Welcome to E-XFL.COM

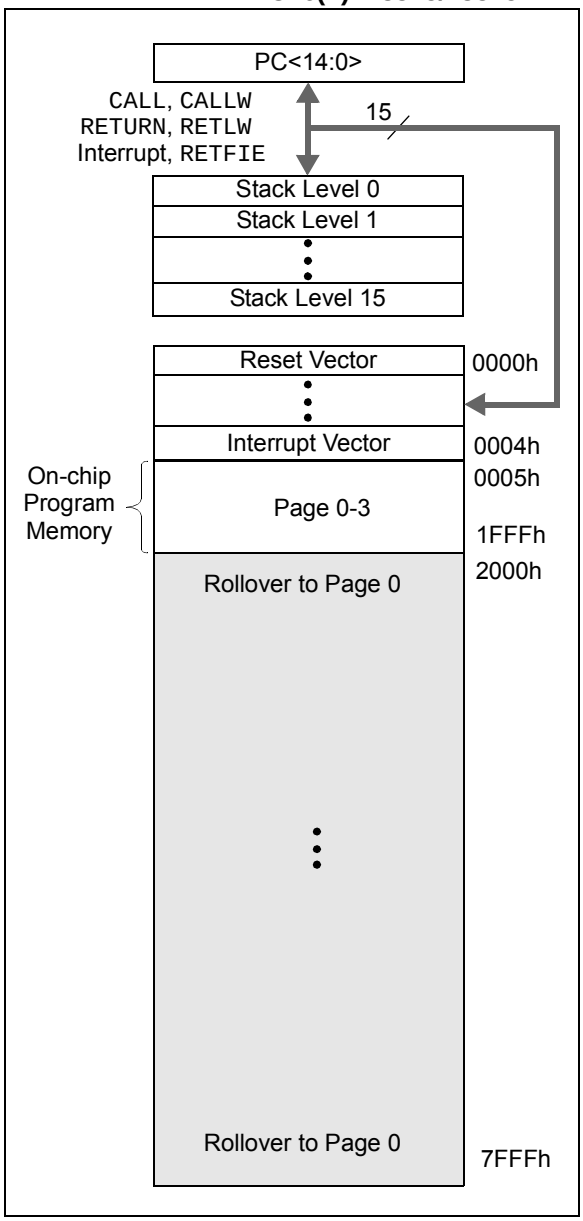
What is "Embedded - Microcontrollers"?

"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	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	18
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 17x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f18345-i-ss

FIGURE 4-1: PROGRAM MEMORY MAP AND STACK FOR PIC16(L)F18325/18345



4.1.1 READING PROGRAM MEMORY AS DATA

There are three methods of accessing constants in program memory. The first method is to use tables of RETLW instructions. The second method is to set an FSR to point to the program memory. The third method is to use the NVMCON registers to access the program memory.

4.1.1.1 RETLW Instruction

The RETLW instruction can be used to provide access to tables of constants. The recommended way to create such a table is shown in Example 4-1.

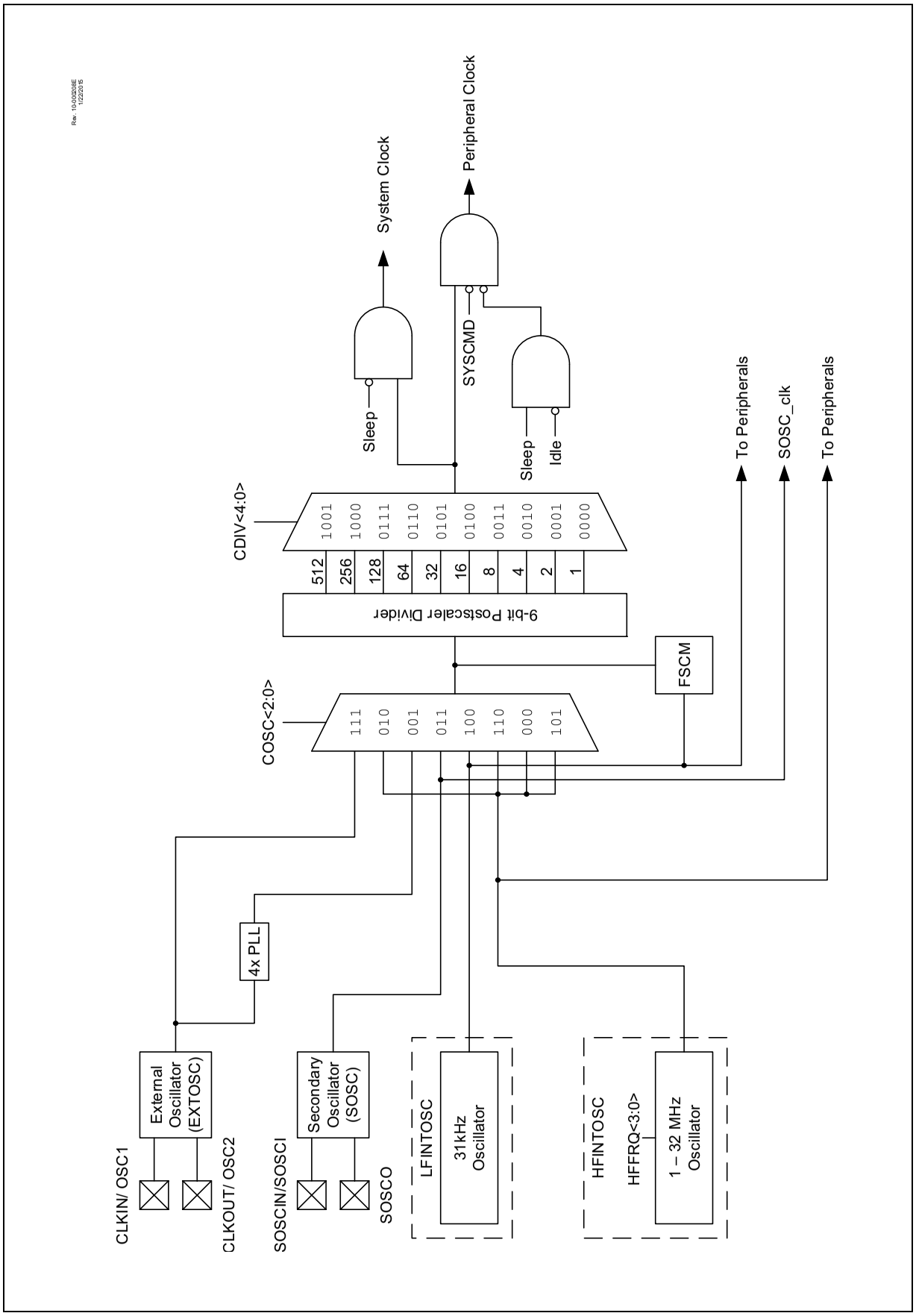
EXAMPLE 4-1: RETLW INSTRUCTION

```
constants
    BRW                 ;Add Index in W to
                        ;program counter to
                        ;select data
    RETLW DATA0         ;Index0 data
    RETLW DATA1         ;Index1 data
    RETLW DATA2
    RETLW DATA3


my_function
    ;… LOTS OF CODE…
    MOVLW       DATA_INDEX
    call constants
    ;… THE CONSTANT IS IN W
```

The BRW instruction makes this type of table very simple to implement. If your code must remain portable with previous generations of microcontrollers, computed GOTO method must be used because the BRW instruction is not available in some devices, such as the PIC16F6XX, PIC16F7XX, PIC16F8XX, and PIC16F9XX devices.

FIGURE 7-1: SIMPLIFIED PIC® MCU CLOCK SOURCE BLOCK DIAGRAM



8.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- GIE bit of the INTCON register
- Interrupt Enable bit(s) (PIEx bits) for the specific interrupt event(s)
- PEIE bit of the INTCON register

The PIR1, PIR2, PIR3 and PIR4 registers record individual interrupts via interrupt flag bits. Interrupt flag bits will be set, regardless of the status of the GIE, PEIE and individual interrupt enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- Critical registers are automatically saved to the shadow registers (See **"Section 8.5 "Automatic Context Saving"**)
- PC is loaded with the interrupt vector 0004h

The firmware within the Interrupt Service Routine (ISR) should determine the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

The `RETFIE` instruction exits the ISR by popping the previous address from the stack, restoring the saved context from the shadow registers and setting the GIE bit.

For additional information on a specific interrupt's operation, refer to its peripheral chapter.

Note 1:	Individual interrupt flag bits are set, regardless of the state of any other enable bits.
2:	All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again.

8.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The interrupt is sampled during Q1 of the instruction cycle. The actual interrupt latency then depends on the instruction that is executing at the time the interrupt is detected. See Figure 8-2 and Figure 8-3 for more details.

12.2.6 ANALOG CONTROL

The ANSELA register (Register 12-4) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELA bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELA bits has no effect on digital output functions. A pin with TRIS clear and ANSEL set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

Note:	The ANSELA bits default to the Analog mode after Reset. To use any pins as digital general purpose or peripheral inputs, the corresponding ANSEL bits must be initialized to '0' by user software.

12.2.7 WEAK PULL-UP CONTROL

The WPUA register (Register 12-5) controls the individual weak pull-ups for each PORT pin.

PORTA pin RA3 includes the $\overline{MCLR}$/VPP input. The $\overline{MCLR}$ input allows the device to be reset, and can be disabled by the MCLRE bit of Configuration Word 2. A weak pull-up is present on the RA3 port pin. This weak pull-up is enabled when $\overline{MCLR}$ is enabled (MCLRE = 1) or the WPUA3 bit is set. The weak pull-up is disabled when is disabled and the WPUA3 bit is clear.

12.2.8 PORTA FUNCTIONS AND OUTPUT PRIORITIES

Each PORTA pin is multiplexed with other functions.

Each pin defaults to the PORT latch data after Reset. Other output functions are selected with the peripheral pin select logic. See **Section 13.0 "Peripheral Pin Select (PPS) Module"** for more information.

Analog input functions, such as ADC and comparator inputs are not shown in the peripheral pin select lists. Digital output functions may continue to control the pin when it is in Analog mode.

REGISTER 12-3: LATA: PORTA DATA LATCH REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0
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 5-4 **LATA<5:4>**: RA<5:4> Output Latch Value bits(1)

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **LATA<2:0>**: RA<2:0> Output Latch Value bits(1)

Note 1: Writes to PORTA are actually written to corresponding LATA register. Reads from PORTA register is return of actual I/O pin values.

REGISTER 12-4: ANSELA: PORTA ANALOG SELECT REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0
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 5-4 **ANSA<5:4>**: Analog Select between Analog or Digital Function on pins RA<5:4>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **ANSA<2:0>**: Analog Select between Analog or Digital Function on pins RA<2:0>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

REGISTER 12-7: SLRCONA: PORTA SLEW RATE CONTROL REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0
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 5-4 **SLRA<5:4>:** PORTA Slew Rate Enable bits
For RA<5:4> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **SLRA<2:0>:** PORTA Slew Rate Enable bits
For RA<2:0> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

REGISTER 12-8: INLVLA: PORTA INPUT LEVEL CONTROL REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0
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 5-0 **INLVLA<5:0>:** PORTA Input Level Select bits
For RA<5:0> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

REGISTER 13-3: PPSLOCK: PPS LOCK REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	PPSLOCKED
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 7-1 **Unimplemented:** Read as ‘0’

bit 0 **PPSLOCKED:** PPS Locked bit
1= PPS is locked. PPS selections can not be changed.
0= PPS is not locked. PPS selections can be changed.

TABLE 13-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE PPS MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	164
INTPPS	—	—	—	INTPPS<4:0>					162
T0CKIPPS	—	—	—	T0CKIPPS<4:0>					162
T1CKIPPS	—	—	—	T1CKIPPS<4:0>					162
T1GPPS	—	—	—	T1GPPS<4:0>					162
T3CKIPPS	—	—	—	T3CKIPPS<4:0>					162
T3GPPS	—	—	—	T3GPPS<4:0>					162
T5CKIPPS	—	—	—	T5CKIPPS<4:0>					162
T5GPPS	—	—	—	T5GPPS<4:0>					162
CCP1PPS	—	—	—	CCP1PPS<4:0>					162
CCP2PPS	—	—	—	CCP2PPS<4:0>					162
CCP3PPS	—	—	—	CCP3PPS<4:0>					162
CCP4PPS	—	—	—	CCP4PPS<4:0>					162
CWG1PPS	—	—	—	CWG1PPS<4:0>					162
CWG2PPS	—	—	—	CWG2PPS<4:0>					162
MDCIN1PPS	—	—	—	MDCIN1PPS<4:0>					162
MDCIN2PPS	—	—	—	MDCIN2PPS<4:0>					162
MDMINPPS	—	—	—	MDMINPPS<4:0>					162
SSP1CLKPPS	—	—	—	SSP1CLKPPS<4:0>					162
SSP1DATPPS	—	—	—	SSP1DATPPS<4:0>					162
SSP1SSPPS	—	—	—	SSP1SSPPS<4:0>					162
SSP2CLKPPS	—	—	—	SSP2CLKPPS<4:0>					162
SSP2DATPPS	—	—	—	SSP2DATPPS<4:0>					162
SSP2SSPPS	—	—	—	SSP2SSPPS<4:0>					162

Legend: — = unimplemented, read as ‘0’. Shaded cells are unused by the PPS module.
Note 1: PIC16(L)F18345 only.

15.0 INTERRUPT-ON-CHANGE

All pins on all ports can be configured to operate as Interrupt-On-Change (IOC) pins. An interrupt can be generated by detecting a signal that has either a rising edge or a falling edge. Any individual pin, or combination of pins, can be configured to generate an interrupt. The interrupt-on-change module has the following features:

- Interrupt-on-Change enable
  - Rising and falling edge detection
- Individual pin configuration
- Individual pin interrupt flags

Figure 15-1 is a block diagram of the IOC module.

15.1 Enabling the Module

To allow individual pins to generate an interrupt, the IOCIE bit of the PIE0 register must be set. If the IOCIE bit is disabled, the edge detection on the pin will still occur, but an interrupt will not be generated.

15.2 Individual Pin Configuration

For each pin, a rising edge detector and a falling edge detector are present. To enable a pin to detect a rising edge, the associated bit of the IOCxP register is set. To enable a pin to detect a falling edge, the associated bit of the IOCxN register is set.

A pin can be configured to detect rising and falling edges simultaneously by setting the associated bits in both of the IOCxP and IOCxN registers.

15.3 Interrupt Flags

The bits located in the IOCxF registers are status flags that correspond to the interrupt-on-change pins of each port. If an expected edge is detected on an appropriately enabled pin, then the status flag for that pin will be set, and an interrupt will be generated if the IOCIE bit is set. The IOCIF bit of the PIR0 register reflects the status of all IOCxF bits.

15.3.1 CLEARING INTERRUPT FLAGS

The individual status flags, (IOCxF register bits), can be cleared by resetting them to zero. If another edge is detected during this clearing operation, the associated status flag will be set at the end of the sequence, regardless of the value actually being written.

In order to ensure that no detected edge is lost while clearing flags, only AND operations masking out known changed bits should be performed. The following sequence is an example of what should be performed.

EXAMPLE 15-1: CLEARING INTERRUPT FLAGS (PORTA EXAMPLE)

```
MOVLW  0xff
XORWF  IOCAF, W
ANDWF  IOCAF, F
```

15.4 Operation in Sleep

The interrupt-on-change interrupt event will wake the device from Sleep mode, if the IOCIE bit is set.

If an edge is detected while in Sleep mode, the affected IOCxF register will be updated prior to the first instruction executed out of Sleep.

18.12 Register Definitions: Comparator Control

REGISTER 18-1: CMxCON0: COMPARATOR Cx CONTROL REGISTER 0

R/W-0/0	R-0/0	U-0	R/W-0/0	U-0	R/W-1/1	R/W-0/0	R/W-0/0
CxON	CxOUT	—	CxPOL	—	CxSP	CxHYS	CxSYNC
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 **CxON:** Comparator Enable bit
1 = Comparator is enabled
0 = Comparator is disabled and consumes no active power

bit 6 **CxOUT:** Comparator Output bit
If CxPOL = 1 (inverted polarity):
1 = CxVP < CxVN
0 = CxVP > CxVN
If CxPOL = 0 (non-inverted polarity):
1 = CxVP > CxVN
0 = CxVP < CxVN

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

bit 4 **CxPOL:** Comparator Output Polarity Select bit
1 = Comparator output is inverted
0 = Comparator output is not inverted

bit 3 **Unimplemented:** Read as '0'

bit 2 **CxSP:** Comparator Speed/Power Select bit
1 = Comparator operates in Normal-Power, High-Speed mode
0 = Reserved. (do not use)

bit 1 **CxHYS:** Comparator Hysteresis Enable bit
1 = Comparator hysteresis enabled
0 = Comparator hysteresis disabled

bit 0 **CxSYNC:** Comparator Output Synchronous Mode bit
1 = Comparator output to Timer1 and I/O pin is synchronous to changes on Timer1 clock source. Output updated on the falling edge of Timer1 clock source.
0 = Comparator output to Timer1 and I/O pin is asynchronous

REGISTER 20-2: CWGxCON1: CWGx CONTROL REGISTER 1

U-0	U-0	R-x	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	IN	—	POLD	POLC	POLB	POLA
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 7-6 **Unimplemented:** Read as ‘0’

bit 5 **IN:** CWGx Data Input Signal (read-only)

bit 4 **Unimplemented:** Read as ‘0’

bit 3 **POLD:** WGxD Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

bit 2 **POLC:** WGxC Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

bit 1 **POLB:** WGxB Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

bit 0 **POLA:** WGxA Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

REGISTER 20-3: CWGxCLKCON: CWGx CLOCK INPUT SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	CS
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 7-1 **Unimplemented:** Read as ‘0’

bit 0 **CS:** CWG Clock Source Selection Select bits

CS	Clock Source
0	FOSC
1	HFINTOSC (remains operating during Sleep)

26.0 TIMER0 MODULE

The Timer0 module is an 8/16-bit timer/counter with the following features:

- 16-bit timer/counter
- 8-bit timer/counter with programmable period
- Synchronous or asynchronous operation
- Selectable clock sources
- Programmable prescaler (independent of Watchdog Timer)
- Programmable postscaler
- Operation during Sleep mode
- Interrupt on match or overflow
- Output on I/O pin (via PPS) or to other peripherals

26.1 Timer0 Operation

Timer0 can operate as either an 8-bit timer/counter or a 16-bit timer/counter. The mode is selected with the T016BIT bit of the T0CON register.

When used with the FOSC/4 clock source, the module is a timer and increments on every instruction cycle. When used with any other clock source, the module can be used as either a timer or a counter and increments on every rising edge of the external source.

26.1.1 16-BIT MODE

In normal operation, TMR0 increments on the rising edge of the clock source. A 15-bit prescaler on the clock input gives several prescale options (see prescaler control bits, T0CKPS<3:0> in the T0CON1 register).

26.1.1.1 Timer0 Reads and Writes in 16-bit Mode

TMR0H is not the actual high byte of Timer0 in 16-bit mode. It is actually a buffered version of the real high byte of Timer0, which is neither directly readable nor writable (see Figure 26-1). TMR0H is updated with the contents of the high byte of Timer0 during a read of TMR0L. This provides the ability to read all 16 bits of Timer0 without having to verify that the read of the high and low byte was valid, due to a rollover between successive reads of the high and low byte.

Similarly, a write to the high byte of Timer0 must also take place through the TMR0H Buffer register. The high byte is updated with the contents of TMR0H when a write occurs to TMR0L. This allows all 16 bits of Timer0 to be updated at once.

26.1.2 8-BIT MODE

In normal operation, TMR0 increments on the rising edge of the clock source. A 15-bit prescaler on the clock input gives several prescale options (see prescaler control bits, T0CKPS<3:0> in the T0CON1 register).

In 8-bit mode, TMR0H no longer functions as the Timer0 high byte, but instead functions as the Period Register (PR). The value of TMR0L is compared to that of TMR0H on each clock cycle. When the two values match, the following events happen:

- TMR0_out goes high for one prescaled clock period
- TMR0L is reset
- The contents of TMR0H are copied to the period buffer

In 8-bit mode, the TMR0L and TMR0H registers are both directly readable and writable. The TMR0L register is cleared on any device Reset, while the TMR0H register initializes at FFh.

Both the prescaler and postscaler counters are cleared on the following events:

- A write to the TMR0L register
- A write to either the T0CON0 or T0CON1 registers.
- Any device Reset – Power-on Reset (POR),$\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset (WDTR) or Brown-out Reset (BOR)

26.1.3 COUNTER MODE

In Counter mode, the prescaler is normally disabled by setting the T0CKPS bits of the T0CON1 register to '0000'. Each rising edge of the clock input (or the output of the prescaler if the prescaler is used) increments the counter by '1'.

26.1.4 TIMER MODE

In Timer mode, the Timer0 module will increment every instruction cycle as long as there is a valid clock signal and the T0CKPS bits of the T0CON1 register (Register 26-4) are set to '0000'. When a prescaler is added, the timer will increment at the rate based on the prescaler value.

26.1.5 ASYNCHRONOUS MODE

When the T0ASYNC bit of the T0CON1 register is set (T0ASYNC = 1), the counter increments with each rising edge of the input source (or output of the prescaler, if used). Asynchronous mode allows the counter to continue operation during Sleep mode provided that the clock also continues to operate during Sleep.

EQUATION 29-2: PULSE WIDTH

$$Pulse\ Width = (CCPRxH{:}CCPRxL\ register\ pair) \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

EQUATION 29-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(CCPRxH{:}CCPRxL\ register\ pair)}{4(PR2 + 1)}$$

The CCPRxH:CCPRxL register pair and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering provides glitchless PWM operation.

The 8-bit timer TMR2/4/6 register is concatenated with either the 2-bit internal system clock (FOSC), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2/4/6 prescaler is set to 1:1.

When the 10-bit time base matches the CCPRxH:CCPRxL register pair, then the CCPx pin is cleared (see Figure 29-4).

29.4.6 PWM RESOLUTION

PWM resolution, expressed in number of bits, defines the maximum number of discrete steps that can be present in a single PWM period. For example, a 10-bit resolution will result in 1024 discrete steps, whereas an 8-bit resolution will result in 256 discrete steps.

The maximum PWM resolution is ten bits when PRx is 255. The resolution is a function of the PRx register value as shown by Equation 29-4.

EQUATION 29-4: PWM RESOLUTION

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

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

TABLE 29-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 20 MHz)

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale	16	4	1	1	1	1
PRx Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

TABLE 29-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 8 MHz)

PWM Frequency	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale	16	4	1	1	1	1
PRx Value	0x65	0x65	0x65	0x19	0x0C	0x09
Maximum Resolution (bits)	8	8	8	6	5	5

29.4.7 OPERATION IN SLEEP MODE

In Sleep mode, the TMR2/4/6 register will not increment and the state of the module will not change. If the CCPx pin is driving a value, it will continue to drive that value. When the device wakes up, TMR2/4/6 will continue from its previous state.

29.4.8 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section 7.0 "Oscillator Module"** for additional details.

29.4.9 EFFECTS OF RESET

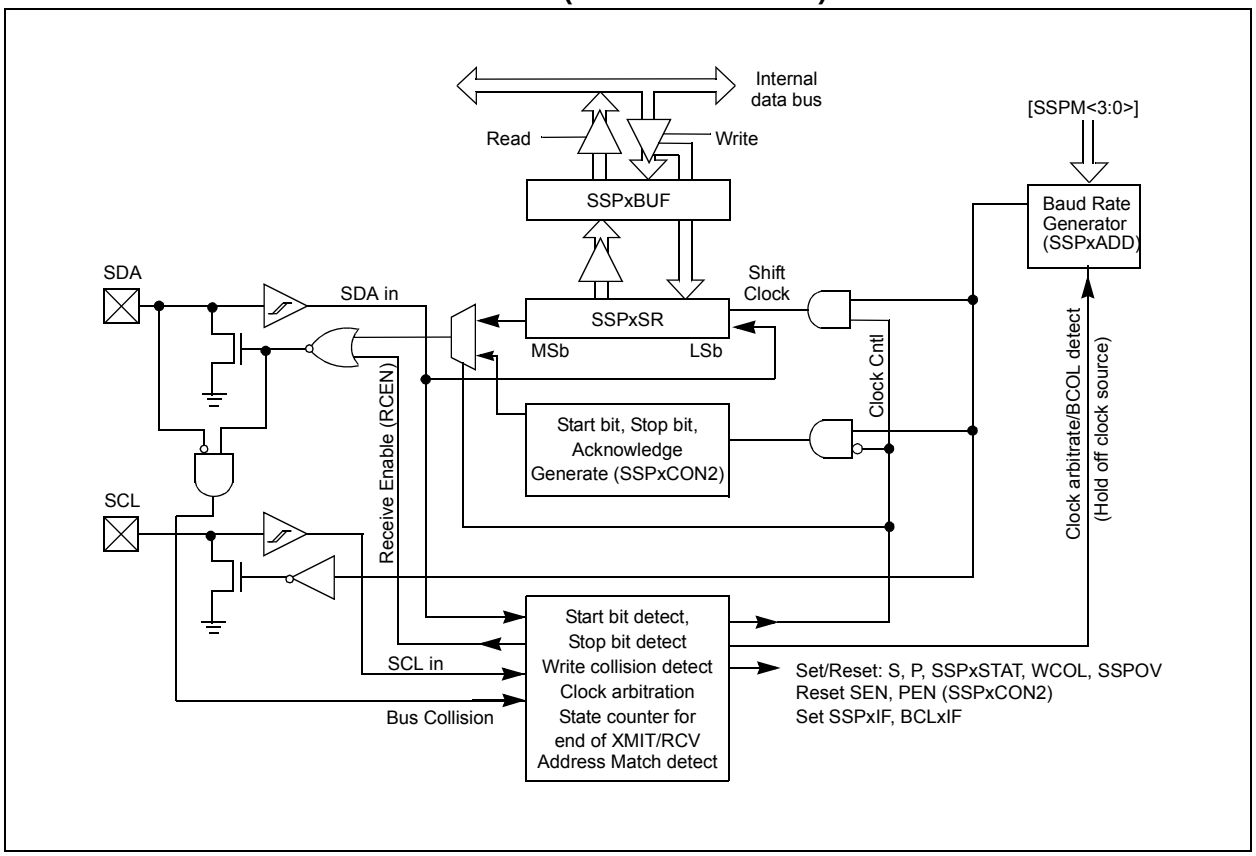
Any Reset will force all ports to Input mode and the CCP registers to their Reset states.

The I^2C interface supports the following modes and features:

- Master mode
- Slave mode
- Byte NACKing (Slave mode)
- Limited multi-master support
- 7-bit and 10-bit addressing
- Start and Stop interrupts
- Interrupt masking
- Clock stretching
- Bus collision detection
- General call address matching
- Address masking
- Selectable SDA hold times

Figure 30-2 is a block diagram of the I^2C interface module in Master mode. Figure 30-3 is a diagram of the I^2C interface module in Slave mode.

FIGURE 30-2: MSSP BLOCK DIAGRAM (I^2C MASTER MODE)



30.6 I²C Master Mode

Master mode is enabled by setting and clearing the appropriate SSPM<3:0> bits in the SSPxCON1 register and by setting the SSPEN bit. In Master mode, the SDA and SCK pins must be configured as inputs. The MSSP peripheral hardware will override the output driver TRIS controls when necessary to drive the pins low.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSPx module is disabled. Control of the I²C bus may be taken when the P bit is set, or the bus is Idle.

In Firmware Controlled Master mode, user code conducts all I²C bus operations based on Start and Stop bit condition detection. Start and Stop condition detection is the only active circuitry in this mode. All other communication is done by the user software directly manipulating the SDA and SCL lines.

The following events will cause the SSP Interrupt Flag bit, SSPxIF, to be set (SSP interrupt, if enabled):

- Start condition generation
- Stop condition generation
- Data transfer byte transmitted/received
- Acknowledge transmitted/received
- Repeated Start generated

Note 1: The MSSPx module, when configured in I²C Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPxBUF register to initiate transmission before the Start condition is complete. In this case, the SSPxBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPxBUF did not occur

2: When in Master mode, Start/Stop detection is masked and an interrupt is generated when the SEN/PEN bit is cleared and the generation is complete.

30.6.1 I²C MASTER MODE OPERATION

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/$\overline{W}$) bit. In this case, the R/$\overline{W}$ bit will be logic '0'. Serial data is transmitted eight bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/$\overline{W}$ bit. In this case, the R/$\overline{W}$ bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate the receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received eight bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

A Baud Rate Generator is used to set the clock frequency output on SCL. See **Section 30.7 "Baud Rate Generator"** for more detail.

31.6 Register Definitions: EUSART1 Control

REGISTER 31-1: TX1STA: TRANSMIT STATUS AND CONTROL 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-1/1	R/W-0/0
CSRC	TX9	TXEN(1)	SYNC	SENDB	BRGH	TRMT	TX9D
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 **CSRC:** Clock Source Select bit
Asynchronous mode:
Unused in this mode – value ignored
Synchronous mode:
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)
1 = Transmit enabled
0 = Transmit disabled

bit 4 **SYNC:** EUSART1 Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENDB:** Send Break Character bit
Asynchronous mode:
1 = Send SYNCH BREAK on next transmission – Start bit, followed by 12 '0' bits, followed by Stop bit; cleared by hardware upon completion
0 = SYNCH BREAK transmission disabled or completed
Synchronous mode:
Unused in this mode – value ignored

bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode – value ignored

bit 1 **TRMT:** Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** Ninth bit of Transmit Data
Can be address/data bit or a parity bit.

Note 1: SREN/CREN overrides TXEN in Sync mode.

REGISTER 31-4: RC1REG(1): RECEIVE DATA REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
RC1REG<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 **RC1REG<7:0>:** Lower eight bits of the received data; read-only; see also RX9D (Register 31-2)

Note 1: RC1REG (including the ninth bit) is double buffered, and data is available while new data is being received.

REGISTER 31-5: TX1REG(1): TRANSMIT DATA REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TX1REG<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 **TX1REG<7:0>:** Lower eight bits of the received data; read-only; see also RX9D (Register 31-1)

Note 1: TX1REG (including the ninth bit) is double buffered, and can be written when previous data has started shifting.

REGISTER 31-6: SP1BRGL(1): BAUD RATE GENERATOR REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SP1BRG<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 **SP1BRG<7:0>:** Lower eight bits of the Baud Rate Generator

Note 1: Writing to SP1BRG resets the BRG counter.

REGISTER 31-7: SP1BRGH(1, 2): BAUD RATE GENERATOR HIGH REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SP1BRG<15:8>							
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 **SP1BRG<15:8>:** Upper eight bits of the Baud Rate Generator

Note 1: SP1BRGH value is ignored for all modes unless BAUD1CON<BRG16> is active.

2: Writing to SP1BRGH resets the BRG counter.

TABLE 31-2: SUMMARY OF REGISTERS ASSOCIATED WITH EUSART1

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	143
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	144
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	149
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	150
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	156
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	157
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	101
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	108
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	103
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	384
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	383
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	385
RC1REG	RC1REG<7:0>								386
TX1REG	TX1REG<7:0>								386
SP1BRGL	SP1BRG<7:0>								386
SP1BRGH	SP1BRG<15:8>								387
RXPPS	—	—	—	RXPPS<4:0>					162
CLCxSELy	—	—	—	LCxDyS<4:0>					229
MDSRC	—	—	—	—	MDMS<3:0>				272

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for the EUSART1 module.

Note 1: PIC16(L)F18345 only.

2: Unimplemented, read as '1'.

TABLE 35-7: EXTERNAL CLOCK/OSCILLATOR TIMING REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
ECL Oscillator							
OS1	FECL	Clock Frequency	—	—	500	kHz	
OS2	TECL_DC	Clock Duty Cycle	40	—	60	%	
ECM Oscillator							
OS3	FECM	Clock Frequency	—	—	4	MHz	**Note 4**
OS4	TECM_DC	Clock Duty Cycle	40	—	60	%	
ECH Oscillator							
OS5	FECH	Clock Frequency	—	—	32	MHz	
OS6	TECH_DC	Clock Duty Cycle	40	—	60	%	
LP Oscillator							
OS7	FLP	Clock Frequency	—	—	100	kHz	**Note 4**
XT Oscillator							
OS8	FXT	Clock Frequency	—	—	4	MHz	**Note 4**
HS Oscillator							
OS9	FHS	Clock Frequency	—	—	20	MHz	**Note 4**
System Clock							
OS20	FOSC	System Clock Frequency	—	—	32	MHz	**Note 2, Note 3**
OS21	FCY	Instruction Frequency	—	FOSC/4	—	MHz	
OS22	TCY	Instruction Period	125	1/FCY	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

2: The system clock frequency (FOSC) is selected by the "main clock switch controls" as described in **Section 7.3 "Clock Switching"**.

3: The system clock frequency (FOSC) must meet the voltage requirements defined in the **Section 35.2 "Standard Operating Conditions"**. LP, XT and HS oscillator modes require an appropriate crystal or resonator to be connected to the device.

4: For clocking the device with an external square wave, one of the EC mode selections must be used.

FIGURE 35-17: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)

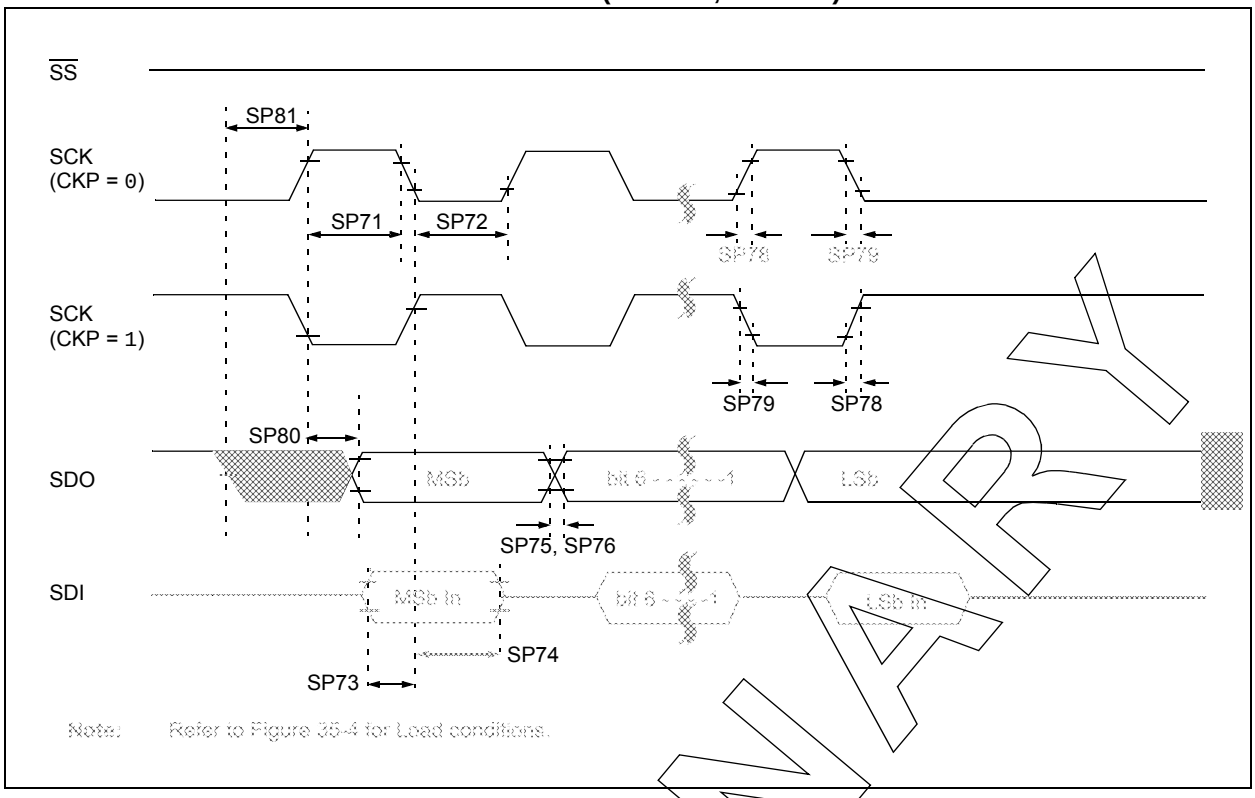
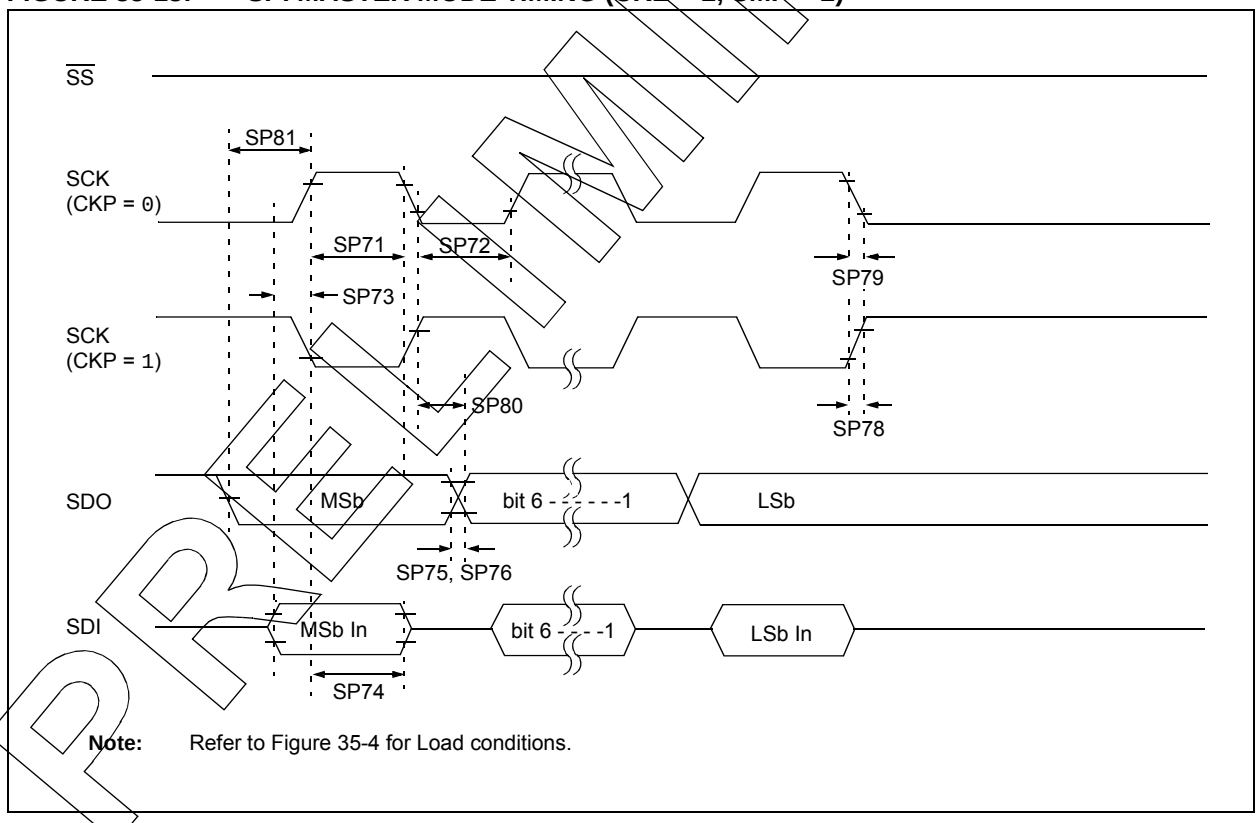
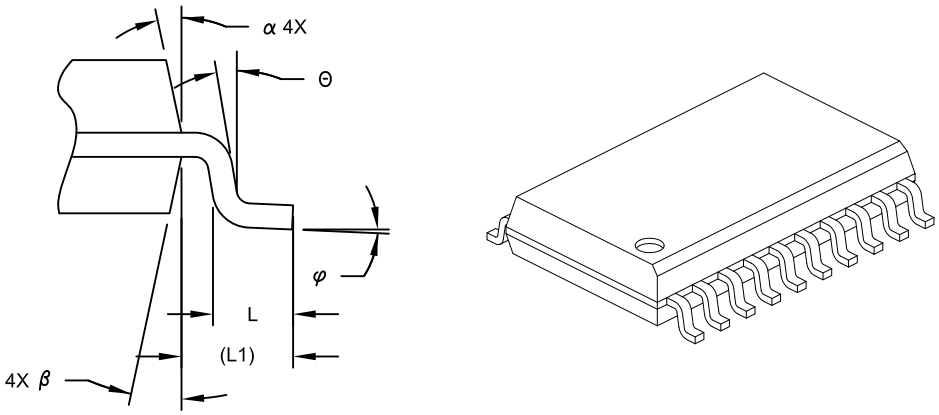


FIGURE 35-18: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)



20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		20	
Pitch	e		1.27 BSC	
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E		10.30 BSC	
Molded Package Width	E1		7.50 BSC	
Overall Length	D		12.80 BSC	
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1		1.40 REF	
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.20	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-094C Sheet 2 of 2

