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	20MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	368 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f87-i-ml

PIC16F87/88

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 2-1.

The Special Function Registers can be classified into two sets: core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. Those related to the operation of the peripheral features are described in detail in the peripheral feature section.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page
Bank 0											
00h[2]	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	26, 135
01h	TMR0	Timer0 Module Register								xxxx xxxx	69
02h[2]	PCL	Program Counter (PC) Least Significant Byte								0000 0000	
03h[2]	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	17
04h[2]	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	135
05h	PORTA	PORTA Data Latch when written; PORTA pins when read (PIC16F87) PORTA Data Latch when written; PORTA pins when read (PIC16F88)								xxxx 0000 xxx0 0000	52
06h	PORTB	PORTB Data Latch when written; PORTB pins when read (PIC16F87) PORTB Data Latch when written; PORTB pins when read (PIC16F88)								xxxx xxxx 00xx xxxx	58
07h	—	Unimplemented								—	—
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah[1,2]	PCLATH	—	—	—	Write Buffer for the Upper 5 bits of the Program Counter					---0 0000	135
0Bh[2]	INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	19, 69, 77
0Ch	PIR1	—	ADIF[4]	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	21, 77
0Dh	PIR2	OSFIF	CMIF	—	EEIF	—	—	—	—	00-0 ----	23, 34
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	77, 83
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	77, 83
10h	T1CON	—	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	-000 0000	72, 83
11h	TMR2	Timer2 Module Register								0000 0000	80, 85
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	80, 85
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	90, 95
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	89, 95
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	83, 85
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	83, 85
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	81, 83
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	98, 99
19h	TXREG	AUSART Transmit Data Register								0000 0000	103
1Ah	RCREG	AUSART Receive Data Register								0000 0000	105
1Bh	—	Unimplemented								—	—
1Ch	—	Unimplemented								—	—
1Dh	—	Unimplemented								—	—
1Eh	ADRESH[4]	A/D Result Register High Byte								xxxx xxxx	120
1Fh	ADCON0[4]	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON	0000 00-0	114, 120

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<12:8>, whose contents are transferred to the upper byte of the program counter.

2: These registers can be addressed from any bank.

3: RA5 is an input only; the state of the TRISA5 bit has no effect and will always read '1'.

4: PIC16F88 device only.

PIC16F87/88

4.5.1   INTRC MODES

Using the internal oscillator as the clock source can eliminate the need for up to two external oscillator pins, after which it can be used for digital I/O. Two distinct configurations are available:

• In INTIO1 mode, the OSC2 pin outputs FOSC/4, while OSC1 functions as RA7 for digital input and output.
• In INTIO2 mode, OSC1 functions as RA7 and OSC2 functions as RA6, both for digital input and output.

4.5.2   OSCTUNE REGISTER

The internal oscillator's output has been calibrated at the factory but can be adjusted in the application. This is done by writing to the OSCTUNE register (Register 4-1). The tuning sensitivity is constant throughout the tuning range. The OSCTUNE register has a tuning range of ±12.5%.

When the OSCTUNE register is modified, the INTOSC and INTRC frequencies will begin shifting to the new frequency. The INTRC clock will reach the new frequency within 8 clock cycles (approximately 8 * 32 μs = 256 μs); the INTOSC clock will stabilize within 1 ms. Code execution continues during this shift. There is no indication that the shift has occurred. Operation of features that depend on the 31.25 kHz INTRC clock source frequency, such as the WDT, Fail-Safe Clock Monitor and peripherals, will also be affected by the change in frequency.

REGISTER 4-1:   OSCTUNE: OSCILLATOR TUNING REGISTER (ADDRESS 90h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0

bit 7   bit 0

bit 7-6   **Unimplemented:** Read as '0'

bit 5-0   **TUN<5:0>:** Frequency Tuning bits

011111 = Maximum frequency
011110 =
•
•
•
000001 =
000000 = Center frequency. Oscillator module is running at the calibrated frequency.
111111 =
•
•
•
100000 = Minimum frequency

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

PIC16F87/88

• Clock before switch: One of INTOSC/INTOSC postscaler (IRCF<2:0> ≠ 000)

1. IRCF bits are modified to a different INTOSC/INTOSC postscaler frequency.
2. The clock switching circuitry waits for a falling edge of the current clock, at which point CLKO is held low.
3. The clock switching circuitry then waits for eight falling edges of requested clock, after which it switches CLKO to this new clock source.
4. The IOFS bit is set.
5. Oscillator switchover is complete.

4.6.6 OSCILLATOR DELAY UPON POWER-UP, WAKE-UP AND CLOCK SWITCHING

Table 4-3 shows the different delays invoked for various clock switching sequences. It also shows the delays invoked for POR and wake-up.

TABLE 4-3: OSCILLATOR DELAY EXAMPLES

Clock Switch		Frequency	Oscillator Delay	Comments
From	**To**			
Sleep/POR	INTRC T1OSC	31.25 kHz 32.768 kHz	CPU Start-up[1]	Following a wake-up from Sleep mode or POR, CPU start-up is invoked to allow the CPU to become ready for code execution.
	INTOSC/ INTOSC Postscaler	125 kHz-8 MHz	4 ms (approx.) and CPU Start-up[1]	
INTRC/Sleep	EC, RC	DC – 20 MHz		
INTRC (31.25 kHz)	EC, RC	DC – 20 MHz		
Sleep	LP, XT, HS	32.768 kHz-20 MHz	1024 Clock Cycles (OST)	Following a change from INTRC, an OST of 1024 cycles must occur.
INTRC (31.25 kHz)	INTOSC/ INTOSC Postscaler	125 kHz-8 MHz	4 ms (approx.)	Refer to **Section 4.6.4 "Modifying the IRCF Bits"** for further details.

Note 1: The 5-10 μs start-up delay is based on a 1 MHz system clock.

PIC16F87/88

6.3 Using Timer0 with an External Clock

When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI, with the internal phase clocks, is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, it is necessary for T0CKI to be high for at least 2 T$_{OSC}$ (and a small RC delay of 20 ns) and low for at least 2 T$_{OSC}$ (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

6.4 Prescaler

There is only one prescaler available, which is mutually exclusively shared between the Timer0 module and the Watchdog Timer. A prescaler assignment for the Timer0 module means that the prescaler cannot be used by the Watchdog Timer and vice versa. This prescaler is not readable or writable (see Figure 6-1).

> **Note:** Although the prescaler can be assigned to either the WDT or Timer0, but not both, a new divide counter is implemented in the WDT circuit to give multiple WDT time-out selections. This allows TMR0 and WDT to each have their own scaler. Refer to **Section 15.12 "Watchdog Timer (WDT)"** for further details.

The PSA and PS2:PS0 bits (OPTION_REG<3:0>) determine the prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF 1`, `MOVWF 1`, `BSF 1,x`....etc.) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer. The prescaler is not readable or writable.

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

REGISTER 6-1: **OPTION_REG: OPTION CONTROL REGISTER (ADDRESS 81h, 181h)**

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

bit 7   bit 0

bit 7 **RBPU:** PORTB Pull-up Enable bit

bit 6 **INTEDG:** Interrupt Edge Select bit

bit 5 **T0CS**: TMR0 Clock Source Select bit
`1` = Transition on T0CKI pin
`0` = Internal instruction cycle clock (CLKO)

bit 4 **T0SE**: TMR0 Source Edge Select bit
`1` = Increment on high-to-low transition on T0CKI pin
`0` = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA**: Prescaler Assignment bit
`1` = Prescaler is assigned to the WDT
`0` = Prescaler is assigned to the Timer0 module

bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
`000`	1 : 2	1 : 1
`001`	1 : 4	1 : 2
`010`	1 : 8	1 : 4
`011`	1 : 16	1 : 8
`100`	1 : 32	1 : 16
`101`	1 : 64	1 : 32
`110`	1 : 128	1 : 64
`111`	1 : 256	1 : 128

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

> **Note:** To avoid an unintended device Reset, the instruction sequence shown in the *"PIC® Mid-Range MCU Family Reference Manual"* (DS33023) must be executed when changing the prescaler assignment from Timer0 to the WDT. This sequence must be followed even if the WDT is disabled.

The maximum PWM resolution (bits) for a given PWM frequency is given by the following formula.

EQUATION 9-3:

$$\text{Resolution} = \frac{\log\left(\frac{F_{OSC}}{F_{PWM}}\right)}{\log(2)} \text{ bits}$$

Note:	If the PWM duty cycle value is longer than the PWM period, the CCP1 pin will not be cleared.

9.3.3 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPR1L register and CCP1CON<5:4> bits.
3. Make the CCP1 pin an output by clearing the TRISB<x> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCP1 module for PWM operation.

Note:	The TRISB bit (0 or 3) is dependant upon the setting of configuration bit 12 (CCPMX).

TABLE 9-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHz

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

TABLE 9-4: REGISTERS ASSOCIATED WITH PWM AND TIMER2

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF[1]	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
8Ch	PIE1	—	ADIE[1]	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
11h	TMR2	Timer2 Module Register								0000 0000	0000 0000
92h	PR2	Timer2 Period Register								1111 1111	1111 1111
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by PWM and Timer2.
Note 1: This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

PIC16F87/88

FIGURE 10-1:   SSP BLOCK DIAGRAM (SPI MODE)
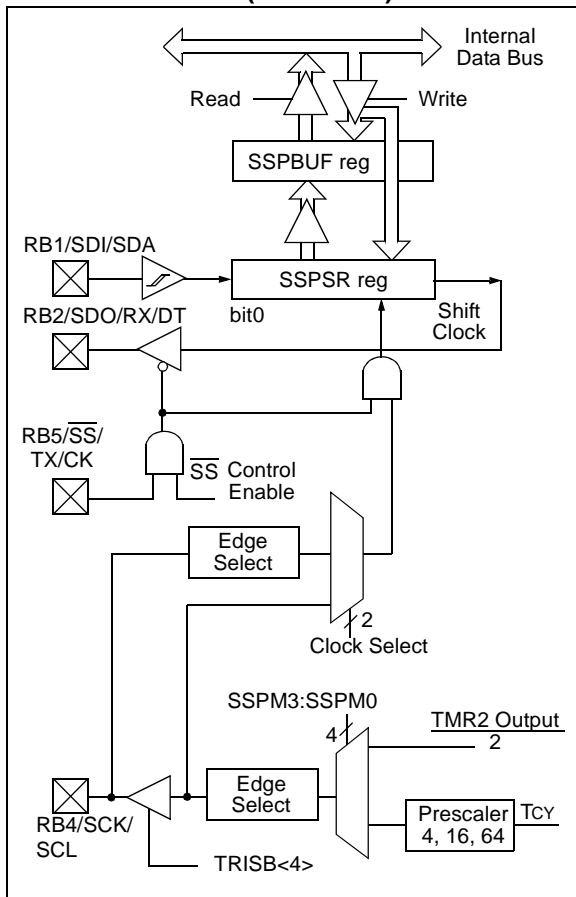
To enable the serial port, SSP Enable bit, SSPEN (SSPCON<5>), must be set. To reset or reconfigure SPI mode, clear bit SSPEN, reinitialize the SSPCON register and then set bit SSPEN. This configures the SDI, SDO, SCK and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, they must have their data direction bits (in the TRISB register) appropriately programmed. That is:

- SDI must have TRISB<1> set
- SDO must have TRISB<2> cleared
- SCK (Master mode) must have TRISB<4> cleared
- SCK (Slave mode) must have TRISB<4> set
- $\overline{SS}$ must have TRISB<5> set

> **Note 1:** When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPCON<3:0> = 0100), the SPI module will reset if the $\overline{SS}$ pin is set to VDD.
>
> **2:** If the SPI is used in Slave mode with CKE = 1, then the $\overline{SS}$ pin control must be enabled.

TABLE 10-1:   REGISTERS ASSOCIATED WITH SPI OPERATION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF[1]	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
8Ch	PIE1	—	ADIE[1]	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the SSP in SPI mode.

Note   1:   This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

10.3.1 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs (TRISB<4,1> set). The SSP module will override the input state with the output data when required (slave-transmitter).

When an address is matched, or the data transfer after an address match is received, the hardware automatically will generate the Acknowledge ($\overline{ACK}$) pulse and then load the SSPBUF register with the received value currently in the SSPSR register.

Either or both of the following conditions will cause the SSP module not to give this $\overline{ACK}$ pulse:

a) The Buffer Full bit, BF (SSPSTAT<0>), was set before the transfer was received.

b) The Overflow bit, SSPOV (SSPCON<6>), was set before the transfer was received.

In this case, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF (PIR1<3>) is set. Table 10-2 shows what happens when a data transfer byte is received, given the status of bits BF and SSPOV. The shaded cells show the condition where user software did not properly clear the overflow condition. Flag bit, BF, is cleared by reading the SSPBUF register while bit, SSPOV, is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the I^2C specification, as well as the requirement of the SSP module, are shown in timing parameter #100 and parameter #101.

10.3.1.1 Addressing

Once the SSP module has been enabled, it waits for a Start condition to occur. Following the Start condition, the eight bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match and the BF and SSPOV bits are clear, the following events occur:

a) The SSPSR register value is loaded into the SSPBUF register.

b) The Buffer Full bit, BF, is set.

c) An $\overline{ACK}$ pulse is generated.

d) SSP Interrupt Flag bit, SSPIF (PIR1<3>), is set (interrupt is generated if enabled) – on the falling edge of the ninth SCL pulse.

In 10-bit Address mode, two address bytes need to be received by the slave device. The five Most Significant bits (MSbs) of the first address byte specify if this is a 10-bit address. Bit R/$\overline{W}$ (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSbs of the address.

The sequence of events for 10-bit Address mode is as follows, with steps 7-9 for slave transmitter:

1. Receive first (high) byte of address (bits SSPIF, BF and UA (SSPSTAT<1>) are set).

2. Update the SSPADD register with second (low) byte of address (clears bit UA and releases the SCL line).

3. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

4. Receive second (low) byte of address (bits SSPIF, BF and UA are set).

5. Update the SSPADD register with the first (high) byte of address; if match releases SCL line, this will clear bit UA.

6. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

7. Receive Repeated Start condition.

8. Receive first (high) byte of address (bits SSPIF and BF are set).

9. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

10.3.1.2 Reception

When the R/$\overline{W}$ bit of the address byte is clear and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register.

When the address byte overflow condition exists, then a no Acknowledge ($\overline{ACK}$) pulse is given. An overflow condition is indicated if either bit, BF (SSPSTAT<0>), is set or bit, SSPOV (SSPCON<6>), is set.

An SSP interrupt is generated for each data transfer byte. Flag bit, SSPIF (PIR1<3>), must be cleared in software. The SSPSTAT register is used to determine the status of the byte.

10.3.1.3 Transmission

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The $\overline{ACK}$ pulse will be sent on the ninth bit and pin RB4/SCK/SCL is held low. The transmit data must be loaded into the SSPBUF register which also loads the SSPSR register. Then, pin RB4/SCK/SCL should be enabled by setting bit CKP (SSPCON<4>). The master device must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master device by stretching the clock. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 10-7).

PIC16F87/88

11.2 AUSART Asynchronous Mode

In this mode, the AUSART uses standard Non-Return-to-Zero (NRZ) format (one Start bit, eight or nine data bits and one Stop bit). The most common data format is 8 bits. An on-chip, dedicated, 8-bit Baud Rate Generator can be used to derive standard baud rate frequencies from the oscillator. The AUSART transmits and receives the LSb first. The transmitter and receiver are functionally independent, but use the same data format and baud rate. The Baud Rate Generator produces a clock, either x16 or x64 of the bit shift rate, depending on bit BRGH (TXSTA<2>). Parity is not supported by the hardware, but can be implemented in software (and stored as the ninth data bit). Asynchronous mode is stopped during Sleep.

Asynchronous mode is selected by clearing bit SYNC (TXSTA<4>).

The AUSART Asynchronous module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver

11.2.1 AUSART ASYNCHRONOUS TRANSMITTER

The AUSART transmitter block diagram is shown in Figure 11-1. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The Shift register obtains its data from the Read/Write Transmit Buffer register, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXREG register (if available). Once the TXREG register transfers the data to the TSR register (occurs in one T_{CY}), the TXREG register is empty and flag bit, TXIF (PIR1<4>), is set. This interrupt can be enabled/disabled by setting/clearing enable bit, TXIE (PIE1<4>). Flag bit TXIF will be set, regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. Status bit TRMT is a read-only bit which is set when the TSR register is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty.

> **Note 1:** The TSR register is not mapped in data memory, so it is not available to the user.
>
> **2:** Flag bit TXIF is set when enable bit TXEN is set. TXIF is cleared by loading TXREG.

Transmission is enabled by setting enable bit TXEN (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data and the Baud Rate Generator (BRG) has produced a shift clock (Figure 11-2). The transmission can also be started by first loading the TXREG register and then setting enable bit TXEN. Normally, when transmission is first started, the TSR register is empty. At that point, transfer to the TXREG register will result in an immediate transfer to TSR, resulting in an empty TXREG. A back-to-back transfer is thus possible (Figure 11-3). Clearing enable bit TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. As a result, the RB5/$\overline{SS}$/TX/CK pin will revert to high-impedance.

In order to select 9-bit transmission, transmit bit, TX9 (TXSTA<6>), should be set and the ninth bit should be written to TX9D (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG register. This is because a data write to the TXREG register can result in an immediate transfer of the data to the TSR register (if the TSR is empty). In such a case, an incorrect ninth data bit may be loaded in the TSR register.

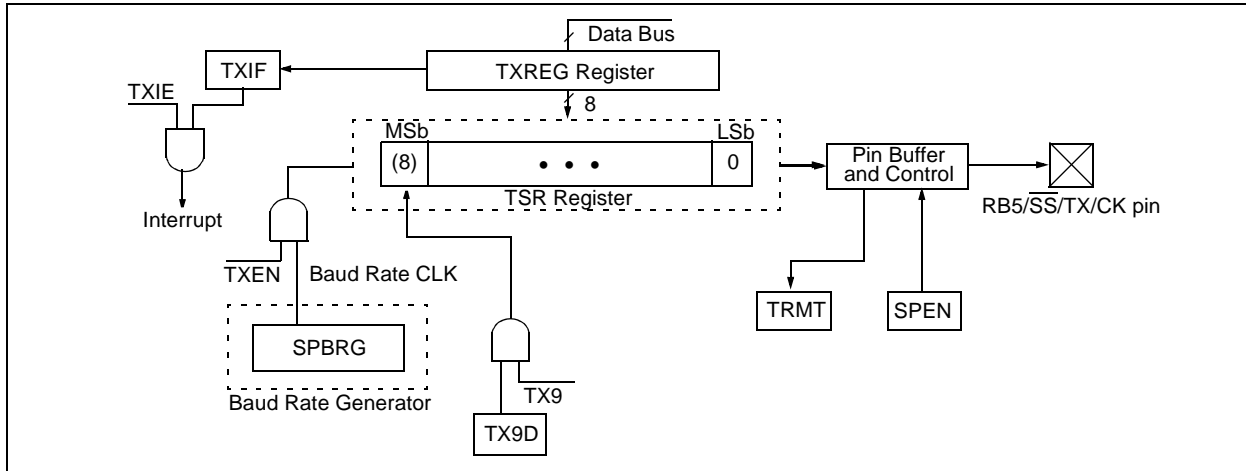**FIGURE 11-1:** **AUSART TRANSMIT BLOCK DIAGRAM**

TABLE 11-10: **REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF[(1)]	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	AUSART Transmit Data Register								0000 0000	0000 0000
8Ch	PIE1	—	ADIE[(1)]	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.

Note 1: This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

FIGURE 11-9: **SYNCHRONOUS TRANSMISSION**

Note: Sync Master mode; SPBRG = 0. Continuous transmission of two 8-bit words.
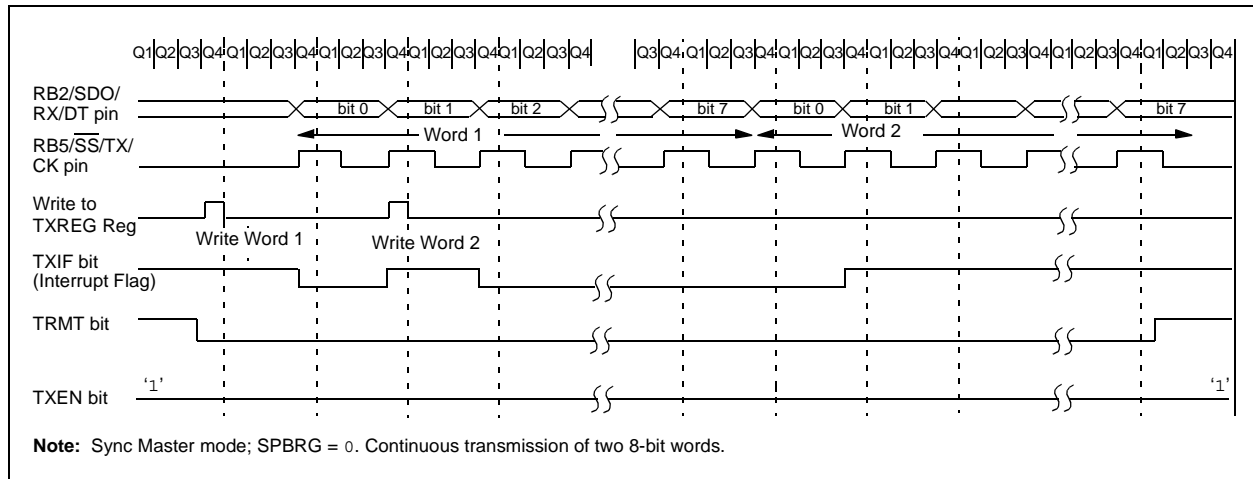
FIGURE 11-10: **SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**
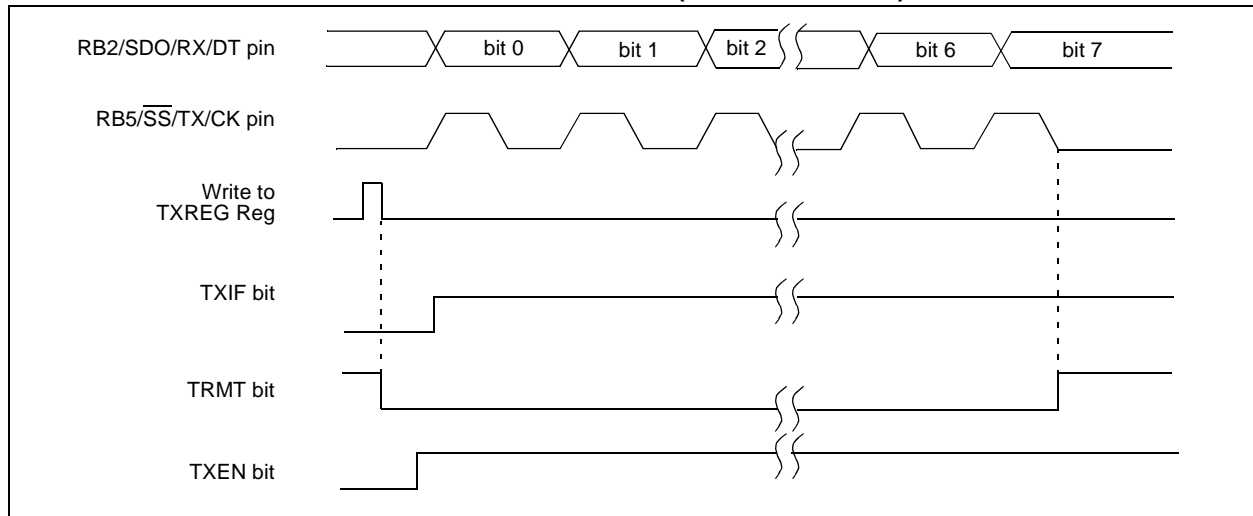
PIC16F87/88

11.3.2 AUSART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either enable bit SREN (RCSTA<5>), or enable bit CREN (RCSTA<4>). Data is sampled on the RB2/SDO/RX/DT pin on the falling edge of the clock. If enable bit SREN is set, then only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set, CREN takes precedence.

After clocking the last bit, the received data in the Receive Shift Register (RSR) is transferred to the RCREG register (if it is empty). When the transfer is complete, interrupt flag bit, RCIF (PIR1<5>), is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit RCIE (PIE1<5>).

Flag bit RCIF is a read-only bit which is reset by the hardware. In this case, it is reset when the RCREG register has been read and is empty. The RCREG is a double-buffered register (i.e., it is a two-deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR register. On the clocking of the last bit of the third byte, if the RCREG register is still full, then Overrun Error bit, OERR (RCSTA<1>), is set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Bit OERR has to be cleared in software (by clearing bit CREN). If bit OERR is set, transfers from the RSR to the RCREG are inhibited, so it is essential to clear bit OERR if it is set. The ninth receive bit is buffered the same way as the receive data. Reading the RCREG register will load bit RX9D with a new value, therefore, it is essential for the user to read the RCSTA register, before reading RCREG, in order not to lose the old RX9D information.

When setting up a synchronous master reception:

1. Initialize the SPBRG register for the appropriate baud rate (**Section 11.1 "AUSART Baud Rate Generator (BRG)"**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Ensure bits CREN and SREN are clear.
4. If interrupts are desired, then set enable bit RCIE.
5. If 9-bit reception is desired, then set bit RX9.
6. If a single reception is required, set bit SREN. For continuous reception, set bit CREN.
7. Interrupt flag bit, RCIF, will be set when reception is complete and an interrupt will be generated if enable bit, RCIE, was set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing bit CREN.
11. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

TABLE 11-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF[1]	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	AUSART Receive Data Register								0000 0000	0000 0000
8Ch	PIE1	—	ADIE[1]	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous master reception.

Note 1: This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

12.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) converter module has seven inputs for 18/20 pin devices (PIC16F88 devices only).

The conversion of an analog input signal results in a corresponding 10-bit digital number. The A/D module has a high and low-voltage reference input that is software selectable to some combination of V_{DD}, V_{SS}, V_{REF}- (RA2) or V_{REF}+ (RA3).

The A/D converter has a unique feature of being able to operate while the device is in Sleep mode. To operate in Sleep, the A/D conversion clock must be derived from the A/D's internal RC oscillator.

The A/D module has five registers:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)
- Analog Select Register (ANSEL)

The ADCON0 register, shown in Register 12-2, controls the operation of the A/D module. The ANSEL register, shown in Register 12-1 and the ADCON1 register, shown in Register 12-3, configure the functions of the port pins. The port pins can be configured as analog inputs (RA3/RA2 can also be voltage references) or as digital I/O.

Additional information on using the A/D module can be found in the "*PIC® Mid-Range MCU Family Reference Manual*" (DS33023).

REGISTER 12-1:   ANSEL: ANALOG SELECT REGISTER (ADDRESS 9Bh) PIC16F88 DEVICES ONLY

U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
bit 7							bit 0

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

bit 6-0   **ANS<6:0>:** Analog Input Select bits

Bits select input function on corresponding AN<6:0> pins.

1 = Analog I/O[1,2]

0 = Digital I/O

Note 1:   Setting a pin to an analog input disables the digital input buffer. The corresponding TRIS bit should be set to input mode when using pins as analog inputs. Only AN2 is an analog I/O, all other ANx pins are analog inputs.

2:   See the block diagrams for the analog I/O pins to see how ANSEL interacts with the CHS bits of the ADCON0 register.

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

15.0 SPECIAL FEATURES OF THE CPU

These devices have a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power-saving operating modes and offer code protection:

• Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
• Interrupts
• Watchdog Timer (WDT)
• Two-Speed Start-up
• Fail-Safe Clock Monitor
• Sleep
• Code Protection
• ID Locations
• In-Circuit Serial Programming™ (ICSP™)

There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only. It is designed to keep the part in Reset while the power supply stabilizes and is enabled or disabled using a configuration bit. With these two timers on-chip, most applications need no external Reset circuitry.

Sleep mode is designed to offer a very low-current Power-down mode. The user can wake-up from Sleep through external Reset, Watchdog Timer wake-up or through an interrupt.

Additional information on special features is available in the "*PIC® Mid-Range MCU Family Reference Manual*" (DS33023).

15.1 Configuration Bits

The configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1'), to select various device configurations. These bits are mapped in program memory locations 2007h and 2008h.

The user will note that address 2007h is beyond the user program memory space which can be accessed only during programming.

15.12.3 TWO-SPEED CLOCK START-UP MODE

Two-Speed Start-up mode minimizes the latency between oscillator start-up and code execution that may be selected with the IESO (Internal/External Switchover) bit in Configuration Word 2. This mode is achieved by initially using the INTRC for code execution until the primary oscillator is stable.

If this mode is enabled and any of the following conditions exist, the system will begin execution with the INTRC oscillator. This results in almost immediate code execution with a minimum of delay.

• POR and after the Power-up Timer has expired (if $\overline{PWRTEN}$ = 0);

• or following a wake-up from Sleep;

• or a Reset when running from T1OSC or INTRC (after a Reset, SCS<1:0> are always set to '00').

> **Note:** Following any Reset, the IRCF bits are zeroed and the frequency selection is forced to 31.25 kHz. The user can modify the IRCF bits to select a higher internal oscillator frequency.

If the primary oscillator is configured to be anything other than XT, LP or HS, then Two-Speed Start-up mode is disabled because the primary oscillator will not require any time to become stable after POR, or an exit from Sleep.

If the IRCF bits of the OSCCON register are configured to a non-zero value prior to entering Sleep mode, the system clock frequency will come from the output of the INTOSC. The IOFS bit in the OSCCON register will be clear until the INTOSC is stable. This will allow the user to determine when the internal oscillator can be used for time critical applications.

Checking the state of the OSTS bit will confirm whether the primary clock configuration is engaged. If not, the OSTS bit will remain clear.

When the device is auto-configured in INTRC mode following a POR or wake-up from Sleep, the rules for entering other oscillator modes still apply, meaning the SCS<1:0> bits in OSCCON can be modified before the OST time-out has occurred. This would allow the application to wake-up from Sleep, perform a few instructions using the INTRC as the clock source and go back to Sleep without waiting for the primary oscillator to become stable.

> **Note:** Executing a `SLEEP` instruction will abort the oscillator start-up time and will cause the OSTS bit to remain clear.

15.12.3.1 Two-Speed Start-up Mode Sequence

1. Wake-up from Sleep, Reset or POR.
2. OSCCON bits configured to run from INTRC (31.25 kHz).
3. Instructions begin execution by INTRC (31.25 kHz).
4. OST enabled to count 1024 clock cycles.
5. OST timed out, wait for falling edge of INTRC.
6. OSTS is set.
7. System clock held low for eight falling edges of new clock (LP, XT or HS).
8. System clock is switched to primary source (LP, XT or HS).

The software may read the OSTS bit to determine when the switchover takes place so that any software timing edges can be adjusted.

FIGURE 15-9:   TWO-SPEED START-UP MODE
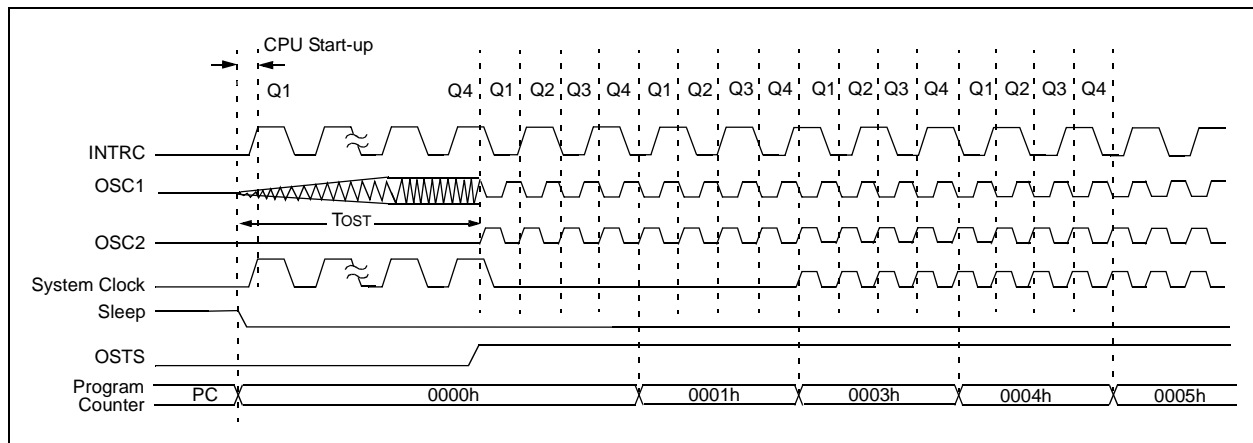
BTFSS	Bit Test f, Skip if Set
Syntax:	[*label*] BTFSS f,b
Operands:	$0 \le f \le 127$ $0 \le b < 7$
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' = 0, the next instruction is executed. If bit 'b' = 1, then the next instruction is discarded and a NOP is executed instead, making this a 2 TCY instruction.

CLRF	Clear f
Syntax:	[*label*] CLRF f
Operands:	$0 \le f \le 127$
Operation:	00h $\rightarrow$ (f), 1 $\rightarrow$ Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

BTFSC	Bit Test, Skip if Clear
Syntax:	[*label*] BTFSC f,b
Operands:	$0 \le f \le 127$ $0 \le b \le 7$
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' = 1, the next instruction is executed. If bit 'b', in register 'f', = 0, the next instruction is discarded and a NOP is executed instead, making this a 2 TCY instruction.

CLRW	Clear W
Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h $\rightarrow$ (W), 1 $\rightarrow$ Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

CALL	Call Subroutine
Syntax:	[*label*] CALL k
Operands:	$0 \le k \le 2047$
Operation:	(PC) + 1 $\rightarrow$ TOS, k $\rightarrow$ PC<10:0>, (PCLATH<4:3>) $\rightarrow$ PC<12:11>
Status Affected:	None
Description:	Call subroutine. First, return address (PC + 1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.

CLRWDT	Clear Watchdog Timer
Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h $\rightarrow$ WDT, 0 $\rightarrow$ WDT prescaler, 1 $\rightarrow$ $\overline{\text{TO}}$, 1 $\rightarrow$ $\overline{\text{PD}}$
Status Affected:	$\overline{\text{TO}}$, $\overline{\text{PD}}$
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{\text{TO}}$ and $\overline{\text{PD}}$ are set.

PIC16F87/88

TABLE 18-10: I²C™ BUS DATA REQUIREMENTS

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
100*	T$_{HIGH}$	Clock High Time	100 kHz mode	4.0	—	µs	
			400 kHz mode	0.6	—	µs	
			SSP Module	1.5 T$_{CY}$	—		
101*	T$_{LOW}$	Clock Low Time	100 kHz mode	4.7	—	µs	
			400 kHz mode	1.3	—	µs	
			SSP Module	1.5 T$_{CY}$	—		
102*	T$_R$	SDA and SCL Rise Time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1 C$_B$	300	ns	C$_B$ is specified to be from 10-400 pF
103*	T$_F$	SDA and SCL Fall Time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1 C$_B$	300	ns	C$_B$ is specified to be from 10-400 pF
90*	T$_{SU:STA}$	Start Condition Setup Time	100 kHz mode	4.7	—	µs	Only relevant for Repeated Start condition
			400 kHz mode	0.6	—	µs	
91*	T$_{HD:STA}$	Start Condition Hold Time	100 kHz mode	4.0	—	µs	After this period, the first clock pulse is generated
			400 kHz mode	0.6	—	µs	
106*	T$_{HD:DAT}$	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	µs	
107*	T$_{SU:DAT}$	Data Input Setup Time	100 kHz mode	250	—	ns	(Note 2)
			400 kHz mode	100	—	ns	
92*	T$_{SU:STO}$	Stop Condition Setup Time	100 kHz mode	4.7	—	µs	
			400 kHz mode	0.6	—	µs	
109*	T$_{AA}$	Output Valid from Clock	100 kHz mode	—	3500	ns	(Note 1)
			400 kHz mode	—	—	ns	
110*	T$_{BUF}$	Bus Free Time	100 kHz mode	4.7	—	µs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	µs	
	C$_B$	Bus Capacitive Loading		—	400	pF	

* These parameters are characterized but not tested.

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

2: A Fast mode (400 kHz) I²C™ bus device can be used in a Standard mode (100 kHz) I²C bus system, but the requirement, T$_{SU:DAT}$ ≥ 250 ns, must then be met. This will automatically be the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line, T$_R$ max. + T$_{SU:DAT}$ = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

PIC16F87/88

FIGURE 19-15: ΔIPD WDT, -40°C TO +125°C (SLEEP MODE, ALL PERIPHERALS DISABLED)
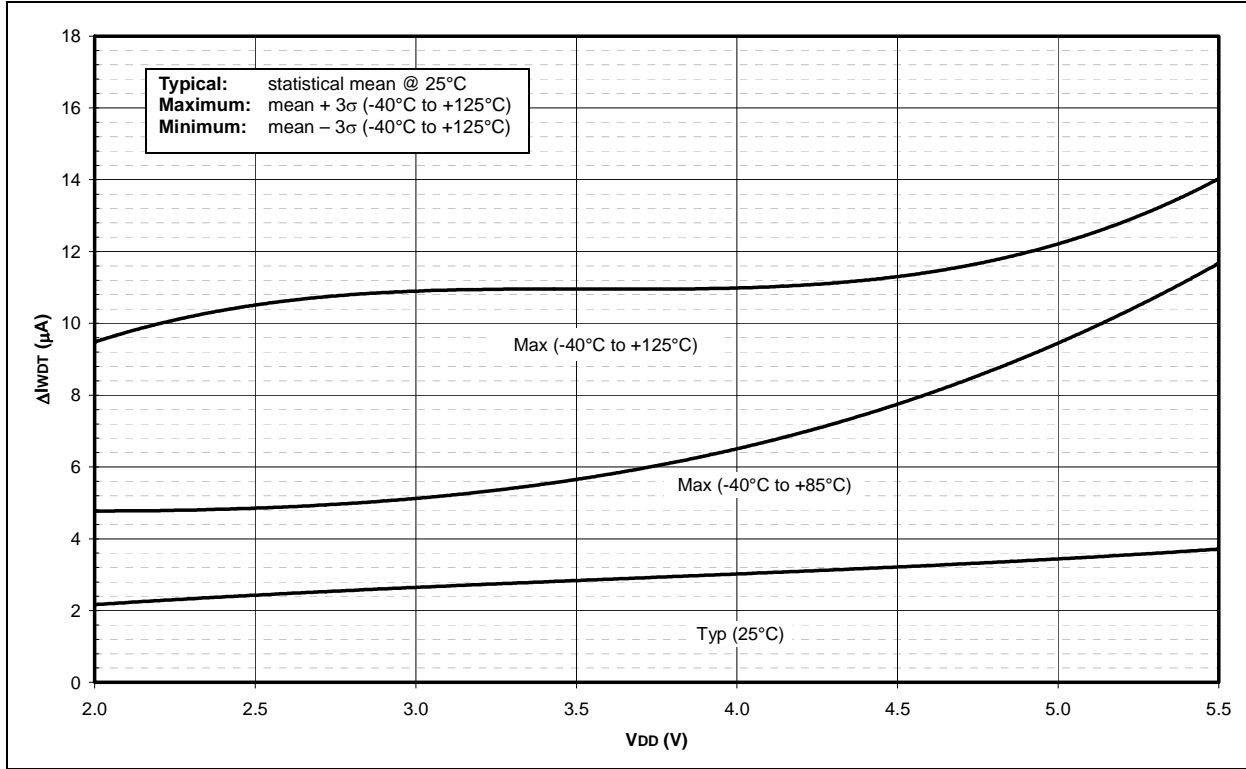
FIGURE 19-16: ΔIPD BOR vs. VDD, -40°C TO +125°C (SLEEP MODE, BOR ENABLED AT 2.00V-2.16V)
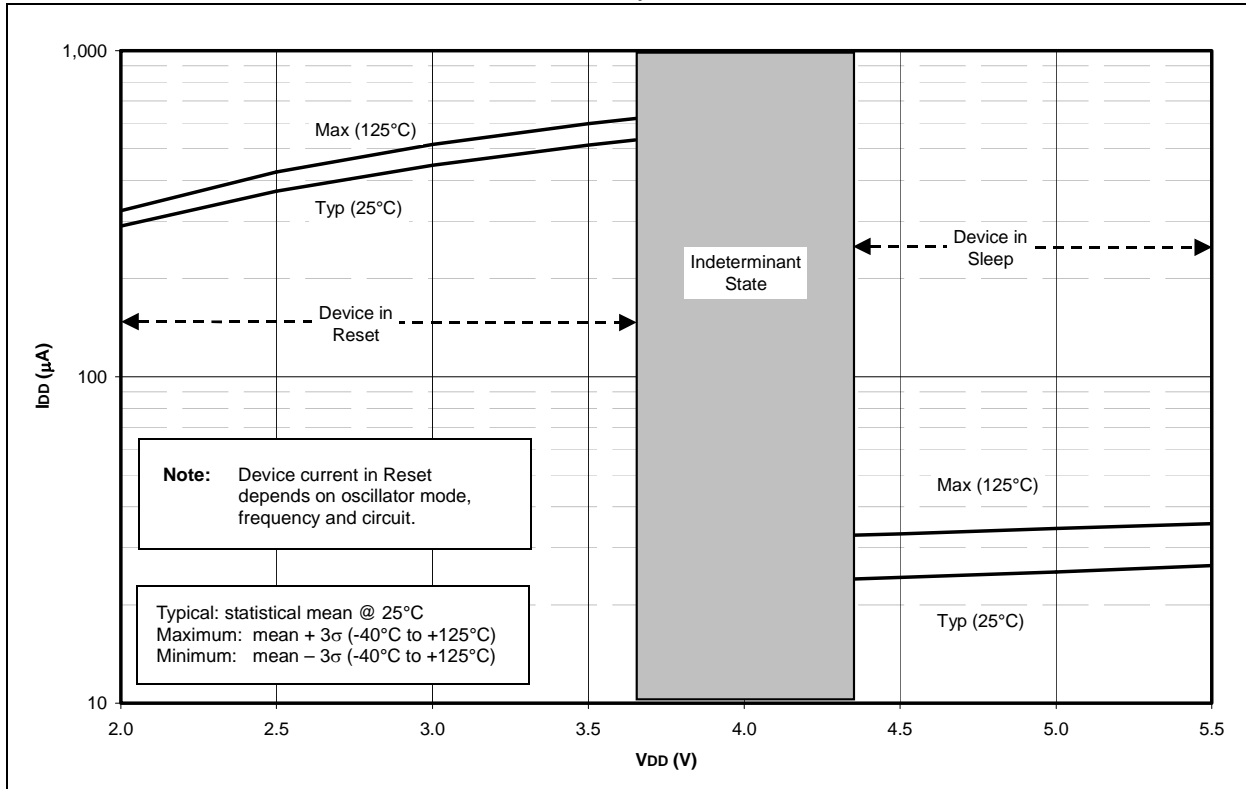
PIC16F87/88

18-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

VIEW C
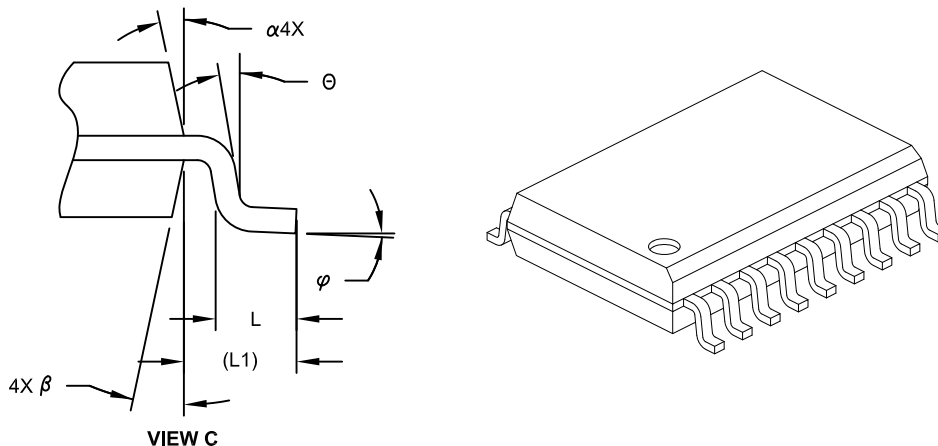
	Units	MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		18	
Pitch	e		1.27 BSC	
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E		10.30 BSC	
Molded Package Width	E1		7.50 BSC	
Overall Length	D		11.55 BSC	
Chamfer (Optional)	h	0.25	-	0.75
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. C04-051C Sheet 2 of 2

28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN]
with 0.55 mm Contact Length

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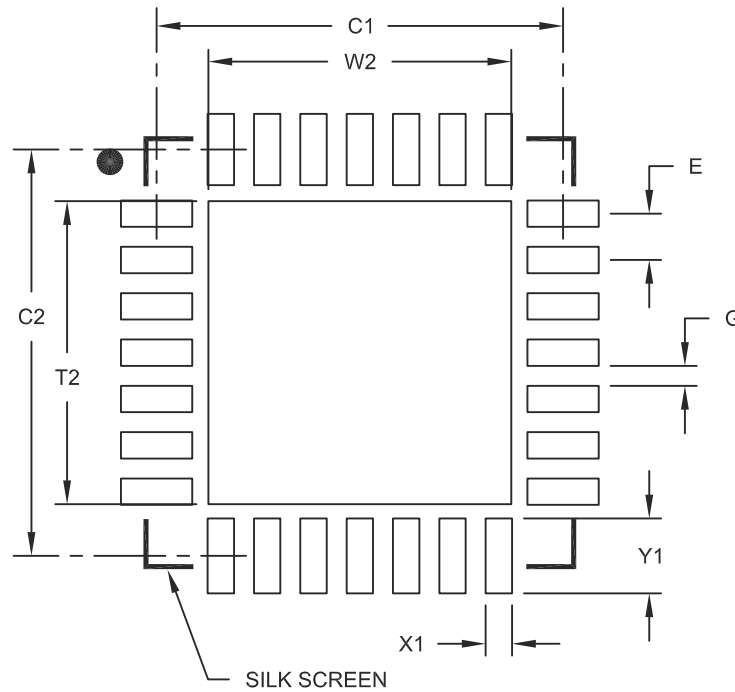
RECOMMENDED LAND PATTERN

	Units		MILLIMETERS	
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Optional Center Pad Width	W2			4.25
Optional Center Pad Length	T2			4.25
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.37
Contact Pad Length (X28)	Y1			1.00
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2105A