Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
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	FLASH
EEPROM Size	256 x 8
RAM Size	368 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 5x10b
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f876-04-so

PIC16F87X

Pin Diagrams
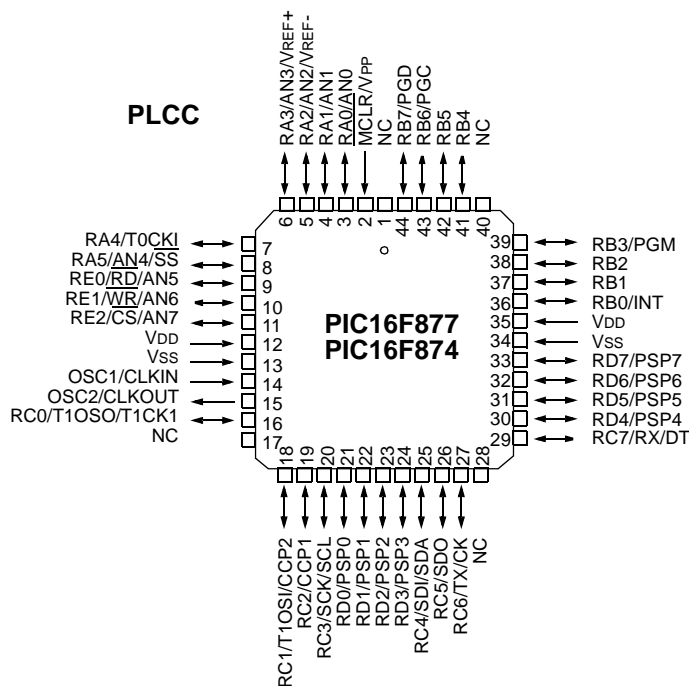
PDIP, SOIC

PIC16F876/873

Pin	Signal		Pin	Signal
1	MCLR/VPP		28	RB7/PGD
2	RA0/AN0		27	RB6/PGC
3	RA1/AN1		26	RB5
4	RA2/AN2/VREF-		25	RB4
5	RA3/AN3/VREF+		24	RB3/PGM
6	RA4/T0CKI		23	RB2
7	RA5/AN4/SS		22	RB1
8	VSS		21	RB0/INT
9	OSC1/CLKIN		20	VDD
10	OSC2/CLKOUT		19	VSS
11	RC0/T1OSO/T1CKI		18	RC7/RX/DT
12	RC1/T1OSI/CCP2		17	RC6/TX/CK
13	RC2/CCP1		16	RC5/SDO
14	RC3/SCK/SCL		15	RC4/SDI/SDA

PLCC

PIC16F877
PIC16F874

Top pins (6,5,4,3,2,1,44,43,42,41,40): RA3/AN3/VREF+, RA2/AN2/VREF-, RA1/AN1, RA0/AN0, MCLR/VPP, NC, RB7/PGD, RB6/PGC, RB5, RB4, NC

Left side:
- 7 RA4/T0CKI
- 8 RA5/AN4/SS
- 9 RE0/RD/AN5
- 10 RE1/WR/AN6
- 11 RE2/CS/AN7
- 12 VDD
- 13 VSS
- 14 OSC1/CLKIN
- 15 OSC2/CLKOUT
- 16 RC0/T1OSO/T1CK1
- 17 NC

Right side:
- 39 RB3/PGM
- 38 RB2
- 37 RB1
- 36 RB0/INT
- 35 VDD
- 34 VSS
- 33 RD7/PSP7
- 32 RD6/PSP6
- 31 RD5/PSP5
- 30 RD4/PSP4
- 29 RC7/RX/DT

Bottom pins (18,19,20,21,22,23,24,25,26,27,28): RC1/T1OSI/CCP2, RC2/CCP1, RC3/SCK/SCL, RD0/PSP0, RD1/PSP1, RD2/PSP2, RD3/PSP3, RC4/SDI/SDA, RC5/SDO, RC6/TX/CK, NC

QFP

PIC16F877
PIC16F874

Top pins (44,43,42,41,40,39,38,37,36,35,34): RC6/TX/CK, RC5/SDO, RC4/SDI/SDA, RD3/PSP3, RD2/PSP2, RD1/PSP1, RD0/PSP0, RC3/SCK/SCL, RC2/CCP1, RC1/T1OSI/CCP2, NC

Left side:
- 1 RC7/RX/DT
- 2 RD4/PSP4
- 3 RD5/PSP5
- 4 RD6/PSP6
- 5 RD7/PSP7
- 6 VSS
- 7 VDD
- 8 RB0/INT
- 9 RB1
- 10 RB2
- 11 RB3/PGM

Right side:
- 33 NC
- 32 RC0/T1OSO/T1CKI
- 31 OSC2/CLKOUT
- 30 OSC1/CLKIN
- 29 VSS
- 28 VDD
- 27 RE2/AN7/CS
- 26 RE1/AN6/WR
- 25 RE0/AN5/RD
- 24 RA5/AN4/SS
- 23 RA4/T0CKI

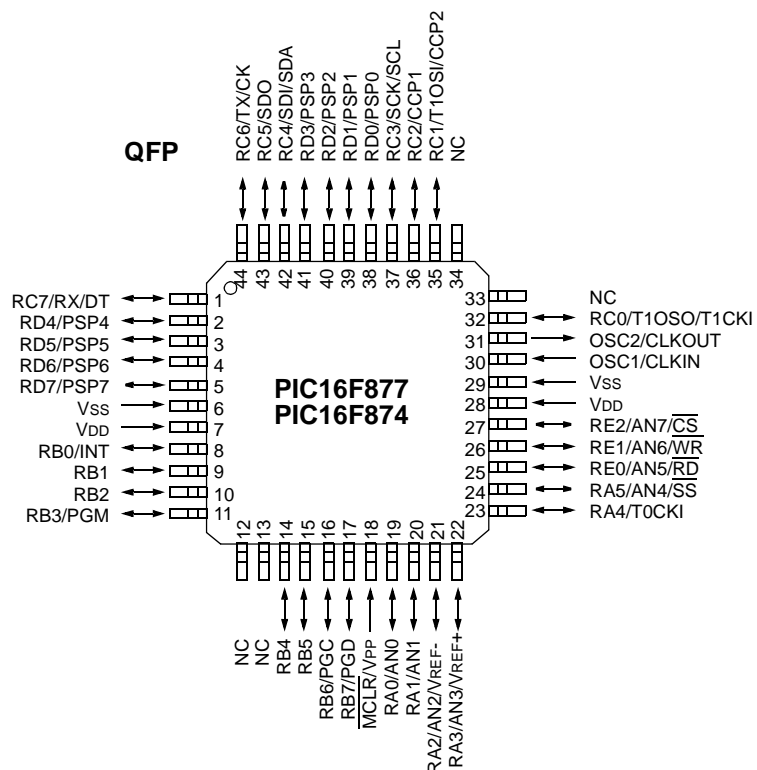Bottom pins (12,13,14,15,16,17,18,19,20,21,22): NC, NC, RB4, RB5, RB6/PGC, RB7/PGD, MCLR/VPP, RA0/AN0, RA1/AN1, RA2/AN2/VREF-, RA3/AN3/VREF+

2.0 MEMORY ORGANIZATION

There are three memory blocks in each of the PIC16F87X MCUs. The Program Memory and Data Memory have separate buses so that concurrent access can occur and is detailed in this section. The EEPROM data memory block is detailed in Section 4.0.

Additional information on device memory may be found in the PIC® MCU Mid-Range Reference Manual, (DS33023).

2.1 Program Memory Organization

The PIC16F87X devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. The PIC16F877/876 devices have 8K x 14 words of FLASH program memory, and the PIC16F873/874 devices have 4K x 14. Accessing a location above the physically implemented address will cause a wraparound.

The RESET vector is at 0000h and the interrupt vector is at 0004h.

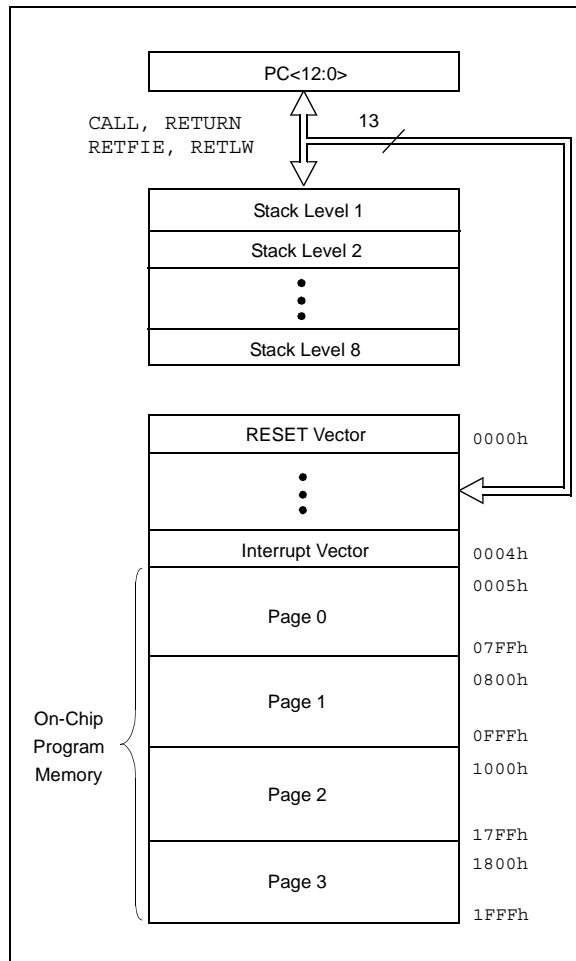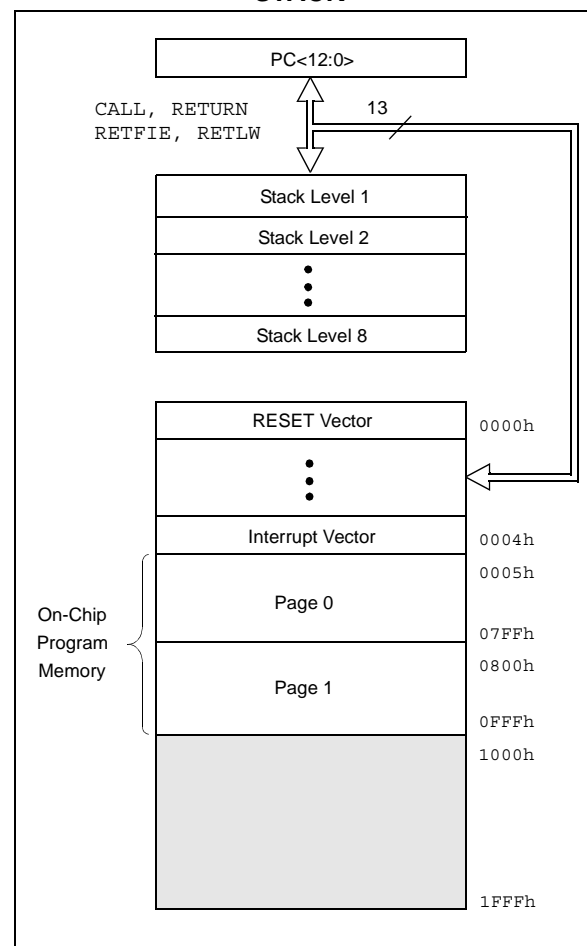**FIGURE 2-1:** **PIC16F877/876 PROGRAM MEMORY MAP AND STACK**

FIGURE 2-2: **PIC16F874/873 PROGRAM MEMORY MAP AND STACK**

2.2.2.2 OPTION_REG Register

The OPTION_REG Register is a readable and writable register, which contains various control bits to configure the TMR0 prescaler/WDT postscaler (single assign- able register known also as the prescaler), the External INT Interrupt, TMR0 and the weak pull-ups on PORTB.

> **Note:** To achieve a 1:1 prescaler assignment for the TMR0 register, assign the prescaler to the Watchdog Timer.

REGISTER 2-2:   OPTION_REG REGISTER (ADDRESS 81h, 181h)

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

bit 7   bit 0

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

1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

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

1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin

bit 5   **T0CS**: TMR0 Clock Source Select bit

1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4   **T0SE**: TMR0 Source Edge Select bit

1 = Increment on high-to-low transition on RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/T0CKI pin

bit 3   **PSA**: Prescaler Assignment bit

1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0   **PS2:PS0**: Prescaler Rate Select bits

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

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

> **Note:** When using low voltage ICSP programming (LVP) and the pull-ups on PORTB are enabled, bit 3 in the TRISB register must be cleared to disable the pull-up on RB3 and ensure the proper oper- ation of the device

2.2.2.6   PIE2 Register

The PIE2 register contains the individual enable bits for the CCP2 peripheral interrupt, the SSP bus collision interrupt, and the EEPROM write operation interrupt.

REGISTER 2-6:   PIE2 REGISTER (ADDRESS 8Dh)

U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
—	Reserved	—	EEIE	BCLIE	—	—	CCP2IE

bit 7   bit 0

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

bit 6   **Reserved**: Always maintain this bit clear

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

bit 4   **EEIE**: EEPROM Write Operation Interrupt Enable

            1 = Enable EE Write Interrupt
            0 = Disable EE Write Interrupt

bit 3   **BCLIE**: Bus Collision Interrupt Enable

            1 = Enable Bus Collision Interrupt
            0 = Disable Bus Collision Interrupt

bit 2-1   **Unimplemented:** Read as '0'

bit 0   **CCP2IE**: CCP2 Interrupt Enable bit

            1 = Enables the CCP2 interrupt
            0 = Disables the CCP2 interrupt

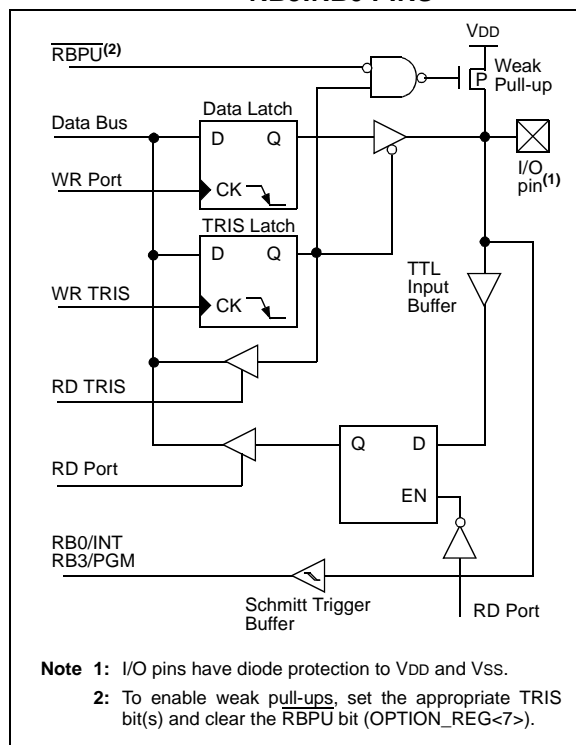| Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

3.2 PORTB and the TRISB Register

PORTB is an 8-bit wide, bi-directional 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 Hi-Impedance 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).

Three pins of PORTB are multiplexed with the Low Voltage Programming function: RB3/PGM, RB6/PGC and RB7/PGD. The alternate functions of these pins are described in the Special Features Section.

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 3-3: **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 OR'ed 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.
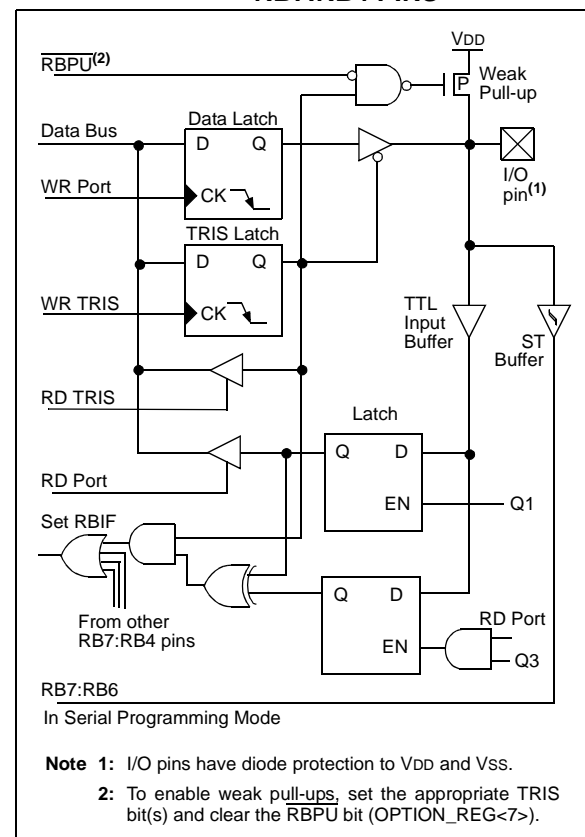
The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

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 Strokes"* (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 12.10.1.

FIGURE 3-4: **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>).

PIC16F87X

REGISTER 9-3: **SSPCON2: SYNC SERIAL PORT CONTROL REGISTER2 (ADDRESS 91h)**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN

bit 7   bit 0

bit 7   **GCEN**: General Call Enable bit (In I^{2}C Slave mode only)
           1 = Enable interrupt when a general call address (0000h) is received in the SSPSR
           0 = General call address disabled

bit 6   **ACKSTAT**: Acknowledge Status bit (In I^{2}C Master mode only)
           <u>In Master Transmit mode:</u>
           1 = Acknowledge was not received from slave
           0 = Acknowledge was received from slave

bit 5   **ACKDT**: Acknowledge Data bit (In I^{2}C Master mode only)
           <u>In Master Receive mode:</u>
           Value that will be transmitted when the user initiates an Acknowledge sequence at the
           end of a receive.
           1 = Not Acknowledge
           0 = Acknowledge

bit 4   **ACKEN**: Acknowledge Sequence Enable bit (In I^{2}C Master mode only)
           <u>In Master Receive mode:</u>
           1 = Initiate Acknowledge sequence on SDA and SCL pins and transmit ACKDT data bit.
               Automatically cleared by hardware.
           0 = Acknowledge sequence idle

bit 3   **RCEN**: Receive Enable bit (In I^{2}C Master mode only)
           1 = Enables Receive mode for I^{2}C
           0 = Receive idle

bit 2   **PEN**: STOP Condition Enable bit (In I^{2}C Master mode only)
           <u>SCK Release Control:</u>
           1 = Initiate STOP condition on SDA and SCL pins. Automatically cleared by hardware.
           0 = STOP condition idle

bit 1   **RSEN**: Repeated START Condition Enable bit (In I^{2}C Master mode only)
           1 = Initiate Repeated START condition on SDA and SCL pins. Automatically cleared by hardware.
           0 = Repeated START condition idle

bit 0   **SEN**: START Condition Enable bit (In I^{2}C Master mode only)
           1 = Initiate START condition on SDA and SCL pins. Automatically cleared by hardware.
           0 = START condition idle

Note: For bits ACKEN, RCEN, PEN, RSEN, SEN: If the I^{2}C module is not in the IDLE
             mode, this bit may not be set (no spooling), and the SSPBUF may not be written (or
             writes to the SSPBUF are disabled).

| 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

9.2   MSSP I^{2}C Operation

The MSSP module in I^{2}C mode, fully implements all master and slave functions (including general call support) and provides interrupts on START and STOP bits in hardware, to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Refer to Application Note AN578, *"Use of the SSP Module in the I^{2}C Multi-Master Environment."*

A "glitch" filter is on the SCL and SDA pins when the pin is an input. This filter operates in both the 100 kHz and 400 kHz modes. In the 100 kHz mode, when these pins are an output, there is a slew rate control of the pin that is independent of device frequency.

FIGURE 9-5:   I^{2}C SLAVE MODE BLOCK DIAGRAM
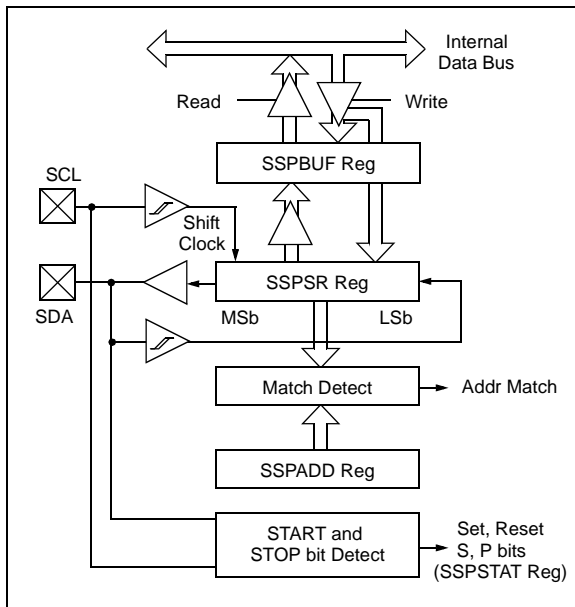
Two pins are used for data transfer. These are the SCL pin, which is the clock, and the SDA pin, which is the data. The SDA and SCL pins are automatically configured when the I^{2}C mode is enabled. The SSP module functions are enabled by setting SSP Enable bit SSPEN (SSPCON<5>).

The MSSP module has six registers for I^{2}C operation. They are the:

• SSP Control Register (SSPCON)
• SSP Control Register2 (SSPCON2)
• SSP Status Register (SSPSTAT)
• Serial Receive/Transmit Buffer (SSPBUF)
• SSP Shift Register (SSPSR) - Not directly accessible
• SSP Address Register (SSPADD)

The SSPCON register allows control of the I^{2}C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I^{2}C modes to be selected:

• I^{2}C Slave mode (7-bit address)
• I^{2}C Slave mode (10-bit address)
• I^{2}C Master mode, clock = OSC/4 (SSPADD +1)
• I^{2}C firmware modes (provided for compatibility to other mid-range products)

Before selecting any I^{2}C mode, the SCL and SDA pins must be programmed to inputs by setting the appropriate TRIS bits. Selecting an I^{2}C mode by setting the SSPEN bit, enables the SCL and SDA pins to be used as the clock and data lines in I^{2}C mode. Pull-up resistors must be provided externally to the SCL and SDA pins for the proper operation of the I^{2}C module.

The CKE bit (SSPSTAT<6:7>) sets the levels of the SDA and SCL pins in either Master or Slave mode. When CKE = 1, the levels will conform to the SMBus specification. When CKE = 0, the levels will conform to the I^{2}C specification.

The SSPSTAT register gives the status of the data transfer. This information includes detection of a START (S) or STOP (P) bit, specifies if the received byte was data or address, if the next byte is the completion of 10-bit address, and if this will be a read or write data transfer.

SSPBUF is the register to which the transfer data is written to, or read from. The SSPSR register shifts the data in or out of the device. In receive operations, the SSPBUF and SSPSR create a doubled buffered receiver. This allows reception of the next byte to begin before reading the last byte of received data. When the complete byte is received, it is transferred to the SSPBUF register and flag bit SSPIF is set. If another complete byte is received before the SSPBUF register is read, a receiver overflow has occurred and bit SSPOV (SSPCON<6>) is set and the byte in the SSPSR is lost.

The SSPADD register holds the slave address. In 10-bit mode, the user needs to write the high byte of the address (`1111 0 A9 A8 0`). Following the high byte address match, the low byte of the address needs to be loaded (A7:A0).

PIC16F87X

FIGURE 9-7: I²C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)
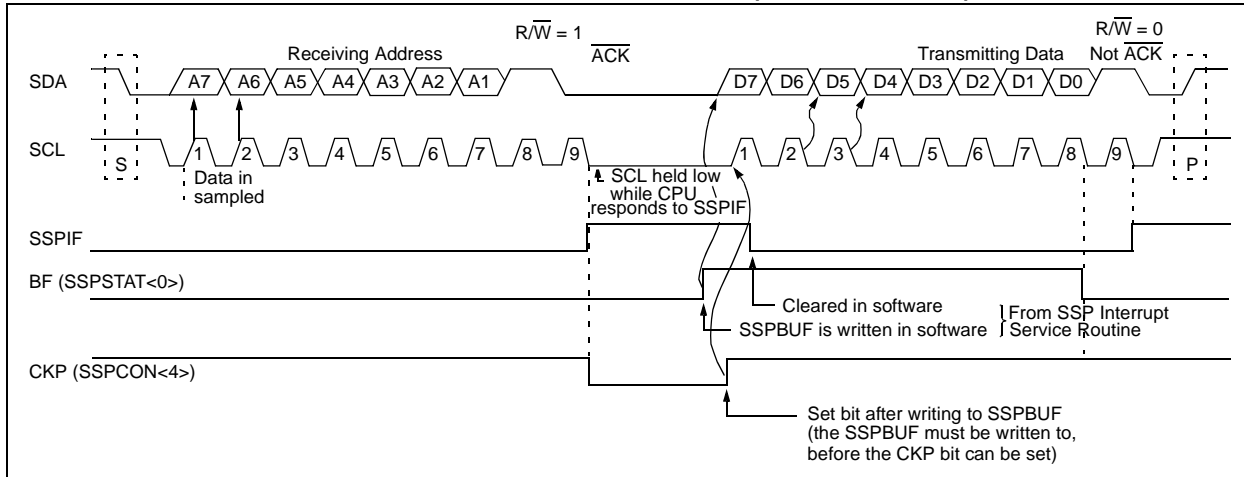
9.2.2 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I²C bus is such that the first byte after the START condition usually determines which device will be the slave addressed by the master. The exception is the general call address, which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

The general call address is one of eight addresses reserved for specific purposes by the I²C protocol. It consists of all 0's with R/$\overline{W}$ = 0.

The general call address is recognized when the General Call Enable bit (GCEN) is enabled (SSPCON2<7> is set). Following a START bit detect, 8 bits are shifted into SSPSR and the address is compared against SSPADD. It is also compared to the general call address and fixed in hardware.

If the general call address matches, the SSPSR is transferred to the SSPBUF, the BF flag is set (eighth bit), and on the falling edge of the ninth bit ($\overline{ACK}$ bit), the SSPIF flag is set.

When the interrupt is serviced, the source for the interrupt can be checked by reading the contents of the SSPBUF to determine if the address was device specific, or a general call address.

In 10-bit mode, the SSPADD is required to be updated for the second half of the address to match, and the UA bit is set (SSPSTAT<1>). If the general call address is sampled when GCEN is set, while the slave is configured in 10-bit address mode, then the second half of the address is not necessary, the UA bit will not be set, and the slave will begin receiving data after the Acknowledge (Figure 9-8).

FIGURE 9-8: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE (7 OR 10-BIT MODE)
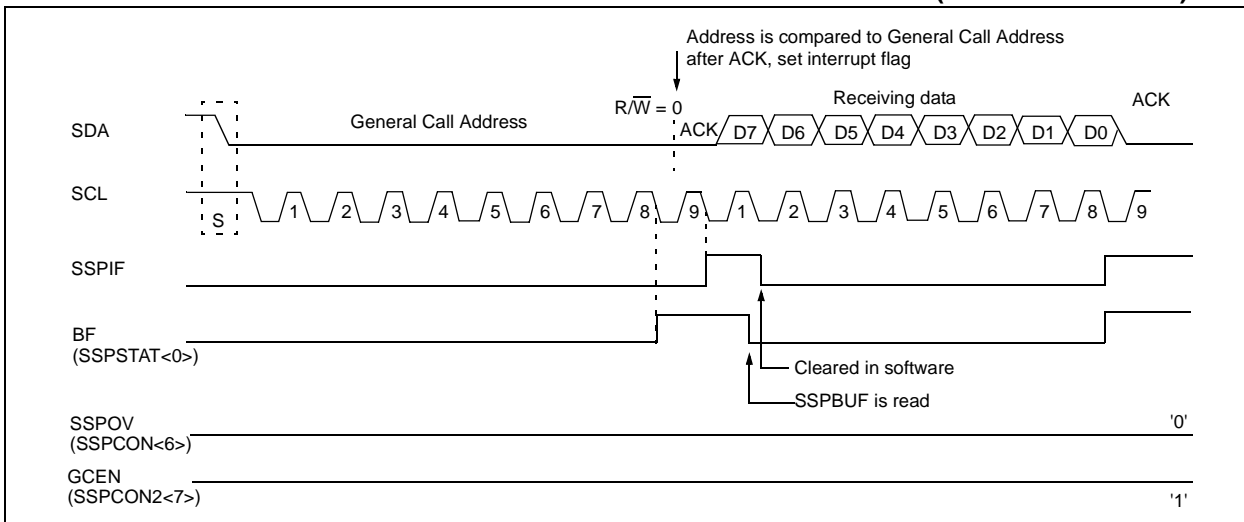
9.2.7 I²C MASTER MODE SUPPORT

Master mode is enabled by setting and clearing the appropriate SSPM bits in SSPCON and by setting the SSPEN bit. Once Master mode is enabled, the user has six options:

• Assert a START condition on SDA and SCL.
• Assert a Repeated START condition on SDA and SCL.
• Write to the SSPBUF register initiating transmission of data/address.
• Generate a STOP condition on SDA and SCL.
• Configure the I²C port to receive data.
• Generate an Acknowledge condition at the end of a received byte of data.

> **Note:** The MSSP Module, when configured in I²C Master mode, does not allow queueing 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.

9.2.7.1 I²C Master Mode Operation

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

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/W̄) bit. In this case, the R/W̄ bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. START and STOP conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/W̄ bit. In this case, the R/W̄ bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. START and STOP conditions indicate the beginning and end of transmission.

The baud rate generator used for SPI mode operation is now used to set the SCL clock frequency for either 100 kHz, 400 kHz, or 1 MHz I²C operation. The baud rate generator reload value is contained in the lower 7 bits of the SSPADD register. The baud rate generator will automatically begin counting on a write to the

SSPBUF. Once the given operation is complete (i.e., transmission of the last data bit is followed by ACK), the internal clock will automatically stop counting and the SCL pin will remain in its last state.

A typical transmit sequence would go as follows:

a) User generates a START condition by setting the START enable bit (SEN) in SSPCON2.
b) SSPIF is set. The module will wait the required start time before any other operation takes place.
c) User loads SSPBUF with address to transmit.
d) Address is shifted out the SDA pin until all 8 bits are transmitted.
e) MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
f) MSSP module generates an interrupt at the end of the ninth clock cycle by setting SSPIF.
g) User loads SSPBUF with eight bits of data.
h) DATA is shifted out the SDA pin until all 8 bits are transmitted.
i) MSSP module shifts in the ACK bit from the slave device, and writes its value into the SSPCON2 register (SSPCON2<6>).
j) MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
k) User generates a STOP condition by setting the STOP enable bit, PEN, in SSPCON2.
l) Interrupt is generated once the STOP condition is complete.

9.2.8 BAUD RATE GENERATOR

In I²C Master mode, the reload value for the BRG is located in the lower 7 bits of the SSPADD register (Figure 9-10). When the BRG is loaded with this value, the BRG counts down to 0 and stops until another reload has taken place. The BRG count is decremented twice per instruction cycle (TCY), on the Q2 and Q4 clock.

In I²C Master mode, the BRG is reloaded automatically. If clock arbitration is taking place, the BRG will be reloaded when the SCL pin is sampled high (Figure 9-11).

> **Note:** Baud Rate = FOSC / (4 * (SSPADD + 1))

FIGURE 9-10: BAUD RATE GENERATOR BLOCK DIAGRAM
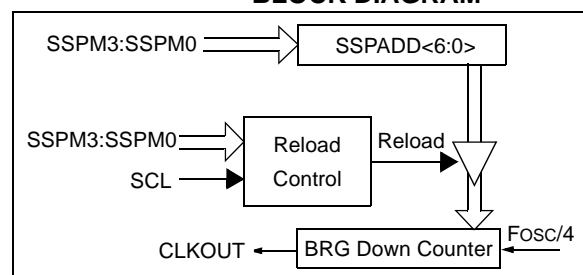
9.2.14 STOP CONDITION TIMING

A STOP bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN (SSPCON2<2>). At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the baud rate generator is reloaded and counts down to 0. When the baud rate generator times out, the SCL pin will be brought high, and one TBRG (baud rate generator rollover count) later, the SDA pin will be de-asserted. When the SDA pin is sampled high while SCL is high, the P bit (SSPSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 9-17).

Whenever the firmware decides to take control of the bus, it will first determine if the bus is busy by checking the S and P bits in the SSPSTAT register. If the bus is busy, then the CPU can be interrupted (notified) when a STOP bit is detected (i.e., bus is free).

9.2.14.1 WCOL Status Flag

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

FIGURE 9-17: STOP CONDITION RECEIVE OR TRANSMIT MODE
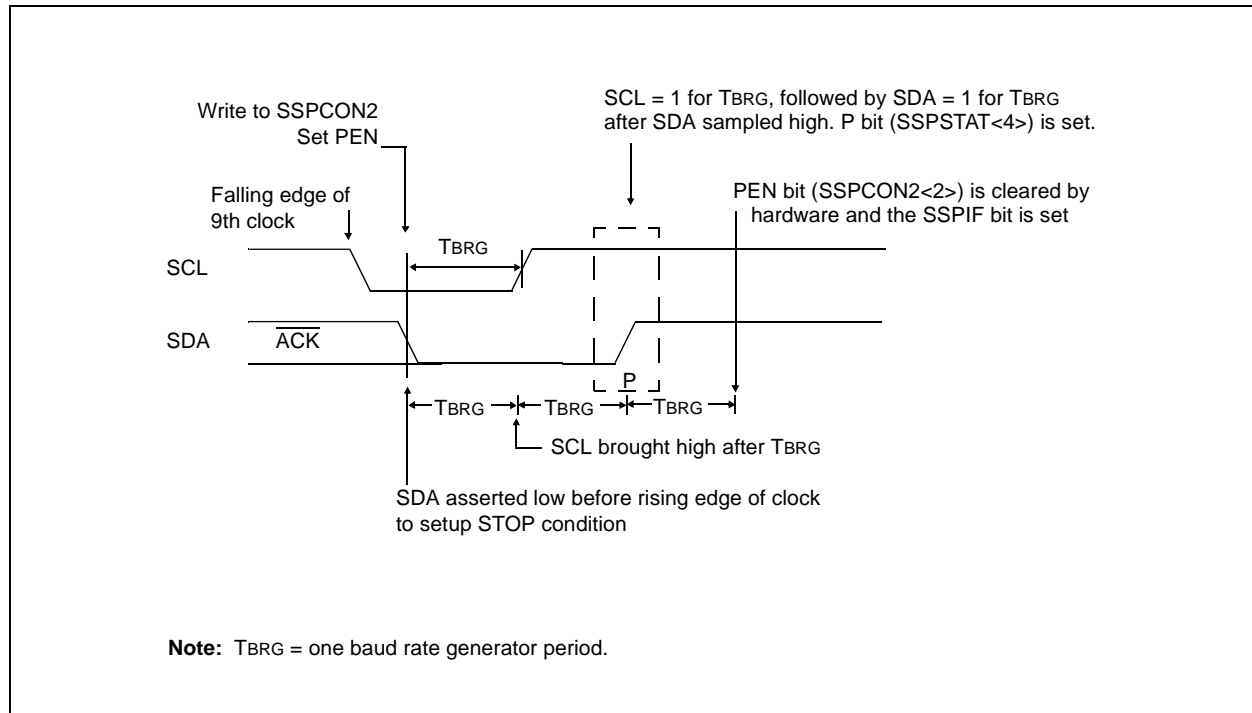
PIC16F87X

9.2.18.2 Bus Collision During a Repeated START Condition

During a Repeated START condition, a bus collision occurs if:

a) A low level is sampled on SDA when SCL goes from low level to high level.

b) SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1'.

When the user de-asserts SDA and the pin is allowed to float high, the BRG is loaded with SSPADD<6:0> and counts down to 0. The SCL pin is then de-asserted, and when sampled high, the SDA pin is sampled. If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data'0'). If, however, SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high to low before the BRG times out, no bus collision occurs, because no two masters can assert SDA at exactly the same time.

If, however, SCL goes from high to low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data'1' during the Repeated START condition.

If at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low, the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated START condition is complete (Figure 9-23).

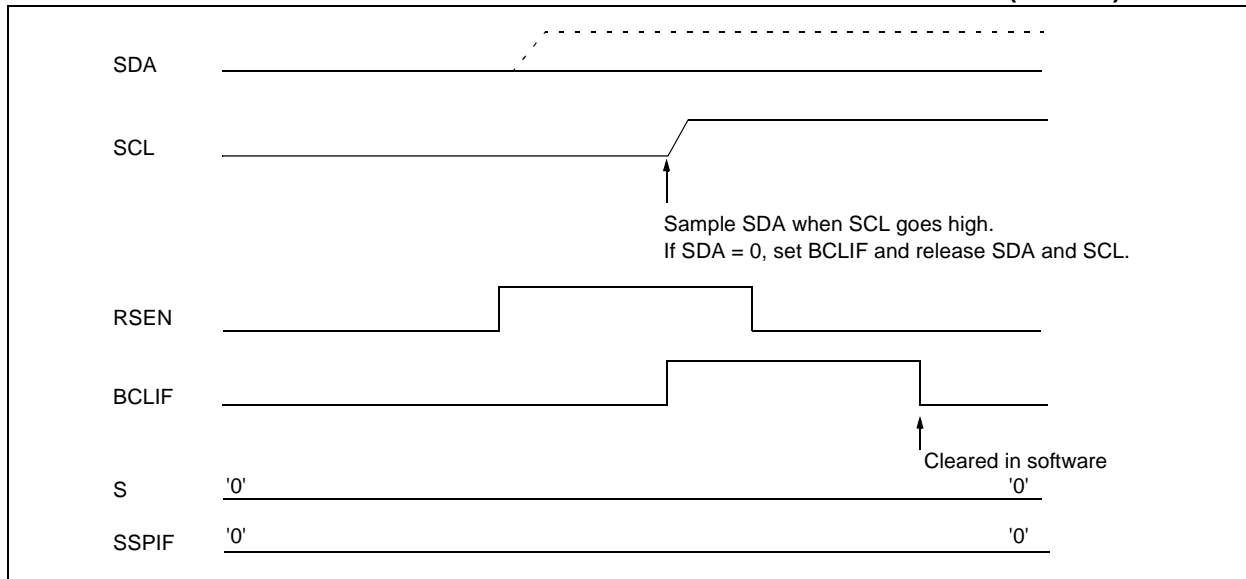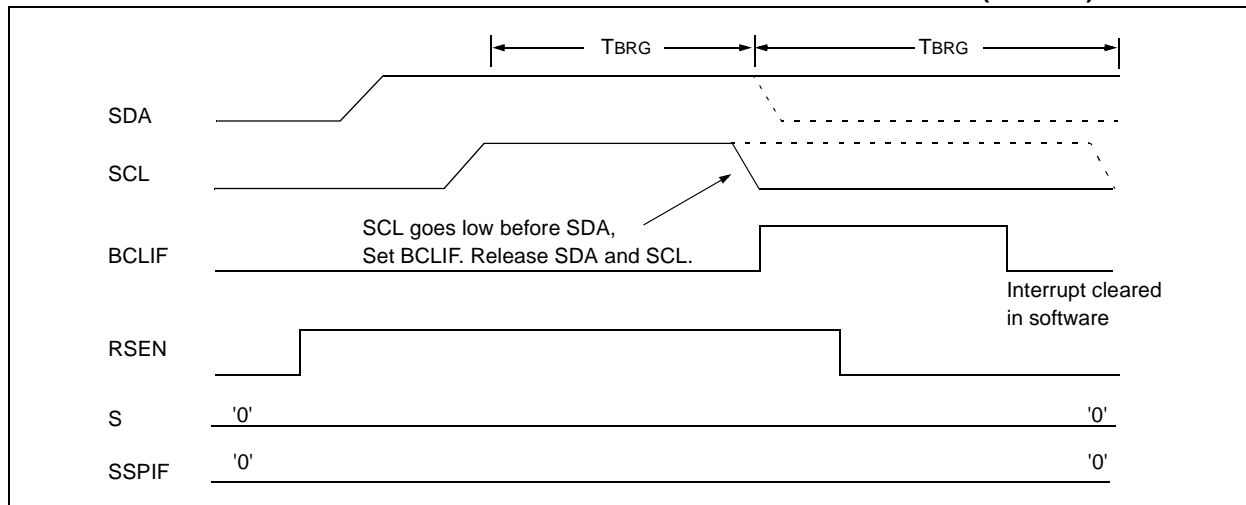**FIGURE 9-23:** **BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)**

FIGURE 9-24: **BUS COLLISION DURING REPEATED START CONDITION (CASE 2)**

10.2.2 USART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 10-4. The data is received on the RC7/RX/DT pin and drives the data recovery block. The data recovery block is actually a high speed shifter, operating at x16 times the baud rate; whereas, the main receive serial shifter operates at the bit rate or at F$_{OSC}$.

Once Asynchronous mode is selected, reception is enabled by setting bit CREN (RCSTA<4>).

The heart of the receiver is the receive (serial) shift register (RSR). After sampling the STOP bit, the received data in the RSR is transferred to the RCREG register (if it is empty). If the transfer is complete, flag bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit RCIE (PIE1<5>). Flag bit RCIF is a read only bit, which is cleared by the hardware. It is cleared when the RCREG register has been read and is empty. The RCREG is a double buffered register (i.e., it is a two deep FIFO). It

is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting to the RSR register. On the detection of the STOP bit of the third byte, if the RCREG register is still full, the overrun error bit OERR (RCSTA<1>) will be set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Overrun bit OERR has to be cleared in software. This is done by resetting the receive logic (CREN is cleared and then set). If bit OERR is set, transfers from the RSR register to the RCREG register are inhibited, and no further data will be received. It is therefore, essential to clear error bit OERR if it is set. Framing error bit FERR (RCSTA<2>) is set if a STOP bit is detected as clear. Bit FERR and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG will load bits RX9D and FERR with new values, therefore, it is essential for the user to read the RCSTA register before reading the RCREG register in order not to lose the old FERR and RX9D information.

FIGURE 10-4: USART RECEIVE BLOCK DIAGRAM
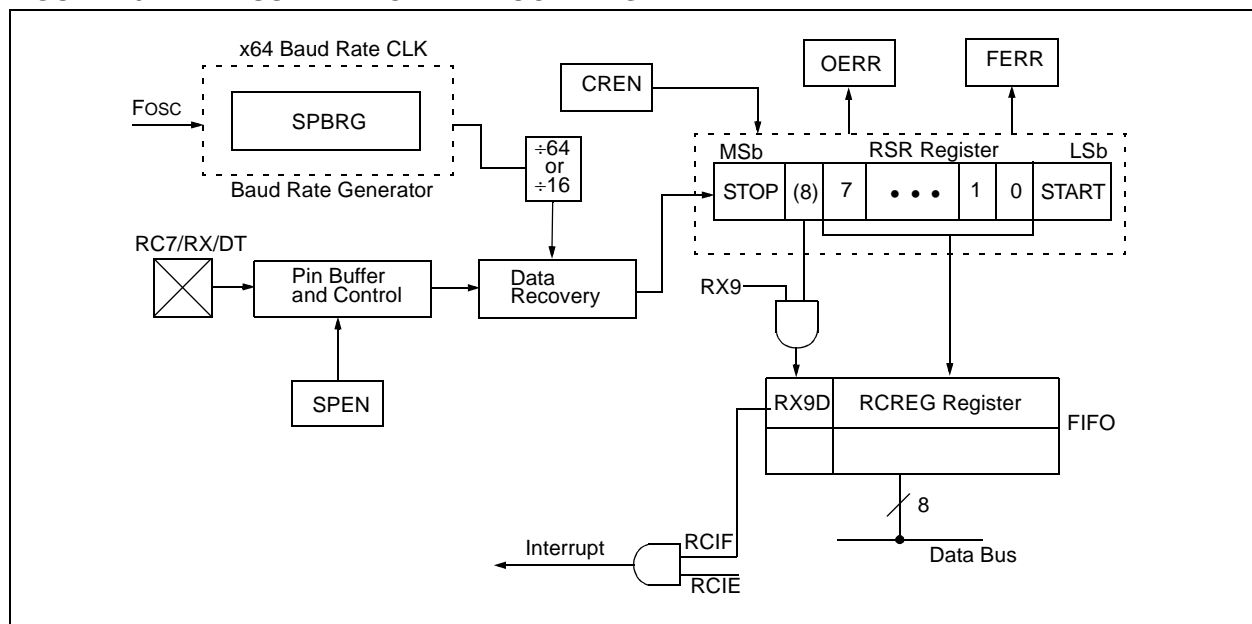
PIC16F87X

12.4 Power-On Reset (POR)

A Power-on Reset pulse is generated on-chip when VDD rise is detected (in the range of 1.2V - 1.7V). To take advantage of the POR, tie the MCLR pin directly (or through a resistor) to VDD. This will eliminate external RC components usually needed to create a Power-on Reset. A maximum rise time for VDD is specified. See Electrical Specifications for details.

When the device starts normal operation (exits the RESET condition), device operating parameters (voltage, frequency, temperature,...) must be met to ensure operation. If these conditions are not met, the device must be held in RESET until the operating conditions are met. Brown-out Reset may be used to meet the start-up conditions. For additional information, refer to Application Note, AN007, "Power-up Trouble Shooting", (DS00007).

12.5 Power-up Timer (PWRT)

The Power-up Timer provides a fixed 72 ms nominal time-out on power-up only from the POR. The Power-up Timer operates on an internal RC oscillator. The chip is kept in RESET as long as the PWRT is active. The PWRT's time delay allows VDD to rise to an acceptable level. A configuration bit is provided to enable/disable the PWRT.

The power-up time delay will vary from chip to chip due to VDD, temperature and process variation. See DC parameters for details (TPWRT, parameter #33).

12.6 Oscillator Start-up Timer (OST)

The Oscillator Start-up Timer (OST) provides a delay of 1024 oscillator cycles (from OSC1 input) after the PWRT delay is over (if PWRT is enabled). This helps to ensure that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP and HS modes and only on Power-on Reset or Wake-up from SLEEP.

12.7 Brown-out Reset (BOR)

The configuration bit, BODEN, can enable or disable the Brown-out Reset circuit. If VDD falls below VBOR (parameter D005, about 4V) for longer than TBOR (parameter #35, about 100µS), the brown-out situation will reset the device. If VDD falls below VBOR for less than TBOR, a RESET may not occur.

Once the brown-out occurs, the device will remain in Brown-out Reset until VDD rises above VBOR. The Power-up Timer then keeps the device in RESET for TPWRT (parameter #33, about 72mS). If VDD should fall below VBOR during TPWRT, the Brown-out Reset process will restart when VDD rises above VBOR with the Power-up Timer Reset. The Power-up Timer is always enabled when the Brown-out Reset circuit is enabled, regardless of the state of the PWRT configuration bit.

12.8 Time-out Sequence

On power-up, the time-out sequence is as follows: The PWRT delay starts (if enabled) when a POR Reset 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 MCLR is kept low long enough, the time-outs will expire. Bringing MCLR high will begin execution immediately. This is useful for testing purposes or to synchronize more than one PIC16F87X device operating in parallel.

Table 12-5 shows the RESET conditions for the STATUS, PCON and PC registers, while Table 12-6 shows the RESET conditions for all the registers.

12.9 Power Control/Status Register (PCON)

The Power Control/Status Register, PCON, has up to two bits depending upon the device.

Bit0 is Brown-out Reset Status bit, BOR. Bit 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 BOR cleared, indicating a BOR occurred. When the Brown-out Reset is disabled, the state of the BOR bit is unpredictable and is, therefore, not valid at any time.

Bit1 is 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
	PWRTE = 0	PWRTE = 1		
XT, HS, LP	72 ms + 1024TOSC	1024TOSC	72 ms + 1024TOSC	1024TOSC
RC	72 ms	—	72 ms	—

12.10 Interrupts

The PIC16F87X family has up to 14 sources of interrupt. The interrupt control register (INTCON) records individual interrupt requests in flag bits. It also has individual and global interrupt enable bits.

> **Note:** Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit, or the GIE bit.

A global interrupt enable bit, GIE (INTCON<7>) enables (if set) all unmasked interrupts, or disables (if cleared) all interrupts. When bit GIE is enabled, and an interrupt's flag bit and mask bit are set, the interrupt will vector immediately. Individual interrupts can be disabled through their corresponding enable bits in various registers. Individual interrupt bits are set, regardless of the status of the GIE bit. The GIE bit is cleared on RESET.

The "return from interrupt" instruction, RETFIE, exits the interrupt routine, as well as sets the GIE bit, which re-enables interrupts.
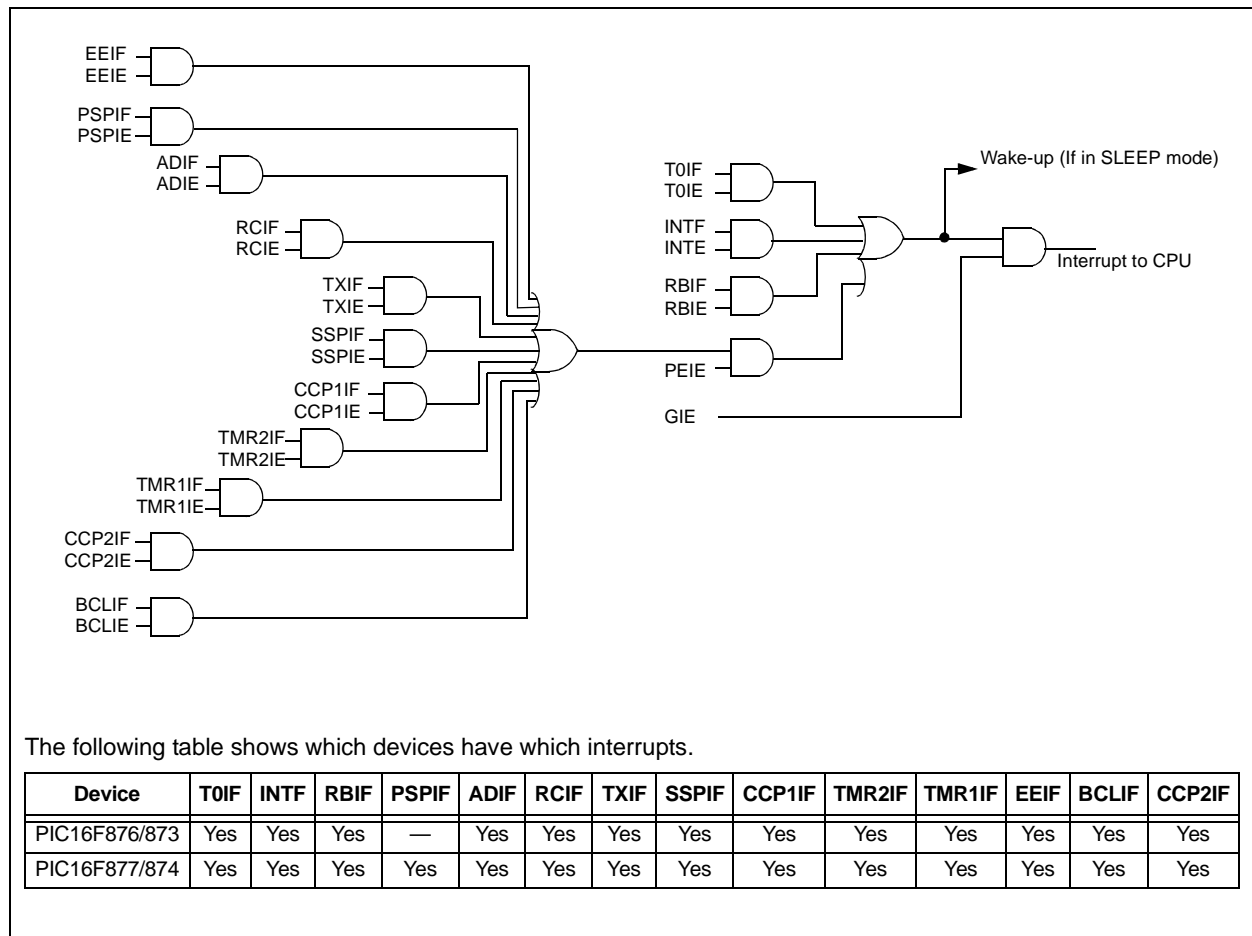
The RB0/INT pin interrupt, the RB port change interrupt, and the TMR0 overflow interrupt flags are contained in the INTCON register.

The peripheral interrupt flags are contained in the special function registers, PIR1 and PIR2. The corresponding interrupt enable bits are contained in special function registers, PIE1 and PIE2, and the peripheral interrupt enable bit is contained in special function register INTCON.

When an interrupt is responded to, the GIE bit is cleared to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with 0004h. Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

For external interrupt events, such as the INT pin or PORTB change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends when the interrupt event occurs. The latency is the same for one or two-cycle instructions. Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit, PEIE bit, or GIE bit.

FIGURE 12-9:   INTERRUPT LOGIC

The following table shows which devices have which interrupts.

Device	T0IF	INTF	RBIF	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	EEIF	BCLIF	CCP2IF
PIC16F876/873	Yes	Yes	Yes	—	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
PIC16F877/874	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

PIC16F87X

12.10.1 INT INTERRUPT

External interrupt on the RB0/INT pin is edge triggered, either rising, if bit INTEDG (OPTION_REG<6>) is set, or falling, if the INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, flag bit INTF (INTCON<1>) is set. This interrupt can be disabled by clearing enable bit INTE (INTCON<4>). Flag bit INTF must be cleared in software in the Interrupt Service Routine before re-enabling this interrupt. The INT interrupt can wake-up the processor from SLEEP, if bit INTE was set prior to going into SLEEP. The status of global interrupt enable bit, GIE, decides whether or not the processor branches to the interrupt vector following wake-up. See Section 12.13 for details on SLEEP mode.

12.10.2 TMR0 INTERRUPT

An overflow (FFh $\rightarrow$ 00h) in the TMR0 register will set flag bit T0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit T0IE (INTCON<5>) (Section 5.0).

12.10.3 PORTB INTCON CHANGE

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit RBIE (INTCON<4>) (Section 3.2).

12.11 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt, (i.e., W register and STATUS register). This will have to be implemented in software.

For the PIC16F873/874 devices, the register W_TEMP must be defined in both banks 0 and 1 and must be defined at the same offset from the bank base address (i.e., If W_TEMP is defined at 0x20 in bank 0, it must also be defined at 0xA0 in bank 1). The registers, PCLATH_TEMP and STATUS_TEMP, are only defined in bank 0.

Since the upper 16 bytes of each bank are common in the PIC16F876/877 devices, temporary holding registers W_TEMP, STATUS_TEMP, and PCLATH_TEMP should be placed in here. These 16 locations don't require banking and therefore, make it easier for context save and restore. The same code shown in Example 12-1 can be used.

EXAMPLE 12-1:   SAVING STATUS, W, AND PCLATH REGISTERS IN RAM

```
    MOVWF    W_TEMP            ;Copy W to TEMP register
    SWAPF    STATUS,W          ;Swap status to be saved into W
    CLRF     STATUS            ;bank 0, regardless of current bank, Clears IRP,RP1,RP0
    MOVWF    STATUS_TEMP       ;Save status to bank zero STATUS_TEMP register
    MOVF     PCLATH, W         ;Only required if using pages 1, 2 and/or 3
    MOVWF    PCLATH_TEMP       ;Save PCLATH into W
    CLRF     PCLATH            ;Page zero, regardless of current page
    :
    :(ISR)                     ;(Insert user code here)
    :
    MOVF     PCLATH_TEMP, W    ;Restore PCLATH
    MOVWF    PCLATH            ;Move W into PCLATH
    SWAPF    STATUS_TEMP,W     ;Swap STATUS_TEMP register into W
                              ;(sets bank to original state)
    MOVWF    STATUS            ;Move W into STATUS register
    SWAPF    W_TEMP,F          ;Swap W_TEMP
    SWAPF    W_TEMP,W          ;Swap W_TEMP into W
```

PIC16F87X

12.17   In-Circuit Serial Programming

PIC16F87X microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data and three other lines for power, ground, and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware, or a custom firmware to be programmed.

When using ICSP, the part must be supplied at 4.5V to 5.5V, if a bulk erase will be executed. This includes reprogramming of the code protect, both from an on-state to off-state. For all other cases of ICSP, the part may be programmed at the normal operating voltages. This means calibration values, unique user IDs, or user code can be reprogrammed or added.

For complete details of serial programming, please refer to the EEPROM Memory Programming Specification for the PIC16F87X (DS39025).

12.18   Low Voltage ICSP Programming

The LVP bit of the configuration word enables low voltage ICSP programming. This mode allows the microcontroller to be programmed via ICSP using a V_{DD} source in the operating voltage range. This only means that V_{PP} does not have to be brought to V_{IHH}, but can instead be left at the normal operating voltage. In this mode, the RB3/PGM pin is dedicated to the programming function and ceases to be a general purpose I/O pin. During programming, V_{DD} is applied to the $\overline{MCLR}$ pin. To enter Programming mode, V_{DD} must be applied to the RB3/PGM, provided the LVP bit is set. The LVP bit defaults to on ('1') from the factory.

> **Note 1:** The High Voltage Programming mode is always available, regardless of the state of the LVP bit, by applying V_{IHH} to the $\overline{MCLR}$ pin.
>
> **2:** While in Low Voltage ICSP mode, the RB3 pin can no longer be used as a general purpose I/O pin.
>
> **3:** When using low voltage ICSP programming (LVP) and the pull-ups on PORTB are enabled, bit 3 in the TRISB register must be cleared to disable the pull-up on RB3 and ensure the proper operation of the device.
>
> **4:** RB3 should not be allowed to float if LVP is enabled. An external pull-down device should be used to default the device to normal operating mode. If RB3 floats high, the PIC16F87X device will enter Programming mode.
>
> **5:** LVP mode is enabled by default on all devices shipped from Microchip. It can be disabled by clearing the LVP bit in the CONFIG register.
>
> **6:** Disabling LVP will provide maximum compatibility to other PIC16CXXX devices.

If Low Voltage Programming mode is not used, the LVP bit can be programmed to a '0' and RB3/PGM becomes a digital I/O pin. However, the LVP bit may only be programmed when programming is entered with V_{IHH} on $\overline{MCLR}$. The LVP bit can only be charged when using high voltage on $\overline{MCLR}$.

It should be noted, that once the LVP bit is programmed to 0, only the High Voltage Programming mode is available and only High Voltage Programming mode can be used to program the device.

When using low voltage ICSP, the part must be supplied at 4.5V to 5.5V, if a bulk erase will be executed. This includes reprogramming of the code protect bits from an on-state to off-state. For all other cases of low voltage ICSP, the part may be programmed at the normal operating voltage. This means calibration values, unique user IDs, or user code can be reprogrammed or added.

14.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. It can also link relocatable objects from pre-compiled libraries, using directives from a linker script.

The MPLIB object librarian is a librarian for pre-compiled code to be used with the MPLINK object linker. When a routine from a library is called from another 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 MPLIB object librarian manages the creation and modification of library files.

The MPLINK object linker features include:

- Integration with MPASM assembler and MPLAB C17 and MPLAB C18 C compilers.
- Allows all memory areas to be defined as sections to provide link-time flexibility.

The MPLIB object librarian features include:

- Easier linking because single libraries can be included instead of many smaller files.
- Helps keep code maintainable by grouping related modules together.
- Allows libraries to be created and modules to be added, listed, replaced, deleted or extracted.

14.5 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC-hosted environment by simulating the PIC MCU series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user-defined key press, to any of the pins. The execution can be performed in single step, execute until break, or trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and the MPLAB C18 C compilers and the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent multi-project software development tool.

14.6 MPLAB ICE High Performance Universal In-Circuit Emulator with MPLAB IDE

The MPLAB ICE universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC MCU microcontrollers (MCUs). Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB ICE in-circuit emulator allows expansion to support new PIC microcontrollers.

The MPLAB ICE in-circuit emulator system has been designed as a real-time emulation system, with advanced features that are generally found on more expensive development tools. The PC platform and Microsoft[®] Windows environment were chosen to best make these features available to you, the end user.

14.7 ICEPIC In-Circuit Emulator

The ICEPIC low cost, in-circuit emulator is a solution for the Microchip Technology PIC16C5X, PIC16C6X, PIC16C7X and PIC16CXXX families of 8-bit One-Time-Programmable (OTP) microcontrollers. The modular system can support different subsets of PIC16C5X or PIC16CXXX products through the use of interchangeable personality modules, or daughter boards. The emulator is capable of emulating without target application circuitry being present.

NOTES:

15.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings †

Ambient temperature under bias...-55 to +125°C

Storage temperature .... -65°C to +150°C

Voltage on any pin with respect to Vss (except VDD, $\overline{MCLR}$. and RA4) .... -0.3 V to (VDD + 0.3 V)

Voltage on VDD with respect to Vss .... -0.3 to +7.5 V

Voltage on $\overline{MCLR}$ with respect to Vss **(Note 2)** ....0 to +14 V

Voltage on RA4 with respect to Vss ....0 to +8.5 V

Total power dissipation **(Note 1)** .... 1.0 W

Maximum current out of Vss pin ....300 mA

Maximum current into VDD pin ....250 mA

Input clamp current, IIK (VI < 0 or VI > VDD)..± 20 mA

Output clamp current, IOK (VO < 0 or VO > VDD) ....± 20 mA

Maximum output current sunk by any I/O pin..25 mA

Maximum output current sourced by any I/O pin ....25 mA

Maximum current sunk by PORTA, PORTB, and PORTE (combined) **(Note 3)**................................200 mA

Maximum current sourced by PORTA, PORTB, and PORTE (combined) **(Note 3)**...........................200 mA

Maximum current sunk by PORTC and PORTD (combined) **(Note 3)** ....200 mA

Maximum current sourced by PORTC and PORTD (combined) **(Note 3)** ....200 mA

Note 1: Power dissipation is calculated as follows: Pdis = VDD x {IDD - Σ IOH} + Σ {(VDD - VOH) x IOH} + Σ(VOl x IOL)

2: Voltage spikes below Vss at the $\overline{MCLR}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin, rather than pulling this pin directly to Vss.

3: PORTD and PORTE are not implemented on PIC16F873/876 devices.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

ON-LINE SUPPORT

Microchip provides on-line support on the Microchip World Wide Web (WWW) site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape or Microsoft Explorer. Files are also available for FTP download from our FTP site.

Connecting to the Microchip Internet Web Site

The Microchip web site is available by using your favorite Internet browser to attach to:

www.microchip.com

The file transfer site is available by using an FTP service to connect to:

ftp://ftp.microchip.com

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products
- Conferences for products, Development Systems, technical information and more
- Listing of seminars and events