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	40MHz
Connectivity	CANbus, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.25K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f2680-i-so

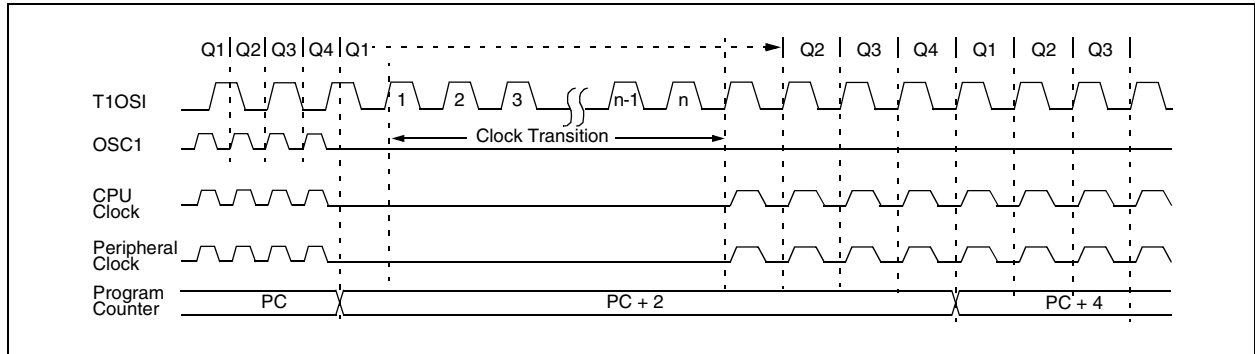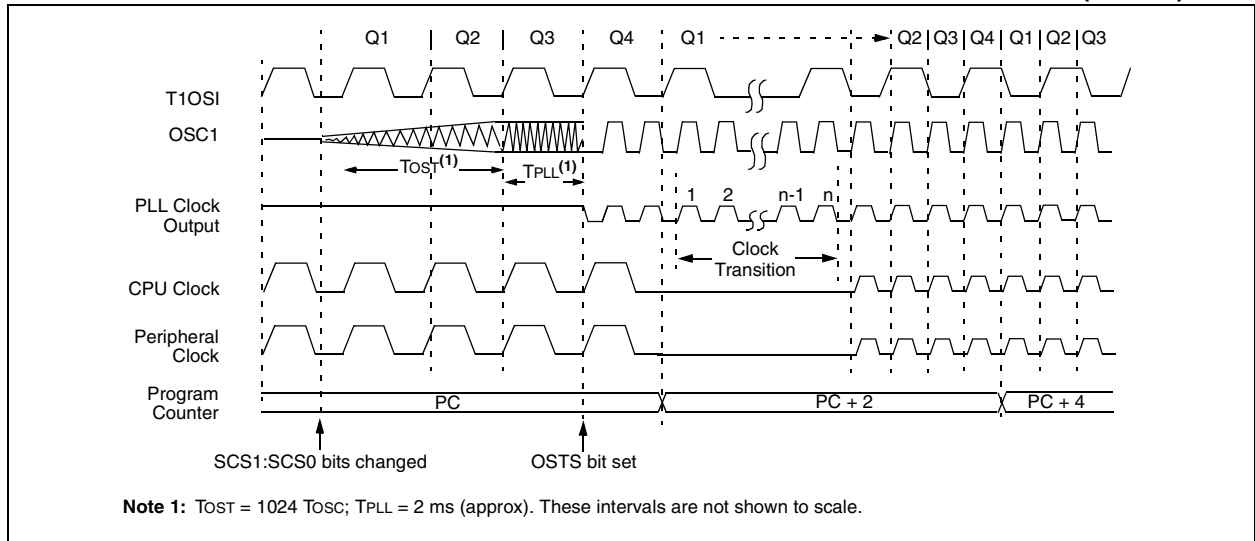**FIGURE 3-1:** **TRANSITION TIMING FOR ENTRY TO SEC_RUN MODE**

FIGURE 3-2: **TRANSITION TIMING FROM SEC_RUN MODE TO PRI_RUN MODE (HSPLL)**

Note 1: $T_{OST} = 1024\ T_{OSC}$; $T_{PLL} = 2$ ms (approx). These intervals are not shown to scale.

3.2.3 RC_RUN MODE

In RC_RUN mode, the CPU and peripherals are clocked from the internal oscillator block using the INTOSC multiplexer; the primary clock is shut down. When using the INTRC source, this mode provides the best power conservation of all the Run modes, while still executing code. It works well for user applications which are not highly timing sensitive or do not require high-speed clocks at all times.

If the primary clock source is the internal oscillator block (either INTRC or INTOSC), there are no distinguishable differences between PRI_RUN and RC_RUN modes during execution. However, a clock switch delay will occur during entry to and exit from RC_RUN mode. Therefore, if the primary clock source is the internal oscillator block, the use of RC_RUN mode is not recommended.

This mode is entered by setting SCS1 to '1'. Although it is ignored, it is recommended that SCS0 also be cleared; this is to maintain software compatibility with future devices. When the clock source is switched to the INTOSC multiplexer (see Figure 3-3), the primary oscillator is shut down and the OSTS bit is cleared. The IRCF bits may be modified at any time to immediately change the clock speed.

> **Note:** Caution should be used when modifying a single IRCF bit. If V_{DD} is less than 3V, it is possible to select a higher clock speed than is supported by the low V_{DD}. Improper device operation may result if the V_{DD}/F_{OSC} specifications are violated.

TABLE 4-4: **INITIALIZATION CONDITIONS FOR ALL REGISTERS**

Register	Applicable Devices				Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt
TOSU	2585	2680	4585	4680	---0 0000	---0 0000	---0 uuuu[3]
TOSH	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu[3]
TOSL	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu[3]
STKPTR	2585	2680	4585	4680	00-0 0000	uu-0 0000	uu-u uuuu[3]
PCLATU	2585	2680	4585	4680	---0 0000	---0 0000	---u uuuu
PCLATH	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
PCL	2585	2680	4585	4680	0000 0000	0000 0000	PC + 2[2]
TBLPTRU	2585	2680	4585	4680	--00 0000	--00 0000	--uu uuuu
TBLPTRH	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
TBLPTRL	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
TABLAT	2585	2680	4585	4680	0000 0000	0000 0000	uuuu uuuu
PRODH	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODL	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
INTCON	2585	2680	4585	4680	0000 000x	0000 000u	uuuu uuuu[1]
INTCON2	2585	2680	4585	4680	1111 -1-1	1111 -1-1	uuuu -u-u[1]
INTCON3	2585	2680	4585	4680	11-0 0-00	11-0 0-00	uu-u u-uu[1]
INDF0	2585	2680	4585	4680	N/A	N/A	N/A
POSTINC0	2585	2680	4585	4680	N/A	N/A	N/A
POSTDEC0	2585	2680	4585	4680	N/A	N/A	N/A
PREINC0	2585	2680	4585	4680	N/A	N/A	N/A
PLUSW0	2585	2680	4585	4680	N/A	N/A	N/A
FSR0H	2585	2680	4585	4680	---- 0000	---- 0000	---- uuuu
FSR0L	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF1	2585	2680	4585	4680	N/A	N/A	N/A
POSTINC1	2585	2680	4585	4680	N/A	N/A	N/A
POSTDEC1	2585	2680	4585	4680	N/A	N/A	N/A
PREINC1	2585	2680	4585	4680	N/A	N/A	N/A
PLUSW1	2585	2680	4585	4680	N/A	N/A	N/A
FSR1H	2585	2680	4585	4680	---- 0000	---- 0000	---- uuuu
FSR1L	2585	2680	4585	4680	xxxx xxxx	uuuu uuuu	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 4-3 for Reset value for specific condition.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled, depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

6: This register reads all '0's until the ECAN™ technology is set up in Mode 1 or Mode 2.

9.1   INTCON Registers

The INTCON registers are readable and writable registers, which contain various enable, priority and flag bits.

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

REGISTER 9-1:   INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF

bit 7   bit 0

bit 7   **GIE/GIEH:** Global Interrupt Enable bit

When IPEN = 0:
1 = Enables all unmasked interrupts
0 = Disables all interrupts

When IPEN = 1:
1 = Enables all high priority interrupts
0 = Disables all high priority interrupts

bit 6   **PEIE/GIEL:** Peripheral Interrupt Enable bit

When IPEN = 0:
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts

When IPEN = 1:
1 = Enables all low priority peripheral interrupts
0 = Disables all low priority peripheral interrupts

bit 5   **TMR0IE:** TMR0 Overflow Interrupt Enable bit

1 = Enables the TMR0 overflow interrupt
0 = Disables the TMR0 overflow interrupt

bit 4   **INT0IE:** INT0 External Interrupt Enable bit

1 = Enables the INT0 external interrupt
0 = Disables the INT0 external interrupt

bit 3   **RBIE:** RB Port Change Interrupt Enable bit

1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2   **TMR0IF:** TMR0 Overflow Interrupt Flag bit

1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1   **INT0IF:** INT0 External Interrupt Flag bit

1 = The INT0 external interrupt occurred (must be cleared in software)
0 = The INT0 external interrupt did not occur

bit 0   **RBIF:** RB Port Change Interrupt Flag bit

1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

     Note: A mismatch condition will continue to set this bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared.

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

REGISTER 9-6: **PIR3: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 3**

Mode 0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	IRXIF	WAKIF	ERRIF	TXB2IF	TXB1IF[1]	TXB0IF[1]	RXB1IF	RXB0IF

Mode 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
	IRXIF	WAKIF	ERRIF	TXBnIF	TXB1IF[1]	TXB0IF[1]	RXBnIF	FIFOWMIF

bit 7   bit 0

bit 7 **IRXIF:** CAN Invalid Received Message Interrupt Flag bit

1 = An invalid message has occurred on the CAN bus
0 = No invalid message on CAN bus

bit 6 **WAKIF:** CAN bus Activity Wake-up Interrupt Flag bit

1 = Activity on CAN bus has occurred
0 = No activity on CAN bus

bit 5 **ERRIF:** CAN bus Error Interrupt Flag bit

1 = An error has occurred in the CAN module (multiple sources)
0 = No CAN module errors

bit 4 When CAN is in Mode 0:
TXB2IF: CAN Transmit Buffer 2 Interrupt Flag bit

1 = Transmit Buffer 2 has completed transmission of a message and may be reloaded
0 = Transmit Buffer 2 has not completed transmission of a message

When CAN is in Mode 1 or 2:
TXBnIF: Any Transmit Buffer Interrupt Flag bit

1 = One or more transmit buffers have completed transmission of a message and may be reloaded
0 = No transmit buffer is ready for reload

bit 3 **TXB1IF:** CAN Transmit Buffer 1 Interrupt Flag bit[1]

1 = Transmit Buffer 1 has completed transmission of a message and may be reloaded
0 = Transmit Buffer 1 has not completed transmission of a message

bit 2 **TXB0IF:** CAN Transmit Buffer 0 Interrupt Flag bit[1]

1 = Transmit Buffer 0 has completed transmission of a message and may be reloaded
0 = Transmit Buffer 0 has not completed transmission of a message

bit 1 When CAN is in Mode 0:
RXB1IF: CAN Receive Buffer 1 Interrupt Flag bit

1 = Receive Buffer 1 has received a new message
0 = Receive Buffer 1 has not received a new message

When CAN is in Mode 1 or 2:
RXBnIF: Any Receive Buffer Interrupt Flag bit

1 = One or more receive buffers has received a new message
0 = No receive buffer has received a new message

bit 0 When CAN is in Mode 0:
RXB0IF: CAN Receive Buffer 0 Interrupt Flag bit

1 = Receive Buffer 0 has received a new message
0 = Receive Buffer 0 has not received a new message

When CAN is in Mode 1:
Unimplemented: Read as '0'

When CAN is in Mode 2:
FIFOWMIF: FIFO Watermark Interrupt Flag bit

1 = FIFO high watermark is reached
0 = FIFO high watermark is not reached

Note 1: In CAN Mode 1 and 2, this bit is forced to '0'.

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

TABLE 10-3: PORTB I/O SUMMARY

Pin Name	Function	I/O	TRIS	Buffer	Description
RB0/INT0/FLT0/AN10	RB0	OUT	0	DIG	LATB<0> data output.
		IN	1	TTL	PORTB<0> data input. Weak pull-up available only in this mode.
	INT0	IN	1	ST	External interrupt 0 input.
	FLT0	IN	1	ST	Enhanced PWM Fault input.
	AN10	IN	1	ANA	A/D input channel 10. Enabled on POR, this analog input overrides the digital input (read as clear – low level).
RB1/INT1/AN8	RB1	OUT	0	DIG	LATB<1> data output.
		IN	1	TTL	PORTB<1> data input. Weak pull-up available only in this mode.
	INT1	IN	1	ST	External interrupt 1 input.
	AN8	IN	1	ANA	A/D input channel 8. Enabled on POR, this analog input overrides the digital input (read as clear – low level).
RB2/INT2/CANTX	RB2	OUT	x	DIG	LATB<2> data output.
		IN	1	TTL	PORTB<2> data input. Weak pull-up available only in this mode.
	INT2	IN	1	ST	External interrupt 2 input.
	CANTX	OUT	1	DIG	CAN transmit signal output. The CAN interface overrides the TRIS<2> control when enabled.
RB3/CANRX	RB3	OUT	0	DIG	LATB<3> data output.
		IN	1	TTL	PORTB<3> data input. Weak pull-up available only in this mode.
	CANRX	IN	1	ST	CAN receive signal input. Pin must be configured as a digital input by setting TRISB<3>.
RB4/KBI0/AN9	RB4	OUT	0	DIG	LATB<4> data output.
		IN	1	TTL	PORTB<4> data input. Weak pull-up available only in this mode.
	KBI0	IN	1	TTL	Interrupt-on-pin change.
	AN9	IN	1	ANA	A/D input channel 9. Enabled on POR, this analog input overrides the digital input (read as clear – low level).
RB5/KBI1/PGM	RB5	OUT	0	DIG	LATB<5> data output.
		IN	1	TTL	PORTB<5> data input. Weak pull-up available only in this mode.
	KBI1	IN	1	TTL	Interrupt-on-pin change.
	PGM	IN	x	ST	Low-Voltage Programming mode entry (ICSP™). Enabling this function overrides digital output.
RB6/KBI2/PGC	RB6	OUT	0	DIG	LATB<6> data output.
		IN	1	TTL	PORTB<6> data input. Weak pull-up available only in this mode.
	KBI2	IN	1	TTL	Interrupt-on-pin change.
	PGC	IN	x	ST	Low-Voltage Programming mode entry (ICSP) clock input.
RB7/KBI3/PGD	RB7	OUT	0	DIG	LATB<7> data output.
		IN	1	TTL	PORTB<7> data input. Weak pull-up available only in this mode.
	KBI3	IN	1	TTL	Interrupt-on-pin change.
	PGD	OUT	x	DIG	Low-Voltage Programming mode entry (ICSP) clock output.
		IN	x	ST	Low-Voltage Programming mode entry (ICSP) clock input.

Legend: PWR = Power Supply; OUT = Output; IN = Input; ANA = Analog Signal; DIG = Digital Output; ST = Schmitt Buffer Input; TTL – TTL Buffer Input

TABLE 10-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	52
LATB	LATB Data Output Register (Read and Write to Data Latch)								52
TRISB	PORTB Data Direction Control Register								52
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
INTCON2	$\overline{RBPU}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	49
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	49
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	50

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PORTB.

11.3   Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module. The prescaler is not directly readable or writable; its value is set by the PSA and T0PS2:T0PS0 bits (T0CON<3:0>) which determine the prescaler assignment and prescale ratio.

Clearing the PSA bit assigns the prescaler to the Timer0 module. When it is assigned, prescale values from 1:2 through 1:256 in power-of-2 increments are selectable.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF TMR0`, `MOVWF TMR0`, `BSF TMR0`, etc.) clear the prescaler count.

> **Note:** Writing to TMR0 when the prescaler is assigned to Timer0 will clear the prescaler count but will not change the prescaler assignment.

11.3.1   SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control and can be changed "on-the-fly" during program execution.

11.4   Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h in 8-bit mode, or from FFFFh to 0000h in 16-bit mode. This overflow sets the TMR0IF flag bit. The interrupt can be masked by clearing the TMR0IE bit (INTCON<5>). Before re-enabling the interrupt, the TMR0IF bit must be cleared in software by the Interrupt Service Routine.

Since Timer0 is shut down in Sleep mode, the TMR0 interrupt cannot awaken the processor from Sleep.

TABLE 11-1:   REGISTERS ASSOCIATED WITH TIMER0

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
TMR0L	Timer0 Module Low Byte Register								50
TMR0H	Timer0 Module High Byte Register								50
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	50
TRISA	—	PORTA Data Direction Register							52

Legend: x = unknown, u = unchanged, — = unimplemented locations, read as '0'.
Shaded cells are not used by Timer0.

17.3.8   OPERATION IN POWER MANAGED MODES

In SPI Master mode, module clocks may be operating at a different speed than when in full power mode; in the case of the Sleep mode, all clocks are halted.

In most power managed modes, a clock is provided to the peripherals. That clock should be from the primary clock source, the secondary clock (Timer1 oscillator at 32.768 kHz) or the INTOSC source. See **Section 2.7 "Clock Sources and Oscillator Switching"** for additional information.

In most cases, the speed that the master clocks SPI data is not important; however, this should be evaluated for each system.

If MSSP interrupts are enabled, they can wake the controller from Sleep mode, or one of the Idle modes, when the master completes sending data. If an exit from Sleep or Idle mode is not desired, MSSP interrupts should be disabled.

If the Sleep mode is selected, all module clocks are halted and the transmission/reception will remain in that state until the devices wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in any power managed mode and data to be shifted into the SPI Transmit/ Receive Shift register. When all 8 bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device.

17.3.9   EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

17.3.10   BUS MODE COMPATIBILITY

Table 17-1 shows the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

TABLE 17-1:   SPI BUS MODES

Standard SPI Mode Terminology	Control Bits State	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

There is also a SMP bit which controls when the data is sampled.

TABLE 17-2:   REGISTERS ASSOCIATED WITH SPI OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	PSPIF[1]	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	52
PIE1	PSPIE[1]	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	52
IPR1	PSPIP[1]	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	52
TRISA	PORTA Data Direction Register								52
TRISC	PORTC Data Direction Register								52
SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								50
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	50
SSPSTAT	SMP	CKE	D/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	UA	BF	50

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the MSSP in SPI mode.

Note 1:   These bits are unimplemented in PIC18F2X8X devices; always maintain these bits clear.

17.4.17.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

a) SDA or SCL are sampled low at the beginning of the Start condition (Figure 17-26).
b) SCL is sampled low before SDA is asserted low (Figure 17-27).

During a Start condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

• the Start condition is aborted,
• the BCLIF flag is set; and
• the MSSP module is reset to its Idle state (Figure 17-26).

The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded from SSPADD<6:0> and counts down to 0. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 17-28). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to 0 and during this time, if the SCL pins are sampled as '0', a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

> **Note:** The reason that bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.

FIGURE 17-26: BUS COLLISION DURING START CONDITION (SDA ONLY)
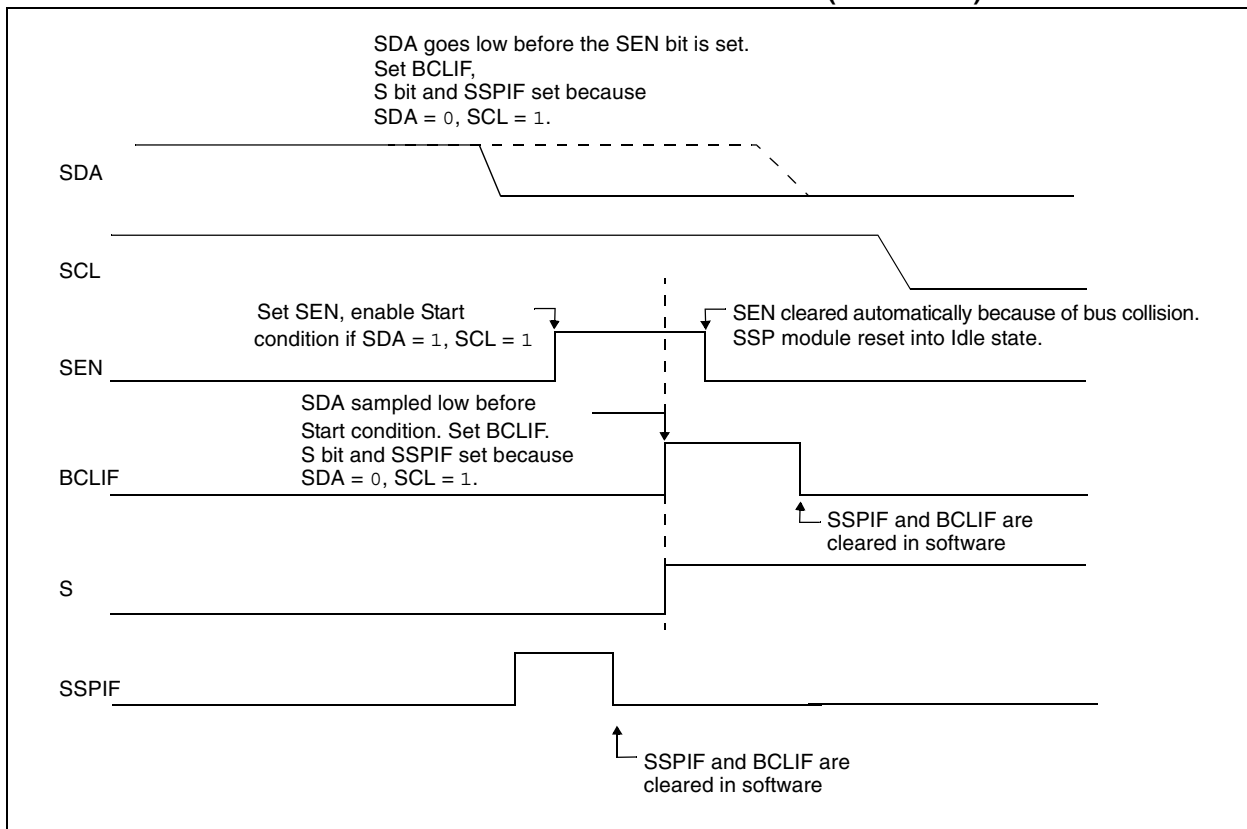
REGISTER 18-1: **TXSTA: TRANSMIT STATUS AND CONTROL REGISTER**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D

bit 7   bit 0

bit 7   **CSRC:** Clock Source Select bit

<u>Asynchronous mode:</u>
Don't care.
<u>Synchronous mode:</u>
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6   **TX9:** 9-bit Transmit Enable bit

1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5   **TXEN:** Transmit Enable bit

1 = Transmit enabled
0 = Transmit disabled

　　　　Note:　SREN/CREN overrides TXEN in Sync mode.

bit 4   **SYNC:** EUSART Mode Select bit

1 = Synchronous mode
0 = Asynchronous mode

bit 3   **SENDB:** Send Break Character bit

<u>Asynchronous mode:</u>
1 = Send Sync Break on next transmission (cleared by hardware upon completion)
0 = Sync Break transmission completed
<u>Synchronous mode:</u>
Don't care.

bit 2   **BRGH:** High Baud Rate Select bit

<u>Asynchronous mode:</u>
1 = High speed
0 = Low speed
<u>Synchronous mode:</u>
Unused in this mode.

bit 1   **TRMT:** Transmit Shift Register Status bit

1 = TSR empty
0 = TSR full

bit 0   **TX9D:** 9th bit of Transmit Data

Can be address/data bit or a parity bit.

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

FIGURE 18-7: **ASYNCHRONOUS RECEPTION**

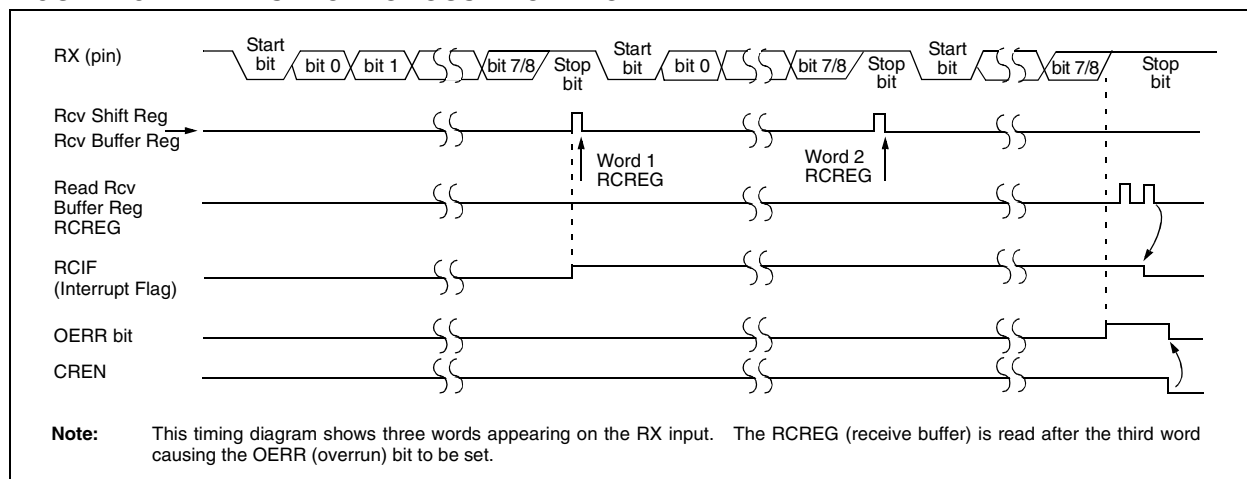**Note:** This timing diagram shows three words appearing on the RX input. The RCREG (receive buffer) is read after the third word causing the OERR (overrun) bit to be set.

TABLE 18-6: **REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	PSPIF[1]	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	52
PIE1	PSPIE[1]	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	52
IPR1	PSPIP[1]	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	52
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	51
RCREG	EUSART Receive Register								51
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	51
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	51
SPBRGH	EUSART Baud Rate Generator Register High Byte								51
SPBRG	EUSART Baud Rate Generator Register Low Byte								51

Legend: — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

Note 1: Reserved in PIC18F2X8X devices; always maintain these bits clear.

20.9 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 20-4. Since the analog pins are connected to a digital output, they have reverse biased diodes to V_{DD} and V_{SS}. The analog input, therefore, must be between V_{SS} and V_{DD}. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up condition may occur. A maximum source impedance of 10 kΩ is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.

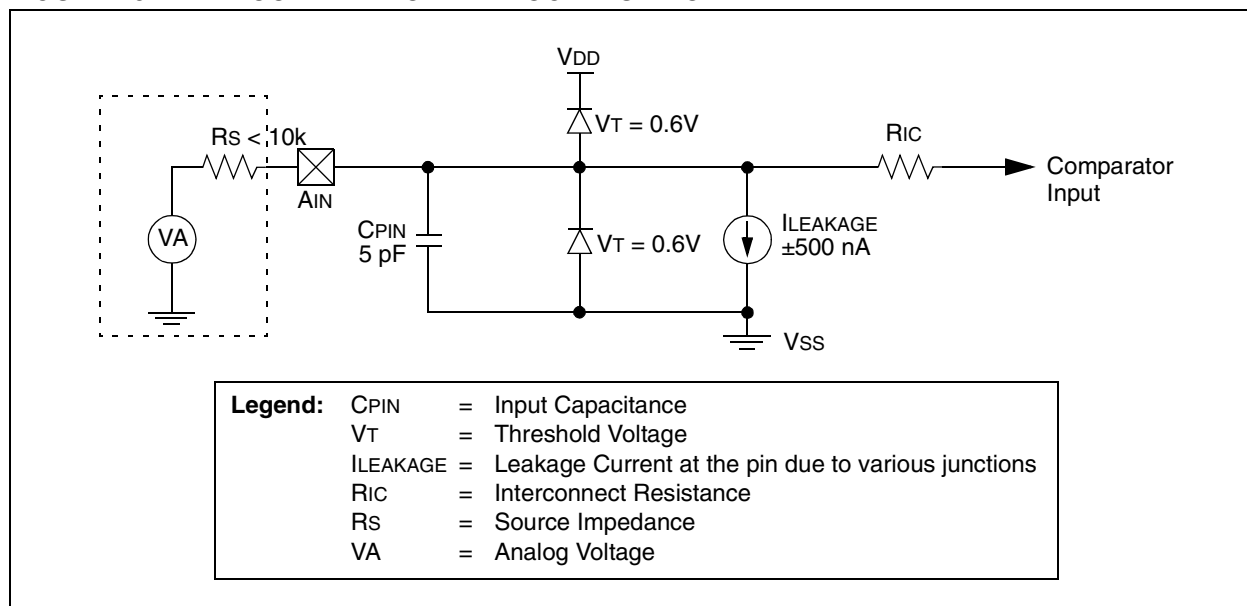**FIGURE 20-4:   COMPARATOR ANALOG INPUT MODEL**

Legend:
CPIN = Input Capacitance
VT = Threshold Voltage
ILEAKAGE = Leakage Current at the pin due to various junctions
RIC = Interconnect Resistance
RS = Source Impedance
VA = Analog Voltage

TABLE 20-1:   REGISTERS ASSOCIATED WITH COMPARATOR MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
CMCON[3]	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	51
CVRCON[3]	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	51
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	52
IPR2	OSCFIP	CMIP[2]	—	EEIP	BCLIP	HLVDIP	TMR3IP	ECCP1IP[2]	51
PIR2	OSCFIF	CMIF[2]	—	EEIF	BCLIF	HLVDIF	TMR3IF	ECCP1IF[2]	51
PIE2	OSCFIE	CMIE[2]	—	EEIE	BCLIE	HLVDIE	TMR3IE	ECCP1IE[2]	52
PORTA	RA7[1]	RA6[1]	RA5	RA4	RA3	RA2	RA1	RA0	52
LATA	LATA7[1]	LATA6[1]	LATA Data Output Register						52
TRISA	TRISA7[1]	TRISA6[1]	PORTA Data Direction Register						52

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the comparator module.

Note 1:   PORTA pins are enabled based on oscillator configuration.

2:    These bits are available in PIC18F4X8X devices and reserved in PIC18F2X8X devices.

3:    These registers are unimplemented on PIC18F2X8X devices.

REGISTER 23-1: **CANCON: CAN CONTROL REGISTER**

	R/W-1	R/W-0	R/W-0	R/S-0	R/W-0	R/W-0	R/W-0	U-0
Mode 0	REQOP2	REQOP1	REQOP0	ABAT	WIN2	WIN1	WIN0	—

	R/W-1	R/W-0	R/W-0	R/S-0	U-0	U-0	U-0	U-0
Mode 1	REQOP2	REQOP1	REQOP0	ABAT	—	—	—	—

	R/W-1	R/W-0	R/W-0	R/S-0	R-0	R-0	R-0	R-0
Mode 2	REQOP2	REQOP1	REQOP0	ABAT	FP3	FP2	FP1	FP0
	bit 7							bit 0

bit 7-5   **REQOP2:REQOP0:** Request CAN Operation Mode bits

$1xx$ = Request Configuration mode
011 = Request Listen Only mode
010 = Request Loopback mode
001 = Request Disable mode
000 = Request Normal mode

bit 4   **ABAT:** Abort All Pending Transmissions bit

1 = Abort all pending transmissions (in all transmit buffers)
0 = Transmissions proceeding as normal

bit 3-1   <u>Mode 0:</u>

WIN2:WIN0: Window Address bits

These bits select which of the CAN buffers to switch into the access bank area. This allows access to the buffer registers from any data memory bank. After a frame has caused an interrupt, the ICODE3:ICODE0 bits can be copied to the WIN3:WIN0 bits to select the correct buffer. See Example 23-2 for a code example.

111 = Receive Buffer 0
110 = Receive Buffer 0
101 = Receive Buffer 1
100 = Transmit Buffer 0
011 = Transmit Buffer 1
010 = Transmit Buffer 2
001 = Receive Buffer 0
000 = Receive Buffer 0

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

bit 4-0   <u>Mode 1:</u>

Unimplemented: Read as '0'

<u>Mode 2:</u>

FP3:FP0: FIFO Read Pointer bits

These bits point to the message buffer to be read.

$0111:0000$ = Message buffer to be read
$1111:1000$ = Reserved

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

REGISTER 23-51: MSEL3: MASK SELECT REGISTER 3[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FIL15_1	FIL15_0	FIL14_1	FIL14_0	FIL13_1	FIL13_0	FIL12_1	FIL12_0
bit 7							bit 0

bit 7-6 **FIL15_1:FIL15_0:** Filter 15 Select bits 1 and 0

11 = No mask
10 = Filter 15
01 = Acceptance Mask 1
00 = Acceptance Mask 0

bit 5-4 **FIL14_1:FIL14_0:** Filter 14 Select bits 1 and 0

11 = No mask
10 = Filter 15
01 = Acceptance Mask 1
00 = Acceptance Mask 0

bit 3-2 **FIL13_1:FIL13_0:** Filter 13 Select bits 1 and 0

11 = No mask
10 = Filter 15
01 = Acceptance Mask 1
00 = Acceptance Mask 0

bit 1-0 **FIL12_1:FIL12_0:** Filter 12 Select bits 1 and 0

11 = No mask
10 = Filter 15
01 = Acceptance Mask 1
00 = Acceptance Mask 0

Note 1: This register is available in Mode 1 and 2 only.

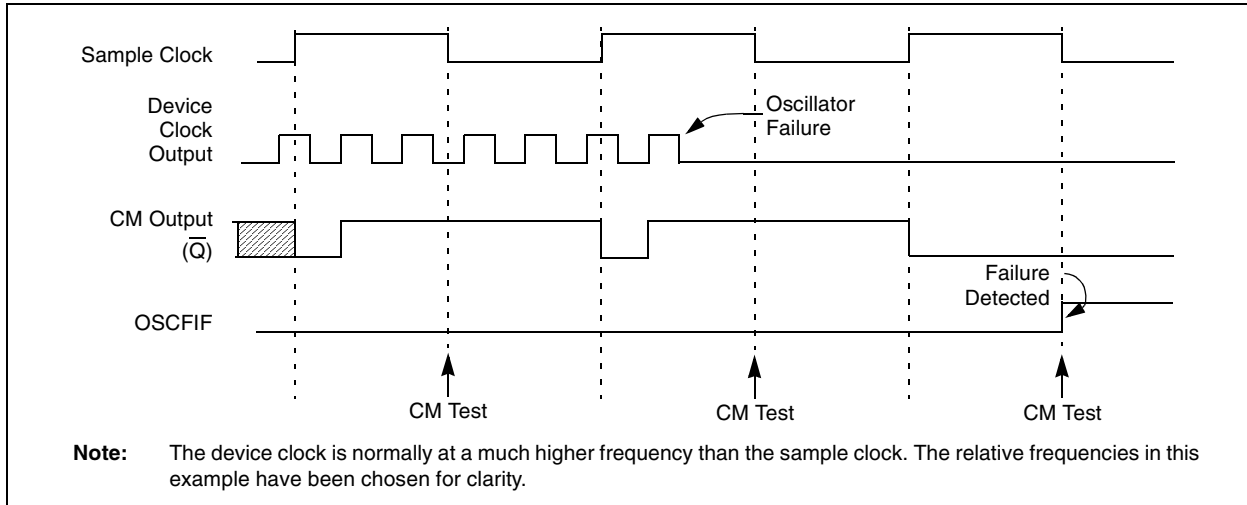| Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

TABLE 23-1: CAN CONTROLLER REGISTER MAP (CONTINUED)

Address[1]	Name
D7Fh	—[4]
D7Eh	—[4]
D7Dh	—[4]
D7Ch	—[4]
D7Bh	RXF11EIDL
D7Ah	RXF11EIDH
D79h	RXF11SIDL
D78h	RXF11SIDH
D77h	RXF10EIDL
D76h	RXF10EIDH
D75h	RXF10SIDL
D74h	RXF10SIDH
D73h	RXF9EIDL
D72h	RXF9EIDH
D71h	RXF9SIDL
D70h	RXF9SIDH
D6Fh	—[4]
D6Eh	—[4]
D6Dh	—[4]
D6Ch	—[4]
D6Bh	RXF8EIDL
D6Ah	RXF8EIDH
D69h	RXF8SIDL
D68h	RXF8SIDH
D67h	RXF7EIDL
D66h	RXF7EIDH
D65h	RXF7SIDL
D64h	RXF7SIDH
D63h	RXF6EIDL
D62h	RXF6EIDH
D61h	RXF6SIDL
D60h	RXF6SIDH

Note 1: Shaded registers are available in Access Bank low area while the rest are available in Bank 15.

2: CANSTAT register is repeated in these locations to simplify application firmware. Unique names are given for each instance of the controller register due to the Microchip header file requirement.

3: These registers are not CAN registers.

4: Unimplemented registers are read as '0'.

FIGURE 24-4: **FSCM TIMING DIAGRAM**

Note: The device clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

24.4.3 FSCM INTERRUPTS IN POWER MANAGED MODES

By entering a power managed mode, the clock multiplexer selects the clock source selected by the OSCCON register. Fail-Safe Monitoring of the power managed clock source resumes in the power managed mode.

If an oscillator failure occurs during power managed operation, the subsequent events depend on whether or not the oscillator failure interrupt is enabled. If enabled (OSCFIF = 1), code execution will be clocked by the INTOSC multiplexer. An automatic transition back to the failed clock source will not occur.

If the interrupt is disabled, subsequent interrupts while in Idle mode will cause the CPU to begin executing instructions while being clocked by the INTOSC source.

24.4.4 POR OR WAKE-UP FROM SLEEP

The FSCM is designed to detect oscillator failure at any point after the device has exited Power-on Reset (POR) or low-power Sleep mode. When the primary device clock is EC, RC or INTRC modes, monitoring can begin immediately following these events.

For oscillator modes involving a crystal or resonator (HS, HSPLL, LP or XT), the situation is somewhat different. Since the oscillator may require a start-up time considerably longer than the FCSM sample clock time, a false clock failure may be detected. To prevent this, the internal oscillator block is automatically configured as the device clock and functions until the primary clock is stable (the OST and PLL timers have timed out). This is identical to Two-Speed Start-up mode. Once the primary clock is stable, the INTRC returns to its role as the FSCM source.

Note: The same logic that prevents false oscillator failure interrupts on POR, or wake from Sleep, will also prevent the detection of the oscillator's failure to start at all following these events. This can be avoided by monitoring the OSTS bit and using a timing routine to determine if the oscillator is taking too long to start. Even so, no oscillator failure interrupt will be flagged.

As noted in **Section 24.3.1 "Special Considerations for Using Two-Speed Start-up"**, it is also possible to select another clock configuration and enter an alternate power managed mode while waiting for the primary clock to become stable. When the new power managed mode is selected, the primary clock is disabled.

BTG — Bit Toggle f

Syntax:	BTG f, b {,a}
Operands:	$0 \leq f \leq 255$ $0 \leq b < 7$ $a \in [0,1]$
Operation:	$(\overline{f<b>}) \rightarrow f<b>$
Status Affected:	None

Encoding:

0111	bbba	ffff	ffff

Description: Bit 'b' in data memory location 'f' is inverted.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset addressing mode whenever f ≤ 95 (5Fh). See **Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f'

Example: BTG PORTC, 4, 0

Before Instruction:
PORTC = 0111 0101 [75h]
After Instruction:
PORTC = 0110 0101 [65h]

BOV — Branch if Overflow

Syntax:	BOV n
Operands:	$-128 \leq n \leq 127$
Operation:	if Overflow bit is '1' $(PC) + 2 + 2n \rightarrow PC$
Status Affected:	None

Encoding:

1110	0100	nnnn	nnnn

Description: If the Overflow bit is '1', then the program will branch.

The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: HERE BOV Jump

Before Instruction
PC = address (HERE)
After Instruction
If Overflow = 1;
PC = address (Jump)
If Overflow = 0;
PC = address (HERE + 2)

CLRF	**Clear f**

Syntax:	CLRF f {,a}
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]
Operation:	000h → f 1 → Z
Status Affected:	Z

Encoding:

0110	101a	ffff	ffff

Description:	Clears the contents of the specified register. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f'

Example: CLRF   FLAG_REG,1

Before Instruction
    FLAG_REG   =   5Ah
After Instruction
    FLAG_REG   =   00h

CLRWDT	**Clear Watchdog Timer**

Syntax:	CLRWDT
Operands:	None
Operation:	000h → WDT, 000h → WDT postscaler, 1 → $\overline{TO}$, 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$

Encoding:

0000	0000	0000	0100

Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the postscaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	No operation

Example: CLRWDT

Before Instruction
    WDT Counter   =   ?
After Instruction
    WDT Counter   =   00h
    WDT Postscaler   =   0
    $\overline{TO}$ =   1
    $\overline{PD}$ =   1

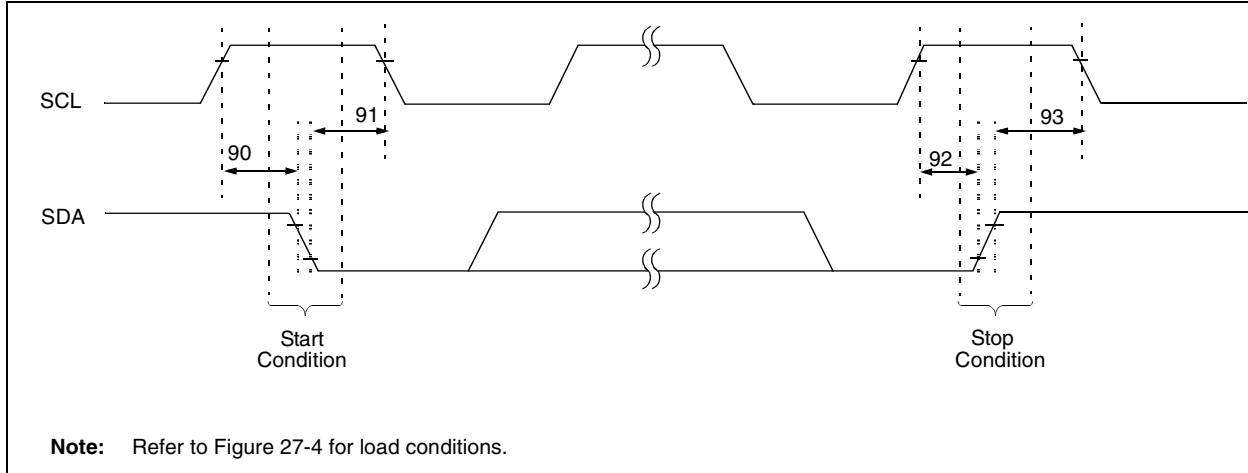# PIC18F2585/2680/4585/4680

FIGURE 27-16: I²C™ BUS START/STOP BITS TIMING

TABLE 27-18: I²C™ BUS START/STOP BITS REQUIREMENTS (SLAVE MODE)

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
90	TSU:STA	Start Condition Setup Time	100 kHz mode	4700	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	600	—		
91	THD:STA	Start Condition Hold Time	100 kHz mode	4000	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	600	—		
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	4700	—	ns	
			400 kHz mode	600	—		
93	THD:STO	Stop Condition Hold Time	100 kHz mode	4000	—	ns	
			400 kHz mode	600	—		

FIGURE 27-17: I²C™ BUS DATA TIMING

Preliminary   © 2007 Microchip Technology Inc.
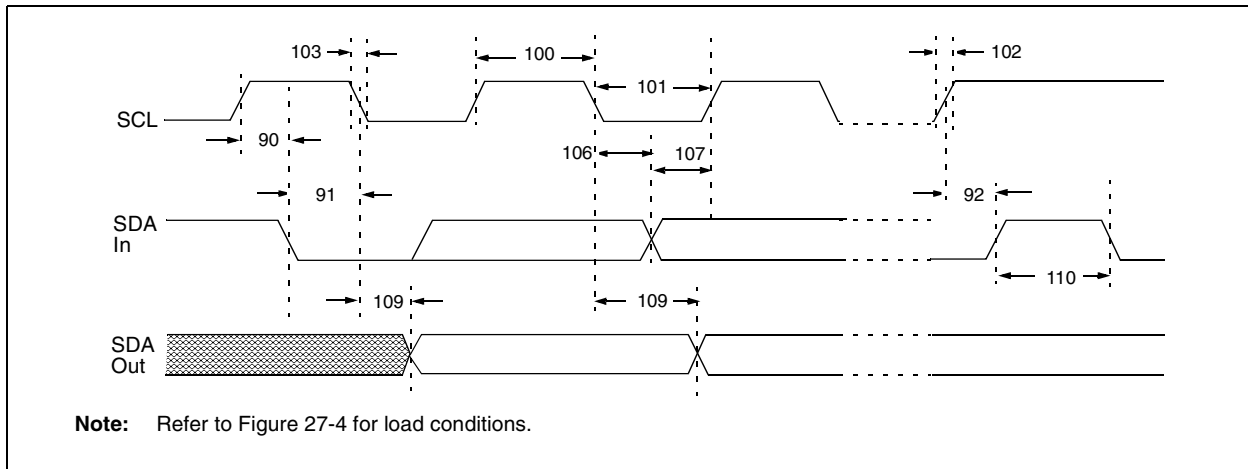
APPENDIX E: MIGRATION FROM MID-RANGE TO ENHANCED DEVICES

A detailed discussion of the differences between the mid-range MCU devices (i.e., PIC16CXXX) and the enhanced devices (i.e., PIC18FXXX) is provided in *AN716, "Migrating Designs from PIC16C74A/74B to PIC18C442."* The changes discussed, while device specific, are generally applicable to all mid-range to enhanced device migrations.

This Application Note is available as Literature Number DS00716.

APPENDIX F: MIGRATION FROM HIGH-END TO ENHANCED DEVICES

A detailed discussion of the migration pathway and differences between the high-end MCU devices (i.e., PIC17CXXX) and the enhanced devices (i.e., PIC18FXXX) is provided in *AN726, "PIC17CXXX to PIC18CXXX Migration."* This Application Note is available as Literature Number DS00726.