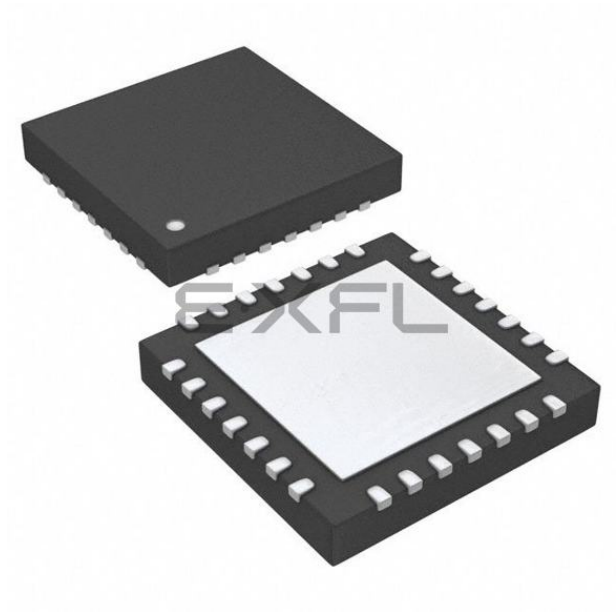


Welcome to E-XFL.COM

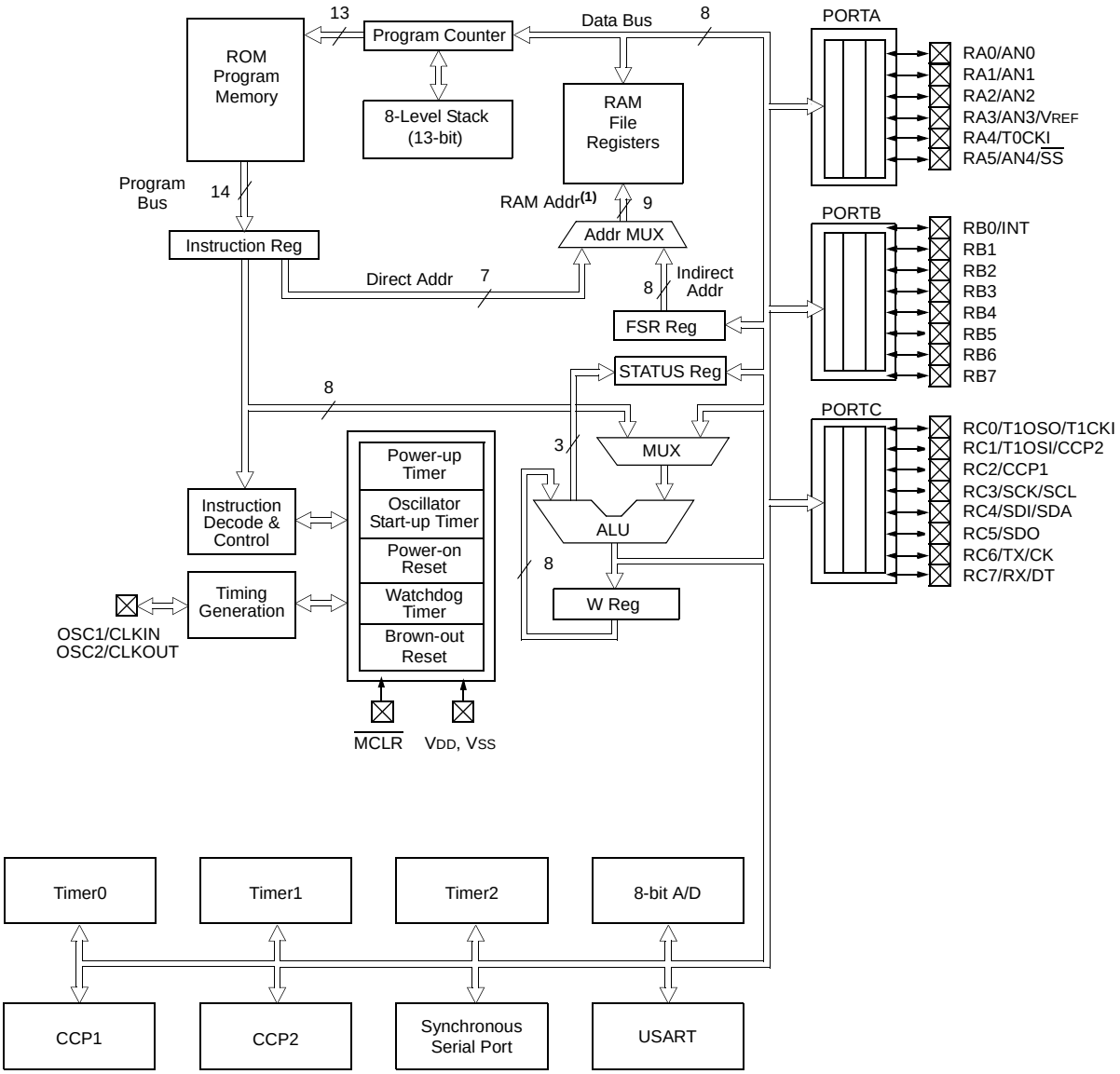
What is "Embedded - Microcontrollers"?

"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	Obsolete
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	22
Program Memory Size	14KB (8K x 14)
Program Memory Type	ROM
EEPROM Size	-
RAM Size	368 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 5x8b
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/pic16cr76-i-ml

FIGURE 1-1: PIC16CR73 AND PIC16CR76 BLOCK DIAGRAM



Device	Program ROM	Data Memory
PIC16CR73	4K	192 Bytes
PIC16CR76	8K	368 Bytes

Note 1: Higher order bits are from the STATUS register.

2.2.2.4 PIE1 Register

The PIE1 register contains the individual enable bits for the peripheral interrupts.

Note:	Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt.

REGISTER 2-4: PIE1: (ADDRESS 8Ch)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 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

bit 7 **PSPIE(1):** Parallel Slave Port Read/Write Interrupt Enable bit
1 = Enables the PSP read/write interrupt
0 = Disables the PSP read/write interrupt

bit 6 **ADIE**: A/D Converter Interrupt Enable bit
1 = Enables the A/D converter interrupt
0 = Disables the A/D converter interrupt

bit 5 **RCIE**: USART Receive Interrupt Enable bit
1 = Enables the USART receive interrupt
0 = Disables the USART receive interrupt

bit 4 **TXIE**: USART Transmit Interrupt Enable bit
1 = Enables the USART transmit interrupt
0 = Disables the USART transmit interrupt

bit 3 **SSPIE**: Synchronous Serial Port Interrupt Enable bit
1 = Enables the SSP interrupt
0 = Disables the SSP interrupt

bit 2 **CCP1IE**: CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 1 **TMR2IE**: TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt

bit 0 **TMR1IE**: TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

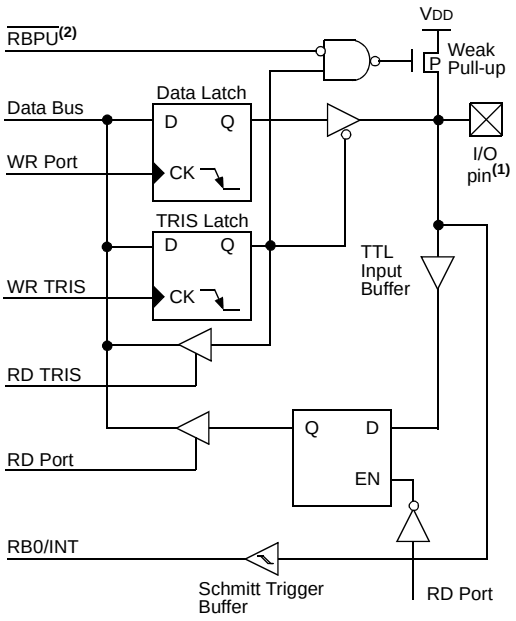
Note 1: PSPIE is reserved on 28-pin devices; always maintain this bit clear.

4.2 PORTB and the TRISB Register

PORTB is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a High-Impendance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., put the contents of the output latch on the selected pin).

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit RBPU (OPTION_REG<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

FIGURE 4-3: BLOCK DIAGRAM OF RB3:RB0 PINS



Note 1: I/O pins have diode protection to VDD and VSS.

2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION_REG<7>).

Four of the PORTB pins (RB7:RB4) have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt-on-change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are ORed together to generate the RB Port Change Interrupt with flag bit RBIF (INTCON<0>).

This interrupt can wake the device from Sleep. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

a) Any read or write of PORTB. This will end the mismatch condition.

b) Clear flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.

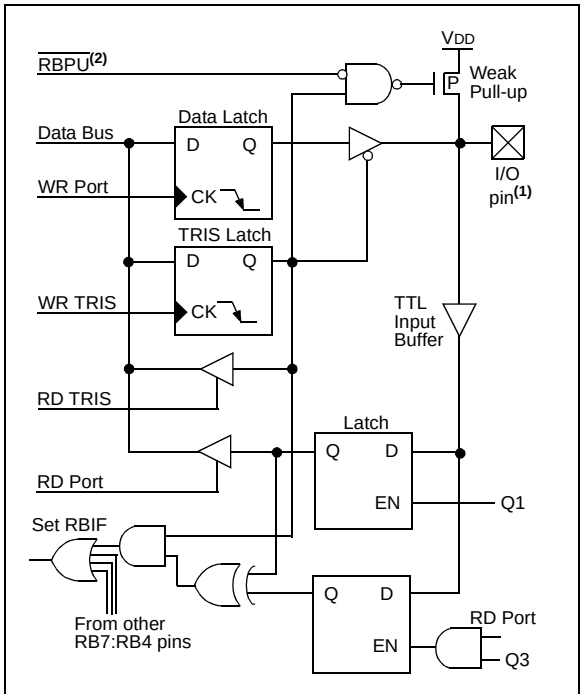
The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

This interrupt on mismatch feature, together with software configureable pull-ups on these four pins, allow easy interface to a keypad and make it possible for wake-up on key depression. Refer to the Embedded Control Handbook, "*Implementing Wake-up on Key Stroke*" (AN552).

RB0/INT is an external interrupt input pin and is configured using the INTEDG bit (OPTION_REG<6>).

RB0/INT is discussed in detail in **Section 12.11.1 "INT Interrupt"**.

FIGURE 4-4: BLOCK DIAGRAM OF RB7:RB4 PINS



Note 1: I/O pins have diode protection to VDD and VSS.

2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION_REG<7>).

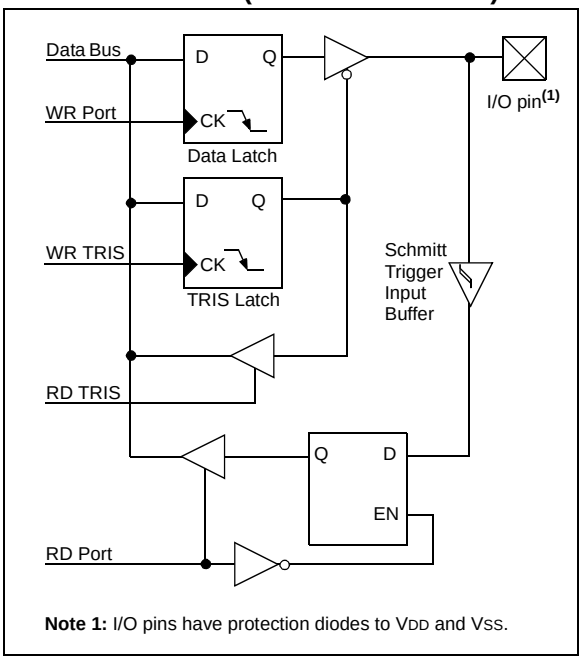
4.4 PORTD and TRISD Registers

This section is not applicable to the PIC16CR73 or PIC16CR76.

PORTD is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configureable as an input or output.

PORTD can be configured as an 8-bit wide microprocessor port (Parallel Slave Port) by setting control bit PSPMODE (TRISE<4>). In this mode, the input buffers are TTL.

FIGURE 4-6: PORTD BLOCK DIAGRAM (IN I/O PORT MODE)



Note 1: I/O pins have protection diodes to VDD and VSS.

TABLE 4-7: PORTD FUNCTIONS

Name	Bit#	Buffer Type	Function
RD0/PSP0	bit 0	ST/TTL(1)	Input/output port pin or parallel slave port bit 0
RD1/PSP1	bit 1	ST/TTL(1)	Input/output port pin or parallel slave port bit 1
RD2/PSP2	bit 2	ST/TTL(1)	Input/output port pin or parallel slave port bit 2
RD3/PSP3	bit 3	ST/TTL(1)	Input/output port pin or parallel slave port bit 3
RD4/PSP4	bit 4	ST/TTL(1)	Input/output port pin or parallel slave port bit 4
RD5/PSP5	bit 5	ST/TTL(1)	Input/output port pin or parallel slave port bit 5
RD6/PSP6	bit 6	ST/TTL(1)	Input/output port pin or parallel slave port bit 6
RD7/PSP7	bit 7	ST/TTL(1)	Input/output port pin or parallel slave port bit

Legend: ST = Schmitt Trigger input, TTL = TTL input

Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port mode.

TABLE 4-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

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
08h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	uuuu uuuu
88h	TRISD	PORTD Data Direction Register								1111 1111	1111 1111
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction bits			0000 -111	0000 -111

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

4.6 Parallel Slave Port

The Parallel Slave Port (PSP) is not implemented on the PIC16CR73 or PIC16CR76.

PORTD operates as an 8-bit wide Parallel Slave Port, or microprocessor port, when control bit PSPMODE (TRISE<4>) is set. In Slave mode, it is asynchronously readable and writable by an external system using the read control input pin RE0/$\overline{RD}$, the write control input pin RE1/$\overline{WR}$, and the Chip Select control input pin RE2/$\overline{CS}$.

The PSP can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an 8-bit latch. Setting bit PSPMODE enables port pin RE0/$\overline{RD}$ to be the $\overline{RD}$ input, RE1/$\overline{WR}$ to be the $\overline{WR}$ input and RE2/$\overline{CS}$ to be the $\overline{CS}$ (Chip Select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (i.e., set). The A/D port Configuration bits PCFG3:PCFG0 (ADCON1<3:0>) must be set to configure pins RE2:RE0 as digital I/O.

There are actually two 8-bit latches, one for data output (external reads) and one for data input (external writes). The firmware writes 8-bit data to the PORTD output data latch and reads data from the PORTD input data latch (note that they have the same address). In this mode, the TRISD register is ignored, since the external device is controlling the direction of data flow.

An external write to the PSP occurs when the $\overline{CS}$ and $\overline{WR}$ lines are both detected low. Firmware can read the actual data on the PORTD pins during this time. When either the $\overline{CS}$ or $\overline{WR}$ lines become high (level triggered), the data on the PORTD pins is latched, and the Input Buffer Full (IBF) status flag bit (TRISE<7>) and interrupt flag bit PSPIF (PIR1<7>) are set on the Q4 clock cycle, following the next Q2 cycle to signal the write is complete (Figure 4-9). Firmware clears the IBF flag by reading the latched PORTD data and clears the PSPIF bit.

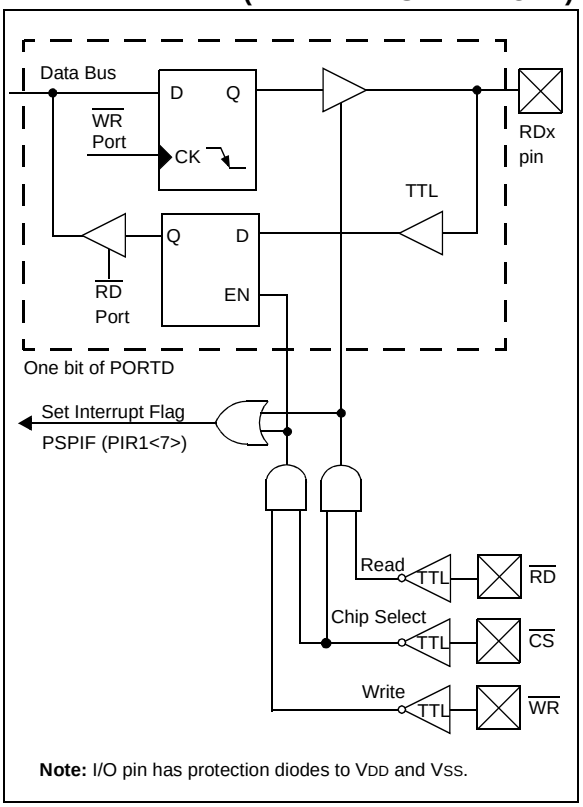
The Input Buffer Overflow (IBOV) status flag bit (TRISE<5>) is set if an external write to the PSP occurs while the IBF flag is set from a previous external write. The previous PORTD data is overwritten with the new data. IBOV is cleared by reading PORTD and clearing IBOV.

A read from the PSP occurs when both the $\overline{CS}$ and $\overline{RD}$ lines are detected low. The data in the PORTD output latch is output to the PORTD pins. The Output Buffer Full (OBF) status flag bit (TRISE<6>) is cleared immediately (Figure 4-10), indicating that the PORTD latch is being read, or has been read by the external bus. If firmware writes new data to the output latch during this time, it is immediately output to the PORTD pins, but OBF will remain cleared.

When either the $\overline{CS}$ or $\overline{RD}$ pins are detected high, the PORTD outputs are disabled, and the interrupt flag bit PSPIF is set on the Q4 clock cycle following the next Q2 cycle, indicating that the read is complete. OBF remains low until firmware writes new data to PORTD.

When not in PSP mode, the IBF and OBF bits are held clear. Flag bit IBOV remains unchanged. The PSPIF bit must be cleared by the user in firmware; the interrupt can be disabled by clearing the interrupt enable bit PSPIE (PIE1<7>).

FIGURE 4-8: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)



Note: I/O pin has protection diodes to V_{DD} and V_{SS}.

5.3 Prescaler

There is only one prescaler available on the microcontroller; it is shared exclusively between the Timer0 module and the Watchdog Timer. The usage of the prescaler is also mutually exclusive: that is, a prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer, and vice versa. This prescaler is not readable or writable (see Figure 5-1).

The PSA and PS2:PS0 bits (OPTION_REG<3:0>) determine the prescaler assignment and prescale ratio. Examples of code for assigning the prescaler assignment are shown in Example 5-1 and Example 5-2. Note that when the prescaler is being assigned to the WDT with ratios other than 1:1, lines 2 and 3 (highlighted) are optional. If a prescale ratio of 1:1 is used, however, these lines must be used to set a temporary value. The final 1:1 value is then set in lines 10 and 11 (highlighted). (Line numbers are included in the example for illustrative purposes only, and are not part of the actual code.)

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.

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

EXAMPLE 5-1: CHANGING THE PRESCALER ASSIGNMENT FROM TIMER0 TO WDT

```
1)  BSF     STATUS, RP0     ; Bank1
2)  MOVLW   b'xx0x0xxx'     ; Select clock source and prescale value of
3)  MOVWF   OPTION_REG      ; other than 1:1
4)  BCF     STATUS, RP0     ; Bank0
5)  CLRF    TMR0            ; Clear TMR0 and prescaler
6)  BSF     STATUS, RP1     ; Bank1
7)  MOVLW   b'xxxx1xxx'     ; Select WDT, do not change prescale value
8)  MOVWF   OPTION_REG
9)  CLRWDT                  ; Clears WDT and prescaler
10) MOVLW   b'xxxx1xxx'     ; Select new prescale value and WDT
11) MOVWF   OPTION_REG
12) BCF     STATUS, RP0     ; Bank0
```

EXAMPLE 5-2: CHANGING THE PRESCALER ASSIGNMENT FROM WDT TO TIMER0

```
CLRWDT                  ; Clear WDT and prescaler
BSF     STATUS, RP0     ; Bank1
MOVLW   b'xxxx0xxx'     ; Select TMR0, new prescale
MOVWF   OPTION_REG      ; value and clock source
BCF     STATUS, RP0     ; Bank0
```

TABLE 5-1: REGISTERS ASSOCIATED WITH TIMER0

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
01h,101h	TMR0	Timer0 Module Register								xxxx xxxx	uuuu uuuu
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
81h,181h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

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

NOTES:

6.4 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{T1SYNC}}$ (T1CON<2>) is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. The timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (**Section 6.4.1 "Reading and writing Timer1 in asynchronous counter mode"**).

In Asynchronous Counter mode, Timer1 cannot be used as a time base for capture or compare operations.

6.4.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L, while the timer is running from an external asynchronous clock, will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the timer register.

Reading the 16-bit value requires some care. The example code provided in Example 6-1 and Example 6-2 demonstrates how to write to and read Timer1 while it is running in Asynchronous mode.

EXAMPLE 6-1: WRITING A 16-BIT FREE-RUNNING TIMER

```
; All interrupts are disabled
CLRF    TMR1L       ; Clear Low byte, Ensures no rollover into TMR1H
MOVLW   HI_BYTE     ; Value to load into TMR1H
MOVWF   TMR1H, F    ; Write High byte
MOVLW   LO_BYTE     ; Value to load into TMR1L
MOVWF   TMR1H, F    ; Write Low byte
; Re-enable the Interrupt (if required)
CONTINUE            ; Continue with your code
```

EXAMPLE 6-2: READING A 16-BIT FREE-RUNNING TIMER

```
; All interrupts are disabled
MOVF    TMR1H, W    ; Read high byte
MOVWF   TMPH
MOVF    TMR1L, W    ; Read low byte
MOVWF   TMPL
MOVF    TMR1H, W    ; Read high byte
SUBWF   TMPH,  W    ; Sub 1st read with 2nd read
BTFSC   STATUS,Z    ; Is result = 0
GOTO    CONTINUE    ; Good 16-bit read
; TMR1L may have rolled over between the read of the high and low bytes.
; Reading the high and low bytes now will read a good value.
MOVF    TMR1H, W    ; Read high byte
MOVWF   TMPH
MOVF    TMR1L, W    ; Read low byte
MOVWF   TMPL        ; Re-enable the Interrupt (if required)
CONTINUE            ; Continue with your code
```

9.0 SYNCHRONOUS SERIAL PORT (SSP) MODULE

9.1 SSP Module Overview

The Synchronous Serial Port (SSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be Serial EEPROMs, shift registers, display drivers, A/D converters, etc. The SSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C)

An overview of I^2C operations and additional information on the SSP module can be found in the "*PIC® Mid-Range MCU Family Reference Manual*" (DS33023).

Refer to Application Note AN578, "*Use of the SSP Module in the I^2C™ Multi-Master Environment*" (DS00578).

9.2 SPI Mode

This section contains register definitions and operational characteristics of the SPI module. Additional information on the SPI module can be found in the "*PIC® Mid-Range MCU Family Reference Manual*" (DS33023).

SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO) RC5/SDO
- Serial Data In (SDI) RC4/SDI/SDA
- Serial Clock (SCK) RC3/SCK/SCL

Additionally, a fourth pin may be used when in a Slave mode of operation:

- Slave Select ($\overline{SS}$) RA5/$\overline{SS}$/AN4

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits in the SSPCON register (SSPCON<5:0>) and SSPSTAT<7:6>. These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Clock edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

TABLE 10-3: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)

BAUD RATE	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	BAUD	% ERROR	SPBRG VALUE (DECIMAL)	BAUD	% ERROR	SPBRG VALUE (DECIMAL)	BAUD	% ERROR	SPBRG VALUE (DECIMAL)
1200	1,221	1.73%	255	1,202	0.16%	207	1,202	0.16%	129
2400	2,404	0.16%	129	2,404	0.16%	103	2,404	0.16%	64
9600	9,470	-1.36%	32	9,615	0.16%	25	9,766	1.73%	15
19,200	19,531	1.73%	15	19,231	0.16%	12	19,531	1.73%	7
38,400	39,063	1.73%	7	35,714	-6.99%	6	39,063	1.73%	3
57,600	62,500	8.51%	4	62,500	8.51%	3	52,083	-9.58%	2
76,800	78,125	1.73%	3	83,333	8.51%	2	78,125	1.73%	1
96,000	104,167	8.51%	2	83,333	-13.19%	2	78,125	-18.62%	1
115,200	104,167	-9.58%	2	125,000	8.51%	1	78,125	-32.18%	1
250,000	312,500	25.00%	0	250,000	0.00%	0	156,250	-37.50%	0

BAUD RATE	Fosc = 4 MHz			Fosc = 3.6864 MHz			Fosc = 3.579545 MHz		
	BAUD	% ERROR	SPBRG VALUE (DECIMAL)	BAUD	% ERROR	SPBRG VALUE (DECIMAL)	BAUD	% ERROR	SPBRG VALUE (DECIMAL)
300	300	0.16%	207	300	0.00%	191	301	0.23%	185
1200	1,202	0.16%	51	1,200	0.00%	47	1,190	-0.83%	46
2400	2,404	0.16%	25	2,400	0.00%	23	2,432	1.32%	22
9600	8,929	-6.99%	6	9,600	0.00%	5	9,322	-2.90%	5
19,200	20,833	8.51%	2	19,200	0.00%	2	18,643	-2.90%	2
38,400	31,250	-18.62%	1	28,800	-25.00%	1	27,965	-27.17%	1
57,600	62,500	8.51%	0	57,600	0.00%	0	55,930	-2.90%	0
76,800	62,500	-18.62%	0	—	—	—	—	—	—

TABLE 10-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)

BAUD RATE	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	BAUD	% ERROR	SPBRG VALUE (DECIMAL)	BAUD	% ERROR	SPBRG VALUE (DECIMAL)	BAUD	% ERROR	SPBRG VALUE (DECIMAL)
2400	—	—	—	—	—	—	2,441	1.73%	255
9600	9,615	0.16%	129	9,615	0.16%	103	9,615	0.16%	64
19,200	19,231	0.16%	64	19,231	0.16%	51	18,939	-1.36%	32
38,400	37,879	-1.36%	32	38,462	0.16%	25	39,063	1.73%	15
57,600	56,818	-1.36%	21	58,824	2.12%	16	56,818	-1.36%	10
76,800	78,125	1.73%	15	76,923	0.16%	12	78,125	1.73%	7
96,000	96,154	0.16%	12	100,000	4.17%	9	89,286	-6.99%	6
115,200	113,636	-1.36%	10	111,111	-3.55%	8	125,000	8.51%	4
250,000	250,000	0.00%	4	250,000	0.00%	3	208,333	-16.67%	2
300,000	312,500	4.17%	3	333,333	11.11%	2	312,500	4.17%	1

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz			Fosc = 3.579545 MHz		
	BAUD	% ERROR	SPBRG VALUE (DECIMAL)	BAUD	% ERROR	SPBRG VALUE (DECIMAL)	BAUD	% ERROR	SPBRG VALUE (DECIMAL)
1200	1,202	0.16%	207	1,200	0.00%	191	1,203	0.23%	185
2400	2,404	0.16%	103	2,400	0.00%	95	2,406	0.23%	92
9600	9,615	0.16%	25	9,600	0.00%	23	9,727	1.32%	22
19,200	19,231	0.16%	12	19,200	0.00%	11	18,643	-2.90%	11
38,400	35,714	-6.99%	6	38,400	0.00%	5	37,287	-2.90%	5
57,600	62,500	8.51%	3	57,600	0.00%	3	55,930	-2.90%	3
76,800	83,333	8.51%	2	76,800	0.00%	2	74,574	-2.90%	2
96,000	83,333	-13.19%	2	115,200	20.00%	1	111,861	16.52%	1
115,200	125,000	8.51%	1	115,200	0.00%	1	111,861	-2.90%	1
250,000	250,000	0.00%	0	230,400	-7.84%	0	223,722	-10.51%	0

TABLE 10-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE 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	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	USART Transmit Data Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 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 slave transmission.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16CR73/76 devices; always maintain these bits clear.

10.4.2 USART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of the Sleep mode. Bit SREN is a "don't care" in Slave mode.

If receive is enabled by setting bit CREN prior to the SLEEP instruction, then a word may be received during Sleep. On completely receiving the word, the RSR register will transfer the data to the RCREG register and if enable bit RCIE bit is set, the interrupt generated will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

Follow these steps when setting up a Synchronous Slave Reception:

1. Enable the synchronous master serial port by setting bits SYNC and SPEN and clearing bit CSRC.
2. If interrupts are desired, set enable bit RCIE.
3. If 9-bit reception is desired, set bit RX9.
4. To enable reception, set enable bit CREN.
5. Flag bit RCIF will be set when reception is complete and an interrupt will be generated, if enable bit RCIE was set.
6. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading the RCREG register.
8. If any error occurred, clear the error by clearing bit CREN.
9. If using interrupts, ensure that GIE and PEIE in the INTCON register are set.

TABLE 10-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE 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	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	USART Receive Data Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 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 slave reception.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16CR73/76 devices, always maintain these bits clear.

12.10 Power Control/Status Register (PCON)

The Power Control/Status Register, PCON, has two bits to indicate the type of Reset that last occurred.

Bit 0 is Brown-out Reset Status bit, $\overline{BOR}$. Bit $\overline{BOR}$ is unknown on a Power-on Reset. It must then be set by the user and checked on subsequent Resets to see if bit $\overline{BOR}$ cleared, indicating a Brown-out Reset occurred. When the Brown-out Reset is disabled, the state of the $\overline{BOR}$ bit is unpredictable.

Bit 1 is $\overline{POR}$ (Power-on Reset Status bit). It is cleared on a Power-on Reset and unaffected otherwise. The user must set this bit following a Power-on Reset.

TABLE 12-3: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up		Brown-out	Wake-up from Sleep
	$\overline{PWRTE}$ = 0	$\overline{PWRTE}$ = 1		
XT, HS, LP	72 ms + 1024 TOSC	1024 TOSC	72 ms + 1024 TOSC	1024 TOSC
RC	72 ms	—	72 ms	—

TABLE 12-4: STATUS BITS AND THEIR SIGNIFICANCE

$\overline{POR}$ (PCON<1>)	$\overline{BOR}$ (PCON<0>)	$\overline{TO}$ (STATUS<4>)	$\overline{PD}$ (STATUS<3>)	Significance
0	x	1	1	Power-on Reset
0	x	0	x	Illegal, $\overline{TO}$ is set on $\overline{POR}$
0	x	x	0	Illegal, $\overline{PD}$ is set on $\overline{POR}$
1	0	1	1	Brown-out Reset
1	1	0	1	WDT Reset
1	1	0	0	WDT Wake-up
1	1	u	u	$\overline{MCLR}$ Reset during normal operation
1	1	1	0	$\overline{MCLR}$ Reset during Sleep or interrupt wake-up from Sleep

TABLE 12-5: RESET CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	000h	0001 1xxx	---- --0x
$\overline{MCLR}$ Reset during normal operation	000h	000u uuuu	---- --uu
$\overline{MCLR}$ Reset during Sleep	000h	0001 0uuu	---- --uu
WDT Reset	000h	0000 1uuu	---- --uu
WDT Wake-up	PC + 1	uuu0 0uuu	---- --uu
Brown-out Reset	000h	0001 1uuu	---- --u0
Interrupt wake-up from Sleep	PC + 1(1)	uuu1 0uuu	---- --uu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0'

Note 1: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

TABLE 12-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS

Register	Devices				Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Reset, WDT Reset	Wake-up via WDT or Interrupt
W	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	73	74	76	77	N/A	N/A	N/A
TMR0	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	73	74	76	77	0000h	0000h	PC + 1(2)
STATUS	73	74	76	77	0001 1xxx	000q quuu(3)	uuuq quuu(3)
FSR	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	73	74	76	77	--0x 0000	--0u 0000	--uu uuuu
PORTB	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTD	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTE	73	74	76	77	---- -xxx	---- -uuu	---- -uuu
PCLATH	73	74	76	77	---0 0000	---0 0000	---u uuuu
INTCON	73	74	76	77	0000 000x	0000 000u	uuuu uuuu(1)
PIR1	73	74	76	77	r000 0000	r000 0000	ruuu uuuu(1)
	73	74	76	77	0000 0000	0000 0000	uuuu uuuu(1)
PIR2	73	74	76	77	---- ---0	---- ---0	---- ---u(1)
TMR1L	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	73	74	76	77	--00 0000	--uu uuuu	--uu uuuu
TMR2	73	74	76	77	0000 0000	0000 0000	uuuu uuuu
T2CON	73	74	76	77	-000 0000	-000 0000	-uuu uuuu
SSPBUF	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPCON	73	74	76	77	0000 0000	0000 0000	uuuu uuuu
CCPR1L	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	73	74	76	77	--00 0000	--00 0000	--uu uuuu
RCSTA	73	74	76	77	0000 -00x	0000 -00x	uuuu -uuu
TXREG	73	74	76	77	0000 0000	0000 0000	uuuu uuuu
RCREG	73	74	76	77	0000 0000	0000 0000	uuuu uuuu
CCPR2L	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2H	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	73	74	76	77	0000 0000	0000 0000	uuuu uuuu
ADRES	73	74	76	77	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	73	74	76	77	0000 00-0	0000 00-0	uuuu uu-u
OPTION_REG	73	74	76	77	1111 1111	1111 1111	uuuu uuuu
TRISA	73	74	76	77	--11 1111	--11 1111	--uu uuuu
TRISB	73	74	76	77	1111 1111	1111 1111	uuuu uuuu
TRISC	73	74	76	77	1111 1111	1111 1111	uuuu uuuu
TRISD	73	74	76	77	1111 1111	1111 1111	uuuu uuuu
TRISE	73	74	76	77	0000 -111	0000 -111	uuuu -uuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition, r = reserved, maintain clear

Note 1: One or more bits in INTCON, PIR1 and/or PIR2 will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

3: See Table 12-5 for Reset value for specific condition.

MOVF	Move f
Syntax:	[*label*] MOVF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are moved to a destination dependant upon the status of 'd'. If 'd' = 0, destination is W register. If 'd' = 1, the destination is file register 'f' itself. 'd' = 1 is useful to test a file register, since status flag Z is affected.

MOVLW	Move Literal to W
Syntax:	[*label*] MOVLW k
Operands:	$0 \le k \le 255$
Operation:	k → (W)
Status Affected:	None
Description:	The eight-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.

MOVWF	Move W to f
Syntax:	[*label*] MOVWF f
Operands:	$0 \le f \le 127$
Operation:	(W) → (f)
Status Affected:	None
Description:	Move data from W register to register 'f'.

NOP	No Operation
Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.

RETFIE	Return from Interrupt
Syntax:	[*label*] RETFIE
Operands:	None
Operation:	TOS → PC, 1 → GIE
Status Affected:	None

RETLW	Return with Literal in W
Syntax:	[*label*] RETLW k
Operands:	$0 \le k \le 255$
Operation:	k → (W); TOS → PC
Status Affected:	None
Description:	Subtract (2's complement method) W register from register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

14.11 PICSTART Plus Development Programmer

The PICSTART Plus Development Programmer is an easy-to-use, low-cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. The PICSTART Plus Development Programmer supports most PIC devices in DIP packages up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus Development Programmer is CE compliant.

14.12 PICkit 2 Development Programmer

The PICkit™ 2 Development Programmer is a low-cost programmer and selected Flash device debugger with an easy-to-use interface for programming many of Microchip's baseline, mid-range and PIC18F families of Flash memory microcontrollers. The PICkit 2 Starter Kit includes a prototyping development board, twelve sequential lessons, software and HI-TECH's PICC™ Lite C compiler, and is designed to help get up to speed quickly using PIC® microcontrollers. The kit provides everything needed to program, evaluate and develop applications using Microchip's powerful, mid-range Flash memory family of microcontrollers.

14.13 Demonstration, Development and Evaluation Boards

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

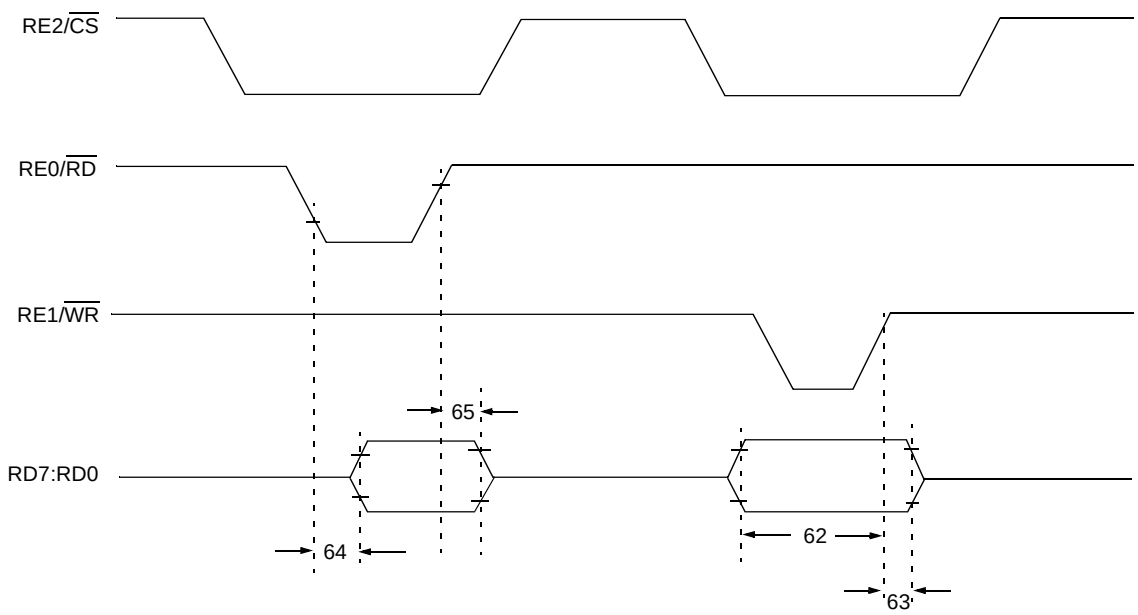
The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart® battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Check the Microchip web page (www.microchip.com) and the latest *"Product Selector Guide"* (DS00148) for the complete list of demonstration, development and evaluation kits.

FIGURE 15-9: PARALLEL SLAVE PORT TIMING (PIC16CR74/77 DEVICES ONLY)



Note: Refer to Figure 15-2 for load conditions.

TABLE 15-6: PARALLEL SLAVE PORT REQUIREMENTS (PIC16CR74/77 DEVICES ONLY)

Parameter No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions
62	TdtV2wrH	Data in valid before $\overline{WR}\uparrow$ or $\overline{CS}\uparrow$ (setup time)		20 25	— —	— —	ns ns	 Extended range only
63*	TwrH2dtI	$\overline{WR}\uparrow$ or $\overline{CS}\uparrow$ to data in invalid (hold time)	Standard(**5V**)	20	—	—	ns	
			Extended(**3V**)	35	—	—	ns	
64	TrdL2dtV	$\overline{RD}\downarrow$ and $\overline{CS}\downarrow$ to data out valid		— —	— —	80 90	ns ns	 Extended range only
65	TrdH2dtI	$\overline{RD}\uparrow$ or $\overline{CS}\downarrow$ to data out invalid		10	—	30	ns	

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 15-9: I²C™ BUS DATA REQUIREMENTS

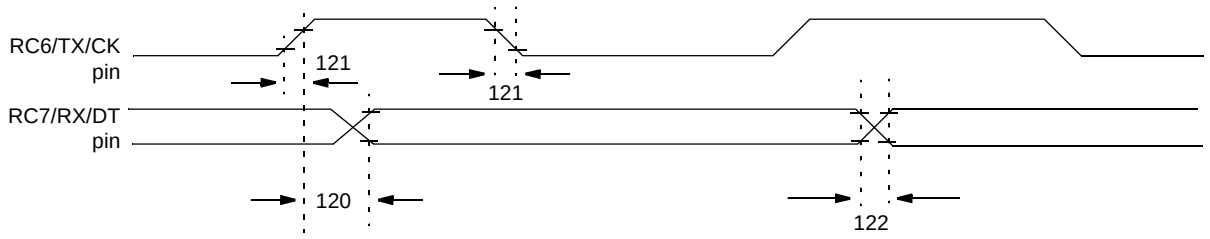
Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
100*	THIGH	Clock high time	100 kHz mode	4.0	—	µs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	µs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5TCY	—		
101*	TLOW	Clock low time	100 kHz mode	4.7	—	µs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	µs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5TCY	—		
102*	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1CB	300	ns	CB is specified to be from 10-400 pF
103*	TF	SDA and SCL fall time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1CB	300	ns	CB is specified to be from 10-400 pF
90*	TSU:STA	Start condition setup time	100 kHz mode	4.7	—	µs	Only relevant for Repeated Start condition
			400 kHz mode	0.6	—	µs	
91*	THD: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*	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	µs	
107*	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	**(Note 2)**
			400 kHz mode	100	—	ns	
92*	TSU:STO	Stop condition setup time	100 kHz mode	4.7	—	µs	
			400 kHz mode	0.6	—	µs	
109*	TAA	Output valid from clock	100 kHz mode	—	3500	ns	**(Note 1)**
			400 kHz mode	—	—	ns	
110*	TBUF	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	
	CB	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 TSU: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 TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

FIGURE 15-16: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING



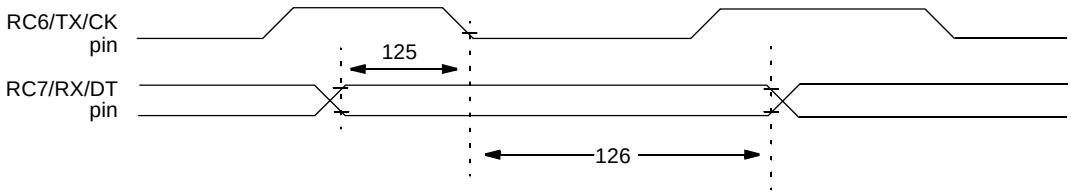
Note: Refer to Figure 15-2 for load conditions.

TABLE 15-10: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock high to data out valid	Standard(**5V**)	—	—	80	ns	
			Extended(**3V**)	—	—	100	ns	
121	Tckrf	Clock out rise time and fall time (Master mode)	Standard(**5V**)	—	—	45	ns	
			Extended(**3V**)	—	—	50	ns	
122	Tdtrf	Data out rise time and fall time	Standard(**5V**)	—	—	45	ns	
			Extended(**3V**)	—	—	50	ns	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 15-17: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



Note: Refer to Figure 15-2 for load conditions.

TABLE 15-11: USART SYNCHRONOUS RECEIVE REQUIREMENTS

Parameter No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
125	TdtV2ckL	SYNC RCV (MASTER & SLAVE) Data setup before CK↓ (DT setup time)	15	—	—	ns	
126	TckL2dtl	Data hold after CK↓ (DT hold time)	15	—	—	ns	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 16-3: TYPICAL IDD vs. FOSC OVER VDD (XT MODE)

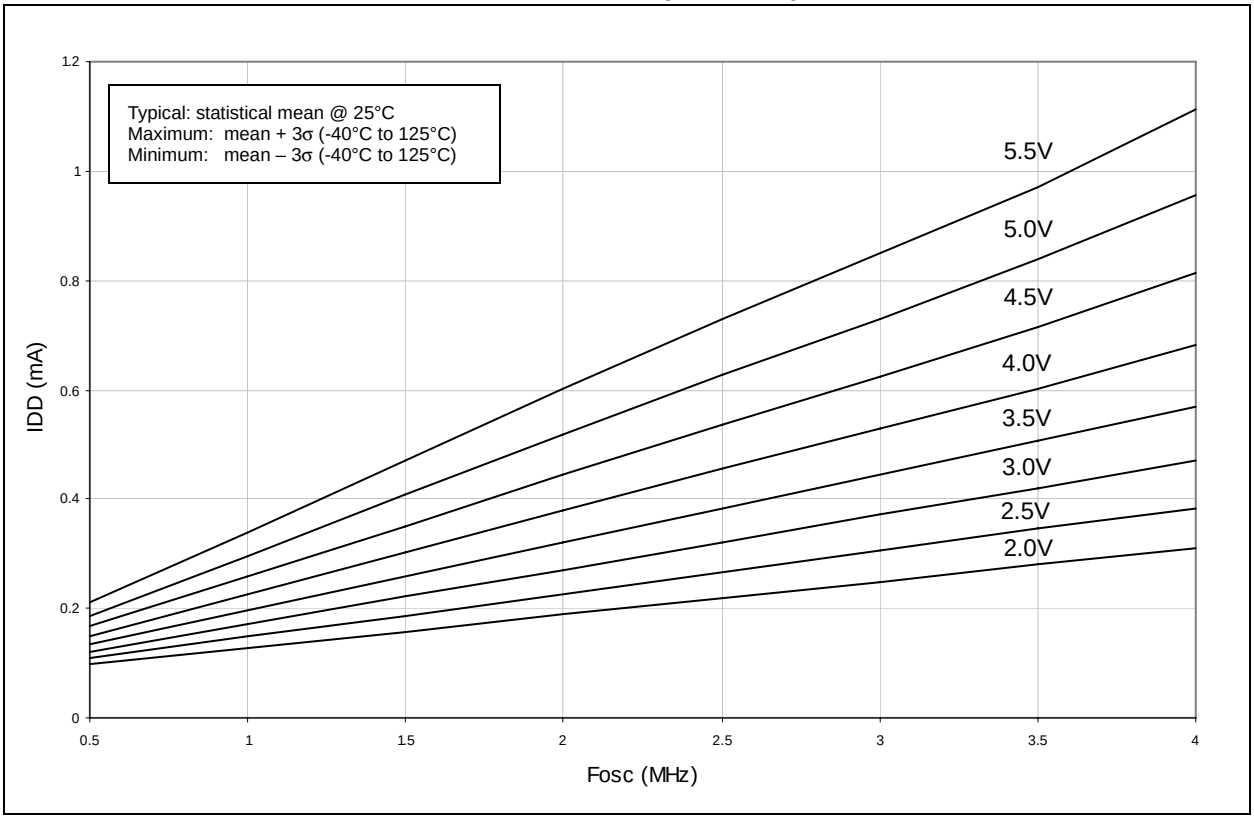
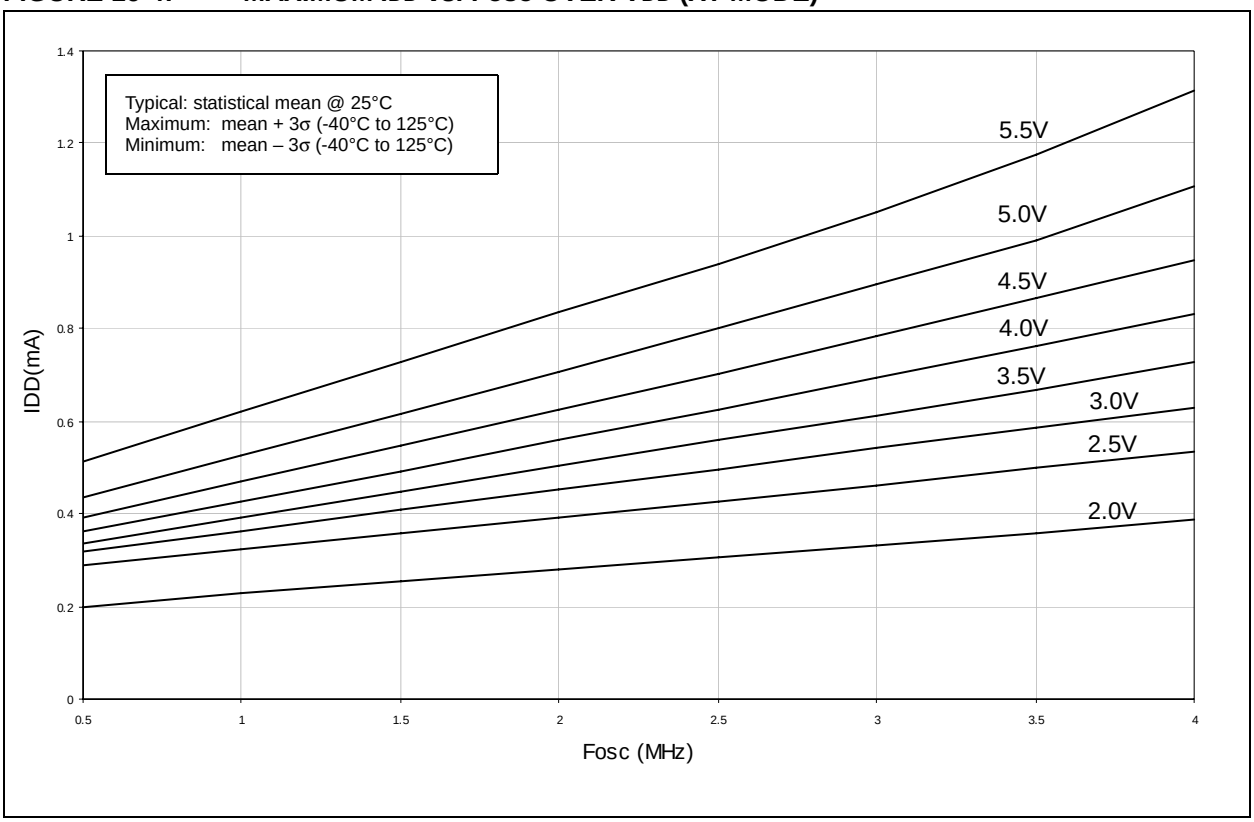


FIGURE 16-4: MAXIMUM IDD vs. FOSC OVER VDD (XT MODE)

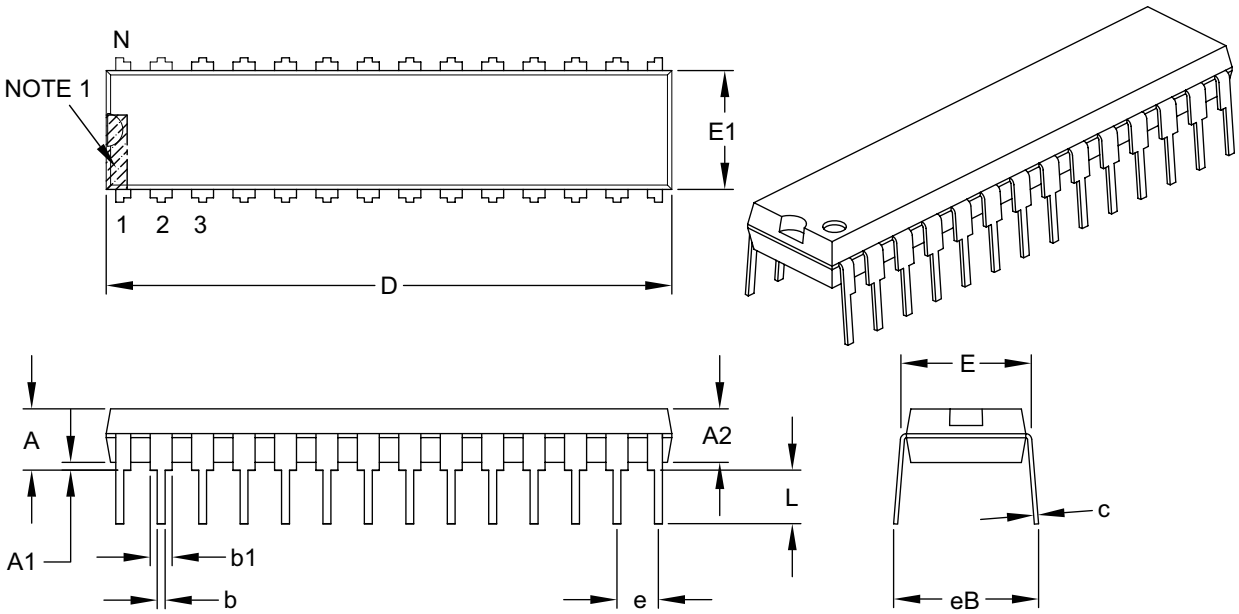


17.2 Package Details

The following sections give the technical details of the packages.

28-Lead Skinny Plastic Dual In-Line (SP) – 300 mil Body [SPDIP]

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



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-070B