MICROCHIP

Section 15. Input Capture

HIGHLIGHTS

This section of the manual contains the following topics:

15.1 INTRODUCTION

This section describes the Input Capture module and its associated operational modes. The Input Capture module is used to capture a timer value from one of two selectable time bases on the occurrence of an event on an input pin. The Input Capture features are useful in applications requiring frequency (Time Period) and pulse measurement. Figure 15-1 depicts a simplified block diagram of the Input Capture module.

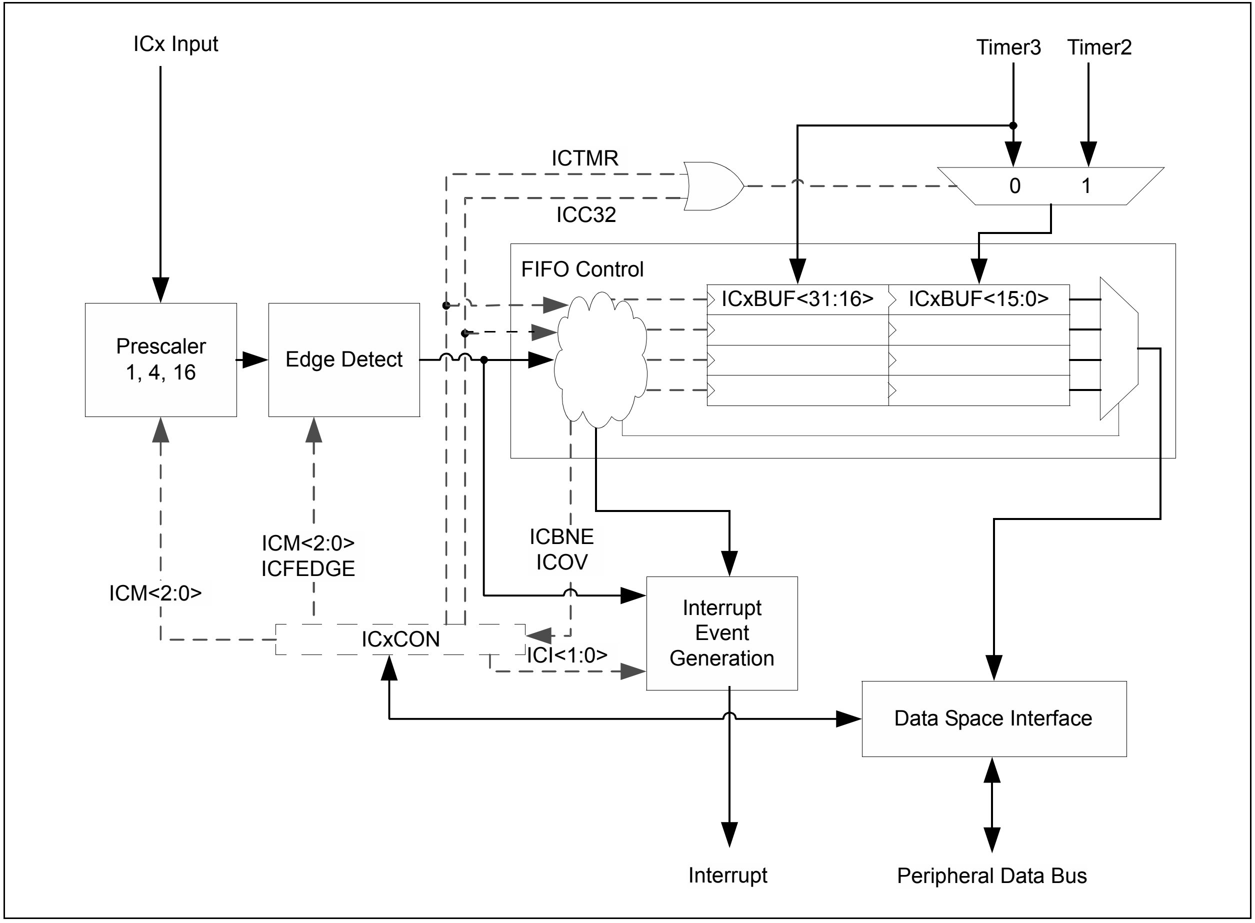
> **Note:** Each PIC32MX device variant may have one or more Input Capture modules. Refer to the specific device data sheet for information on the number of Input Capture modules available in a particular device.
> All Input Capture modules are functionally identical. In this document, an 'x' used in the names of pins, Control/Status bits, and registers denotes the specific Input Capture module.

The Input Capture module has multiple operating modes, which are selected via the ICxCON register. The operating modes include the following:

- Capture timer value on every falling edge of input applied at the ICx pin
- Capture timer value on every rising edge of input applied at the ICx pin
- Capture timer value on every fourth rising edge of input applied at the ICx pin
- Capture timer value on every sixteenth rising edge of input applied at the ICx pin
- Capture timer value on every rising and falling edge of input applied at the ICx pin
- Capture timer value on the specified edge and every edge thereafter

The Input Capture module has a four-level First In First Out (FIFO) buffer. The number of capture events required to generate a CPU interrupt can be selected by the user application. An Input Capture module can also be configured to generate a CPU interrupt on a rising edge of the capture input when the device is in Sleep or Idle mode.

Figure 15-1: Input Capture Module Block Diagram



15.2 INPUT CAPTURE REGISTERS

Each Input Capture module available on PIC32MX devices has the following Special Function Registers (SFRs):

- ICxCON: Input Capture Control Register
- ICxBUF: Input Capture Buffer Register

Each Input Capture module also has the following associated bits for interrupt control:

- Interrupt Enable Control bit (ICxIE)
- Interrupt Flag Status bit (ICxIF)
- Interrupt Priority Control bits (ICxIP)
- Interrupt Subpriority Control bits (ICxIS)

Table 15-1 provides a brief summary of the Input Capture related registers, and is followed by a detailed description of each register.

Table 15-1: Input Capture SFR Summary

Address Offset	Name		Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
0x00	ICxCON[1,2,3]	31:24	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		15:8	ON	FRZ	SIDL	—	—	—	ICFEDGE	ICC32
		7:0	ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>		
0x10	ICxBUF	31:24	ICxBUF<31:24>							
		23:16	ICxBUF<23:16>							
		15:8	ICxBUF<15:8>							
		7:0	ICxBUF<7:0>							

Legend: — = unimplemented, read as ‘0’. Address offset values are shown in hexadecimal.

Note 1: This register has an associated Clear register at an offset of 0x4 bytes. These registers have the same name with CLR appended to the end of the register name (e.g., ICxCONCLR). Writing a ‘1’ to any bit position in the Clear register will clear valid bits in the associated register. Reads from the Clear register should be ignored.

2: This register has an associated Set register at an offset of 0x8 bytes. These registers have the same name with SET appended to the end of the register name (e.g., ICxCONSET). Writing a ‘1’ to any bit position in the Set register will set valid bits in the associated register. Reads from the Set register should be ignored.

3: This register has an associated Invert register at an offset of 0xC bytes. These registers have the same name with INV appended to the end of the register name (e.g., ICxCONINV). Writing a ‘1’ to any bit position in the Invert register will invert valid bits in the associated register. Reads from the Invert register should be ignored.

Register 15-1: ICxCON: Input Capture x Control Register[1,2,3]

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
ON	FRZ	SIDL	—	—	—	ICFEDGE	ICC32
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0
ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>		
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: ('0', '1', x = Unknown)		

bit 31-16 **Reserved:** Write '0'; ignore read

bit 15 **ON:** Input Capture Module Enable bit

1 = Module enabled
0 = Disable and reset module, disable clocks, disable interrupt generation and allow SFR modifications

Note: When using 1:1 PBCLK divisor, the user's software should not read/write the peripheral's SFRs in the SYSCLK cycle immediately following the instruction that clears the module's ON bit.

bit 14 **FRZ:** Freeze in Debug Mode Control bit

1 = Freeze module operation when in Debug mode
0 = Do not freeze module operation when in Debug mode

Note: FRZ is writable in Debug Exception mode only, it is forced to '0' in Normal mode.

bit 13 **SIDL:** Stop in Idle Control bit

1 = Halt in CPU Idle mode
0 = Continue to operate in CPU Idle mode

bit 12-10 **Unimplemented:** Read as '0'

bit 9 **ICFEDGE:** First Capture Edge Select bit (only used in mode 6, ICM<2:0>= 110)

1 = Capture rising edge first
0 = Capture falling edge first

Note 1: This register has an associated Clear register (ICxCONCLR) at an offset of 0x4 bytes. Writing a '1' to any bit position in the Clear register will clear valid bits in the associated register. Reads from the Clear register should be ignored.

2: This register has an associated Set register (ICxCONSET) at an offset of 0x8 bytes. Writing a '1' to any bit position in the Set register will set valid bits in the associated register. Reads from the Set register should be ignored.

3: This register has an associated Invert register (ICxCONINV) at an offset of 0xC bytes. Writing a '1' to any bit position in the Invert register will invert valid bits in the associated register. Reads from the Invert register should be ignored.

Register 15-1: ICxCON: Input Capture x Control Register[1,2,3] (Continued)

bit 8 **ICC32:** 32-bit Capture Select bit

1 = 32-bit timer resource capture
0 = 16-bit timer resource capture

bit 7 **ICTMR:** Timer Select bit (Does not affect timer selection when ICxC32 (ICxCON<8>) is '1')

0 = Timer3 is the counter source for capture
1 = Timer2 is the counter source for capture

bit 6-5 **ICI<1:0>:** Interrupt Control bits

11 = Interrupt on every fourth capture event
10 = Interrupt on every third capture event
01 = Interrupt on every second capture event
00 = Interrupt on every capture event

bit 4 **ICOV:** Input Capture Overflow Status Flag bit (read-only)

1 = Input capture overflow occurred
0 = No input capture overflow occurred

bit 3 **ICBNE:** Input Capture Buffer Not Empty Status bit (read-only)

1 = Input capture buffer is not empty; at least one more capture value can be read
0 = Input capture buffer is empty

bit 2-0 **ICM<2:0>:** Input Capture Mode Select bits

111 = Interrupt-Only mode (only supported while in Sleep mode or Idle mode)
110 = Simple Capture Event mode – every edge, specified edge first and every edge thereafter
101 = Prescaled Capture Event mode – every sixteenth rising edge
100 = Prescaled Capture Event mode – every fourth rising edge
011 = Simple Capture Event mode – every rising edge
010 = Simple Capture Event mode – every falling edge
001 = Edge Detect mode – every edge (rising and falling)
000 = Capture Disable mode

Note 1: This register has an associated Clear register (ICxCONCLR) at an offset of 0x4 bytes. Writing a '1' to any bit position in the Clear register will clear valid bits in the associated register. Reads from the Clear register should be ignored.

2: This register has an associated Set register (ICxCONSET) at an offset of 0x8 bytes. Writing a '1' to any bit position in the Set register will set valid bits in the associated register. Reads from the Set register should be ignored.

3: This register has an associated Invert register (ICxCONINV) at an offset of 0xC bytes. Writing a '1' to any bit position in the Invert register will invert valid bits in the associated register. Reads from the Invert register should be ignored.

Register 15-2: ICxBUF: Input Capture x Buffer Register

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
ICxBUF<31:24>							
bit 31							bit 24

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
ICxBUF<23:16>							
bit 23							bit 16

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
ICxBUF<15:8>							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
ICxBUF<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 31-0 **ICxBUF<31:0>:** Buffer Register bits
Value of the current captured input timer count.

15.3 TIMER SELECTION

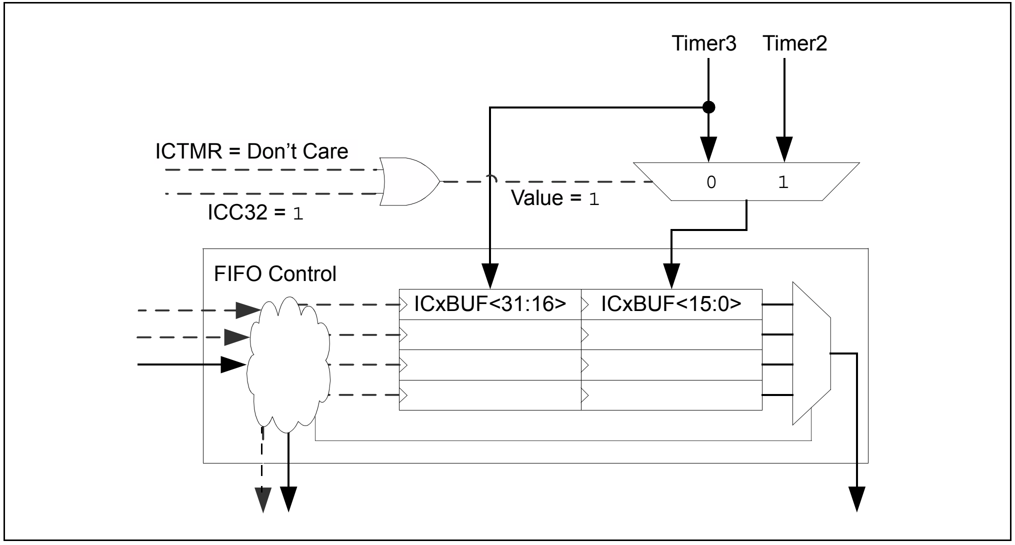
Each PIC32MX device may have one or more Input Capture modules. Each module can select between one of two 16-bit timers for the time base or one 32-bit timer, which is formed by combining two 16-bit timers. Refer to the specific device data sheet for the timers that can be selected.

For 16-bit Capture mode, setting ICTMR (ICxCON<7>) to '0' selects Timer3 for capture. Setting ICTMR (ICxCON<7>) to '1' selects Timer2 for capture.

An Input Capture module configured to support 32-bit capture may use a 32-bit timer resource for capture. By setting ICC32 (ICxCON<8>) to '1', a 32-bit timer resource is captured. The 32-bit timer resource is routed into the module using the existing 16-bit timer inputs. Timer2 provides the lower 16 bits and Timer3 provides the upper 16 bits, as shown in Figure 15-2.

The timer's clock can be set up using the internal peripheral clock source or a synchronized external clock source applied at the TxCK pin.

Figure 15-2: 32-bit Timer Selection Block Diagram



15.4 INPUT CAPTURE ENABLE

After configuration, an Input Capture module is enabled by setting the ON bit (ICxCON<15>). When this bit is cleared, the module is reset. Resetting the module has the following effects:

- Clears the Overflow Condition Flag
- Resets the FIFO to the empty state
- Resets the event count (for interrupt generation)
- Resets the prescaler count

Register reads and writes are allowed regardless of the ON bit (ICxCON<15>) state.

15.5 INPUT CAPTURE EVENT MODES

The Input Capture module captures the value of the selected time base register when an event occurs at the ICx pin. An Input Capture module can be configured in the following modes:

- Simple Capture Event modes:
  - Capture timer value on every falling edge of input at ICx pin
  - Capture timer value on every rising edge of input at ICx pin
  - Capture timer value on every rising and falling edge of input at ICx pin, starting with a specified edge
- Prescaled Capture Event modes:
  - Capture timer value on every fourth rising edge of input at ICx pin
  - Capture timer value on every sixteenth rising edge of input at ICx pin
- Edge Detect mode (See **15.5.3 "Edge Detect (Hall Sensor) Mode"**)
- Interrupt-Only mode (See **15.5.4 "Interrupt-Only Mode"**)

These input capture modes are configured by setting the appropriate Input Capture mode bits ICM<2:0> (ICxCON<2:0>).

When the Input Capture module is disabled (ICM<2:0> = `000`), the input capture logic ignores incoming capture edges and does not generate further capture events or interrupts. The FIFO continues to be operational for reading. Returning the module to any of the other modes resumes operation. A state change on the capture input while capture is disabled does not cause a capture event on exiting the Capture Disable mode.

> **Note:** The prescaler logic continues to run when the Input Capture module is in Capture Disable mode.

15.5.1 Simple Capture Events

The Input Capture module can capture a timer count value based on the selected edge (rising, falling or both, defined by mode) of the input applied to the ICx pin. Table 15-2 provides the mode settings. In Simple Capture Event mode, the prescaler is not used. See Figure 15-3, Figure 15-4 and Figure 15-5 for simplified timing diagrams of a simple capture event.

Table 15-2: Input Capture Mode Settings

ICM<2:0> Setting	Capture Occurs On
`001`	Rising edge
`010`	Falling edge
`110`	Rising and falling edges[1]

Note 1: This capture begins with the edge specified by the ICFEDGE bit (ICxCON<9>).

The input capture logic detects and synchronizes the rising or falling edge of the capture pin signal on the peripheral clock. When the rising/falling edge has occurred, the Input Capture module logic will write the current time base value to the capture buffer and signal the interrupt generation logic.

> **Note:** Since the capture input must be synchronized to the peripheral clock, the module captures the timer count value that is valid 2-3 peripheral clock cycles (T_{PB}) after the capture event.

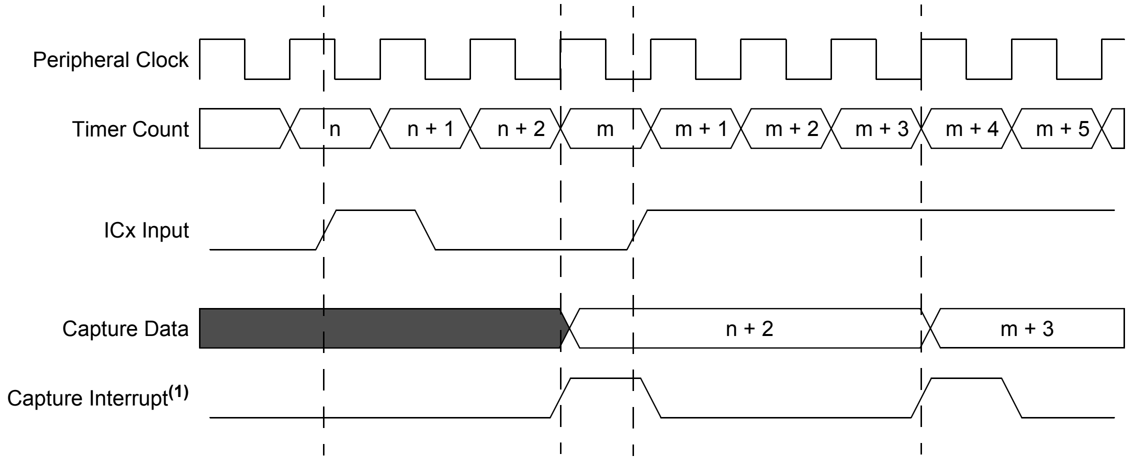
An input capture interrupt event is generated after one, two, three or four timer count captures, as configured by the ICI<1:0> bits (ICxCON<6:5>). See **15.7 "Input Capture Interrupts"** for further details.

Since the capture pin is sampled by the peripheral clock, the capture pulse high and low widths must be greater than the peripheral clock period.

Figure 15-3 depicts two capture events when the Input Capture module is in Simple Capture mode configured to capture every rising edge, ICM<2:0> = 011 (ICxCON<2:0>), with interrupts generated for every event, ICI<1:0> = 00 (ICxCON<6:5>).

The first capture event occurs when the timer value is 'n'. Due to synchronization delay, timer value 'n + 2' is stored in the capture buffer. The second capture event occurs when the timer value is 'm'. Note that 'm + 3' is stored in the capture buffer due to propagation delay as well as the synchronization delay. Interrupt events are generated on each capture event.

Figure 15-3: Simple Capture Event Timing Diagram, Capture Every Rising Edge

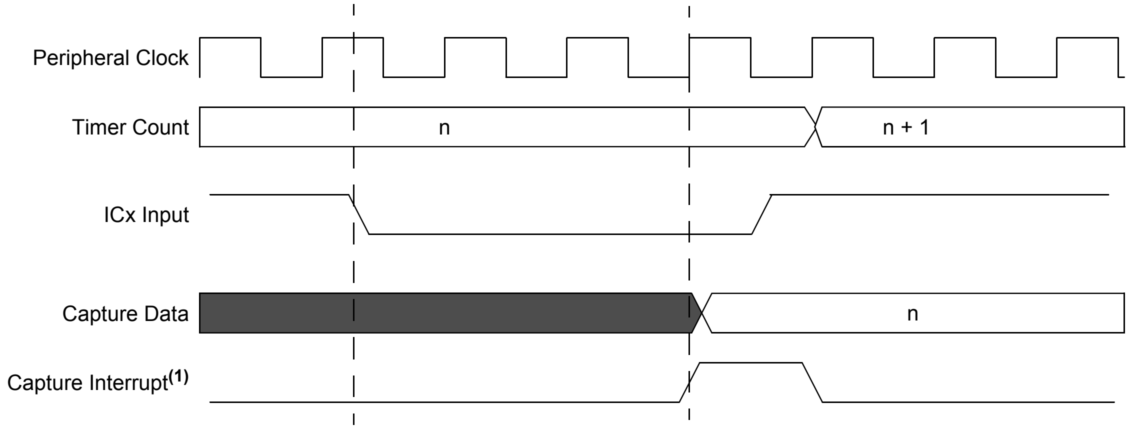


Note 1: This capture event flag is cleared by user software.

Figure 15-4 depicts a capture event when the Input Capture module is in Simple Capture mode configured to capture every falling edge, ICM<2:0> = 010 (ICxCON<2:0>), with interrupts generated for every event, ICI<1:0> = 00 (ICxCON<6:5>). In this example, the timer frequency is slower than the peripheral clock.

The capture event occurs when the timer value is 'n'. Value 'n' is stored in the capture buffer and an interrupt event is generated.

Figure 15-4: Simple Capture Event Timing Diagram, Capture Every Falling Edge

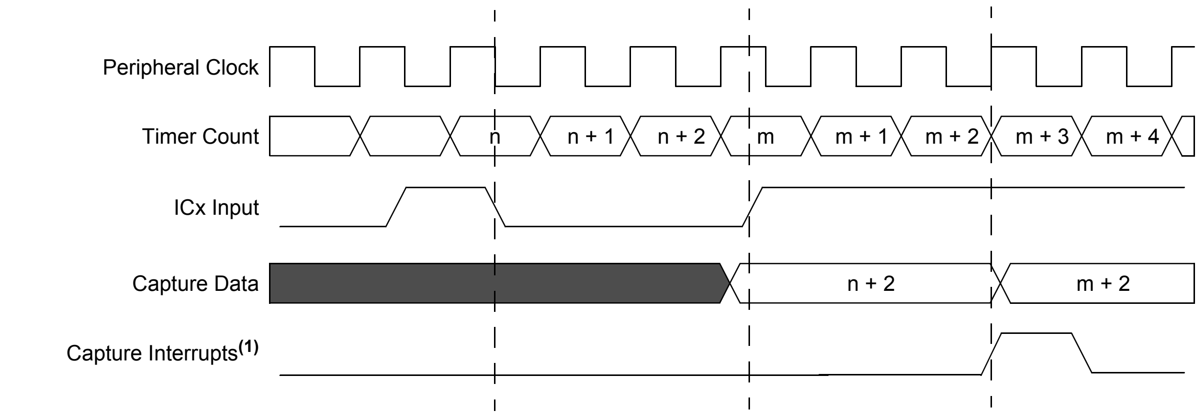


Note 1: This capture event flag is cleared by user software.

Figure 15-5 depicts a capture event when the Input Capture module is in Simple Capture mode configured to capture every edge, ICM<2:0> = `011` (ICxCON<2:0>); starting with a falling edge, ICFEDGE = `0` (ICxCON<9>), with interrupts generated for every second event, ICI<1:0> = `01` (ICxCON<6:5>).

The first falling edge occurs when the timer value is 'n'. Value 'n + 2' is stored in the capture buffer. A subsequent rising edge occurs when the timer value is 'm'. Value 'm + 2' is stored in the capture buffer and an interrupt event is generated.

Figure 15-5: Simple Capture Event Timing Diagram, Capture Every Edge, Falling Edge First



Note 1: This capture event flag is cleared by user software.

15.5.2 Prescaled Capture Event Mode

In Prescaled Capture Event mode, the Input Capture module triggers a capture event on either every fourth or every sixteenth rising edge. Table 15-3 provides the Prescaled Capture Event mode settings.

Table 15-3: Prescaled Capture Event Mode Settings

ICM<2:0> Setting	Number of Rising Edges Before Trigger
`100`	4
`101`	16

The capture prescaler counter is incremented on every rising edge on the capture input. When the prescaler counter equals four or sixteen (depending on the mode selected), the counter outputs a "valid" capture event signal. The valid capture event signal is then synchronized to the peripheral clock. The synchronized capture event signal triggers a timer count capture.

> **Note:** Since the capture input must be synchronized to the peripheral clock, the module captures the timer count value that is valid 2-3 peripheral clock cycles (TPB) after the capture event.

An input capture interrupt is generated after one, two, three or four timer count captures, as configured by the ICI<1:0> bits (ICxCON<6:5>). See **15.7 "Input Capture Interrupts"** for further details.

> **Note:** It is recommended that the user disable the capture module (i.e., clear the ON bit, ICxCON<15>), before switching to Prescaler Capture Event mode. Simply switching to Prescaler Capture Event mode from another active mode does not reset the prescaler and may cause an inadvertent capture event.

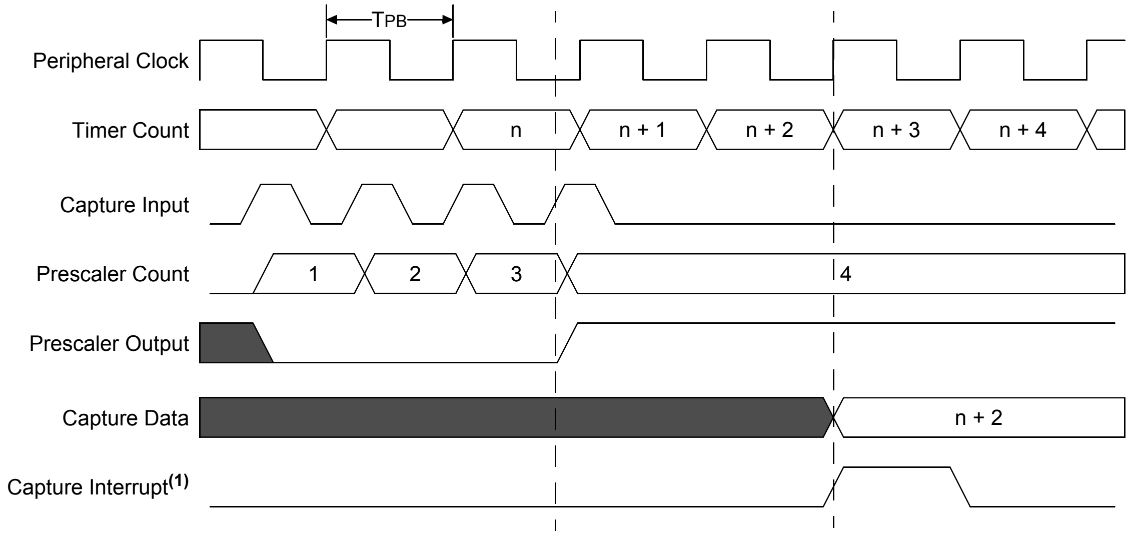
The prescaler counter is cleared when the following events occur:

- The Input Capture module is turned off (i.e., ON = 0 (ICxCON<15>))
- The Input Capture module is reset

Since the capture pin triggers an internal flip-flop, the input capture pulse high and low widths are not limited by the peripheral clock period. Refer to the **"Electrical Characteristics"** section in the specific device data sheet for details on input capture electrical specifications.

Figure 15-6 depicts a capture event when the Input Capture module is in Prescaler Capture Event mode. The prescaler is configured to capture a timer value for every fourth rising edge on the capture input, ICM<2:0> = `100` (ICxCON<2:0>), with interrupts generated for every capture event, ICI<1:0> = `00` (ICxCON<6:5>). The fourth rising edge on the capture input occurs at time 'n'. The prescaler output is synchronized. Due to synchronization delay, timer value 'n + 2' is stored in the capture buffer. An interrupt signal is generated due to the capture event.

Figure 15-6: Prescaler Capture Event Timing Diagram



Note 1: This capture event flag is cleared by user software.

15.5.3 Edge Detect (Hall Sensor) Mode

In Edge Detect mode, the Input Capture module captures a timer count value on every edge of the capture input. Edge Detection mode is selected by setting the ICM<2:0> bits to '`001`' (ICxCON<2:0>). In this mode, the capture prescaler is not used and the Input Capture Overflow bit, ICOV (ICxCON<4>), is not updated. In this mode, the Interrupt Control bits, ICI<1:0> (ICxCON<6:5>), are ignored and an interrupt event is generated for every timer count capture. See Figure 15-7 for a simplified timing diagram.

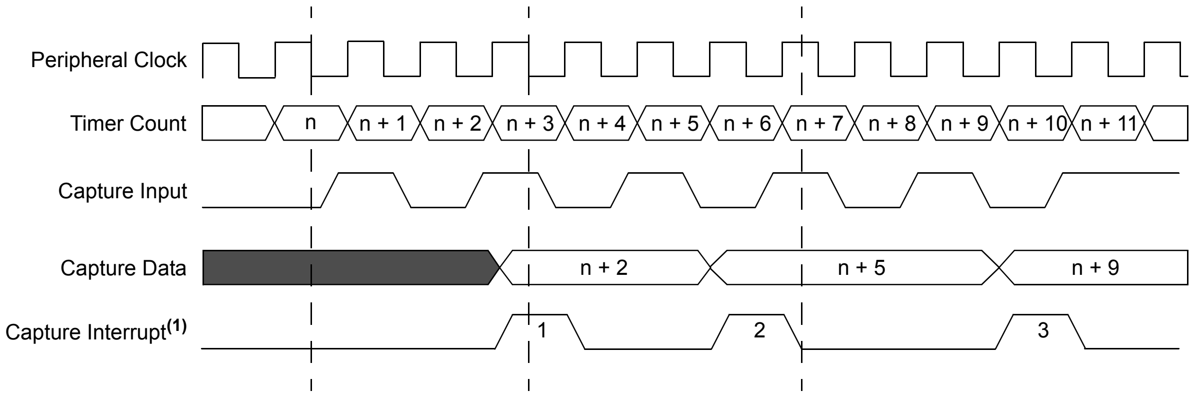
As with the Simple Capture Event mode, the Input Capture logic detects and synchronizes the rising and falling edge of the capture input signal on the peripheral clock. When a rising or falling edge occurs, the Input Capture module writes the time base value to the capture buffer.

> **Note:** Since the capture input must be synchronized to the peripheral clock, the module captures the timer count value that is valid 2-3 peripheral clock cycles (TPB) after the capture event.

Since the capture pin is sampled by the peripheral clock, the capture pulse high and low widths must be greater than the peripheral clock period.

Figure 15-7 depicts three capture events when the Input Capture module is in Edge Detect mode, ICM<2:0> = `001` (ICxCON<2:0). Transitions on the capture input occur at times 'n', 'n + 1' and 'n + 3'. Due to synchronization and propagation delay, timer values 'n + 2', 'n + 4' and 'n + 5' are stored in the capture buffer. Interrupt signals are generated due to each capture input transition.

Figure 15-7: Edge Detect Capture Event Timing Diagram



Note 1: This capture event flag is cleared by user software.

2: On PIC32MX3XX and PIC32MX4XX devices, clearing only Interrupt 3 will also clear Interrupts 1 and 2. On other PIC32MX devices, each interrupt is persistent and must be cleared individually (up to the maximum interrupt depth).

15.5.4 Interrupt-Only Mode

Interrupt-Only mode does not function while the device is running, and only operates when the device is in Sleep or Idle mode; however, during normal operation, when the device is in Sleep or Idle mode and the Input Capture module is set for Interrupt-Only mode (ICM<2:0> = `111`), the Input Capture module functions as an interrupt pin. Any rising edge on the input capture triggers an interrupt, which wakes the device. No timer values are captured and the FIFO buffer is not updated.

Since no timer values are captured, the Timer Select bit, ICTMR (ICxCON<7>), is ignored and there is no need to configure the timer source. The prescaler is not used in this mode, as the wake-up interrupt is generated on the first rising edge. Therefore, the ICI<1:0> bits (ICxCON<6:5>) are ignored. When the device leaves Sleep or Idle mode, the interrupt signal is deasserted. This mode is used strictly as an external wake-up source.

Since the capture pin triggers on an internal flip-flop, the input capture pulse high and low widths are not limited by the peripheral clock period. Refer to the **"Electrical Characteristics"** section in the specific device data sheet for details on input capture electrical specifications.

15.6 CAPTURE BUFFER OPERATION

Each Input Capture module has an associated four-level deep FIFO buffer. The buffer is accessible to the user application via the buffer register (ICxBUF). ICxBUF is written by the input capture logic and can only be read by the user application. Writes to ICxBUF are ignored.

There are two status flags that provide status on the FIFO buffer:

- ICBNE (ICxCON<3>) – Input Capture Buffer Not Empty
- ICOV (ICxCON<4>) – Input Capture Overrun

When the Input Capture module is disabled (i.e., ON = `0` (ICxCON<15>) or Reset), the status flags are cleared and the buffer is cleared to the empty state.

The ICBNE flag is set on the first input capture event and remains set until all capture events have been read from the FIFO. For example, if three capture events have occurred, three reads of the capture FIFO buffer are required before the ICBNE flag is cleared. If four capture events have occurred, four reads are required to clear the ICBNE flag.

Each read of the FIFO buffer adjusts the read pointer, allowing the remaining entries to move to the next available top location of the FIFO. In 32-bit Capture mode, the upper 16 bits must be read last if reading 16 bits at a time. The FIFO read pointer is advanced when reading the Most Significant Byte (MSB).

If the FIFO is full with four capture events and a fifth capture event occurs prior to a read of the FIFO, an overrun condition occurs and the ICOV bit (ICxCON<4>) is set to a logic '`1`'. The fifth capture event is not recorded, subsequent capture events do not alter the current FIFO contents until the overrun condition is cleared, and an input capture error interrupt will be generated.

Note: PIC32MX3XX and PIC32MX4XX microcontrollers do not support the ICxE interrupt. Refer to the specific device data sheet for availability.

The overflow condition is cleared in any of the following ways:

- Module is disabled (i.e., ON = `0` (ICxCON<15>))
- Capture buffer is read until ICBNE = `0` (ICxCON<3>)
- Device is Reset

If the Input Capture module is disabled and at some time re-enabled, the FIFO buffer contents are not defined and a read may yield indeterminate results.

If a FIFO read is performed when no capture event has been received, the read yields indeterminate results.

15.7 INPUT CAPTURE INTERRUPTS

The Input Capture module has the ability to generate an interrupt event signal based upon the selected number of capture events. A capture event is defined by the writing of a timer value into the FIFO. The number of capture events required to trigger an interrupt event is set by the ICI<1:0> control bits (ICxCON<6:5>) . If ICBNE = 0 (ICxCON<3>), the interrupt count is cleared. This allows the user to synchronize the interrupt count to the FIFO status.

For example, assuming that ICI<1:0> = 01 (specifying an interrupt event every second capture event), the following sequence could occur:

1. Turn on module, interrupt count = 0.
2. Capture event. FIFO contains one entry, interrupt count = 1.
3. Read FIFO. FIFO is empty, interrupt count = 0.
4. Capture event. FIFO contains one entry, interrupt count = 1.
5. Capture event. FIFO contains two entries, interrupt count = 2.
6. Interrupt issued. interrupt count = 0.
7. Capture event. FIFO contains three entries, interrupt count = 1.
8. Read FIFO three times. FIFO becomes empty, interrupt count = 0.
9. Capture event. FIFO contains one entry, interrupt count = 1.
10. Read FIFO. FIFO becomes empty, interrupt count = 0.

The first capture event is defined as the capture event occurring after a mode change from the OFF mode or after ICBNE = 0.

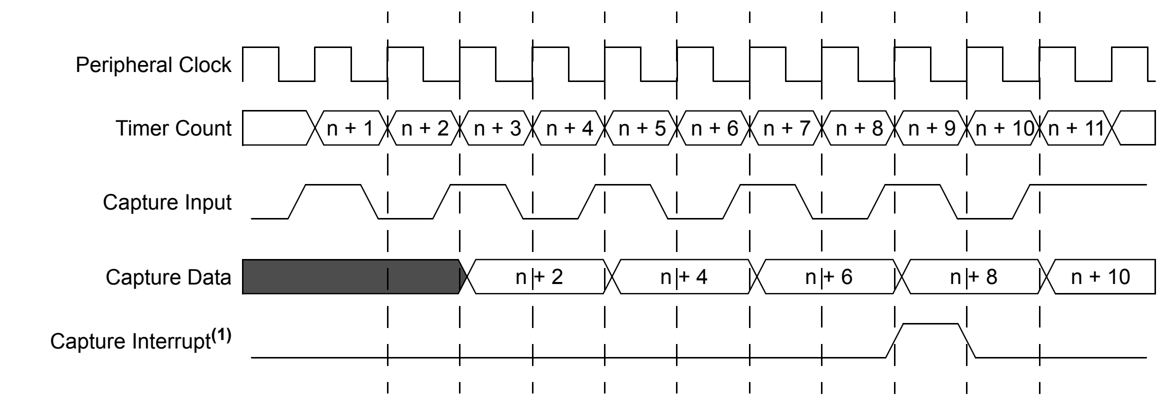
When an overrun occurs (unless ICI<1:0> = 00 or ICM<2:0> = 001), the Input Capture module will stop generating input capture events and generates an input capture error event instead. This interrupt will persist until the overflow condition is cleared (see **15.6 "Capture Buffer Operation"** for details on how to clear the overflow condition).

Note:	PIC32MX3XX and PIC32MX4XX microcontrollers do not support the ICxE interrupt. Refer to the specific device data sheet for availability.

Applications often dictate using the Input Capture pins as auxiliary external interrupt sources. When ICI<1:0> = 00 or ICM<2:0> = 001, interrupt events occur regardless of FIFO overrun. There is no need to perform a dummy read on the capture buffer to clear the event and prevent an overflow in order to ensure that future interrupt events are not inhibited. The ICOV flag (ICxCON<4>) is still set for the overflow condition.

Figure 15-8 depicts five capture events when the Input Capture module is configured to capture timer values on every rising edge (ICM<2:0> = 011) and generate an interrupt for every four captures (ICI<1:0> = 11). Note that the fourth capture causes the capture of value 'n + 8' and triggers an interrupt event.

Figure 15-8: Interrupt Event, ICxCON.ICM<2:0> = 011, ICxCON.ICI<1:0> = 11



Note 1: This capture event flag is cleared by user software.

15.7.1 Interrupt Control Bits

Each Input Capture module has interrupt flag status bits (ICxIF), interrupt error status bits (ICxE), interrupt enable bits (ICxIE), interrupt priority control bits (ICxIP) and secondary interrupt priority control bits (ICxIS). Refer to **8.2 "Control Registers"** in **Section 8. "Interrupts"** (DS61108) for further information on peripheral interrupts.

15.7.2 Interrupt Persistence

Input capture interrupts persist so long as the condition that caused them persists. In addition, they will occur again immediately if the condition is not cleared. Table 15-4 describes the interrupt persistence set and clear conditions.

> **Note:** PIC32MX3XX and PIC32MX4XX microcontrollers do not support the ICxE interrupt or persistent interrupts. Refer to the specific device data sheet for availability.

Table 15-4: Interrupt Persistence Conditions

ICxCON Value	Set Condition	Persistence
ICI<1:0> = 11	Interrupt on every fourth capture event.	Interrupt is active if the number of FIFO entries is equal to 4.
ICI<1:0> = 10	Interrupt on every third capture event.	Interrupt is active if the number of FIFO entries is greater than or equal to 3.
ICI<1:0> = 01	Interrupt on every second capture event.	Interrupt is active if the number of FIFO entries is greater than or equal to 2.
ICI<1:0> = 00 or Edge Detect modes (see the ICM<2:0> bits in the ICxCON register (Register 15-1))	Interrupt on every capture event.	Interrupt is active if the number of FIFO entries is greater than or equal to 1.
ICOV = 1	Interrupt on fifth capture event if FIFO is full.	Interrupt is active until the error condition flag (ICxCON.ICOV) is cleared.

15.8 OPERATION IN POWER-SAVING MODES

15.8.1 Input Capture Operation in Sleep Mode

When the device enters Sleep mode, the peripheral clock is disabled. In Sleep mode, the Input Capture module can only function as an external interrupt source. This mode is enabled by setting the ICM<2:0> control bits = 111 (ICxCON<2:0>), for Interrupt-Only mode. In this mode, a rising edge on the capture pin will generate a device wake-up from Sleep. If the respective module interrupt bit is enabled and the module's priority is of the required priority level, an interrupt will be generated. Refer to **15.5.4 "Interrupt-Only Mode"** for more details.

If the Input Capture module has been configured for a mode other than ICM<2:0> = 111 and the device enters Sleep mode, no external pin stimulus, rising or falling, will generate a wake-up from Sleep.

15.8.2 Input Capture Operation in Idle Mode

When the device enters Idle mode, the peripheral clock sources remain functional and the CPU stops executing code. The Sleep-In-Idle Control bit, SIDL (ICxCON<13>), determines whether the module will stop in Idle mode or continue to operate.

If SIDL is '0', the module continues normal operation in Idle mode. Although Interrupt-Only mode (ICM<2:0> = 111) may generate an interrupt when in Idle mode if SIDL is '0', an interrupt is not generated when the processor is running. Refer to **15.5.4 "Interrupt-Only Mode"** for further details.

If SIDL is '1', the module stops when the device is in Idle mode. The module performs the same procedures when stopped in Idle mode as for Sleep mode. Refer to **15.5.4 "Interrupt-Only Mode"** for further details.

15.8.3 Device Wake-up on Sleep or Idle

While using Interrupt-Only mode, an input capture event can generate a device wake-up or interrupt, if enabled, when the device is in Sleep or Idle mode. Refer to **15.5.4 "Interrupt-Only Mode"** for further details.

15.9 INPUT CAPTURE OPERATION IN DEBUG MODE

The FRZ bit (ICxCON<14>) determines whether the Input Capture module will run or stop while the CPU is executing Debug Exception code (i.e., the application is halted) in Debug mode.

When FRZ is '0', the Input Capture module continues to run even when the application is halted in Debug mode. When FRZ is '1' and the application is halted in Debug mode, the Input Capture module will freeze its operations and make no changes to its current state. The module will resume its operation after the CPU resumes execution.

Note:	The FRZ bit is readable and writable only when the CPU is executing in Debug Exception mode. In all other modes, the FRZ bit reads as '0'. If FRZ bit is changed during Debug mode, the new value does not take effect until the current Debug Exception mode is exited and re-entered. During the Debug Exception mode, the FRZ bit reads the state of the peripheral when entering Debug mode.

15.9.1 Capture Operation During Freeze (FRZ = 1)

When frozen, the capture operation does not cause changes to the module. The edge detection logic prevents any state changes that occur during Freeze from being inadvertently detected after leaving Freeze.

Note:	The prescaler logic is not frozen during Debug mode.

When frozen, the emulator is allowed to read the Input Capture FIFO; however, the FIFO status flags as viewed by the user application do not change.

15.10 I/O PIN CONTROL

When the Input Capture module is enabled, the user application must ensure that the I/O pin direction is configured for an input by setting the associated TRIS bit. The pin direction is not set when the Input Capture module is enabled. Furthermore, all other peripherals multiplexed with the input pin must be disabled.

15.11 DESIGN TIPS

Question 1: ***Can the Input Capture module be used to wake the device from Sleep mode?***

Answer: Yes. When the Input Capture module is configured to ICM<2:0> = `111` (ICxCON<2:0>) and the respective module interrupt enable bit is asserted (ICIE = `1`), a rising edge on the capture pin will wake the device from Sleep. (See **15.5.4 "Interrupt-Only Mode"** for further details.)

15.12 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 PIC32MX device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Input Capture module include the following:

Title	Application Note #
Using the CCP Module(s)	AN594
Implementing Ultrasonic Ranging	AN597

Note: Please visit the Microchip web site (www.microchip.com) for additional application notes and code examples for the PIC32MX family of devices.

15.13 REVISION HISTORY

Revision A (October 2007)

This is the initial released version of this document.

Revision B (October 2007)

Updated document to remove Confidential status.

Revision C (April 2008)

Revised status to Preliminary; Revised U-0 to r-x; Removed ‘x’ in bit names.

Revision D (June 2008)

Revised note for Registers 15-1, 15-3, 15-4, 15-5, 15-6, 15-7, 15-8, 15-9; Change Reserved bits from “Maintain as” to “Write”; Added Note to ON bit (ICxCON Register).

Revision E (October 2009)

This revision includes the following updates:

- Minor updates to text and formatting have been incorporated throughout the document.
- Updated the following figures:
  - Simple Capture Event Timing Diagram, Capture Every Rising Edge (Figure 15-3)
  - Simple Capture Event Timing Diagram, Capture Every Falling Edge (Figure 15-4)
  - Simple Capture Event Timing Diagram, Capture Every Edge, Falling Edge First (Figure 15-5)
  - Prescaler Capture Event Timing Diagram (Figure 15-6)
  - Edge Detect Capture Event Timing Diagram (Figure 15-7)
  - Interrupt Event, ICxCON.ICM<2:0> = `011`, ICxCON.ICI<1:0> = `11` (Figure 15-8)
- Updated the shaded note in **15.1 “Introduction”**.
- Removed the shaded note in **15.2 “Input Capture Registers”**.
- Updated the Interrupts Register Summary (Table 15-1):
  - Removed all references to the Clear, Set and Invert registers
  - Added the Address Offset column
  - Added Notes 1, 2 and 3, which describe the Clear, Set and Invert registers
- Added Notes 1, 2 and 3, which describe the Clear, Set and Invert registers to the ICxCON register (see Register 15-1).
- Removed the IFS0, IEC0, IPC1, IPC2, IPC3, IPC4 and IPC5 registers.
- Updated the bit definition for ICM<2:0> = `111` in the ICxCON: Input Capture x Control Register (Register 15-1).
- Updated the first paragraph of **15.3 “Timer Selection”** to clarify the formation of the timers used for the time base.
- Removed content from the first paragraph of **15.5.1 “Simple Capture Events”**, which is now presented in Table 15-2.
- Removed content from the first paragraph of **15.5.2 “Prescaled Capture Event Mode”**, which is now presented in Table 15-3.
- Updated all three paragraphs of **15.5.4 “Interrupt-Only Mode”**.
- Added a shaded note and updated the sixth paragraph of **15.6 “Capture Buffer Operation”**.
- Updated the capture event sequence in **15.7 “Input Capture Interrupts”**.
- Added a reference to the interrupt error status bits (ICxE) in **15.7.1 “Interrupt Control Bits”**.
- Added **15.7.2 “Interrupt Persistence”**.
- Updated the second paragraph of **15.8.2 “Input Capture Operation in Idle Mode”** to clarify operation during Interrupt-Only mode.

Revision E (October 2009) (Continued)

- Updated **15.8.3 “Device Wake-up on Sleep or Idle”** to clarify operation during Interrupt-Only mode.
- Updated the first paragraph of **15.9.1 “Capture Operation During Freeze (FRZ = `1`)”**, and removed the second paragraph.
- Removed 15.9.2 “Operation of the Capture Buffer in Debug Mode”.

NOTES: