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.

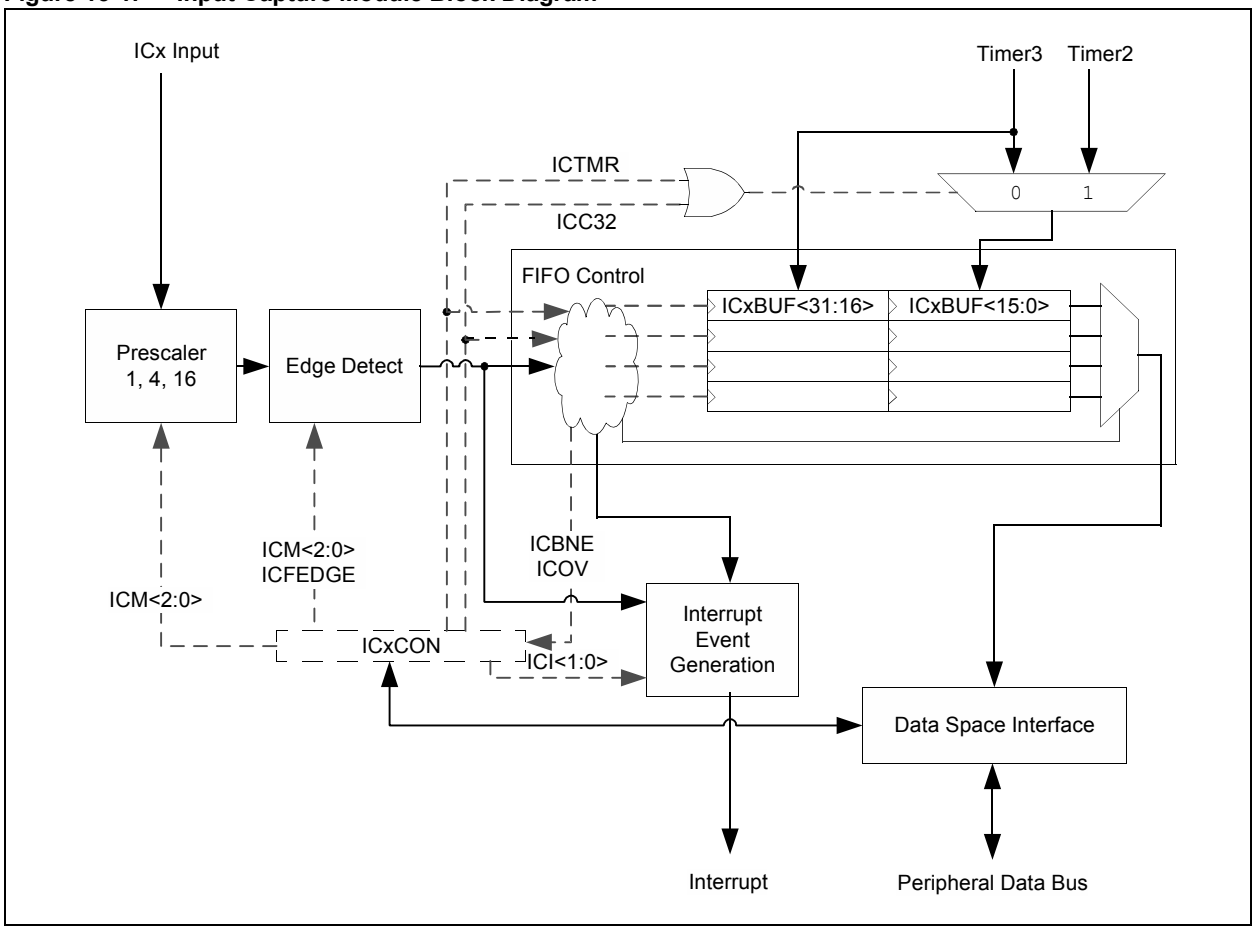
Refer to the specific device data sheet for information on the number of channels available in a particular device. All Input Capture channels are functionally identical. In this section, an 'x' in the pin name or register name denotes the specific Input Capture channel. Timer Y and Timer Z each refer to one of two timer inputs which may be associated with the Input Capture channel.

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 16th 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 FIFO buffer. The number of capture events required to generate a CPU interrupt can be selected by the user. An Input Capture Channel 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

Note: Each PIC32MX device variant may have one or more Input Capture modules. An 'x' used in the names of pins, control/Status bits and registers denotes the particular module. Refer to the specific device data sheets for more details.

Each capture module available on the PIC32MX devices has the following Special Function Registers (SFRs), where 'x' denotes the module number:

- ICxCON: Input Capture Control Register

  ICxCONCLR, ICxCONSET, ICxCONINV: Atomic Bit Manipulation Write-only Registers for ICxCON
- 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)

The tables below provide a brief summary of all the Input Capture related registers, and is followed by a detailed description of each register.

Table 15-1: Input Capture SFR Summary

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
ICxCON	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	ON	FRZ	SIDL	—	—	—	ICFEDGE	ICC32
	7:0	ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>		
ICxCONCLR	31:0	Write clears selected bits in ICxCON, read yields undefined value							
ICxCONSET	31:0	Write sets selected bits in ICxCON, read yields undefined value							
ICxCONINV	31:0	Write inverts selected bits in ICxCON, read yields undefined value							
ICxBUF	31:24	ICxBUF<31:24>							
	23:16	ICxBUF<23:16>							
	15:8	ICxBUF<15:8>							
	7:0	ICxBUF<7:0>							
IFS0	31:24	I2C1MIF	I2C1SIF	I2C1BIF	U1TXIF	U1RXIF	U1EIF	SPI1RXIF	SPI1TXIF
	23:16	SPI1EIF	OC5IF	IC5IF	T5IF	INT4IF	OC4IF	IC4IF	T4IF
	15:8	INT3IF	OC3IF	IC3IF	T3IF	INT2IF	OC2IF	IC2IF	T2IF
	7:0	INT1IF	OC1IF	IC1IF	T1IF	INT0IF	CS1IF	CS0IF	CTIF
IEC0	31:24	I2C1MIE	I2C1SIE	I2C1BIE	U1TXIE	U1RXIE	U1EIE	SPI1RXIE	SPI1TXIE
	23:16	SPI1EIE	OC5IE	IC5IE	T5IE	INT4IE	OC4IE	IC4IE	T4IE
	15:8	INT3IE	OC3IE	IC3IE	T3IE	INT2IE	OC2IE	IC2IE	T2IE
	7:0	INT1IE	OC1IE	IC1IE	T1IE	INT0IE	CS1IE	CS0IE	CTIE
IPC1	31:24	—	—	—	INT1IP<2:0>			INT1IS<1:0>	
	23:16	—	—	—	OC1IP<2:0>			OC1IS<1:0>	
	15:8	—	—	—	IC1IP<2:0>			IC1IS<1:0>	
	7:0	—	—	—	T1IP<2:0>			T1IS<1:0>	
IPC2	31:24	—	—	—	INT2IP<2:0>			INT2IS<1:0>	
	23:16	—	—	—	OC2IP<2:0>			OC2IS<1:0>	
	15:8	—	—	—	IC2IP<2:0>			IC2IS<1:0>	
	7:0	—	—	—	T2IP<2:0>			T2IS<1:0>	

Table 15-1: Input Capture SFR Summary (Continued)

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
IPC3	31:24	—	—	—	INT3IP<2:0>			INT3IS<1:0>	
	23:16	—	—	—	OC3IP<2:0>			OC3IS<1:0>	
	15:8	—	—	—	IC3IP<2:0>			IC3IS<1:0>	
	7:0	—	—	—	T3IP<2:0>			T3IS<1:0>	
IPC4	31:24	—	—	—	INT4IP<2:0>			INT4IS<1:0>	
	23:16	—	—	—	OC4IP<2:0>			OC4IS<1:0>	
	15:8	—	—	—	IC4IP<2:0>			IC4IS<1:0>	
	7:0	—	—	—	T4IP<2:0>			T4IS<1:0>	
IPC5	31:24	—	—	—	SPI1IP<2:0>			SPI1IS<1:0>	
	23:16	—	—	—	OC5IP<2:0>			OC5IS<1:0>	
	15:8	—	—	—	IC5IP<2:0>			IC5IS<1:0>	
	7:0	—	—	—	T5IP<2:0>			T5IS<1:0>	

Register 15-1: ICxCON: Input Capture x Control Register

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	r-x	r-x	r-x	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:** Maintain as ‘0’; Ignore Read

bit 15 **ON:** ON bit
1 = Module enabled
0 = Disable and reset module, disable clocks, disable interrupt generation, and allow SFR modifications

bit 14 **FRZ:** Freeze in DEBUG Mode Control bit (Read/Write only in DEBUG mode; otherwise read as ‘0’)
1 = Freeze module operation when in DEBUG mode
0 = Do not freeze module operation when in DEBUG 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, ICxM = 110)
1 = Capture rising edge first
0 = Capture falling edge first

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

Register 15-1: ICxCON: Input Capture x Control Register (Continued)

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
110 = Simple Capture Event mode – every edge, specified edge first and every edge thereafter
101 = Prescaled Capture Event mode – every 16th rising edge
100 = Prescaled Capture Event mode – every 4th 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

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

Register 15-3: IFS0: Interrupt Flag Status Register 0(1,2)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
I2C1SIF	I2C1BIF	U1TXIF	U1RXIF	U1EIF	SPI1RXIF	SPI1TXIF	SPI1EIF
bit 31							bit 24

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNIF	OC5IF	IC5IF	T5IF	INT4IF	OC4IF	IC4IF	T4IF
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT3IF	OC3IF	IC3IF	T3IF	INT2IF	OC2IF	IC2IF	T2IF
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT1IF	OC1IF	IC1IF	T1IF	INT0IF	CS1IF	CS0IF	CTIF
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 21 **IC5IF:** Input Capture 5 Interrupt Request Flag bit
1 = Interrupt request has occurred
0 = No interrupt request has a occurred

bit 17 **IC4IF:** Input Capture 4 Interrupt Request Flag bit
1 = Interrupt request has occurred
0 = No interrupt request has a occurred

bit 13 **IC3IF:** Input Capture 3 Interrupt Request Flag bit
1 = Interrupt request has occurred
0 = No interrupt request has a occurred

bit 9 **IC2IF:** Input Capture 2 Interrupt Request Flag bit
1 = Interrupt request has occurred
0 = No interrupt request has a occurred

bit 5 **IC1IF:** Input Capture 1 Interrupt Request Flag bit
1 = Interrupt request has occurred
0 = No interrupt request has a occurred

Note 1: Register is cleared on all forms of Reset.

2: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the Input Capture module.

Register 15-4: IEC0: Interrupt Enable Control Register 0(1,2)

I2C1SIE	I2C1BIE	U1TXIE	U1RXIE	U1EIE	SPI1RXIE	SPI1TXIE	SPI1EIE
bit 31							bit 24

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNIE	OC5IE	IC5IE	T5IE	INT4IE	OC4IE	IC4IE	T4IE
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT3IE	OC3IE	IC3IE	T3IE	INT2IE	OC2IE	IC2IE	T2IE
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT1IE	OC1IE	IC1IE	T1IE	INT0IE	CS1IE	CS0IE	CTIE
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 21 **IC5IE:** Input Capture 5 Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 17 **IC4IE:** Input Capture 4 Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 13 **IC3IE:** Input Capture 3 Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 9 **IC2IE:** Input Capture 2 Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 5 **IC1IE:** Input Capture 1 Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

Note 1: Register is cleared on all forms of Reset.
2: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the Input Capture module.

Register 15-5: IPC1: Interrupt Priority Control Register 1(1,2)

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT1IP<2:0>			INT1IS<1:0>	
bit 31							bit 24

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	OC1IP<2:0>			OC1IS<1:0>	
bit 23							bit 16

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	IC1IP<2:0>			IC1IS<1:0>	
bit 15							bit 8

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	T1IP<2:0>			T1IS<1:0>	
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 12-10 **IC1IP<2:0>:** Input Capture 1 Interrupt Priority bits
- 111 = Interrupt priority is 7
- 110 = Interrupt priority is 6
- 101 = Interrupt priority is 5
- 100 = Interrupt priority is 4
- 011 = Interrupt priority is 3
- 010 = Interrupt priority is 2
- 001 = Interrupt priority is 1
- 000 = Interrupt is disabled

bit 9-8 **IC1IS<1:0>:** Input Capture 1 Interrupt Subpriority bits
- 11 = Interrupt subpriority is 3
- 10 = Interrupt subpriority is 2
- 01 = Interrupt subpriority is 1
- 00 = Interrupt subpriority is 0

Note 1: Register is cleared on all forms of Reset.

2: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the Input Capture module.

Register 15-6: IPC2: Interrupt Priority Control Register 2(1,2)

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT2IP<2:0>			INT2IS<1:0>	
bit 31							bit 24

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	OC2IP<2:0>			OC2IS<1:0>	
bit 23							bit 16

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	IC2IP<2:0>			IC2IS<1:0>	
bit 15							bit 8

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	T2IP<2:0>			T2IS<1:0>	
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 12-10 **IC2IP<2:0>:** Input Capture 2 Interrupt Priority bits
- 111 = Interrupt priority is 7
- 110 = Interrupt priority is 6
- 101 = Interrupt priority is 5
- 100 = Interrupt priority is 4
- 011 = Interrupt priority is 3
- 010 = Interrupt priority is 2
- 001 = Interrupt priority is 1
- 000 = Interrupt is disabled

bit 9-8 **IC2IS<1:0>:** Input Capture 2 Interrupt Subpriority bits
- 11 = Interrupt subpriority is 3
- 10 = Interrupt subpriority is 2
- 01 = Interrupt subpriority is 1
- 00 = Interrupt subpriority is 0

Note 1: Register is cleared on all forms of Reset.

2: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the Input Capture module.

Register 15-7: IPC3: Interrupt Priority Control Register 3(1,2)

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT3IP<2:0>			INT3IS<1:0>	
bit 31							bit 24

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	OC3IP<2:0>			OC3IS<1:0>	
bit 23							bit 16

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	IC3IP<2:0>			IC3IS<1:0>	
bit 15							bit 8

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	T3IP<2:0>			T3IS<1:0>	
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 12-10 **IC3IP<2:0>:** Input Capture 3 Interrupt Priority bits

111 = Interrupt priority is 7
110 = Interrupt priority is 6
101 = Interrupt priority is 5
100 = Interrupt priority is 4
011 = Interrupt priority is 3
010 = Interrupt priority is 2
001 = Interrupt priority is 1
000 = Interrupt is disabled

bit 9-8 **IC3IS<1:0>:** Input Capture 3 Interrupt Subpriority bits

11 = Interrupt subpriority is 3
10 = Interrupt subpriority is 2
01 = Interrupt subpriority is 1
00 = Interrupt subpriority is 0

Note 1: Register is cleared on all forms of Reset.

2: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the Input Capture module.

Register 15-8: IPC4: Interrupt Priority Control Register 4(1,2)

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT4IP<2:0>			INT4IS<1:0>	
bit 31							bit 24

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	OC4IP<2:0>			OC4IS<1:0>	
bit 23							bit 16

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	IC4IP<2:0>			IC4IS<1:0>	
bit 15							bit 8

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	T4IP<2:0>			T4IS<1:0>	
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 12-10 **IC4IP<2:0>:** Input Capture 4 Interrupt Priority bits
- 111 = Interrupt priority is 7
- 110 = Interrupt priority is 6
- 101 = Interrupt priority is 5
- 100 = Interrupt priority is 4
- 011 = Interrupt priority is 3
- 010 = Interrupt priority is 2
- 001 = Interrupt priority is 1
- 000 = Interrupt is disabled

bit 9-8 **IC4IS<1:0>:** Input Capture 4 Interrupt Subpriority bits
- 11 = Interrupt subpriority is 3
- 10 = Interrupt subpriority is 2
- 01 = Interrupt subpriority is 1
- 00 = Interrupt subpriority is 0

Note 1: Register is cleared on all forms of Reset.

2: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the Input Capture module.

Register 15-9: IPC5: Interrupt Priority Control Register 5(1,2)

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	SPI1IP<2:0>			SPI1IS<1:0>	
bit 31							bit 24

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	OC5IP<2:0>			OC5IS<1:0>	
bit 23							bit 16

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	IC5IP<2:0>			IC5IS<1:0>	
bit 15							bit 8

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	T5IP<2:0>			T5IS<1:0>	
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 12-10 **IC5IP<2:0>:** Input Capture 5 Interrupt Priority bits
- 111 = Interrupt priority is 7
- 110 = Interrupt priority is 6
- 101 = Interrupt priority is 5
- 100 = Interrupt priority is 4
- 011 = Interrupt priority is 3
- 010 = Interrupt priority is 2
- 001 = Interrupt priority is 1
- 000 = Interrupt is disabled

bit 9-8 **IC5IS<1:0>:** Input Capture 5 Interrupt Subpriority bits
- 11 = Interrupt subpriority is 3
- 10 = Interrupt subpriority is 2
- 01 = Interrupt subpriority is 1
- 00 = Interrupt subpriority is 0

Note 1: Register is cleared on all forms of Reset.

2: Shaded bit names in this Interrupt register control other PIC32MX peripherals and are not related to the Input Capture module.

15.3 TIMER SELECTION

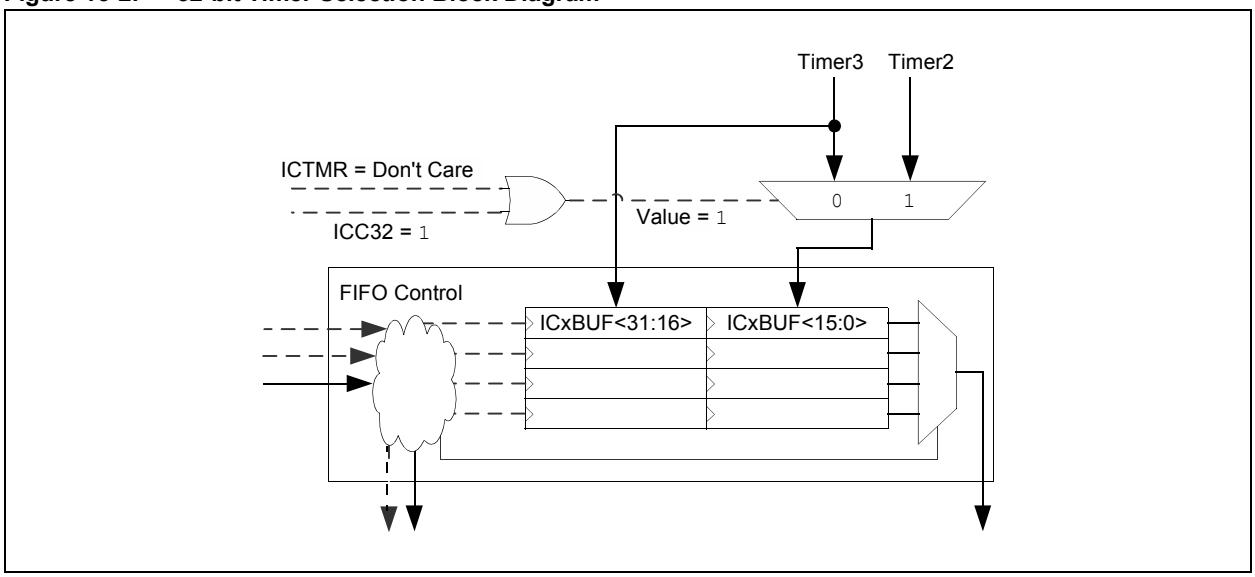
Each PIC32MX device may have one or more input capture channels. Each channel can select between one of two 16-bit timers for the time base or two 16-bit timers together (to form a 32-bit timer). Refer to the device data sheet for the specific timers that can be selected.

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

An input capture channel 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. The Timer 2 provides the lower 16-bits and the Timer 3 provides the upper 16-bits, as shown in Figure 15-2.

The timers 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 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 (ICxCON<15>) bit 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 channel can be configured in the following modes:

1. 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
2. Prescaled Capture Event modes:
   Capture timer value on every 4th rising edge of input at ICx pin
   Capture timer value on every 16th rising edge of input at ICx pin
3. Edge Detect mode (See **15.5.3 "Edge Detect (Hall Sensor) Mode"**)
4. Interrupt Only mode (See **15.5.4 "Interrupt Only Mode"**)

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

When the Input Capture Channel is disabled (ICM = 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 channel 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 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. These modes are specified by setting the ICM (ICxCON<2:0>) bits to '010', '011', or '110'. Setting ICM = 011 configures the module to capture the timer value on any rising edge of the capture input. ICM = 010 configures the module to capture the timer on any falling edge of the capture input. Setting ICM = 110 configures the channel to capture the timer on every transition of the capture input, beginning with the edge specified by ICFEDGE (ICxCON<9>). 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.

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 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 (TPB) after the capture event.

An input capture interrupt event is generated after one, two, three or four timer count captures, as configured by ICI (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 = 011 (ICxCON<2:0>), with interrupts generated for every event, ICI = 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

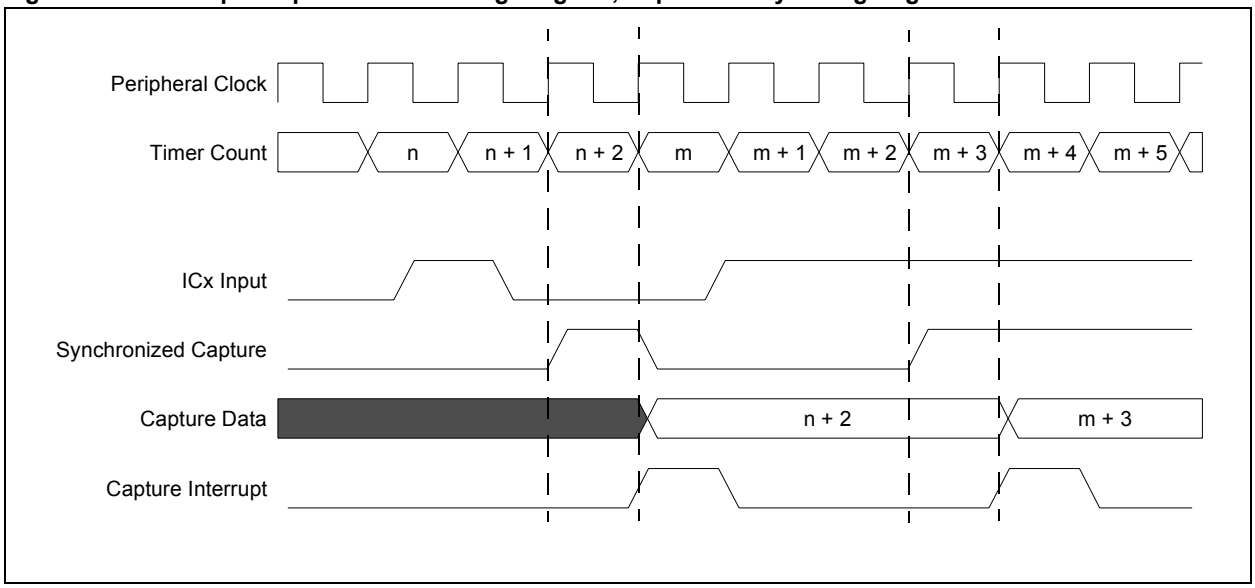


Figure 15-4 depicts a capture event when the Input Capture module is in Simple Capture mode configured to capture every falling edge, ICM = 010 (ICxCON<2:0>), with interrupts generated for every event, ICI = 00 (ICxCON<6:5>). In this example, the timer frequency is slower than the peripheral clock.

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

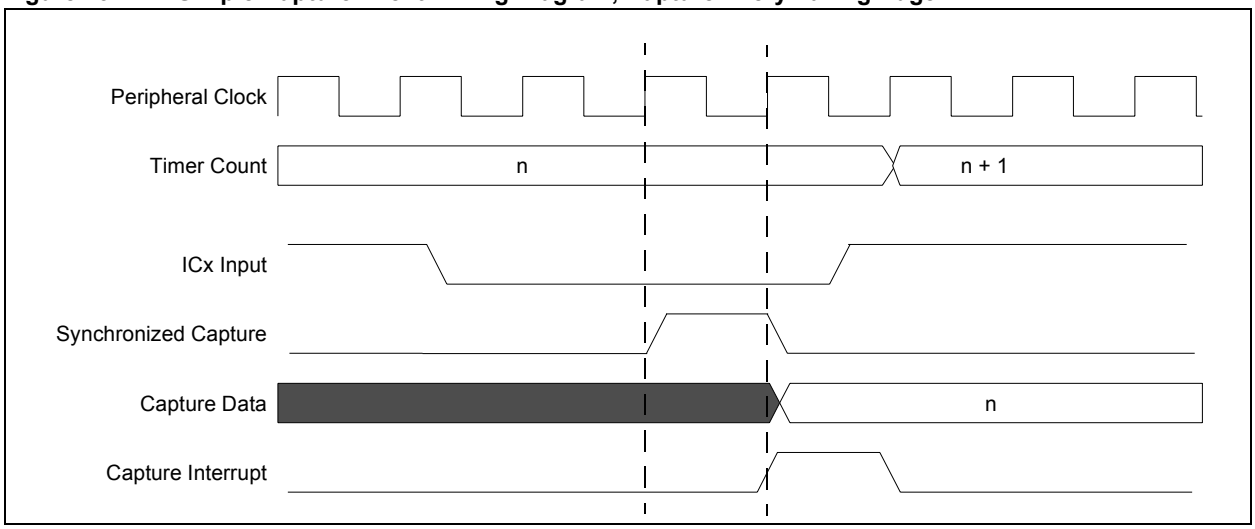
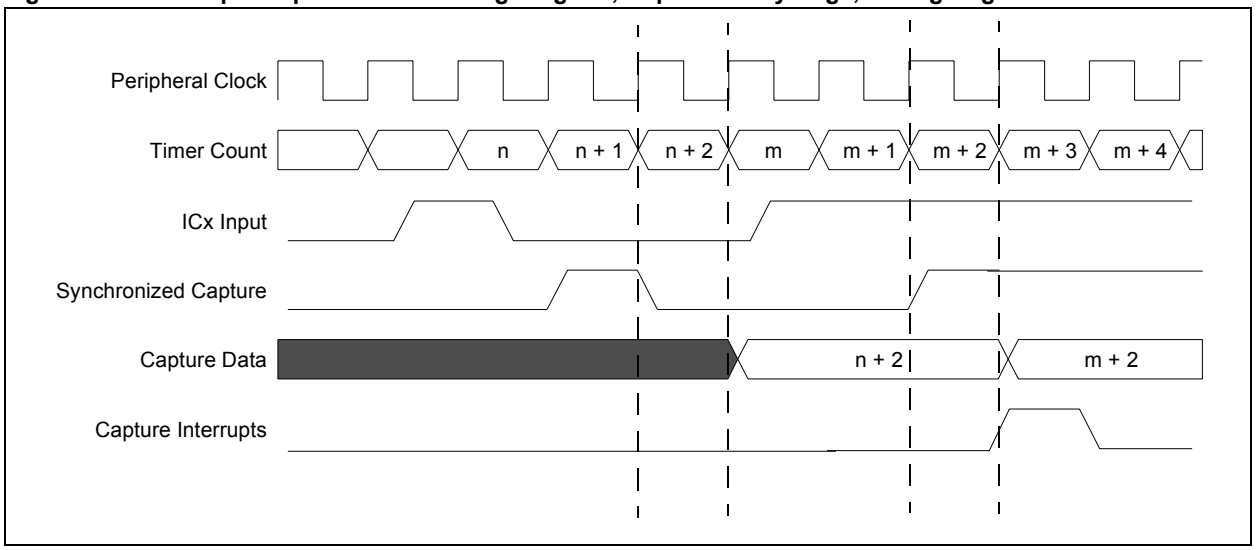


Figure 15-5 depicts a capture event when the Input Capture module is in Simple Capture mode configured to capture every edge, ICM = 011 (ICxCON<2:0>); starting with a falling edge, ICFEDGE = 0 (ICxCON<9>), with interrupts generated for every second event, ICI = 01 (ICxCON<6:5>).

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



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. These modes are selected by setting the ICM (ICxCON<2:0>) bits to '100' or '101', respectively.

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 ICI. 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.

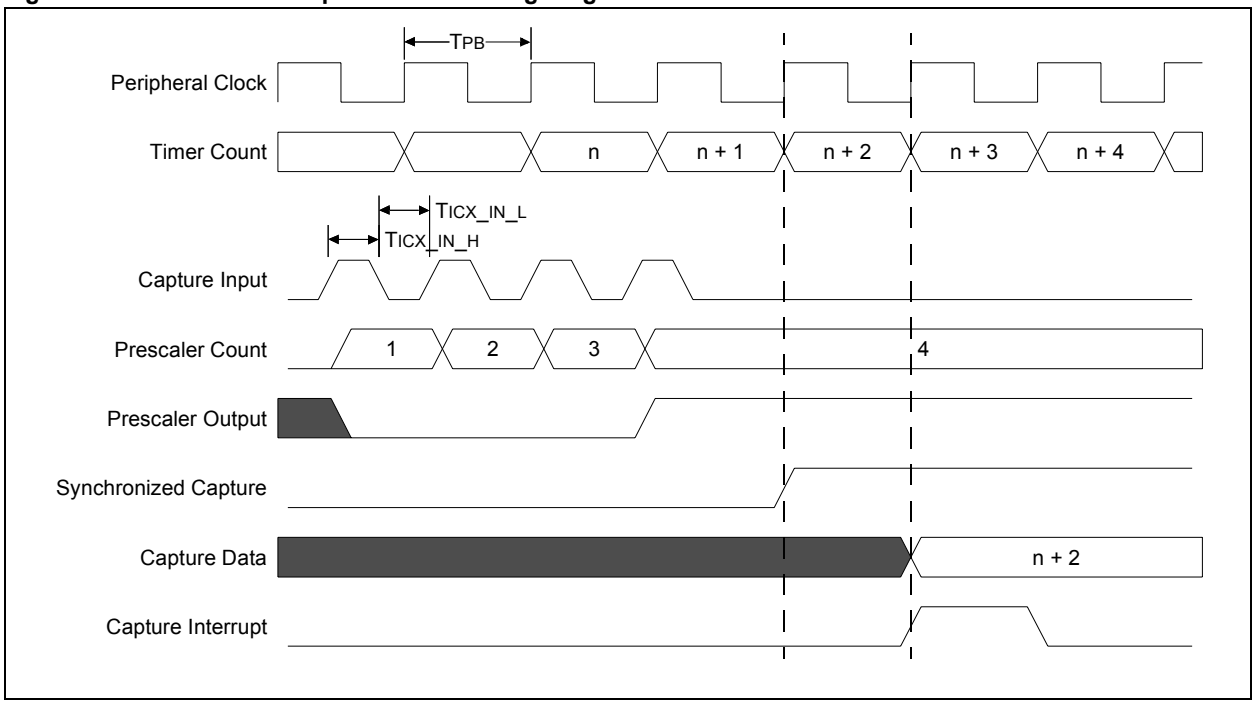
The prescaler counter is cleared when the following events occur:

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

Since the capture pin triggers an internal flip-flop, the input capture pulse high and low widths must be greater than TccL and TccH.

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 = 100 (ICxCON<2:0>), with interrupts generated for every capture event, ICI = 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



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 bit to '001'. In this mode, the capture prescaler is not used and the capture overflow bit, ICOV (ICxCON<4>), is not updated. In this mode, the Interrupt Control bits, ICI (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.

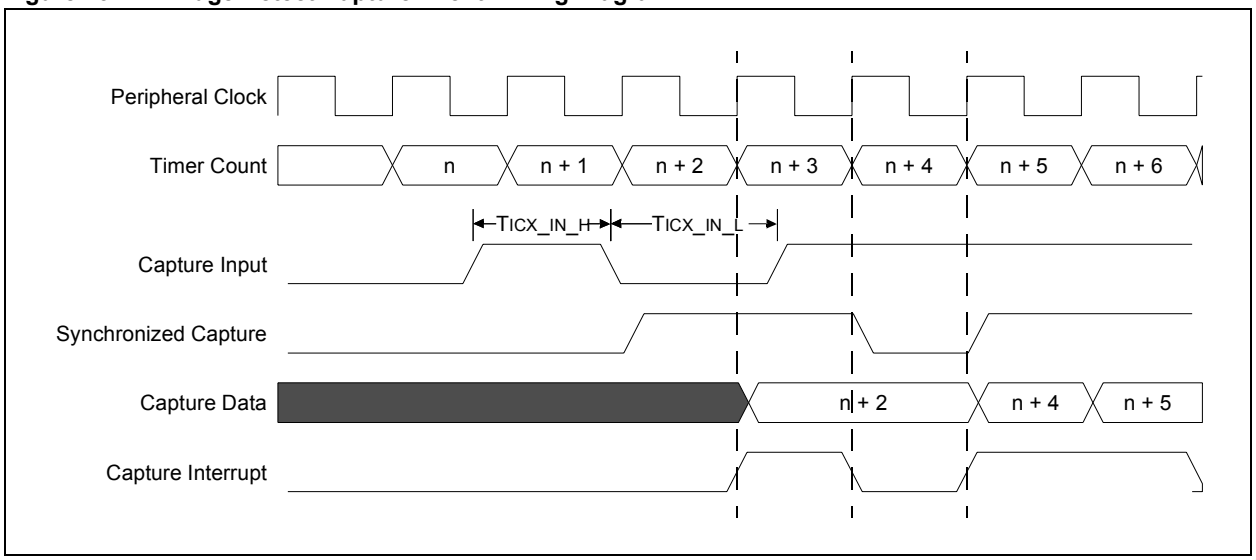
As with the Simple Capture Event mode, the Input Capture logic detects and synchronizes the rising and falling edge of the capture input signal on the peripheral clock. When a rising or falling edge occurs, the capture module writes the time base value to the capture buffer.

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 = 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



15.5.4 Interrupt Only Mode

When the Input Capture module is set for Interrupt Only mode (ICM = 111) and the device is in SLEEP or IDLE mode, the capture input functions as an interrupt pin. Any rising edge on the capture input triggers an interrupt. No timer values are captured and the FIFO buffer is not updated. When the device leaves SLEEP or IDLE mode, the interrupt signal is deasserted.

In this mode, 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. A wake-up interrupt is generated on the first rising edge. Therefore, the Interrupt Control bits, ICI (ICxCON<6:5>), are also ignored. The prescaler is not used in this mode.

Since the capture pin triggers an internal flip-flop, the capture pulse high and low widths must be greater than TccL and TccH.

15.6 CAPTURE BUFFER OPERATION

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

There are two status flags which 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 ICxCON<15>) = 0 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, then three reads of the capture FIFO buffer are required before the ICBNE flag is cleared. If four capture events have occurred, then 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, make sure you read the upper 16 bit last if you are reading 16 bits at a time. The FIFO read pointer is advanced when you read the 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 (ICxCON<4>) bit is set to a logic '1'. In addition, the fifth capture event is not recorded and subsequent capture events do not alter the current FIFO contents until the overrun condition is cleared.

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 reenabled, 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 control bits ICI (ICxCON<6:5>).

If ICBNE = 0 (ICxCON<3>), then the interrupt count is cleared. This allows the user to synchronize the interrupt count to the FIFO status.

For example, assume that ICI = 01, specifying an interrupt event every 2nd capture event.

1. Turn on module.
2. Interrupt count = 0.
3. Capture event. FIFO contains 1 entry.
4. Interrupt count = 1.
5. Read FIFO. FIFO is empty => interrupt count = 0.
6. Capture event. FIFO contains 1 entry.
7. Interrupt count = 1.
8. Capture event. FIFO contains 2 entries.
9. Interrupt count = 2. Issue interrupt.
10. Interrupt count = 0.
11. Capture event. FIFO contains 3 entries.
12. Interrupt count = 1.
13. Read FIFO 3 times.
14. FIFO is empty => interrupt count = 0.
15. Capture event. FIFO contains 1 entry.
16. Interrupt count = 1.
17. Read FIFO. FIFO is empty => interrupt count = 0.

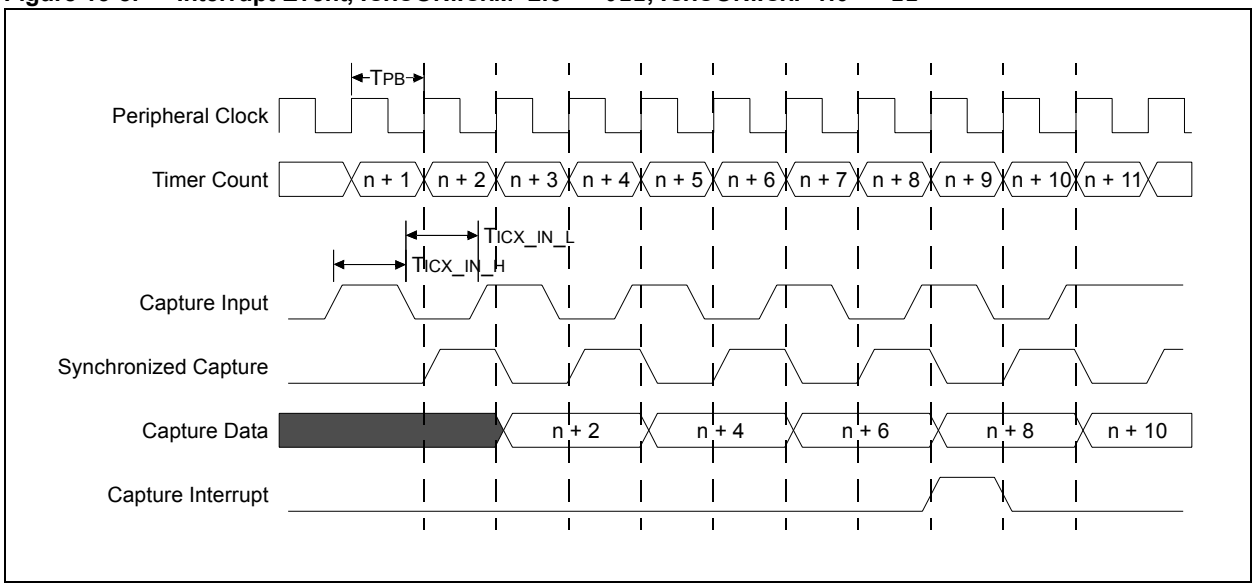
The first capture event is defined as the capture event occurring after a mode change from the OFF mode or after ICBNE = 0.

At overrun, the capture events cease, and therefore, the interrupt events stop – unless ICI = 00 or ICM = 001.

Applications often dictate using the Input Capture pins as auxiliary external interrupt sources. When ICI = 00 or ICM = 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 (ICxCON<4>) flag 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 = 011) and generate an interrupt for every four captures (ICI = 11). Note that the fourth capture causes the capture of value 'n + 8' and triggers an interrupt event.

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



15.7.1 Interrupt Control Bits

Each input capture channel has interrupt flag status bits (ICxIF), interrupt enable bits (ICxIE), interrupt priority control bits (ICxIP) and secondary interrupt priority control bits (ICxIS). Refer to **8.2 "Control Registers"** in **Section 1. "Interrupts"** for further information on peripheral interrupts.

15.8 OPERATION IN POWER-SAVING MODES

Note: In this manual, a distinction is made between a power mode as it is used in a specific module, and a power mode as it is used by the device, e.g., Sleep mode of the Comparator and SLEEP mode of the CPU. To indicate which type of power mode is intended, uppercase and lowercase letters (Sleep, Idle, Debug) signify a module power mode, and all uppercase letters (SLEEP, IDLE, DEBUG) signify a device power mode.

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 control bits ICM = 111 (ICxCON<2:0>). In this mode, a rising edge on the capture pin will generate a device wake-up from SLEEP condition. If the respective module interrupt bit is enabled and the module's priority is of the required priority level, an interrupt will be generated. See **15.5.4 "Interrupt Only Mode"** for more detail.

If the capture module has been configured for a mode other than ICM = 111 and the device does enter the SLEEP mode, no external pin stimulus, rising or falling, will generate a wake-up from SLEEP event.

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. Interrupt only mode (ICM = 111) may generate an interrupt when in IDLE if SIDL is '0'. See **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. See **15.5.4 "Interrupt Only Mode"** for further details.

15.8.3 Device Wake-up on SLEEP/IDLE

An input capture event can generate a device wake-up or interrupt, if enabled, when the device is in IDLE or SLEEP mode. See **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 Timer 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 module will freeze its operations and make no changes to the state of the Input Capture module. 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 input does not cause changes to the module. The edge detection logic runs during Freeze so that any state changes that occur during Freeze will not be inadvertently detected after leaving Freeze.

Clocks to all of the logic within the Input Capture module, with the exception of the SFR logic and the FIFO read logic, are conditioned on 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 do not change.

15.9.2 Operation of the Capture Buffer in Debug Mode

During DEBUG mode, reads from the capture buffer become circular. Reading ICxBUF adjusts only the DEBUG FIFO pointers; no status flags are affected by reads. This allows the DEBUG software to have visibility of the full contents of the FIFO. To enable this, the hardware contains two sets of ICxBUF FIFO pointers: an operating mode set and a DEBUG mode set.

15.10 I/O PIN CONTROL

When the capture module is enabled, the user 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 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 = 111 (ICxCON<2:0>) and the respective channel interrupt enable bit is asserted (ICIE = 1; see Interrupt registers IE0-IE2), a rising edge on the capture pin will wake-up 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.

NOTES:

