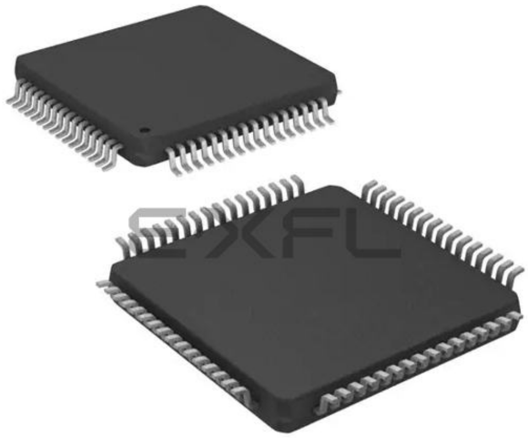
Microchip Technology - PIC18F6390T-I/PT Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, LCD, POR, PWM, WDT
Number of I/O	50
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f6390t-i-pt

TABLE 1-3: PIC18F8X90 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
				PORTD is a bidirectional I/O port.
RD0/SEG0	72			
RD0		I/O	ST	Digital I/O.
SEG0		O	Analog	SEG0 output for LCD.
RD1/SEG1	69			
RD1		I/O	ST	Digital I/O.
SEG1		O	Analog	SEG1 output for LCD.
RD2/SEG2	68			
RD2		I/O	ST	Digital I/O.
SEG2		O	Analog	SEG2 output for LCD.
RD3/SEG3	67			
RD3		I/O	ST	Digital I/O.
SEG3		O	Analog	SEG3 output for LCD.
RD4/SEG4	66			
RD4		I/O	ST	Digital I/O.
SEG4		O	Analog	SEG4 output for LCD.
RD5/SEG5	65			
RD5		I/O	ST	Digital I/O.
SEG5		O	Analog	SEG5 output for LCD.
RD6/SEG6	64			
RD6		I/O	ST	Digital I/O.
SEG6		O	Analog	SEG6 output for LCD.
RD7/SEG7	63			
RD7		I/O	ST	Digital I/O.
SEG7		O	Analog	SEG7 output for LCD.

Legend:

TTL	= TTL compatible input	CMOS	= CMOS compatible input or output
ST	= Schmitt Trigger input with CMOS levels	Analog	= Analog input
I	= Input	O	= Output
P	= Power	OD	= Open-Drain (no P diode to VDD)

Note 1: Default assignment for CCP2 when Configuration bit, CCP2MX, is set.

2: Alternate assignment for CCP2 when Configuration bit, CCP2MX, is cleared.

5.6.3 MAPPING THE ACCESS BANK IN INDEXED LITERAL OFFSET MODE

The use of Indexed Literal Offset Addressing mode effectively changes how the lower part of Access RAM (00h to 5Fh) is mapped. Rather than containing just the contents of the bottom part of Bank 0, this mode maps the contents from Bank 0 and a user-defined "window" that can be located anywhere in the data memory space. The value of FSR2 establishes the lower boundary of the addresses mapped into the window, while the upper boundary is defined by FSR2 plus 95 (5Fh). Addresses in the Access RAM above 5Fh are mapped as previously described (see **Section 5.3.2 "Access Bank"**). An example of Access Bank remapping in this addressing mode is shown in Figure 5-9.

Remapping of the Access Bank applies *only* to operations using the Indexed Literal Offset mode. Operations that use the BSR (Access RAM bit is '1') will continue to use Direct Addressing as before. Any indirect or indexed operation that explicitly uses any of the indirect file operands (including FSR2) will continue to operate as standard Indirect Addressing. Any instruction that uses the Access Bank, but includes a register address of greater than 05Fh, will use Direct Addressing and the normal Access Bank map.

5.6.4 BSR IN INDEXED LITERAL OFFSET MODE

Although the Access Bank is remapped when the extended instruction set is enabled, the operation of the BSR remains unchanged. Direct Addressing, using the BSR to select the data memory bank, operates in the same manner as previously described.

FIGURE 5-9: REMAPPING THE ACCESS BANK WITH INDEXED LITERAL OFFSET ADDRESSING

Example Situation:

`ADDWF f, d, a`

FSR2H:FSR2L = 120h

Locations in the region from the FSR2 Pointer (120h) to the pointer plus 05Fh (17Fh) are mapped to the bottom of the Access RAM (000h-05Fh).

Special File Registers at F60h through FFFh are mapped to 60h through FFh, as usual.

Bank 0 addresses below 5Fh are not available in this mode. They can still be addressed by using the BSR.

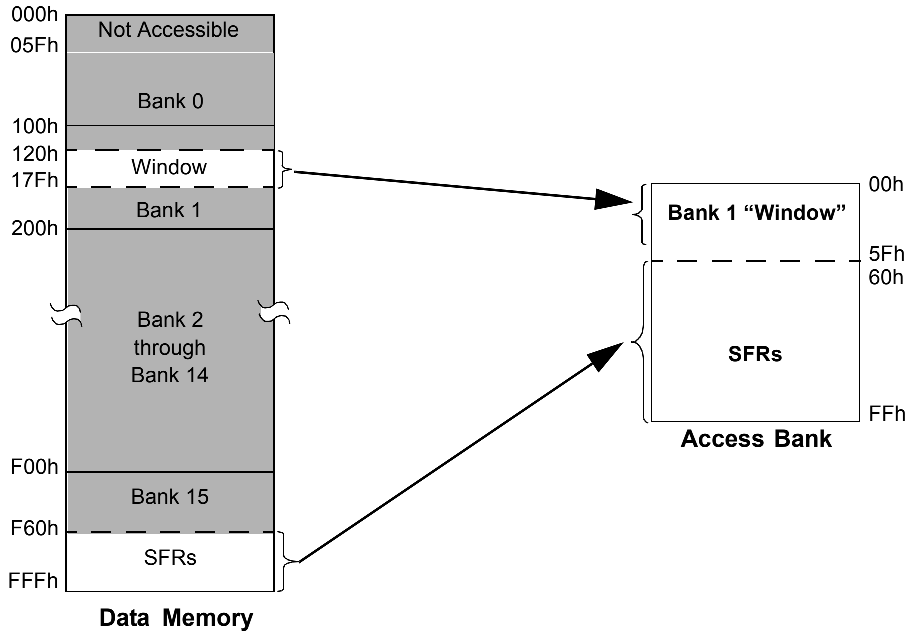


FIGURE 8-1: PIC18F6X90/8X90 INTERRUPT LOGIC

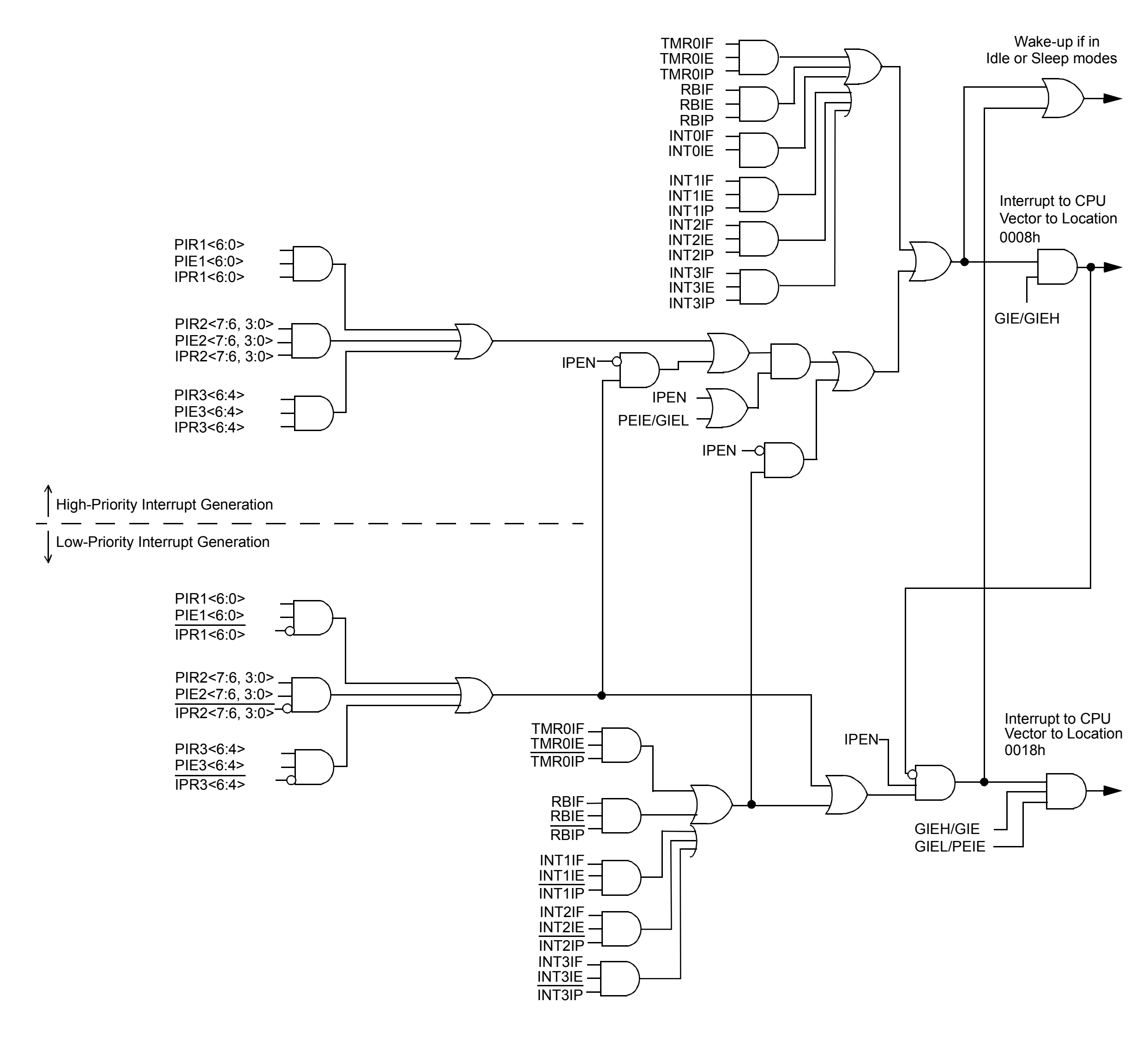


TABLE 9-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	62
LATB	LATB Data Output Register								62
TRISB	PORTB Data Direction Register								62
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	59
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	59
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	59
LCDSE1	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	64

Legend: Shaded cells are not used by PORTB.

9.4 PORTD, TRISD and LATD Registers

PORTD is an 8-bit wide, bidirectional port. The corresponding Data Direction register is TRISD. Setting a TRISD bit (= `1`) will make the corresponding PORTD pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISD bit (= `0`) will make the corresponding PORTD pin an output (i.e., put the contents of the output latch on the selected pin).

The Data Latch register (LATD) is also memory mapped. Read-modify-write operations on the LATD register read and write the latched output value for PORTD.

All pins on PORTD are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

Note: On a Power-on Reset, these pins are configured as digital inputs.

PORTD is also multiplexed with LCD segment drives controlled by the LCDSE0 register. I/O port functions are only available when the segments are disabled.

EXAMPLE 9-4: INITIALIZING PORTD

```
CLRF    PORTD   ; Initialize PORTD by
                ; clearing output
                ; data latches
CLRF    LATD    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISD   ; Set RD<3:0> as inputs
                ; RD<5:4> as outputs
                ; RD<7:6> as inputs
```

TABLE 9-7: PORTD FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	Buffer	Description
RD0/SEG0	RD0	`0`	O	DIG	LATD<0> data output; disabled when LCD segment enabled.
		`1`	I	ST	PORTD<0> data input.
	SEG0	`x`	O	ANA	Segment 0 analog output for LCD.
RD1/SEG1	RD1	`0`	O	DIG	LATD<1> data output; disabled when LCD segment enabled.
		`1`	I	ST	PORTD<1> data input.
	SEG1	`x`	O	ANA	Segment 1 analog output for LCD.
RD2/SEG2	RD2	`0`	O	DIG	LATD<2> data output; disabled when LCD segment enabled.
		`1`	I	ST	PORTD<2> data input.
	SEG2	`x`	O	ANA	Segment 2 analog output for LCD.
RD3/SEG3	RD3	`0`	O	DIG	LATD<3> data output; disabled when LCD segment enabled.
		`1`	I	ST	PORTD<3> data input.
	SEG3	`x`	O	ANA	Segment 3 analog output for LCD.
RD4/SEG4	RD4	`0`	O	DIG	LATD<4> data output; disabled when LCD segment enabled.
		`1`	I	ST	PORTD<4> data input.
	SEG4	`x`	O	ANA	Segment 4 analog output for LCD module.
RD5/SEG5	RD5	`0`	O	DIG	LATD<5> data output; disabled when LCD segment enabled.
		`1`	I	ST	PORTD<5> data input.
	SEG5	`x`	O	ANA	Segment 5 analog output for LCD.
RD6/SEG6	RD6	`0`	O	DIG	LATD<6> data output; disabled when LCD segment enabled.
		`1`	I	ST	PORTD<6> data input.
	SEG6	`x`	O	ANA	Segment 6 analog output for LCD.
RD7/SEG7	RD7	`0`	O	DIG	LATD<7> data output; disabled when LCD segment enabled.
		`1`	I	ST	PORTD<7> data input.
	SEG7	`x`	O	ANA	Segment 7 analog output for LCD.

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input, `x` = Don't care (TRIS bit does not affect port direction or is overridden for this option).

TABLE 9-13: SUMMARY OF REGISTERS ASSOCIATED WITH PORTF

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page
TRISF	PORTF Data Direction Register								62
PORTF	Read PORTF Data Latch/Write PORTF Data Latch								62
LATF	LATF Data Output Register								62
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	61
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	61
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	61
LCDSE2	SE23	SE22	SE21	SE20	SE19	SE18	SE17	SE16	64
LCDSE3	SE31	SE30	SE29	SE28	SE27	SE26	SE25	SE24	64

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used by PORTF.

TABLE 9-18: PORTJ FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	Buffer	Description
RJ0/SEG32	RJ0	0	O	DIG	LATJ<0> data output; disabled when LCD segment enabled.
		1	I	ST	PORTJ<0> data input.
	SEG32	x	O	ANA	Segment 32 analog output for LCD.
RJ1/SEG33	RJ1	0	O	DIG	LATJ<1> data output; disabled when LCD segment enabled.
		1	I	ST	PORTJ<1> data input.
	SEG33	x	O	ANA	Segment 33 analog output for LCD.
RJ2/SEG34	RJ2	0	O	DIG	LATJ<2> data output; disabled when LCD segment enabled.
		1	I	ST	PORTJ<2> data input.
	SEG34	x	O	ANA	Segment 34 analog output for LCD.
RJ3/SEG35	RJ3	0	O	DIG	LATJ<3> data output; disabled when LCD segment enabled.
		1	I	ST	PORTJ<3> data input.
	SEG35	x	O	ANA	Segment 35 analog output for LCD.
RJ4/SEG39	RJ4	0	O	DIG	LATJ<4> data output; disabled when LCD segment enabled.
		1	I	ST	PORTJ<4> data input.
	SEG39	x	O	ANA	Segment 39 analog output for LCD.
RJ5/SEG38	RJ5	0	O	DIG	LATJ<5> data output; disabled when LCD segment enabled.
		1	I	ST	PORTJ<5> data input.
	SEG38	x	O	ANA	Segment 38 analog output for LCD.
RJ6/SEG37	RJ6	0	O	DIG	LATJ<6> data output; disabled when LCD segment enabled.
		1	I	ST	PORTJ<6> data input.
	SEG37	x	O	ANA	Segment 37 analog output for LCD.
RJ7/SEG36	RJ7	0	O	DIG	LATJ<7> data output; disabled when LCD segment enabled.
		1	I	ST	PORTJ<7> data input.
	SEG36	x	O	ANA	Segment 36 analog output for LCD.

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

TABLE 9-19: SUMMARY OF REGISTERS ASSOCIATED WITH PORTJ

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page
PORTJ	Read PORTJ pin/Write PORTJ Data Latch								62
LATJ	LATJ Data Output Register								62
TRISJ	PORTJ Data Direction Register								62
LCDSE4	SE39	SE38	SE37	SE36	SE35	SE34	SE33	SE32	64

11.0 TIMER1 MODULE

The Timer1 timer/counter module incorporates these features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR1H and TMR1L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt-on-overflow
- Reset on CCP Special Event Trigger
- Device clock status flag (T1RUN)

A simplified block diagram of the Timer1 module is shown in Figure 11-1. A block diagram of the module's operation in Read/Write mode is shown in Figure 11-2.

The module incorporates its own low-power oscillator to provide an additional clocking option. The Timer1 oscillator can also be used as a low-power clock source for the microcontroller in power-managed operation.

Timer1 can also be used to provide Real-Time Clock (RTC) functionality to applications with only a minimal addition of external components and code overhead.

Timer1 is controlled through the T1CON Control register (Register 11-1). It also contains the Timer1 Oscillator Enable bit (T1OSCEN). Timer1 can be enabled or disabled by setting or clearing control bit, TMR1ON (T1CON<0>).

REGISTER 11-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
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 **RD16:** 16-Bit Read/Write Mode Enable bit
1 = Enables register read/write of TImer1 in one 16-bit operation
0 = Enables register read/write of Timer1 in two 8-bit operations

bit 6 **T1RUN:** Timer1 System Clock Status bit
1 = Device clock is derived from Timer1 oscillator
0 = Device clock is derived from another source

bit 5-4 **T1CKPS1:T1CKPS0:** Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **T1OSCEN:** Timer1 Oscillator Enable bit
1 = Timer1 oscillator is enabled
0 = Timer1 oscillator is shut off
The oscillator inverter and feedback resistor are turned off to eliminate power drain.

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 External Clock Input Synchronization Select bit
When TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit
1 = External clock from pin RC0/T1OSO/T13CKI (on the rising edge)
0 = Internal clock (FOSC/4)

bit 0 **TMR1ON:** Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1

15.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON1<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)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

The MSSP consists of a Transmit/Receive Shift register (SSPSR) and a Buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full detect bit, BF (SSPSTAT<0>), and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the Write Collision Detect bit, WCOL (SSPCON1<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. The Buffer Full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 15-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the MSSP Status register (SSPSTAT) indicates the various status conditions.

EXAMPLE 15-1: LOADING THE SSPBUF (SSPSR) REGISTER

```
LOOP    BTFSS   SSPSTAT, BF     ;Has data been received (transmit complete)?
        BRA     LOOP            ;No
        MOVF    SSPBUF, W       ;WREG reg = contents of SSPBUF
        MOVWF   RXDATA          ;Save in user RAM, if data is meaningful
        MOVF    TXDATA, W       ;W reg = contents of TXDATA
        MOVWF   SSPBUF          ;New data to xmit
```

15.3.6 SLAVE MODE

In Slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched, the SSPIF interrupt flag bit is set.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. When a byte is received, the device will wake-up from Sleep.

15.3.7 SLAVE SELECT SYNCHRONIZATION

The $\overline{SS}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with $\overline{SS}$ pin control enabled (SSPCON1<3:0> = 04h). The pin must not be driven low for the $\overline{SS}$ pin to function as an input. The data latch must be high. When the $\overline{SS}$ pin is low, transmission and reception are enabled and the SDO pin is driven. When the $\overline{SS}$ pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

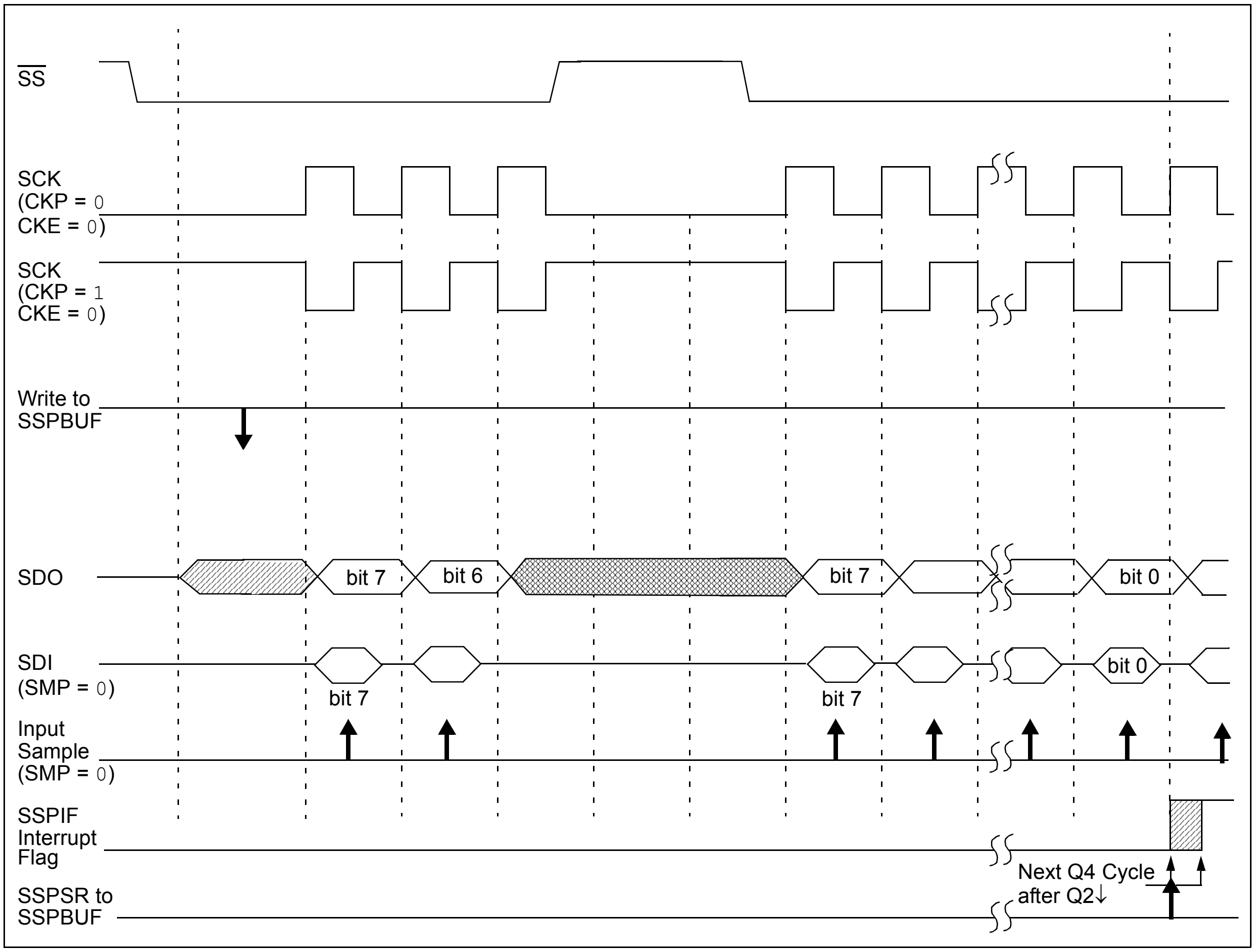
Note 1: When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPCON<3:0> = 0100), the SPI module will reset if the $\overline{SS}$ pin is set to VDD.

2: If the SPI is used in Slave mode with CKE set, then the $\overline{SS}$ pin control must be enabled.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the $\overline{SS}$ pin to a high level or clearing the SSPEN bit.

To emulate two-wire communication, the SDO pin can be connected to the SDI pin. When the SPI needs to operate as a receiver, the SDO pin can be configured as an input. This disables transmissions from the SDO. The SDI can always be left as an input (SDI function) since it cannot create a bus conflict.

FIGURE 15-4: SLAVE SYNCHRONIZATION WAVEFORM



15.4.8 I^{2}C MASTER MODE START CONDITION TIMING

To initiate a Start condition, the user sets the Start Condition Enable bit, SEN (SSPCON2<0>). If the SDA and SCL pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and starts its count. If SCL and SDA are both sampled high when the Baud Rate Generator times out (TBRG), the SDA pin is driven low. The action of the SDA being driven low while SCL is high is the Start condition and causes the S bit (SSPSTAT<3>) to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit (SSPCON2<0>) will be automatically cleared by hardware, the Baud Rate Generator is suspended, leaving the SDA line held low and the Start condition is complete.

Note:	If at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLIF, is set, the Start condition is aborted and the I^{2}C module is reset into its Idle state.

15.4.8.1 WCOL Status Flag

If the user writes the SSPBUF when a Start sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

Note:	Because queueing of events is not allowed, writing to the lower 5 bits of SSPCON2 is disabled until the Start condition is complete.

FIGURE 15-19: FIRST START BIT TIMING

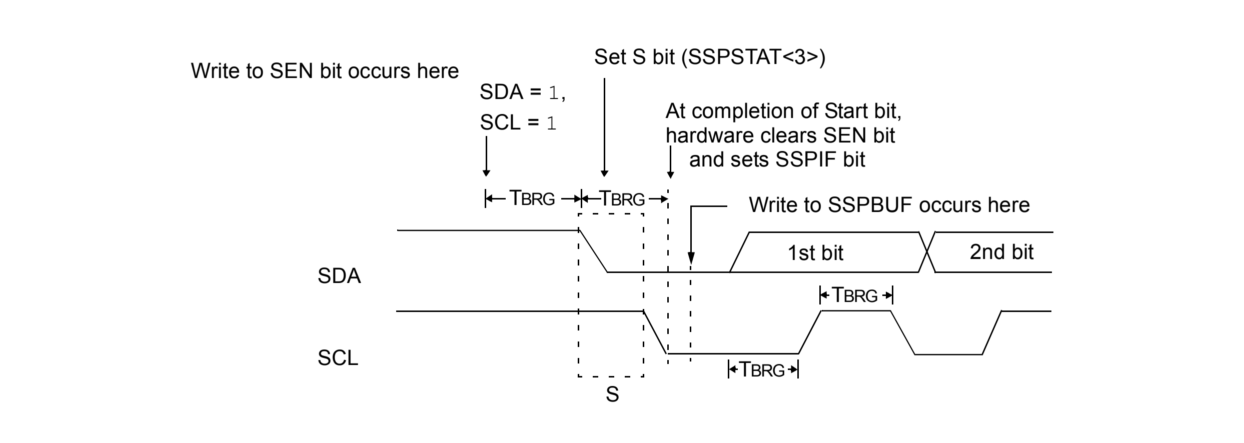
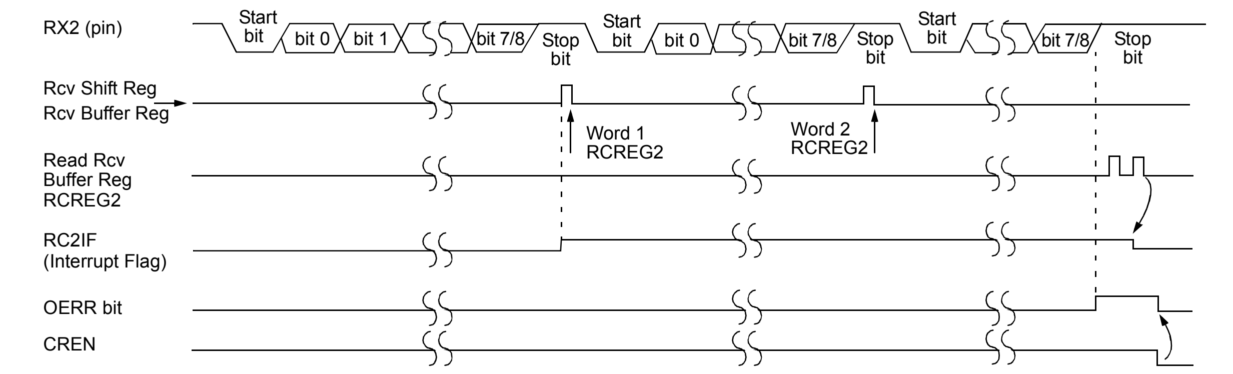

Write to SEN bit occurs here
SDA = 1,
SCL = 1
Set S bit (SSPSTAT<3>)
At completion of Start bit, hardware clears SEN bit and sets SSPIF bit
TBRG
TBRG
Write to SSPBUF occurs here
1st bit
2nd bit
SDA
TBRG
SCL
TBRG
S

FIGURE 17-5: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows three words appearing on the RX2 input. The RCREG2 (Receive Buffer register) is read after the third word causing the OERR (Overrun) bit to be set.

TABLE 17-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	59
PIR3	—	LCDIF	RC2IF	TX2IF	—	—	—	—	61
PIE3	—	LCDIE	RC2IE	TX2IE	—	—	—	—	61
IPR3	—	LCDIP	RC2IP	TX2IP	—	—	—	—	61
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	63
RCREG2	AUSART2 Receive Register								63
TXSTA2	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	63
SPBRG2	AUSART2 Baud Rate Generator Register								63

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

21.6 Operation During Sleep

When enabled, the HLVD circuitry continues to operate during Sleep. If the device voltage crosses the trip point, the HLVDIF bit will be set and the device will wake-up from Sleep. Device execution will continue from the interrupt vector address if interrupts have been globally enabled.

21.7 Effects of a Reset

A device Reset forces all registers to their Reset state. This forces the HLVD module to be turned off.

TABLE 21-1: REGISTERS ASSOCIATED WITH HIGH/LOW-VOLTAGE DETECT MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page
HLVDCON	VDIRMAG	—	IRVST	HLVDEN	HLVDL3	HLVDL2	HLVDL1	HLVDL0	60
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	59
PIR2	OSCFIF	CMIF	—	—	BCLIF	HLVDIF	TMR3IF	CCP2IF	61
PIE2	OSCFIE	CMIE	—	—	BCLIE	HLVDIE	TMR3IE	CCP2IE	61
IPR2	OSCFIP	CMIP	—	—	BCLIP	HLVDIP	TMR3IP	CCP2IP	61

Legend: — = unimplemented, read as ‘0’. Shaded cells are unused by the HLVD module.

The LCDSE5:LCDSE0 registers configure the functions of the port pins. Setting the segment enable bit for a particular segment configures that pin as an LCD driver. There are six LCD Segment Enable registers listed in Table 22-1. The prototype LCDSEx register is shown in Register 22-3.

TABLE 22-1: LCDSE REGISTERS AND ASSOCIATED SEGMENTS

Register	Segments
LCDSE0	7:0
LCDSE1	15:8
LCDSE2	23:16
LCDSE3	31:24
LCDSE4	39:32
LCDSE5	47:40

.

Note: The LCDSE5:LCDSE4 registers are not implemented in PIC18F6X90 devices.

Once the module is initialized for the LCD panel, the individual bits of the LCDDATA23:LCDDATA0 registers are cleared or set to represent a clear or dark pixel, respectively. Specific sets of LCDDATA registers are used with specific segments and common signals. Each bit represents a unique combination of a specific segment connected to a specific common. Individual LCDDATA bits are named by the convention “SxxCy”, with “xx” as the segment number and “y” as the common number. The relationship is summarized in Table 22-2. The prototype LCDDATAx register is shown in Register 22-4.

Note: Writing into the registers, LCDDATA4, LCDDATA5, LCDDATA10, LCDDATA11, LCDDATA16, LCDDATA17, LCDDATA22 and LCDDATA23, in PIC18F6X90 devices will not affect the status of any pixel and these registers can be used as General Purpose Registers.

REGISTER 22-3: LCDSEx: LCD SEGMENTx ENABLE 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
SE(n + 7)	SE(n + 6)	SE(n + 5)	SE(n + 4)	SE(n + 3)	SE(n + 2)	SE(n + 1)	SE(n)
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 7-0 **SE(n + 7):SE(n):** Segment Enable bits

For LCDSE0: n = 0
For LCDSE1: n = 8
For LCDSE2: n = 16
For LCDSE3: n = 24
For LCDSE4: n = 32
For LCDSE5: n = 40

1 = Segment function of the pin is enabled, digital I/O is disabled
0 = I/O function of the pin is enabled

TABLE 24-2: PIC18FXXXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	16-Bit Instruction Word	Status Affected	Notes
				MSb LSb		
BYTE-ORIENTED OPERATIONS						
ADDWF	f, d, a	Add WREG and f	1	`0010 01da ffff ffff`	C, DC, Z, OV, N	1, 2
ADDWFC	f, d, a	Add WREG and Carry bit to f	1	`0010 00da ffff ffff`	C, DC, Z, OV, N	1, 2
ANDWF	f, d, a	AND WREG with f	1	`0001 01da ffff ffff`	Z, N	1,2
CLRF	f, a	Clear f	1	`0110 101a ffff ffff`	Z	2
COMF	f, d, a	Complement f	1	`0001 11da ffff ffff`	Z, N	1, 2
CPFSEQ	f, a	Compare f with WREG, Skip =	1 (2 or 3)	`0110 001a ffff ffff`	None	4
CPFSGT	f, a	Compare f with WREG, Skip >	1 (2 or 3)	`0110 010a ffff ffff`	None	4
CPFSLT	f, a	Compare f with WREG, Skip <	1 (2 or 3)	`0110 000a ffff ffff`	None	1, 2
DECF	f, d, a	Decrement f	1	`0000 01da ffff ffff`	C, DC, Z, OV, N	1, 2, 3, 4
DECFSZ	f, d, a	Decrement f, Skip if 0	1 (2 or 3)	`0010 11da ffff ffff`	None	1, 2, 3, 4
DCFSNZ	f, d, a	Decrement f, Skip if Not 0	1 (2 or 3)	`0100 11da ffff ffff`	None	1, 2
INCF	f, d, a	Increment f	1	`0010 10da ffff ffff`	C, DC, Z, OV, N	1, 2, 3, 4
INCFSZ	f, d, a	Increment f, Skip if 0	1 (2 or 3)	`0011 11da ffff ffff`	None	4
INFSNZ	f, d, a	Increment f, Skip if Not 0	1 (2 or 3)	`0100 10da ffff ffff`	None	1, 2
IORWF	f, d, a	Inclusive OR WREG with f	1	`0001 00da ffff ffff`	Z, N	1, 2
MOVF	f, d, a	Move f	1	`0101 00da ffff ffff`	Z, N	1

MOVFF	f_s, f_d	Move f_s (source) to 1st word
		f_d (destination) 2nd word
			2	`1100 ffff ffff ffff`
				`1111 ffff ffff ffff`
					None	

MOVWF	f, a	Move WREG to f	1	`0110 111a ffff ffff`	None	
MULWF	f, a	Multiply WREG with f	1	`0000 001a ffff ffff`	None	1, 2
NEGF	f, a	Negate f	1	`0110 110a ffff ffff`	C, DC, Z, OV, N	
RLCF	f, d, a	Rotate Left f through Carry	1	`0011 01da ffff ffff`	C, Z, N	1, 2
RLNCF	f, d, a	Rotate Left f (No Carry)	1	`0100 01da ffff ffff`	Z, N	
RRCF	f, d, a	Rotate Right f through Carry	1	`0011 00da ffff ffff`	C, Z, N	
RRNCF	f, d, a	Rotate Right f (No Carry)	1	`0100 00da ffff ffff`	Z, N	
SETF	f, a	Set f	1	`0110 100a ffff ffff`	None	1, 2
SUBFWB	f, d, a	Subtract f from WREG with Borrow	1	`0101 01da ffff ffff`	C, DC, Z, OV, N	
SUBWF	f, d, a	Subtract WREG from f	1	`0101 11da ffff ffff`	C, DC, Z, OV, N	1, 2
SUBWFB	f, d, a	Subtract WREG from f with Borrow	1	`0101 10da ffff ffff`	C, DC, Z, OV, N	
SWAPF	f, d, a	Swap Nibbles in f	1	`0011 10da ffff ffff`	None	4
TSTFSZ	f, a	Test f, Skip if 0	1 (2 or 3)	`0110 011a ffff ffff`	None	1, 2
XORWF	f, d, a	Exclusive OR WREG with f	1	`0001 10da ffff ffff`	Z, N	

Note 1: When a PORT register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '`1`' for a pin configured as an input and is driven low by an external device, the data will be written back with a '`0`'.

2: If this instruction is executed on the TMR0 register (and where applicable, 'd' = `1`), the prescaler will be cleared if assigned.

3: If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

4: Some instructions are two-word instructions. The second word of these instructions will be executed as a `NOP` unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

5: If the table write starts the write cycle to internal memory, the write will continue until terminated.

TABLE 24-2: PIC18FXXXX INSTRUCTION SET (CONTINUED)

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word	Status Affected	Notes
			MSb LSb		
LITERAL OPERATIONS					
ADDLW k	Add Literal and WREG	1	0000 1111 kkkk kkkk	C, DC, Z, OV, N	
ANDLW k	AND Literal with WREG	1	0000 1011 kkkk kkkk	Z, N	
IORLW k	Inclusive OR Literal with WREG	1	0000 1001 kkkk kkkk	Z, N	
LFSR f, k	Move Literal (12-bit) 2nd word	2	1110 1110 00ff kkkk	None	
	to FSR(f) 1st word		1111 0000 kkkk kkkk		
MOVLB k	Move Literal to BSR<3:0>	1	0000 0001 0000 kkkk	None	
MOVLW k	Move Literal to WREG	1	0000 1110 kkkk kkkk	None	
MULLW k	Multiply Literal with WREG	1	0000 1101 kkkk kkkk	None	
RETLW k	Return with Literal in WREG	2	0000 1100 kkkk kkkk	None	
SUBLW k	Subtract WREG from Literal	1	0000 1000 kkkk kkkk	C, DC, Z, OV, N	
XORLW k	Exclusive OR Literal with WREG	1	0000 1010 kkkk kkkk	Z, N	
DATA MEMORY ↔ PROGRAM MEMORY OPERATIONS					
TBLRD*	Table Read	2	0000 0000 0000 1000	None	
TBLRD*+	Table Read with Post-Increment		0000 0000 0000 1001	None	
TBLRD*-	Table Read with Post-Decrement		0000 0000 0000 1010	None	
TBLRD+*	Table Read with Pre-Increment		0000 0000 0000 1011	None	
TBLWT*	Table Write	2	0000 0000 0000 1100	None	5
TBLWT*+	Table Write with Post-Increment		0000 0000 0000 1101	None	5
TBLWT*-	Table Write with Post-Decrement		0000 0000 0000 1110	None	5
TBLWT+*	Table Write with Pre-Increment		0000 0000 0000 1111	None	5

Note 1: When a PORT register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '`1`' for a pin configured as an input and is driven low by an external device, the data will be written back with a '`0`'.

2: If this instruction is executed on the TMR0 register (and where applicable, 'd' = `1`), the prescaler will be cleared if assigned.

3: If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

4: Some instructions are two-word instructions. The second word of these instructions will be executed as a `NOP` unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

5: If the table write starts the write cycle to internal memory, the write will continue until terminated.

MOVFF Move f to f

Syntax: MOVFF f_s,f_d

Operands: $0 \le f_s \le 4095$
$0 \le f_d \le 4095$

Operation: $(f_s) \rightarrow f_d$

Status Affected: None

Encoding:

1st word (source)	1100	ffff	ffff	$\mathtt{ffff}_s$
2nd word (destin.)	1111	ffff	ffff	$\mathtt{ffff}_d$

Description: The contents of source register 'f_s' are moved to destination register 'f_d'. Location of source 'f_s' can be anywhere in the 4096-byte data space (000h to FFFh) and location of destination 'f_d' can also be anywhere from 000h to FFFh.

Either source or destination can be W (a useful special situation).

`MOVFF` is particularly useful for transferring a data memory location to a peripheral register (such as the transmit buffer or an I/O port).

The `MOVFF` instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register

Words: 2

Cycles: 2 (3)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f' (src)	Process Data	No operation
Decode	No operation No dummy read	No operation	Write register 'f' (dest)

Example: `MOVFF REG1, REG2`

Before Instruction

REG1	=	33h
REG2	=	11h

After Instruction

REG1	=	33h
REG2	=	33h

MOVLB Move Literal to Low Nibble in BSR

Syntax: MOVLW k

Operands: $0 \le k \le 255$

Operation: $k \rightarrow$ BSR

Status Affected: None

Encoding:

0000	0001	kkkk	kkkk

Description: The eight-bit literal 'k' is loaded into the Bank Select Register (BSR). The value of BSR<7:4> always remains '`0`', regardless of the value of k_7:k_4.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write literal 'k' to BSR

Example: `MOVLB 5`

Before Instruction

BSR Register	=	02h

After Instruction

BSR Register	=	05h

25.2 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for all PIC MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

25.3 MPLAB C18 and MPLAB C30 C Compilers

The MPLAB C18 and MPLAB C30 Code Development Systems are complete ANSI C compilers for Microchip's PIC18 and PIC24 families of microcontrollers and the dsPIC30 and dsPIC33 family of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use not found with other compilers.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

25.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

25.5 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 Assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire dsPIC30F instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

25.6 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C18 and MPLAB C30 C Compilers, and the MPASM and MPLAB ASM30 Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

26.2 DC Characteristics: Power-Down and Supply Current PIC18F6390/6490/8390/8490 (Industrial) PIC18LF6390/6490/8390/8490 (Industrial) (Continued)

PIC18LF6390/6490/8390/8490 (Industrial)	**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial
PIC18F6390/6490/8390/8490 (Industrial)	**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial

Param No.	Device	Typ	Max	Units	Conditions		
	Supply Current (IDD)[2]						
	PIC18LF6390/6490/8390/8490	0.6	1.7	mA	-40°C	VDD = 2.0V	FOSC = 4 MHz (**RC_RUN** mode, INTOSC source)
		0.6	1.6	mA	+25°C		
		0.6	1.5	mA	+85°C		
	PIC18LF6390/6490/8390/8490	1.0	2.4	mA	-40°C	VDD = 3.0V	
		1.0	2.4	mA	+25°C		
		1.0	2.4	mA	+85°C		
	All devices	2.0	4.2	mA	-40°C	VDD = 5.0V	
		2.0	4	mA	+25°C		
		2.0	3.8	mA	+85°C		
	PIC18LF6390/6490/8390/8490	2.3	6.4	μA	-40°C	VDD = 2.0V	FOSC = 31 kHz (**RC_IDLE** mode, INTRC source)
		2.5	6.4	μA	+25°C		
		2.9	8.8	μA	+85°C		
	PIC18LF6390/6490/8390/8490	3.6	8.8	μA	-40°C	VDD = 3.0V	
		3.8	8.8	μA	+25°C		
		4.6	12	μA	+85°C		
	All devices	7.4	16	μA	-40°C	VDD = 5.0V	
		7.8	16	μA	+25°C		
		9.1	29	μA	+85°C		

Legend: Shading of rows is to assist in readability of the table.

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD or VSS;
$\overline{\text{MCLR}}$ = VDD; WDT enabled/disabled as specified.

3: Low-power Timer1 oscillator selected.

4: BOR and HLVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.