

Welcome to [E-XFL.COM](http://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.

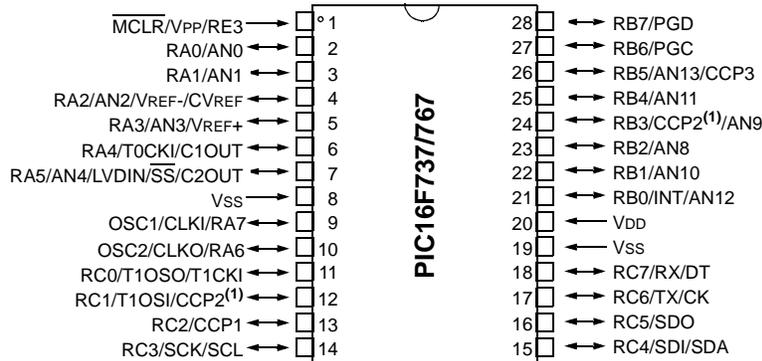
### Applications of "[Embedded - Microcontrollers](#)"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	36
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	368 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 14x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic16f747-i-ml">https://www.e-xfl.com/product-detail/microchip-technology/pic16f747-i-ml</a>

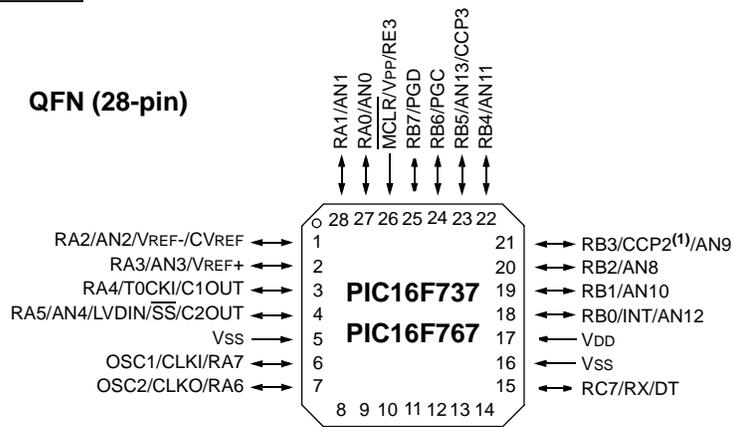
# PIC16F7X7

## Pin Diagrams

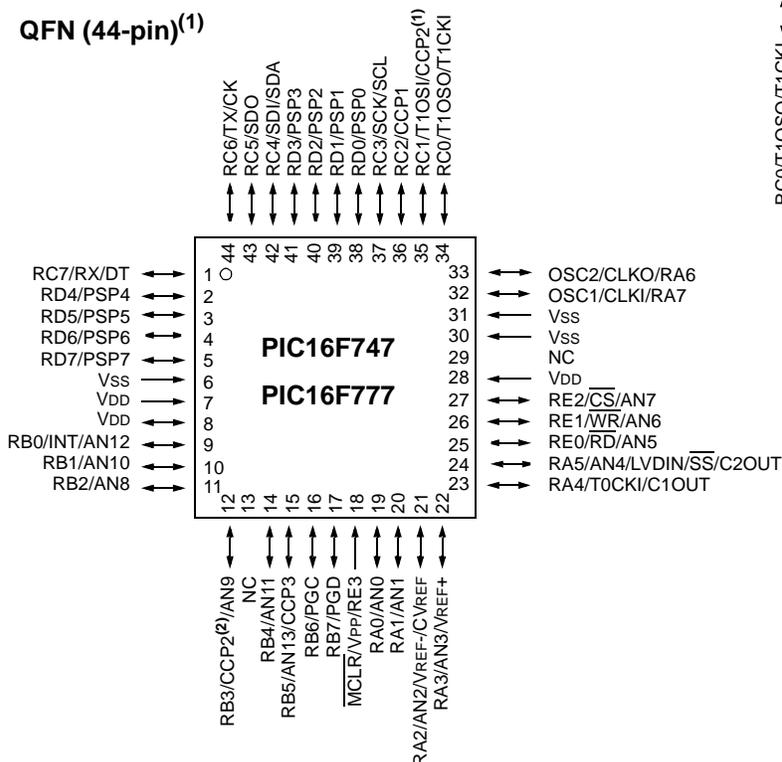
### PDIP, SOIC, SSOP (28-pin)



### QFN (28-pin)



### QFN (44-pin)<sup>(1)</sup>



**Note 1:** For the QFN package, it is recommended that the bottom pad be connected to Vss.

**Note 2:** Pin location of CCP2 is determined by the CCPMX bit in Configuration Word Register 1.

## 1.0 DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC16F737
- PIC16F767
- PIC16F747
- PIC16F777

PIC16F737/767 devices are available only in 28-pin packages, while PIC16F747/777 devices are available in 40-pin and 44-pin packages. All devices in the PIC16F7X7 family share common architecture with the following differences:

- The PIC16F737 and PIC16F767 have one-half of the total on-chip memory of the PIC16F747 and PIC16F777.
- The 28-pin devices have 3 I/O ports, while the 40/44-pin devices have 5.
- The 28-pin devices have 16 interrupts, while the 40/44-pin devices have 17.
- The 28-pin devices have 11 A/D input channels, while the 40/44-pin devices have 14.
- The Parallel Slave Port is implemented only on the 40/44-pin devices.
- Low-Power modes: RC\_RUN allows the core and peripherals to be clocked from the INTRC, while SEC\_RUN allows the core and peripherals to be clocked from the low-power Timer1. Refer to **Section 4.7 “Power-Managed Modes”** for further details.
- Internal RC oscillator with eight selectable frequencies, including 31.25 kHz, 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz and 8 MHz. The INTRC can be configured as a primary or secondary clock source. Refer to **Section 4.5 “Internal Oscillator Block”** for further details.

- The Timer1 module current consumption has been greatly reduced from 20  $\mu$ A (previous PIC16 devices) to 1.8  $\mu$ A typical (32 kHz at 2V), which is ideal for real-time clock applications. Refer to **Section 7.0 “Timer1 Module”** for further details.
- Extended Watchdog Timer (WDT) that can have a programmable period from 1 ms to 268s. The WDT has its own 16-bit prescaler. Refer to **Section 15.17 “Watchdog Timer (WDT)”** for further details.
- Two-Speed Start-up: When the oscillator is configured for LP, XT or HS, this feature will clock the device from the INTRC while the oscillator is warming up. This, in turn, will enable almost immediate code execution. Refer to **Section 15.17.3 “Two-Speed Clock Start-up Mode”** for further details.
- Fail-Safe Clock Monitor: This feature will allow the device to continue operation if the primary or secondary clock source fails by switching over to the INTRC.

The available features are summarized in Table 1-1. Block diagrams of the PIC16F737/767 and PIC16F747/777 devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2 and Table 1-3.

Additional information may be found in the “PIC<sup>®</sup> Mid-Range MCU Family Reference Manual” (DS33023) which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

**TABLE 1-1: PIC16F7X7 DEVICE FEATURES**

Key Features	PIC16F737	PIC16F747	PIC16F767	PIC16F777
Operating Frequency	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	368	368	368	368
Interrupts	16	17	16	17
I/O Ports	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C	Ports A, B, C, D, E
Timers	3	3	3	3
Capture/Compare/PWM Modules	3	3	3	3
Master Serial Communications	MSSP, AUSART	MSSP, AUSART	MSSP, AUSART	MSSP, AUSART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	11 Input Channels	14 Input Channels	11 Input Channels	14 Input Channels
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packaging	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin QFN 44-pin TQFP	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin QFN 44-pin TQFP

**TABLE 1-3: PIC16F747 AND PIC16F777 PINOUT DESCRIPTION (CONTINUED)**

Pin Name	PDIP Pin #	QFN Pin #	TQFP Pin #	I/O/P Type	Buffer Type	Description
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	15	34	32	I/O O I	ST	PORTC is a bidirectional I/O port.  Digital I/O. Timer1 oscillator output. Timer1 external clock input.
RC1/T1OSI/CCP2 RC1 T1OSI CCP2 <sup>(5)</sup>	16	35	35	I/O I I/O	ST	Digital I/O. Timer1 oscillator input. Capture 2 input, Compare 2 output, PWM 2 output.
RC2/CCP1 RC2 CCP1	17	36	36	I/O I/O	ST	Digital I/O. Capture 1 input, Compare 1 output, PWM 1 output.
RC3/SCK/SCL RC3 SCK  SCL	18	37	37	I/O I/O  I/O	ST	Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I <sup>2</sup> C™ mode.
RC4/SDI/SDA RC4 SDI SDA	23	42	42	I/O I I/O	ST	Digital I/O. SPI data in. I <sup>2</sup> C data I/O.
RC5/SDO RC5 SDO	24	43	43	I/O O	ST	Digital I/O. SPI data out.
RC6/TX/CK RC6 TX CK	25	44	44	I/O O I/O	ST	Digital I/O. AUSART asynchronous transmit. AUSART synchronous clock.
RC7/RX/DT RC7 RX DT	26	1	1	I/O I I/O	ST	Digital I/O. AUSART asynchronous receive. AUSART synchronous data.

**Legend:** I = input                      O = output                      I/O = input/output                      P = power  
 — = Not used                      TTL = TTL input                      ST = Schmitt Trigger input

- Note**
- 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
  - 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
  - 3: This buffer is a Schmitt Trigger input when configured as a general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
  - 4: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.
  - 5: Pin location of CCP2 is determined by the CCPMX bit in Configuration Word Register 1.

## 2.2.2.1 Status Register

The Status register contains the arithmetic status of the ALU, the Reset status and the bank select bits for data memory.

The Status register can be the destination for any instruction, as with any other register. If the Status register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the  $\overline{TO}$  and  $\overline{PD}$  bits are not writable, therefore, the result of an instruction with the Status register as destination may be different than intended.

For example, `CLRF STATUS`, will clear the upper three bits and set the Z bit. This leaves the Status register as `000u u1uu` (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the Status register because these instructions do not affect the Z, C or DC bits from the Status register. For other instructions not affecting any Status bits, see **Section 16.0 "Instruction Set Summary"**.

**Note 1:** The  $\overline{C}$  and  $\overline{DC}$  bits operate as a borrow and digit borrow bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

### REGISTER 2-1: STATUS: ARITHMETIC STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)

	R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7								bit 0

- bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)  
 1 = Bank 2, 3 (100h-1FFh)  
 0 = Bank 0, 1 (00h-FFh)
  - bit 6-5 **RP1:RP0:** Register Bank Select bits (used for direct addressing)  
 11 = Bank 3 (180h-1FFh)  
 10 = Bank 2 (100h-17Fh)  
 01 = Bank 1 (80h-FFh)  
 00 = Bank 0 (00h-7Fh)  
 Each bank is 128 bytes.
  - bit 4  **$\overline{TO}$ :** Time-out bit  
 1 = After power-up, `CLRWDT` instruction or `SLEEP` instruction  
 0 = A WDT time-out occurred
  - bit 3  **$\overline{PD}$ :** Power-Down bit  
 1 = After power-up or by the `CLRWDT` instruction  
 0 = By execution of the `SLEEP` instruction
  - bit 2 **Z:** Zero bit  
 1 = The result of an arithmetic or logic operation is zero  
 0 = The result of an arithmetic or logic operation is not zero
  - bit 1 **DC:** Digit Carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)  
 1 = A carry-out from the 4th low-order bit of the result occurred  
 0 = No carry-out from the 4th low-order bit of the result
  - bit 0 **C:** Carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)  
 1 = A carry-out from the Most Significant bit of the result occurred  
 0 = No carry-out from the Most Significant bit of the result occurred
- Note:** For borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low-order bit of the source 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

**TABLE 4-4: CLOCK SWITCHING MODES**

Current System Clock	SCS bits<1:0> Modified to:	Delay	OSTS bit	IOFS bit	T1RUN bit	New System Clock	Comments
LP, XT, HS, T1OSC, EC, RC	10 (INTRC) FOSC<2:0> = LP, XT or HS	8 Clocks of INTRC	0	1 <sup>(1)</sup>	0	INTRC or INTOSC or INTOSC Postscaler	The internal RC oscillator frequency is dependant upon the IRCF bits.
LP, XT, HS, INTRC, EC, RC	01 (T1OSC) FOSC<2:0> = LP, XT or HS	8 Clocks of T1OSC	0	N/A	1	T1OSC	T1OSCEN bit must be enabled.
INTRC T1OSC	00 FOSC<2:0> = EC or FOSC<2:0> = RC	8 Clocks of EC or RC	1	N/A	0	EC or RC	
INTRC T1OSC	00 FOSC<2:0> = LP, XT, HS	1024 Clocks + 8 Clocks of LP, XT, HS	1	N/A	0	LP, XT, HS	During the 1024 clocks, program execution is clocked from the secondary oscillator until the primary oscillator becomes stable.
LP, XT, HS	00 (Due to Reset) LP, XT, HS	1024 Clocks	1	N/A	0	LP, XT, HS	When a Reset occurs, there is no clock transition sequence. Instruction execution and/or peripheral operation is suspended unless Two-Speed Start-up mode is enabled, after which the INTRC will act as the system clock until the Oscillator Start-up Timer has expired.

**Note 1:** If the new clock source is the INTOSC or INTOSC postscaler, then the IOFS bit will be set 4 ms (approx.) after the clock change.

## 5.0 I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Additional information on I/O ports may be found in the “PIC® Mid-Range MCU Family Reference Manual” (DS33023).

### 5.1 PORTA and the TRISA Register

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

Reading the PORTA register reads the status of the pins, whereas writing to it, will write to the port latch.

The RA4 pin is multiplexed with the Timer0 module clock input and one of the comparator outputs to become the RA4/T0CKI/C1OUT pin. Pins RA6 and RA7 are multiplexed with the main oscillator pins; they are enabled as oscillator or I/O pins by the selection of the main oscillator in Configuration Register 1H (see **Section 15.1 “Configuration Bits”** for details). When they are not used as port pins, RA6 and RA7 and their associated TRIS and LAT bits are read as ‘0’.

The other PORTA pins are multiplexed with analog inputs, the analog VREF+ and VREF- inputs and the comparator voltage reference output. The operation of pins RA3:RA0 and RA5 as A/D converter inputs is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register 1). Pins RA0 through RA5 may also be used as comparator inputs or outputs by setting the appropriate bits in the CMCON register.

**Note:** On a Power-on Reset, RA5 and RA3:RA0 are configured as analog inputs and read as ‘0’. RA4 is configured as a digital input.

The RA4/T0CKI/C1OUT pin is a Schmitt Trigger input and an open-drain output. All other PORTA pins have TTL input levels and full CMOS output drivers.

The TRISA register controls the direction of the RA pins even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

#### EXAMPLE 5-1: INITIALIZING PORTA

```
BCF     STATUS, RP0    ;
BCF     STATUS, RP1    ; Bank0
CLRF    PORTA          ; Initialize PORTA by
                        ; clearing output
                        ; data latches
BSF     STATUS, RP0    ; Select Bank 1
MOVLW   0x0F           ; Configure all pins
MOVWF   ADCON1         ; as digital inputs
MOVLW   0xCF           ; Value used to
                        ; initialize data
                        ; direction
MOVWF   TRISA          ; Set RA<3:0> as inputs
                        ; RA<5:4> as outputs
                        ; TRISA<7:6>are always
                        ; read as '0'.
```

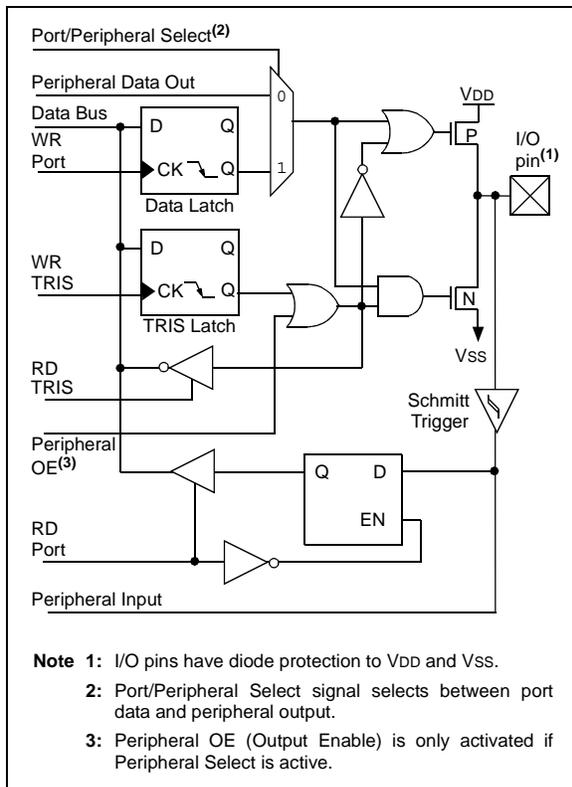
## 5.3 PORTC and the TRISC Register

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

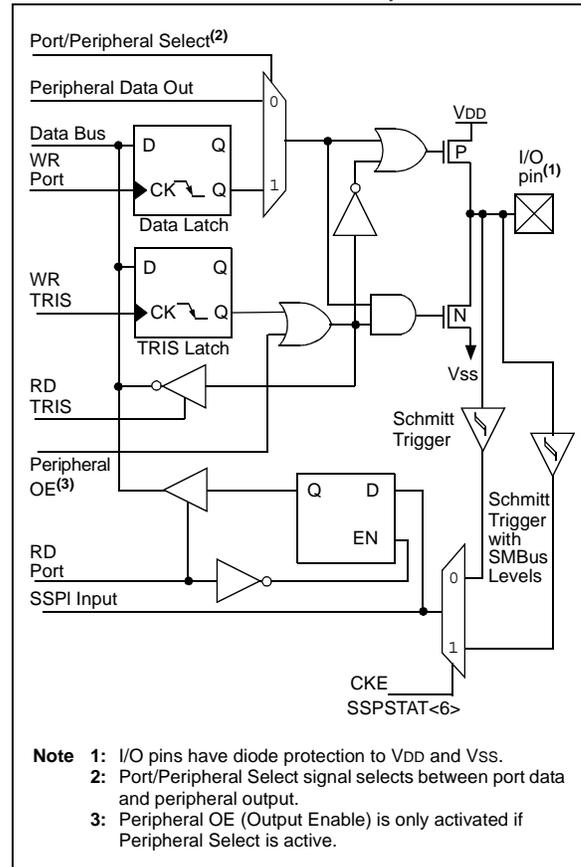
PORTC is multiplexed with several peripheral functions (Table 5-5). PORTC pins have Schmitt Trigger input buffers.

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. Since the TRIS bit override is in effect while the peripheral is enabled, read-modify-write instructions (BSF, BCF, XORWF) with TRISC as destination should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings and to **Section 16.1 “Read-Modify-Write Operations”** for additional information on read-modify-write operations.

**FIGURE 5-16: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<2:0>, RC<7:5> PINS**



**FIGURE 5-17: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<4:3> PINS**



**REGISTER 10-2: SSPCON: MSSP CONTROL (SPI MODE) REGISTER 1 (ADDRESS 14h)**

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| WCOL  | SSPOV | SSPEN | CKP   | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| bit 7 |       |       |       |       |       |       | bit 0 |

bit 7 **WCOL:** Write Collision Detect bit (Transmit mode only)

- 1 = The SSPBUF register is written while it is still transmitting the previous word. (Must be cleared in software.)
- 0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit

SPI Slave mode:

- 1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. (Must be cleared in software.)
- 0 = No overflow

**Note:** In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

bit 5 **SSPEN:** Synchronous Serial Port Enable bit

- 1 = Enables serial port and configures SCK, SDO, SDI and  $\overline{SS}$  as serial port pins
- 0 = Disables serial port and configures these pins as I/O port pins

**Note:** When enabled, these pins must be properly configured as input or output.

bit 4 **CKP:** Clock Polarity Select bit

- 1 = Idle state for clock is a high level
- 0 = Idle state for clock is a low level

bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits

- 0101 = SPI Slave mode, clock = SCK pin.  $\overline{SS}$  pin control disabled.  $\overline{SS}$  can be used as I/O pin.
- 0100 = SPI Slave mode, clock = SCK pin.  $\overline{SS}$  pin control enabled.
- 0011 = SPI Master mode, clock = TMR2 output/2
- 0010 = SPI Master mode, clock = FOSC/64
- 0001 = SPI Master mode, clock = FOSC/16
- 0000 = SPI Master mode, clock = FOSC/4

**Note:** Bit combinations not specifically listed here are either reserved or implemented in I<sup>2</sup>C mode only.

**Legend:**

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

### 10.3.3 ENABLING SPI I/O

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

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<5> bit cleared
- SCK (Master mode) must have TRISC<3> bit cleared
- SCK (Slave mode) must have TRISC<3> bit set
- $\overline{SS}$  must have TRISA<5> bit set

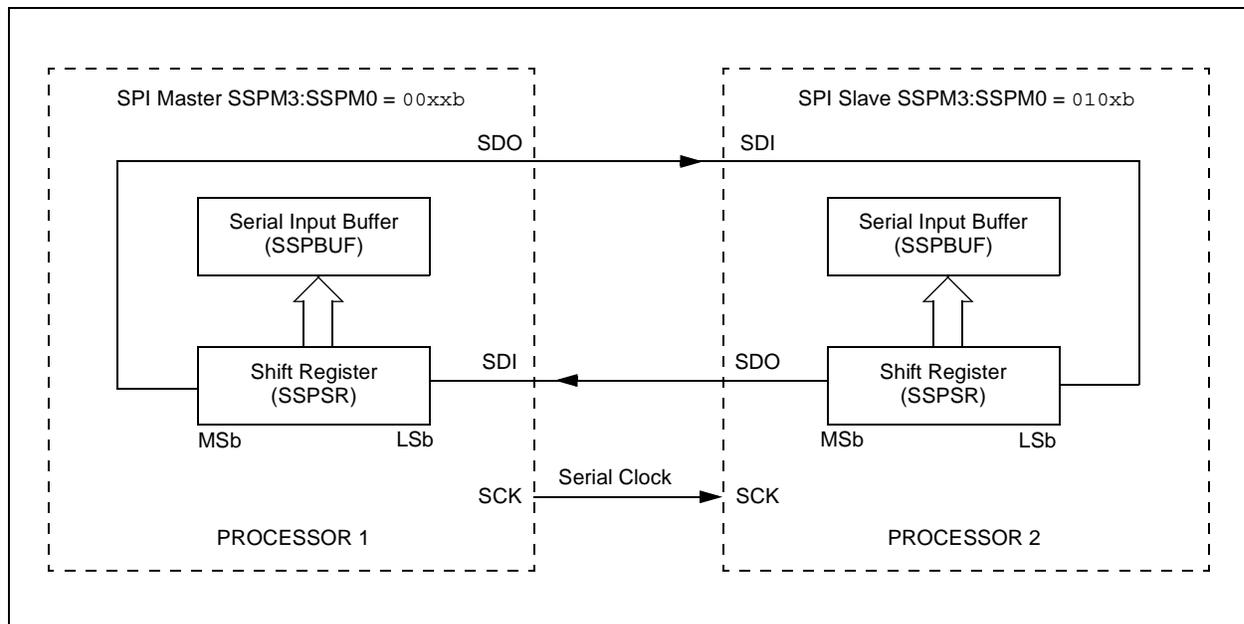
Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

### 10.3.4 TYPICAL CONNECTION

Figure 10-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data – Slave sends dummy data
- Master sends data – Slave sends data
- Master sends dummy data – Slave sends data

**FIGURE 10-2: SPI MASTER/SLAVE CONNECTION**



# PIC16F7X7

## 10.4.4 CLOCK STRETCHING

Both 7-bit and 10-bit Slave modes implement automatic clock stretching during a transmit sequence.

The SEN bit (SSPCON2<0>) allows clock stretching to be enabled during receives. Setting SEN will cause the SCL pin to be held low at the end of each data receive sequence.

### 10.4.4.1 Clock Stretching for 7-bit Slave Receive Mode (SEN = 1)

In 7-bit Slave Receive mode, on the falling edge of the ninth clock, at the end of the ACK sequence if the BF bit is set, the CKP bit in the SSPCON register is automatically cleared, forcing the SCL output to be held low. The CKP being cleared to '0' will assert the SCL line low. The CKP bit must be set in the user's ISR before reception is allowed to continue. By holding the SCL line low, the user has time to service the ISR and read the contents of the SSPBUF before the master device can initiate another receive sequence. This will prevent buffer overruns from occurring (see Figure 10-13).

**Note 1:** If the user reads the contents of the SSPBUF before the falling edge of the ninth clock, thus clearing the BF bit, the CKP bit will not be cleared and clock stretching will not occur.

**2:** The CKP bit can be set in software regardless of the state of the BF bit. The user should be careful to clear the BF bit in the ISR before the next receive sequence in order to prevent an overflow condition.

### 10.4.4.2 Clock Stretching for 10-bit Slave Receive Mode (SEN = 1)

In 10-bit Slave Receive mode during the address sequence, clock stretching automatically takes place but CKP is not cleared. During this time, if the UA bit is set after the ninth clock, clock stretching is initiated. The UA bit is set after receiving the upper byte of the 10-bit address and following the receive of the second byte of the 10-bit address, with the R/W bit cleared to '0'. The release of the clock line occurs upon updating SSPADD. Clock stretching will occur on each data receive sequence as described in 7-bit mode.

**Note:** If the user polls the UA bit and clears it by updating the SSPADD register before the falling edge of the ninth clock occurs and if the user hasn't cleared the BF bit by reading the SSPBUF register before that time, then the CKP bit will still NOT be asserted low. Clock stretching on the basis of the state of the BF bit only occurs during a data sequence, not an address sequence.

### 10.4.4.3 Clock Stretching for 7-bit Slave Transmit Mode

7-bit Slave Transmit mode implements clock stretching by clearing the CKP bit after the falling edge of the ninth clock, if the BF bit is clear. This occurs regardless of the state of the SEN bit.

The user's ISR must set the CKP bit before transmission is allowed to continue. By holding the SCL line low, the user has time to service the ISR and load the contents of the SSPBUF before the master device can initiate another transmit sequence (see Figure 10-9).

**Note 1:** If the user loads the contents of SSPBUF, setting the BF bit before the falling edge of the ninth clock, the CKP bit will not be cleared and clock stretching will not occur.

**2:** The CKP bit can be set in software regardless of the state of the BF bit.

### 10.4.4.4 Clock Stretching for 10-bit Slave Transmit Mode

In 10-bit Slave Transmit mode, clock stretching is controlled during the first two address sequences by the state of the UA bit, just as it is in 10-bit Slave Receive mode. The first two addresses are followed by a third address sequence, which contains the high-order bits of the 10-bit address and the R/W bit set to '1'. After the third address sequence is performed, the UA bit is not set, the module is now configured in Transmit mode and clock stretching is controlled by the BF flag as in 7-bit Slave Transmit mode (see Figure 10-11).

## 10.4.6 MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPM bits in SSPCON and by setting the SSPEN bit. In Master mode, the SCL and SDA lines are manipulated by the MSSP hardware.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I<sup>2</sup>C bus may be taken when the P bit is set or the bus is Idle, with both the S and P bits clear.

In Firmware Controlled Master mode, user code conducts all I<sup>2</sup>C bus operations based on Start and Stop bit conditions.

Once Master mode is enabled, the user has six options:

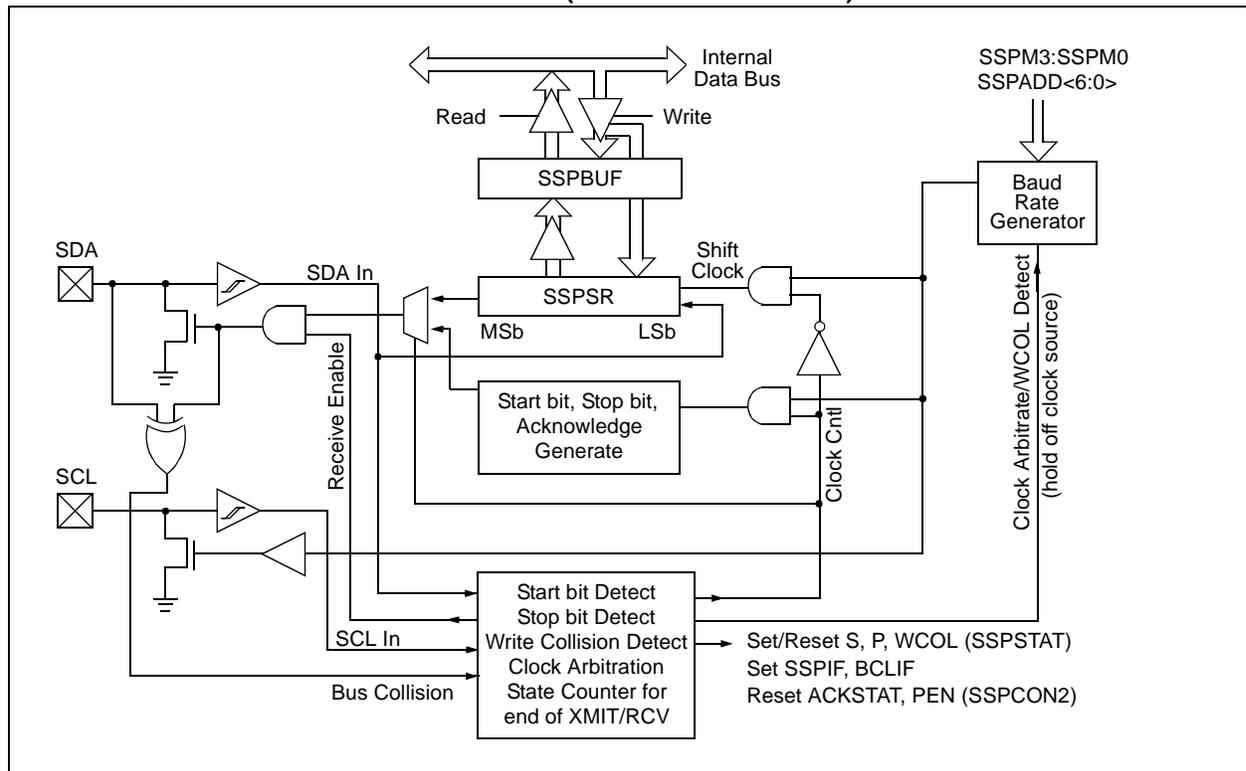
1. Assert a Start condition on SDA and SCL.
2. Assert a Repeated Start condition on SDA and SCL.
3. Write to the SSPBUF register, initiating transmission of data/address.
4. Configure the I<sup>2</sup>C port to receive data.
5. Generate an Acknowledge condition at the end of a received byte of data.
6. Generate a Stop condition on SDA and SCL.

**Note:** The MSSP module, when configured in I<sup>2</sup>C Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPBUF register to initiate transmission before the Start condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur.

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

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

**FIGURE 10-16: MSSP BLOCK DIAGRAM (I<sup>2</sup>C™ MASTER MODE)**



# PIC16F7X7

## 15.8 Low-Voltage Detect

In many applications, the ability to determine if the device voltage (VDD) is below a specified voltage level is a desirable feature. A window of operation for the application can be created where the application software can do “housekeeping tasks” before the device voltage exits the valid operating range. This can be done using the Low-Voltage Detect module.

This module is a software programmable circuitry where a device voltage trip point can be specified. When the voltage of the device becomes lower than the specified point, an interrupt flag is set. If the interrupt is enabled, the program execution will branch to the interrupt vector address and the software can then respond to that interrupt source.

The Low-Voltage Detect circuitry is completely under software control. This allows the circuitry to be turned off by the software which minimizes the current consumption for the device.

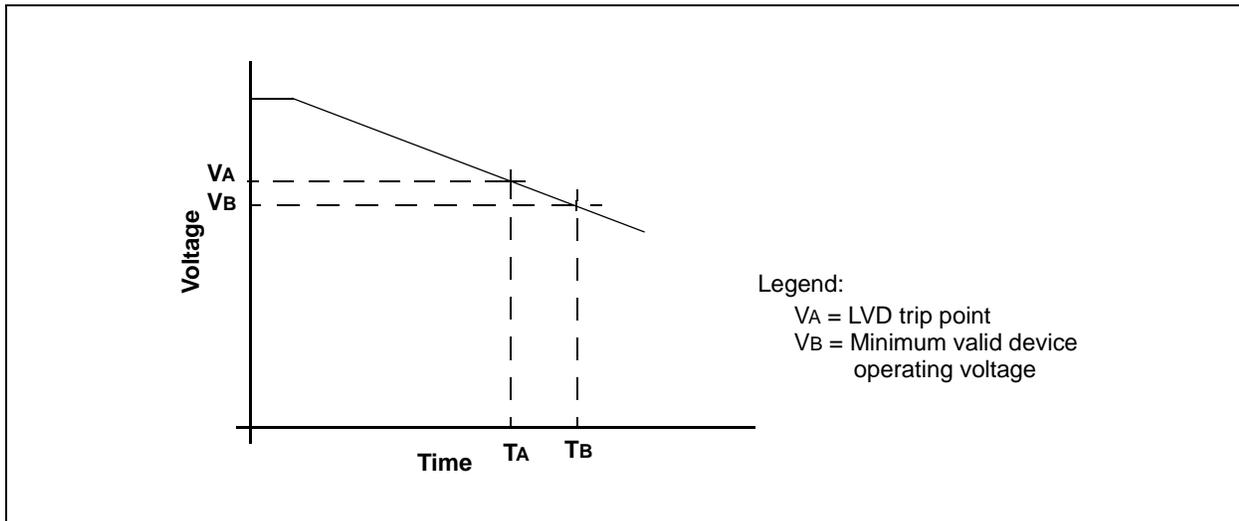
Figure 15-3 shows a possible application voltage curve (typically for batteries). Over time, the device voltage decreases. When the device voltage equals voltage VA, the LVD logic generates an interrupt. This occurs at

time TA. The application software then has the time, until the device voltage is no longer in valid operating range, to shut-down the system. Voltage point VB is the minimum valid operating voltage specification. This occurs at time TB. The difference, TB – TA, is the total time for shutdown.

The block diagram for the LVD module is shown in Figure 15-4. A comparator uses an internally generated reference voltage as the set point. When the selected tap output of the device voltage crosses the set point (is lower than), the LVDIF bit is set.

Each node in the resistor divider represents a “trip point” voltage. The “trip point” voltage is the minimum supply voltage level at which the device can operate before the LVD module asserts an interrupt. When the supply voltage is equal to the trip point, the voltage tapped off of the resistor array is equal to the 1.2V internal reference voltage generated by the voltage reference module. The comparator then generates an interrupt signal setting the LVDIF bit. This voltage is software programmable to any one of 16 values (see Figure 15-4). The trip point is selected by programming the LVDL3:LVDL0 bits (LVDCON<3:0>).

**FIGURE 15-3: TYPICAL LOW-VOLTAGE DETECT APPLICATION**



# PIC16F7X7

---

## 15.10.1 REFERENCE VOLTAGE SET POINT

The internal reference voltage of the LVD module may be used by other internal circuitry (the Programmable Brown-out Reset). If these circuits are disabled (lower current consumption), the reference voltage circuit requires a time to become stable before a low-voltage condition can be reliably detected. This time is invariant of system clock speed. This start-up time is specified in electrical specification parameter #36. The low-voltage interrupt flag will not be enabled until a stable reference voltage is reached. Refer to the waveform in Figure 15-6.

## 15.10.2 CURRENT CONSUMPTION

When the module is enabled, the LVD comparator and voltage divider are enabled and will consume static current. The voltage divider can be tapped from multiple places in the resistor array. Total current consumption, when enabled, is specified in electrical specification parameter #D022B.

## 15.11 Operation During Sleep

When enabled, the LVD circuitry continues to operate during Sleep. If the device voltage crosses the trip point, the LVDIF 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.

## 15.12 Effects of a Reset

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

**Note:** If the LVD is enabled and the BOR module is not enabled, the band gap will require a start-up time of no more than 50  $\mu$ s before the band gap reference is stable. Before enabling the LVD interrupt, the user should ensure that the band gap reference voltage is stable by monitoring the IRVST bit in the LVDCON register. The LVD could cause erroneous interrupts before the band gap is stable.

## 15.13 Time-out Sequence

On power-up, the time-out sequence is as follows: the PWRT delay starts (if enabled) when a POR occurs; then, OST starts counting 1024 oscillator cycles when PWRT ends (LP, XT, HS); when the OST ends, the device comes out of Reset.

If  $\overline{\text{MCLR}}$  is kept low long enough, all delays will expire. Bringing  $\overline{\text{MCLR}}$  high will begin execution immediately. This is useful for testing purposes or to synchronize more than one PIC16F7X7 device operating in parallel.

Table 15-3 shows the Reset conditions for the Status, PCON and PC registers, while Table 15-4 shows the Reset conditions for all the registers.

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

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

## 16.2 Instruction Descriptions

### **ADDLW**      **Add Literal and W**

**Syntax:**      [ *label* ] ADDLW    *k*

**Operands:**     $0 \leq k \leq 255$

**Operation:**     $(W) + k \rightarrow (W)$

**Status Affected:**    C, DC, Z

**Description:**    The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register.

### **ADDWF**      **Add W and f**

**Syntax:**      [ *label* ] ADDWF    *f,d*

**Operands:**     $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:**     $(W) + (f) \rightarrow (\text{destination})$

**Status Affected:**    C, DC, Z

**Description:**    Add the contents of the W register with 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'.

### **ANDLW**      **AND Literal with W**

**Syntax:**      [ *label* ] ANDLW    *k*

**Operands:**     $0 \leq k \leq 255$

**Operation:**     $(W) .\text{AND.} (k) \rightarrow (W)$

**Status Affected:**    Z

**Description:**    The contents of W register are ANDed with the eight-bit literal 'k'. The result is placed in the W register.

### **ANDWF**      **AND W with f**

**Syntax:**      [ *label* ] ANDWF    *f,d*

**Operands:**     $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:**     $(W) .\text{AND.} (f) \rightarrow (\text{destination})$

**Status Affected:**    Z

**Description:**    AND the W register with 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'.

### **BCF**          **Bit Clear f**

**Syntax:**      [ *label* ] BCF      *f,b*

**Operands:**     $0 \leq f \leq 127$   
 $0 \leq b \leq 7$

**Operation:**     $0 \rightarrow (f<b>)$

**Status Affected:**    None

**Description:**    Bit 'b' in register 'f' is cleared.

### **BSF**          **Bit Set f**

**Syntax:**      [ *label* ] BSF      *f,b*

**Operands:**     $0 \leq f \leq 127$   
 $0 \leq b \leq 7$

**Operation:**     $1 \rightarrow (f<b>)$

**Status Affected:**    None

**Description:**    Bit 'b' in register 'f' is set.

### **BTFSS**      **Bit Test f, Skip if Set**

**Syntax:**      [ *label* ] BTFSS    *f,b*

**Operands:**     $0 \leq f \leq 127$   
 $0 \leq b < 7$

**Operation:**    skip if  $(f<b>) = 1$

**Status Affected:**    None

**Description:**    If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2 Tcy instruction.

### **BTFSC**      **Bit Test, Skip if Clear**

**Syntax:**      [ *label* ] BTFSC    *f,b*

**Operands:**     $0 \leq f \leq 127$   
 $0 \leq b \leq 7$

**Operation:**    skip if  $(f<b>) = 0$

**Status Affected:**    None

**Description:**    If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b' in register 'f' is '0', the next instruction is discarded and a NOP is executed instead, making this a 2 Tcy instruction.

## RLF Rotate Left f through Carry

**Syntax:** [ *label* ] RLF f,d

**Operands:**  $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:** See description below

**Status Affected:** C

**Description:** The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is stored back in register 'f'.



## SLEEP Enter Sleep mode

**Syntax:** [ *label* ] SLEEP

**Operands:** None

**Operation:** 00h → WDT,  
 0 → WDT prescaler,  
 1 →  $\overline{TO}$ ,  
 0 → PD

**Status Affected:**  $\overline{TO}$ , PD

**Description:** The Power-Down status bit,  $\overline{PD}$ , is cleared. Time-out status bit,  $\overline{TO}$ , is set. Watchdog Timer and its prescaler are cleared. The processor is put into Sleep mode with the oscillator stopped.

## RETURN Return from Subroutine

**Syntax:** [ *label* ] RETURN

**Operands:** None

**Operation:** TOS → PC

**Status Affected:** None

**Description:** Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

## SUBLW Subtract W from Literal

**Syntax:** [ *label* ] SUBLW k

**Operands:**  $0 \leq k \leq 255$

**Operation:**  $k - (W) \rightarrow (W)$

**Status Affected:** C, DC, Z

**Description:** The W register is subtracted (2's complement method) from the eight-bit literal 'k'. The result is placed in the W register.

## RRF Rotate Right f through Carry

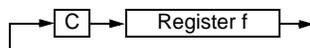
**Syntax:** [ *label* ] RRF f,d

**Operands:**  $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:** See description below

**Status Affected:** C

**Description:** The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.



## SUBWF Subtract W from f

**Syntax:** [ *label* ] SUBWF f,d

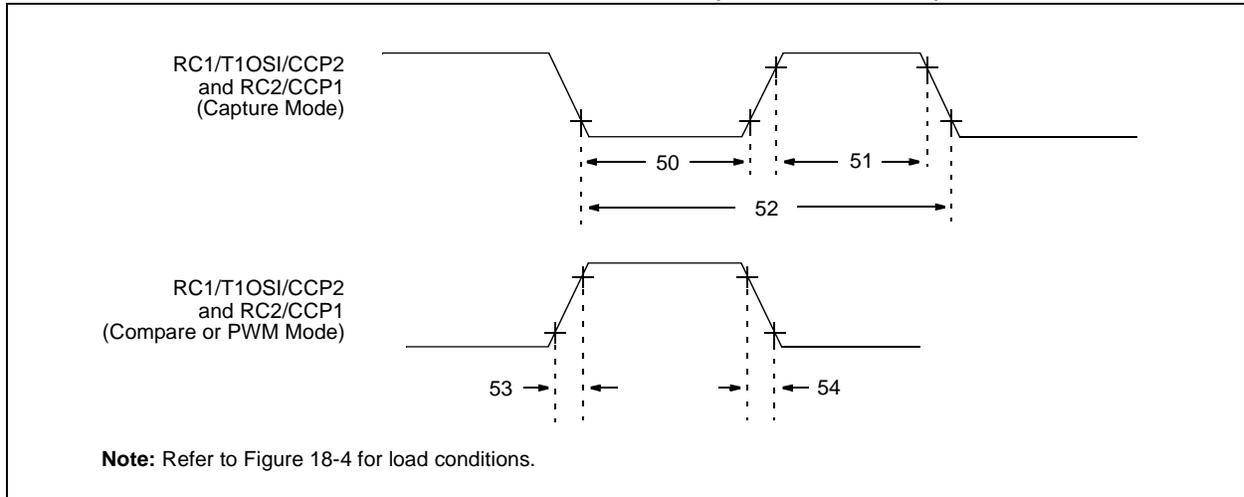
**Operands:**  $0 \leq f \leq 127$   
 $d \in [0,1]$

**Operation:**  $(f) - (W) \rightarrow (\text{destination})$

**Status Affected:** C, DC, Z

**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'.

**FIGURE 18-10: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)**



**TABLE 18-8: CAPTURE/COMPARE/PWM REQUIREMENTS (ALL CCP MODULES)**

Param No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions	
50*	TcCL	CCP1, CCP2 and CCP3 Input Low Time	No prescaler	$0.5 T_{CY} + 20$	—	—	ns		
			With prescaler	PIC16F7X7	10	—	—		ns
				PIC16LF7X7	20	—	—		ns
51*	TcCH	CCP1, CCP2 and CCP3 Input High Time	No prescaler	$0.5 T_{CY} + 20$	—	—	ns		
			With prescaler	PIC16F7X7	10	—	—		ns
				PIC16LF7X7	20	—	—		ns
52*	TcCP	CCP1, CCP2 and CCP3 Input Period		$\frac{3 T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 4 or 16)	
53*	TcCR	CCP1, CCP2 and CCP3 Output Rise Time	PIC16F7X7	—	10	25	ns		
			PIC16LF7X7	—	25	50	ns		
54*	TcCF	CCP1, CCP2 and CCP3 Output Fall Time	PIC16F7X7	—	10	25	ns		
			PIC16LF7X7	—	25	45	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.

# PIC16F7X7

FIGURE 19-3: TYPICAL  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (XT MODE)

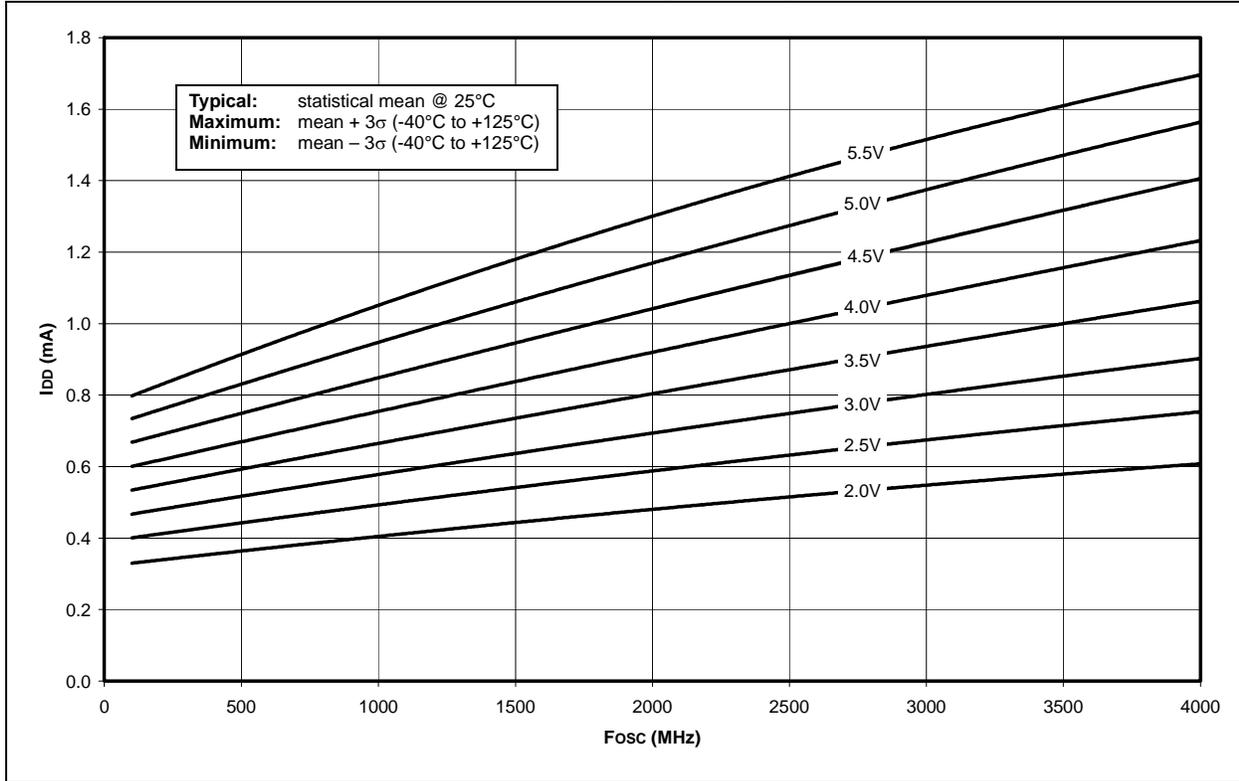
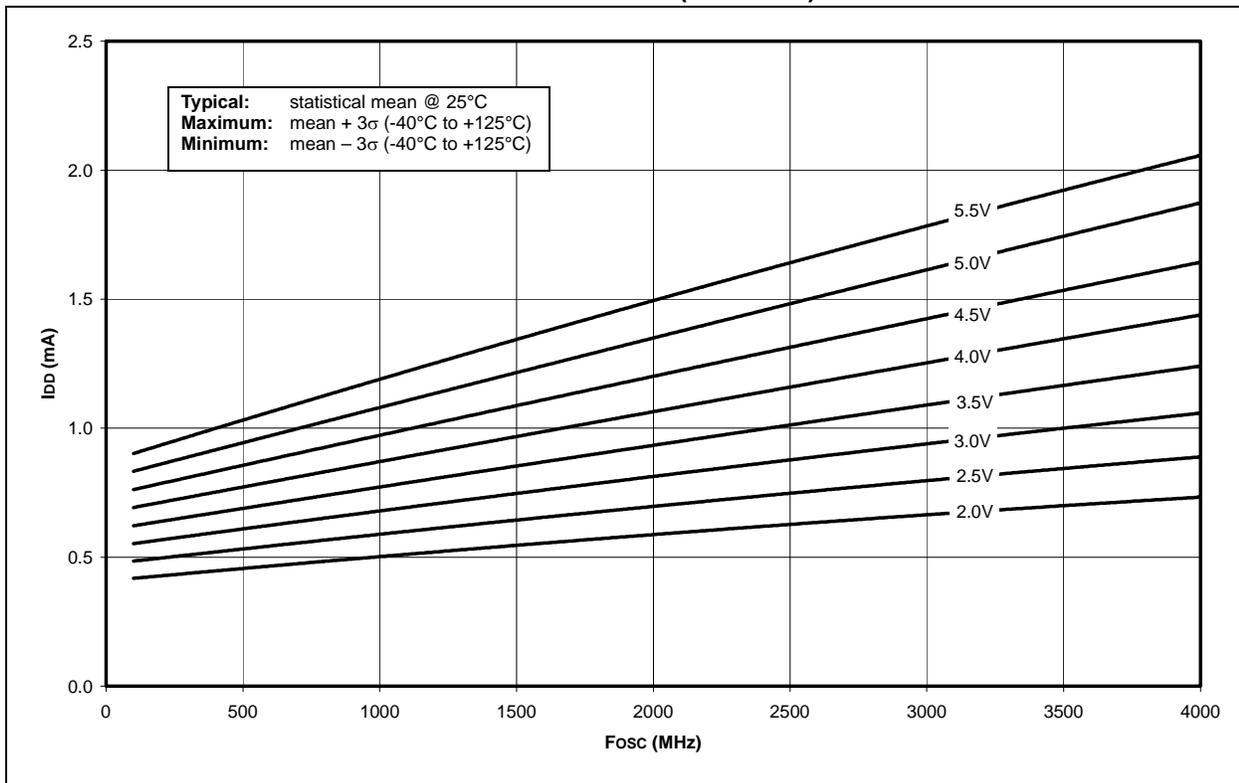


FIGURE 19-4: MAXIMUM  $I_{DD}$  vs.  $F_{osc}$  OVER  $V_{DD}$  (XT MODE)



# PIC16F7X7

FIGURE 19-11: AVERAGE  $F_{osc}$  vs.  $V_{DD}$  FOR VARIOUS VALUES OF R (RC MODE,  $C = 20\text{ pF}$ ,  $+25^\circ\text{C}$ )

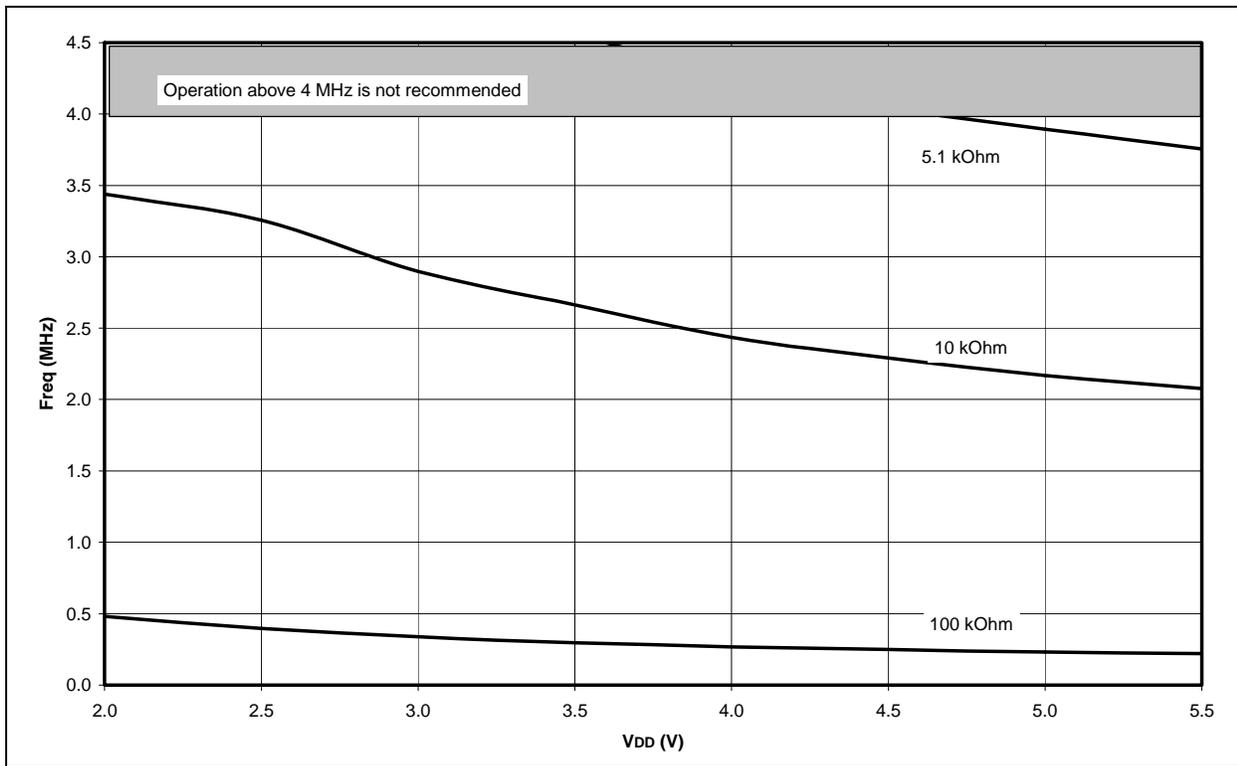
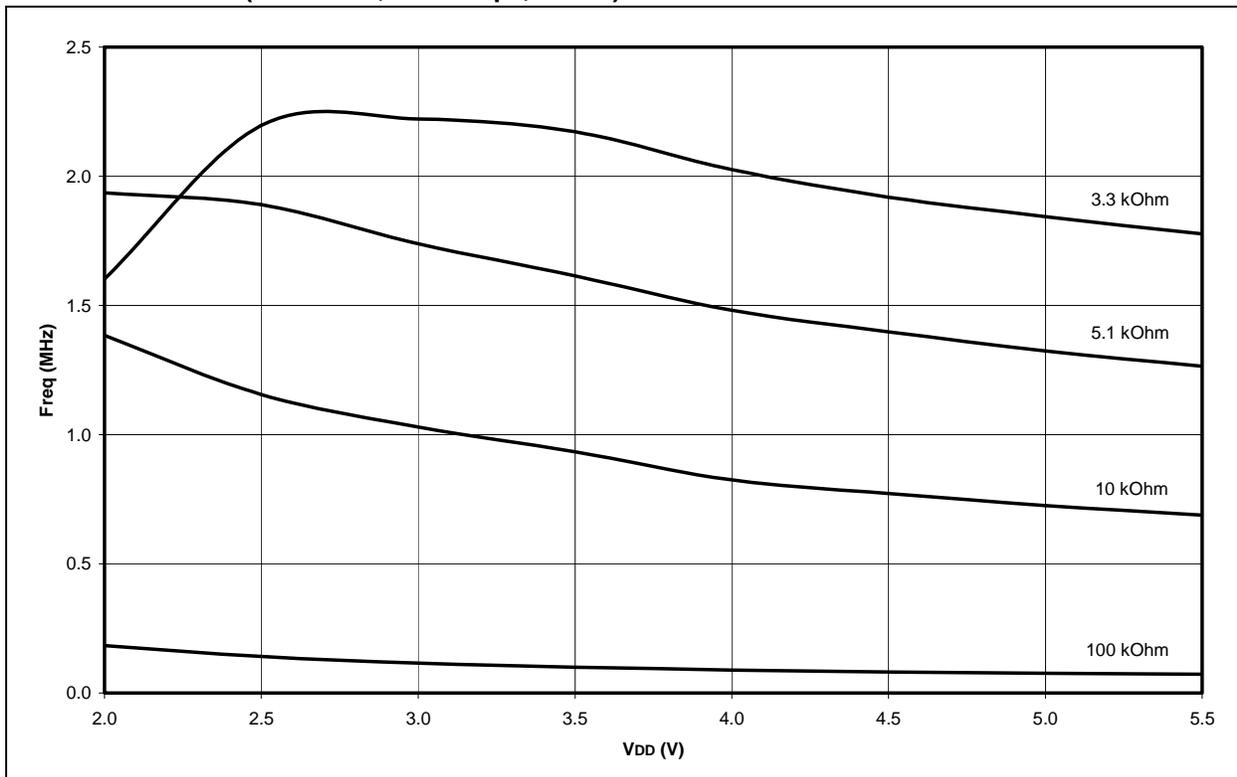


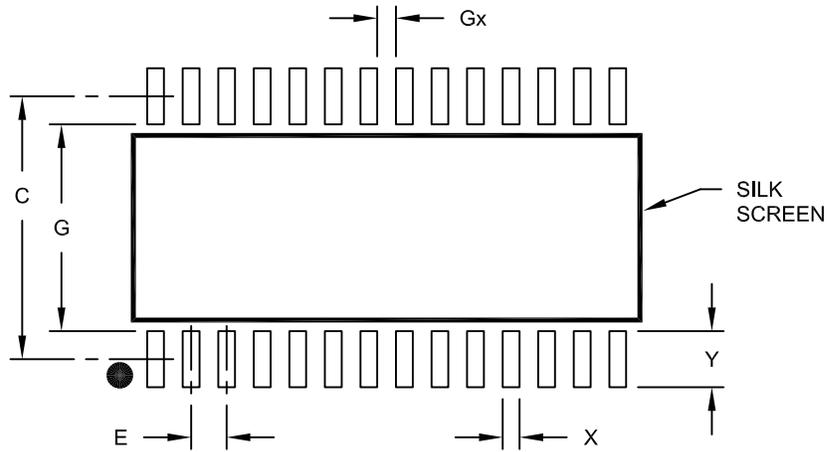
FIGURE 19-12: AVERAGE  $F_{osc}$  vs.  $V_{DD}$  FOR VARIOUS VALUES OF R (RC MODE,  $C = 100\text{ pF}$ ,  $+25^\circ\text{C}$ )



# PIC16F7X7

28-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>



## RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	C		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2052A

# PIC16F7X7

---

NOTES: