



PIC16F87XA

Data Sheet

28/40/44-Pin Enhanced Flash
Microcontrollers

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. 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.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, MPLAB, PIC, PICmicro, PICSTART, PRO MATE and PowerSmart are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


AmpLab, FilterLab, microID, MXDEV, MXLAB, PICMASTER, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

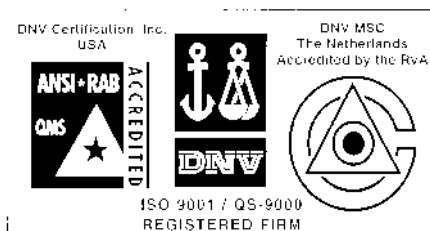
Application Maestro, dsPICDEM, dsPICDEM.net, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICKit, PICDEM, PICDEM.net, PowerCal, PowerInfo, PowerMate, PowerTool, rLAB, rfPIC, Select Mode, SmartSensor, SmartShunt, SmartTel and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2003, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.



PIC16F87XA

28/40/44-Pin Enhanced Flash Microcontrollers

Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

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 other 28-pin or 40/44-pin
PIC16CXXX and PIC16FXXX microcontrollers

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 postscale
- 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
- Synchronous Serial Port (SSP) with SPI™
(Master mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) – 8 bits wide with
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)

Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital
Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
 - Two analog comparators
 - Programmable on-chip voltage reference
(VREF) module
 - Programmable input multiplexing from device
inputs and internal voltage reference
 - Comparator outputs are externally accessible

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash
program memory typical
- 1,000,000 erase/write cycle Data EEPROM
memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™)
via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

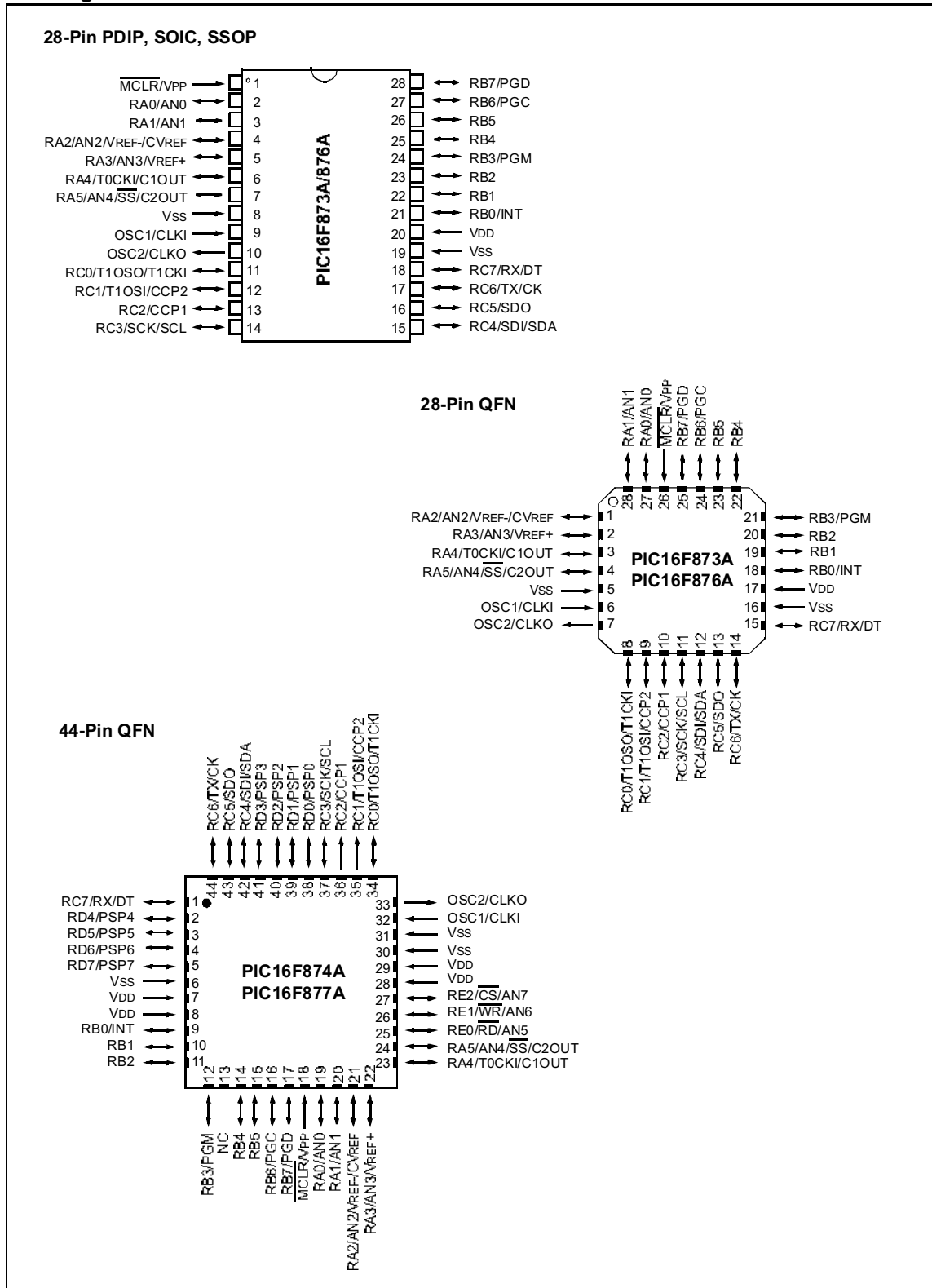
CMOS Technology:

- Low-power, high-speed Flash/EEPROM
technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I ² C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

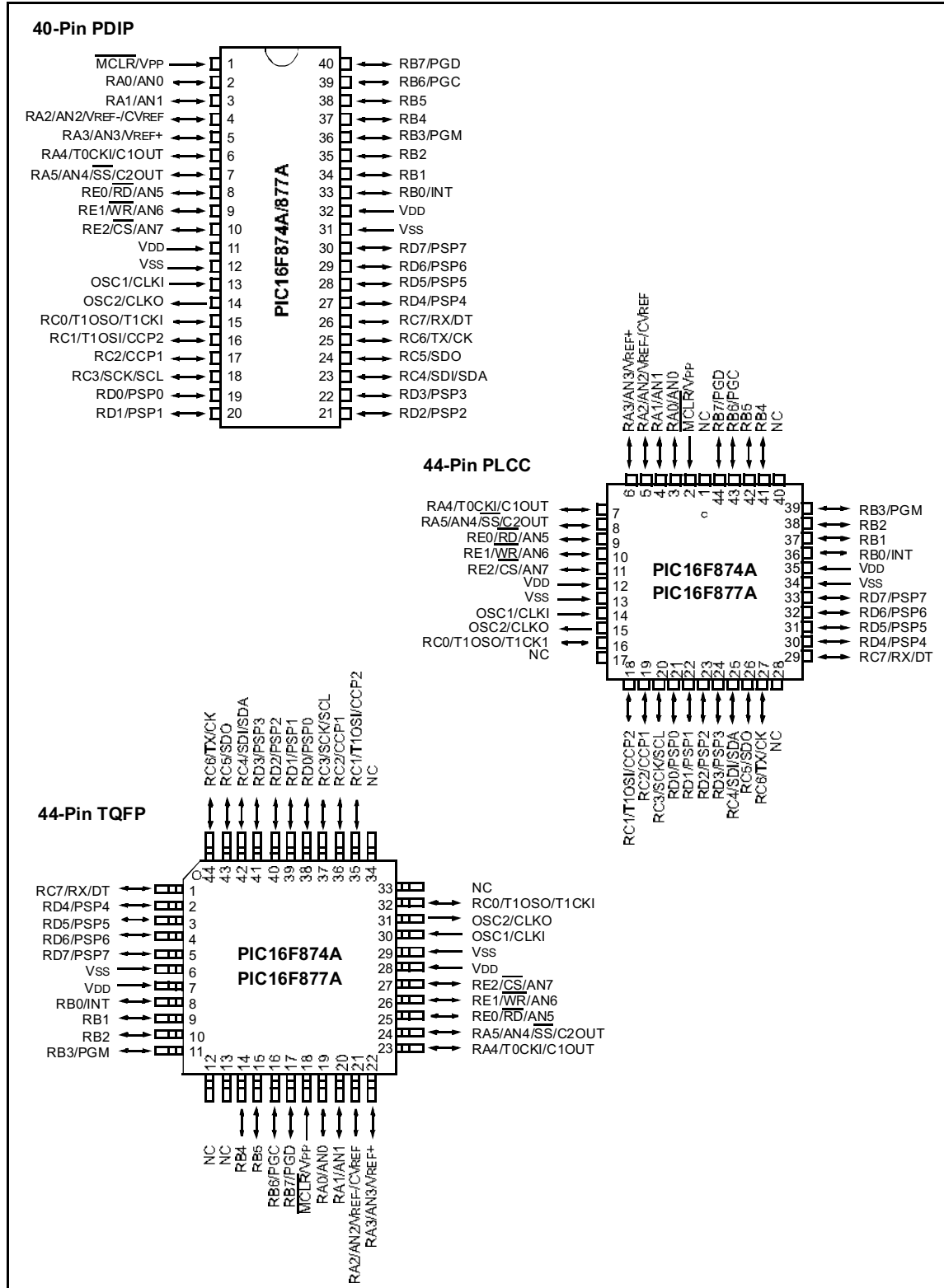
PIC16F87XA

Pin Diagrams



PIC16F87XA

Pin Diagrams (Continued)



PIC16F87XA

Table of Contents

1.0	Device Overview.....	5
2.0	Memory Organization.....	15
3.0	Data EEPROM and Flash Program Memory	33
4.0	I/O Ports.....	41
5.0	Timer0 Module.....	53
6.0	Timer1 Module.....	57
7.0	Timer2 Module.....	61
8.0	Capture/Compare/PWM Modules.....	63
9.0	Master Synchronous Serial Port (MSSP) Module.....	71
10.0	Addressable Universal Synchronous Asynchronous Receiver Transmitter (USART)	111
11.0	Analog-to-Digital Converter (A/D) Module	127
12.0	Comparator Module	135
13.0	Comparator Voltage Reference Module	141
14.0	Special Features of the CPU	143
15.0	Instruction Set Summary.....	159
16.0	Development Support.....	167
17.0	Electrical Characteristics.....	173
18.0	DC and AC Characteristics Graphs and Tables	197
19.0	Packaging Information	209
	Appendix A: Revision History.....	219
	Appendix B: Device Differences.....	219
	Appendix C: Conversion Considerations.....	220
	Index	221
	On-Line Support.....	229
	Systems Information and Upgrade Hot Line	229
	Reader Response	230
	PIC16F87XA Product Identification System.....	231

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at docerrors@mail.microchip.com or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

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

<http://www.microchip.com>

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

Errata

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

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

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

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.

Customer Notification System

Register on our Web site at www.microchip.com/cn to receive the most current information on all of our products.

PIC16F87XA

1.0 DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

PIC16F873A/876A devices are available only in 28-pin packages, while PIC16F874A/877A devices are available in 40-pin and 44-pin packages. All devices in the PIC16F87XA family share common architecture with the following differences:

- The PIC16F873A and PIC16F874A have one-half of the total on-chip memory of the PIC16F876A and PIC16F877A
- The 28-pin devices have three I/O ports, while the 40/44-pin devices have five
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight
- The Parallel Slave Port is implemented only on the 40/44-pin devices

The available features are summarized in Table 1-1. Block diagrams of the PIC16F873A/876A and PIC16F874A/877A devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2 and Table 1-3.

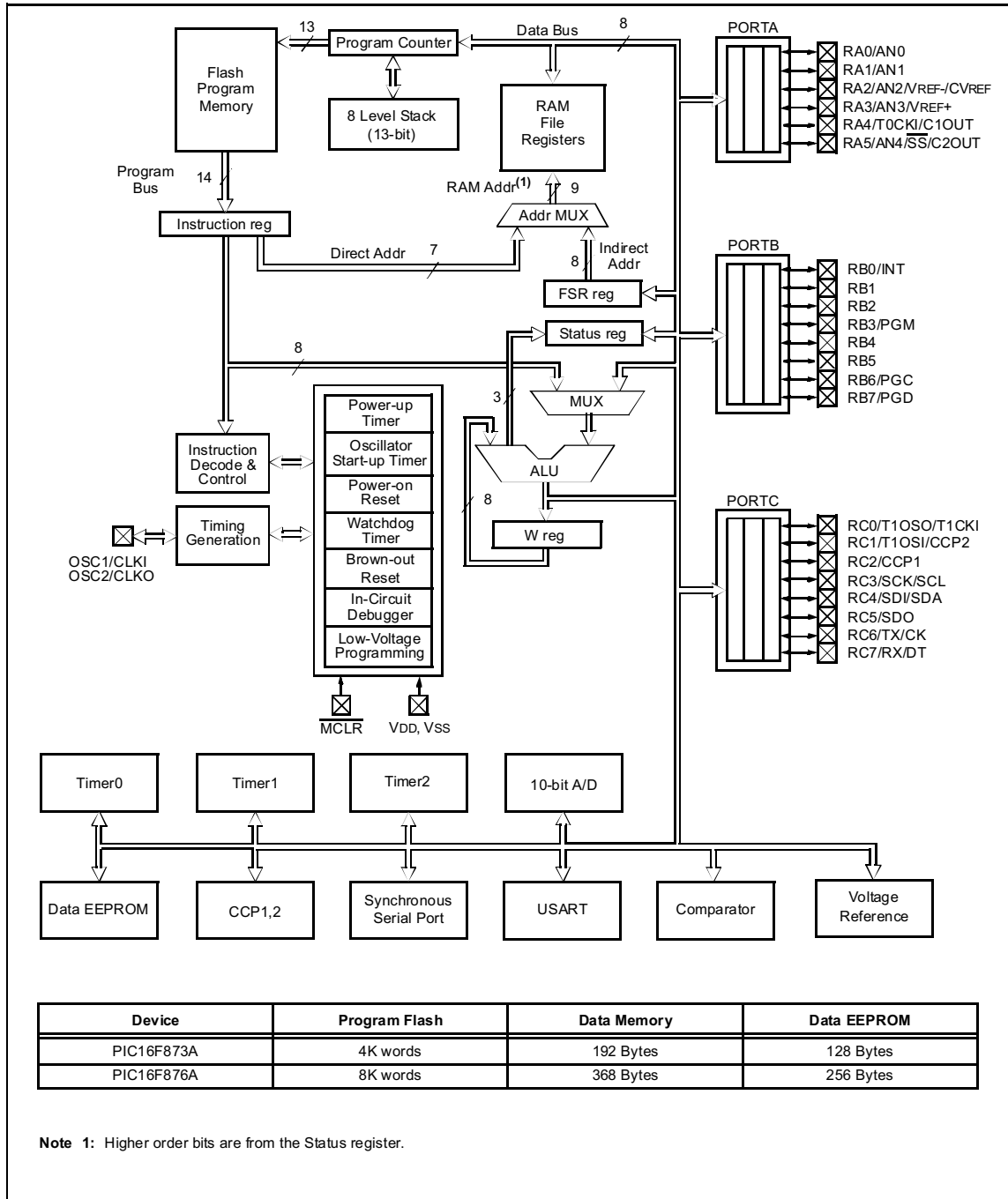
Additional information may be found in the PICmicro® Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly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

TABLE 1-1: PIC16F87XA DEVICE FEATURES

Key Features	PIC16F873A	PIC16F874A	PIC16F876A	PIC16F877A
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 (bytes)	128	128	256	256
Interrupts	14	15	14	15
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
Analog Comparators	2	2	2	2
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN

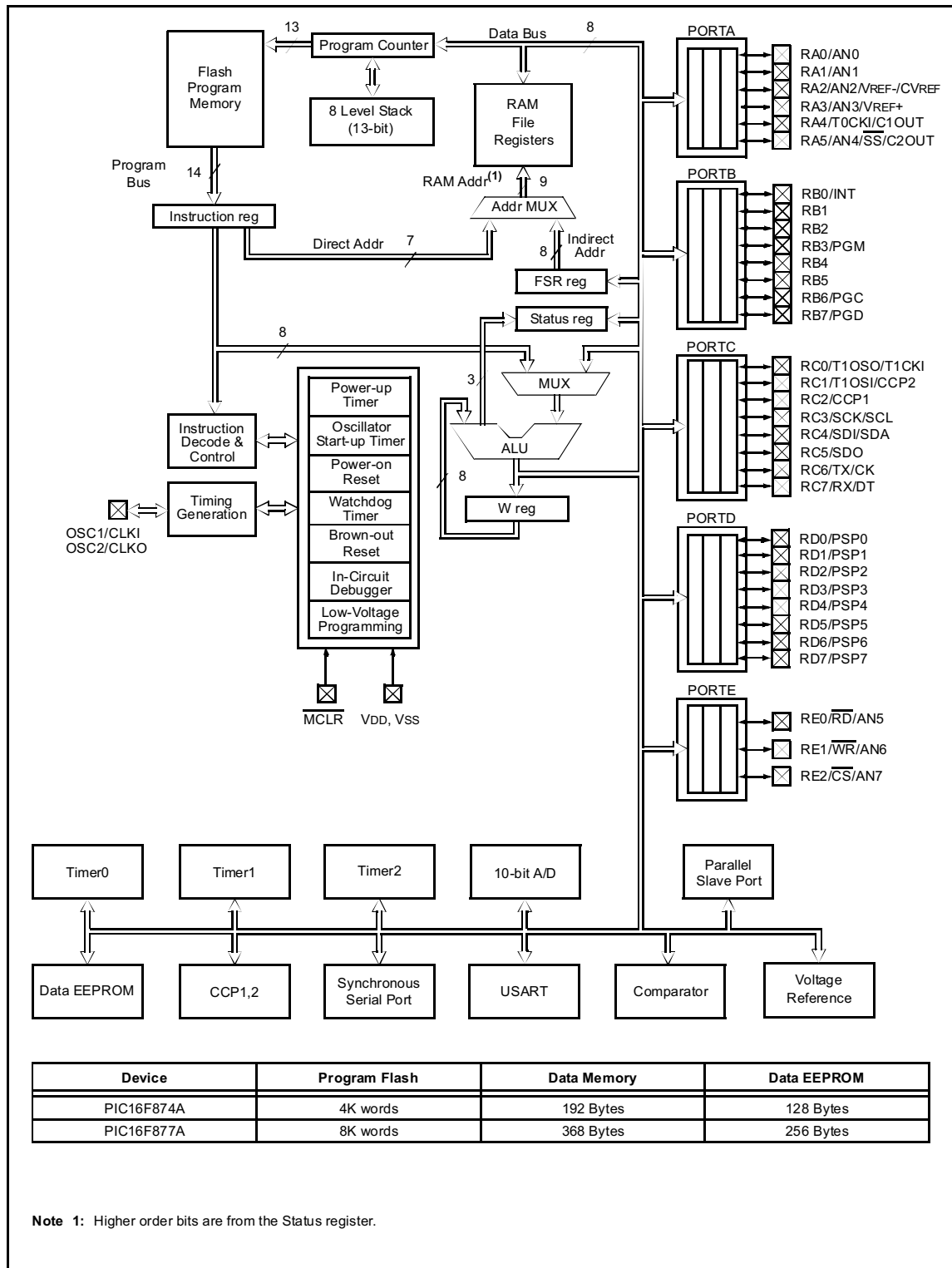
PIC16F87XA

FIGURE 1-1: PIC16F873A/876A BLOCK DIAGRAM



PIC16F87XA

FIGURE 1-2: PIC16F874A/877A BLOCK DIAGRAM



PIC16F87XA

TABLE 1-2: PIC16F873A/876A PINOUT DESCRIPTION

Pin Name	PDIP, SOIC, SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI OSC1 CLKI	9	6	I I	ST/CMOS ⁽³⁾	Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise CMOS. External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO OSC2 CLKO	10	7	O O	—	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
MCLR/VPP MCLR VPP	1	26	I P	ST	Master Clear (input) or programming voltage (output). Master Clear (Reset) input. This pin is an active low Reset to the device. Programming voltage input.
RA0/AN0 RA0 AN0	2	27	I/O I	TTL	PORTA is a bidirectional I/O port. Digital I/O. Analog input 0.
RA1/AN1 RA1 AN1	3	28	I/O I	TTL	
RA2/AN2/VREF-/ CVREF RA2 AN2 VREF- CVREF	4	1	I/O I I O	TTL	
RA3/AN3/VREF+ RA3 AN3 VREF+	5	2	I/O I I	TTL	
RA4/T0CKI/C1OUT RA4 T0CKI C1OUT	6	3	I/O I O	ST	
RA5/AN4/SS/C2OUT RA5 AN4 SS C2OUT	7	4	I/O I I O	TTL	

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.
Note 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
Note 3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

PIC16F87XA

TABLE 1-2: PIC16F873A/876A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP, SOIC, SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RB0/INT RB0 INT	21	18	I/O I	TTL/ST ⁽¹⁾	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs. Digital I/O. External interrupt.
RB1	22	19	I/O	TTL	Digital I/O.
RB2	23	20	I/O	TTL	Digital I/O.
RB3/PGM RB3 PGM	24	21	I/O I	TTL	Digital I/O. Low-voltage (single-supply) ICSP programming enable pin.
RB4	25	22	I/O	TTL	Digital I/O.
RB5	26	23	I/O	TTL	Digital I/O.
RB6/PGC RB6 PGC	27	24	I/O I	TTL/ST ⁽²⁾	Digital I/O. In-circuit debugger and ICSP programming clock.
RB7/PGD RB7 PGD	28	25	I/O I/O	TTL/ST ⁽²⁾	Digital I/O. In-circuit debugger and ICSP programming data.
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	11	8	I/O O I	ST	PORTC is a bidirectional I/O port. Digital I/O. Timer1 oscillator output. Timer1 external clock input.
RC1/T1OSI/CCP2 RC1 T1OSI CCP2	12	9	I/O I I/O	ST	Digital I/O. Timer1 oscillator input. Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1 RC2 CCP1	13	10	I/O I/O	ST	Digital I/O. Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL RC3 SCK SCL	14	11	I/O I/O I/O	ST	Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I ² C mode.
RC4/SDI/SDA RC4 SDI SDA	15	12	I/O I I/O	ST	Digital I/O. SPI data in. I ² C data I/O.
RC5/SDO RC5 SDO	16	13	I/O O	ST	Digital I/O. SPI data out.
RC6/TX/CK RC6 TX CK	17	14	I/O O I/O	ST	Digital I/O. USART asynchronous transmit. USART1 synchronous clock.
RC7/RX/DT RC7 RX DT	18	15	I/O I I/O	ST	Digital I/O. USART asynchronous receive. USART synchronous data.
V _{SS}	8, 19	5, 6	P	—	Ground reference for logic and I/O pins.
V _{DD}	20	17	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.
Note 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
Note 3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

PIC16F87XA

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI OSC1 CLKI	13	14	30	32	I I	ST/CMOS ⁽⁴⁾	Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise CMOS. External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO OSC2 CLKO	14	15	31	33	O O	—	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
MCLR/VPP MCLR VPP	1	2	18	18	I P	ST	Master Clear (input) or programming voltage (output). Master Clear (Reset) input. This pin is an active low Reset to the device. Programming voltage input.
RA0/AN0 RA0 AN0 RA1/AN1 RA1 AN1 RA2/AN2/VREF-/CVREF RA2 AN2 VREF- CVREF RA3/AN3/VREF+ RA3 AN3 VREF+ RA4/T0CKI/C1OUT RA4 T0CKI C1OUT RA5/AN4/SS/C2OUT RA5 AN4 SS C2OUT	2 3 4 5 6 7	3 4 5 6 8	19 20 21 22 23 24	19 20 21 22 23 24	I/O I I/O I I/O I I O I I O I/O I I O	TTL TTL TTL TTL ST TTL	PORTA is a bidirectional I/O port. Digital I/O. Analog input 0. Digital I/O. Analog input 1. Digital I/O. Analog input 2. A/D reference voltage (Low) input. Comparator VREF output. Digital I/O. Analog input 3. A/D reference voltage (High) input. Digital I/O – Open-drain when configured as output. Timer0 external clock input. Comparator 1 output. Digital I/O. Analog input 4. SPI slave select input. Comparator 2 output.

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.
Note 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
Note 3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

PIC16F87XA

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RB0/INT RB0 INT	33	36	8	9	I/O I	TTL/ST ⁽¹⁾	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. Digital I/O. External interrupt.
RB1	34	37	9	10	I/O	TTL	Digital I/O.
RB2	35	38	10	11	I/O	TTL	Digital I/O.
RB3/PGM RB3 PGM	36	39	11	12	I/O I	TTL	Digital I/O. Low-voltage ICSP programming enable pin.
RB4	37	41	14	14	I/O	TTL	Digital I/O.
RB5	38	42	15	15	I/O	TTL	Digital I/O.
RB6/PGC RB6 PGC	39	43	16	16	I/O I	TTL/ST ⁽²⁾	Digital I/O. In-circuit debugger and ICSP programming clock.
RB7/PGD RB7 PGD	40	44	17	17	I/O I/O	TTL/ST ⁽²⁾	Digital I/O. In-circuit debugger and ICSP programming data.

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

- Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt.
Note 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
Note 3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

PIC16F87XA

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	15	16	32	34	I/O O I	ST	PORTC is a bidirectional I/O port. Digital I/O. Timer1 oscillator output. Timer1 external clock input.
RC1/T1OSI/CCP2 RC1 T1OSI CCP2	16	18	35	35	I/O I I/O	ST	Digital I/O. Timer1 oscillator input. Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1 RC2 CCP1	17	19	36	36	I/O I/O	ST	Digital I/O. Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL RC3 SCK SCL	18	20	37	37	I/O I/O I/O	ST	Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I ² C mode.
RC4/SDI/SDA RC4 SDI SDA	23	25	42	42	I/O I I/O	ST	Digital I/O. SPI data in. I ² C data I/O.
RC5/SDO RC5 SDO	24	26	43	43	I/O O	ST	Digital I/O. SPI data out.
RC6/TX/CK RC6 TX CK	25	27	44	44	I/O O I/O	ST	Digital I/O. USART asynchronous transmit. USART1 synchronous clock.
RC7/RX/DT RC7 RX DT	26	29	1	1	I/O I I/O	ST	Digital I/O. USART asynchronous receive. USART synchronous data.

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

- Note** 1: This buffer is a Schmitt Trigger input when configured as 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.

PIC16F87XA

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RD0/PSP0 RD0 PSP0	19	21	38	38	I/O I/O	ST/TTL ⁽³⁾	PORTD is a bidirectional I/O port or Parallel Slave Port when interfacing to a microprocessor bus. Digital I/O. Parallel Slave Port data.
RD1/PSP1 RD1 PSP1	20	22	39	39	I/O I/O	ST/TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD2/PSP2 RD2 PSP2	21	23	40	40	I/O I/O	ST/TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD3/PSP3 RD3 PSP3	22	24	41	41	I/O I/O	ST/TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD4/PSP4 RD4 PSP4	27	30	2	2	I/O I/O	ST/TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD5/PSP5 RD5 PSP5	28	31	3	3	I/O I/O	ST/TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD6/PSP6 RD6 PSP6	29	32	4	4	I/O I/O	ST/TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD7/PSP7 RD7 PSP7	30	33	5	5	I/O I/O	ST/TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RE0/RD/AN5 RE0 RD AN5	8	9	25	25	I/O I I	ST/TTL ⁽³⁾	PORTE is a bidirectional I/O port. Digital I/O. Read control for Parallel Slave Port. Analog input 5.
RE1/WR/AN6 RE1 WR AN6	9	10	26	26	I/O I I	ST/TTL ⁽³⁾	Digital I/O. Write control for Parallel Slave Port. Analog input 6.
RE2/CS/AN7 RE2 CS AN7	10	11	27	27	I/O I I	ST/TTL ⁽³⁾	Digital I/O. Chip select control for Parallel Slave Port. Analog input 7.
Vss	12, 31	13, 34	6, 29	6, 30, 31	P	—	Ground reference for logic and I/O pins.
VDD	11, 32	12, 35	7, 28	7, 8, 28, 29	P	—	Positive supply for logic and I/O pins.
NC	—	1, 17, 28, 40	12, 13, 33, 34	13	—	—	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 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.

PIC16F87XA

NOTES:

PIC16F87XA

2.0 MEMORY ORGANIZATION

There are three memory blocks in each of the PIC16F87XA devices. 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 3.0 "Data EEPROM and Flash Program Memory"**.

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

2.1 Program Memory Organization

The PIC16F87XA devices have a 13-bit program counter capable of addressing an 8K word x 14 bit program memory space. The PIC16F876A/877A devices have 8K words x 14 bits of Flash program memory, while PIC16F873A/874A devices have 4K words x 14 bits. 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: PIC16F876A/877A PROGRAM MEMORY MAP AND STACK

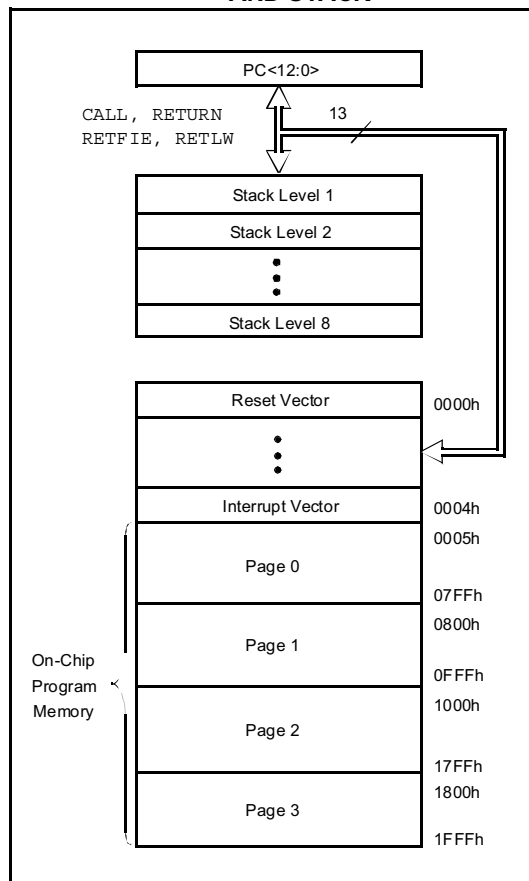
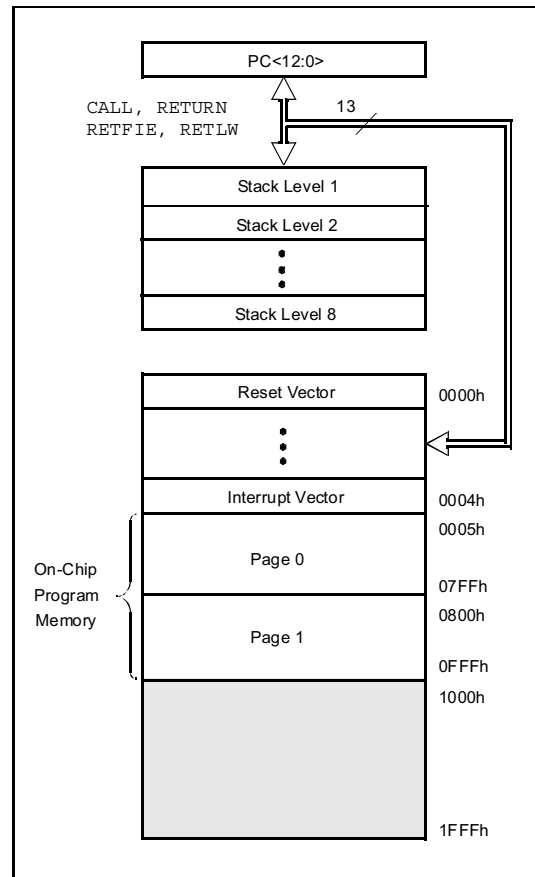


FIGURE 2-2: PIC16F873A/874A PROGRAM MEMORY MAP AND STACK



PIC16F87XA

2.2 Data Memory Organization

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

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

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

Note: The EEPROM data memory description can be found in **Section 3.0 “Data EEPROM and Flash Program Memory”** 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).

PIC16F87XA

FIGURE 2-3: PIC16F876A/877A REGISTER FILE MAP

File Address		File Address		File Address		File Address	
Indirect addr. ^(*)	00h	Indirect addr. ^(*)	80h	Indirect addr. ^(*)	100h	Indirect addr. ^(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD ⁽¹⁾	08h	TRISD ⁽¹⁾	88h		108h		188h
PORTE ⁽¹⁾	09h	TRISE ⁽¹⁾	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved ⁽²⁾	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved ⁽²⁾	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h	General Purpose Register 16 Bytes	111h	General Purpose Register 16 Bytes	191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch	CMCON	9Ch		11Ch		19Ch
CCP2CON	1Dh	CVRCON	9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Register 96 Bytes	20h	General Purpose Register 80 Bytes	A0h		General Purpose Register 80 Bytes		120h
			EFh			1EFh	
			F0h			1F0h	
		accesses 70h-7Fh		accesses 70h-7Fh		accesses 70h-7Fh	
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 the PIC16F876A.
Note 2: These registers are reserved; maintain these registers clear.

PIC16F87XA

FIGURE 2-4: PIC16F873A/874A REGISTER FILE MAP

File Address		File Address		File Address		File Address	
Indirect addr. ^(*)	00h	Indirect addr. ^(*)	80h	Indirect addr. ^(*)	100h	Indirect addr. ^(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD ⁽¹⁾	08h	TRISD ⁽¹⁾	88h		108h		188h
PORTE ⁽¹⁾	09h	TRISE ⁽¹⁾	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved ⁽²⁾	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved ⁽²⁾	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h				
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	CMCON	9Ch				
CCP2CON	1Dh	CVRCON	9Dh				
ADRESH	1Eh	ADRESL	9Eh				
ADCON0	1Fh	ADCON1	9Fh				
	20h		A0h		120h		1A0h
General Purpose Register 96 Bytes		General Purpose Register 96 Bytes		accesses 20h-7Fh		accesses A0h - FFh	
	7Fh		FFh		16Fh		1EFh
					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 the PIC16F873A.
Note 2: These registers are reserved; maintain these registers clear.

PIC16F87XA

2.2.2 SPECIAL FUNCTION REGISTERS

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

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

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:	
Bank 0												
00h ⁽³⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								00 00 00 00	31, 150	
01h	TMR0	Timer0 Module Register								xxxx xxxx	55, 150	
02h ⁽³⁾	PCL	Program Counter (PC) Least Significant Byte								00 00 00 00	30, 150	
03h ⁽³⁾	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	00 01 1xxx	22, 150	
04h ⁽³⁾	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	31, 150	
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						-- 0x 00 00	43, 150	
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	45, 150	
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	47, 150	
08h ⁽⁴⁾	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	48, 150	
09h ⁽⁴⁾	PORTE	—	—	—	—	—	RE2	RE1	RE0	--- -xxx	49, 150	
0Ah ^(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter						-- 0 00 00	30, 150
0Bh ⁽³⁾	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBFIF	00 00 00 0x	24, 150	
0Ch	PIR1	PSPIF ⁽³⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	00 00 00 00	26, 150	
0Dh	PIR2	—	CMIF	—	EEIF	BCLIF	—	—	CCP2IF	- 0 - 0 - - 0	28, 150	
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	60, 150	
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	60, 150	
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	-- 00 00 00	57, 150	
11h	TMR2	Timer2 Module Register								00 00 00 00	62, 150	
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	- 0 00 00 00	61, 150	
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	79, 150	
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	00 00 00 00	82, 82, 150	
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	63, 150	
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	63, 150	
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	-- 00 00 00	64, 150	
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	00 00 00 0x	112, 150	
19h	TXREG	USART Transmit Data Register								00 00 00 00	118, 150	
1Ah	RCREG	USART Receive Data Register								00 00 00 00	118, 150	
1Bh	CCPR2L	Capture/Compare/PWM Register 2 (LSB)								xxxx xxxx	63, 150	
1Ch	CCPR2H	Capture/Compare/PWM Register 2 (MSB)								xxxx xxxx	63, 150	
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	-- 00 00 00	64, 150	
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	133, 150	
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON	00 00 00 - 0	127, 150	

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.
- Note 2:** Bits PSPIE and PSPIF are reserved on PIC16F873A/876A devices; always maintain these bits clear.
- Note 3:** These registers can be addressed from any bank.
- Note 4:** PORTD, PORTE, TRISD and TRISE are not implemented on PIC16F873A/876A devices, read as '0'.
- Note 5:** Bit 4 of EEADRH implemented only on the PIC16F876A/877A devices.

PIC16F87XA

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:	
Bank 1												
80h ⁽³⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)									0000 0000	31, 150
81h	OPTION_REG	RBPV	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	23, 150	
82h ⁽³⁾	PCL	Program Counter (PC) Least Significant Byte									0000 0000	30, 150
83h ⁽³⁾	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxx	22, 150	
84h ⁽³⁾	FSR	Indirect Data Memory Address Pointer									xxxx xxxx	31, 150
85h	TRISA	—	—	PORTA Data Direction Register						—11 1111	43, 150	
86h	TRISB	PORTB Data Direction Register									1111 1111	45, 150
87h	TRISC	PORTC Data Direction Register									1111 1111	47, 150
88h ⁽⁴⁾	TRISD	PORTD Data Direction Register									1111 1111	48, 151
89h ⁽⁴⁾	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction bits				0000 -111	50, 151
8Ah ^(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter						--0 0000	30, 150
8Bh ⁽³⁾	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBF	0000 000x	24, 150	
8Ch	PIE1	PSPIE ⁽²⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	25, 151	
8Dh	PIE2	—	CMIE	—	EEIE	BCLIE	—	—	CCP2IE	-0-0 0--0	27, 151	
8Eh	PCON	—	—	—	—	—	—	POR	BOR	---- -qq	29, 151	
8Fh	—	Unimplemented									—	—
90h	—	Unimplemented									—	—
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	83, 151	
92h	PR2	Timer2 Period Register									1111 1111	62, 151
93h	SSPADDD	Synchronous Serial Port (I ² C mode) Address Register									0000 0000	79, 151
94h	SSPSTAT	SMP	CKE	$D\overline{A}$	P	S	$R\overline{W}$	UA	BF	0000 0000	79, 151	
95h	—	Unimplemented									—	—
96h	—	Unimplemented									—	—
97h	—	Unimplemented									—	—
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	111, 151	
99h	SPBRG	Baud Rate Generator Register									0000 0000	113, 151
9Ah	—	Unimplemented									—	—
9Bh	—	Unimplemented									—	—
9Ch	CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111	135, 151	
9Dh	CVRCON	CVREN	CVROE	CVRR	—	CVR3	CVR2	CVR1	CVR0	000- 0000	141, 151	
9Eh	ADRESL	A/D Result Register Low Byte									xxxx xxxx	133, 151
9Fh	ADCON1	ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0	00-- 0000	128, 151	

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:** Bits PSPIE and PSPIF are reserved on PIC16F873A/876A devices; always maintain these bits clear.
- 3:** These registers can be addressed from any bank.
- 4:** PORTD, PORTE, TRISD and TRISE are not implemented on PIC16F873A/876A devices, read as '0'.
- 5:** Bit 4 of EEADRH implemented only on the PIC16F876A/877A devices.

PIC16F87XA

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:	
Bank 2												
100h ⁽³⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)									00 00 00 00	31, 150
101h	TMR0	Timer0 Module Register									xxxx xxxx	55, 150
102h ⁽³⁾	PCL	Program Counter's (PC) Least Significant Byte									00 00 00 00	30, 150
103h ⁽³⁾	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	00 01 1xxx	22, 150	
104h ⁽³⁾	FSR	Indirect Data Memory Address Pointer									xxxx xxxx	31, 150
105h	—	Unimplemented									—	—
106h	PORTB	PORTB Data Latch when written: PORTB pins when read									xxxx xxxx	45, 150
107h	—	Unimplemented									—	—
108h	—	Unimplemented									—	—
109h	—	Unimplemented									—	—
10Ah ^(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter						-- 0 00 00	30, 150
10Bh ⁽³⁾	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	00 00 00 0x	24, 150	
10Ch	EEDATA	EEPROM Data Register Low Byte									xxxx xxxx	39, 151
10Dh	EEADR	EEPROM Address Register Low Byte									xxxx xxxx	39, 151
10Eh	EEDATH	—	—	EEPROM Data Register High Byte						-- xx xxxx	39, 151	
10Fh	EEADRH	—	—	—	— ⁽⁵⁾	EEPROM Address Register High Byte				---- xxxx	39, 151	
Bank 3												
180h ⁽³⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)									00 00 00 00	31, 150
181h	OPTION_REG	\overline{RBFU}	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	11 11 11 11	23, 150	
182h ⁽³⁾	PCL	Program Counter (PC) Least Significant Byte									00 00 00 00	30, 150
183h ⁽³⁾	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	00 01 1xxx	22, 150	
184h ⁽³⁾	FSR	Indirect Data Memory Address Pointer									xxxx xxxx	31, 150
185h	—	Unimplemented									—	—
186h	TRISB	PORTB Data Direction Register									11 11 11 11	45, 150
187h	—	Unimplemented									—	—
188h	—	Unimplemented									—	—
189h	—	Unimplemented									—	—
18Ah ^(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter						-- 0 00 00	30, 150
18Bh ⁽³⁾	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	00 00 00 0x	24, 150	
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x-- x00 0	34, 151	
18Dh	EECON2	EEPROM Control Register 2 (not a physical register)									---- ----	39, 151
18Eh	—	Reserved; maintain clear									00 00 00 00	—
18Fh	—	Reserved; maintain clear									00 00 00 00	—

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:** Bits PSPIE and PSPIF are reserved on PIC16F873A/876A devices; always maintain these bits clear.
- 3:** These registers can be addressed from any bank.
- 4:** PORTD, PORTE, TRISD and TRISE are not implemented on PIC16F873A/876A devices, read as '0'.
- 5:** Bit 4 of EEADRH implemented only on the PIC16F876A/877A devices.

PIC16F87XA

2.2.2.1 Status Register

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

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

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

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the Status register because these instructions do not affect the Z, C or DC bits from the Status register. For other instructions not affecting any status bits, see **Section 15.0 “Instruction Set Summary”**.

Note: The C and DC bits operate as a borrow and digit borrow bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

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

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	TO	PD	Z	DC	C
bit 7					bit 0		

bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)

1 = Bank 2, 3 (100h-1FFh)
0 = Bank 0, 1 (00h-FFh)

bit 6-5 **RP1:RP0:** Register Bank Select bits (used for direct addressing)

11 = Bank 3 (180h-1FFh)
10 = Bank 2 (100h-17Fh)
01 = Bank 1 (80h-FFh)
00 = Bank 0 (00h-7Fh)
Each bank is 128 bytes.

bit 4 **TO:** Time-out bit

1 = After power-up, `CLRWDT` instruction or `SLEEP` instruction
0 = A WDT time-out occurred

bit 3 **PD:** Power-down bit

1 = After power-up or by the `CLRWDT` instruction
0 = By execution of the `SLEEP` instruction

bit 2 **Z:** Zero bit

1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
(for borrow, the polarity is reversed)

1 = A carry-out from the 4th low order bit of the result occurred
0 = No carry-out from the 4th low order bit of the result

bit 0 **C:** Carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)

1 = A carry-out from the Most Significant bit of the result occurred
0 = No carry-out from the Most Significant bit of the result occurred

Note: For borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high, or low order bit of the source register.

Legend:

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

PIC16F87XA

2.2.2.2 OPTION_REG Register

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

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

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

	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
								bit 0
bit 7								bit 7

- bit 7 **RBPU:** PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values
- bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin
- bit 5 **T0CS:** TMR0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (CLKO)
- bit 4 **T0SE:** TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/T0CKI pin
- bit 3 **PSA:** Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module
- bit 2-0 **PS2:PS0:** Prescaler Rate Select bits

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

Legend:

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

Note: When using Low-Voltage ICSP Programming (LVP) and the pull-ups on PORTB are enabled, bit 3 in the TRISB register must be cleared to disable the pull-up on RB3 and ensure the proper operation of the device

PIC16F87XA

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

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

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF
	bit 7							bit 0

- bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts
- bit 6 **PEIE:** Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
- bit 5 **TMR0IE:** 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 **TMR0IF:** 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; a mismatch condition will continue to set the bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared (must be cleared in software).
0 = None of the RB7:RB4 pins have changed state

Legend:

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

PIC16F87XA

2.2.2.4 PIE1 Register

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

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

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

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0	

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

Note 1: PSPIE is reserved on PIC16F873A/876A devices; always maintain this bit clear.
- 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

Legend:

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

PIC16F87XA

2.2.2.5 PIR1 Register

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

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

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

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

- bit 7 **PSPIF:** Parallel Slave Port Read/Write Interrupt Flag bit⁽¹⁾
 1 = A read or a write operation has taken place (must be cleared in software)
 0 = No read or write has occurred
Note 1: PSPIF is reserved on PIC16F873A/876A devices; always maintain this bit clear.
- bit 6 **ADIF:** A/D Converter Interrupt Flag bit
 1 = An A/D conversion completed
 0 = The A/D conversion is not complete
- bit 5 **RCIF:** USART Receive Interrupt Flag bit
 1 = The USART receive buffer is full
 0 = The USART receive buffer is empty
- bit 4 **TXIF:** USART Transmit Interrupt Flag bit
 1 = The USART transmit buffer is empty
 0 = The USART transmit buffer is full
- bit 3 **SSPIF:** Synchronous Serial Port (SSP) Interrupt Flag bit
 1 = The SSP interrupt condition has occurred and must be cleared in software before returning from the Interrupt Service Routine. The conditions that will set this bit are:
 - SPI – A transmission/reception has taken place.
 - I²C Slave – A transmission/reception has taken place.
 - I²C Master
 - A transmission/reception has taken place.
 - The initiated Start condition was completed by the SSP module.
 - The initiated Stop condition was completed by the SSP module.
 - The initiated Restart condition was completed by the SSP module.
 - The initiated Acknowledge condition was completed by the SSP module.
 - A Start condition occurred while the SSP module was Idle (multi-master system).
 - A Stop condition occurred while the SSP module was Idle (multi-master system).
 0 = No SSP interrupt condition has occurred
- bit 2 **CCP1IF:** CCP1 Interrupt Flag bit
Capture mode:
 1 = A TMR1 register capture occurred (must be cleared in software)
 0 = No TMR1 register capture occurred
Compare mode:
 1 = A TMR1 register compare match occurred (must be cleared in software)
 0 = No TMR1 register compare match occurred
PWM mode:
 Unused in this mode.
- bit 1 **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit
 1 = TMR2 to PR2 match occurred (must be cleared in software)
 0 = No TMR2 to PR2 match occurred
- bit 0 **TMR1IF:** TMR1 Overflow Interrupt Flag bit
 1 = TMR1 register overflowed (must be cleared in software)
 0 = TMR1 register did not overflow

Legend:

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

PIC16F87XA

2.2.2.6 PIE2 Register

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

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

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

	U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
	—	CMIE	—	EEIE	BCLIE	—	—	CCP2IE
bit 7								bit 0

- bit 7 **Unimplemented:** Read as '0'
- bit 6 **CMIE:** Comparator Interrupt Enable bit
 1 = Enables the comparator interrupt
 0 = Disable the comparator interrupt
- bit 5 **Unimplemented:** Read as '0'
- bit 4 **EEIE:** EEPROM Write Operation Interrupt Enable bit
 1 = Enable EEPROM write interrupt
 0 = Disable EEPROM write interrupt
- bit 3 **BCLIE:** Bus Collision Interrupt Enable bit
 1 = Enable bus collision interrupt
 0 = Disable bus collision interrupt
- bit 2-1 **Unimplemented:** Read as '0'
- bit 0 **CCP2IE:** CCP2 Interrupt Enable bit
 1 = Enables the CCP2 interrupt
 0 = Disables the CCP2 interrupt

Legend:

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

PIC16F87XA

2.2.2.7 PIR2 Register

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

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

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

U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
—	CMIF	—	EEIF	BCLIF	—	—	CCP2IF
bit 7				bit 0			

- bit 7 **Unimplemented:** Read as '0'
- bit 6 **CMIF:** Comparator Interrupt Flag bit
 1 = The comparator input has changed (must be cleared in software)
 0 = The comparator input has not changed
- bit 5 **Unimplemented:** Read as '0'
- bit 4 **EEIF:** EEPROM Write Operation Interrupt Flag bit
 1 = The write operation completed (must be cleared in software)
 0 = The write operation is not complete or has not been started
- bit 3 **BCLIF:** Bus Collision Interrupt Flag bit
 1 = A bus collision has occurred in the SSP when configured for I²C Master mode
 0 = No bus collision has occurred
- bit 2-1 **Unimplemented:** Read as '0'
- bit 0 **CCP2IF:** CCP2 Interrupt Flag bit
 Capture mode:
 1 = A TMR1 register capture occurred (must be cleared in software)
 0 = No TMR1 register capture occurred
 Compare mode:
 1 = A TMR1 register compare match occurred (must be cleared in software)
 0 = No TMR1 register compare match occurred
 PWM mode:
 Unused.

Legend:

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

PIC16F87XA

2.2.2.8 PCON Register

The Power Control (PCON) register contains flag bits to allow differentiation between a Power-on Reset (POR), a Brown-out Reset (BOR), a Watchdog Reset (WDT) and an external MCLR Reset.

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

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

	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-1
	—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0	

bit 7-2 **Unimplemented:** Read as ‘0’

bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit

1 = No Power-on Reset occurred

0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{\text{BOR}}$:** Brown-out Reset Status bit

1 = No Brown-out Reset occurred

0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as ‘0’

- n = Value at POR

‘1’ = Bit is set

‘0’ = Bit is cleared

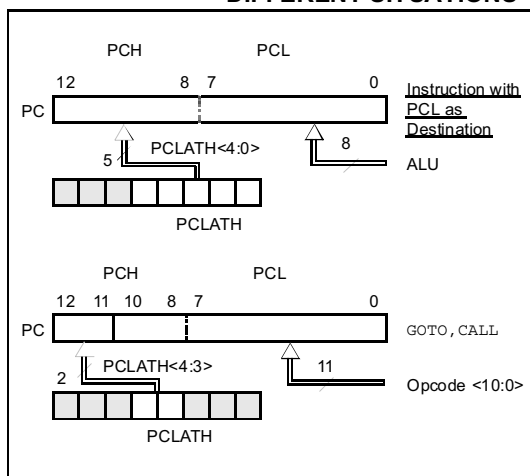
x = Bit is unknown

PIC16F87XA

2.3 PCL and PCLATH

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

FIGURE 2-5: LOADING OF PC IN DIFFERENT SITUATIONS



2.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to the application note, AN556, "Implementing a Table Read" (DS00556).

2.3.2 STACK

The PIC16F87XA family has an 8-level deep x 13-bit wide hardware stack. The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed, or an interrupt causes a branch. The stack is POP'ed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

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

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

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

2.4 Program Memory Paging

All PIC16F87XA devices are capable of addressing a continuous 8K word block of program memory. The CALL and GOTO instructions provide only 11 bits of address to allow branching within any 2K program memory page. When doing a CALL or GOTO instruction, the upper 2 bits of the address are provided by PCLATH<4:3>. When doing a CALL or GOTO instruction, the user must ensure that the page select bits are programmed so that the desired program memory page is addressed. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is popped off the stack. Therefore, manipulation of the PCLATH<4:3> bits is not required for the RETURN instructions (which POPs the address from the stack).

Note: The contents of the PCLATH register are unchanged after a RETURN or RETFIE instruction is executed. The user must rewrite the contents of the PCLATH register for any subsequent subroutine calls or GOTO instructions.

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

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

```

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


2.5 Indirect Addressing, INDF and FSR Registers

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

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

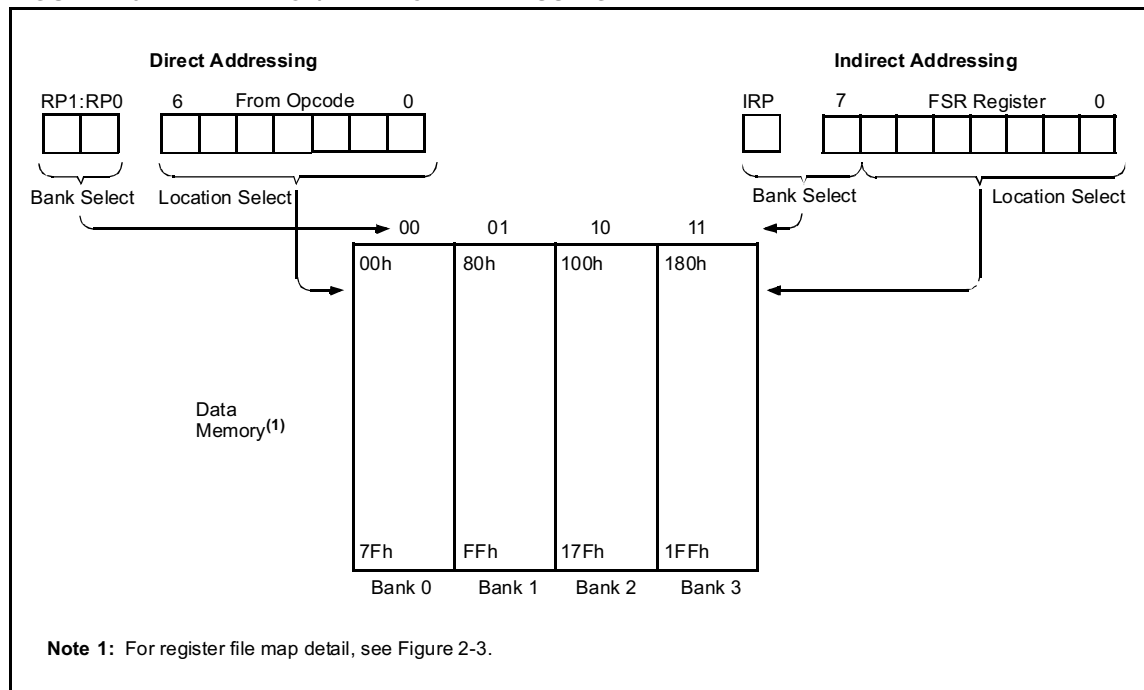
A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 2-2.

EXAMPLE 2-2: INDIRECT ADDRESSING

```

MOV LW 0x20 ;initialize pointer
MOV WF FSR ;to RAM
NEXT   CLR F INDF ;clear INDF register
       INC F FSR,F ;inc pointer
       BT FSS FSR,4 ;all done?
       GOTO NEXT ;no clear next
CONTINUE
:      ;yes continue
    
```

FIGURE 2-6: DIRECT/INDIRECT ADDRESSING



PIC16F87XA

NOTES:

3.0 DATA EEPROM AND FLASH PROGRAM MEMORY

The data EEPROM and Flash program memory is readable and writable during normal operation (over the full VDD range). This 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 this memory:

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

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. These devices have 128 or 256 bytes of data EEPROM (depending on the device), with an address range from 00h to FFh. On devices with 128 bytes, addresses from 80h to FFh are unimplemented and will wraparound to the beginning of data EEPROM memory. When writing to unimplemented locations, the on-chip charge pump will be turned off.

When interfacing the program memory block, the EEDATA and EEDATH registers form a two-byte word that holds the 14-bit data for read/write and the EEADR and EEADRH registers form a two-byte word that holds the 13-bit address of the program memory location being accessed. These devices have 4 or 8K words of program Flash, with an address range from 0000h to 0FFFh for the PIC16F873A/874A and 0000h to 1FFFh for the PIC16F876A/877A. Addresses above the range of the respective device will wraparound to the beginning of program memory.

The EEPROM data memory allows single-byte read and write. The Flash program memory allows single-word reads and four-word block writes. Program memory write operations automatically perform an erase-before-write on blocks of four words. A byte write in data EEPROM memory automatically erases the location and writes the new data (erase-before-write).

The write time is controlled by an on-chip timer. The write/erase voltages are generated by an on-chip charge pump, rated to operate over the voltage range of the device for byte or word operations.

When the device is code-protected, the CPU may continue to read and write the data EEPROM memory. Depending on the settings of the write-protect bits, the device may or may not be able to write certain blocks of the program memory; however, reads of the program memory are allowed. When code-protected, the device programmer can no longer access data or program memory; this does NOT inhibit internal reads or writes.

3.1 EEADR and EEADRH

The EEADRH:EEADR register pair can address up to a maximum of 256 bytes of data EEPROM or up to a maximum of 8K words of program EEPROM. When selecting a data address value, only the LSByte of the address is written to the EEADR register. 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.

If the device contains less memory than the full address reach of the address register pair, the Most Significant bits of the registers are not implemented. For example, if the device has 128 bytes of data EEPROM, the Most Significant bit of EEADR is not implemented on access to data EEPROM.

3.2 EECON1 and EECON2 Registers

EECON1 is the control register for memory accesses.

Control bit, EEPGD, determines if the access will be a program or data memory access. When clear, as it is when reset, 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 or erase, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write or erase operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write (or erase) operation is interrupted by a MCLR 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 data and address will be unchanged in the EEDATA and EEADR registers.

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

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

Note: The self-programming mechanism for Flash program memory has been changed. On previous PIC16F87X devices, Flash programming was done in single-word erase/write cycles. The newer PIC18F87XA devices use a four-word erase/write cycle. See **Section 3.6 "Writing to Flash Program Memory"** for more information.

PIC16F87XA

REGISTER 3-1: EECON1 REGISTER (ADDRESS 18Ch)

R/W-x	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	—	—	—	WRERR	WREN	WR	RD
bit 7				bit 0			

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

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

3.3 Reading 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 cycle in the EEDATA register; therefore, it can be read in the next instruction (see Example 3-1). EEDATA will hold this value until another read or until it is written to by the user (during a write operation).

The steps to reading the EEPROM data memory are:

1. Write the address to EEADR. Make sure that the address is not larger than the memory size of the device.
2. Clear the EEPGD bit to point to EEPROM data memory.
3. Set the RD bit to start the read operation.
4. Read the data from the EEDATA register.

EXAMPLE 3-1: DATA EEPROM READ

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

3.4 Writing to Data EEPROM Memory

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDATA register. Then the user must follow a specific write sequence to initiate the write for each byte.

The write will not initiate if the write sequence is not exactly followed (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. We strongly recommend that interrupts be disabled during this code segment (see Example 3-2).

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

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set. At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. EEIF must be cleared by software.

The steps to write to EEPROM data memory are:

1. If step 10 is not implemented, check the WR bit to see if a write is in progress.
2. Write the address to EEADR. Make sure that the address is not larger than the memory size of the device.
3. Write the 8-bit data value to be programmed in the EEDATA register.
4. Clear the EEPGD bit to point to EEPROM data memory.
5. Set the WREN bit to enable program operations.
6. Disable interrupts (if enabled).
7. Execute the special five instruction sequence:
 - Write 55h to EECON2 in two steps (first to W, then to EECON2)
 - Write AAh to EECON2 in two steps (first to W, then to EECON2)
 - Set the WR bit
8. Enable interrupts (if using interrupts).
9. Clear the WREN bit to disable program operations.
10. At the completion of the write cycle, the WR bit is cleared and the EEIF interrupt flag bit is set. (EEIF must be cleared by firmware.) If step 1 is not implemented, then firmware should check for EEIF to be set, or WR to clear, to indicate the end of the program cycle.

EXAMPLE 3-2: DATA EEPROM WRITE

```
BSF    STATUS,RP1    ;
BSF    STATUS,RP0
BTFSC  EECON1,WR     ;Wait for write
GOTO   $-1          ;to complete
BCF    STATUS,RP0    ;Bank 2
MOVF   DATA_EE_ADDR,W ;Data Memory
MOVWF  EEADR         ;Address to write
MOVF   DATA_EE_DATA,W ;Data Memory Value
MOVWF  EEDATA        ;to write
BSF    STATUS,RP0    ;Bank 3
BCF    EECON1,EEPGD ;Point to DATA
                           ;memory
BSF    EECON1,WREN   ;Enable writes

BCF    INTCON,GIE    ;Disable INTs.
MOVLW  55h           ;
MOVWF  EECON2        ;Write 55h
MOVLW  AAh           ;
MOVWF  EECON2        ;Write AAh
BSF    EECON1,WR     ;Set WR bit to
                           ;begin write
BSF    INTCON,GIE    ;Enable INTs.
BCF    EECON1,WREN   ;Disable writes
```

PIC16F87XA

3.5 Reading Flash Program Memory

To read a program memory location, the user must write 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 program memory Flash controller will use the next two instruction cycles to read the data. This causes these two instructions immediately follow-

ing the “BSF EECON1,RD” instruction to be ignored. The data is available in the very next cycle in the EEDATA and EEDATH registers; therefore, it can be read as two bytes in the following instructions. EEDATA and EEDATH registers will hold this value until another read or until it is written to by the user (during a write operation).

EXAMPLE 3-3: FLASH PROGRAM READ

```
BSF STATUS, RP1 ;
BCF STATUS, RP0 ; Bank 2
MOVLW MS_PROG_EE_ADDR ;
MOVWF EEADRH ; MS Byte of Program Address to read
MOVLW LS_PROG_EE_ADDR ;
MOVWF EEADR ; LS Byte of Program Address to read
BSF STATUS, RP0 ; Bank 3
BSF EECON1, EEPGD ; Point to PROGRAM memory
BSF EECON1, RD ; EE Read

;
Required Sequence ;
NOP ; Any instructions here are ignored as program
NOP ; memory is read in second cycle after BSF EECON1,RD
;
BCF STATUS, RP0 ; Bank 2
MOVF EEDATA, W ; W = LS Byte of Program EEDATA
MOVWF DATAL ;
MOVF EEDATH, W ; W = MS Byte of Program EEDATA
MOVWF DATAH ;
```

3.6 Writing to Flash Program Memory

Flash program memory may only be written to if the destination address is in a segment of memory that is not write-protected, as defined in bits WRT1:WRT0 of the device configuration word (Register 14-1). Flash program memory must be written in four-word blocks. A block consists of four words with sequential addresses, with a lower boundary defined by an address, where $EEADR<1:0> = 00$. At the same time, all block writes to program memory are done as erase and write operations. The write operation is edge-aligned and cannot occur across boundaries.

To write program data, it must first be loaded into the buffer registers (see Figure 3-1). This is accomplished by first writing the destination address to $EEADR$ and $EEADRH$ and then writing the data to $EEDATA$ and $EEDATH$. After the address and data have been set up, then the following sequence of events must be executed:

1. Set the $EEPGRD$ control bit ($EECON1<7>$).
2. Write 55h, then AAh, to $EECON2$ (Flash programming sequence).
3. Set the WR control bit ($EECON1<1>$).

All four buffer register locations **MUST** be written to with correct data. If only one, two or three words are being written to in the block of four words, then a read from the program memory location(s) not being written to must be performed. This takes the data from the program location(s) not being written and loads it into the $EEDATA$ and $EEDATH$ registers. Then the sequence of events to transfer data to the buffer registers must be executed.

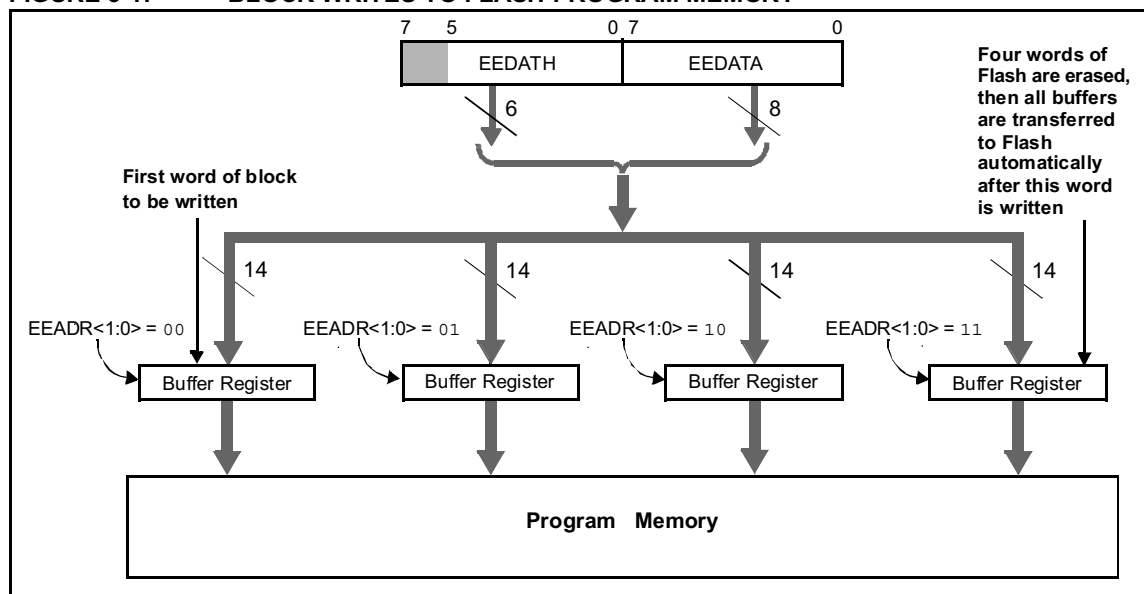
To transfer data from the buffer registers to the program memory, the $EEADR$ and $EEADRH$ must point to the last location in the four-word block ($EEADR<1:0> = 11$). Then the following sequence of events must be executed:

1. Set the $EEPGRD$ control bit ($EECON1<7>$).
2. Write 55h, then AAh, to $EECON2$ (Flash programming sequence).
3. Set control bit WR ($EECON1<1>$) to begin the write operation.

The user must follow the same specific sequence to initiate the write for each word in the program block, writing each program word in sequence (00, 01, 10, 11). When the write is performed on the last word ($EEADR<1:0> = 11$), the block of four words are automatically erased and the contents of the buffer registers are written into the program memory.

After the “ $BSF\ EECON1,WR$ ” instruction, the processor requires two cycles to set up the erase/write operation. The user must place two NOP instructions after the WR bit is set. Since data is being written to buffer registers, the writing of the first three words of the block appears to occur immediately. The processor will halt internal operations for the typical 4 ms, only during the cycle in which the erase takes place (i.e., the last word of the four-word block). This is not Sleep mode as the clocks and peripherals will continue to run. After the write cycle, the processor will resume operation with the third instruction after the $EECON1$ write instruction. If the sequence is performed to any other location, the action is ignored.

FIGURE 3-1: BLOCK WRITES TO FLASH PROGRAM MEMORY



PIC16F87XA

An example of the complete four-word write sequence is shown in Example 3-4. The initial address is loaded into the EEADRH:EEADR register pair; the four words of data are loaded using indirect addressing.

EXAMPLE 3-4: WRITING TO FLASH PROGRAM MEMORY

```
; This write routine assumes the following:
;
; 1. A valid starting address (the least significant bits = '00') is loaded in ADDRH:ADDRL
; 2. The 8 bytes of data are loaded, starting at the address in DATADDR
; 3. ADDRH, ADDRL and DATADDR are all located in shared data memory 0x70 - 0x7f
;
      BSF    STATUS,RP1      ;
      BCF    STATUS,RP0      ; Bank 2
      MOVF   ADDRH,W         ; Load initial address
      MOVWF  EEADRH         ;
      MOVF   ADDRL,W        ;
      MOVWF  EEADR          ;
      MOVF   DATAADDR,W    ; Load initial data address
      MOVWF  FSR            ;
LOOP   MOVF   INDF,W         ; Load first data byte into lower
      MOVWF  EEDATA        ;
      INCF   FSR,F          ; Next byte
      MOVF   INDF,W        ; Load second data byte into upper
      MOVWF  EEDATH        ;
      INCF   FSR,F          ;
      BSF    STATUS,RP0      ; Bank 3
      BSF    EECON1,EEPGD    ; Point to program memory
      BSF    EECON1,WREN     ; Enable writes
      BCF    INTCON,GIE     ; Disable interrupts (if using)
      MOVLW  55h             ; Start of required write sequence:
      MOVWF  EECON2         ; Write 55h
      MOVLW  AAh             ;
      MOVWF  EECON2         ; Write AAh
      BSF    EECON1,WR       ; Set WR bit to begin write
      NOP                    ; Any instructions here are ignored as processor
      NOP                    ; halts to begin write sequence
      NOP                    ; processor will stop here and wait for write complete
      NOP                    ; after write processor continues with 3rd instruction
      BCF    EECON1,WREN     ; Disable writes
      BSF    INTCON,GIE     ; Enable interrupts (if using)
      BCF    STATUS,RP0      ; Bank 2
      INCF   EEADR,F        ; Increment address
      MOVF   EEADR,W        ; Check if lower two bits of address are '00'
      ANDLW  0x03           ; Indicates when four words have been programmed
      XORLW  0x03           ;
      BTFSC  STATUS,Z       ; Exit if more than four words,
      GOTO   LOOP           ; Continue if less than four words
```


3.7 Protection Against Spurious Write

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

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

3.8 Operation During Code-Protect

When the data EEPROM is code-protected, the microcontroller can read and write to the EEPROM normally. However, all external access to the EEPROM is disabled. External write access to the program memory is also disabled.

When program memory is code-protected, the microcontroller can read and write to program memory normally, as well as execute instructions. Writes by the device may be selectively inhibited to regions of the memory depending on the setting of bits WR1:WR0 of the configuration word (see **Section 14.1 “Configuration Bits”** for additional information). External access to the memory is also disabled.

TABLE 3-1: REGISTERS/BITS ASSOCIATED WITH DATA EEPROM AND FLASH PROGRAM MEMORIES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other Resets	
10Ch	EEDATA	EEPROM/Flash Data Register Low Byte								xxxx xxxx	uuuu uuuu	
10Dh	EEADR	EEPROM/Flash Address Register Low Byte								xxxx xxxx	uuuu uuuu	
10Eh	EEDATH	—	—	EEPROM/Flash Data Register High Byte						xxxx xxxx	---0 q000	
10Fh	EEADRH	—	—	—	EEPROM/Flash Address Register High Byte						xxxx xxxx	---- ----
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	---0 q000	
18Dh	EECON2	EEPROM Control Register 2 (not a physical register)								---- ----	---- ----	
0Dh	PIR2	—	CMIF	—	EEIF	BCLIF	—	—	CCP2IF	-0-0 0--0	-0-0 0--0	
8Dh	PIE2	—	CMIE	—	EEIE	BCLIE	—	—	CCP2IE	-0-0 0--0	-0-0 0--0	

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0', q = value depends upon condition.
Shaded cells are not used by data EEPROM or Flash program memory.

PIC16F87XA

NOTES:

4.0 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).

4.1 PORTA and the TRISA Register

PORTA is a 6-bit wide, bidirectional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, the value is modified and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CK1 pin. The RA4/T0CK1 pin is a Schmitt Trigger input and an open-drain output. All other PORTA pins have TTL input levels and full CMOS output drivers.

Other PORTA pins are multiplexed with analog inputs and the analog VREF input for both the A/D converters and the comparators. The operation of each pin is selected by clearing/setting the appropriate control bits in the ADCON1 and/or CMCON registers.

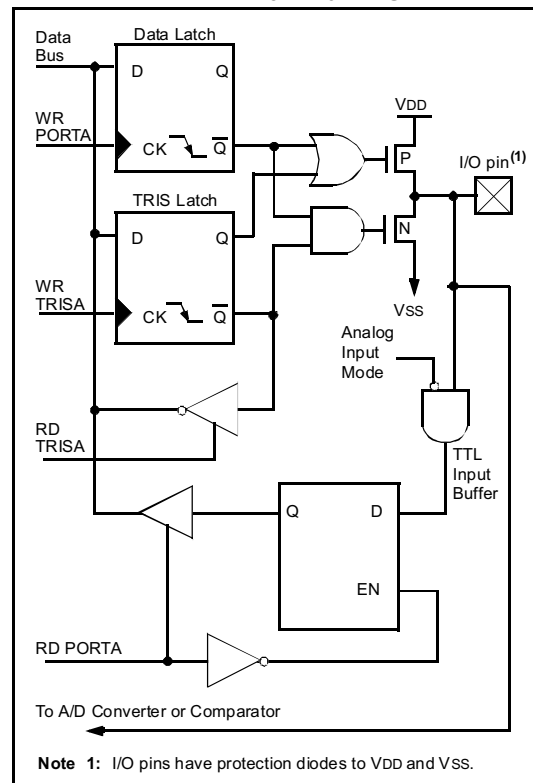
Note: On a Power-on Reset, these pins are configured as analog inputs and read as '0'. The comparators are in the off (digital) state.

The TRISA register controls the direction of the port 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 4-1: INITIALIZING PORTA

```
BCF STATUS, RP0 ;
BCF STATUS, RP1 ; Bank0
CLRF PORTA ; Initialize PORTA by
; clearing output
; data latches
BSF STATUS, RP0 ; Select Bank 1
MOVLW 0x06 ; Configure all pins
MOVWF ADCON1 ; as digital inputs
MOVLW 0xCF ; Value used to
; initialize data
; direction
MOVWF TRISA ; Set RA<3:0> as inputs
; RA<5:4> as outputs
; TRISA<7:6>are always
; read as '0'.
```

FIGURE 4-1: BLOCK DIAGRAM OF RA3:RA0 PINS



PIC16F87XA

TABLE 4-1: PORTA FUNCTIONS

Name	Bit#	Buffer	Function
RA0/AN0	bit 0	TTL	Input/output or analog input.
RA1/AN1	bit 1	TTL	Input/output or analog input.
RA2/AN2/VREF-/CVREF	bit 2	TTL	Input/output or analog input or VREF- or CVREF.
RA3/AN3/VREF+	bit 3	TTL	Input/output or analog input or VREF+.
RA4/T0CKI/C1OUT	bit 4	ST	Input/output or external clock input for Timer0 or comparator output. Output is open-drain type.
RA5/AN4/ \overline{SS} /C2OUT	bit 5	TTL	Input/output or analog input or slave select input for synchronous serial port or comparator output.

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

TABLE 4-2: 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
9Ch	CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111	0000 0111
9Dh	CVRCON	CVREN	CVROE	CVRR	—	CVR3	CVR2	CVR1	CVR0	000- 0000	000- 0000
9Fh	ADCON1	ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0	00-- 0000	00-- 0000

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

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

PIC16F87XA

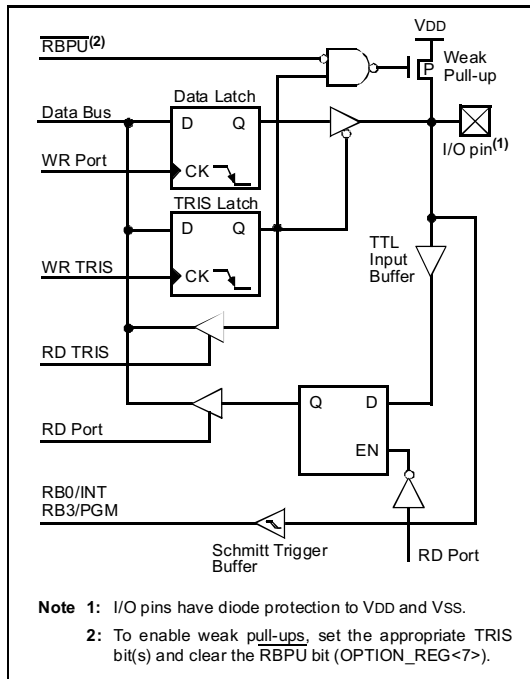
4.2 PORTB and the TRISB Register

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

Three pins of PORTB are multiplexed with the In-Circuit Debugger and Low-Voltage Programming function: RB3/PGM, RB6/PGC and RB7/PGD. The alternate functions of these pins are described in **Section 14.0 "Special Features of the CPU"**.

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit RBPU (OPTION_REG<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

FIGURE 4-4: BLOCK DIAGRAM OF RB3:RB0 PINS



Four of the PORTB pins, RB7:RB4, have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt-on-change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'ed together to generate the RB port change interrupt with flag bit RBIF (INTCON<0>).

This interrupt can wake the device from Sleep. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- Any read or write of PORTB. This will end the mismatch condition.
- Clear flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.

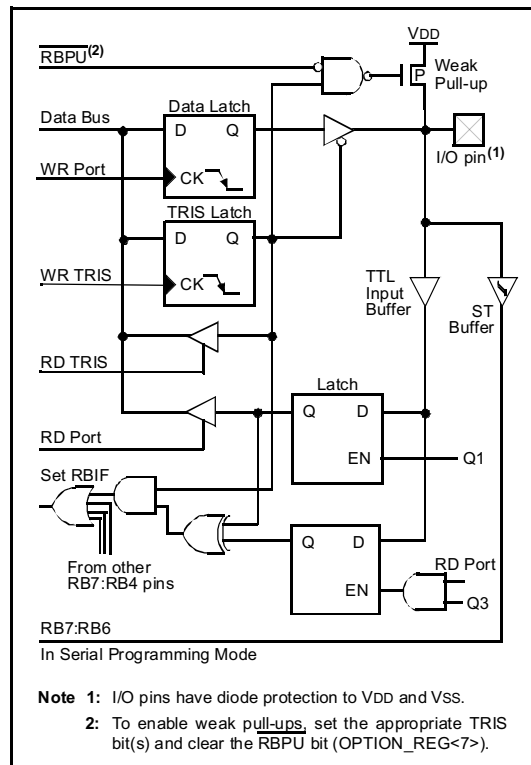
The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

This interrupt-on-mismatch feature, together with software configurable pull-ups on these four pins, allow easy interface to a keypad and make it possible for wake-up on key depression. Refer to the application note, AN552, "Implementing Wake-up on Key Stroke" (DS00552).

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

RB0/INT is discussed in detail in **Section 14.11.1 "INT Interrupt"**.

FIGURE 4-5: BLOCK DIAGRAM OF RB7:RB4 PINS



PIC16F87XA

TABLE 4-3: PORTB FUNCTIONS

Name	Bit#	Buffer	Function
RB0/INT	bit 0	TTL/ST ⁽¹⁾	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit 1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit 2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3/PGM ⁽³⁾	bit 3	TTL	Input/output pin or programming pin in LVP mode. Internal software programmable weak pull-up.
RB4	bit 4	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB5	bit 5	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB6/PGC	bit 6	TTL/ST ⁽²⁾	Input/output pin (with interrupt-on-change) or in-circuit debugger pin. Internal software programmable weak pull-up. Serial programming clock.
RB7/PGD	bit 7	TTL/ST ⁽²⁾	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 or in-circuit debugger.

3: Low-Voltage ICSP Programming (LVP) is enabled by default which disables the RB3 I/O function. LVP must be disabled to enable RB3 as an I/O pin and allow maximum compatibility to the other 28-pin and 40-pin mid-range devices.

TABLE 4-4: 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.

PIC16F87XA

4.3 PORTC and the TRISC Register

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

PORTC is multiplexed with several peripheral functions (Table 4-5). PORTC pins have Schmitt Trigger input buffers.

When the I²C module is enabled, the PORTC<4:3> pins can be configured with normal I²C 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 the destination, should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

FIGURE 4-6: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<2:0>, RC<7:5>

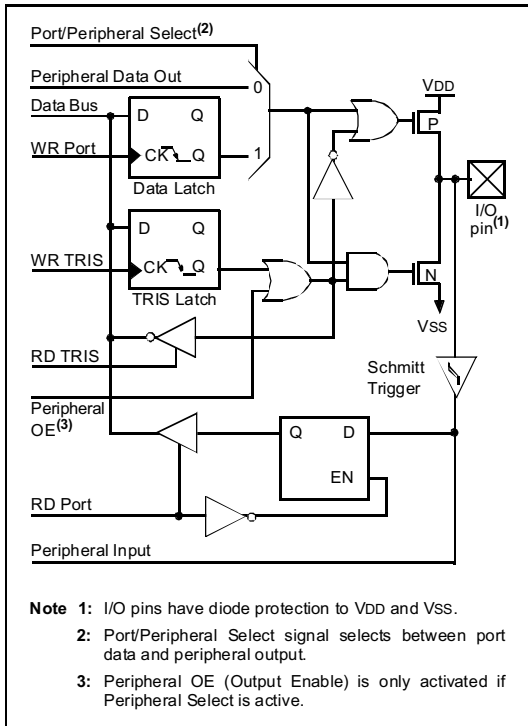
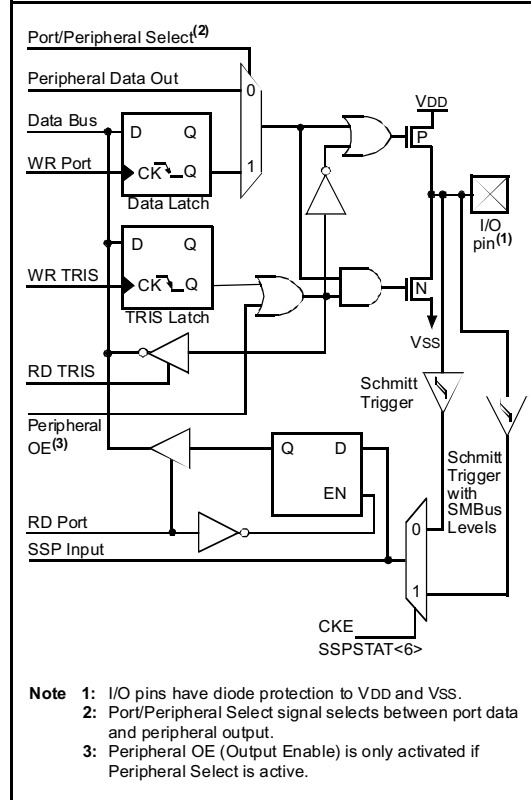


FIGURE 4-7: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<4:3>



PIC16F87XA

TABLE 4-5: PORTC FUNCTIONS

Name	Bit#	Buffer Type	Function
RC0/T1OSO/T1CKI	bit 0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input.
RC1/T1OSI/CCP2	bit 1	ST	Input/output port pin or Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	bit 2	ST	Input/output port pin or Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	bit 3	ST	RC3 can also be the synchronous serial clock for both SPI and I ² C modes.
RC4/SDI/SDA	bit 4	ST	RC4 can also be the SPI data in (SPI mode) or data I/O (I ² C mode).
RC5/SDO	bit 5	ST	Input/output port pin or Synchronous Serial Port data output.
RC6/TX/CK	bit 6	ST	Input/output port pin or USART asynchronous transmit or synchronous clock.
RC7/RX/DT	bit 7	ST	Input/output port pin or USART asynchronous receive or synchronous data.

Legend: ST = Schmitt Trigger input

TABLE 4-6: 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

PIC16F87XA

4.4 PORTD and TRISD Registers

Note: PORTD and TRISD are not implemented on 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 4-8: PORTD BLOCK DIAGRAM (IN I/O PORT MODE)

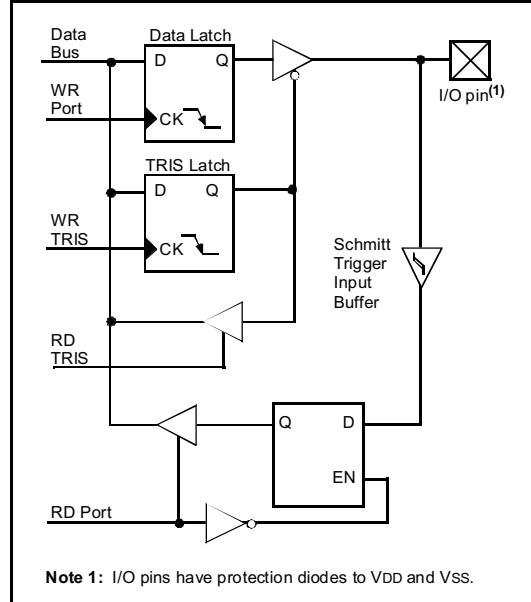


TABLE 4-7: PORTD FUNCTIONS

Name	Bit#	Buffer Type	Function
RD0/PSP0	bit 0	ST/TTL ⁽¹⁾	Input/output port pin or Parallel Slave Port bit 0.
RD1/PSP1	bit 1	ST/TTL ⁽¹⁾	Input/output port pin or Parallel Slave Port bit 1.
RD2/PSP2	bit 2	ST/TTL ⁽¹⁾	Input/output port pin or Parallel Slave Port bit 2.
RD3/PSP3	bit 3	ST/TTL ⁽¹⁾	Input/output port pin or Parallel Slave Port bit 3.
RD4/PSP4	bit 4	ST/TTL ⁽¹⁾	Input/output port pin or Parallel Slave Port bit 4.
RD5/PSP5	bit 5	ST/TTL ⁽¹⁾	Input/output port pin or Parallel Slave Port bit 5.
RD6/PSP6	bit 6	ST/TTL ⁽¹⁾	Input/output port pin or Parallel Slave Port bit 6.
RD7/PSP7	bit 7	ST/TTL ⁽¹⁾	Input/output port pin or Parallel Slave Port bit 7.

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 4-8: 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.

PIC16F87XA

4.5 PORTE and TRISE Register

Note: PORTE and TRISE are not implemented on 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.

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

Register 4-1 shows the TRISE register which also controls the Parallel Slave Port operation.

PORTE pins are multiplexed with analog inputs. When selected for 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 and read as '0'.

FIGURE 4-9: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)

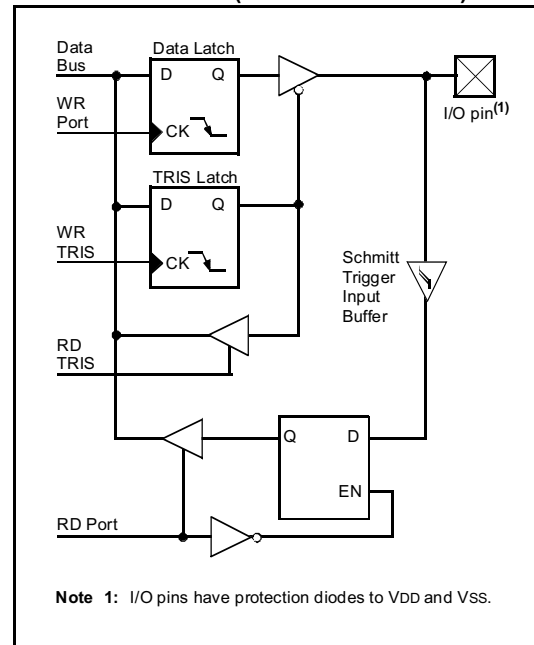


TABLE 4-9: PORTE FUNCTIONS

Name	Bit#	Buffer Type	Function
RE0/ \overline{RD} /AN5	bit 0	ST/TTL ⁽¹⁾	I/O port pin or read control input in Parallel Slave Port mode or analog input: \overline{RD} 1 = Idle 0 = Read operation. Contents of PORTD register are output to PORTD I/O pins (if chip selected).
RE1/ \overline{WR} /AN6	bit 1	ST/TTL ⁽¹⁾	I/O port pin or write control input in Parallel Slave Port mode or analog input: \overline{WR} 1 = Idle 0 = Write operation. Value of PORTD I/O pins is latched into PORTD register (if chip selected).
RE2/ \overline{CS} /AN7	bit 2	ST/TTL ⁽¹⁾	I/O 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.

PIC16F87XA

TABLE 4-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

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
09h	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction bits			0000 -111	0000 -111
9Fh	ADCON1	ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0	00-- 0000	00-- 0000

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

REGISTER 4-1: TRISE REGISTER (ADDRESS 89h)

R-0	R-0	R/W-0	R/W-0	U-0	R/W-1	R/W-1	R/W-1
IBF	OBF	IBOV	PSPMODE	—	Bit 2	Bit 1	Bit 0
bit 7				bit 0			

Parallel Slave Port Status/Control Bits:

- bit 7 **IBF:** Input Buffer Full Status bit
1 = A word has been received and is waiting to be read by the CPU
0 = No word has been received
- bit 6 **OBF:** Output Buffer Full Status bit
1 = The output buffer still holds a previously written word
0 = The output buffer has been read
- bit 5 **IBOV:** Input Buffer Overflow Detect bit (in Microprocessor mode)
1 = A write occurred when a previously input word has not been read (must be cleared in software)
0 = No overflow occurred
- bit 4 **PSPMODE:** Parallel Slave Port Mode Select bit
1 = PORTD functions in Parallel Slave Port mode
0 = PORTD functions in general purpose I/O mode
- bit 3 **Unimplemented:** Read as '0'
- PORTE Data Direction Bits:**
- bit 2 **Bit 2:** Direction Control bit for pin RE2/ $\overline{\text{CS}}$ /AN7
1 = Input
0 = Output
- bit 1 **Bit 1:** Direction Control bit for pin RE1/ $\overline{\text{WR}}$ /AN6
1 = Input
0 = Output
- bit 0 **Bit 0:** Direction Control bit for pin RE0/ $\overline{\text{RD}}$ /AN5
1 = Input
0 = Output

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

4.6 Parallel Slave Port

The Parallel Slave Port (PSP) is not implemented on the PIC16F873A or PIC16F876A.

PORTD operates as an 8-bit wide Parallel Slave Port, or microprocessor port, when control bit PSMODE (TRISE<4>) is set. In Slave mode, it is asynchronously readable and writable by the external world through RD control input pin, RE0/RD/AN5, and WR control input pin, RE1/WR/AN6.

The PSP 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 PSMODE enables port pin RE0/RD/AN5 to be the RD input, RE1/WR/AN6 to be the WR input and RE2/CS/AN7 to be the CS (Chip Select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (set). The A/D port configuration bits, PCFG3:PCFG0 (ADCON1<3:0>), must be set to configure pins RE2:RE0 as digital I/O.

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

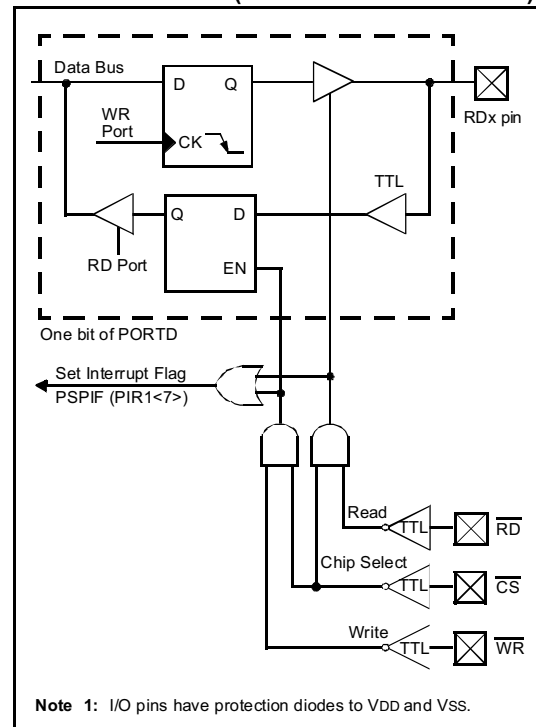
A write to the PSP occurs when both the CS and WR lines are first detected low. When either the CS or WR lines become high (level triggered), the Input Buffer Full (IBF) status flag bit (TRISE<7>) is set on the Q4 clock cycle, following the next Q2 cycle, to signal the write is complete (Figure 4-11). The interrupt flag bit, PSPIF (PIR1<7>), is also set on the same Q4 clock cycle. IBF can only be cleared by reading the PORTD input latch. The Input Buffer Overflow (IBOV) status flag bit (TRISE<5>) is set if a second write to the PSP is attempted when the previous byte has not been read out of the buffer.

A read from the PSP occurs when both the CS and RD lines are first detected low. The Output Buffer Full (OBF) status flag bit (TRISE<6>) is cleared immediately (Figure 4-12), indicating that the PORTD latch is waiting to be read by the external bus. When either the CS or RD pin becomes high (level triggered), the interrupt flag bit PSPIF is set on the Q4 clock cycle, following the next Q2 cycle, indicating that the read is complete. OBF remains low until data is written to PORTD by the user firmware.

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

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

FIGURE 4-10: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)



PIC16F87XA

FIGURE 4-11: PARALLEL SLAVE PORT WRITE WAVEFORMS

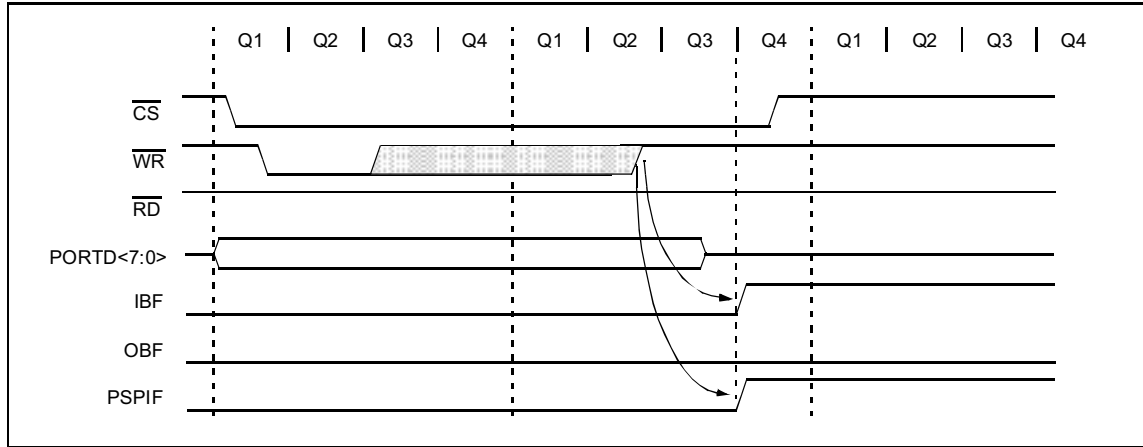


FIGURE 4-12: PARALLEL SLAVE PORT READ WAVEFORMS

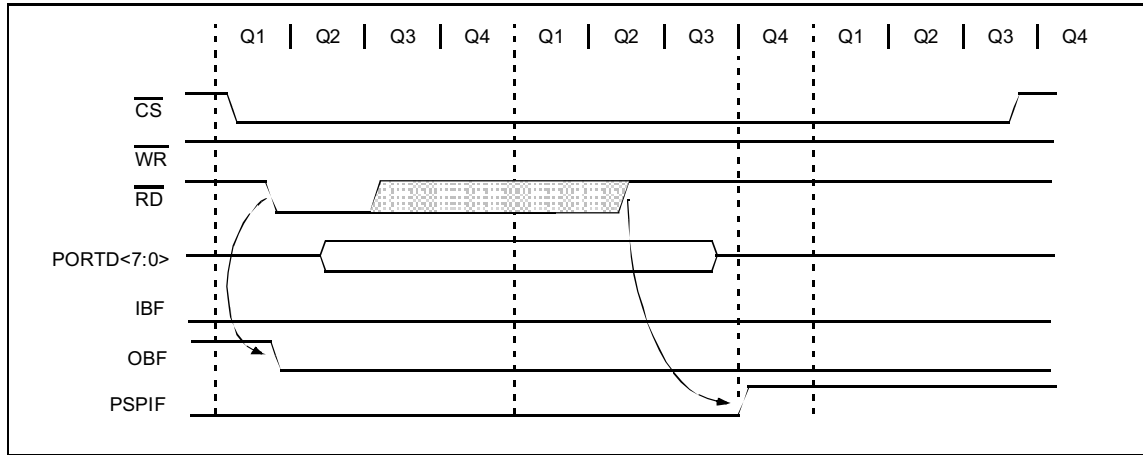


TABLE 4-11: REGISTERS ASSOCIATED WITH PARALLEL SLAVE PORT

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 ⁽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
9Fh	ADCON1	ADFM	ADCS2	-	-	PCFG3	PCFG2	PCFG1	PCFG0	00-- 0000	00-- 0000

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

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F873A/876A; always maintain these bits clear.

5.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

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

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

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

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

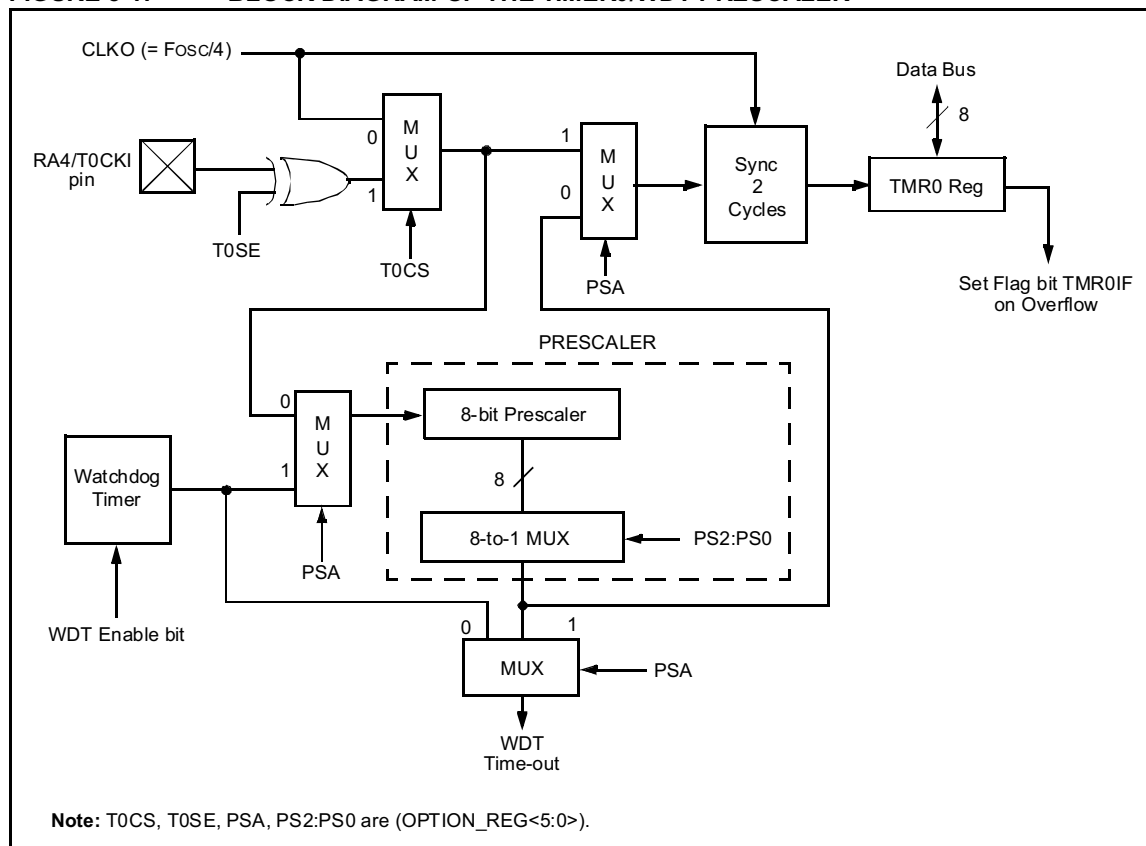
Counter mode is selected by setting bit T0CS (OPTION_REG<5>). In Counter mode, Timer0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit, T0SE (OPTION_REG<4>). Clearing bit T0SE selects the rising edge. Restrictions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

The prescaler is mutually exclusively shared between the Timer0 module and the Watchdog Timer. The prescaler is not readable or writable. **Section 5.3 “Prescaler”** details the operation of the prescaler.

5.1 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h. This overflow sets bit TMR0IF (INTCON<2>). The interrupt can be masked by clearing bit TMR0IE (INTCON<5>). Bit TMR0IF must be cleared in software by the Timer0 module Interrupt Service Routine before re-enabling this interrupt. The TMR0 interrupt cannot awaken the processor from Sleep since the timer is shut-off during Sleep.

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



PIC16F87XA

5.2 Using Timer0 with an External Clock

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

5.3 Prescaler

There is only one prescaler available which is mutually exclusively shared between the Timer0 module and the Watchdog Timer. A prescaler assignment for the

Timer0 module means that there is no prescaler for the Watchdog Timer and vice versa. This prescaler is not readable or writable (see Figure 5-1).

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

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

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

REGISTER 5-1: OPTION_REG REGISTER

	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	RBP	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
	bit 7							bit 0

bit 7	RBP
bit 6	INTEDG
bit 5	T0CS: TMR0 Clock Source Select bit 1 = Transition on T0CKI pin 0 = Internal instruction cycle clock (CLKO)
bit 4	T0SE: TMR0 Source Edge Select bit 1 = Increment on high-to-low transition on T0CKI pin 0 = Increment on low-to-high transition on T0CKI pin
bit 3	PSA: Prescaler Assignment bit 1 = Prescaler is assigned to the WDT 0 = Prescaler is assigned to the Timer0 module
bit 2-0	PS2:PS0: Prescaler Rate Select bits

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

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

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

PIC16F87XA

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 Register								xxxx xxxx	uuuu uuuu
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
81h,181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

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

PIC16F87XA

NOTES:

6.0 TIMER1 MODULE

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

Timer1 can operate in one of two modes:

- As a Timer
- As a Counter

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

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

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

Timer1 also has an internal "Reset input". This Reset can be generated by either of the two CCP modules (Section 8.0 "Capture/Compare/PWM Modules"). Register 6-1 shows the Timer1 Control register.

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

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

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

	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
	bit 7							bit 0
bit 7-6	Unimplemented: Read as '0'							
bit 5-4	T1CKPS1:T1CKPS0: Timer1 Input Clock Prescale Select bits							
	11 = 1:8 prescale value							
	10 = 1:4 prescale value							
	01 = 1:2 prescale value							
	00 = 1:1 prescale value							
bit 3	T1OSCEN: Timer1 Oscillator Enable Control bit							
	1 = Oscillator is enabled							
	0 = Oscillator is shut-off (the oscillator inverter is turned off to eliminate power drain)							
bit 2	T1SYNC: Timer1 External Clock Input Synchronization Control bit							
	<u>When TMR1CS = 1:</u>							
	1 = Do not synchronize external clock input							
	0 = Synchronize external clock input							
	<u>When TMR1CS = 0:</u>							
	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							

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

PIC16F87XA

6.1 Timer1 Operation in Timer Mode

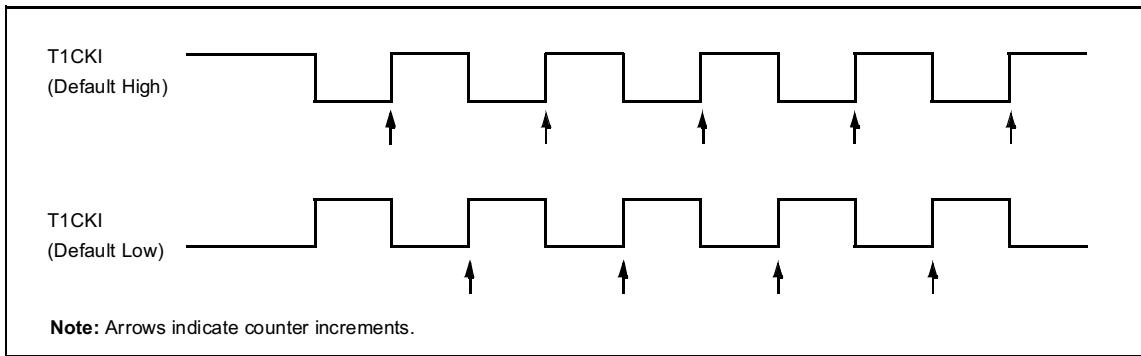
Timer mode is selected by clearing the TMR1CS (T1CON<1>) bit. In this mode, the input clock to the timer is Fosc/4. The synchronize control bit, $\overline{T1SYNC}$ (T1CON<2>), has no effect since the internal clock is always in sync.

6.2 Timer1 Counter Operation

Timer1 may operate in either a Synchronous, or an Asynchronous mode, depending on the setting of the TMR1CS bit.

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

FIGURE 6-1: TIMER1 INCREMENTING EDGE



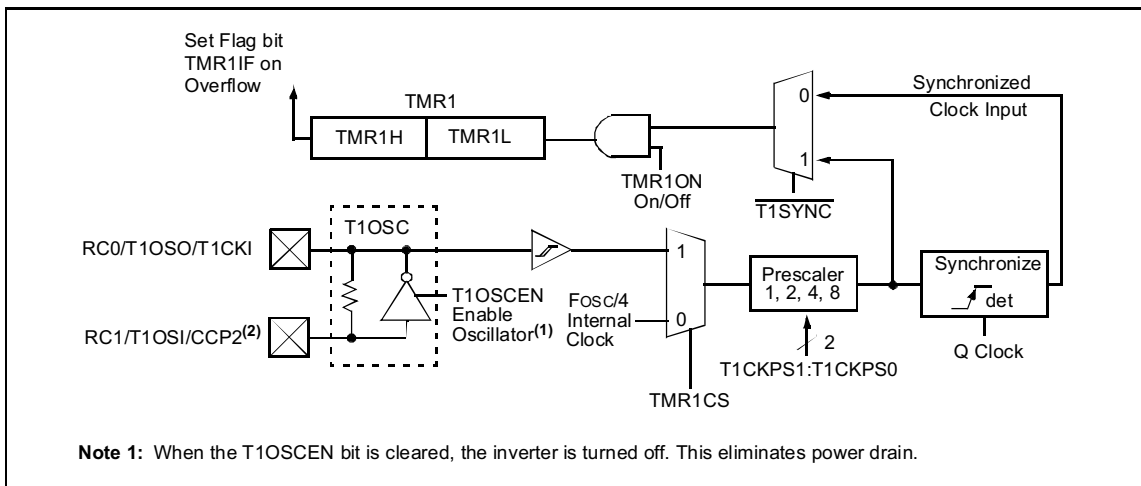
6.3 Timer1 Operation in Synchronized Counter Mode

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

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

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

FIGURE 6-2: TIMER1 BLOCK DIAGRAM



6.4 Timer1 Operation in Asynchronous Counter Mode

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

In Asynchronous Counter mode, Timer1 cannot be used as a time base for capture or compare operations.

6.4.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

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

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

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

6.5 Timer1 Oscillator

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

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

TABLE 6-1: CAPACITOR SELECTION FOR THE TIMER1 OSCILLATOR

Osc Type	Freq.	C1	C2
LP	32 kHz	33 pF	33 pF
	100 kHz	15 pF	15 pF
	200 kHz	15 pF	15 pF
These values are for design guidance only.			
Crystals Tested:			
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.

2: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

6.6 Resetting Timer1 Using a CCP Trigger Output

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

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

Timer1 must be configured for either Timer or Synchronized Counter mode to take advantage of this feature. If Timer1 is running in Asynchronous Counter mode, this Reset operation may not work.

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

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

PIC16F87XA

6.7 Resetting of Timer1 Register Pair (TMR1H, TMR1L)

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

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

6.8 Timer1 Prescaler

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

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	TMR0IE	INTE	RBIE	TMR0IF	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	T1SYN \bar{C}	TMR1CS	TMR1ON	--00 0000	--uu uuuu

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

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

PIC16F87XA

7.0 TIMER2 MODULE

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

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

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

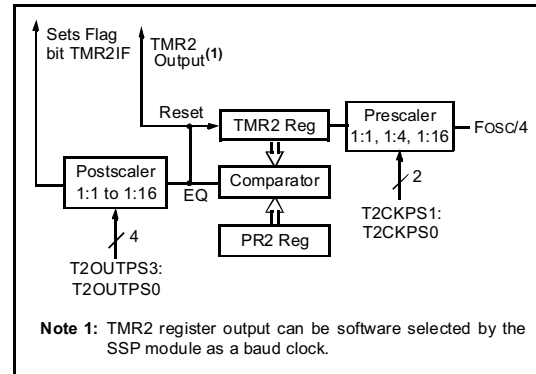
The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit, TMR2IF (PIR1<1>)).

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

Register 7-1 shows the Timer2 Control register.

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

FIGURE 7-1: TIMER2 BLOCK DIAGRAM



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

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

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

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

PIC16F87XA

7.1 Timer2 Prescaler and Postscaler

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

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

TMR2 is not cleared when T2CON is written.

7.2 Output of TMR2

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

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	TMR0IE	INTE	RBIE	TMR0IF	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
11h	TMR2	Timer2 Module's Register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
92h	PR2	Timer2 Period Register								1111 1111	1111 1111

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

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

8.0 CAPTURE/COMPARE/PWM MODULES

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

- 16-bit Capture register
- 16-bit Compare register
- PWM Master/Slave Duty Cycle register

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

CCP1 Module:

Capture/Compare/PWM Register 1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (high byte). The CCP1CON register controls the operation of CCP1. The special event trigger is generated by a compare match and will reset Timer1.

CCP2 Module:

Capture/Compare/PWM Register 2 (CCPR2) is comprised of two 8-bit registers: CCPR2L (low byte) and CCPR2H (high byte). The CCP2CON register controls the operation of CCP2. The special event trigger is generated by a compare match and will reset Timer1 and start an A/D conversion (if the A/D module is enabled).

Additional information on CCP modules is available in the PICmicro® Mid-Range MCU Family Reference Manual (DS33023) and in application note AN594, "Using the CCP Module(s)" (DS00594).

TABLE 8-1: CCP MODE – TIMER RESOURCES REQUIRED

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

TABLE 8-2: INTERACTION OF TWO CCP MODULES

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

PIC16F87XA

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

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0	
bit 7								bit 0

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

bit 5-4 **CCPxX:CCPxY:** PWM Least Significant bits

Capture mode:

Unused.

Compare mode:

Unused.

PWM mode:

These bits are the two LSBs of the PWM duty cycle. The eight MSBs are found in CCPxL.

bit 3-0 **CCPxM3:CCPxM0:** CCPx Mode Select bits

0000 = Capture/Compare/PWM disabled (resets CCPx module)

0100 = Capture mode, every falling edge

0101 = Capture mode, every rising edge

0110 = Capture mode, every 4th rising edge

0111 = Capture mode, every 16th rising edge

1000 = Compare mode, set output on match (CCPxIF bit is set)

1001 = Compare mode, clear output on match (CCPxIF bit is set)

1010 = Compare mode, generate software interrupt on match (CCPxIF bit is set, CCPx pin is unaffected)

1011 = Compare mode, trigger special event (CCPxIF bit is set, CCPx pin is unaffected); CCP1 resets TMR1; CCP2 resets TMR1 and starts an A/D conversion (if A/D module is enabled)

11xx = PWM mode

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

8.1 Capture Mode

In Capture mode, CCP1H:CCP1L captures the 16-bit value of the TMR1 register when an event occurs on pin RC2/CCP1. An event is defined as one of the following:

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

The type of event is configured by control bits, CCP1M3:CCP1M0 (CCPxCON<3:0>). When a capture is made, the interrupt request flag bit, CCP1IF (PIR1<2>), is set. The interrupt flag must be cleared in software. If another capture occurs before the value in register CCP1 is read, the old captured value is overwritten by the new value.

8.1.1 CCP PIN CONFIGURATION

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

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

8.1.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode, or Synchronized Counter mode, for the CCP module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

8.1.3 SOFTWARE INTERRUPT

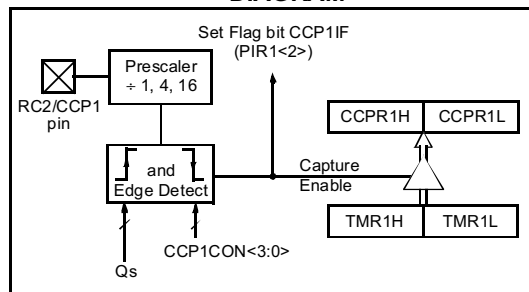
When the Capture mode is changed, a false capture interrupt may be generated. The user should keep bit CCP1IE (PIE1<2>) clear to avoid false interrupts and should clear the flag bit, CCP1IF, following any such change in operating mode.

8.1.4 CCP PRESCALER

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

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

FIGURE 8-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



EXAMPLE 8-1: CHANGING BETWEEN CAPTURE PRESCALERS

```

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

PIC16F87XA

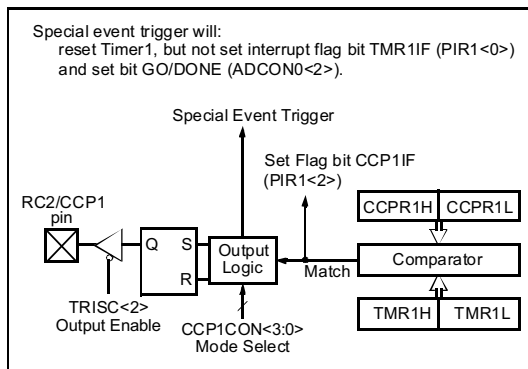
8.2 Compare Mode

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

- Driven high
- Driven low
- Remains unchanged

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

FIGURE 8-2: COMPARE MODE OPERATION BLOCK DIAGRAM



8.2.1 CCP PIN CONFIGURATION

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

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

8.2.2 TIMER1 MODE SELECTION

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

8.2.3 SOFTWARE INTERRUPT MODE

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

8.2.4 SPECIAL EVENT TRIGGER

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

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

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

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

PIC16F87XA

8.3.3 SETUP 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 8-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHz

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12kHz	156.3 kHz	208.3 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	0xFFh	0xFFh	0xFFh	0x3Fh	0x1Fh	0x17h
Maximum Resolution (bits)	10	10	10	8	7	5.5

TABLE 8-4: 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	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- ---0	---- ---0
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- ---0	---- ---0
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh	CCPR2L	Capture/Compare/PWM Register 2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register 2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

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

Note 1: The PSP is not implemented on 28-pin devices; always maintain these bits clear.

PIC16F87XA

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	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBFIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- --0	---- --0
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
11h	TMR2	Timer2 Module's Register								0000 0000	0000 0000
92h	PR2	Timer2 Module's Period Register								1111 1111	1111 1111
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh	CCPR2L	Capture/Compare/PWM Register 2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register 2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

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

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

PIC16F87XA

NOTES:

PIC16F87XA

9.3.1 REGISTERS

The MSSP module has four registers for SPI mode operation. These are:

- MSSP Control Register (SSPCON)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer Register (SSPBUF)
- MSSP Shift Register (SSPSR) – Not directly accessible

SSPCON and SSPSTAT are the control and status registers in SPI mode operation. The SSPCON register is readable and writable. The lower six bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not double-buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

REGISTER 9-1: SSPSTAT: MSSP STATUS REGISTER (SPI MODE) (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
							bit 0

- 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.
- bit 6 **CKE:** SPI Clock Select bit
 1 = Transmit occurs on transition from active to Idle clock state
 0 = Transmit occurs on transition from Idle to active clock state
Note: Polarity of clock state is set by the CKP bit (SSPCON1<4>).
- bit 5 **D/A:** Data/Address bit
 Used in I²C mode only.
- bit 4 **P:** Stop bit
 Used in I²C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.
- bit 3 **S:** Start bit
 Used in I²C mode only.
- bit 2 **R/W:** Read/Write bit information
 Used in I²C mode only.
- bit 1 **UA:** Update Address bit
 Used in I²C mode only.
- bit 0 **BF:** Buffer Full Status bit (Receive mode only)
 1 = Receive complete, SSPBUF is full
 0 = Receive not complete, SSPBUF is empty

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

PIC16F87XA

REGISTER 9-2: SSPCON1: MSSP CONTROL REGISTER 1 (SPI MODE) (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	
bit 7								bit 0

- bit 7 **WCOL:** Write Collision Detect bit (Transmit mode only)
 1 = The SSPBUF register is written while it is still transmitting the previous word. (Must be cleared in software.)
 0 = No collision
- bit 6 **SSPOV:** Receive Overflow Indicator bit
SPI Slave mode:
 1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. (Must be cleared in software.)
 0 = No overflow
Note: In Master mode, the overflow bit is not set, since each new reception (and transmission) is initiated by writing to the SSPBUF register.
- bit 5 **SSPEN:** Synchronous Serial Port Enable bit
 1 = Enables serial port and configures SCK, SDO, SDI, and \overline{SS} as serial port pins
 0 = Disables serial port and configures these pins as I/O port pins
Note: When enabled, these pins must be properly configured as input or output.
- bit 4 **CKP:** Clock Polarity Select bit
 1 = Idle state for clock is a high level
 0 = Idle state for clock is a low level
- bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits
 0101 = SPI Slave mode, clock = SCK pin. \overline{SS} pin control disabled. \overline{SS} can be used as I/O pin.
 0100 = SPI Slave mode, clock = SCK pin. \overline{SS} pin control enabled.
 0011 = SPI Master mode, clock = TMR2 output/2
 0010 = SPI Master mode, clock = FOSC/64
 0001 = SPI Master mode, clock = FOSC/16
 0000 = SPI Master mode, clock = FOSC/4
Note: Bit combinations not specifically listed here are either reserved or implemented in I²C mode only.

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

PIC16F87XA

9.3.2 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)

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 eight bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full detect bit, BF (SSPSTAT<0>), and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the

SSPBUF register during transmission/reception of data will be ignored and the write collision detect bit, WCOL (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 SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 9-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

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

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

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

9.3.3 ENABLING SPI I/O

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

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<5> bit cleared
- SCK (Master mode) must have TRISC<3> bit cleared
- SCK (Slave mode) must have TRISC<3> bit set
- $\overline{\text{SS}}$ must have TRISC<4> bit set

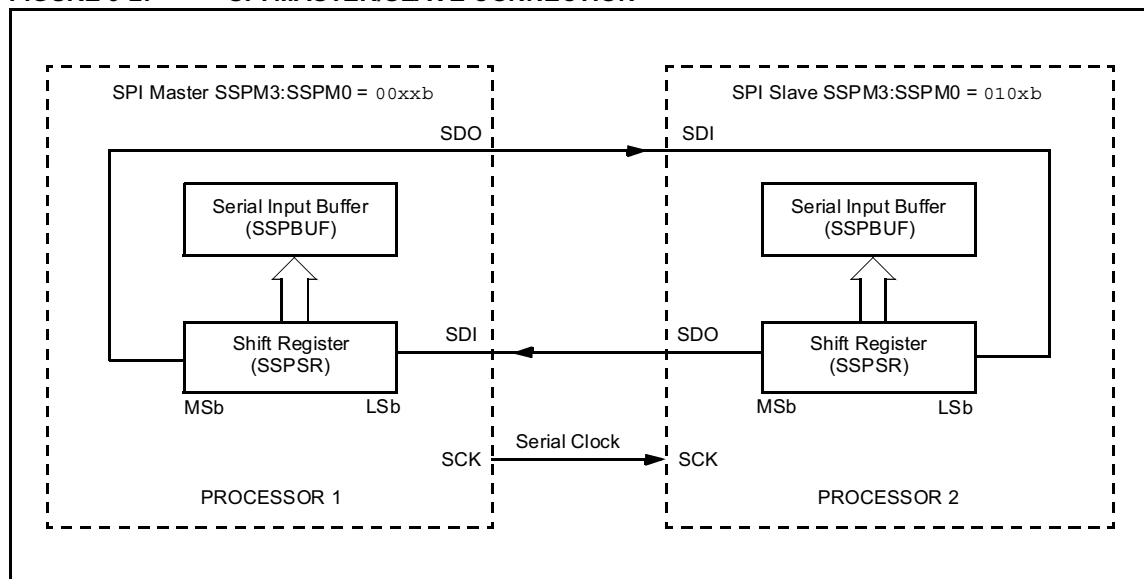
Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

9.3.4 TYPICAL CONNECTION

Figure 9-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

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

FIGURE 9-2: SPI MASTER/SLAVE CONNECTION



PIC16F87XA

9.3.5 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 9-2) 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 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" mode.

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

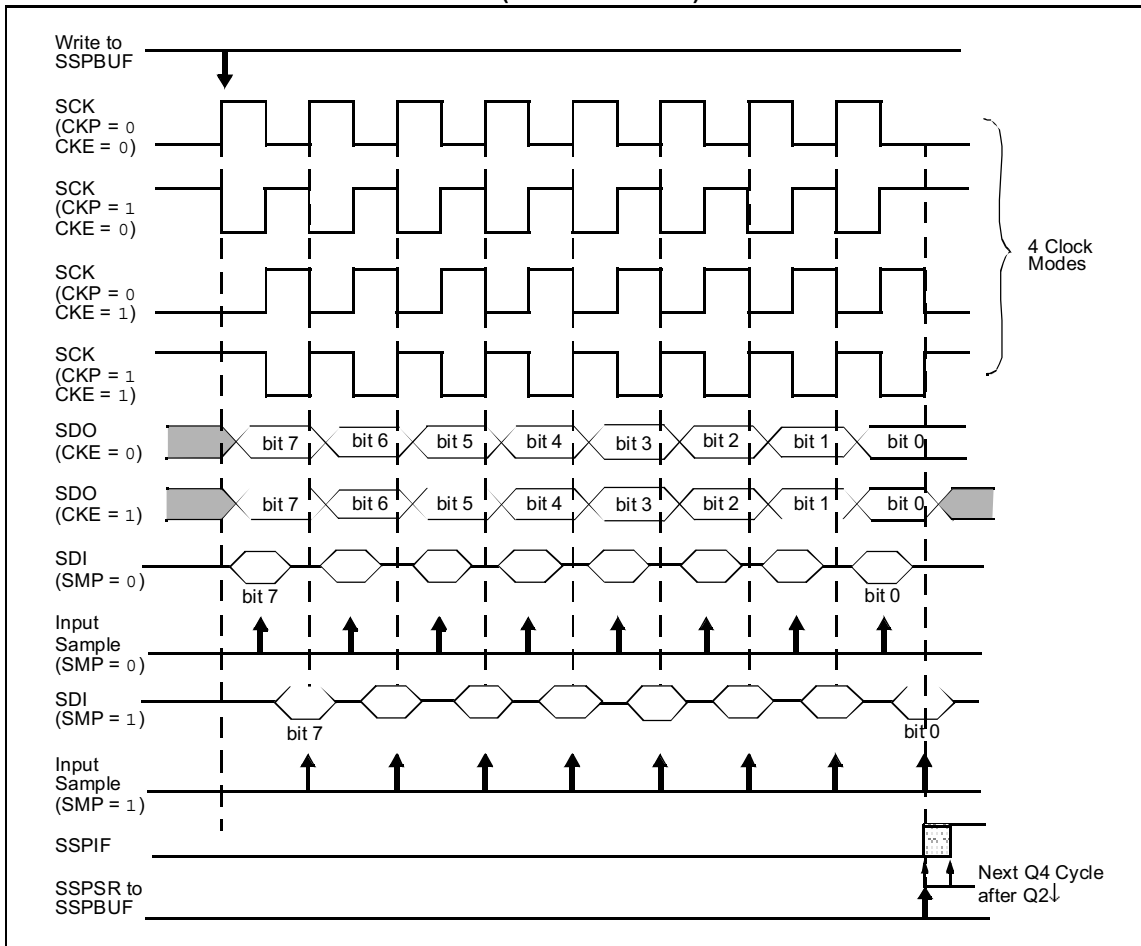
Figure 9-3, Figure 9-5 and Figure 9-6, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

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

This allows a maximum data rate (at 40 MHz) of 10.00 Mbps.

Figure 9-3 shows the waveforms for Master mode. When the CKE bit is set, 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 9-3: SPI MODE WAVEFORM (MASTER MODE)



9.3.6 SLAVE MODE

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

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

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

9.3.7 SLAVE SELECT SYNCHRONIZATION

The \overline{SS} pin allows a Synchronous Slave mode. The SPI must be in Slave mode with \overline{SS} pin control enabled ($SSPCON<3:0> = 04h$). The pin must not be driven low for the \overline{SS} pin to function as an input. The data latch must be high. When the \overline{SS} pin is low, transmission and reception are enabled and the SDO pin is driven. When

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

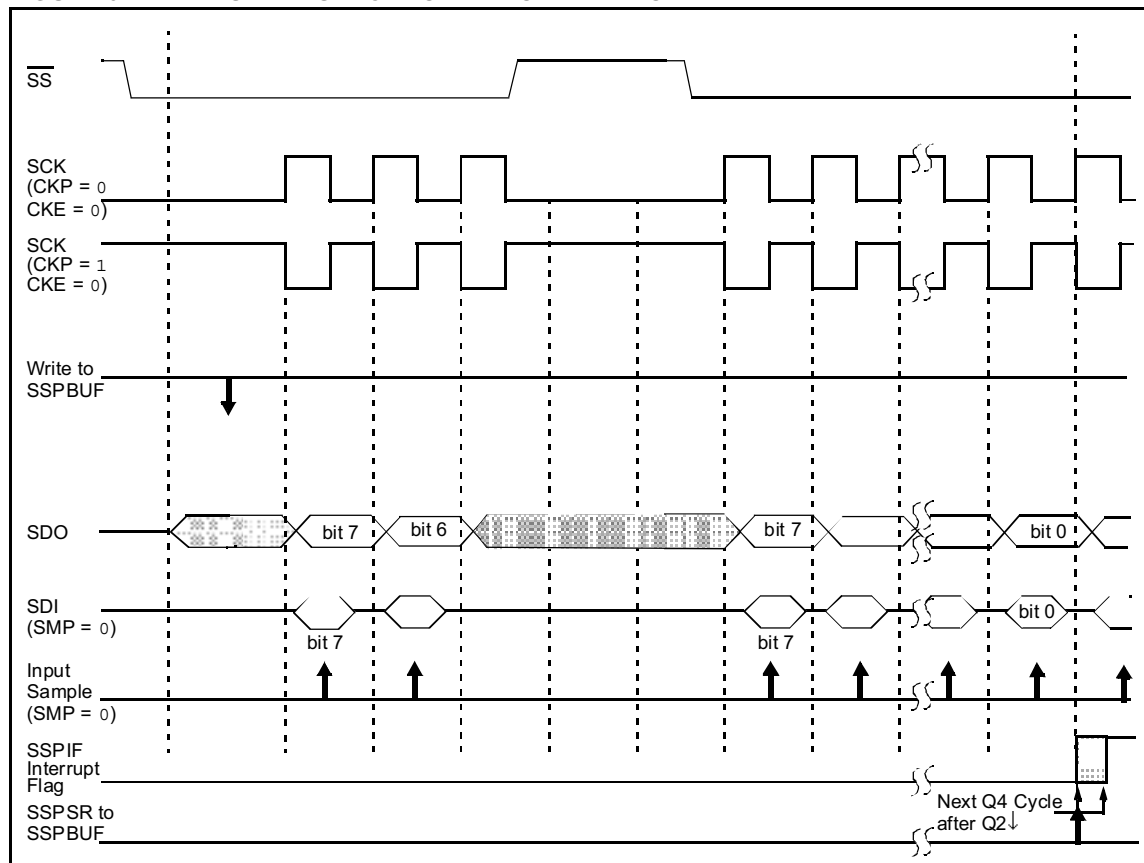
Note 1: When the SPI is in Slave mode with \overline{SS} pin control enabled ($SSPCON<3:0> = 0100$), the SPI module will reset if the \overline{SS} pin is set to VDD.

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

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

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

FIGURE 9-4: SLAVE SYNCHRONIZATION WAVEFORM



PIC16F87XA

FIGURE 9-5: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

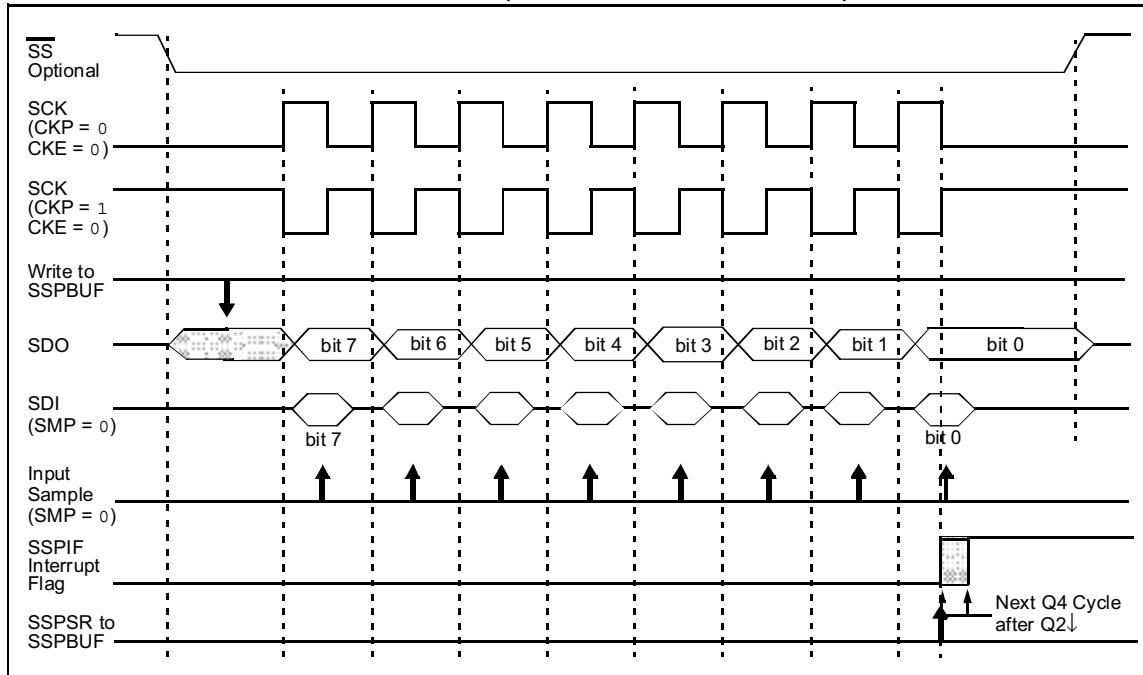
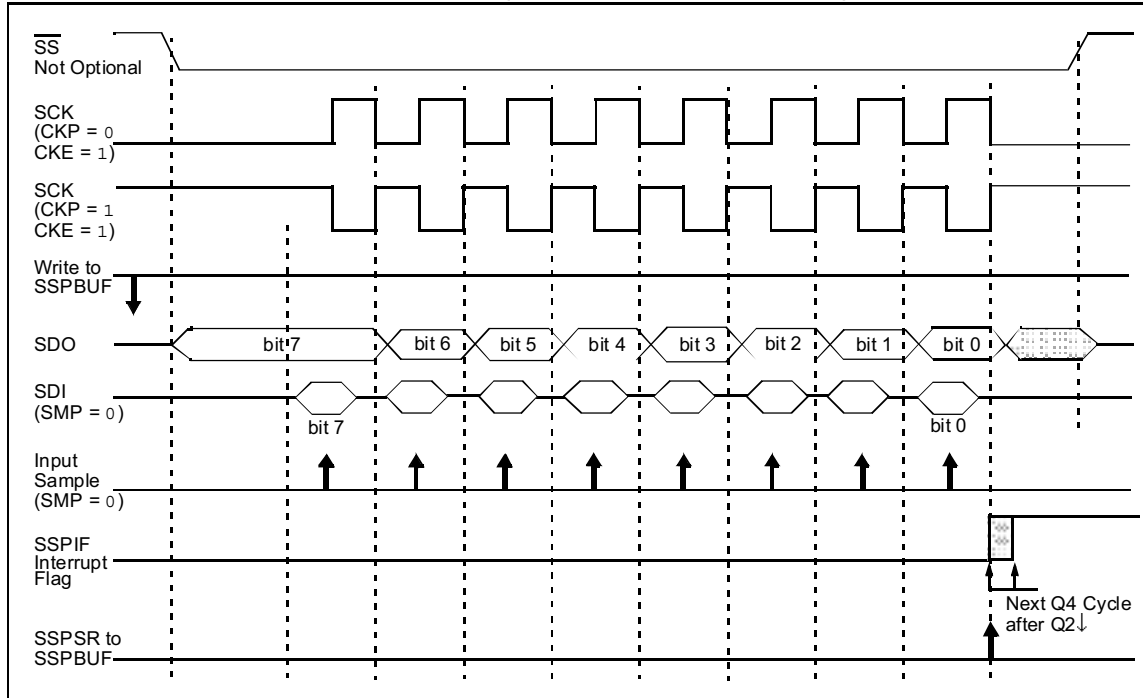


FIGURE 9-6: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



PIC16F87XA

9.3.8 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.

9.3.9 EFFECTS OF A RESET

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

9.3.10 BUS MODE COMPATIBILITY

Table 9-1 shows the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

TABLE 9-1: SPI BUS MODES

Standard SPI Mode Terminology	Control Bits State	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

There is also a SMP bit which controls when the data is sampled.

TABLE 9-2: REGISTERS ASSOCIATED WITH SPI OPERATION

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
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
TRISC	PORTC Data Direction Register								1111 1111	1111 1111
SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
TRISA	PORTA Data Direction Register								--11 1111	--11 1111
SSPSTAT	SMP	CKE	D/Ā	P	S	R/W	UA	BF	0000 0000	0000 0000

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

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

PIC16F87XA

9.4 I²C Mode

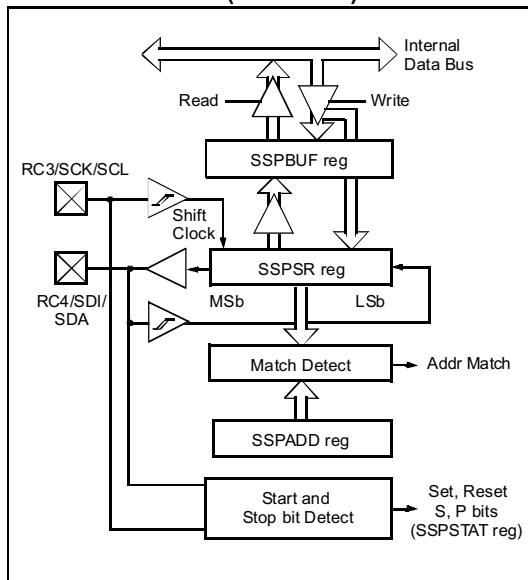
The MSSP module in I²C mode fully implements all master and slave functions (including general call support) and provides interrupts on Start and Stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer:

- Serial clock (SCL) – RC3/SCK/SCL
- Serial data (SDA) – RC4/SDI/SDA

The user must configure these pins as inputs or outputs through the TRISC<4:3> bits.

FIGURE 9-7: MSSP BLOCK DIAGRAM (I²C MODE)



9.4.1 REGISTERS

The MSSP module has six registers for I²C operation. These are:

- MSSP Control Register (SSPCON)
- MSSP Control Register 2 (SSPCON2)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer Register (SSPBUF)
- MSSP Shift Register (SSPSR) – Not directly accessible
- MSSP Address Register (SSPADD)

SSPCON, SSPCON2 and SSPSTAT are the control and status registers in I²C mode operation. The SSPCON and SSPCON2 registers are readable and writable. The lower six bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

SSPADD register holds the slave device address when the SSP is configured in I²C Slave mode. When the SSP is configured in Master mode, the lower seven bits of SSPADD act as the baud rate generator reload value.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not double-buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

PIC16F87XA

REGISTER 9-3: SSPSTAT: MSSP STATUS REGISTER (I²C MODE) (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
bit 7						bit 0	

- bit 7 **SMP:** Slew Rate Control bit
In 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:** SMBus Select bit
In Master or Slave mode:
 1 = Enable SMBus specific inputs
 0 = Disable SMBus specific inputs
- bit 5 **D/A:** Data/Address bit
In Master mode:
 Reserved.
In Slave mode:
 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
 1 = Indicates that a Stop bit has been detected last
 0 = Stop bit was not detected last
Note: This bit is cleared on Reset and when SSPEN is cleared.
- bit 3 **S:** Start bit
 1 = Indicates that a Start bit has been detected last
 0 = Start bit was not detected last
Note: This bit is cleared on Reset and when SSPEN is cleared.
- bit 2 **R/W:** Read/Write bit information (I²C mode only)
In Slave mode:
 1 = Read
 0 = Write
Note: 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 Master mode:
 1 = Transmit is in progress
 0 = Transmit is not in progress
Note: ORing 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 Slave 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
In Transmit mode:
 1 = Receive complete, SSPBUF is full
 0 = Receive not complete, SSPBUF is empty
In Receive mode:
 1 = Data Transmit in progress (does not include the ACK and Stop bits), SSPBUF is full
 0 = Data Transmit complete (does not include the ACK and Stop bits), SSPBUF is empty

Legend:

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

PIC16F87XA

REGISTER 9-4: SSPCON1: MSSP CONTROL REGISTER 1 (I²C MODE) (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	
bit 7								bit 0

- bit 7 **WCOL:** Write Collision Detect bit
In Master Transmit mode:
 1 = A write to the SSPBUF register was attempted while the I²C conditions were not valid for a transmission to be started. (Must be cleared in software.)
 0 = No collision
In Slave Transmit mode:
 1 = The SSPBUF register is written while it is still transmitting the previous word. (Must be cleared in software.)
 0 = No collision
In Receive mode (Master or Slave modes):
 This is a “don’t care” bit.
- bit 6 **SSPOV:** Receive Overflow Indicator bit
In Receive mode:
 1 = A byte is received while the SSPBUF register is still holding the previous byte. (Must be cleared in software.)
 0 = No overflow
In Transmit mode:
 This is a “don’t care” bit in Transmit mode.
- bit 5 **SSPEN:** Synchronous Serial Port Enable bit
 1 = Enables the serial port and configures the SDA and SCL pins as the serial port pins
 0 = Disables the serial port and configures these pins as I/O port pins
Note: When enabled, the SDA and SCL pins must be properly configured as input or output.
- bit 4 **CKP:** SCK Release Control bit
In Slave mode:
 1 = Release clock
 0 = Holds clock low (clock stretch). (Used to ensure data setup time.)
In Master mode:
 Unused in this mode.
- bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits
 1111 = I²C Slave mode, 10-bit address with Start and Stop bit interrupts enabled
 1110 = I²C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
 1011 = I²C Firmware Controlled Master mode (Slave Idle)
 1000 = I²C Master mode, clock = Fosc/(4 * (SSPADD + 1))
 0111 = I²C Slave mode, 10-bit address
 0110 = I²C Slave mode, 7-bit address
Note: Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.

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

PIC16F87XA

REGISTER 9-5: SSPCON2: MSSP CONTROL REGISTER 2 (I²C MODE) (ADDRESS 91h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

- bit 7 **GCEN:** General Call Enable bit (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 (Master Transmit mode only)
 1 = Acknowledge was not received from slave
 0 = Acknowledge was received from slave
- bit 5 **ACKDT:** Acknowledge Data bit (Master Receive mode only)
 1 = Not Acknowledge
 0 = Acknowledge
Note: Value that will be transmitted when the user initiates an Acknowledge sequence at the end of a receive.
- bit 4 **ACKEN:** Acknowledge Sequence Enable bit (Master Receive mode only)
 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 (Master mode only)
 1 = Enables Receive mode for I²C
 0 = Receive Idle
- bit 2 **PEN:** Stop Condition Enable bit (Master mode only)
 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 (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/Stretch Enabled bit
In Master mode:
 1 = Initiate Start condition on SDA and SCL pins. Automatically cleared by hardware.
 0 = Start condition Idle
In Slave mode:
 1 = Clock stretching is enabled for both slave transmit and slave receive (stretch enabled)
 0 = Clock stretching is enabled for slave transmit only (PIC16F87X compatibility)

Legend:

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

Note: For bits ACKEN, RCEN, PEN, RSEN, SEN: If the I²C 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).

PIC16F87XA

9.4.2 OPERATION

The MSSP module functions are enabled by setting MSSP Enable bit, SSPEN (SSPCON<5>).

The SSPCON register allows control of the I²C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I²C modes to be selected:

- I²C Master mode, clock = OSC/4 (SSPADD + 1)
- I²C Slave mode (7-bit address)
- I²C Slave mode (10-bit address)
- I²C Slave mode (7-bit address) with Start and Stop bit interrupts enabled
- I²C Slave mode (10-bit address) with Start and Stop bit interrupts enabled
- I²C Firmware Controlled Master mode, slave is Idle

Selection of any I²C mode, with the SSPEN bit set, forces the SCL and SDA pins to be open-drain, provided these pins are programmed to inputs by setting the appropriate TRISC bits. To ensure proper operation of the module, pull-up resistors must be provided externally to the SCL and SDA pins.

9.4.3 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs (TRISC<4:3> set). The MSSP module will override the input state with the output data when required (slave-transmitter).

The I²C Slave mode hardware will always generate an interrupt on an address match. Through the mode select bits, the user can also choose to interrupt on Start and Stop bits

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

Any combination of the following conditions will cause the MSSP module not to give this \overline{ACK} pulse:

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

In this case, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF (PIR1<3>) is set. The BF bit 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 for proper operation. The high and low times of the I²C specification, as well as the requirement of the MSSP module, are shown in timing parameter #100 and parameter #101.

9.4.3.1 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:

1. The SSPSR register value is loaded into the SSPBUF register.
2. The Buffer Full bit, BF, is set.
3. An \overline{ACK} pulse is generated.
4. MSSP Interrupt Flag bit, SSPIF (PIR1<3>), is set (interrupt is generated if enabled) on the falling edge of the ninth 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 $\overline{R/W}$ (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 '11110 A9 A8 0', where 'A9' and 'A8' are the two MSBs of the address. The sequence of events for 10-bit address is as follows, with steps 7 through 9 for the 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. If match releases SCL line, this will clear bit UA.
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.

9.4.3.2 Reception

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

When the address byte overflow condition exists, then the No Acknowledge (\overline{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 MSSP 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 byte.

If SEN is enabled (SSPCON<0> = 1), RC3/SCK/SCL will be held low (clock stretch) following each data transfer. The clock must be released by setting bit CKP (SSPCON<4>). See **Section 9.4.4 “Clock Stretching”** for more detail.

9.4.3.3 Transmission

When the $\overline{R/W}$ bit of the incoming address byte is set and an address match occurs, the $\overline{R/W}$ bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The \overline{ACK} pulse will be sent on the ninth bit and pin RC3/SCK/SCL is held low regardless of SEN (see **Section 9.4.4 “Clock Stretching”** for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then pin RC3/SCK/SCL should be enabled by setting bit CKP (SSPCON<4>). 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 9-9).

The \overline{ACK} pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line is high (not \overline{ACK}), then the data transfer is complete. In this case, when the \overline{ACK} is latched by the slave, the slave logic is reset (resets SSPSTAT register) and the slave monitors for another occurrence of the Start bit. If the SDA line was low (\overline{ACK}), the next transmit data must be loaded into the SSPBUF register. Again, pin RC3/SCK/SCL must be enabled by setting bit CKP.

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

PIC16F87XA

FIGURE 9-8: I²C SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 7-BIT ADDRESS)

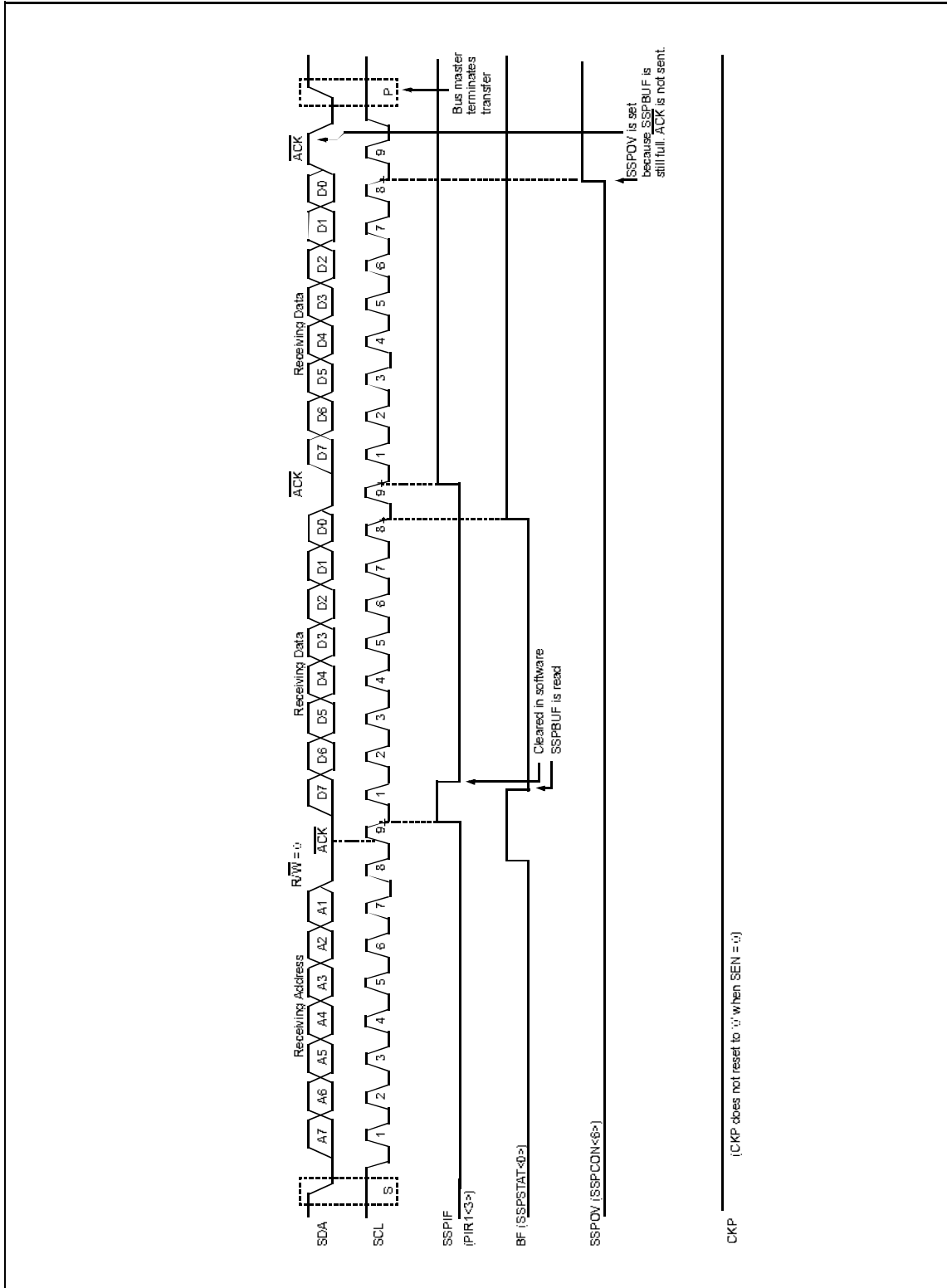
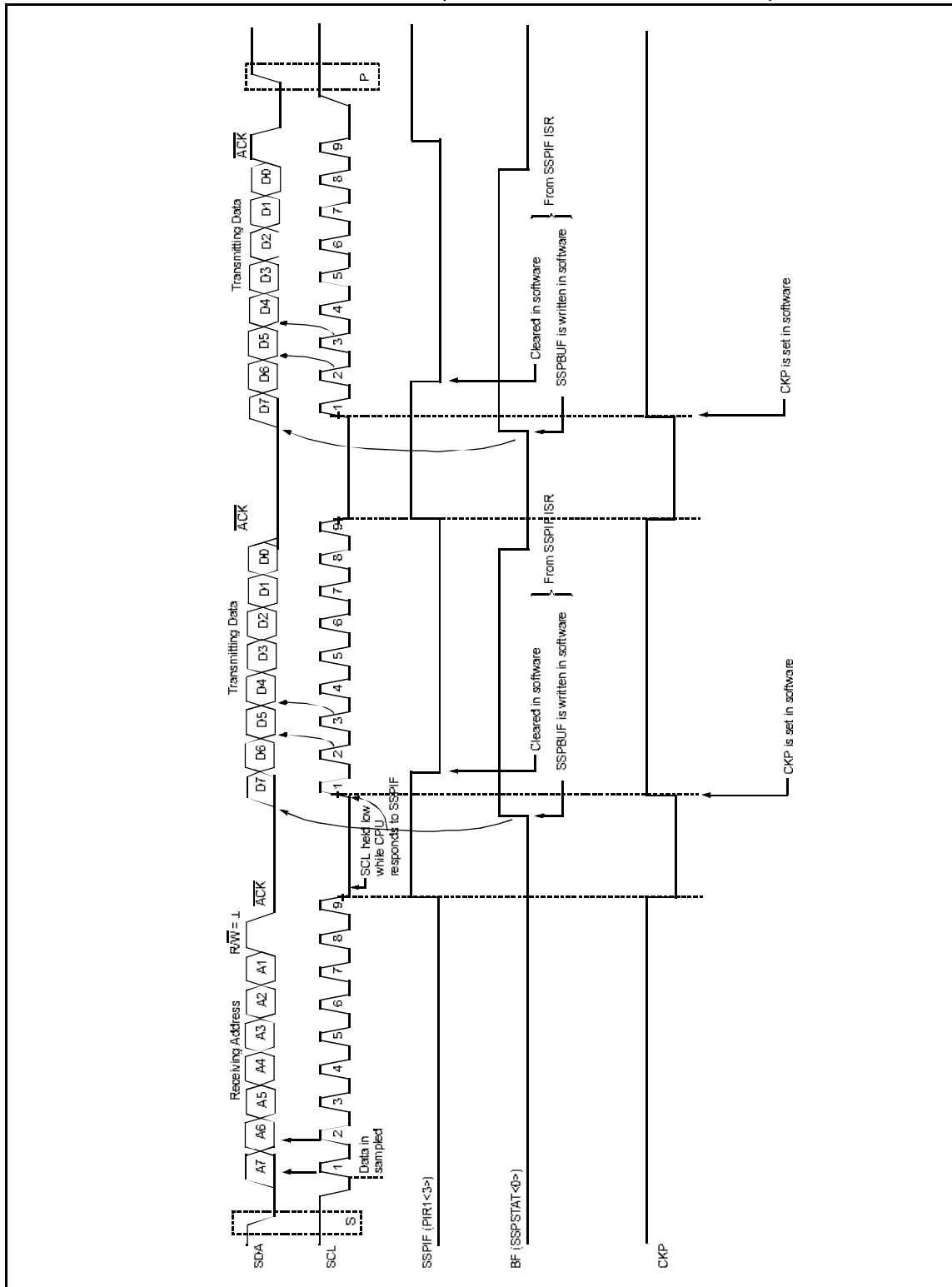


FIGURE 9-9: I²C SLAVE MODE TIMING (TRANSMISSION, 7-BIT ADDRESS)



PIC16F87XA

FIGURE 9-10: I²C SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 10-BIT ADDRESS)

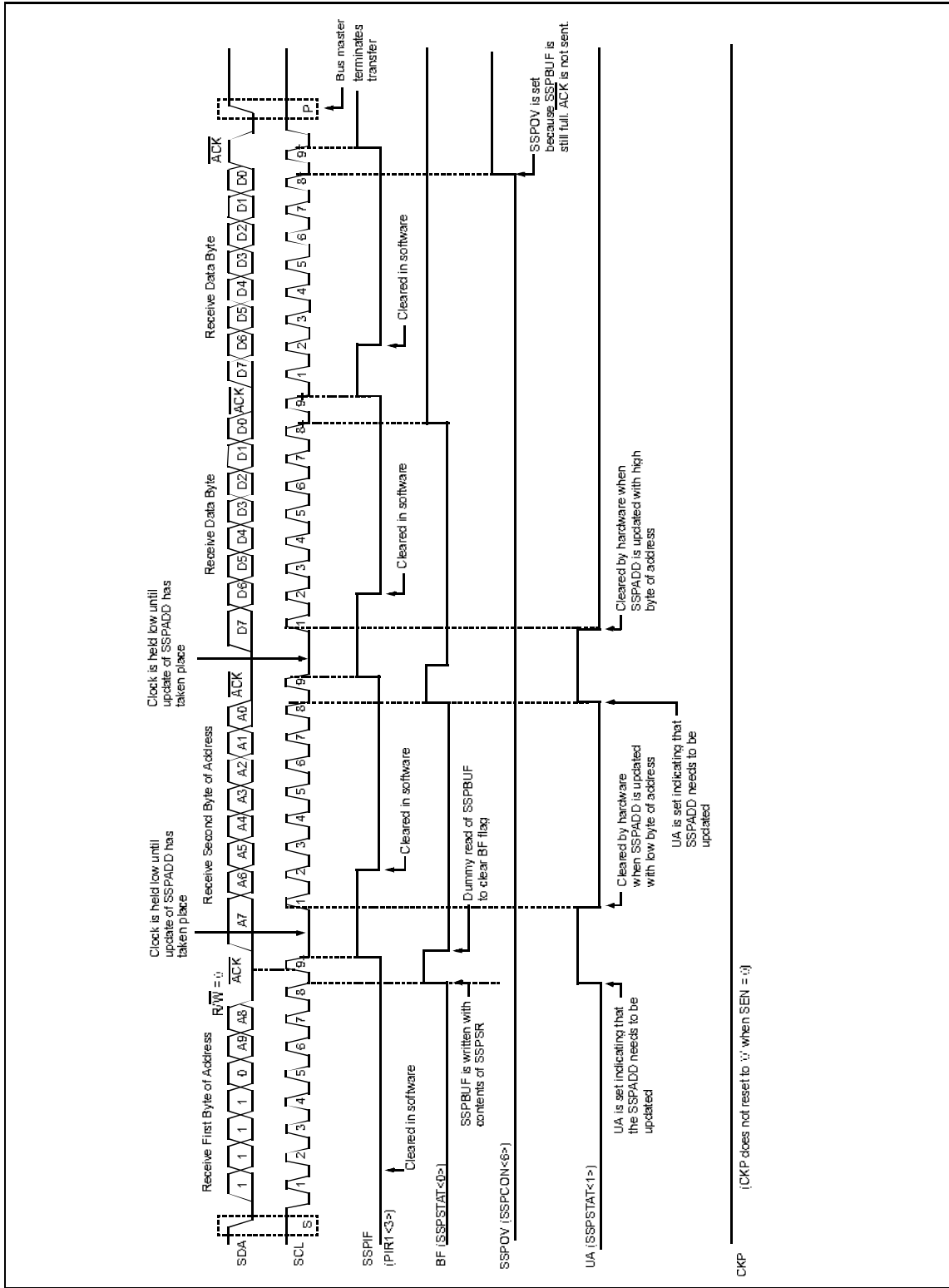
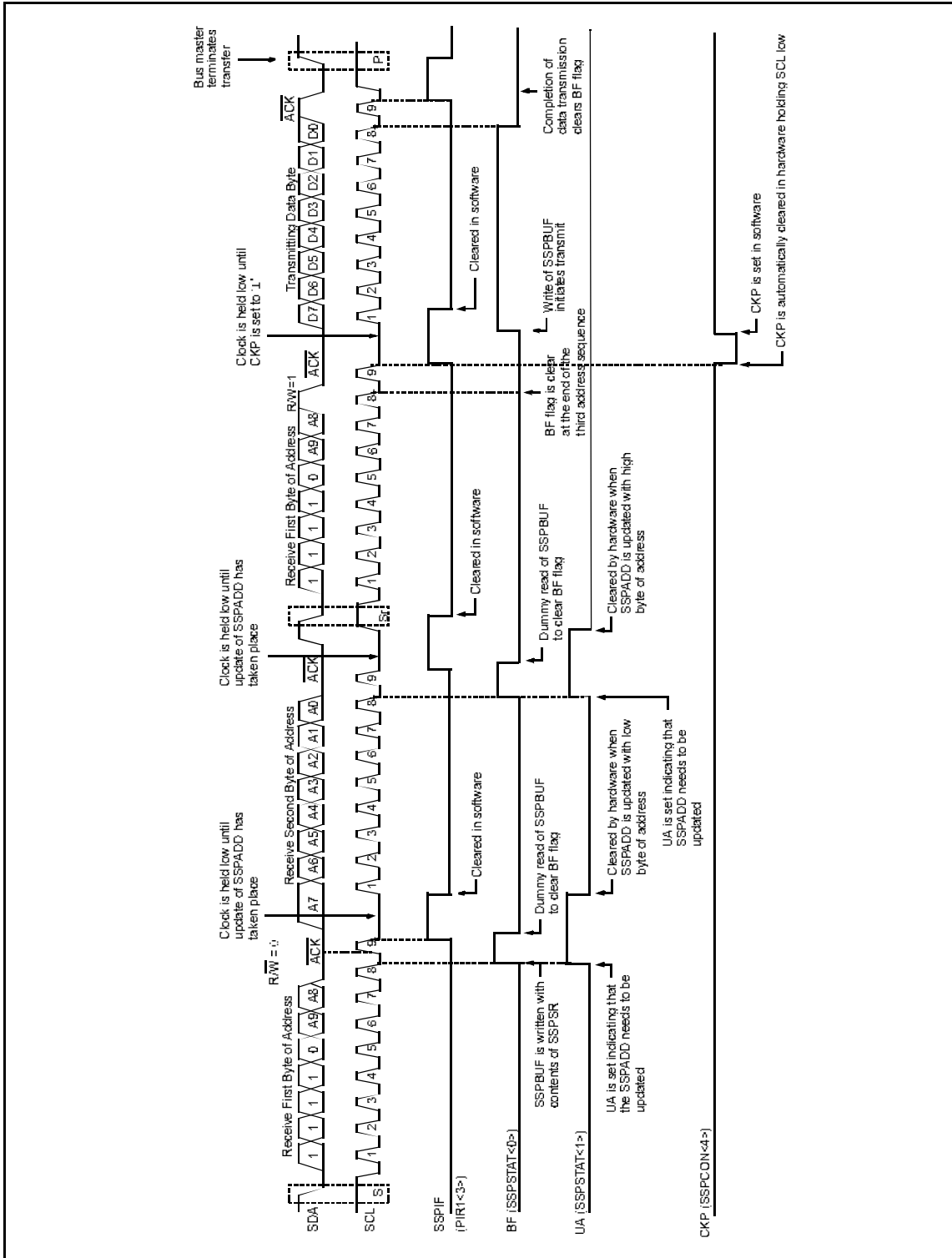


FIGURE 9-11: I²C SLAVE MODE TIMING (TRANSMISSION, 10-BIT ADDRESS)



PIC16F87XA

9.4.4 CLOCK STRETCHING

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

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

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

In 7-bit Slave Receive mode, on the falling edge of the ninth clock at the end of the ACK sequence, if the BF bit is set, the CKP bit in the SSPCON register is automatically cleared, forcing the SCL output to be held low. The CKP bit being cleared to '0' will assert the SCL line low. The CKP bit must be set in the user's ISR before reception is allowed to continue. By holding the SCL line low, the user has time to service the ISR and read the contents of the SSPBUF before the master device can initiate another receive sequence. This will prevent buffer overruns from occurring (see Figure 9-13).

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

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

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

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

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

9.4.4.3 Clock Stretching for 7-bit Slave Transmit Mode

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

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

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

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

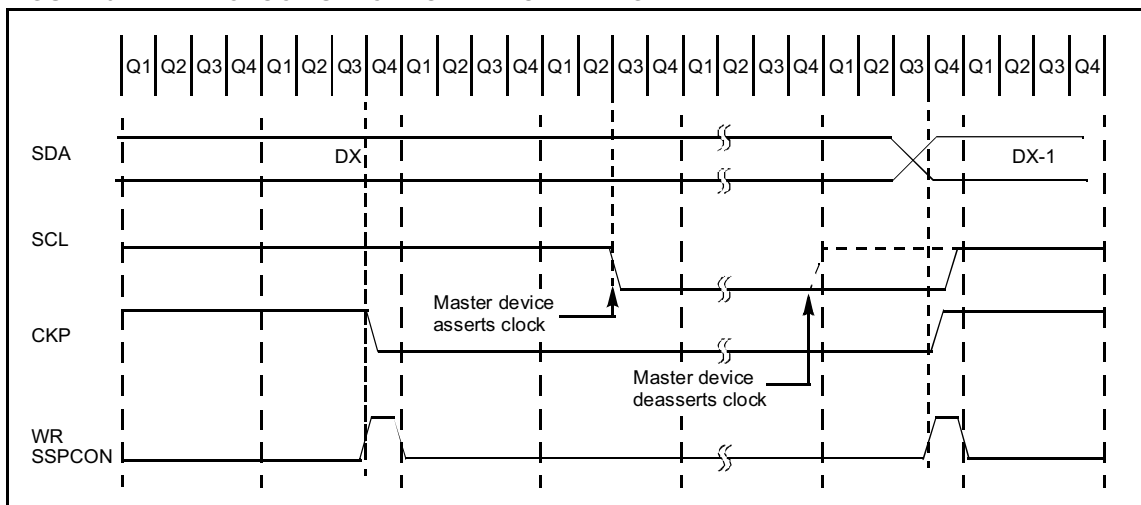
9.4.4.4 Clock Stretching for 10-bit Slave Transmit Mode

In 10-bit Slave Transmit mode, clock stretching is controlled during the first two address sequences by the state of the UA bit, just as it is in 10-bit Slave Receive mode. The first two addresses are followed by a third address sequence, which contains the high order bits of the 10-bit address and the R/W bit set to '1'. After the third address sequence is performed, the UA bit is not set, the module is now configured in Transmit mode and clock stretching is controlled by the BF flag as in 7-bit Slave Transmit mode (see Figure 9-11).

9.4.4.5 Clock Synchronization and the CKP Bit

When the CKP bit is cleared, the SCL output is forced to '0'; however, setting the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external I²C master device has already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I²C bus have deasserted SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 9-12).

FIGURE 9-12: CLOCK SYNCHRONIZATION TIMING



PIC16F87XA

FIGURE 9-13: I²C SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 7-BIT ADDRESS)

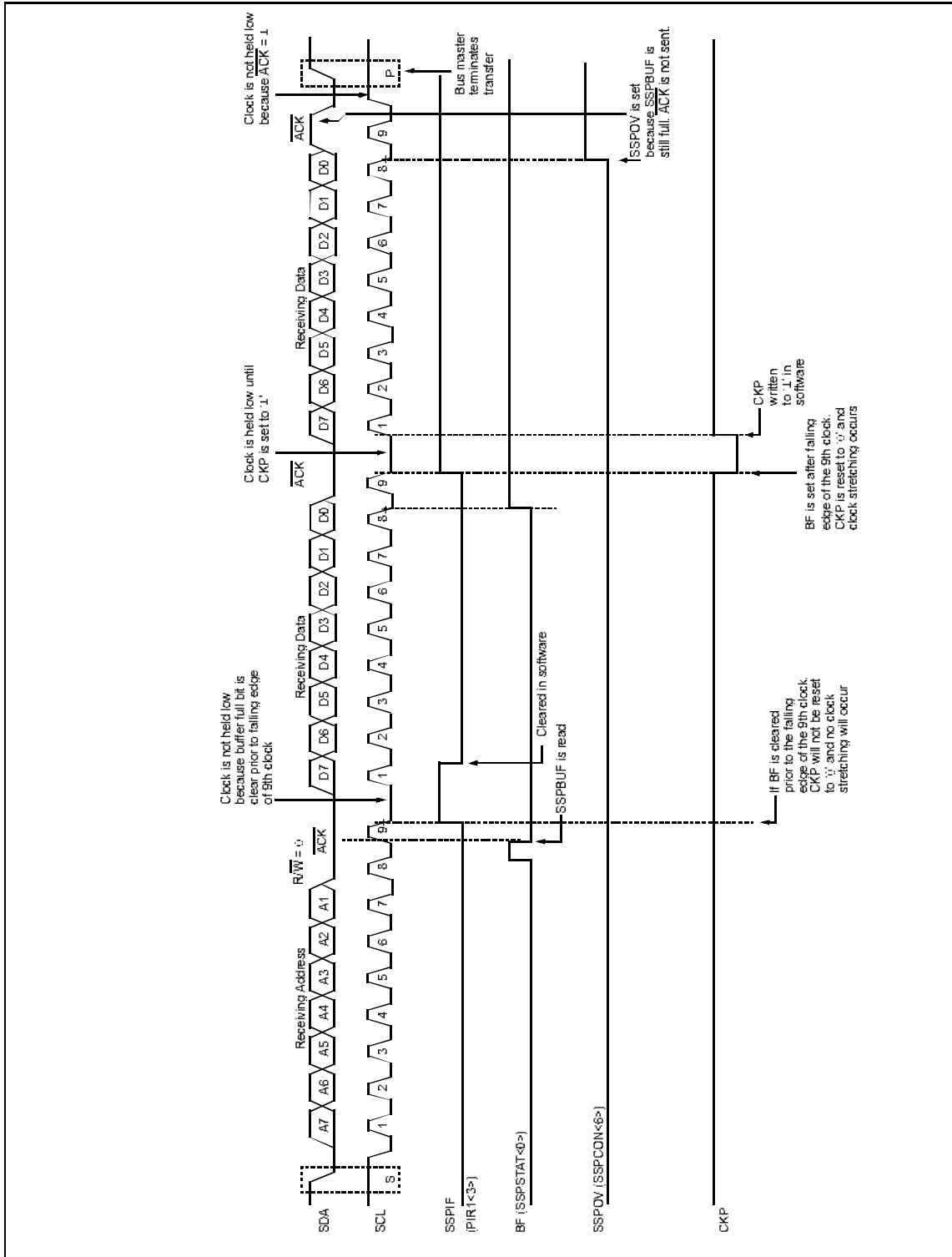
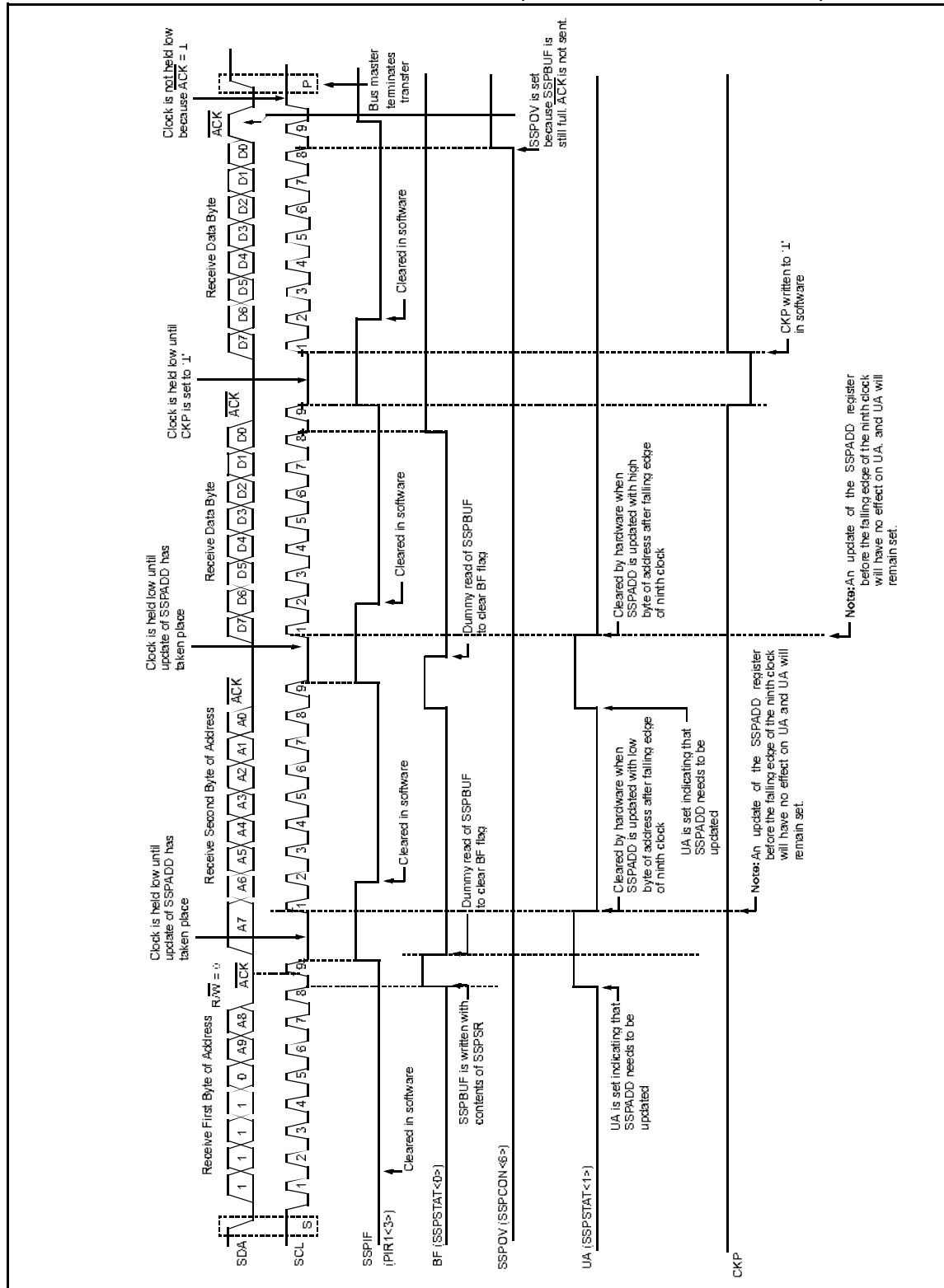


FIGURE 9-14: I²C SLAVE MODE TIMING SEN = 1 (RECEPTION, 10-BIT ADDRESS)



PIC16F87XA

9.4.5 GENERAL CALL ADDRESS SUPPORT

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

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

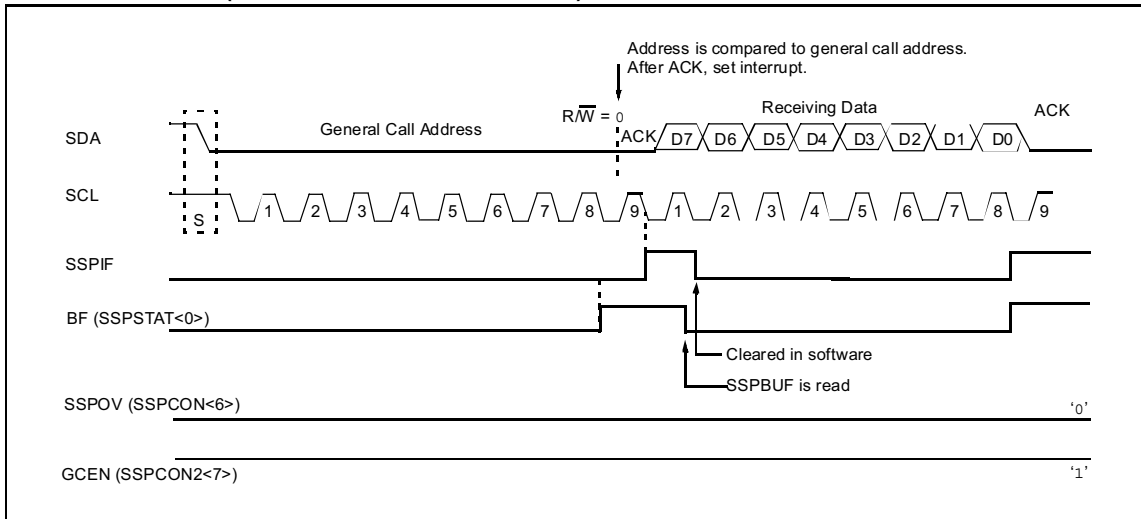
The general call address is recognized when the General Call Enable bit (GCEN) is enabled (SSPCON2<7> set). Following a Start bit detect, 8 bits are shifted into the SSPSR and the address is compared against the SSPADD. It is also compared to the general call address and fixed in hardware.

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

When the interrupt is serviced, the source for the interrupt can be checked by reading the contents of the SSPBUF. The value can be used 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 the GCEN bit 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 9-15).

FIGURE 9-15: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE (7 OR 10-BIT ADDRESS MODE)



9.4.6 MASTER MODE

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

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I²C bus may be taken when the P bit is set or the bus is Idle, with both the S and P bits clear.

In Firmware Controlled Master mode, user code conducts all I²C bus operations based on Start and Stop bit conditions.

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

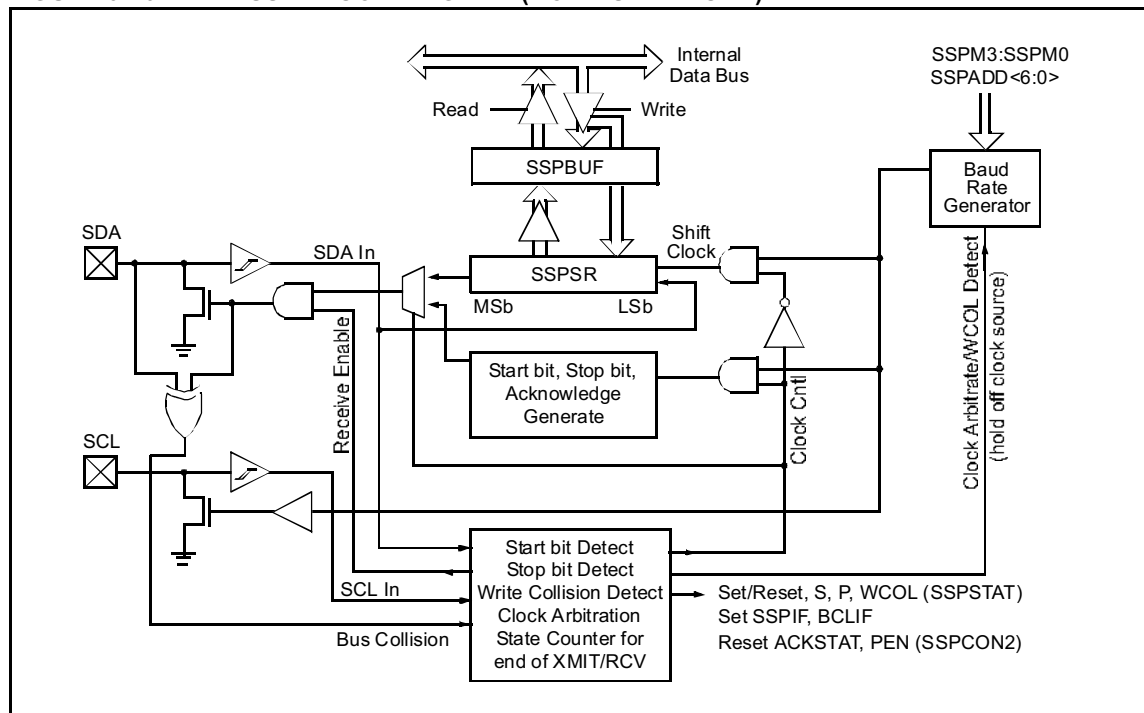
1. Assert a Start condition on SDA and SCL.
2. Assert a Repeated Start condition on SDA and SCL.
3. Write to the SSPBUF register, initiating transmission of data/address.
4. Configure the I²C port to receive data.
5. Generate an Acknowledge condition at the end of a received byte of data.
6. Generate a Stop condition on SDA and SCL.

Note: The MSSP module, when configured in I²C Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPBUF register to initiate transmission before the Start condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur.

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

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

FIGURE 9-16: MSSP BLOCK DIAGRAM (I²C MASTER MODE)



PIC16F87XA

9.4.6.1 I²C Master Mode Operation

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

In Master Transmitter mode, serial data is output through SDA while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/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 the 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 the SPI mode operation is used to set the SCL clock frequency for either 100 kHz, 400 kHz or 1 MHz I²C operation. See **Section 9.4.7 "Baud Rate Generator"** for more detail.

A typical transmit sequence would go as follows:

1. The user generates a Start condition by setting the Start Enable bit, SEN (SSPCON2<0>).
2. SSPIF is set. The MSSP module will wait the required Start time before any other operation takes place.
3. The user loads the SSPBUF with the slave address to transmit.
4. Address is shifted out the SDA pin until all 8 bits are transmitted.
5. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
6. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
7. The user loads the SSPBUF with eight bits of data.
8. Data is shifted out the SDA pin until all 8 bits are transmitted.
9. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
10. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
11. The user generates a Stop condition by setting the Stop Enable bit, PEN (SSPCON2<2>).
12. Interrupt is generated once the Stop condition is complete.

9.4.7 BAUD RATE GENERATOR

In I²C Master mode, the Baud Rate Generator (BRG) reload value is placed in the lower 7 bits of the SSPADD register (Figure 9-17). When a write occurs to SSPBUF, the Baud Rate Generator will automatically begin counting. 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 clocks. In I²C Master mode, the BRG is reloaded automatically.

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

Table 9-3 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPADD.

FIGURE 9-17: BAUD RATE GENERATOR BLOCK DIAGRAM

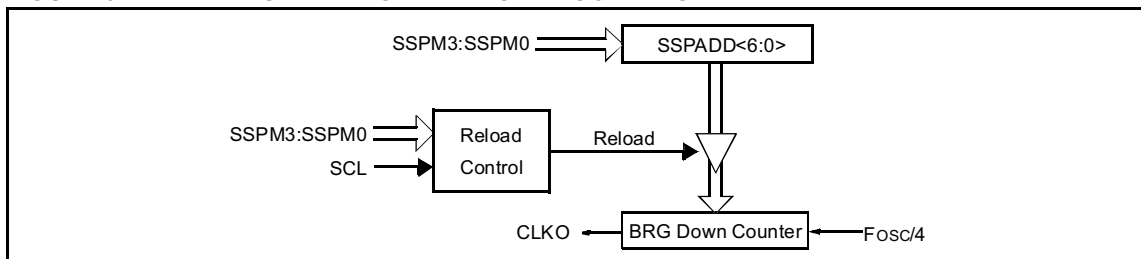


TABLE 9-3: I²C CLOCK RATE W/BRG

Fcy	Fcy*2	BRG Value	Fscl (2 Rollovers of BRG)
10 MHz	20 MHz	19h	400 kHz ⁽¹⁾
10 MHz	20 MHz	20h	312.5 kHz
10 MHz	20 MHz	3Fh	100 kHz
4 MHz	8 MHz	0Ah	400 kHz ⁽¹⁾
4 MHz	8 MHz	0Dh	308 kHz
4 MHz	8 MHz	28h	100 kHz
1 MHz	2 MHz	03h	333 kHz ⁽¹⁾
1 MHz	2 MHz	0Ah	100 kHz
1 MHz	2 MHz	00h	1 MHz ⁽¹⁾

Note 1: The I²C interface does not conform to the 400 kHz I²C specification (which applies to rates greater than 100 kHz) in all details, but may be used with care where higher rates are required by the application.

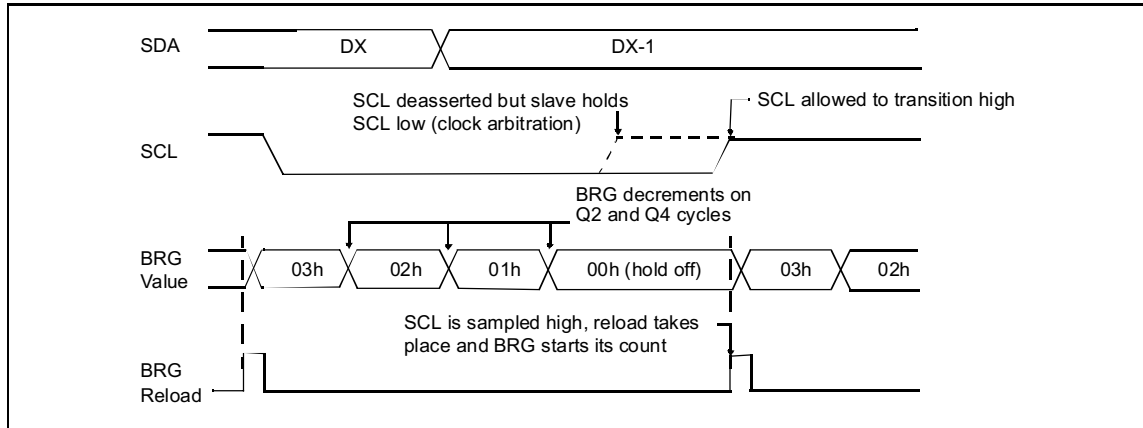
PIC16F87XA

9.4.7.1 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 9-17).

FIGURE 9-18: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION



9.4.8 I²C MASTER MODE START CONDITION TIMING

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

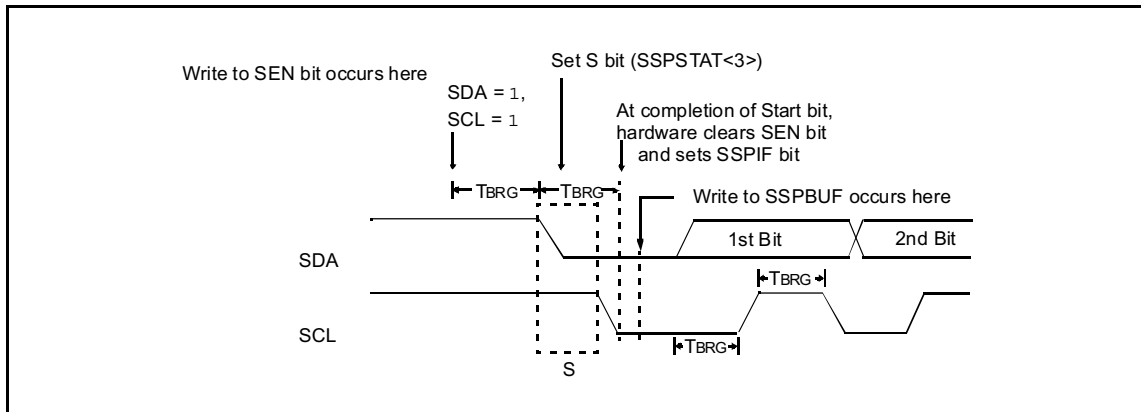
Note: If at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag (BCLIF) is set, the Start condition is aborted and the I²C module is reset into its Idle state.

9.4.8.1 WCOL Status Flag

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

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

FIGURE 9-19: FIRST START BIT TIMING



PIC16F87XA

9.4.9 I²C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition occurs when the RSEN bit (SSPCON2<1>) is programmed high and the I²C logic 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<5:0> and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. Following this, the RSEN bit (SSPCON2<1>) will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition 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.
- 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'.

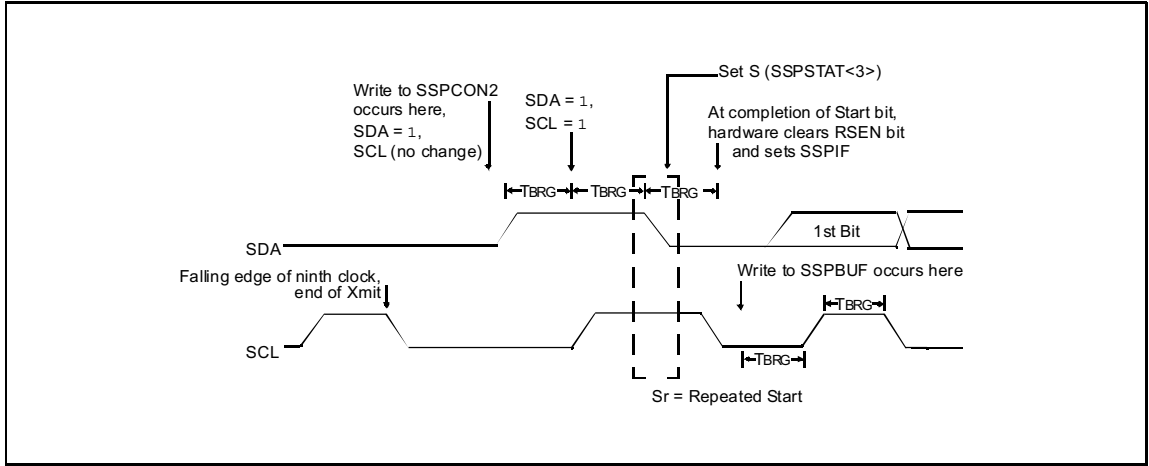
Immediately following the SSPIF bit getting set, the user may write the SSPBUF with the 7-bit address in 7-bit mode or the default first address in 10-bit mode. After the first eight bits are transmitted and an ACK is received, the user may then transmit an additional eight bits of address (10-bit mode)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

9.4.9.1 WCOL Status Flag

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

FIGURE 9-20: REPEAT START CONDITION WAVEFORM



9.4.10 I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPBUF register. This action will set the Buffer Full flag bit, 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 specification, parameter #106). SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high (see data setup time specification, parameter #107). When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an ACK bit during the ninth bit time, if an address match occurred or if data was received properly. The status of ACK is written into the ACKDT bit 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 bit 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 9-21).

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/W bit are completed. On the falling edge of the eighth clock, the master will deassert 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.

9.4.10.1 BF Status Flag

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

9.4.10.2 WCOL Status Flag

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

WCOL must be cleared in software.

9.4.10.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit (SSPCON2<6>) is cleared when the slave has sent an Acknowledge (ACK = 0) and is set when the slave does Not Acknowledge (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.

9.4.11 I²C MASTER MODE RECEPTION

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

Note: The MSSP 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 bit is set, the SSPIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCL low. The MSSP is now in Idle state, awaiting the next command. When the buffer is read by the CPU, the BF flag bit 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>).

9.4.11.1 BF Status Flag

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

9.4.11.2 SSPOV Status Flag

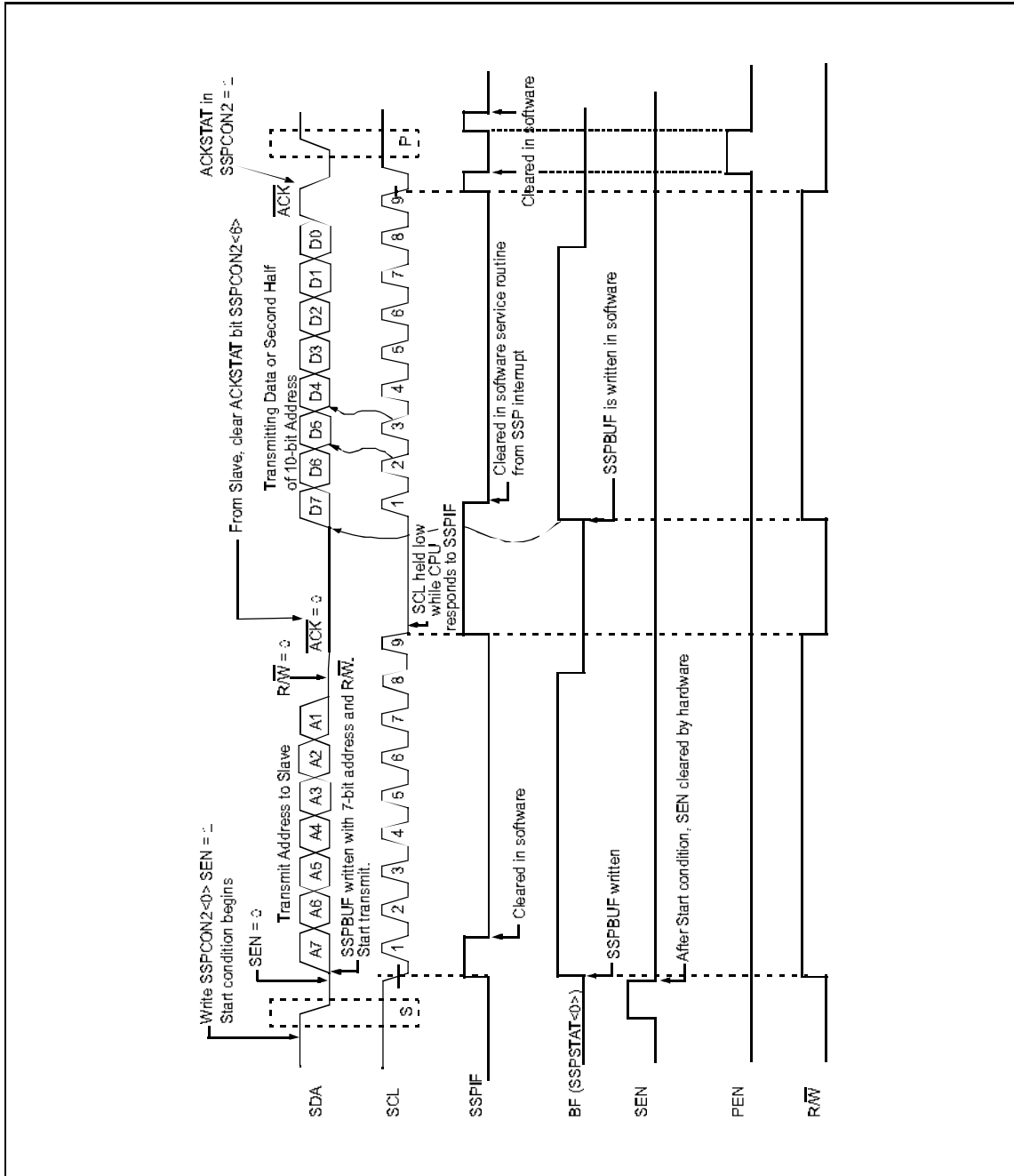
In receive operation, the SSPOV bit is set when 8 bits are received into the SSPSR and the BF flag bit is already set from a previous reception.

9.4.11.3 WCOL Status Flag

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

PIC16F87XA

FIGURE 9-21: I²C MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)



PIC16F87XA

9.4.12 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 are 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 (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the baud rate generator is turned off and the MSSP module then goes into Idle mode (Figure 9-23).

9.4.12.1 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).

9.4.13 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN (SSPCON2<2>). At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to 0. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit (SSPSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 9-24).

9.4.13.1 WCOL Status Flag

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

FIGURE 9-23: ACKNOWLEDGE SEQUENCE WAVEFORM

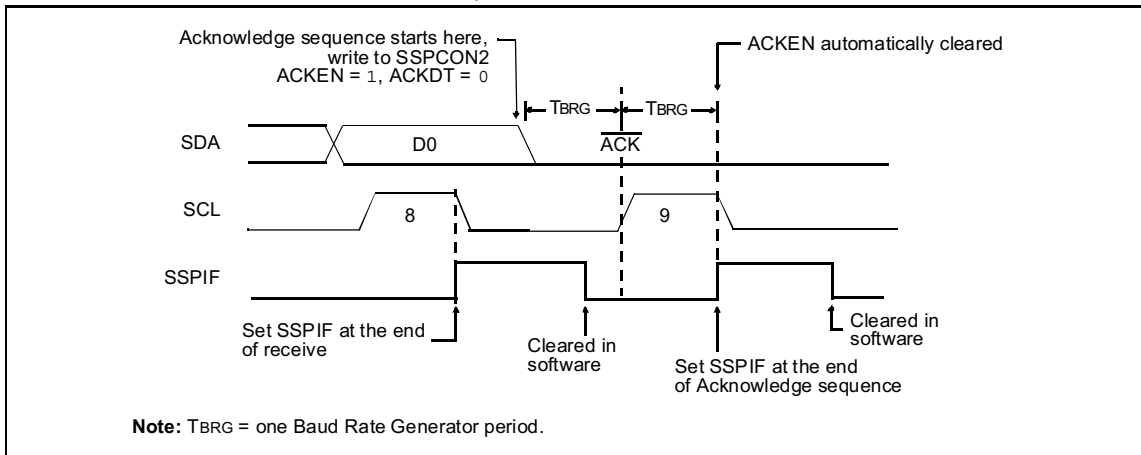
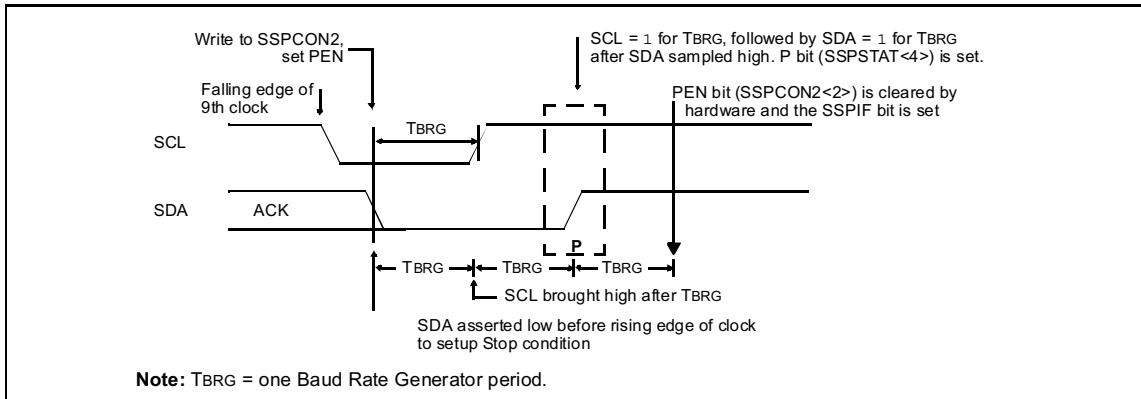


FIGURE 9-24: STOP CONDITION RECEIVE OR TRANSMIT MODE



9.4.14 SLEEP OPERATION

While in Sleep mode, the I²C module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

9.4.15 EFFECT OF A RESET

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

9.4.16 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 I²C bus may be taken when the P bit (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 at 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

9.4.17 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 I²C port to its Idle state (Figure 9-25).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted and the SSPBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I²C bus is free, the user can resume communication by asserting a Start condition.

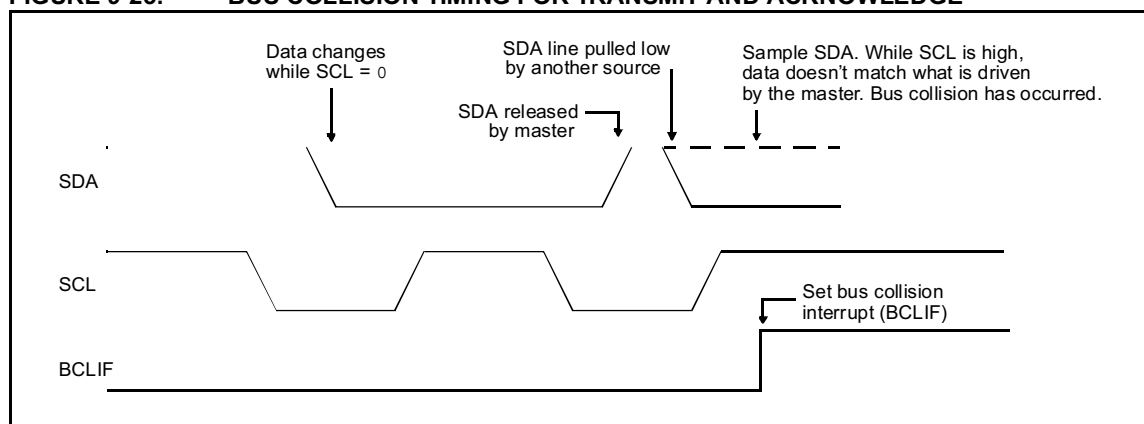
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the I²C bus is free, the user can resume communication by asserting a Start condition.

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

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

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I²C 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 9-25: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE



PIC16F87XA

9.4.17.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

- SDA or SCL are sampled low at the beginning of the Start condition (Figure 9-26).
- SCL is sampled low before SDA is asserted low (Figure 9-27).

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 all of the following occur:

- the Start condition is aborted,
- the BCLIF flag is set and
- the MSSP module is reset to its Idle state (Figure 9-26).

The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded from SSPADD<6:0> and counts down to 0. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 9-28). 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 pin 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. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.

FIGURE 9-26: BUS COLLISION DURING START CONDITION (SDA ONLY)

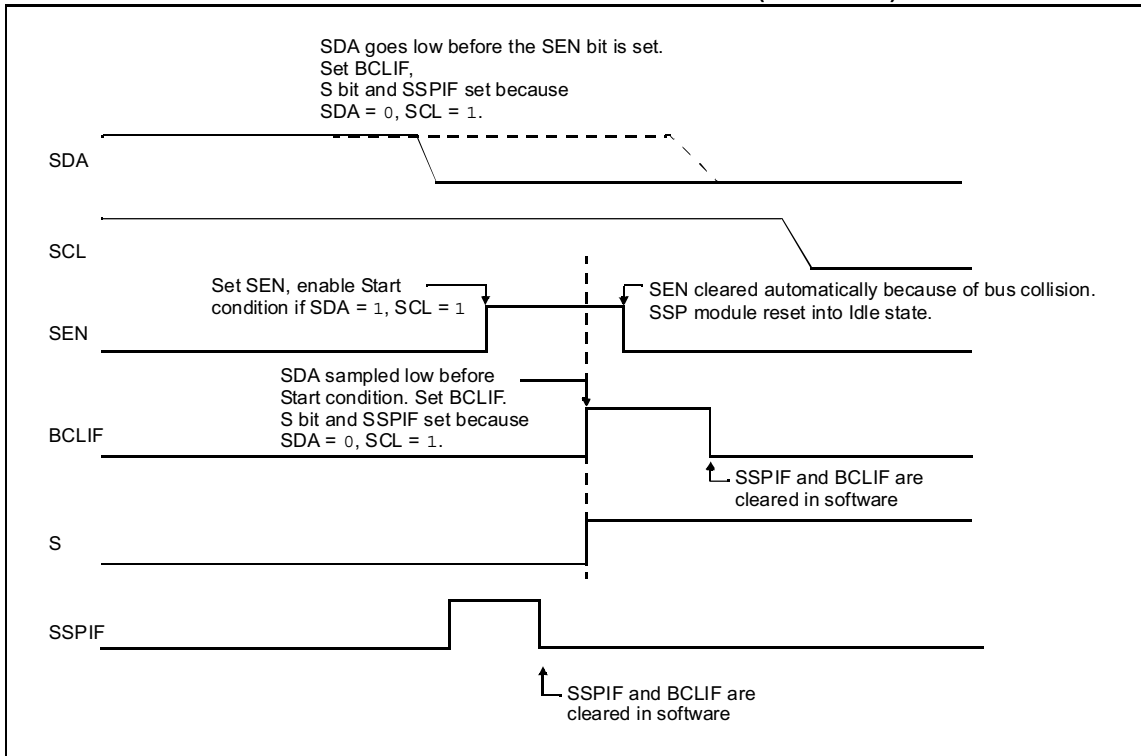


FIGURE 9-27: BUS COLLISION DURING START CONDITION (SCL = 0)

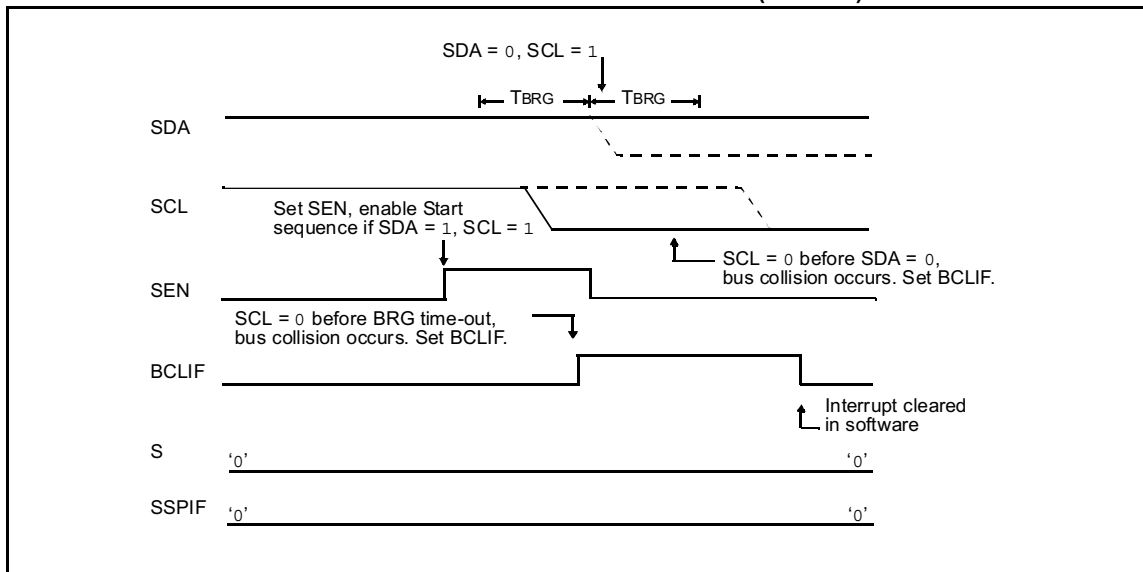
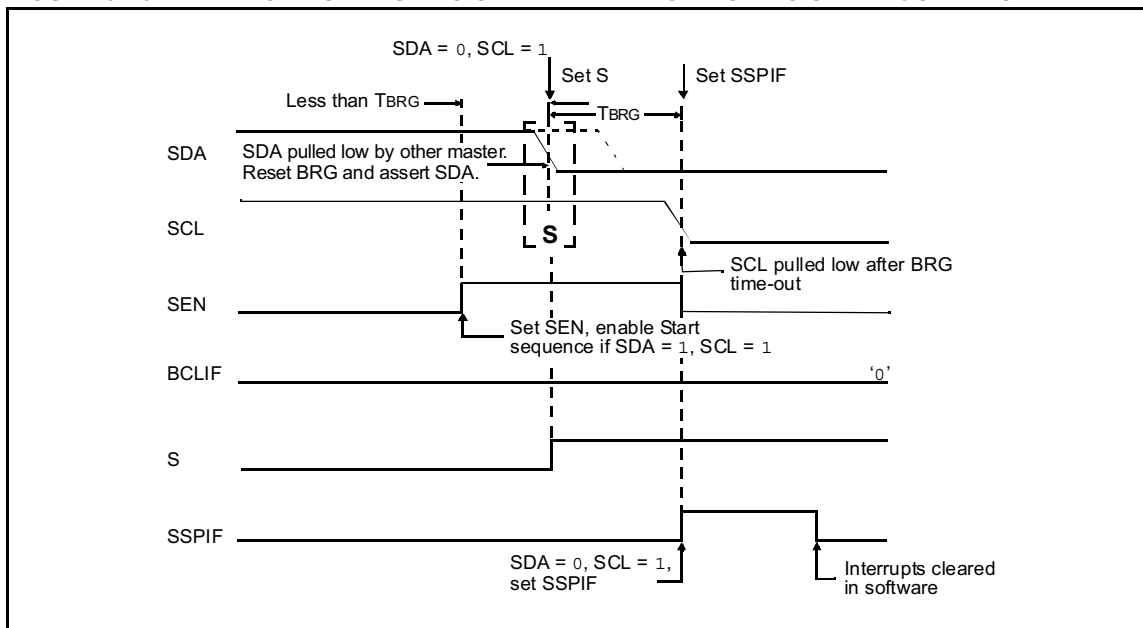


FIGURE 9-28: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION



PIC16F87XA

9.4.17.2 Bus Collision During a Repeated Start Condition

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

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

When the user deasserts SDA and the pin is allowed to float high, the BRG is loaded with SSPADD<6:0> and counts down to 0. The SCL pin is then deasserted and when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', see Figure 9-29). If SDA is sampled high, the BRG is

reloaded and begins counting. If SDA goes from high to low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

If SCL goes from high to low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition (Figure 9-30).

If at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and 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 9-29: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

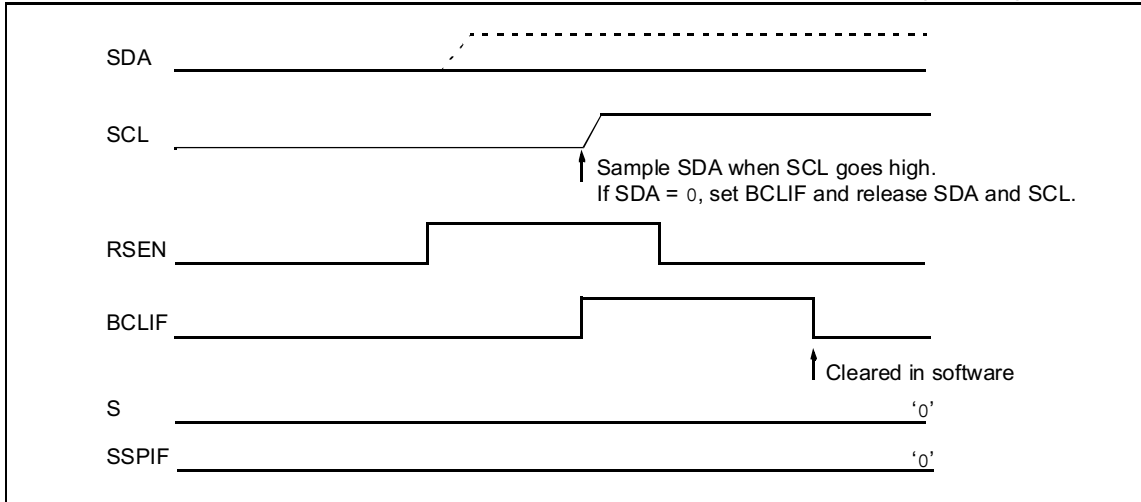
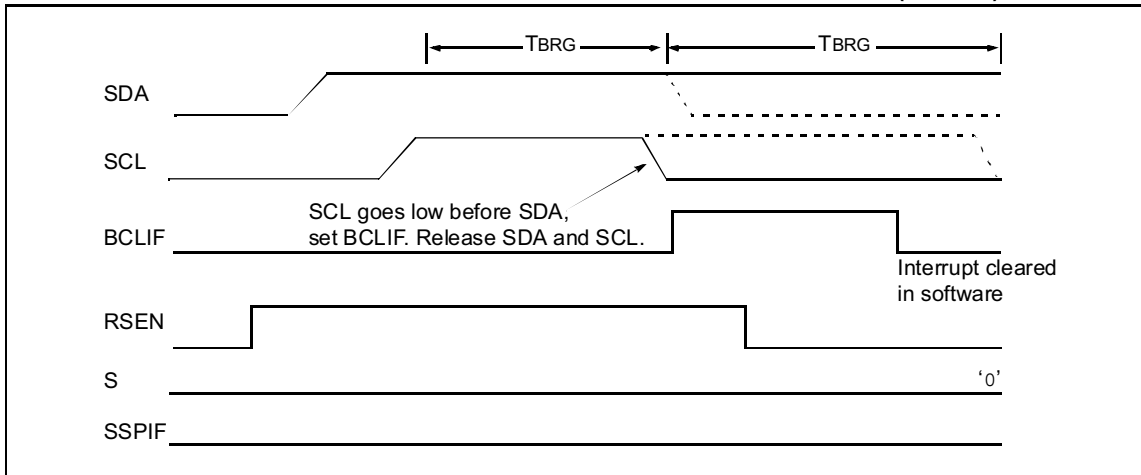


FIGURE 9-30: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)



9.4.17.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

- After the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out.
- After the SCL pin is deasserted, SCL is sampled low before SDA is allowed to float 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' (Figure 9-31). 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 9-32).

FIGURE 9-31: BUS COLLISION DURING A STOP CONDITION (CASE 1)

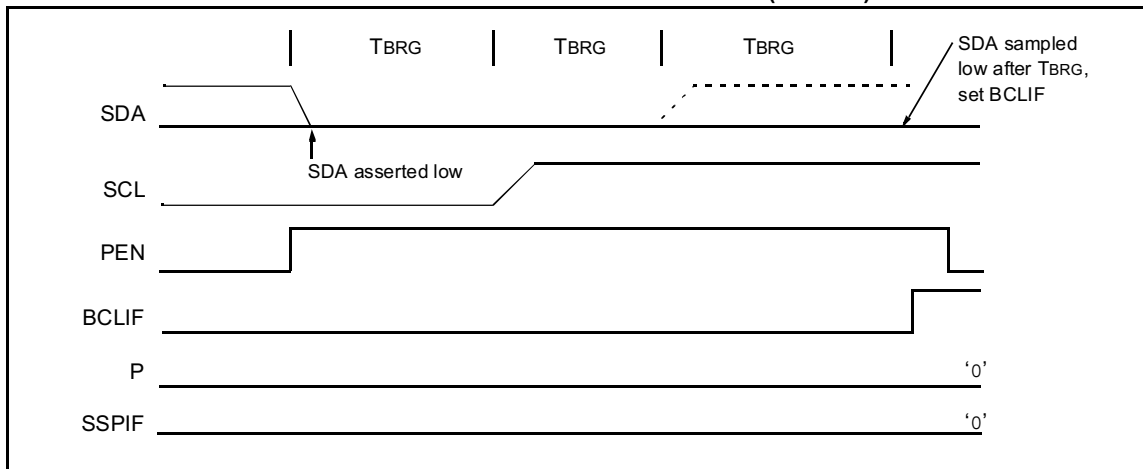
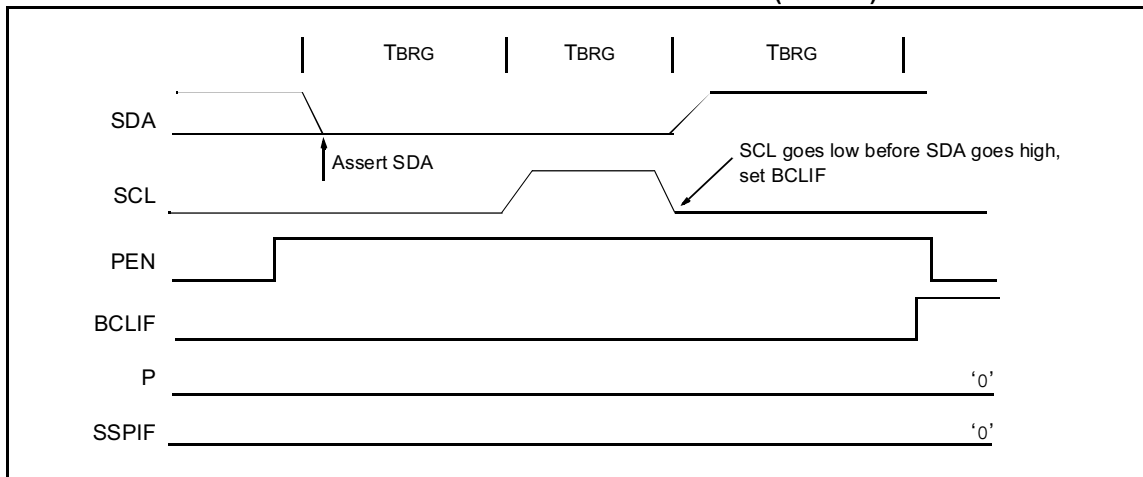


FIGURE 9-32: BUS COLLISION DURING A STOP CONDITION (CASE 2)



PIC16F87XA

NOTES:

10.0 ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

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

The USART can be configured in the following modes:

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

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

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

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

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
							bit 0
							bit 7

- bit 7 **CSRC:** Clock Source Select bit
Asynchronous mode:
 Don't care.
Synchronous mode:
 1 = Master mode (clock generated internally from BRG)
 0 = Slave mode (clock from external source)
- bit 6 **TX9:** 9-bit Transmit Enable bit
 1 = Selects 9-bit transmission
 0 = Selects 8-bit transmission
- bit 5 **TXEN:** Transmit Enable bit
 1 = Transmit enabled
 0 = Transmit disabled
Note: SREN/CREN overrides TXEN in Sync mode.
- bit 4 **SYNC:** USART Mode Select bit
 1 = Synchronous mode
 0 = Asynchronous mode
- bit 3 **Unimplemented:** Read as '0'
- bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
 1 = High speed
 0 = Low speed
Synchronous mode:
 Unused in this mode.
- bit 1 **TRMT:** Transmit Shift Register Status bit
 1 = TSR empty
 0 = TSR full
- bit 0 **TX9D:** 9th bit of Transmit Data, can be Parity bit

Legend:

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

PIC16F87XA

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

- 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:
Don't care.
- 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, enables interrupt and load of the receive buffer 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 but must be calculated by user firmware)

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

10.1 USART Baud Rate Generator (BRG)

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

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

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

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

10.1.1 SAMPLING

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

TABLE 10-1: BAUD RATE FORMULA

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

Legend: 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.

PIC16F87XA

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

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

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207	0.3	0	191
1.2	1.202	0.17	51	1.2	0	47
2.4	2.404	0.17	25	2.4	0	23
9.6	8.929	6.99	6	9.6	0	5
19.2	20.833	8.51	2	19.2	0	2
28.8	31.250	8.51	1	28.8	0	1
33.6	-	-	-	-	-	-
57.6	62.500	8.51	0	57.6	0	0
HIGH	0.244	-	255	0.225	-	255
LOW	62.500	-	0	57.6	-	0

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

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

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-
1.2	1.202	0.17	207	1.2	0	191
2.4	2.404	0.17	103	2.4	0	95
9.6	9.615	0.16	25	9.6	0	23
19.2	19.231	0.16	12	19.2	0	11
28.8	27.798	3.55	8	28.8	0	7
33.6	35.714	6.29	6	32.9	2.04	6
57.6	62.500	8.51	3	57.6	0	3
HIGH	0.977	-	255	0.9	-	255
LOW	250.000	-	0	230.4	-	0

10.2 USART Asynchronous Mode

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

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

The USART Asynchronous module consists of the following important elements:

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

10.2.1 USART ASYNCHRONOUS TRANSMITTER

The USART transmitter block diagram is shown in Figure 10-1. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The shift register obtains its data from the Read/Write Transmit Buffer, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXREG register (if available). Once the TXREG register transfers the data to the TSR register (occurs in one Tcy), the TXREG register is empty and flag bit, TXIF (PIR1<4>), is set. This interrupt can be

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

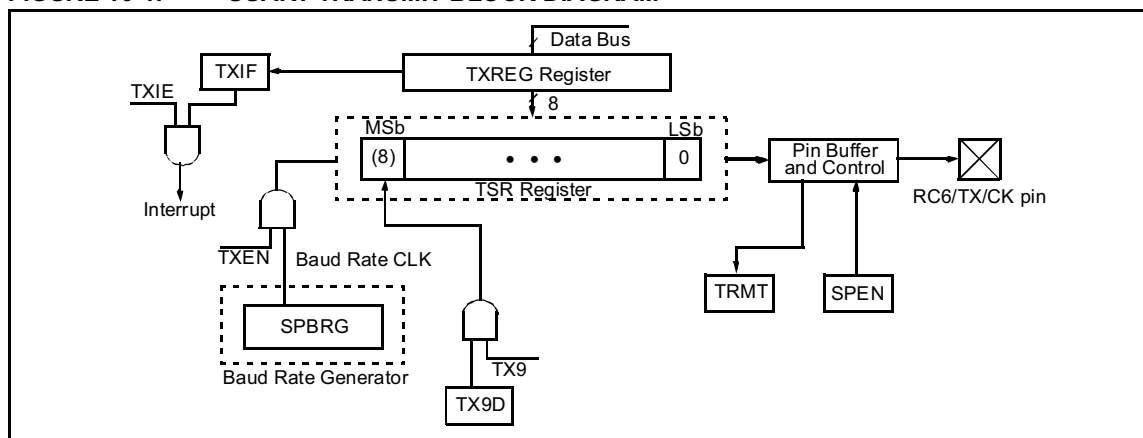
- Note 1:** The TSR register is not mapped in data memory so it is not available to the user.

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

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

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

FIGURE 10-1: USART TRANSMIT BLOCK DIAGRAM



PIC16F87XA

When setting up an Asynchronous Transmission, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH (Section 10.1 "USART Baud Rate Generator (BRG)").
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).
8. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

FIGURE 10-2: ASYNCHRONOUS MASTER TRANSMISSION

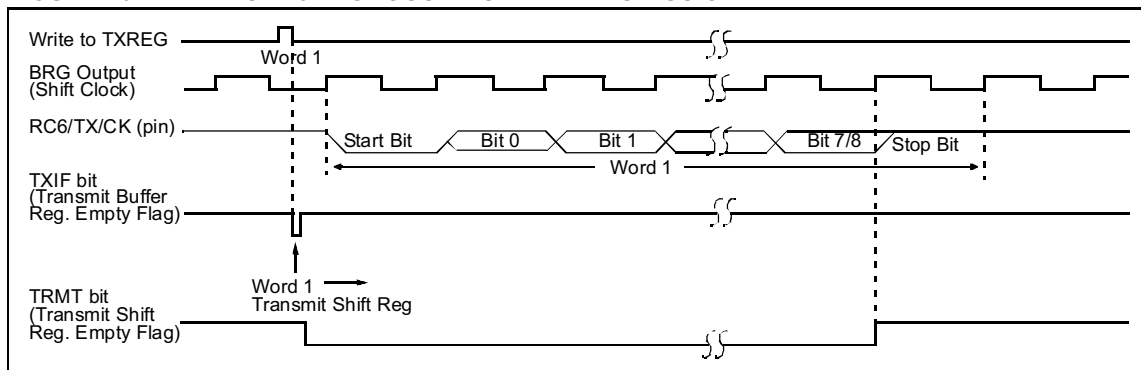


FIGURE 10-3: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)

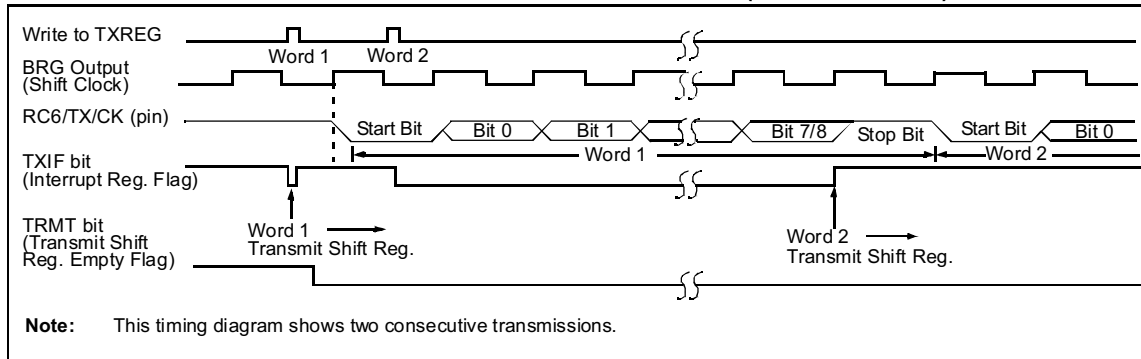


TABLE 10-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	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 28-pin devices; always maintain these bits clear.

10.2.2 USART ASYNCHRONOUS RECEIVER

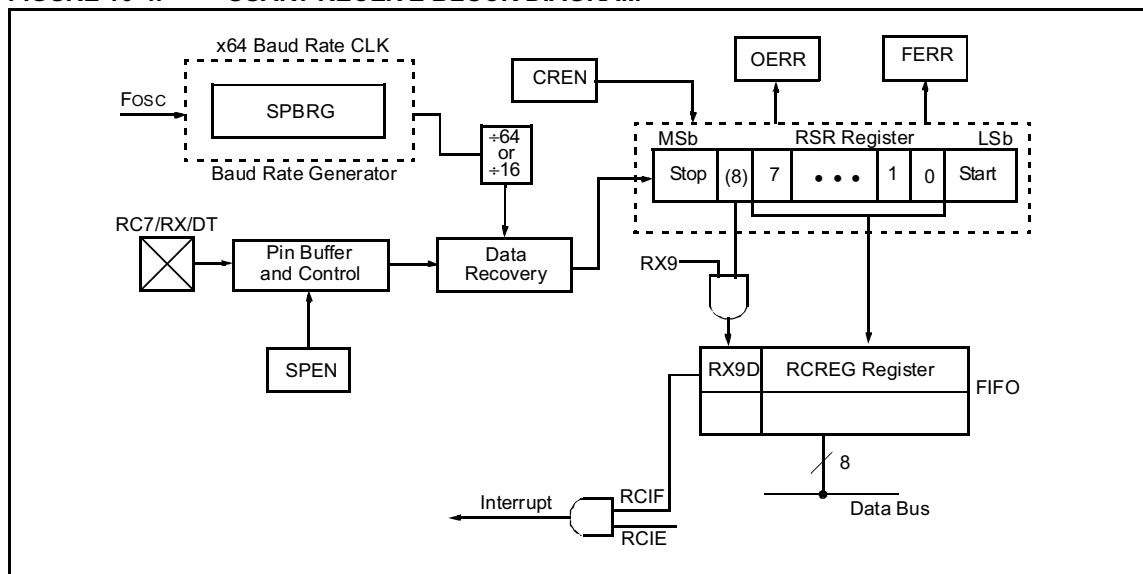
The receiver block diagram is shown in Figure 10-4. The data is received on the RC7/RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter, operating at x16 times the baud rate; whereas the main receive serial shifter operates at the bit rate or at FOSC.

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

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

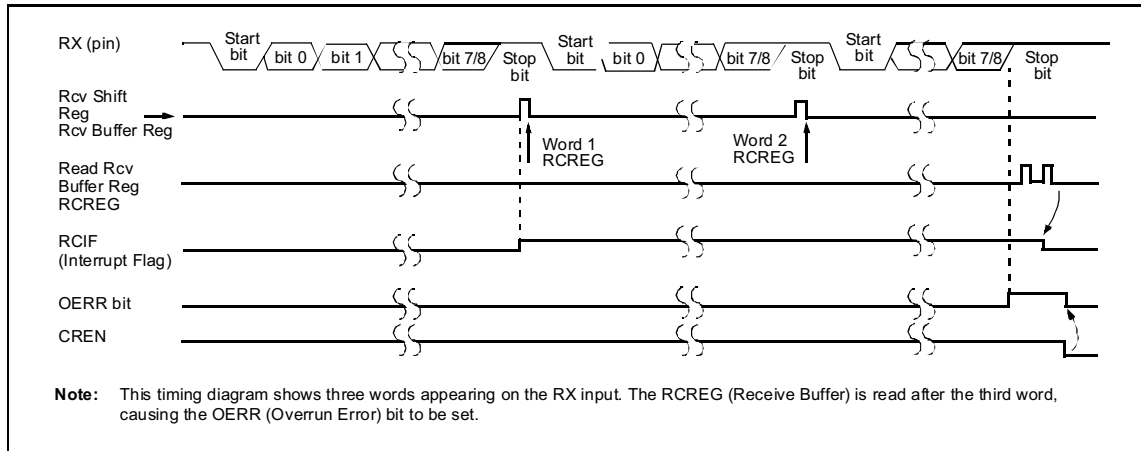
is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting to the RSR register. On the detection of the Stop bit of the third byte, if the RCREG register is still full, the Overrun Error bit, OERR (RCSTA<1>), will be set. The word in the RSR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

FIGURE 10-4: USART RECEIVE BLOCK DIAGRAM



PIC16F87XA

FIGURE 10-5: ASYNCHRONOUS RECEPTION



When setting up an Asynchronous Reception, follow these steps:

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

TABLE 10-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	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.

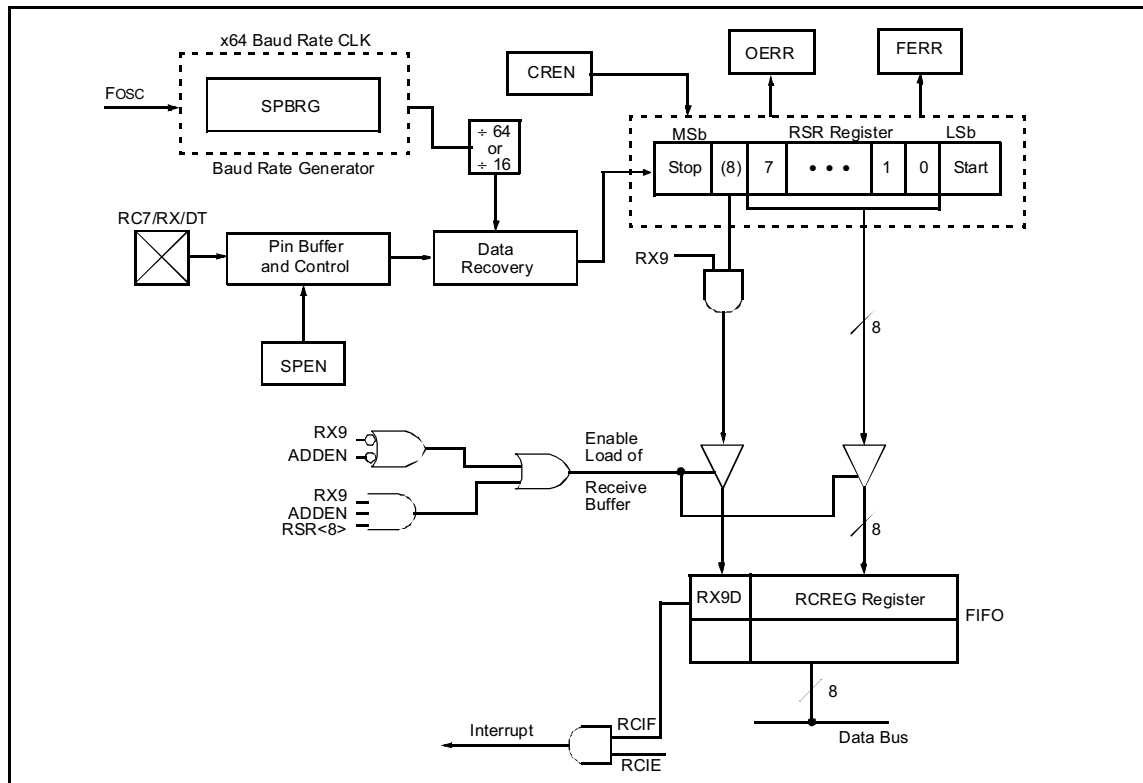
Note 1: Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

10.2.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

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



PIC16F87XA

FIGURE 10-7: ASYNCHRONOUS RECEPTION WITH ADDRESS DETECT

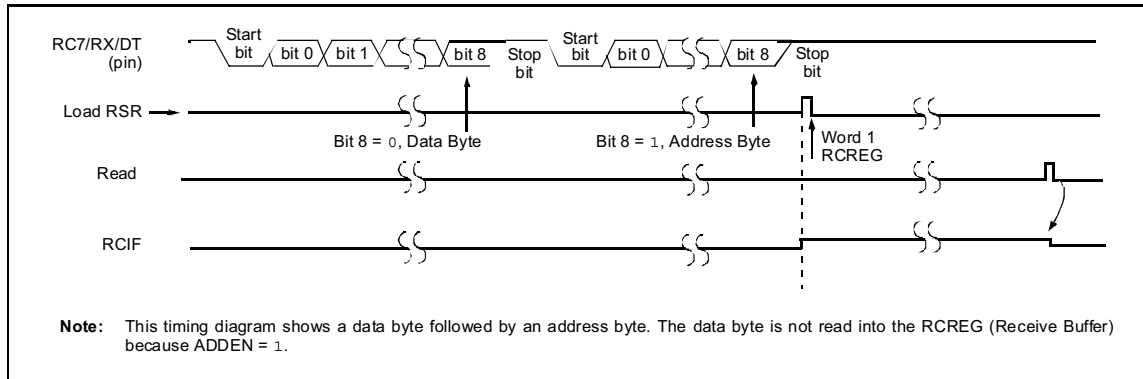


FIGURE 10-8: ASYNCHRONOUS RECEPTION WITH ADDRESS BYTE FIRST

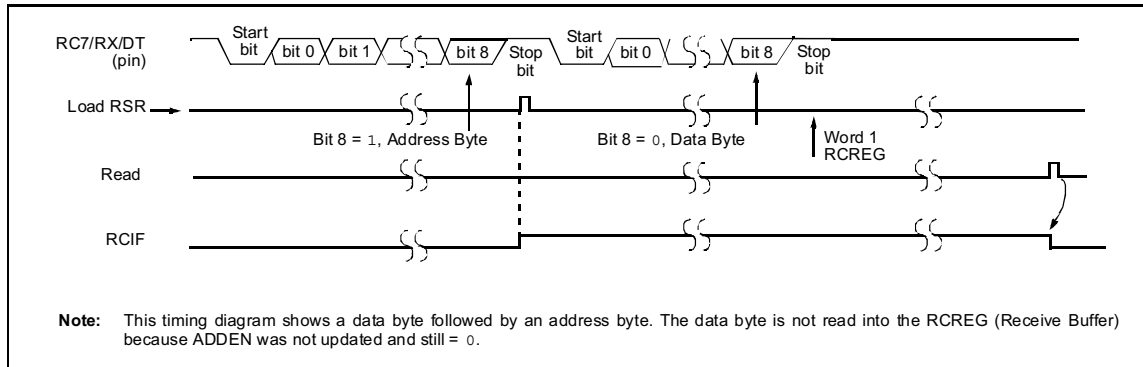


TABLE 10-7: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

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
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	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 ⁽¹⁾	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 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 10-6. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The shift register obtains its data from the Read/Write Transmit Buffer register, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once the TXREG register transfers the data to the TSR register (occurs in one TCYCLE), the TXREG is empty and interrupt bit, TXIF (PIR1<4>), is set. The interrupt can be enabled/disabled by setting/clearing enable bit TXIE (PIE1<4>). Flag bit TXIF will be set regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR is empty. No interrupt logic is tied to this bit so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory so it is not available to the user.

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

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

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

Steps to follow when setting up a Synchronous Master Transmission:

1. Initialize the SPBRG register for the appropriate baud rate (**Section 10.1 "USART Baud Rate Generator (BRG)"**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set bit TX9.
5. Enable the transmission by setting bit TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG register.
8. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCN register are set.

PIC16F87XA

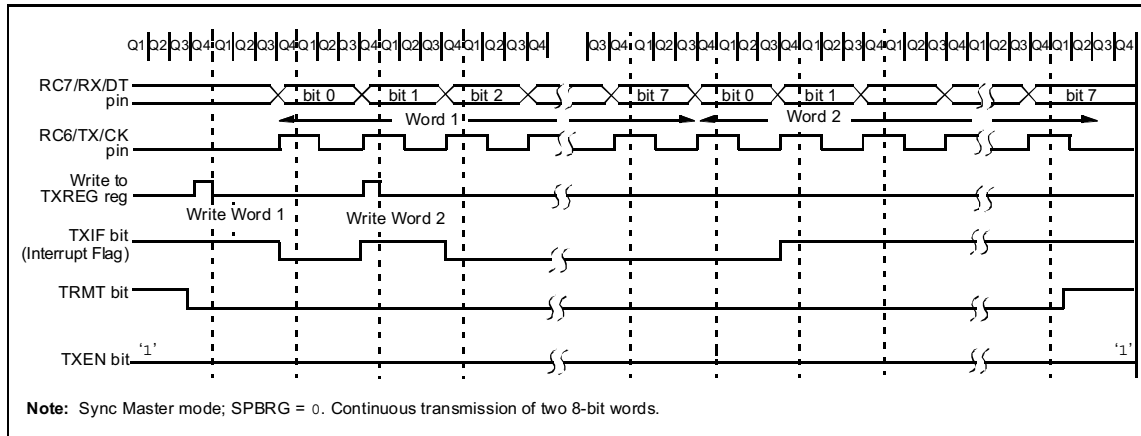
TABLE 10-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

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
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	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.

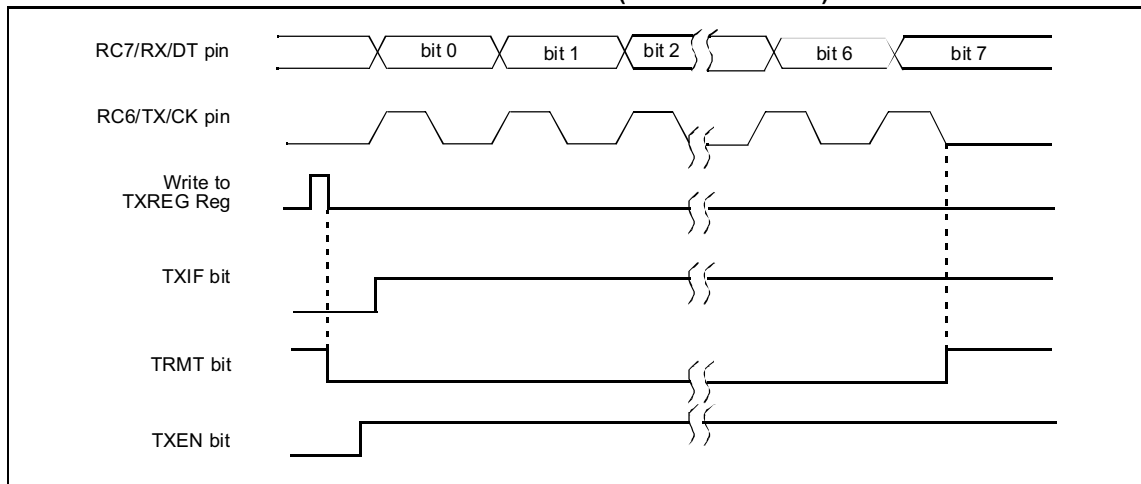
Note 1: Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

FIGURE 10-9: SYNCHRONOUS TRANSMISSION



Note: Sync Master mode; SPBRG = 0. Continuous transmission of two 8-bit words.

FIGURE 10-10: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)



10.3.2 USART SYNCHRONOUS MASTER RECEPTION

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

data. Reading the RCREG register will load bit RX9D with a new value, therefore, it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old RX9D information.

When setting up a Synchronous Master Reception:

1. Initialize the SPBRG register for the appropriate baud rate (**Section 10.1 “USART Baud Rate Generator (BRG)”**).
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.
11. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

TABLE 10-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

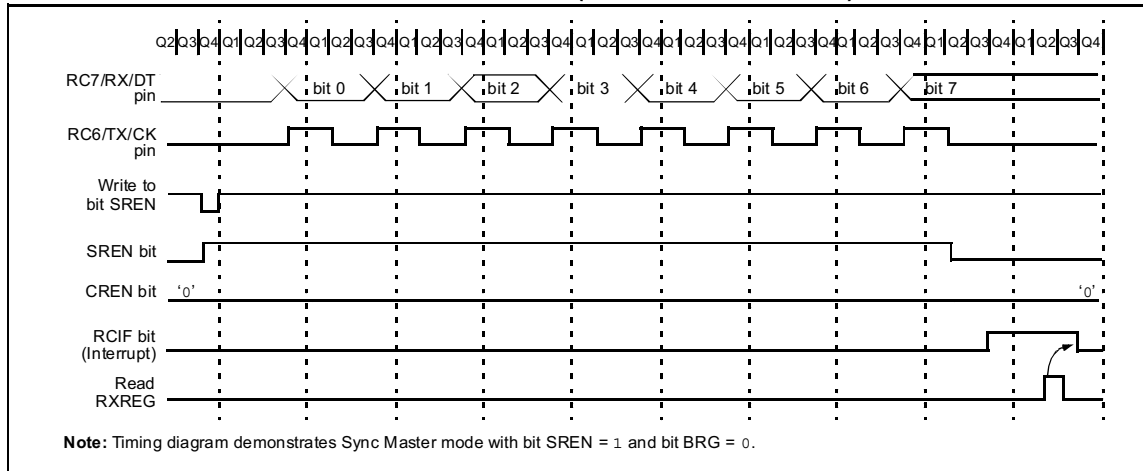
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	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 28-pin devices; always maintain these bits clear.

PIC16F87XA

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 is 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:

- The first word will immediately transfer to the TSR register and transmit.
- The second word will remain in TXREG register.
- Flag bit TXIF will not be set.
- 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.
- 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).

When setting up a Synchronous Slave Transmission, follow these steps:

- Enable the synchronous slave serial port by setting bits SYNC and SPEN and clearing bit CSRC.
- Clear bits CREN and SREN.
- If interrupts are desired, then set enable bit TXIE.
- If 9-bit transmission is desired, then set bit TX9.
- Enable the transmission by setting enable bit TXEN.
- If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
- Start transmission by loading data to the TXREG register.
- If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

PIC16F87XA

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
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	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 ⁽¹⁾	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 28-pin devices; always maintain these bits clear.

10.4.2 USART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of the Sleep mode. Bit SREN is a “don't care” in Slave mode.

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

When setting up a Synchronous Slave Reception, follow these steps:

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

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
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	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 ⁽¹⁾	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 28-pin devices; always maintain these bits clear.

PIC16F87XA

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 40/44-pin devices.

The conversion of an analog input signal results in a corresponding 10-bit digital number. The A/D module has high and low-voltage reference input that is software selectable to some combination of VDD, VSS, RA2 or RA3.

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

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

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)

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

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

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7							bit 0

bit 7-6 **ADCS1:ADCS0**: A/D Conversion Clock Select bits (ADCON0 bits in **bold**)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

bit 5-3 **CHS2:CHS0**: Analog Channel Select bits

000 = Channel 0 (AN0)
 001 = Channel 1 (AN1)
 010 = Channel 2 (AN2)
 011 = Channel 3 (AN3)
 100 = Channel 4 (AN4)
 101 = Channel 5 (AN5)
 110 = Channel 6 (AN6)
 111 = Channel 7 (AN7)

Note: The PIC16F873A/876A devices only implement A/D channels 0 through 4; the unimplemented selections are reserved. Do not select any unimplemented channels with these devices.

bit 2 **GO/DONE**: A/D Conversion Status bit

When ADON = 1:

1 = A/D conversion in progress (setting this bit starts the A/D conversion which is automatically cleared by hardware when the A/D conversion is complete)

0 = A/D conversion not in progress

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

bit 0 **ADON**: A/D On bit

1 = A/D converter module is powered up

0 = A/D converter module is shut-off and consumes no operating current

Legend:

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

PIC16F87XA

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

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7				bit 0			

bit 7 **ADFM:** A/D Result Format Select bit

1 = Right justified. Six (6) Most Significant bits of ADRESH are read as '0'.
 0 = Left justified. Six (6) Least Significant bits of ADRESL are read as '0'.

bit 6 **ADCS2:** A/D Conversion Clock Select bit (ADCON1 bits in shaded area and in **bold**)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	FOSC/2
0	01	FOSC/8
0	10	FOSC/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	FOSC/4
1	01	FOSC/16
1	10	FOSC/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

bit 5-4 **Unimplemented:** Read as '0'

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

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	AN3	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	AN3	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	AN3	VSS	2/1
011x	D	D	D	D	D	D	D	D	—	—	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1/2

A = Analog input D = Digital I/O

C/R = # of analog input channels/# of A/D voltage references

Legend:

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

Note: On any device Reset, the port pins that are multiplexed with analog functions (ANx) are forced to be an analog input.

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

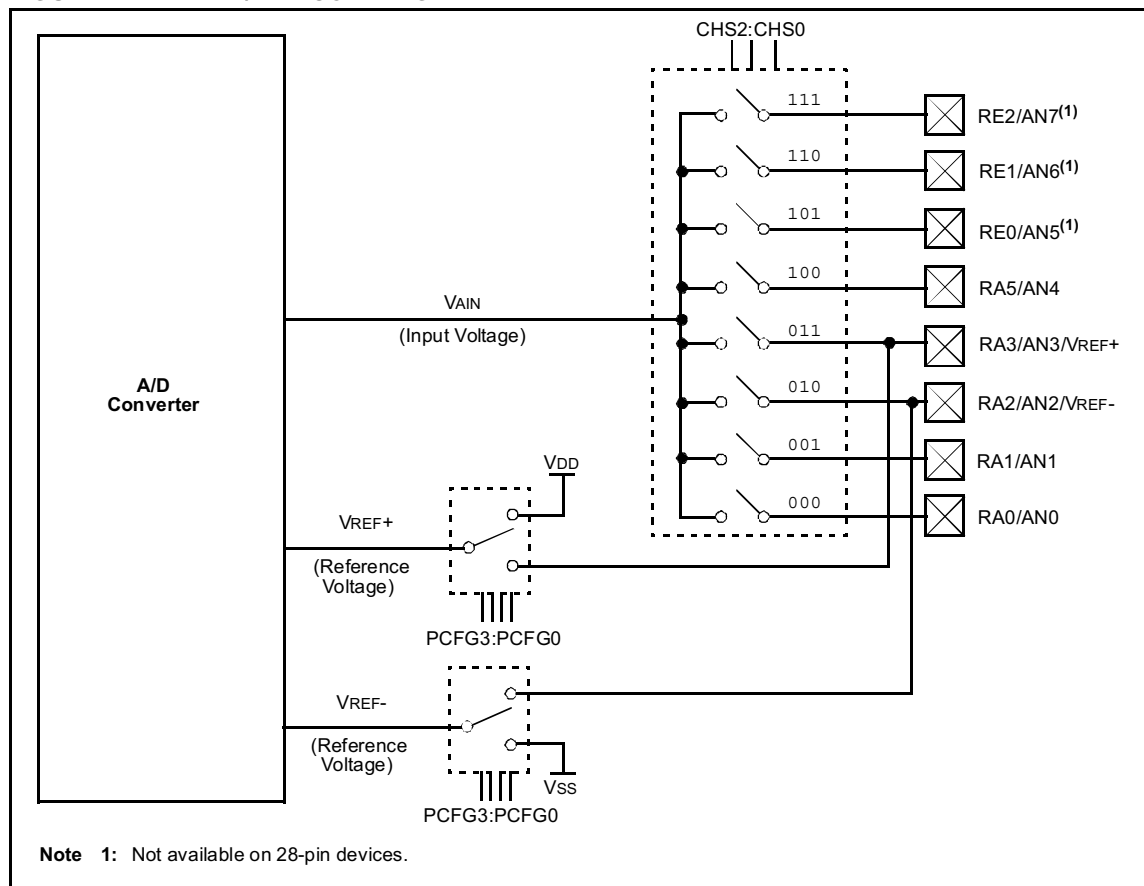
After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs.

To determine sample time, see **Section 11.1 “A/D Acquisition Requirements”**. After this acquisition time has elapsed, the A/D conversion can be started.

To do an A/D Conversion, follow these steps:

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 PEIE bit
 - Set GIE bit
3. Wait the required acquisition time.
4. Start conversion:
 - Set $\overline{\text{GO/DONE}}$ bit (ADCON0)
5. Wait for A/D conversion to complete by either:
 - Polling for the $\overline{\text{GO/DONE}}$ bit to be cleared (interrupts disabled); OR
 - Waiting for the A/D interrupt
6. Read A/D Result register pair (ADRESH:ADRESL), clear bit ADIF if required.
7. For the next conversion, go to step 1 or step 2 as required. The A/D conversion time per bit is defined as TAD.

FIGURE 11-1: A/D BLOCK DIAGRAM



PIC16F87XA

11.1 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 11-2. The source impedance (RS) and the internal sampling switch impedance (RSS) directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD); see Figure 11-2. **The maximum recommended impedance for analog sources is 2.5 kΩ**. As the impedance is decreased, the acquisition time may be

decreased. After the analog input channel is selected (changed), this acquisition must be done before the conversion can be started.

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

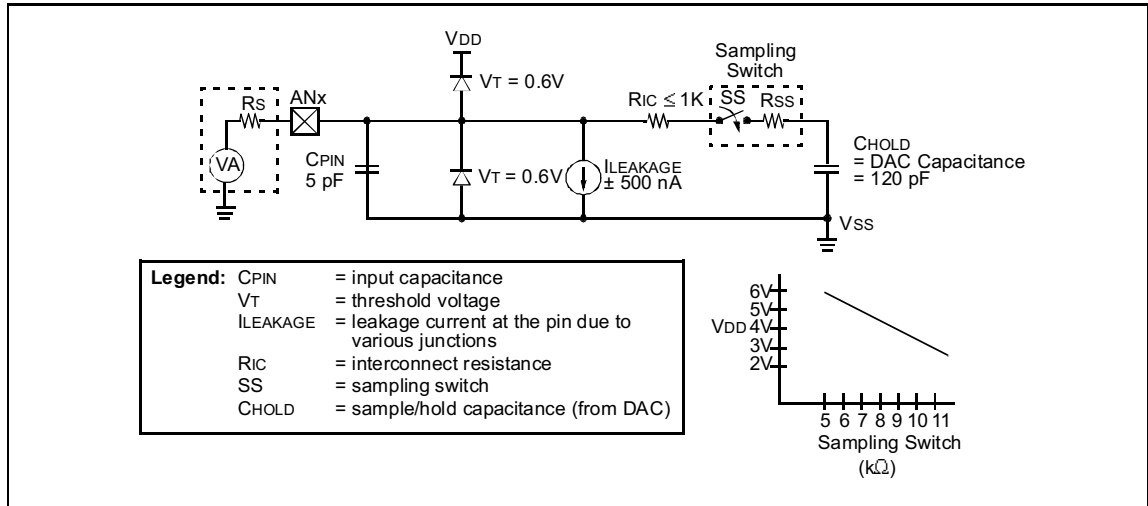
To calculate the minimum acquisition time, TACQ, see the PICmicro® Mid-Range MCU Family Reference Manual (DS33023).

EQUATION 11-1: ACQUISITION TIME

TACQ	=	Amplifier Settling Time + Hold Capacitor Charging Time + Temperature Coefficient
	=	TAMP + TC + TCOFF
	=	2 μs + TC + [(Temperature - 25°C)(0.05 μs/°C)]
TC	=	CHOLD (RIC + RSS + RS) ln(1/2047)
	=	- 120 pF (1 kΩ + 7 kΩ + 10 kΩ) ln(0.0004885)
	=	16.47 μs
TACQ	=	2 μs + 16.47 μs + [(50°C - 25°C)(0.05 μs/°C)]
	=	19.72 μs

- Note 1:** The reference voltage (VREF) has no effect on the equation since it cancels itself out.
- 2:** The charge holding capacitor (CHOLD) is not discharged after each conversion.
- 3:** The maximum recommended impedance for analog sources is 2.5 kΩ. This is required to meet the pin leakage specification.

FIGURE 11-2: 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 12 TAD per 10-bit conversion. The source of the A/D conversion clock is software selected. The seven possible options for TAD are:

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- Internal A/D module RC oscillator (2-6 μ s)

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

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

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.

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

TABLE 11-1: TAD vs. MAXIMUM DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (F))

AD Clock Source (TAD)		Maximum Device Frequency
Operation	ADCS2:ADCS1:ADCS0	
2 TOSC	000	1.25 MHz
4 TOSC	100	2.5 MHz
8 TOSC	001	5 MHz
16 TOSC	101	10 MHz
32 TOSC	010	20 MHz
64 TOSC	110	20 MHz
RC ^(1, 2, 3)	x11	(Note 1)

Note 1: The RC source has a typical TAD time of 4 μ s but can vary between 2-6 μ s.

2: When the device frequencies are greater than 1 MHz, the RC A/D conversion clock source is only recommended for Sleep operation.

3: For extended voltage devices (LF), please refer to **Section 17.0 “Electrical Characteristics”**.

PIC16F87XA

11.4 A/D Conversions

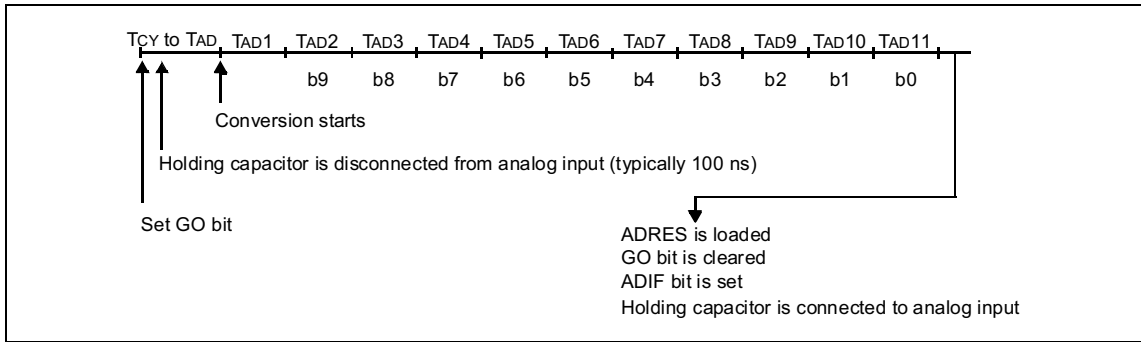
Clearing the $\overline{GO/DONE}$ bit during a conversion will abort the current conversion. The A/D Result register pair will NOT be updated with the partially completed A/D conversion sample. That is, the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or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

is aborted, the next acquisition on the selected channel is automatically started. The $\overline{GO/DONE}$ bit can then be set to start the conversion.

In Figure 11-3, after the GO bit is set, the first time segment has a minimum of T_{CY} and a maximum of T_{AD} .

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

FIGURE 11-3: A/D CONVERSION T_{AD} CYCLES

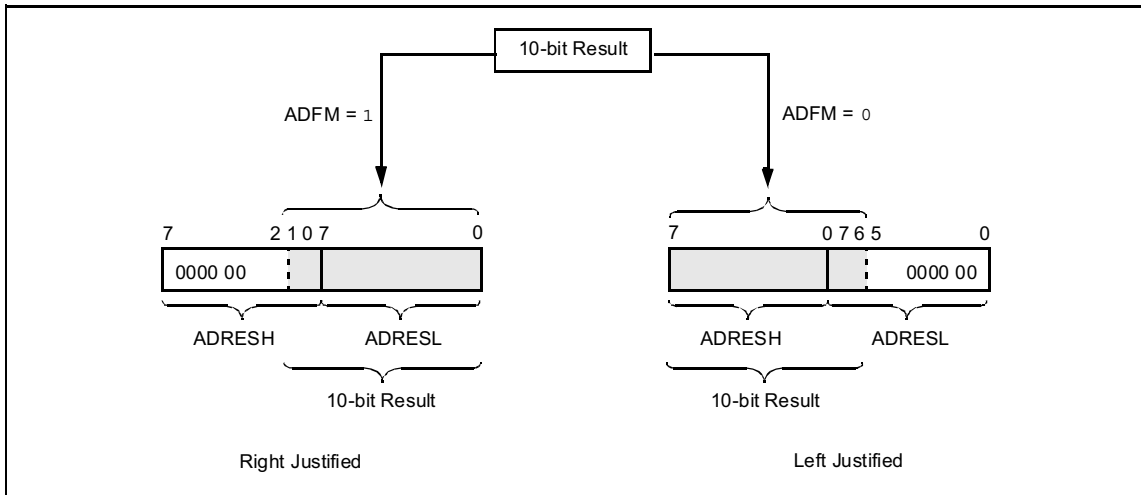


11.4.1 A/D RESULT REGISTERS

The ADRESH:ADRESL register pair is the location where the 10-bit A/D result is loaded at the completion of the A/D conversion. This register pair is 16 bits wide. The A/D module gives the flexibility to left or right justify the 10-bit result in the 16-bit result register. The A/D

Format Select bit (ADFM) controls this justification. Figure 11-4 shows the operation of the A/D result justification. The extra bits are loaded with '0's. When an A/D result will not overwrite these locations (A/D disable), these registers may be used as two general purpose 8-bit registers.

FIGURE 11-4: A/D RESULT JUSTIFICATION



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. All A/D input pins are configured as analog inputs.

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.

TABLE 11-2: REGISTERS/BITS ASSOCIATED WITH A/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 MCLR, WDT
0Bh,8Bh,10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	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
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON	0000 00-0	0000 00-0
9Fh	ADCON1	ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0	00-- 0000	00-- 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 ⁽¹⁾	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction bits			0000 -111	0000 -111
09h ⁽¹⁾	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 are not available on 28-pin devices.

PIC16F87XA

NOTES:

12.0 COMPARATOR MODULE

The comparator module contains two analog comparators. The inputs to the comparators are multiplexed with I/O port pins RA0 through RA3, while the outputs are multiplexed to pins RA4 and RA5. The on-chip voltage reference (Section 13.0 “Comparator Voltage Reference Module”) can also be an input to the comparators.

The CMCON register (Register 12-1) controls the comparator input and output multiplexers. A block diagram of the various comparator configurations is shown in Figure 12-1.

REGISTER 12-1: CMCON REGISTER

	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1
	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
	bit 7							bit 0
bit 7	C2OUT: Comparator 2 Output bit <u>When C2INV = 0:</u> 1 = C2 VIN+ > C2 VIN- 0 = C2 VIN+ < C2 VIN- <u>When C2INV = 1:</u> 1 = C2 VIN+ < C2 VIN- 0 = C2 VIN+ > C2 VIN-							
bit 6	C1OUT: Comparator 1 Output bit <u>When C1INV = 0:</u> 1 = C1 VIN+ > C1 VIN- 0 = C1 VIN+ < C1 VIN- <u>When C1INV = 1:</u> 1 = C1 VIN+ < C1 VIN- 0 = C1 VIN+ > C1 VIN-							
bit 5	C2INV: Comparator 2 Output Inversion bit 1 = C2 output inverted 0 = C2 output not inverted							
bit 4	C1INV: Comparator 1 Output Inversion bit 1 = C1 output inverted 0 = C1 output not inverted							
bit 3	CIS: Comparator Input Switch bit <u>When CM2:CM0 = 110:</u> 1 = C1 VIN- connects to RA3/AN3 C2 VIN- connects to RA2/AN2 0 = C1 VIN- connects to RA0/AN0 C2 VIN- connects to RA1/AN1							
bit 2	CM2:CM0: Comparator Mode bits Figure 12-1 shows the Comparator modes and CM2:CM0 bit settings.							

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

PIC16F87XA

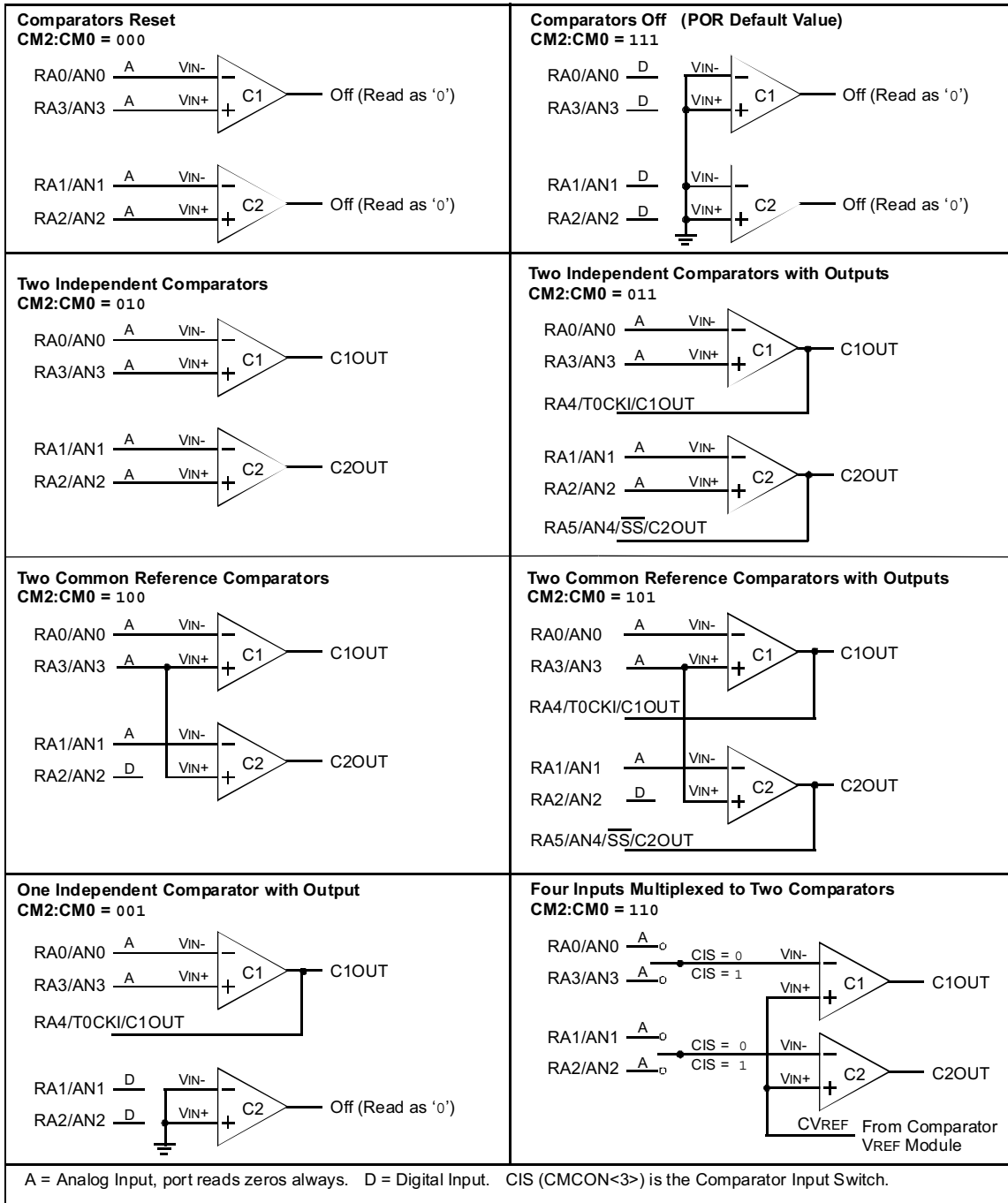
12.1 Comparator Configuration

There are eight modes of operation for the comparators. The CMCON register is used to select these modes. Figure 12-1 shows the eight possible modes. The TRISA register controls the data direction of the comparator pins for each mode. If the Comparator

mode is changed, the comparator output level may not be valid for the specified mode change delay shown in Section 17.0 "Electrical Characteristics".

Note: Comparator interrupts should be disabled during a Comparator mode change. Otherwise, a false interrupt may occur.

FIGURE 12-1: COMPARATOR I/O OPERATING MODES



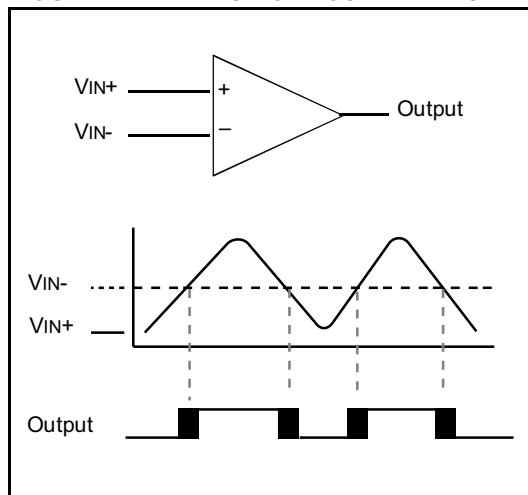
12.2 Comparator Operation

A single comparator is shown in Figure 12-2 along with the relationship between the analog input levels and the digital output. When the analog input at V_{IN+} is less than the analog input V_{IN-} , the output of the comparator is a digital low level. When the analog input at V_{IN+} is greater than the analog input V_{IN-} , the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 12-2 represent the uncertainty due to input offsets and response time.

12.3 Comparator Reference

An external or internal reference signal may be used depending on the comparator operating mode. The analog signal present at V_{IN-} is compared to the signal at V_{IN+} and the digital output of the comparator is adjusted accordingly (Figure 12-2).

FIGURE 12-2: SINGLE COMPARATOR



12.3.1 EXTERNAL REFERENCE SIGNAL

When external voltage references are used, the comparator module can be configured to have the comparators operate from the same or different reference sources. However, threshold detector applications may require the same reference. The reference signal must be between V_{SS} and V_{DD} and can be applied to either pin of the comparator(s).

12.3.2 INTERNAL REFERENCE SIGNAL

The comparator module also allows the selection of an internally generated voltage reference for the comparators. **Section 13.0 “Comparator Voltage Reference Module”** contains a detailed description of the Comparator Voltage Reference module that provides this signal. The internal reference signal is used when comparators are in mode, $CM<2:0> = 110$ (Figure 12-1). In this mode, the internal voltage reference is applied to the V_{IN+} pin of both comparators.

12.4 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output has a valid level. If the internal reference is changed, the maximum delay of the internal voltage reference must be considered when using the comparator outputs. Otherwise, the maximum delay of the comparators should be used (**Section 17.0 “Electrical Characteristics”**).

12.5 Comparator Outputs

The comparator outputs are read through the $CMCON$ register. These bits are read-only. The comparator outputs may also be directly output to the $RA4$ and $RA5$ I/O pins. When enabled, multiplexors in the output path of each pin will switch and the output of each pin will be the unsynchronized output of the comparator. The uncertainty of each of the comparators is related to the input offset voltage and the response time given in the specifications. Figure 12-3 shows the comparator output block diagram.

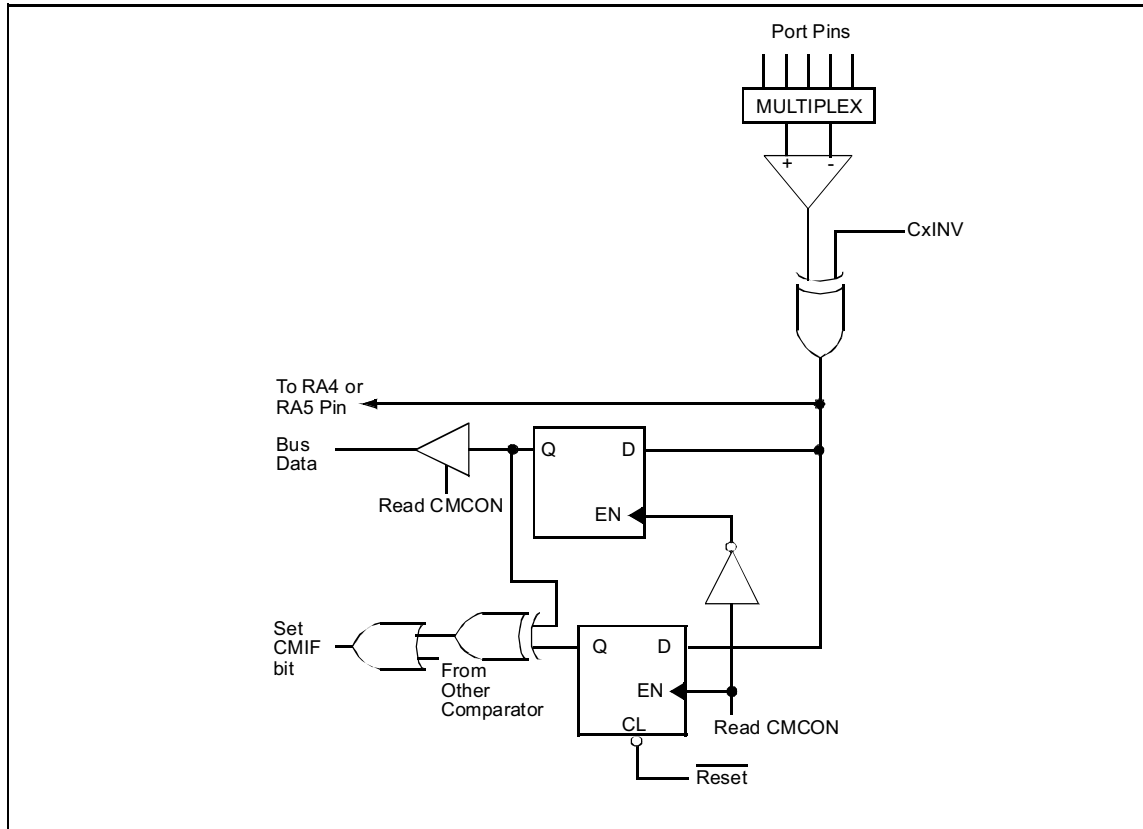
The $TRISA$ bits will still function as an output enable/disable for the $RA4$ and $RA5$ pins while in this mode.

The polarity of the comparator outputs can be changed using the $C2INV$ and $C1INV$ bits ($CMCON<4:5>$).

- Note 1:** When reading the Port register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert an analog input according to the Schmitt Trigger input specification.
- 2:** Analog levels on any pin defined as a digital input may cause the input buffer to consume more current than is specified.
- 3:** $RA4$ is an open collector I/O pin. When used as an output, a pull-up resistor is required.

PIC16F87XA

FIGURE 12-3: COMPARATOR OUTPUT BLOCK DIAGRAM



12.6 Comparator Interrupts

The comparator interrupt flag is set whenever there is a change in the output value of either comparator. Software will need to maintain information about the status of the output bits, as read from CMCON<7:6>, to determine the actual change that occurred. The CMIF bit (PIR registers) is the Comparator Interrupt Flag. The CMIF bit must be reset by clearing it ('0'). Since it is also possible to write a '1' to this register, a simulated interrupt may be initiated.

The CMIE bit (PIE registers) and the PEIE bit (INTCON register) must be set to enable the interrupt. In addition, the GIE bit must also be set. If any of these bits are clear, the interrupt is not enabled, though the CMIF bit will still be set if an interrupt condition occurs.

Note: If a change in the CMCON register (C1OUT or C2OUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CMIF (PIR registers) interrupt flag may not get set.

The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- Any read or write of CMCON will end the mismatch condition.
- Clear flag bit CMIF.

A mismatch condition will continue to set flag bit CMIF. Reading CMCON will end the mismatch condition and allow flag bit CMIF to be cleared.

12.7 Comparator Operation During Sleep

When a comparator is active and the device is placed in Sleep mode, the comparator remains active and the interrupt is functional if enabled. This interrupt will wake-up the device from Sleep mode when enabled. While the comparator is powered up, higher Sleep currents than shown in the power-down current specification will occur. Each operational comparator will consume additional current as shown in the comparator specifications. To minimize power consumption while in Sleep mode, turn off the comparators, $CM<2:0> = 111$, before entering Sleep. If the device wakes up from Sleep, the contents of the CMCON register are not affected.

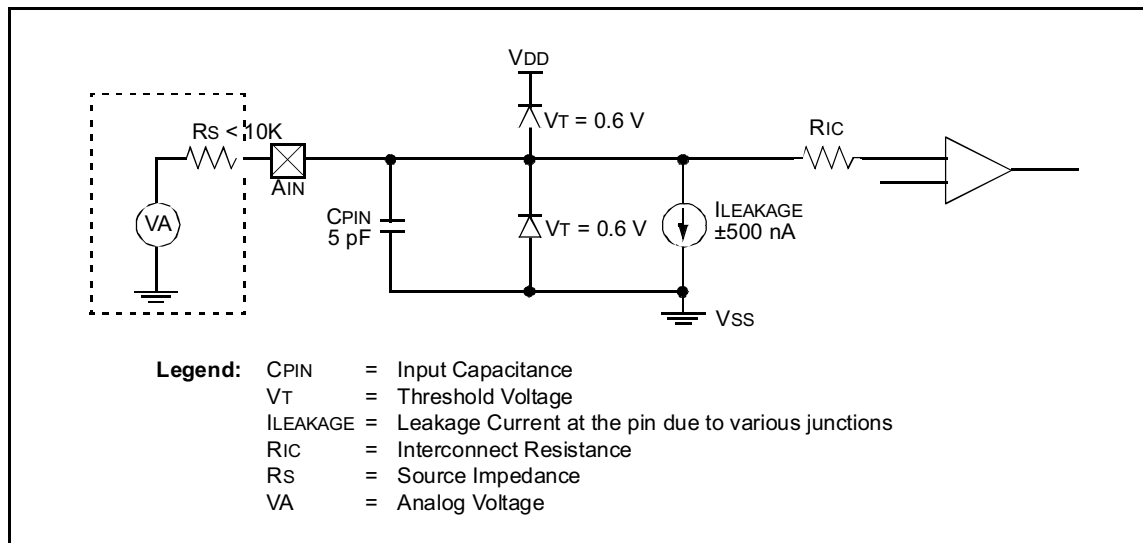
12.8 Effects of a Reset

A device Reset forces the CMCON register to its Reset state, causing the comparator module to be in the Comparator Off mode, $CM<2:0> = 111$. This ensures compatibility to the PIC16F87X devices.

12.9 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 12-4. Since the analog pins are connected to a digital output, they have reverse biased diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up condition may occur. A maximum source impedance of 10 k Ω is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.

FIGURE 12-4: ANALOG INPUT MODEL



PIC16F87XA

TABLE 12-1: REGISTERS ASSOCIATED WITH COMPARATOR MODULE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other Resets
9Ch	CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111	0000 0111
9Dh	CVRCON	CVREN	CVROE	CVRR	—	CVR3	CVR2	CVR1	CVR0	000- 0000	000- 0000
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INTIE	RBIE	TMR0IF	INTIF	RBIF	0000 000x	0000 000u
0Dh	PIR2	—	CMIF	—	—	BCLIF	LVDIF	TMR3IF	CCP2IF	-0-- 0000	-0-- 0000
8Dh	PIE2	—	CMIE	—	—	BCLIE	LVDIE	TMR3IE	CCP2IE	-0-- 0000	-0-- 0000
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are unused by the comparator module.

13.0 COMPARATOR VOLTAGE REFERENCE MODULE

The Comparator Voltage Reference Generator is a 16-tap resistor ladder network that provides a fixed voltage reference when the comparators are in mode '110'. A programmable register controls the function of the reference generator. Register 13-1 lists the bit functions of the CVRCON register.

As shown in Figure 13-1, the resistor ladder is segmented to provide two ranges of CVREF values and has a power-down function to conserve power when the reference is not being used. The comparator reference

supply voltage (also referred to as CVRSRC) comes directly from VDD. It should be noted, however, that the voltage at the top of the ladder is CVRSRC – VSAT, where VSAT is the saturation voltage of the power switch transistor. This reference will only be as accurate as the values of CVRSRC and VSAT.

The output of the reference generator may be connected to the RA2/AN2/VREF-/CVREF pin. This can be used as a simple D/A function by the user if a very high-impedance load is used. The primary purpose of this function is to provide a test path for testing the reference generator function.

REGISTER 13-1: CVRCON CONTROL REGISTER (ADDRESS 9Dh)

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
CVREN	CVROE	CVRR	—	CVR3	CVR2	CVR1	CVR0
bit 7				bit 0			

- bit 7 **CVREN:** Comparator Voltage Reference Enable bit
 1 = CVREF circuit powered on
 0 = CVREF circuit powered down
- bit 6 **CVROE:** Comparator VREF Output Enable bit
 1 = CVREF voltage level is output on RA2/AN2/VREF-/CVREF pin
 0 = CVREF voltage level is disconnected from RA2/AN2/VREF-/CVREF pin
- bit 5 **CVRR:** Comparator VREF Range Selection bit
 1 = 0 to 0.75 CVRSRC, with CVRSRC/24 step size
 0 = 0.25 CVRSRC to 0.75 CVRSRC, with CVRSRC/32 step size
- bit 4 **Unimplemented:** Read as '0'
- bit 3-0 **CVR3:CVR0:** Comparator VREF Value Selection bits $0 \leq VR3:VR0 \leq 15$
When CVRR = 1:
 $CVREF = (VR<3:0>/24) \cdot (CVRSRC)$
When CVRR = 0:
 $CVREF = 1/4 \cdot (CVRSRC) + (VR3:VR0/32) \cdot (CVRSRC)$

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

PIC16F87XA

FIGURE 13-1: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM

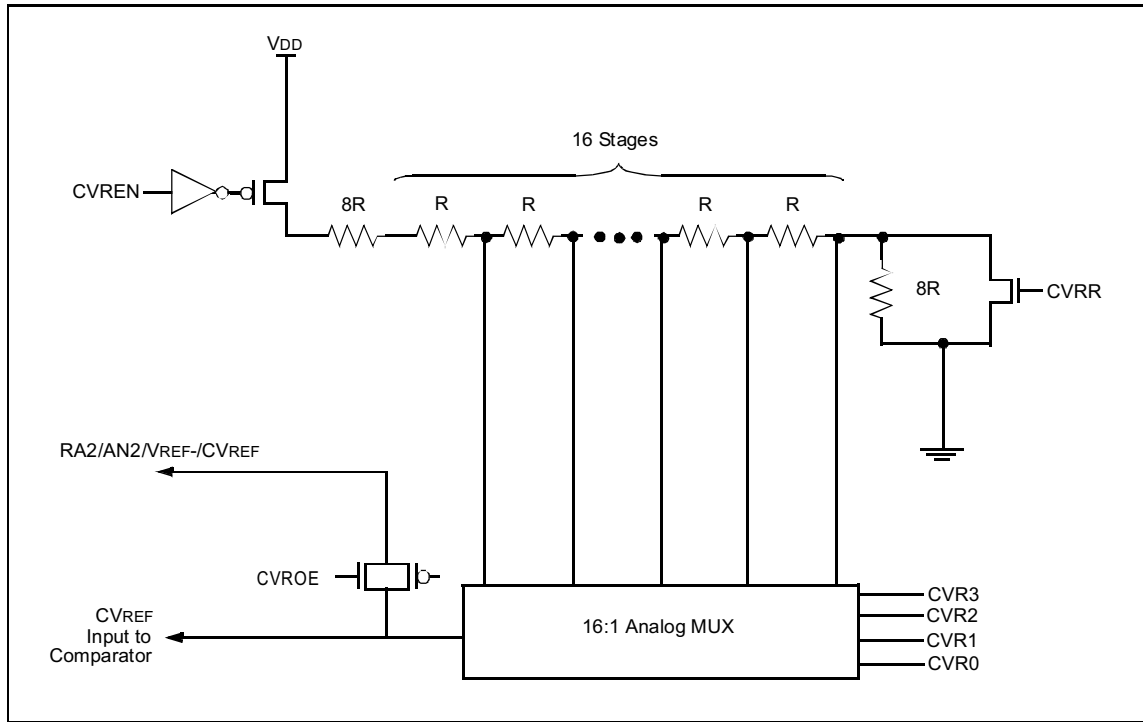


TABLE 13-1: REGISTERS ASSOCIATED WITH COMPARATOR VOLTAGE REFERENCE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other Resets
9Dh	CVRCON	CVREN	CVROE	CVRR	—	CVR3	CVR2	CVR1	CVR0	000- 0000	000- 0000
9Ch	CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111	0000 0111

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'.
 Shaded cells are not used with the comparator voltage reference.

14.0 SPECIAL FEATURES OF THE CPU

All PIC16F87XA devices have a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These are:

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

PIC16F87XA devices have a Watchdog Timer which can be shut-off only through configuration bits. It runs off its own RC oscillator for added reliability.

There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only. It is designed to keep the part in Reset while the power supply stabilizes. With these two timers on-chip, most applications need no external Reset circuitry.

Sleep mode is designed to offer a very low current power-down mode. The user can wake-up from Sleep through external Reset, Watchdog Timer wake-up or through an interrupt.

Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits is used to select various options.

Additional information on special features is available in the PICmicro® Mid-Range MCU Family Reference Manual (DS33023).

14.1 Configuration Bits

The configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1') to select various device configurations. The erased or unprogrammed value of the Configuration Word register is 3FFFh. These bits are mapped in program memory location 2007h.

It is important to note that address 2007h is beyond the user program memory space which can be accessed only during programming.

PIC16F87XA

REGISTER 14-1: CONFIGURATION WORD (ADDRESS 2007h)⁽¹⁾

R/P-1	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1
CP	—	DEBUG	WRT1	WRT0	CPD	LVP	BOREN	—	—	PWRTEN	WDTEN	Fosc1	Fosc0
bit 13											bit0		

- bit 13 **CP:** Flash Program Memory Code Protection bit
 1 = Code protection off
 0 = All program memory code-protected
- bit 12 **Unimplemented:** Read as '1'
- bit 11 **DEBUG:** In-Circuit Debugger Mode bit
 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 10-9 **WRT1:WRT0** Flash Program Memory Write Enable bits
For PIC16F876A/877A:
 11 = Write protection off; all program memory may be written to by EECON control
 10 = 0000h to 00FFh write-protected; 0100h to 1FFFh may be written to by EECON control
 01 = 0000h to 07FFh write-protected; 0800h to 1FFFh may be written to by EECON control
 00 = 0000h to 0FFFh write-protected; 1000h to 1FFFh may be written to by EECON control
For PIC16F873A/874A:
 11 = Write protection off; all program memory may be written to by EECON control
 10 = 0000h to 00FFh write-protected; 0100h to 0FFFh may be written to by EECON control
 01 = 0000h to 03FFh write-protected; 0400h to 0FFFh may be written to by EECON control
 00 = 0000h to 07FFh write-protected; 0800h to 0FFFh may be written to by EECON control
- bit 8 **CPD:** Data EEPROM Memory Code Protection bit
 1 = Data EEPROM code protection off
 0 = Data EEPROM code-protected
- bit 7 **LVP:** Low-Voltage (Single-Supply) In-Circuit Serial Programming Enable bit
 1 = RB3/PGM pin has PGM function; low-voltage programming enabled
 0 = RB3 is digital I/O, HV on MCLR must be used for programming
- bit 6 **BOREN:** Brown-out Reset Enable bit
 1 = BOR enabled
 0 = BOR disabled
- bit 5-4 **Unimplemented:** Read as '1'
- bit 3 **PWRTEN:** Power-up Timer Enable bit
 1 = PWRT disabled
 0 = PWRT enabled
- bit 2 **WDTEN:** Watchdog Timer Enable bit
 1 = WDT enabled
 0 = WDT disabled
- bit 1-0 **Fosc1:Fosc0:** Oscillator Selection bits
 11 = RC oscillator
 10 = HS oscillator
 01 = XT oscillator
 00 = LP oscillator

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
- n = Value when device is unprogrammed		u = Unchanged from programmed state

Note 1: The erased (unprogrammed) value of the Configuration Word is 3FFFh.

14.2 Oscillator Configurations

14.2.1 OSCILLATOR TYPES

The PIC16F87XA 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

14.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In XT, LP or HS modes, a crystal or ceramic resonator is connected to the OSC1/CLKI and OSC2/CLKO pins to establish oscillation (Figure 14-1). The PIC16F87XA oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturer's specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1/CLKI pin (Figure 14-2).

FIGURE 14-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)

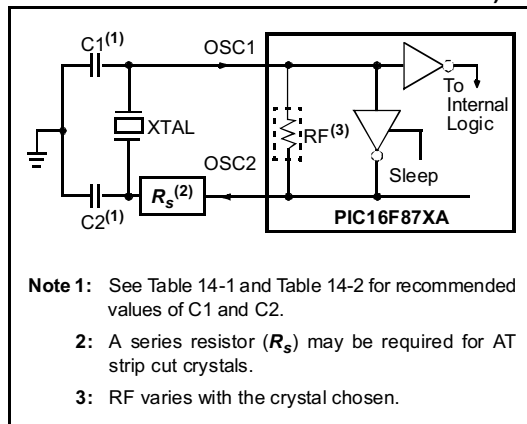


FIGURE 14-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

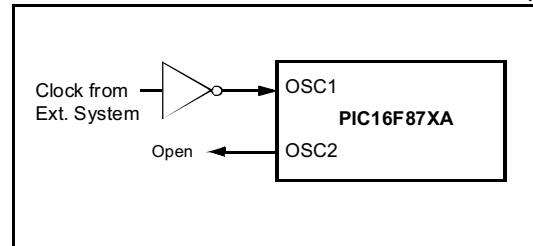


TABLE 14-1: CERAMIC RESONATORS

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 following Table 14-2.			
Resonators Used:			
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.			

PIC16F87XA

TABLE 14-2: 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 following this table.

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: Higher capacitance increases the stability of oscillator but also increases the start-up time.

2: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

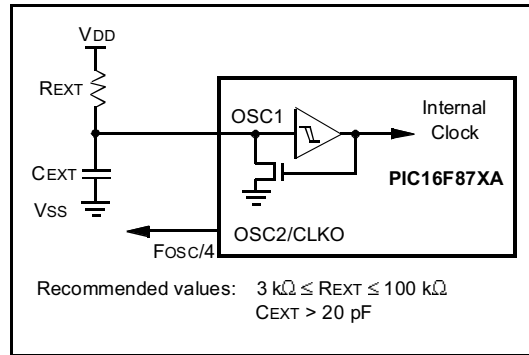
3: R_s may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification.

4: When migrating from other PICmicro® devices, oscillator performance should be verified.

14.2.3 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 (REXT) and capacitor (CEXT) 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 CEXT values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 14-3 shows how the R/C combination is connected to the PIC16F87XA.

FIGURE 14-3: RC OSCILLATOR MODE



PIC16F87XA

14.3 Reset

The PIC16F87XA differentiates between various kinds of Reset:

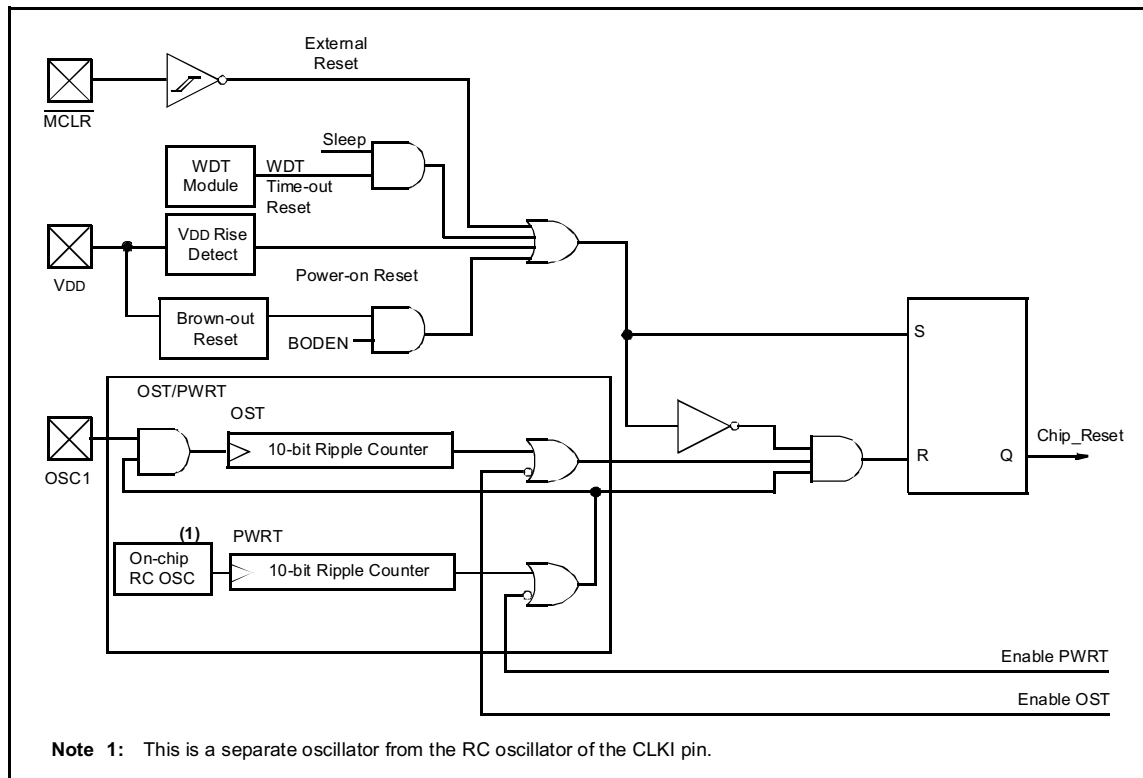
- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ Reset during normal operation
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT Reset (during normal operation)
- WDT Wake-up (during Sleep)
- Brown-out Reset (BOR)

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 14-4. These bits are used in software to determine the nature of the Reset. See Table 14-6 for a full description of Reset states of all registers.

A simplified block diagram of the on-chip Reset circuit is shown in Figure 14-4.

FIGURE 14-4: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



PIC16F87XA

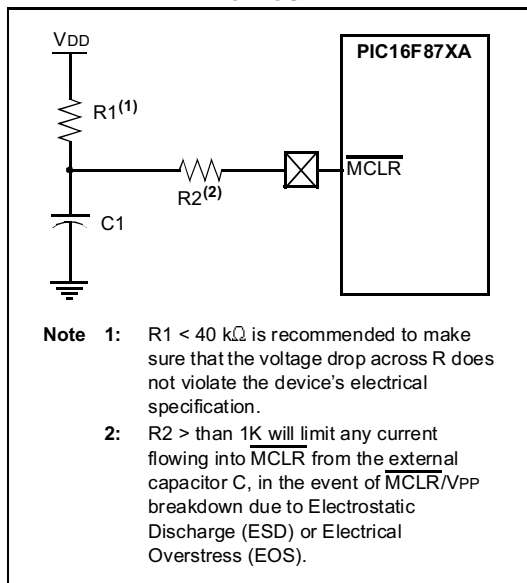
14.4 MCLR

PIC16F87XA devices have a 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.

The behavior of the ESD protection on the $\overline{\text{MCLR}}$ pin differs from previous devices of this family. Voltages applied to the pin that exceed its specification can result in both Resets and current consumption outside of device specification during the Reset event. For this reason, Microchip recommends that the $\overline{\text{MCLR}}$ pin no longer be tied directly to VDD. The use of an RCR network, as shown in Figure 14-5, is suggested.

FIGURE 14-5: RECOMMENDED $\overline{\text{MCLR}}$ CIRCUIT



14.5 Power-on Reset (POR)

A Power-on Reset pulse is generated on-chip when VDD rise is detected (in the range of 1.2V-1.7V). To take advantage of the POR, tie the $\overline{\text{MCLR}}$ pin to VDD through an RC network, as described in **Section 14.4 "MCLR"**. A maximum rise time for VDD is specified. See **Section 17.0 "Electrical Characteristics"** for details.

When the device starts normal operation (exits the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met. Brown-out Reset may be used to meet the start-up conditions. For additional information, refer to application note, AN607, "Power-up Trouble Shooting" (DS00607).

14.6 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 or disable the PWRT.

The power-up time delay will vary from chip to chip due to VDD, temperature and process variation. See **Section 17.0 "Electrical Characteristics"** for details (TPWRT, parameter #33).

14.7 Oscillator Start-up Timer (OST)

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

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

14.8 Brown-out Reset (BOR)

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

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

14.9 Time-out Sequence

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

If $\overline{\text{MCLR}}$ is kept low long enough, the time-outs will expire. Bringing $\overline{\text{MCLR}}$ high will begin execution immediately. This is useful for testing purposes or to synchronize more than one PIC16F87XA device operating in parallel.

Table 14-5 shows the Reset conditions for the Status, PCON and PC registers, while Table 14-6 shows the Reset conditions for all the registers.

PIC16F87XA

14.10 Power Control/Status Register (PCON)

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

Bit 0 is the Brown-out Reset Status bit, $\overline{\text{BOR}}$. The $\overline{\text{BOR}}$ bit is unknown on a Power-on Reset. It must then be set by the user and checked on subsequent Resets to see if it has been cleared, indicating that a BOR has occurred.

When the Brown-out Reset is disabled, the state of the $\overline{\text{BOR}}$ bit is unpredictable and is, therefore, not valid at any time.

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

TABLE 14-3: TIME-OUT IN VARIOUS SITUATIONS

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

TABLE 14-4: STATUS BITS AND THEIR SIGNIFICANCE

$\overline{\text{POR}}$	$\overline{\text{BOR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Condition
0	x	1	1	Power-on Reset
0	x	0	x	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	x	x	0	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
1	0	1	1	Brown-out Reset
1	1	0	1	WDT Reset
1	1	0	0	WDT Wake-up
1	1	u	u	MCLR Reset during normal operation
1	1	1	0	MCLR Reset during Sleep or Interrupt Wake-up from Sleep

Legend: x = don't care, u = unchanged

TABLE 14-5: RESET CONDITIONS FOR SPECIAL REGISTERS

Condition	Program Counter	Status Register	PCON Register
Power-on Reset	000h	0001 1xxx	---- --0x
MCLR Reset during normal operation	000h	000u uuuu	---- --uu
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 ⁽¹⁾	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).

PIC16F87XA

TABLE 14-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS

Register	Devices				Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset	Wake-up via WDT or Interrupt
W	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	73A	74A	76A	77A	N/A	N/A	N/A
TMR0	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	73A	74A	76A	77A	0000 0000	0000 0000	PC + 1 ⁽²⁾
STATUS	73A	74A	76A	77A	0001 1xxx	000q quuu ⁽³⁾	uuuq quuu ⁽³⁾
FSR	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	73A	74A	76A	77A	--0x 0000	--0u 0000	--uu uuuu
PORTB	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTD	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTE	73A	74A	76A	77A	---- -xxx	---- -uuu	---- -uuu
PCLATH	73A	74A	76A	77A	---0 0000	---0 0000	---u uuuu
INTCON	73A	74A	76A	77A	0000 000x	0000 000u	uuuu uuuu ⁽¹⁾
PIR1	73A	74A	76A	77A	r000 0000	r000 0000	ruuu uuuu ⁽¹⁾
	73A	74A	76A	77A	0000 0000	0000 0000	uuuu uuuu ⁽¹⁾
PIR2	73A	74A	76A	77A	-0-0 0--0	-0-0 0--0	-u-u u--u ⁽¹⁾
TMR1L	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	73A	74A	76A	77A	--00 0000	--uu uuuu	--uu uuuu
TMR2	73A	74A	76A	77A	0000 0000	0000 0000	uuuu uuuu
T2CON	73A	74A	76A	77A	-000 0000	-000 0000	-uuu uuuu
SSPBUF	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPCON	73A	74A	76A	77A	0000 0000	0000 0000	uuuu uuuu
CCPR1L	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	73A	74A	76A	77A	--00 0000	--00 0000	--uu uuuu
RCSTA	73A	74A	76A	77A	0000 000x	0000 000x	uuuu uuuu
TXREG	73A	74A	76A	77A	0000 0000	0000 0000	uuuu uuuu
RCREG	73A	74A	76A	77A	0000 0000	0000 0000	uuuu uuuu
CCPR2L	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2H	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	73A	74A	76A	77A	0000 0000	0000 0000	uuuu uuuu
ADRESH	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	73A	74A	76A	77A	0000 00-0	0000 00-0	uuuu uu-u
OPTION_REG	73A	74A	76A	77A	1111 1111	1111 1111	uuuu uuuu
TRISA	73A	74A	76A	77A	--11 1111	--11 1111	--uu uuuu
TRISB	73A	74A	76A	77A	1111 1111	1111 1111	uuuu uuuu
TRISC	73A	74A	76A	77A	1111 1111	1111 1111	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition, r = reserved, maintain clear. Shaded cells indicate conditions do not apply for the designated device.

- 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 14-5 for Reset value for specific condition.

PIC16F87XA

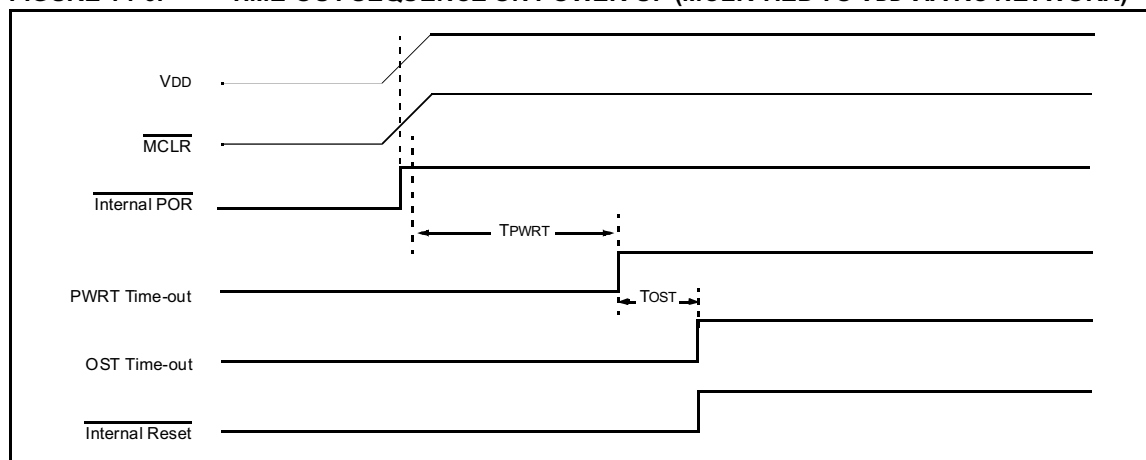
TABLE 14-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Devices				Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset	Wake-up via WDT or Interrupt
	73A	74A	76A	77A			
TRISD	73A	74A	76A	77A	1111 1111	1111 1111	uuuu uuuu
TRISE	73A	74A	76A	77A	0000 -111	0000 -111	uuuu -uuu
PIE1	73A	74A	76A	77A	r000 0000	r000 0000	ruuu uuuu
	73A	74A	76A	77A	0000 0000	0000 0000	uuuu uuuu
PIE2	73A	74A	76A	77A	-0-0 0--0	-0-0 0--0	-u-u u--u
PCON	73A	74A	76A	77A	---- -qg	---- -uu	---- -uu
SSPCON2	73A	74A	76A	77A	0000 0000	0000 0000	uuuu uuuu
PR2	73A	74A	76A	77A	1111 1111	1111 1111	1111 1111
SSPADD	73A	74A	76A	77A	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	73A	74A	76A	77A	--00 0000	--00 0000	--uu uuuu
TXSTA	73A	74A	76A	77A	0000 -010	0000 -010	uuuu -uuu
SPBRG	73A	74A	76A	77A	0000 0000	0000 0000	uuuu uuuu
CMCON	73A	97A	76A	77A	0000 0111	0000 0111	uuuu uuuu
CVRCON	73A	74A	76A	77A	000- 0000	000- 0000	uuu- uuuu
ADRESL	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1	73A	74A	76A	77A	00-- 0000	00-- 0000	uu-- uuuu
EEDATA	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADR	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEDATH	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADRH	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
EECON1	73A	74A	76A	77A	x-- x000	u-- u000	u-- uuuu
EECON2	73A	74A	76A	77A	---- ----	---- ----	---- ----

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition, r = reserved, maintain clear. Shaded cells indicate conditions do not apply for the designated device.

- Note 1:** One or more bits in INTCON, PIR1 and/or PIR2 will be affected (to cause wake-up).
Note 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).
Note 3: See Table 14-5 for Reset value for specific condition.

FIGURE 14-6: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD VIA RC NETWORK)



PIC16F87XA

FIGURE 14-7: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD}): CASE 1

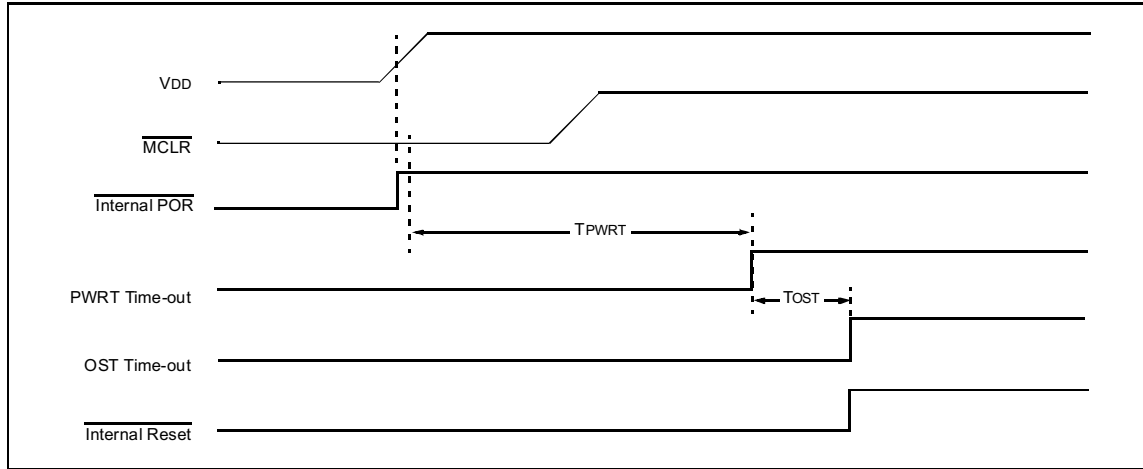


FIGURE 14-8: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD}): CASE 2

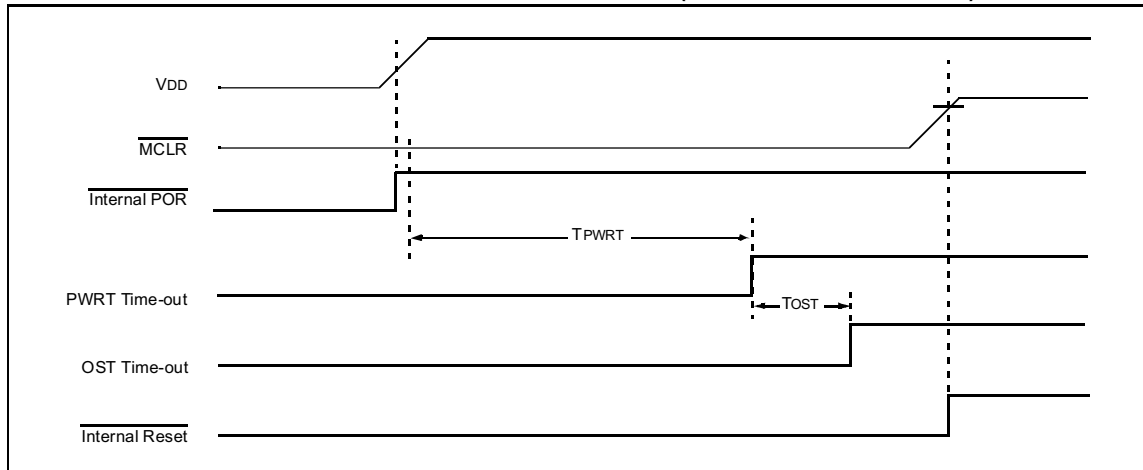
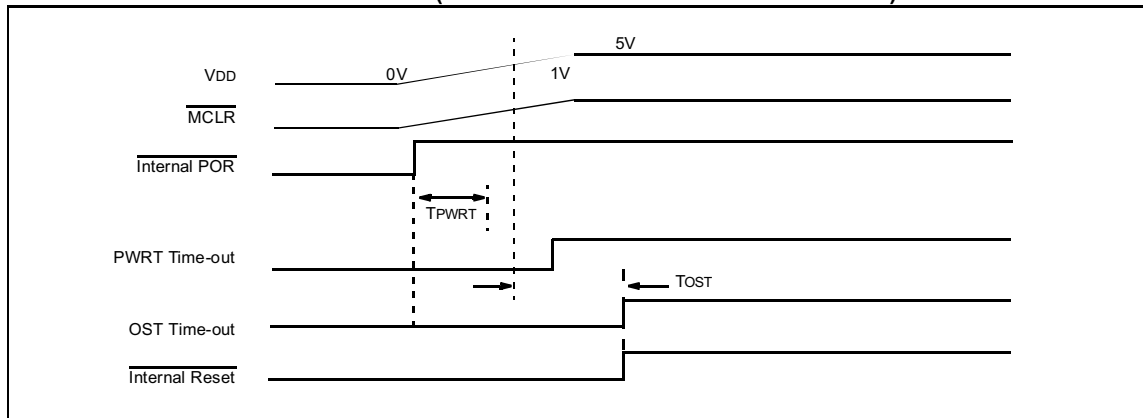


FIGURE 14-9: SLOW RISE TIME ($\overline{\text{MCLR}}$ TIED TO V_{DD} VIA RC NETWORK)



14.11 Interrupts

The PIC16F87XA family has up to 15 sources of interrupt. The Interrupt Control register (INTCON) records individual interrupt requests in flag bits. It also has individual and global interrupt enable bits.

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

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

The "return from interrupt" instruction, RETFIE, exits the interrupt routine, as well as sets the GIE bit, which re-enables interrupts.

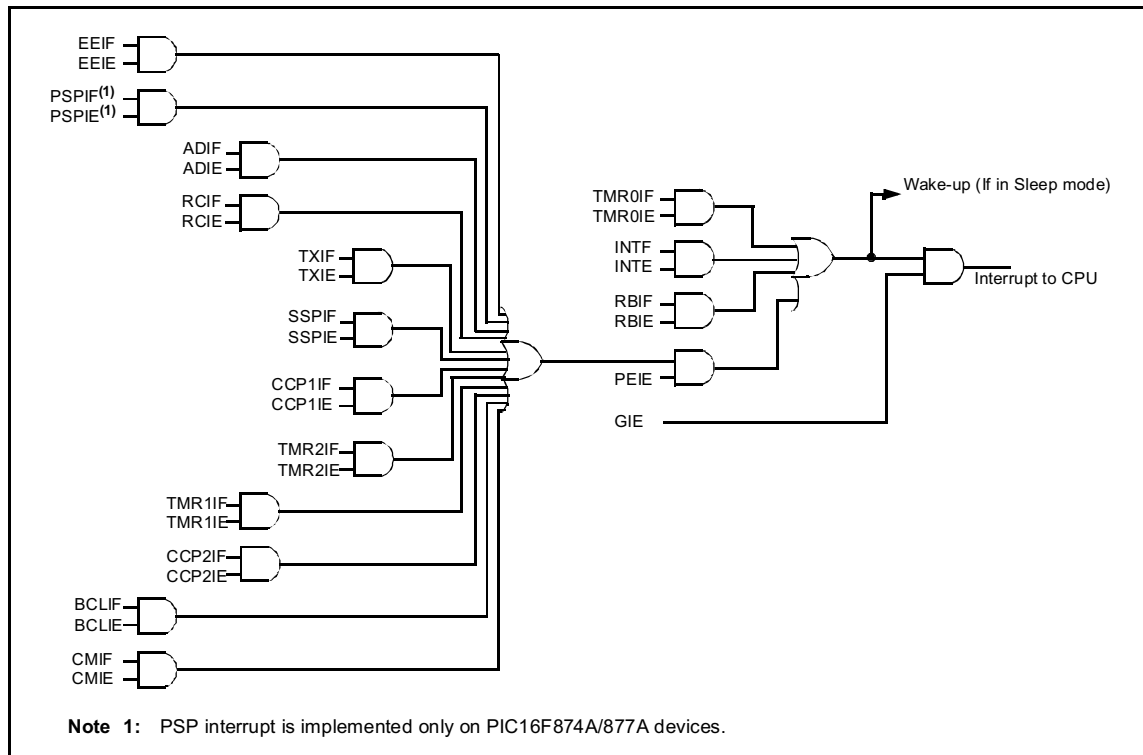
The RB0/INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

The peripheral interrupt flags are contained in the Special Function Registers, PIR1 and PIR2. The corresponding interrupt enable bits are contained in Special Function Registers, PIE1 and PIE2, and the peripheral interrupt enable bit is contained in Special Function Register, INTCON.

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

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

FIGURE 14-10: INTERRUPT LOGIC



PIC16F87XA

14.11.1 INT INTERRUPT

External interrupt on the RB0/INT pin is edge triggered, either rising if bit INTEDG (OPTION_REG<6>) is set or falling if the INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, flag bit, INTF (INTCON<1>), is set. This interrupt can be disabled by clearing enable bit, INTE (INTCON<4>). Flag bit INTF must be cleared in software in the Interrupt Service Routine before re-enabling this interrupt. The INT interrupt can wake-up the processor from Sleep if bit INTE was set prior to going into Sleep. The status of global interrupt enable bit, GIE, decides whether or not the processor branches to the interrupt vector following wake-up. See **Section 14.14 “Power-down Mode (Sleep)”** for details on Sleep mode.

14.11.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 register will set flag bit, TMR0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit, TMR0IE (INTCON<5>). See **Section 5.0 “Timer0 Module”**.

14.11.3 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>). See **Section 4.2 “PORTB and the TRISB Register”**.

14.12 Context Saving During Interrupts

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

For the PIC16F873A/874A devices, the register W_TEMP must be defined in both Banks 0 and 1 and must be defined at the same offset from the bank base address (i.e., If W_TEMP is defined at 0x20 in Bank 0, it must also be defined at 0xA0 in Bank 1). The registers, PCLATH_TEMP and STATUS_TEMP, are only defined in Bank 0.

Since the upper 16 bytes of each bank are common in the PIC16F876A/877A devices, temporary holding registers, W_TEMP, STATUS_TEMP and PCLATH_TEMP, should be placed in here. These 16 locations don't require banking and therefore, make it easier for context save and restore. The same code shown in Example 14-1 can be used.

EXAMPLE 14-1: SAVING STATUS, W AND PCLATH REGISTERS IN RAM

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

14.13 Watchdog Timer (WDT)

The Watchdog Timer is 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/CLKI pin. That means that the WDT will run even if the clock on the OSC1/CLKI and OSC2/CLKO 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 14.1 “Configuration Bits”).

WDT time-out period values may be found in Section 17.0 “Electrical Characteristics” 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 1: 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.

2: 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 14-11: WATCHDOG TIMER BLOCK DIAGRAM

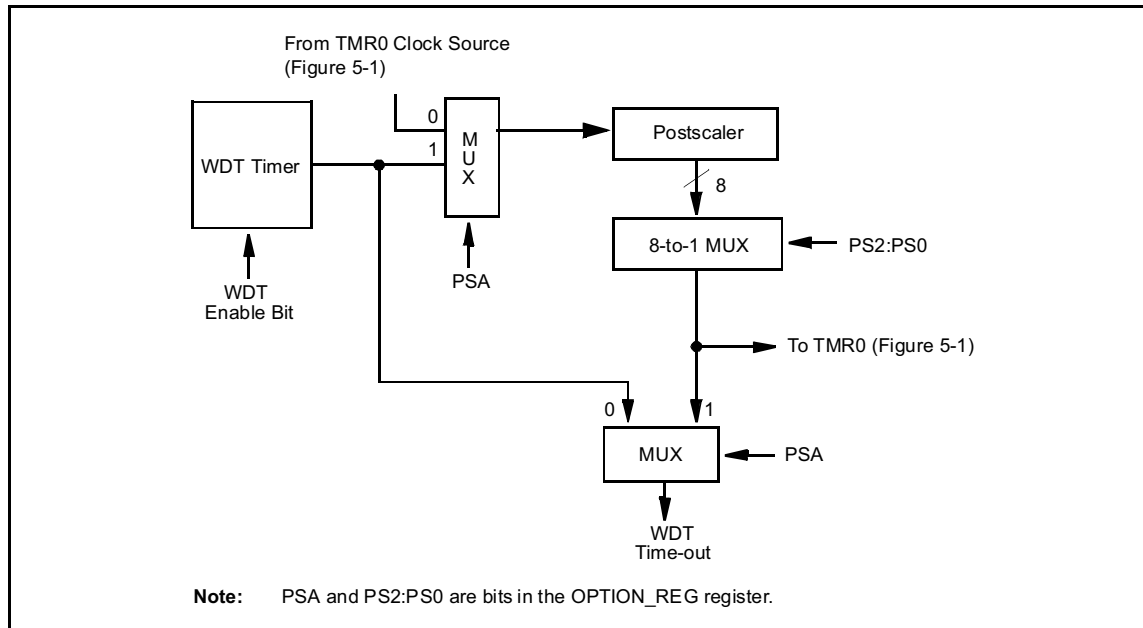


TABLE 14-7: SUMMARY OF WATCHDOG TIMER REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2007h	Config. bits	(1)	BODEN ⁽¹⁾	CP1	CP0	\overline{PWRTE} ⁽¹⁾	WDTE	Fosc1	Fosc0
81h, 181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

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

Note 1: See Register 14-1 for operation of these bits.

PIC16F87XA

14.14 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 `PD` bit (Status<3>) is cleared, the `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 high-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 and disable external clocks. Pull all I/O pins that are high-impedance inputs, high or low externally, to avoid switching currents caused by floating inputs. The `TOCKI` input should also be at `VDD` or `VSS` for lowest current consumption. The contribution from on-chip pull-ups on `PORTB` should also be considered.

The `MCLR` pin must be at a logic high level (V_{IHMC}).

14.14.1 WAKE-UP FROM SLEEP

The device can wake-up from Sleep through one of the following events:

1. External Reset input on `MCLR` pin.
2. Watchdog Timer wake-up (if `WDT` was enabled).
3. Interrupt from `INT` pin, `RB` port change or peripheral interrupt.

External `MCLR` Reset will cause a device Reset. All other events are considered a continuation of program execution and cause a “wake-up”. The `TO` and `PD` bits in the Status register can be used to determine the cause of device Reset. The `PD` bit, which is set on power-up, is cleared when Sleep is invoked. The `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 (PIC16F874/877 only).
2. `TMR1` interrupt. `Timer1` must be operating as an asynchronous counter.
3. `CCP` Capture mode interrupt.
4. Special event trigger (`Timer1` in Asynchronous mode using an external clock).
5. `SSP` (Start/Stop) bit detect interrupt.
6. `SSP` transmit or receive in Slave mode (`SPI/I2C`).
7. `USART` RX or TX (Synchronous Slave mode).
8. A/D conversion (when A/D clock source is `RC`).
9. `EEPROM` write operation completion.
10. Comparator output changes state.

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 prefetched. 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.

14.14.2 WAKE-UP USING INTERRUPTS

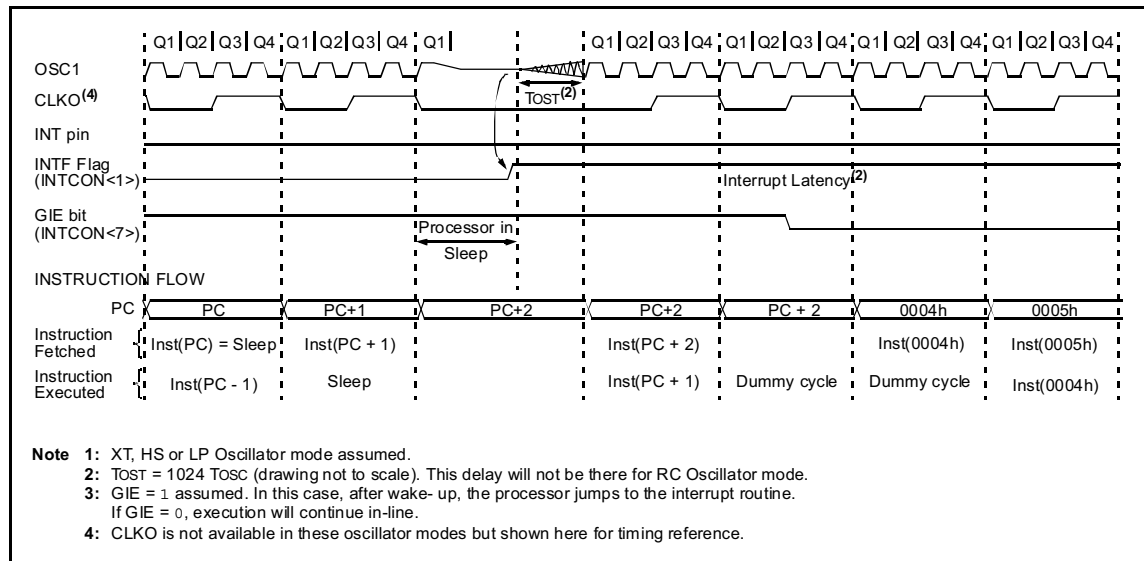
When global interrupts are disabled (`GIE` cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- 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 `TO` bit will not be set and `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 `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 `PD` bit. If the `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 14-12: WAKE-UP FROM SLEEP THROUGH INTERRUPT



14.15 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[®] ICD. When the microcontroller has this feature enabled, some of the resources are not available for general use. Table 14-8 shows which features are consumed by the background debugger.

TABLE 14-8: DEBUGGER RESOURCES

I/O pins	RB6, RB7
Stack	1 level
Program Memory	Address 0000h must be NOP
	Last 100h words
Data Memory	0x070 (0x0F0, 0x170, 0x1F0) 0x1EB-0x1EF

To use the in-circuit debugger function of the microcontroller, the design must implement In-Circuit Serial Programming connections to MCLR/VPP, VDD, GND, RB7 and RB6. This will interface to the in-circuit debugger module available from Microchip or one of the third party development tool companies.

14.16 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.

14.17 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.

PIC16F87XA

14.18 In-Circuit Serial Programming

PIC16F87XA microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

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

For complete details of serial programming, please refer to the "PIC16F87XA Flash Memory Programming Specification" (DS39589).

14.19 Low-Voltage (Single-Supply) ICSP Programming

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

- Note 1:** The High-Voltage Programming mode is always available, regardless of the state of the LVP bit, by applying VIH to the MCLR pin.
- 2:** While in Low-Voltage ICSP mode, the RB3 pin can no longer be used as a general purpose I/O pin.
 - 3:** When using Low-Voltage ICSP Programming (LVP) and the pull-ups on PORTB are enabled, bit 3 in the TRISB register must be cleared to disable the pull-up on RB3 and ensure the proper operation of the device.
 - 4:** RB3 should not be allowed to float if LVP is enabled. An external pull-down device should be used to default the device to normal operating mode. If RB3 floats high, the PIC16F87XA device will enter Programming mode.
 - 5:** LVP mode is enabled by default on all devices shipped from Microchip. It can be disabled by clearing the LVP bit in the CONFIG register.
 - 6:** Disabling LVP will provide maximum compatibility to other PIC16CXXX devices.

If Low-Voltage Programming mode is not used, the LVP bit can be programmed to a '0' and RB3/PGM becomes a digital I/O pin. However, the LVP bit may only be programmed when programming is entered with VIH on MCLR. The LVP bit can only be changed when using high voltage on MCLR.

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

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

15.0 INSTRUCTION SET SUMMARY

The PIC16 instruction set is highly orthogonal and is comprised of three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each PIC16 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 formats for each of the categories is presented in Figure 15-1, while the various opcode fields are summarized in Table 15-1.

Table 15-2 lists the instructions recognized by the MPASM™ Assembler. A complete description of each instruction is also available in the PICmicro® Mid-Range MCU Family Reference Manual (DS33023).

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 bit affected by the operation, while 'f' represents the address 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

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a normal instruction execution time of 1 μs. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles with the second cycle executed as a NOP.

Note: To maintain upward compatibility with future PIC16F87XA products, do not use the **OPTION** and **TRIS** instructions.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

15.1 READ-MODIFY-WRITE OPERATIONS

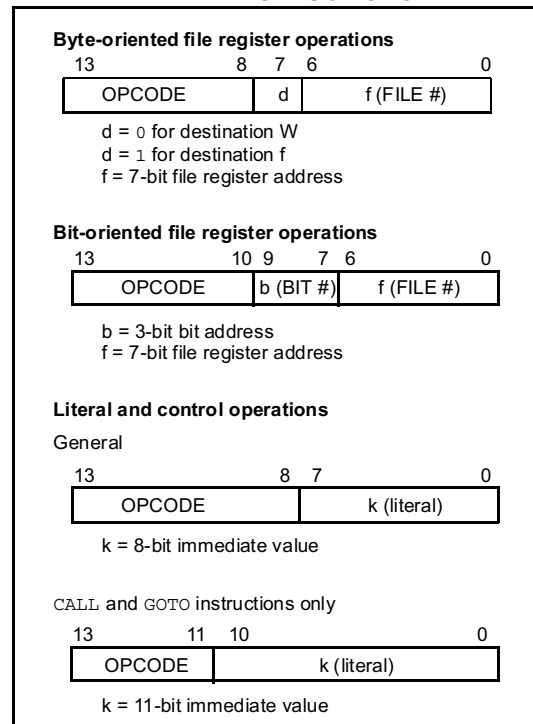
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a "CLRF PORTB" instruction will read PORTB, clear all the data bits, then write the result back to PORTB. This example would have the unintended result that the condition that sets the RBIF flag would be cleared.

TABLE 15-1: 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

FIGURE 15-1: GENERAL FORMAT FOR INSTRUCTIONS



PIC16F87XA

TABLE 15-2: PIC16F87XA INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	14-Bit Opcode		Status Affected	Notes
			MSb	LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	f, d Add W and f	1	00	0111	dfff ffff	C,DC,Z 1,2
ANDWF	f, d AND W with f	1	00	0101	dfff ffff	Z 1,2
CLRF	f Clear f	1	00	0001	1fff ffff	Z 2
CLRWF	- Clear W	1	00	0001	0xxx xxxx	Z
COMF	f, d Complement f	1	00	1001	dfff ffff	Z 1,2
DECf	f, d Decrement f	1	00	0011	dfff ffff	Z 1,2
DECFSZ	f, d Decrement f, Skip if 0	1(2)	00	1011	dfff ffff	1,2,3
INCF	f, d Increment f	1	00	1010	dfff ffff	Z 1,2
INCFSZ	f, d Increment f, Skip if 0	1(2)	00	1111	dfff ffff	1,2,3
IORWF	f, d Inclusive OR W with f	1	00	0100	dfff ffff	Z 1,2
MOVF	f, d Move f	1	00	1000	dfff ffff	Z 1,2
MOVWF	f Move W to f	1	00	0000	1fff ffff	
NOP	- No Operation	1	00	0000	0xx0 0000	
RLF	f, d Rotate Left f through Carry	1	00	1101	dfff ffff	C 1,2
RRF	f, d Rotate Right f through Carry	1	00	1100	dfff ffff	C 1,2
SUBWF	f, d Subtract W from f	1	00	0010	dfff ffff	C,DC,Z 1,2
SWAPF	f, d Swap nibbles in f	1	00	1110	dfff ffff	1,2
XORWF	f, d Exclusive OR W with f	1	00	0110	dfff ffff	Z 1,2
BIT-ORIENTED FILE REGISTER OPERATIONS						
BCF	f, b Bit Clear f	1	01	00bb	bfff ffff	1,2
BSF	f, b Bit Set f	1	01	01bb	bfff ffff	1,2
BTFSC	f, b Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff ffff	3
BTFSS	f, b Bit Test f, Skip if Set	1 (2)	01	11bb	bfff ffff	3
LITERAL AND CONTROL OPERATIONS						
ADDLW	k Add Literal and W	1	11	111x	kkkk kkkk	C,DC,Z Z
ANDLW	k AND Literal with W	1	11	1001	kkkk kkkk	
CALL	k Call Subroutine	2	10	0kkk	kkkk kkkk	
CLRWD _T	- Clear Watchdog Timer	1	00	0000	0110 0100	$\overline{TO,PD}$
GOTO	k Go to Address	2	10	1kkk	kkkk kkkk	
IORLW	k Inclusive OR Literal with W	1	11	1000	kkkk kkkk	Z
MOVLW	k Move Literal to W	1	11	00xx	kkkk kkkk	
RETFIE	- Return from Interrupt	2	00	0000	0000 1001	
RETLW	k Return with Literal in W	2	11	01xx	kkkk kkkk	
RETURN	- Return from Subroutine	2	00	0000	0000 1000	
SLEEP	- Go into Standby mode	1	00	0000	0110 0011	$\overline{TO,PD}$
SUBLW	k Subtract W from Literal	1	11	110x	kkkk kkkk	C,DC,Z
XORLW	k Exclusive OR Literal with W	1	11	1010	kkkk kkkk	Z

- Note 1:** When an I/O register is modified as a function of itself (e.g., MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.
- 3:** If Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

Note: Additional information on the mid-range instruction set is available in the PICmicro® Mid-Range MCU Family Reference Manual (DS33023).

15.2 Instruction Descriptions

ADDLW	Add Literal and W
Syntax:	[<i>label</i>] ADDLW <i>k</i>
Operands:	$0 \leq k \leq 255$
Operation:	$(W) + k \rightarrow (W)$
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register.

BCF	Bit Clear f
Syntax:	[<i>label</i>] BCF <i>f</i> , <i>b</i>
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	$0 \rightarrow (f)$
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

ADDWF	Add W and f
Syntax:	[<i>label</i>] ADDWF <i>f</i> , <i>d</i>
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(W) + (f) \rightarrow (\text{destination})$
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

BSF	Bit Set f
Syntax:	[<i>label</i>] BSF <i>f</i> , <i>b</i>
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	$1 \rightarrow (f)$
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

ANDLW	AND Literal with W
Syntax:	[<i>label</i>] ANDLW <i>k</i>
Operands:	$0 \leq k \leq 255$
Operation:	$(W) \text{ .AND. } (k) \rightarrow (W)$
Status Affected:	Z
Description:	The contents of W register are AND'ed with the eight-bit literal 'k'. The result is placed in the W register.

BTFSS	Bit Test f, Skip if Set
Syntax:	[<i>label</i>] BTFSS <i>f</i> , <i>b</i>
Operands:	$0 \leq f \leq 127$ $0 \leq b < 7$
Operation:	skip if $(f) = 1$
Status Affected:	None
Description:	If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2 T _{CY} instruction.

ANDWF	AND W with f
Syntax:	[<i>label</i>] ANDWF <i>f</i> , <i>d</i>
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(W) \text{ .AND. } (f) \rightarrow (\text{destination})$
Status Affected:	Z
Description:	AND the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

BTFSC	Bit Test, Skip if Clear
Syntax:	[<i>label</i>] BTFSC <i>f</i> , <i>b</i>
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	skip if $(f) = 0$
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b' in register 'f' is '0', the next instruction is discarded and a NOP is executed instead, making this a 2 T _{CY} instruction.

PIC16F87XA

CALL	Call Subroutine
Syntax:	[<i>label</i>] CALL k
Operands:	$0 \leq k \leq 2047$
Operation:	(PC)+ 1 → TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.

CLRWDT	Clear Watchdog Timer
Syntax:	[<i>label</i>] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → \overline{TO} 1 → \overline{PD}
Status Affected:	\overline{TO} , \overline{PD}
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits, \overline{TO} and \overline{PD} , are set.

CLRF	Clear f
Syntax:	[<i>label</i>] CLRF f
Operands:	$0 \leq f \leq 127$
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

COMF	Complement f
Syntax:	[<i>label</i>] COMF f,d
Operands:	$0 \leq f \leq 127$ d ∈ [0,1]
Operation:	$\overline{(f)}$ → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

CLRW	Clear W
Syntax:	[<i>label</i>] CLRW
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

DECF	Decrement f
Syntax:	[<i>label</i>] DECF f,d
Operands:	$0 \leq f \leq 127$ d ∈ [0,1]
Operation:	(f) - 1 → (destination)
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

PIC16F87XA

DECFSZ	Decrement f, Skip if 0
Syntax:	[<i>label</i>] DECFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) - 1 \rightarrow (\text{destination})$; skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2 TCY instruction.

INCFSZ	Increment f, Skip if 0
Syntax:	[<i>label</i>] INCFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) + 1 \rightarrow (\text{destination})$, skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2 TCY instruction.

GOTO	Unconditional Branch
Syntax:	[<i>label</i>] GOTO k
Operands:	$0 \leq k \leq 2047$
Operation:	$k \rightarrow PC<10:0>$ $PCLATH<4:3> \rightarrow PC<12:11>$
Status Affected:	None
Description:	GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

IORLW	Inclusive OR Literal with W
Syntax:	[<i>label</i>] IORLW k
Operands:	$0 \leq k \leq 255$
Operation:	$(W) .OR. k \rightarrow (W)$
Status Affected:	Z
Description:	The contents of the W register are OR'ed with the eight-bit literal 'k'. The result is placed in the W register.

INCF	Increment f
Syntax:	[<i>label</i>] INCF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(f) + 1 \rightarrow (\text{destination})$
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

IORWF	Inclusive OR W with f
Syntax:	[<i>label</i>] IORWF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	$(W) .OR. (f) \rightarrow (\text{destination})$
Status Affected:	Z
Description:	Inclusive OR the W register with register 'f'. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.