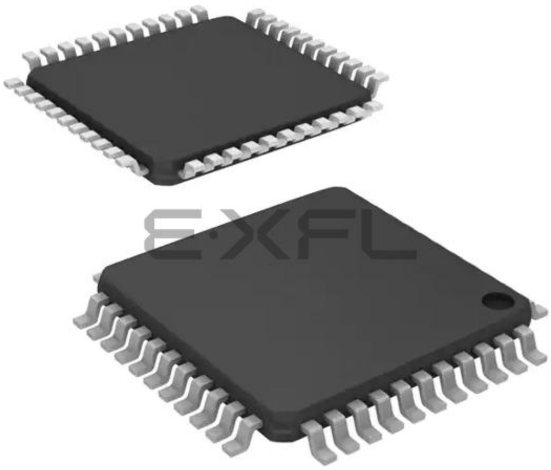


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 35x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf45k42-i-pt

REGISTER 7-2: OSCCON2: OSCILLATOR CONTROL REGISTER 2

U-0	R-f/f	R-f/f	R-f/f	R-f/f	R-f/f	R-f/f	R-f/f
—	COSC<2:0>			CDIV<3:0>			
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **COSC<2:0>:** Current Oscillator Source Select bits (read-only)(1)
Indicates the current source oscillator and PLL combination per Table 7-1.

bit 3-0 **CDIV<3:0>:** Current Divider Select bits (read-only)(1)
Indicates the current postscaler division ratio per Table 7-1.

Note 1: The POR value is the value present when user code execution begins.

REGISTER 7-3: OSCCON3: OSCILLATOR CONTROL REGISTER 3

R/W/HC-0/0	R/W-0/0	U-0	R-0/0	R-0/0	U-0	U-0	U-0
CSWHOLD	SOSCPWR	—	ORDY	NOSCR	—	—	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	HC = Bit is cleared by hardware

bit 7 **CSWHOLD:** Clock Switch Hold bit
1 = Clock switch will hold (with interrupt) when the oscillator selected by NOSC is ready
0 = Clock switch may proceed when the oscillator selected by NOSC is ready; NOSCR becomes '1', the switch will occur

bit 6 **SOSCPWR:** Secondary Oscillator Power Mode Select bit
1 = Secondary oscillator operating in High-Power mode
0 = Secondary oscillator operating in Low-Power mode

bit 5 **Unimplemented:** Read as '0'

bit 4 **ORDY:** Oscillator Ready bit (read-only)
1 = OSCCON1 = OSCCON2; the current system clock is the clock specified by NOSC
0 = A clock switch is in progress

bit 3 **NOSCR:** New Oscillator is Ready bit (read-only)(1)
1 = A clock switch is in progress and the oscillator selected by NOSC indicates a "ready" condition
0 = A clock switch is not in progress, or the NOSC-selected oscillator is not yet ready

bit 2-0 **Unimplemented:** Read as '0'

Note 1: If CSWHOLD = 0, the user may not see this bit set because, when the oscillator becomes ready there may be a delay of one instruction clock before this bit is set. The clock switch occurs in the next instruction cycle and this bit is cleared.

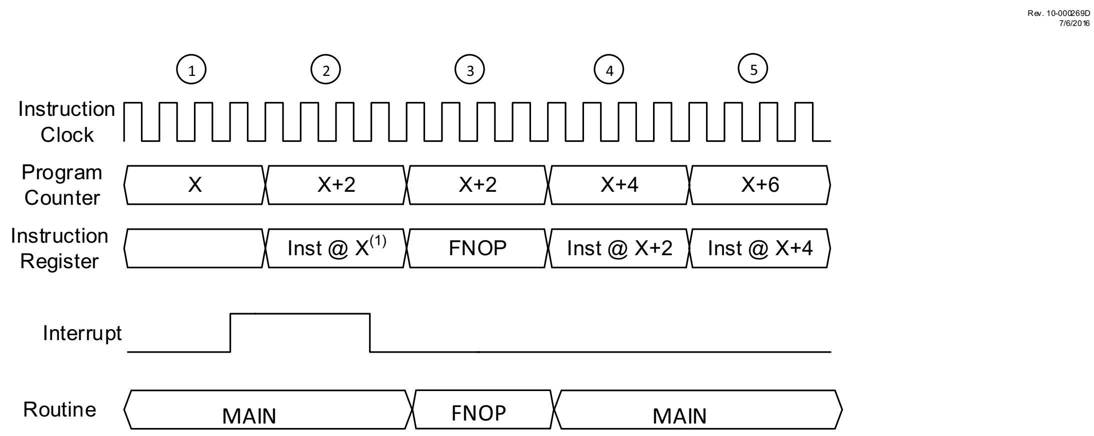
9.7.1 ABORTING INTERRUPTS

If the last instruction before the interrupt controller vectors to the ISR from main routine clears the GIE, PIE or PIR bit associated with the interrupt, the controller executes one force `NOP` cycle before it returns to the main routine.

Figure 9-10 illustrates the sequence of events when a peripheral interrupt is asserted and then cleared on the last executed instruction cycle.

If the GIE, PIE or PIR bit associated with the interrupt is cleared prior to vectoring to the ISR, then the controller continues executing the main routine.

FIGURE 9-10: INTERRUPT TIMING DIAGRAM - ABORTING INTERRUPTS



Note 1: Inst @ X clears the interrupt flag, Example BCF INTCON0, GIE.

Preliminary

REGISTER 9-13: PIR10: PERIPHERAL INTERRUPT REGISTER 10(1)

U-0	U-0	U-0	U-0	U-0	U-0	R/W/HS-0/0	R/W/HS-0/0
—	—	—	—	—	—	CLC4IF	CCP4IF
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	HS = Bit is set in hardware

bit 7-2 **Unimplemented:** Read as ‘0’

bit 1 **CLC4IF:** CLC4 Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

bit 0 **CCP4IF:** CCP4 Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

Note 1: Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit, or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 9-29: IPR4: PERIPHERAL INTERRUPT PRIORITY REGISTER 4

R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
CLC1IP	CWG1IP	NCO1IP	—	CCP1IP	TMR2IP	TMR1GIP	TMR1IP
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7 **CLC1IP:** CLC1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **CWG1IP:** CWG1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **NCO1IP:** NCO1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **Unimplemented:** Read as ‘0’

bit 3 **CCP1IP:** CCP1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **TMR2IP:** TMR2 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **TMR1GIP:** TMR1 Gate Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **TMR1IP:** TMR1 Interrupt Priority bit
1 = High priority
0 = Low priority

REGISTER 15-9: DMAxSPTRU: DMAx SOURCE POINTER UPPER REGISTER

U-0	U-0	R-0	R-0	R-0	R-0	R-0	R-0
—	—	SPTR<21:16>					
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n/n = Value at POR and BOR/Value at all other Resets	1 = bit is set	0 = bit is cleared	x = bit is unknown u = bit is unchanged

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **SPTR<21:16>:** Current Source Address Pointer

REGISTER 15-10: DMAxSSZL: DMAx SOURCE SIZE LOW REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
SSZ<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n/n = Value at POR and BOR/Value at all other Resets	1 = bit is set	0 = bit is cleared	x = bit is unknown u = bit is unchanged

bit 7-0 **SSZ<7:0>:** Source Message Size bits

REGISTER 15-11: DMAxSSZH: DMAx SOURCE SIZE HIGH REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	SSZ<11:8>			
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n/n = Value at POR and BOR/Value at all other Resets	1 = bit is set	0 = bit is cleared	x = bit is unknown u = bit is unchanged

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **SSZ<11:8>:** Source Message Size bits

23.2 Capture Mode

Capture mode makes use of the 16-bit Timer1 resource. When an event occurs on the capture source, the 16-bit CCPRxH:CCPRxL register pair captures and stores the 16-bit value of the TMRxH:TMRxL register pair, respectively. An event is defined as one of the following and is configured by the MODE<3:0> bits of the CCPxCON register:

- Every falling edge of CCPx input
- Every rising edge of CCPx input
- Every 4th rising edge of CCPx input
- Every 16th rising edge of CCPx input
- Every edge of CCPx input (rising or falling)

When a capture is made, the Interrupt Request Flag bit CCPxIF of the respective PIR register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPRxH:CCPRxL register pair is read, the old captured value is overwritten by the new captured value.

Note:	If an event occurs during a 2-byte read, the high and low-byte data will be from different events. It is recommended while reading the CCPRxH:CCPRxL register pair to either disable the module or read the register pair twice for data integrity.

Figure 23-1 shows a simplified diagram of the capture operation.

23.2.1 CAPTURE SOURCES

In Capture mode, the CCPx pin should be configured as an input by setting the associated TRIS control bit.

Note:	If the CCPx pin is configured as an output, a write to the port can cause a capture condition.

The capture source is selected by configuring the CTS<2:0> bits of the CCPxCAP register. Refer to CCPxCAP register (Register 23-3) for a list of sources that can be selected.

23.2.2 TIMER1 MODE RESOURCE

Timer1 must be running in Timer mode or Synchronized Counter mode for the CCP module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

- See **Section 21.0 “Timer1/3/5 Module with Gate Control”** for more information on configuring Timer1.

Note:	Clocking Timer1 from the system clock (FOSC) should not be used in Capture mode. In order for Capture mode to recognize the trigger event on the CCPx pin, Timer1 must be clocked from the instruction clock (FOSC/4) or from an external clock source.

24.1.5 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is ten bits when T2PR is 255. The resolution is a function of the T2PR register value as shown by Equation 24-4.

EQUATION 24-4: PWM RESOLUTION

$$Resolution = \frac{\log[4(T2PR + 1)]}{\log(2)}\ bits$$

Note:	If the pulse-width value is greater than the period, the assigned PWM pin(s) will remain unchanged.

TABLE 24-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 20 MHz)

PWM Frequency	0.31 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale	64	4	1	1	1	1
T2PR Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

TABLE 24-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 8 MHz)

PWM Frequency	0.31 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale	64	4	1	1	1	1
T2PR Value	0x65	0x65	0x65	0x19	0x0C	0x09
Maximum Resolution (bits)	8	8	8	6	5	5

24.1.6 OPERATION IN SLEEP MODE

In Sleep mode, the T2TMR register will not increment and the state of the module will not change. If the PWMx pin is driving a value, it will continue to drive that value. When the device wakes up, T2TMR will continue from its previous state.

24.1.7 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency (FOSC). Any changes in the system clock frequency will result in changes to the PWM frequency. Refer to **Section 7.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for additional details.

24.1.8 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the PWM registers to their Reset states.

25.6.2 GATED TIMER MODE

Gated Timer mode uses the SMTSIGx input to control whether or not the SMT1TMR will increment. Upon a falling edge of the external signal, the SMT1CPW register will update to the current value of the SMT1TMR. Example waveforms for both repeated and single acquisitions are provided in Figure 25-4 and Figure 25-5.

FIGURE 25-9: HIGH AND LOW MEASURE MODE SINGLE ACQUISITION TIMING DIAGRAM

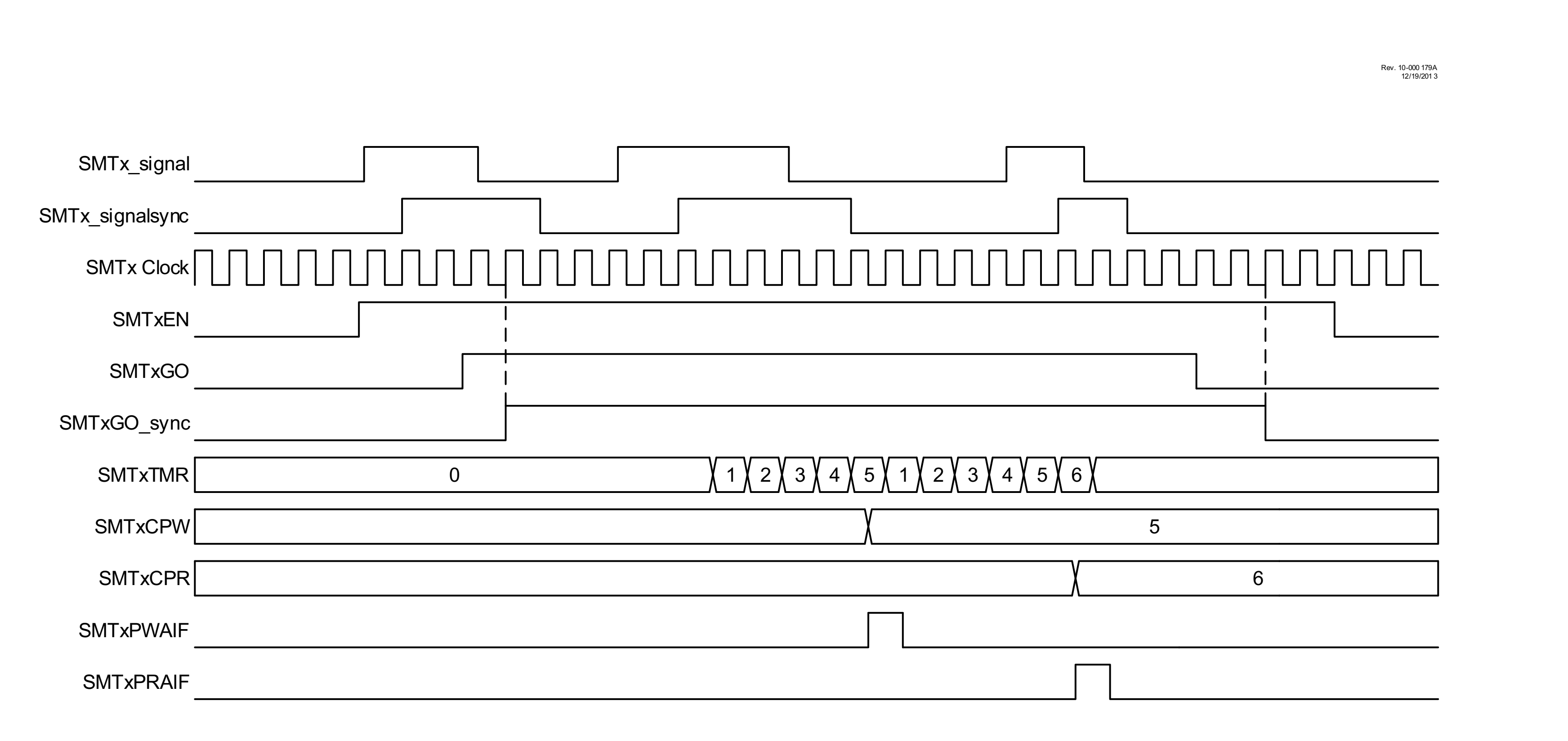
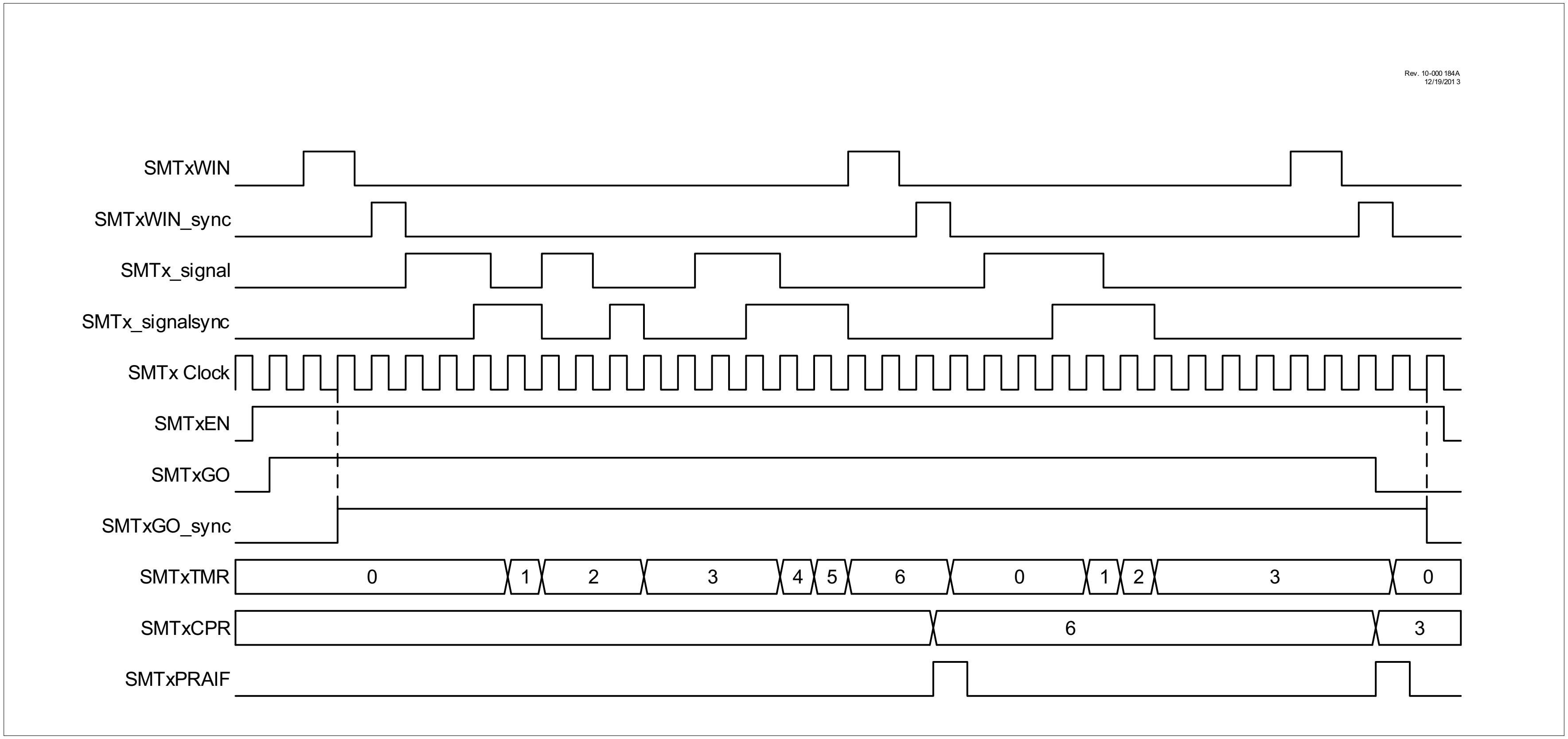


FIGURE 25-12: GATED WINDOWED MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM



25.6.11 WINDOWED COUNTER MODE

This mode counts pulses on the SMT1_signal input, within a window dictated by the SMT1WIN input. It begins counting upon seeing a rising edge of the SMT1WIN input, updates the SMT1CPW register on a falling edge of the SMT1WIN input, and updates the SMT1CPR register on each rising edge of the SMT1WIN input beyond the first. See Figure 25-21 and Figure 25-22.

REGISTER 26-4: CWGxISM: CWGx INPUT SELECTION REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	ISM<4:0>				
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	q = Value depends on condition

bit 7-5 **Unimplemented** Read as ‘0’

bit 4-0 **ISM<4:0>:** CWG Data Input Selection Multiplexer Select bits

ISM<4:0>	CWG1	CWG2	CWG3
	Input Selection	**Input Selection**	**Input Selection**
11111-10011	Reserved	Reserved	Reserved
10010	CLC4_out	CLC4_out	CLC4_out
10001	CLC3_out	CLC3_out	CLC3_out
10000	CLC2_out	CLC2_out	CLC2_out
01111	CLC1_out	CLC1_out	CLC1_out
01110	DSM_out	DSM_out	DSM_out
01101	CMP2OUT	CMP2OUT	CMP2OUT
01100	CMP1OUT	CMP1OUT	CMP1OUT
01011	NCO1OUT	NCO1OUT	NCO1OUT
01010-01001	Reserved	Reserved	Reserved
01000	PWM8OUT	PWM8OUT	PWM8OUT
00111	PWM7OUT	PWM7OUT	PWM7OUT
00110	PWM6OUT	PWM6OUT	PWM6OUT
00101	PWM5OUT	PWM5OUT	PWM5OUT
00100	CCP4_out	CCP4_out	CCP4_out
00011	CCP3_out	CCP3_out	CCP3_out
00010	CCP2_out	CCP2_out	CCP2_out
00001	CCP1_out	CCP1_out	CCP1_out
00000	Pin selected by CWG1PPS	Pin selected by CWG2PPS	Pin selected by CWG3PPS

29.0 ZERO-CROSS DETECTION (ZCD) MODULE

The ZCD module detects when an A/C signal crosses through the ground potential. The actual zero-crossing threshold is the zero-crossing reference voltage, VCPINV, which is typically 0.75V above ground.

The connection to the signal to be detected is through a series current-limiting resistor. The module applies a current source or sink to the ZCD pin to maintain a constant voltage on the pin, thereby preventing the pin voltage from forward biasing the ESD protection diodes. When the applied voltage is greater than the reference voltage, the module sinks current. When the applied voltage is less than the reference voltage, the module sources current. The current source and sink action keeps the pin voltage constant over the full range of the applied voltage. The ZCD module is shown in the simplified block diagram Figure 29-2.

The ZCD module is useful when monitoring an A/C waveform for, but not limited to, the following purposes:

- A/C period measurement
- Accurate long term time measurement
- Dimmer phase delayed drive
- Low EMI cycle switching

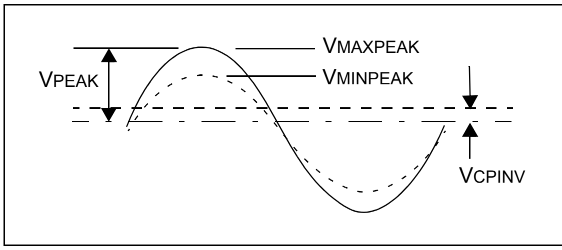
29.1 External Resistor Selection

The ZCD module requires a current-limiting resistor in series with the external voltage source. The impedance and rating of this resistor depends on the external source peak voltage. Select a resistor value that will drop all of the peak voltage when the current through the resistor is nominally 300 µA. Refer to Equation 29-1 and Figure 29-1. Make sure that the ZCD I/O pin internal weak pull-up is disabled so it does not interfere with the current source and sink.

EQUATION 29-1: EXTERNAL RESISTOR

$$R_{SERIES} = \frac{V_{PEAK}}{3 \times 10^{-4}}$$

FIGURE 29-1: EXTERNAL VOLTAGE



30.0 DATA SIGNAL MODULATOR (DSM) MODULE

The Data Signal Modulator (DSM) is a peripheral which allows the user to mix a data stream, also known as a modulator signal, with a carrier signal to produce a modulated output.

Both the carrier and the modulator signals are supplied to the DSM module either internally, from the output of a peripheral, or externally through an input pin.

The modulated output signal is generated by performing a logical “AND” operation of both the carrier and modulator signals and then provided to the MDOUT pin.

The carrier signal is comprised of two distinct and separate signals. A carrier high (CARH) signal and a carrier low (CARL) signal. During the time in which the modulator (MOD) signal is in a logic high state, the DSM mixes the carrier high signal with the modulator signal. When the modulator signal is in a logic low state, the DSM mixes the carrier low signal with the modulator signal.

Using this method, the DSM can generate the following types of Key Modulation schemes:

- Frequency-Shift Keying (FSK)
- Phase-Shift Keying (PSK)
- On-Off Keying (OOK)

Additionally, the following features are provided within the DSM module:

- Carrier Synchronization
- Carrier Source Polarity Select
- Programmable Modulator Data
- Modulated Output Polarity Select
- Peripheral Module Disable, which provides the ability to place the DSM module in the lowest power consumption mode

Figure 30-1 shows a Simplified Block Diagram of the Data Signal Modulator peripheral.

REGISTER 31-14: UxP2H: UART PARAMETER 2 HIGH REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	P2<8>
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **Unimplemented:** Read as ‘0’

bit 0 **P2<8>:** Most Significant Bit of Parameter 2
<u>DMX mode</u>:
Most Significant bit of first address of receive block
<u>DALI mode</u>:
Most Significant bit of number of half-bit periods of idle time in Forward Frame detection threshold
<u>Other modes</u>:
Not used

REGISTER 31-15: UxP2L: UART PARAMETER 2 LOW REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
P2<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-0 **P2<7:0>:** Least Significant Bits of Parameter 2
<u>DMX mode</u>:
Least Significant Byte of first address of receive block
<u>LIN Slave mode</u>:
Number of data bytes to transmit
<u>DALI mode</u>:
Least Significant Byte of number of half-bit periods of idle time in Forward Frame detection threshold
<u>Asynchronous Address mode</u>:
Receiver address
<u>Other modes</u>:
Not used

The SPI transmit output (SDO_out) is available to the remappable PPS SDO pin and internally to the following peripherals:

- Configurable Logic Cell (CLC)
- Data Signal Modulator (DSM)

The SPI bus typically operates with a single master device and one or more slave devices. When multiple slave devices are used, an independent Slave Select connection is required from the master device to each slave device.

The master selects only one slave at a time. Most slave devices have tri-state outputs so their output signal appears disconnected from the bus when they are not selected.

Transmissions typically involve shift registers, eight bits in size, one in the master and one in the slave. With either the master or the slave device, data is always shifted out one bit at a time, with the Most Significant bit (MSb) shifted out first. At the same time, a new bit is shifted into the device. Unlike older Microchip devices, the SPI on the PIC18(L)F2X/4X/5XK42 contains two separate registers for incoming and outgoing data. Both registers also have 2-byte FIFO buffers and allow for DMA bus connections.

Figure 32-2 shows a typical connection between two PIC18F2X/4X/5XK42 devices configured as master and slave devices.

Data is shifted out of the transmit FIFO on the programmed clock edge and into the receive shift register on the opposite edge of the clock.

The master device transmits information on its SDO output pin which is connected to, and received by, the slave's SDI input pin. The slave device transmits information on its SDO output pin, which is connected to, and received by, the master's SDI input pin.

The master device sends out the clock signal. Both the master and the slave devices should be configured for the same clock polarity.

During each SPI clock cycle, a full-duplex data transmission occurs. This means that while the master device is sending out the MSb from its output register (on its SDO pin) and the slave device is reading this bit and saving as the LSb of its input register, that the slave device is also sending out the MSb from its shift register (on its SDO pin) and the master device is reading this bit and saving it as the LSb of its input register.

After eight bits have been shifted out, the master and slave have exchanged register values and stored the incoming data into the receiver FIFOs.

If there is more data to exchange, the registers are loaded with new data and the process repeats itself.

Whether the data is meaningful or not (dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends useful data and slave sends dummy data
- Master sends useful data and slave sends useful data
- Master sends dummy data and slave sends useful data

In this particular SPI module, dummy data may be sent without software involvement, by clearing either the RXR bit (for receiving dummy data) or the TXR bit (for sending dummy data) (see Table 32-1 as well as **Section 32.5 "Master mode"** and **Section 32.6 "Slave Mode"** for further TXR/RXR setting details). This SPI module can send transmissions of any number of bits, and can send information in segments of varying size (from 1-8 bits in width). As such, transmissions may involve any number of clock cycles, depending on the amount of data to be transmitted.

When there is no more data to be transmitted, the master stops sending the clock signal and deselects the slave.

Every slave device connected to the bus that has not been selected through its slave select line disregards the clock and transmission signals and does not transmit out any data of its own.

32.3 SPI MODE OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SPIxCON0<2:0>, SPIxCON1<7:4>, SPIxCON1<2:0>, and SPIxCON2<2:0>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Input, Output, and Slave Select Polarity
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on first/second edge of SCK)
- Clock Rate (Master mode only)
- Slave Select Mode (Master or Slave mode)
- MSB-First or LSB-First
- Receive/Transmit Modes
  - Full duplex
  - Receive-without-transmit
  - Transmit-without-receive
- Transfer Counter Mode (Transmit-without-receive mode)

32.3.1 ENABLING AND DISABLING THE SPI MODULE

To enable the serial peripheral, the SPI enable bit (EN in SPIxCON0) must be set. To reset or reconfigure SPI mode, clear the EN bit, re-initialize the SSPxCONx registers and then set the EN bit. Setting the EN bit enables the SPI inputs and outputs: SDI, SDO, SCK(out), SCK(in), SS(out), and SS(in). All of these inputs and outputs are steered by PPS, and thus must have their functions properly mapped to device pins to function (see **Section 17.0 "Peripheral Pin Select (PPS) Module"**). In addition, SS(out) and SCK(out) must have the pins they are steered to set as outputs (TRIS bits must be '0') in order to properly output. Clearing the TRIS bit of the SDO pin will cause the SPI module to always control that pin, but is not necessary for SDO functionality. (see **Section 32.3.5 "Input and Output Polarity Bits"**). Configurations selected by the following registers should not be changed while the EN bit is set:

- SPIxBAUD
- SPIxCON1
- SPIxCON0 (except to clear the EN bit)

Clearing the EN bit aborts any transmissions in progress, disables the setting of interrupt flags by hardware, and resets the FIFO occupancy (see **Section 32.3.3 "Transmit and Receive FIFOs"** for more FIFO details).

32.3.2 BUSY BIT

While a data transfer is in progress, the SPI module sets the BUSY bit of SPIxCON2. This bit can be polled by the user to determine the current status of the SPI module, and to know when a communication is complete. The following registers/bits should not be written by software while the BUSY bit is set:

- SPIxTCNTH/L
- SPIxTWIDTH
- SPIxCON2
- The CLRBF bit of SPIxSTATUS

Note:	It is also not recommended to read SPIx-TCNTH/L while the BUSY bit is set, as the value in the registers may not be a reliable indicator of the Transfer Counter. Use the Transfer Count Zero Interrupt Flag (the TCZIF bit of SPIxINTF) to accurately determine that the Transfer Counter has reached zero.

FIGURE 32-7: CLOCKING DETAIL-MASTER MODE, CKE/SMP = 0/0

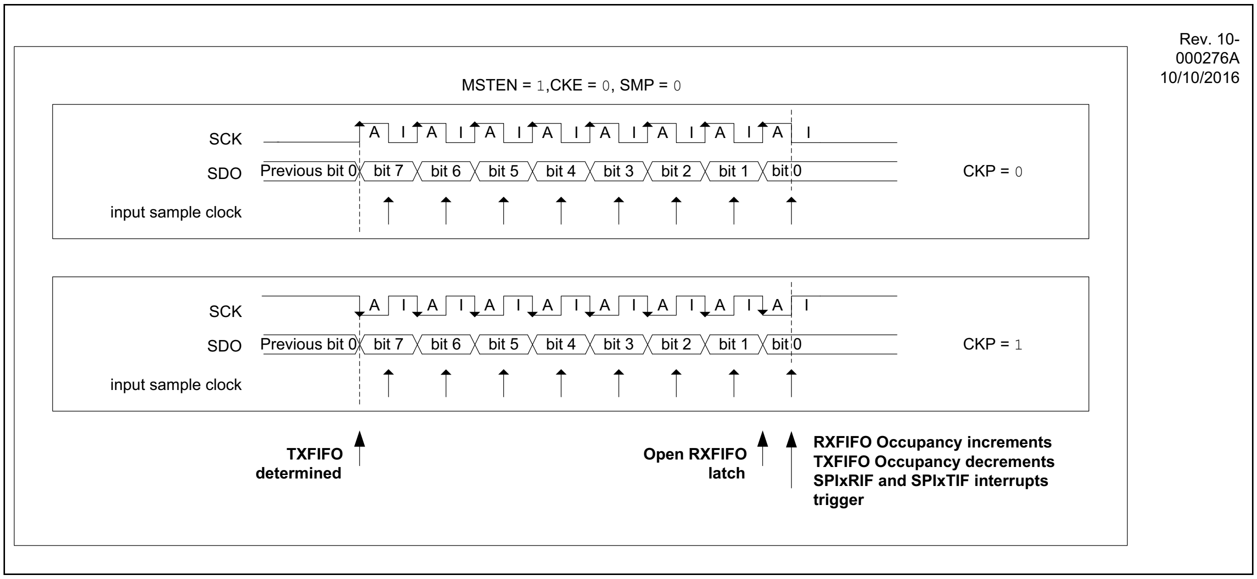
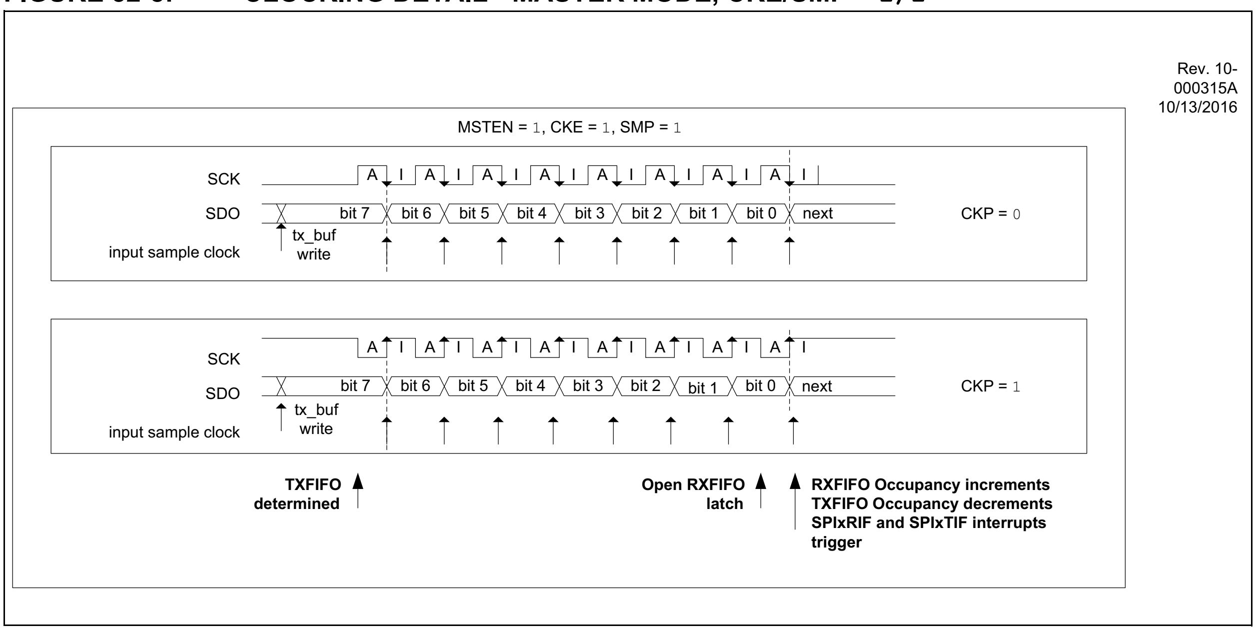


FIGURE 32-8: CLOCKING DETAIL - MASTER MODE, CKE/SMP = 1/1



bit 3 **MDR:** Master Data Request *(Master pause)*

1 = Master state mechine pauses until data is read/written to proceed (SCL is output held low)
0 = Master clocking of data is enabled.

MMA = 1 & RXBF = 1
pause_for_rx - Set by hardware on 7th falling SCL edge
- User must read from I2CRXB to release SCL

MMA = 1 & TXBE = 1 & I2CCNT!= 0
pause_for_tx - Set by hardware on 8th falling SCL edge
- User must write to I2CTXB to release SCL

ADB = 1
- I2CCNT is ignored for the high and low address in 10-bit mode

pause_for_restart - Set by hardware on 9th falling SCL edge
RSEN = 1 & MMA = 1 && I2CCNT = 0 || ACKSTAT = 1
- User must set START or write to I2CTXB to release SCL and shift Restart onto bus

bit 2-0 **MODE<2:0>:** I^{2}C Mode Select bits

111 = I^{2}C Muti-Master mode (SMBus 2.0 Host), [5]
Works as both mode<2:0> = 001 and mode<2:0> = 100
110 = I^{2}C Muti-Master mode (SMBus 2.0 Host), [5]
Works as both mode<2:0> = 000 and mode<2:0> = 100
101 = I^{2}C Master mode, 10-bit address
100 = I^{2}C Master mode, 7-bit address
011 = I^{2}C Slave mode, one 10-bit address with masking
010 = I^{2}C Slave mode, two 10-bit address
001 = I^{2}C Slave mode, two 7-bit address with masking
000 = I^{2}C Slave mode, four 7-bit address

Note 1: SDA and SCL pins must be configured for open-drain with internal or external pull-up
2: SDA and SCL pins must be selected as both input and output in PPS
3: CSTR can be set by more than one hardware source, all sources must be addressed by user software before the SCL line is released. CSTR is a module status bit, and does not show the true bus state.
4: SMA is set on the same SCL edge as CSTR for a matching received address
5: In this mode, ADRIE should be set, this allows an interrupt to clear the BCLIF condition and allow the ACK of matching address.
6: In 10-bit Slave mode, when ADB = 1, CSTR will set when the high address has not been read out of I2CxRXB before the low address is shifted in.

REGISTER 36-4: ADCON3: ADC CONTROL REGISTER 3

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W/HC-0	R/W-0/0	R/W-0/0	R/W-0/0
—	CALC<2:0>			SOI	TMD<2:0>		
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	HC = Bit is cleared by hardware

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **CALC<2:0>:** ADC Error Calculation Mode Select bits

CALC	DSEN = 0 Single-Sample Mode	DSEN = 1 CVD Double-Sample Mode(1)	Application
111	Reserved	Reserved	Reserved
110	Reserved	Reserved	Reserved
101	FLTR-STPT	FLTR-STPT	Average/filtered value vs. setpoint
100	PREV-FLTR	PREV-FLTR	First derivative of filtered value(3) (negative)
011	Reserved	Reserved	Reserved
010	RES-FLTR	(RES-PREV)-FLTR	Actual result vs. averaged/ filtered value
001	RES-STPT	(RES-PREV)-STPT	Actual result vs.setpoint
000	RES-PREV	RES-PREV	First derivative of single measurement(2)
			Actual CVD result in CVD mode(2)

bit 3 **SOI:** ADC Stop-on-Interrupt bit

If CONT = 1:
1 = GO is cleared when the threshold conditions are met, otherwise the conversion is retriggered
0 = GO is not cleared by hardware, must be cleared by software to stop retriggers

bit 2-0 **TMD<2:0>:** Threshold Interrupt Mode Select bits

111 = Interrupt regardless of threshold test results
110 = Interrupt if ERR>UTH
101 = Interrupt if ERR≤UTH
100 = Interrupt if ERR<LTH or ERR>UTH
011 = Interrupt if ERR>LTH and ERR<UTH
010 = Interrupt if ERR≥LTH
001 = Interrupt if ERR<LTH
000 = Never interrupt

Note 1: When PSIS = 0, the value of (RES-PREV) is the value of (S2-S1) from Table 36-2.
2: When PSIS = 0
3: When PSIS = 1.