
第 30 章 捕捉 / 比较 / PWM / 定时器 (MCCP 和 SCCP)

目录

本章包括下列主题：

30.1	简介	30-2
30.2	寄存器	30-3
30.3	时基发生器	30-14
30.4	模块同步输出	30-15
30.5	同步和触发操作	30-16
30.6	定时器模式	30-22
30.7	输出比较和 PWM 模式	30-25
30.8	输入捕捉模式	30-54
30.9	休眠和空闲模式期间的操作	30-61
30.10	复位的影响	30-61
30.11	相关应用笔记	30-62
30.12	版本历史	30-63

注： 本系列参考手册章节旨在用作对器件数据手册的补充。本手册章节可能并不适用于所有 PIC32 器件，具体取决于器件型号。请参见具体器件数据手册中相应章节开头部分的注，以检查本文档是否支持您所使用的器件。

器件数据手册和系列参考手册章节可从 Microchip 网站下载：<http://www.microchip.com>。

30.1 简介

部分 PIC32 系列器件包含了一个或多个捕捉 / 比较 / PWM / 定时器 (CCP) 模块。这些模块类似于许多其他 16 位单片机上的多功能定时器模块。此外，它们还提供与先前所有 PIC32 器件上的输入捕捉、输出比较和通用定时器外设相当的功能。

CCP 模块可以下列三种主要模式之一工作：

- 通用定时器
- 输入捕捉
- 输出比较 / PWM

有两种不同形式的模块，通过模块可产生的 PWM 输出数量来区分。单输出模块 (SCCP) 仅提供一路 PWM 输出。多输出模块 (MCCP) 可以提供最多 6 路输出和丰富的输出控制功能，具体取决于特定器件的引脚数。

所有模块 (SCCP 和 MCCP) 都包含以下特性。

- 用户可选的时钟输入，包括系统时钟和外部时钟输入引脚
- 用于时基的输入时钟预分频器
- 用于模块中断事件或触发事件的输出后分频器
- 用于协调其他 MCCP/SCCP 模块的同步输出信号，带有用户可配置的备用和辅助源选项
- 所有模式和低功耗操作下的完全异步操作
- 用于 A/D 转换的特殊输出触发事件
- 带有可选门控操作的 16 位和 32 位通用定时器模式，用于简单的时间测量
- 捕捉模式：
 - 与 PIC32 系列早期的输入捕捉外设向后兼容
 - 基于外部事件进行 16 位或 32 位时基捕捉
 - 最高 4 级深的 FIFO 捕捉缓冲区
 - 捕捉源输入多路开关
 - 门控捕捉操作可减少噪声导致的误捕捉
- 输出比较 / PWM 模式：
 - 与 PIC32 系列先前的输出比较外设向后兼容
 - 单边沿和双边沿比较模式
 - 中心对齐比较模式
 - 可变频率脉冲模式
 - 外部输入模式

MCCP 模块还包含了以下扩展 PWM 特性：

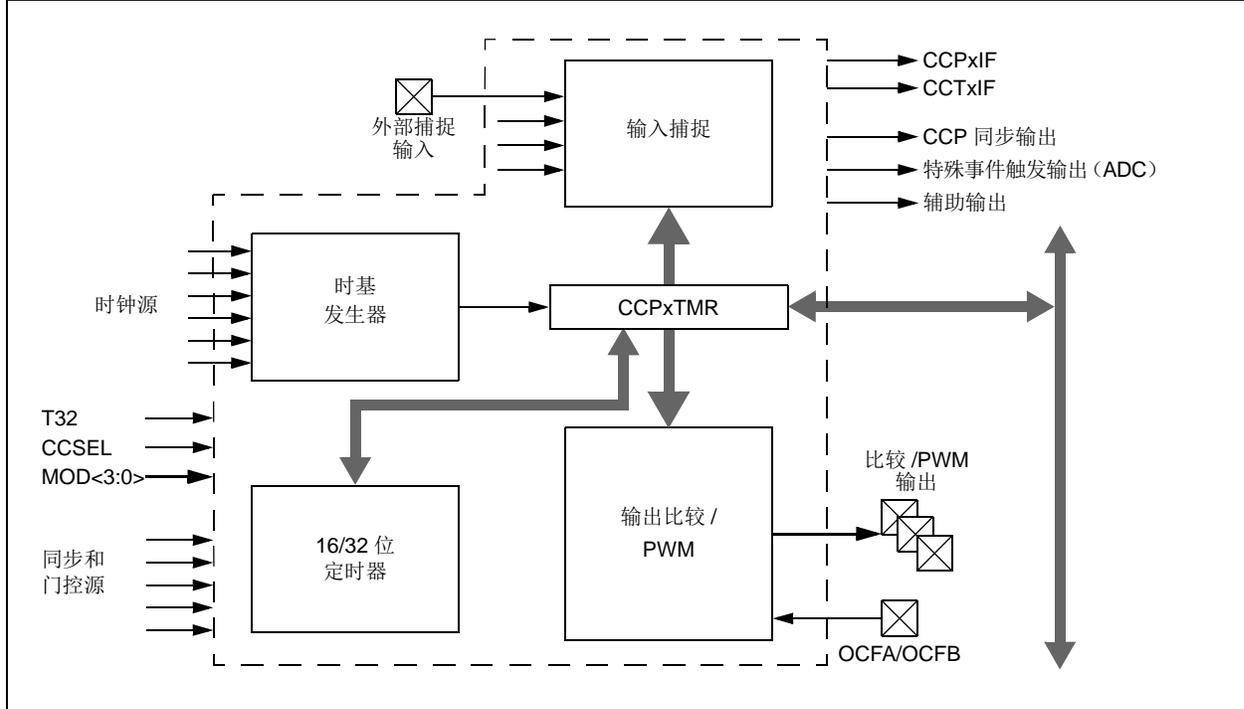
- 单输出可转向模式
- 有刷直流电机（正向和反向）模式
- 带死区延时的半桥模式
- 推挽 PWM 模式
- 输出扫描模式
- 可通过可编程源和关断状态进行自动关断
- 可编程输出极性

第 30 章 捕捉 / 比较 / PWM / 定时器 (MCCP 和 SCCP)

在任何时候，SCCP 和 MCCP 模块只能在 3 种主要模式的其中之一下工作（捕捉、比较或定时器）。除非重新配置模块，否则其他模式不可用。

图 30-1 给出了该模块的概念框图。所有三种模式均使用时基发生器（Time Base Generator, TBG）和通用定时器寄存器（CCPxTMR）。其他共用硬件元件（例如比较器和缓冲寄存器）根据特定模式的需要激活和使用。

图 30-1: MCCP/SCCP 概念框图



30.2 寄存器

每个 MCCP/SCCP 模块最多具有 7 个控制和状态寄存器：

- CCPxCON1（寄存器 30-1）控制所有模式共有的多种特性，包括所有模式的输入时钟选择、时基预分频、定时器同步、触发模式操作和后分频器选择。此外，也通过该寄存器使能模块和选择其工作模式。
- CCPxCON2（寄存器 30-2）控制自动关断和重启操作（主要用于 PWM 操作），也配置其他输入捕捉和输出比较功能，并配置辅助输出操作。
- CCPxCON3（寄存器 30-3）控制多个输出 PWM 死区以及输出比较和 PWM 模式的输出，并配置 MCCP 模块的 PWM 输出模式。
- CCPxSTAT（寄存器 30-4）包含一些指示模块操作状态的只读状态位。

每个模块还包含 4 个缓冲寄存器 / 计数器寄存器，用作定时器值寄存器或数据存放缓冲寄存器：

- CCPxTMR 是 32 位 CCPx 定时器 / 计数器寄存器
- CCPxPR 是 32 位 CCPx 定时器周期寄存器
- CCPxR 是用于输出比较操作的 32 位 CCPx 主数据缓冲寄存器
- CCPxBUF 是 32 位 CCPx 缓冲寄存器，用于输入捕捉 FIFO 操作

表 30-1: 捕捉 / 比较 / PWM / 定时器 (MCCP 和 SCCP) SFR 汇总

名称	位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
CCPxCON1	31:24	OPSSRC	RTRGEN	—	—	OPS<3:0>			
	23:16	TRIGEN	ONESHOT	ALTSYNC	SYNC<4:0>				
	15:8	ON	—	SIDL	CCPSLP	TMRSYNC	CLKSEL<2:0>		
	7:0	TMRPS<1:0>		T32	CCSEL	MOD<3:0>			
CCPxCON2	31:24	OENSYNC	—	OCFEN	OCEEN	OCDEN	OCCEN	OCBEN	OCAEN
	23:16	ICGSM<1:0>		—	AUXOUT<1:0>		ICS<2:0>		
	15:8	PWMRSEN	ASDGM	—	SSDG	—	—	—	—
	7:0	ASDG<7:0>							
CCPxCON3	31:24	OETRIG	OSCNT<2:0>			—	OUTM<2:0>		
	23:16	—	—	POLACE	POLBDF	PSSACE<1:0>		PSSBDF<1:0>	
	15:8	—	—	—	—	—	—	—	—
	7:0	—	—	DT<5:0>					
CCPxSTAT	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	PRLWIP	TMRHWIP	TMRLWIP	RBWIP	RAWIP
	15:8	—	—	—	—	—	ICGARM	—	—
	7:0	CCPTRIG	TRSET	TRCLR	ASEVT	SCEVT	ICDIS	ICOV	ICBNE
CCPxTMR	31:24	TMRH<15:8>							
	23:16	TMRH<7:0>							
	15:8	TMRL<15:8>							
	7:0	TMRL<7:0>							
CCPxPR	31:24	PRH<15:8>							
	23:16	PRH<7:0>							
	15:8	PRL<15:8>							
	7:0	PRL<7:0>							
CCPxRA	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	CMPA<15:8>							
	7:0	CMPA<7:0>							
CCPxRB	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	CMPB<15:8>							
	7:0	CMPB<7:0>							
CCPxBUF	31:24	BUFH<15:8>							
	23:16	BUFH<7:0>							
	15:8	BUFL<15:8>							
	7:0	BUFL<7:0>							

第 30 章 捕捉 / 比较 / PWM / 定时器 (MCCP 和 SCCP)

寄存器 30-1: CCPxCON1: 捕捉 / 比较 / PWMx 控制 1 寄存器

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
OPSSRC ⁽¹⁾	RTRGEN ⁽²⁾	—	—	OPS<3:0> ⁽³⁾			
bit 31							bit 24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TRIGEN	ONESHOT	ALTSYNC	SYNC<4:0> ⁽⁴⁾				
bit 23							bit 16
R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ON	—	SIDL	CCPSLP	TMRSYNC	CLKSEL<2:0> ⁽⁵⁾		
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TMRPS<1:0>		T32	CCSEL	MOD<3:0>			
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = POR 时的值	1 = 置 1	0 = 清零
		x = 未知

- bit 31 **OPSSRC:** 输出后分频器源选择位 ⁽¹⁾
 1 = 输出后分频器对特殊事件触发输出事件进行分频
 0 = 输出后分频器对定时器中断事件进行分频
- bit 30 **RTRGEN:** 重新触发使能位 ⁽²⁾
 1 = 当 CCPTRIG = 1 时可以重新触发时基
 0 = 当 CCPTRIG = 1 时不能重新触发时基
- bit 29-28 **未实现:** 读为 0
- bit 27-24 **OPS<3:0>:** CCPx 中断输出后分频比选择位 ⁽³⁾
 1111 = 在每次发生第 16 次时基周期匹配时产生中断
 1110 = 在每次发生第 15 次时基周期匹配时产生中断
 ...
 0100 = 在每次发生第 5 次时基周期匹配时产生中断
 0011 = 在每次发生第 4 次时基周期匹配或第 4 次输入捕捉事件时产生中断
 0010 = 在每次发生第 3 次时基周期匹配或第 3 次输入捕捉事件时产生中断
 0001 = 在每次发生第 2 次时基周期匹配或第 2 次输入捕捉事件时产生中断
 0000 = 在每次发生时基周期匹配或输入捕捉事件时产生中断
- bit 23 **TRIGEN:** CCPx 触发使能位
 1 = 使能定时器的触发操作
 0 = 禁止定时器的触发操作
- bit 22 **ONESHOT:** 单脉冲模式使能位
 1 = 使能单脉冲触发模式; 触发持续时间由 OSCNT<2:0> 设置
 0 = 禁止单脉冲触发模式

- 注 1:** 该控制位在输入捕捉模式下没有任何作用。
- 2:** 当 TRIGEN = 0 时, 该控制位没有任何作用。
- 3:** 在输入捕捉模式下, 大于 0011 的值会导致 FIFO 缓冲区发生溢出。
- 4:** 关于具体器件系列的同步源, 请参见器件数据手册。
- 5:** 关于具体器件系列的可用时钟源, 请参见器件数据手册。
- 6:** 关于备用同步输出信号, 请参见第 30.4.1 节 “备用同步输出”。

寄存器 30-1: CCPxCON1: 捕捉 / 比较 / PWMx 控制 1 寄存器 (续)

bit 21	ALTSYNC: CCPx 时钟选择位 1 = 使用备用信号作为模块同步输出信号 (6) 0 = 模块同步输出信号为时基复位 / 计满返回事件
bit 20-16	SYNC<4:0>: CCPx 同步源选择位 (4) 11111 = 定时器处于自由运行模式, 并在 FFFFh 处计满返回 (忽略周期寄存器) 11110 = 定时器与源 #30 进行同步 ... 00001 = 时基与源 #1 进行同步 00000 = 无外部同步; 定时器在 FFFFh 处计满返回或与周期寄存器匹配
bit 15	ON: CCPx 模块使能位 1 = 使用由 MOD<3:0> 指定的工作模式使能模块 0 = 禁止模块
bit 14	未实现: 读为 0
bit 13	SIDL: CCPx 空闲模式停止位 1 = 当器件进入空闲模式时, 模块停止工作 0 = 模块在空闲模式下继续工作
bit 12	CCPSLP: CCPx 休眠模式使能位 1 = 模块在休眠模式下继续工作 0 = 模块在休眠模式下不工作
bit 11	TMRSYNC: 时基时钟同步位 1 = 模块时基时钟与内部系统时钟同步; 时间限制适用 0 = 模块时基时钟不与内部系统时钟同步
bit 10-8	CLKSEL<2:0>: CCPx 时基时钟选择位 (5) 111 = 时钟 7 110 = 时钟 6 101 = 时钟 5 100 = 时钟 4 011 = 时钟 3 010 = 时钟 2 001 = 时钟 1 000 = 系统时钟 (Tcy)
bit 7-6	TMRPS<1:0>: CCPx 时基预分频比选择位 11 = 1:64 预分频比 10 = 1:16 预分频比 01 = 1:4 预分频比 00 = 1:1 预分频比
bit 5	T32: 32 位时基选择位 1 = 对于定时器使用 32 位时基, 单边沿输出比较或输入捕捉功能 0 = 对于定时器使用 16 位时基, 单边沿输出比较或输入捕捉功能
bit 4	CCSEL: 捕捉 / 比较模式选择位 1 = 输入捕捉模式 0 = 输出比较 / PWM 或定时器模式 (确切功能由 MOD<3:0> 位选择)

- 注 1:** 该控制位在输入捕捉模式下没有任何作用。
- 2:** 当 TRIGEN = 0 时, 该控制位没有任何作用。
- 3:** 在输入捕捉模式下, 大于 0011 的值会导致 FIFO 缓冲区发生溢出。
- 4:** 关于具体器件系列的同步源, 请参见器件数据手册。
- 5:** 关于具体器件系列的可用时钟源, 请参见器件数据手册。
- 6:** 关于备用同步输出信号, 请参见第 30.4.1 节 “备用同步输出”。

寄存器 30-1: CCPxCON1: 捕捉 / 比较 / PWMx 控制 1 寄存器 (续)

bit 3-0 **MOD<3:0>**: CCPx 模式选择位

CCSEL = 1 (输入捕捉模式):

- 1xxx = 保留
- 011x = 保留
- 0101 = 在每第 16 个上升沿进行捕捉
- 0100 = 在每第 4 个上升沿进行捕捉
- 0011 = 在每个上升沿和下降沿进行捕捉
- 0010 = 在每个下降沿进行捕捉
- 0001 = 在每个上升沿进行捕捉
- 0000 = 在每个上升沿和下降沿进行捕捉 (边沿检测模式)

CCSEL = 0 (输出比较模式):

- 1111 = 外部输入模式: 禁止脉冲发生器, 源通过 ICS<2:0> 选择
- 1110 = 保留
- 110x = 保留
- 10xx = 保留
- 0111 = 可变频率脉冲模式
- 0110 = 中心对齐脉冲比较模式, 进行缓冲
- 0101 = 双边沿比较模式, 进行缓冲
- 0100 = 双边沿比较模式
- 0011 = 16 位 /32 位单边沿模式: 发生比较匹配时翻转输出
- 0010 = 16 位 /32 位单边沿模式: 发生比较匹配时将输出驱动为低电平
- 0001 = 16 位 /32 位单边沿模式: 发生比较匹配时将输出驱动为高电平
- 0000 = 16 位 /32 位定时器模式: 禁止输出功能

- 注 1:** 该控制位在输入捕捉模式下没有任何作用。
- 2:** 当 TRIGEN = 0 时, 该控制位没有任何作用。
- 3:** 在输入捕捉模式下, 大于 0011 的值会导致 FIFO 缓冲区发生溢出。
- 4:** 关于具体器件系列的同步源, 请参见器件数据手册。
- 5:** 关于具体器件系列的可用时钟源, 请参见器件数据手册。
- 6:** 关于备用同步输出信号, 请参见第 30.4.1 节 “备用同步输出”。

寄存器 30-2: CCPxCON2: 捕捉 / 比较 / PWMx 控制 2 寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
OENSYNC	—	OCFEN ⁽¹⁾	OCEEN ⁽¹⁾	OCDEN ⁽¹⁾	OCCEN ⁽¹⁾	OCBEN ⁽¹⁾	OCAEN
bit 31							bit 24

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ICGSM<1:0>		—	AUXOUT<1:0> ⁽²⁾		ICS<2:0> ⁽³⁾		
bit 23							bit 16

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0
PWMRSEN	ASDGM	—	SSDG	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ASDG<7:0> ⁽⁴⁾							
bit 7							bit 0

图注:							
R = 可读位	W = 可写位	U = 未实现位, 读为 0					
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知				

- bit 31 **OENSYNC:** 输出使能同步位
 1 = 由输出使能位产生的更新在下次时基复位或计满返回时发生
 0 = 由输出使能位产生的更新立即发生
- bit 30 **未实现:** 读为 0
- bit 29-24 **OC<F:A>EN:** 输出使能 / 转向控制位 ⁽¹⁾
 1 = OCx 引脚由 CCPx 模块控制, 并产生输出比较或 PWM 信号
 0 = OCx 引脚不由 CCPx 模块控制; 引脚可供端口逻辑或引脚上多路复用的另一个外设使用
- bit 23-22 **ICGSM<1:0>:** 输入捕捉门控源模式控制位
 11 = 保留
 10 = 单脉冲模式: 来自门控源的下降沿禁止以后的捕捉事件 (ICDIS = 1)
 01 = 单脉冲模式: 来自门控源的上升沿使能以后的捕捉事件 (ICDIS = 0)
 00 = 电平敏感模式: 来自门控源的高电平将使能以后的捕捉事件; 低电平将禁止以后的捕捉事件
- bit 21 **未实现:** 读为 0
- bit 20-19 **AUXOUT<1:0>:** 辅助输出信号事件选择位 ⁽²⁾
 11 = 输入捕捉或输出比较事件; 在定时器模式下无信号
 10 = 信号输出取决于模块工作模式 (见表 30-3)
 01 = 时基计满返回事件 (所有模式)
 00 = 禁止

注 1: OCFEN 至 OCBEN (bit<29:25>) 仅在 MCCP 模块中实现。
 注 2: 辅助输出并未在所有器件上实现。详情请参见器件数据手册。
 注 3: 关于具体的输入捕捉源, 请参见器件数据手册。
 注 4: 关于具体器件系列实现的门控源, 请参见具体器件数据手册。

寄存器 30-2: CCPxCON2: 捕捉 / 比较 / PWMx 控制 2 寄存器 (续)

bit 18-16	ICS<2:0> : 输入捕捉源选择位 ⁽³⁾ 111 = 捕捉源 8 110 = 捕捉源 7 101 = 捕捉源 6 100 = 捕捉源 5 011 = 捕捉源 4 010 = 捕捉源 3 001 = 捕捉源 2 000 = 捕捉源 1 (ICx 引脚)
bit 15	PWMRSEN : CCPx PWM 重启使能位 1 = 在关断输入结束之后, ASEVT 位在下一个 PWM 周期开始时自动清零 0 = 必须用软件清零 ASEVT 位, 才会恢复输出引脚上的 PWM 活动
bit 14	ASDGM : CCPx 自动关断门控模式使能位 1 = 等到下一次时基复位或计满返回时才发生关断 0 = 立即发生关断事件
bit 13	未实现 : 读为 0
bit 12	SSDG : CCPx 软件关断 / 门控控制位 1 = 手动强制自动关断、定时器时钟门控或输入捕捉信号门控事件 (ASDGM 位的设置仍然适用) 0 = 模块正常工作
bit 11-8	未实现 : 读为 0
bit 7-0	ASDG<7:0> : CCPx 自动关断 / 门控源使能位 ⁽⁴⁾ 1 = 使能自动关断 / 门控源 n 0 = 禁止自动关断 / 门控源 n

- 注 1: OCFEN 至 OCBEN (bit<29:25>) 仅在 MCCP 模块中实现。
- 2: 辅助输出并未在所有器件上实现。详情请参见器件数据手册。
- 3: 关于具体的输入捕捉源, 请参见器件数据手册。
- 4: 关于具体器件系列实现的门控源, 请参见具体器件数据手册。

PIC32 系列参考手册

寄存器 30-3: CCPxCON3: 捕捉 / 比较 / PWMx 控制 3 寄存器

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
OETRIG	OSCNT<2:0>		—	—	OUTM<2:0> ⁽¹⁾		
bit 31							bit 24

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	POLACE	POLBDF ⁽¹⁾	PSSACE<1:0>		PSSBDF<1:0> ⁽¹⁾	
bit 23							bit 16

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DT<5:0> ⁽¹⁾					
bit 7							bit 0

图注:							
R = 可读位	W = 可写位	U = 未实现位, 读为 0					
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知				

bit 31 **OETRIG:** PWM 死区选择位
 1 = 对于触发模式 (TRIGEN = 1), 只有触发之后模块才会驱动已使能的输出引脚
 0 = 正常输出引脚操作

bit 30-28 **OSCNT<2:0>:** 单脉冲事件计数位
 将单脉冲触发事件的持续时间额外延长 n 个时钟周期 (共 n + 1 个周期)
 111 = 7 个定时器计数周期 (共 8 个周期)
 110 = 6 个定时器计数周期 (共 7 个周期)
 101 = 5 个定时器计数周期 (共 6 个周期)
 100 = 4 个定时器计数周期 (共 5 个周期)
 011 = 3 个定时器计数周期 (共 4 个周期)
 010 = 2 个定时器计数周期 (共 3 个周期)
 001 = 1 个定时器计数周期 (共 2 个周期)
 000 = 不延长单脉冲触发事件 (事件需要 1 个定时器计数周期)

bit 27 **未实现:** 读为 0

bit 26-24 **OUTM<2:0>:** PWMx 输出模式控制位 ⁽¹⁾
 111 = 保留
 110 = 输出扫描模式
 101 = 有刷直流输出模式, 正向
 100 = 有刷直流输出模式, 反向
 011 = 保留
 010 = 半桥输出模式
 001 = 推挽输出模式
 000 = 可转向单输出模式

bit 23-22 **未实现:** 读为 0

bit 21 **POLACE:** CCPx 输出引脚 OCxA、OCxC 和 OCxE 极性控制位
 1 = 输出引脚极性为低电平有效
 0 = 输出引脚极性为高电平有效

注 1: 这些位仅在 MCCP 模块中实现。

寄存器 30-3: CCPxCON3: 捕捉 / 比较 / PWMx 控制 3 寄存器 (续)

bit 20	POLBDF: CCPx 输出引脚 OCxB、OCxD 和 OCxF 极性控制位 ⁽¹⁾ 1 = 输出引脚极性为低电平有效 0 = 输出引脚极性为高电平有效
bit 19-18	PSSACE<1:0>: PWMx 输出引脚 OCxA、OCxC 和 OCxE 关断状态控制位 11 = 引脚在发生关断事件时驱动为有效 10 = 引脚在发生关断事件时驱动为无效 0x = 引脚在发生关断事件时处于高阻态
bit 17-16	PSSBDF<1:0>: PWMx 输出引脚 OCxB、OCxD 和 OCxF 关断状态控制位 ⁽¹⁾ 11 = 引脚在发生关断事件时驱动为有效 10 = 引脚在发生关断事件时驱动为无效 0x = 引脚在发生关断事件时处于高阻态
bit 15-6	未实现: 读为 0
bit 5-0	DT<5:0>: PWM 死区选择位 ⁽¹⁾ 111111 = 在互补输出信号之间插入 63 个死区延时周期 111110 = 在互补输出信号之间插入 62 个死区延时周期 ... 000010 = 在互补输出信号之间插入 2 个死区延时周期 000001 = 在互补输出信号之间插入 1 个死区延时周期 000000 = 禁止死区逻辑

注 1: 这些位仅在 MCCP 模块中实现。

寄存器 30-4: CCPxSTAT: 捕捉 / 比较 / PWMx 状态寄存器 (续)

- bit 5 **TRCLR:** CCPx 触发清零请求位
在 TRIGEN = 1 时, 向该单元写入 1 可取消定时器触发 (该单元总是读为 0)。
- bit 4 **ASEVT:** CCPx 自动关断事件状态 / 控制位
1 = 关断事件正在进行中; CCPx 输出处于关断状态
0 = CCPx 输出正常工作
- bit 3 **SCEVT:** 单边沿比较事件状态位
1 = 发生了单边沿比较事件
0 = 未发生单边沿比较事件
- bit 2 **ICDIS:** 输入捕捉禁止位
1 = 输入捕捉引脚上的事件不产生捕捉事件
0 = 输入捕捉引脚上的事件产生捕捉事件
- bit 1 **ICOV:** 输入捕捉缓冲区溢出状态位
1 = 输入捕捉 FIFO 缓冲区已溢出
0 = 输入捕捉 FIFO 缓冲区未溢出
- bit 0 **ICBNE:** 输入捕捉缓冲区状态位
1 = 输入捕捉缓冲区具有可用数据
0 = 输入捕捉缓冲区为空

注 1: 它不是一个物理位单元, 总是读为 0。写入 1 会启动硬件事件。

30.3 时基发生器

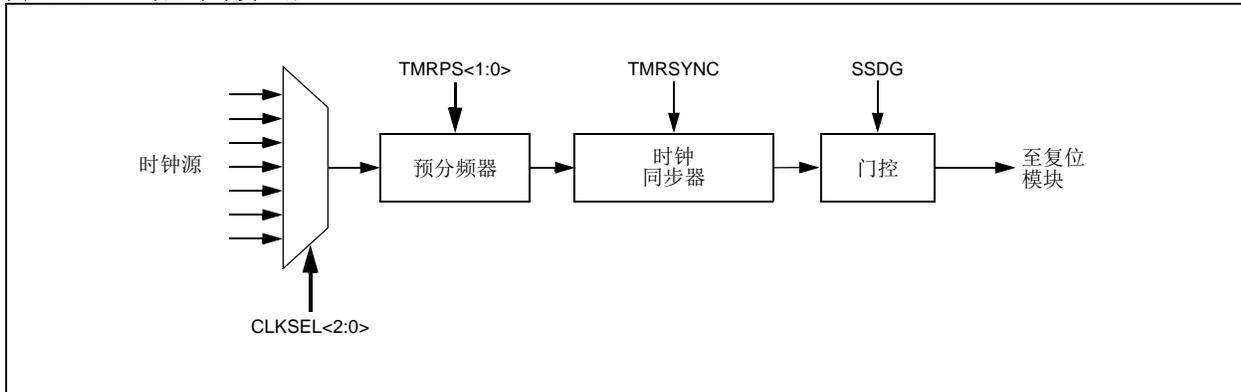
时基发生器（Time Base Generator, TBG）使用单片机上的时钟信号，为模块的其余部分提供时基。它不仅用作定时器模式的时基，而且使输入捕捉和输出比较预模式可以无需依赖另一个片上定时器模块而工作。

最多有 8 个时钟输入可供时钟发生器使用，包括系统时钟（Tcy）和其他片上振荡器源。根据不同的器件，可能还有外部时钟输入可供使用。预分频器可以将选定时钟源分频为合适的频率供模块使用。

TBG 具有将其操作与选定时钟源进行同步的能力，但会受到输入时序或模块工作条件的限制。将 TMRSYNC 位（CCPxCON1<11>）置 1 时，使能时基与输入时钟进行同步。

TBG 如图 30-2 所示。

图 30-2: 时基时钟发生器



30.3.1 门控逻辑

时基发生器具有一个可以禁止定时器递增时钟送至定时器门的硬件门，它仅在定时器模式下可用。

门控功能使用 ASDG<7:0> 控制位（CCPxCON2<7:0>）和 SSDG 位（CCPxCON2<12>）控制。所有这些位一起进行逻辑或运算，以产生 TBG 的门控使能信号。

将任一 ASDGx 位置 1 都会使其相应的硬件触发器；可以通过将其中部分或全部位置 1 来选择多个源。可用的门控和自动关断源取决于器件，并且通常包括诸如比较器输出、I/O 引脚（包括用于 PWM 操作的 OCFA 和 OCFB）、软件控制等的源。来自任何已使能源的输出信号都会禁止 TBG 输出。事件通常是电平敏感型，而不是边沿触发型。

SSDG 位仅仅是一个可以用软件操作的门控源。将 SSDG 置 1 与来自任何硬件源的输入具有相同的效果。

以下几节对门控功能进行了介绍：

- 定时器门控（见第 30.6.3 节“定时器模式的时钟门控”）
- 输出比较的自动关断，MCCP 模块（见第 30.7.7.11 节“自动关断控制”）
- 门控输入捕捉（见第 30.8.4.1 节“输入捕捉信号门控”）

无论工作模式如何，CCP 模块都不是基于门控输入状态来产生中断事件的。如果需要对于某种门控事件产生中断，则必须使用门控源本身来产生中断。

30.4 模块同步输出

默认情况下，MCCP/SCCP 模块基于 CCPxTMR 寄存器的计满返回事件来产生 CCP 同步信号。该信号可供所有其他 CCP 模块用作同步源或触发另一个外设。CCP 同步信号独立于模块的器件级中断或其他输出。

在一些情况下，可能可以基于另一个事件信号来更好地产生 CCP 同步信号，或者需要使用选定信号之外的其他事件输出。CCP 模块包含了一些用户配置选项来处理这些情况。

30.4.1 备用同步输出

通过 ALTSYNC 控制位 (CCPxCON1<21>)，用户可以使用其他同步 / 触发输出代替定时器计满返回事件来产生 CCP 同步信号。当 ALTSYNC = 0 时，在所有工作模式下，CCP 同步输出都是默认的定时器计满返回信号。当 ALTSYNC = 1 时，同步信号取决于特定工作模式。表 30-2 列出了可用备用输出。

表 30-2: 备用同步输出信号

ALTSYNC	CCSEL	MOD<3:0>	输出信号
0	x	全部	标准 (默认) CCP 同步输出
1	0	0000	特殊事件触发输出 (定时器)
1	0	全部 (0000 除外)	输出比较中断事件 (比较)
1	1	全部	输入捕捉事件 (捕捉)

30.4.2 辅助输出信号

MCCP 和 SCCP 模块还可以产生一个不同于 CCP 同步信号 (或其备用信号，如果 ALTSYNC 置 1) 的辅助输出。辅助输出旨在使其他数字外设可以访问内部 CCP 模块信号，例如：

- 时基同步
- 外设触发信号和时钟输入
- 信号门控

输出信号的类型使用 AUXOUT<1:0> 控制位 (CCPxCON2<20:19>) 选择，并且取决于模块工作模式。与备用同步输出相比，每种模式提供了更多的选项，如表 30-3 所示。

并非所有 CCP 模块版本都具有辅助输出功能。详情请参见具体器件的数据手册。

表 30-3: 辅助输出信号

AUXOUT<1:0>	CCSEL	MOD<3:0>	输出信号
00	x	xxxx	禁止 (无输出)
01	0	0000 (定时器模式)	时基周期复位或计满返回
10			特殊事件触发输出
11			无输出
01	0	0001 至 1111 (输出比较模式)	时基周期复位或计满返回
10			输出比较事件信号
11			输出比较信号
01	1	xxxx (输入捕捉模式)	时基周期复位或计满返回
10			反映 ICDIS 位的值
11			输入捕捉事件信号

30.5 同步和触发操作

在模块的大多数主要工作模式下，同步（Sync）和触发模式操作可以视为影响 CCPxTMR 寄存器操作的两种互补模式。两者均使用 SYNC<4:0> 位（CCPxCON1<20:16>）来确定输入信号源。不同之处在于信号如何影响定时器。

在同步模式操作中，通过 ON 位使能时，定时器将自由计数，并在由 SYNC<4:0> 选择的输入置为有效时复位为零。定时器将立即重新从零开始计数，除非由于某种其他原因而受阻。每当 TRIGEN 位（CCPxCON1<23>）清零时，就将使用同步操作。

在触发模式操作中，定时器将保持在复位状态，直到将 SYNC<4:0> 选择的输入置为有效为止。发生这种情况时，定时器将开始计数并一直继续到 TRCLR 位（CCPxSTAT<5>）置 1 为止。每当 TRIGEN 位置 1 时，就将使用触发操作。

根据具体的器件，SYNC<4:0> 位最多允许选择 32 个内部或外部源。一些已实现的源可能可用于触发模式操作，但不能用于同步模式操作。此外，11111（自由运行计数器）对于同步操作是无效的。关于详细信息，请参见器件数据手册。

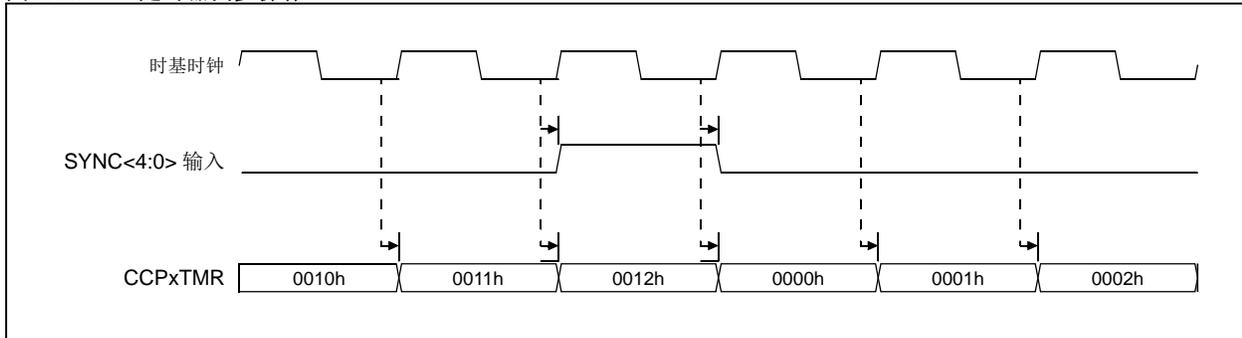
同步和触发模式操作支持对多个模块进行链式同步操作，在模块的定时器和输出比较模式中发挥着重要的作用。

30.5.1 定时器同步操作

在同步操作中，可以使用由 SYNC<4:0> 选择的同步 / 触发输入将定时器与其他模块进行同步。基本操作如图 30-3 所示。每当选定的同步输入置为有效（高电平）时，定时器将在时基信号的下一个上升沿沿计满返回至 0h。

当 TRIGEN（CCPxCON1<23>）清零，并且 SYNC<4:0> 位具有除 11111 之外的任何其他值时，定时器在同步操作下工作。CCPTRIG 位（CCPxSTAT<7>）没有任何作用。

图 30-3: 定时器同步操作



SYNCx 位的值选择为 00000 时，模块将作为一个周期定时器运行，在定时器寄存器与 CCPxPR 的值匹配时自动计满返回到 0h。11111 的值会导致模块以自由运行定时器的方式运行，它将在定时器寄存器达到其最大计数（1）时计满返回。

SYNC<4:0> 位的值为除 00000 或 11111 之外的其他值时，定时器将在由 SYNCx 位选择的输入置为有效时复位。关于特定于器件的可用输入，请参见器件数据手册。

例 30-1: 同步操作设置 (16 位双定时器模式)

```
CCP1CON1bits.TRIGEN = 0;      // Set Sync/Triggered mode (Synchronous Mode)
CCP1CON1bits.SYNC = 0;        // rolls over at FFFFh or match
                                // with period register (self sync)
CCP1CON1bits.T32 = 0;         // 16-bit dual timer mode
CCP1CON1bits.TMRSYNC = 0;     // Set timebase synchronization (Synchronized)
CCP1CON1bits.CLKSEL = 0;      // Set the clock source (Tcy)
CCP1CON1bits.TMRPS = 0;       // Set the clock prescaler (1:1)
CCP1PRbits.PRL = 0X0FFF;      // 16-bit MCCP1 low period bits
CCP1PRbits.PRH = 0X0FFF;      // 16-bit MCCP1 high period bits
CCP1CON1bits.ON = 1;          // Start the Timer
```

30.5.1.1 同步多个模块

每个 CCP 模块都可以产生 CCP 同步输出信号（见第 30.4 节“模块同步输出”），该信号可用于将其操作与其他模块进行同步。该信号不同于模块的中断或任何其他输出信号。所有 CCP 模块都可以通过 SYNC<4:0> 位访问彼此的同步信号；因此，可以将几个模块链接在一起，进行更复杂的同步操作。

图 30-4 给出了同步操作的一个简单示例。在该示例中，将 MCCP2 与 MCCP1 进行同步。每个模块配置为使用相同的时钟源作为其时基。此外，两个模块均使用来自 MCCP1 的 CCP 同步信号作为其同步源输入。CCP1PR 现在同时用作 MCCP1 和 MCCP2 的周期寄存器。

图 30-5 显示了这两个模块之间的时序关系。当 CCP1TMR 和 CCP1PR 之间发生匹配时，同步信号会变为有效。这会导致 CCP1 和 CCP2 中的定时器在下一个定时器输入时钟上升沿变为 0h。

在同步模块时，需要牢记两个重要事项：

- 所有同步的模块都要使用相同的时钟源作为其时基。
- 当初始化同步的模块时，用作同步源的模块应最后使能。这可以确保所有要进行同步的模块的定时器都保持在复位状态，直到最后一个模块初始化为止。

图 30-4: 两个同步定时器的示例 (将 MCCP2 与 MCCP1 进行同步)

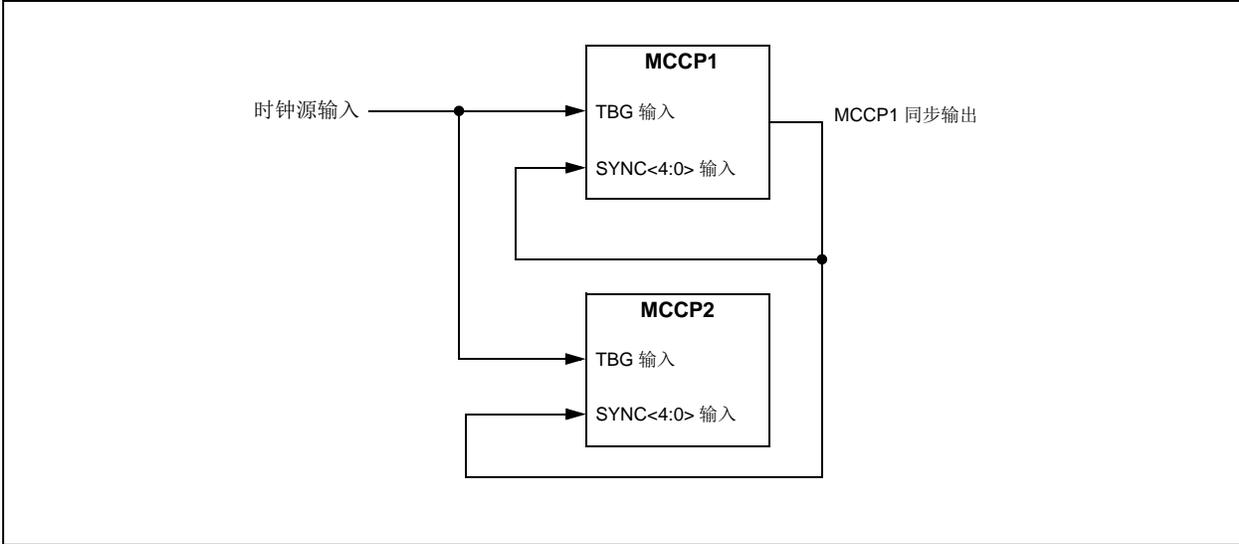
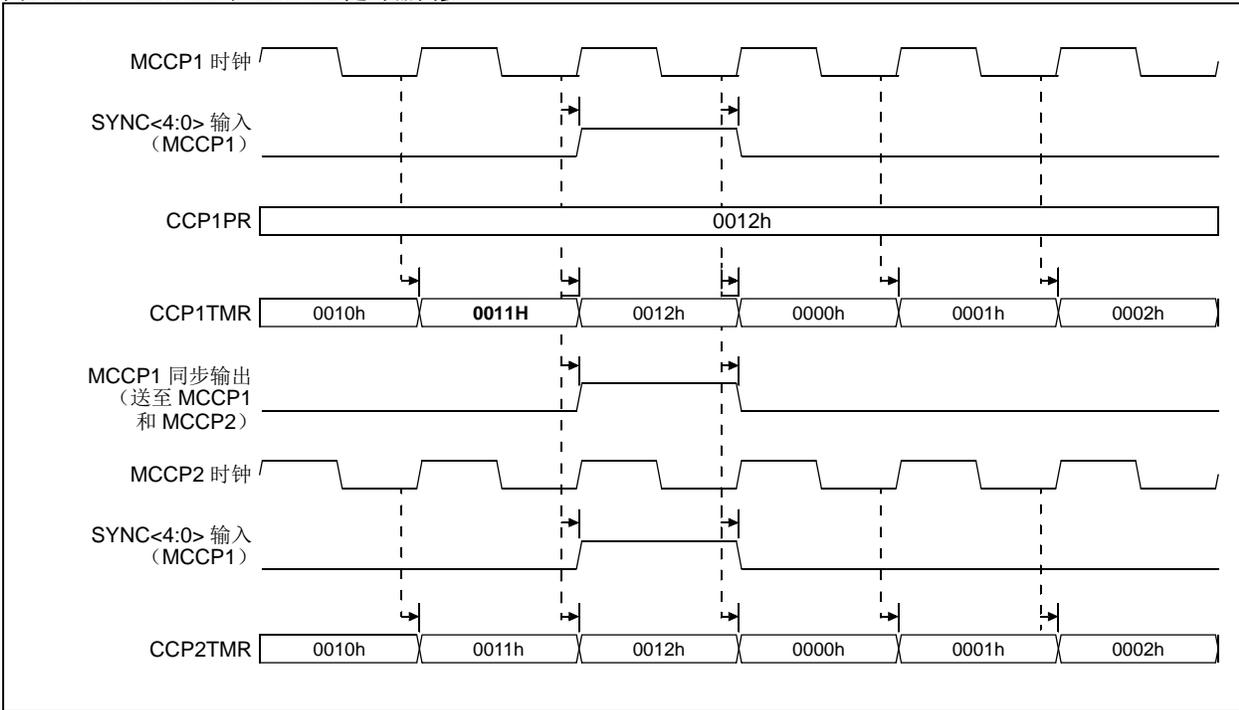


图 30-5: MCCP1 和 MCCP2 定时器同步



30.5.2 定时器触发操作

当 $TRIGEN = 1$ 时，即使能了定时器的触发操作。触发模式操作对于产生时间延时很有用。根据模块的工作模式，可以在延时之后产生一个脉冲或边沿事件。

当配置为触发操作时，模块定时器将保持复位状态，直到由 $SYNC<4:0>$ 选择的源产生触发事件为止。在发生触发事件之后，定时器开始计数。定时器在时基信号的每个正时钟递增。

如果定时器配置为 16 位双定时器操作 ($T32 = 0$)，则只有基于 $CCPxTMR<15:0>$ 的定时器会在触发操作中起作用。基于 $CCPxTMR<31:16>$ 的定时器将作为自由运行定时器工作。

$CCPTRIG$ 状态位 ($CCPxSTAT<7>$) 指示定时器是保持复位状态还是已被释放进行计数。当 $CCPTRIG = 0$ 时，定时器保持复位状态；当 $CCPTRIG = 1$ 时，定时器已被释放。

当在触发模式下工作时，存在两种触发条件：软 / 硬件和仅软件。软 / 硬件触发操作如图 30-6 所示。当使能模块进行触发响应时，定时器将保持复位状态。它会一直保持在该状态，直到 $SYNC<4:0>$ 输入的触发事件置为有效，这会在两个时钟周期内将 $CCPTRIG$ 位置 1。触发信号仅决定时基何时开始计数； $CCPxPR$ 寄存器将设置定时器的周期。不同于同步操作，通过 $SYNC<4:0>$ 位使用的所有触发源都可用于触发操作。

$CCPTRIG$ 可以在任意时刻手动置 1，定时器可以通过向 $TRSET$ 位 ($CCPxSTAT<6>$) 写入 1 从复位中释放。此外， $CCPTRIG$ 位还可以通过用软件向 $TRCLR$ 位 ($CCPxSTAT<5>$) 写入 1 来手动清零。

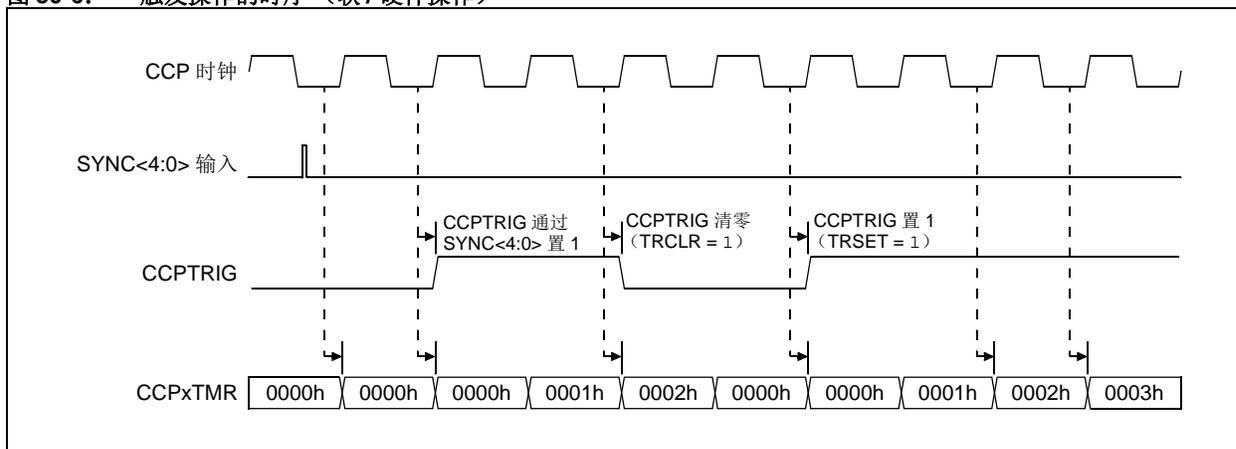
当 $SYNC<4:0> = 11111$ 时，将选择仅软件操作。在该配置下，唯一可以将 $CCPTRIG$ 位置 1 的方法是用软件写入 $TRSET$ 位。这种选择实际上会禁止所有外部硬件触发源。

当用软件清零 $TRIGEN$ 位时，定时器会在下一个定时器时钟的上升沿复位为 0h，并可用于下一次 $SYNC<4:0>$ 输入。

注： $TRSET$ 和 $TRCLR$ 位是始终读为 0 的只写位。向任一单元写入 0 没有任何作用。

例 30-2 给出了为模块配置触发操作的过程。

图 30-6: 触发操作的时序 (软 / 硬件操作)



例 30-2: 定时器触发操作设置 (16 位双定时器模式)

```

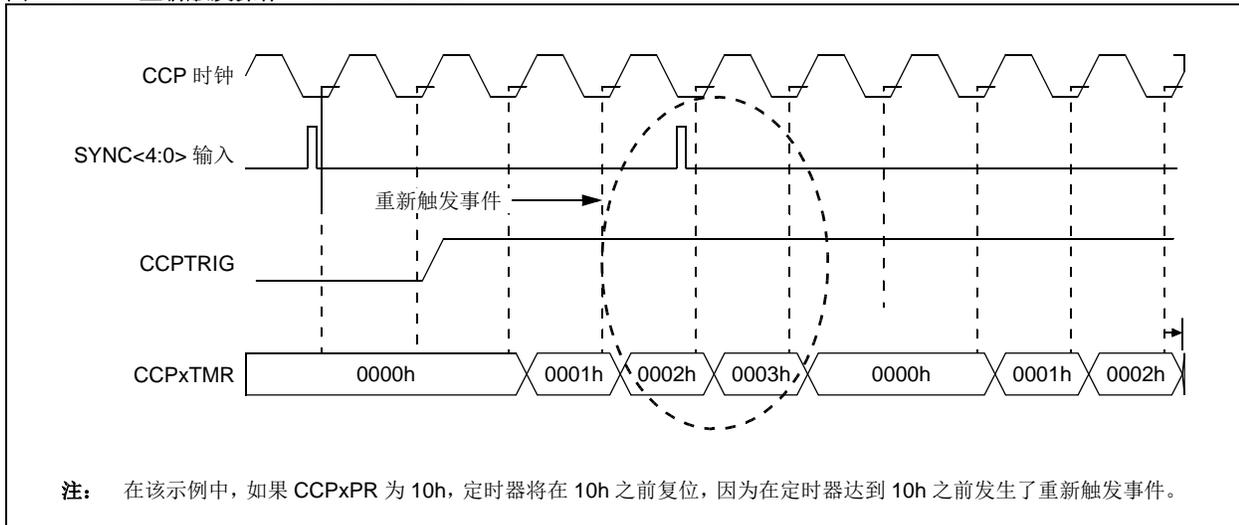
CCP1CON1bits.TRIGEN = 1; // Set Sync/Triggered mode (Triggered Mode)
CCP1CON1bits.SYNC = 0x08; // INT0 as trigger (verify the data sheet
// for Trigger source)
CCP1CON1bits.T32 = 0; // 16-bit dual timer mode
CCP1CON1bits.TMRSYNC = 0; // Set timebase synchronization (Synchronized)
CCP1CON1bits.CLKSEL = 0; // Set the clock source (Tcy)
CCP1CON1bits.TMRPS = 0; // Set the clock prescaler (1:1)
CCP1PRbits.PRL = 0X0FFF; // 16-bit MCCP1 low period bits
CCP1PRbits.PRH = 0X0FFF; // 16-bit MCCP1 high period bits
CCP1CON1bits.ON = 1; // Enable the Timer
    
```

30.5.2.1 重新触发操作

RTRGEN 位 (CCPxCON1<30>) 可在 CCPTRIG 位保持置 1 时重新触发定时器。当 RTRGEN 置 1 时, 触发操作期间发生的第二个触发事件将导致定时器发生复位并重新开始计数。图 30-7 说明了在定时器发生溢出之前, 其如何在触发事件再次出现时重新开始计数。

当 RTRGEN = 1 时, 将无法识别在同一时钟周期内发生的多个触发脉冲, 它们将被视为单个触发事件。如果在两个相邻定时器时钟周期接收到触发脉冲, 时基将另外再于一个额外时钟周期保持复位 (0h)。

图 30-7: 重新触发操作 (RTRGEN = 1)



30.5.2.2 使用异步时钟进行触发操作

模块时基可以使用各种时钟源工作; 它们可能会也可能不会与系统时钟同步。此外, 触发源可能是与模块时基时钟异步的。为了最大程度减少毛刺, 默认情况下, 传入的触发信号将进行锁存, 并与模块的时基时钟源进行同步。

当时基时钟源与系统时钟异步时, 在 CCPTRIG 状态位的值反映触发状态之前, 最多会有两个系统时钟周期的延时。

当时基时钟源与系统时钟异步时, 在来自软件的触发置 1 或清零请求影响模块的触发状态之前, 最多会有两个时基时钟周期的延时。

30.5.2.3 触发操作下的定时器计满返回

当模块配置为触发操作时，由 SYNC<4:0> 选择的信号源不会设置时基计数周期。触发信号的主要用途是指示定时器何时开始计数，而不是何时复位（正如在同步模式下）。

定时器在 CCPxPR 与 CCPxTMR 匹配之后的下一个时钟或在 CCPxTMR 达到其最大计数时（如果 CCPxPR 在模块的特定工作模式下不可用）计满返回至 0h。

SYNC<4:0> 有两个值在触发操作中不允许使用：

- 00000（同步定时器，禁止外部触发事件）
- 选择模块自己的 CCP 同步信号的任何值（触发事件必须来自外部）

如果先初始化并使能由 SYNC<4:0> 选择的触发源，则定时器有可能会错过触发事件。因此，建议在触发源之前先初始化和使能定时器。

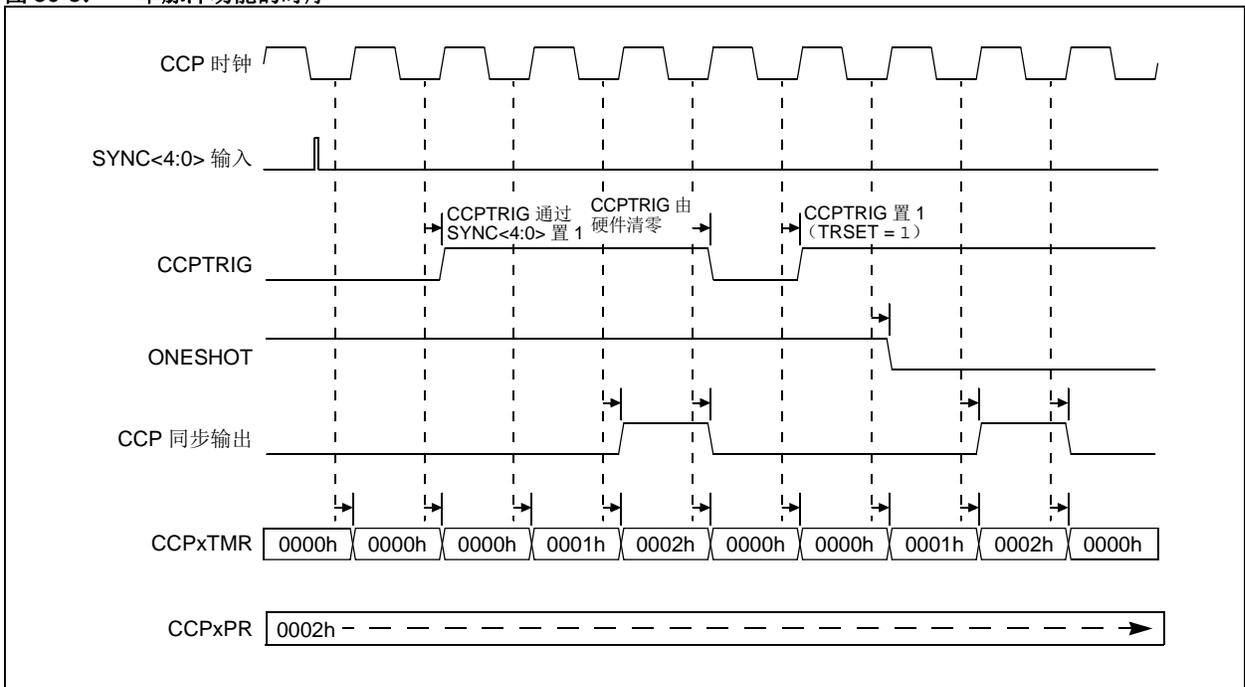
30.5.2.4 单脉冲功能

在触发操作中，定时器可以在单脉冲模式下工作。在该模式下，定时器保持在复位状态，直到发生硬件或软件触发事件。该事件会将 CCPTRIG 位置 1，然后定时器开始计数。当定时器计满返回至 00000000h 时，CCPTRIG 位会由硬件清零。这会将定时器保持在复位状态，直到发生下一次触发事件，从而构成了单脉冲定时器。

单脉冲模式通过将 ONESHOT 位 (CCPxCON1<22>) 置 1 来使能。OSCNT<2:0> 控制位 (CCPxCON3<30:28>) 使单脉冲触发事件可以延长至超过一个 CCP 定时器时钟周期。该功能非常有用，例如，当模块需要在一次触发事件中产生多个脉冲时。

注： 在模块被触发 (CCPTRIG = 1) 时，请不要修改 OSCNT<2:0>；可能会出现意外的结果。

图 30-8: 单脉冲功能的时序



30.6 定时器模式

当 $CCSEL = 0$ 且 $MOD<3:0> = 0000$ 时，模块用作一个定时器。存在两种基本的定时器模式，它们通过 T32 位（ $CCPxCON1<5>$ ）选择；表 30-4 列出了这些模式。在其中任一模式下，定时器可以作为自由运行定时器 / 计数器工作，与其他模块同步工作，或通过其他模块或外部事件触发。

表 30-4: 定时器工作模式

T32	工作模式
0	双定时器模式（16 位）
1	定时器模式（32 位）

30.6.1 双 16 位定时器模式

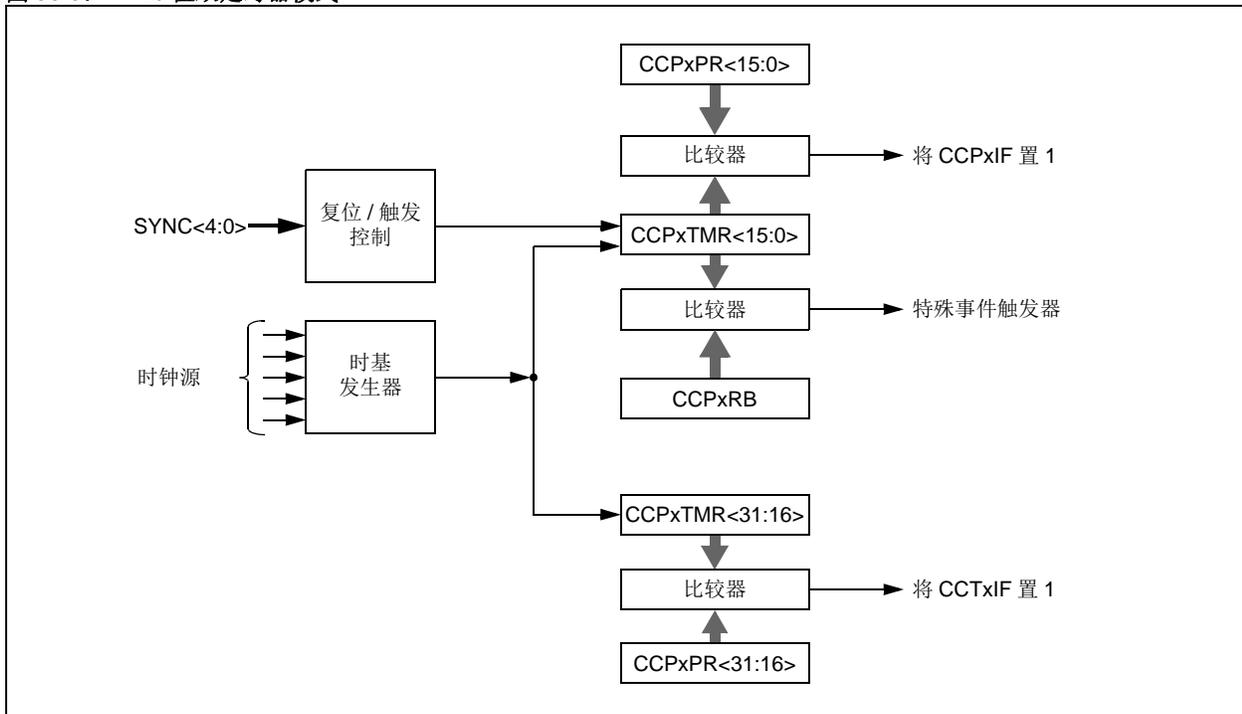
当 $T32 = 0$ 时，即选择双 16 位定时器模式。该模式对于以下功能很有用：

- 定期 CPU 中断
- 用于同步其他 CCP 模块的主时基功能
- 触发定期 ADC 转换
- 从休眠状态定期唤醒（如果相应时钟源可用）

注： 如果使用高速异步时钟源来为时基供应时钟，则用户可能无法读取 $CCPxTMR$ 寄存器。对于低速读操作，可以进行两次读操作，并对结果进行比较。

双 16 位定时器模式提供简单的定时器功能，具有两个独立的 16 位定时器 / 计数器。主定时器基于 $CCPxTMR$ 的低字，它可以完全正常工作，并且可以与器件上的其他模块进行交互。它可以产生由其他 MCCP 模块使用的 CCP 同步信号。此外，它还可以使用由其他模块产生的 $SYNC<4:0>$ 信号。辅助定时器基于 $CCPxTMR$ 的高字，功能有限。它仅可用作调度 CPU 事件的定期中断源。它不会产生类似于主时基的输出触发信号。

图 30-9: 16 位双定时器模式



主定时器和辅助定时器均使用来自 TBG 的相同时钟源，通过 CLKSEL<2:0> 进行选择。用户可通过 CCPxTMR 寄存器访问两个 16 位时基。两个低位和高位定时器寄存器 (CCPxTMR<15:0> 和 CCPxTMR<31:16>) 均基于定时器输入在同一时刻递增；但是，只有主定时器 (CCPxTMR<15:0>) 可以使用外部同步功能。辅助定时器 (CCPxTMR<31:16>) 不具有外部同步功能。

当 SYNC<4:0> = 00000 时，CCPxPR<15:0> 寄存器控制 16 位主时基的周期。当模块配置为使用外部同步源时，16 位主时基会在由 SYNC<4:0> 选择的源置为有效时复位。每当时基计满返回或复位为 0 时，就会产生模块的同步信号。

主定时器可以在 CCPxTMR<15:0> 值复位为 0000h 时产生 CCP 中断。当 SYNC<4:0> = 00000 时，如果 CCPxTMR<15:0> 位与 CCPxPR<15:0> 匹配，将发生这种情况。如果 SYNC<4:0> 不为 00000，则每当由 SYNC<4:0> 选择的信号置为有效时，CCPxTMR<15:0> 位会复位并产生 CCP 中断标志 (CCPxIF) 事件。

CCPxPR<31:16> 寄存器位控制 16 位辅助定时器的计数周期。辅助定时器不支持外部同步，不会受选定 SYNC<4:0> 输入影响。辅助时基在 ON 位 (CCPxCON1<15>) 置 1 时开始计数。当 CCPxPR<31:16> 寄存器位和 CCPxTMR<31:16> 计数值之间发生匹配时，16 位辅助时基会复位，并产生定时器计满返回中断事件 (CCTxIF)。

如果应用程序中不使用任何一个 16 位定时器，可以通过向相应的周期寄存器写入 0000h 来禁止定时器。定时器将保持在复位状态，只要周期寄存器的值为 0，就不会产生任何中断。CCPxPR<31:16> 和 CCPxPR<15:0> 寄存器位将不会在此工作模式下进行缓冲。

要在双 16 位定时器模式下使用模块：

1. 通过设置 CCSEL = 0 来选择模块的时基 / 输出比较模式。
2. 通过设置 T32 = 0 来选择 16 位时基操作。
3. 通过设置 MOD<3:0> = 0000 来选择时基模式。
4. 将 SYNC<4:0> 设置为所需的时基同步源：
 - 在使能定时器之前，先配置并使能由 SYNC<4:0> 选择的外部源。
 - 如果定时器不使用外部同步源 (SYNC<4:0> = 00000)，或如果模块与自身进行同步 (SYNC<4:0> 位选择模块自己的值作为同步源)，会将所需的 16 位主时基计数周期写入 CCPxPR<15:0>。
5. 如果同时也使用辅助定时器，则通过向 CCPxPR<31:16> 写入一个非零值来指定计数周期。
6. 如果正在使用特殊 ADC 触发，会将 CCPxRB 设置为所需的触发输出时间。
7. 通过将 ON 位置 1 使能该模块。
8. 如果在步骤 4 中选择了外部同步源，则配置并使能该源，使 16 位主时基开始计数。

30.6.1.1 特殊事件触发器

在双 16 位定时器模式下，主定时器还会产生一个特殊事件触发输出信号，该信号可以用于启动 ADC 转换和触发其他外设事件。触发周期由 CCPxRB 寄存器的值设置。通过 CCPxRB 设置的触发时间必须短于由 CCPxPR<15:0> 寄存器定义的计数器周期。

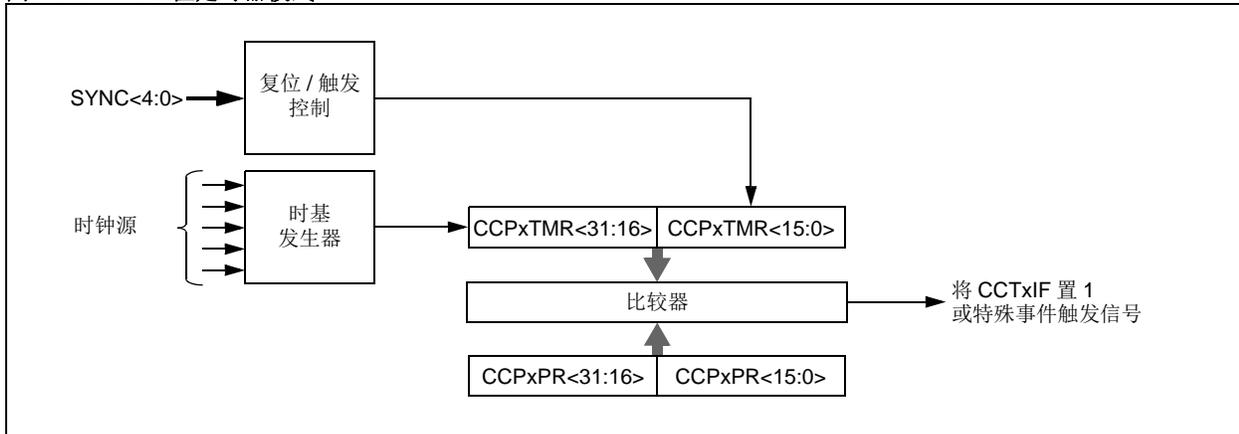
30.6.2 32 位定时器模式

当 $T32 = 1$ 时，将选择 32 位定时器模式。在该模式下， $CCPxTMR$ 寄存器将用作一个 32 位定时器。在需要跟踪长时间周期时，该模式可以提供简单的定时器功能。它对于以下功能很有用：

- 定期 CPU 中断
- 用于其他 CCP 模块的同步和触发信号产生
- 定期 ADC 转换触发
- 从休眠状态定期唤醒（如果相应时钟源可用）

在该工作模式下，CCP 模块无输入或输出功能可供使用。

图 30-10: 32 位定时器模式



未选择外部同步 ($SYNC<4:0> = 00000$) 时， $CCPxPR$ 寄存器用于设置定时器的计数周期。每当模块使能 ($ON = 1$) 时， $CCPxTMR$ 和 $CCPxPR$ 寄存器之间的匹配还会自动产生同步输出信号。

要在 32 位定时器模式下使用模块：

1. 通过设置 $CCSEL = 0$ 来选择模块的时基 / 输出比较模式。
2. 通过设置 $T32 = 1$ 来选择 32 位时基操作。
3. 通过设置 $MOD<3:0> = 0000$ 来选择时基模式。
4. 将 $SYNC<4:0>$ 设置为所需的定时器同步源：
 - 在使能定时器之前，先配置并使能由 $SYNC<4:0>$ 选择的外部源。
 - 如果定时器不使用外部同步源 ($SYNC<4:0> = 00000$)，或如果模块与自身进行同步 ($SYNC<4:0>$ 选择模块自己的值作为同步源)，会将所需的 16 位主时基计数周期写入 $CCPxPR$ 。
5. 通过将 ON 位置 1 使能该模块。

30.6.3 定时器模式的时钟门控

在定时器模式下工作时，可以使用时基门控来门控定时器的操作（关于更多信息，请参见第 30.3.1 节“门控逻辑”）。该功能提供了一种简单的方法来测量外部事件的时间。定时器时钟门控在一个或多个 $ASDG<7:0>$ 位 ($CCPxCON2<7:0>$) 置 1，或者在 $SSDG$ 位 ($CCPxCON2<12>$) 置 1 时使能。

30.7 输出比较和 PWM 模式

当 $CCSEL = 0$ 且 $MOD<3:0>$ 位为 0000 之外的任何其他值时，模块将在输出比较模式下工作。CCPxTMR 的值将与比较寄存器进行比较，具体取决于其工作模式。输出比较模式可以产生单个输出跃变或一系列输出脉冲，并且可以在发生比较匹配事件时产生中断。

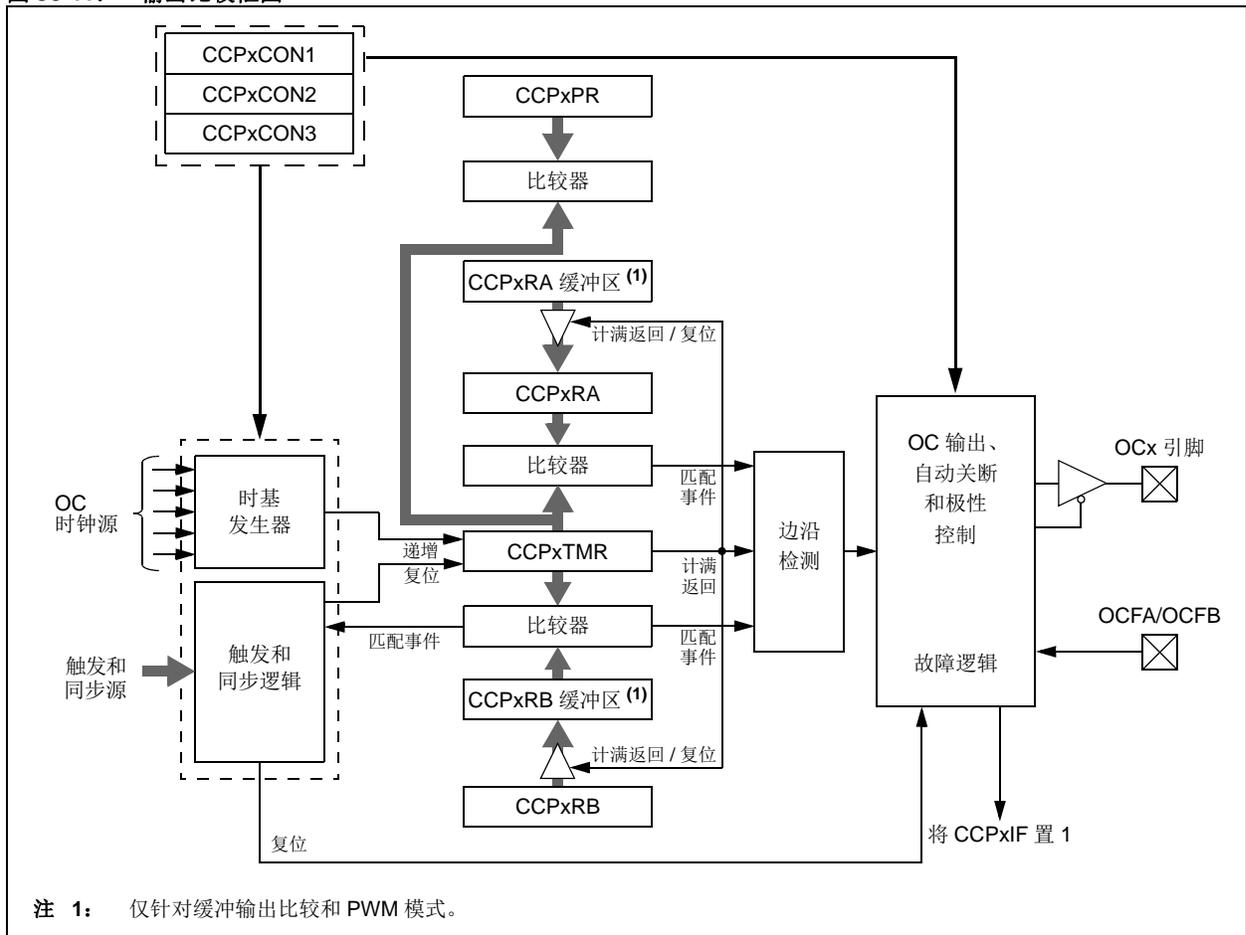
类似于多个早期的 PIC32 模块，输出比较模式可以用作 PWM 发生器。在 MCCP 模块中，有多路 PWM 输出可以用于电源或电机控制应用。

表 30-5 总结了各种输出比较模式。

表 30-5: 输出比较 / PWM 模式

T32	MOD<3:0>	工作模式
0	0001	在比较时输出高电平 (16 位)，单边沿模式
1	0001	在比较时输出高电平 (32 位)，单边沿模式
0	0010	在比较时输出低电平 (16 位)，单边沿模式
1	0010	在比较时输出低电平 (32 位)，单边沿模式
0	0011	在比较时输出翻转 (16 位)，单边沿模式
1	0011	在比较时输出翻转 (32 位)，单边沿模式
0	0100	双边沿比较 (16 位)，双边沿模式
0	0101	双边沿比较 (16 位缓冲)，PWM 模式
0	0110	中心对齐脉冲 (16 位缓冲)，中心 PWM 模式
0	0111	可变频率脉冲 (16 位)

图 30-11: 输出比较框图



30.7.1 单边沿输出比较模式

当 MOD<3:0> = 0001、0010 或 0011 时，选定的输出比较通道将配置为以下单输出比较匹配模式：

- 比较将引脚强制为高电平 (MOD<3:0> = 0001)
- 比较将引脚强制为低电平 (MOD<3:0> = 0010)
- 比较将翻转引脚电平 (MOD<3:0> = 0011)

在单比较模式下，使用 CCPxRA 寄存器。该寄存器中装入一个值，与模块定时器寄存器进行比较。在每次发生比较事件时，将会产生 CPU 中断。

单边沿比较模式使用以下定时器 / 数据寄存器：

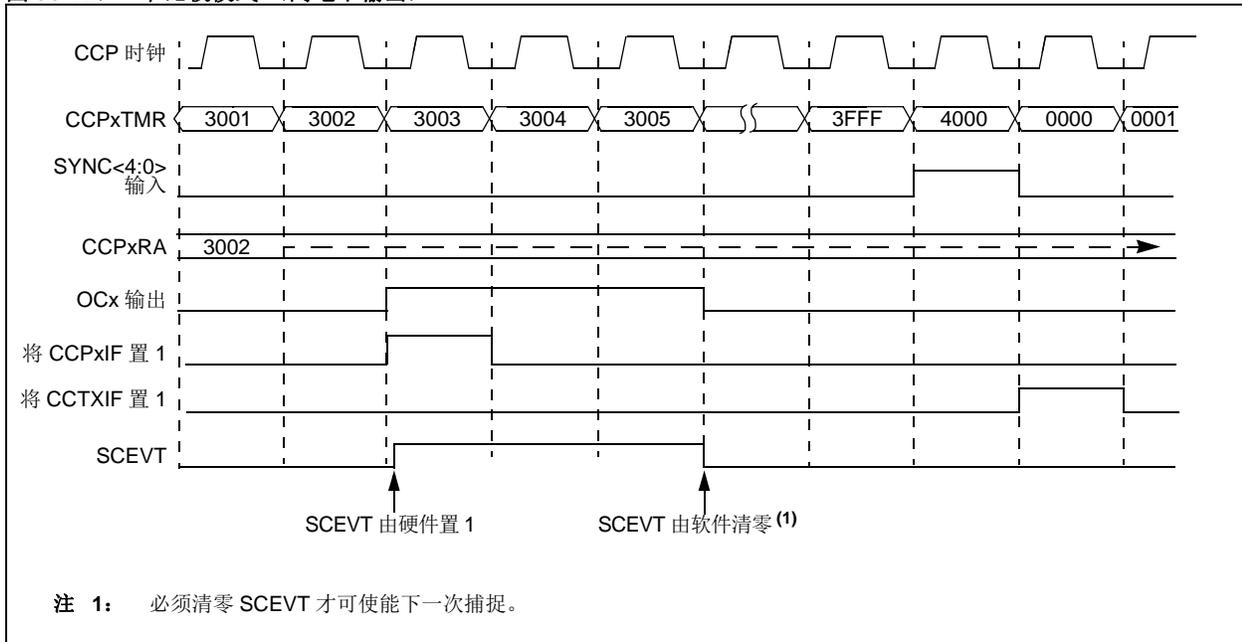
- CCPxTMR<15:0> 用作定时器寄存器 (16 位模式)
- CCPxTMR 用作定时器寄存器 (32 位模式)
- CCPxRA 用作比较值寄存器 (16 位模式)
- CCPxRB:CCPxRA 用作比较值寄存器 (32 位模式)
- CCPxPR<15:0> 用作定时器周期寄存器 (16 位模式)
- CCPxPR 用作定时器周期寄存器 (32 位模式)

30.7.1.1 16 位单边沿比较模式 (高电平输出)

在该模式下 (见 图 30-12)，输出引脚最初驱动为低电平并一直保持低电平，直到定时器和 CCPxRA 寄存器之间发生匹配为止。需要注意的关键时序事件包括：

- 输出引脚在定时器和 CCPxRA 寄存器之间发生匹配的一个时钟周期之后驱动为高电平。输出引脚保持高电平直到改变模式或该模块被禁止。
- 定时器会一直计数至计满返回，或直到选定的 SYNC<4:0> 输入置为有效 (取决于 SYNC<4:0> 的值)，然后在下一个时钟复位至 0h。
- 比较中断信号 (用以将 CCPxIF 置 1) 将置为有效，输出引脚将驱动为高电平。
- 定时器中断信号 (用以将 CCTxIF 置 1) 在发生时基复位或计满返回事件时置为有效一个时钟周期。

图 30-12: 单比较模式 (高电平输出)

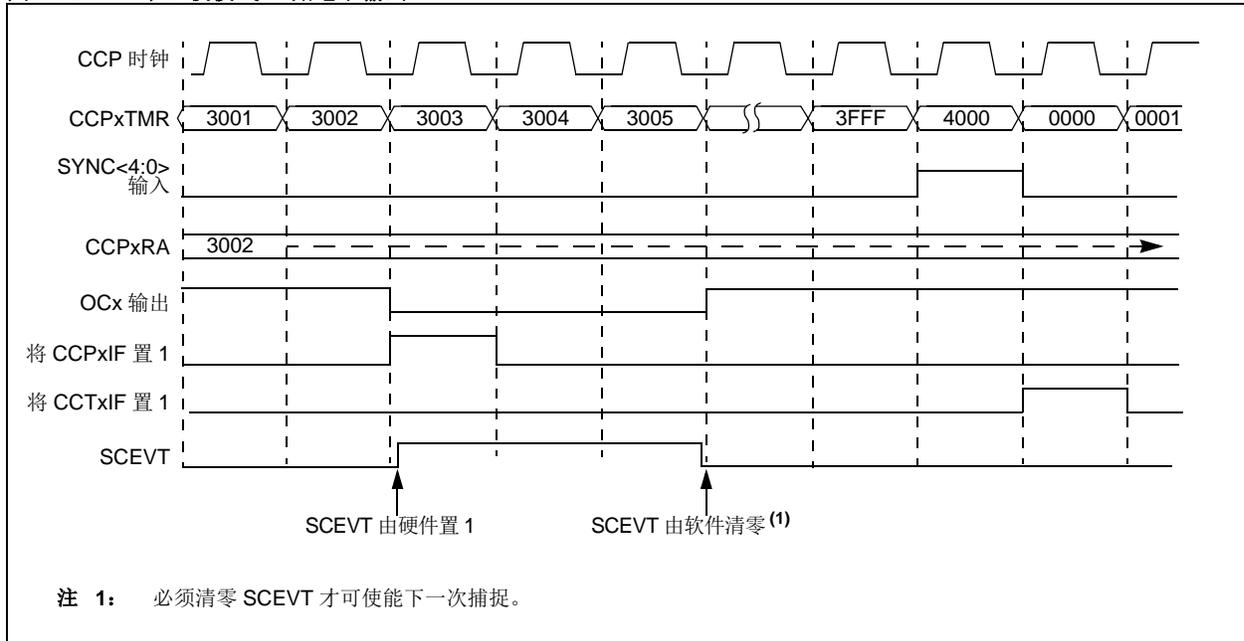


30.7.1.2 16 位单比较模式 (低电平输出)

使能该比较模式之后 (图 30-13), 输出引脚最初将驱动为高电平并保持高电平, 直到定时器和 CCPxRA 寄存器之间发生匹配为止。需要注意的关键时序事件包括:

- 输出引脚在定时器和 CCPxRA 寄存器之间发生匹配的一个时钟周期之后驱动为低电平。输出引脚保持低电平直到改变模式或该模块被禁止。
- 定时器会一直计数至计满返回, 或直到选定的 SYNC<4:0> 输入置为有效, 然后在下一个时钟复位至 0h。
- 比较中断信号 (用以将 CCPxIF 置 1) 将置为有效, 输出引脚将驱动为低电平。
- 定时器中断信号 (用以将 CCTxIF 置 1) 在发生时基复位或计满返回事件时置为有效一个时钟周期。

图 30-13: 单比较模式 (低电平输出)



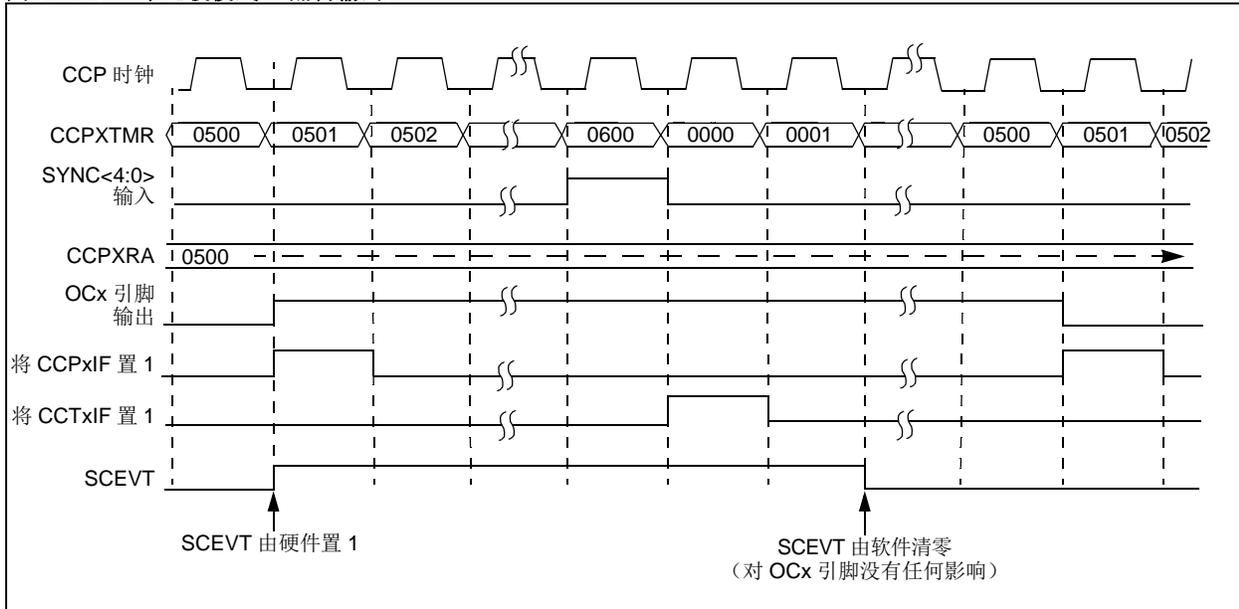
30.7.1.3 16 位单比较模式（翻转输出）

使能该比较模式之后（图 30-14），输出引脚最初将驱动为低电平，并在随后每次定时器和 CCPxRA 寄存器之间发生匹配事件时发生翻转。需要注意的关键时序事件包括：

- 输出引脚的状态在定时器和 CCPxRA 寄存器之间发生匹配的一个时钟周期之后发生翻转。输出引脚保持其新状态直到发生下一次翻转事件、改变模式或该模块被禁止。
- 定时器会一直计数至计满返回，或直到选定的 SYNC<4:0> 输入置为有效，然后在下一个时钟复位至 0h。
- 相应的通道中断输出（CCPxIF）在输出引脚发生翻转时置为有效。
- 时基中断信号（CCTxIF）在发生定时器复位或计满返回事件时产生。

注： 内部 OCx 引脚输出逻辑在器件复位时设置为逻辑 0；但是，可以使用 POLACE 极性控制位来反转翻转模式的初始输出引脚状态。

图 30-14: 单比较模式（翻转输出）



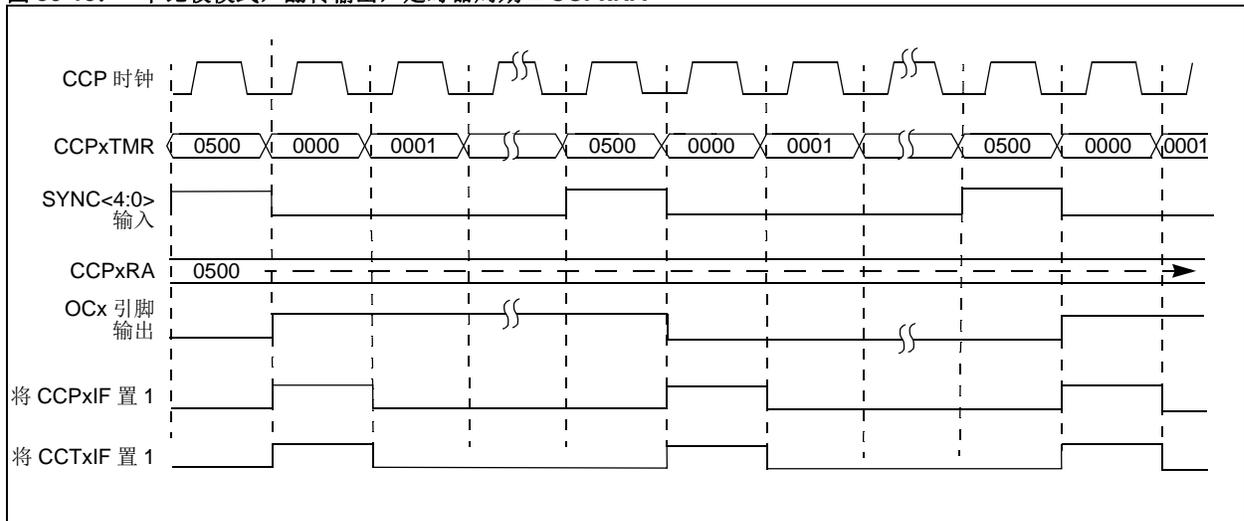
30.7.1.4 16 位单比较模式的特殊情况

在单边沿比较模式下，需要考虑几种特殊情况：

1. 当 CCPxRA 的值大于定时器周期时，比较值将总是大于定时器值。不会发生任何比较事件，比较输出将保持初始状态。
2. 当 CCPxRA 的值等于定时器周期时，比较时间间隔等于定时器周期。将此方式与翻转模式结合使用可以产生固定频率的方波（图 30-15）。
3. 当 CCPxRA = 0h 时，定时器将保持复位状态，触发源置为有效或 CCPTRIG = 0 为触发操作。比较输出将保持在初始状态。当选定的触发源置为无效时，比较输出将发生改变，使定时器可以工作。
4. 如果在比较事件之后将 CCPxRA 清零，则 SYNC<4:0> 信号会置为有效，比较输出将保持其先前状态。
5. 如果在比较事件之后，将 CCPxRA 寄存器修改为大于当前定时器值但小于定时器周期的值，则会在计数周期内产生第二个比较事件。当 MOD<3:0> 为 0001 或 0010 时，还必须清零 SCEVT 位，以便为下一个比较事件复位输出引脚。

注： 对于所有特殊情况，“定时器周期”可以定义为 CCPxPR 匹配或由选定 SYNC<4:0> 源产生的事件。

图 30-15: 单比较模式，翻转输出，定时器周期 = CCPxRA



30.7.1.5 16 位单边沿输出比较事件状态

SCEVT 位 (CCPxSTAT<3>) 指示单边沿比较事件的状态, 它使应用程序无需更改模块工作模式或复位模块即可重新激活单边沿比较事件。它仅在单边沿比较事件期间起作用; 在所有其他模式下, 该位总是读为 0。

当 MOD<3:0> = 0001 时, OCx 引脚在比较事件之后置为高电平, 并且 SCEVT 由硬件设置为 1。应用程序可以用软件清零 SCEVT。清零该位之后, OCx 引脚会复位为低电平输出, 并且比较逻辑发生复位, 以允许下一个上升沿比较事件。

当 MOD<3:0> = 0010 时, OCx 引脚在比较事件之后置为低电平, 并且 SCEVT 由硬件设置为 1。应用程序可以用软件清零 SCEVT。清零该位之后, OCx 引脚会复位为高电平输出, 并且比较逻辑发生复位, 以允许下一个下降沿比较事件。

当 MOD<3:0> = 0011 时, OCx 引脚在比较事件之后发生翻转, 并且 SCEVT 设置为 1。应用程序可以用软件清零 SCEVT。但在位清零时, OCx 引脚的状态不会发生改变。在该模式下, SCEVT 仅提供事件状态信息, 并不会影响 OCx 引脚。

当 MOD<3:0> = 0001 或 0010 时, 应用程序可以通过将 SCEVT 设置为 1 来禁止单边沿输出比较事件。举例来说, 当需要在特定时间间隔内延迟边沿事件时, 该功能会非常有用。在此期间, OCx 引脚不会发生更改。用软件清零 SCEVT 位之后, OCx 输出引脚将复位为初始状态, 上升沿或下降沿将在发生下一个比较事件时产生。

30.7.1.6 使用单比较模式进行 32 位操作

前面的示例均采用 16 位单比较操作 (T32 = 0)。单边沿比较模式还可以使用 32 位时基操作, 通过设置 T32 = 1 来选择。除了将 CCPxRB 寄存器与 CCPxRA 配对来提供 CCPxTMR 的 32 位比较值之外, 32 位模式下的操作是完全相同的。CCPxRA 用于比较值的高 16 位。

没有周期寄存器可用于设置 CCPxTMR 的计数周期。如果需要小于 FFFF FFFFh 的计数周期, 可以通过将模块与外部源进行同步来设置计数周期。

30.7.2 双边沿比较模式

当 $MOD<3:0> = 0100$ 时，输出比较通道会被配置为产生一串连续脉冲。该脉冲串的参数由 $CCPxRA$ 和 $CCPxRB$ 寄存器以及 $CCPxPR<15:0>$ 位决定。

双边沿比较模式仅在 16 位模式下可用。T32 位不起作用。

双边沿比较模式使用以下定时器 / 数据寄存器：

- $CCPxTMR<15:0>$ 用作定时器寄存器
- $CCPxRA$ 用作上升沿值寄存器
- $CCPxRB$ 用作下降沿值寄存器
- $CCPxPR<15:0>$ 用作定时器周期寄存器

图 30-16 显示了双边沿比较模式的信号时序。该模式下的典型操作如下：

1. 使能双边沿比较模式时，引脚状态驱动为低电平。在某个时间点，硬件或软件事件使能（触发）定时器，开始计数过程。
2. 当定时器与比较寄存器 $CCPxRA$ 第一次发生比较匹配时，输出引脚将被驱动为高电平。
3. 当递增定时器计数值与比较寄存器 $CCPxRB$ 发生匹配时，在输出引脚上驱动脉冲的第二个沿（即后沿，从高至低）。在第二次比较时，将产生 $CCPx$ 输出比较中断标志（ $CCPxIF$ ）。
4. 当定时器计满返回（发生周期寄存器匹配事件或由 $SYNC<4:0>$ 定义的事件）时，将产生 $CCPx$ 定时器中断标志（ $CCTxIF$ ）与 CCP 同步信号。
5. 脉冲输出继续重复，直到模式由应用程序或器件复位终止为止。

这是原型情境，此时定时器周期和匹配寄存器均为不同值，其顺序为（定时器周期 $>$ $CCPxRB >$ $CCPxRA$ ）。但是，存在一些条件不同的特殊情境，它们会产生特定类型的输出。表 30-6 列出了这些情境，后面几节对它们进行了介绍。

图 30-16: 典型的双边沿比较时序

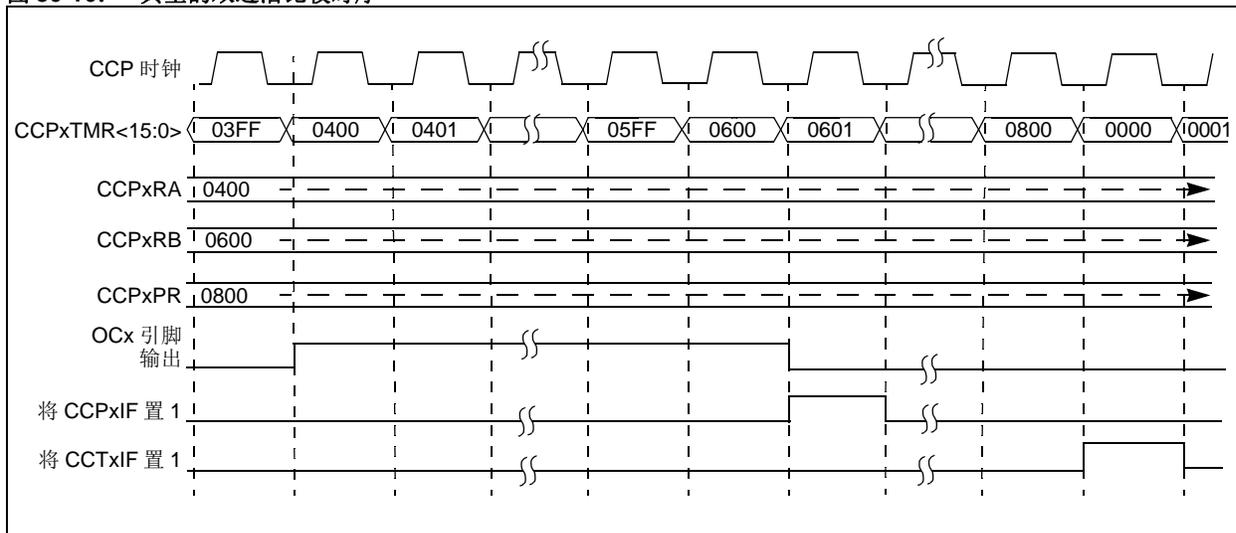


表 30-6: 双边沿比较操作的特殊条件

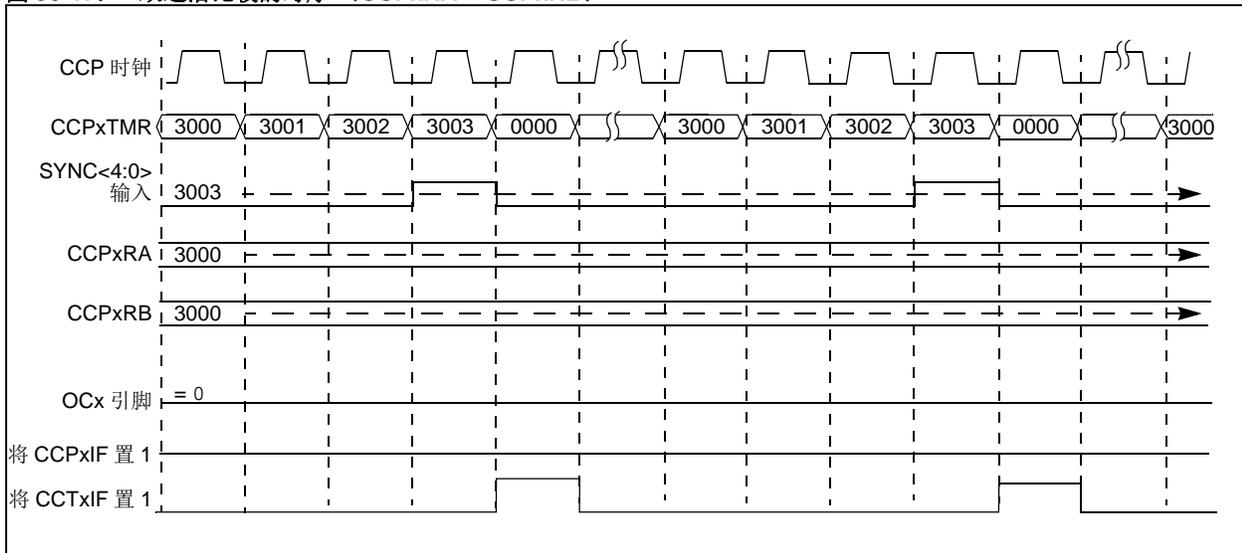
条件	输出	在 OCx 引脚的下 降沿产生输出比较 中断	在定时器与周期 匹配时产生定时 器中断
$CCPxRA = CCPxRB$	无输出， OCx 引脚保持低电平	无	有
$CCPxRA = CCPxRB + 1$	1 个脉冲	有	有
定时器周期 < $CCPxRA$ ⁽¹⁾	无输出	N/A ⁽²⁾	有
定时器周期 = $CCPxRB$ ⁽¹⁾	OCx 在 $CCPxRB$ 处变为 低电平	有	有
定时器周期 < $CCPxRB$ ⁽¹⁾	连续高电平	无	有
定时器周期 = $CCPxRB$, $CCPxRA = 0$ ⁽¹⁾	当 $TMR = 1$ 时 CCP 变为高 电平，当 $CCPxRB$ 与定时器 匹配时变为低电平	有	有
$CCPxRA > CCPxRB$	脉冲串	有	有

- 注 1:** 定时器周期等于周期寄存器值，或由 SYNC<4:0> 选择的输入复位定时器的时间。
注 2: 如果 $CCPxRB$ 也小于定时器周期，则即使 OCx 引脚上没有任何活动，也会产生中断。

30.7.2.1 $CCPxRA = CCPxRB$

如果 $CCPxRA$ 和 $CCPxRB$ 具有相同的值，输出将初始化为低电平并保持低电平；不会产生任何脉冲和任何输出比较中断（图 30-17）。也就是说，PWM 占空比为 0。 $CCPxRB$ 匹配复位 / 清零逻辑将覆盖 $CCPxRA$ 匹配置 1 逻辑，产生的结果就是输出状态不发生改变。

图 30-17: 双边沿比较的时序 ($CCPxRA = CCPxRB$)



30.7.2.2 $CCPxRB = CCPxRA + 1$

当 $CCPxRB$ 的值比 $CCPxRA$ 值大 1，并且两个寄存器均小于周期寄存器时，将产生 1 个 CCP 时钟周期宽的输出脉冲。

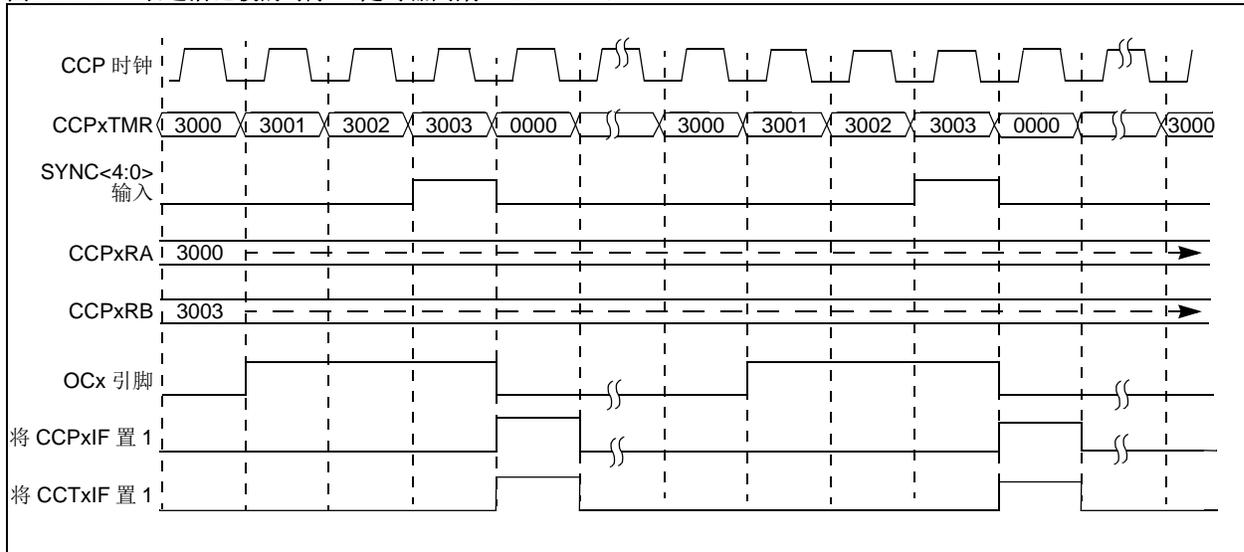
30.7.2.3 定时器周期 < $CCPxRA$

当 $CCPxRA$ 的值大于定时器周期时，不会产生任何输出脉冲。

30.7.2.4 定时器周期 = CCPxRB

当 CCPxRB 的值等于定时器周期时，模块仍然会产生高电平至低电平的跃变。在同步操作中，无论周期是通过外部源（如图 30-18 所示）还是基于与 CCPxPR<15:0> 的匹配事件确定，都是如此。

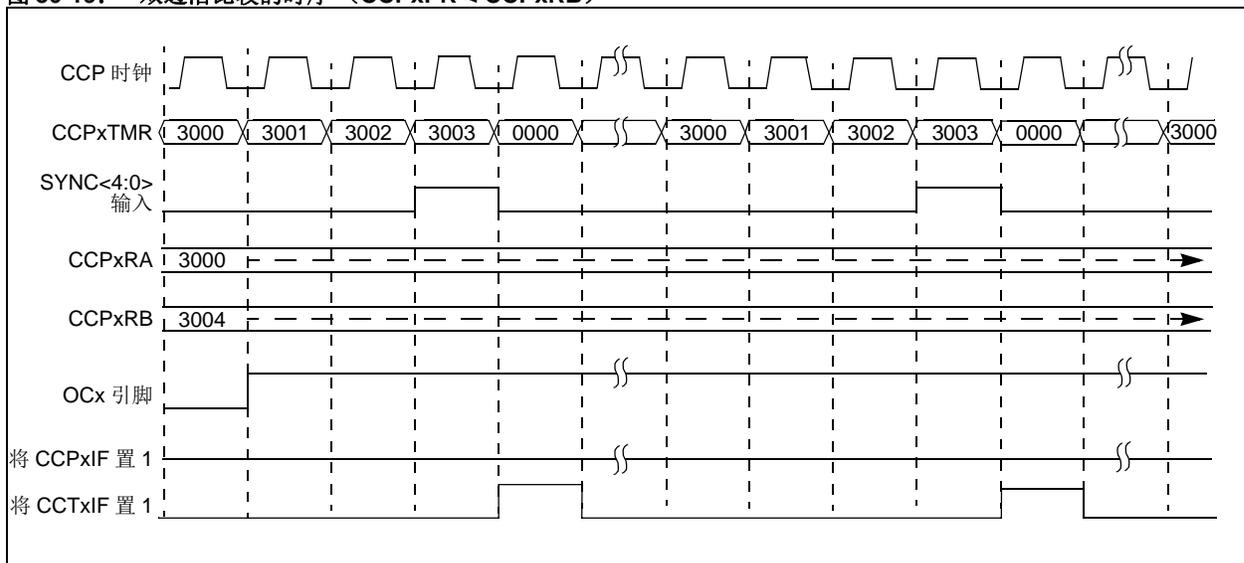
图 30-18: 双边沿比较的时序 (定时器周期 = CCPxRB)



30.7.2.5 定时器周期 < CCPxRB

如果 CCPxPR 的值小于 CCPxRB，但大于 CCPxRA，则只会产生一个引脚跃变，直到 CCPxRB 寄存器内容更改为小于等于 CCPxPR 为止。不会产生任何输出比较中断（图 30-19）。这种条件使模块可以产生 100% 的占空比输出。

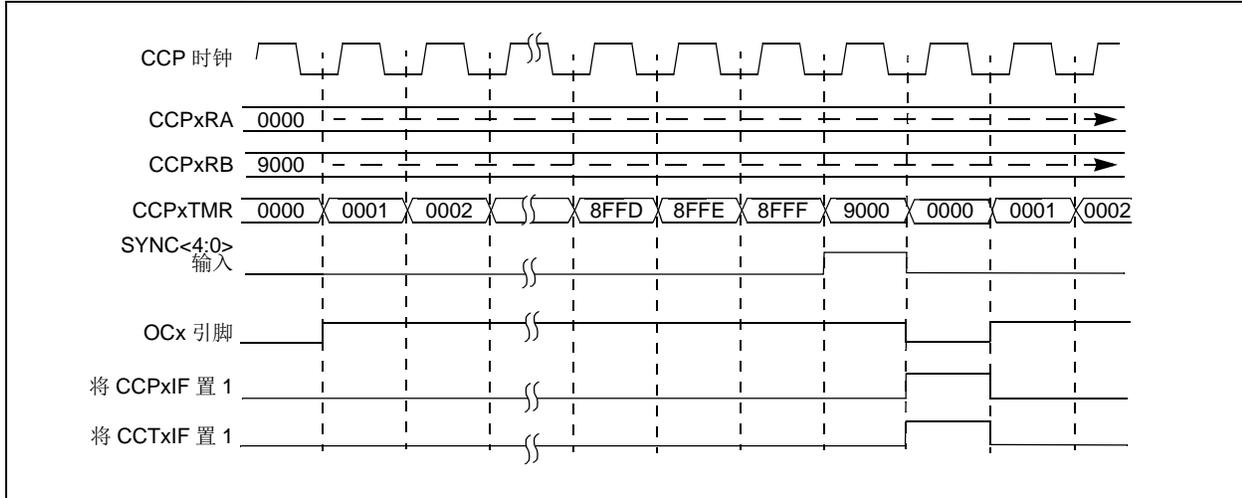
图 30-19: 双边沿比较的时序 (CCPxPR < CCPxRB)



30.7.2.6 CCPxTMR = CCPxRB 且 CCPxRA = 0

在同步操作中，如果 CCPxRA 为 0000h，OCx 输出将在定时器复位之后的第一个时钟置为有效 (CCPxTMR = 0001h)。它会一直保持置为有效，直到 CCPxRB 的值与定时器周期匹配 (当由 SYNC<4:0> 选择的输入置为有效时)。此时，OCx 输出将置为无效，并在下降沿产生 CCPxIF (图 30-20)。

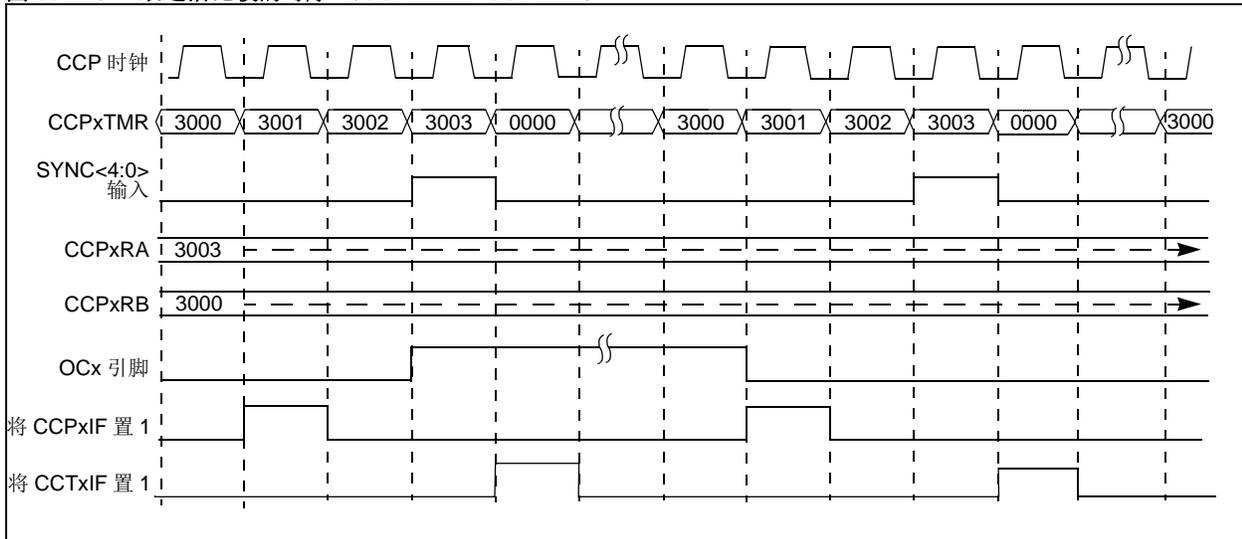
图 30-20: 双边沿比较的时序 (CCPxRA = 0000h, CCPxRB = 定时器周期)



30.7.2.7 CCPxRA > CCPxRB

如果 CCPxRA > CCPxRB，将产生连续的脉冲串。定时器计数递增至第一个匹配值 (CCPxTMR = CCPxRA)，并产生第一个 (上升) 边沿。CCPxTMR 继续递增计数，并在由 SYNC<4:0> 选择的同步源置为有效时复位。然后，定时器递增计数至第二个匹配值 (CCPxTMR = CCPxRB)，此时将产生信号的第二个 (下降) 边沿。在输出脉冲的下降沿，将产生 CCPxIF 中断。序列一直重复，直到模块被禁止 (图 30-21)。

图 30-21: 双边沿比较的时序 (CCPxRA > CCPxRB)



注: 在双边沿比较模式下工作时，CCTxIF 信号在 CCPxRB 寄存器值和 CCPxTMR 之间发生匹配时置为有效。

30.7.3 双边沿缓冲比较 (PWM) 模式

当 MOD<3:0> = 0101 时, 模块的功能与在双边沿比较模式下时相同, 只是 CCPxRA 和 CCPxRB 会进行双缓冲。在输出信号产生的所有其他方面, 操作都是相同的。数据寄存器 (CCPxRA 和 CCPxRB) 的写入数据存储存放在缓冲区中。缓冲区的内容在发生时基复位时传输到 CCPxRA 和 CCPxRB。

双边沿缓冲比较模式仅在 16 位模式下可用。T32 位不起作用。

双边沿缓冲比较模式使用以下定时器 / 数据寄存器:

- CCPxTMR<15:0> 用作定时器寄存器
- CCPxRA 用作下一个周期的上升沿值寄存器
- CCPxRB 用作下一个周期的下降沿值寄存器
- CCPxPR<15:0> 用作定时器周期寄存器

双边沿缓冲比较模式用于产生 PWM 信号。CCPxRA 和 CCPxRB 寄存器进行缓冲使用户可以对 PWM 信号边沿时间进行无毛刺的更新。

如果需要边沿对齐的 PWM 信号, 则将 CCPxRA 的值保持为 0000h。对 CCPxRA 使用一个非零值会产生进行任意相位对齐的 PWM 信号。

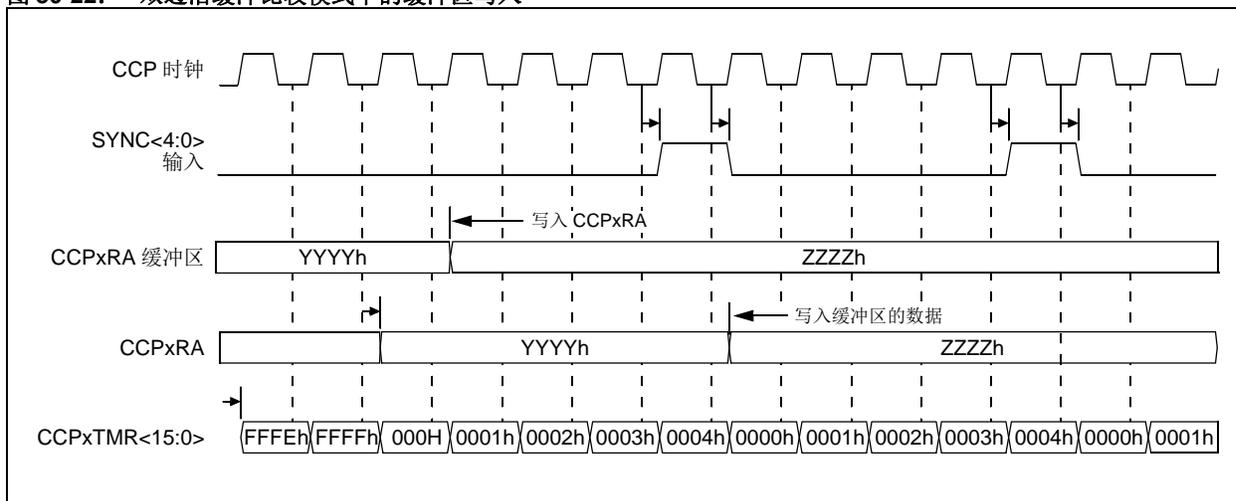
CCPxRA 和 CCPxRB 进行双缓冲。在以下条件下, 数据会从缓冲区写入 CCPxRA 和 CCPxRB:

- 当定时器由于同步事件 (由 SYNC<4:0> 选择的源置为有效) 而复位为 0000h 时
- 当定时器从 FFFFh 计满返回至 0000h 时
- 当禁止该模块 (ON = 0) 时; CCPxRA 和 CCPxRB 的任何写入数据都会立即传输到它们的比较寄存器

图 30-22 显示了在同步操作中写入缓冲区的时序。其中列出了 CCPxRA 及其缓冲区; CCPxRB 及其缓冲区以相同的方式工作。对于输出信号的产生, 请参见第 30.7.2 节“双边沿比较模式”。

例 30-3 给出了为模块配置双边沿缓冲比较模式的过程。

图 30-22: 双边沿缓冲比较模式下的缓冲区写入



例 30-3: 双边沿缓冲比较模式设置

```
// Set MCCP operating mode
CCP1CON1bits.CCSEL = 0;      // Set MCCP operating mode (OC mode)
CCP1CON1bits.MOD = 0b0101;  // Set mode (Buffered Dual-Compare/PWM mode)

//Configure MCCP Timebase
CCP1CON1bits.T32 = 0;      // Set timebase width (16-bit)
CCP1CON1bits.TMRSYNC = 0;  // Set timebase synchronization (Synchronized)
CCP1CON1bits.CLKSEL = 0b000; // Set the clock source (Tcy)
CCP1CON1bits.TMRPS = 0b00; // Set the clock prescaler (1:1)
CCP1CON1bits.TRIGEN = 0;   // Set Sync/Triggered mode (Synchronous)
CCP1CON1bits.SYNC = 0b00000; // Select Sync/Trigger source (Self-sync)

//Configure MCCP output for PWM signal
CCP1CON2bits.OCAEN = 1;    // Enable desired output signals (OC1A)
CCP1CON3bits.OUTM = 0b000; // Set advanced output modes (Standard output)
CCP1CON3bits.POLACE = 0;  // Configure output polarity (Active High)
CCP1TMRbits.TMRL = 0x0000; // Initialize timer prior to enable module.
CCP1PRbits.PRL = 0xFFFF;  // Configure timebase period
CCP1RA = 0x1000;          // Set the rising edge compare value
CCP1RB = 0x8000;          // Set the falling edge compare value
CCP1CON1bits.ON = 1;      // Turn on MCCP module
```

30.7.4 中心对齐脉冲模式

当 $MOD<3:0> = 0110$ 时, 输出比较通道将在中心对齐脉冲模式下提供 PWM 输出。当使用多个 PWM 发生器来控制应用中的电源负载时, 对中心对齐 PWM 信号会很有利。每个 PWM 信号的有效脉冲时间都是围绕虚拟中心点对称的。如果每个 PWM 发生器的占空比不同, 中心对齐可以避免每个电源负载同时发生开关切换。信号的这种对齐方式还允许产生独特的开关模式。

在中心对齐脉冲模式下, $CCPxPR<15:0>$ 寄存器用于设置定时器计数周期 (如果自同步)。将 $CCPxPR<15:0>$ 寄存器值除以 2, 以确定用于脉冲产生的中心参考点。对称脉冲围绕该参考点产生。 $CCPxRA$ 寄存器的值用于指定脉冲总持续时间。

中心对齐脉冲模式仅在 16 位模式下可用。T32 位不起作用。

中心对齐脉冲模式使用以下定时器 / 数据寄存器:

- $CCPxTMR<15:0>$ 用作定时器寄存器
- $CCPxRA$ 用作脉冲宽度寄存器
- $CCPxRB$ 用作触发输出值寄存器
- $CCPxPR<15:0>$ 用作定时器周期寄存器 (影响上升 / 下降沿时间和脉冲中心)

中心对齐模式使用一个专用的 16 位加法器 / 减法器 and 边沿发生逻辑, 它是 CCP 模块硬件的一部分。时基的计数周期由 $CCPxPR$ 寄存器中的值指定, 它将进行双缓冲。缓冲区内内容在发生定时器计满返回事件时, 或在由 $SYNC<4:0>$ 选择的同步源置为有效时更新。进行缓冲的 $CCPxPR$ 寄存器值的高 15 位用作时基计数周期的中心参考值。模块中的硬件加法器使用该值作为基于 $CCPxRA$ 和 $CCPxPR<15:0>$ 中的值计算上升沿和下降沿时间的基线。

当定时器复位并开始递增计数时, 加法器会用 $CCPxPR<15:0>$ 值的二分之一减去 $CCPxRA$ 值的二分之一。其差值将与定时器值进行比较, 以确定何时出现 PWM 信号的上升沿。

出现上升沿之后, 加法器会将 $CCPxRA$ 值的二分之一与 $CCPxPR<15:0>$ 值的二分之一相加。其和值将与定时器值进行比较, 以确定何时出现 PWM 信号的下降沿。

注: 对 $CCPxRA$ 寄存器的更新将进行缓冲, 并在下一个 PWM 周期变为有效。与 $CCPxPR<15:0>$ 的比较总是使用 $CCPxRA$ 的缓冲值来进行。

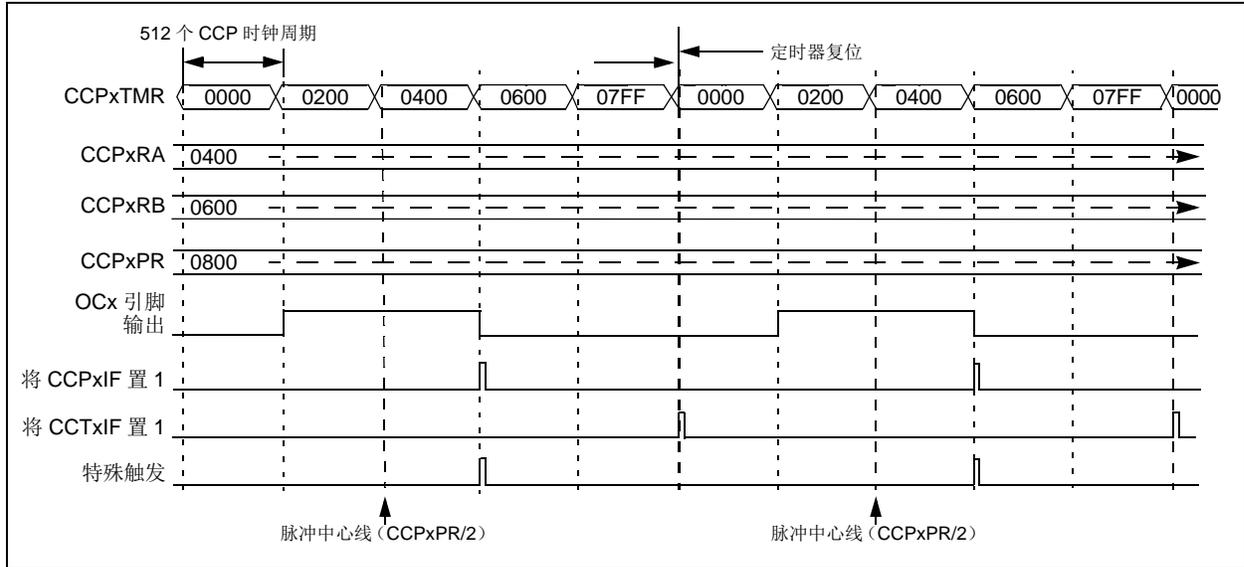
例 30-4 给出了一个说明如何确定上升沿和下降沿值的典型示例。图 30-23 给出了基于寄存器值产生中心对齐脉冲的图示。

例 30-4: 计算中心对齐模式下的上升沿和下降沿值

```

For CCPxRA = 400h and CCPxPR = 800h:
Rising Edge = (CCPxPR<15:0>/2) - (CCPxRA/2)
              = 400h - 200h
              = 200h
Falling Edge = (CCPxPR<15:0>/2) + (CCPxRA/2)
              = 400h + 200h
              = 600h
    
```

图 30-23: 比较中心对齐 PWM

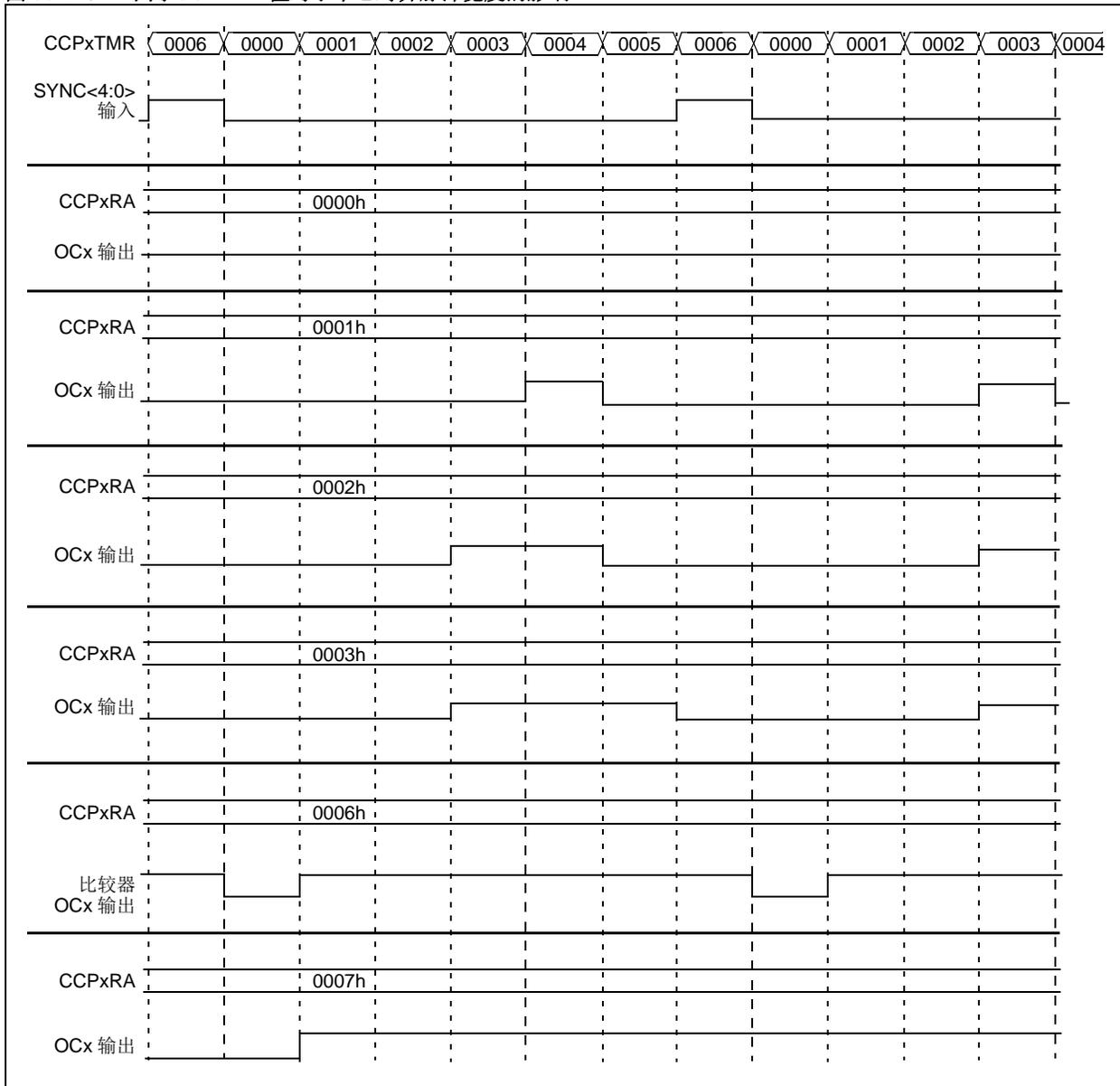


如果 CCPxRA 的值为偶数，则它对上升沿或下降沿的位置没有任何影响。如果其值为奇数，则加法器只会对于下降沿比较事件运用最低位 (LSb) 中的 1。这会将有效脉冲的宽度增加 1 个定时器周期，使围绕虚拟中心点值的 1/2 计数产生不对称。如果不希望产生这种不对称，应用程序应仅向 CCPxRA 写入偶数脉冲宽度值。

注: 要使模块在中心对齐脉冲模式下产生最大占空比，用户必须将时基周期设置为偶数值。向 CCPxRA 寄存器值写入等于 (CCPxPR + 1) 的值时，可以获得 100% 的占空比。所有应用程序都应基于 CCPxPR 的值限制写入 CCPxRA 寄存器的最大值。当 CCPxRA 大于 (CCPxPR + 1) 时，将不会产生任何脉冲。

图 30-24 说明了增大 CCPxRA 值对于脉冲宽度的影响。它还说明了当 CCPxRA 等于零（无输出）、定时器周期（当脉冲宽度等于定时器周期时）或当 CCPxRA 大于周期（输出在初始边沿事件之后保持有效）时的影响。

图 30-24: 不同 CCPxRA 值对于中心对齐脉冲宽度的影响



30.7.5 输出触发

CCPxRB 寄存器的值对于确定脉冲宽度或时序没有任何作用。它用于确定相对于中心对齐脉冲事件的特殊事件触发时间。例如，它对于在 PWM 脉冲中心触发 A/D 转换会很有用，或者在脉冲边沿之一触发某个其他事件可能会更好。用户可以向 CCPxRB 写入适当的值，也可以使用 CCPxRA 和 CCPxPRL 的值自动计算 CCPxRB 的正确值，对其进行更新。

CCPxRB 的值必须小于由 CCPxPRL 寄存器或由来自选定 SYNC<4:0> 输入的事件设置的周期值。特殊事件触发可以用于启动 A/D 转换或触发其他外设事件。

30.7.6 可变频率脉冲模式

当 MOD<3:0> = 0110 时，输出比较通道将在可变频率脉冲模式下提供 PWM 输出。该模式使用一个累加器寄存器和一个加法器寄存器来产生具有 50% 固定占空比的可变频率输出信号。

可变频率脉冲模式使用以下数据寄存器：

- CCPxTMR<15:0> 用作累加器
- CCPxRA 用于存储下一次操作的加法器值

加法器寄存器中的值将在时基的每个上升沿与累加器相加。当累加器寄存器发生溢出时，输出信号会发生翻转。在该模式下不使用 CCPxRB 和 CCPxPR<15:0> 寄存器。

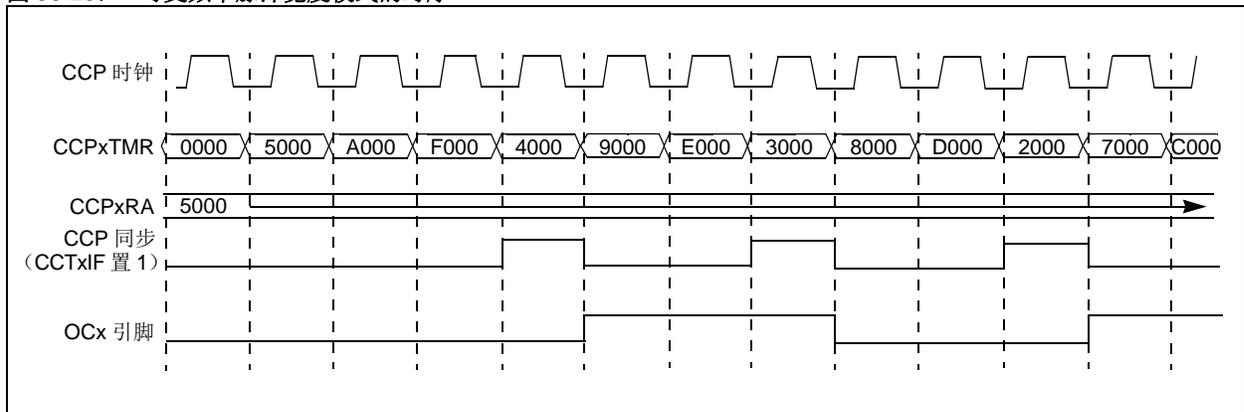
可变频率脉冲模式仅在 16 位模式下可用。T32 位不起作用。

输出频率和 CCPxRA 的值是相关联的，如公式 30-1 所示；在给出一个值的情况下，可以计算另一个值。输出频率是模块时钟源频率 FCLK、CCPxRA 寄存器中的加法器值和累加器大小 (2¹⁶) 的函数。

公式 30-1: F_{OUT} 和 CCPxRA 目标值之间的关系

$$F_{OUT} = \frac{F_{CLK} \cdot CCPxRA}{2 \cdot 2^{16}} \qquad CCPxRA = \frac{2 \cdot 2^{16} \cdot F_{OUT}}{F_{CLK}}$$

图 30-25: 可变频率脉冲宽度模式的时序



30.7.7 比较 /PWM 模式的输出控制

当模块在输出比较模式下工作时，3 个控制块将决定输出比较信号如何送至输出引脚上：

- 输出模式控制块（仅对于 MCCP）
- 自动关断控制块
- 输出极性控制块

MCCP 模块使用输出模式控制块来控制输出比较信号至 6 个可用输出引脚的传送。该控制块实现了多个输出 PWM 的更高级的电机控制和电源控制功能。它未在 SCCP 模块中实现。

自动关断控制块可响应异步外部输入或软件控制，将模块控制下的所有输出引脚都置为预先确定的状态。

输出极性控制块决定模块控制下的每个引脚的输出极性。该控制块在输出引脚的所有其他控制之后生效。

30.7.7.1 输出模式选择

OUTM<2:0> 控制位（CCPxCON3<26:24>）用于选择 MCCP 的输出模式。在输出比较模式下工作时，可以选择几种输出模式之一，它们将以不同的方式使用 OCxA 至 OCxF 输出引脚。在一些输出模式下，使用死区延时发生器来实现输出引脚之间的切换延时。

输出控制逻辑并不决定如何产生输入信号，仅决定信号如何送至输出引脚。输出控制逻辑的信号源可以是可通过 MOD<3:0> 控制位选择的任一输出比较工作模式。

输出控制逻辑不会为了同步而与输入信号产生逻辑进行交互。在某些工作模式下，输出控制逻辑将通过等待输入信号周期边界来将信号切换到不同的输出引脚。

使用 OUTM<2:0> 位可以选择以下输出模式：

- 可转向单输出模式（默认）
- 有刷直流（电机）输出模式（正向和反向）
- 半桥输出模式
- 推挽输出模式
- 输出扫描模式

30.7.7.2 输出引脚使能（MCCP）

由 MCCP 模块控制的每个输出引脚都可以使用 OCxEN 控制位（CCPxCON2<29:24>）单独使能。这其中的每个位（OCAEN 至 OCFEN）控制相应的 CCP 输出：OCxA 至 OCxF。如果将 OCxEN 控制位置 1，相应的 I/O 引脚会接收到由该模块产生的输出比较信号。如果 OCxEN 控制位清零，则 I/O 引脚由端口逻辑或另一个更高优先级外设控制。用户必须小心确保在将 OCxEN 控制位清零时，I/O 引脚处于正确的状态。

当模块在输入捕捉模式或定时器模式下工作时，OCxEN 控制位对模块操作不起任何作用。

OUTM<2:0> 控制位会影响 OCxEN 引脚的功能，具体取决于所选的模式。这些位可以提供转向功能，在特定时间将输出比较信号重定向到不同的引脚。这种转向功能在电机和电源控制应用中非常有用。OCxEN 位也可以用于将模块输出信号重定位到不同的输出引脚集。例如，半桥输出模式会在 OCxA/OCxB、OCxC/OCxD 和 OCxE/OCxF 引脚上复制相同的信号对。用户可以使用 OCxEN 位使能这些引脚对中的任意引脚对，将信号转至方便的位置。

所有 CCP 模块上的 OCAEN 引脚会复位为 1；这使得在器件复位时，默认的输出比较输出引脚在默认情况下是可用的。对于 MCCP 模块，在发生复位时，所有其他 OCxEN 位会复位为 0，从而禁止其他 OCx 引脚。对于使用多个 MCCP 输出引脚的应用，用户需要负责正确地初始化所有输出引脚。

30.7.7.2.1 输出引脚使能 (SCCP)

SCCP 模块只有一个输出引脚 OCxA。在输出比较或 PWM 模式中，OCAEN 位 (CCPxCON2<24>) 是唯一实现的控制位。它决定模块是否具有输出引脚的控制权。默认情况下，器件复位时会使能该 OCxA 输出。

30.7.7.3 可转向单输出模式

可转向单输出模式是控制块的默认输出模式，在 OUTM<2:0> = 000 时选择。在该模式下，由输出比较逻辑产生的单个信号将送至所有可用的模块输出引脚。应用程序可以通过将相应的 OCxEN 位置 1，单独使能每个输出引脚来产生输出比较信号。

30.7.7.4 推挽输出模式

当 OUTM<2:0> = 001 时，将选择推挽 PWM 模式。在该模式下，输出比较信号在连续的时基周期在 OCxA 和 OCxB 输出引脚之间切换。

对于每个周期，其中一个引脚会连接到输出比较信号，另一个引脚则驱动为无效状态。在这种推挽模式下，OCxC/OCxD 和 OCxE/OCxF 输出引脚会复制 OCxA/OCxB 引脚对的输出和端口控制信号。这使用户可以通过使用 OCxEN 控制位将推挽输出信号转至另一个引脚对。要允许模块控制两个输出引脚，必须至少将一对 OCxEN 控制位置 1。

推挽模式常用于驱动直流 / 直流和直流 / 交流电源中的变压器，如图 30-27 所示。每个 PWM 输出引脚驱动变压器的一侧，通过一个外部功率晶体管进行接线。变压器具有一个连接到直流母线电压的中心绕组。应用程序应确保对于变压器的每一侧产生相同的脉冲宽度，以防止变压器绕组中的直流电流；因此，占空比必须对于两个时基周期保持相同。

此外，也可以使用 4 晶体管推挽电路 (图 30-28)。这种实现方式使用第二对输出引脚来驱动第二对功率晶体管。故意交换第二对晶体管的连接，从而使对角线上的每对晶体管在同一时间开启。

图 30-26: 推挽模式下的输出和中断时序

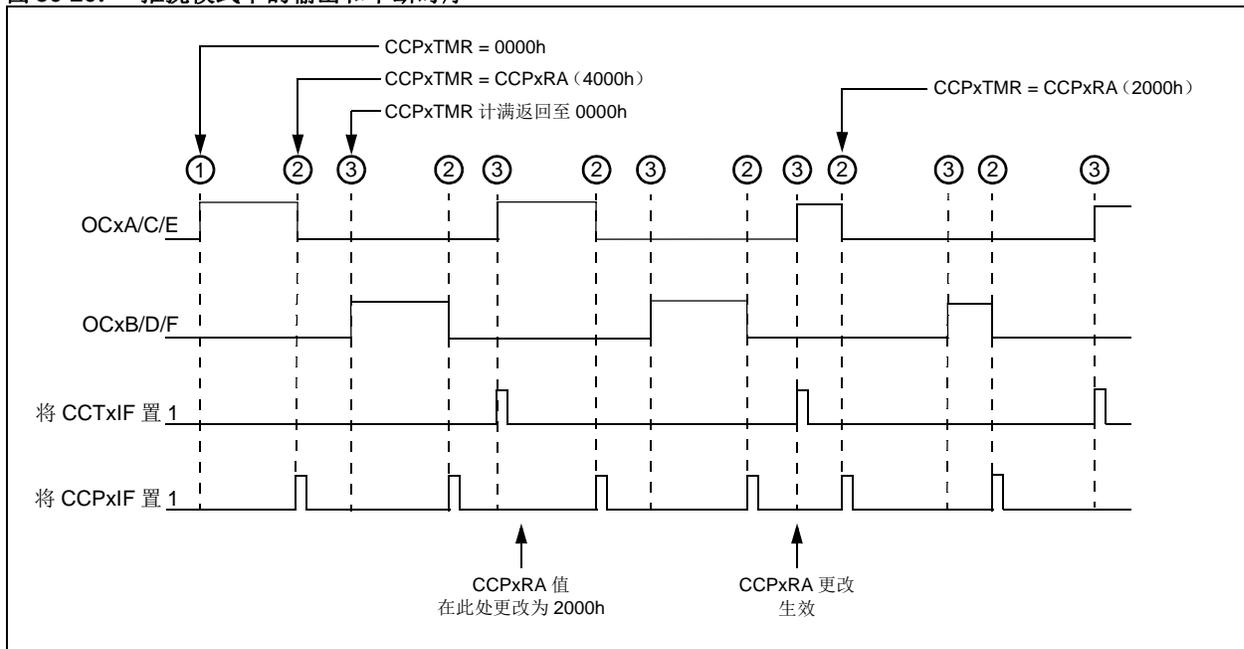


图 30-27: 典型推挽操作

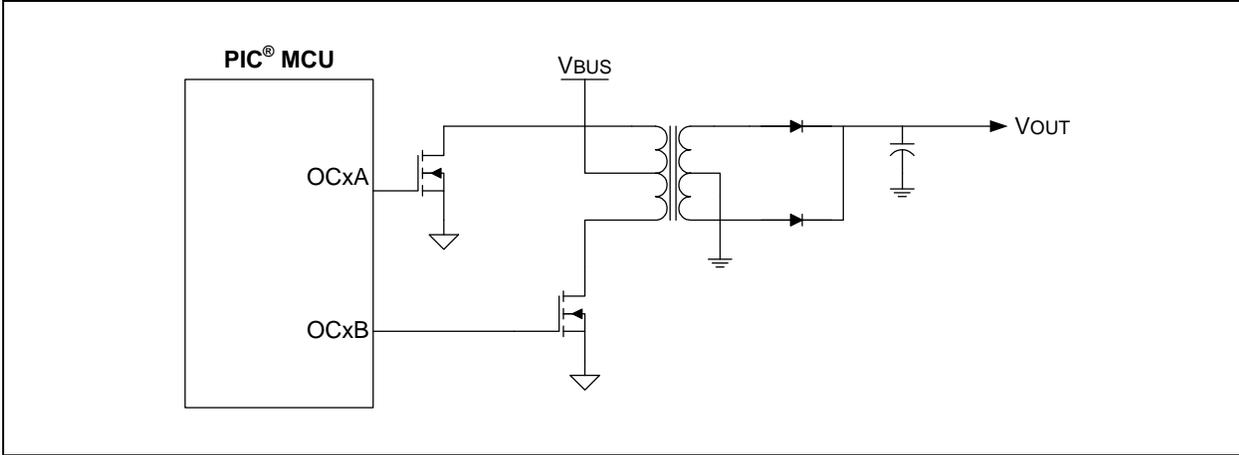
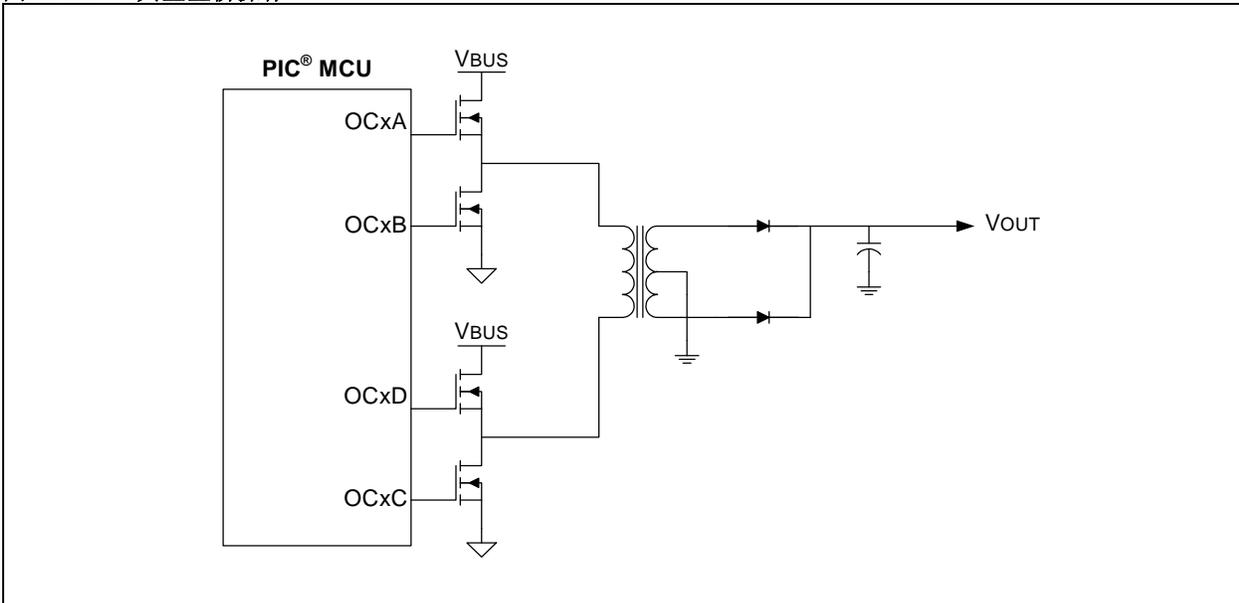


图 30-28: 典型全桥操作



30.7.7.5 半桥输出模式

当 $OUTM\langle 2:0 \rangle = 010$ 时，将选择半桥输出模式。在该模式下，模块在 OCxA 和 OCxB 上产生互补输出信号（图 30-29）。OCxB 信号是 OCxA 信号的反信号。如果使用非零死区延时，它会被插入两个引脚的开关事件之间。

在半桥模式下，OCxC/OCxD 和 OCxE/OCxF 输出引脚会复制 OCxA/OCxB 引脚对的输出和端口控制信号。这用户可以通过使用 OCxEN 控制位将互补输出信号转至另一个引脚对。要产生半桥输出信号，应用程序必须选择至少一对 OCxEN 控制位。

半桥输出模式通常用于控制电源电路，如图 30-30 所示。如果向 $DT\langle 5:0 \rangle$ ($CCPxCON3\langle 5:0 \rangle$) 写入非零的死区值，则会在 OCxA 和 OCxB 信号的开关边沿之间插入延时。

关于死区发生器的更多信息，请参见第 30.7.7.10 节“死区延时发生器”。

图 30-29: 半桥输出

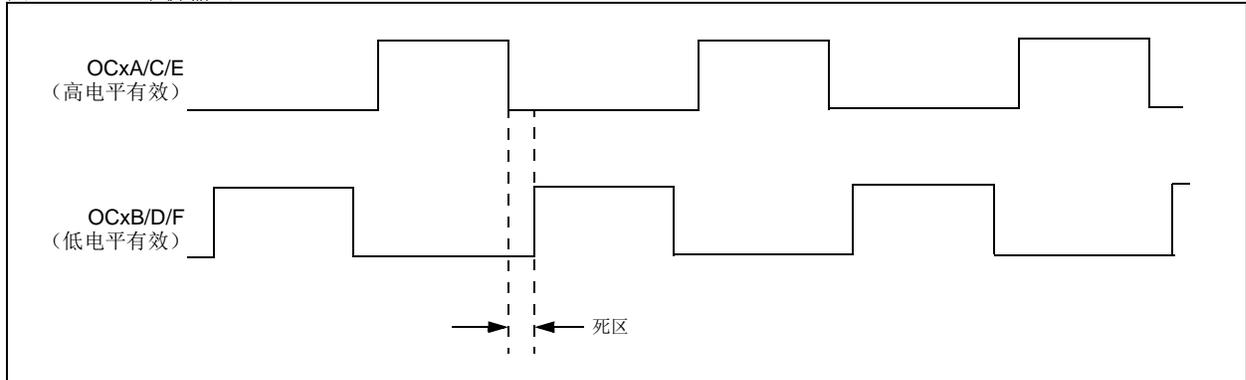
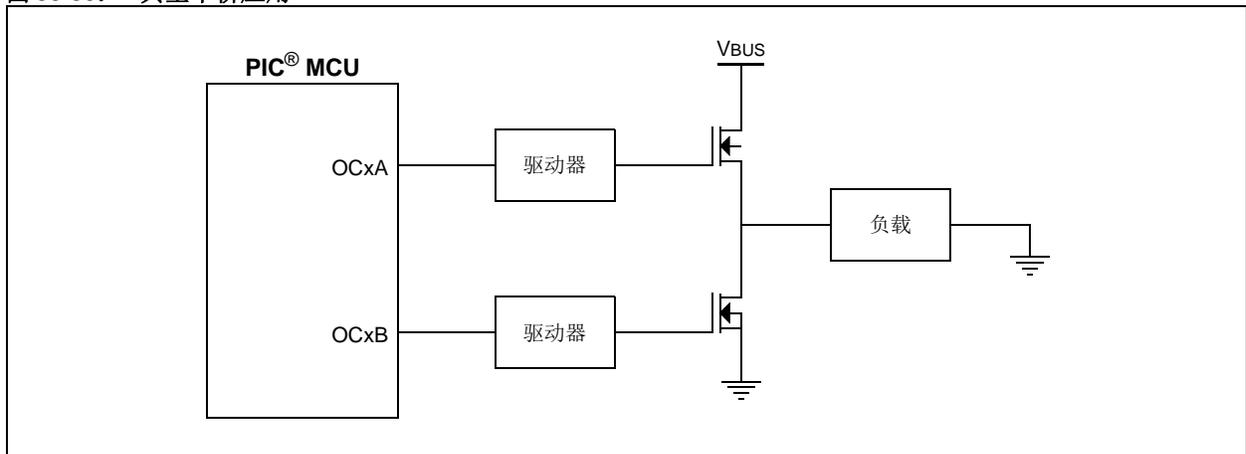


图 30-30: 典型半桥应用



30.7.7.6 有刷直流输出模式

当 $OUTM\langle 2:0 \rangle = 101$ （正向模式）或 100 （反向模式）时，将选择有刷直流输出模式。在这些模式下，输出比较逻辑产生的信号将送至 4 个输出引脚：OCxA 至 OCxD。要让模块控制这些输出引脚，应用程序必须将相应的 OCAEN、OCBEN、OCCEN 和 OCDEN 位置 1。

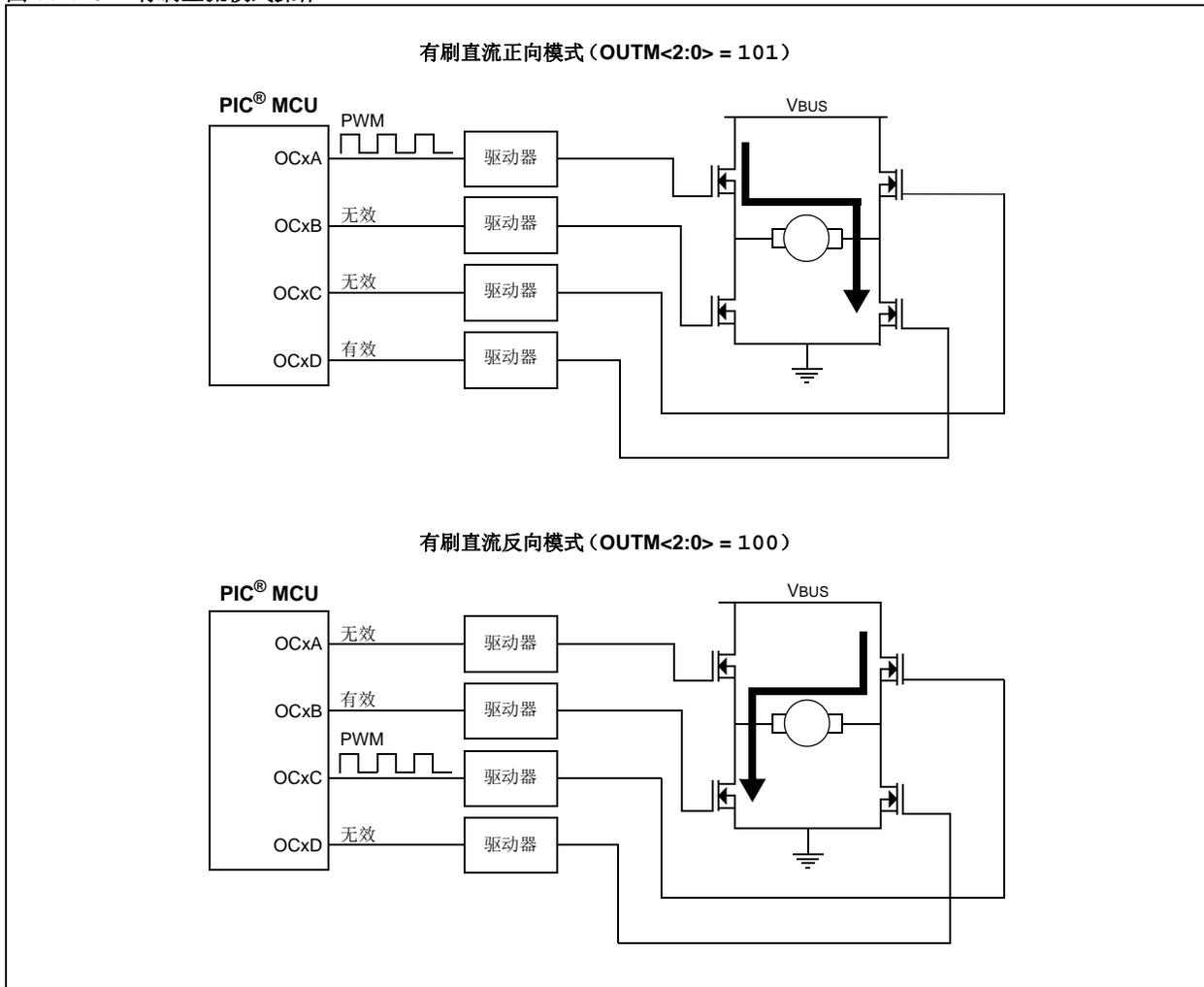
对于每种模式，4 个输出引脚中只有两个引脚会被驱动为有效状态：

- 正向模式 ($OUTM\langle 2:0 \rangle = 101$)：
 - OCxA 引脚接收 PWM 发生器信号
 - OCxD 引脚驱动为有效状态
 - OCxB 和 OCxC 引脚驱动为无效状态
- 反向模式 ($OUTM\langle 2:0 \rangle = 100$)：
 - OCxC 引脚接收 PWM 发生器信号
 - OCxB 引脚驱动为有效状态
 - OCxA 和 OCxD 引脚驱动为无效状态

在有刷直流模式下，OCxE 和 OCxF 输出引脚不受模块控制。用户可使用 OCEEN 和 OCFEN 控制位使能这些引脚；但是，这些引脚将保持在无效状态。

图 30-31 显示了在典型应用中 4 个引脚如何连接和用于控制外部电路。4 个输出信号的实际极性由输出极性控制电路决定（关于更多信息，请参见第 30.7.7.12 节“输出极性控制”）。

图 30-31： 有刷直流模式操作



除了在占空比处于或接近 100% 时的方向模式更改期间，不需要有刷直流模式的死区延时。在运行时，通常预期用户会在正向模式和反向模式之间切换。在 $OUTM<2:0> = 10x$ 时，方向更改通过在应用程序软件中翻转 $OUTM0$ 位而实现。当 $OUTM<2:0>$ 控制位在值 100 和 101 之间切换时，模块逻辑可以检测到该事件并触发死区发生器。方向更改将与 CCP 定时器周期进行同步，并在由 $SYNC<4:0>$ 选择的同步源置为有效时发生。

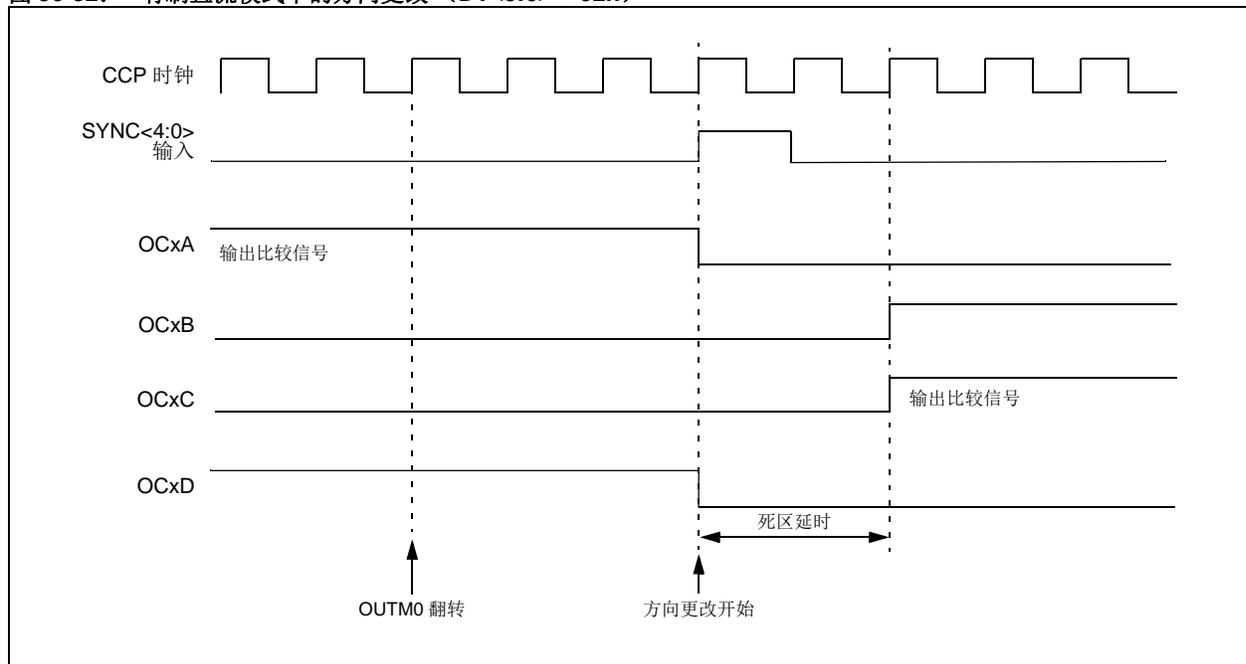
在 PWM 发生器设置为低占空比的情况下进行方向更改时不需要死区，因为有效控制的开关会在方向更改发生之前关闭一段时间。当占空比接近 100% 时，可能需要通过死区来确保由模块控制的顶部和底部开关将至少关闭一定的时间。

在方向更改时，会发生以下事件序列：

1. 在下一个 PWM 时基复位边界，两个当前有效引脚（ $OCxA$ 和 $OCxD$ ，或 $OCxB$ 和 $OCxC$ ）会被驱动为其无效状态。
2. 如果 $DT<5:0>$ 的值为 000h，则立即将一对新的输出引脚置为有效。
3. 如果 $DT<5:0>$ 的值不为零，则在 $OUTM0$ 发生翻转时， $DT<5:0>$ 位将被装入死区延时计数器。新的一对输出引脚将在死区计数器计时期满之后置为有效。

图 30-32 显示了在死区设定为 2 个时钟周期（02h）值时的方向更改的时序。

图 30-32: 有刷直流模式下的方向更改 ($DT<5:0> = 02h$)



30.7.7.7 输出扫描模式

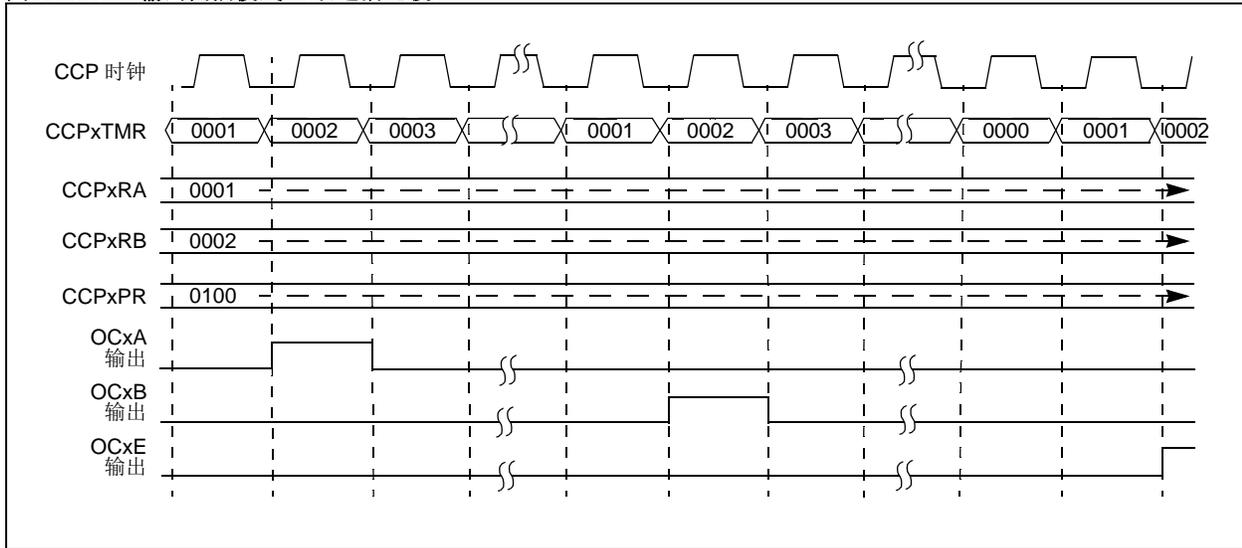
输出扫描模式 (OUTM<2:0> = 110) 类似于单输出可转向模式, 只是输出比较信号会自动在可用 OCx 引脚之间排序。要在输出扫描模式期间使用的 MCCP 输出引脚通过将相应 OCxEN 控制位置 1 来进行选择。例如, 如果将 OCAEN、OCBEN 和 OCEEN 位置 1, MCCP 模块会使用连续模式, 在 OCxA、OCxB 和 OCxE 输出引脚之间自动按顺序对输出比较信号进行转向。

如果禁止了模块 (ON = 0), 扫描序列逻辑会复位, 并在第一个使能引脚上重新开始。当时基发生触发或复位时, 输出比较信号将移至序列中的下一个已使能输出。

图 30-33 显示了使用 OCxA、OCxB 和 OCxE 的输出扫描模式的基本操作。输出扫描模式还可用于触发操作或单脉冲模式, 用以产生序列之间具有延时的输出序列。

注: 在模块使能之前, 输出引脚的实际状态取决于端口引脚的控制逻辑设置或较低优先级的已使能外设。

图 30-33: 输出扫描模式 (双边沿比较)

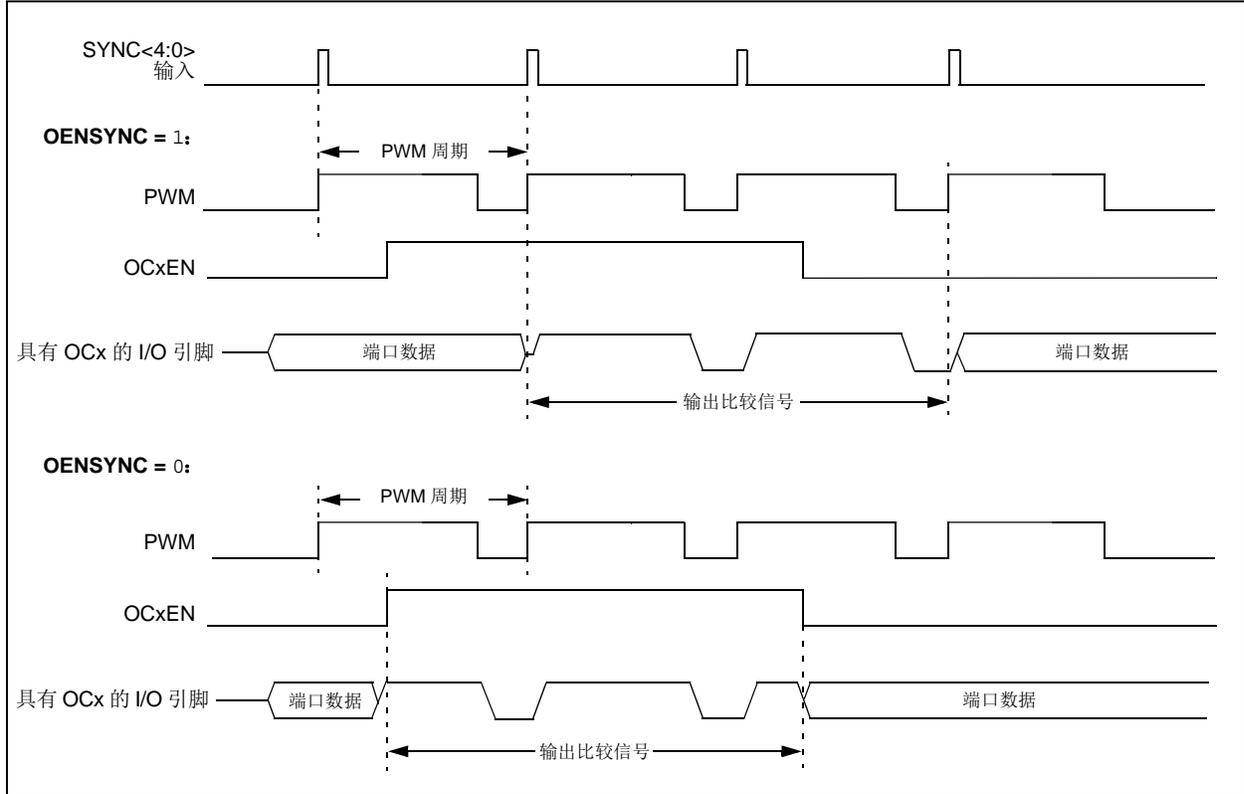


30.7.7.8 输出使能同步

可以选择将对于 OCxEN 控制位的更改与定时器周期进行同步，使软件对 PWM 设置的更改可以与 PWM 周期边界进行同步。这可以防止由于转向变化而产生的不完整的输出脉冲。

OENSYNC 控制位 (CCPxCON2<31>) 控制 PWM 输出与周期边界的同步。当 OENSYNC = 1 时，对 OCxEN 控制位的更改会在定时器复位 (即，由 SYNC<4:0> 选择的同步输入置为有效) 时生效。当 OENSYNC = 0 时，对 OCxEN 控制位的更改将立即生效。图 30-34 说明了 OENSYNC 位对与 OCx 输出共用的 I/O 引脚的效果。

图 30-34: OENSYNC 位操作



30.7.7.9 触发时输出使能

OETRIG 控制位 (CCPxCON3<31>) 使用户可以在触发定时器之前选择 MCCP 输出引脚是保持高阻态还是由模块驱动。OETRIG 控制位只会影响触发模式 (TRIGEN = 1) 下的模块操作。

OETRIG 位功能的操作会因模块的输出模式而异。对于输出扫描模式 (OUTM<2:0> = 110)，在触发时基时，将仅驱动扫描序列中已使能的当前有效输出引脚。扫描序列的所有其他已使能引脚将保持高阻态。

对于 OUTMx 位的所有其他设置，通过 OCxEN 控制位使能的所有输出在时基触发时均驱动为有效。

无论模式如何，在未触发定时器 (CCPTRIG = 0) 时，所有已使能的输出引脚均保持高阻态。

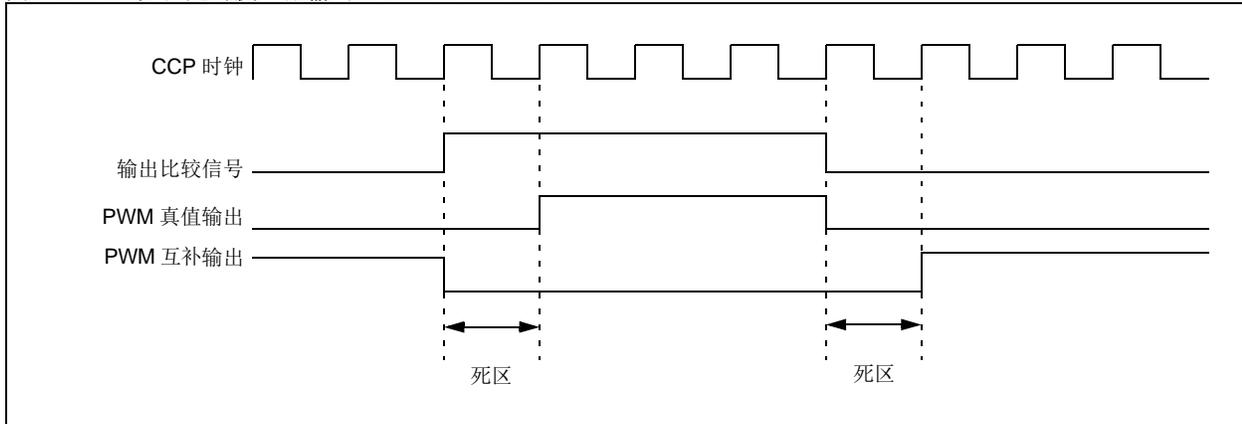
30.7.7.10 死区延时发生器

死区延时发生器在模块特定的多输出 PWM 模式中使用。它基于 PWM 发生器的单个输出信号产生两个输出信号：“真值”信号（与输出比较信号的极性匹配）和反相“互补”信号。它还会在一个信号驱动为无效，另一个信号驱动为有效的时间内产生短暂的延时。

发生器包含了一些用以监视输入信号跃变的边沿探测器和一个数字递减定时器。输出信号的上升沿会被延迟 DT<5:0> 位（CCPxCON3<5:0>）所设置的时钟周期数。如果 DTx 位的值为零，则实际上会禁止死区延时发生器，在每路输出上产生互补的输出信号，且在信号跃变间没有延时。

图 30-35 给出了死区发生器输出的时序图，显示了真值信号和互补信号之间的关系。

图 30-35: 死区延时发生器输出 (DT<5:0> = 01h)

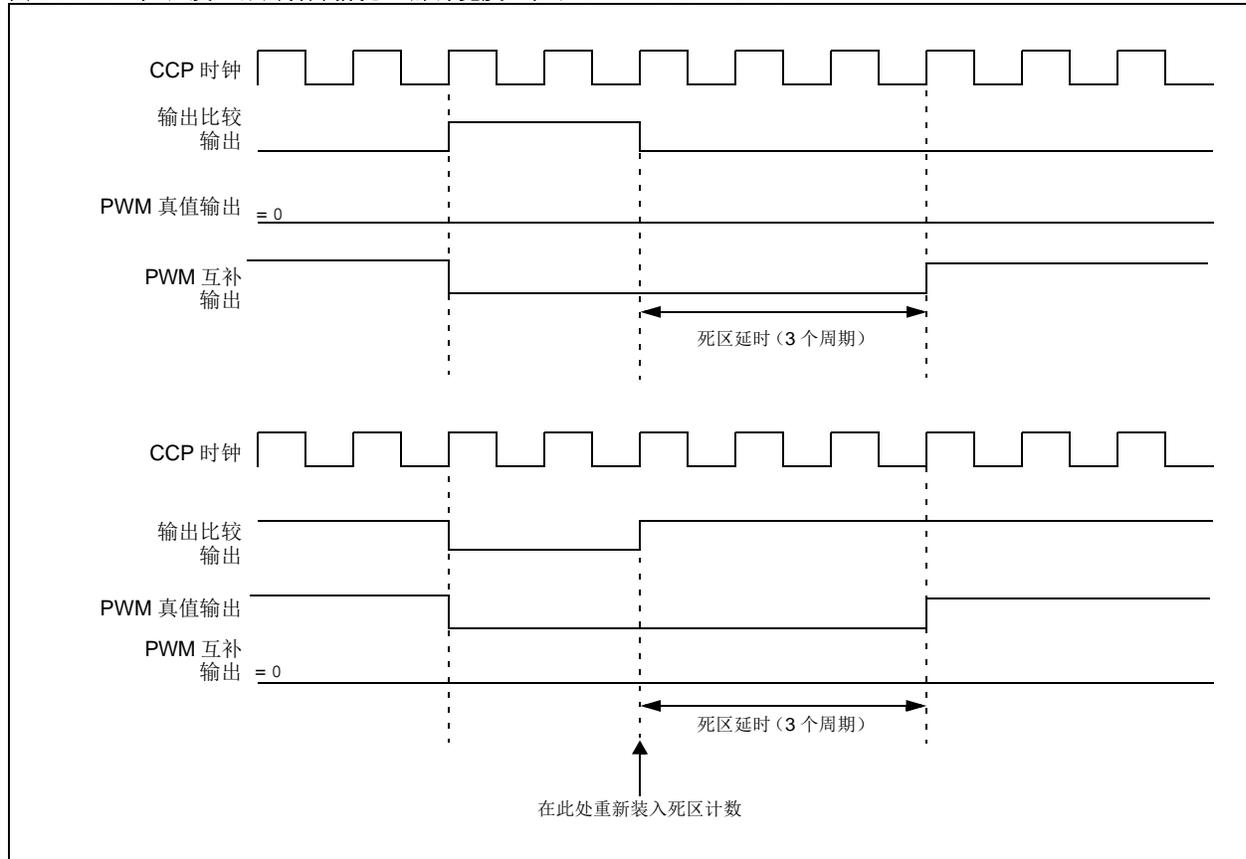


死区发生具有 3 种特殊情况:

1. 当输入信号脉冲宽度等于或小于设定的死区值时, 将不会获取所需输出结果。图 30-36 描述了两种场景, 即输入信号的占空比接近 0% 和接近 100% 时。
2. 当输入占空比设置为 0% 时, 两个输出在死区延时期间均驱动为无效。输入信号上不会产生任何跃变。PWM 真值输出保持低电平, PWM 互补信号保持高电平。
3. 当输入占空比设置为 100% 时, 两个输出在死区延时期间均驱动为无效。输入信号上不会产生任何跃变。PWM 真值输出保持高电平, PWM 互补信号保持低电平。

当输入占空比接近 0% 或 100% 时, 死区延时会在输入信号脉冲宽度中占据更大的部分。这种效应会导致所请求的占空比与实际系统响应之间出现非线性行为。对这种非线性敏感的系统必须避免接近 0% 和 100% 占空比的区域, 或使用输出反馈来补偿占空比。

图 30-36: 死区发生器的特殊情况 (脉冲宽度 < 死区, DT<5:0> = 03h)



30.7.7.10.1 半桥模式的死区延时产生

当模块在半桥模式下工作时，OCxA 输出引脚信号是来自死区延时发生器的同相输出（见图 30-29）。OCxB 输出引脚信号是来自死区延时发生器的反相（互补）输出。这两个信号输出的极性控制在死区延时发生器控制块之后提供。

在这种半桥模式下，OCxC/OCxD 和 OCxE/OCxF 输出引脚会复制 OCxA/OCxB 引脚对的输出和端口控制信号。这使用户可以通过使用 OCxEN 位将互补输出信号重定位至另一个引脚对。

30.7.7.10.2 有刷直流模式的死区延时产生

对于有刷直流工作模式，死区延时发生器可用于在翻转或方向更改期间可选地消隐 OCxA、OCxB、OCxC 和 OCxD 输出引脚信号。

在有刷直流模式下，OCxE 和 OCxF 输出引脚不受模块控制。它们的输出使能信号应驱动为低电平。

30.7.7.10.3 推挽模式的死区延时产生

当 CCP 模块在推挽模式下工作时，死区延时发生器可用于在时基周期边界处可选地消隐 OCx 输出引脚。这 6 个信号输出的极性控制在死区延时发生器控制块之后提供。

30.7.7.10.4 输出扫描模式的死区延时产生

输出扫描模式（OUTM<2:0> = 110）不使用死区延时发生器。

30.7.7.11 自动关断控制

自动关断控制逻辑的主要功能是在驱动外部电源电路时将模块输出引脚置为安全的状态。此外，自动关断功能还可以用于基于外部事件将 CCP 模块的输出引脚置为特定状态。

自动关断控制作为时基门控的一部分而实现（见第 30.3.1 节“门控逻辑”）。用户必须使用 ASDG<7:0> 控制位（CCPxCON2<7:0>）选择自动关断的输入源。自动关断的可用源取决于器件，通常包括诸如比较器输出、I/O 引脚、软件控制（即，SSDG 位）等等之类的源。SSDG 控制位除外（它为高电平有效），来自关断源的低电平输出会将模块 OCx 引脚置为关断状态。自动关断事件是电平敏感型，不是边沿触发型。比较器输出和其他关断源不与系统时钟进行同步，以使 CCP 模块可以立即响应关断输入信号。

发生关断时，模块端口引脚会被置为选定的输出状态。

30.7.7.11.1 自动关断引脚状态

输出引脚的状态由 PSSACE<1:0> 和 PSSBDF<1:0> 控制位（CCPxCON3<19:18,17:16>）控制。PSSACE_x 位会影响 OCxA、OCxC 和 OCxE（高侧）输出引脚的状态。PSSBDF_x 位会影响 OCxB、OCxD 和 OCxF（低侧）输出引脚的状态。这些控制位用于选择 I/O 引脚是驱动为无效，驱动为有效还是置为高阻态。

注： 如果清零了关联的转向控制位（OCxEN），则输出引脚不会受自动关断事件影响。

30.7.7.11.2 软件关断

用户应用程序可以在任意时刻通过将 SSDG 控制位（CCPxCON2<12>）置 1 来调用关断事件。该位的行为与外部关断源完全相同，只是控制位的极性相反。每当 SSDG 位置 1 时，将会产生关断事件。模块输出引脚会变为为其设定的关断状态，并保持在该状态，直到用软件清零 SSDG 位为止。软件关断功能可自己单独使用，也可以与外部源同时使用。

注 1： 如果未使能自动重启，则用户可能还需要清零 ASEVT 状态位。

2： 任何通过 ASDG_x 位和 SSDG 软件关断位选择的已使能关断源都具有高于软件写入 ASEVT 位的优先级。除非所有关断源均为无效，否则将无法退出故障条件。

30.7.7.11.3 自动关断状态

ASEVT 状态位（CCPxSTAT<4>）指示关断事件的状态。如果清零了 ASEVT 位，则与 CCP 模块关联的输出引脚将具有正常活动。

如果 ASEVT 位置 1，则输出引脚将驱动为其关断状态或保持高阻态。此外，ASEVT 位还可用作控制位，以手动复位关断条件，如第 30.7.7.11.4 节“自动重启使能”所述。

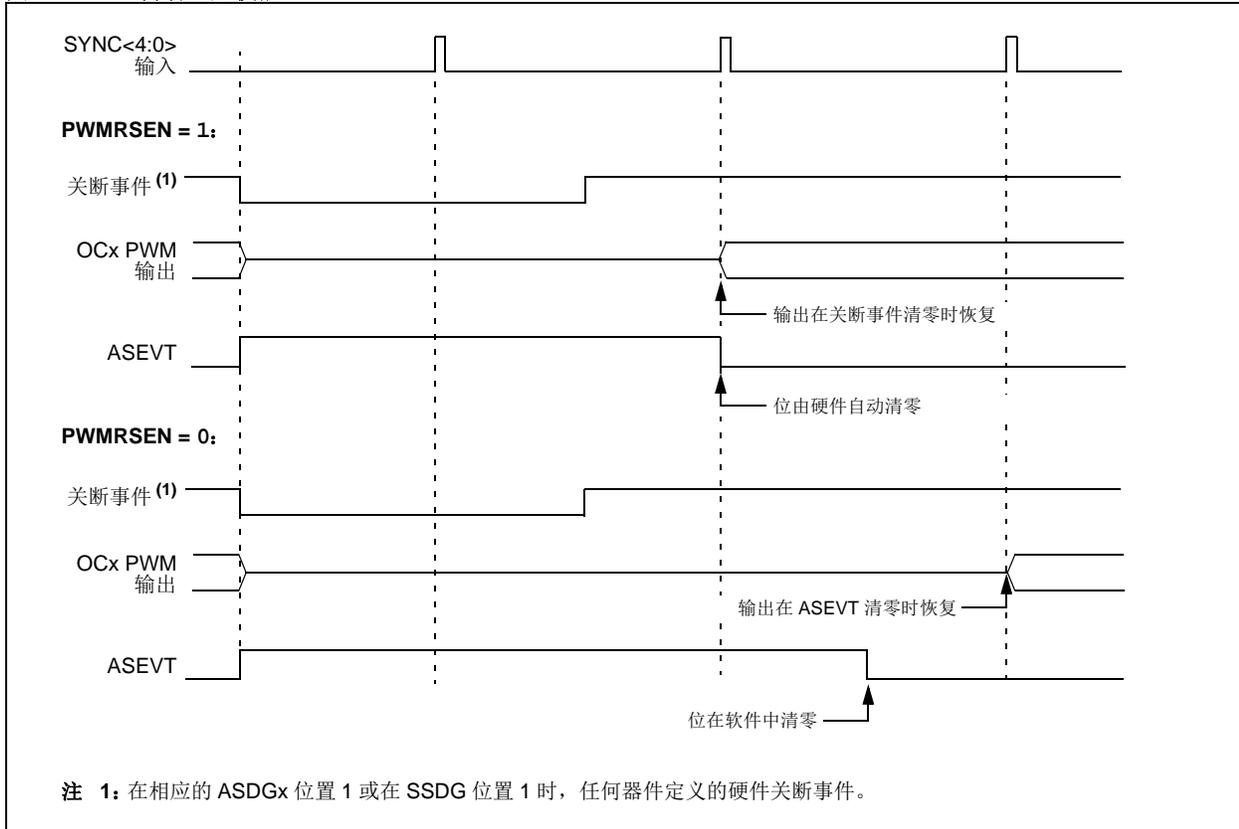
30.7.7.11.4 自动重启使能

PWMRSEN 位 (CCPxCON2<15>) 控制如何结束关断状态。如果 PWMRSEN = 0, 模块将等待, 直到 ASEVT 状态位在用户软件中清零为止。只有 ASEVT 位清零, 且外部关断源信号不再存在时, 才能恢复正常的输出引脚活动。如果外部关断源信号仍然有效, 用户将无法清零 ASEVT 位。

如果 PWMRSEN = 1, 当外部关断源信号变为无效且下一个 PWM 周期开始 (即, 当由 SYNC<4:0> 选择的同步源置为有效) 时, 将自动恢复正常的输出引脚活动。此时, ASEVT 位将在硬件中自动清零。如果关断在新周期开始时仍然有效, 则整个周期都会被禁止, 从而消除狭窄的毛刺脉冲。然后, 将在下一个周期重新开始 PWM 输出。

如果 PWMRSEN = 0, 则在出现关断条件时, PWM 将保持空闲状态, 直到用户手动重启为止。模块可以通过用软件清零 ASEVT 位而重启。

图 30-37: 自动重启使能



30.7.7.11.5 门控自动关断模式

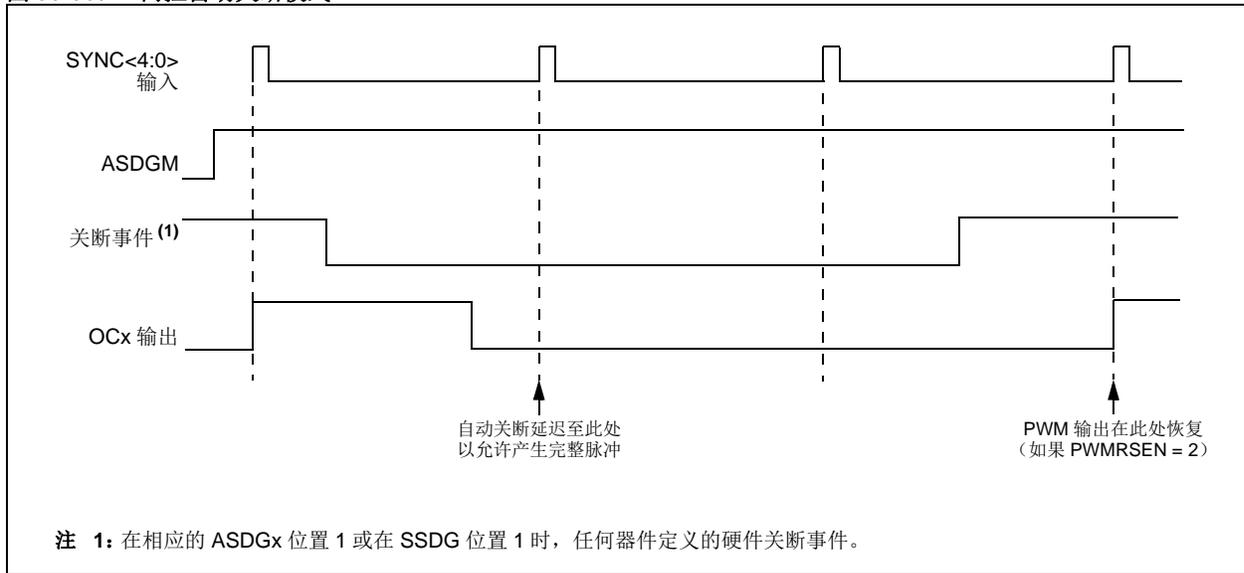
对于某些类型的电源控制应用，延迟自动关断输入的生效时间会非常有用。ASDGM 控制位 (CCPxCON2<14>) 可启用门控关断操作。当 ASDGM 置 1 时，自动关断信号输入的作用直到下一个 PWM 周期边界处才会生效。这使 PWM 发生器可以产生为当前周期设定的完整脉冲。

脉冲将在下一个周期开始时由硬件终止。脉冲将在关断事件结束之后的下一个 PWM 周期恢复。

注： 只有 PWMRSEN = 1 时，脉冲才会自动恢复。如果 PWMRSEN = 0，则必须用软件清零 ASEVT 状态位，脉冲才会恢复。

门控模式支持自动关断模式与比较器一起使用，以实现“脉冲跳跃”或“门控振荡器”开关电源 (Switch Mode Power Supply, SMPS)。电感使用每个脉冲充电一段固定的时间周期，将特定的电量装入电源。如果输出电压 (电流) 足够高，则比较器会对脉冲进行门控。

图 30-38: 门控自动关断模式



注 1: 如果未使能自动重启，则应用程序可能还需要清零 ASEVT 位。

2: 任何通过 ASDGx 位和 SSDG 软件关断位选择的已使能关断源都具有高于软件写入 ASEVT 位的优先级。除非所有关断源均为无效，否则将无法退出故障条件。

30.7.7.12 输出极性控制

输出引脚的极性由 POLACE 和 POLBDF 控制位 (CCPxCON3<21:20>) 控制。POLACE 位可更改 OCxA、OCxC 和 OCxE 引脚的输出极性；POLBDF 位控制 OCxB、OCxD 和 OCxF 引脚的极性。

输出极性控制将在死区控制和自动关断逻辑之后应用于输出信号。极性控制位对于模块的所有输出比较和 PWM 模式有效。

30.8 输入捕捉模式

当 $CCSEL = 1$ 时，模块配置为输入捕捉模式。该模式用于在输入引脚上有事件发生时，捕捉来自独立时基的定时器值。该模式在需要频率（时间周期）和脉冲测量的应用中很有用。

输入捕捉模式使用 $CCPxTMR$ 寄存器作为专用的 16/32 位同步递增定时器，用于进行事件捕捉。当发生捕捉事件时，该值将写入 FIFO 缓冲区。此外也可以读取内部值，读取操作与 $CCPxTMR$ 寄存器之间有一个同步延时。

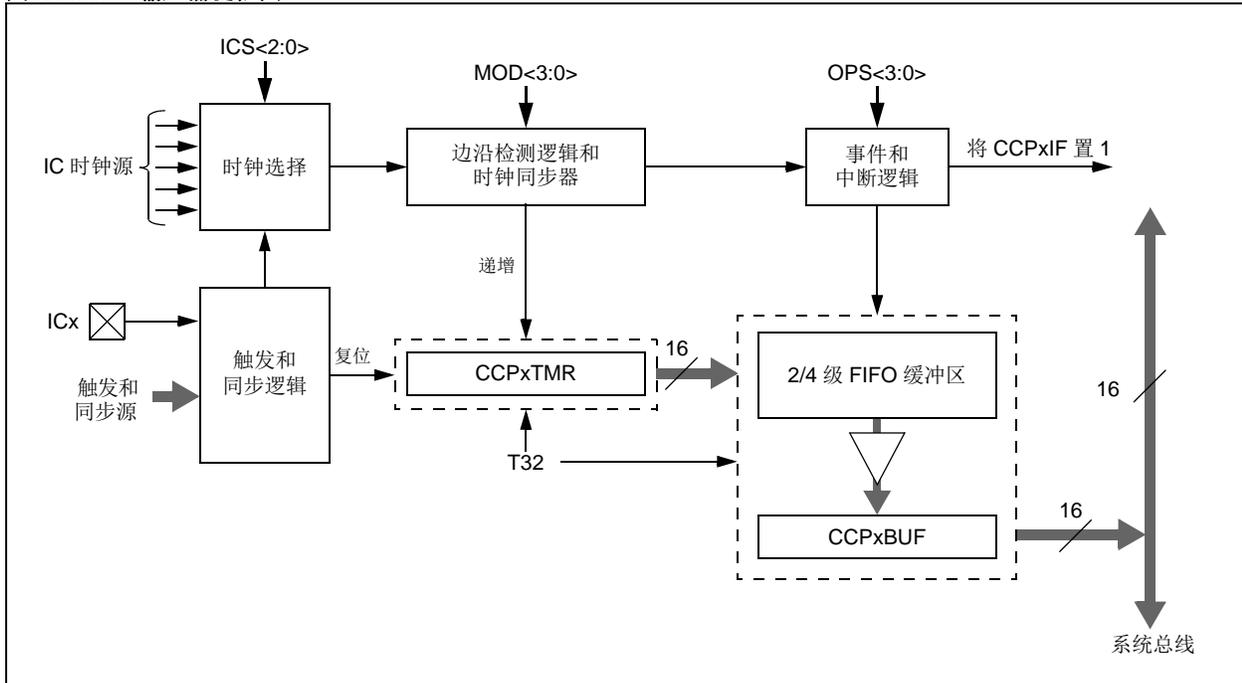
当 $CCSEL$ 置 1 时，输入捕捉模式是唯一可用的主要模式。 $T32$ 和 $MOD<3:0>$ 位决定各种捕捉模式，如表 30-7 所示。

图 30-39 给出了输入捕捉模式的简化框图。

表 30-7: 捕捉模式

T32 ($CCPxCON1<5>$)	MOD<3:0> ($CCPxCON1<3:0>$)	工作模式
0	0000	边沿检测 (16 位捕捉)
1	0000	边沿检测 (32 位捕捉)
0	0001	每个上升沿 (16 位捕捉)
1	0001	每个上升沿 (32 位捕捉)
0	0010	每个下降沿 (16 位捕捉)
1	0010	每个下降沿 (32 位捕捉)
0	0011	每个上升/下降沿 (16 位捕捉)
1	0011	每个上升/下降沿 (32 位捕捉)
0	0100	每次出现第 4 个上升沿 (16 位捕捉)
1	0100	每次出现第 4 个上升沿 (32 位捕捉)
0	0101	每次出现第 16 个上升沿 (16 位捕捉)
1	0101	每次出现第 16 个上升沿 (32 位捕捉)

图 30-39: 输入捕捉框图



30.8.1 初始化

由于模块可用于输入捕捉 / 输出比较 / PWM 模式，因此选择所需的正确操作是第一项任务。最佳做法是清零所有关联的控制寄存器。

当 CCP 模块复位或被禁止 (ON = 0) 时：

- ICOV 和 ICBNE 状态标志清零
- CCPxBUF 及其 FIFO 缓冲区清零
- CCPxTMR 复位为零
- 捕捉预分频计数器复位为零
- 用于中断产生的捕捉事件计数器复位为零

30.8.1.1 模式选择

对于定时器和输出捕捉 / PWM 模式，MOD<3:0> 位用于选择捕捉模式和预分频器选项。为了避免产生意外中断，在更改捕捉模式时，请总是通过清零 ON 位禁止模块。建议在使能模块之前，先在单个操作中将 CCSEL 位置 1 并配置 MOD<3:0> 位。

30.8.1.2 定时器时钟源选择

PIC32 系列器件可能有一个或多个输入捕捉通道。每个通道可以通过使用 CLKSEL<2:0> 位 (CCPxCON1<10:8>)，在 8 个时钟源中选择一个作为其时基，如第 30.3 节“[时基发生器](#)”所述。模块可被设置为使用系统时钟源 (Fosc/2)，或使用在 TxCK 引脚上外接的时钟源，但条件是要使能定时器的同步模式。输入捕捉模式操作应选择输入捕捉 x (Input Capture x, ICx) 引脚。建议在使能模块之前选择时钟源，并且在工作期间不进行更改。

关于可用的定时器输入，请参见具体器件的数据手册。

30.8.1.2.1 32 位输入捕捉支持

输入捕捉模式能够使用 32 位时基进行工作。32 位模式通过将 T32 位置 1 来进行选择。所有输入捕捉功能在 16 位和 32 位模式之间都是相同的，只是在 32 位操作下存在以下变化：

- CCPxTMR 是一个 32 位寄存器
- CCPxBUF 是一个 32 位寄存器
- FIFO 缓冲区在 32 位工作模式下只有两级可用

[例 30-5](#) 给出了设置输入捕捉模式的典型过程。

例 30-5: 输入捕捉模式设置 (每个上升沿, 16 位模式)

```
CCP1CON1bits.CCSEL = 1;           // Input capture mode
CCP1CON1bits.CLKSEL = 0;          // Set the clock source (Tcy)
CCP1CON1bits.T32 = 0;             // 16-bit Dual Timer mode
CCP1CON1bits.MOD = 1;             // Capture ever rising edge of the event
CCP1CON2bits.ICSEL = 0;           // Capture rising edge on the Pin
CCP1CON1bits.IOPS = 0;            // Interrupt on every input capture event
CCP1CON1bits.TMRPS = 0;           // Set the clock prescaler (1:1)
CCP1CON1bits.ON = 1;              // Enable CCP/input capture
```

30.8.2 捕捉事件模式

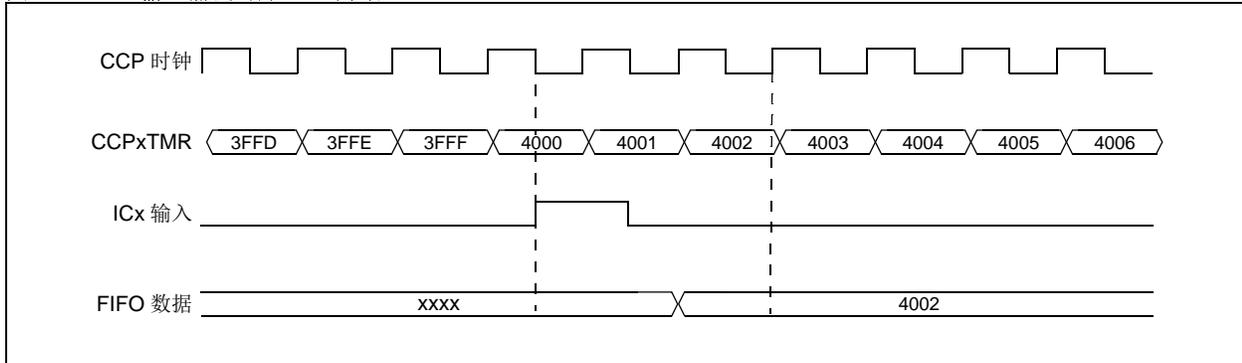
模块可以在发生任何以下 ICx 引脚跃变时捕捉到定时器值：

- 每个上升沿 ($MOD<3:0> = 0001$)
- 每个下降沿 ($MOD<3:0> = 0010$)
- 每个上升沿和下降沿 ($MOD<3:0> = 0000、0011$)

不使用预分频器时，由于输入捕捉引脚在定时器时钟的下降沿进行采样，所以捕捉脉冲宽度必须大于定时器时钟周期再加上某个裕度。

由于内部同步要求，捕捉的定时器值最高将为实际捕捉边沿事件后的 3 个 CCP 时钟周期，如图 30-40 所示。

图 30-40： 输入捕捉时序（上升沿）

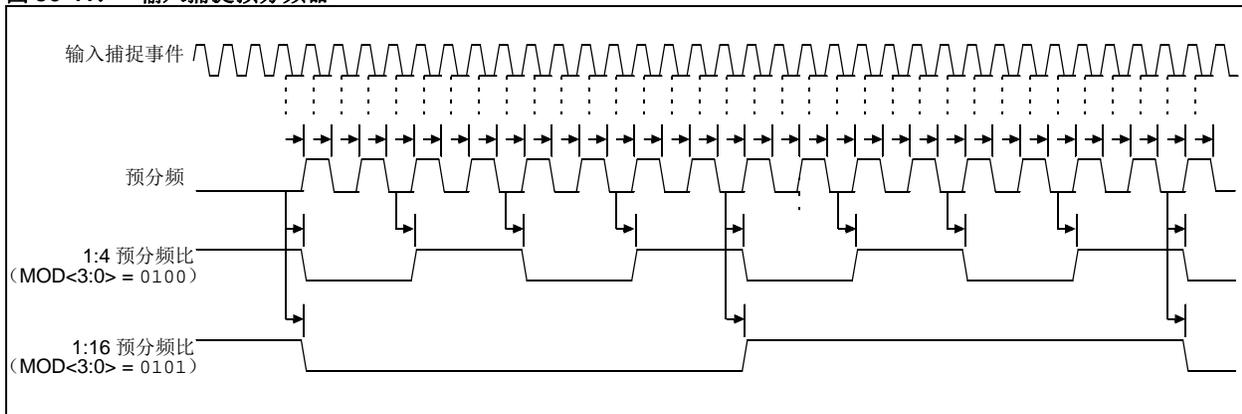


30.8.2.1 输入捕捉预分频器

通过使用输入预分频器，输入捕捉模块可以在 ICx 输入引脚每次出现第 4 个边沿 ($MOD<3:0> = 0100$) 或第 16 个边沿 ($MOD<3:0> = 0101$) 时捕捉定时器值。

捕捉脉冲宽度要求将不同于简单捕捉模式的要求。关于确切规范，请参见具体器件的数据手册。由于内部同步要求，捕捉的定时器值最高将为边沿捕捉事件之后的 1 至 2 个定时器时钟周期，如图 30-41 所示。

图 30-41： 输入捕捉预分频器



30.8.2.2 边沿检测 (霍尔传感器) 模式

边沿检测模式 (MOD<3:0> = 0000) 的操作与捕捉每个边沿模式 (MOD<3:0> = 0011) 相同, 只是在输入捕捉缓冲区溢出 (Input Capture Buffer Overflow, ICOV) 状态标志置 1 时, 捕捉中断事件不会停止。这使连续的捕捉事件流可以触发中断事件, 而无需不断地清空 FIFO。

30.8.2.3 输入捕捉缓冲区

输入捕捉 FIFO 缓冲区最高为 4 级深, 具体取决于所选的捕捉模式。对于 16 位定时器捕捉, FIFO 具有 4 级 (16 位宽); 对于 32 位定时器捕捉, 具有 2 级 (32 位宽)。用户可以选择产生 CPU 中断所需的捕捉事件数。

有两个状态标志提供 FIFO 缓冲区的状态。ICBNE 状态位 (CCPxSTAT<0>) 指示发生了至少一个捕捉事件。ICOV 状态位 (CCPxSTAT<1>) 指示存在的事件多于缓冲区的当前深度 (16 位模式为 4, 32 位模式为 2)。对于 16 位捕捉操作和 32 位捕捉操作, 这些状态标志以相同的方式工作。

当 CCP 模块处于复位或不处于捕捉模式时:

- ICOV 状态标志清零
- ICBNE 状态标志清零
- FIFO 被标记为空
- FIFO 缓冲区的读操作将返回 0

ICBNE 状态标志位将在第一个捕捉事件发生时被置 1, 并且一直保持置 1 状态, 直到所有的捕捉事件都已从 FIFO 中读出。例如, 如果发生了 3 个捕捉事件, 则必须对捕捉 FIFO 缓冲区进行 3 次读操作后才能将 ICBNE 标志清零。FIFO 缓冲区的每次读操作都将使余下的字移动到 FIFO 的下一个可用栈顶单元。

如果 FIFO 缓冲区中已填满捕捉事件, 并且在读取 FIFO 之前发生了另一个捕捉事件, 则将发生溢出条件, ICOV 位会置 1。此外, 导致溢出的捕捉事件不会被记录下来, 后续的捕捉事件将不会被放入 FIFO, 直到通过完全清空 FIFO 而清除溢出条件为止。

当模块不处于输入捕捉模式或当使能边沿检测模式 (MOD<3:0> = 0000) 时, 不会发生溢出条件。

可以通过以下方法之一清除溢出条件:

1. 通过清零 ON 位而禁止该模块。
2. 读取输入捕捉缓冲区, 直到 ICBNE = 0 (对于 32 位捕捉为 2 次, 对于 16 位捕捉为 4 次)。
3. 用软件清零 ICOV 位。这实际上会通过将数据指针复位为指向 FIFO 缓冲区起始位置而丢弃 FIFO 先前存储的所有数据。用软件清零 ICOV 还会导致 ICBNE 位自动清零。
4. 执行器件复位。

在清除溢出条件之后, ICOV 和 ICBNE 状态标志将被清零, 捕捉通道将恢复正常工作。如果禁止了模块, 然后稍后在输入捕捉模式下重新使能, FIFO 缓冲区内容将是未定义的, 读操作将产生不确定的结果。

如果进行了最后一次读取, 并且之后未接收到任何新的捕捉事件, 则执行 FIFO 读操作时, FIFO 读指针和写指针将指向 FIFO 的第一个缓冲区单元。读取 FIFO 将返回在第一个缓冲区单元中存放的值。

每次 CPU 读取缓冲区结果的最高字时, FIFO 指针都会发生调整。这使 16 位 CPU 可以读取 32 位输入捕捉的结果。

30.8.3 输入捕捉中断

当处于输入捕捉模式时，模块具有在发生捕捉事件时产生中断的能力。捕捉事件定义为向 FIFO 写入一个定时器值。

OPS<3:0> 控制位 (CCPxCON1<27:24>) 用于选择中断后分频比，即指定在产生中断之前必须发生的捕捉事件数。选项范围从每次捕捉产生中断到每 4 次捕捉产生中断。第一个捕捉事件定义为在模式从禁止状态 (ON = 0) 发生改变或 ICBNE = 0 之后发生的捕捉事件。

在缓冲区发生溢出时，捕捉事件会停止，并且除非 OPS<3:0> = 0000 (每次捕捉时产生中断)，否则中断也会停止。通过读取而清空 FIFO 也会清零内部中断计数器，并可能会影响何时产生中断。

应用程序通常使用输入捕捉 x 引脚作为辅助外部中断源。在边沿检测模式下，无论 FIFO 是否溢出，都根据 OPS<3:0> 来产生中断。不需要通过对输入捕捉缓冲区进行虚读操作来清除事件，从而防止溢出和禁止所有未来中断。

例如，假设 OPS<3:0> = 0001，即指定每发生 2 次捕捉事件产生中断。以下事件序列将产生单个 CCPxIF，如下所示：

1. 开启模块；事件计数 = 0。
2. 捕捉第一个事件；FIFO 包含 1 个条目，事件计数 = 1。
3. 读取 FIFO；FIFO 为空，事件计数 = 0。
4. 捕捉第二个事件；FIFO 包含 1 个条目，事件计数 = 1。
5. 捕捉第三个事件；FIFO 包含 2 个条目，事件计数 = 2，将 CCPxIF 置 1。
6. 在中断置 1 时清零中断计数 (事件计数 = 0)。
7. 捕捉第四个事件；FIFO 包含 3 个条目，事件计数 = 1。
8. 读取 FIFO 三次；FIFO 为空，中断计数 = 0。
9. 捕捉第 5 个事件；FIFO 包含 1 个条目，事件计数 = 1。
10. 读取 FIFO；FIFO 为空，事件计数 = 0。

30.8.3.1 输入捕捉模式下的定时器中断

在输入捕捉模式下工作时，模块会同时产生定时器中断 (CCTxIF) 和捕捉中断 (CCPxIF)。但是，由于没有可用于设置计数周期的周期寄存器，定时器中断只在定时器计满返回 (从 FFFFh 至 0000h) 时产生。如果需要较短的定时器计数周期，则可以使用第二个 MCCP 模块或外部定时器来为输入捕捉时基提供同步源。

30.8.4 通过同步和触发进行输入捕捉操作

默认情况下，处于输入捕捉模式下的 MCCP 模块使用自由运行定时器工作。在输入捕捉模式下，CCPxPR 寄存器不可用于设置不同的定时器周期。建议将 SYNC<4:0> 保持配置为 11111，以维持自由运行定时器。

在以下任一条件下，定时器将保持为 0000h：

- 触发操作已使能 (TRIGEN = 1)，并且尚未发生触发事件 (CCPTRIG = 0)。
- 已选择了外部同步源 (SYNC<4:0> 具有 11111 之外的值)，它尚未使能。

在任一情况下，都会发生输入捕捉输入事件；但是，FIFO 将总是捕捉 0000h 值。因此，建议不要进行触发操作和外部同步操作。

30.8.4.1 输入捕捉信号门控

输入捕捉源可以通过软件或硬件进行门控，从而允许窗口捕捉测量。该功能可以在传感应用中提供抗噪声功能。

ICDIS 位 (CCPxSTAT<2>) 提供输入信号门控功能的状态。当 ICDIS 位清零时，将允许由边沿检测逻辑产生的捕捉事件。当 ICDIS 位置 1 时，将禁止来自边沿检测逻辑的事件。

时基门控逻辑用于输入捕捉信号门控（关于更多信息，请参见第 30.3.1 节“门控逻辑”）。ASDG<7:0> 控制位 (CCPxCON2<7:0>) 选择一个或多个用于清零 ICDIS 状态 / 控制位的输入源。SSDG 位 (CCPxCON2<12>) 也可用于在软件中对输入捕捉信号进行手动门控。

ASDGx 源和 SSDG 位的行为取决于门控源模式，该模式用 ICGSM<1:0> 控制位 (CCPxCON2<23:22>) 选择。有 3 种不同选项可用：

- 当 ICGSM<1:0> = 00 时，门控为电平敏感型。来自门控源的低电平输入会禁止后续的捕捉事件，ICDIS 位将置 1 来反映这一点。高电平输入会使能后续的捕捉事件，ICDIS 位将清零来反映这一点。
- 当 ICGSM<1:0> = 01 时，门控在门控源的上升沿出现；ICDIS 位会清零，禁止后续的捕捉事件。这是一种单脉冲模式；来自门控源的后续边沿将没有任何作用。
- 当 ICGSM<1:0> = 10 时，门控在门控源的下降沿出现；ICDIS 位会置 1，使能后续的捕捉事件。这是一种单脉冲模式；来自门控源的后续边沿将没有任何作用。

图 30-42 显示了门控捕捉事件的时序。输入事件在时钟源的下降沿进行采样。该示例假定输入捕捉模块配置为捕捉每个上升沿和下降沿 (MOD<3:0> = 0011)。

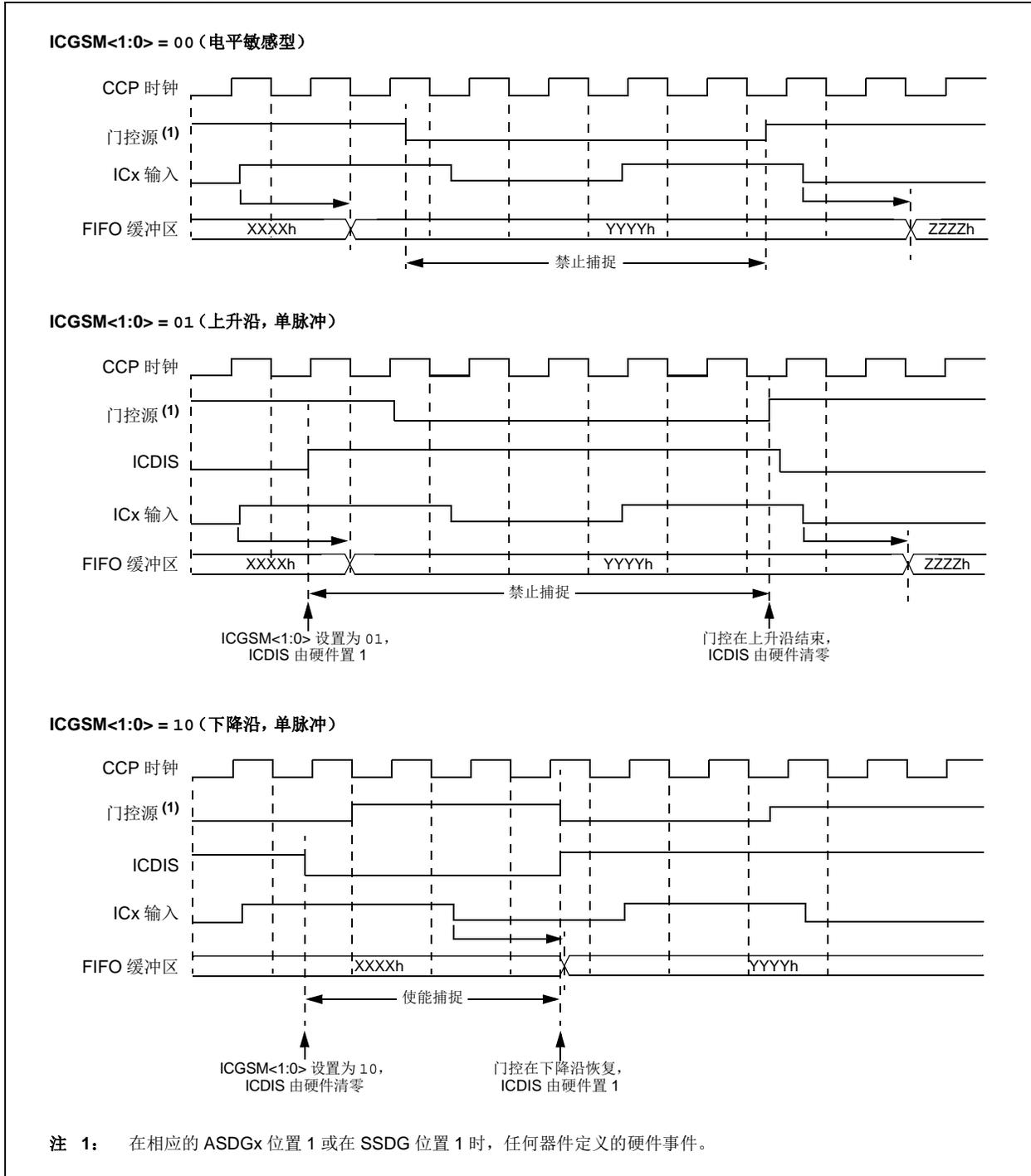
在单脉冲模式下，边沿检测逻辑设置为查找相应的边沿事件；ICDIS 位将保持置 1 或清零（取决于模式），直到发生该类型的事件为止。在门控事件之后，用户可以通过重新写入 ICGSM<1:0> 重新激活门控逻辑。写入这些位的这种操作（即使是相同的值）会复位门控信号边沿检测逻辑，并且还会将 ICDIS 状态位复位为相应的值。

要使用输入捕捉门控：

1. 选择并配置门控源。
2. 使用 ASDG<7:0> 位使能相应的门控信号源；或者，在仅软件控制的事件期间置 1 或清零 SSDG 位。
3. 使用 ICGSM<1:0> 选择门控模式。
4. 使用 MOD<3:0> 和 ICS<2:0> 控制位为模块配置所需的输入捕捉模式和输入源。模块现在将对于门控事件激活。

在 ICDIS 清零之后，下一个有效上升或下降输入信号边沿（取决于捕捉模式）将会触发捕捉事件。

图 30-42: 门控输入捕捉示例



30.9 休眠和空闲模式期间的操作

30.9.1 空闲模式

模块在空闲模式下的行为由 SIDL 位 (CCPxCON1<13>) 决定。如果 SIDL 清零, 在空闲模式下, 模块将继续工作。如果 SIDL 置 1, 则在器件进入空闲模式时, 模块会被禁止。如果调用空闲模式时, 模块正在执行操作, 结果与休眠模式下类似。

30.9.2 休眠模式

模块在休眠模式下的行为由 CCPSLP 位 (CCPxCON1<12>) 决定。如果 CCPSLP 置 1, 假定所选的时钟源保持可用, 模块将在休眠模式下继续工作。要使模块在休眠模式下工作, TMRSYNC 位必须保持清零。

当 CCPSLP 清零, 且器件进入休眠模式时, 模块将被禁止。

30.9.2.1 触发操作和休眠模式

当模块配置为触发操作时, 从外部源接收的触发信号也可以唤醒模块及其时基时钟源。模块必须先请求时基时钟源, 才会开始触发操作。从外部源接收到触发事件时, CCP 模块将使能时基的选定时钟源。当时钟源变为可用时, 模块将开始触发操作。

如果选择了单脉冲触发模式, 则当 CCPTRIG 状态位在硬件中清零时, 将会禁止时基时钟源。时基将保持禁止状态, 直到接收到新的触发信号。

此外, 还可以通过使用低功耗时钟源工作的内部源产生触发信号。

如果使能了休眠模式操作, 则当器件进入休眠模式时, 模块将继续请求所配置的时钟源。

30.10 复位的影响

器件复位会将所有器件寄存器强制设为其复位状态; 这会禁止模块并将其恢复为其默认配置 (16 位定时器)。所有缓冲区和地址寄存器均初始化为 0000h, 所有状态标志均复位。

默认情况下, 与 OCxA (MCCP 模块) 或 OCx (SCCP 模块) 关联的引脚会发生复位, 输出比较功能将控制该引脚。但是, 在模块禁止时, 它没有任何作用。

30.11 相关应用笔记

本节列出了与手册本章内容相关的应用笔记。这些应用笔记可能并不是专为 PIC32 器件系列而编写的，但其概念是相近的，通过适当修改并受到一定限制即可使用。当前与捕捉 / 比较 / PWM / 定时器（MCCP 和 SCCP）相关的应用笔记包括：

标题	应用笔记编号
----	--------

目前没有相关的应用笔记。

注： 如需获取更多 PIC32 系列器件的应用笔记和代码示例，请访问 Microchip 网站 (www.microchip.com)。

30.12 版本历史

版本 A (2016 年 4 月)

本文档的初始版本。

注:

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字千年版权法案 (Digital Millennium Copyright Act)》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适用性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和 / 或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。除非另外声明，在 Microchip 知识产权保护下，不得暗或以其他方式转让任何许可证。

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。Microchip 的 PIC[®] MCU 与 dsPIC[®] DSC、KEELOQ[®] 跳码器件、串行 EEPROM、单片机外设、非易失性存储器 and 模拟产品严格遵守公司的质量体系流程。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949 ==

商标

Microchip 的名称和徽标组合、Microchip 徽标、AnyRate、AVR、AVR 徽标、AVR Freaks、BeaconThings、BitCloud、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、Heldo、JukeBlox、KEELOQ、KEELOQ 徽标、Kleer、LANCheck、LINK MD、maXStylus、maXTouch、MediaLB、megaAVR、MOST、MOST 徽标、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32 徽标、Prochip Designer、QTouch、RightTouch、SAM-BA、SpyNIC、SST、SST 徽标、SuperFlash、tinyAVR、UNI/O 及 XMEGA 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

ClockWorks、The Embedded Control Solutions Company、EtherSynch、Hyper Speed Control、HyperLight Load、IntelliMOS、mTouch、Precision Edge 和 Quiet-Wire 均为 Microchip Technology Inc. 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BodyCom、chipKIT、chipKIT 徽标、CodeGuard、CryptoAuthentication、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet 徽标、Mindi、MiWi、motorBench、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PureSilicon、QMatrix、RightTouch 徽标、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 为 Microchip Technology Inc. 在美国的服务标记。

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. & KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2018, Microchip Technology Inc. 版权所有。

ISBN: 978-1-5224-2378-2

全球销售及服务中心

美洲

公司总部 **Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持:
<http://www.microchip.com/support>

网址: www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

奥斯汀 Austin, TX
Tel: 1-512-257-3370

波士顿 Boston
Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago
Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

达拉斯 Dallas
Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit
Novi, MI
Tel: 1-248-848-4000

休斯敦 Houston, TX
Tel: 1-281-894-5983

印第安纳波利斯 Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453
Tel: 1-317-536-2380

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608
Tel: 1-951-273-7800

罗利 Raleigh, NC
Tel: 1-919-844-7510

纽约 New York, NY
Tel: 1-631-435-6000

圣何塞 San Jose, CA
Tel: 1-408-735-9110
Tel: 1-408-436-4270

加拿大多伦多 Toronto
Tel: 1-905-695-1980
Fax: 1-905-695-2078

亚太地区

中国 - 北京
Tel: 86-10-8569-7000

中国 - 成都
Tel: 86-28-8665-5511

中国 - 重庆
Tel: 86-23-8980-9588

中国 - 东莞
Tel: 86-769-8702-9880

中国 - 广州
Tel: 86-20-8755-8029

中国 - 杭州
Tel: 86-571-8792-8115

中国 - 南京
Tel: 86-25-8473-2460

中国 - 青岛
Tel: 86-532-8502-7355

中国 - 上海
Tel: 86-21-3326-8000

中国 - 沈阳
Tel: 86-24-2334-2829

中国 - 深圳
Tel: 86-755-8864-2200

中国 - 苏州
Tel: 86-186-6233-1526

中国 - 武汉
Tel: 86-27-5980-5300

中国 - 西安
Tel: 86-29-8833-7252

中国 - 厦门
Tel: 86-592-238-8138

中国 - 香港特别行政区
Tel: 852-2943-5100

中国 - 珠海
Tel: 86-756-321-0040

台湾地区 - 高雄
Tel: 886-7-213-7830

台湾地区 - 台北
Tel: 886-2-2508-8600

台湾地区 - 新竹
Tel: 886-3-577-8366

亚太地区

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733

印度 India - Bangalore
Tel: 91-80-3090-4444

印度 India - New Delhi
Tel: 91-11-4160-8631

印度 India - Pune
Tel: 91-20-4121-0141

日本 Japan - Osaka
Tel: 81-6-6152-7160

日本 Japan - Tokyo
Tel: 81-3-6880-3770

韩国 Korea - Daegu
Tel: 82-53-744-4301

韩国 Korea - Seoul
Tel: 82-2-554-7200

马来西亚 Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870

菲律宾 Philippines - Manila
Tel: 63-2-634-9065

新加坡 Singapore
Tel: 65-6334-8870

泰国 Thailand - Bangkok
Tel: 66-2-694-1351

越南 Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

欧洲

奥地利 Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

芬兰 Finland - Espoo
Tel: 358-9-4520-820

法国 France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 Germany - Garching
Tel: 49-8931-9700

德国 Germany - Haan
Tel: 49-2129-3766400

德国 Germany - Heilbronn
Tel: 49-7131-67-3636

德国 Germany - Karlsruhe
Tel: 49-721-625370

德国 Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

德国 Germany - Rosenheim
Tel: 49-8031-354-560

以色列 Israel - Ra'anana
Tel: 972-9-744-7705

意大利 Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

意大利 Italy - Padova
Tel: 39-049-7625286

荷兰 Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

挪威 Norway - Trondheim
Tel: 47-7289-7561

波兰 Poland - Warsaw
Tel: 48-22-3325737

罗马尼亚 Romania - Bucharest
Tel: 40-21-407-87-50

西班牙 Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

瑞典 Sweden - Gothenberg
Tel: 46-31-704-60-40

瑞典 Sweden - Stockholm
Tel: 46-8-5090-4654

英国 UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820