

第 16 章 输出比较

目录

本章包括下列主题：

16.1 简介.....	16-2
16.2 输出比较寄存器.....	16-3
16.3 工作模式.....	16-4
16.4 省电状态下的输出比较操作.....	16-27
16.5 I/O 引脚控制.....	16-28
16.6 寄存器映射.....	16-29
16.7 电气规范.....	16-30
16.8 设计技巧.....	16-31
16.9 相关应用笔记.....	16-32
16.10 版本历史.....	16-33

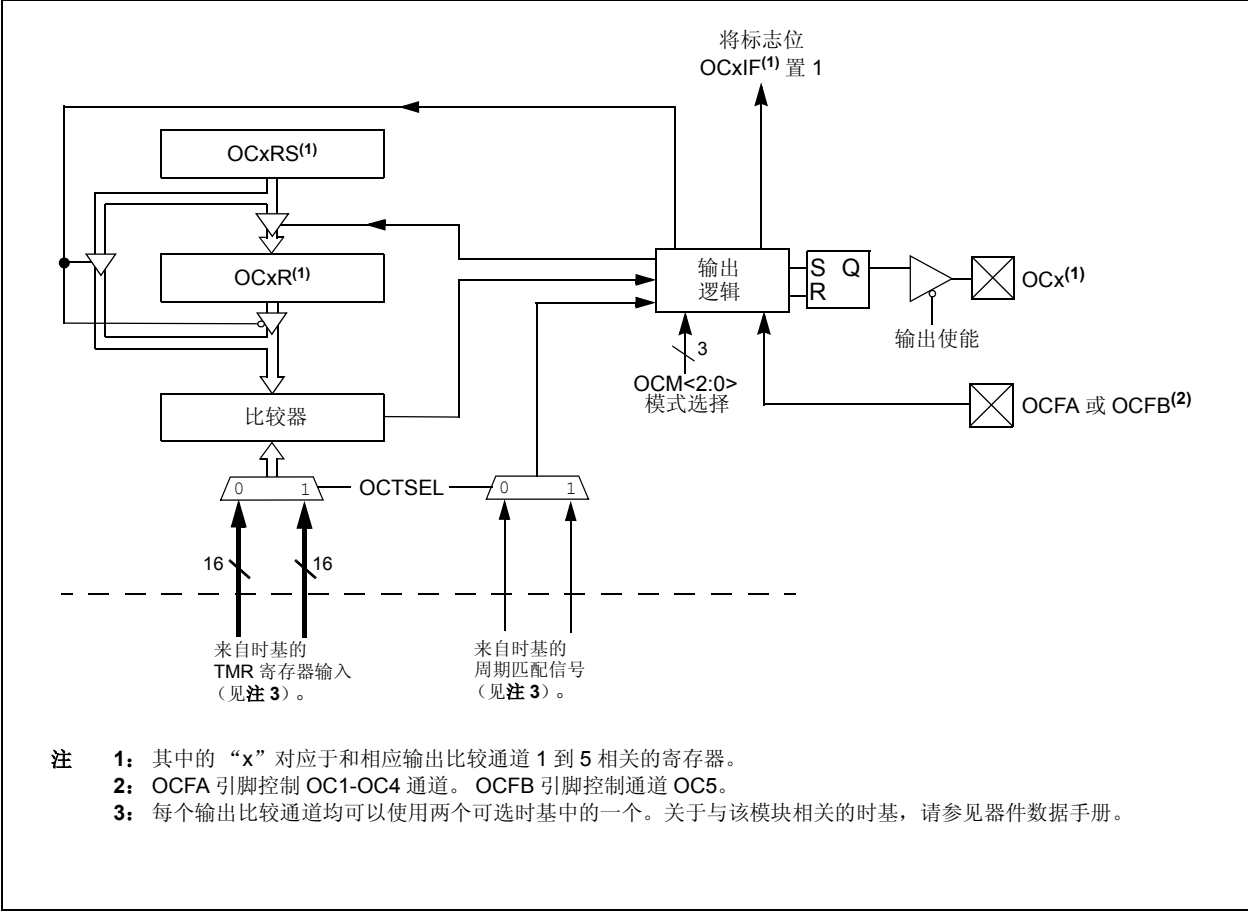
16.1 简介

输出比较模块能将所选时基值与一个或两个比较寄存器的值（取决于所选的工作模式）作比较。此外，它在比较匹配事件发生时能产生单个输出脉冲或一连串输出脉冲。如同大多数 PIC® 外设一样，它在比较匹配事件发生时还能产生中断。

关于在某个特定器件上可用的通道数，请参见具体器件的数据手册。所有输出比较通道在功能上都是相同的。在本章中，引脚、寄存器或位名称中的“x”表示特定的输出比较通道。

每个输出比较通道可以使用两个可选时基之一。使用 OCTSEL 位（OCxCON<3>）来选择时基。关于可以与各个编号的输出比较通道一起使用的特定定时器，请参见器件数据手册。可用时基 Timer2 和 Timer3 不支持异步模式。因此，输出比较模块只能在同步模式下工作。

图 16-1: 输出比较框图



16.2 输出比较寄存器

每个输出比较通道均有以下寄存器：

- OCxCON：输出比较通道的控制寄存器
- OCxR：输出比较通道的数据寄存器
- OCxRS：输出比较通道的辅助数据寄存器

5 个输出比较通道的控制寄存器被命名为 OC1CON 到 OC5CON。所有 5 个控制寄存器具有相同的位定义，表示为以下公共寄存器定义。OCxCON 中的 “x” 表示输出比较通道的编号。

寄存器 16-1: OCxCON: 输出比较 x 控制寄存器

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
—	—	OCSIDL	—	—	—	—	—
bit 15			bit 8				

U-0	U-0	U-0	R-0, HC	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	OCFLT	OCTSEL ⁽¹⁾	OCM2	OCM1	OCM0
bit 7			bit 0				

图注:	HC = 用硬件清零						
R = 可读位	W = 可写位		U = 未实现位，读为 0				
-n = POR 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 控制位
1 = 输出比较 x 将在 CPU 空闲模式下停止
0 = 输出比较 x 将在 CPU 空闲模式下继续工作
- bit 12-5 未实现：读为 0
- bit 4 **OCFLT**：PWM 故障条件状态位
1 = 发生了 PWM 故障条件（仅可用硬件清零）
0 = 未发生 PWM 故障条件（仅当 OCM<2:0> = 111 时，才使用该位）
- bit 3 **OCTSEL**：输出比较 x 定时器选择位 ⁽¹⁾
1 = Timer3 是输出比较 x 的时钟源
0 = Timer2 是输出比较 x 的时钟源
- bit 2-0 **OCM<2:0>**：输出比较 x 模式选择位
111 = OCx 处于 PWM 模式，故障引脚使能
110 = OCx 处于 PWM 模式，故障引脚禁止
101 = 初始化 OCx 引脚为低电平，在 OCx 引脚上产生连续输出脉冲
100 = 初始化 OCx 引脚为低电平，在 OCx 引脚上产生单个输出脉冲
011 = 比较事件使 OCx 引脚的电平交替翻转
010 = 初始化 OCx 引脚为高电平，比较事件强制 OCx 引脚为低电平
001 = 初始化 OCx 引脚为低电平，比较事件强制 OCx 引脚为高电平
000 = 输出比较通道禁止

注 1：关于输出比较模块可用的特定时基，请参见器件数据手册。

16.3 工作模式

每个输出比较模块均有以下工作模式：

- 单比较匹配模式
- 双比较匹配模式产生：
 - 单输出脉冲模式
 - 连续输出脉冲模式
- 简单脉宽调制（Pulse-Width Modulation，PWM）模式：
 - 带有故障保护输入
 - 不带故障保护输入

注 1： 建议用户在切换到新的模式之前，关闭输出比较模块（即，将 OCM<2:0>（OCxCON<2:0>）清零）。

2： 在本章中，对与所选定时器源相关的任何 SFR 的引用，均用“y”下标表示。例如，PRy 是所选定时器源的周期寄存器，而 TyCON 是所选定时器源的定时器控制寄存器。

16.3.1 单比较匹配模式

当控制位 OCM<2:0>（OCxCON<2:0>）被设置为 001、010 或 011 时，所选的输出比较通道被配置为 3 种单比较匹配模式中的一种。

在单比较匹配模式下，将一个值装入 OCxR 寄存器，并将该值与所选的递增定时器寄存器 TMRy 的值作比较。当比较匹配事件发生时，将发生以下事件之一：

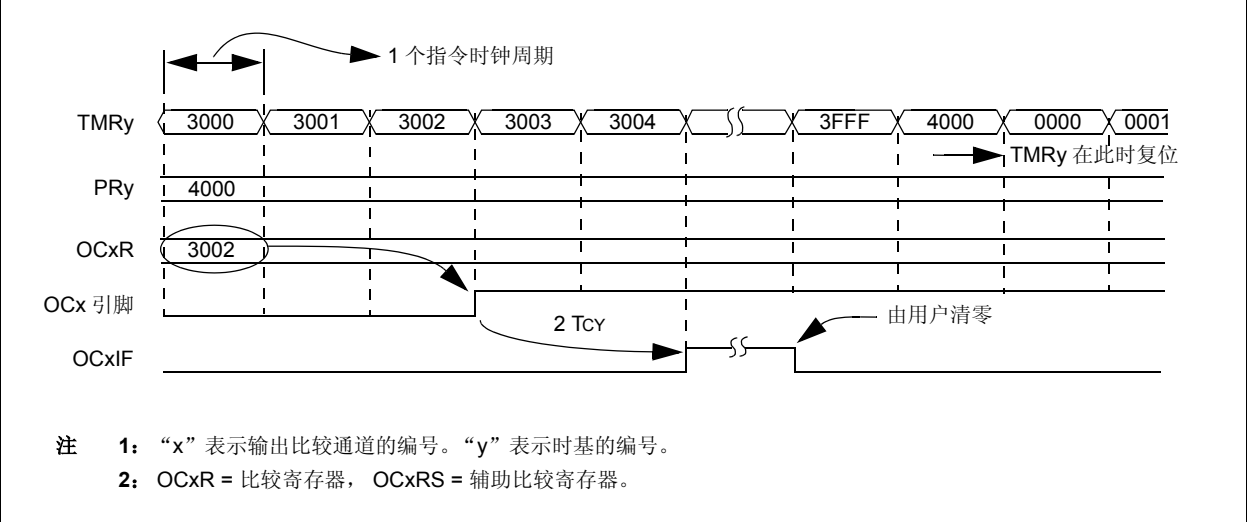
- 当 OCx 引脚的初始状态为低电平时，比较匹配事件强制该引脚为高电平。在单比较匹配事件发生时，产生中断。
- 当 OCx 引脚的初始状态为高电平时，比较匹配事件强制该引脚为低电平。在单比较匹配事件发生时，产生中断。
- 比较匹配事件使 OCx 引脚的电平交替翻转。翻转事件是连续的，且每次翻转事件都会产生一次中断。

16.3.1.1 单比较匹配模式输出驱动为高电平

要将输出比较模块配置为这种模式，需设置控制位 $OCM<2:0> = 001$ 。还应使能 $TMRy$ 。一旦使能了此比较模式，输出引脚 OCx 就被先驱动为低电平，并保持该低电平直到 $TMRy$ 和 $OCxR$ 寄存器之间发生匹配为止。请参见图 16-2，注意以下关键时序事件：

- 在 $TMRy$ 和 $OCxR$ 寄存器之间发生比较匹配后的下一个指令时钟， OCx 引脚被驱动为高电平。 OCx 引脚将保持高电平直到改变模式或该模块被禁止。
- $TMRy$ 将计数到相关的周期寄存器中包含的值后，在下一个指令时钟复位为 0000h。
- 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 16-2： 单比较匹配模式：在比较匹配事件发生时将 OCx 设置为高电平 (1,2)

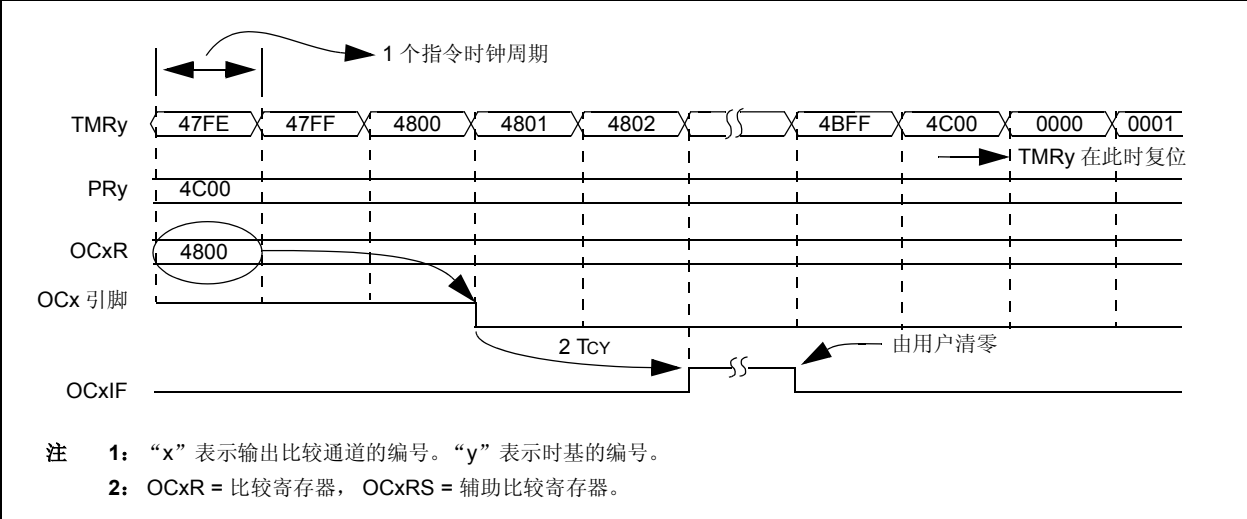


16.3.1.2 单比较匹配模式输出驱动为低电平

要将输出比较模块配置为这种模式，需设置控制位 $OCM<2:0> = 010$ 。还必须使能 $TMRy$ 。一旦使能了此比较模式，输出引脚 OCx 就被先驱动为高电平，并保持该高电平直到定时器和 $OCxR$ 寄存器之间发生匹配为止。请参见图 16-3，注意以下关键时序事件：

- 在 $TMRy$ 和 $OCxR$ 寄存器之间发生比较匹配后的下一个指令时钟， OCx 引脚被驱动为低电平。 OCx 引脚将保持低电平直到改变模式或该模块被禁止。
- $TMRy$ 将计数到相关的周期寄存器中包含的值后，在下一个指令时钟复位为 0000h。
- 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 16-3： 单比较匹配模式：在比较匹配事件发生时强制 OCx 为低电平 (1,2)



16.3.1.3 单比较匹配模式交替翻转输出

要将输出比较模块配置为这种模式，需设置控制位 $OCM<2:0>=011$ 。还必须使能 $TMRy$ 。一旦使能了此比较模式，输出引脚 OCx 将初始化驱动为低电平，并在随后每一次当定时器和 $OCxR$ 寄存器之间发生匹配事件时，交替输出高低电平。请参见图 16-4 和图 16-5，注意以下关键时序事件：

- 在 $TMRy$ 和 $OCxR$ 寄存器之间发生比较匹配后的下一个指令时钟， 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、改变模式或该模块被禁止。
- $TMRy$ 将计数到周期寄存器中的值后，在下一个指令时钟复位为 0000h。
- 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。

注：器件复位时，内部 OCx 引脚输出逻辑被设置为逻辑 0。但是，在交替翻转模式下， OCx 引脚的工作状态可以通过用户软件设置。例 16-1 给出了在交替翻转工作模式下，定义所需的初始化 OCx 引脚状态的代码示例。

图 16-4：单比较匹配模式：在比较匹配事件发生时翻转输出电平（ $PRy > OCxR$ ）^(1,2)

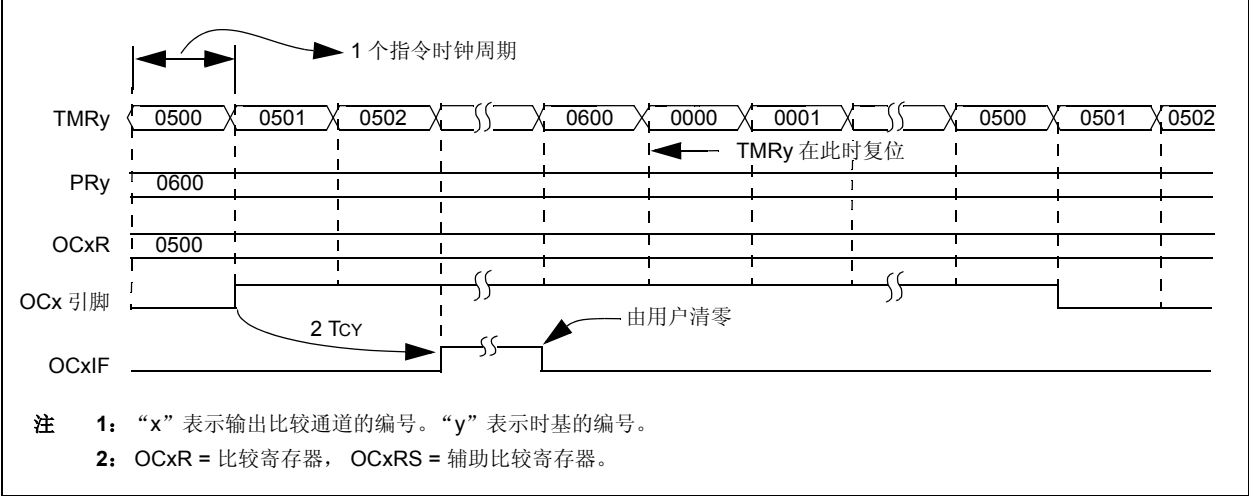
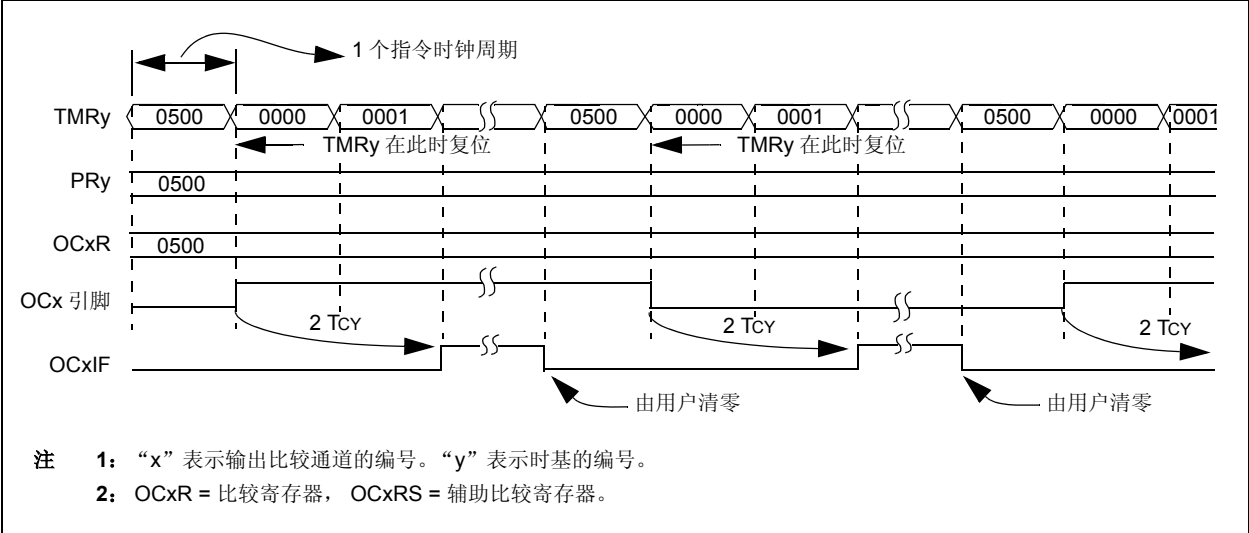


图 16-5：单比较匹配模式：在比较匹配事件发生时翻转输出电平（ $PRy = OCxR$ ）^(1,2)



例 16-1: 单比较匹配模式：翻转模式引脚状态设置

```
// The following code example illustrates how to define the initial
// OC1 pin state for the output compare toggle mode of operation.

// Toggle mode with initial OC1 pin state set low

OC1CON          = 0x0001;          // enable module for OC1 pin low, toggle high
OC1CONbits.OCM1 = 1;              // set module to toggle mode with initial pin
                                   // state low

// Toggle mode with initial OC1 pin state set high

OC1CON          = 0x0002;          // enable module for OC1 pin high, toggle low
OC1CONbits.OCM0 = 1;              // set module to toggle mode with initial pin
                                   // state high
```

例 16-2 给出了单比较匹配模式电平交替翻转事件的配置和中断服务程序的示例代码。

例 16-2: 单比较匹配模式：翻转设置和中断服务

```
// The following code example will set the Output Compare 1 module
// for interrupts on the toggle event and select Timer 2 as the clock
// source for the compare time-base. It is assumed that Timer 2
// and Period Register 2 are properly configured. Timer 2 will
// be enabled here.

OC1CON          = 0x0000;          // Turn off Output Compare 1 Module
OC1CON          = 0x0003;          // Load new compare mode to OC1CON
OC1R            = 0x0500;          // Initialize Compare Register1 with 0x0500
IPC0bits.OC1IP0 = 1;              // Setup Output Compare 1 interrupt for
IPC0bits.OC1IP1 = 0;              // desired priority level
IPC0bits.OC1IP2 = 0;              // (this example assigns level 1 priority)
IFS0bits.OC1IF  = 0;              // Clear Output Compare 1 interrupt flag
IEC0bits.OC1IE  = 1;              // Enable Output Compare 1 interrupts
T2CONbits.TON   = 1;              // Start Timer2 with assumed settings

// Example code for Output Compare 1 ISR:
void __attribute__((__interrupt__)) _OC1Interrupt(void)
{
    IFS0bits.OC1IF = 0;
}
```


16.3.1.4 单比较匹配模式的特殊情况

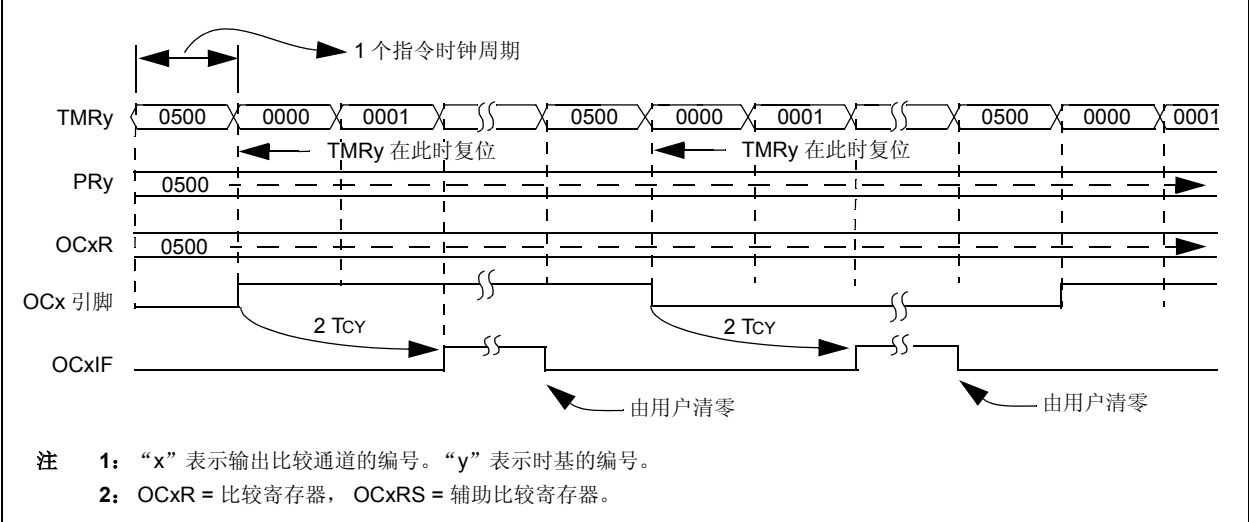
有几种特殊情况需要考虑。

当 $OCxR > PRy$ 时，表示比较值大于定时器计数，将不会发生比较事件，比较器输出将保持初始条件。当 $OCxR = PRy$ 时，表示比较时间间隔与定时器周期相同，比较输出将正常产生。将此方式与交替翻转模式结合使用可以产生固定频率的方波，如图 16-5 所示。

当模块使能为单比较匹配模式，而 $OCxR = 0000h$ ， $PRy = 0000h$ 时，表示未设置定时器计数周期，比较输出将保持初始条件。

如果在比较事件之后， $OCxR$ 和 PRy 寄存器被清零，则比较输出将保持为其先前状态。

图 16-6: 单比较匹配模式：在比较匹配事件发生时翻转输出电平（ $PRy > OCxR$ ）(1,2)



16.3.2 双比较匹配模式

当控制位 $OCM<2:0> = 100$ 或 101 ($OCxCON<2:0>$) 时，所选的输出比较通道被配置为如下两种双比较匹配模式之一：

- 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 $OCxRS$ 寄存器。将 $OCxR$ 寄存器的值与递增定时器计数器 $TMRy$ 的值作比较，并且在比较匹配事件发生时，在 OCx 引脚上产生脉冲的前（上升）沿。然后 $OCxRS$ 寄存器的值与同一个递增定时器计数器 $TMRy$ 的值作比较，并且在比较匹配事件发生时，在 OCx 引脚上产生脉冲的后（下降）沿。

16.3.2.1 双比较匹配模式：单输出脉冲

要将输出比较模块配置为单输出脉冲模式，需设置控制位 $OCM<2:0> = 100$ 。此外，必须选择并使能 $TMRy$ 。一旦使能了此模式，输出引脚 OCx 将驱动为低电平，并保持该低电平直到时基和 $OCxR$ 寄存器之间发生匹配为止。请参见图 16-7 和图 16-9，注意以下关键时序事件：

- 在 $TMRy$ 和 $OCxR$ 寄存器之间发生比较匹配后的下一个指令时钟， OCx 引脚被驱动为高电平。 OCx 引脚将保持高电平直到时基和 $OCxRS$ 寄存器之间发生下一次匹配事件为止。此时，该引脚将被驱动为低电平。 OCx 引脚将保持低电平直到改变模式或该模块被禁止。
- $TMRy$ 将计数到相关的周期寄存器中包含的值后，在下一个指令时钟复位为 $0000h$ 。
- 如果 $TMRy$ 寄存器包含的值小于 $OCxRS$ 寄存器包含的值，那么就不会产生脉冲的下降沿。 OCx 引脚将保持高电平直到 $CxRS \leq PRy$ 、模式改变或复位条件产生。
- 在 OCx 引脚被驱动为低电平后（单脉冲的下降沿）再过 2 个指令时钟，相应通道的中断标志位 $OCxIF$ 被置 1。

图 16-7 和图 16-8 描述了双比较匹配模式产生单输出脉冲的过程。图 16-9 描述了其他时序示例，其中 $OCxRS > PRy$ 。在此示例中，不产生脉冲的下降沿，因为 $TMRy$ 在计数达到 $4100h$ 前就复位了。

图 16-7： 双比较匹配模式 (1,2)

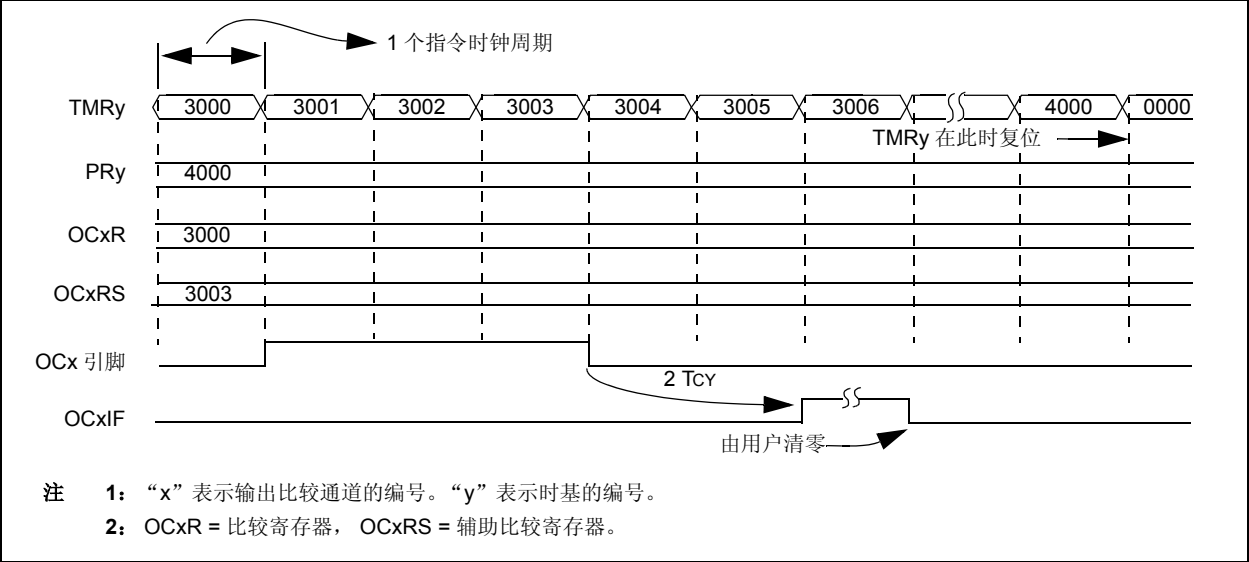


图 16-8: 双比较匹配模式：单输出脉冲模式 (1,2)

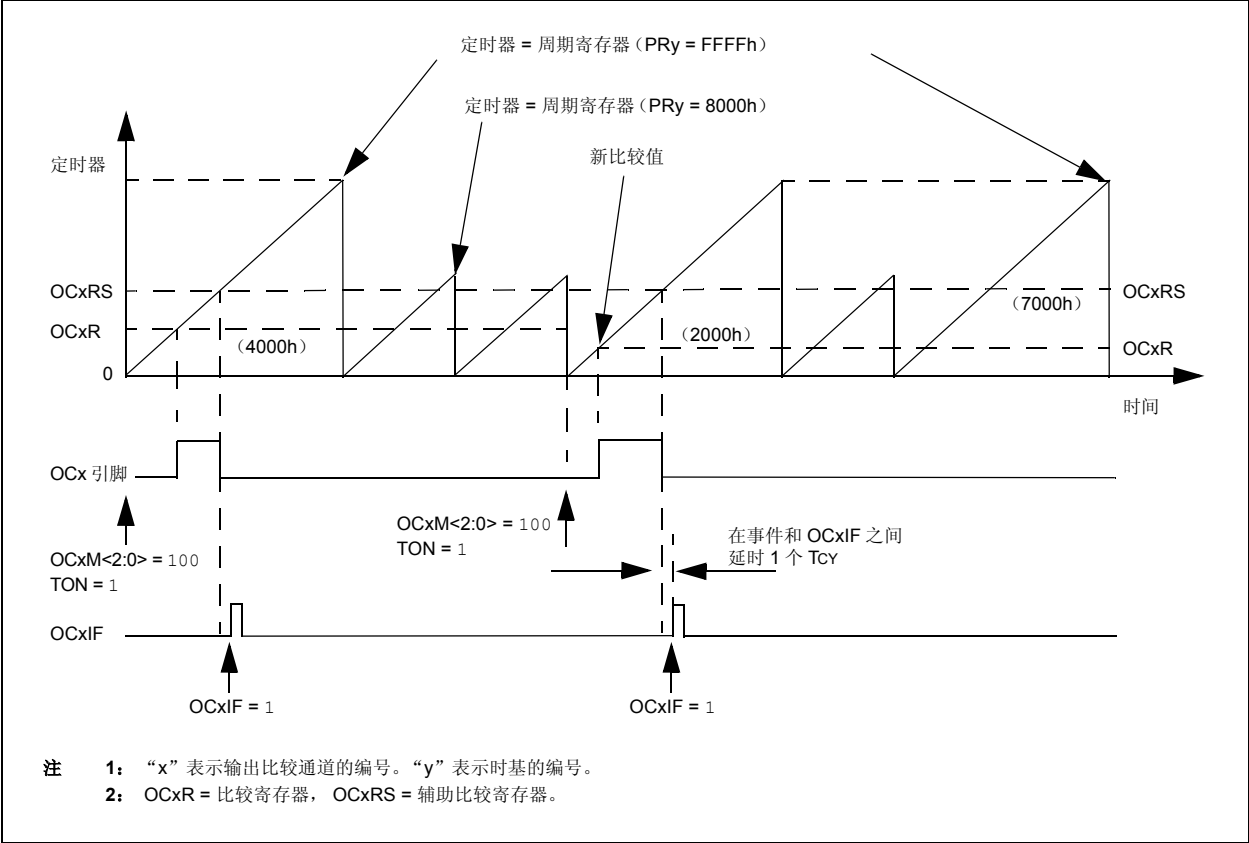
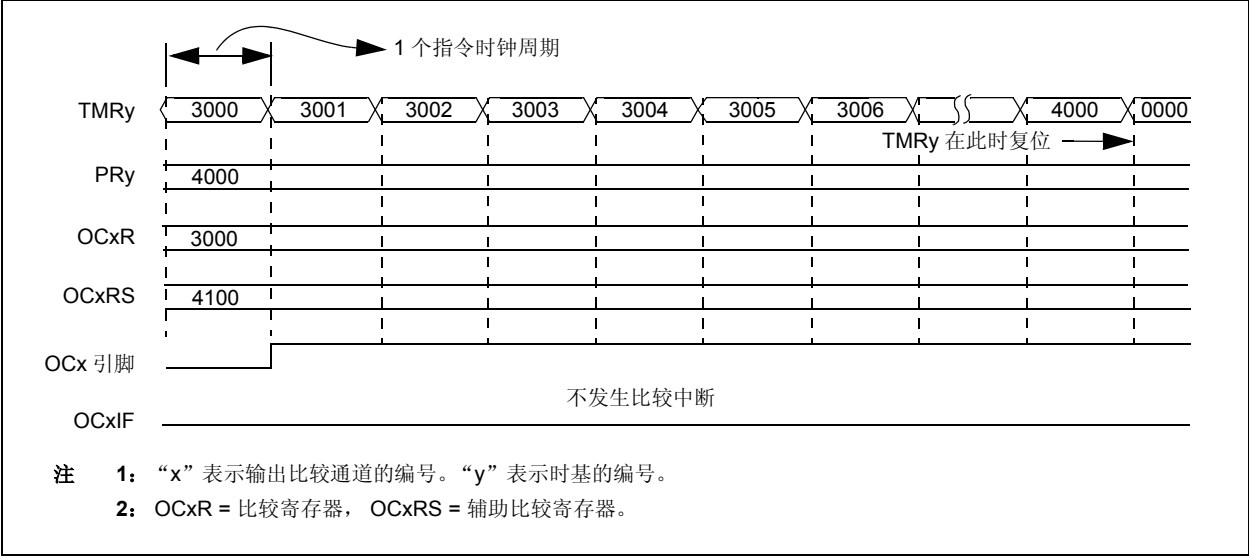


图 16-9: 双比较匹配模式：单输出脉冲模式 (OCxRS > PRy) (1,2)



16.3.2.2 设置产生单输出脉冲

当控制位 **OCM<2:0>** (**OCxCON<2:0>**) 被设置为 100 时, 所选的输出比较通道将 **OCx** 引脚初始化为低电平状态并产生单输出脉冲。

要产生单输出脉冲, 需要遵循以下步骤 (这些步骤假设定时源起初是关闭的, 但这并不是对模块操作的要求):

1. 确定指令时钟周期时间。将定时源的外部时钟频率 (如果使用了) 和定时器预分频比设置考虑进去。
2. 计算相对于 **TMRy** 起始值 (0000h) 的到达输出脉冲下降沿的时间。
3. 根据所需的脉冲宽度和脉冲上升沿时间计算到达脉冲下降沿的时间。
4. 将以上步骤2和步骤3中计算出的值分别写入比较寄存器 **OCxR** 和辅助比较寄存器 **OCxRS**。
5. 将定时器周期寄存器 **PRy** 的值设置为等于或大于辅助比较寄存器 **OCxRS** 中的值。
6. 设置 **OCM<2:0> = 100**, 并将 **OCTSEL** (**OCxCON<3>**) 设置为所需定时源的对应值。此时 **OCx** 引脚状态被驱动为低电平。
7. 将 **TON** (**TyCON<15>**) 位设置为 1, 它将使能 **TMRy** 开始计数。
8. 在 **TMRy** 和 **OCxR** 第一次匹配时, **OCx** 引脚将被驱动为高电平。
9. 当递增定时器 **TMRy** 和辅助比较寄存器 **OCxRS** 发生匹配时, 在 **OCx** 引脚上驱动脉冲的第二个边沿 (即后沿, 从高到低)。**OCx** 引脚上不会驱动输出更多脉冲, **OCx** 引脚将保持为低电平。发生第二次比较匹配事件后, 会导致 **OCxIF** 中断标志位置 1, 如果已通过将 **OCxIE** 位置 1 允许了中断, 则将产生中断。关于外设中断的更多信息, 请参见第 8 章 “中断”。
10. 要启动另一个单脉冲输出, 需更改定时器和比较寄存器的设置, 然后进行写操作将 **OCM<2:0>** (**OCxCON<2:0>**) 位设置为 100。不需要禁止和重新使能定时器并清零 **TMRy** 寄存器, 但这样做对于定义一个起始于已知事件时间边界的脉冲很有利。

在输出脉冲的下降沿出现之后, 不必禁止输出比较模块。重写 **OCxCON** 寄存器的值可以启动另一个脉冲。

注: 当预分频比为 1:1 时, **OCxR** 和 **OCxRS** 之间的最小时间差为 2 个 **Tcy**。

例 16-3 给出了单输出脉冲事件的示例代码。

例 16-3: 单输出脉冲模式设置和中断服务

```
// The following code example will set the Output Compare 1 module
// for interrupts on the single pulse event and select Timer 2
// as the clock source for the compare time-base. It is assumed
// that Timer 2 and Period Register 2 are properly initialized.
// Timer 2 will be enabled here.

OC1CON          = 0x0000;      // Turn off Output Compare 1 Module
OC1CON          = 0x0004;      // Load new compare mode to OC1CON
OC1R            = 0x3000;      // Initialize Compare Register1 with 0x3000
OC1RS           = 0x3003;      // Initialize Secondary Compare Register1 with 0x3003
IPC0bits.OC1IP0 = 1;           // Setup Output Compare 1 interrupt for
IPC0bits.OC1IP1 = 0;           // desired priority level
IPC0bits.OC1IP2 = 0;           // (this example assigns level 1 priority)
IFS0bits.OC1IF  = 0;           // Clear Output Compare 1 interrupt flag
IEC0bits.OC1IE  = 1;           // Enable Output Compare 1 interrupts
T2CONbits.TON   = 1;           // Start Timer2 with assumed settings

// Example code for Output Compare 1 ISR:
void __attribute__((__interrupt__)) _OC1Interrupt(void)
{
    IFS0bits.OC1IF = 0;
}
```

16.3.2.3 双比较匹配模式产生单输出脉冲的特殊情况

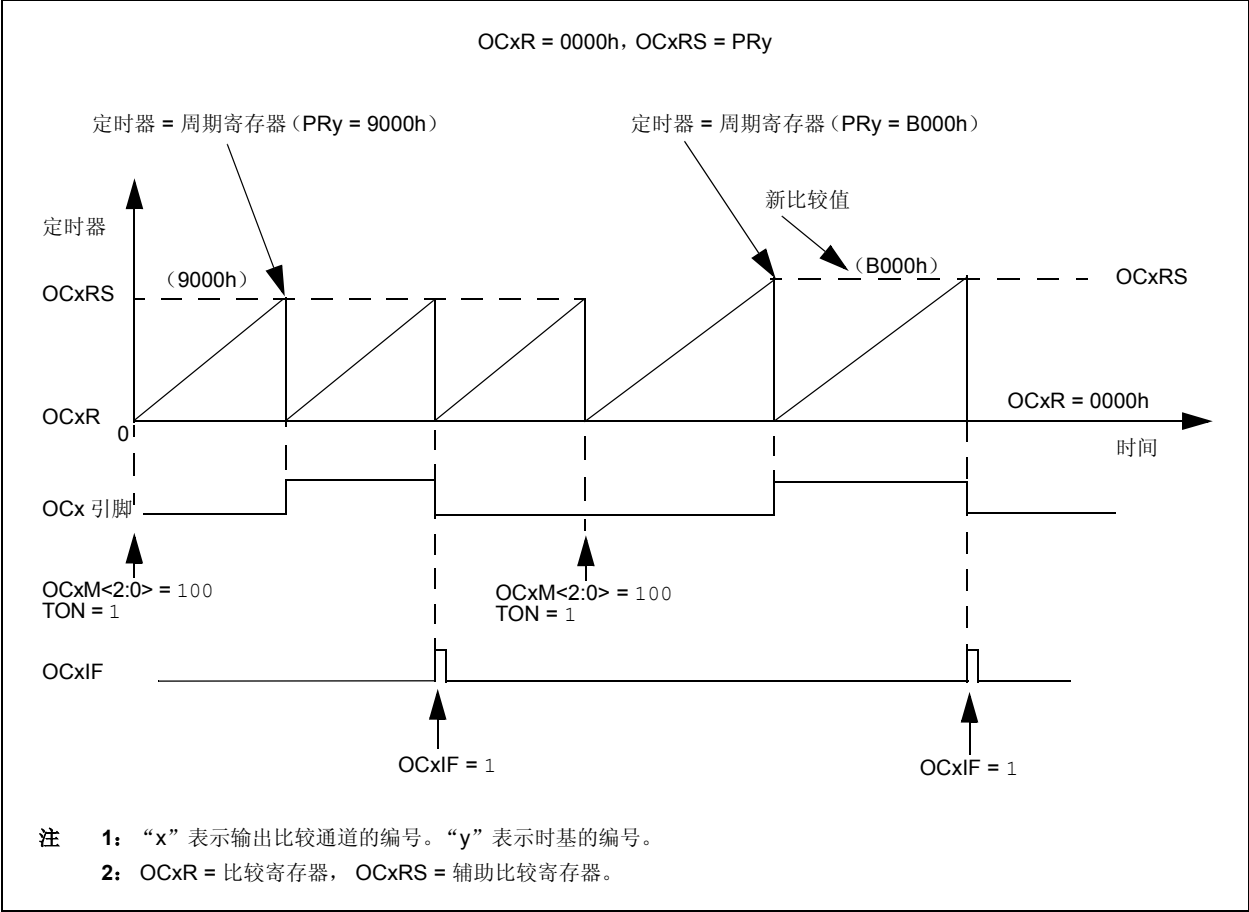
根据 OCxR、OCxRS 和 PRy 值的关系，应该认识到输出比较模块还有一些独特的条件。表 16-1 中说明了这些特殊条件以及模块在这些条件下的表现。

表 16-1: 双比较匹配模式产生单输出脉冲的特殊情况^(1,2)

SFR 逻辑关系	特殊条件	工作原理	OCx 引脚的输出
$PRy \geq OCxRS$ 且 $OCxRS > OCxR$	$OCxR = 0$ ， 初始化 $TMRy = 0$	在第一次 TMRy 从 0000h 计数至 PRy 时，OCx 引脚保持为低电平，不产生任何脉冲。在 TMRy 复位为 0（在周期匹配时）之后，OCx 引脚在 TMRy 与 OCxR 匹配时变为高电平。在下次 TMRy 与 OCxRS 匹配时，OCx 引脚变为低电平并保持为低电平。第二次比较之后，OCxIF 位会被置 1。 有两种可选的初始条件需要考虑： a) 初始化 $TMRy = 0$ ，并设置 $OCxR \geq 1$ 。 b) 初始化 $TMRy = PRy$ ($PRy > 0$)，并设置 $OCxR = 0$ （见图 16-10）。	脉冲将根据设置进行延时，延长时间对应于 PRy 寄存器中的值。
$PRy \geq OCxR$ 且 $OCxR \geq OCxRS$	$OCxR \geq 1$ 且 $PRy \geq 1$	TMRy 计数至 OCxR 的值，并在发生比较匹配事件时（即， $TMRy = OCxR$ ），OCx 引脚驱动为高电平状态。然后，TMRy 继续计数，最终在周期匹配时（即， $PRy = TMRy$ ）复位。然后，定时器重新从 0000h 开始计数至 OCxRS 值，并在发生比较匹配事件时（即， $TMRy = OCxRS$ ），OCx 引脚驱动为低电平状态。第二次比较之后，OCxIF 位会被置 1。	脉冲。
$OCxRS > PRy$ 且 $PRy \geq OCxR$	无	在 OCx 引脚将仅产生上升沿。OCxIF 将不会被置 1。	上升沿 / 转变为高电平。
$OCxR = OCxRS = PRy = 0000h$	无	输出初始化为低电平，并保持为低电平。OCxIF 位将不会被置 1。	保持低电平。
$OCxR > PRy$	无	不支持此模式，定时器在匹配条件之前复位。	保持低电平。

注 1: 本表所述的所有情形下，均假设 TMRy 寄存器被初始化为 0000h。
2: OCxR = 比较寄存器，OCxRS = 辅助比较寄存器，TMRy = 定时器计数寄存器，PRy = 定时器周期寄存器。

图 16-10: 双比较匹配模式：单输出脉冲模式（OCxR = 0000h，OCxRS = PRy）^(1,2)



16.3.2.4 双比较匹配模式：连续输出脉冲

要将输出比较模块配置为这种模式，需设置控制位 $OCM<2:0> = 101$ 。此外，必须选择并使能 $TMRy$ 。一旦使能了此模式，输出引脚 OCx 将驱动为低电平，并保持该低电平直到 $TMRy$ 和 $OCxR$ 寄存器之间发生匹配为止。请参见图 16-11 和图 16-13，注意以下关键时序事件：

- 在 $TMRy$ 和 $OCxR$ 寄存器之间发生比较匹配后的下一个指令时钟， OCx 引脚被驱动为高电平。 OCx 引脚将保持高电平直到时基和 $OCxRS$ 寄存器之间发生下一次匹配事件为止，此时引脚将驱动为低电平。在 OCx 引脚将重复这种脉冲发生序列（即，从低电平变为高电平的边沿，然后是从高电平变为低电平的边沿），而无需用户进一步干预。
- 在 OCx 引脚将产生连续的脉冲，直到模式发生改变或模块被禁止。
- $TMRy$ 将计数到相关的周期寄存器中包含的值后，在下一个指令时钟复位为 $0000h$ 。
- 如果 $TMRy$ 周期寄存器包含的值小于 $OCxRS$ 寄存器包含的值，那么就不会产生下降沿。 OCx 引脚将保持高电平，直到 $OCxRS \leq PRy$ 、模式发生改变或器件复位。
- 在 OCx 引脚被驱动为低电平后（单脉冲的下降沿）再过 2 个指令时钟，相应通道的中断标志位 $OCxIF$ 被置 1。

图 16-11 和图 16-12 所示为双比较匹配模式产生单输出脉冲的过程。图 16-13 所示为其他时序示例，其中 $OCxRS > PRy$ 。在此示例中，不产生脉冲的下降沿，因为时基在计数达到 $OCxRS$ 值之前就复位了。

图 16-11： 双比较匹配模式：连续输出脉冲模式（ $PRy = OCxRS$ ）(1,2)

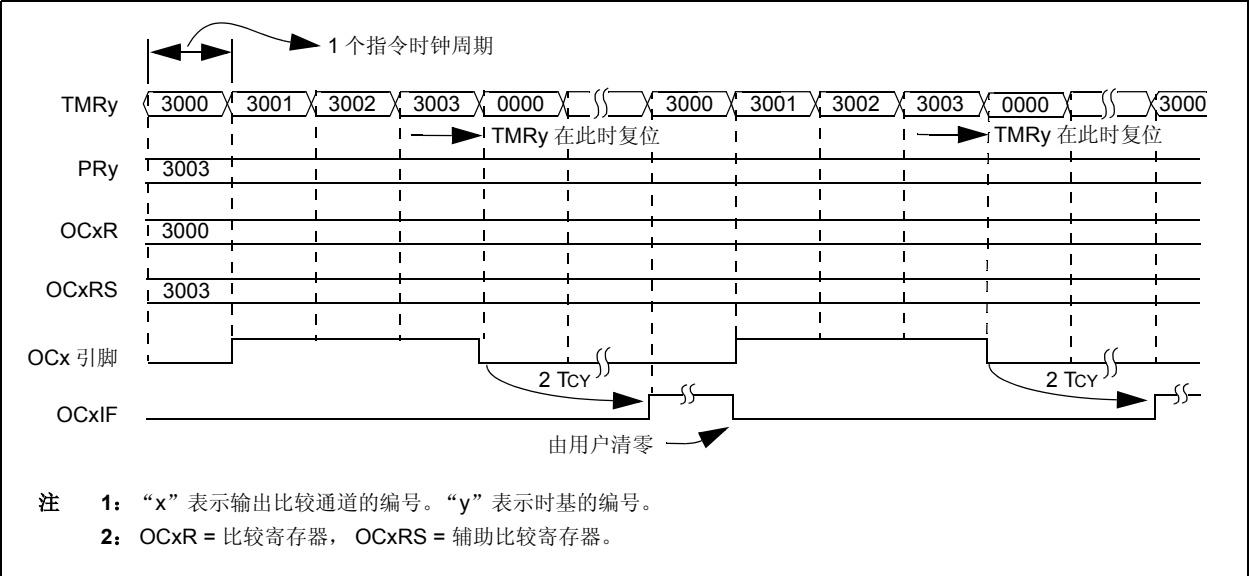


图 16-12: 双比较匹配模式：连续输出脉冲模式 (1,2)

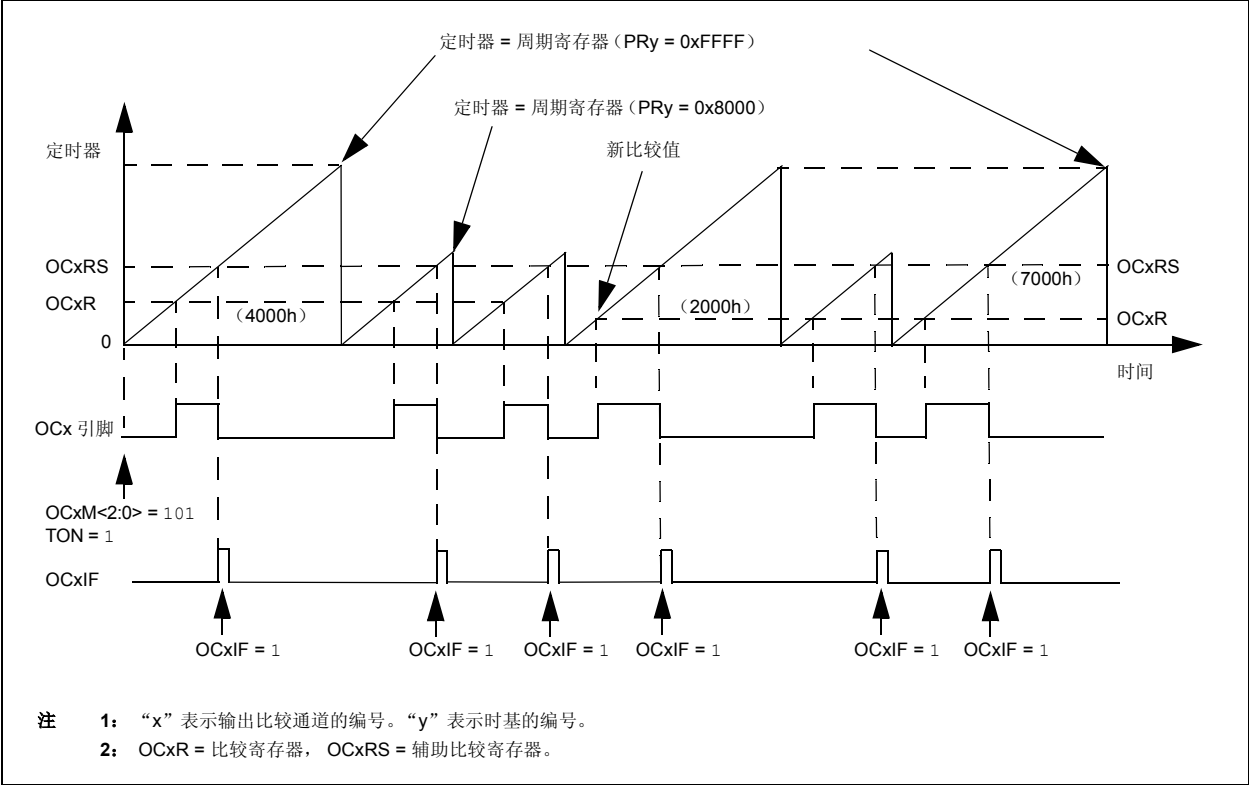
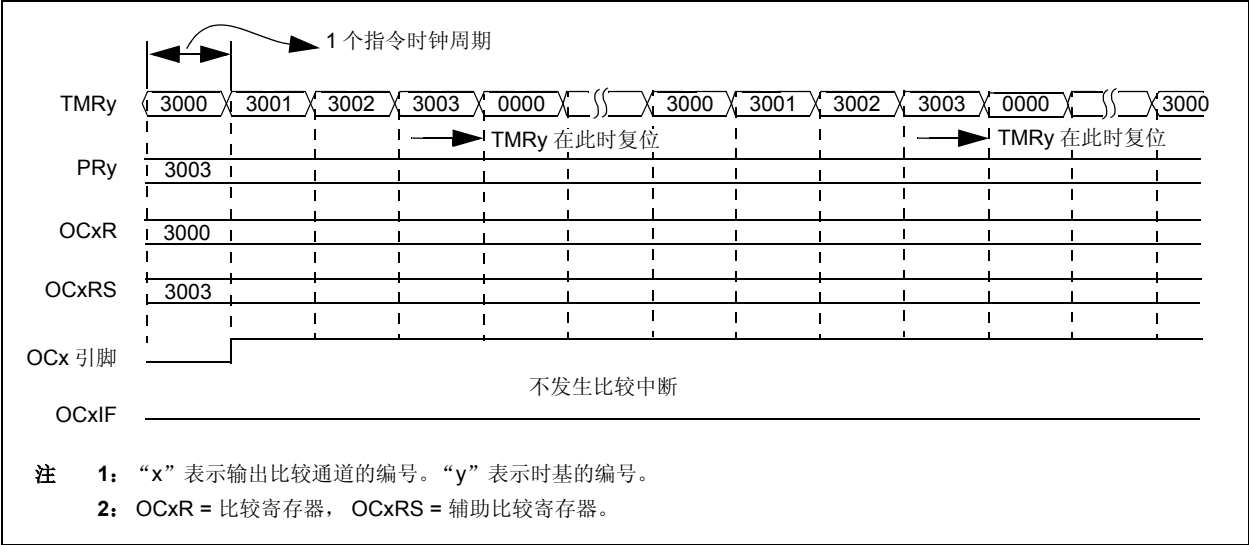


图 16-13: 双比较匹配模式：连续输出脉冲模式 (PRy < OCxRS) (1,2)



16.3.2.5 设置产生连续输出脉冲

当控制位 $OCxM<2:0>$ ($OCxCON<2:0>$) 被设置为 101 时, 所选的输出比较通道将 OCx 引脚初始化为低电平状态, 并在每次发生比较匹配事件时产生输出脉冲。

如果用户要将模块配置为产生连续的输出脉冲流, 则需要执行以下步骤 (这些步骤假设定时器源起初是关闭的, 但这并不是对模块操作的要求):

1. 确定指令时钟周期时间。将定时器源的外部时钟频率 (如果使用了) 和定时器预分频比设置考虑进去。
2. 计算相对于 $TMRy$ 起始值 (0000h) 的到达输出脉冲下降沿的时间。
3. 根据所需的脉冲宽度和脉冲上升沿时间计算到达脉冲下降沿的时间。
4. 将以上步骤2和步骤3中计算出的值分别写入比较寄存器 $OCxR$ 和辅助比较寄存器 $OCxRS$ 。
5. 将定时器周期寄存器 PRy 的值设置为等于或大于辅助比较寄存器 $OCxRS$ 中的值。
6. 设置 $OCM<2:0> = 101$, 并将 $OCTSEL$ ($OCxCON<3>$) 设置为所需定时器源的对应值。此时 OCx 引脚状态被驱动为低电平。
7. 通过设置 TON ($TyCON<15>$) 位为 1 而使能 $TMRy$ 。
8. 在 $TMRy$ 和 $OCxR$ 第一次匹配时, OCx 引脚将被驱动为高电平。
9. 当比较时基 $TMRy$ 和辅助比较寄存器 $OCxRS$ 发生匹配时, 在 OCx 引脚上驱动脉冲的第二个边沿 (即后沿, 从高到低)。
10. 发生第二个比较匹配事件之后, $OCxIF$ 中断标志位会被置 1。
11. 当 $TMRy$ 与其对应的周期寄存器中的值匹配时, $TMRy$ 寄存器复位为 0000h 并重新开始计数。
12. 模块不断重复步骤 8 至步骤 11, 并产生连续的脉冲流。 $OCxIF$ 标志在每次发生 $OCxRS-TMRy$ 比较匹配事件后置 1。

注: 当预分频比为 1:1 时, $OCxR$ 和 $OCxRS$ 之间的最小时间差为 2 个 Tcy 。

例 16-4 所示为连续输出脉冲事件的示例代码。

例 16-4: 连续输出脉冲设置和中断服务

```
// The following code example will set the Output Compare 1 module
// for interrupts on the continuous pulse event and select Timer 2
// as the clock source for the compare time-base. It is assumed
// that Timer 2 and Period Register 2 are properly initialized.
// Timer 2 will be enabled here.

OC1CON          = 0x0000;          // Turn off Output Compare 1 Module
OC1CONbits.OCM  = 0x0005;          // Load new compare mode to OC1CON
OC1R            = 0x3000;          // Initialize Compare Register1 with 0x3000
OC1RS           = 0x3003;          // Initialize Secondary Compare Register1 with 0x3003
IPC0bits.OC1IP0 = 1;              // Setup Output Compare 1 interrupt for
IPC0bits.OC1IP1 = 0;              // desired priority level
IPC0bits.OC1IP2 = 0;              // (this example assigns level 1 priority)
IFS0bits.OC1IF  = 0;              // Clear Output Compare 1 interrupt flag
IEC0bits.OC1IE  = 1;              // Enable Output Compare 1 interrupts
T2CONbits.TON   = 1;              // Start Timer2 with assumed settings

// Example code for Output Compare 1 ISR:
void __attribute__((__interrupt__)) _OC1Interrupt(void)
{
    IFS0bits.OC1IF = 0;
}
```

16.3.2.6 双比较匹配模式产生连续输出脉冲模式的特殊情况

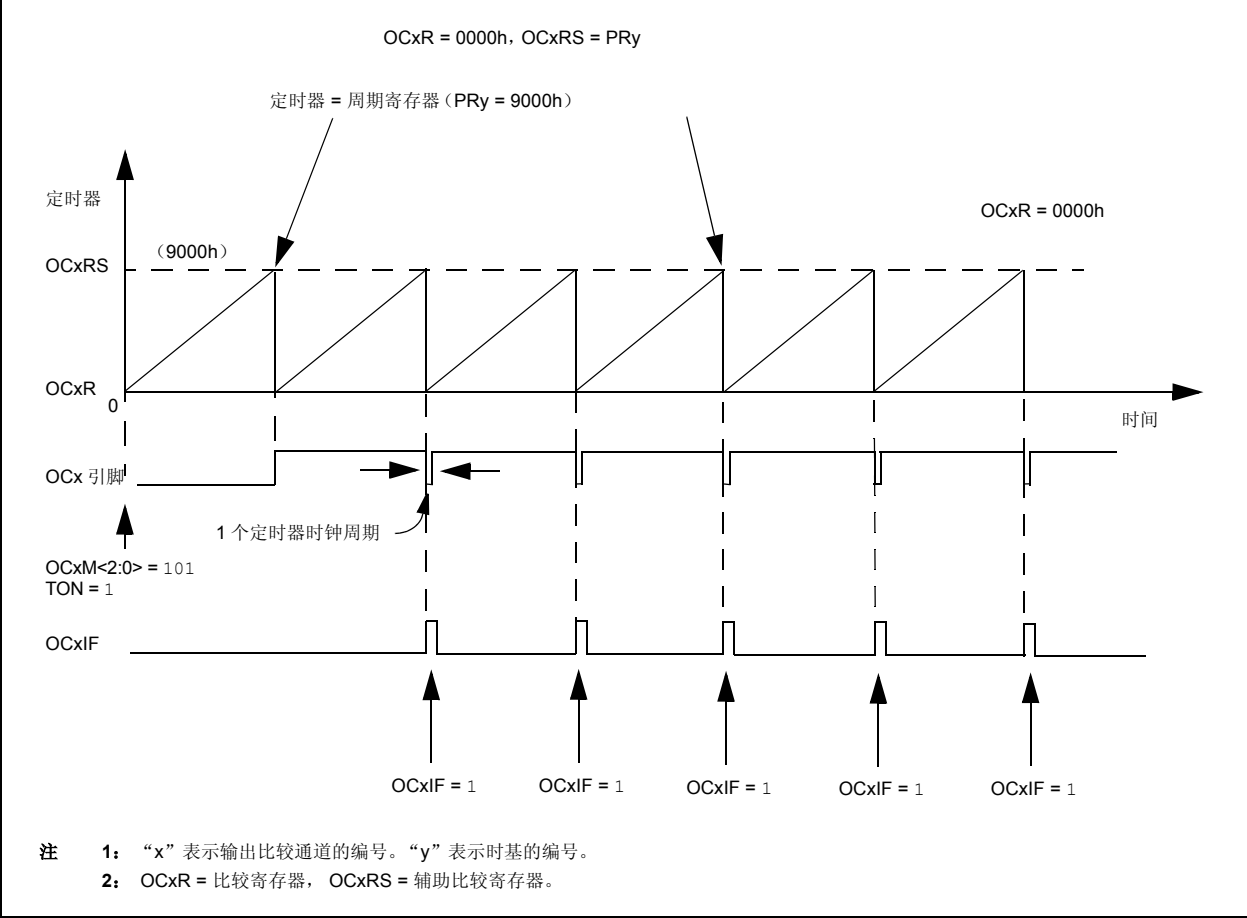
根据 OCxR、OCxRS 和 PRy 值之间的关系，输出比较模块可能不会提供所期望的输出。表 16-2 中说明了这些特殊情况以及这些情况所导致的模块行为。

表 16-2: 双比较匹配模式产生连续输出脉冲模式的特殊情况 (1,2)

SFR 逻辑关系	特殊条件	工作原理	OCx 引脚的输出
$PRy \geq OCxRS$ 且 $OCxRS > OCxR$	$OCxR = 0$, 初始化 $TMRy = 0$	在第一次 $TMRy$ 从 0000h 计数至 PRy 时， OCx 引脚保持为低电平，不产生任何脉冲。在 $TMRy$ 复位为 0（在周期匹配时）之后， OCx 引脚变为高电平。在下次 $TMRy$ 与 $OCxRS$ 匹配时， OCx 引脚变为低电平。如果 $OCxR = 0$ ， $PRy = OCxRS$ ，则引脚将保持一个时钟周期的低电平，然后在下次 $TMRy$ 与 $OCxRS$ 匹配时驱动为高电平。第二次比较之后， $OCxIF$ 位会被置 1。 有两种可选的初始条件需要考虑： a) 初始化 $TMRy = 0$ ，并设置 $OCxR \geq 1$ 。 b) 初始化 $TMRy = PRy$ ($PRy > 0$)，并设置 $OCxR = 0$ （见图 16-14）。	输出连续的脉冲，第一个脉冲根据设置进行延时，延时时间对应于 PRy 寄存器中的值。
$PRy \geq OCxR$ 且 $OCxR \geq OCxRS$	$OCxR \geq 1$ 且 $PRy \geq 1$	$TMRy$ 计数至 $OCxR$ 的值，并在发生比较匹配事件时（即， $TMRy = OCxR$ ）， OCx 引脚驱动为高电平状态。然后， $TMRy$ 继续计数，最终在周期匹配时（即， $PRy = TMRy$ ）复位。然后，定时器重新从 0000h 开始计数至 $OCxRS$ 的值，并在发生比较匹配事件时（即， $TMRy = OCxR$ ）， OCx 引脚驱动为低电平状态。第二次比较之后， $OCxIF$ 位会被置 1。	连续脉冲。
$OCxRS > PRy$ 且 $PRy \geq OCxR$	无	在 OCx 引脚将仅产生一次电平变换，直到 $OCxRS$ 寄存器的值更改为小于等于周期寄存器的值（ PRy ）为止。 $OCxIF$ 直到此时才会被置 1。	上升沿 / 转变为高电平。
$OCxR = OCxRS = PRy = 0000h$	无	输出初始化为低电平，并保持为低电平。 $OCxIF$ 位将不会被置 1。	保持低电平。
$OCxR > PRy$	无	不受支持的模式，定时器在匹配条件之前复位。	保持低电平。

注 1: 本表所述的所有情形下，均假设 $TMRy$ 寄存器被初始化为 0000h。
2: $OCxR$ = 比较寄存器， $OCxRS$ = 辅助比较寄存器， $TMRy$ = 定时器计数器， PRy = 定时器周期寄存器。

图 16-14: 双比较匹配模式: 连续输出脉冲模式 (OCxR = 0x0000, OCxRS = PRy) (1,2)



16.3.3 简单脉宽调制模式

当控制位 OCM<2:0> (OCxCON<2:0>) 被设置为 110 或 111 时, 所选的输出比较通道被配置为简单 PWM (脉宽调制) 工作模式。

有以下两种 PWM 模式可以使用:

- 不带故障保护输入的 PWM
- 带故障保护输入的 PWM

第二种 PWM 模式需使用 OCFA 或 OCFB 故障输入引脚。在该模式下, OCFx 引脚上的异步逻辑电平 0 会使选定的 PWM 通道被关闭 (第 16.3.3.1 节 “使用故障保护输入引脚的 PWM” 中对此进行了描述)。

在 PWM 模式下, OCxR 寄存器是只读从动占空比寄存器, OCxRS 是缓冲寄存器, 由用户写入数据来更新 PWM 占空比。在每次定时器与周期寄存器匹配事件 (PWM 周期结束) 时:

1. TMRy 复位为 0 并重新开始计数。
2. 除非 OCxRS = 0, 否则 OCx 被置 1。
3. 占空比从 OCxRS 传送到 OxCr。
4. TyIF 在 TMRy 与 OCxR 匹配时置 1, OCx 驱动为低电平。

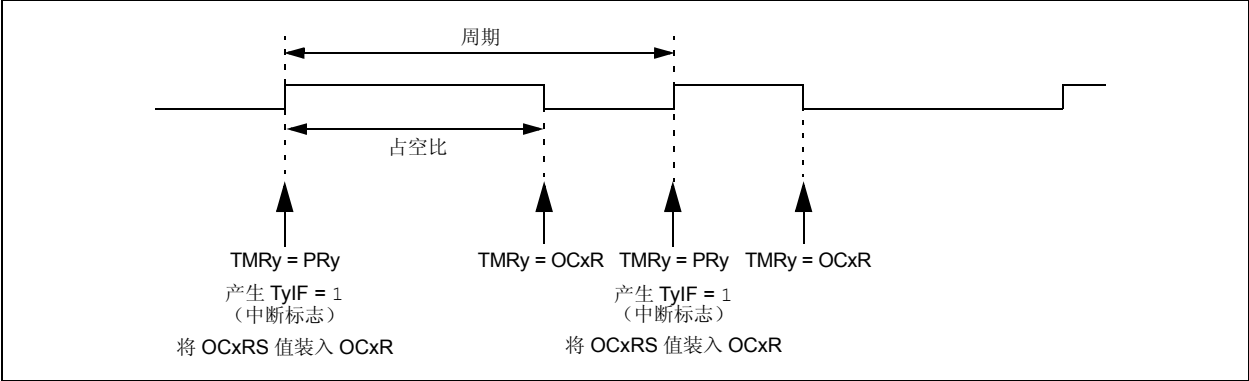
如果要将输出比较模块配置成工作于 PWM 模式, 可采用以下步骤:

1. 通过写选定的定时器周期寄存器 (PRy) 设置 PWM 周期。
2. 通过写 OCxRS 寄存器设置 PWM 占空比。
3. 向 OCxR 寄存器中写入初始占空比。
4. 根据需要允许定时器和输出比较模块的中断。如果要使用 PWM 故障引脚, 则必须设置输出比较中断。
5. 通过写输出比较模式位 OCM<2:0> (OCxCON<2:0>), 将输出比较模块配置为两种 PWM 工作模式中的一种。
6. 设置 TMRy 预分频值, 并通过设置 TON (TxCON<15>) = 1 使能时基。

注: 在第一次使能输出比较模块之前, 必须先初始化 OCxR 寄存器。当模块工作于 PWM 模式时, OCxR 寄存器变为只读的占空比寄存器。OCxR 中保存的值将作为第一个 PWM 周期的 PWM 占空比。占空比缓冲寄存器 OCxRS 中的值在发生时基周期匹配之后才会被传送到 OCxR。

图 16-15 中给出了 PWM 输出波形的示例。

图 16-15: PWM 输出波形



16.3.3.1 使用故障保护输入引脚的 PWM

当输出比较模式位 $OCM<2:0>$ ($OCxCON<2:0>$) 设置为 111 时, 所选的输出比较通道配置为 PWM 工作模式。此时通道具有第 16.3.3 节“简单脉宽调制模式”中描述的所有功能, 同时还具有故障保护输入。

故障保护通过 OCFA 和 OCFB 引脚提供。OCFA 引脚与输出比较通道 1 至 4 关联, 而 OCFB 引脚与输出比较通道 5 关联。

如果在 OCFA/OCFB 引脚检测到逻辑 0, 则所选的 PWM 输出引脚被置为高阻抗状态。用户可以选择在 PWM 引脚连接下拉或上拉电阻, 以便在发生故障条件时提供所需的状态。PWM 输出立即关闭, 关闭操作不与器件时钟源相连。该状态将一直保持, 直到:

- 外部故障条件已经消除, 并且
- 通过写相应的模式位 $OCM<2:0>$ ($OCxCON<2:0>$) 重新使能 PWM 模式。

发生故障条件后, 相应的中断标志位 $OCxIF$ 被置 1, 在允许中断的情况下将产生中断。在检测到故障条件时, $OCFLT$ 位 ($OCxCON<4>$) 驱动为高电平 (逻辑 1)。该位是只读位, 只有在以下情况下才会清零: 外部故障条件已经消除, 并通过写相应的模式位 $OCM<2:0>$ ($OCxCON<2:0>$) 而重新使能 PWM 模式。

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 OCx 输出引脚。

16.3.3.2 PWM 周期

PWM 周期可通过写入 PRy ($TMRy$ 周期寄存器) 来指定。PWM 周期可由以下公式计算:

公式 16-1: 计算 PWM 周期⁽¹⁾

$$\text{PWM 周期} = [(PRy) + 1] \cdot T_{CY} \cdot (TMRy \text{ 预分频值})$$

$$\text{PWM 频率} = 1 / [\text{PWM 周期}]$$

注 1: 基于 $T_{CY} = 2/F_{OSC}$, 打盹模式和 PLL 被禁止。

注: 如果 PRy 值为 N , 则产生的 PWM 周期为 $N + 1$ 个时基计数周期。例如, 如果写入 PRy 寄存器的值为 7, 则将产生包含 8 个时基周期的周期。

16.3.3.3 PWM 占空比

PWM 占空比通过写 $OCxRS$ 寄存器进行指定。可以在任何时候写 $OCxRS$ 寄存器, 但是在 PRy 和 $TMRy$ 发生匹配 (即周期结束) 前占空比值不会被锁存到 $OCxR$ 中。这为 PWM 占空比提供了双缓冲, 对于避免在 PWM 操作中产生毛刺非常重要。在 PWM 模式下, $OCxR$ 是只读寄存器。

PWM 占空比有一些重要的边界参数, 包括:

- 如果占空比寄存器 $OCxR$ 中装入 0000h, 则 OCx 引脚将保持低电平 (占空比为 0%)。
- 如果 $OCxR$ 大于 PRy (定时器周期寄存器), 则引脚将保持高电平 (占空比为 100%)。
- 如果 $OCxR$ 等于 PRy , 则 OCx 引脚对于一个时基计数值保持低电平, 而对于另一个计数值保持高电平。

请参见图 16-16 了解 PWM 模式时序的详细信息。表 16-3 和表 16-4 给出了器件分别以 4 MIPS 和 16 MIPS 工作时, 所对应的 PWM 频率和分辨率的示例。

公式 16-2: 计算最大 PWM 分辨率⁽¹⁾

$$\text{最大 PWM 分辨率 (位)} = \frac{\log_{10}\left(\frac{F_{CY}}{F_{PWM} \cdot (\text{定时器预分频值})}\right)}{\log_{10}(2)} \text{ 位}$$

注 1: 基于 $T_{CY} = 2/F_{OSC}$, 打盹模式和 PLL 被禁止。

例 16-5: PWM 周期和占空比计算

1. 确定对应于所需的 PWM 频率 52.08 kHz 的周期寄存器值, 其中
 $F_{OSC} = 8 \text{ MHz}$, PLL (32 MHz 器件时钟速率) 和 Timer2 预分频比设置为 1:1。
 $T_{CY} = 2/F_{OSC} = 62.5 \text{ ns}$
PWM 周期 = $1/\text{PWM 频率} = 1/52.08 \text{ kHz} = 19.2\mu\text{s}$
PWM 周期 = $(PR2 + 1) \cdot T_{CY} \cdot (\text{Timer2 预分频值})$
 $19.2\mu\text{s} = (PR2 + 1) \cdot 62.5 \text{ ns} \cdot 1$
 $PR2 = 306$

图 16-16: PWM 输出时序^(1,2)

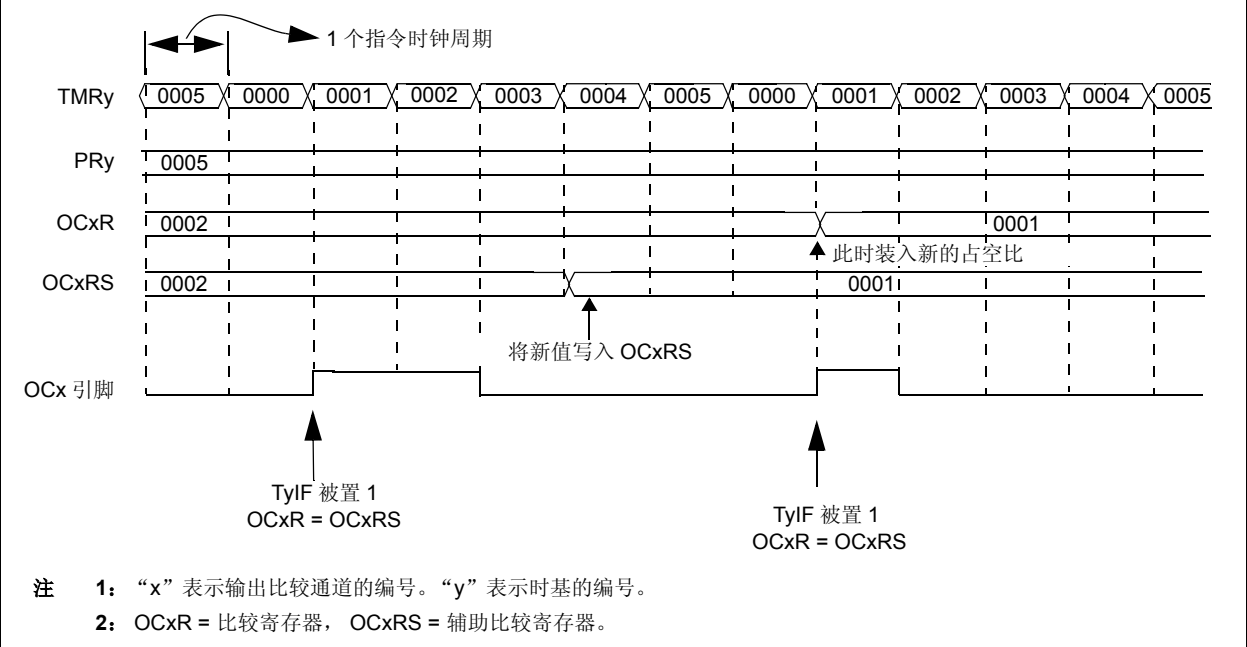


表 16-3: 4 MIPS 时的 PWM 频率和分辨率示例 ($F_{CY} = 4 \text{ MHz}$)⁽¹⁾

PWM 频率	7.6 Hz	61 Hz	122 Hz	977 Hz	3.9 kHz	31.3 kHz	125 kHz
定时器预分频比	8	1	1	1	1	1	1
周期寄存器值	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
分辨率 (位)	16	16	15	12	10	7	5

注 1: 基于 $T_{CY} = 2/F_{OSC}$, 打盹模式和 PLL 被禁止。

表 16-4: 16 MIPS 时的 PWM 频率和分辨率示例 ($F_{CY} = 16 \text{ MHz}$)⁽¹⁾

PWM 频率	30.5 Hz	244 Hz	488 Hz	3.9 kHz	15.6 kHz	125 kHz	500 kHz
定时器预分频比	8	1	1	1	1	1	1
周期寄存器值	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
分辨率 (位)	16	16	15	12	10	7	5

注 1: 基于 $T_{CY} = 2/F_{OSC}$, 打盹模式和 PLL 被禁止。

16.3.3.4 简单 PWM 模式的初始化

- 1. 使能简单 PWM 模式 ($OCxM<2:0> = 110$ 或 111) 之后, 如果 $OCxR = 0000h$, 引脚将被驱动为低电平状态。如果 $OCxR$ 不等于 0, 则引脚将被设置为高电平状态。在一些时序点应该使能定时器, 以便正确执行操作 (见图 16-17 和图 16-18)。
- 2. 如果 $OCxR$ 不等于 0, 引脚设置为高电平状态, 则在占空比和定时器之间第一次匹配时, 引脚驱动为低电平。引脚将保持低电平, 直到定时器和周期寄存器之间发生有效的比较 (见图 16-18)。

图 16-17: 简单 PWM 模式: 初始化为低电平 (1,2)

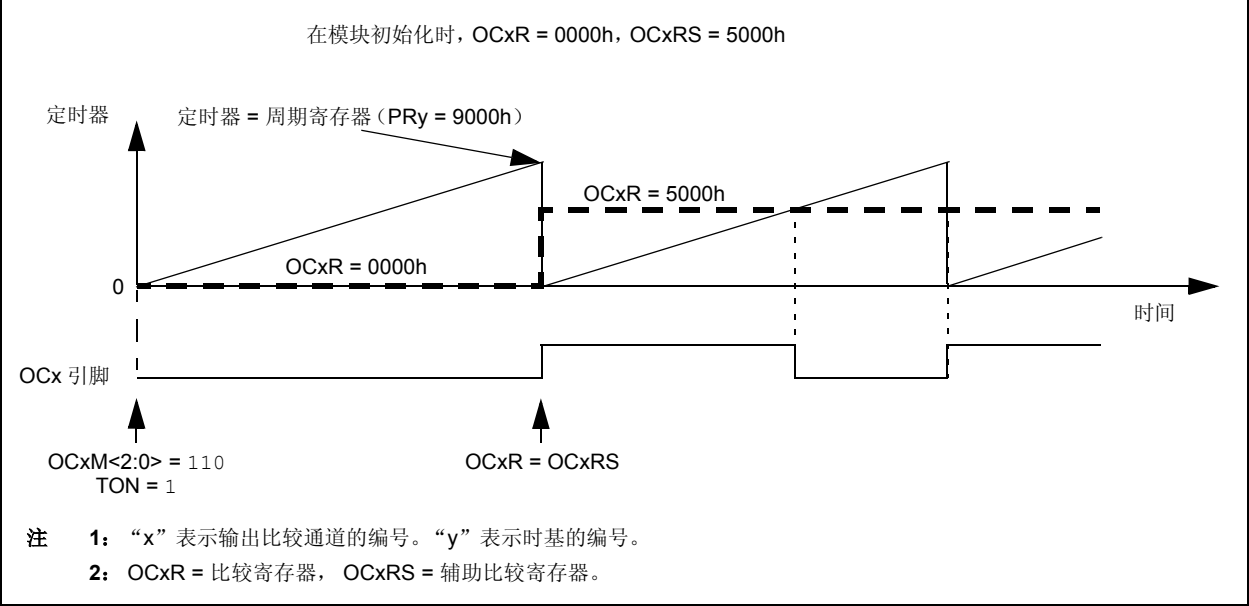
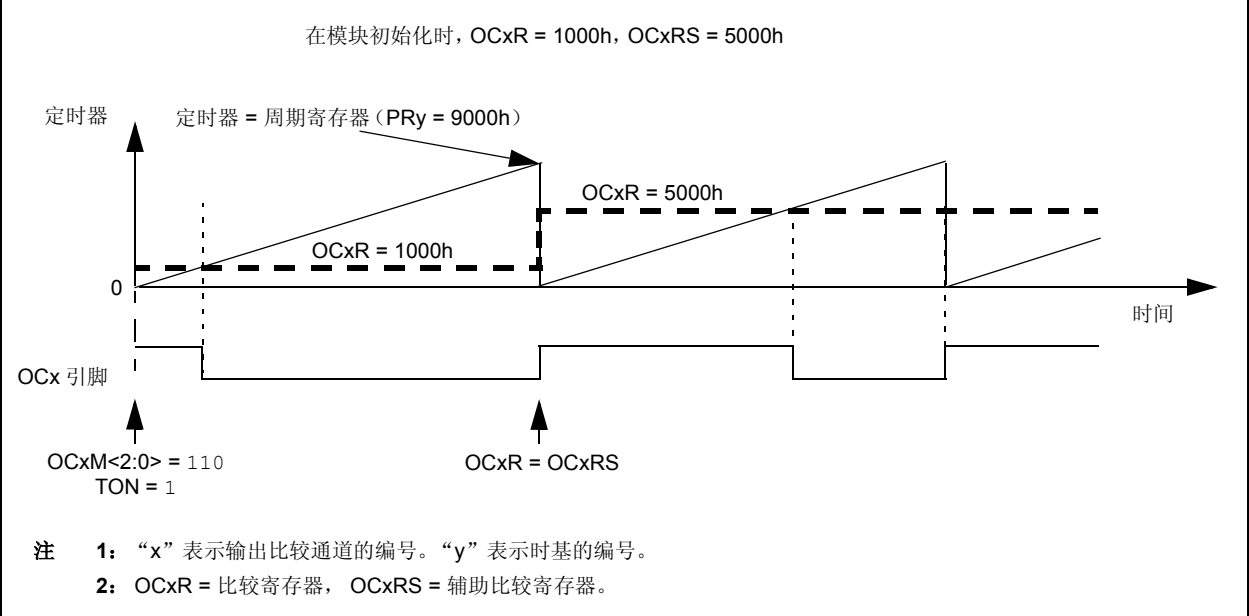


图 16-18: 简单 PWM 模式: 初始化为高电平 (1,2)



例 16-6 给出了 PWM 工作模式的配置和中断服务代码。

例 16-6: 简单 PWM 模式：脉冲设置和中断服务

```
// The following code example will set the Output Compare 1 module
// for PWM mode w/o FAULT pin enabled, a 50% duty cycle and a
// PWM frequency of 52.08 kHz at Fosc = 8 MHz.Timer 2 is selected as
// the clock for the PWM time base and Timer2 interrupts
// are enabled.

OC1CON      = 0x0000;          // Turn off Output Compare 1 Module
OC1R        = 0x0026;          // Initialize Compare Register1 with 0x0026
OC1RS        = 0x0026;          // Initialize Secondary Compare Register1 with 0x0026
OC1CON      = 0x0006;          // Load new compare mode to OC1CON
PR2         = 0x004C;          // Initialize PR2 with 0x004C
IPC1bits.T2IP = 1;             // Setup Output Compare 1 interrupt for
IFS0bits.T2IF = 0;             // Clear Output Compare 1 interrupt flag
IEC0bits.T2IE = 1;             // Enable Output Compare 1 interrupts
T2CONbits.TON = 1;             // Start Timer2 with assumed settings

// Example code for Timer2 ISR:
void __attribute__((__interrupt__)) _T2Interrupt(void)
{
    IFS0bits.T2IF = 0;
}
```

16.3.3.5 简单 PWM 模式特殊比较条件

- 1. 如果 OCxR 和 PWM 周期寄存器等于 0000h，则引脚将设置为低电平。
- 2. 如果 OCxR 等于 0，而 PWM 周期寄存器不等于 0，则引脚将设置为低电平（见图 16-19）。
- 3. 如果 OCxR 大于 PWM 周期寄存器，则引脚将保持高电平（见图 16-20）。
- 4. 如果两者（OCxR 和 PRy）都等于非零值，则输出引脚将在不超出 1 个定时器时钟周期的时间内设置为低电平，然后立即设置为高电平（见图 16-21）。

图 16-19: PWM 输出时序（0% 占空比，OCxR = 0000h）^(1,2)

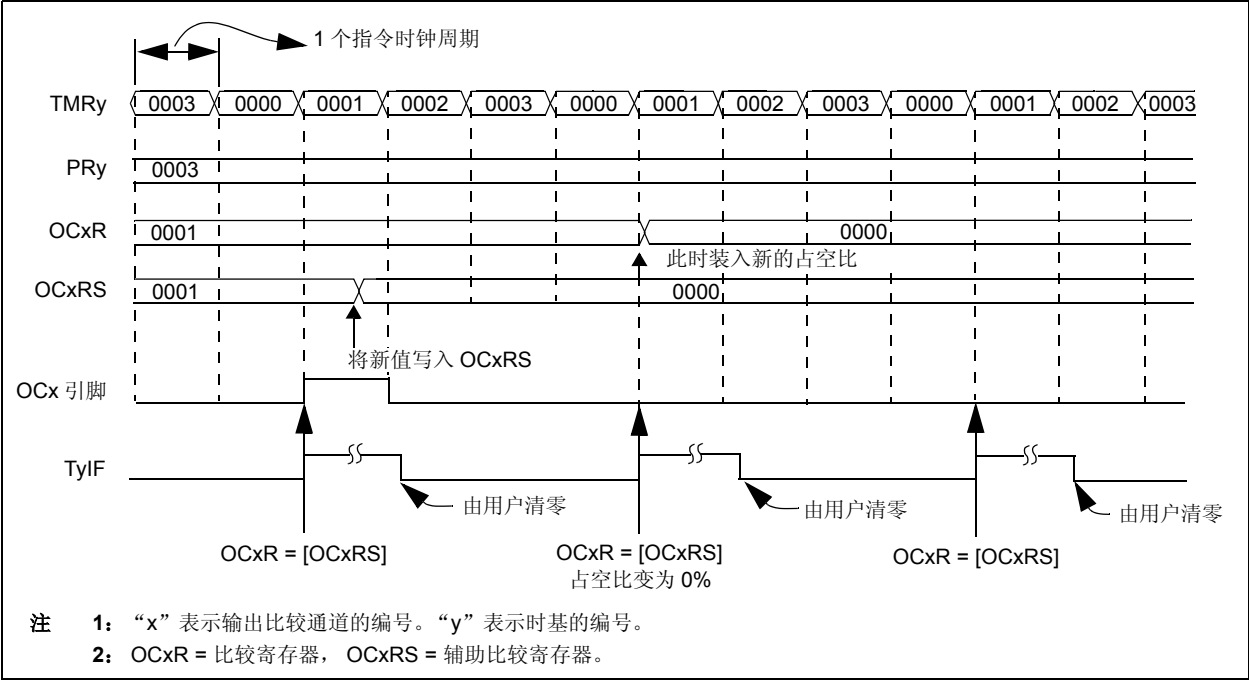


图 16-20: PWM 输出时序 (100% 占空比, $OCxR > PRy$) (1,2)

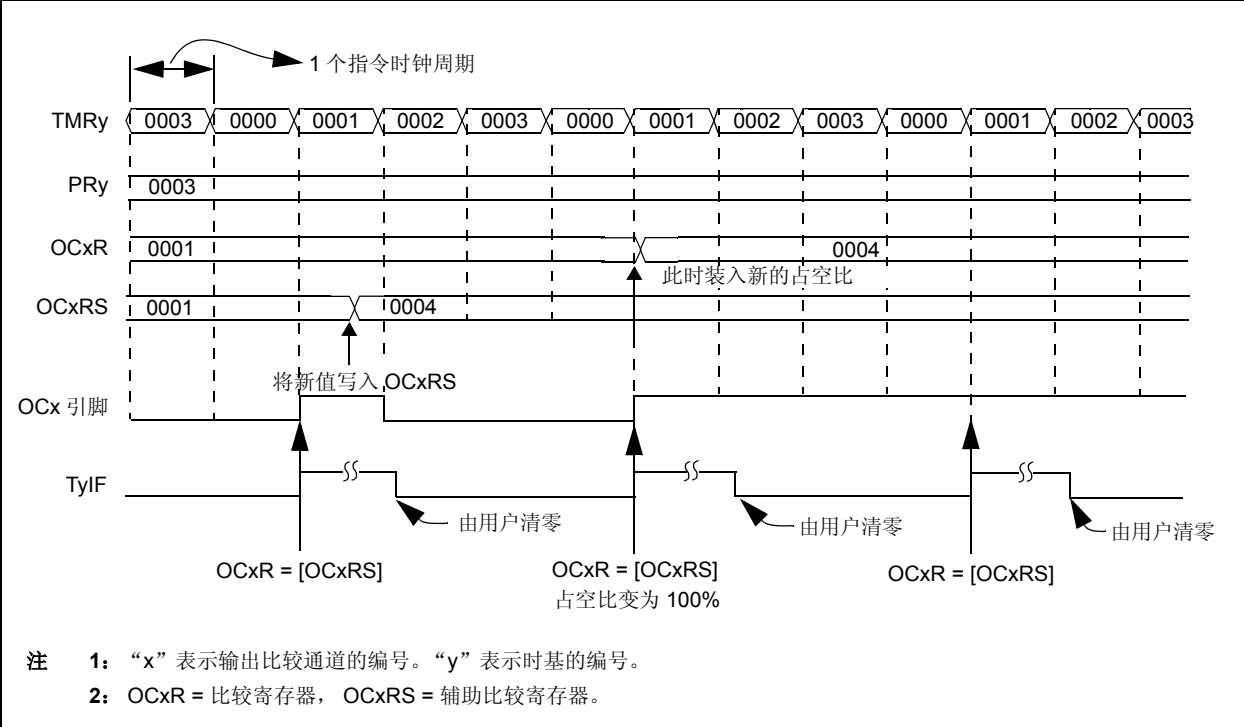
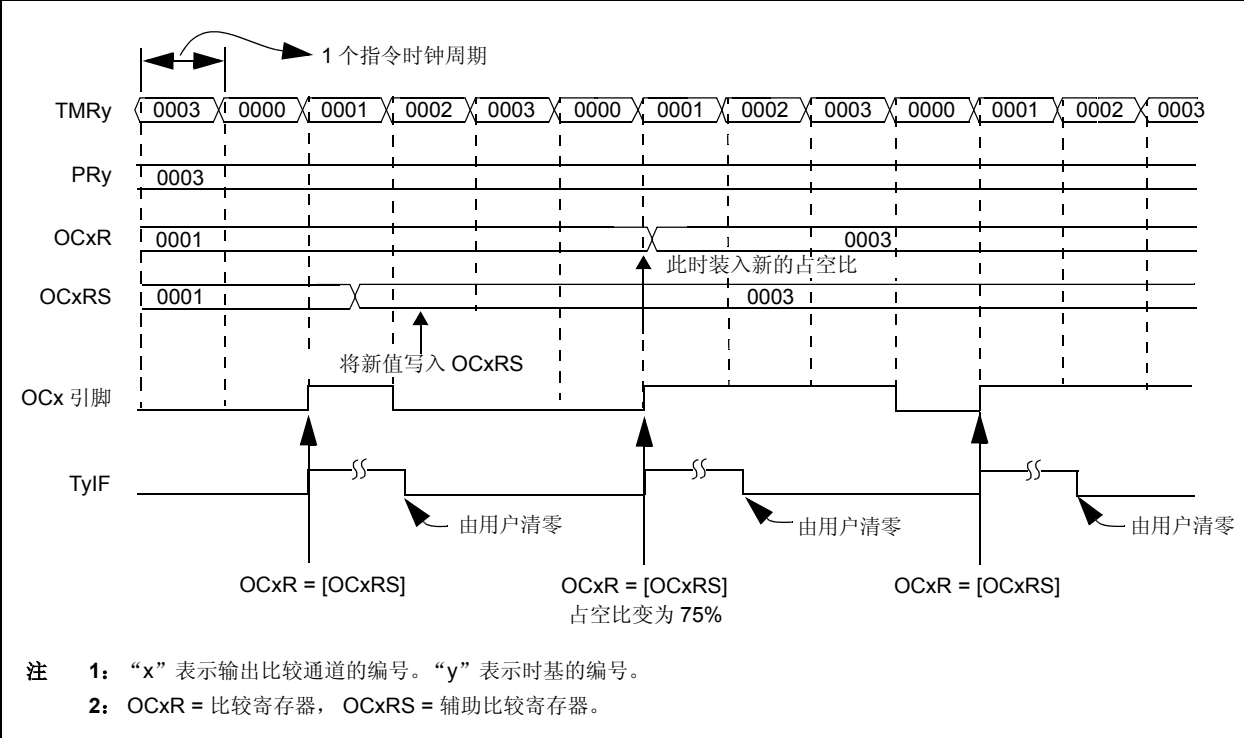


图 16-21: PWM 输出时序 ($OCxR = PRy$) (1,2)



16.4 省电状态下的输出比较操作

16.4.1 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模式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，引脚保持高电平。同理，如果引脚原先为低电平，则在 CPU 进入休眠状态后，引脚保持低电平。在两种情况下，在器件唤醒时，输出比较模块均将继续工作。

16.4.2 PWM 故障模式时休眠

当模块处于 PWM 故障模式时，故障保护电路的异步部分将保持工作状态。如果检测到故障，OCx 引脚将为三态。OCFLT 位将被置 1。在发生故障时将不会产生中断，但中断将被排入队列，在器件唤醒时发生中断。

16.4.3 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 位（OCxCON<13>）用于选择输出捕捉模块在空闲模式是停止还是继续工作。

- 如果 OCSIDL = 1，则在空闲模式下模块将停止工作。在空闲模式下停止时（OCSIDL = 1），模块将执行与休眠模式相同的过程。
- 如果 OCSIDL = 0，则只有当所选时基设置为在空闲模式下工作时，模块才会在空闲模式下继续工作。如果 OCSIDL 位为逻辑 0，则输出比较通道将在 CPU 空闲模式期间工作。并且，还必须将相应的 TSIDL 位设为逻辑 0，以使能时基。

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。

16.4.4 打盹模式

打盹模式下的输出比较操作和正常模式下相同。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。

更多详细信息，请参见第 10 章“省电特性”。

16.4.5 有选择地进行外设模块控制

外设模块禁止（PMD）寄存器提供了一种禁止输出比较模块的方法，它通过停止所有向该模块供应的时钟源来达到目的。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。与模块相关的控制和状态寄存器也被禁止，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。

更多详细信息，请参见第 10 章“省电特性”。

16.5 I/O 引脚控制

当使能了输出比较模块时，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 引脚的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 OCFx 故障引脚配置为输入。使能此特殊 PWM 模式并不会将 OCFx 故障引脚配置为输入。

表 16-5: 与输出比较模块 1-5 相关的引脚

引脚名称	引脚类型	说明
OC1	O	输出比较 /PWM 通道 1
OC2	O	输出比较 /PWM 通道 2
OC3	O	输出比较 /PWM 通道 3
OC4	O	输出比较 /PWM 通道 4
OC5	O	输出比较 /PWM 通道 5
OCFA	I	PWM 故障保护 A 输入（用于通道 1-4）
OCFB	I	PWM 故障保护 B 输入（用于通道 5）

图注： I = 输入， O = 输出

16.6 寄存器映射

表 16-6、表 16-7 和表 16-8 中提供了与 PIC24F 输出比较模块相关的寄存器汇总。

表 16-6: 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
OCxRS	输出比较 x 辅助寄存器																xxxx
OCxR	输出比较 x 寄存器																xxxx
OCxCON			OCSIDL									OCFLT	OCTSEL	OCM2	OCM1	OCM0	0000

图注: x = 复位时的未知值, — = 未实现, 读为 0。所示复位值为十六进制。

表 16-7: 定时器寄存器映射

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
TMR2	Timer2 寄存器																xxxx
TMR3	Timer3 寄存器																xxxx
PR2	周期寄存器 2																FFFF
PR3	周期寄存器 3																FFFF
T2CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T3CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000

图注: x = 复位时的未知值, — = 未实现, 读为 0。所示复位值为十六进制。

表 16-8: 中断控制寄存器映射

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
IFS0	—	—	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPF1IF	T3IF	T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	—	—	—	—	INT1IF	CNIF	CMIF	M2C1IF	SI2C1IF	0000
IFS2	—	—	PMPIF	—	—	—	OC5IF	—	IC5IF	IC4IF	IC3IF	—	—	—	SPI2IF	SPF2IF	0000
IEC1	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	—	—	—	—	INT1IE	CNIE	CMIE	M2C1IE	SI2C1IE	0000
IEC2	—	—	PMPIE	—	—	—	OC5IE	—	IC5IE	IC4IE	IC3IE	—	—	—	SPI2IE	SPF2IE	0000
IPC0	—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0	—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0	4444
IPC1	—	T2IP2	T2IP1	T2IP0	—	OC2IP2	OC2IP1	OC2IP0	—	IC2IP2	IC2IP1	IC2IP0	—	—	—	—	4440
IPC6	—	T4IP2	T4IP1	T4IP0	—	OC4IP2	OC4IP1	OC4IP0	—	OC3IP2	OC3IP1	OC3IP0	—	—	—	—	4440
IPC9	—	IC5IP2	IC5IP1	IC5IP0	—	IC4IP2	IC4IP1	IC4IP0	—	IC3IP2	IC3IP1	IC3IP0	—	—	—	—	4440
IPC10	—	—	—	—	—	—	—	—	—	OC5IP2	OC5IP1	OC5IP0	—	—	—	—	0040

图注: — = 未实现, 读为 0。所示复位值为十六进制。

16.7 电气规范

16.7.1 交流特性

图 16-22: 输出比较时序

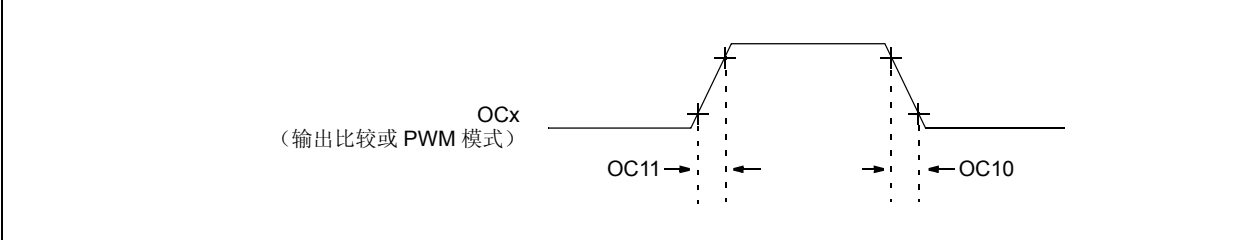


表 16-9: 输出捕捉

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 16-23: PWM 模块时序要求

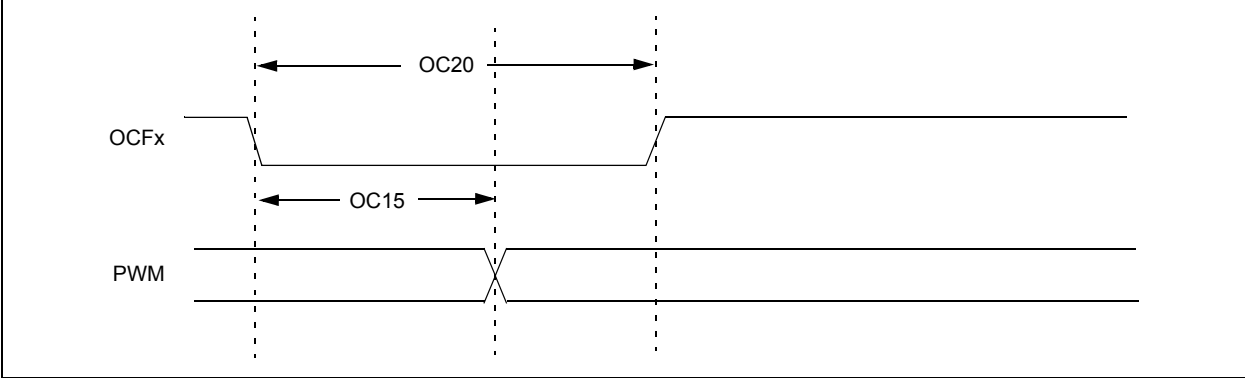


表 16-10: PWM 时序要求

参数编号	符号	特性	最小值	典型值 [†]	最大值	单位	条件
OC15	TFD	故障输出到 PWM I/O 改变	—	—	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

[†] 除非另外声明，否则“典型值”栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

16.8 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: 当相关定时器源的 *TSIDL* 位 (*TxCON*<13>) 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。

问 2: 当所选时基配置为 32 位模式时，是否能使用输出比较模块？

答: 不能。当定时器配合输出比较模块使用时，应该将 *T32* 位 (*TxCON*<3>) 清零。

16.9 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，但其概念是相关的，通过适当修改即可使用，但在使用中可能会受到一定限制。当前与输出比较模块相关的应用笔记有：

标题	应用笔记编号
An I ² C™ Network Protocol for Environmental Monitoring	AN736
Using the CCP Module(s)	AN594
Yet Another Clocking Featuring the PIC16C924	AN649
Using PWM to Generate Analog Output	AN538
Low-Cost Bidirectional Brushed DC Motor Control Using the PIC16F684	AN893
Speed Control of 3-Phase Induction Motor Using PIC18 Microcontrollers	AN843

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)。

16.10 版本历史

版本 A（2006 年 4 月）

这是本文档的初始发行版。

注: