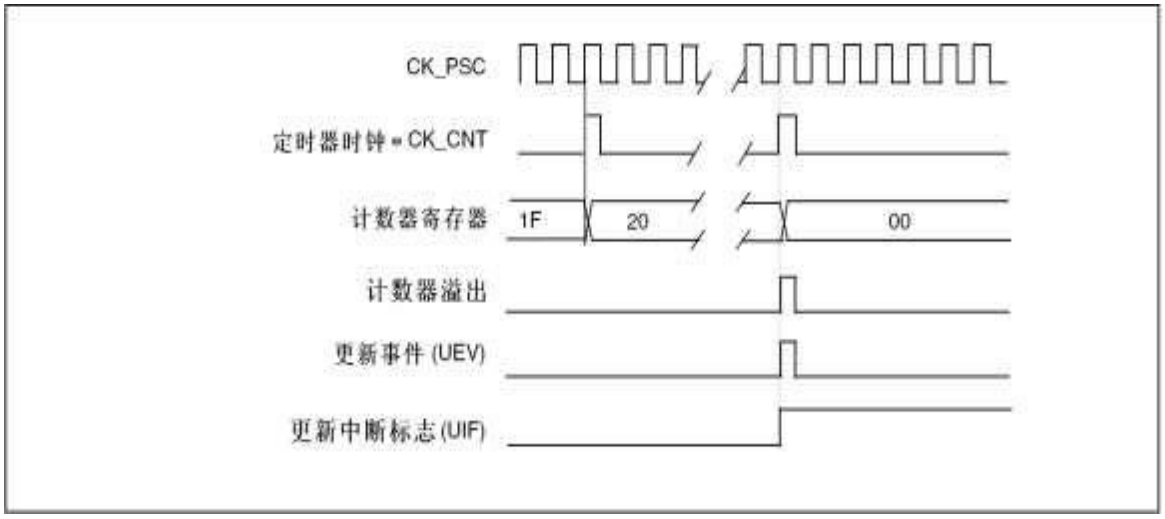


图 12-7 计数器时序图: 内部时钟分频因子为 N

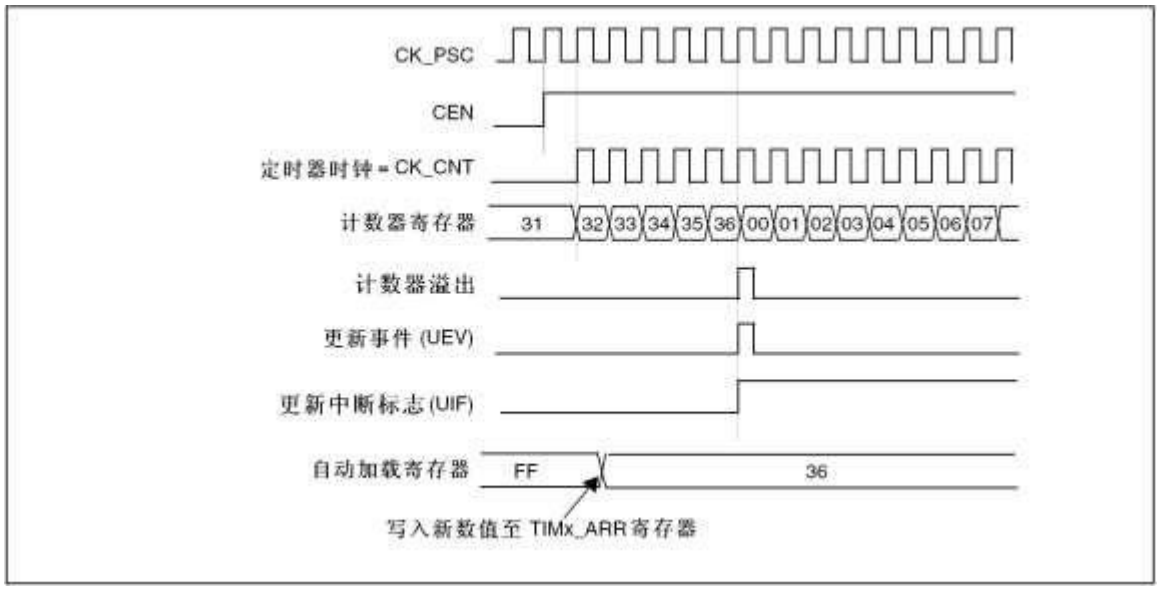


图 12-8 计数器时序图: 当 ARPE=0 时的更新事件 (EPWM_ARR 没有预装入)

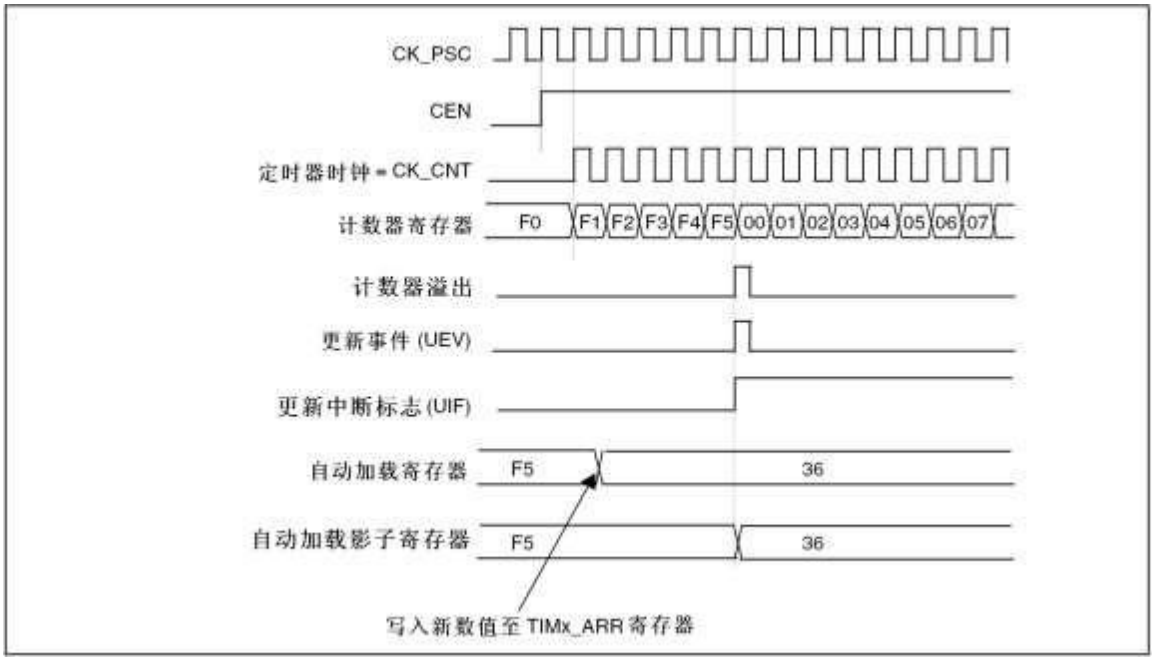


图 12-9 计数器时序图: 当 ARPE=1 时的更新事件 (预装入了 EPWM_ARR)

12.3.2.2 向下计数模式

在向下模式中，计数器从自动装入的值(EPWM_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(EPWM_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产生更新事件。

在 EPWM_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 EPWM_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 EPWM_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)，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(EPWM_SR 寄存器中的 UIF 位)也被设置。

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

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 EPWM_ARR=0x36 时，计数器在不同时钟频率下的操作例子。

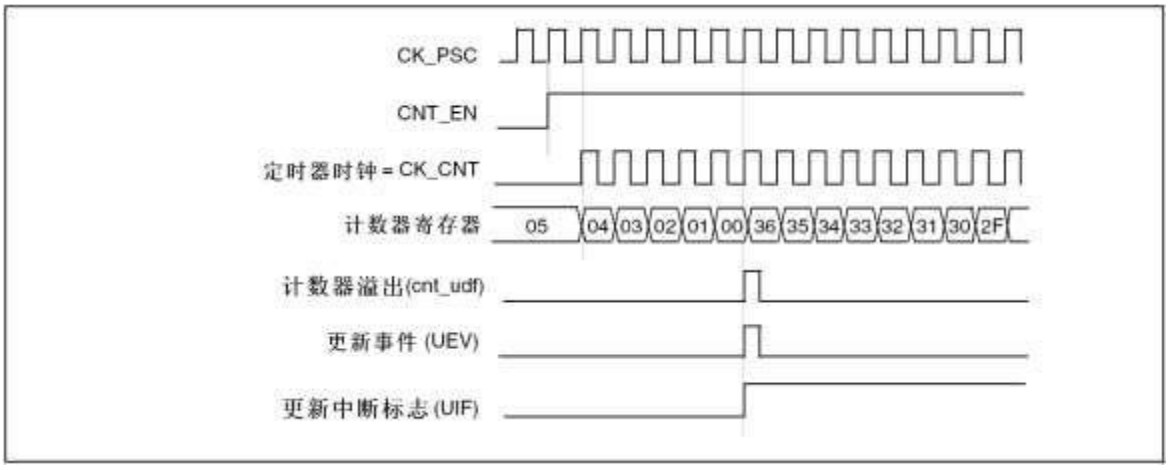


图 12-10 计数器时序图: 内部时钟分频因子为 1

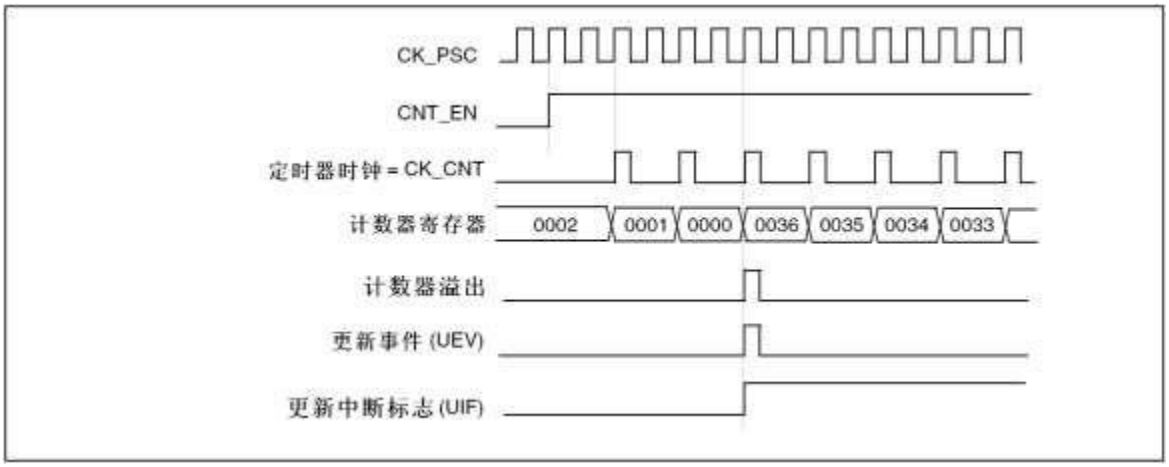


图 12-11 计数器时序图: 内部时钟分频因子为 2

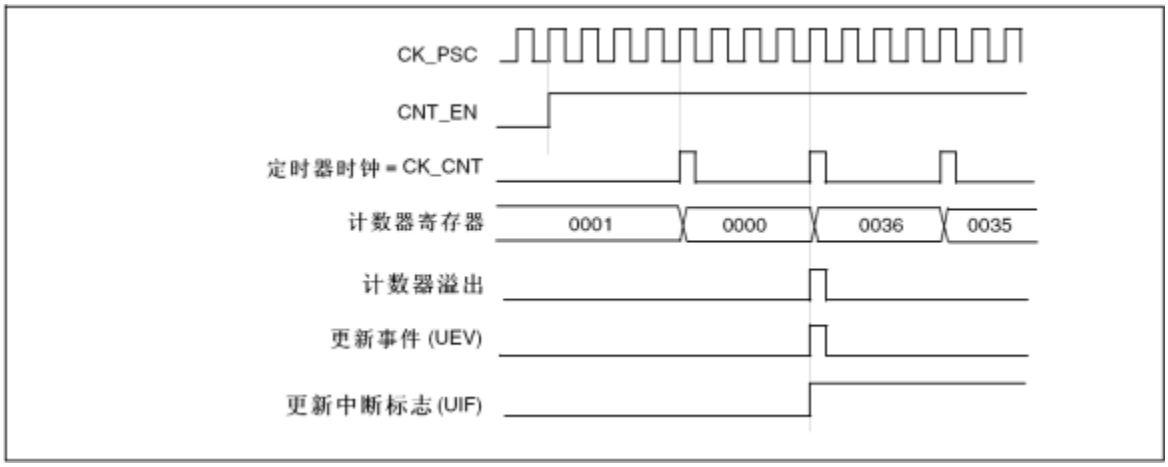


图 12-12 计数器时序图: 内部时钟分频因子为 4

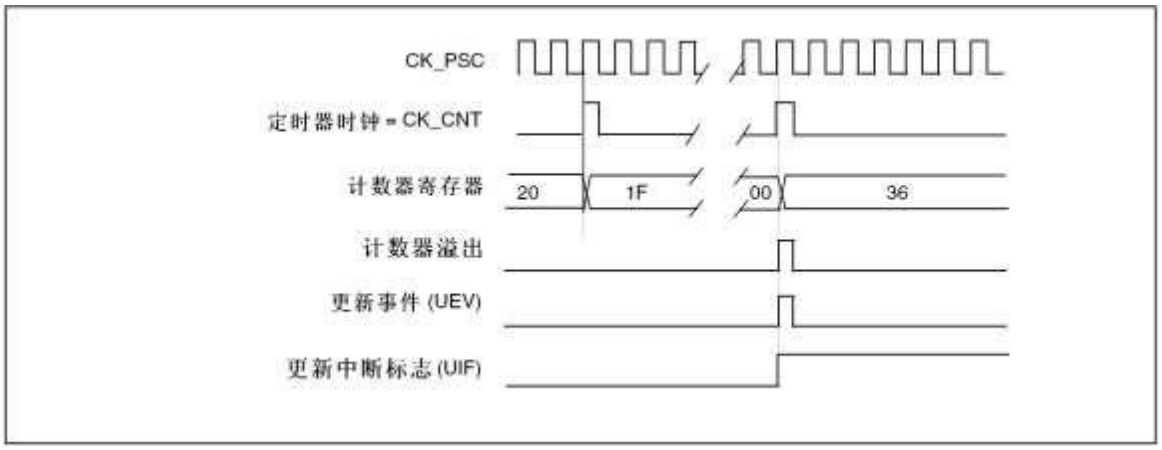


图 12-13 计数器时序图: 内部时钟分频因子为 N

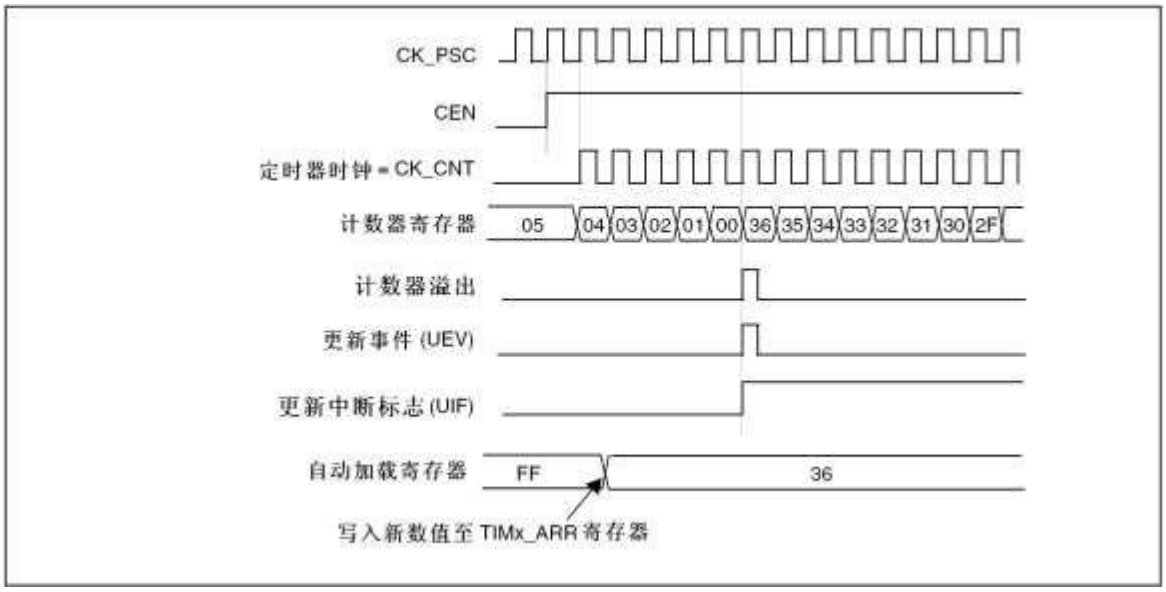


图 12-14 计数器时序图: 当没有使用重复计数器时的更新事件

12.3.2.3 中央对齐模式 (向上 / 向下计数)

在中央对齐模式，计数器从 0 开始计数到自动加载的值(EPWM_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 EPWM_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 EPWM_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 EPWM_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 EPWM_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)，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(EPWM_SR 寄存器中的 UIF 位)也被设置。

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

©Copyright 2026, PADAUK Technology Co. Ltd

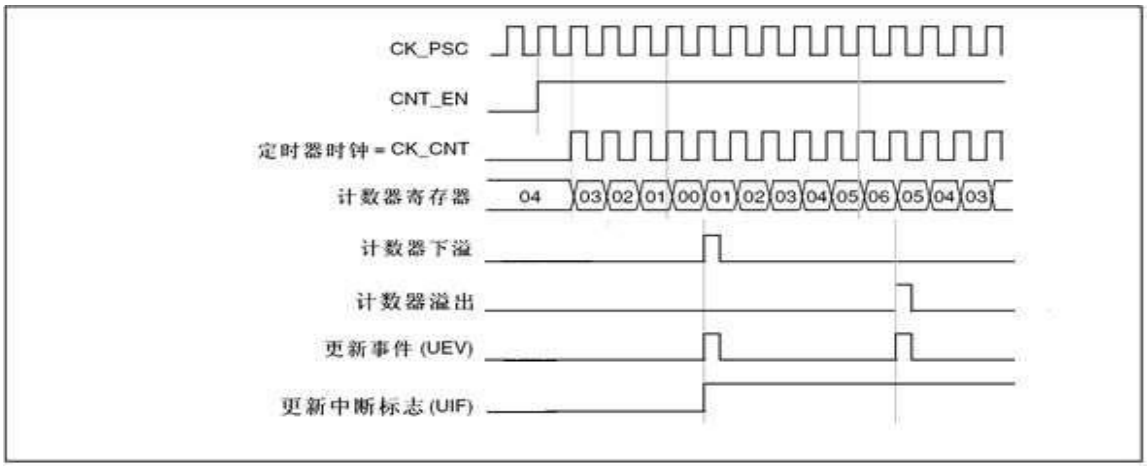


图 12-15 计数器时序图: 内部时钟分频因子为 1，EPWM_ARR = 0x6

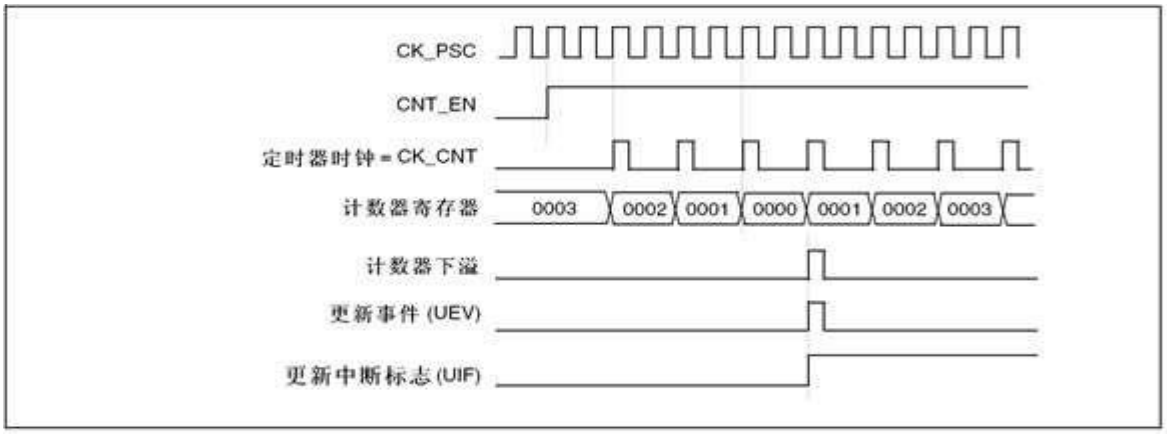


图 12-16 计数器时序图: 内部时钟分频因子为 2

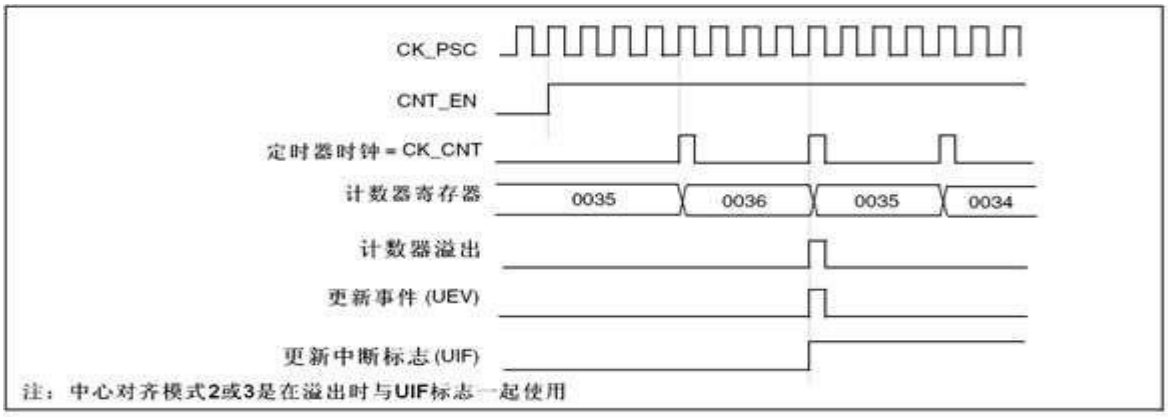


图 12-17 计数器时序图: 内部时钟分频因子为 4，EPWM_ARR = 0x36

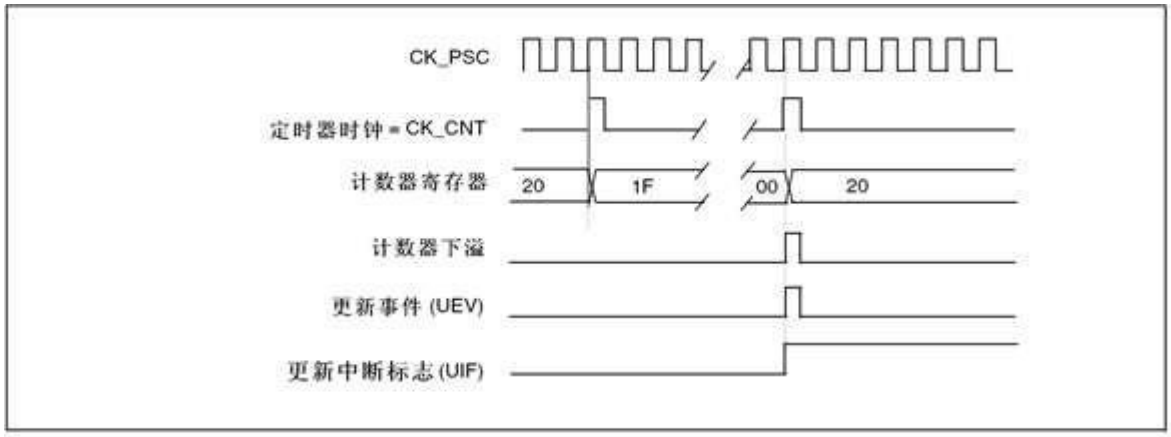


图 12-18 计数器时序图: 内部时钟分频因子为 N

©Copyright 2026, PADAUK Technology Co. Ltd

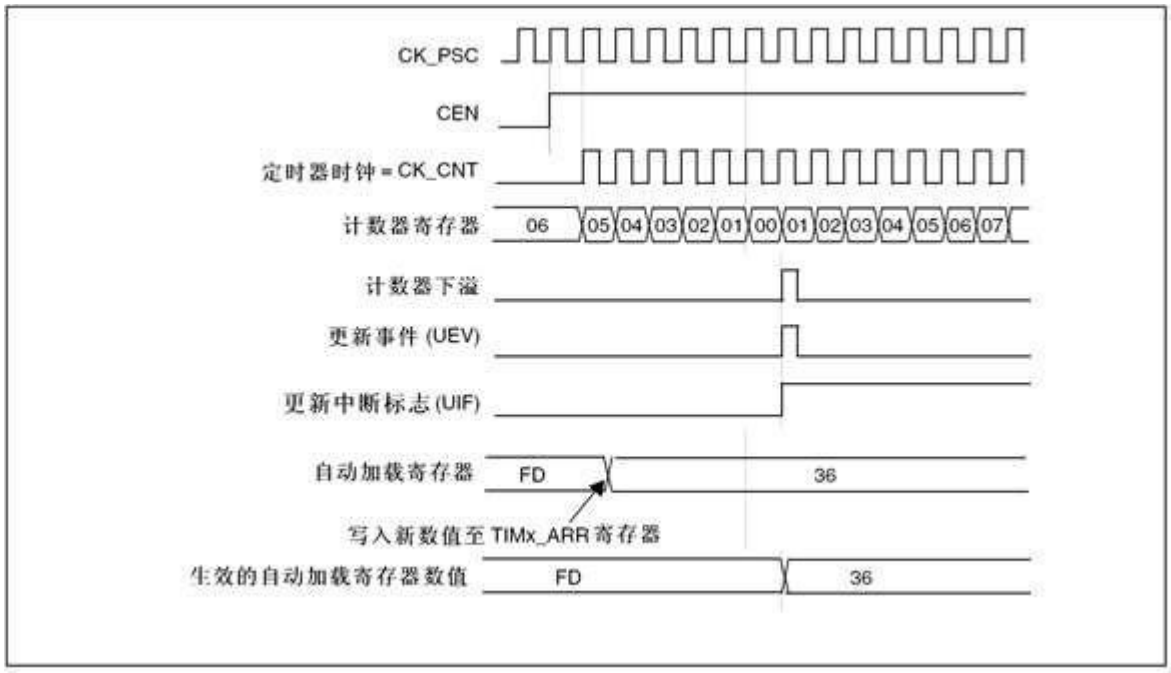


图 12-19 计数器时序图: ARPE = 1 时的更新事件(计数器下溢)

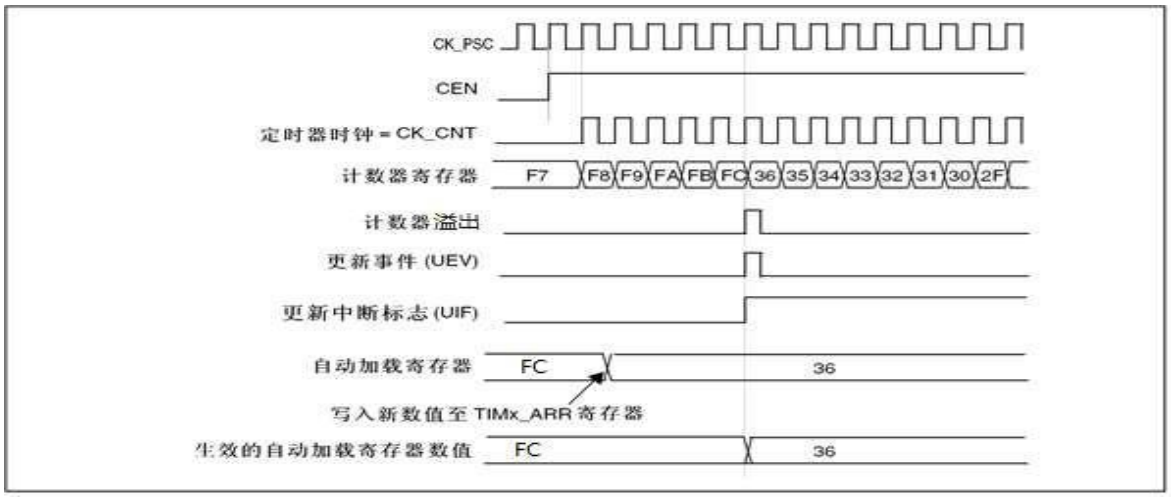


图 12-20 计数器时序图: ARPE = 1 时的更新事件(计数器溢出)

©Copyright 2026, PADAUK Technology Co. Ltd

12.3.3 重复计数器

12.3.1 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(EPWM_ARR 自动重载入寄存器，EPWM_PSC 预装载寄存器，还有在比较模式下的捕获/比较寄存器 EPWM_CCRx)，N 是 EPWM_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。

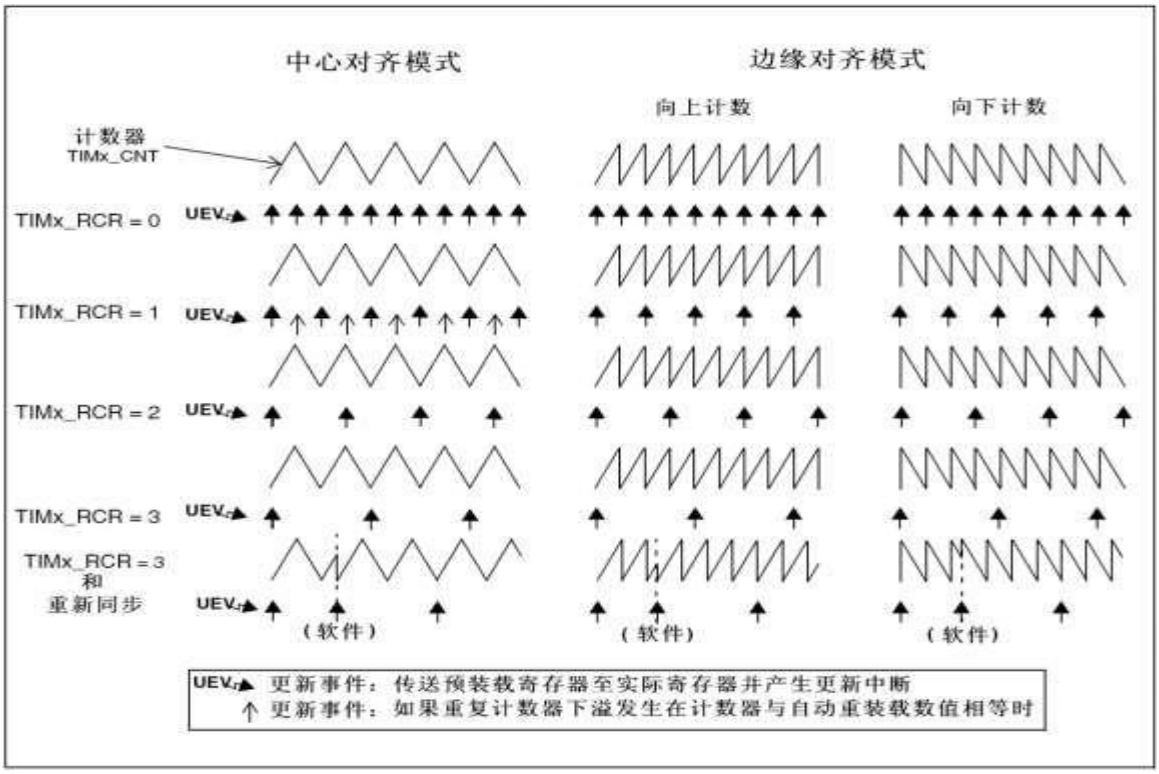


图 12-21 不同模式下更新速率的例子，及 EPWM_RCR 的寄存器设置

重复计数器是自动加载的，重复速率是由 EPWM_RCR 寄存器的值定义 (参看图 12-21)。当更新事件由软件产生 (通过设置 EPWM_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 EPWM_RCR 寄存器中的内容被重载入到重复计数器。

©Copyright 2026, PADAUK Technology Co. Ltd

12.3.4 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 EPWM 而作为另一个定时器TIM2 的预分频器，详见 12.5.3 节。

如果禁止了从模式控制器 (SMS = 000)，则 CEN、DIR (EPWM_CR1 寄存器) 和 UG 位(EPWM_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。

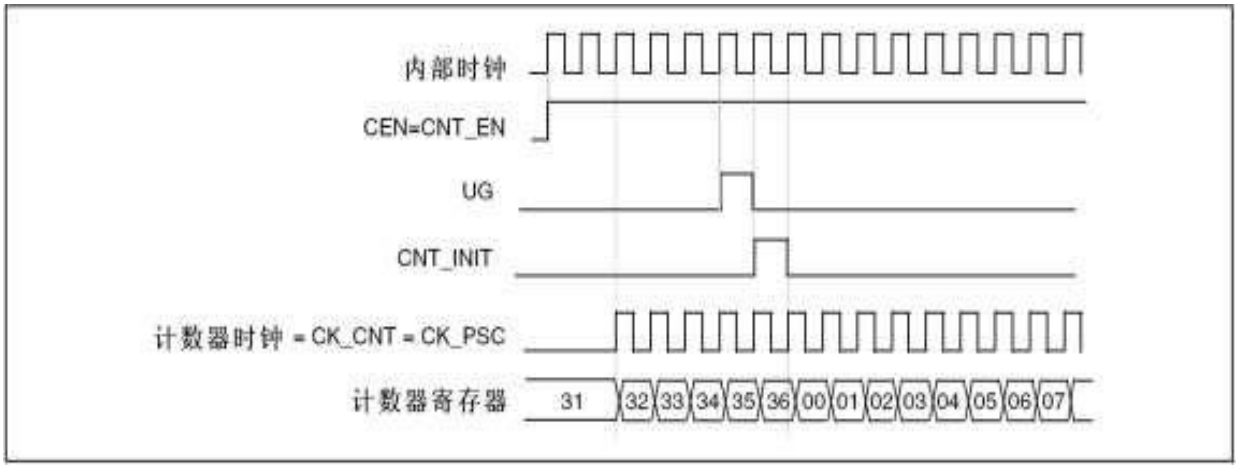


图 12-22 一般模式下的控制电路，内部时钟分频因子为 1

12.3.4.1 外部时钟源模式 1

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

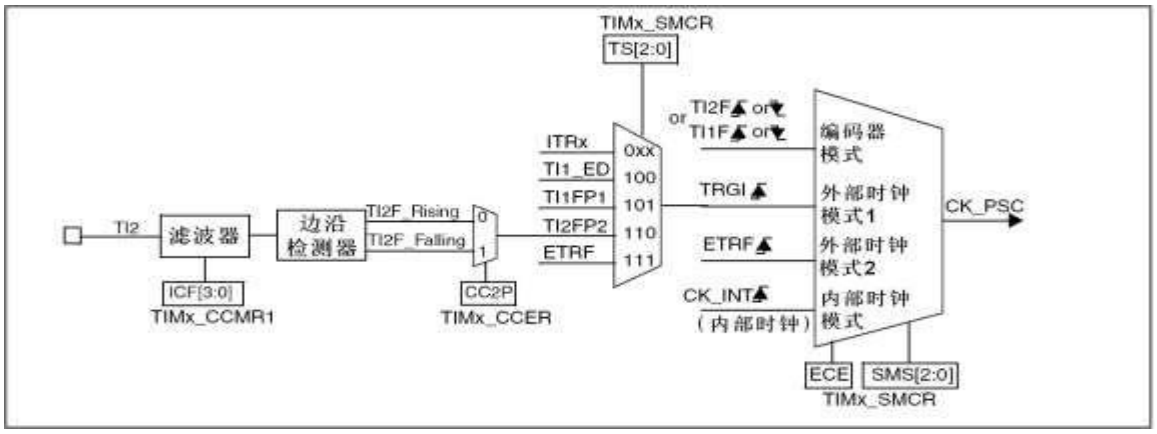


图 12-23 TI2 外部时钟连接例子

例如，要配置向上计数器在 TI2 输入端的上升沿计数，使用下列步骤:

1. 配置 EPWM_CCMR1 寄存器 CC2S = 01，配置信道 2 检测 TI2 输入的上升沿
2. 配置 EPWM_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)
3. 配置 EPWM_CCER 寄存器的 CC2P = 0，选定上升沿极性
4. 配置 EPWM_SMCR 寄存器的 SMS = 111，选择定时器外部时钟模式 1
5. 配置 EPWM_SMCR 寄存器中的 TS = 110，选定 TI2 作为触发输入源
6. 设置 EPWM_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，上升沿出现在 TI2，计数器计数一次，且 TIF 标志被设置。在 TI2 的上升沿和计数器实际时钟之间的延时，取决于在 TI2 输入端的重新同步电路。

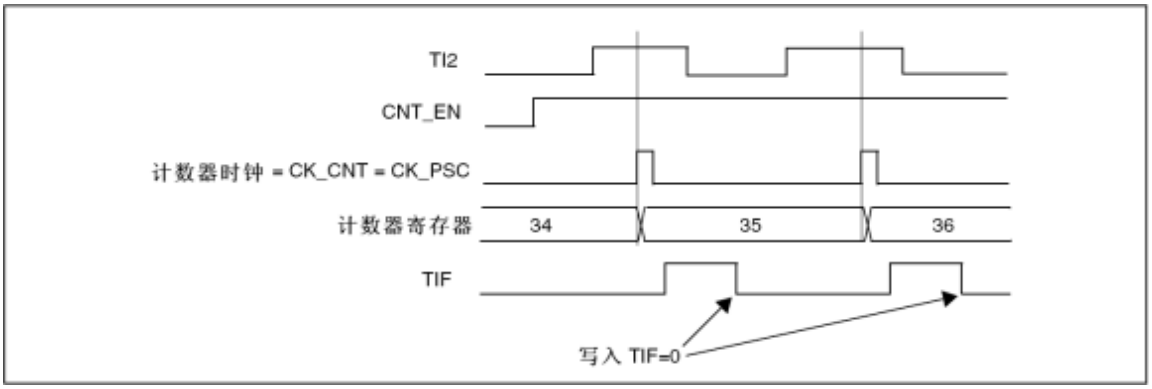


图 12-24 外部时钟模式 1 下的控制电路

12.3.4.2 外部时钟源模式 2

选定此模式的方法为: 令 EPWM_SMCR 寄存器中 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

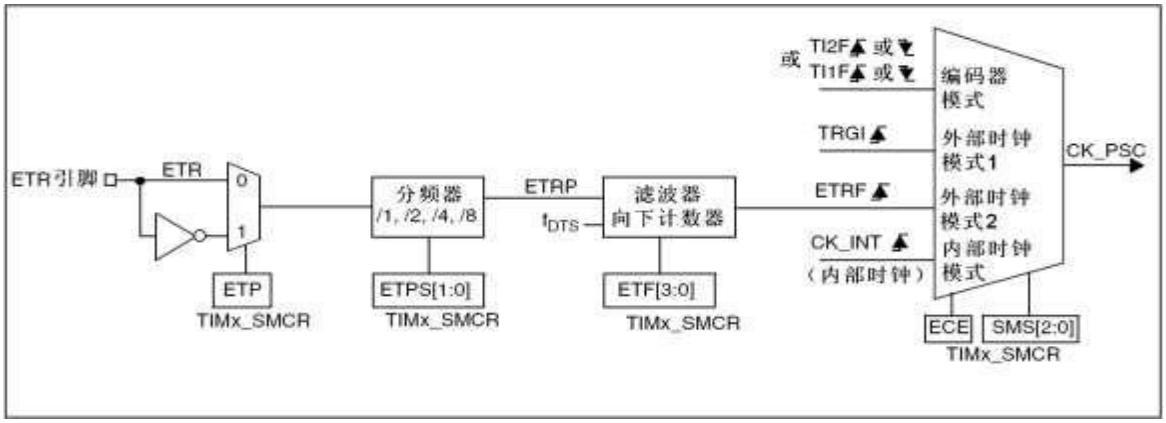


图 12-25 外部触发输入框图

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:

1. 本例中不需要滤波器，置 EPWM_SMCR 寄存器中的 ETF [3:0] = 0000；
2. 设置预分频器，置 EPWM_SMCR 寄存器中的 ETPS [1:0] = 01；
3. 选择 ETR 的上升沿检测，置 EPWM_SMCR 寄存器中的 ETP = 0；
4. 开启外部时钟模式 2，写 EPWM_SMCR 寄存器中的 ECE = 1；
5. 启动计数器，写 EPWM_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。

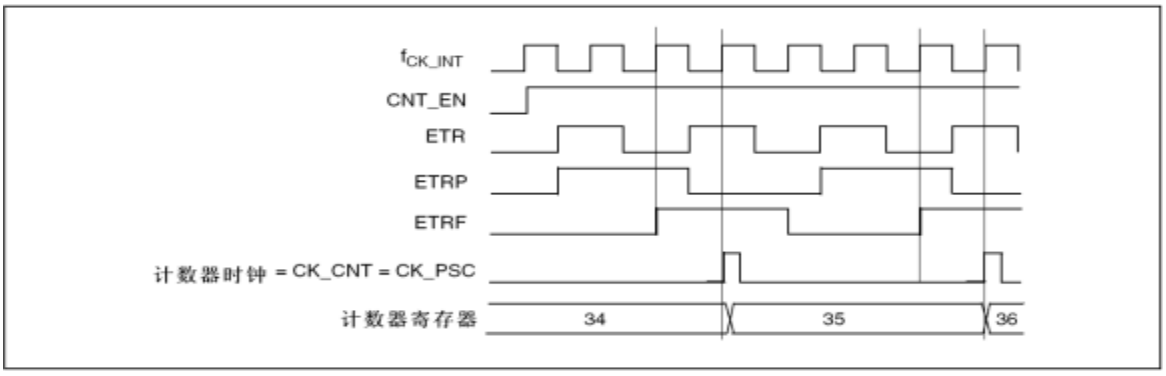


图 12-26 外部时钟模式 2 下的控制电路

12.3.5 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) ，捕获通道的来源由 SYSCFG_EPWM_CON_SEL 设定。

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 12-27 至图 12-30 是一个捕获/比较通道概览。

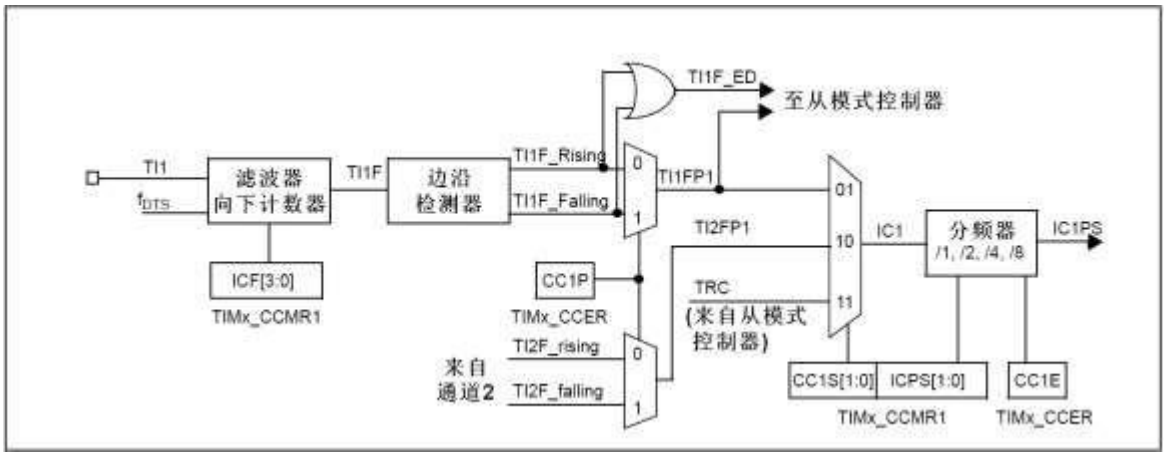


图 12-27 捕获/比较通道(如：通道 1 输入部分)

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。

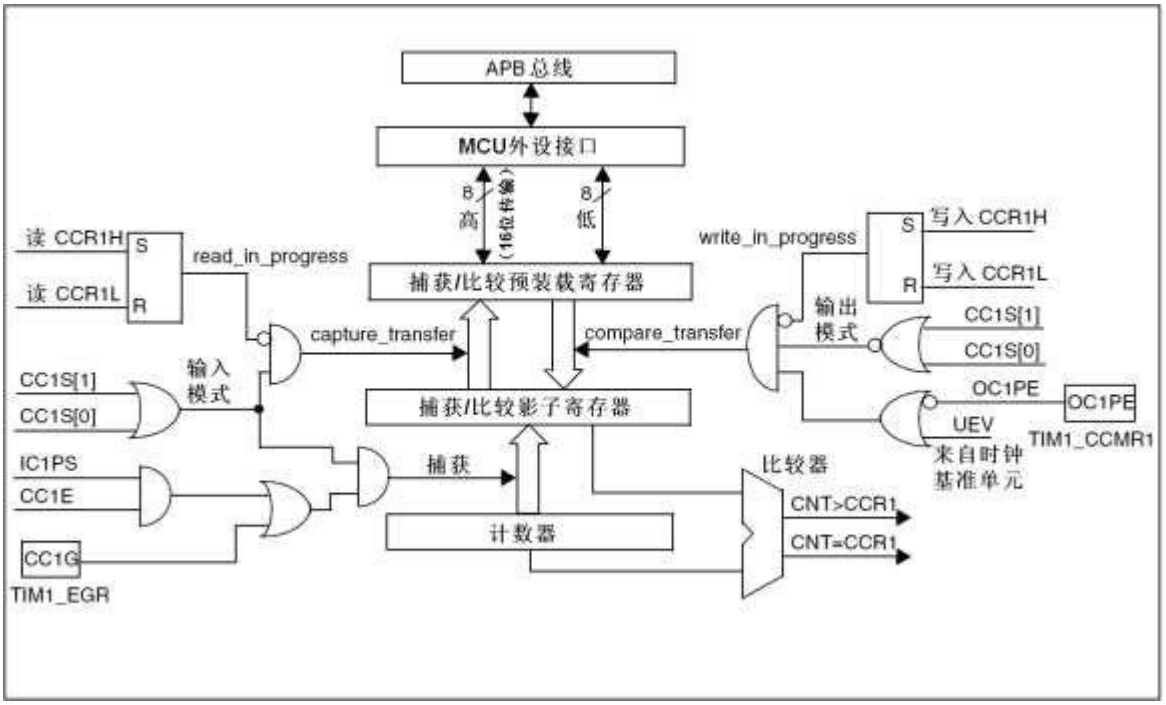


图 12-28 捕获/比较通道 1 的主电路

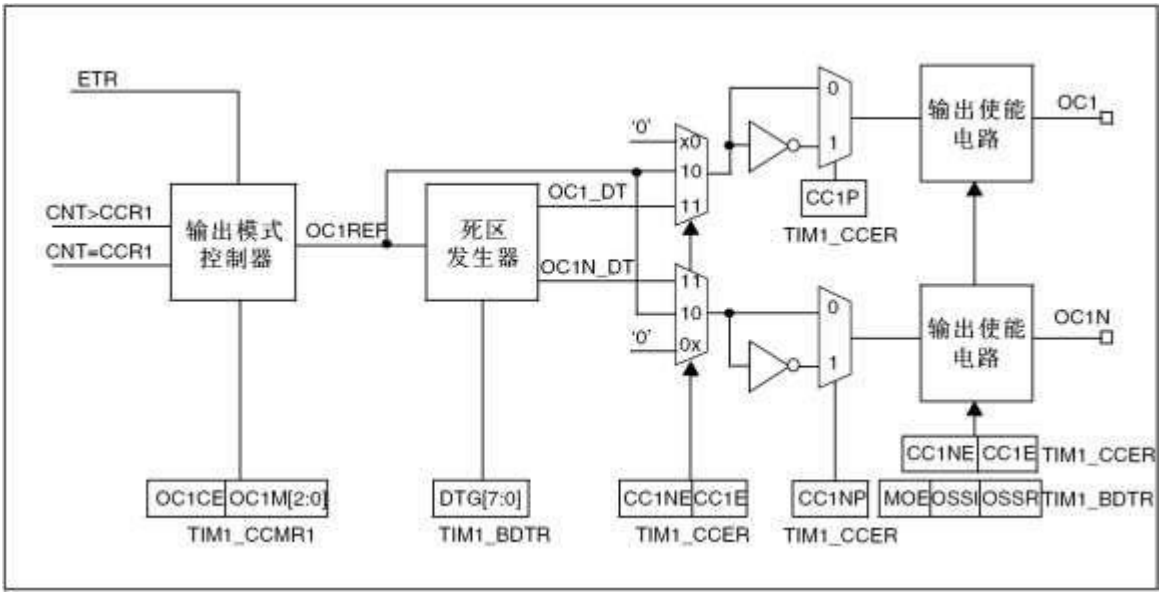


图 12-29 捕获/比较通道的输出部分(通道 1 至 3)

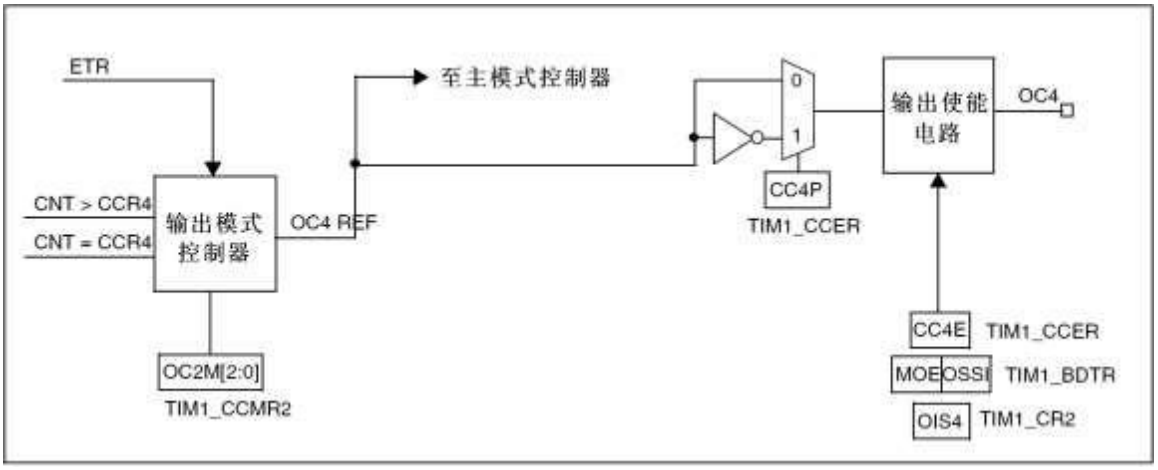


图 12-30 捕获/比较通道的输出部分(通道 4)

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。

©Copyright 2026, PADAUK Technology Co. Ltd

12.3.6 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(EPWM_CCRx)中。当发生捕获事件时，相应的 CCxIF 标志(EPWM_SR 寄存器)被置 1，如果开放了中断，则将产生中断。如果发生捕获事件时 CCxIF 标志已经为高，那么重复捕获标志 CCxOF(EPWM_SR 寄存器)被置 1。写 CCxIF=0 可清除 CCxIF，或读取存储在 EPWM_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 EPWM_CCR1 寄存器中，步骤如下:

- 选择有效输入端：EPWM_CCR1 必须连接到 TI1 输入，所以写入 EPWM_CCMR1 寄存器中的 CC1S=01，只要 CC1S 不为'00'，信道被配置为输入，并且 EPWM_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 EPWM_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 fDTS 频率)连续采样 8 次，以确认在TI1 上一次真实的边沿变换，即在 EPWM_CCMR1 寄存器中写入 IC1F = 0011。
- 选择 TI1 通道的有效转换边沿，在 EPWM_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EPWM_CCMR1 寄存器的IC1PS = 00)。
- 设置EPWM_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EPWM_DIER 寄存器中的 CC1IE 位允许相关中断请求。当发生一个输入捕获时:

- 产生有效的电平转换时，计数器的值被传送到 EPWM_CCR1 寄存器。
- CC1IF 标志被设置(中断标志)。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置 1。
- 如设置了 CC1IE 位，则会产生一个中断。

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

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

©Copyright 2026, PADAUK Technology Co. Ltd

12.3.7 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:

- 2 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(EPWM_CCR1 寄存器)和占空比(EPWM_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 EPWM_CCR1 的有效输入：置 EPWM_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 EPWM_CCR1 中和清除计数器)：置 CC1P = 0 (上升沿有效)。
- 选择 EPWM_CCR2 的有效输入：置 EPWM_CCMR1 寄存器的 CC2S = 10(选中TI1)。
- 选择 TI1FP2 的有效极性(捕获数据到EPWM_CCR2)：置 CC2P = 1(下降沿有效)。
- 选择有效的触发输入信号：置 EPWM_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EPWM_SMCR 中的SMS = 100。
- 使能捕获：置 EPWM_CCER 寄存器中CC1E = 1 且CC2E = 1。

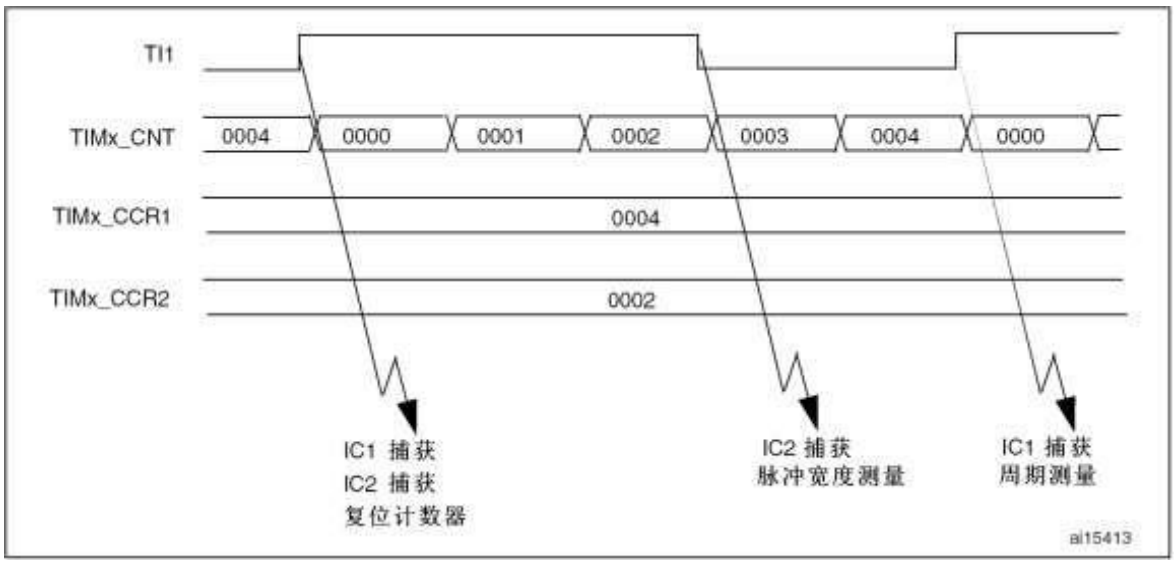


图 12-31 PWM 输入模式时序

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 EPWM_CH1/EPWM_CH2 信号。

12.3.8 强置输出模式

在输出模式 (EPWM_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 EPWM_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 EPWM_CCMRx 寄存器中的 OCxM = 100，可强置 OCxREF 信号为低。

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

12.3.9 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结束。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 (EPWM_CCMRx 寄存器中的 OCxM 位) 和输出极性 (EPWM_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 (EPWM_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽 (EPWM_DIER 寄存器中的 CCxIE 位)，则产生一个中断。

EPWM_CCMRx 中的 OCxPE 位选择 EPWM_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 EPWM_ARR 和 EPWM_CCRx 寄存器中。
3. 如果要产生一个中断请求，设置 CCxIE 位。

©Copyright 2026, PADAUK Technology Co. Ltd

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EPWM_CR1 寄存器的CEN 位启动计数器

EPWM_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 EPWM_CCRx 的影子寄存器只能在发生下一次更新事件时被更新)。

下图给出了一个例子。

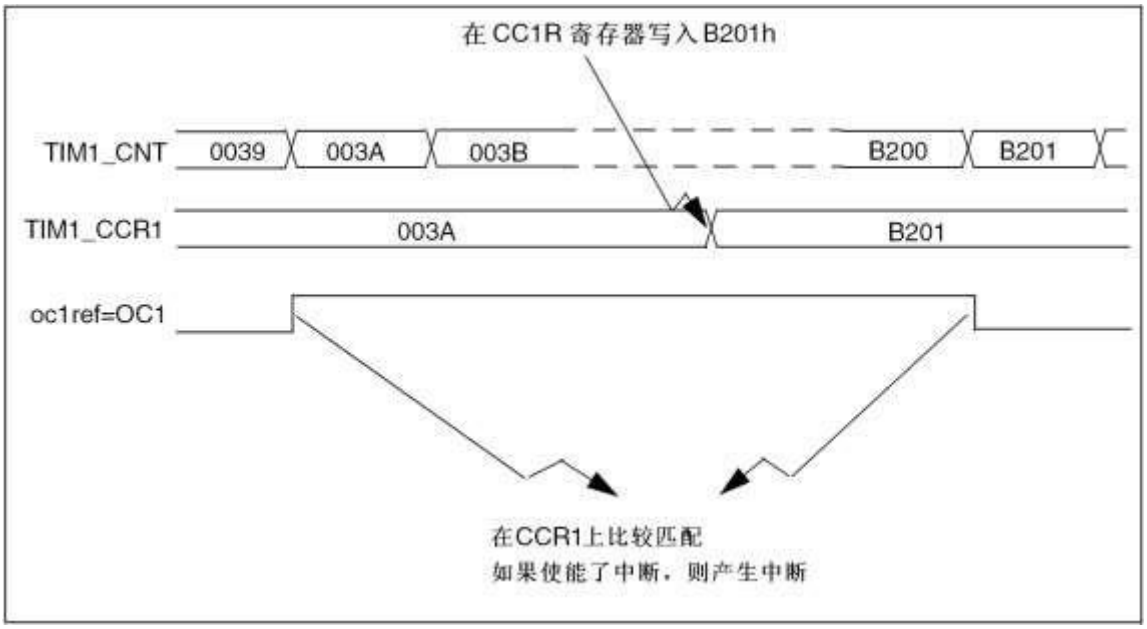


图 12-32 输出比较模式，翻转 OC1

12.3.10 PWM 输出模式

脉冲宽度调制模式可以产生一个由 EPWM_ARR 寄存器确定频率、由 EPWM_CCRx 寄存器确定占空比的信号。在 EPWM_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 EPWM_CCMRx 寄存器的 OCxPE 位使能相应的预装载寄存器，最后还要设置 EPWM_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 EPWM_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 EPWM_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(EPWM_CCER 和 EPWM_BDTR 寄存器中)CCxE、CCxNE、MOE、OSSI 和 OSSR 位的组合控制。详见 EPWM_CCER 寄存器的描述。

在 PWM 模式(模式 1 或模式 2)下，EPWM_CNT 和 EPWM_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 EPWM_CCRx ≤ EPWM_CNT 或者 EPWM_CNT ≤ EPWM_CCRx。

如果使能 EPWM_CR1.ASYMEN 开启 PWM 输出不对称模式, EPWM_CNT 下数将会以 EPWM_CCDRx 作为比较的基准。根据 EPWM_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。

©Copyright 2026, PADAUK Technology Co. Ltd

12.3.10.1 PWM 边沿对齐模式

向上计数配置

当 EPWM_CR1 寄存器中的 DIR 位为低的时候执行向上计数。

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

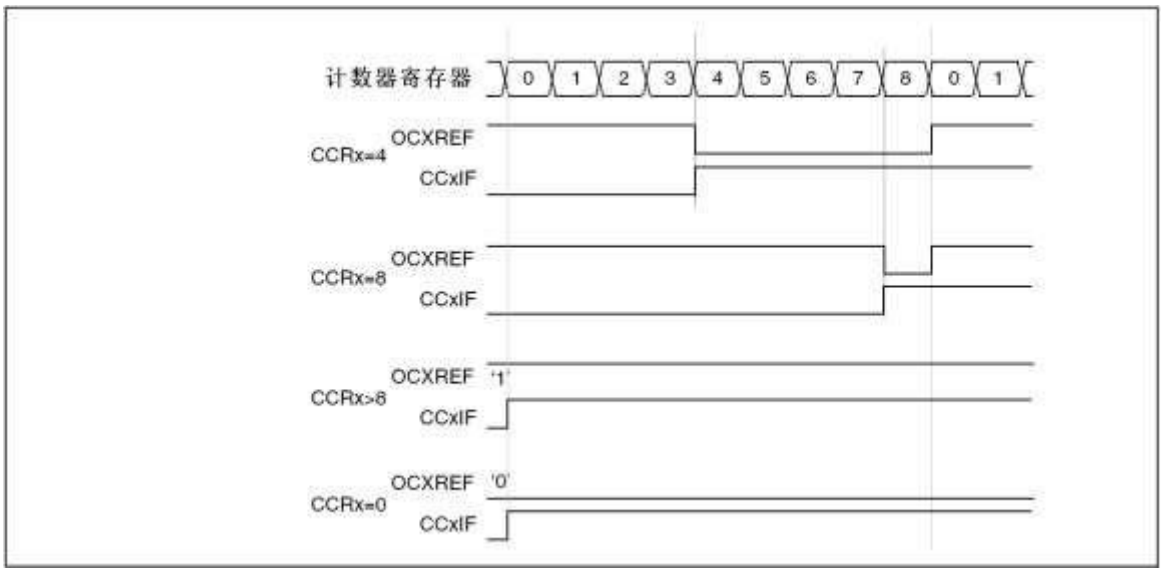


图 12-33 边沿对齐的 PWM 波形(ARR=8)

向下计数的配置

当 EPWM_CR1 寄存器的 DIR 位为高时执行向下计数。在 PWM 模式 1，当 EPWM_CNT > EPWM_CCRx 时参考信号 OCxREF 为低，否则为高。如果 EPWM_CCRx 中的比较值大于 EPWM_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。

12.3.10.2 PWM 中央对齐模式

对称式

当 EPWM_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。EPWM_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,设定 ASYMEN=0,为对称性 PWM 输出.

下图给出了一些中央对齐的 PWM 波形的例子

- EPWM_ARR = 8
- PWM 模式 1 , ASYMEN = 0,为 PWM 输出对称性
- EPWM_CR1 寄存器的CMS = 01，在中央对齐模式 1 下，当计数器向下计数时设置比较标志。

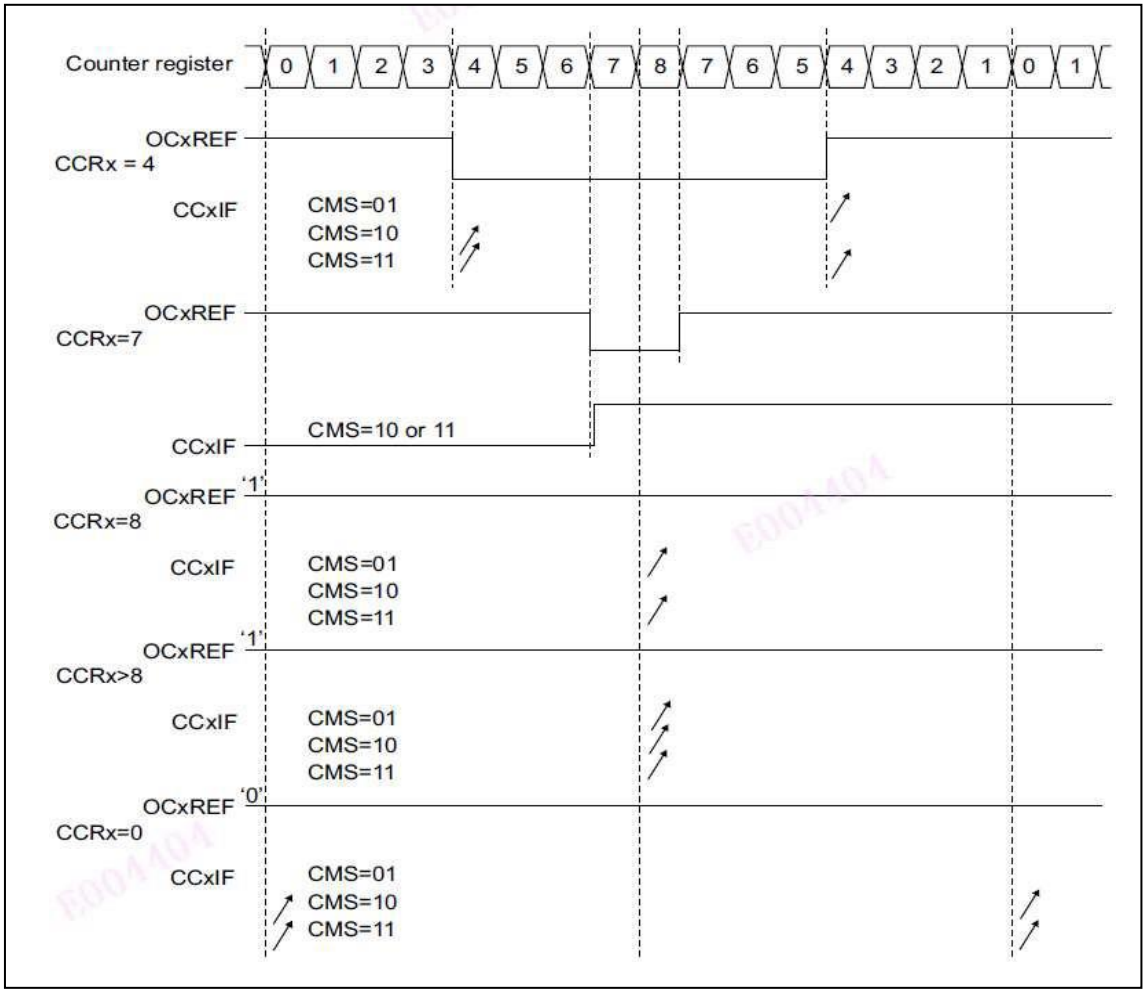


图 12-34 中央对齐的对称性 PWM 波形(ARR = 8)

使用中央对齐模式的提示:

- 进入中央对齐模式时，使用当前的向上/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 EPWM_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(EPWM_CNT > EPWM_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EPWM_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 EPWM_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。

不对称式

当 EPWM_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。EPWM_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,并设定 ASYMEN = 1,为不对称性 PWM 输出.

下图给出了一些中央对齐的 PWM 波形的例子

- EPWM_ARR = 900
- PWM 模式 1 , ASYMEN=1, PWM 输出为不对称
- EPWM_CCRn = 300, EPWM_CCDR1 = 600

EPWM_CR1 寄存器的 CMS = 10，在中央对齐模式 2 下，当计数器向上计数时设置比较标志 CCnIF 当计数器向下计数时设置比较标志 CCDnIF.

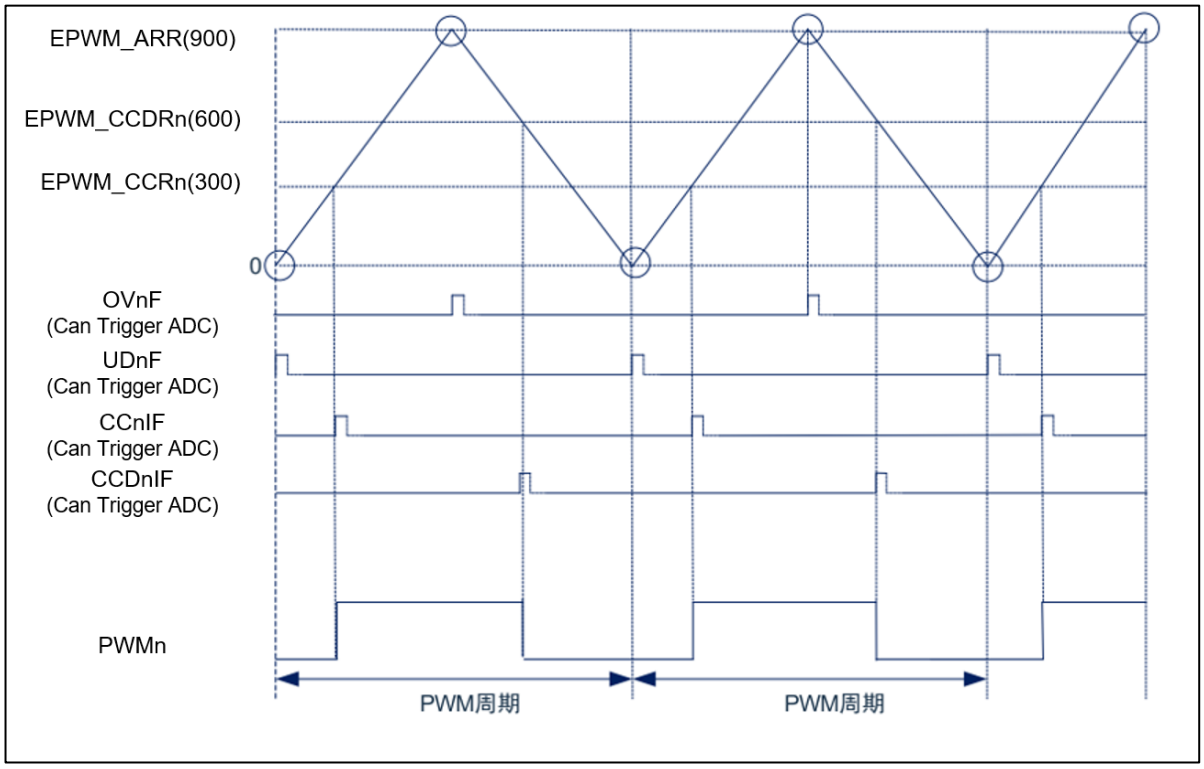


图 12-35 中央对齐的不对称 PWM 波形(EPWM_ARR=900)

12.3.11 互补输出和死区插入

高级控制定时器 (EPWM) 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 EPWM_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：EPWM_CCER 寄存器的 CCxE 和 CCxNE 位，EPWM_BDTR 和 EPWM_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位，详见表 12-4 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)

- DTAE = 0/1: (带死区对称/带死区不对称)

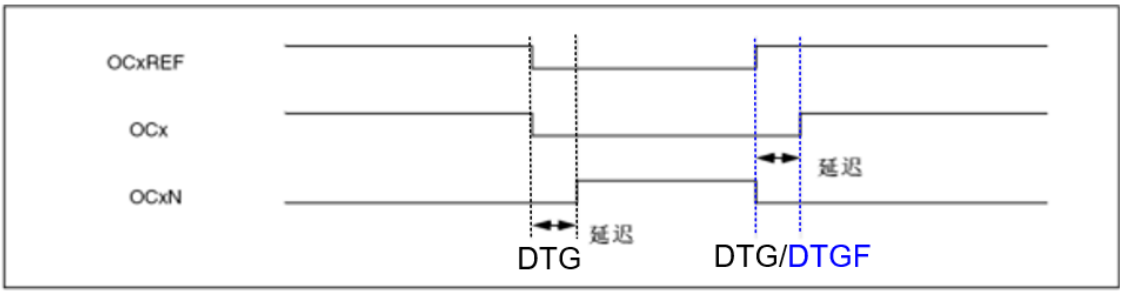


图 12-36 带死区插入的互补输出

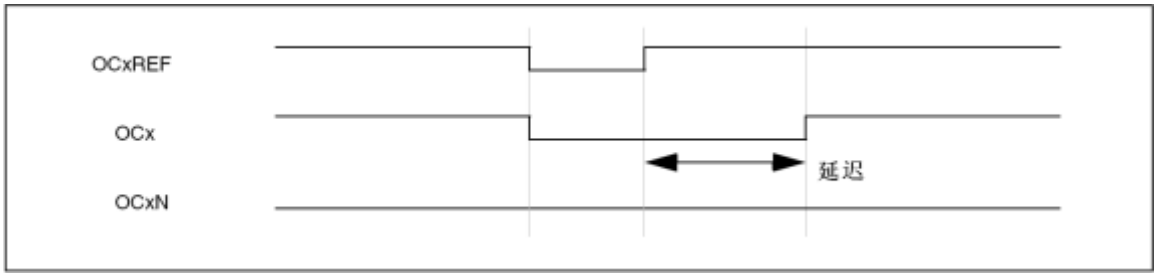


图 12-37 死区波形延迟大于负脉冲

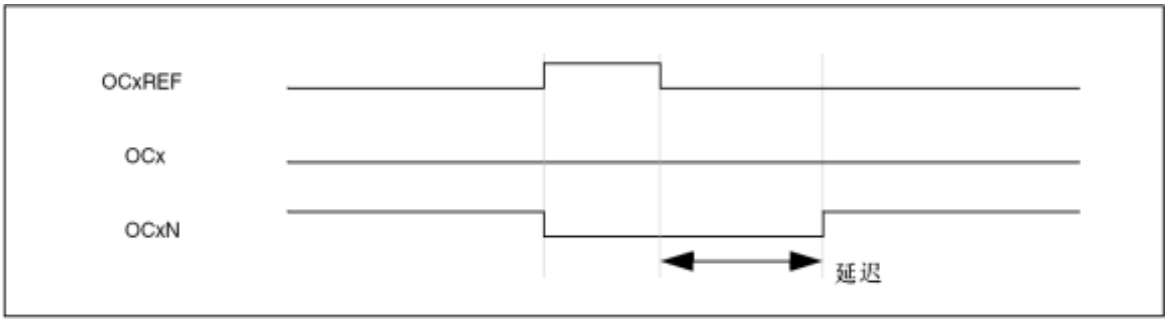


图 12-38 死区波形延迟大于正脉冲

每一个通道的死区延时都是相同的，是由 EPWM_BDTR 寄存器中的 DTG 位编程配置。

详见 EPWM 刹车和死区寄存器(EPWM_BDTR)中的延时计算。

12.3.11.1 复位向 OCxREF 到 OCx 或 OCxN

在输出模式下(强置、输出比较或 PWM)，通过配置 EPWM_CCER 寄存器的 CCxE 和 CCxNE 位，OCxREF 可以被复位向到 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。

©Copyright 2026, PADAUK Technology Co. Ltd

12.3.12 使用刹车功能

当使用刹车功能时，依据相应的控制位(EPWM_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位，EPWM_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 12-4 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时钟失败事件。时钟失败事件由复位时钟控制器中的时钟安全系统产生。系统复位后，刹车电路被禁止，MOE 位为低。设置 EPWM_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。当写入 BKE 和 BKP 位时，在真正写入之前会有 1 个 APB 时钟周期的延迟，因此需要等待一个 APB 时钟周期之后，才能正确地读回写入的位。

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 EPWM_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 EPWM_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 EPWM_DIER 寄存器中的 BIE 位，当刹车状态标志(EPWM_SR 寄存器中的 BIF 位)为'1'时，则产生一个中断。
- 如果设置了 EPWM_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 EPWM_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 EPWM_BDTR 寄存器中的 LOCK 位，从三级保护中选择一种，参看 12.5-18 节 EPWM 刹车和死区寄存器(EPWM_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。

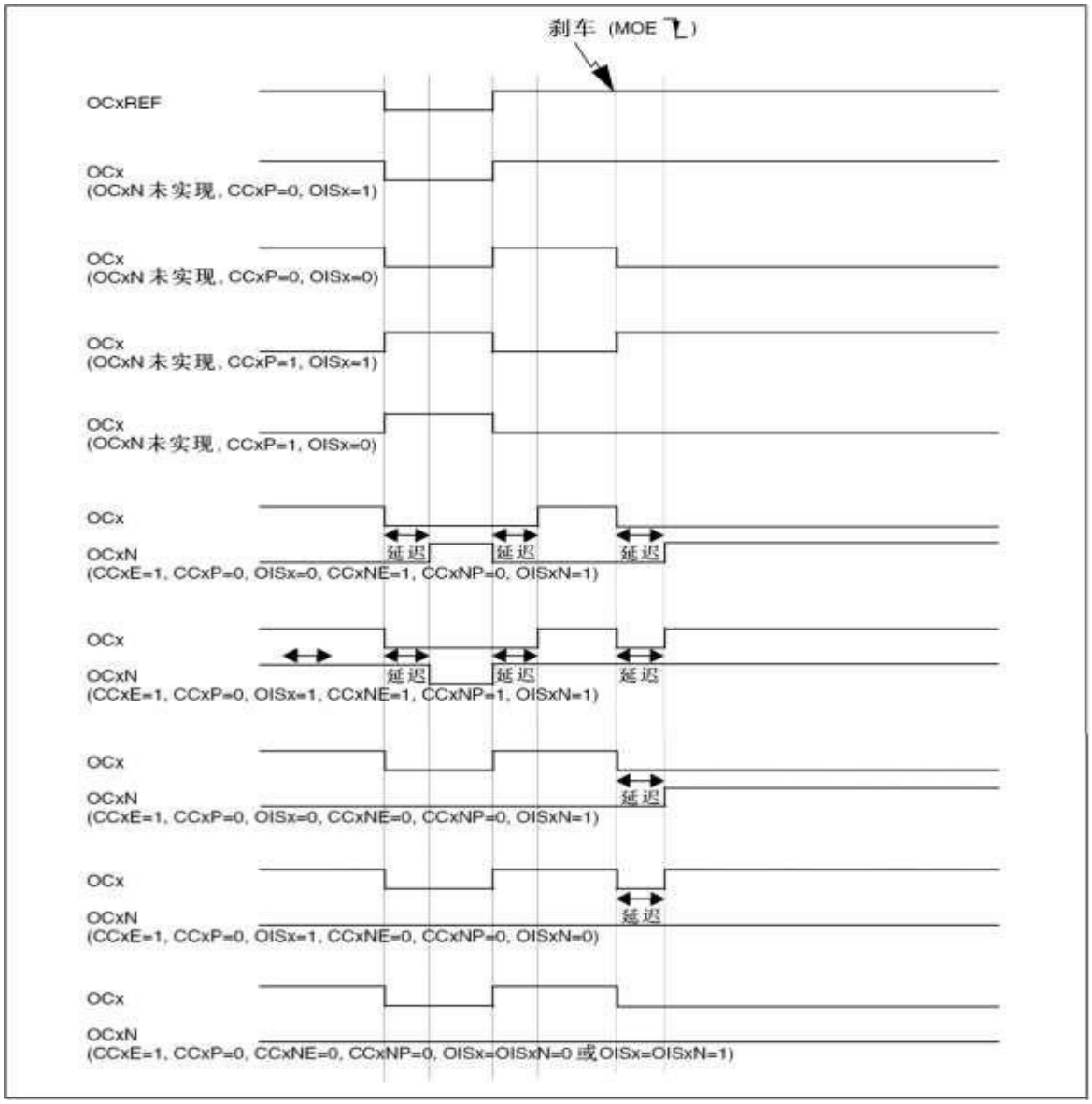


图 12-39 响应刹车的输出

12.3.13 在外部事件时清除 OCxREF 信号

对于一个给定的通道，设置 EPWM_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: EPWM_SMCR 寄存器中的ETPS[1:0] = 00。
2. 必须禁止外部时钟模式, EPWM_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 EPWM 被置于 PWM 模式。

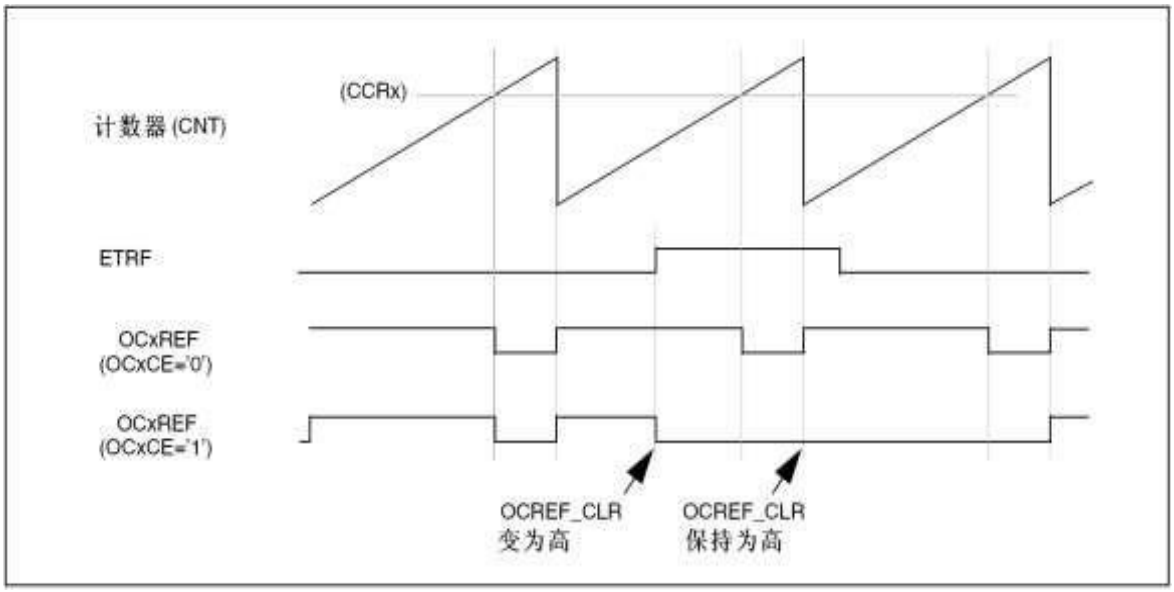


图 12-40 清除 EPWM 的 OCxREF

12.3.14 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更改所有信道的配置。COM 可以通过设置 EPWM_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(EPWM_SR 寄存器中的 COMIF 位)，这时如果已设置了 EPWM_DIER 寄存器的 COMIE 位，则产生一个中断。下图显示当发生 COM 事件时，三种不同配置下 OCx 和 OCxN 输出。

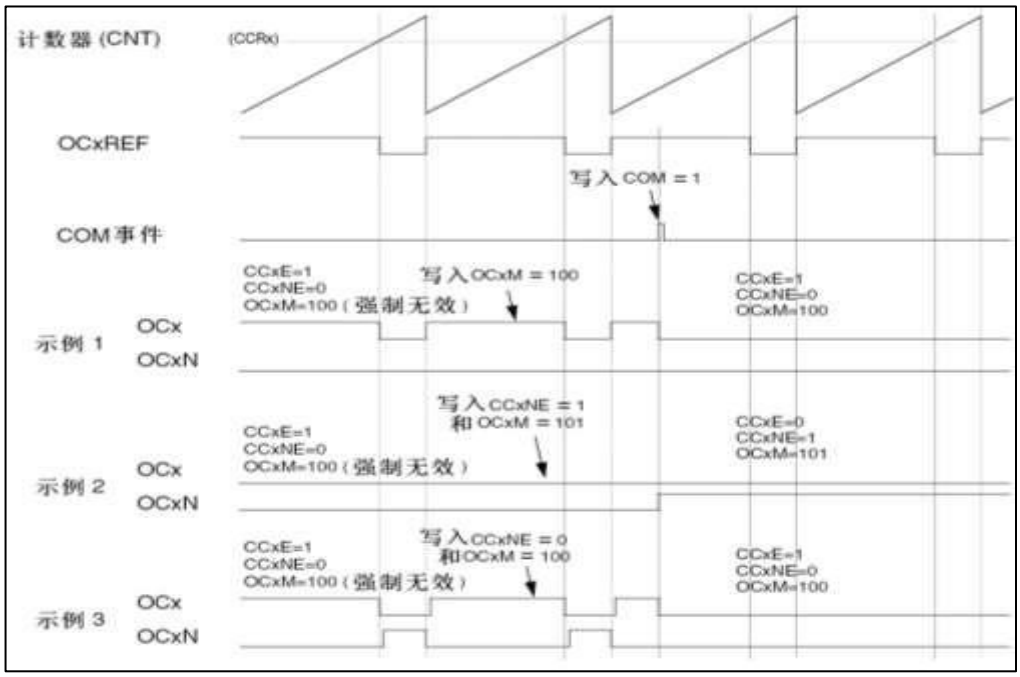


图 12-41 产生六步 PWM，使用 COM 的例子(OSSR=1)

12.3.15 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过程控制的脉冲。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 EPWM_CR1 寄存器中的 OPM 位将选择单脉冲模式，这样可以让计数器自动地在产生下一个更新事件 UEV 时停止。仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前(当定时器正在等待触发)，必须如下配置:

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

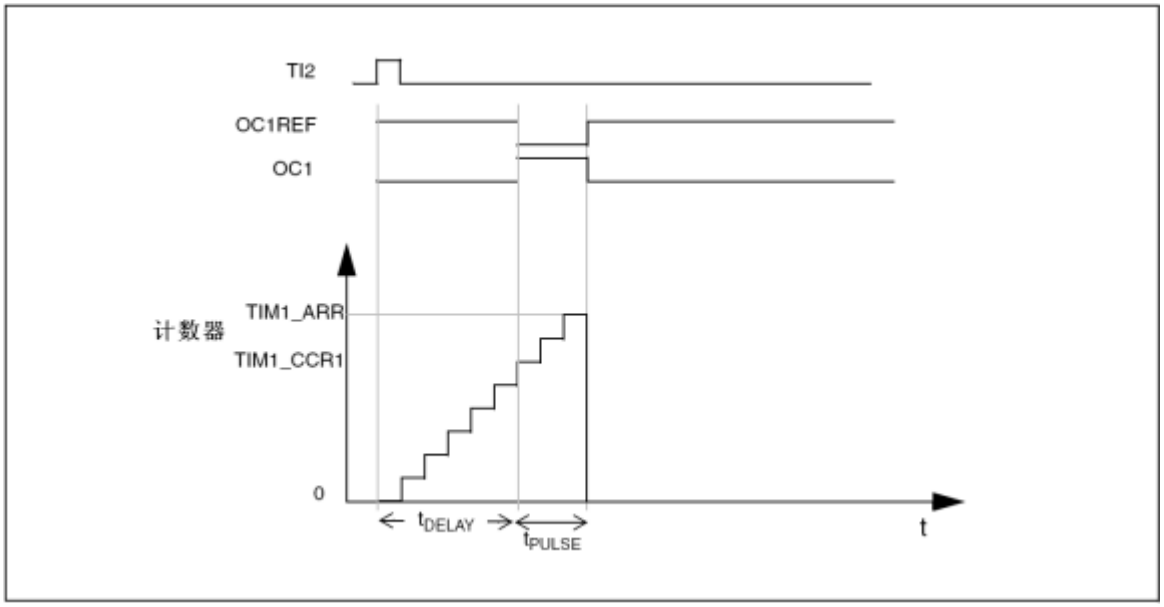


图 12-42 单脉冲模式的例子

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

假定 TI2FP2 作为触发 1:

- 置EPWM_CCMR1 寄存器中的 CC2S = 01，把 TI2FP2 映像到 TI2。
- 置EPWM_CCER 寄存器中的 CC2P = 0，使 TI2FP2 能够检测上升沿。
- 置EPWM_SMCR 寄存器中的 TS = 110，TI2FP2 作为从模式控制器的触发(TRGI)。
- 置EPWM_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 EPWM_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(EPWM_ARR - EPWM_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 EPWM_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 EPWM_CCMR1 中的OC1PE = 1 和 EPWM_CR1 寄存器中的ARPE；然后在 EPWM_CCR1 寄存器中填写比较值，在 EPWM_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，EPWM_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 EPWM_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。

12.3.15.1 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 EPWM_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 只在信道配置为 PWM1 和 PWM2 模式时起作用。

12.3.16 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 EPWM_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 EPWM_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 12-1，假定计数器已经启动(EPWM_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 EPWM_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 EPWM_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 EPWM_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。

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

表 12-1 计数方向与编码器信号的关系

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’(EPWM_CCMR1 寄存器，IC1FP1 映射到TI1)
- CC2S =‘01’(EPWM_CCMR2 寄存器，IC2FP2 映射到TI2)
- CC1P =‘0’(EPWM_CCER 寄存器，IC1FP1 不反相，IC1FP1=TI1)
- CC2P =‘0’(EPWM_CCER 寄存器，IC2FP2 不反相，IC2FP2=TI2)
- SMS =‘011’(EPWM_SMCR 寄存器，所有的输入均在上升沿和下降沿有效)
- CEN=‘1’(EPWM_CR1 寄存器，计数器使能)

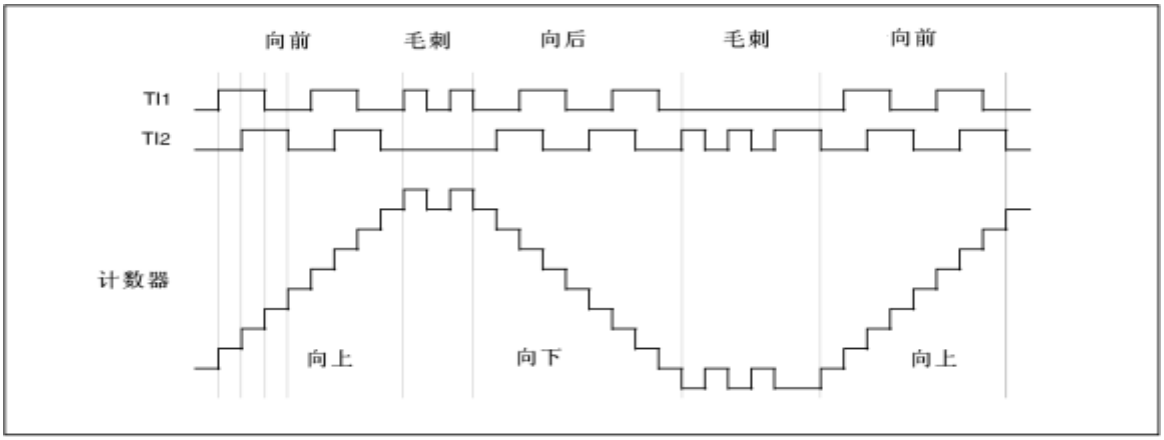


图 12-43 编码器模式下的计数器操作实例

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

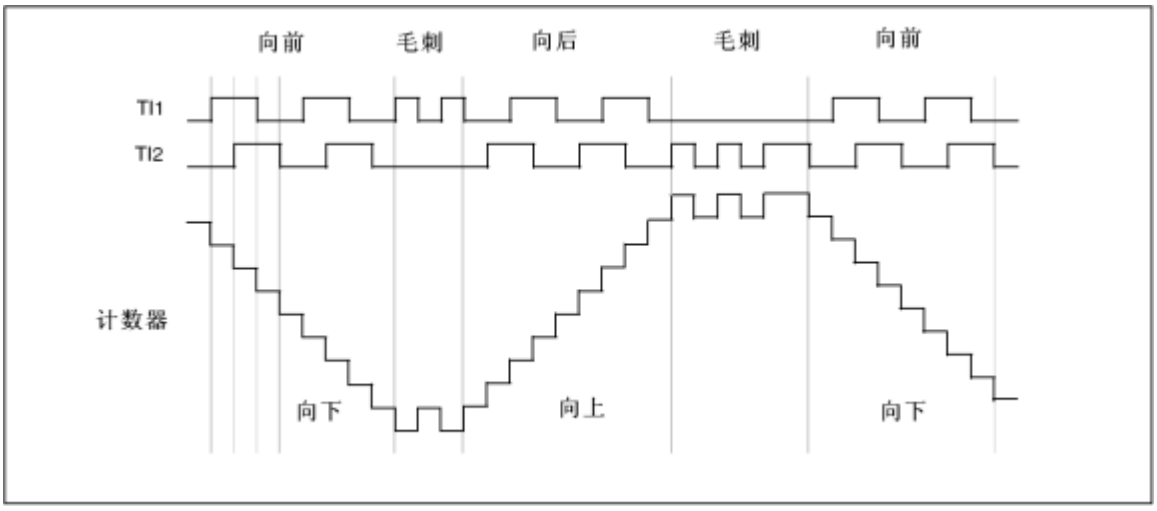


图 12-44 IC1FP1 反相的编码器接口模式实例

当定时器配置成编码器接口模式时，提供传感器当前位置的信息。使用第二个配置在捕获模式的定时器，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)。

12.3.17 定时器输入异或功能

EPWM_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 EPWM_CH1、EPWM_CH2 和 EPWM_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。

©Copyright 2026, PADAUK Technology Co. Ltd

12.3.18 EPWM 定时器和外部触发的同步

EPWM 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。

12.3.18.1 从模式: 复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 EPWM_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 (EPWM_ARR ， EPWM_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:

- 配置信道 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 EPWM_CCMR1 寄存器中CC1S = 01。置 EPWM_CCER 寄存器中 CC1P = 0 以确定极性(只检测上升沿)。
- 置 EPWM_SMCR 寄存器中 SMS = 100，配置定时器为复位模式；置 EPWM_SMCR 寄存器中 TS=101，选择TI1 作为输入源。
- 置 EPWM_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(EPWM_SR 寄存器中的 TIF 位)被设置，根据 EPWM_DIER 寄存器中 TIE(中断使能)位的设置，产生一个中断请求。

下图显示当自动重装载寄存器 EPWM_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。

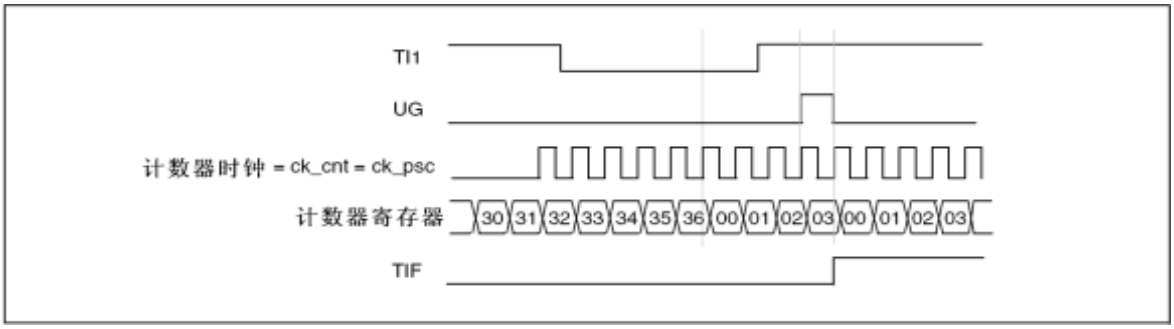


图 12-45 复位模式下的控制电路

12.3.18.2 从模式: 门控模式

按照选中的输入端电平使能计数器。在如下的例子中，计数器只在 TI1 为低时向上计数:

- 配置信道 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 EPWM_CCMR1 寄存器中CC1S = 01。置 EPWM_CCER 寄存器中CC1P = 1 以确定极性(只检测低电平)。
- 置 EPWM_SMCR 寄存器中 SMS = 101，配置定时器为门控模式；置 EPWM_SMCR 寄存器中 TS = 101，选择TI1 作为输入源。
- 置 EPWM_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 EPWM_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。

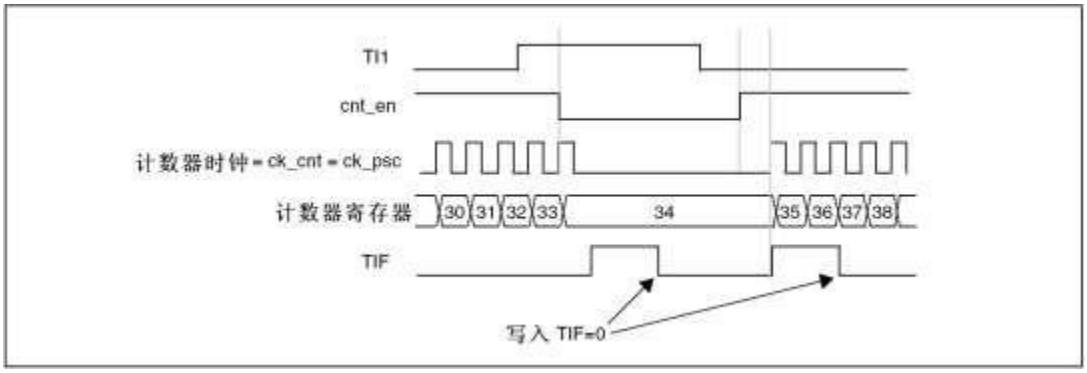


图 12-46 门控模式下的控制电路

12.3.18.3 从模式: 触发模式

输入端上选中的事件使能计数器。在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数:

- 配置信道 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 EPWM_CCMR1 寄存器中 CC2S=01。置 EPWM_CCER 寄存器中 CC2P=1 以确定极性(只检测低电平)。
- 置 EPWM_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 EPWM_SMCR 寄存器中 TS=110，选择 TI2 作为输入源。

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

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

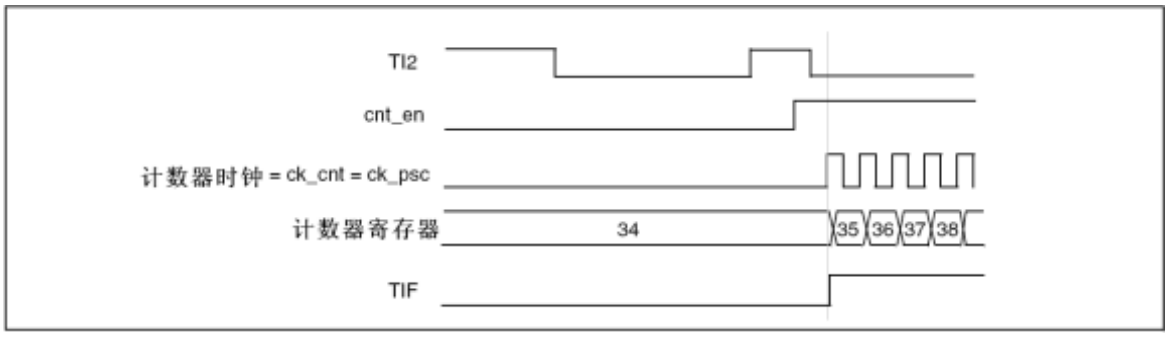


图 12-47 触发器模式下的控制电路

12.3.18.4 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 EPWM_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 EPWM_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. 按如下配置信道 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 EPWM_CCMR1 寄存器中 CC1S = 01，选择输入捕获源
   - 置 EPWM_CCER 寄存器中 CC1P = 0 以确定极性(只检测上升沿)
3. 置 EPWM_SMCR 寄存器中 SMS=110，配置定时器为触发模式。置 EPWM_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。

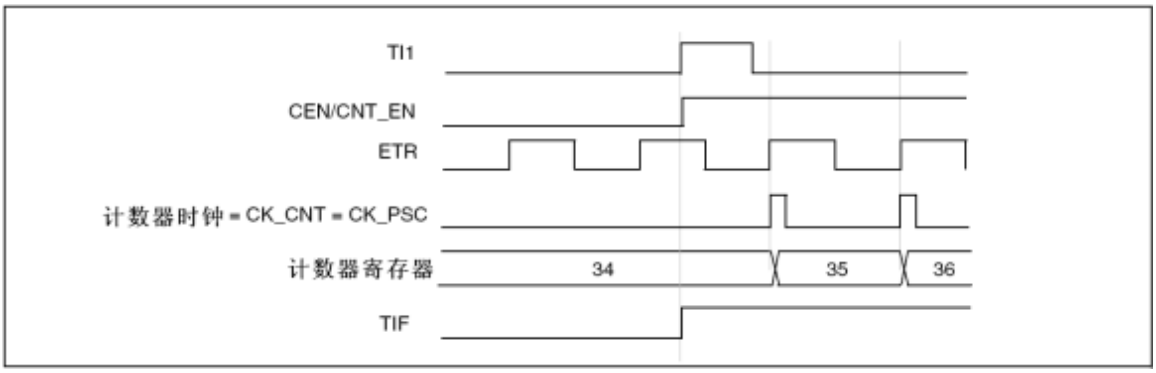


图 12-48 外部时钟模式 2 + 触发模式下的控制电路

12.3.18.5 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。

12.3.18.6 调试模式

当微控制器进入调试模式时(CPU 核心停止)，根据 DBG 模块中 DBG_EPWM_STOP 的设置， EPWM 计数器可以或者继续正常操作，或者停止。

©Copyright 2026, PADAUK Technology Co. Ltd

12.4 EPWM 寄存器列表

可以用字(32 位)的方式操作这些外设寄存器，EPWM 基地址 0x4000 C000。

地址	名称	描述
0x4000C000	EPWM_CR1	EPWM 控制寄存器 1
0x4000C004	EPWM_CR2	EPWM 控制寄存器 2
0x4000C008	EPWM_SMCR	EPWM 从模式控制寄存器
0x4000C00C	EPWM_IER	EPWM 中断使能寄存器
0x4000C010	EPWM_SR	EPWM 状态寄存器
0x4000C014	EPWM_EGR	EPWM 事件产生寄存器
0x4000C018	EPWM_CCMR1	EPWM 捕获/比较模式寄存器 1
0x4000C01C	EPWM_CCMR2	EPWM 捕获/比较模式寄存器 2
0x4000C020	EPWM_CCER	EPWM 捕获/比较使能寄存器
0x4000C024	EPWM_CNT	EPWM 计数器
0x4000C028	EPWM_PSC	EPWM 预分频器
0x4000C02C	EPWM_ARR	EPWM 自动重装载寄存器
0x4000C030	EPWM_RCR	EPWM 重复计数寄存器
0x4000C034	EPWM_CCR1	EPWM 捕获/比较寄存器 1
0x4000C038	EPWM_CCR2	EPWM 捕获/比较寄存器 2
0x4000C03C	EPWM_CCR3	EPWM 捕获/比较寄存器 3
0x4000C040	EPWM_CCR4	EPWM 捕获/比较寄存器 4
0x4000C044	EPWM_BDTR	EPWM 刹车和死区寄存器
0x4000C050	EPWM_CCDR1	EPWM 计数器向下比较寄存器 1
0x4000C054	EPWM_CCDR2	EPWM 计数器向下比较寄存器 2
0x4000C058	EPWM_CCDR3	EPWM 计数器向下比较寄存器 3
0x4000C05C	EPWM_CCDR4	EPWM 计数器向下比较寄存器 4

表 12-2 EPWM 寄存器列表

©Copyright 2026, PADAUK Technology Co. Ltd

12.5 EPWM 寄存器描述

12.5.1 EPWM 控制寄存器 1 (EPWM_CR1)

Bit	Name	R/W	Reset	Description
31:11	-	R	0x0	保留
10	ASYMEN	R/W	0x0	EPWM 中心对齐方式下不对称计数使能 0: 对称计数使能 1: 不对称计数使能
9:8	CKD	R/W	0x0	时钟分频因子 (Clock division) 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。 00: tDTS = tCK_INT 01: tDTS = 2 x tCK_INT 10: tDTS = 4 x tCK_INT 11: 保留，不要使用这个配置
7	ARPE	R/W	0x0	自动重装载预装载允许位(Auto-reload preload enable) 0: EPWM_ARR 寄存器没有缓冲 1: EPWM_ARR 寄存器被装入缓冲器
6:5	CMS	R/W	0x0	选择中央对齐模式 (Center-aligned mode selection) 00: 边沿对齐模式。 计数器依据方向位(DIR)向上或向下计数。 01: 中央对齐模式 1。 计数器交替地向上和向下计数。配置为输出的信道(EPWM_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。 10: 中央对齐模式 2。 计数器交替地向上和向下计数。配置为输出的信道(EPWM_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。 11: 中央对齐模式 3。 计数器交替地向上和向下计数。配置为输出的信道(EPWM_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	R/W	0x0	方向 (Direction) 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	R/W	0x0	单脉冲模式(One pulse mode) 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	R/W	0x0	更新请求源(Update request source) 软件通过该位选择 UEV 事件的源 0: 如果置 0，则下述任一事件产生更新中断: 计数器溢出/下溢 设置 UG 位 从模式控制器产生的更新 1: 如果置 1，则只有计数器溢出/下溢才产生更新中断。
1	UDIS	R/W	0x0	禁止更新(Update disable) 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	R/W	0x0	使能计数器(Counter enable) 0: 禁止计数器； 1: 使能计数器。 注: 在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。 触发模式可以自动地通过硬件设置 CEN 位。

12.5.2 EPWM 控制寄存器 2 (EPWM_CR2)

Bit	Name	R/W	Reset	Description
31:15	-	-	0x0	保留
14	OIS4	R/W	0x0	输出空闲状态 4 (OC4 输出)。参见 OIS1 位。
13	OIS3N	R/W	0x0	输出空闲状态 3 (OC3N 输出)。参见 OIS1N 位。
12	OIS3	R/W	0x0	输出空闲状态 3 (OC3 输出)。参见 OIS1 位。
11	OIS2N	R/W	0x0	输出空闲状态 2 (OC2N 输出)。参见 OIS1N 位。
10	OIS2	R/W	0x0	输出空闲状态 2 (OC2 输出)。参见 OIS1 位。
9	OIS1N	R/W	0x0	输出空闲状态 1(OC1N 输出)(Output Idle state 1) 0: 当 MOE = 0 时，死区后 OC1N = 0； 1: 当 MOE = 0 时，死区后 OC1N = 1。 注: 已经设置了 LOCK(EPWM_BKR 寄存器)级别 1、2 或 3 后，该位不能被修改。
8	OIS1	R/W	0x0	输出空闲状态 1(OC1 输出)(Output Idle state 1) 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(EPWM_BKR 寄存器)级别 1、2 或 3 后，该位不能被修改。
7	TI1S	R/W	0x0	TI1 选择(TI1 selection) 0: EPWM_CH1 引脚连到 TI1 输入； 1: EPWM_CH1、EPWM_CH2 和 EPWM_CH3 引脚经异或后连到 TI1 输入。

6:4	MMS	R/W	0x0	主模式选择(Master mode selection) 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。 可能的组合如下: 000: 复位 – EPWM_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式)，则 TRGO 上的信号相对实际的复位会有一个延迟。 001: 使能– 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。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 EPWM_SMCR 寄存器中 MSM 位的描述)。 010: 更新 – 更新事件被选为触发输入(TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 011: 比较脉冲 – 在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)。 100: 比较 – OC1REF 信号被用于作为触发输出(TRGO)。 101: 比较 – OC2REF 信号被用于作为触发输出(TRGO)。 110: 比较 – OC3REF 信号被用于作为触发输出(TRGO)。 111: 比较 – OC4REF 信号被用于作为触发输出(TRGO)。
3	-	R	0x0	保留
2	CCUS	R/W	0x0	捕获 / 比较控制更新选择 (Capture / compare control update selection) 0: 如果捕获/比较控制位是预装载的 (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	-	R	0x0	保留
0	CCPC	R/W	0x0	捕获 / 比较预装载控制位 (Capture / compare preloaded control) 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。

12.5.3 EPWM 从模式控制寄存器(EPWM_SMCR)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15	ETP	R/W	0x0	外部触发极性 (External trigger polarity) 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	R/W	0x0	外部时钟使能位(External clock enable) 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	R/W	0x0	外部触发预分频(External trigger prescaler) 外部触发信号 ETRP 的频率必须最多是 EPWMCLK 频率的 1/4。 当输入较快的外部时钟时，可以使用预分频降低 ETRP 的频率。 00: 关闭预分频 01: ETRP 频率除以 2 10: ETRP 频率除以 4 11: ETRP 频率除以 8

11:8	ETF	R/W	0x0	外部触发滤波(External trigger filter) 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。 0000: 无滤波器，以 fDTS 采样 0001: 采样频率 fSAMPLING = fCK_INT，N = 2 0010: 采样频率 fSAMPLING = fCK_INT，N = 4 0011: 采样频率 fSAMPLING = fCK_INT，N = 8 0100: 采样频率 fSAMPLING = fDTS/2，N = 6 0101: 采样频率 fSAMPLING = fDTS/2，N = 8 0110: 采样频率 fSAMPLING = fDTS/4，N = 6 0111: 采样频率 fSAMPLING = fDTS/4，N = 8 1000: 采样频率 fSAMPLING = fDTS/8，N = 6 1001: 采样频率 fSAMPLING = fDTS/8，N = 8 1010: 采样频率 fSAMPLING = fDTS /16，N = 5 1011: 采样频率 fSAMPLING = fDTS /16，N = 6 1100: 采样频率 fSAMPLING = fDTS /16，N = 8 1101: 采样频率 fSAMPLING = fDTS /32，N = 5 1110: 采样频率 fSAMPLING = fDTS /32，N = 6 1111: 采样频率 fSAMPLING = fDTS /32，N = 8
7	MSM	R/W	0x0	主/从模式(Master/slave mode) 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	R/W	0x0	触发选择(Trigger selection) 这 3 位选择用于同步计数器的触发输入。 000: 内部触发 0(ITR0) 001: 内部触发 1(ITR1) 010: 内部触发 2(ITR2) 011: 内部触发 3(ITR3) 100: TI1 的边沿检测器(TI1F_ED) 101: 滤波后的定时器输入 1(TI1FP1) 110: 滤波后的定时器输入 2(TI2FP2)

				111: 外部触发输入(ETRF) 更多有关 ITRx 的细节，参见表 12-3。 注: 这些位只能在未用到(如 SMS = 000)时被改变，以避免在改变时产生错误的边沿检测。
3	-	R	0x0	保留
2:0	SMS	R/W	0x0	从模式选择(Slave mode selection) 当选择了外部信号，触发信号(TRGI)的有效边沿与选中的外部输入极性相关(见输入控制寄存器和控制寄存器的说明) 000: 关闭从模式 – 如果 CEN = 1，则预分频器直接由内部时钟驱动。 001: 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。 010: 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。 011: 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。 100: 复位模式 – 选中的触发输入(TRGI)的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。 101: 门控模式 – 当触发输入(TRGI)为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止(但不复位)。计数器的启动和停止都是受控的。 110: 触发模式 – 计数器在触发输入 TRGI 的上升沿启动(但不复位)， 只有计数器的启动是受控的。 111: 外部时钟模式 1 – 选中的触发输入(TRGI)的上升沿驱动计数器。 注：如果 TI1F_EN 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。

从定时器	ITR0 (TS=000)	ITR1 (TS=001)	ITR2 (TS=010)	ITR3 (TS=011)
EPWM	Tim2_trgo	irq_timer0	irq_timer1	irq_lptimer

表 12-3 EPWM 内部触发连接

12.5.4 EPWM 中断使能寄存器 (EPWM_IER)

Bit	Name	R/W	Reset	Description
31:14	-	R	0x0	保留
13	CCD4IE	R/W	0x0	允许比较 4 中断 (ASYMEN = 1 和 向下计数模式有效) 0: 禁止捕获/比较 1 中断； 1: 允许捕获/比较 1 中断。
12	UDIE	R/W	0x0	0: 禁止上溢 中断； 1: 允许上溢 中断
11	OVIE	R/W	0x0	0: 禁止上溢 中断； 1: 允许上溢 中断
10	CCD3IE	R/W	0x0	允许比较 3 中断 (ASYMEN = 1 和 向下计数模式有效) 0: 禁止捕获/比较 1 中断； 1: 允许捕获/比较 1 中断。
9	CCD2IE	R/W	0x0	允许比较 2 中断 (ASYMEN = 1 和 向下计数模式有效) 0: 禁止捕获/比较 1 中断； 1: 允许捕获/比较 1 中断。
8	CCD1IE	R/W	0x0	允许比较 1 中断 (ASYMEN = 1 和 向下计数模式有效) 0: 禁止捕获/比较 1 中断； 1: 允许捕获/比较 1 中断。
7	BIE	R/W	0x0	允许刹车中断(Break interrupt enable) 0: 禁止刹车中断； 1: 允许刹车中断。
6	TIE	R/W	0x0	触发中断使能(Trigger interrupt enable) 0: 禁止触发中断； 1: 使能触发中断。
5	COMIE	R/W	0x0	允许 COM 中断(COM interrupt enable) 0: 禁止 COM 中断； 1: 允许 COM 中断。
4	CC4IE	R/W	0x0	允许捕获/比较 4 中断(Capture/Compare 4 interrupt enable) 0: 禁止捕获/比较 4 中断； 1: 允许捕获/比较 4 中断。

3	CC3IE	R/W	0x0	允许捕获/比较 3 中断(Capture/Compare 3 interrupt enable) 0: 禁止捕获/比较 3 中断； 1: 允许捕获/比较 3 中断。
2	CC2IE	R/W	0x0	允许捕获/比较 2 中断(Capture/Compare 2 interrupt enable) 0: 禁止捕获/比较 2 中断； 1: 允许捕获/比较 2 中断。
1	CC1IE	R/W	0x0	允许捕获/比较 1 中断(Capture/Compare 1 interrupt enable) 0: 禁止捕获/比较 1 中断； 1: 允许捕获/比较 1 中断。
0	UIE	R/W	0x0	允许更新中断(Update interrupt enable) 0: 禁止更新中断； 1: 允许更新中断。

©Copyright 2026, PADAUK Technology Co. Ltd

12.5.5 EPWM 状态寄存器 (EPWM_SR)

Bit	Name	R/W	Reset	Description
31:19	-	R	0x0	保留
18	CC4DIF	R/W1c	0x0	1: EPWM_CNT 向下计数模式的值与 EPWM_CCDR4 的值匹配 CCD2IF 位变高 0: 无匹配发生(写 0 清除)
17	UDIF	R/W1c	0x0	1:EPWM_CNT 向下计数模下溢位 0:无下溢位
16	OVIF	R/W1c	0x0	1: EPWM_CNT 向上计数模上溢位 0:无上溢位
15	CC3DIF	R/W1c	0x0	1: EPWM_CNT 向下计数模式的值与 EPWM_CCDR3 的值匹配 CCD2IF 位变高 0: 无匹配发生(写 0 清除)
14	CC2DIF	R/W1c	0x0	1: EPWM_CNT 向下计数模式的值与 EPWM_CCDR2 的值匹配 CCD2IF 位变高 0: 无匹配发生 (写 0 清除)
13	CC1DIF	R/W1c	0x0	1: EPWM_CNT 向下计数模式的值与 EPWM_CCDR1 的值匹配 CCD1IF 位变高 0: 无匹配发生(写 0 清除)
12	CC4OF	R/W1c	0x0	捕获/比较 4 重复捕获标记(Capture/Compare 4 overcapture flag) 参见 CC1OF 描述。
11	CC3OF	R/W1c	0x0	捕获/比较 3 重复捕获标记(Capture/Compare 3 overcapture flag) 参见 CC1OF 描述。
10	CC2OF	R/W1c	0x0	捕获/比较 2 重复捕获标记(Capture/Compare 2 overcapture flag) 参见 CC1OF 描述。

©Copyright 2026, PADAUK Technology Co. Ltd

9	CC1OF	R/W1c	0x0	捕获/比较 1 重复捕获标记(Capture/Compare 1 overcapture flag) 仅当相应的信道被配置为输入捕获时，该标记可由硬件置 1。写 0 可清除该位。 0: 无重复捕获产生； 1: 计数器的值被捕获到 EPWM_CCR1 寄存器时，CC1IF 的状态已经为'1'。
8	-	R	0x0	保留
7	BIF	R/W1c	0x0	刹车中断标记(Break interrupt flag) 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	R/W1c	0x0	触发器中断标记(Trigger interrupt flag) 当发生触发事件(当从模式控制器处于除门控模式外的其它模式时，在 TRGI 输入端检测到有效边沿，或门控模式下的任一边沿)时由硬件对该位置'1'。它由软件清'0'。 0: 无触发器事件产生； 1: 触发中断等待响应。
5	COMIF	R/W1c	0x0	COM 中断标记(COM interrupt flag) 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	R/W1c	0x0	捕获/比较 4 中断标记(Capture/Compare 4 interrupt flag) 参考 CC1IF 描述。
3	CC3IF	R/W1c	0x0	捕获/比较 3 中断标记(Capture/Compare 3 interrupt flag) 参考 CC1IF 描述。
2	CC2IF	R/W1c	0x0	捕获/比较 2 中断标记(Capture/Compare 2 interrupt flag) 参考 CC1IF 描述。

©Copyright 2026, PADAUK Technology Co. Ltd

1	CC1IF	R/W1c	0x0	捕获/比较 1 中断标记(Capture/Compare 1 interrupt flag) 如果信道 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 EPWM_CR1 寄存器的 CMS 位)。它由软件清'0'。 0: 无匹配发生 1: EPWM_CNT 的值与 EPWM_CCR1 的值匹配 当 EPWM_CCR1 的内容大于 EPWM_APR 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。 如果信道 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 EPWM_CCR1 清'0'。 0: 无输入捕获产生 1: 计数器值已被捕获(拷贝)至 EPWM_CCR1(在 IC1 上检测到与所选极性相同的边沿)
0	UIF	R/W1c	0x0	更新中断标记(Update interrupt flag) 当产生更新事件时该位由硬件置'1'。它由软件清'0'。 0: 无更新事件产生； 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 EPWM_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 EPWM_CR1 寄存器的 URS = 0、UDIS = 0，当设置 EPWM_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 EPWM_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 12.5.3 EPWM 从模式控制寄存器 (EPWM_SMCR)).

12.5.6 EPWM 事件产生寄存器(EPWM_EGR)

Bit	Name	R/W	Reset	Description
31:8	-	R	0x0	保留
7	BG	WO	0x0	产生刹车事件(Break generation) 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，则产生相应的中断。
6	TG	WO	0x0	产生触发事件(Trigger generation) 该位由软件置'1'，用于产生一个触发事件，由硬件自动清'0'。 0: 无动作； 1: EPWM_SR 寄存器的 TIF = 1，若开启对应的中断，则产生相应的中断。
5	COMG	WO	0x0	捕获/比较事件，产生控制更新(Capture/Compare control update generation) 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	WO	0x0	产生捕获/比较 4 事件(Capture/Compare 4 generation) 参考 CC1G 描述。
3	CC3G	WO	0x0	产生捕获/比较 3 事件(Capture/Compare 3 generation) 参考 CC1G 描述。
2	CC2G	WO	0x0	产生捕获/比较 2 事件(Capture/Compare 2 generation) 参考 CC1G 描述。
1	CC1G	WO	0x0	产生捕获/比较 1 事件(Capture/Compare 1 generation) 该位由软件置'1'，用于产生一个捕获/比较事件，由硬件自动清'0'。 0: 无动作 1: 在通道 CC1 上产生一个捕获/比较事件： 若信道 CC1 配置为输出: 设置 CC1IF = 1，若开启对应的中断，则产生相应的中断。 若信道 CC1 配置为输入: 当前的计数器值被捕获至 EPWM_CCR1 寄存器；设置 CC1IF = 1，若开启对应的中断，则产生相应的中断。若 CC1IF 已经为 1，则设置 CC1OF = 1。

©Copyright 2026, PADAUK Technology Co. Ltd

Bit	Name	R/W	Reset	Description
0	UG	WO	0x0	产生更新事件(Update generation) 该位由软件置'1'，由硬件自动清'0'。 0: 无动作 1: 重新初始化计数器，并产生一个更新事件。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 EPWM_ARR 的值。

12.5.7 EPWM 捕获/比较模式寄存器 1 (EPWM_CCMR1)

信道可用于输入(捕获模式)或输出(比较模式)，信道的方向由相应的 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 描述了信道在输出模式下的功能，ICxx 描述了信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:**

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15	OC2CE	R/W	0x0	输出比较 2 清 0 使能(Output Compare 2 clear enable)
14:12	OC2M	R/W	0x0	输出比较 2 模式(Output Compare 2 mode)
11	OC2PE	R/W	0x0	输出比较 2 预装载使能(Output Compare 2 preload enable)
10	OC2FE	R/W	0x0	输出比较 2 快速使能(Output Compare 2 fast enable)
9:8	CC2S	R/W	0x0	捕获/比较 2 选择。(Capture/Compare 2 selection) 该位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC2 信道被配置为输出； 01: CC2 信道被配置为输入，IC2 映射在 TI2 上； 10: CC2 信道被配置为输入，IC2 映射在 TI1 上； 11: CC2 信道被配置为输入，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 EPWM_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时 (EPWM_CCER 寄存器的 CC2E=0) 才是可写的。
7	OC1CE	R/W	0x0	输出比较 1 清'0'使能(Output Compare 1 clear enable) 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	R/W	0x0	输出比较 1 模式(Output Compare 1 mode) 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。 000: 冻结。输出比较寄存器 EPWM_CCR1 与计数器 EPWM_CNT 间的比较对 OC1REF 不起作用； 001: 匹配时设置通道 1 为有效电平。当计数器 EPWM_CNT 的值与捕获/比较寄存器 1(EPWM_CCR1)相同时，强制 OC1REF 为高。 010: 匹配时设置通道 1 为无效电平。当计数器 EPWM_CNT 的值与捕获/比较寄存器 1(EPWM_CCR1)相同时，强制 OC1REF 为低。 011: 翻转。当 EPWM_CCR1 = EPWM_CNT 时，翻转 OC1REF 的电平。 100: 强制为无效电平。强制 OC1REF 为低。 101: 强制为有效电平。强制 OC1REF 为高。 110: PWM 模式 1— 在向上计数时，当 EPWM_CNT < EPWM_CCR1 时通道 1 为有效电平， 否则为无效电平； 在向下计数时，当 EPWM_CNT > EPWM_CCR1 时通道 1 为无效电平(OC1REF = 0)，否则为有效电平(OC1REF = 1)。 111: PWM 模式 2— 在向上计数时，当 EPWM_CNT < EPWM_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 EPWM_CNT > EPWM_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 (EPWM_BDTR 寄存器中的 LOCK 位)并且 CC1S=00(该信道配置成输出)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	R/W	0x0	输出比较 1 预装载使能(Output Compare 1 preload enable) 0: 禁止 EPWM_CCR1 寄存器的预装载功能，可随时写入 EPWM_CCR1 寄存器，并且新写入的数值立即起作用。 1: 开启 EPWM_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，EPWM_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(EPWM_BDTR 寄存器中的 LOCK 位)并且 CC1S = 00(该信道配置成输出)则该位不能被修改。 注 2: 仅在单脉冲模式下(EPWM_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	R/W	0x0	输出比较 1 快速使能(Output Compare 1 fast enable) 该位用于加快 CC 输出对触发输入事件的响应。 0: 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像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 只在信道被配置成 PWM1 或 PWM2 模式时起作用。
1:0	CC1S	R/W	0x0	捕获/比较 1 选择。(Capture/Compare 1 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC1 信道被配置为输出； 01: CC1 信道被配置为输入，IC1 映射在 TI1 上； 10: CC1 信道被配置为输入，IC1 映射在 TI2 上； 11: CC1 信道被配置为输入，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 EPWM_SMCR 寄存器的 TS 位选择)。 注: CC1S 仅在通道关闭时(EPWM_CCER 寄存器的 CC1E=0)才是可写的。

● **输入捕获模式:**

Bit	Name	R/W	Reset	Description
31:16	-	-	0x0	保留
15:12	IC2F	R/W	0x0	输入捕获 2 滤波器(Input capture 2 filter)
11:10	IC2PSC	R/W	0x0	输入/捕获 2 预分频器(Input capture 2 prescaler)
9:8	CC2S	R/W	0x0	捕获/比较 2 选择(Capture/Compare 2 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC2 信道被配置为输出； 01: CC2 信道被配置为输入，IC2 映射在 TI2 上； 10: CC2 信道被配置为输入，IC2 映射在 TI1 上； 11: CC2 信道被配置为输入，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 EPWM_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时(EPWM_CCER 寄存器的 CC2E = 0)才是可写的。
7:4	IC1F	R/W	0x0	输入捕获 1 滤波器(Input capture 1 filter) 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变: 0000: 无滤波器，以 fDTS 采样 0001: 采样频率 fSAMPLING = fCK_INT，N = 2 0010: 采样频率 fSAMPLING = fCK_INT，N = 4 0011: 采样频率 fSAMPLING = fCK_INT，N = 8 0100: 采样频率 fSAMPLING = fDTS/2，N = 6 0101: 采样频率 fSAMPLING = fDTS/2，N = 8 0110: 采样频率 fSAMPLING = fDTS/4，N = 6 0111: 采样频率 fSAMPLING = fDTS/4，N = 8 1000: 采样频率 fSAMPLING = fDTS/8，N = 6 1001: 采样频率 fSAMPLING = fDTS/8，N = 8 1010: 采样频率 fSAMPLING = fDTS/16，N = 5 1011: 采样频率 fSAMPLING = fDTS/16，N = 6 1100: 采样频率 fSAMPLING = fDTS/16，N = 8 1101: 采样频率 fSAMPLING = fDTS/32，N = 5 1110: 采样频率 fSAMPLING = fDTS/32，N = 6

©Copyright 2026, PADAUK Technology Co. Ltd

Bit	Name	R/W	Reset	Description
				1111: 采样频率 fSAMPLING = fDTS/32，N = 8

3:2	IC1PSC	R/W	0x0	输入/捕获 1 预分频器(Input capture 1 prescaler) 这 2 位定义了 CC1 输入(IC1)的预分频系数。 一旦 CC1E = 0(EPWM_CCER 寄存器中)，则预分频器复位。 00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获； 01: 每 2 个事件触发一次捕获； 10: 每 4 个事件触发一次捕获； 11: 每 8 个事件触发一次捕获。
1:0	CC1S	R/W	0x0	捕获/比较 1 选择(Capture/Compare 1 Selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC1 信道被配置为输出； 01: CC1 信道被配置为输入，IC1 映射在 TI1 上； 10: CC1 信道被配置为输入，IC1 映射在 TI2 上； 11: CC1 信道被配置为输入，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 EPWM_SMCR 寄存器的 TS 位选择)。 注: CC1S 仅在通道关闭时(EPWM_CCER 寄存器的 CC1E = 0)才是可写的。

12.5.8 EPWM 捕获/比较模式寄存器 2(EPWM_CCMR2)

● **输出比较模式:**

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15	OC4CE	R/W	0x0	输出比较 4 清 0 使能(Output Compare 4 clear enable)
14:12	OC4M[2:0]	R/W	0x0	输出比较 4 模式(Output Compare 4 mode)
11	OC4PE	R/W	0x0	输出比较 4 预装载使能(Output Compare 4 preload enable)
10	OC4FE	R/W	0x0	输出比较 4 快速使能(Output Compare 4 fast enable)
9:8	CC4S[1:0]	R/W	0x0	捕获/比较 4 选择。(Capture/Compare 4 selection) 该 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC4 信道被配置为输出； 01: CC4 信道被配置为输入，IC4 映射在 TI4 上； 10: CC4 信道被配置为输入，IC4 映射在 TI3 上； 11: CC4 信道被配置为输入，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 EPWM_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(EPWM_CCER 寄存器的 CC4E = 0)才是可写的。
7	OC3CE	R/W	0x0	输出比较 3 清'0'使能(Output Compare 3 clear enable)
6:4	OC3M[2:0]	R/W	0x0	输出比较 3 模式(Output Compare 3 mode)
3	OC3PE	R/W	0x0	输出比较 3 预装载使能(Output Compare 3 preload enable)
2	OC3FE	R/W	0x0	输出比较 3 快速使能(Output Compare 3 fast enable)
1:0	CC3S[1:0]	R/W	0x0	捕获/比较 3 选择。(Capture/Compare 3 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC3 信道被配置为输出； 01: CC3 信道被配置为输入，IC3 映射在 TI3 上； 10: CC3 信道被配置为输入，IC3 映射在 TI4 上； 11: CC3 信道被配置为输入，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 EPWM_SMCR 寄存器的 TS 位选择)。 注: CC3S 仅在通道关闭时(EPWM_CCER 寄存器的 CC3E = 0)才是

				可写的。

● **输入捕获模式:**

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:12	IC4F	R/W	0x0	输入捕获 4 滤波器(Input capture 4 filter)
11:10	IC4PSC	R/W	0x0	输入/捕获 4 预分频器(Input capture 4 prescaler)
9:8	CC4S	R/W	0x0	捕获/比较 4 选择(Capture/Compare 4 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC4 信道被配置为输出； 01: CC4 信道被配置为输入，IC4 映射在 TI4 上； 10: CC4 信道被配置为输入，IC4 映射在 TI3 上； 11: CC4 信道被配置为输入，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 EPWM_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(EPWM_CCER 寄存器的 CC4E = 0)才是可写的。
7:4	IC3F	R/W	0x0	输入捕获 3 滤波器(Input capture 3 filter)
3:2	IC3PSC	R/W	0x0	输入/捕获 3 预分频器(Input capture 3 prescaler)
1:0	CC3S	R/W	0x0	捕获/比较 3 选择(Capture/Compare 3 Selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC3 信道被配置为输出； 01: CC3 信道被配置为输入，IC3 映射在 TI3 上； 10: CC3 信道被配置为输入，IC3 映射在 TI4 上； 11: CC3 信道被配置为输入，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 EPWM_SMCR 寄存器的 TS 位选择)。 注: CC3S 仅在通道关闭时(EPWM_CCER 寄存器的 CC3E = 0)才是可写的。

12.5.9 EPWM 捕获/比较使能寄存器 (EPWM_CCER)

Bit	Name	R/W	Reset	Description
31:14	-	R	0x0	保留
13	CC4P	R/W	0x0	输入/捕获 4 输出极性(Capture/Compare 4 output polarity) 参考 CC1P 的描述。
12	CC4E	R/W	0x0	输入/捕获 4 输出使能(Capture/Compare 4 output enable) 参考 CC1E 的描述。
11	CC3NP	R/W	0x0	输入/捕获 3 互补输出极性(Capture/Compare 3 complementary outputpolarity) 参考 CC1NP 的描述。
10	CC3NE	R/W	0x0	输入/捕获 3 互补输出使能(Capture/Compare 3 complementary output enable) 参考 CC1NE 的描述。
9	CC3P	R/W	0x0	输入/捕获 3 输出极性(Capture/Compare 3 output polarity) 参考 CC1P 的描述。
8	CC3E	R/W	0x0	输入/捕获 3 输出使能(Capture/Compare 3 output enable) 参考 CC1E 的描述。
7	CC2NP	R/W	0x0	输入/捕获 2 互补输出极性(Capture/Compare 2 complementary outputpolarity) 参考 CC1NP 的描述。
6	CC2NE	R/W	0x0	输入/捕获 2 互补输出使能(Capture/Compare 2 complementary output enable) 参考 CC1NE 的描述。
5	CC2P	R/W	0x0	输入/捕获 2 输出极性(Capture/Compare 2 output polarity) 参考 CC1P 的描述。
4	CC2E	R/W	0x0	输入/捕获 2 输出使能(Capture/Compare 2 output enable) 参考 CC1E 的描述。
3	CC1NP	R/W	0x0	输入/捕获 1 互补输出极性(Capture/Compare 1 complementary output polarity) 0: OC1N 高电平有效； 1: OC1N 低电平有效。 注: 一旦 LOCK 级别(EPWM_BDTR 寄存器中的 LOCK 位)设为 3 或 2 且 CC1S = 00(信道配置为输出)则该位不能被修改。

2	CC1NE	R/W	0x0	输入/捕获 1 互补输出使能(Capture/Compare 1 complementary output enable) 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	R/W	0x0	输入/捕获 1 输出极性(Capture/Compare 1 output polarity) CC1 信道配置为输出: 0: OC1 高电平有效； 1: OC1 低电平有效。 CC1 信道配置为输入: 该位选择是 IC1 还是 IC1 的反相信号作为触发或捕获信号。 0: 不反相：捕获发生在 IC1 的上升沿；当用作外部触发器时，IC1 不反相。 1: 反相：捕获发生在 IC1 的下降沿；当用作外部触发器时，IC1 反相。 注: 一旦 LOCK 级别(EPWM_BDTR 寄存器中的 LOCK 位)设为 3 或 2，则该位不能被修改。
0	CC1E	R/W	0x0	输入/捕获 1 输出使能(Capture/Compare 1 output enable) CC1 信道配置为输出: 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 信道配置为输入: 该位决定了计数器的值是否能捕获入 EPWM_CCR1 寄存器。 0: 捕获禁止； 1: 捕获使能。

©Copyright 2026, PADAUK Technology Co. Ltd

控制位					输出状态	
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

<table>
<tr><th colspan="5">控制位</th><th colspan="2">输出状态</th></tr>
<tr><td>MOE</td><td>OSSI</td><td>OSSR</td><td>CCxE</td><td>CCxNE</td><td>OCx 输出状态</td><td>OCxN 输出状态</td></tr>
<tr><td rowspan="8">0</td><td>0</td><td rowspan="8">X</td><td>0</td><td>0</td><td colspan="2" rowspan="4">输出禁止(与定时器断开)
异步地：OC x =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。</td></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td colspan="2" rowspan="4">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。</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>

表 12-4 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位

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

注: 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 以及 AFIO 寄存器。

12.5.10 EPWM 计数器 (EPWM_CNT)

Bit	Name	R/W	Reset	Description
31:16	-	-	0x0	保留
15:0	CNT	R/W	0x0	计数器的值 (Counter value)

12.5.11 EPWM 预分频器 (EPWM_PSC)

Bit	Name	R/W	Reset	Description
31:16	-	-	0x0	保留
15:0	PSC	R/W	0x0	预分频器的值(Prescaler value) 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'。

12.5.12 EPWM 自动重装载寄存器 (EPWM_ARR)

Bit	Name	R/W	Reset	Description
31:16	-	-	0x0	保留
19:0	ARR	R/W	0x0	自动重装载的值(Prescaler value) ARR 包含了将要装载入实际的自动重装载寄存器的值。 详细参考 12.3.1 节: 有关 ARR 的更新和动作。 当自动重装载的值为空时，计数器不工作。

©Copyright 2026, PADAUK Technology Co. Ltd

12.5.13 EPWM 重复计数寄存器(EPWM_RCR)

Bit	Name	R/W	Reset	Description
31:8	-	-	0x0	保留
7:0	REP	R/W	0x0	重复计数器的值(Repetition counter value) 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 EPWM_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；

12.5.14 EPWM 捕获/比较寄存器 1(EPWM_CCR1)

Bit	Name	R/W	Reset	Description
31:20	-	-	0x0	保留
19:0	CCR1	R/W	0x0	捕获/比较通道 1 的值(Capture/Compare 1 value) 若 CC1 信道配置为输出: 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 EPWM_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 EPWM_CNT 的比较，并在 OC1 端口上产生输出信号。 若 CC1 信道配置为输入: CCR1 包含了由上一次输入捕获 1 事件(IC1)传输的计数器值。

12.5.15 EPWM 捕获/比较寄存器 2 (EPWM_CCR2)

Bit	Name	R/W	Reset	Description
31:20	-	-	0x0	保留
19:0	CCR2	R/W	0x0	捕获/比较通道 2 的值(Capture/Compare 2 value) 若 CC2 信道配置为输出: 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 EPWM_CCMR2 寄存器(OC2PE 位)中未选择预装载特性，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 EPWM_CNT 的比较，并在 OC2 端口上产生输出信号。 若 CC2 信道配置为输入: CCR2 包含了由上一次输入捕获 2 事件(IC2)传输的计数器值。

12.5.16 EPWM 捕获/比较寄存器 3 (EPWM_CCR3)

Bit	Name	R/W	Reset	Description
31:20	-	-	0x0	保留
19:0	CCR3	R/W	0x0	捕获/比较通道 3 的值(Capture/Compare 3 value) 若 CC3 信道配置为输出： 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 EPWM_CCMR3 寄存器(OC3PE 位)中未选择预装载特性，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 EPWM_CNT 的比较，并在 OC3 端口上产生输出信号。 若 CC3 信道配置为输入： CCR3 包含了由上一次输入捕获 3 事件(IC3)传输的计数器值。

12.5.17 EPWM 捕获/比较寄存器4 (EPWM_CCR4)

Bit	Name	R/W	Reset	Description
31:20	-	R	0x0	保留
19:0	CCR4	R/W	0x0	捕获/比较通道 4 的值(Capture/Compare 4 value) 若 CC4 信道配置为输出： 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 EPWM_CCMR4 寄存器(OC4PE 位)中未选择预装载特性，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 EPWM_CNT 的比较，并在 OC4 端口上产生输出信号。 若 CC4 信道配置为输入： CCR4 包含了由上一次输入捕获 4 事件(IC4)传输的计数器值。

12.5.18 EPWM 刹车和死区寄存器 (EPWM_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 EPWM_BDTR 寄存器时对它们进行配置。

Bit	Name	R/W	Reset	Description
31:25	-	R	0x0	保留
24	DTAE	R/W	0x0	不对称死区时间使能控制 0: 上升沿和下降沿的死区时间是一样的,由 DTG[7:0]定义 1:上升沿的死区时间,由 DTG[7:0] 定义，
23:16	DTGF[7:0]	R/W	0x0	互补通道 OCxREFC 下降沿死区时间配置 DTGF[7:5]=0xx：DeadTime = (DTGF[7:0] + 0) × tDTS × 1 DTGF[7:5]=10x：DeadTime = (DTGF[5:0] + 64) × tDTS × 2 DTGF[7:5]=110: DeadTime = (DTGF[4:0] + 32) × tDTS × 8 DTGF[7:5]=111: DeadTime = (DTGF[4:0] + 32) × tDTS × 16
15	MOE	R/W	0x0	主输出使能(Main output enable) 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信道有效。 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(EPWM_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 11.5.9 EPWM 和 TIM8 捕获/比较使能寄存器(EPWM_CCER)。

14	AOE	R/W	0x0	自动输出使能(Automatic output enable) 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(EPWM_BDTR 寄存器中的 LOCK 位)设为'1'，则该位不能被修改。
13	BKP	R/W	0x0	刹车输入极性(Break polarity) 0: 刹车输入低电平有效； 1: 刹车输入高电平有效。 注: 一旦 LOCK 级别(EPWM_BDTR 寄存器中的 LOCK 位)设为'1'，则该位不能被修改。 注: 任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。
12	BKE	R/W	0x0	刹车功能使能(Break enable) 0: 禁止刹车输入(BRK 及 CCS 时钟失效事件)； 1: 开启刹车输入(BRK 及 CCS 时钟失效事件)。 注: 当设置了 LOCK 级别 1 时(EPWM_BDTR 寄存器中的 LOCK 位)， 该位不能被修改。 注: 任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。

11	OSSR	R/W	0x0	运行模式下“关闭状态”选择(Off-state selection for Run mode) 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(12.5.9 节，EPWM 捕获/比较使能寄存器(EPWM_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(EPWM_BDTR 寄存器中的 LOCK 位)设为 2，则该位不能被修改。
10	OSSI	R/W	0x0	空闲模式下“关闭状态”选择(Off-state selection for Idle mode) 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(12.5.9 节，EPWM 捕获/比较使能寄存器(EPWM_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(EPWM_BDTR 寄存器中的 LOCK 位)设为 2，则该位不能被修改。

9:8	LOOK[1:0]	R/W	0x0	锁定设置(Lock Configuration) 该位为防止软件错误而提供写保护。 00: 锁定关闭，寄存器无写保护； 01: 锁定级别 1，不能写入 EPWM_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 EPWM_CR2 寄存器的 OISx/OISxN 位； 10: 锁定级别 2，不能写入锁定级别 1 中的各位，也不能写入 CC 极性位(一旦相关通道通过 CCxS 位设为输出，CC 极性位是 EPWM_CCER 寄存器的 CCxP/CCNxP 位)以及 OSSR/OSSI 位； 11: 锁定级别 3，不能写入锁定级别 2 中的各位，也不能写入 CC 控制位(一旦相关通道通过 CCxS 位设为输出， CC 控制位是 EPWM_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 EPWM_BDTR 寄存器，则其内容冻结直至复位。
7:0	DTG[7:0]	R/W	0x0	死区发生器设置(Dead-time generator setup) 这些位定义了插入互补输出之间的死区持续时间。 假设 DT 表示其持续时间： DTG[7:5] = 0xx => DT = DTG[7:0]×Tdtg，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 = 125 ns(8 MHz)，可能的死区时间为： 0 到 15875 ns，若步长时间为 125 ns； 16 us 到 31750 ns，若步长时间为 250 ns； 32 us 到 63 us，若步长时间为 1 us； 64 us 到 12 6us，若步长时间为 2 us； 注: 一旦 LOCK 级别(EPWM_BDTR 寄存器中的 LOCK 位)设为 1、2 或 3，则不能修改这些位。

12.5.19 EPWM 计数器向下比较寄存器1 (EPWM_CCDR1)

Bit	Name	R/W	Reset	Description
31:20	-	R	0x0	保留
19:0	CCDR1	R/W	0x0	EPWM 计数器向下比较值(比较通道 1 的值) (EPWM 中心对齐模式下不对称计数 EPWM_CR1.ASYMEN 使能有效)

12.5.20 EPWM 计数器向下比较寄存器2 (EPWM_CCDR2)

Bit	Name	R/W	Reset	Description
31:20	-	R	0x0	保留
19:0	CCDR2	R/W	0x0	EPWM 计数器向下比较值(比较通道 2 的值) (EPWM 中心对齐模式下不对称计数 EPWM_CR1.ASYMEN 使能有效)

12.5.21 EPWM 计数器向下比较寄存器3 (EPWM_CCDR3)

Bit	Name	R/W	Reset	Description
31:20	-	R	0x0	保留
19:0	CCDR3	R/W	0x0	EPWM 计数器向下比较值(比较通道 3 的值) (EPWM 中心对齐模式下不对称计数 EPWM_CR1.ASYMEN 使 能有效)

©Copyright 2026, PADAUK Technology Co. Ltd

12.5.22 EPWM 计数器向下比较寄存器4 (EPWM_CCDR4)

Bit	Name	R/W	Reset	Description
31:20	-	R	0x0	保留
19:0	CCDR4	R/W	0x0	EPWM 计数器向下比较值(比较通道 4 的值)_ (EPWM 中心对齐模式下不对称计数 EPWM_CR1.ASYMEN 使能有效)

©Copyright 2026, PADAUK Technology Co. Ltd

12.6 TIM2 简介

通用定时器由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 RCC 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。

12.7 TIM2 主要功能

通用 TIM2 定时器功能包括:

- 16 位向上、向下、向上/向下自动装载计数器
- 16 位可编程(可以实时修改)预分频器，计数器时钟频率的分频系数为 1～65536 之间的任意数值
- 4 个独立通道:
  - 输入捕获
  - 输出比较
  - PWM 生成(边缘或中间对齐模式)
  - 单脉冲模式输出
- 使用外部信号控制定时器和定时器互连的同步电路
- 如下事件发生时产生中断:
  - 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
- 触发输入作为外部时钟或者按周期的电流管

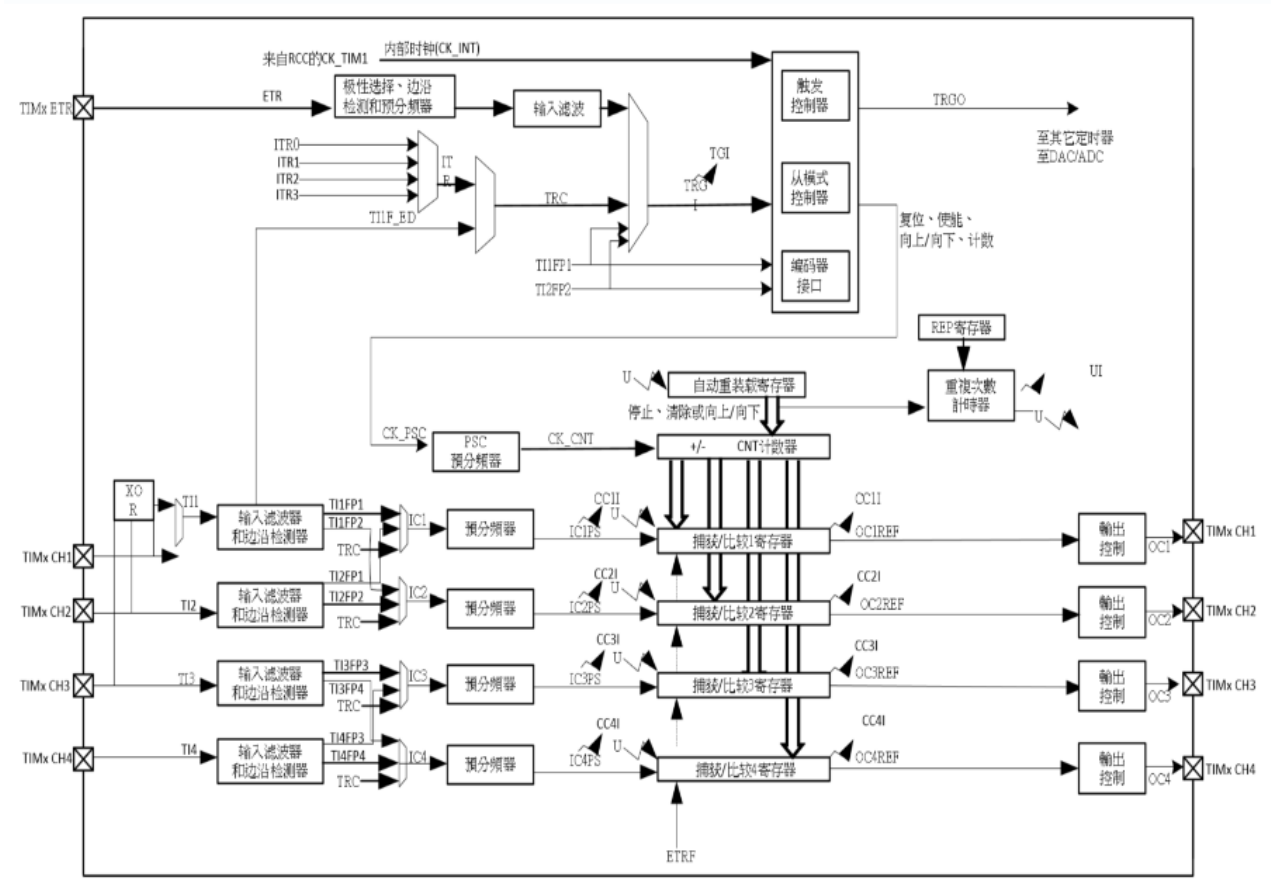


图 12-49 通用定时器框图

注:

Reg 根据控制位的设定，在 U(更新)事件时传送预加载寄存器的内容至工作寄存器

事件

中断

12.8 TIM2 功能描述

12.8.1 时基单元

可编程通用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，在计数器运行时仍可以读写。时基单元包含:

- 计数器寄存器 (TIM2_CNT)
- 预分频器寄存器 (TIM2_PSC)
- 自动装载寄存器 (TIM2_ARR)

自动装载寄存器是预先装载的，写或读自动重装载寄存器将访问预装载寄存器。根据在 TIM2_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 TIM2_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 TIM2_CR1 寄存器中的计数器使能位(CEN)时，CK_CNT 才有效。(有关计数器使能的细节，请参见控制器的从模式描述)。

注: 真正的计数器使能信号 CNT_EN 是在 CEN 的一个时钟周期后被设置。

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 TIM2_PSC 寄存器中的)16 位寄存器控制的 16 位计数器。这个控制寄存器带有缓冲器，它能够在工作时被改变。新的预分频器参数在下一次更新事件到来时被采用。图 13-2 和图 13-3 给出了在预分频器运行时，更改计数器参数的例子。

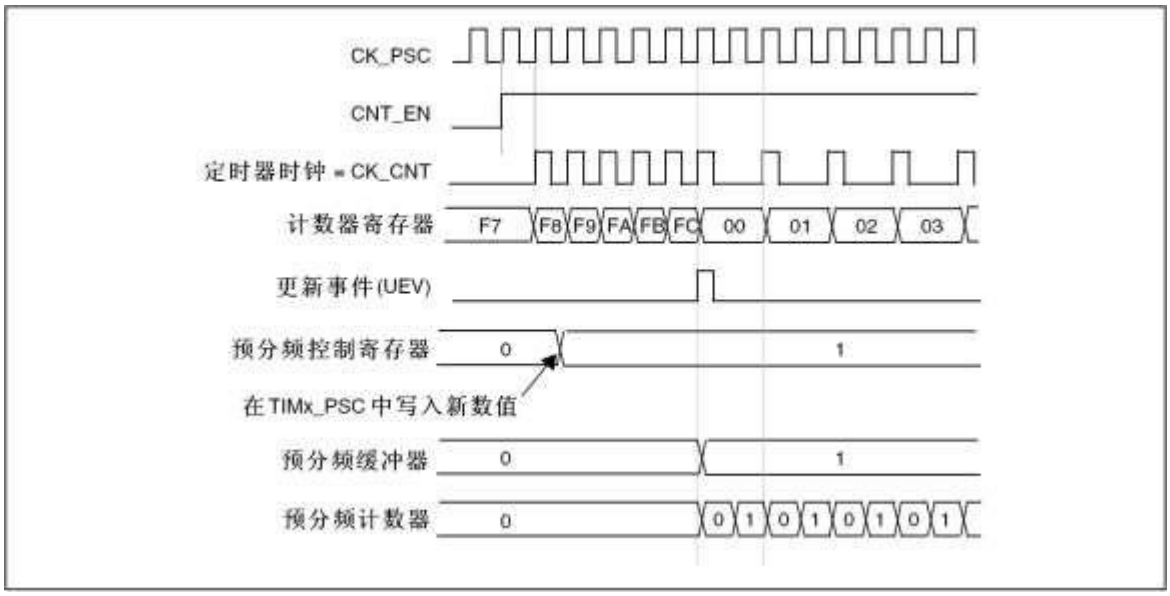


图 12-50 当预分频器的参数从 1 变到 2 时，计数器的时序图

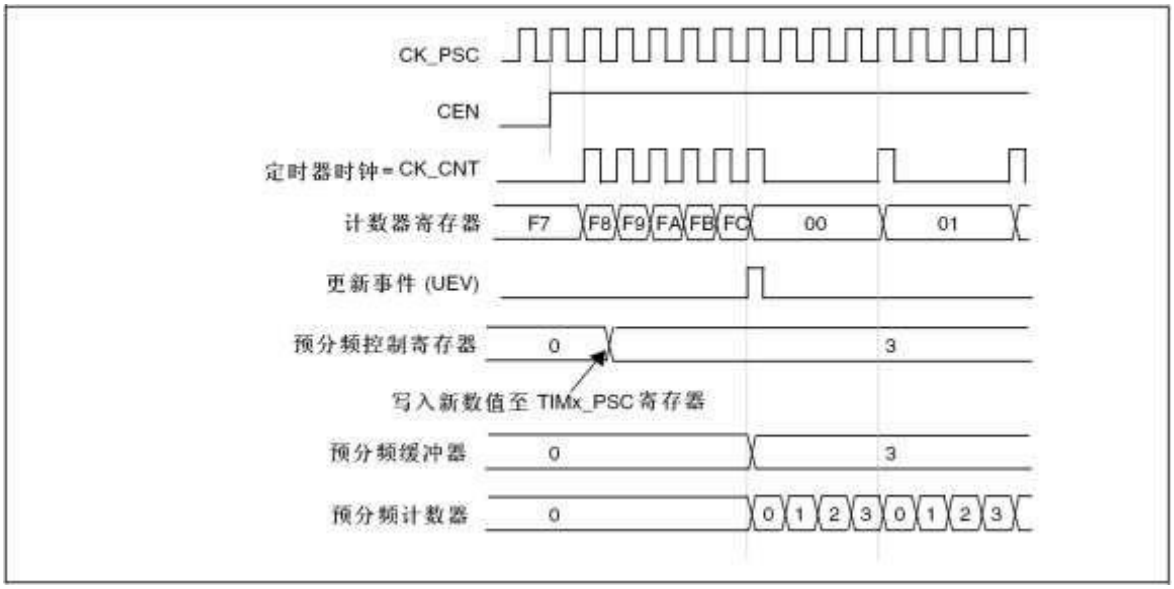


图 12-51 当预分频器的参数从 1 变到 4 时，计数器的时序图

12.8.2 计数器模式

12.8.2.1 向上计数模式

在向上计数模式中，计数器从 0 计数到自动加载值(TIM2_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 TIM2_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 TIM2_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清'0'之前，将不产生更新事件。但是在应该产生更新事件时， 计数器仍会被清'0'，同时预分频器的计数也被请 0(但预分频系数不变)。此外，如果设置了 TIM2_CR1 寄存器中的 URS 位(选择更新请求)，设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志(即不产生中断)；这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时(依据 URS 位)设置更新标志位(TIM2_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(TIM2_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(TIM2_ARR)。

下图给出一些例子，当 TIM2_ARR = 0x36 时计数器在不同时钟频率下的动作。

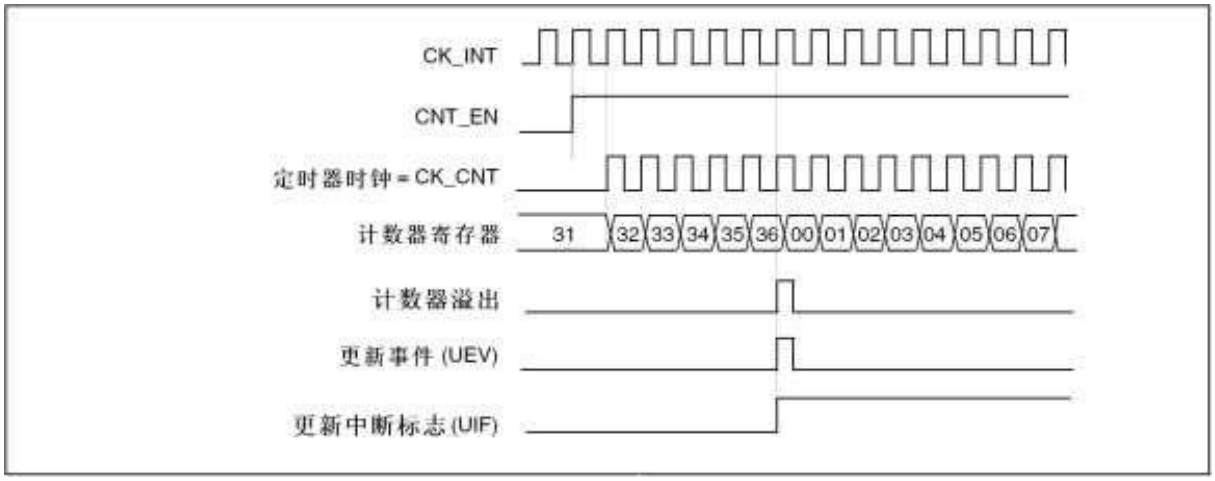


图 12-52 计数器时序图：内部时钟分频因子为 1

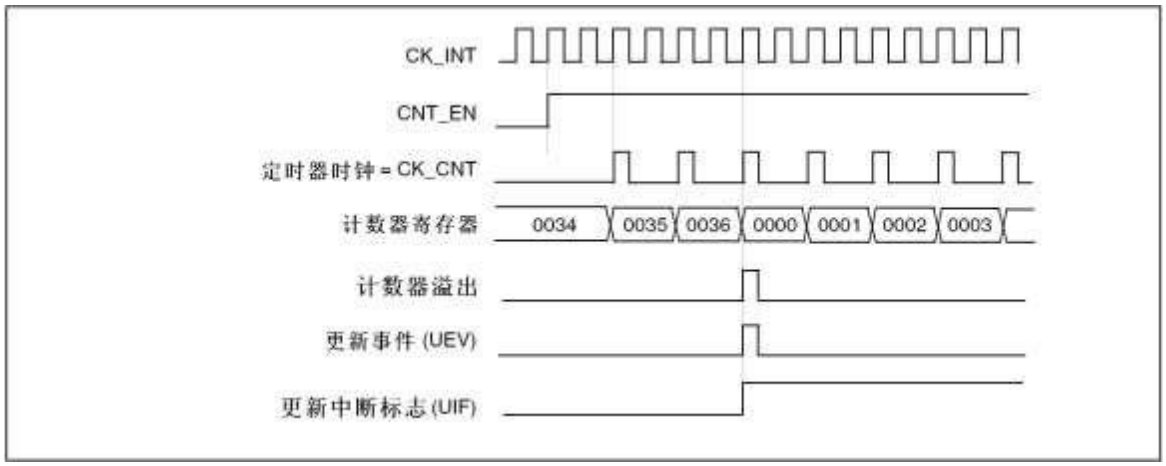


图 12-53 计数器时序图: 内部时钟分频因子为 2

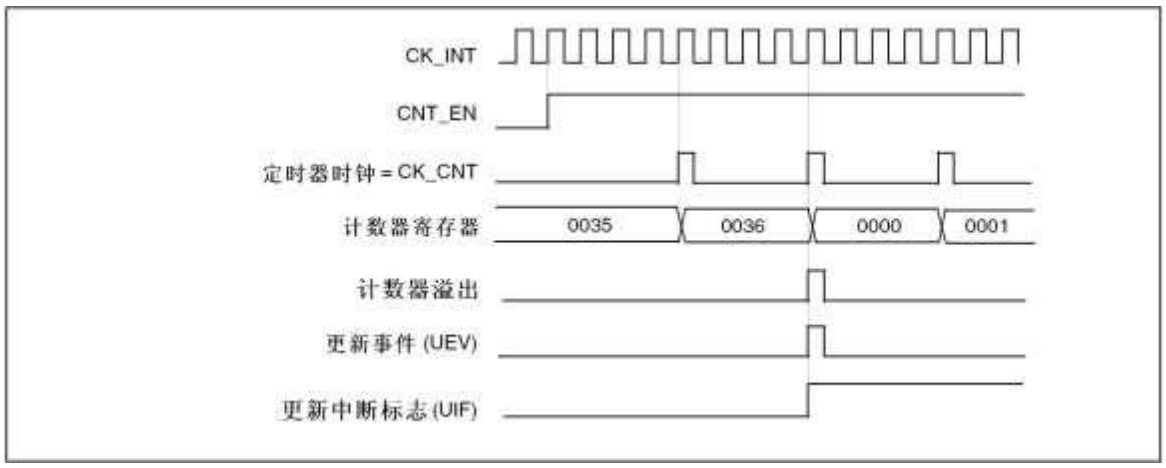


图 12-54 计数器时序图: 内部时钟分频因子为 4

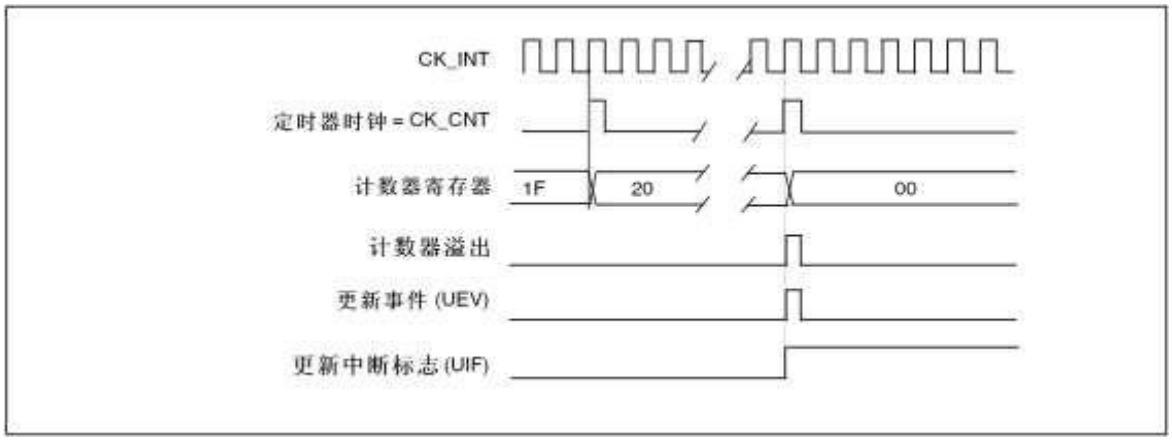


图 12-55 计数器时序图: 内部时钟分频因子为 N

©Copyright 2026, PADAUK Technology Co. Ltd

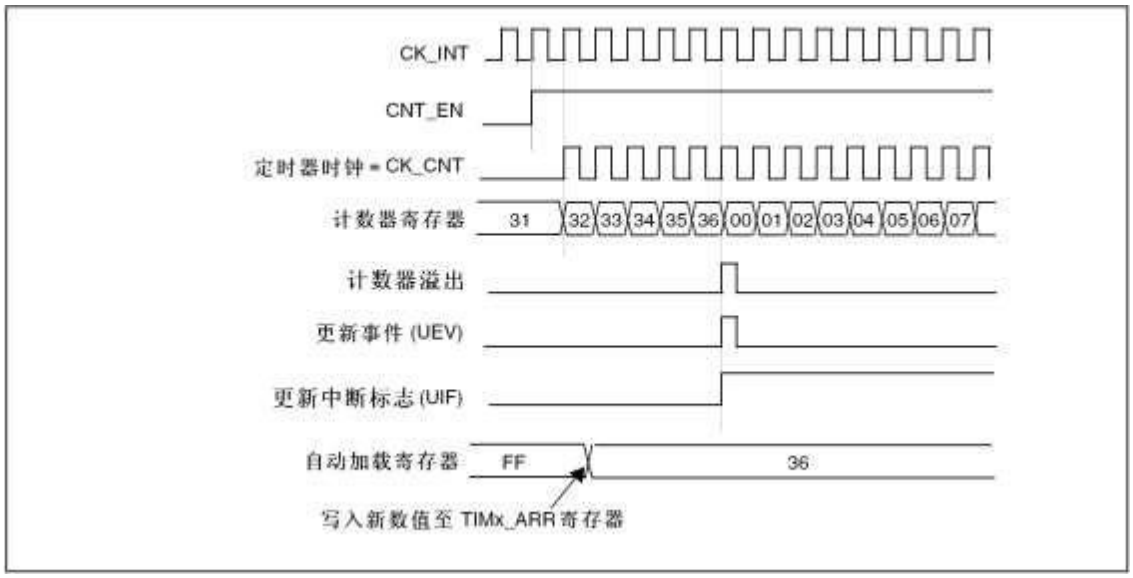


图 12-56 计数器时序图: 当 ARPE=0 时的更新事件(TIM2_ARR 没有预装入)

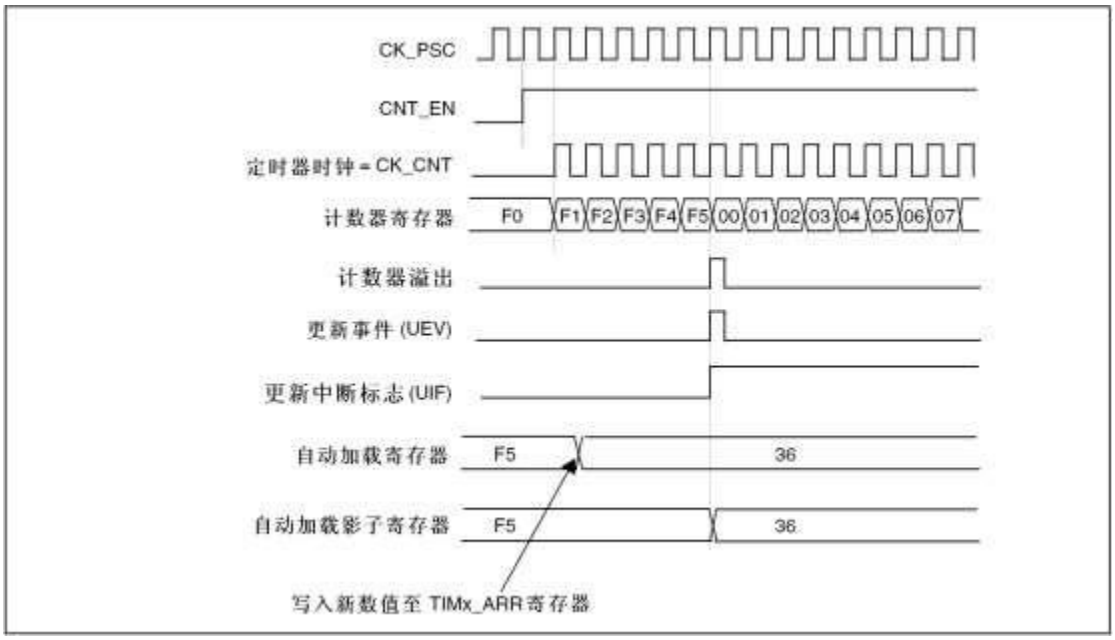


图 12-57 计数器时序图: 当 ARPE = 1 时的更新事件(预装入了 TIM2_ARR)

12.8.2.2 向下计数模式

在向下模式中，计数器从自动装入的值(TIM2_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 TIM2_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 TIM2_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，同时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 TIM2_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)，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(TIM2_SR 寄存器中的 UIF 位)也被设置。

- 预分频器的缓存器被置入预装载寄存器的值(TIM2_PSC 寄存器的值)。
- 当前的自动加载寄存器被更新为预装载值(TIM2_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 TIM2_ARR = 0x36 时，计数器在不同时钟频率下的操作例子。

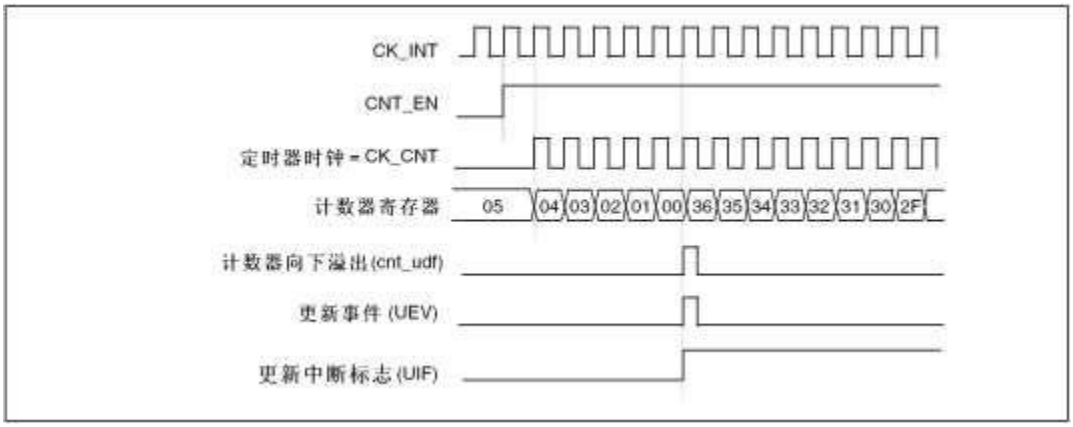


图 12-58 计数器时序图: 内部时钟分频因子为 1

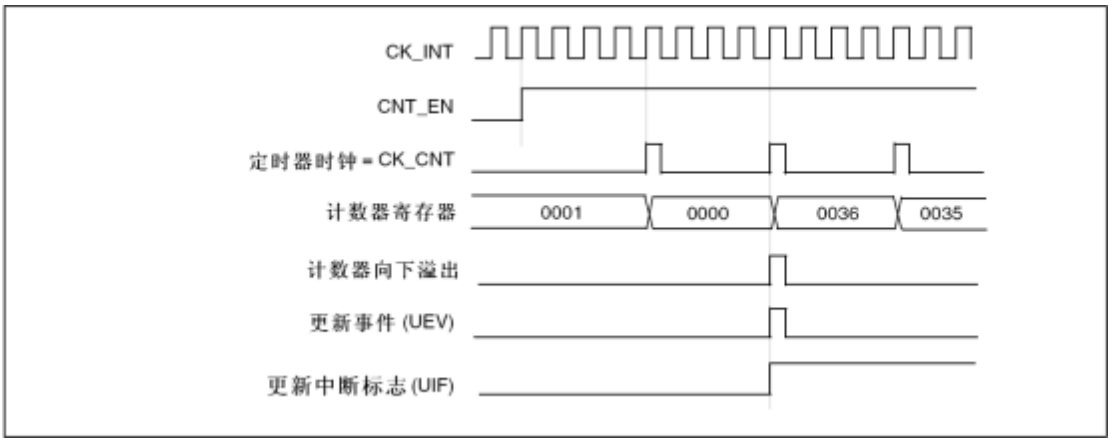


图 12-59 计数器时序图: 内部时钟分频因子为 2

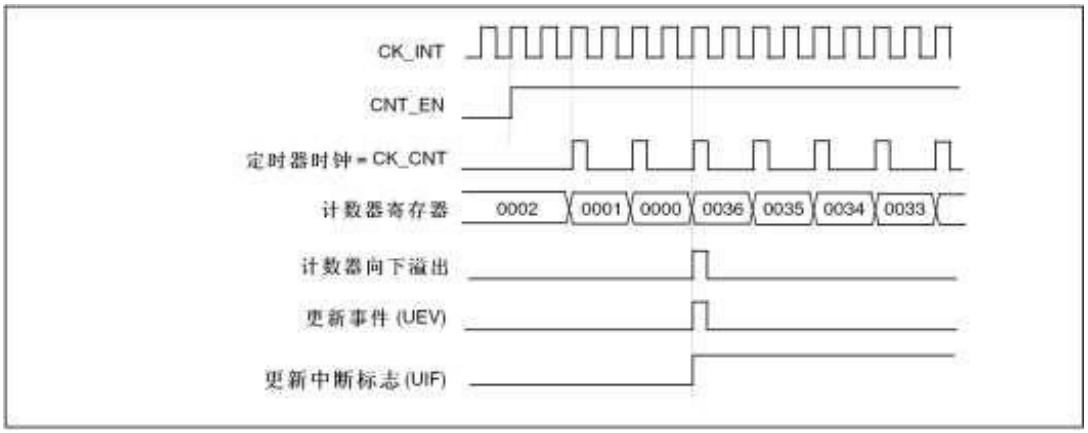


图 12-60 计数器时序图：内部时钟分频因子为 4

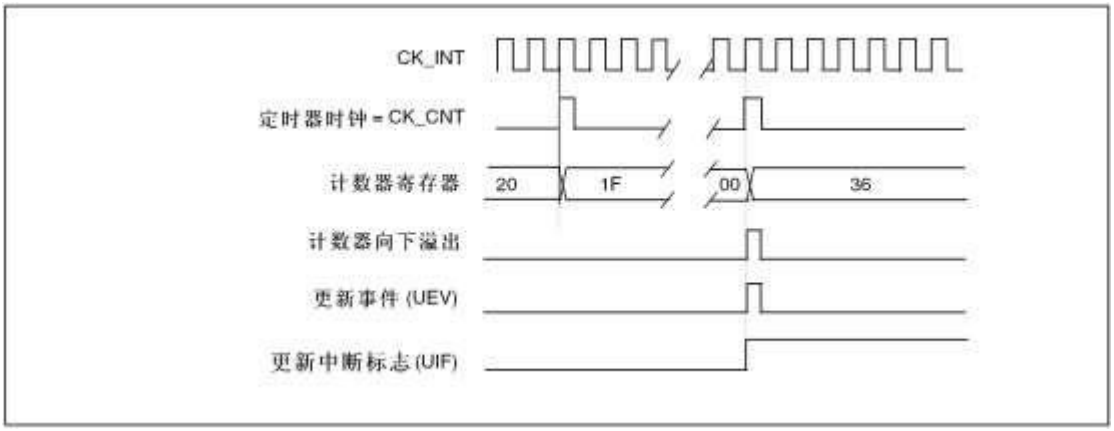


图 12-61 计数器时序图: 内部时钟分频因子为 N

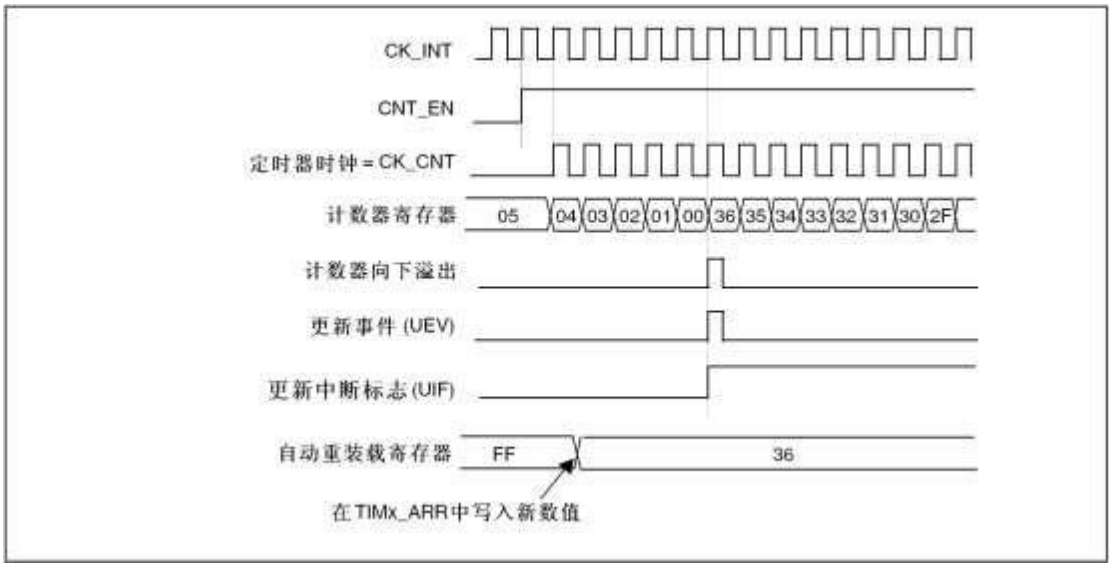


图 12-62 计数器时序图: 当没有使用重复计数器时的更新事件

12.8.2.3 中央对齐模式 (向上/向下计数)

在中央对齐模式，计数器从 0 开始计数到自动加载的值(TIM2_ARR 寄存器)−1，产生一个计数器溢出事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。

在这个模式，不能写入 TIM2_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 TIM2_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 TIM2_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 TIM2_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)，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(TIM2_SR 寄存器中的 UIF 位)也被设置。

- 预分频器的缓存器被加载为预装载(TIM2_PSC 寄存器)的值。
- 当前的自动加载寄存器被更新为预装载值(TIM2_ARR 寄存器中的内容)。

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

以下是一些计数器在不同时钟频率下的操作的例子:

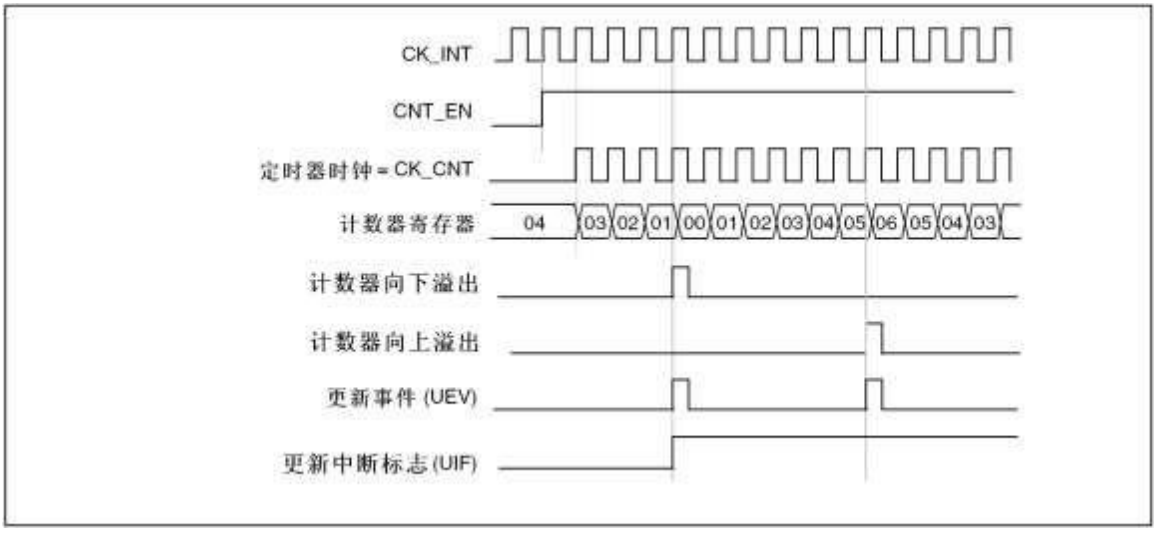


图 12-63 计数器时序图: 内部时钟分频因子为 1，TIM2_ARR = 0x6

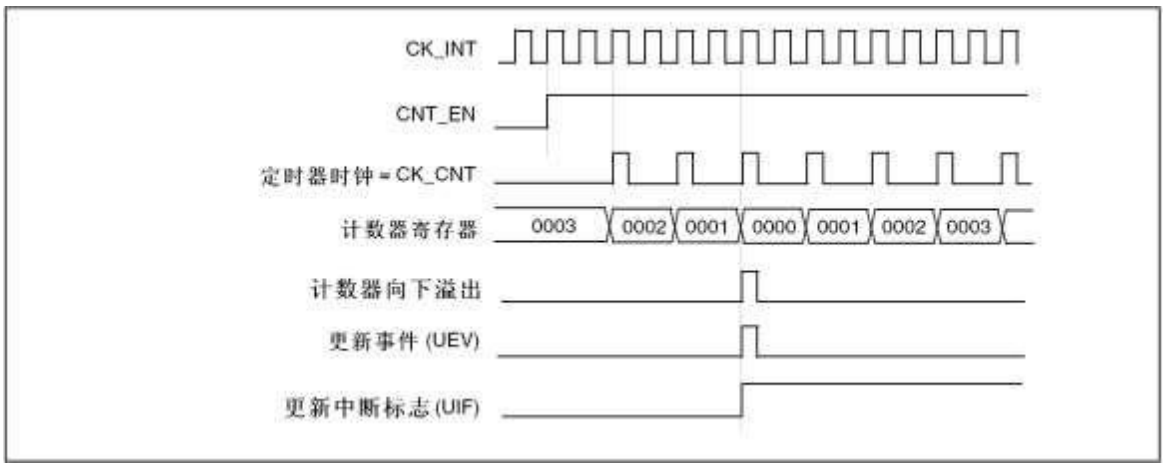


图 12-64 计数器时序图: 内部时钟分频因子为 2

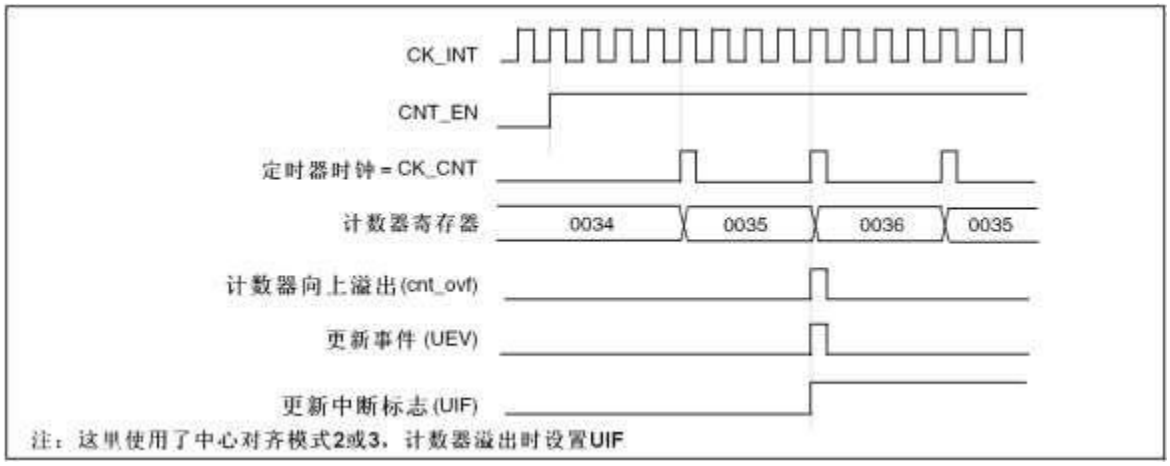


图 12-65 计数器时序图: 内部时钟分频因子为 4，TIM2_ARR = 0x36

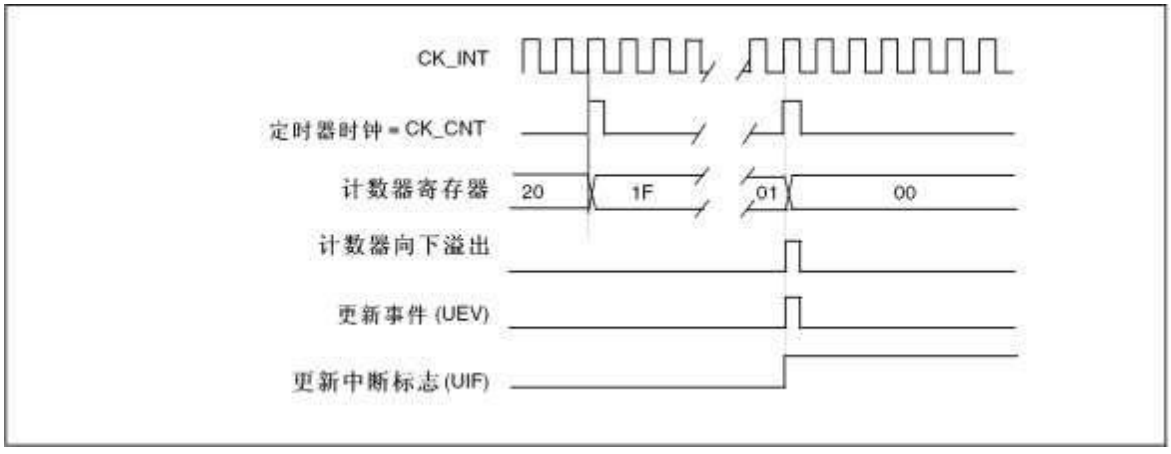


图 12-66 计数器时序图: 内部时钟分频因子为 N

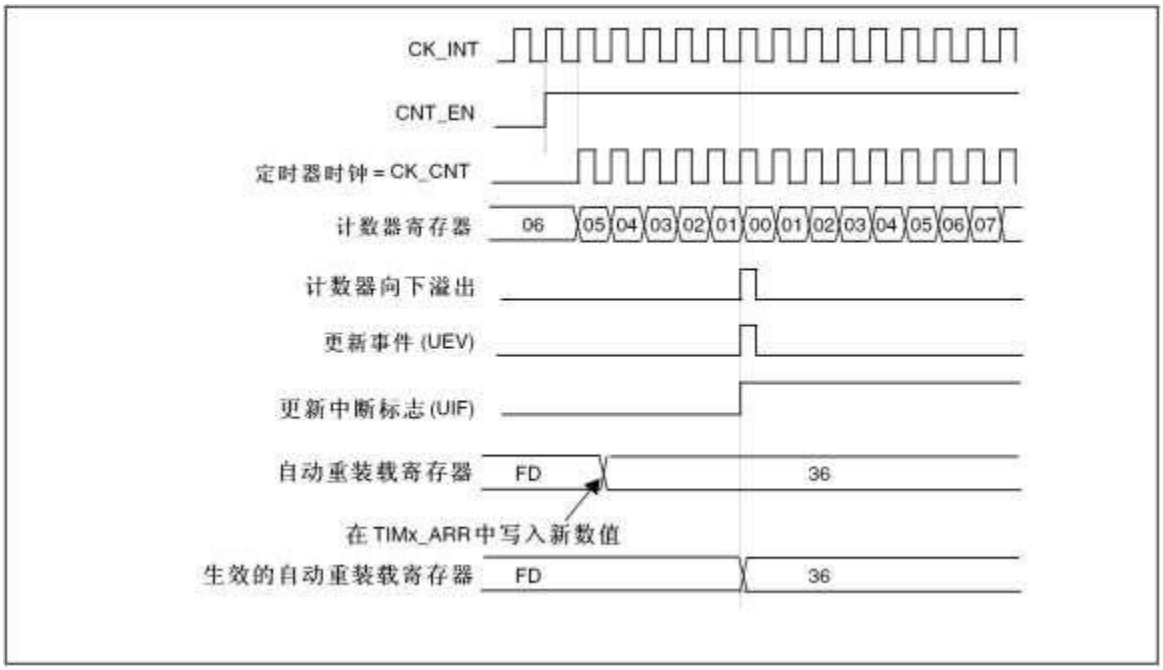


图 12-67 计数器时序图: ARPE=1 时的更新事件(计数器下溢)

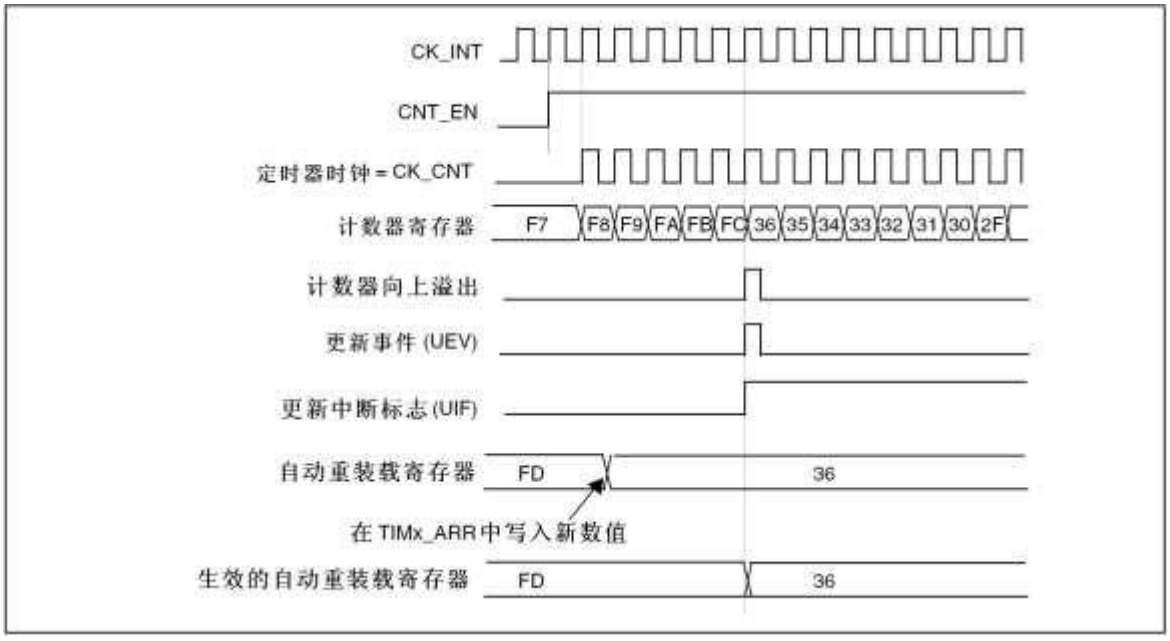


图 12-68 计数器时序图: ARPE=1 时的更新事件(计数器溢出)

12.8.3 时钟选择

计数器时钟可由下列时钟源提供:

- 内部时钟(CK_INT)
- 外部时钟模式 1: 外部输入脚(TIx)
- 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。

12.8.3.1 内部时钟源(CK_INT)

如果禁止了从模式控制器(TIM2_SMCR 寄存器的 SMS = 000)，则 CEN、DIR(TIM2_CR1 寄存器)和 UG 位(TIM2_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。下图显示了控制电路和向上计数器在一般模式下，不带预分频器时的操作。

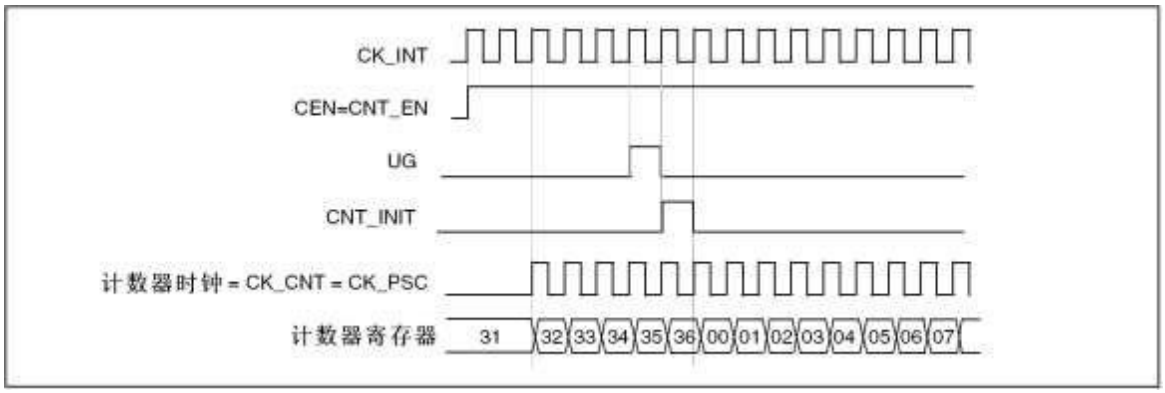


图 12-69 一般模式下的控制电路，内部时钟分频因子为 1

12.8.3.2 外部时钟源模式 1

当 TIM2_SMCR 寄存器的 SMS = 111 时，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

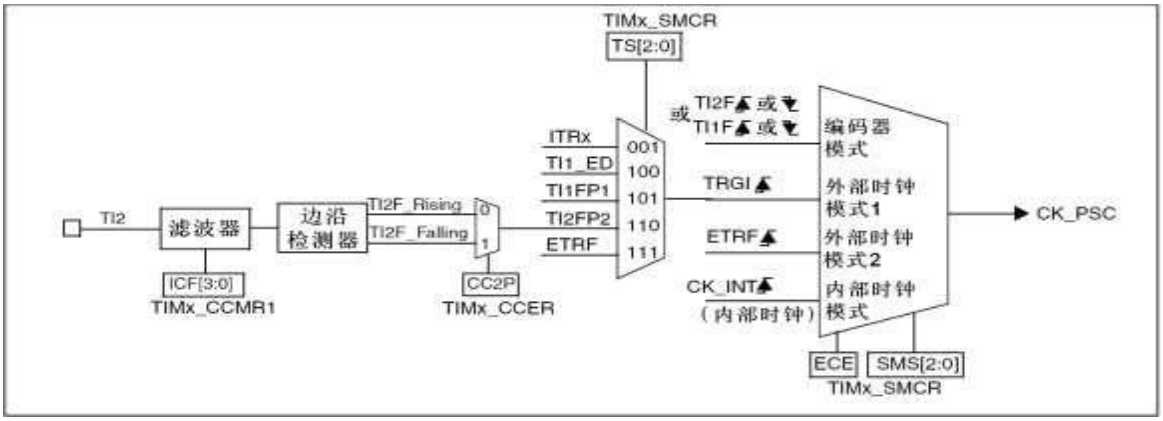


图 12-70 TI2 外部时钟连接例子

例如，要配置向上计数器在 T12 输入端的上升沿计数，使用下列步骤:

1. 配置 TIM2_CCMR1 寄存器CC2S ='01'，配置通道 2 检测 TI2 输入的上升沿
2. 配置 TIM2_CCMR1 寄存器的 IC2F[3:0]，选择输入滤波器带宽(如果不需要滤波器，保持IC2F = 0000)

注: 捕获预分频器不用作触发，所以不需要对它进行配置

3. 配置 TIM2_CCER 寄存器的CC2P ='0'，选定上升沿极性
4. 配置 TIM2_SMCR 寄存器的SMS ='111'，选择定时器外部时钟模式 1
5. 配置 TIM2_SMCR 寄存器中的TS ='110'，选定 TI2 作为触发输入源
6. 设置 TIM2_CR1 寄存器的CEN ='1'，启动计数器

当上升沿出现在 TI2，计数器计数一次，且 TIF 标志被设置。

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

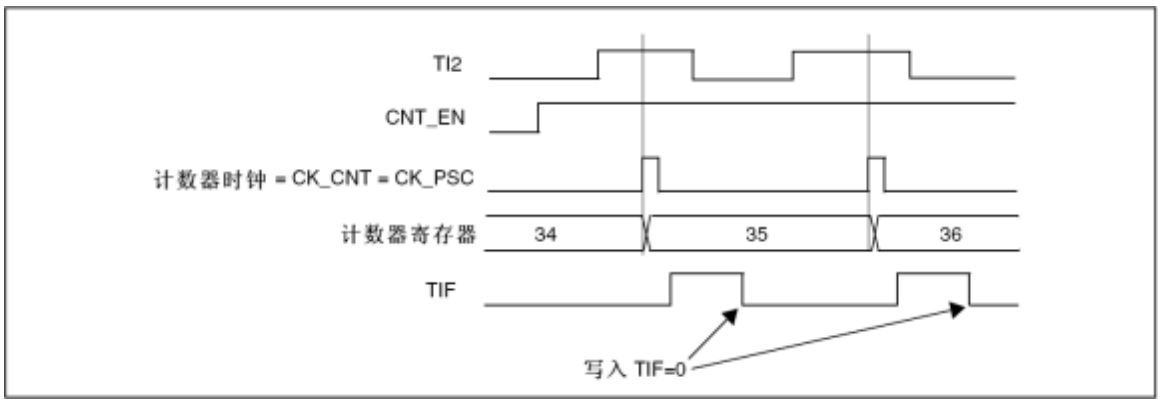


图 12-71 外部时钟模式 1 下的控制电路

12.8.3.3 外部时钟源模式 2

选定此模式的方法为: 令 TIM2_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:

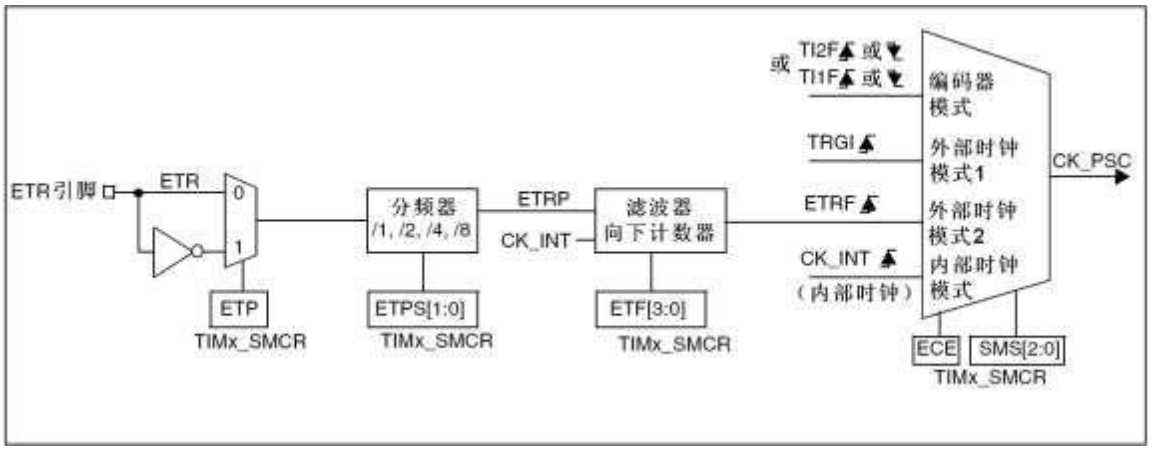


图 12-72 外部触发输入框图

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:

1. 本例中不需要滤波器，置 TIM2_SMCR 寄存器中的 ETF [3:0] = 0000
2. 设置预分频器，置 TIM2_SMCR 寄存器中的 ETPS [1:0] = 01
3. 设置在 ETR 的上升沿检测，置 TIM2_SMCR 寄存器中的 ETP = 0
4. 开启外部时钟模式 2，置 TIM2_SMCR 寄存器中的 ECE = 1
5. 启动计数器，置 TIM2_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。

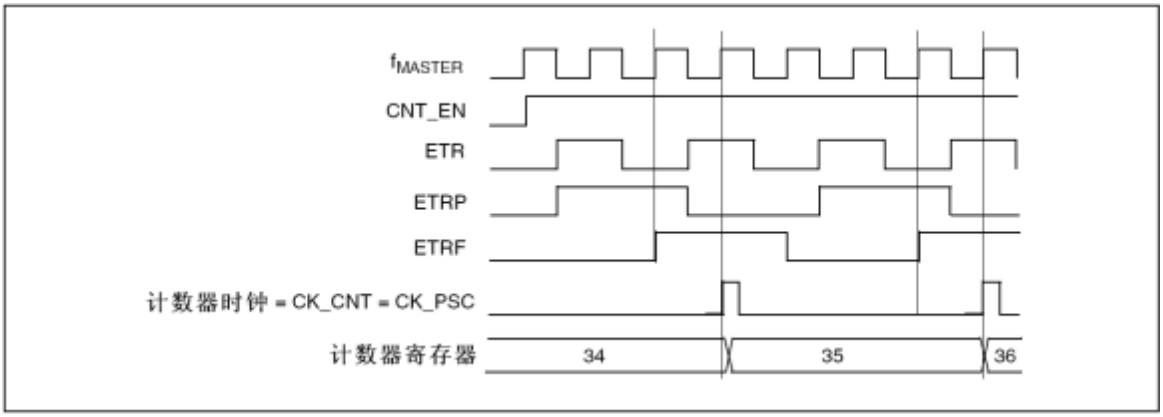


图 12-73 外部时钟模式 2 下的控制电路

©Copyright 2026, PADAUK Technology Co. Ltd

12.8.4 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) 捕获通道的来源由 SYSCFG_TIM2_CON_SEL 设定。

下面几张图是一个捕获/比较通道概览。

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

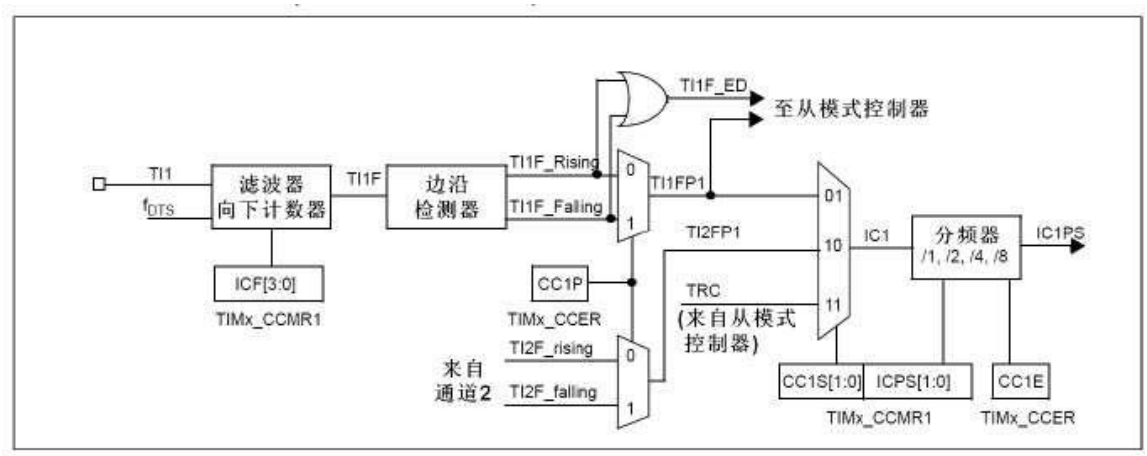


图 12-74 捕获/比较通道(如：通道 1 输入部分)

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。

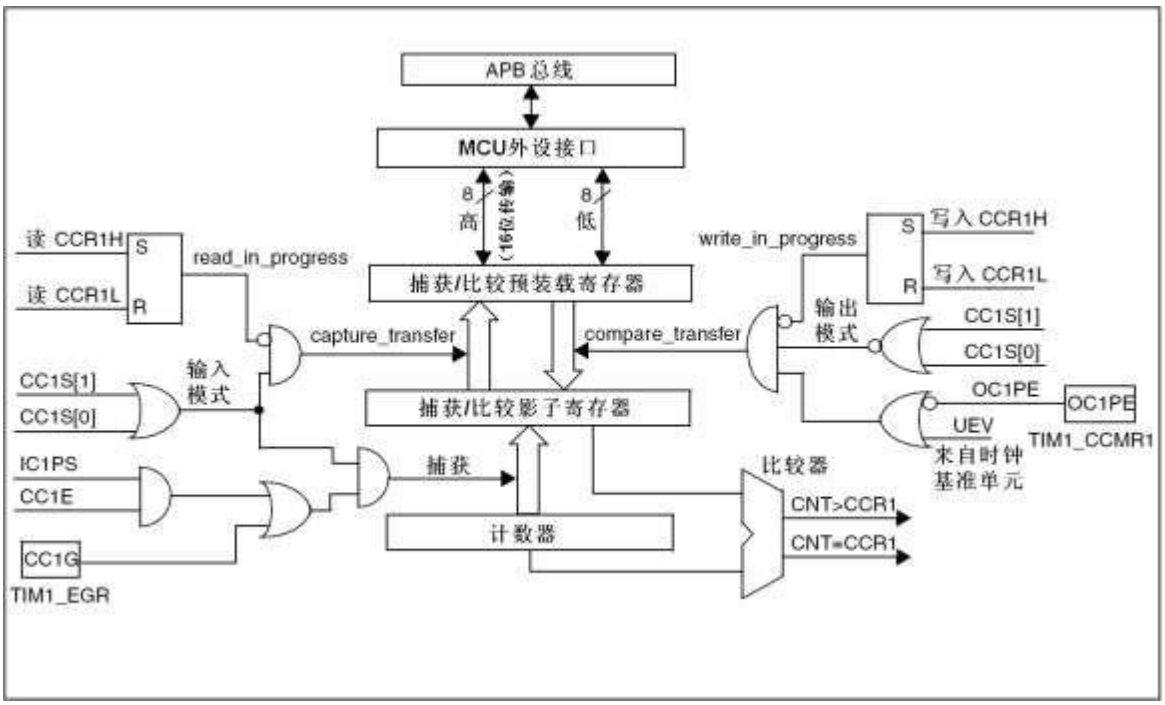


图 12-75 捕获/比较通道 1 的主电路

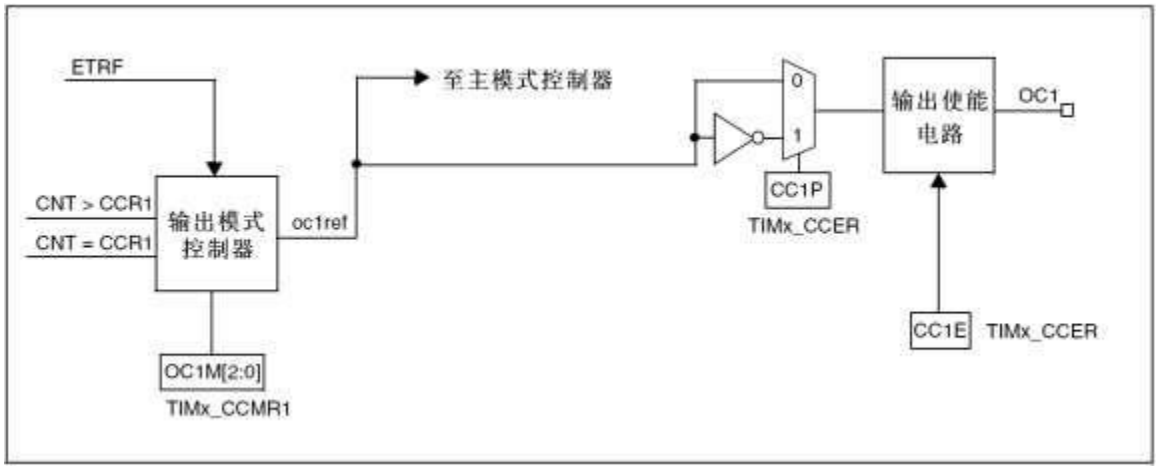


图 12-76 捕获/比较通道的输出部分(通道 1)

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

©Copyright 2026, PADAUK Technology Co. Ltd

12.8.5 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(TIM2_CCRx)中。当捕获事件发生时，相应的 CCxIF 标志(TIM2_SR 寄存器)被置'1'，如果使能了中断，则将产生中断。如果捕获事件发生时 CCxIF 标志已经为高，那么重复捕获标志 CCxOF(TIM2_SR 寄存器)被置'1'。写 CCxIF = 0 可清除 CCxIF，或读取存储在 TIM2_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 TIM2_CCR1 寄存器中，步骤如下:

- 选择有效输入端: TIM2_CCR1 必须连接到 TI1 输入，所以写入 TIM2_CCMR1 寄存器中的 CC1S=01，只要 CC1S 不为'00'，通道被配置为输入，并且 TIM2_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TIM2_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 fDTS 频率)连续采样 8 次，以确认在TI1 上一次真实的边沿变换，即在TIM2_CCMR1 寄存器中写入IC1F = 0011。
- 选择 TI1 通道的有效转换边沿，在 TIM2_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 TIM2_CCMR1 寄存器的 IC1PS = 00)。
- 设置 TIM2_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 TIM2_DIER 寄存器中的 CC1IE 位允许相关中断请求。当发生一个输入捕获时：
- 产生有效的电平转换时，计数器的值被传送到 TIM2_CCR1 寄存器。
- CC1IF 标志被设置(中断标志)。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置'1'。
- 如设置了CC1IE 位，则会产生一个中断。

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

注: 设置 TIM2_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断。

©Copyright 2026, PADAUK Technology Co. Ltd

12.8.6 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(TIM2_CCR1 寄存器)和占空比(TIM2_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 TIM2_CCR1 的有效输入：置 TIM2_CCMR1 寄存器的 CC1S = 01(选择 TI1)。
- 选择 TI1FP1 的有效极性(用来捕获数据到 TIM2_CCR1 中和清除计数器)：置 CC1P = 0(上升沿有效)。
- 选择 TIM2_CCR2 的有效输入：置 TIM2_CCMR1 寄存器的 CC2S = 10(选择 TI1)。
- 选择 TI1FP2 的有效极性(捕获数据到 TIM2_CCR2)：置 CC2P = 1(下降沿有效)。
- 选择有效的触发输入信号：置 TIM2_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 TIM2_SMCR 中的 SMS = 100。
- 使能捕获：置 TIM2_CCER 寄存器中 CC1E = 1 且CC2E = 1。

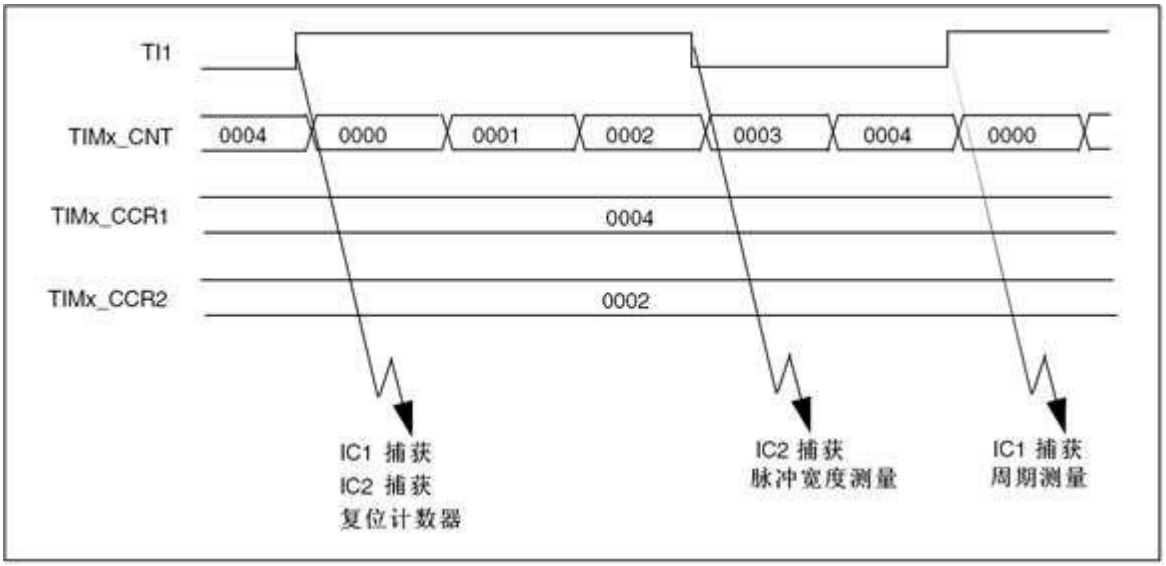


图 12-77 PWM 输入模式时序

由于只有 TI1FP1 和 TI2FP2 连到了从模式控制器， 所以 PWM 输入模式只能使用 TIM2_CH1/TIM2_CH2 信号。

12.8.7 强置输出模式

在输出模式(TIM2_CCMRx 寄存器中 CCxS = 00)下，输出比较信号(OCxREF 和相应的 OCx)能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。置 TIM2_CCMRx 寄存器中相应的 OCxM = 101，即可强置输出比较信号(OCxREF/OCx)为有效状态。这样 OCxREF 被强置为高电平(OCxREF 始终为高电平有效)，同时 OCx 得到 CCxP 极性位相反的值。例如：CCxP = 0(OCx 高电平有效)，则 OCx 被强置为高电平。置 TIM2_CCMRx 寄存器中的 OCxM = 100，可强置 OCxREF 信号为低。该模式下，在 TIM2_CCRx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断。这将会在下面的输出比较模式一节中介绍

12.8.8 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(TIM2_CCMRx 寄存器中的 OCxM 位)和输出极性(TIM2_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(TIM2_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽(TIM2_DIER 寄存器中的 CCxIE 位)，则产生一个中断。

TIM2_CCMRx 中的 OCxPE 位选择 TIM2_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 TIM2_ARR 和 TIM2_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，例如当计数器 CNT 与 CCRx 匹配时翻转 OCx 的输出引脚，CCRx 预装载未用，开启 OCx 输出且高电平有效，则必须设置 OCxM = 011、OCxPE = 0 、CCxP = 0 和CCxE = 1
5. 设置 TIM2_CR1 寄存器的 CEN 位启动计数器

©Copyright 2026, PADAUK Technology Co. Ltd

TIM2_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 TIM2_CCRx 影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。

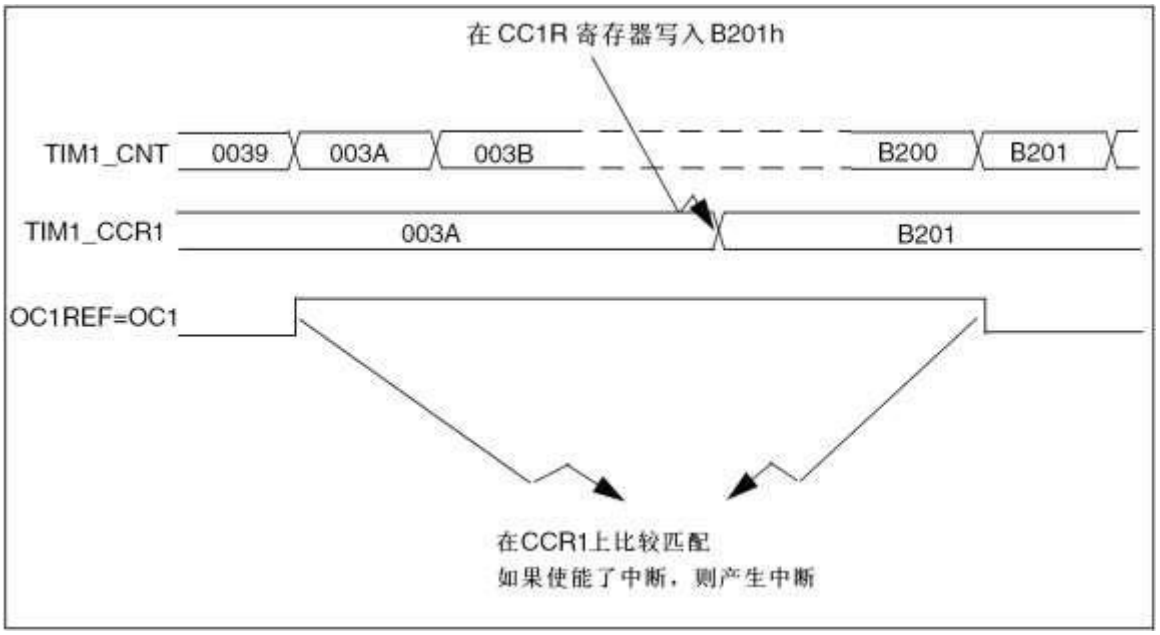


图 12-78 输出比较模式，翻转 OC1

12.8.9 PWM 模式

脉冲宽度调制模式可以产生一个由 TIM2_ARR 寄存器确定频率、由 TIM2_CCRx 寄存器确定占空比的信号。在 TIM2_CCMRx 寄存器中的 OCxM 位写入'110'(PWM 模式 1)或'111'(PWM 模式 2)，能够独立地设置每个 OCx 输出通道产生一路 PWM。必须设置 TIM2_CCMRx 寄存器 OCxPE 位以使能相应的预装载寄存器，最后还要设置 TIM2_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 TIM2_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 TIM2_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。TIM2_CCER 寄存器中的 CCxE 位控制 OCx 输出使能。详见 TIM2_CCERx 寄存器的描述。

在 PWM 模式(模式 1 或模式 2)下，TIM2_CNT 和 TIM2_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 TIM2_CCRx ≤ TIM2_CNT 或者 TIM2_CNT ≤ TIM2_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(TIM2_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 TIM2_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。

©Copyright 2026, PADAUK Technology Co. Ltd

12.8.9.1 PWM 边沿对齐模式

12.8.9.1.1 向上计数配置

当 TIM2_CR1 寄存器中的 DIR 位为低的时候执行向上计数。下面是一个 PWM 模式 1 的例子。当 TIM2_CNT < TIM2_CCRx 时 PWM 信号参考 OCxREF 为高，否则为低。如果 TIM2_CCRx 中的比较值大于自动重装载值(TIM2_ARR)，则 OCxREF 保持为'1'。如果比较值为 0，则 OCxREF 保持为'0'。下图为 TIM2_ARR = 0x8 时边沿对齐的 PWM 波形实例。

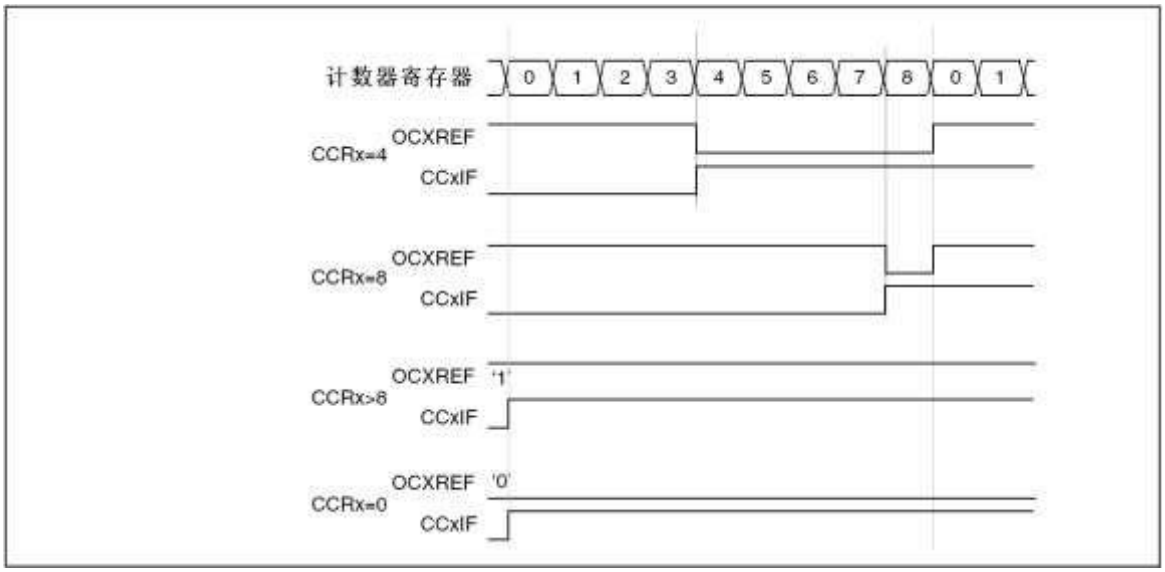


图 12-79 边沿对齐的 PWM 波形(ARR = 0x8)

12.8.9.1.2 向下计数的配置

当 TIM2_CR1 寄存器的 DIR 位为高时执行向下计数。在 PWM 模式 1，当 TIM2_CNT > TIM2_CCRx 时参考信号 OCxREF 为低，否则为高。如果 TIM2_CCRx 中的比较值大于 TIM2_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。

©Copyright 2026, PADAUK Technology Co. Ltd

12.8.9.2 PWM 中央对齐模式

当 TIM2_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'。TIM2_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

- TIM2_ARR = 0x8
- PWM 模式 1
- TIM2_CR1 寄存器中的CMS = 01，在中央对齐模式 1 时，当计数器向下计数时设置比较标志。

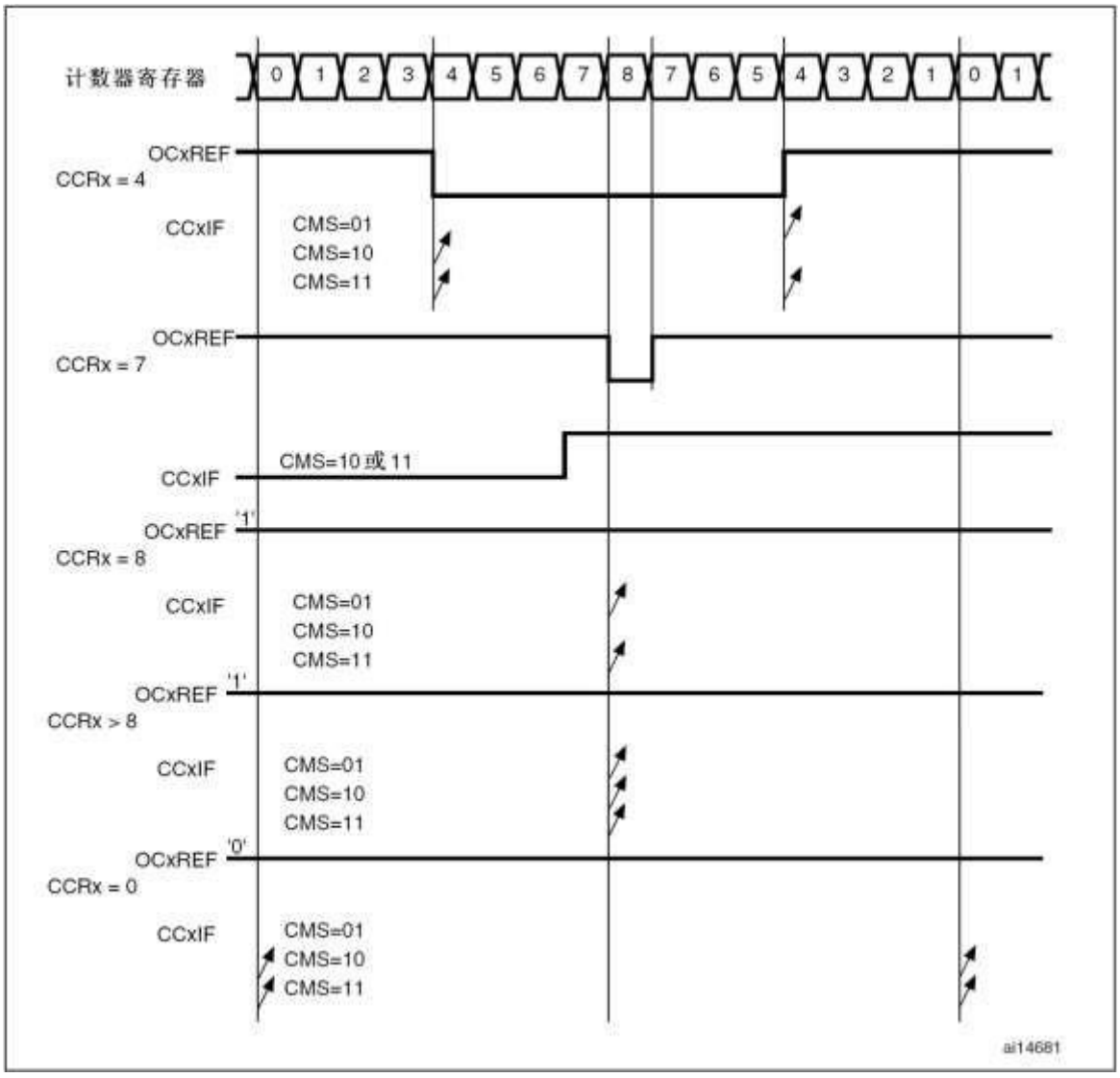


图 12-80 中央对齐的 PWM 波形(ARR = 0x8)

12.8.9.2.1 使用中央对齐模式的提示

1. 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 TIM2_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。
2. 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(TIM2_CNT > TIM2_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 TIM2_ARR 的值写入计数器，方向被更新，但不产生更新事件 UEV。
3. 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 TIM2_EGR 位中的 UG 位)，不要在计数进行过程中修改计数器的值。

12.8.9.3 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 TIM2_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: CNT < CCRx ≤ ARR(特别地，0 < CCRx)

向下计数方式: CNT > CCRx。

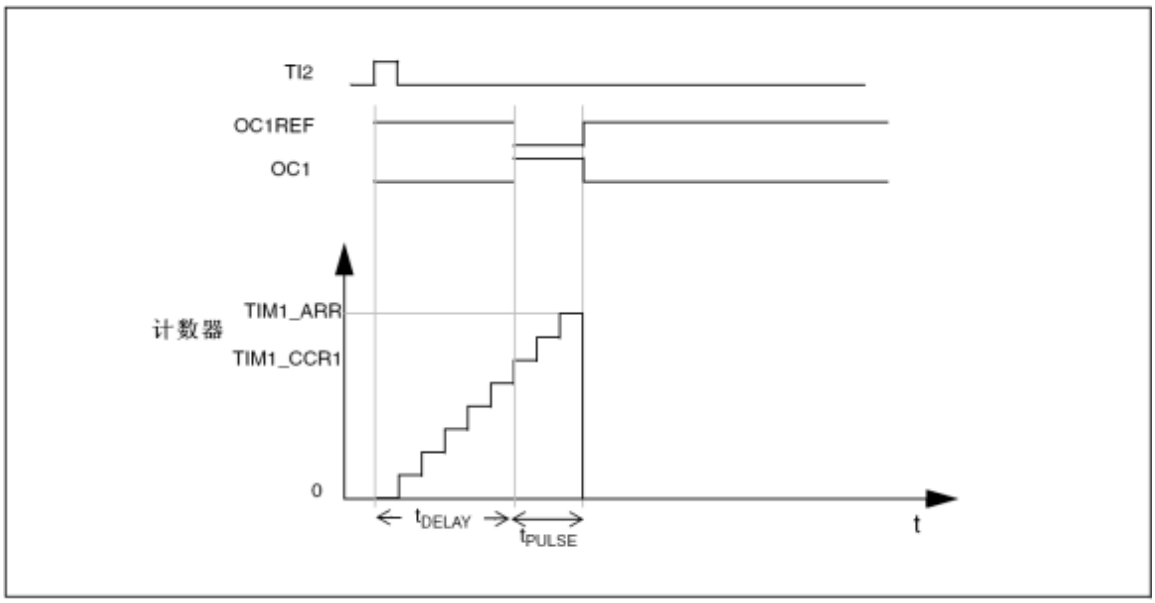


图 12-81 单脉冲模式的例子

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

假定 TI2FP2 作为触发 1:

- 置 TIM2_CCMR1 寄存器中的CC2S ='01'，把 TI2FP2 映像到 TI2。
- 置 TIM2_CCER 寄存器中的CC2P ='0'，使 TI2FP2 能够检测上升沿。
- 置 TIM2_SMCR 寄存器中的TS ='110'，TI2FP2 作为从模式控制器的触发(TRGI)。
- 置 TIM2_SMCR 寄存器中的SMS ='110'(触发模式)，TI2FP2 被用来启动计数器。

OPM 波形由写入比较寄存器的数值决定(要考虑时钟频率和计数器预分频器)

- tDELAY 由写入 TIM2_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(TIM2_ARR - TIM2_CCR1)。
- 假定当发生比较匹配时要产生从'0'到'1'的波形，当计数器到达预装载值时要产生一个从'1'到'0' 的波形；首先要置 TIM2_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 TIM2_CCMR1 中的 OC1PE='1'和 TIM2_CR1 寄存器中的 ARPE； 然后在 TIM2_CCR1 寄存器中填写比较值，在 TIM2_ARR 寄存器中填写自动装载值，修改 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，TIM2_CR1 寄存器中的 DIR 和 CMS 位应该置低。

因为只需一个脉冲，所以必须设置 TIM2_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。

12.8.9.4 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 TIM2_CCMRx 寄存器中的 OCxFE 位；此时 OCxREF(和 OCx) 被强制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。

12.8.10 在外部事件时清除 OCxREF 信号

对于一个给定的通道，设置 TIM2_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：TIM2_SMCR 寄存器中的ETPS[1:0] ='00'。
2. 必须禁止外部时钟模式 2：TIM2_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 TIM2 被置于 PWM 模式。

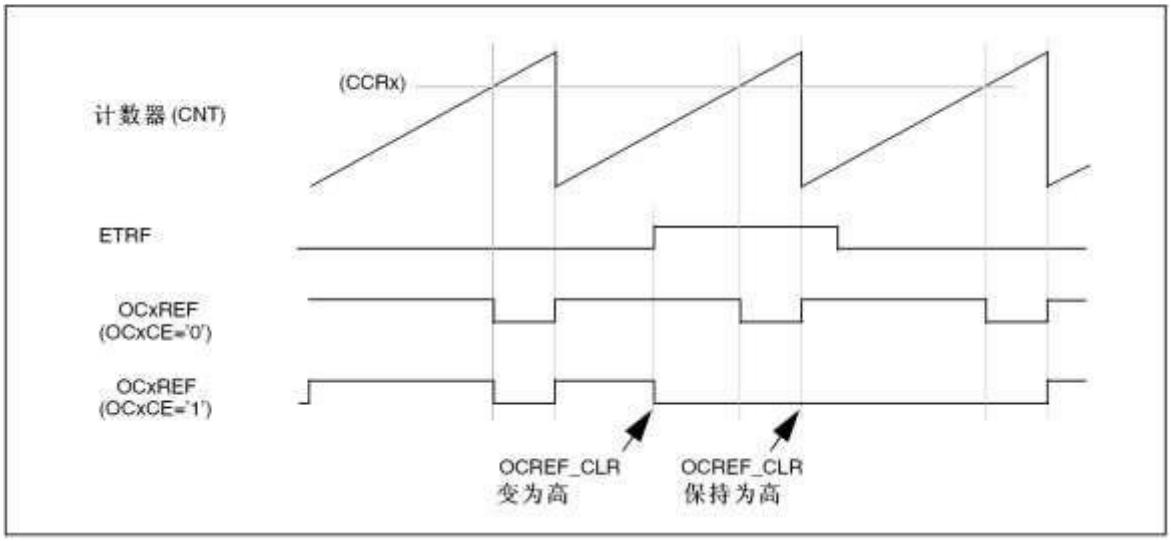


图 12-82 清除 TIM2 的 OCxREF

12.8.11 编码器接口模式

选择编码器接口模式的方法是：如果计数器只在 TI2 的边沿计数，则置 TIM2_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 TIM2_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 12-1，假定计数器已经启动(TIM2_CR1 寄存器中的 CEN ='1')，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 TIM2_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 TIM2_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 TIM2_ARR；同样，捕获器、比较器、预分频器、触发输出特性等仍工作如常。在这个模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。下表列出了所有可能的组合，假设 TI1 和 TI2 不同时变换。

下表為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

表 12-5 计数方向与编码器信号的关系

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。

©Copyright 2026, PADAUK Technology Co. Ltd

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’(TIM2_CCMR1 寄存器，IC1FP1 映射到 TI1)
- CC2S =‘01’(TIM2_CCMR2 寄存器，IC2FP2 映射到 TI2)
- CC1P =‘0’(TIM2_CCER 寄存器，IC1FP1 不反相，IC1FP1 = TI1)
- CC2P =‘0’(TIM2_CCER 寄存器，IC2FP2 不反相，IC2FP2 = TI2)
- SMS =‘011’(TIM2_SMCR 寄存器，所有的输入均在上升沿和下降沿有效)
- CEN =‘1’(TIM2_CR1 寄存器，计数器使能)

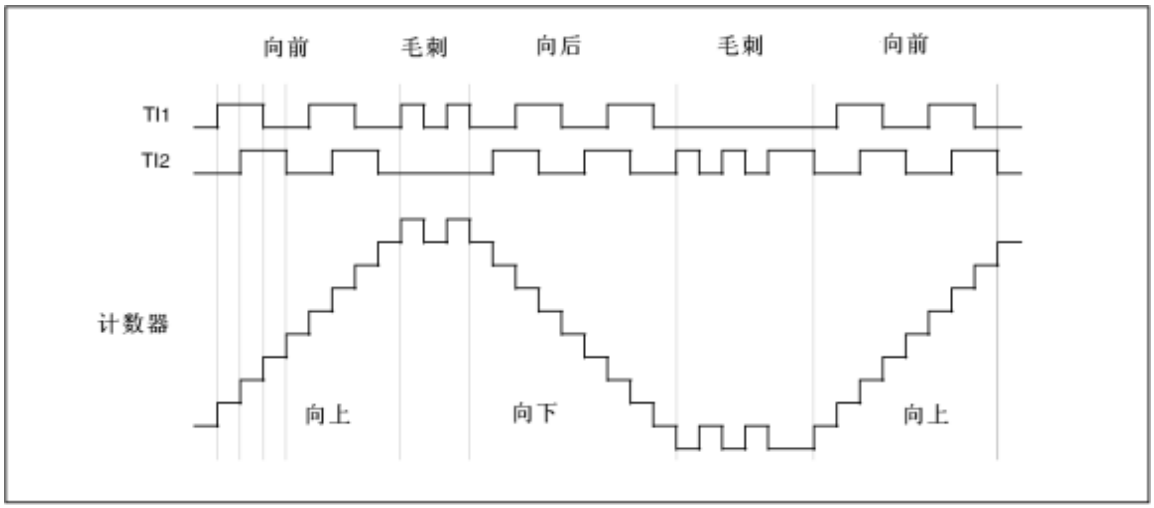


图 12-83 编码器模式下的计数器操作实例

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

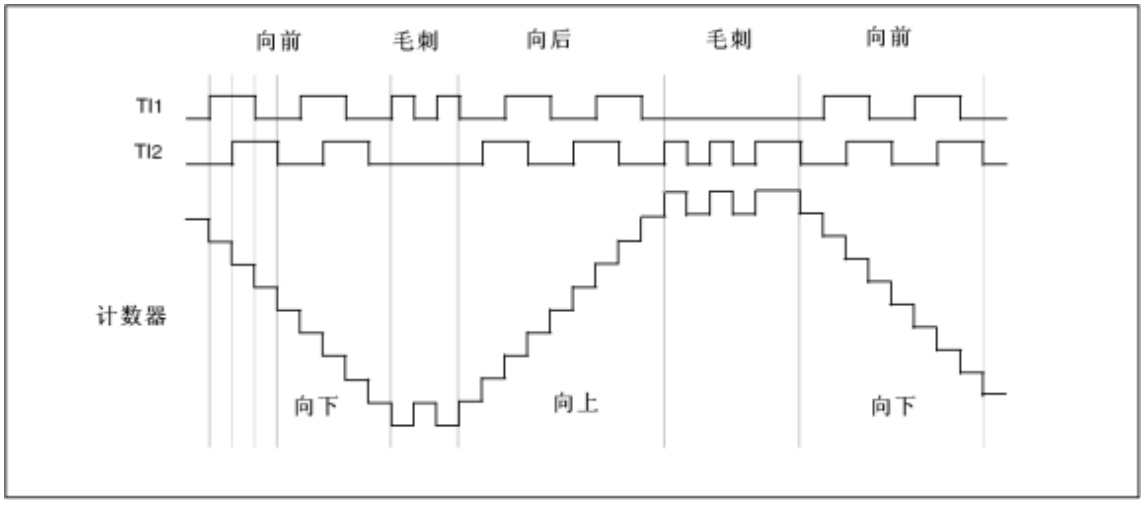


图 12-84 IC1FP1 反相的编码器接口模式实例

当定时器配置成编码器接口模式时，提供传感器当前位置的信息。使用第二个配置在捕获模式的定时器，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)。

12.8.12 定时器输入异或功能

TIM2_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 TIM2_CH1、TIM2_CH2 和 TIM2_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。

12.8.13 定时器和外部触发的同步

TIM2 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。

©Copyright 2026, PADAUK Technology Co. Ltd

12.8.13.1 从模式: 复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIM2_CR1 寄存器的 URS 位为低，还会产生一个更新事件 UEV；然后所有的预装载寄存器(TIM2_ARR ， TIM2_CCRx)都会被更新。

在下面的例子中，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，所以不需要配置它。CC1S 位只选择输入捕获源，即 TIM2_CCMR1 寄存器中 CC1S = 01。置 TIM2_CCER 寄存器中CC1P = 0 以确定极性(只检测上升沿)。
- 置 TIM2_SMCR 寄存器中 SMS = 100，配置定时器为复位模式；置 TIM2_SMCR 寄存器中TS = 101，选择 TI1 作为输入源。
- 置 TIM2_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(TIM2_SR 寄存器中的 TIF 位)被设置，根据 TIM2_DIER 寄存器中 TIE(中断使能)位，产生一个中断请求。

下图显示当自动重装载寄存器 TIM2_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。

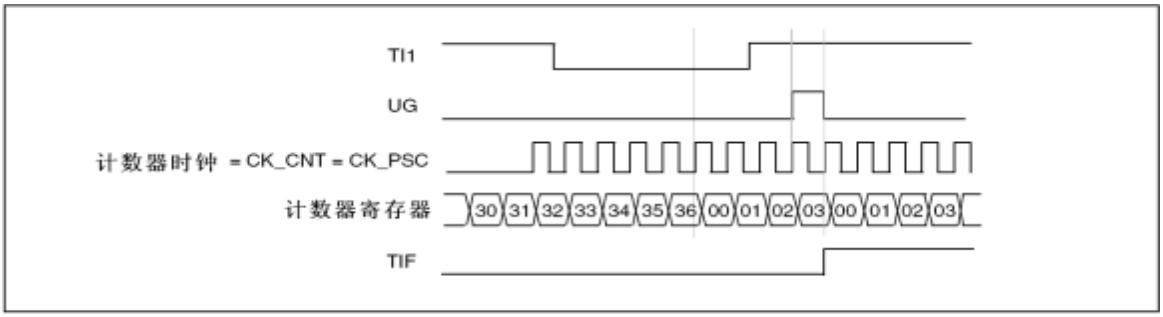


图 12-85 复位模式下的控制电路

12.8.13.2 从模式: 门控模式

按照选中的输入端电平使能计数器。在如下的例子中，计数器只在 TI1 为低时向上计数:

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽(本例中，不需要滤波，所以保持IC1F = 0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输入捕获源，置 TIM2_CCMR1 寄存器中CC1S = 01。置 TIM2_CCER 寄存器中CC1P = 1 以确定极性(只检测低电平)。
- 置 TIM2_SMCR 寄存器中 SMS = 101，配置定时器为门控模式；置 TIM2_SMCR 寄存器中 TS = 101，选择 TI1 作为输入源。
- 置TIM2_CR1 寄存器中CEN = 1，启动计数器。在门控模式下，如果 CEN = 0，则计数器不能启动，不论触发输入电平如何。

只要 TI1 为低，计数器开始依据内部时钟计数，在 TI1 变高时停止计数。当计数器开始或停止时都设置 TIM2_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。

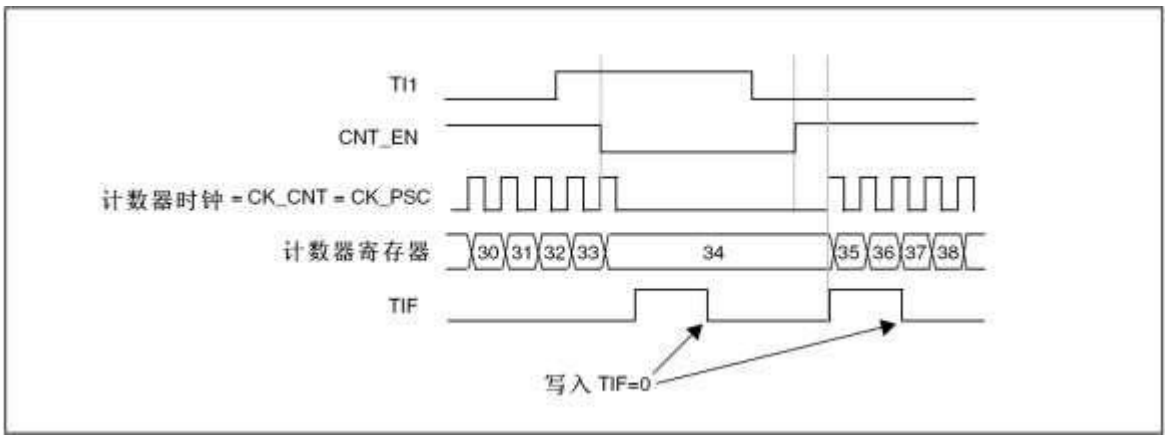


图 12-86 门控模式下的控制电路

12.8.13.3 从模式: 触发模式

输入端上选中的事件使能计数器。在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数:

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽(本例中，不需要任何滤波器，保持IC2F = 0000)。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源， 置 TIM2_CCMR1 寄存器中 CC2S = 01。置 TIM2_CCER 寄存器中 CC2P = 1 以确定极性(只检测低电平)。
- 置 TIM2_SMCR 寄存器中 SMS = 110，配置定时器为触发模式；置 TIM2_SMCR 寄存器中 TS = 110，选择TI2 作为输入源。

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

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

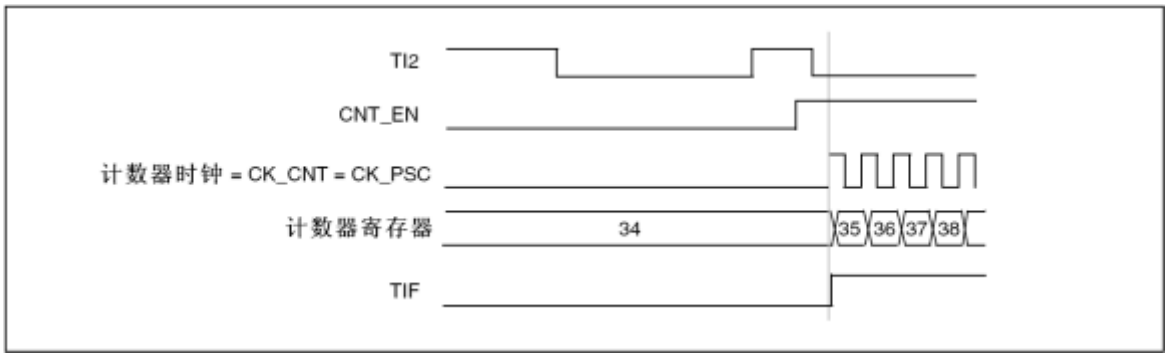


图 12-87 触发器模式下的控制电路

12.8.13.4 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时可以选择另一个输入作为触发输入。不建议使用 TIM2_SMCR 寄存器的 TS 位选择 ETR 作为 TRGI。

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

1. 通过 TIM2_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 TIM2_CCMR1 寄存器中 CC1S = 01，选择输入捕获源
   - 置 TIM2_CCER 寄存器中 CC1P = 0 以确定极性(只检测上升沿)
3. 置 TIM2_SMCR 寄存器中 SMS = 110，配置定时器为触发模式。置 TIM2_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。

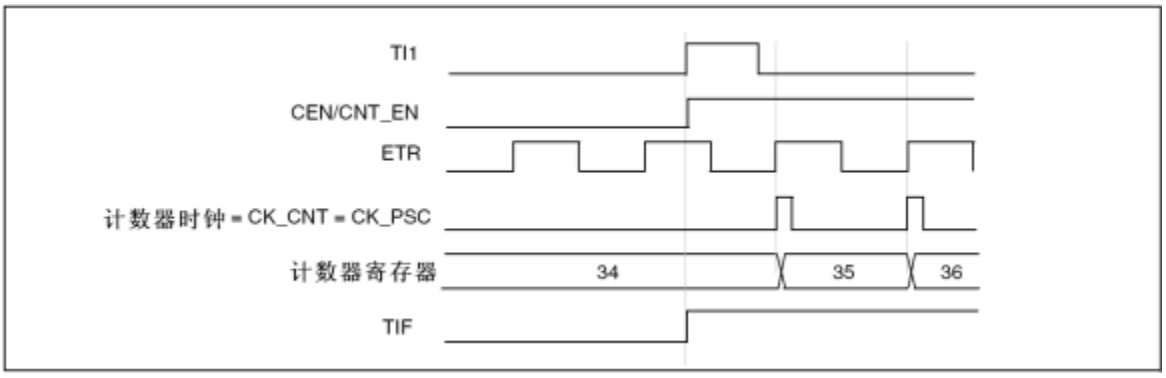


图 12-88 外部时钟模式 2 + 触发模式下的控制电路

12.8.14 定时器同步

所有 TIM2 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。

12.8.14.1 使用一个定时器作为另一个定时器的预分频器

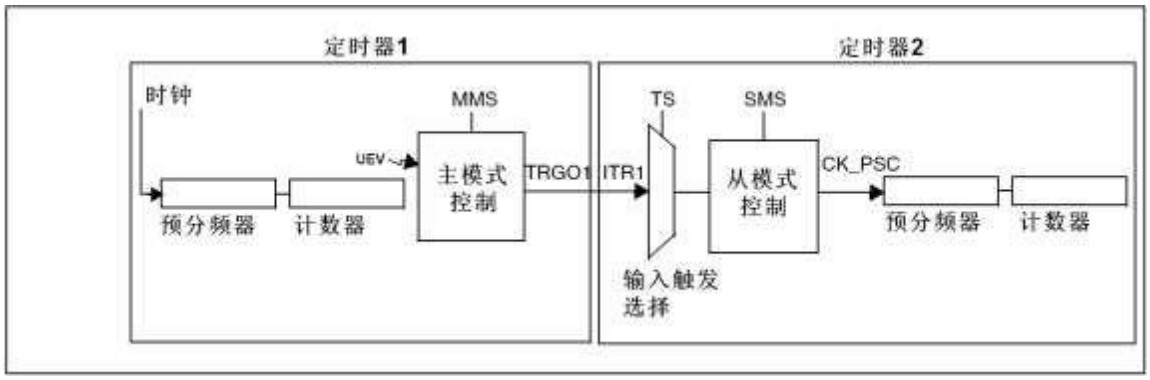


图 12-89 主 / 从定时器的例子

如可以配置定时器 1 作为定时器 2 的预分频器。参考图 12-90，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EPWM_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(TIM2_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。

©Copyright 2026, PADAUK Technology Co. Ltd

12.8.14.2 使用一个定时器使能另一个定时器

在这个例子中，定时器 2 的使能由定时器 1 的输出比较控制。参考图 12-90 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 ($f_{CK\_CNT} = f_{CK\_INT} / 3$) 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(EPWM_CR2 寄存器的 MMS = 100)
- 配置定时器 1 的 OC1REF 波形(EPWM_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 EPWM_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。

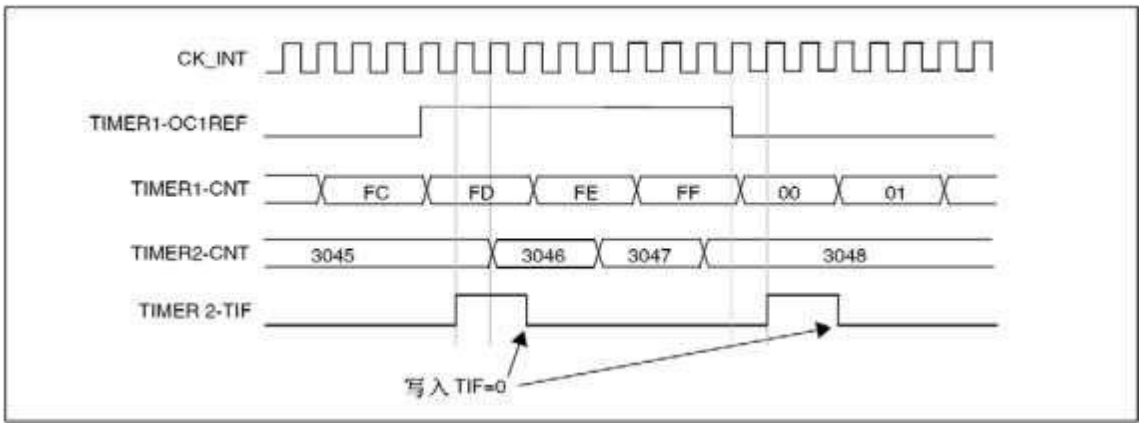


图 12-90 定时器 1 的 OC1REF 控制定时器 2

在图 12-91 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 TIM2_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 EPWM_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(EPWM_CR2 寄存器的 MMS = 100)。
- 配置定时器 1 的 OC1REF 波形(EPWM_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EPWM_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 EPWM_CR1 寄存器的CEN ='1'以启动定时器 1。
- 置 EPWM_CR1 寄存器的CEN ='0'以停止定时器 1。

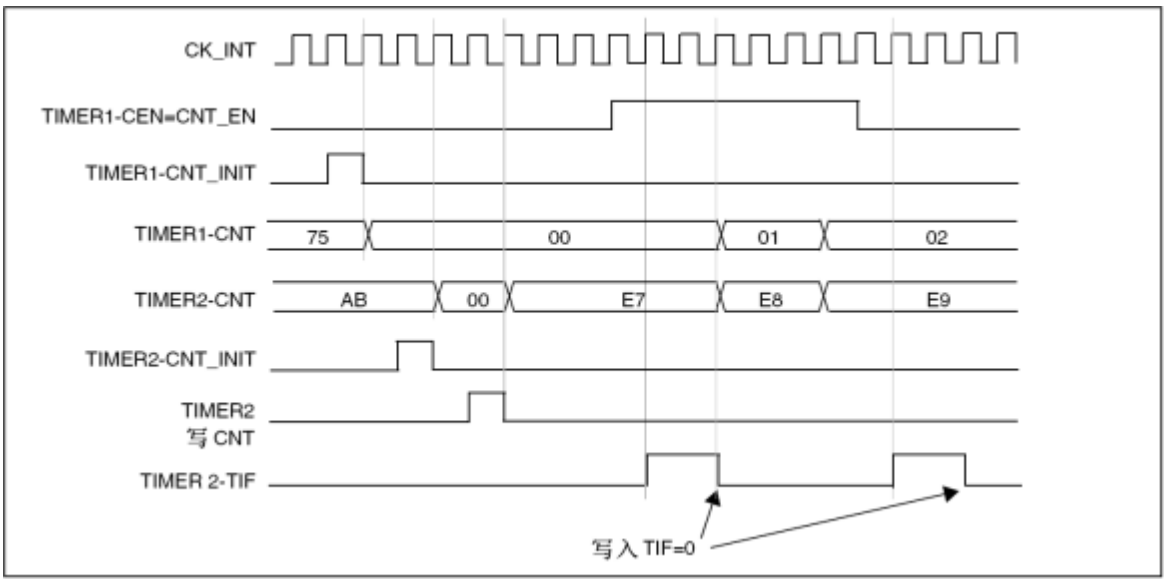


图 12-91 通过使能定时器 1 可以控制定时器 2

12.8.14.3 使用一个定时器去启动另一个定时器

在这个例子中，使用定时器 1 的更新事件使能定时器 2。参考图 12-90 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(EPWM_CR2 寄存器的 MMS=010)。
- 配置定时器 1 的周期(EPWM_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 EPWM_CR1 寄存器的 CEN = 1 以启动定时器 1。

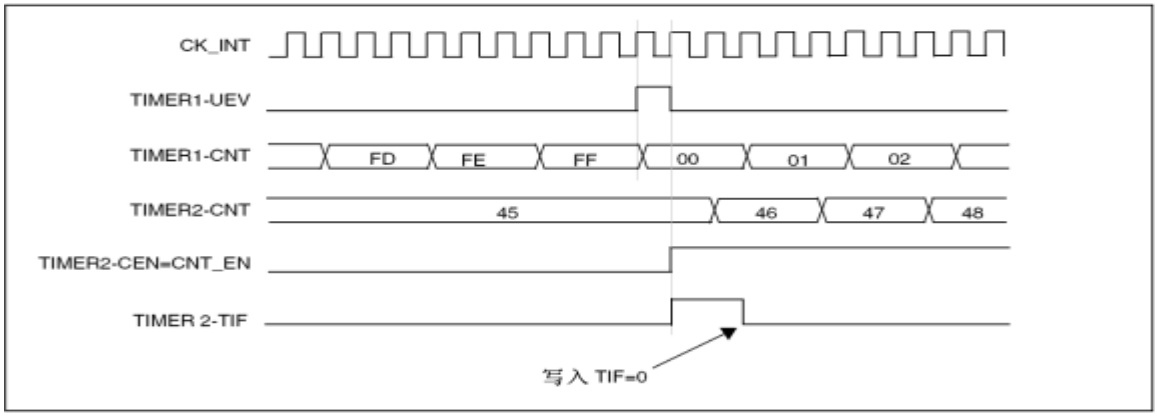


图 12-92 使用定时器 1 的更新触发定时器 2

在上一个例子中，可以在启动计数之前初始化两个计数器。图 12-94 显示在相同配置情况下，使用触发模式而不是门控模式(TIM2_SMCR 寄存器的 SMS = 110)的动作。

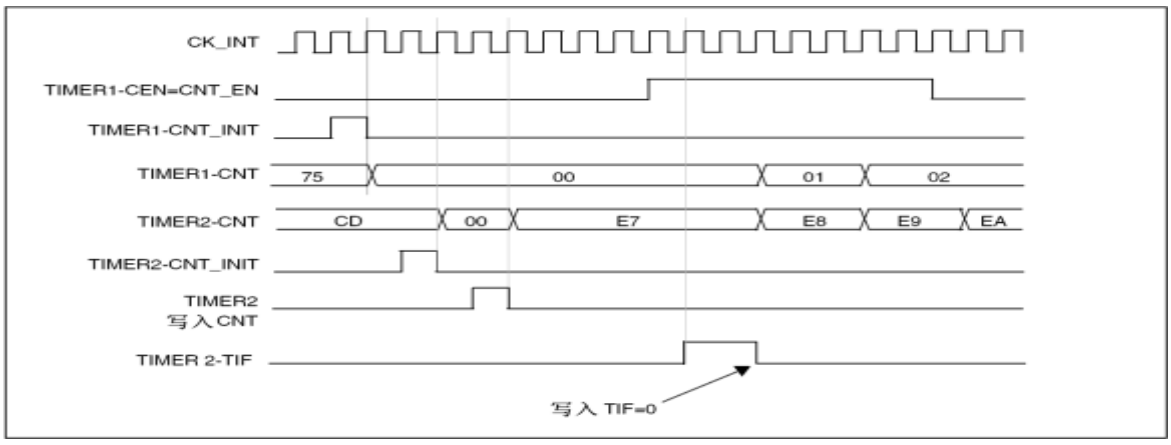


图 12-93 利用定时器 1 的使能触发定时器 2

12.8.14.4 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。参考图 12-90 的连接，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(EPWM_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(EPWM_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 EPWM_CR2 寄存器的CEN = 1 以启动定时器 2。
- 置 EPWM_CR1 寄存器的CEN = 1 以启动定时器 1。

©Copyright 2026, PADAUK Technology Co. Ltd

12.8.14.5 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 2，参见图 12-90。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(EPWM_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(EPWM_SMCR 寄存器的TS ='100')
- 配置定时器 1 为触发模式(EPWM_SMCR 寄存器的SMS ='110')
- 配置定时器 1 为主/从模式，EPWM_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(TIM2_CNT)在定时器间插入一个偏移。下图中能看到主/从模式下在定时器 1 的 CNT_EN 和 CK_PSC 之间有个延迟。

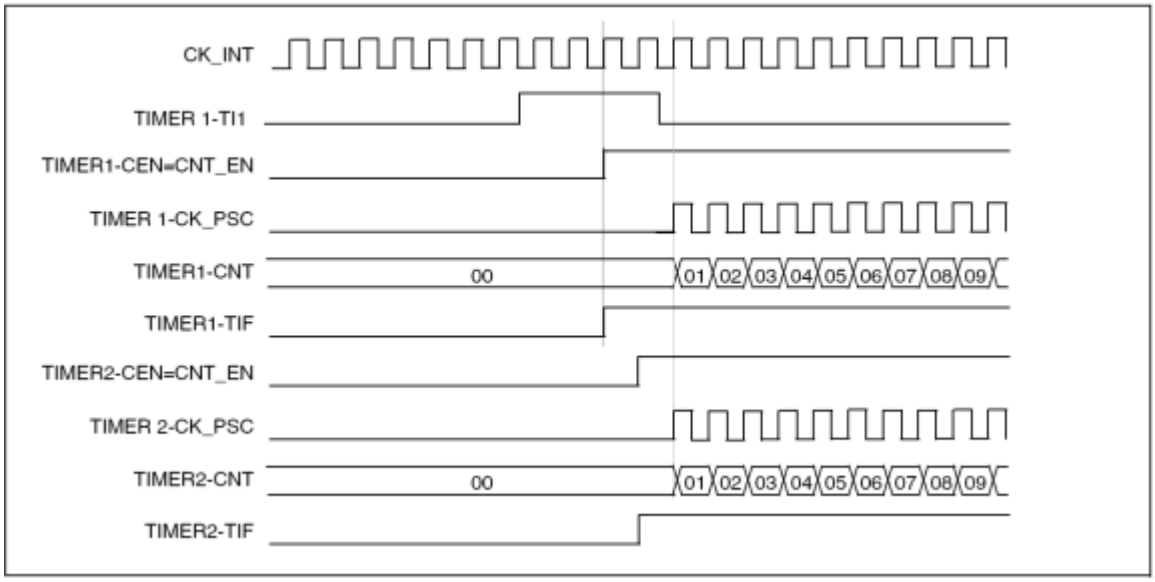


图 12-94 使用定时器 1 的 TI1 输入触发定时器 1 和定时器 2

12.8.15 调试模式

当微控制器进入调试模式，根据 DBG 模块中 DBG_TIM2_STOP 的设置，TIM2 计数器或者继续正常操作，或者停止。

12.9 TIM2 寄存器列表

可以用字(32 位)的方式操作这些外设寄存器。TIM2 基地址 0x4000 1000

地址	寄存器	描述
0x40001000	TIM2_CR1	TIM2 控制寄存器 1
0x40001004	TIM2_CR2	TIM2 控制寄存器 2
0x40001008	TIM2_SMCR	TIM2 从模式控制寄存器
0x4000100C	TIM2_IER	TIM2 中断使能寄存器
0x40001010	TIM2_SR	TIM2 状态寄存器
0x40001014	TIM2_EGR	TIM2 事件产生寄存器
0x40001018	TIM2_CCMR1	TIM2 捕获/比较模式寄存器 1
0x4000101C	TIM2_CCMR2	TIM2 捕获/比较模式寄存器 2
0x40001020	TIM2_CCER	TIM2 捕获/比较使能寄存器
0x40001024	TIM2_CNT	TIM2 计数器
0x40001028	TIM2_PSC	TIM2 预分频器
0x4000102C	TIM2_ARR	TIM2 自动重装载寄存器
0x40001030	-	保留
0x40001034	TIM2_CCR1	TIM2 捕获/比较寄存器 1
0x40001038	TIM2_CCR2	TIM2 捕获/比较寄存器 2
0x4000103C	TIM2_CCR3	TIM2 捕获/比较寄存器 3
0x40001040	TIM2_CCR4	TIM2 捕获/比较寄存器 4

表 12-6 TIM2 寄存器列表

©Copyright 2026, PADAUK Technology Co. Ltd

12.10 TIM2 寄存器描述

12.10.1 TIM2 控制寄存器 1(TIM2_CR1)

Bit	Name	R/W	Reset	Description
31:11	-	R	0x0	保留
10	ASYMEN	R/W	0x0	TIM2 中心对齐方式下不对称计数使能 0: 对称计数使能 1: 不对称计数使能
9:8	CKD	R/W	0x0	时钟分频因子(Clock division) 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。 00: tDTS = tCK_INT 01: tDTS = 2 x tCK_INT 10: tDTS = 4 x tCK_INT 11: 保留，不要使用这个配置
7	ARPE	R/W	0x0	自动重装载预装载允许位(Auto-reload preload enable) 0: TIM2_ARR 寄存器没有缓冲 1: TIM2_ARR 寄存器被装入缓冲器
6:5	CMS	R/W	0x0	选择中央对齐模式(Center-aligned mode selection) 00: 边沿对齐模式。计数器依据方向位(DIR)向上或向下计数。 01: 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(TIM2_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。 10: 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(TIM2_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。 11: 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(TIM2_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	R/W	0x0	方向(Direction) 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	R/W	0x0	单脉冲模式(One pulse mode) 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	R/W	0x0	更新请求源(Update request source) 软件通过该位选择 UEV 事件源 0: 如果使能了更新中断，则下述任一事件产生更新中断： – 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，则只有计数器溢出/下溢才产生更新中断。
1	UDIS	R/W	0x0	禁止更新(Update disable) 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	R/W	0x0	使能计数器(Counter enable) 0: 禁止计数器 1: 使能计数器 注: 在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

12.10.2 TIM2 控制寄存器 2 (TIM2_CR2)

Bit	Name	R/W	Reset	Description
31:8	-	R	0x0	保留
7	TI1S	R/W	0x0	TI1 选择(TI1 selection) 0: TIM2_CH1 引脚连到 TI1 输入； 1: TIM2_CH1、TIM2_CH2 和 TIM2_CH3 引脚经异或后连到 TI1 输入。
6:4	MMS	R/W	0x0	主模式选择(Master mode selection) 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下： 000: 复位 – TIM2_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式)， 则 TRGO 上的信号相对实际的复位会有一个延迟。 001: 使能– 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。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 TIM2_SMCR 寄存器中 MSM 位的描述)。 010: 更新 – 更新事件被选为触发输入(TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 011: 比较脉冲 – 在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)。 100: 比较 – OC1REF 信号被用于作为触发输出(TRGO)。 101: 比较 – OC2REF 信号被用于作为触发输出(TRGO)。 110: 比较 – OC3REF 信号被用于作为触发输出(TRGO)。 111: 比较 – OC4REF 信号被用于作为触发输出(TRGO)。

3	-	-	0x0	保留
2	CCUS	R/W	0x0	捕获/比较控制更新选择(Capture/compare control update selection) 0: 如果捕获/比较控制位是预装载的(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	-	R	0x0	保留
0	CCPC	R/W	0x0	捕获/比较预装载控制位(Capture/compare preloaded control) 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。

12.10.3 TIM2 从模式控制寄存器 (TIM2_SMCR)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15	ETP	R/W	0x0	外部触发极性(External trigger polarity) 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	R/W	0x0	外部时钟使能位(External clock enable) 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	R/W	0x0	外部触发预分频(External trigger prescaler) 外部触发信号 ETRP 的频率必须最多是 TIM2CLK 频率的 1/4。 当输入较快的外部时钟时，可以使用预分频降低 ETRP 的频率。 00: 关闭预分频； 01: ETRP 频率除以 2； 10: ETRP 频率除以 4； 11: ETRP 频率除以 8。

©Copyright 2026, PADAUK Technology Co. Ltd

11:8	ETF	R/W	0x0	外部触发滤波(External trigger filter) 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。 0000: 无滤波器，以 fDTS 采样 0001: 采样频率 fSAMPLING = fCK_INT，N = 2 0010: 采样频率 fSAMPLING = fCK_INT，N = 4 0011: 采样频率 fSAMPLING = fCK_INT，N = 8 0100: 采样频率 fSAMPLING = fDTS/2，N = 6 0101: 采样频率 fSAMPLING = fDTS/2，N = 8 0110: 采样频率 fSAMPLING = fDTS/4，N = 6 0111: 采样频率 fSAMPLING = fDTS/4，N = 8 1000: 采样频率 fSAMPLING = fDTS/8，N = 6 1001: 采样频率 fSAMPLING = fDTS/8，N = 8 1010: 采样频率 fSAMPLING = fDTS/16，N = 5 1011: 采样频率 fSAMPLING = fDTS/16，N = 6 1100: 采样频率 fSAMPLING = fDTS /16，N = 8 1101: 采样频率 fSAMPLING = fDTS /32，N = 5 1110: 采样频率 fSAMPLING = fDTS /32，N = 6 1111: 采样频率 fSAMPLING = fDTS /32，N = 8
7	MSM	R/W	0x0	主/从模式(Master/slave mode) 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	R/W	0x0	触发选择(Trigger selection) 这 3 位选择用于同步计数器的触发输入。 000: 内部触发 0(ITR0)100: TI1 的边沿检测器(TI1F_ED) 001: 内部触发 1(ITR1)101: 滤波后的定时器输入 1(TI1FP1) 010: 内部触发 2(ITR2)110: 滤波后的定时器输入 2(TI2FP2) 011: 内部触发 3(ITR3)111: 外部触发输入(ETRF) 更多有关 ITRx 的细节，参见表 12-1。 注: 这些位只能在未用到(如 SMS = 000)时被改变，以避免在改变时产生错误的边沿检测。
3	-	R	0x0	保留
2:0	SMS	R/W	0x0	从模式选择(Slave mode selection) 当选择了外部信号，触发信号(TRGI)的有效边沿与选中的外部输入极性相关(见输入控制寄存器和控制寄存器的说明) 000: 关闭从模式 – 如果 CEN=1，则预分频器直接由内部时钟驱动。 001: 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。 010: 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。 011: 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。 100: 复位模式 – 选中的触发输入(TRGI)的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。 101: 门控模式 – 当触发输入(TRGI)为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止(但不复位)。计数器的启动和停止都是受控的。 110: 触发模式 – 计数器在触发输入 TRGI 的上升沿启动(但不复位)， 只有计数器的启动是受控的。 111: 外部时钟模式 1 – 选中的触发输入(TRGI)的上升沿驱动计数器。

				注: 如果 TI1F_EN 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。

从定时器	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM2	EPWM_trgo	irq_timer0	irq_timer1	irq_lptimer

表 12-7 TIM2 内部触发连接

注: 如果某个产品中没有相应的定时器，则对应的触发信号 ITRx 也不存在。

12.10.4 TIM2 中断使能寄存器 (TIM2_IER)

Bit	Name	R/W	Reset	Description
31:14	-	-	0x0	保留
13	CCD4IE	R/W	0x0	允许比较 4 中断 (ASYMEN = 1 和 向下计数模式有效) 0: 禁止捕获/比较 4 中断； 1: 允许捕获/比较 4 中断。
12	UDIE	R/W	0x0	0: 禁止下溢 中断； 1: 允许下溢 中断
11	OVIE	R/W	0x0	0: 禁止上溢 中断； 1: 允许上溢 中断
10	CCD3IE	R/W	0x0	允许比较 3 中断 (ASYMEN = 1 和 向下计数模式有效) 0: 禁止捕获/比较 3 中断； 1: 允许捕获/比较 3 中断。
9	CCD2IE	R/W	0x0	允许比较 2 中断 (ASYMEN = 1 和 向下计数模式有效) 0: 禁止捕获/比较 2 中断； 1: 允许捕获/比较 2 中断。
8	CCD1IE	R/W	0x0	允许比较 1 中断 (ASYMEN = 1 和 向下计数模式有效) 0: 禁止捕获/比较 1 中断； 1: 允许捕获/比较 1 中断。
7	-	R	0x0	保留
6	TIE	R/W	0x0	触发中断使能(Trigger interrupt enable) 0: 禁止触发中断； 1: 使能触发中断。
5	-	-	0x0	保留

4	CC4IE	R/W	0x0	允许捕获/比较 4 中断(Capture/Compare 4 interrupt enable) 0: 禁止捕获/比较 4 中断； 1: 允许捕获/比较 4 中断。
3	CC3IE	R/W	0x0	允许捕获/比较 3 中断(Capture/Compare 3 interrupt enable) 0: 禁止捕获/比较 3 中断； 1: 允许捕获/比较 3 中断。
2	CC2IE	R/W	0x0	允许捕获/比较 2 中断(Capture/Compare 2 interrupt enable) 0: 禁止捕获/比较 2 中断； 1: 允许捕获/比较 2 中断。
1	CC1IE	R/W	0x0	允许捕获/比较 1 中断(Capture/Compare 1 interrupt enable) 0: 禁止捕获/比较 1 中断； 1: 允许捕获/比较 1 中断。
0	UIE	R/W	0x0	允许更新中断(Update interrupt enable) 0: 禁止更新中断； 1: 允许更新中断。

©Copyright 2026, PADAUK Technology Co. Ltd

12.10.5 TIM2 状态寄存器 (TIM2_SR)

Bit	Name	R/W	Reset	Description
31:13	-	R	0x0	保留
18	CC4DIF	R/W0c	0x0	1: TIM2_CNT 向下计数模式的值与 TIM2_CCDR4 的值匹配 CCD2IF 位变高 0: 无匹配发生(写 0 清除)
17	UDIF	R/W0c	0x0	1: TIM2_CNT 向下计数模下溢位 0:无下溢位
16	OVIF	R/W0c	0x0	1: TIM2_CNT 向上计数模上溢位 0:无上溢位
15	CC3DIF	R/W0c	0x0	1: TIM2_CNT 向下计数模式的值与 TIM2_CCDR3 的值匹配 CCD2IF 位变高 0: 无匹配发生(写 0 清除)
14	CC2DIF	R/W0c	0x0	1: TIM2_CNT 向下计数模式的值与 TIM2_CCDR2 的值匹配 CCD2IF 位变高 0: 无匹配发生 (写 0 清除)
13	CC1DIF	R/W0c	0x0	1: TIM2_CNT 向下计数模式的值与 TIM2_CCDR1 的值匹配 CCD1IF 位变高 0: 无匹配发生(写 0 清除)
12	CC4OF	R/W0c	0x0	捕获/比较 4 重复捕获标记(Capture/Compare 4 overcapture flag) 参见 CC1OF 描述。
11	CC3OF	R/W0c	0x0	捕获/比较 3 重复捕获标记(Capture/Compare 3 overcapture flag) 参见 CC1OF 描述。

10	CC2OF	R/W0c	0x0	捕获/比较 2 重复捕获标记(Capture/Compare 2 overcapture flag) 参见 CC1OF 描述。
9	CC1OF	R/W0c	0x0	捕获/比较 1 重复捕获标记(Capture/Compare 1 overcapture flag) 仅当相应的通道被配置为输入捕获时，该标记可由硬件置 1。 写 0 可清除该位。 0: 无重复捕获产生； 1: 计数器的值被捕获到 TIM2_CCR1 寄存器时，CC1IF 的状态已经为'1'。
8:7	-	R	0x0	保留
6	TIF	R/W0c	0x0	触发器中断标记(Trigger interrupt flag) 当发生触发事件(当从模式控制器处于除门控模式外的其它模式时，在 TRGI 输入端检测到有效边沿，或门控模式下的任一边沿)时由硬件对该位置'1'。它由软件清'0'。 0: 无触发器事件产生； 1: 触发中断等待响应。
5	0	R	0x0	保留
4	CC4IF	R/W0c	0x0	捕获/比较 4 中断标记(Capture/Compare 4 interrupt flag) 参考 CC1IF 描述。
3	CC3IF	R/W0c	0x0	捕获/比较 3 中断标记(Capture/Compare 3 interrupt flag) 参考 CC1IF 描述。
2	CC2IF	R/W0c	0x0	捕获/比较 2 中断标记(Capture/Compare 2 interrupt flag) 参考 CC1IF 描述。

©Copyright 2026, PADAUK Technology Co. Ltd

1	CC1IF	R/W0c	0x0	捕获/比较 1 中断标记(Capture/Compare 1 interrupt flag) 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 TIM2_CR1 寄存器的 CMS 位)。它由软件清'0'。 0: 无匹配发生； 1: TIM2_CNT 的值与 TIM2_CCR1 的值匹配。 当 TIM2_CCR1 的内容大于 TIM2_APR 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 TIM2_CCR1 清'0'。 0: 无输入捕获产生； 1: 计数器值已被捕获(拷贝)至 TIM2_CCR1(在 IC1 上检测到与所选极性相同的边沿)。
0	UIF	R/W0c	0x0	更新中断标记(Update interrupt flag) 当产生更新事件时该位由硬件置'1'。它由软件清'0'。 0: 无更新事件产生； 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 TIM2_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 TIM2_CR1 寄存器的 URS=0、UDIS=0，当设置 TIM2_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 TIM2_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 13.5.3：从模式控制寄存器(TIM2_SMCR))。

12.10.6 TIM2 事件产生寄存器 (TIM2_EGR)

Bit	Name	R/W	Reset	Description
31:7	-	R	0x0	保留
6	TG	WO	0x0	产生触发事件(Trigger generation) 该位由软件置'1'，用于产生一个触发事件，由硬件自动清'0'。 0: 无动作； 1: TIM2_SR 寄存器的 TIF=1，若开启对应的中断，则产生相应的中断。
5	-	R	0x0	保留
4	CC4G	WO	0x0	产生捕获/比较 4 事件(Capture/Compare 4 generation) 参考 CC1G 描述。
3	CC3G	WO	0x0	产生捕获/比较 3 事件(Capture/Compare 3 generation) 参考 CC1G 描述。
2	CC2G	WO	0x0	产生捕获/比较 2 事件(Capture/Compare 2 generation) 参考 CC1G 描述。
1	CC1G	WO	0x0	产生捕获/比较 1 事件(Capture/Compare 1 generation) 该位由软件置'1'，用于产生一个捕获/比较事件，由硬件自动清'0'。 0: 无动作； 1: 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，则产生相应的中断。 若通道 CC1 配置为输入： 当前的计数器值被捕获至 TIM2_CCR1 寄存器；设置 CC1IF = 1，若开启对应的中断，则产生相应的中断。若 CC1IF 已经为 1，则设置 CC1OF = 1。
0	UG	WO	0x0	产生更新事件(Update generation) 该位由软件置'1'，由硬件自动清'0'。 0: 无动作；

				1: 重新初始化计数器，并产生一个更新事件。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 TIM2_ARR 的值。

12.10.7 TIM2 捕获/比较模式寄存器 1 (TIM2_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:

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15	OC2CE	R/W	0x0	输出比较 2 清 0 使能(Output Compare 2 clear enable)
14:12	OC2M	R/W	0x0	输出比较 2 模式(Output Compare 2 mode)
11	OC2PE	R/W	0x0	输出比较 2 预装载使能(Output Compare 2 preload enable)
10	OC2FE	R/W	0x0	输出比较 2 快速使能(Output Compare 2 fast enable)
9:8	CC2S	R/W	0x0	捕获/比较 2 选择。(Capture/Compare 2 selection) 该位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC2 通道被配置为输出； 01: CC2 通道被配置为输入，IC2 映射在 TI2 上； 10: CC2 通道被配置为输入，IC2 映射在 TI1 上； 11: 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 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时(TIM2_CCER 寄存器的 CC2E = 0)才是可写的。
7	OC1CE	R/W	0x0	输出比较 1 清'0'使能(Output Compare 1 clear enable) 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	R/W	0x0	输出比较 1 模式(Output Compare 1 mode) 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。 000: 冻结。输出比较寄存器 TIM2_CCR1 与计数器 TIM2_CNT 间的比较对 OC1REF 不起作用； 001: 匹配时设置通道 1 为有效电平。当计数器 TIM2_CNT 的值与捕获/比较寄存器 1(TIM2_CCR1)相同时，强制 OC1REF 为高。 010: 匹配时设置通道 1 为无效电平。当计数器 TIM2_CNT 的值与捕获/比较寄存器 1(TIM2_CCR1)相同时，强制 OC1REF 为低。 011: 翻转。当 TIM2_CCR1=TIM2_CNT 时，翻转 OC1REF 的电平。 100: 强制为无效电平。强制 OC1REF 为低。 101: 强制为有效电平。强制 OC1REF 为高。 110: PWM 模式 1－ 在向上计数时，一旦 TIM2_CNT < TIM2_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 TIM2_CNT>TIM2_CCR1 时通道 1 为无效电平(OC1REF = 0)，否则为有效电平(OC1REF = 1)。 111: PWM 模式 2－ 在向上计数时，一旦 TIM2_CNT < TIM2_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 TIM2_CNT > TIM2_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(TIM2_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	R/W	0x0	输出比较 1 预装载使能(Output Compare 1 preload enable) 0: 禁止 TIM2_CCR1 寄存器的预装载功能，可随时写入 TIM2_CCR1 寄存器，并且新写入的数值立即起作用。 1: 开启 TIM2_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIM2_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(TIM2_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(TIM2_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	R/W	0x0	输出比较 1 快速使能(Output Compare 1 fast enable) 该位用于加快 CC 输出对触发输入事件的响应。 0: 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	R/W	0x0	捕获/比较 1 选择。(Capture/Compare 1 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC1 通道被配置为输出； 01: CC1 通道被配置为输入，IC1 映射在 TI1 上； 10: CC1 通道被配置为输入，IC1 映射在 TI2 上； 11: 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 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC1S 仅在通道关闭时(TIM2_CCER 寄存器的 CC1E = 0)才是可写的。

输入捕获模式:

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:12	IC2F	R/W	0x0	输入捕获 2 滤波器(Input capture 2 filter)
11:10	IC2PSC	R/W	0x0	输入/捕获 2 预分频器(Input capture 2 prescaler)
9:8	CC2S	R/W	0x0	捕获/比较 2 选择(Capture/Compare 2 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择: 00: CC2 通道被配置为输出； 01: CC2 通道被配置为输入，IC2 映射在 TI2 上； 10: CC2 通道被配置为输入，IC2 映射在 TI1 上； 11: 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 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时(TIM2_CCER 寄存器的 CC2E = 0)才是可的。

Bit	Name	R/W	Reset	Description
7:4	IC1F	R/W	0x0	输入捕获 1 滤波器(Input capture 1 filter) 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变： 0000: 无滤波器，以 fDTS 采样 0001: 采样频率 fSAMPLING = fCK_INT，N = 2 0010: 采样频率 fSAMPLING = fCK_INT，N = 4 0011: 采样频率 fSAMPLING = fCK_INT，N = 8 0100: 采样频率 fSAMPLING = fDTS/2，N = 6 0101: 采样频率 fSAMPLING = fDTS/2，N = 8 0110: 采样频率 fSAMPLING = fDTS/4，N = 6 0111: 采样频率 fSAMPLING = fDTS/4，N = 8 1000: 采样频率 fSAMPLING = fDTS/8，N = 6 1001: 采样频率 fSAMPLING = fDTS/8，N = 8 1010: 采样频率 fSAMPLING = fDTS/16，N = 5 1011: 采样频率 fSAMPLING = fDTS/16，N = 6 1100: 采样频率 fSAMPLING = fDTS/16，N = 8 1101: 采样频率 fSAMPLING = fDTS/32，N = 5 1110: 采样频率 fSAMPLING = fDTS/32，N = 6 1111: 采样频率 fSAMPLING = fDTS/32，N = 8

3:2	IC1PSC	R/W	0x0	输入/捕获 1 预分频器(Input capture 1 prescaler) 这 2 位定义了 CC1 输入(IC1)的预分频系数。 一旦 CC1E = 0(TIM2_CCER 寄存器中)，则预分频器复位。 00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获； 01: 每 2 个事件触发一次捕获； 10: 每 4 个事件触发一次捕获； 11: 每 8 个事件触发一次捕获。

©Copyright 2026, PADAUK Technology Co. Ltd

1:0	CC1S	R/W	0x0	捕获/比较 1 选择(Capture/Compare 1 Selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC1 通道被配置为输出； 01: CC1 通道被配置为输入，IC1 映射在 TI1 上； 10: CC1 通道被配置为输入，IC1 映射在 TI2 上； 11: 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 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC1S 仅在通道关闭时(TIM2_CCER 寄存器的 CC1E = 0)才是可写的。

12.10.8 TIM2 捕获/比较模式寄存器 2(TIM2_CCMR2)

输出比较模式:

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15	OC4CE	R/W	0x0	输出比较 4 清 0 使能(Output Compare 4 clear enable)
14:12	OC4M	R/W	0x0	输出比较 4 模式(Output Compare 4 mode)
11	OC4PE	R/W	0x0	输出比较 4 预装载使能(Output Compare 4 preload enable)
10	OC4FE	R/W	0x0	输出比较 4 快速使能(Output Compare 4 fast enable)
9:8	CC4S	R/W	0x0	捕获/比较 4 选择。(Capture/Compare 4 selection) 该 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC4 通道被配置为输出； 01: CC4 通道被配置为输入，IC4 映射在 TI4 上； 10: CC4 通道被配置为输入，IC4 映射在 TI3 上； 11: 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 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(TIM2_CCER 寄存器的 CC4E = 0)才是可写的。
7	OC3CE	R/W	0x0	输出比较 3 清'0'使能(Output Compare 3 clear enable)
6:4	OC3M	R/W	0x0	输出比较 3 模式(Output Compare 3 mode)
3	OC3PE	R/W	0x0	输出比较 3 预装载使能(Output Compare 3 preload enable)
2	OC3FE	R/W	0x0	输出比较 3 快速使能(Output Compare 3 fast enable)
1:0	CC3S [1:0]	R/W	0x0	捕获/比较 3 选择。(Capture/Compare 3 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC3 通道被配置为输出； 01: CC3 通道被配置为输入，IC3 映射在 TI3 上； 10: CC3 通道被配置为输入，IC3 映射在 TI4 上； 11: 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 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC3S 仅在通道关闭时(TIM2_CCER 寄存器的 CC3E = 0)才是可写的。

输入捕获模式:

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:12	IC4F	R/W	0x0	输入捕获 4 滤波器(Input capture 4 filter)
11:10	IC4PSC	R/W	0x0	输入/捕获 4 预分频器(Input capture 4 prescaler)
9:8	CC4S	R/W	0x0	捕获/比较 4 选择(Capture/Compare 4 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC4 通道被配置为输出； 01: CC4 通道被配置为输入，IC4 映射在 TI4 上； 10: CC4 通道被配置为输入，IC4 映射在 TI3 上； 11: 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 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(TIM2_CCER 寄存器的 CC4E = 0)才是可写的。
7:4	IC3F	R/W	0x0	输入捕获 3 滤波器(Input capture 3 filter)
3:2	IC3PSC	R/W	0x0	输入/捕获 3 预分频器(Input capture 3 prescaler)
1:0	CC3S	R/W	0x0	捕获/比较 3 选择(Capture/Compare 3 Selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC3 通道被配置为输出； 01: CC3 通道被配置为输入，IC3 映射在 TI3 上； 10: CC3 通道被配置为输入，IC3 映射在 TI4 上； 11: 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 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC3S 仅在通道关闭时(TIM2_CCER 寄存器的 CC3E = 0)才是可写的。

©Copyright 2026, PADAUK Technology Co. Ltd

12.10.9 TIM2 捕获/比较使能寄存器 (TIM2_CCER)

Bit	Name	R/W	Reset	Description
31:14	-	R	0x0	保留
13	CC4P	R/W	0x0	输入/捕获 4 输出极性(Capture/Compare 4 output polarity) 参考 CC1P 的描述。
12	CC4E	R/W	0x0	输入/捕获 4 输出使能(Capture/Compare 4 output enable) 参考 CC1E 的描述。
11:10	-	R	0x0	保留
9	CC3P	R/W	0x0	输入/捕获 3 输出极性(Capture/Compare 3 output polarity) 参考 CC1P 的描述。
8	CC3E	R/W	0x0	输入/捕获 3 输出使能(Capture/Compare 3 output enable) 参考 CC1E 的描述。
7:6	-	R	0x0	保留
5	CC2P	R/W	0x0	输入/捕获 2 输出极性(Capture/Compare 2 output polarity) 参考 CC1P 的描述。
4	CC2E	R/W	0x0	输入/捕获 2 输出使能(Capture/Compare 2 output enable) 参考 CC1E 的描述。
3:2	-	R	0x0	保留

©Copyright 2026, PADAUK Technology Co. Ltd

1	CC1P	R/W	0x0	输入/捕获 1 输出极性(Capture/Compare 1 output polarity) CC1 通道配置为输出： 0: OC1 高电平有效 1: OC1 低电平有效 CC1 通道配置为输入： 该位选择是 IC1 还是 IC1 的反相信号作为触发或捕获信号。 0: 不反相：捕获发生在 IC1 的上升沿；当用作外部触发器时，IC1 不反相。 1: 反相：捕获发生在 IC1 的下降沿；当用作外部触发器时，IC1 反相。 注: 一旦 LOCK 级别(TIM2_BDTR 寄存器中的 LOCK 位)设为 3 或 2，则该位不能被修改。
0	CC1E	R/W	0x0	输入/捕获 1 输出使能(Capture/Compare 1 output enable) 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 TIM2_CCR1 寄存器。 0: 捕获禁止； 1: 捕获使能。

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

表 12-8 标准 OCx 通道的输出控制位

注: 连接到标准 OCx 通道的外部 I/O 引脚状态，取决于 OCx 通道状态和 GPIO 以及 AFIO 寄存器。

12.10.10 TIM2 计数器 (TIM2_CNT)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:0	CNT	R/W	0x0	计数器的值(Counter value)

12.10.11 TIM2 预分频器 (TIM2_PSC)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:0	PSC	R/W	0x0	预分频器的值 (Prescaler value) 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'。

12.10.12 TIM2 自动重装载寄存器 (TIM2_ARR)

Bit	Name	R/W	Reset	Description
31:16	-	-	0x0	保留
15:0	ARR	R/W	0x0	自动重装载的值(Auto-reload value) ARR 包含了将要装载入实际的自动重装载寄存器的值。 详细参考 13.3.1 节：有关 ARR 的更新和动作。 当自动重装载的值为空时，计数器不工作。

12.10.13 TIM2 捕获/比较寄存器 1 (TIM2_CCR1)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:0	CCR1	R/W	0x0	捕获/比较通道 1 的值(Capture/Compare 1 value) 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 TIM2_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 TIM2_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。

12.10.14 TIM2 捕获/比较寄存器 2 (TIM2_CCR2)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:0	CCR2	R/W	0x0	捕获/比较通道 2 的值(Capture/Compare 2 value) 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 TIM2_CCMR1 寄存器(OC2PE 位)中未选择预装载特性， 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 TIM2_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。

12.10.15 TIM2 捕获/比较寄存器 3(TIM2_CCR3)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:0	CCR3	R/W	0x0	捕获/比较通道 3 的值(Capture/Compare 3 value) 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 TIM2_CCMR2 寄存器(OC3PE 位)中未选择预装载特性， 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 TIM2_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。

12.10.16 TIM2 捕获/比较寄存器 4 (TIM2_CCR4)

Bit	Name	R/W	Reset	Description
31:16	0x0	R	-	保留
15:0	CCR4	R/W	0x0	捕获/比较通道 4 的值(Capture/Compare 4 value) 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 TIM2_CCMR2 寄存器(OC4PE 位)中未选择预装载特性， 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 TIM2_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。

12.10.17 TIM2 计数器向下比较寄存器1 (TIM2_CCDR1)

Bit	Name	R/W	Reset	Description
31:20	-	R	0x0	保留
19:0	CCDR1	R/W	0x0	TIM2 计数器向下比较值(比较通道 1 的值) (TIM2 中心对齐模式下不对称计数使能有效)

12.10.18 TIM2 计数器向下比较寄存器2 (TIM2_CCDR2)

Bit	Name	R/W	Reset	Description
31:20	-	R	0x0	保留
19:0	CCDR2	R/W	0x0	TIM2 计数器向下比较值(比较通道 2 的值) (TIM2 中心对齐模式下不对称计数使能有效)

12.10.19 TIM2 计数器向下比较寄存器3 (TIM2_CCDR3)

Bit	Name	R/W	Reset	Description
31:20	-	R	0x0	保留
19:0	CCDR3	R/W	0x0	TIM2 计数器向下比较值(比较通道 3 的值) (TIM2 中心对齐模式下不对称计数使能有效)

12.10.20 TIM2 计数器向下比较寄存器4 (TIM2_CCDR4)

Bit	Name	R/W	Reset	Description
31:20	-	R	0x0	保留
19:0	CCDR4	R/W	0x0	TIM2 计数器向下比较值(比较通道 4 的值) (TIM2 中心对齐模式下不对称计数使能有效)

©Copyright 2026, PADAUK Technology Co. Ltd

13. 看门狗 (WDG)

13.1 功能

13.1.1. WDG时钟源

看门狗 WDG 工作于片内的 OSC_32K 时钟，为一个 32 位宽的递减计数器．计数器在 WDG 启动时由重载设定寄存器载入初值，然后在时钟的驱动下每个周期进行递减计数。

13.1.2. 启动WDG

上电复位后 WDG 模块默认处于禁止状态．应用软件必须设置 WDG_CR.INTE 位启动 WDG 开始工作．同时，该位的置 1 也意味着系统将 WDG 计数器归零

13.1.3. WDG中断

WDG 计数器启动后，递减计数值第一次归零时，将触发 WDG 中断响应。一旦触发了 WDG 中断，在又经过一个WDG 定时时间后系统即被复位．应用软件可以编写自己的非屏蔽中断服务程序来应对此 WDG 中断事件．如果发生 WDG中断，一般意味着应用软件已有过长的间隔时间没有执行清除看门狗工作，可用于软件调试阶段的可靠性诊断．正式发布的应用程序不能在自己的非屏蔽中断服务程序中执行清除看门狗指令，否则看门狗将永远不会引发系统复位，失去了其系统监控的作用

13.1.4. WDG复位

在 WDG 计数器一次归零后，WDG 重新载入初值，继续递减计数．当计数值第二次归零，且 WDG_CR.RSTE 位设为 1 时，将立即引发一次系统硬件复位，即看门狗复位。

13.1.5. WDG定时时间设定

计数重载寄存器 WDG_LOAD 的设定值直接决定了 WDG 的定时时间．按应用所期望设定的看门狗定时时间，计算 WDG_LOAD 重载值的公式为：

$WDG\_LOAD = T_{WDG} * f_{CLK} - 1$

其中:

- T_{WDG} 为期望的 WDG 定时时间，单位为 ms
- f_{CLK} 为 WDG 的时钟频率，单位为 kHz，32 KHz 或系统时钟

例如，若应用软件希望设置约 20 ms 的看门狗定时时间，时钟为内部 32 KHz 时钟，则按上述公式则 f_{CLK} = 32，且精度保证，算得 WDG_LOAD 应赋值为 639。

软件清除 WDG 应用软件在正常运行时，必须在设定的 WDG 定时时间范围内对 WDG_CLR 寄存器写一次任意数，以重载复位 WDG 计数器．正常情况下不应出现 WDG 中断，更不能出现 WDG 复位

13.2 寄存器列表

地址	寄存器	描述
0x4000_4000	WDG_LOAD	WDG 计数重载寄存器
0x4000_4004	WDG_VALUE	WDG 计数值寄存器
0x4000_4008	WDG_CR	WDG 控制寄存器
0x4000_400C	WDG_INTCLR	WDG 中断清除寄存器
0x4000_4010	WDG_RIS	WDG 原始中断标志寄存器
0x4000_4014	WDG_MIS	WDG 掩蔽中断标志寄存器
0x4000_4400	WDG_LOCK	WDG 锁定控制寄存器

表 13-1 WDG 寄存器列表

13.3 寄存器描述

13.3.1 WDG计数重载寄存器 (WDG_LOAD)

Bit	Name	R/W	Reset	Description
31:0	WDG_LOAD	R/W	0xFFFFFFFF	该 32 位宽的寄存器用以设定供 WDG 启动或复位后递减计数器的重载初始值，它决定了 WDG 的溢出复位时间见隔

13.3.2 WDG计数值寄存器 (WDG_VALUE)

Bit	Name	R/W	Reset	Description
31:0	WDG_VALUE	R	0xFFFFFFFF	该 32 位宽的寄存器为 WDG 工作时的当前计数值．WDG 在启动时或被清除后，WDG_LOAD 的设定值被载入该寄存器，然后在 WDG 时钟的驱动下进行递减计数看门狗值寄存器指示现在递减计数器当前计数值。

13.3.3 WDG 控制寄存器 (WDG_CR)

Bit	Name	R/W	Reset	Description
31:3	-	R	0x0	保留
2	DBGE	R/W	0x0	DBGE： WDG 调试使能控制 0: WDG 在调试暂停时被挂起 1: WDG 在调试暂停时持续工作
1	RSTE	R/W	0x0	RSTE： WDG 复位使能控制位 0: WDG 复位禁止 1: WDG 复位使能
0	INTE	R/W	0x0	INTE： WDG 中断使能控制位 0: WDG 模块禁止，无中断响应 1: WDG 模块使能，同时 WDG 中断（NMI）被使能

13.3.4 WDG 中断清除寄存器 (WDG_INTCLR)

WDG 中断清除寄存器（地址：4000400CH）

该寄存器为只写，对其进行任意数据的写操作都会将 WDG_LOAD 的值重载入 WDG_VALE 寄存器内重新启动递减计数，同时复位 WDG 中断标志（如果已经被置 1）

应用软件在设定并启动 WDG 模块后，必须在 WDG 计数归零前对该寄存器进行一次写操作以复位 WDG

13.3.5 WDG 原始中断标志寄存器 (WDG_RIS)

该寄存器为只读，其最低位 RIF 为 WDG 计数归零中断的原始标志位，只要 WDG 计数器发生一次归零，无论 WDG_CR.INTE 设置如何，该位即被置 1．对 WDG_INTCLR 寄存器进行一次写入，即可清除 RIF．

Bit	Name	R/W	Reset	Description
31:1	-	R	0x0	保留
0	RIF	R	0x0	RIF： WDG 原始中断标志信息位 0: WDG 未发生归零 1: WDG 发生归零（但是否响应中断取决于 WDG_CR.INTE 设置）．写 WDG_INTCLR 将其清 0

13.3.6 WDG 掩蔽中断标志寄存器 (WDG_MIS)

该寄存器为只读，其最低位 MIF 为 WDG 计数归零中断原始标志位 RIF 经 INTE 掩蔽后的状态．如果 RIF 和 INTE 同时为 1，则 MIF 为 1，引发 WDG 中断；否则为 0，不产生 WDG 中断．对 WDG_INTCLR 寄存器进行一次写入后清除 RIF，亦同时清楚了 MIF

Bit	Name	R/W	Reset	Description
31:1	-	R	0x0	保留
0	MIF	R	0x0	掩蔽中断标志信息位 0: WDG 掩蔽中断标志无效（可能是实际无中断，或 WDG_CR.INTE 未使能） 1: WDG 掩蔽中断标志有效，写 WDG_INTCLR 将其清 0

13.3.7 WDG锁定控制寄存器 (WDG_LOCK)

该寄存器提供了一种 WDG 系统安全锁定的机制，可避免在系统运行过程中，因软件设计缺陷或程序跑飞时意外篡改 WDG 模块的配置．一旦 WDG 模块被锁定，本模块所有的寄存器都无法被软件改写．

对 WDG_LOCK 寄存器写入一个特殊值“0x1ACCE551”后，将对 WDG 模块解锁，解锁后软件可以自由配置相关的寄存器；对该寄存器写入任意其它值后， WDG 模块即被锁定，模块内任何可配置的寄存器都无法被改写．

任何时候读此寄存器时，仅最低位有意义，指示 WDG 的锁定状态．

Bit	Name	R/W	Reset	Description
31:1	-	R	0x0	保留
0	LOCK	R	0x0	WDG 模块锁定状态位 0: WDG 模块未锁定，模块内相关寄存器可改写 1: WDG 模块被锁定，模块内所有寄存器无法写

13.4 用户操作

两个时钟域间有同步逻辑，当看门狗加载和控制寄存器由 APB 操作更新时，WDGCLK 时钟域的新值将在两个 WDGCLK 周期内生效。当看门狗定时器在 WDGCLK 计数，同步逻辑会先锁住 WDGCLK 上计数器的值，然后再同步 PCLK，以供 CPU 能读取看门狗值寄存器。看门狗清除中断寄存器也需要 2 个 WDGCLK 周期和 2 个 PCLK 时钟同步，软件必须小心处理。

©Copyright 2026, PADAUK Technology Co. Ltd

14. ADC

14.1 概述

芯片内有一个 12 位的模数转换器 ADC，该 ADC 有 16 个通道，允许 ADC 测量 12 个外部输入引脚电压，及其它内部电压通道。

要求: 每次采样间隔需要为大于 12 倍 clk 主频时间。

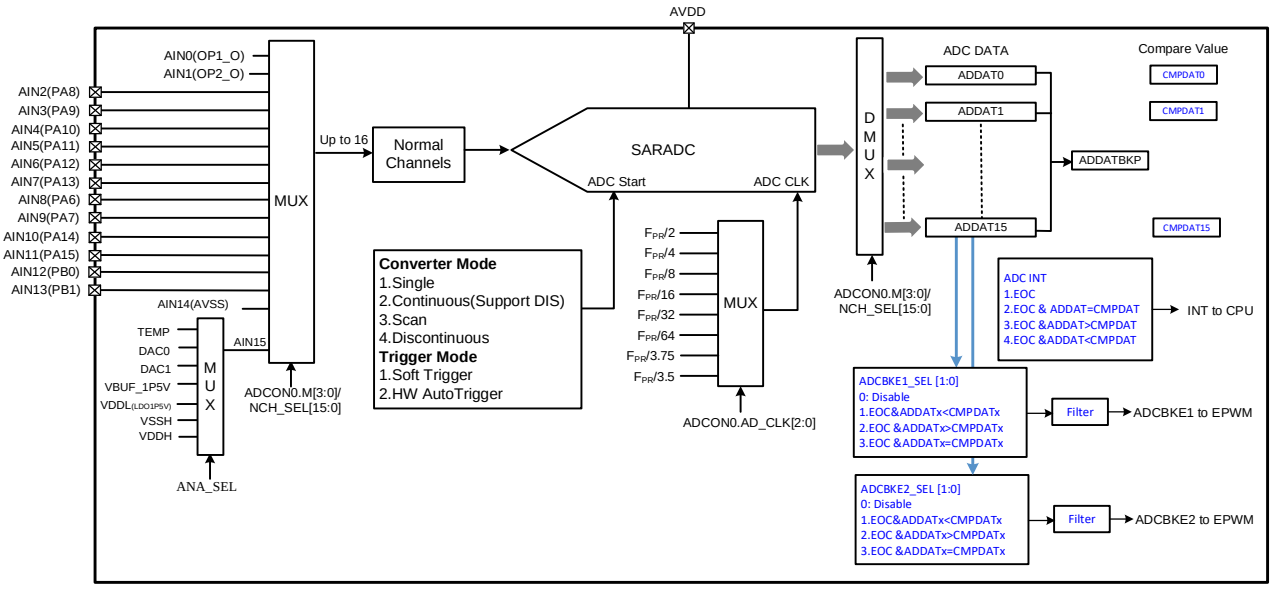


图 14-1 ADC Block Diagram

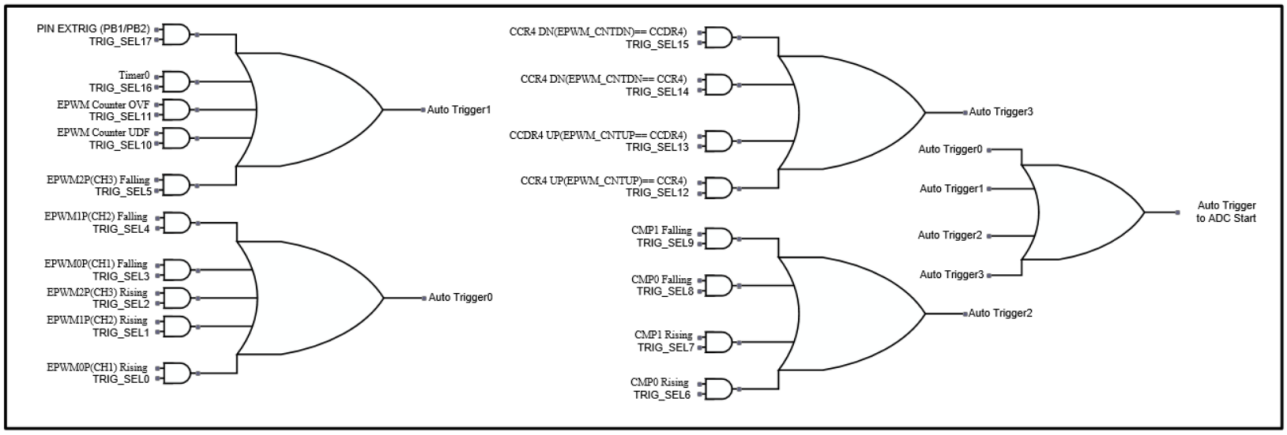


图 14-2 ADC Auto Trigger Block Diagram

14.2 特点

- 12 位的模数转换分辨率
- -40~105 度
- 绝对精度：12 bit 配置
- 最大 16 Mhz 主频输入 ,1 Msps 采样率。(最小 937.5 KHz 主频输入)
- 最多 12 路单端片外模拟信号输入
- 提供**单次转换模式 (Single Mode)，连续转换模式 (Continuous Mode)，扫描模式 (Scan Mode)，间断模式 (Discontinuous Mode)**
- **自动触发模式** (Timer0, EPWM, CMP,外部號入 PIN)
- 片内通道包括:
    - 电源电压（VDD）
    - 地（GND）
    - TEMP, DAC0/1, LDO 1.5V
- 输入电压范围: 0~Vdd

©Copyright 2026, PADAUK Technology Co. Ltd

14.3 转换时序

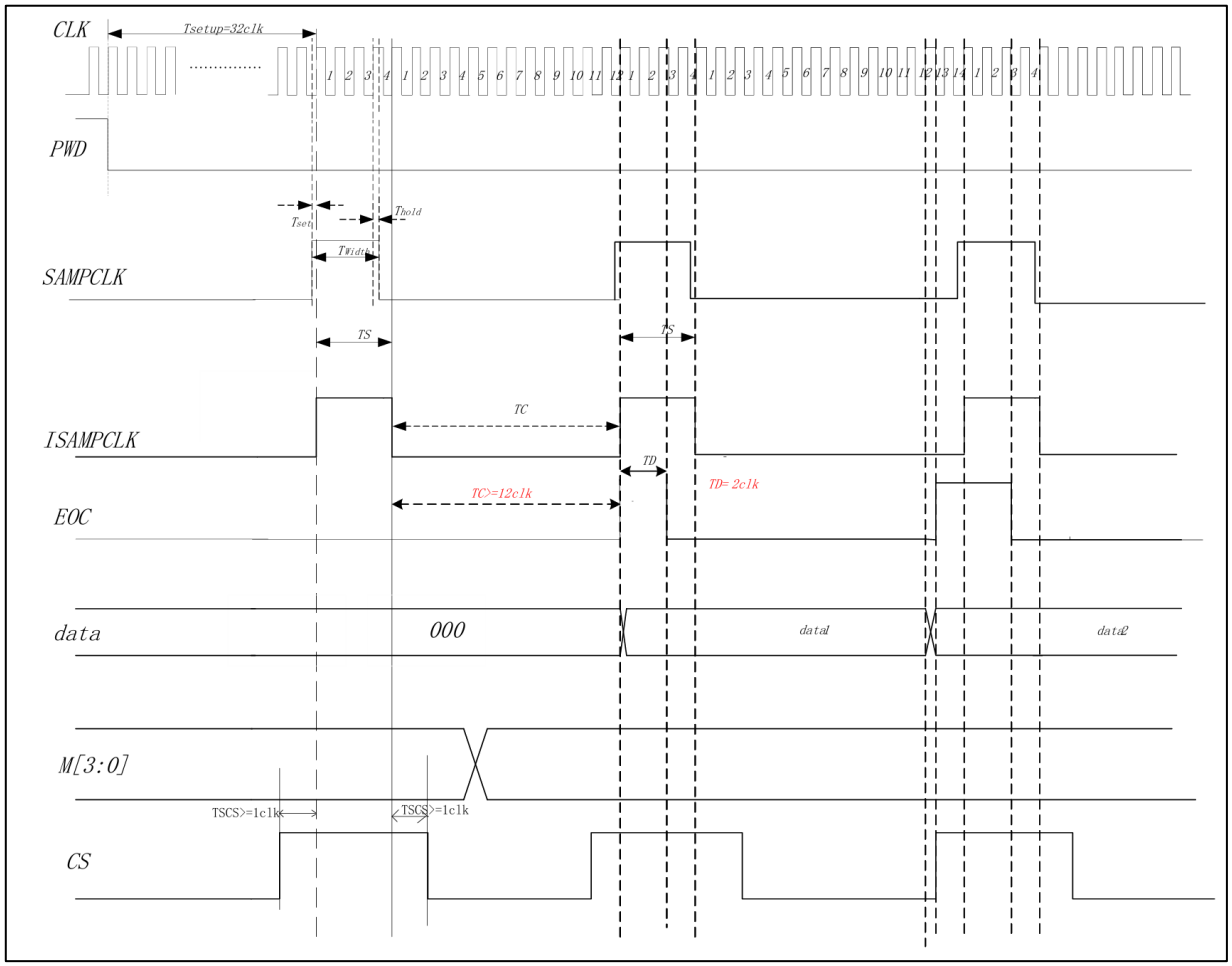


图 14-3 ADC Sample Timing

注意: 在要求 12 bit 转换且要求 1 MSPS 的采样率下，需要使用不等待 eoc 的模式，否则效率无法达到要求。

14.4 功能描述

转换模式设定如下表:

ADC_CON0. CONTINUE	ADC_CON0. ENCONT	ADC_CHSEL. DISCEN	功能说明
0	X	X	**单次转换模式** (ADC_CON0.M[3:0] 为通道选择)
1	1	X	**连续转换模式** (ADC_CHSEL[15:0] 为通道选择) (触发一次**连续**转换无限次)
1	0	0	**扫描模式** (ADC_CHSEL[15:0] 为通道选择) (可设定转换次数)
1	0	1	**间断模式** (ADC_CHSEL[15:0] 为通道选择) (触发多次转换设定的通道.所有设定通道转换完成时,重回转换)

表 14-1 转换模式表

单次转换模式 **(Single Mode)**

在单次转换模式下，ADC 启动后只执行一次转换，可对所有的 16 路 ADC 通道进行转换。该模式既可通过设置 ADC_CON0.START 位启动也可通过设置 ADC_TRIG_SEL.TRIG_SELx[2:0] 或 ADC_CHSEL TRIG_SELx[2:0] 的外部触发启动。一旦 ADC_CON0.M [3:0] 选定通道,触发 ADC 将转换完成, ADC_CON0.START 位自动清零，转换结果保存在 ADC_DAT0~15 寄存器中。

- 例: 如果 M [3:0] 通道选定，设置 ADC_CON0.START 或 外部触发启动(TRIG_SELx[2:0])

- 转换数据被储存在 16 位 ADC_DATx 寄存器中
- DONE(转换结束)标志被设置
- 如果设置了 INT_EN = 0x1，则产生中断。

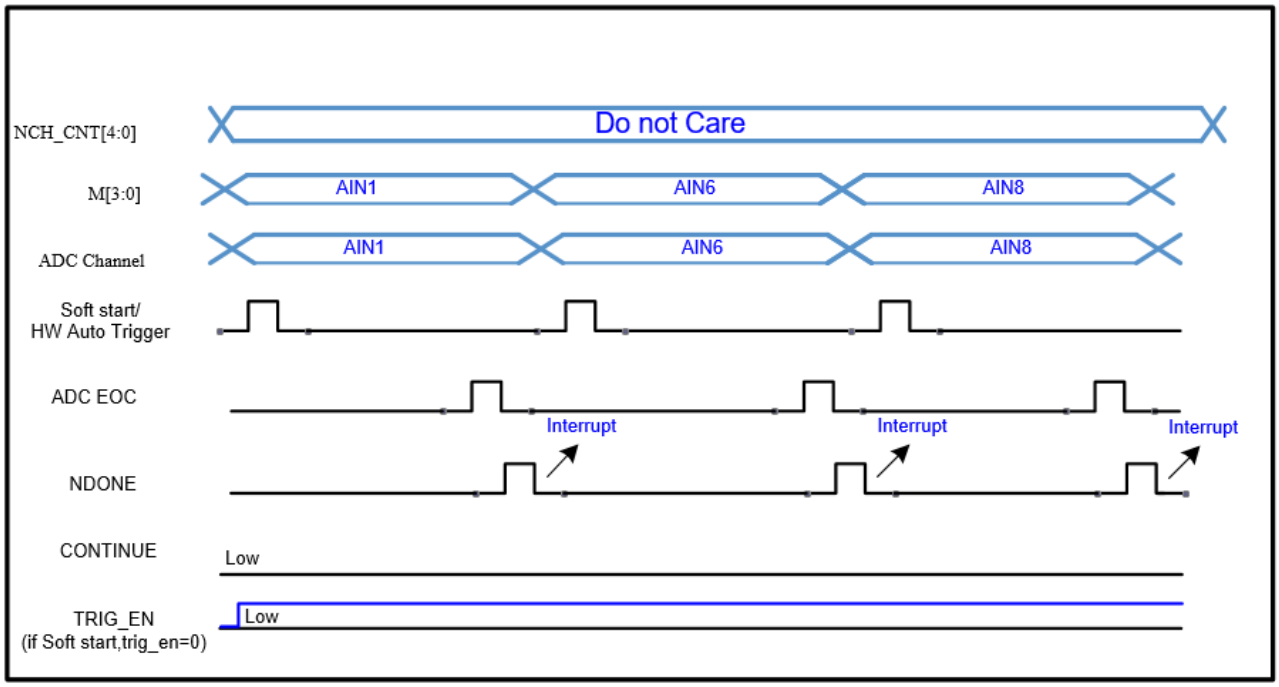


图 14-4 单次转换模式 Timing

14.4.1 连续转换模式(Continuous Mode)

在连续转换模式中，当前面 ADC 转换一结束马上就启动另一次转换。此模式可通过外部触发启动或通过设置 ADC_CON0.START 位启动，此时 ADC_CON0.CONTINUE, ADC_CON0.ENCONT 位须为 1。

- 连续转换无限次数，因此 DONE(转换结束)标志会维持为 0 (此模式不支持中断)

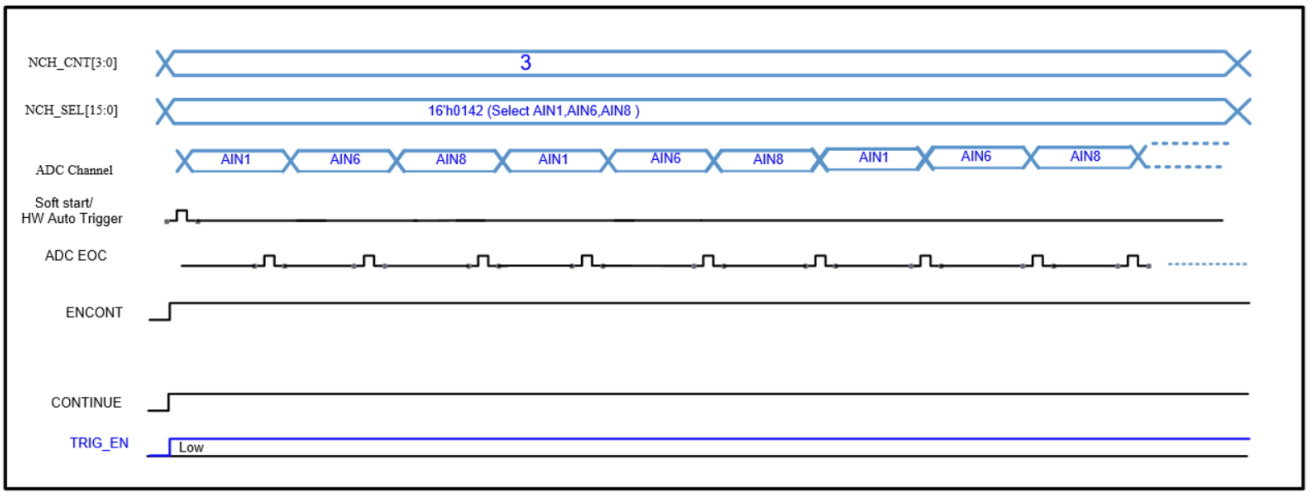


图 14-5 连续转换无限制次数 Timing

14.4.2 扫描模式 (Scan Mode)

如果一个规则通道被转换：(基于 ADC_CHSEL.CH_SEL 通道选择 , ADC_CON0.CONTINUE = 1, ADC_CON0.ENCONT=0, ADC_CHSEL.DISCEN = 0)

- CH_SEL [15:0] = 0x0142 (AIN1,6,8 低通道转换优先权高) ADC_CHSEL.CH_CNT = 0x2(转换 3 次)
- 第一次触发: 转换 AIN1,AIN6,AIN8 转换数据被储存在 16 位 ADDAT1,6,8 寄存器中
- DONE(转换结束)标志被设置
- 如果设置了 INT_EN = 0x1，则产生中断。

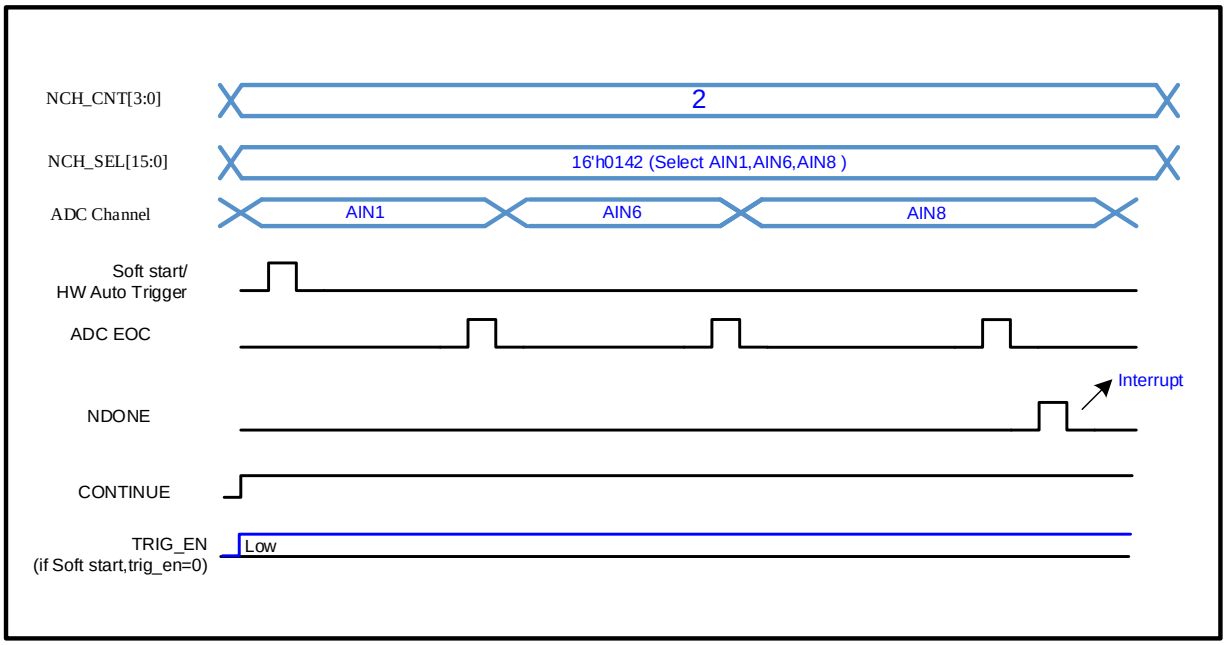


图 14-6 扫描模式通道转换 Timing

- 当转换次数 CH_CNT > CH_SEL 选择的通道数量时，会重复转换 CH_SEL 选择的通道直到转换完 CH_CNT 次
- CH_SEL[15:0] = 0x0101 (AIN0,8 低通道转换优先权高) ADC_CHSEL.CH_CNT = 3(转换 4 次)
- 第一次触发: 重複转换 AIN0, AIN8, AIN0, AIN8 转换数据被储存在 16 位 ADC_DAT0, 8 寄存器中
- DONE(转换结束)标志被设置
- 如果设置了 INT_EN = 0x1，则产生中断。

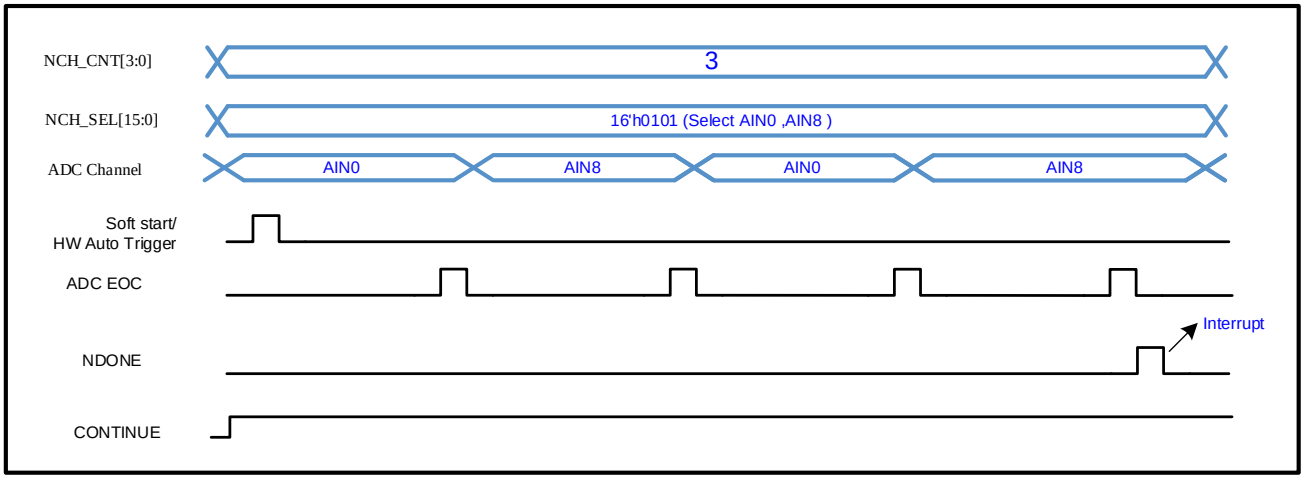


图 14-7 扫描模式通道转换 Timing 2

14.4.3 间断模式 (Discontinuous Mode)

(1) 当 ADC_CHSEL.DISCEN = 1 , ADC_CON0.CONTINUE = 1,ADC_CHSEL.DISCNUM[2:0] = 3,
ADC_CHSEL.CH_CNT [3:0] = 11,

CH_SEL[15:0] = 0xDEB3 (AIN0,1,4,5,7,9,10,11,12,14,15 低通道转换优先权高)

连续转换分成 4 段转换完成 (CH_CNT[3:0]/ DISCNUM[2:0] = 3 余 2 (所有通道转换完成需要的触发次数:3+1=4)

- ADC_CHSEL.DISC_INTSEL = 0 (所有通道转换完成发生中断) ADC_CON0.ENCONT=0

- 第一次触发: 转换 AIN0,AIN1,AIN4, (转换数据被储存在 16 位 ADC_DAT0,1,4 寄存器中)
- 第二次触发: 转换 AIN5,AIN7,AIN9 (转换数据被储存在 16 位 ADC_DAT5,7,9 寄存器中)
- 第三次触发: 转换 AIN10,AIN11,AIN12 (转换数据被储存在 16 位 ADC_DAT10,11,12 寄存器中)
- 第四次触发: 转换 AIN14,AIN15 (转换数据被储存在 16 位 ADC_DAT14,15 寄存器中)
- 产生 DONE 事件(转换结束)标志被设置

  → 如果设置了 INT_EN = 0x1，**则产生中断**
- 第五次触发: 重回转换 AIN0,AIN1,AIN4

- DISC_INTSEL = 1 (每次触发转换完成都发生中断)

- 第一次触发: 转换 AIN0,AIN1,AIN4, (转换数据被储存在 16 位 ADC_DAT0,1,4 寄存器中)

  → 如果设置了 INT_EN = 0x1 **则产生中断**
- 第二次触发: 转换 AIN5,AIN7,AIN9 (转换数据被储存在 16 位 ADC_DAT5,7,9 寄存器中)

  → 如果设置了 INT_EN = 0x1 **则产生中断**
- 第三次触发: 转换 AIN10,AIN11,AIN12 (转换数据被储存在 16 位 ADC_DAT10,11,12 寄存器中)

  → 如果设置了 INT_EN = 0x1 **则产生中断**
- 第四次触发: 转换 AIN14,AIN15 (转换数据被储存在 16 位 ADC_DAT14,15 寄存器中)
- 产生 DONE 事件(转换结束)标志被设置

  → 如果设置了 INT_EN = 0x1 **则产生中断**

- 第五次触发: 重回转换 AIN0,AIN1,AIN4

  → 如果设置了 INT_EN = 0x1 **则产生中断**

(2) DISCEN = 1 , CONTINUE = 1, DISCNUM[2:0] = 1, CH_CN[3:0] = 2

CH_SEL[15:0] = 0x0001 (Only Select AIN0)

连续转换分成 2 段转换完成 (CH_CN[3:0]/ DISCNUM[2:0] = 2)

- DISC_INTSEL = 0 (所有通道转换完成发生中断)
  - 第一次触发: 转换 AIN0 (第一次触发转换数据被储存在 16 位 ADC_DAT0 寄存器中)
  - 第二次触发: 转换 AIN0 (第二次触发转换数据被储存在 16 位 ADC_DAT0 寄存器中,

第一次触发值 ADC_DAT0 会备份储存在 ADC_BAKDAT 寄存器中)

产生 DONE 事件(转换结束)标志被设置

→ 如果设置了 INT_EN = 0x1，**则产生中断**

- DISC_INTSEL=1 (每次触发转换完成都发生中断)
  - 第一次触发: 转换 AIN0 (第一次触发转换数据被储存在 16 位 ADC_DAT0 寄存器中)

    → 如果设置了 INT_EN = 0x1 **则产生中断**
  - 第二次触发: 转换 AIN0 (第二次触发转换数据被储存在 16 位 ADC_DAT0 寄存器中,

    第一次触发值 ADC_DAT0 会备份储存在 ADC_BAKDAT 寄存器中)

    → 如果设置了 INT_EN = 0x1，**则产生中断**

    产生 DONE 事件(转换结束)标志被设置

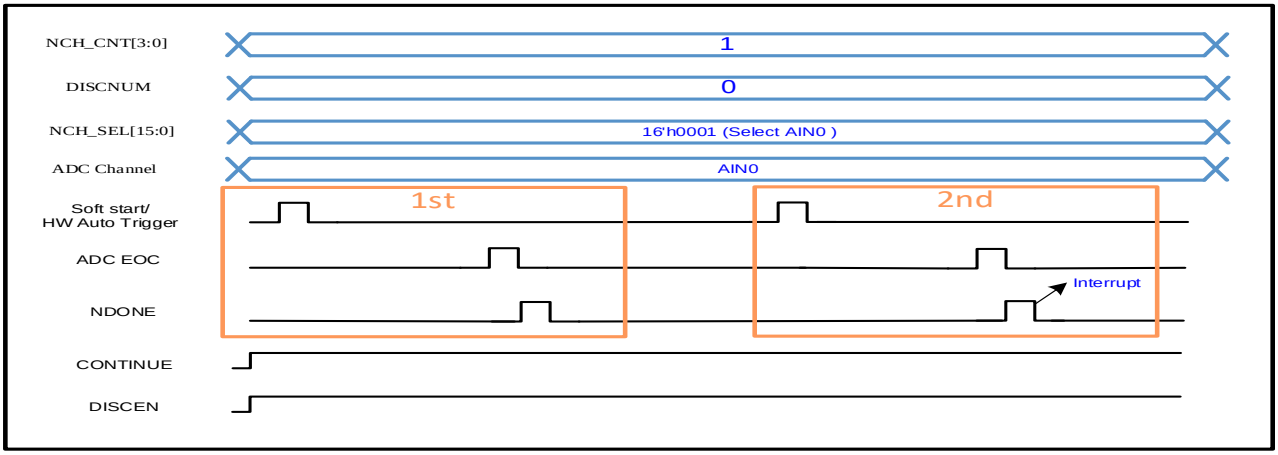


图 14-8 间断模式转换相同通道 Timing(INT_EN= 0x1, DISC_INTSEL=0)

14.4.4 比較與中斷

ADC 的中断由 ADC_CON0.INT_EN 使能,当 INT_EN = 0x1，在 ADC 转换完成, ADC_STAT.DONE 为 1 时, ADC 中断也会同时使能。

连续转换模式/扫描模式/间断模式的转换中，最后一组转换完成才会产生 DONE，因此只在最后一组转换完成后产生 ADC 中断。

而比较的结果所产生的中断(INT_EN=0x2/0x4/0x8)，发生的时间与 INT_EN = 0x1 时相同，当 DONE 为 1 时，如 ADC_STAT.comp_result 比较的结果与 INT_EN 设定的比较条件相同，则产生 ADC 中断，ADC 中断与 DONE 发生时，需要透过两个方法来清除 DONE 以避免继续产生中断。

1. 读取数据结果暂存器(ADC_DATx), 不包含 ADC_BAKDAT

2. 写入 ADC_STAT.INTCLR 为 1

如果没有清除 DONE，在下一次 START 触发时也会自动清除 DONE

(1) 在单次转换模式中(ADC_CON0.CONTINUE = 0)，如果有设定比较结果中断(INT_EN = 0x2/0x4/0x8)，每次转换完成后会根据比较的结果产生中断

当阈值(ADC_DATx.COMP_TH)均设定为 0x555, 分别转换 AIN1/6/8, 在 INT_EN 不同设定下，触发三次转换产生中断的结果以及 Timing 如下表与下图, 橘框分别代表三次触发转换的 Timing 与 Data:

单次转换模式	1st DONE	2nd DONE	3rd DONE
比较结果(Comp_result)	01 (TH < DATA)	11 (TH = DATA)	10 (TH > DATA)
INT_EN = 0x1 (所有 DONE 产生中断)	产生中断	产生中断	产生中断
INT_EN = 0x2 (当 COMP_TH < ADC_DATA DONE 产生中断)	产生中断	不产生中断	不产生中断
INT_EN = 0x4 (当 COMP_TH > ADC_DATA DONE 产生中断)	不产生中断	不产生中断	产生中断
INT_EN = 0x8 (当 COMP_TH = ADC_DATA DONE 产生中断)	不产生中断	产生中断	不产生中断

表 14-2 单次转换模式中断产生表

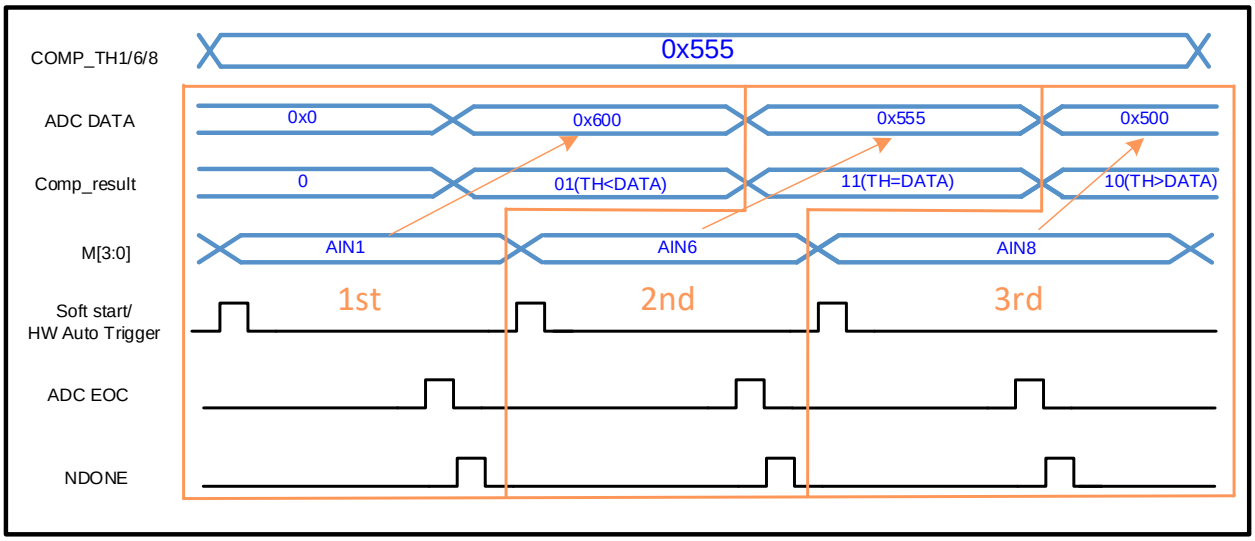


图 14-9 單次转换模式中断产生 Timing

(2) 在连续转换模式/扫描模式/间断模式中，由于最后一组通道转换完成才会使能 DONE，因此过程中转换的通道无法产生中断

在间断模式, CH_CNT = 3, DISCNUM = 1, INT_EN = 0x1，产生两次 START 分别会转换两个通道，如果设定 DISC_INTSEL = 0 的话，只会有最后一次转换完成后会产生中断，但如果设定比较结果中断(INT_EN = 0x2/0x4/0x8)，无论 DISC_INTSEL 设定为何，每一次转换完成跟会根据比较结果产生中断。

当阈值(COMP_TH)均设定为 0x555，分别转换 AIN0/8，在 INT_EN 不同设定下，触发两次转换产生中断的结果以及 Timing 如下表与下图，须注意此时 AIN0 的转换不会产生 DONE，因此也不会产生中断:

间断模式	DISC_INTSEL	1st DONE	2nd DONE
比较结果(Comp_result)		10 (TH > DATA)	01 (TH < DATA)
INT_EN = 0x1	0(CH_CNT 转换完成产生中断)	不产生中断	产生中断
	1 (所有 DONE 产生中断)	产生中断	产生中断
INT_EN = 0x2 (当 COMP_TH < ADC_DATA DONE 产生中断)	X	不产生中断	产生中断
INT_EN = 0x4 (当 COMP_TH > ADC_DATA DONE 产生中断)	X	产生中断	不产生中断
INT_EN = 0x8 (当 COMP_TH = ADC_DATA DONE 产生中断)	X	不产生中断	不产生中断

表 14-3 间断模式中断产生表

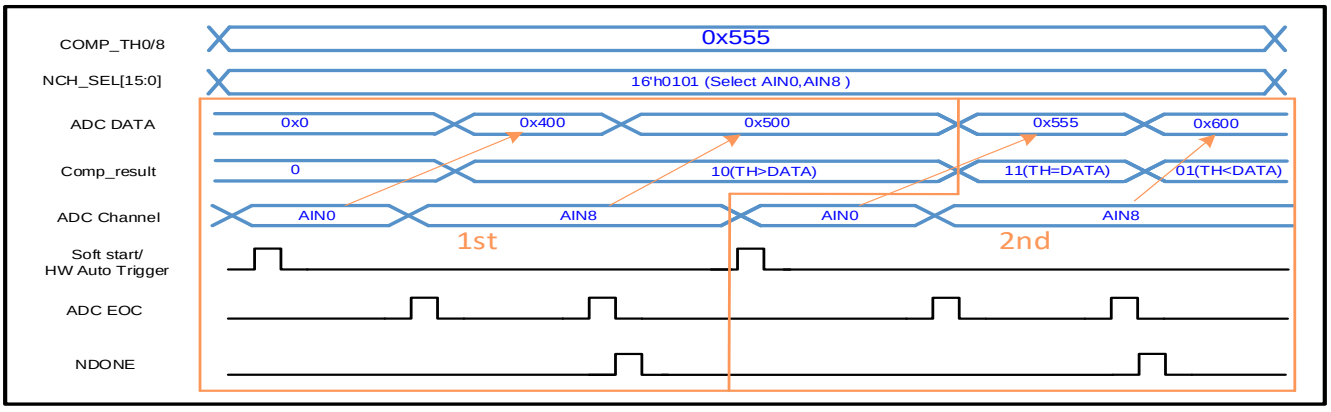


图 14-10 间断模式中断产生 Timing

14.5 用户操作

从以上时序图看出，无论是单次转换或是连续转换，每次使能 ADC IP 后（ADC_CON0 寄存器的 PWD 置 0），都需要软件延时 32 adc clks，以等待 ADC 内部参考源稳定。然后再启动 ADC 转换。

ADC 有四种转换模式:

14.5.1 ADC单次转换模式

1）软件写 ADC_CON0.CONTINUE 为 0 选择单次转换模式。

2）软件写 ADC_CON0 寄存器内的控制位（M[3:0]通道选择、CLK）寄存器配置 ADC。

3）软件写 ADC_CON0 寄存器的 START 位为 1，启动一次转换，转换完成后 ADC 产生 ADC 转换完成标志（ADC_STAT 寄存器的 DONE 置 1）。软件读 ADC_DATx 寄存器后， ADC_STAT 寄存器的 DONE 会被自动清 0。

14.5.2 连续转换模式

1） 软件写 ADC_CON0 寄存器内的控制位（CLK）寄存器配置 ADC。

2） 通道选择为 ADC_CHSEL 寄存器内的控制位的 CH_SEL[15:0]

3）软件写 ADC_CON0 寄存器内 CONTINUE 为 1，ENCONT 为 1 选择连续转换模式。

4）软件选择连续转换模式后，可选择使用 ADC_CON0 寄存器的 START 位软件触发或硬件触发 ADC 开始连续转换通道。

5）ADC 會不間斷轉換 CH_SEL 的通道，直到關閉 ENCONT 或是 CONTINUE 才會停止轉換並置 DONE 為 1。

注: 1.由于连续转换模式永远不会停止转换，因此 DONE 会保持为 0，且不支持产生中断信号

注: 2 ADC 采样率 1 MSPS 说明

在 12 bit 采样率 要达到 1 MSPS 采样。必须使用不等待 eoc 反馈的模式(ADC_STAT.EOC_CHECK_DIS = 0x1)。否则是达不到 1 MSPS

在使用不等待 eoc 的反馈模式，需要打开连续模式采样，且打开不等待 eoc 反馈模式。在这种情况下，启动一次 start 即可。

14.5.3 扫描模式

1） 软件写 ADC_CON0.CLK 寄存器配置 ADC。

2） 通道选择为 ADC_CHSEL 寄存器内的控制位的 CH_SEL[15:0]，轉換次數為 CH_CNT。

3）软件写 ADC_CON0 寄存器内 CONTINUE 为 1，ENCONT 为 0，DISCEN 为 0 開啟扫描模式。

4）软件选择扫描模式后，可选择使用 ADC_CON0 寄存器的 START 位软件触发或硬件触发 ADC 开始连续转换通道

5）在转换完 CH_CNT 次后会停止转换并置 DONE 为 1。

注: 1. 當 CH_CNT > CH_SEL 選擇的通道數量時，會重複轉換 CH_SEL 選擇的通道，直到轉換 CH_CNT 次為止，可參考 14.4.2

注: 2. 當 CH_CNT < CH_SEL 選擇的通道數量時，只會轉換 CH_CNT 次，不會將 CH_SEL 選擇的通道轉換完，下一次觸發也只會從優先級最高的通道開始轉換，不會接續上一次的通道

14.5.4 间断模式

1） 软件写 ADC_CON0.CLK 寄存器配置 ADC。

2） 通道选择为 ADC_CHSEL 寄存器内的控制位的 CH_SEL[15:0]，总转换次数为 CH_CNT，每次触发转换次数为 DISCNUM。

3）软件写 ADC_CON0 寄存器内 CONTINUE 为 1，ENCONT 为 0，DISCEN 为 1 開啟间断模式。

4）软件选择间断模式后，可选择使用 ADC_CON0 寄存器的 START 位软件触发或硬件触发 ADC 开始连续转换通道

5）每次触发都会转换 DISCNUM 次并置 DONE 为 1，在转换完 CH_CNT 次后会停止转换并置 DONE 为 1。

14.6 寄存器列表

地址	寄存器	描述
0x4000_4800	ADC_CON0	ADC 转换控制寄存器
0x4000_4804	ADC_STAT	ADC 转换状态寄存器
0x4000_4808	ADC_DAT0	ADC 转换缓存寄存器 0
0x4000_480C	ADC_DAT1	ADC 转换缓存寄存器 1
0x4000_4810	ADC_DAT2	ADC 转换缓存寄存器 2
0x4000_4814	ADC_DAT3	ADC 转换缓存寄存器 3
0x4000_4818	ADC_DAT4	ADC 转换缓存寄存器 4
0x4000_481C	ADC_DAT5	ADC 转换缓存寄存器 5
0x4000_4820	ADC_DAT6	ADC 转换缓存寄存器 6
0x4000_4824	ADC_DAT7	ADC 转换缓存寄存器 7
0x4000_4828	ADC_DAT8	ADC 转换缓存寄存器 8
0x4000_482C	ADC_DAT9	ADC 转换缓存寄存器 9
0x4000_4830	ADC_DAT10	ADC 转换缓存寄存器 10
0x4000_4834	ADC_DAT11	ADC 转换缓存寄存器 11
0x4000_4838	ADC_DAT12	ADC 转换缓存寄存器 12
0x4000_483C	ADC_DAT13	ADC 转换缓存寄存器 13
0x4000_4840	ADC_DAT14	ADC 转换缓存寄存器 14
0x4000_4844	ADC_DAT15	ADC 转换缓存寄存器 15
0x4000_4848	ADC_CHSEL	ADC 转换通道选择
0x4000_484C	ADC_TRGSEL	ADC 触发源选择
0x4000_4850	ADC_BKSEL	ADC 刹车源配置
0x4000_4854	ADC_BAKDAT	ADC 转换通道备分缓存寄存器

表 14-4 ADC 寄存器列表

©Copyright 2026, PADAUK Technology Co. Ltd

14.7 寄存器描述

14.7.1 ADC 转换控制寄存器 (ADC_CON0)

Bit	Name	R/W	Reset	Description
31	RST	R/W	0x1	ADC 模拟模块复位信号(PWD 須為 0) 1: Reset ADC 0: Normal Mode.
30	PWD	R/W	0x1	ADC 模拟模块 power down 信号, 1: power down 0: 后须 32 个 adc 时钟后, ADC 模块才会开始运作
29	BAKEN	R/W	0x1	ADC 通道转换完成时,会备份缓存目前通道上次 ADC 转换的值. 0:不使能. 1:始能备份
28:23	-	R	0x0	保留
22:20	TZO	R/W	0x4	zero offset change 零電壓偏移 Trim 值,將會校正 ADC 輸出的 data 000: -4 LSB 001: -3 LSB 010: -2 LSB 011: -1 LSB 100: default(+0LSB) 101: +1 LSB 110: +2 LSB 111: +3 LSB

19:16	INT_EN	R/W	0x0	INT_EN: ADC 模块中断输出使能(参考 Table 2/3) 0000: 关闭中断使能 0001: ADC 通道转换完成中断使能(DONE) 0010: ADC 通道转换完成時，阈值比转换结果小中断使能 0100: ADC 通道转换完成時，阈值比转换结果大中断使能 1000: ADC 通道转换完成時，阈值和转换结果相等中断使能
15	ENCONT	R/W	0x0	连续转换模式(CONTINUE 須為 1)(参考 Table 1.) 0: 基于 ADC_CHSEL.CH_CNT[3:0] 设定的次数转换 1: 无限次连续转换模式
14	-	R	0x0	保留
13	ALIGN	R/W	0x0	ALIGN: 讀取 ADC_DATx.DATA 对齐方式选择 0: ADC_DATx.DATA 讀取结果右对齐 1: ADC_DATx.DATA 讀取结果左对齐 注: 如 data 為 0x555, 左對齊讀 data_reg 為 0x5550, 右對齊為 0x0555
12	CONTINUE	R/W	0x0	ADC 转换连续模式使能(参考 Table 14-1.) 0: 单次转换模式, M[3:0] 指定的通道 1: 连续转换模式/间断模式/扫描模式 ENCONT , DISCEN 選擇模式 根據 ADC_CHSEL.CHSEL 選擇通道進行轉換 注: Write (0➔1 or 1➔0): 等待转换完成才会更新
11	TRIG_EN	R/W	0x0	Trigger 信号触发 ADC 转换模式使能 0: 不使能硬件触发 START 采样(软件触发) 1: 使能硬件触发 START 采样(软/硬件触发)

10:8	CLK	R/W	0x1	ADC 转换器工作时钟选择(937.5KHz≦ADC_CLK≦16MHz) 000: PCLK/2 001: PCLK/4 010: PCLK/8 011: PCLK/16 100: PCLK/32 101: PCLK/64 110: PCLK/3.75 (16M@PCLK = 60Mhz) 111: PCLK/3.5 (17M@PCLK = 60Mhz) 默认 4 分频。时钟切换需要 4 个 ADC 时钟周期才能切换完成。因此切换时钟后需要等待 4 个 ADC 时钟周期才能下发 START 请求
7	START	R/W	0x0	AD 规则转换通道转换使能 0: 关闭 ADC 转换 1: 启动 ADC 转换, AutoTrigger(ADC_TRGSEL)觸發也會使能 START, 當轉換尚未結束時(ADC_STATE.DONE 未發生), 所有使能無效 注: 要求(PWD 置 0 后 32 个 adc 时钟后 START 才能有效)
6	-	R	0x0	保留
5	EN	R/W	0x0	ADC Controller 工作使能(不包含寄存器) 0: ADC Controller 不使能 1: ADC Controller 使能
4	-	R	0x0	保留

3:0	M	R/W	0x0	单次转换模式时通道选择: Input channel selection When (CONTINUE =0): 0000: AIN0 0001: AIN1 0010: AIN2 0011: AIN3 0100: AIN4 0101: AIN5 0110: AIN6 0111: AIN7 1000: AIN8 1001: AIN9 1010: AIN10 1011: AIN11 1100: AIN12 1101: AIN13 1110: AIN14 1111: AIN15 [1*] When (CONTINUE =1 转换中尚未发生 Done 时): Read 为目前转换中的通道(依照 ADC_CHSEL.CH_SEL) Write 不变，为单次转换模式通道选择，不影响 CH_SEL 注: [1*] 须配合 AMISC_VBUF_CR.ANA_SEL, AMISC_ADC_AIN_CR.ANA2ADC_EN 使用，须选择 SYSCLK 为 30 Mhz，配合 CLK 設為 101 与 ADC_STAT.TS_SET 设定 18 才能使取樣時間達到 40 μs)

14.7.2 ADC 转换状态寄存器 (ADC_STAT)

支持将 pga0,1, 通道的 ADC 采样数据配置输出到不同的输出寄存器中。

Bit	Name	R/W	Reset	Description
31:28	START_CNT	R	0x0	发送到的 START 统计记录。
27:24	EOC_CNT	R	0x0	接收到的 eoc 统计记录。
23:21	DLY_SET	R/W	0x0	ISAMPLCLK 低电平保持时间, 单位为 pclk(见 Figuer 3.中 TC) 0: dly 12 adc clk (用于 1 MSPS 采样率场景下 16M adc clk) 1: dly 14 adc clk 2: dly 18 adc clk 3: dly 24 adc clk 4: dly 26 adc clk
20:16	TS_SET	R/W	0x3	AD C 采样时间, 即 ISAMPLCLK 高电平保持时间, 单位为 pclk(见 Figuer 3.中 TS) TS = TS_SET +1 clk, 当 TS_SET <= 3, TS 为 4 clk, TS 最多为 32clk 注: ADC 采样间隔为 TS + TC, 預設為 12clk + 4clk = 16clk, 當 adc clk 為 16 Mhz 時, ADC 取樣頻率為 1 Msps
15:7	-	R	0x0	保留
6	EOC_CHECK_DIS	R/W	0x1	0: 连续模式下检查 eoc 反馈后下发下一个请求。 1: 连续模式下不检查 eoc 反馈后下发下一个情求。 注: 要求，连续模式必须有效，且连续模式，EOC_CHECK_DIS 有效后，需要使能一次 adc 采样，作为启动。

				关闭需要取消 EOC_CHECK_DIS 和连续模式。
5:4	COMP_RESULT	R	0x0	COMP_RESULT：比较结果寄存器 00: 当软件读了数据结果寄存器（ADC_DATx）后清零 01: 阈值比 ADC 采样结果小 10: 阈值比 ADC 采样结果大 11: 阈值和 ADC 采样结果相等
3:2	-	R	0x0	保留
1	INT_CLR	R/W	0x0	写 1 清除 ADC 中断旗标 (DONE). 1: 清除中断旗标(DONE), 清除后 INT_CLR 置 0 0: 无动作
0	DONE	R	0x0	DONE: A/D 规则通道转换完成旗标 1: 转换完成有新数据置 1, INT_EN 使能時觸發中斷 0: 当软件读了数据结果寄存器（任意一个 ADC_DATx），或寫 INT_CLR 為 1, 或使能下一次轉換後后清零

14.7.3 ADC 转换通道选择 (ADC_CHSEL)

Bit	Name	R/W	Reset	Description
31:30	EXTRIG_SEL	R/W	0x0	PIN 触发选择 00: PB1 Rising 01: PB1 Falling 10: PB2 Rising 11: PB2 Falling
29:28	TRIG_SEL17	R/W	0x0	始能外部 PIN 触发 11: 保留. 10: 保留 01: 始能始能外部 PIN EXTRIG (PB1/PB2)触发发转换采样 00: 不使能
27:26	TRIG_SEL16	R/W	0x0	TRIG_SEL16 11: 保留. 10: 保留 01: 始能 TIM0 溢出信号触发转换采样 00: 不使能 注: TIM0 计时触发可以选择软件与 EPWM P0,1,2,3 rising/falling
25	DISCEN	R/W	0x0	DISCEN: 使能间断模式(参考 Table 14-1.) 该位由软件设置和清除，用于开启或关闭间断模式 0: 禁用间断模式 1: 使能间断模式 (ADC_CON0.COUNTINUE 须為 1)

24:22	DISCNUM	R/W	0x0	DISCNUM[2:0]: 间断模式转换次数计数 (DISCEN, ADC_CON0.COUNTINUE 须為 1) 软件通过这些位定义在间断模式下，每次 START 触发后转换规则通道的次數 000: 1 次 001: 2 次 …… 111: 8 次
21	DISC_INTSEL	R/W	0x0	选择间断模式的中断规则.(DISCEN = 1 and ADC_CON0.INT_EN[0] = 1 时有效) 0: 全部通道转换完成产生中断 1: 一个通道转换完成产生中断.
20	-	R	0x0	保留
19:16	CH_CNT	R/W	0x0	扫描模式/间断模式转换通道次数. 0000: 转换 1 次通道. (基于 ADC_CHSEL [15:0]) 0001: 转换 2 次通道. (基于 ADC_CHSEL [15:0]) 0010: 转换 3 次通道. (基于 ADC_CHSEL [15:0]) . . . 1110: 连续转换 15 次通道. (基于 ADC_CHSEL [15:0]) 1111: 连续转换 16 次通道. (基于 ADC_CHSEL [15:0]) 注: 转换低通道优先级高: AIN0 (ADC_CHSEL[0] = 1:转换, 0:无需转换)

				➔AIN1 (ADC_CHSEL[1] = 1:转换, 0:无需转换) ➔AIN2 (ADC_CHSEL[2] = 1:转换, 0:无需转换) . . . ➔AIN13 (ADC_CHSEL[13] = 1:转换, 0:无需转换) ➔AIN14 (ADC_CHSEL[14] = 1:转换, 0:无需转换) ➔AIN15 (ADC_CHSEL[15] = 1:转换, 0:无需转换)
15	CH_SEL[15]	R/W	0x0	CH_SEL[15] 1: 选择 AIN15 转换. 0: 无选择
14	CH_SEL[14]	R/W	0x0	CH_SEL[14] 1: 选择 AIN14 转换. 0: 无选择
13	CH_SEL[13]	R/W	0x0	CH_SEL[13] 1: 选择 AIN13 转换. 0: 无选择
12	CH_SEL[12]	R/W	0x0	CH_SEL[12] 1: 选择 AIN12 转换. 0: 无选择
11	CH_SEL[11]	R/W	0x0	CH_SEL[11] 1: 选择 AIN11 转换. 0: 无选择

10	CH_SEL[10]	R/W	0x0	CH_SEL[10] 1: 选择 AIN10 转换. 0: 无选择
9	CH_SEL[9]	R/W	0x0	CH_SEL[9] 1: 选择 AIN9 转换. 0: 无选择
8	CH_SEL[8]	R/W	0x0	CH_SEL[8] 1: 选择 AIN8 转换. 0: 无选择
7	CH_SEL[7]	R/W	0x0	CH_SEL[7] 1: 选择 AIN7 转换. 0: 无选择
6	CH_SEL[6]	R/W	0x0	CH_SEL[6] 1: 选择 AIN6 转换. 0: 无选择
5	CH_SEL[5]	R/W	0x0	CH_SEL[5] 1: 选择 AIN5 转换. 0: 无选择
4	CH_SEL[4]	R/W	0x0	CH_SEL[4] 1: 选择 AIN4 转换. 0: 无选择
3	CH_SEL[3]	R/W	0x0	CH_SEL[3] 1: 选择 AIN3 转换. 0: 无选择

©Copyright 2026, PADAUK Technology Co. Ltd

2	CH_SEL[2]	R/W	0x0	CH_SEL[2] 1: 选择 AIN2 转换. 0: 无选择
1	CH_SEL[1]	R/W	0x0	CH_SEL[1] 1: 选择 AIN1 转换. 0: 无选择
0	CH_SEL[0]	R/W	0x0	CH_SEL[0] 1: 选择 AIN0 转换. 0: 无选择

©Copyright 2026, PADAUK Technology Co. Ltd

14.7.4 ADC 触发源选择 (ADC_TRGSEL)

Bit	Name	R/W	Reset	Description
31:30	TRIG_SEL15	R/W	0x0	TRIG_SEL15: 11: 保留. 10: 保留 01: 始能 CCDR4 DN(EPWM 下数== CCDR4)触发转换采样 00: 不使能
29:28	TRIG_SEL14	R/W	0x0	TRIG_SEL14: 11: 保留. 10: 保留 01: 始能 CCR4 DN(EPWM 下数== CCR4)触发转换采样 00: 不使能
27:26	TRIG_SEL13	R/W	0x0	TRIG_SEL13: 11: 保留. 10: 保留 01: 始能 CCDR4 UP(EPWM 上数== CCDR4)触发转换采样 00: 不使能
25:24	TRIG_SEL12	R/W	0x0	TRIG_SEL12: 11: 保留. 10: 保留 01: 始能 CCR4 UP(EPWM 上数== CCR4)触发转换采样 00: 不使能

23:22	TRIG_SEL11	R/W	0x0	TRIG_SEL11: 11: 保留. 10: 保留 01: 始能 EPWM Counter OVF(上溢)触发转换采样 00: 不使能
21:20	TRIG_SEL10	R/W	0x0	TRIG_SEL10: 11: 保留. 10: 保留 01: 始能 EPWM Counter UDF(下溢)触发转换采样 00: 不使能
19:18	TRIG_SEL9	R/W	0x0	TRIG_SEL9: 11: 保留. 10: 保留 01: 始能 CMP1 Falling 触发转换采样 00: 不使能
17:16	TRIG_SEL8	R/W	0x0	TRIG_SEL8: 11: 保留. 10: 保留 01: 始能 CMP1 Rising 触发转换采样 00: 不使能

15:14	TRIG_SEL7	R/W	0x0	TRIG_SEL7: 11: 保留. 10: 保留 01: 始能 CMP0 Falling 触发转换采样 00: 不使能
13:12	TRIG_SEL6	R/W	0x0	TRIG_SEL6: 11: 保留. 10: 保留 01: 始能 CMP0 Rising 触发转换采样 00: 不使能
11:10	TRIG_SEL5	R/W	0x0	TRIG_SEL5: 11: 保留. 10: 保留 01: 始能 EPWM3P(CH3) Falling 触发转换采样 00: 不使能
9:8	TRIG_SEL4	R/W	0x0	TRIG_SEL4: 11: 保留. 10: 保留 01: 始能 EPWM2P(CH2) Falling 触发转换采样 00: 不使能

©Copyright 2026, PADAUK Technology Co. Ltd

7:6	TRIG_SEL3	R/W	0x0	TRIG_SEL3: 11: 保留. 10: 保留 01: 始能 EPWM1P(CH1) Falling 触发转换采样 00: 不使能
5:4	TRIG_SEL2	R/W	0x0	TRIG_SEL2: 11: 保留. 10: 保留. 01: 始能 EPWM3P(CH3) Rising 触发转换采样 00: 不使能
3:2	TRIG_SEL1	R/W	0x0	TRIG_SEL1: 11: 保留. 10: 保留 01: 始能 EPWM2P(CH2) Rising 触发转换采样 00: 不使能
1:0	TRIG_SEL0	R/W	0x0	TRIG_SEL0: 11: 保留. 10: 保留 01: 始能 EPWM1P(CH1) Rising 触发转换采样 00: 不使能

©Copyright 2026, PADAUK Technology Co. Ltd

14.7.5 ADC 转换缓存寄存器 (ADC_DAT0)

ADC 转换完成后，12 位的数字量被存放在 ADC_DAT 寄存器中，供应用软件读取．启动一次 ADC 转换后，需要等待转换过程结束后才能读到正确的结果，这可通过软件查询（等待 ADC_STAT.DONE 位为 1）或中断响应的方式（需开启系统 ADC 中断并提供中断服务程序）实现。软件读取一 次 ADC_DATA 内的数据或寫入 ADC_STAT.INT_CLR 为 1 后，ADC_STAT.DONE 位则被自动清除。

Bit	Name	R/W	Reset	Description
31:28	-	R	0x0	保留
27:16	CMPTH	R/W	0x0	CMPTH：比较阈值设置寄存器 当 ADC 结果比较使能开启后，其通道的 DATA 会与 CMPTH 比较,其比较结果将会放到 ADC_STAT.COMP_RESULT 寄存器中
15:0	DATA	R	0x0	DATA: 12 位 ADC 转换值（默认为右对齐格式，可通过配置 ADC_CON0.ALIGN 位将转换结果左对齐。） PGA0_O(AIN0) (参看 M/CH_SEL 参数)

14.7.6 ADC 转换缓存寄存器 (ADC_DAT1)

Bit	Name	R/W	Reset	Description
31:28	-	R	0x0	保留
27:16	CMPTH	R/W	0x0	CMPTH: 比较阈值设置寄存器 当 ADC 结果比较使能开启后，其通道的 DATA 会与 CMPTH 比较,其比较结果将会放到 ADC_STAT.COMP_RESULT 寄存器中
15:0	DATA	R	0x0	DATA: 12 位 ADC 转换值（默认为右对齐格式，可通过配置 ADC_CON0.ALIGN 位将转换结果左对齐。） PGA1_O(AIN1)(参看 M/CH_SEL 参数)

©Copyright 2026, PADAUK Technology Co. Ltd

14.7.7 ADC 转换缓存寄存器 (ADC_DAT2)

Bit	Name	R/W	Reset	Description
31:28	-	R	0x0	保留
27:16	CMPTH	R/W	0x0	CMPTH: 比较阈值设置寄存器 当 ADC 结果比较使能开启后，其通道的 DATA 会与 CMPTH 比较， 其比较结果将会放到 ADC_STAT.COMP_RESULT 寄存器中
15:0	DATA	R	0x0	DATA: 12 位 ADC 转换值（默认为右对齐格式，可通过配置 ADC_CON0.ALIGN 位将转换结果左对齐。） PA8(AIN2) (参看 M/CH _SEL 参数)

14.7.8 ADC 转换缓存寄存器 (ADC_DAT3)

Bit	Name	R/W	Reset	Description
31:28	-	R	0x0	保留
27:16	CMPTH	R/W	0x	CMPTH: 比较阈值设置寄存器 当 ADC 结果比较使能开启后，其通道的 DATA 会与 CMPTH 比较， 其比较结果将会放到 ADC_STAT.COMP_RESUL T 寄存器中
15:0	DATA	R	0x0	DATA: 12 位 ADC 转换值（默认为右对齐格式，可通过配置 ADC_CON0.ALIGN 位将转换结果左对齐。） PA9(AIN3) (参看 M/CH _SEL 参数)

©Copyright 2026, PADAUK Technology Co. Ltd

14.7.9 ADC 转换缓存寄存器（ADC_DAT4）

Bit	Name	R/W	Reset	Description
31:28	-	R	0x0	保留
27:16	CMPTH	R/W	0x0	CMPTH: 比较阈值设置寄存器 当 ADC 结果比较使能开启后，其通道的 DATA 会与 CMPTH 比较, 其比较结果将会放到 ADC_STAT.COMP_RESULT 寄存器中
15:0	DATA	R	0x0	DATA: 12 位 AD 转换值（默认为右对齐格式，可通过配置 ADC_CON0.ALIGN 位将转换结果左对齐。） PA10(AIN4) (参看 M/CH _SEL 参数)

14.7.10 ADC 转换缓存寄存器 (ADC_DAT5)

Bit	Name	R/W	Reset	Description
31:28	-	R	0x0	保留
27:16	CMPTH	R/W	0x0	CMPTH: 比较阈值设置寄存器 当 ADC 结果比较使能开启后，其通道的 DATA 会与 CMPTH 比较,其 比较结果将会放到 ADC_STAT.COMP_RESULT 寄存器中
15:0	DATA	R	0x0	DATA: 2 位 ADC 转换值（默认为右对齐格式，可通过配置 ADC_CON0.ALIGN 位将转换结果左对齐。） PA11(AIN5) (参看 M/CH _SEL 参数)

14.7.11 ADC 转换缓存寄存器 (ADC_DAT6)

Bit	Name	R/W	Reset	Description
31:28	-	R	0x0	保留
27:16	CMPTH	R/W	0x0	CMPTH: 比较阈值设置寄存器 当 ADC 结果比较使能开启后，其通道的 DATA 会与 CMPTH 比较,其比较结果将会放到 ADC_STAT.COMP_RESULT 寄存器中
15:0	DATA	R	0x0	DATA: 12 位 ADC 转换值（默认为右对齐格式，可通过配置 ADC_CON0.ALIGN 位将转换结果左对齐。） PA12(AIN6) (参看 M/CH _SEL 参数)

14.7.12 ADC 转换缓存寄存器 (ADC_DAT7)

Bit	Name	R/W	Reset	Description
31:28	-	R	0x0	保留
27:16	CMPTH	R/W	0x0	CMPTH: 比较阈值设置寄存器 当 ADC 结果比较使能开启后，其通道的 DATA 会与 CMPTH 比较,其比较结果将会放到 ADC_STAT.COMP_RESULT 寄存器中
15:0	DATA	R	0x0	DATA: 12 位 ADC 转换值（默认为右对齐格式，可通过配置 ADC_CON0.ALIGN 位将转换结果左对齐。） PA13(AIN7) (参看 M/CH _SEL 参数)

14.7.13 ADC 转换缓存寄存器 (ADC_DAT8)

Bit	Name	R/W	Reset	Description
31:28	-	R	0x0	保留
27:16	CMPTH	R/W	0x0	CMPTH: 比较阈值设置寄存器 当 ADC 结果比较使能开启后，其通道的 DATA 会与 CMPTH 比较,其比较结果将会放到 ADC_STAT.COMP_RESULT 寄存器中
15:0	DATA	R	0x0	DATA: 12 位 AD 转换值（默认为右对齐格式，可通过配置 ADC_CON0.ALIGN 位将转换结果左对齐。） PA6(AIN8) (参看 M/CH _SEL 参数)

14.7.14 ADC 转换缓存寄存器 (ADC_DAT9)

Bit	Name	R/W	Reset	Description
31:28	-	R	0x0	保留
27:16	CMPTH	R/W	0x0	CMPTH: 比较阈值设置寄存器 当 ADC 结果比较使能开启后，其通道的 DATA 会与 CMPTH 比较,其比较结果将会放到 ADC_STAT.COMP_RESULT 寄存器中
15:0	DATA	R	0x0	DATA: 12 位 ADC 转换值（默认为右对齐格式，可通过配置 ADC_CON0.ALIGN 位将转换结果左对齐。） PA7(AIN9) (参看 M/CH _SEL 参数)

14.7.15 ADC 转换缓存寄存器 (ADC_DAT10)

Bit	Name	R/W	Reset	Description
31:28	-	R	0x0	保留
27:16	CMPTH	R/W	0x0	CMPTH: 比较阈值设置寄存器 当 ADC 结果比较使能开启后，其通道的 DATA 会与 CMPTH 比较,其比较结果将会放到 ADC_STAT.COMP_RESULT 寄存器中
15:0	DATA	R	0x0	DATA: 12 位 AD 转换值（默认为右对齐格式，可通过配置 ADC_CON0.ALIGN 位将转换结果左对齐。） PA14(AIN10) (参看 M/CH _SEL 参数)

14.7.16 ADC 转换缓存寄存器 (ADC_DAT11)

Bit	Name	R/W	Reset	Description
31:28	-	R	0x0	保留
27:16	CMPTH	R/W	0x0	CMPTH: 比较阈值设置寄存器 当 ADC 结果比较使能开启后，其通道的 DATA 会与 CMPTH 比较,其比较结果将会放到 ADC_STAT.COMP_RESULT 寄存器中
15:0	DATA	R	0x0	DATA: 12 位 ADC 转换值（默认为右对齐格式，可通过配置 ADC_CON0.ALIGN 位将转换结果左对齐。） PA15(AIN11) (参看 M/CH_SEL 参数)

14.7.17 ADC 转换缓存寄存器 (ADC_DAT12)

Bit	Name	R/W	Reset	Description
31:28	-	R	0x0	保留
27:16	CMPTH	R/W	0x0	CMPTH: 比较阈值设置寄存器 当 ADC 结果比较使能开启后，其通道的 DATA 会与 CMPTH 比较，其比较结果将会放到 ADC_STAT.COMP_RESULT 寄存器中
15:0	DATA	R	0x0	DATA: 12 位 ADC 转换值（默认为右对齐格式，可通过配置 ADC_CON0.ALIGN 位将转换结果左对齐。） PB0(AIN12) (参看 M/CH_SEL 参数)

14.7.18 ADC 转换缓存寄存器 (ADC_DAT13)

Bit	Name	R/W	Reset	Description
31:28	-	R	0x0	保留
27:16	CMPTH	R/W	0x0	CMPTH: 比较阈值设置寄存器 当 ADC 结果比较使能开启后，其通道的 DATA 会与 CMPTH 比较，其比较结果将会放到 ADC_STAT.COMP_RESUL T 寄存器中
15:0	DATA	R	0x0	DATA: 12 位 ADC 转换值（默认为右对齐格式，可通过配置 ADC_CON0.ALIGN 位将转换结果左对齐。） PB1(AIN13) (参看 M/CH _SEL 参数)

©Copyright 2026, PADAUK Technology Co. Ltd

14.7.19 ADC 转换缓存寄存器 (ADC_DAT14)

Bit	Name	R/W	Reset	Description
31:28	-	R	0x0	保留
27:16	CMPTH	R/W	0x0	CMPTH: 比较阈值设置寄存器 当 ADC 结果比较使能开启后，其通道的 DATA 会与 CMPTH 比较,其比较结果将会放到 ADC_STAT.COMP_RESULT 寄存器中
15:0	DATA	R	0x0	DATA: 12 位 ADC 转换值（默认为右对齐格式，可通过配置 ADC_CON0.ALIGN 位将转换结果左对齐。） AVSS(AIN14) (参看 M/CH _SEL 参数)

14.7.20 ADC 转换缓存寄存器 (ADC_DAT15)

Bit	Name	R/W	Reset	Description
31:28	-	R	00	保留
27:16	CMPTH	R/W	0x0	CMPTH: 比较阈值设置寄存器 当 ADC 结果比较使能开启后，其通道的 DATA 会与 CMPTH 比较,其比较结果将会放到 ADC_STAT.COMP_RESULT 寄存器中
15:0	DATA	R	0x0	DATA: 12 位 ADC 转换值（默认为右对齐格式，可通过配置 ADC_CON0.ALIGN 位将转换结果左对齐。） ANA_SEL(AIN15) (参看 AMISC_VBUF_CR.ANA_SEL)

©Copyright 2026, PADAUK Technology Co. Ltd

14.7.21 ADC 刹车源配置 (ADC_BKSEL)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:14	ADCBKE2_EN	R/W	0x0	ADC 刹车源 2 始能选择: 00: ADC 刹车源 2 不使能刹车与 Reset ADCBKE2 Filter 计数器 01: ADC 刹车源 2 选择阈值比转换结果小使能刹车 10: ADC 刹车源 2 选择阈值比转换结果大使能刹车 11: ADC 刹车源 2 选择阈值和转换结果相等使能刹车 以上是根据 ADCBKE2_CH 所选的通道做比较 每次轉換結果會使 ADCBKE2 Filter 的值+1
13:12	ADCBKE2_FLT	R/W	0x0	ADC 刹车源 2 讯号滤波(ADCBKE2 Filter) 00: 1 次 ADC 转换结果符合(ADCBKE2_EN 选择)就刹车 01:连续 2 次 ADC 转换结果符合(ADCBKE2_EN 选择)就刹车 10:连续 4 次 ADC 转换结果符合(ADCBKE2_EN 选择)就刹车 11:连续 7 次 ADC 转换结果符合(ADCBKE2_EN 选择)就刹车 以上是根据 ADCBKE2_CH 所选的通道比较结果做滤波 注: ADCBKE2_EN = 00,才可写入 ADCBKE2_FLT
11:8	ADCBKE2_CH	R/W	0x0	ADC 通道选择为刹车源 2 0x0: 选择通道 0 为刹车源 2 0x1: 选择通道 1 为刹车源 2 0x2: 选择通道 2 为刹车源 2 0xF: 选择通道 15 为刹车源 2 注: ADCBKE2_EN = 00,才可写入 ADCBKE2_CH

7:6	ADCBKE1_EN	R/W	0x0	ADC 刹车源 1 始能选择: 00: ADC 刹车源 1 不使能刹车与 Reset ADCBKE1 Filter 计数 01: ADC 刹车源 1 选择阈值比转换结果小使能刹车 10: ADC 刹车源 1 选择阈值比转换结果大使能刹车 11: ADC 刹车源 1 选择阈值和转换结果相等使能刹车 以上是根据 ADCBKE1_CH 所选的通道做比较 每次轉換結果會使 ADCBKE1 Filter 计数器的值+1
5:4	ADCBKE1_FLT	R/W	0x0	ADC 刹车源 1 讯号滤波(ADCBKE1 Filter) 00: 1 次 ADC 转换结果符合(ADCBKE1_EN 选择)就刹车 01:连续 2 次 ADC 转换结果符合(ADCBKE1_EN 选择)就刹车 10:连续 4 次 ADC 转换结果符合(ADCBKE1_EN 选择)就刹车 11:连续 7 次 ADC 转换结果符合(ADCBKE1_EN 选择)就刹车 以上是根据 ADCBKE1_CH 所选的通道比较结果做滤波 注: ADCBKE1_EN=00,才可写入 ADCBKE1_FLT
3:0	ADCBKE1_CH	R/W	0x0	ADC 通道选择为刹车源 1 0x0: 选择通道 0 为刹车源 1 0x1: 选择通道 1 为刹车源 1 0x2: 选择通道 2 为刹车源 1 0xF: 选择通道 15 为刹车源 1 注: ADCBKE1_EN = 00,才可写入 ADCBKE1_CH

14.7.22 ADC 转换通道备份缓存寄存器 (ADC_BAKDAT)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:0	ADC_BAKDAT	R	0x0	ADC_BAKDAT: 设定 ADC_CON0.BAKEN = 1 时, 当通道转换完成时, 同时会将此通道上次转换的值备份此寄存器.

15. 模拟功能控制(AMISC)

15.1. 概述

PEC930 包含多个需要用户配置的模拟模块。本章还包括相关的杂项控制，以便于软件管理。这些杂项控制通过寄存器实现，用户可以通过配置相应寄存器来管理各种模拟功能。更独立的模拟模块（例如 ADC）具有各自独立的控制器，因此不在本章描述范围内。

15.2. LVD/LVR 功能简介

当 LVD（低压检测）模块被使能后，芯片会持续监测供电电压 VDD。用户可配置 LVD 的滤波时钟参数，并通过 LVD 输出位查看其状态。当 VDD 低于设定的 LVD 阈值且 LVD 中断被使能时，系统将触发一次中断。

当 LVR（低压复位）模块被使能后，芯片同样会持续监测 VDD。当 VDD 低于设定的 LVR 阈值时，系统将触发复位，芯片将重新进入初始化阶段。

详细配置请参考 AMISC_LVD_LVR_CR 寄存器。

15.3. PGA 功能简介

PEC930 芯片包含 3 个基本运算放大器模块和 2 个可编程增益放大器（增益范围: 1X～16X）。配合少量外围器件即可实现基本的信号放大和信号运算功能。

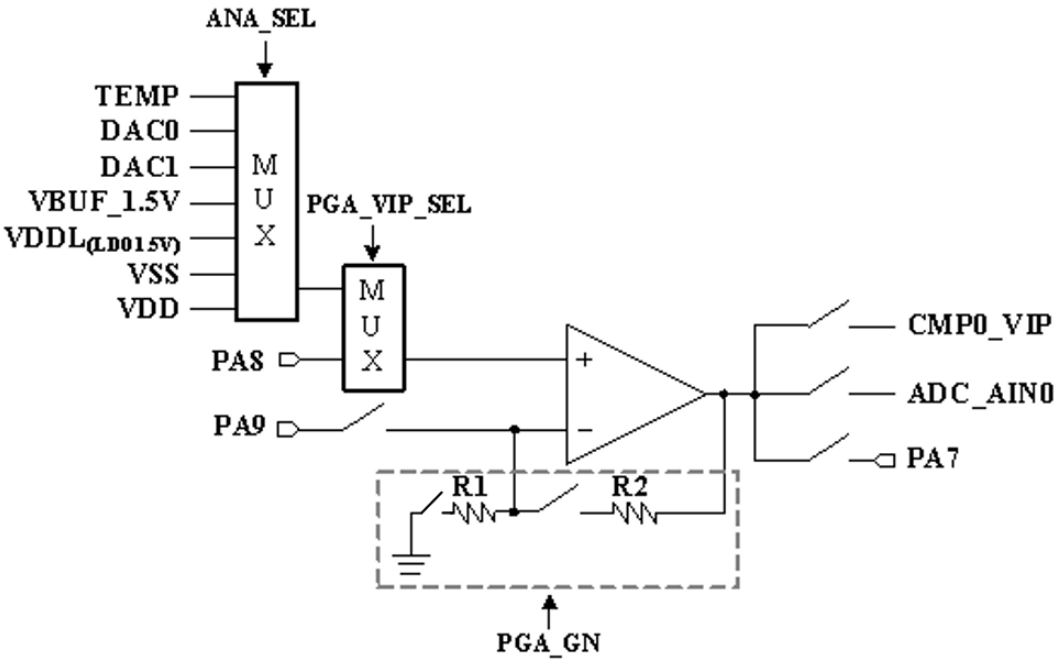


图 15-1 PGA0 方块图

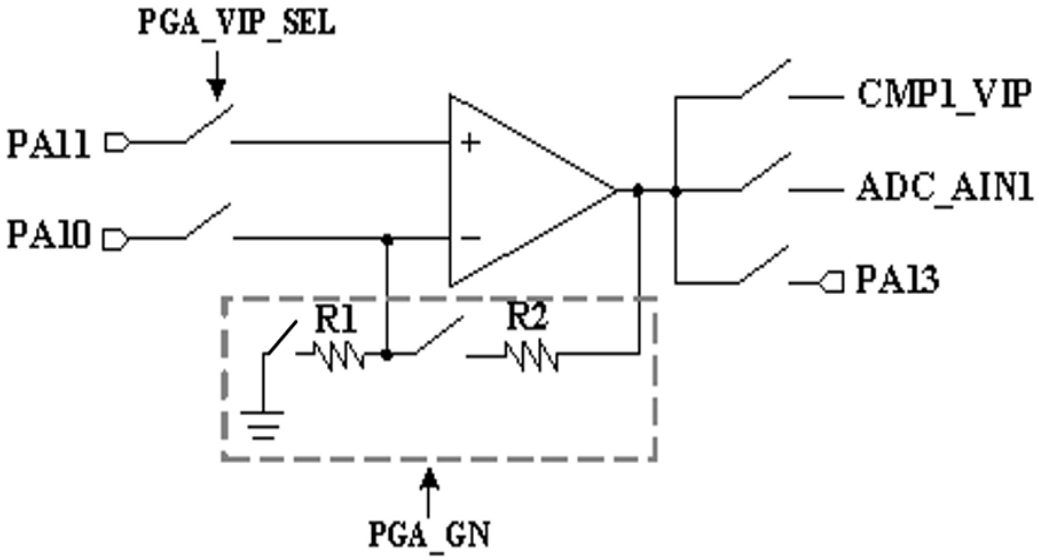


图 15-2 PGA1 方块图

- PGA0/1 使用芯片内部的可编程增益配置。

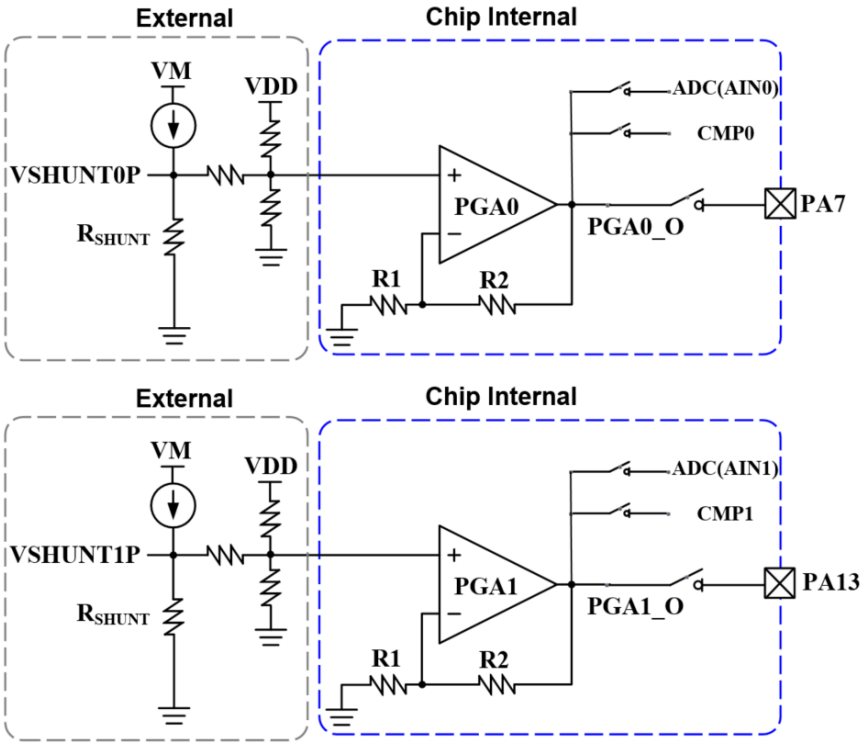


图 15-3 PGA 内部增益方块图

- PGA0/1 使用外部电阻配置增益。

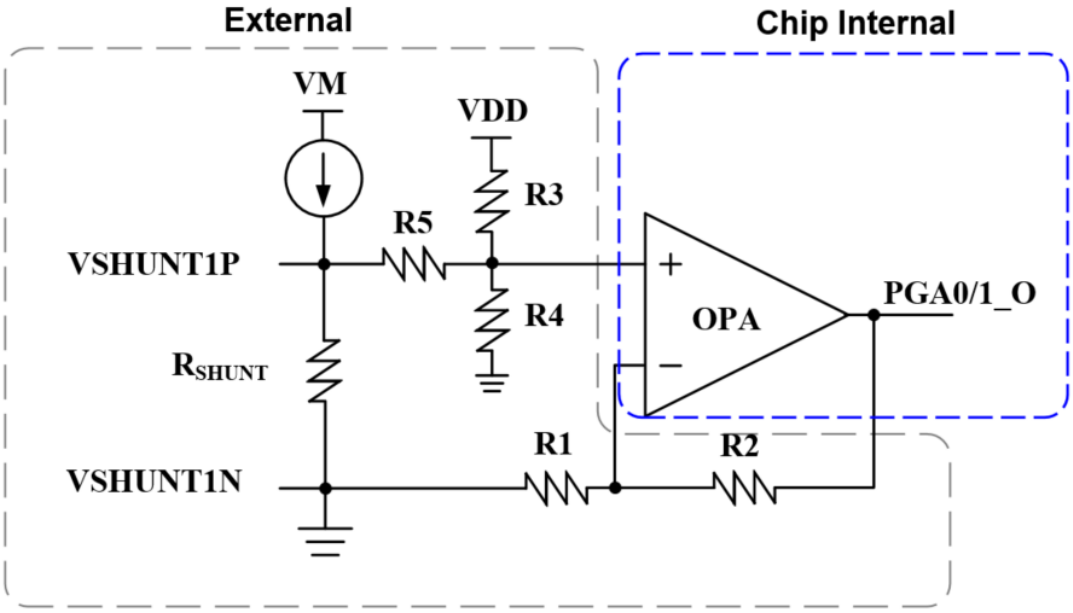


图 15-4 PGA 外部增益方块图

15.4. 寄存器列表

地址	寄存器	描述
0x4000_5800	AMISC_LVD_LVR_CR	LVD/LVR 控制寄存器
0x4000_5804	AMISC_VBUF_CR	VBUF 控制寄存器
0x4000_5810	AMISC_DAC_CR	DAC 控制寄存器
0x4000_5820	AMISC_HSI_CR	HIRC 控制寄存器
0x4000_5824	AMISC_LSI_CR	LIRC 控制寄存器
0x4000_5830	AMISC_ADC_AIN_CR	ADC 類比輸入控制寄存器
0x4000_5860	HWTRIM_LDO_TRIM	LDO 校准讀取寄存器
0x4000_5864	HWTRIM_VBUF_TRIM	VBUF 校准讀取寄存器
0x4000_5868	HWTRIM_HSI_TRIM	HIRC 校准讀取寄存器
0x4000_586C	HWTRIM_LSI_TRIM	LIRC 校准讀取寄存器
0x4000_5870	HWTRIM_MISC_CFG	MISC 配置讀取寄存器
0x4000_5880	OPAMP0_PGA_CR	OPAMP0(PGA0) 控制寄存器
0x4000_5888	OPAMP1_PGA_CR	OPAMP1(PGA1) 控制寄存器

表 15-1 AMISC 寄存器列表

©Copyright 2026, PADAUK Technology Co. Ltd

15.5. 寄存器描述

15.5.1 LVD/LVR控制寄存器 (AMISC_LVD_LVR_CR)

Bit	Name	R/W	Reset	Description
31:16	-	R	0x0	保留
15:13	LVR_SEL	R/W	0x0	LVR 电压阈值选择 0x0 : 2.0V 0x1 : 2.4V 0x2: 2.7V 0x3: 3.0V 0x4: 3.7V Others: 保留
12	-	R	0x0	保留
11	LVD_STATE	R	0x1	VDD 与所选 LVD_SEL 对应的输出状态 0 : VDD 低于所选 LVD 阈值 1 : VDD 高于所选 LVD 阈值
10	TEMP_EN	R/W	0x1	温度传感器使能 0 :关闭 1 :使能
9	LVD_INT_EN	R/W	0x0	LVD 中断使能; 0 : LVD 中断关闭 1 : LVD 中断使能
8	LVR_EN	R/W	0x0	LVR 使能控制; 0 : LVR 关闭 1 : LVR 使能(需使能 AMISC_VBUF_CR.VBUF_EN)

©Copyright 2026, PADAUK Technology Co. Ltd

7:5	LVD_SEL	R/W	0x1	LVD 电压阈值选择 0x0 : 2.0V 0x1 : 2.2V 0x2: 2.4V 0x3: 2.7V 0x4: 3.0V 0x5: 3.7V 0x6: 4.0V 0x7: 4.3V
4	-	R/W	0x0	保留
3:2	LVD_F_SEL	RW	0x0	LVD 滤波时间长度选择 0x0: 3*HCLK 0x1: 3*(HCLK/2) 0x2: 3*(HCLK/4) 0x3: 3*(HCLK/8)
1	LDO_LP_EN	R/W	0x0	LDO 低功耗模块使能 0: 关闭 1: 使能
0	LVD_EN	R/W	0x0	LVD 使能控制 0: LVD 关闭 1: LVD 使能(需使能 AMISC_VBUF_CR.VBUF_EN)

15.5.2 VBUF控制寄存器 (AMISC_VBUF_CR)

Bit	Name	R/W	Reset	Description
31:13	-	R	0x0	保留
12:6	ANA_SEL	R/W	0x0	模拟讯号选择： 0x0: 保留 0x1: TEMP（需使能 AMISC_LVD_LVR_CR.TEMP_EN） 0x2：DAC0（需使能 AMISC_DAC_CR.DAC0_EN） 0x4：DAC1（需使能 AMISC_DAC_CR.DAC1_EN） 0x8：VBUF 1.5V（需使能 AMISC_VBUF_CR.VBUF_EN） 0x10: VDDL 0x20: VSS 0x40: VDD Others: 保留 使用说明：须配合 ANA2PGA_EN、ANA2IO_EN 或 AMISC_ADC_AIN_CR.ANA2ADC_EN 任一寄存器。
5	ANA2PGA_EN	R/W	0x0	模拟讯号輸出到 OPAMP0(PGA0); 0: 关闭 1: 使能，配合 ANA_SEL 选择连接到 OPAMP0(PGA0)正端輸入（参考 OPAMP0_PGA_CR.PGA_VIP_SEL 寄存器）
4	ANA2IO_EN	R/W	0x0	模拟讯号輸出到 IO(PA8)： 0: 关闭 1: 使能，配合 ANA_SEL 选择讯号输出
3:1		R	0	保留
0	VBUF_EN	R/W	0	VBUF 使能控制 0: 关闭 1: 使能

15.5.3 DAC控制寄存器 (AMISC_DAC_CR)

Bit	Name	R/W	Reset	Description
31:24	-	R	0x0	保留
23:22	-	R	0x0	保留
21	DAC1_EN	R/W	0	DAC1 使能控制 0: 关闭 1: 使能
20	DAC0_EN	R/W	0	DAC0 使能控制 0: 关闭 1: 使能
19:10	DAC1B	R/W	0x200	DAC1 位输入
9:0	DAC0B	R/W	0x200	DAC0 位输入

Note: 当进入DeepSleep模式时，需关闭DAC：设定DAC1/0_EN=0x0 且 DAC1/0B = 0x0

15.5.4 HIRC控制寄存器 (AMISC_HSI_CR)

Bit	Name	R/W	Reset	Description
31:24	HSI_EN	R/W	0x1	HIRC 使能控制 (写: 0x80，读: 0x00) HIRC 关闭 (写:0x01，读: 0x01) HIRC 开启 写其它值状态维持不变 注: 慎重关闭 HIRC 时钟！
23:16	-	R	0x0	保留
15:0	LDO_SEL	R/W	0x0101	HIRC 电源来源选择 (写:0x8080，读: 0x0000):选与 LDO 共享电源 (由于封装打线关系，建议选择此选项) (写:0x8001，读: 0x0001):禁用 (写:0x0180，读: 0x0100):关 HIRC 专用 LDO 电源 (写:0x0101，读: 0x0101):选 HIRC 专用 LDO 电源 写其它值状态维持不变 注: 慎重选择 HIRC 电源来源！

15.5.5 LIRC控制寄存器 (AMISC_LSI_CR)

Bit	Name	R/W	Reset	Description
31:24	LSI_EN	R/W	0x1	LIRC 使能控制 (写: 0x80，讀: 0x00)LIRC 关闭 (写: 0x01，讀: 0x01)LIRC 使能 写其它值状态维持不变 注: 慎重关闭 LIRC 时钟！

23:0	-	R	0x0	保留

15.5.6 ADC模拟输入控制寄存器 (AMISC_ADC_AIN_CR)

Bit	Name	R/W	Reset	Description
31:3	-	R	0x0	保留
2:1	-	R/W	0x3	保留
0	ANA2ADC_EN	R/W	0x1	模拟讯号输出到 ADC; 0: 关闭 1: 使能, 配合 AMISC_VBUF_CR.ANA_SEL 选择讯号连接到 ADC AIN15(参考 ADC_CON0.M 寄存器)

15.5.7 LDO校准讀取寄存器 (HWTRIM_LDO_TRIM)

Bit	Name	R/W	Reset	Description
31:12	-	R	0x0	Reserved
11:10	LDO_TUNE	R	0x2	LDO 高温漏电补偿 0x0: strength 0x1: weak 0x2: middle 0x3: off
9:5	LDO_LP_TRIM	R	0x10	LDO 低功耗模块校准 0x0: [vmax] 0x1F: [vmin]
4:0	LDO_TRIM	R	0x10	LDO 校准 0x0: [vmax] 0x1F: [vmin]

15.5.8 VBUF校准讀取寄存器 (HWTRIM_VBUF_TRIM)

Bit	Name	R/W	Reset	Description
31:7	-	R	0x0	保留
6:0	VBUF_TRIM	R	0x20	VBUF 1.5V 电压校准 0x0: 保留 0x1: [vmin] 0x3F: [vmax] 注: 位 6 任意值

15.5.9 HIRC校准讀取寄存器 (HWTRIM_HSI_TRIM)

Bit	Name	R/W	Reset	Description
31:23	-	R	0x0	保留
22:18	-	R	0x10	保留
17:16	HSI_TC	R	0x2	HIRC 温度补偿校准 0x0: [fmax] 0x3: [fmin]
15:9	HSI_FSEL_CFG	R	0x40	HIRC 频率高位校准 0x0: [fmin] 0x70: [fmax]
8:0	HSI_D_CFG	R	0x0	HIRC 频率低位校准 0x0: [fmin] 0x1FF: [fmax]

15.5.10 LIRC校准讀取寄存器 (HWTRIM_LSI_TRIM)

Bit	Name	R/W	Reset	Description
31:8		R	0x0	保留
7:0	LSI_TRIM	R	0xB3	LIRC 频率校准 0x0: [fmin] 0xFF: [fmax]

15.5.11 MISC 配置讀取寄存器 (HWTRIM_MISC_CFG)

Bit	Name	R/W	Reset	Description
31:2	-	R	0x0	保留
1	-	R	0x0	保留
0	EXT_nRST_EN	R	0x1	外部复位使能控制 1: 使能 0: 关闭

15.5.12 OPAMPn (PGAn) 控制寄存器 (OPAMPn_PGA_CR, n=0,1)

Bit	Name	R/W	Reset	Description
31:12	-	R	0x0	保留
11	PGA_VIP_SEL	R/W	0x0	OPAMP(PGA) P 端选择 输入到 PGA0 0: 内部讯号 (参考 AMISC_VBUF_CR.ANA2PGA_EN 和 AMISC_VBUF_CR.ANA_SEL 寄存器使用) 1: IO(PA8) 输入到 PGA1 0: 浮接 1: IO(PA11)
10	PGA_VIN_SEL	R/W	0x0	OPAMP(PGA) N 端選擇: 0: 使用內部增益電路 1: IO(PA9) 輸入到 PGA0 IO(PA10) 輸入到 PGA1
9:8	-	R	0x0	保留
7:2	PGA_GAIN	R/W	0x0	OPAMP (PGA) 增益選擇: 0x0: COMP 模块或外部 PGA 模块 0x1: X1 0x6: X2 0xA: X3 0xE: X4 0x12: X5 0x16: X6 0x1A: X7 0x1E: X8

				0x22: X9 0x26: X10 0x2A: X11 0x2E: X12 0x32: X13 0x36: X14 0x3A: X15 0x3E: X16 others: 保留
1	PGA_IO_EN	R/W	0x0	OPAMP (PGA) 輸出到 IO 使能控制 0: 关闭 1: 使能, PGA0 輸出到 IO(PA[7]) PGA1 輸出到 IO(PA[13])
0	PGA_EN	R/W	0x0	OPAMP(PGA) 使能控制 0: 关闭 1: 使能

©Copyright 2026, PADAUK Technology Co. Ltd

16. 循环冗余校验计算单元(CRC)

16.1 概述

循环冗余校验（CRC）是一种错误侦测码，本模块根据固定的多项式对任意字节数据进行 CRC 计算。在实际应用中，CRC 主要用于数据传输完整性校验、存储数据一致性验证、通信错误检测。

CRC 的应用流程如下: 数据发送端（计算并附加 CRC） → 传输 → 接收端（重新计算并校验 CRC）

16.2 功能描述

本模块依据 ISO/IEC 13239 标准实现，支持的生成多项式为:

(1) CRC-16：$x^{16} + x^{12} + x^5 + 1$（对应多项式：0x1021）

(2) CRC-32：$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$（对应多项式：0x04C11DB7）

初始值: 固定为 0xFFFFFFFF

功能分为: CRC 编码及 CRC 校验

支持三种数据输入宽度: 8 位（byte）、16 位（half-word）、32 位（word）

16.2.1 CRC 编码模式

本模式用于计算原始数据的 CRC 值。该值通常会被附加在原始数据包的末尾，以形成带有校验机制的完整传输帧。

操作步骤:

(1) 模式配置：配置 CRC_CR.POLYSEL 寄存器以选择多项式。
(2) 初始化：向 CRC_DOUT 寄存器写入初始值 0xFFFFFFFF。
(3) 数据输入：依序将原始数据写入 CRC_DIN 寄存器。
- 输入宽度：支持 8 / 16 / 32 位 位宽写入。
- 顺序要求：数据必须按原始顺序逐笔输入。

(4) 结果读取：计算完成后，从 CRC_DOUT 寄存器读取最终 CRC 值。

16.2.2 CRC 校验模式

本模式用于验证接收端数据的完整性。通过对接收到的完整数据（原始数据 + CRC）进行除法计算，判断余数是否为零。

操作步骤:

(1) 模式配置: 配置 CRC_CR.POLYSEL 寄存器以选择多项式。
(2) 初始化: 向 CRC_DOUT 寄存器写入初始值 0xFFFFFFFF。
(3) 数据输入: 依序将完整数据（原始数据 + CRC）写入 CRC_DIN 寄存器。
- 输入宽度: 支持 8 / 16 / 32 位 位宽写入。
- 顺序要求: 数据必须按原顺序输入；CRC 值必须采用小端序 (Little-Endian) 写入（先写低位，再写高位）。

(4) 结果判断: 读取 CRC_CR.VERF 位状态。根据其结果判断校验是否通过。

16.3 寄存器列表

地址	寄存器	描述
0x4001_E000	CRC_CR	CRC 控制寄存器
0x4001_E004	CRC_DIN	CRC 数据输入寄存器
0x4001_E008	CRC_DOUT	CRC 结果输出寄存器

表 16-1 CRC 寄存器列表

16.4 寄存器描述

16.4.1 CRC 控制寄存器 (CRC_CR)

Bit	Name	R/W	Reset	Description
31:2	-	R	0x0	保留
1	VERF	R	0x0	CRC校验结果标志： 用于指示当前数据包的完整性校验结果，在校验模式下，必须在所有原始数据及CRC校验码（16/32位）全部写入CRC数据输入寄存器（CRC_DIN）之后，方可读取此标志位 0: 校验失败 1: 校验通过
0	POLYSEL	R/W	0x0	CRC 生成多项式选择： 选择模组运行的 CRC 数学模型 1: CRC-16 → $x^{16} + x^{12} + x^5 + 1$(0x1021) 0: CRC-32 → $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$(0x04C11DB7)

16.4.2 CRC 数据输入寄存器 (CRC_DIN)

Bit	Name	R/W	Reset	Description
31:0	DIN	R/W	0x0	CRC数据输入寄存器： 用于输入 CRC 待计算的数据. 支持 8 位、16 位及 32 位位宽。硬件将根据写入的指令位宽自动处理输入流

16.4.3 CRC 结果输出寄存器 (CRC_DOUT)

Bit	Name	R/W	Reset	Description
31:0	DOUT	R/W	0x0	CRC 结果输出寄存器： 用于存储计算过程的中间值及最终结果，并兼具复位功能 读取操作：根据标准协议，返回值为当前内部寄存器值的按位取反(Bitwise NOT)，该设计确保读取的结果直接符合协议要求的最终CRC校验值 写入操作：写入0xFFFFFFFF在此寄存器会将CRC计算重设为其初始状态

17. DSP 硬件加速模块

17.1 概述

DSP 硬件加速模块包含一个 32 位有符号除法器以及一个 32 位无符号平方根运算单元。用户可通过 APB 总线访问并配置这些功能单元，通过写入输入数据、触发运算并读取相应结果，以加速 DSP 相关的计算任务。

每个运算单元完成一次运算均需消耗一定数量的时钟周期（详见功能描述章节中的具体时钟周期说明）。

32 位有符号除法器与 32 位无符号平方根运算单元共用输入配置寄存器与结果寄存器。用户在开始运算前必须选择所需的运算模式。当前运算完成之前，不能启动下一次运算。在此期间，对数据源配置寄存器的写操作将无效（被忽略）。

17.2 功能描述

开始运算前，用户必须通过写 DSP_CR 寄存器中的 MODE 字段配置所需的运算模式。对于相同的运算模式，无需重复配置。在对应运算单元运行期间修改此字段将被忽略（参见后续章节的运算流程描述）。

写入输入数据时，用户必须最后写入第二个数据源配置寄存器（例如 DSP_SDAT2），以确保正确触发运算。

17.2.1 32 位有符号数除法

32 位有符号除法运算的数学表达如下:

$$(\mathbf{signed})\ \mathbf{X[31:0]} = \frac{(\mathbf{signed})\ \mathbf{A[31:0]} - (\mathbf{signed})\ \mathbf{Y[31:0]}}{(\mathbf{signed})\ \mathbf{B[31:0]}}$$

- A 输入有效范围: -2^{31} ~ $2^{31}-1$
- B 输入有效范围: -2^{31} ~ $2^{31}-1$
- X 输出有效范围: -2^{31} ~ $2^{31}-1$
- Y 输出有效范围: -2^{31} ~ $2^{31}-1$

除法器采用多周期算法，计算需要 18 个时钟周期。

软件运算步骤如下:

1. 写 DSP_SDAT1 配置被除数 A[31:0]（1 次写指令周期）。
2. 写 DSP_SDAT2 配置除数 B[31:0] 并触发运算（1 次写指令周期）。
3. 等待 18 个时钟周期后，读取 DSP_SR 寄存器中的 DONE 位，确认运算完成（1 次读指令周期）。
4. 读取 DSP_RSLT1 与 DSP_RSLT2 获得商 X[31:0] 与余数 Y[31:0]（共 2 次读指令周期）。

除法运算的控制流程图如下所示:

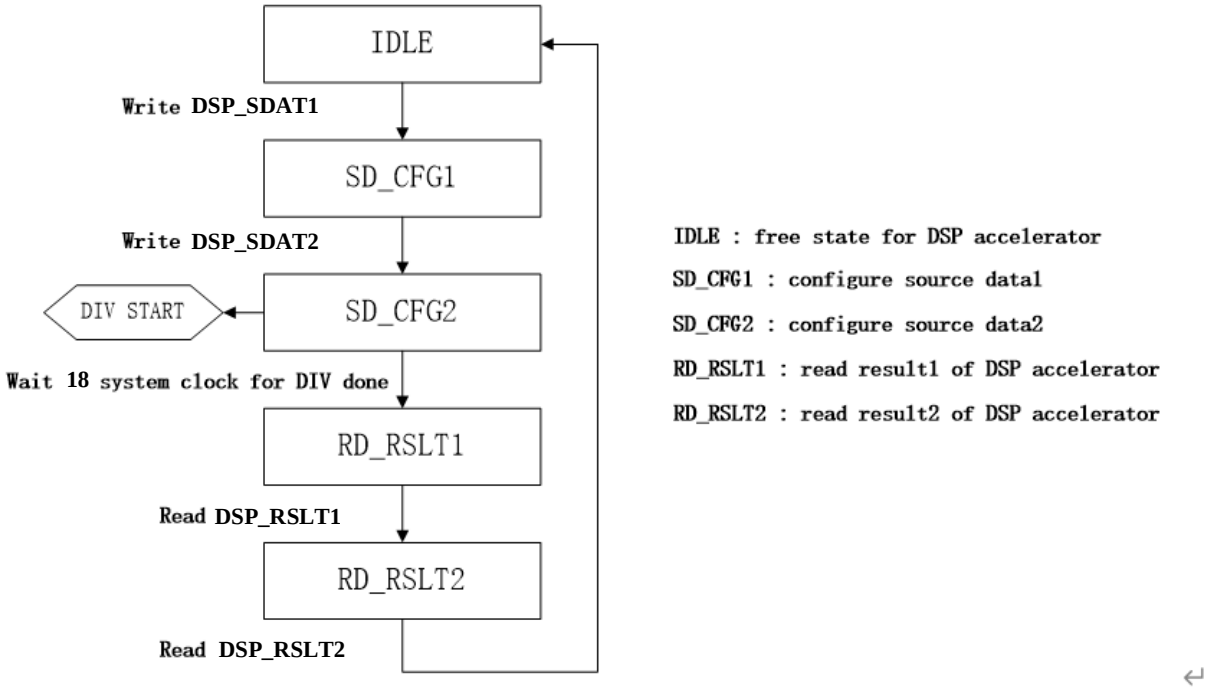


图 17-1 除法运算的控制流程图

17.2.2 32 位无符号平方根

32 位无符号平方根运算的数学表达式如下:

$$\textbf{(unsigned) X[16:0]} = \sqrt{\textbf{(unsigned) A[31:0]}}$$

- A 输入有效范围: $0 \sim 2^{32} - 1$
- X 输出有效范围: $0 \sim 2^{16} - 1$

平方根运算单元采用多周期开根算法，运算周期固定为 18 个系统时钟周期

软件运算步骤:

1. 写 DSP_SDAT1 寄存器，配置输入 A[31:0]（1 个写指令周期）。
2. 写 DSP_SDAT2 寄存器，写入 0 以触发平方根运算（1 个写指令周期）。
3. 等待 18 个系统时钟周期，然后读取 DSP_SR 寄存器的 DONE 位，确认运算完成（1 个读指令周期）。
4. 读取 DSP_RSLT1 寄存器，获取平方根结果 X[16:0]（1 个读指令周期）。

平方根运算的状态控制流程图如下:

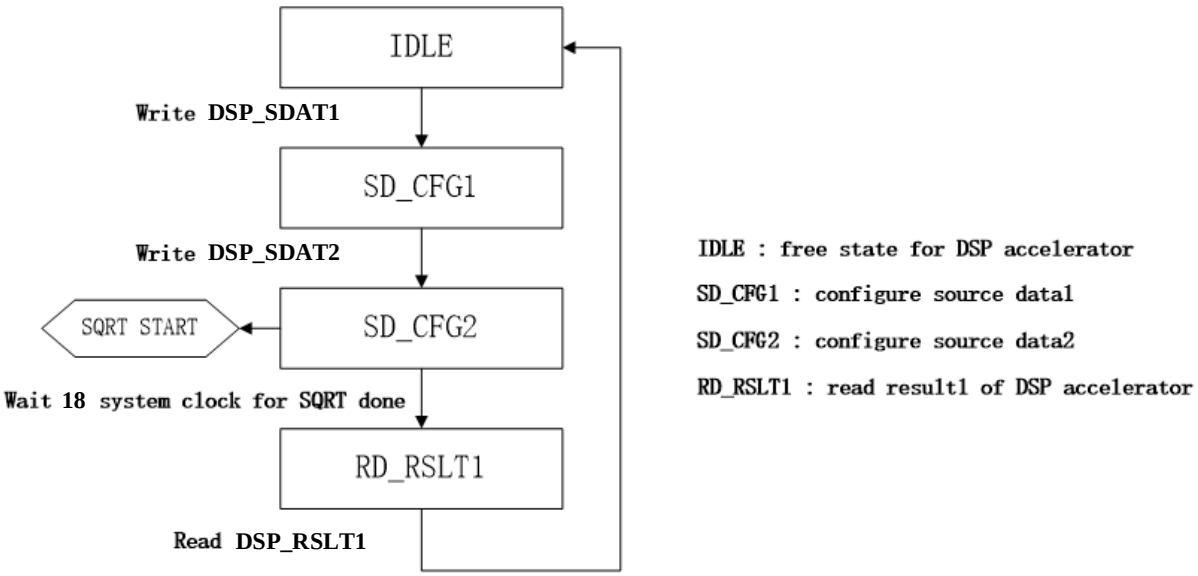


图 17-2 平方根运算的控制流程图

17.3 工作时序

17.3.1. 32 位有符号数除法

32 位有符号除法的工作时序如下图所示:

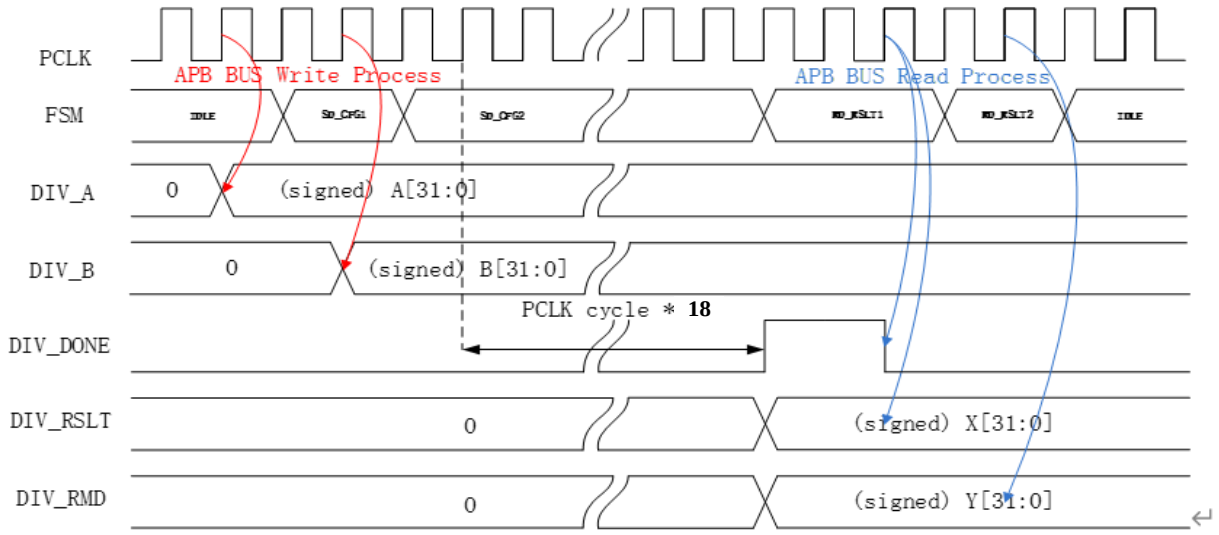


图 17-3 32 位有符号除法的工作时序

17.3.2. 32 位无符号平方根

32 位无符号开平方根的操作时序如下图所示:

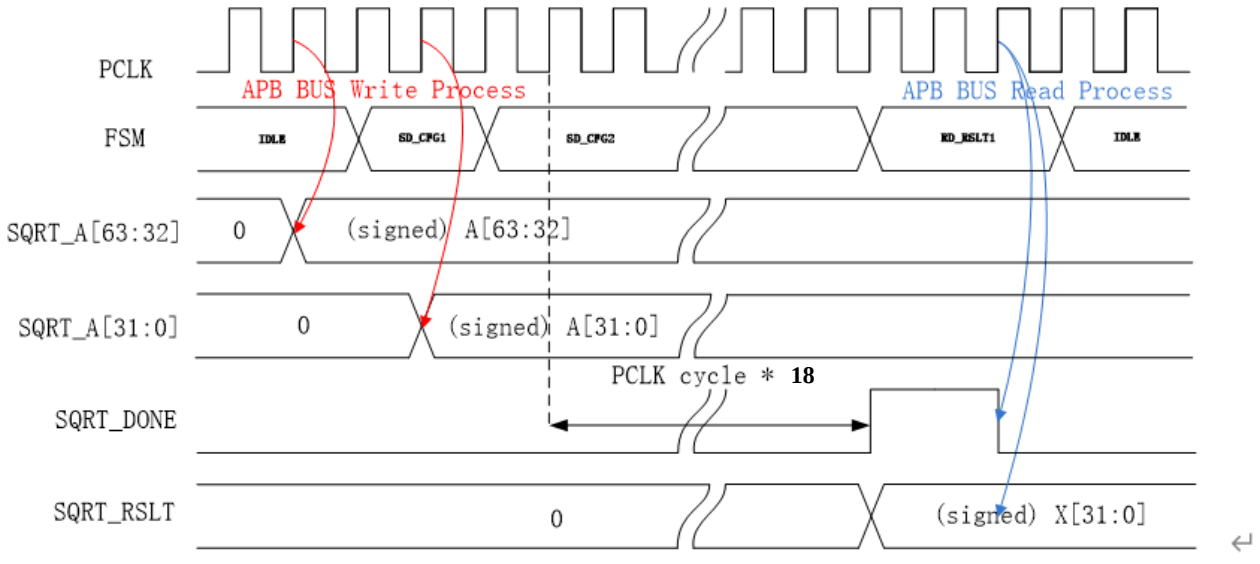


图 17-4 32 位无符号平方根的工作时序

17.4 寄存器列表

地址	寄存器	描述
0x4000_8000	DSP_CR	DSP 硬件加速控制寄存器
0x4000_8004	DSP_SR	DSP 硬件加速状态寄存器
0x4000_8008	DSP_SDAT1	DSP 硬件加速数据源 1 寄存器
0x4000_8010	DSP_SDAT2	DSP 硬件加速数据源 2 寄存器
0x4000_8018	DSP_RSLT1	DSP 硬件加速结果 1 寄存器
0x4000_8020	DSP_RSLT2	DSP 硬件加速结果 2 寄存器

表 17-1 DSP 寄存器列表

17.5 寄存器描述

17.5.1 DSP 硬件加速控制寄存器 (DSP_CR)

Bit	Name	R/W	Reset	Description
31:3	-	R	0x0	保留
2:0	MODE	R/W	0x0	操作模式选择 （必须在开始新的运算模式之前进行配置，且在运算过程中不得修改） 0x1: 32 位有符号除法模式 0x4: 32 位无符号平方根模式 其他: 保留

17.5.2 DSP 硬件加速状态寄存器 (DSP_SR)

Bit	Name	R/W	Reset	Description
31:1	-	R	0x0	保留
0	DONE	R	0x0	32 位有符号除法 / 32 位无符号平方根运算完成标志 0: 运算空闲 1: 运算完成 （在读取DSP_RSLT1 寄存器或 DSP_RSLT2 寄存器后自动清零）

17.5.3 DSP 硬件加速数据源 1 寄存器 (DSP_SDAT1)

Bit	Name	R/W	Reset	Description
31:0	SDAT1	R/W	0x0	不同运算模式下的数据源 1 配置如下: 32 位除法运算: 输入 A[31:0]（32 位有符号数） 32 位平方根运算: 输入 A[31:0]（32 位无符号数）

17.5.4 DSP 硬件加速数据源 2 寄存器 (DSP_SDAT2)

Bit	Name	R/W	Reset	Description
31:0	SDAT2	R/W	0x0	不同运算模式下的数据源 2 配置如下: 32 位除法运算: 输入 B[31:0]（32 位有符号数） 32 位平方根运算: 写入 0（32 位无符号数）

17.5.5 DSP 硬件加速结果 1 寄存器 (DSP_RSLT1)

Bit	Name	R/W	Reset	Description
31:0	RSLT1	R	0x0	不同运算模式下的结果 1: 32 位除法运算: 商输出 X[31:0] 32 位平方根运算: 结果输出 X[31:0]

17.5.6 DSP 硬件加速结果 2 寄存器 (DSP_RSLT2)

Bit	Name	R/W	Reset	Description
31:0	RSLT2	R	0x0	不同运算模式下的结果 2: 32 位除法运算: 余数输出 Y[31:0] 32 位平方根运算: 无效 / 保留

©Copyright 2026, PADAUK Technology Co. Ltd

18. 比较器 (COMP)

18.1 概述

比较器模块负责比较两路模拟输入信号。其输出状态由正负输入端的极性关系决定。

比较器的输出状态可通过对应的寄存器位读取。根据应用需求，该比较结果可用于产生内部中断，或导向外部设备端脚（例如: 端口输出信号）。

比较器输出将经过数字滤波模块处理，以抑制信号在转换期间可能产生的毛刺（glitch）。该滤波机制可确保内部中断产生与外部端脚信号导向的可靠性。

18.2 结构框图

下图显示了比较器模块的内部框图:

VC0A_O 和 VC1A_O 的使用参考 SYSCFG_TIM2_CON_SEL 和 SYSCFG_EPWM_CON_SEL 寄存器中的 COMP1_0A_SEL 位。

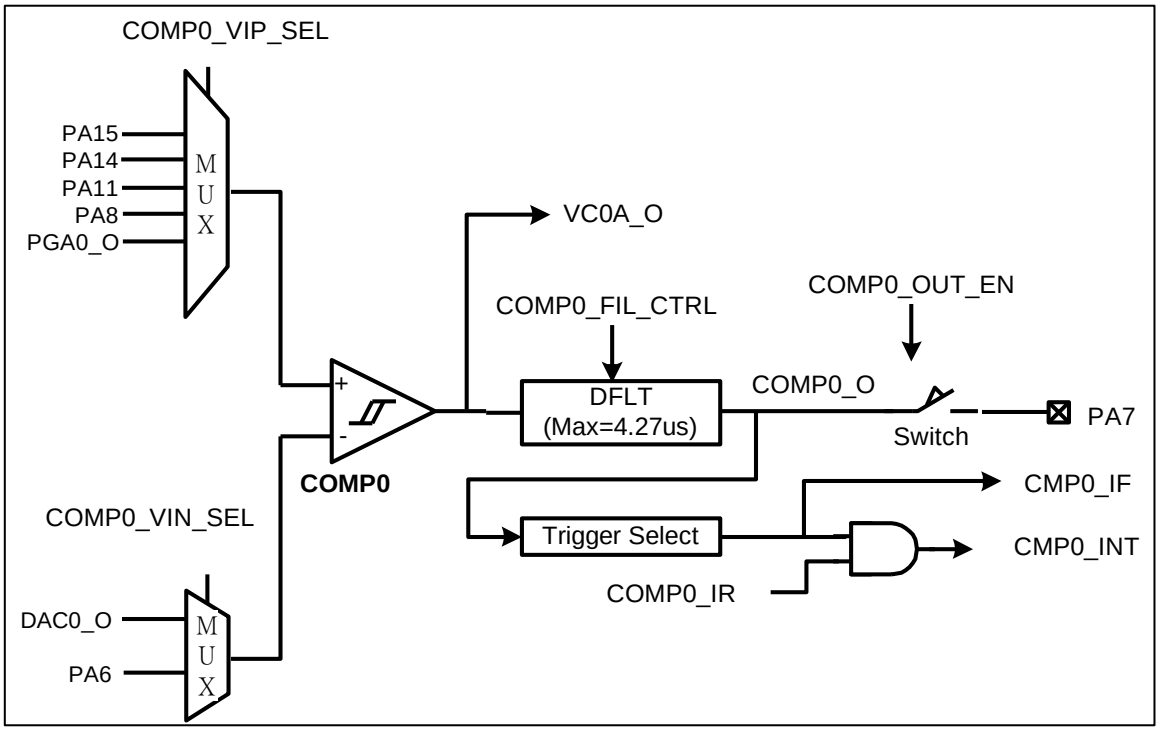


图 18-1 COMP0 方块图

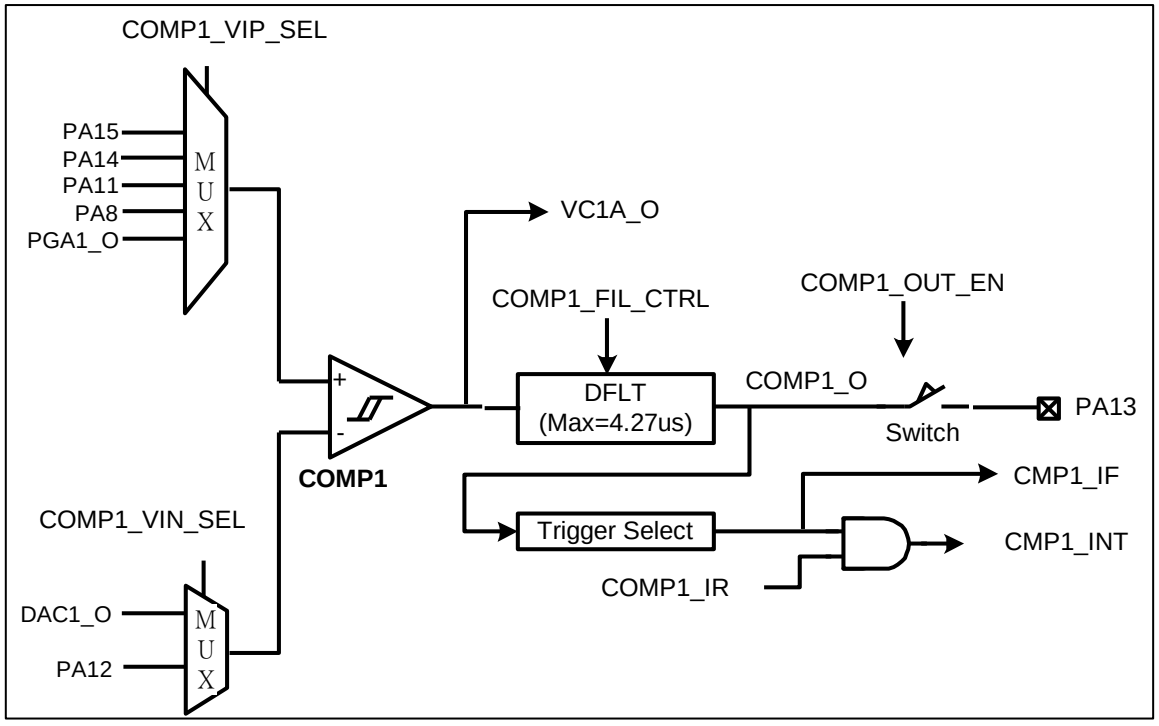


图 18-2 COMP1 方块图

18.3 功能描述

18.3.1 比较器控制

当比较器的正端输入（V+）大于负端输入（V−）时，比较器输出为高电平；否则，输出为低电平。

正端输入和负端输入均可选择。正端输入可以连接到 I/O 端口或 PGA 输出，由 COMP_VIPSEL 寄存器控制。负端输入可以连接到 I/O 端口或 DAC 输出，由 COMP_CTRL.VIN_SEL 位控制。详细配置请参阅相关寄存器描述。

当比较器被禁用时，其输出保持低电平。当启用时，比较器输出在初始化延迟和比较器传输延迟之后有效。初始化延迟取决于负端输入的选择:

- 当负端输入为 I/O 电压时，最大初始化延迟为 17.1 μs。
- 当负端输入为 DAC 输出时，最大初始化延迟为 34.2 μs。

注意:

每次用户更新 DAC 输入数据寄存器 AMISC_DAC_CR（参见第 15 节 – 模拟功能控制）时，都需要执行初始化延迟，以确保 DAC 输出电压稳定。

默认初始化延迟为 60 个系统时钟周期（基于 60 MHz 系统时钟）。假如系统时钟频率发生变化，初始化延迟计数必须相应调整。延迟通过 COMP_INITCNT 寄存器配置，其中 INIT_DELAY 指定系统时钟周期数。

在初始化延迟之后，数字滤波模块被启用。该滤波器抑制输入电压交越导致的比较器输出毛刺，从而在传输延迟后获得有效且稳定的比较器结果。数字滤波器引入了一个额外的延迟，这个延迟是加在传播延迟之上。

具体的比较器控制时序如下:

(1) 负端输入为 IO 电压

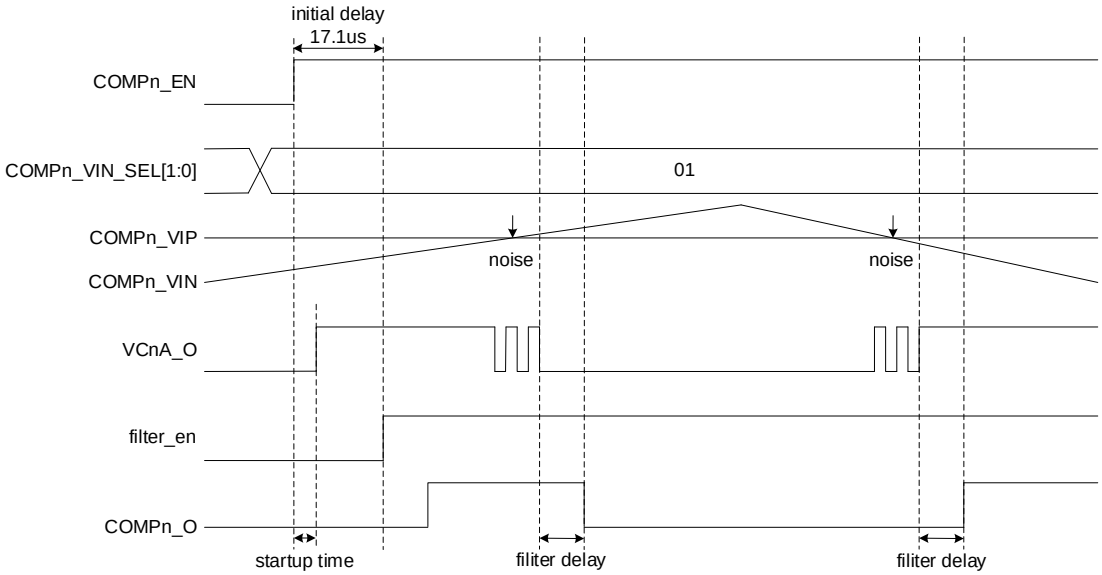


图 18-3 负端输入为 IO 电压的比较器控制时序

(2) 负端输入为 DAC 输出

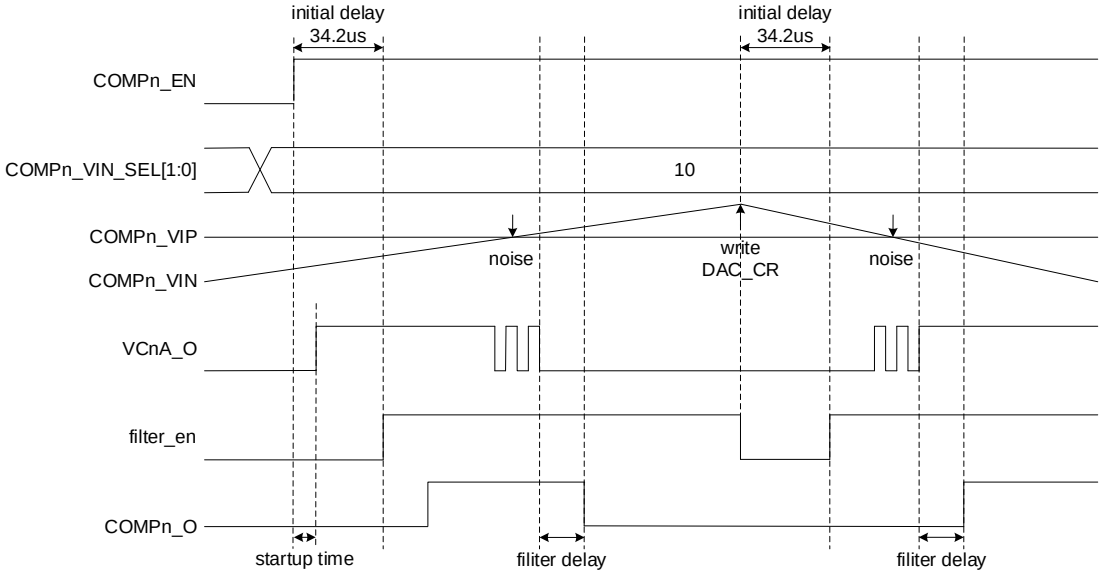


图 18-4 负端输入为 DAC 输出的比较器控制时序

filter_en 作为数字滤波模块的硬件使能控制信号。负责在初始化阶段屏蔽数字滤波功能，为硬件自动控制。当 filter_en 信号为低电平时，数字滤波器不会处理比较器的输出，并保持其先前的数字输出状态。该状态保持机制可避免在初始化延迟期间产生非预期的中断触发。

比较器的输出极性可通过配置 COMP_CTRL 寄存器中的 POL_SEL 位进行修改。当 POL_SEL 位设为 1 时，比较器输出极性将被反相；否则，保持原始输出极性。

18.3.2 数字滤波

比较器输出在正端和负端输入信号交越时容易产生毛刺。为了消除在信号交越过程中产生的不确定状态，比较器结果会被送入数字滤波模块。

用户可以通过 COMP_CTRL 寄存器中的 FIL_CTRL 位（请确认是否为多位）配置滤波采样周期数。该配置可以滤除不同宽度的毛刺，最大可滤波毛刺宽度为 256 个 PCLK 时钟周期。详细配置请参阅 COMP_CTRL 寄存器中 FIL_CTRL 位的描述。

数字滤波的原理如下: 在一个采样周期内，比较器输出会被按照 FIL_CTRL 设置的次数进行采样。如果所有连续采样点具有相同的逻辑电平，则滤波器输出当前电平；否则，滤波器输出保持其先前状态。

为了简化处理，滤波器输出状态分为两种: 高电平和低电平（默认低电平）。

当滤波器输出状态为低电平时，对所有采样点进行逻辑 AND 运算。如果结果为 1（表示所有采样点均为高电平），滤波器输出翻转，输出状态变为高电平。如果结果为 0，输出保持低电平。

当滤波器输出状态为高电平时，对所有采样点进行逻辑 OR 运算。如果结果为 0（表示所有采样点均为低电平），滤波器输出翻转，输出状态变为低电平。如果结果为 1，输出保持高电平。

数字滤波电路的滤波流程示意如下:

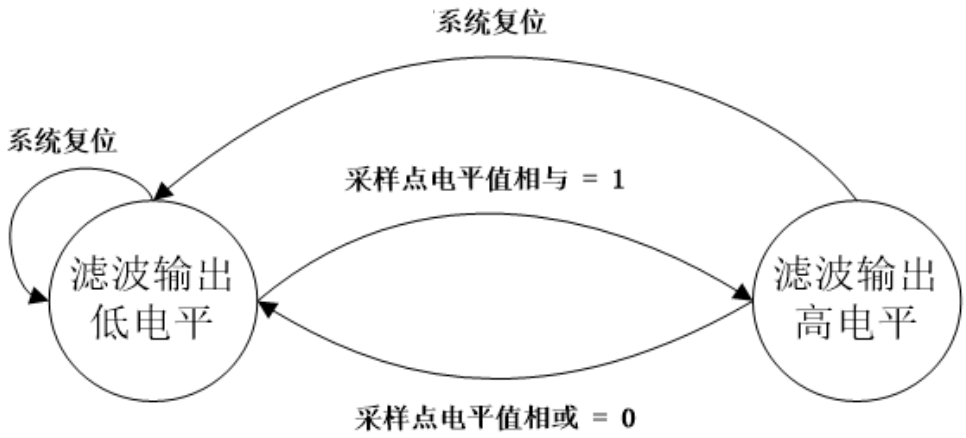


图 18-5 数字滤波电路的滤波流程

FIL_CTRL 位的默认值为不进行滤波采样。由于数字滤波会引入延迟，设置较大的 FIL_CTRL 值可能降低比较器的响应速度，导致无法及时响应快速变化的输入信号。因此，用户必须根据具体应用场景适当配置 FIL_CTRL 值。

18.3.3 中断生成

比较器输出经过数字滤波处理后，可根据应用需求生成相应的中断。

提供三种基于信号边沿变化的中断触发模式：上升沿触发、下降沿触发，以及边沿触发（在上升沿和下降沿均触发）。用户可通过 COMP_IR 中断配置寄存器启用这些中断:

- 将 RIE 位设置为 1，当比较器输出从低电平跳变为高电平（上升沿）时触发中断。
- 将 FIE 位设置为 1，当比较器输出从高电平跳变为低电平（下降沿）时触发中断。
- 将 RIE 和 FIE 位均设置为 1，可在上升沿和下降沿均触发中断。

当在上升沿和下降沿均触发中断时，用户可通过 COMP_IF 中断标志寄存器检查当前触发中断的边沿类型。详细信息请参阅 COMP_IF 寄存器描述。

©Copyright 2026, PADAUK Technology Co. Ltd

18.4 寄存器列表

地址	寄存器	描述
0x4000_8800	COMP0_CTRL	比较器 0 控制寄存器
0x4000_8804	COMP0_VIPSEL	比较器 0 正端輸入选择寄存器
0x4000_8808	COMP0_IR	比较器 0 中断使能寄存器
0x4000_880C	COMP0_IF	比较器 0 中断标志寄存器
0x4000_8810	COMP0_INITCNT	比较器 0 初始化延时配置寄存器
0x4000_9800	COMP1_CTRL	比较器 1 控制寄存器
0x4000_9804	COMP1_VIPSEL	比较器 1 正端輸入选择寄存器
0x4000_9808	COMP1_IR	比较器 1 中断使能寄存器
0x4000_980C	COMP0_IF	比较器 1 中断标志寄存器
0x4000_9810	COMP0_INITCNT	比较器 1 初始化延时配置寄存器

表 18-1 比較器寄存器列表

©Copyright 2026, PADAUK Technology Co. Ltd

18.5 寄存器描述

18.5.1 比较器 n 控制寄存器 (COMPn_CTRL, n = 0, 1)

Bit	Name	R/W	Reset	Description
31:13	-	R	0x0	保留
12	POL_SEL	R/W	0x0	比较器输出极性选择 0: 正常极性 1: 反向极性
11	OUT_EN	R/W	0x0	比较器输出使能 0: 关闭 1: 使能（COMP0: PA7，COMP1: PA13）
10:9	-	R	0x0	保留
8	HYS_EN	R/W	0x0	比较器迟滞使能 0: 关闭 1: 使能
7:4	FIL_CTRL	R/W	0x0	数字滤波采样选择 0x0: 旁路（无滤波） 0x1: 2 次采样 0x2: 4 次采样 0x3: 8 次采样 0x4: 16 次采样 0x5: 32 次采样 0x6: 64 次采样 0x7: 128 次采样 0x8 ~ 0xF: 256 次采样
3:2	VIN_SEL	R/W	0x0	比较器负端输入选择 0x1: IO (COMP0: PA6，COMP1: PA12) 0x2: DAC (COMP0: DAC0, COMP1: DAC1) 其他: 保留
1	COUT	R	0x0	比较器输出结果 0: 低电平 1: 高电平

0	EN	R/W	0x0	比较器模块使能 0: 关闭 1: 使能

18.5.2 比较器 n 正端輸入选择寄存器 (COMPn_VIPSEL, n = 0, 1)

Bit	Name	R/W	Reset	Description
31:6	-	R	0x0	保留
4:0	VIP_SEL	R/W	0x10	比较器正端输入选择 0x10: IO3（PA15） 0x08: IO2（PA14） 0x04: IO1（PA11） 0x02: IO0（PA8） 0x01: PGA (COMP0: PGA0, COMP1: PGA1) 其他: 保留

18.5.3 比较器 n 中断使能寄存器 (COMPn_IR, n = 0, 1)

Bit	Name	R/W	Reset	Description
31:4	-	R	0x0	保留
3:2	-	R/W	0x0	保留
1	RIE	R/W	0x0	比较器输出上升沿中断使能控制 0: 上升沿中断禁止。 1: 上升沿中断使能。
0	FIE	R/W	0x0	比较器输出下降沿中断使能控制 0: 下降沿中断禁止。 1: 下降沿中断使能。

©Copyright 2026, PADAUK Technology Co. Ltd

18.5.4 比较器 n 中断标志寄存器 (COMPn_IF, n = 0, 1)

Bit	Name	R/W	Reset	Description
31:2	-	R	0x0	保留
1	RIF	R/W1c	0x0	比较器输出上升沿中断标志 0: 未发生上升沿中断。 1: 已发生上升沿中断。通过软件向该位写入 1 可清除标志。
0	FIF	R/W1c	0x0	比较器输出下降沿中断标志 0: 未发生下降沿中断。 1: 已发生下降沿中断。通过软件向该位写入 1 可清除标志。

18.5.5 比较器 n 初始化延时配置寄存器 (COMPn_INITCNT, n = 0, 1)

Bit	Name	R/W	Reset	Description
31:10	-	R	0x0	保留
9:0	INIT_DELAY	R/W	0x3C	比較器初始化延时配置。 (默认 60 个系统时钟周期)

19. UART 模块

19.1. UART 模块综述

芯片内集成了 1 个 UART 通讯模块 UART0。下文将叙述 UART0 的相关功能与参数时将其统一描述成 UARTn（n=0~1）。应用系统可以利用 UARTn 模块实现和外界的异步数据通讯（例如 RS232、RS485 等）。

UART 模块主要特性包括:

- 灵活可编程的波特率设定，支持业内标准 9600bps、19200bps、28800bps、38400bps、57600bps、 115.2Kbps 等，或其它特殊应用的波特率
- 可编程的数据传输格式：8 位数据、7 位数据+1 位奇偶校验、8 位数据+1 位奇偶校验、9 位数据
- 可编程奇或偶校验方式
- 可编程设定 0.5 位、1 位、1.5 位或 2 位停止位
- 带 FIFO 缓冲的数据发送: 8 或 9 位数据模式下，缓冲深度 8 级，最大宽度 9 位；
- 带 FIFO 缓冲的数据接收: 8 或 9 位数据模式下，缓冲深度 8 级，最大宽度 9 位；
- 数据收发全双工
- 硬件错误检测

©Copyright 2026, PADAUK Technology Co. Ltd

19.2. UART 基本功能

UART 模块的功能和工作方式通过配置控制寄存器 UART_CR 来实现，该寄存器包含若干用于模式选择和错误检测的控制位，外加一些状态标志位。往发送缓冲寄存器 UART_TXB（物理地址为 UART_DAT，只写）内写入一个数据后，立即启动数据的串行发送过程。UART_TXB 实际上是一个含 8 级 FIFO 缓冲的寄存器组，允许应用软件一次可以连续写入多个发送数据，直至 FIFO 队列满，这样可实现多个数据帧的连续不停顿发送，提高发送效率，降低 CPU 的干预度。

一次数据接收过程完成后，接收到的数据及其附加的校验位信息被存入接收缓冲寄存器 UART_RXB（物理地址为 UART_DAT，只读），并可通过应用软件读出。UART_RXB 实际为 8 级 FIFO 缓冲的寄存器组，无论应用软件是否立即读取刚接收到的数据，UART 模块可继续接收后续的数据并将它们逐个存入接收缓存，直至接收 FIFO 队列满。在接收队列满后如果又有新的数据被收到，则将发生队列溢出的错误，并将状态标志位 OVERR 置 1，此时最前收到的一个数据被最新的数据顶出队列而丢弃，UART_ST 寄存器被相应更新以反映该次新数据的接收。利用硬件的 FIFO 缓冲，应用软件可以在一次从接收队列中连续读取多个数据（只要队列中有数据），提高了 CPU 的运行效率。UART 模块提供片内数据发送端至接收端的回绕功能，无需外部硬件电路和实际线路连接，即可方便地进行软件自诊断。

UART 的数据收发全双工模式使用相同的数据帧结构和传输波特率。发送的数据位信息被送至 TXD 引脚输出，接收的数据位信息则应接入 RXD 引脚．同时需注意:

- 只有当波特率发生器的控制位（RUN）被置 1 后，才能进行串行数据的收发．如果 RUN 位被清 0，则数据收发过程将立即终止，TXD 引脚输出始终保持空闲态（正常极性为高电平 1，反相极性为低电平 0）。除非应用程序确定 UART 通讯处于空闲状态，不然 RUN 位应始终被置为 1。

- 如果将控制模式（UART_CR.MODE）配置成某一个未定义的保留模式，UART 模块的工作将不可预测

- 在 9 位数据模式，或 8 位数据加接收唤醒模式时，必须屏蔽奇偶校验错误中断以防止假的奇偶校验错误．因为在这两种配置模式下数据帧内没有奇偶校验位信息。

©Copyright 2026, PADAUK Technology Co. Ltd

19.3. UART 工作模式

UART 工作模式分 8 位、9 位数据收发模式。

8 位数据帧有两种组成形式:

- 8 位有效数据位（UART_CR.MODE = 0x1）
- 7 位有效数据位加上 1 位奇偶校验位（UART_CR.MODE = 0x3）

9 位数据帧有三种组成形式:

- 9 位有效数据位（UART_CR.MODE = 0x4）
- 8 位有效数据位加上 1 位奇偶校验位（UART_CR.MODE = 0x7）
- 8 位有效数据位加上 1 位唤醒位（UART_CR.MODE = 0x5）

发送校验位的产生可以是奇校验或偶校验，取决于 UART_CR.PAR 位的设定。设为奇校验时，如果有效数据位中的 1 为偶数个，则校验位被置 1；在偶校验时此校验位则被置 0。接收时，校验位和有效数据位一起被存入接收缓冲 UART_RXB 队列中。

在 UART 处于接收唤醒模式时，只有当 RXD 引脚送入的数据帧其第 9 位为 1 时，接收结果才会被存入 UART_RXB 缓冲队列并产生接收中断请求；如果此第 9 位为 0，则无任何中断请求，也不会往 UART_RXB 存入任何数据。这一特性非常适用于主-从结构的多机通讯应用。当主机需要发送一串数据至某一指定地址的从机时，它先发送 8 位有效地址外加第 9 位为 1 的一个 9 位寻址帧，所有从机都能收到这个特殊的寻址帧（因第 9 位为 1）并和本机设定地址作比对，如果地址相符，则立即将本机的 UART 工作模式转为 9 位数据模式，继续接收后面的数据帧（数据帧第 9 位为 0），地址不符的从机则不会响应后续的数据帧，避免过多无谓的中断响应，提高系统运行效率。

软件若在 UART 工作时需要切换模式，需在 UART 为 idle 时切换，并在每次切换完模式之后主动写一次发送队列复位寄存器和接收队列复位寄存器以清空发送队列和接收队列，保证 UART 在新的模式下能够正常收发数据。

©Copyright 2026, PADAUK Technology Co. Ltd

19.4. UART 波特率计算和设定

UART 模块含自动重载的波特率发生器，由 RUN 位控制其工作或停止，定时控制寄存器 UART_BR 分为两个区域: BR 区域的宽度为 16 位，作为系统时钟分频; SR 区域为 2 位宽，用于控制每位码元的采样率．软件通过设置 UART_BR 的值设定 UART 通讯的波特率，具体计算公式如下:

$BPS_{UART} = f_{SYS} / (16 / (2^{SR}) * BR)$ 或

$BR = f_{SYS} / (16 / (2^{SR}) * BPS_{UART})$

其中:

- BPS_{UART} 为期望的通讯波特率，同时用于接收和发送
- f_{SYS} 为系统时钟频率
- BR 为寄存器 UART_BR 内的 BR 域设定值
- SR 为寄存器 UART_BR 内的 SR 域设定值

以下列出了在不同系统时钟频率下常用的波特率设定值和误差（UART_BR.SR = 0x0），可供应用时快速参考。

	f_{CLK} = 64MHz		f_{CLK} = 60MHz		f_{CLK} = 56MHz	
波特率	UART_BR	误差	UART_BR	误差	UART_BR	误差
115200	0x0022	-0.21%	0x0020	0.04%	0x001E	0.10%
76800	0x0034	-0.16%	0x0030	0.06%	0x002D	0.12%
57600	0x0045	-0.64%	0x0041	0.05%	0x003C	0.11%
38400	0x0068	-0.16%	0x0061	0.06%	0x005B	0.12%
19200	0x00D0	-0.16%	0x00C3	0.06%	0x00B6	0.12%
9600	0x01A0	-0.16%	0x0186	0.06%	0x016C	0.12%
4800	0x0341	-0.04%	0x030D	0.06%	0x02D9	0.12%
200	0x4E20	0%	0x493E	0.06%	0x445C	0.12%

表 19-1 常用的波特率设定系数列表

如果需要设定特殊波特率，可按上述公式计算得到 UART_BR 的设定值并确认实际波特率误差不超过 1%，不然数据收发将不能可靠进行。有时为了得到较高的波特率精度，可以考虑使用特殊频率的系统时钟。

©Copyright 2026, PADAUK Technology Co. Ltd

19.5. UART 数据发送

在 RUN 位置 1 且有数据写入 UART_TXB （亦即 UART_DAT）寄存器后，即启动数据帧的串行发送过程。发送的数据帧包含 3 个部分:

- 1 个起始位
- 若干数据位（8 或 9 位），低位先发
- 可选的 1 位校验位
- 停止位（0.5、1、1.5 或 2 位）

发送队列有 8 级的 FIFO 缓存，在先前写入的数据正忙于移位发送时，软件可以继续往 UART_TXB 队列中写入发送数据，直至队列满．当最后一个数据帧的最后 1 位被实际发送出去后，发送器空闲状态标志 TXE 被置 1，告知所有数据的串行发送已完成。

应用软件可以设定串行数据发送的极性，普通极性时，无数据发送时 TXD 引脚保持高电平，发送数据时，起始位为低电平，停止位为高电平，数据位表达 0 为低电平，1 为高电平；反相极性时，所有电平和普通极性相反。

提供反相发送模式可以方便不同电压系统间通讯接口的设计，亦可方便实现开漏驱动型通讯。

数据发送通常会经过以下几步,以只检测 TXE 为例:

1. 设定工作模式 (UART_CR.MODE) ,波特率 (UART_BR), 奇偶校验形式 (UART_CR.PAR), 停止位长度 (UART_CR.STOPB), 极性(UART_CR.TXPOL)
2. 使能波特率发生器 (UART_CR.RUN)
3. 写入 FIFO 缓冲器穿送数据 (UART_DAT)
4. 确认 UART_ST.TXE 为 0, 使能中断 (UART_IE.TXEE)
5. 等待 UART_ST.TXE 触发中断,表示数据以传输完毕,中断中关闭使能 (UART_IE.TXEE)

19.6. UART 数据接收

在 RUN 位置 1，RXEN 置 1 时，UART 模块准备接收串行数据．RXD 引脚上检测到起始位开始时（正相数据的下降沿），即引发一次数据帧的接收过程，UART 模块按既设波特率对每位数据进行 16 次采样并通过多数表决的方式判断数据位 0 或 1，避免偶发干扰造成接收数据误判。

当确认接收到最后一位停止位后，UART 模块将移位收到的数据转送入接收缓冲队列 UART_RXB （亦即 UART_DAT），置队列非空标志 RXNE 为 1，若此时接收队列已满，则置队列满标志 RXF 为 1，但不会更新校验错误标志 PERR 和帧错误标志 FERR （接收数据校验标志将在数据被从接收队列中读出时更新）。然后模块准备接收新数据，等待下一个起始位的出现。

如果在数据帧接收过程中软件清除 RXEN 位，当前正在接收的数据帧依然可以被完整的接收，并更新各状态标志。但后续的数据将不能被接收到。在接收唤醒模式下，只有第 9 位为 1 的数据帧才能被接收存入接收缓冲队列并更新各状态标志，第 9 位为 0 的数据帧被丢弃。

数据接收通常会经过以下几步,以只检测 RXNE 为例:

1. 设定工作模式 (UART_CR.MODE), 波特率 (UART_BR), 奇偶校验形式 (UART_CR.PAR), 停止位长度 (UART_CR.STOPB), 极性(UART_CR.RXPOL)

2. 使能波特率发生器 (UART_CR.RUN), 超时控制 (UART_TO)

3. 使能中断 (UART_IE. RXNEE)

4. 等待接收数据, UART_ST.RXNE 置 1 触发中断, 表示已接收数据, 中断中关闭使能 (UART_IE. RXNEE)

©Copyright 2026, PADAUK Technology Co. Ltd

19.7. UART 错误检测

为确保串行数据通讯的可靠性，UART 模块提供了必要的通讯错误检测机制，并通过状态寄存器 UART_ST 内若干状态位反映不同的错误信息:

- 当 UART_RXB 队列接收数据溢出时, UART_ST.OVERR 被立即置 1。
- 当应用软件从接收队列中读取暂存的接收数据时，每读取一个数据后，其对应的校验信息将同步反映在 UART_ST 状态寄存器中: 若发生奇偶校验错误，UART_ST.PERR 位被置 1；若出现帧接收错误（在停止位处没有检测到 1），UART_ST.FERR 被置 1。

所有的错误标志位，如果对应的中断使能位置 1，在相“或”后可触发 UART 模块的错误中断，软件必须分别查询判断并应对处理。

19.8. UART 中断响应

UART 模块内有两组寄存器，状态寄存器 UART_ST 和中断控制寄存器 UART_IE，用于控制数据通讯的中断响应。状态寄存器内的相关标志位反映发生的是何种中断，中断控制寄存器内的对应位决定了该中断是否被允许响应。多个被允许的中断标志信息通过逻辑“或”后，产生一个统一的串行通讯中断请求送入内核中断控制器，请求系统响应中断服务。

中断标志的清除方式按不同的标志而不同．发送相关的中断标志 TXE 和 TXHE 视写入发送缓冲队列数据的多少由硬件自动清 0；接收相关的中断标志 RXF 在软件从接收缓冲队列内读取一个数据且队列未满时被硬件自动清 0，RXNE 则必须等到软件读完接收缓冲队列内的全部数据后（接收缓冲队列为空）由硬件自动清 0；错误中断标志 PERR、FERR 和 OVERR 则只能通过由软件对中断状态寄存器 UART_ST 的对应位写 1 来清 0。

针对一般通讯过程的中断标志产生条件及含义可归纳如下。数据发送和接收有三个可用的中断标志。

数据发送标志:

- TXE: 当队列中的最后一个数据被转入移位发送器后，队列被清空，该标志位被置 1，但有可能数据的发送移位正在进行过程中。空闲状态下没有数据发送时，队列一定是全空的，故软件使能该中断后，将立即进入中断服务程序。

- TXHE: 当队列中的一个数据被转入移位发送器，且对列可用空间超过总长度的一半或以上，该标志位被置 1。此时发送缓冲队列为半空。

- TXEND: 当最后一个数据的最后一位被实际发送完毕后，该标志位被置 1。此时发送缓冲队列一定为全空。

©Copyright 2026, PADAUK Technology Co. Ltd

数据接收标志:

- RXNE: 当接收到的一个数据从接收移位器转入接收缓冲队列 UART_RXB 后，该标志位被置 1，指示接收缓冲队列为非空（队列中至少有一个数据）。
- RXHF: 当接收到的一个数据从接收移位器转入接收缓冲队列 UART_RXB，且队列超过半满时，该标志位被置 1，指示接收缓冲队列已被使用超过一半以上，但尚有空间。
- RXF: 当接收到的一个数据从接收移位器转入接收缓冲队列 UART_RXB，且队列全满时，该标志位被置 1，指示接收缓冲队列已经全满。如果软件没有及时将队列中的数据读出，下一个接收的数据即会造成接收队列溢出错误。

正反相接收的上述两组标志位，如果对应的中断使能位置 1，在相“或”后触发接收中断，软件必须分别查询判断并应对处理。错误标志的产生见 14.7 节说明。

内核系统为每个 UART 模块提供了一个独立的中断请求源，分别用于响应数据发送、数据接收和错误处理，归类如下:

- 发送中断：若 UART_ST 状态寄存器中的 TXEND、TXHE 或 TXE 位中任意位为 1 且 UART_IE 中断控制寄存器中的对应位也为 1，则 UART 模块向系统发出发送中断请求。在中断服务程序中，应用软件必须同时相关判别标志位和控制位，辨析是由哪几个位产生的中断。
- 接收中断: 若 UART_ST 状态寄存器中的 RXF、RXHF 或 RXNE 位中任意位为 1 且 UART_IE 中断控制
- 寄存器中的对应位也为 1，则 UART 模块向系统发出接收中断请求。在中断服务程序中，应用软件必须同时相关判别标志位和控制位，辨析是由哪几个位产生的中断。
- 错误中断: 若 UART_ST 状态寄存器中的 TOIDLE、TONE、OVERR、FERR 或 PERR 位中任意位为 1 且 UART_IE 中断控制寄存器中的对应位也为 1，则 UART 模块向系统发出错误中断请求。在中断服务程序中，应用软件必须同时相关判别标志位和控制位，辨析是由哪几个位产生的中断。

19.9. 寄存器列表

地址偏移	寄存器	描述
0x4000_2000	UART_DAT	UART 收发数据寄存器
0x4000_2004	UART_CR	UART 模块控制寄存器
0x4000_2008	UART_BR	UART 波特率控制寄存器
0x4000_200C	UART_IE	UART 接收中断使能控制寄存器
0x4000_2010	UART_ST	UART 接收状态寄存器
0x4000_2014	UART_GT	UART 帧间隔时间寄存器
0x4000_2018	UART_TO	UART 超时控制寄存器
0x4000_201C	UART_TXFR	UART 发送队列复位寄存器
0x4000_2020	UART_RXFR	UART 接收队列复位寄存器

表 19-2 UART 寄存器列表

©Copyright 2026, PADAUK Technology Co. Ltd

19.10. 寄存器描述

19.10.1 收发数据 FIFO 缓冲寄存器 (UART_DAT)

UART_DAT 是接收和发送数据 FIFO 缓冲队列，深度为 8 级。

9 位数据帧格式下写入数据低 9 位有效，内容定义如下表。

Bit	Name	R/W	Reset	Description
31:9	-	R	0x0	保留
8	TB8	W	0x0	第 8 位（TB8）可以是数据位，或奇偶校验位，或唤醒位，或固定为 0，这同样取决于控制寄存器中的 MODE 设定
7	TB7	W	0x0	TB7 可以是数据位，也可以是奇偶校验位，这取决于控制寄存器中的 MODE 设定
6:0	TD6-TD0	W	0x0	TD6 - TD0 发送的 7 位数据

数据接收时，UART_DAT 队列内为同相接收的数据。

9 位数据帧格式下读出数据低 10 位有效。内容定义如下表。

Bit	Name	R/W	Reset	Description
31:10	-	R	0x0	保留
9	FE	R	0x0	FE（第 9 位）则为此数据帧的错误标志位。只有在 9 位帧模式下 FE 位域才可读到帧错误标志位
8	RB8	R	0x0	RB8（第 8 位）可以是数据位，或奇偶校验位，或唤醒位，或固定为 0（当数据加校验的长度为 8 位），同样取决于控制寄存器中的 MODE 设定
7	RB7	R	0x0	RB7（第 7 位）可以是数据位，也可以是奇偶校验位，这取决于控制寄存器中的 MODE 设定

©Copyright 2026, PADAUK Technology Co. Ltd

6:0	RD6-RD0	R	0x0	RD6-RD0（第 6 至第 0 位）为接收的 7 位数据

19.10.2 模块控制寄存器 (UART_CR)

Bit	Name	R/W	Reset	Description
31:18	-	R	0x0	保留
17	TXPOL	R/W	0x0	UART 数据发送极性控制 0: 标准数据极性 （空闲为高电平，起始位为低电平；数据 1=高电平，0=低电平） 1: 反相数据极性 （空闲为低电平，起始位为高电平；数据 1=低电平，0=高电平）
16	RXPOL	R/W	0x0	UART 数据接收极性控制 0: 标准数据极性 （空闲为高电平，起始位为低电平；数据 1=高电平，0=低电平） 1: 反相数据极性 （空闲为低电平，起始位为高电平；数据 1=低电平，0=高电平）
15:9	-	R	0x0	保留
8	RXEN	R/W	0x0	数据接收使能 0: 禁止数据接收 1: 允许数据接收
7	RUN	R/W	0x0	波特率发生器控制 控制波特率发生器的工作或停止。当波特率发生器被禁止时，TXD 引脚将维持空闲状态，数据接收则被冻结。 0: 波特率发生器被禁止 1: 波特率发生器正常工作

6	LPB	R/W	0x0	回绕模式控制，控制模块内收发回绕 0: 正常数据接收和发送模式 1: 内部自发自收模式（内部数据回绕）
5	PAR	R/W	0x0	奇偶校验形式 设定奇偶数据位的校验形式 0: 偶校验（有效数据位 1 的个数为奇数时校验位设为 1，保持总的 1 的个数为偶数） 1: 奇校验（有效数据位 1 的个数为偶数时校验位设为 1，保持总的 1 的个数为奇数）
4:3	STOPB	R/W	0x0	停止位长度 设定数据帧停止位的长度 00: 0.5 位停止位 01: 1 位停止位 10: 1.5 位停止位 11: 2 位停止位
2:0	MODE	R/W	0x0	UART 工作模式 设定 UART 模块的工作模式 000: 保留．用户禁用 001: 8 位数据格式 010: 保留．用户禁用 011: 7 位数据 + 1 位奇偶校验 100: 9 位数据 101: 8 位数据 + 1 位唤醒 110: 保留．用户禁用 111: 8 位数据 + 1 位奇偶校验

19.10.3 波特率控制寄存器 (UART_BR)

Bit	Name	R/W	Reset	Description
31:18	-	R	0x0	保留
15:0	BR	R/W	0x0001	波特率分频系数 该值的设定使用方法请参考 “UART 波特率计算和设定章节”

19.10.4 中断控制寄存器 (UART_IE)

Bit	Name	R/W	Reset	Description
31:12	-	R	0x0	保留
11	TXPE	R/W	0x0	发送缓冲队列全满中断允许控制位 0: 禁止中断 1: 允许中断
10	TXEND	R/W	0x0	发送全部完成中断允许控制位 0: 禁止中断 1: 允许中断 由于对应旗标复位值为 1，应在数据开始传送后再使能,并在中断中关闭使能
9	RXFE	R/W	0x0	接收缓冲队列全满中断允许控制位 0: 禁止中断 1: 允许中断
8	RXHFE	R/W	0x0	接收缓冲队列半满中断允许控制位 0: 禁止中断 1: 允许中断
7	TOIDLEE	R/W	0x0	空闲超时中断允许控制位 0: 禁止中断 1: 允许中断 由于对应旗标复位值为 1，应在数据开始传送后再使能，并在中断中关闭使能

6	TONEE	R/W	0x0	接收缓冲队列清空超时中断允许控制位 0: 禁止中断 1: 允许中断
5	OVERRE	R/W	0x0	接收缓冲队列溢出中断允许控制位 0: 禁止中断 1: 允许中断
4	FERRE	R/W	0x0	帧错误中断允许控制位 0: 禁止中断 1: 允许中断
3	PERRE	R/W	0x0	奇偶校验错误中断允许控制位 0: 禁止中断 1: 允许中断
2	TXHEE	R/W	0x0	发送缓冲队列半空中断允许控制位 0: 禁止中断 1: 允许中断 由于对应旗标复位值为 1,应在数据开始传送后再使能,并在中断中关闭使能
1	TXEE	R/W	0x0	发送缓冲队列全空中断允许控制位 0: 禁止中断 1: 允许中断 由于对应旗标复位值为 1,应在数据开始传送后再使能,并在中断中关闭使能
0	RXNEE	R/W	0x0	接收缓冲队列非空中断允许控制位 0: 禁止中断 1: 允许中断

©Copyright 2026, PADAUK Technology Co. Ltd

19.10.5 状态寄存器 (UART_ST)

Bit	Name	R/W	Reset	Description
31:12	-	R	0x0	保留
11	TXF	R	0x0	发送缓冲队列全满中断标志位 0: 发送缓冲队列中数据尚未全满 1: 发送缓冲队列中数据已经全满 该标志位由硬件设定，软件无法改写
10	TXEND	R	0x1	发送全部完成中断标志位 0: 数据发送正在进行 1: 数据发送全部结束 该标志位由硬件设定，软件无法改写
9	RXF	R	0x0	接收缓冲队列全满中断标志位 0: 接收缓冲队列中数据尚未全满 1: 接收缓冲队列中数据已经全满 该标志位由硬件设定，软件无法改写
8	RXHF	R	0x0	接收缓冲队列半满中断标志位 0: 接收缓冲队列中的数据少于或等于队列长度的一半 1: 接收缓冲队列中的数据大于队列长度的一半 该标志位由硬件设定，软件无法改写
7	TOIDLE	R	0x1	空闲超时中断标志位 当 UART_TO 寄存器设定的超时时间已到但接收缓冲队列依然为全空时，该位被置 1. 0: 没有超时 1: 接收缓冲队列全空超时 该标志位由硬件设定，软件无法改写
6	TONE	R	0x0	接收缓冲队列清空超时中断标志位 当 UART_TO 寄存器设定的超时时间已到但接收缓冲队列尚未被清空时，该位被置 1. 0: 没有超时 1: 接收缓冲队列未清空超时 该标志位由硬件设定，软件无法改写

5	OVERR	R/W1c	0x0	接收缓冲队列溢出中断标志位 0: 接收缓冲队列无溢出 1: 接收缓冲队列发生溢出 该标志位必须由软件对 OVERR 位写 1 才能将其清除
4	FERR	R/W1c	0x0	帧错误中断标志位 0: 最近一次接收无帧错误 1: 最近一次接收有帧错误 该标志位必须由软件对 FERR 位写 1 才能将其清除
3	PERR	R/W1c	0x0	奇偶校验错误中断标志位 0: 最近一次接收无奇偶校验错误 1: 最近一次接收检有奇偶校验错误 该标志位必须由软件对 PERR 位写 1 才能将其清除
2	TXHE	R	0x1	发送缓冲队列半空中断标志位 0: 发送缓冲队列中的数据超过队列长度的一半 1: 发送缓冲队列中的数据少于或等于队列长度的一半 该标志位由硬件设定，软件无法改写
1	TXE	R	0x1	发送缓冲队列全空中断标志位 0: 发送缓冲队列中至少有一个数据 1: 发送缓冲队列为空，队列中没有任何数据 该标志位由硬件设定，软件无法改写
0	RXNE	R	0x0	接收缓冲队列非空中断标志位 0: 接收缓冲队列为空，队列中没有任何数据 1: 接收缓冲队列中至少有一个数据 该标志位由硬件设定，软件无法改写

19.10.6 帧间隔时间寄存器 (UART_GT)

Bit	Name	R/W	Reset	Description
31:8	-	R	0x0	保留
7:0	UART_GT	R/W	0x0	模块在发送缓冲队列为空时重新装载此寄存器设定的间隔时间值。

注: UART_GT 寄存器定义了发送两个连续数据帧的间隔时间，以位元的时间宽度为单位．此寄存器仅低 8 位有效.

©Copyright 2026, PADAUK Technology Co. Ltd

19.10.7 超时控制寄存器 (UART_TO)

Bit	Name	R/W	Reset	Description
31:8	-	R	0x0	保留
7:0	UART_TO	R/W	0x0	当设定的超时时间已到但接收缓冲器仍然为非空时，UART_ST.TONE 位被置 1； 当设定的超时时间已到但接收缓冲器仍然为全空 UART_ST.TOIDLE 位被置 1。 如果 UARTn 开始接收数据，UART_ST.TONE 位和 UART_ST.TOIDLE 位的值将被复位成 0，直到设定的超时时间到了才会被更新。 超时时间的计时在 UARTn 被使能且接收为 idle 的时候进行计时，每次读取 UART_DATm 寄存器或重新写 UART_TO 寄存器会重新计时。 如果 UARTn 未被使能，UART_ST.TONE 位和 UART_ST.TOIDLE 位的值将根据接收缓冲器的状态即时更新。

注: UART_TO 寄存器控制数据接收过程的超时判断，超时时间设定以一个位元的时间宽度为最小单位。

19.10.8 发送队列复位寄存器 (UART_TXFR)

Bit	Name	R/W	Reset	Description
31:0	UART_TXFR	W	-	该寄存器位 [31:0] 进行一此写入操作后，不管写入什么数值， UART_DAT 发送缓冲队列即被复位清空。

19.10.9 接收队列复位寄存器 (UART_RXFR)

Bit	Name	R/W	Reset	Description
31:0	UART_RXFR	W	-	该寄存器位 [31:0] 进行一次写入操作后，不管写入什么数值， UART_DAT 接收缓冲队列即被同时复位清空。

20. SPI

20.1. 概述

SPI 串行通讯标准由 Motorola 公司建立，它基于 3 线连接方式实现单片机和外围器件，或者单片机之间的数据通讯。通过从器件的通讯选择控制（片选），可以构成主从方式的 SPI 总线。SPI 通讯所用引脚描述如下:

- SCK: SPI 串行通讯时钟，时钟信号只有 SPI 主机才能发出
- MISO: SPI 串行通讯数据输入。该数据为从机发出，在主机侧为输入。
- MOSI: SPI 串行通讯数据输出。该数据为主机发出，在从机侧为输入。

SPI 通讯是全双工的，发送或接收数据都是高位在先，在数据发送的同时进行数据接收。按通讯时钟的不同提供方式，SPI 通讯分主机和从机两种。SPI 主机的时钟由本地产生，通讯的发起和结束完全由自己主动控制；SPI 从机的时钟为外部输入（来自 SPI 主机），被动响应主机的通讯。一个典型的 SPI 通讯系统构成如图 20-1.

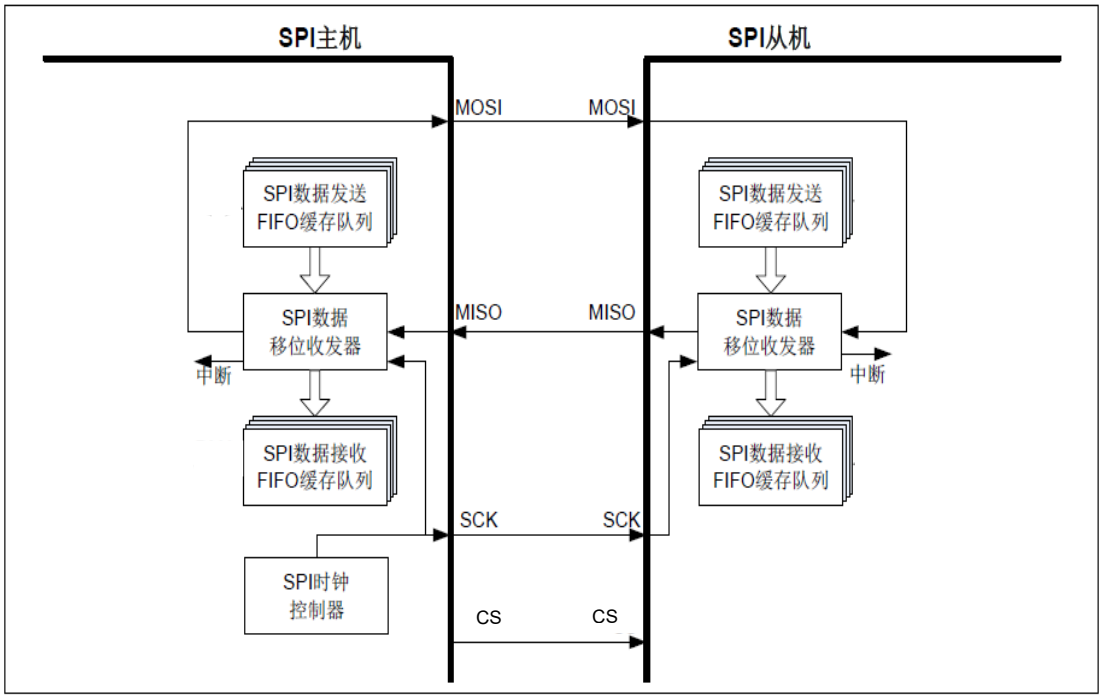


图 20-1 SPI signal interface

PEC930 的 SPI 模块支持主从模式工作．基本特性如下:

- 传输字大小 2bit 可配，支持 8 / 16 / 24 / 32bits datasize
- 模块支持 LSB 和 MSB 两种发送数据配置
- 主机最高传输速率为 10Mbps（系统时钟为 60MHz 时）
- 主机模式下支持 4 路从机片选输出
- 最大 8 级接收 FIFO 缓冲队列，32 位宽
- 最大 8 级发送 FIFO 缓冲队列，32 位宽
- 内部时钟分频后实现主机可编程 SPI 传输速率
- 1 个独立高速 SPI 模块
- 可编程的中断产生

©Copyright 2026, PADAUK Technology Co. Ltd

20.2. SPI 工作模式

20.2.1 主端工作模式

SPI 只能在主模式下发起传输。在主模式下, SPI 控制与 SPI 连接的任何从端之间的数据传输。主端向从端发送消息,从端会立即发送回复。可以配置 SPI_CR 的 bit 10 驱动 CS 引脚的输出从而选择从端进行通信。

当 SPI_ENABLE 寄存器中的 bit0 SPI_EN 位和 SPI_CR 存器位中的 bit0 MODE 位都设置为高时, SPI 将进入主模式。除了这种对外设的自动片选外,还可通过 SPI_CR 寄存器的 MCS 位启用手动外设片选模式。

20.2.2 从端工作模式

在从模式下，SPI 模块接收来自外部 SPI 主端的数据并可同时回传数据。在此模式下，从端时钟输入来自主端的输出，此时钟的极性与相位可通过寄存器配置，详细配置方法见时钟配置部分。当 SPI_CR 寄存器的 bit0 MODE 位为低并且 SPI_ENABLE 寄存器中的 bit0 SPI_EN 位设置为高时，SPI 将工作在从模式。

当 SPI 模块工作在从端模式下如果 CS 引脚为高 (非活动状态)，则主端与此从端的通信未被启动。当 CS 拉低后其必须在访问期间保持拉低状态（活动状态）。如果在传输数据的过程中此信号拉高则 SPI_SR 寄存器的 bit1 MDF (modefail) 位将被拉高，以指示系统出现 modefail，同时当前的传输行为将被中止并禁止更新 FIFO 中的数据。当再次使能 SPI 时（配置 SPI_ENABLE 寄存器的 bit0 SPI_EN 为 1），由 modefail 而导致中断的字传输将被恢复。

为了让 SPI 模块使能后能让从端与外部主端同步，在从端模式下模块可检测 SPI 传输字的边界。当检测到以下任意一种情况时则同步可以实现:

1. 如果 CS 端口为高电平(非活动)，则从端控制器将会把在 CS 信号拉低之后 SCK 的第一个 activeedge 作为一个传输字的开头同时 SPI 的主从端将从此刻开始同步，直到重新设置 SPI_ENABLE 寄存器。
2. 如果 SPI 从端被使能，从端控制器将开始计数 SCK 在非活动状态的时间（以 SPI_REF_CLOCK 的周期数计数），当此计数值与 IDEL COUNT 寄存器中设定的值匹配时从端控制器将认为 SPI 的主端与从端已实现同步。

注: SPI_REF_CLOCK 为 pclk。

©Copyright 2026, PADAUK Technology Co. Ltd

20.2.3 时钟配置

SPI 的时钟相位（CPHA）与时钟极性（CPOL）可配置，其中时钟相位配置可通过写 SPI_CR 寄存器 的 bit 2 CPHA 实现，时钟极性的配置可通过写 SPI_CR 寄存器的 bit3 CPOL 实现。通过在主端模式下配置时钟相位（CPHA），可产生如下影响:

- 当 CPHA = 0, 在字与字的传输之间，主端 SPI 自动将从端片选信号置为无效状态并保持至少一个 SPI_REF_CLOCK 时钟周期，延迟的长度可通过 SPI_DELAY 配置。在此情况下，当前字的最后 1 bit 与下一字的第一个 bit 间将保持至少 3 个 SPI_REF_CLOCK（或外部）时钟周期的延迟，延迟的时间长度可通过 SPI_DELAY 寄存器配置。这种机制可以帮助 TX FIFO 在下一次并行串行转换前卸载其中的数据。

- 当 CPHA = 1, 在字与字传输之间片选信号将不再被置为无效。在此模式下，当前传输字的最后 1 bit 与下一字的第一 bit 间的最小延迟为 1 个 SPI_REF_CLOCK（或外部）时钟周期，其延迟时间也可通过 SPI_DELAY 寄存器配置。同样的，这个延迟的存在也是帮助 TX FIFO 卸载数据以准备好下一次的并行串行数据转换。

不同的时钟极性与相位的配置组合可产生多种不同的数据传输格式，取决于 SPI 模块控制寄存器中的 CPOL（时钟极性）位和 CPHA（时钟相位）位设定，SPI 共有 4 中不同的工作模式，分别为模式 00、模式 01、模式 10 和模式 11。

©Copyright 2026, PADAUK Technology Co. Ltd

20.2.3.1. SPI 模式 00

当 SPI 设为模式 00 时，在空闲状态下 SPI 时钟保持低电平。主机软件把从机的片选信号 CS 拉低后，从机发往主机的第一位数据即出现在主机的 MISO 引脚上就绪；主机往 SPI 数据发送寄存器内写入数据后， 发往从机的第一位数据也即刻出现在 MOSI 引脚上．之后在每一个时钟上升沿主机和从机各自同步采样接 收对方发送过来的数据位并送入内部移位寄存器，在其后的时钟下降沿各自打出下一位数据。当一个 SPI 数据帧（4~32 位）传输结束后，接收到的数据从内部移位寄存器转存入 SPI 数据接收寄存器，SPI 时钟恢复到低电平状态。主机可继续往发送寄存器内写入数据进行连续发送，或者通过软件把从机片选信号拉高，结束 SPI 通讯。

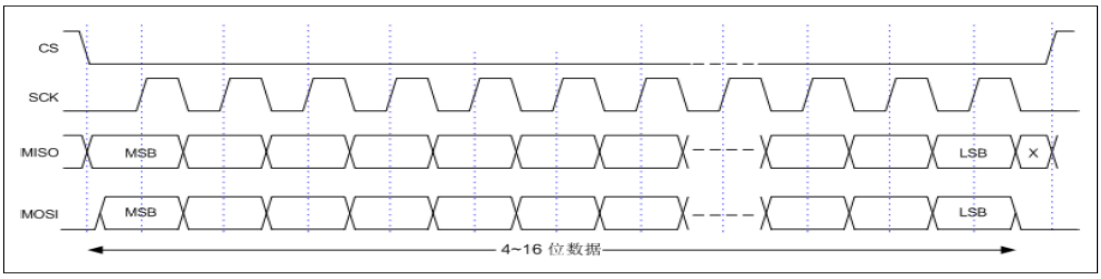


图 20-2 SPI 模式 00 时序图（CPOL = 0, CPHA = 0）

20.2.3.2. SPI 模式 01

当 SPI 设为模式 01 时，在空闲状态下 SPI 时钟保持低电平。主机通过软件将从机的片选信号 CS 拉低， 往 SPI 数据发送寄存器内写入发送数据后，即启动一次 SPI 数据帧的收发过程。在每一个时钟上升沿主机和从机各自将数据打出就绪，然后在其后的时钟下降沿采样接收对方发过来的数据位并送入内部移位寄存器。当一个 SPI 数据帧（4~32 位）传输结束后，接收到的数据从内部移位寄存器转存入 SPI 数据接收寄存器，SPI 时钟恢复到低电平状态。主机可继续往发送寄存器内写入数据进行连续发送，或者通过软件把从机片选信号拉高，结束 SPI 通讯。

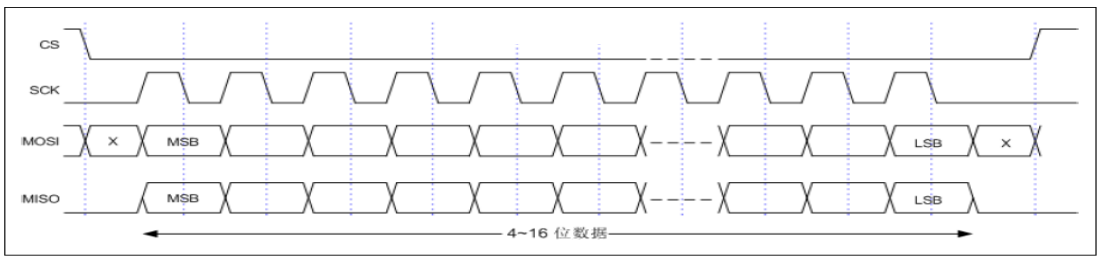


图 20-3 SPI 模式 01 时序图（CPOL = 0, CPHA = 1）

20.2.3.3. SPI 模式 10

当 SPI 设为模式 10 时，在空闲状态下 SPI 时钟保持高电平。主机软件把从机的片选信号 CS 拉低后，从机发往主机的第一位数据即出现在主机的 MISO 引脚上就绪；主机往 SPI 数据发送寄存器内写入数据后，发往从机的第一位数据也即刻出现在 MOSI 引脚上。之后在每一个时钟下降沿主机和从机各自同步采样接收对方发送过来的数据位并送入内部移位寄存器，在其后的时钟上升沿各自打出下一位数据。当一个 SPI 数据帧（4~32 位）传输结束后，接收到的数据从内部移位寄存器转存入 SPI 数据接收寄存器，SPI 时钟恢复到高电平状态。主机可继续往发送寄存器内写入数据进行连续发送，或者通过软件把从机片选信号拉高，结束 SPI 通讯。

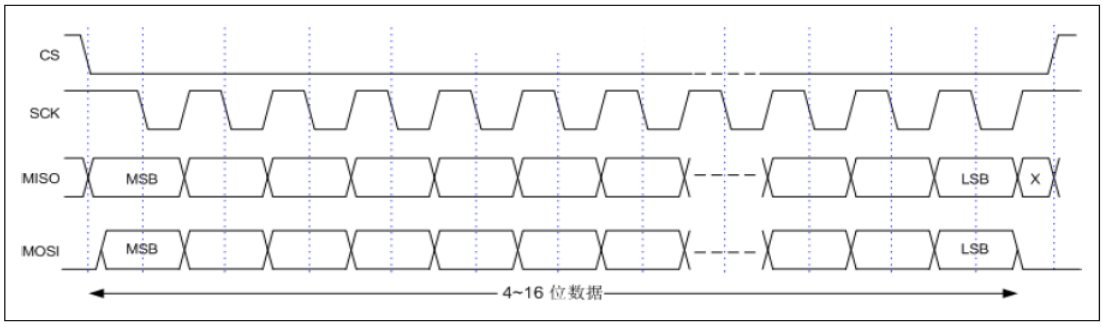


图 20-4 SPI 模式 10 时序图（CPOL = 1, CPHA = 0）

20.2.3.4. SPI 模式 11

当 SPI 设为模式 11 时，在空闲状态下 SPI 时钟保持高电平。主机通过软件将从机的片选信号 CS 拉低，往 SPI 数据发送寄存器内写入数据后，即启动一次 SPI 数据帧的收发过程。在每一个时钟下降沿主机和从机各自将数据打出就绪，然后在其后的时钟上升沿采样接收对方发过来的数据位并送入内部移位寄存器。当一个 SPI 数据帧（4~32 位）传输结束后，接收到的数据从内部移位寄存器转存入 SPI 数据接收寄存器，SPI 时钟恢复到高电平状态。主机可继续往发送寄存器内写入数据进行连续发送，或者通过软件把从机片选信号拉高，结束 SPI 通讯。

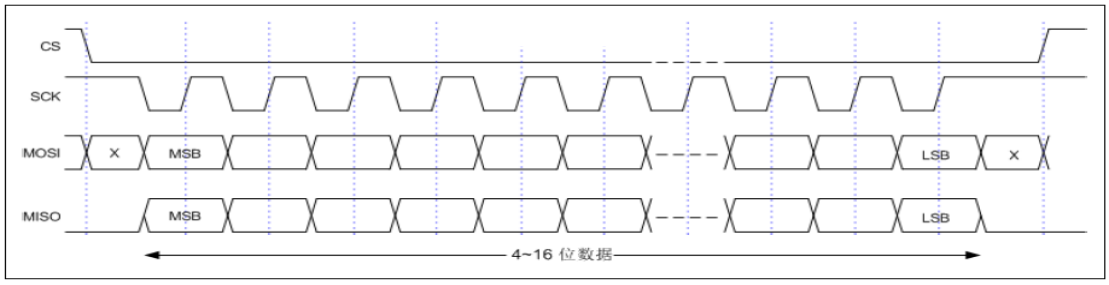


图 20-5 单脉冲模式的例子

©Copyright 2026, PADAUK Technology Co. Ltd

20.3. SPI 通讯波特率计算和设定

当芯片被设置为 SPI 主机时，通讯时钟由自己控制产生并发送至从机，其通讯波特率计算公式如下:

BPSSPI = SPI_REF_CLK / SPI_BD

其中:

- BPSSPI 为期望的 SPI 通讯波特率，同时用于接收和发送
- SPI_REF_CLK 为 SPI 工作时钟频率
- SPI_BD 为 baud rate divisor 分频寄存器的设定值，取值为 2, 4, 8, 16，32，64，128，256

当芯片被设置为 SPI 从机时，SPI 通讯时钟为外部输入。

20.4. SPI 时钟

SPI 的主时钟域为 SPI_REF_CLOCK，所有从移位寄存器读或写入的操作都受此时钟支配。此外，SPI 中的状态机与大多数控制逻辑都与此时钟为同步关系。所有的状态寄存器都与 pclk 为同步关系。此 SPI 模块还具备另一种时序规则，工作方式取决于其被设置成主端或从端模式，描述如下:

1. Master Clock Generation

在主端模式下，SPI 时钟输出的 pin 脚 需要与 SPI_REF_CLOCK 时钟域保持同步关系。此外，输出的频率还可通过设置 *SPI_CR* 寄存器中的 bit [5:3] *BD* 位进一步降低。

2. Slave Clock Generation

在从端模式下，SCK 必须被用于接收数据的采样与从端数据的发送。此时钟仅在从端模式下有效并且由外部的主端所驱动。为了读取 RX FIFO 中的数据并对收到的数据采样，SCK 会与 SPI_REFERENCE_CLOCK 同步，其选通脉冲信号将控制收到的数据的移入操作。

其控制遵守如下规则:

1. 当 SPI 模块使能被关闭，SCK 将始终保持为低
2. 当 SPI 被使能但从端的同步开始条件未满足时 SCK 将被置为低

20.5. SPI FIFOs

发送和接收 FIFO 的深度均为 8。FIFO 的状态可通过读取 STATUS 寄存器获得。SPI 模块的中断控制可通过 FIFO 的状态进行触发。

当 RX FIFO 被写满但仍然有尝试写 RX FIFO 的操作时 RX FIFO 的溢出标识位将被设置为有效，此标识位将一直保持激活直到有读取 RX FIFO 并且对 status 相应位写 1 的操作发生。当 RX FIFO 为满时则写入的数据都无法存入，为避免出现数据无法写入的情况，在 RX FIFO 满载之前必须执行读取操作。如果在从端模式下，RX FIFO 满载将引发传入数据的丢失。

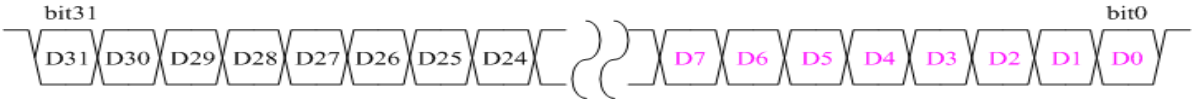
20.6. SPI 数据帧构成

PEC930 的 SPI 模块收发的每一帧数据长度可为 4~32 位编程。在模块通讯空闲状态下 SCK 时钟信号保持在非活跃态（非活跃态电平由 CPOL 位决定），并且当数据接收缓冲队列中存有数据时，通过判断该空闲状态所持续的时间来产生数据接收队列非空超时的状态信息。一帧数据支持 LSB 和 MSB 两种发送数据配置。

20.7. SPI 数据帧传输

20.7.1. 8 位数据传输

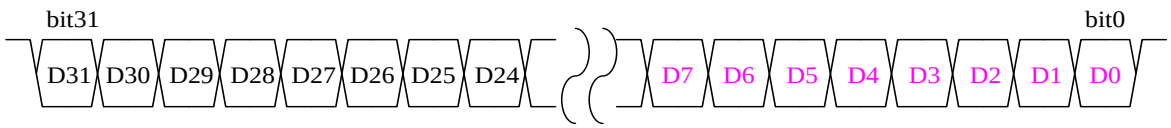
TXFIFO DATA



SPI MSB 先传输

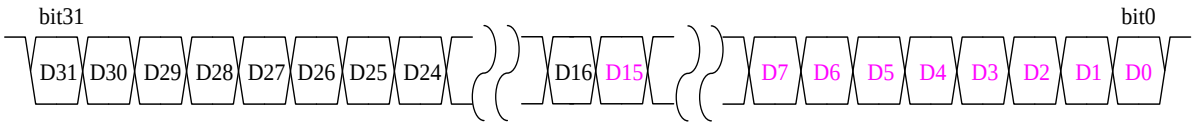
SPI LSB 先传输

RXFIFO DATA

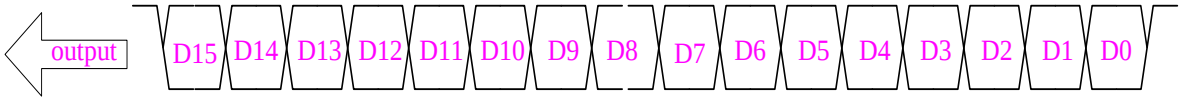


20.7.2. 16 位数据传输

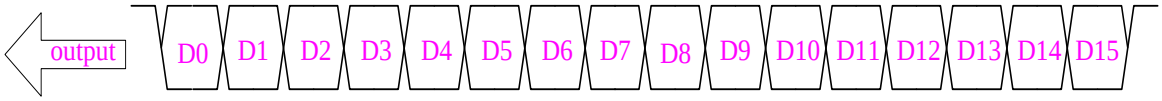
TXFIFO DATA



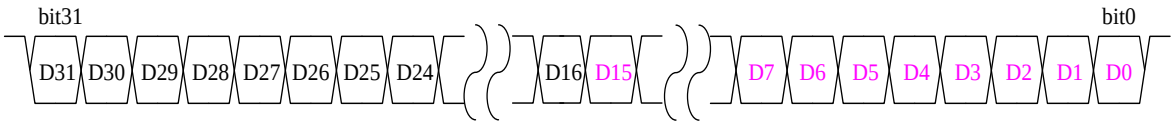
SPI MSB 先传输



SPI LSB 先传输

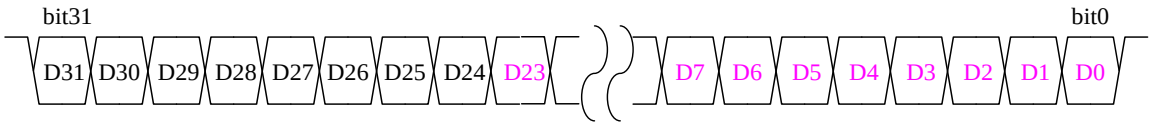


RXFIFO DATA

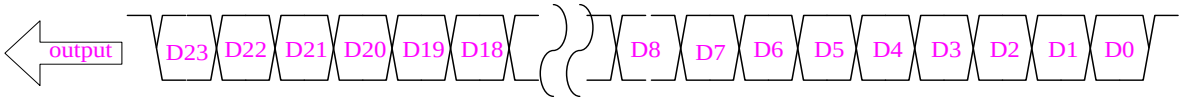


20.7.3. 24 位数据传输

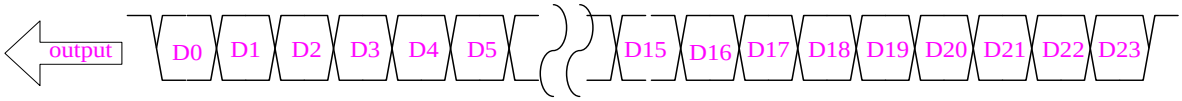
TXFIFO DATA



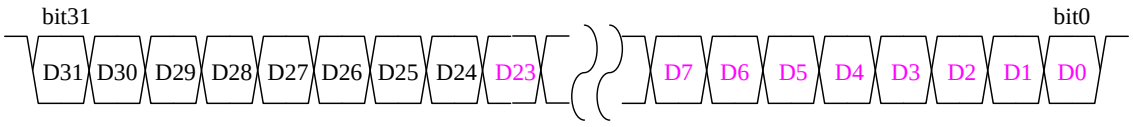
SPI MSB 先传输



SPI LSB 先传输

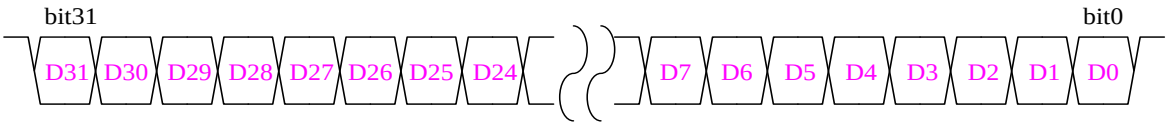


RXFIFO DATA

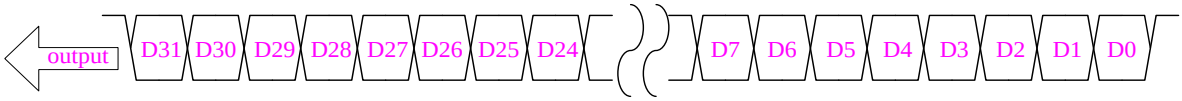


20.7.4. 32 位数据传输

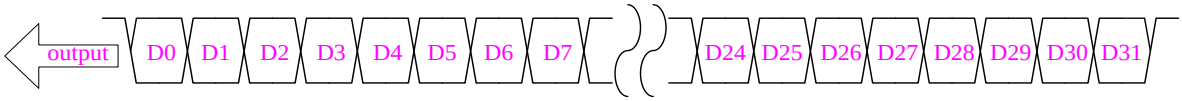
TXFIFO DATA



SPI MSB 先传输



SPI LSB 先传输



RXFIFO DATA

20.8. SPI 中断标志和中断响应

PEC930 的 SPI 模块共有 4 个可产生中断请求的中断源，分别对应数据发送和接收缓冲队列的不同状态:

- 接收队列溢出。当接收队列中已经存满 8 个数据但应用软件没有及时读取，这时又收到第 9 个数据时，缓冲队列就发生溢出，最前收到的数据被顶出队列而丢弃，接收溢出标志被置位。
- 接收队列中有数据（非空）且应用软件读取数据超时．当接收队列中已存有一个或多个数据但软件 没有在一定时间内（64 个系统时钟计数）把它们读出，非空超时标志被置位.
- 接收队列中的数据已滿。当接收队列中存满 8 个数据，接收数据滿載标志被置位。
- 发送队列中的空间已滿。当发送队列中存满 8 个数据，发送空间滿載标志被置位。

上述 4 个状态标志按 SPI 模块接收和发送缓冲队列的实际状态同步反映在 SPI_SR 中，但其不能直接产生中断请求。中断使能控制寄存器 SPI_INTEN 决定了其中哪些中断源被使能进而可发出中断请求。被使能后的中断标志最终向 CPU 发出同一个中断请求，用户软件必须在中断服务程序中分别查询各中断标志以确定具体的中断源并做相应的处理服务。

©Copyright 2026, PADAUK Technology Co. Ltd

20.9. 寄存器列表

地址偏移	寄存器	描述
0x4000_3800	SPI_CR	SPI 控制寄存器
0x4000_3804	SPI_SR	SPI 状态寄存器
0x4000_3808	SPI_INTEN	SPI 中断使能寄存器
0x4000_380C	SPI_INTDIS	SPI 中断禁止寄存器
0x4000_3810	SPI_INTMASK	SPI 中断屏蔽寄存器
0x4000_3814	SPI_ENABLE	SPI 使能制寄存器
0x4000_3818	-	保留
0x4000_381C	SPI_TX	SPI 发送数据 FIFO
0x4000_3820	SPI_RX	SPI 接收数据 FIFO
0x4000_3824	SPI_IDLECNT	SPI 从端待命计数器
0x4000_3828	SPI_TXTH	SPI 发送 FIFO 状态阈值
0x4000_382C	SPI_RXTH	SPI 接收 FIFO 状态阈值

表 20-1 SPI 寄存器列表

©Copyright 2026, PADAUK Technology Co. Ltd

20.10. 寄存器描述

20.10.1. SPI 控制寄存器 (SPI_CR)

Bit	Name	R/W	Reset	Description
31:19	-	R	0x0	保留
18	-	R/W	0x0	保留
17	MODEF	R/W	0x1	ModeFail 功能使能 。 此位默认值 1 表示 ModeFail 功能启用，对此位写 0 将关闭 ModeFail 功能
16	MSTC	R/W	0x0	手动启动命令 启动模式使能时，对此位写 1，在 TX FIFO 不为空时将启用数据传输功能，写 0 无效。读回固定为 0。
15	MSE	R/W	0x0	手动模式使能 当 MSTC = 0，将对 FIFO 执行写入多字节的操作为传输做准备。在手动模式下，如果写 MCS = 0（有效状态）则片选信号将仅在有字写入时变为有效状态。当 MSE = 0 时，手动模式不生效，数据传输将在 FIFO 中写入数据后立刻开始。
14	MCS	R/W	0x0	手动片选 当此位设置为使能时，片选总线将始终被设定的外设选择码驱动而不被模块内的片选状态机所驱动。
13:11	-	R/W	0x0	保留
10	CS	R/W	0x0	外设片选码 0: 外设被选中 1: 没有外设选中

<table>
<tr><td>9</td><td>-</td><td>R/W</td><td>0x0</td><td>保留</td></tr>
<tr><td>8</td><td>LSB</td><td>R/W</td><td>0x0</td><td>LSB 或 MSB 先传输选择
0: MSB 先传输
1: LSB 先传输</td></tr>
<tr><td>7:6</td><td>TXWDSZ</td><td>R/W</td><td>0x0</td><td>传输字的大小.此处必须将字大小设置为与 FIFO 宽度相等或可整除 FIFO 宽度的值，当设置成可整除 FIFO 宽度的字大小时则可使用多字传输功能。
00: 8 bits datasize
01: 16 bits datasize
10: 24 bits datasize
11: 32 bits datasize</td></tr>
<tr><td>5:3</td><td>BD</td><td>R/W</td><td>0x0</td><td>主端模式波特率除数，SPI 波特率可通过 SPI_REFERENCE_CLK / BD 得到，此位可配的
<table>
<tr><td>BD [2:0]</td><td>Discrption</td></tr>
<tr><td>000</td><td>÷ 2</td></tr>
<tr><td>001</td><td>÷ 4</td></tr>
<tr><td>010</td><td>÷ 8</td></tr>
<tr><td>011</td><td>÷ 16</td></tr>
<tr><td>100</td><td>÷ 32</td></tr>
<tr><td>101</td><td>÷ 64</td></tr>
<tr><td>110</td><td>÷ 128</td></tr>
<tr><td>111</td><td>÷ 256</td></tr>
</table></td></tr>
</table>

©Copyright 2026, PADAUK Technology Co. Ltd

2	CPHA	R/W	0x0	时钟相位选择 0: 传输字外 SPI clock 为有效状态（片选不拉高） 1: 传输字外 SPI clock 为无效状态（片选拉高）
1	CPOL	R/W	0x0	时钟极性选择 0: 时钟为低时为待命态 1: 时钟为高时为待命态
0	MODE	R/W	0x0	模式选择 0: 从端模式 1: 主端模式

注意: 此 SPI 可支持对每个 FIFO 字进行多字传输，下表说明 Datasize(以 bit 为单位)与 FIFO 字宽(FF_W)间的关系

FF_W	Datasize	Words Transferred/FIFO Location
32	32	1
32	16	2 (First transfer = FIFO [31:16])
32	8	4 (First transfer = FIFO [31:24])

20.10.2. SPI 状态寄存器 (SPI_SR)

SPI_SR 寄存器为只读寄存器,有效位宽 7 位，当相应事件出现并且中断功能在 MASK 寄存器中被开启时此寄存器中的值将会改变。此寄存器中任一值为高说明相应事件触发了中断信号输出。默认情况下，此寄存器的第 0，1，6 位可通过写 1 清 0.

Bit	Name	R/W	Reset	Description
31:8	-	R	0x0	保留
7	BUSY	R	0x0	SPI 收发忙状态 0: SPI 模块空闲 1: SPI 模块繁忙
6	TXUFL	R/W1c	0x0	TX FIFO 下溢 0: 无下溢出现 1: 出现下溢，表明 FIFO 为空时系统仍在尝试发送数据。如果出现下溢并导致此寄存器位变为 1 后则此位仅在系统复位时被重置并且仅在对此寄存器进行读取操作后清空。
5	RXFUL	R	0x0	RX FIFO 满载 (当前的 FIFO 状态指示) 0: FIFO 未满载 1: FIFO 已经满载
4	RXNEP	R	0x0	RX FIFO 不为空 （当前 FIFO 状态指示) 0: FIFO 小于 设定的阈值 1: FIFO 大于等于设定的阈值
3	TXFUL	R	0x0	TX FIFO 满载 (当前 FIFO 状态指示) 0: FIFO 未满载 1: FIFO 已经满载:

2	TXNFUL	R	0x1	TX FIFO 未满载 (当前 FIFO 状态指示) 0: FIFO 大于等于设定阈值 1: FIFO 小于设定阈值
1	MDF	R/W1c	0x0	模式错误 — 指示系统出现 n_ss_in 上的电压与 SPI 现处模式无法匹配的情况。当此位为 1 时说明 n_ss_in 在从端模式传输字当中拉高（变为无效状态）此状态位变为 1 将关闭 SPI 使能。仅在系统复位后被重置，在读取此寄存器后清空。 (注意！mode fail 仅支持从端模式！) 0: 未出现按模式错误 1: 出现模式错误
0	RECVOV	R/W1c	0x0	接收溢出 ——如果尝试在 RX FIFO 已满时继续向其中写入,则此位变为 1。此位仅在系统重置后进行重置,并且仅在读取此寄存器时清除。如果当此寄存器位已经为高时出现新的数据写入操作到 RX FIFO，此标志将保持为 1。 0:未检测到溢出。 1: 发生了溢出。

20.10.3. SPI 中断使能寄存器 (SPI_INTEN)

Bit	Name	R/W	Reset	Description
31:7	-	R	0x0	保留
6:0	INTEN	WO	0x0	中断使能，开启相应事件产生中断的功能，每一位控制一个中断产生事件的使能，事件与 SPI_SR 寄存器中一一对应 0: 无效 1: 写入 1 的位表示与中断状态寄存器中相应位对应事件的中断产生被打开

注: SPI_INTEN 中断使能寄存器有 7 个有效位，为只写模式。写此寄存器的相应位可使能在 SPI 状态寄存器中对应的中断产生事件。

20.10.4. SPI 中断禁止寄存器 (SPI_INTDIS)

Bit	Name	R/W	Reset	Description
31:7	-	R	0x0	保留
6:0	INTDIS	WO	0x0	中断禁止，关闭相应事件产生中断的功能，每一位可禁用一个中断产生事件的产生，事件与 SPI_SR 寄存器中一一对应 0: 无效 1: 写入 1 的位表示与中断状态寄存器中相应位对应事件的中断产生被禁用

注: SPIn_INTDIS 寄存器有 7 个有效位，为只写模式。写此寄存器的相应位可禁止在 SPI 状态寄存器中对应的中断产生事件产生中断。

20.10.5. SPI 中断屏蔽寄存器 (SPI_INTMASK)

Bit	Name	R/W	Reset	Description
31:7	-	R	0x0	保留
6:0	INTMsk	RO	0x0	0: 与 SPI_SR 寄存器中相应位对应的中断产生事件的中断产生机制被禁用 1: 与 SPI_SR 寄存器中相应位对应的中断产生事件的中断产生机制被使能

注: SPIn_INTMASK 此寄存器为只读寄存器，有 7 个有效位，每一位对应一个中断事件，当读取到 1 时说明该位对应的中断事件被使能。

20.10.6. SPI 使能寄存器 (SPI_ENABLE)

SPI_ENABLE 使能寄存器仅 1 位有效，决定了 spi 模块是否被使能。使能位设为 0 则 SPI 模块将关闭，当在传输中将 SPI 模块关闭则 SPI 模块将在传输完当前字后关闭。在使能关闭模式下，所有 SPI 的输出使能都将关闭并且所有 SPI 模块的 pin 都将被设置成输入模式。

Bit	Name	R/W	Reset	Description
31:1	-	R	0x0	保留
0	SPI_EN	R/W	0x0	SPI 使能控制 0: 当完成当前的数据传输后关闭 SPI 模块 1: 使能 SPI 模块

20.10.7. SPI 发送数据 FIFO (SPI_TX)

Bit	Name	R/W	Reset	Description
31:0	TXFIFO	WO	-	寫入发送数据

20.10.8. SPI 接收数据 FIFO (SPI_RX)

Bit	Name	R/W	Reset	Description
31:0	RXFIFO	RO	-	存入接收数据

20.10.9. SPI 从端待命计数器 (SPI_IDLECNT)

Bit	Name	R/W	Reset	Description
31:8	-	R	0x0	保留
7:0	IDLECNT	R/W	0xFF	在从模式下，SPI 仅在对方主端输入的串行时钟 sclk_in 处于稳定 IDLE 电平时 SPI 参考时钟周期数与预设的 SPI_IDLECNT 相等或大于时才能检测到字传输开始信息。当从端未被片选中时则从端处于静止状态，此时此计时器将根据 SPI 参考时钟对从端处于静止状态的时间进行计数

20.10.10. SPI 发送 FIFO 状态阈值 (SPI_TXTH)

Bit	Name	R/W	Reset	Description
31:3	-	R	0x0	保留
2:0	TXTHOLD	R/W	0x1	此寄存器可用位数与 TX FIFO 深度一致,该寄存器定义 TX FIFO 产生“非满”中断的阈值

20.10.11. SPI 接收 FIFO 状态阈值 (SPI_RXTH)

Bit	Name	R/W	Reset	Description
31:3	-	R	0x0	保留
2:0	RXTHOLD	R/W	0x1	此寄存器可用位数与 RX FIFO 深度一致,该寄存器定义 RX FIFO 产生“非空”中断的阈值

©Copyright 2026, PADAUK Technology Co. Ltd

21. I2C 总线通讯模块

21.1. I2C 模块综述

I2C是嵌入式系统设计中经常被用到的一种串行通讯总线。它基于 SCL（串行时钟）和 SDA（串行数据）双线联机，以主从方式实现多个互联器件之间的双向数据通讯。关于 I2C 协议的详细介绍可从 NXP 公司官方网站下载。

PEC930 片内 I2C 模块支持主模式和从模式通讯方式，其基本特性如下:

- 内含并行数据/串行 I2C 协议转换器
- 提供总线仲裁机制，支持多个主机并存
- 支持 7 位从机寻址模式
- 支持广播呼叫
- 提供数据发送和接收状态标识
- 提供字节传输结束标识
- 通讯错误检测
- 支持标准（100Kbps）或快速（400Kbps）I2C通讯时序

作为 I2C 主机时，其特性包括:

- 产生总线通讯时钟
- 实现总线仲裁
- 7位地址寻址从机，并控制数据读写方向
- 产生总线通讯起始位、重复起始位和停止位
- 检测通讯错误

作为 I2C 从机时，其特性包括:

- 检测起始位、重复起始位和停止位
- 可编程 I2C 7 位地址匹配寻址
- 传输过程中检测起始和停止条件
- 检测通讯错误

I2C 模块在开始工作前，首先必须由软件按芯片的系统时钟频率设定 I2C_CTLSET 中 CR2/CR1/CR0 分频系数，选择合适的 SCL 通讯时钟频率；然后配置 I2C_CTLSET 寄存器 EN 位使能 I2C 模块，通讯过程的状态信息则在I2C_STAT 状态寄存器中反映。如果作为I2C从机使用，则还需在 I2C_ADDR 地址寄存器中设定本机的地址码。

启用 I2C 模块前，还需要将 SDA、SCL 的 IO 复用功能使能，使 SDA、SCL 端口的 GPIO 功能禁止，IO 口作为 SDA、SCL 使用。具体参考 GPIO 模块端口复用的说明。启用 I2C 模块后，SDA 和 SCL 线被配置成开漏输出。支持芯片内部恒流源上拉输出可配（请参考模拟杂项章节恒流源有关寄存器）。外部硬件也可以用上拉电阻将其拉至高电平，上拉电阻阻值视工作电压、通讯速度和总线负载而定，一般可选 5~10K。在发送数据前，SCL 线先被发送方拉低，软件写入一个字节到 I2C_DATA 数据寄存器后，SDA 和 SCL 开始逐位输出数据和时钟信号进行数据发送；在接收到数据后，SCL 线被接收方持续拉低，直到接收方从 I2C_DATA 数据寄存器中读出数据。如果 I2C 模块被禁止，SDA 和 SCL 线可作为普通 IO 引脚功能。

©Copyright 2026, PADAUK Technology Co. Ltd

21.2. I2C 协议简述

I2C 总线协议要求所有挂在总线上的设备，其 SDA 和 SCL 线必须是开漏输出，换句话说，任何设备可以输出低电平将 SDA 或 SCL 线拉低，但不能输出高电平驱动。总线上的高电平只能依靠硬件上拉电阻获得。

一般的 I2C 总线通讯包含四部分:

- 起始位发送，表示一次通讯过程开始
- 从机地址发送
- 数据传输
- 停止位发送，表示一次通讯过程结束

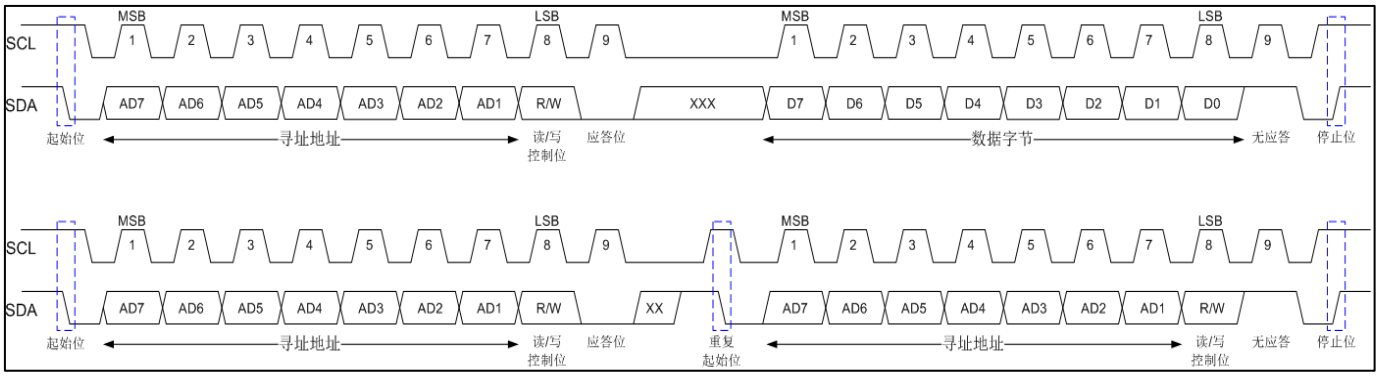


图 21-1 I2C总线通讯波形示意图

21.2.1. 起始位

总线空闲时，没有任何器件占用总线（SCL 和 SDA 线处于逻辑高电平），主机可以通过发送起始位发起通信。起始位定义为，当 SCL 在高电平时，SDA 从高到低的跳变。该信号表示开始新的数据传输（每次数据传输可能包含几个字节的数据），并使所有从机退出空闲状态。

21.2.2. 从机寻址

起始位发出后传输的第一个字节数据是主机发送的从机寻址地址。其中的高 7 位是从机的地址码，最低位 R/W 决定总线上数据传输的方向:

1 = 读取传输，从机向主机传输数据

0 = 写入传输，主机向从机传输数据

当主机发送的地址与某从机地址匹配时，该从机在第 9 个时钟周期时将 SDA 线拉低（见图 21-1），回送应答位进行响应。

I2C 总线上不能有两个地址相同的从机。如果 I2C 模块是主机，它就不应该发送与其自身从机地址相同的地址。I2C 器件不能同时既是主机又是从机，但是如果在寻址过程中出现仲裁丢失，器件就重新返回到从机模式并正确运行，并可被另一个主机寻址。

21.2.3. 数据传输

成功实现从机寻址后，就可以按照主机发送的 R/W 位指定的方向逐字节地进行数据传输。主机发送地址后的所有传输都被称为数据传输，即使它们包含从机的子地址报文。

每个数据字节的长度均为 8 位。只有当 SCL 为低时数据才可以更改，如果 SCL 为高，那么 SDA 必须保持稳定，SCL 的一个时钟脉冲传输一个数据位，最高位被首先传输，如图 21-1 所示。每个数据字节后面都有一个应答位（第9位），该位由从机（发送时）或主机（读取时）回送信号，通过在第 9 个时钟周期时把 SDA 线拉低来实现。这样，一个完整数据字节的传输需要9个时钟脉冲。如果对方在第9个时钟周期时无应答，则其必须保留 SDA 线在高电平。主机将接收到的无应答信号解释为不成功的数据传输；如果主机在接收一个数据字节传输后未应答从机，从机将其理解为数据传输结束，并释放 SDA 线。

对于这两种情况，数据传输都被中止，主机会进行以下两种操作之一:

- 发送停止位，释放总线
- 发送重复起始位，开始新呼叫

21.2.4. 停止位

主机可以通过发送停止位终止通信，以释放总线。然而，主机可以直接发送起始位和呼叫命令，而无需首先发送停止位，这被称为重复启动。停止位的定义是 SCL 在逻辑 1 位置时的从低到高的 SDA 跳变（见图 21-1）。即便从机发出一个应答，主机也可以发送停止位，此时从机必须释放总线。

21.2.5. 重复起始位

如图 21-1 所示，重复起始位是在无需首先生成停止位以终止通信的情况下发送的信号。该信号由主机用来与另外一个从机进行通信，或者在不同模式（发送/接收模式）中与同一从机进行通信，而不需要释放总线。

21.2.6. 总线仲裁

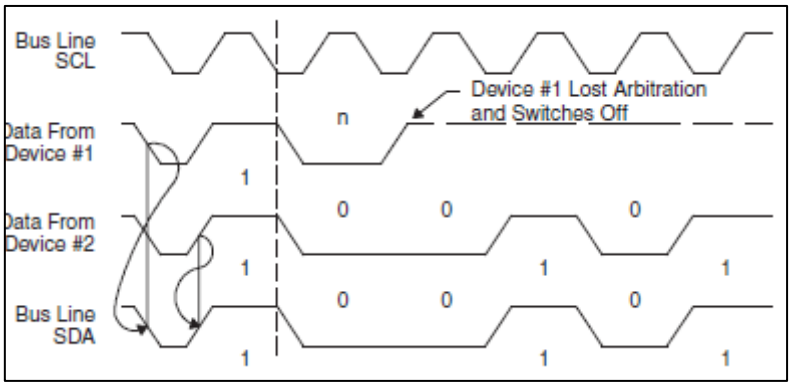


图 21-2 I2C 两个主机仲裁过程示意图

I2C 总线是真正的多主控总线，允许一个以上的主机连接到 I2C 上。如果两个甚至多个主机试图同时控制总线，那么就由时钟同步过程来决定总线时钟，总线低周期等于最长的时钟低段，总线高周期等于最短的时钟高段。竞争主机的相对优先级由数据仲裁过程确定，如果一个总线主机发出逻辑 1，而另一个发出逻辑 0，那么前者就丢失仲裁。失败的一方立即切换到从机接收模式，停止驱动 SDA 输出。在这种情况下，从主模式到从模式的转换不会生成停止条件。与此同时，会通过硬件设置一个状态位，表示仲裁丢失。

©Copyright 2026, PADAUK Technology Co. Ltd

21.2.7. 时钟同步

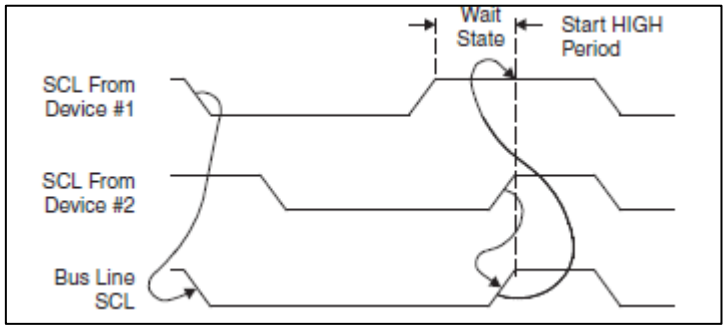


图 21-3 I2C 总线仲裁过程中 SCL 波形示意图

在仲裁过程中，不同主机发送的时钟需要被同步。由于 SCL 线上的线与逻辑，所以 SCL 上的电平跳变受连接到总线上的所有器件影响。图21-3 显示了仲裁过程中，SCL 线上的电平变化。当主机的时钟进入低电平周期后，它将一直保持 SCL 线的低电平，直到时钟到达高电平。然而，如果另一个主机时钟依旧是低电平，这时主机时钟从低到高的变化就不会改变 SCL 线的状态。因此，经过同步的时钟 SCL 拉低的时间由具有最长的低电平周期的主机决定。低电平周期较短的主机在该时段进入高电平等待状态（wait-state）。当所有主机均结束低电平周期时，同步时钟 SCL 线才被释放和拉高。这时主机时钟和 SCL 线的状态一致，所有主机开始计数它们的高电平周期。第一个结束高电平周期的主机再次拉低 SCL 线。

所以，经过同步的时钟 SCL 的低电平周期由所有主机中具有最长的低电平周期的主机决定，高电平周期由所有主机中具有最短高电平周期的主机决定。

21.2.8. 通讯握手

时钟同步机制可以用作数据传输中的通讯握手。在完成一个字节的传输（9个位）后，从机可以保持 SCL 低位。在这种情况下，从机会暂停总线时钟，强迫主机时钟进入等待状态，直到从机释放 SCL 线。

21.2.9. 时钟延展

时钟同步机制可以被从机用于减缓传输的比特速率。在主机已经拉低 SCL 后，从机可以继续拉低 SCL 一定时间，然后再释放它。如果从机 SCL 低电平周期长于主机 SCL 低电平周期，那么就会将 SCL 总线信号低电平周期延展。

21.3. I2C 主模式

I2C 模块在初始化后默认处于从模式状态。在总线处于空闲状态下（I2C_STAT = 0xF8），软件设定 I2C_CTLCLR.STA = 1，模块即进入主模式并在总线上产生一个起始位。起始位产生完毕后 I2C_STAT 更新状态为 0x08，并在中断允许时产生中断请求。此时主机将 SCL 线持续拉低，直到软件往 I2C_DATA 数据寄存器内写入首个字节进行发送。

21.3.1. I2C 主模式寻址方式

I2C 主模式下在起始位后的首个发送字节一定是从机的地址码，支持 7 位寻址模式。

7 位寻址:

- 主机发送地址码字节（7 位地址加最低 1 位读写控制）
- 查询等待或中断响应字节发送完毕
- 若从机应答，则 I2C_STAT = 0x18；若无应答，则 I2C_STAT = 0x20

地址发送完毕且从机正确应答后，主机将 SCL 线持续拉低，下一步主机将进入数据发送或数据接收过程。

21.3.2. I2C 主模式数据发送

主机发送完从机的寻址地址（其中的读写控制位为 0）后，可以继续往 I2C_DATA 寄存器内写入数据进行发送，在数据被写入 I2C_DATA 寄存器前主机将 SCL 线持续拉低。主机每次发送完一个字节且收到从机的应答，则 I2C_STAT = 0x28；若从机无应答，则 I2C_STAT = 0x30，模块必须发送停止位终止 I2C 通讯。当最后一个字节发送完毕后，主机发送一个停止位，模块随即转为从模式。

21.3.3. I2C 主模式数据接收

主机发送完从机的寻址地址（其中的读写控制位为 1）后，主机即开始从从机处读取数据。每读取一个字节后主机视 ACK 位的设定自动产生应答（I2C_CTLSET.AA = 1时）或非应答（I2C_CTLSET.AA = 0时），主机软件可查询I2C_STAT状态或中断响应，此时主机持续将SCL线拉低，直到主机软件把接收的数据从 I2C_DATA 寄存器中读走。在读取最新收到的数据前，如果主机确定下一个字节将是最后一个需读取的字节时，应将 ACK 位清 0（I2C_CTLSET.AA = 0），确保在读取下一个字节时主机会向从机回送非应答位；读取最后一个字节后，发送一个停止位，模块随即转为从模式。

21.3.4. I2C 主模式错误信息

I2C 模块在主模式下如果检测到总线通讯发生错误，I2C_STAT 状态将被置为 0x00，同时模块将释放对总线的控制。

21.4. I2C 从模式

I2C 模块在初始化后默认处于从模式状态，等待总线上产生起始位，随后接收主机发出的寻址地址码并和本机设定地址进行比对匹配。

21.4.1. I2C 从模式地址匹配

I2C 从模式地址寻址为 7 位地址:

- 从机接收的地址码字节高 7 位和 I2C_ADDR 寄存器设定的高 7 位本机地址作匹配比对，接收的地址码字节最低位为读写控制位，不参与地址比对
- 7 位地址比对匹配成功，从机回送应答位；否则回送非应答位，从机等待下一个起始位
- 地址匹配后从机将 SCL 线持续拉低，直到从机软件读一次 I2C_STAT 寄存器后将其释放

21.4.2. I2C 从模式数据接收

在地址匹配流程后（其中的读写控制位为 0）从机即进入数据接收状态。从机每接收到一个字节后即回送一个应答信息（应答或非应答，取决于 ACK 位的设定），并在中断允许时产生中断请求。此时从机将 SCL 线持续拉低，直到从机软件从 I2C_DATA 寄存器内将数据读出。

21.4.3. I2C 从模式数据发送

在地址匹配流程后（其中的读写控制位为 1）从机即进入数据发送状态，将 SCL 线持续拉低，从机软件往 I2C_DATA 寄存器内写入一个字节后 SCL 线被释放，随即开始发送。在收到对方的应答信息后更新 I2C_STAT 状态，中断允许时产生中断请求。

21.4.4. I2C 从模式通讯终止

在最后一个字节传输完成后如果主机发出停止位，从机也随即终止本次通讯过程，并在中断允许时产生中断请求。

21.4.5. I2C 从模式错误信息

I2C 模块在从模式下如果检测到总线通讯发生错误，I2C_STAT 状态将被置为 0x00，同时模块将释放对总线的控制。

21.5. I2C 时钟速度计算和设定

I2C 通讯的时钟速度取决于系统时钟频率和分频系数．分频系数通过 I2C_CTLSET 寄存器中 CR2/CR1/CR0 三位选择．具体计算方法如下:

fSCL = fSYS / DIV

其中:

fSCL 为期望的 I2C 通讯 SCL 时钟频率

fSYS 为系统时钟频率

DIV 为系统时钟分频系数，按表 21-1 选择

应用软件在已知系统时钟频率的前提下，选择合适的分频系数来设定 I2C 通讯的时钟频率．I2C 最高时钟频率不应超过 1Mbps。

CR2	CR1	CR0	DIV
0	0	0	256
0	0	1	244
0	1	0	192
0	1	1	160
1	0	0	960
1	0	1	120
1	1	0	60
1	1	1	保留

表 21-1 產生 I2C 时钟频率时系统时钟分频系数

21.6. I2C 状态信息和中断响应

I2C 模块有多个事件可产生同一个中断标识（I2C_CTLSET.SI）。各事件由 I2C_STAT 寄存器中的位 [7:3] 状态信息表述，见下表说明。

I2C_STAT [7:3]	I²C总线状态	I2C_CTLSET相关位信息				I²C模块的下一步应对
		STA	STO	SI	AA	
00000	主模式或从模式被寻址下I²C通讯错误	0	1	0	X	I²C总线释放
00001	主模式发送起始位(START) 完成	X	0	0	X	发送从机地址+写控制位，接收ACK
00010	主模式发送重复起始位(Repeated START) 完成	X	0	0	X	同发送起始位（00001） 发送从机地址+读控制位，主机转入接收模式
00011	主模式发送从机地址+写控制位完成，收到ACK响应	0 1 0 1	0 0 1 1	0 0 0 0	X X X X	发送数据字节并接收ACK 发送重复起始位 发送停止位，STO标志置位 停止位后紧接着发送起始位，STO标志清零
00100	主模式发送从机地址+写控制位完成，无ACK响应	0 1 0 1	0 0 1 1	0 0 0 0	X X X X	发送数据字节并接收ACK 发送重复起始位 发送停止位，STO标志置位 停止位后紧接着发送起始位，STO标志清零
00101	主模式发送数据字节完成，收到ACK响应	0 1 0 1	0 0 1 1	0 0 0 0	X X X X	发送数据字节并接收ACK 发送重复起始位 发送停止位，STO标志置位 停止位后紧接着发送起始位，STO标志清零

00110	主模式发送数据字节完成，无ACK响应	0 1 0 1	0 0 1 1	0 0 0 0	X X X X	发送数据字节并接收ACK 发送重复起始位 发送停止位，STO标志置位 停止位后紧接着发送起始位，STO标志清零
00111	主模式在发送地址或数据时总线仲裁丢失	0 1	0 0	0 0	X X	I^2C总线被释放 I^2C总线空闲时发送起始位
01000	主模式发送从机地址+读控制位完成，收到ACK响应	0 0	0 0	0 0	0 1	接收数据，无ACK响应 接收数据，有ACK响应
01001	从机地址+读控制位发送完成，无ACK响应	1 0 1	0 1 1	0 0 0	X X X	发送重复起始位 发送停止位，STO标志置位 停止位后紧接着发送起始位，STO标志清零
01010	数据字节接收完成，回送ACK响应	0 0	0 0	0 0	0 1	接收数据，无ACK响应 接收数据，有ACK响应
01011	数据字节接收完成，回送NACK响应	1 0 1	0 1 1	0 0 0	X X X	发送重复起始位 发送停止位，STO标志置位 停止位后紧接着发送起始位，STO标志清零
01100	从模式地址+写控制位接收完成，回送ACK响应	X X	0 0	0 0	0 1	接收数据，回应NACK 接收数据，回应ACK
01101	主模式下总线仲裁丢失，作为从模式收到地址+写控制位，并回送ACK响应	X X	0 0	0 0	0 1	接收数据，回应NACK 接收数据，回应ACK
01110	收到广播寻址（0x00），并回送ACK响应	X X	0 0	0 0	0 1	接收数据，回应NACK 接收数据，回应ACK

01111	主模式下总线仲裁丢失，作为从模式收到广播寻址，并回送ACK响应	X X	0 0	0 0	0 1	接收数据，回应NACK 接收数据，回应ACK
10000	从模式被寻址下收到一个数据字节，，并回送ACK响应	X X	0 0	0 0	0 1	接收数据，回应NACK 接收数据，回应ACK
10001	从模式被寻址下收到一个数据字节，并回送NACK响应	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	模块转为非寻址下的从模式，不再识别地址和广播寻址 模块转为非寻址下的从模式，可识别地址和广播寻址 模块转为非寻址下的从模式，不再识别地址和广播寻址；当总线空闲时将发送起始位 模块转为非寻址下的从模式，可识别地址和广播寻址；当总线空闲时将发送起始位
10010	广播被寻址下收到一个数据字节，并回送ACK响应	X X	0 0	0 0	0 1	接收数据，回应NACK 接收数据，回应ACK
10011	广播被寻址下收到一个数据字节，并回送NACK响应	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	模块转为非寻址下的从模式，不再识别地址和广播寻址 模块转为非寻址下的从模式，可识别地址和广播寻址 模块转为非寻址下的从模式，不再识别地址和广播寻址；当总线空闲时将发送起始位 模块转为非寻址下的从模式，可识别地址和广播寻址；当总线空闲时将发送起始位

10100	从模式被寻址下收到停止位（STOP）或重复起始位（Repeated START）	0	0	0	0	模块转为非寻址下的从模式，不再识别地址和广播寻址
		0	0	0	1	模块转为非寻址下的从模式，可识别地址和广播寻址
		1	0	0	0	模块转为非寻址下的从模式，不再识别地址和广播寻址；当总线空闲时将发送起始位
		1	0	0	1	模块转为非寻址下的从模式，可识别地址和广播寻址；当总线空闲时将发送起始位
10101	从模式地址+读控制位接收完成，回送ACK响应	X	0	0	0	发送最后一个字节，接收ACK应答
		X	0	0	1	发送一个字节，接收ACK应答
10110	主模式下总线仲裁丢失，作为从模式收到地址+读控制位，并回送ACK响应	X	0	0	0	发送最后一个字节，接收ACK应答
		X	0	0	1	发送一个字节，接收ACK应答
10111	从模式发送数据字节完成，收到ACK响应	X	0	0	0	发送最后一个字节，接收ACK应答
		X	0	0	1	发送一个字节，接收ACK应答
11000	从模式发送数据字节完成，无ACK响应	0	0	0	0	模块转为非寻址下的从模式，不再识别地址和广播寻址
		0	0	0	1	模块转为非寻址下的从模式，可识别地址和广播寻址
		1	0	0	0	模块转为非寻址下的从模式，不再识别地址和广播寻址；当总线空闲时将发送起始位
		1	0	0	1	模块转为非寻址下的从模式，可识别地址和广播寻址；当总线空闲时将发送起始位

11001	从模式发送最后一个数据字节完成，收到ACK响应	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	模块转为非寻址下的从模式，不再识别地址和广播寻址 模块转为非寻址下的从模式，可识别地址和广播寻址 模块转为非寻址下的从模式，不再识别地址和广播寻址；当总线空闲时将发送起始位 模块转为非寻址下的从模式，可识别地址和广播寻址；当总线空闲时将发送起始位
11111	总线空闲	0	0	0	0	-

表 21-2 I2C 通訊狀態信息

当除了总线空闲状态外的任一状态发生时，模块置 I2C_CTLSET.S I 位为 1，应用软件可以设定 I2C 模块的系统中断使能位（參考官方 N200 文件），响应中断请求并进入中断服务程序。

21.7. 寄存器列表

地址	寄存器	描述
0x4000_3000	I2C_CTLSET	I2C 控制设定寄存器
0x4000_3004	I2C_STAT	I2C 状态寄存器
0x4000_3008	I2C_DATA	I2C 数据寄存器
0x4000_300C	I2C_ADDR	I2C 地址寄存器
0x4000_3018	I2C_CTLCLR	I2C 控制清除寄存器

表 21-3 I2C 寄存器列表

21.8. 寄存器描述

21.8.1. I2C 控制设定寄存器 (I2C_CTLSET)

Bit	Name	R/W	Reset	Description
31:8	-	R	0x0	保留
7	CR2	R/W	0x0	系统时钟分频系数位 2 读为当前 CR2 位值 写: 0: 无效 1: CR2 置 1 和 CR1/CR0 一起设定 I2C 主模式时钟频率，见表 21-1 说明.
6	EN	R/W	0x0	I2C 模块使能控制 读为当前 EN 位值 0: 禁止 I2C 模块 1: 使能 I2C 模块 写 0: 无效 1: EN 置 1
5	STA	R/W	0x0	I2C 发送起始位 0: STA 置 0 1: STA 置 1
4	STO	R/W	0x0	I2C 发送停止位 0: STO 置 0 1: STO 置 1

3	SI	R/W	0x0	I2C 中断标志位 读 0: I2C 模块无中断 1: I2C 模块有中断 写 0: 无效 1: SI 标志置 1
2	AA	R/W	0x0	应答位控制 读为当前 AA 位值 0: 不发送应答位 1: 在收到地址字节或数据字节后发送应答位 写 0: 无效 1: AA 置 1
1	CR1	R/W	0x0	系统时钟分频系数位 1 读为当前 CR1 位值 写 0: 无效 1: CR1 置 1 和 CR2/CR0 一起设定 I2C 主模式时钟频率，见表 22-1 说明
0	CR0	R/W	0x0	系统时钟分频系数位 0 读为当前 CR0 位值 写 0: 无效 1: CR0 置 1 和 CR2/CR1 一起设定 I2C 主模式时钟频率，见表 21-1 说明

注: I2C_CTLSET 实现 I2C 模块相关控制位置 1，低 8 位有效。对相关位写 1 时将其置位，写 0 无效。

21.8.2. I2C 状态寄存器 (I2C_STAT)

Bit	Name	R/W	Reset	Description
31:8	-	R	0x0	保留
7:3	STAT	R	0x1F	I2C 总线状态 共有可能的 27 种状态，其中的 26 个状态可产生中断请求，见表 21-2 的说明
2:0	-	R	0x0	保留

21.8.3. I2C 数据寄存器 (I2C_DATA)

Bit	Name	R/W	Reset	Description
31:8	-	R	0x0	保留
7:0	DATA	R/W	0x0	8 位数据 保存总线发送或接收的数据 发送时：软件写入 8 位数据后随即启动发送流程 接收时：第一个自动收到的字节为地址的低字节，软件必须逐个从 I2C_DATA 寄存器读取刚接收的数据，才能顺序接收其它数据字节

21.8.4. I2C 地址寄存器 (I2C_ADDR)

Bit	Name	R/W	Reset	Description
31:24	filterdelay_enable	R	0x0	过滤延迟使能控制 该控制位写 0xAC 后，位 [19:16] 才能写入控制数据
23:20	-	R	0x0	保留
19:16	filterdelay	R/W	0x0	输入过滤延迟时钟数
15:8	-	R	0x0	保留
7:1	ADDR	R/W	0x0	从机地址 ADDR [7:1] 为 7 位寻址匹配地址
0	GC	R/W	0x0	广播寻址使能位 0: 禁止从机广播寻址 1: 使能从机广播寻址

21.8.5. I2C 控制清除寄存器 (I2C_CTLCLR)

Bit	Name	R/W	Reset	Description
31:8	-	R	0x0	保留
7	CR2_CLR	R/W	0x0	系统时钟分频系数位 2 读为当前 CR2 位值 写 0: 无效 1: CR2 清 0 和 CR1/CR0 一起设定 I2C 主模式时钟频率，见表 21-1 说明.
6	EN_CLR	R/W	0x0	I2C 模块使能控制 读为当前 EN 位值 0: 禁止 I2C 模块 1: 使能 I2C 模块 写 0: 无效 1: EN 清 0
5	STA_CLR	R/W	0x0	I2C 发送起始位 读为当前 STA 位值 0: 无起始位 1: 发送起始位过程中 写 0: 无效 1: STA 清 0
4	STO_CLR	R/W	0x0	I2C 发送停止位 读为当前 STO 位值 0: 无停止位 1: 发送停止位过程中 写 0: 无效 1: STO 清 0

3	SI_CLR	R/W	0x0	I2C 中断标志位 读 0: I2C 模块无中断 1: I2C 模块有中断 写 0: 无效 1: SI 标志清 0
2	AA_CLR	R/W	0x0	应答位控制 读为当前 AA 位值 0: 不发送应答位 1: 在收到地址字节或数据字节后发送应答位 写 0: 无效 1: AA 位清 0
1	CR1_CLR	R/W	0x0	系统时钟分频系数位 1 读为当前 CR1 位值 写 0: 无效 1: CR1 清 0 和 CR2/CR0 一起设定 I2C 主模式时钟频率，见表 21-1 说明
0	CR0_CLR	R/W	0x0	系统时钟分频系数位 0 读为当前 CR0 位值 写 0: 无效 1: CR0 清 0 和 CR2/CR1 一起设定 I2C 主模式时钟频率，见表 21-1 说明

©Copyright 2026, PADAUK Technology Co. Ltd