
Section 16. Output Compare

HIGHLIGHTS

This section of the manual contains the following major topics:

16.1	Introduction	16-2
16.2	Output Compare Registers	16-3
16.3	Modes of Operation	16-4
16.4	Output Compare Operation in Power-Saving States	16-27
16.5	I/O Pin Control	16-28
16.6	Register Maps	16-29
16.7	Electrical Specifications	16-30
16.8	Design Tips	16-31
16.9	Related Application Notes.....	16-32
16.10	Revision History	16-33

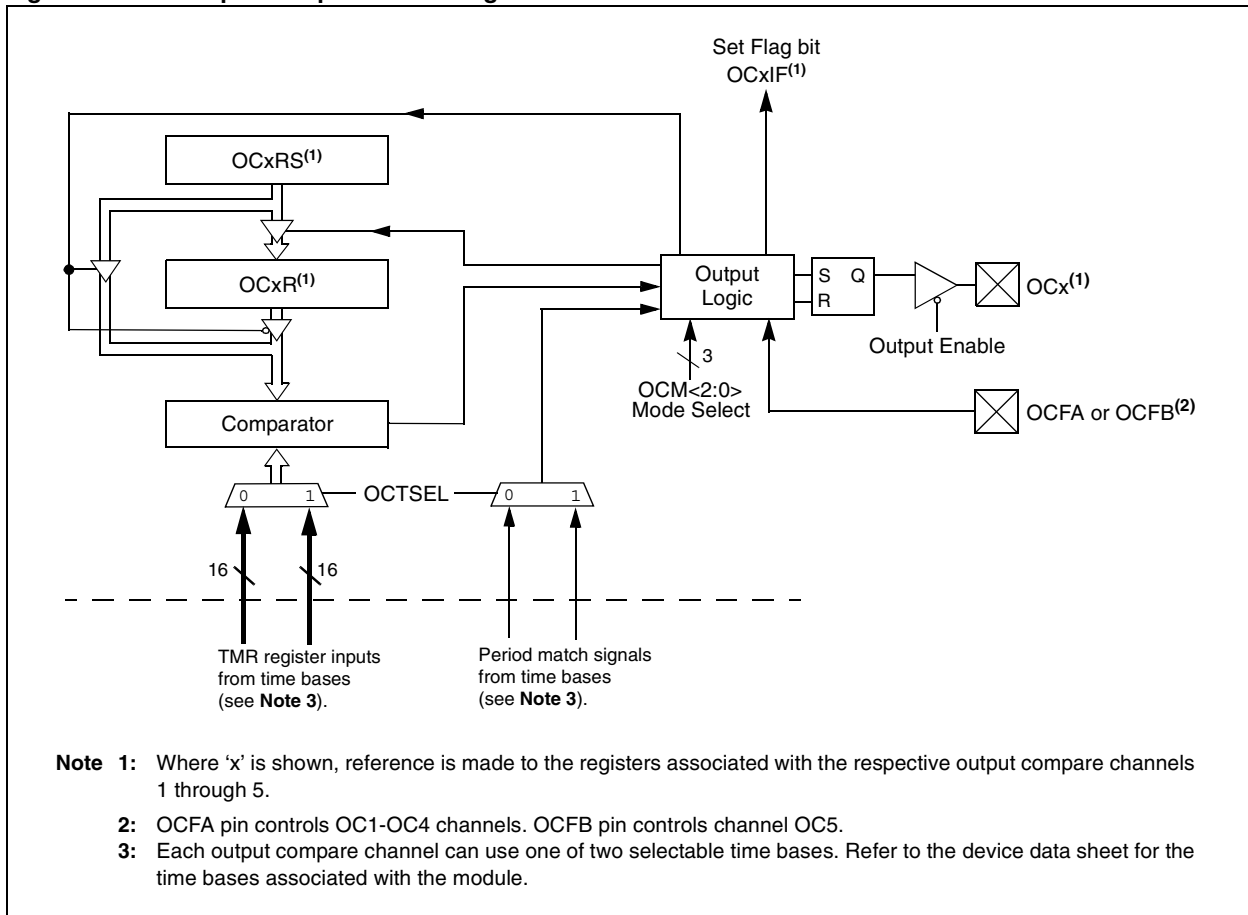
16.1 INTRODUCTION

The output compare module has the ability to compare the value of a selected time base with the value of one or two compare registers (depending on the operation mode selected). Furthermore, it has the ability to generate a single output pulse, or a train of output pulses, on a compare match event. Like most PICmicro® peripherals, it also has the ability to generate interrupts on compare match events.

Refer to the specific device data sheet for the number of channels available in a particular device. All output compare channels are functionally identical. In this section, an 'x' in the pin, register or bit name denotes the specific output compare channel.

Each output compare channel can use one of two selectable time bases. The time base is selected using the OCTSEL bit (OCxCON<3>). Please refer to the device data sheet for the specific timers that can be used with each output compare channel number. The available time bases, Timer2 and Timer3, do not support Asynchronous mode. Therefore, the output compare module will operate only in Synchronous mode.

Figure 16-1: Output Compare Block Diagram



16.2 OUTPUT COMPARE REGISTERS

Each output compare channel has the following registers:

- OCxCON: the control register for the output compare channel
- OCxR: a data register for the output compare channel
- OCxRS: a secondary data register for the output compare channel

The control registers for the 5 output compare channels are named OC1CON through OC5CON. All 5 control registers have identical bit definitions. They are represented by a common register definition below. The 'x' in OCxCON represents the output compare channel number.

Register 16-1: OCxCON: Output Compare x Control Register

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

Legend:	HC = Cleared in Hardware
R = Readable bit	W = Writable bit
-n = Value at POR	'1' = Bit is set
	U = Unimplemented bit, read as '0'
	'0' = Bit is cleared
	x = Bit is unknown

- bit 15-14 **Unimplemented:** Read as '0'
- bit 13 **OCSIDL:** Stop Output Compare x in Idle Mode Control bit
 1 = Output compare x will halt in CPU Idle mode
 0 = Output compare x will continue to operate in CPU Idle mode
- bit 12-5 **Unimplemented:** Read as '0'
- bit 4 **OCFLT:** PWM Fault Condition Status bit
 1 = PWM Fault condition has occurred (cleared in HW only)
 0 = No PWM Fault condition has occurred (this bit is only used when OCM<2:0> = 111)
- bit 3 **OCTSEL:** Output Compare x Timer Select bit⁽¹⁾
 1 = Timer3 is the clock source for Output Compare x
 0 = Timer2 is the clock source for Output Compare x
- bit 2-0 **OCM<2:0>:** Output Compare x Mode Select bits
 111 = PWM mode on OCx, Fault pin enabled
 110 = PWM mode on OCx, Fault pin disabled
 101 = Initialize OCx pin low, generate continuous output pulses on OCx pin
 100 = Initialize OCx pin low, generate single output pulse on OCx pin
 011 = Compare event toggles OCx pin
 010 = Initialize OCx pin high, compare event forces OCx pin low
 001 = Initialize OCx pin low, compare event forces OCx pin high
 000 = Output compare channel is disabled

Note 1: Refer to the device data sheet for specific time bases available to the output compare module.

16.3 MODES OF OPERATION

Each output compare module has the following modes of operation:

- Single Compare Match mode
- Dual Compare Match mode generating:
 - Single Output Pulse mode
 - Continuous Output Pulse mode
- Simple Pulse-Width Modulation mode:
 - with Fault Protection Input
 - without Fault Protection Input

Note 1: It is recommended that the user turn off the output compare module (i.e., clear OCM<2:0> (OCxCON<2:0>)) before switching to a new mode.

2: In this section, a reference to any SFRs associated with the selected timer source is indicated by a 'y' suffix. For example, PRy is the Period register for the selected timer source, while TyCON is the Timer Control register for the selected timer source.

16.3.1 Single Compare Match Mode

When control bits, OCM<2:0> (OCxCON<2:0>), are set to '001', '010' or '011', the selected output compare channel is configured for one of three Single Compare Match modes.

In the Single Compare Match mode, the OCxR register is loaded with a value and is compared to the selected incrementing timer register, TMRy. On a compare match event, one of the following events will take place:

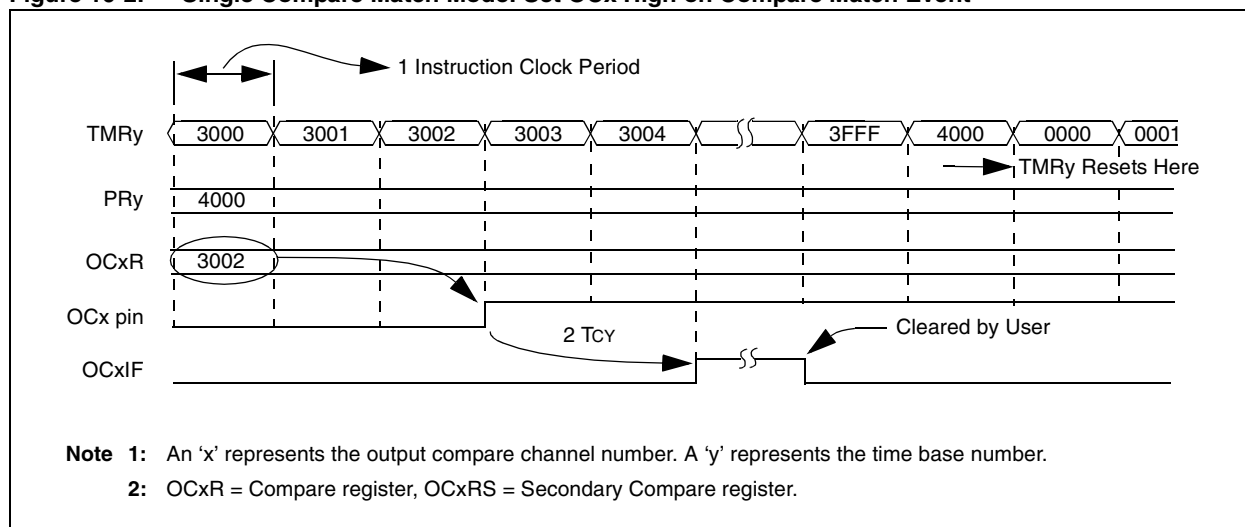
- Compare forces OCx pin high, initial state of pin is low. Interrupt is generated on the single compare match event.
- Compare forces OCx pin low, initial state of pin is high. Interrupt is generated on the single compare match event.
- Compare toggles OCx pin. Toggle event is continuous and an interrupt is generated for each toggle event.

16.3.1.1 SINGLE COMPARE MATCH MODE OUTPUT DRIVEN HIGH

To configure the output compare module for this mode, set control bits $OCM\langle 2:0 \rangle = 001$. The TMRy should also be enabled. Once this Compare mode has been enabled, the output pin, OCx, will be initially driven low and remain low until a match occurs between the TMRy and OCxR registers. Referring to Figure 16-2, there are some key timing events to note:

- The OCx pin is driven high one instruction clock after the compare match occurs between the TMRy and the OCxR register. The OCx pin will remain high until a mode change has been made or the module is disabled.
- The TMRy will count up to the value contained in the associated Period register and then reset to 0000h on the next instruction clock.
- The respective Channel Interrupt Flag, OCxIF, is asserted two instruction clocks after the OCx pin is driven high.

Figure 16-2: Single Compare Match Mode: Set OCx High on Compare Match Event^(1,2)

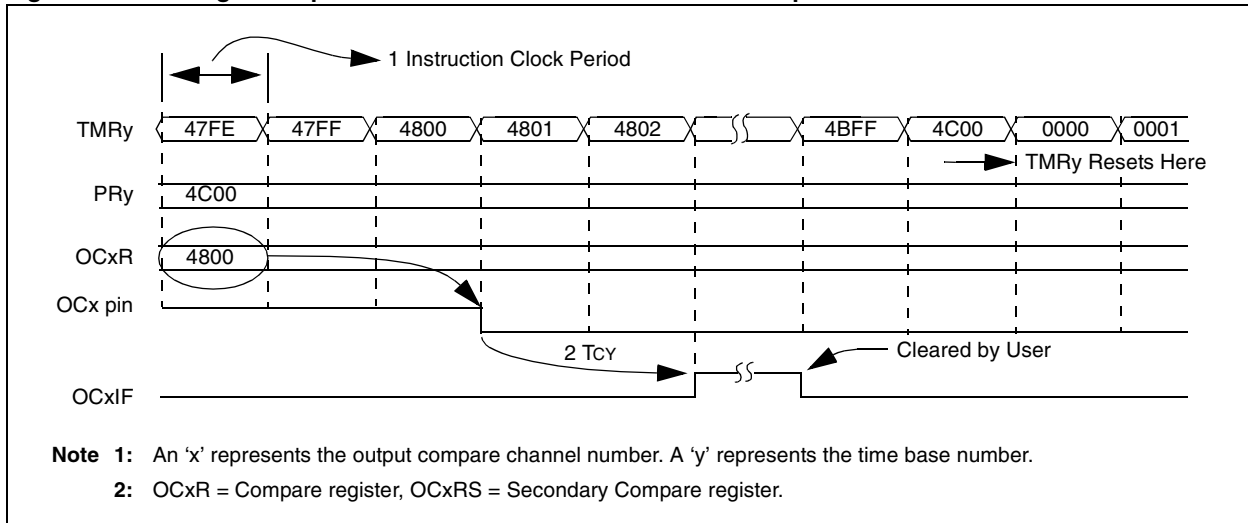


16.3.1.2 SINGLE COMPARE MATCH MODE OUTPUT DRIVEN LOW

To configure the output compare module for this mode, set control bits $OCM<2:0> = 010$. TMRy must also be enabled. Once this Compare mode has been enabled, the output pin, OCx, will be initially driven high and remain high until a match occurs between the Timer and OCxR registers. Referring to Figure 16-3, there are some key timing events to note:

- The OCx pin is driven low one instruction clock after the compare match occurs between the TMRy and the OCxR register. The OCx pin will remain low until a mode change has been made or the module is disabled.
- The TMRy will count up to the value contained in the associated Period register and then reset to 0000h on the next instruction clock.
- The respective Channel Interrupt Flag, OCxIF, is asserted two instruction clocks after OCx pin is driven low.

Figure 16-3: Single Compare Match Mode: Force OCx Low on Compare Match Event^(1,2)



16.3.1.3 SINGLE COMPARE MATCH MODE TOGGLE OUTPUT

To configure the output compare module for this mode, set control bits $OCM<2:0> = 011$. TMRy must also be enabled. Once this Compare mode has been enabled, the output pin, OCx, will be initially driven low and then toggled on each and every subsequent match event between the Timer and OCxR registers. Referring to Figure 16-4 and Figure 16-5, there are some key timing events to note:

- The OCx pin is toggled one instruction clock after the compare match occurs between the TMRy and the OCxR register. The OCx pin will remain at this new state until the next toggle event, or until a mode change has been made, or the module is disabled.
- The TMRy will count up to the contents in the period register and then reset to 0000h on the next instruction clock.
- The respective channel interrupt flag, OCxIF, is asserted two instruction clocks after the OCx pin is toggled.

Note: The internal OCx pin output logic is set to a logic '0' on a device Reset. However, the operational OCx pin state for the Toggle mode can be set by the user software. Example 16-1 shows a code example for defining the desired initial OCx pin state in the Toggle mode of operation.

Figure 16-4: Single Compare Match Mode: Toggle Output on Compare Match Event ($PRy > OCxR$)^(1,2)

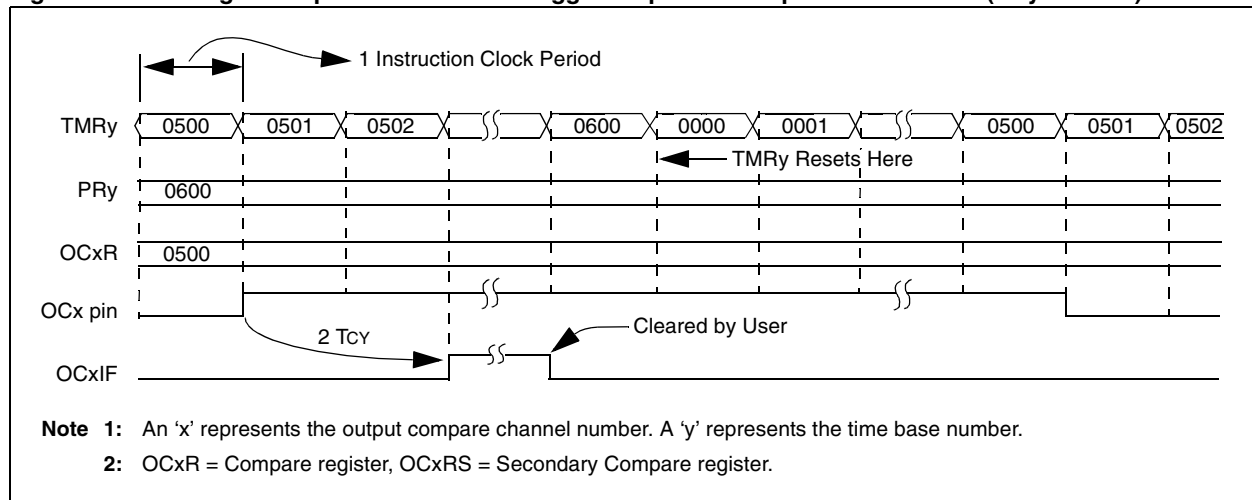
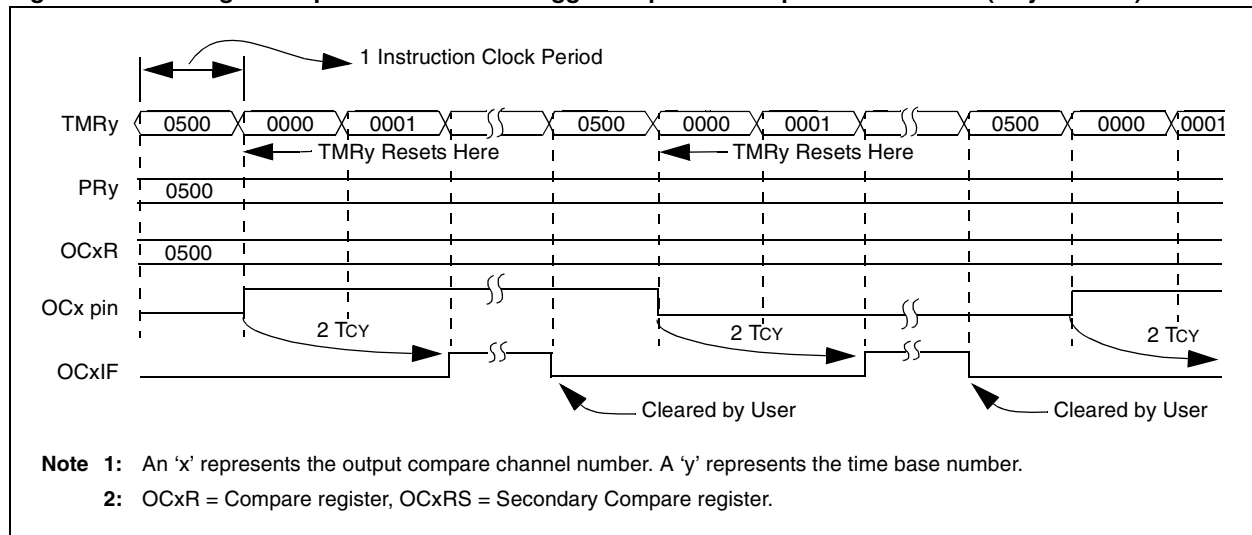


Figure 16-5: Single Compare Match Mode: Toggle Output on Compare Match Event ($PRy = OCxR$)^(1,2)



PIC24F Family Reference Manual

Example 16-1: Single Compare Match Mode: Toggle Mode Pin State Setup

```
// 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
```

Example 16-2 shows example code for the configuration and interrupt service of the Single Compare Match mode toggle event.

Example 16-2: Single Compare Match Mode: Toggle Setup and Interrupt Servicing

```
// 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 SPECIAL CASES OF SINGLE COMPARE MATCH MODE

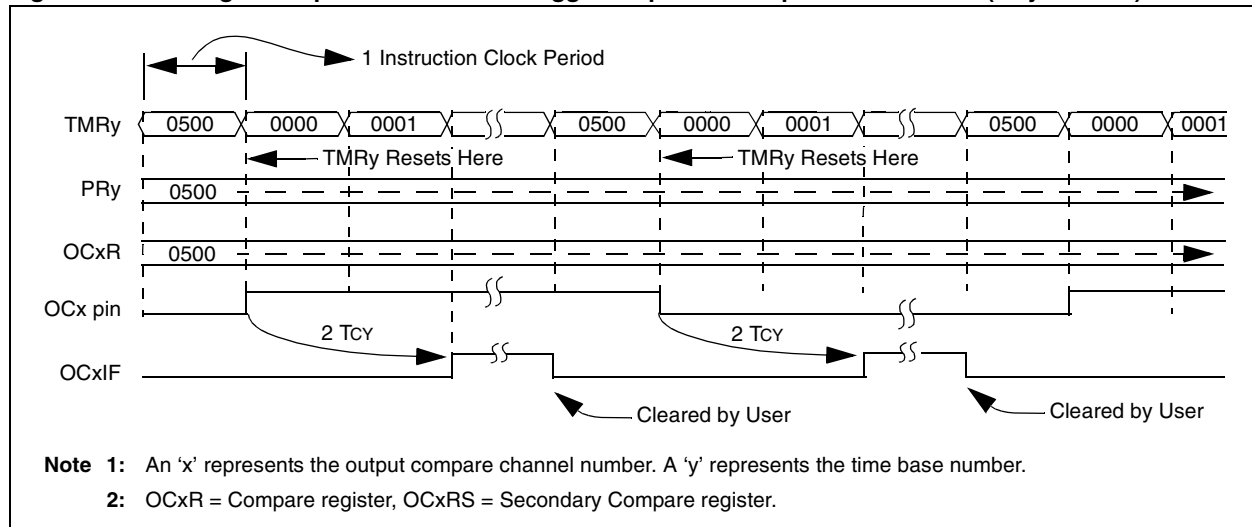
There are several special cases to consider.

When the $OCxR > PRy$, implying that the compare value is greater than the timer count, no compare event will occur and the compare output will remain at the initial condition. When the $OCxR = PRy$, implying that the compare interval is the same as the timer period, the compare output will function normally. Combining this with the Toggle mode can be used to generate a fixed frequency square wave, as shown in Figure 16-5.

When the module is enabled into a Single Compare Match mode and if $OCxR = 0000h$ and $PRy = 0000h$, implying no period for the timer count, then the compare output will remain at the initial condition.

If, after a compare event, the $OCxR$ and PRy registers are cleared, the compare output will remain at its previous state.

Figure 16-6: Single Compare Match Mode: Toggle Output on Compare Match Event ($PRy > OCxR$)^(1,2)



16.3.2 Dual Compare Match Mode

When control bits $OCM\langle 2:0 \rangle = 100$ or 101 ($OCxCON\langle 2:0 \rangle$), the selected output compare channel is configured for one of two Dual Compare Match modes which are:

- Single Output Pulse mode
- Continuous Output Pulse mode

In the Dual Compare mode, the module uses both the $OCxR$ and $OCxRS$ registers for the compare match events. The $OCxR$ register is compared against the incrementing timer count, $TMRy$, and the leading (rising) edge of the pulse is generated at the OCx pin, on a compare match event. The $OCxRS$ register is then compared to the same incrementing timer count, $TMRy$, and the trailing (falling) edge of the pulse is generated at the OCx pin, on a compare match event.

16.3.2.1 DUAL COMPARE MATCH MODE: SINGLE OUTPUT PULSE

To configure the output compare module for the Single Output Pulse mode, set control bits $OCM\langle 2:0 \rangle = 100$. In addition, the $TMRy$ must be selected and enabled. Once this mode has been enabled, the output pin, OCx , will be driven low and remain low until a match occurs between the time base and $OCxR$ registers. Referring to Figure 16-7 and Figure 16-9, there are some key timing events to note:

- The OCx pin is driven high one instruction clock after the compare match occurs between the $TMRy$ and $OCxR$ register. The OCx pin will remain high until the next match event occurs between the time base and the $OCxRS$ register. At this time, the pin will be driven low. The OCx pin will remain low until a mode change has been made, or the module is disabled.
- $TMRy$ will count up to the value contained in the associated period register and then reset to 0000h on the next instruction clock.
- If the $TMRy$ register content is less than the $OCxRS$ register content, then no falling edge of the pulse is generated. The OCx pin will remain high until $OCxRS \leq PRy$, or a mode change or Reset condition has occurred.
- The respective channel interrupt flag, $OCxIF$, is asserted two instruction clocks after the OCx pin is driven low (falling edge of single pulse).

Figure 16-7 and Figure 16-8 depict the Dual Compare Match mode generating a single output pulse. Figure 16-9 depicts another timing example where $OCxRS > PRy$. In this example, no falling edge of the pulse is generated since the $TMRy$ resets before counting up to 4100h.

Figure 16-7: Dual Compare Match Mode^(1,2)

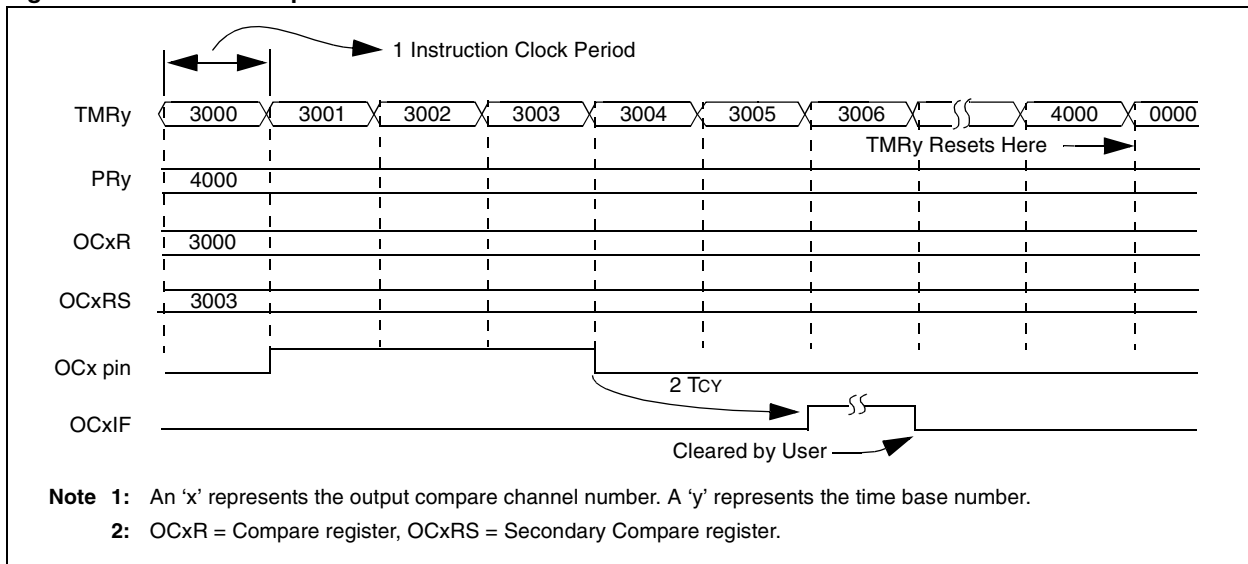


Figure 16-8: Dual Compare Match Mode: Single Output Pulse Mode^(1,2)

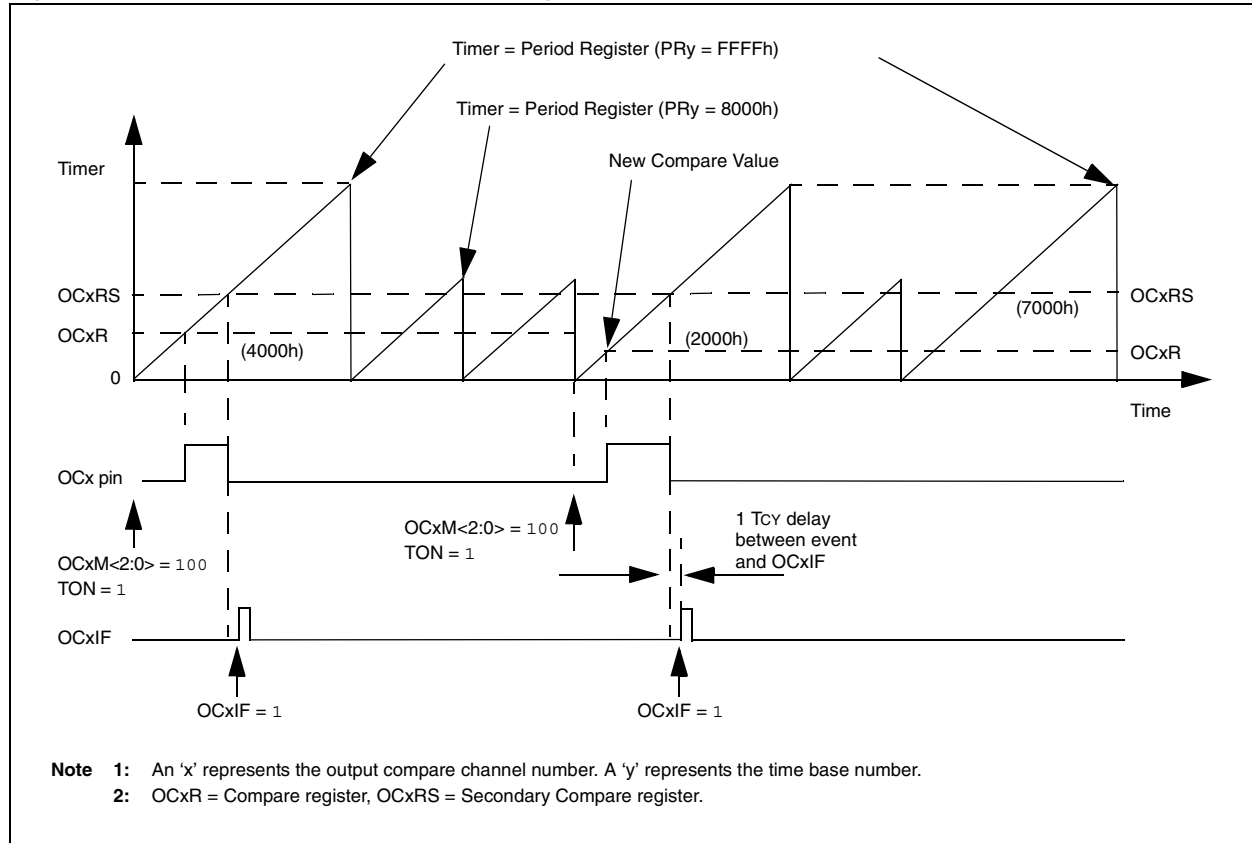
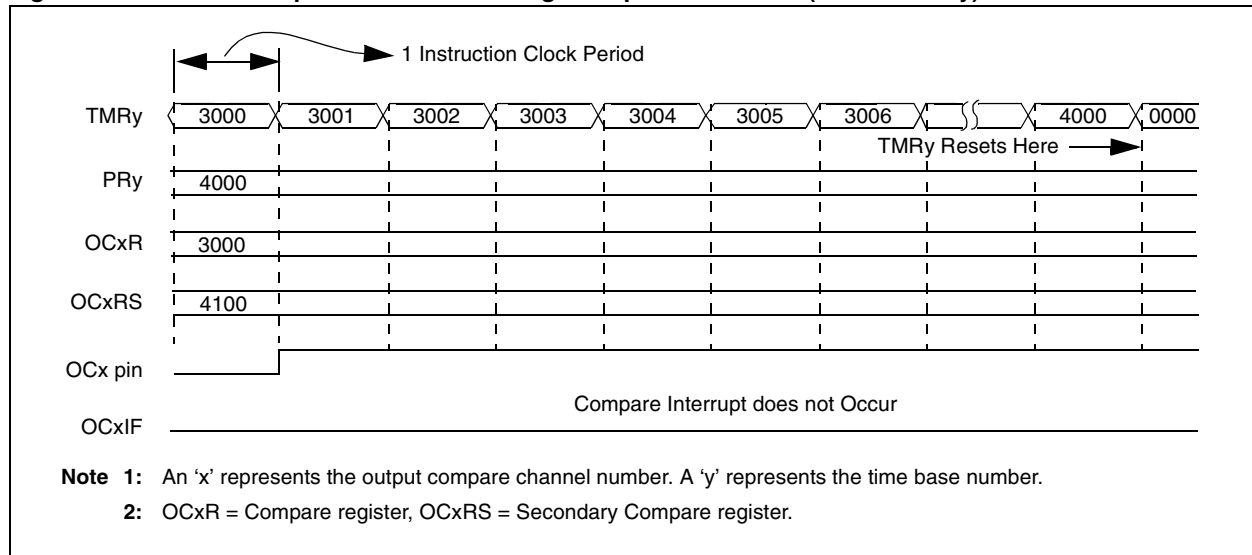


Figure 16-9: Dual Compare Match Mode: Single Output Pulse Mode (OCxRS > PRy)^(1,2)



16.3.2.2 SETUP FOR SINGLE OUTPUT PULSE GENERATION

When control bits, OCM<2:0> (OCxCON<2:0>), are set to '100', the selected output compare channel initializes the OCx pin to the low state and generates a single output pulse.

To generate a single output pulse, the following steps are required (these steps assume timer source is initially turned off, but this is not a requirement for the module operation):

1. Determine the instruction clock cycle time. Take into account the frequency of the external clock to the timer source (if one is used) and the timer prescaler settings.
2. Calculate time to the rising edge of the output pulse relative to the TMRy start value (0000h).
3. Calculate the time to the falling edge of the pulse based on the desired pulse width and the time to the rising edge of the pulse.
4. Write the values computed in steps 2 and 3 above into the Compare register, OCxR, and the Secondary Compare register, OCxRS, respectively.
5. Set Timer Period register, PRy, to value equal to or greater than value in OCxRS, the Secondary Compare register.
6. Set OCM<2:0> = 100 and the OCTSEL (OCxCON<3>) bit to the desired timer source. The OCx pin state will now be driven low.
7. Set the TON (TyCON<15>) bit to '1', which enables the TMRy to count.
8. Upon the first match between TMRy and OCxR, the OCx pin will be driven high.
9. When the incrementing timer, TMRy, matches the Secondary Compare register, OCxRS, the second and trailing edge (high-to-low) of the pulse is driven onto the OCx pin. No additional pulses are driven onto the OCx pin and it remains at low. As a result of the second compare match event, the OCxIF interrupt flag bit set which will result in an interrupt, if it is enabled, by setting the OCxIE bit. For further information on peripheral interrupts, refer to **Section 8. "Interrupts"**.
10. To initiate another single pulse output, change the Timer and Compare register settings, if needed, and then issue a write to set the OCM<2:0> (OCxCON<2:0>) bits to '100'. Disabling and re-enabling the timer and clearing the TMRy register are not required, but may be advantageous for defining a pulse from a known event time boundary.

The output compare module does not have to be disabled after the falling edge of the output pulse. Another pulse can be initiated by rewriting the value of the OCxCON register.

Note: Minimum time difference between OCxR and OCxRS is 2 Tcy when the prescaler is 1:1.

Example 16-3 shows example code for configuration of the single output pulse event.

Example 16-3: Single Output Pulse Mode Setup and Interrupt Servicing

```
// 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;
}
```

PIC24F Family Reference Manual

16.3.2.3 SPECIAL CASES FOR DUAL COMPARE MATCH MODE GENERATING A SINGLE OUTPUT PULSE

Depending on the relationship of the OCxR, OCxRS and PRy values, the output compare module has a few unique conditions which should be understood. These special conditions are specified in Table 16-1, along with the resulting behavior of the module.

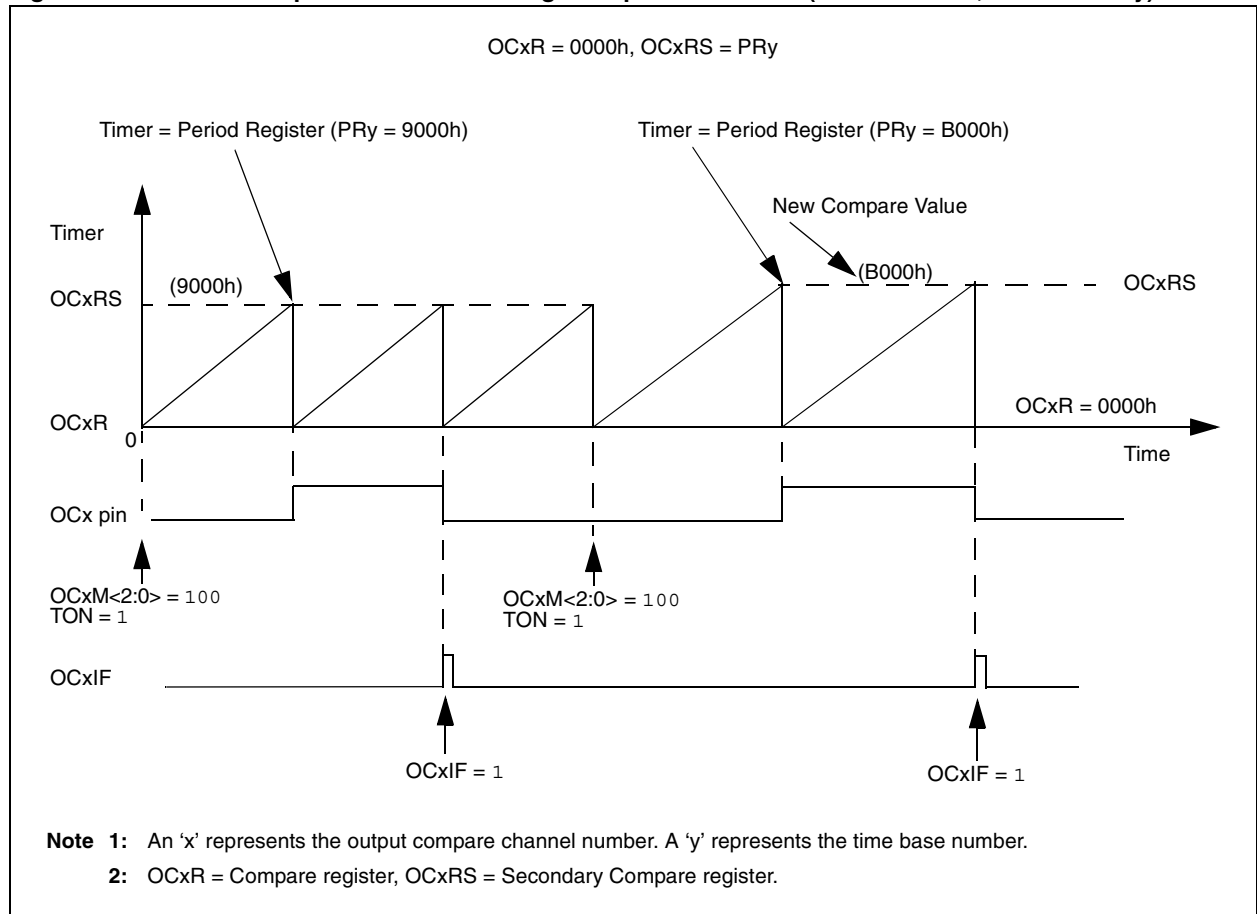
Table 16-1: Special Cases for Dual Compare Match Mode Generating a Single Output Pulse^(1,2)

SFR Logical Relationship	Special Conditions	Operation	Output at OCx
$PRy \geq OCxRS$ and $OCxRS > OCxR$	$OCxR = 0$, Initialize $TMRy = 0$	In the first iteration of the $TMRy$ counting from 0000h up to PRy , the OCx pin remains low, no pulse is generated. After the $TMRy$ resets to zero (on period match), the OCx pin goes high due to match with $OCxR$. Upon the next $TMRy$ to $OCxRS$ match, the OCx pin goes low and remains there. The OCxIF bit will be set as a result of the second compare. There are two alternative initial conditions to consider: a) Initialize $TMRy = 0$ and set $OCxR \geq 1$. b) Initialize $TMRy = PRy$ ($PRy > 0$) and set $OCxR = 0$ (see Figure 16-10).	Pulse will be delayed by the value in the PRy register depending on the setup.
$PRy \geq OCxR$ and $OCxR \geq OCxRS$	$OCxR \geq 1$ and $PRy \geq 1$	$TMRy$ counts up to $OCxR$ and on a compare match event (i.e., $TMRy = OCxR$), the OCx pin is driven to a high state. $TMRy$ then continues to count and eventually resets on period match (i.e., $PRy = TMRy$). The timer then restarts from 0000h and counts up to $OCxRS$, and on a compare match event (i.e., $TMRy = OCxRS$), the OCx pin is driven to a low state. The OCxIF bit will be set as a result of the second compare.	Pulse.
$OCxRS > PRy$ and $PRy \geq OCxR$	None	Only the rising edge will be generated at the OCx pin. The OCxIF will not be set.	Rising edge/transition to high.
$OCxR = OCxRS = PRy = 0000h$	None	Output is initialized low and remains low. The OCxIF bit is not set.	Remains low.
$OCxR > PRy$	None	Unsupported mode, timer resets prior to match condition.	Remains low.

Note 1: In all the cases considered herein, the $TMRy$ register is assumed to be initialized to 0000h.

2: $OCxR$ = Compare register, $OCxRS$ = Secondary Compare register, $TMRy$ = Timery Count register, PRy = Timery Period register.

Figure 16-10: Dual Compare Match Mode: Single Output Pulse Mode (OCxR = 0000h, OCxRS = PRy)^(1,2)



16.3.2.4 DUAL COMPARE MATCH MODE: CONTINUOUS OUTPUT PULSE

To configure the output compare module for this mode, set control bits, $OCM<2:0> = 101$. In addition, the TMRy must be selected and enabled. Once this mode has been enabled, the output pin, OCx, will be driven low and remain low until a match occurs between the TMRy and OCxR register. Referring to Figure 16-11 and Figure 16-13, there are some key timing events to note:

- The OCx pin is driven high one instruction clock after the compare match occurs between the TMRy and OCxR register. The OCx pin will remain high until the next match event occurs between the time base and the OCxRS register, at which time, the pin will be driven low. This pulse generation sequence of a low-to-high and high-to-low edge will repeat on the OCx pin without further user intervention.
- Continuous pulses will be generated on the OCx pin until a mode change is made or the module is disabled.
- The TMRy will count up to the value contained in the associated Period register and then reset to 0000h on the next instruction clock.
- If the TMRy Period register value is less than the OCxRS register value, then no falling edge is generated. The OCx pin will remain high until $OCxRS \leq PRy$, a mode change is made or the device is reset.
- The respective Channel Interrupt Flag, OCxIF, is asserted two instruction clocks after the OCx pin is driven low (falling edge of single pulse).

Figure 16-11 and Figure 16-12 depict the Dual Compare Match mode generating a continuous output pulse. Figure 16-13 depicts another timing example, where $OCxRS > PRy$. In this example, no falling edge of the pulse is generated, since the time base will reset before counting up to the contents of OCxRS.

Figure 16-11: Dual Compare Match Mode: Continuous Output Pulse Mode ($PRy = OCxRS$)^(1,2)

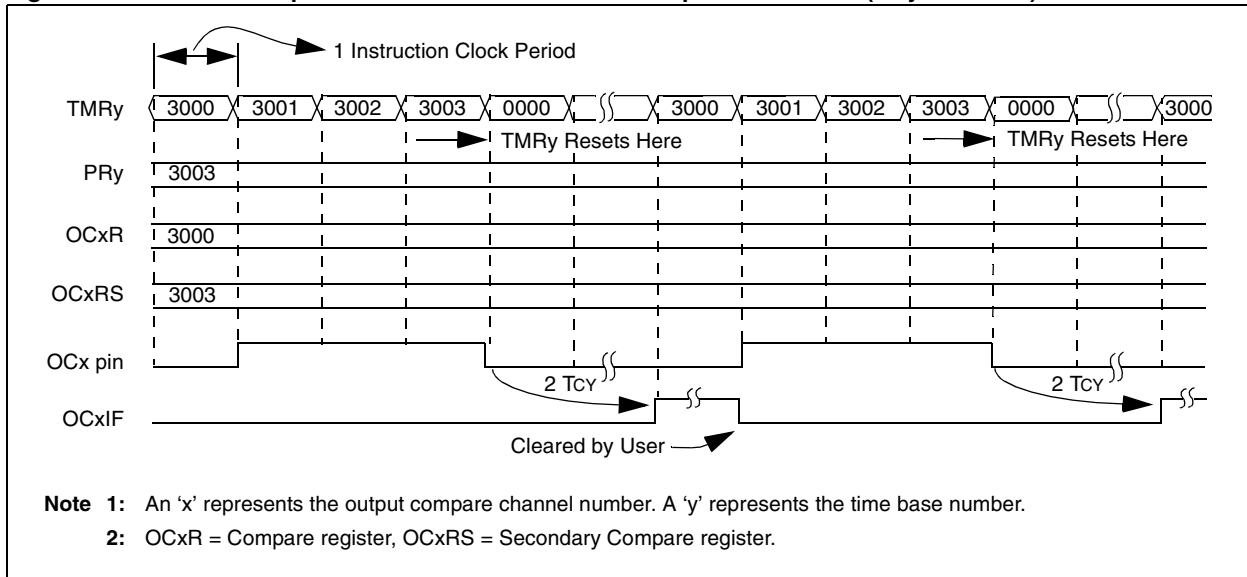


Figure 16-12: Dual Compare Match Mode: Continuous Output Pulse Mode^(1,2)

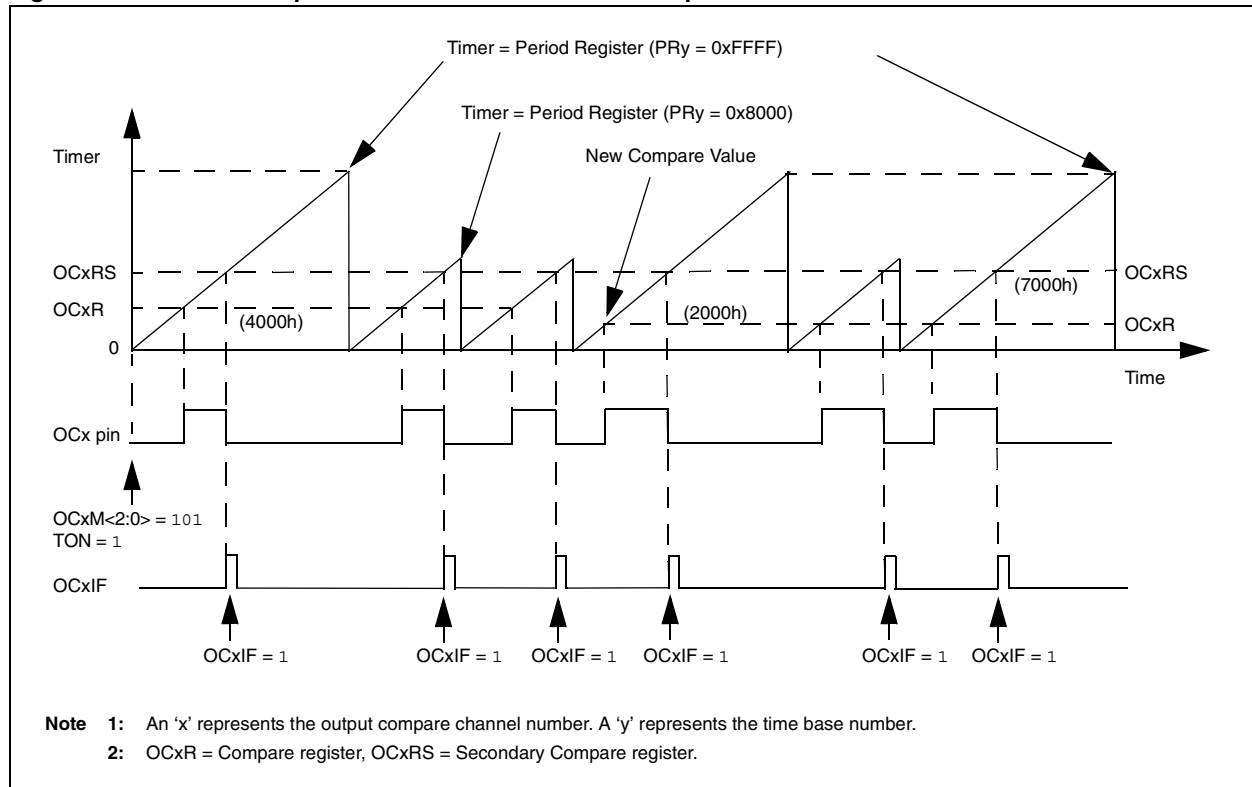
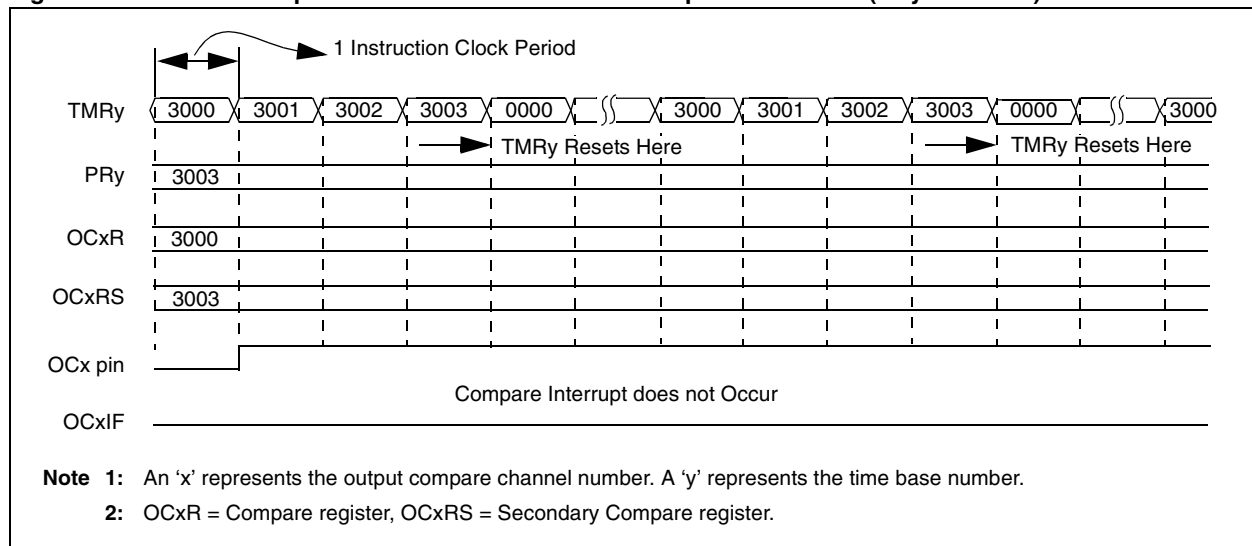


Figure 16-13: Dual Compare Match Mode: Continuous Output Pulse Mode (PRy < OCxRS)^(1,2)



16.3.2.5 SETUP FOR CONTINUOUS OUTPUT PULSE GENERATION

When control bits, OCxM<2:0> (OCxCON<2:0>), are set to '101', the selected output compare channel initializes the OCx pin to the low state and generates output pulses on each and every compare match event.

For the user to configure the module for the generation of a continuous stream of output pulses, the following steps are required (these steps assume timer source is initially turned off, but this is not a requirement for the module operation):

1. Determine the instruction clock cycle time. Take into account the frequency of the external clock to the timer source (if one is used) and the timer prescaler settings.
2. Calculate time to the rising edge of the output pulse relative to the TMRy start value (0000h).
3. Calculate the time to the falling edge of the pulse, based on the desired pulse width and the time to the rising edge of the pulse.
4. Write the values computed in step 2 and 3 above into the Compare register, OCxR, and the Secondary Compare register, OCxRS, respectively.
5. Set Timer Period register, PRy, to value equal to or greater than value in OCxRS, the Secondary Compare register.
6. Set OCM<2:0> = 101 and the OCTSEL (OCxCON<3>) bit to the desired timer source. The OCx pin state will now be driven low.
7. Enable the TMRy by setting the TON (TyCON<15>) bit to '1'.
8. Upon the first match between TMRy and OCxR, the OCx pin will be driven high.
9. When the compare time base, TMRy, matches the Secondary Compare register, OCxRS, the second and trailing edge (high-to-low) of the pulse is driven onto the OCx pin.
10. As a result of the second compare match event, the OCxIF interrupt flag bit is set.
11. When TMRy and the value in its respective Period register match, the TMRy register resets to 0000h and resumes counting.
12. Steps 8 through 11 are repeated and a continuous stream of pulses is generated, indefinitely. The OCxIF flag is set on each OCxRS-TMRy compare match event.

Note: Minimum time difference between OCxR and OCxRS is 2 Tcy when the prescaler is 1:1.

Example 16-4 shows example code for configuration of the continuous output pulse event.

Example 16-4: Continuous Output Pulse Setup and Interrupt Servicing

```
// 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 SPECIAL CASES FOR DUAL COMPARE MATCH MODE GENERATING CONTINUOUS OUTPUT PULSE MODE

Depending on the relationship of the OCxR, OCxRS and PRy values, the output compare module may not provide the expected results. These special cases are specified in Table 16-2, along with the resulting behavior of the module.

Table 16-2: Special Cases for Dual Compare Match Mode Generating Continuous Output Pulse Mode^(1,2)

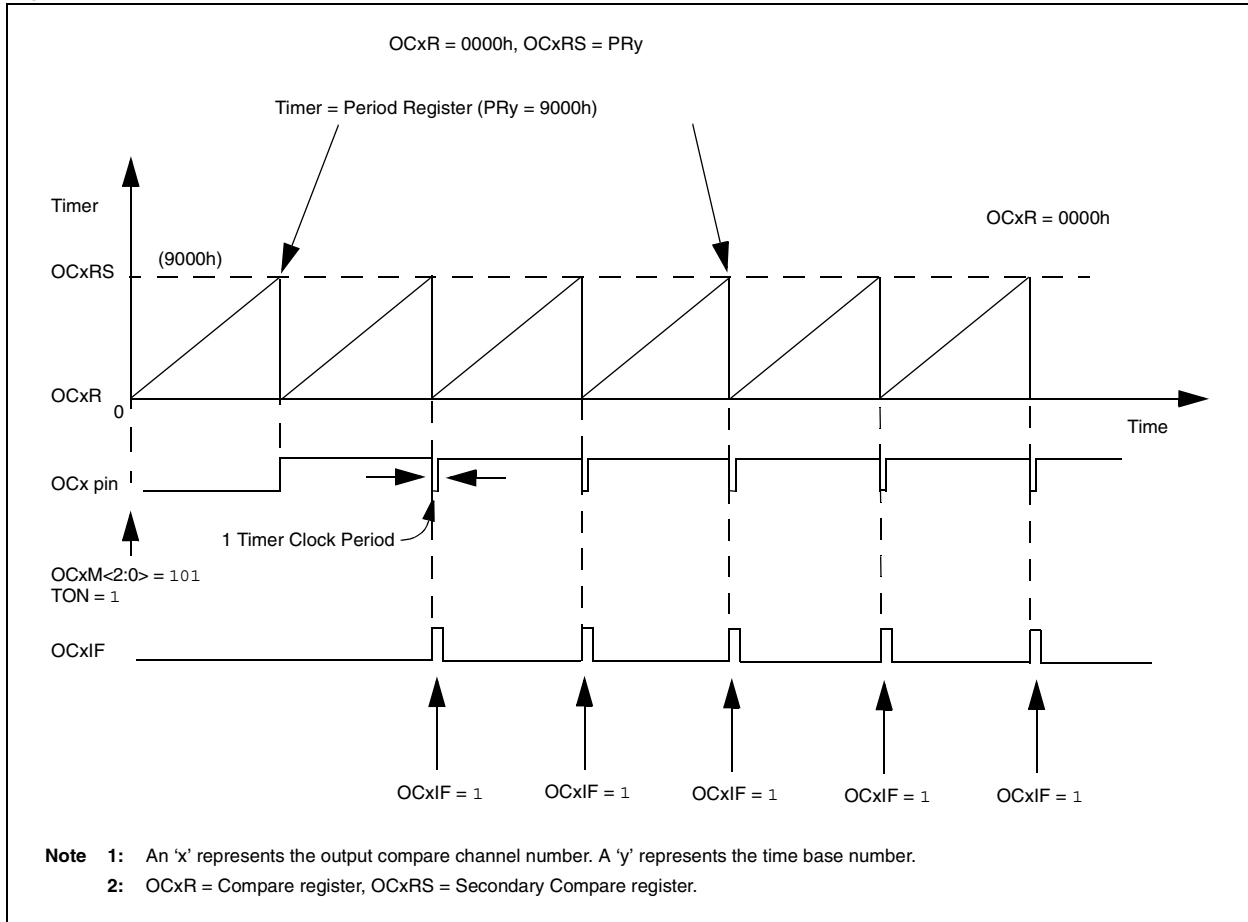
SFR Logical Relationship	Special Conditions	Operation	Output at OCx
$PRy \geq OCxRS$ and $OCxRS > OCxR$	$OCxR = 0$, initialize $TMRy = 0$	In the first iteration of the $TMRy$ counting from 0000h up to PRy , the OCx pin remains low, no pulse is generated. After the $TMRy$ resets to zero (on period match), the OCx pin goes high. Upon the next $TMRy$ to $OCxRS$ match, the OCx pin goes low. If $OCxR = 0$ and $PRy = OCxRS$, the pin will remain low for one clock cycle, then be driven high until the next $TMRy$ to $OCxRS$ match. The OCxIF bit will be set as a result of the second compare. There are two alternative initial conditions to consider: a) Initialize $TMRy = 0$ and set $OCxR \geq 1$. b) Initialize $TMRy = PRy$ ($PRy > 0$) and set $OCxR = 0$ (see Figure 16-14).	Continuous pulses with the first pulse delayed by the value in the PRy register, depending on setup.
$PRy \geq OCxR$ and $OCxR \geq OCxRS$	$OCxR \geq 1$ and $PRy \geq 1$	$TMRy$ counts up to $OCxR$ and on a compare match event (i.e., $TMRy = OCxR$), the OCx pin is driven to a high state. $TMRy$ then continues to count and eventually resets on period match (i.e., $PRy = TMRy$). The timer then restarts from 0000h and counts up to $OCxRS$, and on a compare match event (i.e., $TMRy = OCxR$), the OCx pin is driven to a low state. The OCxIF bit will be set as a result of the second compare.	Continuous pulses.
$OCxRS > PRy$ and $PRy \geq OCxR$	None	Only one transition will be generated at the OCx pin until the OCxRS register contents have been changed to a value less than or equal to the Period register contents (PRy). OCxIF is not set until then.	Rising edge/ transition to high.
$OCxR = OCxRS = PRy = 0000h$	None	Output is initialized low and remains low. The OCxIF bit is not set.	Remains low.
$OCxR > PRy$	None	Unsupported mode, Timer resets prior to match condition.	Remains low.

Note 1: In all the cases considered herein, the $TMRy$ register is assumed to be initialized to 0000h.

2: $OCxR$ = Compare register, $OCxRS$ = Secondary Compare register, $TMRy$ = Timery Count, PRy = Timery Period register.

PIC24F Family Reference Manual

Figure 16-14: Dual Compare Match Mode: Continuous Output Pulse Mode (OCxR = 0x0000, OCxRS = PRy)^(1,2)



16.3.3 Simple Pulse-Width Modulation Mode

When control bits, OCM<2:0> (OCxCON<2:0>), are set to '110' or '111', the selected output compare channel is configured for the Simple PWM (Pulse-Width Modulation) mode of operation.

The following two PWM modes are available:

- PWM without Fault Protection Input
- PWM with Fault Protection Input

The OCFA or OCFB Fault input pin is utilized for the second PWM mode. In this mode, an asynchronous logic level '0' on the OCFx pin will cause the selected PWM channel to be shut down. (Described in **Section 16.3.3.1 "PWM with Fault Protection Input Pin"**.)

In PWM mode, the OCxR register is a read-only slave duty cycle register and OCxRS is a buffer register that is written by the user to update the PWM duty cycle. On every timer to Period register match event (end of PWM period):

1. TMRy is reset to zero and resumes counting.
2. OCx is set unless OCxRS = 0.
3. Duty cycle transferred from OCxRS to OxCr.
4. TyIF is set when TMRy and OCxR match, OCx is driven low.

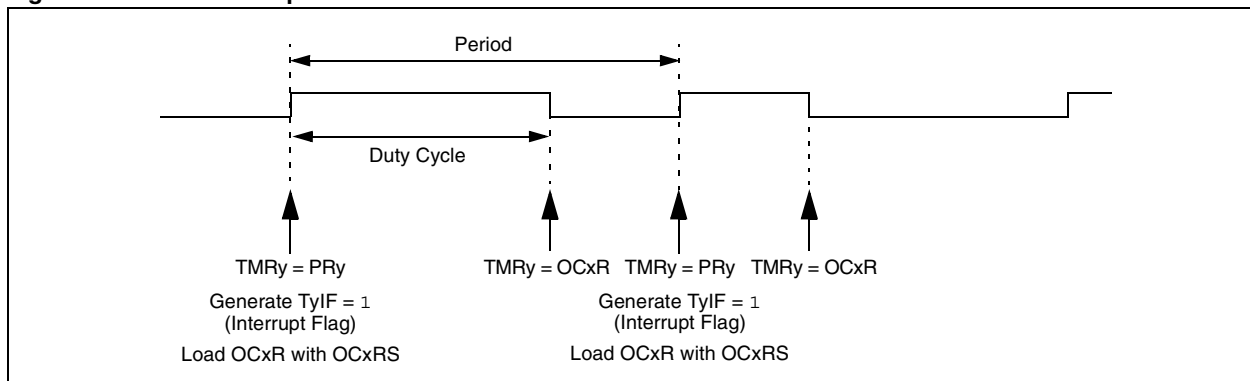
The following steps should be taken when configuring the output compare module for PWM operation:

1. Set the PWM period by writing to the selected Timer Period register (PRy).
2. Set the PWM duty cycle by writing to the OCxRS register.
3. Write the OCxR register with the initial duty cycle.
4. Enable interrupts, if required, for the timer and output compare modules. The output compare interrupt is required for PWM Fault pin utilization.
5. Configure the output compare module for one of two PWM Operation modes by writing to the Output Compare Mode bits, OCM<2:0> (OCxCON<2:0>).
6. Set the TMRy prescale value and enable the time base by setting TON (TxCON<15>) = 1.

Note: The OCxR register should be initialized before the output compare module is first enabled. The OCxR register becomes a read-only duty cycle register when the module is operated in the PWM modes. The value held in OCxR will become the PWM duty cycle for the first PWM period. The contents of the duty cycle buffer register, OCxRS, will not be transferred into OCxR until a time base period match occurs.

An example PWM output waveform is shown in Figure 16-15.

Figure 16-15: PWM Output Waveform



16.3.3.1 PWM WITH FAULT PROTECTION INPUT PIN

When the Output Compare Mode bits, OCM<2:0> (OCxCON<2:0>), are set to '111', the selected output compare channel is configured for the PWM mode of operation. All functions described in **Section 16.3.3 "Simple Pulse-Width Modulation Mode"** apply, with the addition of Fault Protection Input.

Fault protection is provided via the OCFA and OCFB pins. The OCFA pin is associated with the output compare channels 1 through 4, while the OCFB pin is associated with the output compare channel 5.

If a logic '0' is detected on the OCFA/OCFB pin, the selected PWM output pin(s) is placed in the high-impedance state. The user may elect to provide a pull-down or pull-up resistor on the PWM pin to provide for a desired state if a Fault condition occurs. The shutdown of the PWM output is immediate and is not tied to the device clock source. This state will remain until:

- The external Fault condition has been removed and
- The PWM mode is re-enabled by writing to the appropriate mode bits, OCM<2:0> (OCxCON<2:0>).

As a result of the Fault condition, the respective interrupt flag, OCxIF bit, is asserted and an interrupt will be generated, if enabled. Upon detection of the Fault condition, the OCFLT bit (OCxCON<4>) is asserted high (logic '1'). This bit is a read-only bit and will only be cleared once the external Fault condition has been removed and the PWM mode is re-enabled, by writing to the appropriate mode bits, OCM<2:0> (OCxCON<2:0>).

Note: The external Fault pins, if enabled for use, will continue to control the OCx output pins while the device is in Sleep or Idle mode.

16.3.3.2 PWM PERIOD

The PWM period is specified by writing to PRy, the TMRy Period register. The PWM period can be calculated using the following formula:

Equation 16-1: Calculating the PWM Period⁽¹⁾

$$\text{PWM Period} = [\text{PRy} + 1] \cdot \text{Tcy} \cdot (\text{TMRy Prescale Value})$$

$$\text{PWM Frequency} = 1/[\text{PWM Period}]$$

Note 1: Based on Tcy = 2/FOSC, Doze mode and PLL are disabled.

Note: A PRy value of N will produce a PWM period of N + 1 time base count cycles. For example, a value of 7 written into the PRy register will yield a period consisting of 8 time base cycles.

16.3.3.3 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the OCxRS register. The OCxRS register can be written to at any time, but the duty cycle value is not latched into OCxR until a match between PRy and TMRy occurs (i.e., the period is complete). This provides a double buffer for the PWM duty cycle and is essential for glitchless PWM operation. In the PWM mode, OCxR is a read-only register.

Some important boundary parameters of the PWM duty cycle include:

- If the duty cycle register, OCxR, is loaded with 0000h, the OCx pin will remain low (0% duty cycle).
- If OCxR is greater than PRy (Timer Period register), the pin will remain high (100% duty cycle).
- If OCxR is equal to PRy, the OCx pin will be low for one time base count value and high for all other count values.

See Figure 16-16 for PWM mode timing details. Table 16-3 and Table 16-4 show example PWM frequencies and resolutions for a device operating at 4 and 16 MIPS, respectively.

Equation 16-2: Calculation for Maximum PWM Resolution⁽¹⁾

$$\text{Maximum PWM Resolution (bits)} = \frac{\log_{10}\left(\frac{F_{CY}}{F_{PWM} \cdot (\text{Timer Prescale Value})}\right)}{\log_{10}(2)} \text{ bits}$$

Note 1: Based on $T_{CY} = 2/F_{OSC}$, Doze mode and PLL are disabled.

Example 16-5: PWM Period and Duty Cycle Calculation

- Find the Period register value for a desired PWM frequency of 52.08 kHz, where $F_{OSC} = 8$ MHz with PLL (32 MHz device clock rate) and a Timer2 prescaler setting of 1:1.
 $T_{CY} = 2/F_{OSC} = 62.5$ ns
 PWM Period = $1/\text{PWM Frequency} = 1/52.08$ kHz = 19.2s
 PWM Period = $(PR2 + 1) * T_{CY} * (\text{Timer2 Prescale Value})$
 $19.2s = (PR2 + 1) * 62.5$ ns * 1
 $PR2 = 306$

Figure 16-16: PWM Output Timing^(1,2)

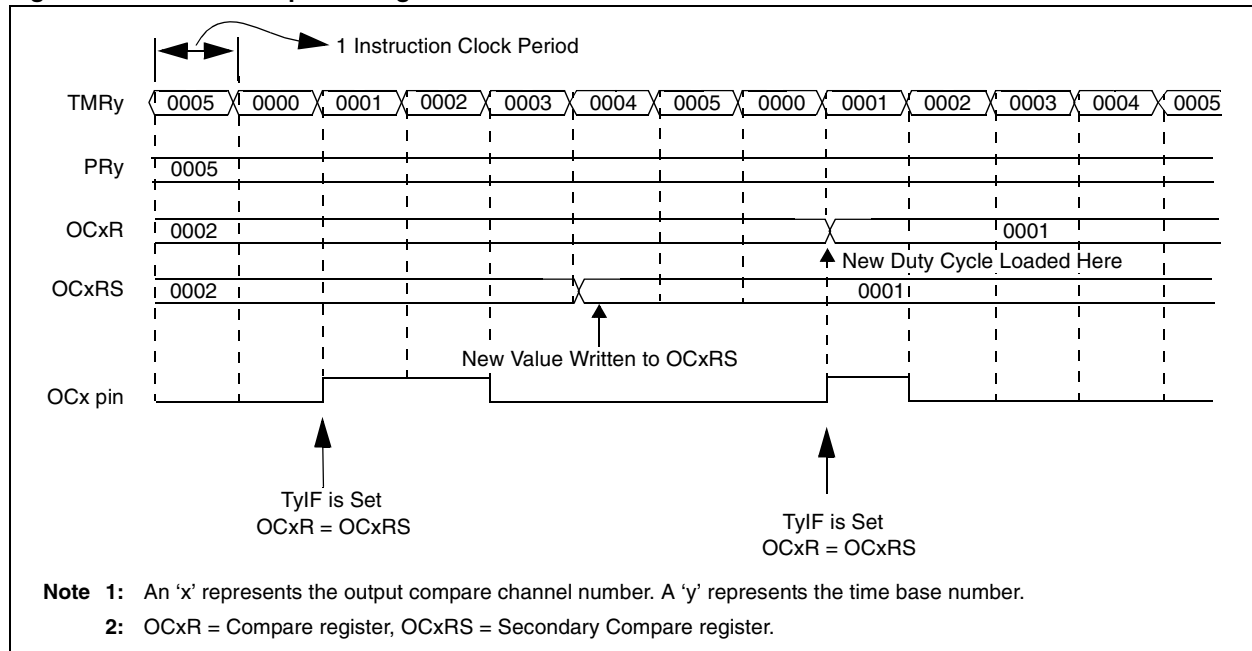


Table 16-3: Example PWM Frequencies and Resolutions at 4 MIPS ($F_{CY} = 4$ MHz)⁽¹⁾

PWM Frequency	7.6 Hz	61 Hz	122 Hz	977 Hz	3.9 kHz	31.3 kHz	125 kHz
Timer Prescaler Ratio	8	1	1	1	1	1	1
Period Register Value	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
Resolution (bits)	16	16	15	12	10	7	5

Note 1: Based on $T_{CY} = 2/F_{OSC}$, Doze mode and PLL are disabled.

Table 16-4: Example PWM Frequencies and Resolutions at 16 MIPS ($F_{CY} = 16$ MHz)⁽¹⁾

PWM Frequency	30.5 Hz	244 Hz	488 Hz	3.9 kHz	15.6 kHz	125 kHz	500 kHz
Timer Prescaler Ratio	8	1	1	1	1	1	1
Period Register Value	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
Resolution (bits)	16	16	15	12	10	7	5

Note 1: Based on $T_{CY} = 2/F_{OSC}$, Doze mode and PLL are disabled.

16.3.3.4 SIMPLE PWM MODE INITIALIZATION

1. Once the Simple PWM mode is enabled, $OCxM\langle 2:0 \rangle = 110$ or 111 , the pin state will be driven low if $OCxR = 0000h$. If $OCxR$ does not equal zero, then the pin state will be set high. At some point, the timer should be enabled to allow for correct operation (see Figure 16-17 and Figure 16-18).
2. When $OCxR$ is not equal to zero and the pin state is set to high, the first match between the duty cycle and the timer drives the pin low. The pin will remain low until a valid compare between the timer and Period register occurs (see Figure 16-18).

Figure 16-17: Simple PWM Mode: Initialized Low^(1,2)

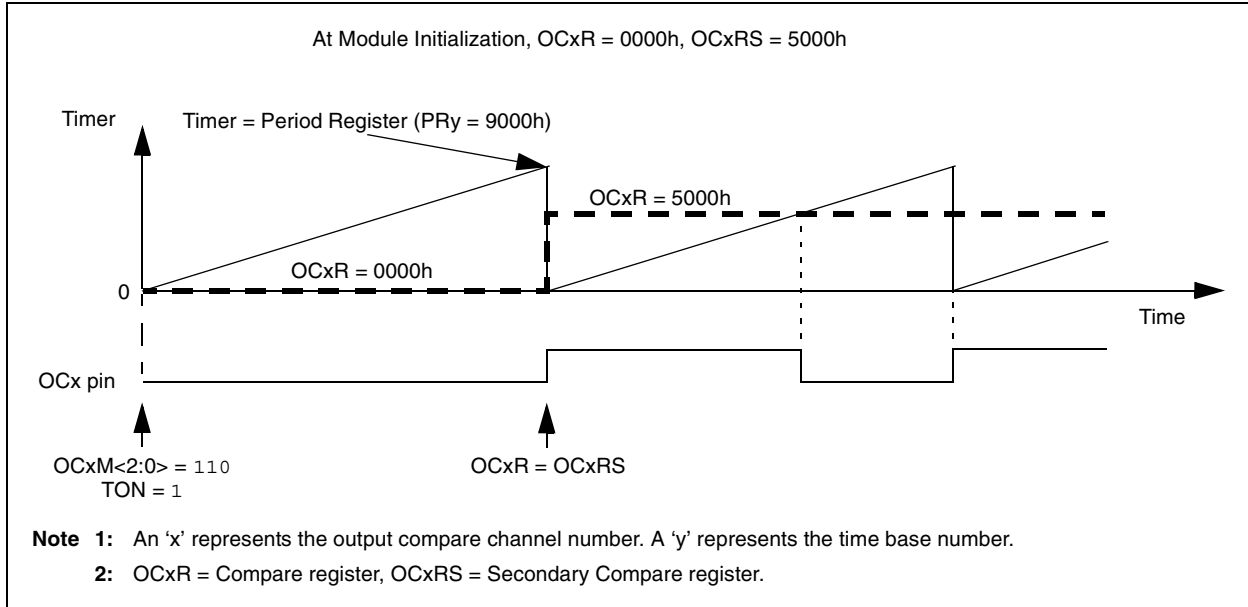
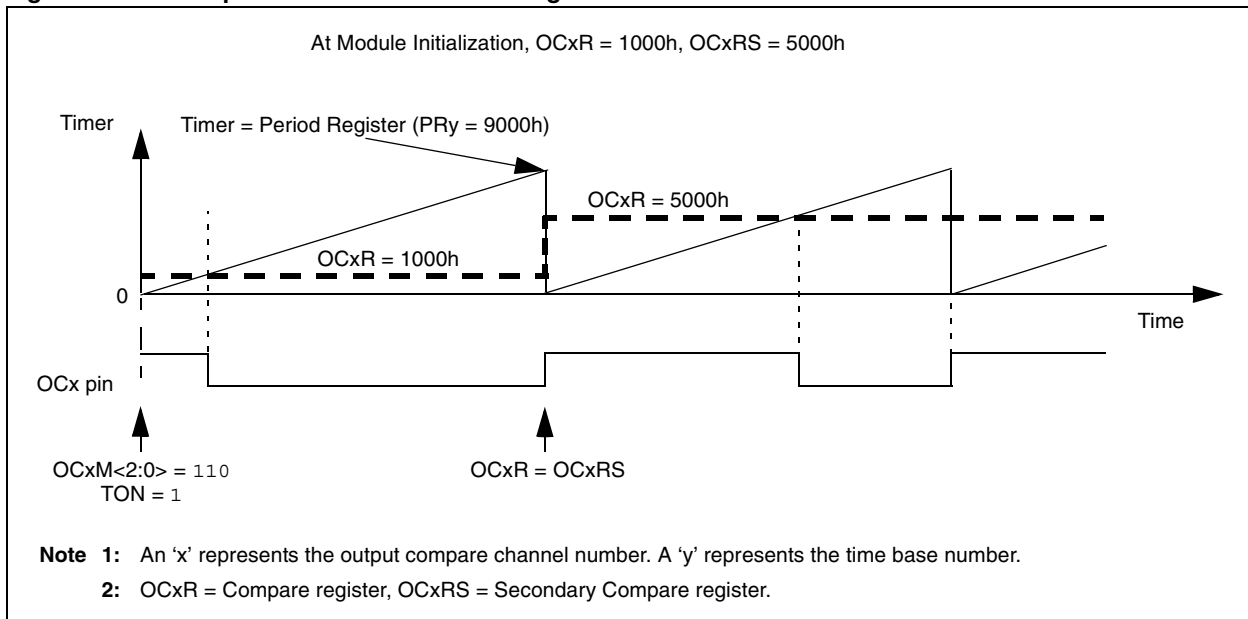


Figure 16-18: Simple PWM Mode: Initialized High^(1,2)



Example 16-6 shows configuration and interrupt service code for the PWM mode of operation.

Example 16-6: Simple PWM Mode: Pulse Setup and Interrupt Servicing

```
// 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.

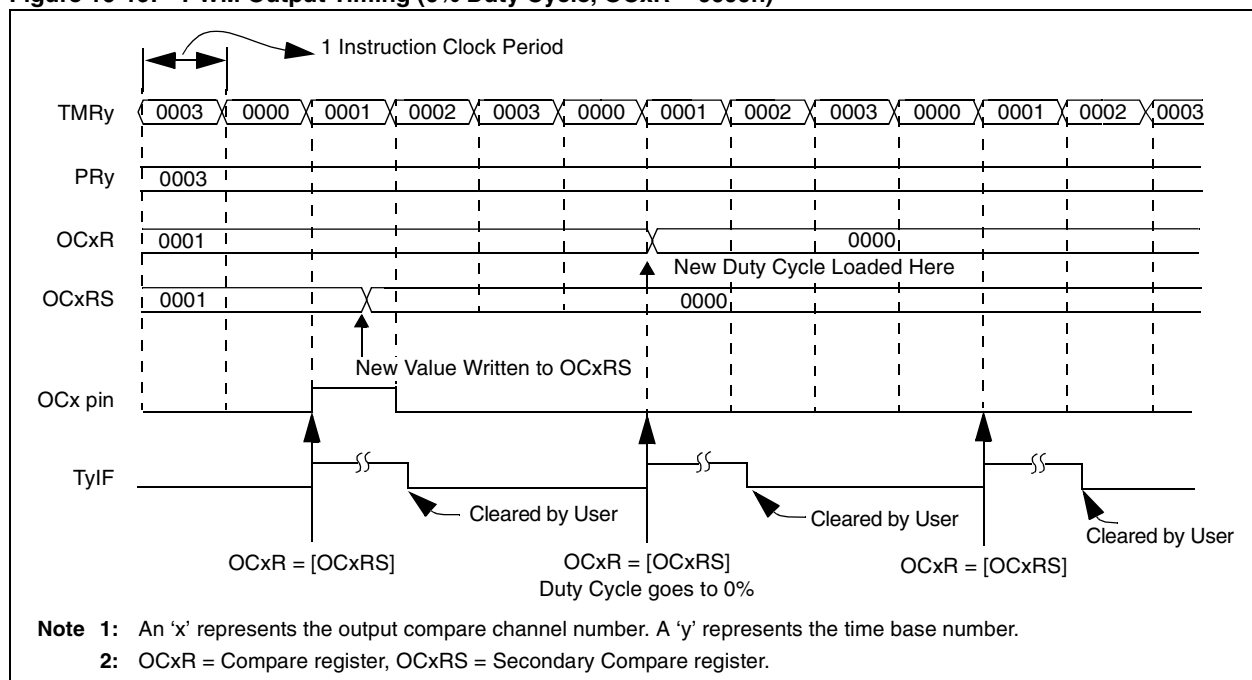
OC1CON      = 0x0000;          // Turn off Output Compare 1 Module
OC1R        = 0x0026;          // Initialize Compare Register1 with 0x0026
OC1RS       = 0x0026;          // Initialize Secondary Compare Register1 with 0x0026
OC1CON      = 0x0006;          // Load new compare mode to OC1CON
PR2         = 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 SIMPLE PWM MODE SPECIAL COMPARE CONDITIONS

1. If OCxR and the PWM Period register equal 0000h, then the pin will be set low.
2. If OCxR is equal to zero and the PWM Period register is equal to a non-zero value, then the pin will be set low (see Figure 16-19).
3. If OCxR is greater than the PWM Period register, the pin will remain high (see Figure 16-20).
4. If both (OCxR and PRy) are equal to some non-zero value, the output pin will go low for no more than 1 timer clock cycle, then immediately be set high (see Figure 16-21).

Figure 16-19: PWM Output Timing (0% Duty Cycle, OCxR = 0000h)^(1,2)



PIC24F Family Reference Manual

Figure 16-20: PWM Output Timing (100% Duty Cycle, OCxR > PRy)^(1,2)

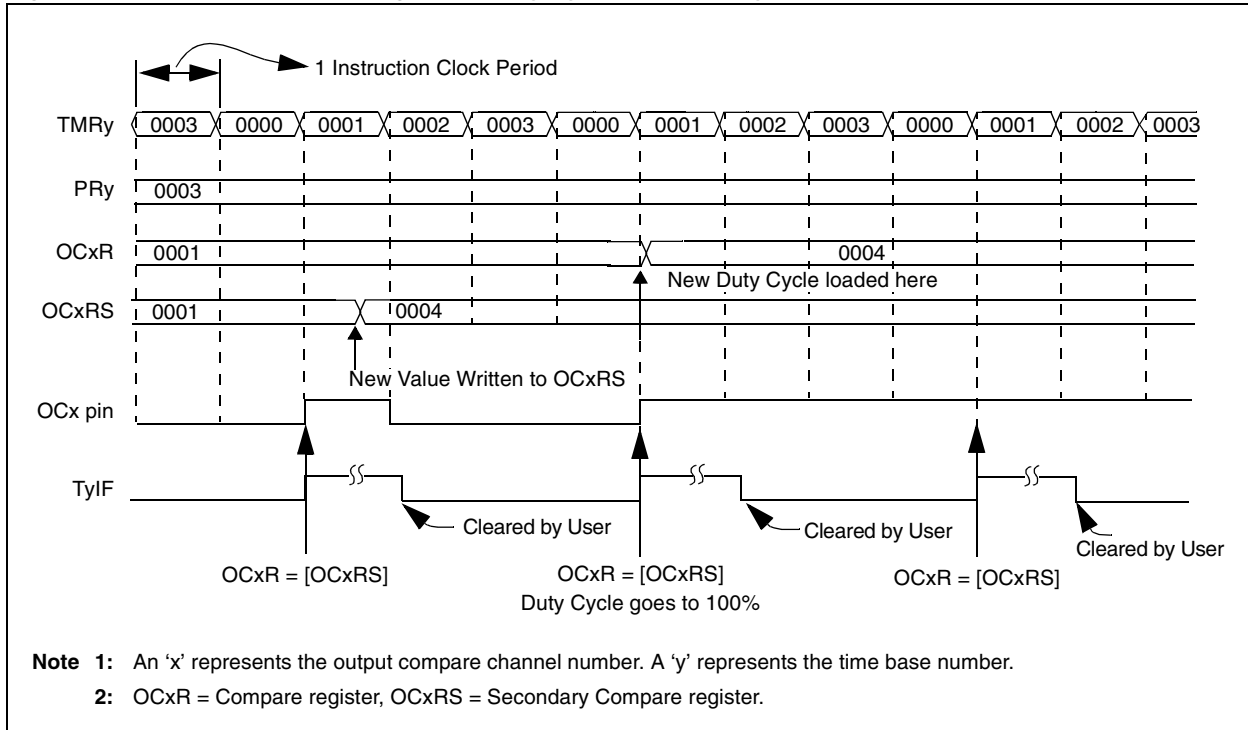
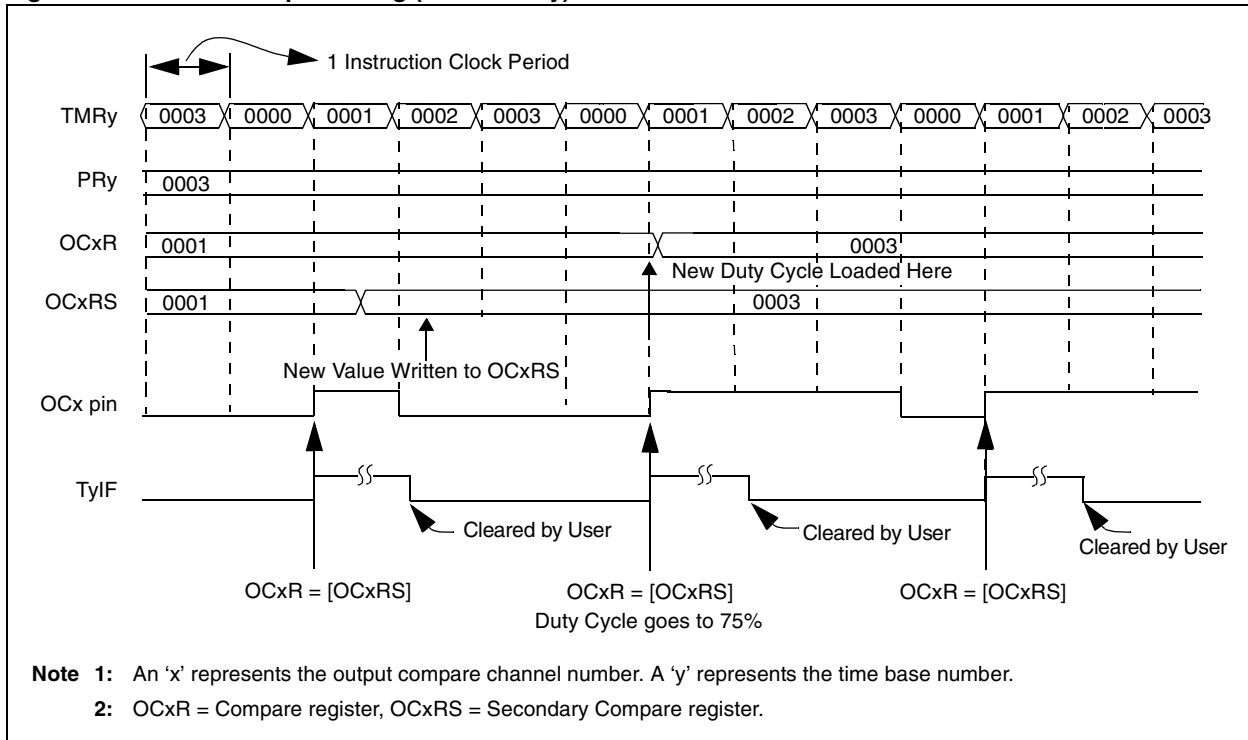


Figure 16-21: PWM Output Timing (OCxR = PRy)^(1,2)



16.4 OUTPUT COMPARE OPERATION IN POWER-SAVING STATES

16.4.1 Output Compare Operation in Sleep Mode

When the device enters Sleep mode, the system clock is disabled. During Sleep, the output compare channel will drive the pin to the same active state as driven prior to entering Sleep. The module will then halt at this state.

For example, if the pin was high and the CPU entered the Sleep state, the pin will stay high. Likewise, if the pin was low and the CPU entered the Sleep state, the pin will stay low. In both cases, when the part wakes up, the output compare module will resume operation.

16.4.2 Sleep With PWM Fault Mode

When the module is in PWM Fault mode, the asynchronous portions of the Fault circuit will remain active. If a Fault is detected, the OCx pin will be tri-stated. The OCFLT bit will be set. An interrupt will not be generated at Fault occurrence, however, the interrupt will be queued and will occur at the time the part wakes up.

16.4.3 Output Compare Operation in Idle Mode

When the device enters Idle mode, the system clock sources remain functional and the CPU stops executing code. The OCSIDL bit (OCxCON<13>) selects if the output capture module will stop in Idle mode or continue operation in Idle mode.

- If OCSIDL = 1, the module will discontinue operation in Idle mode. The module will perform the same procedures when stopped in Idle mode (OCSIDL = 1) as it does for Sleep mode.
- If OCSIDL = 0, the module will continue operation in Idle only if the selected time base is set to operate in Idle mode. The output compare channel(s) will operate during the CPU Idle mode if the OCSIDL bit is a logic '0'. Furthermore, the time base must be enabled with the respective TSIDL bit set to a logic '0'.

Note: The external Fault pins, if enabled for use, will continue to control the associated OCx output pins while the device is in Sleep or Idle mode.

16.4.4 Doze Mode

Output compare operation in Doze mode is the same as in normal mode. When the device enters Doze mode, the system clock sources remain functional and the CPU may run at a slower clock rate.

Refer to **Section 10. "Power-Saving Features"** for further details.

16.4.5 Selective Peripheral Module Control

The Peripheral Module Disable (PMD) registers provide a method to disable the output compare module by stopping all clock sources supplied to it. When the module is disabled, via the appropriate PMD control bit, it is in minimum power consumption state. The control and status registers associated with the module will also be disabled, so writes to these registers will have no effect, and read values will be invalid and return zero.

Refer to **Section 10. "Power-Saving Features"** for further details.

16.5 I/O PIN CONTROL

When the output compare module is enabled, the I/O pin direction is controlled by the compare module. The compare module returns the I/O pin control back to the appropriate LAT and TRIS control bits when it is disabled.

When the Simple PWM with Fault Protection Input mode is enabled, the OCFx Fault pin must be configured for an input by setting the respective TRIS bit. Enabling this special PWM mode does not configure the OCFx Fault pin as an input.

Table 16-5: Pins Associated with Output Compare Modules 1-5

Pin Name	Pin Type	Description
OC1	O	Output Compare/PWM Channel 1
OC2	O	Output Compare/PWM Channel 2
OC3	O	Output Compare/PWM Channel 3
OC4	O	Output Compare/PWM Channel 4
OC5	O	Output Compare/PWM Channel 5
OCFA	I	PWM Fault Protection A Input (for Channels 1-4)
OCFB	I	PWM Fault Protection B Input (for Channel 5)

Legend: I = Input, O = Output

16.6 REGISTER MAPS

The summaries of the registers associated with the PIC24F output compare module are provided in Table 16-6, Table 16-7 and Table 16-8.

Table 16-6: Output Compare Register Map

File Name	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	All Resets
OCxRS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxxx
OCxR	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxxx
OCxCON	—	—	OCxSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM2	OCM1	OCM0	0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Table 16-7: Timer Register Map

File Name	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	All Resets
TM2R2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxxx
TM3R3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxxx
PR2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FFFF
PR3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FFFF
T2CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T3CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Table 16-8: Interrupt Controller Register Map

File Name	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	All Resets
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	M12C1IF	SI2C1IF	0000
IFS2	—	—	PMPIF	—	—	—	OC5IF	—	IC5IF	IC4IF	IC3IF	—	—	—	SPI2IF	SPF2IF	0000
IEC1	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	—	—	—	—	INT1IE	CNIE	CMIE	M12C1IE	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

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

PIC24F Family Reference Manual

16.7 ELECTRICAL SPECIFICATIONS

16.7.1 AC Characteristics

Figure 16-22: Output Compare Timings

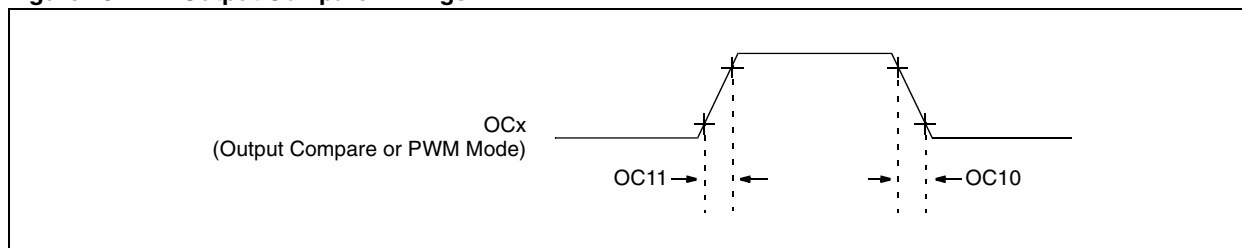


Table 16-9: Output Capture

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
OC11	TccR	OC1 Output Rise Time	—	10	ns	
			—	—	ns	
OC10	TccF	OC1 Output Fall Time	—	10	ns	
			—	—	ns	

Figure 16-23: PWM Module Timing Requirements

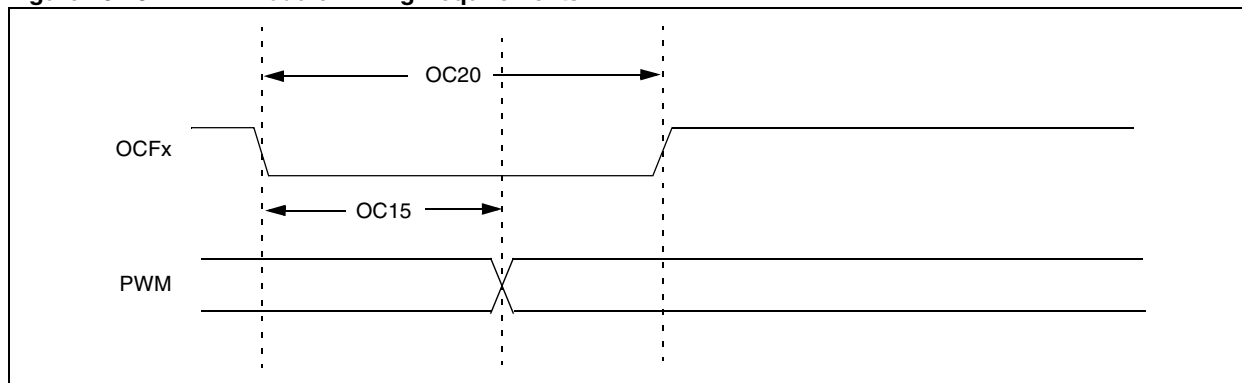


Table 16-10: PWM Timing Requirements

Param. No.	Symbol	Characteristic	Min	Typ [†]	Max	Units	Conditions
OC15	TFD	Fault Input to PWM I/O Change	—	—	25	ns	VDD = 3.0V, -40°C to +85°C
OC20	TFH	Fault Input Pulse Width	50	—	—	ns	VDD = 3.0V, -40°C to +85°C

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

16.8 DESIGN TIPS

Question 1: *The output compare pin stops functioning even when the OCSIDL bit is not set. Why?*

Answer: This is most likely to occur when the TSIDL bit (TxCON<13>) of the associated timer source is set. Therefore, it is the timer that actually goes into Idle mode when the PWRSAV instruction is executed.

Question 2: *Can I use the output compare modules with the selected time base configured for 32-bit mode?*

Answer: No. The T32 bit (TxCON<3>) should be cleared when the timer is used with an output compare module.

16.9 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the PIC24F device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Output Compare module are:

Title	Application Note #
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

Note: Please visit the Microchip web site (www.microchip.com) for additional application notes and code examples for the PIC24F family of devices.

16.10 REVISION HISTORY

Revision A (April 2006)

This is the initial released revision of this document.

NOTES: