

第 35 章 带专用定时器的输出比较

目录

本章包括下列主题：

35.1 简介.....	35-2
35.2 输出比较寄存器.....	35-3
35.3 工作模式.....	35-7
35.4 节能状态下的输出比较操作.....	35-30
35.5 I/O 引脚控制.....	35-30
35.6 寄存器映射.....	35-31
35.7 电气规范.....	35-32
35.8 设计技巧.....	35-33
35.9 相关应用笔记.....	35-34
35.10 版本历史.....	35-35

35.2 输出比较寄存器

每路输出比较通道均包含以下寄存器：

- OCxCON1 和 OCxCON2——用于输出比较通道的控制寄存器
- OCxR——用于输出比较通道的数据寄存器
- OCxRS——用于输出比较通道的辅助数据寄存器
- OCxTMR——内部时基寄存器

9 路输出比较通道的控制寄存器被命名为 OC1CON1 和 OC1CON2 至 OC9CON1 和 OC9CON2。

所有的 9 个控制寄存器都具有相同的位定义。它们可以使用通用的寄存器定义表示，如寄存器 35-1 所列。OCxCON1 中的“x”表示输出比较通道的编号。

寄存器 35-1: OCxCON1: 输出比较 x 控制寄存器 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—
bit 15						bit 8	

R/W-0	U-0	U-0	R/W-0 HCS	R/W-0	R/W-0	R/W-0	R/W-0
ENFLT0	—	—	OCFLT0	TRIGMODE	OCM2 ⁽¹⁾	OCM1 ⁽¹⁾	OCM0 ⁽¹⁾
bit 7						bit 0	

图注:	HCS = 可由硬件清零 / 置 1 的位		
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 15-14 **未实现:** 读为 0

bit 13 **OCSIDL:** 在空闲模式下停止输出比较 x 的控制位

- 1 = 输出比较 x 在 CPU 空闲模式下停止工作
0 = 输出比较 x 在 CPU 空闲模式下继续工作

bit 12-10 **OCTSEL<2:0>:** 输出比较 x 时钟选择位

- 111 = 外设时钟 (Fcy)
110 = 保留
101 = 保留
100 = Timer1 时钟 (仅支持同步时钟)
011 = Timer5 时钟
010 = Timer4 时钟
001 = Timer3 时钟
000 = Timer2 时钟

bit 9-8 **未实现:** 读为 0

bit 7 **ENFLT0:** 故障 0 输入使能位

- 1 = 使能故障输入
0 = 禁止故障输入

bit 6-5 **未实现:** 读为 0

bit 4 **OCFLT0:** PWM 故障条件状态位

- 1 = 发生了 PWM 故障条件
0 = 未发生 PWM 故障条件

bit 3 **TRIGMODE:** 触发状态模式选择位

- 1 = 当 OCxRS = OCxTMR 时, TRIGSTAT (OCxCON2<6>) 被清零, 或用软件清零
0 = TRIGSTAT 只能用软件清零

注 1: 如果器件支持外设引脚选择, 那么 OCx 输出必须也配置给可用的 RPN 引脚。更多信息, 请参见具体器件的数据手册。

2: OCFA 引脚控制 OC1-OC4 通道; OCFB 引脚控制 OC5-OC9 通道。OCxR 和 OCxRS 仅在 PWM 模式下是双重缓冲的。

寄存器 35-1: OCxCON1: 输出比较 x 控制寄存器 1 (续)

bit 2-0	OCM<2:0> : 输出比较模式选择位 ⁽¹⁾
111	= 中心对齐的 PWM 模式: 当 OCxTMR = OCxR 时输出设置为高电平, 当 OCxTMR = OCxRS 时输出设置为低电平 ⁽²⁾
110	= 边沿对齐的 PWM 模式: 当 OCxTMR = 0 时输出设置为高电平, 当 OCxTMR = OCxR 时输出设置为低电平 ⁽²⁾
101	= 双比较连续脉冲模式: 初始化 OCx 引脚为低电平, 在 OCxTMR 与 OCxR 和 OCxRS 交替匹配时连续翻转 OCx 状态
100	= 双比较单次触发模式: 初始化 OCx 引脚为低电平, 在一个周期内 OCxTMR 分别与 OCxR 和 OCxRS 匹配时翻转 OCx 状态
011	= 单比较模式: 与 OCxR 匹配的比较事件使 OCx 引脚的电平连续翻转
010	= 单比较单次触发模式: 初始化 OCx 引脚为高电平, 与 OCxR 匹配的比较事件强制 OCx 引脚为低电平
001	= 单比较单次触发模式: 初始化 OCx 引脚为低电平, 与 OCxR 匹配的比较事件强制 OCx 引脚为高电平
000	= 禁止输出比较通道

- 注 1: 如果器件支持外设引脚选择, 那么 OCx 输出必须也配置给可用的 RPN 引脚。更多信息, 请参见具体器件的数据手册。
- 2: OCFA 引脚控制 OC1-OC4 通道; OCFB 引脚控制 OC5-OC9 通道。OCxR 和 OCxRS 仅在 PWM 模式下是双重缓冲的。

寄存器 35-2: OCxCON2: 输出比较 x 控制寄存器 2

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0
FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	OC32
bit 15							bit 8

R/W-0	R/W-0 HS	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7							bit 0

图注:	HS = 可由硬件置 1 的位		
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 15	FLTMD : 故障模式选择位
	1 = 故障模式将保持到故障源消除; 对应的 OCFLT0 位用软件清零并且新的 PWM 周期开始
	0 = 故障模式将保持到故障源消除, 并且新的 PWM 周期开始
bit 14	FLTOUT : 故障结果位
	1 = 发生故障时 PWM 输出驱动为高电平
	0 = 发生故障时 PWM 输出驱动为低电平
bit 13	FLTTRIEN : 故障输出状态选择位
	1 = 发生故障时 OCx 引脚为三态
	0 = 发生故障时 OCx 引脚的 I/O 状态由 FLTOUT 位定义
bit 12	OCINV : OCMP 反相位
	1 = OCx 输出反相
	0 = OCx 输出同相
bit 11-9	未实现 : 读为 0
bit 8	OC32 : 级联两个 OCx 模块使能位 (32 位操作)
	1 = 使能级联模块工作
	0 = 禁止级联模块工作

- 注 1: 绝不要通过选择该模式或其他相当的 SYNCSEL 设置将 OCx 模块用作它自己的触发源。
- 2: 只用这些输入作为触发源, 从不用作同步源。

寄存器 35-2: OCxCON2: 输出比较 x 控制寄存器 2 (续)

bit 7	OCTRIG: OCx 触发 / 同步选择位 1 = 用 SYNCSELx 位指定的源触发 OCx 0 = 用 SYNCSELx 位指定的源同步 OCx
bit 6	TRIGSTAT: 定时器触发状态位 1 = 定时器源已触发并正在运行 0 = 定时器源未触发, 保持清零
bit 5	OCTRI5: OCx 输出引脚方向选择位 1 = OCx 为三态 0 = 输出比较模块驱动 OCx 引脚
bit 4-0	SYNCSEL<4:0>: 触发 / 同步源选择位 11111 = 该 OCx 模块 ⁽¹⁾ 11110 = 输入捕捉 9 ⁽²⁾ 11101 = 输入捕捉 6 ⁽²⁾ 11100 = CTMU ⁽²⁾ 11011 = A/D ⁽²⁾ 11010 = 比较器 3 ⁽²⁾ 11001 = 比较器 2 ⁽²⁾ 11000 = 比较器 1 ⁽²⁾ 10111 = 输入捕捉 4 ⁽²⁾ 10110 = 输入捕捉 3 ⁽²⁾ 10101 = 输入捕捉 2 ⁽²⁾ 10100 = 输入捕捉 1 ⁽²⁾ 10011 = 输入捕捉 8 ⁽²⁾ 10010 = 输入捕捉 7 ⁽²⁾ 1000x = 保留 01111 = Timer5 01110 = Timer4 01101 = Timer3 01100 = Timer2 01011 = Timer1 01010 = 输入捕捉 5 ⁽²⁾ 01001 = 输出比较 9 01000 = 输出比较 8 00111 = 输出比较 7 00110 = 输出比较 6 00101 = 输出比较 5 00100 = 输出比较 4 00011 = 输出比较 3 00010 = 输出比较 2 00001 = 输出比较 1 00000 = 不与任何其他模块同步

- 注 1: 绝不要通过选择该模式或其他相当的 SYNCSEL 设置将 OCx 模块用作它自己的触发源。
 2: 只用这些输入作为触发源, 从不用作同步源。

PIC24F 系列参考手册

寄存器 35-3: OCxR: 比较寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OCRB15	OCRB14	OCRB13	OCRB12	OCRB11	OCRB10	OCRB9	OCRB8
bit 15							bit 8

R/W-0	R/W-0, HS	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OCRB7	OCRB6	OCRB5	OCRB4	OCRB3	OCRB2	OCRB1	OCRB0
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 15-0

OCRB<15:0>: 主比较寄存器值位

当 OCxR<2:0> = 0b110 时: 该寄存器用作边沿对齐的 PWM 模式下的占空比

当 OCxR<2:0> = 0b111、0b101 或 0b100 时: 该寄存器用于产生正边沿

当 OCxR<2:0> = 0b001、0b010 或 0b011 时: 该寄存器用于产生所有边沿

寄存器 35-4: OCxRS: 辅助比较寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OCRSB15	OCRSB14	OCRSB13	OCRSB12	OCRSB11	OCRSB10	OCRSB9	OCRSB8
bit 15							bit 8

R/W-0	R/W-0, HS	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OCRSB7	OCRSB6	OCRSB5	OCRSB4	OCRSB3	OCRSB2	OCRSB1	OCRSB0
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 15-0

OCRSB<15:0>: 辅助比较寄存器值位

这是周期寄存器:

如果 OCxCON2.SYNCSEL<4:0> = 0x1F

如果 OCxCON2.SYNCSEL<4:0> = N (其中, N 是备选值, 用于选择将该寄存器作为周期寄存器)

如果 OCxCON2.OCTRIG = 1

所有其他条件:

周期在该模块之外确定。当 OCM<2:0> = 0b111、0b101 或 0b100 时, 用于产生负边沿

35.3 工作模式

每路输出比较模块均包含以下工作模式：

- 单比较匹配模式
- 双比较匹配模式产生：
 - 单输出脉冲
 - 连续输出脉冲
- 带 / 不带故障保护的简单脉宽调制模式：
 - 边沿对齐
 - 中心对齐
- 级联模式（32 位操作）

在理解这些模式之前，需要先理解同步 / 触发。在同步操作中，当通过 SYNCSEL<4:0>（OCxCON2<4:0>）位选择的源发送同步信号时，内部定时器会复位（复位为零）。在触发模式下，内部定时器保持在复位状态，直到选定的触发源发送同步信号为止。

同步或触发模式通过 OCTRIG（OCxCON2<7>）位选择，同步 / 触发源可以通过 SYNCSEL<4:0> 位选择，如第 35.2 节“输出比较寄存器”中所示。

- 注**
- 1: SYNCSEL<4:0> = 0b00000 时，定时器处于自由运行模式，不进行同步。
 - 2: SYNCSEL<4:0> = 0b11111 时，当定时器达到 OCxRS 值时定时器会复位，使 OCx 模块使用自己的同步信号。
 - 3: 当定时器与 OCxRS 匹配时，OCx 模块会发送同步 / 触发信号。

关于同步 / 触发模式的更多信息，请参见第 35.3.3.7 节“同步操作”。

35.3.1 单比较匹配模式

当 OCxCON1 寄存器的控制位 OCM2:OCM0 = 0b001、0b010 或 0b011 时，选定的输出比较通道配置为：

- 如果 OCM = 0b001: OCx 引脚初始化设置为低电平；后续与 OCxR 匹配的比较事件会将引脚设置为高电平
- 如果 OCM = 0b010: OCx 引脚初始化设置为高电平；后续与 OCxR 匹配的比较事件会将引脚设置为低电平
- 如果 OCM = 0b011: OCx 引脚初始化设置为低电平；后续与 OCxR 匹配的比较事件会翻转引脚电平

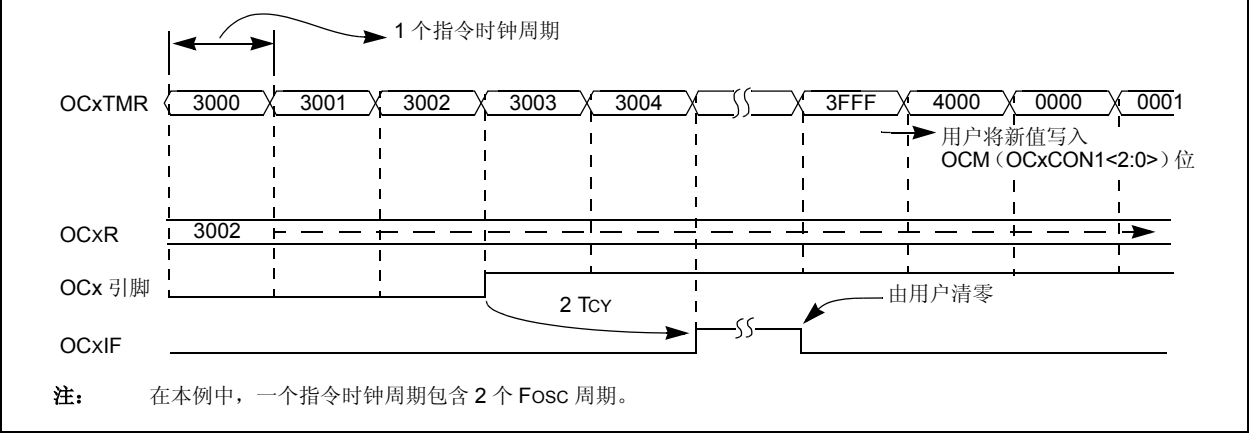
在单比较模式下，OCxR 寄存器用于产生比较事件。该寄存器中装入一个值，与模块定时器寄存器进行比较。发生每个比较事件时，如果 OCx 引脚电平改变，则会产生中断请求。

35.3.1.1 单比较模式输出驱动为高电平

要将模块配置为这种模式，需设置控制位 $OCM<2:0>$ ($OCxCON1<2:0>$) = 0b001。一旦使能该比较模式，输出引脚 OCx 初始化驱动为低电平，并保持低电平，直到定时器与 $OCxR$ 寄存器发生匹配为止。图 35-2 给出了以下关键的时序事件：

- 在定时器和 $OCxR$ 寄存器之间发生比较匹配后经过一个指令时钟， OCx 引脚被驱动为高电平。 OCx 引脚保持高电平，直到模式发生改变或模块关闭为止。
- 定时器计数递增，直到它计满返回或发生同步事件，然后在下一个指令时钟复位（复位为 0x0000）。
- 在 OCx 引脚被驱动为高电平后再过 2 个指令时钟，相应通道的中断标志位 $OCxIF$ 被置 1。

图 35-2: 单比较模式——在比较匹配事件发生时将 OCx 设置为高电平

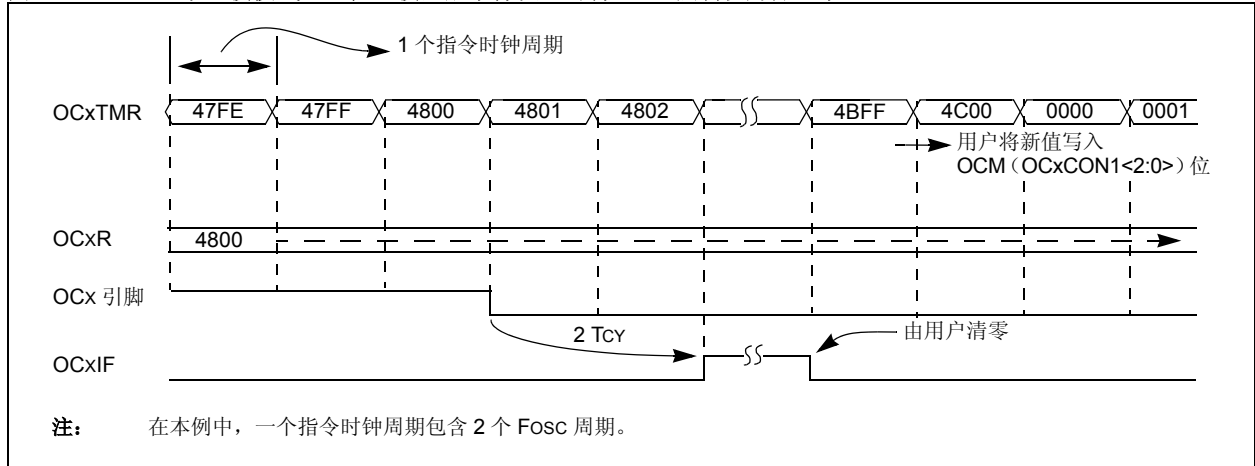


35.3.1.2 单比较模式输出驱动为低电平

要将输出比较模块配置为这种模式，需设置控制位 $OCM<2:0>$ = 0b010。一旦使能该比较模式，输出引脚和 OCx 初始化驱动为高电平，并保持高电平，直到定时器与 $OCxR$ 寄存器发生匹配为止。图 35-3 给出了关键的时序事件。

- 在定时器和 $OCxR$ 寄存器之间发生比较匹配事件后经过一个指令时钟， OCx 引脚被驱动为低电平。 OCx 引脚保持低电平，直到模式发生改变或模块关闭为止。
- 定时器计数递增，直到它计满返回或发生同步事件，然后在下一个指令时钟复位为 0x0000。
- 在 OCx 引脚被驱动为低电平后再过 2 个指令时钟，相应通道的中断标志位 $OCxIF$ 被置 1。

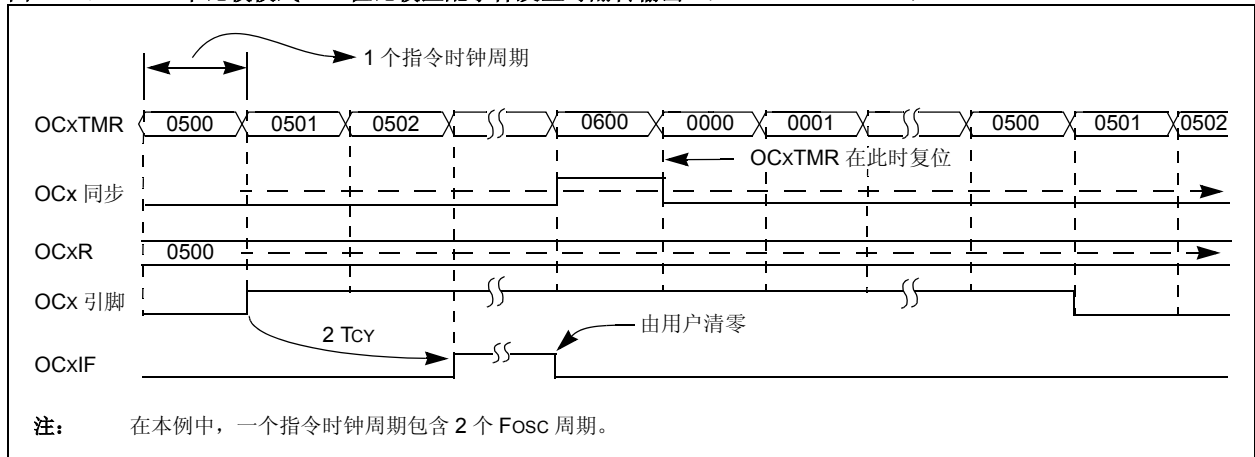
图 35-3: 单比较模式——在比较匹配事件发生时将 OCx 强制设为低电平



35.3.1.3 单比较模式翻转输出

要将输出比较模块配置为这种模式，需设置控制位 $OCM<2:0> = 0b011$ 。一旦使能该比较模式，输出引脚和 OCx 在每次定时器与 OCxR 寄存器发生匹配时翻转电平。图 35-4 给出了关键的时序事件。

- 在定时器和 OCxR 寄存器之间发生比较匹配后经过一个指令时钟，OCx 引脚电平翻转。OCx 引脚将保持此新状态直到发生下一次翻转事件、模式改变或该模块被关闭。
- 定时器计数递增，直到它计满返回或发生同步事件，然后在下一个指令时钟复位为 0x0000。
- 在 OCx 引脚电平翻转之后再过 2 个指令时钟，相应通道的中断标志位 OCxIF 被置 1。
- 器件复位时，内部 OCx 引脚输出逻辑被设置为逻辑 0。但是，用于翻转模式的 OCx 引脚的状态可由用户软件设置。

图 35-4: 单比较模式——在比较匹配事件发生时翻转输出 ($OCxTMR > OCxR$)

例 35-1: 单比较模式翻转输出

```

OC1CON1 = 0;           /* It is a good practice to clear off the control bits initially */
OC1CON2 = 0;
OC1CON1bits.OCTSEL = 0x07; /* This selects the peripheral clock as the clock input to the OC
                             module */
OC1R = 1000;           /* This is just a typical number, user must calculate based on the
                             waveform requirements and the system clock */
OC1CON1bits.OCM = 3;   /* This selects the toggle mode */

```

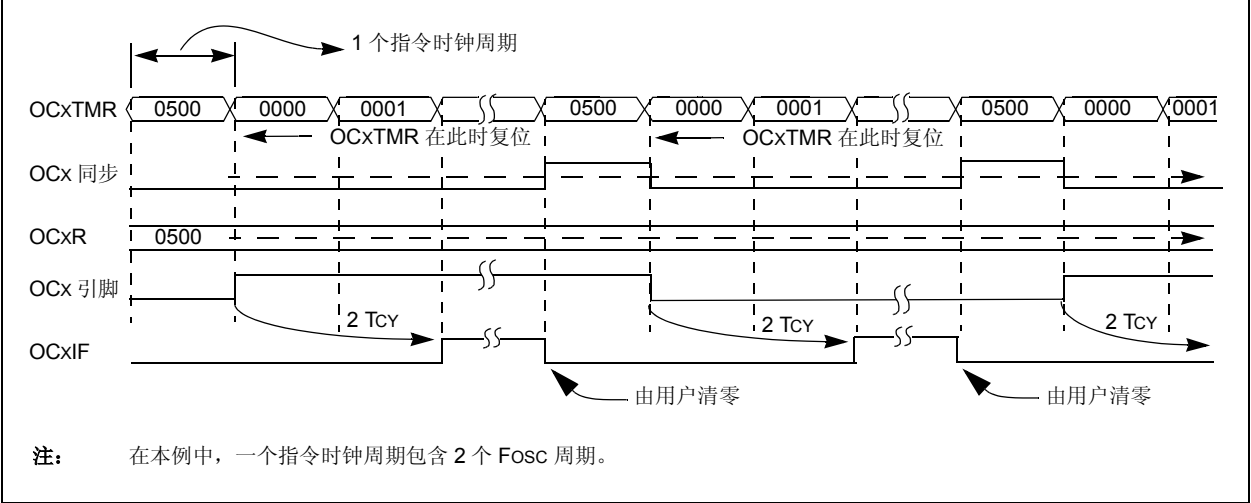
35.3.1.4 单比较模式的特殊情况

考虑以下几种特殊情况：

表 35-1：单比较模式的特殊情况

特殊情况	操作	输出
当 OCxR > 定时器周期（由同步源决定）时	不发生比较事件，比较输出保持初始状态。	输出电平不改变
当 OCxR = 定时器周期（由同步源决定）时	比较输出正常工作。将此方式与翻转模式结合使用可以产生固定频率的方波，如图 35-5 所示。	输出电平翻转
当模块使能为单比较模式时，OCxR = 0x0000，并且定时器保持为复位状态，同步源有效	比较输出保持在初始状态。	输出电平不改变
如果在发生比较事件之后，OCxR 寄存器被清零，并且同步源变为有效	输出保持在新状态。	输出电平不会发生第二次改变

图 35-5：单比较模式——在比较匹配事件（OCxTMR = OCxR）发生时翻转输出



35.3.2 双比较匹配模式

当控制位 OCM<2:0> = 0b100 或 0b101 时，选定的输出比较通道配置为以下两种双比较匹配模式中的一种：

- 单输出脉冲模式
- 连续输出脉冲模式

在双比较模式下，该模块在处理比较匹配事件时使用 OCxR 和 OCxRS 这两个寄存器。将 OCxR 寄存器的值与递增定时器计数器 OCxTMR 的值作比较，并且在比较匹配事件发生时，在 OCx 引脚上产生脉冲的上升（前）沿。然后 OCxRS 寄存器的值与同一个递增定时器计数器 OCxTMR 的值作比较，并且在比较匹配事件发生时，在 OCx 引脚上产生脉冲的下降（后）沿。

35.3.2.1 双比较单脉冲模式

当控制位 OCM<2:0> = 0b100 时，选定的输出比较通道配置为 OCx 引脚初始化为低电平，并产生单个输出脉冲。请参见图 35-6 和图 35-7。

1. 一旦使能双比较单脉冲模式，OCx 引脚将驱动为低电平。
2. 当定时器与 OCxR（比较寄存器）第一次发生比较匹配时，其引脚（OCx）将驱动为高电平。
3. 当递增定时器计数值与比较寄存器 OCxRS 发生匹配时，在 OCx 引脚上驱动脉冲的第二个边沿（即后沿，从高至低）。在第二次比较匹配事件发生时，OCxIF 中断标志位将置 1。

注 1：虽然模式位在脉冲下降沿之后不会改变，但如果另一次写操作向相同的控制位写入相同的值，那么会产生新的单输出脉冲序列。

2：OCxRS 必须至少比 OCxR 大 2。

图 35-6： 双比较模式单输出脉冲

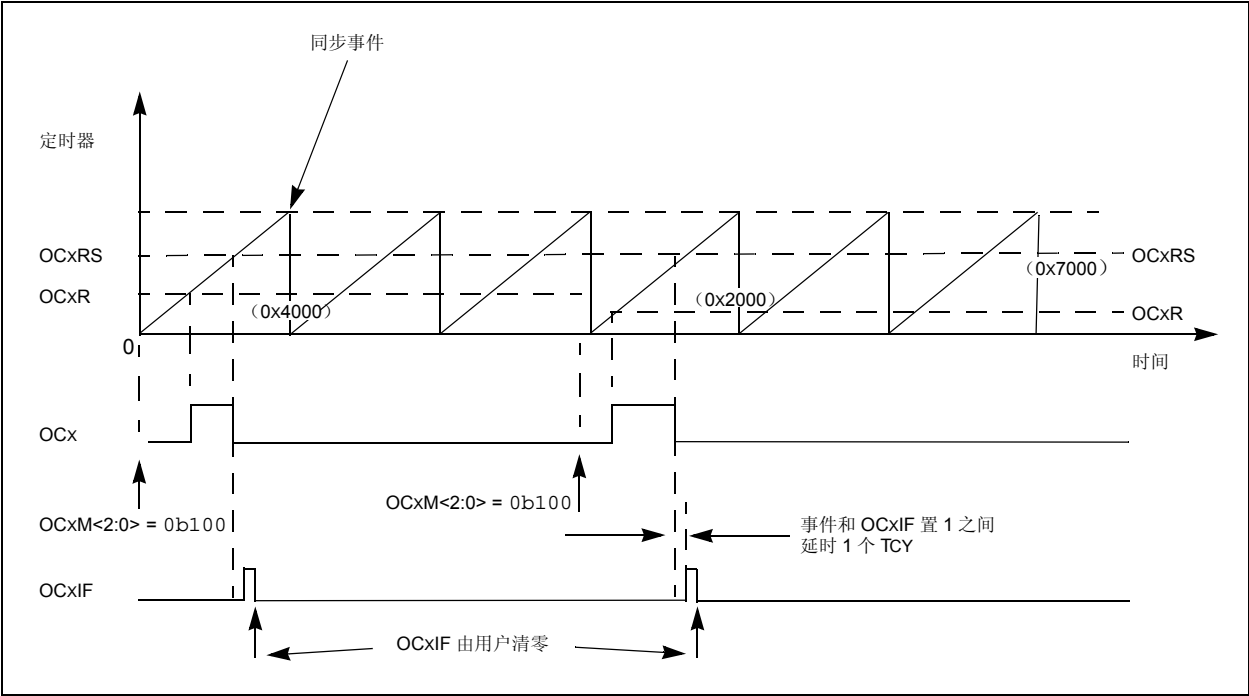
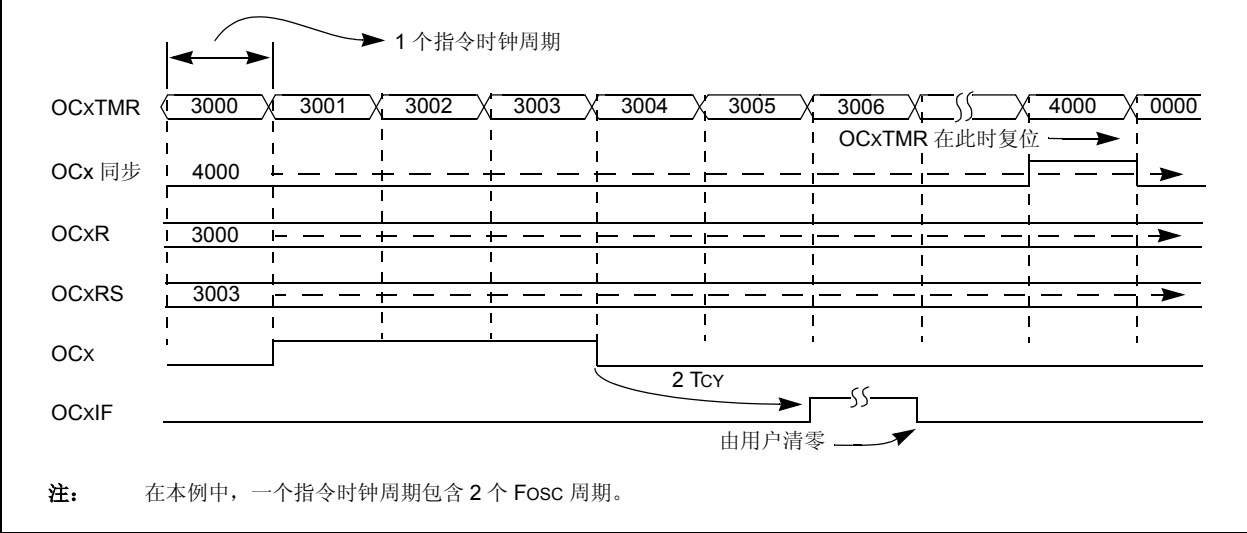


图 35-7: 双比较模式



35.3.2.2 设置产生单输出脉冲

要将模块配置为产生单输出脉冲, 请执行以下步骤:

1. 确定指令周期 Tcy。
2. 根据 Tcy 计算所需的脉冲宽度值。
3. 计算从定时器起始值 0x0000 到脉冲开始的时间。
4. 将脉冲宽度起始和停止时间写入 OCxR 和 OCxRS 比较寄存器。
5. 选择 SYNCSEL<4:0>, 使得在定时器等于或大于 OCxRS 值之后, 同步变为有效。
6. 设置 OCM<2:0> = 0b100。
7. 发起另一个写操作来设置 OCM<2:0> = 0b100, 以使用相同的参数启动另一个单脉冲。
8. 通过写 OCM<2:0> = 0b000 来禁止 OCx, 更改参数, 然后通过写 OCM<2:0> = 0b100 来使能 OCx, 以使用不同的参数启动另一个单脉冲。

注 1: 请参见表 35-2, 其中列出了单输出脉冲宽度计算的几个简单示例。
2: 请参见表 35-3, 其中列出了产生单输出脉冲的双比较匹配模式的几个简单示例。

表 35-2: 双比较模式——单输出脉冲宽度计算示例

指令周期 (Tcy)	期望的导通时间		脉冲起始时间 从定时器 = 0x0000 开始		脉冲结束时间 (OCxRS) 寄存器
	时间	十六进制值	时间	十六进制值 (OCxR)	
30 ns	1 ms	0x0021	10 ms	0x014D	0x016E
30 ns	2 ms	0x0042	10 ms	0x014D	0x018F
50 ns	3 ms	0x003C	10 ms	0x00C8	0x0104
100 ns	5 ms	0x0032	50 ms	0x0064	0x0096
300 ns	10 ms	0x0021	100 ms	0x014D	0x018F
500 ns	20 ms	0x0028	500 ms	0x03E8	0x0410
500 ns	30 ms	0x003C	2 ms	0x0FA0	0x0FDC

公式 35-1:

OCxR 值 = 期望时间 / 指令周期时间 (Tcy)

例 35-2: 双比较模式——单输出脉冲宽度

```
OC1CON1 = 0;          /* It is a good practice to clear off the control bits initially */
OC1CON2 = 0;
OC1CON1bits.OCTSEL = 0x07; /* This selects the peripheral clock as the clock input to the OC
                             module */
OC1R = 1000;           /* This is just a typical number, user must calculate based on the
                         waveform requirements and the system clock */
OC1RS = 2000;
OC1CON1bits.OCM = 4;   /* This selects the Single Output Pulse mode */
```

表 35-3: 双比较匹配模式产生单输出脉冲的特殊情况

特殊情况	操作	输出
当定时器值等于 OCxRS 时发生同步	定时器在下一个周期复位为零，但脉冲不受影响。	脉冲
在定时器值达到 OCxR 之前发生同步	定时器在任意输出翻转之前复位为零。	保持为低电平
在定时器值达到 OCxRS 之前，但达到 OCxR 之后发生同步	仅产生单次翻转（低电平到高电平）（见图 35-8）。	低电平到高电平
OCxR = OCxRS = 0x0000，并发生同步	输出初始化为低电平，并且不发生改变。不产生中断。	保持为低电平
OCxRS < OCxR	定时器计数递增至第一个比较值（TMRx = OCxR），并产生第一个（上升）边沿。然后定时器继续计数，最终在发生同步或计满返回时复位。然后定时器从 0x0000 重新开始计数，并递增至第二个比较值（TMRx = OCxRS），并产生信号的第二个（下降）边沿。输出脉冲的下降沿产生中断条件。	脉冲
OCxR = OCxRS	定时器计数递增至第一个比较值（定时器 = OCxR），并产生第一个（上升）边沿。定时器继续计数，最终在发生同步或从 0xFFFF 计满返回时复位。然后定时器从 0x0000 重新开始计数，并递增至第二个比较值（TMRx = OCxRS），并产生信号的第二个（下降）边沿。输出脉冲的下降沿产生中断条件。	脉冲
OCxR = 0x0000 且 OCxRS > OCxR	定时器计数在第一个周期递增，直到发生同步或计满返回；输出比较引脚保持为低电平。在定时器寄存器复位为零之后，输出比较引脚变为高电平。在下次定时器与寄存器 OCxRS 发生匹配时，输出比较引脚变为低电平并保持低电平。输出脉冲的下降沿产生中断条件（见图 35-9）。	脉冲（第一个周期除外）

图 35-8: 双比较模式——单输出脉冲（在定时器达到 OCxRS 之前发生同步）

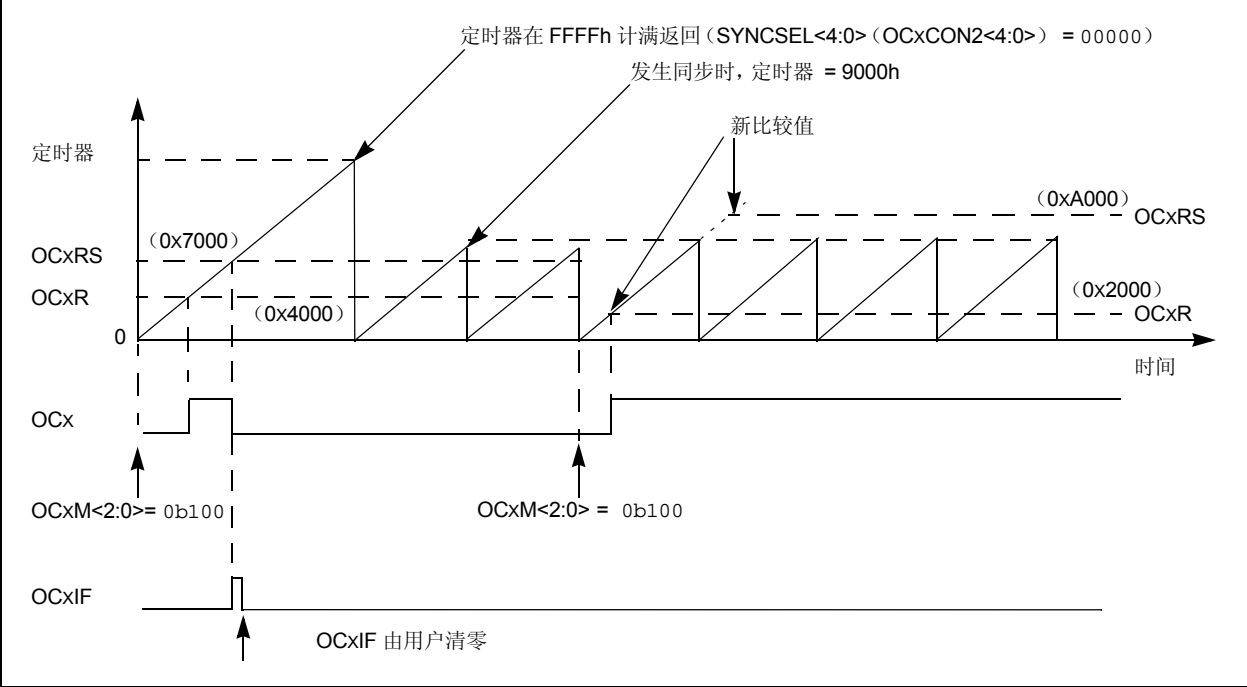
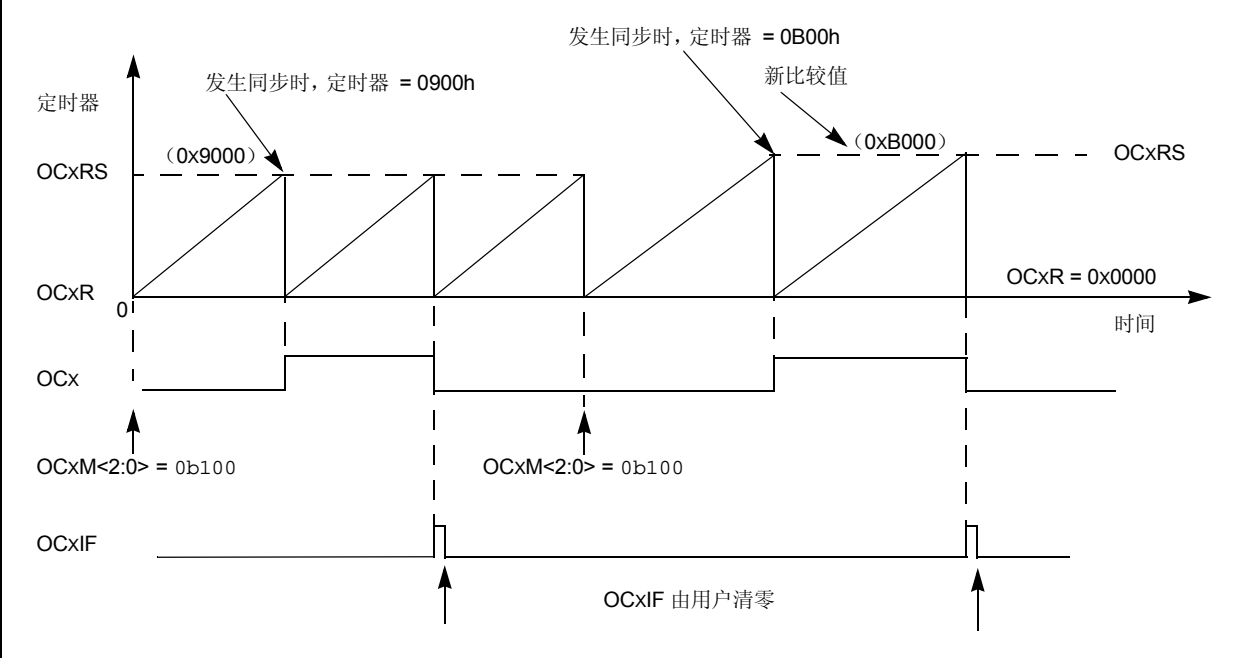


图 35-9: 双比较模式——单输出脉冲（OCxR = 0x0000，OCxRS > OCxR）



35.3.2.3 双比较连续脉冲模式

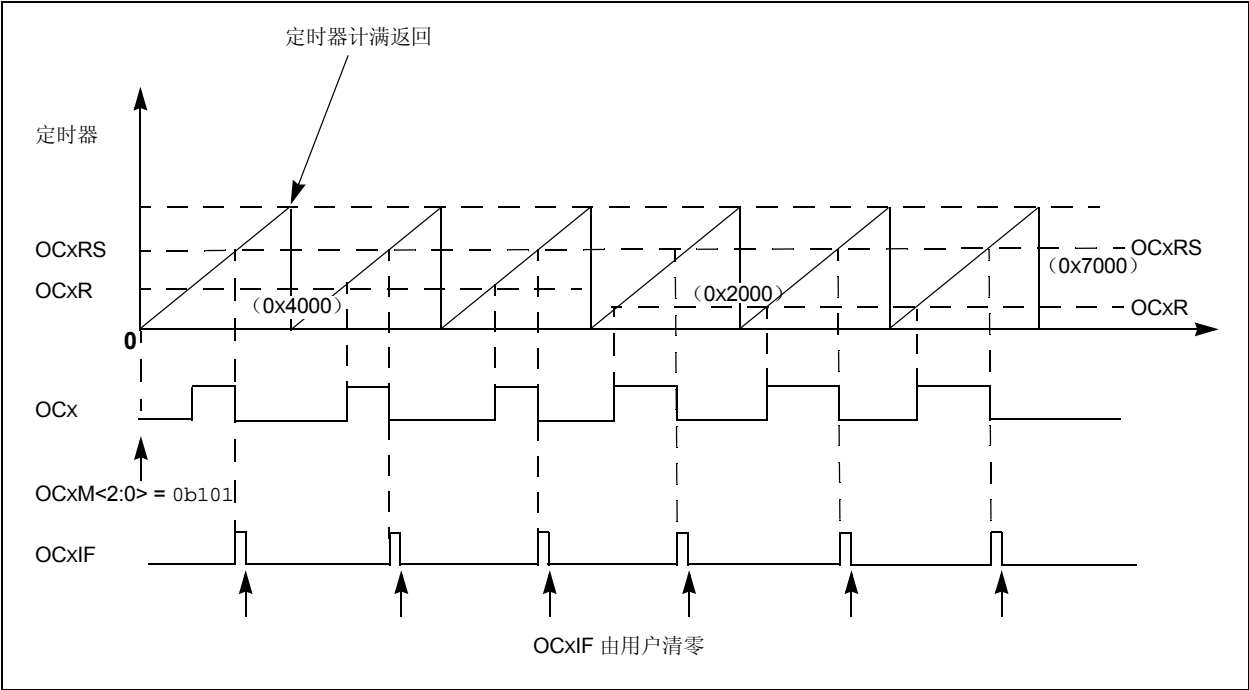
当控制位 $OCM<2:0> = 0b101$ 时，选定的输出比较通道配置为使 OCx 引脚初始化为低电平，并产生连续的输出脉冲。图 35-10 给出了双比较连续输出脉冲模式的图示。

- 一旦使能双比较连续输出脉冲模式，引脚状态将驱动为低电平。
- 当定时器与比较寄存器 $OCxR$ 第一次发生比较匹配时， OCx 引脚将驱动为高电平。
- 当递增定时器计数值与比较寄存器 $OCxRS$ 发生匹配时，在 OCx 引脚上驱动脉冲的第二个边沿（即后沿，从高至低）。在第二次比较匹配事件发生时， $OCxIF$ 中断标志位将置 1。

注 1: 不同于单输出脉冲模式，输出脉冲将无限期地持续，直到模式被用户固件或复位终止为止。每个输出脉冲的下降沿会将中断标志置 1。

2: 产生占空比为 50% 的脉冲的一种方法是设置 $OCxR = OCxRS$ （该值必须处于周期寄存器的取值范围内，周期的值由同步源决定）。

图 35-10： 双比较模式——连续输出脉冲



35.3.2.4 设置产生连续输出脉冲

要将该模块配置为产生连续的输出脉冲流，请执行以下步骤：

- 1. 确定指令周期 T_{CY} 。
- 2. 计算从定时器起始值 0x0000 到脉冲宽度开始的时间。
- 3. 计算从定时器起始值 0x0000 到脉冲宽度停止的时间。
- 4. 将脉冲宽度起始和停止时间分别写入 OCxR 和 OCxRS 比较寄存器。同步信号应在 OCxRS = 定时器时或之后发生。
- 5. 设置 OCM<2:0> = 0b101；必须使能定时器。

例 35-3：产生连续输出脉冲

```
OC1CON1 = 0;
OC1CON2 = 0; /* It is a good practice to clear off the control bits initially */
OC1CON1bits.OCTSEL = 0x07; /* This selects the peripheral clock as the clock input to the OC
                             module */
OC1R = 1000; /* This is just a typical number, user must calculate based on the
              waveform requirements and the system clock */
OC1RS = 2000;
T1CON = 0;
PR1 = 3000; /* Determines the period */
OC1CON2bits.SYNCSEL = 0x0B; /* TMR1 is the sync source */
OC1CON1bits.OCM = 5; /* This selects the Continuous Pulse mode*/
T1CONbits.TON = 1; /* OC1TMR does not run until the sync source is switched on */
```

表 35-4：双比较模式——连续输出脉冲宽度计算示例

指令周期 (T_{CY})	期望的脉冲宽度		脉冲起始时间 从定时器 = 0x0000 开始		周期寄存器 内容	脉冲结束时间 (OCxRS) 寄存器 (对于 50% 占空比)
	时间	十六进制值	时间	十六进制值 (OCxR)		
30 ns	1 ms	0x0021	10 ms	0x014D	0x016F	0x015D
30 ns	2 ms	0x0042	10 ms	0x014D	0x0190	0x016F
50 ns	3 ms	0x003C	10 ms	0x00C8	0x0105	0x00E6
100 ns	5 ms	0x0032	50 ms	0x0064	0x0097	0x007D
300 ns	10 ms	0x0021	100 ms	0x014D	0x0190	0x015D
500 ns	20 ms	0x0028	500 μ s	0x03E8	0x057A	0x03FC
500 ns	30 ms	0x003C	2 ms	0x0FA0	0x0FC9	0x0FBE

公式 35-2：

十六进制值 = 期望时间 / 指令周期时间 (T_{CY})

注：在本例中，使用与 OCx 具有相同时钟的定时器模块作为同步源。

表 35-5: 双比较匹配模式产生连续输出脉冲的特殊情况

特殊情况	操作	输出
当定时器值等于 OCxRS 时发生同步	定时器在下一个周期复位为零，但脉冲不受影响（见图 35-11）。	脉冲
在定时器值达到 OCxR 之前发生同步	定时器在任意输出翻转之前复位为零。	保持为低电平
在定时器值达到 OCxRS 之前，但达到 OCxR 之后发生同步	仅产生单次翻转（低电平到高电平）（见图 35-12）。	低电平到高电平
OCxR = OCxRS = 0x0000，并发生同步	输出初始化为低电平，并且不发生改变。不产生中断。	保持为低电平
OCxRS < OCxR	定时器计数递增至第一个比较值（TMRx = OCxR），并产生第一个（上升）边沿。然后定时器继续计数，最终在发生同步或计满返回时复位。然后定时器从 0x0000 重新开始计数，并递增至第二个比较值（TMRx = OCxRS），并产生信号的第二个（下降）边沿。输出脉冲的下降沿产生中断条件。序列一直重复，直到模块被禁止。	脉冲
OCxR = OCxRS	定时器计数递增至第一个比较值（定时器 = OCxR），并产生第一个（上升）边沿。定时器继续计数，最终在发生同步或从 FFFFh 计满返回时复位。然后定时器从 0x0000 重新开始计数，并递增至第二个比较值（TMRx = OCxRS），并产生信号的第二个（下降）边沿。输出脉冲的下降沿产生中断条件。序列一直重复，直到模块被禁止。	脉冲
OCxR = 0x0000 且 OCxRS > OCxR	定时器计数在第一个周期递增，直到发生同步或计满返回；输出比较引脚保持为低电平。在定时器寄存器复位为零之后，输出比较引脚变为高电平。在下次定时器与寄存器 OCxRS 发生匹配时，输出比较引脚变为低电平并保持低电平。输出脉冲的下降沿产生中断条件（见图 35-13）。序列一直重复，直到模块被禁止。	脉冲（第一个周期除外）

图 35-11: 双比较模式——连续输出脉冲（在定时器 = OCxRS 时发生同步）

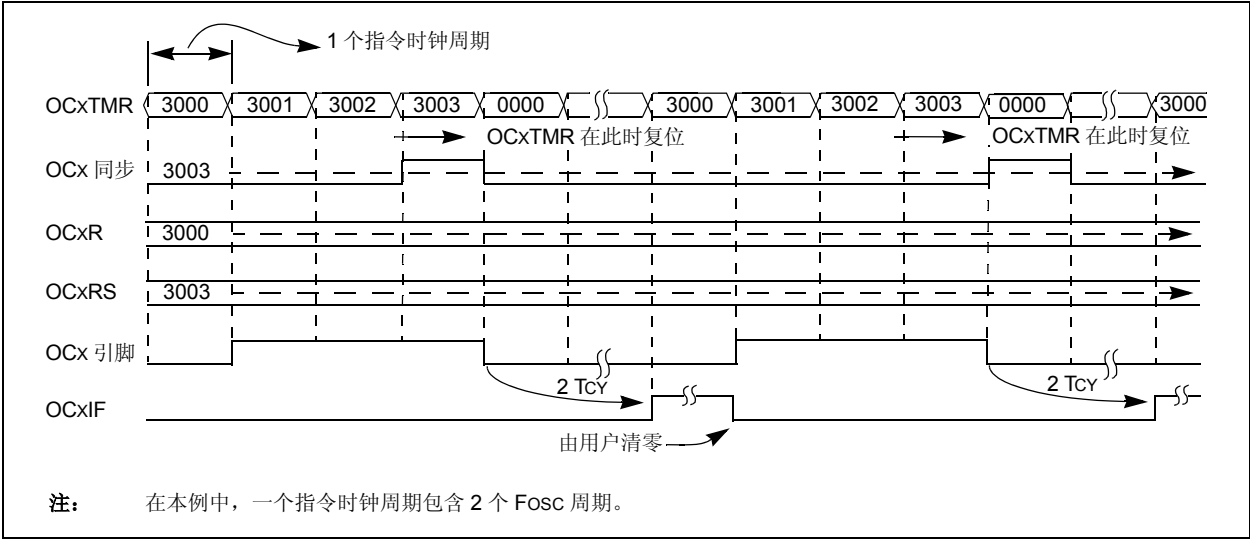


图 35-12: 双比较模式——连续输出脉冲（在定时器达到 OCxRS 之前发生同步）

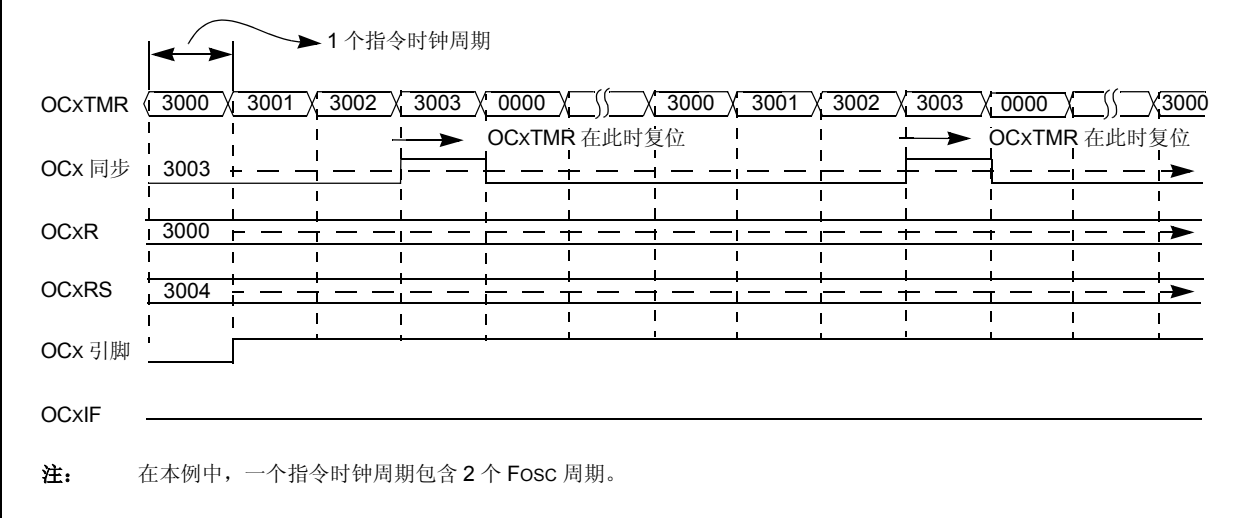
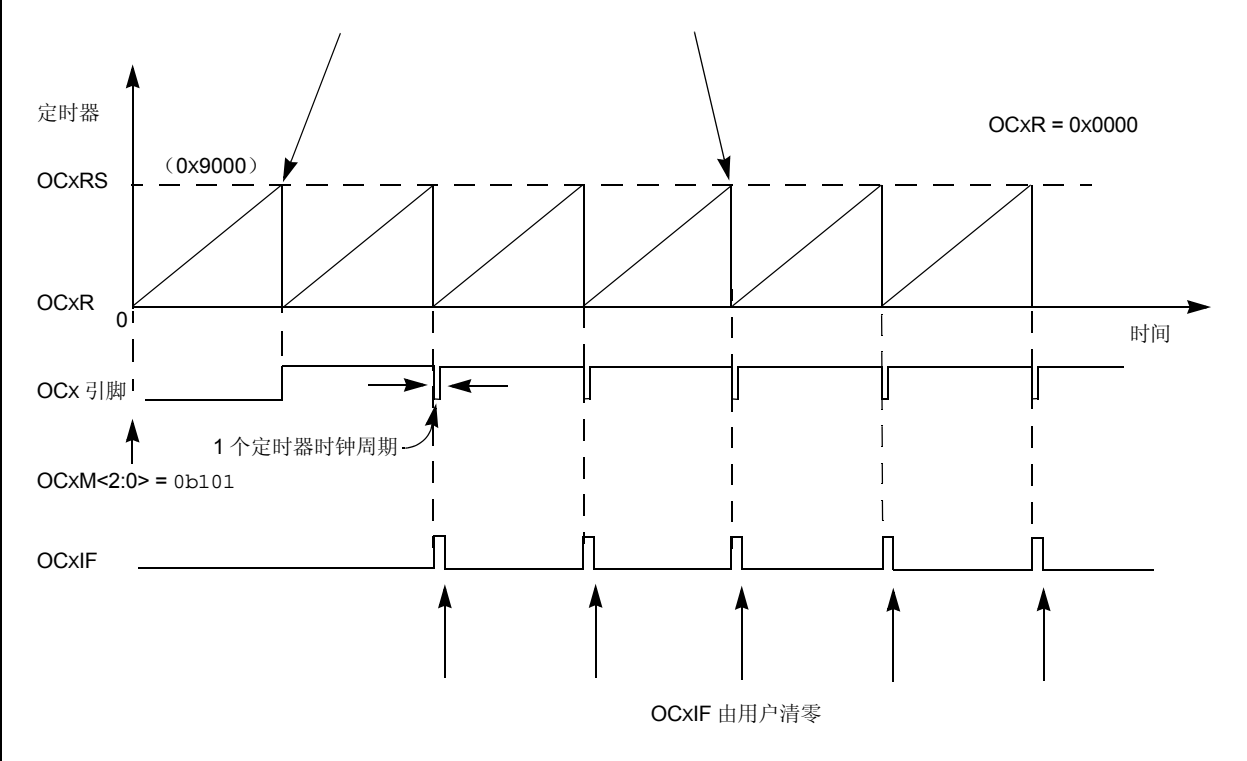


图 35-13: 双比较模式——连续输出脉冲（OCxR = 0x0000（SYNCSEL<4:0> = 0x1F））



35.3.3 脉宽调制模式

当控制位 OCxCON1<OCM2:OCM0> = 0b110 或 0b111 时，即选择了 PWM 模式。寄存器 OCxR 和 OCxRS 在这些模式下是双重缓冲的。这意味着，这些寄存器中的变化只有在定时器从 0xFFFF 计满返回或发生同步事件之后才会得到反映。因此，在操作期间对这些寄存器的任何更改只会在下一个脉冲生效。此外，在这些模式下，还支持故障输入（在下面的章节中说明）。

35.3.3.1 边沿对齐的 PWM 模式

当控制位 OCM<2:0> = 0b110 时，即选择了边沿对齐的 PWM 工作模式。OCxR 中包含当前占空比，而 SYNCSEL 位决定周期。通过设置 SYNCSEL<4:0> = 0x1F，可以配置为由 OCxRS 决定周期。

注： 这是一些应用的移植问题。在旧版本的模块中，OCxRS 用作 OCxR 的双重缓冲器。在当前版本中，两个寄存器都是双重缓冲的。

图 35-14 和图 35-15 给出了 PWM 模式下操作的图示。

边沿对齐的 PWM 模式的工作原理：

- 当发生同步时，在下一个递增周期会发生以下 4 个事件：
 - 定时器复位为零并重新开始计数
 - OCx 引脚被设置为高电平（例外：如果 OCxRS = 0b0000，则 OCx 引脚将不会被置 1）
 - OCxR 和 OCxRS 的缓冲寄存器从 OCxR 和 OCxRS 更新内容
 - 中断标志位 OCxIF 被置 1
- 当定时器与 OCxR 匹配时，引脚将设置为低电平。该匹配不会产生中断。

图 35-14: PWM 输出时序

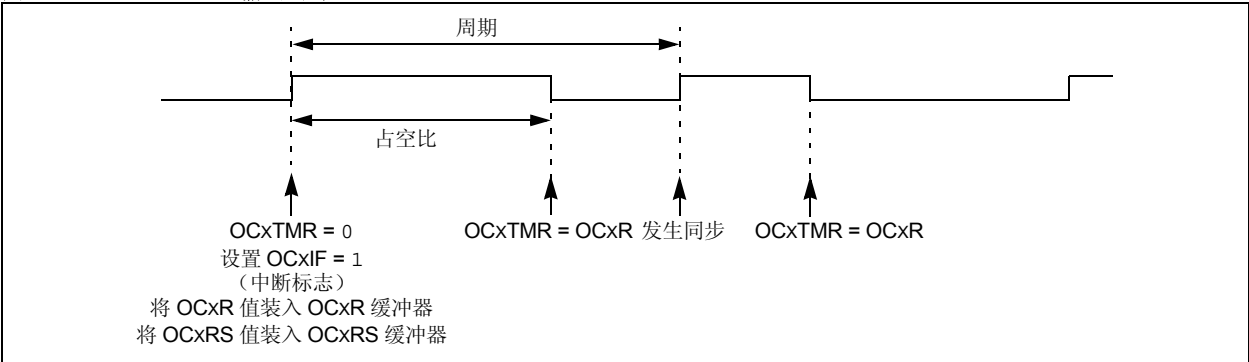
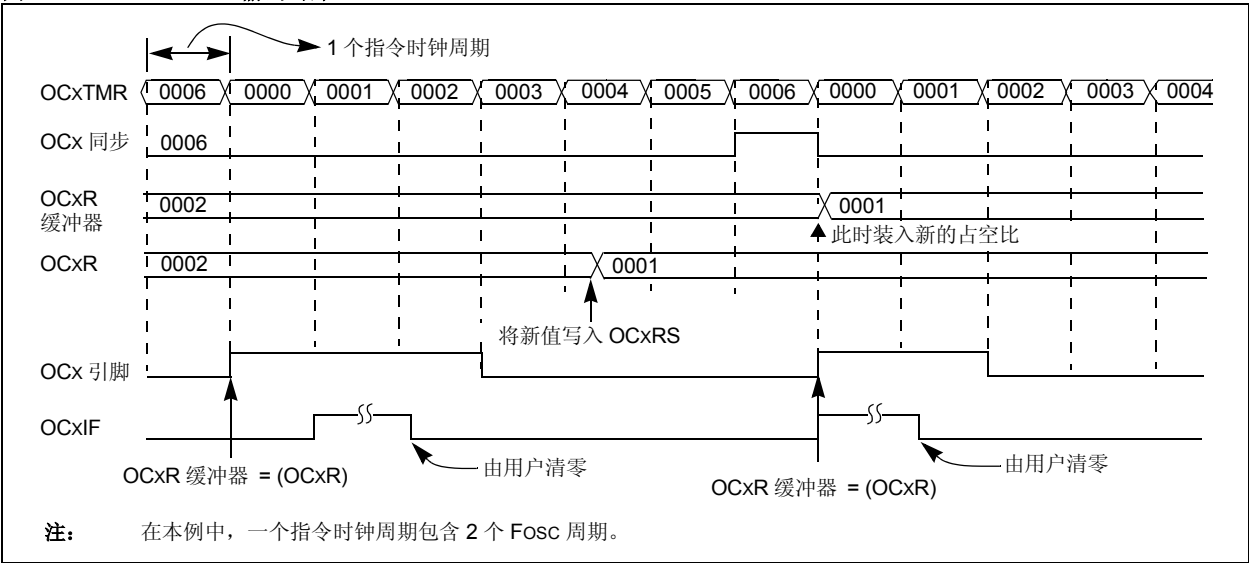


图 35-15: PWM 输出时序



35.3.3.2 边沿对齐的 PWM 模式的初始化

- 1. 一旦通过设置 OCM<2:0> = 0b110 使能 PWM 模式，如果 OCxR = 0x0000，OCx 引脚将驱动为低电平。如果 OCxR 不等于零，OCx 引脚将设置为高电平（见图 35-16 和图 35-17）。
- 2. 如果 OCxR 不等于零并且引脚状态设置为高电平，则在 OCxR 和定时器之间第一次匹配时清零 OCx 引脚。OCx 引脚将保持为低电平，直到在同步之间发生有效的比较，或计满返回为止（见图 35-17）。

图 35-16: 边沿对齐的 PWM 模式，OCxR = 0——在模块初始化时，OCxR = 0x0000，OCxRS = 0x5000

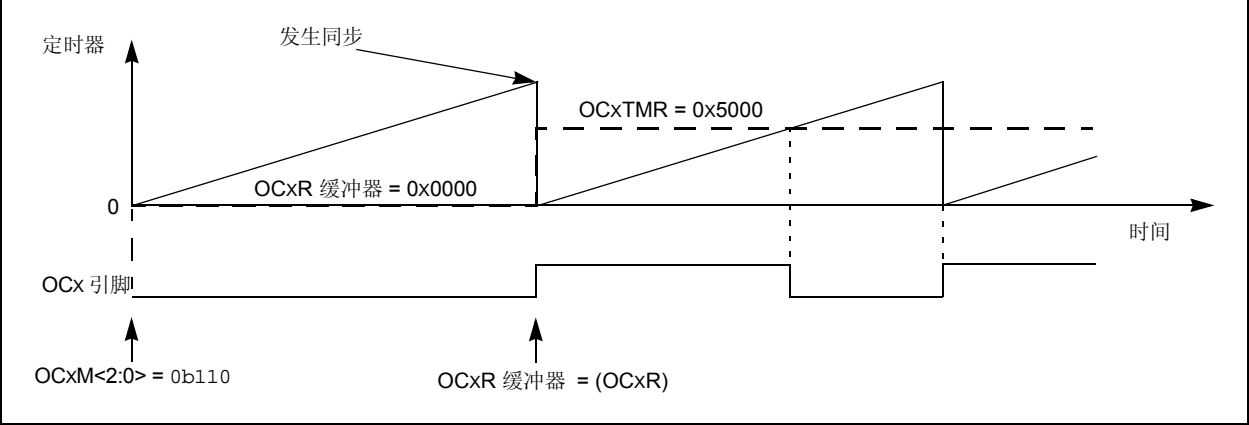
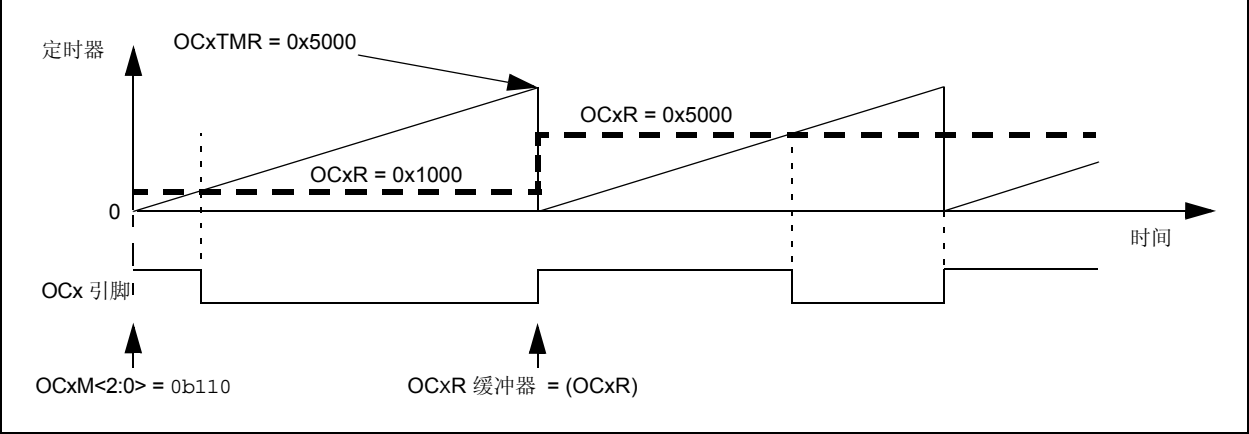


图 35-17: 边沿对齐的 PWM 模式，OCxR > 0——在模块初始化时，OCxR = 0x1000，OCxRS = 0x5000



35.3.3.3 PWM 操作的用户设置

当将输出比较模块配置为 PWM 操作时，需要执行以下步骤：

- 1. 确定指令周期 TcY。
- 2. 根据 TcY 计算期望的脉冲导通时间值，并将其写入 OCxR。
- 3. 根据 TcY 计算周期值，并将其写入 OCxRS。
- 4. 将 0x1F 写入 SYNCSEL<4:0>，以选择自同步。
- 5. 设置所需的时钟源。
- 6. 设置 OCxCON1 的 OCM2:OCM0 = 0b110 以选择边沿对齐的 PWM 模式。

例 35-4: PWM 模式

```
OC1CON1 = 0; /* It is a good practice to clear off the control bits initially */
OC1CON2 = 0;
OC1CON1bits.OSTSEL = 0x07; /* This selects the peripheral clock as the clock input to the OC
                             module */
OC1R = 1000; /* This is just a typical number, user must calculate based on the
              waveform requirements and the system clock */
OC1RS = 2000; /* Determines the Period */
OC1CON1bits.OCM = 6; /* This selects the Edge Aligned PWM mode*/
```

35.3.3.4 PWM 模式的特殊比较条件

表 35-6 列出了 PWM 模式的特殊比较条件。

表 35-6: 特殊比较模式条件

特殊条件	操作	输出
OCxR = 0	OCx 引脚将被设置为低电平（见图 35-18）。	低电平
OCxR > OCxRS	OCx 引脚将被设置为高电平（见图 35-19）。	高电平
OCxR = OCxTMR，并发生同步	OCx 引脚将保持为高电平（见图 35-20）。	高电平

图 35-18: PWM 输出时序（0% 占空比，OCxR = 0x0000）

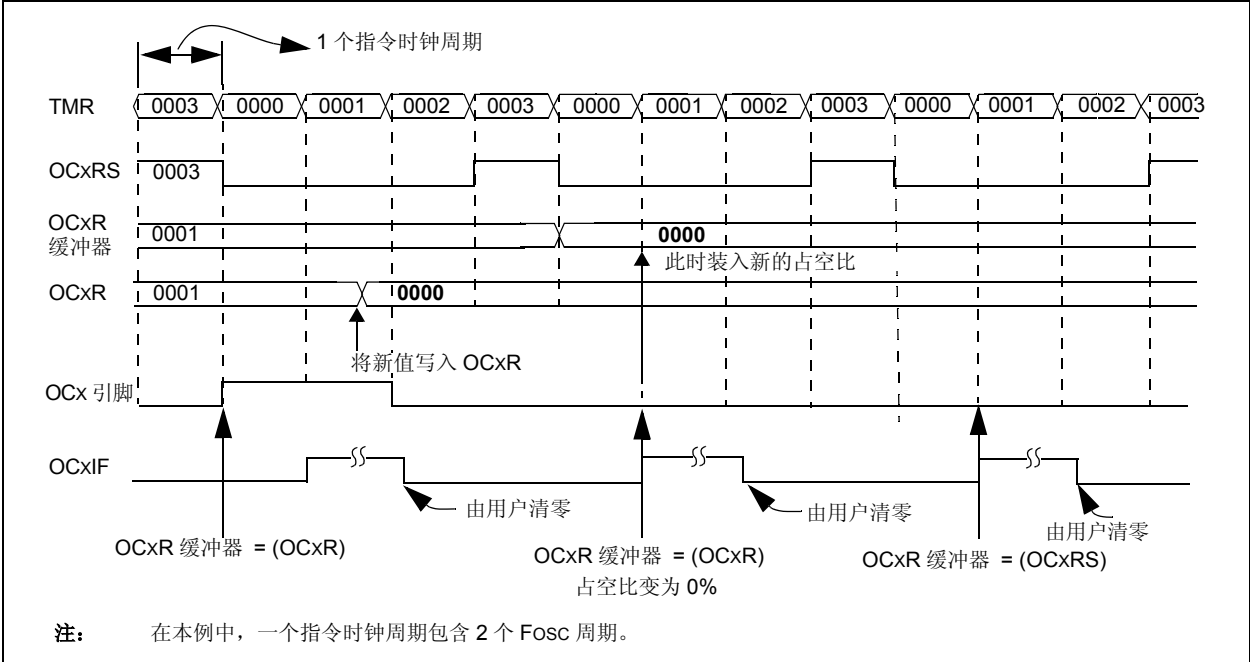


图 35-19: PWM 输出时序 (100% 占空比, OCxR > OCxRS (SYNCSEL<4:0> = 0x1F))

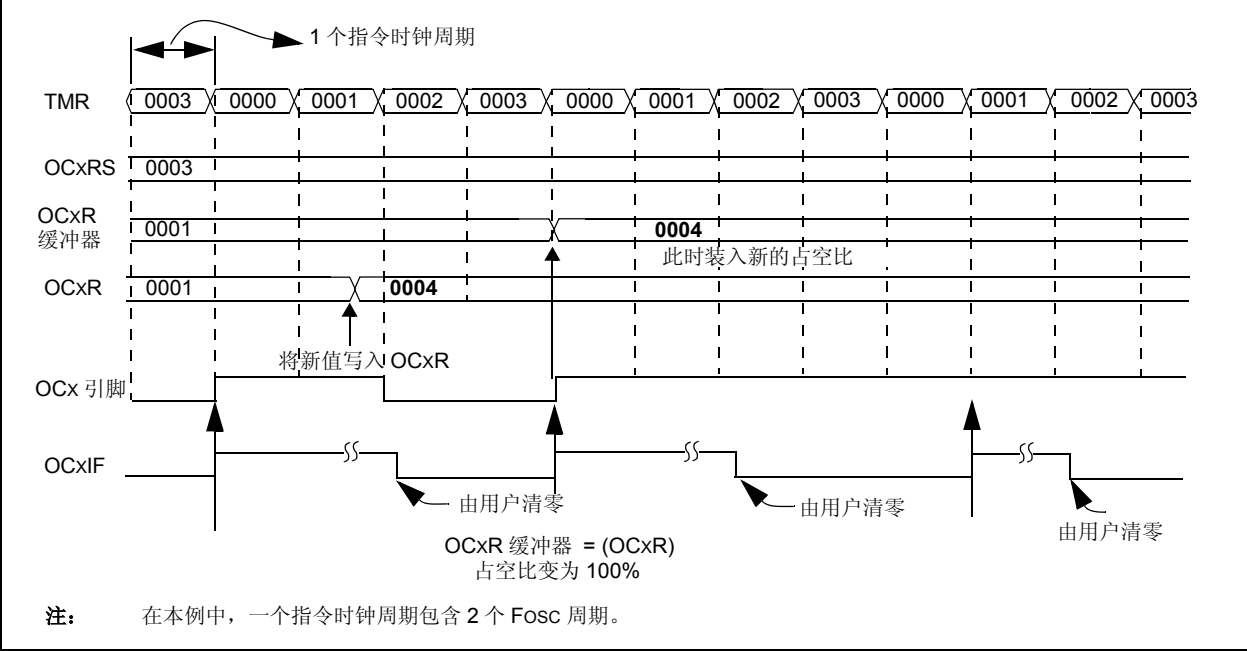
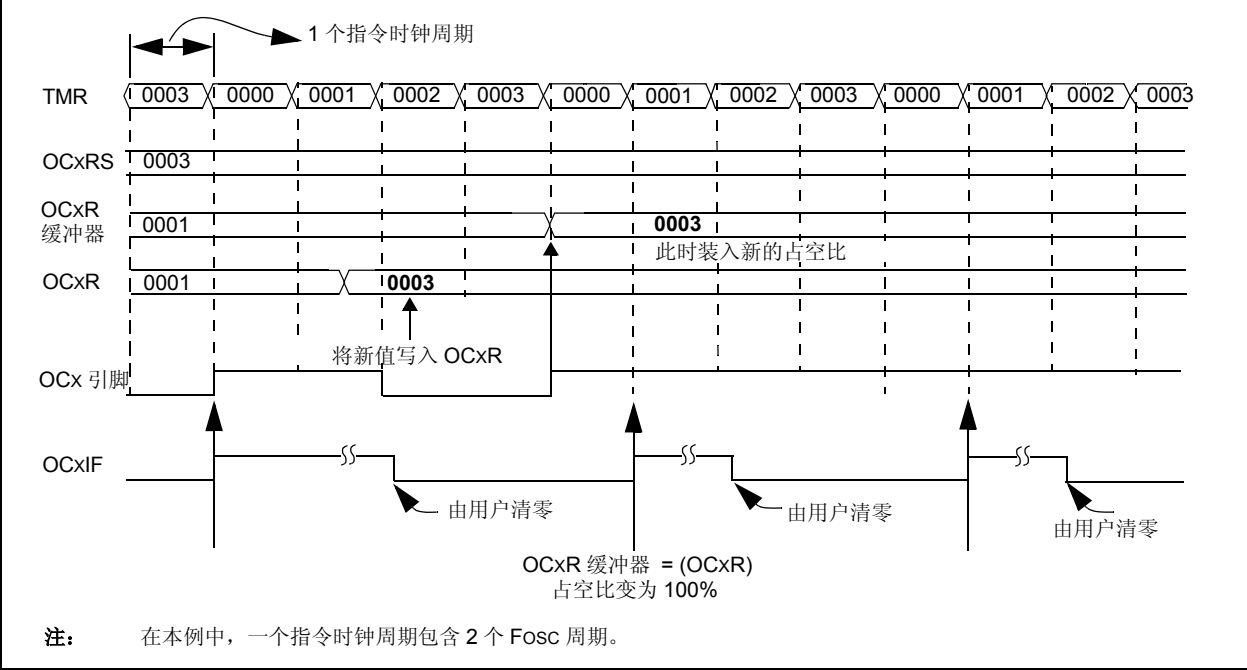


图 35-20: PWM 输出时序 (OCxR = OCxRS (SYNCSEL<4:0> = 0x1F))



35.3.3.5 中心对齐的 PWM 模式

这些模式（OCM<2:0> = 0b111）功能与连续脉冲模式（OCM<2:0> = 0b101）相同。仅有的区别是：

- OCxR 和 OCxRS 是双重缓冲的，这意味着新的寄存器值只有在定时器计满返回或发生同步之后才会生效。
- 使用故障控制和引脚

注 1： 中心对齐并不意味着脉冲严格对齐到脉冲宽度的中心。它只是表示脉冲的有效时间可以定位到周期内的任意位置。

35.3.3.6 故障输入和控制

当工作于中心对齐的 PWM 模式或边沿对齐的 PWM 模式（OCM<2:0>（OCxCON1<2:0>）= 0b111 或 0b110）时，可以激活故障引脚及其控制。有两个故障输入（OCFA 和 OCFB）可供使用，其中，OCFA 控制 OC1-OC4，OCFB 控制 OC5-OC9。故障引脚由寄存器位 ENFLT0（OCxCON1<15>）控制。如果该位为零，则相应的故障输入引脚（OCFA 或 OCFB）被忽略。故障输入的状态可以在寄存器位 OCFLT0（OCxCON1<4>）中观察。

当发生故障（OCFx = 0）时，OCx 引脚输出电平由 FLTOUT（OCxCON2<14>）位决定。在故障条件期间，OCx 引脚的三态由 FLTTRIEN（OCxCON2<13>）控制。

根据 FLTMODE（OCxCON2<15>）位，故障控制可以两种模式工作：

- 无效模式
- 逐周期模式。关于故障引脚如何分配给各种 OCx 外设的详细信息，请参见产品数据手册。

注： 输出比较故障引脚 OCFA 和 OCFB 是低电平有效引脚。

35.3.3.6.1 无效模式

当 FLTMODE（OCxCON2<15>）= 1 时，故障输入以无效模式工作（见图 35-21）。如果故障输入变为有效（0），OCFLT0（OCxCON1<12>）位将置 1，模块将处于故障状态。

它将保持在故障状态，直到同时满足以下三个条件：

- 故障输入变为无效
- OCFLT0（OCxCON1<12>）位用软件清零
- 启动了新的定时器周期（定时器变为 0000h）

35.3.3.6.2 逐周期模式

当 FLTMODE（OCxCON2<15>）= 0 时，故障输入以逐周期模式工作（见图 35-22）。如果故障输入变为有效（0），OCFLT0（OCxCON1<12>）位将置 1，模块将处于故障状态。

它将保持在故障状态，直到同时满足以下两个条件：

- 故障输入变为无效
- 启动了新的定时器周期（定时器变为 0x0000）

图 35-21: 故障输入引脚时序, 无效模式

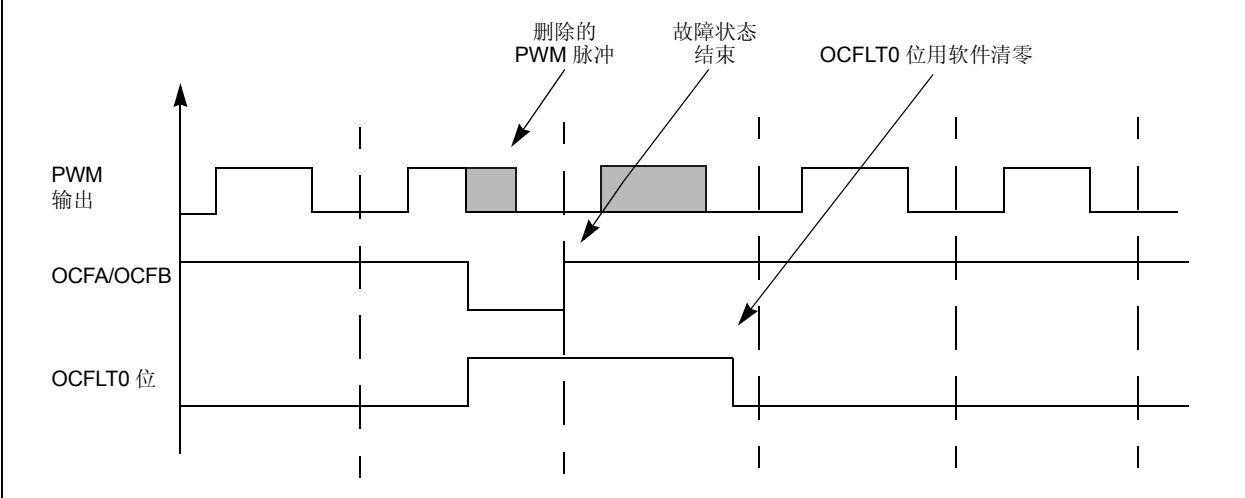
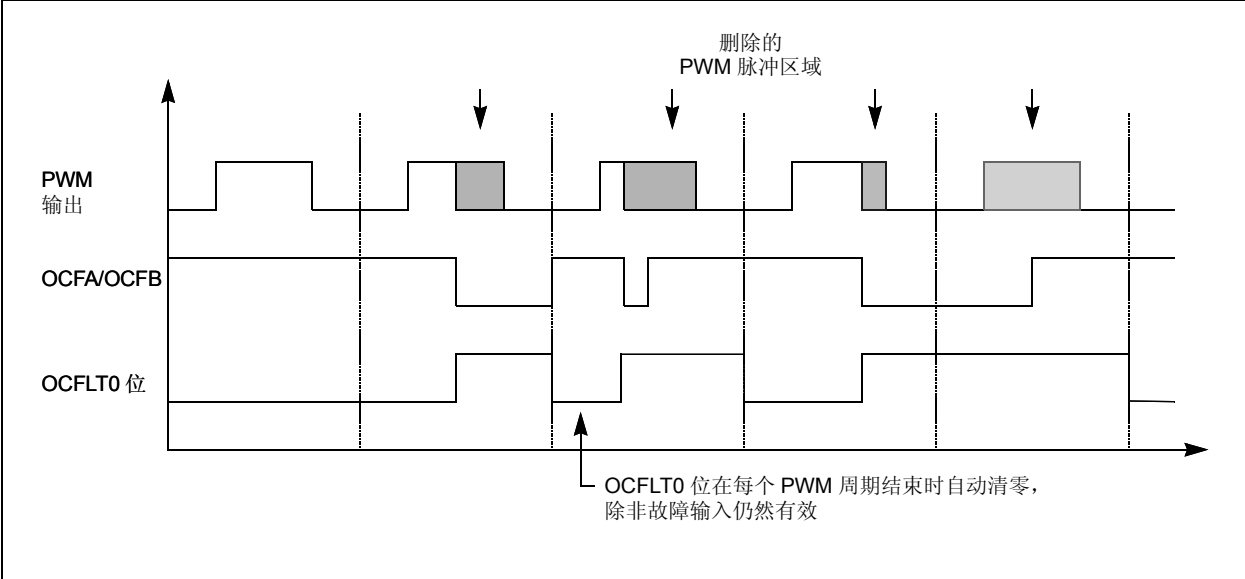


图 35-22: 故障输入引脚时序, 逐周期模式



35.3.3.7 同步操作

当 OCTRIG (OCxCON2<7>) = 0 时，即使能了定时器的同步操作。

在同步操作中，TRIGSTAT (OCxCON2<6>) 位没有任何作用。定时器可以通过使用同步 / 触发输入与其他模块进行同步（见寄存器 35-2）。每当选定的模块接收到同步信号时，定时器将在选定时钟的下一个负边沿返回到 0x0000。

35.3.3.8 在同步应用中使用模块定时器

图 35-23 给出了同步的连接图示，图 35-24 给出了多个模块进行同步的时序图示。OC2 与 OC1 进行同步。OC1 和 OC2 使用 SYNCSEL<4:0> (OCxCON2<4:0>) 位选择来自 OC1 的同步信号来进行同步。OC1RS 寄存器现在成为 OC1 和 OC2 的周期寄存器。

当 OC1RS 寄存器与 OC1 定时器值匹配时，OC1 模块会产生同步信号。这会导致 OC1 和 OC2 中的定时器在下一个正时钟边沿变为零。

注： 进行同步的模块应选择相同的时钟源，以确保正常工作。

图 35-23: 同步操作集成 (TRIGEN = 0)

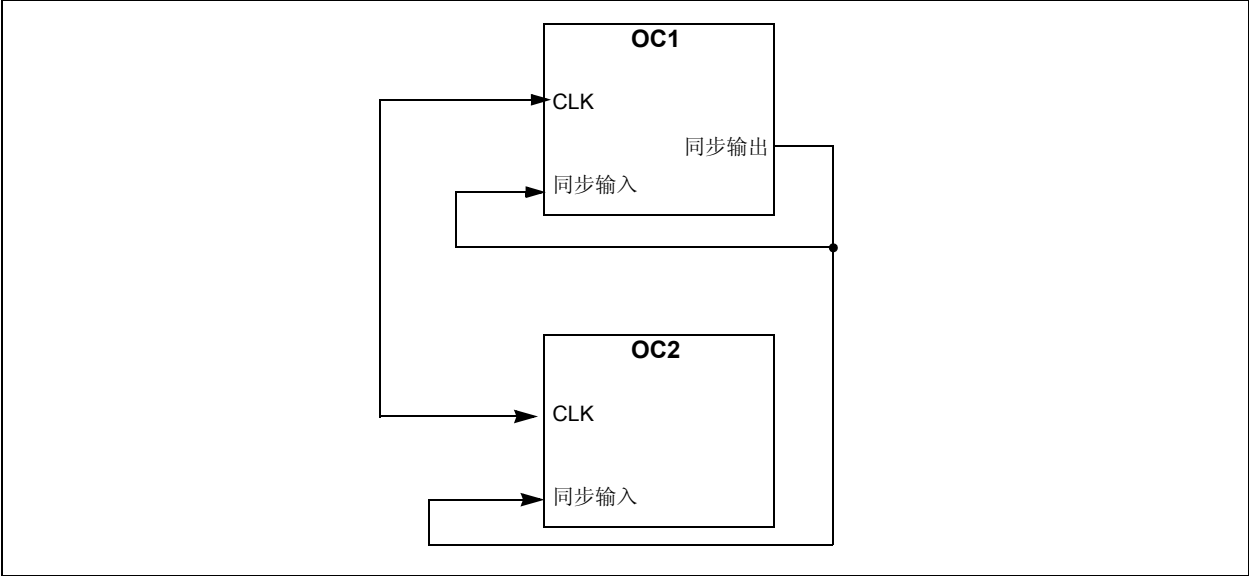
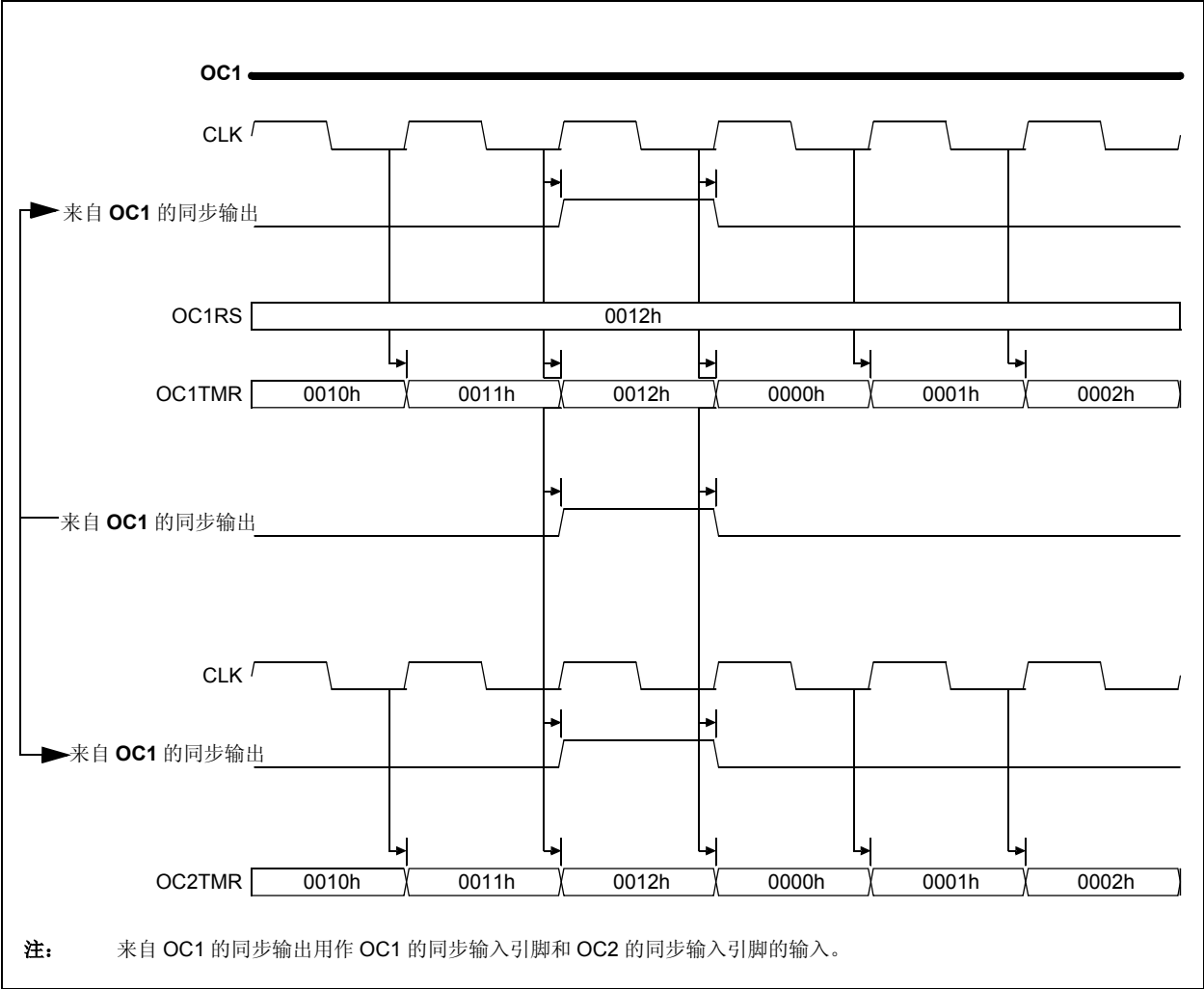


图 35-24: 同步操作



当初始化要进行同步的模块时，用作同步源的模块应最后使能。如图 35-23 中所示，OC2 应首先初始化，OC1 应最后初始化。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。

35.3.3.9 触发操作

当 OCTRIG (OCxCON2<7>) = 1 时，即使能了定时器的触发操作。当配置为进行触发操作时，模块定时器保持在复位状态，直到发生触发事件为止。在发生触发事件之后，定时器开始计数。触发源通过 SYNCSEL 位选择。

35.3.3.10 OCxCON2 触发功能

TRIGSTAT (OCxCON2<6>) 位用于将定时器保持在复位状态，或释放定时器使之开始计数。它通过以下方式控制定时器：

- TRIGSTAT = 0
 - 定时器保持在复位状态
- TRIGSTAT = 1
 - 定时器从复位状态释放
 - 定时器在每个时钟的正边沿递增

当工作于触发模式时，存在两种触发条件：

- 由硬件 / 软件将 TRIGSTAT 位置 1
- 仅用软件将 TRIGSTAT 位置 1

在两种情况下，触发器始终用软件清零。

35.3.3.10.1 由硬件 / 软件将 TRIGSTAT 置 1

在以下情况时，TRIGSTAT (OCxCON2<6>) 位可由硬件或软件置 1：

- SYNCSEL (OCxCON2<4:0>) 位不等于 0b00000 (见第 35.3.3.12 节“非法设置”)

当模块使能为触发响应时，定时器将保持在清零状态。它会一直保持在这种清零状态，直到发生触发事件，该事件会将 TRIGSTAT 位置 1。此外，也可以通过写 TRIGSTAT 位来置 1 该位，将定时器从复位状态释放。

35.3.3.10.2 仅用软件将 TRIGSTAT 置 1

当 SYNCSEL<4:0> = 0b00000 时，TRIGSTAT 位只能用软件置 1。

35.3.3.11 清零 TRIGSTAT 位

TRIGSTAT 位只能用软件清零，方法是向该位写入 0。当用软件清零 TRIGSTAT 位时，定时器会在下一个定时器时钟的上升沿复位为 0x0000，并可用于下一次触发。

35.3.3.12 非法设置

模块选择自身作为触发源是非法的。因此，在触发模式下，不允许使用 SYNCSEL<4:0> 的两个可能值：

- SYNCSEL<4:0> = 0x1F
- SYNCSEL<4:0> = N；其中，N 是选择相同模块的第二种设置（见寄存器 35-2）。

- | |
|--|
| <p>注 1: 当工作于单次触发模式时，不能在软件中更改 TRIGSTAT。关于该模式的更多信息，请参见第 35.3.3.13.2 节“单次触发功能”。</p> <p>2: 触发源将与 OCx 时钟进行同步。</p> <p>3: 在用户软件中应注意防止这些非法条件。</p> |
|--|

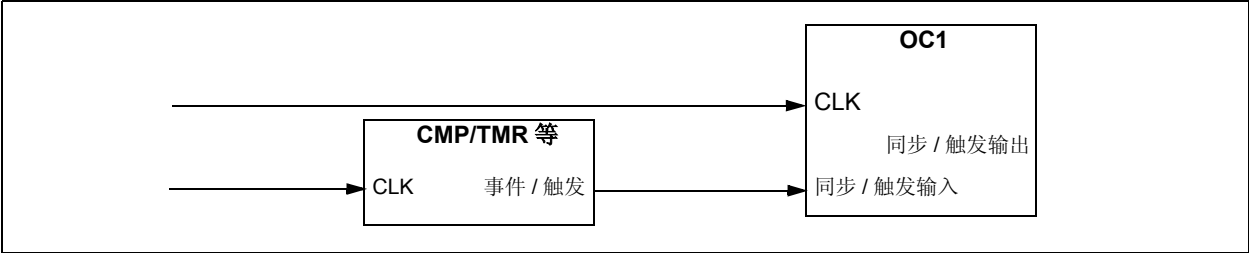
35.3.3.13 在触发应用中使用 OCx 模块

图 35-25 给出了在触发应用中模块定时器的典型应用的图示。在该应用中，触发事件可以由另一个输出比较模块、定时器模块、IC 模块、模拟比较器或其他外设功能产生。关于触发源的完整列表，请参见产品数据手册。

35.3.3.13.1 在触发应用中初始化 OCx 模块

用户会错过在初始化 OCx 模块之前发生的所有触发事件。因此，要避免错过触发事件，建议在使能触发源之前先使能模块。

图 35-25: 触发操作集成 (TRIGEN = 1)



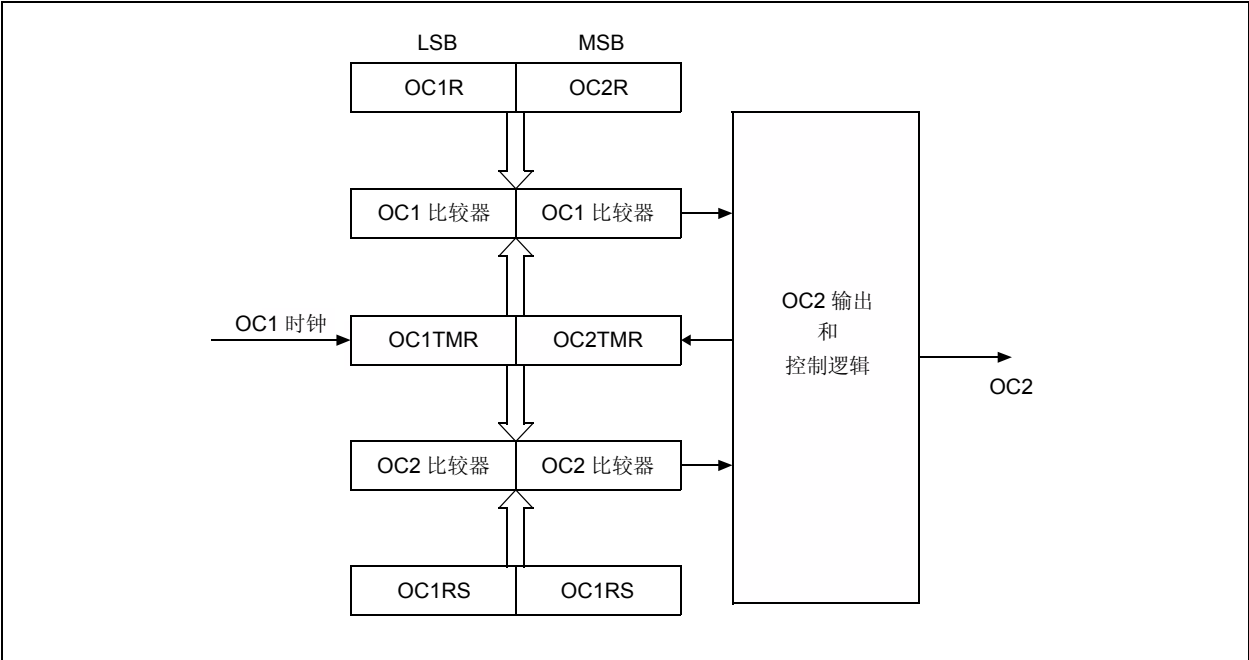
35.3.3.13.2 单次触发功能

当用作触发器时，定时器可以单次触发模式工作。该模式为每次触发产生一个脉冲。单次触发模式通过将 TRIGMODE (OCxCON1<3>) 位置 1 来使能。在单次触发模式下，定时器保持在复位状态，直到发生触发事件。该事件会将 TRIGSTAT 位置 1，然后定时器开始计数。当定时器计满返回到 0000h 时，如果 TRIGMODE = 1，则 TRIGSTAT 位将由硬件清零。这会将定时器保持在复位状态，直到发生下一次触发事件，从而构成了单次触发定时器。

35.3.4 级联模式

当 16 位定时器不够时，可以将 OCx 模块成对组合，级联组成 32 位定时器（见图 35-26）。它们组合为奇编号和偶编号对（1-2、3-4 和 5-6 等）。当进行级联时，奇编号 OCx 模块构成定时器 / 比较器的低 16 位，偶编号模块构成高 16 位。偶编号模块的 OCx 引脚将作为级联定时器的输出。

图 35-26: 级联操作



35.3.5 设置模块用作级联操作

在本节，假定 OC1 是奇编号 OC 模块，OC2 是偶编号模块。

对奇编号模块进行如下设置：

- OC32 (OC1CON1<8>) = 1
- OCTRIG (OC1CON2<7>) 可以为 1 或 0，根据定时器是进行同步或进行触发
- OCTRIS (OC1CON2<5>) = 1 (因为 OC1 引脚将不用，输出应为三态)

对偶编号模块进行如下设置：

- OC32 (OC2CON1<8>) = 1
- OCTRIG (OC2CON2<7>) = 0 (进行级联时，偶编号定时器必须以同步模式工作)
- OCTRIS (OC2CON2<5>) = 0 (因为 OC2 将不用，输出应使能)

35.3.5.1 在级联应用中初始化模块

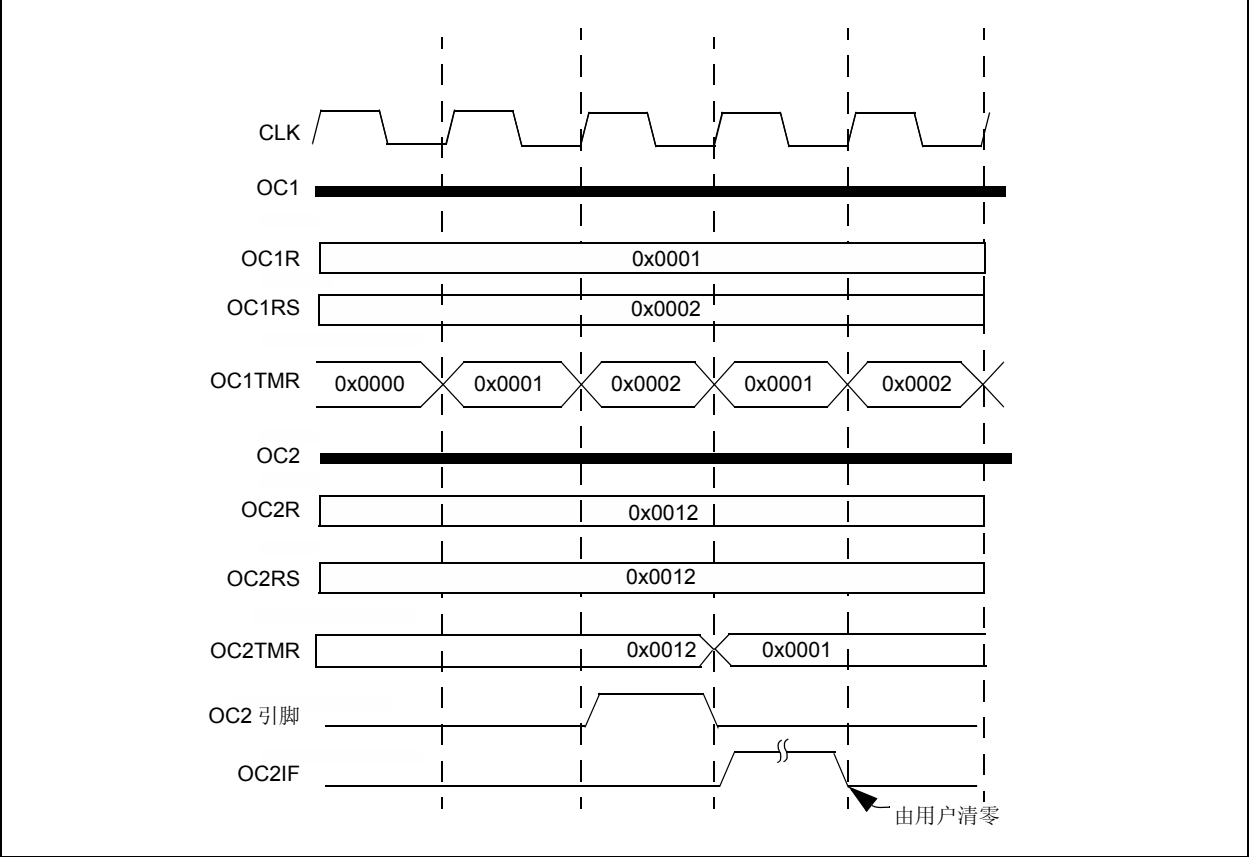
当初始化进行级联的模块时，偶编号模块应首先初始化，奇编号模块应最后初始化。

35.3.5.2 定时器时钟选择

该时钟应在使能模块之前选择，并且在工作期间不应更改。图 35-27 给出了级联操作的波形图示。

注： 偶编号和奇编号 OC 模块必须具有相同的时钟。

图 35-27： 双比较模式下的级联操作



35.3.5.3 由奇编号模块触发的级联操作

当两个模块进行级联来构成 32 位定时器时，可以通过设置奇编号模块（OCTRIG = 1）来触发定时器。奇编号模块保持在复位状态，直到发生触发事件。发生触发事件之后，奇编号和偶编号模块开始正常计数。

35.3.5.4 同步多个进行级联的模块对

多个 32 位模块对也可以进行同步，例如：

- 要将 OC3 + OC4 对与 OC1 + OC2 对进行同步：
 - OC1 和 OC2 按第 35.3.5 节“设置模块用作级联操作”中的定义进行设置
 - OC3 和 OC4 以相同的方式设置，但 SYNCSEL = 1（同步输出信号来自 OC1）；这允许来自 OC1 的同步输出信号将 OC3 保持在复位状态

例 35-5: 级联模式下的输出比较

```
OC1CON1 = 0;           /* It is a good practice to clear off the control bits initially */
OC1CON2 = 0;
OC2CON1 = 0;
OC2CON2 = 0;

OC1CON1bits.OCTSEL = 0x07; /* This selects the peripheral clock as the clock input to the OC
                             module */
OC2CON1bits.OCTSEL = 0x07;

OC1R = 0x1000;          /* Determines the On-Time */
OC2R = 0x0002;          /* Determines the On-Time */

OC1RS = 0x2000;         /* Determines the Period */
OC2RS = 0x0003;         /* Determines the Period */

OC1CON2bits.SYNCSEL = 0x1F;
OC2CON2bits.SYNCSEL = 0x1F;

OC1CON2bits.OCTRIS = 1; /* Odd module's output is not required */

/* Even module must be enabled first */
/* Odd module must be enabled last */

OC2CON2bits.OC32 = 1;
OC1CON2bits.OC32 = 1;

OC2CON1bits.OCM = 6;     /* This selects the Edge Aligned PWM mode */
OC1CON1bits.OCM = 6;
```

35.4 节能状态下的输出比较操作

35.4.1 休眠模式下的输出比较操作

当器件进入休眠模式后，系统时钟被禁止。在休眠期间，输出比较通道会将引脚驱动为与在进入休眠状态之前相同的有效状态。然后模块将停止在该状态。

例如：

如果引脚原先为高电平，则在 CPU 进入休眠状态后，引脚保持为高电平。类似地，如果引脚原先为低电平，则在 CPU 进入休眠状态后，引脚保持为低电平。在两种情况下，在器件唤醒时，输出比较模块都将继续工作。

35.4.2 PWM 故障模式时休眠

- 当模块处于 PWM 故障模式时，故障电路的异步部分将保持工作状态。
- 如果检测到故障，OCx 的输出由 OCxCON2 寄存器的 FLTOUT 和 OCTRIS 设置决定。
- OCFLT0 位将被置 1。中断不会在发生故障时产生，而是加入队列并在器件被唤醒时产生。

35.4.3 空闲模式下的输出比较操作

当器件进入空闲模式后，系统时钟源保持工作，但 CPU 停止执行代码。OCSIDL (OCxCON<13>) 位用于选择输出比较模块在空闲模式下是停止工作还是继续工作。

- 如果 OCSIDL = 1，则在空闲模式下模块将停止工作。在空闲模式下停止 (OCSIDL = 1) 时，模块将执行与休眠模式相同的过程。
- 如果 OCSIDL = 0，当 OCSIDL 位是逻辑 0 时，输出比较通道将在 CPU 空闲模式下工作。此外，还必须通过将相应的 TSIDL 位设置为逻辑 0 来使能时基；如果是在内部，定时器将用作时钟源。

注： 在器件处于休眠或空闲模式时，外部故障引脚（如果使能）将继续控制相关的 OCx 输出引脚。

35.4.4 打盹模式

打盹模式下的输出比较操作和正常模式下的一样。当器件进入打盹模式后，系统时钟源保持工作，但 CPU 可能以较低时钟速率运行。

35.4.5 选择性外设模块控制

外设模块禁止 (Peripheral Module Disable, PMD) 寄存器提供了一种方法，可通过停止向输出比较模块提供的所有时钟源来禁止该模块。当通过相应的 PMD 控制位禁止模块后，模块处于最低功耗状态。与模块相关的控制寄存器和状态寄存器也将被禁止。因此，对这些寄存器的写操作不起作用，读取的值也无效并返回零。

35.5 I/O 引脚控制

当输出比较模块被使能时，I/O 引脚方向由比较模块控制。当比较模块被禁止时，它会将 I/O 引脚控制权归还给相应的 LAT 和 TRIS 控制位。当使能了具有故障保护输入模式的简单 PWM 时，必须通过将相应的 TRIS 位置 1 以将 OCFA/OCFB 故障引脚配置为输入。使能此特殊 PWM 模式并不会将 OCFA/OCFB 故障引脚配置为输入。

注： 关于可用输出比较引脚的信息，请参见产品数据手册。

35.6 寄存器映射

表 35-7 中提供了与 PIC24F 带专用定时器的输出比较模块相关的寄存器汇总。

表 35-7: 带专用定时器的输出比较寄存器映射

寄存器名称	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
OCxCON1	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLT0	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OCxCON2	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	0000
OCxRS	输出比较 x 辅助寄存器																0000
OCxR	输出比较 x 寄存器																0000
OCxTMR	定时器值 x 寄存器																xxxx

图注: — = 未实现，读为 0。复位值以十六进制显示。

注: 关于带专用定时器的输出比较映射的详细信息，请参见具体器件的数据手册。

35.7 电气规范

35.7.1 交流特性

图 35-28: 输出比较时序

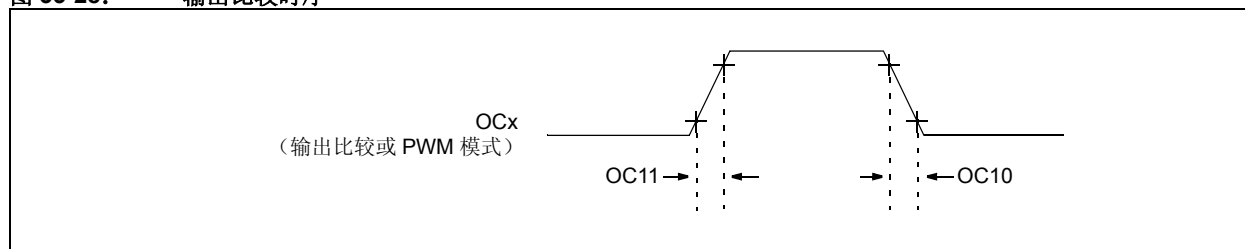


表 35-8: 输出捕捉

参数编号	符号	特性	最小值	最大值	单位	条件
OC11	TccR	OC1 输出上升时间	—	10	ns	—
			—	—	ns	—
OC10	TccF	OC1 输出下降时间	—	10	ns	—
			—	—	ns	—

图 35-29: PWM 模块时序要求

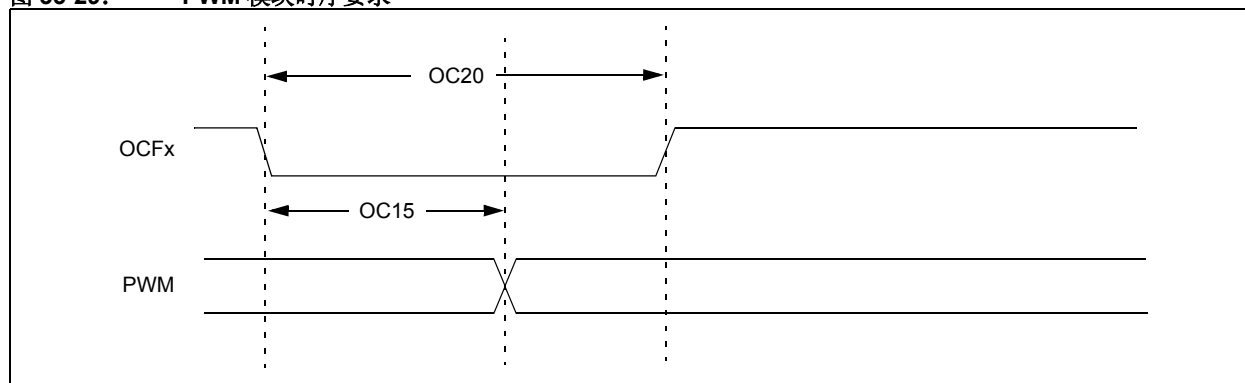


表 35-9: PWM 时序要求

参数编号	符号	特性	最小值	典型值 ⁽¹⁾	最大值	单位	条件
OC15	TfD	故障输入到 PWM I/O 发生变化的时间	—	—	25	ns	VDD = 3.0V, -40°C 至 +85°C
OC20	TfH	故障输入脉冲宽度	50	—	—	ns	VDD = 3.0V, -40°C 至 +85°C

注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

35.8 设计技巧

问 1: *即使当 OCSIDL 位没有置 1 时，输出比较引脚也停止工作。为什么？*

答: 如果相关定时器源用作时钟源，当它的 TSIDL (TxCON<13>) 位置 1 时，最可能发生此问题。因此，当执行 PWRSV 指令时，实际上是定时器进入了空闲模式，并未产生时钟。

问 2: *我可以将 OC2 和 OC3 级联在一起吗？*

答: 不能；只能对 OC1-OC2、OC3-OC4 等模块对进行级联，OC9 不能组对。

35.9 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 PIC24F 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。当前与带专用定时器的输出比较模块相关的应用笔记有：

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 PIC24F 系列器件的应用笔记和代码示例，请访问 Microchip 网站 (www.microchip.com)。

35.10 版本历史

版本 A（2008 年 2 月）

这是本文档的初始版本。

注: