



MICROCHIP

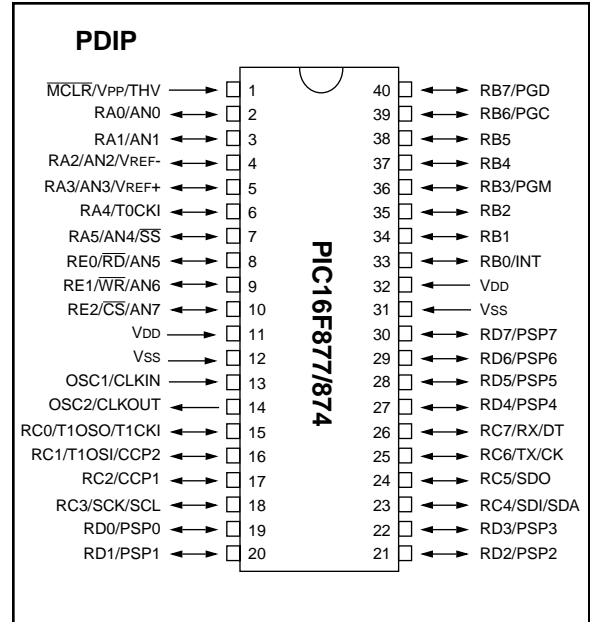
# PIC16F87X

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

### 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 PIC16C73/74/76/77**
- Interrupt capability (up to 14 internal/external interrupt 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™ via two pins
- ★ • Only single 5V source needed for programming
- ★ • 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

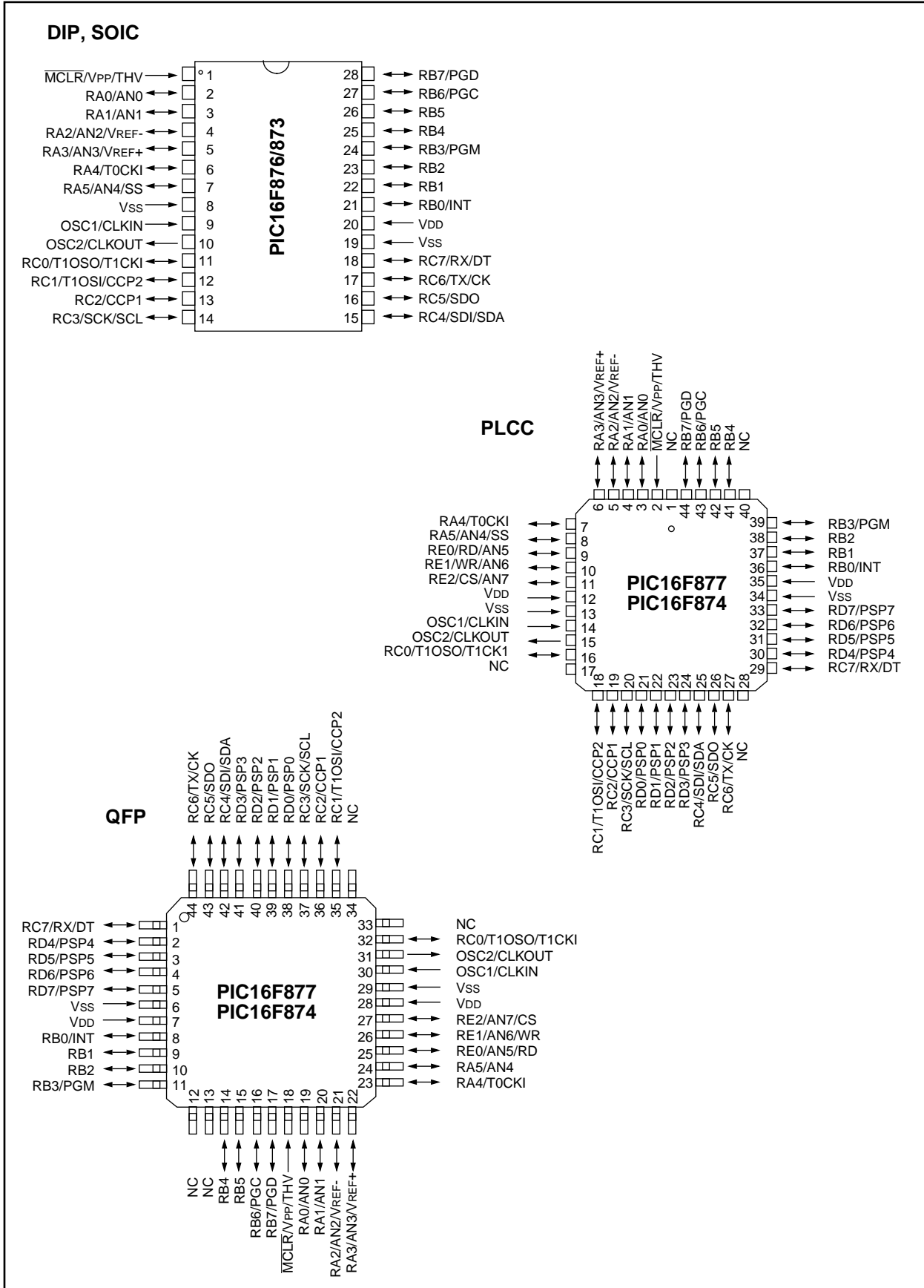


### Peripheral Features:

- 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	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	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 .....	49
7.0	Timer2 Module .....	53
8.0	Capture/Compare/PWM (CCP) Module(s) .....	55
9.0	Master Synchronous Serial Port (MSSP) Module .....	61
10.0	Universal Synchronous Asynchronous Receiver Transmitter (USART).....	105
11.0	Analog-to-Digital Converter (A/D) Module.....	121
12.0	Special Features of the CPU.....	133
13.0	Instruction Set Summary .....	151
14.0	Development Support.....	153
15.0	Electrical Characteristics .....	157
16.0	DC and AC Characteristics Graphs and Tables.....	177
17.0	Packaging Information.....	179
Appendix A:	Revision History.....	187
Appendix B:	Device Differences.....	187
Appendix C:	Conversion Considerations.....	187
Index .....		191

### *To Our Valued Customers*

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

To obtain the most up-to-date version of this data sheet, please check 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.

#### **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: (602) 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.

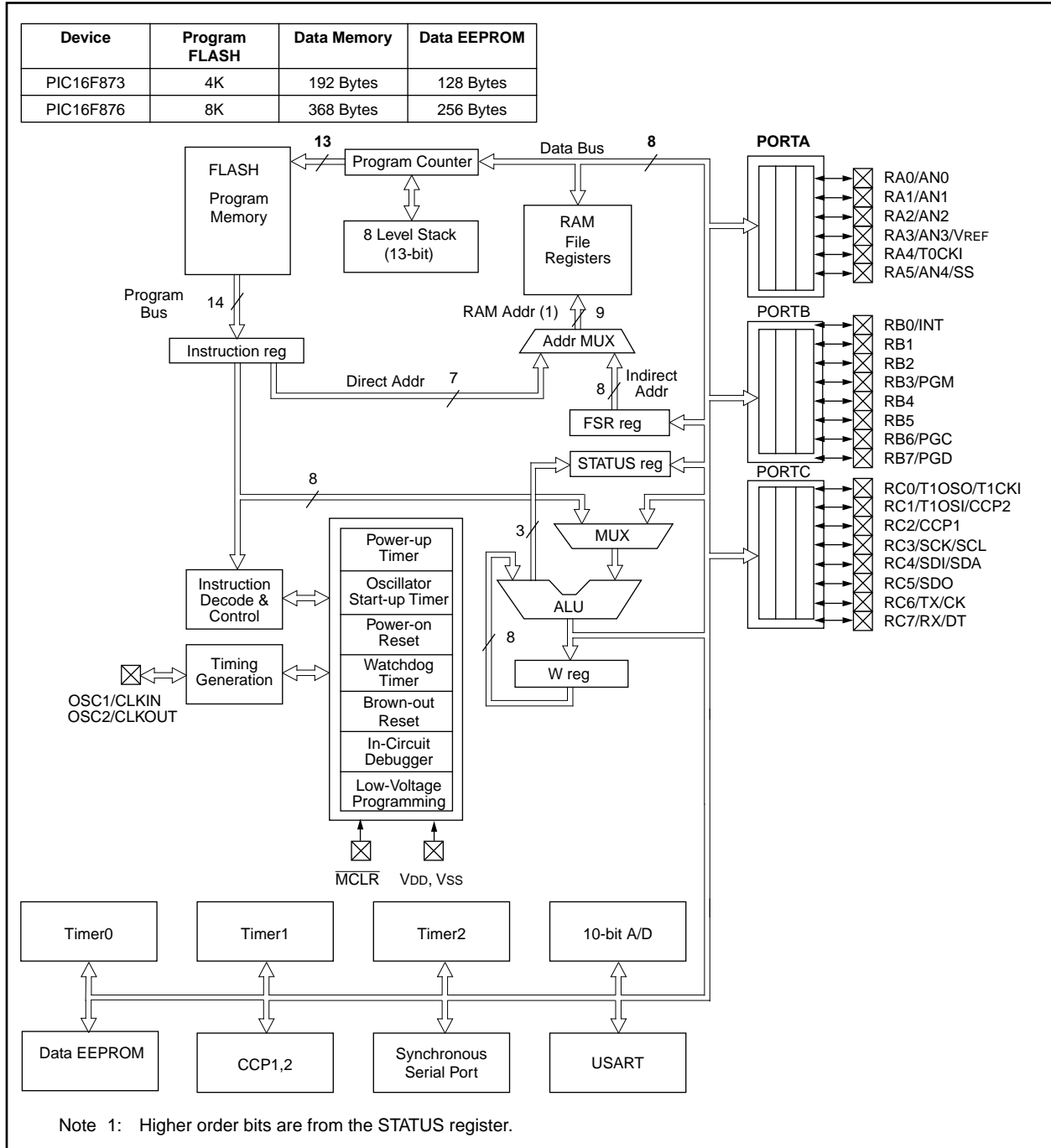
## 1.0 DEVICE OVERVIEW

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.

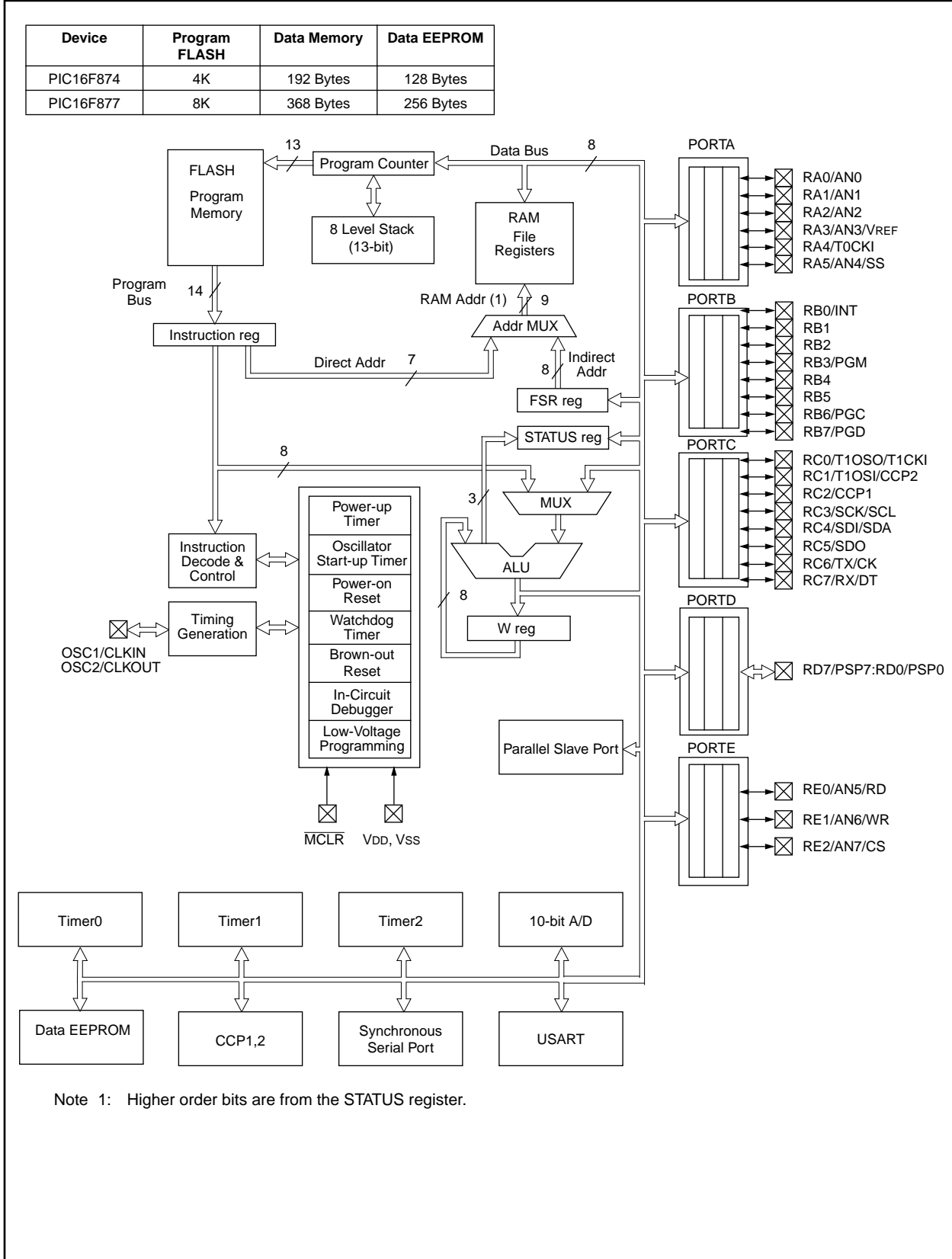
The following two figures are device block diagrams sorted by pin number; 28-pin for Figure 1-1 and 40-pin for Figure 1-2. The 28-pin and 40-pin pinouts are listed in Table 1-1 and Table 1-2, respectively.

**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	<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 module. 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	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>	<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	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	<p>PORTC is a bi-directional I/O port.</p> <p>RC0 can also be the Timer1 oscillator output or 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	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 CLKOUT 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	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 timer/counter. Output is open drain type. RA5 can also be analog input4 or the slave select for the synchronous serial port.
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>	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	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>	

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 (Cont.'d)**

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	PORTC is a bi-directional I/O port. RC0 can also be the Timer1 oscillator output or a Timer1 clock input.
RC1/T1OSI/CCP2	16	18	35	I/O	ST	RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	17	19	36	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	18	20	37	I/O	ST	RC3 can also be the synchronous serial clock input/output for both SPI and I <sup>2</sup> C modes.
RC4/SDI/SDA	23	25	42	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I <sup>2</sup> C mode).
RC5/SDO	24	26	43	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6/TX/CK	25	27	44	I/O	ST	RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	26	29	1	I/O	ST	RC7 can also be the USART Asynchronous Receive or Synchronous Data.
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/RD $\bar{D}$ /AN5	8	9	25	I/O	ST/TTL <sup>(3)</sup>	PORTE is a bi-directional I/O port. RE0 can also be read control for the parallel slave port, or analog input5.
RE1/W $\bar{R}$ /AN6	9	10	26	I/O	ST/TTL <sup>(3)</sup>	RE1 can also be write control for the parallel slave port, or analog input6.
RE2/C $\bar{S}$ /AN7	10	11	27	I/O	ST/TTL <sup>(3)</sup>	RE2 can also be select control for the parallel slave port, or analog input7.
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.  
 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.

# PIC16F87X

---

NOTES:

## 2.0 MEMORY ORGANIZATION

There are three memory blocks in each of these PIC-micros. 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](#).

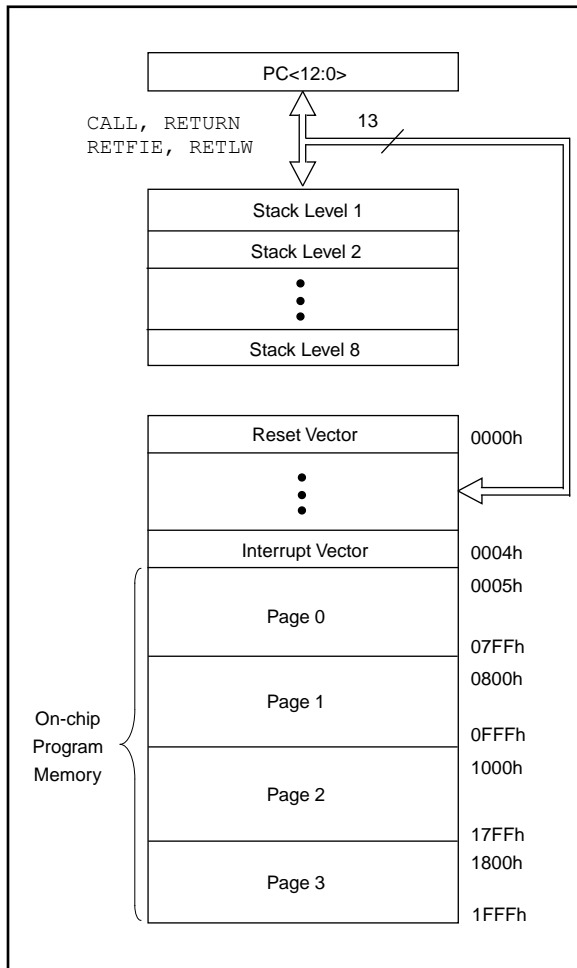
Additional information on device memory may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

### 2.1 Program Memory Organization

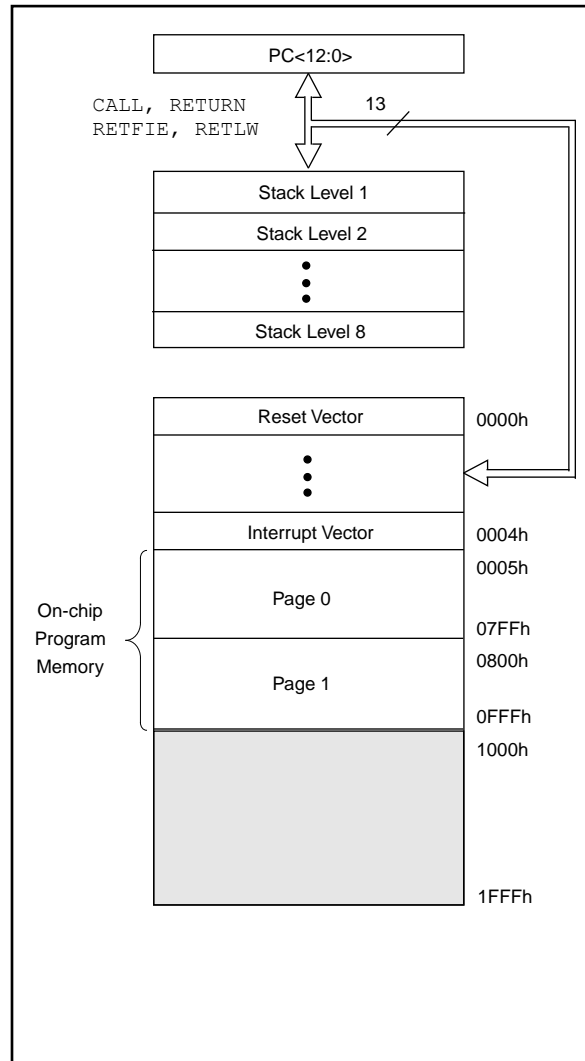
The PIC16F87X PICmicros have a 13-bit program counter capable of addressing an 8K x 14 program memory space. The PIC16F877/876 devices have 8K x 14 words of FLASH program memory and the PIC16F873/874 devices have 4K x 14. Accessing a location above the physically implemented address will cause a wraparound.

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

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



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



# 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 and RP0 are the bank select bits.

RP1	RP0	(STATUS<6:5>)
-----	-----	---------------

- = 00 → Bank0
- = 01 → Bank1
- = 10 → Bank2
- = 11 → Bank3

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 7.0 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**

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

# PIC16F87X

**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	T0IE	INTE	RBIE	T0IF	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	—	(6)	—	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	T1OSCEN	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	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.



**TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (Cont'd)**

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	T0IE	INTE	RBIE	T0IF	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	—	(6)	—	EEIE	BCLIE	—	—	CCP2IE	-r-0 0--0	-r-0 0--0	
8Eh	PCON	—	—	—	—	—	—	POR	BOR	---- -rq	---- -uu	
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  $\overline{MCLR}$  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 (Cont'd)**

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	$\overline{TO}$	$\overline{PD}$	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	$\overline{RBPU}$	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	$\overline{TO}$	$\overline{PD}$	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.

## 2.2.2.1 STATUS REGISTER

The STATUS register, shown in Figure 2-5, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

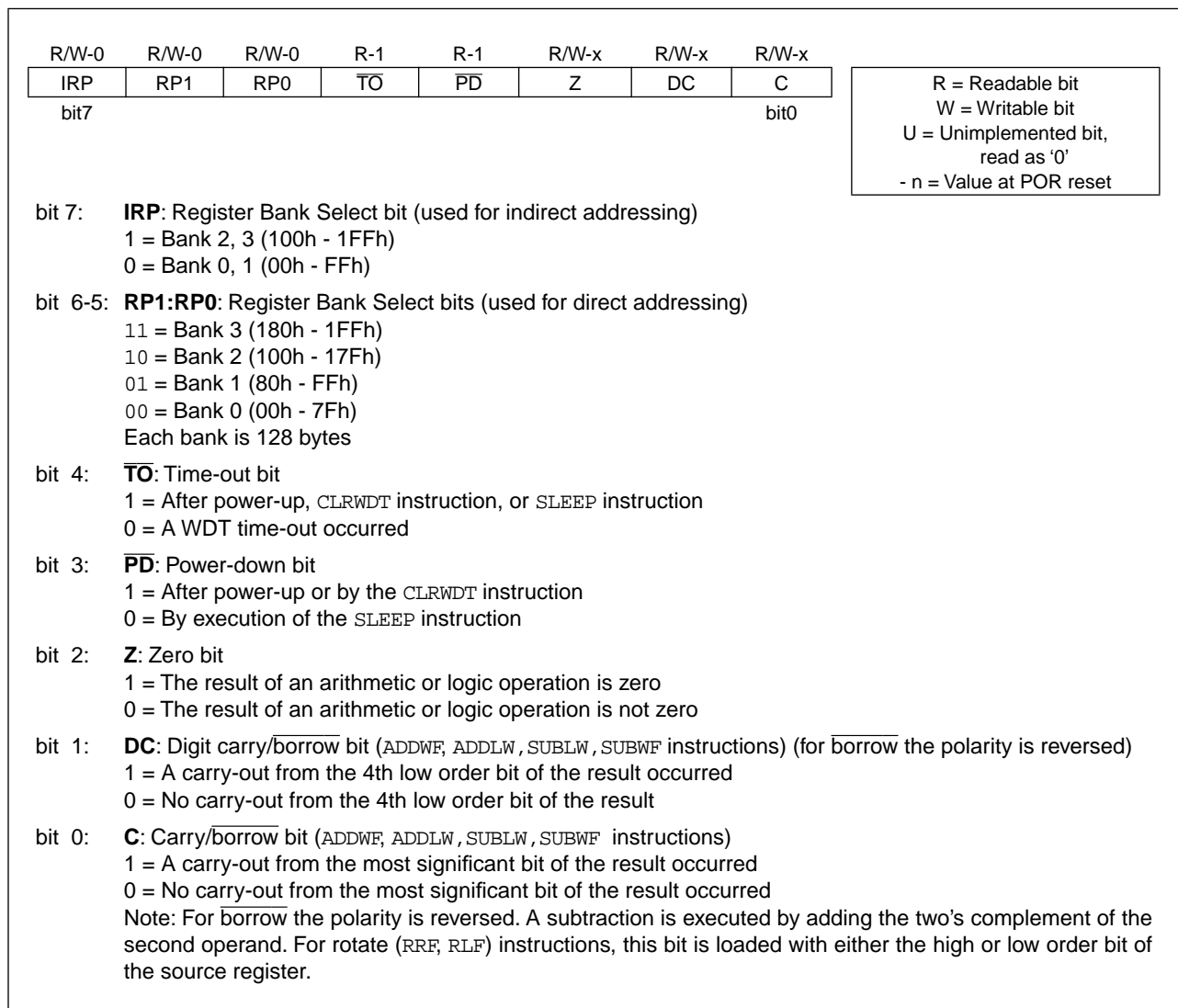
The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, 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.

**FIGURE 2-5: STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)**



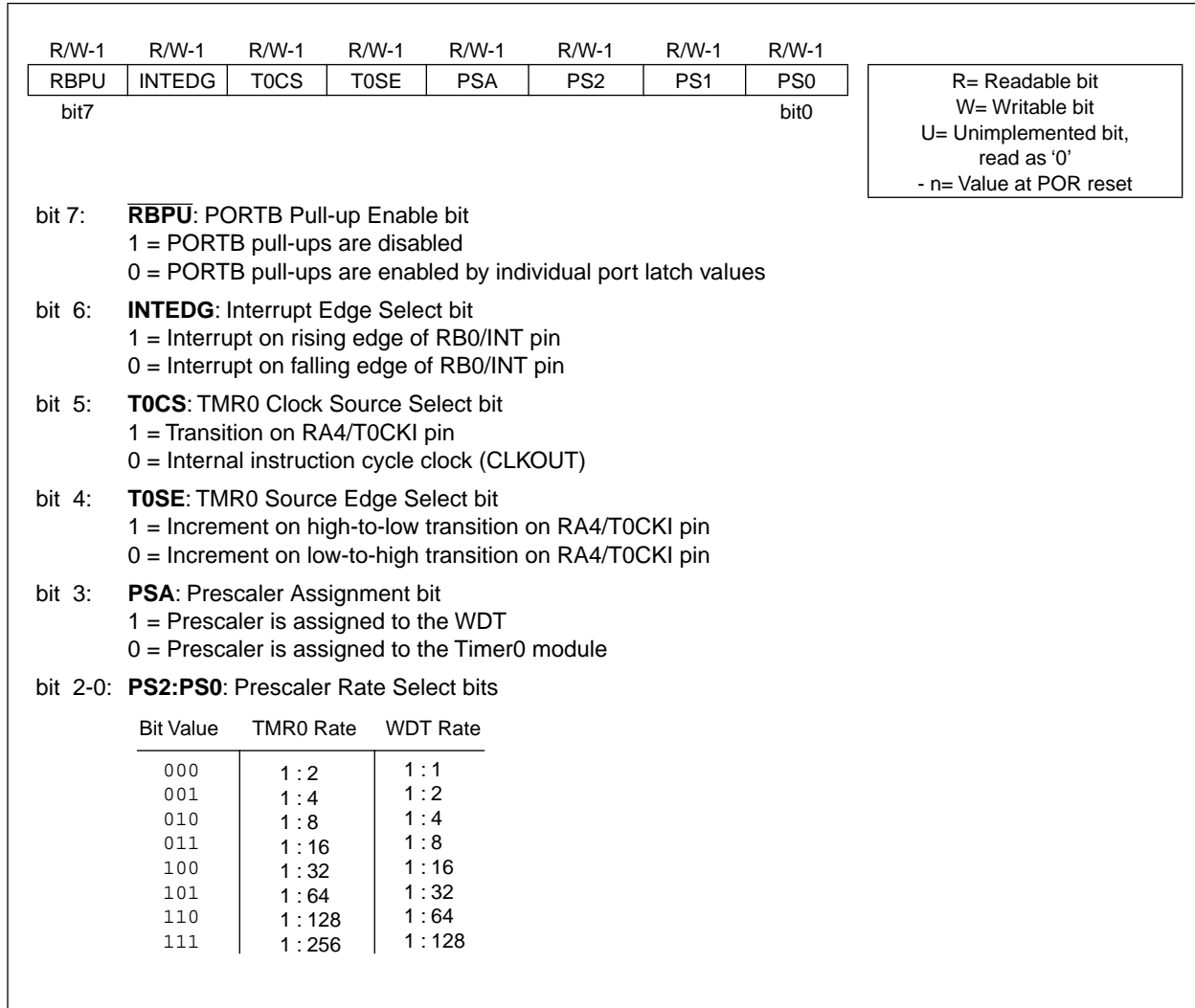
# PIC16F87X

## 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.

**FIGURE 2-6: OPTION\_REG REGISTER (ADDRESS 81h, 181h)**



## 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.

**FIGURE 2-7: 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-x
	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF
bit7							bit0

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

bit 7: **GIE:** Global Interrupt Enable bit  
 1 = Enables all un-masked interrupts  
 0 = Disables all interrupts

bit 6: **PEIE:** Peripheral Interrupt Enable bit  
 1 = Enables all un-masked peripheral interrupts  
 0 = Disables all peripheral interrupts

bit 5: **TOIE:** TMR0 Overflow Interrupt Enable bit  
 1 = Enables the TMR0 interrupt  
 0 = Disables the TMR0 interrupt

bit 4: **INTE:** RB0/INT External Interrupt Enable bit  
 1 = Enables the RB0/INT external interrupt  
 0 = Disables the RB0/INT external interrupt

bit 3: **RBIE:** RB Port Change Interrupt Enable bit  
 1 = Enables the RB port change interrupt  
 0 = Disables the RB port change interrupt

bit 2: **TOIF:** TMR0 Overflow Interrupt Flag bit  
 1 = TMR0 register has overflowed (must be cleared in software)  
 0 = TMR0 register did not overflow

bit 1: **INTF:** 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

bit 0: **RBIF:** 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

# PIC16F87X

## 2.2.2.4 PIE1 REGISTER

This register contains the individual enable bits for the peripheral interrupts.

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

**FIGURE 2-8: 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

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

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

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

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

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

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

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

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

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

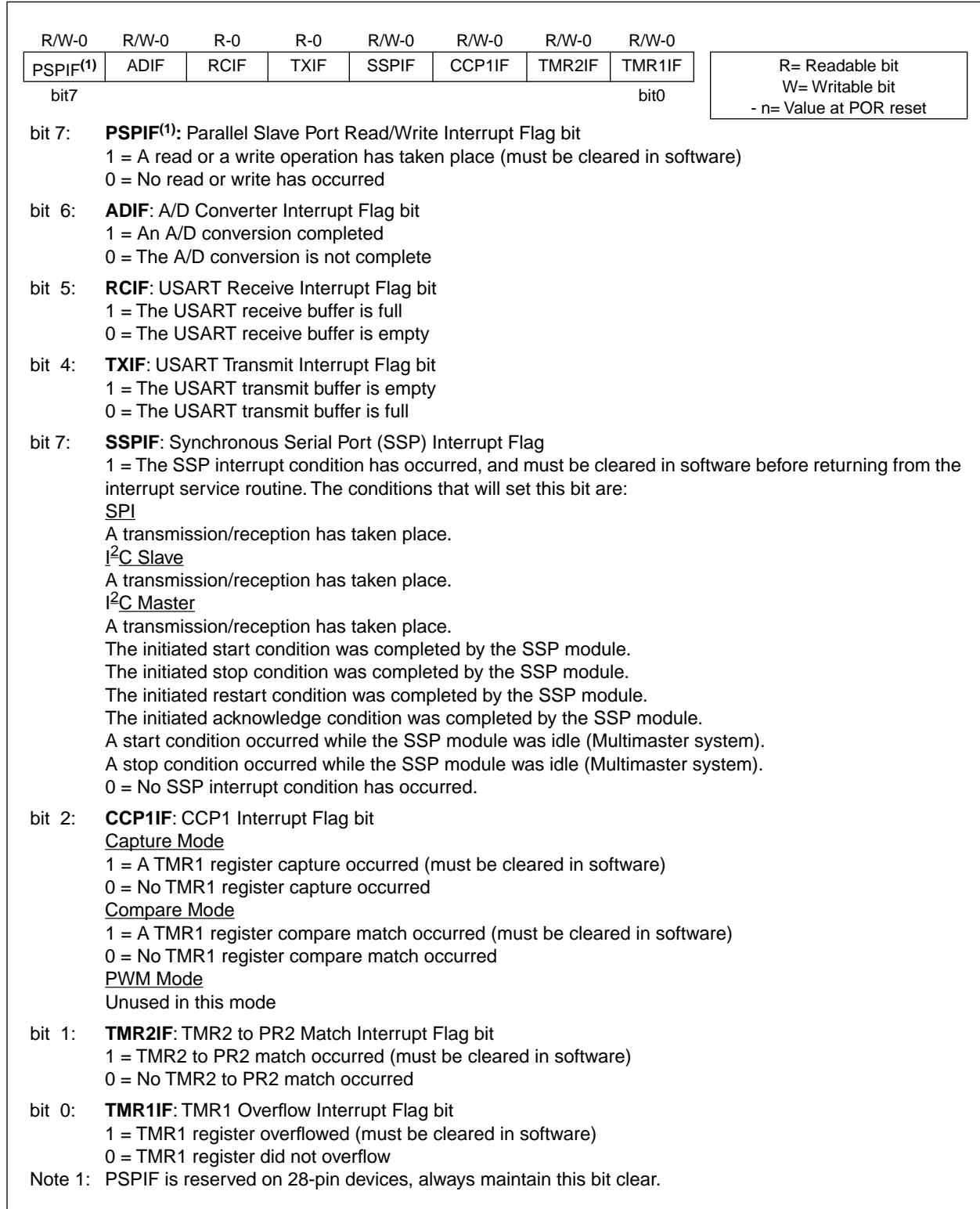
Note 1: PSPIE is reserved on 28-pin devices, always maintain this bit clear.

## 2.2.2.5 PIR1 REGISTER

This 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>).

**FIGURE 2-9: PIR1 REGISTER (ADDRESS 0Ch)**

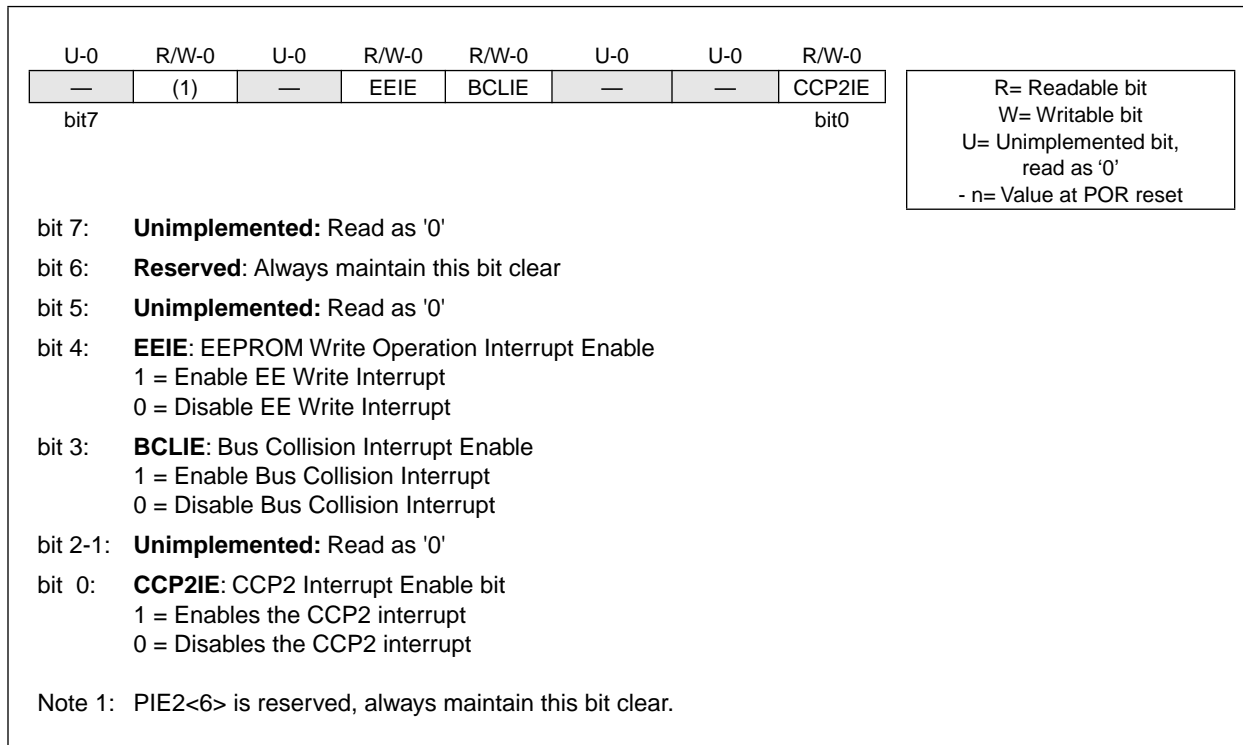


# PIC16F87X

## 2.2.2.6 PIE2 REGISTER

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

**FIGURE 2-10: PIE2 REGISTER (ADDRESS 8Dh)**



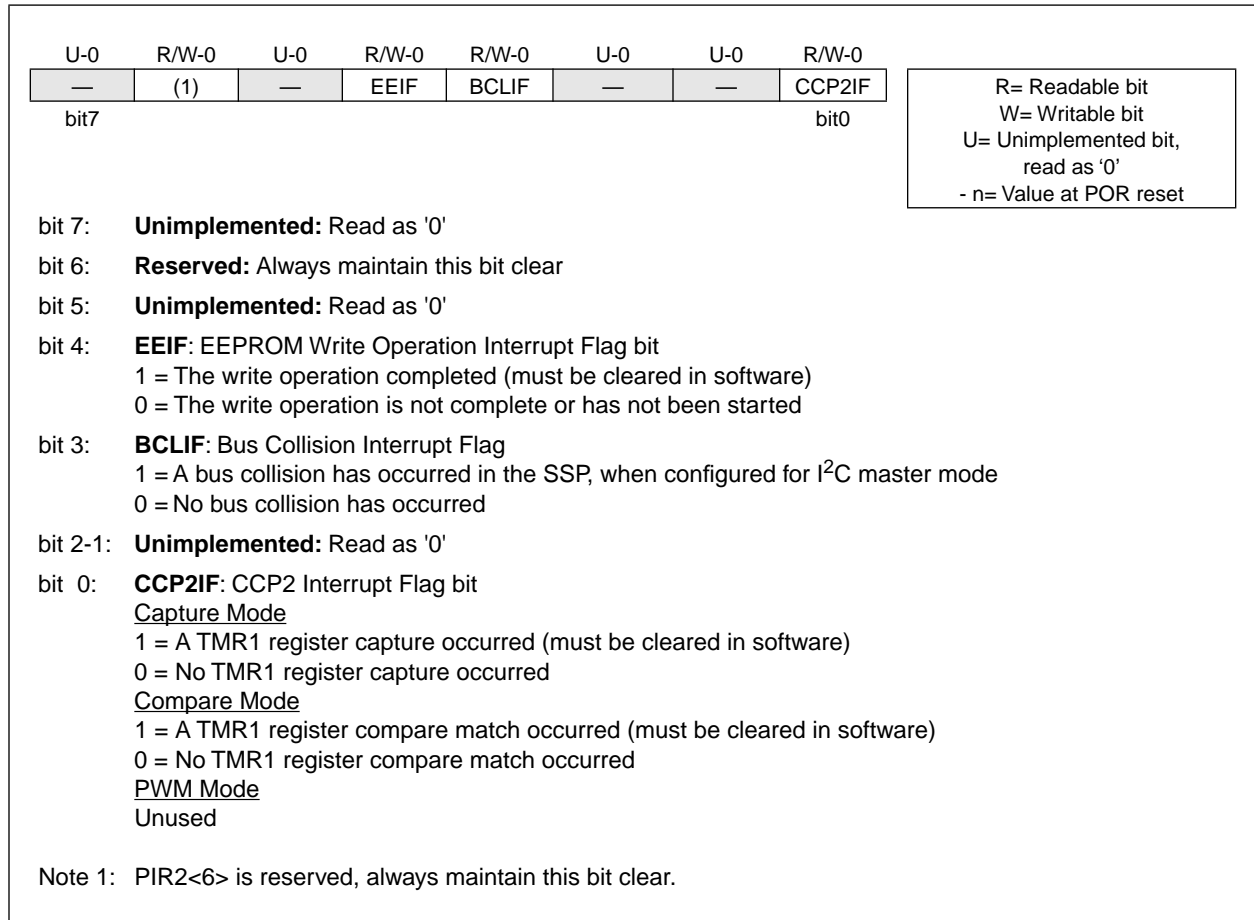


## 2.2.2.7 PIR2 REGISTER

This 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.

**FIGURE 2-11: PIR2 REGISTER (ADDRESS 0Dh)**



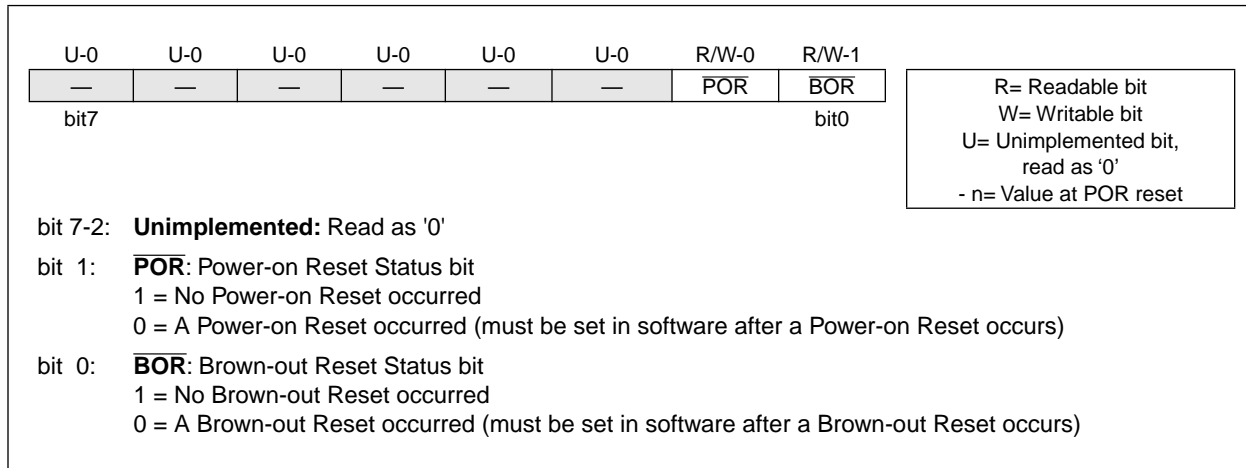
# PIC16F87X

## 2.2.2.8 PCON REGISTER

The Power Control (PCON) register contains a flag bit to allow differentiation between a Power-on Reset (POR) to an external  $\overline{MCLR}$  Reset or WDT Reset. Those devices with brown-out detection circuitry contain an additional bit to differentiate a Brown-out Reset condition from a Power-on Reset condition.

**Note:**  $\overline{BOR}$  is unknown on Power-on Reset if the BOR circuit is disabled by clearing the BODEN bit in Configuration Word. The  $\overline{BOR}$  status bit is a don't care and is not necessarily predictable if the brown-out circuit is disabled.

**FIGURE 2-12: PCON REGISTER (ADDRESS 8Eh)**



## 2.3 PCL and PCLATH

The program counter (PC) specifies the address of the instruction to fetch for execution. The PC is 13 bits wide. The low byte is called the PCL register. This register is readable and writable. The high byte is called the PCH register. This register contains the PC<12:8> bits and is not directly readable or writable. All updates to the PCH register go through the PCLATH register.

### 2.3.1 STACK

The stack allows a combination of up to 8 program calls and interrupts to occur. The stack contains the return address from this branch in program execution.

Midrange devices have 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 modified when the stack is PUSHed or POPed.

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).

## 2.4 Program Memory Paging

PIC16F87X 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 pushed onto the stack. Therefore, manipulation of the PCLATH<4:3> bits are not required for the return instructions (which POPs the address from the stack).

The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a *pointer*). This is indirect addressing.

### EXAMPLE 2-1: INDIRECT ADDRESSING

- Register file 05 contains the value 10h
- Register file 06 contains the value 0Ah
- Load the value 05 into the FSR register
- A read of the INDF register will return the value of 10h
- Increment the value of the FSR register by one (FSR = 06)
- A read of the INDF register now will return the value of 0Ah.

Reading INDF itself indirectly (FSR = 0) will produce 00h. Writing to the INDF register indirectly results in a no-operation (although STATUS bits may be affected).

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

# PIC16F87X

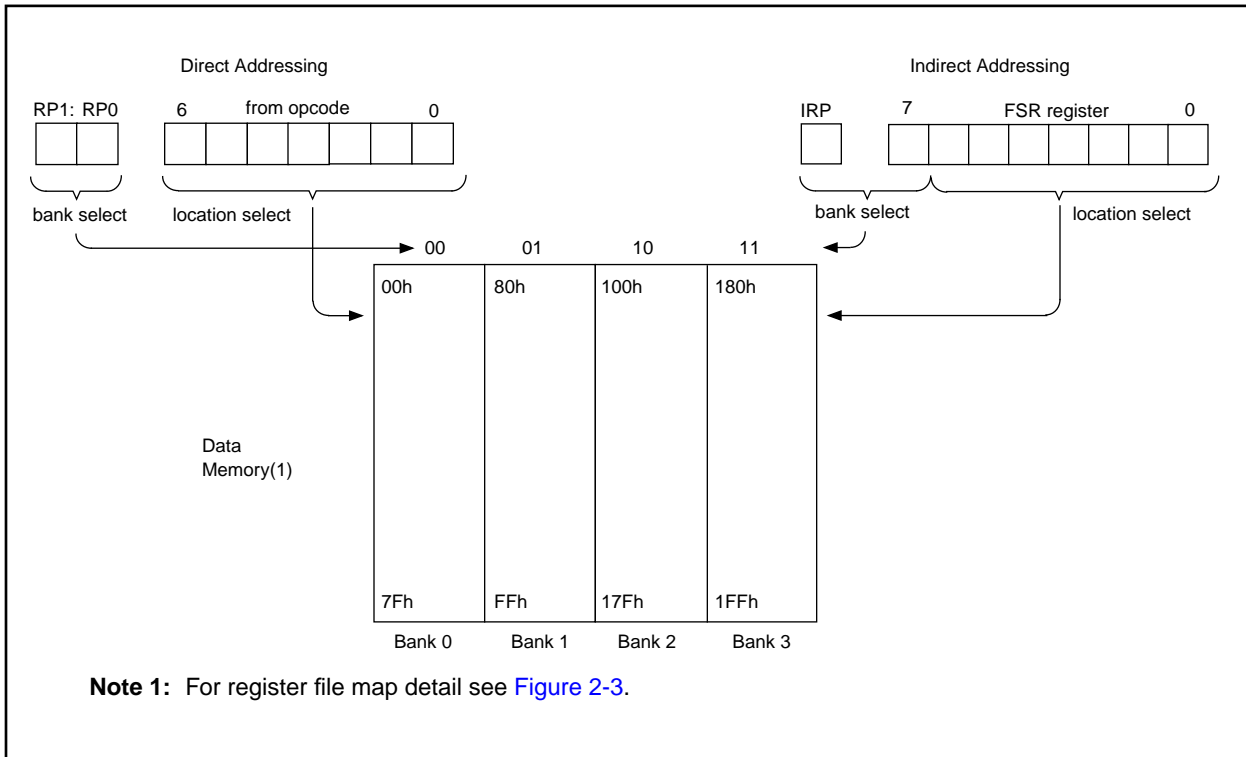
## EXAMPLE 2-2: HOW TO CLEAR RAM USING INDIRECT ADDRESSING

```

movlw 0x20 ;initialize pointer
movwf FSR ; to RAM
NEXT   clrf INDF ;clear INDF register
       incf FSR ;inc pointer
       btfss FSR,4 ;all done?
       goto NEXT ;NO, clear next
CONTINUE
       : ;YES, continue
    
```

An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in [Figure 2-13](#).

**FIGURE 2-13: DIRECT/INDIRECT ADDRESSING**



## 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, this 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 RA port 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).

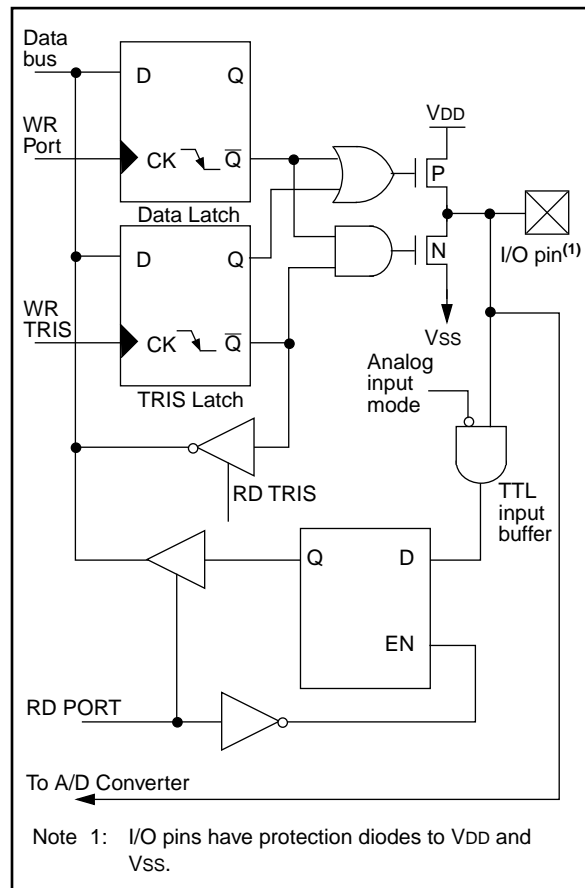
**Note:** On a Power-on Reset, these pins are configured as analog inputs and read as '0'.

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.

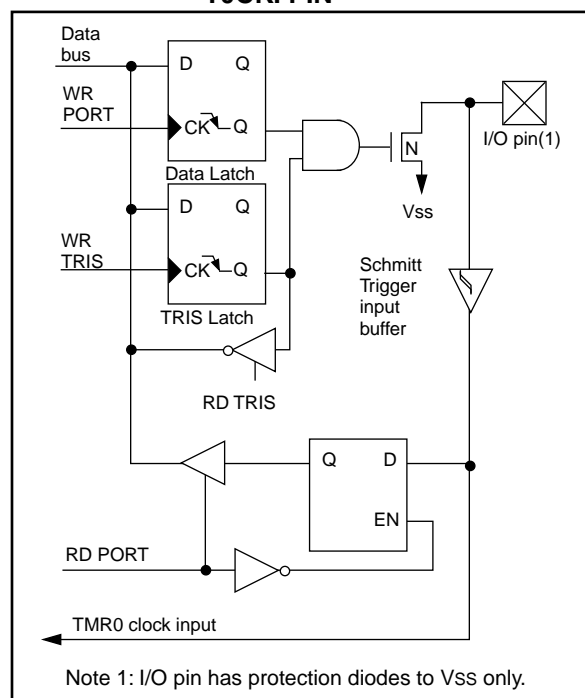
#### EXAMPLE 3-1: INITIALIZING PORTA

```
BCF STATUS, RP0 ;
CLRF PORTA      ; Initialize PORTA by
                ; clearing output
                ; data latches
BSF STATUS, RP0 ; Select Bank 1
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.



# 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.

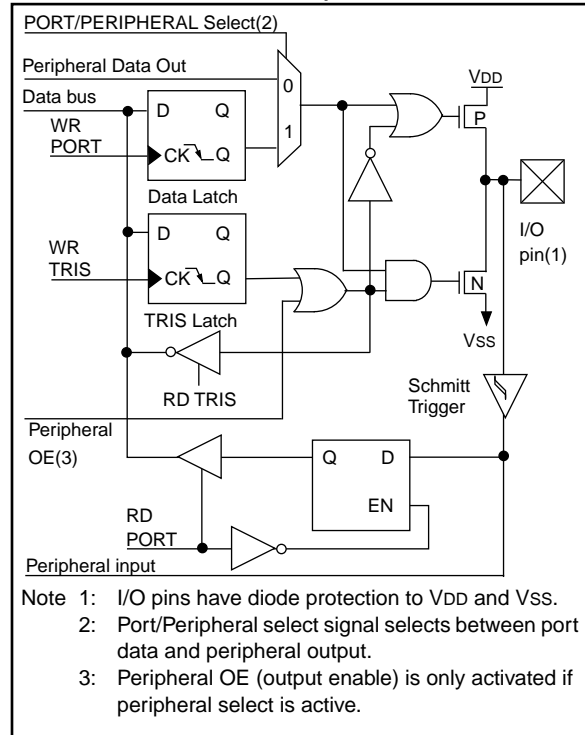
When the I<sup>2</sup>C module is enabled, the PORTC (3:4) pins can be configured with normal I<sup>2</sup>C levels or with SMBUS levels by using the CKE bit (SSPSTAT <6>).

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.

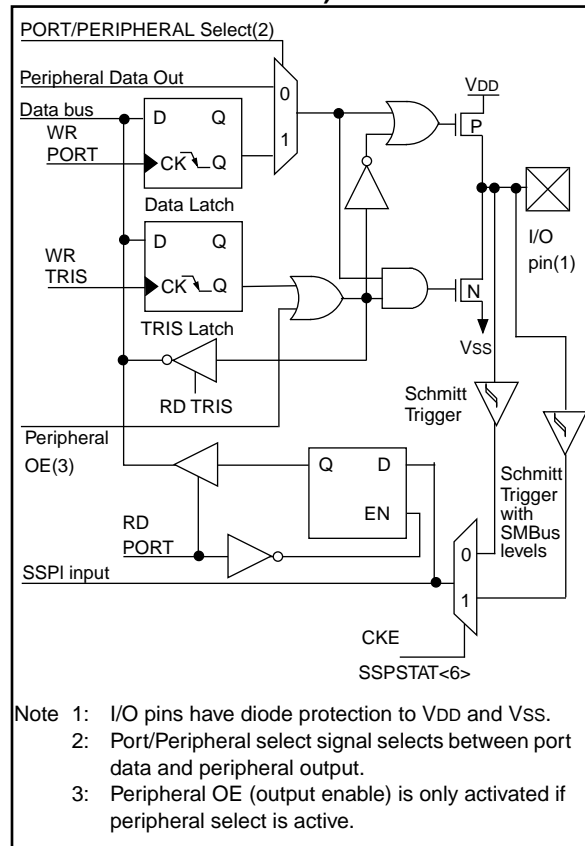
#### EXAMPLE 3-1: INITIALIZING PORTC

```
BCF STATUS, RP0 ; Select Bank 0
CLRF PORTC ; Initialize PORTC by
; clearing output
; data latches
BSF STATUS, RP0 ; Select Bank 1
MOVLW 0xCF ; Value used to
; initialize data
; direction
MOVWF TRISC ; Set RC<3:0> as inputs
; RC<5:4> as outputs
; RC<7:6> as inputs
```

**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.

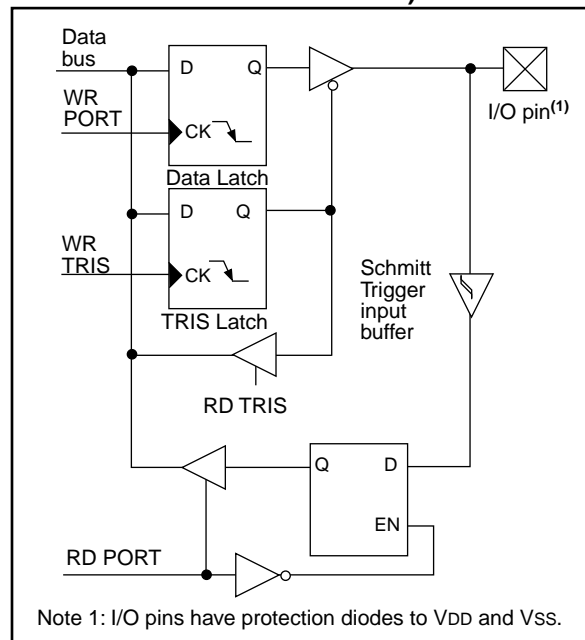
## 3.4 PORTD and TRISD Registers

This section is not applicable to the 28-pin devices.

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 28-pin devices.

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.

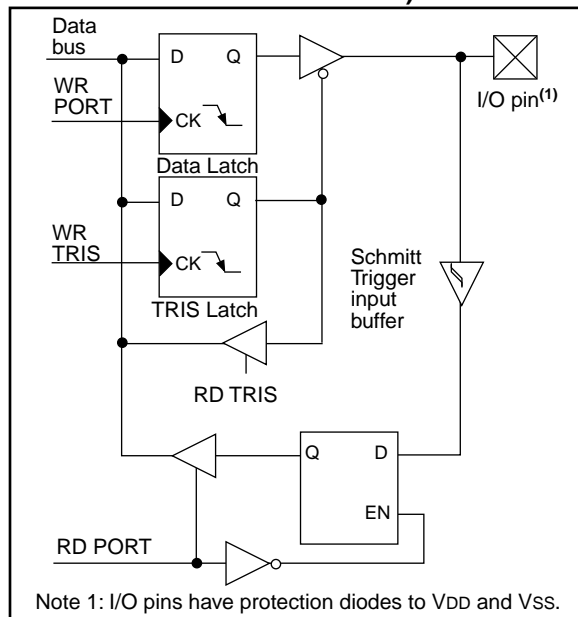
Figure 3-9 shows the TRISE register, which also controls the parallel slave port operation.

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)**



**FIGURE 3-9: 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	
IBF	OBF	IBOV	PSPMODE	—	bit2	bit1	bit0	R= Readable bit W= Writable bit U= Unimplemented bit, read as '0' - n= Value at POR reset
							bit7	
<p>bit 7 : <b>IBF</b>: 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</p> <p>bit 6: <b>OBF</b>: Output Buffer Full Status bit                      1 = The output buffer still holds a previously written word                      0 = The output buffer has been read</p> <p>bit 5: <b>IBOV</b>: 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</p> <p>bit 4: <b>PSPMODE</b>: Parallel Slave Port Mode Select bit                      1 = Parallel slave port mode                      0 = General purpose I/O mode</p> <p>bit 3: <b>Unimplemented</b>: Read as '0'</p> <p><b>PORTE Data Direction Bits</b></p> <p>bit 2: <b>Bit2</b>: Direction Control bit for pin RE2/<math>\overline{CS}</math>/AN7                      1 = Input                      0 = Output</p> <p>bit 1: <b>Bit1</b>: Direction Control bit for pin RE1/<math>\overline{WR}</math>/AN6                      1 = Input                      0 = Output</p> <p>bit 0: <b>Bit0</b>: Direction Control bit for pin RE0/<math>\overline{RD}</math>/AN5                      1 = Input                      0 = Output</p>								

**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.

## 3.6 Parallel Slave Port

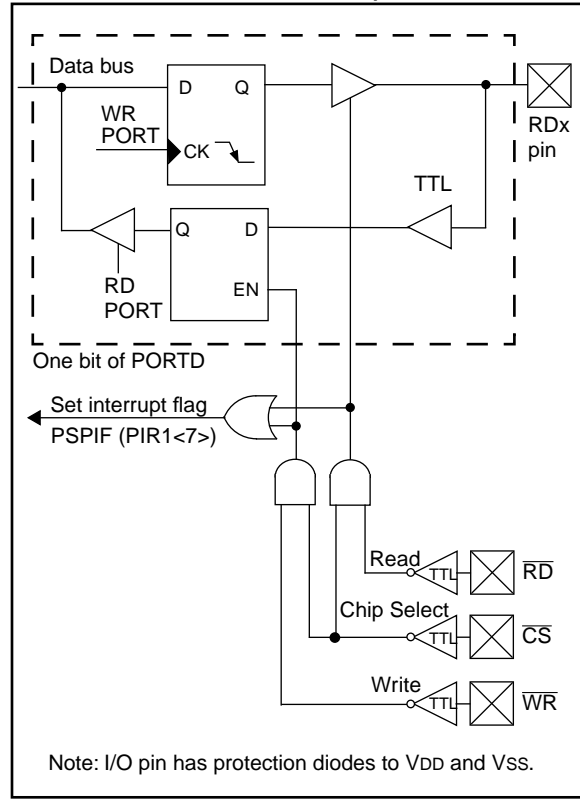
The Parallel Slave Port is not implemented on the 28-pin devices.

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/ $\overline{WR}$ .

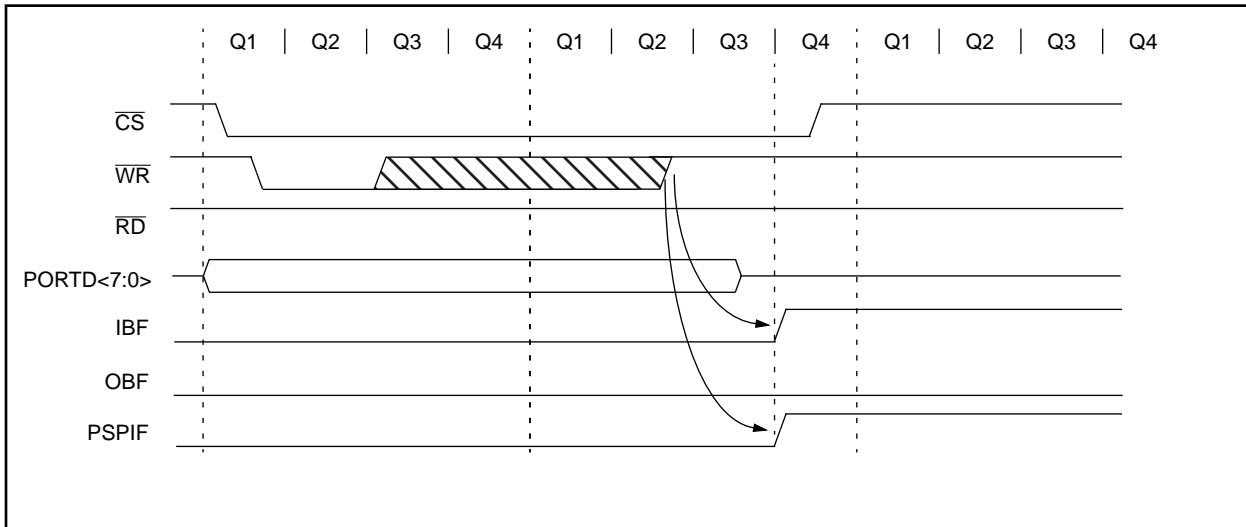
It can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an 8-bit latch. Setting bit PSPMODE enables port pin RE0/ $\overline{RD}$  to be the  $\overline{RD}$  input, RE1/ $\overline{WR}$  to be the  $\overline{WR}$  input and RE2/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

A write to the PSP occurs when both the  $\overline{CS}$  and  $\overline{WR}$  lines are first detected low. A read from the PSP occurs when both the  $\overline{CS}$  and  $\overline{RD}$  lines are first detected low.

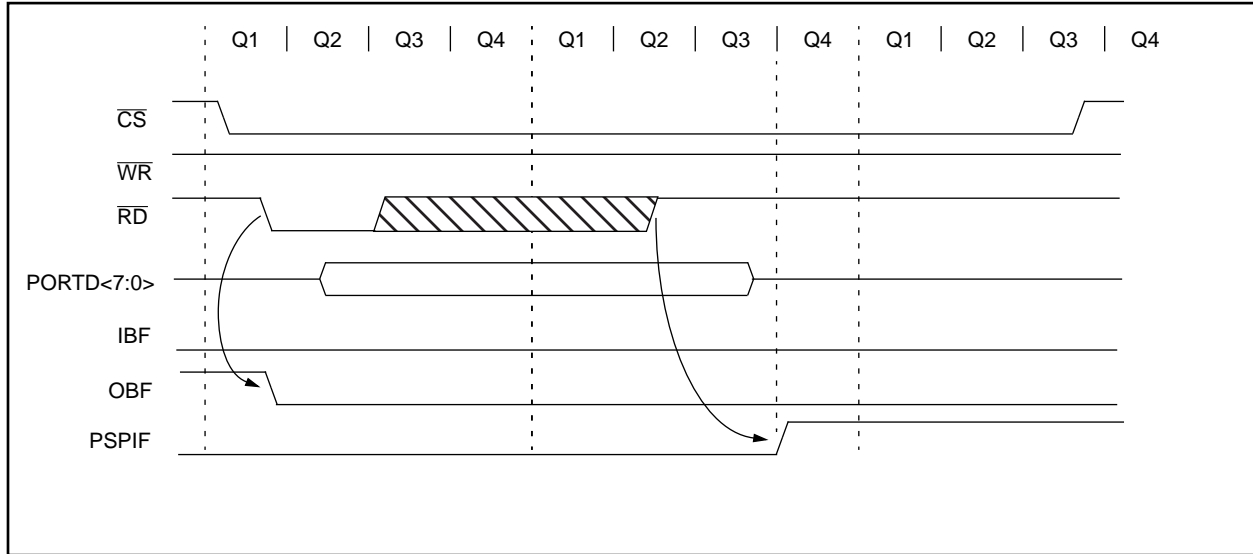
**FIGURE 3-10: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)**



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



**FIGURE 3-12: 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. The data memory is not directly mapped in the register file space. Instead it is indirectly addressed through the Special Function Registers.

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 reads allow checksum calculation for UL type applications. A byte or word write automatically erases the location and writes the new data (erase before write).

When interfacing to the program memory block, the EEDATH:EEDATA registers form a 2 byte word which holds the 14-bit data for read/write, and the EEADRH:EEADR registers form a 2 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 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 PIC16F874/873 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.

# PIC16F87X

## 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.

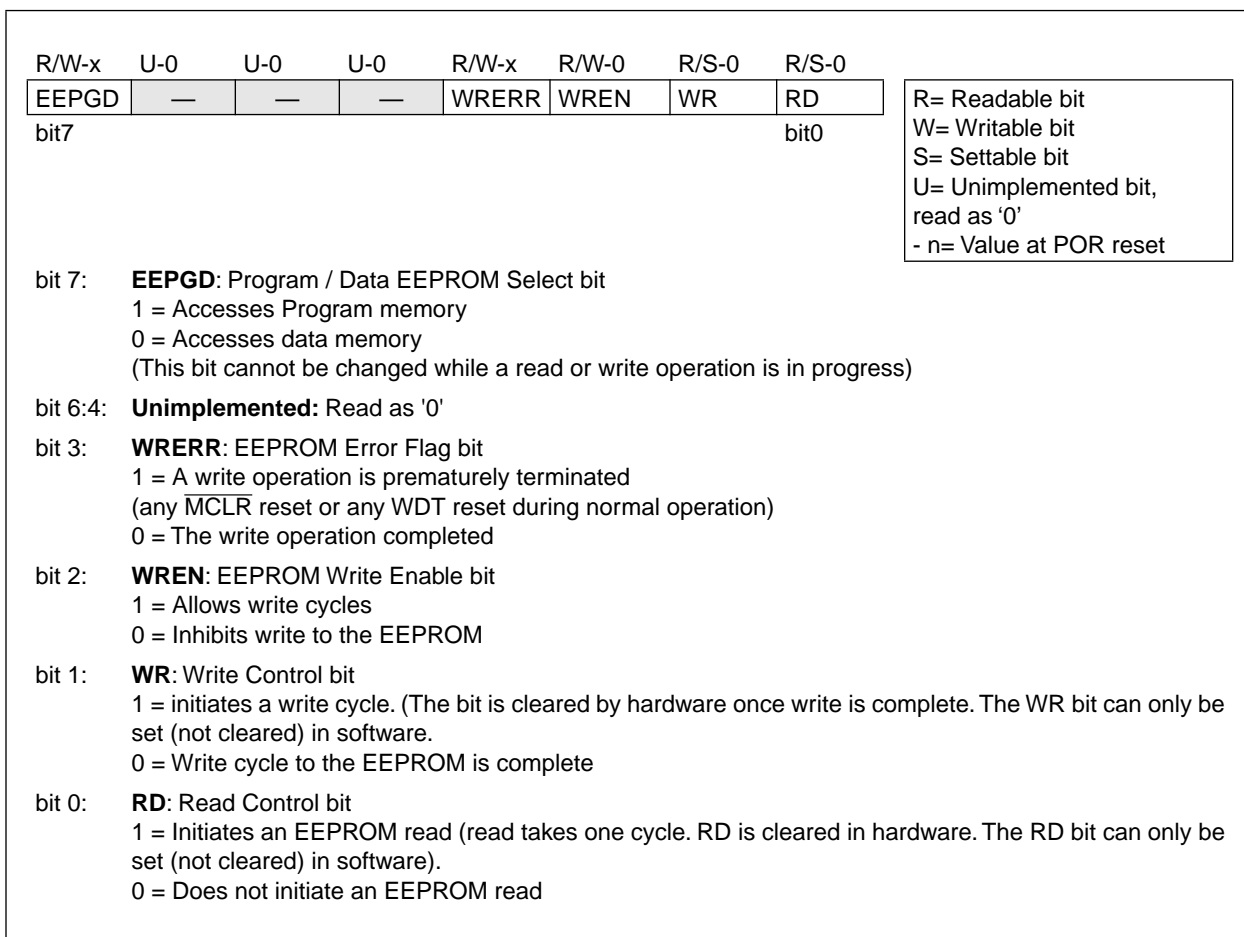
Control bits RD and WR initiate read and write operations, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at the

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  $\overline{MCLR}$  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.

**FIGURE 4-1: EECON1 REGISTER (ADDRESS 18Ch)**



## 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, in 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 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., lost 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, set the EEPGD control bit (EECON1<7>) and then set control bit RD (EECON1<0>). Once the read control bit is set, the microcontroller will use the second instruction cycle to read the data. This causes the second instruction immediately following the "BSF EECON1,RD" instruction to be ignored. The data is

available, in the third cycle, in the EEDATA and EEDATH registers; therefore it can be read as two bytes in the following instructions. The data can be read out of EEDATH:EEDATA starting with the third instruction cycle after the BSF EECON1,RD instruction. 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  ADDRRL        ;
MOVWF  EEADR         ; LSByte of Program Address to read
BSF    STATUS, RP0    ; Bank 3
BSF    EECON1, EEPGD  ; Point to PROGRAM memory
BSF    EECON1, RD     ; EEPROM Read

NOP                    ; Any instructions here are ignored as program
NOP                    ; memory is read in third cycle after BSF EECON1,RD

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 control 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.

After the “BSF EECON1,WR” instruction, the microcontroller will execute the next instruction and then ignore the subsequent instruction. A NOP instruction should be used in both places. The microcontroller will then halt internal operations for the TPEW (parameter D133) in which the write takes place. This is not a SLEEP mode, as the clocks and peripherals will continue to run. 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  ADDRDL         ;
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
;
MOVLW  55h            ;
MOVWF  EECON2         ; Write 55h
MOVLW  AAh            ;
MOVWF  EECON2         ; Write AAh
BSF    EECON1, WR     ; Set WR bit to begin write
;
;
NOP    ; Instructions here are ignored by the microcontroller
;
NOP    ; 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).

## 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. 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.

### 4.9.3 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.

### 4.9.4 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 7-1 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 7-1: 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

## 5.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

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

Figure 5-1 is a simplified block diagram of the Timer0 module.

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

### 5.1 Timer0 Operation

Timer0 can operate as a timer or as a counter.

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 below.

When an external clock input is used for Timer0, it must meet certain requirements. The requirements ensure the external clock can be synchronized with the internal phase clock (TOSC). Also, there is a delay in the actual incrementing of Timer0 after synchronization.

Additional information on external clock requirements is available in the PICmicro™ Mid-Range Reference Manual, DS33023.

### 5.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer (Figure 5-2). For simplicity, this counter is being referred to as “prescaler” throughout this data sheet. Note that there is only one prescaler available which is mutually exclusively shared between the Timer0 module and the Watchdog Timer. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer, and vice-versa.

The prescaler is not readable or writable.

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

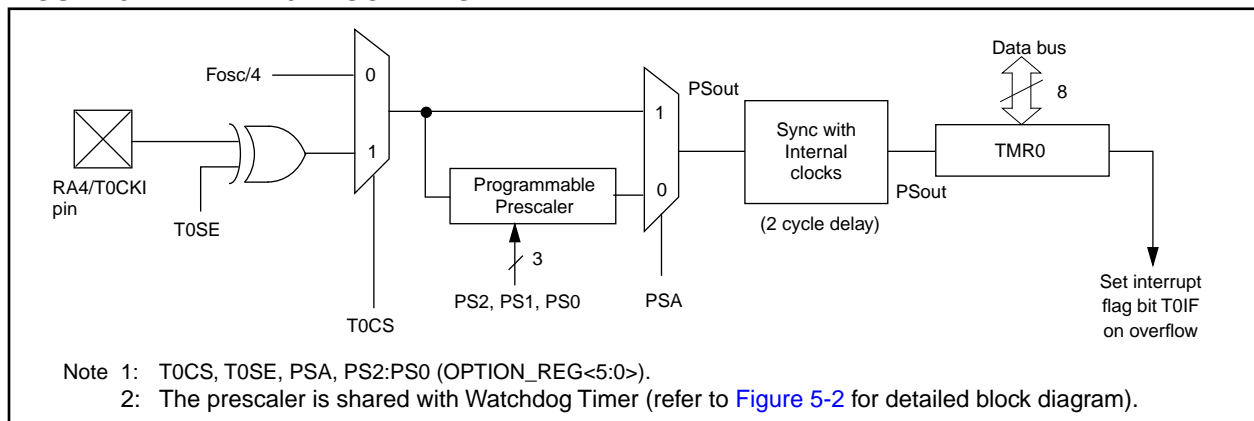
Clearing bit PSA will assign the prescaler to the Timer0 module. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4, ..., 1:256 are selectable.

Setting bit PSA will assign the prescaler to the Watchdog Timer (WDT). When the prescaler is assigned to the WDT, prescale values of 1:1, 1:2, ..., 1:128 are selectable.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g. CLRF TMR0, MOVWF TMR0, BSF TMR0,x...etc.) will clear the prescaler. When assigned to WDT, a CLRWDT instruction will clear the prescaler along with the WDT.

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

**FIGURE 5-1: TIMER0 BLOCK DIAGRAM**



# PIC16F87X

## 5.2.1 SWITCHING PRESCALER ASSIGNMENT

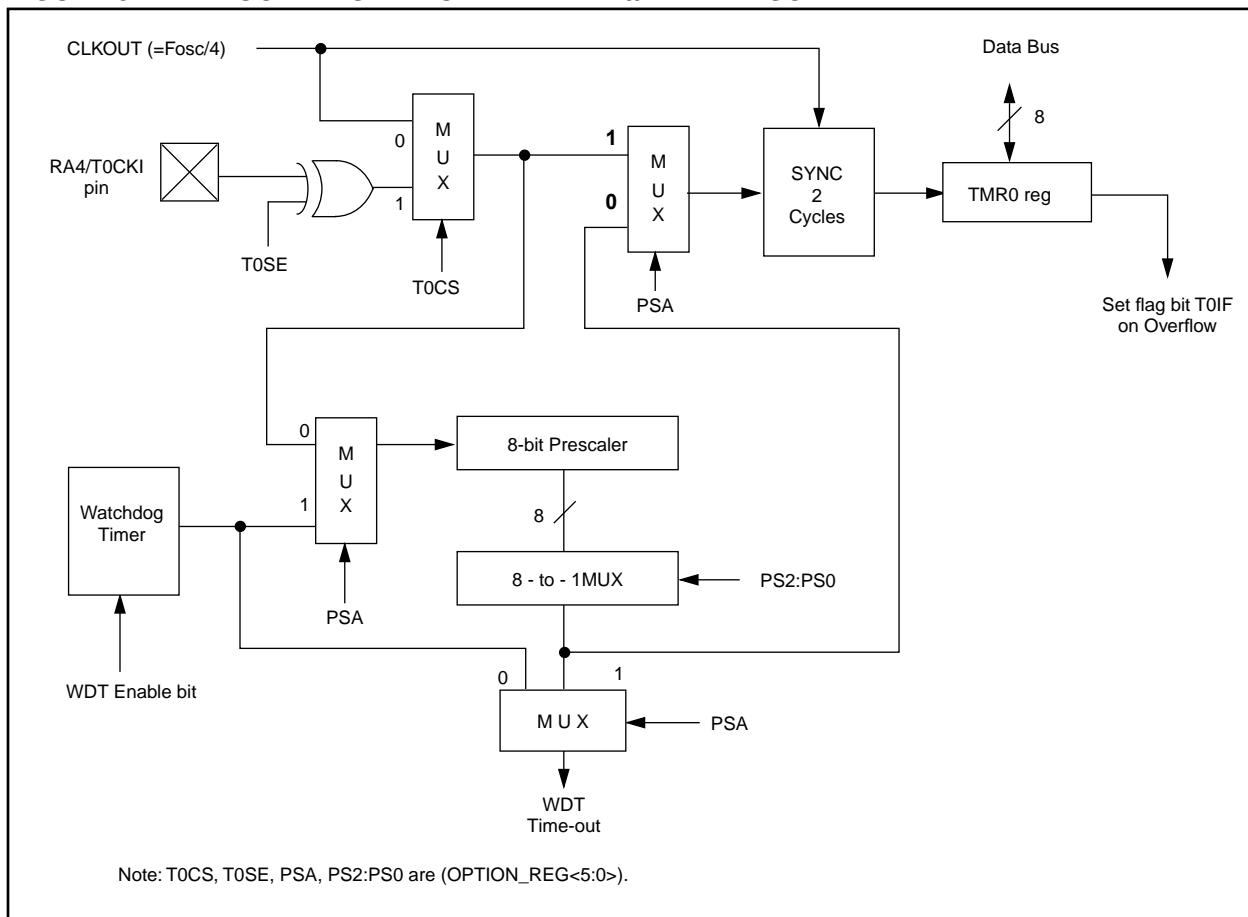
The prescaler assignment is fully under software control, i.e., it can be changed “on the fly” during program execution.

**Note:** To avoid an unintended device RESET, a specific instruction sequence (shown in the PICmicro™ Mid-Range 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.

## 5.3 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 5-2: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER**



**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 xxxxx	uuuu uuuu
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
81h,181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	PORTA Data Direction Register								--11 1111	--11 1111

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



## 6.0 TIMER1 MODULE

The Timer1 module timer/counter has the following features:

- 16-bit timer/counter (Two 8-bit registers; TMR1H and TMR1L)
- Readable and writable (Both registers)
- Internal or external clock select
- Interrupt on overflow from FFFFh to 0000h
- Reset from CCP module trigger

Timer1 has a control register, shown in [Figure 6-1](#). Timer1 can be enabled/disabled by setting/clearing control bit TMR1ON (T1CON<0>).

[Figure 6-3](#) is a simplified block diagram of the Timer1 module.

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

## 6.1 Timer1 Operation

Timer1 can operate in one of these modes:

- As a timer
- As a synchronous counter
- As an asynchronous 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.

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

Timer1 also has an internal “reset input”. This reset can be generated by the CCP module ([Section 8.0](#)).

**FIGURE 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  
Note: The oscillator inverter and feedback resistor are 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

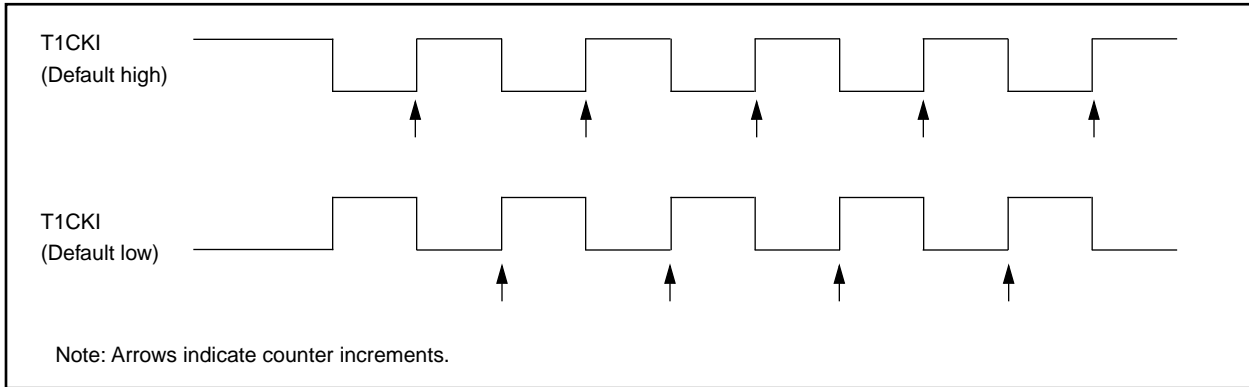
# PIC16F87X

## 6.1.1 TIMER1 COUNTER OPERATION

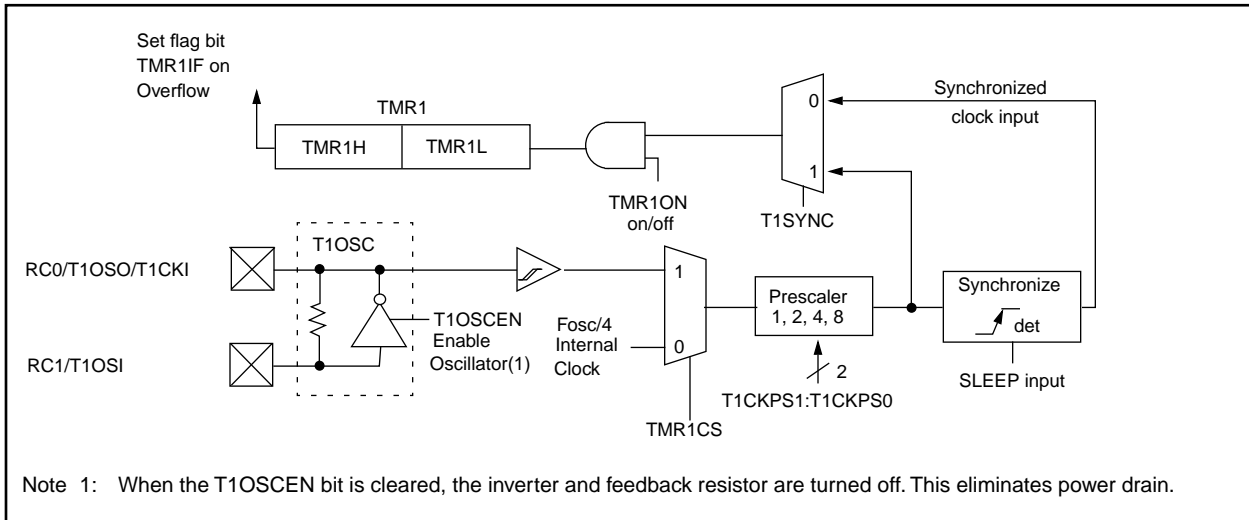
In this mode, Timer1 is being incremented via an external source. Increments occur on a rising edge.

Timer1 is enabled in counter mode, the module must first have a falling edge before the counter begins to increment.

**FIGURE 6-2: TIMER1 INCREMENTING EDGE**



**FIGURE 6-3: TIMER1 BLOCK DIAGRAM**



## 6.2 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 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
<b>These values are for design guidance only.</b>			
<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	
Note 1: Higher capacitance increases the stability of oscillator but also increases the start-up time. Note 2: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.			

## 6.3 Timer1 Interrupt

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>).

## 6.4 Resetting Timer1 using a CCP Trigger Output

If the CCP module is configured in compare mode to generate a "special event trigger" (CCP1M3:CCP1M0 = 1011), this signal will reset Timer1, and for CCP2 only, start an A/D conversion (if the A/D module is enabled).

**Note:** The special event triggers from the CCP1 module 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, the write will take precedence.

In this mode of operation, the CCPR1H:CCPR1L registers pair effectively becomes the period register for Timer1.

**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	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	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	T1SYNC	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: These bits are reserved on the 28-pin devices, always maintain these bits clear.

# PIC16F87X

---

NOTES:

## 7.0 TIMER2 MODULE

The Timer2 module timer has the following features:

- 8-bit timer (TMR2 register)
- 8-bit period register (PR2)
- Readable and writable (Both registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR2 match of PR2
- SSP module optional use of TMR2 output to generate clock shift

Timer2 has a control register, shown in [Figure 7-1](#). Timer2 can be shut off by clearing control bit TMR2ON (T2CON<2>) to minimize power consumption.

[Figure 7-2](#) is a simplified block diagram of the Timer2 module.

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

## 7.1 Timer2 Operation

Timer2 can be used as the PWM time-base for PWM mode of the CCP module.

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 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>)).

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

TMR2 is not cleared when T2CON is written.

**FIGURE 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

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

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

bit 6-3: **TOUTPS3:TOUTPS0:** Timer2 Output Postscale Select bits  
 0000 = 1:1 Postscale  
 0001 = 1:2 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

# PIC16F87X

## 7.2 Timer2 Interrupt

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.

## 7.3 Output of TMR2

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

FIGURE 7-2: TIMER2 BLOCK DIAGRAM

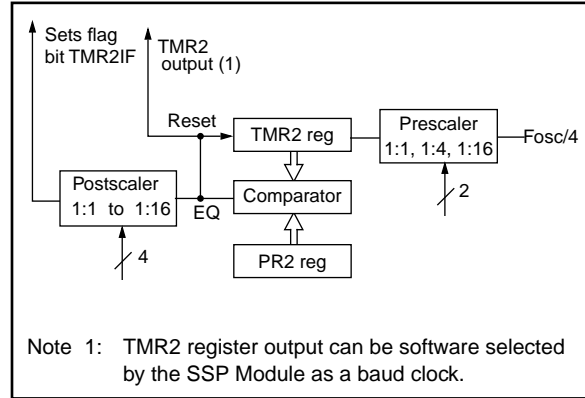


TABLE 7-1 REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

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	RBFIF	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: These bits are reserved on the 28-pin devices, always maintain these bits clear.

## 8.0 CAPTURE/COMPARE/PWM (CCP) MODULE(S)

Each CCP (Capture/Compare/PWM) module contains a 16-bit register which can operate as a 16-bit capture register, as a 16-bit compare register or as a PWM master/slave Duty Cycle register. Table 8-1 shows the timer resources of the CCP module modes.

The operation of CCP1 is identical to that of CCP2, with the exception of the special trigger. Therefore, operation of a CCP module in the following sections is described with respect to CCP1.

Table 8-2 shows the interaction of the CCP modules.

### 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. All are readable and writable.

**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

### CCP2 Module

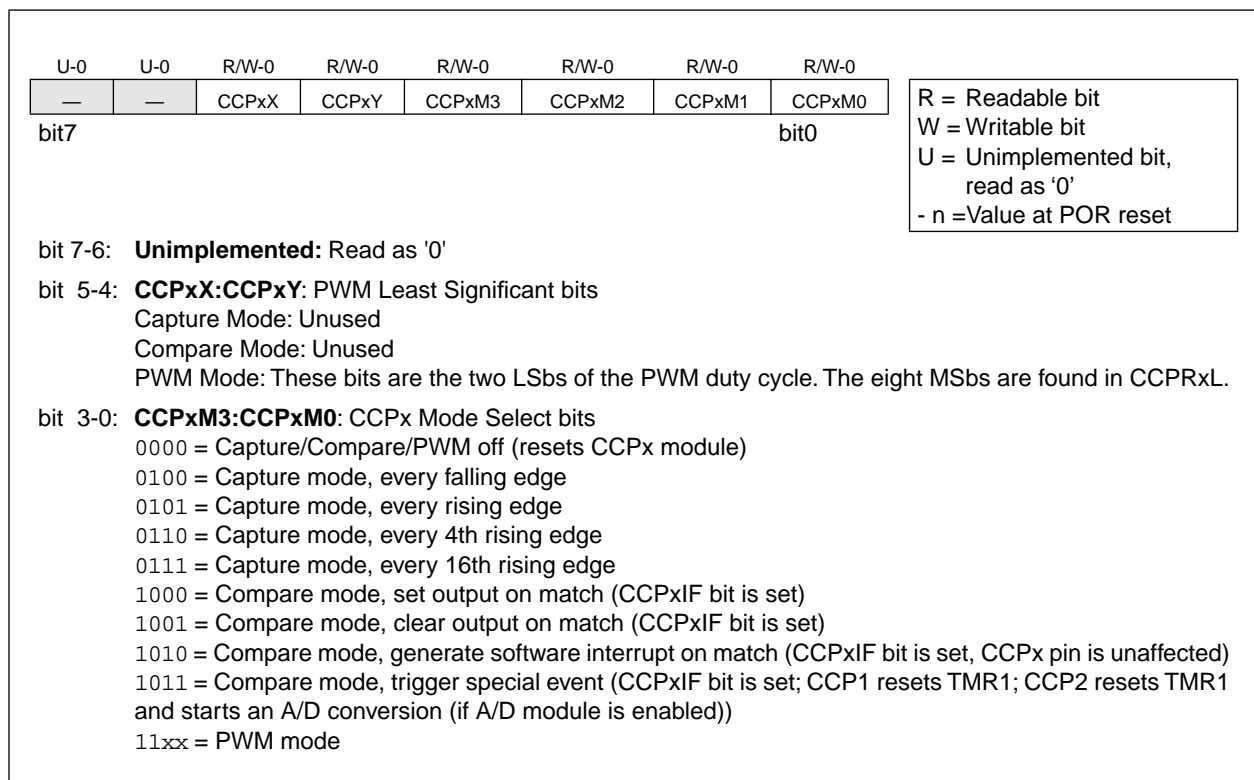
Capture/Compare/PWM Register2 (CCPR2) is comprised of two 8-bit registers: CCPR2L (low byte) and CCPR2H (high byte). The CCP2CON register controls the operation of CCP2. All are readable and writable.

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

**TABLE 8-1 CCP MODE - TIMER RESOURCE**

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

**FIGURE 8-1: CCP1CON REGISTER (ADDRESS 17h) / CCP2CON REGISTER (ADDRESS 1Dh)**



# PIC16F87X

## 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

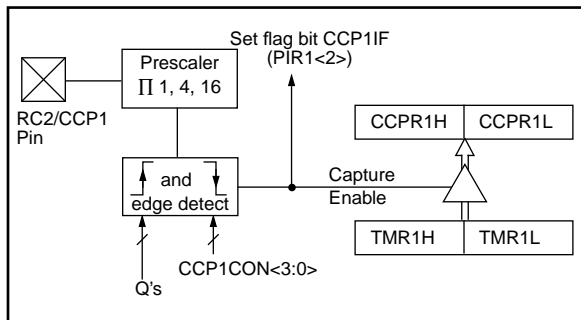
An event is selected by control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). When a capture is made, the interrupt request flag bit CCP1IF (PIR1<2>) is set. It must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value will be lost.

### 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 is configured as an output, a write to the port can cause a capture condition.

**FIGURE 8-2: 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. This means that 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
                    ; mode value and CCP ON
MOVWF  CCP1CON      ;Load CCP1CON with this
                    ; value
```



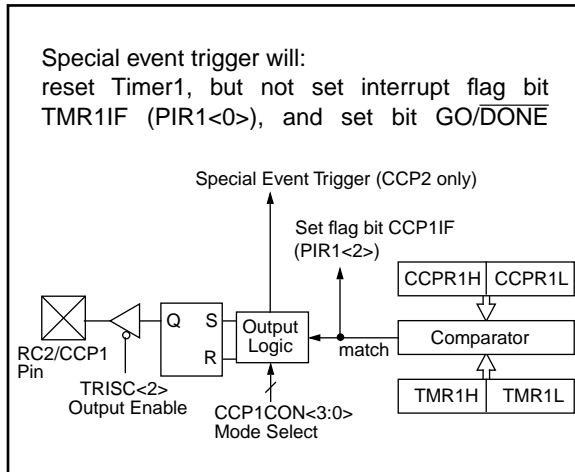
## 8.2 Compare Mode

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 8-3: 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 is chosen the CCP1 pin is not affected. Only a CCP interrupt is generated (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 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 CCP2 module will not set interrupt flag bit TMR1IF (PIR1<0>).

**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	TOIE	INTE	RBIE	TOIF	INTF	RBF	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
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	T1SYNC	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

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

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

## 8.3 PWM Mode

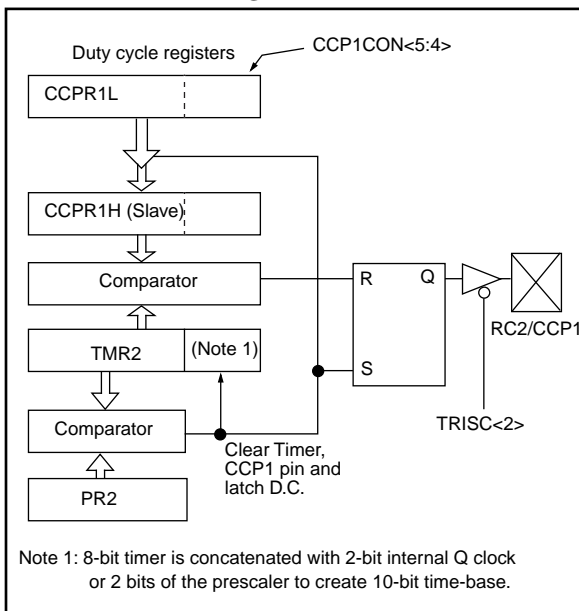
In Pulse Width Modulation (PWM) mode, the CCP1 pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output.

**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 8-4 shows a simplified block diagram of the CCP module in PWM mode.

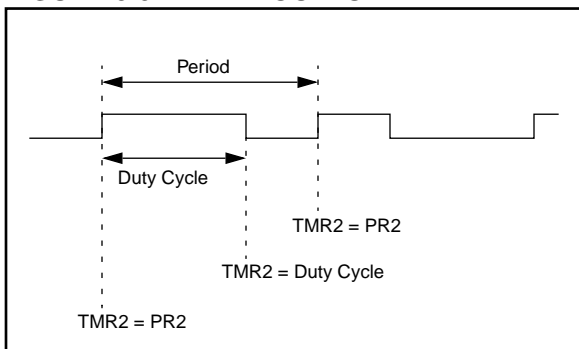
For a step by step procedure on how to set up the CCP module for PWM operation, see Section 8.3.3.

**FIGURE 8-4: SIMPLIFIED PWM BLOCK DIAGRAM**



A PWM output (Figure 8-5) 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 8-5: 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 7.0) 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:

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

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

For an example PWM period and duty cycle calculation, see the Midrange Reference Manual (DS33023).

## 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.

**TABLE 8-4 EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHz**

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	5.5

**TABLE 8-5 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
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
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

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 28-pin devices, always maintain these bits clear.

# PIC16F87X

---

NOTES:

## 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 9-2, and Figure 9-3 shows the block diagrams for the two different I<sup>2</sup>C modes of operation.

**FIGURE 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/A	P	S	R/W	UA	BF
bit7							bit0

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

bit 7: **SMP:** Sample bit  
SPI Master Mode  
1 = Input data sampled at end of data output time  
0 = Input data sampled at middle of data output time  
SPI Slave Mode  
SMP must be cleared when SPI is used in slave mode  
In I<sup>2</sup>C master or slave mode:  
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)

bit 6: **CKE:** SPI Clock Edge Select (Figure 9-6, Figure 9-8, and Figure 9-9)  
SPI Mode:  
**CKP = 0**  
1 = Data transmitted on rising edge of SCK  
0 = Data transmitted on falling edge of SCK  
**CKP = 1**  
1 = Data transmitted on falling edge of SCK  
0 = Data transmitted on rising edge of SCK  
In I<sup>2</sup>C Master or Slave Mode:  
1 = Input levels conform to SMBUS spec  
0 = Input levels conform to I<sup>2</sup>C specs

bit 5: **D/A:** 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

bit 4: **P:** 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

bit 3: **S:** 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

bit 2: **R/W:** 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 ACK bit.  
In I<sup>2</sup>C slave mode:  
1 = Read  
0 = Write  
In I<sup>2</sup>C master mode:  
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.

bit 1: **UA:** 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

bit 0: **BF:** Buffer Full Status bit  
Receive (SPI and I<sup>2</sup>C modes)  
1 = Receive complete, SSPBUF is full  
0 = Receive not complete, SSPBUF is empty  
Transmit (I<sup>2</sup>C mode only)  
1 = Data Transmit in progress (does not include the  $\overline{\text{ACK}}$  and stop bits), SSPBUF is full  
0 = Data Transmit complete (does not include the  $\overline{\text{ACK}}$  and stop bits), SSPBUF is empty

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

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

R = Readable bit  
W = Writable bit  
- 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  $\overline{SS}$  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

**FIGURE 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 ( $\overline{SS}$ )

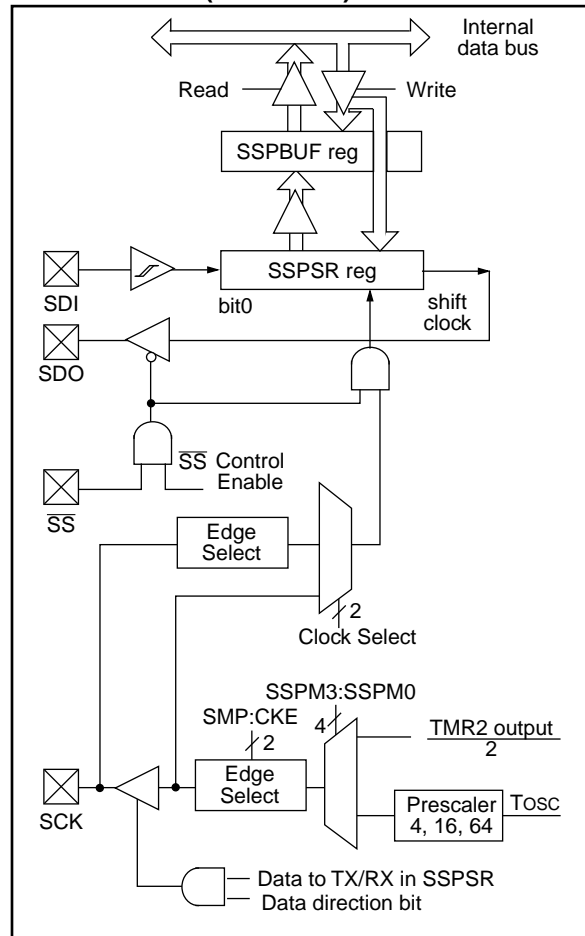
### 9.1.1 OPERATION

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 9-4: MSSP BLOCK DIAGRAM (SPI MODE)**



The MSSP consists of a transmit/receive Shift Register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR, until the received data is ready. Once the 8-bits of data have been received, that byte is moved to the SSPBUF register. Then the buffer full detect bit BF (SSPSTAT<0>) and the interrupt flag bit SSPIF (PIR1<3>) are set. This double buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored, and the write collision detect bit WCOL (SSPCON<7>) will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. Buffer full bit, BF (SSPSTAT<0>), indicates when the SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, bit BF is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally the MSSP Interrupt is used to

determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. [Example 9-1](#) shows the loading of the SSPBUF (SSPSR) for data transmission.

## EXAMPLE 9-1: LOADING THE SSPBUF (SSPSR) REGISTER

```

BSF    STATUS, RP0    ;Specify Bank 1
LOOP  BTFSS SSPSTAT, BF ;Has data been
                                ;received
                                ;(transmit
                                ;complete)?

GOTO   LOOP          ;No
BCF    STATUS, RP0    ;Specify Bank 0
MOVF   SSPBUF, W      ;W reg = contents
                                ;of SSPBUF
MOVWF  RXDATA         ;Save in user RAM
MOVF   TXDATA, W      ;W reg = contents
                                ; of TXDATA
MOVWF  SSPBUF         ;New data to xmit
    
```

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

### 9.1.2 ENABLING SPI I/O

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 SDI,

SDO, SCK, and  $\overline{SS}$  pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed. That is:

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

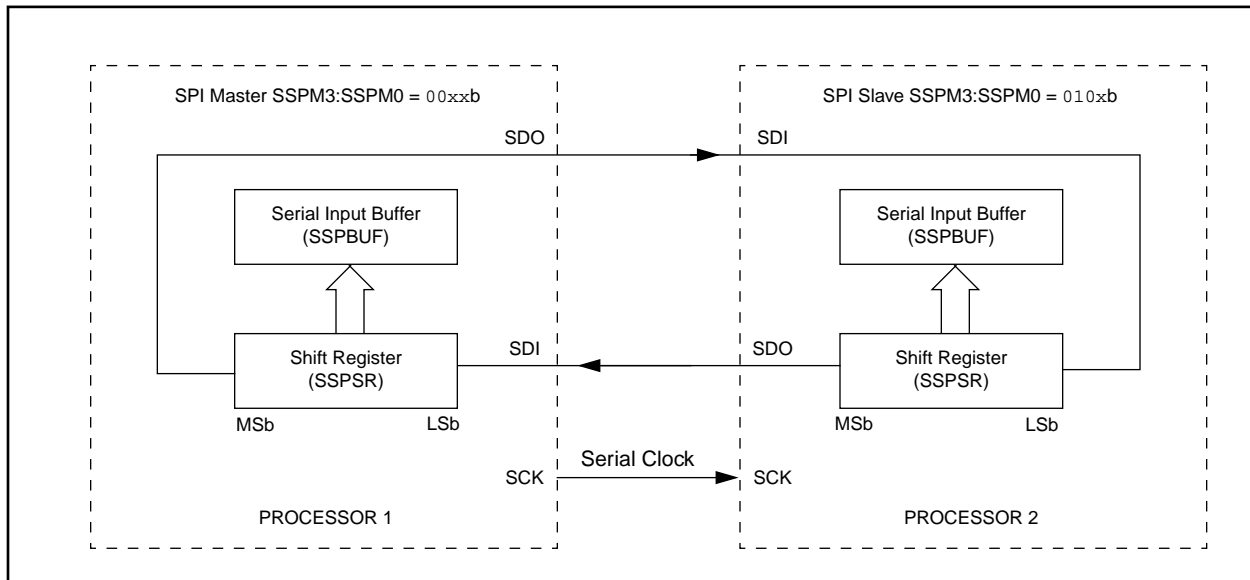
Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

### 9.1.3 TYPICAL CONNECTION

[Figure 9-5](#) shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge, and latched on the opposite edge of the clock. Both processors should be programmed to same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data — Slave sends dummy data
- Master sends data — Slave sends data
- Master sends dummy data — Slave sends data

**FIGURE 9-5: SPI MASTER/SLAVE CONNECTION**



## 9.1.4 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

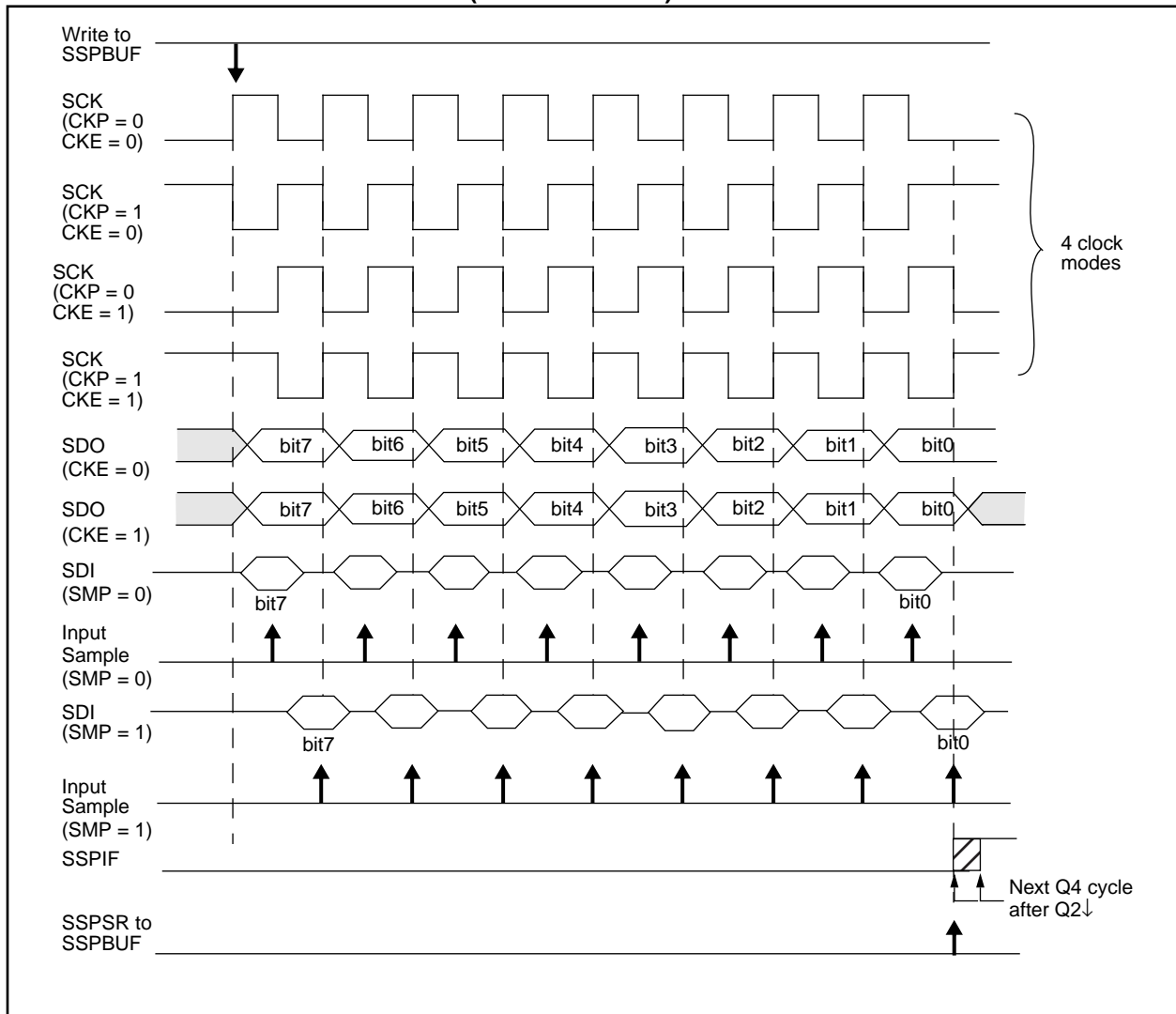
[Figure 9-6](#), [Figure 9-8](#), and [Figure 9-9](#) where the MSB is transmitted first. In master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

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

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 9-6: SPI MODE WAVEFORM (MASTER MODE)**



# PIC16F87X

## 9.1.5 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.

## 9.1.6 SLAVE SELECT SYNCHRONIZATION

The  $\overline{SS}$  pin allows a synchronous slave mode. The SPI must be in slave mode with  $\overline{SS}$  pin control enabled (SSPCON<3:0> = 0100). The pin must not be driven low for the  $\overline{SS}$  pin to function as an input. TRISA<5> must be set. When the  $\overline{SS}$  pin is low, transmission and reception are enabled and the SDO pin is driven. When the  $\overline{SS}$  pin goes high, the

SDO pin is no longer driven, even if in the middle of a transmitted byte, and becomes a floating output. External pull-up/ pull-down resistors may be desirable, depending on the application.

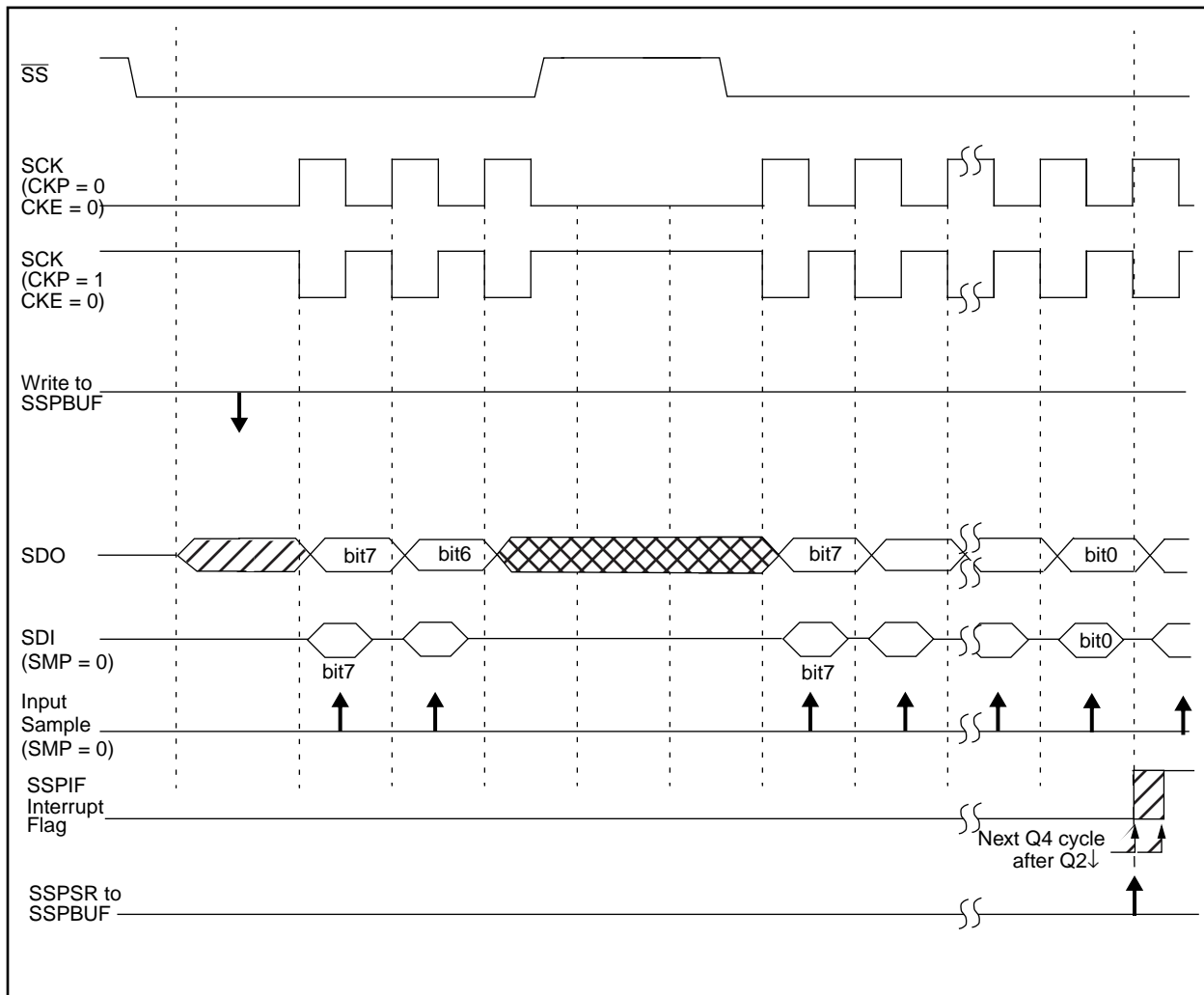
**Note:** When the SPI module is in Slave Mode with  $\overline{SS}$  pin control enabled, (SSPCON<3:0> = 0100) the SPI module will reset if the  $\overline{SS}$  pin is set to VDD.

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

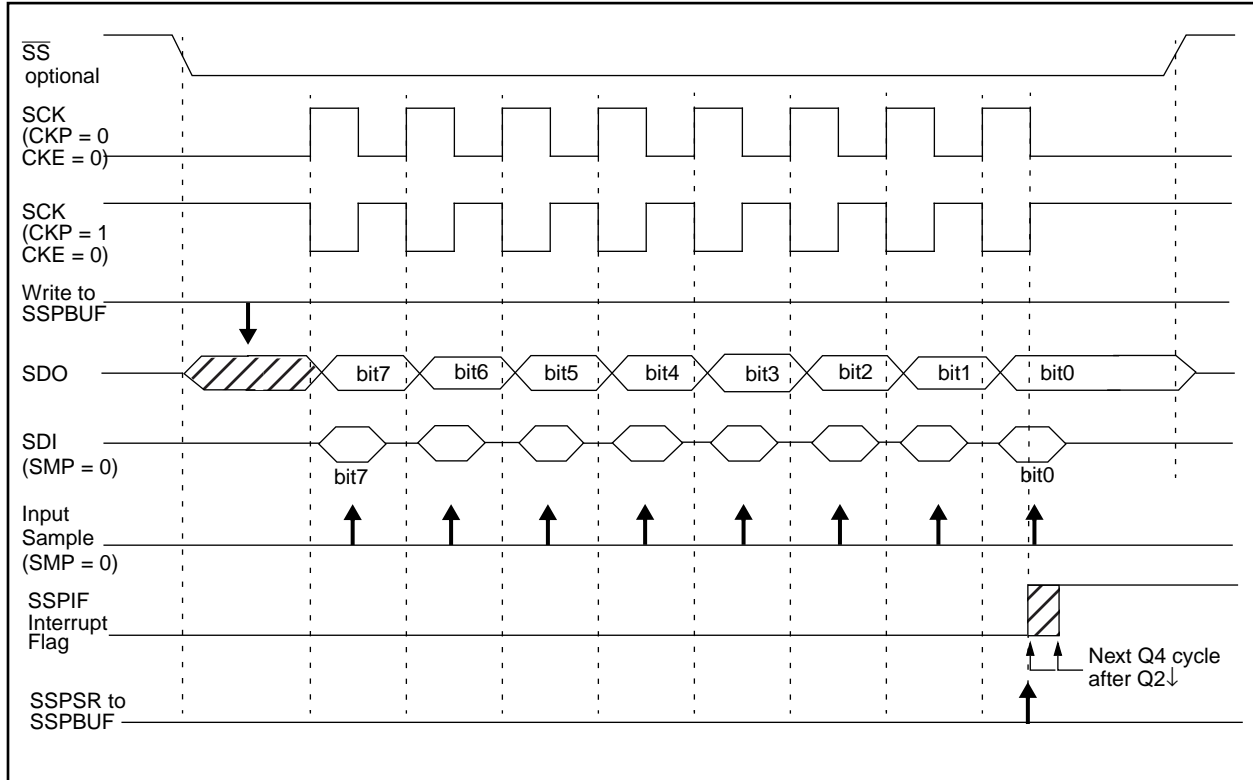
When the SPI module resets, the bit counter is forced to 0. This can be done by either forcing the  $\overline{SS}$  pin to a high level or clearing the SSPEN bit.

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

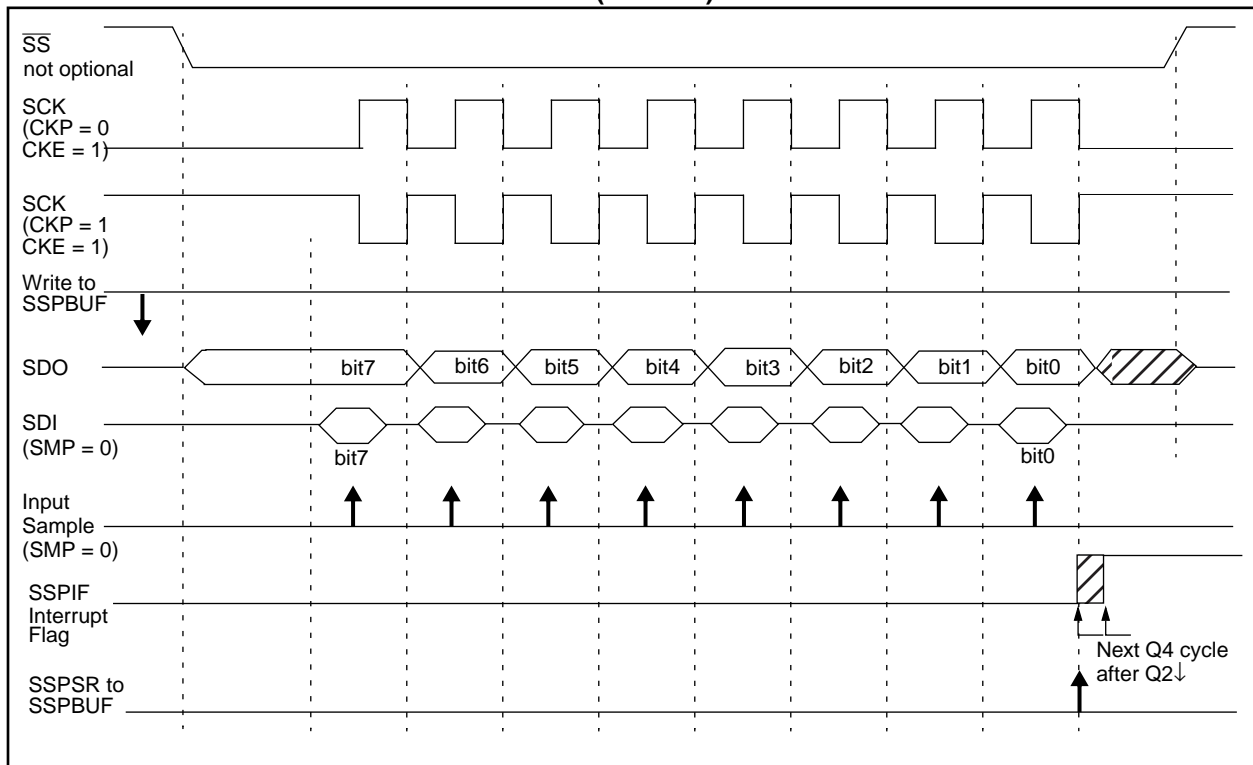
**FIGURE 9-7: SLAVE SYNCHRONIZATION WAVEFORM**



**FIGURE 9-8: SPI SLAVE MODE WAVEFORM (CKE = 0)**



**FIGURE 9-9: SPI SLAVE MODE WAVEFORM (CKE = 1)**



# PIC16F87X

## 9.1.7 SLEEP OPERATION

In master mode all module clocks are halted, and the transmission/reception will remain in that state until the device wakes from sleep. After the device returns to normal mode, the module will continue to transmit/receive data.

In slave mode, the SPI transmit/receive shift register operates asynchronously to the device. This allows the device to be placed in sleep mode, and data to be shifted into the SPI transmit/receive shift register. When all 8-bits have been received, the MSSP interrupt flag bit will be set and if enabled will wake the device from sleep.

## 9.1.8 EFFECTS OF A RESET

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

**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	TOIE	INTE	RBIE	TOIF	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.

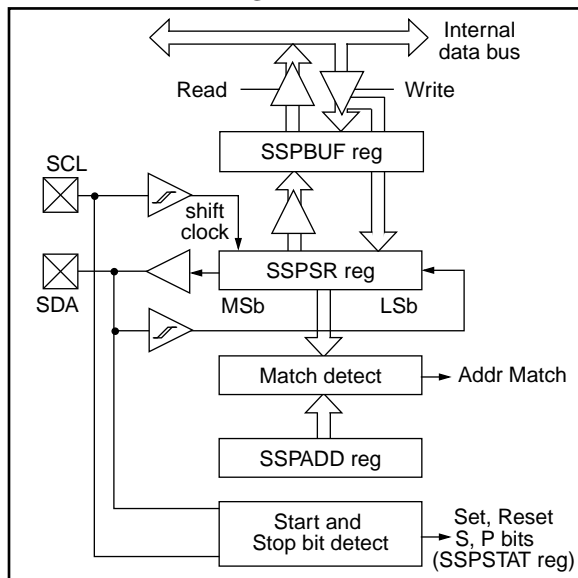
## 9.2 MSSP I<sup>2</sup>C Operation

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.

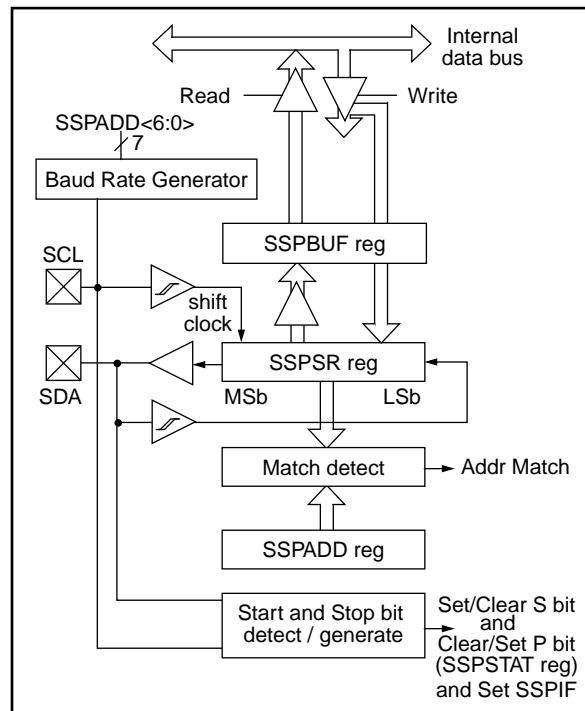
Refer to Application Note AN578, "Use of the SSP Module in the I<sup>2</sup>C Multi-Master Environment."

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 slew rate control of the pin that is independent of device frequency.

**FIGURE 9-10: I<sup>2</sup>C SLAVE MODE BLOCK DIAGRAM**



**FIGURE 9-11: I<sup>2</sup>C MASTER 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 that 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<67>) 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.

# PIC16F87X

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 bit UA (SSPSTAT<1>) are set).
2. Update the SSPADD register with 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  $R/\bar{W}$  bit of the address byte is clear and an address match occurs, the  $R/\bar{W}$  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 ( $\bar{ACK}$ ) 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  $\bar{ACK}$  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 $\bar{ACK}$ 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.

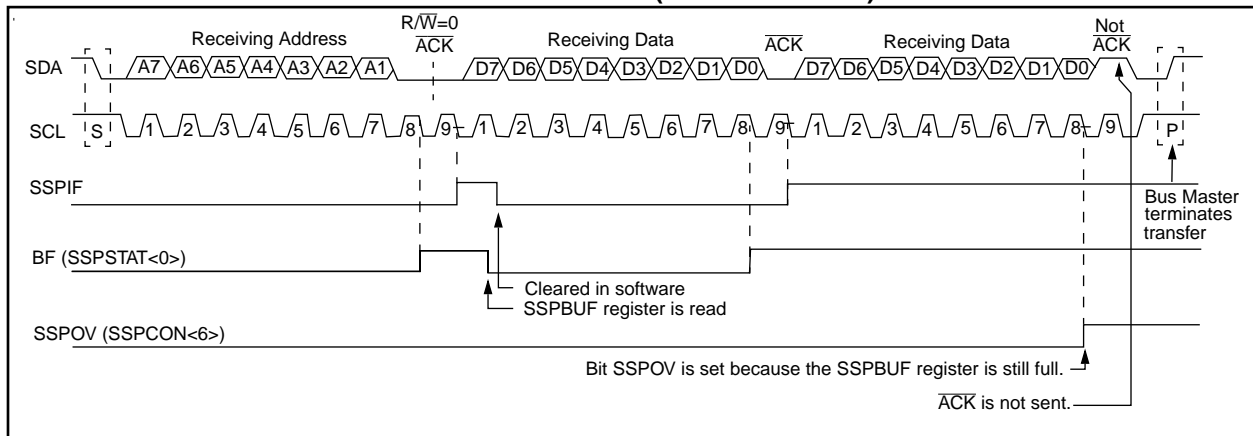
## 9.2.1.3 SLAVE TRANSMISSION

When the  $R/\bar{W}$  bit of the incoming address byte is set and an address match occurs, the  $R/\bar{W}$  bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The  $\bar{ACK}$  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 9-13).

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.

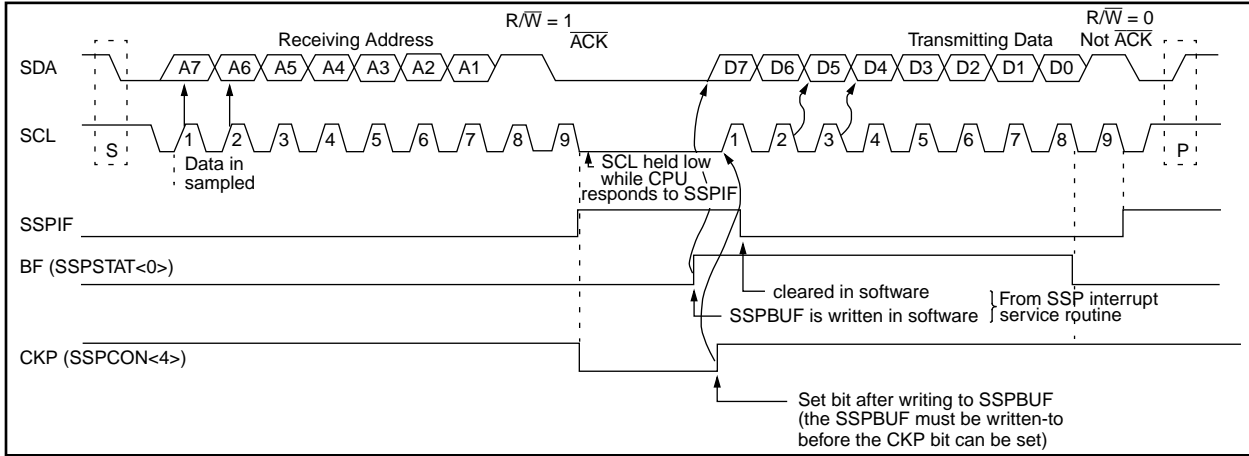
As a slave-transmitter, the  $\bar{ACK}$  pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not  $\bar{ACK}$ ), then the data transfer is complete. When the not  $\bar{ACK}$  is latched by the slave, the slave logic is reset and the slave then monitors for another occurrence of the START bit. If the SDA line was low ( $\bar{ACK}$ ), 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 9-12: I<sup>2</sup>C WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)**



# PIC16F87X

**FIGURE 9-13: I<sup>2</sup>C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)**



**FIGURE 9-14: I<sup>2</sup>C SLAVE-TRANSMITTER (10-BIT ADDRESS)**

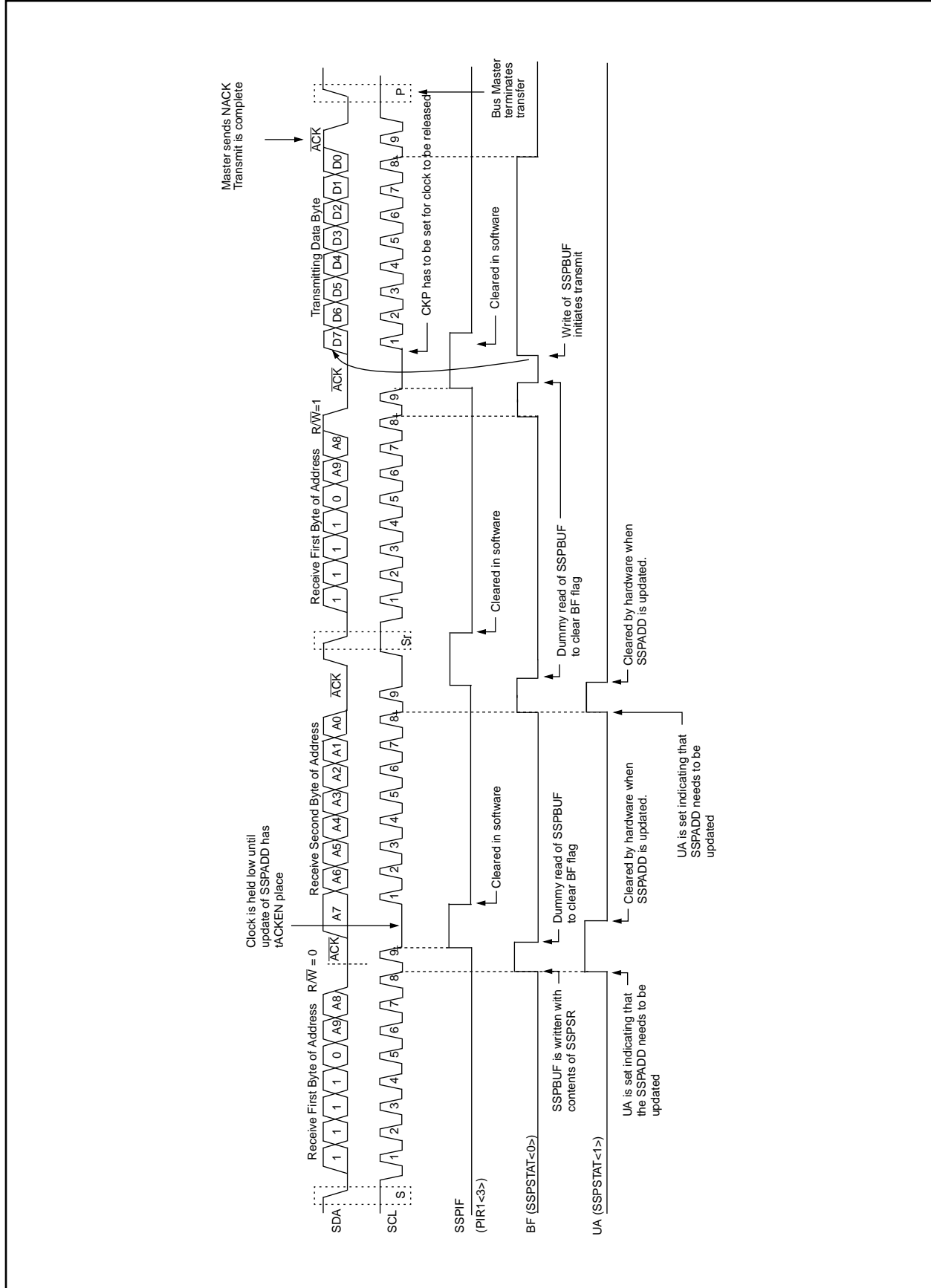
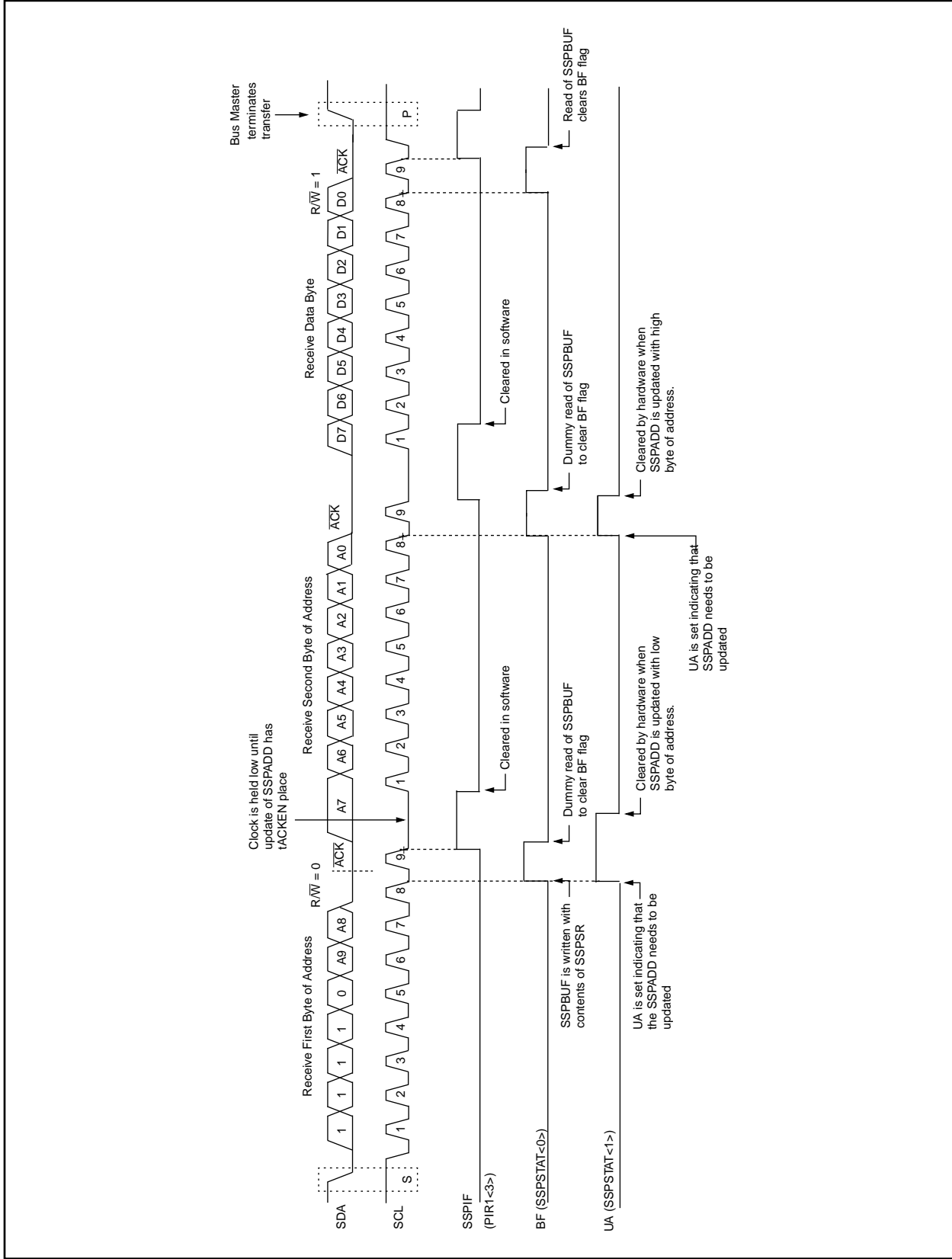


FIGURE 9-15: I<sup>2</sup>C SLAVE-RECEIVER (10-BIT ADDRESS)



## 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  $R/\bar{W} = 0$

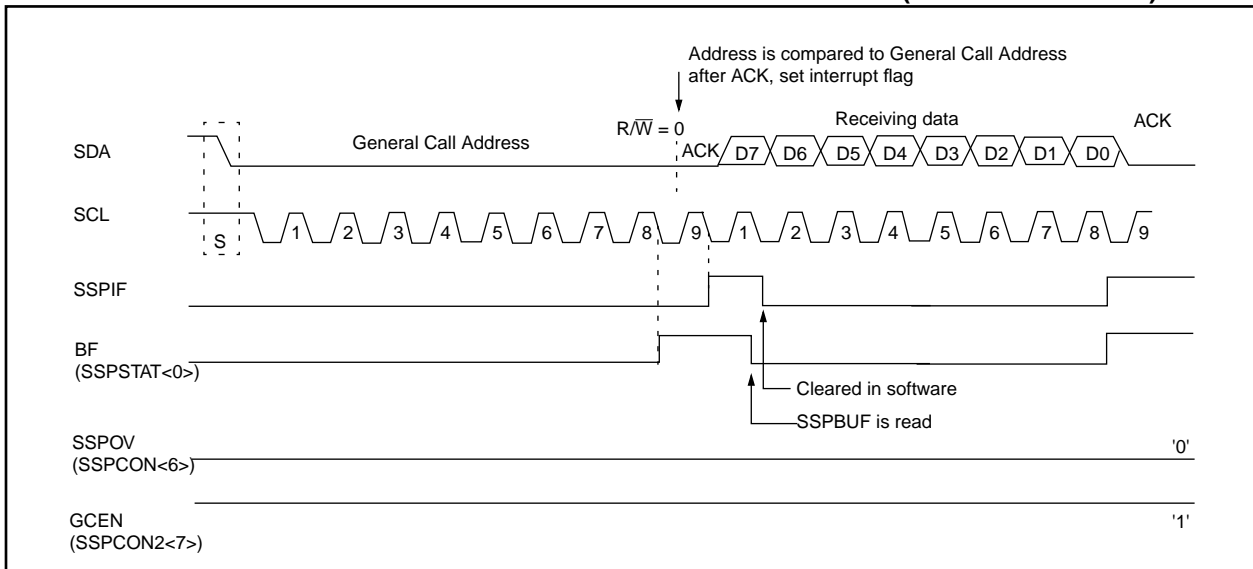
The general call address is recognized when the General Call Enable bit (GCEN) is enabled (SSPCON2<7> is set). Following a start-bit detect, 8-bits are shifted into SSPSR and the address is compared against SSPADD, and is also compared to the general call address, fixed in hardware.

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 9-16).

**FIGURE 9-16: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE (7 OR 10-BIT MODE)**



# PIC16F87X

## 9.2.3 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).

## 9.2.4 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.

## 9.2.5 MASTER MODE

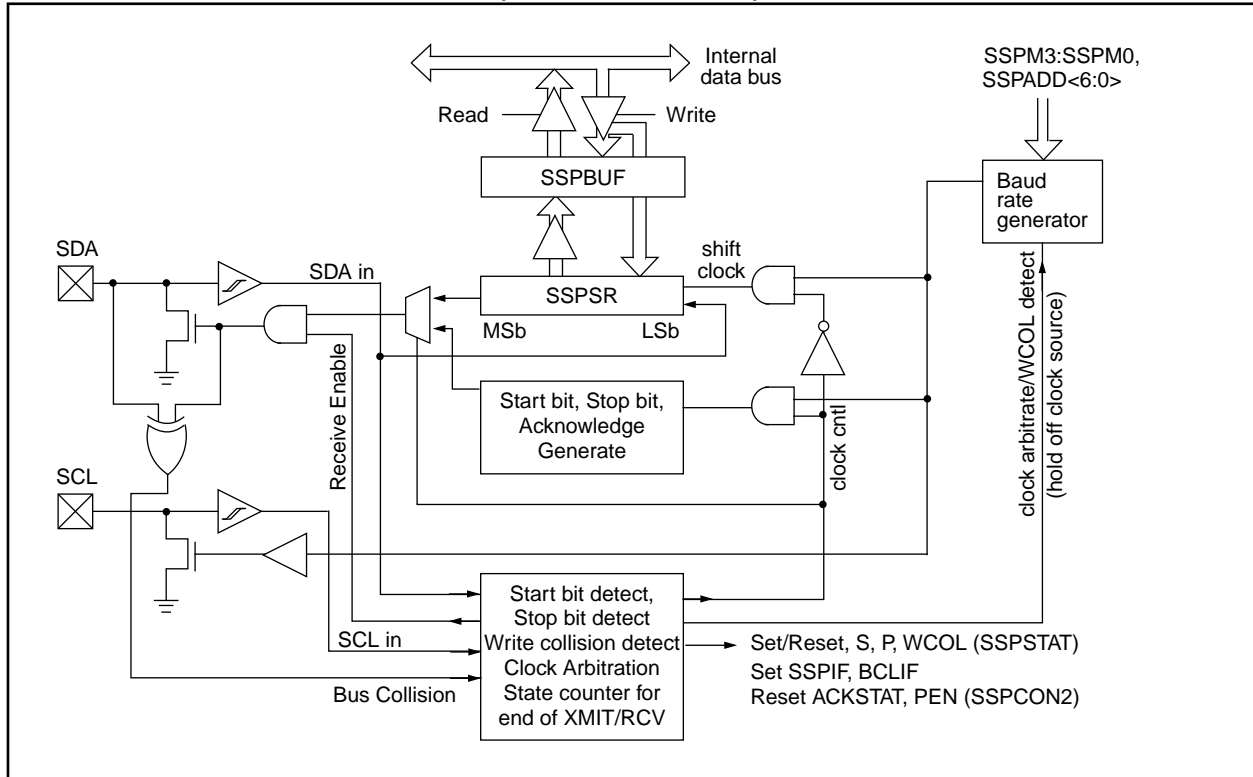
Master mode of operation is supported by interrupt generation on the detection of the START and STOP conditions. The STOP (P) and START (S) bits are cleared from a reset or when the MSSP module is disabled. Control of the I<sup>2</sup>C bus may be TAKEN when the P bit is set, or the bus is idle with 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 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 9-17: 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.

## 9.2.7.4 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.



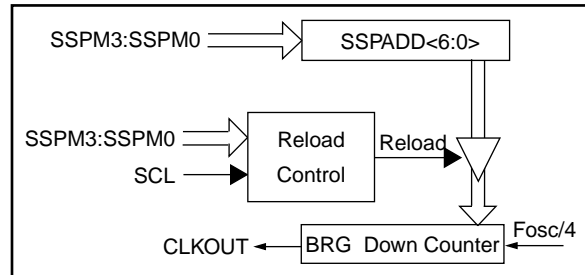
- i) The MSSP Module shifts in the ACK bit from the slave device, and writes its value into the SSPCON2 register ( SSPCON2<6>).
- j) The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
- k) The user generates a STOP condition by setting the STOP enable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

## 9.2.8 BAUD RATE GENERATOR

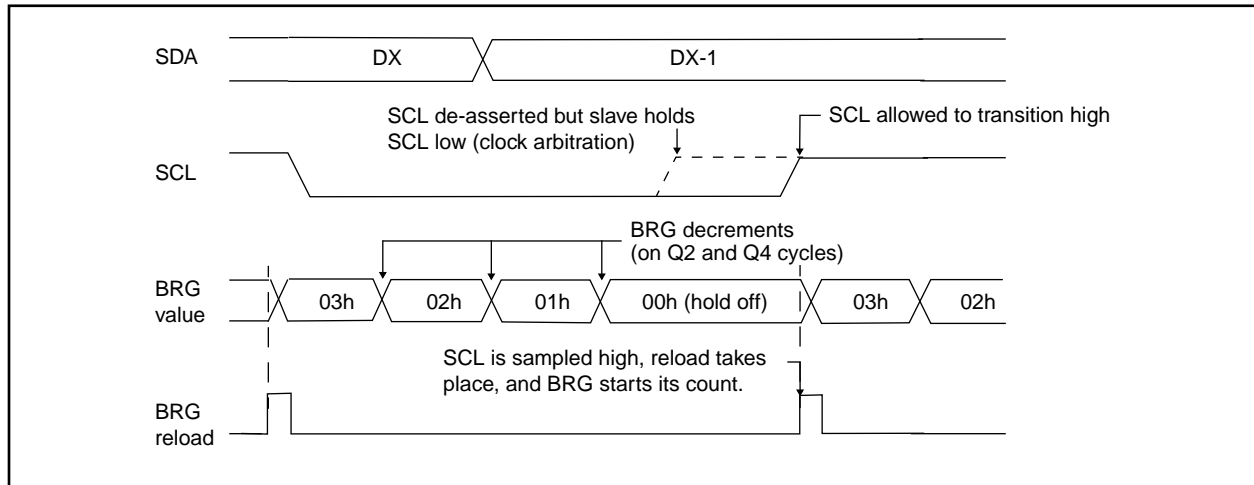
In I<sup>2</sup>C master mode, the reload value for the BRG is located in the lower 7 bits of the SSPADD register (Figure 9-18). When the BRG is loaded with this value, the BRG counts down to 0 and stops until another reload has TAKEN place. The BRG count is decremented twice per instruction cycle (TCY), 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 9-19).

**FIGURE 9-18: BAUD RATE GENERATOR BLOCK DIAGRAM**



**FIGURE 9-19: 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 ( $T_{BRG}$ ), 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 ( $T_{BRG}$ ) 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.

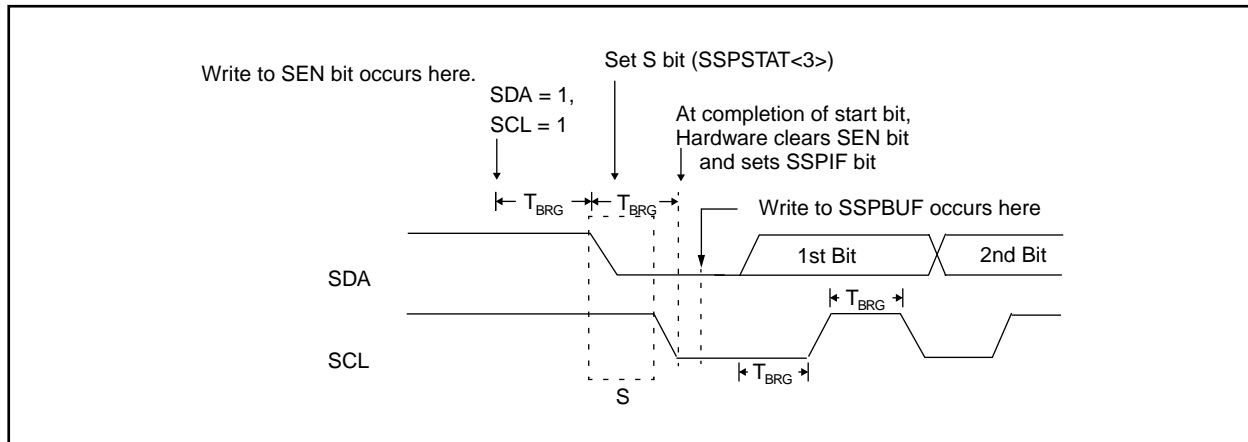
**Note:** If at the beginning of START condition the SDA and SCL pins are already sampled low, or if during the START condition the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag (BCLIF) is set, the START condition is aborted, and the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

### 9.2.9.5 WCOL STATUS FLAG

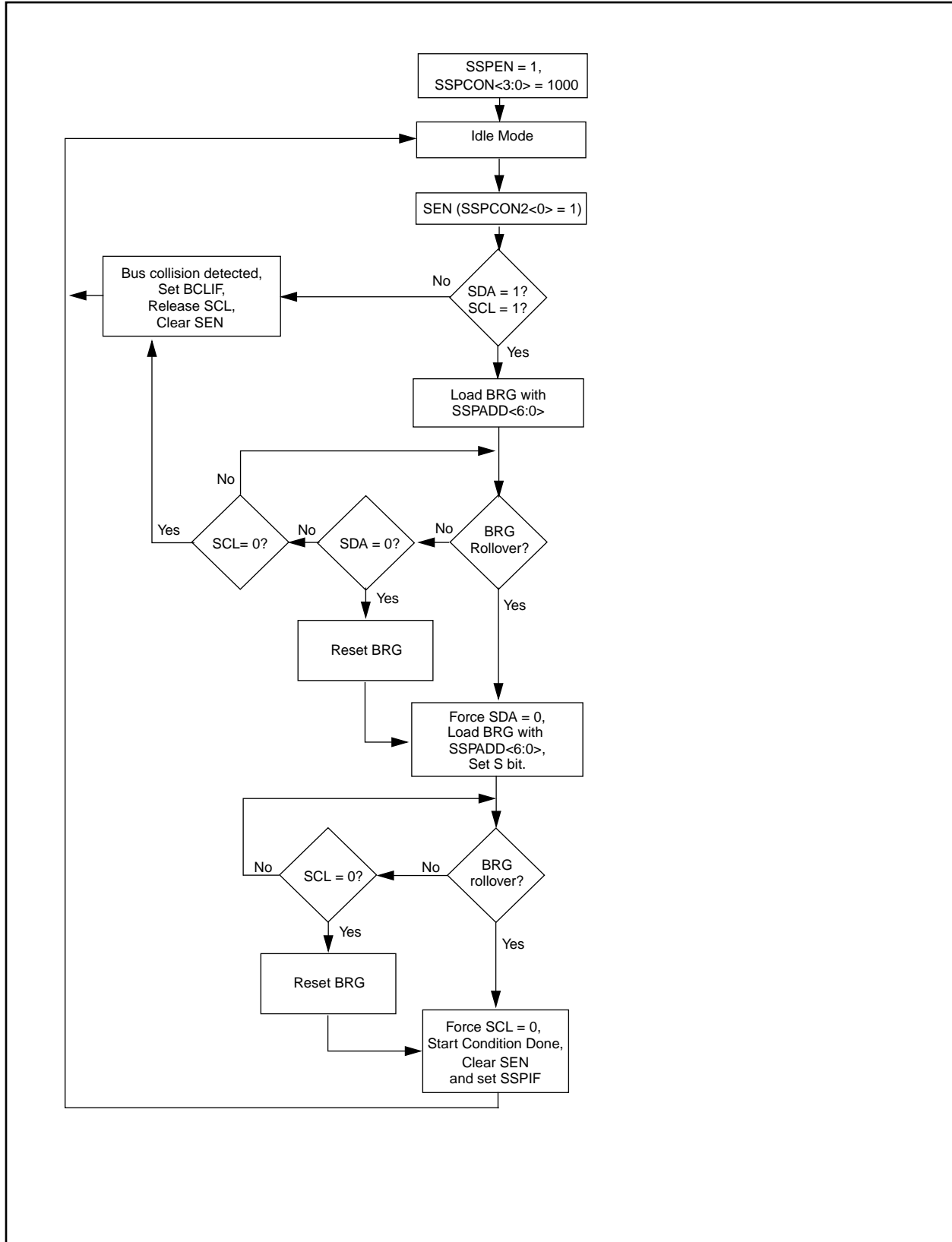
If the user writes the SSPBUF when an START sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

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

**FIGURE 9-20: FIRST START BIT TIMING**



**FIGURE 9-21: START CONDITION FLOWCHART**



# PIC16F87X

## 9.2.10 I<sup>2</sup>C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition occurs when the RSEN bit (SSPCON2<1>) is programmed high and the I<sup>2</sup>C module is in the idle state. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the baud rate generator is loaded with the contents of SSPADD<6:0>, and begins counting. The SDA pin is released (brought high) for one baud rate generator count ( $T_{BRG}$ ). When the baud rate generator times out, if SDA is sampled high, the SCL pin will be de-asserted (brought high). When SCL is sampled high the baud rate generator is re-loaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one  $T_{BRG}$ . This action is then followed by assertion of the SDA pin (SDA is low) for one  $T_{BRG}$  while SCL is high. Following this, the RSEN bit in the SSPCON2 register will be automatically cleared, and the baud rate generator is not reloaded, leaving the SDA pin held low. As soon as a start con-

dition is detected on the SDA and SCL pins, the S bit (SSPSTAT<3>) will be set. The SSPIF bit will not be set until the baud rate generator has timed-out.

**Note 1:** If RSEN is programmed while any other event is in progress, it will not take effect.

**Note 2:** A bus collision during the Repeated Start condition occurs if:

- SDA is sampled low when SCL goes from low to high.
- SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data "1".

Immediately following the SSPIF bit getting set, the user may write the SSPBUF with the 7-bit address in 7-bit mode, or the default first address in 10-bit mode. After the first eight bits are transmitted and an ACK is received, the user may then transmit an additional eight bits of address (10-bit mode) or eight bits of data (7-bit mode).

### 9.2.10.6 WCOL STATUS FLAG

If the user writes the SSPBUF when a Repeated 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 of the lower 5 bits of SSPCON2 is disabled until the Repeated Start condition is complete.

**FIGURE 9-22: REPEAT START CONDITION WAVEFORM**

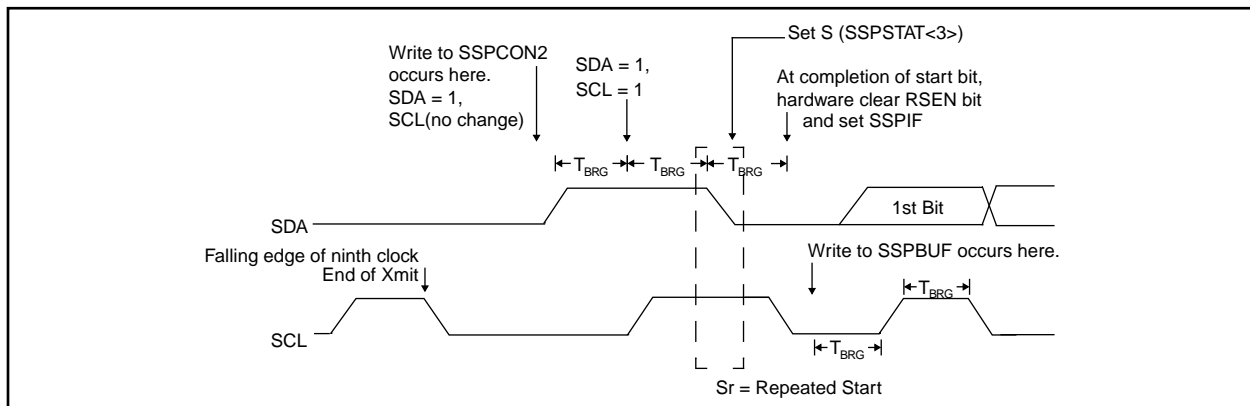


FIGURE 9-23: REPEATED START CONDITION FLOWCHART (PAGE 1)

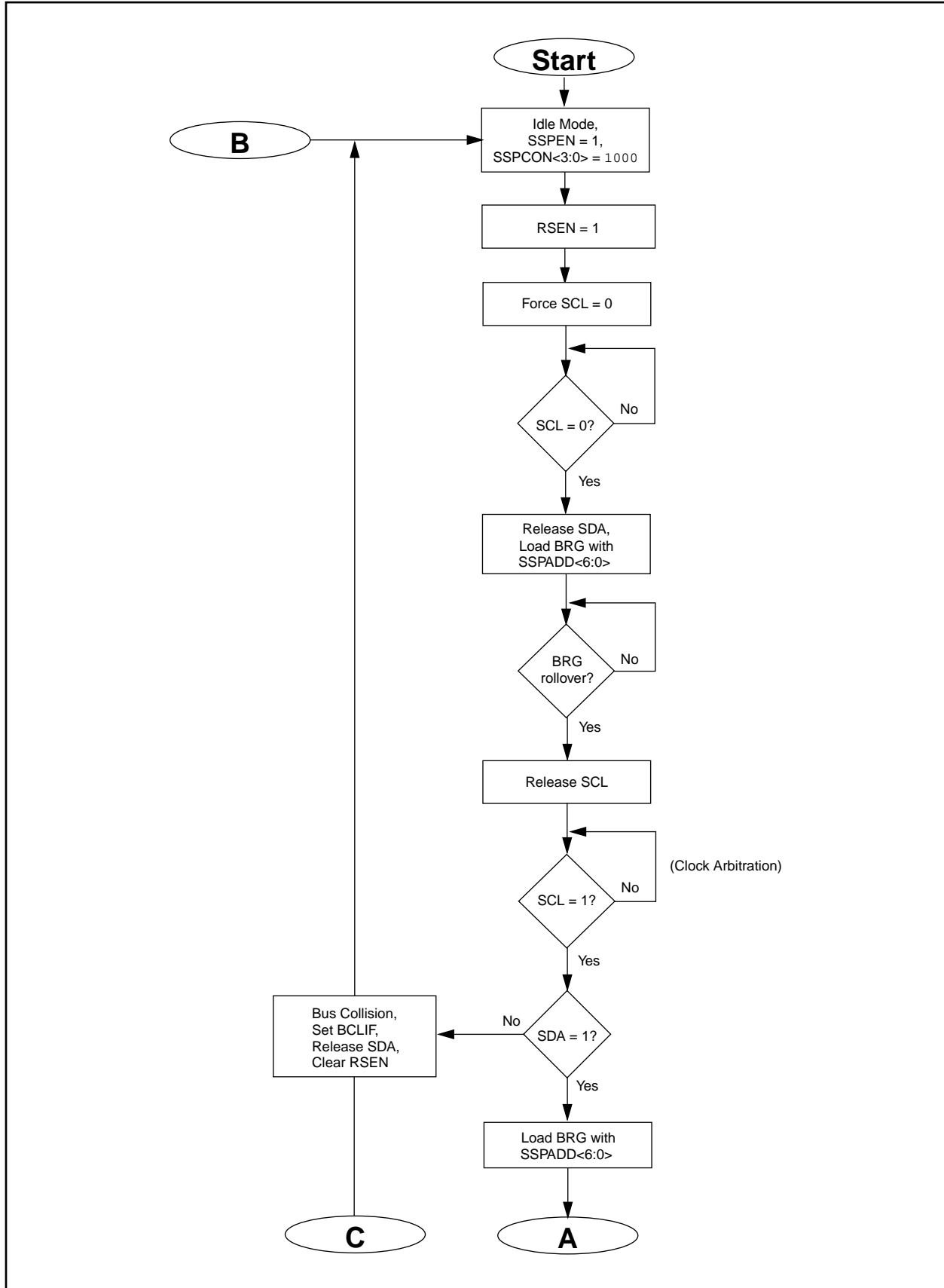
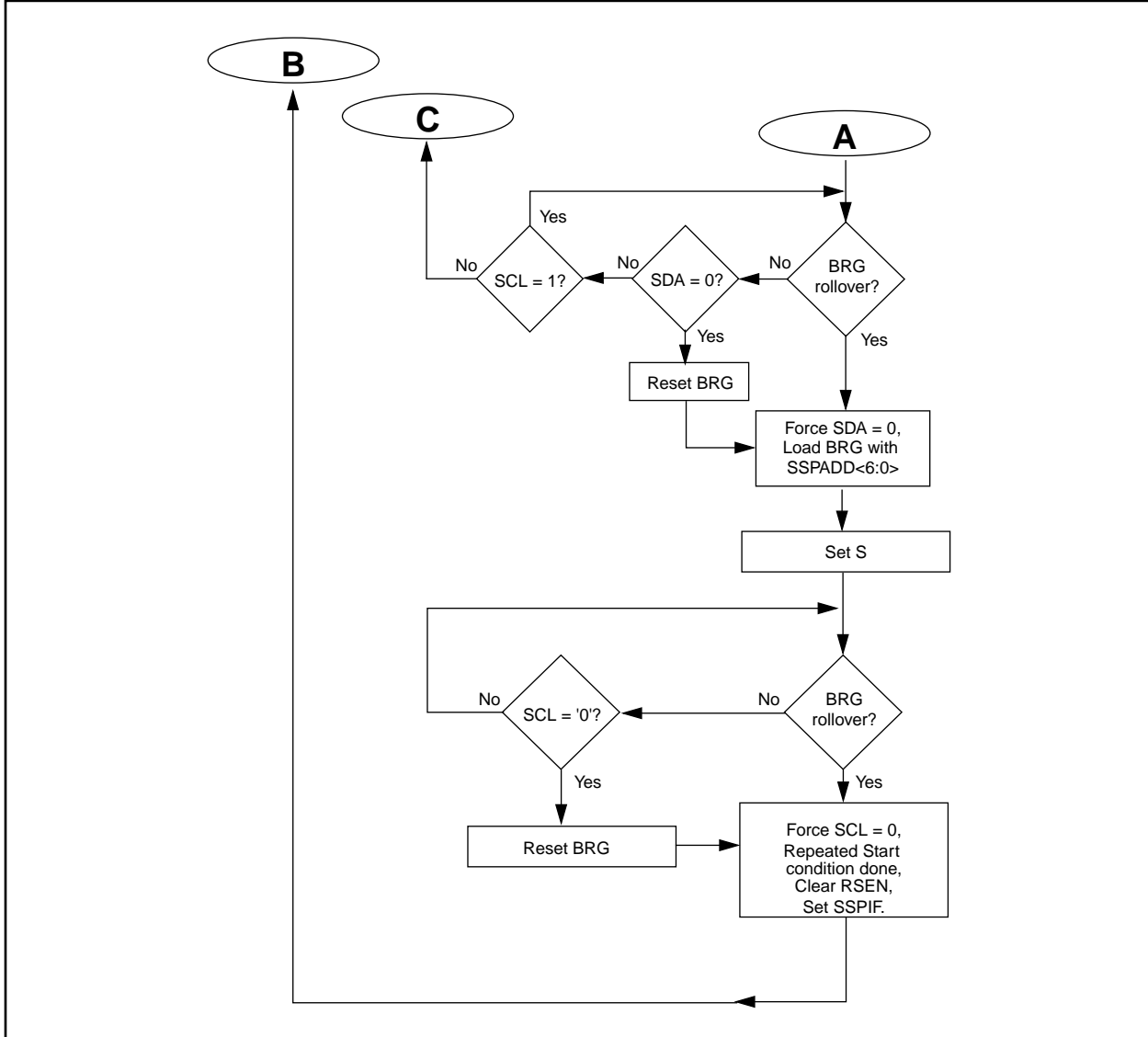


FIGURE 9-24: REPEATED START CONDITION FLOWCHART (PAGE 2)



## 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 roll over count ( $T_{BRG}$ ). 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  $T_{BRG}$ , 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  $\overline{ACK}$  bit during the ninth bit time, if an address match occurs or if data was received properly. The status of  $\overline{ACK}$  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 9-26).

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  $R/\overline{W}$  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.

### 9.2.11.7 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.

### 9.2.11.8 WCOL STATUS FLAG

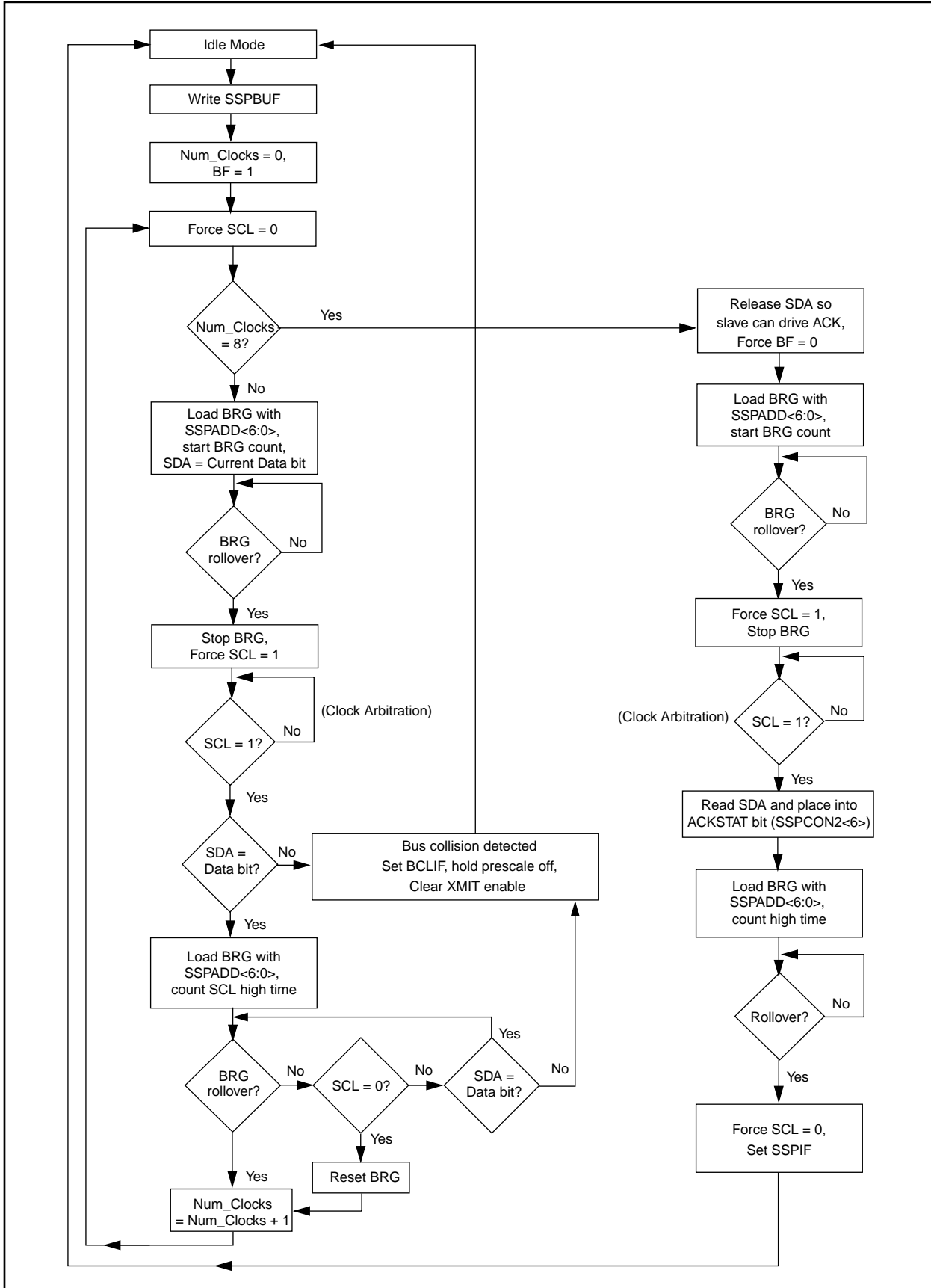
If the user writes the SSPBUF when a transmit is already in progress (i.e. SSPSR is still shifting out a data byte), then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

WCOL must be cleared in software.

### 9.2.11.9 ACKSTAT STATUS FLAG

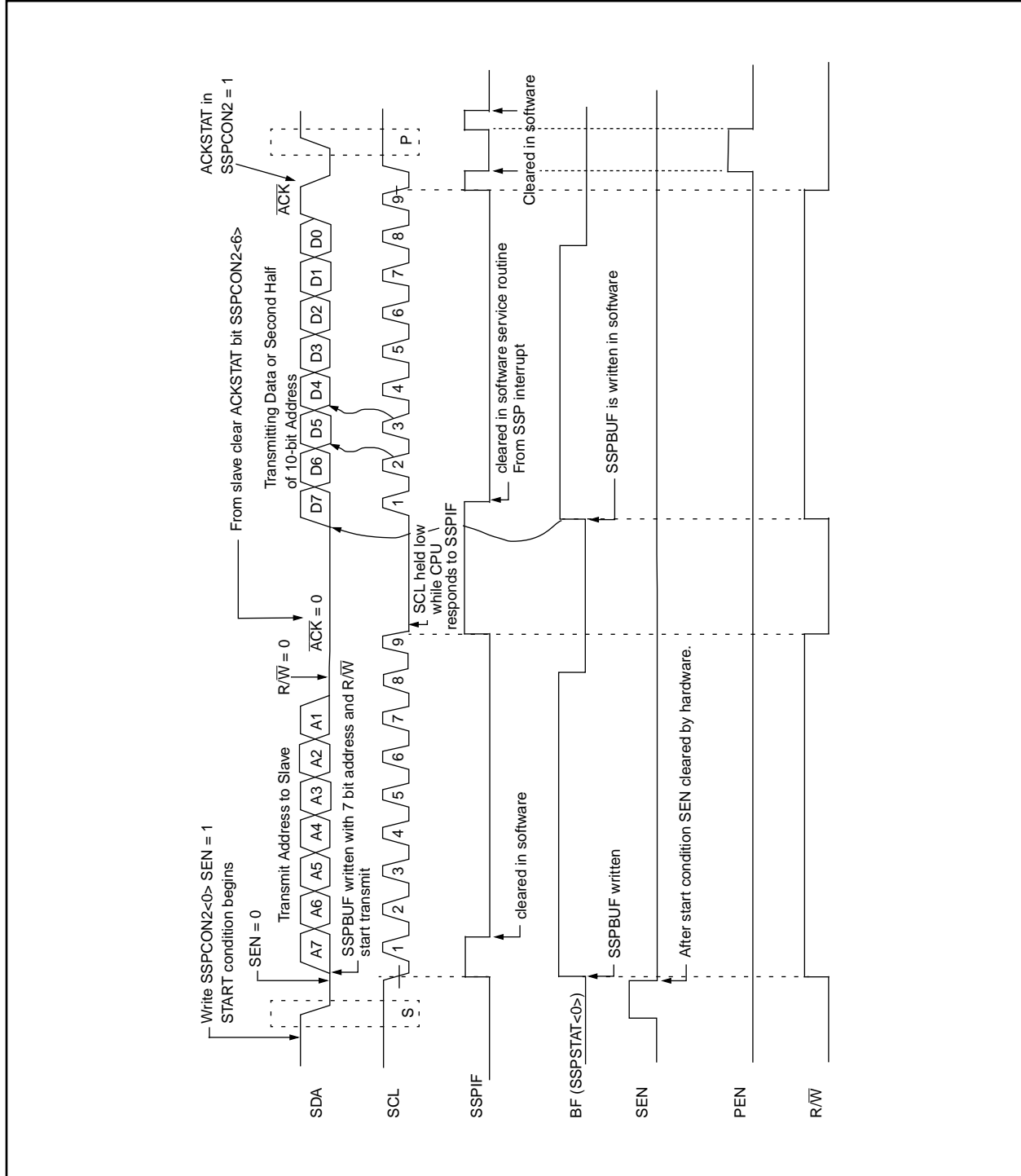
In transmit mode, the ACKSTAT bit (SSPCON2<6>) is cleared when the slave has sent an acknowledge ( $\overline{ACK} = 0$ ), and is set when the slave does not acknowledge ( $\overline{ACK} = 1$ ). A slave sends an acknowledge when it has recognized its address (including a general call), or when the slave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

FIGURE 9-25: MASTER TRANSMIT FLOWCHART





**FIGURE 9-26: 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>).

**Note:** The SSP Module must be in an **IDLE STATE** before the RCEN bit is set, or the RCEN bit will be disregarded.

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>).

### 9.2.12.10 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.

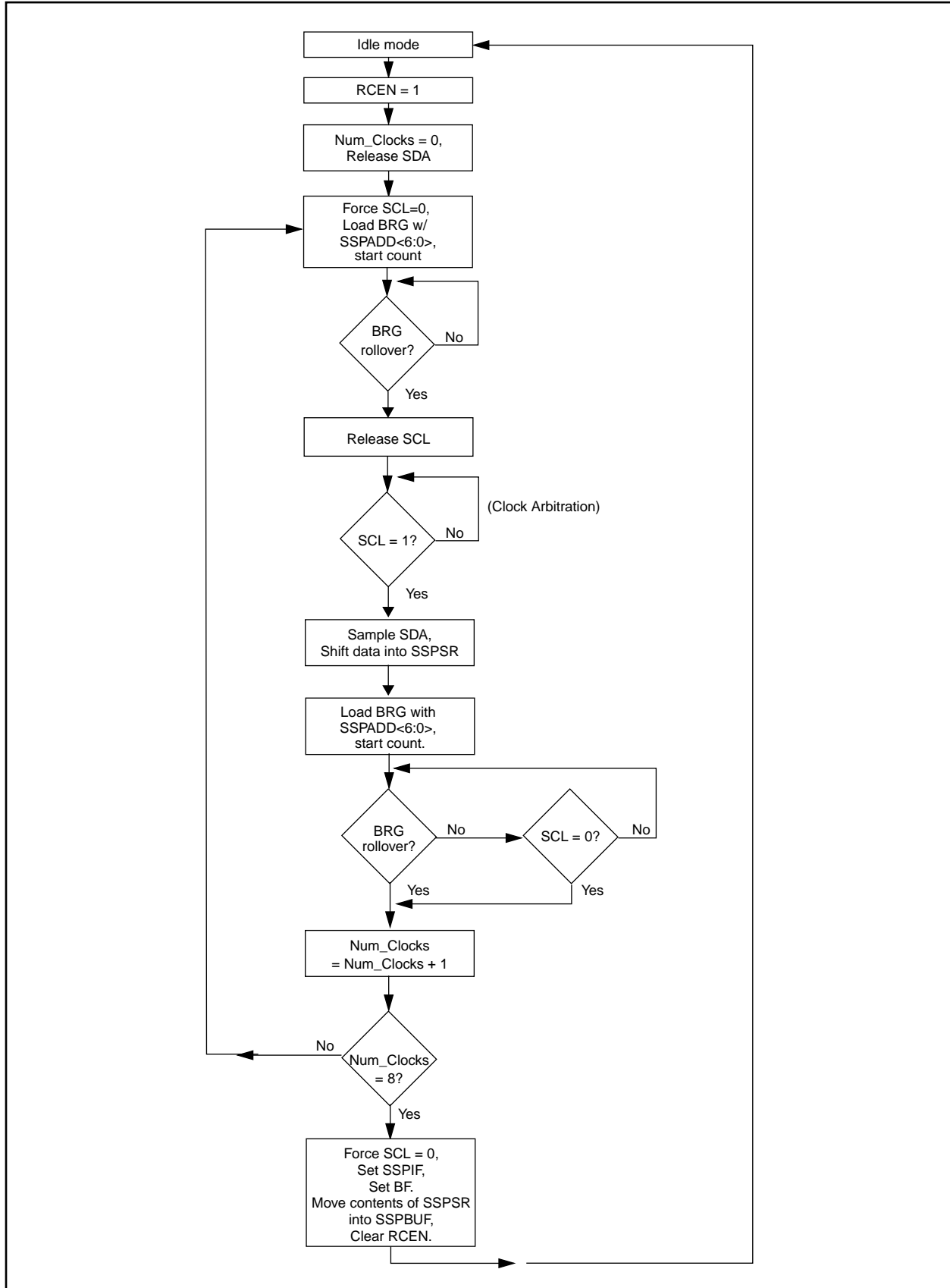
### 9.2.12.11 SSPOV STATUS FLAG

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.

### 9.2.12.12 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).

**FIGURE 9-27: MASTER RECEIVER FLOWCHART**





## 9.2.13 ACKNOWLEDGE SEQUENCE TIMING

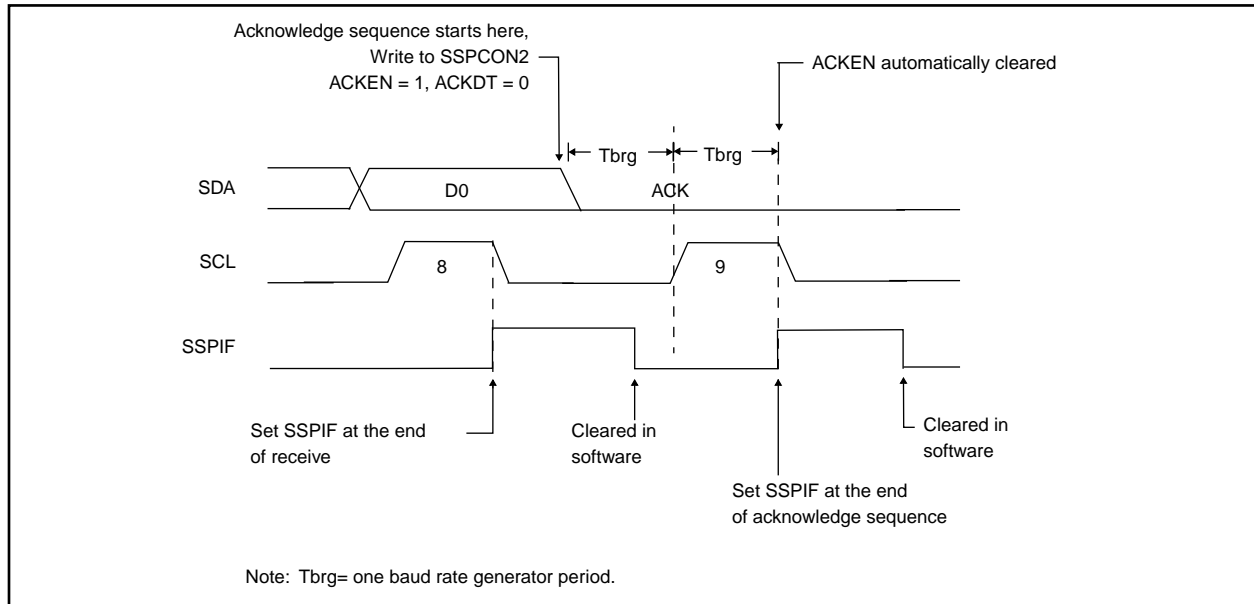
An acknowledge sequence is enabled by setting the acknowledge sequence enable bit, ACKEN (SSPCON2<4>). When this bit is set, the SCL pin is pulled low and the contents of the acknowledge data bit is presented on the SDA pin. If the user wishes to generate an acknowledge, then the ACKDT bit should be 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 de-asserted (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 9-29)

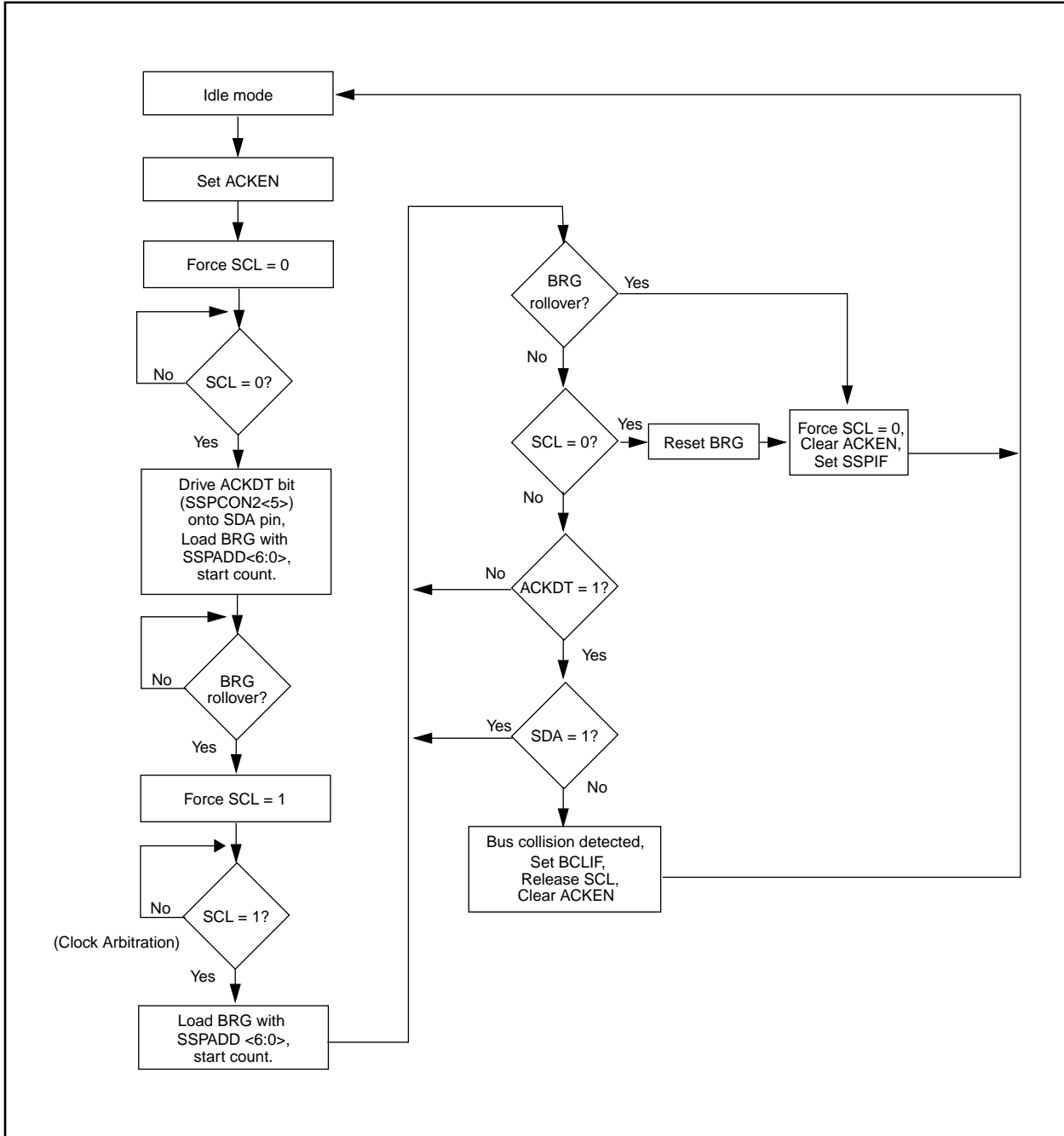
### 9.2.13.13 WCOL STATUS FLAG

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

**FIGURE 9-29: ACKNOWLEDGE SEQUENCE WAVEFORM**



**FIGURE 9-30: ACKNOWLEDGE FLOWCHART**



## 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  $T_{BRG}$  (baud rate generator rollover count) later, the SDA pin will be de-asserted. When the SDA pin is sampled high

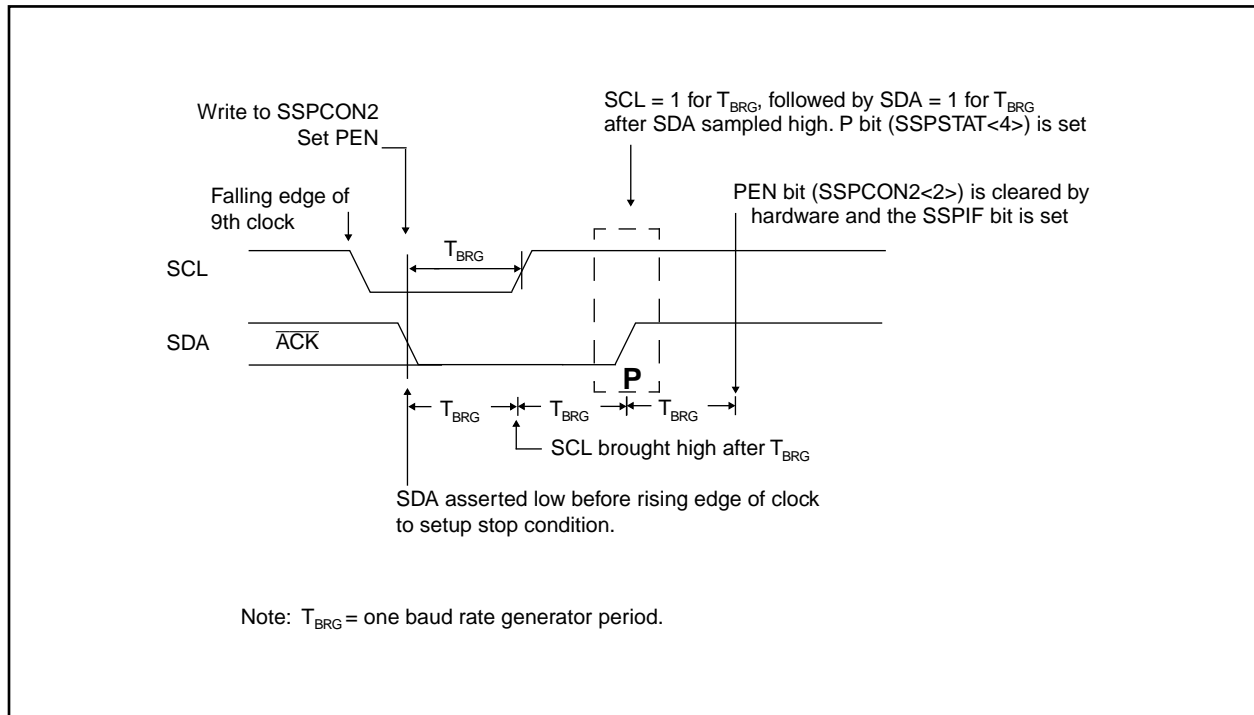
while SCL is high, the P bit (SSPSTAT<4>) is set. A  $T_{BRG}$  later the PEN bit is cleared and the SSPIF bit is set (Figure 9-31).

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).

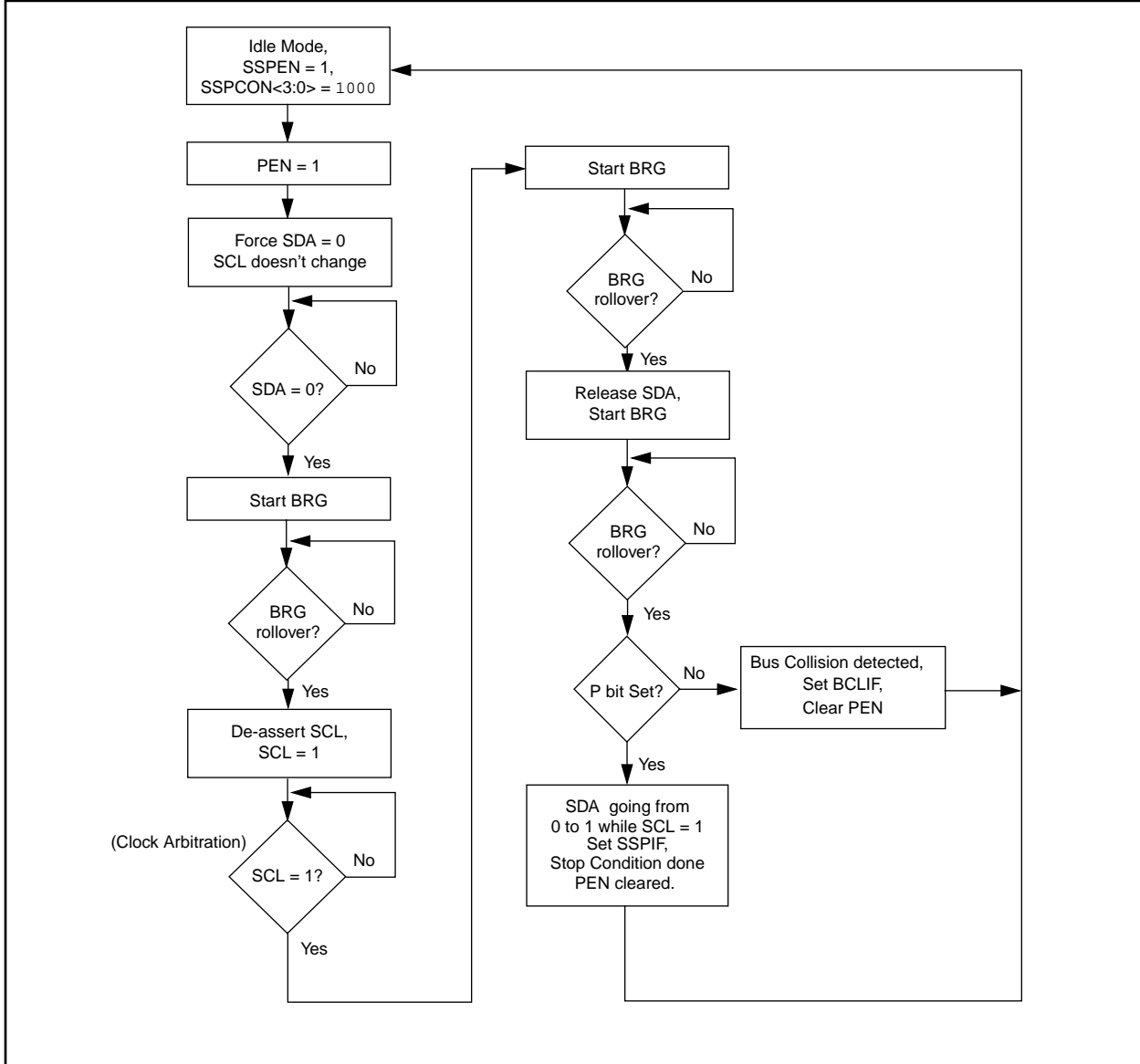
### 9.2.14.14 WCOL STATUS FLAG

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 9-31: STOP CONDITION RECEIVE OR TRANSMIT MODE**



**FIGURE 9-32: STOP CONDITION FLOWCHART**





## 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 9-33).

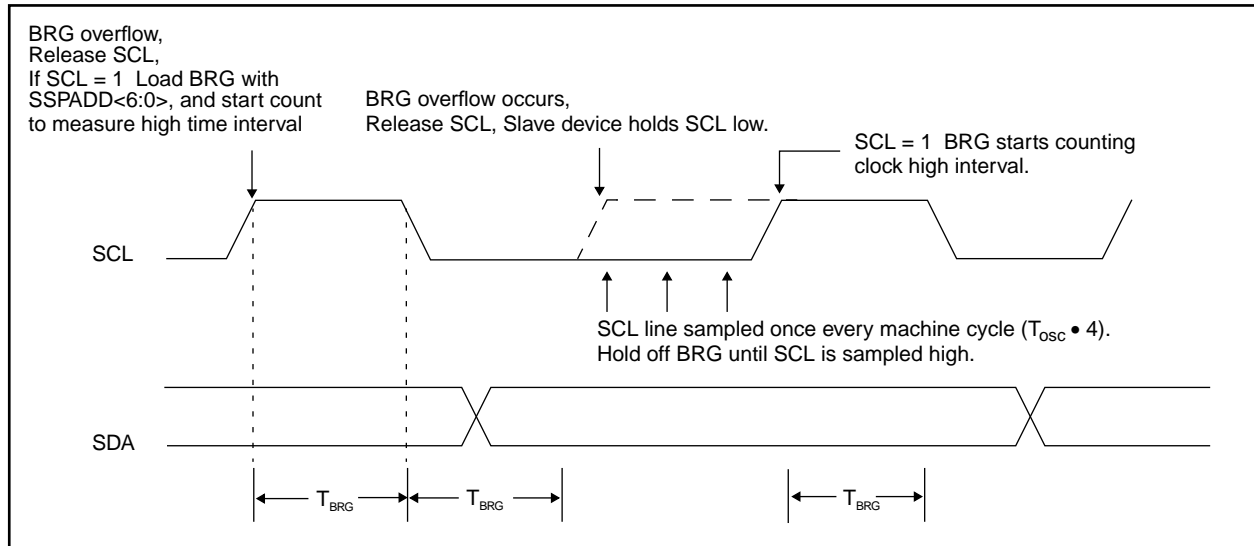
## 9.2.16 SLEEP OPERATION

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 9-33: 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', then 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 9-34).

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 de-asserted, 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.

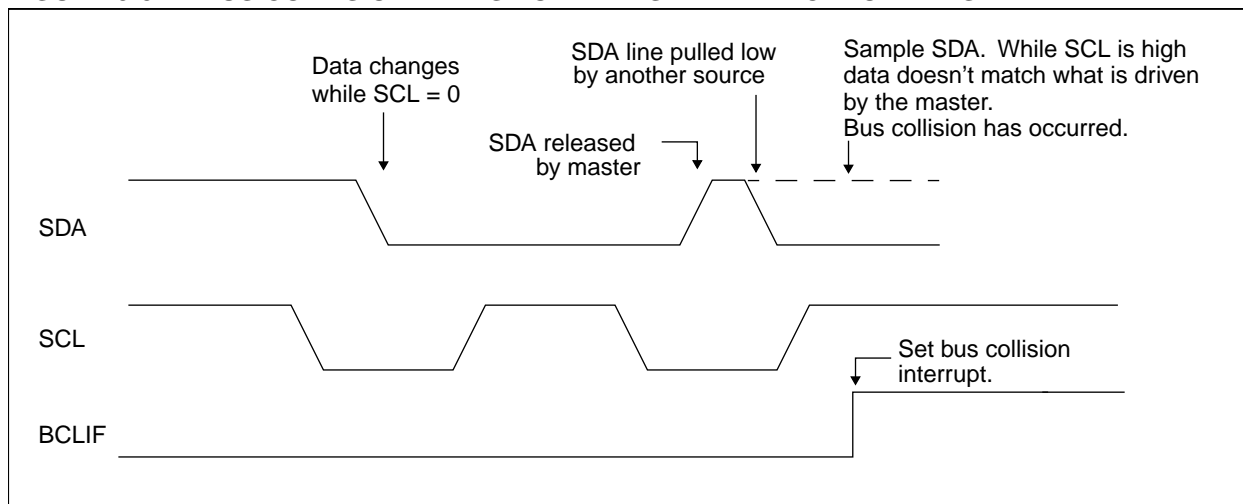
If a START, Repeated Start, STOP, or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are de-asserted, and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision interrupt service 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 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 9-34: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE**



## 9.2.18.15 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 9-35).
- SCL is sampled low before SDA is asserted low. (Figure 9-36).

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 9-35).

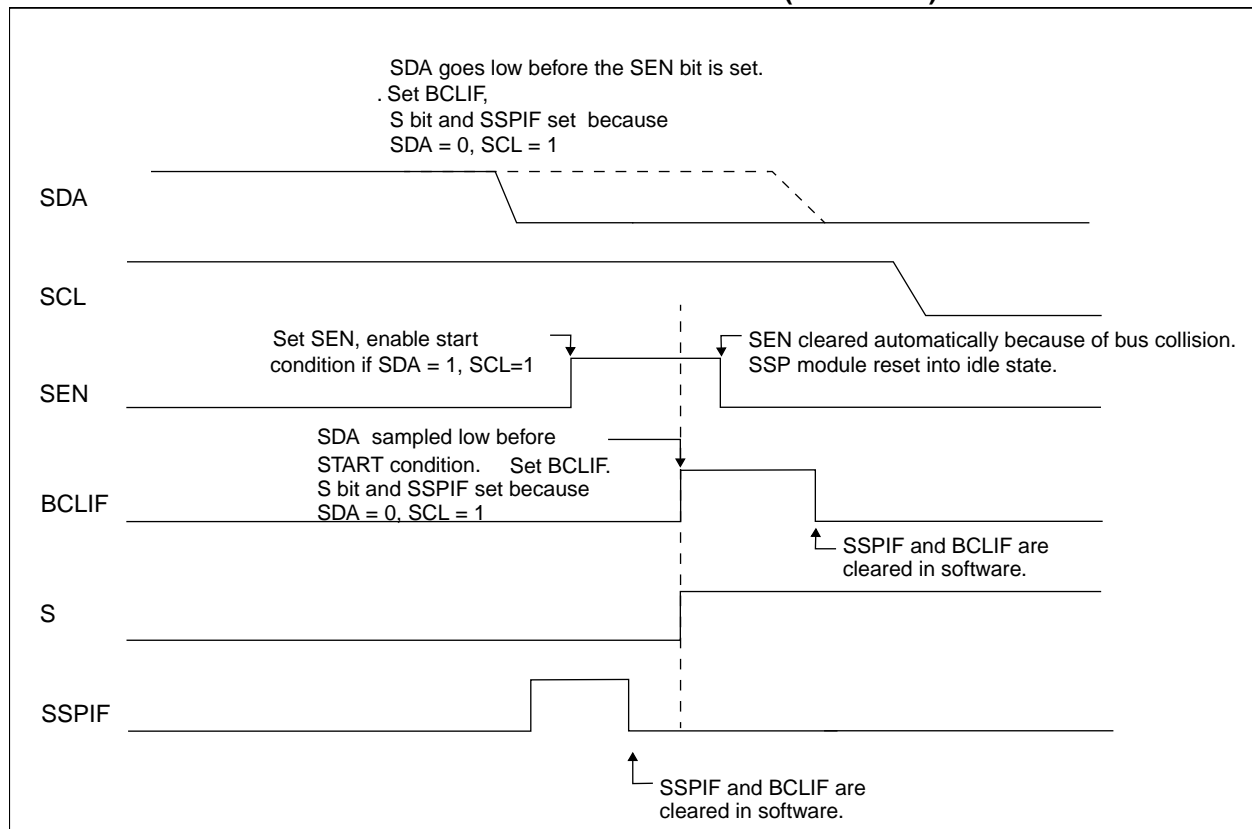
The START condition begins with the SDA and SCL pins de-asserted. When the SDA pin is sampled high, the baud rate generator is loaded from SSPADD<6:0> and counts down to 0. If the SCL pin is sampled low

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.

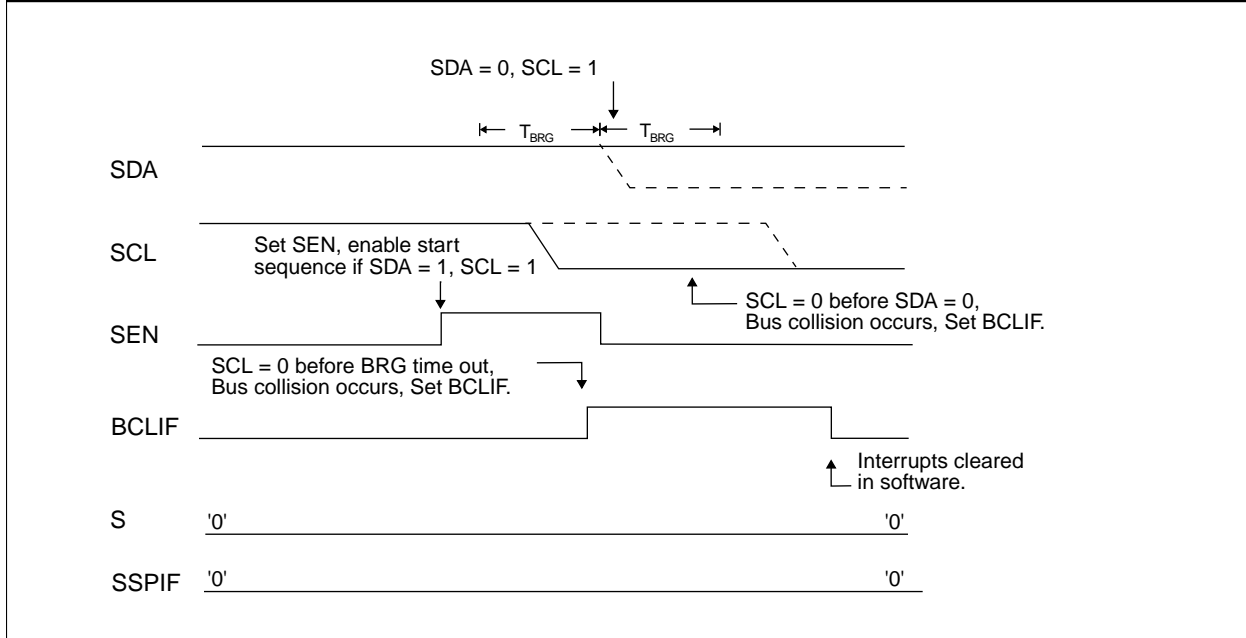
If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 9-37). If however a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The baud rate generator is then reloaded and counts down to 0, and during this time, if the SCL pins is 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, and if the address is the same, arbitration must be allowed to continue into the data portion, REPEATED START, or STOP conditions.

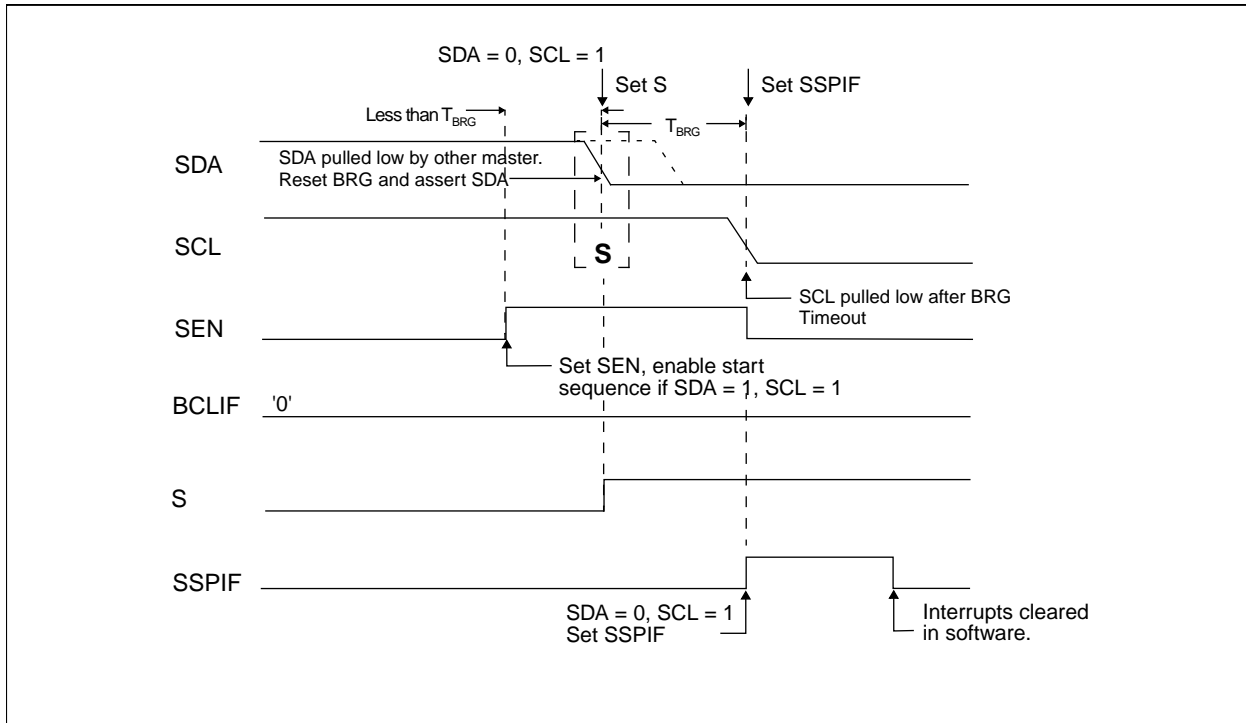
**FIGURE 9-35: BUS COLLISION DURING START CONDITION (SDA ONLY)**



**FIGURE 9-36: BUS COLLISION DURING START CONDITION (SCL = 0)**



**FIGURE 9-37: BRG RESET DUE TO SDA COLLISION DURING START CONDITION**



## 9.2.18.16 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'.

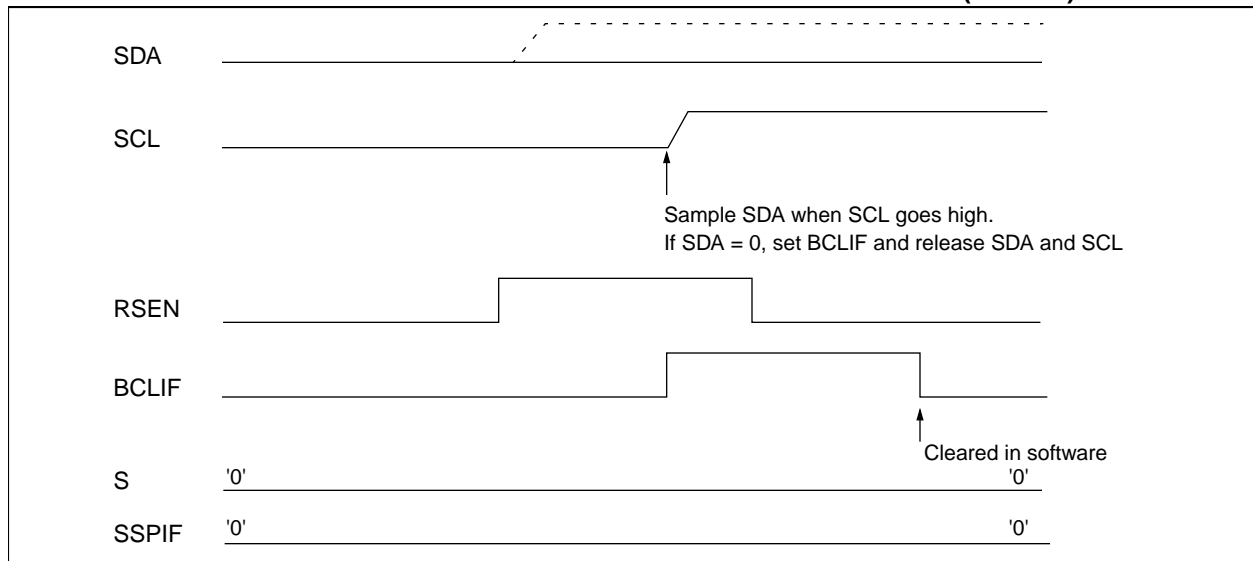
When the user de-asserts SDA and the pin is allowed to float high, the BRG is loaded with SSPADD<6:0>, and counts down to 0. The SCL pin is then de-asserted, and when sampled high, the SDA pin is sampled. If SDA is low, a bus collision has occurred (i.e. another master is attempting to transmit a data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

however SDA is sampled high then 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.

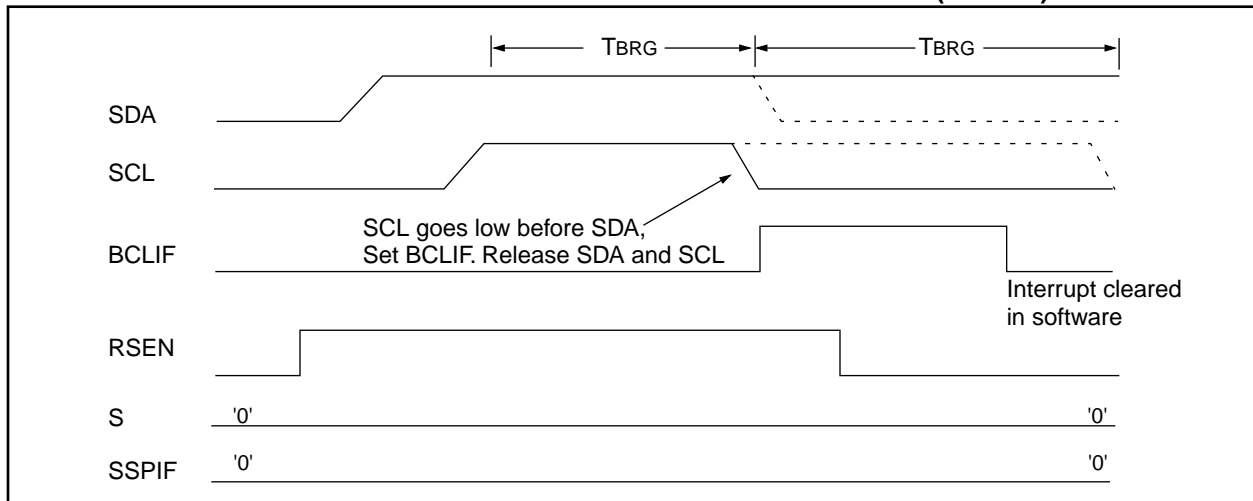
If, however, SCL goes from high to low before the BRG times out and SDA has not already been asserted, then a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition.

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 9-38).

**FIGURE 9-38: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)**



**FIGURE 9-39: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)**



# PIC16F87X

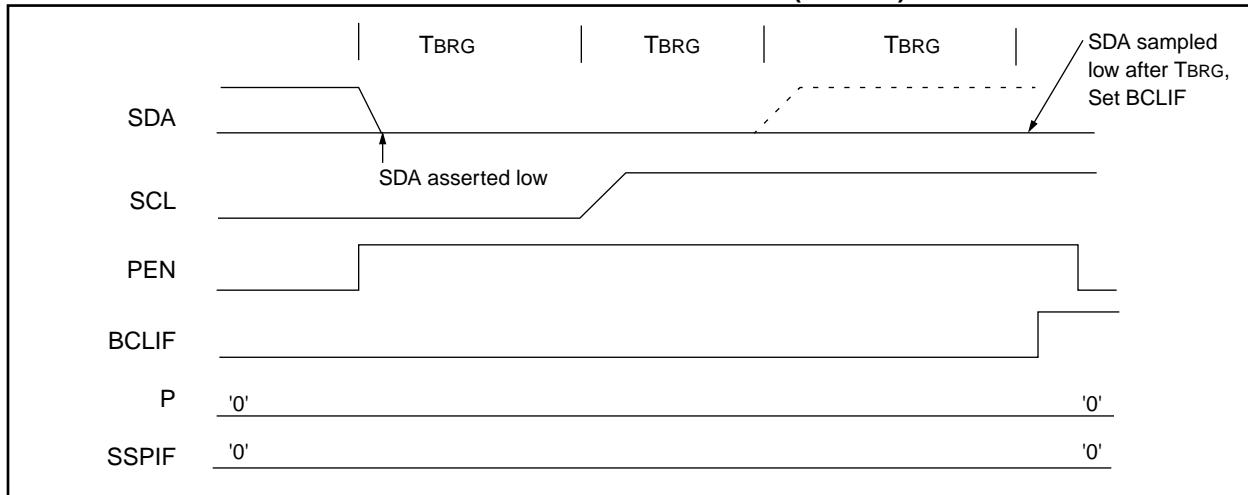
## 9.2.18.17 BUS COLLISION DURING A STOP CONDITION

Bus collision occurs during a STOP condition if:

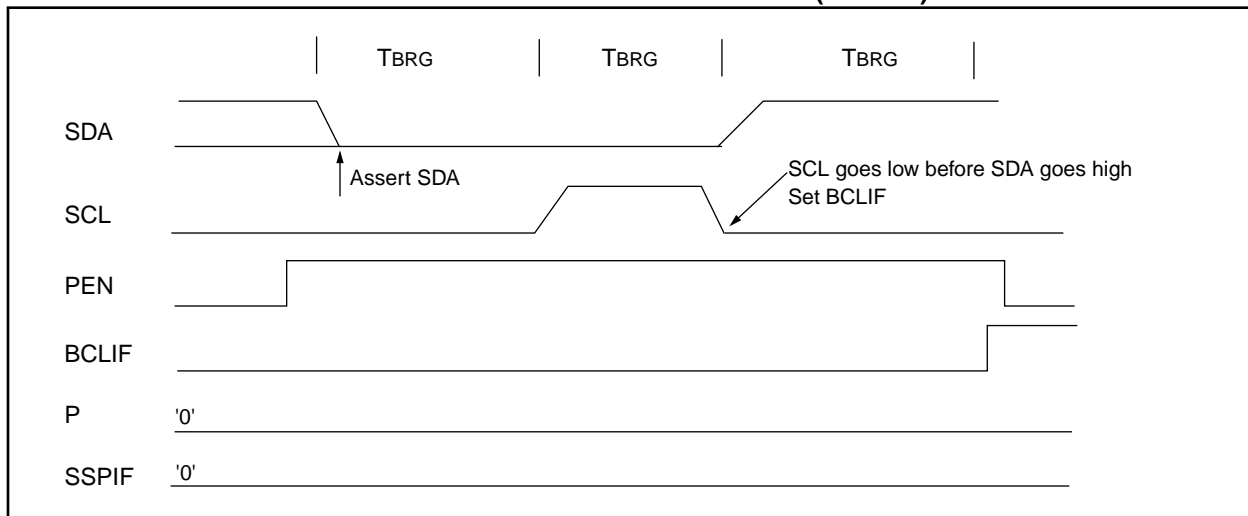
- After the SDA pin has been de-asserted and allowed to float high, SDA is sampled low after the BRG has timed out.
- After the SCL pin is de-asserted, SCL is sampled low before SDA goes high.

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 another case of another master attempting to drive a data '0' (Figure 9-40).

**FIGURE 9-40: BUS COLLISION DURING A STOP CONDITION (CASE 1)**



**FIGURE 9-41: 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 9-42 depends on the following parameters

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

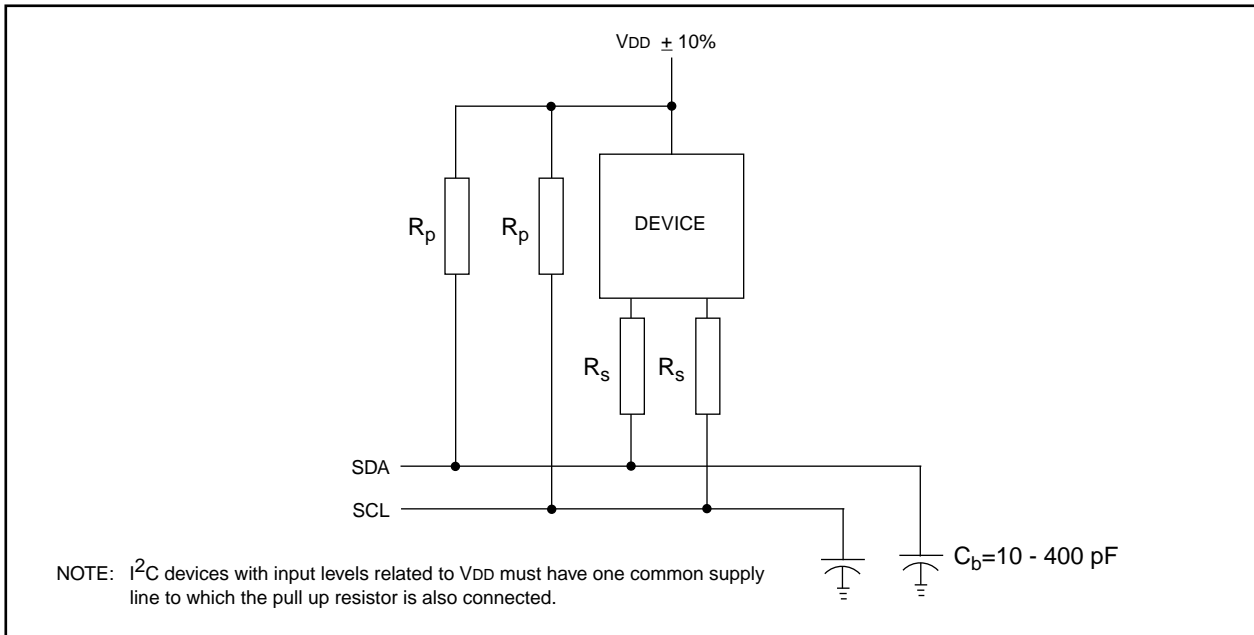
The supply voltage limits the minimum value of resistor  $R_p$  due to the specified minimum sink current of 3 mA at  $V_{OL\ max} = 0.4V$  for the specified output stages. For

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 9-42. 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 9-42).

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 9-42: 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.

**FIGURE 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
							bit0
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin-left: auto; margin-right: auto;">                     R = Readable bit                      W = Writable bit                      U = Unimplemented bit, read as '0'                      - n = Value at POR reset                 </div>							
bit 7: <b>CSRC:</b> Clock Source Select bit <u>Asynchronous mode</u> Don't care <u>Synchronous mode</u> 1 = Master mode (Clock generated internally from BRG) 0 = Slave mode (Clock from external source)							
bit 6: <b>TX9:</b> 9-bit Transmit Enable bit 1 = Selects 9-bit transmission 0 = Selects 8-bit transmission							
bit 5: <b>TXEN:</b> Transmit Enable bit 1 = Transmit enabled 0 = Transmit disabled Note: SREN/CREN overrides TXEN in SYNC mode.							
bit 4: <b>SYNC:</b> USART Mode Select bit 1 = Synchronous mode 0 = Asynchronous mode							
bit 3: <b>Unimplemented:</b> Read as '0'							
bit 2: <b>BRGH:</b> High Baud Rate Select bit <u>Asynchronous mode</u> 1 = High speed 0 = Low speed <u>Synchronous mode</u> Unused in this mode							
bit 1: <b>TRMT:</b> Transmit Shift Register Status bit 1 = TSR empty 0 = TSR full							
bit 0: <b>TX9D:</b> 9th bit of transmit data. Can be parity bit.							

# PIC16F87X

**FIGURE 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
							bit0
bit7							

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.

[Example 10-1](#) shows the calculation of the baud rate error for the following conditions:

FOSC = 16 MHz  
 Desired Baud Rate = 9600  
 BRGH = 0  
 SYNC = 0

### EXAMPLE 10-1: CALCULATING BAUD RATE ERROR

$$\begin{aligned} \text{Desired Baud rate} &= F_{osc} / (64 (X + 1)) \\ 9600 &= 16000000 / (64 (X + 1)) \\ X &= \lfloor 25.042 \rfloor = 25 \\ \text{Calculated Baud Rate} &= 16000000 / (64 (25 + 1)) \\ &= 9615 \\ \text{Error} &= \frac{(\text{Calculated Baud Rate} - \text{Desired Baud Rate})}{\text{Desired Baud Rate}} \\ &= (9615 - 9600) / 9600 \\ &= 0.16\% \end{aligned}$$

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 SYNCHRONOUS MODE**

BAUD RATE (K)	FOSC = 20 MHz			16 MHz			10 MHz			7.15909 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	NA	-	-	NA	-	-	NA	-	-	NA	-	-
2.4	NA	-	-	NA	-	-	NA	-	-	NA	-	-
9.6	NA	-	-	NA	-	-	9.766	+1.73	255	9.622	+0.23	185
19.2	19.53	+1.73	255	19.23	+0.16	207	19.23	+0.16	129	19.24	+0.23	92
76.8	76.92	+0.16	64	76.92	+0.16	51	75.76	-1.36	32	77.82	+1.32	22
96	96.15	+0.16	51	95.24	-0.79	41	96.15	+0.16	25	94.20	-1.88	18
300	294.1	-1.96	16	307.69	+2.56	12	312.5	+4.17	7	298.3	-0.57	5
500	500	0	9	500	0	7	500	0	4	NA	-	-
HIGH	5000	-	0	4000	-	0	2500	-	0	1789.8	-	0
LOW	19.53	-	255	15.625	-	255	9.766	-	255	6.991	-	255

BAUD RATE (K)	FOSC = 5.0688 MHz			4 MHz			3.579545 MHz			1 MHz			32.768 kHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-	0.303	+1.14	26
1.2	NA	-	-	NA	-	-	NA	-	-	1.202	+0.16	207	1.170	-2.48	6
2.4	NA	-	-	NA	-	-	NA	-	-	2.404	+0.16	103	NA	-	-
9.6	9.6	0	131	9.615	+0.16	103	9.622	+0.23	92	9.615	+0.16	25	NA	-	-
19.2	19.2	0	65	19.231	+0.16	51	19.04	-0.83	46	19.24	+0.16	12	NA	-	-
76.8	79.2	+3.13	15	76.923	+0.16	12	74.57	-2.90	11	83.34	+8.51	2	NA	-	-
96	97.48	+1.54	12	1000	+4.17	9	99.43	+3.57	8	NA	-	-	NA	-	-
300	316.8	+5.60	3	NA	-	-	298.3	-0.57	2	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	1267	-	0	100	-	0	894.9	-	0	250	-	0	8.192	-	0
LOW	4.950	-	255	3.906	-	255	3.496	-	255	0.9766	-	255	0.032	-	255

**TABLE 10-4 BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)**

BAUD RATE (K)	FOSC = 20 MHz			16 MHz			10 MHz			7.15909 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	1.221	+1.73	255	1.202	+0.16	207	1.202	+0.16	129	1.203	+0.23	92
2.4	2.404	+0.16	129	2.404	+0.16	103	2.404	+0.16	64	2.380	-0.83	46
9.6	9.469	-1.36	32	9.615	+0.16	25	9.766	+1.73	15	9.322	-2.90	11
19.2	19.53	+1.73	15	19.23	+0.16	12	19.53	+1.73	7	18.64	-2.90	5
76.8	78.13	+1.73	3	83.33	+8.51	2	78.13	+1.73	1	NA	-	-
96	104.2	+8.51	2	NA	-	-	NA	-	-	NA	-	-
300	312.5	+4.17	0	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	312.5	-	0	250	-	0	156.3	-	0	111.9	-	0
LOW	1.221	-	255	0.977	-	255	0.6104	-	255	0.437	-	255

BAUD RATE (K)	FOSC = 5.0688 MHz			4 MHz			3.579545 MHz			1 MHz			32.768 kHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.31	+3.13	255	0.3005	-0.17	207	0.301	+0.23	185	0.300	+0.16	51	0.256	-14.67	1
1.2	1.2	0	65	1.202	+1.67	51	1.190	-0.83	46	1.202	+0.16	12	NA	-	-
2.4	2.4	0	32	2.404	+1.67	25	2.432	+1.32	22	2.232	-6.99	6	NA	-	-
9.6	9.9	+3.13	7	NA	-	-	9.322	-2.90	5	NA	-	-	NA	-	-
19.2	19.8	+3.13	3	NA	-	-	18.64	-2.90	2	NA	-	-	NA	-	-
76.8	79.2	+3.13	0	NA	-	-	NA	-	-	NA	-	-	NA	-	-
96	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
300	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	79.2	-	0	62.500	-	0	55.93	-	0	15.63	-	0	0.512	-	0
LOW	0.3094	-	255	3.906	-	255	0.2185	-	255	0.0610	-	255	0.0020	-	255

**TABLE 10-5 BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)**

BAUD RATE	FOSC = 20 MHz			16 MHz			10 MHz			7.16 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
9.6	9.615	+0.16	129	9.615	+0.16	103	9.615	+0.16	64	9.520	-0.83	46
19.2	19.230	+0.16	64	19.230	+0.16	51	18.939	-1.36	32	19.454	+1.32	22
38.4	37.878	-1.36	32	38.461	+0.16	25	39.062	+1.7	15	37.286	-2.90	11
57.6	56.818	-1.36	21	58.823	+2.12	16	56.818	-1.36	10	55.930	-2.90	7
115.2	113.636	-1.36	10	111.111	-3.55	8	125	+8.51	4	111.860	-2.90	3
250	250	0	4	250	0	3	NA	-	-	NA	-	-
625	625	0	1	NA	-	-	625	0	0	NA	-	-
1250	1250	0	0	NA	-	-	NA	-	-	NA	-	-

BAUD RATE (K)	FOSC = 5.068 MHz			4 MHz			3.579 MHz			1 MHz			32.768 kHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
9.6	9.6	0	32	NA	-	-	9.727	+1.32	22	8.928	-6.99	6	NA	-	-
19.2	18.645	-2.94	16	1.202	+0.17	207	18.643	-2.90	11	20.833	+8.51	2	NA	-	-
38.4	39.6	+3.12	7	2.403	+0.13	103	37.286	-2.90	5	31.25	-18.61	1	NA	-	-
57.6	52.8	-8.33	5	9.615	+0.16	25	55.930	-2.90	3	62.5	+8.51	0	NA	-	-
115.2	105.6	-8.33	2	19.231	+0.16	12	111.860	-2.90	1	NA	-	-	NA	-	-
250	NA	-	-	NA	-	-	223.721	-10.51	0	NA	-	-	NA	-	-
625	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1250	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-

## 10.2 USART Asynchronous Mode

In this mode, the USART uses standard nonreturn-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 10-3. 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 indicated 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.

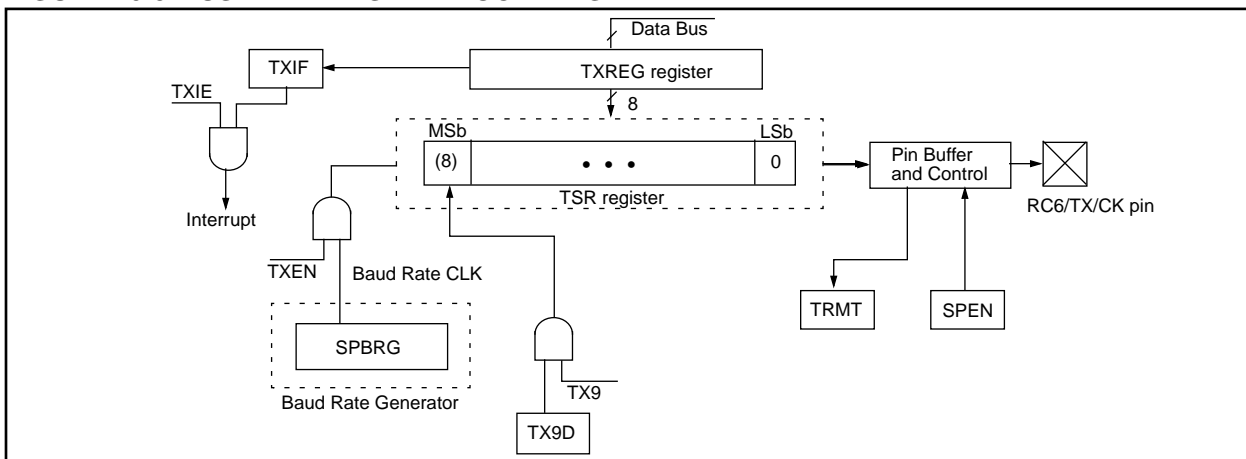
**Note 1:** The TSR register is not mapped in data memory so it is not available to the user.

**Note 2:** Flag bit TXIF is set when enable bit TXEN is set.

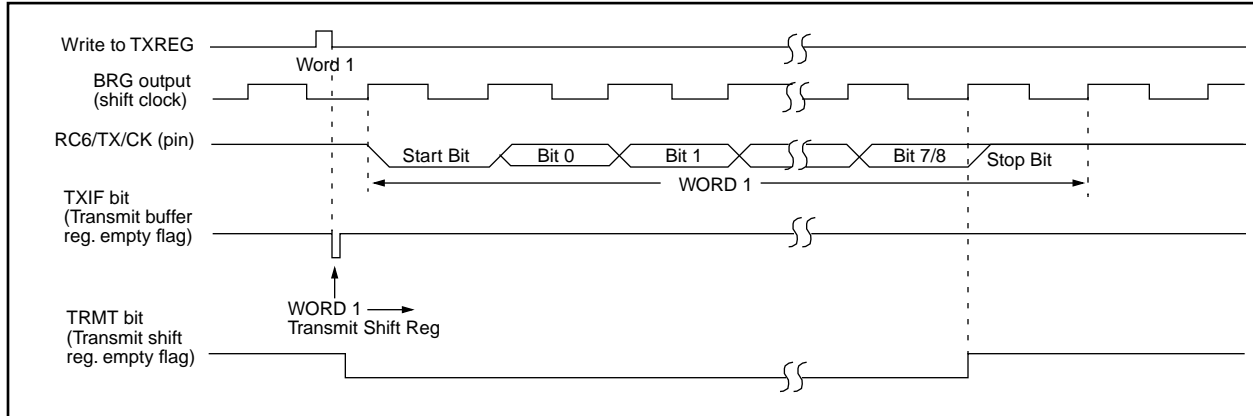
Steps to follow when setting up an Asynchronous Transmission:

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).

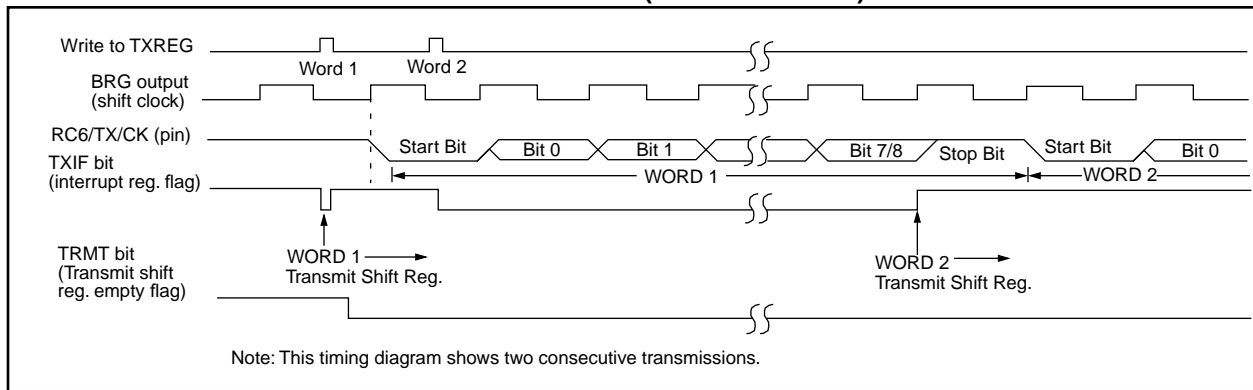
**FIGURE 10-3: USART TRANSMIT BLOCK DIAGRAM**



**FIGURE 10-4: ASYNCHRONOUS TRANSMISSION**



**FIGURE 10-5: ASYNCHRONOUS TRANSMISSION (BACK TO BACK)**



**TABLE 10-6 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	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 locations read as '0'. Shaded cells are not used for Asynchronous Transmission.

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

## 10.2.2 USART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in [Figure 10-6](#). 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.

The USART module has a special provision for multi-processor communication. When the RX9 bit is set in the RCSTA register, 9-bits are received and the ninth bit is placed in the RX9D status bit of the RSTA register. The port can be programmed such that when the stop bit is received, the serial port interrupt will only be activated if the RX9D bit = 1. This feature is enabled by setting the ADDEN bit RCSTA<3> in the RCSTA register. This feature can be used in a multi-processor system as follows:

A master processor intends to transmit a block of data to one of many slaves. It must first send out an address byte that identifies the target slave. An address byte is identified by the RX9D bit being a '1' (instead of a '0' for a data byte). If the ADDEN bit is set in the slave's RCSTA register, all data bytes will be ignored. However, if the ninth received bit is equal to a '1', indicating that the received byte is an address, the slave will be interrupted and the contents of the RSR register will be transferred into the receive buffer. This allows the slave to be interrupted only by addresses, so that the slave can examine the received byte to see if it is addressed. The addressed slave will then clear its ADDEN bit and prepare to receive data bytes from the master.

When ADDEN is set, all data bytes are ignored. Following the STOP bit, the data will not be loaded into the receive buffer, and no interrupt will occur. If another byte is shifted into the RSR register, the previous data byte will be lost.

The ADDEN bit will only take effect when the receiver is configured in 9-bit asynchronous mode.

The receiver block diagram is shown in [Figure 10-6](#).

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

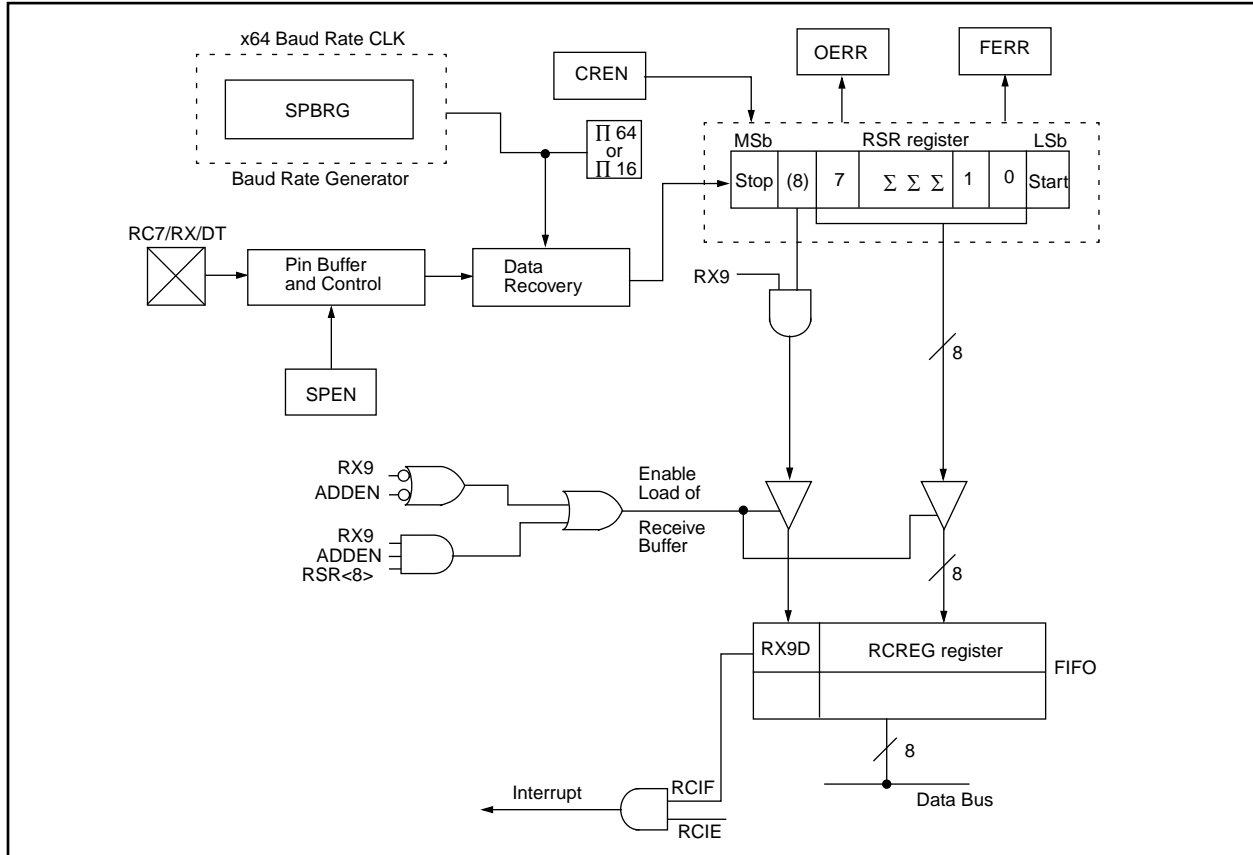
## 10.2.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

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

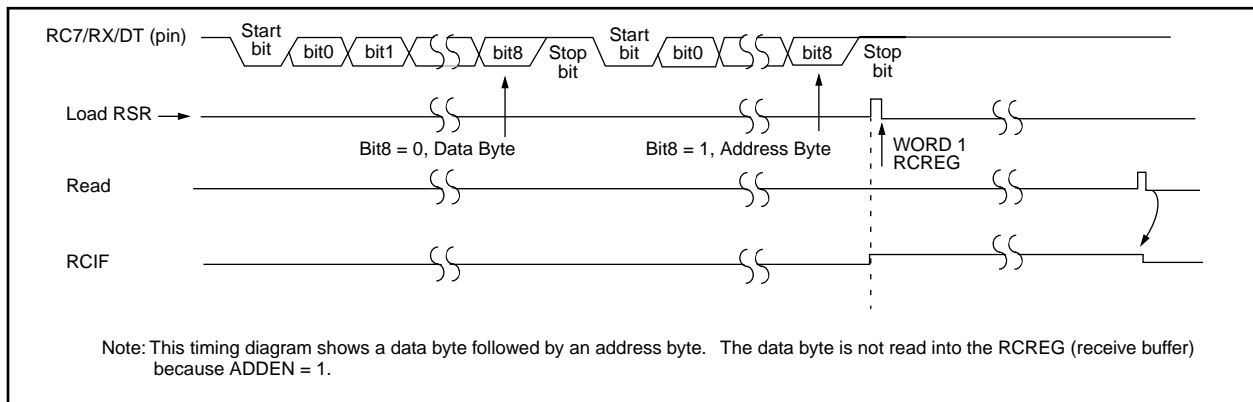
- Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH.
- Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
- If interrupts are desired, then set enable bit RCIE.
- Set bit RX9 to enable 9-bit reception.
- Set ADDEN 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.



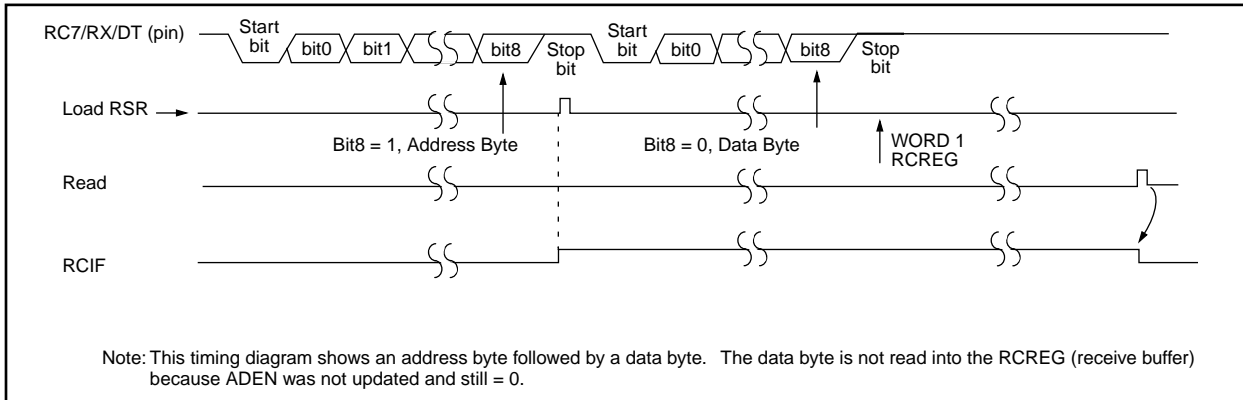
**FIGURE 10-6: USART RECEIVE BLOCK DIAGRAM**



**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 manner 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 10-3](#). 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.

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, then set enable bit TXIE.
4. If 9-bit transmission is desired, then 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.

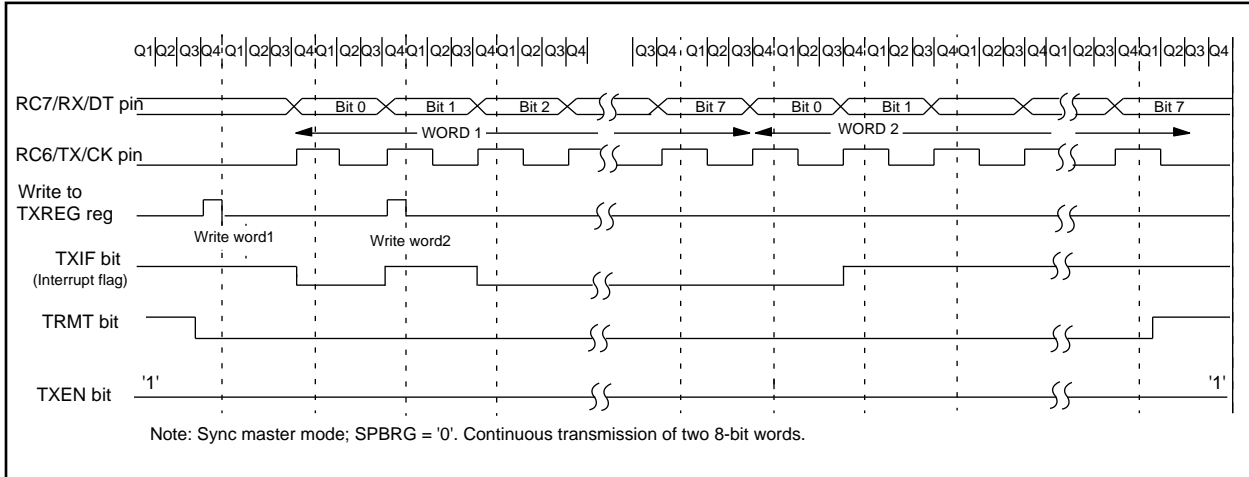
**TABLE 10-8 REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

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	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

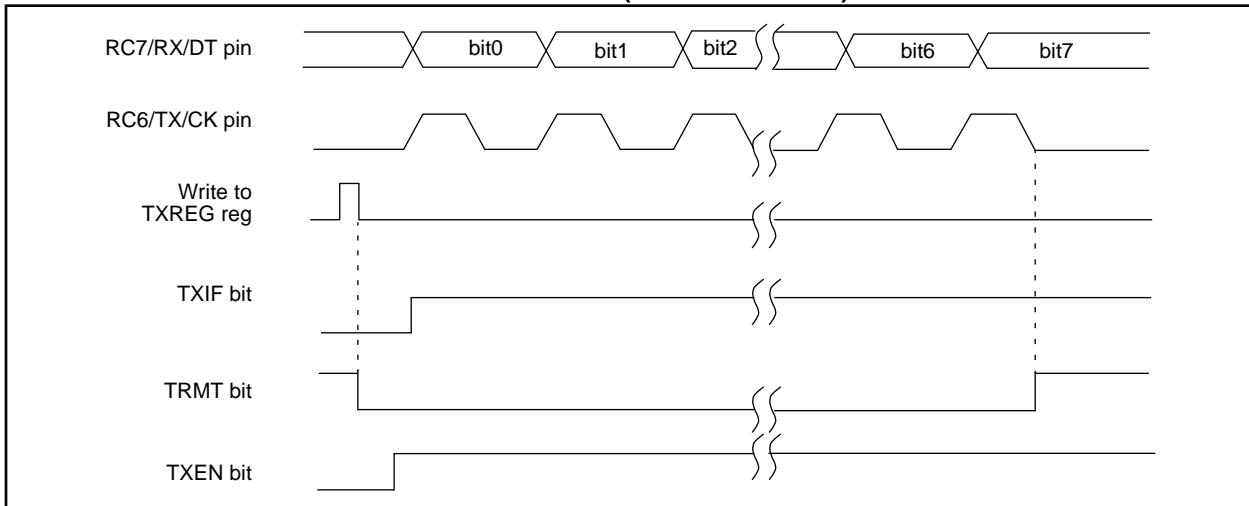
Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for Synchronous Master Transmission.

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 then CREN takes precedence.

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.

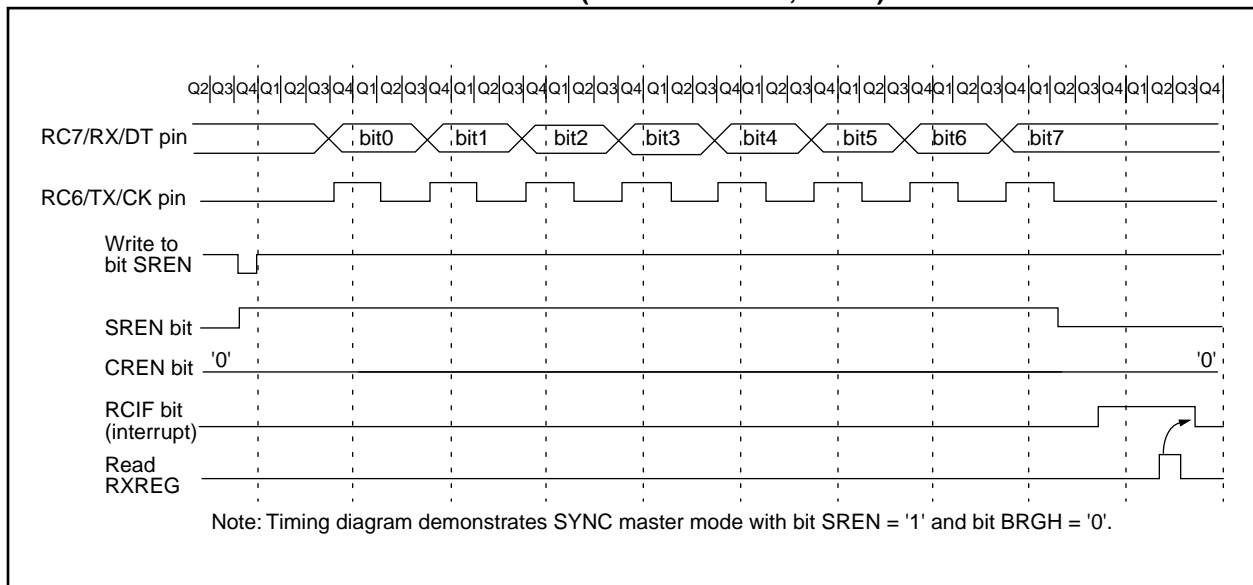
**TABLE 10-9 REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION**

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	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 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. Also, 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, then set enable bit RCIE.
3. If 9-bit reception is desired, then 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. This A/D conversion, of the analog input signal, results in a corresponding 10-bit digital number.

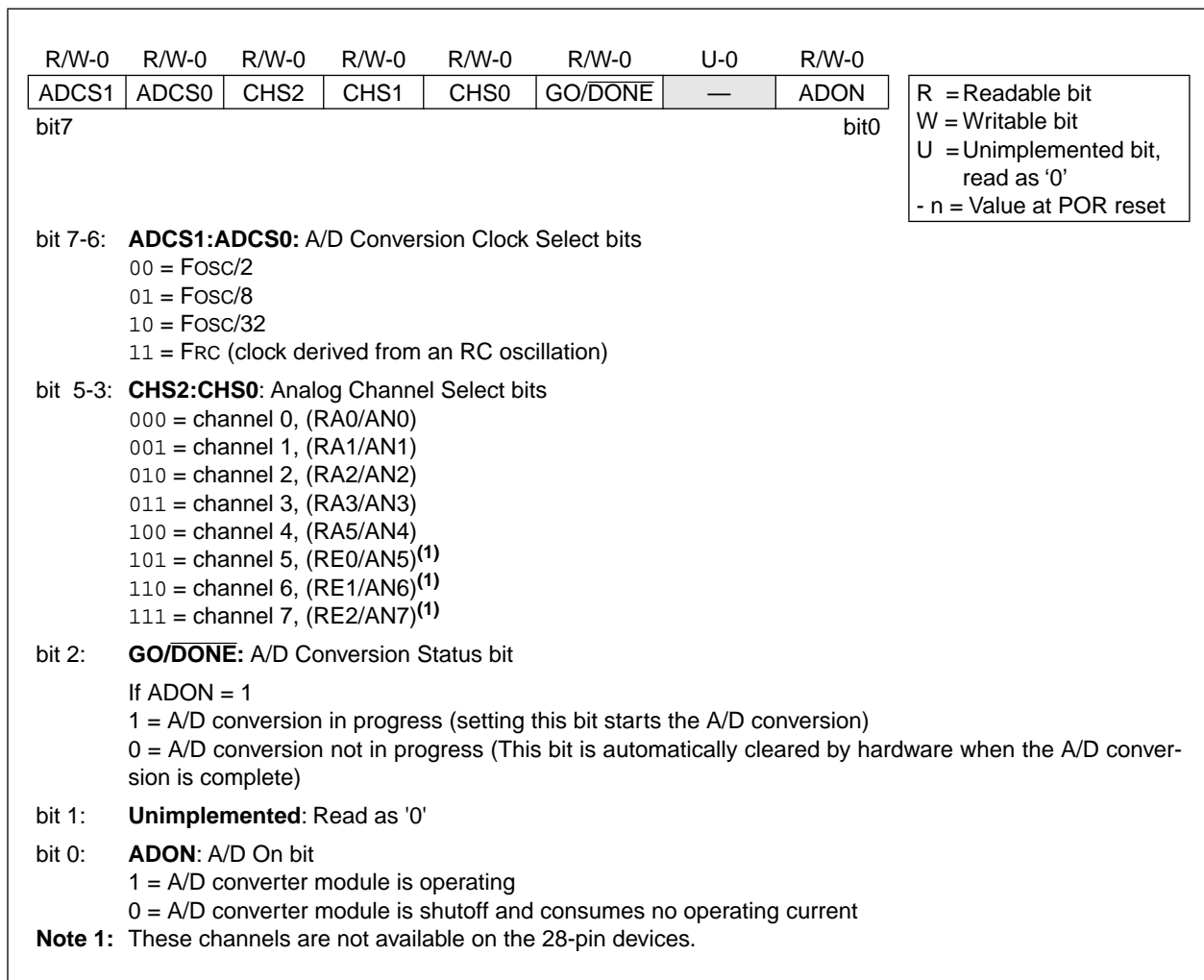
The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode. To operate in sleep, the A/D clock must be derived from the A/D's internal RC oscillator.

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 [Figure 11-1](#), controls the operation of the A/D module. The ADCON1 register, shown in [Figure 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.

**FIGURE 11-1: ADCON0 REGISTER (ADDRESS: 1Fh)**



# PIC16F87X

**FIGURE 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
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.

The ADRESH:ADRESL registers contains 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 GO/DONE bit (ADCON0<2>) is cleared, and A/D interrupt flag bit ADIF is set. The block diagram of the A/D module is shown in [Figure 11-3](#).

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 GO/DONE bit (ADCON0)
5. Wait for A/D conversion to complete, by either:
  - Polling for the 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.

# PIC16F87X

FIGURE 11-3: A/D BLOCK DIAGRAM

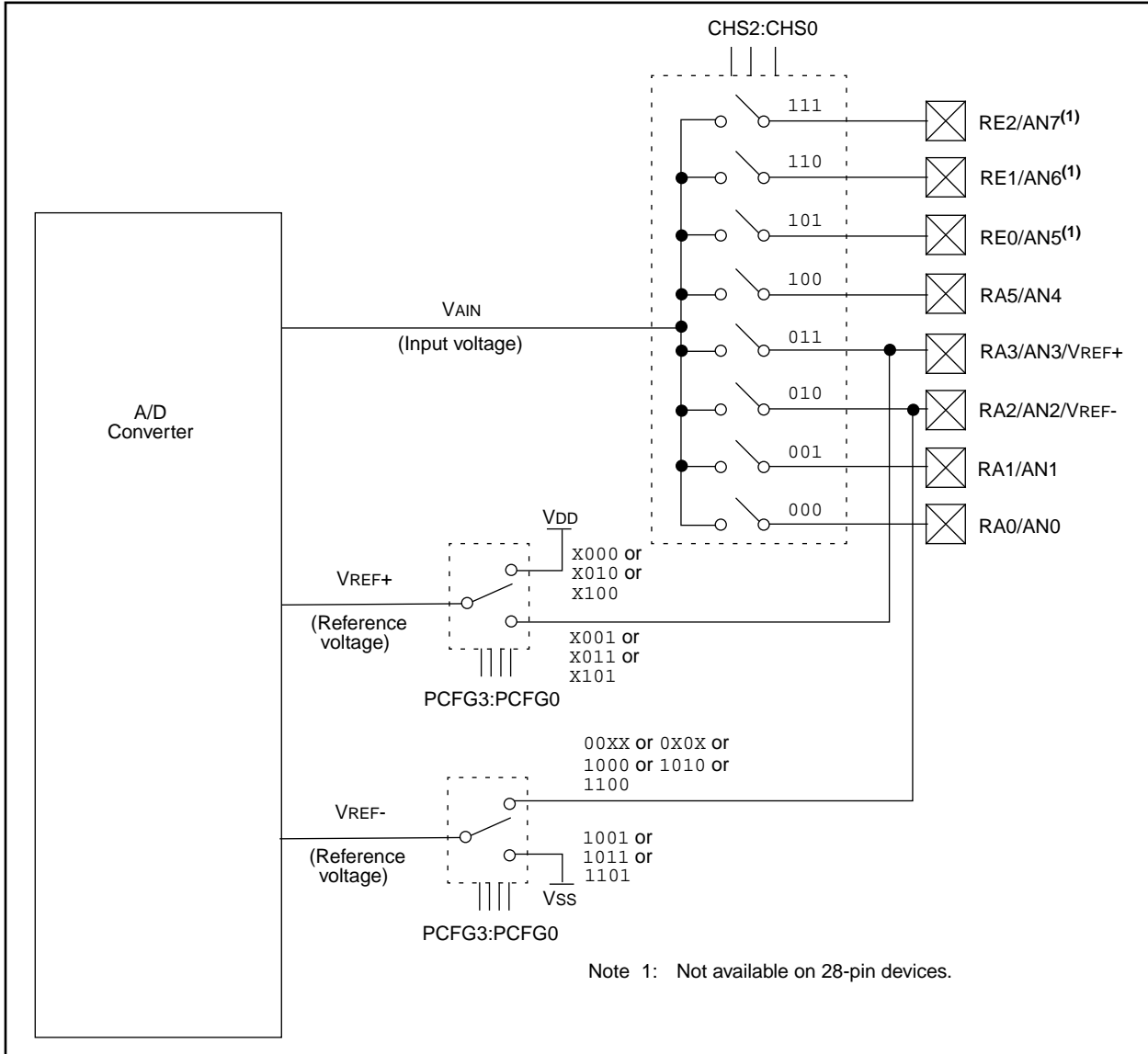
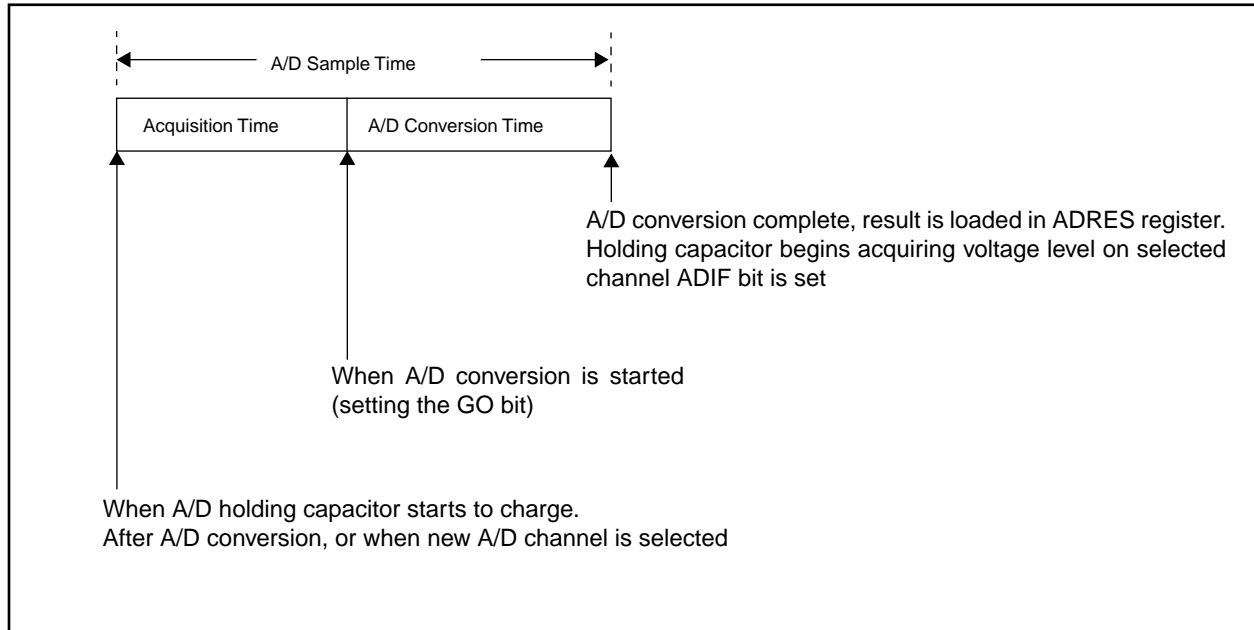


Figure 11-4 shows the conversion sequence, and the terms that are used. Acquisition time is the time that the A/D module's holding capacitor is connected to the external voltage level. Then there is the conversion time of 12 TAD, which is started when the GO bit is set. The

sum of these two times is the sampling time. There is a minimum acquisition time to ensure that the holding capacitor is charged to a level that will give the desired accuracy for the A/D conversion.

**FIGURE 11-4: A/D CONVERSION SEQUENCE**



## 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 11-5. The source impedance (Rs) and the internal sampling switch (Rss) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (Rss) impedance varies over the device voltage (VDD), Figure 11-5. The maximum recommended impedance for analog sources is 10 kW. 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.

Example 11-1 shows the calculation of the minimum required acquisition time TACQ. This calculation is based on the following application system assumptions.

CHOLD	=	120 pF
Rs	=	10 kΩ
Conversion Error	≤	1/2 LSb
VDD	=	5V → Rss = 7 kΩ
		(see graph in Figure 11-5)
Temperature	=	50°C (system max.)
VHOLD	=	0V @ time = 0

### EQUATION 11-1: ACQUISITION TIME

$$\begin{aligned} T_{ACQ} &= \text{Amplifier Settling Time} + \text{Holding Capacitor Charging Time} + \text{Temperature Coefficient} \\ &= T_{AMP} + T_C + T_{COFF} \end{aligned}$$

# PIC16F87X

## EQUATION 11-2: A/D MINIMUM CHARGING TIME

$$V_{HOLD} = (V_{REF} - (V_{REF}/2048)) \cdot (1 - e^{(-T_c/CHOLD(RIC + R_{SS} + R_s))})$$

or

$$T_c = -(120 \text{ pF})(1 \text{ k}\Omega + R_{SS} + R_s) \ln(1/2047)$$

## EXAMPLE 11-1: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME

$$T_{ACQ} = T_{AMP} + T_c + T_{COFF}$$

Temperature coefficient is only required for temperatures > 25°C.

$$T_{ACQ} = 2 \mu\text{s} + T_c + [(Temp - 25^\circ\text{C})(0.05 \mu\text{s}/^\circ\text{C})]$$

$$T_c = -CHOLD (RIC + R_{SS} + R_s) \ln(1/2047)$$

$$-120 \text{ pF} (1 \text{ k}\Omega + 7 \text{ k}\Omega + 10 \text{ k}\Omega) \ln(0.0004885)$$

$$-120 \text{ pF} (18 \text{ k}\Omega) \ln(0.0004885)$$

$$-2.16 \mu\text{s} (-7.6241)$$

$$16.47 \mu\text{s}$$

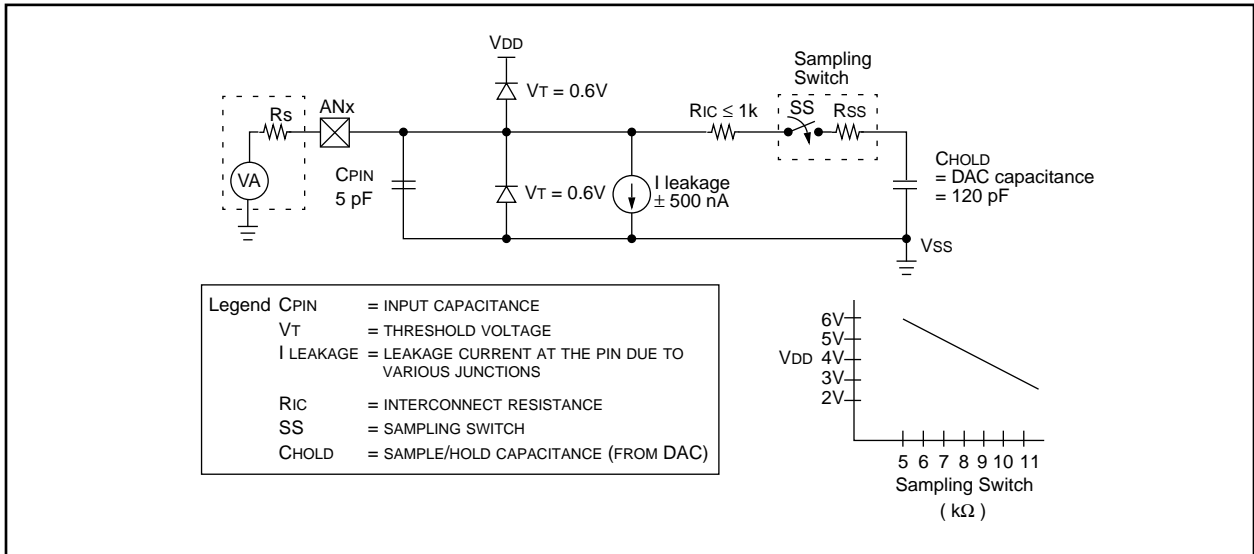
$$T_{ACQ} = 2 \mu\text{s} + 16.47 \mu\text{s} + [(50^\circ\text{C} - 25^\circ\text{C})(0.05 \mu\text{s}/^\circ\text{C})]$$

$$18.447 \mu\text{s} + 1.25 \mu\text{s}$$

$$19.72 \mu\text{s}$$

- Note 1:** The reference voltage (VREF) has no effect on the equation, since it cancels itself out.
- Note 2:** The charge holding capacitor (CHOLD) is not discharged after each conversion.
- Note 3:** The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.
- Note 4:** After a conversion has completed, a 2.0TAD 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 11-5: ANALOG INPUT MODEL**



## 11.2 Selecting the A/D Conversion Clock

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

- 2Tosc
- 8Tosc
- 32Tosc
- Internal RC oscillator

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

Table 11-1 and Table 11-2 show the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

**TABLE 11-1 TAD vs. DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (C))**

AD Clock Source (TAD)		Device Frequency			
Operation	ADCS1:ADCS0	20 MHz	5 MHz	1.25 MHz	333.33 kHz
2Tosc	00	100 ns <sup>(2)</sup>	400 ns <sup>(2)</sup>	1.6 $\mu$ s	6 $\mu$ s
8Tosc	01	400 ns <sup>(2)</sup>	1.6 $\mu$ s	6.4 $\mu$ s	24 $\mu$ s <sup>(3)</sup>
32Tosc	10	1.6 $\mu$ s	6.4 $\mu$ s	25.6 $\mu$ s <sup>(3)</sup>	96 $\mu$ s <sup>(3)</sup>
RC	11	2 - 6 $\mu$ s <sup>(1, 4)</sup>	2 - 6 $\mu$ s <sup>(1, 4)</sup>	2 - 6 $\mu$ s <sup>(1, 4)</sup>	2 - 6 $\mu$ s <sup>(1)</sup>

Legend: Shaded cells are outside of recommended ranges.

- Note 1:** The RC source has a typical TAD time of 4  $\mu$ s.  
**2:** These values violate the minimum required TAD time.  
**3:** For faster conversion times, the selection of another clock source is recommended.  
**4:** When the device frequencies is greater than 1 MHz, the RC A/D conversion clock source is only recommended for sleep operation.  
**5:** For extended voltage devices (LC), please refer to 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 (VOH or VOL) 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.

**Note 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 devices specification.

# PIC16F87X

## 11.4 A/D Conversions

Example 11-1 shows how to perform an A/D conversion. The analog pins are configured as analog inputs. The analog references (VREF) are the device VDD and VSS. The A/D interrupt is enabled, and the A/D conversion clock is FRC, with the result being left justified. The conversion is performed on the RA0/AN0 pin (channel 0).

**Note:** The GO/DONE bit should NOT be set in the same instruction that turns on the A/D.

Clearing the 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.

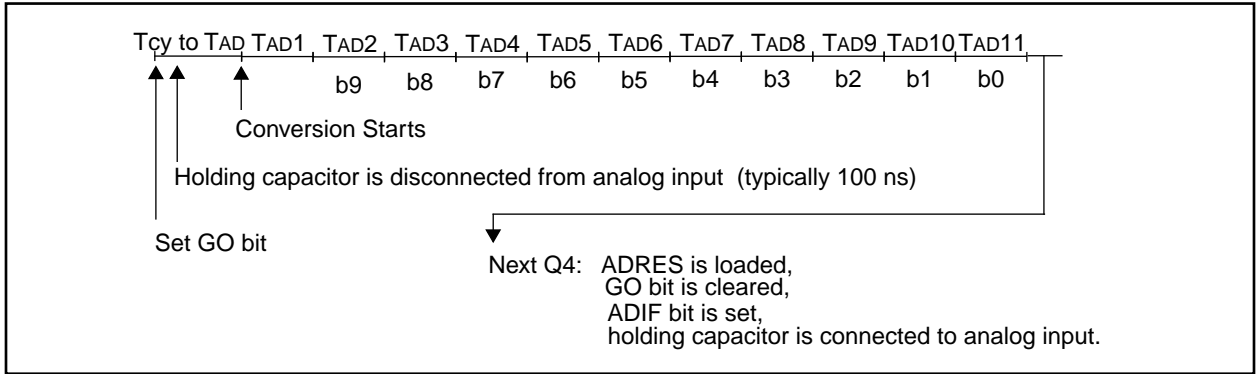
In Figure 11-6, after the GO bit is set, the first time segment has a minimum of TCY and a maximum of TAD.

### EXAMPLE 11-1: A/D CONVERSION

```

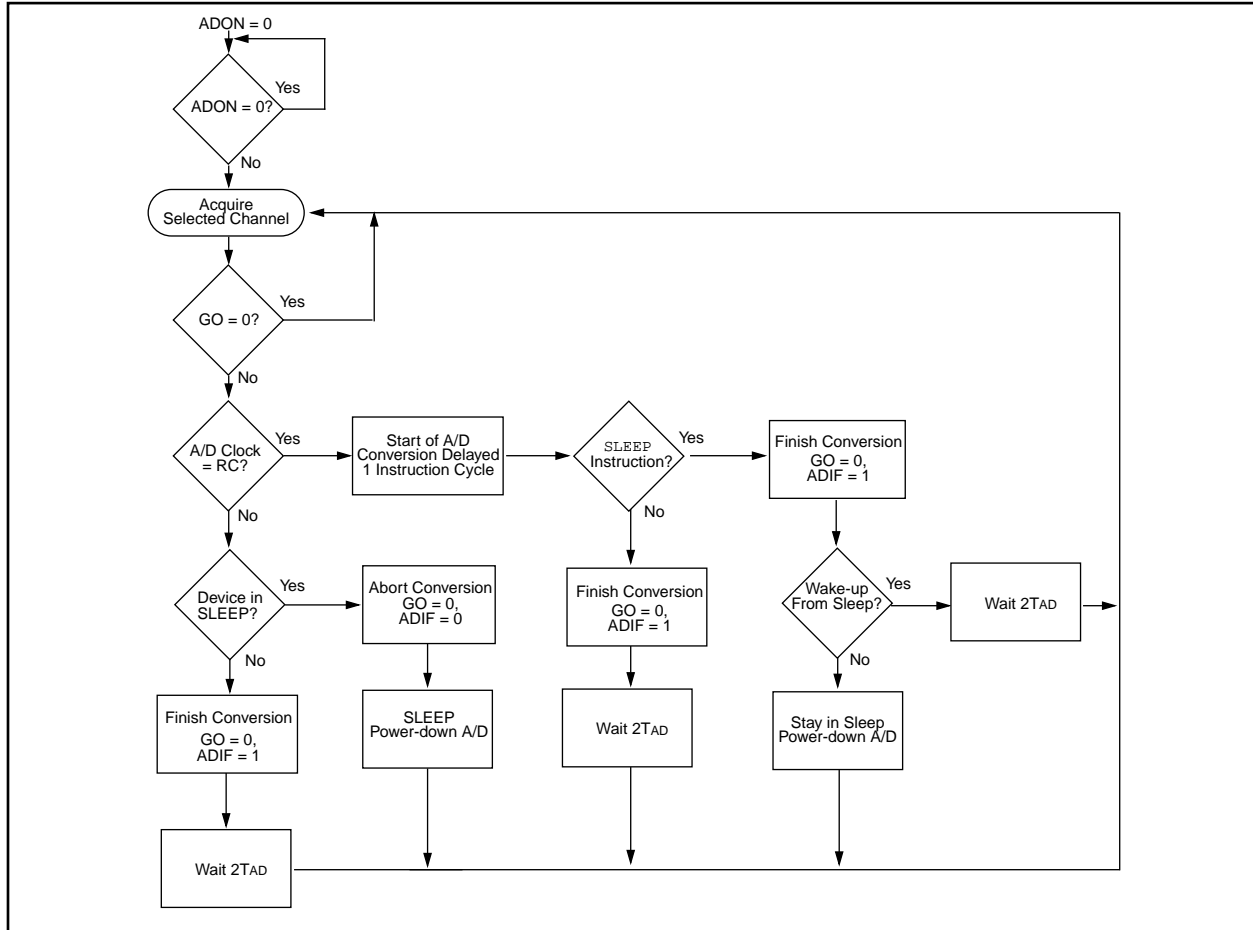
BSF    STATUS, RP0      ; Select Bank 1
BCF    STATUS, RP1      ;
CLRF   ADCON1           ; Configure A/D inputs
BSF    PIE1,  ADIE      ; Enable A/D interrupts
BCF    STATUS, RP0      ; Select Bank 0
MOVLW  11000001         ; RC Clock, A/D is on, Channel 0 is selected
MOVWF  ADCON0           ;
BCF    PIR1,  ADIF      ; Clear A/D interrupt flag bit
BSF    INTCON, PEIE     ; Enable peripheral interrupts
BSF    INTCON, GIE      ; Enable all interrupts
;
; Ensure that the required sampling time for the selected input channel has elapsed.
; Then the conversion may be started.
;
BSF    ADCON0, GO       ; Start A/D Conversion
:      ; The ADIF bit will be set and the GO/DONE bit
:      ; is cleared upon completion of the A/D Conversion.
    
```

**FIGURE 11-6: A/D CONVERSION TAD CYCLES**





**FIGURE 11-7: FLOWCHART OF A/D OPERATION**



### 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 11-8 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.

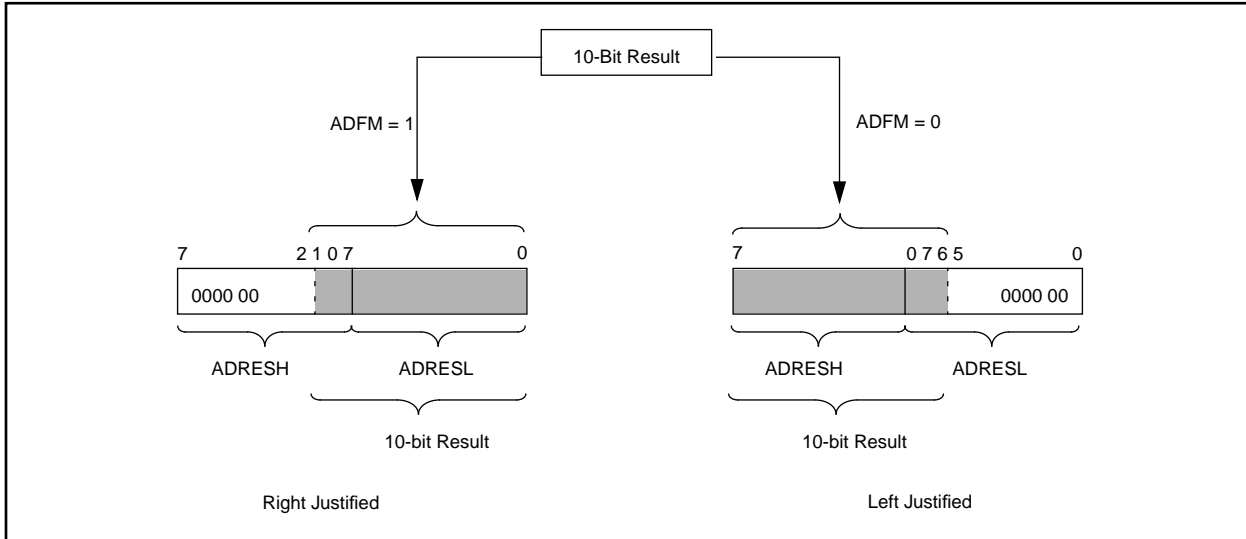
**Note:** For the A/D module to operate in SLEEP, the A/D clock source must be set to RC (ADCS1:ADCS0 = 11). To allow the conversion to occur during SLEEP, ensure the SLEEP instruction immediately follows the instruction that sets the GO/DONE bit.

## 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 11-8: A/D RESULT JUSTIFICATION**



## 11.7 A/D Accuracy/Error

In systems where the device frequency is low, use of the A/D RC clock is preferred. At moderate to high frequencies, TAD should be derived from the device oscillator.

The absolute accuracy specified for the A/D converter includes the sum of all contributions for quantization error, integral error, differential error, full scale error, offset error, and monotonicity. It is defined as the maximum deviation from an actual transition versus an ideal transition for any code. The absolute error of the A/D converter is specified at  $< \pm 1$  LSB for  $V_{DD} = V_{REF}$  (over the device's specified operating range). However, the accuracy of the A/D converter will degrade as  $V_{REF}$  diverges from  $V_{DD}$ .

For a given range of analog inputs, the output digital code will be the same. This is due to the quantization of the analog input to a digital code. Quantization error is typically  $\pm 1/2$  LSB and is inherent in the analog to digital conversion process. The only way to reduce quantization error is to increase the resolution of the A/D converter or oversample.

Offset error measures the first actual transition of a code versus the first ideal transition of a code. Offset error shifts the entire transfer function. Offset error can be calibrated out of a system or introduced into a system through the interaction of the total leakage current and source impedance at the analog input.

Gain error measures the maximum deviation of the last actual transition and the last ideal transition adjusted for offset error. This error appears as a change in slope of the transfer function. The difference in gain error to

full scale error is that full scale does not take offset error into account. Gain error can be calibrated out in software.

Linearity error refers to the uniformity of the code changes. Linearity errors cannot be calibrated out of the system. Integral non-linearity error measures the actual code transition versus the ideal code transition adjusted by the gain error for each code.

Differential non-linearity measures the maximum actual code width versus the ideal code width. This measure is unadjusted.

The maximum pin leakage current is specified in the Device Data Sheet electrical specification parameter #D060.

In systems where the device frequency is low, use of the A/D RC clock is preferred. At moderate to high frequencies, TAD should be derived from the device oscillator. TAD must not violate the minimum and should be minimized to reduce inaccuracies due to noise and sampling capacitor bleed off.

In systems where the device will enter SLEEP mode after the start of the A/D conversion, the RC clock source selection is required. In this mode, the digital noise from the modules in SLEEP are stopped. This method gives high accuracy.

## 11.8 Connection Considerations

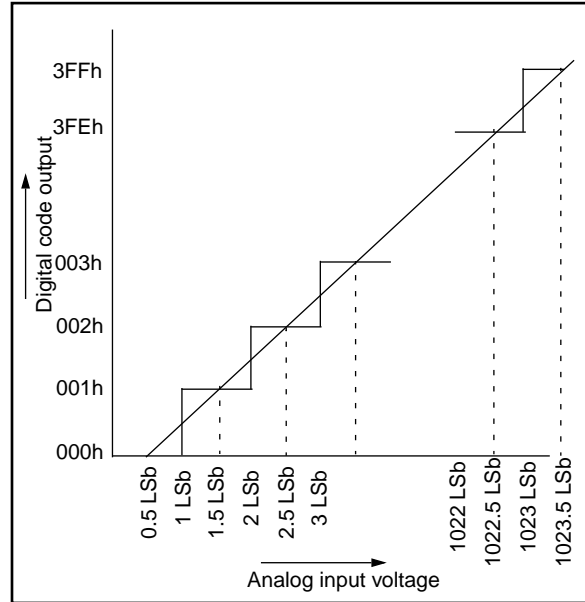
If the input voltage exceeds the rail values ( $V_{SS}$  or  $V_{DD}$ ) by greater than 0.3V, then the accuracy of the conversion is out of specification.

An external RC filter is sometimes added for anti-aliasing of the input signal. The R component should be selected to ensure that the total source impedance is kept under the 10 k $\Omega$  recommended specification. Any external components connected (via hi-impedance) to an analog input pin (capacitor, zener diode, etc.) should have very little leakage current at the pin.

## 11.9 Transfer Function

The transfer function of the A/D converter is as follows: the first transition occurs when the analog input voltage ( $V_{AIN}$ ) equals Analog  $V_{REF} / 1024$  (Figure 11-9).

FIGURE 11-9: A/D TRANSFER FUNCTION



## 11.10 References

A good reference for the understanding A/D converter is the "Analog-Digital Conversion Handbook" third edition, published by Prentice Hall (ISBN 0-13-03-2848-0).

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								—	—
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 PICmicros 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 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, 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.

**FIGURE 12-1: CONFIGURATION WORD**

CP1	CP0	BKBUG	-	WRT	CPD	LVP	BODEN	CP1	CP0	PWRTE	WDTE	FOSC1	FOSC0	Register: CONFIG
													Address: 2007h	
bit13													bit0	
bit 13-12:														
bit 5-4: <b>CP1:CP0:</b> 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: <b>DEBUG:</b> 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: <b>Unimplemented:</b> Read as '1'														
bit 9: <b>WRT:</b> 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: <b>CPD:</b> Data EE Memory Code Protection														
1 = Code protection off														
0 = Data EEPROM memory code protected														
bit 7: <b>LVP:</b> Low voltage 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: <b>BODEN:</b> Brown-out Reset Enable bit <sup>(1)</sup>														
1 = BOR enabled														
0 = BOR disabled														
bit 3: <b>PWRTE:</b> Power-up Timer Enable bit <sup>(1)</sup>														
1 = PWRT disabled														
0 = PWRT enabled														
bit 2: <b>WDTE:</b> Watchdog Timer Enable bit														
1 = WDT enabled														
0 = WDT disabled														
bit 1-0: <b>FOSC1:FOSC0:</b> 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

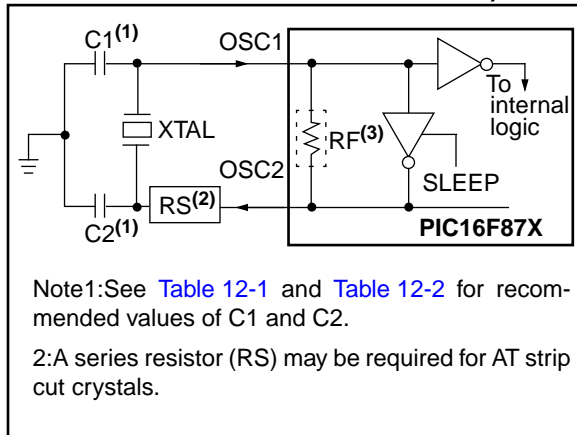
The PIC16F87X can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

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

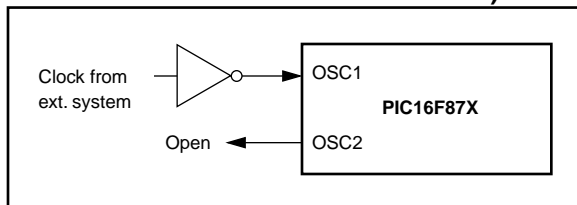
### 12.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

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 12-2). 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 12-3).

**FIGURE 12-2: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)**



**FIGURE 12-3: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)**



**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.

**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.

Crystals Used		
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

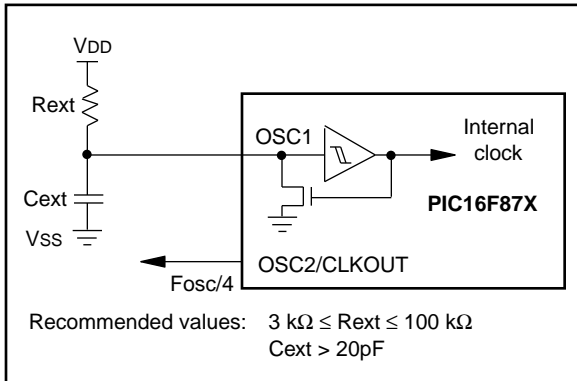
- Note 1: Recommended values of C1 and C2 are identical to the ranges tested (Table 12-1).  
2: Higher capacitance increases the stability of oscillator but also increases the start-up time.  
3: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.  
4: Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification.

### 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 12-4 shows how the R/C combination is connected to the PIC16F87X.

# PIC16F87X

FIGURE 12-4: 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.

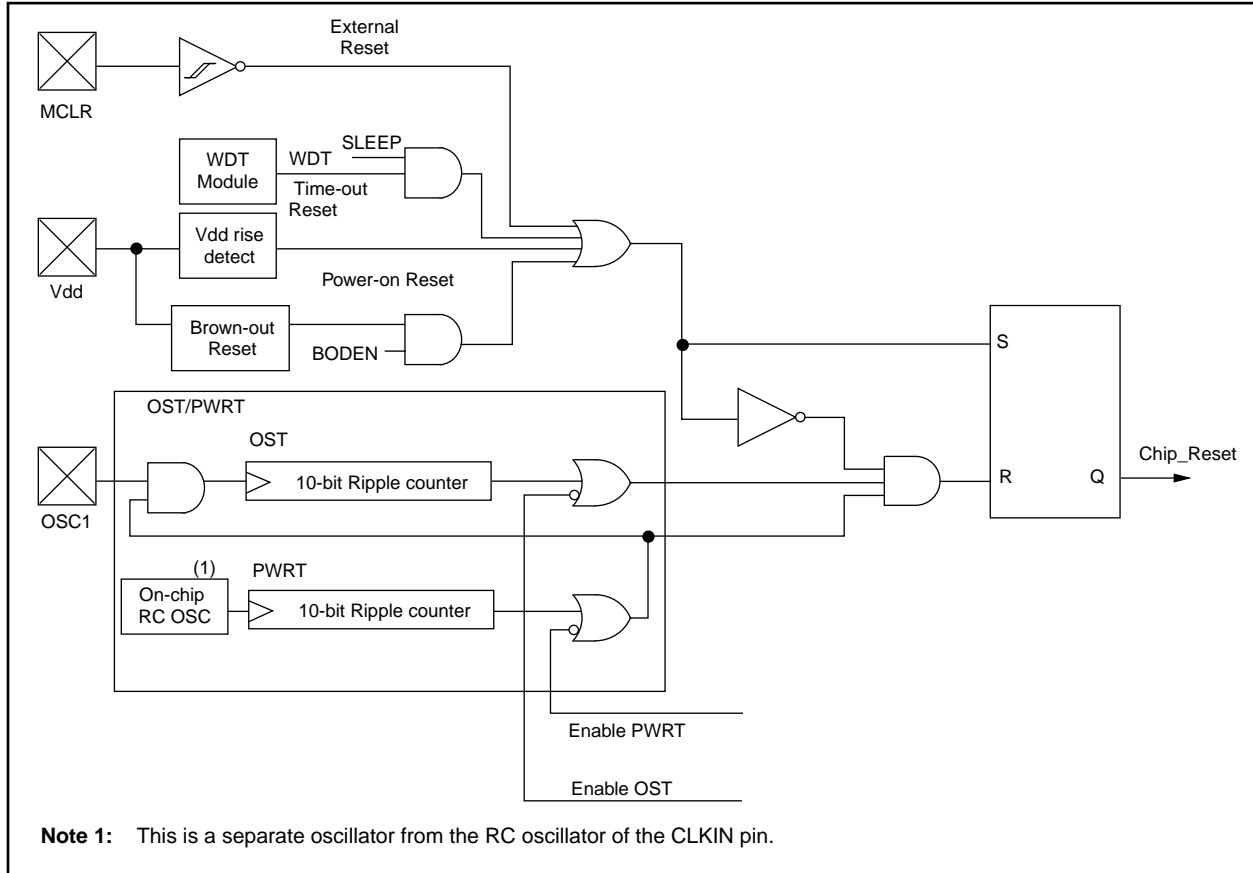
A simplified block diagram of the on-chip reset circuit is shown in Figure 12-5.

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 12-5: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**

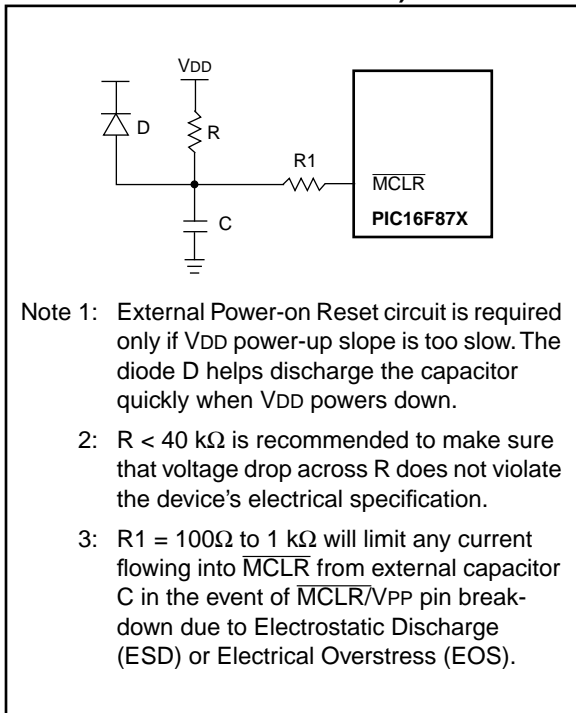


## 12.4 Power-On Reset (POR)

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, just 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. For a slow rise time, see Figure 12-6.

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.

**FIGURE 12-6: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)**



## 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.

## 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)

A configuration bit, BODEN, can disable (if clear/programmed) or enable (if set) the Brown-out Reset circuitry. If VDD falls below 4.0V (3.8V - 4.2V range) for greater than parameter #35, the brown-out situation will reset the chip. A reset may not occur if VDD falls below 4.0V for less than parameter #35. The chip will remain in Brown-out Reset until VDD rises above BVDD. The Power-up Timer will now be invoked and will keep the chip in RESET an additional 72 ms. If VDD drops below BVDD while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be initialized. Once VDD rises above BVDD, the Power-up Timer will execute a 72 ms time delay. The Power-up Timer should always be enabled when Brown-out Reset is enabled.

## 12.8 Time-out Sequence

On power-up the time-out sequence is as follows: First PWRT time-out is invoked after the POR time delay has expired. Then OST is activated. The total time-out will vary based on oscillator configuration and the status of the PWRT. For example, in RC mode with the PWRT disabled, there will be no time-out at all. [Figure 12-7](#), [Figure 12-8](#), [Figure 12-9](#) and [Figure 12-10](#) depict time-out sequences on power-up.

Since the time-outs occur from the POR pulse, if  $\overline{\text{MCLR}}$  is kept low long enough, the time-outs will expire. Then bringing  $\overline{\text{MCLR}}$  high will begin execution immediately ([Figure 12-9](#)). This is useful for testing purposes or to synchronize more than one PIC16F87X device operating in parallel.

[Table 12-5](#) shows the reset conditions for some special function registers, while [Table 12-6](#) shows the reset conditions for all the registers.

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

Oscillator Configuration	Power-up		Brown-out	Wake-up from SLEEP
	PWRTE = 0	PWRTE = 1		
XT, HS, LP	72 ms + 1024Tosc	1024Tosc	72 ms + 1024Tosc	1024Tosc
RC	72 ms	—	72 ms	—

**TABLE 12-4 STATUS BITS AND THEIR SIGNIFICANCE**

POR	BOR	$\overline{\text{TO}}$	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, 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).

## 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.

# 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
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

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 (Cont.'d)**

Register	Devices				Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset	Wake-up via WDT or Interrupt
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
PCON	873	874	876	877	---- -qqr	---- --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	xxxx xxxx	uuuu uuuu	uuuu 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	---- ----	---- ----	---- ----

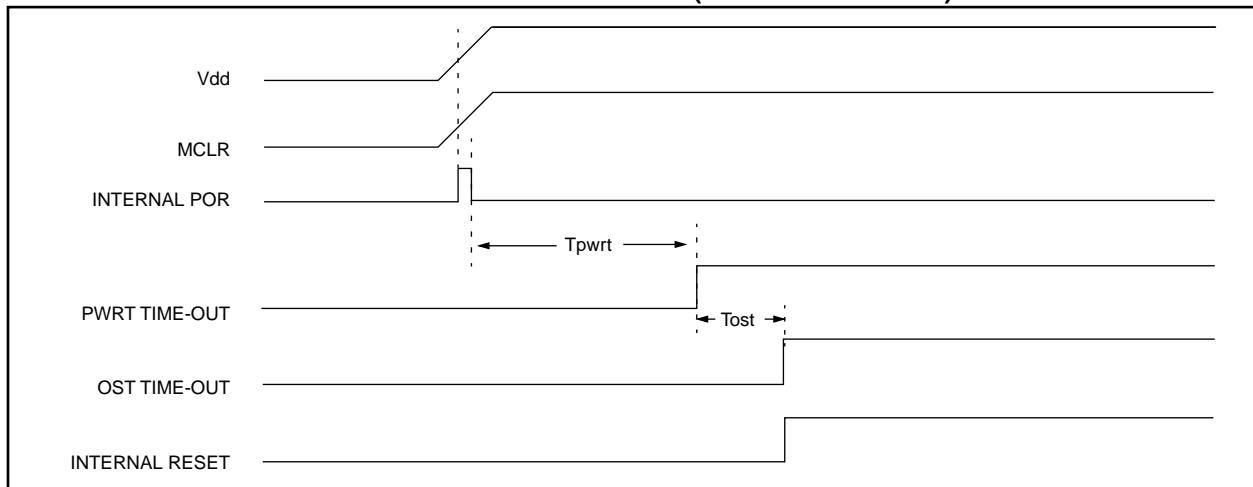
Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition, r = reserved maintain clear.

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).

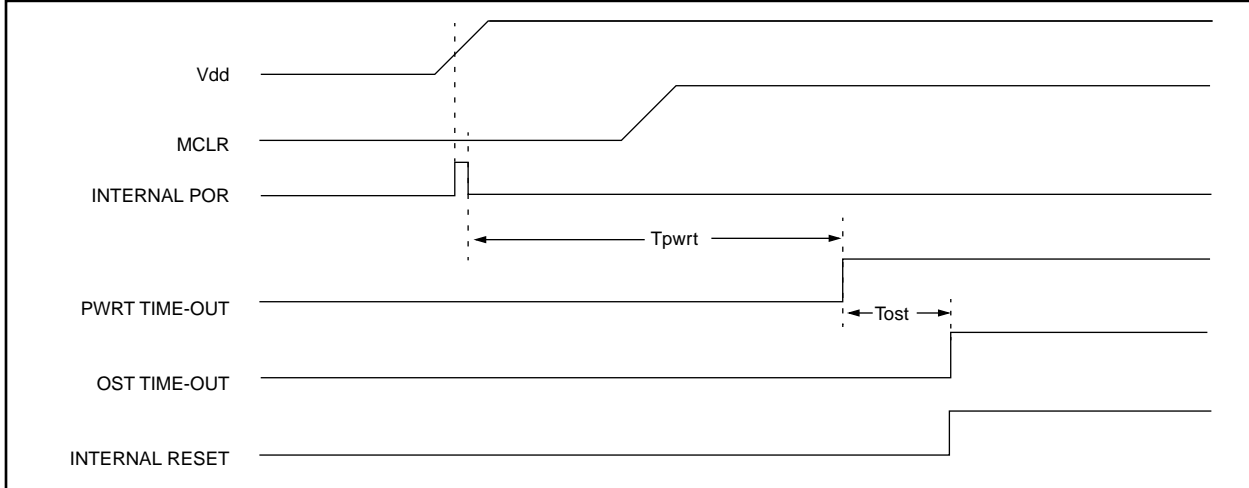
3: See Table 12-5 for reset value for specific condition.

**FIGURE 12-7: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD)**

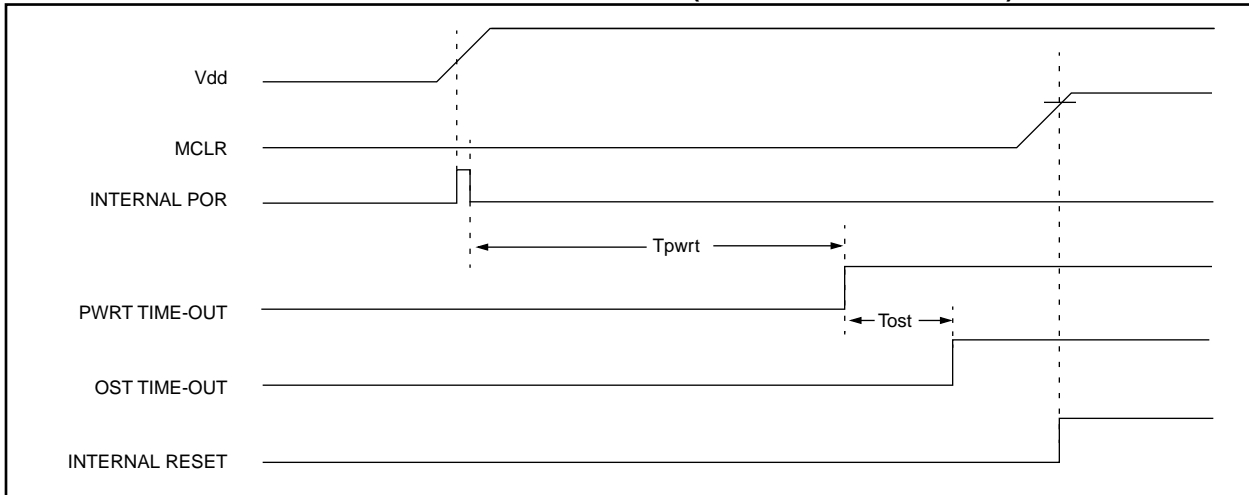


# PIC16F87X

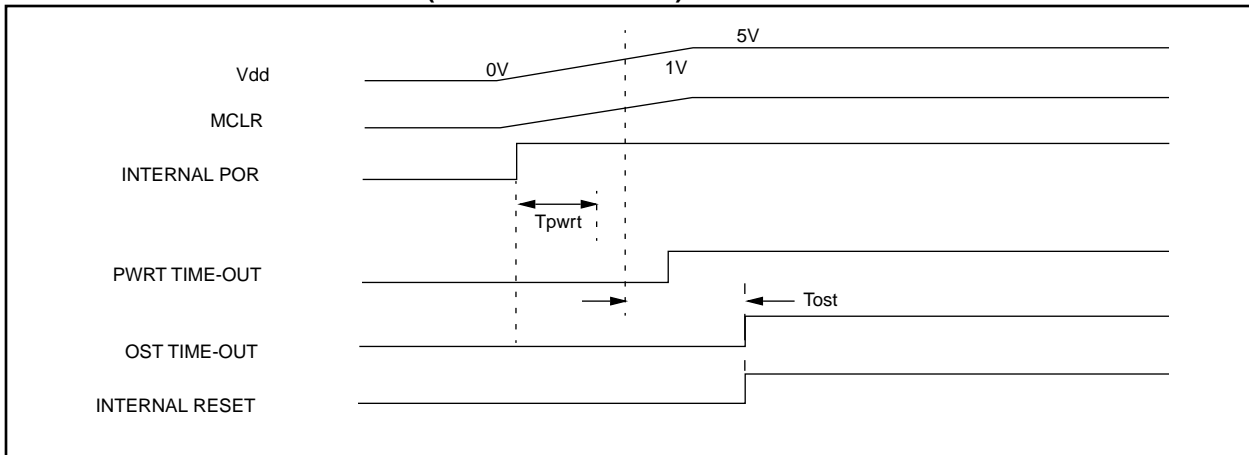
**FIGURE 12-8: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{DD}$ ): CASE 1**



**FIGURE 12-9: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{DD}$ ): CASE 2**



**FIGURE 12-10: SLOW RISE TIME ( $\overline{\text{MCLR}}$  TIED TO  $V_{DD}$ )**



## 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.

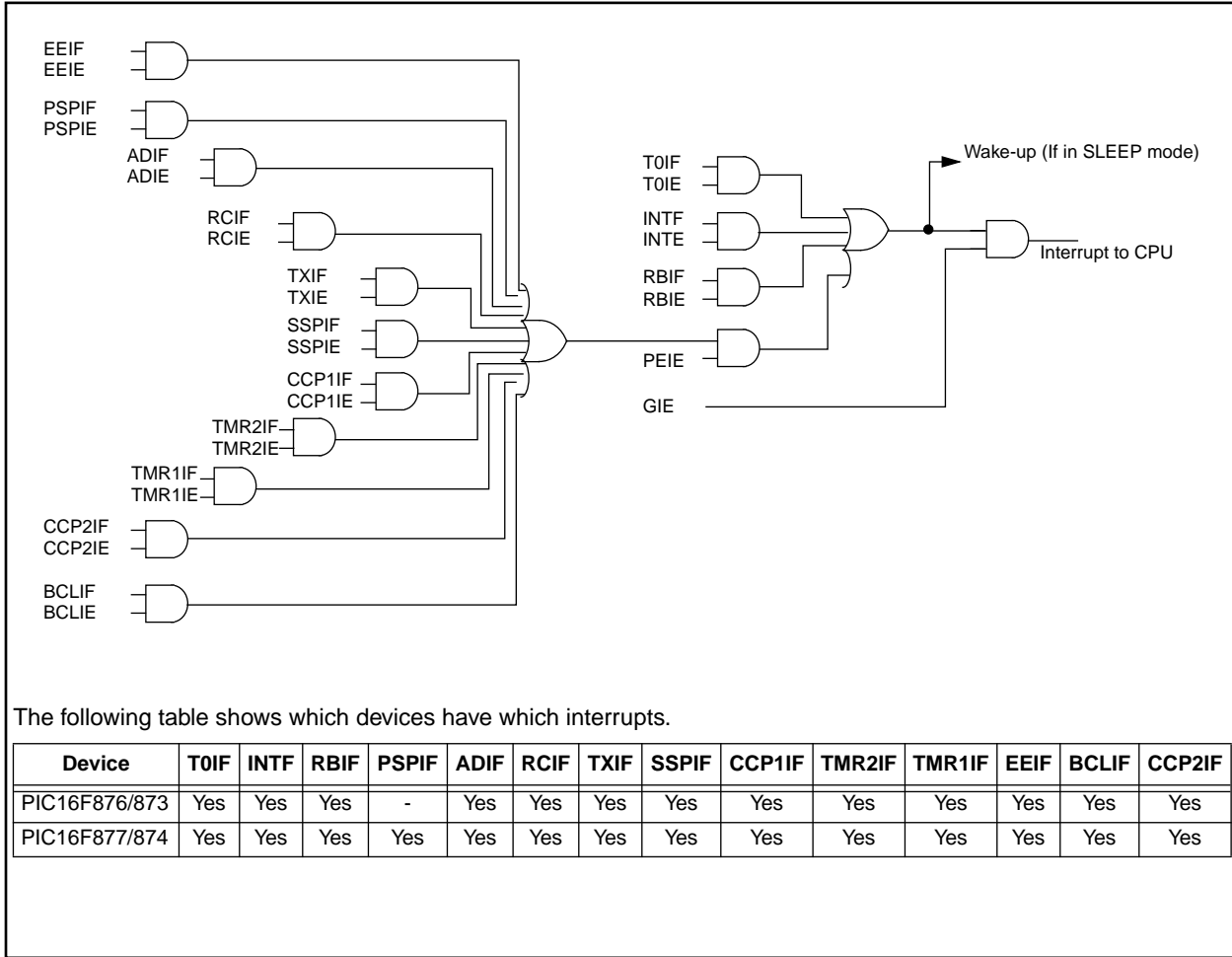
The peripheral interrupt flags are contained in the special function registers PIR1 and PIR2. The corresponding interrupt enable bits are contained in special function registers PIE1 and PIE2, and the peripheral interrupt enable bit is contained in special function register INTCON.

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

# PIC16F87X

**FIGURE 12-11: INTERRUPT LOGIC**





## 12.10.1 INT INTERRUPT

External interrupt on 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.

[Example 12-1](#) stores and restores the W and STATUS registers. The register, W\_TEMP, must be defined in each bank 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 example:

- a) Stores the W register.
- b) Stores the STATUS register in bank 0.
- c) Stores the PCLATH register.
- d) Executes the interrupt service routine code (User-generated).
- e) Restores the STATUS register (and bank select bit).
- f) Restores the W and PCLATH registers.

### 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
BCF     STATUS, IRP      ;Return to Bank 0
MOVF     FSR, W           ;Copy FSR to W
MOVWF    FSR_TEMP        ;Copy FSR from W to FSR_TEMP
:
:(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

```

# PIC16F87X

## 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.

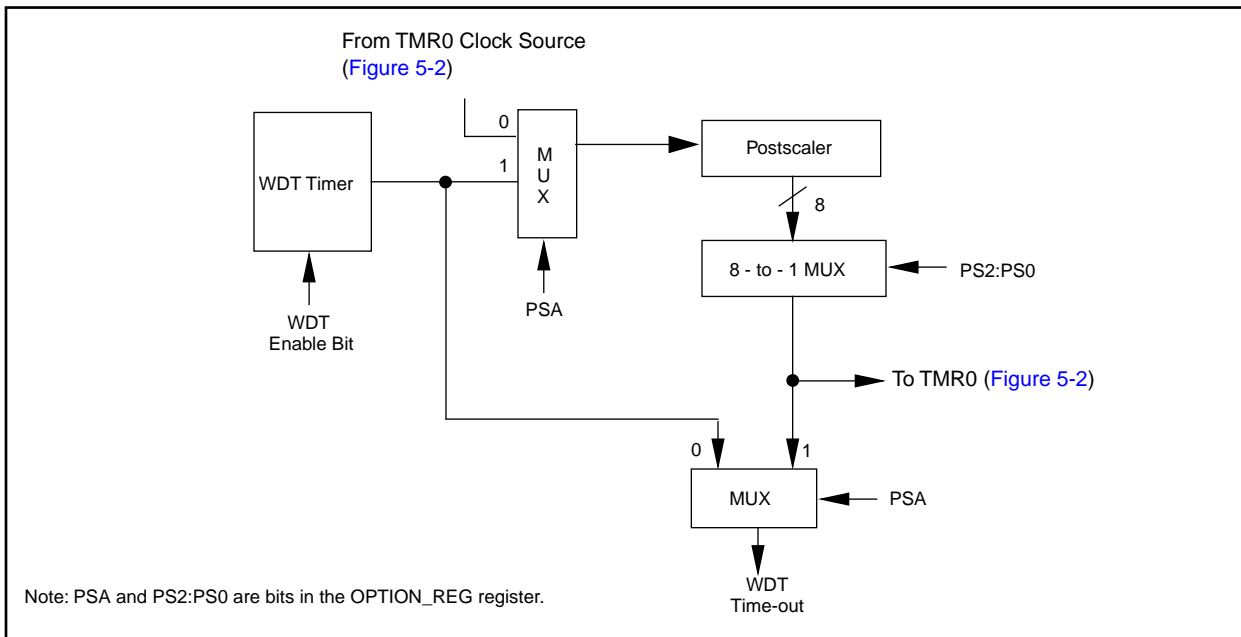
The WDT can be permanently disabled by clearing configuration bit WDTE (Section 12.1).

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 12-12: WATCHDOG TIMER BLOCK DIAGRAM**



**FIGURE 12-13: 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	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

Legend: Shaded cells are not used by the Watchdog Timer.

Note 1: See Figure 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 VDD, or VSS, ensure no external circuitry is drawing current from the I/O pin, power-down the A/D, 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 VDD or VSS 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 (VIHMC).

### 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/I<sup>2</sup>C).
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

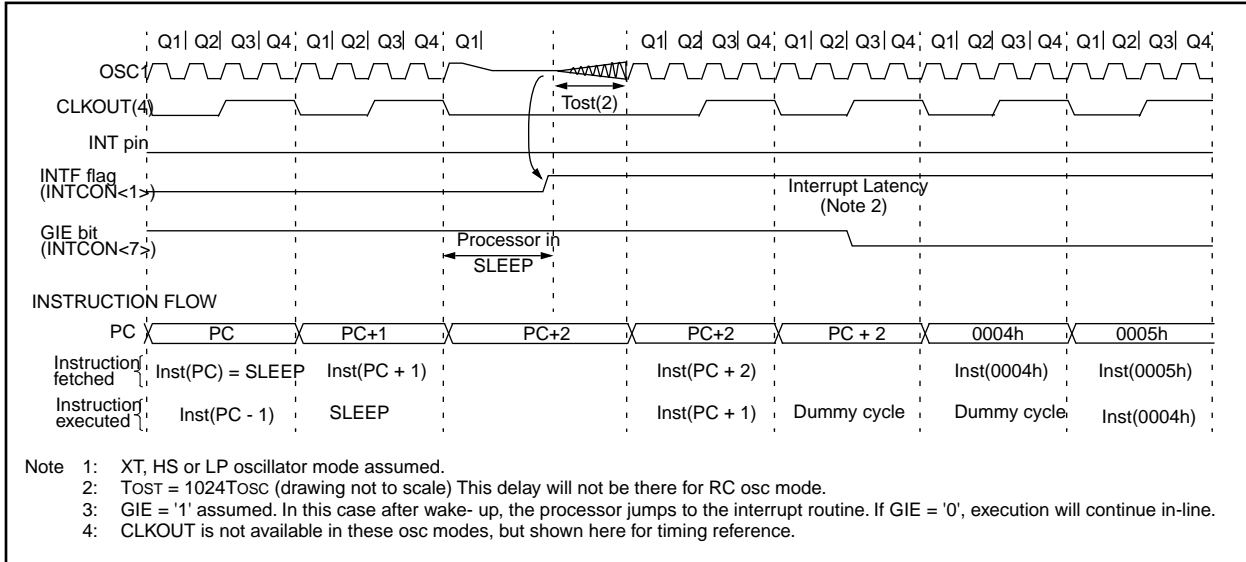
When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- 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 12-14: 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	Last 100h words
Data Memory	TBD

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.

For ROM devices, these values are submitted along with the ROM code.

## 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.

The PIC16F87X devices can be programmed only (no bulk operations such as erase) over the entire V<sub>DD</sub> operating range using ICSP. Please refer to the PIC16F87X Programming Specification, (DS39025).

For complete details of serial programming, please refer to the In-Circuit Serial Programming (ICSP™) Guide, (DS30277B).

## 12.18 Low Voltage Programming

The LVP bit of the configuration word enables low voltage programming. This mode allows the microcontroller to be programmed via ICSP using a V<sub>DD</sub> source in the operating voltage range. This mode removes the requirement of V<sub>IHH</sub> to be placed on the MCLR pin. The LVP bit is normally erased to '1' which enables the low voltage programming. In this mode, the RB3/PGM pin is dedicated to the programming function and ceases to be a general purpose I/O pin. V<sub>DD</sub> is applied to the MCLR pin during low voltage programming. The device will enter programming mode when a '1' is placed on the RB3/PGM pin.

- Note 1:** The high voltage programming mode is always available, regardless of the state of the LVP bit, by applying V<sub>IHH</sub> to the MCLR pin.
- 2:** While in this mode the RB3 pin can no longer be used as a general purpose I/O pin.
- 3:** V<sub>DD</sub> must be 5.0V ±10% during erase/program operations while in low voltage programming mode.

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. To program the device, V<sub>IHH</sub> must be placed onto MCLR during programming. The LVP bit may only be programmed when programming is entered with V<sub>IHH</sub> on MCLR. The LVP bit cannot be programmed when programming is entered with RB3/PGM.

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.

# PIC16F87X

---

NOTES:

## 13.0 INSTRUCTION SET SUMMARY

Each PIC16CXXX 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 μs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs.

Table 13-2 lists the instructions recognized by the MPASM assembler.

Figure 13-1 shows the general formats that the instructions can have.

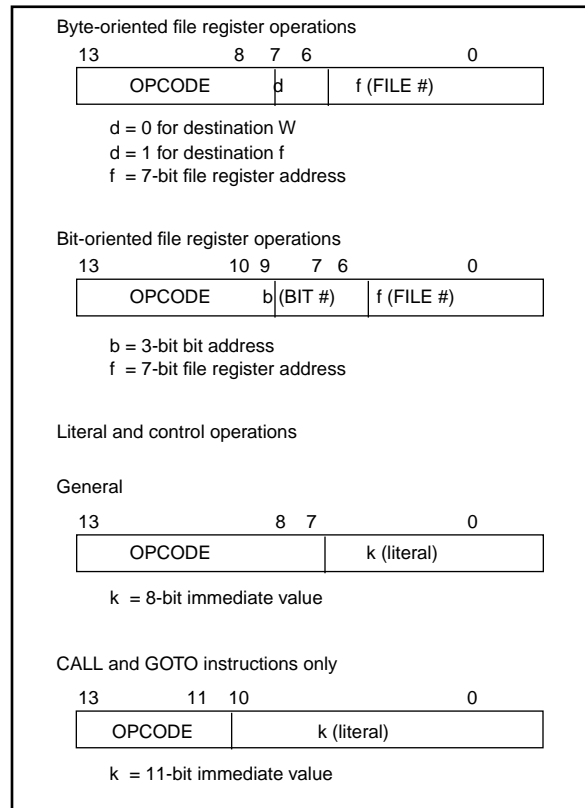
**Note:** To maintain upward compatibility with future PIC16CXXX products, do not use the OPTION and TRIS instructions.

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>								
<b>ADDWF</b>	f, d	Add W and f	1	00	0111	dfff ffff	C,DC,Z	1,2
<b>ANDWF</b>	f, d	AND W with f	1	00	0101	dfff ffff	Z	1,2
<b>CLRF</b>	f	Clear f	1	00	0001	1fff ffff	Z	2
<b>CLRW</b>	-	Clear W	1	00	0001	0xxx xxxx	Z	
<b>COMF</b>	f, d	Complement f	1	00	1001	dfff ffff	Z	1,2
<b>DECF</b>	f, d	Decrement f	1	00	0011	dfff ffff	Z	1,2
<b>DECFSZ</b>	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff ffff		1,2,3
<b>INCF</b>	f, d	Increment f	1	00	1010	dfff ffff	Z	1,2
<b>INCFSZ</b>	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff ffff		1,2,3
<b>IORWF</b>	f, d	Inclusive OR W with f	1	00	0100	dfff ffff	Z	1,2
<b>MOVF</b>	f, d	Move f	1	00	1000	dfff ffff	Z	1,2
<b>MOVWF</b>	f	Move W to f	1	00	0000	1fff ffff		
<b>NOP</b>	-	No Operation	1	00	0000	0xx0 0000		
<b>RLF</b>	f, d	Rotate Left f through Carry	1	00	1101	dfff ffff	C	1,2
<b>RRF</b>	f, d	Rotate Right f through Carry	1	00	1100	dfff ffff	C	1,2
<b>SUBWF</b>	f, d	Subtract W from f	1	00	0010	dfff ffff	C,DC,Z	1,2
<b>SWAPF</b>	f, d	Swap nibbles in f	1	00	1110	dfff ffff		1,2
<b>XORWF</b>	f, d	Exclusive OR W with f	1	00	0110	dfff ffff	Z	1,2
<b>BIT-ORIENTED FILE REGISTER OPERATIONS</b>								
<b>BCF</b>	f, b	Bit Clear f	1	01	00bb	bfff ffff		1,2
<b>BSF</b>	f, b	Bit Set f	1	01	01bb	bfff ffff		1,2
<b>BTFSC</b>	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff ffff		3
<b>BTFSS</b>	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff ffff		3
<b>LITERAL AND CONTROL OPERATIONS</b>								
<b>ADDLW</b>	k	Add literal and W	1	11	111x	kkkk kkkk	C,DC,Z	
<b>ANDLW</b>	k	AND literal with W	1	11	1001	kkkk kkkk	Z	
<b>CALL</b>	k	Call subroutine	2	10	0kkk	kkkk kkkk		
<b>CLRWDT</b>	-	Clear Watchdog Timer	1	00	0000	0110 0100	$\overline{TO}, \overline{PD}$	
<b>GOTO</b>	k	Go to address	2	10	1kkk	kkkk kkkk		
<b>IORLW</b>	k	Inclusive OR literal with W	1	11	1000	kkkk kkkk	Z	
<b>MOVLW</b>	k	Move literal to W	1	11	00xx	kkkk kkkk		
<b>RETFIE</b>	-	Return from interrupt	2	00	0000	0000 1001		
<b>RETLW</b>	k	Return with literal in W	2	11	01xx	kkkk kkkk		
<b>RETURN</b>	-	Return from Subroutine	2	00	0000	0000 1000		
<b>SLEEP</b>	-	Go into standby mode	1	00	0000	0110 0011	$\overline{TO}, \overline{PD}$	
<b>SUBLW</b>	k	Subtract W from literal	1	11	110x	kkkk kkkk	C,DC,Z	
<b>XORLW</b>	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.



## 14.0 DEVELOPMENT SUPPORT

### 14.1 Development Tools

The PICmicro<sup>®</sup> microcontrollers are supported with a full range of hardware and software development tools:

- MPLAB<sup>™</sup>-ICE Real-Time In-Circuit Emulator
- ICEPIC<sup>™</sup> Low-Cost PIC16C5X and PIC16CXXX In-Circuit Emulator
- PRO MATE<sup>®</sup> II Universal Programmer
- PICSTART<sup>®</sup> Plus Entry-Level Prototype Programmer
- SIMICE
- PICDEM-1 Low-Cost Demonstration Board
- PICDEM-2 Low-Cost Demonstration Board
- PICDEM-3 Low-Cost Demonstration Board
- MPASM Assembler
- MPLAB<sup>™</sup> SIM Software Simulator
- MPLAB-C17 (C Compiler)
- Fuzzy Logic Development System (*fuzzyTECH<sup>®</sup>-MP*)
- KEELOQ<sup>®</sup> Evaluation Kits and Programmer

### 14.2 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). MPLAB-ICE is supplied with 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 all new Microchip 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 compatible 386 (and higher) machine platform and Microsoft Windows<sup>®</sup> 3.x or Windows 95 environment were chosen to best make these features available to you, the end user.

MPLAB-ICE is available in two versions. MPLAB-ICE 1000 is a basic, low-cost emulator system with simple trace capabilities. It shares processor modules with the MPLAB-ICE 2000. This is a full-featured emulator system with enhanced trace, trigger, and data monitoring features. Both systems will operate across the entire operating speed range of the PICmicro MCU.

### 14.3 ICEPIC: Low-Cost PICmicro In-Circuit Emulator

ICEPIC is a low-cost in-circuit emulator solution for the Microchip PIC12CXXX, PIC16C5X and PIC16CXXX families of 8-bit OTP microcontrollers.

ICEPIC is designed to operate on PC-compatible machines ranging from 386 through Pentium<sup>™</sup> based machines under Windows 3.x, Windows 95, or Windows NT environment. ICEPIC features real time, non-intrusive emulation.

### 14.4 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 displaying 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 PIC12CXXX, PIC14C000, PIC16C5X, PIC16CXXX and PIC17CXX devices. It can also set configuration and code-protect bits in this mode.

### 14.5 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 is not recommended for production programming.

PICSTART Plus supports all PIC12CXXX, PIC14C000, PIC16C5X, PIC16CXXX and PIC17CXX devices with up to 40 pins. Larger pin count devices such as the PIC16C923, PIC16C924 and PIC17C756 may be supported with an adapter socket. PICSTART Plus is CE compliant.

### 14.6 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<sup>™</sup>-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.

## **14.7 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.

## **14.8 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.

## **14.9 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 program-

mer 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.

## **14.10 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 based application which contains:

- A full featured editor
- Three operating modes
  - editor
  - emulator
  - simulator
- A project manager
- Customizable tool bar and key mapping
- A status bar with project information
- Extensive 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

The ability to use MPLAB with Microchip's simulator allows a consistent platform and the ability to easily switch from the low cost simulator to the full featured emulator with minimal retraining due to development tools.

## **14.11 Assembler (MPASM)**

The MPASM Universal Macro Assembler is a PC-hosted symbolic assembler. It supports all microcontroller series including the PIC12C5XX, PIC14000, PIC16C5X, PIC16CXXX, and PIC17CXX families.

MPASM offers full featured Macro capabilities, conditional assembly, and several source and listing formats. It generates various object code formats to support Microchip's development tools as well as third party programmers.

MPASM allows full symbolic debugging from MPLAB-ICE, Microchip's Universal Emulator System.

MPASM has the following features to assist in developing software for specific use applications.

- Provides translation of Assembler source code to object code for all Microchip microcontrollers.
- Macro assembly capability.
- Produces all the files (Object, Listing, Symbol, and special) required for symbolic debug with Microchip's emulator systems.
- Supports Hex (default), Decimal and Octal source and listing formats.

MPASM provides a rich directive language to support programming of the PICmicro. Directives are helpful in making the development of your assemble source code shorter and more maintainable.

#### **14.12 Software Simulator (MPLAB-SIM)**

The MPLAB-SIM Software Simulator allows code development in a PC host environment. It allows the user to simulate the PICmicro series microcontrollers on an instruction level. On any given instruction, the user may examine or modify any of the data areas or provide external stimulus to any of the pins. The input/output radix can be set by the user and the execution can be performed in; single step, execute until break, or in a trace mode.

MPLAB-SIM fully supports symbolic debugging using MPLAB-C17 and MPASM. The Software Simulator offers the low cost flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

#### **14.13 MPLAB-C17 Compiler**

The MPLAB-C17 Code Development System is a complete ANSI 'C' compiler and integrated development environment for Microchip's PIC17CXXX family of microcontrollers. The compiler provides powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compiler provides symbol information that is compatible with the MPLAB IDE memory display.

#### **14.14 Fuzzy Logic Development System (fuzzyTECH-MP)**

*fuzzyTECH-MP* fuzzy logic development tool is available in two versions - a low cost introductory version, MP Explorer, for designers to gain a comprehensive working knowledge of fuzzy logic system design; and a full-featured version, *fuzzyTECH-MP*, Edition for implementing more complex systems.

Both versions include Microchip's *fuzzyLAB*<sup>™</sup> demonstration board for hands-on experience with fuzzy logic systems implementation.

#### **14.15 SEEVAL<sup>®</sup> 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<sup>™</sup> and secure serials. The Total Endurance<sup>™</sup> 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.

#### **14.16 KEELOQ<sup>®</sup> 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.

# PIC16F87X

TABLE 14-1 DEVELOPMENT TOOLS FROM MICROCHIP

	PIC12C5XX	PIC14000	PIC16C5X	PIC16CXX	PIC16C6X	PIC16C7XX	PIC16C8X	PIC16C9XX	PIC17C4X	PIC17C7XX	24CXX 25CXX 93CXX	HCS200 HCS300 HCS301
<b>Emulator Products</b>												
MPLAB™-ICE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
ICEPIC™ Low-Cost In-Circuit Emulator			✓	✓	✓	✓	✓	✓				
<b>Software Tools</b>												
MPLAB™ Integrated Development Environment	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
MPLAB™ C17* Compiler									✓	✓		
fuzzyTECH®-MP Explorer/Edition Fuzzy Logic Dev. Tool	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Total Endurance™ Software Model											✓	
<b>Programmers</b>												
PICSTART® Plus Low-Cost Universal Dev. Kit	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
PRO MATE® II Universal Programmer	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
KEELOQ® Programmer												✓
SEEVAL® Designers Kit											✓	
<b>Demo Boards</b>												
SIMICE	✓		✓									
PICDEM-14A		✓										
PICDEM-1			✓	✓			✓		✓			
PICDEM-2					✓	✓						
PICDEM-3								✓				
KEELOQ® Evaluation Kit												✓
KEELOQ Transponder Kit												✓

## 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:  $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$

**Note 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.

**Note 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

**TABLE 15-1 CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)**

OSC	PIC16F873-04 PIC16F874-04 PIC16F876-04 PIC16F877-04	PIC16F873-20 PIC16F874-20 PIC16F876-20 PIC16F877-20	PIC16LF873-04 PIC16LF874-04 PIC16LF876-04 PIC16LF877-04
RC	VDD: 4.0V to 5.5V IDD: 5 mA max. at 5.5V IPD: 16 $\mu$ A max. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.0 mA typ. at 5.5V IPD: 1.5 $\mu$ A typ. at 4V Freq: 4 MHz max.	VDD: 2.0V to 5.5V IDD: 3.8 mA max. at 3.0V IPD: 5 $\mu$ A max. at 3V Freq: 4 MHz max.
XT	VDD: 4.0V to 5.5V IDD: 5 mA max. at 5.5V IPD: 16 $\mu$ A max. at 4V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 2.0 mA typ. at 5.5V IPD: 1.5 $\mu$ A typ. at 4V Freq: 4 MHz max.	VDD: 2.0V to 5.5V IDD: 3.8 mA max. at 3.0V IPD: 5 $\mu$ A max. at 3V Freq: 4 MHz max.
HS	VDD: 4.5V to 5.5V IDD: 13.5 mA typ. at 5.5V IPD: 1.5 $\mu$ A typ. at 4.5V Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 20 mA max. at 5.5V IPD: 1.5 $\mu$ A typ. at 4.5V Freq: 20 MHz max.	Not recommended for use in HS mode
LP	VDD: 4.0V to 5.5V IDD: 52.5 $\mu$ A typ. at 32 kHz, 4.0V IPD: 0.9 $\mu$ A typ. at 4.0V Freq: 200 kHz max.	Not recommended for use in LP mode	VDD: 2.0V to 5.5V IDD: 48 $\mu$ A max. at 32 kHz, 3.0V IPD: 5.0 $\mu$ A max. at 3.0V Freq: 200 kHz max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that ensures the specifications required.

**Note:** This is an advanced copy of the data sheet and therefore the contents and specifications are subject to change based on device characterization.

**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°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D001 D001A	Supply Voltage	VDD	4.0 4.5	- -	5.5 5.5	V V	XT, RC and LP osc configuration HS osc configuration
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	BVDD	3.7	4.0	4.3	V	BODEN bit in configuration word enabled
D010	Supply Current (Note 2,5)	IDD	-	2.0	5	mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 5.5V (Note 4)
D013			-	10	20	mA	HS osc configuration FOSC = 20 MHz, VDD = 5.5V
D015*	Brown-out Reset Current (Note 6)	ΔIBOR	-	85	200	μA	BOR enabled VDD = 5.0V
D020 D021 D021A D021B	Power-down Current (Note 3,5)	IPD	-	10.5 1.5 1.5 2.5	42 16 19 19	μA μA μA μA	VDD = 4.0V, WDT enabled, -40°C to +85°C VDD = 4.0V, WDT disabled, -0°C to +70°C VDD = 4.0V, WDT disabled, -40°C to +85°C VDD = 4.0V, WDT disabled, -40°C to +125°C
D023*	Brown-out Reset Current (Note 6)	ΔIBOR	-	85	200	μA	BOR enabled VDD = 5.0V

\* 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 μA to the specification. This value is from characterization and is for design guidance only. This is not tested.

6: The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

# PIC16F87X

## 15.2 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	BVDD	3.7	4.0	4.3	V	BODEN bit in configuration word enabled
D010	Supply Current (Note 2,5)	IDD	-	2.0	3.8	mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 3.0V (Note 4)
D010A			-	20	48	$\mu\text{A}$	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

\* 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.



**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)**

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					
		Operating voltage $V_{DD}$ range as described in DC spec <a href="#">Section 15.1</a> and <a href="#">Section 15.2</a> .					
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
<b>Input Low Voltage</b>							
D030	I/O ports	$V_{IL}$	$V_{SS}$	-	$0.15V_{DD}$	V	For entire $V_{DD}$ range
D030A	with TTL buffer		$V_{SS}$	-	$0.8\text{V}$	V	$4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$
D031	with Schmitt Trigger buffer		$V_{SS}$	-	$0.2V_{DD}$	V	
D032	$\overline{\text{MCLR}}$ , OSC1 (in RC mode)		$V_{SS}$	-	$0.2V_{DD}$	V	
D033	OSC1 (in XT, HS and LP) Ports RC3 and RC4		$V_{SS}$	-	$0.3V_{DD}$	V	Note1
D034	with Schmitt Trigger buffer		$V_{SS}$	-	$0.3V_{DD}$	V	For entire $V_{DD}$ range
D034A	with SMBus		-0.5	-	0.6	V	for $V_{DD} = 4.5$ to $5.5\text{V}$
<b>Input High Voltage</b>							
D040	I/O ports	$V_{IH}$		-			
D040A	with TTL buffer		2.0	-	$V_{DD}$	V	$4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$
D041	with Schmitt Trigger buffer		$0.25V_{DD} + 0.8\text{V}$	-	$V_{DD}$	V	For entire $V_{DD}$ range
D042	$\overline{\text{MCLR}}$		$0.8V_{DD}$	-	$V_{DD}$	V	
D042A	OSC1 (XT, HS and LP)		$0.7V_{DD}$	-	$V_{DD}$	V	Note1
D043	OSC1 (in RC mode) Ports RC3 and RC4		$0.9V_{DD}$	-	$V_{DD}$	V	
D044	with Schmitt Trigger buffer		$0.7V_{DD}$	-	$V_{DD}$	V	For entire $V_{DD}$ range
D044A	with SMBus		1.4	-	5.5	V	for $V_{DD} = 4.5$ to $5.5\text{V}$
D070	PORTB weak pull-up current	$I_{PURB}$	50	250	400	$\mu\text{A}$	$V_{DD} = 5\text{V}$ , $V_{PIN} = V_{SS}$
<b>Input Leakage Current</b> (Notes 2, 3)							
D060	I/O ports	$I_{IL}$	-	-	$\pm 1$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , Pin at hi-impedance
D061	$\overline{\text{MCLR}}$ , RA4/T0CKI		-	-	$\pm 5$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$
D063	OSC1		-	-	$\pm 5$	$\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , XT, HS and LP osc configuration

\* 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.
- The leakage current on the  $\overline{\text{MCLR}}$  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.
  - Negative current is defined as current sourced by the pin.

# PIC16F87X

<b>DC CHARACTERISTICS</b> <b>Standard Operating Conditions (unless otherwise stated)</b> 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 <a href="#">Section 15.1</a> and <a href="#">Section 15.2</a> .							
Param No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D080	<b>Output Low Voltage</b> I/O ports	$V_{OL}$	-	-	0.6	V	$I_{OL} = 8.5\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , -40°C to +85°C
D080A			-	-	0.6	V	$I_{OL} = 7.0\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , -40°C to +125°C
D083	OSC2/CLKOUT (RC osc config)		-	-	0.6	V	$I_{OL} = 1.6\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , -40°C to +85°C
D083A			-	-	0.6	V	$I_{OL} = 1.2\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , -40°C to +125°C
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°C to +85°C
D090A			$V_{DD} - 0.7$	-	-	V	$I_{OH} = -2.5\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , -40°C to +125°C
D092	OSC2/CLKOUT (RC osc config)		$V_{DD} - 0.7$	-	-	V	$I_{OH} = -1.3\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , -40°C to +85°C
D092A			$V_{DD} - 0.7$	-	-	V	$I_{OH} = -1.0\text{ mA}$ , $V_{DD} = 4.5\text{V}$ , -40°C to +125°C
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		CB	-	-	400	pF	
<b>Data EEPROM Memory</b>							
D120	Endurance	ED	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	TDEW	-	4	10	ms	
<b>Program FLASH Memory</b>							
D130	Endurance	EP	1000	-	-	E/W	25°C at 5V
D131	$V_{DD}$ for read	$V_{PR}$	$V_{min}$	-	5.5	V	$V_{min}$ = min operating voltage
D132	$V_{DD}$ for erase/write	$V_{PEW}$	4.5	-	5.5	V	using ICSP port
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	TPEW	-	4	10	ms	

\* 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.

## 15.4 Timing Parameter Symbology

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	$\overline{\text{CS}}$	rw	RD or WR
di	SDI	sc	SCK
do	SDO	ss	SS
dt	Data in	t0	T0CKI
io	I/O port	t1	T1CKI
mc	$\overline{\text{MCLR}}$	wr	WR

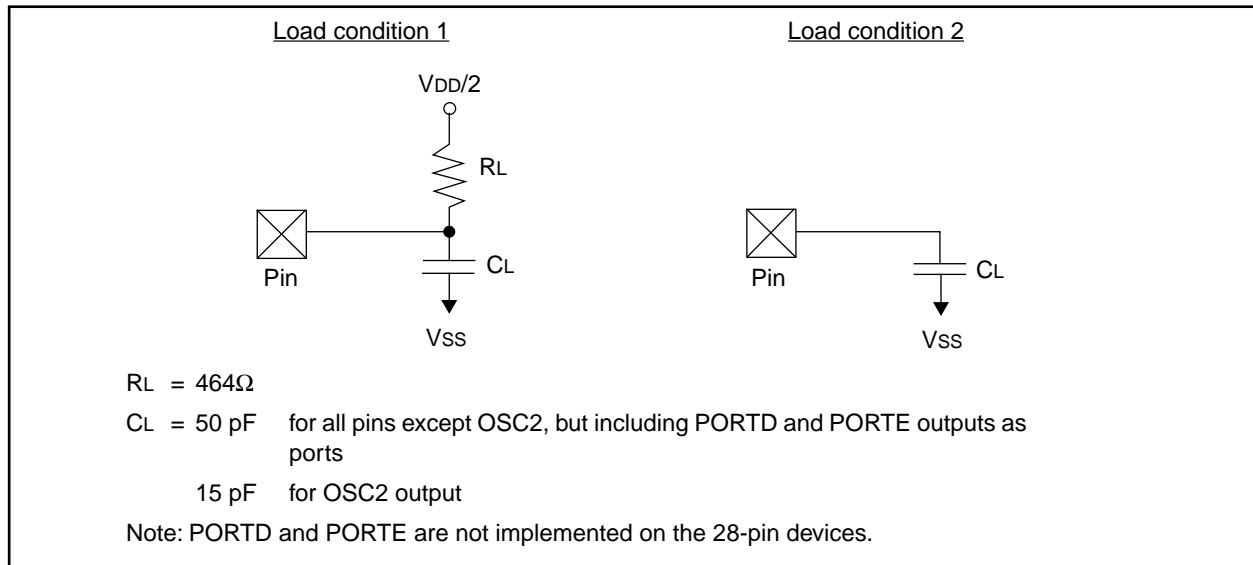
Uppercase letters and their meanings:

<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>		High	High
AA	output access	Low	Low
BUF	Bus free		

Tcc:st (I<sup>2</sup>C specifications only)

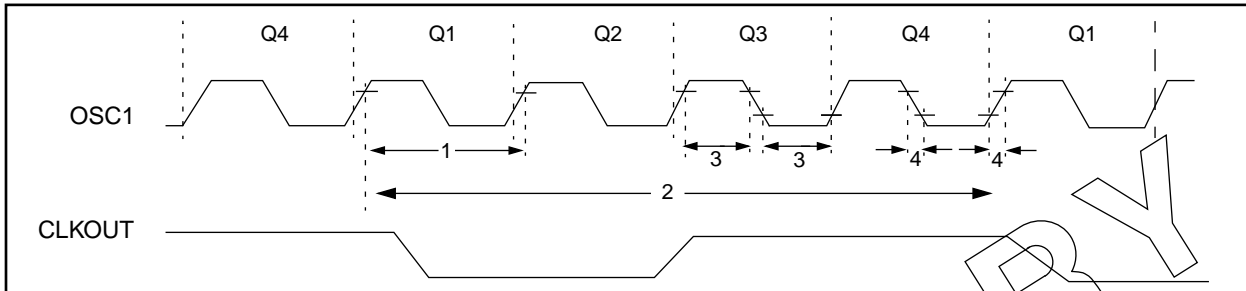
<b>CC</b>			
HD	Hold	SU	Setup
<b>ST</b>		STO	STOP condition
DAT	DATA input hold		
STA	START condition		

**FIGURE 15-1: LOAD CONDITIONS**



# PIC16F87X

**FIGURE 15-2: EXTERNAL CLOCK TIMING**



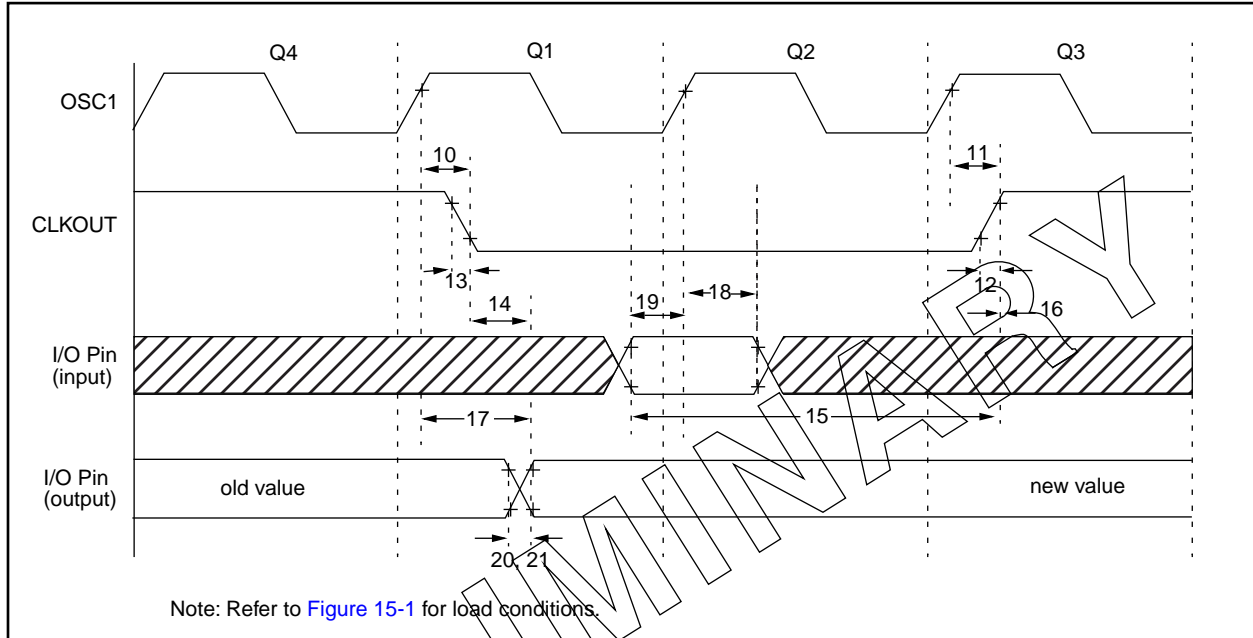
**TABLE 15-2 EXTERNAL CLOCK TIMING REQUIREMENTS**

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	—	20	MHz	HS 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	—	250	ns	HS osc mode (-04)			
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			

† 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.

**FIGURE 15-3: CLKOUT AND I/O TIMING**



**TABLE 15-3 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*	TckH2iol	Port in hold after CLKOUT ↑	0	—	—	ns	Note 1
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	50	150	ns	
18*	TosH2iol	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)	—	—	80	ns
21*	TioF	Port output fall time	Standard (F)	—	10	40	ns
			Extended (LF)	—	—	80	ns
22††*	Tinp	INT pin high or low time	TCY	—	—	ns	
23††*	Trbp	RB7:RB4 change INT high or low time	TCY	—	—	ns	

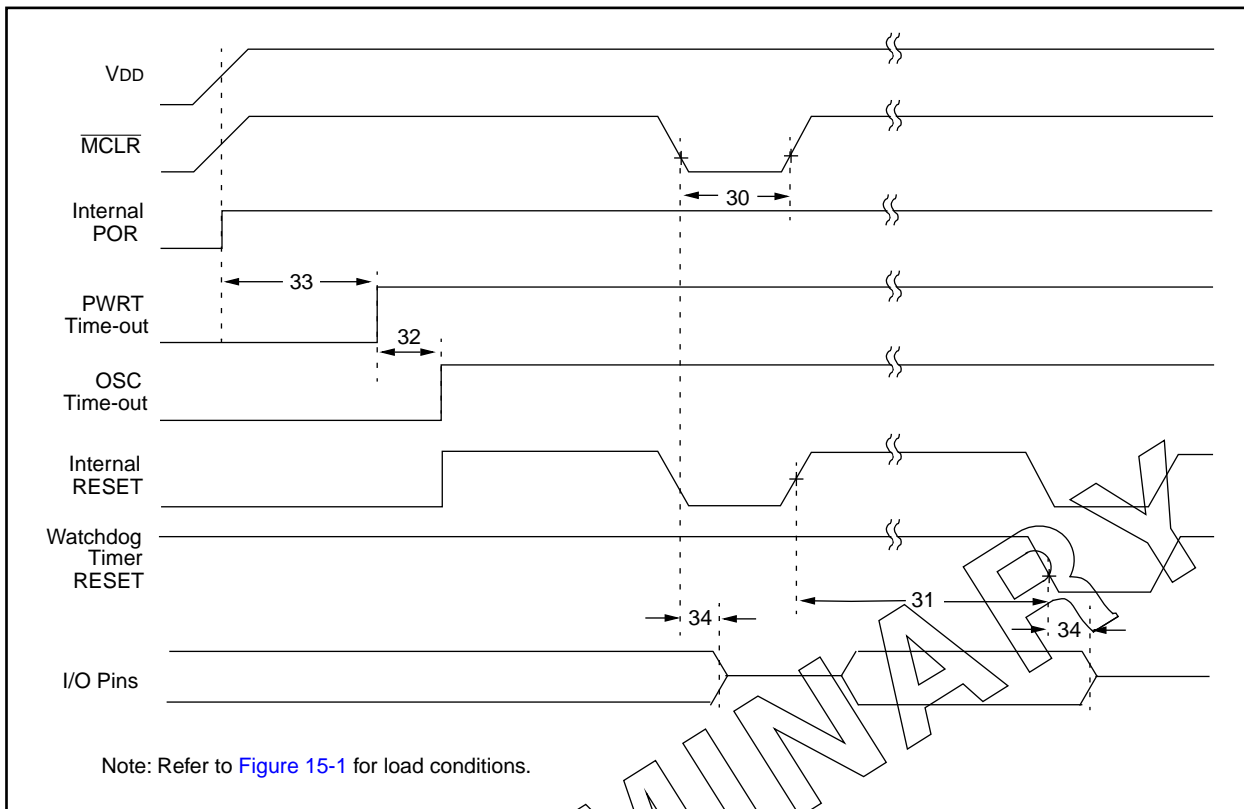
\* These parameters are characterized but not tested.

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

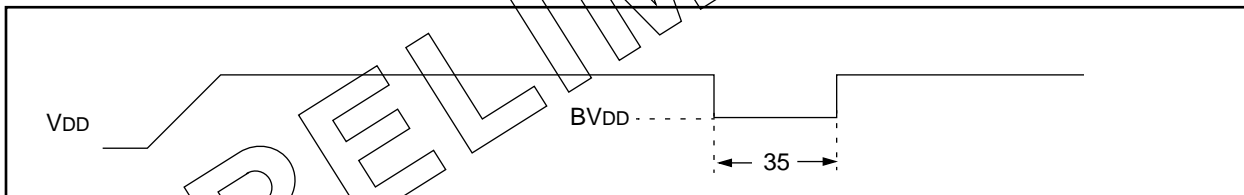
†† These parameters are asynchronous events not related to any internal clock edges.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x Tosc.

**FIGURE 15-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**



**FIGURE 15-5: BROWN-OUT RESET TIMING**



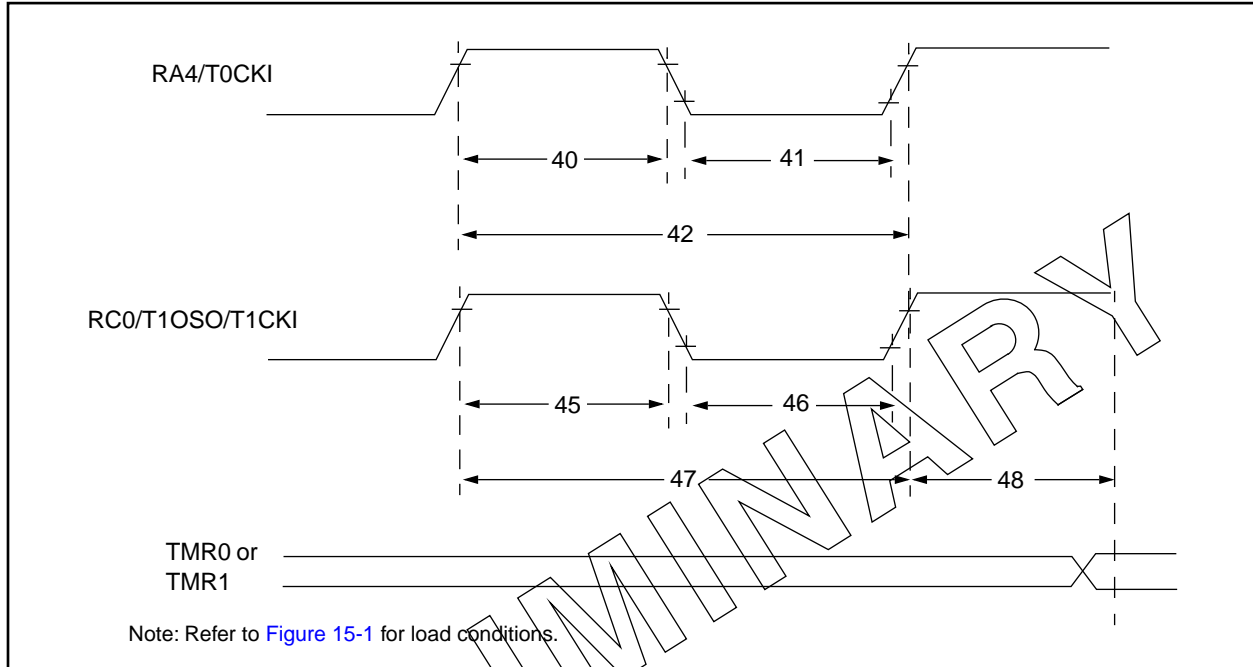
**TABLE 15-4 RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	Tmcl	MCLR Pulse Width (low)	2	—	—	μs	VDD = 5V, -40°C to +125°C
31*	Twdt	Watchdog Timer Time-out Period (No Prescaler)	7	18	33	ms	VDD = 5V, -40°C to +125°C
32	Tost	Oscillation Start-up Timer Period	—	1024TOSC	—	—	TOSC = OSC1 period
33*	Tpwrt	Power up Timer Period	28	72	132	ms	VDD = 5V, -40°C to +125°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 ≤ BVDD (D005)

\* 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 15-6: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**



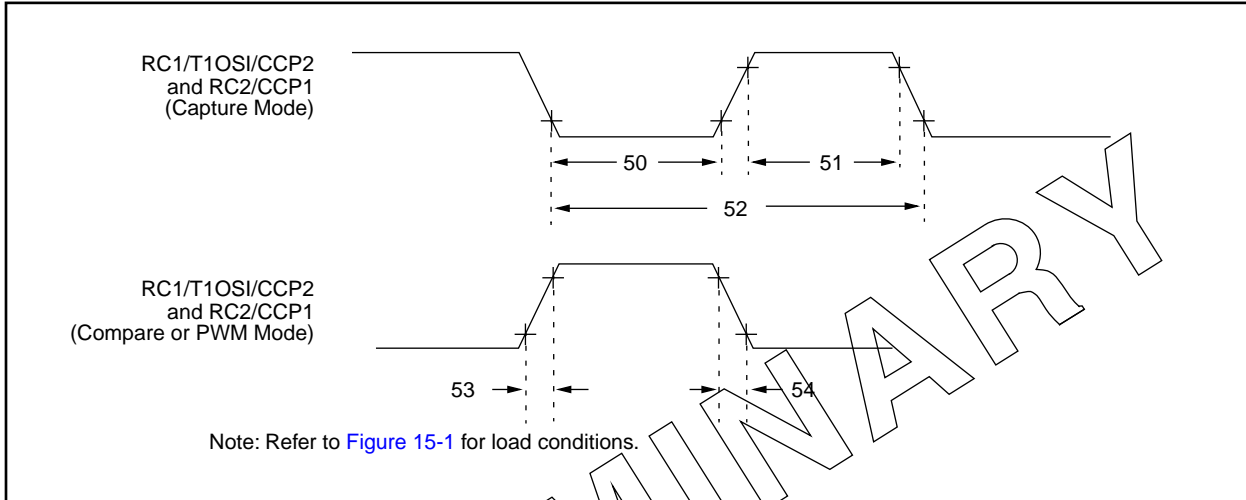
**TABLE 15-5 TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions	
40*	Tt0H	T0CKI High Pulse Width	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	Must also meet parameter 42	
			With Prescaler	10	—	—	ns		
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	Must also meet parameter 42	
			With Prescaler	10	—	—	ns		
42*	Tt0P	T0CKI Period	No Prescaler	$T_{CY} + 40$	—	—	ns	N = prescale value (2, 4, ..., 256)	
			With Prescaler	Greater of: $20$ or $T_{CY} + 40$ N	—	—	ns		
45*	Tt1H	T1CKI High Time	Synchronous, Prescaler = 1	$0.5T_{CY} + 20$	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2,4,8	PIC16F7X PIC16LF7X	15 25	—	—		ns
			Asynchronous	PIC16F7X	30	—	—		ns
				PIC16LF7X	50	—	—		ns
46*	Tt1L	T1CKI Low Time	Synchronous, Prescaler = 1	$0.5T_{CY} + 20$	—	—	ns	Must also meet parameter 47	
			Synchronous, Prescaler = 2,4,8	PIC16F7X PIC16LF7X	15 25	—	—		ns
			Asynchronous	PIC16F7X	30	—	—		ns
				PIC16LF7X	50	—	—		ns
47*	Tt1P	T1CKI input period	Synchronous	PIC16F7X	Greater of: $30$ or $T_{CY} + 40$ N	—	—	ns	N = prescale value (1, 2, 4, 8)
				PIC16LF7X	Greater of: $50$ or $T_{CY} + 40$ N	—	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	PIC16F7X	60	—	—	ns	
				PIC16LF7X	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		$2T_{osc}$	—	$7T_{osc}$	—		

\* 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 15-7: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)**



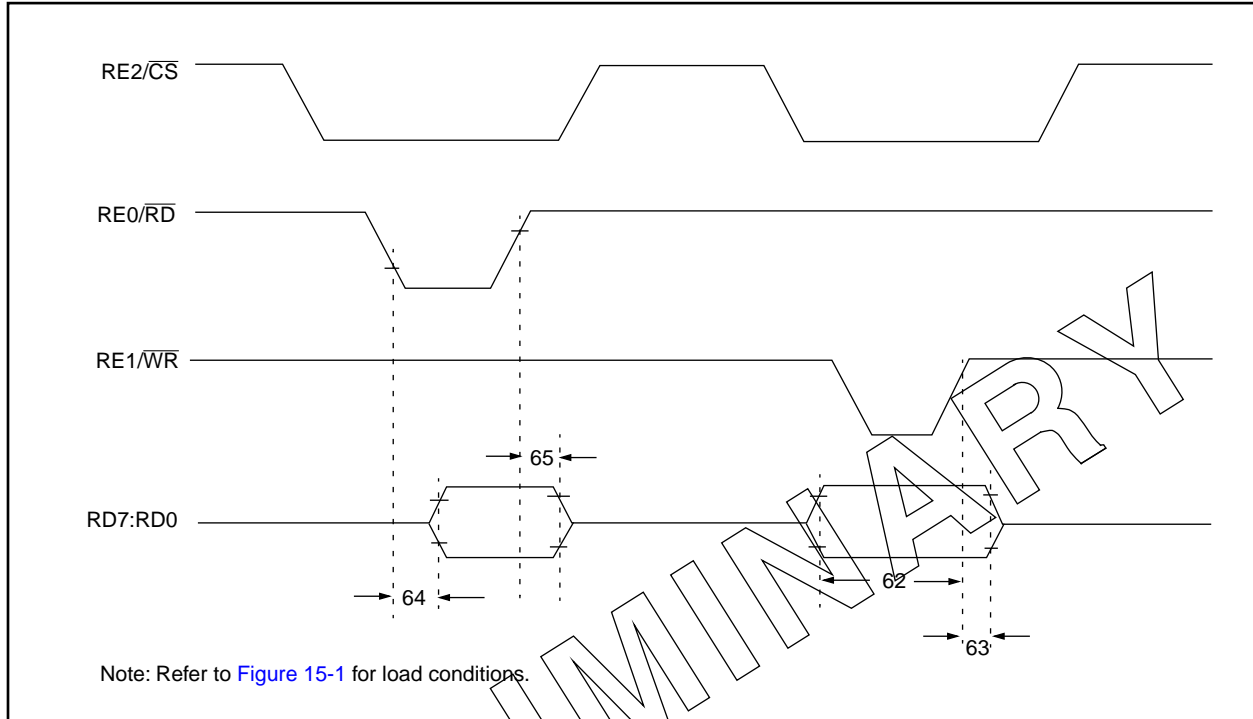
**TABLE 15-6 CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1 AND CCP2)**

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	45	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.



**FIGURE 15-8: PARALLEL SLAVE PORT TIMING (40-PIN DEVICES ONLY)**



**TABLE 15-7 PARALLEL SLAVE PORT REQUIREMENTS (40-PIN DEVICES ONLY)**

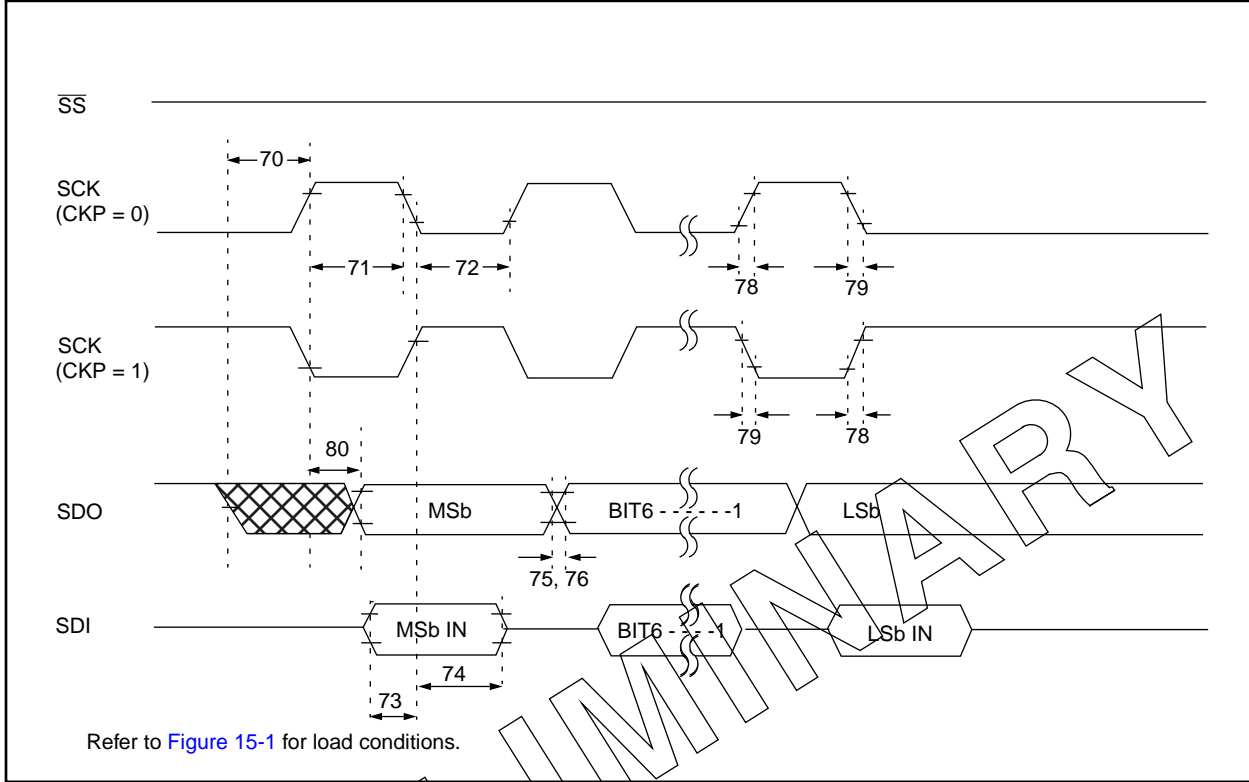
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
62	TdtV2wrH	Data in valid before WR↑ or CS↑ (setup time)	20 25	— —	— —	ns ns	Extended Range Only	
63*	TwrH2dtI	WR↑ or CS↑ to data-in invalid (hold time)	Standard(F)	20	—	—	ns	
			Extended(LF)	35	—	—	ns	
64	TrdL2dtV	RD↓ and CS↓ to data-out 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.

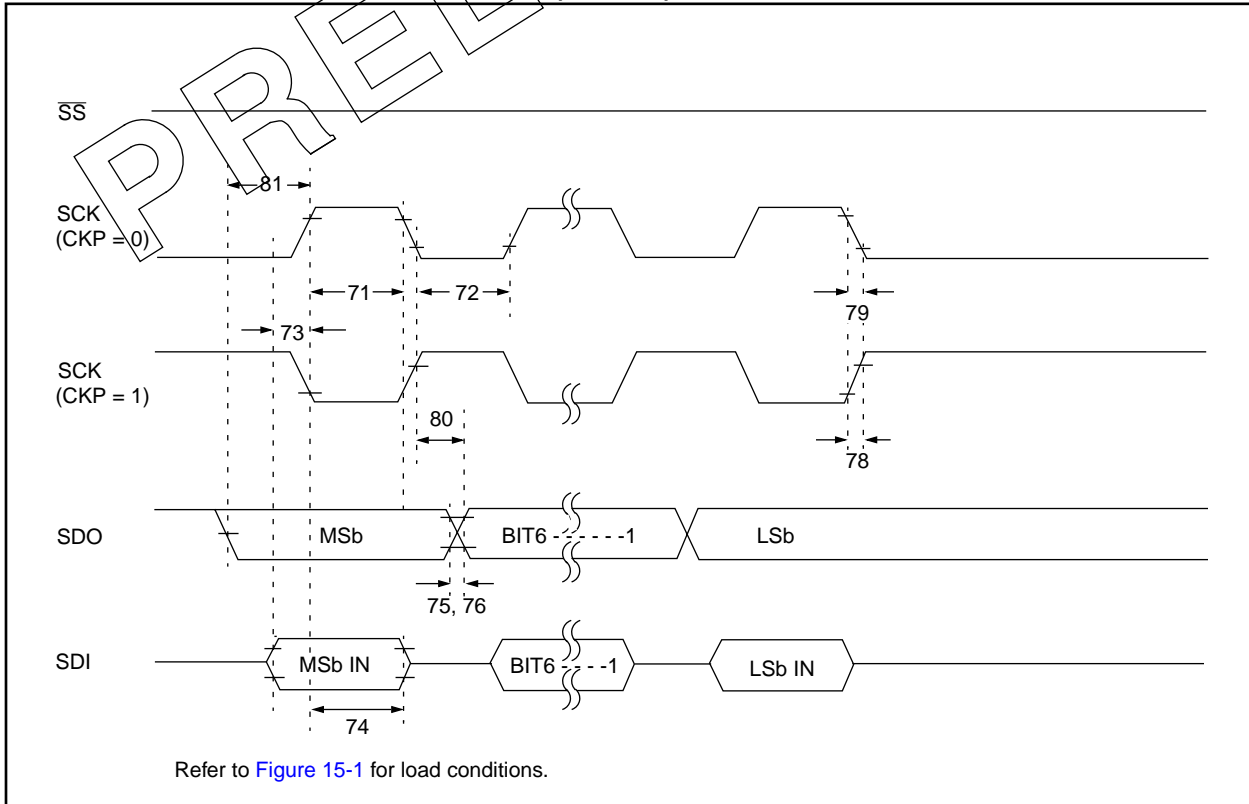
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

# PIC16F87X

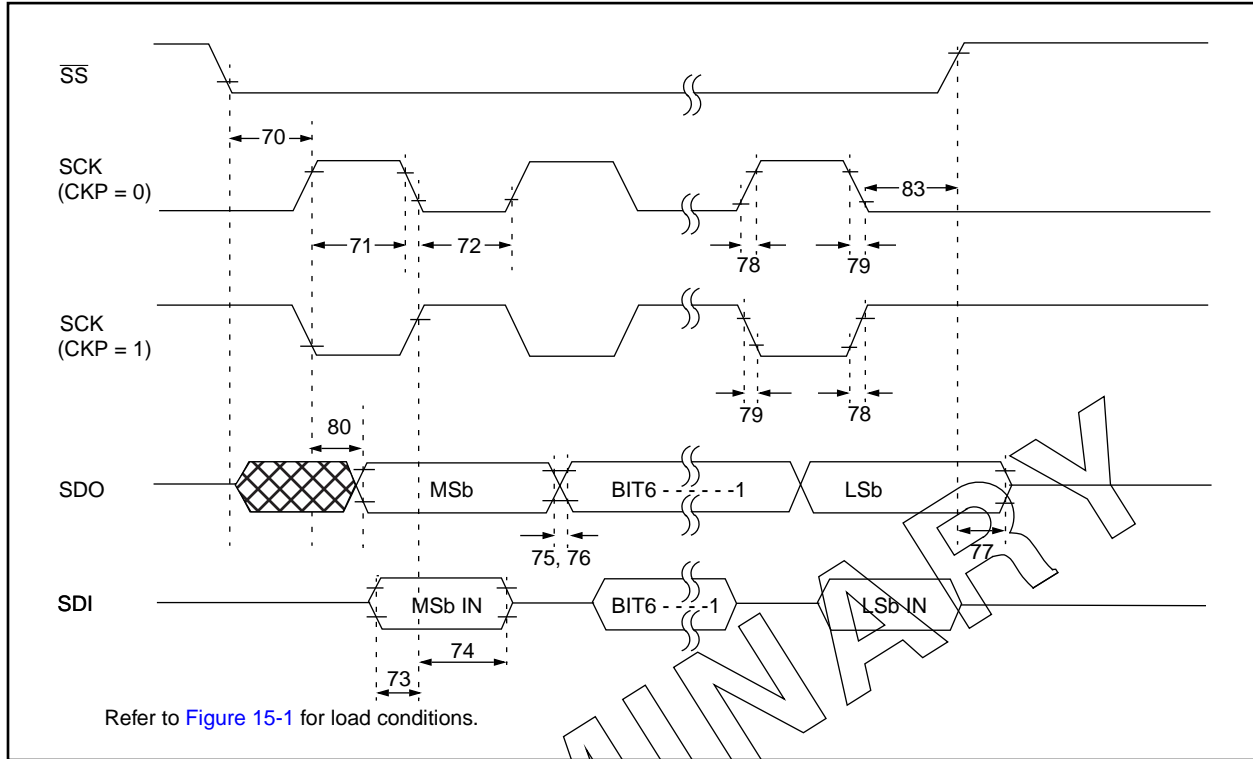
**FIGURE 15-9: SPI MASTER MODE TIMING (CKE = 0)**



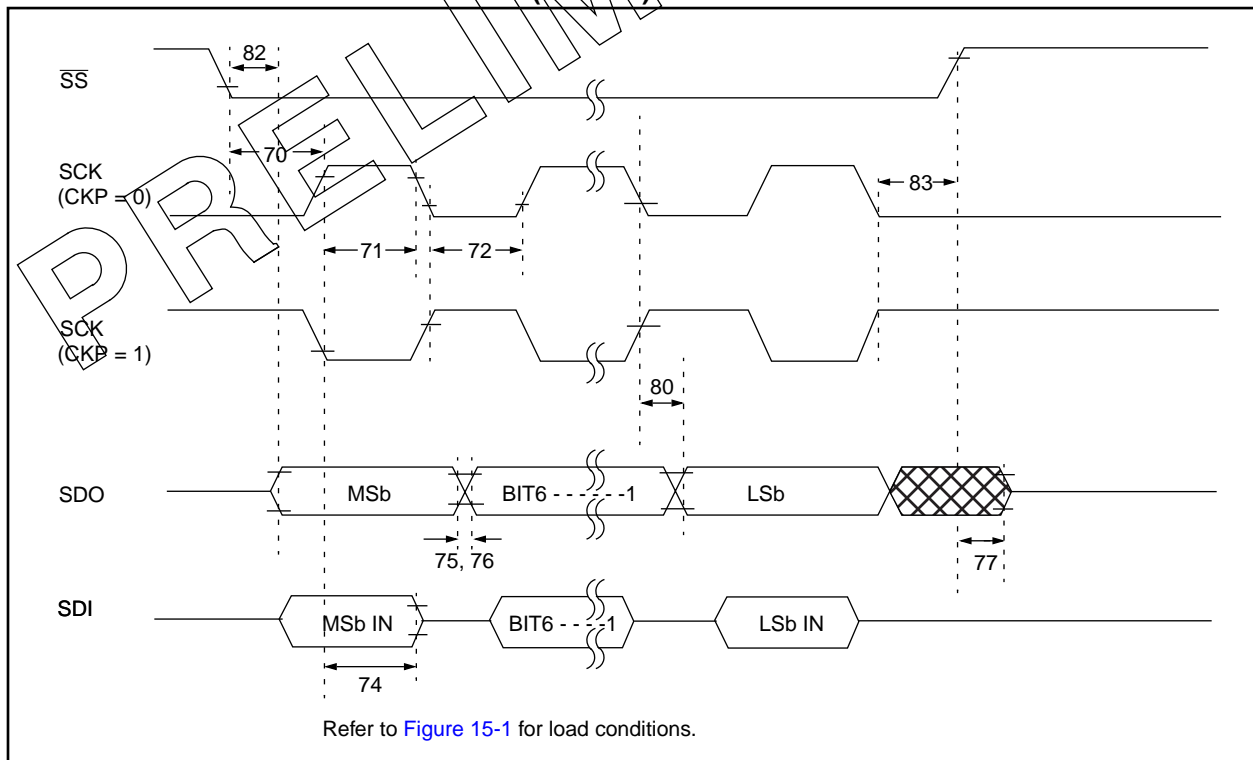
**FIGURE 15-10: SPI MASTER MODE TIMING (CKE = 1)**



**FIGURE 15-11: SPI SLAVE MODE TIMING (CKE = 0)**



**FIGURE 15-12: SPI SLAVE MODE TIMING (CKE = 1)**



# PIC16F87X

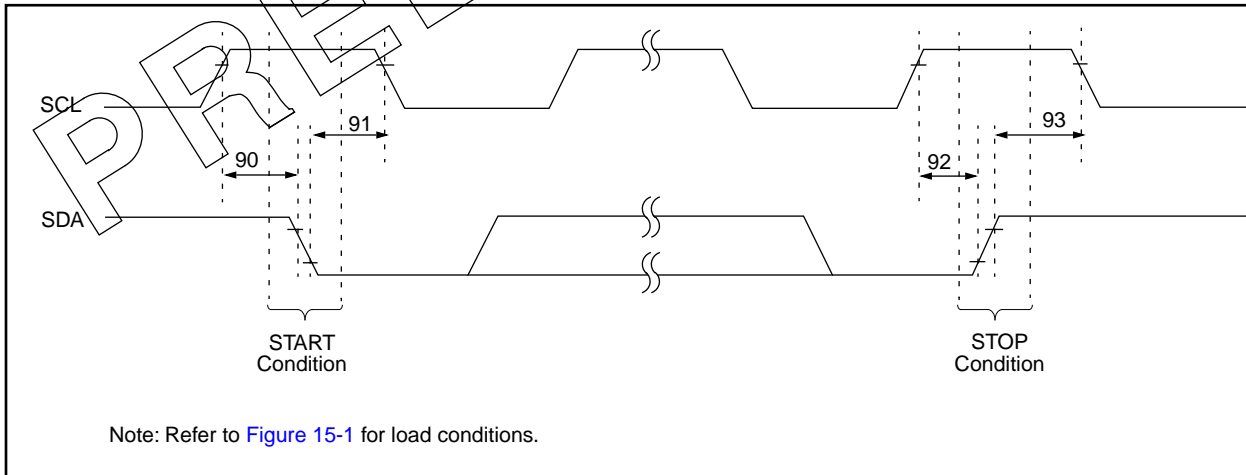
**TABLE 15-8 SPI MODE REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
70*	TssL2scH, TssL2scL	SS↓ to SCK↓ or SCK↑ input	T <sub>CY</sub>	—	—	ns	
71*	TscH	SCK input high time (slave mode)	T <sub>CY</sub> + 20	—	—	ns	
72*	TscL	SCK input low time (slave mode)	T <sub>CY</sub> + 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	—	10	25	ns	
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)	—	10	25	ns	
79*	TscF	SCK output fall time (master mode)	—	10	25	ns	
80*	Tsch2doV, TscL2doV	SDO data output valid after SCK edge	—	—	50	ns	
81*	TdoV2scH, TdoV2scL	SDO data output setup to SCK edge	T <sub>CY</sub>	—	—	ns	
82*	TssL2doV	SDO data output valid after SS↓ edge	—	—	50	ns	
83*	Tsch2ssH, TscL2ssH	SS↑ after SCK edge	1.5T <sub>CY</sub> + 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.

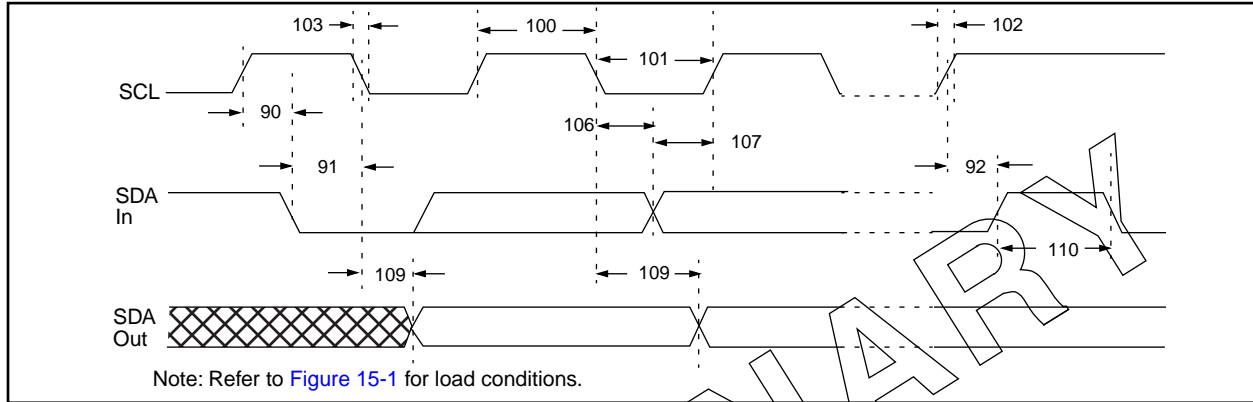
**FIGURE 15-13: I<sup>2</sup>C BUS START/STOP BITS TIMING**



**TABLE 15-9 I<sup>2</sup>C BUS START/STOP BITS REQUIREMENTS**

Parameter No.	Sym	Characteristic	Min	Typ	Max	Units	Conditions	
90	TSU:STA	START condition	100 kHz mode	4700	—	—	ns	Only relevant for repeated START condition
		Setup time	400 kHz mode	600	—	—		
91	THD:STA	START condition	100 kHz mode	4000	—	—	ns	After this period the first clock pulse is generated
		Hold time	400 kHz mode	600	—	—		
92	TSU:STO	STOP condition	100 kHz mode	4700	—	—	ns	
		Setup time	400 kHz mode	600	—	—		
93	THD:STO	STOP condition	100 kHz mode	4000	—	—	ns	
		Hold time	400 kHz mode	600	—	—		

**FIGURE 15-14: I<sup>2</sup>C BUS DATA TIMING**

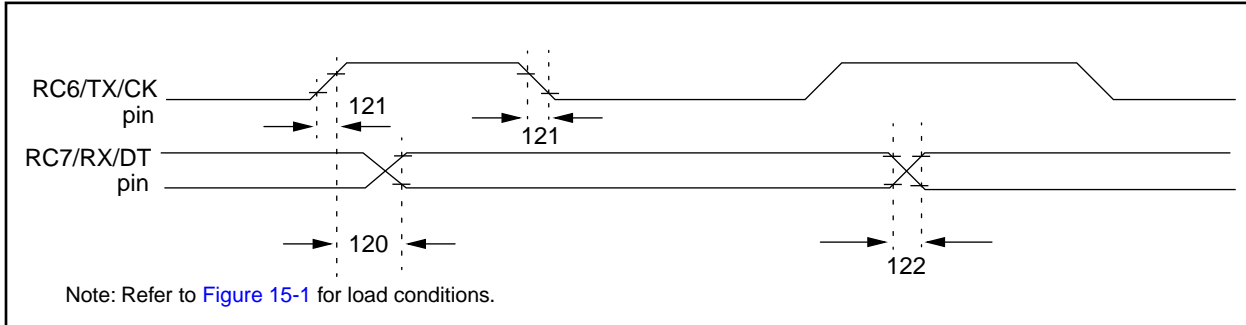


**TABLE 15-10 I<sup>2</sup>C BUS DATA REQUIREMENTS**

Parameter 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
101	TLOW	Clock low time	SSP Module	1.5Tcy	—		
			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
102	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1Cb	300	ns	Cb is specified to be from 10 to 400 pF
103	TF	SDA and SCL fall time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1Cb	300	ns	Cb is specified to be from 10 to 400 pF
90	TSU:STA	START condition setup time	100 kHz mode	4.7	—	μs	Only relevant for repeated START condition
			400 kHz mode	0.6	—	μs	
91	THD:STA	START condition hold time	100 kHz mode	4.0	—	μs	After this period the first clock pulse is generated
			400 kHz mode	0.6	—	μs	
106	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
107	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	Note 2
			400 kHz mode	100	—	ns	
92	TSU:STO	STOP condition setup time	100 kHz mode	4.7	—	μs	
			400 kHz mode	0.6	—	μs	
109	TAA	Output valid from clock	100 kHz mode	—	3500	ns	Note 1
			400 kHz mode	—	—	ns	
110	TBUF	Bus free time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
	Cb	Bus capacitive loading	—	400	pF		

- 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.
- Note 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 15-15: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**

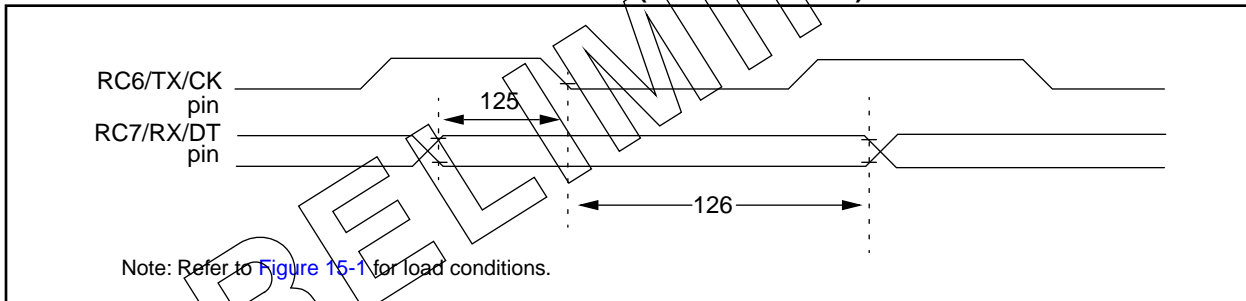


**TABLE 15-11 USART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock high to data out valid	Standard(F)	—	—	80	ns
			Extended(LF)	—	—	100	ns
121	Tckrf	Clock out rise time and fall time (Master Mode)	Standard(F)	—	—	45	ns
			Extended(LF)	—	—	50	ns
122	Tdtrf	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 15-16: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**



**TABLE 15-12 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.

**TABLE 15-13 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	—	—	$< \pm 1$	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	—	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	$V_{DD} - 2.5V$	—	$V_{DD} + 0.3V$	V		
A22	VREF-	Reference voltage low	$V_{SS} - 0.3V$	—	$V_{REF+} - 2.0V$	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)	—	—	1000	—	$\mu A$	During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see <a href="#">Section 10.1</a> .
			—	—	10	—	$\mu A$	During A/D Conversion cycle

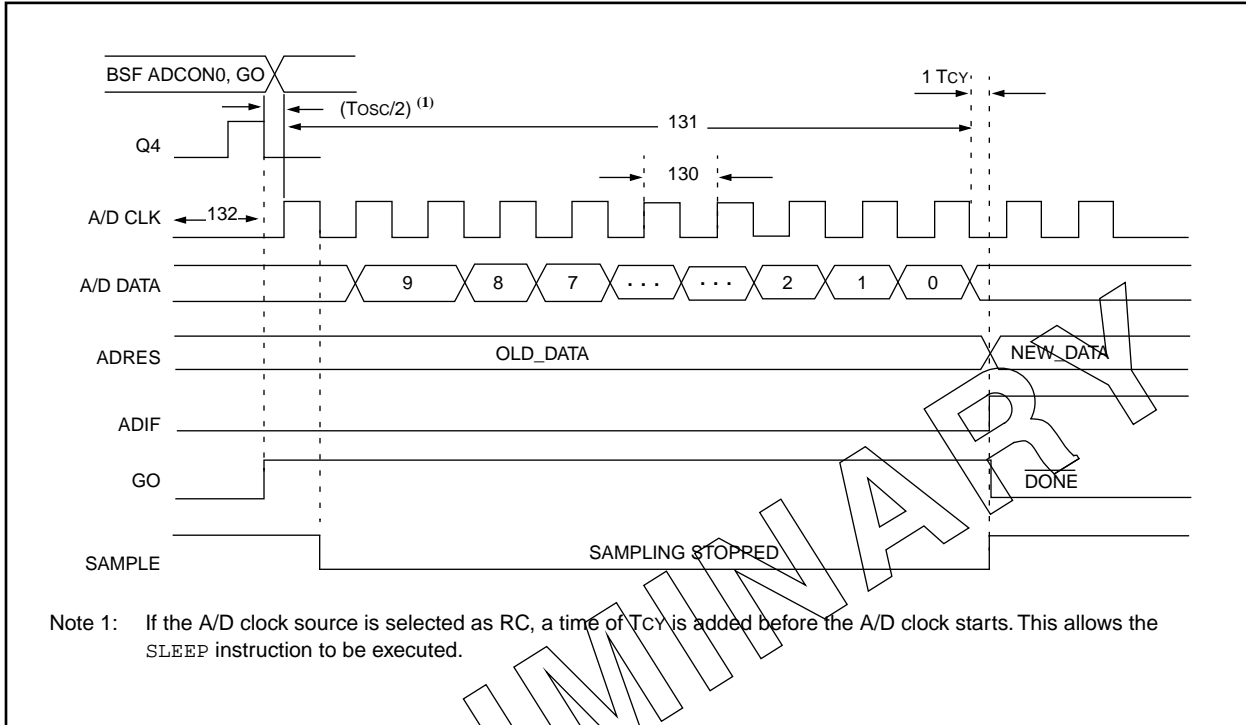
\* These parameters are characterized but not tested.

† 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.

**FIGURE 15-17: A/D CONVERSION TIMING**



**TABLE 15-14 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, $V_{REF}$ full range
			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	$T_{CONV}$	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.

Note 2: See Section 10.1 for min conditions.



## 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 guaranteed 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 + 3 $\sigma$ ) or (mean - 3 $\sigma$ ) respectively, where  $\sigma$  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

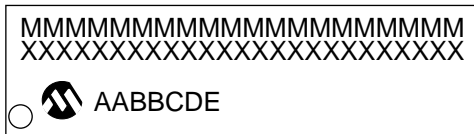
#### 28-Lead PDIP (Skinny DIP)



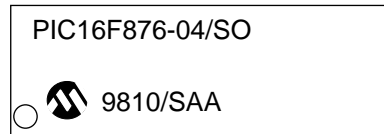
#### Example



#### 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)

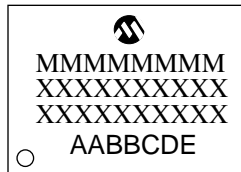
40-Lead PDIP



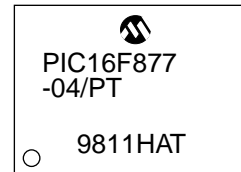
Example



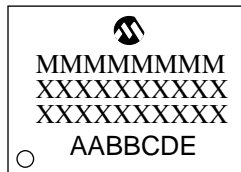
44-Lead TQFP



Example



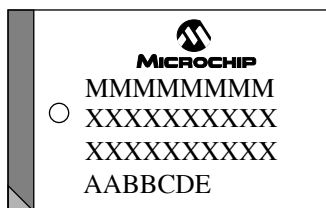
44-Lead MQFP



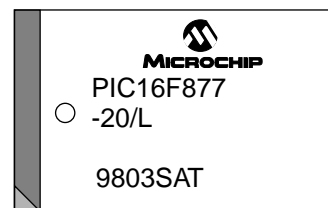
Example



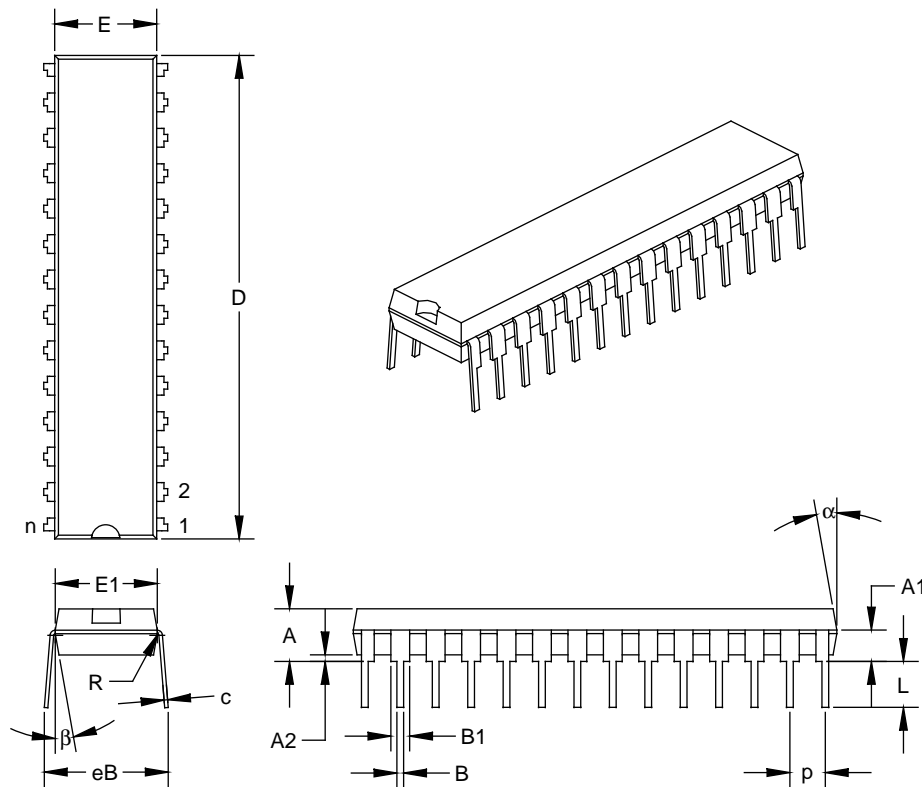
44-Lead PLCC



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			0.300			7.62	
Number of Pins	n		28			28	
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

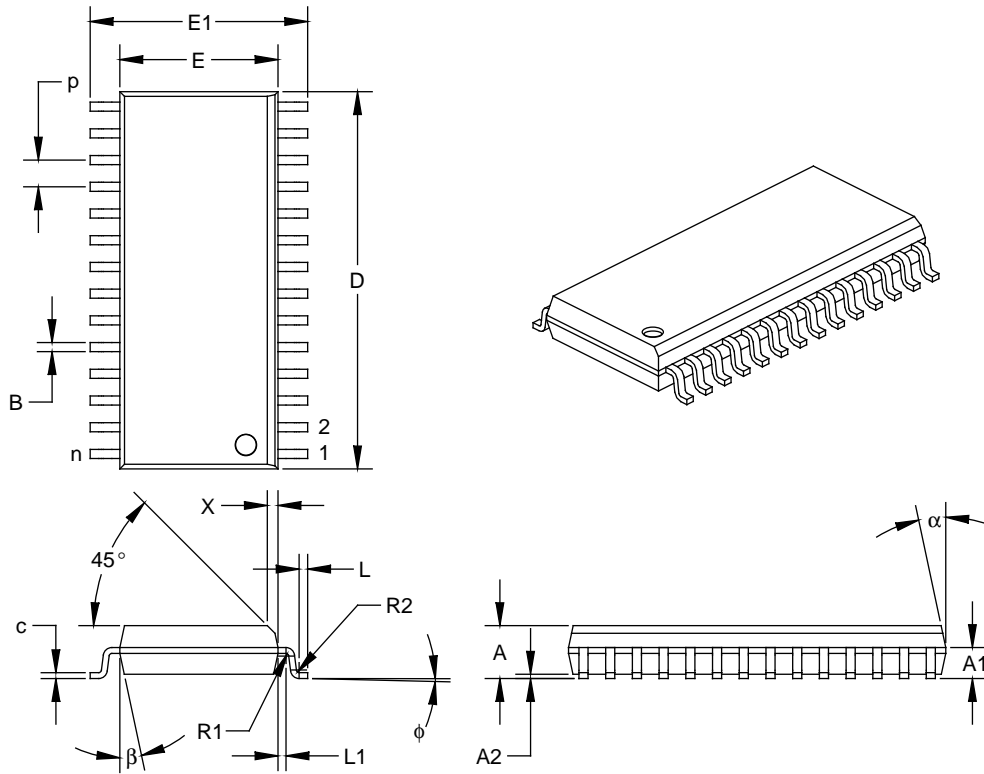
\* Controlling Parameter.

<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



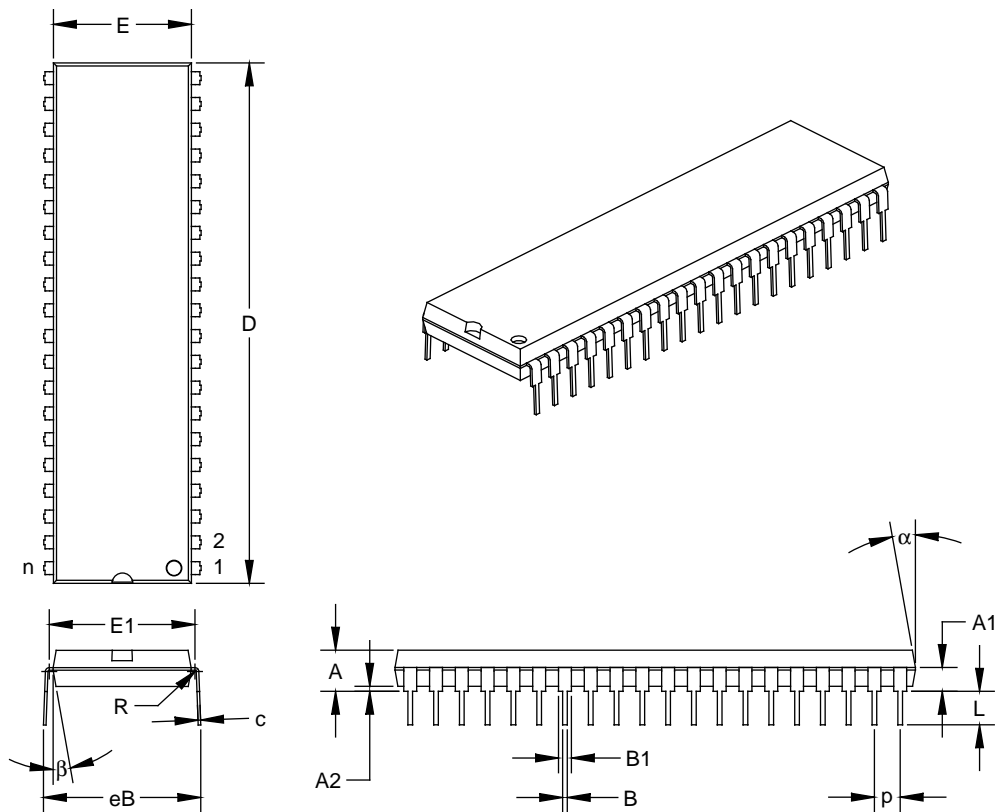
Units		INCHES*			MILLIMETERS		
		MIN	NOM	MAX	MIN	NOM	MAX
Dimension Limits							
Pitch	p		0.050			1.27	
Number of Pins	n		28			28	
Overall Pack. Height	A	0.093	0.099	0.104	2.36	2.50	2.64
Shoulder Height	A1	0.048	0.058	0.068	1.22	1.47	1.73
Standoff	A2	0.004	0.008	0.011	0.10	0.19	0.28
Molded Package Length	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†	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‡	2.013	2.018	2.023	51.13	51.26	51.38
Molded Package Width	E‡	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

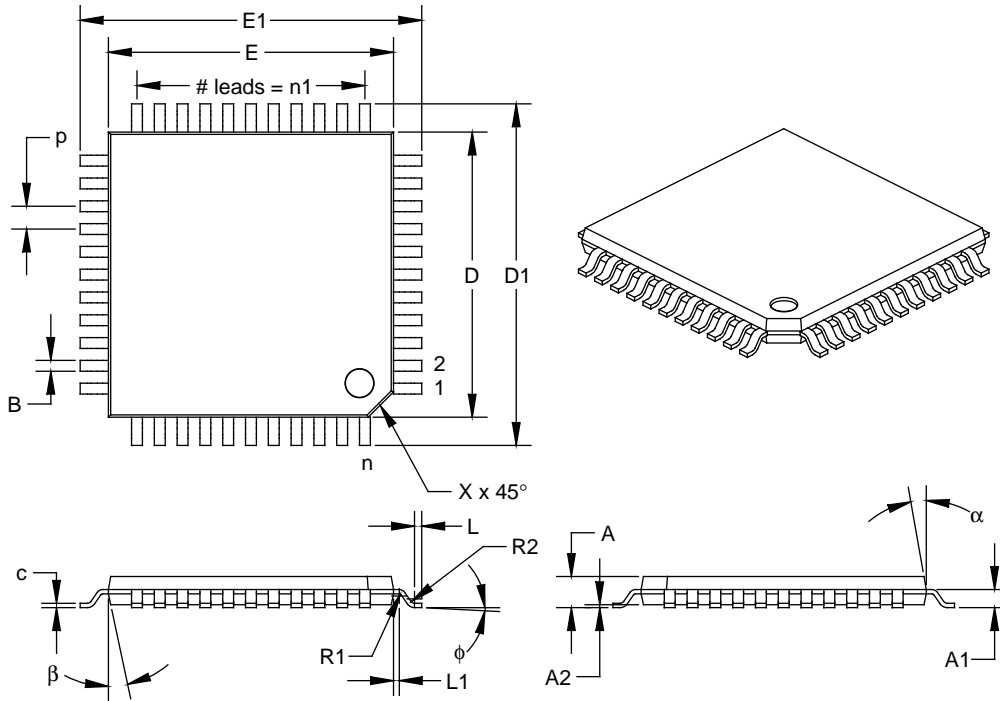
\* Controlling Parameter.

† 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."

# 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†	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‡	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	0.64	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

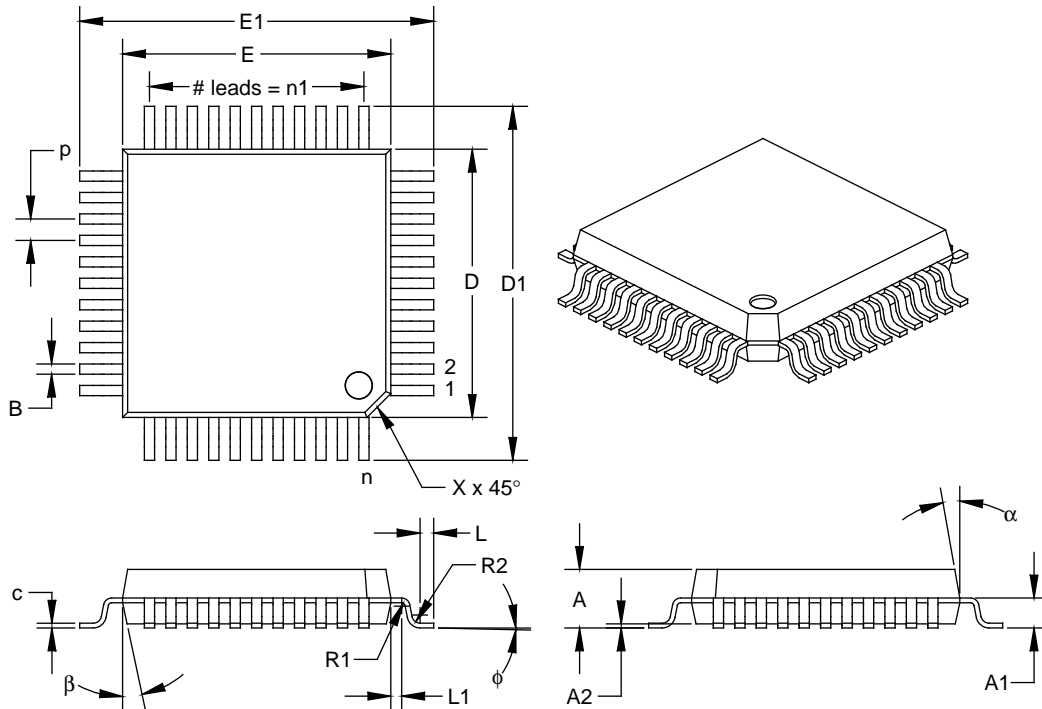
\* Controlling Parameter.

† 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
Dimension Limits							
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.

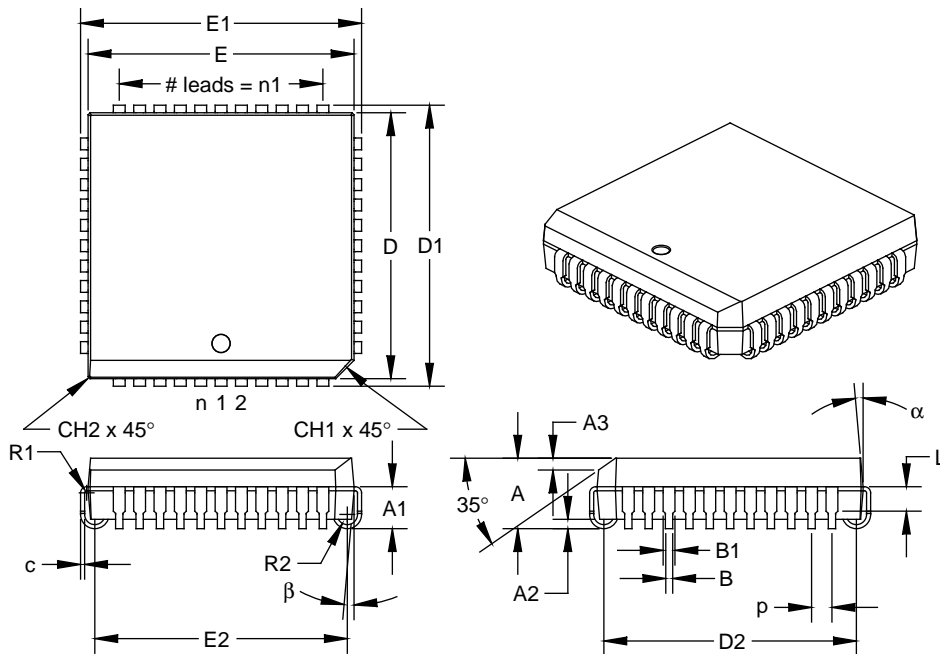
† Dimension "B" does not include dam-bar protrusions. Dam-bar protrusions shall not exceed 0.003" (0.076 mm) per side or 0.006" (0.152 mm) more than dimension "B."

‡ 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	Dimension Limits	INCHES*			MILLIMETERS			
		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	98	This is a new data sheet. However, these devices are similar to the PIC16C7X devices found in the PIC16C7X Data Sheet (DS30390)
B	98	Data Memory Map for PIC16F873/874, moved ADFM bit from ADCON1<5> to ADCON1<7>

## 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 CER-DIP, 28-pin SOIC, 28-pin SSOP	40-pin PDIP, 40-pin windowed CER-DIP, 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:

## A

A/D .....	121
Accuracy/Error .....	130
ADCON0 Register .....	121
ADCON1 Register .....	122
ADIF bit .....	123
Analog Input Model Block Diagram .....	126
Analog Port Pins .....	7, 8, 9, 37, 38
Block Diagram .....	124
Configuring Analog Port Pins .....	127
Configuring the Interrupt .....	123
Configuring the Module .....	123
Connection Considerations .....	131
Conversion Clock .....	127
Conversions .....	128
Delays .....	126
Effects of a Reset .....	130
Equations .....	126
Flowchart of A/D Operation .....	129
GO/DONE bit .....	123
Internal Sampling Switch (R <sub>ss</sub> ) Impedence .....	125
Operation During Sleep .....	129
Sampling Requirements .....	125
Sampling Time .....	126
Source Impedence .....	125
Special Event Trigger (CCP) .....	57
Time Delays .....	126
Transfer Function .....	131
Absolute Maximum Ratings .....	157
ACK .....	72
Acknowledge Data bit, AKD .....	64
Acknowledge Pulse .....	72
Acknowledge Sequence Enable bit, AKE .....	64
Acknowledge Status bit, AKS .....	64
ADRES Register .....	16, 121
AKD .....	64
AKE .....	64
AKS .....	64, 87
Application Note AN578, "Use of the SSP Module in the I <sup>2</sup> C Multi-Master Environment." .....	71
Architecture	
PIC16C63A/PIC16C73B Block Diagram .....	5
PIC16C65B/PIC16C74B Block Diagram .....	6
Assembler	
MPASM Assembler .....	154

## B

Banking, Data Memory .....	12, 19
Baud Rate Generator .....	81
BCLIF .....	25
BF .....	62, 72, 87, 90
Block Diagrams	
A/D .....	124
Analog Input Model .....	126
Baud Rate Generator .....	81
I <sup>2</sup> C Master Mode .....	79
I <sup>2</sup> C Module .....	71
SSP (I <sup>2</sup> C Mode) .....	71
SSP (SPI Mode) .....	65
BRG .....	81
Brown-out Reset (BOR) .....	133, 136, 138, 139, 140
BOR Status (BOR Bit) .....	26
Buffer Full bit, BF .....	72
Buffer Full Status bit, BF .....	62
Bus Arbitration .....	98

## Bus Collision

Section .....	98
Bus Collision During a RESTART Condition .....	101
Bus Collision During a Start Condition .....	99
Bus Collision During a Stop Condition .....	102
Bus Collision Interrupt Flag bit, BCLIF .....	25

## C

Capture (CCP Module) .....	56
Block Diagram .....	56
CCP Pin Configuration .....	56
CCPR1H:CCPR1L Registers .....	56
Changing Between Capture Prescalers .....	56
Software Interrupt .....	56
Timer1 Mode Selection .....	56
Capture/Compare/PWM (CCP) .....	55
CCP1 .....	55
CCP1CON Register .....	55
CCPR1H Register .....	55
CCPR1L Register .....	55
RC2/CCP1 Pin .....	7, 9
CCP2 .....	55
CCP2CON Register .....	55
CCPR2H Register .....	55
CCPR2L Register .....	55
RC1/T1OSI/CCP2 Pin .....	7, 9
Interaction of Two CCP Modules .....	55
Timer Resources .....	55
CCP1CON .....	18
CCP1CON Register .....	55
CCP1M3:CCP1M0 Bits .....	55
CCP1X:CCP1Y Bits .....	55
CCP2CON .....	18
CCP2CON Register .....	55
CCP2M3:CCP2M0 Bits .....	55
CCP2X:CCP2Y Bits .....	55
CCPR1H Register .....	16, 18
CCPR1L Register .....	18
CCPR2H Register .....	16, 18
CCPR2L Register .....	16, 18
CKE .....	62
CKP .....	63
Clock Polarity Select bit, CKP .....	63
Code Examples	
Loading the SSPBUF register .....	66
Code Protection .....	133, 148
Compare (CCP Module) .....	57
Block Diagram .....	57
CCP Pin Configuration .....	57
CCPR1H:CCPR1L Registers .....	57
Software Interrupt .....	57
Special Event Trigger .....	51, 57
Timer1 Mode Selection .....	57
Configuration Bits .....	133
Conversion Considerations .....	187

## D

D/A .....	62
Data Memory .....	12
Bank Select (RP1:RP0 Bits) .....	12, 19
General Purpose Registers .....	12
Register File Map .....	13
Special Function Registers .....	15
Data/Address bit, D/A .....	62

# PIC16F87X

DC Characteristics		
PIC16C76	159	
PIC16C77	159	
Development Support	153	
Development Tools	153	
Device Differences	187	
Direct Addressing	28	
<b>E</b>		
Electrical Characteristics	157	
Errata	4	
External Power-on Reset Circuit	138	
<b>F</b>		
Firmware Instructions	151	
Flowcharts		
Acknowledge	94	
Master Receiver	91	
Master Transmit	88	
Restart Condition	85	
Start Condition	83	
Stop Condition	96	
FSR Register	16, 17, 18	
Fuzzy Logic Dev. System ( <i>fuzzyTECH</i> <sup>®</sup> -MP)	155	
<b>G</b>		
GCE	64	
General Call Address Sequence	77	
General Call Address Support	77	
General Call Enable bit, GCE	64	
<b>I</b>		
I/O Ports	29	
I <sup>2</sup> C	71	
I <sup>2</sup> C Master Mode Receiver Flowchart	91	
I <sup>2</sup> C Master Mode Reception	90	
I <sup>2</sup> C Master Mode Restart Condition	84	
I <sup>2</sup> C Mode Selection	71	
I <sup>2</sup> C Module		
Acknowledge Flowchart	94	
Acknowledge Sequence timing	93	
Addressing	72	
Baud Rate Generator	81	
Block Diagram	79	
BRG Block Diagram	81	
BRG Reset due to SDA Collision	100	
BRG Timing	81	
Bus Arbitration	98	
Bus Collision	98	
Acknowledge	98	
Restart Condition	101	
Restart Condition Timing (Case1)	101	
Restart Condition Timing (Case2)	101	
Start Condition	99	
Start Condition Timing	99, 100	
Stop Condition	102	
Stop Condition Timing (Case1)	102	
Stop Condition Timing (Case2)	102	
Transmit Timing	98	
Bus Collision timing	98	
Clock Arbitration	97	
Clock Arbitration Timing (Master Transmit)	97	
Conditions to not give ACK Pulse	72	
General Call Address Support	77	
Master Mode	79	
Master Mode 7-bit Reception timing	92	
Master Mode Operation	80	
Master Mode Start Condition	82	
Master Mode Transmission	87	
Master Mode Transmit Sequence	80	
Master Transmit Flowchart	88	
Multi-Master Communication	98	
Multi-master Mode	80	
Operation	71	
Repeat Start Condition timing	84	
Restart Condition Flowchart	85	
Slave Mode	72	
Slave Reception	73	
Slave Transmission	73	
SSPBUF	72	
Start Condition Flowchart	83	
Stop Condition Flowchart	96	
Stop Condition Receive or Transmit timing	95	
Stop Condition timing	95	
Waveforms for 7-bit Reception	73	
Waveforms for 7-bit Transmission	74	
I <sup>2</sup> C Module Address Register, SSPADD	72	
I <sup>2</sup> C Slave Mode	72	
ICEPIC Low-Cost PIC16CXXX In-Circuit Emulator	153	
ID Locations	133, 148	
In-Circuit Serial Programming (ICSP)	133, 148	
INDF	18	
INDF Register	16, 17	
Indirect Addressing	28	
FSR Register	12	
Instruction Format	151	
Instruction Set	151	
Summary Table	152	
INTCON	18	
INTCON Register	21	
GIE Bit	21	
INTE Bit	21	
INTF Bit	21	
PEIE Bit	21	
RBIE Bit	21	
RBIF Bit	21, 31	
TOIE Bit	21	
TOIF Bit	21	
Inter-Integrated Circuit (I <sup>2</sup> C)	61	
Internal Sampling Switch (R <sub>ss</sub> ) Impedance	125	
Interrupt Sources	133, 143	
Block Diagram	144	
Capture Complete (CCP)	56	
Compare Complete (CCP)	57	
Interrupt on Change (RB7:RB4)	31	
RB0/INT Pin, External	7, 8, 145	
TMR0 Overflow	48, 145	
TMR1 Overflow	49, 51	
TMR2 to PR2 Match	54	
TMR2 to PR2 Match (PWM)	53, 58	
USART Receive/Transmit Complete	105	
Interrupts		
Bus Collision Interrupt	25	
Synchronous Serial Port Interrupt	23	
Interrupts, Context Saving During	145	
Interrupts, Enable Bits		
CCP1 Enable (CCP1IE Bit)	56	
Global Interrupt Enable (GIE Bit)	21, 143	
Interrupt on Change (RB7:RB4) Enable (RBIE Bit)	21, 145	
Peripheral Interrupt Enable (PEIE Bit)	21	
RB0/INT Enable (INTE Bit)	21	
TMR0 Overflow Enable (TOIE Bit)	21	

Interrupts, Flag Bits		Pinout Descriptions	
CCP1 Flag (CCP1IF Bit)	56, 57	PIC16C63A/PIC16C73B	7
Interrupt on Change (RB7:RB4) Flag		PIC16C65B/PIC16C74B	8
(RBIF Bit)	21, 31, 145	PIR1 Register	23
RB0/INT Flag (INTF Bit)	21	PIR2 Register	25
TMR0 Overflow Flag (T0IF Bit)	21, 145	Pointer, FSR	27
<b>K</b>		PORTA	7, 8, 18
KeeLoq® Evaluation and Programming Tools	155	Analog Port Pins	7, 8
<b>M</b>		Initialization	29
Master Clear ( $\overline{\text{MCLR}}$ )	7, 8	PORTA Register	29
$\overline{\text{MCLR}}$ Reset, Normal Operation	136, 139, 140	RA3:RA0 and RA5 Port Pins	29
$\overline{\text{MCLR}}$ Reset, SLEEP	136, 139, 140	RA4/T0CKI Pin	7, 8, 29
Memory Organization		RA5/ $\overline{\text{SS}}$ /AN4 Pin	7, 8
Data Memory	12	TRISA Register	29
Program Memory	11	PORTA Register	16
MPLAB Integrated Development Environment Software	154	PORTB	7, 8, 18
Multi-Master Communication	98	Initialization	31
Multi-Master Mode	80	PORTB Register	31
<b>O</b>		Pull-up Enable ( $\overline{\text{RBP}}\overline{\text{U}}$ Bit)	20
OPCODE Field Descriptions	151	RB0/INT Edge Select (INTEDG Bit)	20
OPTION	18	RB0/INT Pin, External	7, 8, 145
OPTION_REG Register	20	RB3:RB0 Port Pins	31
INTEDG Bit	20	RB7:RB4 Interrupt on Change	145
PS2:PS0 Bits	20, 47	RB7:RB4 Interrupt on Change Enable	
PSA Bit	20, 47	(RBIE Bit)	21, 145
$\overline{\text{RBP}}\overline{\text{U}}$ Bit	20	RB7:RB4 Interrupt on Change Flag	
T0CS Bit	20, 47	(RBIF Bit)	21, 31, 145
T0SE Bit	20, 47	RB7:RB4 Port Pins	31
OSC1/CLKIN Pin	7, 8	TRISB Register	31
OSC2/CLKOUT Pin	7, 8	PORTB Register	16
Oscillator Configuration	133, 134	PORTC	7, 9, 18
HS	134, 139	Block Diagram	33
LP	134, 139	Initialization	33
RC	134, 135, 139	PORTC Register	33
XT	134, 139	RC0/T1OSO/T1CKI Pin	7, 9
Oscillator, Timer1	49, 51	RC1/T1OSI/CCP2 Pin	7, 9
Oscillator, WDT	146	RC2/CCP1 Pin	7, 9
<b>P</b>		RC3/SCK/SCL Pin	7, 9
P	62	RC4/SDI/SDA Pin	7, 9
Packaging	179	RC5/SDO Pin	7, 9
Paging, Program Memory	11, 27	RC6/TX/CK Pin	7, 9, 106
Parallel Slave Port (PSP)	9, 35, 38	RC7/RX/DT Pin	7, 9, 106, 107
Block Diagram	38	TRISC Register	33, 105
RE0/ $\overline{\text{RD}}$ /AN5 Pin	9, 37, 38	PORTC Register	16
RE1/ $\overline{\text{WR}}$ /AN6 Pin	9, 37, 38	PORTD	9, 18, 38
RE2/ $\overline{\text{CS}}$ /AN7 Pin	9, 37, 38	Block Diagram	35
Read Waveforms	39	Parallel Slave Port (PSP) Function	35
Select (PSPMODE Bit)	35, 36, 38	PORTD Register	35
Write Waveforms	38	TRISD Register	35
PCL Register	16, 17, 18	PORTD Register	16
PCLATH Register	16, 17, 18	PORTE	9, 18
PCON Register	18, 26, 139	Analog Port Pins	9, 37, 38
BOR Bit	26	Block Diagram	36
$\overline{\text{POR}}$ Bit	26	Input Buffer Full Status (IBF Bit)	36
PICDEM-1 Low-Cost PICmicro Demo Board	154	Input Buffer Overflow (IBOV Bit)	36
PICDEM-2 Low-Cost PIC16CXX Demo Board	154	Output Buffer Full Status (OBF Bit)	36
PICDEM-3 Low-Cost PIC16CXXX Demo Board	154	PORTE Register	36
PICSTART® Plus Entry Level Development System	153	PSP Mode Select (PSPMODE Bit)	35, 36, 38
PIE1 Register	18, 22	RE0/ $\overline{\text{RD}}$ /AN5 Pin	9, 37, 38
PIE2 Register	18, 24	RE1/ $\overline{\text{WR}}$ /AN6 Pin	9, 37, 38
		RE2/ $\overline{\text{CS}}$ /AN7 Pin	9, 37, 38
		TRISE Register	36
		PORTE Register	16

# PIC16F87X

Postscaler, Timer2			
Select (TOUTPS3:TOUTPS0 Bits)	53		
Postscaler, WDT	47		
Assignment (PSA Bit)	20, 47		
Block Diagram	48		
Rate Select (PS2:PS0 Bits)	20, 47		
Switching Between Timer0 and WDT	48		
Power-on Reset (POR)	133, 136, 138, 139, 140		
Oscillator Start-up Timer (OST)	133, 138		
POR Status ( $\overline{\text{POR}}$ Bit)	26		
Power Control (PCON) Register	139		
Power-down ( $\overline{\text{PD}}$ Bit)	19, 136		
Power-on Reset Circuit, External	138		
Power-up Timer (PWRT)	133, 138		
Time-out ( $\overline{\text{TO}}$ Bit)	19, 136		
Time-out Sequence	139		
Time-out Sequence on Power-up	141, 142		
PR2	18		
PR2 Register	17		
Prescaler, Capture	56		
Prescaler, Timer0	47		
Assignment (PSA Bit)	20, 47		
Block Diagram	48		
Rate Select (PS2:PS0 Bits)	20, 47		
Switching Between Timer0 and WDT	48		
Prescaler, Timer1	50		
Select (T1CKPS1:T1CKPS0 Bits)	49		
Prescaler, Timer2	58		
Select (T2CKPS1:T2CKPS0 Bits)	53		
PRO MATE® II Universal Programmer	153		
Product Identification System	199		
Program Counter			
PCL Register	27		
PCLATH Register	27, 145		
Reset Conditions	139		
Program Memory	11		
Interrupt Vector	11		
Paging	11, 27		
Program Memory Map	11		
Reset Vector	11		
Program Verification	148		
Programming Pin (Vpp)	7, 8		
Programming, Device Instructions	151		
PWM (CCP Module)	58		
Block Diagram	58		
CCPR1H:CCPR1L Registers	58		
Duty Cycle	58		
Example Frequencies/Resolutions	59		
Output Diagram	58		
Period	58		
Set-Up for PWM Operation	59		
TMR2 to PR2 Match	53, 58		
<b>Q</b>			
Q-Clock	58		
<b>R</b>			
R/ $\overline{\text{W}}$	62		
R/W bit	72		
R/ $\overline{\text{W}}$ bit	73		
RCE, Receive Enable bit, RCE	64		
RCREG	18		
RCSTA Register	18, 106		
CREN Bit	106		
FERR Bit	106		
OERR Bit	106		
RX9 Bit	106		
RX9D Bit	106		
SPEN Bit	105, 106		
SREN Bit	106		
Read/Write bit, R/ $\overline{\text{W}}$	62		
Receive Overflow Indicator bit, SSPOV	63		
Register File	12		
Register File Map	13		
Registers			
FSR			
Summary	18		
INDF			
Summary	18		
INTCON			
Summary	18		
OPTION			
Summary	18		
PCL			
Summary	18		
PCLATH			
Summary	18		
PORTB			
Summary	18		
SSPSTAT	62		
STATUS			
Summary	18		
Summary	16		
TMR0			
Summary	18		
TRISB			
Summary	18		
Reset	133, 136		
Block Diagram	137		
Reset Conditions for All Registers	140		
Reset Conditions for PCON Register	139		
Reset Conditions for Program Counter	139		
Reset Conditions for STATUS Register	139		
WDT Reset. See Watchdog Timer (WDT)			
Restart Condition Enabled bit, RSE	64		
Revision History	187		
RSE	64		
<b>S</b>			
S	62		
SAE	64		
SCK	65		
SCL	72		
SDA	72		
SDI	65		
SDO	65		
SEEVAL® Evaluation and Programming System	155		
Serial Clock, SCK	65		
Serial Clock, SCL	72		
Serial Data Address, SDA	72		
Serial Data In, SDI	65		
Serial Data Out, SDO	65		
Slave Select Synchronization	68		
Slave Select, SS	65		
SLEEP	133, 136, 147		
SMP	62		
Software Simulator (MPLAB-SIM)	155		
SPBRG	18		



SPBRG Register .....	17	Stack .....	27
SPE .....	64	Start bit (S) .....	62
Special Features of the CPU .....	133	Start Condition Enabled bit, SAE .....	64
Special Function Registers .....	15	STATUS Register .....	18, 19, 145
PIC16C73 .....	16	C Bit .....	19
PIC16C73A .....	16	DC Bit .....	19
PIC16C74 .....	16	IRP Bit .....	19
PIC16C74A .....	16	PD Bit .....	19, 136
PIC16C76 .....	16	RP1:RP0 Bits .....	19
PIC16C77 .....	16	TO Bit .....	19, 136
Speed, Operating .....	1	Z Bit .....	19
SPI		Stop bit (P) .....	62
Master Mode .....	67	Stop Condition Enable bit .....	64
Serial Clock .....	65	Synchronous Serial Port .....	61
Serial Data In .....	65	Synchronous Serial Port Enable bit, SSPEN .....	63
Serial Data Out .....	65	Synchronous Serial Port Interrupt .....	23
Serial Peripheral Interface (SPI) .....	61	Synchronous Serial Port Mode Select bits, SSPM3:SSPM0 .....	63
Slave Select .....	65		
SPI clock .....	67	<b>T</b>	
SPI Mode .....	65	T1CON .....	18
SPI Clock Edge Select, CKE .....	62	T1CON Register .....	18, 49
SPI Data Input Sample Phase Select, SMP .....	62	T1CKPS1:T1CKPS0 Bits .....	49
SPI Master/Slave Connection .....	66	T1OSCEN Bit .....	49
SPI Module		T1SYNC Bit .....	49
Master/Slave Connection .....	66	TMR1CS Bit .....	49
Slave Mode .....	68	TMR1ON Bit .....	49
Slave Select Synchronization .....	68	T2CON Register .....	18, 53
Slave Synch Timnig .....	68	T2CKPS1:T2CKPS0 Bits .....	53
SS .....	65	TMR2ON Bit .....	53
SSP .....	61	TOUTPS3:TOUTPS0 Bits .....	53
Block Diagram (SPI Mode) .....	65	TAD .....	127
RA5/SS/AN4 Pin .....	7, 8	Timer0 .....	47
RC3/SCK/SCL Pin .....	7, 9	Block Diagram .....	47
RC4/SDI/SDA Pin .....	7, 9	Clock Source Edge Select (T0SE Bit) .....	20, 47
RC5/SDO Pin .....	7, 9	Clock Source Select (T0CS Bit) .....	20, 47
SPI Mode .....	65	Overflow Enable (T0IE Bit) .....	21
SSPADD .....	72	Overflow Flag (T0IF Bit) .....	21, 145
SSPBUF .....	67, 72	Overflow Interrupt .....	48, 145
SSPCON1 .....	63	RA4/T0CKI Pin, External Clock .....	7, 8
SSPCON2 .....	64	Timer1 .....	49
SSPSR .....	67, 72	Block Diagram .....	50
SSPSTAT .....	62, 72	Capacitor Selection .....	51
TMR2 Output for Clock Shift .....	53, 54	Clock Source Select (TMR1CS Bit) .....	49
SSP I <sup>2</sup> C		External Clock Input Sync (T1SYNC Bit) .....	49
SSP I <sup>2</sup> C Operation .....	71	Module On/Off (TMR1ON Bit) .....	49
SSP Module		Oscillator .....	49, 51
SPI Master Mode .....	67	Oscillator Enable (T1OSCEN Bit) .....	49
SPI Master./Slave Connection .....	66	Overflow Interrupt .....	49, 51
SPI Slave Mode .....	68	RC0/T1OSO/T1CKI Pin .....	7, 9
SSPCON1 Register .....	71	RC1/T1OSI/CCP2 Pin .....	7, 9
SSP Overflow Detect bit, SSPOV .....	72	Special Event Trigger (CCP) .....	51, 57
SSPADD Register .....	17, 18	T1CON Register .....	49
SSPBUF .....	18, 72	TMR1H Register .....	49
SSPBUF Register .....	16	TMR1L Register .....	49
SSPCON Register .....	16	Timer2	
SSPCON1 .....	63, 71	Block Diagram .....	54
SSPCON2 .....	64	PR2 Register .....	53, 58
SSPEN .....	63	SSP Clock Shift .....	53, 54
SSPIF .....	23, 73	T2CON Register .....	53
SSPM3:SSPM0 .....	63	TMR2 Register .....	53
SSPOV .....	63, 72, 90	TMR2 to PR2 Match Interrupt .....	53, 54, 58
SSPSTAT .....	62, 72		
SSPSTAT Register .....	17, 18		

# PIC16F87X

Timing Diagrams		
A/D Conversion .....	176	
Acknowledge Sequence Timing .....	93	
Baud Rate Generator with Clock Arbitration .....	81	
BRG Reset Due to SDA Collision .....	100	
Brown-out Reset .....	166	
Bus Collision		
Start Condition Timing .....	99	
Bus Collision During a Restart Condition (Case 1) ..	101	
Bus Collision During a Restart Condition (Case2) ...	101	
Bus Collision During a Start Condition (SCL = 0) ....	100	
Bus Collision During a Stop Condition .....	102	
Bus Collision for Transmit and Acknowledge .....	98	
Capture/Compare/PWM .....	168	
CLKOUT and I/O .....	165	
I <sup>2</sup> C Bus Data .....	173	
I <sup>2</sup> C Bus Start/Stop bits .....	172	
I <sup>2</sup> C Master Mode First Start bit timing .....	82	
I <sup>2</sup> C Master Mode Reception timing .....	92	
I <sup>2</sup> C Master Mode Transmission timing .....	89	
Master Mode Transmit Clock Arbitration .....	97	
Power-up Timer .....	166	
Repeat Start Condition .....	84	
Reset .....	166	
Slave Synchronization .....	68	
Start-up Timer .....	166	
Stop Condition Receive or Transmit .....	95	
Time-out Sequence on Power-up .....	141, 142	
Timer0 .....	167	
Timer1 .....	167	
USART Asynchronous Master Transmission .....	111	
USART Synchronous Receive .....	174	
USART Synchronous Reception .....	117	
USART Synchronous Transmission .....	116, 174	
USART, Asynchronous Reception .....	113	
Wake-up from SLEEP via Interrupt .....	148	
Watchdog Timer .....	166	
TMR0 .....	18	
TMR0 Register .....	16	
TMR1H .....	18	
TMR1H Register .....	16	
TMR1L .....	18	
TMR1L Register .....	16	
TMR2 .....	18	
TMR2 Register .....	16	
TRISA .....	18	
TRISA Register .....	17	
TRISB .....	18	
TRISB Register .....	17	
TRISC .....	18	
TRISC Register .....	17	
TRISD .....	18	
TRISD Register .....	17	
TRISE .....	18	
TRISE Register .....	17, 36	
IBF Bit .....	36	
IBOV Bit .....	36	
OBF Bit .....	36	
PSPMODE Bit .....	35, 36, 38	
TXREG .....	18	
TXSTA .....	18	
TXSTA Register .....	105	
BRGH Bit .....	105, 107	
CSRC Bit .....	105	
SYNC Bit .....	105	
TRMT Bit .....	105	
TX9 Bit .....	105	
TX9D Bit .....	105	
TXEN Bit .....	105	
<b>U</b>		
UA .....	62	
Universal Synchronous Asynchronous Receiver Transmitter (USART)		
Asynchronous Receiver		
Setting Up Reception .....	112	
Timing Diagram .....	113	
Update Address, UA .....	62	
USART .....	105	
Asynchronous Mode .....	110	
Master Transmission .....	111	
Receive Block Diagram .....	113	
Transmit Block Diagram .....	110	
Baud Rate Generator (BRG) .....	107	
Baud Rate Error, Calculating .....	107	
Baud Rate Formula .....	107	
Baud Rates, Asynchronous Mode (BRGH=0) .....	108	
Baud Rates, Asynchronous Mode (BRGH=1) .....	109	
Baud Rates, Synchronous Mode .....	108	
High Baud Rate Select (BRGH Bit) .....	105, 107	
Sampling .....	107	
Clock Source Select (CSRC Bit) .....	105	
Continuous Receive Enable (CREN Bit) .....	106	
Framing Error (FERR Bit) .....	106	
Mode Select (SYNC Bit) .....	105	
Overrun Error (OERR Bit) .....	106	
RC6/TX/CK Pin .....	7, 9	
RC7/RX/DT Pin .....	7, 9	
RCSTA Register .....	106	
Receive Data, 9th bit (RX9D Bit) .....	106	
Receive Enable, 9-bit (RX9 Bit) .....	106	
Serial Port Enable (SPEN Bit) .....	105, 106	
Single Receive Enable (SREN Bit) .....	106	
Synchronous Master Mode .....	115	
Reception .....	117	
Transmission .....	116	
Synchronous Slave Mode .....	118	
Transmit Data, 9th Bit (TX9D) .....	105	
Transmit Enable (TXEN Bit) .....	105	
Transmit Enable, Nine-bit (TX9 Bit) .....	105	
Transmit Shift Register Status (TRMT Bit) .....	105	
TXSTA Register .....	105	

## W

W Register .....	145
Wake-up from SLEEP .....	133, 147
Interrupts .....	139, 140
MCLR Reset .....	140
Timing Diagram .....	148
WDT Reset .....	140
Watchdog Timer (WDT) .....	133, 146
Block Diagram .....	146
Enable (WDTE Bit) .....	146
Programming Considerations .....	146
RC Oscillator .....	146
Time-out Period .....	146
WDT Reset, Normal Operation .....	136, 139, 140
WDT Reset, SLEEP .....	136, 139, 140
Waveform for General Call Address Sequence .....	77
WCOL .....	63, 82, 87, 90, 93, 95
WCOL Status Flag .....	82
Write Collision Detect bit, WCOL .....	63
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

1-602-786-7302 for the rest of the world.

981103

**Trademarks:** The Microchip name, logo, PIC, PICmicro, PICSTART, PICMASTER and PRO MATE are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. *FlexROM*, *MPLAB* and *fuzzy-LAB* 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 (602) 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: **DS30292A**

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 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:**

- g) PIC16F877 -20/P 301 = Commercial temp., PDIP package, 4 MHz, normal VDD limits, QTP pattern #301.
- h) PIC16F876 - 04I/SO = Industrial temp., SOIC package, 200 kHz, Extended VDD limits.
- i) PIC16F877 - 04I/P = Industrial temp., PDIP package, 10MHz, normal VDD limits.

**Note 1:** C = CMOS  
LC = Low Power CMOS  
T = in tape and reel - SOIC, PLCC, MQFP, TQFP packages only.

**2:** b = blank

\* 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).



**MICROCHIP**

## WORLDWIDE SALES AND SERVICE

### AMERICAS

#### Corporate Office

Microchip Technology Inc.  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 602-786-7200 Fax: 602-786-7277  
Technical Support: 602 786-7627  
Web: <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.  
14651 Dallas Parkway, Suite 816  
Dallas, TX 75240-8809  
Tel: 972-991-7177 Fax: 972-991-8588

#### Dayton

Microchip Technology Inc.  
Two Prestige Place, Suite 150  
Miamisburg, OH 45342  
Tel: 937-291-1654 Fax: 937-291-9175

#### Detroit

Microchip Technology Inc.  
42705 Grand River, Suite 201  
Novi, MI 48375-1727  
Tel: 248-374-1888 Fax: 248-374-2874

#### Los Angeles

Microchip Technology Inc.  
18201 Von Karman, Suite 1090  
Irvine, CA 92612  
Tel: 714-263-1888 Fax: 714-263-1338

#### New York

Microchip Technology Inc.  
150 Motor Parkway, Suite 202  
Hauppauge, NY 11788  
Tel: 516-273-5305 Fax: 516-273-5335

#### 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  
RM 3801B, Tower Two  
Metroplaza  
223 Hing Fong Road  
Kwai Fong, N.T., Hong Kong  
Tel: 852-2-401-1200 Fax: 852-2-401-3431

#### 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  
Winnersh Triangle  
Wokingham  
Berkshire, England RG41 5TU  
Tel: 44-1189-21-5858 Fax: 44-1189-21-5835

#### France

Arizona Microchip Technology SARM  
Zone Industrielle de la Bonde  
2 Rue du Buisson aux Fraises  
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: 39-39-6899939 Fax: 39-39-6899883

10/27/98



*Microchip received ISO 9001 Quality System certification for its worldwide headquarters, design, and wafer fabrication facilities in January, 1997. Our field-programmable PICmicro® 8-bit MCUs, KEELoc® code hopping devices, Serial EEPROMs, related specialty memory products and development systems conform to the stringent quality standards of the International Standard Organization (ISO).*

All rights reserved. © 1998 Microchip Technology Incorporated. Printed in the USA. 12/98 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.