

## 28/40-pin 8-Bit CMOS FLASH Microcontrollers

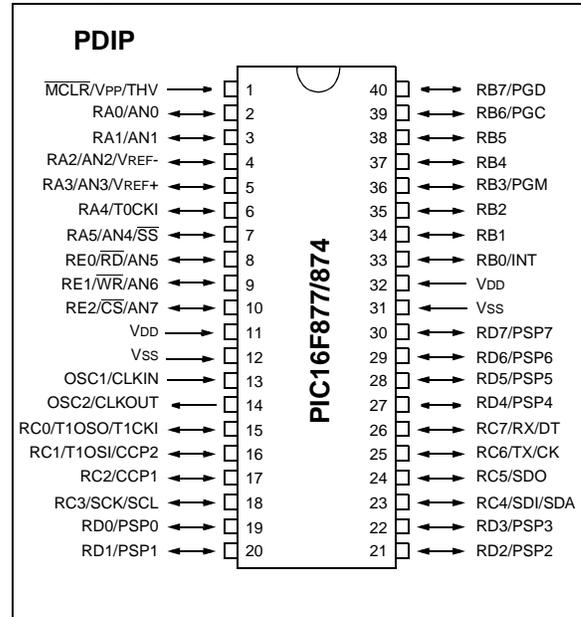
### Devices Included in this Data Sheet:

- PIC16F873                      • PIC16F876
- PIC16F874                      • PIC16F877

### Microcontroller Core Features:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input  
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,  
Up to 368 x 8 bytes of Data Memory (RAM)  
Up to 256 x 8 bytes of EEPROM data memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and  
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC  
oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low-power, high-speed CMOS FLASH/EEPROM  
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two  
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low-power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

### Pin Diagram

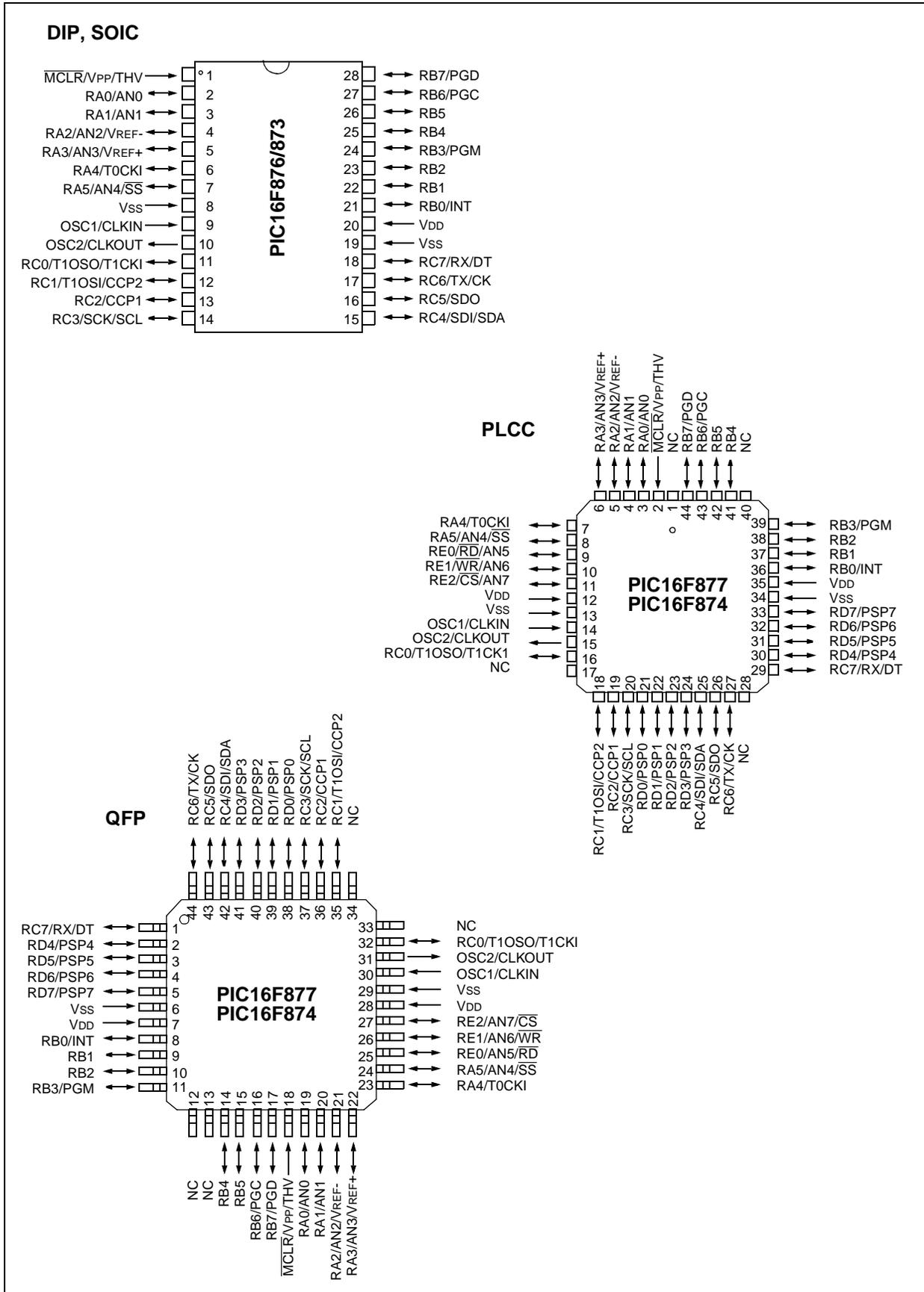


### Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,  
can be incremented during sleep via external  
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period  
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master  
Mode) and I<sup>2</sup>C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver  
Transmitter (USART/SCI) with 9-bit address  
detection
- Parallel Slave Port (PSP) 8-bits wide, with  
external  $\overline{RD}$ ,  $\overline{WR}$  and  $\overline{CS}$  controls (40/44-pin only)
- Brown-out detection circuitry for  
Brown-out Reset (BOR)

# PIC16F87X

## Pin Diagrams



<b>Key Features PICmicro™ Mid-Range Reference Manual (DS33023)</b>	<b>PIC16F873</b>	<b>PIC16F874</b>	<b>PIC16F876</b>	<b>PIC16F877</b>
Operating Frequency	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)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
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	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions

# PIC16F87X

## Table of Contents

1.0	Device Overview .....	5
2.0	Memory Organization.....	11
3.0	I/O Ports .....	29
4.0	Data EEPROM and FLASH Program Memory .....	41
5.0	Timer0 Module .....	47
6.0	Timer1 Module .....	51
7.0	Timer2 Module .....	55
8.0	Capture/Compare/PWM (CCP) Module(s).....	57
9.0	Master Synchronous Serial Port (MSSP) Module.....	63
10.0	Universal Synchronous Asynchronous Receiver Transmitter (USART) .....	95
11.0	Analog-to-Digital Converter (A/D) Module.....	111
12.0	Special Features of the CPU.....	121
13.0	Instruction Set Summary.....	137
14.0	Development Support .....	145
15.0	Electrical Characteristics.....	151
16.0	DC and AC Characteristics Graphs and Tables.....	173
17.0	Packaging Information .....	175
Appendix A:	Revision History .....	183
Appendix B:	Device Differences.....	183
Appendix C:	Conversion Considerations.....	183
Index	.....	185
On-Line Support	.....	191
Product Identification System	.....	193

### *To Our Valued Customers*

#### **Most Current Data Sheet**

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

<http://www.microchip.com>

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number. e.g., DS30000A is version A of document DS30000.

#### **New Customer Notification System**

Register on our web site ([www.microchip.com/cn](http://www.microchip.com/cn)) to receive the most current information on our products.

#### **Errata**

An errata sheet may exist for current devices, describing minor operational differences (from the data sheet) and recommended workarounds. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 786-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

#### **Corrections to this Data Sheet**

We constantly strive to improve the quality of all our products and documentation. We have spent a great deal of time to ensure that this document is correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error, please:

- Fill out and mail in the reader response form in the back of this data sheet.
- E-mail us at [webmaster@microchip.com](mailto:webmaster@microchip.com).

We appreciate your assistance in making this a better document.

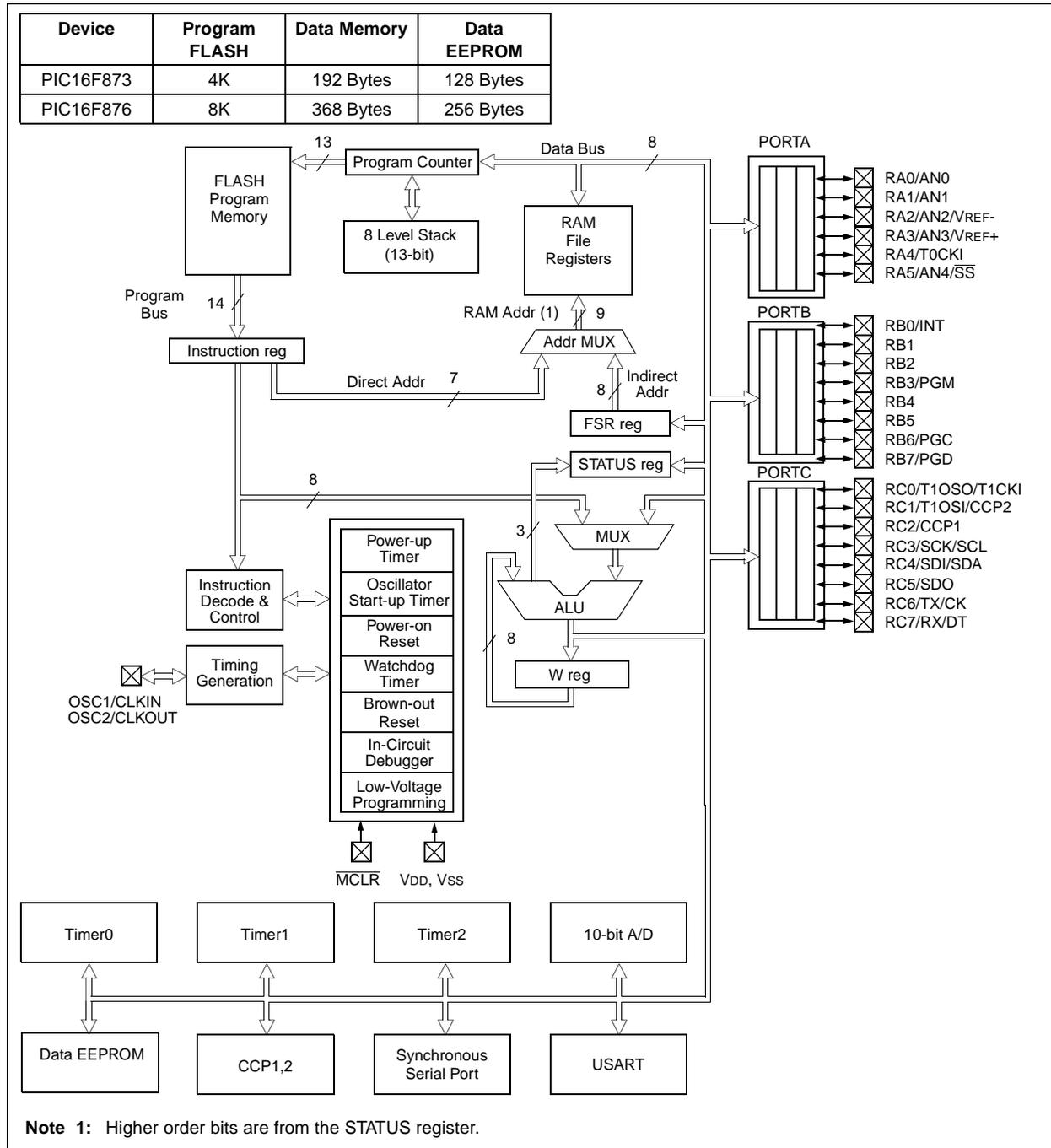
## 1.0 DEVICE OVERVIEW

This document contains device-specific information. Additional information may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip website. 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.

There are four devices (PIC16F873, PIC16F874, PIC16F876 and PIC16F877) covered by this data sheet. The PIC16F876/873 devices come in 28-pin packages and the PIC16F877/874 devices come in 40-pin packages. The 28-pin devices do not have a Parallel Slave Port implemented.

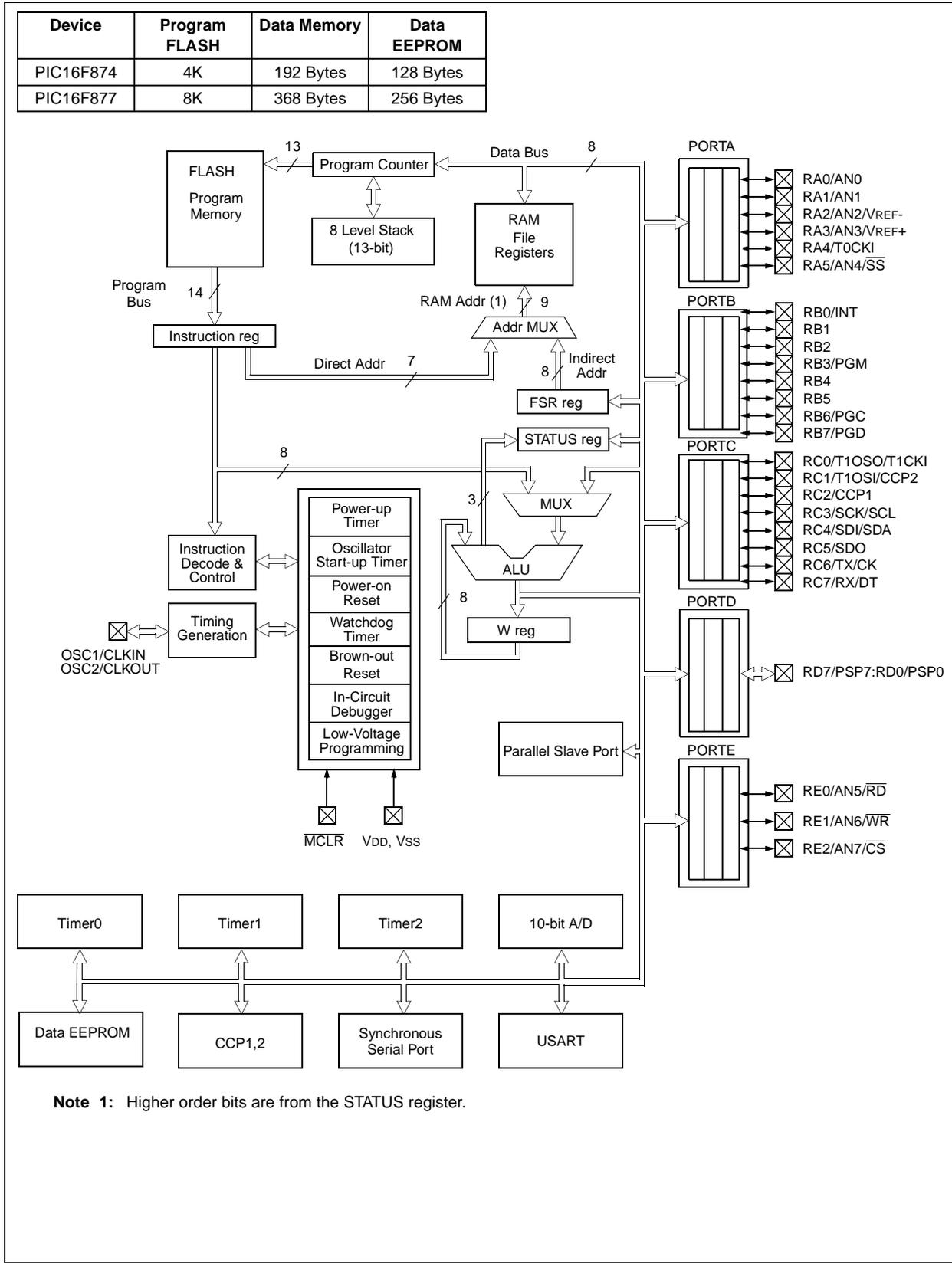
The following two figures are device block diagrams sorted by pin number; 28-pin for Figure 1-1 and 40-pin for Figure 1-2. The 28-pin and 40-pin pinouts are listed in Table 1-1 and Table 1-2, respectively.

**FIGURE 1-1: PIC16F873 AND PIC16F876 BLOCK DIAGRAM**



# PIC16F87X

**FIGURE 1-2: PIC16F874 AND PIC16F877 BLOCK DIAGRAM**



**TABLE 1-1: PIC16F873 AND PIC16F876 PINOUT DESCRIPTION**

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS <sup>(3)</sup>	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP/THV	1	1	I/P	ST	Master clear (reset) input or programming voltage input or high voltage test mode control. This pin is an active low reset to the device.
RA0/AN0	2	2	I/O	TTL	PORTA is a bi-directional I/O port. RA0 can also be analog input0 RA1 can also be analog input1 RA2 can also be analog input2 or negative analog reference voltage RA3 can also be analog input3 or positive analog reference voltage RA4 can also be the clock input to the Timer0 module. Output is open drain type. RA5 can also be analog input4 or the slave select for the synchronous serial port.
RA1/AN1	3	3	I/O	TTL	
RA2/AN2/VREF-	4	4	I/O	TTL	
RA3/AN3/VREF+	5	5	I/O	TTL	
RA4/T0CKI	6	6	I/O	ST	
RA5/SS/AN4	7	7	I/O	TTL	
RB0/INT	21	21	I/O	TTL/ST <sup>(1)</sup>	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. RB0 can also be the external interrupt pin.  RB3 can also be the low voltage programming input Interrupt on change pin. Interrupt on change pin. Interrupt on change pin or In-Circuit Debugger pin. Serial programming clock. Interrupt on change pin or In-Circuit Debugger pin. Serial programming data.
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL	
RB4	25	25	I/O	TTL	
RB5	26	26	I/O	TTL	
RB6/PGC	27	27	I/O	TTL/ST <sup>(2)</sup>	
RB7/PGD	28	28	I/O	TTL/ST <sup>(2)</sup>	
RC0/T1OSO/T1CKI	11	11	I/O	ST	PORTC is a bi-directional I/O port. RC0 can also be the Timer1 oscillator output or Timer1 clock input. RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output. RC2 can also be the Capture1 input/Compare1 output/PWM1 output. RC3 can also be the synchronous serial clock input/output for both SPI and I <sup>2</sup> C modes. RC4 can also be the SPI Data In (SPI mode) or data I/O (I <sup>2</sup> C mode). RC5 can also be the SPI Data Out (SPI mode). RC6 can also be the USART Asynchronous Transmit or Synchronous Clock. RC7 can also be the USART Asynchronous Receive or Synchronous Data.
RC1/T1OSI/CCP2	12	12	I/O	ST	
RC2/CCP1	13	13	I/O	ST	
RC3/SCK/SCL	14	14	I/O	ST	
RC4/SDI/SDA	15	15	I/O	ST	
RC5/SDO	16	16	I/O	ST	
RC6/TX/CK	17	17	I/O	ST	
RC7/RX/DT	18	18	I/O	ST	
Vss	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	P	—	Positive supply for logic and I/O pins.

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 the 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 in RC oscillator mode and a CMOS input otherwise.

# PIC16F87X

**TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION**

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS <sup>(4)</sup>	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLK-OUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP/THV	1	2	18	I/P	ST	Master clear (reset) input or programming voltage input or high voltage test mode control. This pin is an active low reset to the device.
RA0/AN0	2	3	19	I/O	TTL	<p>PORTA is a bi-directional I/O port.</p> <p>RA0 can also be analog input0</p> <p>RA1 can also be analog input1</p> <p>RA2 can also be analog input2 or negative analog reference voltage</p> <p>RA3 can also be analog input3 or positive analog reference voltage</p> <p>RA4 can also be the clock input to the Timer0 timer/counter. Output is open drain type.</p> <p>RA5 can also be analog input4 or the slave select for the synchronous serial port.</p>
RA1/AN1	3	4	20	I/O	TTL	
RA2/AN2/VREF-	4	5	21	I/O	TTL	
RA3/AN3/VREF+	5	6	22	I/O	TTL	
RA4/T0CKI	6	7	23	I/O	ST	
RA5/SS/AN4	7	8	24	I/O	TTL	
RB0/INT	33	36	8	I/O	TTL/ST <sup>(1)</sup>	<p>PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.</p> <p>RB0 can also be the external interrupt pin.</p> <p>RB3 can also be the low voltage programming input</p> <p>Interrupt on change pin.</p> <p>Interrupt on change pin.</p> <p>Interrupt on change pin or In-Circuit Debugger pin. Serial programming clock.</p> <p>Interrupt on change pin or In-Circuit Debugger pin. Serial programming data.</p>
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3/PGM	36	39	11	I/O	TTL	
RB4	37	41	14	I/O	TTL	
RB5	38	42	15	I/O	TTL	
RB6/PGC	39	43	16	I/O	TTL/ST <sup>(2)</sup>	
RB7/PGD	40	44	17	I/O	TTL/ST <sup>(2)</sup>	
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	<p>PORTC is a bi-directional I/O port.</p> <p>RC0 can also be the Timer1 oscillator output or a Timer1 clock input.</p> <p>RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.</p> <p>RC2 can also be the Capture1 input/Compare1 output/PWM1 output.</p> <p>RC3 can also be the synchronous serial clock input/output for both SPI and I<sup>2</sup>C modes.</p> <p>RC4 can also be the SPI Data In (SPI mode) or data I/O (I<sup>2</sup>C mode).</p> <p>RC5 can also be the SPI Data Out (SPI mode).</p> <p>RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.</p> <p>RC7 can also be the USART Asynchronous Receive or Synchronous Data.</p>
RC1/T1OSI/CCP2	16	18	35	I/O	ST	
RC2/CCP1	17	19	36	I/O	ST	
RC3/SCK/SCL	18	20	37	I/O	ST	
RC4/SDI/SDA	23	25	42	I/O	ST	
RC5/SDO	24	26	43	I/O	ST	
RC6/TX/CK	25	27	44	I/O	ST	
RC7/RX/DT	26	29	1	I/O	ST	

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 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.

**TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION (CONTINUED)**

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
RD0/PSP0	19	21	38	I/O	ST/TTL <sup>(3)</sup>	PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus.
RD1/PSP1	20	22	39	I/O	ST/TTL <sup>(3)</sup>	
RD2/PSP2	21	23	40	I/O	ST/TTL <sup>(3)</sup>	
RD3/PSP3	22	24	41	I/O	ST/TTL <sup>(3)</sup>	
RD4/PSP4	27	30	2	I/O	ST/TTL <sup>(3)</sup>	
RD5/PSP5	28	31	3	I/O	ST/TTL <sup>(3)</sup>	
RD6/PSP6	29	32	4	I/O	ST/TTL <sup>(3)</sup>	
RD7/PSP7	30	33	5	I/O	ST/TTL <sup>(3)</sup>	
RE0/ $\overline{\text{RD}}$ /AN5	8	9	25	I/O	ST/TTL <sup>(3)</sup>	<p>PORTE is a bi-directional I/O port.</p> <p>RE0 can also be read control for the parallel slave port, or analog input5.</p>
RE1/ $\overline{\text{WR}}$ /AN6	9	10	26	I/O	ST/TTL <sup>(3)</sup>	<p>RE1 can also be write control for the parallel slave port, or analog input6.</p>
RE2/ $\overline{\text{CS}}$ /AN7	10	11	27	I/O	ST/TTL <sup>(3)</sup>	<p>RE2 can also be select control for the parallel slave port, or analog input7.</p>
VSS	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
VDD	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28,40	12,13,33,34		—	These pins are not internally connected. These pins should be left unconnected.

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.  
**Note 2:** This buffer is a Schmitt Trigger input when used in serial programming mode.  
**Note 3:** This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).  
**Note 4:** This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

# PIC16F87X

---

NOTES:

## 2.0 MEMORY ORGANIZATION

There are three memory blocks in each of these PICmicro 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.

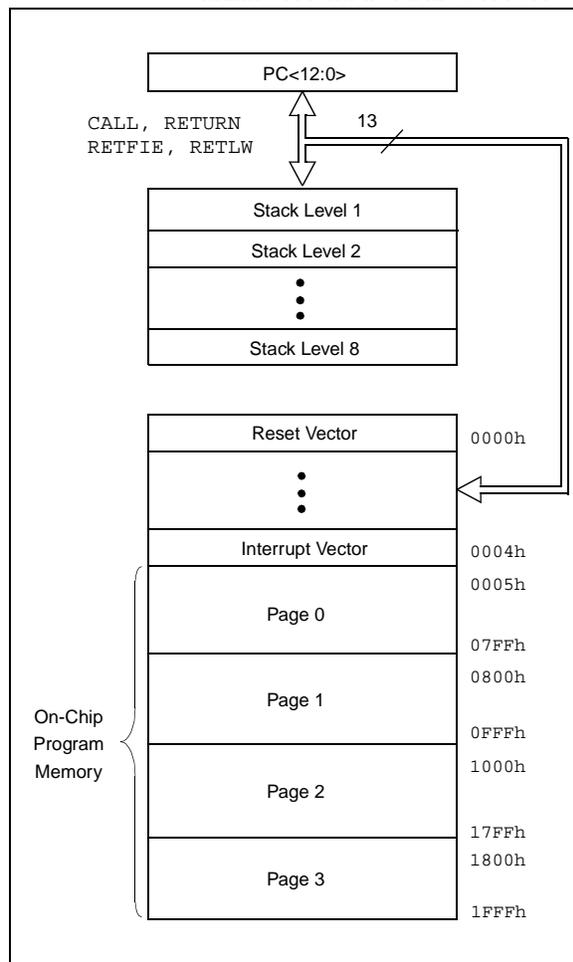
Additional information on device memory may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

### 2.1 Program Memory Organization

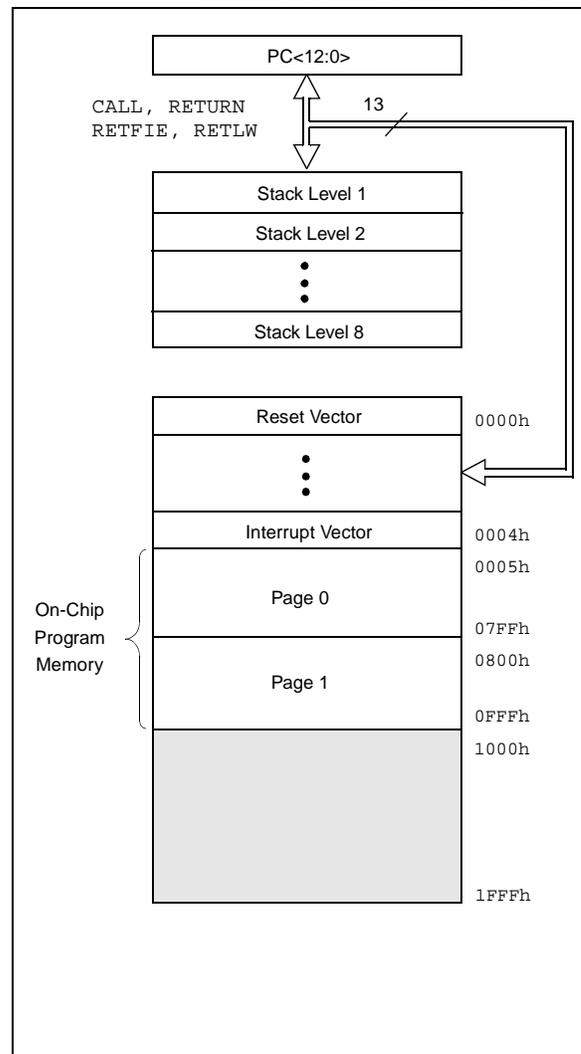
The PIC16F87X devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. The PIC16F877/876 devices have 8K x 14 words of FLASH program memory and the PIC16F873/874 devices have 4K x 14. Accessing a location above the physically implemented address will cause a wrap-around.

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**



# PIC16F87X

---

## 2.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the Special Function Registers. Bits RP1 (STATUS<6>) and RP0 (STATUS<5>) are the bank select bits.

RP1:RP0	Bank
00	0
01	1
10	2
11	3

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some "high use" Special Function Registers from one bank may be mirrored in another bank for code reduction and quicker access.

<b>Note:</b> EEPROM Data Memory description can be found in Section 4.0 of this Data Sheet
--------------------------------------------------------------------------------------------

### 2.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly, or indirectly through the File Select Register FSR.

**FIGURE 2-3: PIC16F877/876 REGISTER FILE MAP**

						File Address	
Indirect addr. (*)	00h	Indirect addr. (*)	80h	Indirect addr. (*)	100h	Indirect addr. (*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD (1)	08h	TRISD (1)	88h		108h		188h
PORTE (1)	09h	TRISE (1)	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved(2)	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved(2)	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h	General Purpose Register 16 Bytes	117h	General Purpose Register 16 Bytes	197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
	20h		A0h		120h		1A0h
General Purpose Register 96 Bytes		General Purpose Register 80 Bytes		General Purpose Register 80 Bytes		General Purpose Register 80 Bytes	
		accesses 70h-7Fh	EFh F0h	accesses 70h-7Fh	16Fh 170h	accesses 70h - 7Fh	1EFh 1F0h
Bank 0	7Fh	Bank 1	FFh	Bank 2	17Fh	Bank 3	1FFh

■ Unimplemented data memory locations, read as '0'.  
 \* Not a physical register.

**Note 1:** These registers are not implemented on 28-pin devices.  
**Note 2:** These registers are reserved, maintain these registers clear.

# PIC16F87X

FIGURE 2-4: PIC16F874/873 REGISTER FILE MAP

						File Address	
Indirect addr. <sup>(*)</sup>	00h	Indirect addr. <sup>(*)</sup>	80h	Indirect addr. <sup>(*)</sup>	100h	Indirect addr. <sup>(*)</sup>	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD <sup>(1)</sup>	08h	TRISD <sup>(1)</sup>	88h		108h		188h
PORTE <sup>(1)</sup>	09h	TRISE <sup>(1)</sup>	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved <sup>(2)</sup>	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved <sup>(2)</sup>	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h				
T2CON	12h	PR2	92h				
SSPBUF	13h	SSPADD	93h				
SSPCON	14h	SSPSTAT	94h				
CCPR1L	15h		95h				
CCPR1H	16h		96h				
CCP1CON	17h		97h				
RCSTA	18h	TXSTA	98h				
TXREG	19h	SPBRG	99h				
RCREG	1Ah		9Ah				
CCPR2L	1Bh		9Bh				
CCPR2H	1Ch		9Ch				
CCP2CON	1Dh		9Dh				
ADRESH	1Eh	ADRESL	9Eh				
ADCON0	1Fh	ADCON1	9Fh				
	20h		A0h		120h		1A0h
General Purpose Register		General Purpose Register		accesses 20h-7Fh		accesses A0h - FFh	
96 Bytes		96 Bytes			16Fh		1EFh
	7Fh		FFh		170h		1F0h
Bank 0		Bank 1		Bank 2		Bank 3	
					17Fh		1FFh

■ Unimplemented data memory locations, read as '0'.  
 \* Not a physical register.

**Note 1:** These registers are not implemented on 28-pin devices.  
**Note 2:** These registers are reserved, maintain these registers clear.

## 2.2.2 SPECIAL FUNCTION REGISTERS

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

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

**TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets (2)		
<b>Bank 0</b>													
00h <sup>(4)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000		
01h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu		
02h <sup>(4)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000		
03h <sup>(4)</sup>	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu		
04h <sup>(4)</sup>	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu		
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read								--0x 0000	--0u 0000
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu		
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	uuuu uuuu		
08h <sup>(5)</sup>	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	uuuu uuuu		
09h <sup>(5)</sup>	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu		
0Ah <sup>(1,4)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---	0 0000	---	0 0000
0Bh <sup>(4)</sup>	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u		
0Ch	PIR1	PSPIF <sup>(3)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000		
0Dh	PIR2	—	<b>(6)</b>	—	EEIF	BCLIF	—	—	CCP2IF	-r-0 0--0	-r-0 0--0		
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu		
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu		
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu		
11h	TMR2	Timer2 module's register								0000 0000	0000 0000		
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000		
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu		
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000		
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu		
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu		
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000		
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x		
19h	TXREG	USART Transmit Data Register								0000 0000	0000 0000		
1Ah	RCREG	USART Receive Data Register								0000 0000	0000 0000		
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	uuuu uuuu		
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	uuuu uuuu		
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000		
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu		
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	$\overline{GO/DONE}$	—	ADON	0000 00-0	0000 00-0		

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

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

**2:** Other (non power-up) resets include external reset through MCLR and Watchdog Timer Reset.

**3:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

**4:** These registers can be addressed from any bank.

**5:** PORTD, PORTE, TRISD, and TRISE are not physically implemented on the 28-pin devices, read as '0'.

**6:** PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

# PIC16F87X

**TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets (2)	
<b>Bank 1</b>												
80h <sup>(4)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000	
81h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111	
82h <sup>(4)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000	
83h <sup>(4)</sup>	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu	
84h <sup>(4)</sup>	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu	
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111	
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111	
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111	
88h <sup>(5)</sup>	TRISD	PORTD Data Direction Register								1111 1111	1111 1111	
89h <sup>(5)</sup>	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits				0000 -111	0000 -111
8Ah <sup>(1,4)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter						---0 0000	---0 0000
8Bh <sup>(4)</sup>	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u	
8Ch	PIE1	PSPIE <sup>(3)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000	
8Dh	PIE2	—	<b>(6)</b>	—	EEIE	BCLIE	—	—	CCP2IE	-r-0 0--0	-r-0 0--0	
8Eh	PCON	—	—	—	—	—	—	POR	BOR	---- -rqq	---- -ruu	
8Fh	—	Unimplemented								—	—	
90h	—	Unimplemented								—	—	
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000	
92h	PR2	Timer2 Period Register								1111 1111	1111 1111	
93h	SSPADD	Synchronous Serial Port (I <sup>2</sup> C mode) Address Register								0000 0000	0000 0000	
94h	SSPSTAT	SMP	CKE	D/ $\overline{A}$	P	S	R/ $\overline{W}$	UA	BF	0000 0000	0000 0000	
95h	—	Unimplemented								—	—	
96h	—	Unimplemented								—	—	
97h	—	Unimplemented								—	—	
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010	
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000	
9Ah	—	Unimplemented								—	—	
9Bh	—	Unimplemented								—	—	
9Ch	—	Unimplemented								—	—	
9Dh	—	Unimplemented								—	—	
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu	
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	0--- 0000	0--- 0000	

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.
- 2:** Other (non power-up) resets include external reset through MCLR and Watchdog Timer Reset.
- 3:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.
- 4:** These registers can be addressed from any bank.
- 5:** PORTD, PORTE, TRISD, and TRISE are not physically implemented on the 28-pin devices, read as '0'.
- 6:** PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

**TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets (2)
<b>Bank 2</b>											
100h <sup>(4)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
101h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
102h <sup>(4)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
103h <sup>(4)</sup>	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	000q quuu
104h <sup>(4)</sup>	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
105h	—	Unimplemented								—	—
106h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
107h	—	Unimplemented								—	—
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah <sup>(1,4)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
10Bh <sup>(4)</sup>	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
10Ch	EEDATA	EEPROM data register								xxxx xxxx	uuuu uuuu
10Dh	EEADR	EEPROM address register								xxxx xxxx	uuuu uuuu
10Eh	EEDATH	—	—	EEPROM data register high byte					xxxx xxxx	uuuu uuuu	
10Fh	EEADRH	—	—	—	EEPROM address register high byte					xxxx xxxx	uuuu uuuu
<b>Bank 3</b>											
180h <sup>(4)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
181h	OPTION_REG	RBP $\bar{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
182h <sup>(4)</sup>	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
183h <sup>(4)</sup>	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	000q quuu
184h <sup>(4)</sup>	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
185h	—	Unimplemented								—	—
186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
187h	—	Unimplemented								—	—
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah <sup>(1,4)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
18Bh <sup>(4)</sup>	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	x--- u000
18Dh	EECON2	EEPROM control register2 (not a physical register)								---- ----	---- ----
18Eh	—	Reserved maintain clear								0000 0000	0000 0000
18Fh	—	Reserved maintain clear								0000 0000	0000 0000

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

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

**2:** Other (non power-up) resets include external reset through MCLR and Watchdog Timer Reset.

**3:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

**4:** These registers can be addressed from any bank.

**5:** PORTD, PORTE, TRISD, and TRISE are not physically implemented on the 28-pin devices, read as '0'.

**6:** PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

# PIC16F87X

## 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  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  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 the "Instruction Set Summary."

**Note 1:** The C and 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 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	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
bit7						bit0	

R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 - n = Value at POR reset

- 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:  **$\overline{\text{TO}}$ :** Time-out bit  
 1 = After power-up, `CLRWDT` instruction, or `SLEEP` instruction  
 0 = A WDT time-out occurred
- bit 3:  **$\overline{\text{PD}}$ :** 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)  
 (for borrow the polarity is reversed)  
 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.

## 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 assignable 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	
$\overline{\text{RBP}}\text{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	
bit7								bit0

R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 - n= Value at POR reset

bit 7:  **$\overline{\text{RBP}}\text{U}$ : 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

**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 operation of the device.

# PIC16F87X

## 2.2.2.3 INTCON REGISTER

The INTCON Register is a readable and writable register, which contains various enable and flag bits for the TMR0 register overflow, RB Port change and External RB0/INT pin interrupts.

**Note:** Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

### REGISTER 2-3: INTCON REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit7							bit0
<div style="float: right; border: 1px solid black; padding: 5px; margin-top: 10px;">                     R = Readable bit                      W = Writable bit                      U = Unimplemented bit, read as '0'                      - n= Value at POR reset                 </div> <p>bit 7: <b>GIE:</b> Global Interrupt Enable bit                      1 = Enables all un-masked interrupts                      0 = Disables all interrupts</p> <p>bit 6: <b>PEIE:</b> Peripheral Interrupt Enable bit                      1 = Enables all un-masked peripheral interrupts                      0 = Disables all peripheral interrupts</p> <p>bit 5: <b>TOIE:</b> TMR0 Overflow Interrupt Enable bit                      1 = Enables the TMR0 interrupt                      0 = Disables the TMR0 interrupt</p> <p>bit 4: <b>INTE:</b> RB0/INT External Interrupt Enable bit                      1 = Enables the RB0/INT external interrupt                      0 = Disables the RB0/INT external interrupt</p> <p>bit 3: <b>RBIE:</b> RB Port Change Interrupt Enable bit                      1 = Enables the RB port change interrupt                      0 = Disables the RB port change interrupt</p> <p>bit 2: <b>TOIF:</b> TMR0 Overflow Interrupt Flag bit                      1 = TMR0 register has overflowed (must be cleared in software)                      0 = TMR0 register did not overflow</p> <p>bit 1: <b>INTF:</b> RB0/INT External Interrupt Flag bit                      1 = The RB0/INT external interrupt occurred (must be cleared in software)                      0 = The RB0/INT external interrupt did not occur</p> <p>bit 0: <b>RBIF:</b> RB Port Change Interrupt Flag bit                      1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)                      0 = None of the RB7:RB4 pins have changed state</p>							

## 2.2.2.4 PIE1 REGISTER

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

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

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	
bit7								bit0
<p>bit 7: <b>PSPIE<sup>(1)</sup></b>: Parallel Slave Port Read/Write Interrupt Enable bit            1 = Enables the PSP read/write interrupt            0 = Disables the PSP read/write interrupt</p> <p>bit 6: <b>ADIE</b>: A/D Converter Interrupt Enable bit            1 = Enables the A/D converter interrupt            0 = Disables the A/D converter interrupt</p> <p>bit 5: <b>RCIE</b>: USART Receive Interrupt Enable bit            1 = Enables the USART receive interrupt            0 = Disables the USART receive interrupt</p> <p>bit 4: <b>TXIE</b>: USART Transmit Interrupt Enable bit            1 = Enables the USART transmit interrupt            0 = Disables the USART transmit interrupt</p> <p>bit 3: <b>SSPIE</b>: Synchronous Serial Port Interrupt Enable bit            1 = Enables the SSP interrupt            0 = Disables the SSP interrupt</p> <p>bit 2: <b>CCP1IE</b>: CCP1 Interrupt Enable bit            1 = Enables the CCP1 interrupt            0 = Disables the CCP1 interrupt</p> <p>bit 1: <b>TMR2IE</b>: TMR2 to PR2 Match Interrupt Enable bit            1 = Enables the TMR2 to PR2 match interrupt            0 = Disables the TMR2 to PR2 match interrupt</p> <p>bit 0: <b>TMR1IE</b>: TMR1 Overflow Interrupt Enable bit            1 = Enables the TMR1 overflow interrupt            0 = Disables the TMR1 overflow interrupt</p> <p><b>Note 1:</b> PSPIE is reserved on 28-pin devices; always maintain this bit clear.</p>								

R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 - n = Value at POR reset

# PIC16F87X

## 2.2.2.5 PIR1 REGISTER

The PIR1 register contains the individual flag bits for the peripheral interrupts.

**Note:** Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt bits are clear prior to enabling an interrupt.

### REGISTER 2-5: PIR1 REGISTER (ADDRESS 0Ch)

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF

bit7

bit0

R = Readable bit  
W = Writable bit  
- n= Value at POR reset

- bit 7: **PSPIF<sup>(1)</sup>**: Parallel Slave Port Read/Write Interrupt Flag bit  
1 = A read or a write operation has taken place (must be cleared in software)  
0 = No read or write has occurred
- bit 6: **ADIF**: A/D Converter Interrupt Flag bit  
1 = An A/D conversion completed  
0 = The A/D conversion is not complete
- bit 5: **RCIF**: USART Receive Interrupt Flag bit  
1 = The USART receive buffer is full  
0 = The USART receive buffer is empty
- bit 4: **TXIF**: USART Transmit Interrupt Flag bit  
1 = The USART transmit buffer is empty  
0 = The USART transmit buffer is full
- bit 7: **SSPIF**: Synchronous Serial Port (SSP) Interrupt Flag  
1 = The SSP interrupt condition has occurred, and must be cleared in software before returning from the interrupt service routine. The conditions that will set this bit are:  
SPI  
A transmission/reception has taken place.  
I<sup>2</sup>C Slave  
A transmission/reception has taken place.  
I<sup>2</sup>C Master  
A transmission/reception has taken place.  
The initiated start condition was completed by the SSP module.  
The initiated stop condition was completed by the SSP module.  
The initiated restart condition was completed by the SSP module.  
The initiated acknowledge condition was completed by the SSP module.  
A start condition occurred while the SSP module was idle (Multimaster system).  
A stop condition occurred while the SSP module was idle (Multimaster system).  
0 = No SSP interrupt condition has occurred.
- bit 2: **CCP1IF**: CCP1 Interrupt Flag bit  
Capture Mode  
1 = A TMR1 register capture occurred (must be cleared in software)  
0 = No TMR1 register capture occurred  
Compare Mode  
1 = A TMR1 register compare match occurred (must be cleared in software)  
0 = No TMR1 register compare match occurred  
PWM Mode  
Unused in this mode
- bit 1: **TMR2IF**: TMR2 to PR2 Match Interrupt Flag bit  
1 = TMR2 to PR2 match occurred (must be cleared in software)  
0 = No TMR2 to PR2 match occurred
- bit 0: **TMR1IF**: TMR1 Overflow Interrupt Flag bit  
1 = TMR1 register overflowed (must be cleared in software)  
0 = TMR1 register did not overflow

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

## 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
—	—	—	EEIE	BCLIE	—	—	CCP2IE
bit7							bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n= Value at POR reset

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

# PIC16F87X

## 2.2.2.7 PIR2 REGISTER

The PIR2 register contains the flag bits for the CCP2 interrupt, the SSP bus collision interrupt and the EEPROM write operation interrupt.

**Note:** Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

### REGISTER 2-7: PIR2 REGISTER (ADDRESS 0Dh)

U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	
—	—	—	EEIF	BCLIF	—	—	CCP2IF	
bit7								bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

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

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

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

bit 4: **EEIF:** EEPROM Write Operation Interrupt Flag bit  
1 = The write operation completed (must be cleared in software)  
0 = The write operation is not complete or has not been started

bit 3: **BCLIF:** Bus Collision Interrupt Flag  
1 = A bus collision has occurred in the SSP, when configured for I<sup>2</sup>C master mode  
0 = No bus collision has occurred

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

bit 0: **CCP2IF:** CCP2 Interrupt Flag bit  
Capture Mode  
1 = A TMR1 register capture occurred (must be cleared in software)  
0 = No TMR1 register capture occurred  
Compare Mode  
1 = A TMR1 register compare match occurred (must be cleared in software)  
0 = No TMR1 register compare match occurred  
PWM Mode  
Unused

## 2.2.2.8 PCON REGISTER

The Power Control (PCON) Register contains flag bits to allow differentiation between a Power-on Reset (POR), a Brown-out Reset ( $\overline{\text{BOR}}$ ), a Watch-dog Reset (WDT) and an external  $\overline{\text{MCLR}}$  Reset.

**Note:**  $\overline{\text{BOR}}$  is unknown on POR. It must be set by the user and checked on subsequent rests to see if BOR is clear, indicating a brown-out has occurred. The BOR status bit is a don't care and is not predictable if the brown-out circuit is disabled (by clearing the BODEN bit in the configuration word).

### REGISTER 2-8: PCON REGISTER (ADDRESS 8Eh)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-1
—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit7						bit0	

R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 - n= Value at POR reset

bit 7-2: **Unimplemented:** Read as '0'

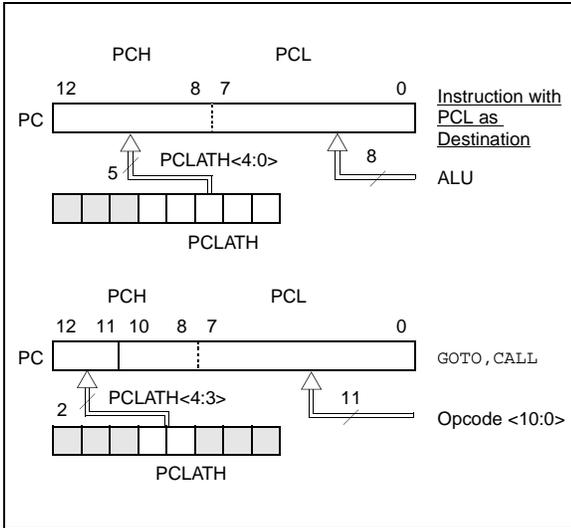
bit 1:  **$\overline{\text{POR}}$ :** Power-on Reset Status bit  
 1 = No Power-on Reset occurred  
 0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0:  **$\overline{\text{BOR}}$ :** Brown-out Reset Status bit  
 1 = No Brown-out Reset occurred  
 0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

## 2.3 PCL and PCLATH

The program counter (PC) is 13-bits wide. The low byte comes from the PCL register, which is a readable and writable register. The upper bits (PC<12:8>) are not readable, but are indirectly writable through the PCLATH register. On any reset, the upper bits of the PC will be cleared. Figure 2-5 shows the two situations for the loading of the PC. The upper example in the figure shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in the figure shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

**FIGURE 2-5: LOADING OF PC IN DIFFERENT SITUATIONS**



### 2.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256 byte block). Refer to the application note, "Implementing a Table Read" (AN556).

### 2.3.2 STACK

The PIC16CXX family has an 8-level deep x 13-bit wide hardware stack. The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

**Note 1:** There are no status bits to indicate stack overflow or stack underflow conditions.

**2:** There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address.

## 2.4 Program Memory Paging

PIC16CXX devices are capable of addressing a continuous 8K word block of program memory. The CALL and GOTO instructions provide only 11 bits of address to allow branching within any 2K program memory page. When doing a CALL or GOTO instruction, the upper 2 bits of the address are provided by PCLATH<4:3>. When doing a CALL or GOTO instruction, the user must ensure that the page select bits are programmed so that the desired program memory page is addressed. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is popped off the stack. Therefore, manipulation of the PCLATH<4:3> bits are not required for the return instructions (which POPs the address from the stack)

Example 2-1 shows the calling of a subroutine in page 1 of the program memory. This example assumes that PCLATH is saved and restored by the interrupt service routine (if interrupts are used).

### EXAMPLE 2-1: CALL OF A SUBROUTINE IN PAGE 1 FROM PAGE 0

```

ORG 0x500
BCF PCLATH,4
BSF PCLATH,3 ;Select page 1 (800h-FFFh)
CALL SUB1_P1 ;Call subroutine in
:           ;page 1 (800h-FFFh)
:
ORG 0x900 ;page 1 (800h-FFFh)
SUB1_P1
:           ;called subroutine
:           ;page 1 (800h-FFFh)
:
RETURN ;return to Call subroutine
;in page 0 (000h-7FFh)
    
```

## 2.5 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF register itself indirectly (FSR = '0') will read 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 2-6.

A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 2-2.

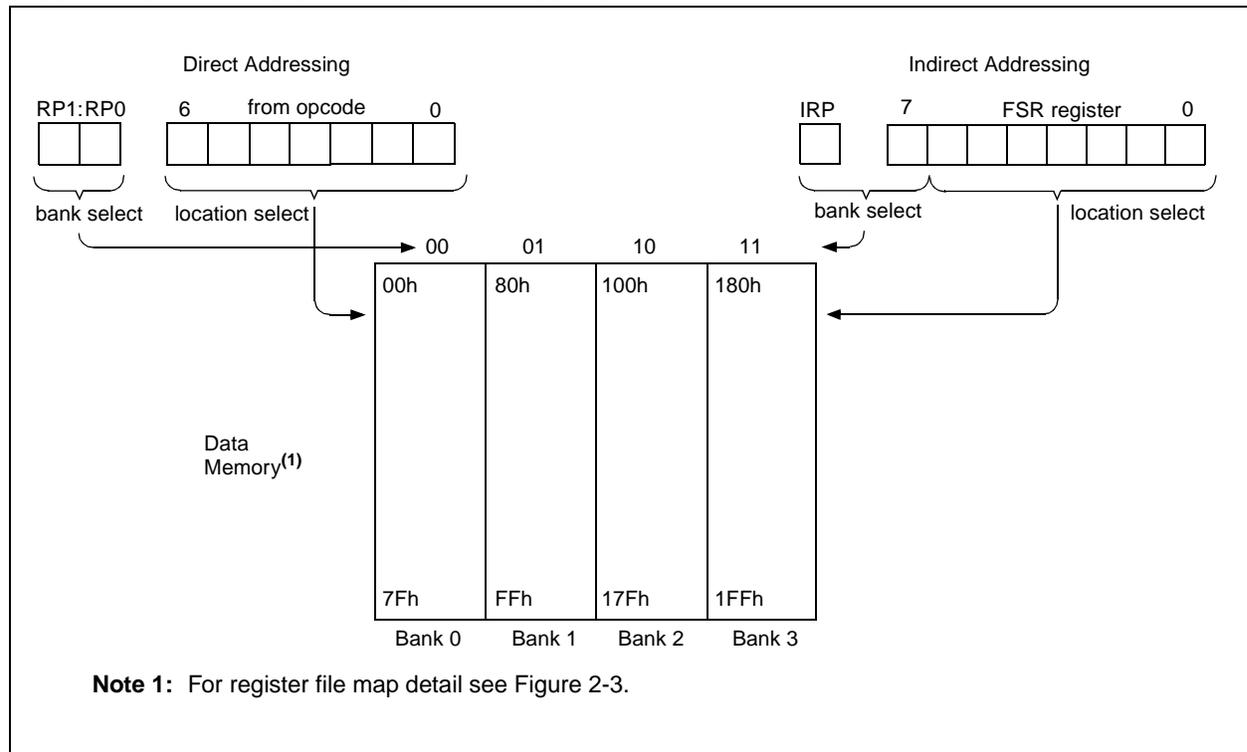
### EXAMPLE 2-2: INDIRECT ADDRESSING

```

movlw 0x20 ;initialize pointer
movwf FSR ;to RAM
NEXT   clrf INDF ;clear INDF register
       incf FSR,F ;inc pointer
       btfss FSR,4 ;all done?
       goto NEXT ;no clear next

CONTINUE
:      ;yes continue
    
```

**FIGURE 2-6: DIRECT/INDIRECT ADDRESSING**



# PIC16F87X

---

NOTES:

## 3.0 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 PICmicro™ Mid-Range Reference Manual, (DS33023).

### 3.1 PORTA and the TRISA Register

PORTA is a 6-bit wide bi-directional 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 hi-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. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, the value is modified and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other PORTA pins have TTL input levels and full CMOS output drivers.

Other PORTA pins are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1).

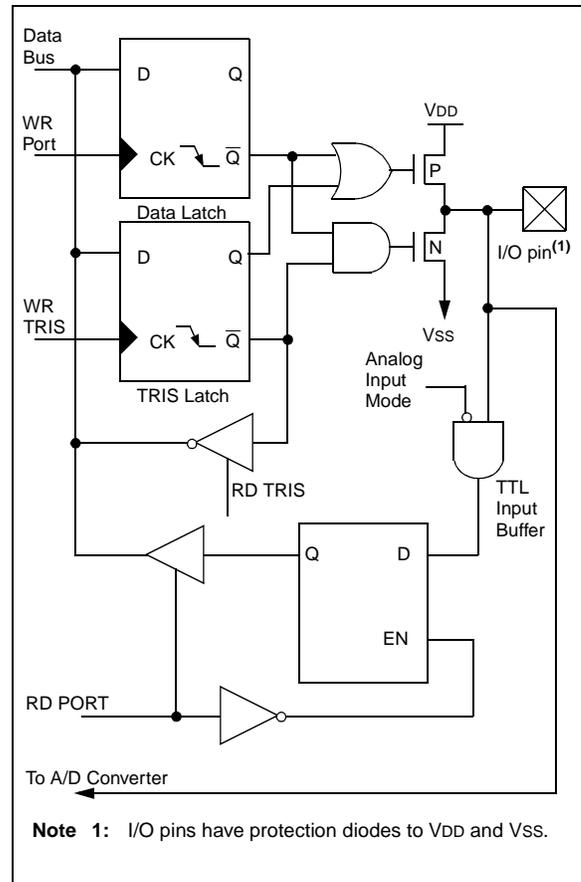
**Note:** On a Power-on Reset, these pins are configured as analog inputs and read as '0'.

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.

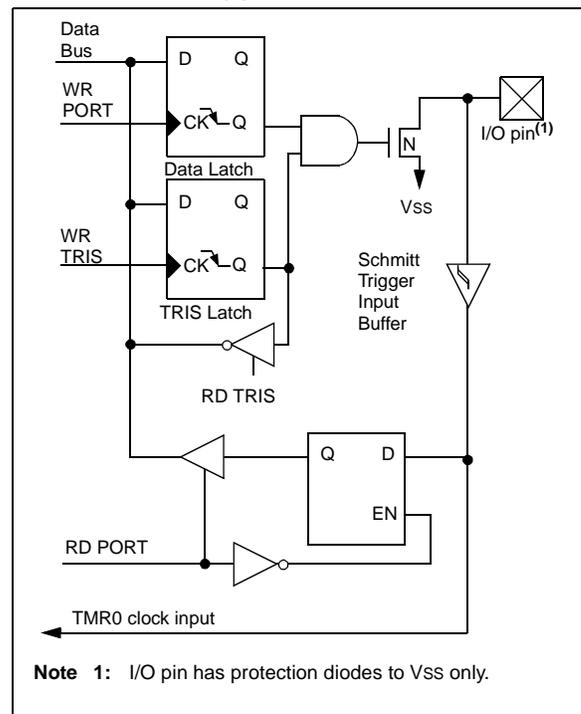
#### EXAMPLE 3-1: INITIALIZING PORTA

```
BCF STATUS, RP0 ;
BCF STATUS, RP1 ; Bank0
CLRF PORTA ; Initialize PORTA by
; clearing output
; data latches
BSF STATUS, RP0 ; Select Bank 1
MOVLW 0x06 ; 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'.
```

**FIGURE 3-1: BLOCK DIAGRAM OF RA3:RA0 AND RA5 PINS**



**FIGURE 3-2: BLOCK DIAGRAM OF RA4/T0CKI PIN**



# PIC16F87X

**TABLE 3-1: PORTA FUNCTIONS**

Name	Bit#	Buffer	Function
RA0/AN0	bit0	TTL	Input/output or analog input
RA1/AN1	bit1	TTL	Input/output or analog input
RA2/AN2	bit2	TTL	Input/output or analog input
RA3/AN3/VREF	bit3	TTL	Input/output or analog input or VREF
RA4/T0CKI	bit4	ST	Input/output or external clock input for Timer0 Output is open drain type
RA5/ $\overline{SS}$ /AN4	bit5	TTL	Input/output or slave select input for synchronous serial port or analog input

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

**TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000

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

**Note:** When using the SSP module in SPI slave mode and  $\overline{SS}$  enabled, the A/D converter must be set to one of the following modes where PCFG3:PCFG0 = 0100, 0101, 011x, 1101, 1110, 1111.

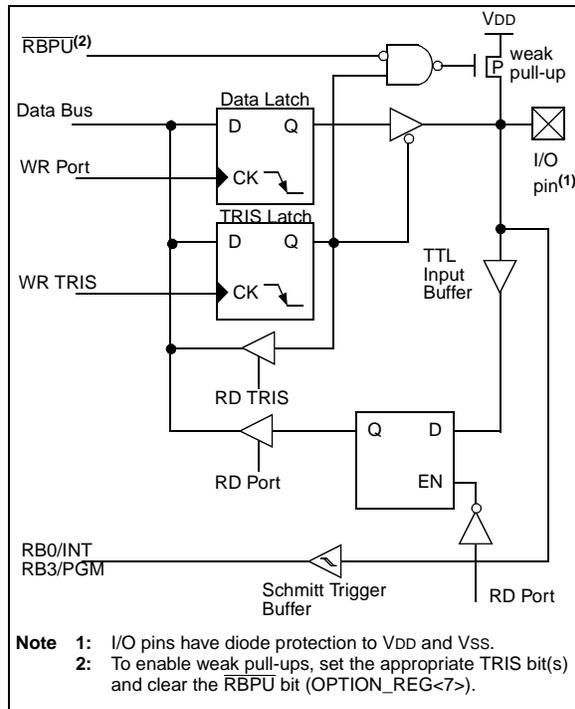
## 3.2 PORTB and the TRISB Register

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  $\overline{\text{RBP}}\text{U}$  (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**



Four of PORTB's 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:

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

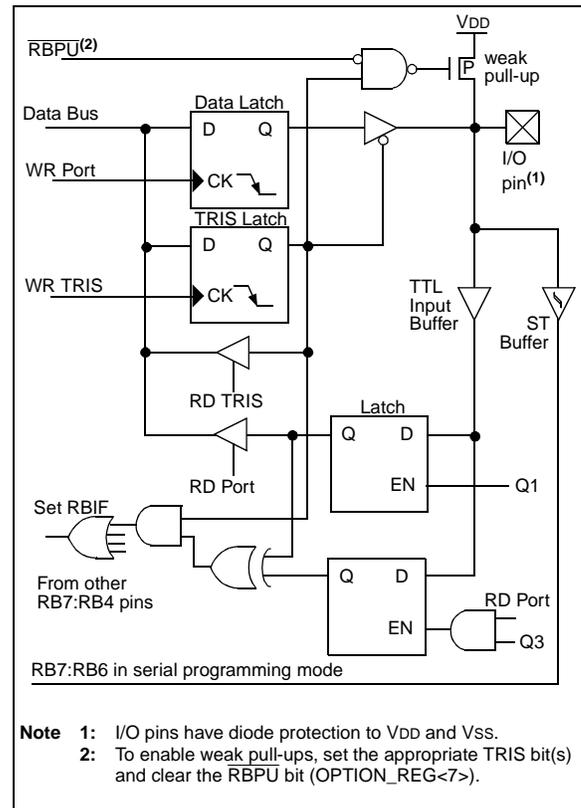
The interrupt on change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt on change feature. Polling of PORTB is not recommended while using the interrupt on change feature.

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

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

RB0/INT is discussed in detail in Section 12.10.1.

**FIGURE 3-4: BLOCK DIAGRAM OF RB7:RB4 PINS**



**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 operation of the device.

# PIC16F87X

**TABLE 3-3: PORTB FUNCTIONS**

Name	Bit#	Buffer	Function
RB0/INT	bit0	TTL/ST <sup>(1)</sup>	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3/PGM	bit3	TTL	Input/output pin or programming pin in LVP mode. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB6/PGC	bit6	TTL/ST <sup>(2)</sup>	Input/output pin (with interrupt on change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming clock.
RB7/PGD	bit7	TTL/ST <sup>(2)</sup>	Input/output pin (with interrupt on change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming data.

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

**Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt.

**2:** This buffer is a Schmitt Trigger input when used in serial programming mode.

**TABLE 3-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
06h, 106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h, 186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
81h, 181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

### 3.3 PORTC and the TRISC Register

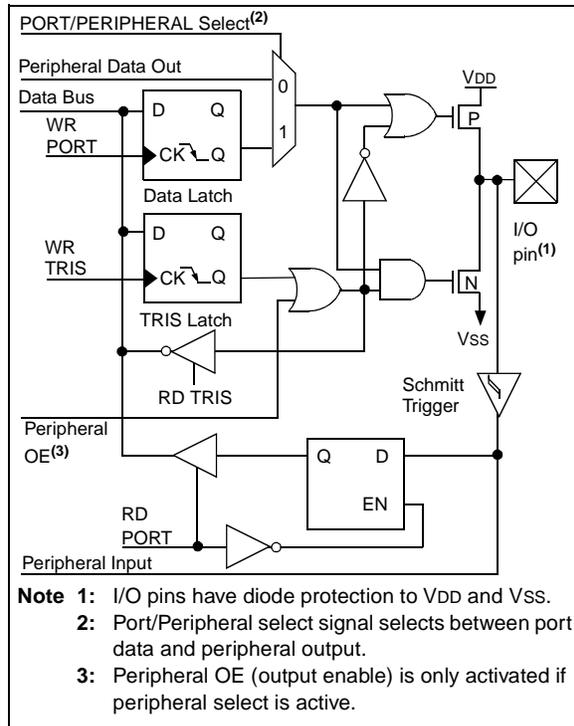
PORTC is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISC. Setting a TRISC bit (=1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a hi-impedance mode). Clearing a TRISC bit (=0) will make the corresponding PORTC pin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

PORTC is multiplexed with several peripheral functions (Table 3-5). PORTC pins have Schmitt Trigger input buffers.

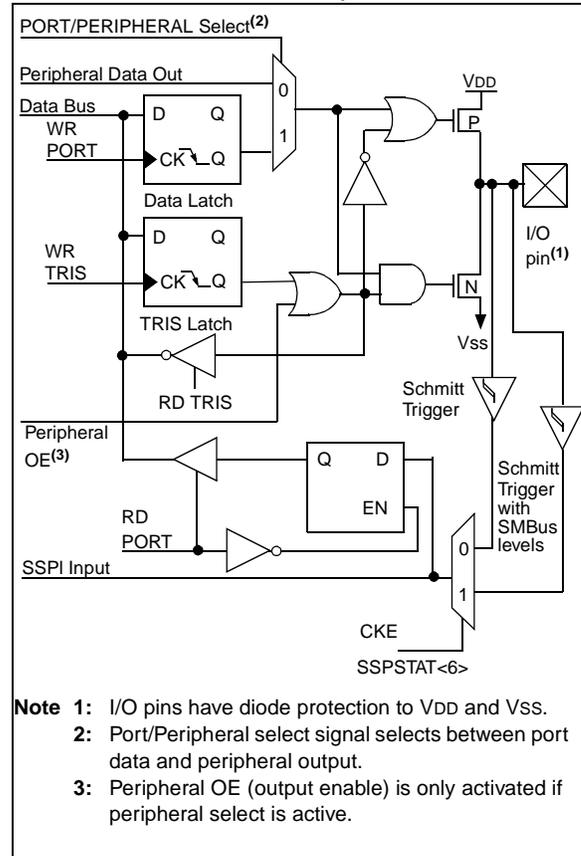
When the I<sup>2</sup>C module is enabled, the PORTC (3:4) pins can be configured with normal I<sup>2</sup>C levels or with SMBUS levels by using the CKE bit (SSPSTAT <6>).

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.

**FIGURE 3-5: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<0:2> RC<5:7>**



**FIGURE 3-6: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<3:4>**



# PIC16F87X

**TABLE 3-5: PORTC FUNCTIONS**

Name	Bit#	Buffer Type	Function
RC0/T1OSO/T1CKI	bit0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input
RC1/T1OSI/CCP2	bit1	ST	Input/output port pin or Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output
RC2/CCP1	bit2	ST	Input/output port pin or Capture1 input/Compare1 output/PWM1 output
RC3/SCK/SCL	bit3	ST	RC3 can also be the synchronous serial clock for both SPI and I <sup>2</sup> C modes.
RC4/SDI/SDA	bit4	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I <sup>2</sup> C mode).
RC5/SDO	bit5	ST	Input/output port pin or Synchronous Serial Port data output
RC6/TX/CK	bit6	ST	Input/output port pin or USART Asynchronous Transmit or Synchronous Clock
RC7/RX/DT	bit7	ST	Input/output port pin or USART Asynchronous Receive or Synchronous Data

Legend: ST = Schmitt Trigger input

**TABLE 3-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged.

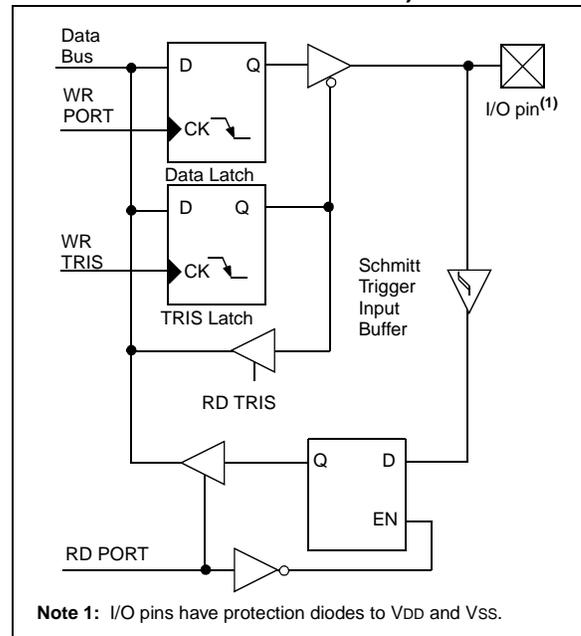
## 3.4 PORTD and TRISD Registers

This section is not applicable to the PIC16F873 or PIC16F876.

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

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

**FIGURE 3-7: PORTD BLOCK DIAGRAM (IN I/O PORT MODE)**



**TABLE 3-7: PORTD FUNCTIONS**

Name	Bit#	Buffer Type	Function
RD0/PSP0	bit0	ST/TTL <sup>(1)</sup>	Input/output port pin or parallel slave port bit0
RD1/PSP1	bit1	ST/TTL <sup>(1)</sup>	Input/output port pin or parallel slave port bit1
RD2/PSP2	bit2	ST/TTL <sup>(1)</sup>	Input/output port pin or parallel slave port bit2
RD3/PSP3	bit3	ST/TTL <sup>(1)</sup>	Input/output port pin or parallel slave port bit3
RD4/PSP4	bit4	ST/TTL <sup>(1)</sup>	Input/output port pin or parallel slave port bit4
RD5/PSP5	bit5	ST/TTL <sup>(1)</sup>	Input/output port pin or parallel slave port bit5
RD6/PSP6	bit6	ST/TTL <sup>(1)</sup>	Input/output port pin or parallel slave port bit6
RD7/PSP7	bit7	ST/TTL <sup>(1)</sup>	Input/output port pin or parallel slave port bit7

Legend: ST = Schmitt Trigger input TTL = TTL input

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

**TABLE 3-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
08h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	uuuu uuuu
88h	TRISD	PORTD Data Direction Register								1111 1111	1111 1111
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111

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

# PIC16F87X

## 3.5 PORTE and TRISE Register

This section is not applicable to the PIC16F873 or PIC16F876.

PORTE has three pins, RE0/ $\overline{RD}$ /AN5, RE1/ $\overline{WR}$ /AN6 and RE2/ $\overline{CS}$ /AN7, which are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers.

I/O PORTE becomes control inputs for the microprocessor port when bit PSPMODE (TRISE<4>) is set. In this mode, the user must make sure that the TRISE<2:0> bits are set (pins are configured as digital inputs). Ensure ADCON1 is configured for digital I/O. In this mode, the input buffers are TTL.

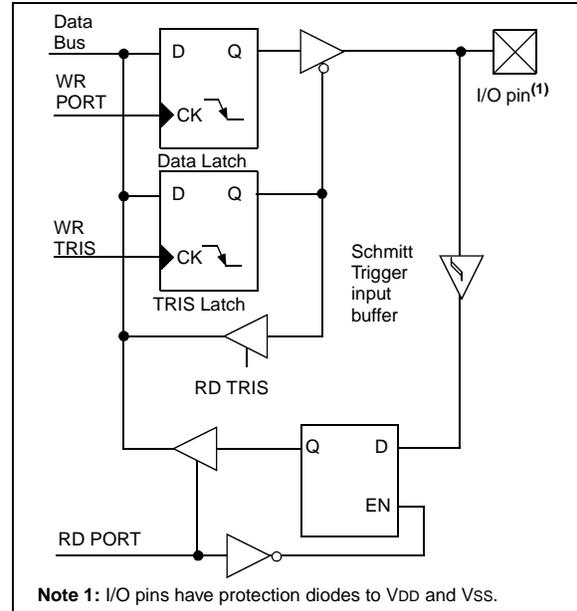
Register 3-1 shows the TRISE register, which also controls the parallel slave port operation.

PORTE pins are multiplexed with analog inputs. When selected as an analog input, these pins will read as '0's.

TRISE controls the direction of the RE pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs.

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

**FIGURE 3-8: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)**



**REGISTER 3-1: TRISE REGISTER (ADDRESS 89h)**

R-0	R-0	R/W-0	R/W-0	U-0	R/W-1	R/W-1	R/W-1	R = Readable bit W = Writable bit U = Unimplemented bit, read as '0' - n = Value at POR reset
IBF	OBF	IBOV	PSPMODE	—	bit2	bit1	bit0	
bit7							bit0	

**Parallel Slave Port Status/Control Bits**

- bit 7 : **IBF:** Input Buffer Full Status bit  
 1 = A word has been received and is waiting to be read by the CPU  
 0 = No word has been received
- bit 6 : **OBF:** Output Buffer Full Status bit  
 1 = The output buffer still holds a previously written word  
 0 = The output buffer has been read
- bit 5 : **IBOV:** Input Buffer Overflow Detect bit (in microprocessor mode)  
 1 = A write occurred when a previously input word has not been read (must be cleared in software)  
 0 = No overflow occurred

- bit 4 : **PSPMODE:** Parallel Slave Port Mode Select bit  
 1 = Parallel slave port mode  
 0 = General purpose I/O mode

bit 3 : **Unimplemented:** Read as '0'

**PORTE Data Direction Bits**

- bit 2 : **Bit2:** Direction Control bit for pin RE2/ $\overline{CS}$ /AN7  
 1 = Input  
 0 = Output
- bit 1 : **Bit1:** Direction Control bit for pin RE1/ $\overline{WR}$ /AN6  
 1 = Input  
 0 = Output
- bit 0 : **Bit0:** Direction Control bit for pin RE0/ $\overline{RD}$ /AN5  
 1 = Input  
 0 = Output

**TABLE 3-9: PORTE FUNCTIONS**

Name	Bit#	Buffer Type	Function
RE0/ $\overline{RD}$ /AN5	bit0	ST/TTL <sup>(1)</sup>	Input/output port pin or read control input in parallel slave port mode or analog input: $\overline{RD}$ 1 = Not a read operation 0 = Read operation. Reads PORTD register (if chip selected)
RE1/ $\overline{WR}$ /AN6	bit1	ST/TTL <sup>(1)</sup>	Input/output port pin or write control input in parallel slave port mode or analog input: $\overline{WR}$ 1 = Not a write operation 0 = Write operation. Writes PORTD register (if chip selected)
RE2/ $\overline{CS}$ /AN7	bit2	ST/TTL <sup>(1)</sup>	Input/output port pin or chip select control input in parallel slave port mode or analog input: $\overline{CS}$ 1 = Device is not selected 0 = Device is selected

Legend: ST = Schmitt Trigger input TTL = TTL input

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

**TABLE 3-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE**

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
09h	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000

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

# PIC16F87X

## 3.6 Parallel Slave Port

The Parallel Slave Port is not implemented on the PIC16F873 or PIC16F876.

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

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

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

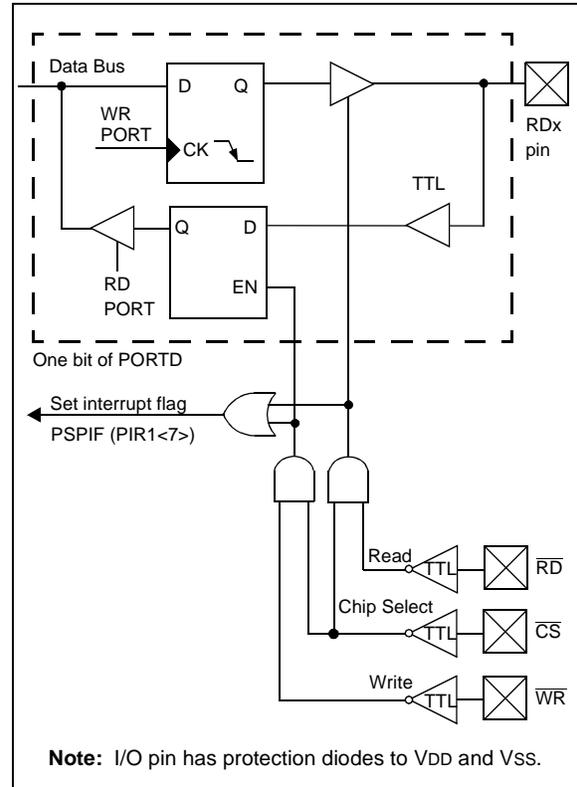
A write to the PSP occurs when both the  $\overline{CS}$  and  $\overline{WR}$  lines are first detected low. When either the  $\overline{CS}$  or  $\overline{WR}$  lines become high (level triggered), the Input Buffer Full (IBF) status flag bit (TRISE<7>) is set on the Q4 clock cycle, following the next Q2 cycle, to signal the write is complete (Figure 3-10). The interrupt flag bit PSPIF (PIR1<7>) is also set on the same Q4 clock cycle. IBF can only be cleared by reading the PORTD input latch. The Input Buffer Overflow (IBOV) status flag bit (TRISE<5>) is set if a second write to the PSP is attempted when the previous byte has not been read out of the buffer.

A read from the PSP occurs when both the  $\overline{CS}$  and  $\overline{RD}$  lines are first detected low. The Output Buffer Full (OBF) status flag bit (TRISE<6>) is cleared immediately (Figure 3-11) indicating that the PORTD latch is waiting to be read by the external bus. When either the  $\overline{CS}$  or  $\overline{RD}$  pin becomes high (level triggered), the interrupt flag bit PSPIF is set on the Q4 clock cycle, following the next Q2 cycle, indicating that the read is complete. OBF remains low until data is written to PORTD by the user firmware.

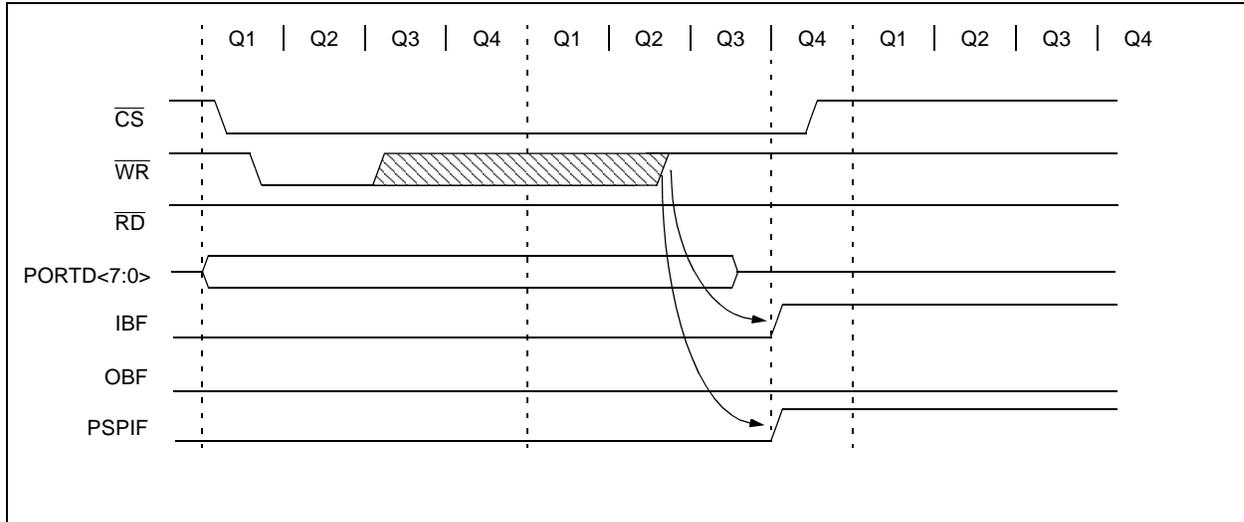
When not in PSP mode, the IBF and OBF bits are held clear. However, if flag bit IBOV was previously set, it must be cleared in firmware.

An interrupt is generated and latched into flag bit PSPIF when a read or write operation is completed. PSPIF must be cleared by the user in firmware and the interrupt can be disabled by clearing the interrupt enable bit PSPIE (PIE1<7>).

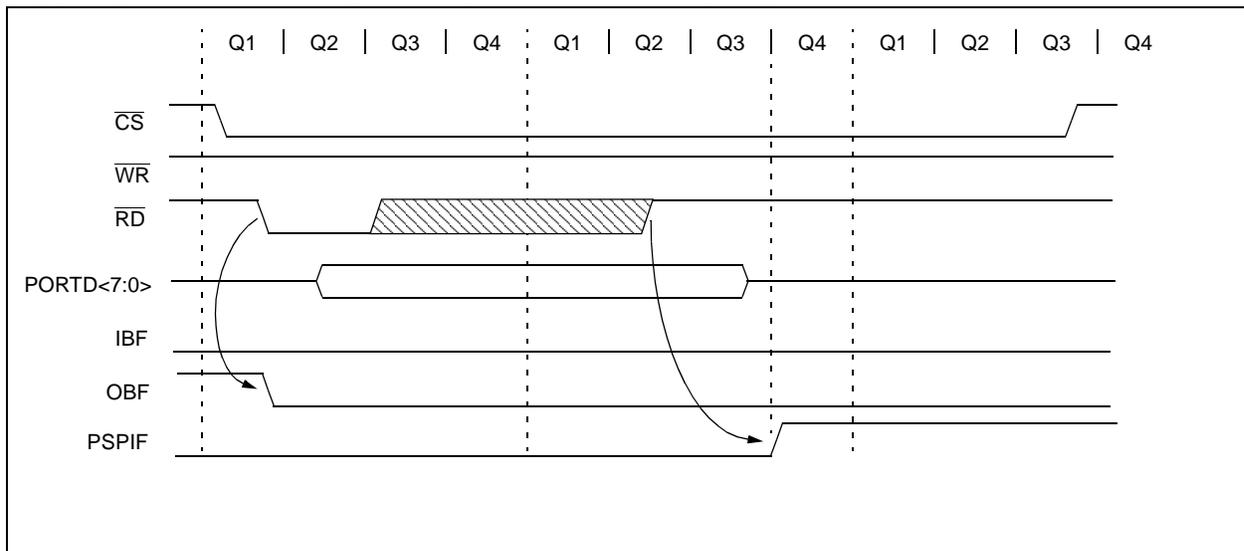
FIGURE 3-9: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)



**FIGURE 3-10: PARALLEL SLAVE PORT WRITE WAVEFORMS**



**FIGURE 3-11: PARALLEL SLAVE PORT READ WAVEFORMS**



**TABLE 3-11: REGISTERS ASSOCIATED WITH PARALLEL SLAVE PORT**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets	
08h	PORTD	Port data latch when written: Port pins when read								xxxx xxxx	uuuu uuuu	
09h	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu	
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits				0000 -111	0000 -111
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000	
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000	
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000	

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by the Parallel Slave Port.

# PIC16F87X

---

NOTES:

## 4.0 DATA EEPROM AND FLASH PROGRAM MEMORY

The Data EEPROM and FLASH Program Memory are readable and writable during normal operation over the entire VDD range. A bulk erase operation may not be issued from user code (which includes removing code protection). The data memory is not directly mapped in the register file space. Instead it is indirectly addressed through the Special Function Registers (SFR).

There are six SFRs used to read and write the program and data EEPROM memory. These registers are:

- EECON1
- EECON2
- EEDATA
- EEDATH
- EEADR
- EEADRH

The EEPROM data memory allows byte read and write. When interfacing to the data memory block, EEDATA holds the 8-bit data for read/write and EEADR holds the address of the EEPROM location being accessed. The registers EEDATH and EEADRH are not used for data EEPROM access. These devices have up to 256 bytes of data EEPROM with an address range from 0h to FFh.

The EEPROM data memory is rated for high erase/write cycles. The write time is controlled by an on-chip timer. The write time will vary with voltage and temperature, as well as from chip-to-chip. Please refer to the specifications for exact limits.

The program memory allows word reads and writes. Program memory access allows for checksum calculation and calibration table storage. A byte or word write automatically erases the location and writes the new data (erase before write). Writing to program memory will cease operation until the write is complete. The program memory cannot be accessed during the write, therefore code cannot execute. During the write operation, the oscillator continues to clock the peripherals, and therefore they continue to operate. Interrupt events will be detected and essentially “queued” until the write is completed. When the write completes, the next instruction in the pipeline is executed and the branch to the interrupt vector address will occur.

When interfacing to the program memory block, the EEDATH:EEDATA registers form a two byte word, which holds the 14-bit data for read/write. The EEADRH:EEADR registers form a two byte word, which holds the 13-bit address of the EEPROM location being accessed. These devices can have up to 8K words of program EEPROM with an address range from 0h to 3FFFh. The unused upper bits in both the EEDATH and EEDATA registers all read as “0’s”.

The value written to program memory does not need to be a valid instruction. Therefore, up to 14-bit numbers can be stored in memory for use as calibration parameters, serial numbers, packed 7-bit ASCII, etc. Executing a program memory location containing data that forms an invalid instruction results in a NOP.

### 4.1 EEADR

The address registers can address up to a maximum of 256 bytes of data EEPROM or up to a maximum of 8K words of program FLASH.

When selecting a program address value, the MSByte of the address is written to the EEADRH register and the LSByte is written to the EEADR register. When selecting a data address value, only the LSByte of the address is written to the EEADR register.

On the PIC16F873/874 devices with 128 bytes of EEPROM, the MSbit of the EEADR must always be cleared to prevent inadvertent access to the wrong location. This also applies to the program memory. The upper MSbits of EEADRH must always be clear.

### 4.2 EECON1 and EECON2 Registers

EECON1 is the control register for memory accesses.

EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the memory write sequence.

Control bit EEPGD determines if the access will be a program or a data memory access. When clear, any subsequent operations will operate on the data memory. When set, any subsequent operations will operate on the program memory.

Control bits RD and WR initiate read and write operations, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at the completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental or premature termination of a write operation.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a MCLR reset or a WDT time-out reset during normal operation. In these situations, following reset, the user can check the WRERR bit and rewrite the location. The value of the data and address registers and the EEPGD bit remains unchanged.

Interrupt flag bit EEIF, in the PIR2 register, is set when write is complete. It must be cleared in software.

# PIC16F87X

## REGISTER 4-1: EECON1 REGISTER (ADDRESS 18Ch)

R/W-x	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	—	—	—	WRERR	WREN	WR	RD
bit7							bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit,  
read as '0'  
- n = Value at POR reset

- bit 7: **EEPGD**: Program / Data EEPROM Select bit  
1 = Accesses Program memory  
0 = Accesses data memory  
(This bit cannot be changed while a read or write operation is in progress)
- bit 6:4: **Unimplemented**: Read as '0'
- bit 3: **WRERR**: EEPROM Error Flag bit  
1 = A write operation is prematurely terminated  
(any MCLR reset or any WDT reset during normal operation)  
0 = The write operation completed
- bit 2: **WREN**: EEPROM Write Enable bit  
1 = Allows write cycles  
0 = Inhibits write to the EEPROM
- bit 1: **WR**: Write Control bit  
1 = initiates a write cycle. (The bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.  
0 = Write cycle to the EEPROM is complete
- bit 0: **RD**: Read Control bit  
1 = Initiates an EEPROM read RD is cleared in hardware. The RD bit can only be set (not cleared) in software.  
0 = Does not initiate an EEPROM read

## 4.3 Reading the Data EEPROM Memory

To read a data memory location, the user must write the address to the EEADR register, clear the EEPGD control bit (EECON1<7>) and then set control bit RD (EECON1<0>). The data is available in the very next instruction cycle of the EEDATA register, therefore it can be read by the next instruction. EEDATA will hold this value until another read operation or until it is written to by the user (during a write operation).

### EXAMPLE 4-1: DATA EEPROM READ

```
BSF STATUS, RP1 ;
BCF STATUS, RP0 ;Bank 2
MOVLW DATA_EE_ADDR ;
MOVWF EEADR ;Data Memory Address to read
BSF STATUS, RP0 ;Bank 3
BCF EECON1, EEPGD ;Point to DATA memory
BSF EECON1, RD ;EEPROM Read
BCF STATUS, RP0 ;Bank 2
MOVF EEDATA, W ;W = EEDATA
```

## 4.4 Writing to the Data EEPROM Memory

To write an EEPROM data location, the address must first be written to the EEADR register and the data written to the EEDATA register. Then the sequence in Example 4-2 must be followed to initiate the write cycle.

### EXAMPLE 4-2: DATA EEPROM WRITE

```
BSF STATUS, RP1 ;
BCF STATUS, RP0 ; Bank 2
MOVLW DATA_EE_ADDR ;
MOVWF EEADR ; Data Memory Address to write
MOVLW DATA_EE_DATA ;
MOVWF EEDATA ; Data Memory Value to write
BSF STATUS, RP0 ; Bank 3
BCF EECON1, EEPGD ; Point to DATA memory
BSF EECON1, WREN ; Enable writes

BCF INTCON, GIE ; Disable Interrupts
MOVLW 55h ;
Required MOVWF EECON2 ; Write 55h
Sequence MOVLW AAh ;
MOVWF EECON2 ; Write AAh
BSF EECON1, WR ; Set WR bit to begin write
BSF INTCON, GIE ; Enable Interrupts

SLEEP ; Wait for interrupt to signal write complete
BCF EECON1, WREN ; Disable writes
```

The write will not initiate if the above sequence is not exactly followed (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. It is strongly recommended that interrupts be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable writes. This mechanism prevents accidental writes to data EEPROM due to unexpected code execution (i.e., runaway programs). The WREN bit should be kept clear at all times, except when updating the EEPROM. The WREN bit is not cleared by hardware

After a write sequence has been initiated, clearing the WREN bit will not affect the current write cycle. The WR bit will be inhibited from being set unless the WREN bit

is set. The WREN bit must be set on a previous instruction. Both WR and WREN cannot be set with the same instruction.

At the completion of the write cycle, the WR bit is cleared in hardware and the EEPROM Write Complete Interrupt Flag bit (EEIF) is set. EEIF must be cleared by software.

## 4.5 Reading the FLASH Program Memory

A program memory location may be read by writing two bytes of the address to the EEADR and EEADRH registers, setting the EEPGD control bit (EECON1<7>) and then setting control bit RD (EECON1<0>). Once the read control bit is set, the microcontroller will use the next two instruction cycles to read the data. The

data is available in the EEDATA and EEDATH registers after the second NOP instruction. Therefore, it can be read as two bytes in the following instructions. The EEDATA and EEDATH registers will hold this value until another read operation or until it is written to by the user (during a write operation).

### EXAMPLE 4-3: FLASH PROGRAM READ

```
BSF    STATUS, RP1    ;
BCF    STATUS, RP0    ; Bank 2
MOVLW  ADDRH          ;
MOVWF  EEADRH         ; MSByte of Program Address to read
MOVLW  ADDRL          ;
MOVWF  EEADR          ; LSByte of Program Address to read
BSF    STATUS, RP0    ; Bank 3
BSF    EECON1, EEPGD  ; Point to PROGRAM memory
Required BSF    EECON1, RD    ; EEPROM Read
Sequence
NOP                                ; memory is read in the next two cycles after BSF EECON1,RD
NOP                                ;

BCF    STATUS, RP0    ; Bank 2

MOVF   EEDATA, W      ; W = LSByte of Program EEDATA
MOVF   EEDATH, W      ; W = MSByte of Program EEDATA
```

## 4.6 Writing to the FLASH Program Memory

A word of the FLASH program memory may only be written to if the word is in a non-code protected segment of memory and the WRT configuration bit is set. To write a FLASH program location, the first two bytes of the address must be written to the EEADR and EEADRH registers and two bytes of the data to the EEDATA and EEDATH registers, set the EEPGD con-

trol bit (EECON1<7>), and then set control bit WR (EECON1<1>). The sequence in Example 4-4 must be followed to initiate a write to program memory.

The microcontroller will then halt internal operations during the next two instruction cycles for the TPEW (parameter D133) in which the write takes place. This is not SLEEP mode, as the clocks and peripherals will continue to run. Therefore, the two instructions following the “BSF EECON, WR” should be NOP instructions. After the write cycle, the microcontroller will resume operation with the 3rd instruction after the EECON1 write instruction.

### EXAMPLE 4-4: FLASH PROGRAM WRITE

```

BSF    STATUS, RP1    ;
BCF    STATUS, RP0    ; Bank 2
MOVLW  ADDRH         ;
MOVWF  EEADRH        ; MSByte of Program Address to read
MOVLW  ADDRHL        ;
MOVWF  EEADR         ; LSByte of Program Address to read
MOVLW  DATAH        ;
MOVWF  EEDATH        ; MS Program Memory Value to write
MOVLW  DATAL         ;
MOVWF  EEDATA        ; LS Program Memory Value to write
BSF    STATUS, RP0    ; Bank 3
BSF    EECON1, EEPGD  ; Point to PROGRAM memory
BSF    EECON1, WREN   ; Enable writes

BCF    INTCON, GIE    ; Disable Interrupts

Required Sequence
MOVLW  55h           ;
MOVWF  EECON2        ; Write 55h
MOVLW  AAh           ;
MOVWF  EECON2        ; Write AAh
BSF    EECON1, WR    ; Set WR bit to begin write

NOP
NOP
; Instructions here are ignored by the microcontroller
; Microcontroller will halt operation and wait for
; a write complete. After the write
; the microcontroller continues with 3rd instruction

BSF    INTCON, GIE    ; Enable Interrupts
BCF    EECON1, WREN   ; Disable writes

```

## 4.7 Write Verify

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

Generally a write failure will be a bit which was written as a '1', but reads back as a '0' (due to leakage off the bit).

# PIC16F87X

## 4.8 Protection Against Spurious Write

### 4.8.1 EEPROM DATA MEMORY

There are conditions when the device may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built-in. On power-up, the WREN bit is cleared. Also, the Power-up Timer (72 ms duration) prevents EEPROM write.

The write initiate sequence and the WREN bit together help prevent an accidental write during brown-out, power glitch, or software malfunction.

### 4.8.2 PROGRAM FLASH MEMORY

To protect against spurious writes to FLASH program memory, the WRT bit in the configuration word may be programmed to '0' to prevent writes. The write initiate sequence must also be followed. WRT and the configuration word cannot be programmed by user code, only through the use of an external programmer.

## 4.9 Operation during Code Protect

Each reprogrammable memory block has its own code protect mechanism. External Read and Write operations are disabled if either of these mechanisms are enabled.

### 4.9.1 DATA EEPROM MEMORY

The microcontroller itself can both read and write to the internal Data EEPROM, regardless of the state of the code protect configuration bit.

### 4.9.2 PROGRAM FLASH MEMORY

The microcontroller can read and execute instructions out of the internal FLASH program memory, regardless of the state of the code protect configuration bits. However the WRT configuration bit and the code protect bits have different effects on writing to program memory. Table 4-1 shows the various configurations and status of reads and writes. To erase the WRT or code protection bits in the configuration word requires that the device be fully erased.

**TABLE 4-1: READ/WRITE STATE OF INTERNAL FLASH PROGRAM MEMORY**

Configuration Bits			Memory Location	Internal Read	Internal Write	ICSP Read	ICSP Write
CP1	CP0	WRT					
0	0	x	All program memory	Yes	No	No	No
0	1	0	Unprotected areas	Yes	No	Yes	No
0	1	0	Protected areas	Yes	No	No	No
0	1	1	Unprotected areas	Yes	Yes	Yes	No
0	1	1	Protected areas	Yes	No	No	No
1	0	0	Unprotected areas	Yes	No	Yes	No
1	0	0	Protected areas	Yes	No	No	No
1	0	1	Unprotected areas	Yes	Yes	Yes	No
1	0	1	Protected areas	Yes	No	No	No
1	1	0	All program memory	Yes	No	Yes	Yes
1	1	1	All program memory	Yes	Yes	Yes	Yes

**TABLE 4-2: REGISTERS ASSOCIATED WITH DATA EEPROM/PROGRAM FLASH**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
10Dh	EEADR	EEPROM address register							xxxx xxxx	uuuu uuuu	
10Fh	EEADRH	—	—	—	EEPROM address high			xxxx xxxx	uuuu uuuu		
10Ch	EEDATA	EEPROM data register							xxxx xxxx	uuuu uuuu	
10Eh	EEDATH	—	—	EEPROM data register high					xxxx xxxx	uuuu uuuu	
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	x--- u000
18Dh	EECON2	EEPROM control register2 (not a physical register)									
8Dh	PIE2	—	(1)	—	EEIE	BCLIE	—	—	CCP2IE	-r-0 0--0	-r-0 0--0
0Dh	PIR2	—	(1)	—	EEIF	BCLIF	—	—	CCP2IF	-r-0 0--0	-r-0 0--0

Legend: x = unknown, u = unchanged, r = reserved, - = unimplemented read as '0'. Shaded cells are not used during FLASH/EEPROM access.

**Note 1:** These bits are reserved; always maintain these bits clear.

## 5.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 5-1 is a block diagram of the Timer0 module and the prescaler shared with the WDT.

Additional information on the Timer0 module is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

Timer mode is selected by clearing bit T0CS (OPTION\_REG<5>). In timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If the TMR0 register is written, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0 register.

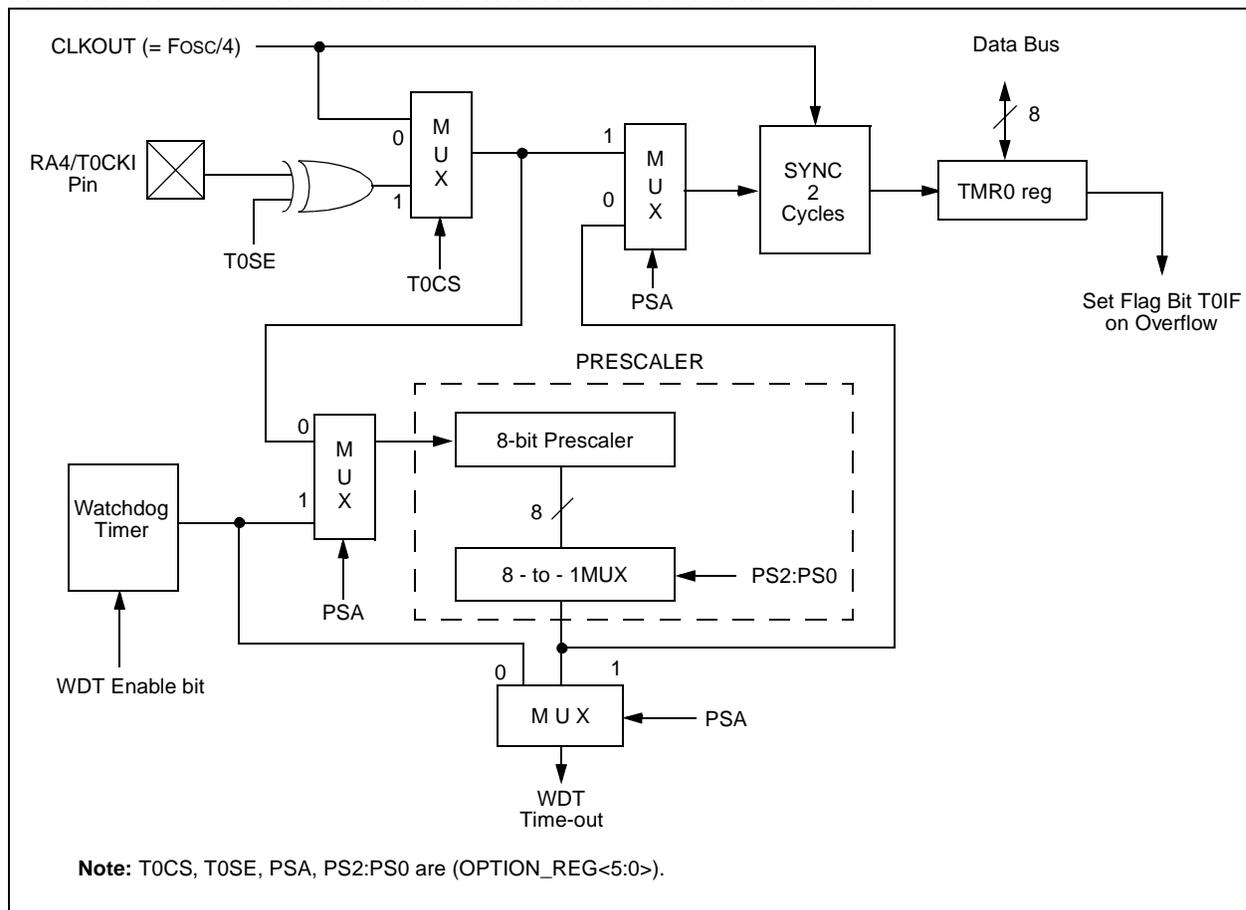
Counter mode is selected by setting bit T0CS (OPTION\_REG<5>). In counter mode, Timer0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit T0SE (OPTION\_REG<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 5.2.

The prescaler is mutually exclusively shared between the Timer0 module and the watchdog timer. The prescaler is not readable or writable. Section 5.3 details the operation of the prescaler.

### 5.1 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h. This overflow sets bit T0IF (INTCON<2>). The interrupt can be masked by clearing bit T0IE (INTCON<5>). Bit T0IF must be cleared in software by the Timer0 module interrupt service routine before re-enabling this interrupt. The TMR0 interrupt cannot awaken the processor from SLEEP since the timer is shut off during SLEEP.

**FIGURE 5-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER**



# PIC16F87X

## 5.2 Using Timer0 with an External Clock

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

## 5.3 Prescaler

There is only one prescaler available, which is mutually exclusively shared between the Timer0 module and the watchdog timer. A prescaler assignment for the Timer0

module means that there is no prescaler for the watchdog timer, and vice-versa. This prescaler is not readable or writable (see Figure 5-1).

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

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

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

### REGISTER 5-1: OPTION\_REG REGISTER

	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:	<b>RBPU</b>							
bit 6:	<b>INTEDG</b>							
bit 5:	<b>T0CS:</b> TMR0 Clock Source Select bit 1 = Transition on T0CKI pin 0 = Internal instruction cycle clock (CLKOUT)							
bit 4:	<b>T0SE:</b> TMR0 Source Edge Select bit 1 = Increment on high-to-low transition on T0CKI pin 0 = Increment on low-to-high transition on T0CKI pin							
bit 3:	<b>PSA:</b> Prescaler Assignment bit 1 = Prescaler is assigned to the WDT 0 = Prescaler is assigned to the Timer0 module							
bit 2-0:	<b>PS2:PS0:</b> 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			

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

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

**TABLE 5-1: REGISTERS ASSOCIATED WITH TIMER0**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
01h,101h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
81h,181h	OPTION_REG	RBP $\bar{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

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

# PIC16F87X

---

NOTES:

## 6.0 TIMER1 MODULE

The Timer1 module is a 16-bit timer/counter consisting of two 8-bit registers (TMR1H and TMR1L), which are readable and writable. The TMR1 Register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The TMR1 Interrupt, if enabled, is generated on overflow, which is latched in interrupt flag bit TMR1IF (PIR1<0>). This interrupt can be enabled/disabled by setting/clearing TMR1 interrupt enable bit TMR1IE (PIE1<0>).

Timer1 can operate in one of two modes:

- As a timer
- As a counter

The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>).

In timer mode, Timer1 increments every instruction cycle. In counter mode, it increments on every rising edge of the external clock input.

Timer1 can be enabled/disabled by setting/clearing control bit TMR1ON (T1CON<0>).

Timer1 also has an internal "reset input". This reset can be generated by either of the two CCP modules (Section 8.0). Register 6-1 shows the Timer1 control register.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI/CCP2 and RC0/T1OSO/T1CKI pins become inputs. That is, the TRISC<1:0> value is ignored.

Additional information on timer modules is available in the PICmicro™ Mid-range MCU Family Reference Manual (DS33023).

### REGISTER 6-1: T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	
bit7								bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

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

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 Control bit  
1 = Oscillator is enabled  
0 = Oscillator is shut off (The oscillator inverter is turned off to eliminate power drain)

bit 2: **T1SYNC:** Timer1 External Clock Input Synchronization Control bit

TMR1CS = 1  
1 = Do not synchronize external clock input  
0 = Synchronize external clock input

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/T1CKI (on the rising edge)  
0 = Internal clock (Fosc/4)

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

# PIC16F87X

## 6.1 Timer1 Operation in Timer Mode

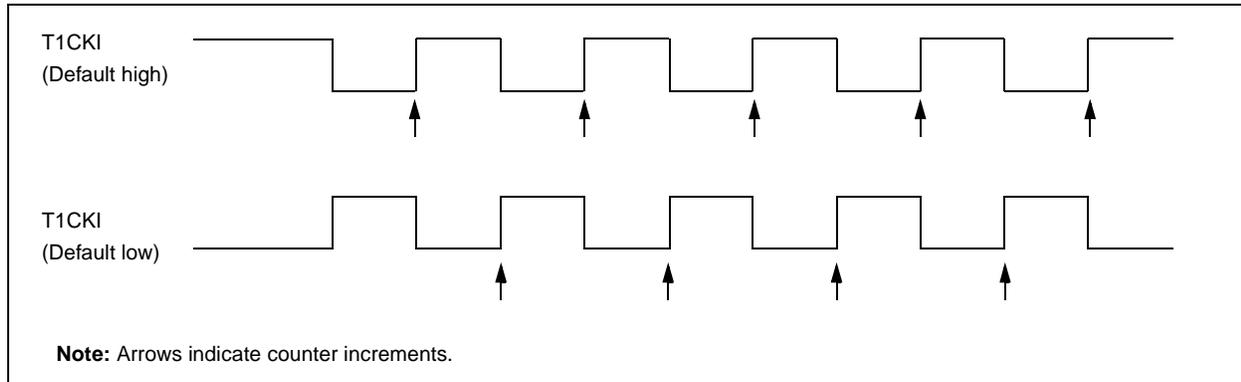
Timer mode is selected by clearing the TMR1CS (T1CON<1>) bit. In this mode, the input clock to the timer is FOSC/4. The synchronize control bit  $\overline{T1SYNC}$  (T1CON<2>) has no effect since the internal clock is always in sync.

## 6.2 Timer1 Counter Operation

Timer1 may operate in asynchronous or usynchronous mode depending on the setting of the TMR1CS bit.

When Timer1 is being incremented via an external source, increments occur on a rising edge. After Timer1 is enabled in counter mode, the module must first have a falling edge before the counter begins to increment.

FIGURE 6-1: TIMER1 INCREMENTING EDGE



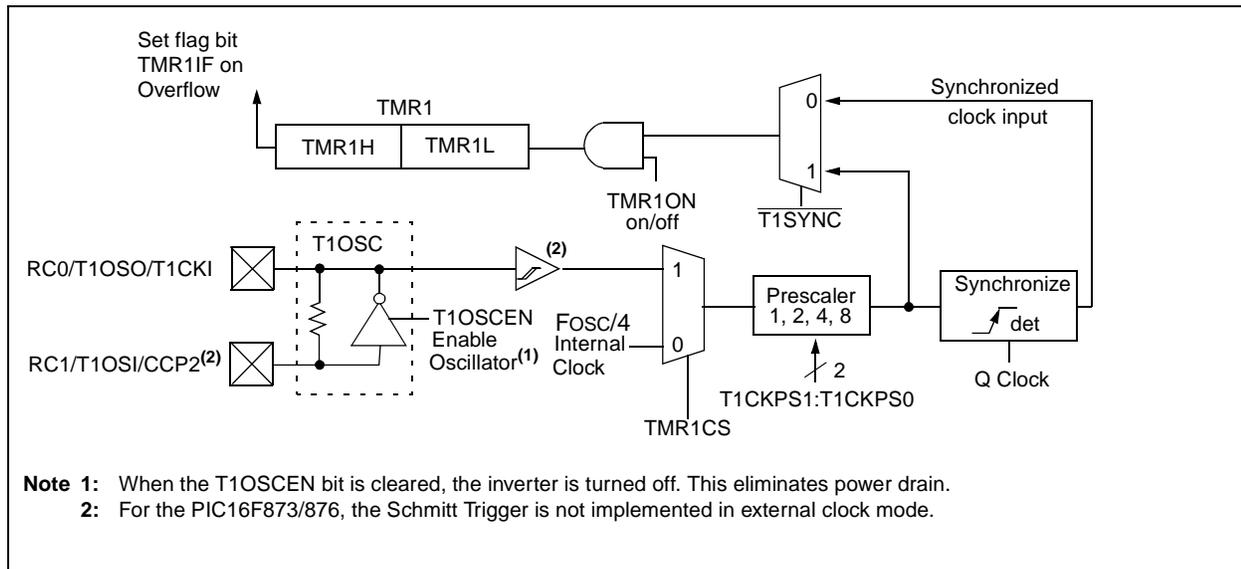
## 6.3 Timer1 Operation in Synchronized Counter Mode

Counter mode is selected by setting bit TMR1CS. In this mode, the timer increments on every rising edge of clock input on pin RC1/T1OSI/CCP2, when bit T1OSCEN is set, or on pin RC0/T1OSO/T1CKI, when bit T1OSCEN is cleared.

If  $\overline{T1SYNC}$  is cleared, then the external clock input is synchronized with internal phase clocks. The synchronization is done after the prescaler stage. The prescaler stage is an asynchronous ripple-counter.

In this configuration, during SLEEP mode, Timer1 will not increment even if the external clock is present, since the synchronization circuit is shut off. The prescaler however will continue to increment.

FIGURE 6-2: TIMER1 BLOCK DIAGRAM



## 6.4 Timer1 Operation in Asynchronous Counter Mode

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

In asynchronous counter mode, Timer1 can not be used as a time-base for capture or compare operations.

### 6.4.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

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

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

Reading the 16-bit value requires some care. Examples 12-2 and 12-3 in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023) show how to read and write Timer1 when it is running in asynchronous mode.

## 6.5 Timer1 Oscillator

A crystal oscillator circuit is built-in between pins T1OSI (input) and T1OSO (amplifier output). It is enabled by setting control bit T1OSCEN (T1CON<3>). The oscillator is a low power oscillator rated up to 200 kHz. It will continue to run during SLEEP. It is primarily intended for use with a 32 kHz crystal. Table 6-1 shows the capacitor selection for the Timer1 oscillator.

The Timer1 oscillator is identical to the LP oscillator. The user must provide a software time delay to ensure proper oscillator start-up.

**TABLE 6-1: CAPACITOR SELECTION FOR THE TIMER1 OSCILLATOR**

Osc Type	Freq	C1	C2
LP	32 kHz	33 pF	33 pF
	100 kHz	15 pF	15 pF
	200 kHz	15 pF	15 pF
These values are for design guidance only.			
<b>Crystals Tested:</b>			
32.768 kHz	Epson C-001R32.768K-A	± 20 PPM	
100 kHz	Epson C-2 100.00 KC-P	± 20 PPM	
200 kHz	STD XTL 200.000 kHz	± 20 PPM	
<b>Note 1:</b> Higher capacitance increases the stability of oscillator but also increases the start-up time. <b>2:</b> Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.			

## 6.6 Resetting Timer1 using a CCP Trigger Output

If the CCP1 or CCP2 module is configured in compare mode to generate a “special event trigger” (CCP1M3:CCP1M0 = 1011), this signal will reset Timer1.

**Note:** The special event triggers from the CCP1 and CCP2 modules will not set interrupt flag bit TMR1IF (PIR1<0>).

Timer1 must be configured for either timer or synchronized counter mode to take advantage of this feature. If Timer1 is running in asynchronous counter mode, this reset operation may not work.

In the event that a write to Timer1 coincides with a special event trigger from CCP1 or CCP2, the write will take precedence.

In this mode of operation, the CCPRxH:CCPRxL register pair effectively becomes the period register for Timer1.

## 6.7 Resetting of Timer1 Register Pair (TMR1H, TMR1L)

TMR1H and TMR1L registers are not reset to 00h on a POR or any other reset except by the CCP1 and CCP2 special event triggers.

T1CON register is reset to 00h on a Power-on Reset or a Brown-out Reset, which shuts off the timer and leaves a 1:1 prescale. In all other resets, the register is unaffected.

## 6.8 Timer1 Prescaler

The prescaler counter is cleared on writes to the TMR1H or TMR1L registers.

# PIC16F87X

**TABLE 6-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
0Bh,8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN $\bar{C}$	TMR1CS	TMR1ON	--00 0000	--uu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by the Timer1 module.

**Note 1:** Bits PSPIE and PSPIF are reserved on the PIC16F873/874; always maintain these bits clear.

## 7.0 TIMER2 MODULE

Timer2 is an 8-bit timer with a prescaler and a postscaler. It can be used as the PWM time-base for the PWM mode of the CCP module(s). The TMR2 register is readable and writable, and is cleared on any device reset.

The input clock ( $F_{osc}/4$ ) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits T2CKPS1:T2CKPS0 (T2CON<1:0>).

The Timer2 module has an 8-bit period register PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is initialized to FFh upon reset.

The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit TMR2IF, (PIR1<1>)).

Timer2 can be shut off by clearing control bit TMR2ON (T2CON<2>) to minimize power consumption.

Register 7-1 shows the Timer2 control register.

Additional information on timer modules is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

## 7.1 Timer2 Prescaler and Postscaler

The prescaler and postscaler counters are cleared when any of the following occurs:

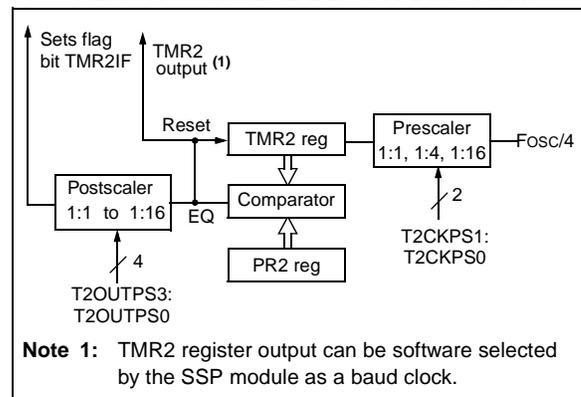
- a write to the TMR2 register
- a write to the T2CON register
- any device reset (POR, MCLR reset, WDT reset or BOR)

TMR2 is not cleared when T2CON is written.

## 7.2 Output of TMR2

The output of TMR2 (before the postscaler) is fed to the SSPort module, which optionally uses it to generate shift clock.

**FIGURE 7-1: TIMER2 BLOCK DIAGRAM**



**REGISTER 7-1: T2CON: TIMER2 CONTROL REGISTER (ADDRESS 12h)**

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit7							bit0

**bit 7: Unimplemented:** Read as '0'  
**bit 6-3: TOUTPS3:TOUTPS0:** Timer2 Output Postscale Select bits  
 0000 = 1:1 Postscale  
 0001 = 1:2 Postscale  
 0010 = 1:3 Postscale  
 •  
 •  
 •  
 1111 = 1:16 Postscale  
**bit 2: TMR2ON:** Timer2 On bit  
 1 = Timer2 is on  
 0 = Timer2 is off  
**bit 1-0: T2CKPS1:T2CKPS0:** Timer2 Clock Prescale Select bits  
 00 = Prescaler is 1  
 01 = Prescaler is 4  
 1x = Prescaler is 16

R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 - n = Value at POR reset

# PIC16F87X

**TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
11h	TMR2	Timer2 module's register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
92h	PR2	Timer2 Period Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by the Timer2 module.

**Note 1:** Bits PSPIE and PSPIF are reserved on the PIC16F873/874; always maintain these bits clear.

## 8.0 CAPTURE/COMPARE/PWM MODULES

Each Capture/Compare/PWM (CCP) module contains a 16-bit register which can operate as a:

- 16-bit Capture register
- 16-bit Compare register
- PWM master/slave Duty Cycle register

Both the CCP1 and CCP2 modules are identical in operation, with the exception being the operation of the special event trigger. [Table 8-1](#) and [Table 8-2](#) show the resources and interactions of the CCP module(s). In the following sections, the operation of a CCP module is described with respect to CCP1. CCP2 operates the same as CCP1, except where noted.

### CCP1 Module:

Capture/Compare/PWM Register1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (high byte). The CCP1CON register controls the operation of CCP1. The special event trigger is generated by a compare match and will reset Timer1.

### CCP2 Module:

Capture/Compare/PWM Register1 (CCPR2) is comprised of two 8-bit registers: CCPR2L (low byte) and CCPR2H (high byte). The CCP2CON register controls the operation of CCP2. The special event trigger is generated by a compare match and will reset Timer1 and start an A/D conversion (if the A/D module is enabled).

Additional information on CCP modules is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023) and in Application Note 594, "Using the CCP Modules" (DS00594).

**TABLE 8-1: CCP MODE - TIMER RESOURCES REQUIRED**

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

**TABLE 8-2: INTERACTION OF TWO CCP MODULES**

CCPx Mode	CCPy Mode	Interaction
Capture	Capture	Same TMR1 time-base.
Capture	Compare	The compare should be configured for the special event trigger, which clears TMR1.
Compare	Compare	The compare(s) should be configured for the special event trigger, which clears TMR1.
PWM	PWM	The PWMs will have the same frequency and update rate (TMR2 interrupt).
PWM	Capture	None.
PWM	Compare	None.

# PIC16F87X

## REGISTER 8-1: CCP1CON REGISTER/CCP2CON REGISTER (ADDRESS: 17h/1dh)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CCPxX	CCPY	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit7							bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

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

bit 5-4: **CCPxX:CCPY:** PWM Least Significant bits  
Capture Mode: Unused  
Compare Mode: Unused  
PWM Mode: These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPxL.

bit 3-0: **CCPxM3:CCPxM0:** CCPx Mode Select bits  
0000 = Capture/Compare/PWM off (resets CCPx module)  
0100 = Capture mode, every falling edge  
0101 = Capture mode, every rising edge  
0110 = Capture mode, every 4th rising edge  
0111 = Capture mode, every 16th rising edge  
1000 = Compare mode, set output on match (CCPxIF bit is set)  
1001 = Compare mode, clear output on match (CCPxIF bit is set)  
1010 = Compare mode, generate software interrupt on match (CCPxIF bit is set, CCPx pin is unaffected)  
1011 = Compare mode, trigger special event (CCPxIF bit is set, CCPx pin is unaffected); CCP1 resets TMR1; CCP2 resets TMR1 and starts an A/D conversion (if A/D module is enabled)  
11xx = PWM mode

## 8.1 Capture Mode

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of the TMR1 register when an event occurs on pin RC2/CCP1. An event is defined as:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

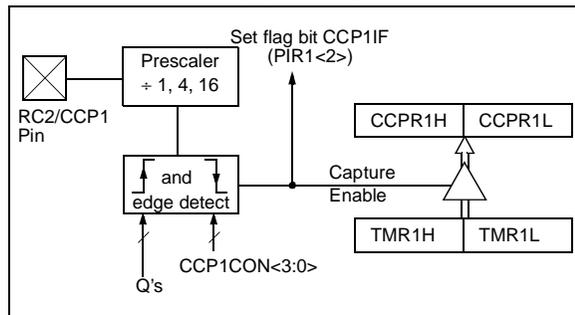
An event is selected by control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). When a capture is made, the interrupt request flag bit CCP1IF (PIR1<2>) is set. The interrupt flag must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value will be lost.

### 8.1.1 CCP PIN CONFIGURATION

In Capture mode, the RC2/CCP1 pin should be configured as an input by setting the TRISC<2> bit.

**Note:** If the RC2/CCP1 pin is configured as an output, a write to the port can cause a capture condition.

**FIGURE 8-1: CAPTURE MODE OPERATION BLOCK DIAGRAM**



### 8.1.2 TIMER1 MODE SELECTION

Timer1 must be running in timer mode or synchronized counter mode for the CCP module to use the capture feature. In asynchronous counter mode, the capture operation may not work.

### 8.1.3 SOFTWARE INTERRUPT

When the capture mode is changed, a false capture interrupt may be generated. The user should keep bit CCP1IE (PIE1<2>) clear to avoid false interrupts and should clear the flag bit CCP1IF following any such change in operating mode.

### 8.1.4 CCP PRESCALER

There are four prescaler settings, specified by bits CCP1M3:CCP1M0. Whenever the CCP module is turned off, or the CCP module is not in capture mode, the prescaler counter is cleared. Any reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared, therefore, the first capture may be from a non-zero prescaler. [Example 8-1](#) shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

### EXAMPLE 8-1: CHANGING BETWEEN CAPTURE PRESCALERS

```

CLRf   CCP1CON   ;Turn CCP module off
MOVLW  NEW_CAPT_PS ;Load the W reg with
                    ; the new prescaler
MOVWF  CCP1CON   ; move value and CCP ON
                    ;Load CCP1CON with this
                    ; value
    
```

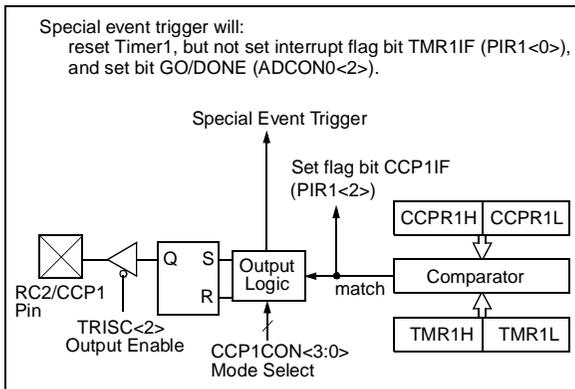
## 8.2 Compare Mode

In Compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1 register pair value. When a match occurs, the RC2/CCP1 pin is:

- Driven high
- Driven low
- Remains unchanged

The action on the pin is based on the value of control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). At the same time, interrupt flag bit CCP1IF is set.

**FIGURE 8-2: COMPARE MODE OPERATION BLOCK DIAGRAM**



### 8.2.1 CCP PIN CONFIGURATION

The user must configure the RC2/CCP1 pin as an output by clearing the TRISC<2> bit.

**Note:** Clearing the CCP1CON register will force the RC2/CCP1 compare output latch to the default low level. This is not the data latch.

### 8.2.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode if the CCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

### 8.2.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen, the CCP1 pin is not affected. The CCP1IF bit is set causing a CCP interrupt (if enabled).

### 8.2.4 SPECIAL EVENT TRIGGER

In this mode, an internal hardware trigger is generated, which may be used to initiate an action.

The special event trigger output of CCP1 resets the TMR1 register pair. This allows the CCPR1 register to effectively be a 16-bit programmable period register for Timer1.

The special event trigger output of CCP2 resets the TMR1 register pair and starts an A/D conversion (if the A/D module is enabled).

**Note:** The special event trigger from the CCP1 and CCP2 modules will not set interrupt flag bit TMR1IF (PIR1<0>).

## 8.3 PWM Mode (PWM)

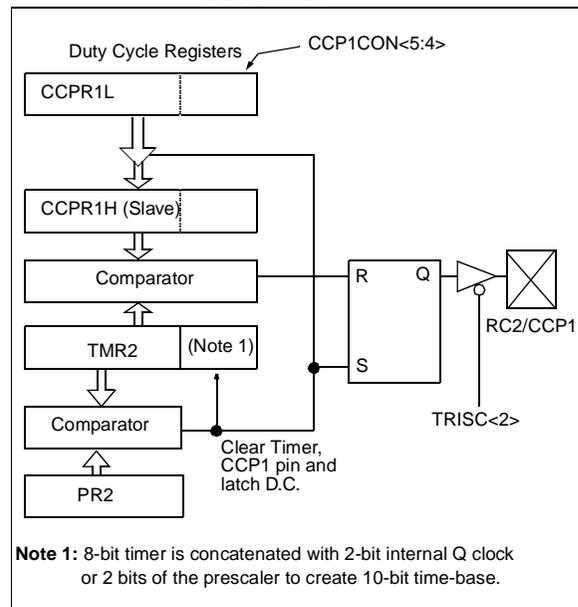
In pulse width modulation mode, the CCPx pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output.

**Note:** Clearing the CCP1CON register will force the CCP1 PWM output latch to the default low level. This is not the PORTC I/O data latch.

Figure 8-3 shows a simplified block diagram of the CCP module in PWM mode.

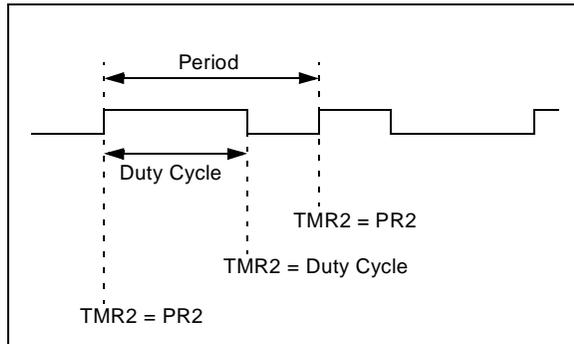
For a step-by-step procedure on how to set up the CCP module for PWM operation, see Section 8.3.3.

**FIGURE 8-3: SIMPLIFIED PWM BLOCK DIAGRAM**



A PWM output (Figure 8-4) has a time-base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

**FIGURE 8-4: PWM OUTPUT**



### 8.3.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

$$\text{PWM period} = [(PR2) + 1] \cdot 4 \cdot T_{osc} \cdot (\text{TMR2 prescale value})$$

PWM frequency is defined as  $1 / [\text{PWM period}]$ .

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCP1 pin is set (exception: if PWM duty cycle = 0%, the CCP1 pin will not be set)
- The PWM duty cycle is latched from CCPR1L into CCPR1H

**Note:** The Timer2 postscaler (see Section 8.1) is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

### 8.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available. The CCPR1L contains the eight MSBs and the CCP1CON<5:4> contains the two LSBs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. The following equation is used to calculate the PWM duty cycle in time:

$$\text{PWM duty cycle} = (\text{CCPR1L:CCP1CON<5:4>}) \cdot T_{osc} \cdot (\text{TMR2 prescale value})$$

CCPR1L and CCP1CON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPR1H until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read-only register.

The CCPR1H register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

When the CCPR1H and 2-bit latch match TMR2 concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP1 pin is cleared.

Maximum PWM resolution (bits) for a given PWM frequency:

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

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

### 8.3.3 SET-UP FOR PWM OPERATION

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

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

# PIC16F87X

**TABLE 8-3: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, AND TIMER1**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
0Bh,8Bh,10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBFIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- --0	---- --0
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN $\bar{C}$	TMR1CS	TMR1ON	--00 0000	--uu uuuu
15h	CCPR1L	Capture/Compare/PWM register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh	CCPR2L	Capture/Compare/PWM register2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM register2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by Capture and Timer1.

**Note 1:** The PSP is not implemented on the PIC16F873/876; always maintain these bits clear.

**TABLE 8-4: REGISTERS ASSOCIATED WITH PWM AND TIMER2**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
0Bh,8Bh,10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBFIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- --0	---- --0
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
11h	TMR2	Timer2 module's register								0000 0000	0000 0000
92h	PR2	Timer2 module's period register								1111 1111	1111 1111
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
15h	CCPR1L	Capture/Compare/PWM register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh	CCPR2L	Capture/Compare/PWM register2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM register2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by PWM and Timer2.

**Note 1:** Bits PSPIE and PSPIF are reserved on the PIC16F873/876; always maintain these bits clear.

## 9.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

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

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I<sup>2</sup>C)

Figure 9-1 shows a block diagram for the SPI mode, while Figure 9-5 and Figure 9-9 show the block diagrams for the two different I<sup>2</sup>C modes of operation.

# PIC16F87X

## REGISTER 9-1: SSPSTAT: SYNC SERIAL PORT STATUS REGISTER (ADDRESS: 94h)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0	
SMP	CKE	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	
bit7								bit0
<p>bit 7: <b>SMP:</b> Sample bit  <u>SPI Master Mode</u>            1 = Input data sampled at end of data output time            0 = Input data sampled at middle of data output time  <u>SPI Slave Mode</u>            SMP must be cleared when SPI is used in slave mode  <u>In I<sup>2</sup>C master or slave mode:</u>            1 = Slew rate control disabled for standard speed mode (100 kHz and 1 MHz)            0 = Slew rate control enabled for high speed mode (400 kHz)</p>								
<p>bit 6: <b>CKE:</b> SPI Clock Edge Select (Figure 9-4, Figure 9-5 and Figure 9-6)  <u>SPI Mode:</u>            CKP = 0            1 = Transmit happens on transition from active clock state to idle clock state            0 = Transmit happens on transition from idle clock state to active clock state            CKP = 1            1 = Data transmitted on falling edge of SCK            0 = Data transmitted on rising edge of SCK  <u>In I<sup>2</sup>C Master or Slave Mode:</u>            1 = Input levels conform to SMBUS spec            0 = Input levels conform to I<sup>2</sup>C specs</p>								
<p>bit 5: <b>D/<math>\bar{A}</math>:</b> Data/Address bit (I<sup>2</sup>C mode only)            1 = Indicates that the last byte received or transmitted was data            0 = Indicates that the last byte received or transmitted was address</p>								
<p>bit 4: <b>P:</b> Stop bit            (I<sup>2</sup>C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared)            1 = Indicates that a stop bit has been detected last (this bit is '0' on RESET)            0 = Stop bit was not detected last</p>								
<p>bit 3: <b>S:</b> Start bit            (I<sup>2</sup>C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared)            1 = Indicates that a start bit has been detected last (this bit is '0' on RESET)            0 = Start bit was not detected last</p>								
<p>bit 2: <b>R/<math>\bar{W}</math>:</b> Read/Write bit information (I<sup>2</sup>C mode only)            This bit holds the R/W bit information following the last address match. This bit is only valid from the address match to the next start bit, stop bit or not <math>\bar{ACK}</math> bit.  <u>In I<sup>2</sup>C slave mode:</u>            1 = Read            0 = Write  <u>In I<sup>2</sup>C master mode:</u>            1 = Transmit is in progress            0 = Transmit is not in progress.            Or'ing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in IDLE mode.</p>								
<p>bit 1: <b>UA:</b> Update Address (10-bit I<sup>2</sup>C mode only)            1 = Indicates that the user needs to update the address in the SSPADD register            0 = Address does not need to be updated</p>								
<p>bit 0: <b>BF:</b> Buffer Full Status bit  <u>Receive (SPI and I<sup>2</sup>C modes)</u>            1 = Receive complete, SSPBUF is full            0 = Receive not complete, SSPBUF is empty  <u>Transmit (I<sup>2</sup>C mode only)</u>            1 = Data Transmit in progress (does not include the <math>\bar{ACK}</math> and stop bits), SSPBUF is full            0 = Data Transmit complete (does not include the <math>\bar{ACK}</math> and stop bits), SSPBUF is empty</p>								

R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 - n = Value at POR reset

## REGISTER 9-2: SSPCON: SYNC SERIAL PORT CONTROL REGISTER (ADDRESS 14h)

R/W-0								
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	
bit7								bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

bit 7: **WCOL**: Write Collision Detect bit  
Master Mode:  
1 = A write to SSPBUF was attempted while the I<sup>2</sup>C conditions were not valid  
0 = No collision  
Slave Mode:  
1 = SSPBUF register is written while still transmitting the previous word (must be cleared in software)  
0 = No collision

bit 6: **SSPOV**: Receive Overflow Indicator bit  
In SPI mode  
1 = A new byte is received while SSPBUF holds previous data. Data in SSPSR is lost on overflow. In slave mode the user must read the SSPBUF, even if only transmitting data, to avoid overflows. In master mode the overflow bit is not set since each operation is initiated by writing to the SSPBUF register. (Must be cleared in software).  
0 = No overflow  
In I<sup>2</sup>C mode  
1 = A byte is received while the SSPBUF is holding the previous byte. SSPOV is a "don't care" in transmit mode. (Must be cleared in software).  
0 = No overflow

bit 5: **SSPEN**: Synchronous Serial Port Enable bit  
In SPI mode, when enabled, these pins must be properly configured as input or output.  
1 = Enables serial port and configures SCK, SDO, SDI, and SS as the source of the serial port pins  
0 = Disables serial port and configures these pins as I/O port pins  
In I<sup>2</sup>C mode, when enabled, these pins must be properly configured as input or output.  
1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins  
0 = Disables serial port and configures these pins as I/O port pins

bit 4: **CKP**: Clock Polarity Select bit  
In SPI mode  
1 = Idle state for clock is a high level  
0 = Idle state for clock is a low level  
In I<sup>2</sup>C slave mode, SCK release control  
1 = Enable clock  
0 = Holds clock low (clock stretch) (Used to ensure data setup time)  
In I<sup>2</sup>C master mode  
Unused in this mode

bit 3-0: **SSPM3:SSPM0**: Synchronous Serial Port Mode Select bits  
0000 = SPI master mode, clock = Fosc/4  
0001 = SPI master mode, clock = Fosc/16  
0010 = SPI master mode, clock = Fosc/64  
0011 = SPI master mode, clock = TMR2 output/2  
0100 = SPI slave mode, clock = SCK pin.  $\overline{SS}$  pin control enabled.  
0101 = SPI slave mode, clock = SCK pin.  $\overline{SS}$  pin control disabled.  $\overline{SS}$  can be used as I/O pin  
0110 = I<sup>2</sup>C slave mode, 7-bit address  
0111 = I<sup>2</sup>C slave mode, 10-bit address  
1000 = I<sup>2</sup>C master mode, clock = Fosc / (4 \* (SSPADD+1) )  
1011 = I<sup>2</sup>C firmware controlled master mode (slave idle)  
1110 = I<sup>2</sup>C firmware controlled master mode, 7-bit address with start and stop bit interrupts enabled  
1111 = I<sup>2</sup>C firmware controlled master mode, 10-bit address with start and stop bit interrupts enabled.  
1001, 1010, 1100, 1101 = reserved

# 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	
bit7								bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, Read as '0'  
- n = Value at POR reset

bit 7: **GCEN**: General Call Enable bit (In I<sup>2</sup>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<sup>2</sup>C master mode only)  
In master transmit mode:  
1 = Acknowledge was not received from slave  
0 = Acknowledge was received from slave

bit 5: **ACKDT**: Acknowledge Data bit (In I<sup>2</sup>C master mode only)  
In master receive mode:  
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<sup>2</sup>C master mode only).  
In master receive mode:  
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<sup>2</sup>C master mode only).  
1 = Enables Receive mode for I<sup>2</sup>C  
0 = Receive idle

bit 2: **PEN**: Stop Condition Enable bit (In I<sup>2</sup>C master mode only).  
SCK release control  
1 = Initiate Stop condition on SDA and SCL pins. Automatically cleared by hardware.  
0 = Stop condition idle

bit 1: **RSEN**: Repeated Start Condition Enabled bit (In I<sup>2</sup>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 Enabled bit (In I<sup>2</sup>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<sup>2</sup>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).

## 9.1 SPI Mode

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

- Serial Data Out (SDO)
- Serial Data In (SDI)
- Serial Clock (SCK)

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

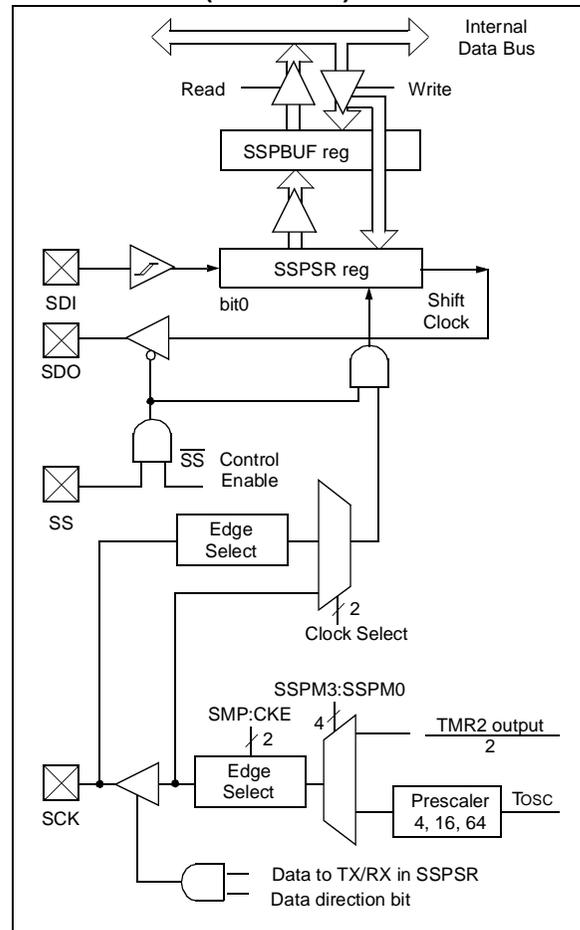
- Slave Select ( $\overline{SS}$ )

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

- Master Mode (SCK is the clock output)
- Slave Mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- 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)

Figure 9-4 shows the block diagram of the MSSP module when in SPI mode.

**FIGURE 9-1: MSSP BLOCK DIAGRAM (SPI MODE)**



To enable the serial port, MSSP Enable bit, SSPEN (SSPCON<5>) must be set. To reset or reconfigure SPI mode, clear bit SSPEN, re-initialize the SSPCON registers, and then set bit SSPEN. This configures the 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:

- SDO must have TRISC<5> cleared
- SCK (Master mode) must have TRISC<3> cleared
- SCK (Slave mode) must have TRISC<3> set
- $\overline{SS}$  must have TRISA<5> set

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

## 9.1.1 MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2, [Figure 9-5](#)) is to broadcast data by the software protocol.

In master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI module is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications as a “line activity monitor”.

The clock polarity is selected by appropriately programming bit CKP (SSPCON<4>). This then would give waveforms for SPI communication as shown in

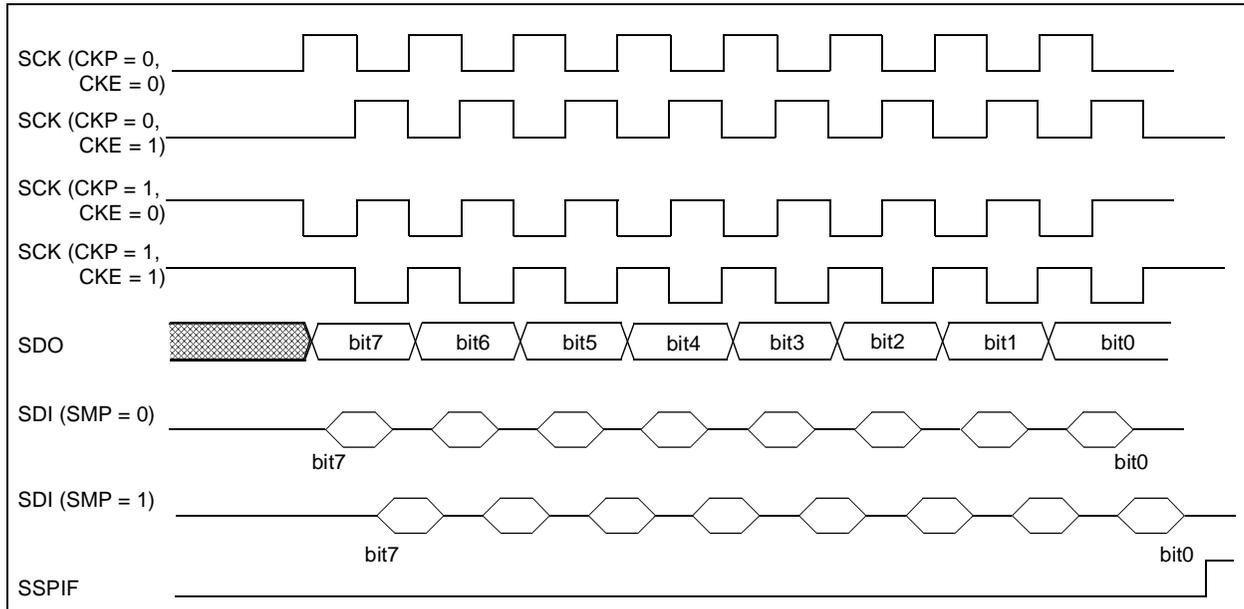
[Figure 9-6](#), [Figure 9-8](#) and [Figure 9-9](#) where the MSb is transmitted first. In master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- $F_{OSC}/4$  (or  $T_{CY}$ )
- $F_{OSC}/16$  (or  $4 \cdot T_{CY}$ )
- $F_{OSC}/64$  (or  $16 \cdot T_{CY}$ )
- Timer2 output/2

This allows a maximum bit clock frequency (at 20 MHz) of 5.0 MHz.

[Figure 9-6](#) shows the waveforms for Master mode. When  $CKE = 1$ , the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPBUF is loaded with the received data is shown.

**FIGURE 9-2: SPI MODE TIMING, MASTER MODE**



## 9.1.2 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 interrupt flag bit SSPIF (PIR1<3>) is set.

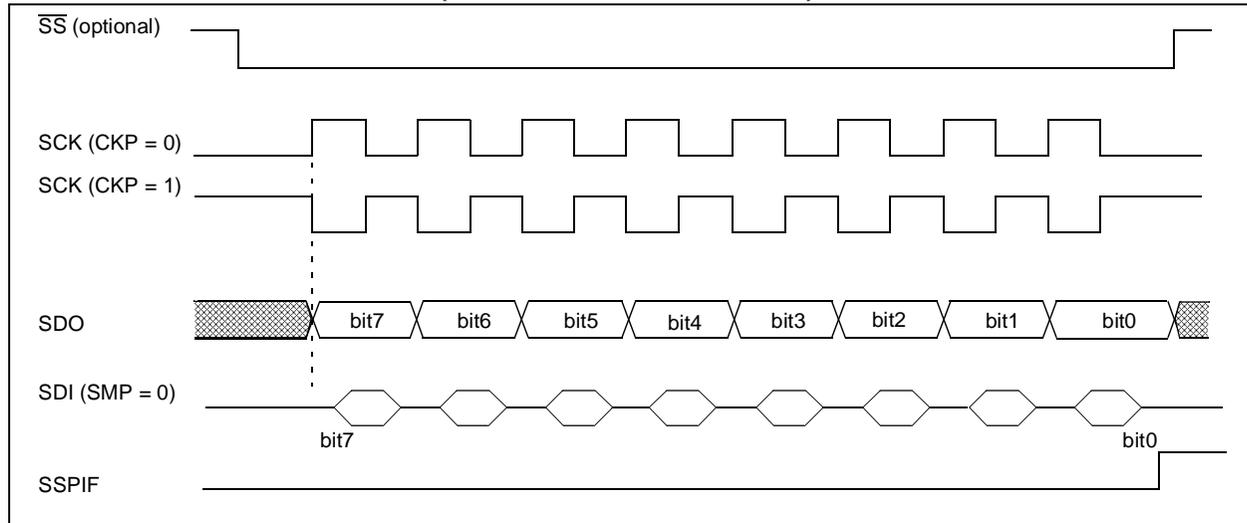
While in slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

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

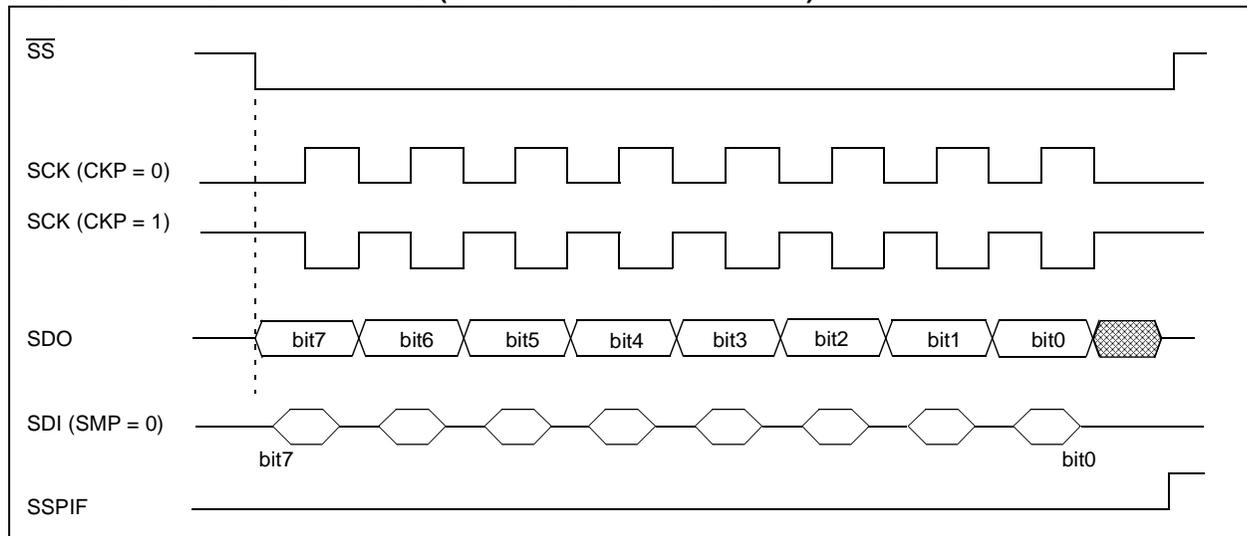
**Note:** When the SPI module is in Slave Mode with  $\overline{SS}$  pin control enabled, (SSPCON<3:0> = 0100) the SPI module will reset if the  $\overline{SS}$  pin is set to VDD.

**Note:** If the SPI is used in Slave Mode with CKE = '1', then  $\overline{SS}$  pin control must be enabled.

**FIGURE 9-3: SPI MODE TIMING (SLAVE MODE WITH CKE = 0)**



**FIGURE 9-4: SPI MODE TIMING (SLAVE MODE WITH CKE = 1)**



# PIC16F87X

**TABLE 9-1 REGISTERS ASSOCIATED WITH SPI OPERATION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR, BOR	MCLR, WDT
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D $\bar{A}$	P	S	R $\bar{W}$	UA	BF	0000 0000	0000 0000

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

**Note 1:** These bits are reserved on the 28-pin devices; always maintain these bits clear.

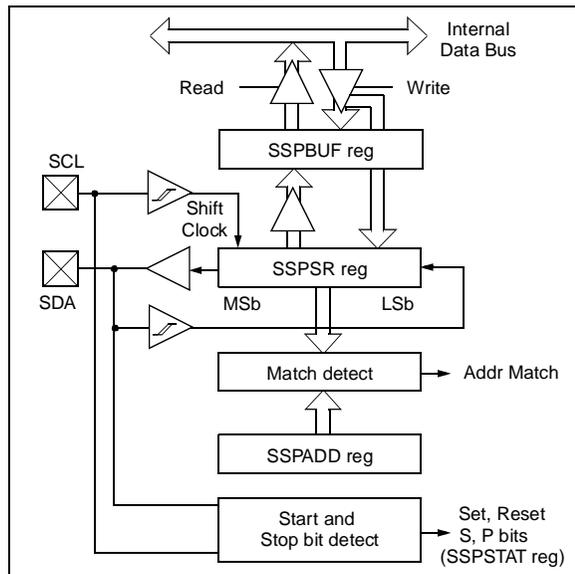
## 9.2 MSSP I<sup>2</sup>C Operation

The MSSP module in I<sup>2</sup>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<sup>2</sup>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 slow rate control of the pin that is independent of device frequency.

**FIGURE 9-5: I<sup>2</sup>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<sup>2</sup>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<sup>2</sup>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<sup>2</sup>C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I<sup>2</sup>C modes to be selected:

- I<sup>2</sup>C Slave mode (7-bit address)
- I<sup>2</sup>C Slave mode (10-bit address)
- I<sup>2</sup>C Master mode, clock = OSC/4 (SSPADD +1)

Before selecting any I<sup>2</sup>C mode, the SCL and SDA pins must be programmed to inputs by setting the appropriate TRIS bits. Selecting an I<sup>2</sup>C mode, by setting the SSPEN bit, enables the SCL and SDA pins to be used as the clock and data lines in I<sup>2</sup>C mode.

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<sup>2</sup>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).

## 9.2.1 SLAVE MODE

In slave mode, the SCL and SDA pins must be configured as inputs. The MSSP module will override the input state with the output data when required (slave-transmitter).

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

There are certain conditions that will cause the MSSP module not to give this ACK pulse. These are if either (or both):

- a) The buffer full bit BF (SSPSTAT<0>) was set before the transfer was received.
- b) The overflow bit SSPOV (SSPCON<6>) was set before the transfer was received.

If the BF bit is set, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF and SSPOV are set. Table 9-2 shows what happens when a data transfer byte is received, given the status of bits BF and SSPOV. The shaded cells show the condition where user software did not properly clear the overflow condition. Flag bit BF is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low time for proper operation. The high and low times of the I<sup>2</sup>C specification, as well as the requirement of the MSSP module, is shown in timing parameter #100 and parameter #101 of the electrical specifications.

### 9.2.1.1 ADDRESSING

Once the MSSP module has been enabled, it waits for a START condition to occur. Following the START condition, the 8-bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match, and the BF and SSPOV bits are clear, the following events occur:

- a) The SSPSR register value is loaded into the SSPBUF register on the falling edge of the 8th SCL pulse.
- b) The buffer full bit, BF, is set on the falling edge of the 8th SCL pulse.
- c) An  $\overline{\text{ACK}}$  pulse is generated.
- d) SSP interrupt flag bit, SSPIF (PIR1<3>), is set (interrupt is generated if enabled) on the falling edge of the 9th SCL pulse.

In 10-bit address mode, two address bytes need to be received by the slave. The five Most Significant bits (MSBs) of the first address byte specify if this is a 10-bit address. Bit R/W (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSBs of the address. The sequence of events for a 10-bit address is as follows, with steps 7-9 for slave-transmitter:

1. Receive first (high) byte of Address (bits SSPIF, BF and UA (SSPSTAT<1>) are set).
2. Update the SSPADD register with the second (low) byte of Address (clears bit UA and releases the SCL line).
3. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
4. Receive second (low) byte of Address (bits SSPIF, BF and UA are set).
5. Update the SSPADD register with the first (high) byte of Address. This will clear bit UA and release the SCL line.
6. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
7. Receive Repeated Start condition.
8. Receive first (high) byte of Address (bits SSPIF and BF are set).
9. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

**Note:** Following the Repeated Start condition (step 7) in 10-bit mode, the user only needs to match the first 7-bit address. The user does not update the SSPADD for the second half of the address.

## 9.2.1.2 SLAVE RECEPTION

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

When the address byte overflow condition exists, then no acknowledge (ACK) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set or bit SSPOV (SSPCON<6>) is set.

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

**Note:** The SSPBUF will be loaded if the SSPOV bit is set and the BF flag is cleared. If a read of the SSPBUF was performed, but the user did not clear the state of the SSPOV bit before the next receive occurred, the  $\overline{ACK}$  is not sent and the SSPBUF is updated.

**TABLE 9-2 DATA TRANSFER RECEIVED BYTE ACTIONS**

Status Bits as Data Transfer is Received		SSPSR → SSPBUF	Generate $\overline{ACK}$ Pulse	Set bit SSPIF (SSP Interrupt occurs if enabled)
BF	SSPOV			
0	0	Yes	Yes	Yes
1	0	No	No	Yes
1	1	No	No	Yes
0	1	Yes	No	Yes

**Note 1:** Shaded cells show the conditions where the user software did not properly clear the overflow condition.

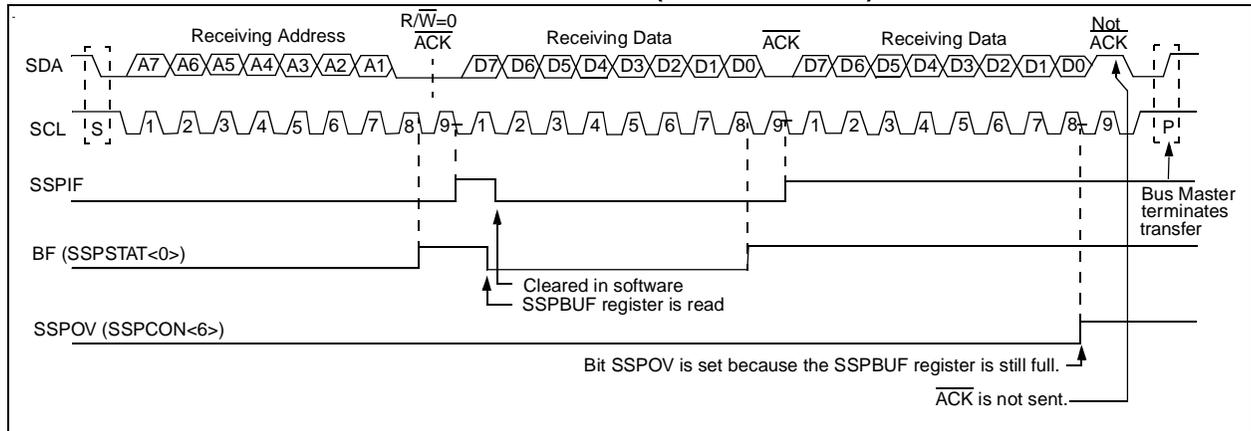
## 9.2.1.3 SLAVE TRANSMISSION

When the  $\overline{R/W}$  bit of the incoming address byte is set and an address match occurs, the  $\overline{R/W}$  bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The  $\overline{ACK}$  pulse will be sent on the ninth bit, and the SCL pin is held low. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then the SCL pin should be enabled by setting bit CKP (SSPCON<4>). The master must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master by stretching the clock. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 9-7).

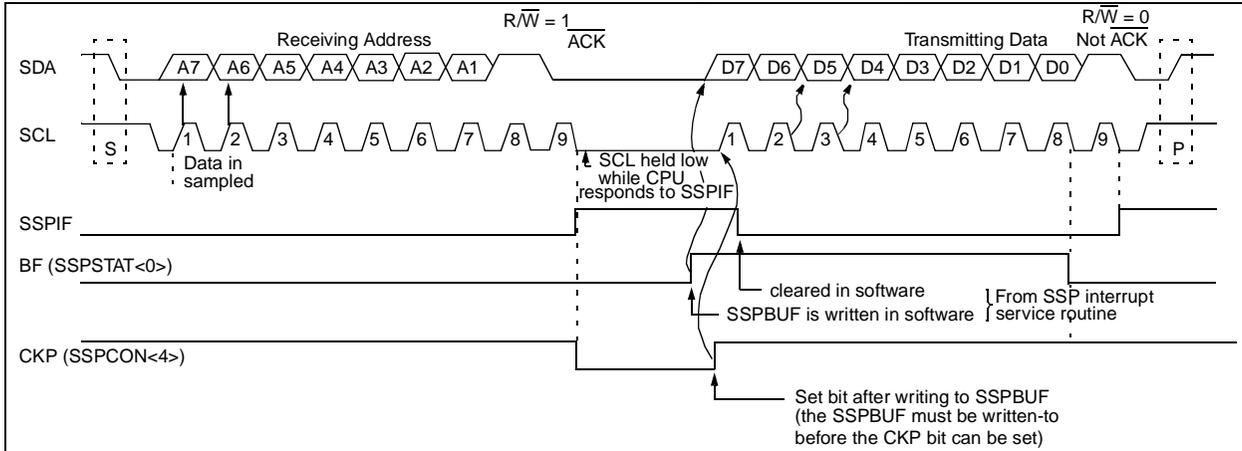
An SSP interrupt is generated for each data transfer byte. The SSPIF flag bit must be cleared in software and the SSPSTAT register is used to determine the status of the byte transfer. The SSPIF flag bit is set on the falling edge of the ninth clock pulse.

As a slave-transmitter, the  $\overline{ACK}$  pulse from the master receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line is high (not  $\overline{ACK}$ ), then the data transfer is complete. When the not  $\overline{ACK}$  is latched by the slave, the slave logic is reset and the slave then monitors for another occurrence of the START bit. If the SDA line was low ( $\overline{ACK}$ ), the transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then the SCL pin should be enabled by setting the CKP bit.

**FIGURE 9-6: I<sup>2</sup>C WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)**



**FIGURE 9-7: I<sup>2</sup>C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)**



### 9.2.2 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I<sup>2</sup>C 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 I<sup>2</sup>C protocol. It consists of all 0's with R/W = 0

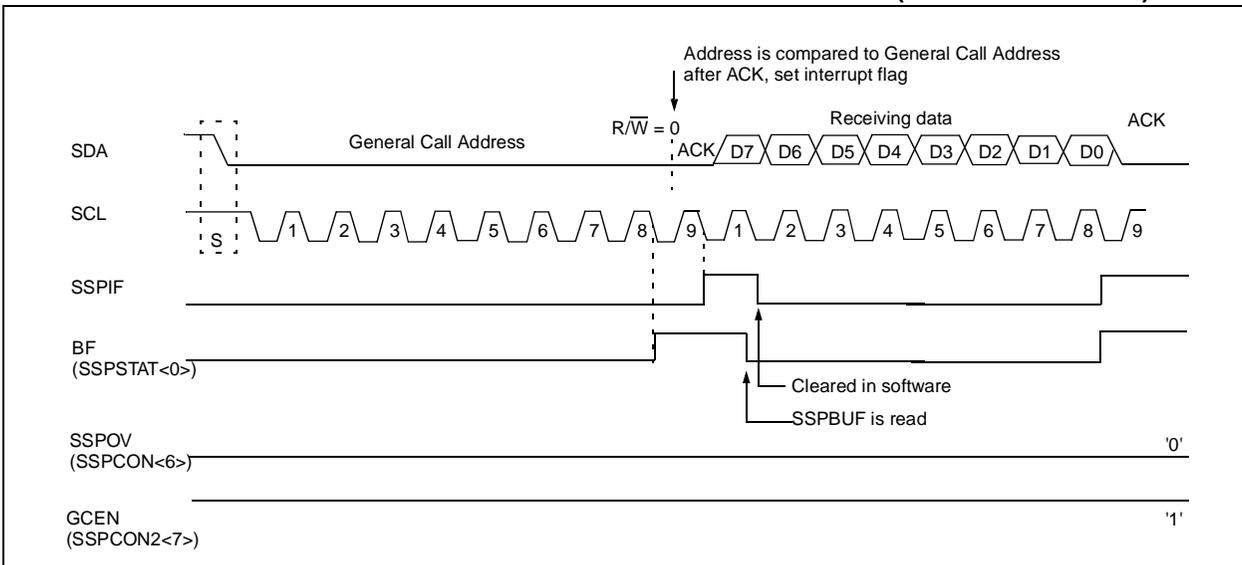
The general call address is recognized when the General Call Enable bit (GCEN) is enabled (SSPCON2<7> is set). Following a start-bit detect, 8-bits are shifted into SSPSR and the address is compared against SSPADD. It is also compared to the general call address and fixed in hardware.

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 (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)**



## 9.2.3 SLEEP OPERATION

While in sleep mode, the I<sup>2</sup>C module can receive addresses or data. When an address match or complete byte transfer occurs, wake the processor from sleep (if the SSP interrupt is enabled).

## 9.2.4 EFFECTS OF A RESET

A reset disables the SSP module and terminates the current transfer.

**TABLE 9-3 REGISTERS ASSOCIATED WITH I<sup>2</sup>C OPERATION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR, BOR	MCLR, WDT
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
0Dh	PIR2	—	(2)	—	EEIF	BCLIF	—	—	CCP2IF	-r-0 0--0	-r-0 0--0
8Dh	PIE2	—	(2)	—	EEIE	BCLIE	—	—	CCP2IE	-r-0 0--0	-r-0 0--0
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000
94h	SSPSTAT	SMP	CKE	D $\bar{A}$	P	S	R $\bar{W}$	UA	BF	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by the SSP in I<sup>2</sup>C mode.

**Note 1:** These bits are reserved on the 28-pin devices; always maintain these bits clear.

**2:** These bits are reserved on these devices; always maintain these bits clear.

# PIC16F87X

## 9.2.5 MASTER MODE

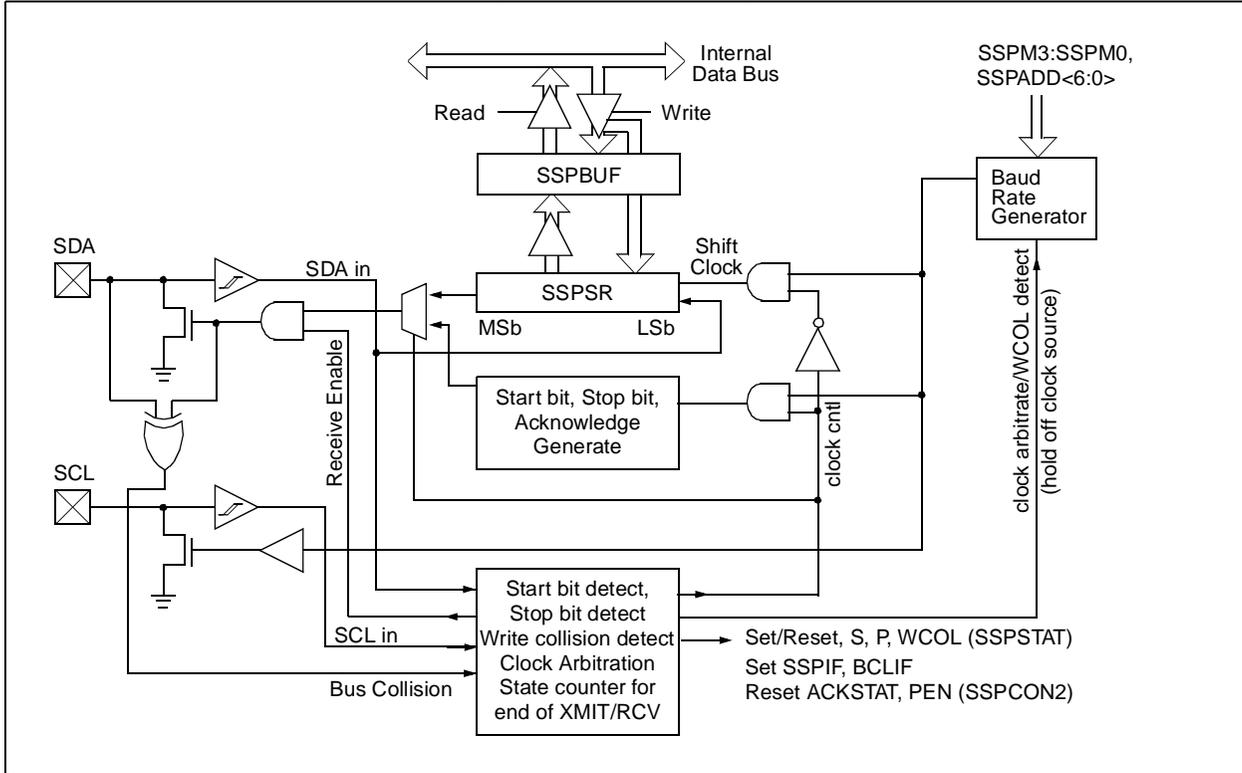
Master mode of operation is supported by interrupt generation on the detection of the START and STOP conditions. The STOP (P) and START (S) bits are cleared from a reset or when the MSSP module is disabled. Control of the I<sup>2</sup>C bus may be TAKEN when the P bit is set, or the bus is idle with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

In master mode, the SCL and SDA lines are manipulated by the MSSP hardware.

The following events will cause the 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 9-9: SSP BLOCK DIAGRAM (I<sup>2</sup>C MASTER MODE)**



## 9.2.6 MULTI-MASTER MODE

In multi-master mode, the interrupt generation on the detection of the START and STOP conditions allows the determination of when the bus is free. 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 bit P (SSPSTAT<4>) is set, or the bus is idle with both the S and P bits clear. When the bus is busy, enabling the SSP Interrupt will generate the interrupt when the STOP condition occurs.

In multi-master operation, the SDA line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed in hardware, with the result placed in the BCLIF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

## 9.2.7 I<sup>2</sup>C 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 I<sup>2</sup>C port to receive data.
- Generate an Acknowledge condition at the end of a received byte of data.

**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.

## 9.2.7.1 I<sup>2</sup>C 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 I<sup>2</sup>C 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/W) bit. In this case, the R/W 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/W bit. In this case, the R/W 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 I<sup>2</sup>C 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) The 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) The user loads the SSPBUF with address to transmit.
- d) Address is shifted out the SDA pin until all 8 bits are transmitted.
- e) The MSSP Module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
- f) The module generates an interrupt at the end of the ninth clock cycle by setting SSPIF.
- g) The user loads the SSPBUF with eight bits of data.
- h) DATA is shifted out the SDA pin until all 8 bits are transmitted.

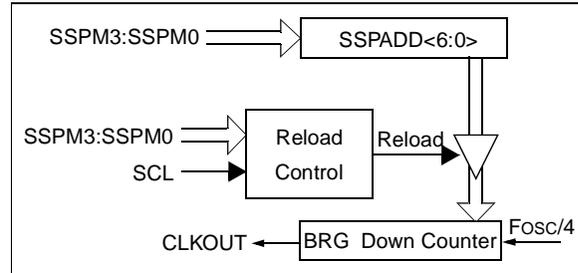
- i) The MSSP module shifts in the ACK bit from the slave device, and writes its value into the SSPCON2 register ( SSPCON2<6>).
- j) The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
- k) The user generates a STOP condition by setting the STOP enable bit PEN in SSPCON2.
- l) Interrupt is generated once the STOP condition is complete.

## 9.2.8 BAUD RATE GENERATOR

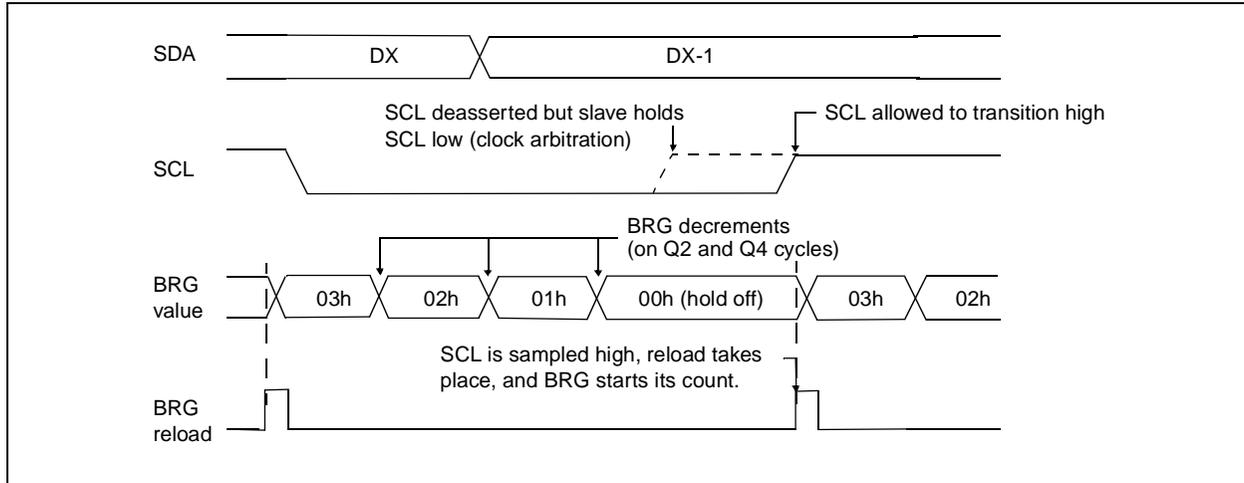
In I<sup>2</sup>C master mode, the reload value for the BRG is located in the lower 7 bits of the SSPADD register (Figure 9-10). When the BRG is loaded with this value, the BRG counts down to 0 and stops until another reload has taken place. The BRG count is decremented twice per instruction cycle (TCY), on the Q2 and Q4 clock.

In I<sup>2</sup>C master mode, the BRG is reloaded automatically. If Clock Arbitration is taking place for instance, the BRG will be reloaded when the SCL pin is sampled high (Figure 9-11).

**FIGURE 9-10: BAUD RATE GENERATOR BLOCK DIAGRAM**



**FIGURE 9-11: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION**



## 9.2.9 I<sup>2</sup>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 re-loaded 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.

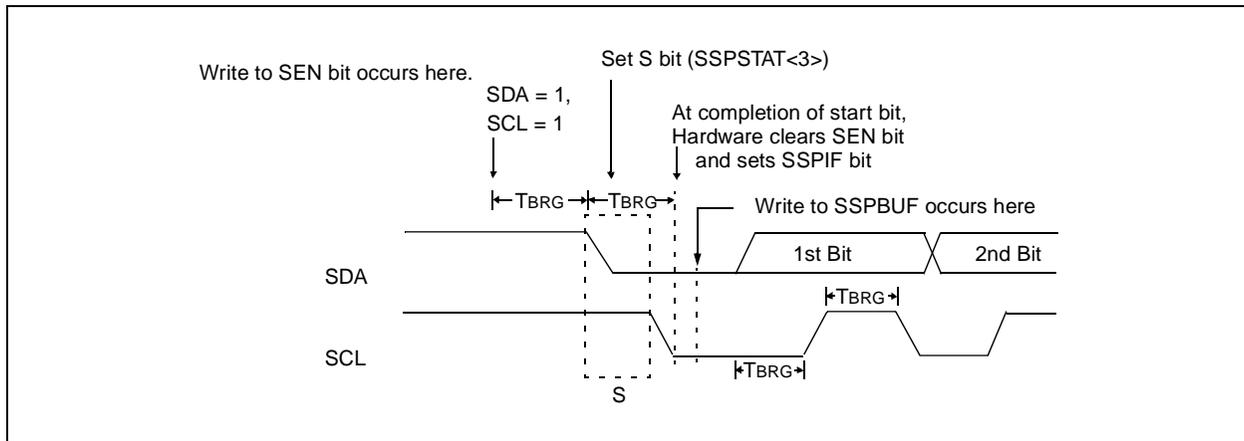
**Note:** If at the beginning of START condition the SDA and SCL pins are already sampled low, or if during the START condition the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag (BCLIF) is set, the START condition is aborted, and the I<sup>2</sup>C module is reset into its IDLE state.

### 9.2.9.1 WCOL STATUS FLAG

If the user writes the SSPBUF when an START sequence is in progress, then 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 9-12: FIRST START BIT TIMING**





### 9.2.11 I<sup>2</sup>C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or either half of a 10-bit address is accomplished by simply writing a value to SSPBUF register. This action will set the buffer full flag (BF) and allow the baud rate generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted (see data hold time spec). SCL is held low for one baud rate generator rollover count (TBRG). Data should be valid before SCL is released high (see data setup time spec). When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA allowing the slave device being addressed to respond with an  $\overline{\text{ACK}}$  bit during the ninth bit time, if an address match occurs or if data was received properly. The status of  $\overline{\text{ACK}}$  is read into the ACKDT on the falling edge of the ninth clock. If the master receives an acknowledge, the acknowledge status bit (ACKSTAT) is cleared. If not, the bit is set. After the ninth clock, the SSPIF is set and the master clock (baud rate generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 9-14).

After the write to the SSPBUF, each bit of address will be shifted out on the falling edge of SCL until all seven address bits and the  $\overline{\text{R}/\overline{\text{W}}}$  bit are completed. On the falling edge of the eighth clock, the master will de-assert the SDA pin allowing the slave to respond with an acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the ACK bit is loaded into the ACKSTAT status bit (SSPCON2<6>). Following the falling edge of the ninth clock transmission of the address, the SSPIF is set, the BF flag is cleared, and the baud rate generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

#### 9.2.11.1 BF STATUS FLAG

In transmit mode, the BF bit (SSPSTAT<0>) is set when the CPU writes to SSPBUF and is cleared when all 8 bits are shifted out.

#### 9.2.11.2 WCOL STATUS FLAG

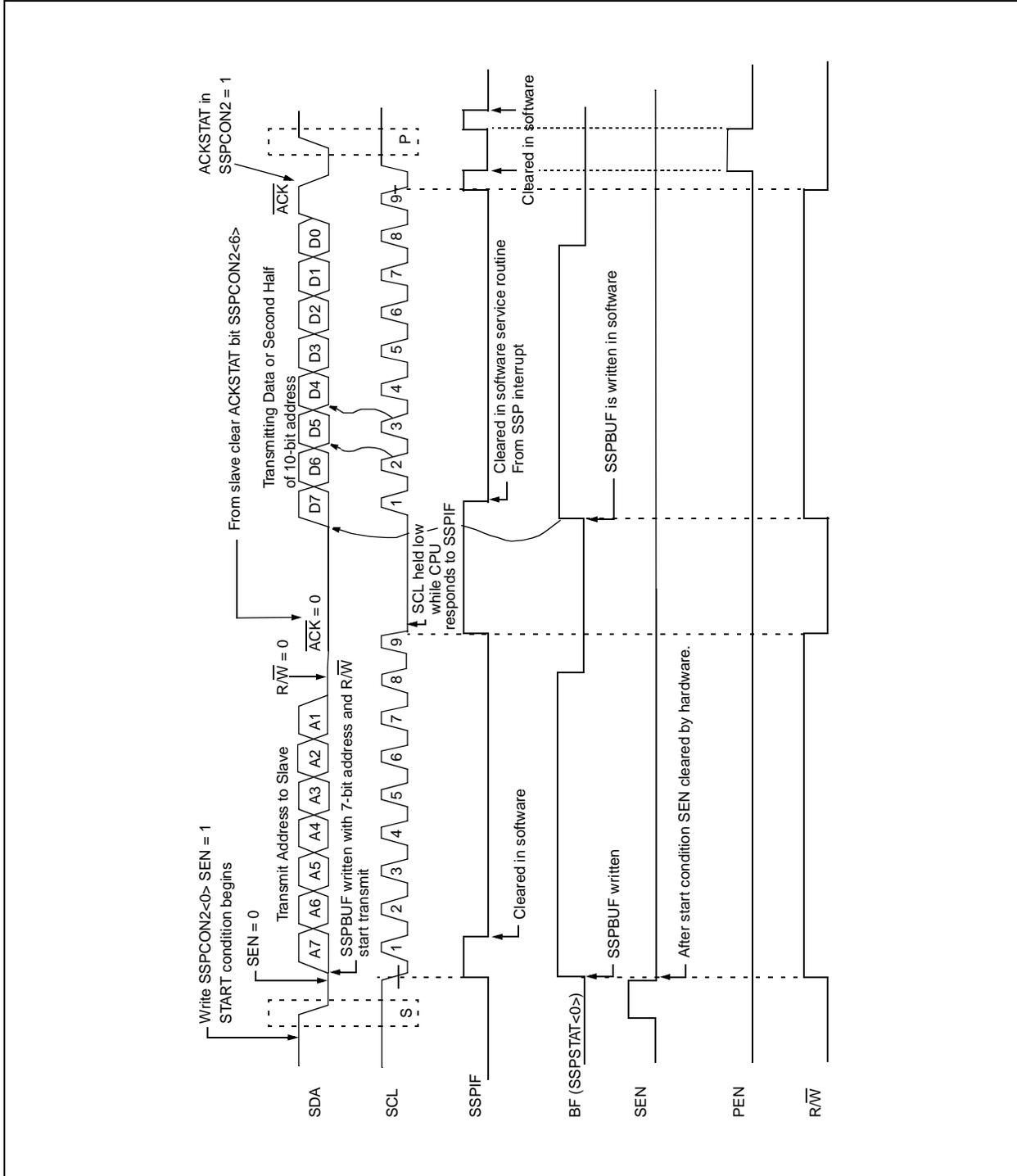
If the user writes the SSPBUF when a transmit is already in progress (i.e. SSPSR is still shifting out a data byte), then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

WCOL must be cleared in software.

#### 9.2.11.3 ACKSTAT STATUS FLAG

In transmit mode, the ACKSTAT bit (SSPCON2<6>) is cleared when the slave has sent an acknowledge ( $\overline{\text{ACK}} = 0$ ), and is set when the slave does not acknowledge ( $\overline{\text{ACK}} = 1$ ). A slave sends an acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

FIGURE 9-14: I<sup>2</sup>C MASTER MODE TIMING (TRANSMISSION, 7 OR 10-BIT ADDRESS)



---

### 9.2.12 I<sup>2</sup>C MASTER MODE RECEPTION

Master mode reception is enabled by programming the receive enable bit, RCEN (SSPCON2<3>).

<p><b>Note:</b> The SSP module must be in an IDLE STATE before the RCEN bit is set or the RCEN bit will be disregarded.</p>
-----------------------------------------------------------------------------------------------------------------------------

The baud rate generator begins counting, and on each rollover, the state of the SCL pin changes (high to low/low to high), and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF flag is set, the SSPIF is set, and the baud rate generator is suspended from counting, holding SCL low. The SSP is now in IDLE state, awaiting the next command. When the buffer is read by the CPU, the BF flag is automatically cleared. The user can then send an acknowledge bit at the end of reception, by setting the acknowledge sequence enable bit, ACKEN (SSPCON2<4>).

#### 9.2.12.1 BF STATUS FLAG

In receive operation, BF is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when SSPBUF is read.

#### 9.2.12.2 SSPOV STATUS FLAG

In receive operation, SSPOV is set when 8 bits are received into the SSPSR, and the BF flag is already set from a previous reception.

#### 9.2.12.3 WCOL STATUS FLAG

If the user writes the SSPBUF when a receive is already in progress (i.e. SSPSR is still shifting in a data byte), then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).



## 9.2.13 ACKNOWLEDGE SEQUENCE TIMING

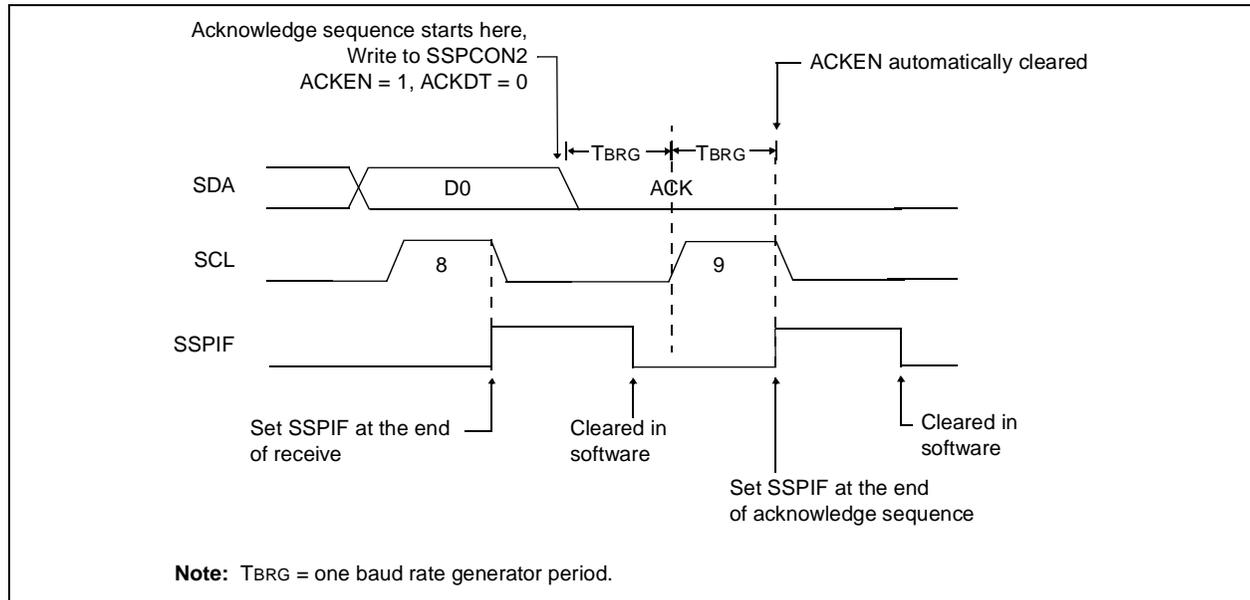
An acknowledge sequence is enabled by setting the acknowledge sequence enable bit, ACKEN (SSPCON2<4>). When this bit is set, the SCL pin is pulled low and the contents of the acknowledge data bit is presented on the SDA pin. If the user wishes to generate an acknowledge, the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an acknowledge sequence. The baud rate generator then counts for one rollover period ( $T_{BRG}$ ), and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration),

the baud rate generator counts for  $T_{BRG}$ . The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the baud rate generator is turned off, and the SSP module then goes into IDLE mode. (Figure 9-16)

### 9.2.13.1 WCOL STATUS FLAG

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

**FIGURE 9-16: ACKNOWLEDGE SEQUENCE WAVEFORM**



# PIC16F87X

## 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 deasserted. When the SDA pin is sampled high

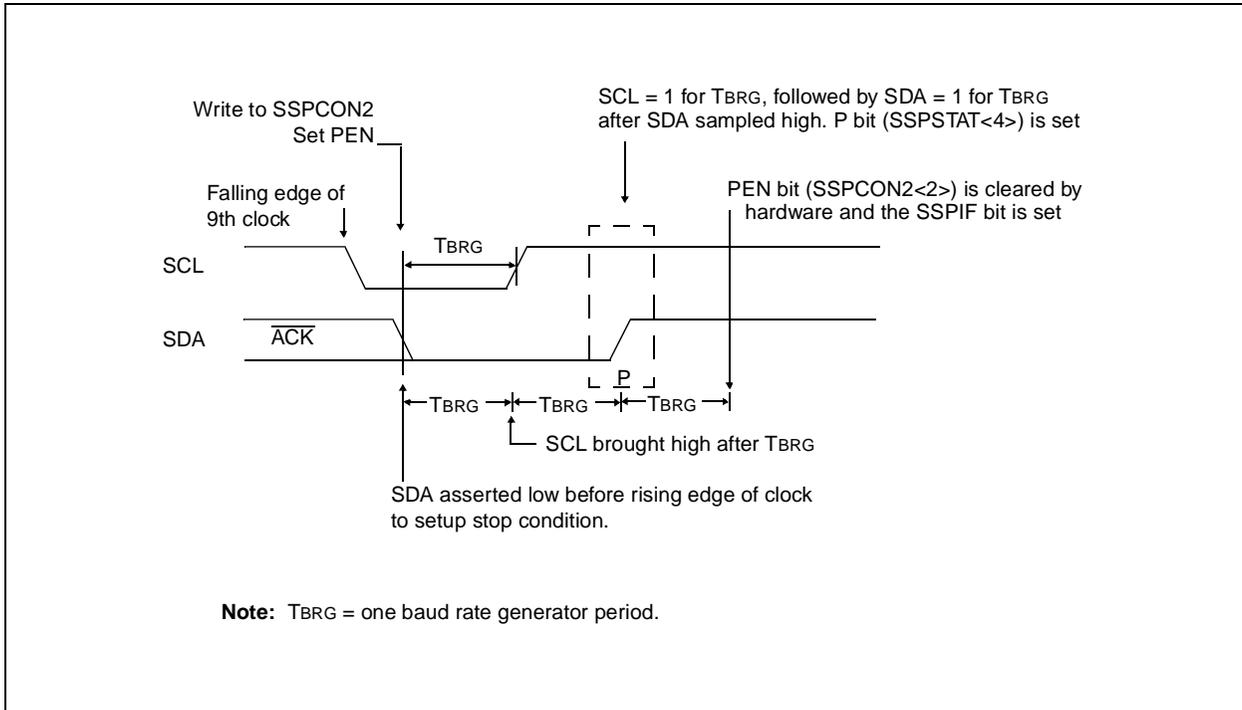
while SCL is high, the P bit (SSPSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 9-17).

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**



## 9.2.15 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any receive, transmit, or repeated start/stop condition, deasserts the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the baud rate generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the baud rate generator is reloaded with the contents of SSPADD<6:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 9-18).

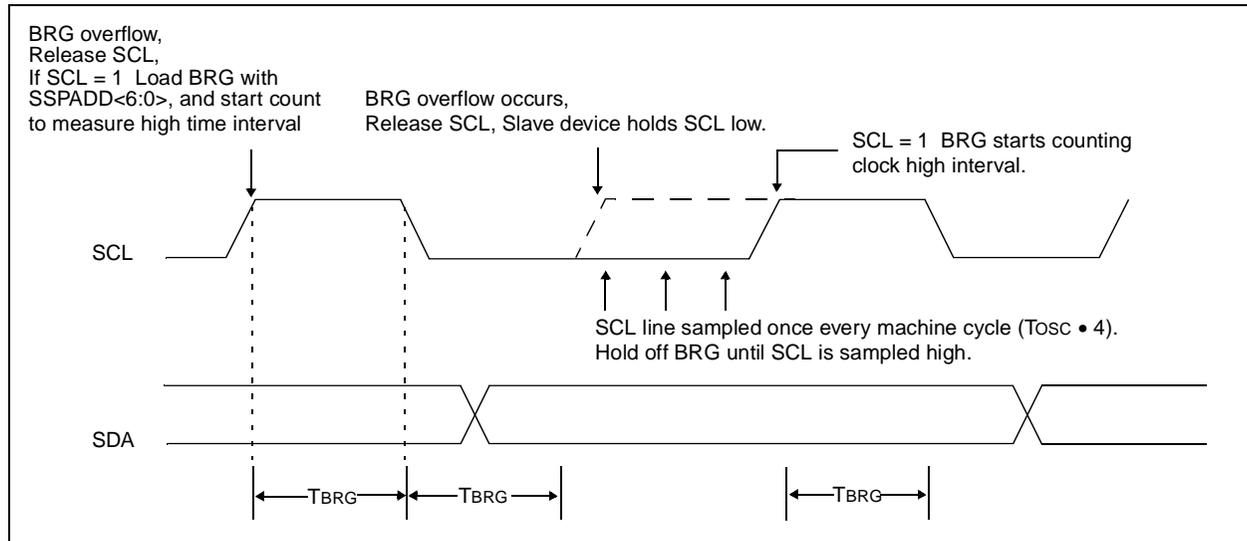
## 9.2.16 SLEEP OPERATION

While in sleep mode, the I<sup>2</sup>C module can receive addresses or data, and when an address match or complete byte transfer occurs, wake the processor from sleep (if the SSP interrupt is enabled).

## 9.2.17 EFFECTS OF A RESET

A reset disables the SSP module and terminates the current transfer.

**FIGURE 9-18: CLOCK ARBITRATION TIMING IN MASTER TRANSMIT MODE**



# PIC16F87X

## 9.2.18 MULTI-MASTER COMMUNICATION, BUS COLLISION, AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin = '0', a bus collision has TAKEN place. The master will set the Bus Collision Interrupt Flag, BCLIF and reset the I<sup>2</sup>C port to its IDLE state. (Figure 9-19).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted, and the SSPBUF can be written to. When the user services the bus collision interrupt service routine, and if the I<sup>2</sup>C bus is free, the user can resume communication by asserting a START condition.

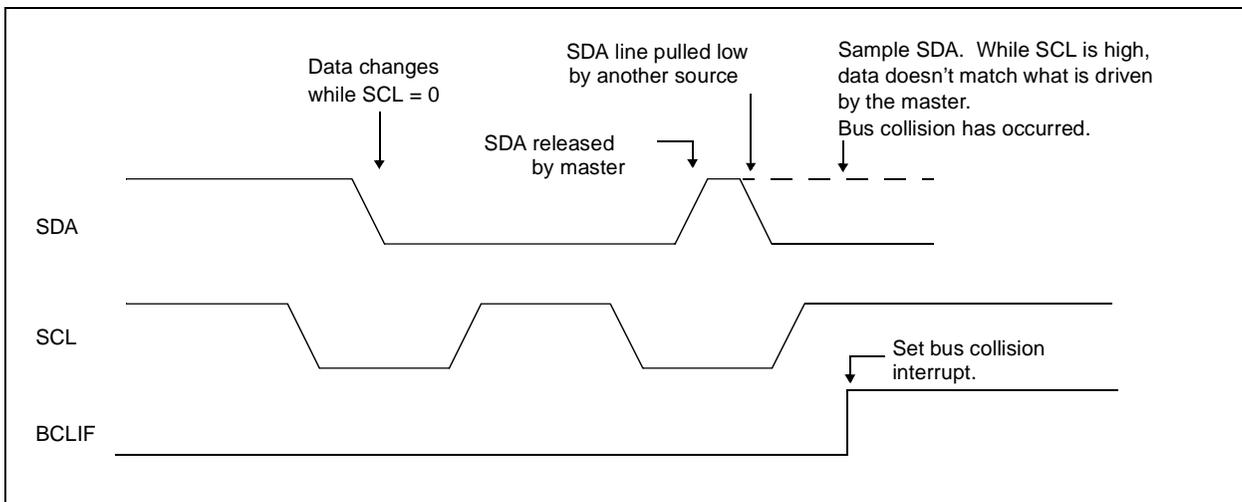
If a START, Repeated Start, STOP or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted, and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision interrupt service routine, and if the I<sup>2</sup>C bus is free, the user can resume communication by asserting a START condition.

The Master will continue to monitor the SDA and SCL pins, and if a STOP condition occurs, the SSPIF bit will be set.

A write to the SSPBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In multi-master mode, the interrupt generation on the detection of start and stop conditions allows the determination of when the bus is free. Control of the I<sup>2</sup>C bus can be TAKEN when the P bit is set in the SSPSTAT register, or the bus is idle and the S and P bits are cleared.

**FIGURE 9-19: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE**



## 9.2.18.1 BUS COLLISION DURING A START CONDITION

During a START condition, a bus collision occurs if:

- SDA or SCL are sampled low at the beginning of the START condition (Figure 9-20).
- SCL is sampled low before SDA is asserted low. (Figure 9-21).

During a START condition both the SDA and the SCL pins are monitored.

If:

the SDA pin is already low  
or the SCL pin is already low,

then:

the START condition is aborted,  
and the BCLIF flag is set,  
and the SSP module is reset to its IDLE state  
(Figure 9-20).

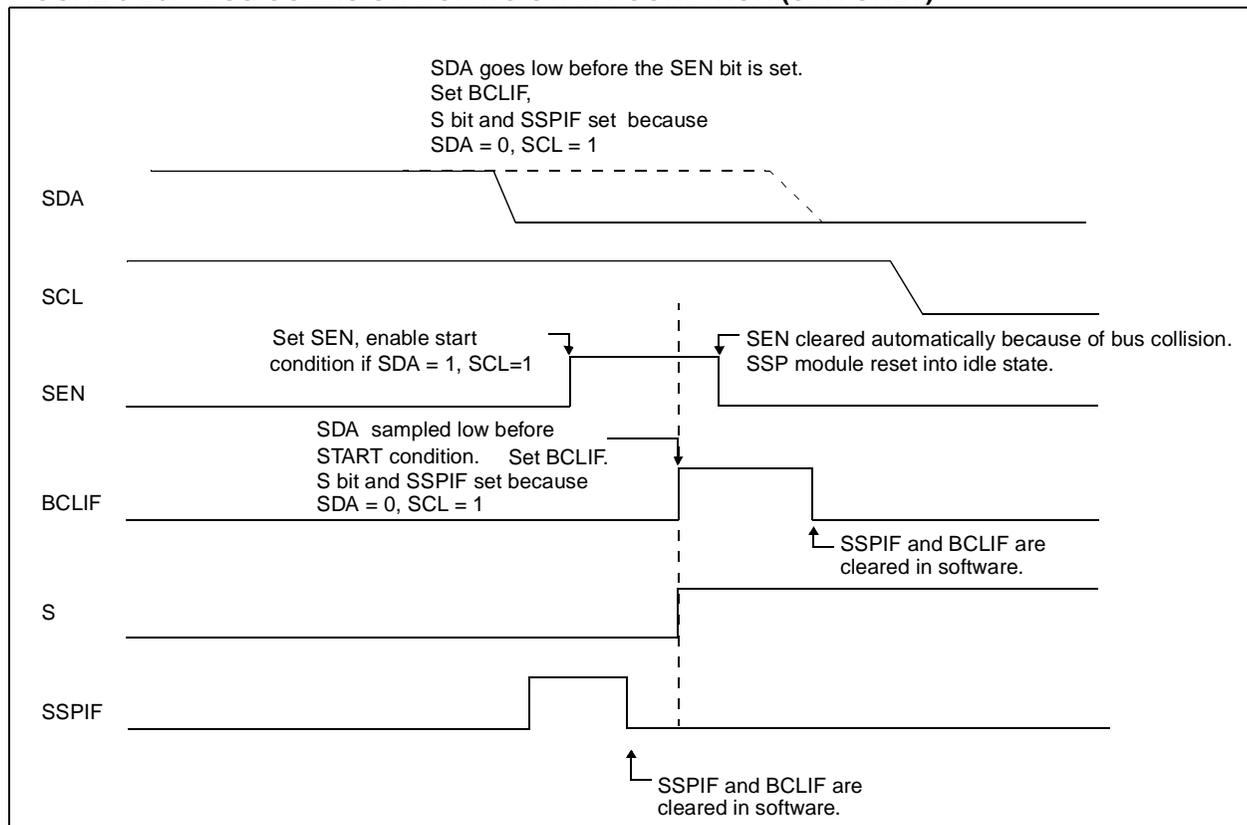
The START condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the baud rate generator is loaded from SSPADD<6:0> and counts down to 0. If the SCL pin is sampled low

while SDA is high, a bus collision occurs, because it is assumed that another master is attempting to drive a data '1' during the START condition.

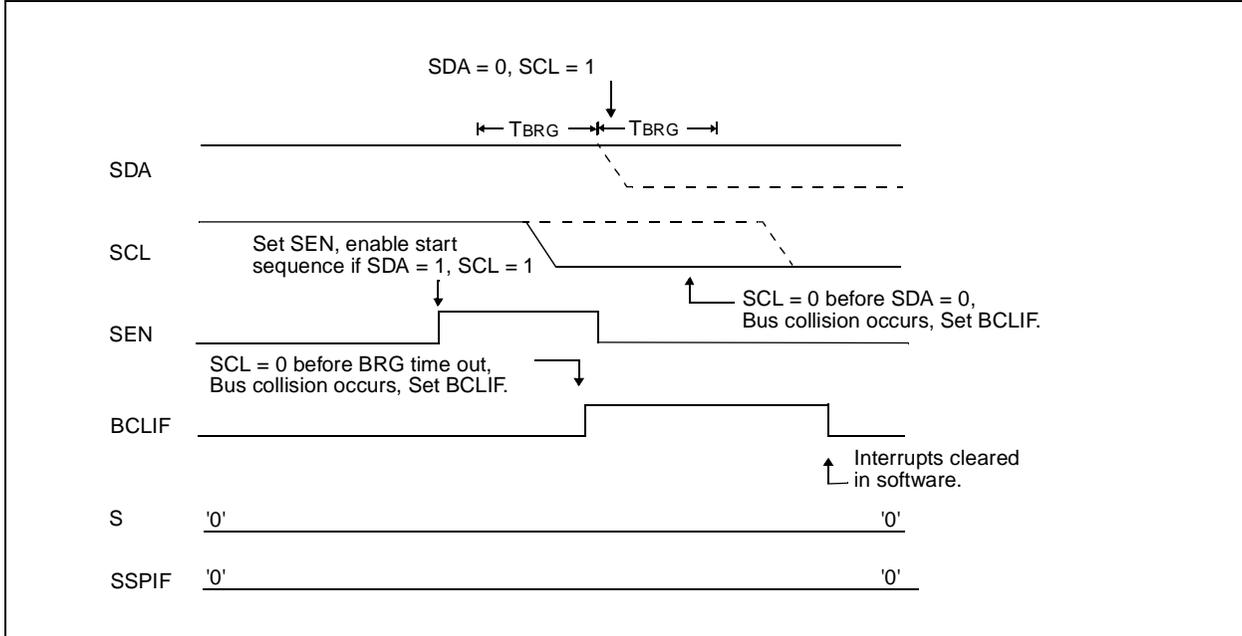
If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 9-22). If however a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The baud rate generator is then reloaded and counts down to 0. During this time, if the SCL pins are sampled as '0', a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

**Note:** The reason that bus collision is not a factor during a START condition is that no two bus masters can assert a START condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision, because the two masters must be allowed to arbitrate the first address following the START condition. If the address is the same, arbitration must be allowed to continue into the data portion, REPEATED START or STOP conditions.

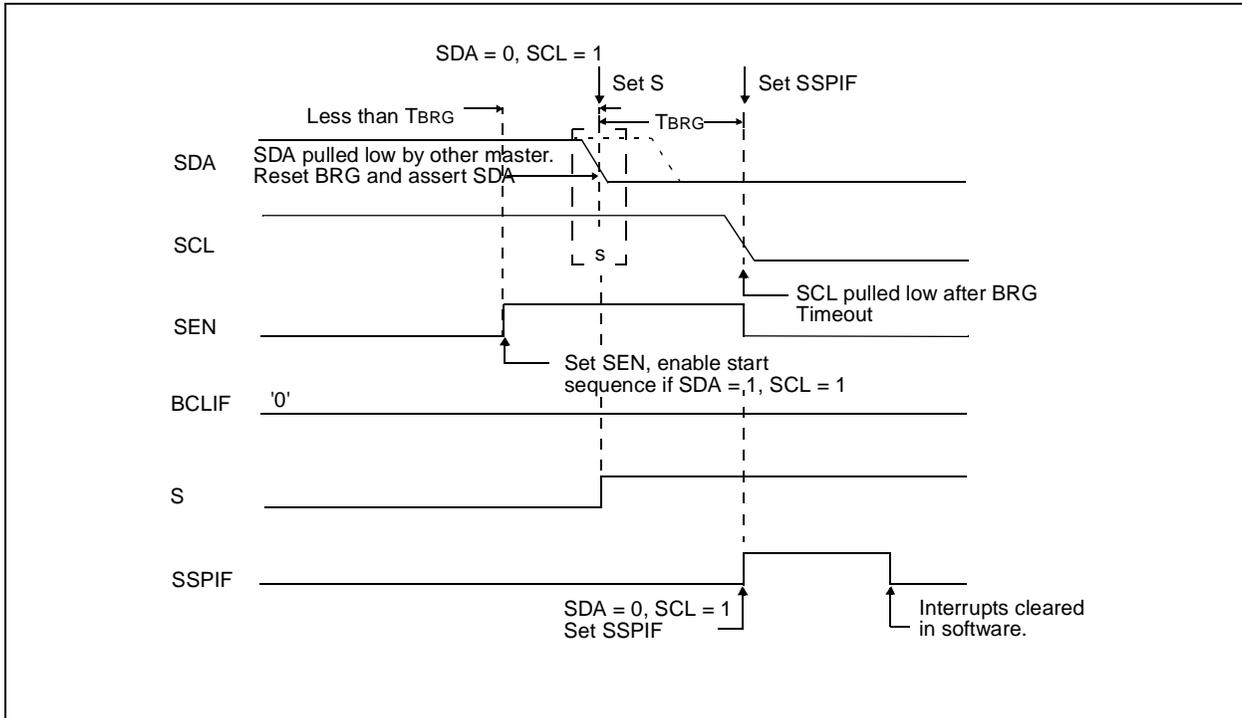
**FIGURE 9-20: BUS COLLISION DURING START CONDITION (SDA ONLY)**



**FIGURE 9-21: BUS COLLISION DURING START CONDITION (SCL = 0)**



**FIGURE 9-22: BRG RESET DUE TO SDA COLLISION DURING START CONDITION**



## 9.2.18.2 BUS COLLISION DURING A REPEATED START CONDITION

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

- A low level is sampled on SDA when SCL goes from low level to high level.
- SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1'.

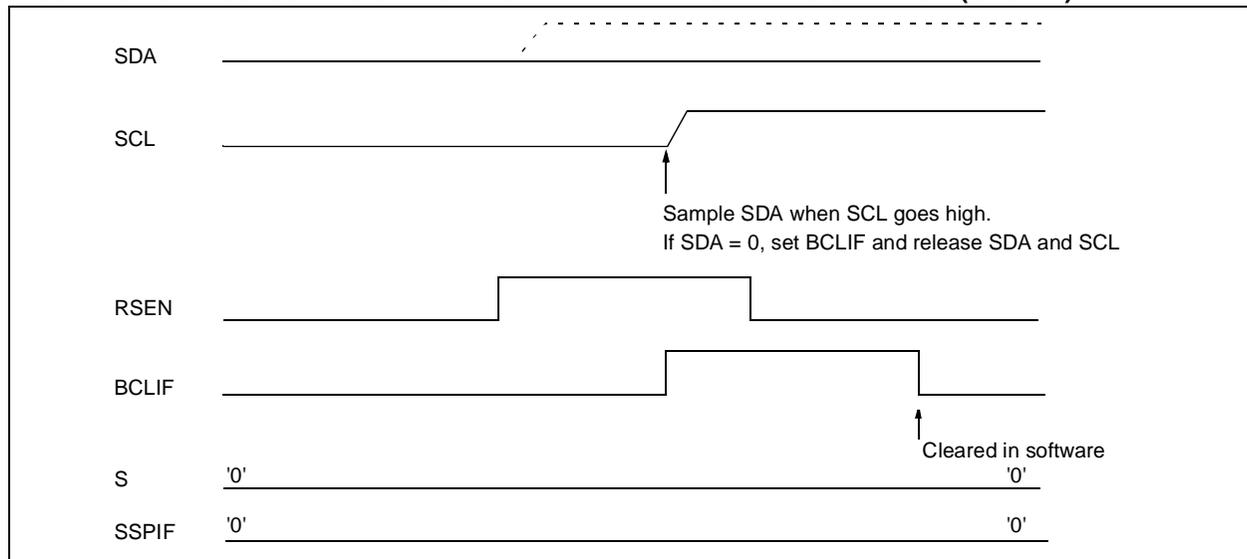
When the user deasserts SDA and the pin is allowed to float high, the BRG is loaded with SSPADD<6:0> and counts down to 0. The SCL pin is then deasserted, and when sampled high, the SDA pin is sampled. If SDA is low, a bus collision has occurred (i.e. another master is attempting to transmit a data '0'). 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.

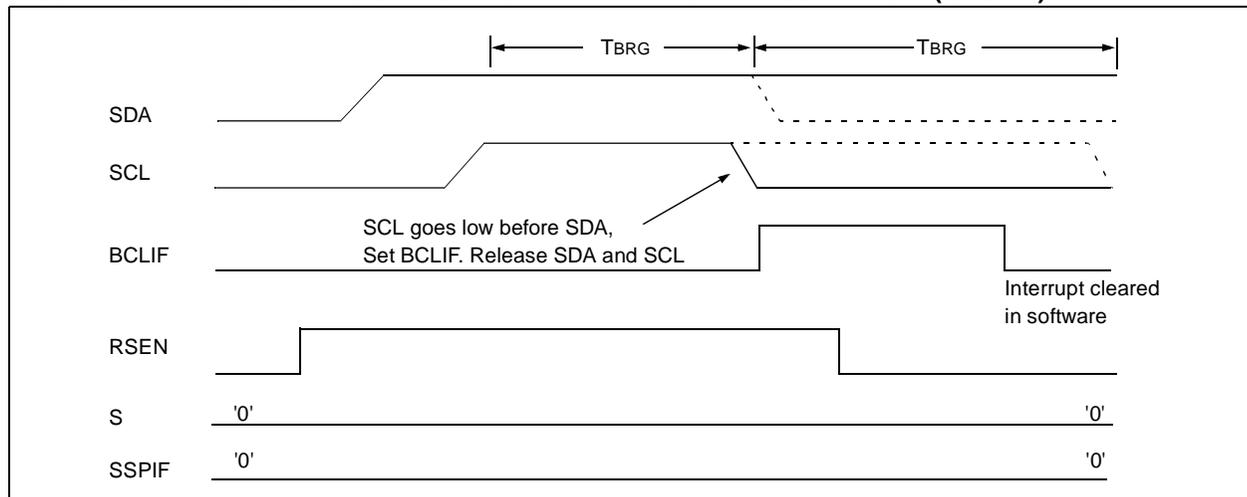
If, however, SCL goes from high to low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition.

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)**



# PIC16F87X

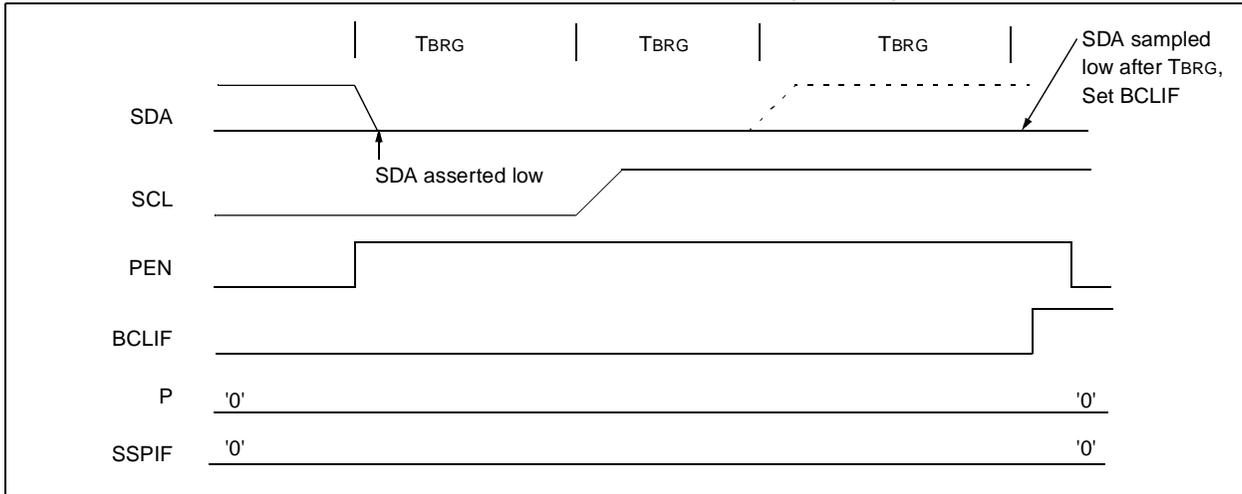
## 9.2.18.3 BUS COLLISION DURING A STOP CONDITION

Bus collision occurs during a STOP condition if:

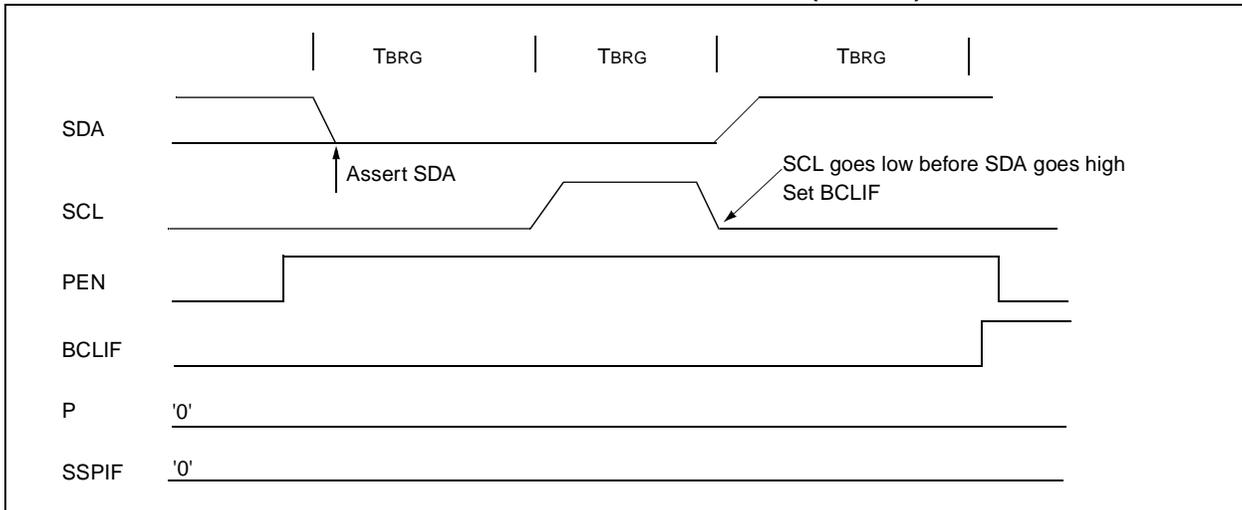
- After the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out.
- After the SCL pin is deasserted, SCL is sampled low before SDA goes high.

The STOP condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the baud rate generator is loaded with SSPADD<6:0> and counts down to 0. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0'. If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is a case of another master attempting to drive a data '0' (Figure 9-25).

**FIGURE 9-25: BUS COLLISION DURING A STOP CONDITION (CASE 1)**



**FIGURE 9-26: BUS COLLISION DURING A STOP CONDITION (CASE 2)**



## 9.3 Connection Considerations for I<sup>2</sup>C Bus

For standard-mode I<sup>2</sup>C bus devices, the values of resistors  $R_p$  and  $R_s$  in Figure 9-27 depend on the following parameters:

- Supply voltage
- Bus capacitance
- Number of connected devices (input current + leakage current).

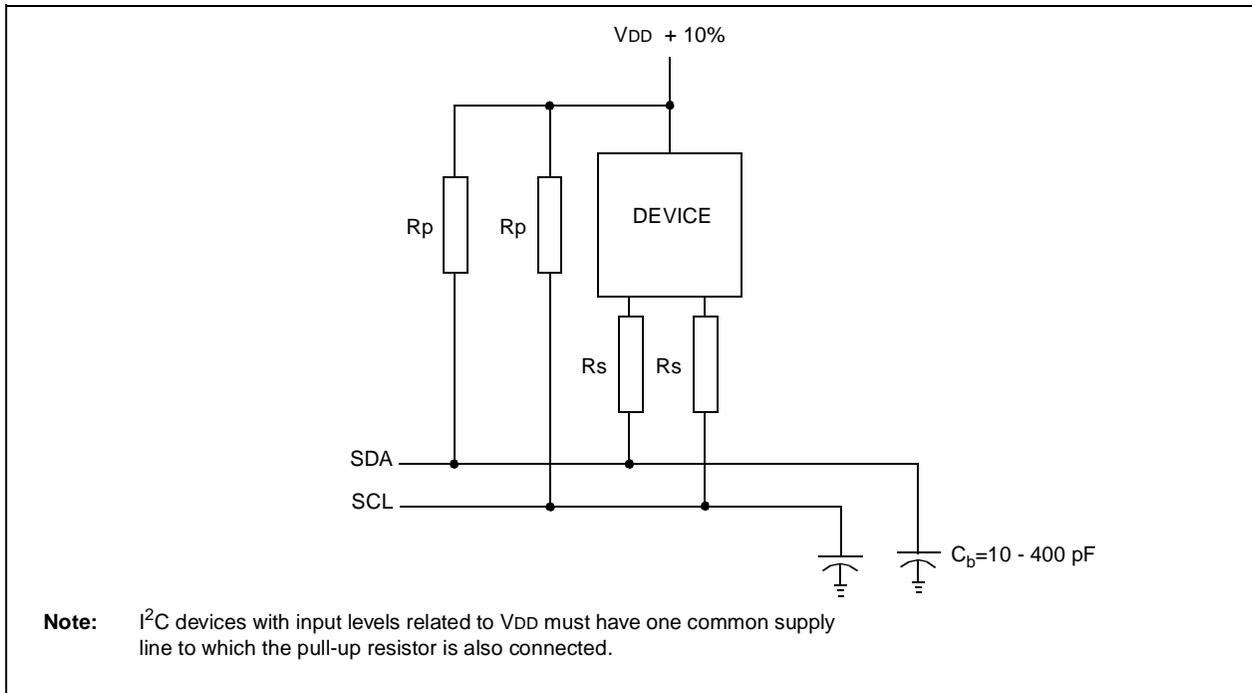
The supply voltage limits the minimum value of resistor  $R_p$  due to the specified minimum sink current of 3 mA at  $V_{OL\ max} = 0.4V$  for the specified output stages. For

example, with a supply voltage of  $V_{DD} = 5V \pm 10\%$  and  $V_{OL\ max} = 0.4V$  at 3 mA,  $R_{p\ min} = (5.5 - 0.4) / 0.003 = 1.7\ k\Omega$ .  $V_{DD}$  as a function of  $R_p$  is shown in Figure 9-27. The desired noise margin of  $0.1V_{DD}$  for the low level limits the maximum value of  $R_s$ . Series resistors are optional and used to improve ESD susceptibility.

The bus capacitance is the total capacitance of wire, connections, and pins. This capacitance limits the maximum value of  $R_p$  due to the specified rise time (Figure 9-27).

The SMP bit is the slew rate control enabled bit. This bit is in the SSPSTAT register, and controls the slew rate of the I/O pins when in I<sup>2</sup>C mode (master or slave).

**FIGURE 9-27: SAMPLE DEVICE CONFIGURATION FOR I<sup>2</sup>C BUS**



# PIC16F87X

---

NOTES:

## 10.0 ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

The Universal Synchronous Asynchronous Receiver Transmitter (USART) module is one of the two serial I/O modules. (USART is also known as a Serial Communications Interface or SCI). The USART can be configured as a full duplex asynchronous system that can communicate with peripheral devices such as CRT terminals and personal computers, or it can be configured as a half duplex synchronous system that can communicate with peripheral devices such as A/D or D/A integrated circuits, serial EEPROMs etc.

The USART can be configured in the following modes:

- Asynchronous (full duplex)
- Synchronous - Master (half duplex)
- Synchronous - Slave (half duplex)

Bit SPEN (RCSTA<7>) and bits TRISC<7:6> have to be set in order to configure pins RC6/TX/CK and RC7/RX/DT as the Universal Synchronous Asynchronous Receiver Transmitter.

The USART module also has a multi-processor communication capability using 9-bit address detection.

### REGISTER 10-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0	
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	
bit7								bit0

R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 - n = Value at POR reset

bit 7: **CSRC:** Clock Source Select bit  
 Asynchronous mode  
 Don't care  
 Synchronous mode  
 1 = Master mode (Clock generated internally from BRG)  
 0 = Slave mode (Clock from external source)

bit 6: **TX9:** 9-bit Transmit Enable bit  
 1 = Selects 9-bit transmission  
 0 = Selects 8-bit transmission

bit 5: **TXEN:** Transmit Enable bit  
 1 = Transmit enabled  
 0 = Transmit disabled  
**Note:** SREN/CREN overrides TXEN in SYNC mode.

bit 4: **SYNC:** USART Mode Select bit  
 1 = Synchronous mode  
 0 = Asynchronous mode

bit 3: **Unimplemented:** Read as '0'

bit 2: **BRGH:** High Baud Rate Select bit  
 Asynchronous mode  
 1 = High speed  
 0 = Low speed  
 Synchronous mode  
 Unused in this mode

bit 1: **TRMT:** Transmit Shift Register Status bit  
 1 = TSR empty  
 0 = TSR full

bit 0: **TX9D:** 9th bit of transmit data. Can be parity bit.

# PIC16F87X

## REGISTER 10-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit7							bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

bit 7: **SPEN**: Serial Port Enable bit  
1 = Serial port enabled (Configures RC7/RX/DT and RC6/TX/CK pins as serial port pins)  
0 = Serial port disabled

bit 6: **RX9**: 9-bit Receive Enable bit  
1 = Selects 9-bit reception  
0 = Selects 8-bit reception

bit 5: **SREN**: Single Receive Enable bit  
Asynchronous mode  
Don't care  
Synchronous mode - master  
1 = Enables single receive  
0 = Disables single receive  
This bit is cleared after reception is complete.  
Synchronous mode - slave  
Unused in this mode

bit 4: **CREN**: Continuous Receive Enable bit  
Asynchronous mode  
1 = Enables continuous receive  
0 = Disables continuous receive  
Synchronous mode  
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)  
0 = Disables continuous receive

bit 3: **ADDEN**: Address Detect Enable bit  
Asynchronous mode 9-bit (RX9 = 1)  
1 = Enables address detection, enable interrupt and load of the receive burffer when RSR<8> is set  
0 = Disables address detection, all bytes are received, and ninth bit can be used as parity bit

bit 2: **FERR**: Framing Error bit  
1 = Framing error (Can be updated by reading RCREG register and receive next valid byte)  
0 = No framing error

bit 1: **OERR**: Overrun Error bit  
1 = Overrun error (Can be cleared by clearing bit CREN)  
0 = No overrun error

bit 0: **RX9D**: 9th bit of received data (Can be parity bit)

## 10.1 USART Baud Rate Generator (BRG)

The BRG supports both the asynchronous and synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. In asynchronous mode, bit BRGH (TXSTA<2>) also controls the baud rate. In synchronous mode, bit BRGH is ignored. Table 10-1 shows the formula for computation of the baud rate for different USART modes which only apply in master mode (internal clock).

Given the desired baud rate and Fosc, the nearest integer value for the SPBRG register can be calculated using the formula in Table 10-1. From this, the error in baud rate can be determined.

It may be advantageous to use the high baud rate (BRGH = 1) even for slower baud clocks. This is because the  $F_{osc}/(16(X + 1))$  equation can reduce the baud rate error in some cases.

Writing a new value to the SPBRG register causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

### 10.1.1 SAMPLING

The data on the RC7/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin.

**TABLE 10-1: BAUD RATE FORMULA**

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{osc}/(64(X+1))$	Baud Rate = $F_{osc}/(16(X+1))$
1	(Synchronous) Baud Rate = $F_{osc}/(4(X+1))$	NA

X = value in SPBRG (0 to 255)

**TABLE 10-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used by the BRG.

# PIC16F87X

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

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	1.221	1.75	255	1.202	0.17	207	1.202	0.17	129
2.4	2.404	0.17	129	2.404	0.17	103	2.404	0.17	64
9.6	9.766	1.73	31	9.615	0.16	25	9.766	1.73	15
19.2	19.531	1.72	15	19.231	0.16	12	19.531	1.72	7
28.8	31.250	8.51	9	27.778	3.55	8	31.250	8.51	4
33.6	34.722	3.34	8	35.714	6.29	6	31.250	6.99	4
57.6	62.500	8.51	4	62.500	8.51	3	52.083	9.58	2
HIGH	1.221	-	255	0.977	-	255	0.610	-	255
LOW	312.500	-	0	250.000	-	0	156.250	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207	0.301	0.33	185
1.2	1.202	0.17	51	1.216	1.33	46
2.4	2.404	0.17	25	2.432	1.33	22
9.6	8.929	6.99	6	9.322	2.90	5
19.2	20.833	8.51	2	18.643	2.90	2
28.8	31.250	8.51	1	-	-	-
33.6	-	-	-	-	-	-
57.6	62.500	8.51	0	55.930	2.90	0
HIGH	0.244	-	255	0.218	-	255
LOW	62.500	-	0	55.930	-	0

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

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	-	-	-	-	-	-	-	-	-
2.4	-	-	-	-	-	-	2.441	1.71	255
9.6	9.615	0.16	129	9.615	0.16	103	9.615	0.16	64
19.2	19.231	0.16	64	19.231	0.16	51	19.531	1.72	31
28.8	29.070	0.94	42	29.412	2.13	33	28.409	1.36	21
33.6	33.784	0.55	36	33.333	0.79	29	32.895	2.10	18
57.6	59.524	3.34	20	58.824	2.13	16	56.818	1.36	10
HIGH	4.883	-	255	3.906	-	255	2.441	-	255
LOW	1250.000	-	0	1000.000	-	0	625.000	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-
1.2	1.202	0.17	207	1.203	0.25	185
2.4	2.404	0.17	103	2.406	0.25	92
9.6	9.615	0.16	25	9.727	1.32	22
19.2	19.231	0.16	12	18.643	2.90	11
28.8	27.798	3.55	8	27.965	2.90	7
33.6	35.714	6.29	6	31.960	4.88	6
57.6	62.500	8.51	3	55.930	2.90	3
HIGH	0.977	-	255	0.874	-	255
LOW	250.000	-	0	273.722	-	0

## 10.2 USART Asynchronous Mode

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

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

The USART Asynchronous module consists of the following important elements:

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

### 10.2.1 USART ASYNCHRONOUS TRANSMITTER

The USART transmitter block diagram is shown in Figure 10-1. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the STOP bit has been transmitted from the previous load. As soon as the STOP bit is transmitted, the TSR is loaded with new data from the TXREG register (if available). Once the TXREG register transfers the data to the TSR register (occurs in one T<sub>CY</sub>), the TXREG register is empty and flag bit TXIF (PIR1<4>) is set. This interrupt can be enabled/disabled by setting/clearing enable bit TXIE

(PIE1<4>). Flag bit TXIF will be set, regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicates the status of the TXREG register, another bit TRMT (TXSTA<1>) shows the status of the TSR register. Status bit TRMT is a read only bit, which is set when the TSR register is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty.

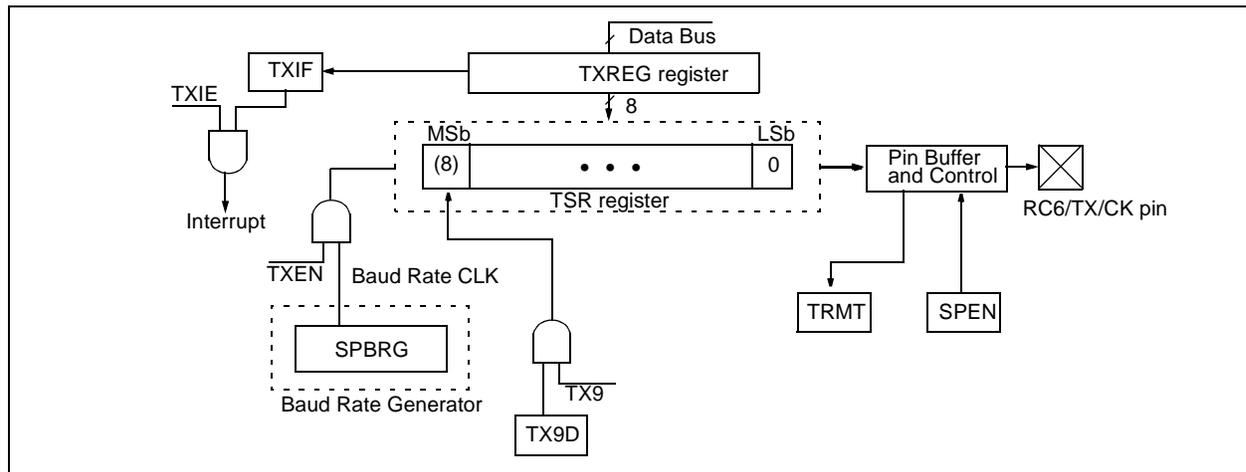
**Note 1:** The TSR register is not mapped in data memory, so it is not available to the user.

**2:** Flag bit TXIF is set when enable bit TXEN is set. TXIF is cleared by loading TXREG.

Transmission is enabled by setting enable bit TXEN (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data and the baud rate generator (BRG) has produced a shift clock (Figure 10-2). The transmission can also be started by first loading the TXREG register and then setting enable bit TXEN. Normally, when transmission is first started, the TSR register is empty. At that point, transfer to the TXREG register will result in an immediate transfer to TSR, resulting in an empty TXREG. A back-to-back transfer is thus possible (Figure 10-3). Clearing enable bit TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. As a result, the RC6/TX/CK pin will revert to hi-impedance.

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

**FIGURE 10-1: USART TRANSMIT BLOCK DIAGRAM**

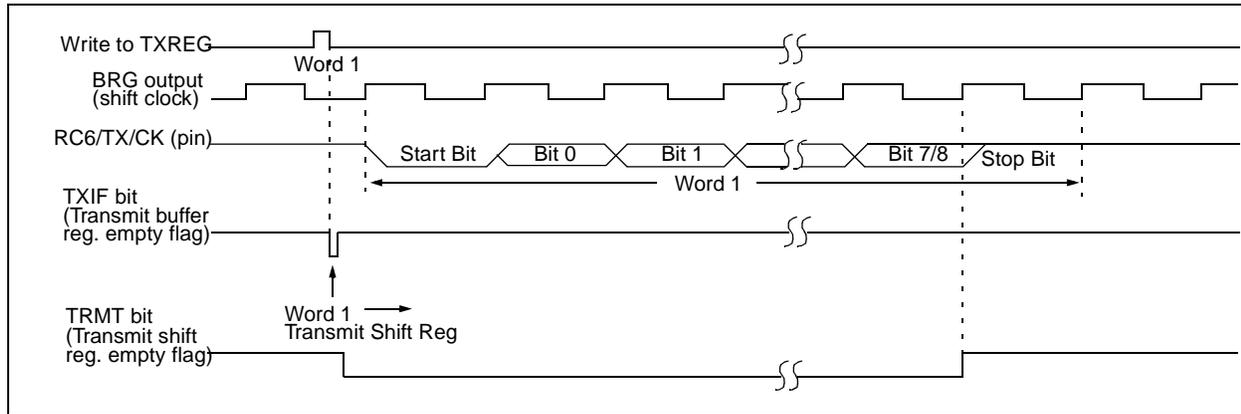


# PIC16F87X

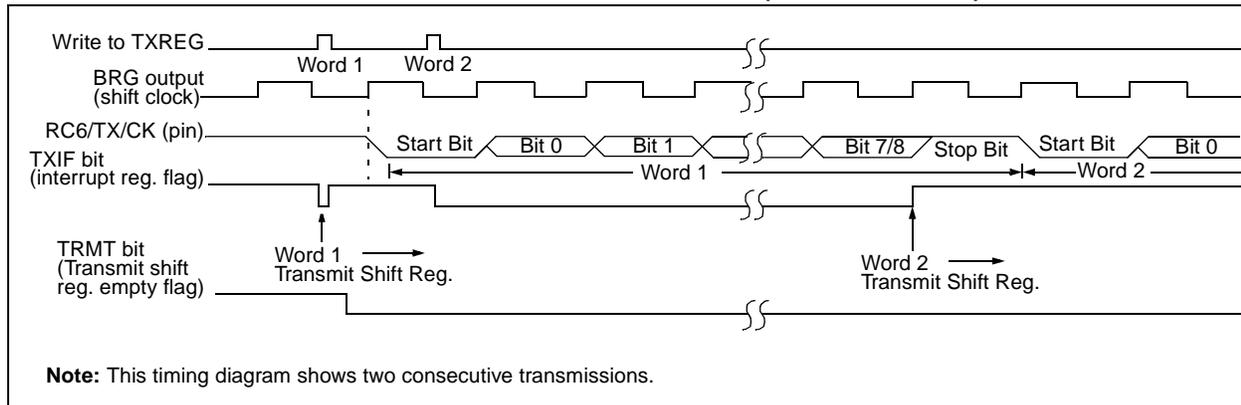
Steps to follow when setting up an Asynchronous Transmission:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH. (Section 10.1)
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set transmit bit TX9.
5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).

**FIGURE 10-2: ASYNCHRONOUS MASTER TRANSMISSION**



**FIGURE 10-3: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)**



**TABLE 10-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

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

**Note 1:** Bits PSPIE and PSPIF are reserved on the PIC16F873/876; always maintain these bits clear.

## 10.2.2 USART ASYNCHRONOUS RECEIVER

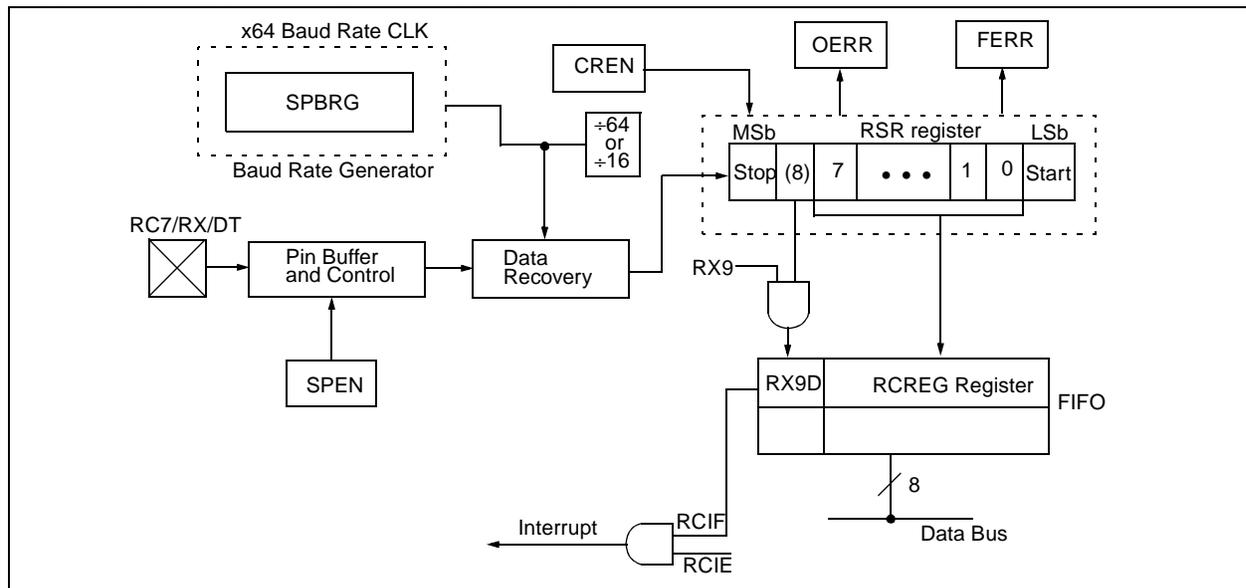
The receiver block diagram is shown in Figure 10-4. The data is received on the RC7/RX/DT pin and drives the data recovery block. The data recovery block is actually a high speed shifter operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at FOSC.

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

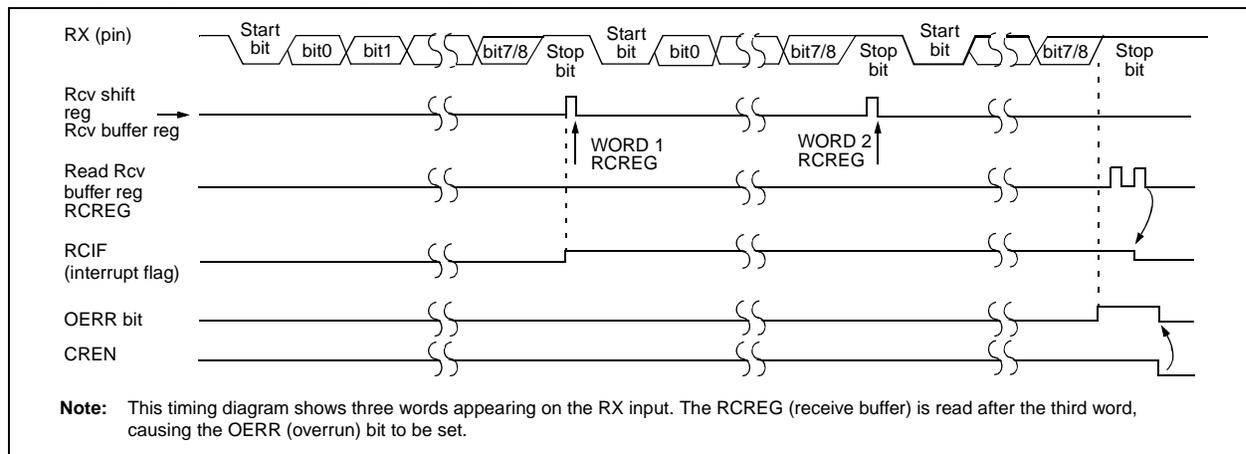
The heart of the receiver is the receive (serial) shift register (RSR). After sampling the STOP bit, the received data in the RSR is transferred to the RCREG register (if it is empty). If the transfer is complete, flag bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled/disabled by setting/clearing 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 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, so it is 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 RCREG register in order not to lose the old FERR and RX9D information.

**FIGURE 10-4: USART RECEVE BLOCK DIAGRAM**



**FIGURE 10-5: ASYNCHRONOUS RECEPTION**



# PIC16F87X

Steps to follow when setting up an Asynchronous Reception:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH. (Section 10.1).
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit RCIE.
4. If 9-bit reception is desired, then set bit RX9.
5. Enable the reception by setting bit CREN.
6. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE is set.
7. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG register.
9. If any error occurred, clear the error by clearing enable bit CREN.

**TABLE 10-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

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

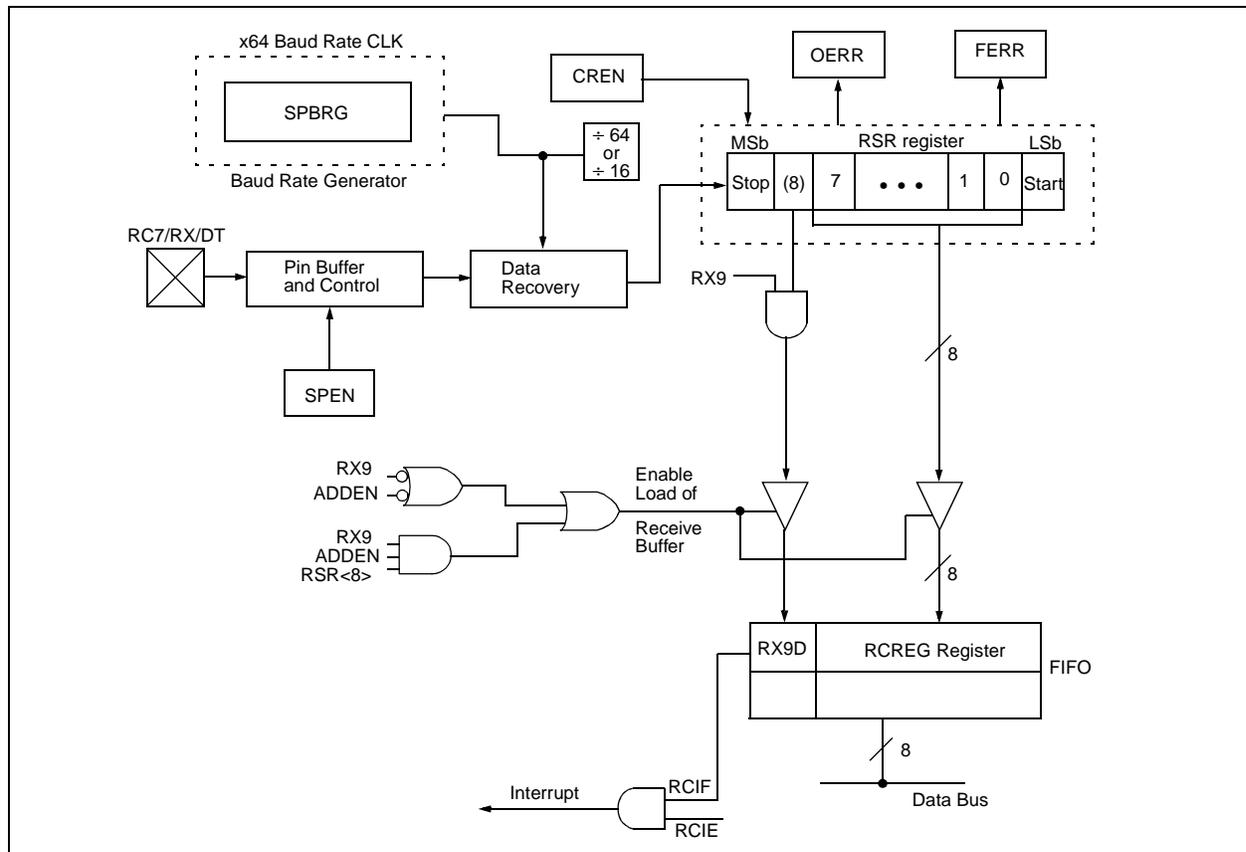
**Note 1:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

## 10.2.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

Steps to follow when setting up an Asynchronous Reception with Address Detect Enabled:

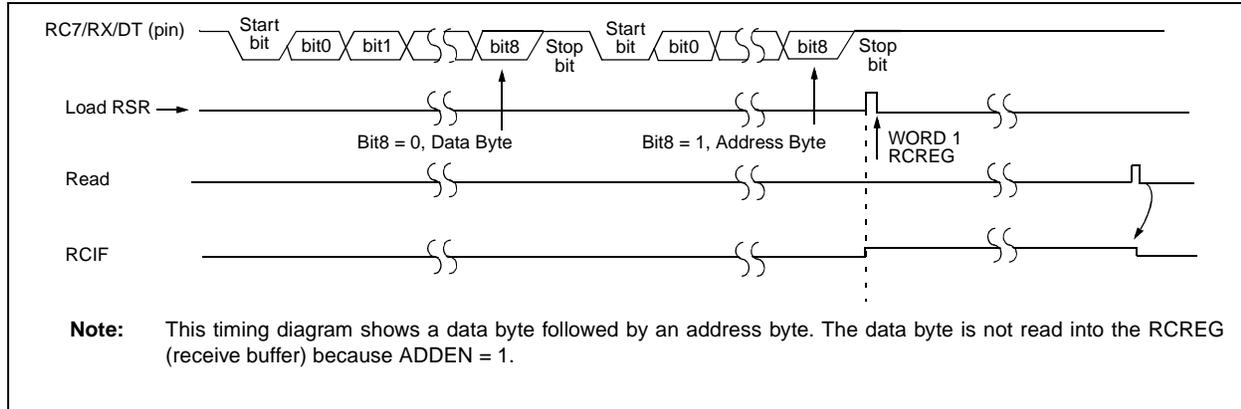
- Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH.
- Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
- If interrupts are desired, then set enable bit RCIE.
- Set bit RX9 to enable 9-bit reception.
- Set ADDEN to enable address detect.
- Enable the reception by setting enable bit CREN.
- Flag bit RCIF will be set when reception is complete, and an interrupt will be generated if enable bit RCIE was set.
- Read the RCSTA register to get the ninth bit and determine if any error occurred during reception.
- Read the 8-bit received data by reading the RCREG register, to determine if the device is being addressed.
- If any error occurred, clear the error by clearing enable bit CREN.
- If the device has been addressed, clear the ADDEN bit to allow data bytes and address bytes to be read into the receive buffer, and interrupt the CPU.

**FIGURE 10-6: USART RECEIVE BLOCK DIAGRAM**

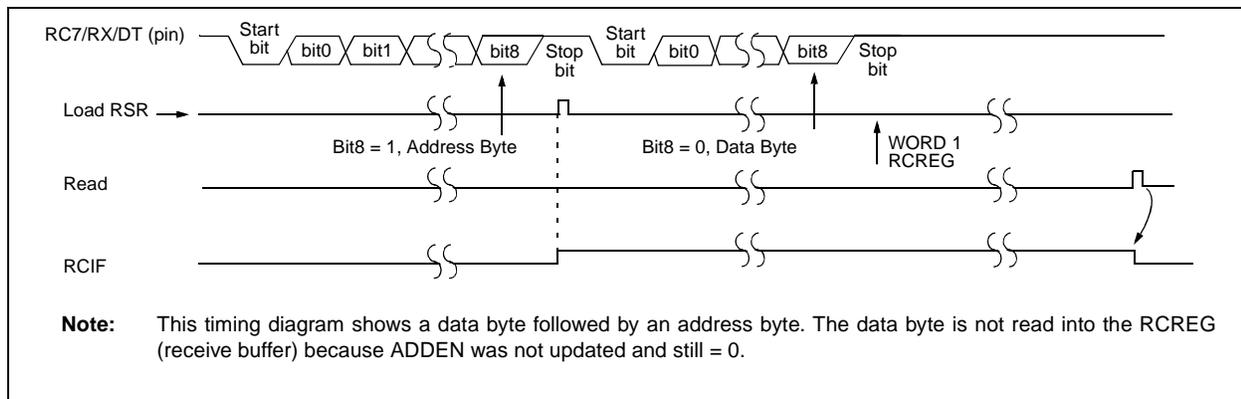


# PIC16F87X

**FIGURE 10-7: ASYNCHRONOUS RECEPTION WITH ADDRESS DETECT**



**FIGURE 10-8: ASYNCHRONOUS RECEPTION WITH ADDRESS BYTE FIRST**



**TABLE 10-7: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for Asynchronous Reception.

**Note 1:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

## 10.3 USART Synchronous Master Mode

In Synchronous Master mode, the data is transmitted in a half-duplex manne (i.e., transmission and reception do not occur at the same time). When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting bit SYNC (TXSTA<4>). In addition, enable bit SPEN (RCSTA<7>) is set in order to configure the RC6/TX/CK and RC7/RX/DT I/O pins to CK (clock) and DT (data) lines respectively. The Master mode indicates that the processor transmits the master clock on the CK line. The Master mode is entered by setting bit CSRC (TXSTA<7>).

### 10.3.1 USART SYNCHRONOUS MASTER TRANSMISSION

The USART transmitter block diagram is shown in Figure 10-6. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer register TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once the TXREG register transfers the data to the TSR register (occurs in one Tcycle), the TXREG is empty and interrupt bit TXIF (PIR1<4>) is set. The interrupt can be enabled/disabled by setting/clearing enable bit TXIE (PIE1<4>). Flag bit TXIF will be set regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicates the status of the TXREG register, another bit TRMT (TXSTA<1>) shows the status of the TSR register. TRMT is a read only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory, so it is not available to the user.

Transmission is enabled by setting enable bit TXEN (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data. The first data bit will be shifted out on the next available rising edge of the clock on the CK line. Data out is stable around the falling edge of the synchronous clock (Figure 10-9). The transmission can also be started by first loading the TXREG register and then setting bit TXEN (Figure 10-10). This is advantageous when slow baud rates are selected, since the BRG is kept in reset when bits TXEN, CREN and SREN are clear. Setting enable bit TXEN will start the BRG, creating a shift clock immediately. Normally, when transmission is first started, the TSR register is empty, so a transfer to the TXREG register will result in an immediate transfer to TSR resulting in an empty TXREG. Back-to-back transfers are possible.

Clearing enable bit TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. The DT and CK pins will revert to hi-impedance. If either bit CREN or bit SREN is set during a transmission, the transmission is aborted and the DT

pin reverts to a hi-impedance state (for a reception). The CK pin will remain an output if bit CSRC is set (internal clock). The transmitter logic, however, is not reset, although it is disconnected from the pins. In order to reset the transmitter, the user has to clear bit TXEN. If bit SREN is set (to interrupt an on-going transmission and receive a single word), then after the single word is received, bit SREN will be cleared and the serial port will revert back to transmitting, since bit TXEN is still set. The DT line will immediately switch from hi-impedance receive mode to transmit and start driving. To avoid this, bit TXEN should be cleared.

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

Steps to follow when setting up a Synchronous Master Transmission:

1. Initialize the SPBRG register for the appropriate baud rate (Section 10.1).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set bit TX9.
5. Enable the transmission by setting bit TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG register.

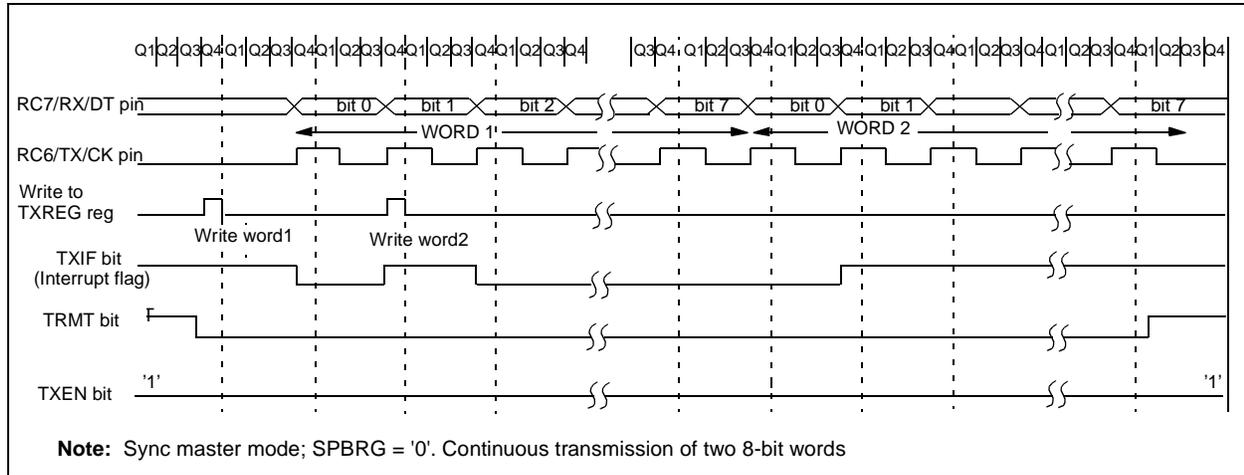
# PIC16F87X

**TABLE 10-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

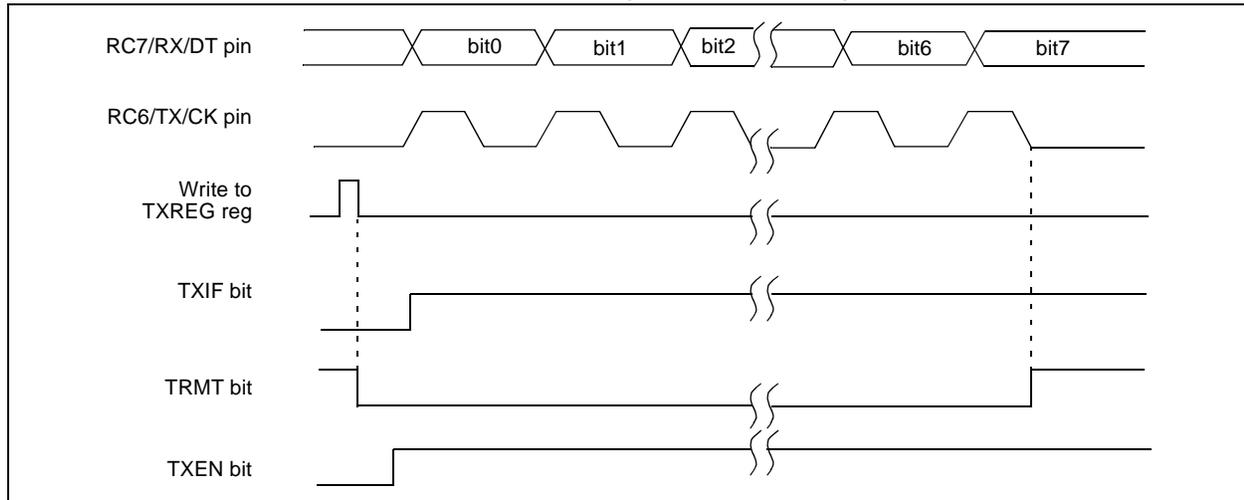
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.  
**Note 1:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

**FIGURE 10-9: SYNCHRONOUS TRANSMISSION**



**FIGURE 10-10: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**



## 10.3.2 USART SYNCHRONOUS MASTER RECEPTION

Once synchronous mode is selected, reception is enabled by setting either enable bit SREN (RCSTA<5>) or enable bit CREN (RCSTA<4>). Data is sampled on the RC7/RX/DT pin on the falling edge of the clock. If enable bit SREN is set, then only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set, CREN takes precedence. After clocking the last bit, the received data in the Receive Shift Register (RSR) is transferred to the RCREG register (if it is empty). When the transfer is complete, interrupt flag bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit RCIE (PIE1<5>). Flag bit RCIF is a read only bit, which is reset by the hardware. In this case, it is reset when the RCREG register has been read and is empty. The RCREG is a double buffered register (i.e., it is a two deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR register. On the clocking of the last bit of the third byte, if the RCREG register is still full, then overrun error bit OERR (RCSTA<1>) is set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Bit OERR has to be cleared in software (by clearing bit CREN). If bit OERR is set, transfers from the RSR to the RCREG are inhibited, so it is essential to clear bit

OERR if it is set. The ninth receive bit is buffered the same way as the receive data. Reading the RCREG register will load bit RX9D with a new value, therefore it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old RX9D information.

Steps to follow when setting up a Synchronous Master Reception:

1. Initialize the SPBRG register for the appropriate baud rate. (Section 10.1)
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Ensure bits CREN and SREN are clear.
4. If interrupts are desired, then set enable bit RCIE.
5. If 9-bit reception is desired, then set bit RX9.
6. If a single reception is required, set bit SREN. For continuous reception set bit CREN.
7. Interrupt flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing bit CREN.

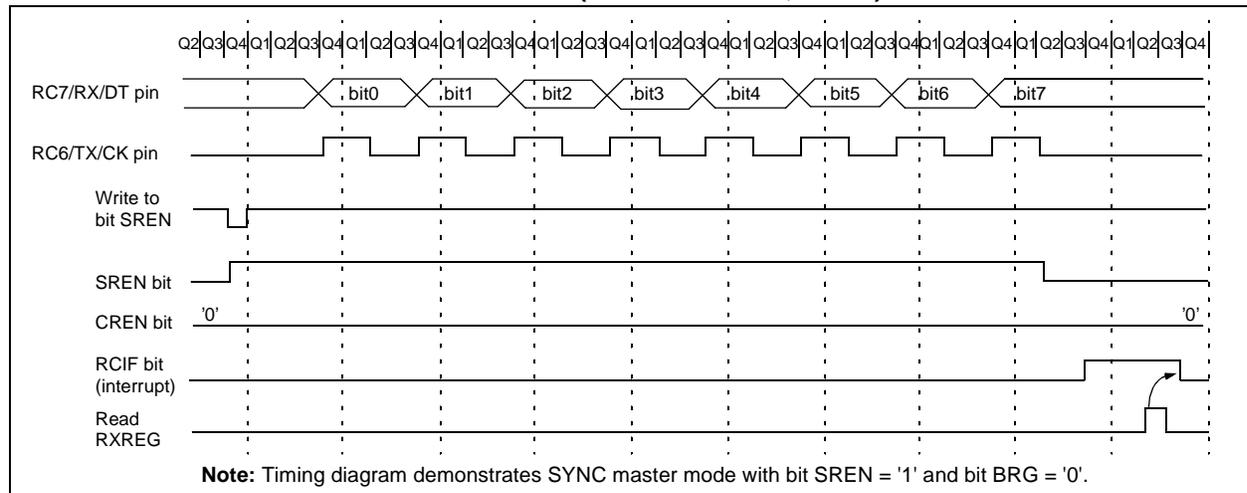
**TABLE 10-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

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

**Note 1:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

**FIGURE 10-11: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)**



## 10.4 USART Synchronous Slave Mode

Synchronous slave mode differs from the Master mode in the fact that the shift clock is supplied externally at the RC6/TX/CK pin (instead of being supplied internally in master mode). This allows the device to transfer or receive data while in SLEEP mode. Slave mode is entered by clearing bit CSRC (TXSTA<7>).

### 10.4.1 USART SYNCHRONOUS SLAVE TRANSMIT

The operation of the synchronous master and slave modes are identical except in the case of the SLEEP mode.

If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

- a) The first word will immediately transfer to the TSR register and transmit.
- b) The second word will remain in TXREG register.
- c) Flag bit TXIF will not be set.
- d) When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and flag bit TXIF will now be set.
- e) If enable bit TXIE is set, the interrupt will wake the chip from SLEEP and if the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

Steps to follow when setting up a Synchronous Slave Transmission:

1. Enable the synchronous slave serial port by setting bits SYNC and SPEN and clearing bit CSRC.
2. Clear bits CREN and SREN.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set bit TX9.
5. Enable the transmission by setting enable bit TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG register.

### 10.4.2 USART SYNCHRONOUS SLAVE RECEPTION

The operation of the synchronous master and slave modes is identical, except in the case of the SLEEP mode. Bit SREN is a "don't care" in slave mode.

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

Steps to follow when setting up a Synchronous Slave Reception:

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

**TABLE 10-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Slave Transmission.

**Note 1:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

**TABLE 10-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Slave Reception.

**Note 1:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

# PIC16F87X

---

NOTES:

## 11.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) Converter module has five inputs for the 28-pin devices and eight for the other devices.

The analog input charges a sample and hold capacitor. The output of the sample and hold capacitor is the input into the converter. The converter then generates a digital result of this analog level via successive approximation. The A/D conversion of the analog input signal results in a corresponding 10-bit digital number. The A/D module has high and low voltage reference input that is software selectable to some combination of VDD, VSS, RA2 or RA3.

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

The A/D module has four registers. These registers are:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register0 (ADCON0)
- A/D Control Register1 (ADCON1)

The ADCON0 register, shown in Register 11-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 11-2, configures the functions of the port pins. The port pins can be configured as analog inputs (RA3 can also be the voltage reference) or as digital I/O.

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

### REGISTER 11-1: ADCON0 REGISTER (ADDRESS: 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit7							bit0

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

bit 7-6: **ADCS1:ADCS0: A/D Conversion Clock Select bits**  
00 = Fosc/2  
01 = Fosc/8  
10 = Fosc/32  
11 = FRC (clock derived from an RC oscillation)

bit 5-3: **CHS2:CHS0: Analog Channel Select bits**  
000 = channel 0, (RA0/AN0)  
001 = channel 1, (RA1/AN1)  
010 = channel 2, (RA2/AN2)  
011 = channel 3, (RA3/AN3)  
100 = channel 4, (RA5/AN4)  
101 = channel 5, (RE0/AN5)<sup>(1)</sup>  
110 = channel 6, (RE1/AN6)<sup>(1)</sup>  
111 = channel 7, (RE2/AN7)<sup>(1)</sup>

bit 2: **GO/DONE: A/D Conversion Status bit**  
If ADON = 1  
1 = A/D conversion in progress (setting this bit starts the A/D conversion)  
0 = A/D conversion not in progress (This bit is automatically cleared by hardware when the A/D conversion is complete)

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

bit 0: **ADON: A/D On bit**  
1 = A/D converter module is operating  
0 = A/D converter module is shutoff and consumes no operating current

**Note 1:** These channels are not available on the 28-pin devices.

# PIC16F87X

## REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit7				bit0			

R = Readable bit  
W = Writable bit  
U = Unimplemented bit, read as '0'  
- n = Value at POR reset

bit 7: **ADFM: A/D Result format select**  
1 = Right Justified. 6 most significant bits of ADRESH are read as '0'.  
0 = Left Justified. 6 least significant bits of ADRESL are read as '0'.

bit 6-4: **Unimplemented: Read as '0'**

bit 3-0: **PCFG3:PCFG0: A/D Port Configuration Control bits**

PCFG3: PCFG0	AN7 <sup>(1)</sup> RE2	AN6 <sup>(1)</sup> RE1	AN5 <sup>(1)</sup> RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN / Refs <sup>(2)</sup>
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

A = Analog input

D = Digital I/O

**Note 1:** These channels are not available on the 28-pin devices.

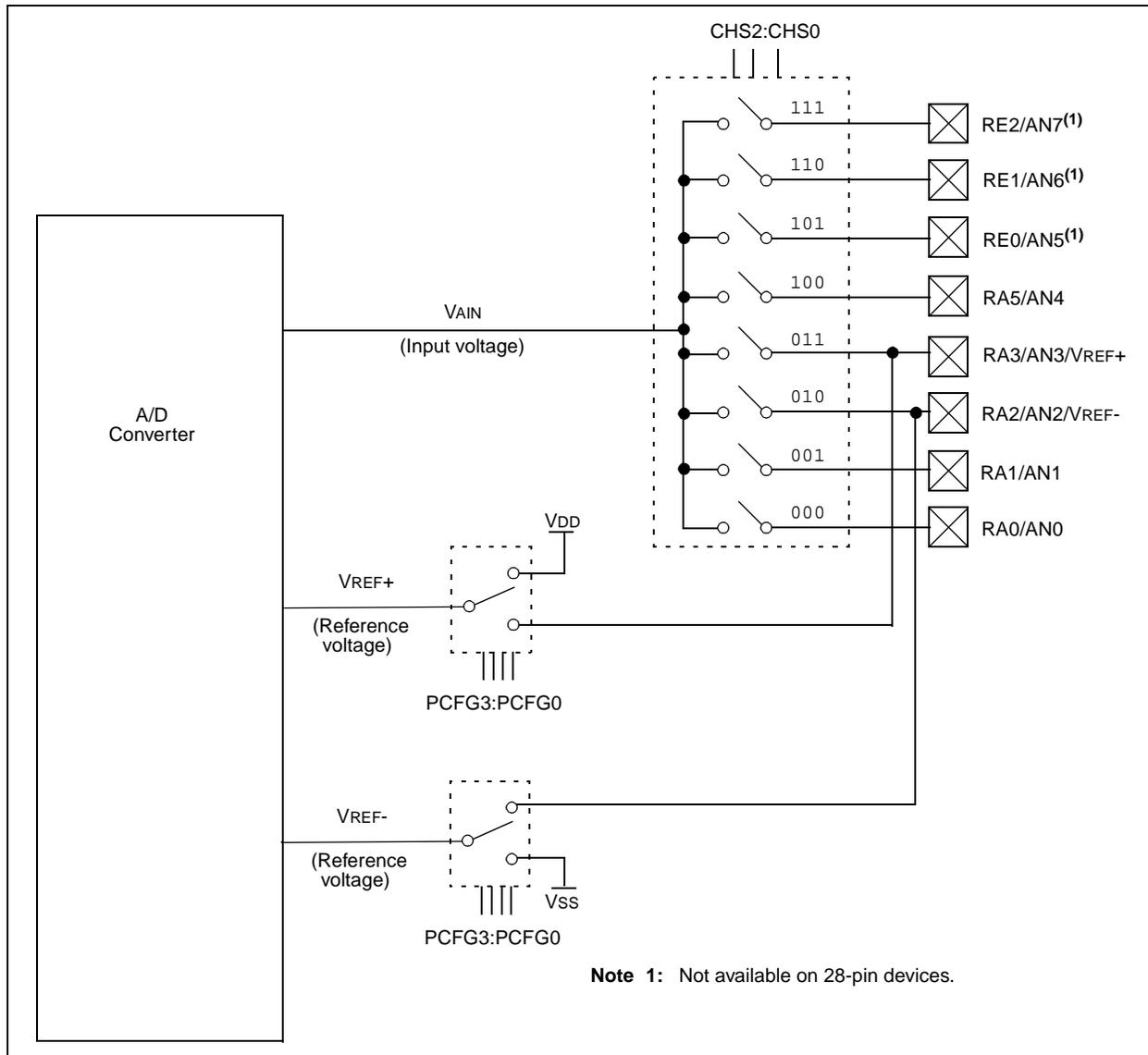
**Note 2:** This column indicates the number of analog channels available as A/D inputs and the number of analog channels used as voltage reference inputs.

The ADRESH:ADRESL registers contain the 10-bit result of the A/D conversion. When the A/D conversion is complete, the result is loaded into this A/D result register pair, the  $\overline{\text{GO/DONE}}$  bit (ADCON0<2>) is cleared and the A/D interrupt flag bit ADIF is set. The block diagram of the A/D module is shown in Figure 11-1.

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs. To determine sample time, see Section 11.1. After this acquisition time has elapsed, the A/D conversion can be started. The following steps should be followed for doing an A/D conversion:

1. Configure the A/D module:
  - Configure analog pins / voltage reference / and digital I/O (ADCON1)
  - Select A/D input channel (ADCON0)
  - Select A/D conversion clock (ADCON0)
  - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
  - Clear ADIF bit
  - Set ADIE bit
  - Set GIE bit
3. Wait the required acquisition time.
4. Start conversion:
  - Set  $\overline{\text{GO/DONE}}$  bit (ADCON0)
5. Wait for A/D conversion to complete, by either:
  - Polling for the  $\overline{\text{GO/DONE}}$  bit to be clearedOR
  - Waiting for the A/D interrupt
6. Read A/D Result register pair (ADRESH:ADRESL), clear bit ADIF if required.
7. For next conversion, go to step 1 or step 2 as required. The A/D conversion time per bit is required. The A/D conversion time per bit is defined as TAD. A minimum wait of 2TAD is required before next acquisition starts.

**FIGURE 11-1: A/D BLOCK DIAGRAM**



## 11.1 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 11-2. The source impedance ( $R_S$ ) and the internal sampling switch ( $R_{SS}$ ) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch ( $R_{SS}$ ) impedance varies over the device voltage ( $V_{DD}$ ), Figure 11-2. **The maximum recommended impedance for analog sources is 10 k $\Omega$ .** As the impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (changed), this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, Equation 11-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

To calculate the minimum acquisition time,  $T_{ACQ}$ , see the PICmicro™ Mid-Range Reference Manual (DS33023).

## EQUATION 11-1: ACQUISITION TIME

$$\begin{aligned}
 T_{ACQ} &= \text{Amplifier Settling Time} + \\
 &\quad \text{Hold Capacitor Charging Time} + \\
 &\quad \text{Temperature Coefficient} \\
 &= T_{AMP} + T_C + T_{COFF} \\
 &= 2\mu\text{S} + T_C + [(\text{Temperature} - 25^\circ\text{C})(0.05\mu\text{S}/^\circ\text{C})] \\
 T_C &= \text{CHOLD} (\text{RIC} + \text{RSS} + \text{RS}) \ln(1/2047) \\
 &= -120\text{pF} (1\text{k}\Omega + 7\text{k}\Omega + 10\text{k}\Omega) \ln(0.0004885) \\
 &= 16.47\mu\text{S} \\
 T_{ACQ} &= 2\mu\text{S} + 16.47\mu\text{S} + [(50^\circ\text{C} - 25^\circ\text{C})(0.05\mu\text{S}/^\circ\text{C})] \\
 &= 19.72\mu\text{S}
 \end{aligned}$$

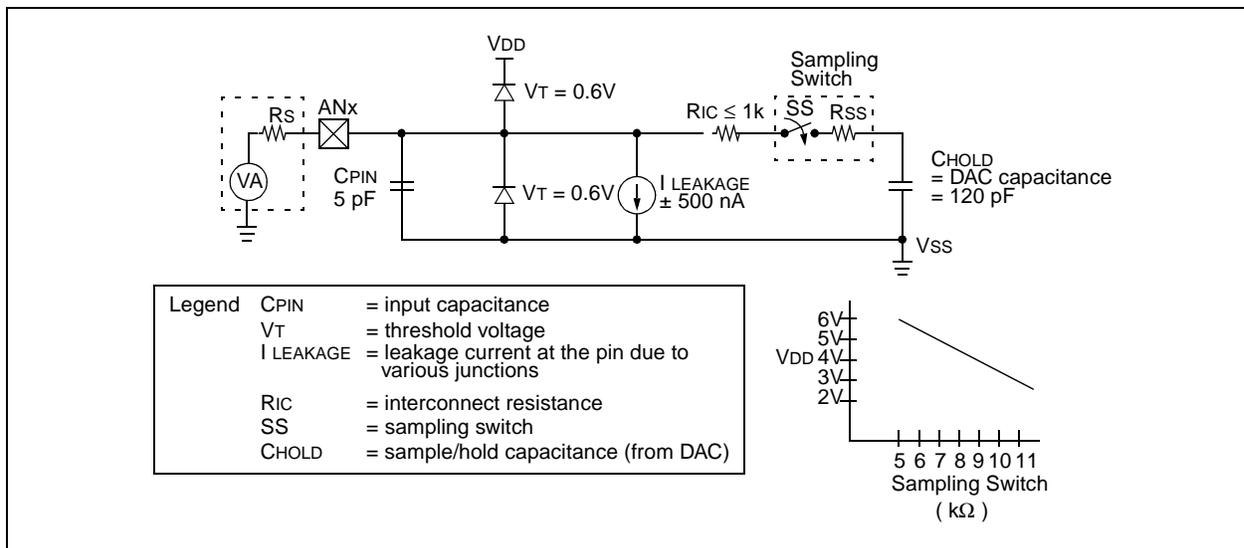
**Note 1:** The reference voltage ( $V_{REF}$ ) has no effect on the equation, since it cancels itself out.

**2:** The charge holding capacitor ( $\text{CHOLD}$ ) is not discharged after each conversion.

**3:** The maximum recommended impedance for analog sources is  $10\text{ k}\Omega$ . This is required to meet the pin leakage specification.

**4:** After a conversion has completed, a  $2.0T_{AD}$  delay must complete before acquisition can begin again. During this time, the holding capacitor is not connected to the selected A/D input channel.

**FIGURE 11-2: ANALOG INPUT MODEL**



# PIC16F87X

## 11.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as  $T_{AD}$ . The A/D conversion requires a minimum  $12T_{AD}$  per 10-bit conversion. The source of the A/D conversion clock is software selected. The four possible options for  $T_{AD}$  are:

- $2T_{OSC}$
- $8T_{OSC}$
- $32T_{OSC}$
- Internal RC oscillator

For correct A/D conversions, the A/D conversion clock ( $T_{AD}$ ) must be selected to ensure a minimum  $T_{AD}$  time of  $1.6 \mu s$ .

Table 11-1 shows the resultant  $T_{AD}$  times derived from the device operating frequencies and the A/D clock source selected.

**TABLE 11-1:  $T_{AD}$  vs. MAXIMUM DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (C))**

AD Clock Source ( $T_{AD}$ )		Maximum Device Frequency
Operation	ADCS1:ADCS0	Max.
$2T_{OSC}$	00	1.25 MHz
$8T_{OSC}$	01	5 MHz
$32T_{OSC}$	10	20 MHz
RC <sup>(1, 2, 3)</sup>	11	Note 1

- Note 1:** The RC source has a typical  $T_{AD}$  time of  $4 \mu s$  but can vary between  $2-6 \mu s$ .
- 2:** When the device frequencies are greater than 1 MHz, the RC A/D conversion clock source is only recommended for sleep operation.
- 3:** For extended voltage devices (LC), please refer to the Electrical Specifications section.

## 11.3 Configuring Analog Port Pins

The  $ADCON1$ , and  $TRIS$  registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding  $TRIS$  bits set (input). If the  $TRIS$  bit is cleared (output), the digital output level ( $V_{OH}$  or  $V_{OL}$ ) will be converted.

The A/D operation is independent of the state of the  $CHS2:CHS0$  bits and the  $TRIS$  bits.

**Note 1:** When reading the port register, any pin configured as an analog input channel will read as cleared (a low level). Pins configured as digital inputs will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.

- 2:** Analog levels on any pin that is defined as a digital input (including the  $AN7:AN0$  pins), may cause the input buffer to consume current that is out of the device specifications.

## 11.4 A/D Conversions

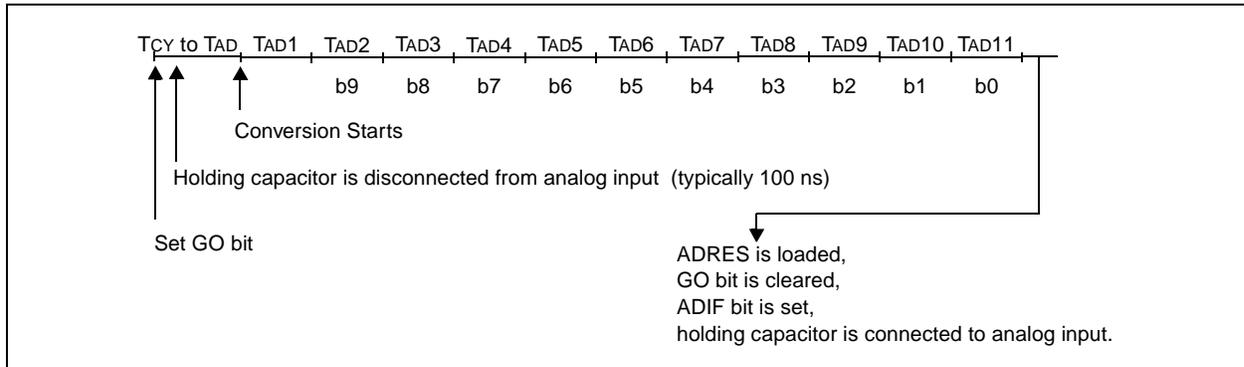
Clearing the  $\overline{\text{GO/DONE}}$  bit during a conversion will abort the current conversion. The A/D result register pair will NOT be updated with the partially completed A/D conversion sample. That is, the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers). After the A/D conversion is aborted, a 2TAD wait is

required before the next acquisition is started. After this 2TAD wait, acquisition on the selected channel is automatically started.

In Figure 11-3, after the GO bit is set, the first time segment has a minimum of TCY and a maximum of TAD.

**Note:** The  $\overline{\text{GO/DONE}}$  bit should **NOT** be set in the same instruction that turns on the A/D.

**FIGURE 11-3: A/D CONVERSION TAD CYCLES**



# PIC16F87X

## 11.4.1 A/D RESULT REGISTERS

The ADRESH:ADRESL register pair is the location where the 10-bit A/D result is loaded at the completion of the A/D conversion. This register pair is 16-bits wide. The A/D module gives the flexibility to left or right justify the 10-bit result in the 16-bit result register. The A/D Format Select bit (ADFM) controls this justification. Figure 11-4 shows the operation of the A/D result justification. The extra bits are loaded with '0's'. When an A/D result will not overwrite these locations (A/D disable), these registers may be used as two general purpose 8-bit registers.

## 11.5 A/D Operation During Sleep

The A/D module can operate during SLEEP mode. This requires that the A/D clock source be set to RC (ADCS1:ADCS0 = 11). When the RC clock source is selected, the A/D module waits one instruction cycle before starting the conversion. This allows the SLEEP instruction to be executed, which eliminates all digital switching noise from the conversion. When the conversion is completed the GO/DONE bit will be cleared and the result loaded into the ADRES register. If the A/D interrupt is enabled, the device will wake-up from

SLEEP. If the A/D interrupt is not enabled, the A/D module will then be turned off, although the ADON bit will remain set.

When the A/D clock source is another clock option (not RC), a SLEEP instruction will cause the present conversion to be aborted and the A/D module to be turned off, though the ADON bit will remain set.

Turning off the A/D places the A/D module in its lowest current consumption state.

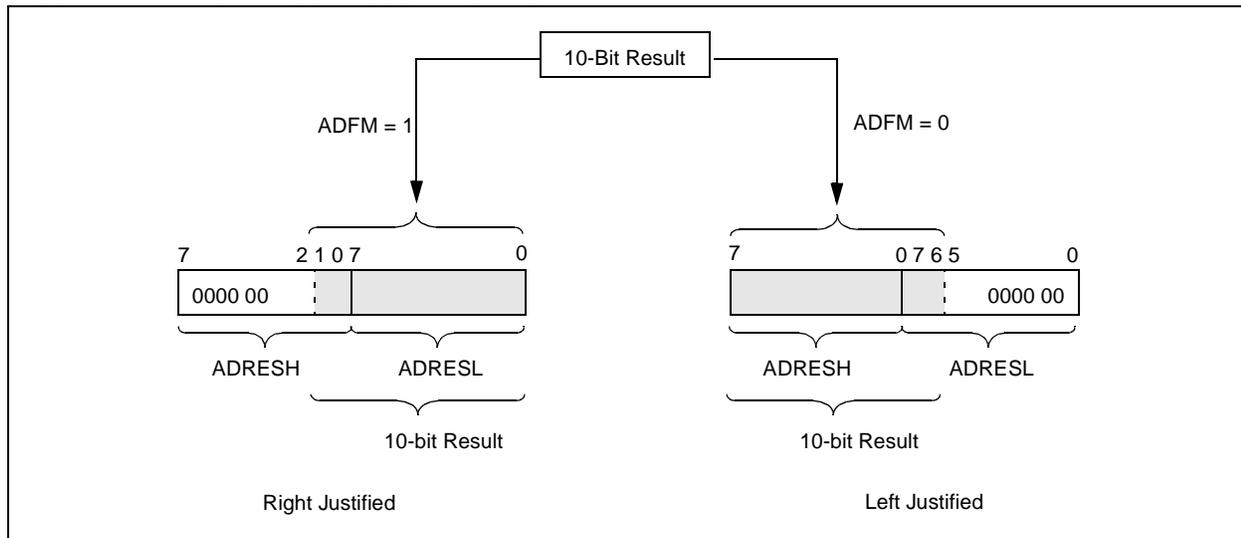
**Note:** For the A/D module to operate in SLEEP, the A/D clock source must be set to RC (ADCS1:ADCS0 = 11). To allow the conversion to occur during SLEEP, ensure the SLEEP instruction immediately follows the instruction that sets the GO/DONE bit.

## 11.6 Effects of a Reset

A device reset forces all registers to their reset state. This forces the A/D module to be turned off, and any conversion is aborted.

The value that is in the ADRESH:ADRESL registers is not modified for a Power-on Reset. The ADRESH:ADRESL registers will contain unknown data after a Power-on Reset.

FIGURE 11-4: A/D RESULT JUSTIFICATION



**TABLE 11-2: REGISTERS/BITS ASSOCIATED WITH A/D**

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR, BOR	MCLR, WDT
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON	0000 00-0	0000 00-0
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	--0u 0000
89h <sup>(1)</sup>	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111
09h <sup>(1)</sup>	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used for A/D conversion.

**Note 1:** These registers/bits are not available on the 28-pin devices.

# PIC16F87X

---

NOTES:

## 12.0 SPECIAL FEATURES OF THE CPU

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

- OSC Selection
- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- SLEEP
- Code protection
- ID locations
- In-Circuit Serial Programming
- Low Voltage In-Circuit Serial Programming
- In-Circuit Debugger

These devices have a watchdog timer, which can be shut off only through configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only. It is designed to keep the part in reset while the power supply stabilizes. With these two timers on-chip, most applications need no external reset circuitry.

SLEEP mode is designed to offer a very low current power-down mode. The user can wake-up from SLEEP through external reset, Watchdog Timer Wake-up, or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits are used to select various options.

Additional information on special features is available in the PICmicro™ Mid-Range Reference Manual, (DS33023).

## 12.1 Configuration Bits

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

The user will note that address 2007h is beyond the user program memory space. In fact, it belongs to the special test/configuration memory space (2000h - 3FFFh), which can be accessed only during programming.

# PIC16F87X

## REGISTER 12-1: CONFIGURATION WORD

CP1	CP0	DEBUG	—	WRT	CPD	LVP	BODEN	CP1	CP0	PWRTE	WDTE	F0SC1	F0SC0
													bit0

Register: CONFIG  
Address 2007h

bit13

bit 13-12:

bit 5-4: **CP1:CP0:** Flash Program Memory Code Protection bits <sup>(2)</sup>

- 11 = Code protection off
- 10 = 1F00h to 1FFFh code protected (PIC16F877, 876)
- 10 = 0F00h to 0FFFh code protected (PIC16F874, 873)
- 01 = 1000h to 1FFFh code protected (PIC16F877, 876)
- 01 = 0800h to 0FFFh code protected (PIC16F874, 873)
- 00 = 0000h to 1FFFh code protected (PIC16F877, 876)
- 00 = 0000h to 0FFFh code protected (PIC16F874, 873)

bit 11: **DEBUG:** In-Circuit Debugger Mode

- 1 = In-Circuit Debugger disabled, RB6 and RB7 are general purpose I/O pins.
- 0 = In-Circuit Debugger enabled, RB6 and RB7 are dedicated to the debugger.

bit 10: **Unimplemented: Read as '1'**

bit 9: **WRT:** Flash Program Memory Write Enable

- 1 = Unprotected program memory may be written to by EECON control
- 0 = Unprotected program memory may not be written to by EECON control

bit 8: **CPD:** Data EE Memory Code Protection

- 1 = Code protection off
- 0 = Data EEPROM memory code protected

bit 7: **LVP:** Low Voltage In-Circuit Serial Programming Enable bit

- 1 = RB3/PGM pin has PGM function, low voltage programming enabled
- 0 = RB3 is digital I/O, HV on  $\overline{MCLR}$  must be used for programming

bit 6: **BODEN:** Brown-out Reset Enable bit <sup>(1)</sup>

- 1 = BOR enabled
- 0 = BOR disabled

bit 3: **PWRTE:** Power-up Timer Enable bit <sup>(1)</sup>

- 1 = PWRT disabled
- 0 = PWRT enabled

bit 2: **WDTE:** Watchdog Timer Enable bit

- 1 = WDT enabled
- 0 = WDT disabled

bit 1-0: **FOSC1:FOSC0:** Oscillator Selection bits

- 11 = RC oscillator
- 10 = HS oscillator
- 01 = XT oscillator
- 00 = LP oscillator

**Note 1:** Enabling Brown-out Reset automatically enables Power-up Timer (PWRT), regardless of the value of bit  $\overline{PWRTE}$ . Ensure the Power-up Timer is enabled anytime Brown-out Reset is enabled.

**2:** All of the CP1:CP0 pairs have to be given the same value to enable the code protection scheme listed.

## 12.2 Oscillator Configurations

### 12.2.1 OSCILLATOR TYPES

The PIC16F87X can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

### 12.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In XT, LP or HS modes, a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 12-1). The PIC16F87X oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1/CLKIN pin (Figure 12-2).

TABLE 12-1: CERAMIC RESONATORS

Ranges Tested:			
Mode	Freq	OSC1	OSC2
XT	455 kHz	68 - 100 pF	68 - 100 pF
	2.0 MHz	15 - 68 pF	15 - 68 pF
	4.0 MHz	15 - 68 pF	15 - 68 pF
HS	8.0 MHz	10 - 68 pF	10 - 68 pF
	16.0 MHz	10 - 22 pF	10 - 22 pF
These values are for design guidance only. See notes at bottom of page.			
Resonators Used:			
455 kHz	Panasonic EFO-A455K04B	± 0.3%	
2.0 MHz	Murata Erie CSA2.00MG	± 0.5%	
4.0 MHz	Murata Erie CSA4.00MG	± 0.5%	
8.0 MHz	Murata Erie CSA8.00MT	± 0.5%	
16.0 MHz	Murata Erie CSA16.00MX	± 0.5%	
All resonators used did not have built-in capacitors.			

FIGURE 12-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)

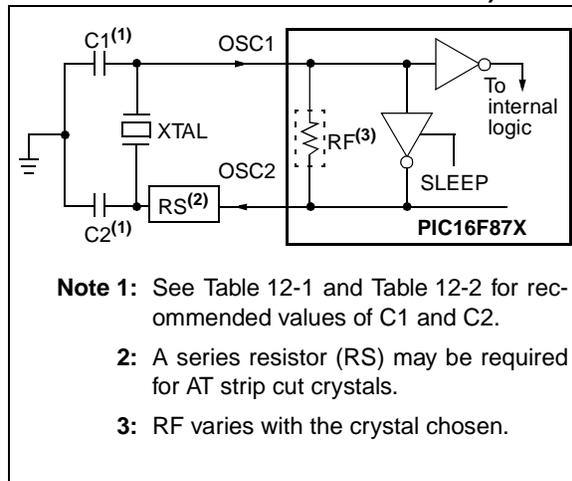
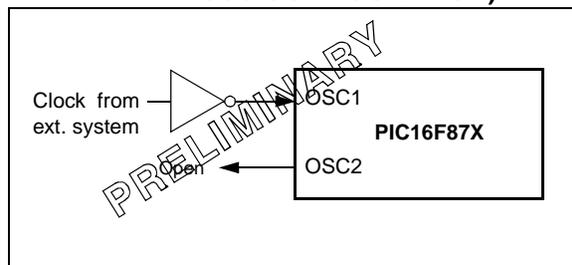


FIGURE 12-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)



# PIC16F87X

**TABLE 12-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR**

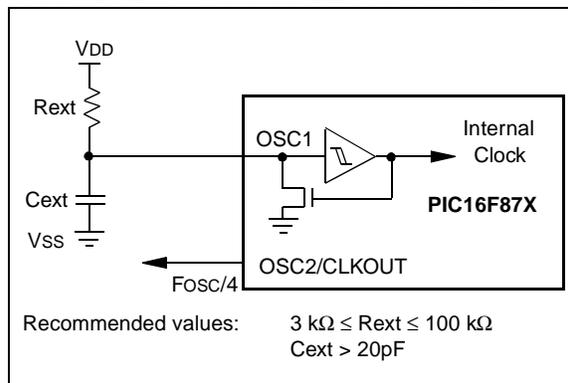
Osc Type	Crystal Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF
These values are for design guidance only. See notes at bottom of page.			
<b>Crystals Used</b>			
32 kHz	Epson C-001R32.768K-A	± 20 PPM	
200 kHz	STD XTL 200.000KHz	± 20 PPM	
1 MHz	ECS ECS-10-13-1	± 50 PPM	
4 MHz	ECS ECS-40-20-1	± 50 PPM	
8 MHz	EPSON CA-301 8.000M-C	± 30 PPM	
20 MHz	EPSON CA-301 20.000M-C	± 30 PPM	

- Note 1:** Higher capacitance increases the stability of oscillator but also increases the start-up time.
- Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.
  - Rs may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification.
  - When migrating from other PICmicro devices, oscillator performance should be verified.

## 12.2.3 RC OSCILLATOR

For timing insensitive applications, the “RC” device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (R<sub>EXT</sub>) and capacitor (C<sub>EXT</sub>) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low C<sub>EXT</sub> values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 12-3 shows how the R/C combination is connected to the PIC16F87X.

**FIGURE 12-3: RC OSCILLATOR MODE**



## 12.3 Reset

The PIC16F87X differentiates between various kinds of reset:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$  reset during normal operation
- $\overline{\text{MCLR}}$  reset during SLEEP
- WDT Reset (during normal operation)
- WDT Wake-up (during SLEEP)
- Brown-out Reset (BOR)

Some registers are not affected in any reset condition. Their status is unknown on POR and unchanged in any other reset. Most other registers are reset to a "reset state" on Power-on Reset (POR), on the  $\overline{\text{MCLR}}$  and

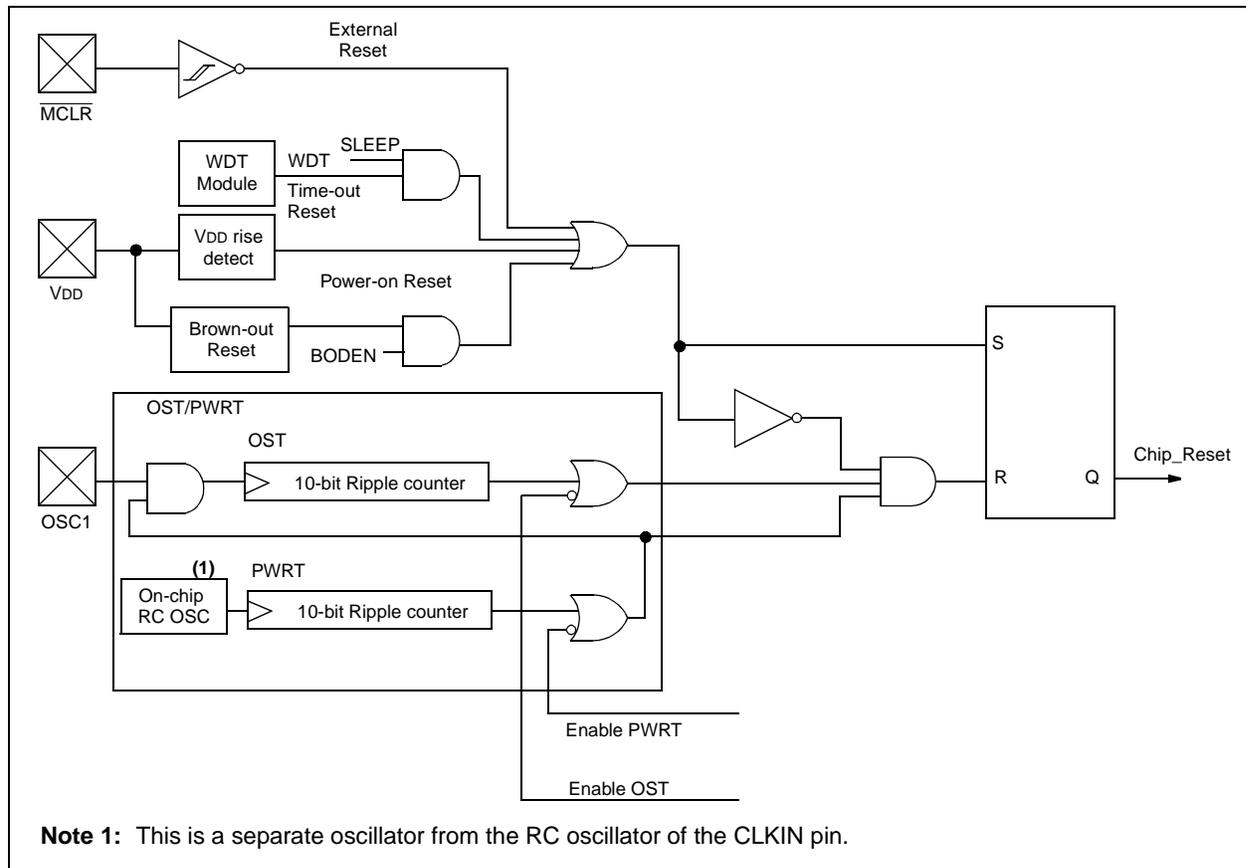
WDT Reset, on  $\overline{\text{MCLR}}$  reset during SLEEP, and Brown-out Reset (BOR). They are not affected by a WDT Wake-up, which is viewed as the resumption of normal operation. The  $\overline{\text{TO}}$  and  $\overline{\text{PD}}$  bits are set or cleared differently in different reset situations as indicated in Table 12-4. These bits are used in software to determine the nature of the reset. See Table 12-6 for a full description of reset states of all registers.

A simplified block diagram of the on-chip reset circuit is shown in Figure 12-4.

These devices have a  $\overline{\text{MCLR}}$  noise filter in the  $\overline{\text{MCLR}}$  reset path. The filter will detect and ignore small pulses.

It should be noted that a WDT Reset does not drive  $\overline{\text{MCLR}}$  pin low.

**FIGURE 12-4: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**



## 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  $\overline{\text{MCLR}}$  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 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This ensures 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 100 $\mu$ S), 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  $\overline{\text{MCLR}}$  is kept low long enough, the time-outs will expire. Bringing  $\overline{\text{MCLR}}$  high will begin execution immediately. This is useful for testing purposes or to synchronize more than one PIC16CXX 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,  $\overline{\text{BOR}}$ . 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 BOR occurred. The  $\overline{\text{BOR}}$  bit is a "don't care" bit and is not necessarily predictable if the Brown-out Reset circuitry is disabled (by clearing bit BODEN in the Configuration Word).

Bit1 is  $\overline{\text{POR}}$  (Power-on Reset Status bit). It is cleared on a Power-on Reset and unaffected otherwise. The user must set this bit following a Power-on Reset.

**TABLE 12-3: TIME-OUT IN VARIOUS SITUATIONS**

Oscillator Configuration	Power-up		Brown-out	Wake-up from SLEEP
	$\overline{\text{PWRTÉ}} = 0$	$\overline{\text{PWRTÉ}} = 1$		
XT, HS, LP	72 ms + 1024Tosc	1024Tosc	72 ms + 1024Tosc	1024Tosc
RC	72 ms	—	72 ms	—

**TABLE 12-4: STATUS BITS AND THEIR SIGNIFICANCE**

POR	BOR	TO	PD	
0	x	1	1	Power-on Reset
0	x	0	x	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	x	x	0	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
1	0	1	1	Brown-out Reset
1	1	0	1	WDT Reset
1	1	0	0	WDT Wake-up
1	1	u	u	$\overline{\text{MCLR}}$ Reset during normal operation
1	1	1	0	$\overline{\text{MCLR}}$ Reset during SLEEP or interrupt wake-up from SLEEP

**TABLE 12-5: RESET CONDITION FOR SPECIAL REGISTERS**

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

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

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

# PIC16F87X

**TABLE 12-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS**

Register	Devices				Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset	Wake-up via WDT or Interrupt
	873	874	876	877			
W	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	873	874	876	877	N/A	N/A	N/A
TMR0	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	873	874	876	877	0000h	0000h	PC + 1 <sup>(2)</sup>
STATUS	873	874	876	877	0001 1xxx	000q quuu <sup>(3)</sup>	uuuq quuu <sup>(3)</sup>
FSR	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	873	874	876	877	--0x 0000	--0u 0000	--uu uuuu
PORTB	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTD	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTE	873	874	876	877	---- -xxx	---- -uuu	---- -uuu
PCLATH	873	874	876	877	---0 0000	---0 0000	---u uuuu
INTCON	873	874	876	877	0000 000x	0000 000u	uuuu uuuu <sup>(1)</sup>
PIR1	873	874	876	877	r000 0000	r000 0000	ruuu uuuu <sup>(1)</sup>
	873	874	876	877	0000 0000	0000 0000	uuuu uuuu <sup>(1)</sup>
PIR2	873	874	876	877	-r-0 0--0	-r-0 0--0	-r-u u--u <sup>(1)</sup>
TMR1L	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	873	874	876	877	--00 0000	--uu uuuu	--uu uuuu
TMR2	873	874	876	877	0000 0000	0000 0000	uuuu uuuu
T2CON	873	874	876	877	-000 0000	-000 0000	-uuu uuuu
SSPBUF	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPCON	873	874	876	877	0000 0000	0000 0000	uuuu uuuu
CCPR1L	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	873	874	876	877	--00 0000	--00 0000	--uu uuuu
RCSTA	873	874	876	877	0000 000x	0000 000x	uuuu uuuu
TXREG	873	874	876	877	0000 0000	0000 0000	uuuu uuuu
RCREG	873	874	876	877	0000 0000	0000 0000	uuuu uuuu
CCPR2L	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2H	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	873	874	876	877	0000 0000	0000 0000	uuuu uuuu
ADRESH	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	873	874	876	877	0000 00-0	0000 00-0	uuuu uu-u
OPTION_REG	873	874	876	877	1111 1111	1111 1111	uuuu uuuu
TRISA	873	874	876	877	--11 1111	--11 1111	--uu uuuu
TRISB	873	874	876	877	1111 1111	1111 1111	uuuu uuuu
TRISC	873	874	876	877	1111 1111	1111 1111	uuuu uuuu
TRISD	873	874	876	877	1111 1111	1111 1111	uuuu uuuu
TRISE	873	874	876	877	0000 -111	0000 -111	uuuu -uuu
PIE1	873	874	876	877	r000 0000	r000 0000	ruuu uuuu
	873	874	876	877	0000 0000	0000 0000	uuuu uuuu
PIE2	873	874	876	877	-r-0 0--0	-r-0 0--0	-r-u u--u

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

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

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

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

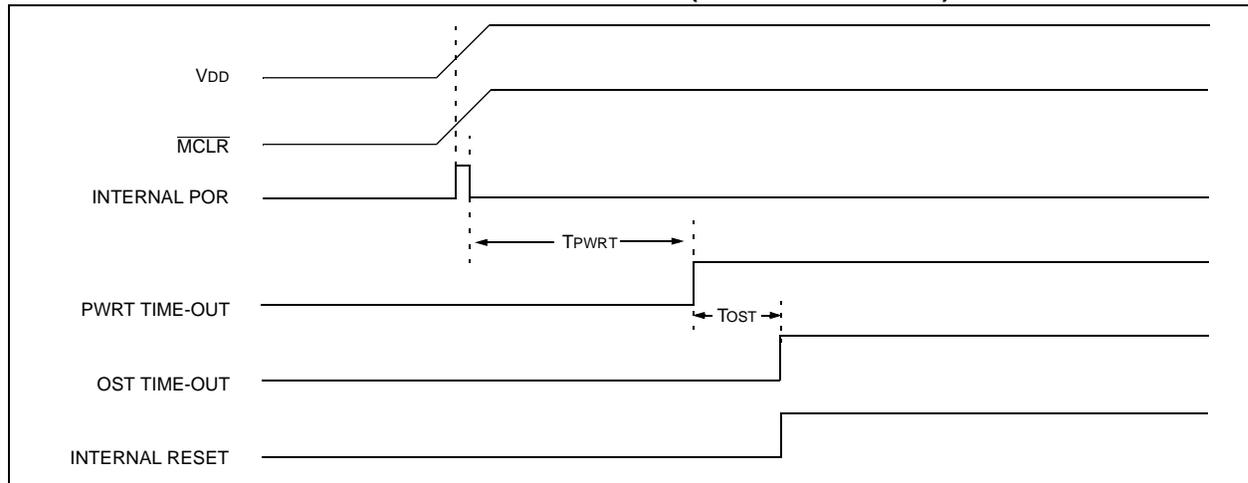
**TABLE 12-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)**

Register	Devices				Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset	Wake-up via WDT or Interrupt
PCON	873	874	876	877	---- --q <sub>q</sub>	---- --uu	---- --uu
PR2	873	874	876	877	1111 1111	1111 1111	1111 1111
SSPADD	873	874	876	877	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	873	874	876	877	--00 0000	--00 0000	--uu uuuu
TXSTA	873	874	876	877	0000 -010	0000 -010	uuuu -uuu
SPBRG	873	874	876	877	0000 0000	0000 0000	uuuu uuuu
ADRESL	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1	873	874	876	877	0--- 0000	0--- 0000	u--- uuuu
EEDATA	873	874	876	877	0--- 0000	0--- 0000	u--- uuuu
EEADR	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEDATH	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADRH	873	874	876	877	xxxx xxxx	uuuu uuuu	uuuu uuuu
EECON1	873	874	876	877	x--- x000	u--- u000	u--- uuuu
EECON2	873	874	876	877	---- ----	---- ----	---- ----

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

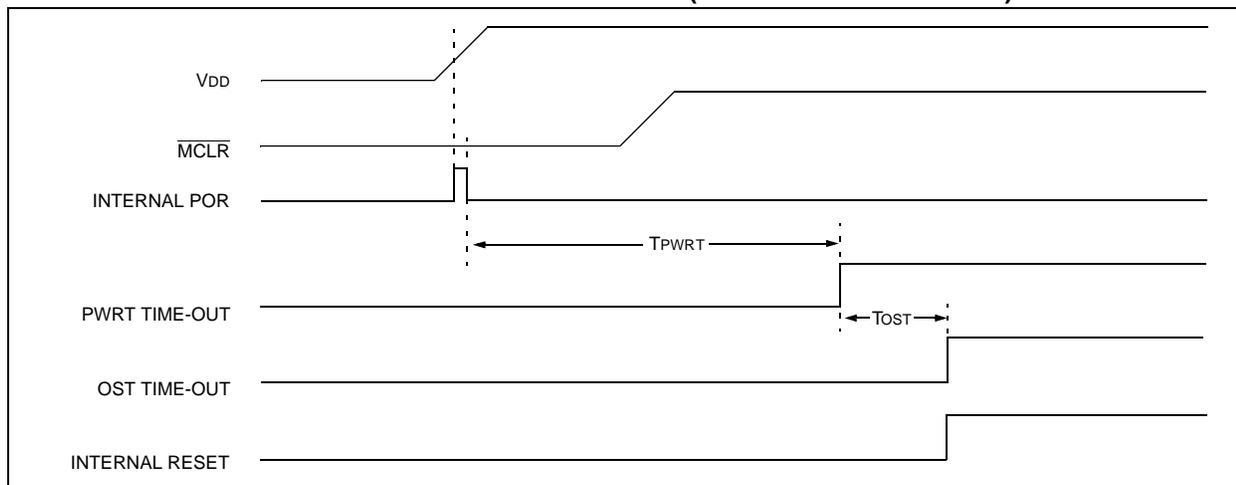
- Note 1:** One or more bits in INTCON, PIR1 and/or PIR2 will be affected (to cause wake-up).
- 2:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).
- 3:** See Table 12-5 for reset value for specific condition.

**FIGURE 12-5: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO V<sub>DD</sub>)**

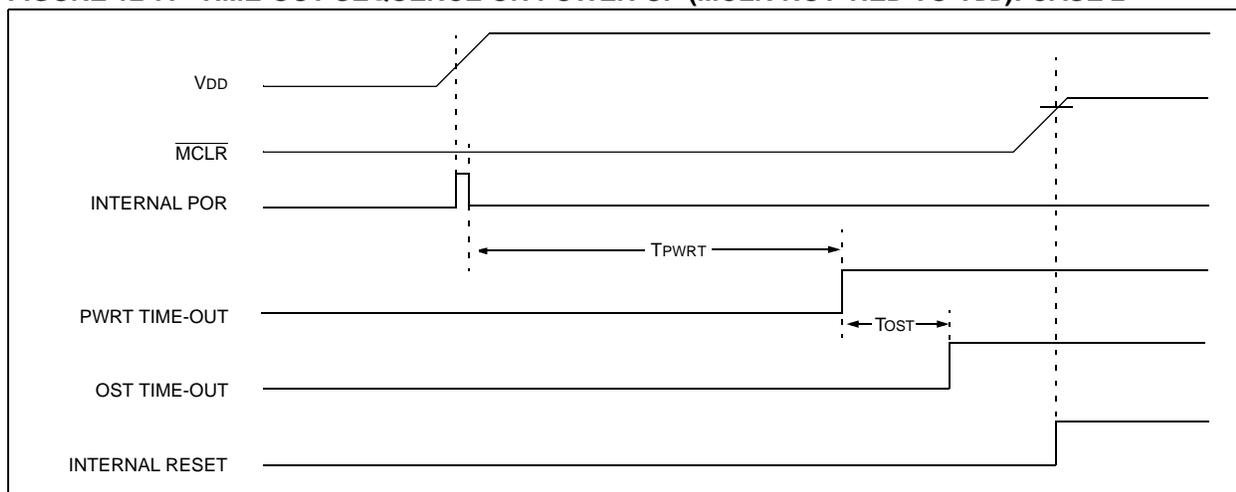


# PIC16F87X

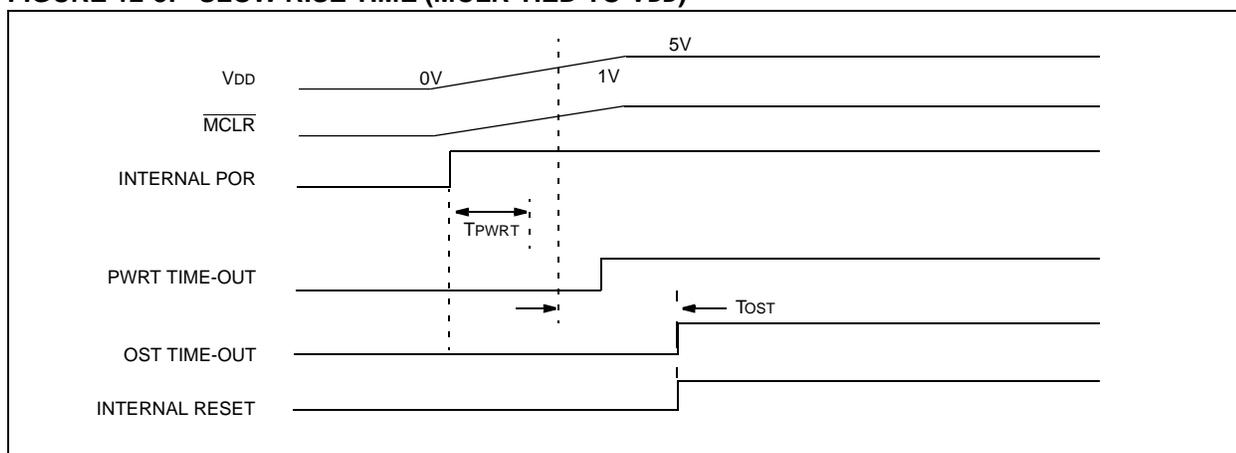
**FIGURE 12-6: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{\text{DD}}$ ): CASE 1**



**FIGURE 12-7: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{\text{DD}}$ ): CASE 2**



**FIGURE 12-8: SLOW RISE TIME ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$ )**



## 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 un-masked 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.

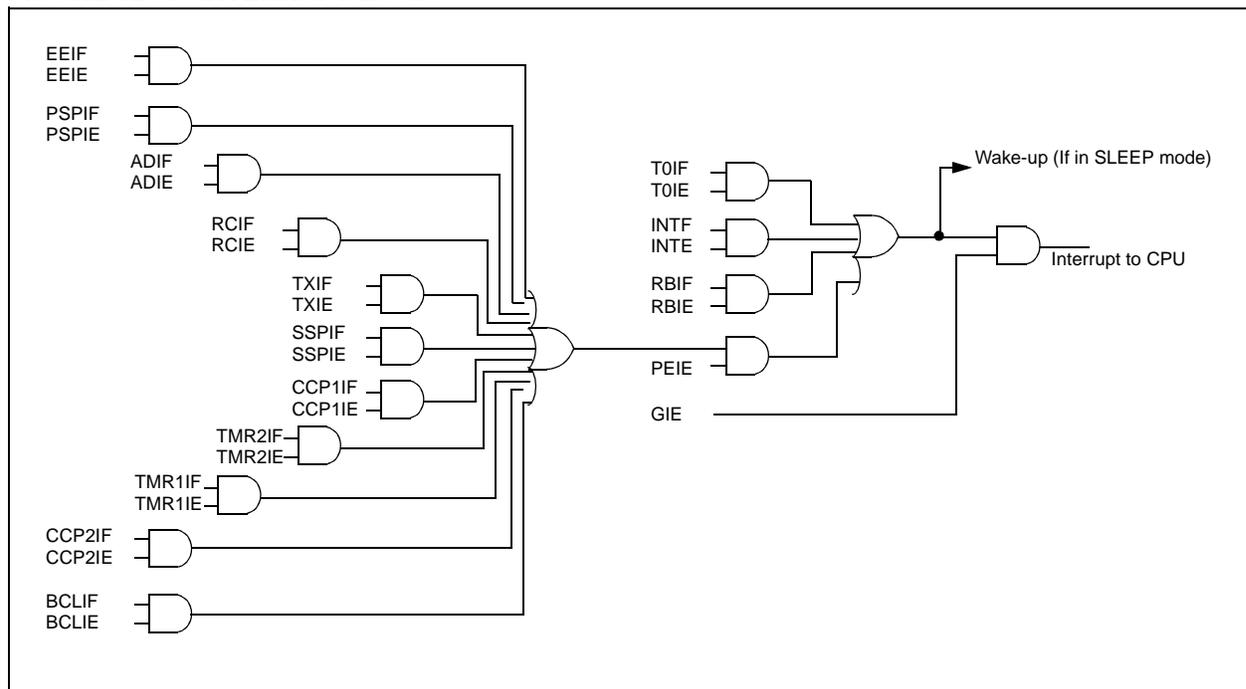
The RB0/INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

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 or the 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

## 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 → 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 basic code 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)
:
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
```

## 12.12 Watchdog Timer (WDT)

The Watchdog Timer is as a free running on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKIN pin. That means that the WDT will run, even if the clock on the OSC1/CLKIN and OSC2/CLKOUT pins of the device has been stopped, for example, by execution of a SLEEP instruction.

During normal operation, a WDT time-out generates a device RESET (Watchdog Timer Reset). If the device is in SLEEP mode, a WDT time-out causes the device to wake-up and continue with normal operation (Watchdog Timer Wake-up). The  $\overline{TO}$  bit in the STATUS register will be cleared upon a Watchdog Timer time-out.

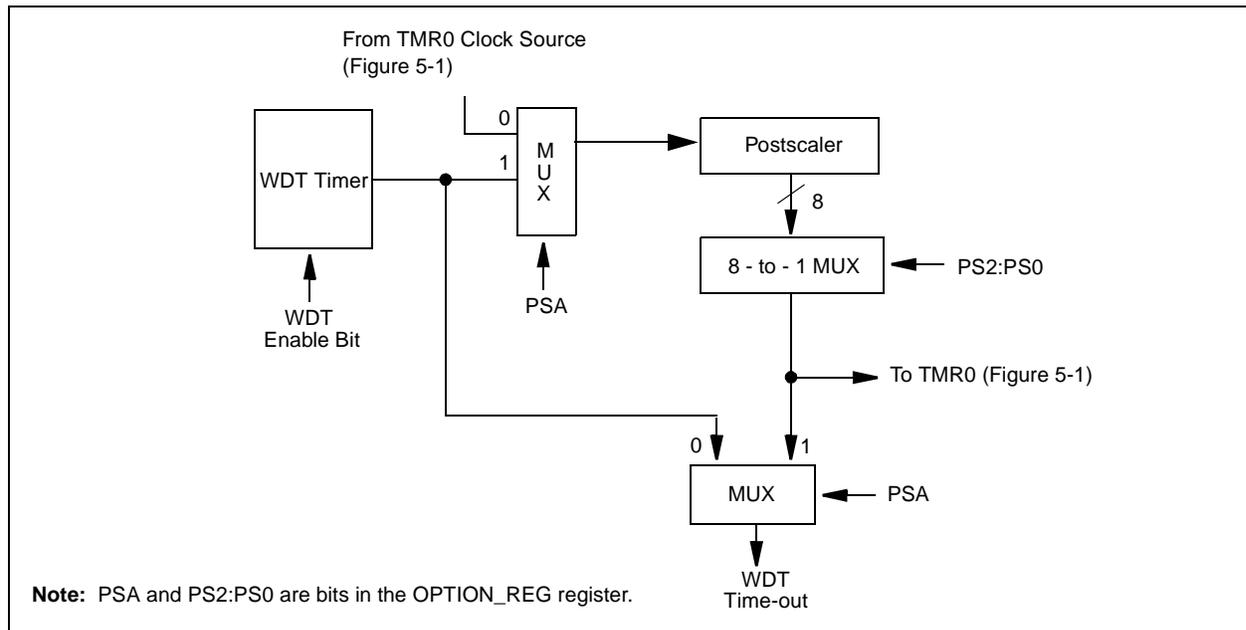
The WDT can be permanently disabled by clearing configuration bit WDTE (Section 12.1).

WDT time-out period values may be found in the Electrical Specifications section under parameter #31. Values for the WDT prescaler (actually a postscaler, but shared with the Timer0 prescaler) may be assigned using the OPTION\_REG register.

**Note:** The CLRWDT and SLEEP instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET condition.

**Note:** When a CLRWDT instruction is executed and the prescaler is assigned to the WDT, the prescaler count will be cleared, but the prescaler assignment is not changed.

**FIGURE 12-10: WATCHDOG TIMER BLOCK DIAGRAM**



**FIGURE 12-11: SUMMARY OF WATCHDOG TIMER REGISTERS**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2007h	Config. bits	(1)	BODEN <sup>(1)</sup>	CP1	CP0	PWRTE <sup>(1)</sup>	WDTE	FOSC1	FOSC0
81h,181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

Legend: Shaded cells are not used by the Watchdog Timer.

**Note 1:** See Register 12-1 for operation of these bits.

## 12.13 Power-down Mode (SLEEP)

Power-down mode is entered by executing a `SLEEP` instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the  $\overline{PD}$  bit (STATUS<3>) is cleared, the  $\overline{TO}$  (STATUS<4>) bit is set, and the oscillator driver is turned off. The I/O ports maintain the status they had before the `SLEEP` instruction was executed (driving high, low, or hi-impedance).

For lowest current consumption in this mode, place all I/O pins at either  $V_{DD}$  or  $V_{SS}$ , ensure no external circuitry is drawing current from the I/O pin, power-down the A/D and disable external clocks. Pull all I/O pins that are hi-impedance inputs, high or low externally, to avoid switching currents caused by floating inputs. The `T0CKI` input should also be at  $V_{DD}$  or  $V_{SS}$  for lowest current consumption. The contribution from on-chip pull-ups on `PORTB` should be considered.

The  $\overline{MCLR}$  pin must be at a logic high level ( $V_{IHMC}$ ).

### 12.13.1 WAKE-UP FROM SLEEP

The device can wake up from `SLEEP` through one of the following events:

1. External reset input on  $\overline{MCLR}$  pin.
2. Watchdog Timer wake-up (if `WDT` was enabled).
3. Interrupt from `INT` pin, `RB` port change or some Peripheral Interrupts.

External  $\overline{MCLR}$  Reset will cause a device reset. All other events are considered a continuation of program execution and cause a "wake-up". The  $\overline{TO}$  and  $\overline{PD}$  bits in the `STATUS` register can be used to determine the cause of device reset. The  $\overline{PD}$  bit, which is set on power-up, is cleared when `SLEEP` is invoked. The  $\overline{TO}$  bit is cleared if a `WDT` time-out occurred and caused wake-up.

The following peripheral interrupts can wake the device from `SLEEP`:

1. `PSP` read or write.
2. `TMR1` interrupt. `Timer1` must be operating as an asynchronous counter.
3. `CCP` capture mode interrupt.
4. Special event trigger (`Timer1` in asynchronous mode using an external clock).
5. `SSP` (Start/Stop) bit detect interrupt.
6. `SSP` transmit or receive in slave mode (`SPI/I2C`).
7. `USART` RX or TX (synchronous slave mode).
8. A/D conversion (when A/D clock source is `RC`).
9. `EEPROM` write operation completion

Other peripherals cannot generate interrupts since during `SLEEP`, no on-chip clocks are present.

When the `SLEEP` instruction is being executed, the next instruction (`PC + 1`) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding

interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the `GIE` bit. If the `GIE` bit is clear (disabled), the device continues execution at the instruction after the `SLEEP` instruction. If the `GIE` bit is set (enabled), the device executes the instruction after the `SLEEP` instruction and then branches to the interrupt address (0004h). In cases where the execution of the instruction following `SLEEP` is not desirable, the user should have a `NOP` after the `SLEEP` instruction.

### 12.13.2 WAKE-UP USING INTERRUPTS

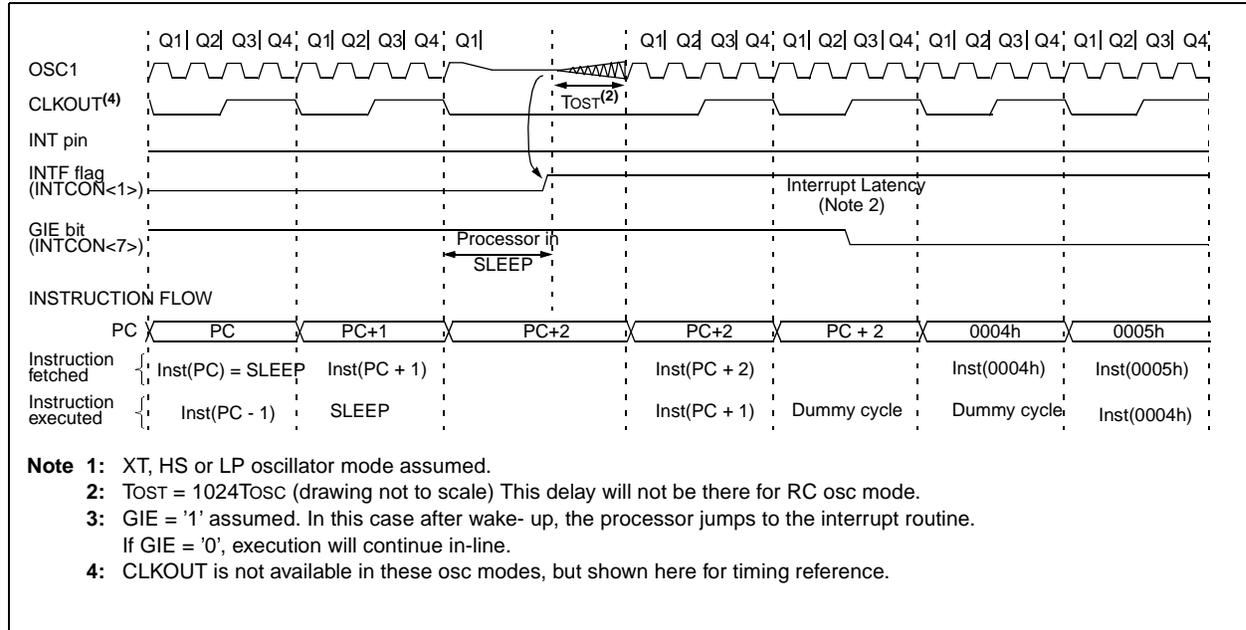
When global interrupts are disabled (`GIE` cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a `SLEEP` instruction, the `SLEEP` instruction will complete as a `NOP`. Therefore, the `WDT` and `WDT` postscaler will not be cleared, the  $\overline{TO}$  bit will not be set and  $\overline{PD}$  bits will not be cleared.
- If the interrupt occurs **during or after** the execution of a `SLEEP` instruction, the device will immediately wake up from sleep. The `SLEEP` instruction will be completely executed before the wake-up. Therefore, the `WDT` and `WDT` postscaler will be cleared, the  $\overline{TO}$  bit will be set and the `PD` bit will be cleared.

Even if the flag bits were checked before executing a `SLEEP` instruction, it may be possible for flag bits to become set before the `SLEEP` instruction completes. To determine whether a `SLEEP` instruction executed, test the  $\overline{PD}$  bit. If the  $\overline{PD}$  bit is set, the `SLEEP` instruction was executed as a `NOP`.

To ensure that the `WDT` is cleared, a `CLRWDT` instruction should be executed before a `SLEEP` instruction.

**FIGURE 12-12: WAKE-UP FROM SLEEP THROUGH INTERRUPT**



## 12.14 In-Circuit Debugger

When the DEBUG bit in the configuration word is programmed to a '0', the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB. When the microcontroller has this feature enabled, some of the resources are not available for general use. Table 12-7 shows which features are consumed by the background debugger.

**TABLE 12-7: DEBUGGER RESOURCES**

I/O pins	RB6, RB7
Stack	1 level
Program Memory	Address 0000h must be NOP
	Last 100h words
Data Memory	0x070(0x0F0, 0x170, 0x1F0) 0x1EB - 0x1EF

To use the In-Circuit Debugger function of the microcontroller, the design must implement In-Circuit Serial Programming connections to MCLR/VPP, VDD, GND, RB7 and RB6. This will interface to the In-Circuit Debugger module available from Microchip or one of the third party development tool companies.

## 12.15 Program Verification/Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out for verification purposes.

## 12.16 ID Locations

Four memory locations (2000h - 2003h) are designated as ID locations where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution but are readable and writable during program/verify. It is recommended that only the 4 least significant bits of the ID location are used.

## 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 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 In-Circuit Serial Programming (ICSP™) Guide, (DS30277B).

## 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 VDD source in the operating voltage range. This only means that VPP does not have to be brought to VIH, 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, VDD is applied to the MCLR pin. To enter programming mode, VDD 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 VIH to the 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.

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 VIH on MCLR. The LVP bit can only be changed when using high voltage on 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 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.

## 13.0 INSTRUCTION SET SUMMARY

Each PIC16CXX instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CXX instruction set summary in Table 13-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 13-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For **literal and control** operations, 'k' represents an eight or eleven bit constant or literal value.

**TABLE 13-1: OPCODE FIELD DESCRIPTIONS**

Field	Description
f	Register file address (0x00 to 0x7F)
w	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
PC	Program Counter
TO	Time-out bit
PD	Power-down bit

The instruction set is highly orthogonal and is grouped into three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction

execution time is 1  $\mu$ s. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2  $\mu$ s.

Table 13-2 lists the instructions recognized by the MPASM assembler.

Figure 13-1 shows the general formats that the instructions can have.

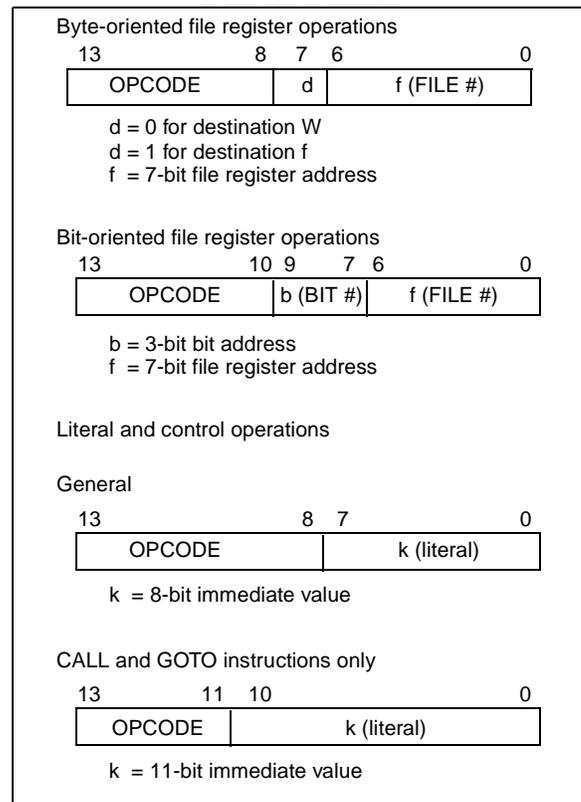
**Note:** To maintain upward compatibility with future PIC16CXX products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

**FIGURE 13-1: GENERAL FORMAT FOR INSTRUCTIONS**



A description of each instruction is available in the PICmicro™ Mid-Range Reference Manual, (DS33023).

# PIC16F87X

**TABLE 13-2: PIC16CXXX INSTRUCTION SET**

Mnemonic, Operands	Description	Cycles	14-Bit Opcode		Status Affected	Notes
			MSb	LSb		
<b>BYTE-ORIENTED FILE REGISTER OPERATIONS</b>						
ADDWF	f, d Add W and f	1	00	0111 dfff ffff	C,DC,Z	1,2
ANDWF	f, d AND W with f	1	00	0101 dfff ffff	Z	1,2
CLRF	f Clear f	1	00	0001 lfff ffff	Z	2
CLRWF	- Clear W	1	00	0001 0xxx xxxx	Z	
COMF	f, d Complement f	1	00	1001 dfff ffff	Z	1,2
DECF	f, d Decrement f	1	00	0011 dfff ffff	Z	1,2
DECFSZ	f, d Decrement f, Skip if 0	1(2)	00	1011 dfff ffff		1,2,3
INCF	f, d Increment f	1	00	1010 dfff ffff	Z	1,2
INCFSZ	f, d Increment f, Skip if 0	1(2)	00	1111 dfff ffff		1,2,3
IORWF	f, d Inclusive OR W with f	1	00	0100 dfff ffff	Z	1,2
MOVF	f, d Move f	1	00	1000 dfff ffff	Z	1,2
MOVWF	f Move W to f	1	00	0000 lfff ffff		
NOP	- No Operation	1	00	0000 0xx0 0000		
RLF	f, d Rotate Left f through Carry	1	00	1101 dfff ffff	C	1,2
RRF	f, d Rotate Right f through Carry	1	00	1100 dfff ffff	C	1,2
SUBWF	f, d Subtract W from f	1	00	0010 dfff ffff	C,DC,Z	1,2
SWAPF	f, d Swap nibbles in f	1	00	1110 dfff ffff		1,2
XORWF	f, d Exclusive OR W with f	1	00	0110 dfff ffff	Z	1,2
<b>BIT-ORIENTED FILE REGISTER OPERATIONS</b>						
BCF	f, b Bit Clear f	1	01	00bb bfff ffff		1,2
BSF	f, b Bit Set f	1	01	01bb bfff ffff		1,2
BTFSC	f, b Bit Test f, Skip if Clear	1 (2)	01	10bb bfff ffff		3
BTFSS	f, b Bit Test f, Skip if Set	1 (2)	01	11bb bfff ffff		3
<b>LITERAL AND CONTROL OPERATIONS</b>						
ADDLW	k Add literal and W	1	11	111x kkkk kkkk	C,DC,Z	
ANDLW	k AND literal with W	1	11	1001 kkkk kkkk	Z	
CALL	k Call subroutine	2	10	0kkk kkkk kkkk		
CLRWDT	- Clear Watchdog Timer	1	00	0000 0110 0100	$\overline{TO}, \overline{PD}$	
GOTO	k Go to address	2	10	1kkk kkkk kkkk		
IORLW	k Inclusive OR literal with W	1	11	1000 kkkk kkkk	Z	
MOVLW	k Move literal to W	1	11	00xx kkkk kkkk		
RETFIE	- Return from interrupt	2	00	0000 0000 1001		
RETLW	k Return with literal in W	2	11	01xx kkkk kkkk		
RETURN	- Return from Subroutine	2	00	0000 0000 1000		
SLEEP	- Go into standby mode	1	00	0000 0110 0011	$\overline{TO}, \overline{PD}$	
SUBLW	k Subtract W from literal	1	11	110x kkkk kkkk	C,DC,Z	
XORLW	k Exclusive OR literal with W	1	11	1010 kkkk kkkk	Z	

**Note 1:** When an I/O register is modified as a function of itself ( e.g., MOVF PORTB, 1), 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 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 to the Timer0 Module.

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

**Note:** Additional information on the mid-range instruction set is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

## 13.1 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.

### **ANDWF**      **AND W with f**

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

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

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

**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'.

### **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'.

### **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.

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

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

**Operands:**     $0 \leq k \leq 255$

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

**Status Affected:**    Z

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

### **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.

# PIC16F87X

<b>BTFS</b>	<b>Bit Test f, Skip if Set</b>
Syntax:	<code>[label] BTFS f,b</code>
Operands:	$0 \leq f \leq 127$ $0 \leq b < 7$
Operation:	skip if $(f \ll b) = 1$
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 2TCY instruction.

<b>BTFS</b>	<b>Bit Test, Skip if Clear</b>
Syntax:	<code>[label] BTFS f,b</code>
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	skip if $(f \ll b) = 0$
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 2TCY instruction.

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

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

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

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

<b>COMF</b>	<b>Complement f</b>
Syntax:	[ <i>label</i> ] COMF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(\bar{f}) \rightarrow (\text{destination})$
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'.

<b>GOTO</b>	<b>Unconditional Branch</b>
Syntax:	[ <i>label</i> ] GOTO k
Operands:	$0 \leq k \leq 2047$
Operation:	$k \rightarrow PC<10:0>$ $PCLATH<4:3> \rightarrow PC<12:11>$
Status Affected:	None
Description:	GOTO is an unconditional branch. The eleven bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two cycle instruction.

<b>DECF</b>	<b>Decrement f</b>
Syntax:	[ <i>label</i> ] DECF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) - 1 \rightarrow (\text{destination})$
Status Affected:	Z
Description:	Decrement 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'.

<b>INCF</b>	<b>Increment f</b>
Syntax:	[ <i>label</i> ] INCF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) + 1 \rightarrow (\text{destination})$
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.

<b>DECFSZ</b>	<b>Decrement f, Skip if 0</b>
Syntax:	[ <i>label</i> ] DECFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) - 1 \rightarrow (\text{destination});$ skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, then a NOP is executed instead making it a 2TCY instruction.

<b>INCFSZ</b>	<b>Increment f, Skip if 0</b>
Syntax:	[ <i>label</i> ] INCFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) + 1 \rightarrow (\text{destination}),$ skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, a NOP is executed instead making it a 2TCY instruction.

# PIC16F87X

---

**IORLW**            **Inclusive OR Literal with W**

---

Syntax:            [ *label* ] IORLW k

Operands:         $0 \leq k \leq 255$

Operation:        (W) .OR. k  $\rightarrow$  (W)

Status Affected: Z

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

**IORWF**            **Inclusive OR W with f**

---

Syntax:            [ *label* ] IORWF f,d

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

Operation:        (W) .OR. (f)  $\rightarrow$  (destination)

Status Affected: Z

Description:      Inclusive OR the W register with register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

**MOVF**             **Move f**

---

Syntax:            [ *label* ] MOVF f,d

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

Operation:        (f)  $\rightarrow$  (destination)

Status Affected: Z

Description:      The contents of register f are moved to a destination dependant upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.

**MOVLW**            **Move Literal to W**

---

Syntax:            [ *label* ] MOVLW k

Operands:         $0 \leq k \leq 255$

Operation:        k  $\rightarrow$  (W)

Status Affected: None

Description:      The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.

**MOVWF**            **Move W to f**

---

Syntax:            [ *label* ] MOVWF f

Operands:         $0 \leq f \leq 127$

Operation:        (W)  $\rightarrow$  (f)

Status Affected: None

Description:      Move data from W register to register 'f'.

**NOP**                **No Operation**

---

Syntax:            [ *label* ] NOP

Operands:        None

Operation:        No operation

Status Affected: None

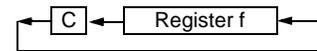
Description:      No operation.

## **RETFIE**      **Return from Interrupt**

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

## **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'.



## **RETLW**        **Return with Literal in W**

Syntax:      [ *label* ] RETLW k  
 Operands:     $0 \leq k \leq 255$   
 Operation:     $k \rightarrow (W)$ ;  
                   TOS → PC  
 Status Affected: None  
 Description:    The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two cycle instruction.

## **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'.



## **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.

## **SLEEP**

Syntax:      [ *label*    SLEEP  
                   ] ]  
 Operands:    None  
 Operation:    00h → WDT,  
                   0 → WDT prescaler,  
                   1 →  $\overline{TO}$ ,  
                   0 →  $\overline{PD}$   
 Status Affected:  $\overline{TO}$ ,  $\overline{PD}$   
 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.

# PIC16F87X

---

## **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.

## **XORLW**      **Exclusive OR Literal with W**

---

Syntax:      *[label]*    XORLW   k

Operands:     $0 \leq k \leq 255$

Operation:     $(W) .XOR. k \rightarrow (W)$

Status Affected:    Z

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

## **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'.

## **XORWF**      **Exclusive OR W with f**

---

Syntax:      *[label]*    XORWF   f,d

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

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

Status Affected:    Z

Description:    Exclusive OR 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'.

## **SWAPF**      **Swap Nibbles in f**

---

Syntax:      *[label]*    SWAPF   f,d

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

Operation:     $(f<3:0>) \rightarrow (\text{destination}<7:4>),$   
 $(f<7:4>) \rightarrow (\text{destination}<3:0>)$

Status Affected:    None

Description:    The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in W register. If 'd' is 1, the result is placed in register 'f'.

## 14.0 DEVELOPMENT SUPPORT

The PICmicro<sup>®</sup> microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB<sup>®</sup> IDE Software
- Assemblers/Compilers/Linkers
  - MPASM Assembler
  - MPLAB-C17 and MPLAB-C18 C Compilers
  - MPLINK/MPLIB Linker/Librarian
- Simulators
  - MPLAB-SIM Software Simulator
- Emulators
  - MPLAB-ICE Real-Time In-Circuit Emulator
  - PICMASTER<sup>®</sup>/PICMASTER-CE In-Circuit Emulator
  - ICEPIC<sup>™</sup>
- In-Circuit Debugger
  - MPLAB-ICD for PIC16F877
- Device Programmers
  - PRO MATE<sup>®</sup> II Universal Programmer
  - PICSTART<sup>®</sup> Plus Entry-Level Prototype Programmer
- Low-Cost Demonstration Boards
  - SIMICE
  - PICDEM-1
  - PICDEM-2
  - PICDEM-3
  - PICDEM-17
  - SEEVAL<sup>®</sup>
  - KEELOQ<sup>®</sup>

### 14.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8-bit microcontroller market. MPLAB is a Windows<sup>®</sup>-based application which contains:

- Multiple functionality
  - editor
  - simulator
  - programmer (sold separately)
  - emulator (sold separately)
- A full featured editor
- A project manager
- Customizable tool bar and key mapping
- A status bar
- On-line help

MPLAB allows you to:

- Edit your source files (either assembly or 'C')
- One touch assemble (or compile) and download to PICmicro tools (automatically updates all project information)
- Debug using:
  - source files
  - absolute listing file
  - object code

The ability to use MPLAB with Microchip's simulator, MPLAB-SIM, allows a consistent platform and the ability to easily switch from the cost-effective simulator to the full featured emulator with minimal retraining.

### 14.2 MPASM Assembler

MPASM is a full featured universal macro assembler for all PICmicro MCU's. It can produce absolute code directly in the form of HEX files for device programmers, or it can generate relocatable objects for MPLINK.

MPASM has a command line interface and a Windows shell and can be used as a standalone application on a Windows 3.x or greater system. MPASM generates relocatable object files, Intel standard HEX files, MAP files to detail memory usage and symbol reference, an absolute LST file which contains source lines and generated machine code, and a COD file for MPLAB debugging.

MPASM features include:

- MPASM and MPLINK are integrated into MPLAB projects.
- MPASM allows user defined macros to be created for streamlined assembly.
- MPASM allows conditional assembly for multi purpose source files.
- MPASM directives allow complete control over the assembly process.

### 14.3 MPLAB-C17 and MPLAB-C18 C Compilers

The MPLAB-C17 and MPLAB-C18 Code Development Systems are complete ANSI 'C' compilers and integrated development environments for Microchip's PIC17CXXX and PIC18CXXX family of microcontrollers, respectively. These compilers provide powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compilers provide symbol information that is compatible with the MPLAB IDE memory display.

## 14.4 MPLINK/MPLIB Linker/Librarian

MPLINK is a relocatable linker for MPASM and MPLAB-C17 and MPLAB-C18. It can link relocatable objects from assembly or C source files along with pre-compiled libraries using directives from a linker script.

MPLIB is a librarian for pre-compiled code to be used with MPLINK. When a routine from a library is called from another source file, only the modules that contains that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications. MPLIB manages the creation and modification of library files.

MPLINK features include:

- MPLINK works with MPASM and MPLAB-C17 and MPLAB-C18.
- MPLINK allows all memory areas to be defined as sections to provide link-time flexibility.

MPLIB features include:

- MPLIB makes linking easier because single libraries can be included instead of many smaller files.
- MPLIB helps keep code maintainable by grouping related modules together.
- MPLIB commands allow 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 host environment by simulating the PICmicro 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.

MPLAB-SIM fully supports symbolic debugging using MPLAB-C17 and MPLAB-C18 and MPASM. 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 PICmicro microcontrollers (MCUs). Software control of MPLAB-ICE is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, "make" and download, and source debugging from a single environment.

Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB-ICE allows expansion to support new PICmicro microcontrollers.

The MPLAB-ICE 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 3.x/95/98 environment were chosen to best make these features available to you, the end user.

MPLAB-ICE 2000 is a full-featured emulator system with enhanced trace, trigger, and data monitoring features. Both systems use the same processor modules and will operate across the full operating speed range of the PICmicro MCU.

## 14.7 PICMASTER/PICMASTER CE

The PICMASTER system from Microchip Technology is a full-featured, professional quality emulator system. This flexible in-circuit emulator provides a high-quality, universal platform for emulating Microchip 8-bit PICmicro microcontrollers (MCUs). PICMASTER systems are sold worldwide, with a CE compliant model available for European Union (EU) countries.

## 14.8 ICEPIC

ICEPIC is a low-cost in-circuit emulation 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.

## 14.9 MPLAB-ICD In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB-ICD, is a powerful, low-cost run-time development tool. This tool is based on the flash PIC16F877 and can be used to develop for this and other PICmicro microcontrollers from the PIC16CXXX family. MPLAB-ICD utilizes the In-Circuit Debugging capability built into the PIC16F87X. This feature, along with Microchip's In-Circuit Serial Programming protocol, offers cost-effective in-circuit flash programming and debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by watching variables, single-stepping and setting break points. Running at full speed enables testing hardware in real-time. The MPLAB-ICD is also a programmer for the flash PIC16F87X family.

## **14.10 PRO MATE II Universal Programmer**

The PRO MATE II Universal Programmer is a full-featured programmer capable of operating in stand-alone mode as well as PC-hosted mode. PRO MATE II is CE compliant.

The PRO MATE II has programmable VDD and VPP supplies which allows it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for instructions and error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode the PRO MATE II can read, verify or program PICmicro devices. It can also set code-protect bits in this mode.

## **14.11 PICSTART Plus Entry Level Development System**

The PICSTART programmer is an easy-to-use, low-cost prototype programmer. It connects to the PC via one of the COM (RS-232) ports. MPLAB Integrated Development Environment software makes using the programmer simple and efficient.

PICSTART Plus supports all PICmicro devices with up to 40 pins. Larger pin count devices such as the PIC16C92X, and PIC17C76X may be supported with an adapter socket. PICSTART Plus is CE compliant.

## **14.12 SIMICE Entry-Level Hardware Simulator**

SIMICE is an entry-level hardware development system designed to operate in a PC-based environment with Microchip's simulator MPLAB-SIM. Both SIMICE and MPLAB-SIM run under Microchip Technology's MPLAB Integrated Development Environment (IDE) software. Specifically, SIMICE provides hardware simulation for Microchip's PIC12C5XX, PIC12CE5XX, and PIC16C5X families of PICmicro 8-bit microcontrollers. SIMICE works in conjunction with MPLAB-SIM to provide non-real-time I/O port emulation. SIMICE enables a developer to run simulator code for driving the target system. In addition, the target system can provide input to the simulator code. This capability allows for simple and interactive debugging without having to manually generate MPLAB-SIM stimulus files. SIMICE is a valuable debugging tool for entry-level system development.

## **14.13 PICDEM-1 Low-Cost PICmicro Demonstration Board**

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The users can program the sample microcontrollers provided with

the PICDEM-1 board, on a PRO MATE II or PICSTART-Plus programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the MPLAB-ICE emulator and download the firmware to the emulator for testing. Additional prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push-button switches and eight LEDs connected to PORTB.

## **14.14 PICDEM-2 Low-Cost PIC16CXX Demonstration Board**

The PICDEM-2 is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE II programmer or PICSTART-Plus, and easily test firmware. The MPLAB-ICE emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I<sup>2</sup>C bus and separate headers for connection to an LCD module and a keypad.

## **14.15 PICDEM-3 Low-Cost PIC16CXXX Demonstration Board**

The PICDEM-3 is a simple demonstration board that supports the PIC16C923 and PIC16C924 in the PLCC package. It will also support future 44-pin PLCC microcontrollers with a LCD Module. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-3 board, on a PRO MATE II programmer or PICSTART Plus with an adapter socket, and easily test firmware. The MPLAB-ICE emulator may also be used with the PICDEM-3 board to test firmware. Additional prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features include an RS-232 interface, push-button switches, a potentiometer for simulated analog input, a thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM-3 board is an LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM-3 provides an additional RS-232 interface and Windows 3.1 software for showing the demultiplexed LCD signals on a PC. A simple serial interface allows the user to construct a hardware demultiplexer for the LCD signals.

## **14.16 PICDEM-17**

The PICDEM-17 is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756, PIC17C762, and PIC17C766. All necessary hardware is included to run basic demo programs, which are supplied on a 3.5-inch disk. A programmed sample is included, and the user may erase it and program it with the other sample programs using the PRO MATE II or PICSTART Plus device programmers and easily debug and test the sample code. In addition, PICDEM-17 supports down-loading of programs to and executing out of external FLASH memory on board. The PICDEM-17 is also usable with the MPLAB-ICE or PICMASTER emulator, and all of the sample programs can be run and modified using either emulator. Additionally, a generous prototype area is available for user hardware.

## **14.17 SEEVAL Evaluation and Programming System**

The SEEVAL SEEPROM Designer's Kit supports all Microchip 2-wire and 3-wire Serial EEPROMs. The kit includes everything necessary to read, write, erase or program special features of any Microchip SEEPROM product including Smart Serials™ and secure serials. The Total Endurance™ Disk is included to aid in trade-off analysis and reliability calculations. The total kit can significantly reduce time-to-market and result in an optimized system.

## **14.18 KEELOQ Evaluation and Programming Tools**

KEELOQ evaluation and programming tools support Microchips HCS Secure Data Products. The HCS evaluation kit includes an LCD display to show changing codes, a decoder to decode transmissions, and a programming interface to program test transmitters.

**TABLE 14-1: DEVELOPMENT TOOLS FROM MICROCHIP**

Tool	PIC12CXX	PIC1400	PIC16C5X	PIC16C6X	PIC16CXX	PIC16F62X	PIC16C7X	PIC16C7XX	PIC16C8X	PIC16F8XX	PIC16C9XX	PIC17C4X	PIC17C7XX	PIC18CXX2	24CX/25CX/93CX	HCSXX	MCRFXXX	MCP2510
<b>Software Tools</b>																		
MPLAB® Integrated Development Environment	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MPLAB® C17 Compiler																		
MPLAB® C18 Compiler																		
MPASM/MPLINK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MPLAB®-ICE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PICMASTER/PICMASTER-CE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ICEPIC™ Low-Cost In-Circuit Emulator	✓		✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<b>Emulators</b>																		
MPLAB®-ICD In-Circuit Debugger				✓*			✓*			✓								
<b>Debuggers</b>																		
PICSTART® Plus Low-Cost Universal Dev. Kit	✓	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PRO MATE® II Universal Programmer	✓	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<b>Programmers</b>																		
SIMICE	✓		✓															
PICDEM-1			✓		✓		†		✓									
PICDEM-2							†							✓				
PICDEM-3																		
PICDEM-14A											✓							
PICDEM-17													✓					
<b>Demo Boards and Eval Kits</b>																		
KEELOQ® Evaluation Kit																		
KEELOQ Transponder Kit																		
microID™ Programmer's Kit																		
125 kHz microID Developer's Kit																		
125 kHz Anticollision microID Developer's Kit																		
13.56 MHz Anticollision microID Developer's Kit																		
MCP2510 CAN Developer's Kit																		
MCP2510 CAN Developer's Kit																		✓

\* Contact the Microchip Technology Inc. web site at [www.microchip.com](http://www.microchip.com) for information on how to use the MPLAB-ICD In-Circuit Debugger (DV164001) with PIC16C62, 63, 64, 65, 72, 73, 74, 76, 77

\*\* Contact Microchip Technology Inc. for availability date.

† Development tool is available on select devices.

# PIC16F87X

---

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, $\overline{\text{MCLR}}$ , and RA4).....	-0.3V to (VDD + 0.3V)
Voltage on VDD with respect to VSS .....	-0.3 to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to VSS (Note 2).....	0 to +14V
Voltage on RA4 with respect to Vss .....	0 to +8.5V
Total power dissipation (Note 1).....	1.0W
Maximum current out of VSS pin .....	300 mA
Maximum current into VDD pin .....	250 mA
Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > VDD).....	± 20 mA
Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > 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:  $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$

**2:** Voltage spikes below VSS at the  $\overline{\text{MCLR}}$  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  $\overline{\text{MCLR}}$  pin, rather than pulling this pin directly to VSS.

**3:** PORTD and PORTE are not implemented on the 28-pin 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.

# PIC16F87X

FIGURE 15-1: PIC16FXXX-20 VOLTAGE-FREQUENCY GRAPH

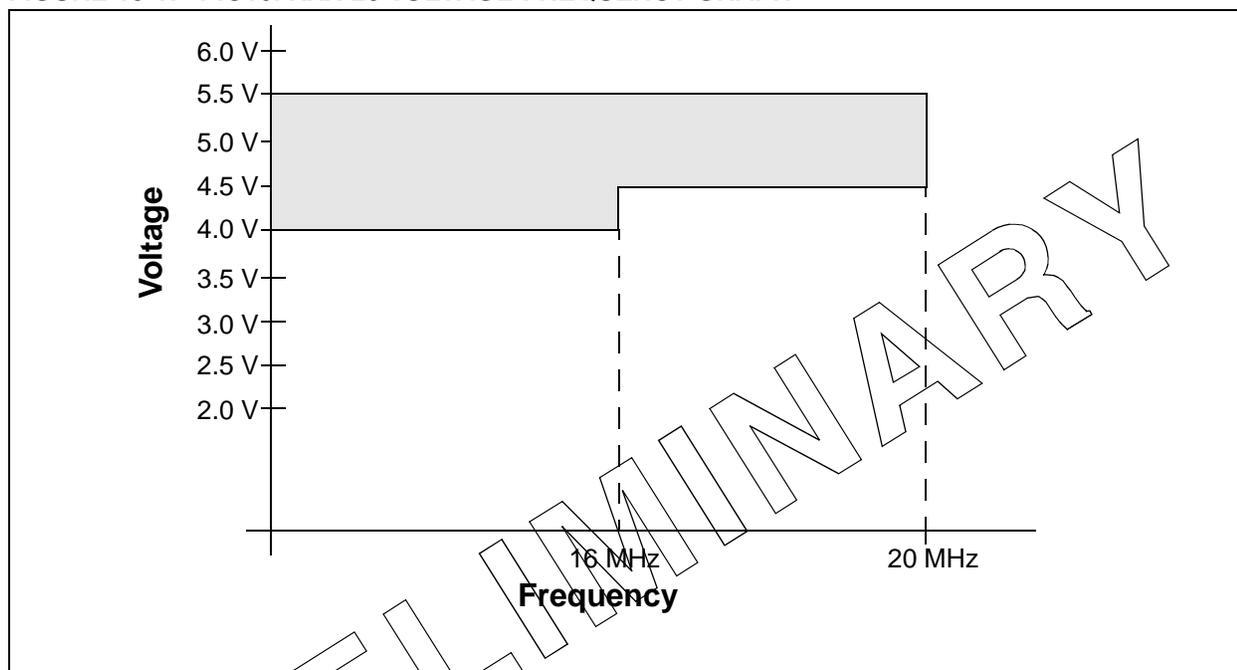


FIGURE 15-2: PIC16LFXXX-04 VOLTAGE-FREQUENCY GRAPH

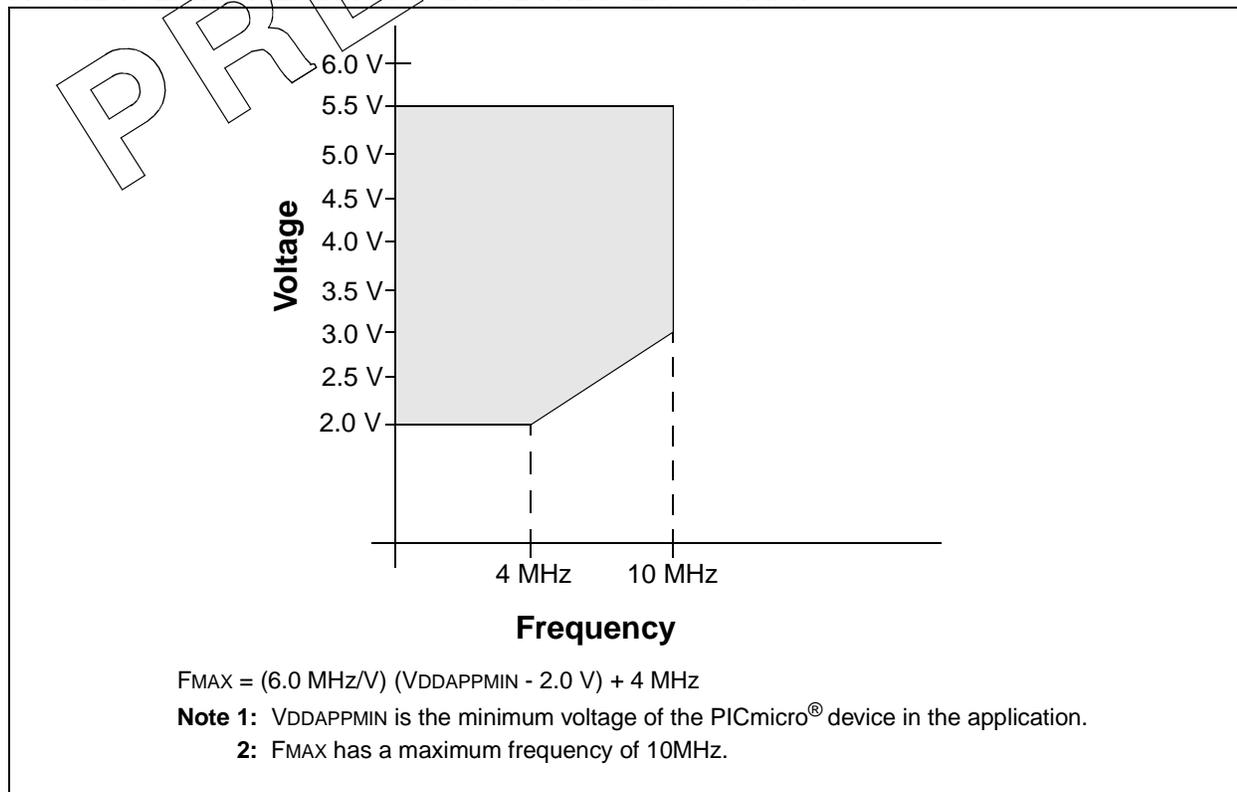
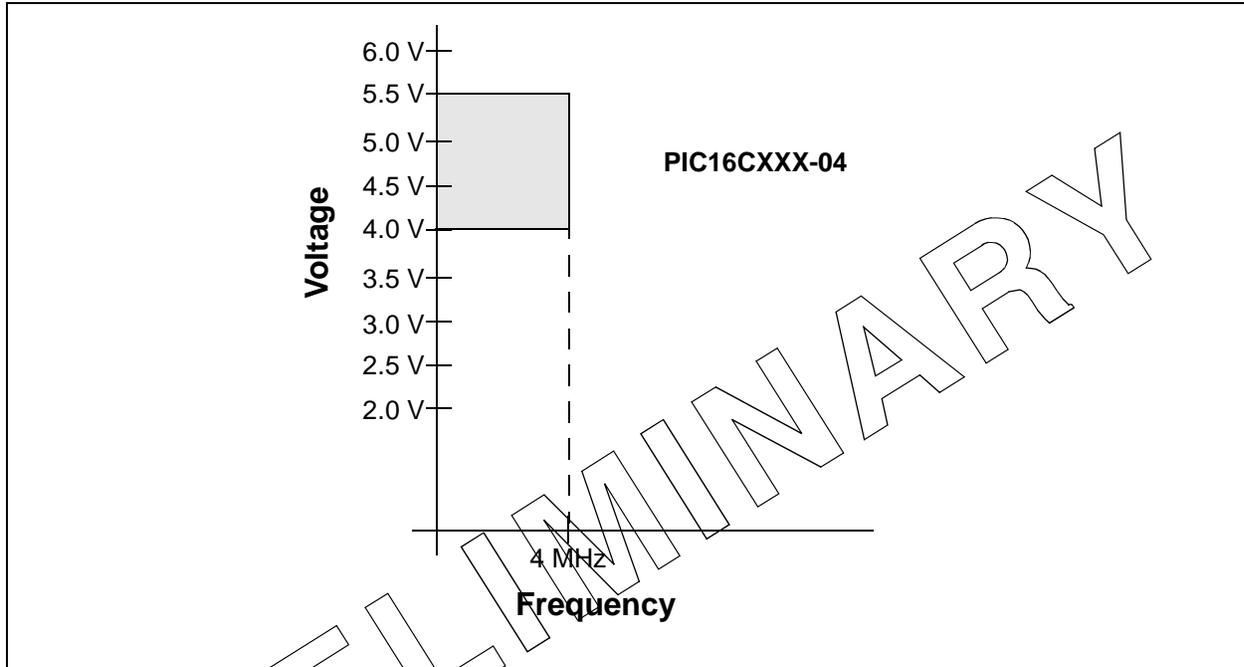


FIGURE 15-3: PIC16FXXX-04 VOLTAGE-FREQUENCY GRAPH



PRELIMINARY

# PIC16F87X

## 15.1 DC Characteristics: **PIC16F873/874/876/877-04 (Commercial, Industrial)** **PIC16F873/874/876/877-20 (Commercial, Industrial)**

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D001 D001A	Supply Voltage	VDD	4.0 4.5 VBOR*	- - -	5.5 5.5 5.5	V V V	XT, RC and LP osc configuration HS osc configuration BOR enabled, Fmax = 14MHz (Note 7)
D002*	RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	
D003	VDD start voltage to ensure internal Power-on Reset signal	VPOR	-	VSS	-	V	See section on Power-on Reset for details
D004*	VDD rise rate to ensure internal Power-on Reset signal	SVDD	0.05	-	-	V/ms	See section on Power-on Reset for details
D005	Brown-out Reset Voltage	VBOR	3.7	4.0	4.35	V	BODEN bit in configuration word enabled
D010 D013	Supply Current (Note 2,3)	IDD	-	1.6 7	4 15	mA mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 5.5V (Note 4) HS osc configuration FOSC = 20 MHz, VDD = 5.5V
D015*	Brown-out Reset Current (Note 6)	$\Delta$ IBOR	-	85	200	$\mu\text{A}$	BOR enabled VDD = 5.0V
D020 D021 D021A	Power-down Current (Note 3,5)	IPD	-	10.5 1.5 1.5	42 16 19	$\mu\text{A}$ $\mu\text{A}$ $\mu\text{A}$	VDD = 4.0V, WDT enabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ VDD = 4.0V, WDT disabled, $-0^{\circ}\text{C}$ to $+70^{\circ}\text{C}$ VDD = 4.0V, WDT disabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D023*	Brown-out Reset Current (Note 6)	$\Delta$ IBOR	-	85	200	$\mu\text{A}$	BOR enabled VDD = 5.0V

Legend: \* 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.

**Note 1:** This is the limit to which VDD can be lowered without losing RAM data.

**2:** The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, 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 tristated, pulled to VDD

MCLR = VDD; WDT enabled/disabled as specified.

**3:** 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 hi-impedance state and tied to VDD and VSS.

**4:** For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula  $I_r = V_{DD}/2R_{ext}$  (mA) with Rext in kOhm.

**5:** Timer1 oscillator (when enabled) adds approximately 20  $\mu\text{A}$  to the specification. This value is from characterization and is for design guidance only. This is not tested.

**6:** The  $\Delta$  current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

**7:** When BOR is enabled, the device will operate correctly until the VBOR voltage trip point is reached.

## 15.2 DC Characteristics: PIC16LF873/874/876/877-04 (Commercial, Industrial)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D001	Supply Voltage	VDD	2.0	-	5.5	V	LP, XT, RC osc configuration (DC, 4 MHz)
D002*	RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	
D003	VDD start voltage to ensure internal Power-on Reset signal	VPOR	-	VSS	-	V	See section on Power-on Reset for details
D004*	VDD rise rate to ensure internal Power-on Reset signal	SVDD	0.05	-	-	V/ms	See section on Power-on Reset for details
D005	Brown-out Reset Voltage	VBOR	3.7	4.0	4.35	V	BODEN bit in configuration word enabled
D010	Supply Current (Note 2,5)	IDD	-	0.6	2.0	mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 3.0V (Note 4)
D010A			-	20	35	$\mu\text{A}$	LP osc configuration FOSC = 32 kHz, VDD = 3.0V, WDT disabled
D015*	Brown-out Reset Current (Note 6)	$\Delta\text{IBOR}$	-	85	200	$\mu\text{A}$	BOR enabled VDD = 5.0V
D020	Power-down Current (Note 3,5)	IPD	-	7.5	30	$\mu\text{A}$	VDD = 3.0V, WDT enabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D021			-	0.9	5	$\mu\text{A}$	VDD = 3.0V, WDT disabled, $0^{\circ}\text{C}$ to $+70^{\circ}\text{C}$
D021A			-	0.9	5	$\mu\text{A}$	VDD = 3.0V, WDT disabled, $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D023*	Brown-out Reset Current (Note 6)	$\Delta\text{IBOR}$	-	85	200	$\mu\text{A}$	BOR enabled VDD = 5.0V

Legend: \* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V,  $25^{\circ}\text{C}$  unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** This is the limit to which VDD can be lowered without losing RAM data.

**2:** The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, 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 tristated, pulled to VDD

MCLR = VDD; WDT enabled/disabled as specified.

**3:** 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 hi-impedance state and tied to VDD and VSS.

**4:** For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula  $I_r = V_{DD}/2R_{ext}$  (mA) with Rext in kOhm.

**5:** Timer1 oscillator (when enabled) adds approximately  $20 \mu\text{A}$  to the specification. This value is from characterization and is for design guidance only. This is not tested.

**6:** The  $\Delta$  current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

# PIC16F87X

**15.3 DC Characteristics:** **PIC16F873/874/876/877-04 (Commercial, Industrial)**  
**PIC16F873/874/876/877-20 (Commercial, Industrial)**  
**PIC16LF873/874/876/877-04 (Commercial, Industrial)**

Standard Operating Conditions (unless otherwise stated)							
DC CHARACTERISTICS							
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial							
Operating voltage $V_{DD}$ range as described in DC spec Section 15.1 and Section 15.2.							
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D030 D030A D031 D032 D033	<b>Input Low Voltage</b> I/O ports with TTL buffer with Schmitt Trigger buffer MCLR, OSC1 (in RC mode) OSC1 (in XT, HS and LP)	$V_{IL}$	$V_{SS}$	-	$0.15V_{DD}$ $0.8V$ $0.2V_{DD}$ $0.2V_{DD}$	V	For entire $V_{DD}$ range $4.5V \leq V_{DD} \leq 5.5V$
D034 D034A	Ports RC3 and RC4 with Schmitt Trigger buffer with SMBus		$V_{SS}$ $-0.5$	- -	$0.3V_{DD}$ $0.6$	V	For entire $V_{DD}$ range for $V_{DD} = 4.5$ to $5.5V$
D040 D040A D041 D042 D042A D043 D044 D044A	<b>Input High Voltage</b> I/O ports with TTL buffer with Schmitt Trigger buffer MCLR OSC1 (XT, HS and LP) OSC1 (in RC mode) Ports RC3 and RC4 with Schmitt Trigger buffer with SMBus	$V_{IH}$	$2.0$ $0.25V_{DD} + 0.8V$ $0.8V_{DD}$ $0.8V_{DD}$ $0.7V_{DD}$ $0.9V_{DD}$ $0.7V_{DD}$ $1.4$	- - - - - - - -	$V_{DD}$ $V_{DD}$ $V_{DD}$ $V_{DD}$ $V_{DD}$ $V_{DD}$ $V_{DD}$ $5.5$	V	$4.5V \leq V_{DD} \leq 5.5V$ For entire $V_{DD}$ range For entire $V_{DD}$ range Note1 For entire $V_{DD}$ range for $V_{DD} = 4.5$ to $5.5V$
D070	PORTB weak pull-up current	$I_{PURB}$	50	250	400	$\mu\text{A}$	$V_{DD} = 5V, V_{PIN} = V_{SS}$
D060 D061 D063	<b>Input Leakage Current</b> (Notes 2, 3) I/O ports MCLR, RA4/T0CKI OSC1	$I_{IL}$	-	-	$\pm 1$ $\pm 5$ $\pm 5$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , Pin at hi-impedance $V_{SS} \leq V_{PIN} \leq V_{DD}$ $V_{SS} \leq V_{PIN} \leq V_{DD}$ , XT, HS and LP osc configuration
D080 D083	<b>Output Low Voltage</b> I/O ports OSC2/CLKOUT (RC osc config)	$V_{OL}$	-	-	$0.6$ $0.6$	V	$I_{OL} = 8.5 \text{ mA}, V_{DD} = 4.5V, -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ $I_{OL} = 1.6 \text{ mA}, V_{DD} = 4.5V, -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$

Legend: \* 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.

- Note 1:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16F87X be driven with external clock in RC mode.
- 2:** The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3:** Negative current is defined as current sourced by the pin.

Standard Operating Conditions (unless otherwise stated)							
DC CHARACTERISTICS							
		Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial					
		Operating voltage $V_{DD}$ range as described in DC spec Section 15.1 and Section 15.2.					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D090	<b>Output High Voltage</b> I/O ports (Note 3)	$V_{OH}$	$V_{DD} - 0.7$	-	-	V	$I_{OH} = -3.0\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D092	OSC2/CLKOUT (RC osc config)		$V_{DD} - 0.7$	-	-	V	
D150*	<b>Open-Drain High Voltage</b>	$V_{OD}$	-	-	8.5	V	RA4 pin
<b>Capacitive Loading Specs on Output Pins</b>							
D100	OSC2 pin	$C_{OSC2}$	-	-	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.
D101	All I/O pins and OSC2 (in RC mode) SCL, SDA in I <sup>2</sup> C mode	$C_{IO}$	-	-	50	pF	
D102		$C_B$	-	-	400	pF	
<b>Data EEPROM Memory</b>							
D120	Endurance	$E_D$	100K	-	-	E/W	25°C at 5V
D121	$V_{DD}$ for read/write	$V_{DRW}$	$V_{min}$	-	5.5	V	Using EECON to read/write $V_{min}$ = min operating voltage
D122	Erase/write cycle time	$T_{DEW}$	-	4	8	ms	
<b>Program FLASH Memory</b>							
D130	Endurance	$E_P$	1000	-	-	E/W	25°C at 5V
D131	$V_{DD}$ for read	$V_{PR}$	$V_{min}$	-	5.5	V	$V_{min}$ = min operating voltage
D132a	$V_{DD}$ for erase/write		$V_{min}$	-	5.5	V	using EECON to read/write, $V_{min}$ = min operating voltage
D133	Erase/Write cycle time	$T_{PEW}$	-	4	8	ms	

Legend: \* 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.

- Note 1:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16F87X be driven with external clock in RC mode.
- 2:** The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3:** Negative current is defined as current sourced by the pin.

# PIC16F87X

## 15.4 Timing Parameter Symbolology

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS
3. Tcc:ST (I<sup>2</sup>C specifications only)
4. Ts (I<sup>2</sup>C specifications only)

<b>T</b>		<b>T</b>	
F	Frequency	T	Time

Lowercase letters (pp) and their meanings:

<b>pp</b>			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	RD
cs	$\overline{CS}$	rw	RD or WR
di	SDI	sc	SCK
do	SDO	ss	SS
dt	Data in	t0	T0CKI
io	I/O port	t1	T1CKI
mc	$\overline{MCLR}$	wr	WR

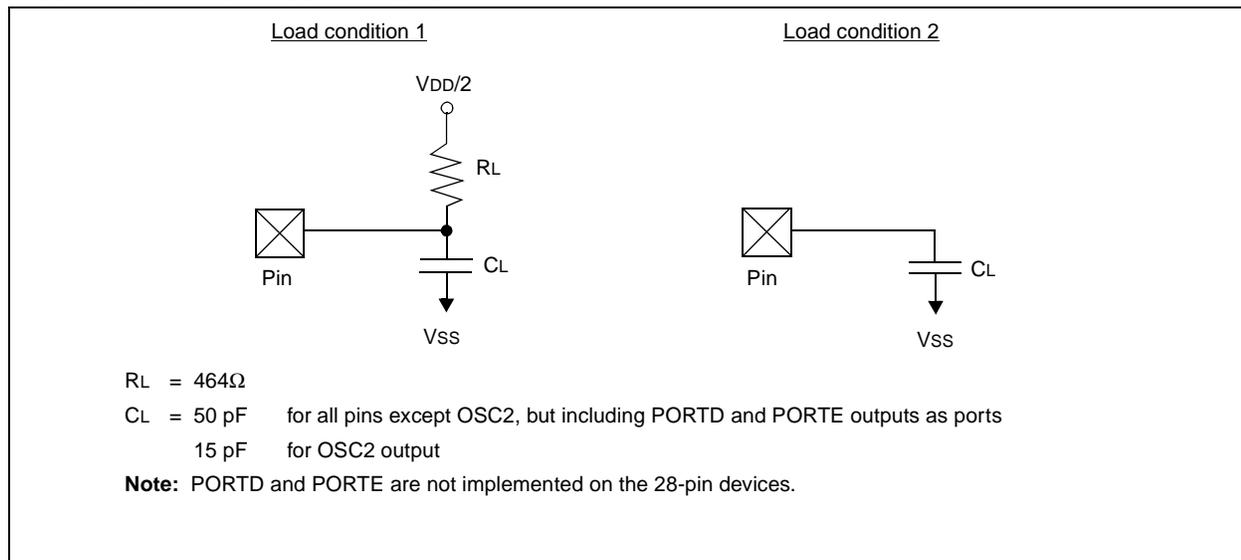
Uppercase letters and their meanings:

<b>S</b>			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (Hi-impedance)	V	Valid
L	Low	Z	Hi-impedance
<b>I<sup>2</sup>C only</b>			
AA	output access	High	High
BUF	Bus free	Low	Low

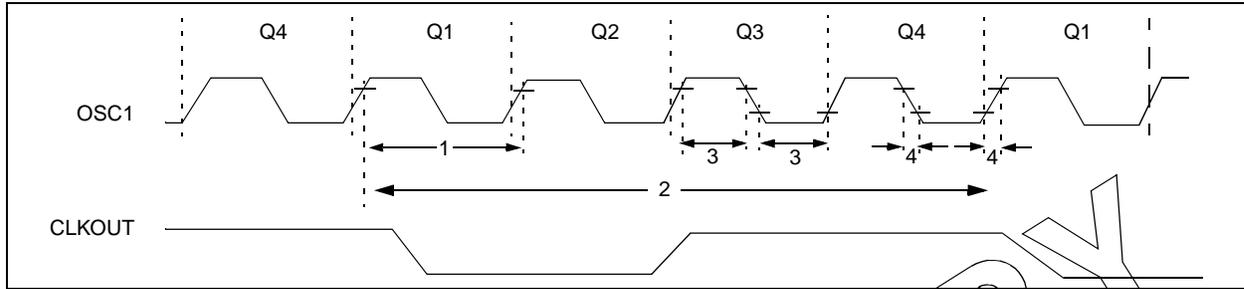
Tcc:ST (I<sup>2</sup>C specifications only)

<b>CC</b>			
HD	Hold	SU	Setup
<b>ST</b>			
DAT	DATA input hold	STO	STOP condition
STA	START condition		

**FIGURE 15-4: LOAD CONDITIONS**



**FIGURE 15-5: EXTERNAL CLOCK TIMING**



**TABLE 15-1: EXTERNAL CLOCK TIMING REQUIREMENTS**

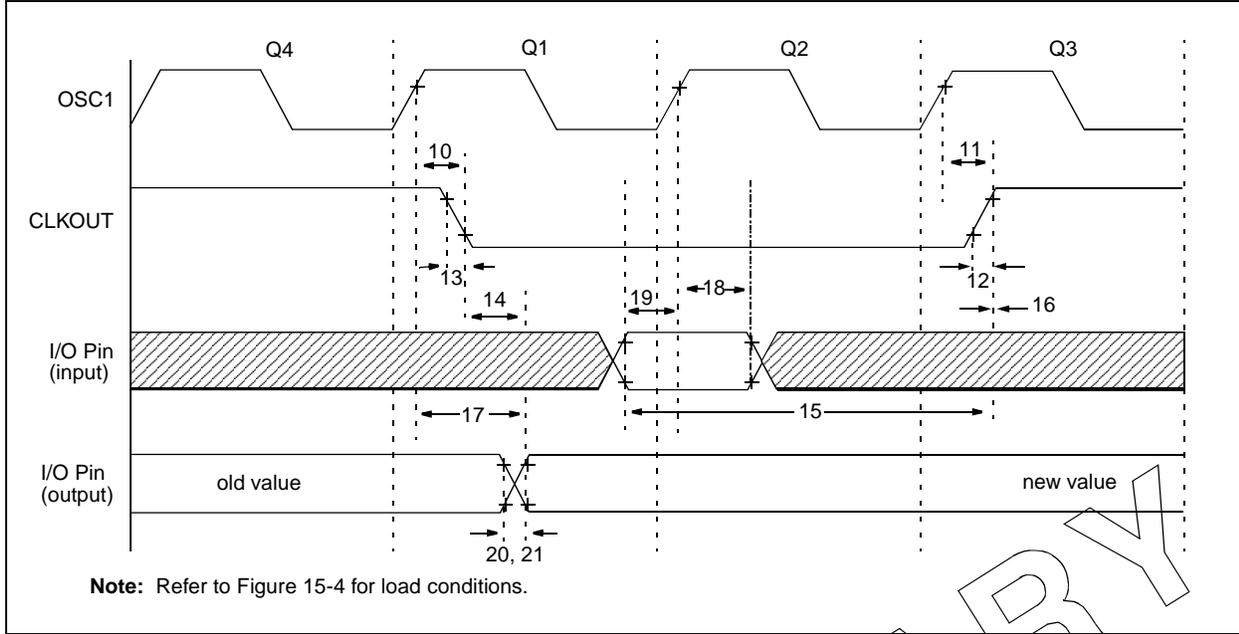
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	FOSC	<b>External CLKIN Frequency (Note 1)</b>	DC	—	4	MHz	XT and RC osc mode
			DC	—	4	MHz	HS osc mode (-04)
			DC	—	20	MHz	HS osc mode (-20)
			DC	—	200	kHz	LP osc mode
		<b>Oscillator Frequency (Note 1)</b>	DC	—	4	MHz	RC osc mode
			0.1	—	4	MHz	XT osc mode
4 5	— —		20 200	MHz kHz	HS osc mode LP osc mode		
1	TOSC	<b>External CLKIN Period (Note 1)</b>	250	—	—	ns	XT and RC osc mode
			250	—	—	ns	HS osc mode (-04)
			50	—	—	ns	HS osc mode (-20)
			5	—	—	µs	LP osc mode
		<b>Oscillator Period (Note 1)</b>	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
250 50 5	— — —		250 250 —	ns ns µs	HS osc mode (-04) HS osc mode (-20) LP osc mode		
2	TCY	<b>Instruction Cycle Time (Note 1)</b>	200	TCY	DC	ns	TCY = 4/FOSC
3	TosL, TosH	<b>External Clock in (OSC1) High or Low Time</b>	100	—	—	ns	XT oscillator
			2.5	—	—	µs	LP oscillator
			15	—	—	ns	HS oscillator
4	TosR, TosF	<b>External Clock in (OSC1) Rise or Fall Time</b>	—	—	25	ns	XT oscillator
			—	—	50	ns	LP oscillator
			—	—	15	ns	HS oscillator

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

**Note 1:** Instruction cycle period (TCY) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1/CLKIN pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

# PIC16F87X

**FIGURE 15-6: CLKOUT AND I/O TIMING**



**TABLE 15-2: CLKOUT AND I/O TIMING REQUIREMENTS**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
10*	TosH2ckL	OSC1↑ to CLKOUT↓	—	75	200	ns	Note 1	
11*	TosH2ckH	OSC1↑ to CLKOUT↑	—	75	200	ns	Note 1	
12*	TckR	CLKOUT rise time	—	35	100	ns	Note 1	
13*	TckF	CLKOUT fall time	—	35	100	ns	Note 1	
14*	TckL2ioV	CLKOUT ↓ to Port out valid	—	—	0.5TCY + 20	ns	Note 1	
15*	TioV2ckH	Port in valid before CLKOUT ↑	ToSC + 200	—	—	ns	Note 1	
16*	TckH2ioI	Port in hold after CLKOUT ↑	0	—	—	ns	Note 1	
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	100	255	ns		
18*	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	Standard (F)	100	—	ns		
		Extended (LF)	200	—	ns			
19*	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	0	—	—	ns		
20*	TioR	Port output rise time	Standard (F)	—	10	40	ns	
			Extended (LF)	—	—	145	ns	
21*	TioF	Port output fall time	Standard (F)	—	10	40	ns	
			Extended (LF)	—	—	145	ns	
22††*	Tinp	INT pin high or low time	TCY	—	—	ns		
23††*	Trbp	RB7:RB4 change INT high or low time	TCY	—	—	ns		

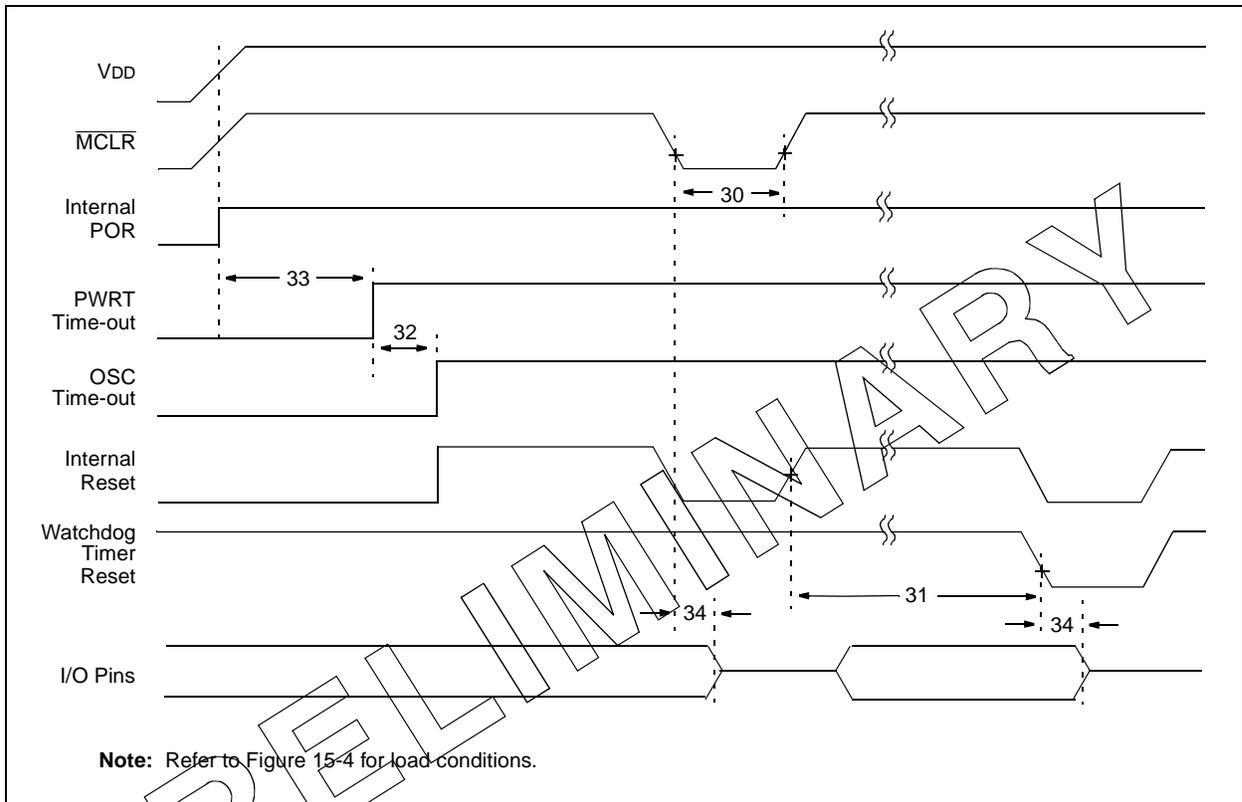
Legend: \* These parameters are characterized but not tested.

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

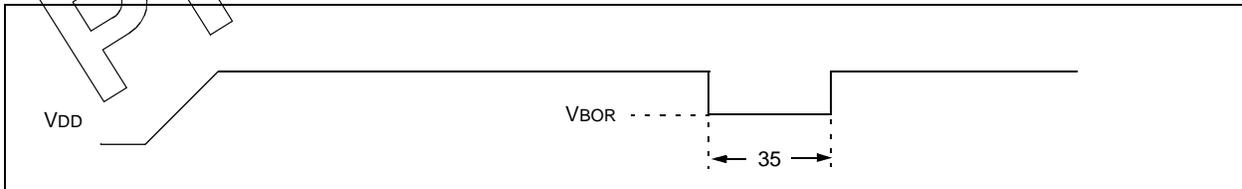
†† These parameters are asynchronous events not related to any internal clock edges.

**Note 1:** Measurements are taken in RC Mode where CLKOUT output is 4 x TosC.

**FIGURE 15-7: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**



**FIGURE 15-8: BROWN-OUT RESET TIMING**



**TABLE 15-3: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET REQUIREMENTS**

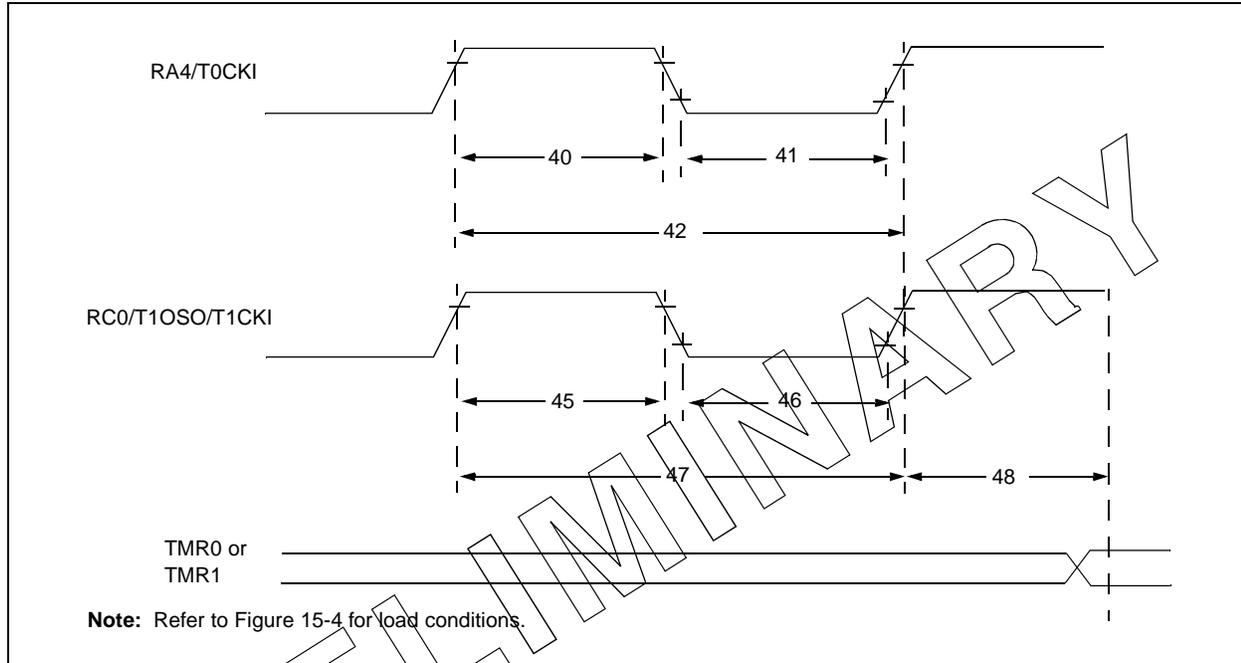
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	Tmcl	MCLR Pulse Width (low)	2	—	—	μs	VDD = 5V, -40°C to +85°C
31*	Twdt	Watchdog Timer Time-out Period (No Prescaler)	7	18	33	ms	VDD = 5V, -40°C to +85°C
32	Tost	Oscillation Start-up Timer Period	—	1024 TOSC	—	—	TOSC = OSC1 period
33*	Tpwrt	Power up Timer Period	28	72	132	ms	VDD = 5V, -40°C to +85°C
34	Tioz	I/O Hi-impedance from MCLR Low or Watchdog Timer Reset	—	—	2.1	μs	
35	TBOR	Brown-out Reset pulse width	100	—	—	μs	VDD ≤ VBOR (D005)

Legend: \* 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.

# PIC16F87X

**FIGURE 15-9: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**



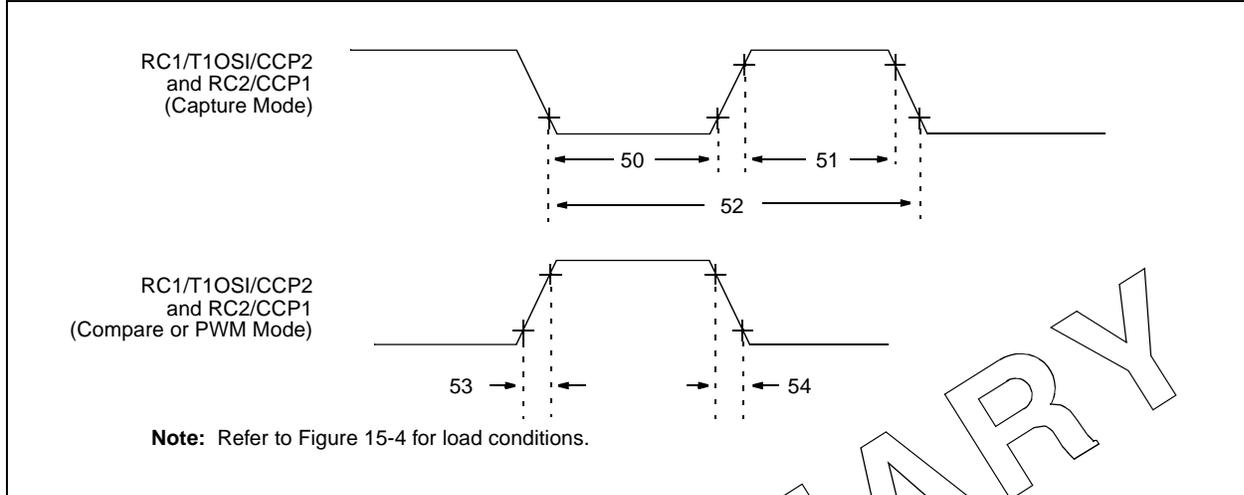
**TABLE 15-4: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions	
40*	Tt0H	T0CKI High Pulse Width	No Prescaler	$0.5TCY + 20$	—	—	ns	Must also meet parameter 42	
			With Prescaler	10	—	—	ns		
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	$0.5TCY + 20$	—	—	ns	Must also meet parameter 42	
			With Prescaler	10	—	—	ns		
42*	Tt0P	T0CKI Period	No Prescaler	$TCY + 40$	—	—	ns	N = prescale value (2, 4, ..., 256)	
			With Prescaler	Greater of: $20$ or $\frac{TCY + 40}{N}$	—	—	ns		
45*	Tt1H	T1CKI High Time	Synchronous, Prescaler = 1	$0.5TCY + 20$	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2,4,8	Standard(F)	15	—	—		ns
			Asynchronous	Standard(F)	30	—	—		ns
				Extended(LF)	50	—	—		ns
46*	Tt1L	T1CKI Low Time	Synchronous, Prescaler = 1	$0.5TCY + 20$	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2,4,8	Standard(F)	15	—	—		ns
			Asynchronous	Standard(F)	30	—	—		ns
				Extended(LF)	50	—	—		ns
47*	Tt1P	T1CKI input period	Synchronous	Standard(F)	Greater of: $30$ or $\frac{TCY + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
				Extended(LF)	Greater of: $50$ or $\frac{TCY + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	Standard(F)	60	—	—	ns	
				Extended(LF)	100	—	—	ns	
	Ft1	Timer1 oscillator input frequency range (oscillator enabled by setting bit T1OSCEN)		DC	—	200	kHz		
48	TCKEZtmr1	Delay from external clock edge to timer increment		$2Tosc$	—	$7Tosc$	—		

\* 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.

**FIGURE 15-10: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)**



**TABLE 15-5: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1 AND CCP2)**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
50*	TccL	CCP1 and CCP2 input low time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	
			With Prescaler	Standard(F)	10	—	—	ns
				Extended(LF)	20	—	—	ns
51*	TccH	CCP1 and CCP2 input high time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	
			With Prescaler	Standard(F)	10	—	—	ns
				Extended(LF)	20	—	—	ns
52*	TccP	CCP1 and CCP2 input period	$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 4 or 16)	
53*	TccR	CCP1 and CCP2 output rise time	Standard(F)	—	10	25	ns	
			Extended(LF)	—	25	50	ns	
54*	TccF	CCP1 and CCP2 output fall time	Standard(F)	—	10	25	ns	
			Extended(LF)	—	25	45	ns	

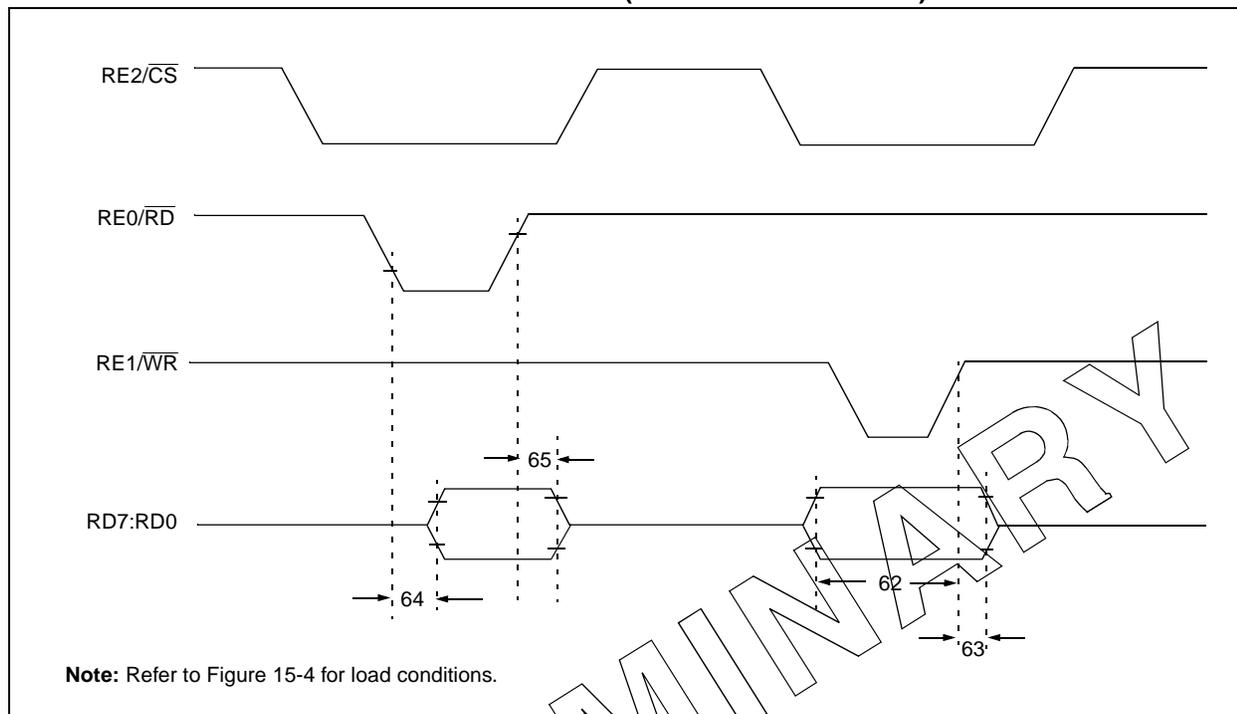
\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated.

These parameters are for design guidance only and are not tested.

# PIC16F87X

**FIGURE 15-11: PARALLEL SLAVE PORT TIMING (40-PIN DEVICES ONLY)**



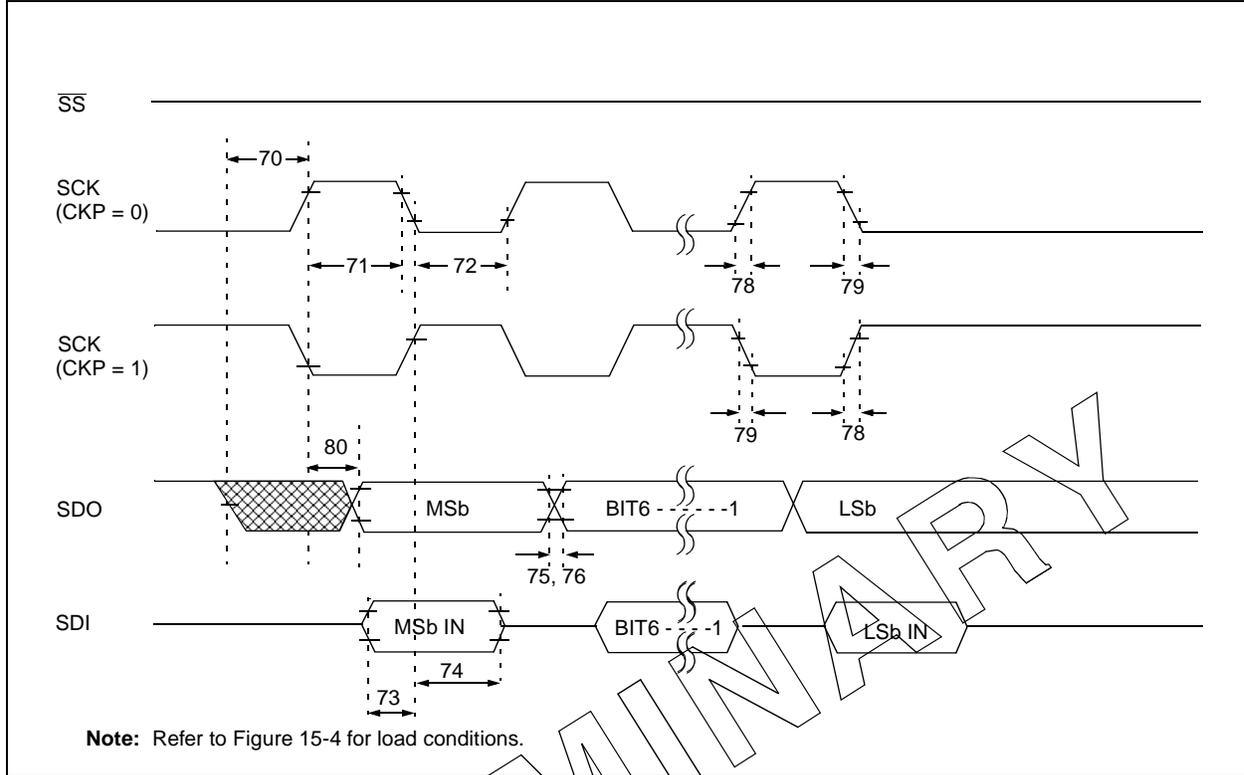
**TABLE 15-6: PARALLEL SLAVE PORT REQUIREMENTS (40-PIN DEVICES ONLY)**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
62	TdtV2wrH	Data in valid before WR↑ or CS↑ (setup time)	20 25	—	—	ns ns	Extended Range Only
63*	TwrH2dtH	WR↑ or CS↑ to data-in invalid (hold time)	20	—	—	ns	
			Standard(F)				
			Extended(LF)	35	—	—	ns
64	TrdL2dtV	RD↓ and CS↓ to data-out valid	—	—	80 90	ns ns	Extended Range Only
65	TrdH2dtI	RD↑ or CS↓ to data-out invalid	10	—	30	ns	

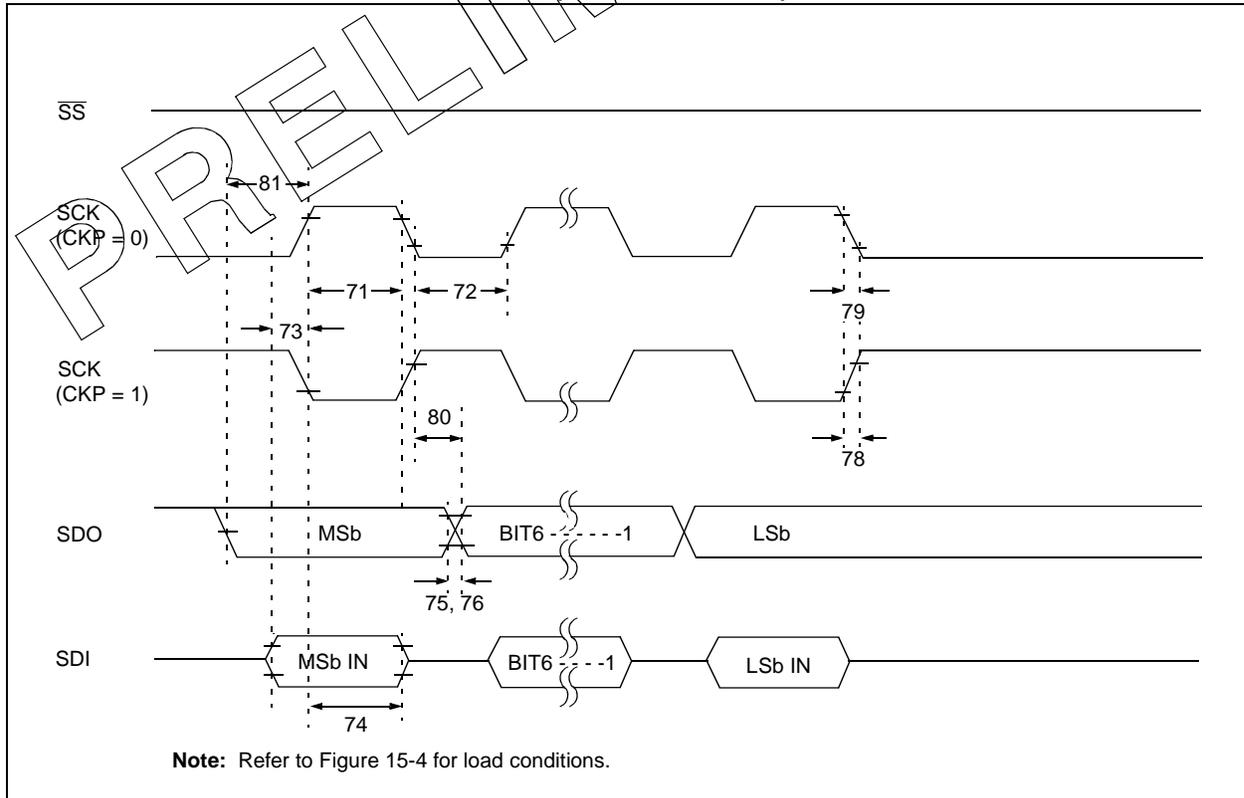
\* These parameters are characterized but not tested.

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

**FIGURE 15-12: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)**

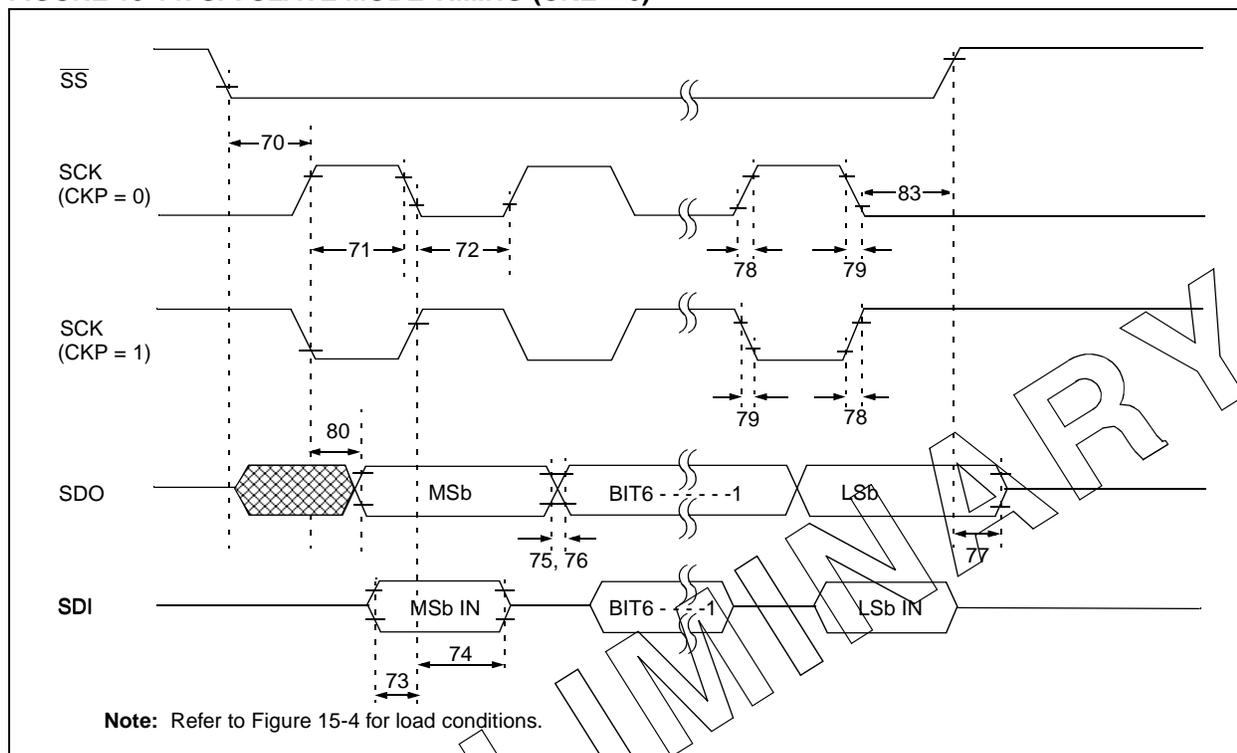


**FIGURE 15-13: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)**

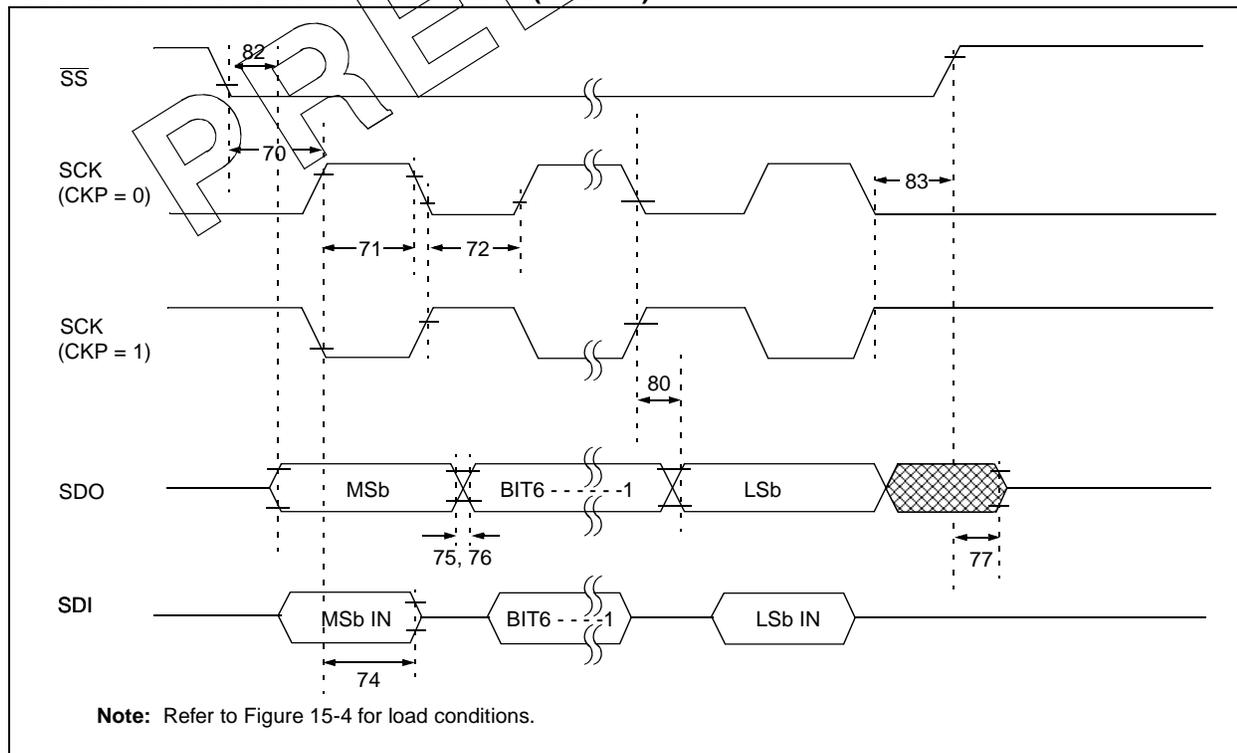


# PIC16F87X

**FIGURE 15-14: SPI SLAVE MODE TIMING (CKE = 0)**



**FIGURE 15-15: SPI SLAVE MODE TIMING (CKE = 1)**



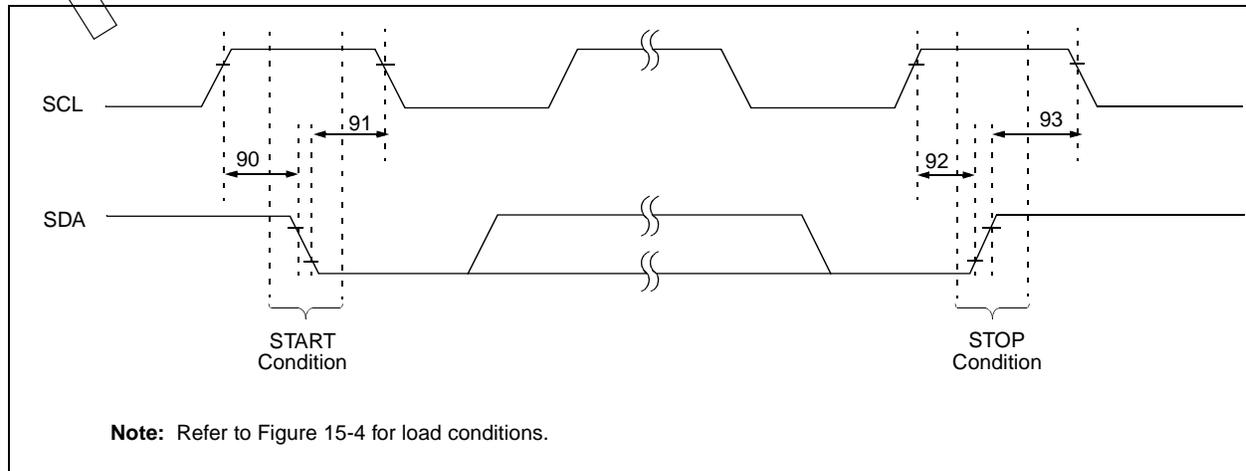
**TABLE 15-7: SPI MODE REQUIREMENTS**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
70*	Tssl2sch, Tssl2scl	SS↓ to SCK↓ or SCK↑ input	Tcy	—	—	ns	
71*	Tsch	SCK input high time (slave mode)	Tcy + 20	—	—	ns	
72*	Tscl	SCK input low time (slave mode)	Tcy + 20	—	—	ns	
73*	TdiV2sch, TdiV2scl	Setup time of SDI data input to SCK edge	100	—	—	ns	
74*	Tsch2diL, TscL2diL	Hold time of SDI data input to SCK edge	100	—	—	ns	
75*	TdoR	SDO data output rise time	Standard(F)	10	25	ns	
			Extended(LF)	25	50		
76*	TdoF	SDO data output fall time	—	10	25	ns	
77*	TssH2doZ	SS↑ to SDO output hi-impedance	10	—	50	ns	
78*	TscR	SCK output rise time (master mode)	Standard(F)	10	25	ns	
			Extended(LF)	25	50		
79*	TscF	SCK output fall time (master mode)	—	10	25	ns	
80*	Tsch2doV, TscL2doV	SDO data output valid after SCK edge	Standard(F)	—	50	ns	
			Extended(LF)	—	145		
81*	TdoV2sch, TdoV2scl	SDO data output setup to SCK edge	Tcy	—	—	ns	
82*	Tssl2doV	SDO data output valid after SS↓ edge	—	—	50	ns	
83*	Tsch2ssH, TscL2ssH	SS↓ after SCK edge	1.5Tcy + 40	—	—	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.

**FIGURE 15-16: I<sup>2</sup>C BUS START/STOP BITS TIMING**

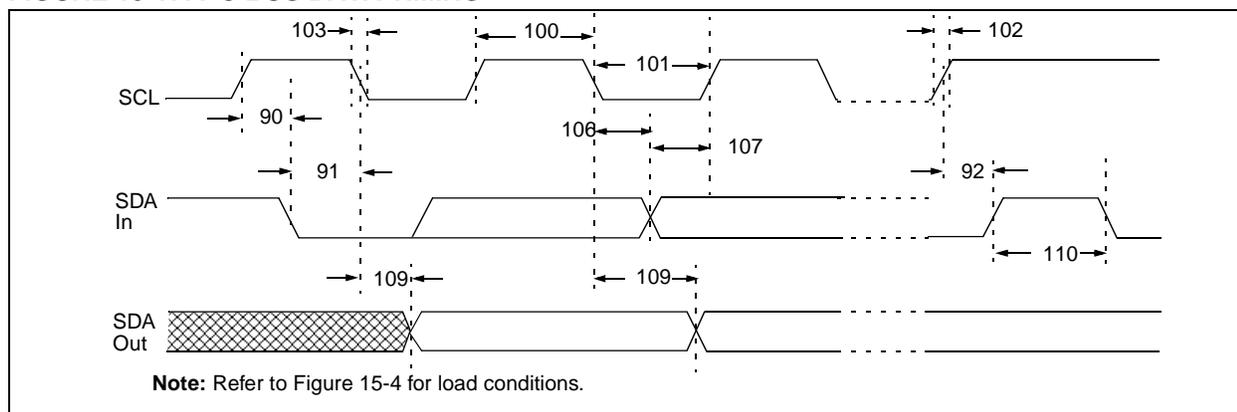


**TABLE 15-8: I<sup>2</sup>C BUS START/STOP BITS REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ	Max	Units	Conditions	
90	TSU:STA	START condition Setup time	100 kHz mode	4700	—	—	ns	Only relevant for repeated START condition
			400 kHz mode	600	—	—		
91	THD:STA	START condition Hold time	100 kHz mode	4000	—	—	ns	After this period the first clock pulse is generated
			400 kHz mode	600	—	—		
92	TSU:STO	STOP condition Setup time	100 kHz mode	4700	—	—	ns	
			400 kHz mode	600	—	—		
93	THD:STO	STOP condition Hold time	100 kHz mode	4000	—	—	ns	
			400 kHz mode	600	—	—		

# PIC16F87X

**FIGURE 15-17: I<sup>2</sup>C BUS DATA TIMING**



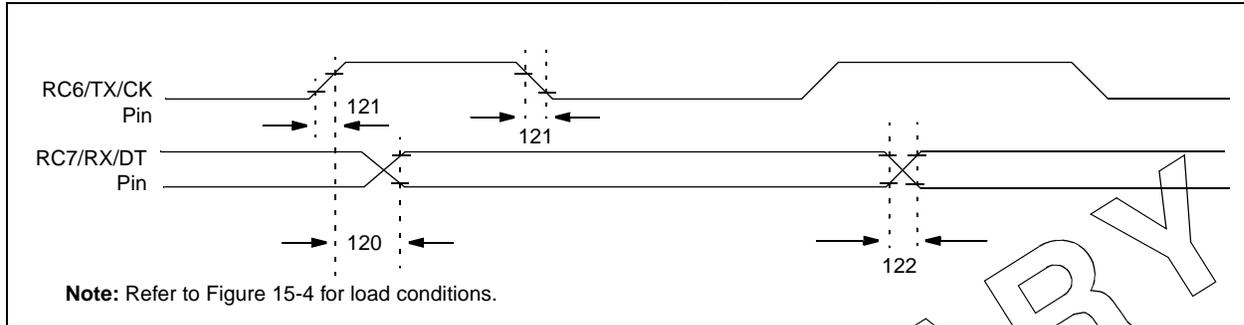
**TABLE 15-9: I<sup>2</sup>C BUS DATA REQUIREMENTS**

Param No.	Sym	Characteristic	Min	Max	Units	Conditions	
100	THIGH	Clock high time	100 kHz mode	4.0	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	μs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5T <sub>CY</sub>	—	—	—
101	TLOW	Clock low time	100 kHz mode	4.7	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	μs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5T <sub>CY</sub>	—	—	—
102	Tr	SDA and SCL rise time	100 kHz mode	—	1000	ns	Cb is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1Cb	300	ns	
103	Tf	SDA and SCL fall time	100 kHz mode	—	300	ns	Cb is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1Cb	300	ns	
90	TSU:STA	START condition setup time	100 kHz mode	4.7	—	μs	Only relevant for repeated START condition
			400 kHz mode	0.6	—	μs	
91	THD:STA	START condition hold time	100 kHz mode	4.0	—	μs	After this period the first clock pulse is generated
			400 kHz mode	0.6	—	μs	
106	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
107	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	Note 2
			400 kHz mode	100	—	ns	
92	TSU:STO	STOP condition setup time	100 kHz mode	4.7	—	μs	
			400 kHz mode	0.6	—	μs	
109	TAA	Output valid from clock	100 kHz mode	—	3500	ns	Note 1
			400 kHz mode	—	—	ns	
110	TBUF	Bus free time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
	Cb	Bus capacitive loading	—	400	pF		

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

**2:** A fast-mode (400 kHz) I<sup>2</sup>C-bus device can be used in a standard-mode (100 kHz) I<sup>2</sup>C-bus system, but the requirement tsu; DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line Tr max. +tsu; DAT = 1000 + 250 = 1250 ns (according to the standard-mode I<sup>2</sup>C bus specification) before the SCL line is released.

**FIGURE 15-18: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**

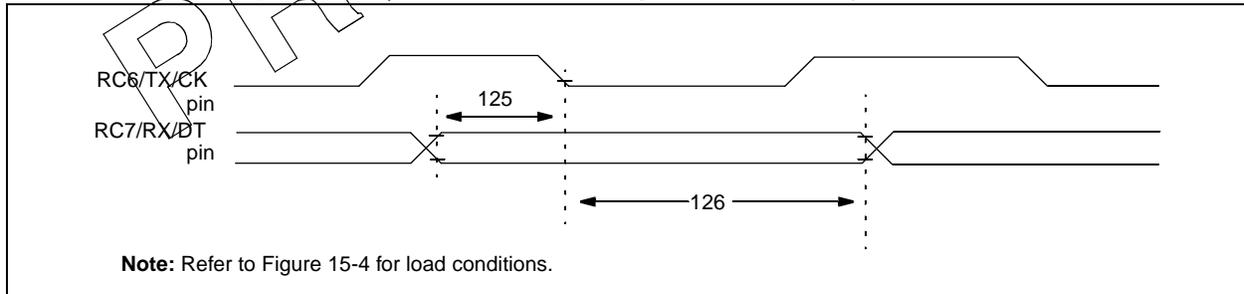


**TABLE 15-10: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE)	Standard(F)	—	—	80	ns
		Clock high to data out valid	Extended(LF)	—	—	100	ns
121	Tckrf	Clock out rise time and fall time (Master Mode)	Standard(F)	—	—	45	ns
			Extended(LF)	—	—	50	ns
122	TdtV	Data out rise time and fall time	Standard(F)	—	—	45	ns
			Extended(LF)	—	—	50	ns

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

**FIGURE 15-19: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**



**TABLE 15-11: USART SYNCHRONOUS RECEIVE REQUIREMENTS**

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

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

# PIC16F87X

**TABLE 15-12: PIC16F873/874/876/877-04 (COMMERCIAL, INDUSTRIAL)  
PIC16F873/874/876/877-20 (COMMERCIAL, INDUSTRIAL)  
PIC16LF873/874/876/877-04 (COMMERCIAL, INDUSTRIAL)**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
A01	NR	Resolution	—	—	10-bits	bit	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$
A03	EIL	Integral linearity error	—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$
A04	EDL	Differential linearity error	—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$
A06	E0FF	Offset error	—	—	$< \pm 2$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$
A07	EGN	Gain error	—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$ , $V_{SS} \leq V_{AIN} \leq V_{REF}$
A10	—	Monotonicity <sup>(3)</sup>	—	guaranteed	—	—	$V_{SS} \leq V_{AIN} \leq V_{REF}$
A20	VREF	Reference voltage ( $V_{REF+} - V_{REF-}$ )	2.0V	—	$V_{DD} + 0.3$	V	Absolute minimum electrical spec. To ensure 10-bit accuracy.
A21	VREF+	Reference voltage High	$AV_{DD} - 2.5V$	—	$AV_{DD} + 0.3V$	V	
A22	VREF-	Reference voltage low	$AV_{SS} - 0.3V$	—	$V_{REF+} - 2.0V$	V	
A25	VAIN	Analog input voltage	$V_{SS} - 0.3$	—	$V_{REF} + 0.3$	V	
A30	ZAIN	Recommended impedance of analog voltage source	—	—	10.0	k $\Omega$	
A40	IAD	A/D conversion current ( $V_{DD}$ )	Standard	220	—	$\mu A$	Average current consumption when A/D is on. (Note 1)
			Extended	90	—	$\mu A$	
A50	IREF	VREF input current (Note 2)	10	—	1000	$\mu A$	During VAIN acquisition. Based on differential of $V_{HOLD}$ to VAIN to charge $CHOLD$ , see Section 11.1.
			—	—	10	$\mu A$	During A/D Conversion cycle

\* 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.

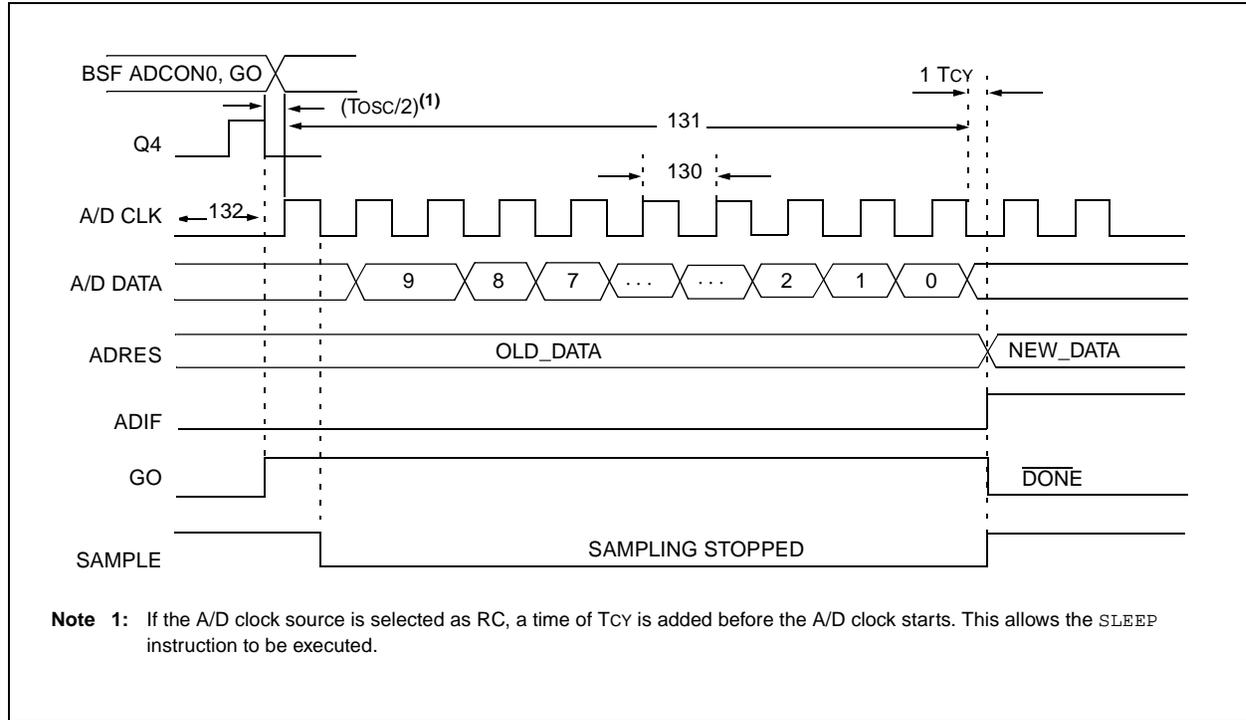
**Note 1:** When A/D is off, it will not consume any current other than minor leakage current.

The power-down current spec includes any such leakage from the A/D module.

**2:** VREF current is from RA3 pin or VDD pin, whichever is selected as reference input.

**3:** The A/D conversion result never decreases with an increase in the Input Voltage, and has no missing codes.

**FIGURE 15-20: A/D CONVERSION TIMING**



**TABLE 15-13: A/D CONVERSION REQUIREMENTS**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
130	TAD	A/D clock period	Standard(F)	1.6	—	—	$\mu$ s	TOSC based, $V_{REF} \geq 3.0V$
			Extended(LF)	3.0	—	—	$\mu$ s	TOSC based, $V_{REF} \geq 2.0V$
			Standard(F)	2.0	4.0	6.0	$\mu$ s	A/D RC Mode
			Extended(LF)	3.0	6.0	9.0	$\mu$ s	A/D RC Mode
131	TCNV	Conversion time (not including S/H time) (Note 1)		—	12	TAD		
132	TACQ	Acquisition time	Note 2	40	—	$\mu$ s	The minimum time is the amplifier settling time. This may be used if the "new" input voltage has not changed by more than 1 LSb (i.e., 20.0 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).	
			10*	—	—	$\mu$ s		
134	TGO	Q4 to A/D clock start	—	$T_{osc}/2$ §	—	—	If the A/D clock source is selected as RC, a time of $T_{CY}$ is added before the A/D clock starts. This allows the SLEEP instruction to be executed.	

\* 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.

§ This specification ensured by design.

**Note 1:** ADRES register may be read on the following  $T_{CY}$  cycle.

**2:** See Section 11.1 for min conditions.

# PIC16F87X

---

NOTES:

## 16.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

The graphs and tables provided in this section are for **design guidance** and are **not tested**.

In some graphs or tables, the data presented are **outside specified operating range** (i.e., outside specified V<sub>DD</sub> range). This is for **information only** and devices are ensured to operate properly only within the specified range.

The data presented in this section is a **statistical summary** of data collected on units from different lots over a period of time and matrix samples. 'Typical' represents the mean of the distribution at 25°C. 'Max' or 'min' represents (mean + 3σ) or (mean - 3σ) respectively, where σ is standard deviation, over the whole temperature range.

**Graphs and Tables not available at this time.**

# PIC16F87X

---

NOTES:

## 17.0 PACKAGING INFORMATION

### 17.1 Package Marking Information

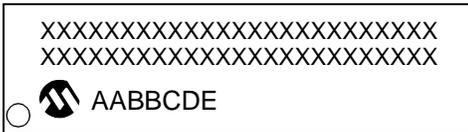
28-Lead PDIP (Skinny DIP)



Example



28-Lead SOIC



Example



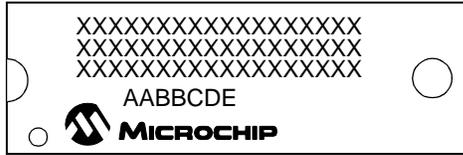
<b>Legend:</b>	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calendar year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured
		O = Outside Vendor
		C = 5" Line
		S = 6" Line
		H = 8" Line
	D	Mask revision number
	E	Assembly code of the plant or country of origin in which part was assembled
<b>Note:</b> In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.		

\* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask rev#, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

# PIC16F87X

## Package Marking Information (Cont'd)

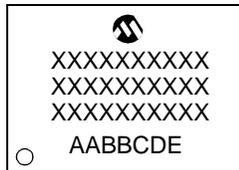
40-Lead PDIP



Example



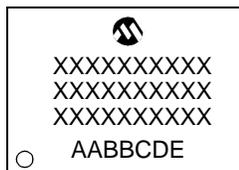
44-Lead TQFP



Example



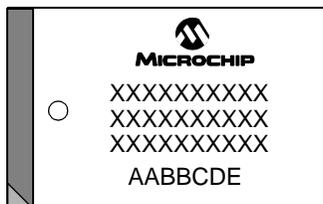
44-Lead MQFP



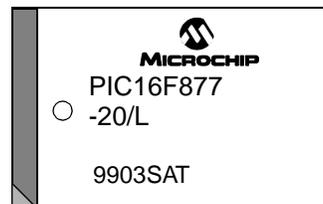
Example



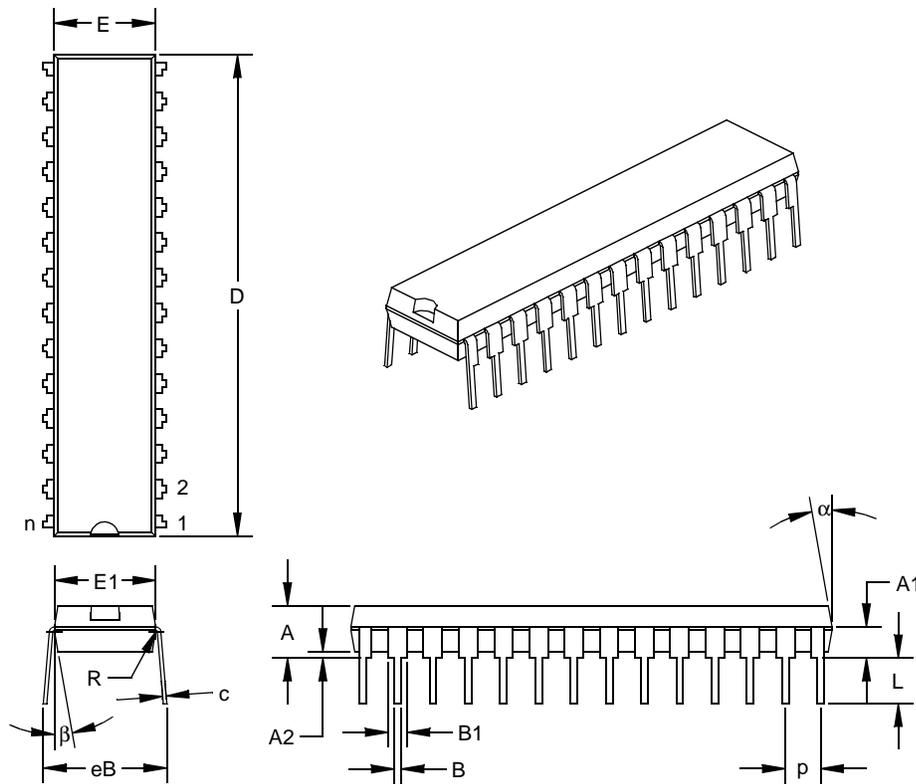
44-Lead PLCC



Example



## 17.2 K04-070 28-Lead Skinny Plastic Dual In-line (SP) – 300 mil



Units		INCHES*			MILLIMETERS		
		MIN	NOM	MAX	MIN	NOM	MAX
Dimension Limits			0.300			7.62	
PCB Row Spacing			28			28	
Number of Pins	n						
Pitch	p		0.100			2.54	
Lower Lead Width	B	0.016	0.019	0.022	0.41	0.48	0.56
Upper Lead Width	B1 <sup>†</sup>	0.040	0.053	0.065	1.02	1.33	1.65
Shoulder Radius	R	0.000	0.005	0.010	0.00	0.13	0.25
Lead Thickness	c	0.008	0.010	0.012	0.20	0.25	0.30
Top to Seating Plane	A	0.140	0.150	0.160	3.56	3.81	4.06
Top of Lead to Seating Plane	A1	0.070	0.090	0.110	1.78	2.29	2.79
Base to Seating Plane	A2	0.015	0.020	0.025	0.38	0.51	0.64
Tip to Seating Plane	L	0.125	0.130	0.135	3.18	3.30	3.43
Package Length	D <sup>‡</sup>	1.345	1.365	1.385	34.16	34.67	35.18
Molded Package Width	E <sup>‡</sup>	0.280	0.288	0.295	7.11	7.30	7.49
Radius to Radius Width	E1	0.270	0.283	0.295	6.86	7.18	7.49
Overall Row Spacing	eB	0.320	0.350	0.380	8.13	8.89	9.65
Mold Draft Angle Top	alpha	5	10	15	5	10	15
Mold Draft Angle Bottom	beta	5	10	15	5	10	15

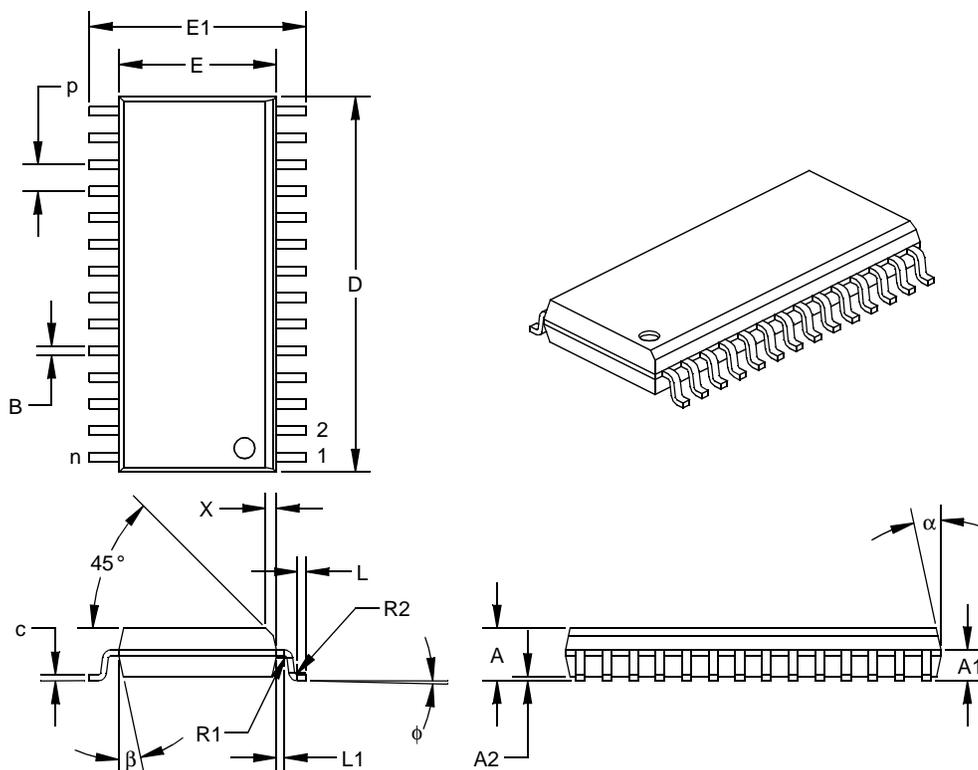
\* Controlling Parameter.

<sup>†</sup> Dimension "B1" does not include dam-bar protrusions. Dam-bar protrusions shall not exceed 0.003" (0.076 mm) per side or 0.006" (0.152 mm) more than dimension "B1."

<sup>‡</sup> Dimensions "D" and "E" do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010" (0.254 mm) per side or 0.020" (0.508 mm) more than dimensions "D" or "E."

# PIC16F87X

## 17.3 K04-052 28-Lead Plastic Small Outline (SO) – Wide, 300 mil



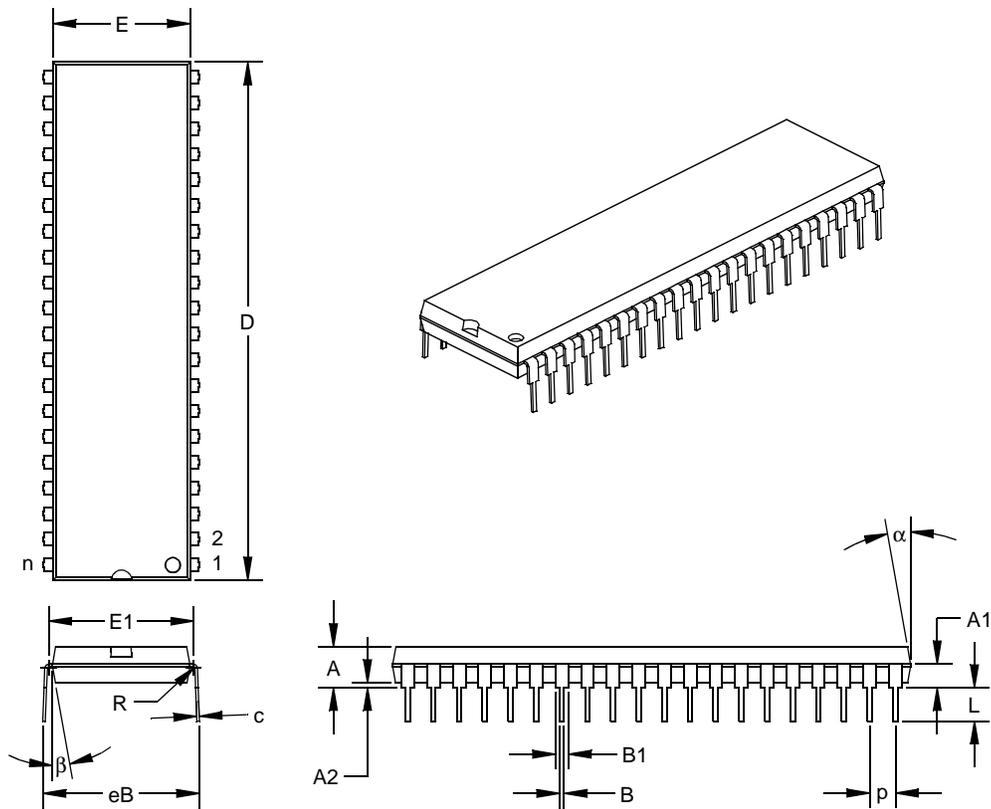
Units		INCHES*			MILLIMETERS		
		MIN	NOM	MAX	MIN	NOM	MAX
Pitch	p		0.050			1.27	
Number of Pins	n		28			28	
Overall Pack. Height	A	0.093	0.099	0.104	2.36	2.50	2.64
Shoulder Height	A1	0.048	0.058	0.068	1.22	1.47	1.73
Standoff	A2	0.004	0.008	0.011	0.10	0.19	0.28
Molded Package Length	D <sup>‡</sup>	0.700	0.706	0.712	17.78	17.93	18.08
Molded Package Width	E <sup>‡</sup>	0.292	0.296	0.299	7.42	7.51	7.59
Outside Dimension	E1	0.394	0.407	0.419	10.01	10.33	10.64
Chamfer Distance	X	0.010	0.020	0.029	0.25	0.50	0.74
Shoulder Radius	R1	0.005	0.005	0.010	0.13	0.13	0.25
Gull Wing Radius	R2	0.005	0.005	0.010	0.13	0.13	0.25
Foot Length	L	0.011	0.016	0.021	0.28	0.41	0.53
Foot Angle	φ	0	4	8	0	4	8
Radius Centerline	L1	0.010	0.015	0.020	0.25	0.38	0.51
Lead Thickness	c	0.009	0.011	0.012	0.23	0.27	0.30
Lower Lead Width	B <sup>†</sup>	0.014	0.017	0.019	0.36	0.42	0.48
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

\* Controlling Parameter.

† Dimension "B" does not include dam-bar protrusions. Dam-bar protrusions shall not exceed 0.003" (0.076 mm) per side or 0.006" (0.152 mm) more than dimension "B."

‡ Dimensions "D" and "E" do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010" (0.254 mm) per side or 0.020" (0.508 mm) more than dimensions "D" or "E."

## 17.4 K04-016 40-Lead Plastic Dual In-line (P) – 600 mil



Units		INCHES*			MILLIMETERS		
		MIN	NOM	MAX	MIN	NOM	MAX
Dimension Limits							
PCB Row Spacing			0.600			15.24	
Number of Pins	n		40			40	
Pitch	p		0.100			2.54	
Lower Lead Width	B	0.016	0.018	0.020	0.41	0.46	0.51
Upper Lead Width	B1 <sup>†</sup>	0.045	0.050	0.055	1.14	1.27	1.40
Shoulder Radius	R	0.000	0.005	0.010	0.00	0.13	0.25
Lead Thickness	c	0.009	0.010	0.011	0.23	0.25	0.28
Top to Seating Plane	A	0.110	0.160	0.160	2.79	4.06	4.06
Top of Lead to Seating Plane	A1	0.073	0.093	0.113	1.85	2.36	2.87
Base to Seating Plane	A2	0.020	0.020	0.040	0.51	0.51	1.02
Tip to Seating Plane	L	0.125	0.130	0.135	3.18	3.30	3.43
Package Length	D <sup>‡</sup>	2.013	2.018	2.023	51.13	51.26	51.38
Molded Package Width	E <sup>‡</sup>	0.530	0.535	0.540	13.46	13.59	13.72
Radius to Radius Width	E1	0.545	0.565	0.585	13.84	14.35	14.86
Overall Row Spacing	eB	0.630	0.610	0.670	16.00	15.49	17.02
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

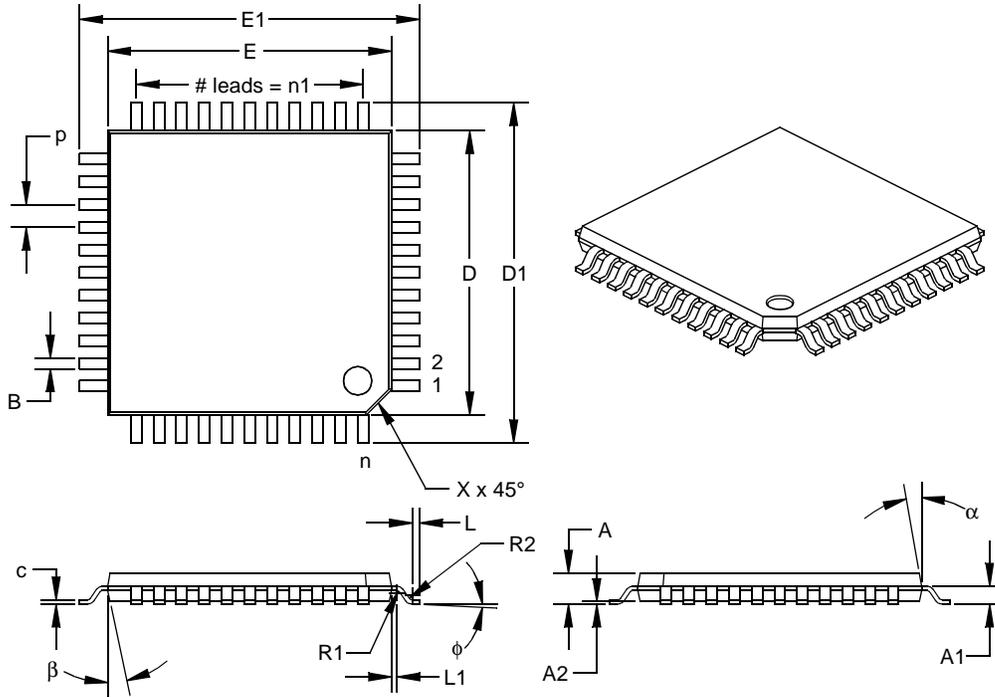
\* Controlling Parameter.

<sup>†</sup> Dimension "B1" does not include dam-bar protrusions. Dam-bar protrusions shall not exceed 0.003" (0.076 mm) per side or 0.006" (0.152 mm) more than dimension "B1."

<sup>‡</sup> Dimensions "D" and "E" do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010" (0.254 mm) per side or 0.020" (0.508 mm) more than dimensions "D" or "E."

# PIC16F87X

## 17.5 K04-076 44-Lead Plastic Thin Quad Flatpack (PT) 10x10x1 mm Body, 1.0/0.1 mm Lead Form



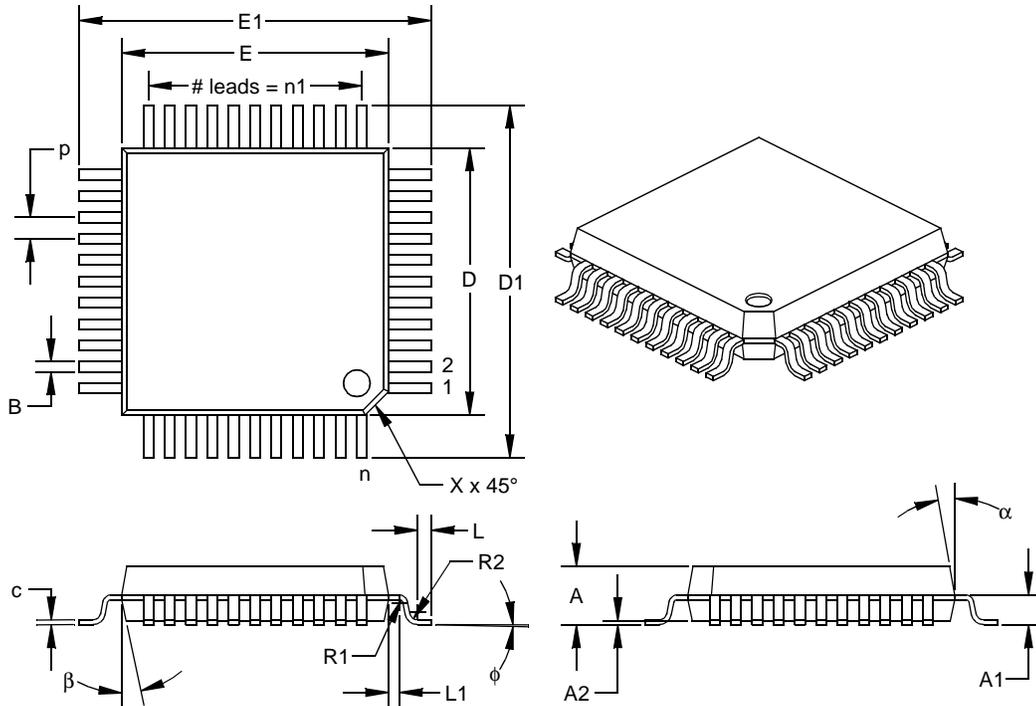
Units		INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Pitch	p		0.031			0.80	
Number of Pins	n		44			44	
Pins along Width	n1		11			11	
Overall Pack. Height	A	0.039	0.043	0.047	1.00	1.10	1.20
Shoulder Height	A1	0.015	0.025	0.035	0.38	0.64	0.89
Standoff	A2	0.002	0.004	0.006	0.05	0.10	0.15
Shoulder Radius	R1	0.003	0.003	0.010	0.08	0.08	0.25
Gull Wing Radius	R2	0.003	0.006	0.008	0.08	0.14	0.20
Foot Length	L	0.005	0.010	0.015	0.13	0.25	0.38
Foot Angle	φ	0	3.5	7	0	3.5	7
Radius Centerline	L1	0.003	0.008	0.013	0.08	0.20	0.33
Lead Thickness	c	0.004	0.006	0.008	0.09	0.15	0.20
Lower Lead Width	B <sup>†</sup>	0.012	0.015	0.018	0.30	0.38	0.45
Outside Tip Length	D1	0.463	0.472	0.482	11.75	12.00	12.25
Outside Tip Width	E1	0.463	0.472	0.482	11.75	12.00	12.25
Molded Pack. Length	D <sup>‡</sup>	0.390	0.394	0.398	9.90	10.00	10.10
Molded Pack. Width	E <sup>‡</sup>	0.390	0.394	0.398	9.90	10.00	10.10
Pin 1 Corner Chamfer	X	0.025	0.035	0.045	0.64	0.89	1.14
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	12	15	5	12	15

\* Controlling Parameter.

† Dimension "B" does not include dam-bar protrusions. Dam-bar protrusions shall not exceed 0.003" (0.076 mm) per side or 0.006" (0.152 mm) more than dimension "B."

‡ Dimensions "D" and "E" do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010" (0.254 mm) per side or 0.020" (0.508 mm) more than dimensions "D" or "E."  
JEDEC equivalent:MS-026 ACB

## 17.6 K04-071 44-Lead Plastic Quad Flatpack (PQ) 10x10x2 mm Body, 1.6/0.15 mm Lead Form



Units		INCHES			MILLIMETERS*		
		MIN	NOM	MAX	MIN	NOM	MAX
Pitch	p		0.031			0.80	
Number of Pins	n		44			44	
Pins along Width	n1		11			11	
Overall Pack. Height	A	0.079	0.086	0.093	2.00	2.18	2.35
Shoulder Height	A1	0.032	0.044	0.056	0.81	1.11	1.41
Standoff	A2	0.002	0.006	0.010	0.05	0.15	0.25
Shoulder Radius	R1	0.005	0.005	0.010	0.13	0.13	0.25
Gull Wing Radius	R2	0.005	0.012	0.015	0.13	0.30	0.38
Foot Length	L	0.015	0.020	0.025	0.38	0.51	0.64
Foot Angle	phi	0	3.5	7	0	3.5	7
Radius Centerline	L1	0.011	0.016	0.021	0.28	0.41	0.53
Lead Thickness	c	0.005	0.007	0.009	0.13	0.18	0.23
Lower Lead Width	B†	0.012	0.015	0.018	0.30	0.37	0.45
Outside Tip Length	D1	0.510	0.520	0.530	12.95	13.20	13.45
Outside Tip Width	E1	0.510	0.520	0.530	12.95	13.20	13.45
Molded Pack. Length	D‡	0.390	0.394	0.398	9.90	10.00	10.10
Molded Pack. Width	E‡	0.390	0.394	0.398	9.90	10.00	10.10
Pin 1 Corner Chamfer	X	0.025	0.035	0.045	0.635	0.89	1.143
Mold Draft Angle Top	alpha	5	10	15	5	10	15
Mold Draft Angle Bottom	beta	5	12	15	5	12	15

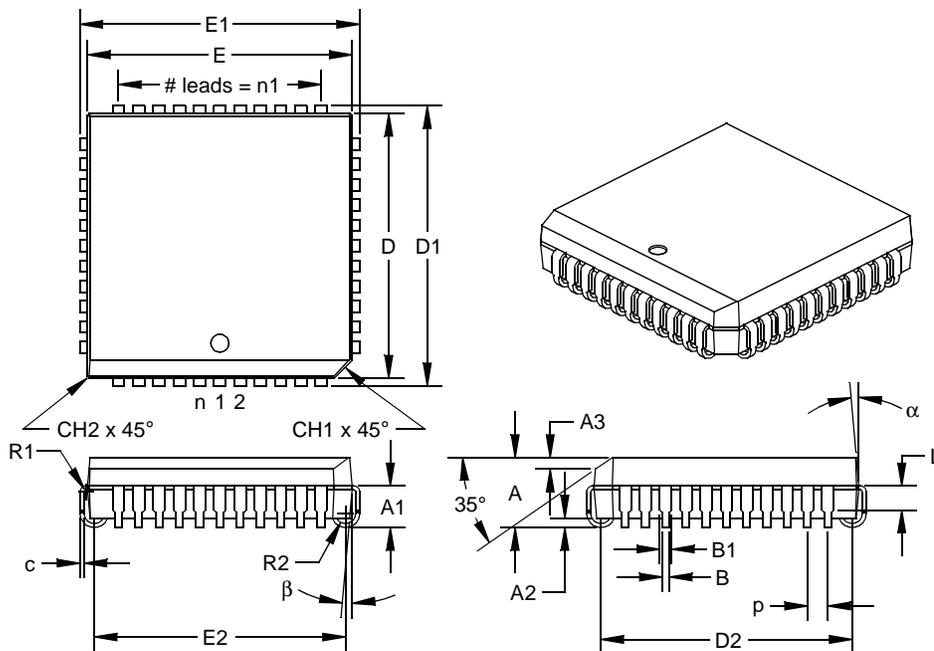
\* Controlling Parameter.

† Dimension "B" does not include dam-bar protrusions. Dam-bar protrusions shall not exceed 0.003" (0.076 mm) per side or 0.006" (0.152 mm) more than dimension "B."

‡ Dimensions "D" and "E" do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010" (0.254 mm) per side or 0.020" (0.508 mm) more than dimensions "D" or "E."  
JEDEC equivalent:MS-022 AB

# PIC16F87X

## 17.7 K04-048 44-Lead Plastic Leaded Chip Carrier (L) – Square



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		44			44	
Pitch	p		0.050			1.27	
Overall Pack. Height	A	0.165	0.173	0.180	4.19	4.38	4.57
Shoulder Height	A1	0.095	0.103	0.110	2.41	2.60	2.79
Standoff	A2	0.015	0.023	0.030	0.38	0.57	0.76
Side 1 Chamfer Dim.	A3	0.024	0.029	0.034	0.61	0.74	0.86
Corner Chamfer (1)	CH1	0.040	0.045	0.050	1.02	1.14	1.27
Corner Chamfer (other)	CH2	0.000	0.005	0.010	0.00	0.13	0.25
Overall Pack. Width	E1	0.685	0.690	0.695	17.40	17.53	17.65
Overall Pack. Length	D1	0.685	0.690	0.695	17.40	17.53	17.65
Molded Pack. Width	E <sup>‡</sup>	0.650	0.653	0.656	16.51	16.59	16.66
Molded Pack. Length	D <sup>‡</sup>	0.650	0.653	0.656	16.51	16.59	16.66
Footprint Width	E2	0.610	0.620	0.630	15.49	15.75	16.00
Footprint Length	D2	0.610	0.620	0.630	15.49	15.75	16.00
Pins along Width	n1		11			11	
Lead Thickness	c	0.008	0.010	0.012	0.20	0.25	0.30
Upper Lead Width	B1 <sup>†</sup>	0.026	0.029	0.032	0.66	0.74	0.81
Lower Lead Width	B	0.015	0.018	0.021	0.38	0.46	0.53
Upper Lead Length	L	0.050	0.058	0.065	1.27	1.46	1.65
Shoulder Inside Radius	R1	0.003	0.005	0.010	0.08	0.13	0.25
J-Bend Inside Radius	R2	0.015	0.025	0.035	0.38	0.64	0.89
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

\* Controlling Parameter.

† Dimension "B1" does not include dam-bar protrusions. Dam-bar protrusions shall not exceed 0.003" (0.076 mm) per side or 0.006" (0.152 mm) more than dimension "B1."

‡ Dimensions "D" and "E" do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010" (0.254 mm) per side or 0.020" (0.508 mm) more than dimensions "D" or "E."

JEDEC equivalent: MO-047 AC

## APPENDIX A: REVISION HISTORY

Version	Date	Revision Description
A	1998	This is a new data sheet. However, these devices are similar to the PIC16C7X devices found in the PIC16C7X Data Sheet (DS30390). Data Memory Map for PIC16F873/874, moved ADFM bit from ADCON1<5> to ADCON1<7>
B	1999	FLASH EEPROM access information.

## APPENDIX B: DEVICE DIFFERENCES

The differences between the devices in this data sheet are listed in Table B-1.

**TABLE B-1: DEVICE DIFFERENCES**

Difference	PIC16F876/873	PIC16F877/874
A/D	5 channels, 10bits	8 channels, 10bits
Parallel Slave Port	no	yes
Packages	28-pin PDIP, 28-pin windowed CERDIP, 28-pin SOIC	40-pin PDIP, 44-pin TQFP, 44-pin MQFP, 44-pin PLCC

## APPENDIX C: CONVERSION CONSIDERATIONS

Considerations for converting from previous versions of devices to the ones listed in this data sheet are listed in Table C-1.

**TABLE C-1: CONVERSION CONSIDERATIONS**

Characteristic	PIC16C7X	PIC16F87X
Pins	28/40	28/40
Timers	3	3
Interrupts	11 or 12	13 or 14
Communication	PSP, USART, SSP (SPI, I <sup>2</sup> C Slave)	PSP, USART, SSP (SPI, I <sup>2</sup> C Master/Slave)
Frequency	20 MHz	20 MHz
A/D	8-bit	10-bit
CCP	2	2
Program Memory	4K, 8K EPROM	4K, 8K FLASH
RAM	192, 368 bytes	192, 368 bytes
EEPROM data	None	128, 256 bytes
Other	—	In-Circuit Debugger, Low Voltage Programming

# PIC16F87X

---

NOTES:

## INDEX

### A

A/D	111
ADCON0 Register	111
ADCON1 Register	112
ADIF bit	113
Analog Input Model Block Diagram	115
Analog Port Pins	7, 8, 9, 37, 38
Block Diagram	114
Configuring Analog Port Pins	116
Configuring the Interrupt	113
Configuring the Module	113
Conversion Clock	116
Conversions	117
Delays	115
Effects of a Reset	118
GO/DONE bit	113
Internal Sampling Switch (Rss) Impedence	114
Operation During Sleep	118
Sampling Requirements	114
Source Impedence	114
Time Delays	115
Absolute Maximum Ratings	151
ACK	72
Acknowledge Data bit	66
Acknowledge Pulse	72
Acknowledge Sequence Enable bit	66
Acknowledge Status bit	66
ADRES Register	15, 111
Application Note AN578, "Use of the SSP Module in the I2C Multi-Master Environment."	71
Application Notes	
AN552 (Implementing Wake-up on Key Strokes Using PIC16CXXX)	31
AN556 (Table Reading Using PIC16CXX)	26
Architecture	
PIC16F873/PIC16F876 Block Diagram	5
PIC16F874/PIC16F877 Block Diagram	6
Assembler	
MPASM Assembler	145

### B

Banking, Data Memory	12, 18
Baud Rate Generator	78
BCLIF	24
BF	64, 72, 81, 83
Block Diagrams	
A/D	114
Analog Input Model	115
Baud Rate Generator	78
Capture	59
Compare	60
I <sup>2</sup> C Master Mode	76
I <sup>2</sup> C Module	71
PWM	60
SSP (I <sup>2</sup> C Mode)	71
SSP (SPI Mode)	67
Timer0/WDT Prescaler	47
Timer2	55
USART Receive	101
USART Transmit	99
BRG	78
BRGH bit	97
Brown-out Reset (BOR)	121, 125, 127, 128
BOR Status (BOR Bit)	25

Buffer Full bit, BF	72
Buffer Full Status bit, BF	64
Bus Arbitration	88
Bus Collision Section	88
Bus Collision During a RESTART Condition	91
Bus Collision During a Start Condition	89
Bus Collision During a Stop Condition	92
Bus Collision Interrupt Flag bit, BCLIF	24

### C

Capture/Compare/PWM	
Capture	
Block Diagram	59
CCP1CON Register	58
CCP1IF	59
Mode	59
Prescaler	59
CCP Timer Resources	57
Compare	
Block Diagram	60
Mode	60
Software Interrupt Mode	60
Special Event Trigger	60
Special Trigger Output of CCP1	60
Special Trigger Output of CCP2	60
Interaction of Two CCP Modules	57
Section	57
Special Event Trigger and A/D Conversions	60
Capture/Compare/PWM (CCP)	
CCP1	
RC2/CCP1 Pin	7, 8
CCP2	
RC1/T1OSI/CCP2 Pin	7, 8
PWM Block Diagram	60
PWM Mode	60
CCP1CON	17
CCP2CON	17
CCPR1H Register	15, 17, 57
CCPR1L Register	17, 57
CCPR2H Register	15, 17
CCPR2L Register	15, 17
CCPxM0 bit	58
CCPxM1 bit	58
CCPxM2 bit	58
CCPxM3 bit	58
CCPxX bit	58
CCPxY bit	58
CKE	64
CKP	65
Clock Polarity Select bit, CKP	65
Code Examples	
Call of a Subroutine in Page 1 from Page 0	26
Indirect Addressing	27
Code Protection	121, 135
Computed GOTO	26
Configuration Bits	121
Conversion Considerations	183

### D

D/A	64
Data Memory	12
Bank Select (RP1:RP0 Bits)	12, 18
General Purpose Registers	12
Register File Map	13, 14
Special Function Registers	15
Data/Address bit, D/A	64
DC Characteristics	154

# PIC16F87X

Development Support .....	145	I <sup>2</sup> C Module Address Register, SSPADD .....	72
Device Differences .....	183	I <sup>2</sup> C Slave Mode .....	72
Device Overview .....	5	ID Locations .....	121, 135
Direct Addressing .....	27, 28	In-Circuit Serial Programming (ICSP) .....	121, 136
<b>E</b>		INDF .....	17
Electrical Characteristics .....	151	INDF Register .....	15, 16, 27
Errata .....	4	Indirect Addressing .....	27, 28
<b>F</b>		FSR Register .....	12
Firmware Instructions .....	137	Instruction Format .....	137
FSR Register .....	15, 16, 17, 27	Instruction Set .....	137
<b>G</b>		ADDLW .....	139
General Call Address Sequence .....	74	ADDWF .....	139
General Call Address Support .....	74	ANDLW .....	139
General Call Enable bit .....	66	ANDWF .....	139
<b>I</b>		BCF .....	139
I/O Ports .....	29	BSF .....	139
I <sup>2</sup> C .....	71	BTFSC .....	140
I <sup>2</sup> C Master Mode Reception .....	83	BTFSS .....	140
I <sup>2</sup> C Master Mode Restart Condition .....	80	CALL .....	140
I <sup>2</sup> C Mode Selection .....	71	CLRF .....	140
I <sup>2</sup> C Module		CLRWF .....	140
Acknowledge Sequence timing .....	85	CLRWDT .....	140
Addressing .....	72	COMF .....	141
Baud Rate Generator .....	78	DECF .....	141
Block Diagram .....	76	DECFSZ .....	141
BRG Block Diagram .....	78	GOTO .....	141
BRG Reset due to SDA Collision .....	90	INCF .....	141
BRG Timing .....	78	INCFSZ .....	141
Bus Arbitration .....	88	IORLW .....	142
Bus Collision .....	88	IORWF .....	142
Acknowledge .....	88	MOVF .....	142
Restart Condition .....	91	MOVLW .....	142
Restart Condition Timing (Case1) .....	91	MOVWF .....	142
Restart Condition Timing (Case2) .....	91	NOP .....	142
Start Condition .....	89	RETFIE .....	143
Start Condition Timing .....	89, 90	RETLW .....	143
Stop Condition .....	92	RETURN .....	143
Stop Condition Timing (Case1) .....	92	RLF .....	143
Stop Condition Timing (Case2) .....	92	RRF .....	143
Transmit Timing .....	88	SLEEP .....	143
Bus Collision timing .....	88	SUBLW .....	144
Clock Arbitration .....	87	SUBWF .....	144
Clock Arbitration Timing (Master Transmit) .....	87	SWAPF .....	144
Conditions to not give ACK Pulse .....	72	XORLW .....	144
General Call Address Support .....	74	XORWF .....	144
Master Mode .....	76	Summary Table .....	138
Master Mode 7-bit Reception timing .....	84	INTCON .....	17
Master Mode Operation .....	77	INTCON Register .....	20
Master Mode Start Condition .....	79	GIE Bit .....	20
Master Mode Transmission .....	81	INTE Bit .....	20
Master Mode Transmit Sequence .....	77	INTF Bit .....	20
Multi-Master Communication .....	88	PEIE Bit .....	20
Multi-master Mode .....	77	RBIE Bit .....	20
Operation .....	71	RBIF Bit .....	20, 31
Repeat Start Condition timing .....	80	TOIE Bit .....	20
Slave Mode .....	72	TOIF Bit .....	20
Slave Reception .....	73	Inter-Integrated Circuit (I <sup>2</sup> C) .....	63
Slave Transmission .....	73	Internal Sampling Switch (Rss) Impedance .....	114
SSPBUF .....	72	Interrupt Sources .....	121, 131
Stop Condition Receive or Transmit timing .....	86	Block Diagram .....	131
Stop Condition timing .....	86	Interrupt on Change (RB7:RB4 ) .....	31
Waveforms for 7-bit Reception .....	73	RB0/INT Pin, External .....	7, 8, 132
Waveforms for 7-bit Transmission .....	74	TMR0 Overflow .....	132
		USART Receive/Transmit Complete .....	95

Interrupts	
Bus Collision Interrupt .....	24
Synchronous Serial Port Interrupt .....	22
Interrupts, Context Saving During .....	132
Interrupts, Enable Bits	
Global Interrupt Enable (GIE Bit) .....	20, 131
Interrupt on Change (RB7:RB4) Enable (RBIE Bit) .....	20, 132
Peripheral Interrupt Enable (PEIE Bit) .....	20
RB0/INT Enable (INTE Bit) .....	20
TMR0 Overflow Enable (T0IE Bit) .....	20
Interrupts, Flag Bits	
Interrupt on Change (RB7:RB4) Flag (RBIF Bit) .....	20, 31, 132
RB0/INT Flag (INTF Bit) .....	20
TMR0 Overflow Flag (T0IF Bit) .....	20, 132
<b>K</b>	
KeeLoq® Evaluation and Programming Tools .....	148
<b>L</b>	
Loading of PC .....	26
<b>M</b>	
Master Clear ( $\overline{\text{MCLR}}$ ) .....	7, 8
$\overline{\text{MCLR}}$ Reset, Normal Operation .....	125, 127, 128
$\overline{\text{MCLR}}$ Reset, SLEEP .....	125, 127, 128
Memory Organization	
Data Memory .....	12
Program Memory .....	11
MPLAB Integrated Development Environment Software ..	145
Multi-Master Communication .....	88
Multi-Master Mode .....	77
<b>O</b>	
OPCODE Field Descriptions .....	137
OPTION .....	17
OPTION_REG Register .....	19
INTEDG Bit .....	19
PS2:PS0 Bits .....	19
PSA Bit .....	19
$\overline{\text{RBP}}\overline{\text{U}}$ Bit .....	19
T0CS Bit .....	19
T0SE Bit .....	19
OSC1/CLKIN Pin .....	7, 8
OSC2/CLKOUT Pin .....	7, 8
Oscillator Configuration .....	121, 123
HS .....	123, 127
LP .....	123, 127
RC .....	123, 124, 127
XT .....	123, 127
Oscillator, WDT .....	133
Output of TMR2 .....	55
<b>P</b>	
P .....	64
Packaging .....	175
Paging, Program Memory .....	11, 26
Parallel Slave Port (PSP) .....	9, 35, 38
Block Diagram .....	38
RE0/ $\overline{\text{RD}}$ /AN5 Pin .....	9, 37, 38
RE1/ $\overline{\text{WR}}$ /AN6 Pin .....	9, 37, 38
RE2/ $\overline{\text{CS}}$ /AN7 Pin .....	9, 37, 38
Read Waveforms .....	39
Select (PSPMODE Bit) .....	35, 36, 38
Write Waveforms .....	39
PCL Register .....	15, 16, 17, 26
PCLATH Register .....	15, 16, 17, 26
PCON Register .....	17, 25, 126
$\overline{\text{BOR}}$ Bit .....	25
POR Bit .....	25
PIC16F876 Pinout Description .....	7
PICDEM-1 Low-Cost PICmicro Demo Board .....	147
PICDEM-2 Low-Cost PIC16CXX Demo Board .....	147
PICDEM-3 Low-Cost PIC16CXXX Demo Board .....	147
PICSTART® Plus Entry Level Development System .....	147
PIE1 Register .....	17, 21
PIE2 Register .....	17, 23
Pinout Descriptions	
PIC16F873/PIC16F876 .....	7
PIC16F874/PIC16F877 .....	8
PIR1 Register .....	22
PIR2 Register .....	24
POP .....	26
PORTA .....	7, 8, 17
Analog Port Pins .....	7, 8
Initialization .....	29
PORTA Register .....	29
RA3, RA0 and RA5 Port Pins .....	29
RA4/T0CKI Pin .....	7, 8, 29
RA5/ $\overline{\text{SS}}$ /AN4 Pin .....	7, 8
TRISA Register .....	29
PORTA Register .....	15
PORTB .....	7, 8, 17
PORTB Register .....	31
Pull-up Enable ( $\overline{\text{RBP}}\overline{\text{U}}$ Bit) .....	19
RB0/INT Edge Select (INTEDG Bit) .....	19
RB0/INT Pin, External .....	7, 8, 132
RB3:RB0 Port Pins .....	31
RB7:RB4 Interrupt on Change .....	132
RB7:RB4 Interrupt on Change Enable (RBIE Bit) .....	20, 132
RB7:RB4 Interrupt on Change Flag (RBIF Bit) .....	20, 31, 132
RB7:RB4 Port Pins .....	31
TRISB Register .....	31
PORTB Register .....	15
PORTC .....	7, 8, 17
Block Diagram .....	33
PORTC Register .....	33
RC0/T1OSO/T1CKI Pin .....	7, 8
RC1/T1OSI/CCP2 Pin .....	7, 8
RC2/CCP1 Pin .....	7, 8
RC3/SCK/SCL Pin .....	7, 8
RC4/SDI/SDA Pin .....	7, 8
RC5/SDO Pin .....	7, 8
RC6/TX/CK Pin .....	7, 8, 96
RC7/RX/DT Pin .....	7, 8, 96, 97
TRISC Register .....	33, 95
PORTC Register .....	15
PORTD .....	9, 17, 38
Block Diagram .....	35
Parallel Slave Port (PSP) Function .....	35
PORTD Register .....	35
TRISD Register .....	35
PORTD Register .....	15
PORTE .....	9, 17
Analog Port Pins .....	9, 37, 38
Block Diagram .....	36
Input Buffer Full Status (IBF Bit) .....	36
Input Buffer Overflow (IBOV Bit) .....	36
Output Buffer Full Status (OBF Bit) .....	36
PORTE Register .....	36

# PIC16F87X

PSP Mode Select (PSPMODE Bit)	35, 36, 38
RE0/RD/AN5 Pin	9, 37, 38
RE1/WR/AN6 Pin	9, 37, 38
RE2/CS/AN7 Pin	9, 37, 38
TRISE Register	36
PORTE Register	15
Postscaler, WDT	
Assignment (PSA Bit)	19
Rate Select (PS2:PS0 Bits)	19
Power-on Reset (POR)	121, 125, 126, 127, 128
Oscillator Start-up Timer (OST)	121, 126
POR Status ( $\overline{\text{POR}}$ Bit)	25
Power Control (PCON) Register	126
Power-down (PD Bit)	18, 125
Power-up Timer (PWRT)	121, 126
Time-out ( $\overline{\text{TO}}$ Bit)	18, 125
Time-out Sequence on Power-up	129, 130
PR2	17
PR2 Register	16, 55
Prescaler, Timer0	
Assignment (PSA Bit)	19
Rate Select (PS2:PS0 Bits)	19
PRO MATE® II Universal Programmer	147
Product Identification System	191
Program Counter	
Reset Conditions	127
Program Memory	11
Interrupt Vector	11
Paging	11, 26
Program Memory Map	11
Reset Vector	11
Program Verification	135
Programming Pin (VPP)	7, 8
Programming, Device Instructions	137
PUSH	26
<b>R</b>	
R/W	64
R/W bit	72
R/W bit	73
RCREG	17
RCSTA Register	17, 96
CREN Bit	96
FERR Bit	96
OERR Bit	96
RX9 Bit	96
RX9D Bit	96
SPEN Bit	95, 96
SREN Bit	96
Read/Write bit, R/W	64
Receive Enable bit	66
Receive Overflow Indicator bit, SSPOV	65
Register File	12
Register File Map	13, 14
Registers	
FSR Summary	17
INDF Summary	17
INTCON Summary	17
OPTION Summary	17
PCL Summary	17
PCLATH Summary	17
PORTB Summary	17
SSPSTAT	64
STATUS Summary	17
TMR0 Summary	17
TRISB Summary	17
Reset	121, 125
Block Diagram	125
Reset Conditions for All Registers	128
Reset Conditions for PCON Register	127
Reset Conditions for Program Counter	127
Reset Conditions for STATUS Register	127
Restart Condition Enabled bit	66
Revision History	183
<b>S</b>	
SCK	67
SCL	72
SDA	72
SDI	67
SDO	67
SEEVAL® Evaluation and Programming System	148
Serial Clock, SCK	67
Serial Clock, SCL	72
Serial Data Address, SDA	72
Serial Data In, SDI	67
Serial Data Out, SDO	67
Slave Select, $\overline{\text{SS}}$	67
SLEEP	121, 125, 134
SMP	64
Software Simulator (MPLAB-SIM)	146
SPBRG	17
SPBRG Register	16
Special Features of the CPU	121
Special Function Registers	15
Speed, Operating	1
SPI	
Master Mode	68
Master Mode Timing	68
Serial Clock	67
Serial Data In	67
Serial Data Out	67
Serial Peripheral Interface (SPI)	63
Slave Mode Timing	69
Slave Mode Timing Diagram	69
Slave Select	67
SPI clock	68
SPI Mode	67
SPI Clock Edge Select, CKE	64
SPI Data Input Sample Phase Select, SMP	64
SPI Module	
Slave Mode	69
$\overline{\text{SS}}$	67
SSP	63
Block Diagram (SPI Mode)	67
RA5/ $\overline{\text{SS}}$ /AN4 Pin	7, 8
RC3/SCK/SCL Pin	7, 8
RC4/SDI/SDA Pin	7, 8
RC5/SDO Pin	7, 8
SPI Mode	67
SSPADD	72
SSPBUF	68, 72
SSPCON1	65
SSPCON2	66
SSPSR	68, 72
SSPSTAT	64, 72
SSP I <sup>2</sup> C	
SSP I <sup>2</sup> C Operation	71

SSP Module		Timers	
SPI Master Mode	68	Timer0	
SPI Slave Mode	69	External Clock	48
SSPCON1 Register	71	Interrupt	47
SSP Overflow Detect bit, SSPOV	72	Prescaler	48
SSPADD Register	16, 17	Prescaler Block Diagram	47
SSPBUF	17, 72	Section	47
SSPBUF Register	15	T0CKI	48
SSPCON Register	15	Timer1	
SSPCON1	65, 71	Asynchronous Counter Mode	53
SSPCON2	66	Capacitor Selection	53
SSPEN	65	Operation in Timer Mode	52
SSPIF	22, 73	Oscillator	53
SSPM3:SSPM0	65	Prescaler	53
SSPOV	65, 72, 83	Resetting of Timer1 Registers	53
SSPSTAT	64, 72	Resetting Timer1 using a CCP Trigger Output	53
SSPSTAT Register	16, 17	Synchronized Counter Mode	52
Stack	26	T1CON	51
Overflows	26	TMR1H	53
Underflow	26	TMR1L	53
Start bit (S)	64	Timer2	
Start Condition Enabled bit	66	Block Diagram	55
STATUS Register	17, 18	Postscaler	55
C Bit	18	Prescaler	55
DC Bit	18	T2CON	55
IRP Bit	18	Timing Diagrams	
$\overline{PD}$ Bit	18, 125	A/D Conversion	172
RP1:RP0 Bits	18	Acknowledge Sequence Timing	85
TO Bit	18, 125	Baud Rate Generator with Clock Arbitration	78
Z Bit	18	BRG Reset Due to SDA Collision	90
Stop bit (P)	64	Brown-out Reset	162
Stop Condition Enable bit	66	Bus Collision	
Synchronous Serial Port	63	Start Condition Timing	89
Synchronous Serial Port Enable bit, SSPEN	65	Bus Collision During a Restart Condition (Case 1)	91
Synchronous Serial Port Interrupt	22	Bus Collision During a Restart Condition (Case2)	91
Synchronous Serial Port Mode Select bits,		Bus Collision During a Start Condition (SCL = 0)	90
SSPM3:SSPM0	65	Bus Collision During a Stop Condition	92
<b>T</b>		Bus Collision for Transmit and Acknowledge	88
T1CKPS0 bit	51	Capture/Compare/PWM	164
T1CKPS1 bit	51	CLKOUT and I/O	161
T1CON	17	I <sup>2</sup> C Bus Data	169
T1CON Register	17, 51	I <sup>2</sup> C Bus Start/Stop bits	168
T1OSCEN bit	51	I <sup>2</sup> C Master Mode First Start bit timing	79
$\overline{T1SYNC}$ bit	51	I <sup>2</sup> C Master Mode Reception timing	84
T2CKPS0 bit	55	I <sup>2</sup> C Master Mode Transmission timing	82
T2CKPS1 bit	55	Master Mode Transmit Clock Arbitration	87
T2CON Register	17, 55	Power-up Timer	162
TAD	116	Repeat Start Condition	80
Timer0		Reset	162
Clock Source Edge Select (T0SE Bit)	19	SPI Master Mode	68
Clock Source Select (T0CS Bit)	19	SPI Slave Mode (CKE = 1)	69
Overflow Enable (T0IE Bit)	20	SPI Slave Mode Timing (CKE = 0)	69
Overflow Flag (T0IF Bit)	20, 132	Start-up Timer	162
Overflow Interrupt	132	Stop Condition Receive or Transmit	86
RA4/T0CKI Pin, External Clock	7, 8	Time-out Sequence on Power-up	129, 130
Timer1	51	Timer0	163
RC0/T1OSO/T1CKI Pin	7, 8	Timer1	163
RC1/T1OSI/CCP2 Pin	7, 8	USART Asynchronous Master Transmission	100
		USART Asynchronous Reception	101
		USART Synchronous Receive	170
		USART Synchronous Reception	107
		USART Synchronous Transmission	106, 170
		USART, Asynchronous Reception	104
		Wake-up from SLEEP via Interrupt	135
		Watchdog Timer	162

# PIC16F87X

TMR0	17
TMR0 Register	15
TMR1CS bit	51
TMR1H	17
TMR1H Register	15
TMR1L	17
TMR1L Register	15
TMR1ON bit	51
TMR2	17
TMR2 Register	15
TMR2ON bit	55
TOUTPS0 bit	55
TOUTPS1 bit	55
TOUTPS2 bit	55
TOUTPS3 bit	55
TRISA	17
TRISA Register	16
TRISB	17
TRISB Register	16
TRISC	17
TRISC Register	16
TRISD	17
TRISD Register	16
TRISE	17
TRISE Register	16, 36
IBF Bit	36
IBOV Bit	36
OBF Bit	36
PSPMODE Bit	35, 36, 38
TXREG	17
TXSTA	17
TXSTA Register	95
BRGH Bit	95
CSRC Bit	95
SYNC Bit	95
TRMT Bit	95
TX9 Bit	95
TX9D Bit	95
TXEN Bit	95

## U

UA	64
Universal Synchronous Asynchronous Receiver Transmitter (USART)	
Asynchronous Receiver	
Setting Up Reception	103
Timing Diagram	104
Update Address, UA	64
USART	95
Asynchronous Mode	99
Receive Block Diagram	103
Asynchronous Receiver	101
Asynchronous Reception	102
Asynchronous Transmitter	99
Baud Rate Generator (BRG)	97
Baud Rate Formula	97
Baud Rates, Asynchronous Mode (BRGH=0)	98
High Baud Rate Select (BRGH Bit)	95
Sampling	97
Clock Source Select (CSRC Bit)	95
Continuous Receive Enable (CREN Bit)	96
Framing Error (FERR Bit)	96
Mode Select (SYNC Bit)	95
Overrun Error (OERR Bit)	96
RC6/TX/CK Pin	7, 8
RC7/RX/DT Pin	7, 8

RCSTA Register	96
Receive Block Diagram	101
Receive Data, 9th bit (RX9D Bit)	96
Receive Enable, 9-bit (RX9 Bit)	96
Serial Port Enable (SPEN Bit)	95, 96
Single Receive Enable (SREN Bit)	96
Synchronous Master Mode	105
Synchronous Master Reception	107
Synchronous Master Transmission	105
Synchronous Slave Mode	108
Transmit Block Diagram	99
Transmit Data, 9th Bit (TX9D)	95
Transmit Enable (TXEN Bit)	95
Transmit Enable, Nine-bit (TX9 Bit)	95
Transmit Shift Register Status (TRMT Bit)	95
TXSTA Register	95

## W

Wake-up from SLEEP	121, 134
Interrupts	127, 128
MCLR Reset	128
Timing Diagram	135
WDT Reset	128
Watchdog Timer (WDT)	121, 133
Block Diagram	133
Enable (WDTE Bit)	133
Programming Considerations	133
RC Oscillator	133
Time-out Period	133
WDT Reset, Normal Operation	125, 127, 128
WDT Reset, SLEEP	125, 127, 128
Waveform for General Call Address Sequence	74
WCOL	65, 79, 81, 83, 85, 86
WCOL Status Flag	79
Write Collision Detect bit, WCOL	65
WWW, On-Line Support	4

## 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](http://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

## Systems Information and Upgrade Hot Line

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits. The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada, and  
1-480-786-7302 for the rest of the world.

981103

Trademarks: The Microchip name, logo, PIC, PICmicro, PICSTART, PICMASTER, PRO MATE and MPLAB are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. *FlexROM* and *fuzzyLAB* are trademarks and SQTP is a service mark of Microchip in the U.S.A. All other trademarks mentioned herein are the property of their respective companies.

# PIC16F87X

---

## READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 786-7578.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager Total Pages Sent  
RE: Reader Response  
From: Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City / State / ZIP / Country \_\_\_\_\_  
Telephone: (\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_ FAX: (\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_

Application (optional):

Would you like a reply? \_\_\_Y \_\_\_N

Device: **PIC16F87X** Literature Number: **DS30292B**

Questions:

1. What are the best features of this document?

---

---

2. How does this document meet your hardware and software development needs?

---

---

3. Do you find the organization of this data sheet easy to follow? If not, why?

---

---

4. What additions to the data sheet do you think would enhance the structure and subject?

---

---

5. What deletions from the data sheet could be made without affecting the overall usefulness?

---

---

6. Is there any incorrect or misleading information (what and where)?

---

---

7. How would you improve this document?

---

---

8. How would you improve our software, systems, and silicon products?

---

---

## PIC16F87X PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

<u>PART NO.</u>	<u>-XX</u>	<u>X</u>	<u>/XX</u>	<u>XXX</u>
Device	Frequency Range	Temperature Range	Package	Pattern
Device	PIC16F87X <sup>(1)</sup> , PIC16F87XT <sup>(2)</sup> ; VDD range 4.0V to 5.5V PIC16LF87X <sup>(1)</sup> , PIC16LF87XT <sup>(2)</sup> ; VDD range 2.0V to 5.5V			
Frequency Range	04 = 4 MHz 20 = 20 MHz			
Temperature Range	b <sup>(3)</sup> = 0°C to 70°C (Commercial) I = -40°C to +85°C (Industrial)			
Package	PQ = MQFP (Metric PQFP) PT = TQFP (Thin Quad Flatpack) SO = SOIC SP = Skinny plastic dip P = PDIP L = PLCC			
Pattern	QTP, SQTP, Code or Special Requirements (blank otherwise)			

**Examples:**

f) PIC16F877 -20/P 301 = Commercial temp., PDIP package, 4 MHz, normal VDD limits, QTP pattern #301.

g) PIC16F876 - 04I/SO = Industrial temp., SOIC package, 200 kHz, Extended VDD limits.

h) PIC16F877 - 04I/P = Industrial temp., PDIP package, 10MHz, normal VDD limits.

**Note 1:** F = CMOS FLASH  
LF = Low Power CMOS FLASH  
T = in tape and reel - SOIC, PLCC, MQFP, TQFP packages only.

\* JW Devices are UV erasable and can be programmed to any device configuration. JW Devices meet the electrical requirement of each oscillator type (including LC devices).

### Sales and Support

#### Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 786-7277
3. The Microchip Worldwide Site ([www.microchip.com](http://www.microchip.com))

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

#### New Customer Notification System

Register on our web site ([www.microchip.com/cn](http://www.microchip.com/cn)) to receive the most current information on our products.

# PIC16F87X

---

NOTES:

NOTES:

# PIC16F87X

---

NOTES:

NOTES:

# PIC16F87X

---

NOTES:

NOTES:



## WORLDWIDE SALES AND SERVICE

### AMERICAS

#### Corporate Office

Microchip Technology Inc.  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-786-7200 Fax: 480-786-7277  
Technical Support: 480-786-7627  
Web Address: <http://www.microchip.com>

#### Atlanta

Microchip Technology Inc.  
500 Sugar Mill Road, Suite 200B  
Atlanta, GA 30350  
Tel: 770-640-0034 Fax: 770-640-0307

#### Boston

Microchip Technology Inc.  
5 Mount Royal Avenue  
Marlborough, MA 01752  
Tel: 508-480-9990 Fax: 508-480-8575

#### Chicago

Microchip Technology Inc.  
333 Pierce Road, Suite 180  
Itasca, IL 60143  
Tel: 630-285-0071 Fax: 630-285-0075

#### Dallas

Microchip Technology Inc.  
4570 Westgrove Drive, Suite 160  
Addison, TX 75248  
Tel: 972-818-7423 Fax: 972-818-2924

#### Dayton

Microchip Technology Inc.  
Two Prestige Place, Suite 150  
Miamisburg, OH 45342  
Tel: 937-291-1654 Fax: 937-291-9175

#### Detroit

Microchip Technology Inc.  
Tri-Atria Office Building  
32255 Northwestern Highway, Suite 190  
Farmington Hills, MI 48334  
Tel: 248-538-2250 Fax: 248-538-2260

#### Los Angeles

Microchip Technology Inc.  
18201 Von Karman, Suite 1090  
Irvine, CA 92612  
Tel: 949-263-1888 Fax: 949-263-1338

#### New York

Microchip Technology Inc.  
150 Motor Parkway, Suite 202  
Hauppauge, NY 11788  
Tel: 631-273-5305 Fax: 631-273-5335

#### San Jose

Microchip Technology Inc.  
2107 North First Street, Suite 590  
San Jose, CA 95131  
Tel: 408-436-7950 Fax: 408-436-7955

### AMERICAS (continued)

#### Toronto

Microchip Technology Inc.  
5925 Airport Road, Suite 200  
Mississauga, Ontario L4V 1W1, Canada  
Tel: 905-405-6279 Fax: 905-405-6253

### ASIA/PACIFIC

#### Hong Kong

Microchip Asia Pacific  
Unit 2101, Tower 2  
Metroplaza  
223 Hing Fong Road  
Kwai Fong, N.T., Hong Kong  
Tel: 852-2-401-1200 Fax: 852-2-401-3431

#### Beijing

Microchip Technology, Beijing  
Unit 915, 6 Chaoyangmen Bei Dajie  
Dong Erhuan Road, Dongcheng District  
New China Hong Kong Manhattan Building  
Beijing 100027 PRC  
Tel: 86-10-85282100 Fax: 86-10-85282104

#### India

Microchip Technology Inc.  
India Liaison Office  
No. 6, Legacy, Convent Road  
Bangalore 560 025, India  
Tel: 91-80-229-0061 Fax: 91-80-229-0062

#### Japan

Microchip Technology Intl. Inc.  
Benex S-1 6F  
3-18-20, Shinyokohama  
Kohoku-Ku, Yokohama-shi  
Kanagawa 222-0033 Japan  
Tel: 81-45-471-6166 Fax: 81-45-471-6122

#### Korea

Microchip Technology Korea  
168-1, Youngbo Bldg. 3 Floor  
Samsung-Dong, Kangnam-Ku  
Seoul, Korea  
Tel: 82-2-554-7200 Fax: 82-2-558-5934

#### Shanghai

Microchip Technology  
RM 406 Shanghai Golden Bridge Bldg.  
2077 Yan'an Road West, Hong Qiao District  
Shanghai, PRC 200335  
Tel: 86-21-6275-5700 Fax: 86 21-6275-5060

### ASIA/PACIFIC (continued)

#### Singapore

Microchip Technology Singapore Pte Ltd.  
200 Middle Road  
#07-02 Prime Centre  
Singapore 188980  
Tel: 65-334-8870 Fax: 65-334-8850

#### Taiwan, R.O.C

Microchip Technology Taiwan  
10F-1C 207  
Tung Hua North Road  
Taipei, Taiwan, ROC  
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

### EUROPE

#### United Kingdom

Arizona Microchip Technology Ltd.  
505 Eskdale Road  
Winkersham Triangle  
Wokingham  
Berkshire, England RG41 5TU  
Tel: 44 118 921 5858 Fax: 44-118 921-5835

#### Denmark

Microchip Technology Denmark ApS  
Regus Business Centre  
Lautrup hof 1-3  
Ballerup DK-2750 Denmark  
Tel: 45 4420 9895 Fax: 45 4420 9910

#### France

Arizona Microchip Technology SARL  
Parc d'Activite du Moulin de Massy  
43 Rue du Saule Trapu  
Batiment A - 1er Etage  
91300 Massy, France  
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

#### Germany

Arizona Microchip Technology GmbH  
Gustav-Heinemann-Ring 125  
D-81739 München, Germany  
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

#### Italy

Arizona Microchip Technology SRL  
Centro Direzionale Colleoni  
Palazzo Taurus 1 V. Le Colleoni 1  
20041 Agrate Brianza  
Milan, Italy  
Tel: 39-039-65791-1 Fax: 39-039-6899883

11/15/99



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and water fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOC® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.

All rights reserved. © 1999 Microchip Technology Incorporated. Printed in the USA. 11/99 Printed on recycled paper.

Information contained in this publication regarding device applications and the like is intended for suggestion only and may be superseded by updates. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights. The Microchip logo and name are registered trademarks of Microchip Technology Inc. in the U.S.A. and other countries. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.