Section 13. Output Compare

HIGHLIGHTS

This section of the manual contains the following major topics:

13.1 Introduction 13-2
13.2 Output Compare Registers 13-3
13.3 Output Compare Operation 13-5
13.5 Output Compare Operation in Power-Saving States 13-19
13.6 I/O Pin Control 13-19
13.7 Register Maps 13-19
13.8 Design Tips 13-22
13.9 Related Application Notes 13-23
13.10 Revision History 13-24

13.1 INTRODUCTION

This section describes Output Compare module and its associated operational modes. Figure 13-1 shows how the Output Compare module uses a timer. The module compares the value of the timer with the value of one or two compare registers depending on the operating mode selected. The state of the output pin changes when the timer value matches the compare register value. The Output Compare module generates either a single output pulse or a sequence of output pulses, by changing the state of the output pin on the compare match events. It can also generate interrupts on compare match events.

The Output Compare module has multiple operating modes:

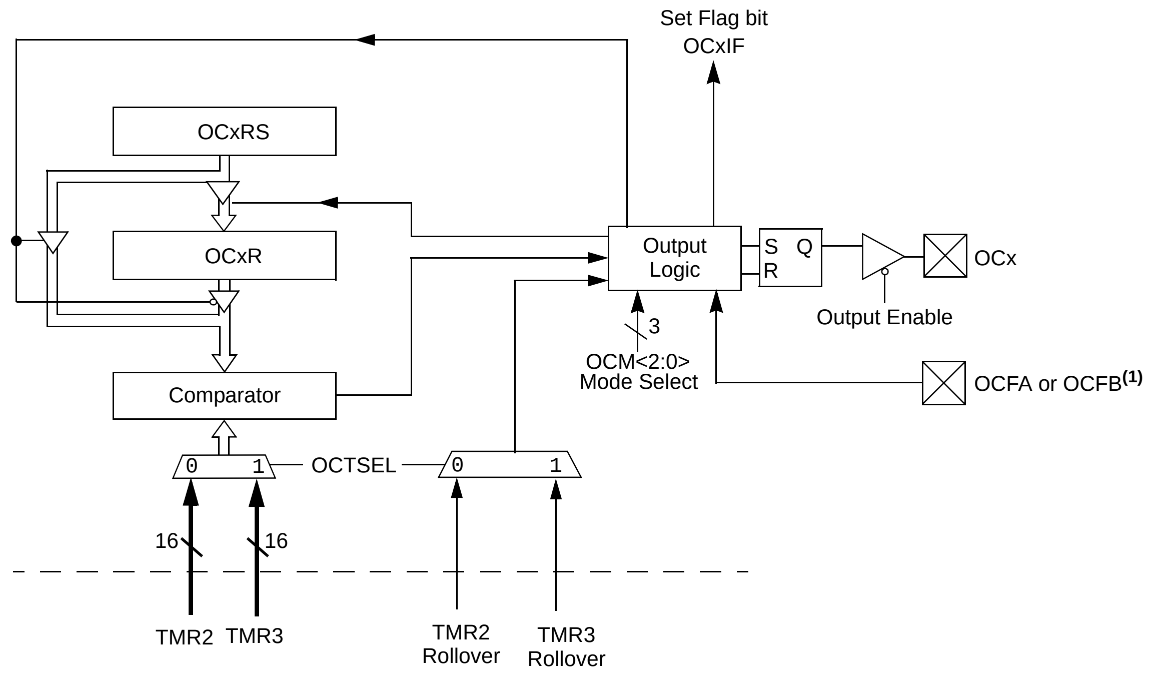
- Active Low One-Shot mode
- Active High One-Shot mode
- Toggle mode
- Delayed One-Shot mode
- Continuous Pulse mode
- PWM mode without fault protection
- PWM mode with fault protection

Note 1: Each dsPIC33F device variant has one or more Output Compare modules. Each Output Compare module can select Timer2 or Timer3 for its time base. For more details, refer to the specific data sheet.

2: An ‘x’ used in the names of pins, control/status bits and registers denotes a particular Output Compare module number (x = 1 to 8).

3: A ‘y’ used in the names of the registers denotes a particular timer (y = 2 or 3).

Figure 13-1: Output Compare Module Block Diagram



Note 1: OCFA pin controls OC1-OC4 modules. OCFB pin controls OC5-OC8 modules.

13.2 OUTPUT COMPARE REGISTERS

Each Output Compare module has the following registers:

- OCxCON: Output Compare Control register
- OCxR: Output Compare register
- OCxRS: Secondary Output Compare register

This section describes these registers.

Register 13-1: 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	<OCM2:0>		
bit 7							bit 0

Legend:	HC = Cleared in Hardware	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
-n = Value at POR	‘0’ = Bit is cleared	

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13 **OCSIDL:** Stop Output Compare x in Idle Mode Control bit
1 = Output compare x halts in CPU Idle mode
0 = Output compare x continues 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 hardware only)
0 = No PWM Fault condition has occurred (this bit is used only when OCM<2:0> = 111)

bit 3 **OCTSEL:** Output Compare x Timer Select bit
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 with fault protection. PWM mode on OCx, Fault pin is enabled
110 = PWM mode without fault protection. PWM mode on OCx, Fault pin is disabled
101 = Continuous Pulse mode. Initialize OCx pin low, generate continuous output pulses on OCx pin
100 = Delayed One-Shot mode. Initialize OCx pin low, generate single output pulse on OCx pin
011 = Toggle mode. Compare event toggles OCx pin
010 = Active High One-Shot mode. Initialize OCx pin high, compare event forces OCx pin low
001 = Active Low One-Shot mode. Initialize OCx pin low, compare event forces OCx pin high
000 = Module Disabled. Output Compare module is disabled

Note: The user software must disable the associated Output Compare module when writing to the output compare control registers to avoid malfunctions.

Register 13-2: OCxR: Output Compare Register

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
Compare Value<15:8>							
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
Compare Value<7:0>							
bit 7							bit 0

Legend:	HC = Cleared in Hardware	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
-n = Value at POR	‘0’ = Bit is cleared	

bit 15-0 **Compare Value<15:0>**

.

Register 13-3: OCxRS: Secondary Output Compare Register

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
Secondary Compare Value<15:8>							
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
Secondary Compare Value<7:0>							
bit 7							bit 0

Legend:	HC = Cleared in Hardware	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
-n = Value at POR	‘0’ = Bit is cleared	

bit 15-0 **Secondary Compare Value<15:0>**

13.3 OUTPUT COMPARE OPERATION

13.3.1 Timer Selection

The Output Compare module can select either Timer2 or Timer3 for its time base. The timer resource is selected by configuring the Output Compare Timer Select (OCTSEL) bit in the Output Compare Control (OCxCON<3>) register.

The selected timer starts from zero and increments on every clock until it reaches the value in the Period Register (PRy). When the period value is reached, the timer resets to zero and starts incrementing once again. The timers can be clocked using an internal clock source (FOSC/2) or a synchronized external clock source applied at the TxCK pin.

13.3.2 Output Compare Modes

Configure the Output Compare modes by setting the appropriate Output Compare mode (OCM<2:0>) bits in the Output Compare Control (OCxCON<2:0>) register. Table 13-1 lists the different bit settings for the Output Compare modes. Figure 13-2 illustrates the output compare operation for various modes.

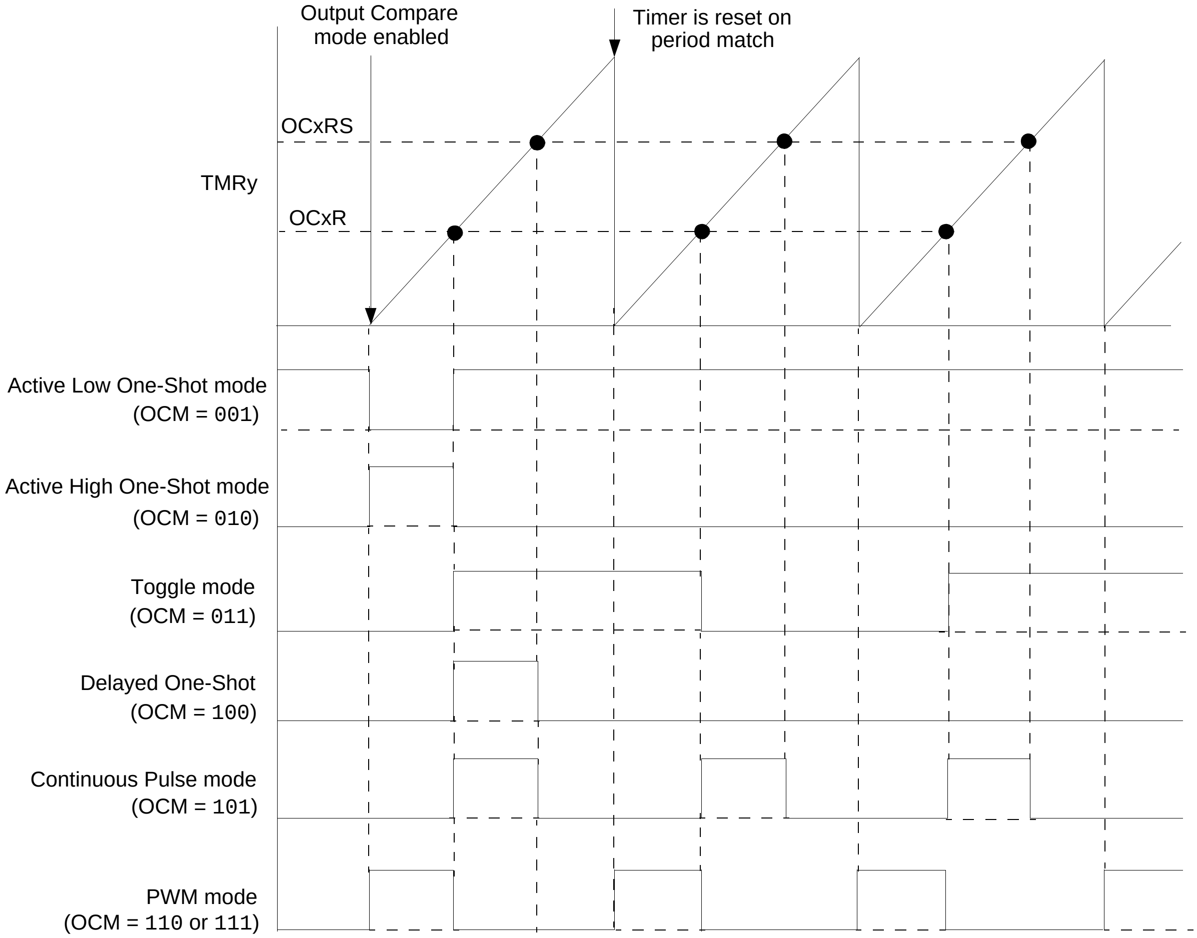
Note: The user software must disable the associated timer when writing to the output compare control registers to avoid malfunctions.

Table 13-1: Output Compare Modes

OCM<2:0>	Mode	OCx Pin Initial State	OCx Interrupt Generation
000	Module Disabled	Controlled by GPIO register	—
001	Active Low One-Shot mode	0	OCx Rising edge
010	Active High One-Shot mode	1	OCx Falling edge
011	Toggle mode	Current output is maintained	OCx Rising and Falling edge
100	Delayed One-Shot mode	0	OCx Falling edge
101	Continuous Pulse mode	0	OCx Falling edge
110	PWM mode without fault protection	0, if OCxR is zero 1, if OCxR is non-zero	No interrupt
111	PWM mode with fault protection	0, if OCxR is zero 1, if OCxR is non-zero	OCFA[1] Falling edge for OC1 to OC4 OCFB[1] Falling edge for OC5 to OC8

Note 1: OCFA and OCFB pins are output compare fault input pins. The OCFA pin is associated with the Output Compare module 1 to 4. The OCFB pin is associated with the Output Compare module 5 to 8.

Figure 13-2: Output Compare Operation



13.3.2.1 ACTIVE LOW ONE-SHOT MODE

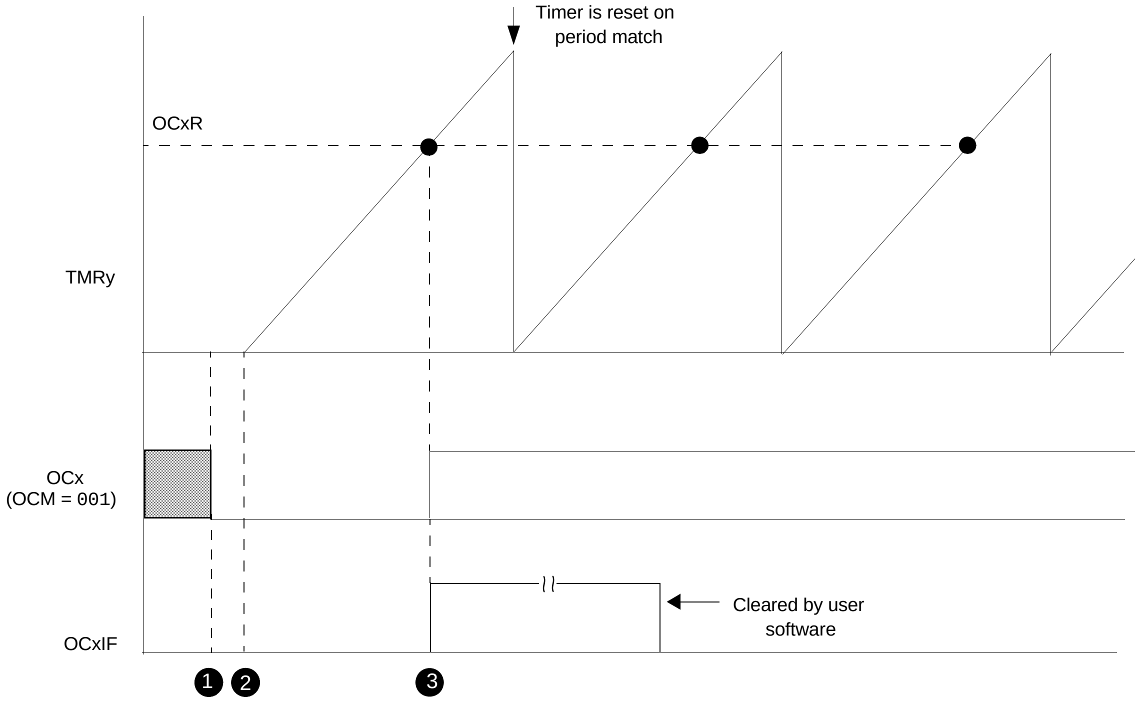
Active Low One-Shot mode provides an active low output pulse asserted once. The duration of the pulse is specified through the Output Compare (OCxR) register. The one-shot pulse can be re-triggered by writing to the Output Compare Control (OCxCON) register.

When the Active Low One-Shot mode is enabled:

1. The Output Compare (OCx) pin is driven low immediately.
2. The OCx pin is driven high on a compare match between the Timer and Output Compare (OCxR) register.
3. An Output Compare interrupt is generated on the rising edge of the OCx pin.

Figure 13-3 illustrates the Active Low One-Shot mode of operation. Example 13-1 illustrates the code sequence to set up the Output Compare module in Active Low One-Shot mode.

Figure 13-3: Active Low One-Shot Mode



1. Active Low One-Shot mode is enabled, the OCx pin is driven low immediately.
2. Timer is enabled and starts incrementing.
3. On a compare match, the OCx pin is driven high and an Output Compare interrupt flag is set.

Example 13-1: Setup for Output Compare Module in Active Low One-Shot mode

```
// Initialize Output Compare Module
OC1CONbits.OCM = 0b000; // Disable Output Compare Module
OC1CONbits.OCTSEL = 0;  // Select Timer 2 as output compare time base
OC1R = 100;             // Load the Compare Register Value
IPC0bits.OC1IP = 0x01;  // Set Output Compare 1 Interrupt Priority Level
IFS0bits.OC1IF = 0;     // Clear Output Compare 1 Interrupt Flag
IEC0bits.OC1IE = 1;     // Enable Output Compare 1 interrupt
OC1CONbits.OCM = 0b001; // Select the Output Compare mode

 // Initialize and enable Timer2

 /* Example code for Output Compare 1 ISR*/
 void __attribute__((__interrupt__)) _OC1Interrupt( void )
 {
 /* Interrupt Service Routine code goes here */
 IFS0bits.OC1IF = 0;    // Clear OC1 interrupt flag
}
```

13.3.2.2 ACTIVE HIGH ONE-SHOT MODE

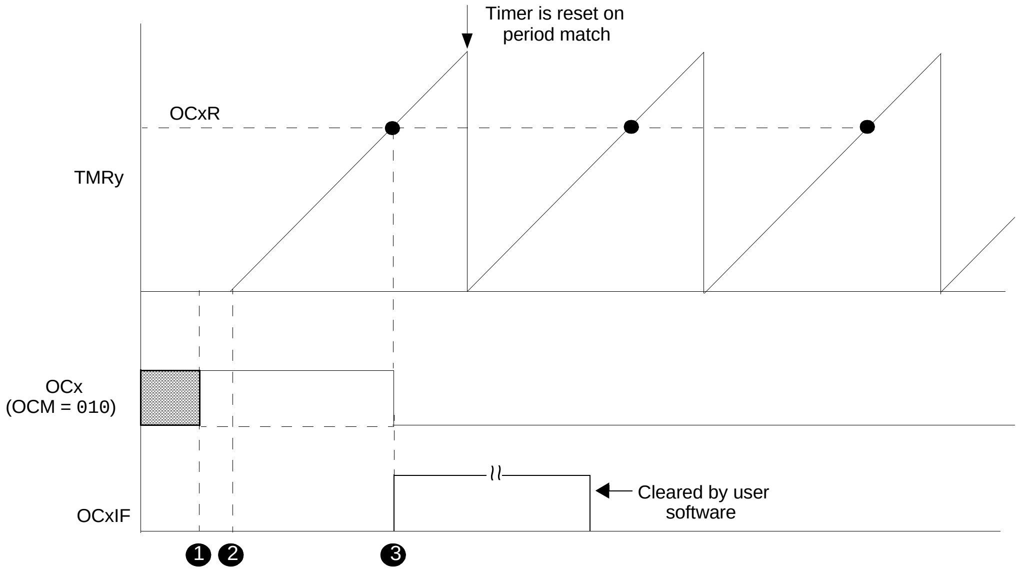
Active High One-Shot mode provides an active high output pulse asserted once. The duration of the pulse is specified through the Output Compare (OCxR) register. The one-shot pulse can be re-triggered by writing to the Output Compare Control (OCxCON) register.

When Active High One-Shot mode is enabled:

1. The Output Compare (OCx) pin is driven high immediately.
2. The OCx pin is driven low on a compare match between the timer and the Output Compare (OCxR) register.
3. An output compare interrupt is generated on the falling edge of the OCx pin.

Figure 13-4 illustrates the Active High One-Shot mode operation. Example 13-2 illustrates the code sequence to set up Output Compare module in Active High One-Shot mode.

Figure 13-4: Active High One-Shot Mode



1. Active High One-Shot mode is enabled, the OCx pin is driven high immediately.
2. Timer is enabled and starts incrementing.
3. On a compare match the OCx pin is driven low and an Output Compare interrupt is generated.

Example 13-2: Setup for Output Compare Module in Active High One-Shot mode

```
// Initialize Output Compare Module
OC1CONbits.OCM = 0b000; // Disable Output Compare Module
OC1CONbits.OCTSEL = 0;  // Select Timer 2 as output compare time base
OC1R = 100;             // Load the Compare Register Value
IPC0bits.OC1IP = 0x01;  // Set Output Compare 1 Interrupt Priority Level
IFS0bits.OC1IF = 0;     // Clear Output Compare 1 Interrupt Flag
IEC0bits.OC1IE = 1;     // Enable Output Compare 1 interrupt
OC1CONbits.OCM = 0b010; // Select the Output Compare mode

// Initialize and enable Timer2

/* Example code for Output Compare 1 ISR*/
void __attribute__((__interrupt__)) _OC1Interrupt( void )
{
/* Interrupt Service Routine code goes here */
IFS0bits.OC1IF = 0;     // Clear OC1 interrupt flag
}
```

13.3.2.3 TOGGLE MODE

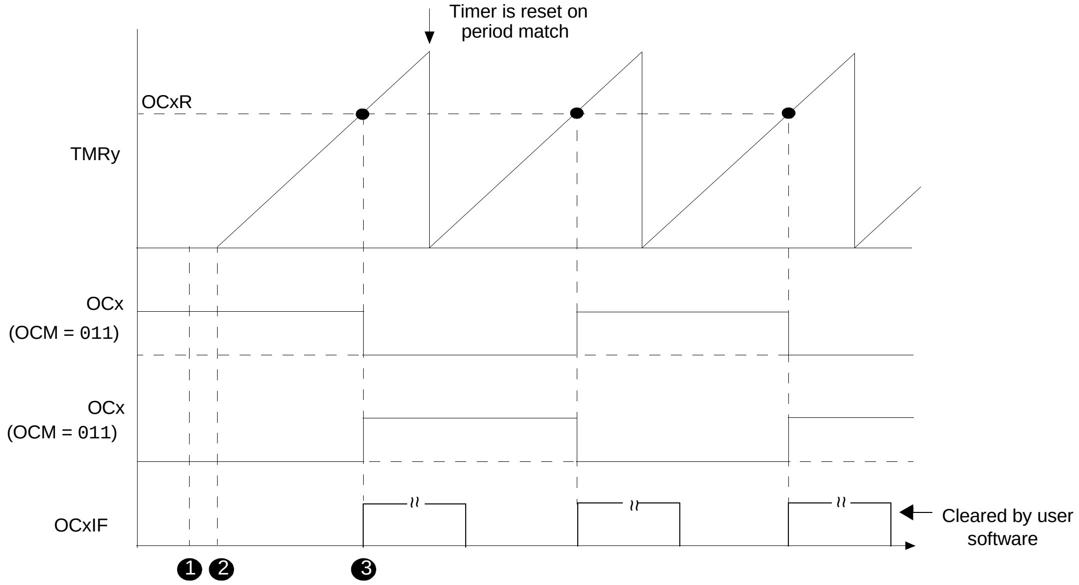
Toggle mode is generally used to generate a 50% duty cycle square waveform. When Toggle mode is enabled, the internal output signal from the Output Compare module is driven onto the OCx pin.

- The initial state for Toggle mode can be set to logic ‘1’ by selecting Active High One-Shot mode before configuring Toggle mode
- Initial state for Toggle mode can be set to logic ‘0’ by selecting Active Low One-Shot mode before configuring Toggle mode

The OCx pin is toggled on every compare match between the timer and the OCxR register. An output compare interrupt is generated on the rising and falling edges of the OCx pin.

Figure 13-5 illustrates Toggle mode operation. Example 13-3 illustrates the code sequence to set up Output Compare module in the Toggle mode.

Figure 13-5: Toggle Mode



1. Toggle mode is enabled, the OCx pin maintains the current state.
2. Timer is enabled and starts incrementing.
3. On a compare match the OCx pin state is toggled and an output compare interrupt flag is set.

Example 13-3: Setup Output Compare Module in Toggle mode

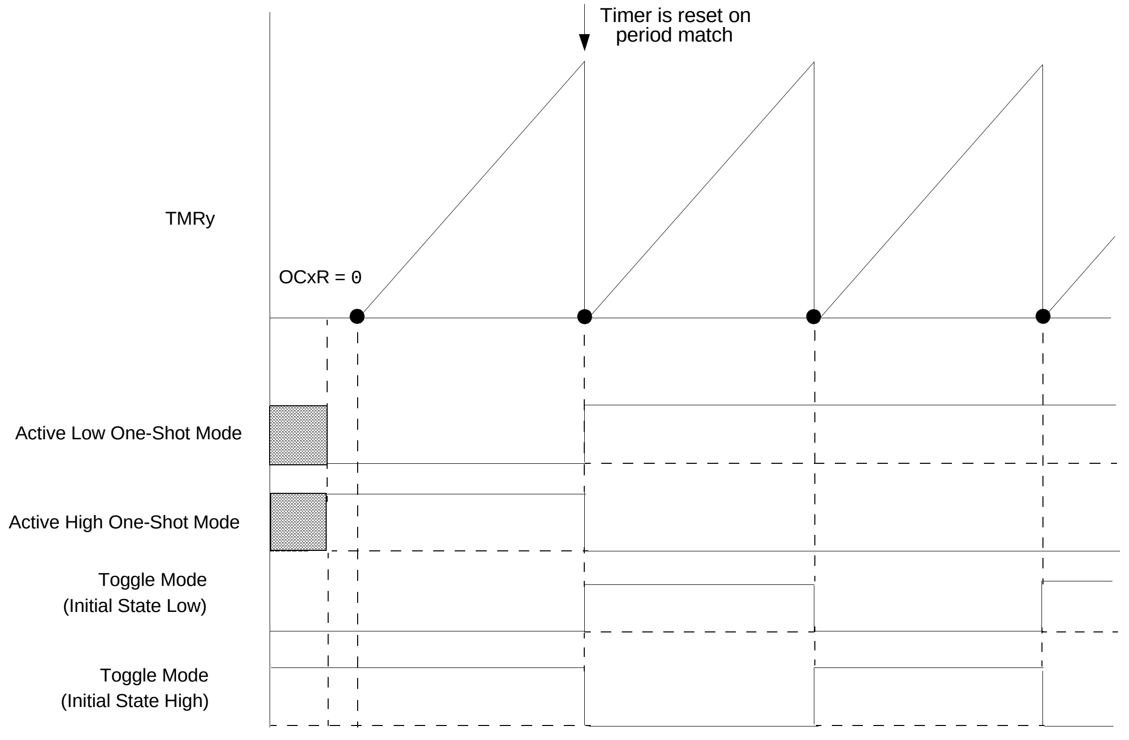
```
// Initialize Output Compare Module
OC1CONbits.OCM = 0b000;  // Disable Output Compare Module
OC1CONbits.OCM = 0b010;  // Define Initial State for OC1 Pin (High if OCM=0b010)
OC1CONbits.OCM = 0b000;  // Disable Output Compare Module
OC1CONbits.OCTSEL = 0;   // Select Timer2 as output compare time base
OC1R= 100;               // Load the Compare Register Value
IPC0bits.OC1IP = 0x01;   // Set Output Compare 1 Interrupt Priority Level
IFS0bits.OC1IF = 0;      // Clear Output Compare 1 Interrupt Flag
IEC0bits.OC1IE = 1;      // Enable Output Compare 1 interrupt
OC1CONbits.OCM = 0b011;  // Select the Output Compare mode

// Initialize and enable Timer2

/* Example code for Output Compare 1 ISR*/
void __attribute__((__interrupt__)) _OC1Interrupt( void )
{
/* Interrupt Service Routine code goes here */
IFS0bits.OC1IF = 0;    // Clear OC1 interrupt flag
}
```

13.3.2.4 SPECIAL CASES FOR ONE-SHOT AND TOGGLE MODES

When the value of the Output Compare register is zero (OCxR = 0), the compare match occurs only on the second timer cycle. Figure 13-6 illustrates the output compare operation under special cases for the following modes:

- Active Low One-Shot mode
- Active High One-Shot mode
- Toggle mode

Figure 13-6: Special Cases for One-Shot and Toggle Modes


Timer is reset on period match
TMRy
OCxR = 0
Active Low One-Shot Mode
Active High One-Shot Mode
Toggle Mode (Initial State Low)
Toggle Mode (Initial State High)

13.3.2.5 DELAYED ONE-SHOT MODE

Delayed One-Shot mode provides a delayed active high output pulse asserted once. The Output Compare (OCxR) register value controls the delay for the one-shot pulse. The pulse width is specified through the Secondary Output Compare (OCxRS) register. The one-shot pulse can be re-triggered by writing to the Output Compare Control (OCxCON) register.

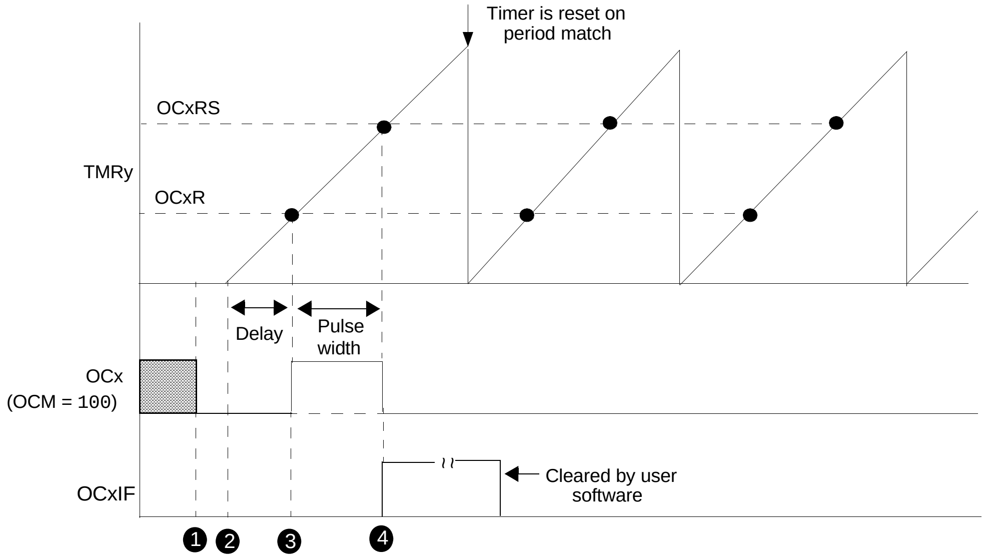
When Delayed One-Shot mode is enabled:

1. The Output Compare (OCx) pin is driven low immediately.
2. The timer value is compared with the Output Compare (OCxR) register and the OCx pin is driven high on a compare match.
3. The timer value is compared with the Secondary Output Compare (OCxRS) register and the OCx pin is driven low on a compare match.
4. An output compare interrupt is generated on the falling edge of the OCx pin.

The Output Compare module halts further comparison until it is re-triggered.

Figure 13-7 illustrates Delayed One-shot operation. Example 13-4 illustrates the code sequence to set up Output Compare module in the Delayed One-Shot mode.

Figure 13-7: Delayed One-Shot mode



1. Delayed One-Shot mode is enabled, the OCx pin is driven low immediately.
2. Timer is enabled and starts incrementing.
3. On compare match with OCxR register, the OCx pin is driven high.
4. On compare match with OCxRS register, OCx pin is driven low and an Output Compare interrupt flag is set.

Example 13-4: Setup for Output Compare Module in Delayed One-Shot pulse mode

```
// Initialize Output Compare Module
OC1CONbits.OCM = 0b000;  // Disable Output Compare Module
OC1CONbits.OCTSEL = 0;   // Select Timer 2 as output compare time base
OC1R = 100;              // Load the Compare Register Value for rising edge
OC1RS = 200;             // Load the Compare Register Value for falling edge
IPC0bits.OC1IP = 0x01;   // Set Output Compare 1 Interrupt Priority Level
IFS0bits.OC1IF = 0;      // Clear Output Compare 1 Interrupt Flag
IEC0bits.OC1IE = 1;      // Enable Output Compare 1 interrupt
OC1CONbits.OCM = 0b100;  // Select the Output Compare mode

// Initialize and enable Timer2

/* Example code for Output Compare 1 ISR*/
void __attribute__((__interrupt__)) _OC1Interrupt( void )
{
/* Interrupt Service Routine code goes here */
IFS0bits.OC1IF = 0;      // Clear OC1 interrupt flag
}
```

13.3.2.6 CONTINUOUS PULSE MODE

Continuous Pulse mode provides an output pulse on every timer cycle. This mode is useful to generate fixed duty cycle output. Figure 13-8 shows the block diagram of the Output Compare module in Continuous Pulse mode.

When Continuous Pulse mode is enabled:

1. The Output Compare (OCx) pin is driven low immediately.
2. The timer value is compared with the Output Compare (OCxR) register and the OCx pin is driven high on a compare match.
3. The timer value is compared with the Secondary Output Compare (OCxRS) register and the OCx pin is driven low on a compare match.
4. An output compare interrupt is generated on the falling edge of the OCx pin.

Figure 13-9 illustrates the Continuous Pulse mode of operation.

Figure 13-8: Continuous Pulse Mode Block Diagram

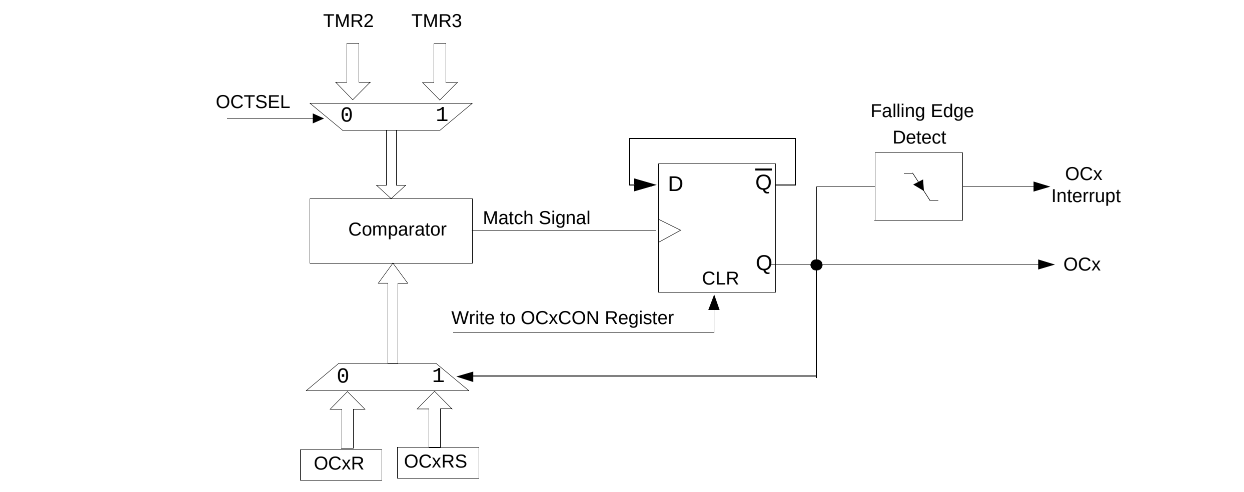
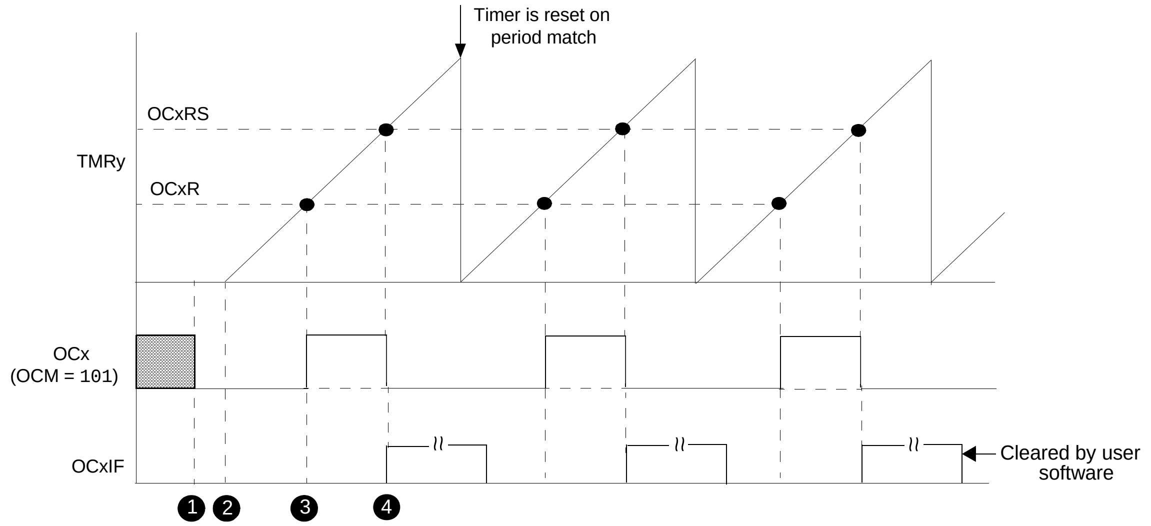


Figure 13-9: Continuous Pulse Mode



1. Continuous Pulse mode is enabled, the OCx pin is driven low immediately.
2. Timer is enabled and it starts incrementing.
3. On compare match with OCxR register, the OCx pin is driven high.
4. On compare match with OCxRS register, OCx pin is driven low.

Example 13-4 illustrates the code sequence to set up the Output Compare module in Continuous Pulse mode.

Example 13-5: Setup Output Compare module in Continuous Pulse mode

```
// Initialize Output Compare Module
OC1CONbits.OCM = 0b000;  // Disable Output Compare Module
OC1CONbits.OCTSEL = 0;   // Select Timer 2 as output compare time base
OC1R = 100;              // Load the Compare Register Value for rising edge
OC1RS = 200;             // Load the Compare Register Value for falling edge
IPC0bits.OC1IP = 0x01;   // Set Output Compare 1 Interrupt Priority Level
IFS0bits.OC1IF = 0;      // Clear Output Compare 1 Interrupt Flag
IEC0bits.OC1IE = 1;      // Enable Output Compare 1 interrupt
OC1CONbits.OCM = 0b101;  // Select the Output Compare mode

// Initialize and enable Timer2

/* Example code for Output Compare 1 ISR*/
void __attribute__((__interrupt__)) _OC1Interrupt( void )
{
/* Interrupt Service Routine code goes here */
IFS0bits.OC1IF = 0;      // Clear OC1 interrupt flag
}
```

13.3.2.7 SPECIAL CASES FOR DELAYED ONE-SHOT AND CONTINUOUS PULSE MODE

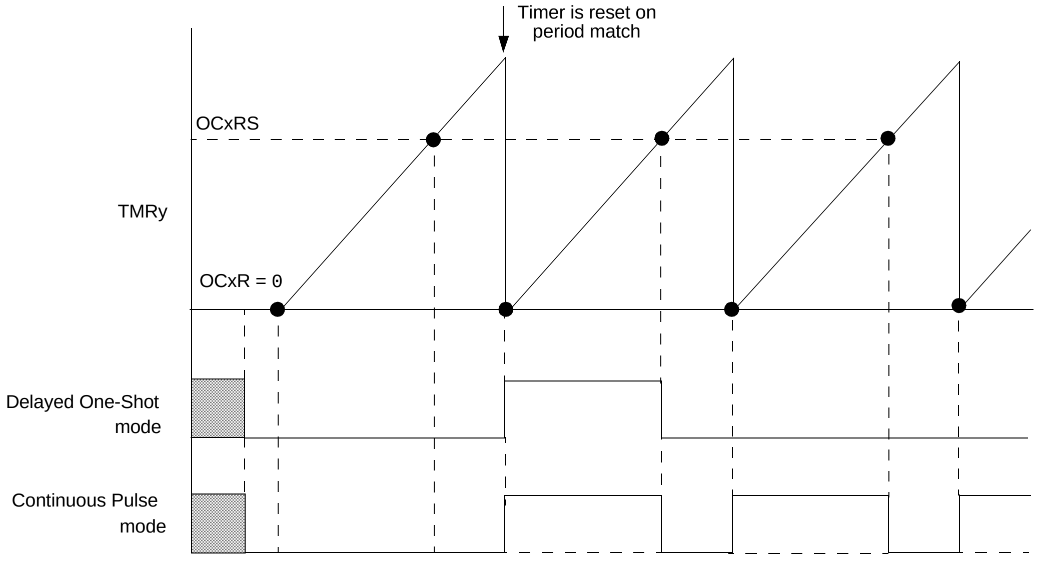
For proper operation in Delayed One-Shot and Continuous Pulse modes, the OCxR, OCxRS and PRy values must have the following relationships:

- OCxRS ≥ OCxR
- PRy >=OCxRS

When the value of Output Compare register is zero (OCxR = `0`) a compare match occurs only on the second timer cycle. Figure 13-6 illustrates the Output Compare operation under special cases for the following modes:

- Delayed One-Shot mode
- Continuous Pulse mode

Figure 13-10: Special Cases for Delayed One-Shot and Continuous Mode



13.3.2.8 PWM MODE WITHOUT FAULT PROTECTION

PWM mode is used to generate variable duty cycle output. The PWM duty cycle is specified through Secondary Output Compare (OCxRS) register, and it acts as a shadow register for the Output Compare (OCxR) register. This prevents glitches in the PWM output. In PWM mode, the Output Compare (OCxR) register is a read-only compare register. Figure 13-11 shows the block diagram of the Output Compare module in PWM mode.

When the PWM mode is enabled, the OCx pin is:

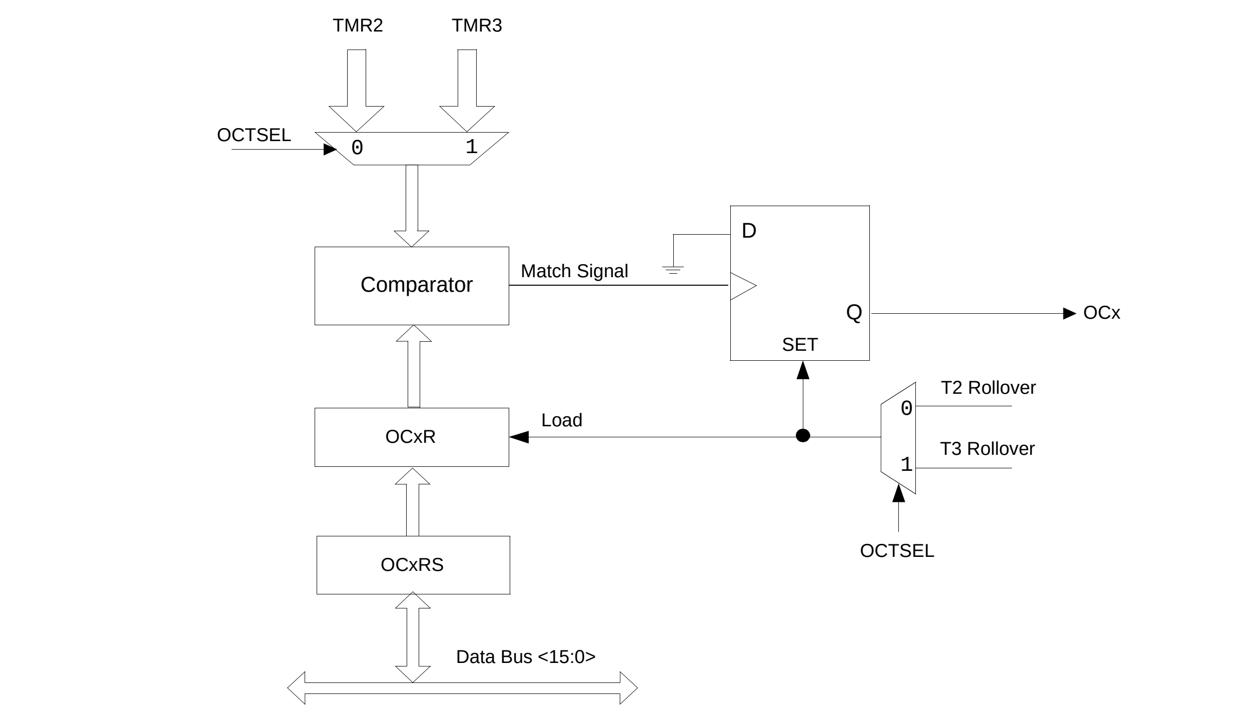
- Driven high if the OCxR register value is non-zero (refer to Case 1 in Figure 13-12)
- Driven low if the OCxR register value is zero (refer to Case 2 in Figure 13-12)

When a selected timer is enabled, it starts incrementing until it reaches the value in the period register. The Compare Register (OCxR) value is constantly compared with the timer value. When a match occurs, the OCx pin is driven low.

On a timer rollover, the OCxRS value is loaded into the OCxR register and the OCx pin is:

- Driven high if the OCxR register value is non-zero
- Driven low if the OCxR register value is zero

Figure 13-11: PWM Mode Block Diagram



13.3.2.8.1 PWM Period

The PWM period is specified by writing to PRy, the TMRy Period register. Equation 13-1 calculates the PWM period.

Equation 13-1: Calculating the PWM Period

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

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

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

13.3.2.8.2 PWM Duty Cycle

The PWM duty cycle is specified by writing to the OCxRS register. The duty cycle value can be written at any time, but the duty cycle value is not latched onto the OCxR register until timer Reset on a period match. This provides a double buffer for the PWM duty cycle and is essential for glitch-free PWM operation. In PWM mode, OCxR is a read-only register.

Some important boundary parameters of the PWM duty cycle include:

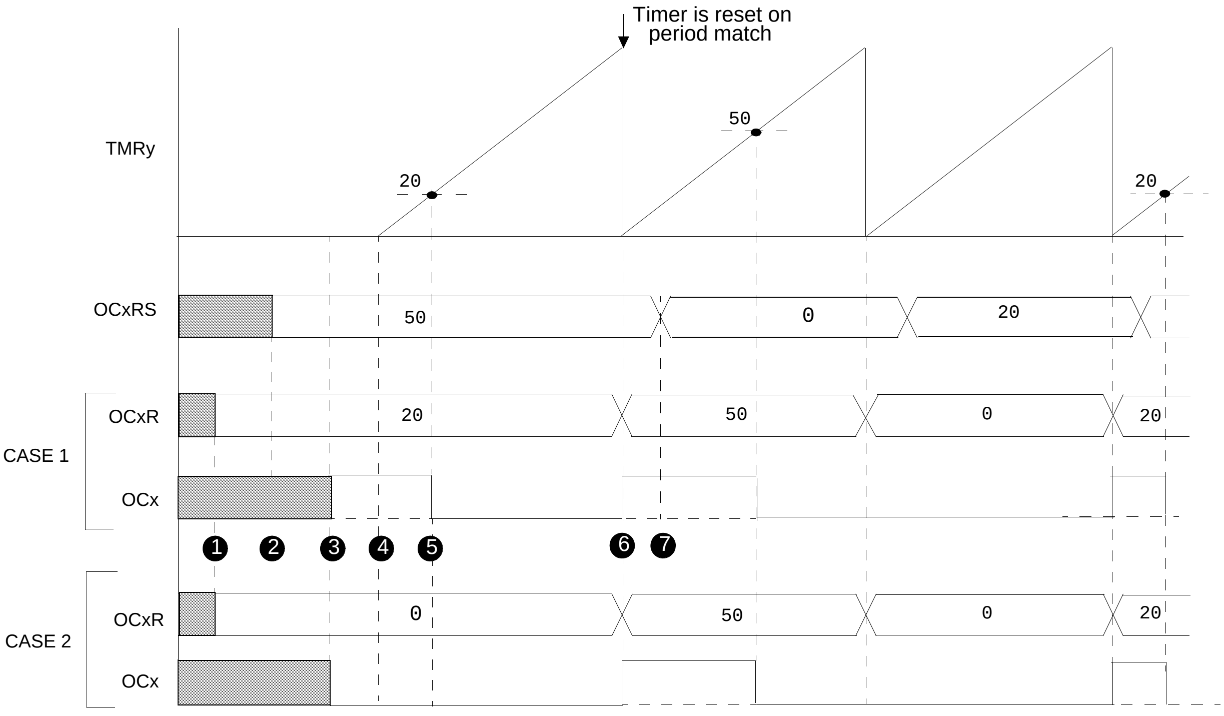
- If the duty cycle register, OCxR, is loaded with 0000h, the OCx pin remains low (0% duty cycle)
- If the OCxR register is greater than PRy (Timer Period register), the pin remains high (100% duty cycle)
- If the OCxR register is equal to PRy, the OCx pin is low for one time base count value and high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

The PWM resolution depends on PWM frequency and the timer clock frequency. The timer clock is derived from the internal clock (FCY) divided by a programmable prescaler. For more details, refer to **Section 11. "Timers"**.

Equation 13-2: Calculation for PWM Resolution

$$\text{PWM Resolution (bits)} = \log_2 \left(\frac{\text{PWM Frequency}}{\text{Timer Clock Frequency}} \right) \text{ bits}$$

Figure 13-12: PWM Mode Operation



1. The duty cycle for first PWM cycle is written to OCxR register before enabling the PWM mode.
2. The duty cycle for second PWM cycle is written to OCxRS register.
3. PWM mode is enabled, OCx pin is driven low if OCxR is zero. OCx pin is driven high if OCxR is non-zero.
4. Timer is enabled and starts incrementing.
5. On a compare match, the OCx pin is driven low.
6. On timer rollover, OCxRS value is loaded into OCxR register. OCx pin is driven high if OCxR is zero. OCx pin is driven low if OCxR is non-zero.
7. The duty cycle for third PWM cycle is written to the OCxRS register.

Example 13-6 illustrates the code sequence to set up the Output Compare module in PWM mode.

Example 13-6: Setup for Output Compare Module in PWM Mode

```
// Initialize Output Compare Module
OC1CONbits.OCM = 0b000; // Disable Output Compare Module
OC1R = 100;             // Write the duty cycle for the first PWM pulse
OC1RS = 200;            // Write the duty cycle for the second PWM pulse
OC1CONbits.OCTSEL = 0;  // Select Timer 2 as output compare time base
OC1R = 100;             // Load the Compare Register Value
OC1CONbits.OCM = 0b110; // Select the Output Compare mode

// Initialize and enable Timer2
T2CONbits.TON = 0;      // Disable Timer
T2CONbits.TCS = 0;      // Select internal instruction cycle clock
T2CONbits.TGATE = 0;    // Disable Gated Timer mode
T2CONbits.TCKPS = 0b00; // Select 1:1 Prescaler
TMR2 = 0x00;            // Clear timer register
PR2 = 500;              // Load the period value

IPC1bits.T2IP = 0x01;   // Set Timer 2 Interrupt Priority Level
IFS0bits.T2IF = 0;      // Clear Timer 2 Interrupt Flag
IEC0bits.T2IE = 1;      // Enable Timer 2 interrupt
T2CONbits.TON = 1;      // Start Timer

/* Example code for Timer 2 ISR*/
void __attribute__((__interrupt__)) _T2Interrupt( void )
{
/* Interrupt Service Routine code goes here */
OC1RS = 300;            // Write Duty Cycle value for next PWM cycle
IFS0bits.T2IF = 0;      // Clear Timer 2 interrupt flag
}
```

13.3.2.9 PWM MODE WITH FAULT PROTECTION

When the PWM mode with fault protection is enabled, the module operates similarly to PWM mode. In addition, it tri-states the PWM output on a Fault condition.

Fault protection is provided through the OCFA and OCFB pins. The OCFA pin is associated with Output Compare modules 1 to 4, and the OCFB pin is associated with Output Compare modules 5 to 8.

If a logic ‘0’ is detected on the OCFA/OCFB pin, the selected Output Compare pin is tri-stated and the Output Compare Fault (OCFLT) flag (OCxCON<4>) is set. Then the Output Compare Interrupt Flag (OCxIF) is set and an interrupt is generated, if enabled.

The output is tri-stated and OCxIF is set until the external Fault condition is removed and the PWM mode is re-enabled by writing to the appropriate mode bits, OCM<2:0> (OCxCON<2:0>).

> **Note:** The user application can provide a pull-down or pull-up resistor on the OCx pin to allow a desired state if a Fault condition occurs.

13.4 OUTPUT COMPARE OPERATION WITH DMA

Some dsPIC33F family devices include a Direct Memory Access (DMA) module, which allows data transfer from data memory to the Output Compare module without CPU intervention. Consult the specific dsPIC33F device data sheet to see if DMA is present on your particular device. For more information on the DMA module, refer to **Section 22. Direct Memory Access (DMA).**

The DMA channel must be initialized with the following:

- Initialize the DMA Channel Peripheral Address (DMAxPAD) register with the address of the Output Compare (OCxR) register or the Secondary Output Compare (OCxRS) register
- Set the Transfer Direction (DIR) bit in DMA Control (DMAxCON<13>) register. In this condition, data is read from the dual port DMA memory and written to the peripheral special function register
- The DMA Request Source Selection (IRQSEL<6:0>) bits in the DMA Request (DMAxREQ<6:0>) register must select the DMA transfer request source

Example 13-7 provides sample code that modulates the PWM duty cycle without CPU intervention. The duty cycle values stored in an array are transferred to OCxRS register on every timer interrupt.

Example 13-7: Code to Modulate the PWM Duty Cycle Without CPU Intervention

```
// Initialize Output Compare Module in PWM mode
OC1CONbits.OCM = 0b000;   // Disable Output Compare Module
OC1R=100;                 // Write the duty cycle for the first PWM pulse
OC1RS=200;                // Write the duty cycle for the second PWM pulse
OC1CONbits.OCTSEL = 0;    // Select Timer 2 as output compare time base
OC1R= 100;                // Load the Compare Register Value
OC1CONbits.OCM = 0b110;   // Select the Output Compare mode

// Initialize Timer2
T2CONbits.TON = 0;        // Disable Timer
T2CONbits.TCS = 0;        // Select internal instruction cycle clock
T2CONbits.TGATE = 0;      // Disable Gated Timer mode
T2CONbits.TCKPS = 0b00;   // Select 1:1 Prescaler
TMR2 = 0x00;              // Clear timer register
PR2 = 500;                // Load the period value

// Define a Buffer in DMA RAM to store duty cycle information
unsigned int BufferA[256] __attribute__((space(dma)));

// Setup and Enable DMA Channel
DMA0CONbits.AMODE = 0b00; // Register indirect with post increment
DMA0CONbits.MODE = 0b00;  // Continuous, Ping-Pong mode Disabled
DMA0CONbits.DIR = 0;      // Peripheral to RAM
DMA0PAD = (int)&OC1RS;    // Address of the secondary output compare register
DMA0REQ = 7;              // Select Timer2 interrupt as DMA request source
DMA0CNT = 255;            // Number of words to buffer.
DMA0STA = __builtin_dmaoffset(&BufferA);

IFS0bits.DMA0IF = 0;      // Clear the DMA interrupt flag
IEC0bits.DMA0IE = 1;      // Enable DMA interrupt
DMA0CONbits.CHEN = 1;     // Enable DMA channel

// Enable Timer
T2CONbits.TON = 1;        // Start Timer

//  DMA Interrupt Handler
void __attribute__((__interrupt__)) _DMA0Interrupt(void)
{
IFS0bits.DMA0IF = 0;      // Clear the DMA0 Interrupt Flag
}
```

13.5 OUTPUT COMPARE OPERATION IN POWER-SAVING STATES

13.5.1 Output Compare Operation in Sleep Mode

The Output Compare module does not operate when the device is in Sleep mode. The OCx pin is driven to the default initial state. Table 13-1 provides the initial states for different Output Compare modes.

When the Output Compare module is configured for PWM mode with fault protection, the Fault condition tri-states the output in Sleep mode. If a fault is detected, the OCx pin is tri-stated, and the OCFLT bit (OCxCON<4>) is set. An interrupt is not generated at fault occurrence; however, the interrupt is queued and occurs when the part wakes up.

13.5.2 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 Output Compare Stop-in Idle (OCSIDL) bit (OCxCON<13>) determines whether the module stops or continues operation in Idle mode.

- If OCSIDL = `0`, the module continues operation in Idle mode, providing full functionality of Output Compare module. The selected timer must also be enabled to operate in Idle mode.
- If OCSIDL = `1`, the module stops operation in Idle mode. The module performs the same functions when stopped in Idle mode as it does for Sleep mode. Refer to **Section 13.5.1 “Output Compare Operation in Sleep Mode”**.

13.6 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 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 the special PWM mode does not configure the OCFx Fault pin as an input.

> **Note:** On some devices, the output compare pin direction bits must be configured for output by clearing the respective TRIS bit. For details, refer to the specific device data sheet.

13.7 REGISTER MAPS

The summaries of the registers associated with the dsPIC33F Output Compare module are provided in Table 13-2, Table 13-3 and Table 13-4.

Table 13-2: Output Compare Register Map

SFR 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
OC1RS	Output Compare 1 Secondary Register																xxxx
OC1R	Output Compare 1 Register																xxxx
OC1CON	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000
OC2RS	Output Compare 2 Secondary Register																xxxx
OC2R	Output Compare 2 Register																xxxx
OC2CON	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000
OC3RS	Output Compare 3 Secondary Register																xxxx
OC3R	Output Compare 3 Register																xxxx
OC3CON	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000
OC4RS	Output Compare 4 Secondary Register																xxxx
OC4R	Output Compare 4 Register																xxxx
OC4CON	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000
OC5RS	Output Compare 5 Secondary Register																xxxx
OC5R	Output Compare 5 Register																xxxx
OC5CON	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000
OC6RS	Output Compare 6 Secondary Register																xxxx
OC6R	Output Compare 6 Register																xxxx
OC6CON	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000
OC7RS	Output Compare 7 Secondary Register																xxxx
OC7R	Output Compare 7 Register																xxxx
OC7CON	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000
OC8RS	Output Compare 8 Secondary Register																xxxx
OC8R	Output Compare 8 Register																xxxx
OC8CON	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000

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

Table 13-3: Timer Register Map

SFR 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
TMR2	Timer2 Register																xxxx
TMR3HLD	Timer3 Holding Register (for 32-bit timer operations only)																xxxx
TMR3	Timer3 Register																xxxx
PR2	Period Register 2																FFFF
PR3	Period Register 3																FFFF
T2CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS<1:0>		T32	—	TCS	—	0000
T3CON	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS<1:0>		—	—	TCS	—	0000

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

Table 13-4: Interrupt Controller Register Map

SFR 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	—	—	—	—	—	—	—	—	—	OC2IF	—	—	—	OC1IF	—	—	0000
IFS1	—	—	—	—	—	OC4IF	OC3IF	—	—	—	—	—	—	—	—	—	0000
IFS2	—	—	—	OC8IF	OC7IF	OC6IF	OC5IF	—	—	—	—	—	—	—	—	—	0000
IEC0	—	—	—	—	—	—	—	—	—	OC2IE	—	—	—	OC1IE	—	—	0000
IEC1	—	—	—	—	—	OC4IE	OC3IE	—	—	—	—	—	—	—	—	—	0000
IEC2	—	—	—	OC8IE	OC7IE	OC6IE	OC5IE	—	—	—	—	—	—	—	—	—	0000
IPC0	—	—	—	—	—	OC1IP<2:0>			—	—	—	—	—	—	—	—	4444
IPC1	—	—	—	—	—	OC2IP<2:0>			—	—	—	—	—	—	—	—	4444
IPC5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4444
IPC6	—	—	—	—	—	OC4IP<2:0>			—	OC3IP<2:0>			—	—	—	—	4444
IPC10	—	OC7IP<2:0>			—	OC6IP<2:0>			—	OC5IP<2:0>			—	—	—	—	4444
IPC11	—	—	—	—	—	—	—	—	—	—	—	—	—	OC8IP<2:0>			4444

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

13.8 DESIGN TIPS

Question 1: ***The Output Compare pin stops functioning even when the OCSIDL bit (OCxCON<13>) is not set. Why?***

Answer: This most likely occurs 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.

13.9 RELATED APPLICATION NOTES

This section lists application notes related to this section of the manual. These application notes may not be written specifically for the dsPIC33F 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^2C™ 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: For additional application notes and code examples for the dsPIC33F device family, visit the Microchip web site (www.microchip.com).

13.10 REVISION HISTORY

Revision A (May 2007)

This is the initial released version of this document.