



PIC18F87J11 Family Data Sheet

64/80-Pin High-Performance,
1-Mbit Flash Microcontrollers
with nanoWatt Technology

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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

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

AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, PS logo, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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.

© 2007, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona, Gresham, Oregon and Mountain View, California. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMS, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

== ISO/TS 16949:2002 ==



MICROCHIP

PIC18F87J11 FAMILY

64/80-Pin High-Performance, 1-Mbit Flash Microcontrollers with nanoWatt Technology

Flexible Oscillator Structure:

- Four Crystal modes, including High-Precision PLL
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - Provides 8 user-selectable frequencies from 31 kHz to 8 MHz
 - Provides a complete range of clock speeds, from 31 kHz to 32 MHz when used with PLL
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25mA on PORTB and PORTC
- Four Programmable External Interrupts
- Four Input Change Interrupts
- One 8/16-Bit Timer/Counter
- Two 8-Bit Timers/Counters
- Two 16-Bit Timers/Counters
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Two Master Synchronous Serial Port (MSSP) modules supporting 3-Wire SPI (all 4 modes) and I²C™ Master and Slave modes
- Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect

Peripheral Highlights (continued):

- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port (PMP/EPSP) with 16 Address Lines
- Dual Analog Comparators with Input Multiplexing
- 10-Bit, up to 15-Channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep

External Memory Bus

(80-pin devices only):

- Address Capability of up to 2 Mbytes
- 8-Bit or 16-Bit Interface
- 12-Bit, 16-Bit and 20-Bit Addressing modes

Special Microcontroller Features:

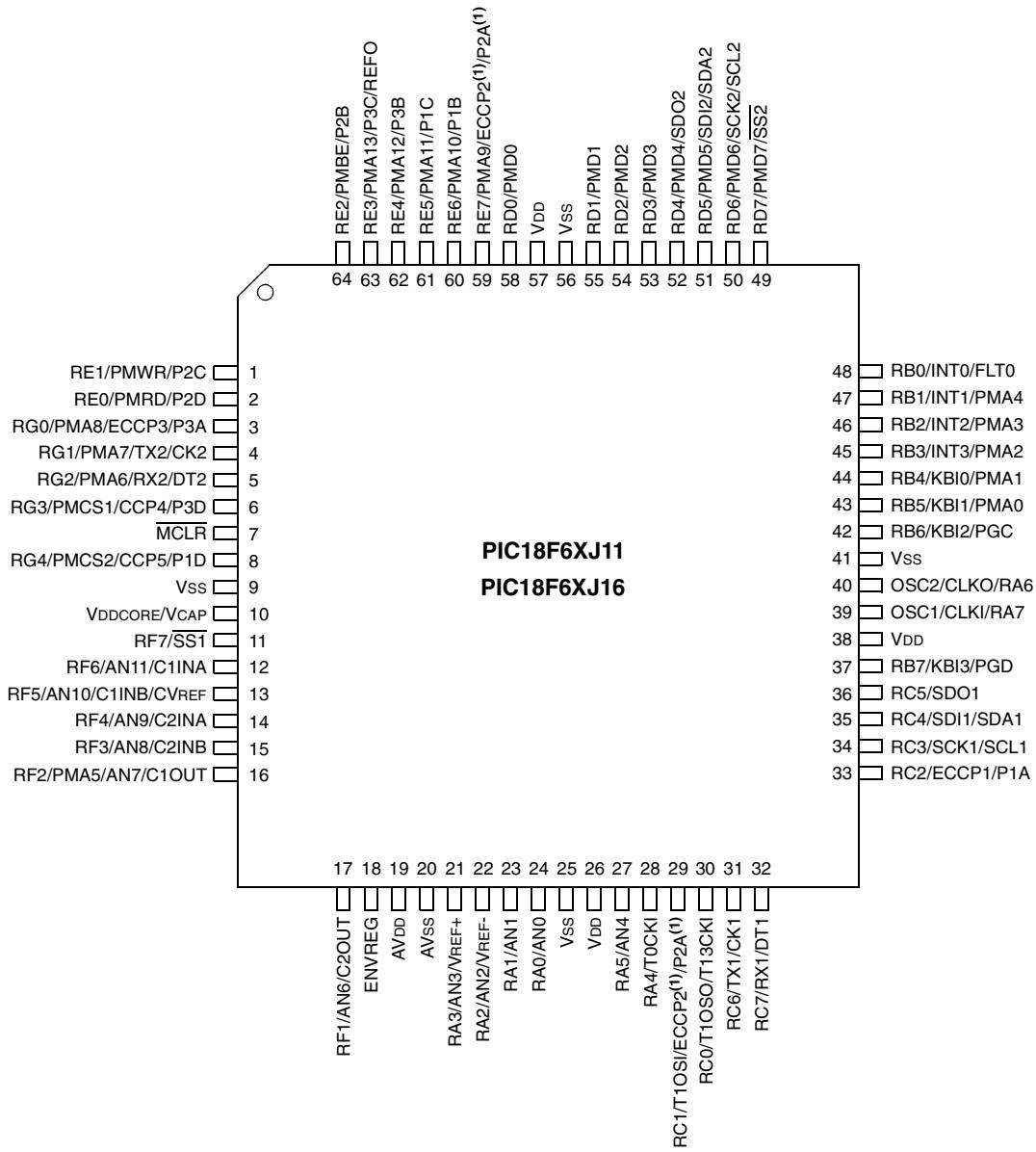
- Low-Power, High-Speed CMOS Flash Technology
- C Compiler Optimized Architecture for Re-Entrant Code
- Power Management Features:
 - Run: CPU on, peripherals on
 - Idle: CPU off, peripherals on
 - Sleep: CPU off, peripherals off
- Priority Levels for Interrupts
- Self-Programmable under Software Control
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
 - Programmable period from 4 ms to 131s
- Single-Supply In-Circuit Serial Programming™ (ICSP™) via Two Pins
- In-Circuit Debug (ICD) with 3 Breakpoints via Two Pins
- Operating Voltage Range of 2.0V to 3.6V
- 5.5V Tolerant Inputs (digital only pins)
- On-Chip 2.5V Regulator
- Flash Program Memory of 10000 Erase/Write Cycles and 20-Year Data Retention

Device	Flash Program Memory (bytes)	SRAM Data Memory (bytes)	I/O	10-Bit A/D (ch)	CCP/ECCP (PWM)	MSSP		EUSART	Comparators	Timers 8/16-Bit	External Bus	PMP/EPSP	
						SPI	Master I ² C™						
PIC18F66J11	64 kB	3930	52	11	2/3	2	Y	Y	2	2	2/3	N	Y
PIC18F66J16	96 kB	3930	52	11	2/3	2	Y	Y	2	2	2/3	N	Y
PIC18F67J11	128 kB	3930	52	11	2/3	2	Y	Y	2	2	2/3	N	Y
PIC18F86J11	64 kB	3930	68	15	2/3	2	Y	Y	2	2	2/3	Y	Y
PIC18F86J16	96 kB	3930	68	15	2/3	2	Y	Y	2	2	2/3	Y	Y
PIC18F87J11	128 kB	3930	68	15	2/3	2	Y	Y	2	2	2/3	Y	Y

PIC18F87J11 FAMILY

Pin Diagrams

64-Pin TQFP

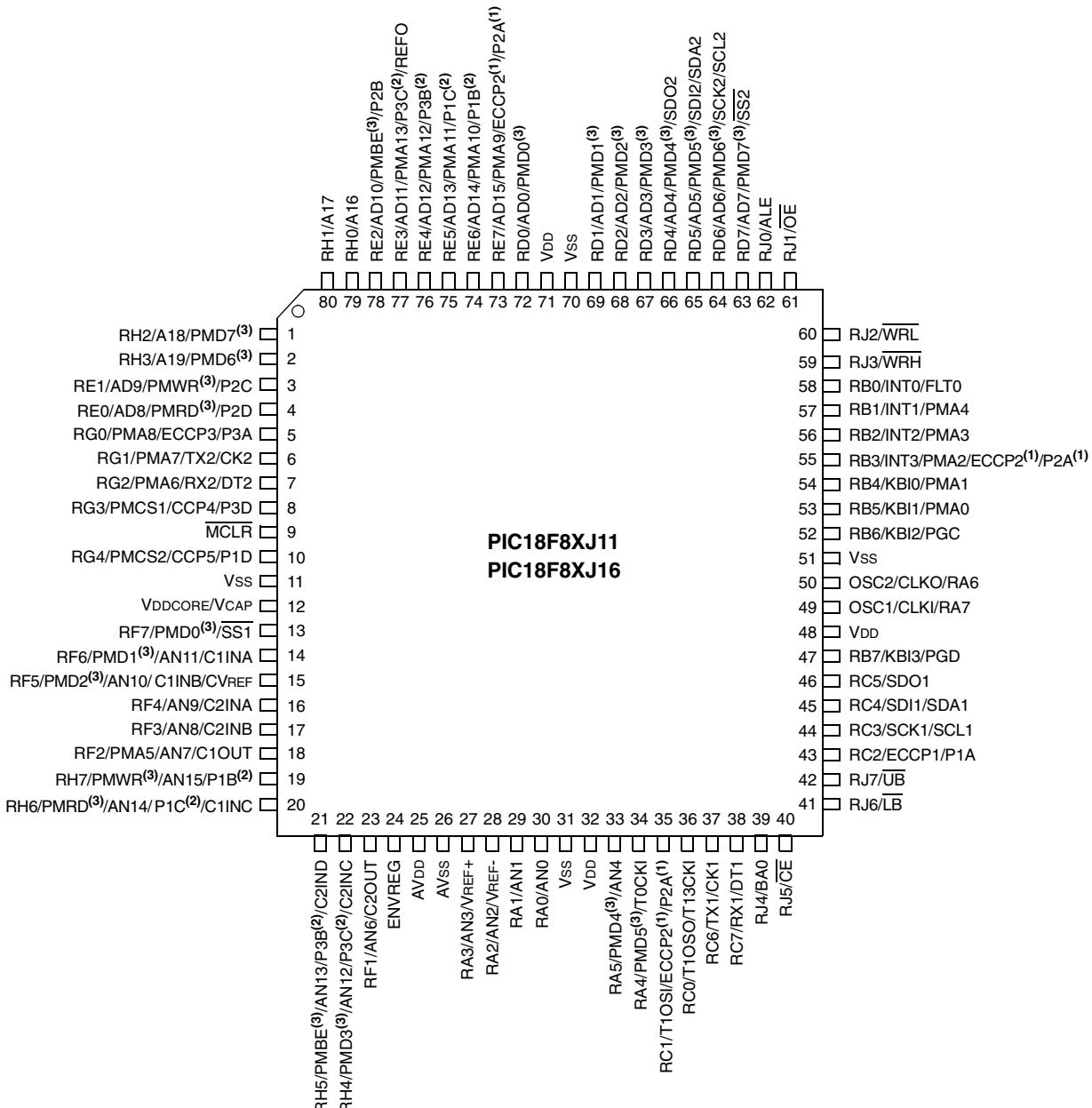


Note 1: The ECCP2/P2A pin placement depends on the CCP2MX Configuration bit setting.

PIC18F87J11 FAMILY

Pin Diagrams (Continued)

80-Pin TQFP



Note 1: The ECCP2/P2A pin placement depends on the CCP2MX Configuration bit and Processor mode settings.

2: P1B, P1C, P3B, and P3C pin placement depends on the ECCPMX Configuration bit setting.

3: PMP pin placement depends on the PMPMX Configuration bit setting.

PIC18F87J11 FAMILY

Table of Contents

1.0	Device Overview	7
2.0	Oscillator Configurations	31
3.0	Power-Managed Modes	41
4.0	Reset	49
5.0	Memory Organization	61
6.0	Flash Program Memory	87
7.0	External Memory Bus	97
8.0	8 x 8 Hardware Multiplier	109
9.0	Interrupts	111
10.0	I/O Ports	127
11.0	Parallel Master Port	157
12.0	Timer0 Module	181
13.0	Timer1 Module	185
14.0	Timer2 Module	191
15.0	Timer3 Module	193
16.0	Timer4 Module	197
17.0	Capture/Compare/PWM (CCP) Modules	199
18.0	Enhanced Capture/Compare/PWM (ECCP) Module	207
19.0	Master Synchronous Serial Port (MSSP) Module	223
20.0	Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART)	269
21.0	10-bit Analog-to-Digital Converter (A/D) Module	291
22.0	Comparator Module	301
23.0	Comparator Voltage Reference Module	309
24.0	Special Features of the CPU	313
25.0	Instruction Set Summary	329
26.0	Development Support	379
27.0	Electrical Characteristics	383
28.0	DC and AC Characteristics Graphs and Tables	423
29.0	Packaging Information	425
	Appendix A: Revision History	429
	Appendix B: Device Differences	429
	The Microchip Web Site	431
	Customer Change Notification Service	431
	Customer Support	431
	Reader Response	432
	Index	433
	Product Identification System	445

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@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)

When contacting a sales office, 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 to receive the most current information on all of our products.

PIC18F87J11 FAMILY

NOTES:

1.0 DEVICE OVERVIEW

This document contains device-specific information for the following devices:

- PIC18F66J11
- PIC18F66J16
- PIC18F67J11
- PIC18F86J11
- PIC18F86J16
- PIC18F87J11

This family introduces a line of low-voltage, general purpose microcontrollers with the main traditional advantage of all PIC18 microcontrollers – namely, high computational performance and a rich feature set – at an extremely competitive price point. These features make the PIC18F87J11 Family a logical choice for many high-performance applications, where an extended peripheral feature set is required, and cost is a primary consideration.

1.1 Core Features

1.1.1 nanoWatt TECHNOLOGY

All of the devices in the PIC18F87J11 family incorporate a range of features that can significantly reduce power consumption during operation. Key items include:

- **Alternate Run Modes:** By clocking the controller from the Timer1 source or the internal RC oscillator, power consumption during code execution can be reduced by as much as 90%.
- **Multiple Idle Modes:** The controller can also run with its CPU core disabled but the peripherals still active. In these states, power consumption can be reduced even further, to as little as 4% of normal operation requirements.
- **On-the-Fly Mode Switching:** The power-managed modes are invoked by user code during operation, allowing the user to incorporate power-saving ideas into their application's software design.

1.1.2 OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC18F87J11 Family offer four different oscillator options, allowing users a range of choices in developing application hardware. These include:

- Two Crystal modes, using crystals or ceramic resonators.
- Two External Clock modes, offering the option of a divide-by-4 clock output.
- An internal oscillator block which provides an 8 MHz clock and an INTRC source (approximately 31 kHz, stable over temperature and VDD), as well as a range of 6 user-selectable clock frequencies, between 125 kHz to 4 MHz, for a total of 8 clock frequencies. This option frees an oscillator pin for use as an additional general purpose I/O.

- A Phase Lock Loop (PLL) frequency multiplier, available to all of the oscillator modes, which allows a wide range of clock speeds from 16 MHz to 48 MHz

The internal oscillator block provides a stable reference source that gives the family additional features for robust operation:

- **Fail-Safe Clock Monitor:** This option constantly monitors the main clock source against a reference signal provided by the internal oscillator. If a clock failure occurs, the controller is switched to the internal oscillator, allowing for continued low-speed operation or a safe application shutdown.
- **Two-Speed Start-up:** This option allows the internal oscillator to serve as the clock source from Power-on Reset, or wake-up from Sleep mode, until the primary clock source is available.

1.1.3 EXPANDED MEMORY

The PIC18F87J11 family provides ample room for application code, from 64 Kbytes to 128 Kbytes of code space. The Flash cells for program memory are rated to last up to 10,000 erase/write cycles. Data retention without refresh is conservatively estimated to be greater than 20 years.

The Flash program memory is readable, writable, and during normal operation, the PIC18F87J11 Family also provides plenty of room for dynamic application data with up to 3930 bytes of data RAM.

1.1.4 EXTERNAL MEMORY BUS

In the event that 128 Kbytes of memory are inadequate for an application, the 80-pin members of the PIC18F87J11 Family also implement an External Memory Bus (EMB). This allows the controller's internal program counter to address a memory space of up to 2 Mbytes, permitting a level of data access that few 8-bit devices can claim. This allows additional memory options, including:

- Using combinations of on-chip and external memory up to the 2-Mbyte limit
- Using external Flash memory for reprogrammable application code or large data tables
- Using external RAM devices for storing large amounts of variable data

1.1.5 EXTENDED INSTRUCTION SET

The PIC18F87J11 Family implements the optional extension to the PIC18 instruction set, adding 8 new instructions and an Indexed Addressing mode. Enabled as a device configuration option, the extension has been specifically designed to optimize re-entrant application code originally developed in high-level languages, such as 'C'.

PIC18F87J11 FAMILY

1.1.6 EASY MIGRATION

Regardless of the memory size, all devices share the same rich set of peripherals, allowing for a smooth migration path as applications grow and evolve.

The consistent pinout scheme used throughout the entire family also aids in migrating to the next larger device. This is true when moving between the 64-pin members, between the 80-pin members, or even jumping from 64-pin to 80-pin devices.

The PIC18F87J11 Family is also pin compatible with other PIC18 families, such as the PIC18F87J10, PIC18F85J11, PIC18F8720 and PIC18F8722. This allows a new dimension to the evolution of applications, allowing developers to select different price points within Microchip's PIC18 portfolio, while maintaining the same feature set.

1.2 Other Special Features

- **Communications:** The PIC18F87J11 Family incorporates a range of serial and parallel communication peripherals. These devices all include 2 independent Enhanced USARTs and 2 Master SSP modules, capable of both SPI and I²C™ (Master and Slave) modes of operation. The devices also have a parallel port and can be configured to function as either a Parallel Master Port or as a Parallel Slave Port.
- **CCP Modules:** All devices in the family incorporate two Capture/Compare/PWM (CCP) modules and three Enhanced CCP (ECCP) modules to maximize flexibility in control applications. Up to four different time bases may be used to perform several different operations at once. Each of the three ECCP modules offers up to four PWM outputs, allowing for a total of 12 PWMs. The ECCPs also offer many beneficial features, including polarity selection, programmable dead time, auto-shutdown and restart, and Half-Bridge and Full-Bridge Output modes.
- **10-Bit A/D Converter:** This module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated without waiting for a sampling period, and thus, reducing code overhead.
- **Extended Watchdog Timer (WDT):** This enhanced version incorporates a 16-bit prescaler, allowing an extended time-out range that is stable across operating voltage and temperature. See **Section 27.0 “Electrical Characteristics”** for time-out periods.

1.3 Details on Individual Family Members

Devices in the PIC18F87J11 Family are available in 64-pin and 80-pin packages. Block diagrams for the two groups are shown in Figure 1-1 and Figure 1-2. The devices are differentiated from each other in three ways:

1. Flash program memory (three sizes, ranging from 64 Kbytes for PIC18FX6J11 devices to 128 Kbytes for PIC18FX7J11 devices).
2. I/O ports (7 bidirectional ports on 64-pin devices, 9 bidirectional ports on 80-pin devices).
3. A/D input channels (11 on 64-pin devices, 15 on 80-pin devices).

All other features for devices in this family are identical. These are summarized in Table 1-1 and Table 1-2.

The pinouts for all devices are listed in Table 1-3 and Table 1-4.

PIC18F87J11 FAMILY

TABLE 1-1: DEVICE FEATURES FOR THE PIC18F6XJ1X (64-PIN DEVICES)

Features	PIC18F66J11	PIC18F66J16	PIC18F67J11
Operating Frequency	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz
Program Memory (Bytes)	64K	96K	128K
Program Memory (Instructions)	32768	49152	65536
Data Memory (Bytes)	3930	3930	3930
Interrupt Sources	29		
I/O Ports		Ports A, B, C, D, E, F, G	
Timers		5	
Capture/Compare/PWM Modules		2	
Enhanced Capture/Compare/PWM Modules		3	
Serial Communications		MSSP (2), Enhanced USART (2)	
Parallel Communications (PMP)		Yes	
10-Bit Analog-to-Digital Module		11 Input Channels	
Resets (and Delays)		POR, BOR, RESET Instruction, Stack Full, Stack Underflow, MCLR, WDT (PWRT, OST)	
Instruction Set		75 Instructions, 83 with Extended Instruction Set Enabled	
Packages		64-Pin TQFP	

TABLE 1-2: DEVICE FEATURES FOR THE PIC18F8XJ1X (80-PIN DEVICES)

Features	PIC18F86J11	PIC18F86J16	PIC18F87J11
Operating Frequency	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz
Program Memory (Bytes)	64K	96K	128K
Program Memory (Instructions)	32768	49152	65536
Data Memory (Bytes)	3930	3930	3930
Interrupt Sources	29		
I/O Ports		Ports A, B, C, D, E, F, G, H, J	
Timers		5	
Capture/Compare/PWM Modules		2	
Enhanced Capture/Compare/PWM Modules		3	
Serial Communications		MSSP (2), Enhanced USART (2)	
Parallel Communications (PMP)		Yes	
10-Bit Analog-to-Digital Module		15 Input Channels	
Resets (and Delays)		POR, BOR, RESET Instruction, Stack Full, Stack Underflow, MCLR, WDT (PWRT, OST)	
Instruction Set		75 Instructions, 83 with Extended Instruction Set Enabled	
Packages		80-Pin TQFP	

PIC18F87J11 FAMILY

FIGURE 1-1: PIC18F6XJ1X (64-PIN) BLOCK DIAGRAM

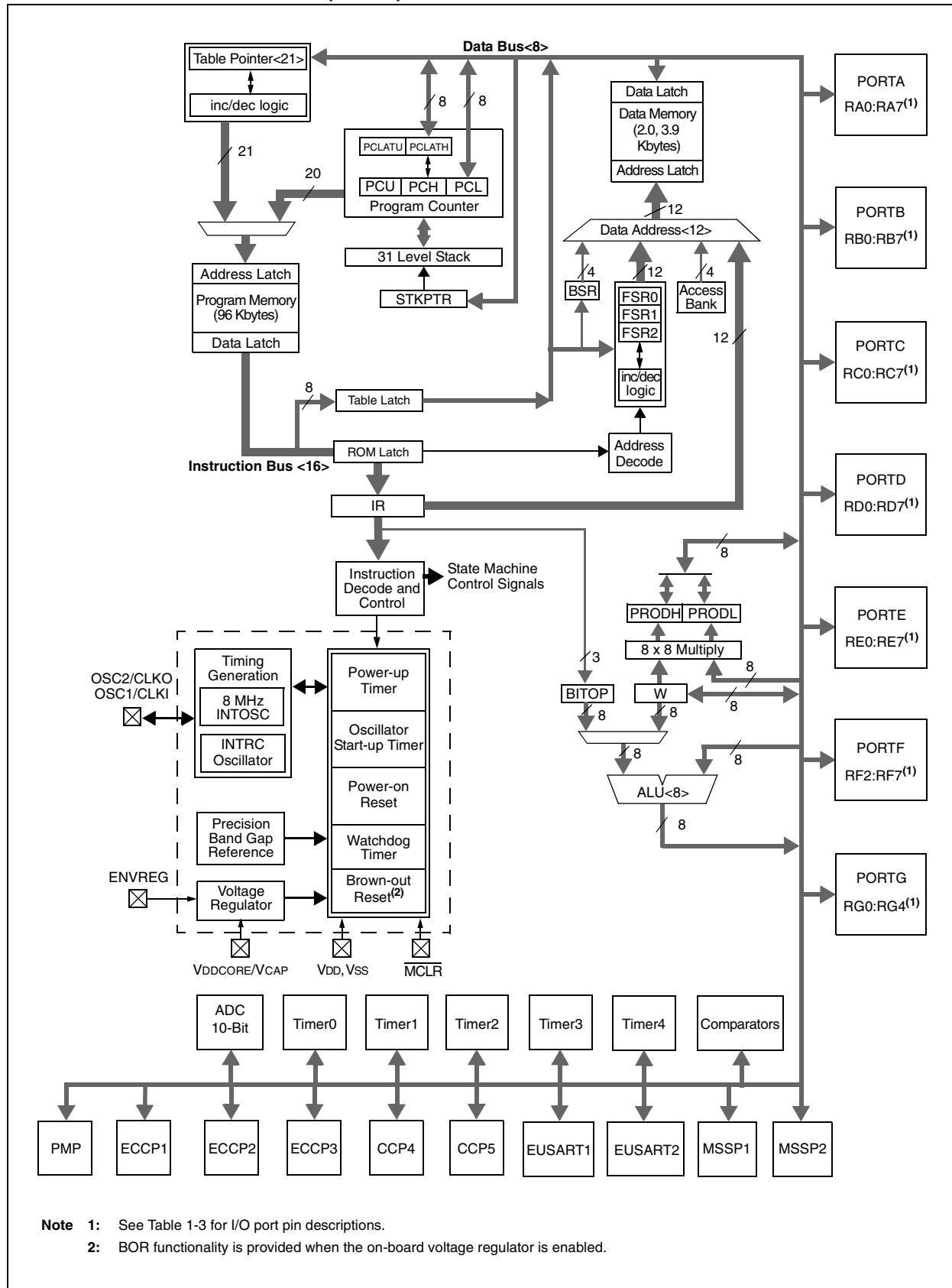
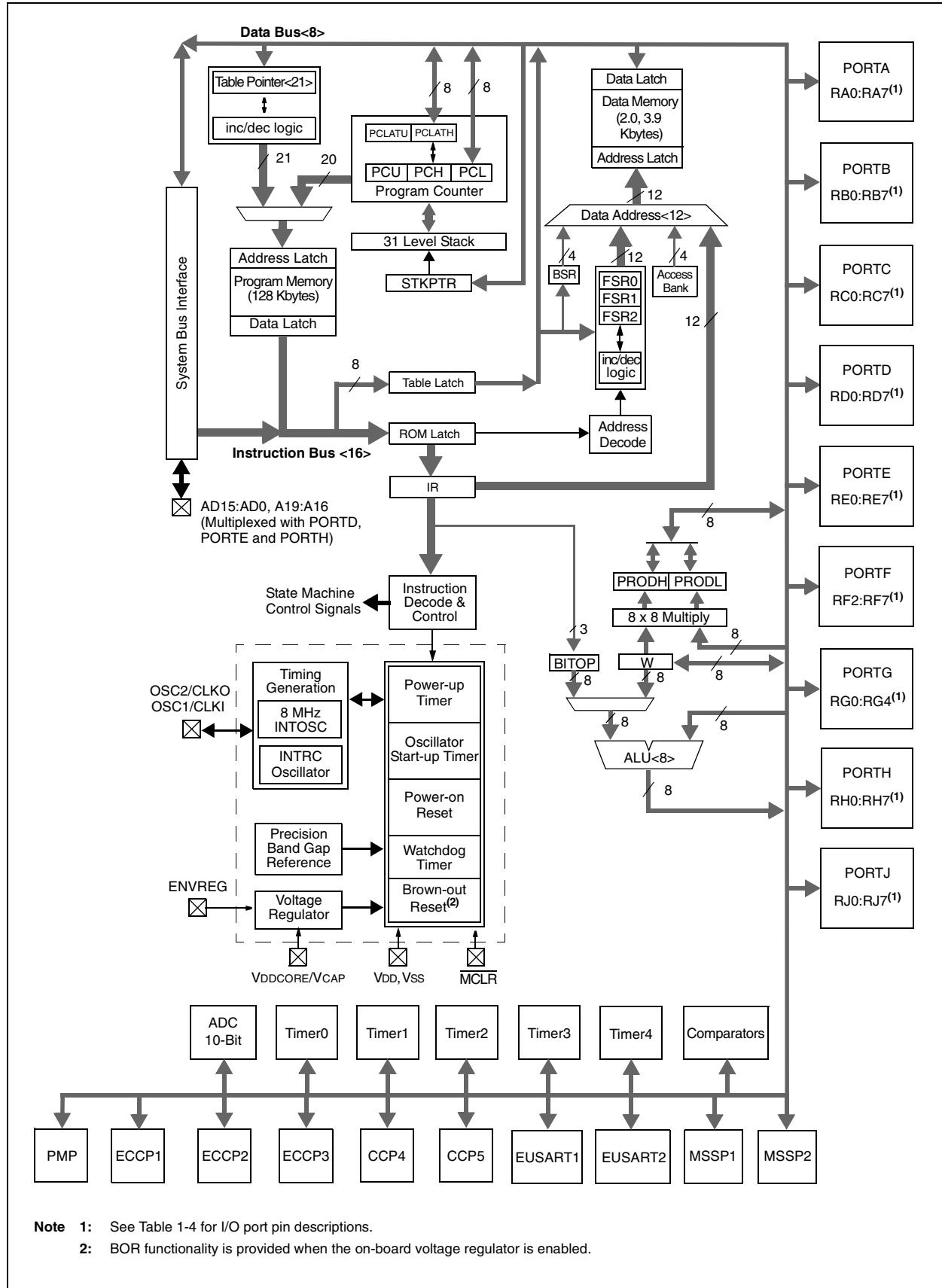


FIGURE 1-2: PIC18F8XJ1X (80-PIN) BLOCK DIAGRAM



PIC18F87J11 FAMILY

TABLE 1-3: PIC18F6XJ1X PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	64-TQFP			
MCLR	7	I	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
OSC1/CLKI/RA7	39	I	ST	Oscillator crystal or external clock input. Available only in external oscillator modes (EC/ECPLL and HS/HSPLL). Main oscillator input connection.
OSC1		I	ST	Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; CMOS otherwise.
CLKI		I	CMOS	Main clock input connection.
RA7		I/O	TTL	External clock source input. Always associated with pin function OSC1. (See related OSC1/CLKI, OSC2/CLKO pins.) General purpose I/O pin. Available only in INTIO2 and INTPLL2 Oscillator modes.
OSC2/CLKO/RA6	40	O	—	Oscillator crystal or clock output. Available only in external oscillator modes (EC/ECPLL and HS/HSPLL). Main oscillator feedback output connection.
OSC2		O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO		O	—	System cycle clock output ($F_{osc}/4$). In EC, ECPLL, INTIO1 and INTPLL1 Oscillator modes, OSC2 pin outputs CLKO which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
RA6		I/O	TTL	General purpose I/O pin. Available only in INTIO1 and INTPLL1 Oscillator modes.

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

I = Input

P = Power

CMOS = CMOS compatible input or output

Analog = Analog input

O = Output

OD = Open-Drain (no P diode to VDD)

Note 1: Default assignment for ECCP2/P2A when Configuration bit, CCP2MX, is set.

2: Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared.

PIC18F87J11 FAMILY

TABLE 1-3: PIC18F6XJ1X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number 64-TQFP	Pin Type	Buffer Type	Description
RA0/AN0 RA0 AN0	24	I/O I	TTL Analog	PORTA is a bidirectional I/O port. Digital I/O. Analog input 0.
RA1/AN1 RA1 AN1	23	I/O I	TTL Analog	Digital I/O. Analog input 1.
RA2/AN2/VREF- RA2 AN2 VREF-	22	I/O I I	TTL Analog Analog	Digital I/O. Analog input 2. A/D reference voltage (low) input.
RA3/AN3/VREF+ RA3 AN3 VREF+	21	I/O I I	TTL Analog Analog	Digital I/O. Analog input 3. A/D reference voltage (high) input.
RA4/T0CKI RA4 T0CKI	28	I/O I	ST ST	Digital I/O. Timer0 external clock input.
RA5/AN4 RA5 AN4	27	I/O I	TTL Analog	Digital I/O. Analog input 4.
RA6	—	—	—	See the OSC2/CLKO/RA6 pin.
RA7	—	—	—	See the OSC1/CLKI/RA7 pin.

Legend: TTL = TTL compatible input

CMOS = CMOS compatible input or output

ST = Schmitt Trigger input with CMOS levels

Analog = Analog input

I = Input

O = Output

P = Power

OD = Open-Drain (no P diode to VDD)

Note 1: Default assignment for CCP2/P2A when Configuration bit, CCP2MX, is set.

2: Alternate assignment for CCP2/P2A when Configuration bit, CCP2MX, is cleared.

PIC18F87J11 FAMILY

TABLE 1-3: PIC18F6XJ1X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number 64-TQFP	Pin Type	Buffer Type	Description
RB0/FLT0/INT0	48	I/O	TTL	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0		I	ST	Digital I/O.
FLT0		I	ST	ECCP1/2/3 Fault input.
INT0		I	ST	External interrupt 0.
RB1/INT1/PMA4	47	I/O	TTL	Digital I/O.
RB1		I	ST	External interrupt 1.
INT1		O	—	Parallel Master Port address.
PMA4				
RB2/INT2/PMA3	46	I/O	TTL	Digital I/O.
RB2		I	ST	External interrupt 2.
INT2		O	—	Parallel Master Port address.
PMA3				
RB3/INT3/PMA2	45	I/O	TTL	Digital I/O.
RB3		I	ST	External interrupt 3.
INT3		O	—	Parallel Master Port address.
PMA2				
RB4/KBI0/PMA1	44	I/O	TTL	Digital I/O.
RB4		I	TTL	Interrupt-on-change pin.
KBI0		I/O	—	Parallel Master Port address.
PMA1				
RB5/KBI1/PMA0	43	I/O	TTL	Digital I/O.
RB5		I	TTL	Interrupt-on-change pin.
KBI1		I/O	—	Parallel Master Port address.
PMA0				
RB6/KBI2/PGC	42	I/O	TTL	Digital I/O.
RB6		I	TTL	Interrupt-on-change pin.
KBI2		I/O	ST	In-Circuit Debugger and ICSP™ programming clock pin.
PGC				
RB7/KBI3/PGD	37	I/O	TTL	Digital I/O.
RB7		I	TTL	Interrupt-on-change pin.
KBI3		I/O	ST	In-Circuit Debugger and ICSP programming data pin.
PGD				

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

I = Input

P = Power

CMOS = CMOS compatible input or output

Analog = Analog input

O = Output

OD = Open-Drain (no P diode to VDD)

Note 1: Default assignment for ECCP2/P2A when Configuration bit, CCP2MX, is set.

2: Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared.

PIC18F87J11 FAMILY

TABLE 1-3: PIC18F6XJ1X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number 64-TQFP	Pin Type	Buffer Type	Description
RC0/T1OSO/T13CKI RC0 T1OSO T13CKI	30	I/O O I	ST — ST	PORTC is a bidirectional I/O port. Digital I/O. Timer1 oscillator output. Timer1/Timer3 external clock input.
RC1/T1OSI/ECCP2/P2A RC1 T1OSI ECCP2 ⁽¹⁾ P2A ⁽¹⁾	29	I/O I I/O O	ST CMOS ST —	Digital I/O. Timer1 oscillator input. Capture 2 input/Compare 2 output/PWM2 output. ECCP2 PWM output A.
RC2/ECCP1/P1A RC2 ECCP1 P1A	33	I/O I/O O	ST ST —	Digital I/O. Capture 1 input/Compare 1 output/PWM1 output. ECCP1 PWM output A.
RC3/SCK1/SCL1 RC3 SCK1 SCL1	34	I/O I/O I/O	ST ST ST	Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I ² C TM mode.
RC4/SDI1/SDA1 RC4 SDI1 SDA1	35	I/O I I/O	ST ST ST	Digital I/O. SPI data in. I ² C data I/O.
RC5/SDO1 RC5 SDO1	36	I/O O	ST —	Digital I/O. SPI data out.
RC6/TX1/CK1 RC6 TX1 CK1	31	I/O O I/O	ST — ST	Digital I/O. EUSART1 asynchronous transmit. EUSART1 synchronous clock (see related RX1/DT1).
RC7/RX1/DT1 RC7 RX1 DT1	32	I/O I I/O	ST ST ST	Digital I/O. EUSART1 asynchronous receive. EUSART1 synchronous data (see related TX1/CK1).

Legend: TTL = TTL compatible input

CMOS = CMOS compatible input or output

ST = Schmitt Trigger input with CMOS levels

Analog = Analog input

I = Input

O = Output

P = Power

OD = Open-Drain (no P diode to VDD)

Note 1: Default assignment for ECCP2/P2A when Configuration bit, CCP2MX, is set.

2: Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared.

PIC18F87J11 FAMILY

TABLE 1-3: PIC18F6XJ1X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number 64-TQFP	Pin Type	Buffer Type	Description
RD0/PMD0	58	I/O	ST	PORTD is a bidirectional I/O port.
RD0		I/O	TTL	Digital I/O.
PMD0				Parallel Master Port data.
RD1/PMD1	55	I/O	ST	Digital I/O.
RD1		I/O	TTL	Parallel Master Port data.
PMD1				
RD2/PMD2	54	I/O	ST	Digital I/O.
RD2		I/O	TTL	Parallel Master Port data.
PMD2				
RD3/PMD3	53	I/O	ST	Digital I/O.
RD3		I/O	TTL	Parallel Master Port data.
PMD3				
RD4/PMD4/SDO2	52	I/O	ST	Digital I/O.
RD4		I/O	TTL	Parallel Master Port data.
PMD4				
SDO2		O	—	SPI data out.
RD5/PMD5/SDI2/SDA2	51	I/O	ST	Digital I/O.
RD5		I/O	TTL	Parallel Master Port data.
PMD5				
SDI2		I	ST	SPI data in.
SDA2		I/O	ST	I ² C™ data I/O.
RD6/PMD6/SCK2/SCL2	50	I/O	ST	Digital I/O.
RD6		I/O	TTL	Parallel Master Port data.
PMD6				
SCK2		I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL2		I/O	ST	Synchronous serial clock input/output for I ² C mode.
RD7/PMD7/SS2	49	I/O	ST	Digital I/O.
RD7		I/O	TTL	Parallel Master Port data.
PMD7				
SS2		I	TTL	SPI slave select input.

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

I = Input

P = Power

CMOS = CMOS compatible input or output

Analog = Analog input

O = Output

OD = Open-Drain (no P diode to VDD)

Note 1: Default assignment for ECCP2/P2A when Configuration bit, CCP2MX, is set.

2: Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared.

PIC18F87J11 FAMILY

TABLE 1-3: PIC18F6XJ1X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number 64-TQFP	Pin Type	Buffer Type	Description
RE0/PMRD/P2D RE0 PMRD P2D	2	I/O I/O O	ST — —	PORTE is a bidirectional I/O port. Digital I/O. Parallel Master Port read strobe. ECCP2 PWM output D.
RE1/PMWR/P2C RE1 PMWR P2C	1	I/O I/O O	ST — —	Digital I/O. Parallel Master Port write strobe. ECCP2 PWM output C.
RE2/PMBE/P2B RE2 PMBE P2B	64	I/O O O	ST — —	Digital I/O. Parallel Master Port byte enable ECCP2 PWM output B.
RE3/PMA13/P3C/REF0 RE3 PMA13 P3C REF0	63	I/O O O O	ST — — —	Digital I/O. Parallel Master Port address. ECCP3 PWM output C. Reference clock out.
RE4/PMA12/P3B RE4 PMA12 P3B	62	I/O O O	ST — —	Digital I/O. Parallel Master Port address. ECCP3 PWM output B.
RE5/PMA11/P1C RE5 PMA11 P1C	61	I/O O O	ST — —	Digital I/O. Parallel Master Port address. ECCP1 PWM output C.
RE6/PMA10/P1B RE6 PMA10 P1B	60	I/O O O	ST — —	Digital I/O. Parallel Master Port address. ECCP1 PWM output B.
RE7/PMA9/ECCP2/P2A RE7 PMA9 ECCP2 ⁽²⁾ P2A ⁽²⁾	59	I/O O I/O O	ST — ST —	Digital I/O. Parallel Master Port address. Capture 2 input/Compare 2 output/PWM2 output. ECCP2 PWM output A.

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

I = Input

P = Power

CMOS = CMOS compatible input or output

Analog = Analog input

O = Output

OD = Open-Drain (no P diode to VDD)

Note 1: Default assignment for ECCP2/P2A when Configuration bit, CCP2MX, is set.

2: Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared.

PIC18F87J11 FAMILY

TABLE 1-3: PIC18F6XJ1X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number 64-TQFP	Pin Type	Buffer Type	Description
RF1/AN6/C2OUT	17	I/O	ST	PORTF is a bidirectional I/O port.
RF1		I	Analog	Digital I/O.
AN6		O	—	Analog input 6.
C2OUT		—	—	Comparator 2 output.
RF2/PMA5/AN7/C1OUT	16	I/O	ST	Digital I/O.
RF2		O	—	Parallel Master Port address.
PMA5		I	Analog	Analog input 7.
AN7		O	—	Comparator 1 output.
C1OUT		—	—	
RF3/AN8/C2INB	15	I	ST	Digital input.
RF3		I	Analog	Analog input 8.
AN8		I	Analog	Comparator 2 input B.
C2INB		—	—	
RF4/AN9/C2INA	14	I	ST	Digital input.
RF4		I	Analog	Analog input 8.
AN9		I	Analog	Comparator 2 input A.
C2INA		—	—	
RF5/AN10/C1INB/CVREF	13	I	ST	Digital input.
RF5		I	Analog	Analog input 10.
AN10		I	Analog	Comparator 1 input B.
C1INB		O	Analog	Comparator reference voltage output.
CVREF		—	—	
RF6/AN11/C1INA	12	I/O	ST	Digital I/O.
RF6		I	Analog	Analog input 11.
AN11		I	Analog	Comparator 1 input A.
C1INA		—	—	
RF7/SS1	11	I/O	ST	Digital I/O.
RF7		I	TTL	SPI slave select input.
SS1		—	—	

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

I = Input

P = Power

CMOS = CMOS compatible input or output

Analog = Analog input

O = Output

OD = Open-Drain (no P diode to VDD)

Note 1: Default assignment for ECCP2/P2A when Configuration bit, CCP2MX, is set.

2: Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared.

PIC18F87J11 FAMILY

TABLE 1-3: PIC18F6XJ1X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number 64-TQFP	Pin Type	Buffer Type	Description
RG0/PMA8/ECCP3/P3A RG0 PMA8 ECCP3 P3A	3	I/O	ST	PORTG is a bidirectional I/O port.
		O	—	Digital I/O.
		I/O	ST	Parallel Master Port address.
		O	—	Capture 3 input/Compare 3 output/PWM3 output. ECCP3 PWM output A.
RG1/PMA7/TX2/CK2 RG1 PMA7 TX2 CK2	4	I/O	ST	Digital I/O.
		O	—	Parallel Master Port address.
		O	—	EUSART2 asynchronous transmit.
		I/O	ST	EUSART2 synchronous clock (see related RX2/DT2).
RG2/PMA6/RX2/DT2 RG2 PMA6 RX2 DT2	5	I/O	ST	Digital I/O.
		O	—	Parallel Master Port address.
		I	ST	EUSART2 asynchronous receive.
		I/O	ST	EUSART2 synchronous data (see related TX2/CK2).
RG3/PMCS1/CCP4/P3D RG3 PMCS1 CCP4 P3D	6	I/O	ST	Digital I/O.
		O	—	Parallel Master Port chip select 1.
		I/O	ST	Capture 4 input/Compare 4 output/PWM4 output.
		O	—	ECCP3 PWM output D.
RG4/PMCS2/CCP5/P1D RG4 PMCS2 CCP5 P1D	8	I/O	ST	Digital I/O.
		O	—	Parallel Master Port chip select 2.
		I/O	ST	Capture 5 input/Compare 5 output/PWM5 output.
		O	—	ECCP1 PWM output D.
VSS	9, 25, 41, 56	P	—	Ground reference for logic and I/O pins.
VDD	26, 38, 57	P	—	Positive supply for peripheral digital logic and I/O pins.
AVss	20	P	—	Ground reference for analog modules.
AVDD	19	P	—	Positive supply for analog modules.
ENVREG	18	I	ST	Enable for on-chip voltage regulator.
VDDCORE/VCAP VDDCORE VCAP	10	P	—	Core logic power or external filter capacitor connection. Positive supply for microcontroller core logic (regulator disabled).
				External filter capacitor connection (regulator enabled).
		P	—	

Legend: TTL = TTL compatible input

CMOS = CMOS compatible input or output

ST = Schmitt Trigger input with CMOS levels

Analog = Analog input

I = Input

O = Output

P = Power

OD = Open-Drain (no P diode to VDD)

Note 1: Default assignment for ECCP2/P2A when Configuration bit, CCP2MX, is set.

2: Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared.

PIC18F87J11 FAMILY

TABLE 1-4: PIC18F8XJ1X PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	80-TQFP			
MCLR	9	I	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
OSC1/CLKI/RA7	49	I	ST	Oscillator crystal or external clock input. Available only in external oscillator modes (EC/ECPLL and HS/HSPLL). Main oscillator input connection.
OSC1		I	ST	Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; CMOS otherwise.
CLKI		I	CMOS	Main clock input connection.
RA7		I/O	TTL	External clock source input. Always associated with pin function OSC1. (See related OSC1/CLKI, OSC2/CLKO pins.) General purpose I/O pin. Available only in INTIO2 and INTPLL2 Oscillator modes.
OSC2/CLKO/RA6	50	O	—	Oscillator crystal or clock output. Available only in external oscillator modes (EC/ECPLL and HS/HSPLL). Main oscillator feedback output connection.
OSC2		O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO		O	—	System cycle clock output ($F_{osc}/4$). In EC, ECPLL, INTIO1 and INTPLL1 Oscillator modes, OSC2 pin outputs CLKO which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
RA6		I/O	TTL	General purpose I/O pin. Available only in INTIO and INTPLL Oscillator modes.

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

I = Input

P = Power

CMOS = CMOS compatible input or output

Analog = Analog input

O = Output

OD = Open-Drain (no P diode to VDD)

Note 1: Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared (Extended Microcontroller mode).

2: Default assignment for ECCP2/P2A for all devices in all operating modes (CCP2MX is set).

3: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).

4: Alternate assignment for ECCP2/P2A when CCP2MX is cleared (Microcontroller mode).

5: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

6: Default assignment for PMP data and control pins when PMPMX Configuration bit is set.

7: Alternate assignment for PMP data and control pins when PMPMX Configuration bit is cleared (programmed).

PIC18F87J11 FAMILY

TABLE 1-4: PIC18F8XJ1X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	80-TQFP			
RA0/AN0 RA0 AN0	30	I/O I	TTL Analog	PORTA is a bidirectional I/O port. Digital I/O. Analog input 0.
RA1/AN1 RA1 AN1	29	I/O I	TTL Analog	Digital I/O. Analog input 1.
RA2/AN2/VREF- RA2 AN2 VREF-	28	I/O I I	TTL Analog Analog	Digital I/O. Analog input 2. A/D reference voltage (low) input.
RA3/AN3/VREF+ RA3 AN3 VREF+	27	I/O I I	TTL Analog Analog	Digital I/O. Analog input 3. A/D reference voltage (high) input.
RA4/PMD5/T0CKI RA4 PMD5 ⁽⁷⁾ T0CKI	34	I/O I/O I	ST TTL ST	Digital I/O. Parallel Master Port data. Timer0 external clock input.
RA5/PMD4/AN4 RA5 PMD4 ⁽⁷⁾ AN4	33	I/O I/O I	TTL TTL Analog	Digital I/O. Parallel Master Port data. Analog input 4.
RA6	—	—	—	See the OSC2/CLKO/RA6 pin.
RA7	—	—	—	See the OSC1/CLKI/RA7 pin.

Legend: TTL = TTL compatible input
 ST = Schmitt Trigger input with CMOS levels
 I = Input
 P = Power

CMOS = CMOS compatible input or output
 Analog = Analog input
 O = Output
 OD = Open-Drain (no P diode to VDD)

- Note 1:** Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared (Extended Microcontroller mode).
2: Default assignment for ECCP2/P2A for all devices in all operating modes (CCP2MX is set).
3: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).
4: Alternate assignment for ECCP2/P2A when CCP2MX is cleared (Microcontroller mode).
5: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).
6: Default assignment for PMP data and control pins when PMPMX Configuration bit is set.
7: Alternate assignment for PMP data and control pins when PMPMX Configuration bit is cleared (programmed).

PIC18F87J11 FAMILY

TABLE 1-4: PIC18F8XJ1X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	80-TQFP			
RB0/FLT0/INT0 RB0 FLT0 INT0	58	I/O I I	TTL ST ST	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs. Digital I/O. ECCP1/2/3 Fault input. External interrupt 0.
RB1/INT1/PMA4 RB1 INT1 PMA4	57	I/O I O	TTL ST —	Digital I/O. External interrupt 1. Parallel Master Port address.
RB2/INT2/PMA3 RB2 INT2 PMA3	56	I/O I O	TTL ST —	Digital I/O. External interrupt 2. Parallel Master Port address.
RB3/INT3/PMA2/ ECCP2/P2A RB3 INT3 PMA2 ECCP2 ⁽¹⁾ P2A ⁽¹⁾	55	I/O I O I/O O	TTL ST — ST —	Digital I/O. External interrupt 3. Parallel Master Port address. Capture 2 input/Compare 2 output/PWM2 output. ECCP2 PWM output A.
RB4/KBI0/PMA1 RB4 KBI0 PMA1	54	I/O I I/O	TTL TTL —	Digital I/O. Interrupt-on-change pin. Parallel Master Port address.
RB5/KBI1/PMA0 RB5 KBI1 PMA0	53	I/O I I/O	TTL TTL —	Digital I/O. Interrupt-on-change pin. Parallel Master Port address.
RB6/KBI2/PGC RB6 KBI2 PGC	52	I/O I I/O	TTL TTL ST	Digital I/O. Interrupt-on-change pin. In-Circuit Debugger and ICSP™ programming clock pin.
RB7/KBI3/PGD RB7 KBI3 PGD	47	I/O I I/O	TTL TTL ST	Digital I/O. Interrupt-on-change pin. In-Circuit Debugger and ICSP programming data pin.

Legend: TTL = TTL compatible input

CMOS = CMOS compatible input or output

ST = Schmitt Trigger input with CMOS levels

Analog = Analog input

I = Input

O = Output

P = Power

OD = Open-Drain (no P diode to VDD)

Note 1: Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared (Extended Microcontroller mode).

2: Default assignment for ECCP2/P2A for all devices in all operating modes (CCP2MX is set).

3: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).

4: Alternate assignment for ECCP2/P2A when CCP2MX is cleared (Microcontroller mode).

5: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

6: Default assignment for PMP data and control pins when PMPMX Configuration bit is set.

7: Alternate assignment for PMP data and control pins when PMPMX Configuration bit is cleared (programmed).

PIC18F87J11 FAMILY

TABLE 1-4: PIC18F8XJ1X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	80-TQFP			
RC0/T1OSO/T13CKI RC0 T1OSO T13CKI	36	I/O O I	ST — ST	PORTC is a bidirectional I/O port. Digital I/O. Timer1 oscillator output. Timer1/Timer3 external clock input.
RC1/T1OSI/ECCP2/P2A RC1 T1OSI ECCP2 ⁽²⁾ P2A ⁽²⁾	35	I/O I I/O O	ST CMOS ST —	Digital I/O. Timer1 oscillator input. Capture 2 input/Compare 2 output/PWM2 output. ECCP2 PWM output A.
RC2/ECCP1/P1A RC2 ECCP1 P1A	43	I/O I/O O	ST ST —	Digital I/O. Capture 1 input/Compare 1 output/PWM1 output. ECCP1 PWM output A.
RC3/SCK1/SCL1 RC3 SCK1 SCL1	44	I/O I/O I/O	ST ST ST	Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I ² C TM mode.
RC4/SDI1/SDA1 RC4 SDI1 SDA1	45	I/O I I/O	ST ST ST	Digital I/O. SPI data in. I ² C data I/O.
RC5/SDO1 RC5 SDO1	46	I/O O	ST —	Digital I/O. SPI data out.
RC6/TX1/CK1 RC6 TX1 CK1	37	I/O O I/O	ST — ST	Digital I/O. EUSART1 asynchronous transmit. EUSART1 synchronous clock (see related RX1/DT1).
RC7/RX1/DT1 RC7 RX1 DT1	38	I/O I I/O	ST ST ST	Digital I/O. EUSART1 asynchronous receive. EUSART1 synchronous data (see related TX1/CK1).

Legend: TTL = TTL compatible input

CMOS = CMOS compatible input or output

ST = Schmitt Trigger input with CMOS levels

Analog = Analog input

I = Input

O = Output

P = Power

OD = Open-Drain (no P diode to VDD)

Note 1: Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared (Extended Microcontroller mode).

2: Default assignment for ECCP2/P2A for all devices in all operating modes (CCP2MX is set).

3: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).

4: Alternate assignment for ECCP2/P2A when CCP2MX is cleared (Microcontroller mode).

5: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

6: Default assignment for PMP data and control pins when PMPMX Configuration bit is set.

7: Alternate assignment for PMP data and control pins when PMPMX Configuration bit is cleared (programmed).

PIC18F87J11 FAMILY

TABLE 1-4: PIC18F8XJ1X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number 80-TQFP	Pin Type	Buffer Type	Description
RD0/AD0/PMD0 RD0 AD0 PMD0 ⁽⁶⁾	72	I/O I/O I/O	ST TTL TTL	PORTD is a bidirectional I/O port. Digital I/O. External memory address/data 0. Parallel Master Port data.
RD1/AD1/PMD1 RD1 AD1 PMD1 ⁽⁶⁾	69	I/O I/O I/O	ST TTL TTL	Digital I/O. External memory address/data 1. Parallel Master Port data.
RD2/AD2/PMD2 RD2 AD2 PMD2 ⁽⁶⁾	68	I/O I/O I/O	ST TTL TTL	Digital I/O. External memory address/data 2. Parallel Master Port data.
RD3/AD3/PMD3 RD3 AD3 PMD3 ⁽⁶⁾	67	I/O I/O I/O	ST TTL TTL	Digital I/O. External memory address/data 3. Parallel Master Port data.
RD4/AD4/PMD4/SDO2 RD4 AD4 PMD4 ⁽⁶⁾ SDO2	66	I/O I/O I/O O	ST TTL TTL —	Digital I/O. External memory address/data 4. Parallel Master Port data. SPI data out.
RD5/AD5/PMD5/ SDI2/SDA2 RD5 AD5 PMD5 ⁽⁶⁾ SDI2 SDA2	65	I/O I/O I/O I/O	ST TTL TTL ST	Digital I/O. External memory address/data 5. Parallel Master Port data. SPI data in. I ² C™ data I/O.
RD6/AD6/PMD6/ SCK2/SCL2 RD6 AD6 PMD6 ⁽⁶⁾ SCK2 SCL2	64	I/O I/O I/O I/O	ST TTL TTL ST	Digital I/O. External memory address/data 6. Parallel Master Port data. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I ² C mode.
RD7/AD7/PMD7/SS2 RD7 AD7 PMD7 ⁽⁶⁾ SS2	63	I/O I/O I/O I	ST TTL TTL TTL	Digital I/O. External memory address/data 7. Parallel Master Port data. SPI slave select input.

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

I = Input

P = Power

CMOS = CMOS compatible input or output

Analog = Analog input

O = Output

OD = Open-Drain (no P diode to VDD)

Note 1: Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared (Extended Microcontroller mode).

2: Default assignment for ECCP2/P2A for all devices in all operating modes (CCP2MX is set).

3: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).

4: Alternate assignment for ECCP2/P2A when CCP2MX is cleared (Microcontroller mode).

5: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

6: Default assignment for PMP data and control pins when PMPMX Configuration bit is set.

7: Alternate assignment for PMP data and control pins when PMPMX Configuration bit is cleared (programmed).

PIC18F87J11 FAMILY

TABLE 1-4: PIC18F8XJ1X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	80-TQFP			
RE0/AD8/PMRD/P2D RE0 AD8 PMRD ⁽⁶⁾ P2D	4	I/O I/O I/O O	ST TTL — —	PORTE is a bidirectional I/O port. Digital I/O. External memory address/data 8. Parallel Master Port read strobe. ECCP2 PWM output D.
RE1/AD9/PMWR/P2C RE1 AD9 PMWR ⁽⁶⁾ P2C	3	I/O I/O I/O O	ST TTL — —	Digital I/O. External memory address/data 9. Parallel Master Port write strobe. ECCP2 PWM output C.
RE2/AD10/PMBE/P2B RE2 AD10 PMBE ⁽⁶⁾ P2B	78	I/O I/O O O	ST TTL — —	Digital I/O. External memory address/data 10. Parallel Master Port byte enable. ECCP2 PWM output B.
RE3/AD11/PMA13/P3C/REF0 RE3 AD11 PMA13 P3C ⁽³⁾ REF0	77	I/O I/O O O O	ST TTL — — —	Digital I/O. External memory address/data 11. Parallel Master Port address. ECCP3 PWM output C. Reference clock out.
RE4/AD12/PMA12/P3B RE4 AD12 PMA12 P3B ⁽³⁾	76	I/O I/O O O	ST TTL — —	Digital I/O. External memory address/data 12. Parallel Master Port address. ECCP3 PWM output B.
RE5/AD13/PMA11/P1C RE5 AD13 PMA11 P1C ⁽³⁾	75	I/O I/O O O	ST TTL — —	Digital I/O. External memory address/data 13. Parallel Master Port address. ECCP1 PWM output C.
RE6/AD14/PMA10/P1B RE6 AD14 PMA10 P1B ⁽³⁾	74	I/O I/O O O	ST TTL — —	Digital I/O. External memory address/data 14. Parallel Master Port address. ECCP1 PWM output B.
RE7/AD15/PMA9/ECCP2/P2A RE7 AD15 PMA9 ECCP2 ⁽⁴⁾ P2A ⁽⁴⁾	73	I/O I/O O I/O O	ST TTL — ST —	Digital I/O. External memory address/data 15. Parallel Master Port address. Capture 2 input/Compare 2 output/PWM2 output. ECCP2 PWM output A.

Legend: TTL = TTL compatible input

CMOS = CMOS compatible input or output

ST = Schmitt Trigger input with CMOS levels

Analog = Analog input

I = Input

O = Output

P = Power

OD = Open-Drain (no P diode to VDD)

Note 1: Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared (Extended Microcontroller mode).

2: Default assignment for ECCP2/P2A for all devices in all operating modes (CCP2MX is set).

3: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).

4: Alternate assignment for ECCP2/P2A when CCP2MX is cleared (Microcontroller mode).

5: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

6: Default assignment for PMP data and control pins when PMPMX Configuration bit is set.

7: Alternate assignment for PMP data and control pins when PMPMX Configuration bit is cleared (programmed).

PIC18F87J11 FAMILY

TABLE 1-4: PIC18F8XJ1X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number 80-TQFP	Pin Type	Buffer Type	Description
RF1/AN6/C2OUT	23	I/O	ST	PORTF is a bidirectional I/O port.
RF1		I	Analog	Digital I/O.
AN6		O	—	Analog input 6.
C2OUT		—	—	Comparator 2 output.
RF2/PMA5/AN7/C1OUT	18	I/O	ST	Digital I/O.
RF2		O	—	Parallel Master Port address.
PMA5		I	Analog	Analog input 7.
AN7		O	—	Comparator 1 output.
C1OUT		—	—	
RF3/AN8/C2INB	17	I	ST	Digital input.
RF3		I	Analog	Analog input 8.
AN8		I	Analog	Comparator 2 input B.
C2INB		—	—	
RF4/AN9/C2INA	16	I	ST	Digital input.
RF4		I	Analog	Analog input 8.
AN9		I	Analog	Comparator 2 input A.
C2INA		—	—	
RF5/PMD2/AN10/ C1INB/CVREF	15	I/O	ST	Digital I/O.
RF5		I/O	TTL	Parallel Master Port address.
PMD2 ⁽⁷⁾		I	Analog	Analog input 10.
AN10		I	Analog	Comparator 1 input B.
C1INB		O	Analog	Comparator reference voltage output.
CVREF		—	—	
RF6/PMD1/AN11/C1INA	14	I/O	ST	Digital I/O.
RF6		I/O	TTL	Parallel Master Port address.
PMD1 ⁽⁷⁾		I	Analog	Analog input 11.
AN11		I	Analog	Comparator 1 input A.
C1INA		—	—	
RF7/PMD0/SS1	13	I/O	ST	Digital I/O.
RF7		I/O	TTL	Parallel Master Port address.
PMD0 ⁽⁷⁾		I	TTL	SPI slave select input.
SS1		—	—	

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

I = Input

P = Power

CMOS = CMOS compatible input or output

Analog = Analog input

O = Output

OD = Open-Drain (no P diode to Vdd)

Note 1: Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared (Extended Microcontroller mode).

2: Default assignment for ECCP2/P2A for all devices in all operating modes (CCP2MX is set).

3: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).

4: Alternate assignment for ECCP2/P2A when CCP2MX is cleared (Microcontroller mode).

5: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

6: Default assignment for PMP data and control pins when PMPMX Configuration bit is set.

7: Alternate assignment for PMP data and control pins when PMPMX Configuration bit is cleared (programmed).

PIC18F87J11 FAMILY

TABLE 1-4: PIC18F8XJ1X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	80-TQFP			
RG0/PMA8/ECCP3/P3A RG0 PMA8 ECCP3 P3A	5	I/O O I/O O	ST — ST —	PORTG is a bidirectional I/O port. Digital I/O. Parallel Master Port address. Capture 3 input/Compare 3 output/PWM3 output. ECCP3 PWM output A.
RG1/PMA7/TX2/CK2 RG1 PMA7 TX2 CK2	6	I/O O O I/O	ST — — ST	Digital I/O. Parallel Master Port address. EUSART2 asynchronous transmit. EUSART2 synchronous clock (see related RX2/DT2).
RG2/PMA6/RX2/DT2 RG2 PMA6 RX2 DT2	7	I/O I/O I I/O	ST — ST ST	Digital I/O. Parallel Master Port address. EUSART2 asynchronous receive. EUSART2 synchronous data (see related TX2/CK2).
RG3/PMCS1/CCP4/P3D RG3 PMCS1 CCP4 P3D	8	I/O I/O I/O O	ST — ST —	Digital I/O. Parallel Master Port chip select 1. Capture 4 input/Compare 4 output/PWM4 output. ECCP3 PWM output D.
RG4/PMCS2/CCP5/P1D RG4 PMCS2 CCP5 P1D	10	I/O O I/O O	ST — ST —	Digital I/O. Parallel Master Port chip select 2. Capture 5 input/Compare 5 output/PWM5 output. ECCP1 PWM output D.

Legend: TTL = TTL compatible input
 ST = Schmitt Trigger input with CMOS levels
 I = Input
 P = Power

CMOS = CMOS compatible input or output
 Analog = Analog input
 O = Output
 OD = Open-Drain (no P diode to VDD)

- Note 1:** Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared (Extended Microcontroller mode).
2: Default assignment for ECCP2/P2A for all devices in all operating modes (CCP2MX is set).
3: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).
4: Alternate assignment for ECCP2/P2A when CCP2MX is cleared (Microcontroller mode).
5: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).
6: Default assignment for PMP data and control pins when PMPMX Configuration bit is set.
7: Alternate assignment for PMP data and control pins when PMPMX Configuration bit is cleared (programmed).

PIC18F87J11 FAMILY

TABLE 1-4: PIC18F8XJ1X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number 80-TQFP	Pin	Buffer	Description
		Type	Type	
RH0/A16 RH0 A16	79	I/O O	ST TTL	PORTH is a bidirectional I/O port. Digital I/O. External memory address/data 16.
RH1/A17 RH1 A17	80	I/O O	ST TTL	Digital I/O. External memory address/data 17.
RH2/A18/PMD7 RH2 A18 PMD7 ⁽⁷⁾	1	I/O O I/O	ST TTL TTL	Digital I/O. External memory address/data 18. Parallel Master Port data.
RH3/A19/PMD6 RH3 A19 PMD6 ⁽⁷⁾	2	I/O O I/O	ST TTL TTL	Digital I/O. External memory address/data 19. Parallel Master Port data.
RH4/PMD3/AN12/ P3C/C2INC RH4 PMD3 ⁽⁷⁾ AN12 P3C ⁽⁵⁾ C2INC	22	I/O I/O I O I	ST TTL Analog — Analog	Digital I/O. Parallel Master Port address. Analog input 12. ECCP3 PWM output C. Comparator 2 input C.
RH5/PMBE/AN13/ P3B/C2IND RH5 PMBE ⁽⁷⁾ AN13 P3B ⁽⁵⁾ C2IND	21	I/O O I O I	ST — Analog — Analog	Digital I/O. Parallel Master Port byte enable. Analog input 13. ECCP3 PWM output B. Comparator 2 input D.
RH6/PMRD/AN14/ P1C/C1INC RH6 PMRD ⁽⁷⁾ AN14 P1C ⁽⁵⁾ C1INC	20	I/O I/O I O I	ST — Analog — Analog	Digital I/O. Parallel Master Port read strobe. Analog input 14. ECCP1 PWM output C. Comparator 1 input C.
RH7/PMWR/AN15/P1B RH7 PMWR ⁽⁷⁾ AN15 P1B ⁽⁵⁾	19	I/O I/O I O	ST — Analog —	Digital I/O. Parallel Master Port write strobe. Analog input 15. ECCP1 PWM output B.

Legend: TTL = TTL compatible input

ST = Schmitt Trigger input with CMOS levels

I = Input

P = Power

CMOS = CMOS compatible input or output

Analog = Analog input

O = Output

OD = Open-Drain (no P diode to VDD)

Note 1: Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared (Extended Microcontroller mode).

2: Default assignment for ECCP2/P2A for all devices in all operating modes (CCP2MX is set).

3: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).

4: Alternate assignment for ECCP2/P2A when CCP2MX is cleared (Microcontroller mode).

5: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

6: Default assignment for PMP data and control pins when PMPMX Configuration bit is set.

7: Alternate assignment for PMP data and control pins when PMPMX Configuration bit is cleared (programmed).

PIC18F87J11 FAMILY

TABLE 1-4: PIC18F8XJ1X PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	80-TQFP			
RJ0/ALE RJ0 ALE	62	I/O O	ST —	PORTJ is a bidirectional I/O port. Digital I/O. External memory address latch enable.
RJ1/ <u>OE</u> RJ1 <u>OE</u>	61	I/O O	ST —	Digital I/O. External memory output enable.
RJ2/ <u>WRL</u> RJ2 <u>WRL</u>	60	I/O O	ST —	Digital I/O. External memory write low control.
RJ3/ <u>WRH</u> RJ3 <u>WRH</u>	59	I/O O	ST —	Digital I/O. External memory write high control.
RJ4/BA0 RJ4 BA0	39	I/O O	ST —	Digital I/O. External memory byte address 0 control.
RJ5/ <u>CE</u> RJ5 <u>CE</u>	40	I/O O	ST —	Digital I/O External memory chip enable control.
RJ6/ <u>LB</u> RJ6 <u>LB</u>	41	I/O O	ST —	Digital I/O. External memory low byte control.
RJ7/ <u>UB</u> RJ7 <u>UB</u>	42	I/O O	ST —	Digital I/O. External memory high byte control.
VSS	11, 31, 51, 70	P	—	Ground reference for logic and I/O pins.
VDD	32, 48, 71	P	—	Positive supply for peripheral digital logic and I/O pins.
AVss	26	P	—	Ground reference for analog modules.
AVDD	25	P	—	Positive supply for analog modules.
ENVREG	24	I	ST	Enable for on-chip voltage regulator.
VDDCORE/VCAP VDDCORE VCAP	12	P	—	Core logic power or external filter capacitor connection. Positive supply for microcontroller core logic (regulator disabled).
		P	—	External filter capacitor connection (regulator enabled).

Legend: TTL = TTL compatible input
 ST = Schmitt Trigger input with CMOS levels
 I = Input
 P = Power

CMOS = CMOS compatible input or output
 Analog = Analog input
 O = Output
 OD = Open-Drain (no P diode to VDD)

- Note 1:** Alternate assignment for ECCP2/P2A when Configuration bit, CCP2MX, is cleared (Extended Microcontroller mode).
2: Default assignment for ECCP2/P2A for all devices in all operating modes (CCP2MX is set).
3: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).
4: Alternate assignment for ECCP2/P2A when CCP2MX is cleared (Microcontroller mode).
5: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).
6: Default assignment for PMP data and control pins when PMPMX Configuration bit is set.
7: Alternate assignment for PMP data and control pins when PMPMX Configuration bit is cleared (programmed).

PIC18F87J11 FAMILY

NOTES:

2.0 OSCILLATOR CONFIGURATIONS

2.1 Oscillator Types

The PIC18F87J11 family of devices can be operated in eight different oscillator modes:

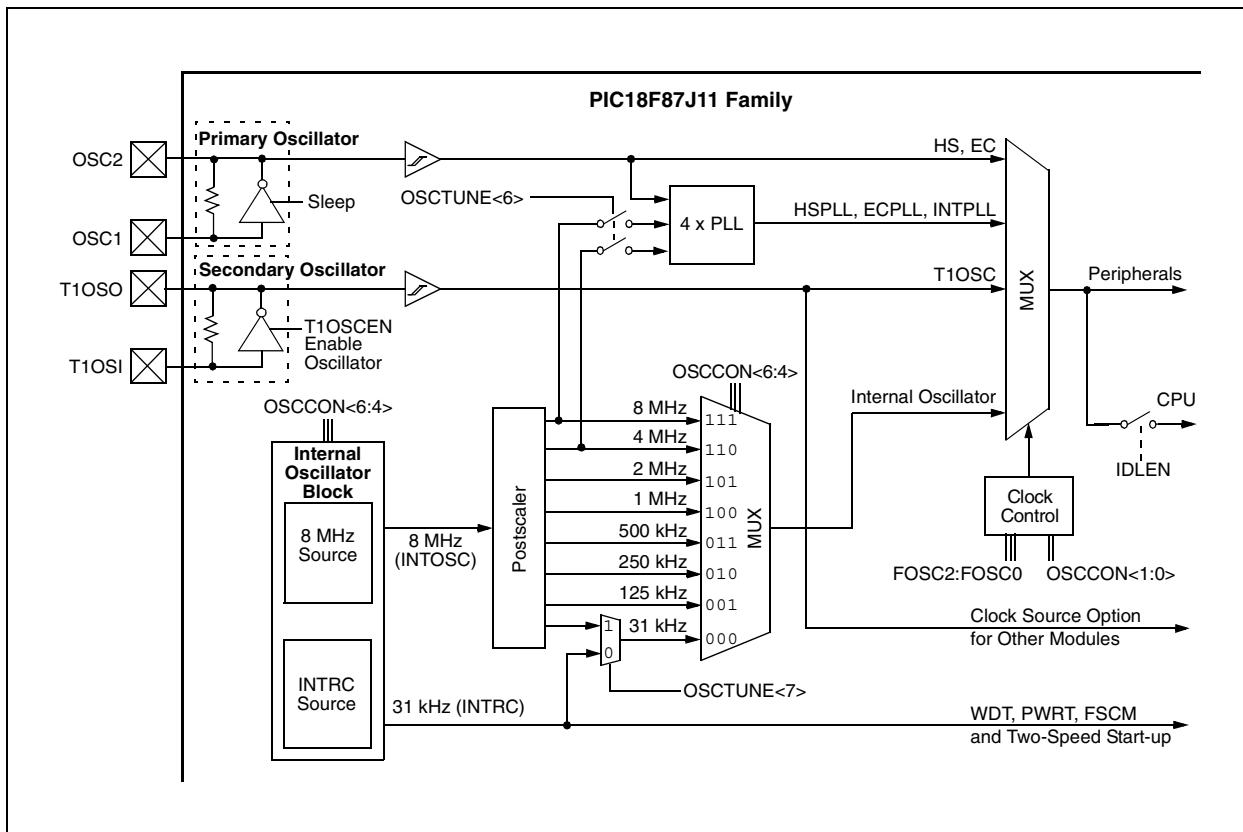
1. HS High-Speed Crystal/Resonator
2. HSPLL High-Speed Crystal/Resonator with Software PLL Control
3. EC External Clock with Fosc/4 Output
4. ECPLL External Clock with Software PLL Control
5. INTIO1 Internal Oscillator Block with Fosc/4 Output on RA6 and I/O on RA7
6. INTIO2 Internal Oscillator Block with I/O on RA6 and RA7
7. INTPLL1 Internal Oscillator Block with Software PLL Control, Fosc/4 Output on RA6 and I/O on RA7
8. INTPLL2 Internal Oscillator Block with Software PLL Control and I/O on RA6 and RA7

All of these modes are selected by the user by programming the FOSC2:FOSC0 Configuration bits.

In addition, PIC18F87J11 Family devices can switch between different clock sources, either under software control or automatically under certain conditions. This allows for additional power savings by managing device clock speed in real time without resetting the application.

The clock sources for the PIC18F87J11 family of devices are shown in Figure 2-1.

FIGURE 2-1: PIC18F87J11 FAMILY CLOCK DIAGRAM



PIC18F87J11 FAMILY

2.2 Control Registers

The OSCCON register (Register 2-1) controls the main aspects of the device clock's operation. It selects the oscillator type to be used, which of the power-managed modes to invoke and the output frequency of the INTOSC source. It also provides status on the oscillators.

The OSCTUNE register (Register 2-2) controls the tuning and operation of the internal oscillator block. It also implements the PLLN bits which control the operation of the Phase Locked Loop (PLL) (see **Section 2.4.3 “PLL Frequency Multiplier”**).

REGISTER 2-1: OSCCON: OSCILLATOR CONTROL REGISTER⁽¹⁾

R/W-0	R/W-1	R/W-1	R/W-0	R ⁽²⁾	U-1	R/W-0	R/W-0
IDLEN	IRCF2 ⁽³⁾	IRCF1 ⁽³⁾	IRCF0 ⁽³⁾	OSTS	—	SCS1 ⁽⁵⁾	SCS0 ⁽⁵⁾
bit 7	bit 0						

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

bit 7

IDLEN: Idle Enable bit

- 1 = Device enters an Idle mode when a SLEEP instruction is executed
0 = Device enters Sleep mode when a SLEEP instruction is executed

bit 6-4

IRCF2:IRCF0: INTOSC Source Frequency Select bits⁽³⁾

- 111 = 8 MHz (INTOSC drives clock directly)
110 = 4 MHz (default)
101 = 2 MHz
100 = 1 MHz
011 = 500 kHz
010 = 250 kHz
001 = 125 kHz
000 = 31 kHz (from either INTOSC/256 or INTRC)⁽⁴⁾

bit 3

OSTS: Oscillator Start-up Timer Time-out Status bit⁽²⁾

- 1 = Oscillator Start-up Timer (OST) time-out has expired; primary oscillator is running
0 = Oscillator Start-up Timer (OST) time-out is running; primary oscillator is not ready

bit 2

Unimplemented: Read as ‘1’

bit 1-0

SCS1:SCS0: System Clock Select bits⁽⁵⁾

- 11 = Internal oscillator block
10 = Primary oscillator
01 = Timer1 oscillator
00 = Default primary oscillator (as defined by FOSC2:FOSC0 Configuration bits)

Note 1: Default (legacy) SFR at this address, available when WDTCON<4> = 0.

2: Reset state depends on state of the IESO Configuration bit.

3: Modifying these bits will cause an immediate clock frequency switch if the internal oscillator is providing the device clocks.

4: Source selected by the INTSRC bit (OSCTUNE<7>), see text.

5: Modifying these bits will cause an immediate clock source switch.

REGISTER 2-2: OSCTUNE: OSCILLATOR TUNING REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTSRC	PLLEN	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0
bit 7	bit 0						

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

bit 7	INTSRC: Internal Oscillator Low-Frequency Source Select bit 1 = 31.25 kHz device clock derived from 8 MHz INTOSC source (divide-by-256 enabled) 0 = 31 kHz device clock derived from INTRC 31 kHz oscillator
bit 6	PLLEN: Frequency Multiplier PLL Enable bit 1 = PLL enabled 0 = PLL disabled
bit 5-0	TUN5:TUN0: Fast RC Oscillator (INTOSC) Frequency Tuning bits 011111 = Maximum frequency • • 000001 000000 = Center frequency. Fast RC oscillator is running at the calibrated frequency. 111111 • • 100000 = Minimum frequency

2.3 Clock Sources and Oscillator Switching

Essentially, PIC18F87J11 Family devices have three independent clock sources:

- Primary oscillators
- Secondary oscillators
- Internal oscillator

The **primary oscillators** can be thought of as the main device oscillators. These are any external oscillators connected to the OSC1 and OSC2 pins, and include the External Crystal and Resonator modes and the External Clock modes. If selected by the FOSC2:FOSC0 Configuration bits, the internal oscillator block (either the 31 kHz INTRC or the 8 MHz INTOSC source) may be considered a primary oscillator. The particular mode is defined by the FOSC Configuration bits. The details of these modes are covered in **Section 2.4 “External Oscillator Modes”**.

The **secondary oscillators** are external clock sources that are not connected to the OSC1 or OSC2 pins. These sources may continue to operate even after the

controller is placed in a power-managed mode. PIC18F87J11 Family devices offer the Timer1 oscillator as a secondary oscillator source. This oscillator, in all power-managed modes, is often the time base for functions such as a Real-Time Clock (RTC). The Timer1 oscillator is discussed in greater detail in **Section 13.0 “Timer1 Module”**

In addition to being a primary clock source in some circumstances, the **internal oscillator** is available as a power-managed mode clock source. The INTRC source is also used as the clock source for several special features, such as the WDT and Fail-Safe Clock Monitor. The internal oscillator block is discussed in more detail in **Section 2.5 “Internal Oscillator Block”**.

The PIC18F87J11 Family includes features that allow the device clock source to be switched from the main oscillator, chosen by device configuration, to one of the alternate clock sources. When an alternate clock source is enabled, various power-managed operating modes are available.

PIC18F87J11 FAMILY

2.3.1 CLOCK SOURCE SELECTION

The System Clock Select bits, SCS1:SCS0 (OSCCON<1:0>), select the clock source. The available clock sources are the primary clock defined by the FOSC2:FOSC0 Configuration bits, the secondary clock (Timer1 oscillator) and the internal oscillator. The clock source changes after one or more of the bits are written to, following a brief clock transition interval.

The OSTS (OSCCON<3>) and T1RUN (T1CON<6>) bits indicate which clock source is currently providing the device clock. The OSTS bit indicates that the Oscillator Start-up Timer (OST) has timed out and the primary clock is providing the device clock in primary clock modes. The T1RUN bit indicates when the Timer1 oscillator is providing the device clock in secondary clock modes. In power-managed modes, only one of these bits will be set at any time. If neither of these bits is set, the INTRC is providing the clock, or the internal oscillator has just started and is not yet stable.

The IDLEN bit determines if the device goes into Sleep mode or one of the Idle modes when the SLEEP instruction is executed.

The use of the flag and control bits in the OSCCON register is discussed in more detail in **Section 3.0 “Power-Managed Modes”**.

- Note 1:** The Timer1 oscillator must be enabled to select the secondary clock source. The Timer1 oscillator is enabled by setting the T1OSCEN bit in the Timer1 Control register (T1CON<3>). If the Timer1 oscillator is not enabled, then any attempt to select a secondary clock source when executing a SLEEP instruction will be ignored.
- 2:** It is recommended that the Timer1 oscillator be operating and stable before executing the SLEEP instruction or a very long delay may occur while the Timer1 oscillator starts.

2.3.1.1 System Clock Selection and Device Resets

Since the SCS bits are cleared on all forms of Reset, this means the primary oscillator defined by the FOSC2:FOSC0 Configuration bits is used as the primary clock source on device Resets. This could either be the internal oscillator block by itself, or one of the other primary clock source (HS, EC, HSPLL, ECPLL1/2 or INTPLL1/2).

In those cases when the internal oscillator block, without PLL, is the default clock on Reset, the Fast RC oscillator (INTOSC) will be used as the device clock source. It will initially start at 4 MHz; the postscaler selection that corresponds to the Reset value of the IRCF2:IRCF0 bits ('110').

Regardless of which primary oscillator is selected, INTRC will always be enabled on device power-up. It serves as the clock source until the device has loaded its configuration values from memory. It is at this point that the FOSC Configuration bits are read and the oscillator selection of the operational mode is made.

Note that either the primary clock source, or the internal oscillator, will have two bit setting options for the possible values of the SCS1:SCS0 bits at any given time.

2.3.2 OSCILLATOR TRANSITIONS

PIC18F87J11 family devices contain circuitry to prevent clock “glitches” when switching between clock sources. A short pause in the device clock occurs during the clock switch. The length of this pause is the sum of two cycles of the old clock source and three to four cycles of the new clock source. This formula assumes that the new clock source is stable.

Clock transitions are discussed in greater detail in **Section 3.1.2 “Entering Power-Managed Modes”**.

2.4 External Oscillator Modes

2.4.1 CRYSTAL OSCILLATOR/CERAMIC RESONATORS (HS MODES)

In HS or HSPLL Oscillator modes, a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation. Figure 2-2 shows the pin connections.

The oscillator design requires the use of a crystal rated for parallel resonant operation.

Note: Use of a crystal rated for series resonant operation may give a frequency out of the crystal manufacturer's specifications.

TABLE 2-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Typical Capacitor Values Used:			
Mode	Freq.	OSC1	OSC2
HS	8.0 MHz	27 pF	27 pF
	16.0 MHz	22 pF	22 pF

Capacitor values are for design guidance only.

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application. Refer to the following application notes for oscillator specific information:

- AN588, "PIC® Microcontroller Oscillator Design Guide"
- AN826, "Crystal Oscillator Basics and Crystal Selection for rfPIC® and PIC® Devices"
- AN849, "Basic PIC® Oscillator Design"
- AN943, "Practical PIC® Oscillator Analysis and Design"
- AN949, "Making Your Oscillator Work"

See the notes following Table 2-2 for additional information.

TABLE 2-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Crystal Freq.	Typical Capacitor Values Tested:	
		C1	C2
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF

Capacitor values are for design guidance only.

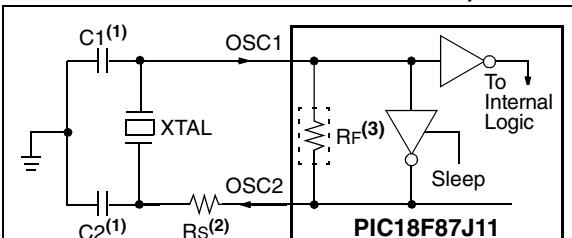
Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application.

Refer to the Microchip application notes cited in Table 2-1 for oscillator specific information. Also see the notes following this table for additional information.

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: Rs may be required to avoid overdriving crystals with low drive level specification.
- 4: Always verify oscillator performance over the VDD and temperature range that is expected for the application.

FIGURE 2-2: CRYSTAL/CERAMIC RESONATOR OPERATION (HS OR HSPLL CONFIGURATION)



Note 1: See Table 2-1 and Table 2-2 for initial values of C1 and C2.

Note 2: A series resistor (Rs) may be required for AT strip cut crystals.

Note 3: RF varies with the oscillator mode chosen.

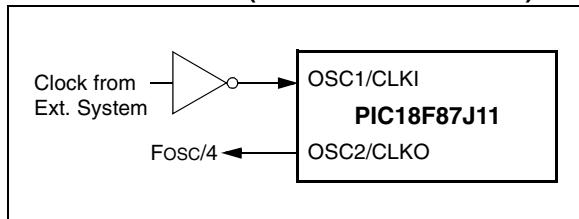
PIC18F87J11 FAMILY

2.4.2 EXTERNAL CLOCK INPUT (EC MODES)

The EC and ECPLL Oscillator modes require an external clock source to be connected to the OSC1 pin. There is no oscillator start-up time required after a Power-on Reset or after an exit from Sleep mode.

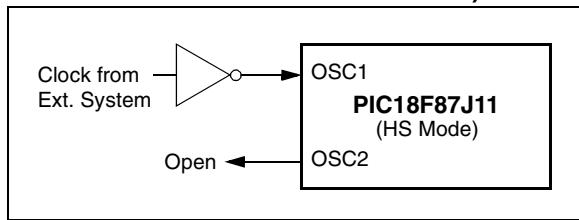
In the EC Oscillator mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 2-3 shows the pin connections for the EC Oscillator mode.

FIGURE 2-3: EXTERNAL CLOCK INPUT OPERATION (EC CONFIGURATION)



An external clock source may also be connected to the OSC1 pin in the HS mode, as shown in Figure 2-4. In this configuration, the divide-by-4 output on OSC2 is not available. Current consumption in this configuration will be somewhat higher than EC mode, as the internal oscillator's feedback circuitry will be enabled (in EC mode, the feedback circuit is disabled).

FIGURE 2-4: EXTERNAL CLOCK INPUT OPERATION (HS OSC CONFIGURATION)



2.4.3 PLL FREQUENCY MULTIPLIER

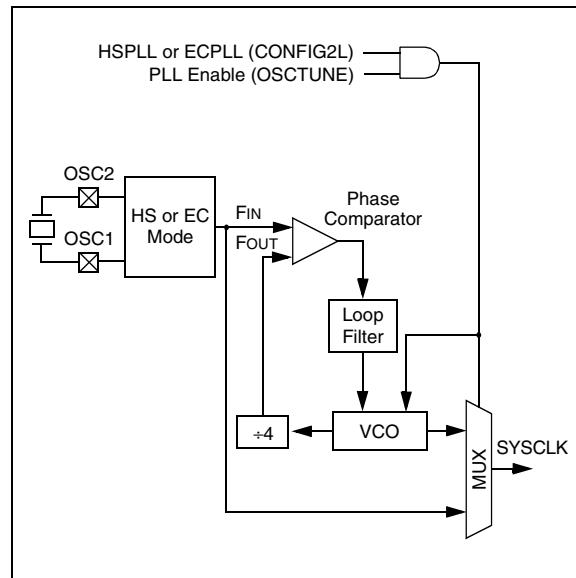
A Phase Locked Loop (PLL) circuit is provided as an option for users who want to use a lower frequency oscillator circuit, or to clock the device up to its highest rated frequency from a crystal oscillator. This may be useful for customers who are concerned with EMI due to high-frequency crystals, or users who require higher clock speeds from an internal oscillator.

2.4.3.1 HSPLL and ECPLL Modes

The HSPLL and ECPLL modes provide the ability to selectively run the device at 4 times the external oscillating source to produce frequencies up to 48 MHz.

The PLL is enabled by programming the FOSC2:FOSC0 Configuration bits to either '111' (for ECPLL) or '101' (for HSPLL). In addition, the PLL bit (OSCTUNE<6>) must also be set. Clearing PLL enables the PLL, regardless of the chosen oscillator configuration. It also allows additional flexibility for controlling the application's clock speed in software.

FIGURE 2-5: PLL BLOCK DIAGRAM



2.4.3.2 PLL and INTOSC

The PLL is also available to the internal oscillator block when the internal oscillator block is configured as the primary clock source. In this configuration, the PLL is enabled in software and generates a clock output of up to 32 MHz. The operation of INTOSC with the PLL is described in **Section 2.5.2 “INTPLL Modes”**.

2.5 Internal Oscillator Block

The PIC18F87J11 Family of devices includes an internal oscillator block which generates two different clock signals; either can be used as the microcontroller's clock source. This may eliminate the need for an external oscillator circuit on the OSC1 and/or OSC2 pins.

The main output is the Fast RC oscillator, or INTOSC, an 8 MHz clock source which can be used to directly drive the device clock. It also drives a postscaler, which can provide a range of clock frequencies from 31 kHz to 4 MHz. INTOSC is enabled when a clock frequency from 125 kHz to 8 MHz is selected. The INTOSC output can also be enabled when 31 kHz is selected, depending on the INTSRC bit (OSCTUNE<7>).

The other clock source is the internal RC oscillator (INTRC), which provides a nominal 31 kHz output. INTRC is enabled if it is selected as the device clock source; it is also enabled automatically when any of the following are enabled:

- Power-up Timer
- Fail-Safe Clock Monitor
- Watchdog Timer
- Two-Speed Start-up

These features are discussed in greater detail in **Section 24.0 "Special Features of the CPU"**.

The clock source frequency (INTOSC direct, INTOSC with postscaler or INTRC direct) is selected by configuring the IRCF bits of the OSCCON register. The default frequency on device Resets is 4 MHz.

2.5.1 INTIO MODES

Using the internal oscillator as the clock source eliminates the need for up to two external oscillator pins, which can then be used for digital I/O. Two distinct oscillator configurations, which are determined by the FOSC Configuration bits, are available:

- In INTIO1 mode, the OSC2 pin outputs Fosc/4, while OSC1 functions as RA7 for digital input and output. (see Figure 2-6) for digital input and output.
- In INTIO2 mode, OSC1 functions as RA7 and OSC2 functions as RA6, both for digital input and output. (see Figure 2-7), both for digital input and output.

FIGURE 2-6: INTIO1 OSCILLATOR MODE

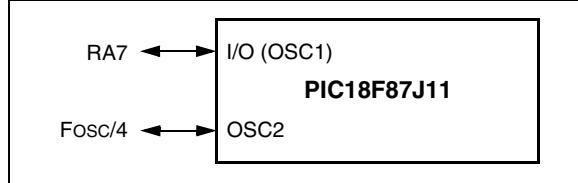


FIGURE 2-7: INTIO2 OSCILLATOR MODE



2.5.2 INTPLL MODES

The 4x Phase Locked Loop (PLL) can be used with the internal oscillator block to produce faster device clock speeds than are normally possible with the internal oscillator sources. When enabled, the PLL produces a clock speed of 16 MHz or 32 MHz.

PLL operation is controlled through software. The control bit, PLLEN (OSCTUNE<6>), is used to enable or disable its operation. The PLL is available only to INTOSC when the device is configured to use one of the INTPLL modes as the primary clock source (FOSC2:FOSC0 = 011 or 010). Additionally, the PLL will only function when the selected output frequency is either 4 MHz or 8 MHz (OSCCON<6:4> = 111 or 110).

Like the INTIO modes, there are two distinct INTPLL modes available:

- In INTPLL1 mode, the OSC2 pin outputs Fosc/4, while OSC1 functions as RA7 for digital input and output. Externally, this is identical in appearance to INTIO1 (Figure 2-6).
- In INTPLL2 mode, OSC1 functions as RA7 and OSC2 functions as RA6, both for digital input and output. Externally, this is identical to INTIO2 (Figure 2-7).

2.5.3 INTERNAL OSCILLATOR OUTPUT FREQUENCY AND TUNING

The internal oscillator block is calibrated at the factory to produce an INTOSC output frequency of 8 MHz. It can be adjusted in the user's application by writing to TUN5:TUN0 (OSCTUNE<5:0>) in the OSCTUNE register (Register 2-2).

When the OSCTUNE register is modified, the INTOSC frequency will begin shifting to the new frequency. The oscillator will stabilize within 1 ms. Code execution continues during this shift and there is no indication that the shift has occurred.

The INTRC oscillator operates independently of the INTOSC source. Any changes in INTOSC across voltage and temperature are not necessarily reflected by changes in INTRC or vice versa. The frequency of INTRC is not affected by OSCTUNE.

2.5.4 INTOSC FREQUENCY DRIFT

The INTOSC frequency may drift as VDD or temperature changes, and can affect the controller operation in a variety of ways. It is possible to adjust the INTOSC frequency by modifying the value in the OSCTUNE register. Depending on the device, this may have no effect on the INTRC clock source frequency.

Tuning INTOSC requires knowing when to make the adjustment, in which direction it should be made, and in some cases, how large a change is needed. Three compensation techniques are shown here.

2.5.4.1 Compensating with the EUSART

An adjustment may be required when the EUSART begins to generate framing errors or receives data with errors while in Asynchronous mode. Framing errors indicate that the device clock frequency is too high. To adjust for this, decrement the value in OSCTUNE to reduce the clock frequency. On the other hand, errors in data may suggest that the clock speed is too low. To compensate, increment OSCTUNE to increase the clock frequency.

2.5.4.2 Compensating with the Timers

This technique compares device clock speed to some reference clock. Two timers may be used; one timer is clocked by the peripheral clock, while the other is clocked by a fixed reference source, such as the Timer1 oscillator.

Both timers are cleared, but the timer clocked by the reference generates interrupts. When an interrupt occurs, the internally clocked timer is read and both timers are cleared. If the internally clocked timer value is much greater than expected, then the internal oscillator block is running too fast. To adjust for this, decrement the OSCTUNE register.

2.5.4.3 Compensating with the CCP Module in Capture Mode

A CCP module can use free-running Timer1 (or Timer3), clocked by the internal oscillator block and an external event with a known period (i.e., AC power frequency). The time of the first event is captured in the CCPRxH:CCPRxL registers and is recorded for use later. When the second event causes a capture, the time of the first event is subtracted from the time of the second event. Since the period of the external event is known, the time difference between events can be calculated.

If the measured time is much greater than the calculated time, the internal oscillator block is running too fast. To compensate, decrement the OSCTUNE register. If the measured time is much less than the calculated time, the internal oscillator block is running too slow. To compensate, increment the OSCTUNE register.

2.6 Reference Clock Output

In addition to the Fosc/4 clock output in certain oscillator modes, the device clock in the PIC18F87J11 family can also be configured to provide a reference clock output signal to a port pin. This feature is available in all oscillator configurations and allows the user to select a greater range of clock sub-multiples to drive external devices in the application.

This reference clock output is controlled by the REFOCON register (Register 2-3). Setting the ROON bit (REFOCON<7>) makes the clock signal available on the REFO (RE3) pin. The RODIV3:RODIV0 bits enable the selection of 16 different clock divider options.

The ROSSLP and ROSEL bits (REFOCON<5:4>) control the availability of the reference output during Sleep mode. The ROSEL bit determines if the oscillator on OSC1 and OSC2, or the current system clock source, is used for the reference clock output. The ROSSLP bit determines if the reference source is available on RE3 when the device is in Sleep mode.

To use the reference clock output in Sleep mode, both the ROSSLP and ROSEL bits must be set. The device clock must also be configured for an EC or HS mode; otherwise, the oscillator on OSC1 and OSC2 will be powered down when the device enters Sleep mode. Clearing the ROSEL bit allows the reference output frequency to change as the system clock changes during any clock switches.

The REFOCON register is an alternate SFR, and shares the same memory address as the OSCCON register. It is accessed by setting the ADSHR bit in the WDTCON register (WDTCON<4>).

REGISTER 2-3: REFOCON: REFERENCE OSCILLATOR CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROON	—	ROSSLP	ROSEL ⁽¹⁾	RODIV3	RODIV2	RODIV1	RODIV0
bit 7							bit 0

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

bit 7	ROON: Reference Oscillator Output Enable bit 1 = Reference oscillator output available on REFO pin 0 = Reference oscillator output disabled
bit 6	Unimplemented: Read as '0'
bit 5	ROSSLP: Reference Oscillator Output Stop in Sleep bit 1 = Reference oscillator continues to run in Sleep 0 = Reference oscillator is disabled in Sleep
bit 4	ROSEL: Reference Oscillator Source Select bit ⁽¹⁾ 1 = Primary oscillator (EC or HS) used as the base clock 0 = System clock used as the base clock; base clock reflects any clock switching of the device
bit 3-0	RODIV3:RODIV0: Reference Oscillator Divisor Select bits 1111 = Base clock value divided by 32,768 1110 = Base clock value divided by 16,384 1101 = Base clock value divided by 8,192 1100 = Base clock value divided by 4,096 1011 = Base clock value divided by 2,048 1010 = Base clock value divided by 1,024 1001 = Base clock value divided by 512 1000 = Base clock value divided by 256 0111 = Base clock value divided by 128 0110 = Base clock value divided by 64 0101 = Base clock value divided by 32 0100 = Base clock value divided by 16 0011 = Base clock value divided by 8 0010 = Base clock value divided by 4 0001 = Base clock value divided by 2 0000 = Base clock value

Note 1: If ROSEL = 1, an EC or HS oscillator must be configured as the default oscillator with the FOSC Configuration bits to maintain clock output during Sleep mode.

PIC18F87J11 FAMILY

2.7 Effects of Power-Managed Modes on the Various Clock Sources

When PRI_IDLE mode is selected, the designated primary oscillator continues to run without interruption. For all other power-managed modes, the oscillator using the OSC1 pin is disabled. The OSC1 pin (and OSC2 pin if used by the oscillator) will stop oscillating.

In secondary clock modes (SEC_RUN and SEC_IDLE), the Timer1 oscillator is operating and providing the device clock. The Timer1 oscillator may also run in all power-managed modes if required to clock Timer1 or Timer3.

In RC_RUN and RC_IDLE modes, the internal oscillator provides the device clock source. The 31 kHz INTRC output can be used directly to provide the clock and may be enabled to support various special features, regardless of the power-managed mode (see **Section 24.2 “Watchdog Timer (WDT)”** through **Section 24.5 “Fail-Safe Clock Monitor”** for more information on WDT, Fail-Safe Clock Monitor and Two-Speed Start-up).

If the Sleep mode is selected, all clock sources are stopped. Since all the transistor switching currents have been stopped, Sleep mode achieves the lowest current consumption of the device (only leakage currents).

Enabling any on-chip feature that will operate during Sleep will increase the current consumed during Sleep. The INTRC is required to support WDT operation. The

Timer1 oscillator may be operating to support a Real-Time Clock (RTC). Other features may be operating that do not require a device clock source (i.e., MSSP slave, PSP, INTx pins and others). Peripherals that may add significant current consumption are listed in **Section 27.2 “DC Characteristics: Power-Down and Supply Current”**.

2.8 Power-up Delays

Power-up delays are controlled by two timers, so that no external Reset circuitry is required for most applications. The delays ensure that the device is kept in Reset until the device power supply is stable under normal circumstances and the primary clock is operating and stable. For additional information on power-up delays, see **Section 4.6 “Power-up Timer (PWRT)”**.

The first timer is the Power-up Timer (PWRT), which provides a fixed delay on power-up (parameter 33, Table 27-12); it is always enabled.

The second timer is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable (HS modes). The OST does this by counting 1024 oscillator cycles before allowing the oscillator to clock the device.

There is a delay of interval Tcsd (parameter 38, Table 27-12), following POR, while the controller becomes ready to execute instructions.

TABLE 2-3: OSC1 AND OSC2 PIN STATES IN SLEEP MODE

Oscillator Mode	OSC1 Pin	OSC2 Pin
EC, ECPLL	Floating, pulled by external clock	At logic low (clock/4 output)
HS, HSPLL	Feedback inverter disabled at quiescent voltage level	Feedback inverter disabled at quiescent voltage level
INTOSC, INTPLL1/2	I/O pin RA6, direction controlled by TRISA<6>	I/O pin RA6, direction controlled by TRISA<7>

Note: See **Section 4.0 “Reset”** for time-outs due to Sleep and MCLR Reset.

3.0 POWER-MANAGED MODES

The PIC18F87J11 Family of devices provides the ability to manage power consumption by simply managing clocking to the CPU and the peripherals. In general, a lower clock frequency and a reduction in the number of circuits being clocked constitutes lower consumed power. For the sake of managing power in an application, there are three primary modes of operation:

- Run mode
- Idle mode
- Sleep mode

These modes define which portions of the device are clocked and at what speed. The Run and Idle modes may use any of the three available clock sources (primary, secondary or internal oscillator block); the Sleep mode does not use a clock source.

The power-managed modes include several power-saving features offered on previous devices. One is the clock switching feature, offered in other PIC18 devices, allowing the controller to use the Timer1 oscillator in place of the primary oscillator. Also included is the Sleep mode, offered by all PIC® devices, where all device clocks are stopped.

3.1 Selecting Power-Managed Modes

Selecting a power-managed mode requires two decisions: if the CPU is to be clocked or not and which clock source is to be used. The IDLEN bit (OSCCON<7>) controls CPU clocking, while the SCS1:SCS0 bits (OSCCON<1:0>) select the clock source. The individual modes, bit settings, clock sources and affected modules are summarized in Table 3-1.

TABLE 3-1: POWER-MANAGED MODES

Mode	OSCCON<7,1:0>		Module Clocking		Available Clock and Oscillator Source
	IDLEN ⁽¹⁾	SCS1:SCS0	CPU	Peripherals	
Sleep	0	N/A	Off	Off	None – All clocks are disabled
PRI_RUN	N/A	10	Clocked	Clocked	Primary – HS, EC, HSPLL, ECPLL, INTOSC oscillator; this is the normal, full-power execution mode
SEC_RUN	N/A	01	Clocked	Clocked	Secondary – Timer1 oscillator
RC_RUN	N/A	11	Clocked	Clocked	Internal oscillator block ⁽²⁾
PRI_IDLE	1	10	Off	Clocked	Primary – HS, EC, HSPLL, ECPLL, INTOSC
SEC_IDLE	1	01	Off	Clocked	Secondary – Timer1 oscillator
RC_IDLE	1	11	Off	Clocked	Internal oscillator block ⁽²⁾

Note 1: IDLEN reflects its value when the SLEEP instruction is executed.

2: Includes INTRC and INTOSC postcaler (internal oscillator block).

3.1.1 CLOCK SOURCES

The SCS1:SCS0 bits allow the selection of one of three clock sources for power-managed modes. They are:

- the primary clock, as defined by the FOSC2:FOSC0 Configuration bits
- the secondary clock (Timer1 oscillator)
- the internal oscillator

3.1.2 ENTERING POWER-MANAGED MODES

Switching from one power-managed mode to another begins by loading the OSCCON register. The SCS1:SCS0 bits select the clock source and determine which Run or Idle mode is to be used. Changing these bits causes an immediate switch to the new clock source, assuming that it is running. The switch may also be subject to clock transition delays. These are discussed in **Section 3.1.3 “Clock Transitions and Status Indicators”** and subsequent sections.

Entry to the power-managed Idle or Sleep modes is triggered by the execution of a SLEEP instruction. The actual mode that results depends on the status of the IDLEN bit.

Depending on the current mode and the mode being switched to, a change to a power-managed mode does not always require setting all of these bits. Many transitions may be done by changing the oscillator select bits, or changing the IDLEN bit, prior to issuing a SLEEP instruction. If the IDLEN bit is already configured correctly, it may only be necessary to perform a SLEEP instruction to switch to the desired mode.

3.1.3 CLOCK TRANSITIONS AND STATUS INDICATORS

The length of the transition between clock sources is the sum of two cycles of the old clock source and three to four cycles of the new clock source. This formula assumes that the new clock source is stable.

Two bits indicate the current clock source and its status: OSTS (OSCCON<3>) and T1RUN (T1CON<6>). In general, only one of these bits will be set while in a given power-managed mode. When the OSTS bit is set, the primary clock is providing the device clock. When the T1RUN bit is set, the Timer1 oscillator is providing the clock. If neither of these bits is set, INTRC is clocking the device.

Note: Executing a *SLEEP* instruction does not necessarily place the device into Sleep mode. It acts as the trigger to place the controller into either the Sleep mode, or one of the Idle modes, depending on the setting of the IDLEN bit.

3.1.4 MULTIPLE SLEEP COMMANDS

The power-managed mode that is invoked with the *SLEEP* instruction is determined by the setting of the IDLEN bit at the time the instruction is executed. If another *SLEEP* instruction is executed, the device will enter the power-managed mode specified by IDLEN at that time. If IDLEN has changed, the device will enter the new power-managed mode specified by the new setting.

3.2 Run Modes

In the Run modes, clocks to both the core and peripherals are active. The difference between these modes is the clock source.

3.2.1 PRI_RUN MODE

The PRI_RUN mode is the normal, full-power execution mode of the microcontroller. This is also the default mode upon a device Reset unless Two-Speed Start-up is enabled (see **Section 24.4 “Two-Speed Start-up”** for details). In this mode, the OSTS bit is set. (see **Section 2.2 “Control Registers”**).

3.2.2 SEC_RUN MODE

The SEC_RUN mode is the compatible mode to the “clock switching” feature offered in other PIC18 devices. In this mode, the CPU and peripherals are clocked from the Timer1 oscillator. This gives users the option of lower power consumption while still using a high-accuracy clock source.

SEC_RUN mode is entered by setting the SCS1:SCS0 bits to ‘01’. The device clock source is switched to the Timer1 oscillator (see Figure 3-1), the primary oscillator is shut down, the T1RUN bit (T1CON<6>) is set and the OSTS bit is cleared.

Note: The Timer1 oscillator should already be running prior to entering SEC_RUN mode. If the T1OSCEN bit is not set when the SCS1:SCS0 bits are set to '01', entry to SEC_RUN mode will not occur. If the Timer1 oscillator is enabled, but not yet running, device clocks will be delayed until the oscillator has started. In such situations, initial oscillator operation is far from stable and unpredictable operation may result.

On transitions from SEC_RUN mode to PRI_RUN mode, the peripherals and CPU continue to be clocked from the Timer1 oscillator while the primary clock is started. When the primary clock becomes ready, a clock switch back to the primary clock occurs (see Figure 3-2). When the clock switch is complete, the T1RUN bit is cleared, the OSTS bit is set and the primary clock is providing the clock. The IDLEN and SCS bits are not affected by the wake-up; the Timer1 oscillator continues to run.

FIGURE 3-1: TRANSITION TIMING FOR ENTRY TO SEC_RUN MODE

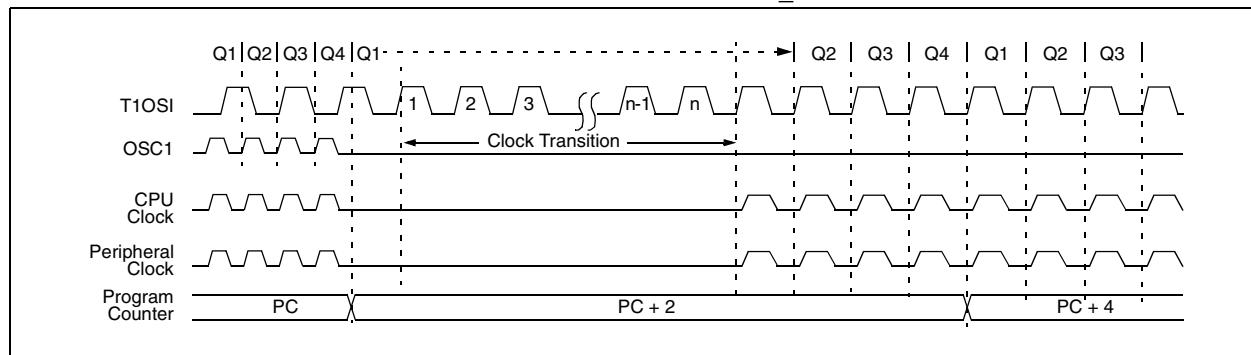
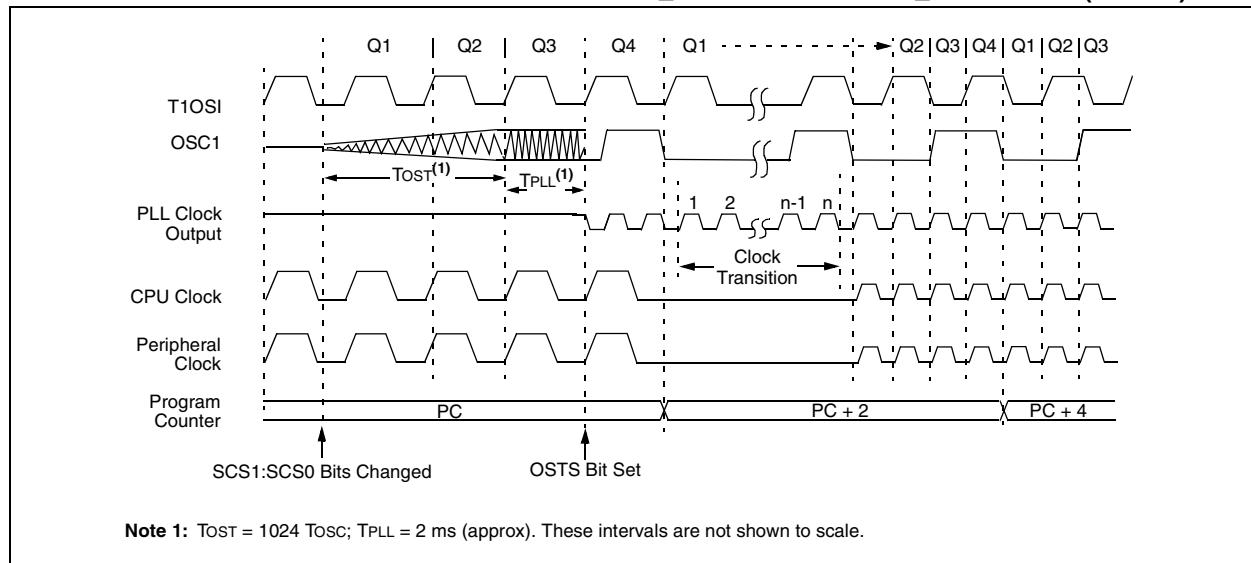


FIGURE 3-2: TRANSITION TIMING FROM SEC_RUN MODE TO PRI_RUN MODE (HSPLL)



PIC18F87J11 FAMILY

3.2.3 RC_RUN MODE

In RC_RUN mode, the CPU and peripherals are clocked from the internal oscillator; the primary clock is shut down. This mode provides the best power conservation of all the Run modes while still executing code. It works well for user applications which are not highly timing sensitive or do not require high-speed clocks at all times.

This mode is entered by setting SCS<1:0> to '11'. When the clock source is switched to the internal oscillator block (see Figure 3-3), the primary oscillator is shut down and the OSTST bit is cleared.

On transitions from RC_RUN mode to PRI_RUN mode, the device continues to be clocked from the INTOSC block while the primary clock is started. When the primary clock becomes ready, a clock switch to the primary clock occurs (see Figure 3-4). When the clock switch is complete, the OSTST bit is set and the primary clock is providing the device clock. The IDLEN and SCS bits are not affected by the switch. The INTRC block source will continue to run if either the WDT or the Fail-Safe Clock Monitor is enabled.

FIGURE 3-3: TRANSITION TIMING TO RC_RUN MODE

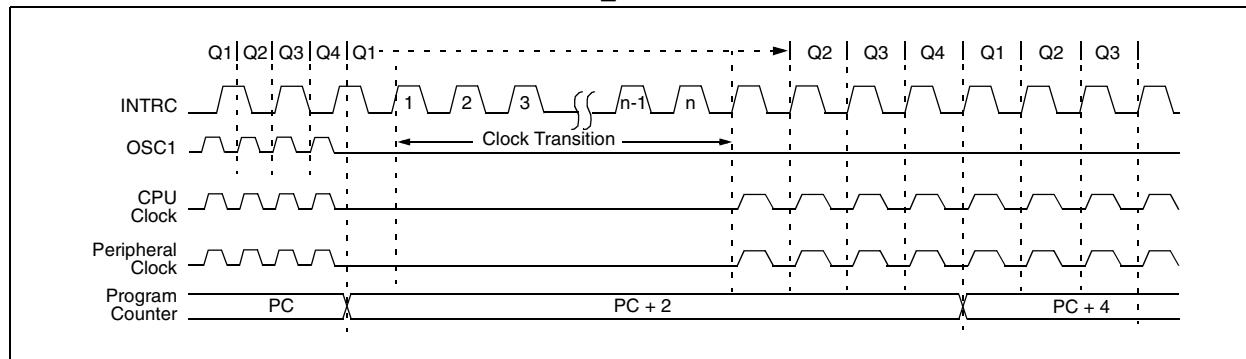
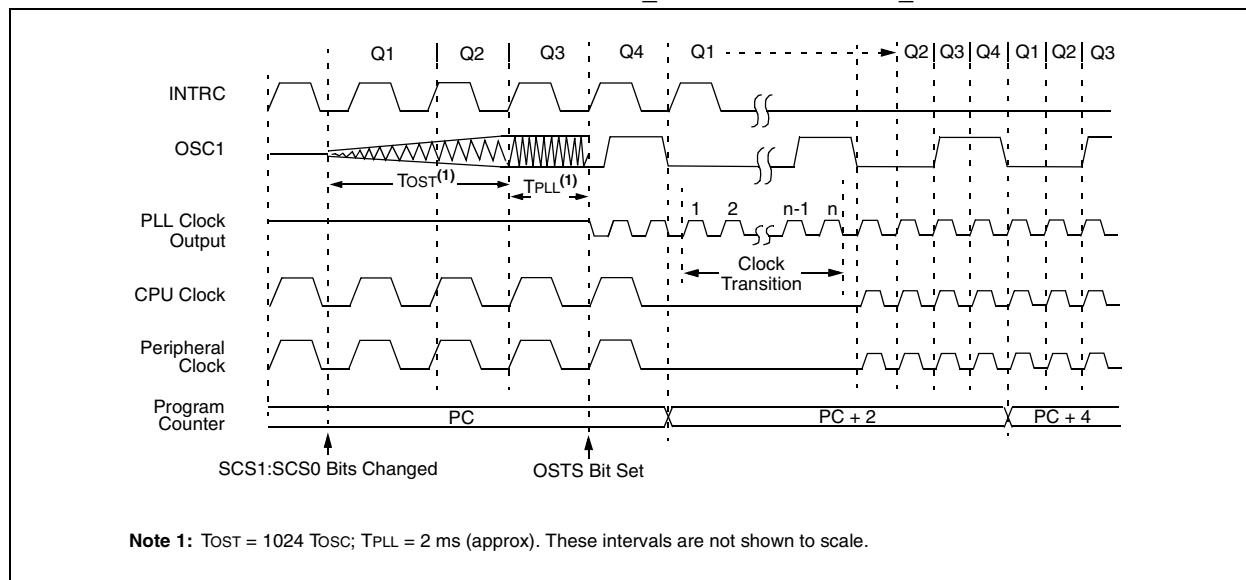


FIGURE 3-4: TRANSITION TIMING FROM RC_RUN MODE TO PRI_RUN MODE



3.3 Sleep Mode

The power-managed Sleep mode is identical to the legacy Sleep mode offered in all other PIC devices. It is entered by clearing the IDLEN bit (the default state on device Reset) and executing the *SLEEP* instruction. This shuts down the selected oscillator (Figure 3-5). All clock source status bits are cleared.

Entering the Sleep mode from any other mode does not require a clock switch. This is because no clocks are needed once the controller has entered Sleep. If the WDT is selected, the INTRC source will continue to operate. If the Timer1 oscillator is enabled, it will also continue to run.

When a wake event occurs in Sleep mode (by interrupt, Reset or WDT time-out), the device will not be clocked until the clock source selected by the SCS1:SCS0 bits becomes ready (see Figure 3-6), or it will be clocked from the internal oscillator if either the Two-Speed Start-up or the Fail-Safe Clock Monitor are enabled (see **Section 24.0 “Special Features of the CPU”**). In either case, the OSTST bit is set when the primary clock is providing the device clocks. The IDLEN and SCS bits are not affected by the wake-up.

3.4 Idle Modes

The Idle modes allow the controller's CPU to be selectively shut down while the peripherals continue to operate. Selecting a particular Idle mode allows users to further manage power consumption.

If the IDLEN bit is set to ‘1’ when a *SLEEP* instruction is executed, the peripherals will be clocked from the clock source selected using the SCS1:SCS0 bits; however, the CPU will not be clocked. The clock source status bits are not affected. Setting IDLEN and executing a *SLEEP* instruction provides a quick method of switching from a given Run mode to its corresponding Idle mode.

If the WDT is selected, the INTRC source will continue to operate. If the Timer1 oscillator is enabled, it will also continue to run.

Since the CPU is not executing instructions, the only exits from any of the Idle modes are by interrupt, WDT time-out or a Reset. When a wake event occurs, CPU execution is delayed by an interval of T_{CSD} (parameter 38, Table 27-12) while it becomes ready to execute code. When the CPU begins executing code, it resumes with the same clock source for the current Idle mode. For example, when waking from RC_IDLE mode, the internal oscillator block will clock the CPU and peripherals (in other words, RC_RUN mode). The IDLEN and SCS bits are not affected by the wake-up.

While in any Idle mode or the Sleep mode, a WDT time-out will result in a WDT wake-up to the Run mode currently specified by the SCS1:SCS0 bits.

FIGURE 3-5: TRANSITION TIMING FOR ENTRY TO SLEEP MODE

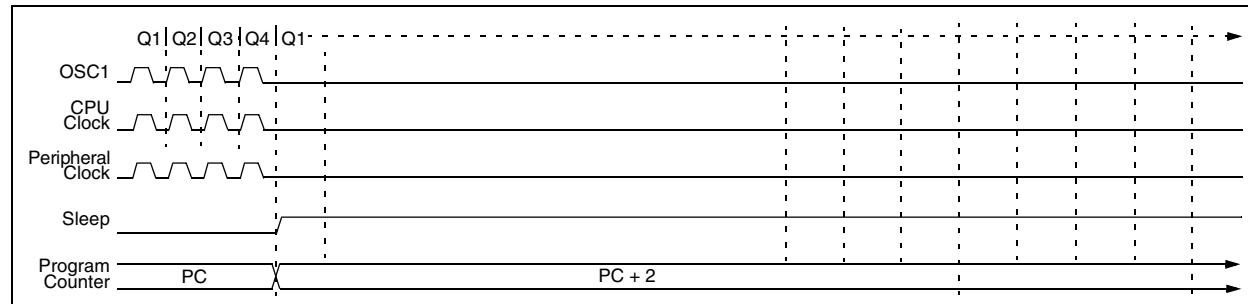
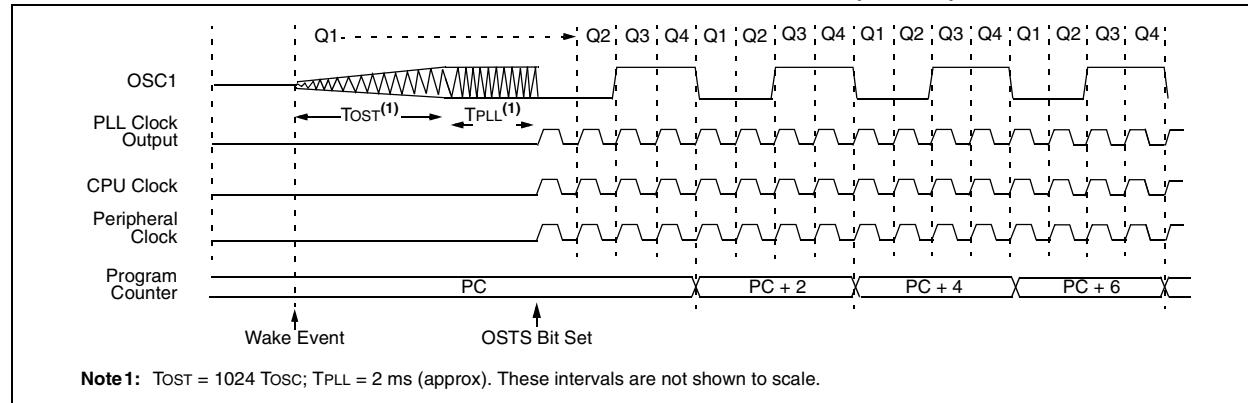


FIGURE 3-6: TRANSITION TIMING FOR WAKE FROM SLEEP (HSPLL)



PIC18F87J11 FAMILY

3.4.1 PRI_IDLE MODE

This mode is unique among the three low-power Idle modes, in that it does not disable the primary device clock. For timing sensitive applications, this allows for the fastest resumption of device operation with its more accurate primary clock source, since the clock source does not have to "warm up" or transition from another oscillator.

PRI_IDLE mode is entered from PRI_RUN mode by setting the IDLEN bit and executing a SLEEP instruction. If the device is in another Run mode, set IDLEN first, then set the SCS bits to '10' and execute SLEEP. Although the CPU is disabled, the peripherals continue to be clocked from the primary clock source specified by the FOSC1:FOSC0 Configuration bits. The OSTS bit remains set (see Figure 3-7).

When a wake event occurs, the CPU is clocked from the primary clock source. A delay of interval TcSD is required between the wake event and when code execution starts. This is required to allow the CPU to become ready to execute instructions. After the wake-up, the OSTS bit remains set. The IDLEN and SCS bits are not affected by the wake-up (see Figure 3-8).

3.4.2 SEC_IDLE MODE

In SEC_IDLE mode, the CPU is disabled but the peripherals continue to be clocked from the Timer1 oscillator. This mode is entered from SEC_RUN by setting the IDLEN bit and executing a SLEEP instruction. If the device is in another Run mode, set IDLEN first, then set SCS1:SCS0 to '01' and execute SLEEP. When the clock source is switched to the Timer1 oscillator, the primary oscillator is shut down, the OSTS bit is cleared and the T1RUN bit is set.

When a wake event occurs, the peripherals continue to be clocked from the Timer1 oscillator. After an interval of TcSD following the wake event, the CPU begins executing code being clocked by the Timer1 oscillator. The IDLEN and SCS bits are not affected by the wake-up; the Timer1 oscillator continues to run (see Figure 3-8).

Note: The Timer1 oscillator should already be running prior to entering SEC_IDLE mode. If the T1OSCEN bit is not set when the SLEEP instruction is executed, the SLEEP instruction will be ignored and entry to SEC_IDLE mode will not occur. If the Timer1 oscillator is enabled, but not yet running, peripheral clocks will be delayed until the oscillator has started. In such situations, initial oscillator operation is far from stable and unpredictable operation may result.

FIGURE 3-7: TRANSITION TIMING FOR ENTRY TO IDLE MODE

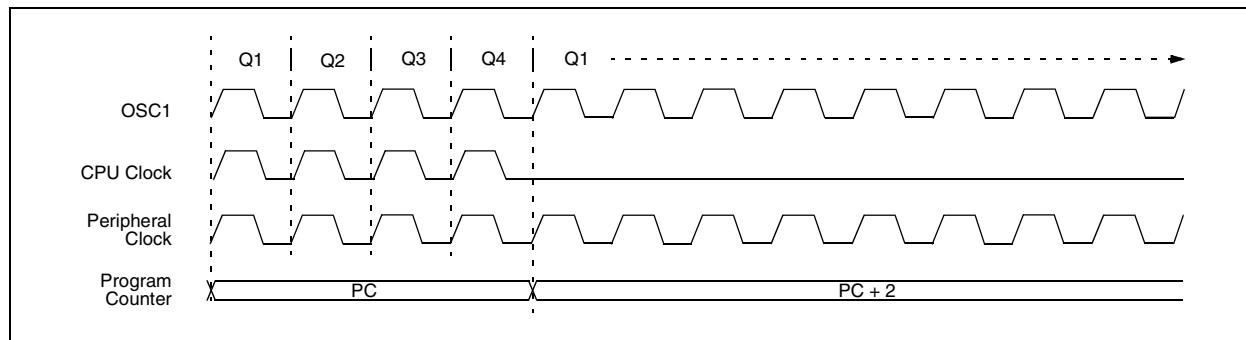
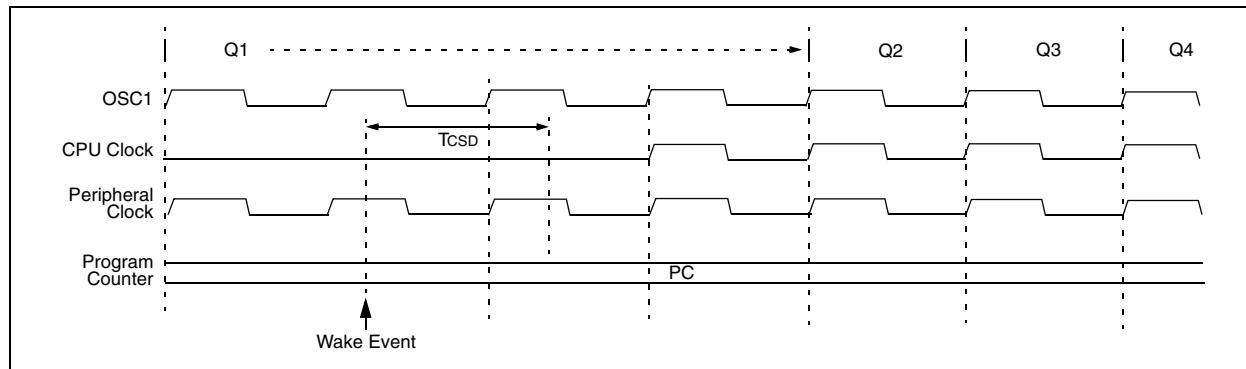


FIGURE 3-8: TRANSITION TIMING FOR WAKE FROM IDLE TO RUN MODE



3.4.3 RC_IDLE MODE

In RC_IDLE mode, the CPU is disabled but the peripherals continue to be clocked from the internal oscillator block. This mode allows for controllable power conservation during Idle periods.

From RC_RUN, this mode is entered by setting the IDLEN bit and executing a SLEEP instruction. If the device is in another Run mode, first set IDLEN, then clear the SCS bits and execute SLEEP. When the clock source is switched to the INTOSC block, the primary oscillator is shut down and the OSTS bit is cleared.

When a wake event occurs, the peripherals continue to be clocked from the internal oscillator block. After a delay of Tcsd following the wake event, the CPU begins executing code being clocked by the INTRC. The IDLEN and SCS bits are not affected by the wake-up. The INTRC source will continue to run if either the WDT or the Fail-Safe Clock Monitor is enabled.

3.5 Exiting Idle and Sleep Modes

An exit from Sleep mode, or any of the Idle modes, is triggered by an interrupt, a Reset or a WDT time-out. This section discusses the triggers that cause exits from power-managed modes. The clocking subsystem actions are discussed in each of the power-managed modes sections (see **Section 3.2 “Run Modes”**, **Section 3.3 “Sleep Mode”** and **Section 3.4 “Idle Modes”**).

3.5.1 EXIT BY INTERRUPT

Any of the available interrupt sources can cause the device to exit from an Idle mode, or the Sleep mode, to a Run mode. To enable this functionality, an interrupt source must be enabled by setting its enable bit in one of the INTCON or PIE registers. The exit sequence is initiated when the corresponding interrupt flag bit is set.

On all exits from Idle or Sleep modes by interrupt, code execution branches to the interrupt vector if the GIE/GIEH bit (INTCON<7>) is set. Otherwise, code execution continues or resumes without branching (see **Section 9.0 “Interrupts”**).

A fixed delay of interval, Tcsd, following the wake event is required when leaving Sleep and Idle modes. This delay is required for the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

3.5.2 EXIT BY WDT TIME-OUT

A WDT time-out will cause different actions depending on which power-managed mode the device is in when the time-out occurs.

If the device is not executing code (all Idle modes and Sleep mode), the time-out will result in an exit from the power-managed mode (see **Section 3.2 “Run Modes”** and **Section 3.3 “Sleep Mode”**). If the device is executing code (all Run modes), the time-out will result in a WDT Reset (see **Section 24.2 “Watchdog Timer (WDT)”**).

The Watchdog Timer and postscaler are cleared by one of the following events:

- Executing a SLEEP or CLRWDT instruction
- The loss of a currently selected clock source (if the Fail-Safe Clock Monitor is enabled)

3.5.3 EXIT BY RESET

Exiting an Idle or Sleep mode by Reset automatically forces the device to run from the INTRC.

3.5.4 EXIT WITHOUT AN OSCILLATOR START-UP DELAY

Certain exits from power-managed modes do not invoke the OST at all. There are two cases:

- PRI_IDLE mode, where the primary clock source is not stopped; and
- The primary clock source is either the EC or ECPLL mode.

In these instances, the primary clock source either does not require an oscillator start-up delay, since it is already running (PRI_IDLE), or normally does not require an oscillator start-up delay (EC). However, a fixed delay of interval, Tcsd, following the wake event is still required when leaving Sleep and Idle modes to allow the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

PIC18F87J11 FAMILY

NOTES:

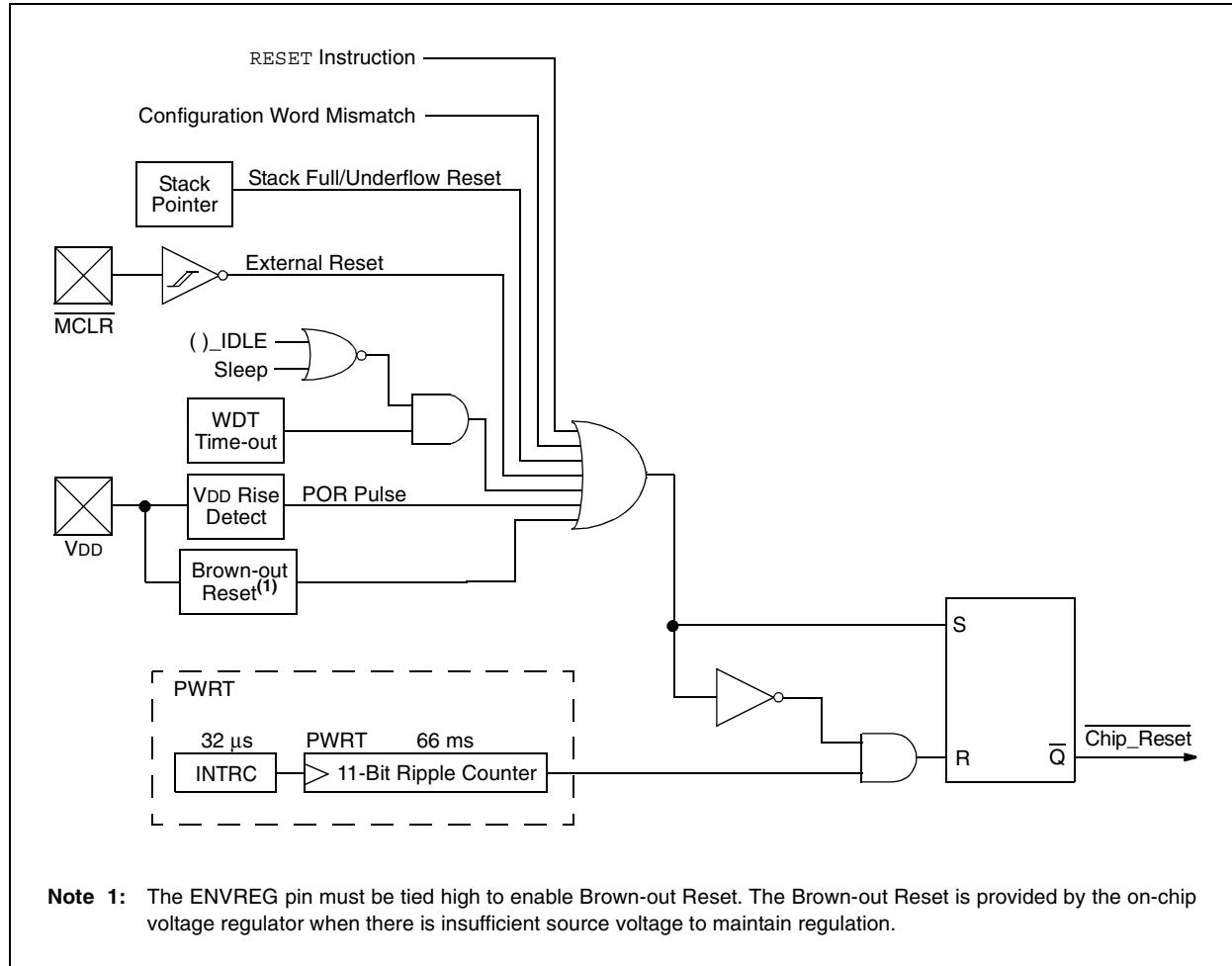
4.0 RESET

The PIC18F87J11 Family of devices differentiate between various kinds of Reset:

- a) Power-on Reset (POR)
- b) MCLR Reset during normal operation
- c) MCLR Reset during power-managed modes
- d) Watchdog Timer (WDT) Reset (during execution)
- e) Configuration Mismatch (CM)
- f) Brown-out Reset (BOR)
- g) RESET Instruction
- h) Stack Full Reset
- i) Stack Underflow Reset

This section discusses Resets generated by MCLR, POR and BOR and covers the operation of the various start-up timers. Stack Reset events are covered in **Section 5.1.6.4 “Stack Full and Underflow Resets”**. WDT Resets are covered in **Section 24.2 “Watchdog Timer (WDT)”**.

FIGURE 4-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



A simplified block diagram of the on-chip Reset circuit is shown in Figure 4-1.

4.1 RCON Register

Device Reset events are tracked through the RCON register (Register 4-1). The lower five bits of the register indicate that a specific Reset event has occurred. In most cases, these bits can only be set by the event and must be cleared by the application after the event. The state of these flag bits, taken together, can be read to indicate the type of Reset that just occurred. This is described in more detail in **Section 4.7 “Reset State of Registers”**.

The RCON register also has a control bit for setting interrupt priority (IPEN). Interrupt priority is discussed in **Section 9.0 “Interrupts”**.

PIC18F87J11 FAMILY

REGISTER 4-1: RCON: RESET CONTROL REGISTER

R/W-0	U-0	R/W-1	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	<u>CM</u>	<u>RI</u>	<u>TO</u>	<u>PD</u>	<u>POR</u>	<u>BOR</u>
bit 7							bit 0

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

bit 7	IPEN: Interrupt Priority Enable bit 1 = Enable priority levels on interrupts 0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)
bit 6	Unimplemented: Read as '0'
bit 5	CM: Configuration Mismatch Flag bit 1 = A Configuration Mismatch Reset has not occurred 0 = A Configuration Mismatch Reset has occurred (must be set in software after a Configuration Mismatch Reset occurs)
bit 4	RI: RESET Instruction Flag bit 1 = The RESET instruction was not executed (set by firmware only) 0 = The RESET instruction was executed causing a device Reset (must be set in software after a Brown-out Reset occurs)
bit 3	TO: Watchdog Time-out Flag bit 1 = Set by power-up, CLRWDT instruction or SLEEP instruction 0 = A WDT time-out occurred
bit 2	PD: Power-Down Detection Flag bit 1 = Set by power-up or by the CLRWDT instruction 0 = Set by execution of the SLEEP instruction
bit 1	POR: Power-on Reset Status bit 1 = A Power-on Reset has not occurred (set by firmware only) 0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)
bit 0	BOR: Brown-out Reset Status bit 1 = A Brown-out Reset has not occurred (set by firmware only) 0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

- Note 1:** It is recommended that the POR bit be set after a Power-on Reset has been detected, so that subsequent Power-on Resets may be detected.
- 2:** If the on-chip voltage regulator is disabled, BOR remains '0' at all times. See **Section 4.4.1 "Detecting BOR"** for more information.
- 3:** Brown-out Reset is said to have occurred when BOR is '0' and POR is '1' (assuming that POR was set to '1' by software immediately after a Power-on Reset).

4.2 Master Clear (MCLR)

The MCLR pin provides a method for triggering a hard external Reset of the device. A Reset is generated by holding the pin low. PIC18 extended microcontroller devices have a noise filter in the MCLR Reset path which detects and ignores small pulses.

The MCLR pin is not driven low by any internal Resets, including the WDT.

4.3 Power-on Reset (POR)

A Power-on Reset condition is generated on-chip whenever VDD rises above a certain threshold. This allows the device to start in the initialized state when VDD is adequate for operation.

To take advantage of the POR circuitry, tie the MCLR pin through a resistor (1 k Ω to 10 k Ω) to VDD. This will eliminate external RC components usually needed to create a Power-on Reset delay. A minimum rise rate for VDD is specified (parameter D004). For a slow rise time, see Figure 4-2.

When the device starts normal operation (i.e., 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.

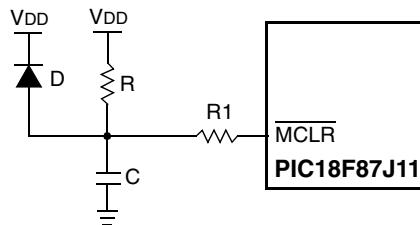
Power-on Reset events are captured by the POR bit (RCON<1>). The state of the bit is set to '0' whenever a Power-on Reset occurs; it does not change for any other Reset event. POR is not reset to '1' by any hardware event. To capture multiple events, the user manually resets the bit to '1' in software following any Power-on Reset.

4.4 Brown-out Reset (BOR)

The PIC18F87J11 family of devices incorporates a simple Brown-out Reset function when the internal regulator is enabled (ENVREG pin is tied to VDD). Any drop of VDD below VBOR (parameter D005) for greater than time TBOR (parameter 35) will reset the device. A Reset may or may not occur if VDD falls below VBOR for less than TBOR. The chip will remain in Brown-out Reset until VDD rises above VBOR.

Once a Brown-out Reset has occurred, the Power-up Timer will keep the chip in Reset for TPWRT (parameter 33). If VDD drops below VBOR while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be initialized. Once VDD rises above VBOR, the Power-up Timer will execute the additional time delay.

FIGURE 4-2: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



- Note 1:** External Power-on Reset circuit is required only if the VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.
- 2:** R < 40 k Ω is recommended to make sure that the voltage drop across R does not violate the device's electrical specification.
- 3:** R1 \geq 1 k Ω will limit any current flowing into MCLR from external capacitor C, in the event of MCLR/VPP pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

4.4.1 DETECTING BOR

The BOR bit always resets to '0' on any Brown-out Reset or Power-on Reset event. This makes it difficult to determine if a Brown-out Reset event has occurred just by reading the state of BOR alone. A more reliable method is to simultaneously check the state of both POR and BOR. This assumes that the POR bit is reset to '1' in software immediately after any Power-on Reset event. If BOR is '0' while POR is '1', it can be reliably assumed that a Brown-out Reset event has occurred.

If the voltage regulator is disabled, Brown-out Reset functionality is disabled. In this case, the BOR bit cannot be used to determine a Brown-out Reset event. The BOR bit is still cleared by a Power-on Reset event.

4.5 Configuration Mismatch (CM)

The Configuration Mismatch (CM) Reset is designed to detect and attempt to recover from random, memory corrupting events. These include Electrostatic Discharge (ESD) events, which can cause widespread, single-bit changes throughout the device and result in catastrophic failure.

In PIC18FXXJ Flash devices, the device Configuration registers (located in the configuration memory space) are continuously monitored during operation by comparing their values to complimentary shadow registers. If a mismatch is detected between the two sets of registers, a CM Reset automatically occurs. These events are captured by the CM bit (RCON<5>). The state of the bit is set to '0' whenever a CM event occurs; it does not change for any other Reset event.

PIC18F87J11 FAMILY

A CM Reset behaves similarly to a Master Clear Reset, RESET instruction, WDT time-out or Stack Event Resets. As with all hard and power Reset events, the device Configuration Words are reloaded from the Flash Configuration Words in program memory as the device restarts.

4.6 Power-up Timer (PWRT)

PIC18F87J11 Family devices incorporate an on-chip Power-up Timer (PWRT) to help regulate the Power-on Reset process. The PWRT is always enabled. The main function is to ensure that the device voltage is stable before code is executed.

The Power-up Timer (PWRT) of the PIC18F87J11 Family devices is an 11-bit counter which uses the INTRC source as the clock input. This yields an approximate time interval of $2048 \times 32 \mu\text{s} = 66 \text{ ms}$. While the PWRT is counting, the device is held in Reset.

The power-up time delay depends on the INTRC clock and will vary from chip-to-chip due to temperature and process variation. See DC parameter 33 for details.

4.6.1 TIME-OUT SEQUENCE

If enabled, the PWRT time-out is invoked after the POR pulse has cleared. The total time-out will vary based on the status of the PWRT. Figure 4-3, Figure 4-4, Figure 4-5 and Figure 4-6 all depict time-out sequences on power-up with the Power-up Timer enabled.

Since the time-outs occur from the POR pulse, if MCLR is kept low long enough, the PWRT will expire. Bringing MCLR high will begin execution immediately (Figure 4-5). This is useful for testing purposes, or to synchronize more than one PIC18FXXXX device operating in parallel.

FIGURE 4-3: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD, VDD RISE < TPWRT)

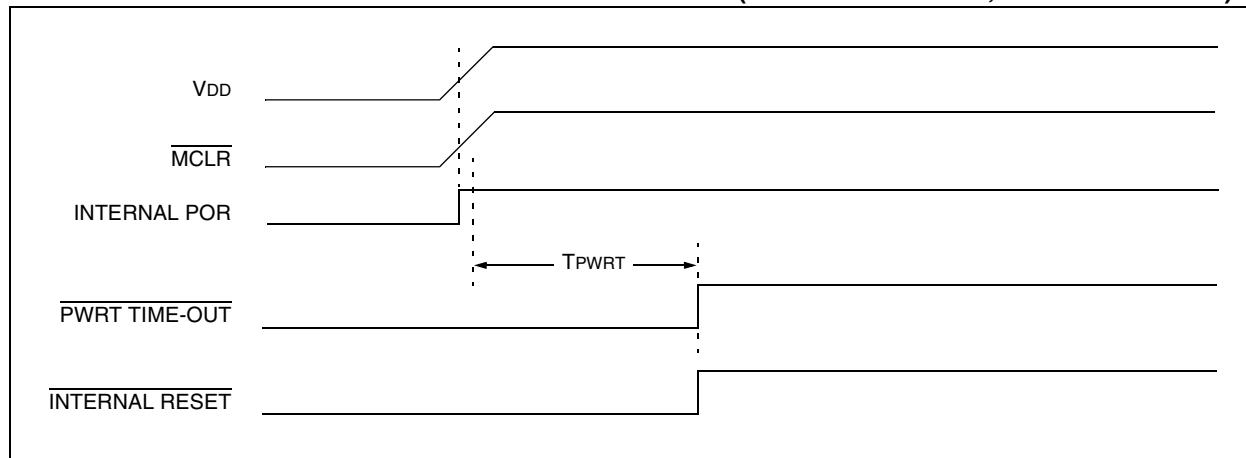
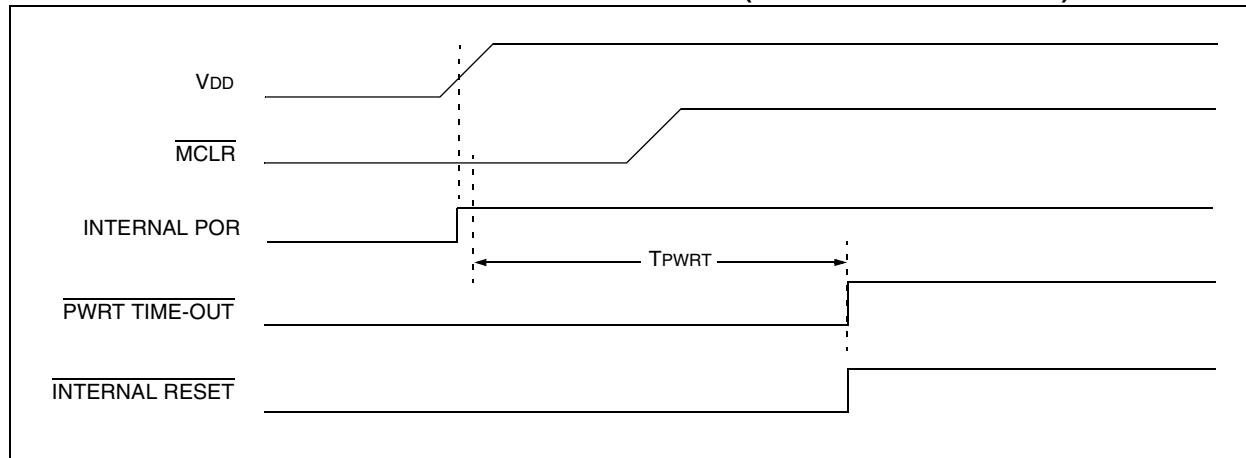


FIGURE 4-4: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 1



PIC18F87J11 FAMILY

FIGURE 4-5: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 2

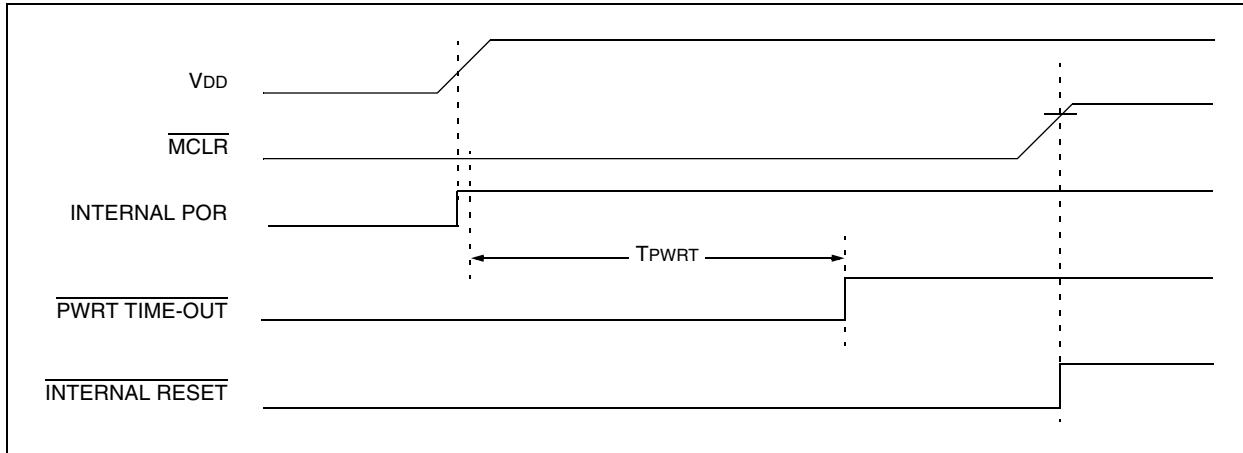
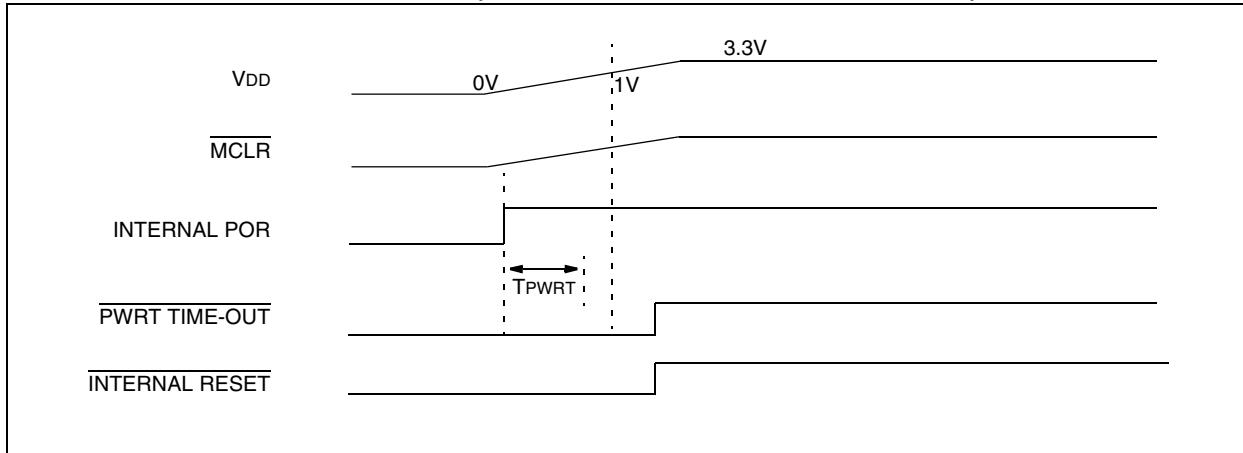


FIGURE 4-6: SLOW RISE TIME (MCLR TIED TO VDD, VDD RISE > TPWRT)



PIC18F87J11 FAMILY

4.7 Reset State of Registers

Most registers are unaffected by a Reset. Their status is unknown on POR and unchanged by all other Resets. The other registers are forced to a "Reset state" depending on the type of Reset that occurred.

Most registers are not affected by a WDT wake-up, since this is viewed as the resumption of normal operation. Status bits from the RCON register (CM, RI, TO, PD, POR and BOR) are set or cleared differently in

different Reset situations, as indicated in Table 4-1. These bits are used in software to determine the nature of the Reset.

Table 4-2 describes the Reset states for all of the Special Function Registers. These are categorized by Power-on and Brown-out Resets, Master Clear and WDT Resets and WDT wake-ups.

TABLE 4-1: STATUS BITS, THEIR SIGNIFICANCE AND THE INITIALIZATION CONDITION FOR RCON REGISTER

Condition	Program Counter ⁽¹⁾	RCON Register						STKPTR Register	
		CM	RI	TO	PD	POR	BOR	STKFUL	STKUNF
Power-on Reset	0000h	1	1	1	1	0	0	0	0
RESET instruction	0000h	u	0	u	u	u	u	u	u
Brown-out Reset	0000h	1	1	1	1	u	0	u	u
Configuration Mismatch Reset	0000h	0	u	u	u	u	u	u	u
MCLR Reset during power-managed Run modes	0000h	u	u	1	u	u	u	u	u
MCLR Reset during power-managed Idle modes and Sleep mode	0000h	u	u	1	0	u	u	u	u
MCLR Reset during full-power execution	0000h	u	u	u	u	u	u	u	u
Stack Full Reset (STVREN = 1)	0000h	u	u	u	u	u	u	1	u
Stack Underflow Reset (STVREN = 1)	0000h	u	u	u	u	u	u	u	1
Stack Underflow Error (not an actual Reset, STVREN = 0)	0000h	u	u	u	u	u	u	u	1
WDT time-out during full-power or power-managed Run modes	0000h	u	u	0	u	u	u	u	u
WDT time-out during power-managed Idle or Sleep modes	PC + 2	u	u	0	0	u	u	u	u
Interrupt exit from power-managed modes	PC + 2	u	u	u	0	u	u	u	u

Legend: u = unchanged

Note 1: When the wake-up is due to an interrupt and the GIEH or GIEL bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

PIC18F87J11 FAMILY

TABLE 4-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS

Register	Applicable Devices	Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets, CM Resets	Wake-up via WDT or Interrupt
TOSU	PIC18F6XJ1X	PIC18F8XJ1X	---0 0000	---0 0000 ---0 _{uuuu} ⁽¹⁾
TOSH	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000 _{uuuu uuuu} ⁽¹⁾
TOSL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000 _{uuuu uuuu} ⁽¹⁾
STKPTR	PIC18F6XJ1X	PIC18F8XJ1X	00-0 0000	uu-0 0000 _{uu-u uuuu} ⁽¹⁾
PCLATU	PIC18F6XJ1X	PIC18F8XJ1X	---0 0000	---0 0000 ---u _{uuuu}
PCLATH	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000 _{uuuu uuuu}
PCL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000 PC + 2 ⁽²⁾
TBLPTRU	PIC18F6XJ1X	PIC18F8XJ1X	--00 0000	--00 0000 --uu _{uuuu}
TBLPTRH	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000 _{uuuu uuuu}
TBLPTRL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000 _{uuuu uuuu}
TABLAT	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000 _{uuuu uuuu}
PRODH	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu _{uuuu} _{uuuu uuuu}
PRODL	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu _{uuuu} _{uuuu uuuu}
INTCON	PIC18F6XJ1X	PIC18F8XJ1X	0000 000x	0000 000u _{uuuu uuuu} ⁽³⁾
INTCON2	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111 _{uuuu uuuu} ⁽³⁾
INTCON3	PIC18F6XJ1X	PIC18F8XJ1X	1100 0000	1100 0000 _{uuuu uuuu} ⁽³⁾
INDF0	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A N/A
POSTINC0	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A N/A
POSTDEC0	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A N/A
PREINC0	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A N/A
PLUSW0	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A N/A
FSR0H	PIC18F6XJ1X	PIC18F8XJ1X	---- xxxx	---- _{uuuu} _{uuuu}
FSR0L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	_{uuuu} _{uuuu} _{uuuu uuuu}
WREG	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	_{uuuu} _{uuuu} _{uuuu uuuu}
INDF1	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A N/A
POSTINC1	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A N/A
POSTDEC1	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A N/A
PREINC1	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A N/A
PLUSW1	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A N/A
FSR1H	PIC18F6XJ1X	PIC18F8XJ1X	---- xxxx	---- _{uuuu} _{uuuu}
FSR1L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	_{uuuu} _{uuuu} _{uuuu uuuu}
BSR	PIC18F6XJ1X	PIC18F8XJ1X	---- 0000	---- 0000 _{uuuu}

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

Shaded cells indicate conditions do not apply for the designated device.

- Note 1:** When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.
- 2:** When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).
- 3:** One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).
- 4:** See Table 4-1 for Reset value for specific condition.

PIC18F87J11 FAMILY

TABLE 4-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets, CM Resets	Wake-up via WDT or Interrupt
INDF2	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
POSTINC2	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
POSTDEC2	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
PREINC2	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
PLUSW2	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
FSR2H	PIC18F6XJ1X	PIC18F8XJ1X	---- xxxx	---- uuuu	---- uuuu
FSR2L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	PIC18F6XJ1X	PIC18F8XJ1X	--x xxxx	--u uuuu	--u uuuu
TMR0H	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
TMR0L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
OSCCON	PIC18F6XJ1X	PIC18F8XJ1X	0110 q100	0110 q100	0110 q10u
REFOCON	PIC18F6XJ1X	PIC18F8XJ1X	0-00 0000	u-uu uuuu	u-uu uuuu
CM1CON	PIC18F6XJ1X	PIC18F8XJ1X	0001 1111	uuuu uuuu	uuuu uuuu
CM2CON	PIC18F6XJ1X	PIC18F8XJ1X	0001 1111	uuuu uuuu	uuuu uuuu
RCON ⁽⁴⁾	PIC18F6XJ1X	PIC18F8XJ1X	0-11 1100	0-qq qquu	u-qq qquu
TMR1H	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
ODCON1	PIC18F6XJ1X	PIC18F8XJ1X	--0 0000	--u uuuu	--u uuuu
TMR1L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
ODCON2	PIC18F6XJ1X	PIC18F8XJ1X	---- --00	---- --uu	---- --uu
T1CON	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	u0uu uuuu	uuuu uuuu
ODCON3	PIC18F6XJ1X	PIC18F8XJ1X	---- --00	---- --uu	---- --uu
TMR2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PADCFG1	PIC18F6XJ1X	PIC18F8XJ1X	---- ---0	---- ---u	---- ---u
PR2	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	1111 1111
MEMCON	PIC18F6XJ1X	PIC18F8XJ1X	0-00 --00	0-00 --00	u-uu --uu
T2CON	PIC18F6XJ1X	PIC18F8XJ1X	-000 0000	-000 0000	-uuu uuuu
SSP1BUF	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSP1ADD	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
SSP1MSK	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	uuuu uuuu	uuuu uuuu
SSP1STAT	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
SSP1CON1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
SSP1CON2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu

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

Shaded cells indicate conditions do not apply for the designated device.

Note 1: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

4: See Table 4-1 for Reset value for specific condition.

PIC18F87J11 FAMILY

TABLE 4-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices	Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets, CM Resets	Wake-up via WDT or Interrupt
ADRESH	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
ADRESL	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
ADCON0	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
ADCON1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
ANCON0	PIC18F6XJ1X	PIC18F8XJ1X	00-0 0000	uu-u uuuu
ANCON1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	uuuu uuuu
WDTCON	PIC18F6XJ1X	PIC18F8XJ1X	0x-0 ---0	0x-u ---0
ECCP1AS	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
ECCP1DEL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
CCPR1H	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
CCPR1L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
CCP1CON	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
ECCP2AS	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
ECCP2DEL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
CCPR2H	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
CCPR2L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
CCP2CON	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
ECCP3AS	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
ECCP3DEL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
CCPR3H	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
CCPR3L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
CCP3CON	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
SPBRG1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
RCREG1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
TXREG1	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
TXSTA1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0010	0000 0010
RCSTA1	PIC18F6XJ1X	PIC18F8XJ1X	0000 000x	0000 000x
SPBRG2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
RCREG2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
TXREG2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
TXSTA2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0010	0000 0010
EECON2	PIC18F6XJ1X	PIC18F8XJ1X	---- ----	---- ----
EECON1	PIC18F6XJ1X	PIC18F8XJ1X	--00 x00-	--00 u00-

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

Shaded cells indicate conditions do not apply for the designated device.

- Note 1:** When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.
- 2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).
 - 3: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).
 - 4: See Table 4-1 for Reset value for specific condition.

PIC18F87J11 FAMILY

TABLE 4-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets, CM Resets	Wake-up via WDT or Interrupt
IPR3	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
PIR3	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu ⁽³⁾
PIE3	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
IPR2	PIC18F6XJ1X	PIC18F8XJ1X	111- 1111	111- 1111	uuu- uuuu
PIR2	PIC18F6XJ1X	PIC18F8XJ1X	000- 0000	000- 0000	uuu- uuuu ⁽³⁾
PIE2	PIC18F6XJ1X	PIC18F8XJ1X	000- 0000	000- 0000	uuu- uuuu
IPR1	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
PIR1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu ⁽³⁾
PIE1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
RCSTA2	PIC18F6XJ1X	PIC18F8XJ1X	0000 000x	0000 000x	uuuu uuuu
OSCTUNE	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
TRISJ	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
TRISH	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
TRISG	PIC18F6XJ1X	PIC18F8XJ1X	---1 1111	---1 1111	--u uuuu
TRISF	PIC18F6XJ1X	PIC18F8XJ1X	1111 111-	1111 111-	uuuu uuu-
TRISE	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
TRISD	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
TRISC	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
TRISB	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
TRISA	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
LATJ	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATH	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATG	PIC18F6XJ1X	PIC18F8XJ1X	---x xxxx	--u uuuu	--u uuuu
LATF	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxx-	uuuu uuu-	uuuu uuu-
LATE	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATD	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATC	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATB	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATA	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu

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

- Note 1:** When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and Tosl are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.
- 2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).
 - 3: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).
 - 4: See Table 4-1 for Reset value for specific condition.

PIC18F87J11 FAMILY

TABLE 4-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices	Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets, CM Resets	Wake-up via WDT or Interrupt
PORTJ	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
PORTH	PIC18F6XJ1X	PIC18F8XJ1X	0000 xxxx	uuuu uuuu
PORTG	PIC18F6XJ1X	PIC18F8XJ1X	000x xxxx	000u uuuu
PORTF	PIC18F6XJ1X	PIC18F8XJ1X	x001 100-	xuuu uuu-
PORTE	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
PORTD	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
PORTC	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
PORTB	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
PORTA	PIC18F6XJ1X	PIC18F8XJ1X	000x 0000	000u 0000
SPBRGH1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
BAUDCON1	PIC18F6XJ1X	PIC18F8XJ1X	0100 0-00	0100 0-00
SPBRGH2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
BAUDCON2	PIC18F6XJ1X	PIC18F8XJ1X	0100 0-00	0100 0-00
TMR3H	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
TMR3L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
T3CON	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	uuuu uuuu
TMR4	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
PR4	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111
CVRCON	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
T4CON	PIC18F6XJ1X	PIC18F8XJ1X	-000 0000	-000 0000
CCPR4H	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
CCPR4L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
CCP4CON	PIC18F6XJ1X	PIC18F8XJ1X	--00 0000	--00 0000
CCPR5H	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
CCPR5L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
CCP5CON	PIC18F6XJ1X	PIC18F8XJ1X	--00 0000	--00 0000
SSP2BUF	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu
SSP2ADD	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
SSP2MSK	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
SSP2STAT	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
SSP2CON1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
SSP2CON2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000
CMSTAT	PIC18F6XJ1X	PIC18F8XJ1X	---- --11	---- --11

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

Shaded cells indicate conditions do not apply for the designated device.

- Note 1:** When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.
- 2:** When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).
- 3:** One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).
- 4:** See Table 4-1 for Reset value for specific condition.

PIC18F87J11 FAMILY

TABLE 4-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets, CM Resets	Wake-up via WDT or Interrupt
PMADDRH	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMDOUT1H	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMADDRL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMDOUT1L	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMDIN1H	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMDIN1L	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMCONH	PIC18F6XJ1X	PIC18F8XJ1X	0-00 0000	0-00 0000	u-uu uuuu
PMCONL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMMODEH	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMMODEL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMDOUT2H	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMDOUT2L	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMDIN2H	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMDIN2L	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMEH	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMEL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMSTATH	PIC18F6XJ1X	PIC18F8XJ1X	00-- 0000	00-- 0000	uu-- uuuu
PMSTATL	PIC18F6XJ1X	PIC18F8XJ1X	10-- 1111	10-- 1111	uu-- uuuu

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

Shaded cells indicate conditions do not apply for the designated device.

- Note 1:** When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.
- 2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).
 - 3: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).
 - 4: See Table 4-1 for Reset value for specific condition.

5.0 MEMORY ORGANIZATION

There are two types of memory in PIC18 Flash microcontroller devices:

- Program Memory
- Data RAM

As Harvard architecture devices, the data and program memories use separate busses; this allows for concurrent access of the two memory spaces.

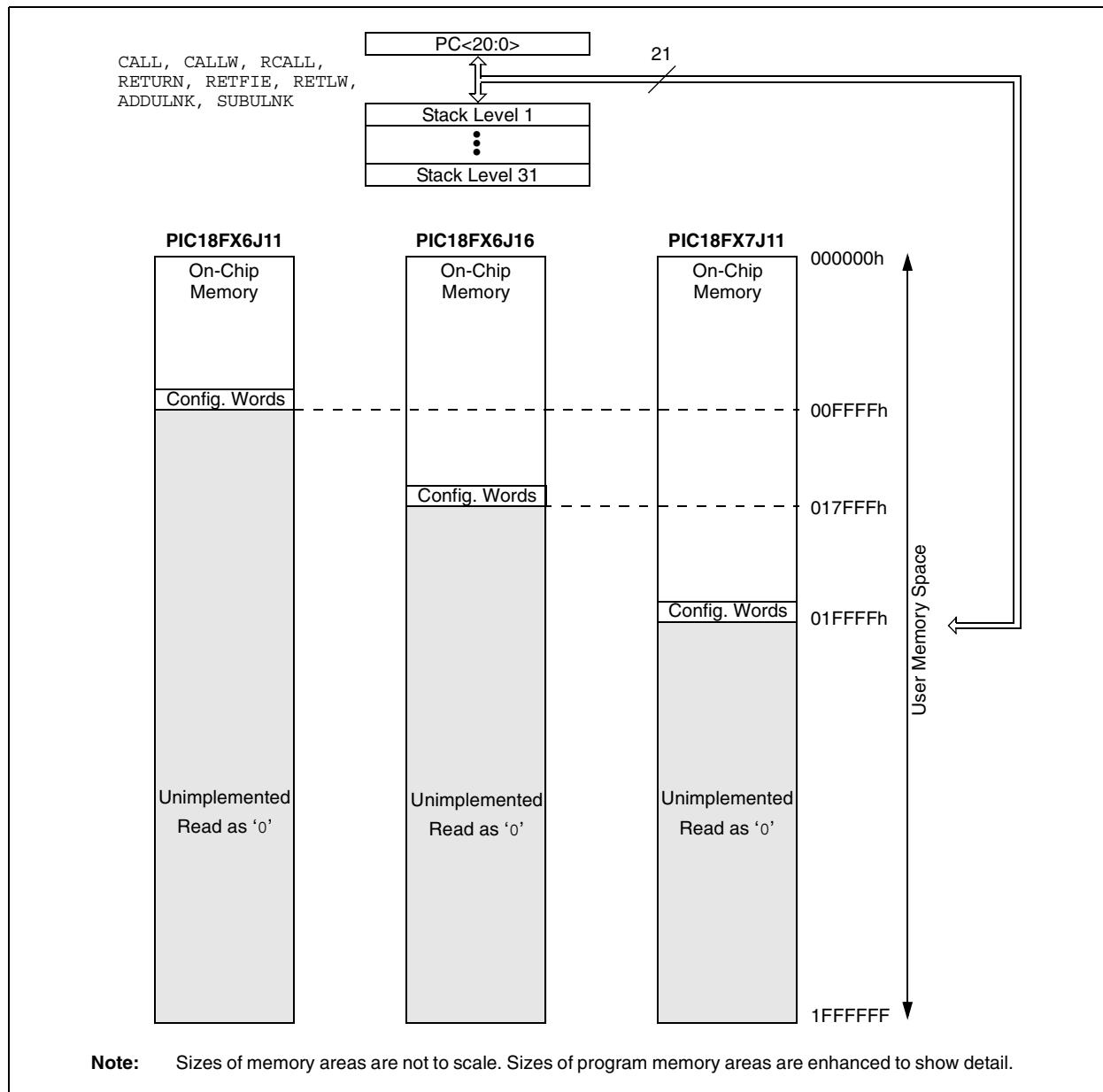
Additional detailed information on the operation of the Flash program memory is provided in **Section 6.0 “Flash Program Memory”**.

5.1 Program Memory Organization

PIC18 microcontrollers implement a 21-bit program counter which is capable of addressing a 2-Mbyte program memory space. Accessing a location between the upper boundary of the physically implemented memory and the 2-Mbyte address will return all ‘0’s (a NOP instruction).

The entire PIC18F87J11 Family of devices offers three different on-chip Flash program memory sizes, from 64 Kbytes (up to 16,384 single-word instructions) to 128 Kbytes (65,536 single-word instructions). The program memory maps for individual family members are shown in Figure 5-3.

FIGURE 5-1: MEMORY MAPS FOR PIC18F87J11 FAMILY DEVICES



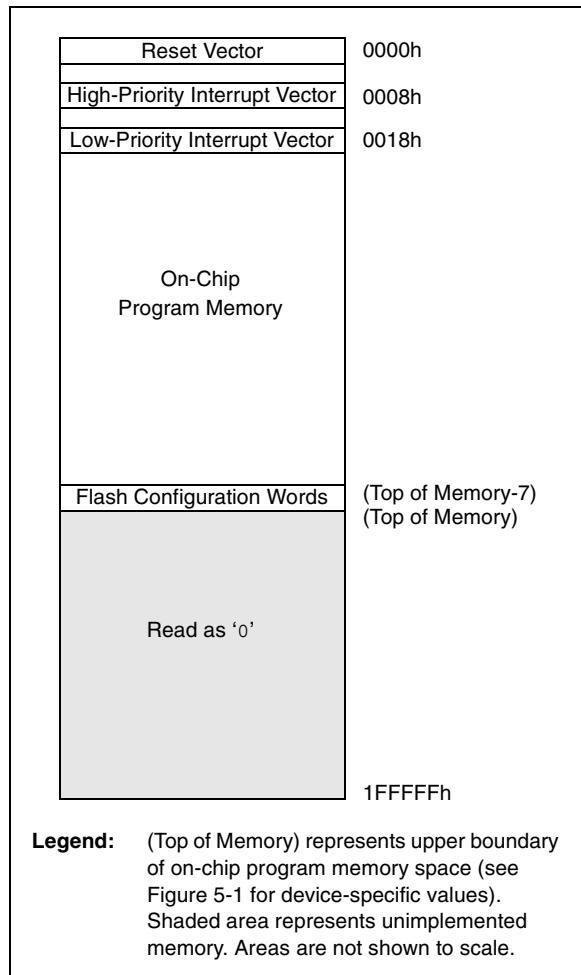
PIC18F87J11 FAMILY

5.1.1 HARD MEMORY VECTORS

All PIC18 devices have a total of three hard-coded return vectors in their program memory space. The Reset vector address is the default value to which the program counter returns on all device Resets; it is located at 0000h.

PIC18 devices also have two interrupt vector addresses for the handling of high-priority and low-priority interrupts. The high-priority interrupt vector is located at 0008h and the low-priority interrupt vector is at 0018h. Their locations in relation to the program memory map are shown in Figure 5-2.

FIGURE 5-2: HARD VECTOR AND CONFIGURATION WORD LOCATIONS FOR PIC18F87J11 FAMILY DEVICES



5.1.2 FLASH CONFIGURATION WORDS

Because PIC18F87J11 Family devices do not have persistent configuration memory, the top four words of on-chip program memory are reserved for configuration information. On Reset, the configuration information is copied into the Configuration registers.

The Configuration Words are stored in their program memory location in numerical order, starting with the lower byte of CONFIG1 at the lowest address and ending with the upper byte of CONFIG4. For these devices, only Configuration Words, CONFIG1 through CONFIG3, are used; CONFIG4 is reserved. The actual addresses of the Flash Configuration Word for devices in the PIC18F87J11 Family are shown in Table 5-1. Their location in the memory map is shown with the other memory vectors in Figure 5-2.

Additional details on the device Configuration Words are provided in **Section 24.1 “Configuration Bits”**.

TABLE 5-1: FLASH CONFIGURATION WORD FOR PIC18F87J11 FAMILY DEVICES

Device	Program Memory (Kbytes)	Configuration Word Addresses
PIC18F66J11	64	FFF8h to FFFFh
PIC18F86J11		
PIC18F66J16	96	17FF8h to 17FFFh
PIC18F86J16		
PIC18F67J11	128	1FFF8h to 1FFFFh
PIC18F87J11		

5.1.3 PIC18F8XJ11/8XJ16 PROGRAM MEMORY MODES

The 80-pin devices in this family can address up to a total of 2 Mbytes of program memory. This is achieved through the external memory bus. There are two distinct operating modes available to the controllers:

- Microcontroller (MC)
- Extended Microcontroller (EMC)

The program memory mode is determined by setting the EMB Configuration bits (CONFIG3L<5:4>), as shown in Register 5-1. (See also **Section 24.1 “Configuration Bits”** for additional details on the device Configuration bits.)

The program memory modes operate as follows:

- The **Microcontroller Mode** accesses only on-chip Flash memory. Attempts to read above the top of on-chip memory causes a read of all ‘0’s (a NOP instruction).

The Microcontroller mode is also the only operating mode available to 64-pin devices.

- The **Extended Microcontroller Mode** allows access to both internal and external program memories as a single block. The device can access its entire on-chip program memory; above this, the device accesses external program memory up to the 2-Mbyte program space limit. Execution automatically switches between the two memories as required.

The setting of the EMB Configuration bits also controls the address bus width of the external memory bus. This is covered in more detail in **Section 7.0 “External Memory Bus”**.

In all modes, the microcontroller has complete access to data RAM.

Figure 5-3 compares the memory maps of the different program memory modes. The differences between on-chip and external memory access limitations are more fully explained in Table 5-2.

REGISTER 5-1: CONFIG3L: CONFIGURATION REGISTER 3 LOW

R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	U-0	U-0	U-0
WAIT ⁽¹⁾	BW ⁽¹⁾	EMB1 ⁽¹⁾	EMB0 ⁽¹⁾	EASHFT ⁽¹⁾	—	—	—
bit 7	bit 0						

Legend:	WO = Write-Once bit	
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

bit 7	WAIT: External Bus Wait Enable bit ⁽¹⁾ 1 = Wait states on the external bus are disabled 0 = Wait states on the external bus are enabled and selected by MEMCON<5:4>
bit 6	BW: Data Bus Width Select bit ⁽¹⁾ 1 = 16-Bit Data Width modes 0 = 8-Bit Data Width modes
bit 5-4	EMB1:EMB0: External Memory Bus Configuration bits ⁽¹⁾ 11 = Microcontroller mode, external bus disabled 10 = Extended Microcontroller mode, 12-bit address width for external bus 01 = Extended Microcontroller mode, 16-bit address width for external bus 00 = Extended Microcontroller mode, 20-bit address width for external bus
bit 3	EASHFT: External Address Bus Shift Enable bit ⁽¹⁾ 1 = Address shifting enabled – external address bus is shifted to start at 000000h 0 = Address shifting disabled – external address bus reflects the PC value
bit 2-0	Unimplemented: Read as ‘0’

Note 1: Implemented only on 80-pin devices.

PIC18F87J11 FAMILY

5.1.4 EXTENDED MICROCONTROLLER MODE AND ADDRESS SHIFTING

By default, devices in Extended Microcontroller mode directly present the program counter value on the external address bus for those addresses in the range of the external memory space. In practical terms, this means addresses in the external memory device below the top of on-chip memory are unavailable.

To avoid this, the Extended Microcontroller mode implements an address shifting option to enable automatic address translation. In this mode, addresses presented on the external bus are shifted down by the size of the on-chip program memory and are remapped to start at 0000h. This allows the complete use of the external memory device's memory space as an extension of the device's on-chip program memory.

FIGURE 5-3: MEMORY MAPS FOR PIC18F87J11 FAMILY PROGRAM MEMORY MODES

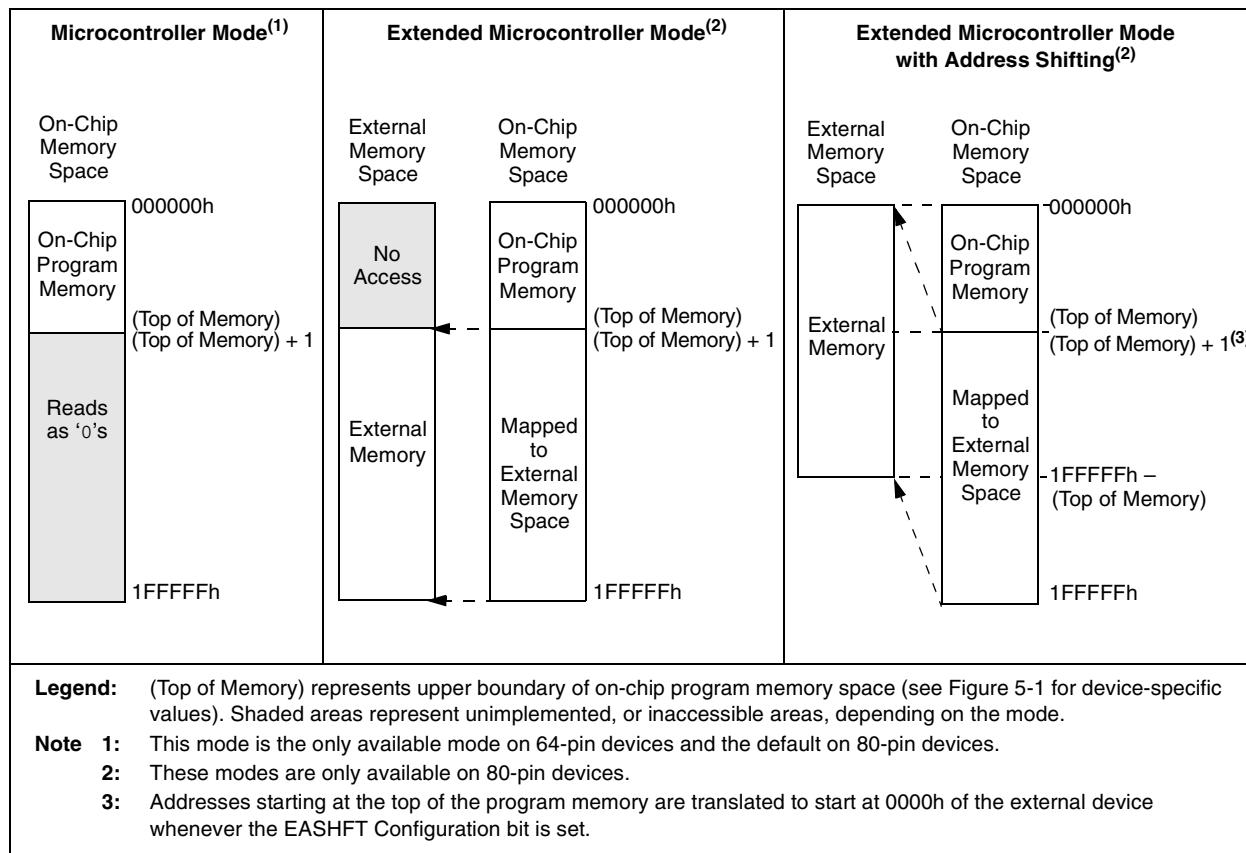


TABLE 5-2: MEMORY ACCESS FOR PIC18F8X11/8616 PROGRAM MEMORY MODES

Operating Mode	Internal Program Memory			External Program Memory		
	Execution From	Table Read From	Table Write To	Execution From	Table Read From	Table Write To
Microcontroller	Yes	Yes	Yes	No Access	No Access	No Access
Extended Microcontroller	Yes	Yes	Yes	Yes	Yes	Yes

5.1.5 PROGRAM COUNTER

The Program Counter (PC) specifies the address of the instruction to fetch for execution. The PC is 21 bits wide and is contained in three separate 8-bit registers. The low byte, known as the PCL register, is both readable and writable. The high byte, or PCH register, contains the PC_{15:8} bits; it is not directly readable or writable. Updates to the PCH register are performed through the PCLATH register. The upper byte is called PCU. This register contains the PC_{20:16} bits; it is also not directly readable or writable. Updates to the PCU register are performed through the PCLATU register.

The contents of PCLATH and PCLATU are transferred to the program counter by any operation that writes PCL. Similarly, the upper two bytes of the program counter are transferred to PCLATH and PCLATU by an operation that reads PCL. This is useful for computed offsets to the PC (see **Section 5.1.8.1 “Computed GOTO”**).

The PC addresses bytes in the program memory. To prevent the PC from becoming misaligned with word instructions, the Least Significant bit of PCL is fixed to a value of ‘0’. The PC increments by 2 to address sequential instructions in the program memory.

The CALL, RCALL, GOTO and program branch instructions write to the program counter directly. For these instructions, the contents of PCLATH and PCLATU are not transferred to the program counter.

5.1.6 RETURN ADDRESS STACK

The return address stack allows any combination of up to 31 program calls and interrupts to occur. The PC is pushed onto the stack when a CALL or RCALL instruction is executed, or an interrupt is Acknowledged. The PC value is pulled off the stack on a RETURN, RETLW or a RETFIE instruction (and on ADDULNK and SUBULNK instructions if the extended instruction set is enabled). PCLATU and PCLATH are not affected by any of the RETURN or CALL instructions.

The stack operates as a 31-word by 21-bit RAM and a 5-bit Stack Pointer, STKPTR. The stack space is not part of either program or data space. The Stack Pointer is readable and writable and the address on the top of the stack is readable and writable through the Top-of-Stack Special Function Registers. Data can also be pushed to, or popped from the stack, using these registers.

A CALL type instruction causes a push onto the stack. The Stack Pointer is first incremented and the location pointed to by the Stack Pointer is written with the contents of the PC (already pointing to the instruction following the CALL). A RETURN type instruction causes a pop from the stack. The contents of the location pointed to by the STKPTR are transferred to the PC and then the Stack Pointer is decremented.

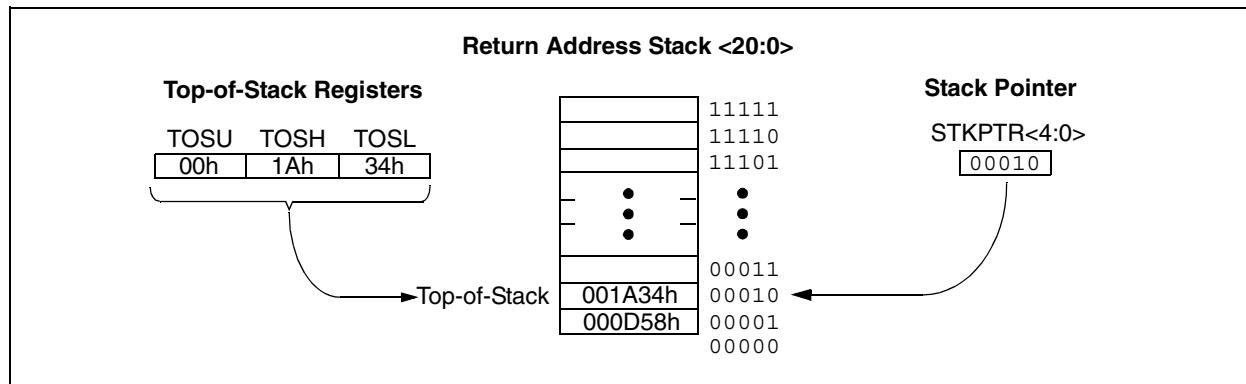
The Stack Pointer is initialized to ‘00000’ after all Resets. There is no RAM associated with the location corresponding to a Stack Pointer value of ‘00000’; this is only a Reset value. Status bits indicate if the stack is full, has overflowed or has underflowed.

5.1.6.1 Top-of-Stack Access

Only the top of the return address stack (TOS) is readable and writable. A set of three registers, TOSU:TOSH:TOSL, hold the contents of the stack location pointed to by the STKPTR register (Figure 5-4). This allows users to implement a software stack if necessary. After a CALL, RCALL or interrupt (and ADDULNK and SUBULNK instructions if the extended instruction set is enabled), the software can read the pushed value by reading the TOSU:TOSH:TOSL registers. These values can be placed on a user-defined software stack. At return time, the software can return these values to TOSU:TOSH:TOSL and do a return.

The user must disable the global interrupt enable bits while accessing the stack to prevent inadvertent stack corruption.

FIGURE 5-4: RETURN ADDRESS STACK AND ASSOCIATED REGISTERS



PIC18F87J11 FAMILY

5.1.6.2 Return Stack Pointer (STKPTR)

The STKPTR register (Register 5-2) contains the Stack Pointer value, the STKFUL (Stack Full) status bit and the STKUNF (Stack Underflow) status bits. The value of the Stack Pointer can be 0 through 31. The Stack Pointer increments before values are pushed onto the stack and decrements after values are popped off the stack. On Reset, the Stack Pointer value will be zero. The user may read and write the Stack Pointer value. This feature can be used by a Real-Time Operating System (RTOS) for return stack maintenance.

After the PC is pushed onto the stack 31 times (without popping any values off the stack), the STKFUL bit is set. The STKFUL bit is cleared by software or by a POR.

The action that takes place when the stack becomes full depends on the state of the STVREN (Stack Overflow Reset Enable) Configuration bit. (Refer to **Section 24.1 “Configuration Bits”** for a description of the device Configuration bits.) If STVREN is set (default), the 31st push will push the (PC + 2) value onto the stack, set the STKFUL bit and reset the device. The STKFUL bit will remain set and the Stack Pointer will be set to zero.

If STVREN is cleared, the STKFUL bit will be set on the 31st push and the Stack Pointer will increment to 31. Any additional pushes will not overwrite the 31st push and the STKPTR will remain at 31.

When the stack has been popped enough times to unload the stack, the next pop will return a value of zero to the PC and set the STKUNF bit, while the Stack Pointer remains at zero. The STKUNF bit will remain set until cleared by software or until a POR occurs.

Note: Returning a value of zero to the PC on an underflow has the effect of vectoring the program to the Reset vector, where the stack conditions can be verified and appropriate actions can be taken. This is not the same as a Reset, as the contents of the SFRs are not affected.

5.1.6.3 PUSH and POP Instructions

Since the Top-of-Stack is readable and writable, the ability to push values onto the stack and pull values off the stack, without disturbing normal program execution, is a desirable feature. The PIC18 instruction set includes two instructions, **PUSH** and **POP**, that permit the TOS to be manipulated under software control. TOSU, TOSH and TOSL can be modified to place data or a return address on the stack.

The **PUSH** instruction places the current PC value onto the stack. This increments the Stack Pointer and loads the current PC value onto the stack.

The **POP** instruction discards the current TOS by decrementing the Stack Pointer. The previous value pushed onto the stack then becomes the TOS value.

REGISTER 5-2: STKPTR: STACK POINTER REGISTER

R/C-0	R/C-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STKFUL ⁽¹⁾	STKUNF ⁽¹⁾	—	SP4	SP3	SP2	SP1	SP0
bit 7	bit 0						

Legend:

R = Readable bit

-n = Value at POR

C = Clearable-only bit

W = Writable bit

'1' = Bit is set

U = Unimplemented bit, read as '0'

'0' = Bit is cleared

x = Bit is unknown

- bit 7 **STKFUL:** Stack Full Flag bit⁽¹⁾
1 = Stack became full or overflowed
0 = Stack has not become full or overflowed
- bit 6 **STKUNF:** Stack Underflow Flag bit⁽¹⁾
1 = Stack underflow occurred
0 = Stack underflow did not occur
- bit 5 **Unimplemented:** Read as '0'
- bit 4-0 **SP4:SP0:** Stack Pointer Location bits

Note 1: Bit 7 and bit 6 are cleared by user software or by a POR.

5.1.6.4 Stack Full and Underflow Resets

Device Resets on stack overflow and stack underflow conditions are enabled by setting the STVREN bit in Configuration Register 1L. When STVREN is set, a full or underflow condition will set the appropriate STKFUL or STKUNF bit and then cause a device Reset. When STVREN is cleared, a full or underflow condition will set the appropriate STKFUL or STKUNF bit, but not cause a device Reset. The STKFUL or STKUNF bits are cleared by the user software or a Power-on Reset.

5.1.7 FAST REGISTER STACK

A Fast Register Stack is provided for the STATUS, WREG and BSR registers to provide a “fast return” option for interrupts. This stack is only one level deep and is neither readable nor writable. It is loaded with the current value of the corresponding register when the processor vectors for an interrupt. All interrupt sources will push values into the Stack registers. The values in the registers are then loaded back into the working registers if the RETFIE, FAST instruction is used to return from the interrupt.

If both low and high-priority interrupts are enabled, the Stack registers cannot be used reliably to return from low-priority interrupts. If a high-priority interrupt occurs while servicing a low-priority interrupt, the Stack register values stored by the low-priority interrupt will be overwritten. In these cases, users must save the key registers in software during a low-priority interrupt.

If interrupt priority is not used, all interrupts may use the Fast Register Stack for returns from interrupt. If no interrupts are used, the Fast Register Stack can be used to restore the STATUS, WREG and BSR registers at the end of a subroutine call. To use the Fast Register Stack for a subroutine call, a CALL label, FAST instruction must be executed to save the STATUS, WREG and BSR registers to the Fast Register Stack. A RETURN, FAST instruction is then executed to restore these registers from the Fast Register Stack.

Example 5-1 shows a source code example that uses the Fast Register Stack during a subroutine call and return.

EXAMPLE 5-1: FAST REGISTER STACK CODE EXAMPLE

```
CALL SUB1, FAST      ;STATUS, WREG, BSR
                      ;SAVED IN FAST REGISTER
                      ;STACK
•
•
SUB1   •
•
RETURN FAST       ;RESTORE VALUES SAVED
                  ;IN FAST REGISTER STACK
```

5.1.8 LOOK-UP TABLES IN PROGRAM MEMORY

There may be programming situations that require the creation of data structures, or look-up tables, in program memory. For PIC18 devices, look-up tables can be implemented in two ways:

- Computed GOTO
- Table Reads

5.1.8.1 Computed GOTO

A computed GOTO is accomplished by adding an offset to the program counter. An example is shown in Example 5-2.

A look-up table can be formed with an ADDWF PCL instruction and a group of RETLW nn instructions. The W register is loaded with an offset into the table before executing a call to that table. The first instruction of the called routine is the ADDWF PCL instruction. The next instruction executed will be one of the RETLW nn instructions that returns the value ‘nn’ to the calling function.

The offset value (in WREG) specifies the number of bytes that the program counter should advance and should be multiples of 2 (LSb = 0).

In this method, only one data byte may be stored in each instruction location and room on the return address stack is required.

EXAMPLE 5-2: COMPUTED GOTO USING AN OFFSET VALUE

MOVF	OFFSET, W
CALL	TABLE
ORG	nn00h
TABLE	ADDWF PCL
	RETLW nnh
	RETLW nnh
	RETLW nnh
.	.
.	.
.	.

5.1.8.2 Table Reads

A better method of storing data in program memory allows two bytes of data to be stored in each instruction location.

Look-up table data may be stored two bytes per program word while programming. The Table Pointer (TBLPTR) specifies the byte address and the Table Latch (TABLAT) contains the data that is read from the program memory. Data is transferred from program memory one byte at a time.

Table read operation is discussed further in Section 6.1 “Table Reads and Table Writes”.

PIC18F87J11 FAMILY

5.2 PIC18 Instruction Cycle

5.2.1 CLOCKING SCHEME

The microcontroller clock input, whether from an internal or external source, is internally divided by four to generate four non-overlapping quadrature clocks (Q1, Q2, Q3 and Q4). Internally, the program counter is incremented on every Q1; the instruction is fetched from the program memory and latched into the Instruction Register (IR) during Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 5-5.

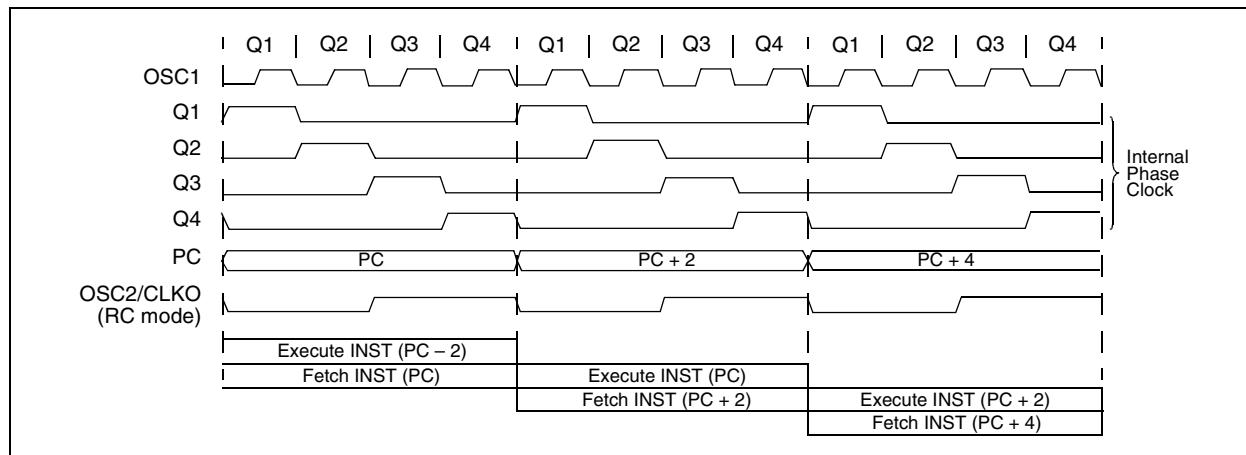
5.2.2 INSTRUCTION FLOW/PIPELINING

An “Instruction Cycle” consists of four Q cycles, Q1 through Q4. The instruction fetch and execute are pipelined in such a manner that a fetch takes one instruction cycle, while the decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., GOTO), then two cycles are required to complete the instruction (Example 5-3).

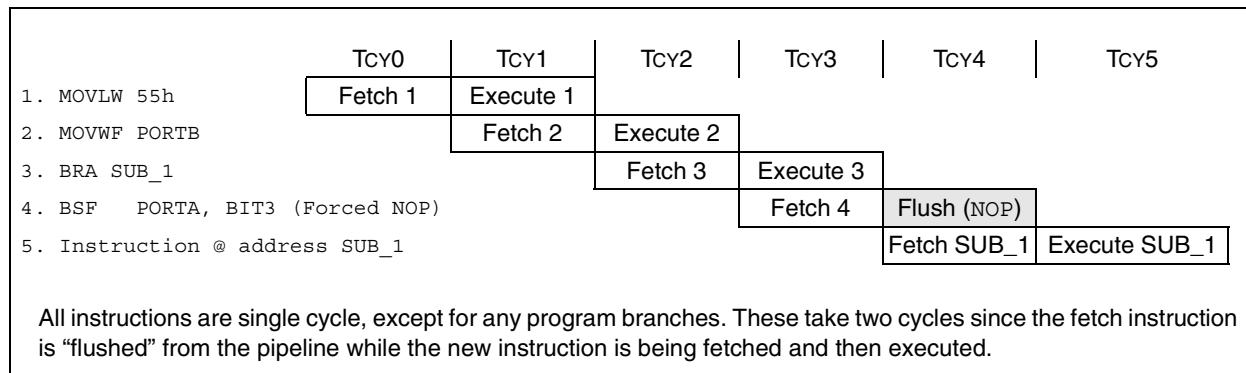
A fetch cycle begins with the Program Counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the Instruction Register (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3 and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 5-5: CLOCK/INSTRUCTION CYCLE



EXAMPLE 5-3: INSTRUCTION PIPELINE FLOW



5.2.3 INSTRUCTIONS IN PROGRAM MEMORY

The program memory is addressed in bytes. Instructions are stored as two bytes or four bytes in program memory. The Least Significant Byte of an instruction word is always stored in a program memory location with an even address (LSB = 0). To maintain alignment with instruction boundaries, the PC increments in steps of 2 and the LSB will always read '0' (see **Section 5.1.5 "Program Counter"**).

Figure 5-6 shows an example of how instruction words are stored in the program memory.

The CALL and GOTO instructions have the absolute program memory address embedded into the instruction. Since instructions are always stored on word boundaries, the data contained in the instruction is a word address. The word address is written to PC<20:1> which accesses the desired byte address in program memory. Instruction #2 in Figure 5-6 shows how the instruction, GOTO 0006h, is encoded in the program memory. Program branch instructions, which encode a relative address offset, operate in the same manner. The offset value stored in a branch instruction represents the number of single-word instructions that the PC will be offset by. **Section 25.0 "Instruction Set Summary"** provides further details of the instruction set.

FIGURE 5-6: INSTRUCTIONS IN PROGRAM MEMORY

Program Memory Byte Locations →		Word Address ↓
LSB = 1	LSB = 0	
		000000h
		000002h
		000004h
		000006h
0Fh	55h	000008h
EFh	03h	0000Ah
F0h	00h	0000Ch
C1h	23h	0000Eh
F4h	56h	000010h
		000012h
		000014h

Instruction 1: MOVLW 055h
 Instruction 2: GOTO 0006h
 Instruction 3: MOVFF 123h, 456h

5.2.4 TWO-WORD INSTRUCTIONS

The standard PIC18 instruction set has four two-word instructions: CALL, MOVFF, GOTO and LSR. In all cases, the second word of the instructions always has '1111' as its four Most Significant bits; the other 12 bits are literal data, usually a data memory address.

The use of '1111' in the 4 MSbs of an instruction specifies a special form of NOP. If the instruction is executed in proper sequence – immediately after the first word – the data in the second word is accessed

and used by the instruction sequence. If the first word is skipped for some reason and the second word is executed by itself, a NOP is executed instead. This is necessary for cases when the two-word instruction is preceded by a conditional instruction that changes the PC. Example 5-4 shows how this works.

Note: See **Section 5.5 "Program Memory and the Extended Instruction Set"** for information on two-word instructions in the extended instruction set.

EXAMPLE 5-4: TWO-WORD INSTRUCTIONS

CASE 1:	
Object Code	Source Code
0110 0110 0000 0000	TSTFSZ REG1 ; is RAM location 0?
1100 0001 0010 0011	MOVFF REG1, REG2 ; No, skip this word
1111 0100 0101 0110	; Execute this word as a NOP
0010 0100 0000 0000	ADDWF REG3 ; continue code

CASE 2:	
Object Code	Source Code
0110 0110 0000 0000	TSTFSZ REG1 ; is RAM location 0?
1100 0001 0010 0011	MOVFF REG1, REG2 ; Yes, execute this word
1111 0100 0101 0110	; 2nd word of instruction
0010 0100 0000 0000	ADDWF REG3 ; continue code

5.3 Data Memory Organization

Note: The operation of some aspects of data memory are changed when the PIC18 extended instruction set is enabled. See **Section 5.6 “Data Memory and the Extended Instruction Set”** for more information.

The data memory in PIC18 devices is implemented as static RAM. Each register in the data memory has a 12-bit address, allowing up to 4096 bytes of data memory. The memory space is divided into as many as 16 banks that contain 256 bytes each. The PIC18F87J11 family implements all available banks and provide 3936 bytes of data memory available to the user. Figure 5-7 shows the data memory organization for the devices.

The data memory contains Special Function Registers (SFRs) and General Purpose Registers (GPRs). The SFRs are used for control and status of the controller and peripheral functions, while GPRs are used for data storage and scratchpad operations in the user's application. Any read of an unimplemented location will read as '0's.

The instruction set and architecture allow operations across all banks. The entire data memory may be accessed by Direct, Indirect or Indexed Addressing modes. Addressing modes are discussed later in this section.

To ensure that commonly used registers (select SFRs and select GPRs) can be accessed in a single cycle, PIC18 devices implement an Access Bank. This is a 256-byte memory space that provides fast access to select SFRs and the lower portion of GPR Bank 0 without using the BSR. **Section 5.3.2 “Access Bank”** provides a detailed description of the Access RAM.

5.3.1 BANK SELECT REGISTER

Large areas of data memory require an efficient addressing scheme to make rapid access to any address possible. Ideally, this means that an entire address does not need to be provided for each read or write operation. For PIC18 devices, this is accomplished with a RAM banking scheme. This divides the memory space into 16 contiguous banks of 256 bytes. Depending on the instruction, each location can be addressed directly by its full 12-bit address, or an 8-bit low-order address and a 4-bit Bank Pointer.

Most instructions in the PIC18 instruction set make use of the Bank Pointer, known as the Bank Select Register (BSR). This SFR holds the 4 Most Significant bits of a location's address; the instruction itself includes the 8 Least Significant bits. Only the four lower bits of the BSR are implemented (BSR3:BSR0). The upper four bits are unused; they will always read '0' and cannot be written to. The BSR can be loaded directly by using the MOVLB instruction.

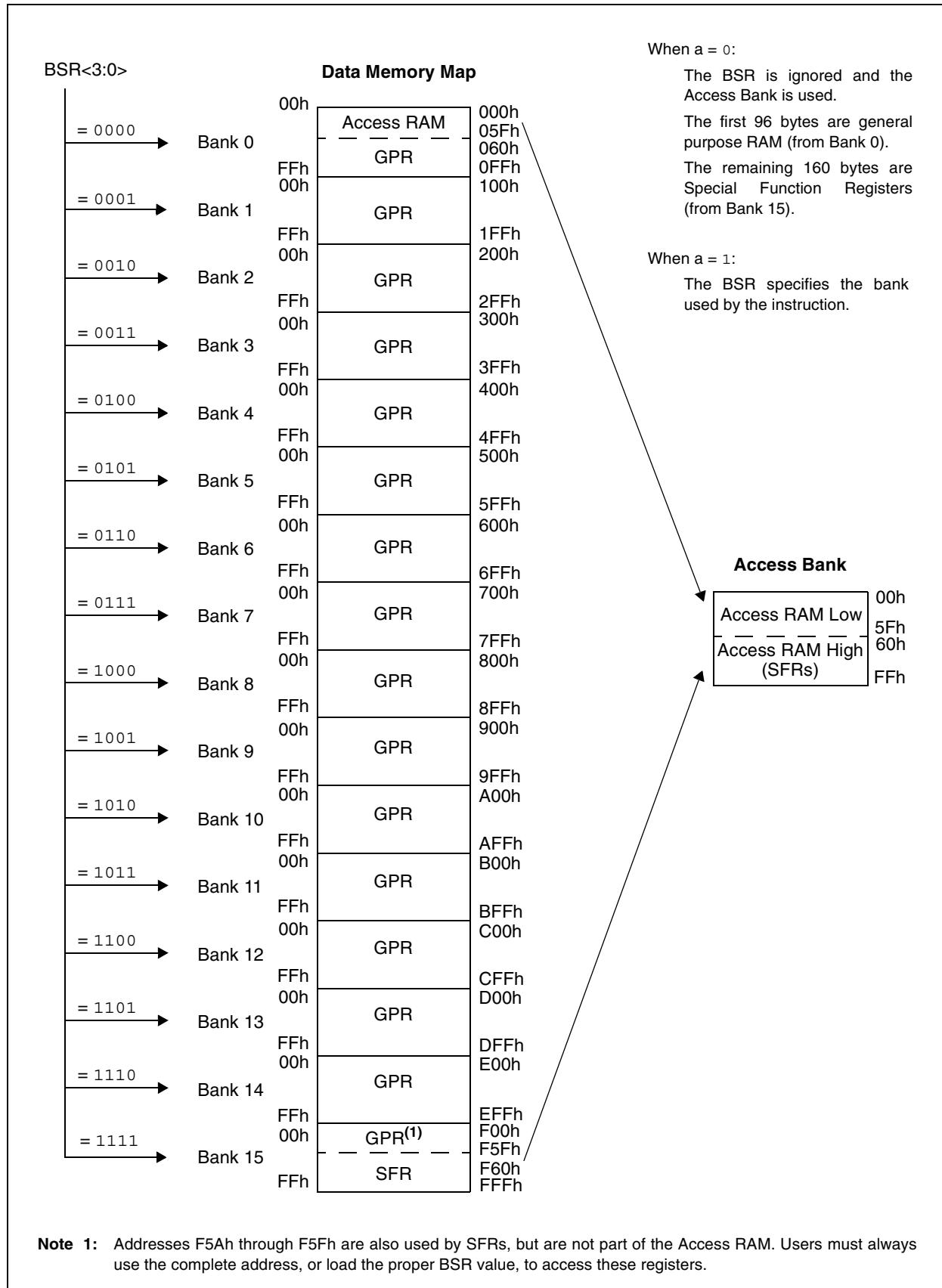
The value of the BSR indicates the bank in data memory. The 8 bits in the instruction show the location in the bank and can be thought of as an offset from the bank's lower boundary. The relationship between the BSR's value and the bank division in data memory is shown in Figure 5-8.

Since up to 16 registers may share the same low-order address, the user must always be careful to ensure that the proper bank is selected before performing a data read or write. For example, writing what should be program data to an 8-bit address of F9h while the BSR is 0Fh, will end up resetting the program counter.

While any bank can be selected, only those banks that are actually implemented can be read or written to. Writes to unimplemented banks are ignored, while reads from unimplemented banks will return '0's. Even so, the STATUS register will still be affected as if the operation was successful. The data memory map in Figure 5-7 indicates which banks are implemented.

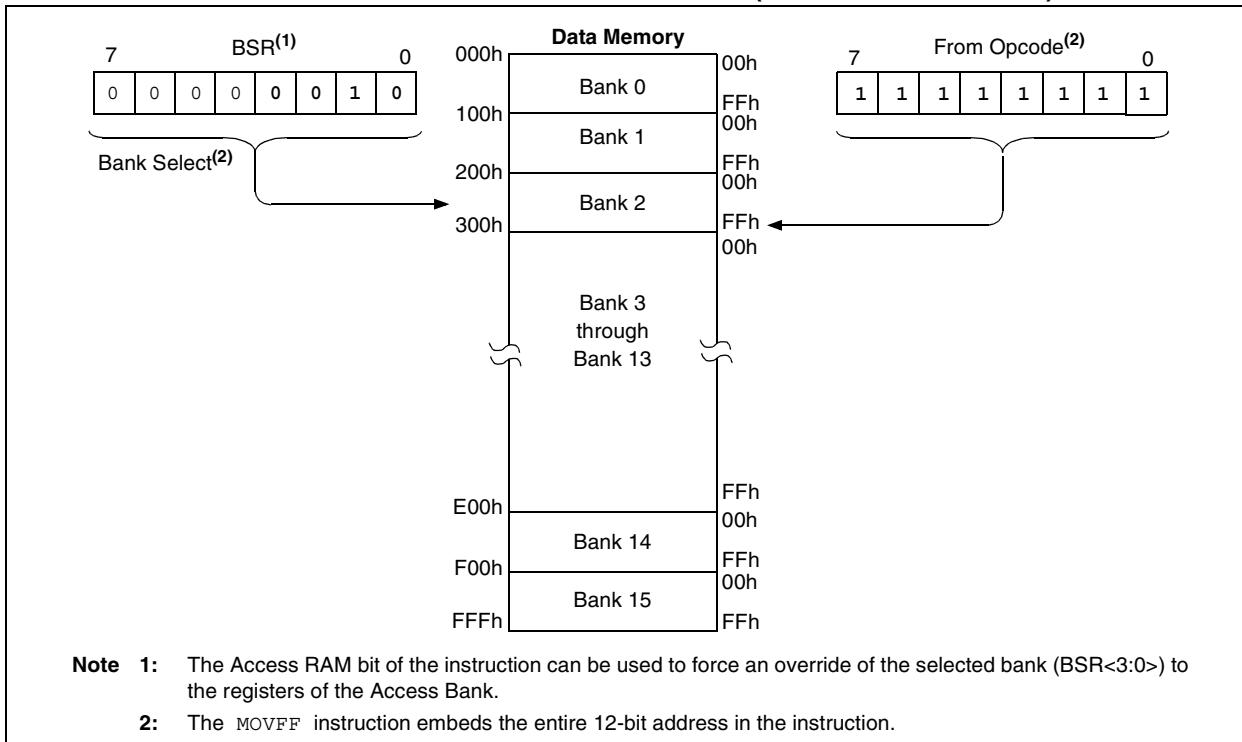
In the core PIC18 instruction set, only the MOVFF instruction fully specifies the 12-bit address of the source and target registers. This instruction ignores the BSR completely when it executes. All other instructions include only the low-order address as an operand and must use either the BSR or the Access Bank to locate their target registers.

FIGURE 5-7: DATA MEMORY MAP FOR PIC18F87J11 FAMILY DEVICES



PIC18F87J11 FAMILY

FIGURE 5-8: USE OF THE BANK SELECT REGISTER (DIRECT ADDRESSING)



5.3.2 ACCESS BANK

While the use of the BSR with an embedded 8-bit address allows users to address the entire range of data memory, it also means that the user must always ensure that the correct bank is selected. Otherwise, data may be read from or written to the wrong location. This can be disastrous if a GPR is the intended target of an operation, but an SFR is written to instead. Verifying and/or changing the BSR for each read or write to data memory can become very inefficient.

To streamline access for the most commonly used data memory locations, the data memory is configured with an Access Bank, which allows users to access a mapped block of memory without specifying a BSR. The Access Bank consists of the first 96 bytes of memory (00h-5Fh) in Bank 0 and the last 160 bytes of memory (60h-FFh) in Bank 15. The lower half is known as the "Access RAM" and is composed of GPRs. The upper half is where the device's SFRs are mapped. These two areas are mapped contiguously in the Access Bank and can be addressed in a linear fashion by an 8-bit address (Figure 5-7).

The Access Bank is used by core PIC18 instructions that include the Access RAM bit (the 'a' parameter in the instruction). When 'a' is equal to '1', the instruction uses the BSR and the 8-bit address included in the opcode for the data memory address. When 'a' is '0', however, the instruction is forced to use the Access Bank address map; the current value of the BSR is ignored entirely.

Using this "forced" addressing allows the instruction to operate on a data address in a single cycle without updating the BSR first. For 8-bit addresses of 60h and above, this means that users can evaluate and operate on SFRs more efficiently. The Access RAM below 60h is a good place for data values that the user might need to access rapidly, such as immediate computational results or common program variables. Access RAM also allows for faster and more code efficient context saving and switching of variables.

The mapping of the Access Bank is slightly different when the extended instruction set is enabled (XINST Configuration bit = 1). This is discussed in more detail in **Section 5.6.3 "Mapping the Access Bank in Indexed Literal Offset Mode"**.

5.3.3 GENERAL PURPOSE REGISTER FILE

PIC18 devices may have banked memory in the GPR area. This is data RAM which is available for use by all instructions. GPRs start at the bottom of Bank 0 (address 000h) and grow upwards towards the bottom of the SFR area. GPRs are not initialized by a Power-on Reset and are unchanged on all other Resets.

5.3.4 SPECIAL FUNCTION REGISTERS

The Special Function Registers (SFRs) are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. SFRs start at the top of data memory (FFFh) and extend downward to occupy more than the top half of Bank 15 (F5Ah to FFFh). A list of these registers is given in Table 5-3, Table 5-4 and Table 5-5.

The SFRs can be classified into two sets: those associated with the “core” device functionality (ALU, Resets and interrupts) and those related to the peripheral functions. The Reset and interrupt registers are described in their respective chapters, while the

ALU’s STATUS register is described later in this section. Registers related to the operation of the peripheral features are described in the chapter for that peripheral.

The SFRs are typically distributed among the peripherals whose functions they control. Unused SFR locations are unimplemented and read as ‘0’s

Note: Addresses, F5Ah through F5Fh, are not part of the Access Bank. These registers must always be accessed using the Bank Select Register.

TABLE 5-3: SPECIAL FUNCTION REGISTER MAP FOR PIC18F87J11 FAMILY DEVICES

Address	Name	Address	Name	Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2 ⁽¹⁾	FBFh	ECCP1AS	F9Fh	IPR1	F7Fh	SPBRGH1	F5Fh	PMDIN2H
FFEh	TOSH	FDEh	POSTINC2 ⁽¹⁾	FBEh	ECCP1DEL	F9Eh	PIR1	F7Eh	BAUDCON1	F5Eh	PMDIN2L
FFDh	TOSL	FDDh	POSTDEC2 ⁽¹⁾	FBDh	CCPR1H	F9Dh	PIE1	F7Dh	SPBRGH2	F5Dh	PMEH
FFCh	STKPTR	FDCh	PREINC2 ⁽¹⁾	FBCh	CCPR1L	F9Ch	RCSTA2	F7Ch	BAUDCON2	F5Ch	PMEL
FFBh	PCLATU	FDBh	PLUSW2 ⁽¹⁾	FBBh	CCP1CON	F9Bh	OSCTUNE	F7Bh	TMR3H	F5Bh	PMSTATH
FFAh	PCLATH	FDAh	FSR2H	FBAh	ECCP2AS	F9Ah	TRISJ ⁽²⁾	F7Ah	TMR3L	F5Ah	PMSTATL
FF9h	PCL	FD9h	FSR2L	FB9h	ECCP2DEL	F99h	TRISH ⁽²⁾	F79h	T3CON	F59h	—
FF8h	TBLPTRU	FD8h	STATUS	FB8h	CCPR2H	F98h	TRISG	F78h	TMR4	F58h	—
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	CCPR2L	F97h	TRISF	F77h	PR4 ⁽³⁾	F57h	—
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	CCP2CON	F96h	TRISE	F76h	T4CON	F56h	—
FF5h	TABLAT	FD5h	T0CON	FB5h	ECCP3AS	F95h	TRISD	F75h	CCPR4H	F55h	—
FF4h	PRODH	FD4h	—	FB4h	ECCP3DEL	F94h	TRISC	F74h	CCPR4L	F54h	—
FF3h	PRODL	FD3h	OSCCON ⁽³⁾	FB3h	CCPR3H	F93h	TRISB	F73h	CCP4CON	F53h	—
FF2h	INTCON	FD2h	CM1CON	FB2h	CCPR3L	F92h	TRISA	F72h	CCPR5H	F52h	—
FF1h	INTCON2	FD1h	CM2CON	FB1h	CCP3CON	F91h	LATJ ⁽²⁾	F71h	CCPR5L	F51h	—
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRG1	F90h	LATH ⁽²⁾	F70h	CCP5CON	F50h	—
FEFh	INDF0 ⁽¹⁾	FCFh	TMR1H ⁽³⁾	FAFh	RCREG1	F8Fh	LATG	F6Fh	SSP2BUF	F4Fh	—
FEEh	POSTINC0 ⁽¹⁾	FCEh	TMR1L ⁽³⁾	FAEh	TXREG1	F8Eh	LATF	F6Eh	SSP2ADD	F4Eh	—
FEDh	POSTDEC0 ⁽¹⁾	FCDh	T1CON ⁽³⁾	FADh	TXSTA1	F8Dh	LATE	F6Dh	SSP2STAT	F4Dh	—
FECh	PREINC0 ⁽¹⁾	FCCh	TMR2 ⁽³⁾	FACh	RCSTA1	F8Ch	LATD	F6Ch	SSP2CON1	F4Ch	—
FEBh	PLUSW0 ⁽¹⁾	FCBh	PR2 ⁽³⁾	FABh	SPBRG2	F8Bh	LATC	F6Bh	SSP2CON2	F4Bh	—
FEAh	FSR0H	FCAh	T2CON	FAAh	RCREG2	F8Ah	LATB	F6Ah	CMSTAT	F4Ah	—
FE9h	FSR0L	FC9h	SSP1BUF	FA9h	TXREG2	F89h	LATA	F69h	PMADDRH ⁽⁴⁾	F49h	—
FE8h	WREG	FC8h	SSP1ADD	FA8h	TXSTA2	F88h	PORTJ ⁽²⁾	F68h	PMADDRL ⁽⁴⁾	F48h	—
FE7h	INDF1 ⁽¹⁾	FC7h	SSP1STAT	FA7h	EECON2	F87h	PORTH ⁽²⁾	F67h	PMDIN1H	F47h	—
FE6h	POSTINC1 ⁽¹⁾	FC6h	SSP1CON1	FA6h	EECON1	F86h	PORTG	F66h	PMDIN1L	F46h	—
FE5h	POSTDEC1 ⁽¹⁾	FC5h	SSP1CON2	FA5h	IPR3	F85h	PORTF	F65h	PMCONH	F45h	—
FE4h	PREINC1 ⁽¹⁾	FC4h	ADRESH	FA4h	PIR3	F84h	PORTE	F64h	PMCONL	F44h	—
FE3h	PLUSW1 ⁽¹⁾	FC3h	ADRESL	FA3h	PIE3	F83h	PORTD	F63h	PMMODEH	F43h	—
FE2h	FSR1H	FC2h	ADCON0 ⁽³⁾	FA2h	IPR2	F82h	PORTC	F62h	PMMODEL	F42h	—
FE1h	FSR1L	FC1h	ADCON1 ⁽³⁾	FA1h	PIR2	F81h	PORTB	F61h	PMDOUT2H	F41h	—
FE0h	BSR	FC0h	WDTCON	FA0h	PIE2	F80h	PORTA	F60h	PMDOUT2L	F40h	—

- Note**
- 1: This is not a physical register.
 - 2: This register is not available on 64-pin devices.
 - 3: This register shares the same address with another register (see Table 5-4 for alternate register).
 - 4: The PMADDRH/L and PMDOUT1H/L register pairs share the same address. PMADDR is used in Master modes and PMDOUT1 is used in Slave modes.

PIC18F87J11 FAMILY

5.3.4.1 Shared Address SFRs

In several locations in the SFR bank, a single address is used to access two different hardware registers. In these cases, a “legacy” register of the standard PIC18 SFR set (such as OSCCON, T1CON, etc.) shares its address with an alternate register. These alternate registers are associated with enhanced configuration options for peripherals, or with new device features not included in the standard PIC18 SFR map. A complete list of shared register addresses and the registers associated with them is provided in Table 5-4.

Access to the alternate registers is enabled in software by setting the ADSHR bit in the WDTCON register (Register 5-3). ADSHR must be manually set or cleared to access the alternate or legacy registers, as required. Since the bit remains in a given state until changed, users should always verify the state of ADSHR before writing to any of the shared SFR addresses.

5.3.4.2 Context Defined SFRs

In addition to the shared address SFRs, there are several registers that share the same address in the SFR space, but are not accessed with the ADSHR bit. Instead, the register’s definition and use depends on the operating mode of its associated peripheral. These registers are:

- SSPxADD and SSPxMSK: These are two separate hardware registers, accessed through a single SFR address. The operating mode of the MSSP module determines which register is being accessed. See **Section 19.4.3.4 “7-Bit Address Masking Mode”** for additional details.
- PMADDRH/L and PMDOUT2H/L: In this case, these named buffer pairs are actually the same physical registers. The PMP module’s operating mode determines what function the registers take on. See **Section 11.1.2 “Data Registers”** for additional details.

TABLE 5-4: SHARED SFR ADDRESSES FOR PIC18F87J11 FAMILY DEVICES

Address	Name	Address	Name	Address	Name
FD3h	(D) OSCCON	FCDh	T1CON	FC2h	ADCON0
	(A) REFOCON		ODCON3		ANCON1
FCFh	(D) TMR1H	FCCh	TMR2	FC1h	ADCON1
	(A) ODCON1		PADCFG1		ANCON0
FCEh	(D) TMR1L	FCBh	PR2	F77h	PR4
	(A) ODCON2		MEMCON ⁽¹⁾		CVRCON

Legend: (D) = Default SFR, accessible only when ADSHR = 0; (A) = Alternate SFR, accessible only when ADSHR = 1.

Note 1: Implemented in 80-pin devices only.

REGISTER 5-3: WDTCON: WATCHDOG TIMER CONTROL REGISTER

R/W-0	R-x	U-0	R/W-0	U-0	U-0	U-0	U-0
REGSLP	LVDSTAT	—	ADSHR	—	—	—	SWDTEN
bit 7							bit 0

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

bit 7 **REGSLP:** Voltage Regulator Low-Power Operation Enable bit
For details of bit operation, see Register 24-9.

bit 6 **LVDSTAT:** LVD Status bit

1 = VDDCORE > 2.45V

0 = VDDCORE < 2.45V

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **ADSHR:** Shared Address SFR Select bit

1 = Alternate SFR is selected

0 = Default (Legacy) SFR is selected

bit 3-1 **Unimplemented:** Read as ‘0’

bit 0 **SWDTEN:** Software Controlled Watchdog Timer Enable bit

For details of bit operation, see Register 24-9.

TABLE 5-5: REGISTER FILE SUMMARY (PIC18F87J11 FAMILY)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on Page:
TOSU	—	—	—	Top-of-Stack Upper Byte (TOS<20:16>)					---0 0000	55, 65
TOSH	Top-of-Stack High Byte (TOS<15:8>)								0000 0000	55, 65
Tosl	Top-of-Stack Low Byte (TOS<7:0>)								0000 0000	55, 65
STKPTR	STKFUL	STKUNF	—	SP4	SP3	SP2	SP1	SP0	00-0 0000	55, 66
PCLATU	—	—	bit 21 ⁽¹⁾	Holding Register for PC<20:16>					---0 0000	55, 65
PCLATH	Holding Register for PC<15:8>								0000 0000	55, 65
PCL	PC Low Byte (PC<7:0>)								0000 0000	55, 65
TBLPTRU	—	—	bit 21	Program Memory Table Pointer Upper Byte (TBLPTR<20:16>)					--00 0000	55, 96
TBLPTRH	Program Memory Table Pointer High Byte (TBLPTR<15:8>)								0000 0000	55, 96
TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								0000 0000	55, 96
TABLAT	Program Memory Table Latch								0000 0000	55, 96
PRODH	Product Register High Byte								xxxx xxxx	55, 109
PRODL	Product Register Low Byte								xxxx xxxx	55, 109
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	55, 113
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	1111 1111	55, 113
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	1100 0000	55, 113
INDF0	Uses contents of FSR0 to address data memory – value of FSR0 not changed (not a physical register)								N/A	55, 82
POSTINC0	Uses contents of FSR0 to address data memory – value of FSR0 post-incremented (not a physical register)								N/A	55, 83
POSTDEC0	Uses contents of FSR0 to address data memory – value of FSR0 post-decremented (not a physical register)								N/A	55, 83
PREINC0	Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register)								N/A	55, 83
PLUSW0	Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register) – value of FSR0 offset by W								N/A	55, 83
FSR0H	—	—	—	—	Indirect Data Memory Address Pointer 0 High Byte				---- xxxx	55, 82
FSR0L	Indirect Data Memory Address Pointer 0 Low Byte								xxxx xxxx	55, 82
WREG	Working Register								xxxx xxxx	55, 67
INDF1	Uses contents of FSR1 to address data memory – value of FSR1 not changed (not a physical register)								N/A	55, 82
POSTINC1	Uses contents of FSR1 to address data memory – value of FSR1 post-incremented (not a physical register)								N/A	55, 83
POSTDEC1	Uses contents of FSR1 to address data memory – value of FSR1 post-decremented (not a physical register)								N/A	55, 83
PREINC1	Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register)								N/A	55, 83
PLUSW1	Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register) – value of FSR1 offset by W								N/A	55, 83
FSR1H	—	—	—	—	Indirect Data Memory Address Pointer 1 High Byte				---- xxxx	55, 82
FSR1L	Indirect Data Memory Address Pointer 1 Low Byte								xxxx xxxx	55, 82
BSR	—	—	—	—	Bank Select Register				---- 0000	55, 70
INDF2	Uses contents of FSR2 to address data memory – value of FSR2 not changed (not a physical register)								N/A	56, 82
POSTINC2	Uses contents of FSR2 to address data memory – value of FSR2 post-incremented (not a physical register)								N/A	56, 83
POSTDEC2	Uses contents of FSR2 to address data memory – value of FSR2 post-decremented (not a physical register)								N/A	56, 83
PREINC2	Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register)								N/A	56, 83
PLUSW2	Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register) – value of FSR2 offset by W								N/A	56, 83

Legend: x = unknown, u = unchanged, - = unimplemented, q = value depends on condition. **Bold** indicates shared access SFRs.

- Note 1:** Bit 21 of the PC is only available in Serial Programming modes.
- 2:** Default (legacy) SFR at this address, available when WDTCON<4> = 0.
- 3:** Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.
- 4:** Reset value is '0' when Two-Speed Start-up is enabled and '1' if disabled.
- 5:** The SSPxMSK registers are only accessible when SSPxCON2<3:0> = 1001.
- 6:** Alternate names and definitions for these bits when the MSSP modules are operating in I²C™ Slave mode. See **Section 19.4.3.2 "Address Masking Modes"** for details.
- 7:** These bits and/or registers are only available in 80-pin devices; otherwise, they are unimplemented and read as '0'. Reset values are shown for 80-pin devices.
- 8:** These bits are only available in select oscillator modes (FOSC2 Configuration bit = 0); otherwise, they are unimplemented.
- 9:** The PMADDR/PMDOUT1H and PMADDR/PMDOUT1L register pairs share the physical registers and addresses, but have different functions determined by the module's operating mode. See **Section 11.1.2 "Data Registers"** for more information.

PIC18F87J11 FAMILY

TABLE 5-5: REGISTER FILE SUMMARY (PIC18F87J11 FAMILY) (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on Page:
FSR2H	—	—	—	—	Indirect Data Memory Address Pointer 2 High Byte				---- xxxx	56, 82
FSR2L	Indirect Data Memory Address Pointer 2 Low Byte								xxxx xxxx	56, 82
STATUS	—	—	—	N	OV	Z	DC	C	---x xxxx	56, 80
TMR0H	Timer0 Register High Byte								0000 0000	56, 183
TMR0L	Timer0 Register Low Byte								xxxx xxxx	56, 183
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111	56, 182
OSCCON⁽²⁾/ REFOCON⁽³⁾	IDLEN	IRCF2	IRCF1	IRCF0	OSTS ⁽⁴⁾	—	SCS1	SCS0	0110 q100	56, 32
	ROON	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0	0-00 0000	56, 39
CM1CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	0001 1111	56, 302
CM2CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	0001 1111	56, 302
RCON	IPEN	—	CM	RI	TO	PD	POR	BOR	0-11 1100	54, 56, 125
TMR1H⁽²⁾/ ODCON1⁽³⁾	Timer1 Register High Byte								xxxx xxxx	56, 186
	—	—	—	CCP5OD	CCP4OD	ECCP3OD	ECCP2OD	ECCP1OD	--0 0000	56, 129
TMR1L⁽²⁾/ ODCON2⁽³⁾	Timer1 Register Low Byte								xxxx xxxx	56, 186
	—	—	—	—	—	—	U2OD	U1OD	---- --00	56, 129
T1CON⁽²⁾/ ODCON3⁽³⁾	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000	56, 186
	—	—	—	—	—	—	SPI2OD	SPI1OD	---- --00	56, 129
TMR2⁽²⁾/ PADCFG1⁽³⁾	Timer2 Register								0000 0000	56, 191
	—	—	—	—	—	—	—	PMPTTL	---- --0	56, 130
PR2⁽²⁾/ MEMCON^(3,7)	Timer2 Period Register								1111 1111	56, 191
	EDBIS	—	WAIT1	WAIT0	—	—	WM1	WM0	0-00 --00	56, 98
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	56, 191
SSP1BUF	MSSP1 Receive Buffer/Transmit Register								xxxx xxxx	56, 224, 233
SSP1ADD/ SSP1MSK⁽⁵⁾	MSSP1 Address Register (I ² C™ Slave mode), MSSP1 Baud Rate Reload Register (I ² C Master mode)								0000 0000	56, 233
	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	0000 0000	56, 240
SSP1STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	56, 224, 234
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	56, 225, 235
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN/	SEN	0000 0000	56, 236, 268
	GCEN	ACKSTAT	ADMSK5 ⁽⁶⁾	ADMSK4 ⁽⁶⁾	ADMSK3 ⁽⁶⁾	ADMSK2 ⁽⁶⁾	ADMSK1 ⁽⁶⁾	SEN		
ADRESH	A/D Result Register High Byte								xxxx xxxx	57, 291
ADRESL	A/D Result Register Low Byte								xxxx xxxx	57, 291
ADCON0⁽²⁾/ ANCON1⁽³⁾	VCFG1	VCFG0	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	0000 0000	57, 291
	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	0000 0000	57, 293
ADCON1⁽²⁾/ ANCON0⁽³⁾	ADFM	ADCAL	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0000 0000	57, 292
	PCFG7	PCFG6	—	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	00-0 0000	57, 293
WDTCON	REGSLP	LVDSTAT	—	ADSHR	—	—	—	SWDTEN	0x-0 ---0	57, 321

Legend: x = unknown, u = unchanged, - = unimplemented, q = value depends on condition. **Bold** indicates shared access SFRs.

- Note 1:** Bit 21 of the PC is only available in Serial Programming modes.
- 2:** Default (legacy) SFR at this address, available when WDTCON<4> = 0.
- 3:** Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.
- 4:** Reset value is '0' when Two-Speed Start-up is enabled and '1' if disabled.
- 5:** The SSPxMSK registers are only accessible when SSPxCON<3:0> = 1001.
- 6:** Alternate names and definitions for these bits when the MSSP modules are operating in I²C™ Slave mode. See **Section 19.4.3.2 "Address Masking Modes"** for details.
- 7:** These bits and/or registers are only available in 80-pin devices; otherwise, they are unimplemented and read as '0'. Reset values are shown for 80-pin devices.
- 8:** These bits are only available in select oscillator modes (FOSC2 Configuration bit = 0); otherwise, they are unimplemented.
- 9:** The PMADDRH/PMDOUT1H and PMADDRL/PMDOUT1L register pairs share the physical registers and addresses, but have different functions determined by the module's operating mode. See **Section 11.1.2 "Data Registers"** for more information.

PIC18F87J11 FAMILY

TABLE 5-5: REGISTER FILE SUMMARY (PIC18F87J11 FAMILY) (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on Page:
ECCP1AS	ECCP1ASE	ECCP1AS2	ECCP1AS1	ECCP1AS0	PSS1AC1	PSS1AC0	PSS1BD1	PSS1BD0	0000 0000	57, 222
ECCP1DEL	P1RSEN	P1DC6	P1DC5	P1DC4	P1DC3	P1DC2	P1DC1	P1DC0	0000 0000	57, 222
CCPR1H	Capture/Compare/PWM Register 1 High Byte								xxxxx xxxx	57, 222
CCPR1L	Capture/Compare/PWM Register 1 Low Byte								xxxxx xxxx	57, 222
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	57, 222
ECCP2AS	ECCP2ASE	ECCP2AS2	ECCP2AS1	ECCP2AS0	PSS2AC1	PSS2AC0	PSS2BD1	PSS2BD0	0000 0000	57, 222
ECCP2DEL	P2RSEN	P2DC6	P2DC5	P2DC4	P2DC3	P2DC2	P2DC1	P2DC0	0000 0000	57, 222
CCPR2H	Capture/Compare/PWM Register 2 High Byte								xxxxx xxxx	57, 222
CCPR2L	Capture/Compare/PWM Register 2 Low Byte								xxxxx xxxx	57, 222
CCP2CON	P2M1	P2M0	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	0000 0000	57, 222
ECCP3AS	ECCP3ASE	ECCP3AS2	ECCP3AS1	ECCP3AS0	PSS3AC1	PSS3AC0	PSS3BD1	PSS3BD0	0000 0000	57, 222
ECCP3DEL	P3RSEN	P3DC6	P3DC5	P3DC4	P3DC3	P3DC2	P3DC1	P3DC0	0000 0000	57, 222
CCPR3H	Capture/Compare/PWM Register 1 High Byte								xxxxx xxxx	57, 222
CCPR3L	Capture/Compare/PWM Register 1 Low Byte								xxxxx xxxx	57, 222
CCP3CON	P3M1	P3M0	DC3B1	DC3B0	CCP3M3	CCP3M2	CCP3M1	CCP3M0	0000 0000	57, 222
SPBRG1	EUSART1 Baud Rate Generator Register Low Byte								0000 0000	57, 273
RCREG1	EUSART1 Receive Register								0000 0000	57, 281, 282
TXREG1	EUSART1 Transmit Register								xxxxx xxxx	57, 279, 280
TXSTA1	CSRC	TX9	TXEN	SYNC	SENDDB	BRGH	TRMT	TX9D	0000 0010	57, 279
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	57, 281
SPBRG2	EUSART2 Baud Rate Generator Register Low Byte								0000 0000	57, 273
RCREG2	EUSART2 Receive Register								0000 0000	57, 281, 282
TXREG2	EUSART2 Transmit Register								0000 0000	57, 279, 280
TXSTA2	CSRC	TX9	TXEN	SYNC	SENDDB	BRGH	TRMT	TX9D	0000 0010	57, 279
EECON2	Program Memory Control Register 2 (not a physical register)								-----	57, 88
EECON1	—	—	WPROG	FREE	WRERR	WREN	WR	—	--00 x00-	57, 88
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	1111 1111	58, 122
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	0000 0000	58, 116
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	0000 0000	58, 119
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	111- 1111	58, 122
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	000- 0000	58, 116
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	000- 0000	58, 119
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	1111 1111	58, 122
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	58, 116
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	0000 0000	58, 119
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	58, 281
OSCTUNE	INTSRC	PLLEN	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	0000 0000	58, 33

Legend: x = unknown, u = unchanged, - = unimplemented, q = value depends on condition. **Bold** indicates shared access SFRs.

Note 1: Bit 21 of the PC is only available in Serial Programming modes.

2: Default (legacy) SFR at this address, available when WDTCON<4> = 0.

3: Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.

4: Reset value is '0' when Two-Speed Start-up is enabled and '1' if disabled.

5: The SSPxMSK registers are only accessible when SSPxCON2<3:0> = 1001.

6: Alternate names and definitions for these bits when the MSSP modules are operating in I²C™ Slave mode. See **Section 19.4.3.2 "Address Masking Modes"** for details.

7: These bits and/or registers are only available in 80-pin devices; otherwise, they are unimplemented and read as '0'. Reset values are shown for 80-pin devices.

8: These bits are only available in select oscillator modes (FOSC2 Configuration bit = 0); otherwise, they are unimplemented.

9: The PMADDRH/PMDOUT1H and PMADDRL/PMDOUT1L register pairs share the physical registers and addresses, but have different functions determined by the module's operating mode. See **Section 11.1.2 "Data Registers"** for more information.

PIC18F87J11 FAMILY

TABLE 5-5: REGISTER FILE SUMMARY (PIC18F87J11 FAMILY) (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on Page:
TRISJ ⁽⁷⁾	TRISJ7	TRISJ6	TRISJ5	TRISJ4	TRISJ3	TRISJ2	TRISJ1	TRISJ0	1111 1111	58, 155
TRISH ⁽⁷⁾	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	1111 1111	58, 153
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	---1 1111	58, 150
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	1111 111-	58, 147
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	1111 1111	58, 144
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	58, 141
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISCO	1111 1111	58, 138
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	58, 135
TRISA	TRISA7 ⁽⁸⁾	TRISA6 ⁽⁸⁾	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	58, 132
LATJ ⁽⁷⁾	LATJ7	LATJ6	LATJ5	LATJ4	LATJ3	LATJ2	LATJ1	LATJ0	xxxx xxxx	58, 155
LATH ⁽⁷⁾	LATH7	LATH6	LATH5	LATH4	LATH3	LATH2	LATH1	LATH0	xxxx xxxx	58, 153
LATG	—	—	—	LATG4	LATG3	LATG2	LATG1	LATG0	- -x xxxx	58, 150
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	—	xxxx xxxx-	58, 147
LATE	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	xxxx xxxx	58, 144
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	xxxx xxxx	58, 141
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx xxxx	58, 138
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx xxxx	58, 135
LATA	LATA7 ⁽⁸⁾	LATA6 ⁽⁸⁾	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx xxxx	58, 132
PORTJ ⁽⁷⁾	RJ7	RJ6	RJ5	RJ4	RJ3	RJ2	RJ1	RJ0	xxxx xxxx	59, 155
PORTH ⁽⁷⁾	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0	0000 xxxx	59, 153
PORTG	RDPU	REPU	RJPU ⁽⁷⁾	RG4	RG3	RG2	RG1	RG0	000x xxxx	59, 150
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	—	x000 000-	59, 147
PORTE	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	xxxx xxxx	59, 144
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	59, 141
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	59, 138
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	59, 135
PORTA	RA7 ⁽⁸⁾	RA6 ⁽⁸⁾	RA5	RA4	RA3	RA2	RA1	RA0	000x 0000	59, 132
SPBRGH1	EUSART1 Baud Rate Generator Register High Byte								0000 0000	59, 273
BAUDCON1	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	0100 0-00	59, 273
SPBRGH2	EUSART2 Baud Rate Generator Register High Byte								0000 0000	59, 273
BAUDCON2	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	0100 0-00	59, 273
TMR3H	Timer3 Register High Byte								xxxx xxxx	59, 198
TMR3L	Timer3 Register Low Byte								xxxx xxxx	59, 198
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON	0000 0000	59, 198
TMR4	Timer4 Register								0000 0000	59, 197
PR4⁽²⁾/CVRCON⁽³⁾	Timer4 Period Register								1111 1111	59, 198
	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000 0000	59, 310
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	-000 0000	59, 197

Legend: x = unknown, u = unchanged, - = unimplemented, q = value depends on condition. **Bold** indicates shared access SFRs.

Note 1: Bit 21 of the PC is only available in Serial Programming modes.

2: Default (legacy) SFR at this address, available when WDTCON<4> = 0.

3: Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.

4: Reset value is '0' when Two-Speed Start-up is enabled and '1' if disabled.

5: The SSPxMSK registers are only accessible when SSPxCON2<3:0> = 1001.

6: Alternate names and definitions for these bits when the MSSP modules are operating in I²C™ Slave mode. See **Section 19.4.3.2 "Address Masking Modes"** for details

7: These bits and/or registers are only available in 80-pin devices; otherwise, they are unimplemented and read as '0'. Reset values are shown for 80-pin devices.

8: These bits are only available in select oscillator modes (FOSC2 Configuration bit = 0); otherwise, they are unimplemented.

9: The PMADDRH/PMDOUT1H and PMADDRL/PMDOUT1L register pairs share the physical registers and addresses, but have different functions determined by the module's operating mode. See **Section 11.1.2 "Data Registers"** for more information.

PIC18F87J11 FAMILY

TABLE 5-5: REGISTER FILE SUMMARY (PIC18F87J11 FAMILY) (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on Page:						
CCPR4H	Capture/Compare/PWM Register 4 High Byte								xxxx xxxx	59, 200						
CCPR4L	Capture/Compare/PWM Register 4 Low Byte								xxxx xxxx	59, 200						
CCP4CON	—	—	DC4B1	DC4B0	CCP4M3	CCP4M2	CCP4M1	CCP4M0	--00 0000	59, 200						
CCPR5H	Capture/Compare/PWM Register 5 High Byte								xxxx xxxx	59, 200						
CCPR5L	Capture/Compare/PWM Register 5 Low Byte								xxxx xxxx	59, 200						
CCP5CON	—	—	DC5B1	DC5B0	CCP5M3	CCP5M2	CCP5M1	CCP5M0	--00 0000	59, 200						
SSP2BUF	MSSP2 Receive Buffer/Transmit Register								xxxx xxxx	59, 224, 233						
SSP2ADD/ SSP2MSK ⁽⁵⁾	MSSP2 Address Register (I ² C™ Slave mode), MSSP2 Baud Rate Reload Register (I ² C Master mode)								0000 0000	59, 233						
	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	0000 0000	59, 240						
SSP2STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	59, 224, 234						
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	59, 225, 235						
SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN/	SEN	0000 0000	59, 236, 268						
	GCEN	ACKSTAT	ADMSK5 ⁽⁶⁾	ADMSK4 ⁽⁶⁾	ADMSK3 ⁽⁶⁾	ADMSK2 ⁽⁶⁾	ADMSK1 ⁽⁶⁾	SEN								
CMSTAT	—	—	—	—	—	—	COUT2	COUT1	---- --11	59, 303						
PMADDRH / PMDOUT1H ⁽⁹⁾	CS2	CS1	Parallel Master Port Address, High Byte						0000 0000	60, 164						
PMADDRL/ PMDOUT1L ⁽⁹⁾	Parallel Port Out Data High Byte (Buffer 1)								0000 0000	60, 167						
	Parallel Master Port Address Low Byte								0000 0000	60, 164						
	Parallel Port Out Data Low Byte (Buffer 0)								0000 0000	60, 164						
PMDIN1H	Parallel Port In Data High Byte (Buffer 1)								0000 0000	60, 164						
PMDIN1L	Parallel Port In Data Low Byte (Buffer 0)								0000 0000	60, 164						
PMCONH	PMPPEN	—	PSIDL	ADRMUX1	ADRMUX0	PTBEEN	PTWREN	PTRDEN	0-00 0000	60, 158						
PMCONL	CSF1	CSF0	ALP	CS2P	CS1P	BEP	WRSP	RDSP	0000 0000	60, 159						
PMODEEH	BUSY	IRQM1	IRQMO	INCM1	INCM0	MODE16	MODE1	MODE0	0000 0000	60, 160						
PMODEML	WAITB1	WAITB0	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1	WAITE0	0000 0000	60, 161						
PMDOUT2H	Parallel Port Out Data High Byte (Buffer 3)								0000 0000	60, 164						
PMDOUT2L	Parallel Port Out Data Low Byte (Buffer 2)								0000 0000	60, 164						
PMDIN2H	Parallel Port In Data High Byte (Buffer 3)								0000 0000	60, 164						
PMDIN2L	Parallel Port In Data Low Byte (Buffer 2)								0000 0000	60, 164						
PMEH	PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8	0000 0000	60, 161						
PMEL	PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0	0000 0000	60, 162						
PMSTATH	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F	00-- 0000	60, 162						
PMSTATL	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E	10-- 1111	60, 163						

Legend: x = unknown, u = unchanged, - = unimplemented, q = value depends on condition. **Bold** indicates shared access SFRs.

Note 1: Bit 21 of the PC is only available in Serial Programming modes.

2: Default (legacy) SFR at this address, available when WDTCON<4> = 0.

3: Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.

4: Reset value is '0' when Two-Speed Start-up is enabled and '1' if disabled.

5: The SSPxMSK registers are only accessible when SSPxCON2<3:0> = 1001.

6: Alternate names and definitions for these bits when the MSSP modules are operating in I²C™ Slave mode. See **Section 19.4.3.2 "Address Masking Modes"** for details.

7: These bits and/or registers are only available in 80-pin devices; otherwise, they are unimplemented and read as '0'. Reset values are shown for 80-pin devices.

8: These bits are only available in select oscillator modes (FOSC2 Configuration bit = 0); otherwise, they are unimplemented.

9: The PMADDRH/PMDOUT1H and PMADDRL/PMDOUT1L register pairs share the physical registers and addresses, but have different functions determined by the module's operating mode. See **Section 11.1.2 "Data Registers"** for more information.

PIC18F87J11 FAMILY

5.3.5 STATUS REGISTER

The STATUS register, shown in Register 5-4, contains the arithmetic status of the ALU. The STATUS register can be the operand for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC, C, OV or N bits, then the write to these five bits is disabled.

These bits are set or cleared according to the device logic. Therefore, the result of an instruction with the STATUS register as destination may be different than intended. For example, CLRF STATUS will set the Z bit but leave the other bits unchanged. The STATUS register then reads back as '000u uuu'. It is

recommended, therefore, that only BCF, BSF, SWAPF, MOVFF and MOVWF instructions are used to alter the STATUS register because these instructions do not affect the Z, C, DC, OV or N bits in the STATUS register.

For other instructions not affecting any Status bits, see the instruction set summaries in Table 25-2 and Table 25-3.

Note: The C and DC bits operate as a borrow and digit borrow bit respectively, in subtraction.

REGISTER 5-4: STATUS REGISTER

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	N	OV	Z	DC ⁽¹⁾	C ⁽²⁾
bit 7							bit 0

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

- bit 7-5 **Unimplemented:** Read as '0'
- bit 4 **N:** Negative bit
This bit is used for signed arithmetic (2's complement). It indicates whether the result was negative (ALU MSB = 1).
1 = Result was negative
0 = Result was positive
- bit 3 **OV:** Overflow bit
This bit is used for signed arithmetic (2's complement). It indicates an overflow of the 7-bit magnitude which causes the sign bit (bit 7) to change state.
1 = Overflow occurred for signed arithmetic (in this arithmetic operation)
0 = No overflow occurred
- 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⁽¹⁾
For ADDWF, ADDLW, SUBLW and SUBWF instructions:
1 = A carry-out from the 4th low-order bit of the result occurred
0 = No carry-out from the 4th low-order bit of the result
- bit 0 **C:** Carry/borrow bit⁽²⁾
For ADDWF, ADDLW, SUBLW and 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 1:** For borrow, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either bit 4 or bit 3 of the source register.
- 2:** For borrow, the polarity is reversed. A subtraction is executed by adding the 2'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.

5.4 Data Addressing Modes

Note: The execution of some instructions in the core PIC18 instruction set are changed when the PIC18 extended instruction set is enabled. See **Section 5.6 “Data Memory and the Extended Instruction Set”** for more information.

While the program memory can be addressed in only one way – through the program counter – information in the data memory space can be addressed in several ways. For most instructions, the addressing mode is fixed. Other instructions may use up to three modes, depending on which operands are used and whether or not the extended instruction set is enabled.

The addressing modes are:

- Inherent
 - Literal
 - Direct
 - Indirect

An additional addressing mode, Indexed Literal Offset, is available when the extended instruction set is enabled (XINST Configuration bit = 1). Its operation is discussed in greater detail in **Section 5.6.1 “Indexed Addressing with Literal Offset”**.

5.4.1 INHERENT AND LITERAL ADDRESSING

Many PIC18 control instructions do not need any argument at all; they either perform an operation that globally affects the device, or they operate implicitly on one register. This addressing mode is known as Inherent Addressing. Examples include SLEEP, RESET and DAW.

Other instructions work in a similar way, but require an additional explicit argument in the opcode. This is known as Literal Addressing mode, because they require some literal value as an argument. Examples include ADDLW and MOVLW, which respectively, add or move a literal value to the W register. Other examples include CALL and GOTO, which include a 20-bit program memory address.

5.4.2 DIRECT ADDRESSING

Direct Addressing specifies all or part of the source and/or destination address of the operation within the opcode itself. The options are specified by the arguments accompanying the instruction.

In the core PIC18 instruction set, bit-oriented and byte-oriented instructions use some version of Direct Addressing by default. All of these instructions include some 8-bit Literal Address as their Least Significant Byte. This address specifies either a register address in one of the banks of data RAM (**Section 5.3.3 “General**

Purpose Register File”), or a location in the Access Bank (**Section 5.3.2 “Access Bank”**) as the data source for the instruction.

The Access RAM bit ‘a’ determines how the address is interpreted. When ‘a’ is ‘1’, the contents of the BSR (**Section 5.3.1 “Bank Select Register”**) are used with the address to determine the complete 12-bit address of the register. When ‘a’ is ‘0’, the address is interpreted as being a register in the Access Bank. Addressing that uses the Access RAM is sometimes also known as Direct Forced Addressing mode.

A few instructions, such as MOVFF, include the entire 12-bit address (either source or destination) in their opcodes. In these cases, the BSR is ignored entirely.

The destination of the operation's results is determined by the destination bit 'd'. When 'd' is '1', the results are stored back in the source register, overwriting its original contents. When 'd' is '0', the results are stored in the W register. Instructions without the 'd' argument have a destination that is implicit in the instruction; their destination is either the target register being operated on or the W register.

5.4.3 INDIRECT ADDRESSING

Indirect Addressing allows the user to access a location in data memory without giving a fixed address in the instruction. This is done by using File Select Registers (FSRs) as pointers to the locations to be read or written to. Since the FSRs are themselves located in RAM as Special Function Registers, they can also be directly manipulated under program control. This makes FSRs very useful in implementing data structures such as tables and arrays in data memory.

The registers for Indirect Addressing are also implemented with Indirect File Operands (INDFs) that permit automatic manipulation of the pointer value with auto-incrementing, auto-decrementing or offsetting with another value. This allows for efficient code using loops, such as the example of clearing an entire RAM bank in Example 5-5. It also enables users to perform Indexed Addressing and other Stack Pointer operations for program memory in data memory.

EXAMPLE 5-5: HOW TO CLEAR RAM (BANK 1) USING INDIRECT ADDRESSING

	LFSR	FSR0, 100h ;	
NEXT	CLRFL	POSTINC0 ;	Clear INDF
			; register then
			; inc pointer
	BTFS	FSROH, 1 ;	All done with
			; Bank1?
	BRA	NEXT ;	NO, clear next
CONTINUE			; YES, continue

PIC18F87J11 FAMILY

5.4.3.1 FSR Registers and the INDF Operand

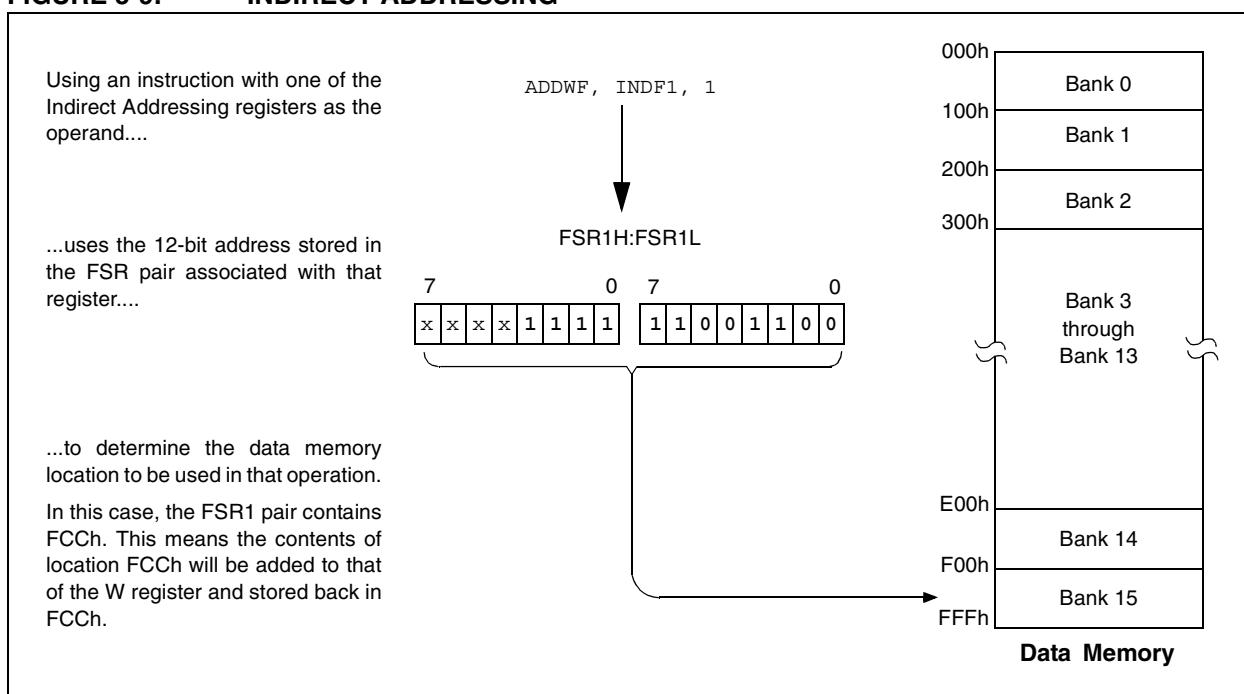
At the core of Indirect Addressing are three sets of registers: FSR0, FSR1 and FSR2. Each represents a pair of 8-bit registers, FSRnH and FSRnL. The four upper bits of the FSRnH register are not used, so each FSR pair holds a 12-bit value. This represents a value that can address the entire range of the data memory in a linear fashion. The FSR register pairs, then, serve as pointers to data memory locations.

Indirect Addressing is accomplished with a set of Indirect File Operands, INDF0 through INDF2. These can be thought of as “virtual” registers: they are mapped in

the SFR space but are not physically implemented. Reading or writing to a particular INDF register actually accesses its corresponding FSR register pair. A read from INDF1, for example, reads the data at the address indicated by FSR1H:FSR1L. Instructions that use the INDF registers as operands actually use the contents of their corresponding FSR as a pointer to the instruction’s target. The INDF operand is just a convenient way of using the pointer.

Because Indirect Addressing uses a full 12-bit address, data RAM banking is not necessary. Thus, the current contents of the BSR and the Access RAM bit have no effect on determining the target address.

FIGURE 5-9: INDIRECT ADDRESSING



5.4.3.2 FSR Registers and POSTINC, POSTDEC, PREINC and PLUSW

In addition to the INDF operand, each FSR register pair also has four additional indirect operands. Like INDF, these are “virtual” registers that cannot be indirectly read or written to. Accessing these registers actually accesses the associated FSR register pair, but also performs a specific action on its stored value. They are:

- POSTDEC: accesses the FSR value, then automatically decrements it by ‘1’ afterwards
- POSTINC: accesses the FSR value, then automatically increments it by ‘1’ afterwards
- PREINC: increments the FSR value by ‘1’, then uses it in the operation
- PLUSW: adds the signed value of the W register (range of -127 to 128) to that of the FSR and uses the new value in the operation

In this context, accessing an INDF register uses the value in the FSR registers without changing them. Similarly, accessing a PLUSW register gives the FSR value offset by the value in the W register; neither value is actually changed in the operation. Accessing the other virtual registers changes the value of the FSR registers.

Operations on the FSRs with POSTDEC, POSTINC and PREINC affect the entire register pair; that is, roll-overs of the FSRnL register from FFh to 00h carry over to the FSRnH register. On the other hand, results of these operations do not change the value of any flags in the STATUS register (e.g., Z, N, OV, etc.).

The PLUSW register can be used to implement a form of Indexed Addressing in the data memory space. By manipulating the value in the W register, users can reach addresses that are fixed offsets from pointer addresses. In some applications, this can be used to implement some powerful program control structure, such as software stacks, inside of data memory.

5.4.3.3 Operations by FSRs on FSRs

Indirect Addressing operations that target other FSRs or virtual registers represent special cases. For example, using an FSR to point to one of the virtual registers will not result in successful operations. As a specific case, assume that FSR0H:FSR0L contains FE7h, the address of INDF1. Attempts to read the value of the INDF1, using INDF0 as an operand, will return 00h. Attempts to write to INDF1, using INDF0 as the operand, will result in a NOP.

On the other hand, using the virtual registers to write to an FSR pair may not occur as planned. In these cases, the value will be written to the FSR pair but without any incrementing or decrementing. Thus, writing to INDF2 or POSTDEC2 will write the same value to the FSR2H:FSR2L.

Since the FSRs are physical registers mapped in the SFR space, they can be manipulated through all direct operations. Users should proceed cautiously when working on these registers, particularly if their code uses Indirect Addressing.

Similarly, operations by Indirect Addressing are generally permitted on all other SFRs. Users should exercise the appropriate caution that they do not inadvertently change settings that might affect the operation of the device.

5.5 Program Memory and the Extended Instruction Set

The operation of program memory is unaffected by the use of the extended instruction set.

Enabling the extended instruction set adds five additional two-word commands to the existing PIC18 instruction set: ADDFSR, CALLW, MOVSF, MOVSS and SUBFSR. These instructions are executed as described in **Section 5.2.4 “Two-Word Instructions”**.

5.6 Data Memory and the Extended Instruction Set

Enabling the PIC18 extended instruction set (XINST Configuration bit = 1) significantly changes certain aspects of data memory and its addressing. Specifically, the use of the Access Bank for many of the core PIC18 instructions is different. This is due to the introduction of a new addressing mode for the data memory space. This mode also alters the behavior of Indirect Addressing using FSR2 and its associated operands.

What does not change is just as important. The size of the data memory space is unchanged, as well as its linear addressing. The SFR map remains the same. Core PIC18 instructions can still operate in both Direct and Indirect Addressing mode; inherent and literal instructions do not change at all. Indirect Addressing with FSR0 and FSR1 also remains unchanged.

5.6.1 INDEXED ADDRESSING WITH LITERAL OFFSET

Enabling the PIC18 extended instruction set changes the behavior of Indirect Addressing using the FSR2 register pair and its associated file operands. Under the proper conditions, instructions that use the Access Bank – that is, most bit-oriented and byte-oriented instructions – can invoke a form of Indexed Addressing using an offset specified in the instruction. This special addressing mode is known as Indexed Addressing with Literal Offset, or Indexed Literal Offset mode.

When using the extended instruction set, this addressing mode requires the following:

- The use of the Access Bank is forced ('a' = 0); and
- The file address argument is less than or equal to 5Fh.

Under these conditions, the file address of the instruction is not interpreted as the lower byte of an address (used with the BSR in Direct Addressing) or as an 8-bit address in the Access Bank. Instead, the value is interpreted as an offset value to an Address Pointer specified by FSR2. The offset and the contents of FSR2 are added to obtain the target address of the operation.

5.6.2 INSTRUCTIONS AFFECTED BY INDEXED LITERAL OFFSET MODE

Any of the core PIC18 instructions that can use Direct Addressing are potentially affected by the Indexed Literal Offset Addressing mode. This includes all byte-oriented and bit-oriented instructions, or almost one-half of the standard PIC18 instruction set. Instructions that only use Inherent or Literal Addressing modes are unaffected.

Additionally, byte-oriented and bit-oriented instructions are not affected if they use the Access Bank (Access RAM bit is '1') or include a file address of 60h or above. Instructions meeting these criteria will continue to execute as before. A comparison of the different possible addressing modes when the extended instruction set is enabled is shown in Figure 5-10.

Those who desire to use byte-oriented or bit-oriented instructions in the Indexed Literal Offset mode should note the changes to assembler syntax for this mode. This is described in more detail in **Section 25.2.1 “Extended Instruction Syntax”**.

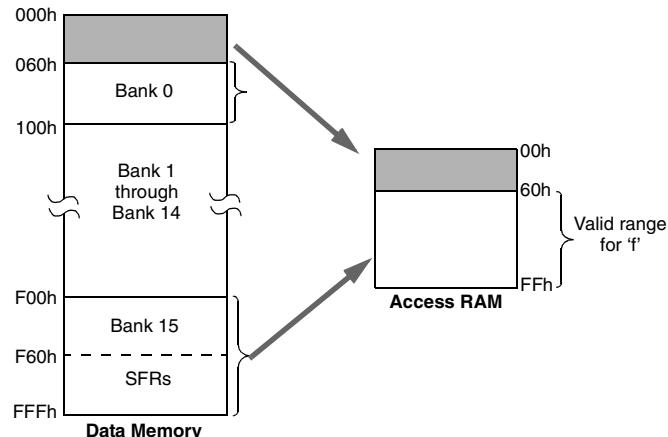
FIGURE 5-10: COMPARING ADDRESSING OPTIONS FOR BIT-ORIENTED AND BYTE-ORIENTED INSTRUCTIONS (EXTENDED INSTRUCTION SET ENABLED)

EXAMPLE INSTRUCTION: ADDWF, f, d, a (Opcode: 0010 01da ffff ffff)

When a = 0 and f \geq 60h:

The instruction executes in Direct Forced mode. 'f' is interpreted as a location in the Access RAM between 060h and FFFh. This is the same as locations F60h to FFFh (Bank 15) of data memory.

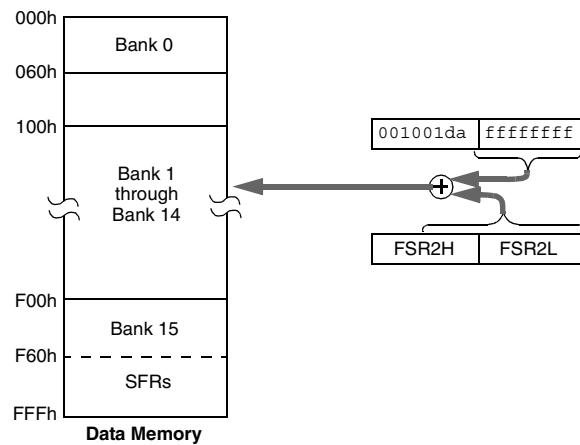
Locations below 060h are not available in this addressing mode.



When a = 0 and f \leq 5Fh:

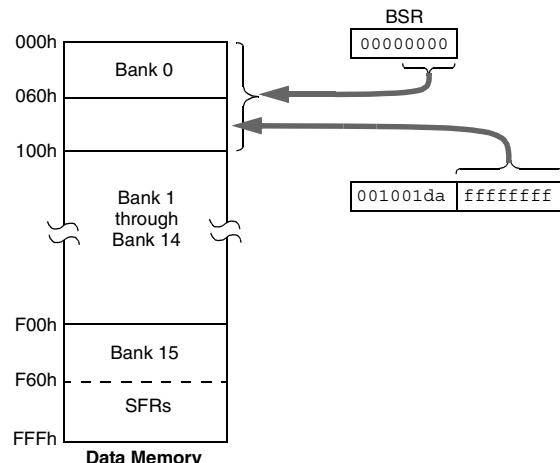
The instruction executes in Indexed Literal Offset mode. 'f' is interpreted as an offset to the address value in FSR2. The two are added together to obtain the address of the target register for the instruction. The address can be anywhere in the data memory space.

Note that in this mode, the correct syntax is now:
ADDWF [k], d
where 'k' is the same as 'f'.



When a = 1 (all values of f):

The instruction executes in Direct mode (also known as Direct Long mode). 'f' is interpreted as a location in one of the 16 banks of the data memory space. The bank is designated by the Bank Select Register (BSR). The address can be in any implemented bank in the data memory space.



PIC18F87J11 FAMILY

5.6.3 MAPPING THE ACCESS BANK IN INDEXED LITERAL OFFSET MODE

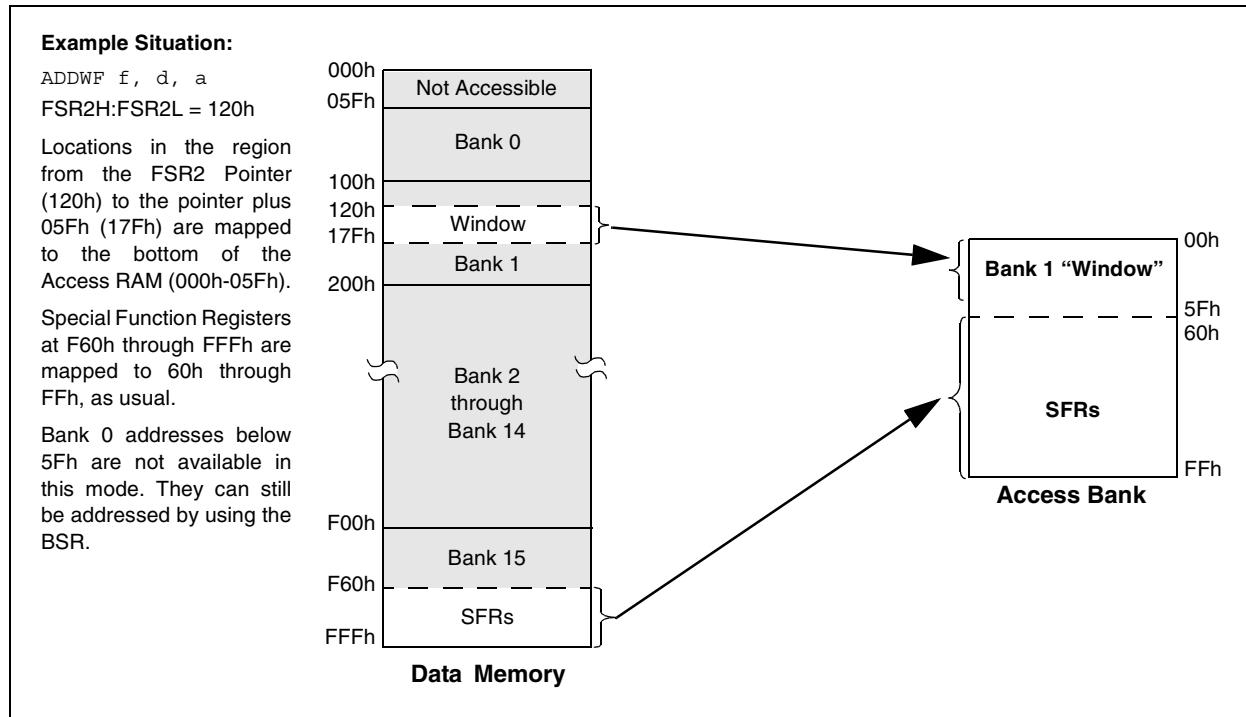
The use of Indexed Literal Offset Addressing mode effectively changes how the lower part of Access RAM (00h to 5Fh) is mapped. Rather than containing just the contents of the bottom part of Bank 0, this mode maps the contents from Bank 0 and a user-defined “window” that can be located anywhere in the data memory space. The value of FSR2 establishes the lower boundary of the addresses mapped into the window, while the upper boundary is defined by FSR2 plus 95 (5Fh). Addresses in the Access RAM above 5Fh are mapped as previously described (see **Section 5.3.2 “Access Bank”**). An example of Access Bank remapping in this addressing mode is shown in Figure 5-11.

Remapping of the Access Bank applies *only* to operations using the Indexed Literal Offset mode. Operations that use the BSR (Access RAM bit is ‘1’) will continue to use Direct Addressing as before. Any Indirect or Indexed Addressing operation that explicitly uses any of the indirect file operands (including FSR2) will continue to operate as standard Indirect Addressing. Any instruction that uses the Access Bank, but includes a register address of greater than 05Fh, will use Direct Addressing and the normal Access Bank map.

5.6.4 BSR IN INDEXED LITERAL OFFSET MODE

Although the Access Bank is remapped when the extended instruction set is enabled, the operation of the BSR remains unchanged. Direct Addressing, using the BSR to select the data memory bank, operates in the same manner as previously described.

FIGURE 5-11: REMAPPING THE ACCESS BANK WITH INDEXED LITERAL OFFSET ADDRESSING



6.0 FLASH PROGRAM MEMORY

The Flash program memory is readable, writable and erasable during normal operation over the entire V_{DD} range.

A read from program memory is executed on one byte at a time. A write to program memory is executed on blocks of 64 bytes at a time or two bytes at a time. Program memory is erased in blocks of 1024 bytes at a time. A bulk erase operation may not be issued from user code.

Writing or erasing program memory will cease instruction fetches until the operation is complete. The program memory cannot be accessed during the write or erase, therefore, code cannot execute. An internal programming timer terminates program memory writes and erases.

A value written to program memory does not need to be a valid instruction. Executing a program memory location that forms an invalid instruction results in a NOP.

6.1 Table Reads and Table Writes

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)

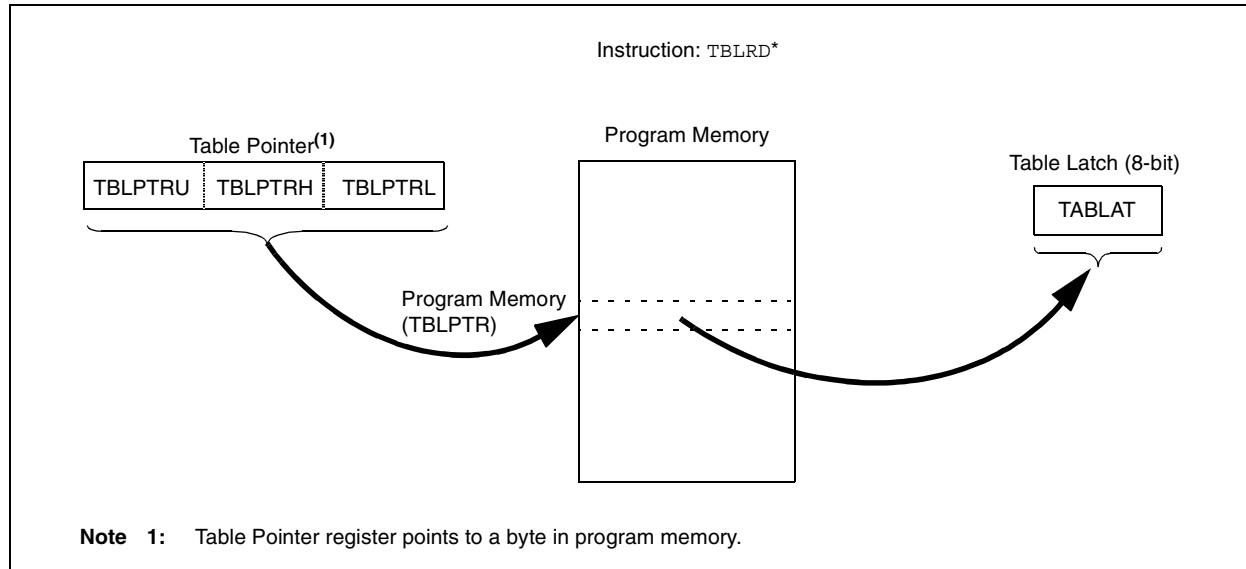
The program memory space is 16 bits wide, while the data RAM space is 8 bits wide. Table reads and table writes move data between these two memory spaces through an 8-bit register (TABLAT).

Table read operations retrieve data from program memory and place it into the data RAM space. Figure 6-1 shows the operation of a table read with program memory and data RAM.

Table write operations store data from the data memory space into holding registers in program memory. The procedure to write the contents of the holding registers into program memory is detailed in **Section 6.5 “Writing to Flash Program Memory”**. Figure 6-2 shows the operation of a table write with program memory and data RAM.

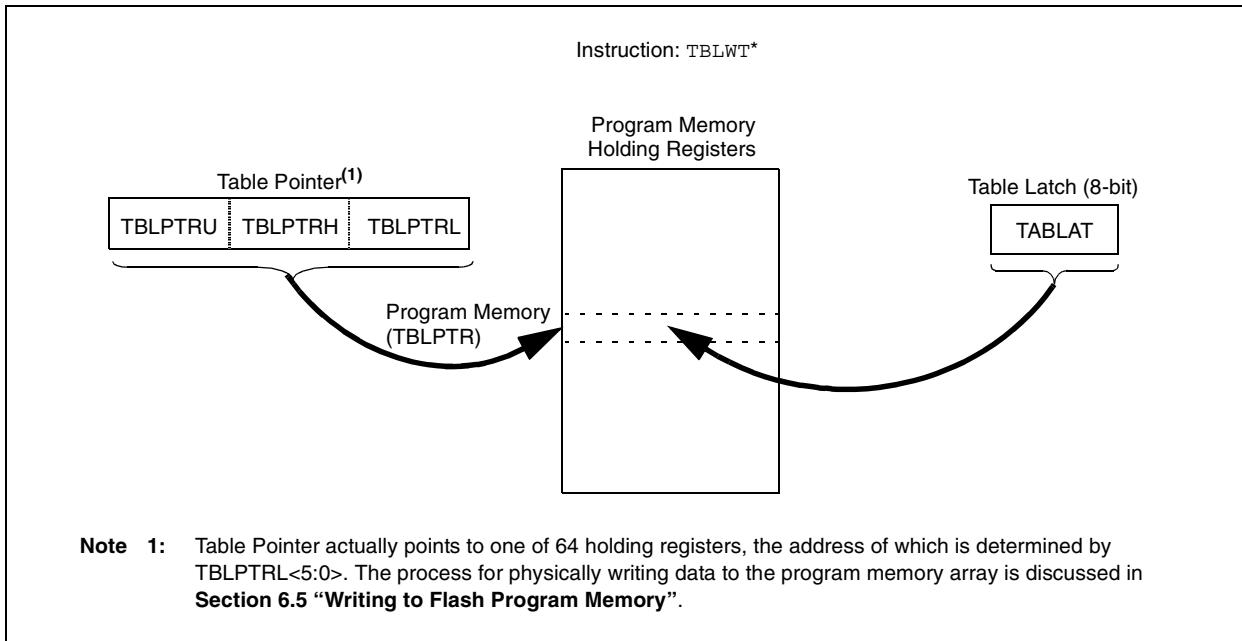
Table operations work with byte entities. A table block containing data, rather than program instructions, is not required to be word-aligned. Therefore, a table block can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions will need to be word-aligned.

FIGURE 6-1: TABLE READ OPERATION



PIC18F87J11 FAMILY

FIGURE 6-2: TABLE WRITE OPERATION



6.2 Control Registers

Several control registers are used in conjunction with the TBLRD and TBLWT instructions. These include the:

- EECON1 register
- EECON2 register
- TABLAT register
- TBLPTR registers

6.2.1 EECON1 AND EECON2 REGISTERS

The EECON1 register (Register 6-1) is the control register for memory accesses. The EECON2 register is not a physical register; it is used exclusively in the memory write and erase sequences. Reading EECON2 will read all '0's.

The WPROG bit, when set, allows the user to program a single word (two bytes) upon the execution of the WR command. If this bit is cleared, the WR command programs a block of 64 bytes.

The FREE bit, when set, will allow a program memory erase operation. When FREE is set, the erase operation is initiated on the next WR command. When FREE is clear, only writes are enabled.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set in hardware when the WR bit is set and cleared when the internal programming timer expires and the write operation is complete.

Note: During normal operation, the WRERR is read as '1'. This can indicate that a write operation was prematurely terminated by a Reset, or a write operation was attempted improperly.

The WR control bit initiates write operations. The bit cannot be cleared, only set, in software. It is cleared in hardware at the completion of the write operation.

REGISTER 6-1: EECON1: EEPROM CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-x	R/W-0	R/S-0	U-0
—	—	WPROG	FREE	WRERR ⁽¹⁾	WREN	WR	—
bit 7							bit 0

Legend:	S = Set-only bit (cannot be cleared in software)
R = Readable bit	W = Writable bit
-n = Value at POR	U = Unimplemented bit, read as '0' '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 7-6	Unimplemented: Read as '0'
bit 5	WPROG: One Word Wide Program bit 1 = Program 64 bytes on the next WR command 0 = Program 2 bytes on the next WR command
bit 4	FREE: Flash Row Erase Enable bit 1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation) 0 = Perform write only
bit 3	WRERR: Flash Program Error Flag bit ⁽¹⁾ 1 = A write operation is prematurely terminated (any Reset during self-timed programming in normal operation, or an improper write attempt) 0 = The write operation completed
bit 2	WREN: Flash Program Write Enable bit 1 = Allows write cycles to Flash program memory 0 = Inhibits write cycles to Flash program memory
bit 1	WR: Write Control bit 1 = Initiates a program memory erase cycle or write cycle (The operation is self-timed and the bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.) 0 = Write cycle is complete
bit 0	Unimplemented: Read as '0'

Note 1: When a WRERR occurs, the EEPGD and CFGS bits are not cleared. This allows tracing of the error condition.

PIC18F87J11 FAMILY

6.2.2 TABLE LATCH REGISTER (TABLAT)

The Table Latch (TABLAT) is an 8-bit register mapped into the SFR space. The Table Latch register is used to hold 8-bit data during data transfers between program memory and data RAM.

6.2.3 TABLE POINTER REGISTER (TBLPTR)

The Table Pointer (TBLPTR) register addresses a byte within the program memory. The TBLPTR is comprised of three SFR registers: Table Pointer Upper Byte, Table Pointer High Byte and Table Pointer Low Byte (TBLPTRU:TBLPTRH:TBLPTRL). These three registers join to form a 22-bit wide pointer. The low-order 21 bits allow the device to address up to 2 Mbytes of program memory space. The 22nd bit allows access to the device ID, the user ID and the Configuration bits.

The Table Pointer register, TBLPTR, is used by the TBLRD and TBLWT instructions. These instructions can update the TBLPTR in one of four ways based on the table operation. These operations are shown in Table 6-1. These operations on the TBLPTR only affect the low-order 21 bits.

6.2.4 TABLE POINTER BOUNDARIES

TBLPTR is used in reads, writes and erases of the Flash program memory.

When a TBLRD is executed, all 22 bits of the TBLPTR determine which byte is read from program memory into TABLAT.

When a TBLWT is executed, the seven LSbs of the Table Pointer register (TBLPTR<6:0>) determine which of the 64 program memory holding registers is written to. When the timed write to program memory begins (via the WR bit), the 12 MSbs of the TBLPTR (TBLPTR<21:10>) determine which program memory block of 1024 bytes is written to. For more detail, see **Section 6.5 “Writing to Flash Program Memory”**.

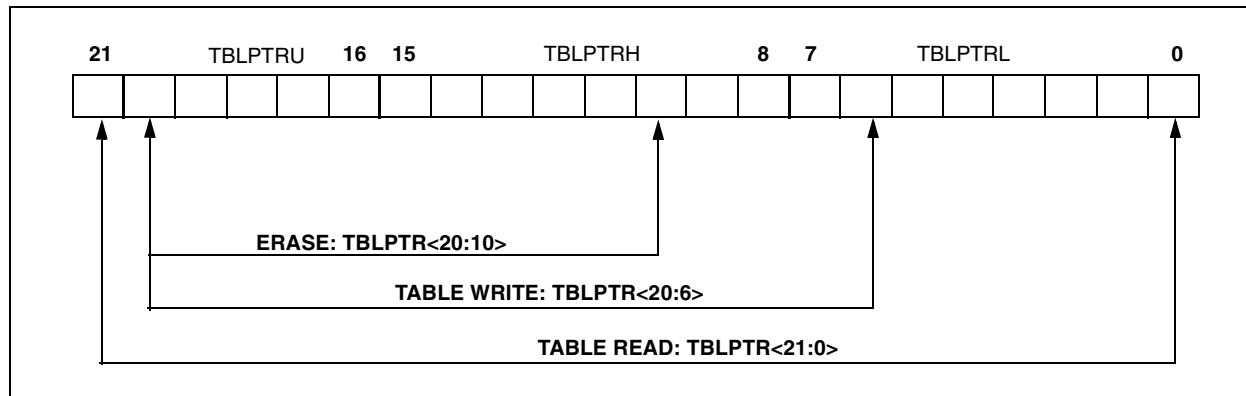
When an erase of program memory is executed, the 12 MSbs of the Table Pointer register point to the 1024-byte block that will be erased. The Least Significant bits are ignored.

Figure 6-3 describes the relevant boundaries of TBLPTR based on Flash program memory operations.

TABLE 6-1: TABLE POINTER OPERATIONS WITH TBLRD AND TBLWT INSTRUCTIONS

Example	Operation on Table Pointer
TBLRD* TBLWT*	TBLPTR is not modified
TBLRD*+ TBLWT*+	TBLPTR is incremented after the read/write
TBLRD*- TBLWT*-	TBLPTR is decremented after the read/write
TBLRD+* TBLWT+*	TBLPTR is incremented before the read/write

FIGURE 6-3: TABLE POINTER BOUNDARIES BASED ON OPERATION



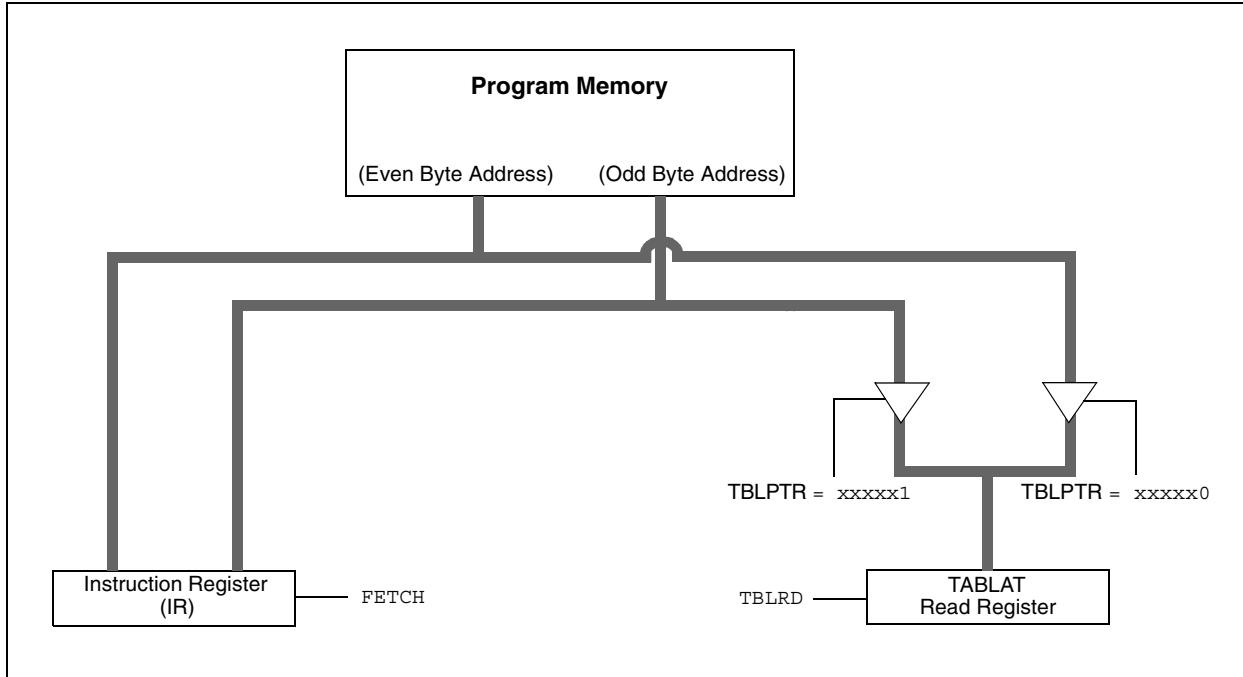
6.3 Reading the Flash Program Memory

The TBLRD instruction is used to retrieve data from program memory and places it into data RAM. Table reads from program memory are performed one byte at a time.

TBLPTR points to a byte address in program space. Executing TBLRD places the byte pointed to into TABLAT. In addition, TBLPTR can be modified automatically for the next table read operation.

The internal program memory is typically organized by words. The Least Significant bit of the address selects between the high and low bytes of the word. Figure 6-4 shows the interface between the internal program memory and the TABLAT.

FIGURE 6-4: READS FROM FLASH PROGRAM MEMORY



EXAMPLE 6-1: READING A FLASH PROGRAM MEMORY WORD

```

MOVlw  CODE_ADDR_UPPER      ; Load TBLPTR with the base
MOVwf  TBLPTRU             ; address of the word
MOVlw  CODE_ADDR_HIGH
MOVwf  TBLPTRH
MOVlw  CODE_ADDR_LOW
MOVwf  TBLPTRL

READ_WORD
    TBLRD*+
    MOVF   TABLAT, W          ; read into TABLAT and increment
    MOVWF WORD_EVEN           ; get data
    TBLRD*+
    MOVF   TABLAT, W          ; read into TABLAT and increment
    MOVWF WORD_ODD             ; get data

```

PIC18F87J11 FAMILY

6.4 Erasing Flash Program Memory

The minimum erase block is 512 words or 1024 bytes. Only through the use of an external programmer, or through ICSP control, can larger blocks of program memory be bulk erased. Word erase in the Flash array is not supported.

When initiating an erase sequence from the microcontroller itself, a block of 1024 bytes of program memory is erased. The Most Significant 12 bits of the TBLPTR<21:10> point to the block being erased. TBLPTR<9:0> are ignored.

The EECON1 register commands the erase operation. The WREN bit must be set to enable write operations. The FREE bit is set to select an erase operation. For protection, the write initiate sequence for EECON2 must be used.

A long write is necessary for erasing the internal Flash. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer.

6.4.1 FLASH PROGRAM MEMORY ERASE SEQUENCE

The sequence of events for erasing a block of internal program memory location is:

1. Load Table Pointer register with address of row being erased.
2. Set the WREN and FREE bits (EECON1<2,4>) to enable the erase operation.
3. Disable interrupts.
4. Write 55h to EECON2.
5. Write 0AAh to EECON2.
6. Set the WR bit. This will begin the row erase cycle.
7. The CPU will stall for duration of the erase for TiW (see parameter D133A).
8. Re-enable interrupts.

EXAMPLE 6-2: ERASING A FLASH PROGRAM MEMORY ROW

	MOVLW CODE_ADDR_UPPER	; load TBLPTR with the base
	MOVWF TBLPTRU	; address of the memory block
	MOVLW CODE_ADDR_HIGH	
	MOVWF TBLPTRH	
	MOVLW CODE_ADDR_LOW	
	MOVWF TBLPTRL	
ERASE_ROW	BSF EECON1, FREE	; enable Row Erase operation
	BCF INTCON, GIE	; disable interrupts
Required Sequence	MOVLW 55h	
	MOVWF EECON2	; write 55h
	MOVLW 0AAh	
	MOVWF EECON2	; write 0AAh
	BSF EECON1, WR	; start erase (CPU stall)
	BSF INTCON, GIE	; re-enable interrupts

6.5 Writing to Flash Program Memory

The programming block is 32 words or 64 bytes. Programming one word or two bytes at a time is also supported.

Table writes are used internally to load the holding registers needed to program the Flash memory. There are 64 holding registers used by the table writes for programming.

Since the Table Latch (TABLAT) is only a single byte, the TBLWT instruction may need to be executed 64 times for each programming operation (if WPROG = 0). All of the table write operations will essentially be short writes because only the holding registers are written. At the end of updating the 64 holding registers, the EECON1 register must be written to in order to start the programming operation with a long write.

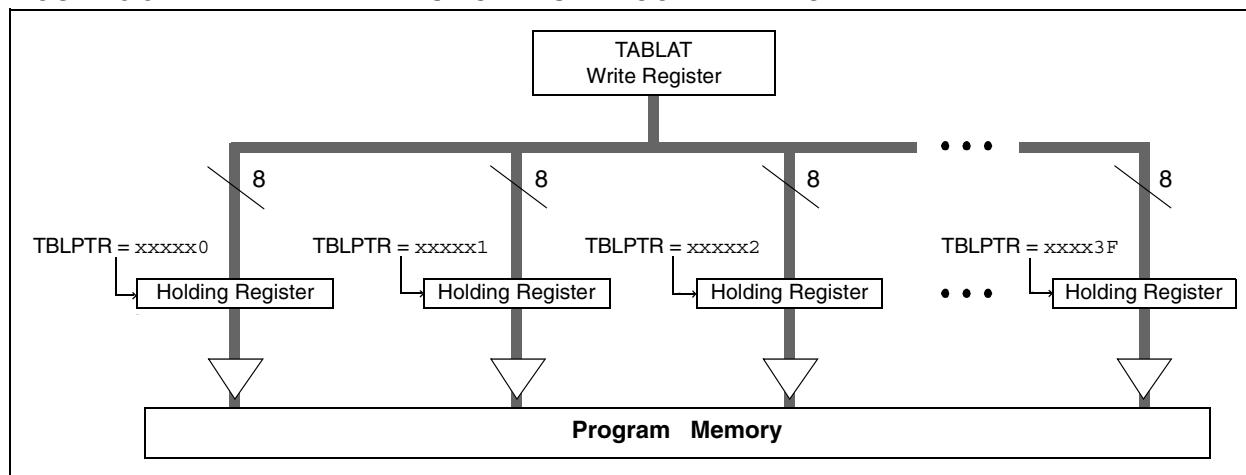
The long write is necessary for programming the internal Flash. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer.

The on-chip timer controls the write time. The write/erase voltages are generated by an on-chip charge pump, rated to operate over the voltage range of the device.

Note 1: Unlike previous PIC18 Flash devices, members of the PIC18F87J11 family do not reset the holding registers after a write occurs. The holding registers must be cleared or overwritten before a programming sequence.

2: To maintain the endurance of the program memory cells, each Flash byte should not be programmed more than one time between erase operations. Before attempting to modify the contents of the target cell a second time, a row erase of the target row, or a bulk erase of the entire memory, must be performed.

FIGURE 6-5: TABLE WRITES TO FLASH PROGRAM MEMORY



6.5.1 FLASH PROGRAM MEMORY WRITE SEQUENCE

The sequence of events for programming an internal program memory location should be:

1. Read 1024 bytes into RAM.
2. Update data values in RAM as necessary.
3. Load Table Pointer register with address being erased.
4. Execute the row erase procedure.
5. Load Table Pointer register with address of first byte being written, minus 1.
6. Write the 64 bytes into the holding registers with auto-increment.
7. Set the WREN bit (EECON1<2>) to enable byte writes.

8. Disable interrupts.
9. Write 55h to EECON2.
10. Write 0AAh to EECON2.
11. Set the WR bit. This will begin the write cycle.
12. The CPU will stall for duration of the write for Tiw (parameter D133A).
13. Re-enable interrupts.
14. Repeat steps 6 through 13 until all 1024 bytes are written to program memory.
15. Verify the memory (table read).

An example of the required code is shown in Example 6-3 on the following page.

Note: Before setting the WR bit, the Table Pointer address needs to be within the intended address range of the 64 bytes in the holding register.

PIC18F87J11 FAMILY

EXAMPLE 6-3: WRITING TO FLASH PROGRAM MEMORY

	MOVLW CODE_ADDR_UPPER MOVWF TBLPTRU MOVLW CODE_ADDR_HIGH MOVWF TBLPTRH MOVLW CODE_ADDR_LOW MOVWF TBLPTRL	; Load TBLPTR with the base address ; of the memory block, minus 1
ERASE_BLOCK	BSF EECON1, WREN BSF EECON1, FREE BCF INTCON, GIE MOVLW 55h MOVWF EECON2 MOVLW 0AAh MOVWF EECON2 BSF EECON1, WR BSF INTCON, GIE MOVLW D'16' MOVWF WRITE_COUNTER	; enable write to memory ; enable Row Erase operation ; disable interrupts ; write 55h ; write 0AAh ; start erase (CPU stall) ; re-enable interrupts ; Need to write 16 blocks of 64 to write ; one erase block of 1024
RESTART_BUFFER	MOVLW D'64' MOVWF COUNTER MOVLW BUFFER_ADDR_HIGH MOVWF FSR0H MOVLW BUFFER_ADDR_LOW MOVWF FSR0L	; point to buffer
FILL_BUFFER	...	; read the new data from I2C, SPI, ; PSP, USART, etc.
WRITE_BUFFER	MOVLW D'64 MOVWF COUNTER	; number of bytes in holding register
WRITE_BYTE_TO_HREGS	MOVFF POSTINCO, WREG MOVWF TABLAT TBLWT+*	; get low byte of buffer data ; present data to table latch ; write data, perform a short write ; to internal TBLWT holding register. ; loop until buffers are full
	DECFSZ COUNTER BRA WRITE_BYTE_TO_HREGS	
PROGRAM_MEMORY	BSF EECON1, WREN BCF INTCON, GIE Required Sequence MOVLW 55h MOVWF EECON2 MOVLW 0AAh MOVWF EECON2 BSF EECON1, WR BSF INTCON, GIE BCF EECON1, WREN DECFSZ WRITE_COUNTER BRA RESTART_BUFFER	; enable write to memory ; disable interrupts ; write 55h ; write 0AAh ; start program (CPU stall) ; re-enable interrupts ; disable write to memory ; done with one write cycle ; if not done replacing the erase block

6.5.2 FLASH PROGRAM MEMORY WRITE SEQUENCE (WORD PROGRAMMING).

The PIC18F87J11 Family of devices have a feature that allows programming a single word (two bytes). This feature is enable when the WPROG bit is set. If the memory location is already erased, the following sequence is required to enable this feature:

1. Load the Table Pointer register with the address of the data to be written
2. Write the 2 bytes into the holding registers and perform a table write

3. Set the WREN bit (EECON1<2>) to enable byte writes.
4. Disable interrupts.
5. Write 55h to EECON2.
6. Write 0AAh to EECON2.
7. Set the WR bit. This will begin the write cycle.
8. The CPU will stall for duration of the write for T_{IW} (see parameter D133A).
9. Re-enable interrupts.

EXAMPLE 6-4: SINGLE-WORD WRITE TO FLASH PROGRAM MEMORY

MOVLW CODE_ADDR_UPPER	; Load TBLPTR with the base address
MOVWF TBLPTRU	
MOVLW CODE_ADDR_HIGH	
MOVWF TBLPTRH	
MOVLW CODE_ADDR_LOW	
MOVWF TBLPTRL	
MOVLW DATA0	
MOVWF TABLAT	
TBLWT*+	
MOVLW DATA1	
MOVWF TABLAT	
TBLWT*	
PROGRAM_MEMORY	
BSF EECOM1, WPROG	; enable single word write
BSF EECOM1, WREN	; enable write to memory
BCF INTCON, GIE	; disable interrupts
Required Sequence	MOVLW 55h
MOVWF EECOM2	; write 55h
MOVLW 0AAh	
MOVWF EECOM2	; write 0AAh
BSF EECOM1, WR	; start program (CPU stall)
BSF INTCON, GIE	; re-enable interrupts
BCF EECOM1, WPROG	; disable single word write
BCF EECOM1, WREN	; disable write to memory

PIC18F87J11 FAMILY

6.5.3 WRITE VERIFY

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

6.5.4 UNEXPECTED TERMINATION OF WRITE OPERATION

If a write is terminated by an unplanned event, such as loss of power or an unexpected Reset, the memory location just programmed should be verified and reprogrammed if needed. If the write operation is interrupted by a MCLR Reset or a WDT time-out Reset during normal operation, the user can check the WRERR bit and rewrite the location(s) as needed.

6.6 Flash Program Operation During Code Protection

See **Section 24.6 “Program Verification and Code Protection”** for details on code protection of Flash program memory.

TABLE 6-2: REGISTERS ASSOCIATED WITH PROGRAM FLASH MEMORY

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
TBLPTRU	—	—	bit 21	Program Memory Table Pointer Upper Byte (TBLPTR<20:16>)					55
TBPLTRH	Program Memory Table Pointer High Byte (TBLPTR<15:8>)								55
TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								55
TABLAT	Program Memory Table Latch								55
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
EECON2	Program Memory Control Register 2 (not a physical register)								57
EECON1	—	—	WPROG	FREE	WRERR	WREN	WR	—	57

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used during Flash program memory access.

7.0 EXTERNAL MEMORY BUS

Note: The external memory bus is not implemented on 64-pin devices.

The External Memory Bus (EMB) allows the device to access external memory devices (such as Flash, EPROM, SRAM, etc.) as program or data memory. It supports both 8 and 16-Bit Data Width modes and three address widths of up to 20 bits.

The bus is implemented with 28 pins, multiplexed across four I/O ports. Three ports (PORTD, PORTE and PORTH) are multiplexed with the address/data bus for a total of 20 available lines, while PORTJ is multiplexed with the bus control signals.

A list of the pins and their functions is provided in Table 7-1.

TABLE 7-1: PIC18F87J11 FAMILY EXTERNAL BUS – I/O PORT FUNCTIONS

Name	Port	Bit	External Memory Bus Function
RD0/AD0	PORTD	0	Address bit 0 or Data bit 0
RD1/AD1	PORTD	1	Address bit 1 or Data bit 1
RD2/AD2	PORTD	2	Address bit 2 or Data bit 2
RD3/AD3	PORTD	3	Address bit 3 or Data bit 3
RD4/AD4	PORTD	4	Address bit 4 or Data bit 4
RD5/AD5	PORTD	5	Address bit 5 or Data bit 5
RD6/AD6	PORTD	6	Address bit 6 or Data bit 6
RD7/AD7	PORTD	7	Address bit 7 or Data bit 7
RE0/AD8	PORTE	0	Address bit 8 or Data bit 8
RE1/AD9	PORTE	1	Address bit 9 or Data bit 9
RE2/AD10	PORTE	2	Address bit 10 or Data bit 10
RE3/AD11	PORTE	3	Address bit 11 or Data bit 11
RE4/AD12	PORTE	4	Address bit 12 or Data bit 12
RE5/AD13	PORTE	5	Address bit 13 or Data bit 13
RE6/AD14	PORTE	6	Address bit 14 or Data bit 14
RE7/AD15	PORTE	7	Address bit 15 or Data bit 15
RH0/A16	PORTH	0	Address bit 16
RH1/A17	PORTH	1	Address bit 17
RH2/A18	PORTH	2	Address bit 18
RH3/A19	PORTH	3	Address bit 19
RJ0/ALE	PORTJ	0	Address Latch Enable (ALE) Control pin
RJ1/OE	PORTJ	1	Output Enable (\overline{OE}) Control pin
RJ2/WRL	PORTJ	2	Write Low (\overline{WRL}) Control pin
RJ3/WRH	PORTJ	3	Write High (\overline{WRH}) Control pin
RJ4/BA0	PORTJ	4	Byte Address bit 0 (BA0)
RJ5/CE	PORTJ	5	Chip Enable (\overline{CE}) Control pin
RJ6/LB	PORTJ	6	Lower Byte Enable (\overline{LB}) Control pin
RJ7/UB	PORTJ	7	Upper Byte Enable (\overline{UB}) Control pin

Note: For the sake of clarity, only I/O port and external bus assignments are shown here. One or more additional multiplexed features may be available on some pins.

PIC18F87J11 FAMILY

7.1 External Memory Bus Control

The operation of the interface is controlled by the MEMCON register (Register 7-1). This register is available in all program memory operating modes except Microcontroller mode. In this mode, the register is disabled and cannot be written to.

The EBDIS bit (MEMCON<7>) controls the operation of the bus and related port functions. Clearing EBDIS enables the interface and disables the I/O functions of the ports, as well as any other functions multiplexed to those pins. Setting the bit enables the I/O ports and other functions, but allows the interface to override everything else on the pins when an external memory operation is required. By default, the external bus is always enabled and disables all other I/O.

The operation of the EBDIS bit is also influenced by the program memory mode being used. This is discussed in more detail in **Section 7.5 “Program Memory Modes and the External Memory Bus”**.

The WAIT bits allow for the addition of wait states to external memory operations. The use of these bits is discussed in **Section 7.3 “Wait States”**.

The WM bits select the particular operating mode used when the bus is operating in 16-Bit Data Width mode. These are discussed in more detail in **Section 7.6 “16-Bit Data Width Modes”**. These bits have no effect when an 8-bit Data Width mode is selected.

The MEMCON register (see Register 7-1) shares the same memory space as the PR2 register and can be alternately selected based on the designation of the ADSHR bit in the WDTCON register (see Register 24-9).

REGISTER 7-1: MEMCON: EXTERNAL MEMORY BUS CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0
bit 7	bit 0						

Legend:

R = Readable bit
-n = Value at POR

W = Writable bit
'1' = Bit is set

U = Unimplemented bit, read as '0'
'0' = Bit is cleared
x = Bit is unknown

- bit 7 **EBDIS:** External Bus Disable bit
1 = External bus enabled when microcontroller accesses external memory; otherwise, all external bus drivers are mapped as I/O ports
0 = External bus always enabled, I/O ports are disabled
- bit 6 **Unimplemented:** Read as '0'
- bit 5-4 **WAIT1:WAIT0:** Table Reads and Writes Bus Cycle Wait Count bits
11 = Table reads and writes will wait 0 Tcy
10 = Table reads and writes will wait 1 Tcy
01 = Table reads and writes will wait 2 Tcy
00 = Table reads and writes will wait 3 Tcy
- bit 3-2 **Unimplemented:** Read as '0'
- bit 1-0 **WM1:WM0:** TBLWT Operation with 16-Bit Data Bus Width Select bits
1x = Word Write mode: TABLAT word output, WRH active when TABLAT written
01 = Byte Select mode: TABLAT data copied on both MSB and LSB, WRH and (UB or LB) will activate
00 = Byte Write mode: TABLAT data copied on both MSB and LSB, WRH or WRL will activate

7.2 Address and Data Width

The PIC18F87J11 Family of devices can be independently configured for different address and data widths on the same memory bus. Both address and data width are set by Configuration bits in the CONFIG3L register. As Configuration bits, this means that these options can only be configured by programming the device and are not controllable in software.

The BW bit selects an 8-bit or 16-bit data bus width. Setting this bit (default) selects a data width of 16 bits.

The EMB1:EMB0 bits determine both the program memory operating mode and the address bus width. The available options are 20-bit, 16-bit and 12-bit, as well as Microcontroller mode (external bus disabled). Selecting a 16-bit or 12-bit width makes a corresponding number of high-order lines available for I/O functions. These pins are no longer affected by the setting of the EBDIS bit. For example, selecting a 16-Bit Addressing mode (EMB1:EMB0 = 01) disables A19:A16 and allows PORTH<3:0> to function without interruptions from the bus. Using the smaller address widths allows users to tailor the memory bus to the size of the external memory space for a particular design while freeing up pins for dedicated I/O operation.

Because the EMB bits have the effect of disabling pins for memory bus operations, it is important to always select an address width at least equal to the data width. If a 12-bit address width is used with a 16-bit data width, the upper four bits of data will not be available on the bus.

All combinations of address and data widths require multiplexing of address and data information on the same lines. The address and data multiplexing, as well as I/O ports made available by the use of smaller address widths, are summarized in Table 7-2.

TABLE 7-2: ADDRESS AND DATA LINES FOR DIFFERENT ADDRESS AND DATA WIDTHS

Data Width	Address Width	Multiplexed Data and Address Lines (and Corresponding Ports)	Address Only Lines (and Corresponding Ports)	Ports Available for I/O
8-bit	12-bit	AD7:AD0 (PORTD<7:0>)	AD11:AD8 (PORTE<3:0>)	PORTE<7:4>, All of PORTH
	16-bit		AD15:AD8 (PORTE<7:0>)	All of PORTH
	20-bit		A19:A16, AD15:AD8 (PORTH<3:0>, PORTE<7:0>)	—
16-bit	16-bit	AD15:AD0 (PORTD<7:0>, PORTE<7:0>)	—	All of PORTH
	20-bit		A19:A16 (PORTH<3:0>)	—

7.2.1 ADDRESS SHIFTING ON THE EXTERNAL BUS

By default, the address presented on the external bus is the value of the PC. In practical terms, this means that addresses in the external memory device below the top of on-chip memory are unavailable to the microcontroller. To access these physical locations, the glue logic between the microcontroller and the external memory must somehow translate addresses.

To simplify the interface, the external bus offers an extension of Extended Microcontroller mode that automatically performs address shifting. This feature is controlled by the EASHFT Configuration bit. Setting this bit offsets addresses on the bus by the size of the microcontroller's on-chip program memory and sets the bottom address at 0000h. This allows the device to use the entire range of physical addresses of the external memory.

7.2.2 21-BIT ADDRESSING

As an extension of 20-bit address width operation, the external memory bus can also fully address a 2-Mbyte memory space. This is done by using the Bus Address bit 0 (BA0) control line as the Least Significant bit of the address. The UB and LB control signals may also be used with certain memory devices to select the upper and lower bytes within a 16-bit wide data word.

This addressing mode is available in both 8-bit and certain 16-Bit Data Width modes. Additional details are provided in **Section 7.6.3 “16-Bit Byte Select Mode”** and **Section 7.7 “8-Bit Data Width Mode”**.

PIC18F87J11 FAMILY

7.3 Wait States

While it may be assumed that external memory devices will operate at the microcontroller clock rate, this is often not the case. In fact, many devices require longer times to write or retrieve data than the time allowed by the execution of table read or table write operations.

To compensate for this, the external memory bus can be configured to add a fixed delay to each table operation using the bus. Wait states are enabled by setting the WAIT Configuration bit. When enabled, the amount of delay is set by the WAIT1:WAIT0 bits (MEMCON<5:4>). The delay is based on multiples of microcontroller instruction cycle time and are added following the instruction cycle when the table operation is executed. The range is from no delay to 3 T_{CY} (default value).

7.4 Port Pin Weak Pull-ups

With the exception of the upper address lines, A19:A16, the pins associated with the external memory bus are equipped with weak pull-ups. The pull-ups are controlled by the upper three bits of the PORTG register (PORTG<7:5>). They are named RDPU, REPU and RJPU and control pull-ups on PORTD, PORTE and PORTJ, respectively. Setting one of these bits enables the corresponding pull-ups for that port. All pull-ups are disabled by default on all device Resets.

In Extended Microcontroller mode, the port pull-ups can be useful in preserving the memory state on the external bus while the bus is temporarily disabled (EBDIS = '1').

7.5 Program Memory Modes and the External Memory Bus

The PIC18F87J11 Family of devices is capable of operating in one of two program memory modes, using combinations of on-chip and external program memory. The functions of the multiplexed port pins depend on the program memory mode selected, as well as the setting of the EBDIS bit.

In **Microcontroller Mode**, the bus is not active and the pins have their port functions only. Writes to the MEMCOM register are not permitted. The Reset value of EBDIS ('0') is ignored and EMB pins behave as I/O ports.

In **Extended Microcontroller Mode**, the external program memory bus shares I/O port functions on the pins. When the device is fetching or doing table read/table write operations on the external program memory space, the pins will have the external bus function.

If the device is fetching and accessing internal program memory locations only, the EBDIS control bit will change the pins from external memory to I/O port

functions. When EBDIS = 0, the pins function as the external bus. When EBDIS = 1, the pins function as I/O ports.

If the device fetches or accesses external memory while EBDIS = 1, the pins will switch to external bus. If the EBDIS bit is set by a program executing from external memory, the action of setting the bit will be delayed until the program branches into the internal memory. At that time, the pins will change from external bus to I/O ports.

If the device is executing out of internal memory when EBDIS = 0, the memory bus address/data and control pins will not be active. They will go to a state where the active address/data pins are tri-state; the CE, OE, WRH, WRL, UB and LB signals are '1' and ALE and BA0 are '0'. Note that only those pins associated with the current address width are forced to tri-state; the other pins continue to function as I/O. In the case of 16-bit address width, for example, only AD<15:0> (PORTD and PORTE) are affected; A19:A16 (PORTH<3:0>) continue to function as I/O.

In all external memory modes, the bus takes priority over any other peripherals that may share pins with it. This includes the Parallel Master Port and serial communication modules which would otherwise take priority over the I/O port.

7.6 16-Bit Data Width Modes

In 16-Bit Data Width mode, the external memory interface can be connected to external memories in three different configurations:

- 16-Bit Byte Write
- 16-Bit Word Write
- 16-Bit Byte Select

The configuration to be used is determined by the WM1:WM0 bits in the MEMCON register (MEMCON<1:0>). These three different configurations allow the designer maximum flexibility in using both 8-bit and 16-bit devices with 16-bit data.

For all 16-bit modes, the Address Latch Enable (ALE) pin indicates that the address bits, AD<15:0>, are available on the external memory interface bus. Following the address latch, the Output Enable signal (OE) will enable both bytes of program memory at once to form a 16-bit instruction word. The Chip Enable signal (CE) is active at any time that the microcontroller accesses external memory, whether reading or writing; it is inactive (asserted high) whenever the device is in Sleep mode.

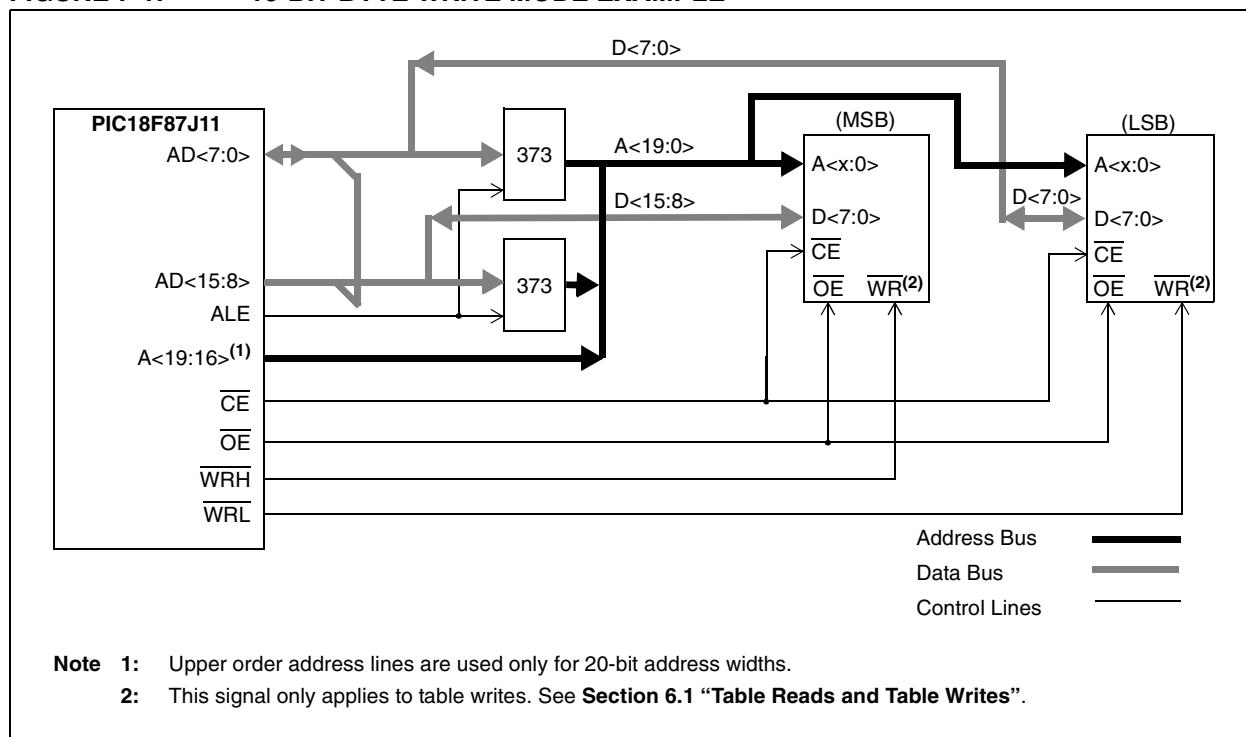
In Byte Select mode, JEDEC standard Flash memories will require BA0 for the byte address line and one I/O line to select between Byte and Word mode. The other 16-bit modes do not need BA0. JEDEC standard static RAM memories will use the UB or LB signals for byte selection.

7.6.1 16-BIT BYTE WRITE MODE

Figure 7-1 shows an example of 16-Bit Byte Write mode for PIC18F87J11 Family devices. This mode is used for two separate 8-bit memories connected for 16-bit operation. This generally includes basic EPROM and Flash devices. It allows table writes to byte-wide external memories.

During a TBLWT instruction cycle, the TABLAT data is presented on the upper and lower bytes of the AD15:AD0 bus. The appropriate WRH or WRL control line is strobed on the LSb of the TBLPTR.

FIGURE 7-1: 16-BIT BYTE WRITE MODE EXAMPLE



PIC18F87J11 FAMILY

7.6.2 16-BIT WORD WRITE MODE

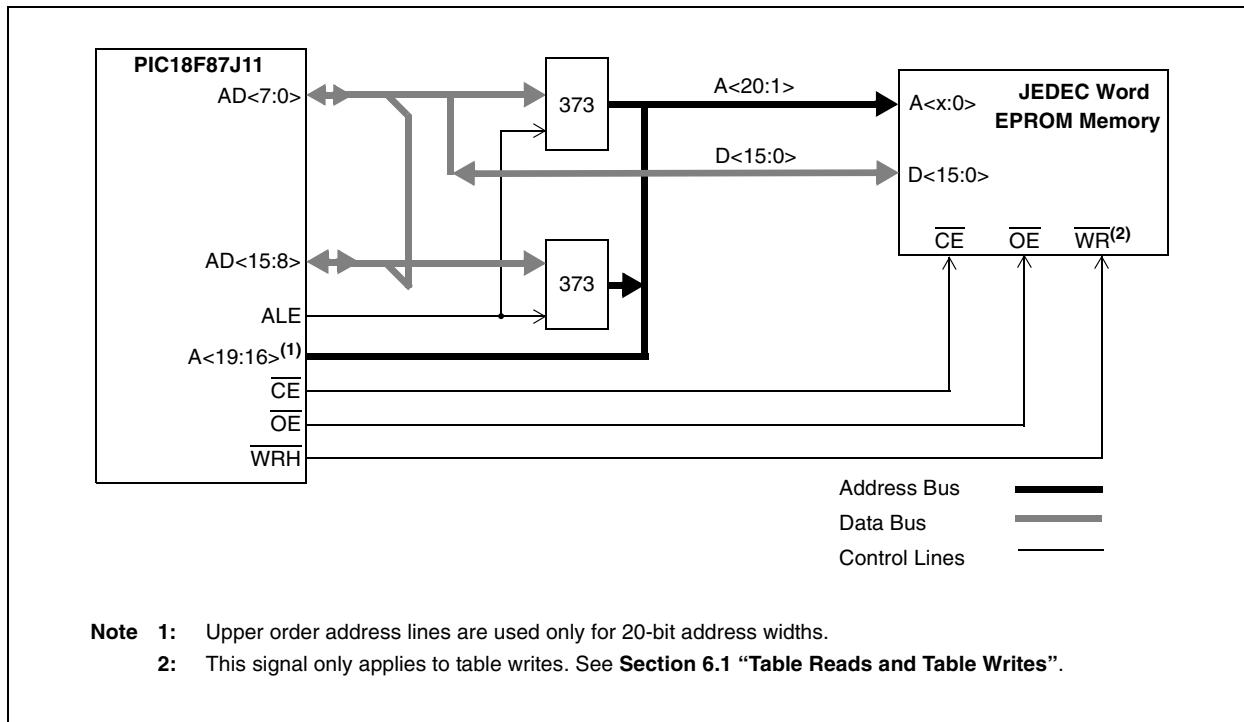
Figure 7-2 shows an example of 16-Bit Word Write mode for PIC18F87J11 Family devices. This mode is used for word-wide memories which include some of the EPROM and Flash-type memories. This mode allows opcode fetches and table reads from all forms of 16-bit memory and table writes to any type of word-wide external memories. This method makes a distinction between TBLWT cycles to even or odd addresses.

During a TBLWT cycle to an even address (TBLPTR<0> = 0), the TABLAT data is transferred to a holding latch and the external address data bus is tri-stated for the data portion of the bus cycle. No write signals are activated.

During a TBLWT cycle to an odd address (TBLPTR<0> = 1), the TABLAT data is presented on the upper byte of the AD15:AD0 bus. The contents of the holding latch are presented on the lower byte of the AD15:AD0 bus.

The WRH signal is strobed for each write cycle; the WRL pin is unused. The signal on the BA0 pin indicates the LSb of the TBLPTR, but it is left unconnected. Instead, the UB and LB signals are active to select both bytes. The obvious limitation to this method is that the table write must be done in pairs on a specific word boundary to correctly write a word location.

FIGURE 7-2: 16-BIT WORD WRITE MODE EXAMPLE



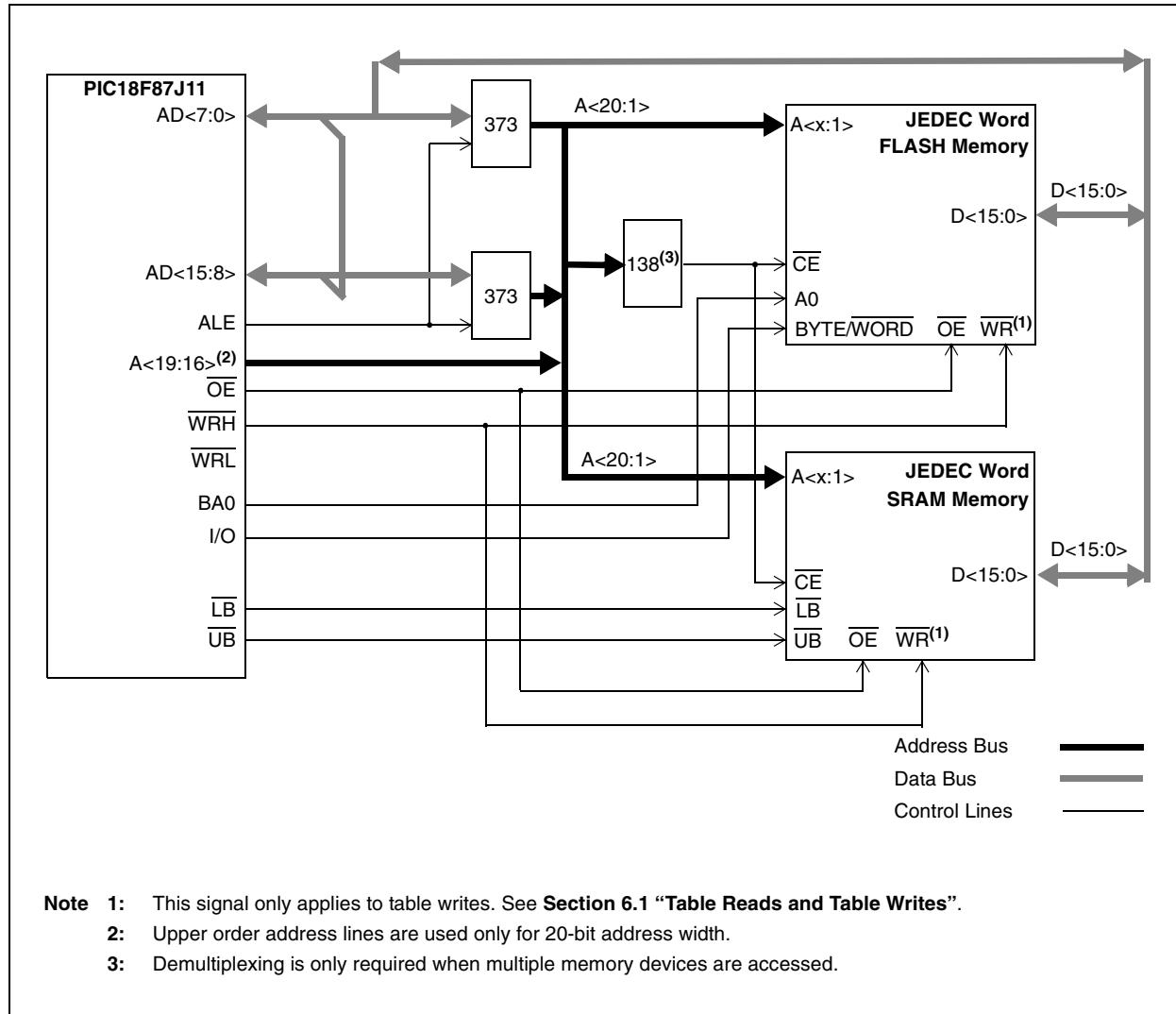
7.6.3 16-BIT BYTE SELECT MODE

Figure 7-3 shows an example of 16-Bit Byte Select mode. This mode allows table write operations to word-wide external memories with byte selection capability. This generally includes both word-wide Flash and SRAM devices.

During a TBLWT cycle, the TABLAT data is presented on the upper and lower byte of the AD15:AD0 bus. The WRH signal is strobed for each write cycle; the WRL pin is not used. The BA0 or UB/LB signals are used to select the byte to be written, based on the Least Significant bit of the TBLPTR register.

Flash and SRAM devices use different control signal combinations to implement Byte Select mode. JEDEC standard Flash memories require that a controller I/O port pin be connected to the memory's BYTE/WORD pin to provide the select signal. They also use the BA0 signal from the controller as a byte address. JEDEC standard static RAM memories, on the other hand, use the UB or LB signals to select the byte.

FIGURE 7-3: 16-BIT BYTE SELECT MODE EXAMPLE



PIC18F87J11 FAMILY

7.6.4 16-BIT MODE TIMING

The presentation of control signals on the external memory bus is different for the various operating modes. Typical signal timing diagrams are shown in Figure 7-4 and Figure 7-5.

FIGURE 7-4: EXTERNAL MEMORY BUS TIMING FOR TBLRD (EXTENDED MICROCONTROLLER MODE)

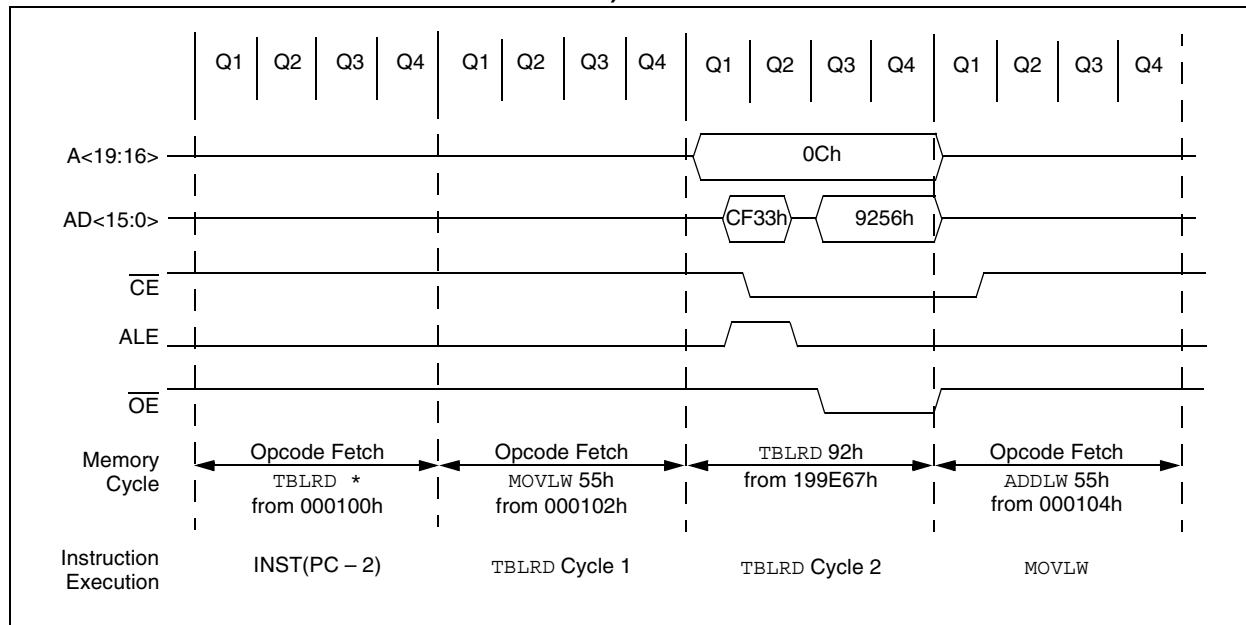
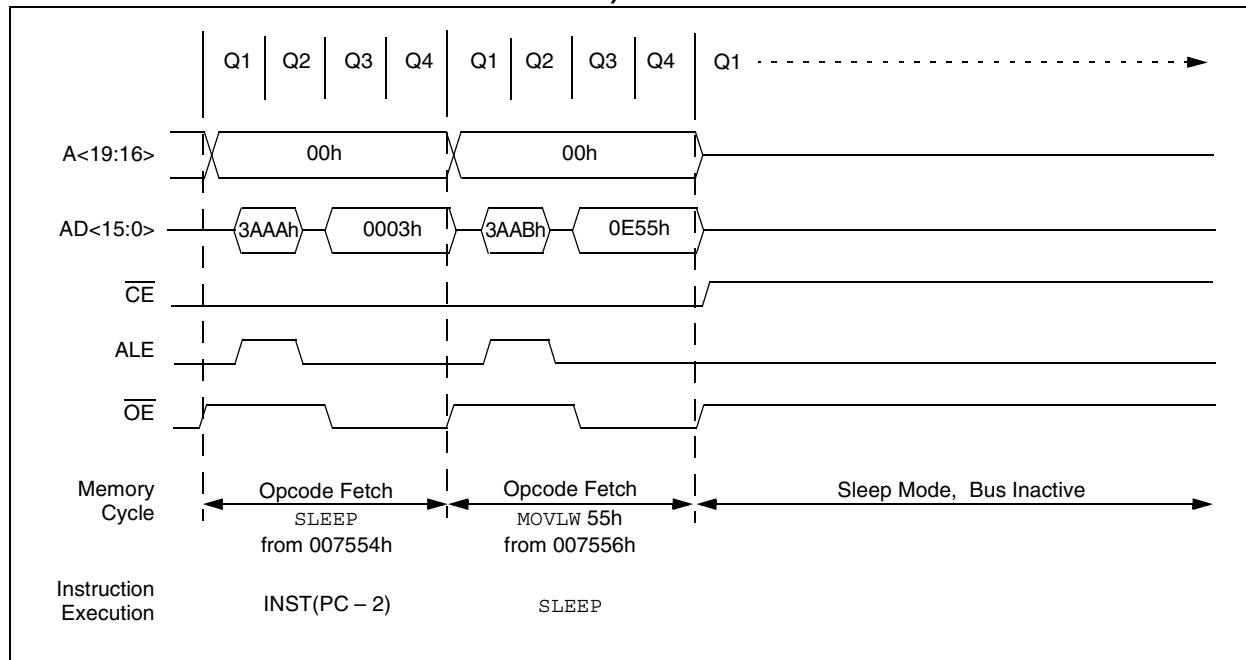


FIGURE 7-5: EXTERNAL MEMORY BUS TIMING FOR SLEEP (EXTENDED MICROCONTROLLER MODE)



7.7 8-Bit Data Width Mode

In 8-Bit Data Width mode, the external memory bus operates only in Multiplexed mode; that is, data shares the 8 Least Significant bits of the address bus.

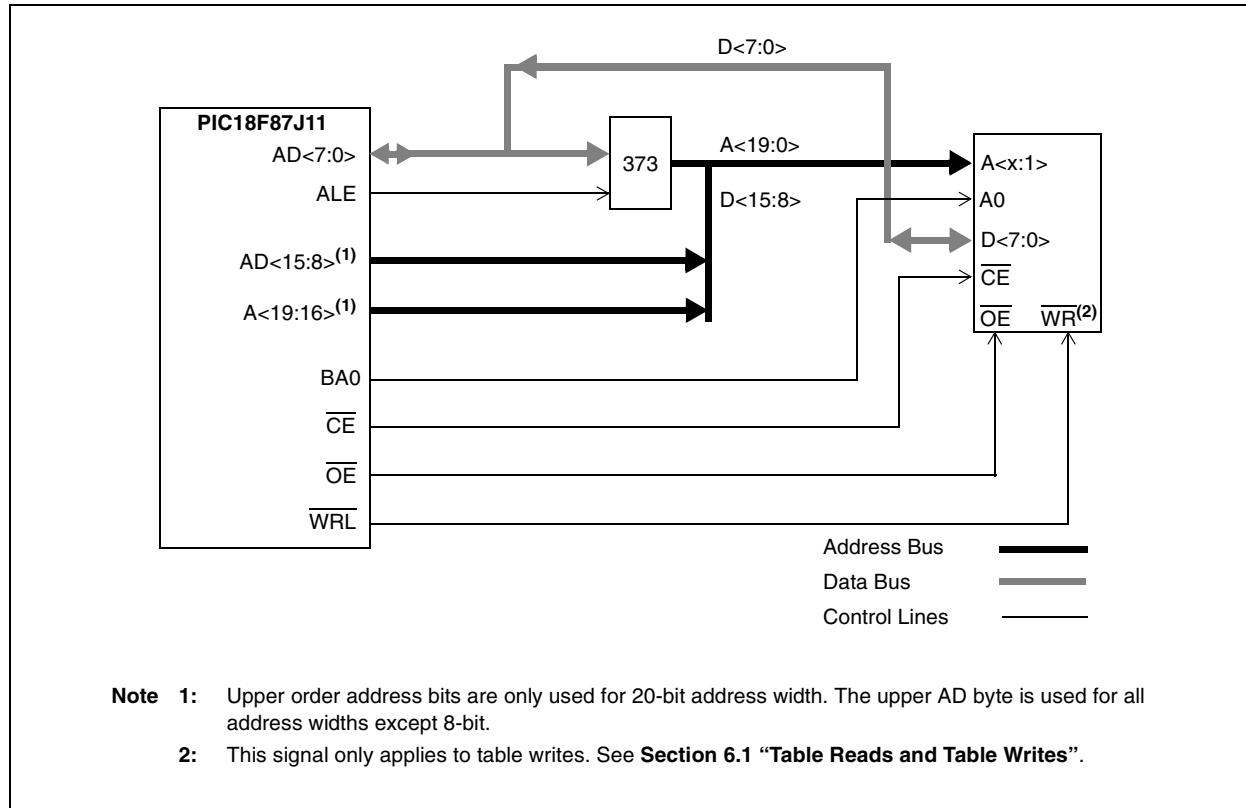
Figure 7-6 shows an example of 8-Bit Multiplexed mode for 80-pin devices. This mode is used for a single 8-bit memory connected for 16-bit operation. The instructions will be fetched as two 8-bit bytes on a shared data/address bus. The two bytes are sequentially fetched within one instruction cycle (T_{CY}). Therefore, the designer must choose external memory devices according to timing calculations based on 1/2 T_{CY} (2 times the instruction rate). For proper memory speed selection, glue logic propagation delay times must be considered, along with setup and hold times.

The Address Latch Enable (ALE) pin indicates that the address bits, AD<15:0>, are available on the external memory interface bus. The Output Enable signal (\overline{OE})

will enable one byte of program memory for a portion of the instruction cycle, then BA0 will change and the second byte will be enabled to form the 16-bit instruction word. The Least Significant bit of the address, BA0, must be connected to the memory devices in this mode. The Chip Enable signal (\overline{CE}) is active at any time that the microcontroller accesses external memory, whether reading or writing. It is inactive (asserted high) whenever the device is in Sleep mode. This generally includes basic EPROM and Flash devices. It allows table writes to byte-wide external memories.

During a TBLWT instruction cycle, the TABLAT data is presented on the upper and lower bytes of the AD15:AD0 bus. The appropriate level of the BA0 control line is strobed on the LSb of the TBLPTR.

FIGURE 7-6: 8-BIT MULTIPLEXED MODE EXAMPLE



PIC18F87J11 FAMILY

7.7.1 8-BIT MODE TIMING

The presentation of control signals on the external memory bus is different for the various operating modes. Typical signal timing diagrams are shown in Figure 7-7 and Figure 7-8.

FIGURE 7-7: EXTERNAL MEMORY BUS TIMING FOR TBLRD (EXTENDED MICROCONTROLLER MODE)

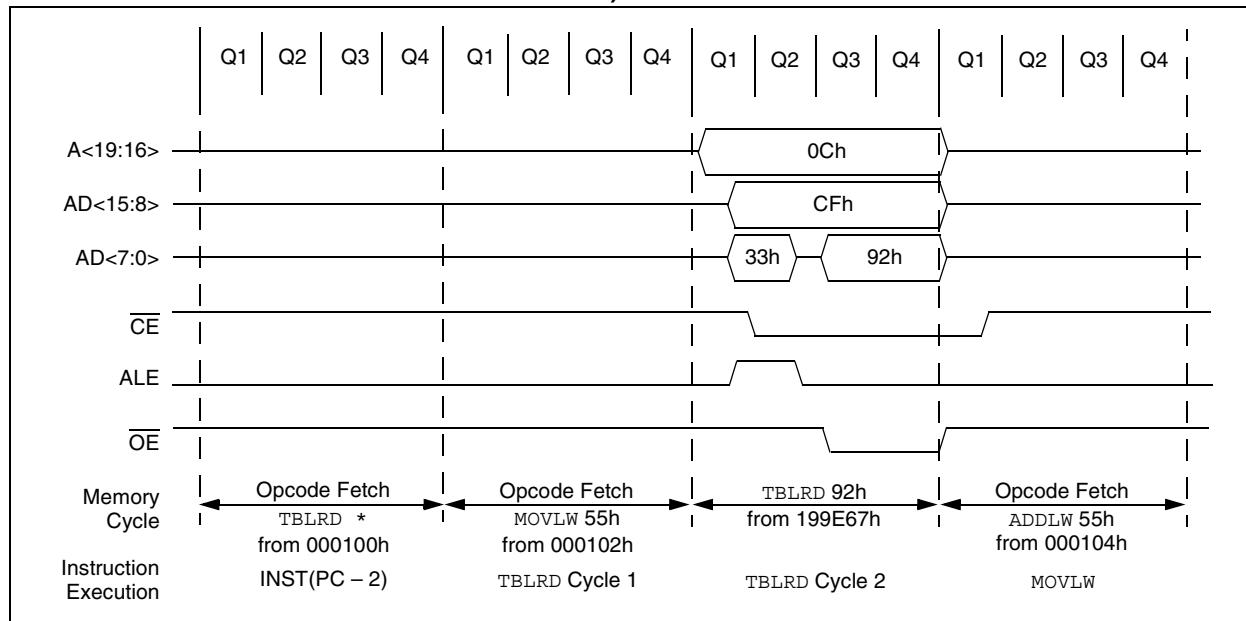
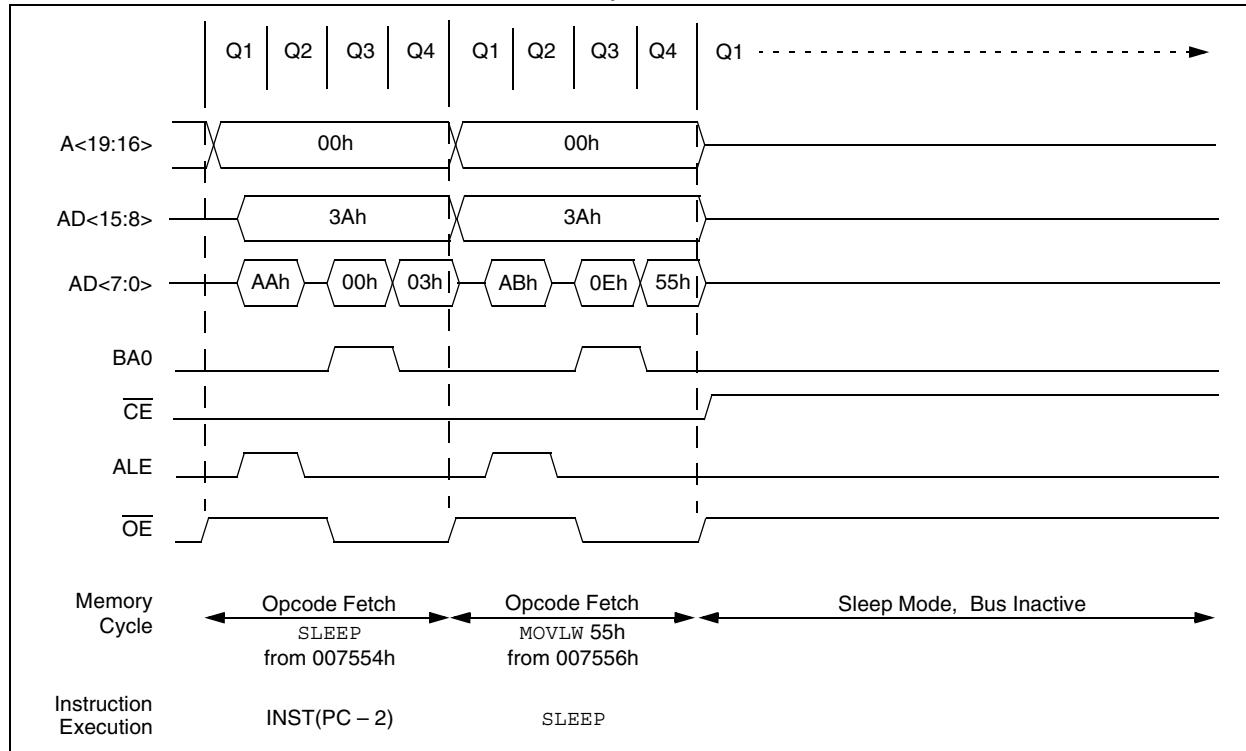


FIGURE 7-8: EXTERNAL MEMORY BUS TIMING FOR SLEEP (EXTENDED MICROCONTROLLER MODE)



7.8 Operation in Power-Managed Modes

In alternate, power-managed Run modes, the external bus continues to operate normally. If a clock source with a lower speed is selected, bus operations will run at that speed. In these cases, excessive access times for the external memory may result if wait states have been enabled and added to external memory operations. If operations in a lower power Run mode are anticipated, users should provide in their applications for adjusting memory access times at the lower clock speeds.

In Sleep and Idle modes, the microcontroller core does not need to access data; bus operations are suspended. The state of the external bus is frozen, with the address/data pins and most of the control pins holding at the same state they were in when the mode was invoked. The only potential changes are the CE, LB and UB pins, which are held at logic high.

PIC18F87J11 FAMILY

NOTES:

8.0 8 x 8 HARDWARE MULTIPLIER

8.1 Introduction

All PIC18 devices include an 8 x 8 hardware multiplier as part of the ALU. The multiplier performs an unsigned operation and yields a 16-bit result that is stored in the product register pair, PRODH:PRODL. The multiplier's operation does not affect any flags in the STATUS register.

Making multiplication a hardware operation allows it to be completed in a single instruction cycle. This has the advantages of higher computational throughput and reduced code size for multiplication algorithms and allows the PIC18 devices to be used in many applications previously reserved for digital signal processors. A comparison of various hardware and software multiply operations, along with the savings in memory and execution time, is shown in Table 8-1.

8.2 Operation

Example 8-1 shows the instruction sequence for an 8 x 8 unsigned multiplication. Only one instruction is required when one of the arguments is already loaded in the WREG register.

Example 8-2 shows the sequence to do an 8 x 8 signed multiplication. To account for the sign bits of the arguments, each argument's Most Significant bit (MSb) is tested and the appropriate subtractions are done.

EXAMPLE 8-1: 8 x 8 UNSIGNED MULTIPLY ROUTINE

```
MOVF ARG1, W      ;
MULWF ARG2        ; ARG1 * ARG2 ->
                  ; PRODH:PRODL
```

EXAMPLE 8-2: 8 x 8 SIGNED MULTIPLY ROUTINE

```
MOVF ARG1, W      ;
MULWF ARG2        ; ARG1 * ARG2 ->
                  ; PRODH:PRODL
BTFS C ARG2, SB   ; Test Sign Bit
SUBWF PRODH, F    ; PRODH = PRODH
                  ; - ARG1
MOVF ARG2, W      ;
BTFS C ARG1, SB   ; Test Sign Bit
SUBWF PRODH, F    ; PRODH = PRODH
                  ; - ARG2
```

TABLE 8-1: PERFORMANCE COMPARISON FOR VARIOUS MULTIPLY OPERATIONS

Routine	Multiply Method	Program Memory (Words)	Cycles (Max)	Time		
				@ 48 MHz	@ 10 MHz	@ 4 MHz
8 x 8 unsigned	Without hardware multiply	13	69	5.7 µs	27.6 µs	69 µs
	Hardware multiply	1	1	83.3 ns	400 ns	1 µs
8 x 8 signed	Without hardware multiply	33	91	7.5 µs	36.4 µs	91 µs
	Hardware multiply	6	6	500 ns	2.4 µs	6 µs
16 x 16 unsigned	Without hardware multiply	21	242	20.1 µs	96.8 µs	242 µs
	Hardware multiply	28	28	2.3 µs	11.2 µs	28 µs
16 x 16 signed	Without hardware multiply	52	254	21.6 µs	102.6 µs	254 µs
	Hardware multiply	35	40	3.3 µs	16.0 µs	40 µs

PIC18F87J11 FAMILY

Example 8-3 shows the sequence to do a 16×16 unsigned multiplication. Equation 8-1 shows the algorithm that is used. The 32-bit result is stored in four registers (RES3:RES0).

EQUATION 8-1: 16 x 16 UNSIGNED MULTIPLICATION ALGORITHM

$$\begin{aligned} \text{RES3:RES0} &= \text{ARG1H:ARG1L} \bullet \text{ARG2H:ARG2L} \\ &= (\text{ARG1H} \bullet \text{ARG2H} \bullet 2^{16}) + \\ &\quad (\text{ARG1H} \bullet \text{ARG2L} \bullet 2^8) + \\ &\quad (\text{ARG1L} \bullet \text{ARG2H} \bullet 2^8) + \\ &\quad (\text{ARG1L} \bullet \text{ARG2L}) \end{aligned}$$

EXAMPLE 8-3: 16 x 16 UNSIGNED MULTIPLY ROUTINE

```

MOVF  ARG1L, W
MULWF ARG2L          ; ARG1L * ARG2L->
                     ; PRODH:PRODL
MOVFF  PRODH, RES1  ;
MOVFF  PRODL, RES0  ;
;

MOVF  ARG1H, W
MULWF ARG2H          ; ARG1H * ARG2H->
                     ; PRODH:PRODL
MOVFF  PRODH, RES3  ;
MOVFF  PRODL, RES2  ;
;

MOVF  ARG1L, W
MULWF ARG2H          ; ARG1L * ARG2H->
                     ; PRODH:PRODL
MOVF  PRODL, W       ;
ADDWF  RES1, F        ; Add cross
MOVF  PRODH, W       ; products
ADDWFC RES2, F        ;
CLRF   WREG           ;
ADDWFC RES3, F        ;
;

MOVF  ARG1H, W
MULWF ARG2L          ; ARG1H * ARG2L->
                     ; PRODH:PRODL
MOVF  PRODL, W       ;
ADDWF  RES1, F        ; Add cross
MOVF  PRODH, W       ; products
ADDWFC RES2, F        ;
CLRF   WREG           ;
ADDWFC RES3, F        ;
;

BTFS  ARG2H, 7         ; ARG2H:ARG2L neg?
BRA   SIGN_ARG1        ; no, check ARG1
MOVF  ARG1L, W       ;
SUBWF RES2             ;
MOVF  ARG1H, W       ;
SUBWFB RES3             ;
;

CONT_CODE
:
```

Example 8-4 shows the sequence to do a 16×16 signed multiply. Equation 8-2 shows the algorithm used. The 32-bit result is stored in four registers (RES3:RES0). To account for the sign bits of the arguments, the MSb for each argument pair is tested and the appropriate subtractions are done.

EQUATION 8-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM

$$\begin{aligned} \text{RES3:RES0} &= \text{ARG1H:ARG1L} \bullet \text{ARG2H:ARG2L} \\ &= (\text{ARG1H} \bullet \text{ARG2H} \bullet 2^{16}) + \\ &\quad (\text{ARG1H} \bullet \text{ARG2L} \bullet 2^8) + \\ &\quad (\text{ARG1L} \bullet \text{ARG2H} \bullet 2^8) + \\ &\quad (\text{ARG1L} \bullet \text{ARG2L}) + \\ &\quad (-1 \bullet \text{ARG2H} \ll 7 \gg \bullet \text{ARG1H:ARG1L} \bullet 2^{16}) + \\ &\quad (-1 \bullet \text{ARG1H} \ll 7 \gg \bullet \text{ARG2H:ARG2L} \bullet 2^{16}) \end{aligned}$$

EXAMPLE 8-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```

MOVF  ARG1L, W
MULWF ARG2L          ; ARG1L * ARG2L ->
                     ; PRODH:PRODL
MOVFF  PRODH, RES1  ;
MOVFF  PRODL, RES0  ;
;

MOVF  ARG1H, W
MULWF ARG2H          ; ARG1H * ARG2H ->
                     ; PRODH:PRODL
MOVFF  PRODH, RES3  ;
MOVFF  PRODL, RES2  ;
;

MOVF  ARG1L, W
MULWF ARG2H          ; ARG1L * ARG2H ->
                     ; PRODH:PRODL
MOVF  PRODL, W       ;
ADDWF  RES1, F        ; Add cross
MOVF  PRODH, W       ; products
ADDWFC RES2, F        ;
CLRF   WREG           ;
ADDWFC RES3, F        ;
;

MOVF  ARG1H, W
MULWF ARG2L          ; ARG1H * ARG2L ->
                     ; PRODH:PRODL
MOVF  PRODL, W       ;
ADDWF  RES1, F        ; Add cross
MOVF  PRODH, W       ; products
ADDWFC RES2, F        ;
CLRF   WREG           ;
ADDWFC RES3, F        ;
;

BTFS  ARG2H, 7         ; ARG2H:ARG2L neg?
BRA   SIGN_ARG1        ; no, check ARG1
MOVF  ARG1L, W       ;
SUBWF RES2             ;
MOVF  ARG1H, W       ;
SUBWFB RES3             ;
;

SIGN_ARG1
BTFS  ARG1H, 7         ; ARG1H:ARG1L neg?
BRA   CONT_CODE        ; no, done
MOVF  ARG2L, W       ;
SUBWF RES2             ;
MOVF  ARG2H, W       ;
SUBWFB RES3             ;
;

CONT_CODE
:
```

9.0 INTERRUPTS

Members of the PIC18F87J11 Family of devices have multiple interrupt sources and an interrupt priority feature that allows most interrupt sources to be assigned a high-priority level or a low-priority level. The high-priority interrupt vector is at 0008h and the low-priority interrupt vector is at 0018h. High-priority interrupt events will interrupt any low-priority interrupts that may be in progress.

There are thirteen registers which are used to control interrupt operation. These registers are:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2, PIR3
- PIE1, PIE2, PIE3
- IPR1, IPR2, IPR3

It is recommended that the Microchip header files supplied with MPLAB® IDE be used for the symbolic bit names in these registers. This allows the assembler/compiler to automatically take care of the placement of these bits within the specified register.

In general, interrupt sources have three bits to control their operation. They are:

- **Flag bit** to indicate that an interrupt event occurred
- **Enable bit** that allows program execution to branch to the interrupt vector address when the flag bit is set
- **Priority bit** to select high-priority or low-priority

The interrupt priority feature is enabled by setting the IPEN bit (RCON<7>). When interrupt priority is enabled, there are two bits which enable interrupts globally. Setting the GIEH bit (INTCON<7>) enables all interrupts that have the priority bit set (high priority). Setting the GIEL bit (INTCON<6>) enables all interrupts that have the priority bit cleared (low priority). When the interrupt flag, enable bit and appropriate global interrupt enable bit are set, the interrupt will vector immediately to address 0008h or 0018h, depending on the priority bit setting. Individual interrupts can be disabled through their corresponding enable bits.

When the IPEN bit is cleared (default state), the interrupt priority feature is disabled and interrupts are compatible with PIC16 mid-range devices. In Compatibility mode, the interrupt priority bits for each source have no effect. INTCON<6> is the PEIE bit which enables/disables all peripheral interrupt sources. INTCON<7> is the GIE bit which enables/disables all interrupt sources. All interrupts branch to address 0008h in Compatibility mode.

When an interrupt is responded to, the global interrupt enable bit is cleared to disable further interrupts. If the IPEN bit is cleared, this is the GIE bit. If interrupt priority levels are used, this will be either the GIEH or GIEL bit. High-priority interrupt sources can interrupt a low-priority interrupt. Low-priority interrupts are not processed while high-priority interrupts are in progress.

The return address is pushed onto the stack and the PC is loaded with the interrupt vector address (0008h or 0018h). Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bits must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

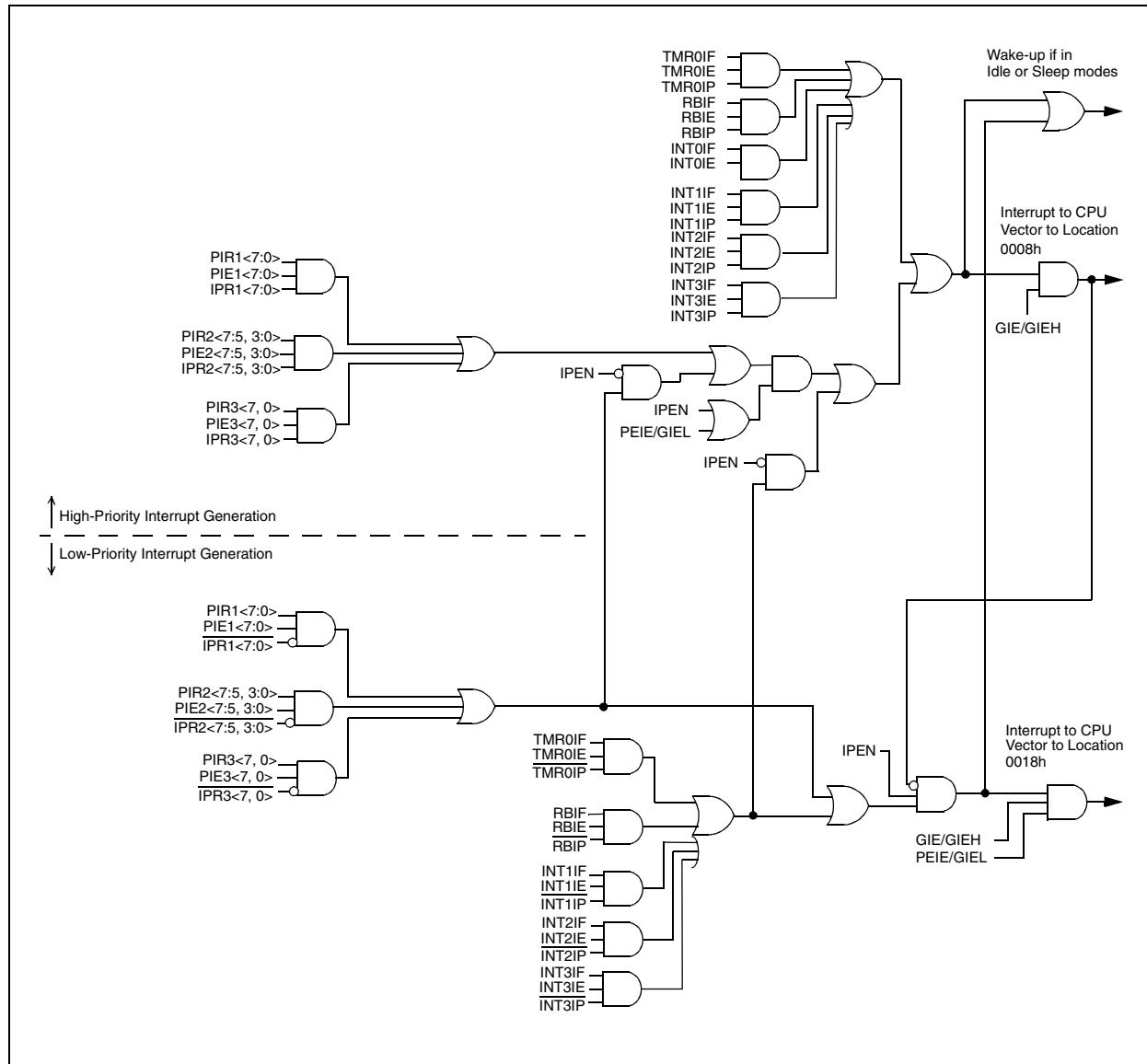
The “return from interrupt” instruction, RETFIE, exits the interrupt routine and sets the GIE bit (GIEH or GIEL if priority levels are used) which re-enables interrupts.

For external interrupt events, such as the INTx pins or the PORTB input change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency is the same for one or two-cycle instructions. Individual interrupt flag bits are set regardless of the status of their corresponding enable bit or the GIE bit.

Note: Do not use the MOVFF instruction to modify any of the interrupt control registers while any interrupt is enabled. Doing so may cause erratic microcontroller behavior.

PIC18F87J11 FAMILY

FIGURE 9-1: PIC18F87J11 FAMILY INTERRUPT LOGIC



9.1 INTCON Registers

The INTCON registers are readable and writable registers which contain various enable, priority and flag bits.

Note: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global interrupt enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER

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/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF ⁽¹⁾
bit 7	bit 0						

Legend:

R = Readable bit
-n = Value at POR

W = Writable bit
'1' = Bit is set

U = Unimplemented bit, read as '0'
'0' = Bit is cleared
x = Bit is unknown

bit 7 **GIE/GIEH:** Global Interrupt Enable bit

When IPEN = 0:

1 = Enables all unmasked interrupts
0 = Disables all interrupts

When IPEN = 1:

1 = Enables all high-priority interrupts
0 = Disables all interrupts

bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit

When IPEN = 0:

1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts

When IPEN = 1:

1 = Enables all low-priority peripheral interrupts
0 = Disables all low-priority peripheral interrupts

bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit

1 = Enables the TMR0 overflow interrupt
0 = Disables the TMR0 overflow interrupt

bit 4 **INT0IE:** INT0 External Interrupt Enable bit

1 = Enables the INT0 external interrupt
0 = Disables the INT0 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 **INT0IF:** INT0 External Interrupt Flag bit

1 = The INT0 external interrupt occurred (must be cleared in software)
0 = The INT0 external interrupt did not occur

bit 0 **RBIF:** RB Port Change Interrupt Flag bit⁽¹⁾

1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

Note 1: A mismatch condition will continue to set this bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared.

PIC18F87J11 FAMILY

REGISTER 9-2: INTCON2: INTERRUPT CONTROL REGISTER 2

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	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP
bit 7	bit 0						

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

- bit 7 **RBPU:** PORTB Pull-up Enable bit
1 = All PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values
- bit 6 **INTEDG0:** External Interrupt 0 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge
- bit 5 **INTEDG1:** External Interrupt 1 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge
- bit 4 **INTEDG2:** External Interrupt 2 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge
- bit 3 **INTEDG3:** External Interrupt 3 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge
- bit 2 **TMR0IP:** TMR0 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 1 **INT3IP:** INT3 External Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 0 **RBIP:** RB Port Change Interrupt Priority bit
1 = High priority
0 = Low priority

Note: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global interrupt enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

REGISTER 9-3: INTCON3: INTERRUPT CONTROL REGISTER 3

R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF
bit 7	bit 0						

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

- | | |
|-------|--|
| bit 7 | INT2IP: INT2 External Interrupt Priority bit
1 = High priority
0 = Low priority |
| bit 6 | INT1IP: INT1 External Interrupt Priority bit
1 = High priority
0 = Low priority |
| bit 5 | INT3IE: INT3 External Interrupt Enable bit
1 = Enables the INT3 external interrupt
0 = Disables the INT3 external interrupt |
| bit 4 | INT2IE: INT2 External Interrupt Enable bit
1 = Enables the INT2 external interrupt
0 = Disables the INT2 external interrupt |
| bit 3 | INT1IE: INT1 External Interrupt Enable bit
1 = Enables the INT1 external interrupt
0 = Disables the INT1 external interrupt |
| bit 2 | INT3IF: INT3 External Interrupt Flag bit
1 = The INT3 external interrupt occurred (must be cleared in software)
0 = The INT3 external interrupt did not occur |
| bit 1 | INT2IF: INT2 External Interrupt Flag bit
1 = The INT2 external interrupt occurred (must be cleared in software)
0 = The INT2 external interrupt did not occur |
| bit 0 | INT1IF: INT1 External Interrupt Flag bit
1 = The INT1 external interrupt occurred (must be cleared in software)
0 = The INT1 external interrupt did not occur |

Note: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global interrupt enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

PIC18F87J11 FAMILY

9.2 PIR Registers

The PIR registers contain the individual flag bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are three Peripheral Interrupt Request (Flag) registers (PIR1, PIR2, PIR3).

Note 1: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit, GIE (INTCON<7>).

2: User software should ensure the appropriate interrupt flag bits are cleared prior to enabling an interrupt and after servicing that interrupt.

REGISTER 9-4: PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF
bit 7	bit 0						

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

bit 7

PMPIF: Parallel Master Port Read/Write Interrupt Flag bit

1 = A read or a write operation has taken place (must be cleared in software)
0 = No read or write has occurred

bit 6

ADIF: A/D Converter Interrupt Flag bit

1 = An A/D conversion completed (must be cleared in software)
0 = The A/D conversion is not complete

bit 5

RC1IF: EUSART1 Receive Interrupt Flag bit

1 = The EUSART1 receive buffer, RCREG1, is full (cleared when RCREG1 is read)
0 = The EUSART1 receive buffer is empty

bit 4

TX1IF: EUSART1 Transmit Interrupt Flag bit

1 = The EUSART1 transmit buffer, TXREG1, is empty (cleared when TXREG1 is written)
0 = The EUSART1 transmit buffer is full

bit 3

SSP1IF: MSSP1 Interrupt Flag bit

1 = The transmission/reception is complete (must be cleared in software)
0 = Waiting to transmit/receive

bit 2

CCP1IF: ECCP1 Interrupt Flag bit

Capture mode:

1 = A TMR1/TMR3 register capture occurred (must be cleared in software)
0 = No TMR1/TMR3 register capture occurred

Compare mode:

1 = A TMR1/TMR3 register compare match occurred (must be cleared in software)
0 = No TMR1/TMR3 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

REGISTER 9-5: PIR2: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF
bit 7							bit 0

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

- bit 7 **OSCFIF:** Oscillator Fail Interrupt Flag bit
 1 = Device oscillator failed, clock input has changed to INTOSC (must be cleared in software)
 0 = Device clock operating
- bit 6 **CM2IF:** Comparator 2 Interrupt Flag bit
 1 = Comparator input has changed (must be cleared in software)
 0 = Comparator input has not changed
- bit 5 **CM1IF:** Comparator 1 Interrupt Flag bit
 1 = Comparator input has changed (must be cleared in software)
 0 = Comparator input has not changed
- bit 4 **Unimplemented:** Read as '0'
- bit 3 **BCL1IF:** Bus Collision Interrupt Flag bit (MSSP1 module)
 1 = A bus collision occurred (must be cleared in software)
 0 = No bus collision occurred
- bit 2 **LVDIF:** Low-Voltage Detect Interrupt Flag bit
 1 = A low-voltage condition occurred (must be cleared in software)
 0 = VDDCORE has not fallen below the low-voltage trip point (about 2.45V)
- bit 1 **TMR3IF:** TMR3 Overflow Interrupt Flag bit
 1 = TMR3 register overflowed (must be cleared in software)
 0 = TMR3 register did not overflow
- bit 0 **CCP2IF:** ECCP2 Interrupt Flag bit
Capture mode:
 1 = A TMR1/TMR3 register capture occurred (must be cleared in software)
 0 = No TMR1/TMR3 register capture occurred
Compare mode:
 1 = A TMR1/TMR3 register compare match occurred (must be cleared in software)
 0 = No TMR1/TMR3 register compare match occurred
PWM mode:
 Unused in this mode.

PIC18F87J11 FAMILY

REGISTER 9-6: PIR3: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 3

R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF
bit 7	bit 0						

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

bit 7

SSP2IF: MSSP2 Interrupt Flag bit

1 = The transmission/reception is complete (must be cleared in software)

0 = Waiting to transmit/receive

bit 6

BCL2IF: Bus Collision Interrupt Flag bit (MSSP2 module)

1 = A bus collision occurred (must be cleared in software)

0 = No bus collision occurred

bit 5

RC2IF: EUSART2 Receive Interrupt Flag bit

1 = The EUSART2 receive buffer, RCREG2, is full (cleared when RCREG2 is read)

0 = The EUSART2 receive buffer is empty

bit 4

TX2IF: EUSART2 Transmit Interrupt Flag bit

1 = The EUSART2 transmit buffer, TXREG2, is empty (cleared when TXREG2 is written)

0 = The EUSART2 transmit buffer is full

bit 3

TMR4IF: TMR4 to PR4 Match Interrupt Flag bit

1 = TMR4 to PR4 match occurred (must be cleared in software)

0 = No TMR4 to PR4 match occurred

bit 2

CCP5IF: CCP5 Interrupt Flag bit

Capture mode:

1 = A TMR1/TMR3 register capture occurred (must be cleared in software)

0 = No TMR1/TMR3 register capture occurred

Compare mode:

1 = A TMR1/TMR3 register compare match occurred (must be cleared in software)

0 = No TMR1/TMR3 register compare match occurred

PWM mode:

Unused in this mode.

bit 1

CCP4IF: CCP4 Interrupt Flag bit

Capture mode:

1 = A TMR1/TMR3 register capture occurred (must be cleared in software)

0 = No TMR1/TMR3 register capture occurred

Compare mode:

1 = A TMR1/TMR3 register compare match occurred (must be cleared in software)

0 = No TMR1/TMR3 register compare match occurred

PWM mode:

Unused in this mode.

bit 0

CCP3IF: ECCP3 Interrupt Flag bit

Capture mode:

1 = A TMR1/TMR3 register capture occurred (must be cleared in software)

0 = No TMR1/TMR3 register capture occurred

Compare mode:

1 = A TMR1/TMR3 register compare match occurred (must be cleared in software)

0 = No TMR1/TMR3 register compare match occurred

PWM mode:

Unused in this mode.

9.3 PIE Registers

The PIE registers contain the individual enable bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are three Peripheral Interrupt Enable registers (PIE1, PIE2, PIE3). When IPEN = 0, the PEIE bit must be set to enable any of these peripheral interrupts.

REGISTER 9-7: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE
bit 7							

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

- bit 7 **PMPIE:** Parallel Master Port Read/Write Interrupt Enable bit
1 = Enables the PM read/write interrupt
0 = Disables the PM read/write interrupt
- bit 6 **ADIE:** A/D Converter Interrupt Enable bit
1 = Enables the A/D interrupt
0 = Disables the A/D interrupt
- bit 5 **RC1IE:** EUSART1 Receive Interrupt Enable bit
1 = Enables the EUSART1 receive interrupt
0 = Disables the EUSART1 receive interrupt
- bit 4 **TX1IE:** EUSART1 Transmit Interrupt Enable bit
1 = Enables the EUSART1 transmit interrupt
0 = Disables the EUSART1 transmit interrupt
- bit 3 **SSP1IE:** MSSP1 Interrupt Enable bit
1 = Enables the MSSP1 interrupt
0 = Disables the MSSP1 interrupt
- bit 2 **CCP1IE:** ECCP1 Interrupt Enable bit
1 = Enables the ECCP1 interrupt
0 = Disables the ECCP1 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

PIC18F87J11 FAMILY

REGISTER 9-8: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE
bit 7	bit 0						

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

bit 7 **OSCFIE:** Oscillator Fail Interrupt Enable bit

1 = Enabled

0 = Disabled

bit 6 **CM2IE:** Comparator 2 Interrupt Enable bit

1 = Enabled

0 = Disabled

bit 5 **CM1IE:** Comparator 1 Interrupt Enable bit

1 = Enabled

0 = Disabled

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

bit 3 **BCL1IE:** Bus Collision Interrupt Enable bit (MSSP1 module)

1 = Enabled

0 = Disabled

bit 2 **LVDIE:** Low-Voltage Detect Interrupt Enable bit

1 = Enabled

0 = Disabled

bit 1 **TMR3IE:** TMR3 Overflow Interrupt Enable bit

1 = Enabled

0 = Disabled

bit 0 **CCP2IE:** ECCP2 Interrupt Enable bit

1 = Enabled

0 = Disabled

REGISTER 9-9: PIE3: PERIPHERAL INTERRUPT ENABLE REGISTER 3

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE
bit 7							

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

- bit 7 **SSP2IE:** MSSP2 Interrupt Enable bit
1 = Enabled
0 = Disabled
- bit 6 **BCL2IE:** Bus Collision Interrupt Enable bit (MSSP2 module)
1 = Enabled
0 = Disabled
- bit 5 **RC2IE:** EUSART2 Receive Interrupt Enable bit
1 = Enabled
0 = Disabled
- bit 4 **TX2IE:** EUSART2 Transmit Interrupt Enable bit
1 = Enabled
0 = Disabled
- bit 3 **TMR4IE:** TMR4 to PR4 Match Interrupt Enable bit
1 = Enabled
0 = Disabled
- bit 2 **CCP5IE:** CCP5 Interrupt Enable bit
1 = Enabled
0 = Disabled
- bit 1 **CCP4IE:** CCP4 Interrupt Enable bit
1 = Enabled
0 = Disabled
- bit 0 **CCP3IE:** ECCP3 Interrupt Enable bit
1 = Enabled
0 = Disabled

PIC18F87J11 FAMILY

9.4 IPR Registers

The IPR registers contain the individual priority bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are three Peripheral Interrupt Priority registers (IPR1, IPR2, IPR3). Using the priority bits requires that the Interrupt Priority Enable (IPEN) bit be set.

REGISTER 9-10: IPR1: PERIPHERAL INTERRUPT PRIORITY REGISTER 1

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP
bit 7						bit 0	

Legend:

R = Readable bit
-n = Value at POI

W = Writable bit
‘1’ = Bit is set

U = Unimplemented bit, read as '0'
'0' = Bit is cleared x = Bit is set

x = Bit is unknown

- | | |
|-------|---|
| bit 7 | PMPIP: Parallel Master Port Read/Write Interrupt Priority bit
1 = High priority
0 = Low priority |
| bit 6 | ADIP: A/D Converter Interrupt Priority bit
1 = High priority
0 = Low priority |
| bit 5 | RC1IP: EUSART1 Receive Interrupt Priority bit
1 = High priority
0 = Low priority |
| bit 4 | TX1IP: EUSART1 Transmit Interrupt Priority bit
1 = High priority
0 = Low priority |
| bit 3 | SSP1IP: MSSP1 Interrupt Priority bit
1 = High priority
0 = Low priority |
| bit 2 | CCP1IP: ECCP1 Interrupt Priority bit
1 = High priority
0 = Low priority |
| bit 1 | TMR2IP: TMR2 to PR2 Match Interrupt Priority bit
1 = High priority
0 = Low priority |
| bit 0 | TMR1IP: TMR1 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority |

REGISTER 9-11: IPR2: PERIPHERAL INTERRUPT PRIORITY REGISTER 2

R/W-1	R/W-1	R/W-1	U-0	R/W-1	R/W-1	R/W-1	R/W-1
OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP
bit 7							bit 0

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

- bit 7 **OSCFIP:** Oscillator Fail Interrupt Priority bit
 1 = High priority
 0 = Low priority
- bit 6 **CM2IP:** Comparator 2 Interrupt Priority bit
 1 = High priority
 0 = Low priority
- bit 5 **C12IP:** Comparator 1 Interrupt Priority bit
 1 = High priority
 0 = Low priority
- bit 4 **Unimplemented:** Read as '0'
- bit 3 **BCL1IP:** Bus Collision Interrupt Priority bit (MSSP1 module)
 1 = High priority
 0 = Low priority
- bit 2 **LVDIP:** Low-Voltage Detect Interrupt Priority bit
 1 = High priority
 0 = Low priority
- bit 1 **TMR3IP:** TMR3 Overflow Interrupt Priority bit
 1 = High priority
 0 = Low priority
- bit 0 **CCP2IP:** CCP2 Interrupt Priority bit
 1 = High priority
 0 = Low priority

PIC18F87J11 FAMILY

REGISTER 9-12: IPR3: PERIPHERAL INTERRUPT PRIORITY REGISTER 3

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP
bit 7	bit 0						

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

bit 7 **SSP2IP:** MSSP2 Interrupt Priority bit

1 = High priority

0 = Low priority

bit 6 **BCL2IP:** Bus Collision Interrupt Priority bit (MSSP2 module)

1 = High priority

0 = Low priority

bit 5 **RC2IP:** EUSART2 Receive Interrupt Priority bit

1 = High priority

0 = Low priority

bit 4 **TX2IP:** EUSART2 Transmit Interrupt Priority bit

1 = High priority

0 = Low priority

bit 3 **TMR4IE:** TMR4 to PR4 Interrupt Priority bit

1 = High priority

0 = Low priority

bit 2 **CCP5IP:** CCP5 Interrupt Priority bit

1 = High priority

0 = Low priority

bit 1 **CCP4IP:** CCP4 Interrupt Priority bit

1 = High priority

0 = Low priority

bit 0 **CCP3IP:** ECCP3 Interrupt Priority bit

1 = High priority

0 = Low priority

9.5 RCON Register

The RCON register contains bits used to determine the cause of the last Reset or wake-up from Idle or Sleep modes. RCON also contains the bit that enables interrupt priorities (IPEN).

REGISTER 9-13: RCON: RESET CONTROL REGISTER

R/W-0	U-0	R/W-1	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	\overline{CM}	\overline{RI}	\overline{TO}	\overline{PD}	\overline{POR}	\overline{BOR}
bit 7	bit 0						

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

- bit 7 **IPEN:** Interrupt Priority Enable bit
1 = Enable priority levels on interrupts
0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)
- bit 6 **Unimplemented:** Read as '0'
- bit 5 **CM:** Configuration Mismatch Flag bit
For details of bit operation, see Register 4-1.
- bit 4 **RI:** RESET Instruction Flag bit
For details of bit operation, see Register 4-1.
- bit 3 **TO:** Watchdog Timer Time-out Flag bit
For details of bit operation, see Register 4-1.
- bit 2 **PD:** Power-Down Detection Flag bit
For details of bit operation, see Register 4-1.
- bit 1 **POR:** Power-on Reset Status bit
For details of bit operation, see Register 4-1.
- bit 0 **BOR:** Brown-out Reset Status bit
For details of bit operation, see Register 4-1.

PIC18F87J11 FAMILY

9.6 INTx Pin Interrupts

External interrupts on the RB0/INT0, RB1/INT1, RB2/INT2 and RB3/INT3 pins are edge-triggered. If the corresponding INTEDG_x bit in the INTCON2 register is set (= 1), the interrupt is triggered by a rising edge; if the bit is clear, the trigger is on the falling edge. When a valid edge appears on the RB_x/INT_x pin, the corresponding flag bit, INT_xIF, is set. This interrupt can be disabled by clearing the corresponding enable bit, INT_xIE. Flag bit, INT_xIF, must be cleared in software in the Interrupt Service Routine before re-enabling the interrupt.

All external interrupts (INT0, INT1, INT2 and INT3) can wake-up the processor from the power-managed modes if bit INT_xIE was set prior to going into the power-managed modes. If the Global Interrupt Enable bit, GIE, is set, the processor will branch to the interrupt vector following wake-up.

Interrupt priority for INT1, INT2 and INT3 is determined by the value contained in the interrupt priority bits, INT1IP (INTCON3<6>), INT2IP (INTCON3<7>) and INT3IP (INTCON2<1>). There is no priority bit associated with INT0. It is always a high-priority interrupt source.

9.7 TMR0 Interrupt

In 8-bit mode (which is the default), an overflow in the TMR0 register (FFh → 00h) will set flag bit, TMR0IF. In 16-bit mode, an overflow in the TMR0H:TMR0L register pair (FFFFh → 0000h) will set TMR0IF. The interrupt can be enabled/disabled by setting/clearing enable bit, TMR0IE (INTCON<5>). Interrupt priority for Timer0 is determined by the value contained in the interrupt priority bit, TMR0IP (INTCON2<2>). See **Section 12.0 “Timer0 Module”** for further details on the Timer0 module.

9.8 PORTB Interrupt-on-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<3>). Interrupt priority for PORTB interrupt-on-change is determined by the value contained in the interrupt priority bit, RBIP (INTCON2<0>).

9.9 Context Saving During Interrupts

During interrupts, the return PC address is saved on the stack. Additionally, the WREG, STATUS and BSR registers are saved on the Fast Return Stack. If a fast return from interrupt is not used (see **Section 5.3 “Data Memory Organization”**), the user may need to save the WREG, STATUS and BSR registers on entry to the Interrupt Service Routine. Depending on the user's application, other registers may also need to be saved. Example 9-1 saves and restores the WREG, STATUS and BSR registers during an Interrupt Service Routine.

EXAMPLE 9-1: SAVING STATUS, WREG AND BSR REGISTERS IN RAM

```
MOVWF    W_TEMP           ; W_TEMP is in virtual bank
MOVFF    STATUS, STATUS_TEMP ; STATUS_TEMP located anywhere
MOVFF    BSR, BSR_TEMP     ; BSR_TMEP located anywhere
;
; USER ISR CODE
;
MOVFF    BSR_TEMP, BSR      ; Restore BSR
MOVFF    W_TEMP, W          ; Restore WREG
MOVFF    STATUS_TEMP, STATUS ; Restore STATUS
```

10.0 I/O PORTS

Depending on the device selected and features enabled, there are up to nine ports available. Some pins of the I/O ports are multiplexed with an alternate function from 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.

Each port has three memory-mapped registers for its operation:

- TRIS register (Data Direction register)
- PORT register (reads the levels on the pins of the device)
- LAT register (Output Latch register)

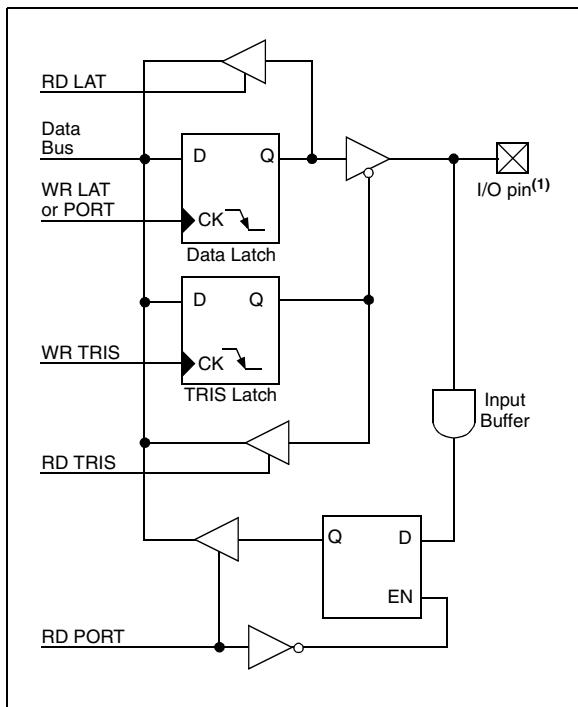
Reading the PORT register reads the current status of the pins, whereas writing to the PORT register writes to the Output Latch (LAT) register.

Setting a TRIS bit (= 1) makes the corresponding PORT pin an input (i.e., puts the corresponding output driver in a high-impedance mode). Clearing a TRIS bit (= 0) makes the corresponding PORT pin an output (i.e., puts the contents of the corresponding LAT bit on the selected pin).

The Output Latch (LAT register) is useful for read-modify-write operations on the value that the I/O pins are driving. Read-modify-write operations on the LAT register read and write the latched output value for the PORT register.

A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 10-1.

FIGURE 10-1: GENERIC I/O PORT OPERATION



10.1 I/O Port Pin Capabilities

When developing an application, the capabilities of the port pins must be considered. Outputs on some pins have higher output drive strength than others. Similarly, some pins can tolerate higher than VDD input levels.

10.1.1 INPUT PINS AND VOLTAGE CONSIDERATIONS

The voltage tolerance of pins used as device inputs is dependent on the pin's input function. Pins that are used as digital only inputs are able to handle DC voltages up to 5.5V, a level typical for digital logic circuits. In contrast, pins that also have analog input functions of any kind (such as A/D and comparator inputs) can only tolerate voltages up to VDD. Voltage excursions beyond VDD on these pins should be avoided.

Table 10-1 summarizes the input capabilities. Refer to **Section 27.0 “Electrical Characteristics”** for more details.

TABLE 10-1: INPUT VOLTAGE LEVELS

Port or Pin	Tolerated Input	Description
PORTE<7:0>	VDD	Only VDD input levels tolerated.
PORTE<7:0>		
PORTE<7:0>		
PORTE<7:0> ⁽¹⁾		
PORTE<7:0>	5.5V	Tolerates input levels above VDD, useful for most standard logic.
PORTE<7:0>		

Note 1: These ports are not available on 64-pin devices.

10.1.2 PIN OUTPUT DRIVE

When used as digital I/O, the output pin drive strengths vary for groups of pins intended to meet the needs for a variety of applications. In general, there are three classes of output pins in terms of drive capability.

PORTE and PORTC, as well as PORTA<7:6>, are designed to drive higher current loads, such as LEDs. PORTD, PORTE and PORTJ are capable of driving digital circuits associated with external memory devices; they can also drive LEDs, but only those with smaller current requirements. PORTF, PORTG and PORTH, along with PORTA<5:0>, have the lowest drive level, but are capable of driving normal digital circuit loads with a high input impedance.

PIC18F87J11 FAMILY

Table 10-2 summarizes the output capabilities of the ports. Refer to the “**Absolute Maximum Ratings**” in **Section 27.0 “Electrical Characteristics”** for more details.

TABLE 10-2: OUTPUT DRIVE LEVELS

Port	Drive	Description
PORTA	Minimum	Intended for indication.
PORTF		
PORTG		
PORTH ⁽¹⁾		
PORTD	Medium	Sufficient drive levels for external memory interfacing as well as indication.
PORTE		
PORTJ ⁽¹⁾		
PORTB	High	Suitable for direct LED drive levels.
PORTC		

Note 1: These ports are not available on 64-pin devices.

10.1.3 PULL-UP CONFIGURATION

Four of the I/O ports (PORTB, PORTD, PORTE and PORTJ) implement configurable weak pull-ups on all pins. These are internal pull-ups that allow floating digital input signals to be pulled to a consistent level, without the use of external resistors.

The pull-ups are enabled with a single bit for each of the ports: RBPU (INTCON2<7>) for PORTB, and RDPU, REPU and RJPU (PORTG<7:5>) for the other ports.

10.1.4 OPEN-DRAIN OUTPUTS

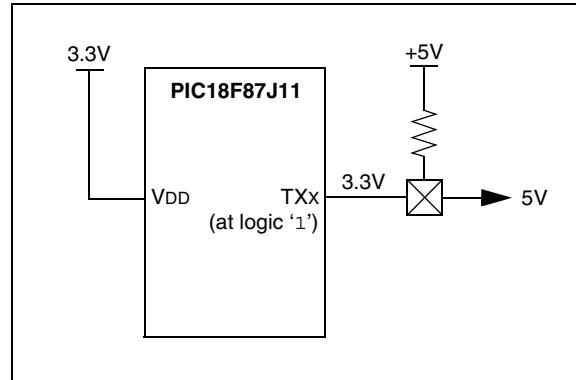
The output pins for several peripherals are also equipped with a configurable, open-drain output option. This allows the peripherals to communicate with external digital logic operating at a higher voltage level, without the use of level translators.

The open-drain option is implemented on port pins specifically associated with the data and clock outputs of the EUSARTs, the MSSP modules (in SPI mode) and the CCP and ECCP modules. It is selectively enabled by setting the open-drain control bit for the corresponding module in the ODCON registers (Register 10-1, Register 10-2 and Register 10-3). Their configuration is discussed in more detail with the individual port where these peripherals are multiplexed.

The ODCON registers all reside in the SFR configuration space and share the same SFR addresses as the Timer1 registers (see **Section 5.3.4.1 “Shared Address SFRs”** for more details). The ODCON registers are accessed by setting the ADSHR bit (WDTCON<4>).

When the open-drain option is required, the output pin must also be tied through an external pull-up resistor provided by the user to a higher voltage level, up to 5V on digital only pins (Figure 10-2). When a digital logic high signal is output, it is pulled up to the higher voltage level.

FIGURE 10-2: USING THE OPEN-DRAIN OUTPUT (EUSARTx SHOWN AS EXAMPLE)



10.1.5 TTL INPUT BUFFER OPTION

Many of the digital I/O ports use Schmitt Trigger (ST) input buffers. While this form of buffering works well with many types of input, some applications may require TTL-level signals to interface with external logic devices. This is particularly true with the EMB and the Parallel Master Port (PMP), which are particularly likely to be interfaced to TTL-level logic or memory devices.

The inputs for the PMP can be optionally configured for TTL buffers with the PMPTTL bit in the PADCFG1 register (Register 10-4). Setting this bit configures all data and control input pins for the PMP to use TTL buffers. By default, these PMP inputs use the port’s ST buffers.

As with the ODCON registers, the PADCFG1 register resides in the SFR configuration space; it shares the same memory address as the TMR2 register. PADCFG1 is accessed by setting the ADSHR bit (WDTCON<4>).

REGISTER 10-1: ODCON1: PERIPHERAL OPEN-DRAIN CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	CCP5OD	CCP4OD	ECCP3OD	ECCP2OD	ECCP1OD
bit 7	bit 0						

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

bit 7-5 **Unimplemented:** Read as '0'

bit 4-3 **CCP5OD:CCP4OD:** CCPx Open-Drain Output Enable bits

1 = Open-drain output on CCPx pin (Capture/PWM modes) enabled
0 = Open-drain output disabled

bit 2-0 **ECCP3OD:ECCP1OD:** ECCPx Open-Drain Output Enable bits

1 = Open-drain output on ECCPx pin (Capture mode) enabled
0 = Open-drain output disabled

REGISTER 10-2: ODCON2: PERIPHERAL OPEN-DRAIN CONTROL REGISTER 2

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	U2OD	U1OD
bit 7	bit 0						

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

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

bit 1-0 **U2OD:U1OD:** EUSARTx Open-Drain Output Enable bits

1 = Open-drain output on TXx pin enabled
0 = Open-drain output disabled

REGISTER 10-3: ODCON3: PERIPHERAL OPEN-DRAIN CONTROL REGISTER 3

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	SPI2OD	SPI1OD
bit 7	bit 0						

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

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

bit 1-0 **SPI2OD:SPI1OD:** SPI Open-Drain Output Enable bits

1 = Open-drain output on SDOx pin enabled
0 = Open-drain output disabled

PIC18F87J11 FAMILY

REGISTER 10-4: PADCFG1: I/O PAD CONFIGURATION CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	PMPTTL
bit 7							bit 0

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

bit 7-1 **Unimplemented:** Read as '0'bit 0 **PMPTTL:** PMP Module TTL Input Buffer Select bit

1 = PMP module uses TTL input buffers

0 = PMP module uses Schmitt Trigger input buffers

10.2 PORTA, TRISA and LATA Registers

PORTA is an 8-bit wide, bidirectional port. It may function as a 6-bit or 7-bit port, depending on the oscillator mode selected. The corresponding Data Direction and Output Latch registers are TRISA and LATA.

The RA4 pin is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin; it is also multiplexed as the Parallel Master Port data pin (in 80-pin devices). The other PORTA pins are multiplexed with the analog VREF+ and VREF- inputs. The operation of pins, RA<5:3:0>, as A/D Converter inputs is selected by clearing or setting the appropriate PCFG control bits in the ANCON0 register.

- Note 1:** RA5 (RA5/PMD4/AN4) is multiplexed as an analog input in all devices and Parallel Master Port data in 80-pin devices.
- 2:** RA5 and RA3:RA0 are configured as analog inputs on any Reset and are read as '0'. RA4 is configured as a digital input.

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

The TRISA register controls the direction of the PORTA 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.

OSC2/CLKO/RA6 and OSC1/CLK1/RA7 normally serve as the external circuit connections for the external (primary) oscillator circuit (HS and HSPLL Oscillator modes), or the external clock input (EC and ECPLL Oscillator modes). In these cases, RA6 and RA7 are not available as digital I/O, and their corresponding TRIS and LAT bits are read as '0'.

For INTIO and INTPLL Oscillator modes (FOSC2 Configuration bit is '0'), either RA7 or both RA6 and RA7 automatically become available as digital I/O, depend-

ing on the oscillator mode selected. When RA6 is not configured as a digital I/O, in these cases, it provides a clock output at Fosc/4. A list of the possible configurations for RA6 and RA7, based on oscillator mode, is provided in Table 10-3. For these pins, the corresponding PORTA, TRISA and LATA bits are only defined when the pins are configured as I/O.

TABLE 10-3: FUNCTION OF RA7:RA6 IN INTIO AND INTPLL MODES

Oscillator Mode (FOSC2:FOSC0 Configuration)	RA6	RA7
INTPLL1 (011)	CLKO	I/O
INTPLL2 (010)	I/O	I/O
INTIO1 (001)	CLKO	I/O
INTIO2 (000)	I/O	I/O

Legend: CLKO = Fosc/4 clock output;
I/O = digital port.

EXAMPLE 10-1: INITIALIZING PORTA

```
CLRF  PORTA      ; Initialize PORTA by
                  ; clearing output
                  ; data latches
CLRF  LATA       ; Alternate method to
                  ; clear data latches
BSF   WDTCON,ADSHR ; Enable write/read to
                  ; the shared SFR
MOVLW 1Fh        ; Configure A/D
MOVWF  ANCON0     ; for digital inputs
BCF   WDTCON,ADSHR ; Disable write/read
                  ; to the shared SFR
MOVLW 0CFh        ; Value used to
                  ; initialize
                  ; data direction
MOVWF  TRISA      ; Set RA<3:0> as inputs,
                  ; RA<5:4> as outputs
```

TABLE 10-4: PORTA FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RA0/AN0	RA0	0	O	DIG	LATA<0> data output; not affected by analog input.
		1	I	TTL	PORTA<0> data input; disabled when analog input enabled.
	AN0	1	I	ANA	A/D input channel 0. Default input configuration on POR; does not affect digital output.
RA1/AN1	RA1	0	O	DIG	LATA<1> data output; not affected by analog input.
		1	I	TTL	PORTA<1> data input; disabled when analog input enabled.
	AN1	1	I	ANA	A/D input channel 1. Default input configuration on POR; does not affect digital output.
RA2/AN2/VREF-	RA2	0	O	DIG	LATA<2> data output; not affected by analog input. Disabled when CVREF output enabled.
		1	I	TTL	PORTA<2> data input. Disabled when analog functions enabled; disabled when CVREF output enabled.
	AN2	1	I	ANA	A/D input channel 2. Default input configuration on POR; not affected by analog output.
	VREF-	1	I	ANA	A/D low reference voltage input.
RA3/AN3/VREF+	RA3	0	O	DIG	LATA<3> data output; not affected by analog input.
		1	I	TTL	PORTA<3> data input; disabled when analog input enabled.
	AN3	1	I	ANA	A/D input channel 3. Default input configuration on POR.
	VREF+	1	I	ANA	A/D high reference voltage input.
RA4/PMD5/T0CKI/	RA4	0	O	DIG	LATA<4> data output.
		1	I	ST	PORTA<4> data input; default configuration on POR.
	PMD5 ⁽¹⁾	x	O	DIG	Parallel Master Port data output.
		x	I	TTL	Parallel Master Port data output.
	T0CKI	x	I	ST	Timer0 clock input.
RA5/PMD4/AN4	RA5	0	O	DIG	LATA<5> data output; not affected by analog input.
		1	I	TTL	PORTA<5> data input; disabled when analog input enabled.
	PMD4 ⁽¹⁾	x	O	DIG	Parallel Master Port data output.
		x	I	TTL	Parallel Master Port data output.
	AN4	1	I	ANA	A/D input channel 4. Default configuration on POR.
OSC2/CLKO/RA6	OSC2	x	O	ANA	Main oscillator feedback output connection (HS and HSPLL modes).
	CLKO	x	O	DIG	System cycle clock output, Fosc/4 (EC, ECPLL, INTIO1 and INTPLL1 modes).
	RA6	0	O	DIG	LATA<6> data output; disabled when FOSC2 Configuration bit is set.
		1	I	TTL	PORTA<6> data input; disabled when FOSC2 Configuration bit is set.
OSC1/CLKI/RA7	OSC1	x	I	ANA	Main oscillator input connection (HS and HSPLL modes).
	CLKI	x	I	ANA	Main external clock source input (EC and ECPLL modes).
	RA7	0	O	DIG	LATA<7> data output; disabled when FOSC2 Configuration bit is set.
		1	I	TTL	PORTA<7> data input; disabled when FOSC2 Configuration bit is set.

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input,
TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: Alternate PMP configuration when the PMPMX Configuration bit is '0'; available on 80-pin devices only.

PIC18F87J11 FAMILY

TABLE 10-5: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTA	RA7 ⁽¹⁾	RA6 ⁽¹⁾	RA5	RA4	RA3	RA2	RA1	RA0	59
LATA	LATA7 ⁽¹⁾	LATA6 ⁽¹⁾	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	58
TRISA	TRISA7 ⁽¹⁾	TRISA6 ⁽¹⁾	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	58
ANCON0 ⁽²⁾	PCFG7	PCFG6	—	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	57

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PORTA.

Note 1: Implemented only in specific oscillator modes (FOSC2 Configuration bit = 0); otherwise read as '0'.

2: Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.

10.3 PORTB, TRISB and LATB Registers

PORTB is an 8-bit wide, bidirectional port. The corresponding Data Direction register is TRISB. All pins on PORTB are digital only and tolerate voltages up to 5.5V.

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 (INTCON2<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.

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 ORed together to generate the RB Port Change Interrupt with Flag bit, RBIF (INTCON<0>).

This interrupt can wake the device from power-managed modes. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- Any read or write of PORTB (except with the MOVFF (ANY), PORTB instruction). 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.

For 80-pin devices, RB3 can be configured as the alternate peripheral pin for the ECCP2 module and Enhanced PWM output 2A by clearing the CCP2MX Configuration bit. This applies only to 80-pin devices operating in Extended Microcontroller mode. If the device is in Microcontroller mode, the alternate assignment for ECCP2 is RE7. As with other ECCP2 configurations, the user must ensure that the TRISB<3> bit is set appropriately for the intended operation. Ports, RB1, RB2, RB3, RB4 and RB5, are multiplexed with the Parallel Master Port address.

EXAMPLE 10-2: INITIALIZING PORTB

```
CLRF    PORTB    ; Initialize PORTB by
                  ; clearing output
                  ; data latches
CLRF    LATB     ; Alternate method to clear
                  ; output data latches
MOVLW  0CFh    ; Value used to initialize
                  ; data direction
MOVWF  TRISB    ; Set RB<3:0> as inputs
                  ; RB<5:4> as outputs
                  ; RB<7:6> as inputs
```

PIC18F87J11 FAMILY

TABLE 10-6: PORTB FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RB0/INT0/FLT0	RB0	0	O	DIG	LATB<0> data output.
		1	I	TTL	PORTB<0> data input; weak pull-up when RBPU bit is cleared.
	INT0	1	I	ST	External interrupt 0 input.
	FLT0	1	I	ST	Enhanced PWM Fault input (ECCP1 module); enabled in software.
RB1/INT1/PMA4	RB1	0	O	DIG	LATB<1> data output.
		1	I	TTL	PORTB<1> data input; weak pull-up when RBPU bit is cleared.
	INT1	1	I	ST	External interrupt 1 input.
	PMA4	x	O	—	Parallel Master Port address out.
RB2/INT2/PMA3	RB2	0	O	DIG	LATB<2> data output.
		1	I	TTL	PORTB<2> data input; weak pull-up when RBPU bit is cleared.
	INT2	1	I	ST	External interrupt 2 input.
	PMA3	x	O	—	Parallel Master Port address out.
RB3/INT3/PMA2/ECCP2/P2A	RB3	0	O	DIG	LATB<3> data output.
		1	I	TTL	PORTB<3> data input; weak pull-up when RBPU bit is cleared.
	INT3	1	I	ST	External interrupt 3 input.
	PMA2	x	O	—	Parallel Master Port address out.
	ECCP2 ⁽¹⁾	0	O	DIG	ECCP2 compare output and CCP2 PWM output; takes priority over port data.
		1	I	ST	ECCP2 capture input.
	P2A ⁽¹⁾	0	O	DIG	ECCP2 Enhanced PWM output, channel A. May be configured for tri-state during Enhanced PWM shutdown events. Takes priority over port data.
RB4/KBI0/PMA1	RB4	0	O	DIG	LATB<4> data output.
		1	I	TTL	PORTB<4> data input; weak pull-up when RBPU bit is cleared.
	KBI0		I	TTL	Interrupt-on-pin change.
	PMA1	x	O	—	Parallel Master Port address out.
RB5/KBI1/PMA0	RB5	0	O	DIG	LATB<5> data output.
		1	I	TTL	PORTB<5> data input; weak pull-up when RBPU bit is cleared.
	KBI1		I	TTL	Interrupt-on-pin change.
	PMA0	x	O	—	Parallel Master Port address out.
RB6/KBI2/PGC	RB6	0	O	DIG	LATB<6> data output.
		1	I	TTL	PORTB<6> data input; weak pull-up when RBPU bit is cleared.
	KBI2	1	I	TTL	Interrupt-on-pin change.
	PGC	x	I	ST	Serial execution (ICSP™) clock input for ICSP and ICD operation. ⁽²⁾
RB7/KBI3/PGD	RB7	0	O	DIG	LATB<7> data output.
		1	I	TTL	PORTB<7> data input; weak pull-up when RBPU bit is cleared.
	KBI3	1	I	TTL	Interrupt-on-pin change.
	PGD	x	O	DIG	Serial execution data output for ICSP and ICD operation. ⁽²⁾
		x	I	ST	Serial execution data input for ICSP and ICD operation. ⁽²⁾

Legend: O = Output, I = Input, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input,
x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: Alternate assignment for ECCP2/P2A when the CCP2MX Configuration bit is cleared (Extended Microcontroller mode, 80-pin devices only). Default assignment is RC1.

2: All other pin functions are disabled when ICSP™ or ICD is enabled.

PIC18F87J11 FAMILY

TABLE 10-7: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	59
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	58
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	58
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	55
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	55

Legend: Shaded cells are not used by PORTB.

10.4 PORTC, TRISC and LATC Registers

PORTC is an 8-bit wide, bidirectional port. Only PORTC pins, RC2 through RC7, are digital only pins and can tolerate input voltages up to 5.5V.

PORTC is multiplexed with ECCP, MSSP and EUSART peripheral functions (Table 10-8). The pins have Schmitt Trigger input buffers. The pins for ECCP, SPI and EUSART are also configurable for open-drain output whenever these functions are active. Open-drain configuration is selected by setting the SPIxOD, ECCPxOD, and UxOD control bits in the ODCON registers (see **Section 10.1.3 “Pull-up Configuration”** for more information).

RC1 is normally configured as the default peripheral pin for the ECCP2 module. Assignment of ECCP2 is controlled by Configuration bit, CCP2MX (default state, CCP2MX = 1).

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. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

Note: These pins are configured as digital inputs on any device Reset.

The contents of the TRISC register are affected by peripheral overrides. Reading TRISC always returns the current contents, even though a peripheral device may be overriding one or more of the pins.

EXAMPLE 10-3: INITIALIZING PORTC

```
CLRF    PORTC    ; Initialize PORTC by
                  ; clearing output
                  ; data latches
CLRF    LATC     ; Alternate method to clear
                  ; output data latches
MOVLW   0CFh     ; Value used to initialize
                  ; data direction
MOVWF   TRISC    ; Set RC<3:0> as inputs
                  ; RC<5:4> as outputs
                  ; RC<7:6> as inputs
```

TABLE 10-8: PORTC FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RC0/T1OSO/ T13CKI	RC0	0	O	DIG	LATC<0> data output.
		1	I	ST	PORTC<0> data input.
	T1OSO	x	O	ANA	Timer1 oscillator output; enabled when Timer1 oscillator enabled. Disables digital I/O.
	T13CKI	1	I	ST	Timer1/Timer3 counter input.
RC1/T1OSI/ ECCP2/P2A	RC1	0	O	DIG	LATC<1> data output.
		1	I	ST	PORTC<1> data input.
	T1OSI	x	I	ANA	Timer1 oscillator input; enabled when Timer1 oscillator enabled. Disables digital I/O.
	ECCP2 ⁽¹⁾	0	O	DIG	ECCP2 compare output and ECCP2 PWM output; takes priority over port data.
		1	I	ST	ECCP2 capture input.
	P2A ⁽¹⁾	0	O	DIG	ECCP2 Enhanced PWM output, channel A. May be configured for tri-state during Enhanced PWM shutdown events. Takes priority over port data.
RC2/ECCP1/ P1A	RC2	0	O	DIG	LATC<2> data output.
		1	I	ST	PORTC<2> data input.
	ECCP1	0	O	DIG	ECCP1 compare output and ECCP1 PWM output; takes priority over port data.
		1	I	ST	ECCP1 capture input.
	P1A	0	O	DIG	ECCP1 Enhanced PWM output, channel A. May be configured for tri-state during Enhanced PWM shutdown events. Takes priority over port data.
RC3/SCK1/ SCL1	RC3	0	O	DIG	LATC<3> data output.
		1	I	ST	PORTC<3> data input.
	SCK1	0	O	DIG	SPI clock output (MSSP1 module); takes priority over port data.
		1	I	ST	SPI clock input (MSSP1 module).
	SCL1	0	O	DIG	I ² C TM clock output (MSSP1 module); takes priority over port data.
		1	I	ST	I ² C clock input (MSSP1 module); input type depends on module setting.
RC4/SDI1/ SDA1	RC4	0	O	DIG	LATC<4> data output.
		1	I	ST	PORTC<4> data input.
	SDI1	1	I	ST	SPI data input (MSSP1 module).
	SDA1	1	O	DIG	I ² C data output (MSSP1 module); takes priority over port data.
		1	I	ST	I ² C data input (MSSP1 module); input type depends on module setting.
RC5/SDO1	RC5	0	O	DIG	LATC<5> data output.
		1	I	ST	PORTC<5> data input.
	SDO1	0	O	DIG	SPI data output (MSSP1 module); takes priority over port data.
RC6/TX1/CK1	RC6	0	O	DIG	LATC<6> data output.
		1	I	ST	PORTC<6> data input.
	TX1	1	O	DIG	Synchronous serial data output (EUSART1 module); takes priority over port data.
	CK1	1	O	DIG	Synchronous serial data input (EUSART1 module). User must configure as an input.
		1	I	ST	Synchronous serial clock input (EUSART1 module).
RC7/RX1/DT1	RC7	0	O	DIG	LATC<7> data output.
		1	I	ST	PORTC<7> data input.
	RX1	1	I	ST	Asynchronous serial receive data input (EUSART1 module).
	DT1	1	O	DIG	Synchronous serial data output (EUSART1 module); takes priority over port data.
		1	I	ST	Synchronous serial data input (EUSART1 module). User must configure as an input.

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input,
TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: Default assignment for ECCP2/P2A when CCP2MX Configuration bit is set.

PIC18F87J11 FAMILY

TABLE 10-9: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	59
LATC	LATC7	LATBC6	LATC5	LATCB4	LATC3	LATC2	LATC1	LATC0	58
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	58

10.5 PORTD, TRISD and LATD Registers

PORTD is an 8-bit wide, bidirectional port. All pins on PORTD are digital only and tolerate voltages up to 5.5V.

All pins on PORTD are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

Note: These pins are configured as digital inputs on any device Reset.

On 80-pin devices, PORTD is multiplexed with the system bus as part of the external memory interface. I/O port and other functions are only available when the interface is disabled by setting the EBDIS bit (MEMCON<7>). When the interface is enabled, PORTD is the low-order byte of the multiplexed address/data bus (AD7:AD0). The TRISD bits are also overridden.

PORTD is also multiplexed with the data functions of the Parallel Master Port data. In this mode, Parallel Master Port takes priority over the other digital I/O (but not the external memory bus). This multiplexing is available when PMPMX = 1. When the Parallel Master Port is active, the input buffers are TTL. For more information, refer to **Section 11.0 “Parallel Master Port”**.

Each of the PORTD pins has a weak internal pull-up. This is performed by clearing bit RDPU (PORTG<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on all device Resets.

EXAMPLE 10-4: INITIALIZING PORTD

```
CLRF    PORTD    ; Initialize PORTD by
                  ; clearing output
                  ; data latches
CLRF    LATD     ; Alternate method to clear
                  ; output data latches
MOVLW  0CFh    ; Value used to initialize
                  ; data direction
MOVWF  TRISD    ; Set RD<3:0> as inputs
                  ; RD<5:4> as outputs
                  ; RD<7:6> as inputs
```

PIC18F87J11 FAMILY

TABLE 10-10: PORTD FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RD0/AD0/ PMD0	RD0	0	O	DIG	LATD<0> data output.
		1	I	ST	PORTD<0> data input.
	AD0 ⁽²⁾	x	O	DIG	External memory interface, address/data bit 0 output. ⁽¹⁾
		x	TTL		External memory interface, data bit 0 input. ⁽¹⁾
	PMD0 ⁽³⁾	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
	RD1/AD1/ PMD1	RD1	0	O	DIG
			1	I	ST
	AD1 ⁽²⁾	x	O	DIG	External memory interface, address/data bit 1 output. ⁽¹⁾
		x	I	TTL	External memory interface, data bit 1 input. ⁽¹⁾
	PMD1 ⁽³⁾	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
	RD2/AD2/ PMD2	RD2	0	O	DIG
			1	I	ST
	AD2 ⁽²⁾	x	O	DIG	External memory interface, address/data bit 2 output. ⁽¹⁾
		x	I	TTL	External memory interface, data bit 2 input. ⁽¹⁾
	PMD2 ⁽³⁾	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
	RD3/AD3/ PMD3	RD3	0	O	DIG
			1	I	ST
	AD3 ⁽²⁾	x	O	DIG	External memory interface, address/data bit 3 output. ⁽¹⁾
		x	I	TTL	External memory interface, data bit 3 input. ⁽¹⁾
	PMD3 ⁽³⁾	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
	RD4/AD4/ PMD4/SDO2	RD4	0	O	DIG
			1	I	ST
	AD4 ⁽²⁾	x	O	DIG	External memory interface, address/data bit 4 output. ⁽¹⁾
		x	I	TTL	External memory interface, data bit 4 input. ⁽¹⁾
	PMD4 ⁽³⁾	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
		SDO2	0	O	DIG
					SPI data output (MSSP2 module); takes priority over port data.
RD5/AD5/ PMD5/SDI2/ SDA2	RD5	0	O	DIG	LATD<5> data output.
		1	I	ST	PORTD<5> data input.
	AD5 ⁽²⁾	x	O	DIG	External memory interface, address/data bit 5 output. ⁽¹⁾
		x	I	TTL	External memory interface, data bit 5 input. ⁽¹⁾
	PMD5 ⁽³⁾	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
	SDI2	1	I	ST	SPI data input (MSSP2 module).
	SDA2	1	O	DIG	I ² C™ data output (MSSP2 module); takes priority over port data.
		1	I	ST	I ² C data input (MSSP2 module); input type depends on module setting.

Legend: O = Output, I = Input, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input,
x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: External memory interface I/O takes priority over all other digital and PMP I/O.

2: Available on 80-pin devices only.

3: Default configuration for PMP (PMPMX Configuration bit = 1).

PIC18F87J11 FAMILY

TABLE 10-10: PORTD FUNCTIONS (CONTINUED)

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RD6/AD6/ PMD6/SCK2/ SCL2	RD6	0	O	DIG	LATD<6> data output.
		1	I	ST	PORTD<6> data input.
	AD6 ⁽²⁾	x	O	DIG-3	External memory interface, address/data bit 6 output. ⁽¹⁾
		x	I	TTL	External memory interface, data bit 6 input. ⁽¹⁾
	PMD6 ⁽³⁾	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
	SCK2	0	O	DIG	SPI clock output (MSSP2 module); takes priority over port data.
		1	I	ST	SPI clock input (MSSP2 module).
	SCL2	0	O	DIG	I ² C TM clock output (MSSP2 module); takes priority over port data.
		1	I	ST	I ² C clock input (MSSP2 module); input type depends on module setting.
RD7/AD7/ PMD7/SS2	RD7	0	O	DIG	LATD<7> data output.
		1	I	ST	PORTD<7> data input.
	AD7 ⁽²⁾	x	O	DIG	External memory interface, address/data bit 7 output. ⁽¹⁾
		x	I	TTL	External memory interface, data bit 7 input. ⁽¹⁾
	PMD7 ⁽³⁾	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
	SS2	x	I	TTL	Slave select input for MSSP2 module.

Legend: O = Output, I = Input, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input,
x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: External memory interface I/O takes priority over all other digital and PMP I/O.

2: Available on 80-pin devices only.

3: Default configuration for PMP (PMPMX Configuration bit = 1).

TABLE 10-11: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	59
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	58
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	58
PORTG	RDPU	REPU	RJPU ⁽¹⁾	RG4	RG3	RG2	RG1	RG0	59

Legend: Shaded cells are not used by PORTD.

Note 1: Unimplemented on 64-pin devices, read as '0'.

PIC18F87J11 FAMILY

10.6 PORTE, TRISE and LATE Registers

PORTE is an 8-bit wide, bidirectional port. All pins on PORTE are digital only and tolerate voltages up to 5.5V.

All pins on PORTE are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

Note: These pins are configured as digital inputs on any device Reset.

On 80-pin devices, PORTE is multiplexed with the system bus as part of the external memory interface. I/O port and other functions are only available when the interface is disabled, by setting the EBDIS bit (MEMCON<7>). When the interface is enabled, PORTE is the high-order byte of the multiplexed address/data bus (AD15:AD8). The TRISE bits are also overridden.

Each of the PORTE pins has a weak internal pull-up. A single control bit can turn off all the pull-ups. This is performed by clearing bit REPU (PORTG<6>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on any device Reset.

PORTE is also multiplexed with Enhanced PWM outputs B and C for ECCP1 and ECCP3 and outputs B, C and D for ECCP2. For all devices, their default assignments are on PORTE<6:0>. On 80-pin devices, the multiplexing for the outputs of ECCP1 and ECCP3 is controlled by the ECCPMX Configuration bit. Clearing this bit reassigns the P1B/P1C and P3B/P3C outputs to PORTH.

For devices operating in Microcontroller mode, pin RE7 can be configured as the alternate peripheral pin for the ECCP2 module and Enhanced PWM output 2A. This is done by clearing the CCP2MX Configuration bit.

PORTE is also multiplexed with the Parallel Master Port address lines. When PMPMX = 0, RE1 and RE0 are multiplexed with the control signals PMWR and PMRD.

RE3 can also be configured as the Reference Clock Output (REFO) from the system clock. For further details, refer to **Section 2.6 “Reference Clock Output”**.

EXAMPLE 10-5: INITIALIZING PORTE

```
CLRF    PORTE    ; Initialize PORTE by
                  ; clearing output
                  ; data latches
CLRF    LATE     ; Alternate method to clear
                  ; output data latches
MOVLW  03h     ; Value used to initialize
                  ; data direction
MOVWF  TRISE    ; Set RE<1:0> as inputs
                  ; RE<7:2> as outputs
```

TABLE 10-12: PORTE FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RE0/AD8/ PMRD/P2D	RE0	0	O	DIG	LATE<0> data output.
		1	I	ST	PORTE<0> data input.
	AD8 ⁽³⁾	x	O	DIG	External memory interface, address/data bit 8 output. ⁽²⁾
		x	I	TTL	External memory interface, data bit 8 input. ⁽²⁾
	PMRD ⁽⁵⁾	x	O	DIG	Parallel Master Port read strobe pin.
		x	I	TTL	Parallel Master Port read pin.
	P2D	0	O	DIG	ECCP2 Enhanced PWM output, channel D; takes priority over port and PMP data. May be configured for tri-state during Enhanced PWM shutdown events.
RE1/AD9/ PMWR/P2C	RE1	0	O	DIG	LATE<1> data output.
		1	I	ST	PORTE<1> data input.
	AD9 ⁽³⁾	x	O	DIG	External memory interface, address/data bit 9 output. ⁽²⁾
		x	I	TTL	External memory interface, data bit 9 input. ⁽²⁾
	PMWR ⁽⁵⁾	x	O	DIG	Parallel Master Port write strobe pin.
		x	I	TTL	Parallel Master Port write pin.
	P2C	0	O	DIG	ECCP2 Enhanced PWM output, channel C; takes priority over port and PMP data. May be configured for tri-state during Enhanced PWM shutdown events.
RE2/AD10/ PMBE/P2B	RE2	0	O	DIG	LATE<2> data output.
		1	I	ST	PORTE<2> data input.
	AD10 ⁽³⁾	x	O	DIG	External memory interface, address/data bit 10 output. ⁽²⁾
		x	I	TTL	External memory interface, data bit 10 input. ⁽²⁾
	PMBE ⁽⁵⁾	x	O	DIG	Parallel Master Port byte enable.
	P2B	0	O	DIG	ECCP2 Enhanced PWM output, channel B; takes priority over port and PMP data. May be configured for tri-state during Enhanced PWM shutdown events.
	RE3	0	O	DIG	LATE<3> data output.
RE3/AD11/ PMA13/P3C/ REFO	AD11 ⁽³⁾	1	I	ST	PORTE<3> data input.
		x	O	DIG	External memory interface, address/data bit 11 output. ⁽²⁾
	x	I	TTL		External memory interface, data bit 11 input. ⁽²⁾
	PMA13	x	O	DIG	Parallel Master Port address.
	P3C ⁽¹⁾	0	O	DIG	ECCP3 Enhanced PWM output, channel C; takes priority over port and PMP data. May be configured for tri-state during Enhanced PWM shutdown events.
	REFO	x	O	DIG	Reference output clock.
	RE4	0	O	DIG	LATE<4> data output.
RE4/AD12/ PMA12/P3B	AD12 ⁽³⁾	1	I	ST	PORTE<4> data input.
		x	O	DIG	External memory interface, address/data bit 12 output. ⁽²⁾
	x	I	TTL		External memory interface, data bit 12 input. ⁽²⁾
	PMA12	x	O	DIG	Parallel Master Port address.
	P3B ⁽¹⁾	0	O	DIG	ECCP3 Enhanced PWM output, channel B; takes priority over port and PMP data. May be configured for tri-state during Enhanced PWM shutdown events.

Legend: O = Output, I = Input, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input,
x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: Default assignments for P1B/P1C and P3B/P3C when ECCPMX Configuration bit is set (80-pin devices only).

2: External memory interface I/O takes priority over all other digital and PMP I/O.

3: Available on 80-pin devices only.

4: Alternate assignment for ECCP2/P2A when ECCP2MX Configuration bit is cleared (all devices in Microcontroller mode).

5: Default configuration for PMP (PMPMX Configuration bit = 1).

PIC18F87J11 FAMILY

TABLE 10-12: PORTE FUNCTIONS (CONTINUED)

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RE5/AD13/ PMA11/P1C	RE5	0	O	DIG	LATE<5> data output.
		1	I	ST	PORTE<5> data input.
	AD13 ⁽³⁾	x	O	DIG	External memory interface, address/data bit 13 output. ⁽²⁾
		x	I	TTL	External memory interface, data bit 13 input. ⁽²⁾
	PMA11	x	O	DIG	Parallel Master Port address.
	P1C ⁽¹⁾	0	O	DIG	ECCP1 Enhanced PWM output, channel C; takes priority over port and PMP data. May be configured for tri-state during Enhanced PWM shutdown events.
RE6/AD14/ PMA10/P1B	RE6	0	O	DIG	LATE<6> data output.
		1	I	ST	PORTE<6> data input.
	AD14 ⁽³⁾	x	O	DIG	External memory interface, address/data bit 14 output. ⁽²⁾
		x	I	TTL	External memory interface, data bit 14 input. ⁽²⁾
	PMA10	x	O	DIG	Parallel Master Port address.
	P1B ⁽¹⁾	0	O	DIG	ECCP1 Enhanced PWM output, channel B; takes priority over port and PMP data. May be configured for tri-state during Enhanced PWM shutdown events.
RE7/AD15/ PMA9/ECCP2/ P2A	RE7	0	O	DIG	LATE<7> data output.
		1	I	ST	PORTE<7> data input.
	AD15 ⁽³⁾	x	O	DIG	External memory interface, address/data bit 15 output. ⁽²⁾
		x	I	TTL	External memory interface, data bit 15 input. ⁽²⁾
	PMA9	x	O	DIG	Parallel Master Port address.
	ECCP2 ⁽⁴⁾	0	O	DIG	ECCP2 compare output and ECCP2 PWM output; takes priority over port data.
		1	I	ST	ECCP2 capture input.
	P2A ⁽⁴⁾	0	O	DIG	ECCP2 Enhanced PWM output, channel A; takes priority over port and PMP data. May be configured for tri-state during Enhanced PWM shutdown events.

Legend: O = Output, I = Input, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input,
x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

- Note 1:** Default assignments for P1B/P1C and P3B/P3C when ECCPMX Configuration bit is set (80-pin devices only).
2: External memory interface I/O takes priority over all other digital and PMP I/O.
3: Available on 80-pin devices only.
4: Alternate assignment for ECCP2/P2A when ECCP2MX Configuration bit is cleared (all devices in Microcontroller mode).
5: Default configuration for PMP (PMPMX Configuration bit = 1).

TABLE 10-13: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTE	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	59
LATE	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	58
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	58
PORTG	RDPU	REPU	RJPU ⁽¹⁾	RG4	RG3	RG2	RG1	RG0	59

Legend: Shaded cells are not used by PORTE.

Note 1: Unimplemented on 64-pin devices, read as '0'.

10.7 PORTF, LATF and TRISF Registers

PORTF is a 7-bit wide, bidirectional port. Only pin 7 of PORTF has no analog input; it is the only pin that can tolerate voltages up to 5.5V.

All pins on PORTF are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

PORTF is multiplexed with analog peripheral functions. RF1 through RF6 may also be used as analog input channels for the A/D Converter. All pins may be used as comparator inputs or outputs by setting the appropriate bits in the CMCON register. To use RF<6:3> as digital inputs, it is also necessary to turn off the comparators.

- Note 1:** On device Resets, pins RF6:RF1 are configured as analog inputs and are read as '0'.
- 2:** To configure PORTF as digital I/O, set the corresponding bits in ANCON0 and ANCON1.

When Configuration bit, PMPMX = 0, PORTF is multiplexed with the Parallel Master Port data. This multiplexing is available only in 80-pin devices.

EXAMPLE 10-6: INITIALIZING PORTF

```
CLRF  PORTF          ; Initialize PORTF by
                      ; clearing output
                      ; data latches
CLRF  LATF           ; Alternate method to
                      ; clear output latches
BSF   WDTCON,ADSHR  ; Enable write/read to
                      ; the shared SFR
MOVLW C0h            ; make RF1:RF2 digital
MOVWF ANCON0          ;
MOVLW 0Fh             ; make RF<6:3> digital
MOVWF ANCON1          ;
BCF   WDTCON,ADSHR  ; Disable write/read to
                      ; the shared SFR
MOVLW CEh            ;
MOVWF TRISF           ; Set RF5:RF4 as outputs,
                      ; RF<7:6>,<3:1> as inputs
```

PIC18F87J11 FAMILY

TABLE 10-14: PORTF FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RF1/AN6/ C2OUT	RF1	0	O	DIG	LATF<1> data output; not affected by analog input.
		1	I	ST	PORTF<1> data input; disabled when analog input enabled.
	AN6	1	I	ANA	A/D input channel 6. Default configuration on POR.
	C2OUT	x	O	DIG	Comparator 2 output.
RF2/PMA5/ AN7/C1OUT	RF2	0	O	DIG	LATF<2> data output; not affected by analog input.
		1	I	ST	PORTF<2> data input; disabled when analog input enabled.
	PMA5	x	O	DIG	Parallel Master Port address.
	AN7	1	I	ANA	A/D input channel 7. Default configuration on POR.
	C1OUT	x	O	DIG	Comparator 1 output.
RF3/AN8/ C2INB	RF3	0	O	DIG	LATF<3> data output; not affected by analog input.
		1	I	ST	PORTF<3> data input; disabled when analog input enabled.
	AN8	1	I	ANA	A/D input channel 8. Default configuration on POR.
	C2INB	x	I	ANA	Comparator 2 input B.
RF4/AN9/ C2INA	RF4	0	O	DIG	LATF<4> data output; not affected by analog input.
		1	I	ST	PORTF<4> data input; disabled when analog input enabled.
	AN9	1	I	ANA	A/D input channel 9. Default configuration on POR.
	C2INA	x	I	ANA	Comparator 2 input A.
RF5/PMD2/ AN10/C1INB/ CVREF	RF5	0	O	DIG	LATF<5> data output; not affected by analog input. Disabled when CVREF output enabled.
		1	I	ST	PORTF<5> data input; disabled when analog input enabled. Disabled when CVREF output enabled.
	PMD2 ⁽¹⁾	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
	AN10	1	I	ANA	A/D input channel 10 and Comparator C1+ input. Default input configuration on POR.
	C1INB	x	I	ANA	Comparator 1 input B.
	CVREF	x	O	ANA	Comparator voltage reference output. Enabling this feature disables digital I/O.
RF6/PMD1/ AN11/C1INA	RF6	0	O	DIG	LATF<6> data output; not affected by analog input.
		1	I	ST	PORTF<6> data input; disabled when analog input enabled.
	PMD1 ⁽¹⁾	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
	AN11	1	I	ANA	A/D input channel 11 and comparator C1- input. Default input configuration on POR; does not affect digital output.
	C1INA	x	I	ANA	Comparator 1 input A.
RF7/PMD0/ SS1	RF7	0	O	DIG	LATF<7> data output.
		1	I	ST	PORTF<7> data input.
	PMD0 ⁽¹⁾	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
	SS1	1	I	TTL	Slave select input for MSSP1 module.

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input,
TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: Alternate PMP configuration when the PMPMX Configuration bit = 0; available on 80-pin devices only.

PIC18F87J11 FAMILY

TABLE 10-15: SUMMARY OF REGISTERS ASSOCIATED WITH PORTF

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	—	59
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	—	58
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	58
ANCON0 ⁽¹⁾	PCFG7	PCFG6	—	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	57
ANCON1 ⁽¹⁾	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	57

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PORTF.

Note 1: Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.

10.8 PORTG, TRISG and LATG Registers

PORTG is a 5-bit wide, bidirectional port. All pins on PORTG are digital only and tolerate voltages up to 5.5V.

PORTG is multiplexed with EUSART2 functions (Table 10-16). PORTG pins have Schmitt Trigger input buffers. PORTG is also multiplexed with address and control functions of the Parallel Master Port.

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTG 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. The user should refer to the corresponding peripheral section for the correct TRIS bit settings. The pin override value is not loaded into the TRIS register. This allows read-modify-write of the TRIS register without concern due to peripheral overrides.

Although the port itself is only five bits wide, PORTG<7:5> bits are still implemented. These are used to control the weak pull-ups on the I/O ports associated with the external memory bus (PORTD, PORTE and PORTJ). Setting these bits enables the pull-ups. Since these are control bits and are not associated with port I/O, the corresponding TRISG and LATG bits are not implemented.

EXAMPLE 10-7: INITIALIZING PORTG

```
CLRF    PORTG    ; Initialize PORTG by
                  ; clearing output
                  ; data latches
CLRF    LATG     ; Alternate method to clear
                  ; output data latches
MOVLW   04h      ; Value used to initialize
                  ; data direction
MOVWF   TRISG    ; Set RG1:RG0 as outputs
                  ; RG2 as input
                  ; RG4:RG3 as outputs
```

TABLE 10-16: PORTG FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RG0/PMA8/ ECCP3/P3A	RG0	0	O	DIG	LATG<0> data output.
		1	I	ST	PORTG<0> data input.
	PMA8	x	O	DIG	Parallel Master Port address.
			O	DIG	ECCP3 compare and PWM output; takes priority over port data.
	ECCP3		I	ST	ECCP3 capture input.
			O	DIG	ECCP3 Enhanced PWM output, channel A; takes priority over port and PMP data. May be configured for tri-state during Enhanced PWM shutdown events.
RG1/PMA7/ TX2/CK2	RG1	0	O	DIG	LATG<1> data output.
		1	I	ST	PORTG<1> data input.
	PMA7	x	O	DIG	Parallel Master Port address.
		1	O	DIG	Synchronous serial data output (EUSART2 module); takes priority over port data.
	CK2	1	O	DIG	Synchronous serial data input (EUSART2 module). User must configure as an input.
		1	I	ST	Synchronous serial clock input (EUSART2 module).
RG2/PMA6/ RX2/DT2	RG2	0	O	DIG	LATG<2> data output.
		1	I	ST	PORTG<2> data input.
	PMA6	x	O	DIG	Parallel Master Port address.
		1	I	ST	Asynchronous serial receive data input (EUSART2 module).
	RX2	1	O	DIG	Synchronous serial data output (EUSART2 module); takes priority over port data.
		1	I	ST	Synchronous serial data input (EUSART2 module). User must configure as an input.
RG3/PMCS1/ CCP4/P3D	RG3	0	O	DIG	LATG<3> data output.
		1	I	ST	PORTG<3> data input.
	PMCS1	x	O	DIG	Parallel Master Port address chip select 1
		x	I	TTL	Parallel Master Port address chip select 1 in.
	CCP4	0	O	DIG	CCP4 compare output and CCP4 PWM output; takes priority over port data.
		1	I	ST	CCP4 capture input.
	P3D	0	O	DIG	ECCP3 Enhanced PWM output, channel D; takes priority over port and PMP data. May be configured for tri-state during Enhanced PWM shutdown events.
RG4/PMCS2/ CCP5/P1D	RG4	0	O	DIG	LATG<4> data output.
		1	I	ST	PORTG<4> data input.
	PMCS2	x	O	DIG	Parallel Master Port address chip select 2
		0	O	DIG	CCP5 compare output and CCP5 PWM output; takes priority over port data.
	CCP5	1	I	ST	CCP5 capture input.
		0	O	DIG	ECCP1 Enhanced PWM output, channel D; takes priority over port and PMP data. May be configured for tri-state during Enhanced PWM shutdown events.
	P1D				

Legend: O = Output, I = Input, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input,
x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

PIC18F87J11 FAMILY

TABLE 10-17: SUMMARY OF REGISTERS ASSOCIATED WITH PORTG

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTG	RDPU	REPU	RJPU ⁽¹⁾	RG4	RG3	RG2	RG1	RG0	59
LATG	—	—	—	LATG4	LATG3	LATG2	LATG1	LATG0	58
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	58

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used by PORTG.

Note 1: Unimplemented on 64-pin devices, read as ‘0’.

10.9 PORTH, LATH and TRISH Registers

Note: PORTH is available only on 80-pin devices.

PORTH is an 8-bit wide, bidirectional I/O port. PORTH pins <3:0> are digital only and tolerate voltages up to 5.5V.

All pins on PORTH are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

When the external memory interface is enabled, four of the PORTH pins function as the high-order address lines for the interface. The address output from the interface takes priority over other digital I/O. The corresponding TRISH bits are also overridden. PORTH pins, RH4 through RH7, are multiplexed with analog converter inputs. The operation of these pins as analog inputs is selected by clearing or setting the corresponding bits in the ANCON1 register. RH2 to RH6 are multiplexed with the Parallel Master Port and RH4 to RH6 are multiplexed as comparator inputs.

PORTH can also be configured as the alternate Enhanced PWM output channels B and C for the ECCP1 and ECCP3 modules. This is done by clearing the ECCPMX Configuration bit.

EXAMPLE 10-8: INITIALIZING PORTH

```
CLRF  PORTH      ; Initialize PORTH by
; clearing output
; data latches
CLRF  LATH       ; Alternate method to
; clear output latches
BSF   WDTCON,ADSHR ; Enable write/read to
; the shared SFR
MOVLW F0h        ; Configure PORTH as
MOVWF ANCON1     ; digital I/O
BCF   WDTCON,ADSHR ; Disable write/read to
; the shared SFR
MOVLW 0CFh       ; Value used to initialize
; data direction
MOVWF TRISH      ; Set RH<3:0> as inputs
; RH<5:4> as outputs
; RH<7:6> as inputs
```

PIC18F87J11 FAMILY

TABLE 10-18: PORTH FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RH0/A16	RH0	0	O	DIG	LATH<0> data output.
		1	I	ST	PORTH<0> data input.
	A16	x	O	DIG	External memory interface, address line 16. Takes priority over port data.
RH1/A17	RH1	0	O	DIG	LATH<1> data output.
		1	I	ST	PORTH<1> data input.
	A17	x	O	DIG	External memory interface, address line 17. Takes priority over port data.
RH2/A18/ PMD7	RH2	0	O	DIG	LATH<2> data output.
		1	I	ST	PORTH<2> data input.
	A18	x	O	DIG	External memory interface, address line 18. Takes priority over port data.
	PMD7 ⁽²⁾	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
RH3/A19/ PMD6	RH3	0	O	DIG	LATH<3> data output.
		1	I	ST	PORTH<3> data input.
	A19	x	O	DIG	External memory interface, address line 19. Takes priority over port data.
	PMD6 ⁽²⁾	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
RH4/PMD3/ AN12/P3C/ C2INC	RH4	0	O	DIG	LATH<4> data output.
		1	I	ST	PORTH<4> data input.
	PMD3 ⁽²⁾	x	I	TTL	Parallel Master Port data out.
		x	O	DIG	Parallel Master Port data input.
	AN12		I	ANA	A/D input channel 12. Default input configuration on POR; does not affect digital output.
	P3C ⁽¹⁾	0	O	DIG	ECCP3 Enhanced PWM output, channel C; takes priority over port and PMP data. May be configured for tri-state during Enhanced PWM shutdown events.
	C2INC	x	I	ANA	Comparator 2 input C.
RH5/PMBE/ AN13/P3B/ C2IND	RH5	0	O	DIG	LATH<5> data output.
		1	I	ST	PORTH<5> data input.
	PMBE ⁽²⁾	x	O	DIG	Parallel Master Port data byte enable.
	AN13		I	ANA	A/D input channel 13. Default input configuration on POR; does not affect digital output.
	P3B ⁽¹⁾	0	O	DIG	ECCP3 Enhanced PWM output, channel B; takes priority over port and PMP data. May be configured for tri-state during Enhanced PWM shutdown events.
	C2IND	x	I	ANA	Comparator 2 input D.
RH6/PMRD/ AN14/P1C/ C1INC	RH6	0	O	DIG	LATH<6> data output.
		1	I	ST	PORTH<6> data input.
	PMRD ⁽²⁾	x	O	DIG	Parallel Master Port read strobe.
		x	I	TTL	Parallel Master Port read in.
	AN14		I	ANA	A/D input channel 14. Default input configuration on POR; does not affect digital output.
	P1C ⁽¹⁾	0	O	DIG	ECCP1 Enhanced PWM output, channel C; takes priority over port and PMP data. May be configured for tri-state during Enhanced PWM shutdown events.
	C1INC	x	I	ANA	Comparator 1 input C.

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input,
TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: Alternate assignments for P1B/P1C and P3B/P3C when the ECCPMX Configuration bit is cleared. Default assignments are PORTE<6:3>.

2: Alternate PMP configuration when the PMPMX Configuration bit = 0; available on 80-pin devices only.

TABLE 10-18: PORTH FUNCTIONS (CONTINUED)

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RH7/PMWR/ AN15/P1B	RH7	0	O	DIG	LATH<7> data output.
		1	I	ST	PORTH<7> data input.
	PMWR ⁽²⁾	x	O	DIG	Parallel Master Port write strobe.
		x	I	TTL	Parallel Master Port write in.
	AN15		I	ANA	A/D input channel 15. Default input configuration on POR; does not affect digital output.
	P1B ⁽¹⁾	0	O	DIG	ECCP1 Enhanced PWM output, channel B; takes priority over port and PMP data. May be configured for tri-state during Enhanced PWM shutdown events.

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: Alternate assignments for P1B/P1C and P3B/P3C when the ECCPMX Configuration bit is cleared. Default assignments are PORTE<6:3>.

2: Alternate PMP configuration when the PMPMX Configuration bit = 0; available on 80-pin devices only.

TABLE 10-19: SUMMARY OF REGISTERS ASSOCIATED WITH PORTH

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTH ⁽¹⁾	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0	58
LATH ⁽¹⁾	LATH7	LATH6	LATH5	LATH4	LATH3	LATH2	LATH1	LATH0	59
TRISH ⁽¹⁾	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	58
ANCON1 ⁽²⁾	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	57

Legend: Shaded cells are not used by PORTH.

Note 1: Unimplemented on 64-pin devices, read as '0'.

2: Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.

PIC18F87J11 FAMILY

10.10 PORTJ, TRISJ and LATJ Registers

Note: PORTJ is available only on 80-pin devices.

PORTJ is an 8-bit wide, bidirectional port. All pins on PORTJ are digital only and tolerate voltages up to 5.5V.

All pins on PORTJ are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

Note: These pins are configured as digital inputs on any device Reset.

When the external memory interface is enabled, all of the PORTJ pins function as control outputs for the interface. This occurs automatically when the interface is enabled by clearing the EBDIS control bit (MEMCON<7>). The TRISJ bits are also overridden.

Each of the PORTJ pins has a weak internal pull-up. A single control bit can turn off all the pull-ups. This is performed by clearing bit RJPU (PORTG<5>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on any device Reset.

EXAMPLE 10-9: INITIALIZING PORTJ

```
CLRF    PORTJ    ; Initialize PORTG by
                  ; clearing output
                  ; data latches
CLRF    LATJ     ; Alternate method to clear
                  ; output data latches
MOVLW  0CFh    ; Value used to initialize
                  ; data direction
MOVWF  TRISJ    ; Set RJ3:RJ0 as inputs
                  ; RJ5:RJ4 as output
                  ; RJ7:RJ6 as inputs
```

TABLE 10-20: PORTJ FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RJ0/ALE	RJ0	0	O	DIG	LATJ<0> data output.
		1	I	ST	PORTJ<0> data input.
	ALE	x	O	DIG	External memory interface address latch enable control output; takes priority over digital I/O.
RJ1/ <u>OE</u>	RJ1	0	O	DIG	LATJ<1> data output.
		1	I	ST	PORTJ<1> data input.
	<u>OE</u>	x	O	DIG	External memory interface output enable control output; takes priority over digital I/O.
RJ2/ <u>WRL</u>	RJ2	0	O	DIG	LATJ<2> data output.
		1	I	ST	PORTJ<2> data input.
	<u>WRL</u>	x	O	DIG	External memory bus write low byte control; takes priority over digital I/O.
RJ3/ <u>WRH</u>	RJ3	0	O	DIG	LATJ<3> data output.
		1	I	ST	PORTJ<3> data input.
	<u>WRH</u>	x	O	DIG	External memory interface write high byte control output; takes priority over digital I/O.
RJ4/BA0	RJ4	0	O	DIG	LATJ<4> data output.
		1	I	ST	PORTJ<4> data input.
	BA0	x	O	DIG	External memory interface byte address 0 control output; takes priority over digital I/O.
RJ5/ <u>CE</u>	RJ5	0	O	DIG	LATJ<5> data output.
		1	I	ST	PORTJ<5> data input.
	<u>CE</u>	x	O	DIG	External memory interface chip enable control output; takes priority over digital I/O.
RJ6/ <u>LB</u>	RJ6	0	O	DIG	LATJ<6> data output.
		1	I	ST	PORTJ<6> data input.
	<u>LB</u>	x	O	DIG	External memory interface lower byte enable control output; takes priority over digital I/O.
RJ7/ <u>UB</u>	RJ7	0	O	DIG	LATJ<7> data output.
		1	I	ST	PORTJ<7> data input.
	<u>UB</u>	x	O	DIG	External memory interface upper byte enable control output; takes priority over digital I/O.

Legend: O = Output, I = Input, DIG = Digital Output, ST = Schmitt Buffer Input,
x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

TABLE 10-21: SUMMARY OF REGISTERS ASSOCIATED WITH PORTJ

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTJ ⁽¹⁾	RJ7	RJ6	RJ5	RJ4	RJ3	RJ2	RJ1	RJ0	59
LATJ ⁽¹⁾	LATJ7	LATJ6	LATJ5	LATJ4	LATJ3	LATJ2	LATJ1	LATJ0	58
TRISJ ⁽¹⁾	TRISJ7	TRISJ6	TRISJ5	TRISJ4	TRISJ3	TRISJ2	TRISJ1	TRISJ0	58
PORTG	RDPU	REPU	RJPU ⁽¹⁾	RG4	RG3	RG2	RG1	RG0	59

Legend: Shaded cells are not used by PORTJ.

Note 1: Unimplemented on 64-pin devices, read as '0'.

PIC18F87J11 FAMILY

NOTES:

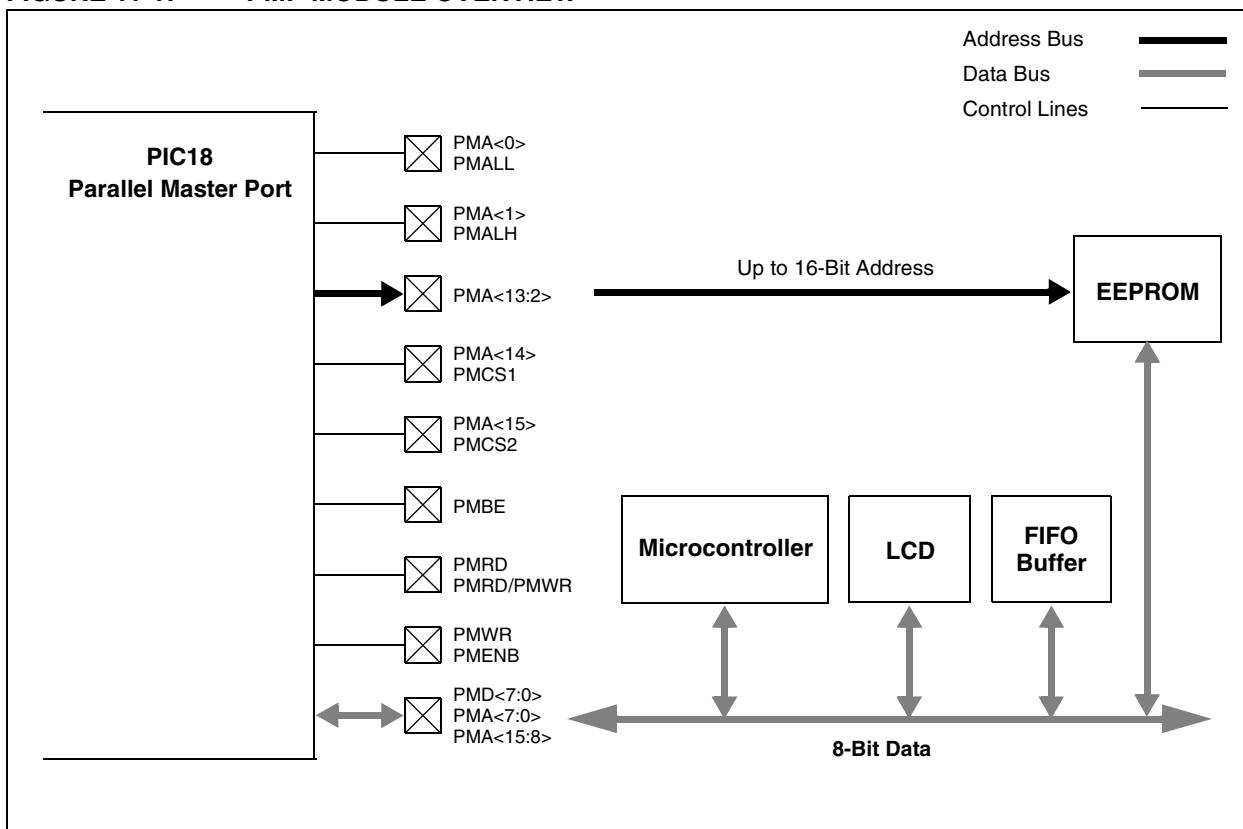
11.0 PARALLEL MASTER PORT

The Parallel Master Port module (PMP) is a parallel, 8-bit I/O module, specifically designed to communicate with a wide variety of parallel devices, such as communication peripherals, LCDs, external memory devices and microcontrollers. Because the interface to parallel peripherals varies significantly, the PMP is highly configurable. The PMP module can be configured to serve as either a Parallel Master Port or as a Parallel Slave Port.

Key features of the PMP module include:

- Up to 16 Programmable Address Lines
- Up to Two Chip Select Lines
- Programmable Strobe Options
 - Individual Read and Write Strobes or;
 - Read/Write Strobe with Enable Strobe
- Address Auto-Increment/Auto-Decrement
- Programmable Address/Data Multiplexing
- Programmable Polarity on Control Signals
- Legacy Parallel Slave Port Support
- Enhanced Parallel Slave Support
 - Address Support
 - 4-Byte Deep, Auto-Incrementing Buffer
- Programmable Wait States
- Selectable Input Voltage Levels

FIGURE 11-1: PMP MODULE OVERVIEW



PIC18F87J11 FAMILY

11.1 Module Registers

The PMP module has a total of 14 Special Function Registers for its operation, plus one additional register to set configuration options. Of these, 8 registers are used for control and 6 are used for PMP data transfer.

11.1.1 CONTROL REGISTERS

The eight PMP Control registers are:

- PMCONH and PMCONL
- PMMODEH and PMMODEL
- PMSTATL and PMSTATH
- PMEH and PMEL

The PMCON registers (Register 11-1 and Register 11-2) control basic module operations, including turning the module on or off. They also configure address multiplexing and control strobe configuration.

The PMMODE registers (Register 11-3 and Register 11-4) configure the various Master and Slave Operating modes, the data width and interrupt generation.

The PMEH and PMEL registers (Register 11-5 and Register 11-6) configure the module's operation at the hardware (I/O pin) level.

The PMSTAT registers (Register 11-7 and Register 11-8) provide status flags for the module's input and output buffers, depending on the operating mode.

REGISTER 11-1: PMCONH: PARALLEL PORT CONTROL HIGH BYTE REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMSEN	—	PSIDL	ADRMUX1	ADRMUX0	PTBEEN	PTWREN	PTRDEN
bit 7							

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

bit 7	PMSEN: Parallel Master Port Enable bit 1 = PMP enabled 0 = PMP disabled, no off-chip access performed
bit 6	Unimplemented: Read as '0'
bit 5	PSIDL: Stop in Idle Mode bit 1 = Discontinue module operation when device enters Idle mode 0 = Continue module operation in Idle mode
bit 4-3	ADRMUX1:ADRMUX0: Address/Data Multiplexing Selection bits 11 = Reserved 10 = All 16 bits of address are multiplexed on PMD<7:0> pins 01 = Lower 8 bits of address are multiplexed on PMD<7:0> pins, upper 8 bits are on PMA<15:8> 00 = Address and data appear on separate pins
bit 2	PTBEEN: Byte Enable Port Enable bit (16-bit Master mode) 1 = PMBE port enabled 0 = PMBE port disabled
bit 1	PTWREN: Write Enable Strobe Port Enable bit 1 = PMWR/PMENB port enabled 0 = PMWR/PMENB port disabled
bit 0	PTRDEN: Read/Write Strobe Port Enable bit 1 = PMRD/PMWR port enabled 0 = PMRD/PMWR port disabled

REGISTER 11-2: PMCONL: PARALLEL PORT CONTROL LOW BYTE REGISTER

R/W-0	R/W-0	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0	R/W-0	R/W-0
CSF1	CSF0	ALP	CS2P	CS1P	BEP	WRSP	RDSP
bit 7	bit 0						

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

bit 7-6	CSF1:CSF0: Chip Select Function bits
	11 = Reserved
	10 = PMCS1 and PMCS2 function as chip select
	01 = PMCS2 functions as chip select, PMCS1 used as address bit 14 (PMADDRH address bit 6)
	00 = PMCS2 and PMCS1 used as address bits 15 and 14 (PMADDRH address bits 7 and 6)
bit 5	ALP: Address Latch Polarity bit ⁽¹⁾
	1 = Active-high (<u>PMALL</u> and <u>PMALH</u>)
	0 = Active-low (<u>PMALL</u> and <u>PMALH</u>)
bit 4	CS2P: Chip Select 2 Polarity bit ⁽¹⁾
	1 = Active-high (<u>PMCS2</u>)
	0 = Active-low (<u>PMCS2</u>)
bit 3	CS1P: Chip Select 1 Polarity bit ⁽¹⁾
	1 = Active-high (<u>PMCS1/PMCS</u>)
	0 = Active-low (<u>PMCS1/PMCS</u>)
bit 2	BEP: Byte Enable Polarity bit
	1 = Byte enable active-high (<u>PMBE</u>)
	0 = Byte enable active-low (<u>PMBE</u>)
bit 1	WRSP: Write Strobe Polarity bit
	<u>For Slave modes and Master mode 2 (PMMODEH<1:0> = 00, 01, 10):</u>
	1 = Write strobe active-high (<u>PMWR</u>)
	0 = Write strobe active-low (<u>PMWR</u>)
	<u>For Master mode 1 (PMMODEH<1:0> = 11):</u>
	1 = Enable strobe active-high (<u>PMENB</u>)
	0 = Enable strobe active-low (<u>PMENB</u>)
bit 0	RDSP: Read Strobe Polarity bit
	<u>For Slave modes and Master mode 2 (PMMODEH<1:0> = 00, 01, 10):</u>
	1 = Read strobe active-high (<u>PMRD</u>)
	0 = Read strobe active-low (<u>PMRD</u>)
	<u>For Master mode 1 (PMMODEH<1:0> = 11):</u>
	1 = Read/write strobe active-high (<u>PMRD/PMWR</u>)
	0 = Read/write strobe active-low (<u>PMRD/PMWR</u>)

Note 1: These bits have no effect when their corresponding pins are used as address lines.

PIC18F87J11 FAMILY

REGISTER 11-3: PMMODEH: PARALLEL PORT MODE HIGH BYTE REGISTER

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0
bit 7	bit 0						

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

- bit 7 **BUSY:** Busy bit (Master mode only)
1 = Port is busy
0 = Port is not busy
- bit 6-5 **IRQM1:IRQM0:** Interrupt Request Mode bits
11 = Interrupt generated when Read Buffer 3 is read or Write Buffer 3 is written (Buffered PSP mode) or on a read or write operation when PMA<1:0> = 11 (Addressable PSP mode only)
10 = No interrupt generated, processor stall activated
01 = Interrupt generated at the end of the read/write cycle
00 = No interrupt generated
- bit 4-3 **INCM1:INCM0:** Increment Mode bits
11 = PSP read and write buffers auto-increment (Legacy PSP mode only)
10 = Decrement ADDR<15,13:0> by 1 every read/write cycle
01 = Increment ADDR<15,13:0> by 1 every read/write cycle
00 = No increment or decrement of address
- bit 2 **MODE16:** 8/16-Bit Mode bit
1 = 16-Bit mode: data register is 16 bits, a read or write to the data register invokes two 8-bit transfers
0 = 8-Bit mode: data register is 8 bits, a read or write to the data register invokes one 8-bit transfer
- bit 1-0 **MODE1:MODE0:** Parallel Port Mode Select bits
11 = Master mode 1 (PMCSx, PMRD/PMWR, PMENB, PMBE, PMA<x:0> and PMD<7:0>)
10 = Master mode 2 (PMCSx, PMRD, PMWR, PMBE, PMA<x:0> and PMD<7:0>)
01 = Enhanced PSP, control signals (PMRD, PMWR, PMCS, PMD<7:0> and PMA<1:0>)
00 = Legacy Parallel Slave Port, control signals (PMRD, PMWR, PMCS and PMD<7:0>)

REGISTER 11-4: PMMODEL: PARALLEL PORT MODE LOW BYTE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WAITB1 ⁽¹⁾	WAITB0 ⁽¹⁾	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1 ⁽¹⁾	WAITE0 ⁽¹⁾
bit 7							bit 0

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

bit 7-6 **WAITB1:WAITB0:** Data Setup to Read/Write Wait State Configuration bits⁽¹⁾

11 = Data wait of 4 Tcy; multiplexed address phase of 4 Tcy

10 = Data wait of 3 Tcy; multiplexed address phase of 3 Tcy

01 = Data wait of 2 Tcy; multiplexed address phase of 2 Tcy

00 = Data wait of 1 Tcy; multiplexed address phase of 1 Tcy

bit 5-2 **WAITM3:WAITM0:** Read to Byte Enable Strobe Wait State Configuration bits

1111 = Wait of additional 15 Tcy

...

0001 = Wait of additional 1 Tcy

0000 = No additional wait cycles (operation forced into one Tcy)

bit 1-0 **WAITE1:WAITE0:** Data Hold After Strobe Wait State Configuration bits⁽¹⁾

11 = Wait of 4 Tcy

10 = Wait of 3 Tcy

01 = Wait of 2 Tcy

00 = Wait of 1 Tcy

Note 1: WAITB and WAITE bits are ignored whenever WAITM3:WAITM0 = 0000.

REGISTER 11-5: PMEH: PARALLEL PORT ENABLE HIGH BYTE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8
bit 7							bit 0

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

bit 7-6 **PTEN15:PTEN14:** PMCSx Strobe Enable bits

1 = PMA15 and PMA14 function as either PMA<15:14> or PMCS2 and PMCS1

0 = PMA15 and PMA14 function as port I/O

bit 5-0 **PTEN13:PTEN8:** PMP Address Port Enable bits

1 = PMA<13:8> function as PMP address lines

0 = PMA<13:8> function as port I/O

PIC18F87J11 FAMILY

REGISTER 11-6: PMEL: PARALLEL PORT ENABLE LOW BYTE REGISTER

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| PTEN7 | PTEN6 | PTEN5 | PTEN4 | PTEN3 | PTEN2 | PTEN1 | PTEN0 |
| bit 7 | bit 0 | | | | | | |

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

bit 7-2 **PTEN7:PTEN2:** PMP Address Port Enable bits

1 = PMA<7:2> function as PMP address lines

0 = PMA<7:2> function as port I/O

bit 1-0 **PTEN1:PTEN0:** PMALH/PMALL Strobe Enable bits

1 = PMA1 and PMA0 function as either PMA<1:0> or PMALH and PMALL

0 = PMA1 and PMA0 pads functions as port I/O

REGISTER 11-7: PMSTATH: PARALLEL PORT STATUS HIGH BYTE REGISTER

R-0	R/W-0	U-0	U-0	R-0	R-0	R-0	R-0
IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F
bit 7	bit 0						

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

bit 7 **IBF:** Input Buffer Full Status bit

1 = All writable input buffer registers are full

0 = Some or all of the writable input buffer registers are empty

bit 6 **IBOV:** Input Buffer Overflow Status bit

1 = A write attempt to a full input byte register occurred (must be cleared in software)

0 = No overflow occurred

bit 5-4 **Unimplemented:** Read as '0'bit 3-0 **IB3F:IB0F:** Input Buffer Status Full bits

1 = Input buffer contains data that has not been read (reading buffer will clear this bit)

0 = Input buffer does not contain any unread data

REGISTER 11-8: PMSTATL: PARALLEL PORT STATUS LOW BYTE REGISTER

R-1	R/W-0	U-0	U-0	R-1	R-1	R-1	R-1
OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E
bit 7	bit 0						

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

- bit 7 **OBE:** Output Buffer Empty Status bit
1 = All readable output buffer registers are empty
0 = Some or all of the readable output buffer registers are full
- bit 6 **OBUF:** Output Buffer Underflow Status bit
1 = A read occurred from an empty output byte register (must be cleared in software)
0 = No underflow occurred
- bit 5-4 **Unimplemented:** Read as '0'
- bit 3-0 **OBnE:** Output Buffer n Status Empty bit
1 = Output buffer is empty (writing data to the buffer will clear this bit)
0 = Output buffer contains data that has not been transmitted

PIC18F87J11 FAMILY

11.1.2 DATA REGISTERS

The PMP module uses 6 registers for transferring data into and out of the microcontroller. They are arranged as three pairs to allow the option of 16-bit data operations:

- PMDIN1H and PMDIN1L
- PMDIN2H and PMDIN2L
- PMADDRH/PMDOUT1H and PMADDRL/PMDOUT1L
- PMDOUT2H and PMDOUT2L

The PMDIN1 register is used for incoming data in Slave modes, and both input and output data in Master modes. The PMDIN2 register is used for buffering input data in select Slave modes.

The PMADDRx/PMDOUT1x registers are actually a single register pair; the name and function is dictated by the module's operating mode. In Master modes, the registers functions as the PMADDRH and PMADDRL registers, and contain the address of any incoming or outgoing data. In Slave modes, the registers function as PMDOUT1H and PMDOUT1L and are used for outgoing data.

PMADDRH differs from PMADDRL in that it can also have limited PMP control functions. When the module is operating in select Master mode configurations, the

upper two bits of the register can be used to determine the operation of chip select signals. If chip select signals are not used, PMADDR simply functions to hold the upper 8 bits of the address. The function of the individual bits in PMADDRH is shown in Register 11-9.

The PMDOUT2H and PMDOUT2L registers are only used in buffered Slave modes and serve as a buffer for outgoing data.

11.1.3 PAD CONFIGURATION CONTROL REGISTER

In addition to the module level configuration options, the PMP module can also be configured at the I/O pin for electrical operation. This option allows users to select either the normal Schmitt Trigger input buffer on digital I/O pins shared with the PMP, or use TTL level compatible buffers instead. Buffer configuration is controlled by the PMPTTL bit in the PADCFG1 register.

The PADCFG1 register is one of the shared address SFRs, and has the same address as the TMR2 register. PADCFG1 is accessed by setting the ADSHR bit (WDTCON<4>). Refer to **Section 5.3.4.1 "Shared Address SFRs"** for more information.

REGISTER 11-9: PMADDRH: PARALLEL PORT ADDRESS REGISTER, HIGH BYTE (MASTER MODES ONLY)⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CS2	CS1			ADDR<13:8>			
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit
-n = Value at Reset 1 = bit is set

U = Unimplemented bit, read as '0'
0 = bit is cleared x = bit is unknown

bit 7 **CS2:** Chip Select 2 bit

If PMCON<7:6> = 10 or 01:

1 = Chip Select 2 is active
0 = Chip Select 2 is inactive

If PMCON<7:6> = 11 or 00:

Bit functions as ADDR<15>.

bit 6 **CS1:** Chip Select 1 bit

If PMCON<7:6> = 10:

1 = Chip Select 1 is active
0 = Chip Select 1 is inactive

If PMCON<7:6> = 11 or 0x:

Bit functions as ADDR<14>.

bit 5-0 **ADDR13:ADDR0:** Destination Address bits

Note 1: In Enhanced Slave mode, PMADDRH functions as PMDOUT1H, one of the Output Data Buffer registers.

11.1.4 PMP MULTIPLEXING OPTIONS (80-PIN DEVICES)

By default, the PMP and the external memory bus multiplex some of their signals to the same I/O pins on PORTD and PORTE. It is possible that some applications may require the PMP signals to be located elsewhere. For these instances, the 80-pin devices can be configured to multiplex the PMP to different I/O ports. PMP configuration is determined by the PMPMX Configuration bit setting; by default, the PMP and EMB modules share PORTD and PORTE. The optional pin configuration is shown in Table 11-1.

TABLE 11-1: PMP PIN MULTIPLEXING FOR 80-PIN DEVICES

PMP Function	Pin Assignment	
	PMPMX = 1	PMPMX = 0
PMD0	PORTD<0>	PORTF<7>
PMD1	PORTD<1>	PORTF<6>
PMD2	PORTD<2>	PORTF<5>
PMD3	PORTD<3>	PORTH<4>
PMD4	PORTD<4>	PORTA<5>
PMD5	PORTD<5>	PORTA<4>
PMD6	PORTD<6>	PORTH<3>
PMD7	PORTD<7>	PORTH<2>
PMBE	PORTE<2>	PORTH<5>
PMWR	PORTE<1>	PORTH<7>
PMRD	PORTE<0>	PORTH<6>

11.2 Slave Port Modes

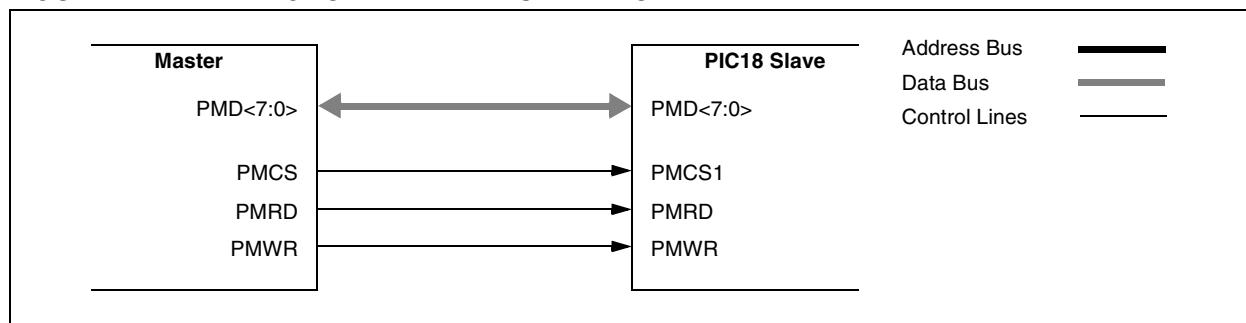
The primary mode of operation for the module is configured using the MODE1:MODE0 bits in the PMMODEH register. The setting affects whether the module acts as a slave or a master and it determines the usage of the control pins.

11.2.1 LEGACY MODE (PSP)

In Legacy mode (PMODEH<1:0> = 00 and PMPEN = 1), the module is configured as a Parallel Slave Port with the associated enabled module pins dedicated to the module. In this mode, an external device, such as another microcontroller or microprocessor, can asynchronously read and write data using the 8-bit data bus (PMD<7:0>), the read (PMRD), write (PMWR) and chip select (PMCS1) inputs. It acts as a slave on the bus and responds to the read/write control signals.

Figure 11-2 shows the connection of the Parallel Slave Port. When chip select is active and a write strobe occurs (PMCS = 1 and PMWR = 1), the data from PMD<7:0> is captured into the PMDIN1L register.

FIGURE 11-2: LEGACY PARALLEL SLAVE PORT EXAMPLE



PIC18F87J11 FAMILY

11.2.1.1 WRITE TO SLAVE PORT

When chip select is active and a write strobe occurs ($\text{PMCS} = 1$ and $\text{PMWR} = 1$), the data from $\text{PMD}_{<7:0>}$ is captured into the PMDINL1 register. The PMPIF and IBF flag bits are set when the write ends. The timing for the control signals in Write mode is shown in Figure 11-3. The polarity of the control signals are configurable.

11.2.1.2 READ FROM SLAVE PORT

When chip select is active and a read strobe occurs ($\text{PMCS} = 1$ and $\text{PMRD} = 1$), the data from the PMDOUTL1 register ($\text{PMDOUTL1}_{<7:0>}$) is presented onto $\text{PMD}_{<7:0>}$. The timing for the control signals in Read mode is shown in Figure 11-4.

FIGURE 11-3: PARALLEL SLAVE PORT WRITE WAVEFORMS

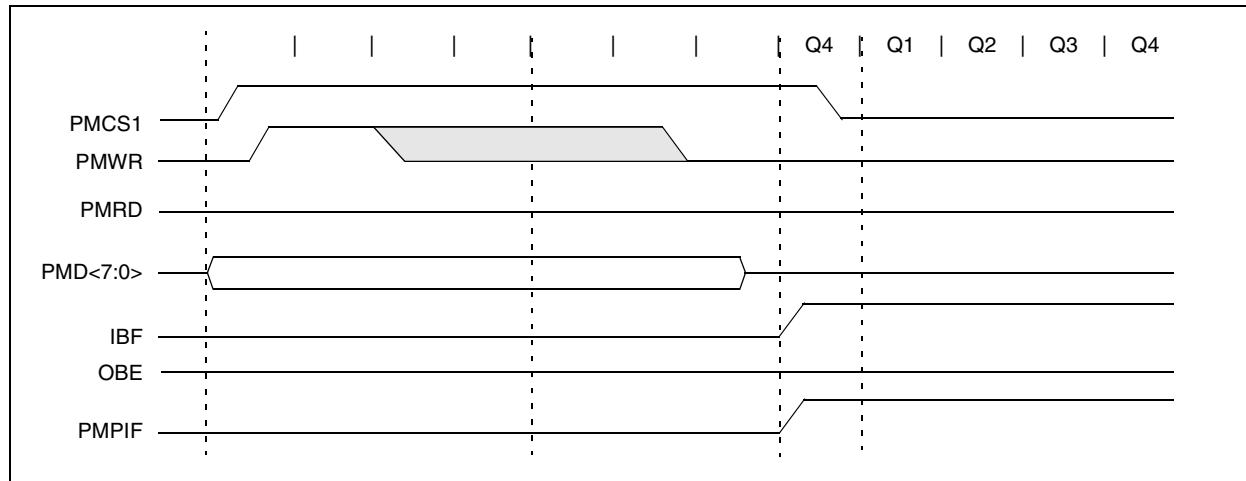
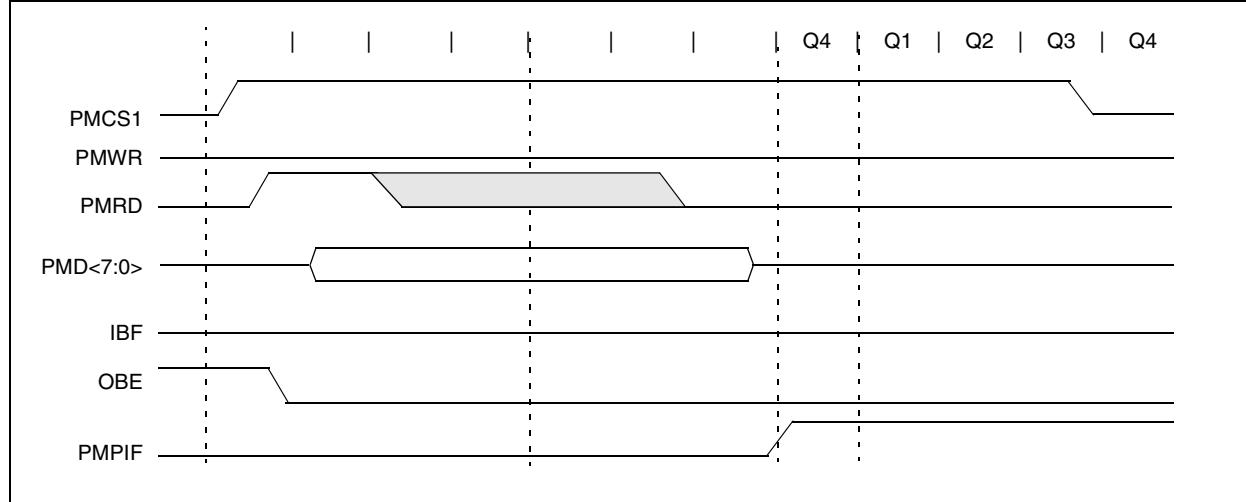


FIGURE 11-4: PARALLEL SLAVE PORT READ WAVEFORMS



11.2.2 BUFFERED PARALLEL SLAVE PORT MODE

Buffered Parallel Slave Port mode is functionally identical to the Legacy Parallel Slave Port mode with one exception: the implementation of 4-level read and write buffers. Buffered PSP mode is enabled by setting the INCM bits in the PMMODE register. If the INCM<1:0> bits are set to '11', the PMP module will act as the Buffered Parallel Slave Port.

When the Buffered mode is active, the PMDIN1L, PMDIN1H, PMDIN2L and PMDIN2H registers become the write buffers and the PMDOUT1L, PMDOUT1H, PMDOUT2L and PMDOUT2H registers become the read buffers. Buffers are numbered 0 through 3, starting with the lower byte of PMDIN1L to PMDIN2H as the read buffers, and PMDOUT1L to PMDOUT2H as the write buffers.

11.2.2.1 READ FROM SLAVE PORT

For read operations, the bytes will be sent out sequentially, starting with Buffer 0 (PMDOUT1L<7:0>) and ending with Buffer 3 (PMDOUT2H<7:0>) for every read strobe. The module maintains an internal pointer to keep track of which buffer is to be read. Each of the buffers has a corresponding read status bit, OBxE, in the PMSTATL register. This bit is cleared when a buffer contains data that has not been written to the bus, and is set when data is written to the bus. If the current buffer location being read from is empty, a buffer under-

flow is generated, and the Buffer Overflow flag bit OBUF is set. If all 4 OBxE status bits are set, then the Output Buffer Empty flag (OBE) will also be set.

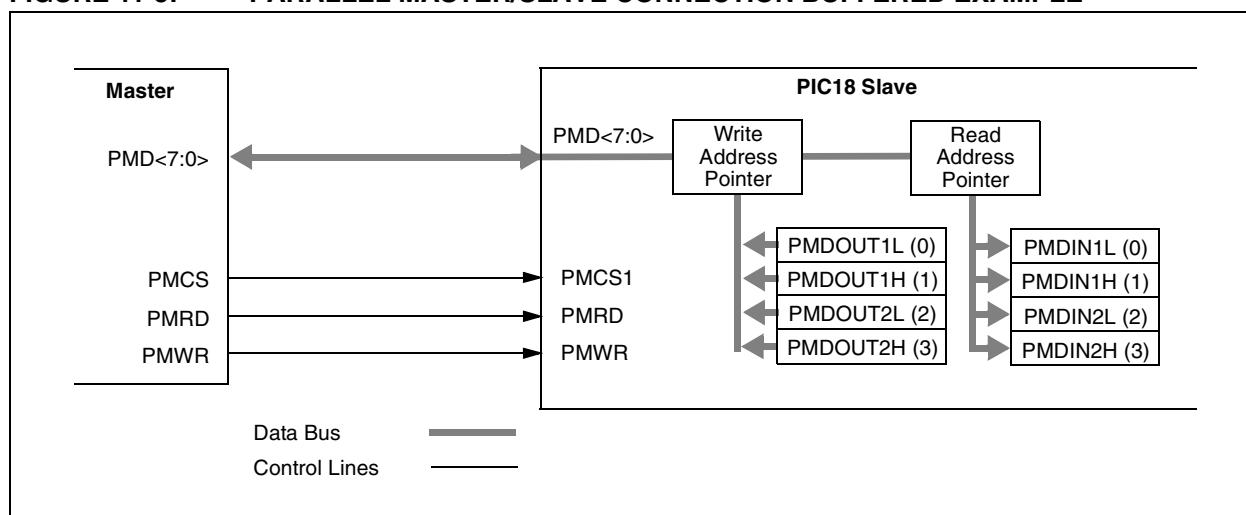
11.2.2.2 WRITE TO SLAVE PORT

For write operations, the data is stored sequentially, starting with Buffer 0 (PMDIN1L<7:0>) and ending with Buffer 3 (PMDIN2H<7:0>). As with read operations, the module maintains an internal pointer to the buffer that is to be written next.

The input buffers have their own write status bits, IBxF in the PMSTATH register. The bit is set when the buffer contains unread incoming data, and cleared when the data has been read. The flag bit is set on the write strobe. If a write occurs on a buffer when its associated IBxF bit is set, the Buffer Overflow flag, IBOV, is set; any incoming data in the buffer will be lost. If all 4 IBxF flags are set, the Input Buffer Full Flag (IBF) is set.

In Buffered Slave mode, the module can be configured to generate an interrupt on every read or write strobe (IRQM1:IRQM0 = 01). It can be configured to generate an interrupt on a read from Read Buffer 3 or a write to Write Buffer 3, which is essentially an interrupt every fourth read or write strobe (RQM1:RQM0 = 11). When interrupting every fourth byte for input data, all input buffer registers should be read to clear the IBxF flags. If these flags are not cleared, then there is a risk of hitting an overflow condition.

FIGURE 11-5: PARALLEL MASTER/SLAVE CONNECTION BUFFERED EXAMPLE



PIC18F87J11 FAMILY

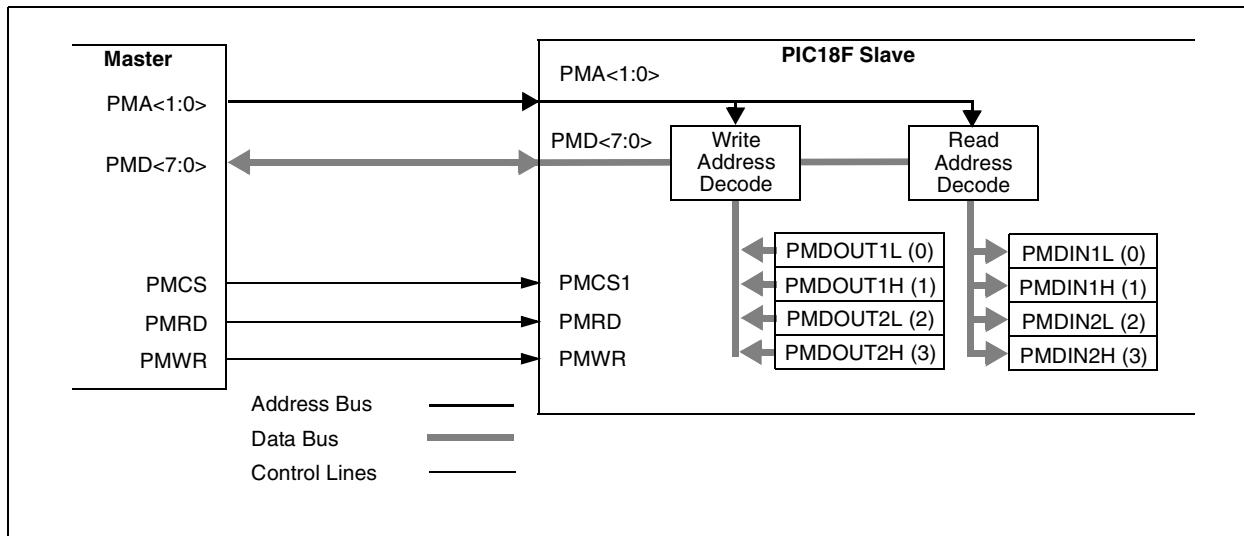
11.2.3 ADDRESSABLE PARALLEL SLAVE PORT MODE

In the Addressable Parallel Slave Port mode ($\text{PMMODEH}\langle 1:0 \rangle = 01$), the module is configured with two extra inputs, $\text{PMA}\langle 1:0 \rangle$, which are the address lines 1 and 0. This makes the 4-byte buffer space directly addressable as fixed pairs of read and write buffers. As with Buffered Legacy mode, data is output from PMDOUT1L , PMDOUT1H , PMDOUT2L and PMDOUT2H , and is read in PMDIN1L , PMDIN1H , PMDIN2L and PMDIN2H . Table 11-2 shows the buffer addressing for the incoming address to the input and output registers.

TABLE 11-2: SLAVE MODE BUFFER ADDRESSING

PMADDR $\langle 1:0 \rangle$	Output Register (Buffer)	Input Register (Buffer)
00	$\text{PMDOUT1L}(0)$	$\text{PMDIN1L}(0)$
01	$\text{PMDOUT1H}(1)$	$\text{PMDIN1H}(1)$
10	$\text{PMDOUT2L}(2)$	$\text{PMDIN2L}(2)$
11	$\text{PMDOUT2H}(3)$	$\text{PMDIN2H}(3)$

FIGURE 11-6: PARALLEL MASTER/SLAVE CONNECTION ADDRESSED BUFFER EXAMPLE

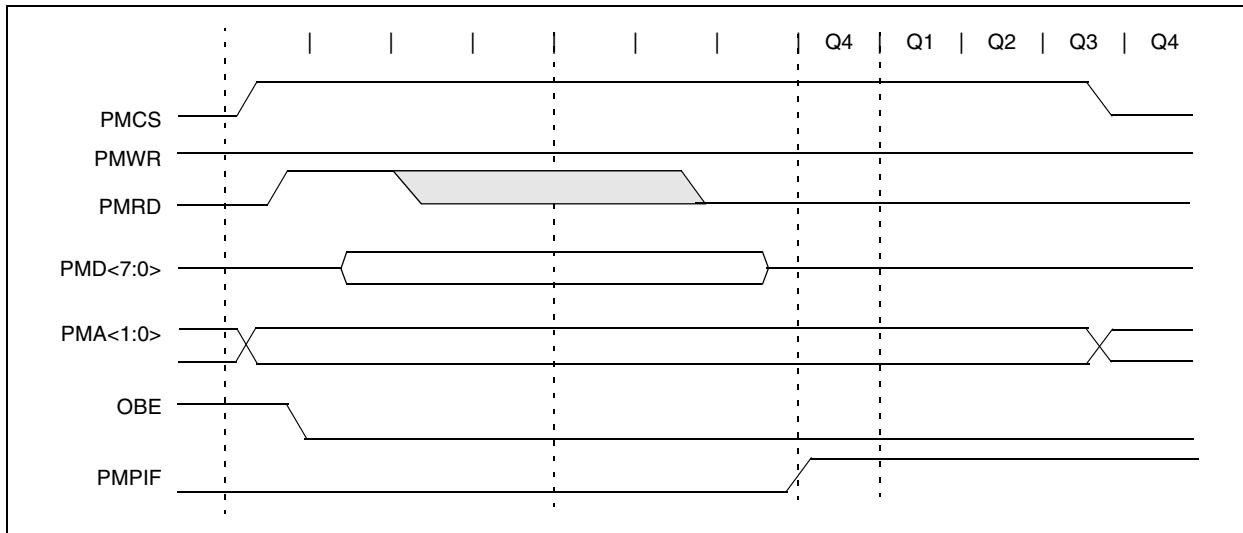


11.2.3.1 READ FROM SLAVE PORT

When chip select is active and a read strobe occurs ($\text{PMCS} = 1$ and $\text{PMRD} = 1$), the data from one of the four output bytes is presented onto $\text{PMD}_{<7:0>}$. Which byte is read depends on the 2-bit address placed on $\text{ADDR}_{<1:0>}$. Table 11-2 shows the corresponding output registers and their associated address.

When an output buffer is read, the corresponding OBxE bit is set. The OBE flag bit is set when all the buffers are empty. If any buffer is already empty ($\text{OBxE} = 1$), the next read to that buffer will generate an OBUF event.

FIGURE 11-7: PARALLEL SLAVE PORT READ WAVEFORMS

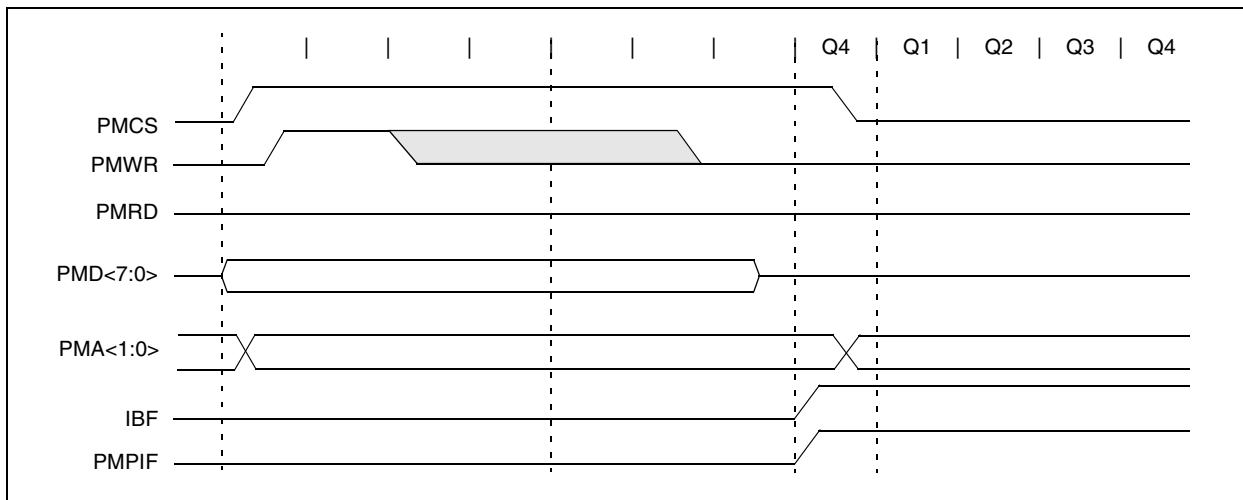


11.2.3.2 WRITE TO SLAVE PORT

When chip select is active and a write strobe occurs ($\text{PMCS} = 1$ and $\text{PMWR} = 1$), the data from $\text{PMD}_{<7:0>}$ is captured into one of the four input buffer bytes. Which byte is written depends on the 2-bit address placed on $\text{ADDR}_{<1:0>}$. Table 11-2 shows the corresponding input registers and their associated address.

When an input buffer is written, the corresponding IBxF bit is set. The IBF flag bit is set when all the buffers are written. If any buffer is already written ($\text{IBxF} = 1$), the next write strobe to that buffer will generate an OBUF event and the byte will be discarded.

FIGURE 11-8: PARALLEL SLAVE PORT WRITE WAVEFORMS



11.3 Master Port Modes

In its Master modes, the PMP module provides an 8-bit data bus, up to 16 bits of address, and all the necessary control signals to operate a variety of external parallel devices, such as memory devices, peripherals and slave microcontrollers. To use the PMP as a master, the module must be enabled ($\text{PMPEN} = 1$) and the mode must be set to one of the two possible Master modes ($\text{PMMODEH<1:0>} = 10$ or 11).

Because there are a number of parallel devices with a variety of control methods, the PMP module is designed to be extremely flexible to accommodate a range of configurations. Some of these features include:

- 8 and 16-Bit Data modes on an 8-bit data bus
- Configurable address/data multiplexing
- Up to two chip select lines
- Up to 16 selectable address lines
- Address auto-increment and auto-decrement
- Selectable polarity on all control lines
- Configurable wait states at different stages of the read/write cycle

11.3.1 PMP AND I/O PIN CONTROL

Multiple control bits are used to configure the presence or absence of control and address signals in the module. These bits are PTBEN, PTWREN, PTRDEN, and PTEN $<15:0>$. They give the user the ability to conserve pins for other functions and allow flexibility to control the external address. When any one of these bits is set, the associated function is present on its associated pin; when clear, the associated pin reverts to its defined I/O port function.

Setting a PTEN bit will enable the associated pin as an address pin and drive the corresponding data contained in the PMADDR register. Clearing the PTEN x bit will force the pin to revert to its original I/O function.

For the pins configured as chip select (PMCS1 or PMCS2) with the corresponding PTEN x bit set, chip select pins drive inactive data (with polarity defined by the CS1P and CS2P bits) when a read or write operation is not being performed. The PTEN0 and PTEN1 bits also control the PMALL and PMALH signals. When multiplexing is used, the associated address latch signals should be enabled.

11.3.2 READ/WRITE CONTROL

The PMP module supports two distinct read/write _signaling methods. In Master mode 1, read and write strobes are combined into a single control line, PMRD/PMWR. A second control line, PMENB, determines when a read or write action is to be taken. In Master mode 2, separate read and write strobes (PMRD and PMWR) are supplied on separate pins.

All control signals (PMRD, PMWR, PMBE, PMENB, PMAL and PMCSx) can be individually configured as either positive or negative polarity. Configuration is controlled by separate bits in the PMCONL register. Note that the polarity of control signals that share the same output pin (for example, PMWR and PMENB) are controlled by the same bit; the configuration depends on which Master Port mode is being used.

11.3.3 DATA WIDTH

The PMP supports data widths of both 8 and 16 bits. The data width is selected by the MODE16 bit ($\text{PMMODEH<2>} = 1$). Because the data path into and out of the module is only 8 bits wide, 16-bit operations are always handled in a multiplexed fashion, with the Least Significant Byte of data being presented first. To differentiate data bytes, the Byte Enable (PMBE) control strobe is used to signal when the Most Significant Byte of data is being presented on the data lines.

11.3.4 ADDRESS MULTIPLEXING

In either of the Master modes ($\text{PMMODEH<1:0>} = 1x$), the user can configure the address bus to be multiplexed together with the data bus. This is accomplished using the ADRMUX1:ADRMUX0 bits (PMCONH $<4:3>$). There are three address multiplexing modes available; typical pinout configurations for these modes are shown in Figure 11-9, Figure 11-10 and Figure 11-11.

In Demultiplexed mode ($\text{PMCONH<4:3>} = 00$), data and address information are completely separated. Data bits are presented on PMD $<7:0>$, and address bits are presented on PMADDRH $<7:0>$ and PMADDRL $<7:0>$.

In Partially Multiplexed mode ($\text{PMCONH<4:3>} = 01$), the lower eight bits of the address are multiplexed with the data pins on PMD $<7:0>$. The upper eight bits of address are unaffected and are presented on PMADDRH $<7:0>$. The PMA0 pin is used as an address latch and presents the Address Latch Low (PMALL) enable strobe. The read and write sequences are extended by a complete CPU cycle during which the address is presented on the PMD $<7:0>$ pins.

In Fully Multiplexed mode ($\text{PMCONH<4:3>} = 10$), the entire 16 bits of the address are multiplexed with the data pins on PMD $<7:0>$. The PMA0 and PMA1 pins are used to present Address Latch Low (PMALL) enable and Address Latch High (PMALH) enable strobes, respectively. The read and write sequences are extended by two complete CPU cycles. During the first cycle, the lower eight bits of the address are presented on the PMD $<7:0>$ pins with the PMALL strobe active. During the second cycle, the upper eight bits of the address are presented on the PMD $<7:0>$ pins with the PMALH strobe active. In the event the upper address bits are configured as chip select pins, the corresponding address bits are automatically forced to '0'.

PIC18F87J11 FAMILY

FIGURE 11-9: DEMULTIPLEXED ADDRESSING MODE (SEPARATE READ AND WRITE STROBES, TWO CHIP SELECTS)

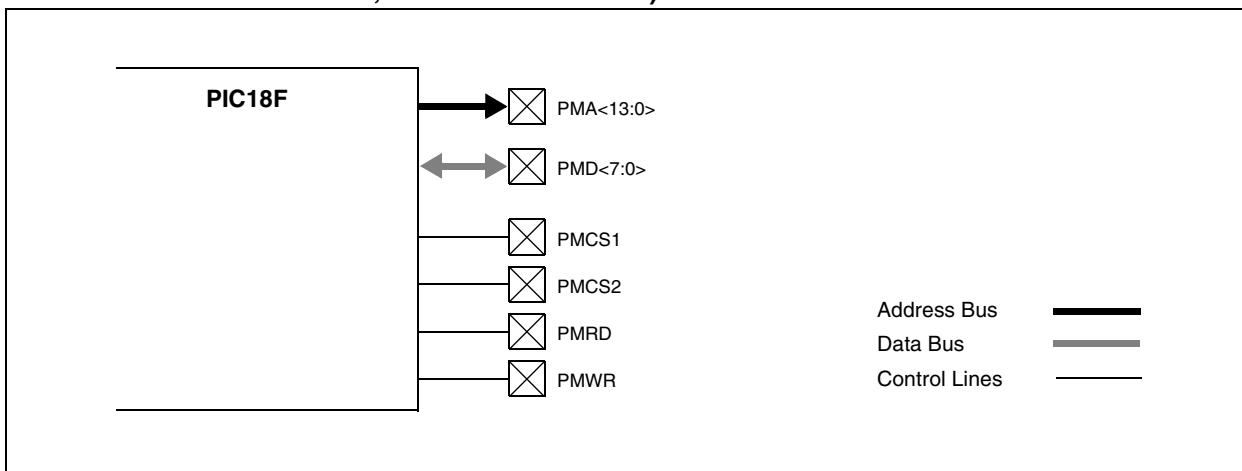


FIGURE 11-10: PARTIALLY MULTIPLEXED ADDRESSING MODE (SEPARATE READ AND WRITE STROBES, TWO CHIP SELECTS)

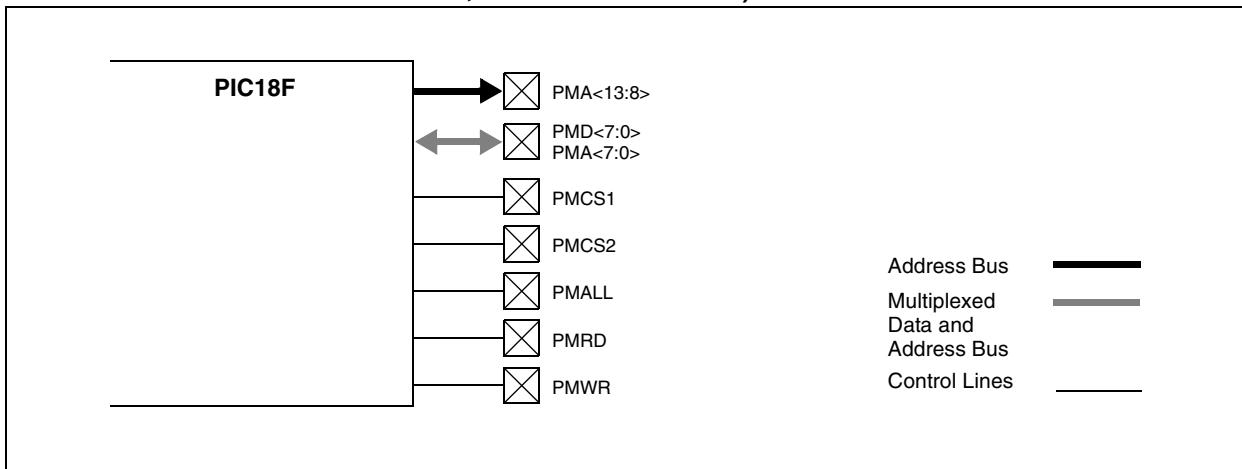
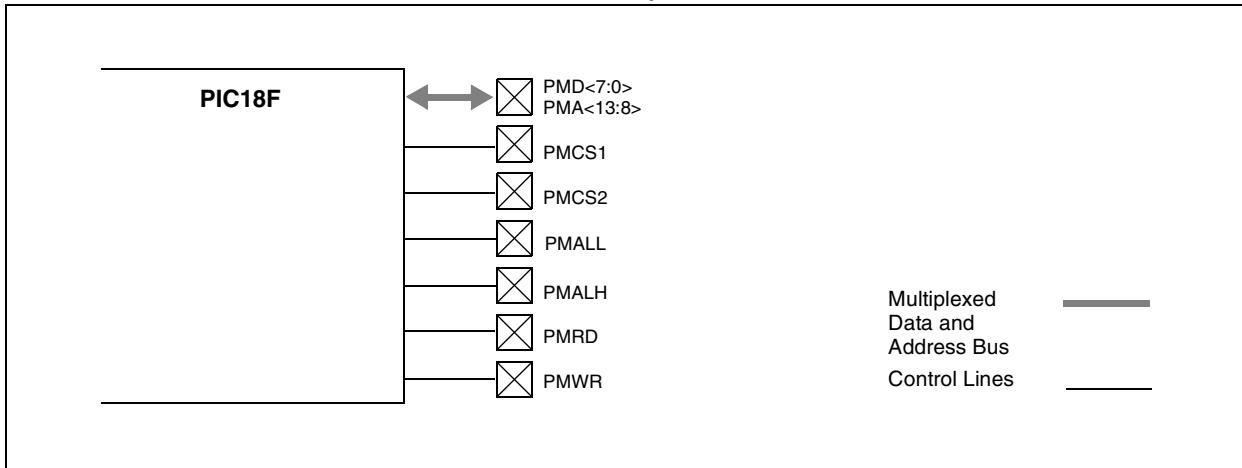


FIGURE 11-11: FULLY MULTIPLEXED ADDRESSING MODE (SEPARATE READ AND WRITE STROBES, TWO CHIP SELECTS)



PIC18F87J11 FAMILY

11.3.5 CHIP SELECT FEATURES

Up to two chip select lines, PMCS1 and PMCS2, are available for the Master modes of the PMP. The two chip select lines are multiplexed with the Most Significant bits of the address bus (PMADDRH<6> and PMADDRH<7>). When a pin is configured as a chip select, it is not included in any address auto-increment/decrement. The function of the chip select signals is configured using the chip select function bits (PMCONL <7:6>).

11.3.6 AUTO-INCREMENT/DECREMENT

While the module is operating in one of the Master modes, the INCM bits (PMMODEH<3:4>) control the behavior of the address value. The address can be made to automatically increment or decrement after each read and write operation. The address increments once each operation is completed and the BUSY bit goes to '0'. If the chip select signals are disabled and configured as address bits, the bits will participate in the increment and decrement operations; otherwise, the CS2 and CS1 bit values will be unaffected.

11.3.7 WAIT STATES

In Master mode, the user has control over the duration of the read, write and address cycles by configuring the module wait states. Three portions of the cycle, the beginning, middle, and end, are configured using the corresponding WAITBx, WAITMx and WAITEx bits in the PMMODEL register.

The WAITB1:WAITB0 bits (PMMODEL<7:6>) set the number of wait cycles for the data setup prior to the PMRD/PWWT strobe in Mode 10, or prior to the PMENB strobe in Mode 11. The WAITM3:WAITM0 bits (PMMODEL<5:2>) set the number of wait cycles for the PMRD/PWWT strobe in Mode 10, or for the PMENB strobe in Mode 11. When this wait state setting is 0, then WAITB and WAITE have no effect. The WAITE1:WAITE0 bits (PMMODEL<1:0>) define the number of wait cycles for the data hold time after the PMRD/PWWT strobe in Mode 10, or after the PMENB strobe in Mode 11.

11.3.8 READ OPERATION

To perform a read on the Parallel Master Port, the user reads the PMDIN1L register. This causes the PMP to output the desired values on the chip select lines and the address bus. Then the read line (PMRD) is strobed. The read data is placed into the PMDIN1L register.

If the 16-bit mode is enabled (MODE16 = 1), the read of the low byte of the PMDIN1L register will initiate two bus reads. The first read data byte is placed into the PMDIN1L register, and the second read data is placed into the PMDIN1H.

Note that the read data obtained from the PMDIN1L register is actually the read value from the previous read operation. Hence, the first user read will be a dummy read to initiate the first bus read and fill the read register. Also, the requested read value will not be ready until after the BUSY bit is observed low. Thus, in a back-to-back read operation, the data read from the register will be the same for both reads. The next read of the register will yield the new value.

11.3.9 WRITE OPERATION

To perform a write onto the parallel bus, the user writes to the PMDIN1L register. This causes the module to first output the desired values on the chip select lines and the address bus. The write data from the PMDIN1L register is placed onto the PMD<7:0> data bus. Then the write line (PMWR) is strobed. If the 16-bit mode is enabled (MODE16 = 1), the write to the PMDIN1L register will initiate two bus writes. First write will consist of the data contained in PMDIN1L and the second write will contain the PMDIN1H.

11.3.10 PARALLEL MASTER PORT STATUS

11.3.10.1 The BUSY Bit

In addition to the PMP interrupt, a BUSY bit is provided to indicate the status of the module. This bit is only used in Master mode. While any read or write operation is in progress, the BUSY bit is set for all but the very last CPU cycle of the operation. In effect, if a single-cycle read or write operation is requested, the BUSY bit will never be active. This allows back-to-back transfers. While the bit is set, any request by the user to initiate a new operation will be ignored (i.e., writing or reading the lower byte of the PMDIN1L register will not initiate either a read nor a write).

11.3.10.2 INTERRUPTS

When the PMP module interrupt is enabled for Master mode, the module will interrupt on every completed read or write cycle; otherwise, the BUSY bit is available to query the status of the module.

11.3.11 MASTER MODE TIMING

This section contains a number of timing examples that represent the common Master mode configuration options. These options vary from 8-bit to 16-bit data, fully demultiplexed to fully multiplexed address, as well as wait states.

FIGURE 11-12: READ AND WRITE TIMING, 8-BIT DATA, DEMULTIPLEXED ADDRESS

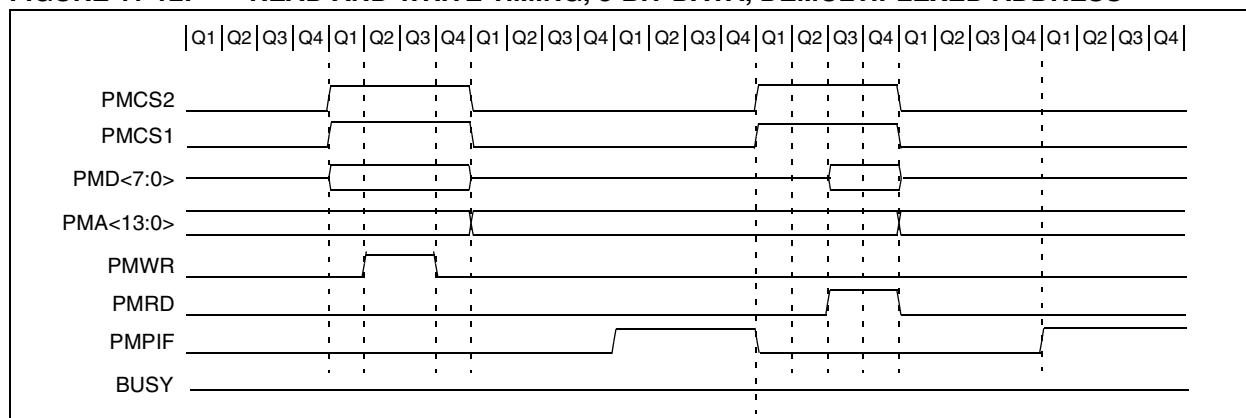


FIGURE 11-13: READ TIMING, 8-BIT DATA, PARTIALLY MULTIPLEXED ADDRESS

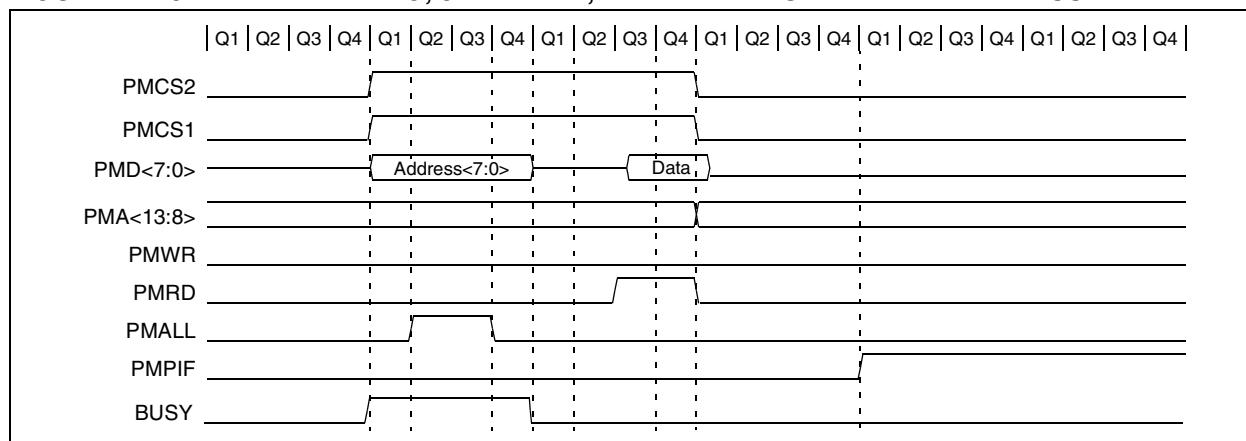
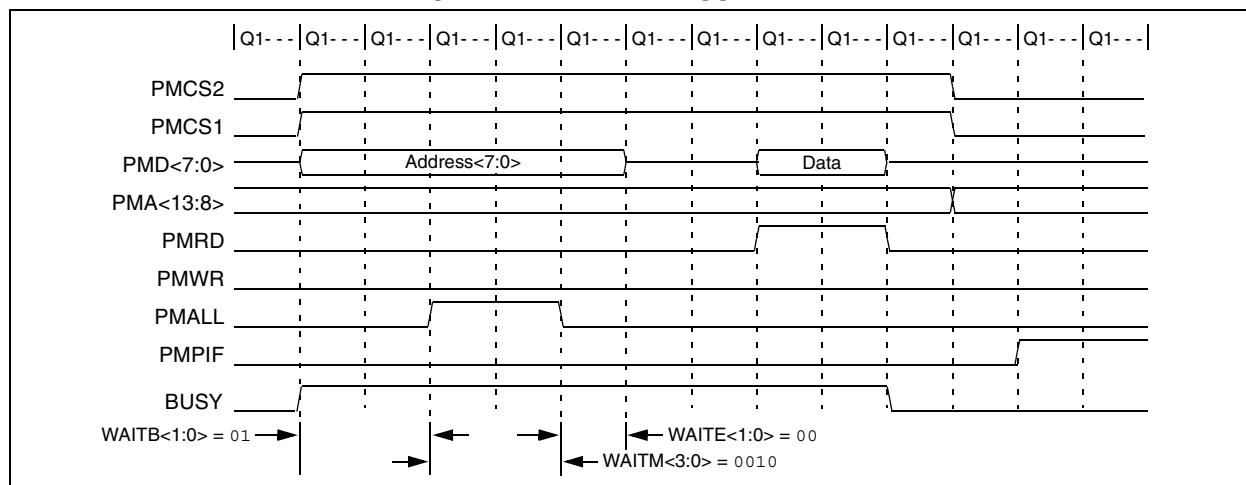


FIGURE 11-14: READ TIMING, 8-BIT DATA, WAIT STATES ENABLED, PARTIALLY MULTIPLEXED ADDRESS



PIC18F87J11 FAMILY

FIGURE 11-15: WRITE TIMING, 8-BIT DATA, PARTIALLY MULTIPLEXED ADDRESS

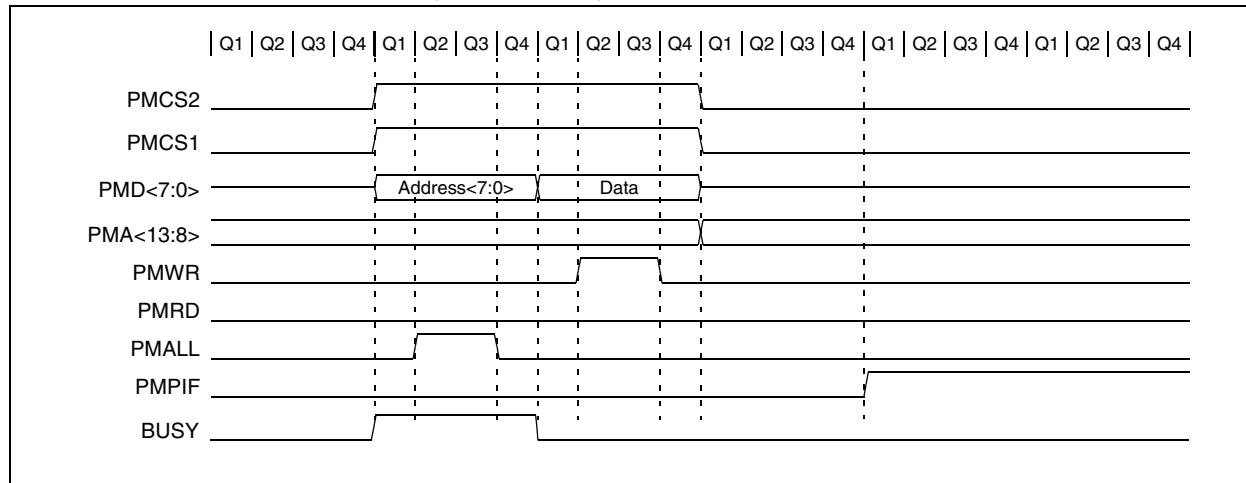


FIGURE 11-16: WRITE TIMING, 8-BIT DATA, WAIT STATES ENABLED, PARTIALLY MULTIPLEXED ADDRESS

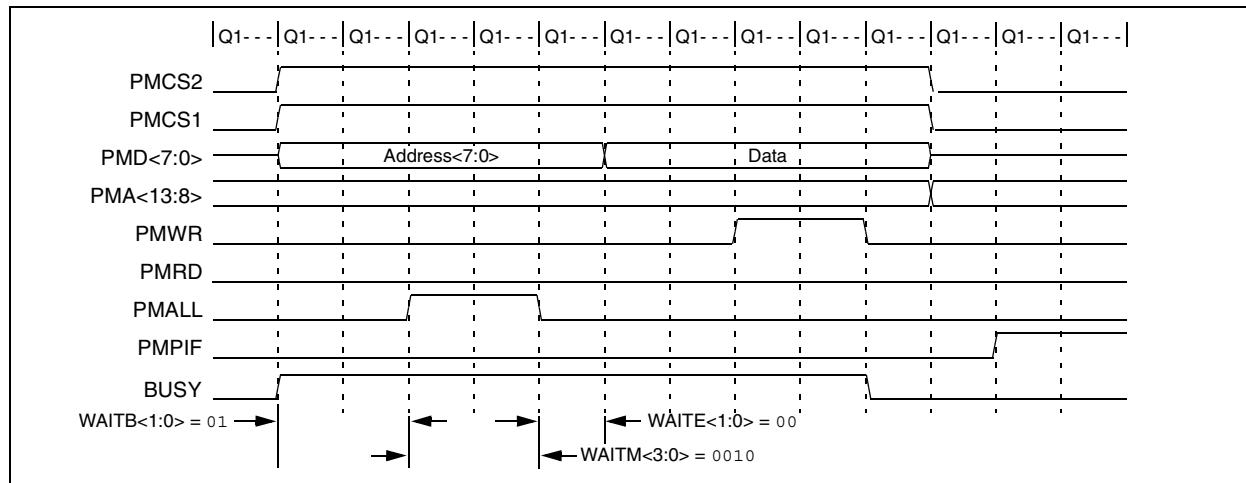


FIGURE 11-17: READ TIMING, 8-BIT DATA, PARTIALLY MULTIPLEXED ADDRESS, ENABLE STROBE

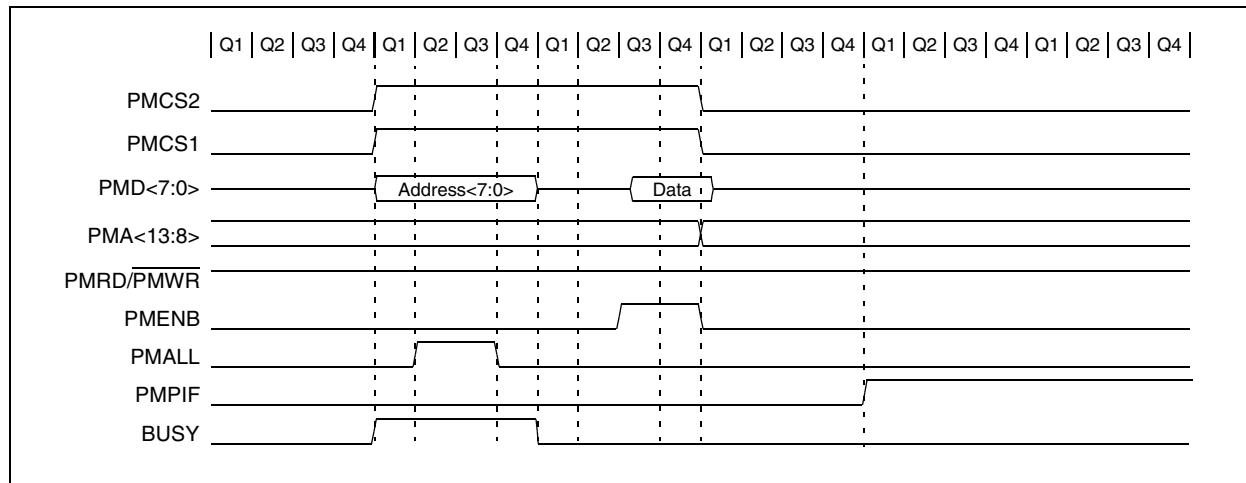


FIGURE 11-18: WRITE TIMING, 8-BIT DATA, PARTIALLY MULTIPLEXED ADDRESS, ENABLE STROBE

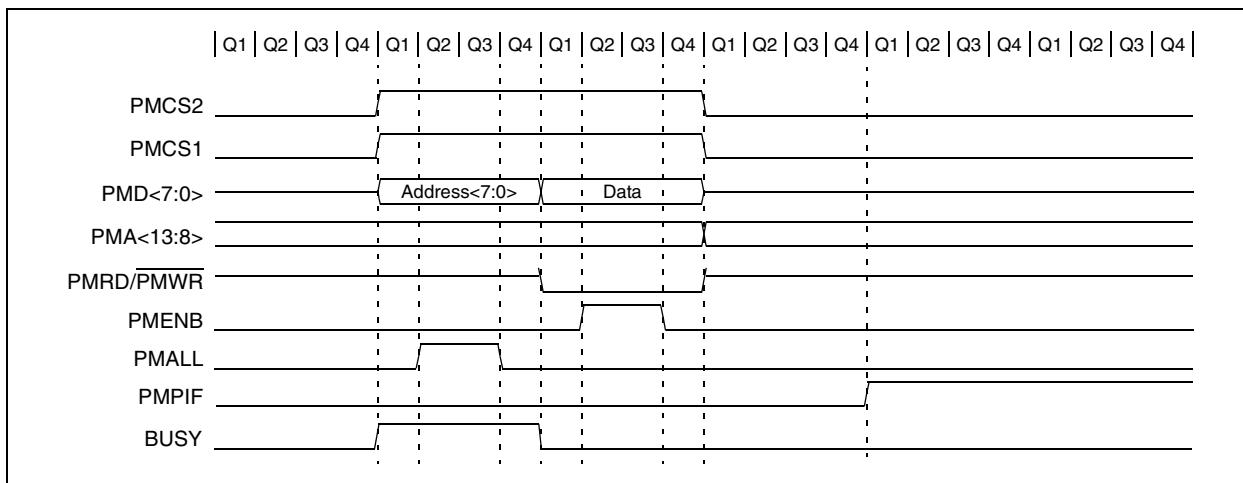


FIGURE 11-19: READ TIMING, 8-BIT DATA, FULLY MULTIPLEXED 16-BIT ADDRESS

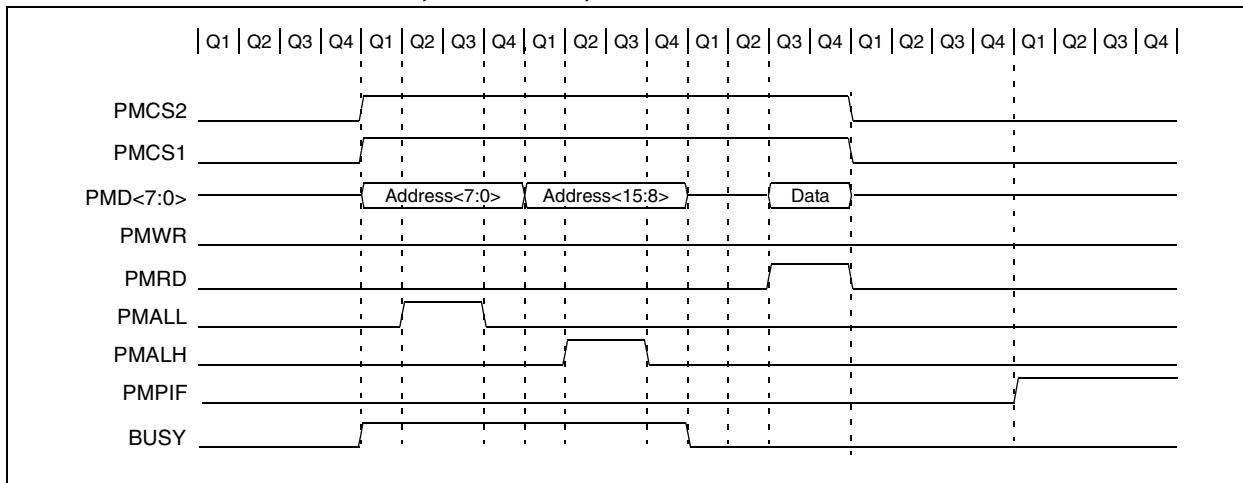
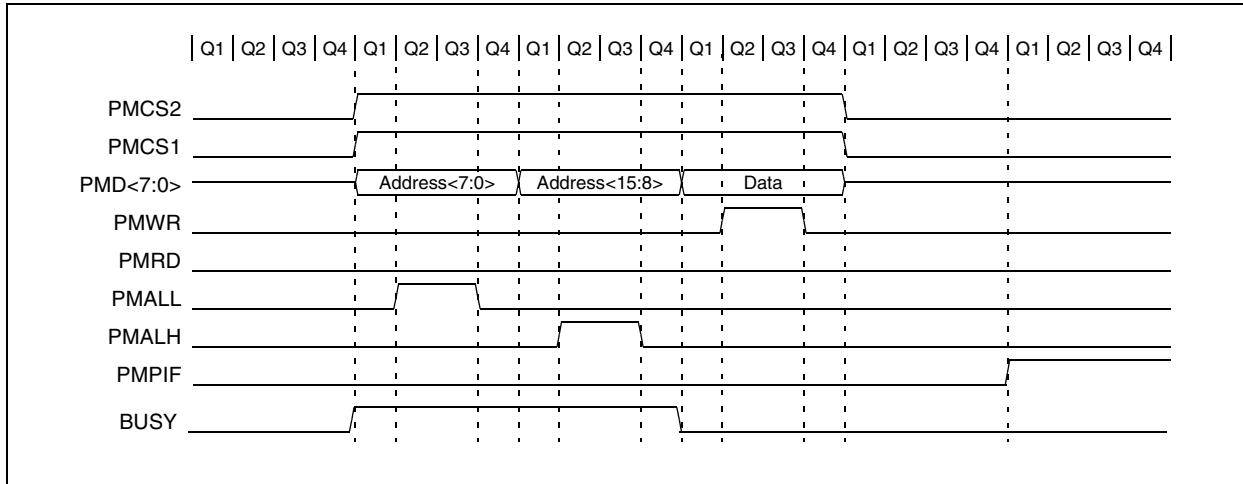


FIGURE 11-20: WRITE TIMING, 8-BIT DATA, FULLY MULTIPLEXED 16-BIT ADDRESS



PIC18F87J11 FAMILY

FIGURE 11-21: READ TIMING, 16-BIT DATA, DEMULTIPLEXED ADDRESS

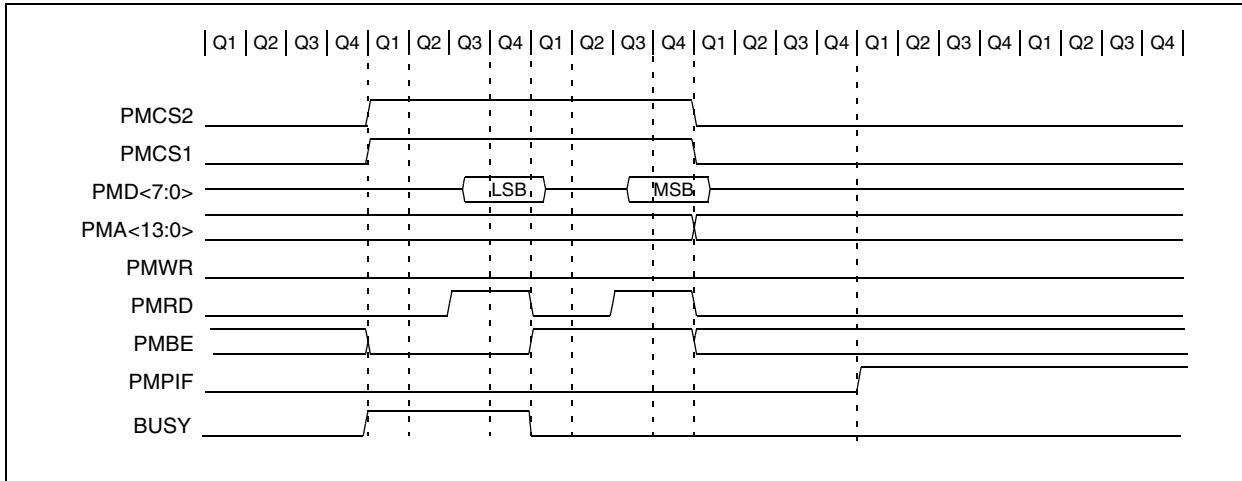
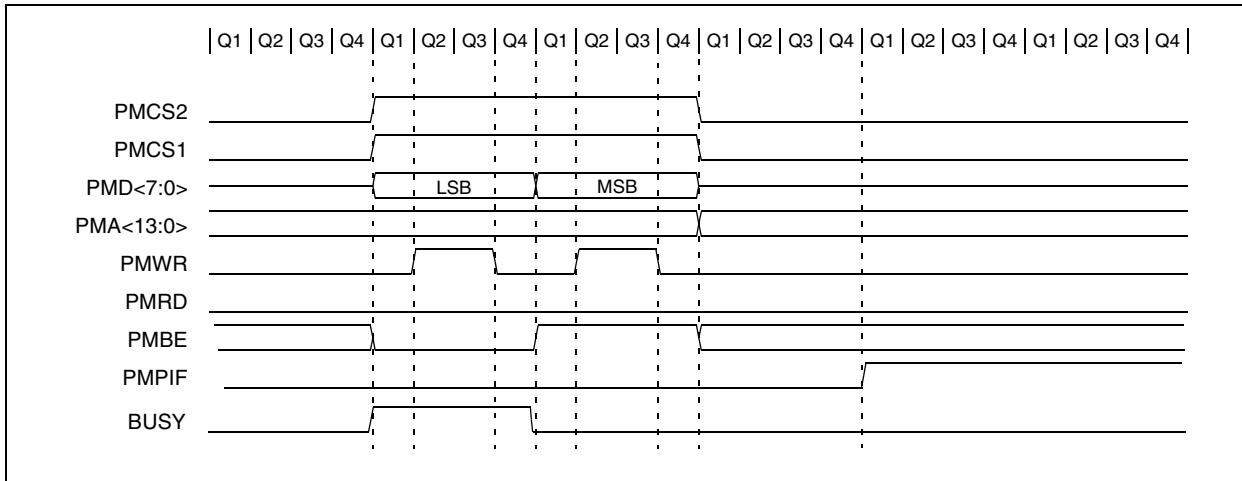
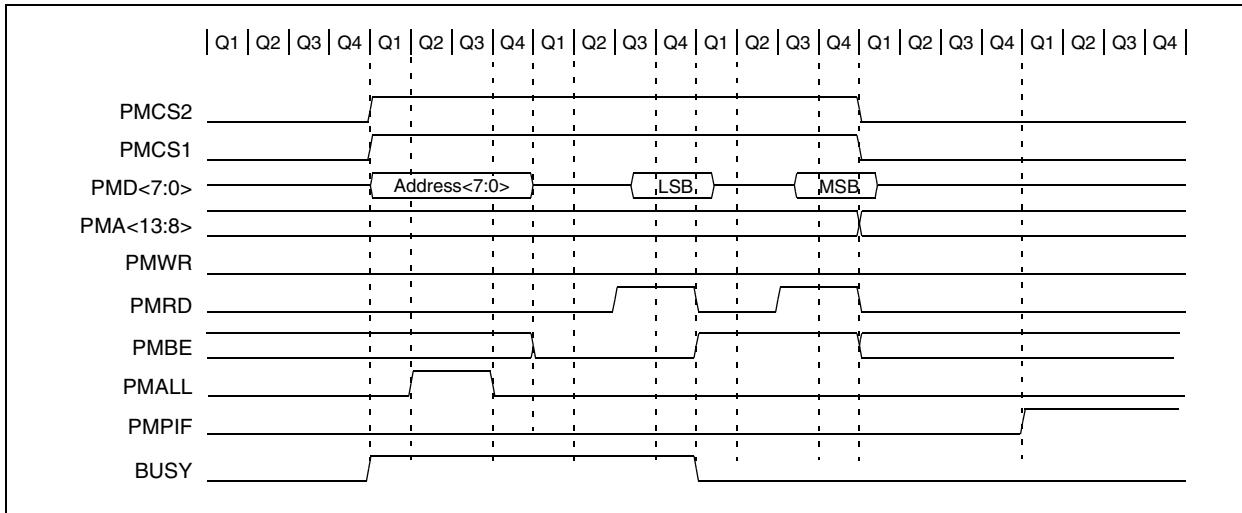


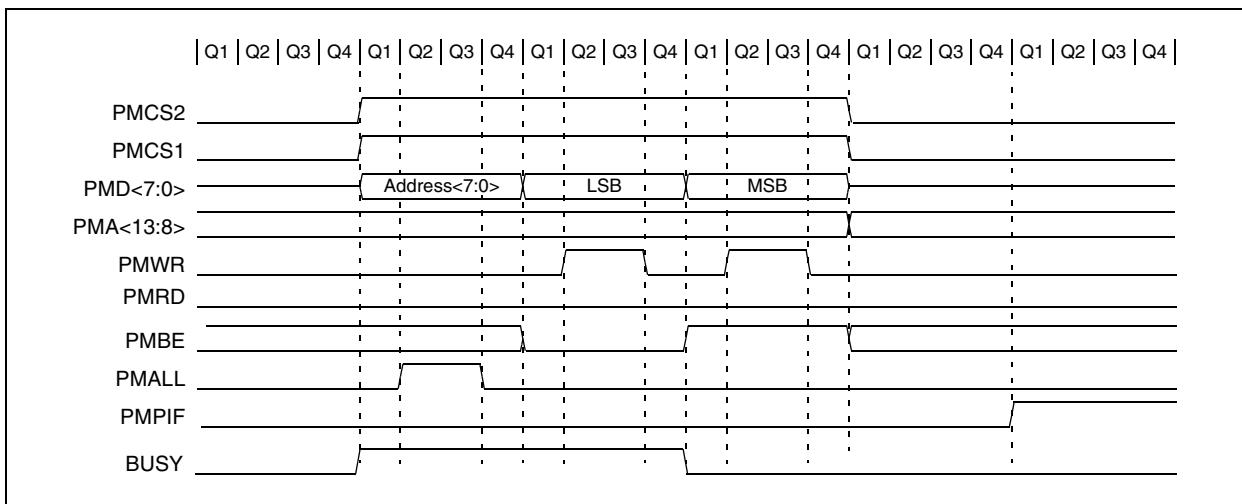
FIGURE 11-22: WRITE TIMING, 16-BIT DATA, DEMULTIPLEXED ADDRESS



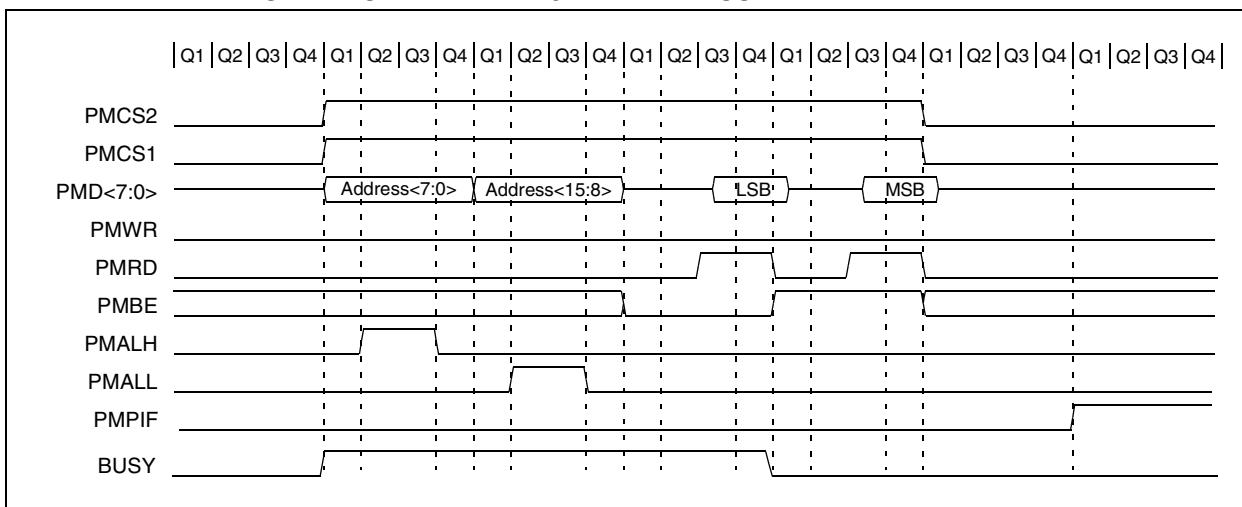
**FIGURE 11-23: READ TIMING, 16-BIT MULTIPLEXED DATA,
PARTIALLY MULTIPLEXED ADDRESS**



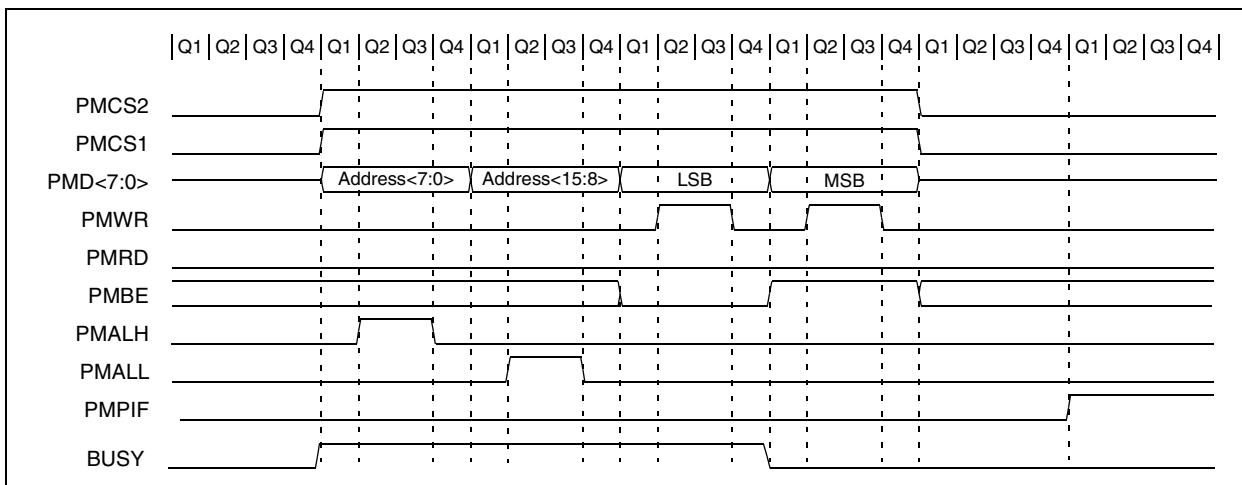
**FIGURE 11-24: WRITE TIMING, 16-BIT MULTIPLEXED DATA,
PARTIALLY MULTIPLEXED ADDRESS**



**FIGURE 11-25: READ TIMING, 16-BIT MULTIPLEXED DATA,
FULLY MULTIPLEXED 16-BIT ADDRESS**



**FIGURE 11-26: WRITE TIMING, 16-BIT MULTIPLEXED DATA,
FULLY MULTIPLEXED 16-BIT ADDRESS**



PIC18F87J11 FAMILY

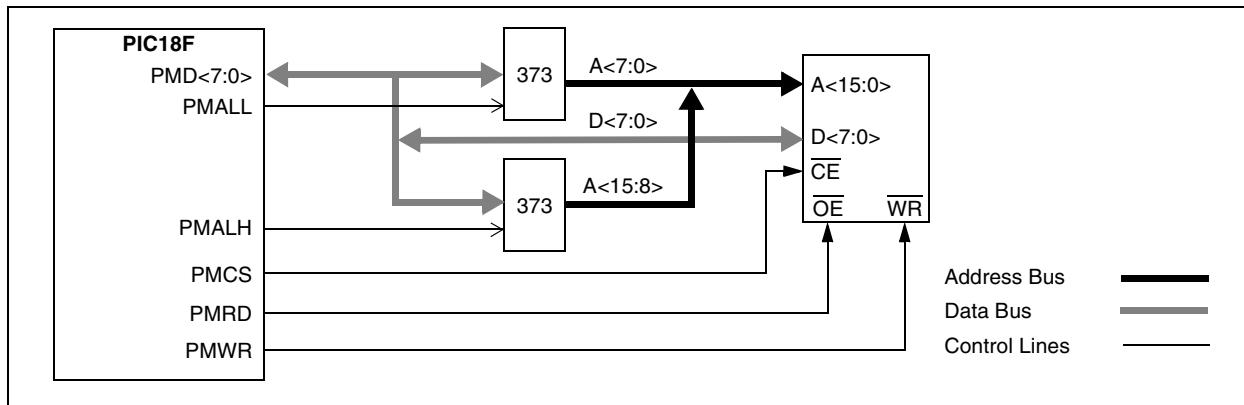
11.4 Application Examples

This section introduces some potential applications for the PMP module.

11.4.1 MULTIPLEXED MEMORY OR PERIPHERAL

Figure 11-27 demonstrates the hookup of a memory or other addressable peripheral in Full Multiplex mode. Consequently, this mode achieves the best pin saving from the microcontroller perspective. However, for this configuration, there needs to be some external latches to maintain the address.

FIGURE 11-27: EXAMPLE OF A MULTIPLEXED ADDRESSING APPLICATION



11.4.2 PARTIALLY MULTIPLEXED MEMORY OR PERIPHERAL

Partial multiplexing implies using more pins; however, for a few extra pins, some extra performance can be achieved. Figure 11-28 shows an example of a mem-

ory or peripheral that is partially multiplexed with an external latch. If the peripheral has internal latches as shown in Figure 11-29, then no extra circuitry is required except for the peripheral itself.

FIGURE 11-28: EXAMPLE OF A PARTIALLY MULTIPLEXED ADDRESSING APPLICATION

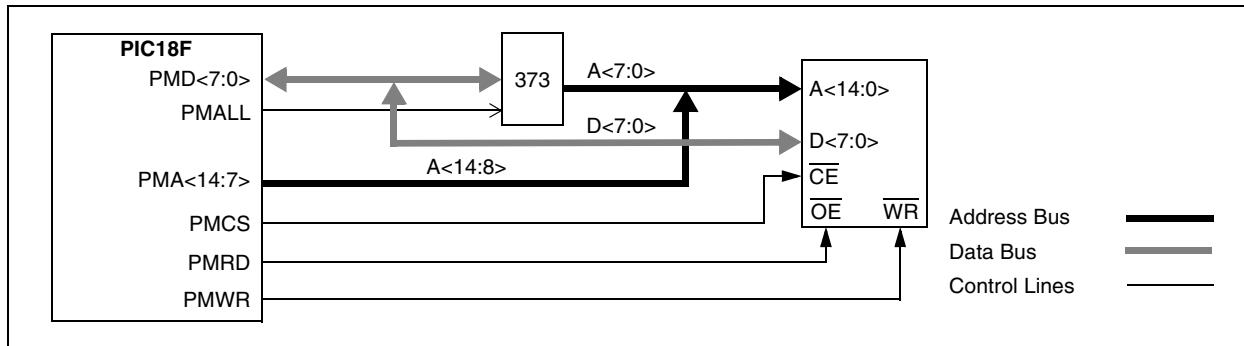
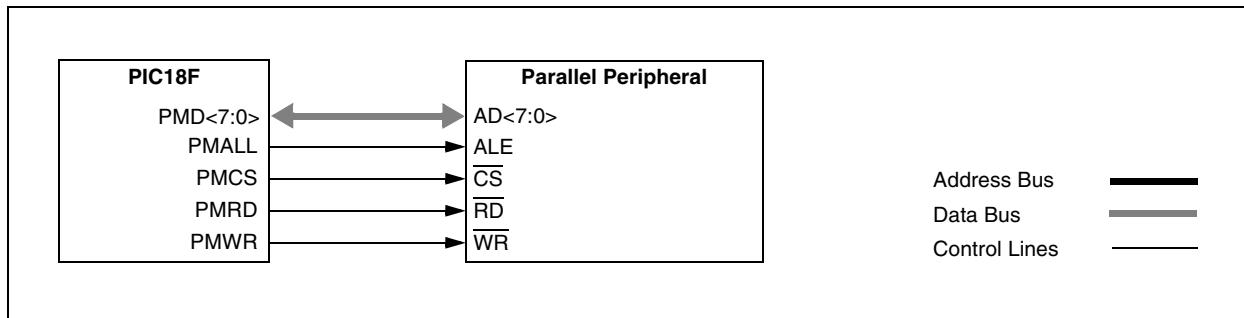


FIGURE 11-29: EXAMPLE OF AN 8-BIT MULTIPLEXED ADDRESS AND DATA APPLICATION



11.4.3 PARALLEL EEPROM EXAMPLE

Figure 11-30 shows an example connecting parallel EEPROM to the PMP. Figure 11-31 shows a slight variation to this, configuring the connection for 16-bit data from a single EEPROM.

FIGURE 11-30: PARALLEL EEPROM EXAMPLE (UP TO 15-BIT ADDRESS, 8-BIT DATA)

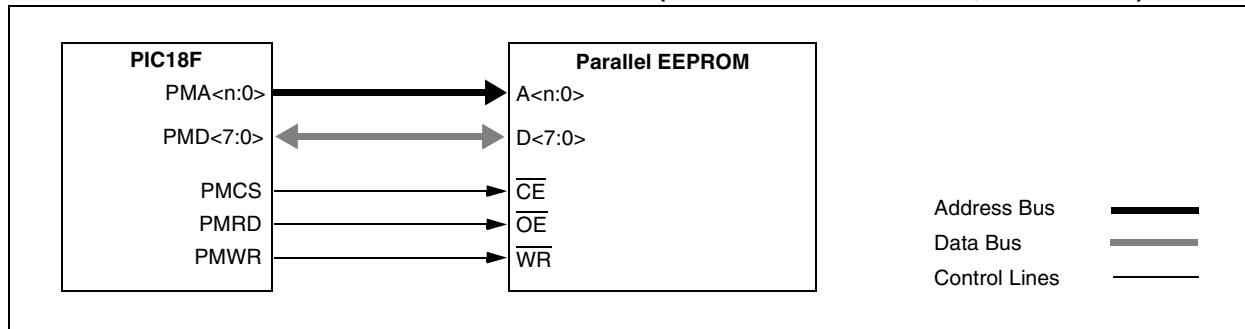
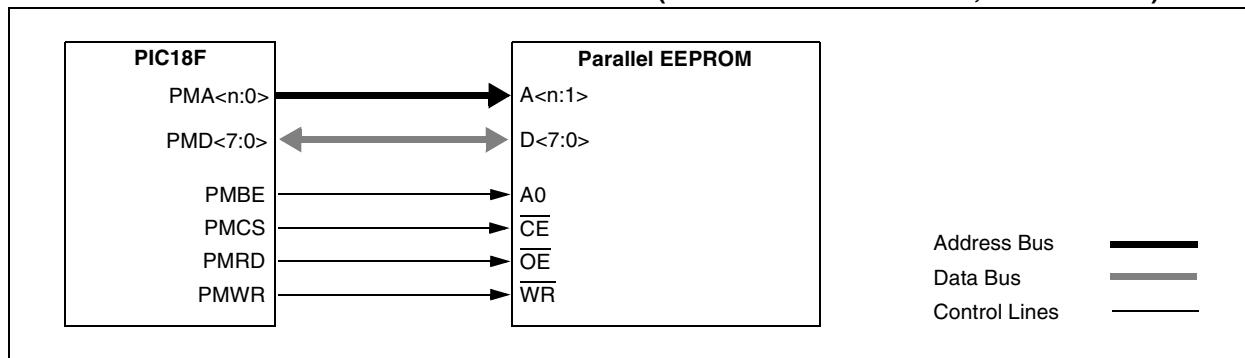


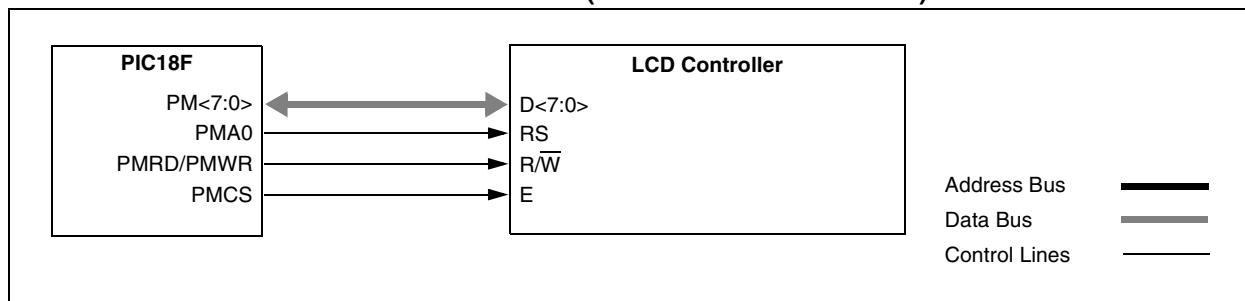
FIGURE 11-31: PARALLEL EEPROM EXAMPLE (UP TO 15-BIT ADDRESS, 16-BIT DATA)



11.4.4 LCD CONTROLLER EXAMPLE

The PMP module can be configured to connect to a typical LCD controller interface, as shown in Figure 11-32. In this case, the PMP module is configured for active-high control signals since common LCD displays require active-high control.

FIGURE 11-32: LCD CONTROL EXAMPLE (BYTE MODE OPERATION)



PIC18F87J11 FAMILY

TABLE 11-3: REGISTERS ASSOCIATED WITH PMP MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:						
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55						
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58						
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58						
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58						
PMCONH	PM PEN	—	PSIDL	ADRMUX1	ADRMUX0	PTBEEN	PTWREN	PTRDEN	60						
PMCONL	CSF1	CSF0	ALP	CS2P	CS1P	BEP	WRSP	RDSP	60						
PMADDRH/	CS2	CS1	Parallel Master Port Address, High Byte						60						
PMDO OUT1H ⁽¹⁾	Parallel Port Out Data High Byte (Buffer 1)								60						
PMADDR/	Parallel Master Port Address Low Byte								60						
PMDO OUT1L ⁽¹⁾	Parallel Port Out Data Low Byte (Buffer 0)								60						
PMDO OUT2H	Parallel Port Out Data High Byte (Buffer 3)								60						
PMDO OUT2L	Parallel Port Out Data Low Byte (Buffer 2)								60						
PMDIN1H	Parallel Port In Data High Byte (Buffer 1)								60						
PMDIN1L	Parallel Port In Data Low Byte (Buffer 0)								60						
PMDIN2H	Parallel Port In Data High Byte (Buffer 3)								60						
PMDIN2L	Parallel Port In Data Low Byte (Buffer 2)								60						
PMMODEH	BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0	60						
PMMODEL	WAITB1	WAITB0	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1	WAITE0	60						
PMEH	PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8	60						
PMEL	PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0	60						
PMSTATH	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F	60						
PMSTATL	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E	60						
PADCFG1 ⁽²⁾	—	—	—	—	—	—	—	PMPTTL	56						

Legend: — = unimplemented, read as '0'. Shaded cells are not used during PMP operation.

Note 1: The PMADDRH/PMDO OUT1H and PMADDR/PMDO OUT1L register pairs share the physical registers and addresses, but have different functions determined by the module's operating mode.

2: Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.

12.0 TIMER0 MODULE

The Timer0 module incorporates the following features:

- Software selectable operation as a timer or counter in both 8-bit or 16-bit modes
- Readable and writable registers
- Dedicated 8-bit, software programmable prescaler
- Selectable clock source (internal or external)
- Edge select for external clock
- Interrupt-on-overflow

The T0CON register (Register 12-1) controls all aspects of the module's operation, including the prescale selection. It is both readable and writable.

A simplified block diagram of the Timer0 module in 8-bit mode is shown in Figure 12-1. Figure 12-2 shows a simplified block diagram of the Timer0 module in 16-bit mode.

REGISTER 12-1: T0CON: TIMER0 CONTROL 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
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7	bit 0						

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

bit 7	TMR0ON: Timer0 On/Off Control bit 1 = Enables Timer0 0 = Stops Timer0
bit 6	T08BIT: Timer0 8-Bit/16-Bit Control bit 1 = Timer0 is configured as an 8-bit timer/counter 0 = Timer0 is configured as a 16-bit timer/counter
bit 5	T0CS: Timer0 Clock Source Select bit 1 = Transition on T0CKI pin 0 = Internal instruction cycle clock (CLKO)
bit 4	T0SE: Timer0 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: Timer0 Prescaler Assignment bit 1 = Timer0 prescaler is not assigned. Timer0 clock input bypasses prescaler. 0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.
bit 2-0	T0PS2:T0PS0: Timer0 Prescaler Select bits 111 = 1:256 Prescale value 110 = 1:128 Prescale value 101 = 1:64 Prescale value 100 = 1:32 Prescale value 011 = 1:16 Prescale value 010 = 1:8 Prescale value 001 = 1:4 Prescale value 000 = 1:2 Prescale value

PIC18F87J11 FAMILY

12.1 Timer0 Operation

Timer0 can operate as either a timer or a counter. The mode is selected with the T0CS bit (T0CON<5>). In Timer mode ($T0CS = 0$), the module increments on every clock by default unless a different prescaler value is selected (see **Section 12.3 “Prescaler”**). If the TMR0 register is written to, 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.

The Counter mode is selected by setting the T0CS bit ($= 1$). In this mode, Timer0 increments either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit, T0SE (T0CON<4>); clearing this bit selects the rising edge. Restrictions on the external clock input are discussed below.

An external clock source can be used to drive Timer0; however, it must meet certain requirements to ensure that the external clock can be synchronized with the

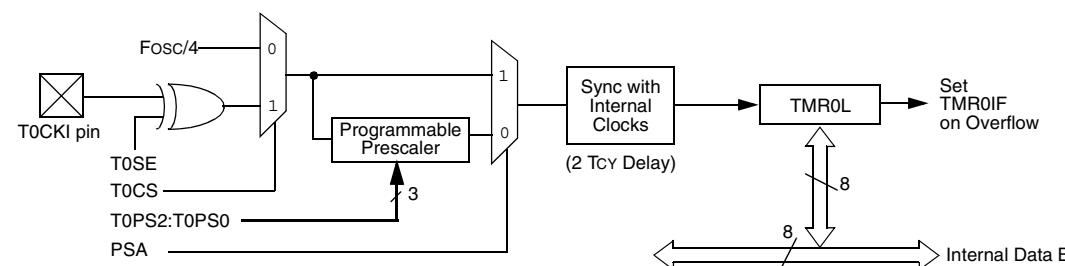
internal phase clock (Tosc). There is a delay between synchronization and the onset of incrementing the timer/counter.

12.2 Timer0 Reads and Writes in 16-Bit Mode

TMR0H is not the actual high byte of Timer0 in 16-bit mode. It is actually a buffered version of the real high byte of Timer0 which is not directly readable nor writable (refer to Figure 12-2). TMR0H is updated with the contents of the high byte of Timer0 during a read of TMR0L. This provides the ability to read all 16 bits of Timer0 without having to verify that the read of the high and low byte were valid, due to a rollover between successive reads of the high and low byte.

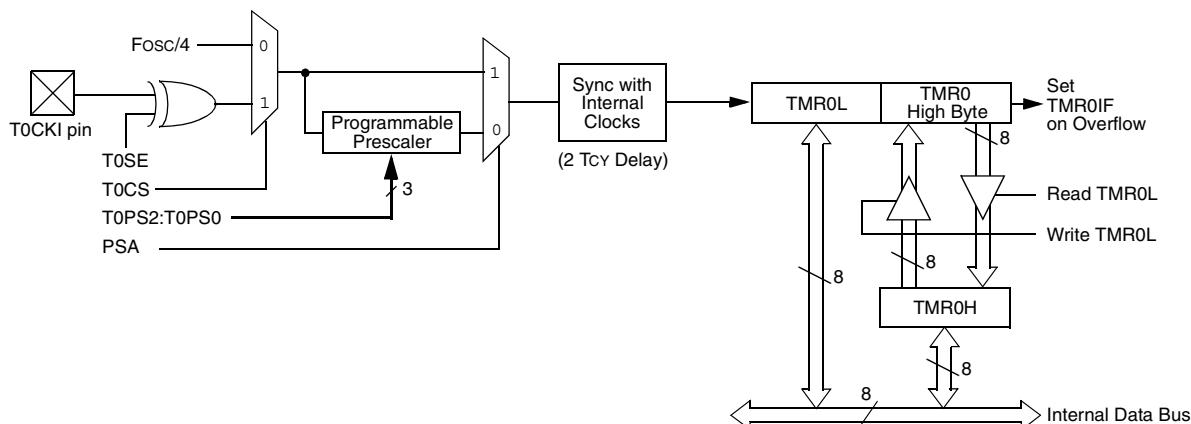
Similarly, a write to the high byte of Timer0 must also take place through the TMR0H Buffer register. The high byte is updated with the contents of TMR0H when a write occurs to TMR0L. This allows all 16 bits of Timer0 to be updated at once.

FIGURE 12-1: TIMER0 BLOCK DIAGRAM (8-BIT MODE)



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI max. prescale.

FIGURE 12-2: TIMER0 BLOCK DIAGRAM (16-BIT MODE)



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI max. prescale.

12.3 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module. The prescaler is not directly readable or writable. Its value is set by the PSA and T0PS2:T0PS0 bits (T0CON<3:0>) which determine the prescaler assignment and prescale ratio.

Clearing the PSA bit assigns the prescaler to the Timer0 module. When it is assigned, prescale values from 1:2 through 1:256 in power-of-2 increments are selectable.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., CLRF TMR0, MOVWF TMR0, BSF TMR0, etc.) clear the prescaler count.

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

12.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control and can be changed "on-the-fly" during program execution.

12.4 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h in 8-bit mode, or from FFFFh to 0000h in 16-bit mode. This overflow sets the TMR0IF flag bit. The interrupt can be masked by clearing the TMR0IE bit (INTCON<5>). Before re-enabling the interrupt, the TMR0IF bit must be cleared in software by the Interrupt Service Routine.

Since Timer0 is shut down in Sleep mode, the TMR0 interrupt cannot awaken the processor from Sleep.

TABLE 12-1: REGISTERS ASSOCIATED WITH TIMER0

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
TMR0L	Timer0 Register Low Byte								56
TMR0H	Timer0 Register High Byte								56
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	56
TRISA	TRISA7 ⁽¹⁾	TRISA6 ⁽¹⁾	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	58

Legend: — = unimplemented, read as '0'. Shaded cells are not used by Timer0.

Note 1: These bits are only available in select oscillator modes (FOSC2 Configuration bit = 0); otherwise, they are unimplemented.

PIC18F87J11 FAMILY

NOTES:

13.0 TIMER1 MODULE

The Timer1 timer/counter module incorporates these features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR1H and TMR1L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt on overflow
- Reset on ECCPx Special Event Trigger
- Device clock status flag (T1RUN)

A simplified block diagram of the Timer1 module is shown in Figure 13-1. A block diagram of the module's operation in Read/Write mode is shown in Figure 13-2.

The module incorporates its own low-power oscillator to provide an additional clocking option. The Timer1 oscillator can also be used as a low-power clock source for the microcontroller in power-managed operation.

Timer1 can also be used to provide Real-Time Clock (RTC) functionality to applications with only a minimal addition of external components and code overhead.

Timer1 is controlled through the T1CON Control register (Register 13-1). It also contains the Timer1 Oscillator Enable bit (T1OSCEN). Timer1 can be enabled or disabled by setting or clearing control bit, TMR1ON (T1CON<0>).

REGISTER 13-1: T1CON: TIMER1 CONTROL REGISTER⁽¹⁾

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
bit 7							bit 0

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

bit 7 **RD16:** 16-Bit Read/Write Mode Enable bit
 1 = Enables register read/write of Timer1 in one 16-bit operation
 0 = Enables register read/write of Timer1 in two 8-bit operations

bit 6 **T1RUN:** Timer1 System Clock Status bit
 1 = Device clock is derived from Timer1 oscillator
 0 = Device clock is derived from another source

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 bit
 1 = Timer1 oscillator is enabled
 0 = Timer1 oscillator is shut off
 The oscillator inverter and feedback resistor are turned off to eliminate power drain.

bit 2 **T1SYNC:** Timer1 External Clock Input Synchronization Select bit
When TMR1CS = 1:
 1 = Do not synchronize external clock input
 0 = Synchronize external clock input

When TMR1CS = 0:
 This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit
 1 = External clock from pin RC0/T1OSO/T13CKI (on the rising edge)
 0 = Internal clock (Fosc/4)

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

Note 1: Default (legacy) SFR at this address, available when WDTCON<4> = 0.

PIC18F87J11 FAMILY

13.1 Timer1 Operation

Timer1 can operate in one of these modes:

- Timer
- Synchronous Counter
- Asynchronous Counter

The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>). When TMR1CS is cleared (= 0), Timer1 increments on every internal instruction

cycle (Fosc/4). When the bit is set, Timer1 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled.

When Timer1 is enabled, the RC1/T1OSI and RC0/T1OSO/T13CKI pins become inputs. This means the values of TRISC<1:0> are ignored and the pins are read as '0'.

FIGURE 13-1: TIMER1 BLOCK DIAGRAM

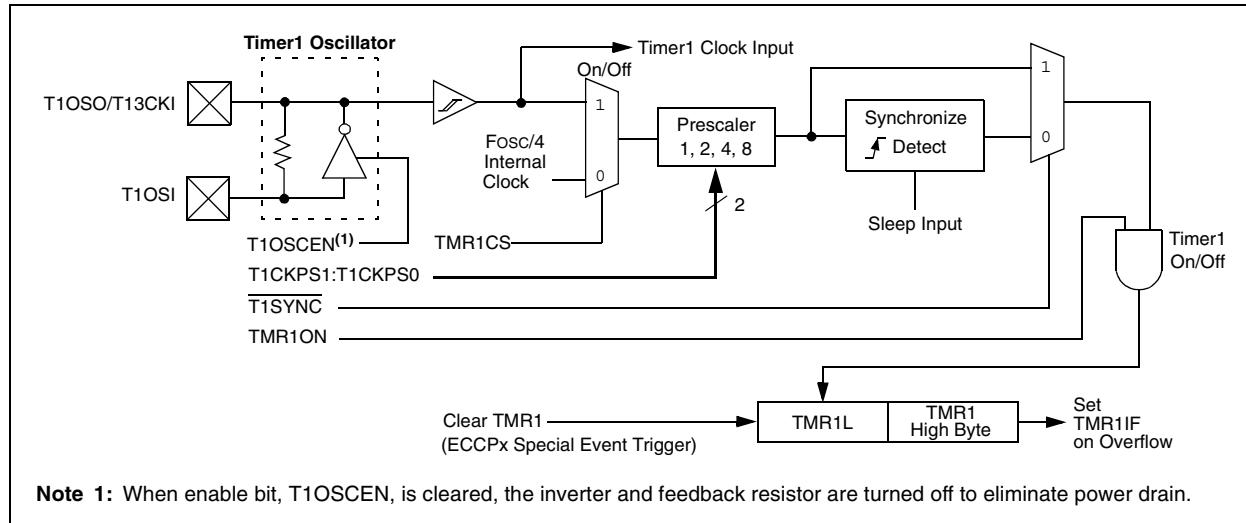
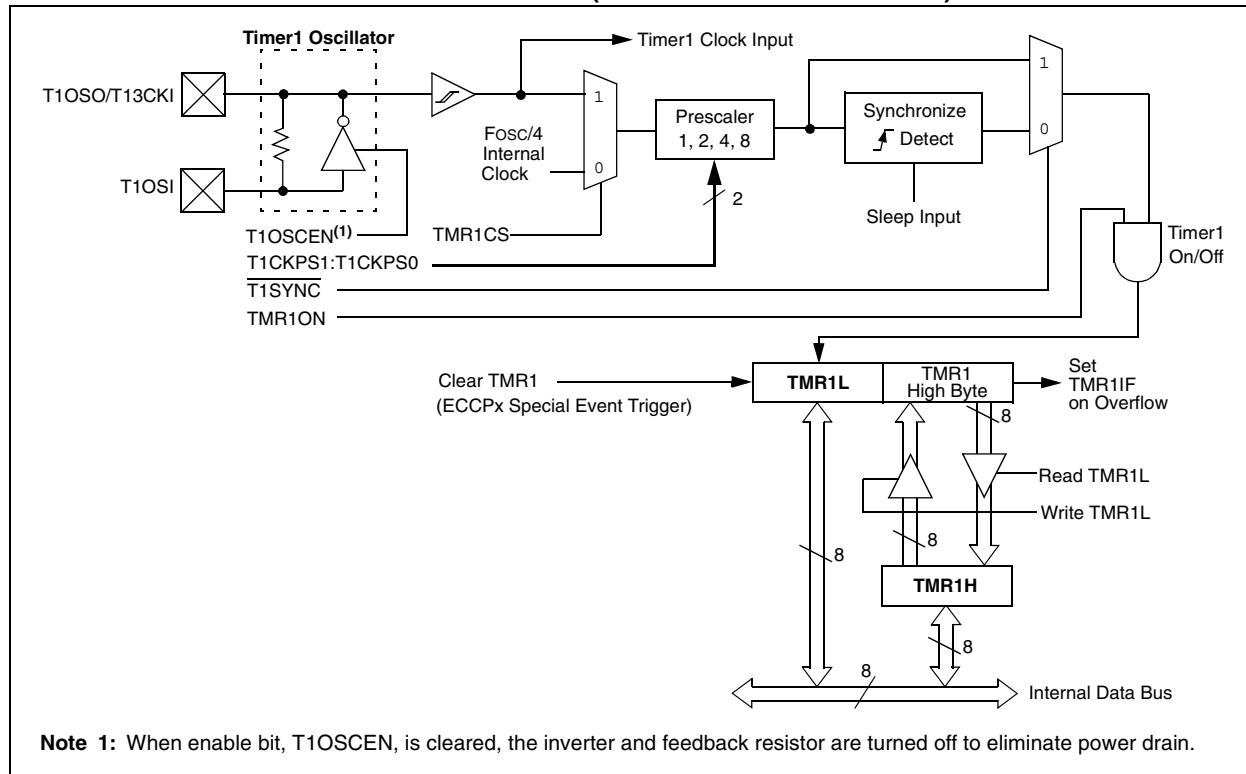


FIGURE 13-2: TIMER1 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)



13.2 Timer1 16-Bit Read/Write Mode

Timer1 can be configured for 16-bit reads and writes (see Figure 13-2). When the RD16 control bit, T1CON<7>, is set, the address for TMR1H is mapped to a buffer register for the high byte of Timer1. A read from TMR1L will load the contents of the high byte of Timer1 into the Timer1 High Byte Buffer register. This provides the user with the ability to accurately read all 16 bits of Timer1 without having to determine whether a read of the high byte, followed by a read of the low byte, has become invalid due to a rollover between reads.

A write to the high byte of Timer1 must also take place through the TMR1H Buffer register. The Timer1 high byte is updated with the contents of TMR1H when a write occurs to TMR1L. This allows a user to write all 16 bits to both the high and low bytes of Timer1 at once.

The high byte of Timer1 is not directly readable or writable in this mode. All reads and writes must take place through the Timer1 High Byte Buffer register. Writes to TMR1H do not clear the Timer1 prescaler. The prescaler is only cleared on writes to TMR1L.

13.3 Timer1 Oscillator

An on-chip crystal oscillator circuit is incorporated between pins T1OSI (input) and T1OSO (amplifier output). It is enabled by setting the Timer1 Oscillator Enable bit, T1OSCEN (T1CON<3>). The oscillator is a low-power circuit rated for 32 kHz crystals. It will continue to run during all power-managed modes. The circuit for a typical LP oscillator is shown in Figure 13-3. Table 13-1 shows the capacitor selection for the Timer1 oscillator.

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

FIGURE 13-3: EXTERNAL COMPONENTS FOR THE TIMER1 LP OSCILLATOR

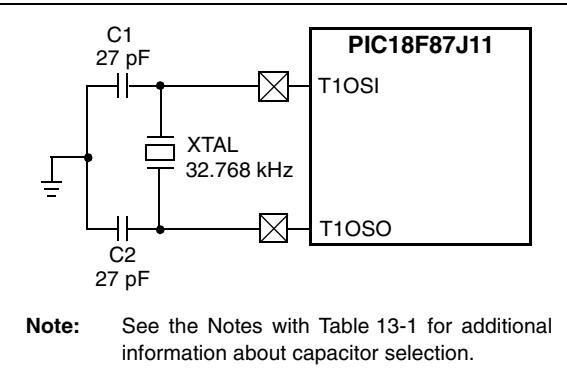


TABLE 13-1: CAPACITOR SELECTION FOR THE TIMER1 OSCILLATOR^(2,3,4)

Oscillator Type	Freq.	C1	C2
LP	32 kHz	27 pF ⁽¹⁾	27 pF ⁽¹⁾

Note 1: Microchip suggests these values as a starting point in validating the oscillator circuit.

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

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

4: Capacitor values are for design guidance only.

13.3.1 USING TIMER1 AS A CLOCK SOURCE

The Timer1 oscillator is also available as a clock source in power-managed modes. By setting the clock select bits, SCS1:SCS0 (OSCCON<1:0>), to '01', the device switches to SEC_RUN mode; both the CPU and peripherals are clocked from the Timer1 oscillator. If the IDLEN bit (OSCCON<7>) is cleared and a SLEEP instruction is executed, the device enters SEC_IDLE mode. Additional details are available in **Section 3.0 “Power-Managed Modes”**.

Whenever the Timer1 oscillator is providing the clock source, the Timer1 system clock status flag, T1RUN (T1CON<6>), is set. This can be used to determine the controller's current clocking mode. It can also indicate the clock source being currently used by the Fail-Safe Clock Monitor. If the Clock Monitor is enabled and the Timer1 oscillator fails while providing the clock, polling the T1RUN bit will indicate whether the clock is being provided by the Timer1 oscillator or another source.

13.3.2 TIMER1 OSCILLATOR LAYOUT CONSIDERATIONS

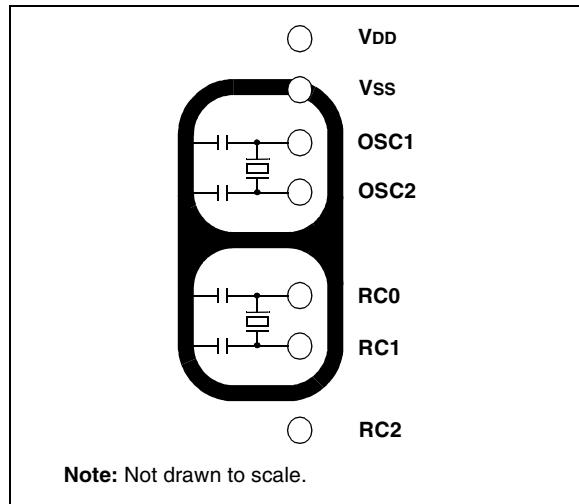
The Timer1 oscillator circuit draws very little power during operation. Due to the low-power nature of the oscillator, it may also be sensitive to rapidly changing signals in close proximity.

The oscillator circuit, shown in Figure 13-3, should be located as close as possible to the microcontroller. There should be no circuits passing within the oscillator circuit boundaries other than Vss or Vdd.

PIC18F87J11 FAMILY

If a high-speed circuit must be located near the oscillator (such as the ECCP1 pin in Output Compare or PWM mode, or the primary oscillator using the OSC2 pin), a grounded guard ring around the oscillator circuit, as shown in Figure 13-4, may be helpful when used on a single-sided PCB or in addition to a ground plane.

FIGURE 13-4: OSCILLATOR CIRCUIT WITH GROUNDED GUARD RING



13.4 Timer1 Interrupt

The TMR1 register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The Timer1 interrupt, if enabled, is generated on overflow which is latched in interrupt flag bit, TMR1IF (PIR1<0>). This interrupt can be enabled or disabled by setting or clearing the Timer1 Interrupt Enable bit, TMR1IE (PIE1<0>).

13.5 Resetting Timer1 Using the ECCPx Special Event Trigger

If ECCP1 or ECCP2 is configured to use Timer1 and to generate a Special Event Trigger in Compare mode (CCPxM3:CCPxM0 = 1011), this signal will reset Timer3. The trigger from ECCP2 will also start an A/D conversion if the A/D module is enabled (see **Section 18.2.1 “Special Event Trigger”** for more information).

The module must be configured as either a timer or a synchronous counter to take advantage of this feature. When used this way, the CCPRxH:CCPRxL register pair effectively becomes a period register for Timer1.

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, the write operation will take precedence.

Note: The Special Event Triggers from the ECCPx module will not set the TMR1IF interrupt flag bit (PIR1<0>).

13.6 Using Timer1 as a Real-Time Clock

Adding an external LP oscillator to Timer1 (such as the one described in **Section 13.3 “Timer1 Oscillator”**) gives users the option to include RTC functionality to their applications. This is accomplished with an inexpensive watch crystal to provide an accurate time base and several lines of application code to calculate the time. When operating in Sleep mode and using a battery or supercapacitor as a power source, it can completely eliminate the need for a separate RTC device and battery backup.

The application code routine, RTCisr, shown in Example 13-1, demonstrates a simple method to increment a counter at one-second intervals using an Interrupt Service Routine. Incrementing the TMR1 register pair to overflow triggers the interrupt and calls the routine which increments the seconds counter by one. Additional counters for minutes and hours are incremented as the previous counter overflows.

Since the register pair is 16 bits wide, counting up to overflow the register directly from a 32.768 kHz clock would take 2 seconds. To force the overflow at the required one-second intervals, it is necessary to preload it. The simplest method is to set the MSb of TMR1H with a BSF instruction. Note that the TMR1L register is never preloaded or altered; doing so may introduce cumulative error over many cycles.

For this method to be accurate, Timer1 must operate in Asynchronous mode and the Timer1 overflow interrupt must be enabled (PIE1<0> = 1), as shown in the routine, RTCinit. The Timer1 oscillator must also be enabled and running at all times.

13.7 Considerations in Asynchronous Counter Mode

Following a Timer1 interrupt and an update to the TMR1 registers, the Timer1 module uses a falling edge on its clock source to trigger the next register update on the rising edge. If the update is completed after the clock input has fallen, the next rising edge will not be counted.

If the application can reliably update TMR1 before the timer input goes low, no additional action is needed. Otherwise, an adjusted update can be performed following a later Timer1 increment. This can be done by monitoring TMR1L within the interrupt routine until it increments, and then updating the TMR1H:TMR1L register pair while the clock is low, or one-half of the period of the clock source. Assuming that Timer1 is being used as a Real-Time Clock, the clock source is a 32.768 kHz crystal oscillator. In this case, one-half period of the clock is 15.25 μ s.

The Real-Time Clock application code in Example 13-1 shows a typical ISR for Timer1, as well as the optional code required if the update cannot be done reliably within the required interval.

EXAMPLE 13-1: IMPLEMENTING A REAL-TIME CLOCK USING A TIMER1 INTERRUPT SERVICE

```

RTCinit
    MOVLW   80h          ; Preload TMR1 register pair
    MOVWF   TMR1H         ; for 1 second overflow
    CLRF    TMR1L
    MOVLW   b'00001111'   ; Configure for external clock,
    MOVWF   T1CON         ; Asynchronous operation, external oscillator
    CLRF    secs
    CLRF    mins
    MOVLW   .12
    MOVWF   hours
    BSF    PIE1, TMR1IE   ; Enable Timer1 interrupt
    RETURN

RTCisr
; Insert the next 4 lines of code when TMR1
; can not be reliably updated before clock pulse goes low
; wait for TMR1L to become clear
; (may already be clear)
; wait for TMR1L to become set
; TMR1 has just incremented
; If TMR1 update can be completed before clock pulse goes low
; Start ISR here
    BTFSC  TMR1L, 0
    BRA    $-2
    BTFSS  TMR1L, 0
    BRA    $-2
    BSF    TMR1H, 7       ; Preload for 1 sec overflow
    BCF    PIR1, TMR1IF   ; Clear interrupt flag
    INCF   secs, F        ; Increment seconds
    MOVLW  .59
    CPFSGT secs
    RETURN
    CLRF   secs
    INCF   mins, F        ; Increment minutes
    MOVLW  .59
    CPFSGT mins
    RETURN
    CLRF   mins
    INCF   hours, F        ; Increment hours
    MOVLW  .23
    CPFSGT hours
    RETURN
    CLRF   hours
    RETURN

```

TABLE 13-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	MPPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	MPPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	MPPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
TMR1L ⁽¹⁾	Timer1 Register Low Byte								56
TMR1H ⁽¹⁾	Timer1 Register High Byte								56
T1CON ⁽¹⁾	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	56

Legend: Shaded cells are not used by the Timer1 module.

Note 1: Default (legacy) SFR at this address, available when WDTCON<4> = 0.

PIC18F87J11 FAMILY

NOTES:

14.0 TIMER2 MODULE

The Timer2 module incorporates the following features:

- 8-Bit Timer and Period registers (TMR2 and PR2, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4 and 1:16)
- Software programmable postscale (1:1 through 1:16)
- Interrupt on TMR2 to PR2 match
- Optional use as the shift clock for the MSSP modules

The module is controlled through the T2CON register (Register 14-1) which enables or disables the timer and configures the prescaler and postscale. Timer2 can be shut off by clearing control bit, TMR2ON (T2CON<2>), to minimize power consumption.

A simplified block diagram of the module is shown in Figure 14-1.

14.1 Timer2 Operation

In normal operation, TMR2 is incremented from 00h on each clock (Fosc/4). A 4-bit counter/prescaler on the clock input gives direct input, divide-by-4 and divide-by-16 prescale options. These are selected by the prescaler control bits, T2CKPS1:T2CKPS0 (T2CON<1:0>). The value of TMR2 is compared to that of the Period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/postscale (see **Section 14.2 “Timer2 Interrupt”**).

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset, while the PR2 register initializes at FFh. Both the prescaler and postscale counters are cleared on the following events:

- a write to the TMR2 register
- a write to the T2CON register
- any device Reset (Power-on Reset, MCLR Reset, Watchdog Timer Reset or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

REGISTER 14-1: T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7	bit 0						

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

bit 7 **Unimplemented:** Read as ‘0’

bit 6-3 **T2OUTPS3:T2OUTPS0:** Timer2 Output Postscale Select bits

0000 = 1:1 Postscale

0001 = 1:2 Postscale

•

•

•

1111 = 1:16 Postscale

bit 2 **TMR2ON:** Timer2 On bit

1 = Timer2 is on

0 = Timer2 is off

bit 1-0 **T2CKPS1:T2CKPS0:** Timer2 Clock Prescale Select bits

00 = Prescaler is 1

01 = Prescaler is 4

1x = Prescaler is 16

PIC18F87J11 FAMILY

14.2 Timer2 Interrupt

Timer2 can also generate an optional device interrupt. The Timer2 output signal (TMR2 to PR2 match) provides the input for the 4-bit output counter/postscaler. This counter generates the TMR2 match interrupt flag which is latched in TMR2IF (PIR1<1>). The interrupt is enabled by setting the TMR2 Match Interrupt Enable bit, TMR2IE (PIE1<1>).

A range of 16 postscale options (from 1:1 through 1:16 inclusive) can be selected with the postscaler control bits, T2OUTPS3:T2OUTPS0 (T2CON<6:3>).

14.3 Timer2 Output

The unscaled output of TMR2 is available primarily to the ECCPx/CCPx modules, where it is used as a time base for operations in PWM mode.

Timer2 can be optionally used as the shift clock source for the MSSP modules operating in SPI mode. Additional information is provided in [Section 19.0 “Master Synchronous Serial Port \(MSSP\) Module”](#).

FIGURE 14-1: TIMER2 BLOCK DIAGRAM

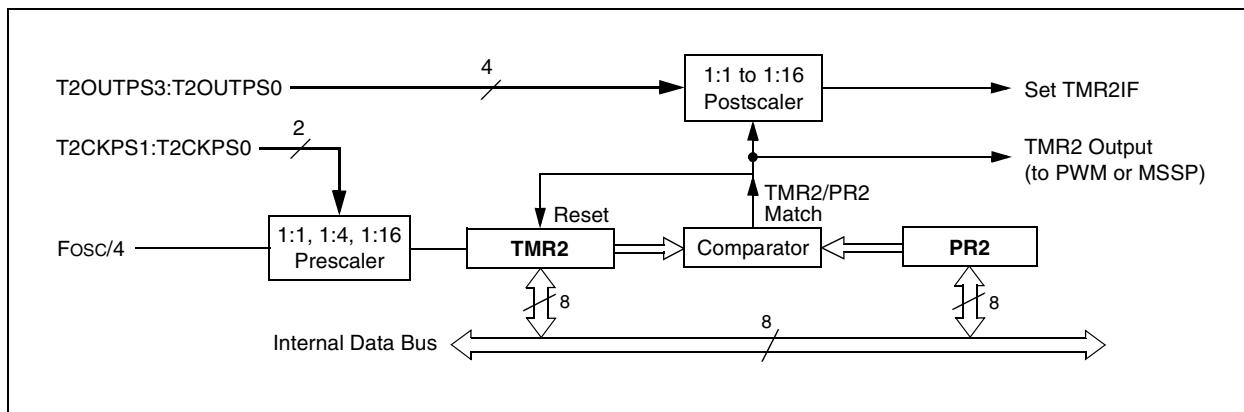


TABLE 14-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	MPPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	MPPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	MPPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
TMR2 ⁽¹⁾	Timer2 Register								56
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	56
PR2 ⁽¹⁾	Timer2 Period Register								56

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used by the Timer2 module.

Note 1: Default (legacy) SFR at this address, available when WDTCON<4> = 0.

15.0 TIMER3 MODULE

The Timer3 timer/counter module incorporates these features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR3H and TMR3L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt-on-overflow
- Module Reset on ECCPx Special Event Trigger

A simplified block diagram of the Timer3 module is shown in Figure 15-1. A block diagram of the module's operation in Read/Write mode is shown in Figure 15-2.

The Timer3 module is controlled through the T3CON register (Register 15-1). It also selects the clock source options for the CCP and ECCP modules; see **Section 17.1.1 “CCP Modules and Timer Resources”** for more information.

REGISTER 15-1: T3CON: TIMER3 CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON
bit 7	bit 0						

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

- bit 7 **RD16:** 16-Bit Read/Write Mode Enable bit
 1 = Enables register read/write of Timer3 in one 16-bit operation
 0 = Enables register read/write of Timer3 in two 8-bit operations
- bit 6,3 **T3CCP2:T3CCP1:** Timer3 and Timer1 to ECCPx/CCPx Enable bits
 11 = Timer3 and Timer4 are the clock sources for all ECCPx/CCPx modules
 10 = Timer3 and Timer4 are the clock sources for CCP3, CCP4 and CCP5;
 Timer1 and Timer2 are the clock sources for CCP1 and CCP2
 01 = Timer3 and Timer4 are the clock sources for CCP2, CCP3, CCP4 and CCP5;
 Timer1 and Timer2 are the clock sources for CCP1
 00 = Timer1 and Timer2 are the clock sources for all ECCPx/CCPx modules
- bit 5-4 **T3CKPS1:T3CKPS0:** Timer3 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 2 **T3SYNC:** Timer3 External Clock Input Synchronization Control bit
 (Not usable if the device clock comes from Timer1/Timer3.)
When TMR3CS = 1:
 1 = Do not synchronize external clock input
 0 = Synchronize external clock input
When TMR3CS = 0:
 This bit is ignored. Timer3 uses the internal clock when TMR3CS = 0.
- bit 1 **TMR3CS:** Timer3 Clock Source Select bit
 1 = External clock input from Timer1 oscillator or T13CKI (on the rising edge after the first falling edge)
 0 = Internal clock (Fosc/4)
- bit 0 **TMR3ON:** Timer3 On bit
 1 = Enables Timer3
 0 = Stops Timer3

PIC18F87J11 FAMILY

15.1 Timer3 Operation

Timer3 can operate in one of three modes:

- Timer
- Synchronous Counter
- Asynchronous Counter

The operating mode is determined by the clock select bit, TMR3CS (T3CON<1>). When TMR3CS is cleared (= 0), Timer3 increments on every internal instruction cycle (Fosc/4). When the bit is set, Timer3 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled.

As with Timer1, the RC1/T1OSI and RC0/T1OSO/T13CKI pins become inputs when the Timer1 oscillator is enabled. This means the values of TRISC<1:0> are ignored and the pins are read as '0'.

FIGURE 15-1: TIMER3 BLOCK DIAGRAM

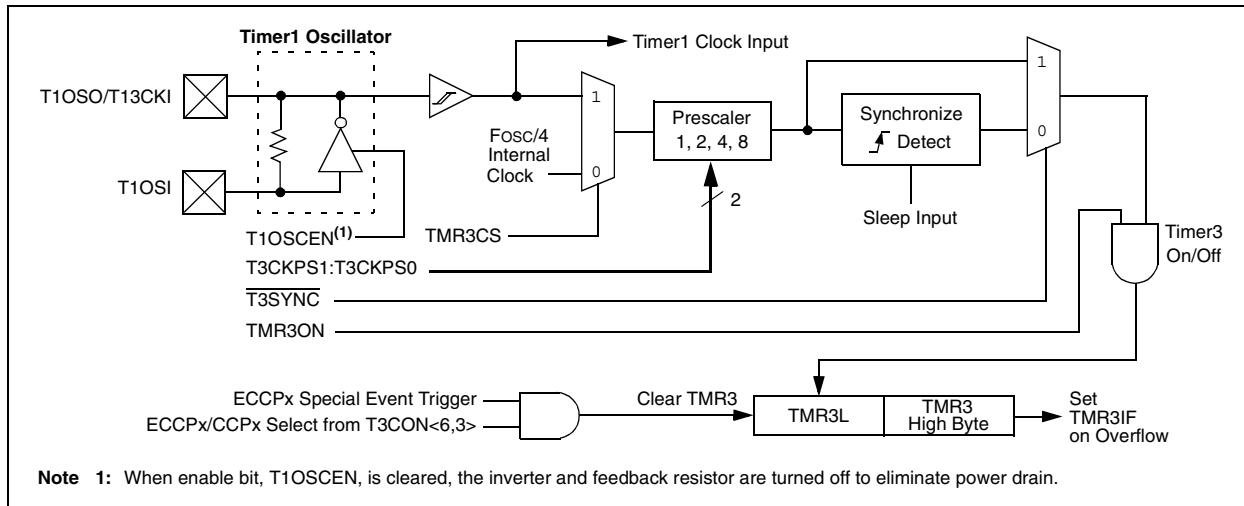
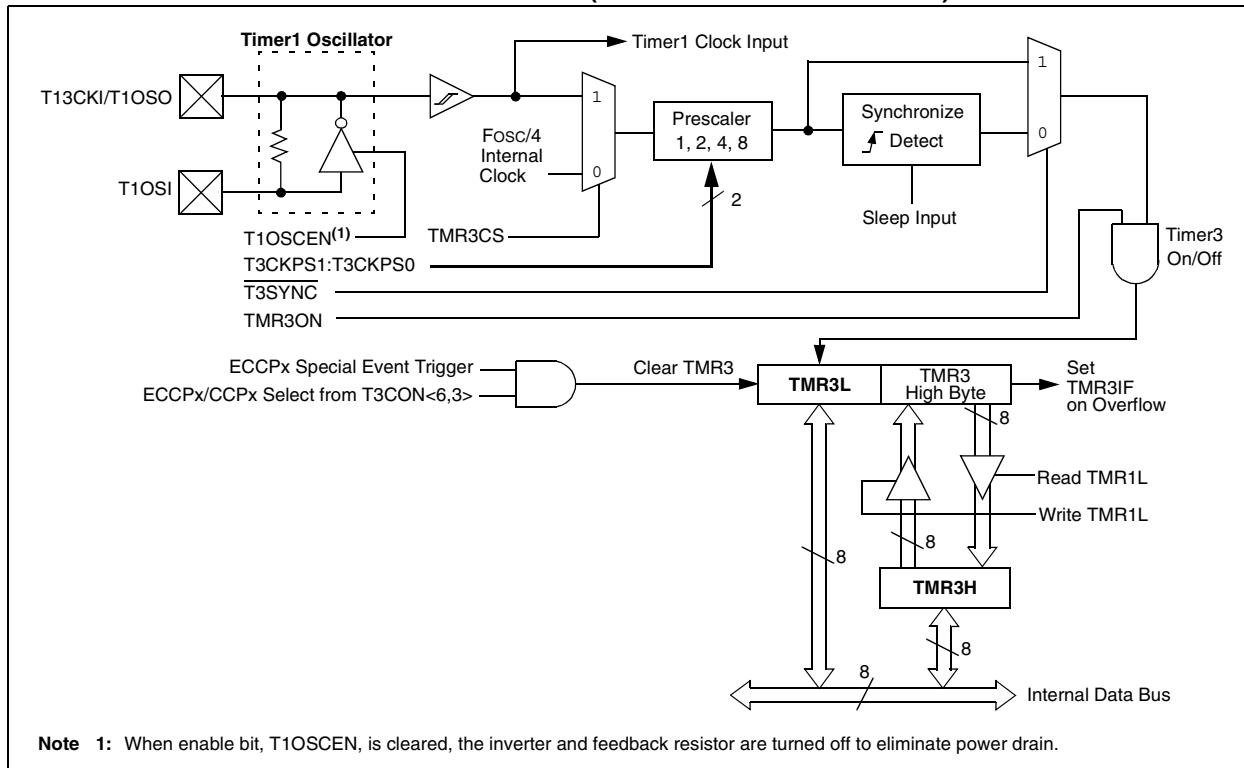


FIGURE 15-2: TIMER3 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)



15.2 Timer3 16-Bit Read/Write Mode

Timer3 can be configured for 16-bit reads and writes (see Figure 15-2). When the RD16 control bit (T3CON<7>) is set, the address for TMR3H is mapped to a buffer register for the high byte of Timer3. A read from TMR3L will load the contents of the high byte of Timer3 into the Timer3 High Byte Buffer register. This provides the user with the ability to accurately read all 16 bits of Timer1 without having to determine whether a read of the high byte, followed by a read of the low byte, has become invalid due to a rollover between reads.

A write to the high byte of Timer3 must also take place through the TMR3H Buffer register. The Timer3 high byte is updated with the contents of TMR3H when a write occurs to TMR3L. This allows a user to write all 16 bits to both the high and low bytes of Timer3 at once.

The high byte of Timer3 is not directly readable or writable in this mode. All reads and writes must take place through the Timer3 High Byte Buffer register.

Writes to TMR3H do not clear the Timer3 prescaler. The prescaler is only cleared on writes to TMR3L.

15.3 Using the Timer1 Oscillator as the Timer3 Clock Source

The Timer1 internal oscillator may be used as the clock source for Timer3. The Timer1 oscillator is enabled by setting the T1OSCEN (T1CON<3>) bit. To use it as the Timer3 clock source, the TMR3CS bit must also be set. As previously noted, this also configures Timer3 to increment on every rising edge of the oscillator source.

The Timer1 oscillator is described in **Section 13.0 “Timer1 Module”**.

15.4 Timer3 Interrupt

The TMR3 register pair (TMR3H:TMR3L) increments from 0000h to FFFFh and overflows to 0000h. The Timer3 interrupt, if enabled, is generated on overflow and is latched in interrupt flag bit, TMR3IF (PIR2<1>). This interrupt can be enabled or disabled by setting or clearing the Timer3 Interrupt Enable bit, TMR3IE (PIE2<1>).

15.5 Resetting Timer3 Using the ECCPx Special Event Trigger

If ECCP1 or ECCP2 is configured to use Timer3 and to generate a Special Event Trigger in Compare mode (CCPxM3:CCPxM0 = 1011), this signal will reset Timer3. The trigger from ECCP2 will also start an A/D conversion if the A/D module is enabled (see **Section 18.2.1 “Special Event Trigger”** for more information).

The module must be configured as either a timer or synchronous counter to take advantage of this feature. When used this way, the CCPRxH:CCPRxL register pair effectively becomes a period for Timer3.

If Timer3 is running in Asynchronous Counter mode, the Reset operation may not work.

In the event that a write to Timer3 coincides with a Special Event Trigger from an ECCPx module, the write will take precedence.

Note: The Special Event Triggers from the ECCPx module will not set the TMR3IF interrupt flag bit (PIR1<0>).

TABLE 15-1: REGISTERS ASSOCIATED WITH TIMER3 AS A TIMER/COUNTER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	58
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	58
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	58
TMR3L	Timer3 Register Low Byte								59
TMR3H	Timer3 Register High Byte								59
T1CON ⁽¹⁾	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	56
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON	59

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used by the Timer3 module.

Note 1: Default (legacy) SFR at this address, available when WDTCON<4> = 0.

PIC18F87J11 FAMILY

NOTES:

16.0 TIMER4 MODULE

The Timer4 timer module has the following features:

- 8-bit timer register (TMR4)
- 8-bit period register (PR4)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR4 match of PR4

Timer4 has a control register shown in Register 16-1. Timer4 can be shut off by clearing control bit, TMR4ON (T4CON<2>), to minimize power consumption. The prescaler and postscaler selection of Timer4 are also controlled by this register. Figure 16-1 is a simplified block diagram of the Timer4 module.

16.1 Timer4 Operation

Timer4 can be used as the PWM time base for the PWM mode of the ECCPx/CCPx modules. The TMR4 register is readable and writable and is cleared on any device Reset. The input clock (Fosc/4) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits T4CKPS1:T4CKPS0 (T4CON<1:0>). The match output of TMR4 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR4 interrupt, latched in flag bit, TMR4IF (PIR3<3>).

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

- a write to the TMR4 register
- a write to the T4CON register
- any device Reset (Power-on Reset, MCLR Reset, Watchdog Timer Reset or Brown-out Reset)

TMR4 is not cleared when T4CON is written.

REGISTER 16-1: T4CON: TIMER4 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0
bit 7	bit 0						

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

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

bit 6-3 **T4OUTPS3:T4OUTPS0:** Timer4 Output Postscale Select bits

0000 = 1:1 Postscale

0001 = 1:2 Postscale

•

•

•

1111 = 1:16 Postscale

bit 2 **TMR4ON:** Timer4 On bit

1 = Timer4 is on

0 = Timer4 is off

bit 1-0 **T4CKPS1:T4CKPS0:** Timer4 Clock Prescale Select bits

00 = Prescaler is 1

01 = Prescaler is 4

1x = Prescaler is 16

PIC18F87J11 FAMILY

16.2 Timer4 Interrupt

The Timer4 module has an 8-bit period register, PR4, which is both readable and writable. Timer4 increments from 00h until it matches PR4 and then resets to 00h on the next increment cycle. The PR4 register is initialized to FFh upon Reset.

16.3 Output of TMR4

The output of TMR4 (before the postscaler) is used only as a PWM time base for the ECCPx/CCPx modules. It is not used as a baud rate clock for the MSSP modules as is the Timer2 output.

FIGURE 16-1: TIMER4 BLOCK DIAGRAM

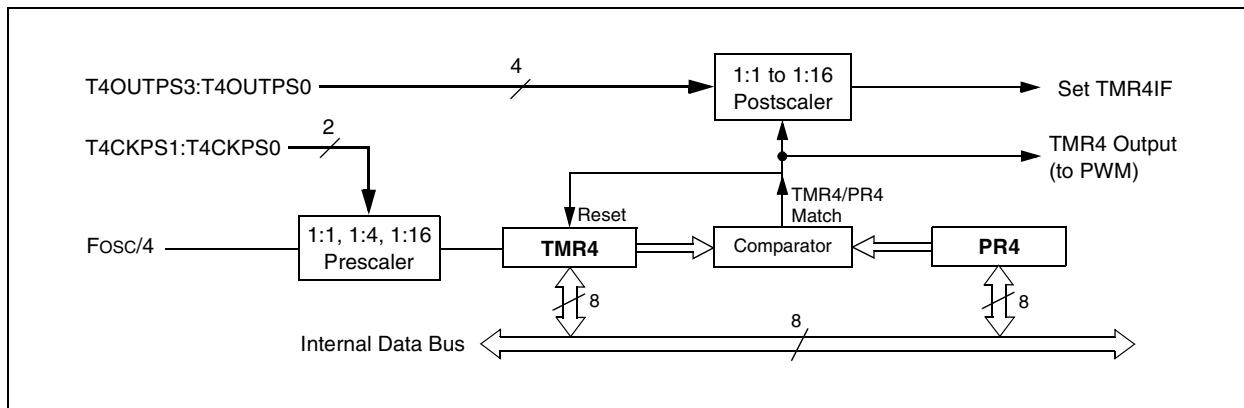


TABLE 16-1: REGISTERS ASSOCIATED WITH TIMER4 AS A TIMER/COUNTER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
TMR4	Timer4 Register								59
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	59
PR4	Timer4 Period Register								59

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the Timer4 module.

17.0 CAPTURE/COMPARE/PWM (CCP) MODULES

Members of the PIC18F87J11 family of devices all have a total of five CCP (Capture/Compare/PWM) modules. Two of these (CCP4 and CCP5) implement standard Capture, Compare and Pulse-Width Modulation (PWM) modes and are discussed in this section. The other three modules (ECCP1, ECCP2, ECCP3) implement standard Capture and Compare modes, as well as Enhanced PWM modes. These are discussed in **Section 18.0 “Enhanced Capture/Compare/PWM (ECCP) Module”**.

Each CCP/ECCP module contains a 16-bit register which can operate as a 16-bit Capture register, a 16-bit Compare register or a PWM Master/Slave Duty Cycle register. For the sake of clarity, all CCP module operation in the following sections is described with respect to CCP4, but is equally applicable to CCP5.

Capture and Compare operations described in this chapter apply to all standard and Enhanced CCP modules. The operations of PWM mode, described in **Section 17.4 “PWM Mode”**, apply to CCP4 and CCP5 only.

Note: Throughout this section and **Section 18.0 “Enhanced Capture/Compare/PWM (ECCP) Module”**, references to register and bit names that may be associated with a specific CCP module are referred to generically by the use of ‘x’ or ‘y’ in place of the specific module number. Thus, “CCPxCON” might refer to the control register for ECCP1, ECCP2, ECCP3, CCP4 or CCP5.

REGISTER 17-1: CCPxCON: CCPx CONTROL REGISTER (CCP4 MODULE, CCP5 MODULE)

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						

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

bit 7-6	Unimplemented: Read as ‘0’
bit 5-4	CCPx<X:Y>: PWM Duty Cycle bit 1 and bit 0 for CCPx Module Capture mode: Unused. Compare mode: Unused. PWM mode: These bits are the two Least Significant bits (bit 1 and bit 0) of the 10-bit PWM duty cycle. The eight Most Significant bits (DCx9:DCx2) of the duty cycle are found in CCPRxL.
bit 3-0	CCPxM3:CCPxM0: CCPx Module Mode Select bits 0000 = Capture/Compare/PWM disabled (resets CCPx module) 0001 = Reserved 0010 = Compare mode, toggle output on match (CCPxIF bit is set) 0011 = Reserved 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: initialize CCPx pin low; on compare match, force CCPx pin high (CCPxIF bit is set) 1001 = Compare mode: initialize CCPx pin high; on compare match, force CCPx pin low (CCPxIF bit is set) 1010 = Compare mode: generate software interrupt on compare match (CCPxIF bit is set, CCPx pin reflects I/O state) 1011 = Compare mode: trigger special event, reset timer, start A/D conversion on CCPx match (CCPxIF bit is set) ⁽¹⁾ 11xx = PWM mode

PIC18F87J11 FAMILY

17.1 CCP Module Configuration

Each Capture/Compare/PWM module is associated with a control register (generically, CCPxCON) and a data register (CCPRx). The data register, in turn, is comprised of two 8-bit registers: CCPRxL (low byte) and CCPRxH (high byte). All registers are both readable and writable.

17.1.1 CCP MODULES AND TIMER RESOURCES

The ECCP/CCP modules utilize Timers 1, 2, 3 or 4, depending on the mode selected. Timer1 and Timer3 are available to modules in Capture or Compare modes, while Timer2 and Timer4 are available for modules in PWM mode.

TABLE 17-1: CCP MODE – TIMER RESOURCE

CCP Mode	Timer Resource
Capture	Timer1 or Timer3
Compare	Timer1 or Timer3
PWM	Timer2 or Timer4

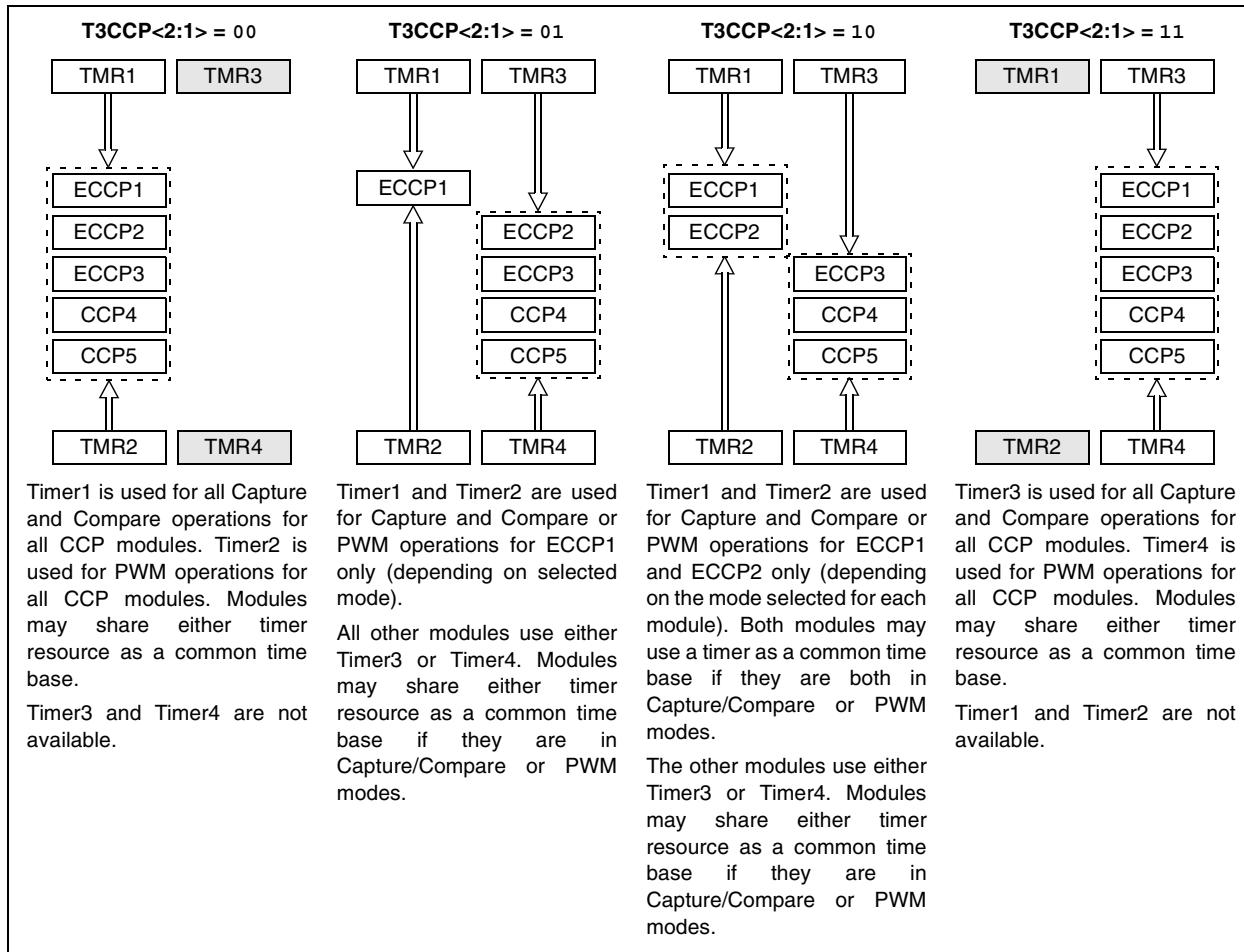
The assignment of a particular timer to a module is determined by the timer to CCP enable bits in the T3CON register (Register 15-1, page 193). Depending on the configuration selected, up to four timers may be active at once, with modules in the same configuration (Capture/Compare or PWM) sharing timer resources. The possible configurations are shown in Figure 17-1.

17.1.2 OPEN-DRAIN OUTPUT OPTION

When operating in Output mode (i.e., in Compare or PWM modes), the drivers for the CCP pins can be optionally configured as open-drain outputs. This feature allows the voltage level on the pin to be pulled to a higher level through an external pull-up resistor, and allows the output to communicate with external circuits without the need for additional level shifters. For more information, see **Section 10.1.4 “Open-Drain Outputs”**.

The open-drain output option is controlled by the bits in the ODCON1 register. Setting the appropriate bit configures the pin for the corresponding module for open-drain operation. The ODCON1 memory shares the same address space as TMR1H. The ODCON1 register can be accessed by setting the ADSHR bit in the WDTCON register (WDTCON<4>).

FIGURE 17-1: ECCPx/CCPx AND TIMER INTERCONNECT CONFIGURATIONS



Timer1 is used for all Capture and Compare operations for all CCP modules. Timer2 is used for PWM operations for all CCP modules. Modules may share either timer resource as a common time base.

Timer3 and Timer4 are not available.

Timer1 and Timer2 are used for Capture and Compare or PWM operations for ECCP1 only (depending on selected mode).

All other modules use either Timer3 or Timer4. Modules may share either timer resource as a common time base if they are in Capture/Compare or PWM modes.

Timer1 and Timer2 are used for Capture and Compare or PWM operations for ECCP1 and ECCP2 only (depending on the mode selected for each module). Both modules may use a timer as a common time base if they are both in Capture/Compare or PWM modes.

The other modules use either Timer3 or Timer4. Modules may share either timer resource as a common time base if they are in Capture/Compare or PWM modes.

Timer3 is used for all Capture and Compare operations for all CCP modules. Timer4 is used for PWM operations for all CCP modules. Modules may share either timer resource as a common time base.

Timer1 and Timer2 are not available.

17.2 Capture Mode

In Capture mode, the CCPRxH:CCPRxL register pair captures the 16-bit value of the TMR1 or TMR3 registers when an event occurs on the corresponding CCP pin. An event is defined as one of the following:

- every falling edge
- every rising edge
- every 4th rising edge
- every 16th rising edge

The event is selected by the mode select bits, CCPxM3:CCPxM0 (CCPxCON<3:0>). When a capture is made, the interrupt request flag bit, CCPxIF, is set; it must be cleared in software. If another capture occurs before the value in register CCPRx is read, the old captured value is overwritten by the new captured value.

17.2.1 CCP PIN CONFIGURATION

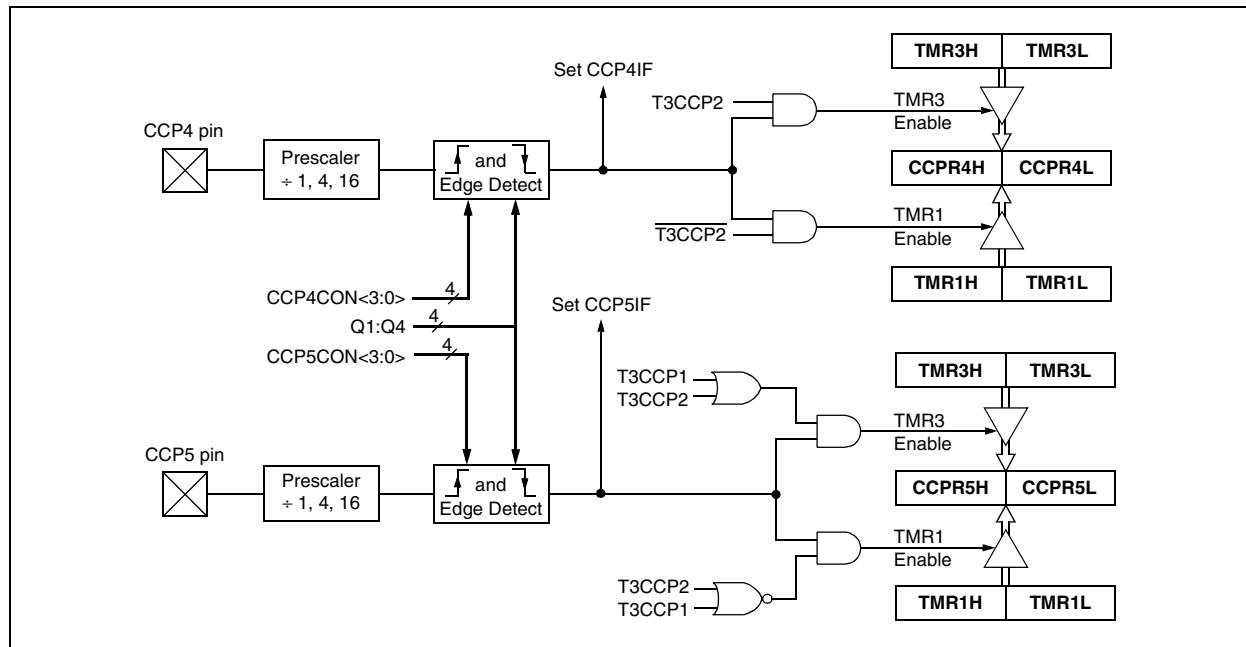
In Capture mode, the appropriate CCP pin should be configured as an input by setting the corresponding TRIS direction bit.

Note: If RG4/CCP5 is configured as an output, a write to the port can cause a capture condition.

17.2.2 TIMER1/TIMER3 MODE SELECTION

The timers that are to be used with the capture feature (Timer1 and/or Timer3) must be running in Timer mode or Synchronized Counter mode. In Asynchronous Counter mode, the capture operation will not work. The timer to be used with each CCP module is selected in the T3CON register (see **Section 17.1.1 “CCP Modules and Timer Resources”**).

FIGURE 17-2: CAPTURE MODE OPERATION BLOCK DIAGRAM



17.2.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE interrupt enable bit clear to avoid false interrupts. The interrupt flag bit, CCPxIF, should also be cleared following any such change in operating mode.

17.2.4 CCP PRESCALER

There are four prescaler settings in Capture mode. They are specified as part of the operating mode selected by the mode select bits (CCPxM3:CCPxM0). Whenever the CCP module is turned off or Capture mode is disabled, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared; therefore, the first capture may be from a non-zero prescaler. Example 17-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

EXAMPLE 17-1: CHANGING BETWEEN CAPTURE PRESCALERS (CCP5 SHOWN)

```

CLRF  CCP5CON      ; Turn CCP module off
MOVLW NEW_CAPT_PS ; Load WREG with the
                    ; new prescaler mode
                    ; value and CCP ON
MOVWF CCP5CON      ; Load CCP5CON with
                    ; this value

```

PIC18F87J11 FAMILY

17.3 Compare Mode

In Compare mode, the 16-bit CCPRx register value is constantly compared against either the TMR1 or TMR3 register pair value. When a match occurs, the CCP pin can be:

- driven high
- driven low
- toggled (high-to-low or low-to-high)
- remains unchanged (that is, reflects the state of the I/O latch)

The action on the pin is based on the value of the mode select bits (CCPxM3:CCPxM0). At the same time, the interrupt flag bit, CCPxIF, is set.

17.3.1 CCP PIN CONFIGURATION

The user must configure the CCP pin as an output by clearing the appropriate TRIS bit.

Note: Clearing the CCP5CON register will force the RG4 compare output latch (depending on device configuration) to the default low level. This is not the PORTB or PORTC I/O data latch.

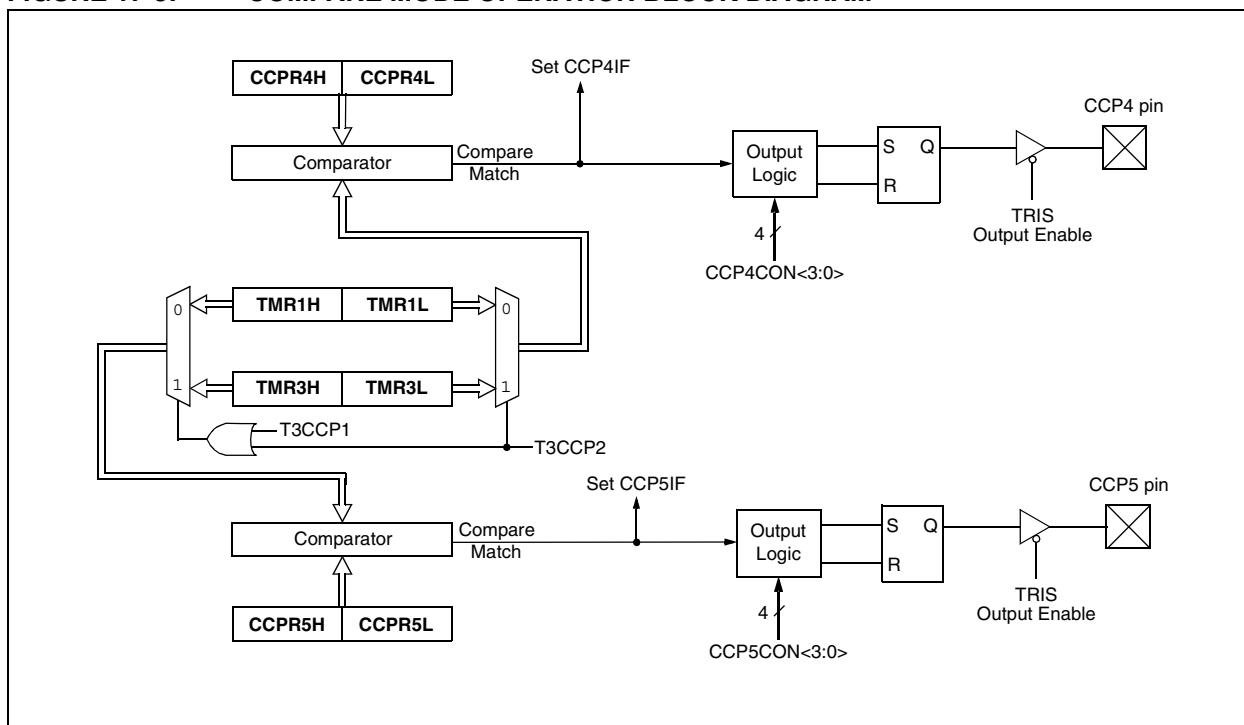
17.3.2 TIMER1/TIMER3 MODE SELECTION

Timer1 and/or Timer3 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.

17.3.3 SOFTWARE INTERRUPT MODE

When the Generate Software Interrupt mode is chosen (CCPxM3:CCPxM0 = 1010), the corresponding CCP pin is not affected. Only a CCP interrupt is generated, if enabled, and the CCPxE bit is set.

FIGURE 17-3: COMPARE MODE OPERATION BLOCK DIAGRAM



PIC18F87J11 FAMILY

TABLE 17-2: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, TIMER1 AND TIMER3

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
RCON	IPEN	—	CM	RI	TO	PD	POR	BOR	56
PIR1	MPPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	MPPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	MPPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	58
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	58
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	58
TMR1L ⁽¹⁾	Timer1 Register Low Byte								56
TMR1H ⁽¹⁾	Timer1 Register High Byte								56
ODCON1 ⁽²⁾	—	—	—	CCP5OD	CCP4OD	ECCP3OD	ECCP2OD	ECCP1OD	56
T1CON ⁽¹⁾	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	SYNC	TMR1CS	TMR1ON	56
TMR3H	Timer3 Register High Byte								59
TMR3L	Timer3 Register Low Byte								59
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	SYNC	TMR3CS	TMR3ON	59
CCPR4L	Capture/Compare/PWM Register 4 Low Byte								59
CCPR4H	Capture/Compare/PWM Register 4 High Byte								59
CCPR5L	Capture/Compare/PWM Register 5 Low Byte								59
CCPR5H	Capture/Compare/PWM Register 5 High Byte								59
CCP4CON	—	—	DC4B1	DC4B0	CCP4M3	CCP4M2	CCP4M1	CCP4M0	59
CCP5CON	—	—	DC5B1	DC5B0	CCP5M3	CCP5M2	CCP5M1	CCP5M0	59

Legend: — = unimplemented, read as '0'. Shaded cells are not used by Capture/Compare, Timer1 or Timer3.

Note 1: Default (legacy) SFR at this address, available when WDTCON<4> = 0.

2: Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.

PIC18F87J11 FAMILY

17.4 PWM Mode

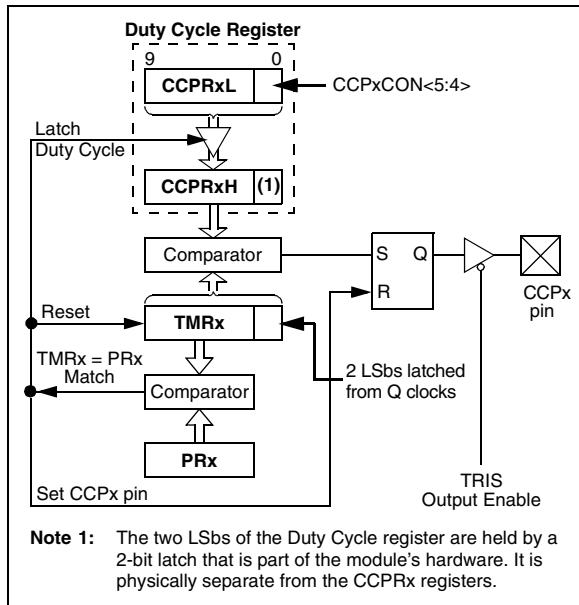
In Pulse-Width Modulation (PWM) mode, the CCP pin produces up to a 10-bit resolution PWM output. Since the CCP4 and CCP5 pins are multiplexed with a PORTG data latch, the appropriate TRISG bit must be cleared to make the CCP4 or CCP5 pin an output.

Note: Clearing the CCPxCON register will force the RG3 or RG4 output latch (depending on device configuration) to the default low level. This is not the PORTG I/O data latch.

Figure 17-4 shows a simplified block diagram of the CCP module in PWM mode.

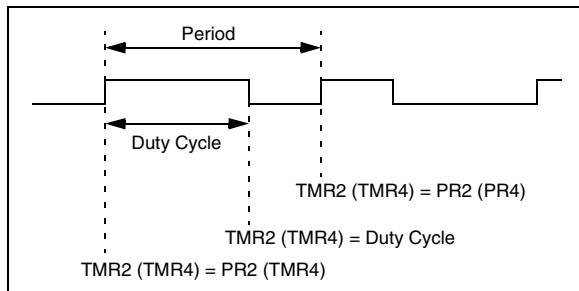
For a step-by-step procedure on how to set up a CCP module for PWM operation, see **Section 17.4.3 “Setup for PWM Operation”**.

FIGURE 17-4: SIMPLIFIED PWM BLOCK DIAGRAM



A PWM output (Figure 17-5) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

FIGURE 17-5: PWM OUTPUT



17.4.1 PWM PERIOD

The PWM period is specified by writing to the PR2 (PR4) register. The PWM period can be calculated using Equation 17-1:

EQUATION 17-1:

$$\text{PWM Period} = [(PR2 + 1) \cdot 4 \cdot TOSC \cdot (\text{TMR2 Prescale Value})]$$

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

When TMR2 (TMR4) is equal to PR2 (PR4), the following three events occur on the next increment cycle:

- TMR2 (TMR4) is cleared
- The CCP pin is set (exception: if PWM duty cycle = 0%, the CCP pin will not be set)
- The PWM duty cycle is latched from CCPRxL into CCPRxH

Note: The Timer2 and Timer 4 postscalers (see **Section 14.0 “Timer2 Module”** and **Section 16.0 “Timer4 Module”**) are not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

17.4.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPRxL register and to the CCPxCON<5:4> bits. Up to 10-bit resolution is available. The CCPRxL contains the eight MSbs and the CCPxCON<5:4> contains the two LSbs. This 10-bit value is represented by CCPRxL:CCPxCON<5:4>. Equation 17-2 is used to calculate the PWM duty cycle in time.

EQUATION 17-2:

$$\text{PWM Duty Cycle} = (CCPRxL:CCPxCON<5:4>) \cdot TOSC \cdot (\text{TMR2 Prescale Value})$$

CCPRxL and CCPxCON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPRxH until after a match between PR2 (PR4) and TMR2 (TMR4) occurs (i.e., the period is complete). In PWM mode, CCPRxH is a read-only register.

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

When the CCPRxH and 2-bit latch match TMR2 (TMR4), concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 (TMR4) prescaler, the CCP pin is cleared.

The maximum PWM resolution (bits) for a given PWM frequency is given by Equation 17-3:

EQUATION 17-3:

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

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

17.4.3 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 (PR4) register.
2. Set the PWM duty cycle by writing to the CCPRxL register and CCPxCON<5:4> bits.
3. Make the CCP pin an output by clearing the appropriate TRIS bit.
4. Set the TMR2 (TMR4) prescale value, then enable Timer2 (Timer4) by writing to T2CON (T4CON).
5. Configure the CCP module for PWM operation.

TABLE 17-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz

PWM Frequency	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	FFh	FFh	FFh	3Fh	1Fh	17h
Maximum Resolution (bits)	10	10	10	8	7	6.58

PIC18F87J11 FAMILY

TABLE 17-4: REGISTERS ASSOCIATED WITH PWM, TIMER2 AND TIMER4

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMROIE	INT0IE	RBIE	TMROIF	INT0IF	RBIF	55
RCON	IPEN	—	CM	RI	TO	PD	POR	BOR	56
PIR1	MPPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	MPPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	MPPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	58
TMR2 ⁽¹⁾	Timer2 Register								56
PR2 ⁽¹⁾	Timer2 Period Register								56
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	56
TMR4	Timer4 Register								59
PR4	Timer4 Period Register								59
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	59
CCPR4L	Capture/Compare/PWM Register 4 Low Byte								59
CCPR4H	Capture/Compare/PWM Register 4 High Byte								59
CCPR5L	Capture/Compare/PWM Register 5 Low Byte								59
CCPR5H	Capture/Compare/PWM Register 5 High Byte								59
CCP4CON	—	—	DC4B1	DC4B0	CCP4M3	CCP4M2	CCP4M1	CCP4M0	59
CCP5CON	—	—	DC5B1	DC5B0	CCP5M3	CCP5M2	CCP5M1	CCP5M0	59
ODCON1 ⁽²⁾	—	—	—	CCP5OD	CCP4OD	ECCP3OD	ECCP2OD	ECCP1OD	56

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PWM, Timer2 or Timer4.

Note 1: Default (legacy) SFR at this address, available when WDTCON<4> = 0.

2: Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.

18.0 ENHANCED CAPTURE/COMPARE/PWM (ECCP) MODULE

In the PIC18F87J11 family of devices, three of the CCP modules are implemented as standard CCP modules with Enhanced PWM capabilities. These include the provision for 2 or 4 output channels, user-selectable polarity, dead-band control and automatic shutdown and restart. The Enhanced features are discussed in detail in **Section 18.4 “Enhanced PWM Mode”**. Capture, Compare and single-output PWM functions of the ECCP module are the same as described for the standard CCP module.

The control register for the Enhanced CCP module is shown in Register 18-1. It differs from the CCP4CON/CCP5CON registers in that the two Most Significant bits are implemented to control PWM functionality.

In addition to the expanded range of modes available through the Enhanced CCPxCON register, the ECCP modules each have two additional registers associated with Enhanced PWM operation and auto-shutdown features. They are:

- ECCPxDEL (ECCPx PWM Delay)
- ECCPxAS (ECCPx Auto-Shutdown Control)

REGISTER 18-1: CCPxCON: ECCPx CONTROL REGISTER (ECCP1/ECCP2/ECCP3)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0

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

bit 7-6	PxM1:PxM0: Enhanced PWM Output Configuration bits <u>If CCPxM3:CCPxM2 = 00, 01, 10:</u> xx = PxA assigned as Capture/Compare input/output; Px B, Px C, Px D assigned as port pins <u>If CCPxM3:CCPxM2 = 11:</u> 00 = Single output: Px A modulated; Px B, Px C, Px D assigned as port pins 01 = Full-bridge output forward: P1D modulated; P1A active; P1B, P1C inactive 10 = Half-bridge output: P1A, P1B modulated with dead-band control; P1C, P1D assigned as port pins 11 = Full-bridge output reverse: P1B modulated; P1C active; P1A, P1D inactive
bit 5-4	DCxB1:DCxB0: PWM Duty Cycle bit 1 and bit 0 <u>Capture mode:</u> Unused. <u>Compare mode:</u> Unused. <u>PWM mode:</u> These bits are the two LSbs of the 10-bit PWM duty cycle. The eight MSbs of the duty cycle are found in CCPRxL.
bit 3-0	CCPxM3:CCPxM0: Enhanced CCPx Module Mode Select bits 0000 = Capture/Compare/PWM off (resets ECCPx module) 0001 = Reserved 0010 = Compare mode, toggle output on match 0011 = Capture mode 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: initialize ECCPx pin low; set output on compare match (set CCPxIF) 1001 = Compare mode: initialize ECCPx pin high; clear output on compare match (set CCPxIF) 1010 = Compare mode: generate software interrupt only; ECCPx pin reverts to I/O state 1011 = Compare mode: trigger special event (ECCPx resets TMR1 or TMR3, sets CCPxIF bit, ECCPx trigger also starts A/D conversion if A/D module is enabled) ⁽¹⁾ 1100 = PWM mode: Px A, Px C active-high; Px B, Px D active-high 1101 = PWM mode: Px A, Px C active-high; Px B, Px D active-low 1110 = PWM mode: Px A, Px C active-low; Px B, Px D active-high 1111 = PWM mode: Px A, Px C active-low; Px B, Px D active-low

Note 1: Implemented only for ECCP1 and ECCP2; same as ‘1010’ for ECCP3.

PIC18F87J11 FAMILY

18.1 ECCP Outputs and Configuration

Each of the Enhanced CCP modules may have up to four PWM outputs, depending on the selected operating mode. These outputs, designated Px_A through Px_D, are multiplexed with various I/O pins. Some ECCP pin assignments are constant, while others change based on device configuration. For those pins that do change, the controlling bits are:

- CCP2MX Configuration bit
- ECCPMX Configuration bit (80-pin devices only)
- Program Memory Operating mode, set by the EMB Configuration bits (80-pin devices only)

The pin assignments for the Enhanced CCP modules are summarized in Table 18-1, Table 18-2 and Table 18-3. To configure the I/O pins as PWM outputs, the proper PWM mode must be selected by setting the PxM_x and CCPxM_x bits (CCPxCON<7:6> and <3:0>, respectively). The appropriate TRIS direction bits for the corresponding port pins must also be set as outputs.

18.1.1 ECCP1/ECCP3 OUTPUTS AND PROGRAM MEMORY MODE

In 80-pin devices, the use of Extended Microcontroller mode has an indirect effect on the use of ECCP1 and ECCP3 in Enhanced PWM modes. By default, PWM outputs, P1B/P1C and P3B/P3C, are multiplexed to PORTE pins along with the high-order byte of the external memory bus. When the bus is active in Extended Microcontroller mode, it overrides the Enhanced CCP outputs and makes them unavailable. Because of this, ECCP1 and ECCP3 can only be used in compatible (single output) PWM modes when the device is in Extended Microcontroller mode and default pin configuration.

An exception to this configuration is when a 12-bit address width is selected for the external bus (EMB1:EMB0 Configuration bits = 01). In this case, the upper pins of PORTE continue to operate as digital I/O, even when the external bus is active. P1B/P1C and P3B/P3C remain available for use as Enhanced PWM outputs.

If an application requires the use of additional PWM outputs during enhanced microcontroller operation, the P1B/P1C and P3B/P3C outputs can be reassigned to the upper bits of PORTH. This is done by clearing the ECCPMX Configuration bit.

18.1.2 ECCP2 OUTPUTS AND PROGRAM MEMORY MODES

For 80-pin devices, the program memory mode of the device (**Section 5.1.3 “PIC18F8xJ11/8XJ16 Program Memory Modes”**) also impacts pin multiplexing for the module.

The ECCP2 input/output (ECCP2/P2A) can be multiplexed to one of three pins. The default assignment (CCP2MX Configuration bit is set) for all devices is RC1. Clearing CCP2MX reassigns ECCP2/P2A to RE7.

An additional option exists for 80-pin devices. When these devices are operating in Microcontroller mode, the multiplexing options described above still apply. In Extended Microcontroller mode, clearing CCP2MX reassigns ECCP2/P2A to RB3.

Changing the pin assignment of ECCP2 does not automatically change any requirements for configuring the port pin. Users must always verify that the appropriate TRIS register is configured correctly for ECCP2 operation regardless of where it is located.

18.1.3 USE OF CCP4 AND CCP5 WITH ECCP1 AND ECCP3

Only the ECCP2 module has four dedicated output pins that are available for use. Assuming that the I/O ports or other multiplexed functions on those pins are not needed, they may be used whenever needed without interfering with any other CCP module.

ECCP1 and ECCP3, on the other hand, only have three dedicated output pins: ECCPx/PxA, Px_B and Px_C. Whenever these modules are configured for Quad PWM mode, the pin normally used for CCP4 or CCP5 becomes the Px_D output pins for ECCP3 and ECCP1, respectively. The CCP4 and CCP5 modules remain functional but their outputs are overridden.

18.1.4 ECCP MODULES AND TIMER RESOURCES

Like the standard CCP modules, the ECCP modules can utilize Timers 1, 2, 3 or 4, depending on the mode selected. Timer1 and Timer3 are available for modules in Capture or Compare modes, while Timer2 and Timer4 are available for modules in PWM mode. Additional details on timer resources are provided in **Section 17.1.1 “CCP Modules and Timer Resources”**.

18.1.5 OPEN-DRAIN OUTPUT OPTION

When operating in compare or standard PWM modes, the drivers for the ECCP pins can be optionally configured as open-drain outputs. This feature allows the voltage level on the pin to be pulled to a higher level through an external pull-up resistor, and allows the output to communicate with external circuits without the need for additional level shifters. For more information, see **Section 10.1.4 “Open-Drain Outputs”**

The open-drain output option is controlled by the bits in the ODCON1 register. Setting the appropriate bit configures the pin for the corresponding module for open-drain operation. The ODCON1 memory shares the same address space as of TMR1H. The ODCON1 register can be accessed by setting the ADSHR bit in the WDTCON register (WDTCON<4>).

PIC18F87J11 FAMILY

TABLE 18-1: PIN CONFIGURATIONS FOR ECCP1

ECCP Mode	CCP1CON Configuration	RC2	RE6	RE5	RG4	RH7	RH6
All PIC18F6XJ1X Devices:							
Compatible CCP	00xx 11xx	ECCP1	RE6	RE5	RG4/CCP5	N/A	N/A
Dual PWM	10xx 11xx	P1A	P1B	RE5	RG4/CCP5	N/A	N/A
Quad PWM ⁽¹⁾	x1xx 11xx	P1A	P1B	P1C	P1D	N/A	N/A
PIC18F8XJ1X Devices, ECCPMX = 0, Microcontroller mode:							
Compatible CCP	00xx 11xx	ECCP1	RE6/AD14	RE5/AD13	RG4/CCP5	RH7/AN15	RH6/AN14
Dual PWM	10xx 11xx	P1A	RE6/AD14	RE5/AD13	RG4/CCP5	P1B	RH6/AN14
Quad PWM ⁽¹⁾	x1xx 11xx	P1A	RE6/AD14	RE5/AD13	P1D	P1B	P1C
PIC18F8XJ1X Devices, ECCPMX = 1, Extended Microcontroller mode, 16-Bit or 20-Bit Address Width:							
Compatible CCP	00xx 11xx	ECCP1	RE6/AD14	RE5/AD13	RG4/CCP5	RH7/AN15	RH6/AN14
PIC18F8XJ1X Devices, ECCPMX = 1, Microcontroller mode or Extended Microcontroller mode, 12-Bit Address Width:							
Compatible CCP	00xx 11xx	ECCP1	RE6/AD14	RE5/AD13	RG4/CCP5	RH7/AN15	RH6/AN14
Dual PWM	10xx 11xx	P1A	P1B	RE5/AD13	RG4/CCP5	RH7/AN15	RH6/AN14
Quad PWM ⁽¹⁾	x1xx 11xx	P1A	P1B	P1C	P1D	RH7/AN15	RH6/AN14

Legend: x = Don't care, N/A = Not Available. Shaded cells indicate pin assignments not used by ECCP1 in a given mode.

Note 1: With ECCP1 in Quad PWM mode, CCP5's output is overridden by P1D; otherwise, CCP5 is fully operational.

TABLE 18-2: PIN CONFIGURATIONS FOR ECCP2

ECCP Mode	CCP2CON Configuration	RB3	RC1	RE7	RE2	RE1	RE0
All Devices, CCP2MX = 1, Either Operating mode:							
Compatible CCP	00xx 11xx	RB3/INT3	ECCP2	RE7	RE2	RE1	RE0
Dual PWM	10xx 11xx	RB3/INT3	P2A	RE7	P2B	RE1	RE0
Quad PWM	x1xx 11xx	RB3/INT3	P2A	RE7	P2B	P2C	P2D
All Devices, CCP2MX = 0, Microcontroller mode:							
Compatible CCP	00xx 11xx	RB3/INT3	RC1/T1OS1	ECCP2	RE2	RE1	RE0
Dual PWM	10xx 11xx	RB3/INT3	RC1/T1OS1	P2A	P2B	RE1	RE0
Quad PWM	x1xx 11xx	RB3/INT3	RC1/T1OS1	P2A	P2B	P2C	P2D
PIC18F8XJ1X Devices, CCP2MX = 0, Extended Microcontroller mode:							
Compatible CCP	00xx 11xx	ECCP2	RC1/T1OS1	RE7/AD15	RE2/CS	RE1/W RD	RE0/R RD
Dual PWM	10xx 11xx	P2A	RC1/T1OS1	RE7/AD15	P2B	RE1/W RD	RE0/R RD
Quad PWM	x1xx 11xx	P2A	RC1/T1OS1	RE7/AD15	P2B	P2C	P2D

Legend: x = Don't care. Shaded cells indicate pin assignments not used by ECCP2 in a given mode.

PIC18F87J11 FAMILY

TABLE 18-3: PIN CONFIGURATIONS FOR ECCP3

ECCP Mode	CCP3CON Configuration	RG0	RE4	RE3	RG3	RH5	RH4
PIC18F6XJ1X Devices:							
Compatible CCP	00xx 11xx	ECCP3	RE4	RE3	RG3/CCP4	N/A	N/A
Dual PWM	10xx 11xx	P3A	P3B	RE3	RG3/CCP4	N/A	N/A
Quad PWM ⁽¹⁾	x1xx 11xx	P3A	P3B	P3C	P3D	N/A	N/A
PIC18F8XJ1X Devices, ECCPMX = 0, Microcontroller mode:							
Compatible CCP	00xx 11xx	ECCP3	RE6/AD14	RE5/AD13	RG3/CCP4	RH7/AN15	RH6/AN14
Dual PWM	10xx 11xx	P3A	RE6/AD14	RE5/AD13	RG3/CCP4	P3B	RH6/AN14
Quad PWM ⁽¹⁾	x1xx 11xx	P3A	RE6/AD14	RE5/AD13	P3D	P3B	P3C
PIC18F8XJ1X Devices, ECCPMX = 1, Extended Microcontroller mode, 16-Bit or 20-Bit Address Width:							
Compatible CCP	00xx 11xx	ECCP3	RE6/AD14	RE5/AD13	RG3/CCP4	RH7/AN15	RH6/AN14
PIC18F8XJ1X Devices, ECCPMX = 1, Microcontroller mode or Extended Microcontroller mode, 12-Bit Address Width:							
Compatible CCP	00xx 11xx	ECCP3	RE4/AD12	RE3/AD11	RG3/CCP4	RH5/AN13	RH4/AN12
Dual PWM	10xx 11xx	P3A	P3B	RE3/AD11	RG3/CCP4	RH5/AN13	RH4/AN12
Quad PWM ⁽¹⁾	x1xx 11xx	P3A	P3B	P3C	P3D	RH5/AN13	RH4/AN12

Legend: x = Don't care, N/A = Not Available. Shaded cells indicate pin assignments not used by ECCP3 in a given mode.

Note 1: With ECCP3 in Quad PWM mode, CCP4's output is overridden by P1D; otherwise, CCP4 is fully operational.

18.2 Capture and Compare Modes

Except for the operation of the Special Event Trigger discussed below, the Capture and Compare modes of the ECCP module are identical in operation to that of CCP4. These are discussed in detail in **Section 17.2 “Capture Mode”** and **Section 17.3 “Compare Mode”**.

18.2.1 SPECIAL EVENT TRIGGER

ECCP1 and ECCP2 incorporate an internal hardware trigger that is generated in Compare mode on a match between the CCPRx register pair and the selected timer. This can be used in turn to initiate an action. This mode is selected by setting CCPxCON<3:0> to '1011'.

The Special Event Trigger output of either ECCP1 or ECCP2 resets the TMR1 or TMR3 register pair, depending on which timer resource is currently selected. This allows the CCPRx register pair to effectively be a 16-bit programmable period register for Timer1 or Timer3. In addition, the ECCP2 Special Event Trigger will also start an A/D conversion if the A/D module is enabled.

Special Event Triggers are not implemented for ECCP3, CCP4 or CCP5. Selecting the Special Event Trigger mode for these modules has the same effect as selecting the Compare with Software Interrupt mode (CCPxM3:CCPxM0 = 1010).

Note: The Special Event Trigger from ECCP2 will not set the Timer1 or Timer3 interrupt flag bits.

18.3 Standard PWM Mode

When configured in Single Output mode, the ECCP module functions identically to the standard CCP module in PWM mode, as described in **Section 17.4 “PWM Mode”**. This is also sometimes referred to as “Compatible CCP” mode as in Tables 18-1 through 18-3.

Note: When setting up single output PWM operations, users are free to use either of the processes described in **Section 17.4.3 “Setup for PWM Operation”** or **Section 18.4.9 “Setup for PWM Operation”**. The latter is more generic but will work for either single or multi-output PWM.

18.4 Enhanced PWM Mode

The Enhanced PWM mode provides additional PWM output options for a broader range of control applications. The module is a backward compatible version of the standard CCP module and offers up to four outputs, designated PxA through PxD. Users are also able to select the polarity of the signal (either active-high or active-low). The module's output mode and polarity are configured by setting the PxM1:PxM0 and CCPxM3:CCPxM0 bits of the CCPxCON register (CCPxCON<7:6> and CCPxCON<3:0>, respectively).

For the sake of clarity, Enhanced PWM mode operation is described generically throughout this section with respect to the ECCP1 and TMR2 modules. Control register names are presented in terms of ECCP1. All three Enhanced modules, as well as the two timer resources, can be used interchangeably and function identically. TMR2 or TMR4 can be selected for PWM operation by selecting the proper bits in T3CON.

Figure 18-1 shows a simplified block diagram of PWM operation. All control registers are double-buffered and are loaded at the beginning of a new PWM cycle (the period boundary when Timer2 resets) in order to prevent glitches on any of the outputs. The exception is the ECCPx PWM Delay register, ECCPxDEL, which is loaded at either the duty cycle boundary or the boundary period (whichever comes first). Because of the buffering, the module waits until the assigned timer resets instead of starting immediately. This means that

Enhanced PWM waveforms do not exactly match the standard PWM waveforms, but are instead offset by one full instruction cycle (4 Tosc).

As before, the user must manually configure the appropriate TRIS bits for output.

18.4.1 PWM PERIOD

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

EQUATION 18-1:

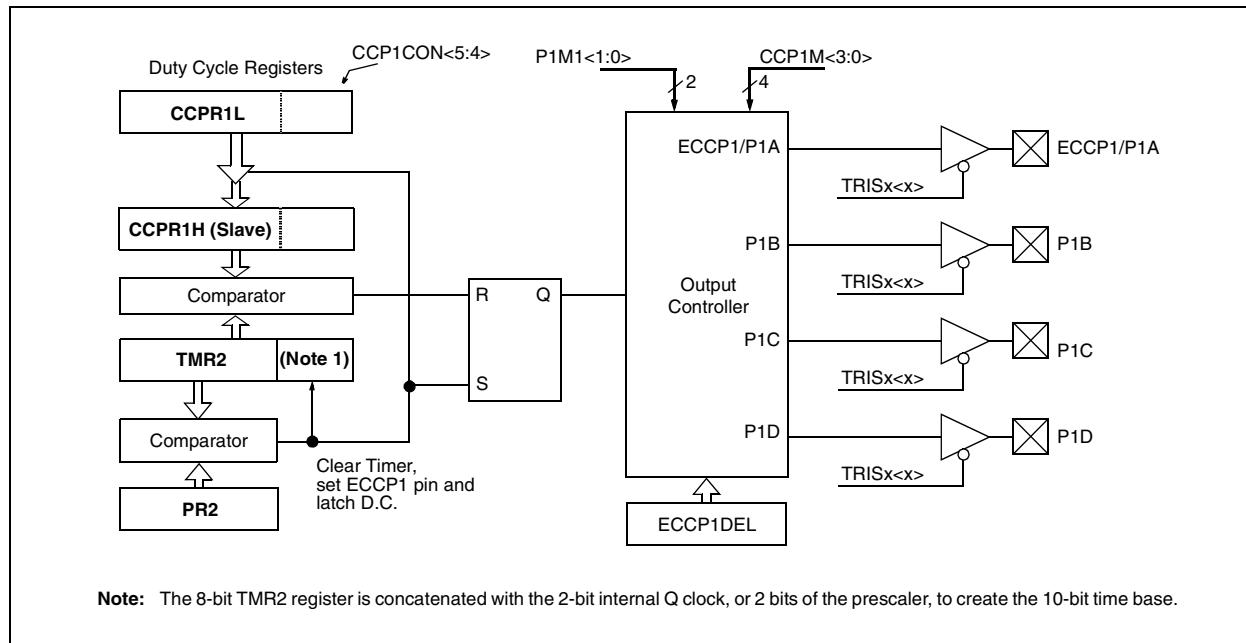
$$\text{PWM Period} = [(PR2 + 1) \cdot 4 \cdot TOSC \cdot (TMR2 \text{ Prescale Value})]$$

PWM frequency is defined as $1/\text{[PWM period]}$. When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The ECCP1 pin is set (if PWM duty cycle = 0%, the ECCP1 pin will not be set)
- The PWM duty cycle is copied from CCPR1L into CCPR1H

Note: The Timer2 postscaler (see **Section 14.0 “Timer2 Module”**) is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

FIGURE 18-1: SIMPLIFIED BLOCK DIAGRAM OF THE ENHANCED PWM MODULE



PIC18F87J11 FAMILY

18.4.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available. The CCPR1L contains the eight MSbs and the CCP1CON<5:4> contains the two LSbs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. The PWM duty cycle is calculated by the following equation:

EQUATION 18-2:

$$\text{PWM Duty Cycle} = (\text{CCPR1L:CCP1CON<5:4>} \cdot \text{Tosc} \cdot (\text{TMR2 Prescale Value}))$$

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

The CCPR1H register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation. When the CCPR1H and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or two bits of the TMR2 prescaler, the ECCP1 pin is cleared. The maximum PWM resolution (bits) for a given PWM frequency is given by the equation:

EQUATION 18-3:

$$\text{PWM Resolution (max)} = \frac{\log\left(\frac{\text{Fosc}}{\text{FPWM}}\right)}{\log(2)} \text{ bits}$$

Note: If the PWM duty cycle value is longer than the PWM period, the ECCP1 pin will not be cleared.

18.4.3 PWM OUTPUT CONFIGURATIONS

The P1M1:P1M0 bits in the CCP1CON register allow one of four configurations:

- Single Output
- Half-Bridge Output
- Full-Bridge Output, Forward mode
- Full-Bridge Output, Reverse mode

The Single Output mode is the standard PWM mode discussed in **Section 18.4 “Enhanced PWM Mode”**. The Half-Bridge and Full-Bridge Output modes are covered in detail in the sections that follow.

The general relationship of the outputs in all configurations is summarized in Figure 18-2.

TABLE 18-4: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz

PWM Frequency	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	FFh	FFh	FFh	3Fh	1Fh	17h
Maximum Resolution (bits)	10	10	10	8	7	6.58

FIGURE 18-2: PWM OUTPUT RELATIONSHIPS (ACTIVE-HIGH STATE)

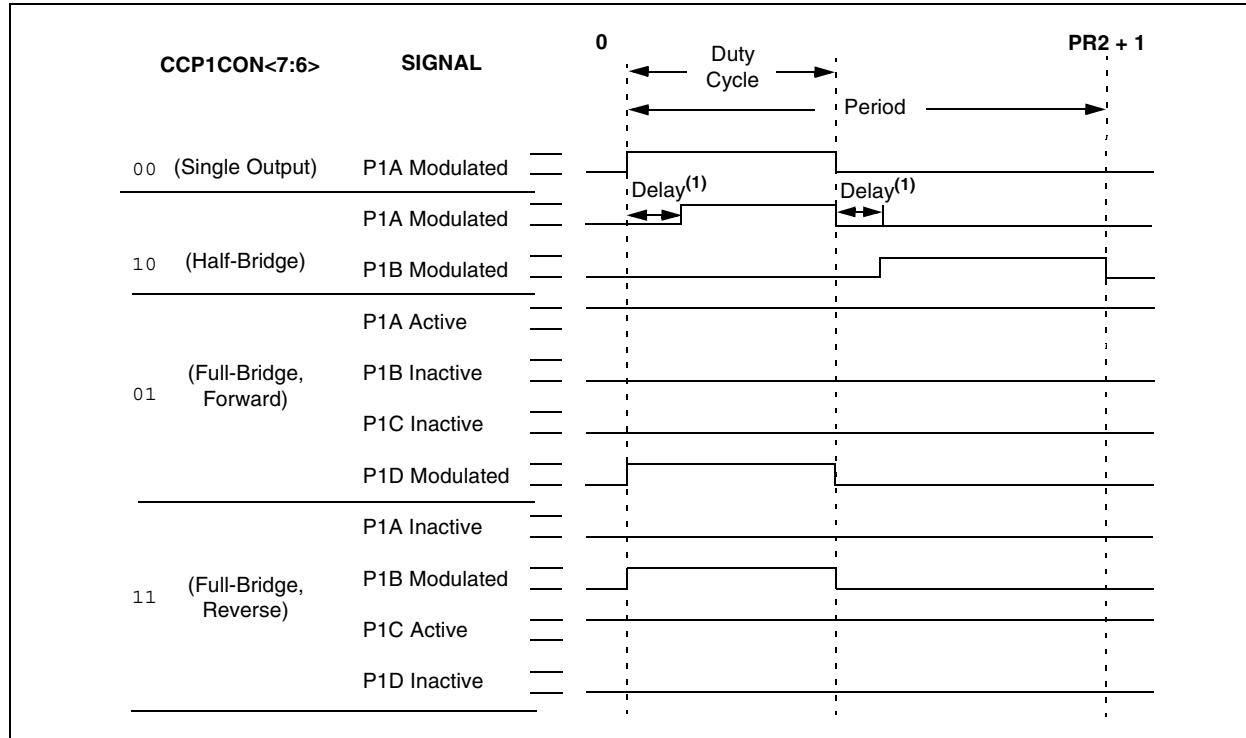
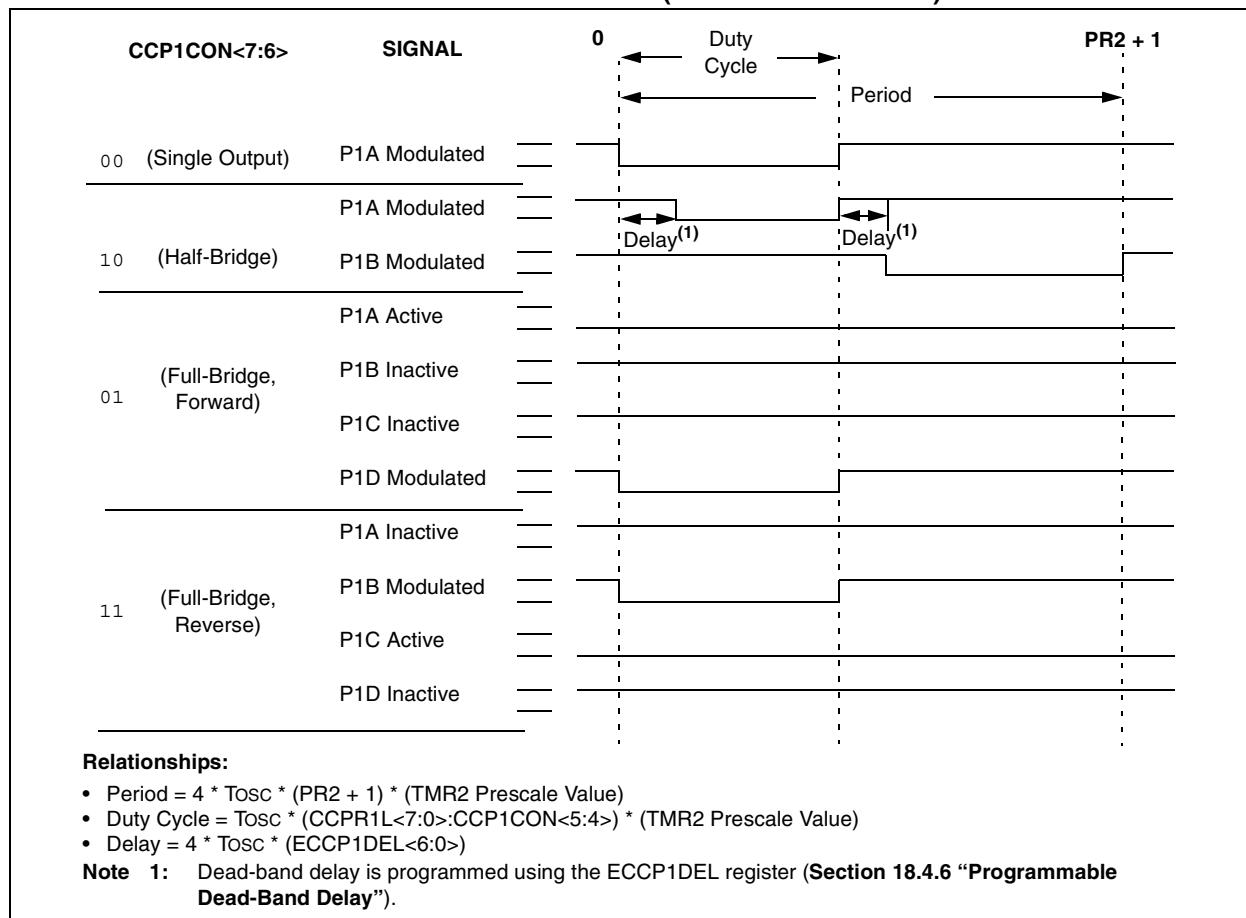


FIGURE 18-3: PWM OUTPUT RELATIONSHIPS (ACTIVE-LOW STATE)



PIC18F87J11 FAMILY

18.4.4 HALF-BRIDGE MODE

In the Half-Bridge Output mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the P1A pin, while the complementary PWM output signal is output on the P1B pin (Figure 18-4). This mode can be used for half-bridge applications, as shown in Figure 18-5, or for full-bridge applications, where four power switches are being modulated with two PWM signals.

In Half-Bridge Output mode, the programmable dead-band delay can be used to prevent shoot-through current in half-bridge power devices. The value of bits P1DC6:P1DC0 sets the number of instruction cycles before the output is driven active. If the value is greater than the duty cycle, the corresponding output remains inactive during the entire cycle. See **Section 18.4.6 “Programmable Dead-Band Delay”** for more details on dead-band delay operations.

Since the P1A and P1B outputs are multiplexed with the PORTC<2> and PORTE<6> data latches, the TRISC<2> and TRISE<6> bits must be cleared to configure P1A and P1B as outputs.

FIGURE 18-4: HALF-BRIDGE PWM OUTPUT

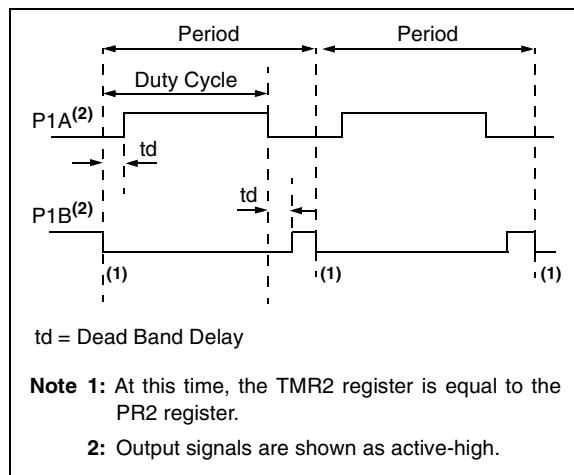
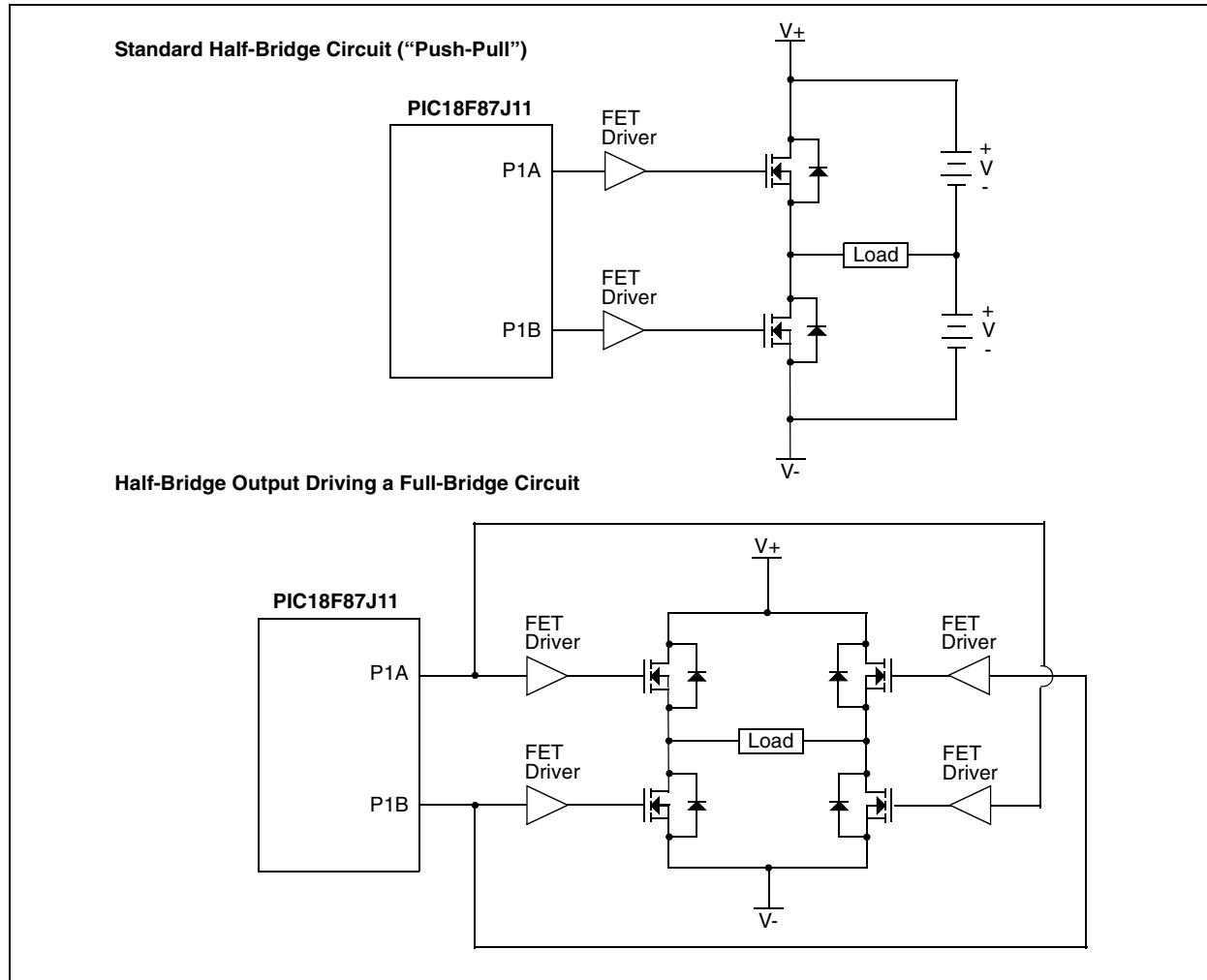


FIGURE 18-5: EXAMPLES OF HALF-BRIDGE OUTPUT MODE APPLICATIONS

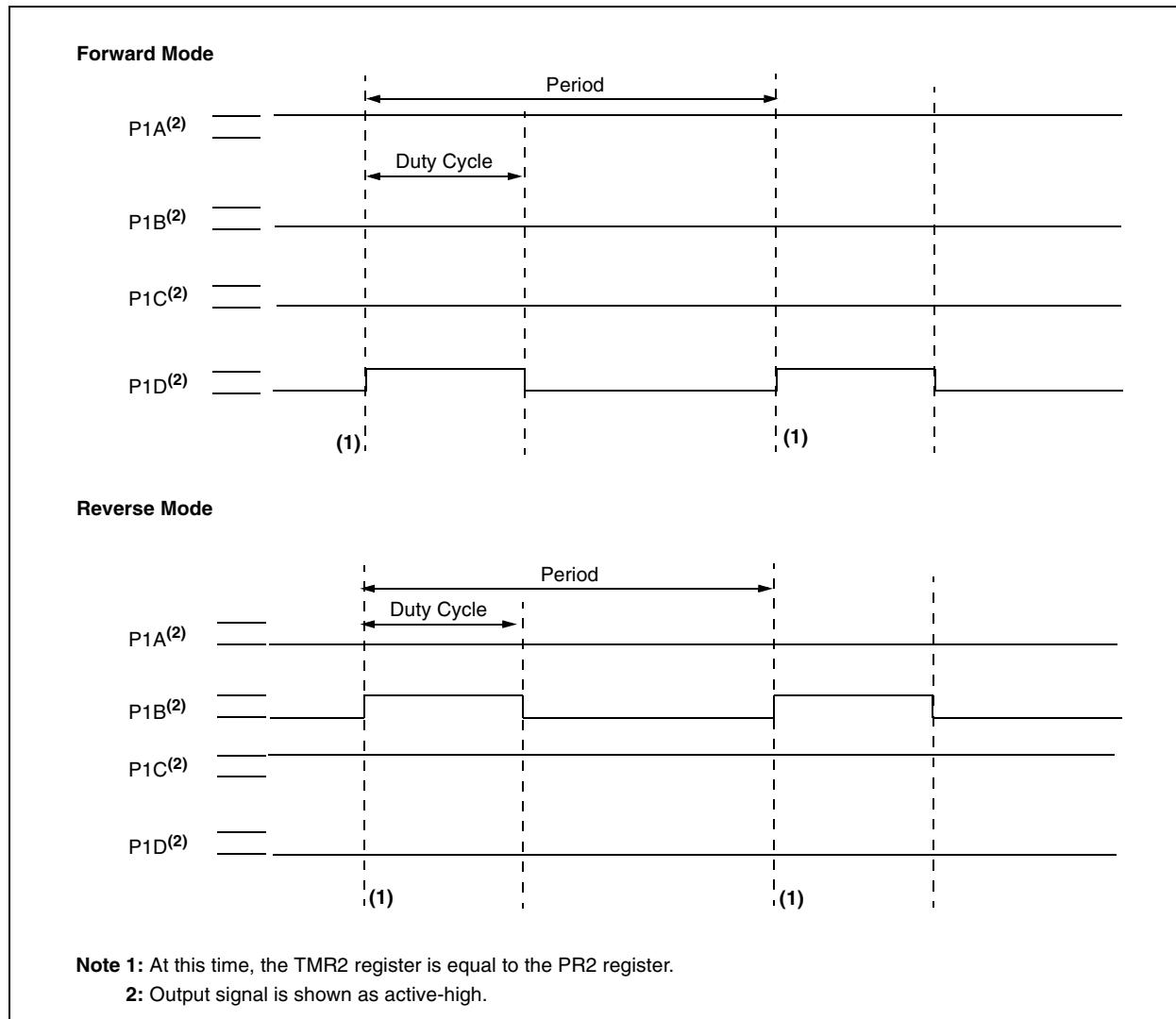


18.4.5 FULL-BRIDGE MODE

In Full-Bridge Output mode, four pins are used as outputs; however, only two outputs are active at a time. In the Forward mode, pin P1A is continuously active and pin P1D is modulated. In the Reverse mode, pin P1C is continuously active and pin P1B is modulated. These are illustrated in Figure 18-6.

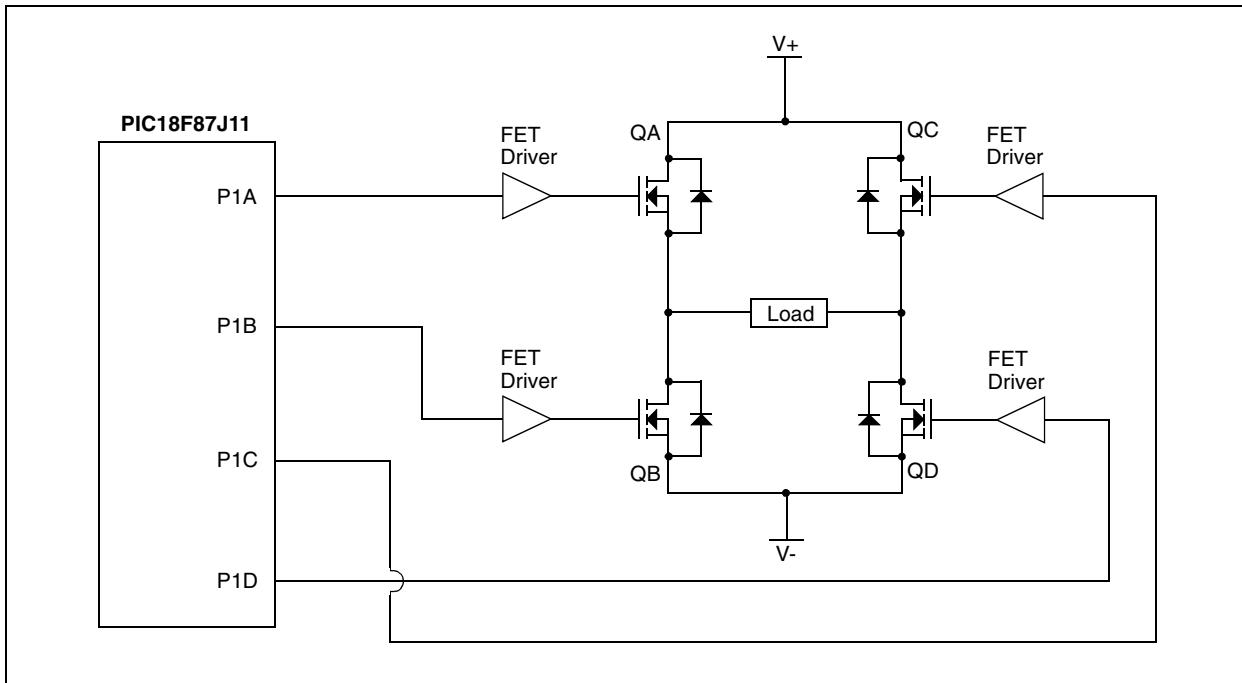
P1A, P1B, P1C and P1D outputs are multiplexed with the port pins as described in Table 18-1, Table 18-2 and Table 18-3. The corresponding TRIS bits must be cleared to make the P1A, P1B, P1C and P1D pins outputs.

FIGURE 18-6: FULL-BRIDGE PWM OUTPUT



PIC18F87J11 FAMILY

FIGURE 18-7: EXAMPLE OF FULL-BRIDGE OUTPUT APPLICATION



18.4.5.1 Direction Change in Full-Bridge Output Mode

In the Full-Bridge Output mode, the P1M1 bit in the CCP1CON register allows users to control the forward/reverse direction. When the application firmware changes this direction control bit, the module will assume the new direction on the next PWM cycle.

Just before the end of the current PWM period, the modulated outputs (P1B and P1D) are placed in their inactive state, while the unmodulated outputs (P1A and P1C) are switched to drive in the opposite direction. This occurs in a time interval of $(4 \text{ Tosc} * (\text{Timer2 Prescale Value}))$ before the next PWM period begins. The Timer2 prescaler will be either 1, 4 or 16, depending on the value of the T2CKPS bits (T2CON<1:0>). During the interval from the switch of the unmodulated outputs to the beginning of the next period, the modulated outputs (P1B and P1D) remain inactive. This relationship is shown in Figure 18-8.

Note that in the Full-Bridge Output mode, the ECCP1 module does not provide any dead-band delay. In general, since only one output is modulated at all times, dead-band delay is not required. However, there is a situation where a dead-band delay might be required. This situation occurs when both of the following conditions are true:

1. The direction of the PWM output changes when the duty cycle of the output is at or near 100%.
2. The turn-off time of the power switch, including the power device and driver circuit, is greater than the turn-on time.

Figure 18-9 shows an example where the PWM direction changes from forward to reverse at a near 100% duty cycle. At time t1, the outputs, P1A and P1D, become inactive, while output, P1C, becomes active. In this example, since the turn-off time of the power devices is longer than the turn-on time, a shoot-through current may flow through power devices, QC and QD (see Figure 18-7), for the duration of 't'. The same phenomenon will occur to power devices, QA and QB, for PWM direction change from reverse to forward.

If changing PWM direction at high duty cycle is required for an application, one of the following requirements must be met:

1. Reduce PWM for a PWM period before changing directions.
2. Use switch drivers that can drive the switches off faster than they can drive them on.

Other options to prevent shoot-through current may exist.

FIGURE 18-8: PWM DIRECTION CHANGE

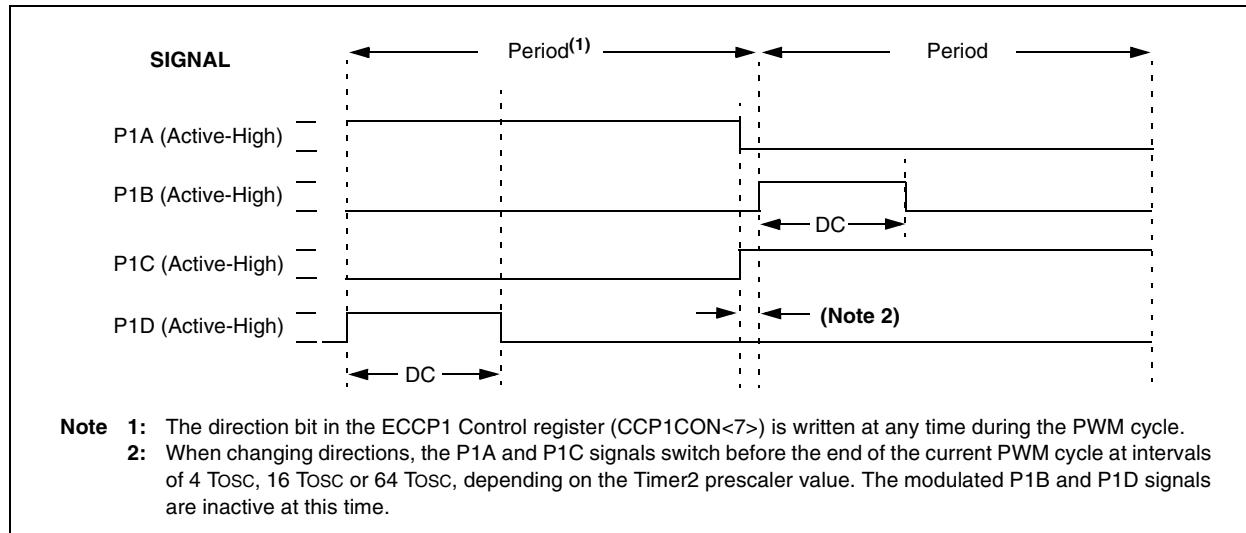
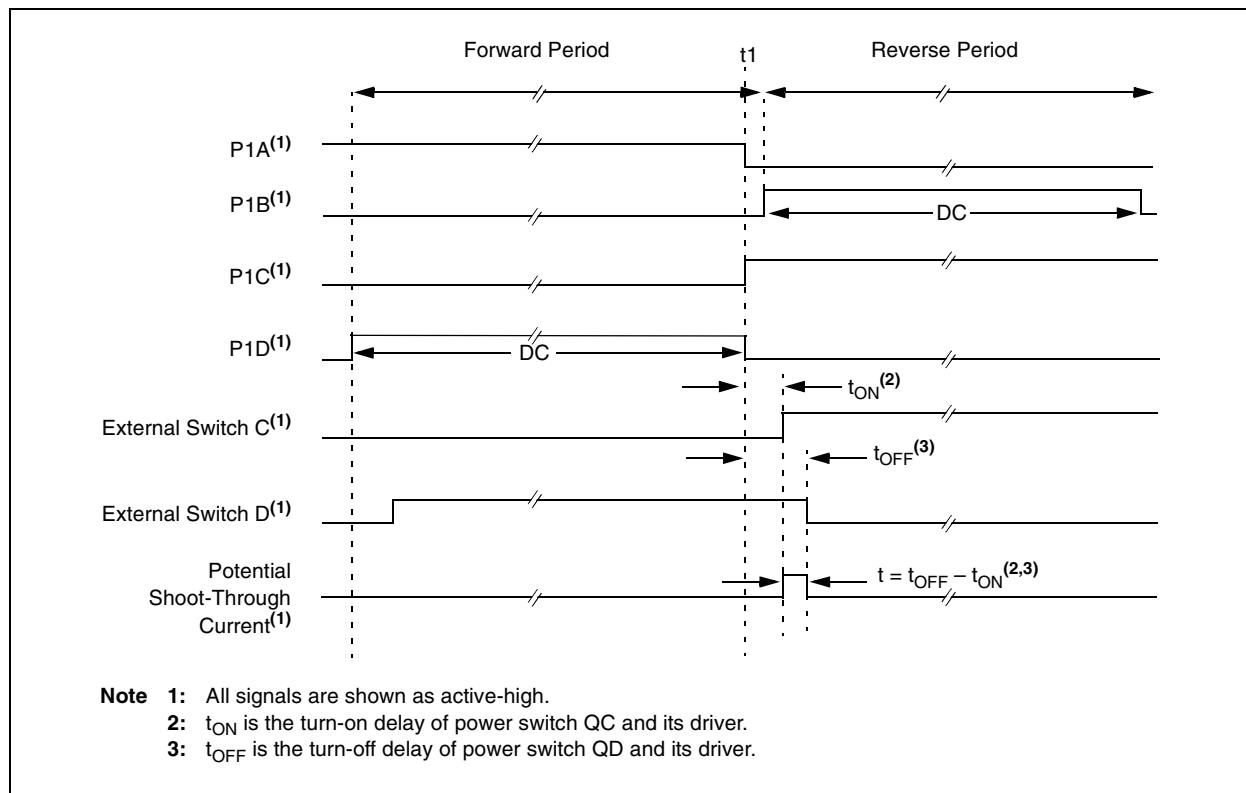


FIGURE 18-9: PWM DIRECTION CHANGE AT NEAR 100% DUTY CYCLE



PIC18F87J11 FAMILY

18.4.6 PROGRAMMABLE DEAD-BAND DELAY

In half-bridge applications, where all power switches are modulated at the PWM frequency at all times, the power switches normally require more time to turn off than to turn on. If both the upper and lower power switches are switched at the same time (one turned on and the other turned off), both switches may be on for a short period of time until one switch completely turns off. During this brief interval, a very high current (*shoot-through current*) may flow through both power switches, shorting the bridge supply. To avoid this potentially destructive shoot-through current from flowing during switching, turning on either of the power switches is normally delayed to allow the other switch to completely turn off.

In the Half-Bridge Output mode, a digitally programmable, dead-band delay is available to avoid shoot-through current from destroying the bridge power switches. The delay occurs at the signal transition from the non-active state to the active state (see Figure 18-4 for illustration). The lower seven bits of the ECCPxDEL register (Register 18-2) set the delay period in terms of microcontroller instruction cycles (Tcy or 4 Tosc).

18.4.7 ENHANCED PWM AUTO-SHUTDOWN

When the ECCP1 is programmed for any of the Enhanced PWM modes, the active output pins may be configured for auto-shutdown. Auto-shutdown immediately places the Enhanced PWM output pins into a defined shutdown state when a shutdown event occurs.

A shutdown event can be caused by either of the two comparator modules or the FLT0 pin (or any combination of these three sources). The comparators may be used to monitor a voltage input proportional to a current being monitored in the bridge circuit. If the voltage exceeds a threshold, the comparator switches state and triggers a shutdown. Alternatively, a low-level digital signal on the FLT0 pin can also trigger a shutdown. The auto-shutdown feature can be disabled by not selecting any auto-shutdown sources. The auto-shutdown sources to be used are selected using the ECCP1AS2:ECCP1AS0 bits (ECCP1AS<6:4>).

When a shutdown occurs, the output pins are asynchronously placed in their shutdown states, specified by the PSS1AC1:PSS1AC0 and PSS1BD1:PSS1BD0 bits (ECCP1AS3:ECCP1AS0). Each pin pair (P1A/P1C and P1B/P1D) may be set to drive high, drive low or be tri-stated (not driving). The ECCP1ASE bit (ECCP1AS<7>) is also set to hold the Enhanced PWM outputs in their shutdown states.

The ECCP1ASE bit is set by hardware when a shutdown event occurs. If automatic restarts are not enabled, the ECCP1ASE bit is cleared by firmware when the cause of the shutdown clears. If automatic restarts are enabled, the ECCP1ASE bit is automatically cleared when the cause of the auto-shutdown has cleared.

If the ECCP1ASE bit is set when a PWM period begins, the PWM outputs remain in their shutdown state for that entire PWM period. When the ECCP1ASE bit is cleared, the PWM outputs will return to normal operation at the beginning of the next PWM period.

Note: Writing to the ECCP1ASE bit is disabled while a shutdown condition is active.

REGISTER 18-2: ECCPxDEL: ECCPx PWM DELAY REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PxRSEN	PxDC6	PxDC5	PxDC4	PxDC3	PxDC2	PxDC1	PxDC0
bit 7	bit 0						

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

bit 7

PxRSEN: PWM Restart Enable bit

1 = Upon auto-shutdown, the ECCPxASE bit clears automatically once the shutdown event goes away; the PWM restarts automatically

0 = Upon auto-shutdown, ECCPxASE must be cleared in software to restart the PWM

bit 6-0

PxDC6:PxDC0: PWM Delay Count bits

Delay time, in number of Fosc/4 (4 * Tosc) cycles, between the scheduled and actual time for a PWM signal to transition to active.

REGISTER 18-3: ECCPxAS: ECCPx AUTO-SHUTDOWN CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ECCPxASE	ECCPxAS2	ECCPxAS1	ECCPxAS0	PSSxAC1	PSSxAC0	PSSxBD1	PSSxBD0
bit 7	bit 0						

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

bit 7	ECCPxASE: ECCPx Auto-Shutdown Event Status bit 0 = ECCPx outputs are operating 1 = A shutdown event has occurred; ECCPx outputs are in shutdown state
bit 6-4	ECCPxAS2:ECCPxAS0: ECCPx Auto-Shutdown Source Select bits 000 = Auto-shutdown is disabled 001 = Comparator 1 output 010 = Comparator 2 output 011 = Either Comparator 1 or 2 100 = FLT0 101 = FLT0 or Comparator 1 110 = FLT0 or Comparator 2 111 = FLT0 or Comparator 1 or Comparator 2
bit 3-2	PSSxAC1:PSSxAC0: Pins A and C Shutdown State Control bits 00 = Drive Pins A and C to '0' 01 = Drive Pins A and C to '1' 1x = Pins A and C tri-state
bit 1-0	PSSxBD1:PSSxBD0: Pins B and D Shutdown State Control bits 00 = Drive Pins B and D to '0' 01 = Drive Pins B and D to '1' 1x = Pins B and D tri-state

18.4.7.1 Auto-Shutdown and Automatic Restart

The auto-shutdown feature can be configured to allow automatic restarts of the module following a shutdown event. This is enabled by setting the P1RSEN bit of the ECCP1DEL register (ECCP1DEL<7>).

In Shutdown mode with P1RSEN = 1 (Figure 18-10), the ECCP1ASE bit will remain set for as long as the cause of the shutdown continues. When the shutdown condition clears, the ECCP1ASE bit is cleared. If P1RSEN = 0 (Figure 18-11), once a shutdown condition occurs, the ECCP1ASE bit will remain set until it is cleared by firmware. Once ECCP1ASE is cleared, the Enhanced PWM will resume at the beginning of the next PWM period.

Note: Writing to the ECCP1ASE bit is disabled while a shutdown condition is active.

Independent of the P1RSEN bit setting, if the auto-shutdown source is one of the comparators, the shutdown condition is a level. The ECCP1ASE bit cannot be cleared as long as the cause of the shutdown persists.

The Auto-Shutdown mode can be forced by writing a '1' to the ECCP1ASE bit.

18.4.8 START-UP CONSIDERATIONS

When the ECCP1 module is used in the PWM mode, the application hardware must use the proper external pull-up and/or pull-down resistors on the PWM output pins. When the microcontroller is released from Reset, all of the I/O pins are in the high-impedance state. The external circuits must keep the power switch devices in the OFF state until the microcontroller drives the I/O pins with the proper signal levels, or activates the PWM output(s).

PIC18F87J11 FAMILY

The CCP1M1:CCP1M0 bits (CCP1CON<1:0>) allow the user to choose whether the PWM output signals are active-high or active-low for each pair of PWM output pins (P1A/P1C and P1B/P1D). The PWM output polarities must be selected before the PWM pins are configured as outputs. Changing the polarity configuration while the PWM pins are configured as outputs is not recommended since it may result in damage to the application circuits.

The P1A, P1B, P1C and P1D output latches may not be in the proper states when the PWM module is initialized. Enabling the PWM pins for output at the same time as the ECCP1 module may cause damage to the application circuit. The ECCP1 module must be enabled in the proper output mode and complete a full PWM cycle before configuring the PWM pins as outputs. The completion of a full PWM cycle is indicated by the TMR2IF bit being set as the second PWM period begins.

FIGURE 18-10: PWM AUTO-SHUTDOWN (P1RSEN = 1, AUTO-RESTART ENABLED)

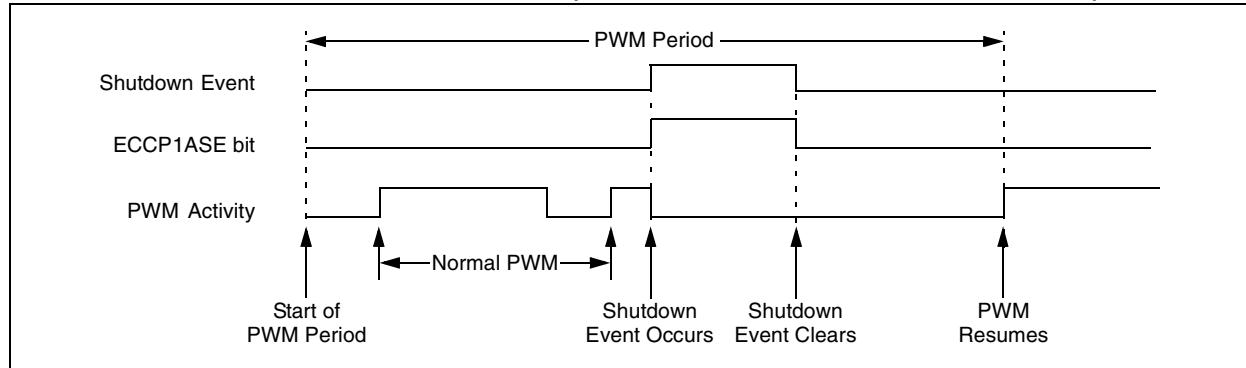
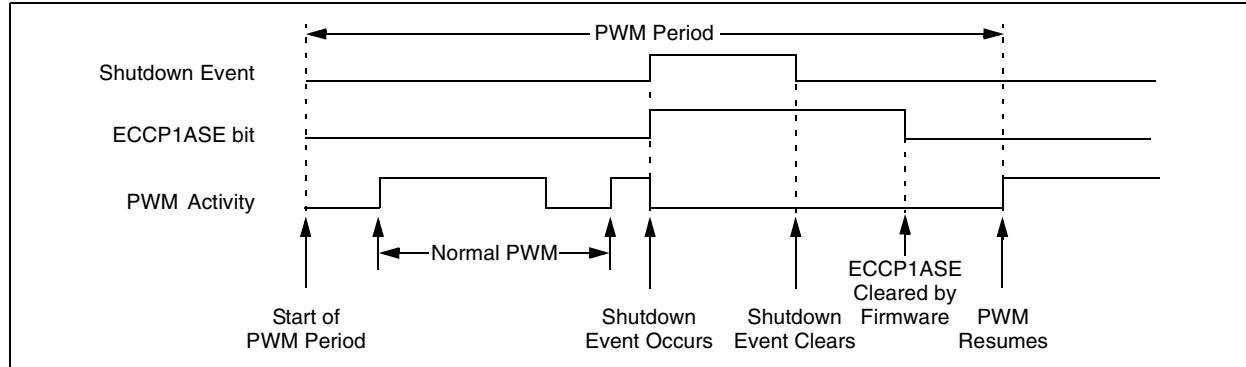


FIGURE 18-11: PWM AUTO-SHUTDOWN (P1RSEN = 0, AUTO-RESTART DISABLED)



18.4.9 SETUP FOR PWM OPERATION

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

1. Configure the PWM pins Px A and Px B (and Px C and Px D, if used) as inputs by setting the corresponding TRIS bits.
2. Set the PWM period by loading the PR2 (PR4) register.
3. Configure the ECCP module for the desired PWM mode and configuration by loading the CCPxCON register with the appropriate values:
 - Select one of the available output configurations and direction with the PxM1:PXM0 bits.
 - Select the polarities of the PWM output signals with the CCPxM3:CCPxM0 bits.
4. Set the PWM duty cycle by loading the CCPRxL register and the CCPxCON<5:4> bits.
5. For auto-shutdown:
 - Disable auto-shutdown; ECCPxASE = 0
 - Configure auto-shutdown source
 - Wait for Run condition
6. For Half-Bridge Output mode, set the dead-band delay by loading ECCPxDEL<6:0> with the appropriate value.
7. If auto-shutdown operation is required, load the ECCPxAS register:
 - Select the auto-shutdown sources using the ECCPxAS2:ECCPxAS0 bits.
 - Select the shutdown states of the PWM output pins using the PSSxAC1:PSSxAC0 and PSSxBD1:PSSxBD0 bits.
 - Set the ECCPxASE bit (ECCPxAS<7>).

8. If auto-restart operation is required, set the PxRSEN bit (ECCPxDEL<7>).
9. Configure and start TMRn (TMR2 or TMR4):
 - Clear the TMRn interrupt flag bit by clearing the TMRnIF bit (PIR1<1> for Timer2 or PIR3<3> for Timer4).
 - Set the TMRn prescale value by loading the TnCKPS bits (TnCON<1:0>).
 - Enable Timer2 (or Timer4) by setting the TMRnON bit (TnCON<2>).
10. Enable PWM outputs after a new PWM cycle has started:
 - Wait until TMRn overflows (TMRnIF bit is set).
 - Enable the ECCPx/PxA, Px B, Px C and/or Px D pin outputs by clearing the respective TRIS bits.
 - Clear the ECCPxASE bit (ECCPxAS<7>).

18.4.10 EFFECTS OF A RESET

Both Power-on Reset and subsequent Resets will force all ports to Input mode and the ECCP registers to their Reset states.

This forces the Enhanced CCP module to reset to a state compatible with the standard CCP module.

PIC18F87J11 FAMILY

TABLE 18-5: REGISTERS ASSOCIATED WITH ECCP MODULES AND TIMER1 TO TIMER4

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
RCON	IPEN	—	CM	RI	TO	PD	POR	BOR	56
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	58
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	58
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	58
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	58
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	58
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	58
TRISH ⁽¹⁾	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	58
TMR1L ⁽³⁾	Timer1 Register Low Byte								56
TMR1H ⁽³⁾	Timer1 Register High Byte								56
ODCON1 ⁽⁴⁾	—	—	—	CCP5OD	CCP4OD	ECCP3OD	ECCP2OD	ECCP1OD	56
T1CON ⁽³⁾	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	56
TMR2 ⁽³⁾	Timer2 Register								56
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	56
PR2 ⁽³⁾	Timer2 Period Register								56
TMR3L	Timer3 Register Low Byte								59
TMR3H	Timer3 Register High Byte								59
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON	59
TMR4	Timer4 Register								59
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	59
PR4 ⁽³⁾	Timer4 Period Register								59
CCPRxL ⁽²⁾	Capture/Compare/PWM Register x Low Byte								57
CCPRxH ⁽²⁾	Capture/Compare/PWM Register x High Byte								57
CCPxCON ⁽²⁾	PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	57
ECCPxAS ⁽²⁾	ECCPxASE	ECCPxAS2	ECCPxAS1	ECCPxAS0	PSSxAC1	PSSxAC0	PSSxBD1	PSSxBD0	57
ECCPxDEL ⁽²⁾	PxRSEN	PxDC6	PxDC5	PxDC4	PxDC3	PxDC2	PxDC1	PxDC0	57

Legend: — = unimplemented, read as '0'. Shaded cells are not used during ECCP operation.

Note 1: Available on 80-pin devices only.

- 2: Generic term for all of the identical registers of this name for all Enhanced CCP modules, where 'x' identifies the individual module (ECCP1, ECCP2 or ECCP3). Bit assignments and Reset values for all registers of the same generic name are identical.
- 3: Default (legacy) SFR at this address, available when WDTCON<4> = 0.
- 4: Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.

19.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

19.1 Master SSP (MSSP) Module Overview

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

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C TM)
 - Full Master mode
 - Slave mode (with general address call)

The I^2C interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode with 5-bit and 7-bit address masking (with address masking for both 10-bit and 7-bit addressing)

All members of the PIC18F87J11 Family have two MSSP modules, designated as MSSP1 and MSSP2. Each module operates independently of the other.

Note: Throughout this section, generic references to an MSSP module in any of its operating modes may be interpreted as being equally applicable to MSSP1 or MSSP2. Register names and module I/O signals use the generic designator 'x' to indicate the use of a numeral to distinguish a particular module when required. Control bit names are not individuated.

19.2 Control Registers

Each MSSP module has three associated control registers. These include a status register (SSPxSTAT) and two control registers (SSPxCON1 and SSPxCON2). The use of these registers and their individual configuration bits differ significantly depending on whether the MSSP module is operated in SPI or I^2C mode.

Additional details are provided under the individual sections.

Note: In devices with more than one MSSP module, it is very important to pay close attention to SSPxCON register names. SSP1CON1 and SSP1CON2 control different operational aspects of the same module, while SSP1CON1 and SSP2CON1 control the same features for two different modules.

19.3 SPI Mode

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

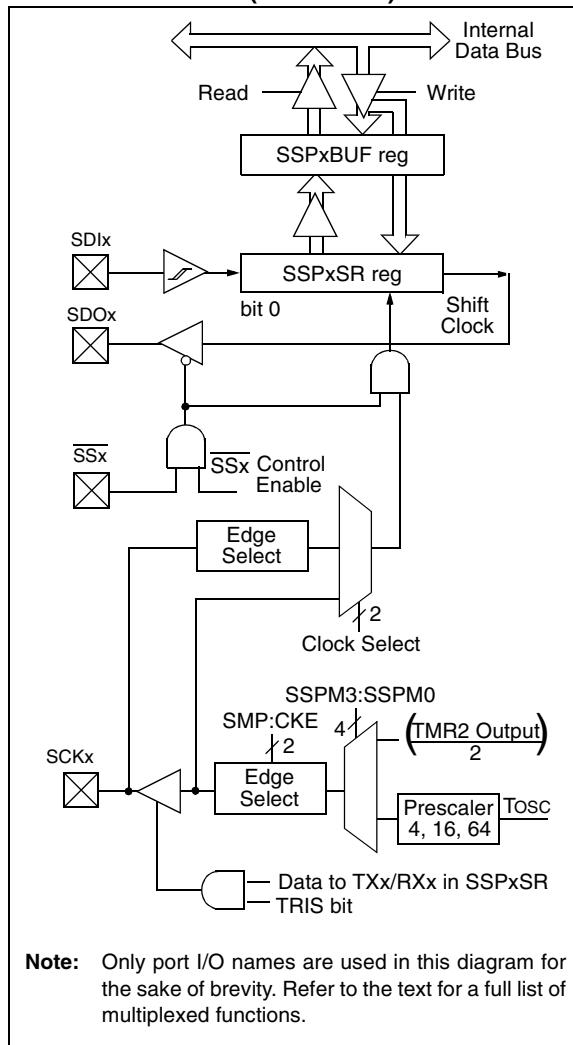
- Serial Data Out (SDO_x) – RC5/SDO1 or RD4/SDO2
- Serial Data In (SDI_x) – RC4/SDI1/SDA1 or RD5/SDI2/SDA2
- Serial Clock (SCK_x) – RC3/SCK1/SCL1 or RD6/SCK2/SCL2

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

- Slave Select ($\overline{SS_x}$) – RF7/SS1 or RD7/SS2

Figure 19-1 shows the block diagram of the MSSP module when operating in SPI mode.

FIGURE 19-1: MSSP BLOCK DIAGRAM (SPI MODE)



PIC18F87J11 FAMILY

19.3.1 REGISTERS

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

- MSSPx Control Register 1 (SSPxCON1)
- MSSPx Status Register (SSPxSTAT)
- Serial Receive/Transmit Buffer Register (SSPxBUF)
- MSSPx Shift Register (SSPxSR) – Not directly accessible

SSPxCON1 and SSPxSTAT are the control and status registers in SPI mode operation. The SSPxCON1 register is readable and writable. The lower 6 bits of the SSPxSTAT are read-only. The upper two bits of the SSPxSTAT are read/write.

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

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

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

REGISTER 19-1: SSPxSTAT: MSSPx STATUS REGISTER (SPI MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE ⁽¹⁾	D/A	P	S	R/W	UA	BF
bit 7	bit 0						

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

bit 7	SMP: Sample bit <u>SPI Master mode:</u> 1 = Input data sampled at end of data output time 0 = Input data sampled at middle of data output time <u>SPI Slave mode:</u> SMP must be cleared when SPI is used in Slave mode.
bit 6	CKE: SPI Clock Select bit ⁽¹⁾ 1 = Transmit occurs on transition from active to Idle clock state 0 = Transmit occurs on transition from Idle to active clock state
bit 5	D/A: Data/Address bit Used in I ² C mode only.
bit 4	P: Stop bit Used in I ² C mode only. This bit is cleared when the MSSPx module is disabled, SSPEN is cleared.
bit 3	S: Start bit Used in I ² C mode only.
bit 2	R/W: Read/Write Information bit Used in I ² C mode only.
bit 1	UA: Update Address bit Used in I ² C mode only.
bit 0	BF: Buffer Full Status bit (Receive mode only) 1 = Receive complete, SSPxBUF is full 0 = Receive not complete, SSPxBUF is empty

Note 1: Polarity of clock state is set by the CKP bit (SSPxCON1<4>).

REGISTER 19-2: SSPxCON1: MSSPx CONTROL REGISTER 1 (SPI MODE)

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 ⁽¹⁾	SSPEN ⁽²⁾	CKP	SSPM3 ⁽³⁾	SSPM2 ⁽³⁾	SSPM1 ⁽³⁾	SSPM0 ⁽³⁾	
bit 7					bit 0			

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

bit 7	WCOL: Write Collision Detect bit 1 = The SSPxBUF 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 ⁽¹⁾ SPI Slave mode: 1 = A new byte is received while the SSPxBUF register is still holding the previous data. In case of overflow, the data in SSPxSR is lost. Overflow can only occur in Slave mode. The user must read the SSPxBUF, even if only transmitting data, to avoid setting overflow (must be cleared in software). 0 = No overflow
bit 5	SSPEN: Master Synchronous Serial Port Enable bit ⁽²⁾ 1 = Enables serial port and configures SCKx, SDIx, SDIx and <u>SSx</u> as serial port pins 0 = Disables serial port and configures these pins as I/O port pins
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: Master Synchronous Serial Port Mode Select bits ⁽³⁾ 0101 = SPI Slave mode, clock = SCKx pin, <u>SSx</u> pin control disabled, <u>SSx</u> can be used as I/O pin 0100 = SPI Slave mode, clock = SCKx pin, <u>SSx</u> 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 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPxBUF register.

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

3: Bit combinations not specifically listed here are either reserved or implemented in I²C mode only.

PIC18F87J11 FAMILY

19.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPxCON1<5:0> and SSPxSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCKx is the clock output)
- Slave mode (SCKx is the clock input)
- Clock Polarity (Idle state of SCKx)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCKx)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

Each MSSP module consists of a transmit/receive shift register (SSPxSR) and a buffer register (SSPxBUF). The SSPxSR shifts the data in and out of the device, MSb first. The SSPxBUF holds the data that was written to the SSPxSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPxBUF register. Then, the Buffer Full detect bit, BF (SSPxSTAT<0>) and the interrupt flag bit, SSPxFIF, are set. This double-buffering of the received data (SSPxBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPxBUF register during transmission/reception of data will be ignored and the Write Collision Detect bit, WCOL (SSPxCON1<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPxBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPxBUF should be read before the next byte of data to transfer is written to the SSPxBUF. The Buffer Full bit, BF (SSPxSTAT<0>), indicates when SSPxBUF has been loaded with the received data (transmission is complete). When the SSPxBUF 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. 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 19-1 shows the loading of the SSPxBUF (SSPxSR) for data transmission.

The SSPxSR is not directly readable or writable and can only be accessed by addressing the SSPxBUF register. Additionally, the SSPxSTAT register indicates the various status conditions.

19.3.3 OPEN-DRAIN OUTPUT OPTION

The drivers for the SDOx output and SCKx clock pins can be optionally configured as open-drain outputs. This feature allows the voltage level on the pin to be pulled to a higher level through an external pull-up resistor, and allows the output to communicate with external circuits without the need for additional level shifters. For more information, see **Section 10.1.4 “Open-Drain Outputs”**.

The open-drain output option is controlled by the SPI2OD and SPI1OD bits (ODCON3<1:0>). Setting an SPIxOD bit configures the SDOx and SCKx pins for the corresponding module for open-drain operation.

The ODCON3 register shares the same address as the T1CON register. The ODCON3 register is accessed by setting the ADSHR bit in the WDTCON register (WDTCON<4>).

EXAMPLE 19-1: LOADING THE SSP1BUF (SSP1SR) REGISTER

LOOP	BTFS	SSP1STAT, BF	;Has data been received (transmit complete)?
	BRA	LOOP	;No
	MOVF	SSP1BUF, W	;WREG reg = contents of SSP1BUF
	MOVWF	RXDATA	;Save in user RAM, if data is meaningful
	MOVF	TXDATA, W	;W reg = contents of TXDATA
	MOVWF	SSP1BUF	;New data to xmit

19.3.4 ENABLING SPI I/O

To enable the serial port, MSSP Enable bit, SSPEN ($\text{SSPxCON}1<5>$), must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, reinitialize the SSPxCON registers and then set the SSPEN bit. This configures the SDIx, SDOx, SCKx and SSx 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 as follows:

- SDIx is automatically controlled by the SPI module
- SDOx must have the TRISC<5> or TRISD<4> bit cleared
- SCKx (Master mode) must have the TRISC<3> or TRISD<6> bit cleared
- SCKx (Slave mode) must have the TRISC<3> or TRISD<6> bit set
- SSx must have the TRISF<7> or TRISD<7> bit set

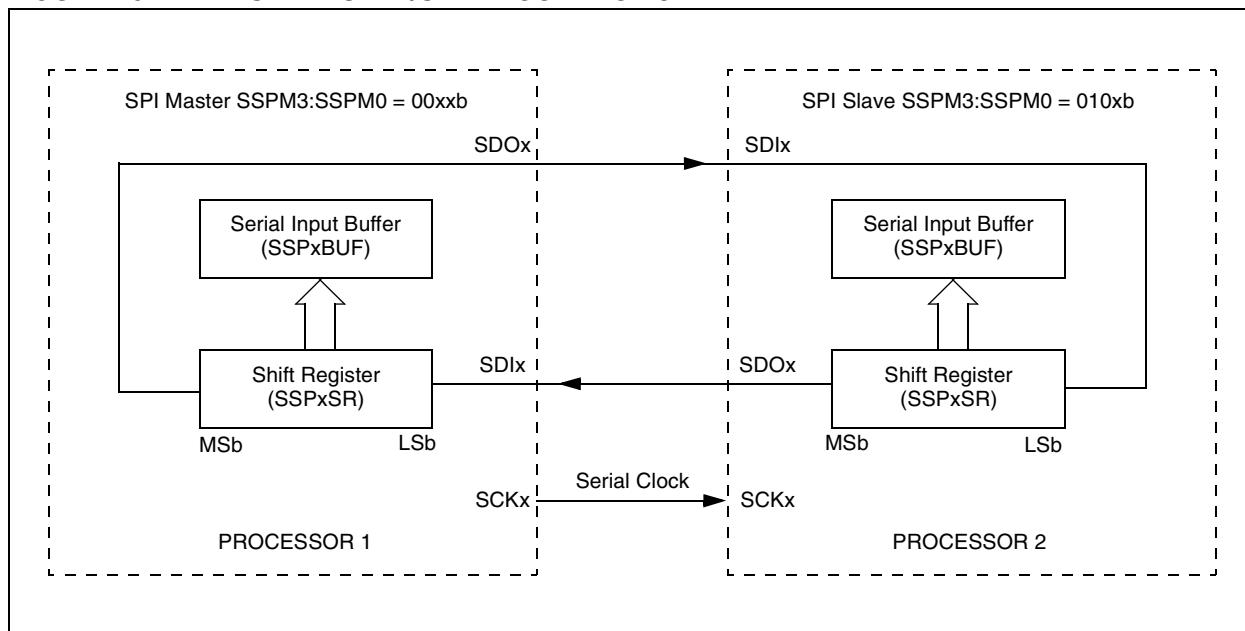
Any serial port function that is not desired may be overridden by programming the corresponding Data Direction (TRIS) register to the opposite value.

19.3.5 TYPICAL CONNECTION

Figure 19-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCKx 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 19-2: SPI MASTER/SLAVE CONNECTION



PIC18F87J11 FAMILY

19.3.6 MASTER MODE

The master can initiate the data transfer at any time because it controls the SCKx. The master determines when the slave (Processor 1, Figure 19-2) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPxBUF register is written to. If the SPI is only going to receive, the SDOx output could be disabled (programmed as an input). The SSPxSR register will continue to shift in the signal present on the SDIx pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPxBUF 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 (SSPxCON1<4>). This then, would give waveforms for SPI communication as

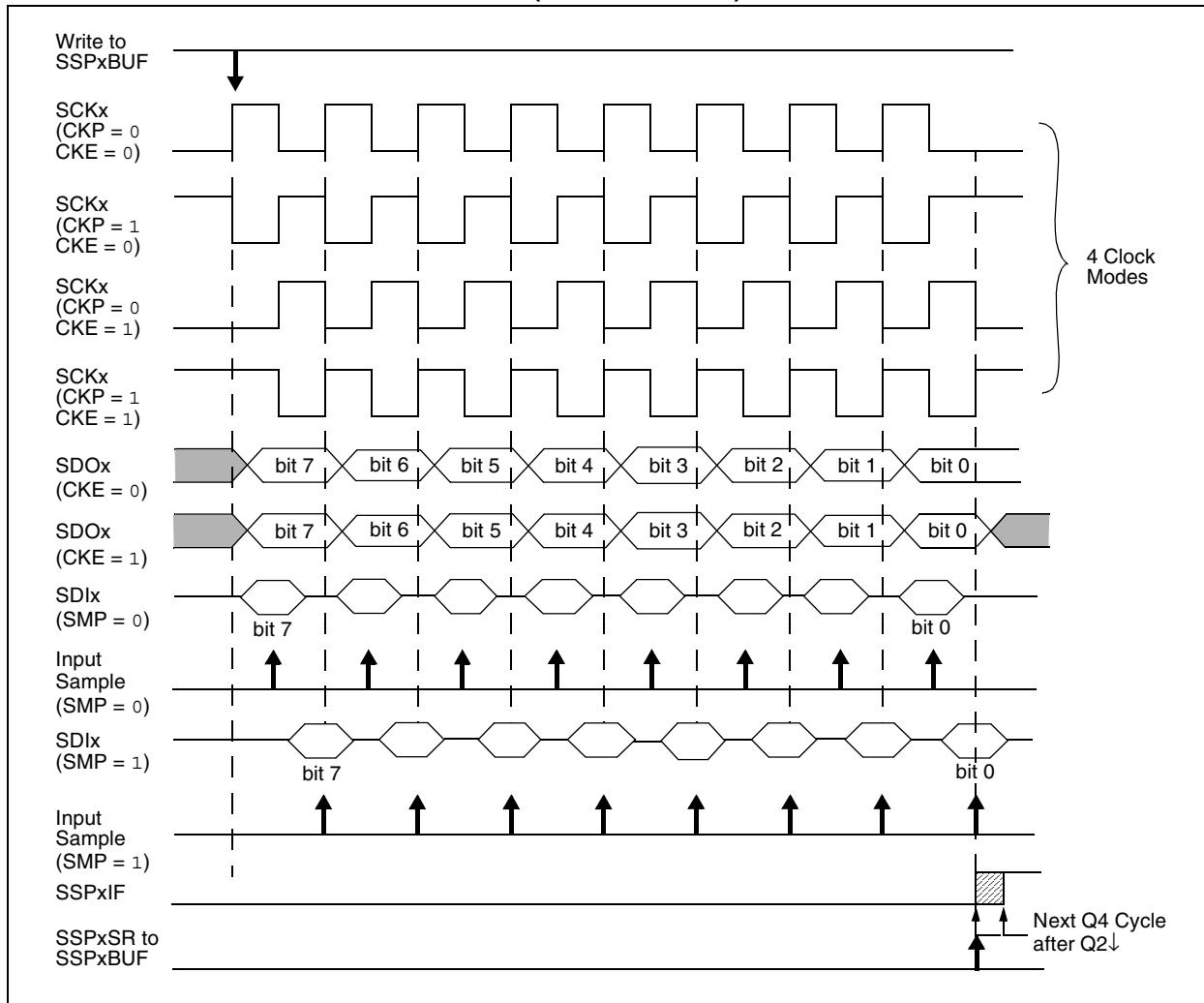
shown in Figure 19-3, Figure 19-5 and Figure 19-6, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- Fosc/4 (or Tcy)
- Fosc/16 (or $4 \cdot Tcy$)
- Fosc/64 (or $16 \cdot Tcy$)
- Timer2 output/2

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

Figure 19-3 shows the waveforms for Master mode. When the CKE bit is set, the SDOx data is valid before there is a clock edge on SCKx. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPxBUF is loaded with the received data is shown.

FIGURE 19-3: SPI MODE WAVEFORM (MASTER MODE)



19.3.7 SLAVE MODE

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

While in Slave mode, the external clock is supplied by the external clock source on the SCKx 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 can be configured to wake-up from Sleep.

19.3.8 SLAVE SELECT SYNCHRONIZATION

The SS_x pin allows a Synchronous Slave mode. The SPI must be in Slave mode with the SS_x pin control enabled (SSPxCON1<3:0> = 04h). When the SS_x pin is low, transmission and reception are enabled and the SDO_x pin is driven. When the SS_x pin goes high, the SDO_x 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.

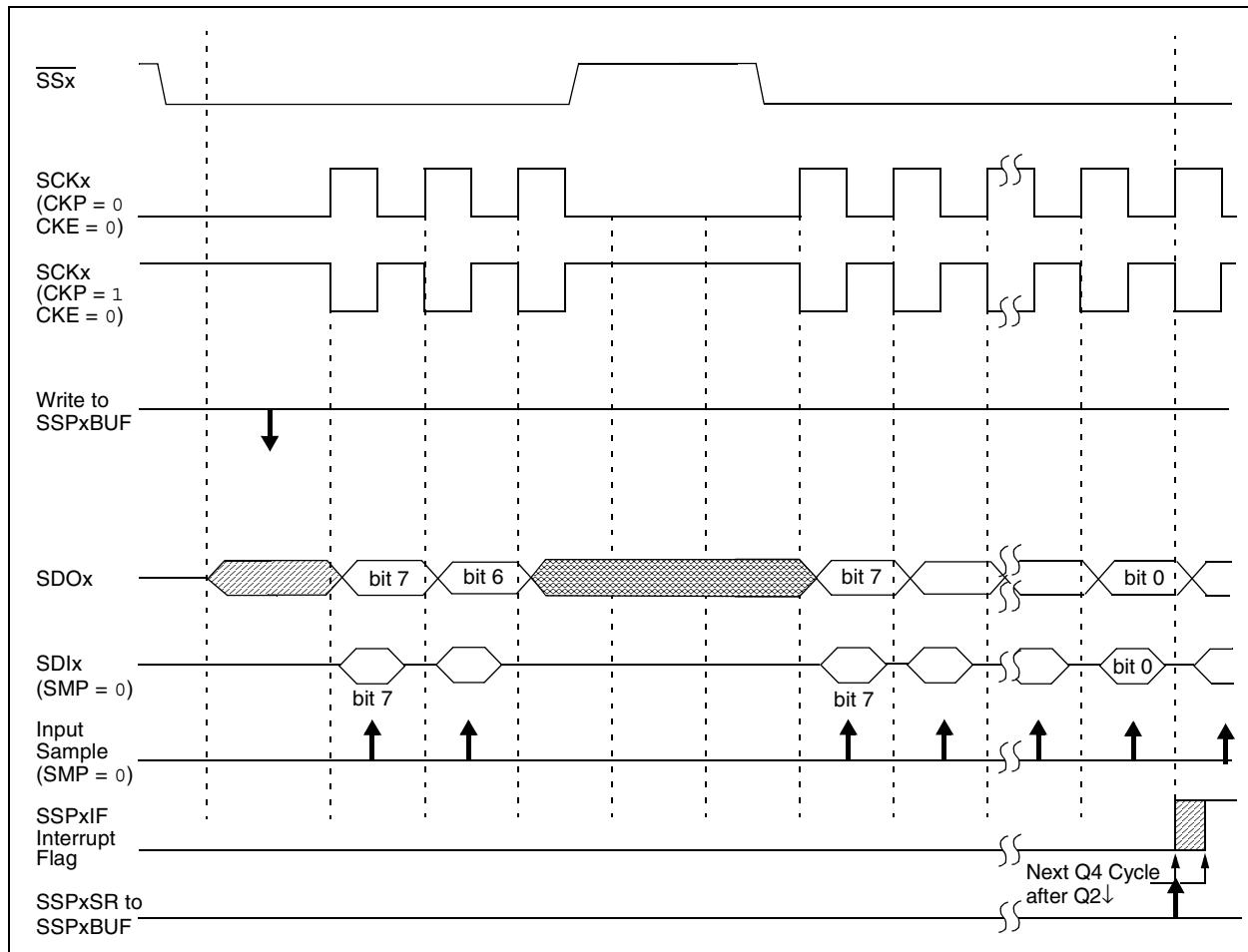
Note 1: When the SPI is in Slave mode with SS_x pin control enabled (SSPxCON1<3:0> = 0100), the SPI module will reset if the SS_x pin is set to VDD.

2: If the SPI is used in Slave mode with CKE set, then the SS_x 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 SS_x pin to a high level or clearing the SSPEN bit.

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

FIGURE 19-4: SLAVE SYNCHRONIZATION WAVEFORM



PIC18F87J11 FAMILY

FIGURE 19-5: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

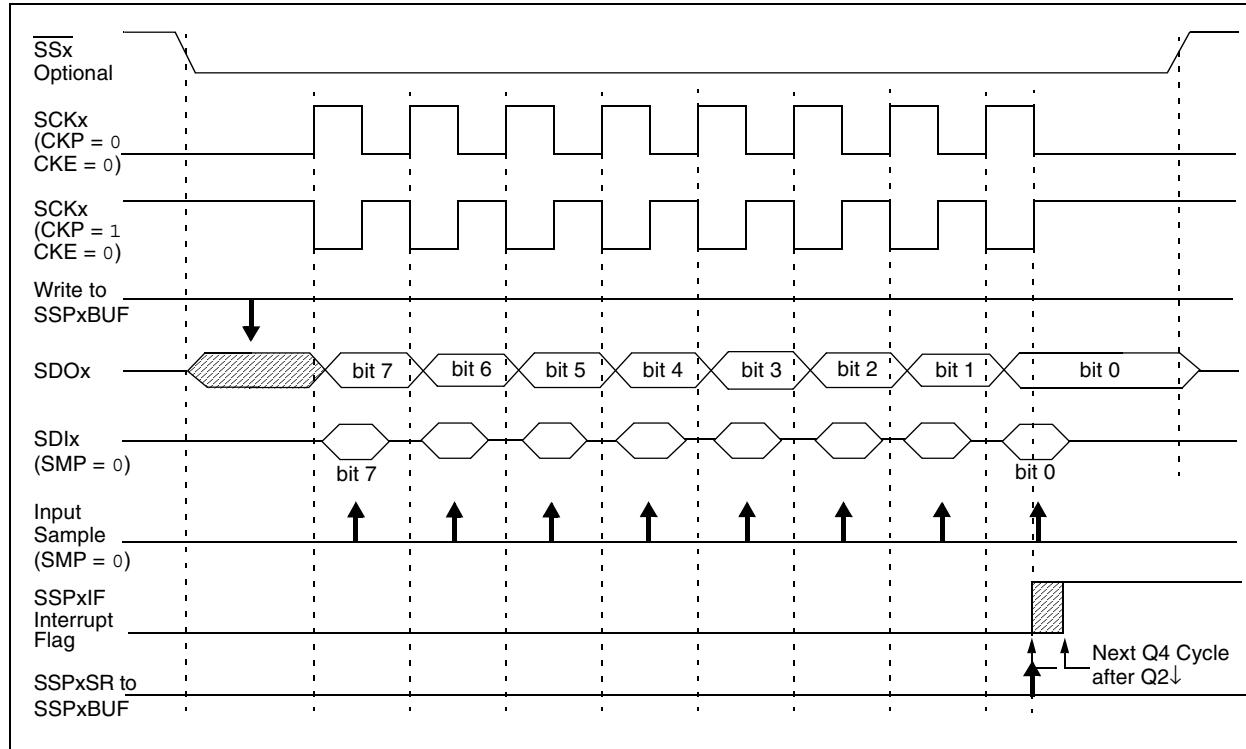
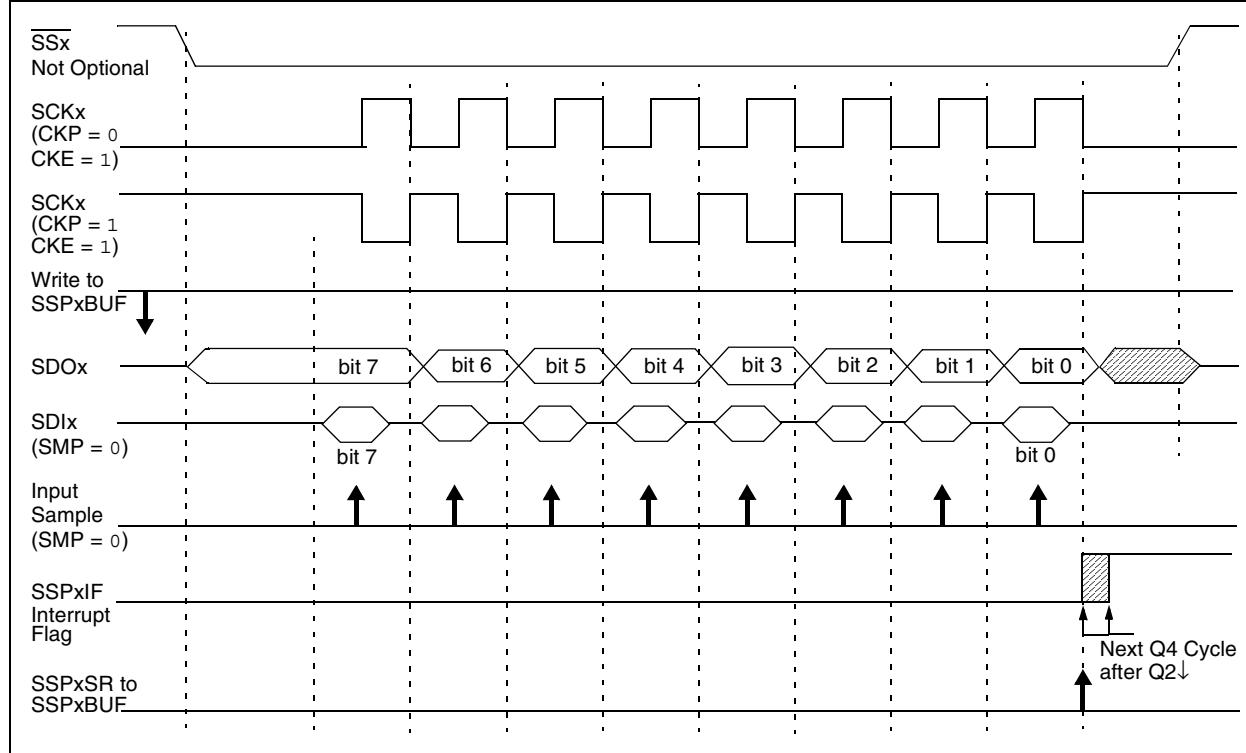


FIGURE 19-6: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



19.3.9 OPERATION IN POWER-MANAGED MODES

In SPI Master mode, module clocks may be operating at a different speed than when in full power mode; in the case of the Sleep mode, all clocks are halted.

In Idle modes, a clock is provided to the peripherals. That clock can be from the primary clock source, the secondary clock (Timer1 oscillator) or the INTOSC source. See **Section 2.3 “Clock Sources and Oscillator Switching”** for additional information.

In most cases, the speed that the master clocks SPI data is not important; however, this should be evaluated for each system.

If MSSP interrupts are enabled, they can wake the controller from Sleep mode, or one of the Idle modes, when the master completes sending data. If an exit from Sleep or Idle mode is not desired, MSSP interrupts should be disabled.

If the Sleep mode is selected, all module clocks are halted and the transmission/reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in any power-managed 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.

19.3.10 EFFECTS OF A RESET

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

19.3.11 BUS MODE COMPATIBILITY

Table 19-1 shows the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

TABLE 19-1: 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 an SMP bit which controls when the data is sampled.

19.3.12 SPI CLOCK SPEED AND MODULE INTERACTIONS

Because MSSP1 and MSSP2 are independent modules, they can operate simultaneously at different data rates. Setting the SSPM3:SSPM0 bits of the SSPxCON1 register determines the rate for the corresponding module.

An exception is when both modules use Timer2 as a time base in Master mode. In this instance, any changes to the Timer2 module's operation will affect both MSSP modules equally. If different bit rates are required for each module, the user should select one of the other three time base options for one of the modules.

PIC18F87J11 FAMILY

TABLE 19-2: REGISTERS ASSOCIATED WITH SPI OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	58
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	58
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	—	—	58
SSP1BUF	MSSP1 Receive Buffer/Transmit Register								56
SSPxCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	56, 59
SSPxSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	56, 59
SSP2BUF	MSSP2 Receive Buffer/Transmit Register								59
ODCON3 ⁽¹⁾	—	—	—	—	—	—	SPI2OD	SPI1OD	56

Legend: Shaded cells are not used by the MSSP module in SPI mode.

Note 1: Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.

19.4 I²C Mode

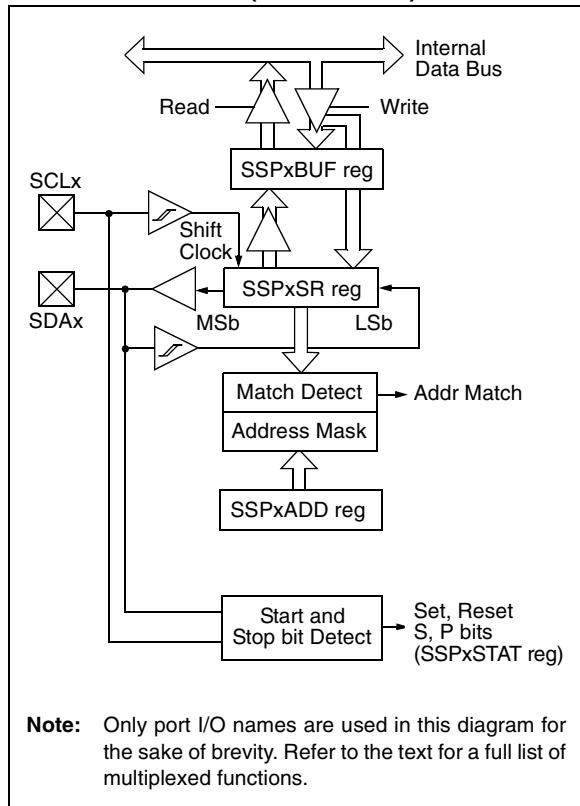
The MSSP module in I²C mode fully implements all master and slave functions (including general call support), and provides interrupts on Start and Stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit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

Two pins are used for data transfer:

- Serial Clock (SCLx) – RC3/SCK1/SCL1 or RD6/SCK2/SCL2
- Serial Data (SDAx) – RC4/SDI1/SDA1 or RD5/SDI2/SDA2

The user must configure these pins as inputs by setting the associated TRIS bits.

FIGURE 19-7: MSSP BLOCK DIAGRAM (I²C™ MODE)



19.4.1 REGISTERS

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

- MSSPx Control Register 1 (SSPxCON1)
- MSSPx Control Register 2 (SSPxCON2)
- MSSPx Status Register (SSPxSTAT)
- Serial Receive/Transmit Buffer Register (SSPxBUF)
- MSSPx Shift Register (SSPxSR) – Not directly accessible
- MSSPx Address Register (SSPxADD)
- I²C Slave Address Mask Register (SSPxMSK)

SSPxCON1, SSPxCON2 and SSPxSTAT are the control and status registers in I²C mode operation. The SSPxCON1 and SSPxCON2 registers are readable and writable. The lower 6 bits of the SSPxSTAT are read-only. The upper two bits of the SSPxSTAT are read/write.

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

SSPxADD contains the slave device address when the MSSP is configured in I²C Slave mode. When the MSSP is configured in Master mode, the lower seven bits of SSPxADD act as the Baud Rate Generator reload value.

SSPxMSK holds the slave address mask value when the module is configured for 7-bit Address Masking mode. While it is a separate register, it shares the same SFR address as SSPxADD; it is only accessible when the SSPM3:SSPM0 bits are specifically set to permit access. Additional details are provided in **Section 19.4.3.4 “7-Bit Address Masking Mode”**.

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

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

PIC18F87J11 FAMILY

REGISTER 19-3: SSPxSTAT: MSSPx STATUS REGISTER (I²C™ MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P ⁽¹⁾	S ⁽¹⁾	R/W ^(2,3)	UA	BF
bit 7	bit 0						

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

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⁾

1 = Indicates that a Stop bit has been detected last

0 = Stop bit was not detected last

bit 3 **S:** Start bit⁽¹⁾

1 = Indicates that a Start bit has been detected last

0 = Start bit was not detected last

bit 2 **R/W:** Read/Write Information bit^(2,3)

In Slave mode:

1 = Read

0 = Write

In Master mode:

1 = Transmit is in progress

0 = Transmit is not in progress

bit 1 **UA:** Update Address bit (10-Bit Slave mode only)

1 = Indicates that the user needs to update the address in the SSPxADD register

0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit

In Transmit mode:

1 = SSPxBUF is full

0 = SSPxBUF is empty

In Receive mode:

1 = SSPxBUF is full (does not include the ACK and Stop bits)

0 = SSPxBUF is empty (does not include the ACK and Stop bits)

Note 1: This bit is cleared on Reset and when SSPEN is cleared.

2: 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.

3: ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSPx is in Active mode.

REGISTER 19-4: SSPxCON1: MSSPx CONTROL REGISTER 1 (I²C™ MODE)

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 ⁽¹⁾	CKP	SSPM3 ⁽²⁾	SSPM2 ⁽²⁾	SSPM1 ⁽²⁾	SSPM0 ⁽²⁾
bit 7							bit 0

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

bit 7 **WCOL:** Write Collision Detect bit

In Master Transmit mode:

1 = A write to the SSPxBUF 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 SSPxBUF 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 SSPxBUF 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:** Master Synchronous Serial Port Enable bit⁽¹⁾

1 = Enables the serial port and configures the SDAx and SCLx pins as the serial port pins
0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** SCKx Release Control bit

In Slave mode:

1 = Releases 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:** Master Synchronous Serial Port Mode Select bits⁽²⁾

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)
1001 = Load SSPMSK register at SSPADD SFR address^(3,4)
1000 = I²C Master mode, clock = Fosc/(4 * (SSPxADD + 1))
0111 = I²C Slave mode, 10-bit address
0110 = I²C Slave mode, 7-bit address

Note 1: When enabled, the SDAx and SCLx pins must be configured as inputs.

2: Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.

3: When SSPM3:SSPM0 = 1001, any reads or writes to the SSPxADD SFR address actually accesses the SSPxMSK register.

4: This mode is only available when 7-Bit Address Masking mode is selected (MSSPMSK Configuration bit is '1').

PIC18F87J11 FAMILY

REGISTER 19-5: SSPxCON2: MSSPx CONTROL REGISTER 2 (I²C™ MASTER MODE)

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 ⁽¹⁾	ACKEN ⁽²⁾	RCEN ⁽²⁾	PEN ⁽²⁾	RSEN ⁽²⁾	SEN ⁽²⁾
bit 7	bit 0						

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

- bit 7 **GCEN:** General Call Enable bit
Unused in Master mode.
- 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⁾
1 = Not Acknowledge
0 = Acknowledge
- bit 4 **ACKEN:** Acknowledge Sequence Enable bit⁽²⁾
1 = Initiates Acknowledge sequence on SDAx and SCLx pins and transmit ACKDT data bit.
Automatically cleared by hardware.
0 = Acknowledge sequence Idle
- bit 3 **RCEN:** Receive Enable bit (Master Receive mode only)⁽²⁾
1 = Enables Receive mode for I²C
0 = Receive Idle
- bit 2 **PEN:** Stop Condition Enable bit⁽²⁾
1 = Initiates Stop condition on SDAx and SCLx pins. Automatically cleared by hardware.
0 = Stop condition Idle
- bit 1 **RSEN:** Repeated Start Condition Enable bit⁽²⁾
1 = Initiates Repeated Start condition on SDAx and SCLx pins. Automatically cleared by hardware.
0 = Repeated Start condition Idle
- bit 0 **SEN:** Start Condition Enable bit⁽²⁾
1 = Initiates Start condition on SDAx and SCLx pins. Automatically cleared by hardware.
0 = Start condition Idle

Note 1: Value that will be transmitted when the user initiates an Acknowledge sequence at the end of a receive.

2: If the I²C module is active, these bits may not be set (no spooling) and the SSPxBUF may not be written (or writes to the SSPxBUF are disabled).

REGISTER 19-6: SSPxCON2: MSSPx CONTROL REGISTER 2 (I²C™ SLAVE MODE)

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	ADMSK5	ADMSK4	ADMSK3	ADMSK2	ADMSK1	SEN ⁽¹⁾
bit 7	bit 0						

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

bit 7	GCEN: General Call Enable bit 1 = Enables interrupt when a general call address (0000h) is received in the SSPSR 0 = General call address disabled
bit 6	ACKSTAT: Acknowledge Status bit Unused in Slave mode.
bit 5-2	ADMSK5:ADMSK2: Slave Address Mask Select bits (5-Bit Address Masking mode) 1 = Masking of corresponding bits of SSPxADD enabled 0 = Masking of corresponding bits of SSPxADD disabled
bit 1	ADMSK1: Slave Address Least Significant bit(s) Mask Select bit <u>In 7-Bit Addressing mode:</u> 1 = Masking of SSPxADD<1> only enabled 0 = Masking of SSPxADD<1> only disabled <u>In 10-Bit Addressing mode:</u> 1 = Masking of SSPxADD<1:0> enabled 0 = Masking of SSPxADD<1:0> disabled
bit 0	SEN: Stretch Enable bit ⁽¹⁾ 1 = Clock stretching is enabled for both slave transmit and slave receive (stretch enabled) 0 = Clock stretching is disabled

Note 1: If the I²C module is active, this bit may not be set (no spooling) and the SSPBUF may not be written (or writes to the SSPBUF are disabled).

REGISTER 19-7: SSPxMSK: I²C™ SLAVE ADDRESS MASK REGISTER (7-BIT MASKING MODE)⁽¹⁾

| R/W-1 |
|-------|-------|-------|-------|-------|-------|-------|---------------------|
| MSK7 | MSK6 | MSK5 | MSK4 | MSK3 | MSK2 | MSK1 | MSK0 ⁽²⁾ |
| bit 7 | bit 0 | | | | | | |

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

bit 7-0	MSK7:MSK0: Slave Address Mask Select bit 1 = Masking of corresponding bit of SSPxADD enabled 0 = Masking of corresponding bit of SSPxADD disabled
---------	--

Note 1: This register shares the same SFR address as SSPxADD, and is only addressable in select MSSPx operating modes. See **Section 19.4.3.4 "7-Bit Address Masking Mode"** for more details.

2: MSK0 is not used as a mask bit in 7-bit addressing.

19.4.2 OPERATION

The MSSP module functions are enabled by setting the MSSP Enable bit, SSPEN (SSPxCON1<5>).

The SSPxCON1 register allows control of the I²C operation. Four mode selection bits (SSPxCON1<3:0>) allow one of the following I²C modes to be selected:

- I²C Master mode, clock
- 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 SCLx and SDAx pins to be open-drain, provided these pins are programmed as inputs by setting the appropriate TRISC or TRISD bits. To ensure proper operation of the module, pull-up resistors must be provided externally to the SCLx and SDAx pins.

19.4.3 SLAVE MODE

In Slave mode, the SCLx and SDAx 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. Address masking will allow the hardware to generate an interrupt for more than one address (up to 31 in 7-bit addressing and up to 63 in 10-bit addressing). 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 (ACK) pulse and load the SSPxBUF register with the received value currently in the SSPxSR register.

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

- The Buffer Full bit, BF (SSPxSTAT<0>), was set before the transfer was received.
- The overflow bit, SSPOV (SSPxCON1<6>), was set before the transfer was received.

In this case, the SSPxSR register value is not loaded into the SSPxBUF, but bit SSPxIF is set. The BF bit is cleared by reading the SSPxBUF register, while bit SSPOV is cleared through software.

The SCLx 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.

19.4.3.1 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 SSPxSR register. All incoming bits are sampled with the rising edge of the clock (SCLx) line. The value of register, SSPxSR<7:1>, is compared to the value of the SSPxADD register. The address is compared on the falling edge of the eighth clock (SCLx) pulse. If the addresses match and the BF and SSPOV bits are clear, the following events occur:

1. The SSPxSR register value is loaded into the SSPxBUF register.
2. The Buffer Full bit, BF, is set.
3. An ACK pulse is generated.
4. The MSSP Interrupt Flag bit, SSPxIF, is set (and interrupt is generated, if enabled) on the falling edge of the ninth SCLx pulse.

In 10-Bit Addressing mode, two address bytes need to be received by the slave. The five Most Significant bits (MSbs) of the first address byte specify if this is a 10-bit address. Bit R/W (SSPxSTAT<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 addressing is as follows, with steps 7 through 9 for the slave-transmitter:

1. Receive first (high) byte of address (bits SSPxIF, BF and UA are set on address match).
2. Update the SSPxADD register with second (low) byte of address (clears bit UA and releases the SCLx line).
3. Read the SSPxBUF register (clears bit, BF) and clear flag bit, SSPxIF.
4. Receive second (low) byte of address (bits SSPxIF, BF and UA are set).
5. Update the SSPxADD register with the first (high) byte of address. If match releases SCLx line, this will clear bit UA.
6. Read the SSPxBUF register (clears bit BF) and clear flag bit SSPxIF.
7. Receive Repeated Start condition.
8. Receive first (high) byte of address (bits SSPxIF and BF are set).
9. Read the SSPxBUF register (clears bit BF) and clear flag bit, SSPxIF.

19.4.3.2 Address Masking Modes

Masking an address bit causes that bit to become a "don't care". When one address bit is masked, two addresses will be Acknowledged and cause an interrupt. It is possible to mask more than one address bit at a time, which greatly expands the number of addresses Acknowledged.

The I²C Slave behaves the same way whether address masking is used or not. However, when address masking is used, the I²C slave can Acknowledge multiple addresses and cause interrupts. When this occurs, it is necessary to determine which address caused the interrupt by checking the SSPxBUF.

The PIC18F87J11 Family of devices is capable of using two different Address Masking modes in I²C Slave operation: 5-Bit Address Masking and 7-Bit Address Masking. The Masking mode is selected at device configuration using the MSSPMSK Configuration bit. The default device configuration is 7-Bit Address Masking.

Both Masking modes, in turn, support address masking of 7-bit and 10-bit addresses. The combination of Masking modes and addresses provide different ranges of Acknowledgable addresses for each combination.

While both Masking modes function in roughly the same manner, the way they use address masks are different.

19.4.3.3 5-Bit Address Masking Mode

As the name implies, 5-Bit Address Masking mode uses an address mask of up to 5 bits to create a range of addresses to be Acknowledged, using bits 5 through 1 of the incoming address. This allows the module to

Acknowledge up to 31 addresses when using 7-bit addressing, or 63 addresses with 10-bit addressing (see Example 19-2). This Masking mode is selected when the MSSPMSK Configuration bit is programmed ('0').

The address mask in this mode is stored in the SSPxCON2 register, which stops functioning as a control register in I²C Slave mode (Register 19-6). In 7-Bit Address Masking mode, address mask bits, ADMSK<5:1> (SSPxCON2<5:1>), mask the corresponding address bits in the SSPxADD register. For any ADMSK bits that are set (ADMSK<n> = 1), the corresponding address bit is ignored (SSPxADD<n> = x). For the module to issue an address Acknowledge, it is sufficient to match only on addresses that do not have an active address mask.

In 10-Bit Address Masking mode, bits ADMSK<5:2> mask the corresponding address bits in the SSPxADD register. In addition, ADMSK1 simultaneously masks the two LSbs of the address (SSPxADD<1:0>). For any ADMSK bits that are active (ADMSK<n> = 1), the corresponding address bit is ignored (SPxADD<n> = x). Also note, that although in 10-Bit Address Masking mode, the upper address bits reuse part of the SSPxADD register bits. The address mask bits do not interact with those bits; they only affect the lower address bits.

Note 1: ADMSK1 masks the two Least Significant bits of the address.

2: The two Most Significant bits of the address are not affected by address masking.

EXAMPLE 19-2: ADDRESS MASKING EXAMPLES IN 5-BIT MASKING MODE

7-Bit Addressing:

SSPADD<7:1> = A0h (1010000) (SSPADD<0> is assumed to be 0)

ADMSK<5:1> = 00111

Addresses Acknowledged: A0h, A2h, A4h, A6h, A8h, AAh, ACh, AEh

10-Bit Addressing:

SSPADD<7:0> = A0h (10100000) (The two MSb of the address are ignored in this example, since they are not affected by masking)

ADMSK<5:1> = 00111

Addresses Acknowledged: A0h, A1h, A2h, A3h, A4h, A5h, A6h, A7h, A8h, A9h, AAh, ABh, ACh, ADh, AEh, AFh

PIC18F87J11 FAMILY

19.4.3.4 7-Bit Address Masking Mode

Unlike 5-bit masking, 7-Bit Address Masking mode uses a mask of up to 8 bits (in 10-bit addressing) to define a range of addresses than can be Acknowledged, using the lowest bits of the incoming address. This allows the module to Acknowledge up to 127 different addresses with 7-bit addressing, or 255 with 10-bit addressing (see Example 19-3). This mode is the default configuration of the module, and is selected when MSSPMSK is unprogrammed ('1').

The address mask for 7-Bit Address Masking mode is stored in the SSPxMSK register, instead of the SSPxCON2 register. SSPxMSK is a separate hardware register within the module, but it is not directly addressable. Instead, it shares an address in the SFR space with the SSPxADD register. To access the SSPxMSK register, it is necessary to select MSSP mode, '1001' (SSPCON1<3:0> = 1001), and then read or write to the location of SSPxADD.

To use 7-Bit Address Masking mode, it is necessary to initialize SSPxMSK with a value before selecting the I²C Slave Addressing mode. Thus, the required sequence of events is:

1. Select SSPxMSK Access mode (SSPxCON2<3:0> = 1001).
2. Write the mask value to the appropriate SSPADD register address (FC8h for MSSP1, F6Eh for MSSP2).
3. Set the appropriate I²C Slave mode (SSPxCON2<3:0> = 0111 for 10-bit addressing, 0110 for 7-bit addressing).

Setting or clearing mask bits in SSPxMSK behaves in the opposite manner of the ADMASK bits in 5-Bit Address Masking mode. That is, clearing a bit in SSPxMSK causes the corresponding address bit to be masked; setting the bit requires a match in that position. SSPxMSK resets to all '1's upon any Reset condition and, therefore, has no effect on the standard MSSP operation until written with a mask value.

With 7-bit addressing, SSPxMSK<7:1> bits mask the corresponding address bits in the SSPxADD register. For any SSPxMSK bits that are active (SSPxMSK<n> = 0), the corresponding SSPxADD address bit is ignored (SSPxADD<n> = x). For the module to issue an address Acknowledge, it is sufficient to match only on addresses that do not have an active address mask.

With 10-bit addressing, SSPxMSK<7:0> bits mask the corresponding address bits in the SSPxADD register. For any SSPxMSK bits that are active (= 0), the corresponding SSPxADD address bit is ignored (SSPxADD<n> = x).

Note: The two Most Significant bits of the address are not affected by address masking.

EXAMPLE 19-3: ADDRESS MASKING EXAMPLES IN 7-BIT MASKING MODE

7-Bit Addressing:

SSPxADD<7:1> = 1010 000

SSPxMSK<7:1> = 1111 001

Addresses Acknowledged = A8h, A6h, A4h, A0h

10-Bit Addressing:

SSPxADD<7:0> = 1010 0000 (The two MSb are ignored in this example since they are not affected)

SSPxMSK<5:1> = 1111 0

Addresses Acknowledged = A8h, A6h, A4h, A0h

19.4.3.5 Reception

When the R/W bit of the address byte is clear and an address match occurs, the R/W bit of the SSPxSTAT register is cleared. The received address is loaded into the SSPxBUF register and the SDAx line is held low (ACK).

When the address byte overflow condition exists, then the no Acknowledge (ACK) pulse is given. An overflow condition is defined as either bit, BF (SSPxSTAT<0>), is set or bit, SSPOV (SSPxCON1<6>), is set.

An MSSP interrupt is generated for each data transfer byte. The interrupt flag bit, SSPxIF, must be cleared in software. The SSPxSTAT register is used to determine the status of the byte.

If SEN is enabled (SSPxCON2<0> = 1), SCLx will be held low (clock stretch) following each data transfer. The clock must be released by setting bit, CKP (SSPxCON1<4>). See **Section 19.4.4 “Clock Stretching”** for more details.

19.4.3.6 Transmission

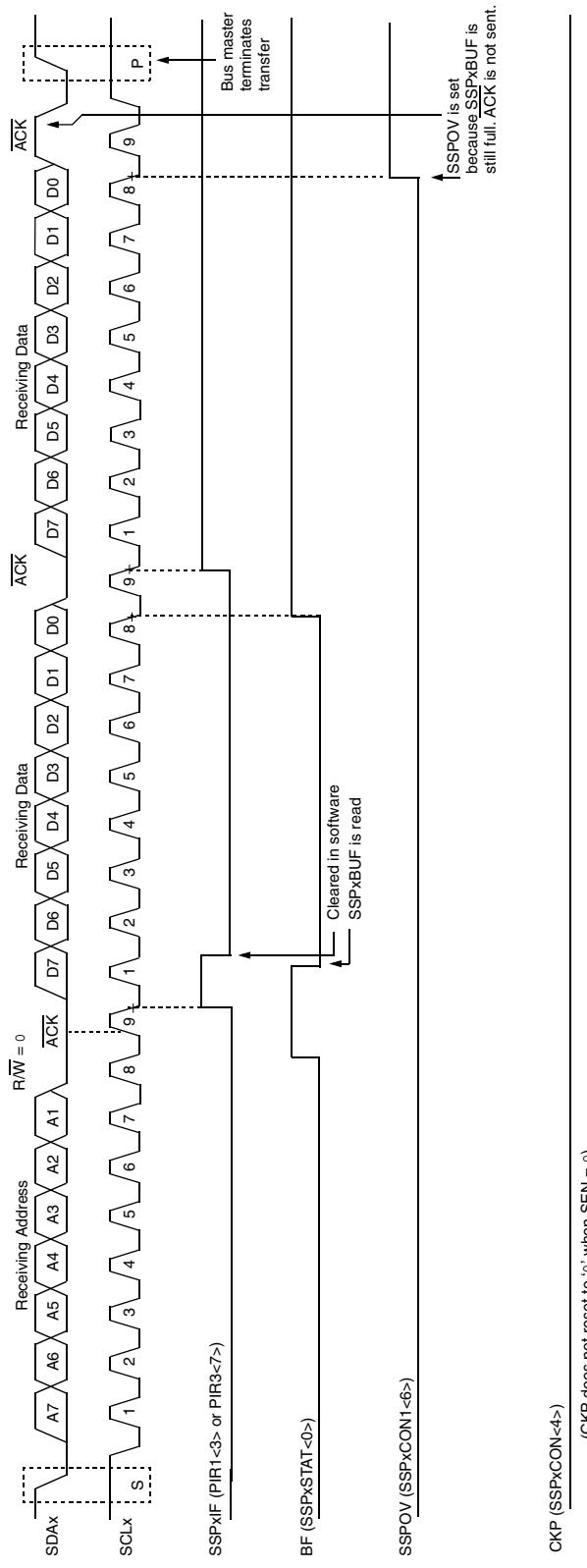
When the R/W bit of the incoming address byte is set and an address match occurs, the R/W bit of the SSPxSTAT register is set. The received address is loaded into the SSPxBUF register. The ACK pulse will be sent on the ninth bit and pin SCLx is held low regardless of SEN (see **Section 19.4.4 “Clock Stretching”** for more details). 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 SSPxBUF register which also loads the SSPxSR register. Then, pin SCLx should be enabled by setting bit, CKP (SSPxCON1<4>). The eight data bits are shifted out on the falling edge of the SCLx input. This ensures that the SDAx signal is valid during the SCLx high time (Figure 19-10).

The ACK pulse from the master-receiver is latched on the rising edge of the ninth SCLx input pulse. If the SDAx line is high (not ACK), then the data transfer is complete. In this case, when the ACK is latched by the slave, the slave logic is reset (resets the SSPxSTAT register) and the slave monitors for another occurrence of the Start bit. If the SDAx line was low (ACK), the next transmit data must be loaded into the SSPxBUF register. Again, pin SCLx must be enabled by setting bit, CKP.

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

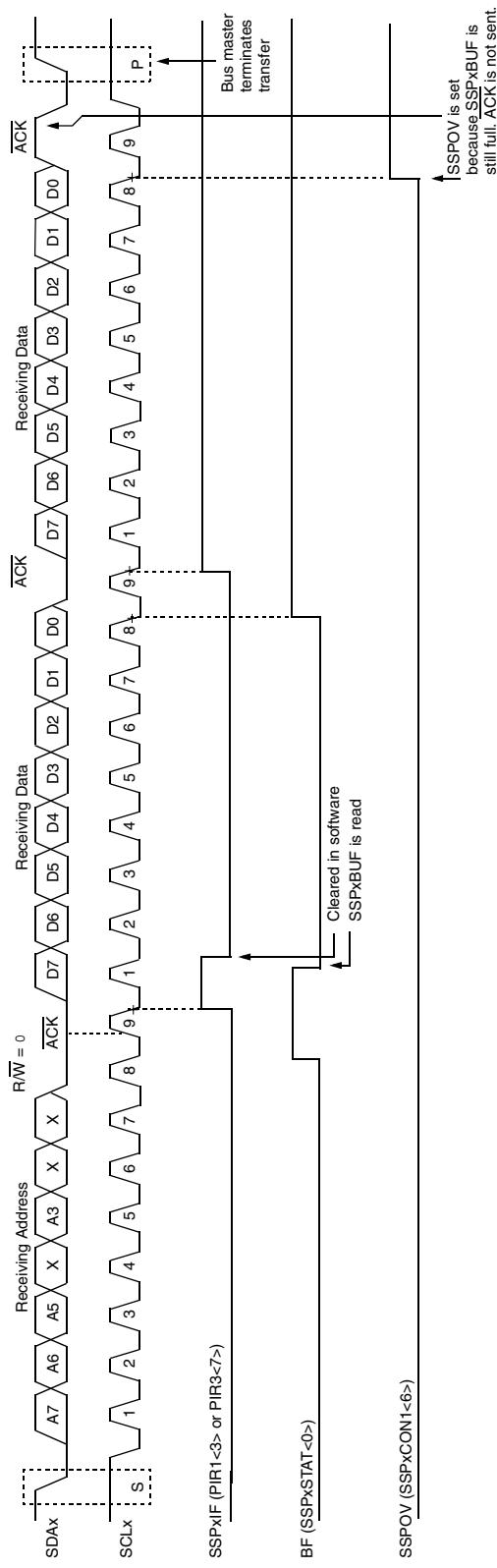
PIC18F87J11 FAMILY

FIGURE 19-8: I²C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 7-BIT ADDRESS)



PIC18F87J11 FAMILY

FIGURE 19-9: I²C™ SLAVE MODE TIMING WITH SEN = 0 AND ADMSK<5:1> = 01011 (RECEPTION, 7-BIT ADDRESS)

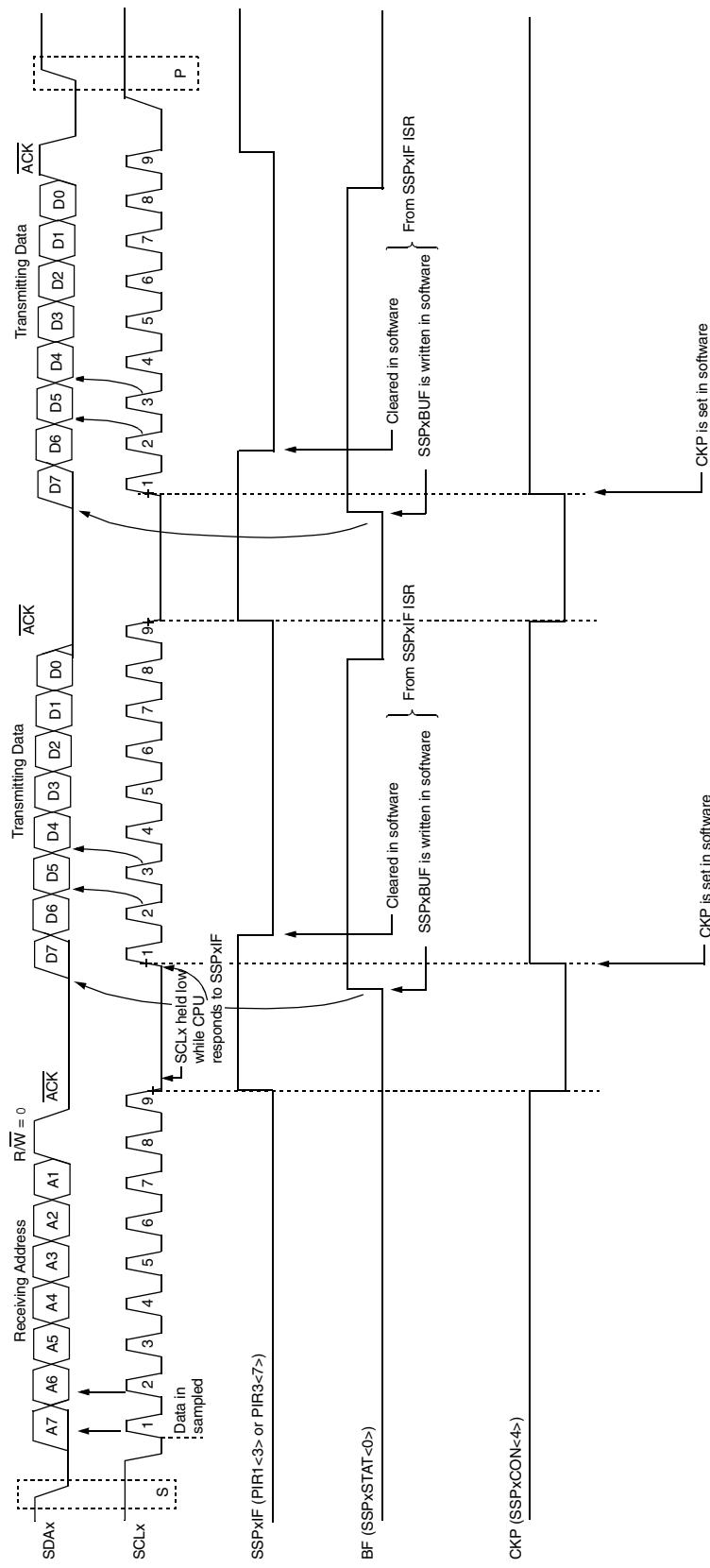


Note 1: x = Don't care (i.e., address bit can either be a '1' or a '0').

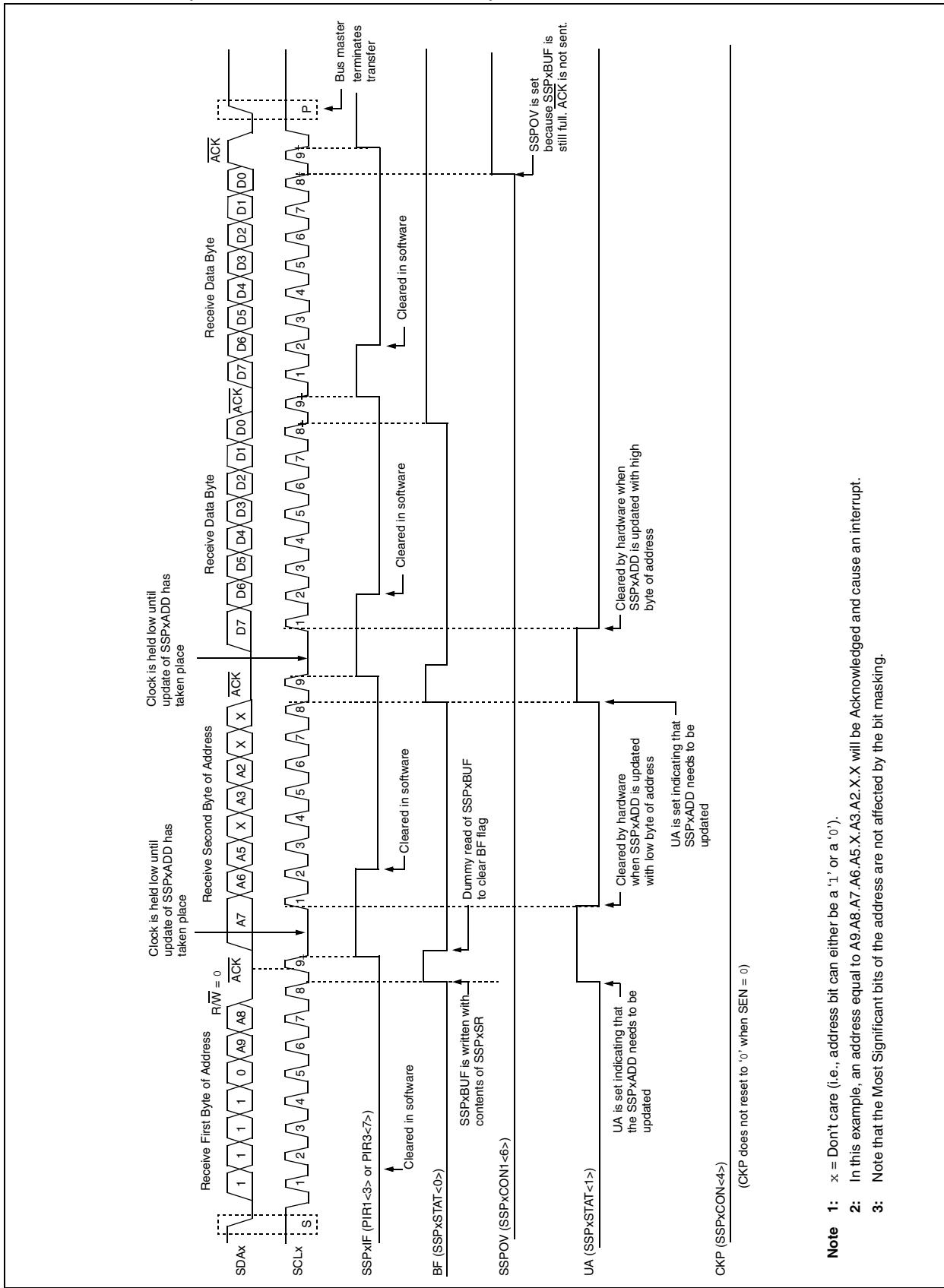
1: $x =$ Don't care (i.e., address bit can either be a '1' or a '0').
2: In this example, an address equal to A7.A6.A5.XA3.X.X will be Acknowledged and cause an interrupt.

PIC18F87J11 FAMILY

FIGURE 19-10: I²C™ SLAVE MODE TIMING (TRANSMISSION, 7-BIT ADDRESS)



**FIGURE 19-11: I²C™ SLAVE MODE TIMING WITH SEN = 0 AND ADMASK<5:1> = 01001
(RECEPTION, 10-BIT ADDRESS)**



PIC18F87J11 FAMILY

FIGURE 19-12: I²C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 10-BIT ADDRESS)

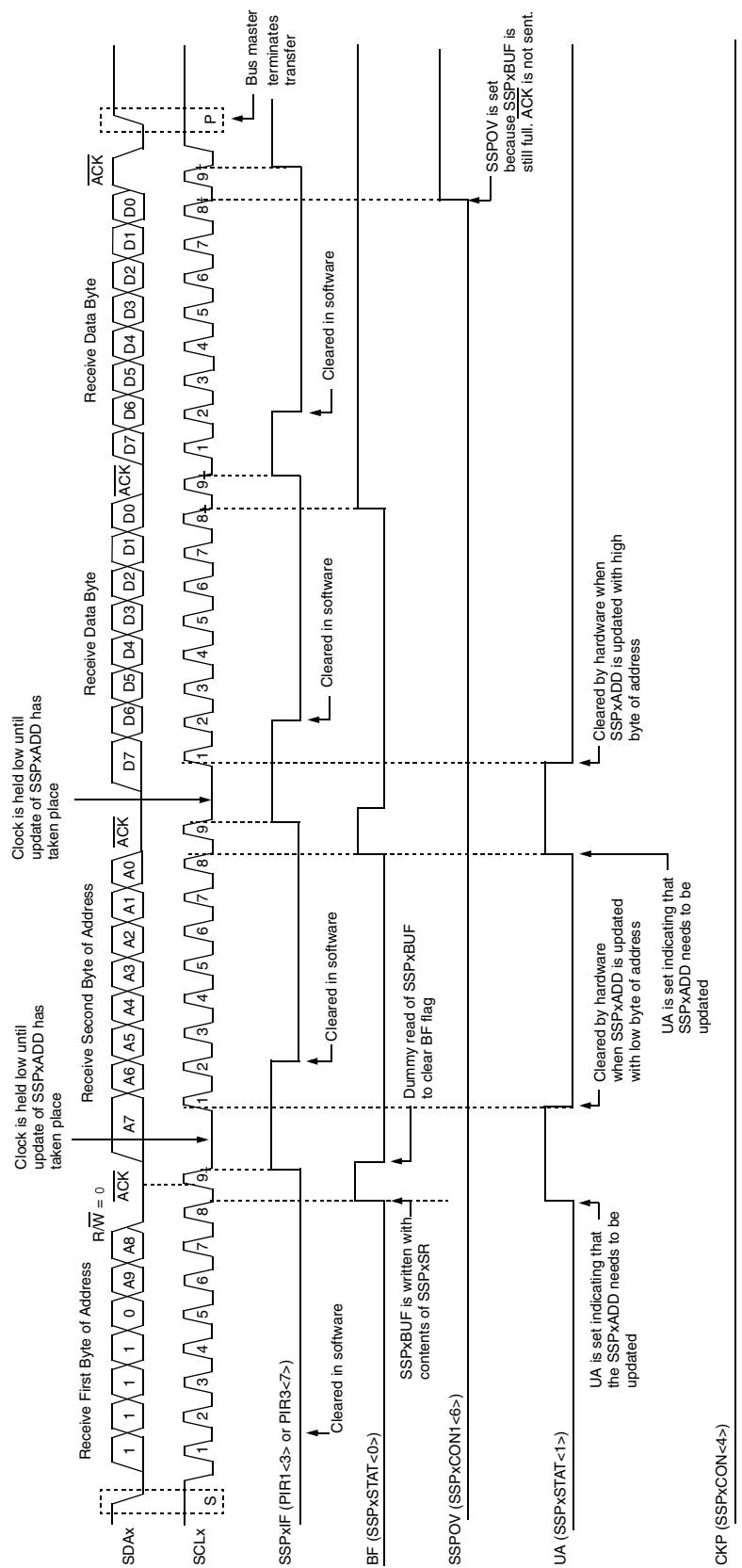
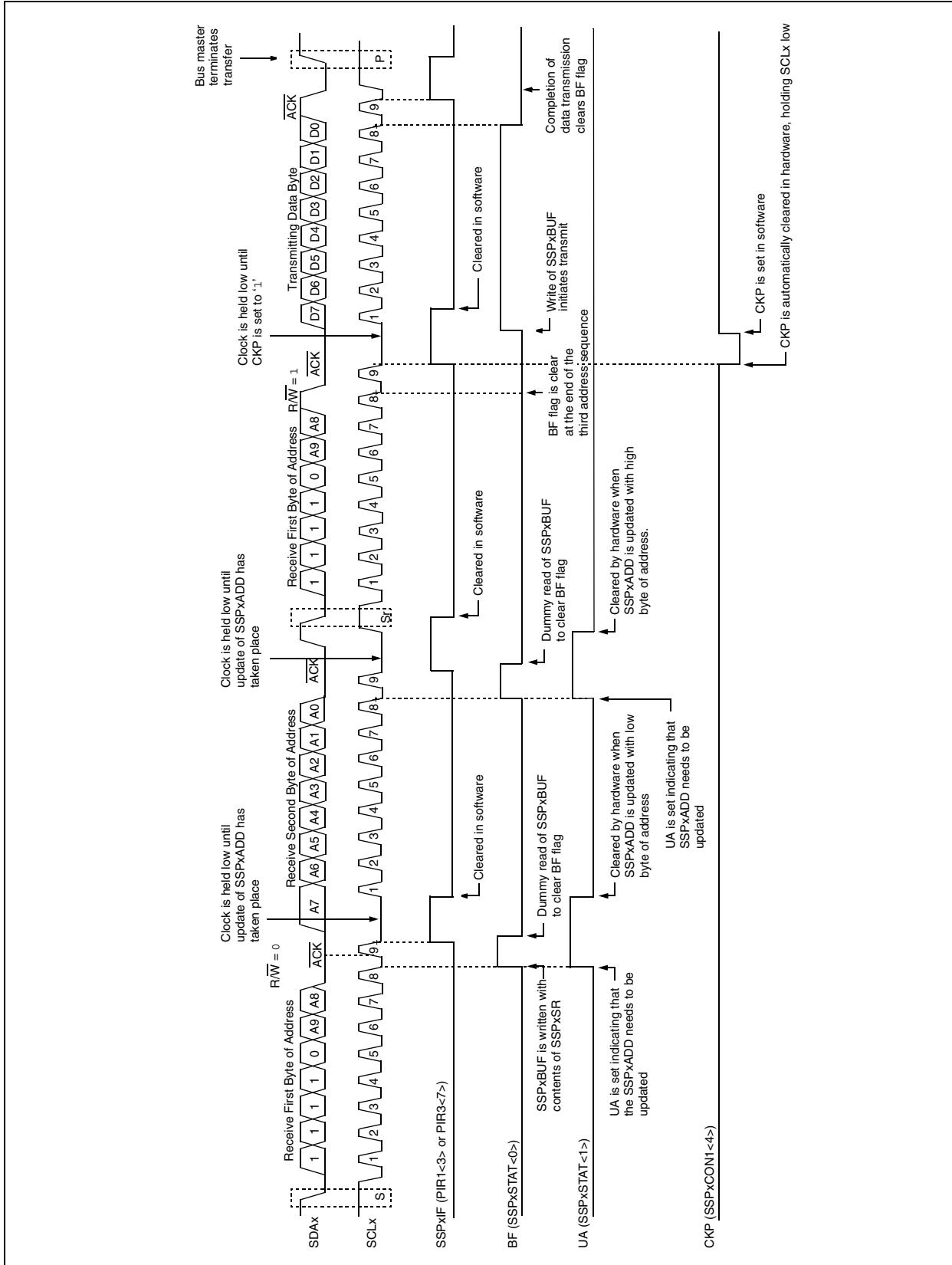


FIGURE 19-13: I²C™ SLAVE MODE TIMING (TRANSMISSION, 10-BIT ADDRESS)



PIC18F87J11 FAMILY

19.4.4 CLOCK STRETCHING

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

The SEN bit (SSPxCON2<0>) allows clock stretching to be enabled during receives. Setting SEN will cause the SCLx pin to be held low at the end of each data receive sequence.

19.4.4.1 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 SSPxCON1 register is automatically cleared, forcing the SCLx output to be held low. The CKP bit being cleared to '0' will assert the SCLx line low. The CKP bit must be set in the user's ISR before reception is allowed to continue. By holding the SCLx line low, the user has time to service the ISR and read the contents of the SSPxBUF before the master device can initiate another receive sequence. This will prevent buffer overruns from occurring (see Figure 19-15).

- Note 1:** If the user reads the contents of the SSPxBUF 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.

19.4.4.2 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 SSPxADD. 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 SSPxADD register before the falling edge of the ninth clock occurs, and if the user hasn't cleared the BF bit by reading the SSPxBUF 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.

19.4.4.3 Clock Stretching for 7-Bit Slave Transmit Mode

The 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 SCLx line low, the user has time to service the ISR and load the contents of the SSPxBUF before the master device can initiate another transmit sequence (see Figure 19-10).

- Note 1:** If the user loads the contents of SSPxBUF, 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.

19.4.4.4 Clock Stretching for 10-Bit Slave Transmit Mode

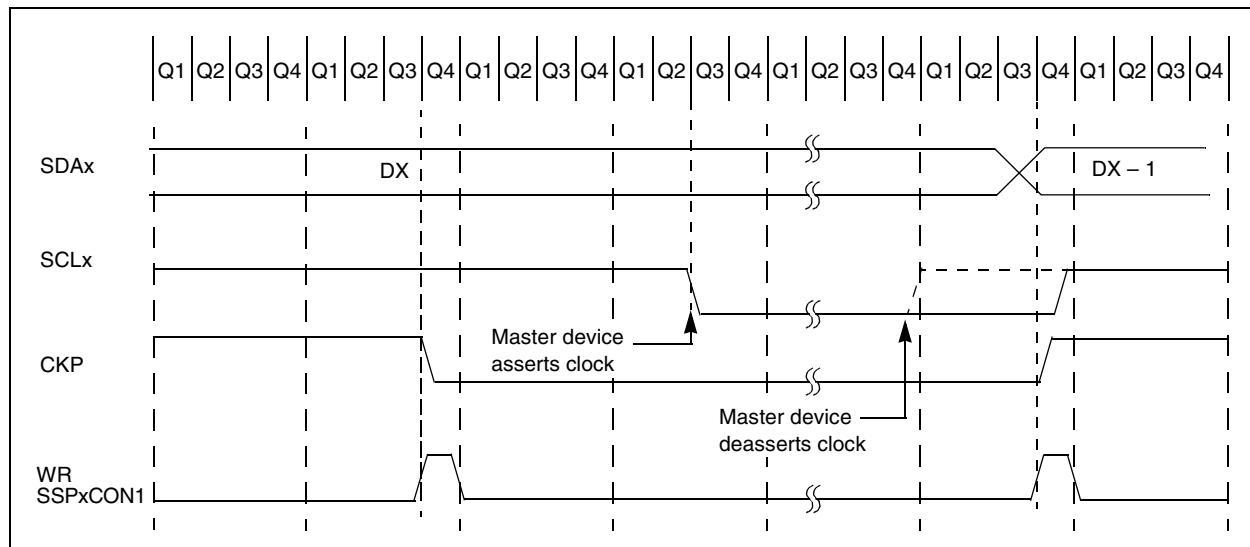
In 10-Bit Slave Transmit mode, clock stretching is controlled during the first two address sequences by the state of the UA bit, just as it is in 10-Bit Slave Receive mode. The first two addresses are followed by a third address sequence, which contains the high-order bits of the 10-bit address and the 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 19-13).

19.4.4.5 Clock Synchronization and the CKP bit

When the CKP bit is cleared, the SCL_x output is forced to '0'. However, clearing the CKP bit will not assert the SCL_x output low until the SCL_x output is already sampled low. Therefore, the CKP bit will not assert the SCL_x line until an external I²C master device has

already asserted the SCL_x line. The SCL_x output will remain low until the CKP bit is set and all other devices on the I²C bus have deasserted SCL_x. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL_x (see Figure 19-14).

FIGURE 19-14: CLOCK SYNCHRONIZATION TIMING



PIC18F87J11 FAMILY

FIGURE 19-15: I²C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 7-BIT ADDRESS)

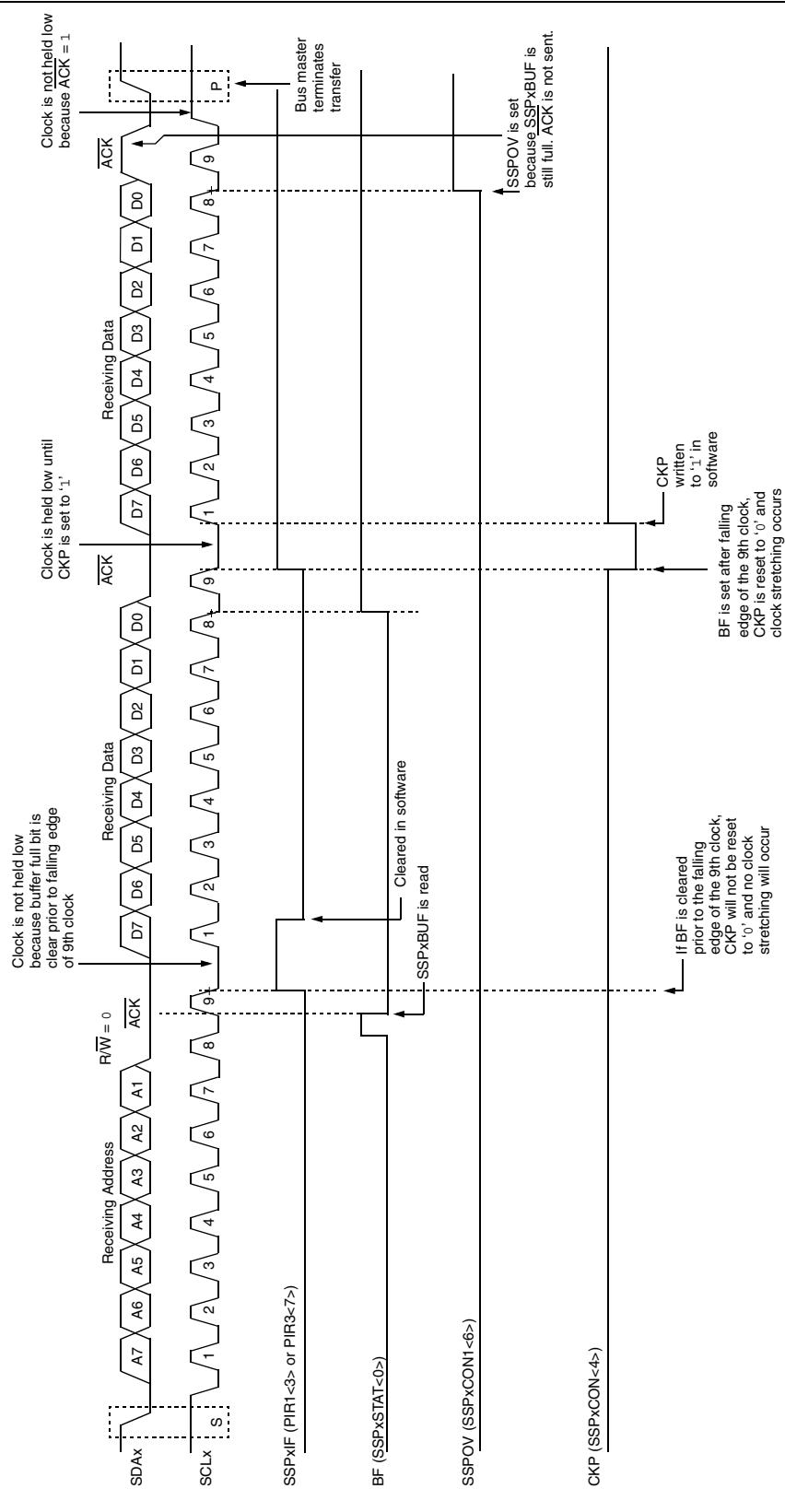
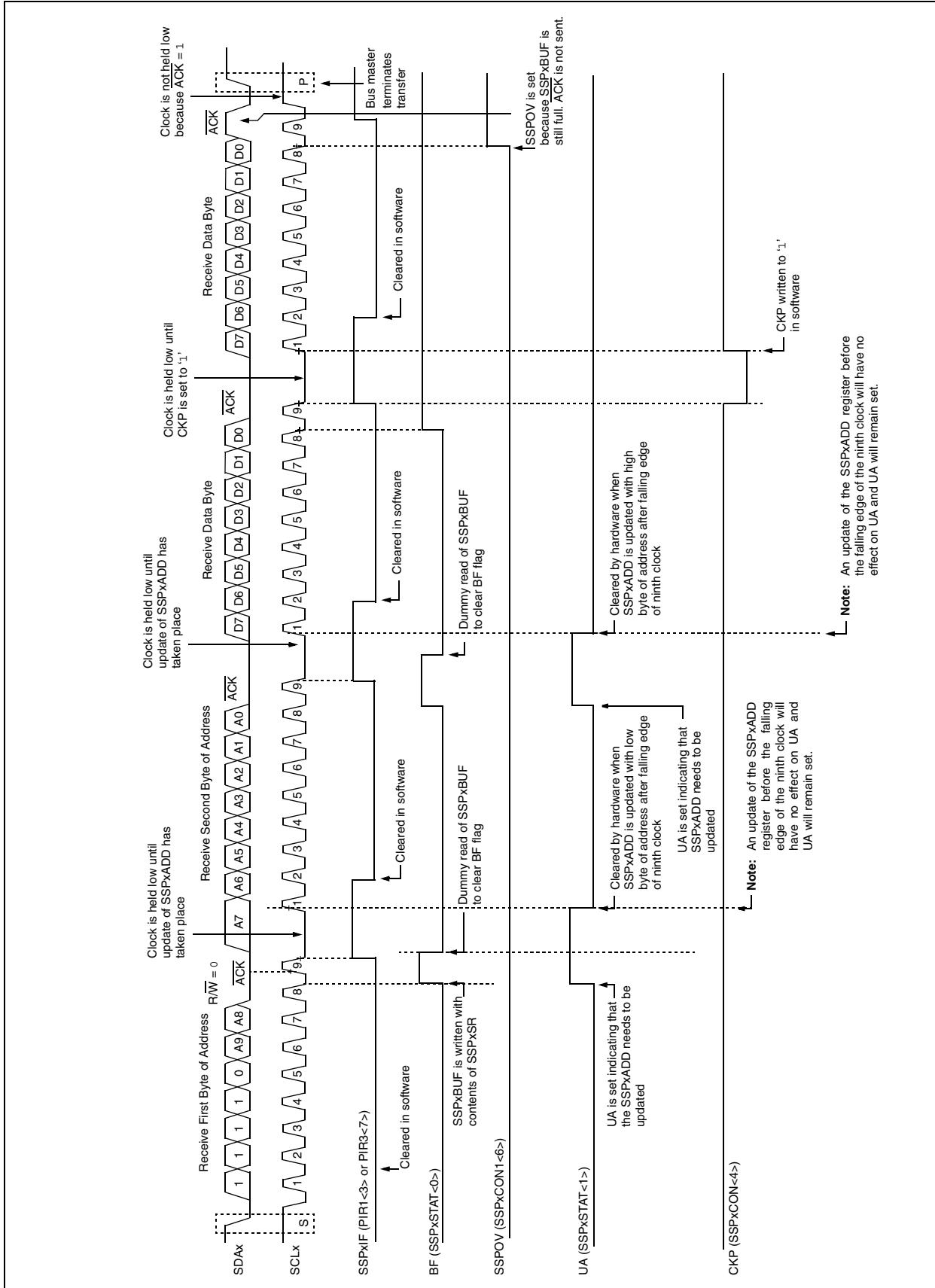


FIGURE 19-16: I²C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 10-BIT ADDRESS)



PIC18F87J11 FAMILY

19.4.5 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.

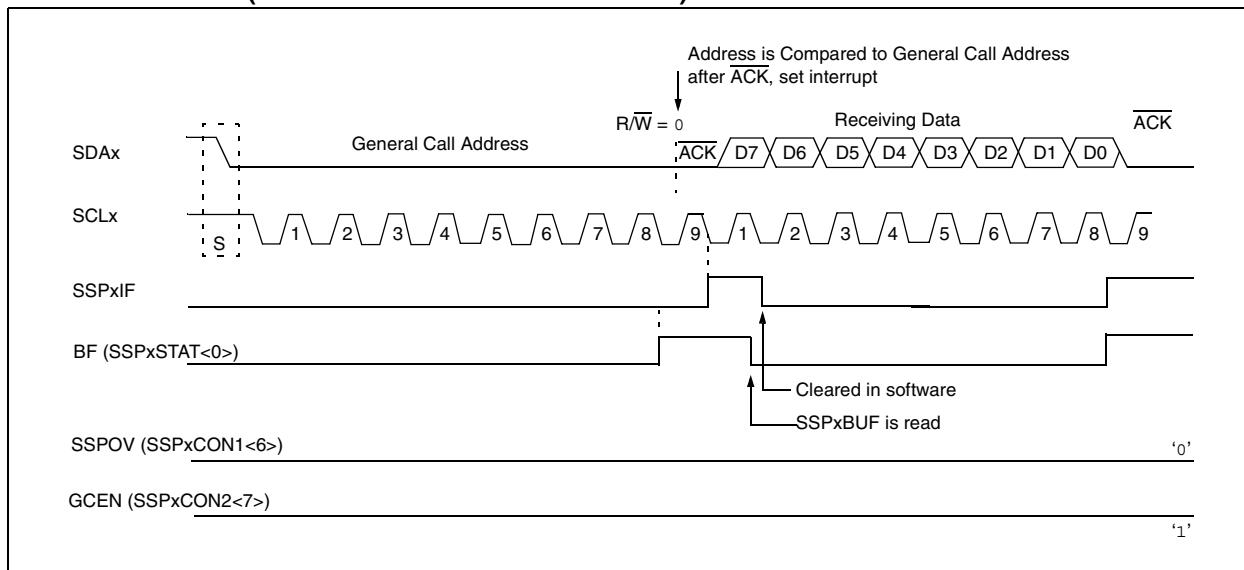
The general call address is recognized when the General Call Enable bit, GCEN, is enabled (SSPxCON2<7> set). Following a Start bit detect, 8 bits are shifted into the SSPxSR and the address is compared against the SSPxADD. It is also compared to the general call address and fixed in hardware.

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

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

In 10-Bit Addressing mode, the SSPxADD is required to be updated for the second half of the address to match and the UA bit is set (SSPxSTAT<1>). If the general call address is sampled when the GCEN bit is set, while the slave is configured in 10-Bit Addressing 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 19-17).

**FIGURE 19-17: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE
(7 OR 10-BIT ADDRESSING MODE)**



19.4.6 MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPM bits in SSPxCON1 and by setting the SSPEN bit. In Master mode, the SCL_x and SDAx lines are manipulated by the MSSP hardware if the TRIS bits are set.

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.

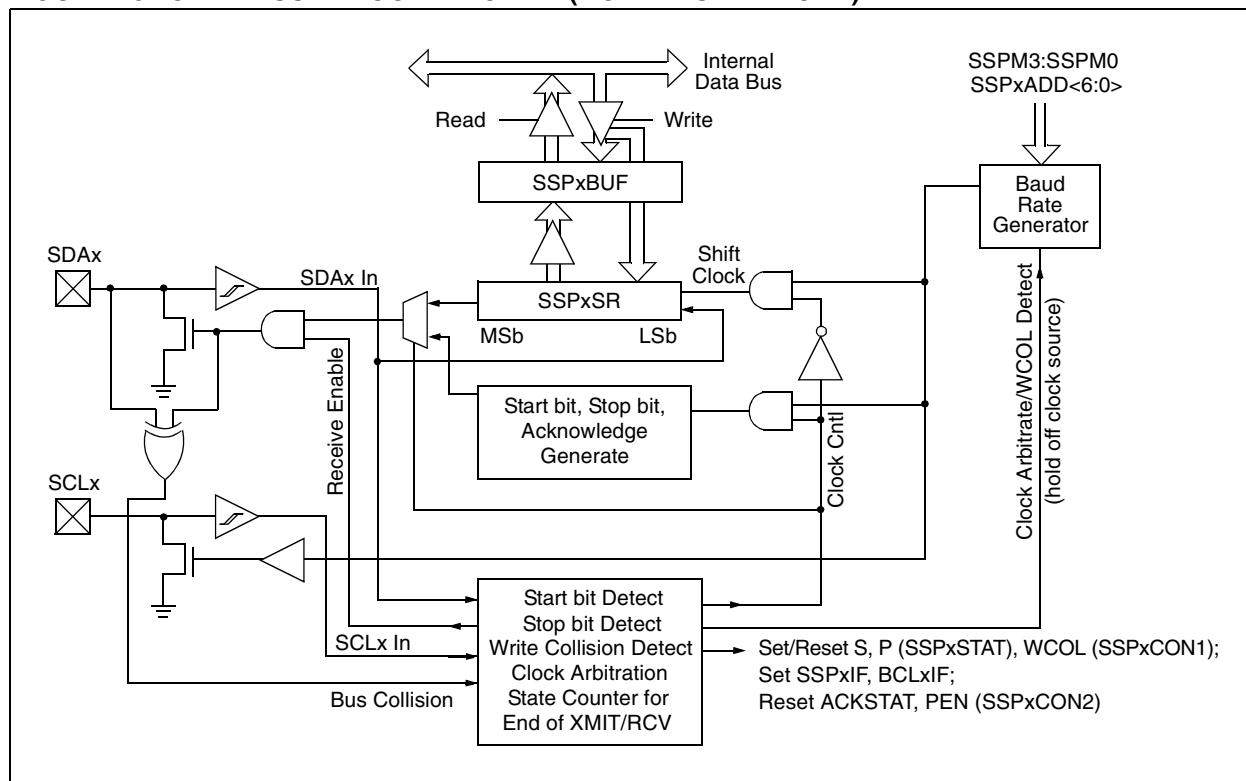
1. Assert a Start condition on SDAx and SCL_x.
2. Assert a Repeated Start condition on SDAx and SCL_x.
3. Write to the SSPxBUF register initiating transmission of data/address.
4. Configure the I²C port to receive data.
5. Generate an Acknowledge condition at the end of a received byte of data.
6. Generate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

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

The following events will cause the MSSP Interrupt Flag bit, SSPxFIF, to be set (and MSSP interrupt, if enabled):

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

FIGURE 19-18: MSSP BLOCK DIAGRAM (I²C™ MASTER MODE)



19.4.6.1 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 SDAx while SCLx 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 SDAx, while SCLx 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 SCLx clock frequency for either 100 kHz, 400 kHz or 1 MHz I²C operation. See **Section 19.4.7 “Baud Rate”** for more details.

A typical transmit sequence would go as follows:

1. The user generates a Start condition by setting the Start Enable bit, SEN (SSPxCON2<0>).
2. SSPxIF is set. The MSSP module will wait the required start time before any other operation takes place.
3. The user loads the SSPxBUF with the slave address to transmit.
4. Address is shifted out the SDAx 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 SSPxCN2 register (SSPxCON2<6>).
6. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
7. The user loads the SSPxBUF with eight bits of data.
8. Data is shifted out the SDAx 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 SSPxCN2 register (SSPxCON2<6>).
10. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
11. The user generates a Stop condition by setting the Stop Enable bit, PEN (SSPxCON2<2>).
12. Interrupt is generated once the Stop condition is complete.

19.4.7 BAUD RATE

In I²C Master mode, the Baud Rate Generator (BRG) reload value is placed in the lower 7 bits of the SSPxADD register (Figure 19-19). When a write occurs to SSPxBUF, 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 (T_{CY}) 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 ACK), the internal clock will automatically stop counting and the SCLx pin will remain in its last state.

Table 19-3 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPxADD.

19.4.7.1 Baud Rate and Module Interdependence

Because MSSP1 and MSSP2 are independent, they can operate simultaneously in I²C Master mode at different baud rates. This is done by using different BRG reload values for each module.

Because this mode derives its basic clock source from the system clock, any changes to the clock will affect both modules in the same proportion. It may be possible to change one or both baud rates back to a previous value by changing the BRG reload value.

FIGURE 19-19: BAUD RATE GENERATOR BLOCK DIAGRAM

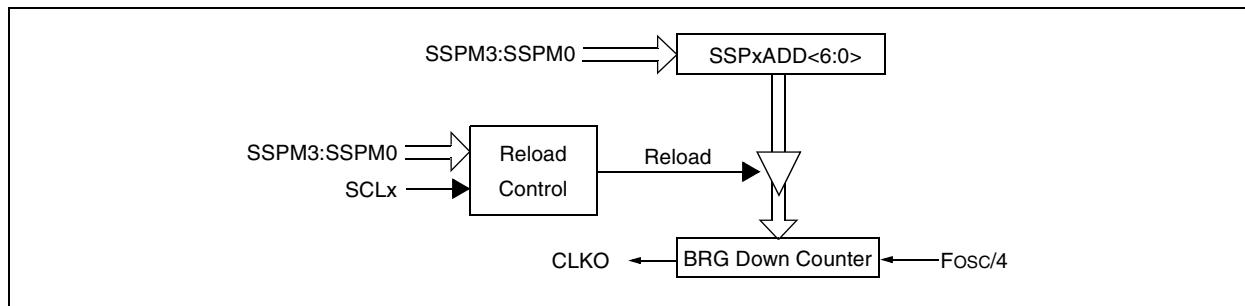


TABLE 19-3: I²C™ CLOCK RATE w/BRG

Fosc	F _{CY}	F _{CY} * 2	BRG Value	F _{SCL} (2 Rollovers of BRG)
40 MHz	10 MHz	20 MHz	18h	400 kHz ⁽¹⁾
40 MHz	10 MHz	20 MHz	1Fh	312.5 kHz
40 MHz	10 MHz	20 MHz	63h	100 kHz
16 MHz	4 MHz	8 MHz	09h	400 kHz ⁽¹⁾
16 MHz	4 MHz	8 MHz	0Ch	308 kHz
16 MHz	4 MHz	8 MHz	27h	100 kHz
4 MHz	1 MHz	2 MHz	02h	333 kHz ⁽¹⁾
4 MHz	1 MHz	2 MHz	09h	100 kHz
4 MHz	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.

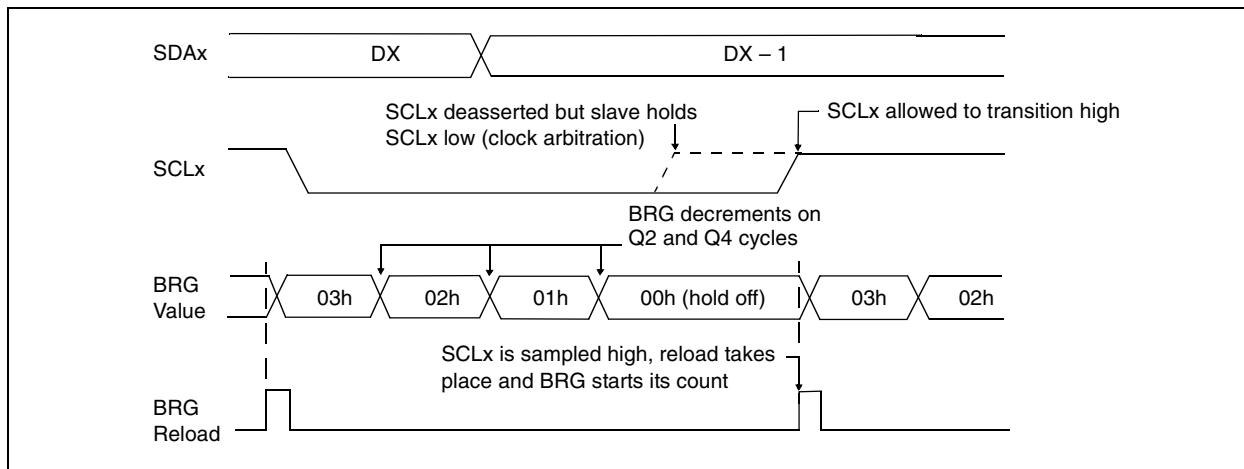
PIC18F87J11 FAMILY

19.4.7.2 Clock Arbitration

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, deasserts the SCL_x pin (SCL_x allowed to float high). When the SCL_x pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCL_x pin is actually sampled high. When the

SCL_x pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<6:0> and begins counting. This ensures that the SCL_x 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 19-20).

FIGURE 19-20: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION



19.4.8 I²C MASTER MODE START CONDITION TIMING

To initiate a Start condition, the user sets the Start Enable bit, SEN (SSPxCON2<0>). If the SDAx and SCLx pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<6:0> and starts its count. If SCLx and SDAx are both sampled high when the Baud Rate Generator times out (TBRG), the SDAx pin is driven low. The action of the SDAx being driven low while SCLx is high is the Start condition and causes the S bit (SSPxSTAT<3>) to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPxADD<6:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit (SSPxCON2<0>) will be automatically cleared by hardware. The Baud Rate Generator is suspended, leaving the SDAx line held low and the Start condition is complete.

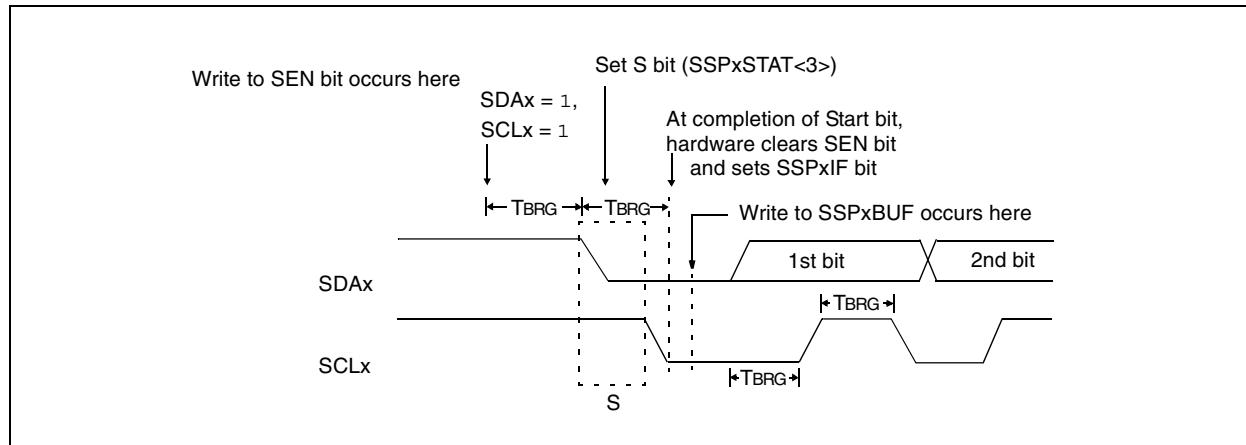
Note: If, at the beginning of the Start condition, the SDAx and SCLx pins are already sampled low or if during the Start condition, the SCLx line is sampled low before the SDAx line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLxIF, is set, the Start condition is aborted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

19.4.8.1 WCOL Status Flag

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

Note: Because queueing of events is not allowed, writing to the lower 5 bits of SSPxCON2 is disabled until the Start condition is complete.

FIGURE 19-21: FIRST START BIT TIMING



PIC18F87J11 FAMILY

19.4.9 I²C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition occurs when the RSEN bit (SSPxCON2<1>) is programmed high and the I²C logic module is in the Idle state. When the RSEN bit is set, the SCLx pin is asserted low. When the SCLx pin is sampled low, the Baud Rate Generator is loaded with the contents of SSPxADD<5:0> and begins counting. The SDAx pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out and if SDAx is sampled high, the SCLx pin will be deasserted (brought high). When SCLx is sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<6:0> and begins counting. SDAx and SCLx must be sampled high for one TBRG. This action is then followed by assertion of the SDAx pin (SDAx = 0) for one TBRG while SCLx is high. Following this, the RSEN bit (SSPxCON2<1>) will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDAx pin held low. As soon as a Start condition is detected on the SDAx and SCLx pins, the S bit (SSPxSTAT<3>) will be set. The SSPxIF 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:

- SDAx is sampled low when SCLx goes from low-to-high.
- SCLx goes low before SDAx is asserted low. This may indicate that another master is attempting to transmit a data '1'.

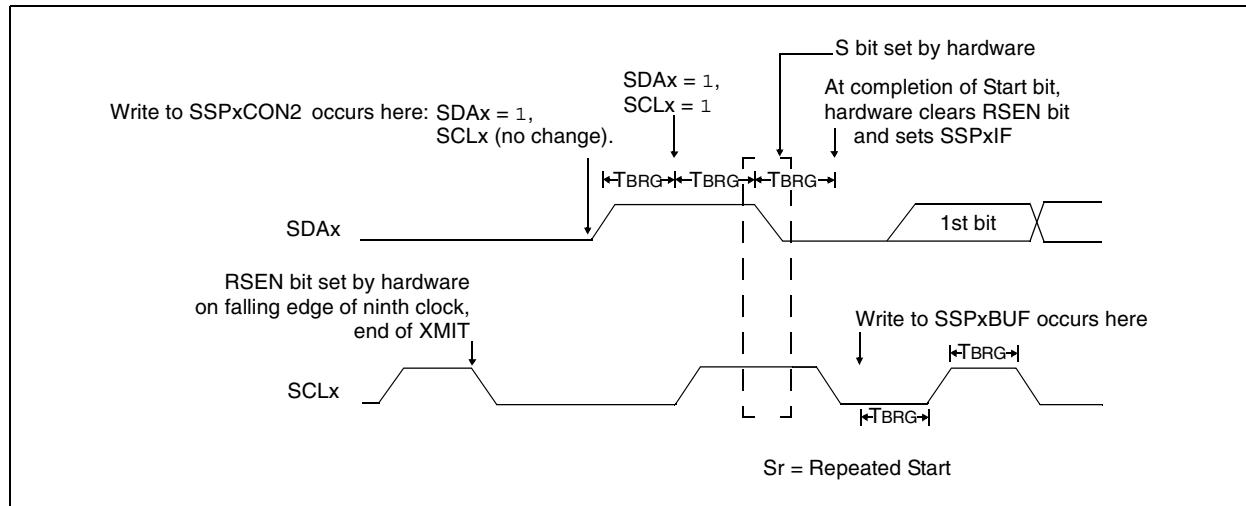
Immediately following the SSPxIF bit getting set, the user may write the SSPxBUF with the 7-bit address in 7-bit mode, or the default first address in 10-bit mode. After the first eight bits are transmitted and an ACK is received, the user may then transmit an additional eight bits of address (10-bit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

19.4.9.1 WCOL Status Flag

If the user writes the SSPxBUF 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 queueing of events is not allowed, writing of the lower 5 bits of SSPxCON2 is disabled until the Repeated Start condition is complete.

FIGURE 19-22: REPEATED START CONDITION WAVEFORM



19.4.10 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 SSPxBUF 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 SDAx pin after the falling edge of SCLx is asserted (see data hold time specification parameter 106). SCLx is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCLx is released high (see data setup time specification parameter 107). When the SCLx pin is released high, it is held that way for TBRG. The data on the SDAx pin must remain stable for that duration and some hold time after the next falling edge of SCLx. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDAx. 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 SSPxIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPxBUF, leaving SCLx low and SDAx unchanged (Figure 19-23).

After the write to the SSPxBUF, each bit of the address will be shifted out on the falling edge of SCLx 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 SDAx pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDAx pin to see if the address was recognized by a slave. The status of the ACK bit is loaded into the ACKSTAT status bit (SSPxCON2<6>). Following the falling edge of the ninth clock transmission of the address, the SSPxIF flag is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPxBUF takes place, holding SCLx low and allowing SDAx to float.

19.4.10.1 BF Status Flag

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

19.4.10.2 WCOL Status Flag

If the user writes the SSPxBUF when a transmit is already in progress (i.e., SSPxSR is still shifting out a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur) after 2 Tcy after the SSPxBUF write. If SSPxBUF is rewritten within 2 Tcy, the WCOL bit is set and SSPxBUF is updated. This may result in a corrupted transfer.

The user should verify that the WCOL bit is clear after each write to SSPxBUF to ensure the transfer is correct. In all cases, WCOL must be cleared in software.

19.4.10.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit (SSPxCON2<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.

19.4.11 I²C MASTER MODE RECEPTION

Master mode reception is enabled by programming the Receive Enable bit, RCEN (SSPxCON2<3>).

Note: The MSSP module must be in an inactive 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 SCLx pin changes (high-to-low/low-to-high) and data is shifted into the SSPxSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPxSR are loaded into the SSPxBUF, the BF flag bit is set, the SSPxIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCLx 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 (SSPxCON2<4>).

19.4.11.1 BF Status Flag

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

19.4.11.2 SSPOV Status Flag

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

19.4.11.3 WCOL Status Flag

If the user writes the SSPxBUF when a receive is already in progress (i.e., SSPxSR 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).

PIC18F87J11 FAMILY

FIGURE 19-23: I²C™ MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)

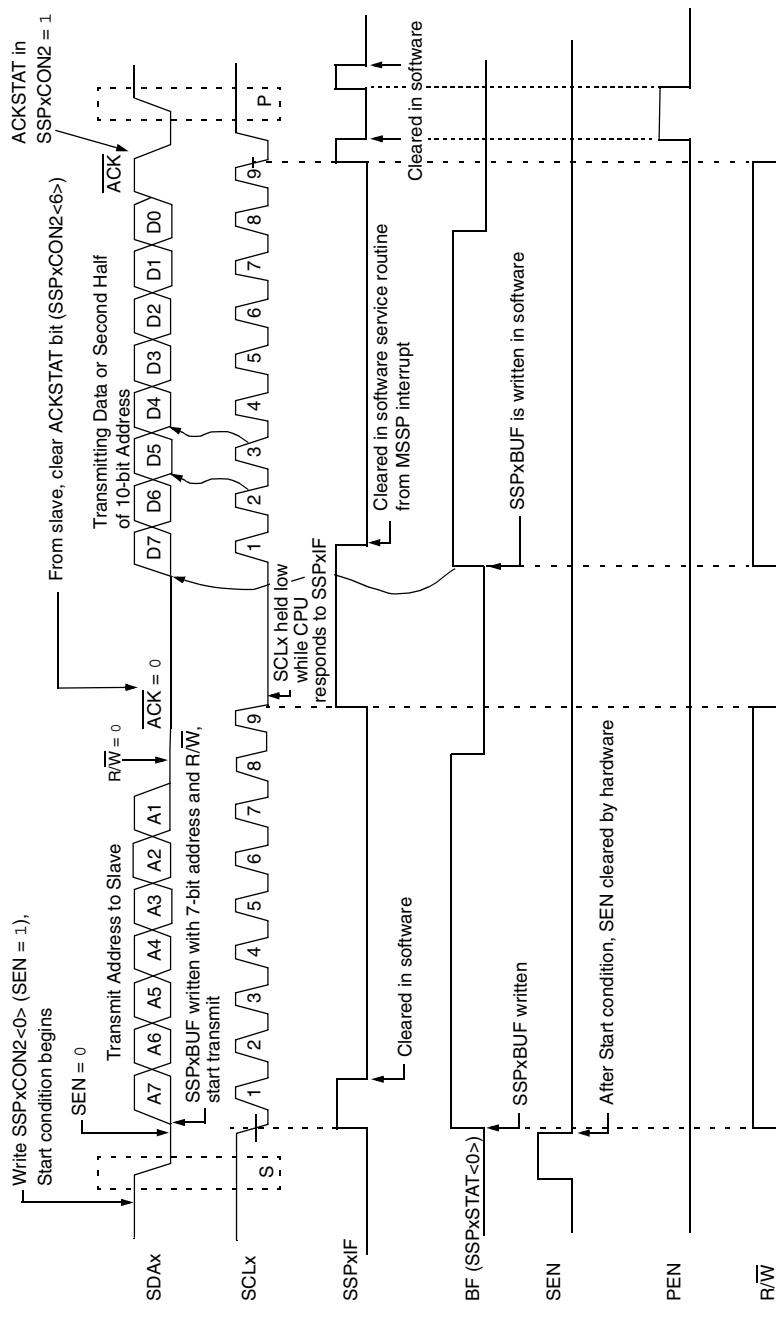
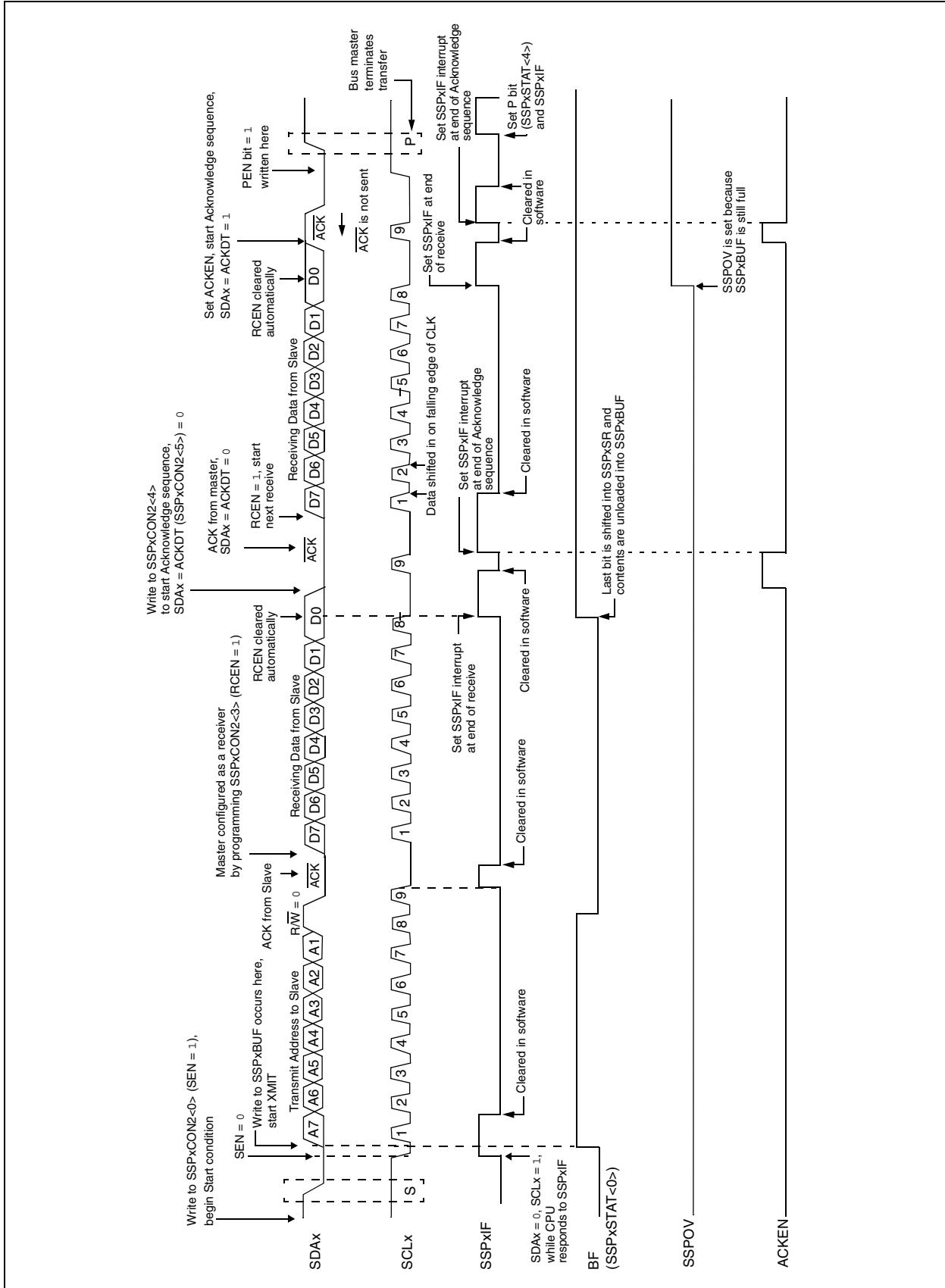


FIGURE 19-24: I²C™ MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)



PIC18F87J11 FAMILY

19.4.12 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN (SSPxCON2<4>). When this bit is set, the SCLx pin is pulled low and the contents of the Acknowledge data bit are presented on the SDAx 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 SCLx pin is deasserted (pulled high). When the SCLx pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG; the SCLx 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 an inactive state (Figure 19-25).

19.4.12.1 WCOL Status Flag

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

19.4.13 STOP CONDITION TIMING

A Stop bit is asserted on the SDAx pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN (SSPxCON2<2>). At the end of a receive/transmit, the SCLx line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDAx line low. When the SDAx line is sampled low, the Baud Rate Generator is reloaded and counts down to 0. When the Baud Rate Generator times out, the SCLx pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDAx pin will be deasserted. When the SDAx pin is sampled high while SCLx is high, the P bit (SSPxSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPxFIF bit is set (Figure 19-26).

19.4.13.1 WCOL Status Flag

If the user writes the SSPxBUF 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 19-25: ACKNOWLEDGE SEQUENCE WAVEFORM

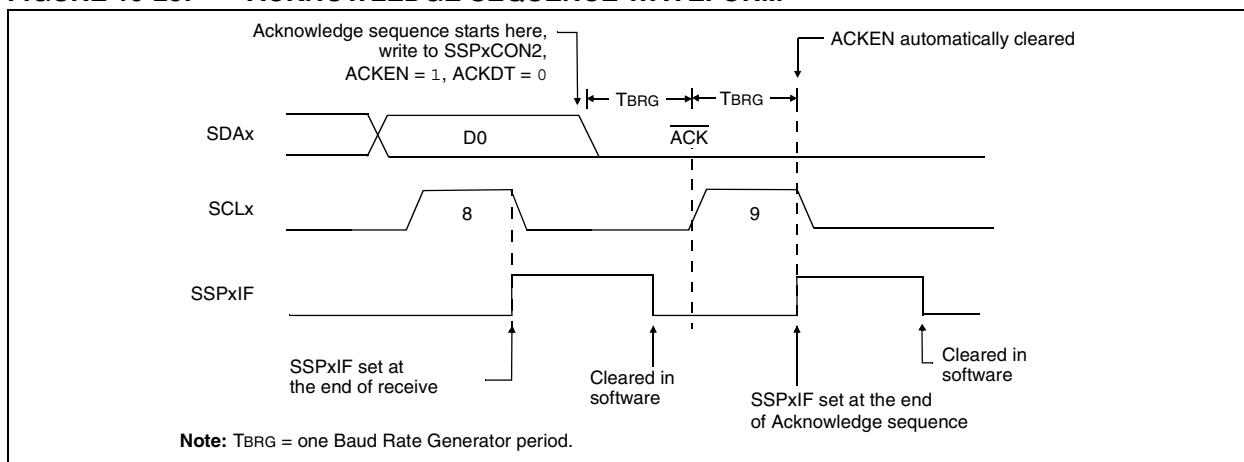
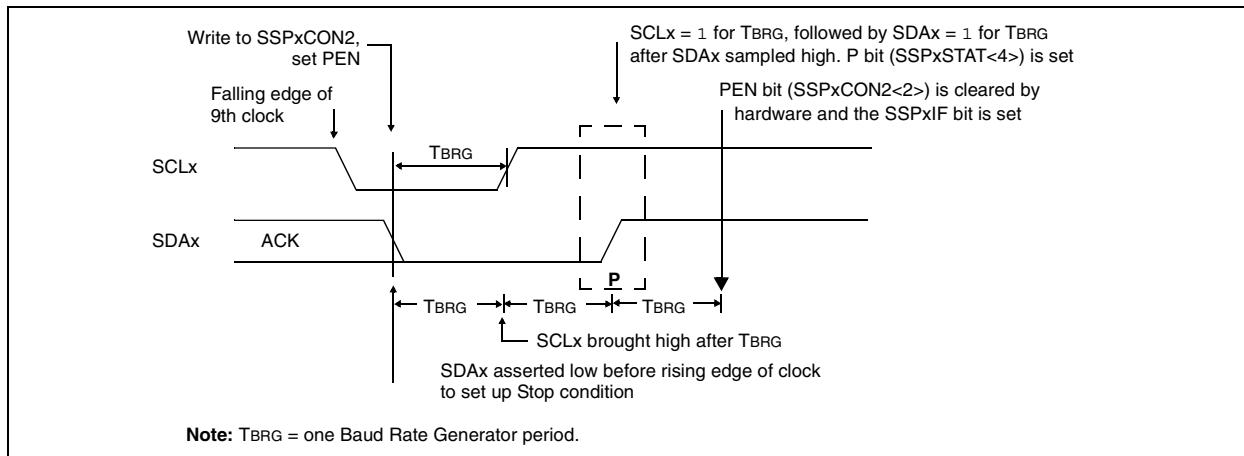


FIGURE 19-26: STOP CONDITION RECEIVE OR TRANSMIT MODE



19.4.14 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).

19.4.15 EFFECTS OF A RESET

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

19.4.16 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 (SSPxSTAT<4>) is set, or the bus is Idle, with both the S and P bits clear. When the bus is busy, enabling the MSSP interrupt will generate the interrupt when the Stop condition occurs.

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

The states where arbitration can be lost are:

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

19.4.17 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 SDAx pin, arbitration takes place when the master outputs a '1' on SDAx, by letting SDAx float high, and another master asserts a '0'. When the SCLx pin floats high, data should be stable. If the expected data on SDAx is a '1' and the data sampled on the SDAx pin = 0, then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLxIF and reset the I²C port to its Idle state (Figure 19-27).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDAx and SCLx lines are deasserted and the SSPxBUF 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.

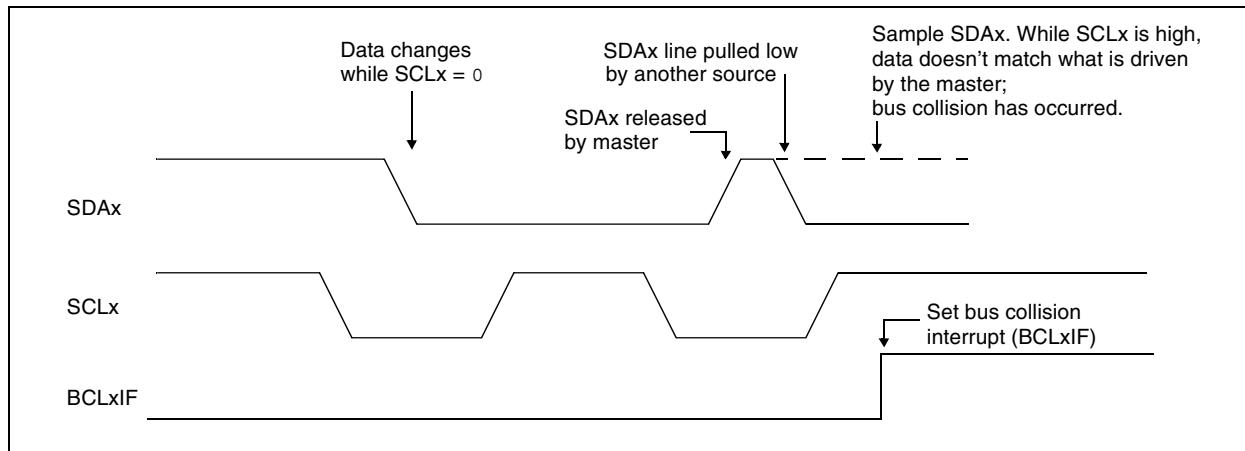
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDAx and SCLx lines are deasserted and the respective control bits in the SSPxCON2 register are cleared. When the user services the bus collision Interrupt Service 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 SDAx and SCLx pins. If a Stop condition occurs, the SSPxIF bit will be set.

A write to the SSPxBUF 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 SSPxSTAT register, or the bus is Idle and the S and P bits are cleared.

FIGURE 19-27: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE



PIC18F87J11 FAMILY

19.4.17.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

- SDAx or SCLx is sampled low at the beginning of the Start condition (Figure 19-28).
- SCLx is sampled low before SDAx is asserted low (Figure 19-29).

During a Start condition, both the SDAx and the SCLx pins are monitored.

If the SDAx pin is already low, or the SCLx pin is already low, then all of the following occur:

- the Start condition is aborted,
- the BCLxIF flag is set and
- the MSSP module is reset to its inactive state (Figure 19-28)

The Start condition begins with the SDAx and SCLx pins deasserted. When the SDAx pin is sampled high, the Baud Rate Generator is loaded from SSPxADD<6:0> and counts down to 0. If the SCLx pin is sampled low while SDAx 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 SDAx pin is sampled low during this count, the BRG is reset and the SDAx line is asserted early (Figure 19-30). If, however, a '1' is sampled on the SDAx pin, the SDAx pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to 0. If the SCLx pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCLx 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 SDAx 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 19-28: BUS COLLISION DURING START CONDITION (SDAx ONLY)

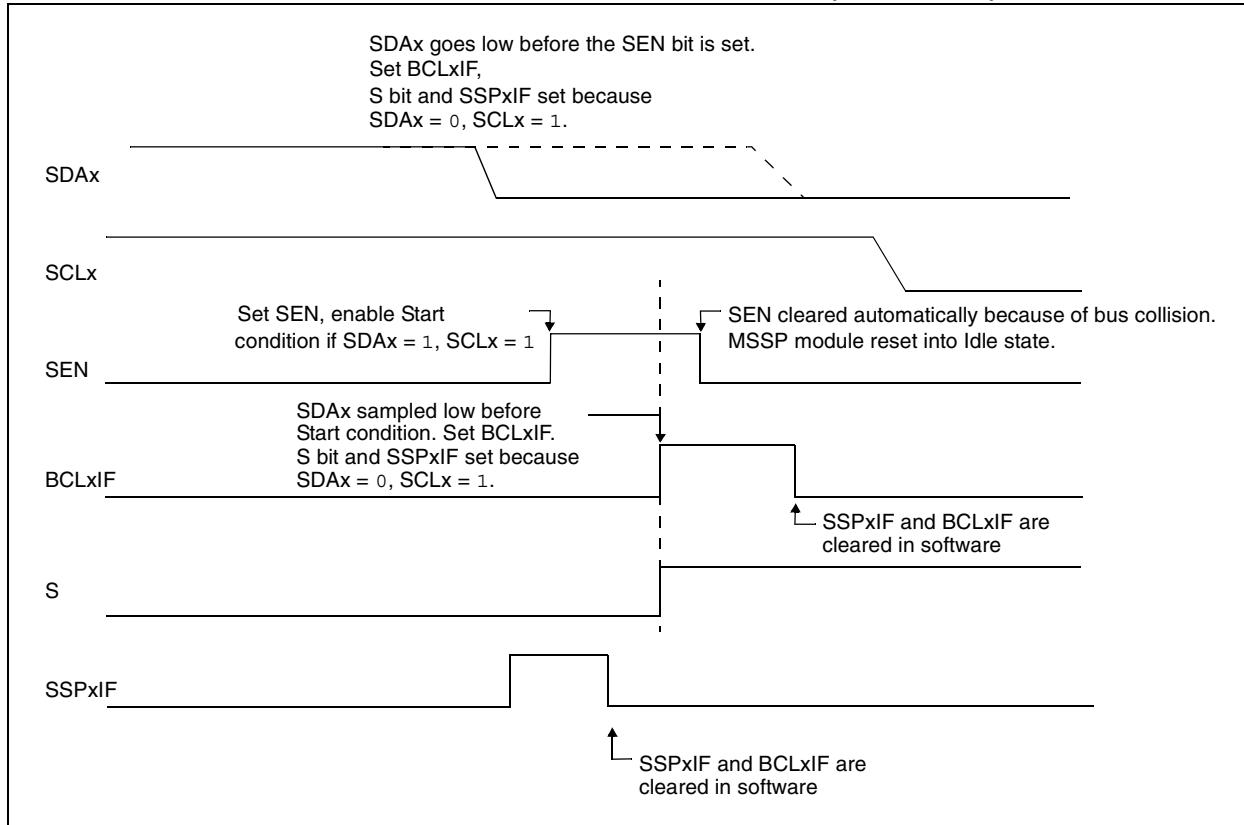


FIGURE 19-29: BUS COLLISION DURING START CONDITION ($SCLx = 0$)

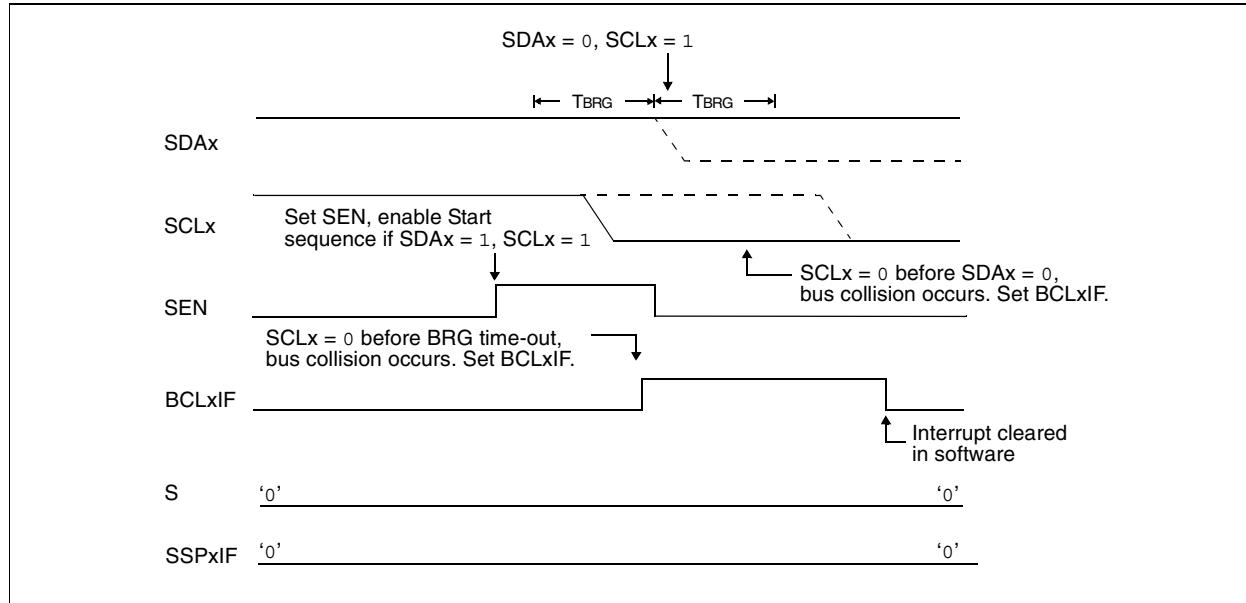
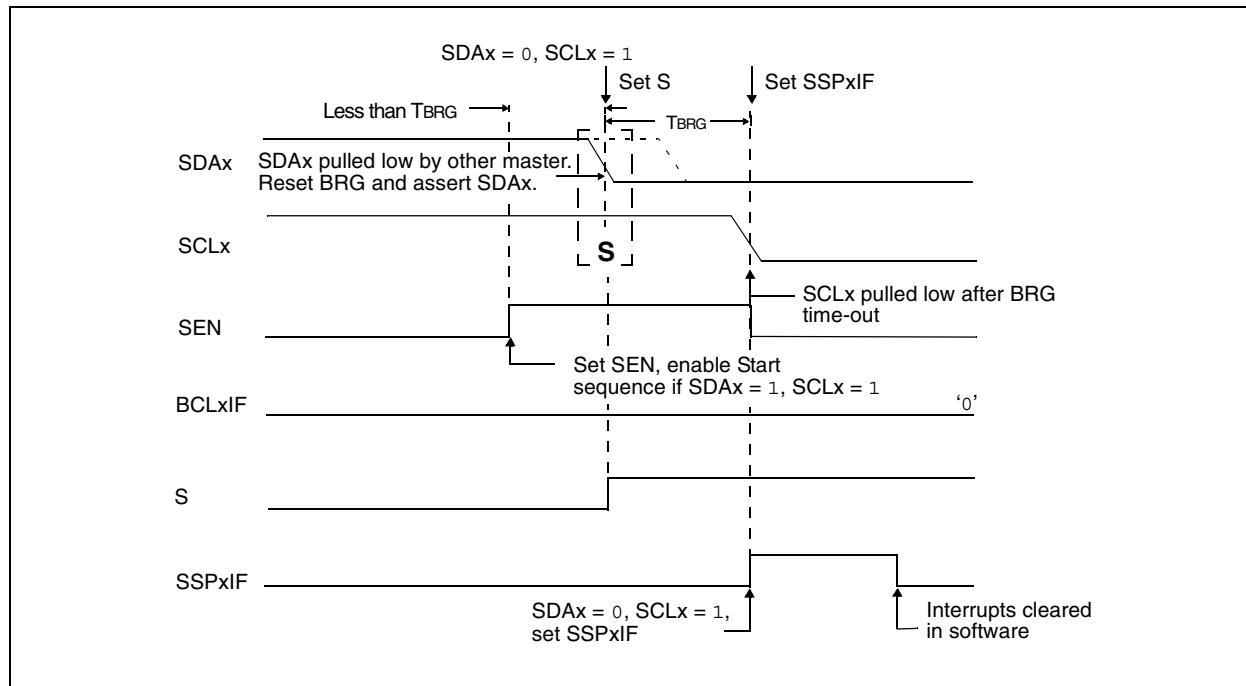


FIGURE 19-30: BRG RESET DUE TO SDAX ARBITRATION DURING START CONDITION



PIC18F87J11 FAMILY

19.4.17.2 Bus Collision During a Repeated Start Condition

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

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

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

If SDAx is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', Figure 19-31). If SDAx is sampled high, the BRG is reloaded and begins counting. If SDAx goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDAx at exactly the same time.

If SCLx goes from high-to-low before the BRG times out and SDAx 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 (see Figure 19-32).

If, at the end of the BRG time-out, both SCLx and SDAx are still high, the SDAx pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCLx pin, the SCLx pin is driven low and the Repeated Start condition is complete.

FIGURE 19-31: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

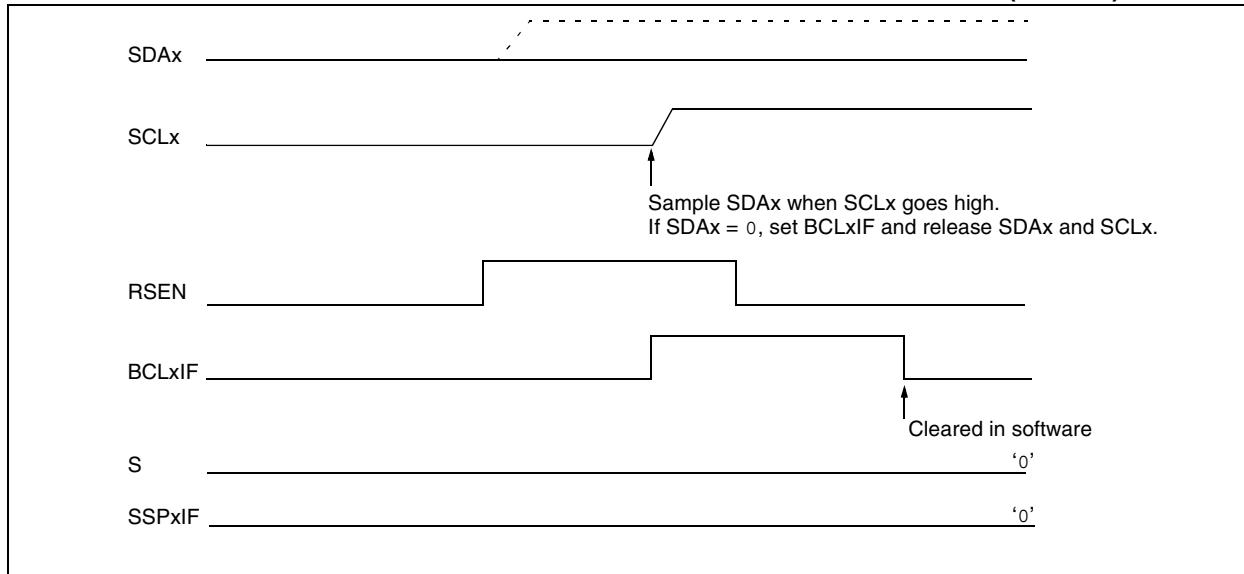
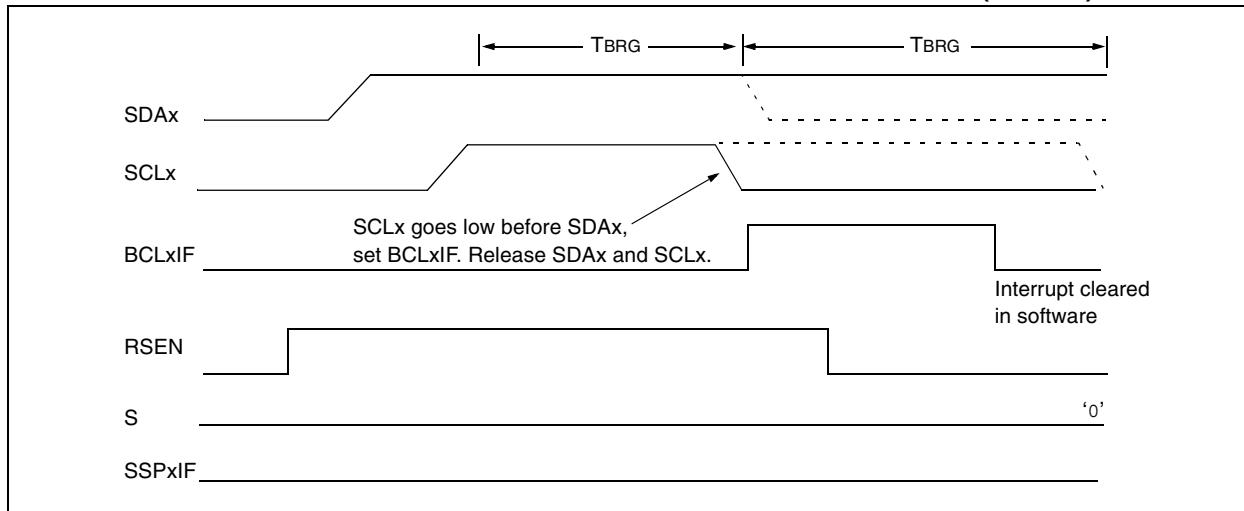


FIGURE 19-32: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)



19.4.17.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

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

The Stop condition begins with SDAx asserted low. When SDAx is sampled low, the SCLx pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPxADD<6:0> and counts down to 0. After the BRG times out, SDAx is sampled. If SDAx is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 19-33). If the SCLx pin is sampled low before SDAx is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 19-34).

FIGURE 19-33: BUS COLLISION DURING A STOP CONDITION (CASE 1)

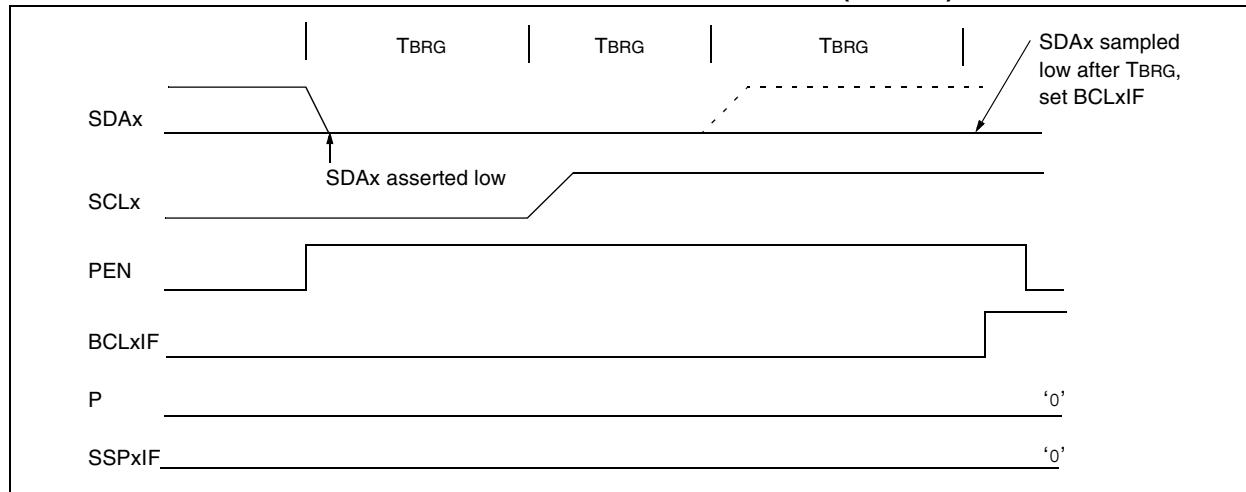
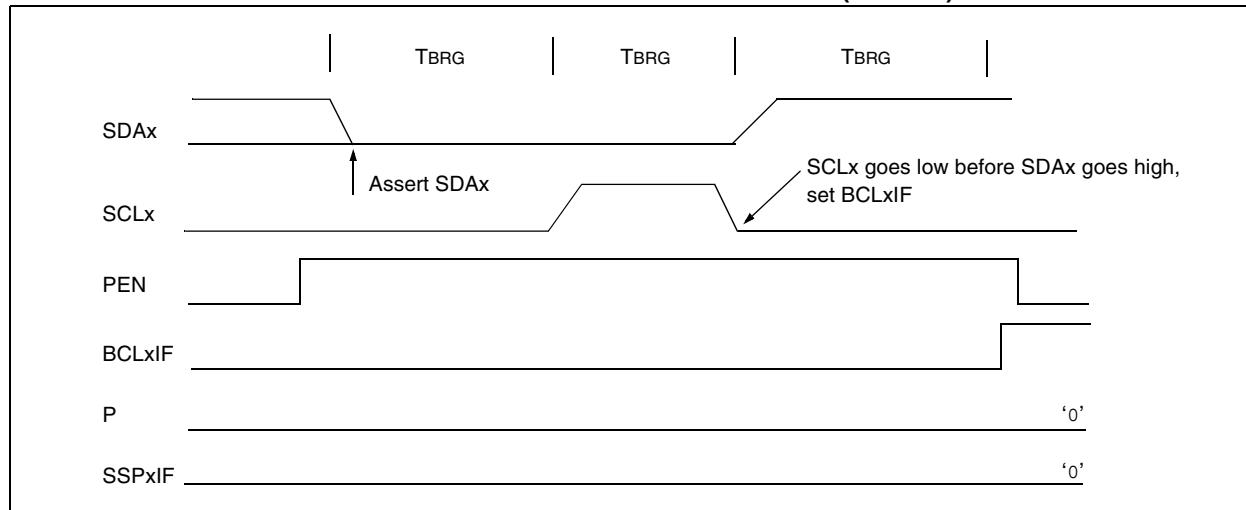


FIGURE 19-34: BUS COLLISION DURING A STOP CONDITION (CASE 2)



PIC18F87J11 FAMILY

TABLE 19-4: REGISTERS ASSOCIATED WITH I²C™ OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	58
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	58
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISCO	58
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	58
SSP1BUF	MSSP1 Receive Buffer/Transmit Register								56
SSP1ADD	MSSP1 Address Register (I ² C™ Slave mode), MSSP1 Baud Rate Reload Register (I ² C Master mode)								56
SSP1MSK ⁽¹⁾	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	56
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	56
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	56
	GCEN	ACKSTAT	ADMSK5 ⁽²⁾	ADMSK4 ⁽²⁾	ADMSK3 ⁽²⁾	ADMSK2 ⁽²⁾	ADMSK1 ⁽²⁾	SEN	
SSP1STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	56
SSP2BUF	MSSP2 Receive Buffer/Transmit Register								59
SSP2ADD	MSSP2 Address Register (I ² C Slave mode), MSSP2 Baud Rate Reload Register (I ² C Master mode)								59
SSP2MSK ⁽¹⁾	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	59
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	59
SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	59
	GCEN	ACKSTAT	ADMSK5 ⁽²⁾	ADMSK4 ⁽²⁾	ADMSK3 ⁽²⁾	ADMSK2 ⁽²⁾	ADMSK1 ⁽²⁾	SEN	
SSP2STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	59

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the MSSP module in I²C™ mode.

Note 1: SSPxMSK shares the same address in SFR space as SSPxADD, but is only accessible in certain I²C™ Slave operating modes in 7-bit Masking mode. See **Section 19.4.3.4 “7-Bit Address Masking Mode”** for more details.

2: Alternate bit definitions for use in I²C Slave mode operations only.

20.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of two serial I/O modules. (Generically, the EUSART is also known as a Serial Communications Interface or SCI.) The EUSART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also 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 Enhanced USART module implements additional features, including automatic baud rate detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break character transmit. These make it ideally suited for use in Local Interconnect Network bus (LIN bus) systems.

All members of the PIC18F87J11 family are equipped with two independent EUSART modules, referred to as EUSART1 and EUSART2. They can be configured in the following modes:

- Asynchronous (full duplex) with:
 - Auto-wake-up on character reception
 - Auto-baud calibration
 - 12-bit Break character transmission
- Synchronous – Master (half duplex) with selectable clock polarity
- Synchronous – Slave (half duplex) with selectable clock polarity

The pins of EUSART1 and EUSART2 are multiplexed with the functions of PORTC (RC6/TX1/CK1 and RC7/RX1/DT1) and PORTG (RG1/TX2/CK2 and RG2/RX2/DT2), respectively. In order to configure these pins as an EUSART:

- For EUSART1:
 - bit SPEN (RCSTA1<7>) must be set (= 1)
 - bit TRISC<7> must be set (= 1)
 - bit TRISC<6> must be cleared (= 0) for Asynchronous and Synchronous Master modes
 - bit TRISC<6> must be set (= 1) for Synchronous Slave mode
- For EUSART2:
 - bit SPEN (RCSTA2<7>) must be set (= 1)
 - bit TRISG<2> must be set (= 1)
 - bit TRISG<1> must be cleared (= 0) for Asynchronous and Synchronous Master modes
 - bit TRISC<6> must be set (= 1) for Synchronous Slave mode

Note: The EUSART control will automatically reconfigure the pin from input to output as needed.

The operation of each Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTAx)
- Receive Status and Control (RCSTAx)
- Baud Rate Control (BAUDCONx)

These are detailed on the following pages in Register 20-1, Register 20-2 and Register 20-3, respectively.

Note: Throughout this section, references to register and bit names that may be associated with a specific EUSART module are referred to generically by the use of 'x' in place of the specific module number. Thus, "RCSTAx" might refer to the Receive Status register for either EUSART1 or EUSART2.

PIC18F87J11 FAMILY

REGISTER 20-1: TXSTAx: TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN ⁽¹⁾	SYNC	SENDDB	BRGH	TRMT	TX9D
bit 7							bit 0

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

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⁾

1 = Transmit enabled

0 = Transmit disabled

bit 4 **SYNC:** EUSART Mode Select bit

1 = Synchronous mode

0 = Asynchronous mode

bit 3 **SENDDB:** Send Break Character bit

Asynchronous mode:

1 = Send Sync Break on next transmission (cleared by hardware upon completion)

0 = Sync Break transmission completed

Synchronous mode:

Don't care.

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 address/data bit or a parity bit.

Note 1: SREN/CREN overrides TXEN in Sync mode.

REGISTER 20-2: RCSTAx: RECEIVE STATUS AND CONTROL REGISTER

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

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

bit 7	SPEN: Serial Port Enable bit 1 = Serial port enabled (configures RXx/DTx and TXx/CKx pins as serial port pins) 0 = Serial port disabled (held in Reset)
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 <u>Asynchronous mode:</u> Don't care. <u>Synchronous mode – Master:</u> 1 = Enables single receive 0 = Disables single receive This bit is cleared after reception is complete. <u>Synchronous mode – Slave:</u> Don't care.
bit 4	CREN: Continuous Receive Enable bit <u>Asynchronous mode:</u> 1 = Enables receiver 0 = Disables receiver <u>Synchronous mode:</u> 1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN) 0 = Disables continuous receive
bit 3	ADDEN: Address Detect Enable bit <u>Asynchronous mode 9-Bit (RX9 = 1):</u> 1 = Enables address detection, enables interrupt and loads 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 <u>Asynchronous mode 9-Bit (RX9 = 0):</u> Don't care.
bit 2	FERR: Framing Error bit 1 = Framing error (can be updated by reading RCREGx register and receiving 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 This can be address/data bit or a parity bit and must be calculated by user firmware.

PIC18F87J11 FAMILY

REGISTER 20-3: BAUDCONx: BAUD RATE CONTROL REGISTER

R/W-0	R-1	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

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

bit 7 **ABDOVF:** Auto-Baud Acquisition Rollover Status bit1 = A BRG rollover has occurred during Auto-Baud Rate Detect mode
(must be cleared in software)

0 = No BRG rollover has occurred

bit 6 **RCIDL:** Receive Operation Idle Status bit1 = Receive operation is Idle
0 = Receive operation is activebit 5 **DTRXP:** Data/Receive Polarity Select bitAsynchronous mode:1 = Receive data (RXx) is inverted (active low)
0 = Receive data (RXx) is not inverted (active high)Synchronous mode:1 = Data (DTx) is inverted (active low)
0 = Data (DTx) is not inverted (active high)bit 4 **SCKP:** Synchronous Clock Polarity Select bitAsynchronous mode:1 = Idle state for transmit (TXx) is a low level
0 = Idle state for transmit (TXx) is a high levelSynchronous mode:1 = Idle state for clock (CKx) is a high level
0 = Idle state for clock (CKx) is a low levelbit 3 **BRG16:** 16-Bit Baud Rate Register Enable bit1 = 16-bit Baud Rate Generator – SPBRGHx and SPBRGx
0 = 8-bit Baud Rate Generator – SPBRGx only (Compatible mode), SPBRGHx value ignoredbit 2 **Unimplemented:** Read as '0'bit 1 **WUE:** Wake-up Enable bitAsynchronous mode:1 = EUSART will continue to sample the RXx pin – interrupt generated on falling edge; bit cleared in hardware on following rising edge
0 = RXx pin not monitored or rising edge detectedSynchronous mode:

Unused in this mode.

bit 0 **ABDEN:** Auto-Baud Detect Enable bitAsynchronous mode:1 = Enable baud rate measurement on the next character. Requires reception of a Sync field (55h); cleared in hardware upon completion.
0 = Baud rate measurement disabled or completedSynchronous mode:

Unused in this mode.

20.1 Baud Rate Generator (BRG)

The BRG is a dedicated, 8-bit or 16-bit generator that supports both the Asynchronous and Synchronous modes of the EUSART. By default, the BRG operates in 8-bit mode; setting the BRG16 bit (BAUDCONx<3>) selects 16-bit mode.

The SPBRGHx:SPBRGx register pair controls the period of a free-running timer. In Asynchronous mode, bits BRGH (TXSTA_x<2>) and BRG16 (BAUDCON_x<3>) also control the baud rate. In Synchronous mode, BRGH is ignored. Table 20-1 shows the formula for computation of the baud rate for different EUSART modes which only apply in Master mode (internally generated clock).

Given the desired baud rate and Fosc, the nearest integer value for the SPBRGHx:SPBRGx registers can be calculated using the formulas in Table 20-1. From this, the error in baud rate can be determined. An example calculation is shown in Example 20-1. Typical baud rates and error values for the various Asynchronous modes are shown in Table 20-2. It may be advantageous to use

the high baud rate (BRGH = 1) or the 16-bit BRG to reduce the baud rate error, or achieve a slow baud rate for a fast oscillator frequency.

Writing a new value to the SPBRGHx:SPBRGx registers 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.

20.1.1 OPERATION IN POWER-MANAGED MODES

The device clock is used to generate the desired baud rate. When one of the power-managed modes is entered, the new clock source may be operating at a different frequency. This may require an adjustment to the value in the SPBRGx register pair.

20.1.2 SAMPLING

The data on the RX_x pin (either RC7/RX1/DT1 or RG2/RX2/DT2) is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX_x pin.

TABLE 20-1: BAUD RATE FORMULAS

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	Fosc/[64 (n + 1)]
0	0	1	8-bit/Asynchronous	Fosc/[16 (n + 1)]
0	1	0	16-bit/Asynchronous	
0	1	1	16-bit/Asynchronous	Fosc/[4 (n + 1)]
1	0	x	8-bit/Synchronous	
1	1	x	16-bit/Synchronous	

Legend: x = Don't care, n = value of SPBRGHx:SPBRGx register pair

PIC18F87J11 FAMILY

EXAMPLE 20-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, and 8-bit BRG:

$$\text{Desired Baud Rate} = \text{FOSC}/(64 ([\text{SPBRGHx}:\text{SPBRGx}] + 1))$$

Solving for SPBRGHx:SPBRGx:

$$\begin{aligned} X &= ((\text{FOSC}/\text{Desired Baud Rate})/64) - 1 \\ &= ((16000000/9600)/64) - 1 \\ &= [25.042] = 25 \end{aligned}$$

$$\begin{aligned} \text{Calculated Baud Rate} &= 16000000/(64 (25 + 1)) \\ &= 9615 \end{aligned}$$

$$\begin{aligned} \text{Error} &= (\text{Calculated Baud Rate} - \text{Desired Baud Rate})/\text{Desired Baud Rate} \\ &= (9615 - 9600)/9600 = 0.16\% \end{aligned}$$

TABLE 20-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
BAUDCONx	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	59
SPBRGHx	EUSARTx Baud Rate Generator Register High Byte								59
SPBRGx	EUSARTx Baud Rate Generator Register Low Byte								59

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

PIC18F87J11 FAMILY

TABLE 20-3: BAUD RATES FOR ASYNCHRONOUS MODES

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1.201	-0.16	103
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2.403	-0.16	51
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9.615	-0.16	12
19.2	19.531	1.73	31	19.531	1.73	15	19.531	1.73	7	—	—	—
57.6	56.818	-1.36	10	62.500	8.51	4	52.083	-9.58	2	—	—	—
115.2	125.000	8.51	4	104.167	-9.58	2	78.125	-32.18	1	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz					
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.16	207	0.300	-0.16	103	0.300	-0.16	51	—	—	—
1.2	1.202	0.16	51	1.201	-0.16	25	1.201	-0.16	12	—	—	—
2.4	2.404	0.16	25	2.403	-0.16	12	—	—	—	—	—	—
9.6	8.929	-6.99	6	—	—	—	—	—	—	—	—	—
19.2	20.833	8.51	2	—	—	—	—	—	—	—	—	—
57.6	62.500	8.51	0	—	—	—	—	—	—	—	—	—
115.2	62.500	-45.75	0	—	—	—	—	—	—	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	2.441	1.73	255	2.403	-0.16	207
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz					
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	0.300	-0.16	207	—	—	—
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51	—	—	—
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25	—	—	—
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—	—	—	—

PIC18F87J11 FAMILY

TABLE 20-3: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.00	8332	0.300	0.02	4165	0.300	0.02	2082	0.300	-0.04	1665
1.2	1.200	0.02	2082	1.200	-0.03	1041	1.200	-0.03	520	1.201	-0.16	415
2.4	2.402	0.06	1040	2.399	-0.03	520	2.404	0.16	259	2.403	-0.16	207
9.6	9.615	0.16	259	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz					
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.04	832	0.300	-0.16	415	0.300	-0.16	207	—	—	—
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51	—	—	—
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25	—	—	—
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.00	33332	0.300	0.00	16665	0.300	0.00	8332	0.300	-0.01	6665
1.2	1.200	0.00	8332	1.200	0.02	4165	1.200	0.02	2082	1.200	-0.04	1665
2.4	2.400	0.02	4165	2.400	0.02	2082	2.402	0.06	1040	2.400	-0.04	832
9.6	9.606	0.06	1040	9.596	-0.03	520	9.615	0.16	259	9.615	-0.16	207
19.2	19.193	-0.03	520	19.231	0.16	259	19.231	0.16	129	19.230	-0.16	103
57.6	57.803	0.35	172	57.471	-0.22	86	58.140	0.94	42	57.142	0.79	34
115.2	114.943	-0.22	86	116.279	0.94	42	113.636	-1.36	21	117.647	-2.12	16

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz					
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.01	3332	0.300	-0.04	1665	0.300	-0.04	832	—	—	—
1.2	1.200	0.04	832	1.201	-0.16	415	1.201	-0.16	207	—	—	—
2.4	2.404	0.16	415	2.403	-0.16	207	2.403	-0.16	103	—	—	—
9.6	9.615	0.16	103	9.615	-0.16	51	9.615	-0.16	25	—	—	—
19.2	19.231	0.16	51	19.230	-0.16	25	19.230	-0.16	12	—	—	—
57.6	58.824	2.12	16	55.555	3.55	8	—	—	—	—	—	—
115.2	111.111	-3.55	8	—	—	—	—	—	—	—	—	—

20.1.3 AUTO-BAUD RATE DETECT

The Enhanced USART module supports the automatic detection and calibration of baud rate. This feature is active only in Asynchronous mode and while the WUE bit is clear.

The automatic baud rate measurement sequence (Figure 20-1) begins whenever a Start bit is received and the ABDEN bit is set. The calculation is self-averaging.

In the Auto-Baud Rate Detect (ABD) mode, the clock to the BRG is reversed. Rather than the BRG clocking the incoming RXx signal, the RXx signal is timing the BRG. In ABD mode, the internal Baud Rate Generator is used as a counter to time the bit period of the incoming serial byte stream.

Once the ABDEN bit is set, the state machine will clear the BRG and look for a Start bit. The Auto-Baud Rate Detect must receive a byte with the value 55h (ASCII "U", which is also the LIN bus Sync character) in order to calculate the proper bit rate. The measurement is taken over both a low and a high bit time in order to minimize any effects caused by asymmetry of the incoming signal. After a Start bit, the SPBRGx begins counting up, using the preselected clock source on the first rising edge of RXx. After eight bits on the RXx pin or the fifth rising edge, an accumulated value totalling the proper BRG period is left in the SPBRGHx:SPBRGx register pair. Once the 5th edge is seen (this should correspond to the Stop bit), the ABDEN bit is automatically cleared.

If a rollover of the BRG occurs (an overflow from FFFFh to 0000h), the event is trapped by the ABDOVF status bit (BAUDCONx<7>). It is set in hardware by BRG rollovers and can be set or cleared by the user in software. ABD mode remains active after rollover events and the ABDEN bit remains set (Figure 20-2).

While calibrating the baud rate period, the BRG registers are clocked at 1/8th the preconfigured clock rate. Note that the BRG clock will be configured by the BRG16 and BRGH bits. Independent of the BRG16 bit setting, both the SPBRGx and SPBRGHx will be used as a 16-bit counter. This allows the user to verify that no carry occurred for 8-bit modes by checking for 00h in the SPBRGHx register. Refer to Table 20-4 for counter clock rates to the BRG.

While the ABD sequence takes place, the USART state machine is held in Idle. The RCxIF interrupt is set once the fifth rising edge on RXx is detected. The value in the RCREGx needs to be read to clear the RCxIF interrupt. The contents of RCREGx should be discarded.

Note 1: If the WUE bit is set with the ABDEN bit, Auto-Baud Rate Detection will occur on the byte *following* the Break character.

2: It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible due to bit error rates. Overall system timing and communication baud rates must be taken into consideration when using the Auto-Baud Rate Detection feature.

TABLE 20-4: BRG COUNTER CLOCK RATES

BRG16	BRGH	BRG Counter Clock
0	0	Fosc/512
0	1	Fosc/128
1	0	Fosc/128
1	1	Fosc/32

Note: During the ABD sequence, SPBRGx and SPBRGHx are both used as a 16-bit counter, independent of BRG16 setting.

20.1.3.1 ABD and EUSART Transmission

Since the BRG clock is reversed during ABD acquisition, the EUSART transmitter cannot be used during ABD. This means that whenever the ABDEN bit is set, TXREGx cannot be written to. Users should also ensure that ABDEN does not become set during a transmit sequence. Failing to do this may result in unpredictable EUSART operation.

PIC18F87J11 FAMILY

FIGURE 20-1: AUTOMATIC BAUD RATE CALCULATION

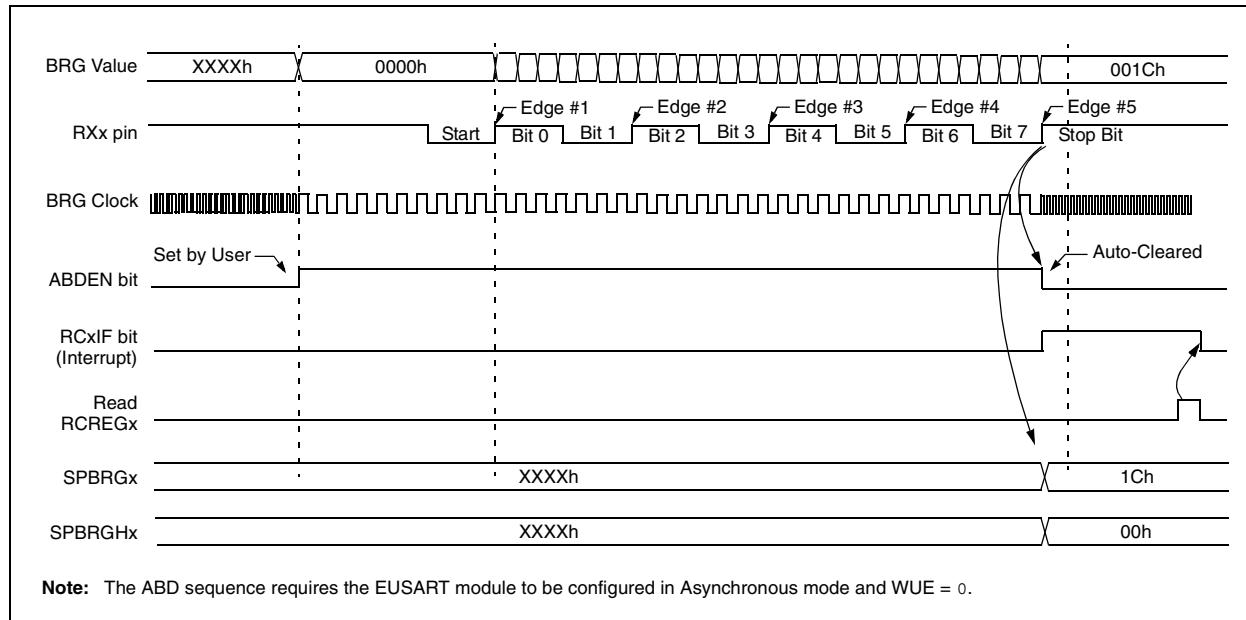
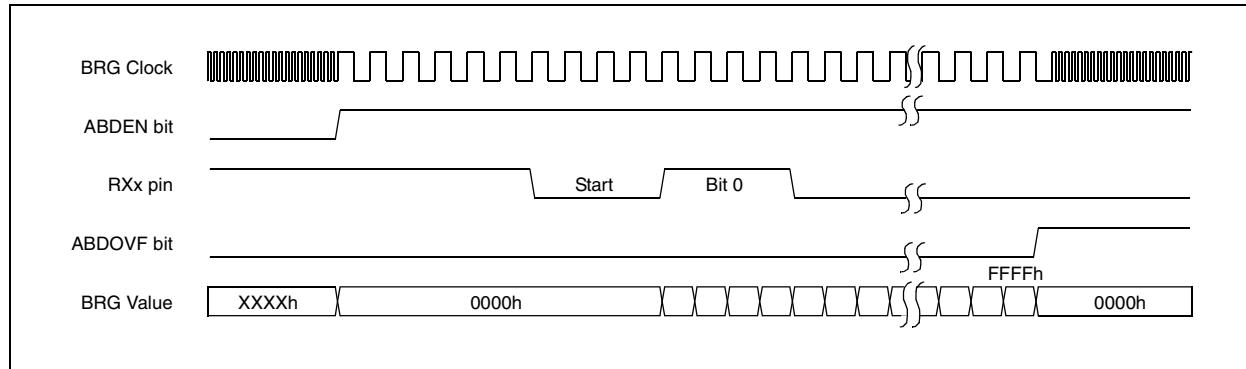


FIGURE 20-2: BRG OVERFLOW SEQUENCE



20.2 EUSART Asynchronous Mode

The Asynchronous mode of operation is selected by clearing the SYNC bit (TXSTAx<4>). In this mode, the EUSART 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/16-bit Baud Rate Generator can be used to derive standard baud rate frequencies from the oscillator.

The EUSART transmits and receives the LSb first. The EUSART's transmitter and receiver are functionally independent but use the same data format and baud rate. The Baud Rate Generator produces a clock, either x16 or x64 of the bit shift rate, depending on the BRGH and BRG16 bits (TXSTAx<2> and BAUDCONx<3>). Parity is not supported by the hardware but can be implemented in software and stored as the 9th data bit.

When operating in Asynchronous mode, the EUSART module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver
- Auto-Wake-up on Sync Break Character
- 12-Bit Break Character Transmit
- Auto-Baud Rate Detection

20.2.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 20-3. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The Shift register obtains its data from the Read/Write Transmit Buffer register, TXREGx. The TXREGx 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 TXREGx register (if available).

Once the TXREGx register transfers the data to the TSR register (occurs in one Tcy), the TXREGx register is empty and the TXxIF flag bit is set. This interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TXxE. TXxIF will be set regardless of the state of TXxE; it cannot be cleared in software. TXxIF is also not cleared immediately upon loading TXREGx, but becomes valid in the second instruction cycle following the load instruction. Polling TXxIF immediately following a load of TXREGx will return invalid results.

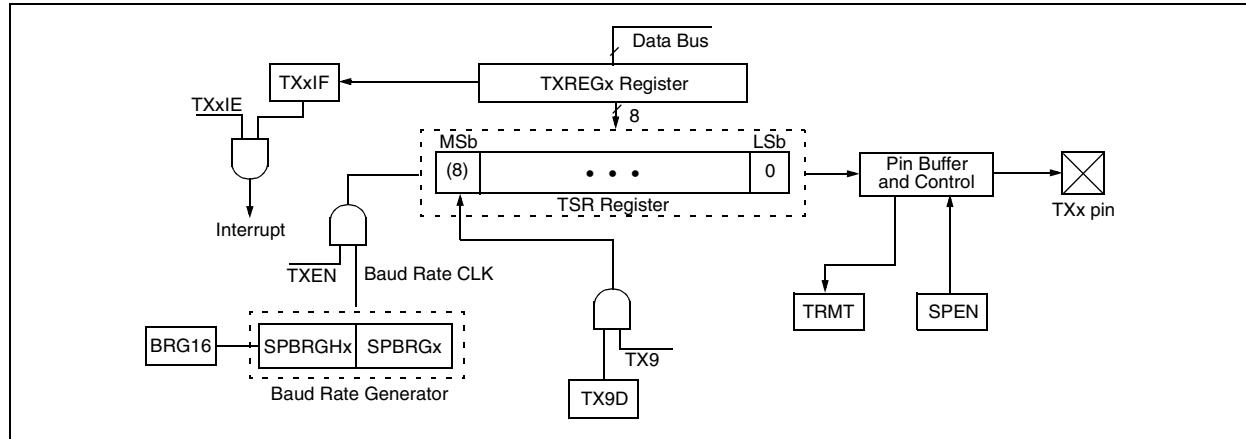
While TXxIF indicates the status of the TXREGx register; another bit, TRMT (TXSTAx<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR register is empty. No interrupt logic is tied to this bit so the user has to poll this bit in order to determine if the TSR register is empty.

- Note 1:** The TSR register is not mapped in data memory, so it is not available to the user.
2: Flag bit, TXxIF, is set when enable bit, TXEN, is set.

To set up an Asynchronous Transmission:

1. Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit, SPEN.
3. If interrupts are desired, set enable bit, TXxE.
4. If 9-bit transmission is desired, set transmit bit TX9. Can be used as address/data bit.
5. Enable the transmission by setting bit, TXEN, which will also set bit, TXxIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit, TX9D.
7. Load data to the TXREGx register (starts transmission).
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

FIGURE 20-3: EUSART TRANSMIT BLOCK DIAGRAM



PIC18F87J11 FAMILY

FIGURE 20-4: ASYNCHRONOUS TRANSMISSION

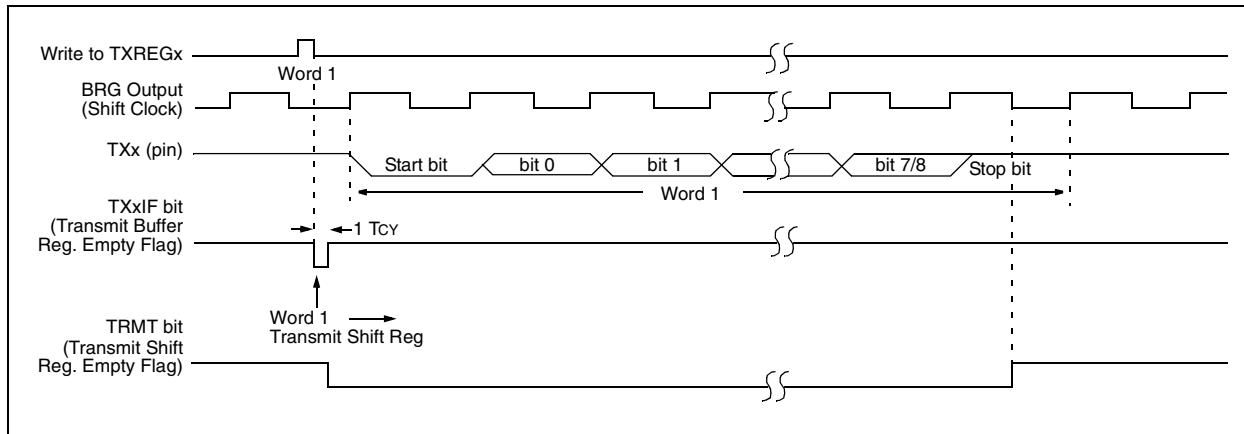


FIGURE 20-5: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)

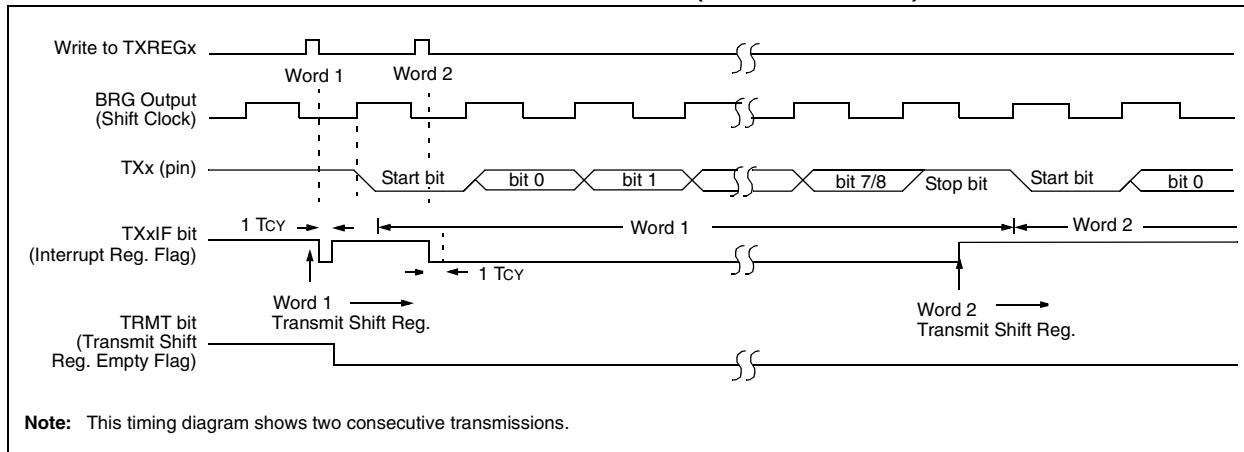


TABLE 20-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PPMIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PPMIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PPMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
TXREGx	EUSARTx Transmit Register								57
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDDB	BRGH	TRMT	TX9D	57
BAUDCONx	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	59
SPBRGHx	EUSARTx Baud Rate Generator Register High Byte								59
SPBRGx	EUSARTx Baud Rate Generator Register Low Byte								59

Legend: — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.

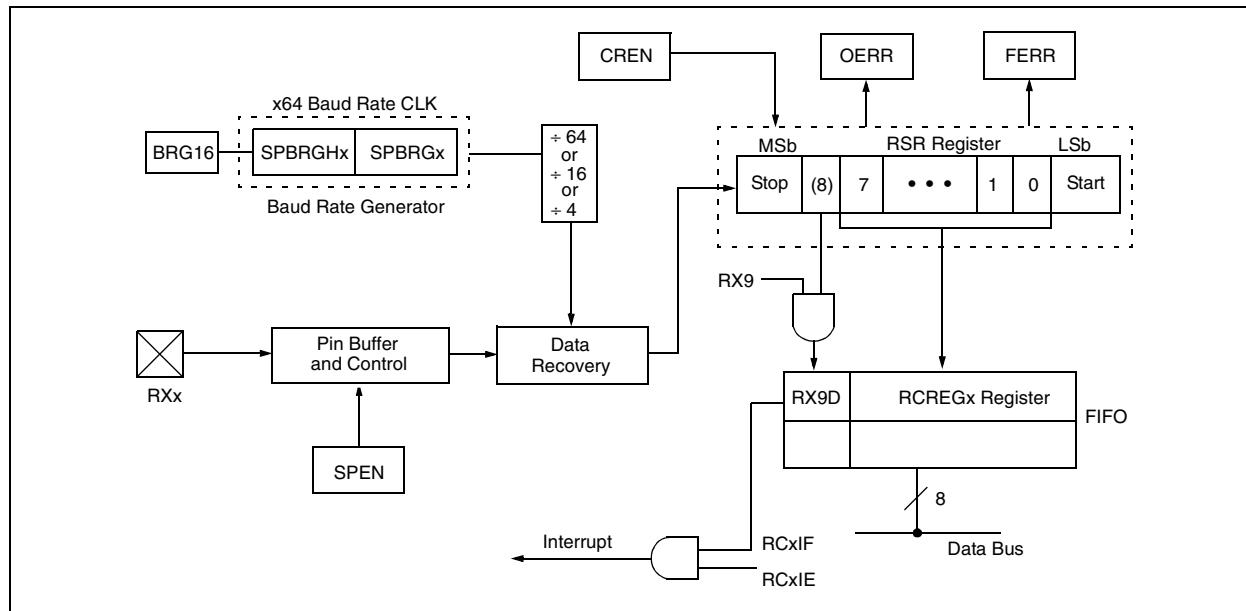
20.2.2 EUSART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 20-6. The data is received on the RXx 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. This mode would typically be used in RS-232 systems.

To set up an Asynchronous Reception:

1. Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit, SYNC, and setting bit, SPEN.
3. If interrupts are desired, set enable bit, RCxIE.
4. If 9-bit reception is desired, set bit, RX9.
5. Enable the reception by setting bit, CREN.
6. Flag bit, RCxIF, will be set when reception is complete and an interrupt will be generated if enable bit, RCxIE, was set.
7. Read the RCSTAx register to get the 9th bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREGx register.
9. If any error occurred, clear the error by clearing enable bit, CREN.
10. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

FIGURE 20-6: EUSART RECEIVE BLOCK DIAGRAM



20.2.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

1. Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are required, set the RCEN bit and select the desired priority level with the RCxIP bit.
4. Set the RX9 bit to enable 9-bit reception.
5. Set the ADDEN bit to enable address detect.
6. Enable reception by setting the CREN bit.
7. The RCxIF bit will be set when reception is complete. The interrupt will be Acknowledged if the RCxIE and GIE bits are set.
8. Read the RCSTAx register to determine if any error occurred during reception, as well as read bit 9 of data (if applicable).
9. Read RCREGx to determine if the device is being addressed.
10. If any error occurred, clear the CREN bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and interrupt the CPU.

PIC18F87J11 FAMILY

FIGURE 20-7: ASYNCHRONOUS RECEPTION

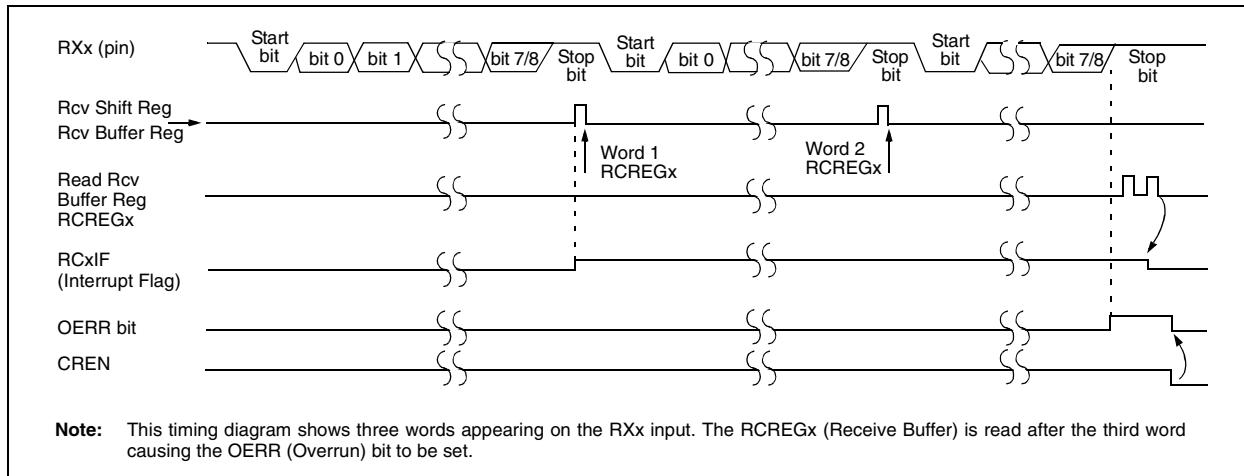


TABLE 20-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PPMIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PPMIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PPMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
RCREGx	EUSARTx Receive Register								57
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
BAUDCONx	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	59
SPBRGHx	EUSARTx Baud Rate Generator Register High Byte								59
SPBRGx	EUSARTx Baud Rate Generator Register Low Byte								59

Legend: — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

20.2.4 AUTO-WAKE-UP ON SYNC BREAK CHARACTER

During Sleep mode, all clocks to the EUSART are suspended. Because of this, the Baud Rate Generator is inactive and a proper byte reception cannot be performed. The auto-wake-up feature allows the controller to wake-up due to activity on the RXx/DTx line while the EUSART is operating in Asynchronous mode.

The auto-wake-up feature is enabled by setting the WUE bit (BAUDCONx<1>). Once set, the typical receive sequence on RXx/DTx is disabled and the EUSART remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a high-to-low transition on

the RXx/DTx line. (This coincides with the start of a Sync Break or a Wake-up Signal character for the LIN protocol.)

Following a wake-up event, the module generates an RCxIF interrupt. The interrupt is generated synchronously to the Q clocks in normal operating modes (Figure 20-8) and asynchronously if the device is in Sleep mode (Figure 20-9). The interrupt condition is cleared by reading the RCREGx register.

The WUE bit is automatically cleared once a low-to-high transition is observed on the RXx line following the wake-up event. At this point, the EUSART module is in Idle mode and returns to normal operation. This signals to the user that the Sync Break event is over.

20.2.4.1 Special Considerations Using Auto-Wake-up

Since auto-wake-up functions by sensing rising edge transitions on RXx/DTx, information with any state changes before the Stop bit may signal a false End-of-Character (EOC) and cause data or framing errors. To work properly, therefore, the initial character in the transmission must be all '0's. This can be 00h (8 bytes) for standard RS-232 devices or 000h (12 bits) for LIN bus.

Oscillator start-up time must also be considered, especially in applications using oscillators with longer start-up intervals (i.e., HS or HSPLL mode). The Sync Break (or Wake-up Signal) character must be of sufficient length and be followed by a sufficient interval to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART.

20.2.4.2 Special Considerations Using the WUE Bit

The timing of WUE and RCxIF events may cause some confusion when it comes to determining the validity of received data. As noted, setting the WUE bit places the EUSART in an Idle mode. The wake-up event causes a receive interrupt by setting the RCxIF bit. The WUE bit is cleared after this when a rising edge is seen on RXx/DTx. The interrupt condition is then cleared by reading the RCREGx register. Ordinarily, the data in RCREGx will be dummy data and should be discarded.

The fact that the WUE bit has been cleared (or is still set) and the RCxIF flag is set should not be used as an indicator of the integrity of the data in RCREGx. Users should consider implementing a parallel method in firmware to verify received data integrity.

To assure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

FIGURE 20-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING NORMAL OPERATION

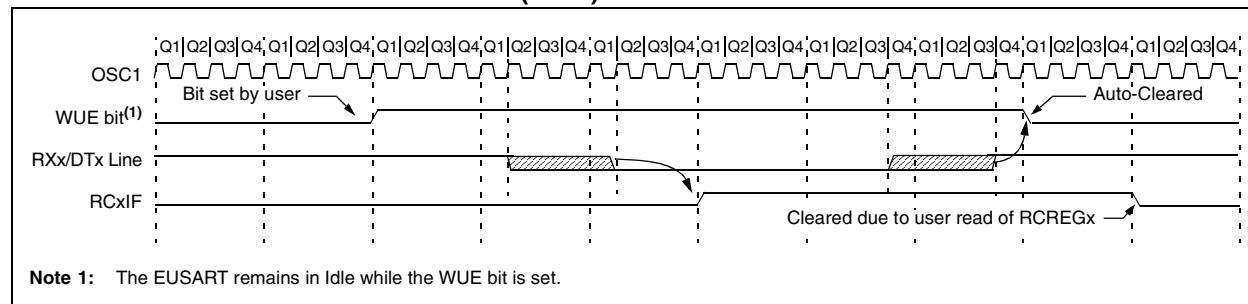
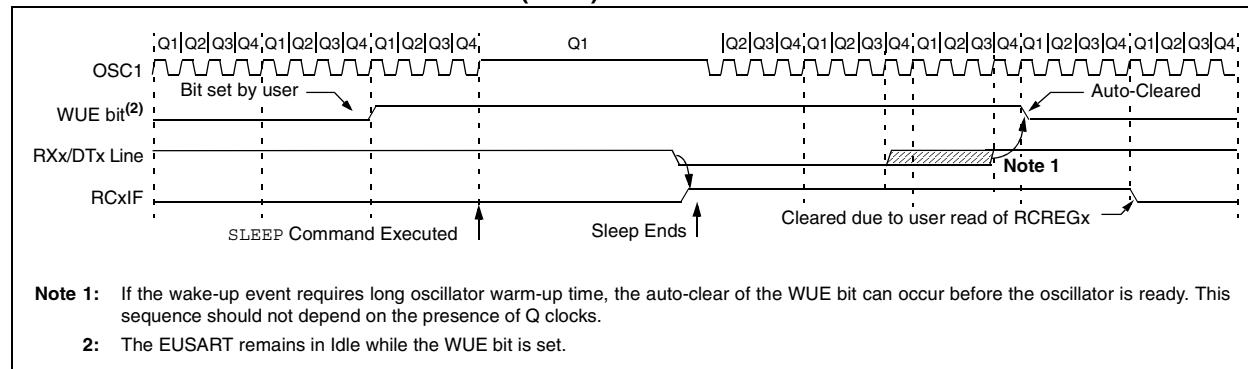


FIGURE 20-9: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP



PIC18F87J11 FAMILY

20.2.5 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. The Break character transmit consists of a Start bit, followed by twelve ‘0’ bits and a Stop bit. The Frame Break character is sent whenever the SENDB and TXEN bits (TXSTAx<3> and TXSTAx<5>) are set while the Transmit Shift Register is loaded with data. Note that the value of data written to TXREGx will be ignored and all ‘0’s will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

Note that the data value written to the TXREGx for the Break character is ignored. The write simply serves the purpose of initiating the proper sequence.

The TRMT bit indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 20-10 for the timing of the Break character sequence.

20.2.5.1 Break and Sync Transmit Sequence

The following sequence will send a message frame header made up of a Break, followed by an Auto-Baud Sync byte. This sequence is typical of a LIN bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENDB bits to set up the Break character.
3. Load the TXREGx with a dummy character to initiate transmission (the value is ignored).
4. Write ‘55h’ to TXREGx to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware. The Sync character now transmits in the preconfigured mode.

When the TXREGx becomes empty, as indicated by the TXxIF, the next data byte can be written to TXREGx.

20.2.6 RECEIVING A BREAK CHARACTER

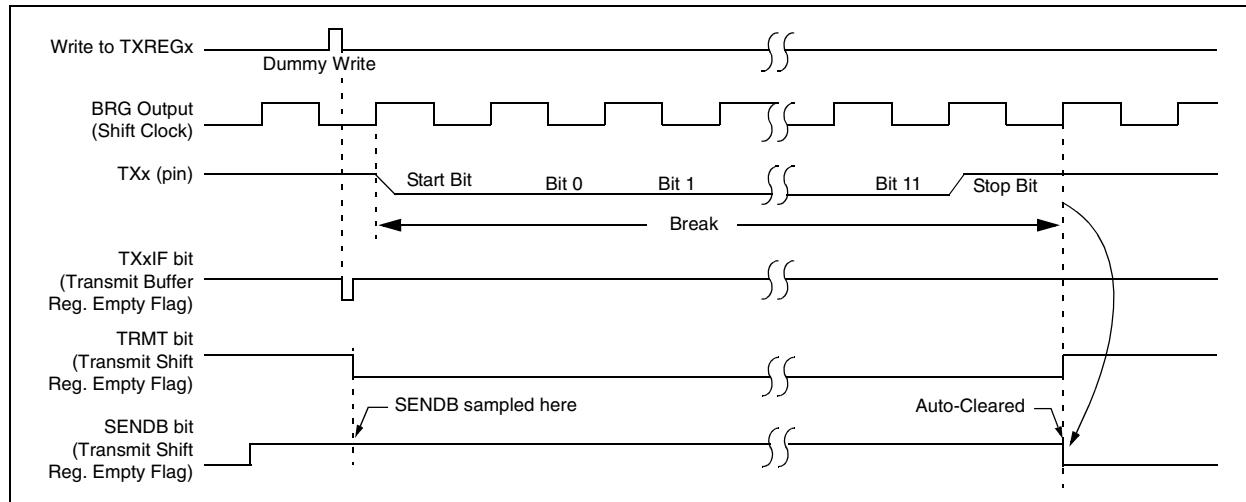
The Enhanced USART module can receive a Break character in two ways.

The first method forces configuration of the baud rate at a frequency of 9/13 the typical speed. This allows for the Stop bit transition to be at the correct sampling location (13 bits for Break versus Start bit and 8 data bits for typical data).

The second method uses the auto-wake-up feature described in **Section 20.2.4 “Auto-Wake-up on Sync Break Character”**. By enabling this feature, the EUSART will sample the next two transitions on RXx/DTx, cause an RCxIF interrupt and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Rate Detect feature. For both methods, the user can set the ABDEN bit once the TXxIF interrupt is observed.

FIGURE 20-10: SEND BREAK CHARACTER SEQUENCE



20.3 EUSART Synchronous Master Mode

The Synchronous Master mode is entered by setting the CSRC bit (TXSTAx<7>). In this 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 (TXSTAx<4>). In addition, enable bit, SPEN (RCSTAx<7>), is set in order to configure the TXx and RXx pins to CKx (clock) and DTx (data) lines, respectively.

The Master mode indicates that the processor transmits the master clock on the CKx line. Clock polarity is selected with the SCKP bit (BAUDCONx<4>). Setting SCKP sets the Idle state on CKx as high, while clearing the bit sets the Idle state as low. This option is provided to support Microwire devices with this module.

20.3.1 EUSART SYNCHRONOUS MASTER TRANSMISSION

The EUSART transmitter block diagram is shown in Figure 20-3. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The Shift register obtains its data from the Read/Write Transmit Buffer register, TXREGx. The TXREGx 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 TXREGx (if available).

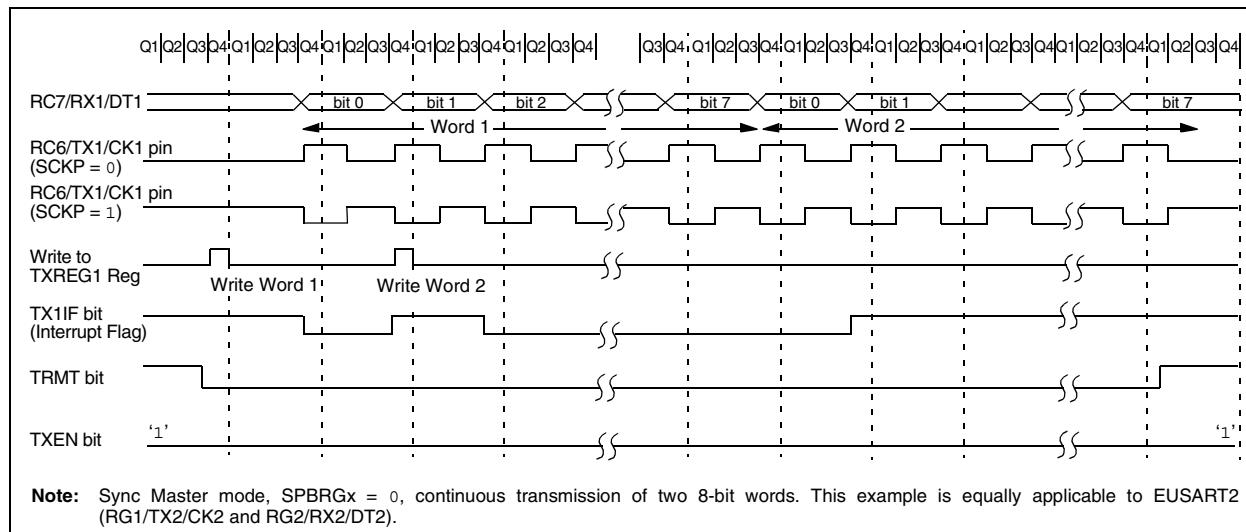
Once the TXREGx register transfers the data to the TSR register (occurs in one Tcy), the TXREGx is empty and the TXxIF flag bit is set. The interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TXxE. TXxE is set regardless of the state of enable bit, TXxE; it cannot be cleared in software. It will reset only when new data is loaded into the TXREGx register.

While flag bit, TXxE, indicates the status of the TXREGx register, another bit, TRMT (TXSTAx<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 must 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.

To set up a Synchronous Master Transmission:

1. Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRG16 bit, as required, to achieve the desired baud rate.
2. Enable the synchronous master serial port by setting bits, SYNC, SPEN and CSRC.
3. If interrupts are desired, set enable bit, TXxE.
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 TXREGx register.
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

FIGURE 20-11: SYNCHRONOUS TRANSMISSION



PIC18F87J11 FAMILY

FIGURE 20-12: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)

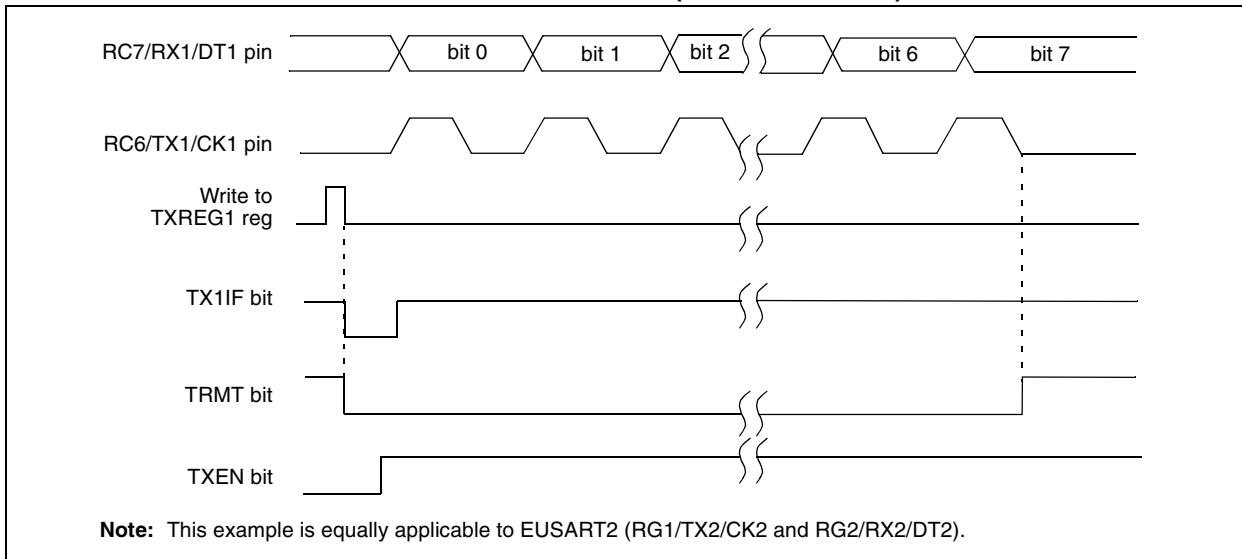


TABLE 20-7: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	MPPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	MPPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	MPPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
TXREGx	EUSARTx Transmit Register								57
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
BAUDCONx	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	59
SPBRGHx	EUSARTx Baud Rate Generator Register High Byte								59
SPBRGx	EUSARTx Baud Rate Generator Register Low Byte								59

Legend: — = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.

20.3.2 EUSART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either the Single Receive Enable bit, SREN (RCSTAx<5>) or the Continuous Receive Enable bit, CREN (RCSTAx<4>). Data is sampled on the RXx pin on the falling edge of the clock.

If enable bit, SREN, is set, only a single word is received. If enable bit, CREN, is set, the reception is continuous until CREN is cleared. If both bits are set, then CREN takes precedence.

To set up a Synchronous Master Reception:

1. Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRG16 bit, as required, to achieve the desired baud rate.
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, set enable bit, RCxIE.
5. If 9-bit reception is desired, set bit, RX9.
6. If a single reception is required, set bit, SREN. For continuous reception, set bit, CREN.
7. Interrupt flag bit, RCxIF, will be set when reception is complete and an interrupt will be generated if the enable bit, RCxIE, was set.
8. Read the RCSTAx register to get the 9th bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREGx register.
10. If any error occurred, clear the error by clearing bit CREN.
11. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

FIGURE 20-13: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)

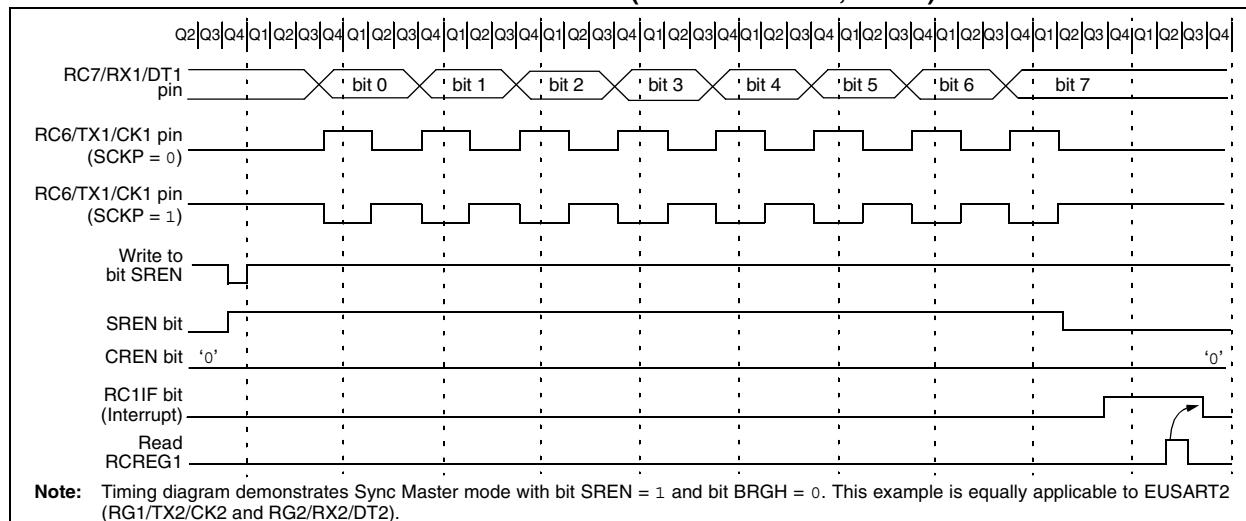


TABLE 20-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	MPPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	MPPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	MPPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
RCREGx	EUSARTx Receive Register								57
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
BAUDCONx	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	59
SPBRGHx	EUSARTx Baud Rate Generator Register High Byte								59
SPBRGx	EUSARTx Baud Rate Generator Register Low Byte								59

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

PIC18F87J11 FAMILY

20.4 EUSART Synchronous Slave Mode

Synchronous Slave mode is entered by clearing bit, CSRC (TXSTAX<7>). This mode differs from the Synchronous Master mode in that the shift clock is supplied externally at the CKx pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in any low-power mode.

20.4.1 EUSART SYNCHRONOUS SLAVE TRANSMISSION

The operation of the Synchronous Master and Slave modes is identical, except in the case of Sleep mode.

If two words are written to the TXREGx 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 the TXREGx register.
- Flag bit, TXxIF, will not be set.
- When the first word has been shifted out of TSR, the TXREGx register will transfer the second word to the TSR and flag bit, TXxIF, will now be set.
- If enable bit, TXxE, is set, the interrupt will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Transmission:

- 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, set enable bit, TXxE.
- If 9-bit transmission is desired, 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 TXREGx register.
- If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

TABLE 20-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	MPPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	MPPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	MPPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
TXREGx	EUSARTx Transmit Register								57
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
BAUDCONx	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	59
SPBRGHx	EUSARTx Baud Rate Generator Register High Byte								59
SPBRGx	EUSARTx Baud Rate Generator Register Low Byte								59

Legend: — = unimplemented, read as '0'. Shaded cells are not used for synchronous slave transmission.

20.4.2 EUSART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of Sleep, or any Idle mode and bit, SREN, which is a “don’t care” in Slave mode.

If receive is enabled by setting the CREN bit prior to entering Sleep or any Idle mode, then a word may be received while in this low-power mode. Once the word is received, the RSR register will transfer the data to the RCREGx register. If the RCxIE enable bit is set, the interrupt generated will wake the chip from the low-power mode. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Reception:

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

TABLE 20-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
RCREGx	EUSARTx Receive Register								57
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
BAUDCONx	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	59
SPBRGHx	EUSARTx Baud Rate Generator Register High Byte								59
SPBRGx	EUSARTx Baud Rate Generator Register Low Byte								59

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used for synchronous slave reception.

PIC18F87J11 FAMILY

NOTES:

21.0 10-BIT ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) Converter module has 11 inputs for the 64-pin devices and 15 for the 80-pin devices. This module allows conversion of an analog input signal to a corresponding 10-bit digital number.

The module has six registers:

- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)

- A/D Port Configuration Register 2 (ANCON0)
- A/D Port Configuration Register 1 (ANCON1)
- A/D Result Registers (ADRESH and ADRESL)

The ADCON0 register, shown in Register 21-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 21-2, configures the A/D clock source, programmed acquisition time and justification.

REGISTER 21-1: ADCON0: A/D CONTROL REGISTER 0⁽¹⁾

R/W-0	R/W-0						
VCFG1	VCFG0	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
bit 7							bit 0

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

bit 7-6 **VCFG1:** Voltage Reference Configuration bit (VREF- source)

1 = VREF- (AN2)

0 = AVSS

bit **VCFG0:** Voltage Reference Configuration bit (VREF+ source)

1 = VREF+ (AN3)

0 = AVDD

bit 5-2 **CHS3:CHS0:** Analog Channel Select bits

0000 = Channel 00 (AN0)

0001 = Channel 01 (AN1)

0010 = Channel 02 (AN2)

0011 = Channel 03 (AN3)

0100 = Channel 04 (AN4)

0101 = Unused

0110 = Channel 06 (AN6)

0111 = Channel 07 (AN7)

1000 = Channel 08 (AN8)

1001 = Channel 09 (AN9)

1010 = Channel 10 (AN10)

1011 = Channel 11 (AN11)

1100 = Channel 12 (AN12)^(2,3)

1101 = Channel 13 (AN13)^(2,3)

1110 = Channel 14 (AN14)^(2,3)

1111 = Channel 15 (AN15)^(2,3)

bit 1 **GO/DONE:** A/D Conversion Status bit

When ADON = 1:

1 = A/D conversion in progress

0 = A/D Idle

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

1 = A/D Converter module is enabled

0 = A/D Converter module is disabled

Note 1: Default (legacy) SFR at this address, available when WDTCON<4> = 0.

2: These channels are not implemented on 64-pin devices.

3: Performing a conversion on unimplemented channels will return random values.

PIC18F87J11 FAMILY

REGISTER 21-2: ADCON1: A/D CONTROL REGISTER 1⁽¹⁾

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| ADFM | ADCAL | ACQT2 | ACQT1 | ACQT0 | ADCS2 | ADCS1 | ADCS0 |
| bit 7 | bit 0 | | | | | | |

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

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

1 = Right justified

0 = Left justified

bit 6 **ADCAL:** A/D Calibration bit

1 = Calibration is performed on next A/D conversion

0 = Normal A/D Converter operation (no conversion is performed)

bit 5-3 **ACQT2:ACQT0:** A/D Acquisition Time Select bits

111 = 20 TAD

110 = 16 TAD

101 = 12 TAD

100 = 8 TAD

011 = 6 TAD

010 = 4 TAD

001 = 2 TAD

000 = 0 TAD⁽¹⁾

bit 2-0 **ADCS2:ADCS0:** A/D Conversion Clock Select bits

111 = FRC (clock derived from A/D RC oscillator)⁽²⁾

110 = FOSC/64

101 = Fosc/16

100 = FOSC/4

011 = FRC (clock derived from A/D RC oscillator)⁽²⁾

010 = FOSC/32

001 = Fosc/8

000 = Fosc/2

Note 1: Default (legacy) SFR at this address, available when WDTCON<4> = 0.

2: If the A/D FRC clock source is selected, a delay of one TCY (instruction cycle) is added before the A/D clock starts. This allows the SLEEP instruction to be executed before starting a conversion.

The ANCON0 and ANCON1 registers are used to configure the operation of the I/O pin associated with each analog channel. Setting any one of the PCFG bits configures the corresponding pin to operate as a digital only I/O. Clearing a bit configures the pin to operate as an analog input for either the A/D Converter or the comparator module; all digital peripherals are disabled, and digital inputs read as '0'. As a rule, I/O pins that are multiplexed with analog inputs default to analog operation on device Resets.

ANCON0 and ANCON1 are shared address SFRs, and use the same addresses as the ADCON1 and ADCON0 registers. The ANCON registers are accessed by setting the ADSHR bit (WDTCON<4>). See **Section 5.3.4.1 "Shared Address SFRs"** for more information.

REGISTER 21-3: ANCON0: A/D PORT CONFIGURATION REGISTER 2

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG7	PCFG6	—	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
bit 7	bit 0						

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

bit 7-6 **PCFG7:PCFG6:** Analog Port Configuration bits (AN7 and AN6)

1 = Pin configured as a digital port

0 = Pin configured as an analog channel; digital input disabled and reads '0'

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

bit 4-0 **PCFG4:PCFG0:** Analog Port Configuration bits (AN4 through AN0)

1 = Pin configured as a digital port

0 = Pin configured as an analog channel; digital input disabled and reads '0'

REGISTER 21-4: ANCON1: A/D PORT CONFIGURATION REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG15 ⁽¹⁾	PCFG14 ⁽¹⁾	PCFG13 ⁽¹⁾	PCFG12 ⁽¹⁾	PCFG11	PCFG10	PCFG9	PCFG8
bit 7	bit 0						

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

bit 7-0 **PCFG15:PCFG8:** Analog Port Configuration bits (AN15 through AN8)

1 = Pin configured as a digital port

0 = Pin configured as an analog channel; digital input disabled and reads '0'

Note 1: AN15 through AN12 are implemented only on 80-pin devices. For 64-pin devices, the corresponding PCFGx bits are still implemented for these channels, but have no effect.

PIC18F87J11 FAMILY

The analog reference voltage is software selectable to either the device's positive and negative supply voltage (AVDD and AVSS), or the voltage level on the RA3/AN3/VREF+ and RA2/AN2/VREF- pins.

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 conversion clock must be derived from the A/D's internal RC oscillator.

The output of the sample and hold is the input into the converter, which generates the result via successive approximation.

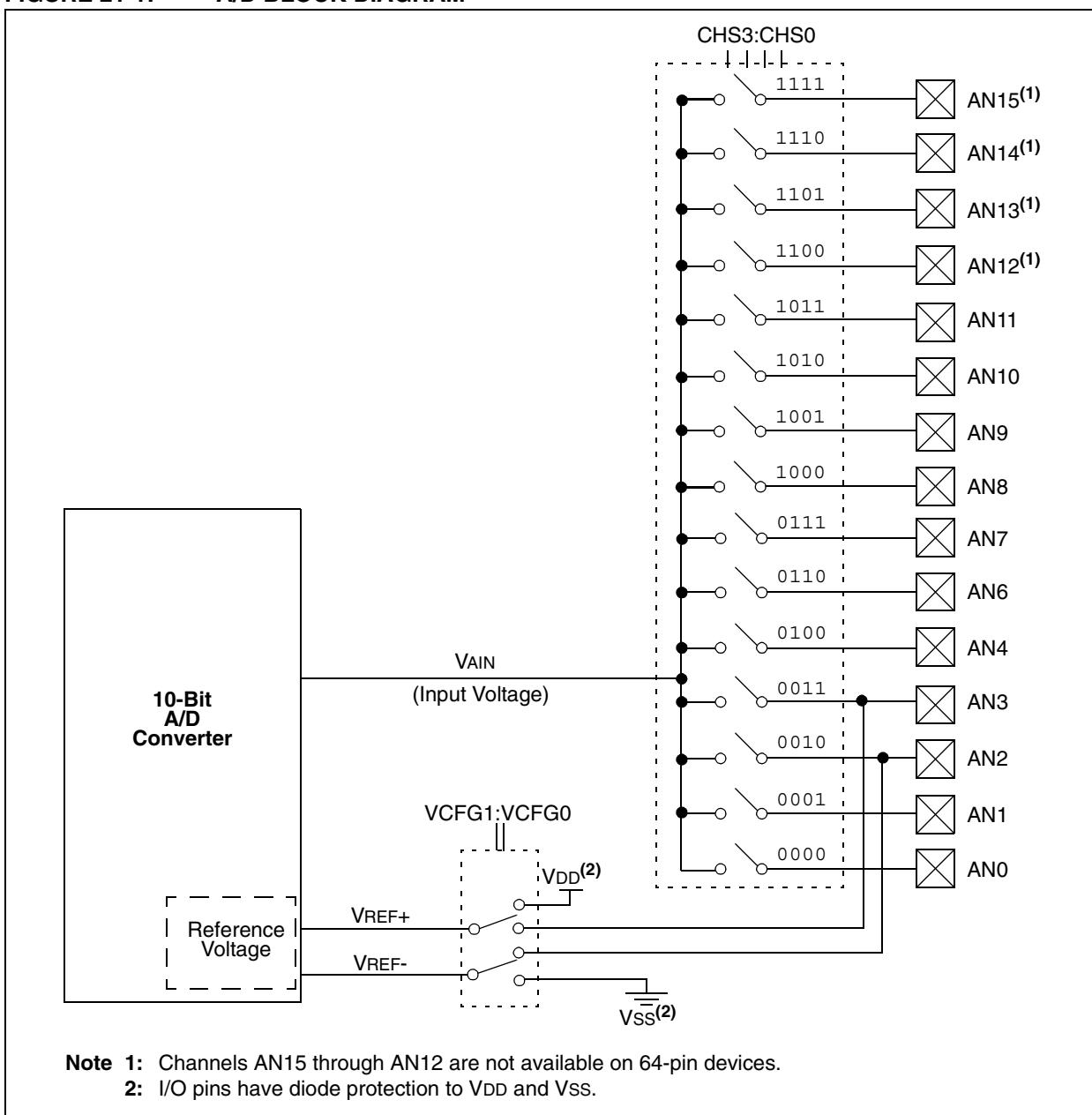
Each port pin associated with the A/D Converter can be configured as an analog input or as a digital I/O. The ADRESH and ADRESL registers contain the result of

the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRESH:ADRESL register pair, the GO/DONE bit (ADCON0<1>) is cleared and A/D Interrupt Flag bit, ADIF, is set.

A device Reset forces all registers to their Reset state. This forces the A/D module to be turned off and any conversion in progress is aborted. The value in the ADRESH:ADRESL register pair is not modified for a Power-on Reset. These registers will contain unknown data after a Power-on Reset.

The block diagram of the A/D module is shown in Figure 21-1.

FIGURE 21-1: A/D BLOCK DIAGRAM



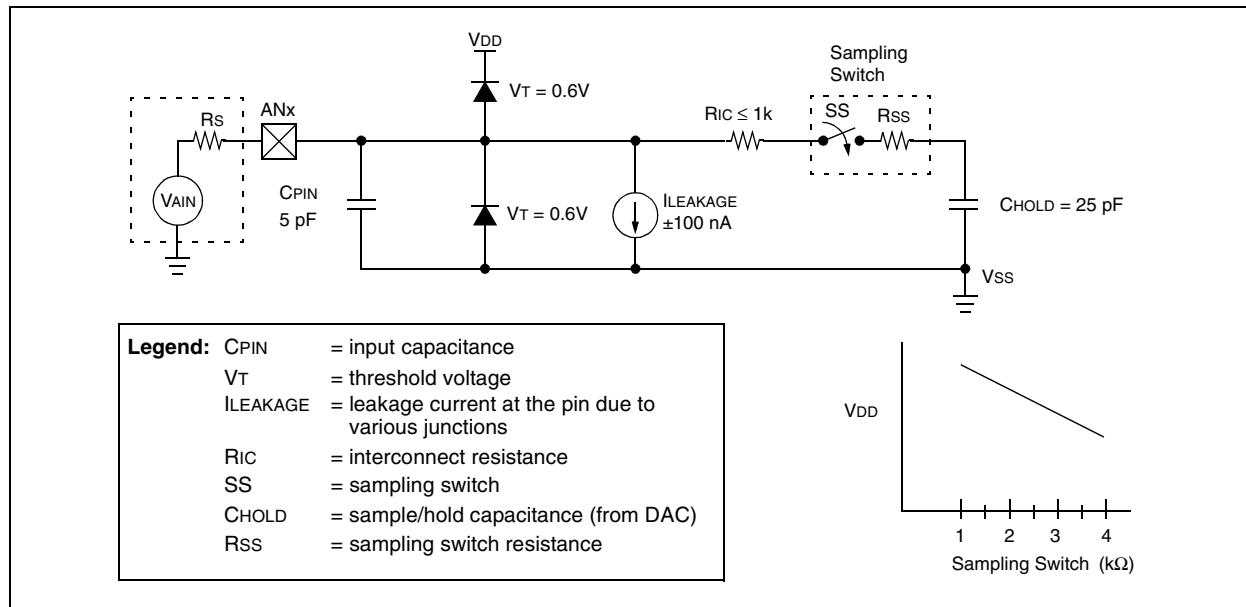
After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as an input. To determine acquisition time, see **Section 21.1 “A/D Acquisition Requirements”**. After this acquisition time has elapsed, the A/D conversion can be started. An acquisition time can be programmed to occur between setting the GO/DONE bit and the actual start of the conversion.

The following steps should be followed to do an A/D conversion:

1. Configure the A/D module:
 - Configure the required ADC pins as analog pins using ANCON0, ANCON1
 - Set voltage reference using ADCON0
 - Select A/D input channel (ADCON0)
 - Select A/D acquisition time (ADCON1)
 - Select A/D conversion clock (ADCON1)
 - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
 - Clear ADIF bit
 - Set ADIE bit
 - Set GIE bit

3. Wait the required acquisition time (if required).
4. Start conversion:
 - Set GO/DONE bit (ADCON0<1>)
5. Wait for A/D conversion to complete, by either:
 - Polling for the GO/DONE bit to be cleared
 - OR
 - Waiting for the A/D interrupt
6. Read A/D Result registers (ADRESH:ADRESL); clear bit, ADIF, if required.
7. For next conversion, go to step 1 or step 2, as required. The A/D conversion time per bit is defined as TAD. A minimum wait of 2 TAD is required before next acquisition starts.

FIGURE 21-2: ANALOG INPUT MODEL



PIC18F87J11 FAMILY

21.1 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 21-2. The source impedance (Rs) and the internal sampling switch (Rss) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (Rss) impedance varies over the device voltage (VDD). The source impedance affects the offset voltage at the analog input (due to pin leakage current). **The maximum recommended impedance for analog sources is 2.5 kΩ.** After the analog input channel is selected (changed), the channel must be sampled for at least the minimum acquisition time before starting a conversion.

Note: When the conversion is started, the holding capacitor is disconnected from the input pin.

To calculate the minimum acquisition time, Equation 21-1 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.

Equation 21-3 shows the calculation of the minimum required acquisition time, TACQ. This calculation is based on the following application system assumptions:

CHOLD	=	25 pF
Rs	=	2.5 kΩ
Conversion Error	≤	1/2 LSb
VDD	=	3V → Rss = 2 kΩ
Temperature	=	85°C (system max.)

EQUATION 21-1: ACQUISITION TIME

$$\begin{aligned} \text{TACQ} &= \text{Amplifier Settling Time} + \text{Holding Capacitor Charging Time} + \text{Temperature Coefficient} \\ &= \text{TAMP} + \text{TC} + \text{TCOFF} \end{aligned}$$

EQUATION 21-2: A/D MINIMUM CHARGING TIME

$$\begin{aligned} \text{VHOLD} &= (\text{VREF} - (\text{VREF}/2048)) \cdot (1 - e^{(-\text{TC}/\text{CHOLD}(\text{RIC} + \text{RSS} + \text{RS}))}) \\ \text{or} \\ \text{TC} &= -(\text{CHOLD})(\text{RIC} + \text{RSS} + \text{RS}) \ln(1/2048) \end{aligned}$$

EQUATION 21-3: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME

$$\begin{aligned} \text{TACQ} &= \text{TAMP} + \text{TC} + \text{TCOFF} \\ \text{TAMP} &= 0.2 \mu\text{s} \\ \text{TCOFF} &= (\text{Temp} - 25^\circ\text{C})(0.02 \mu\text{s}/^\circ\text{C}) \\ &\quad (85^\circ\text{C} - 25^\circ\text{C})(0.02 \mu\text{s}/^\circ\text{C}) \\ &\quad 1.2 \mu\text{s} \\ \text{Temperature coefficient is only required for temperatures } > 25^\circ\text{C. Below } 25^\circ\text{C, TCOFF} &= 0 \text{ ms.} \\ \text{TC} &= -(\text{CHOLD})(\text{RIC} + \text{RSS} + \text{RS}) \ln(1/2048) \mu\text{s} \\ &\quad -(25 \text{ pF})(1 \text{ k}\Omega + 2 \text{ k}\Omega + 2.5 \text{ k}\Omega) \ln(0.0004883) \mu\text{s} \\ &\quad 1.05 \mu\text{s} \\ \text{TACQ} &= 0.2 \mu\text{s} + 1.05 \mu\text{s} + 1.2 \mu\text{s} \\ &= 2.45 \mu\text{s} \end{aligned}$$

21.2 Selecting and Configuring Automatic Acquisition Time

The ADCON1 register allows the user to select an acquisition time that occurs each time the GO/DONE bit is set.

When the GO/DONE bit is set, sampling is stopped and a conversion begins. The user is responsible for ensuring the required acquisition time has passed between selecting the desired input channel and setting the GO/DONE bit. This occurs when the ACQT2:ACQTO bits (ADCON1<5:3>) remain in their Reset state ('000') and is compatible with devices that do not offer programmable acquisition times.

If desired, the ACQT bits can be set to select a programmable acquisition time for the A/D module. When the GO/DONE bit is set, the A/D module continues to sample the input for the selected acquisition time, then automatically begins a conversion. Since the acquisition time is programmed, there may be no need to wait for an acquisition time between selecting a channel and setting the GO/DONE bit.

In either case, when the conversion is completed, the GO/DONE bit is cleared, the ADIF flag is set and the A/D begins sampling the currently selected channel again. If an acquisition time is programmed, there is nothing to indicate if the acquisition time has ended or if the conversion has begun.

21.3 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 11 TAD per 10-bit conversion. The source of the A/D conversion clock is software selectable.

There are seven possible options for TAD:

- 2 Tosc
- 4 Tosc
- 8 Tosc
- 16 Tosc
- 32 Tosc
- 64 Tosc
- Internal RC Oscillator

For correct A/D conversions, the A/D conversion clock (TAD) must be as short as possible but greater than the minimum TAD (see parameter 130 in Table 27-30 for more information).

Table 21-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

TABLE 21-1: TAD vs. DEVICE OPERATING FREQUENCIES

AD Clock Source (TAD)		Maximum Device Frequency
Operation	ADCS2:ADCS0	
2 Tosc	000	2.86 MHz
4 Tosc	100	5.71 MHz
8 Tosc	001	11.43 MHz
16 Tosc	101	22.86 MHz
32 Tosc	010	40.00 MHz
64 Tosc	110	40.00 MHz
RC ⁽²⁾	x11	1.00 MHz ⁽¹⁾

Note 1: The RC source has a typical TAD time of 4 μ s.

2: For device frequencies above 1 MHz, the device must be in Sleep mode for the entire conversion or the A/D accuracy may be out of specification.

21.4 Configuring Analog Port Pins

The ANCON0, ANCON1, TRISA, TRISF and TRISH registers control the operation of the A/D port pins. The port pins needed 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 CHS3:CHS0 bits and the TRIS bits.

Note 1: When reading the PORT register, all pins configured as analog input channels 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 be accurately converted.

2: Analog levels on any pin defined as a digital input may cause the digital input buffer to consume current out of the device's specification limits.

PIC18F87J11 FAMILY

21.5 A/D Conversions

Figure 21-3 shows the operation of the A/D Converter after the GO/DONE bit has been set and the ACQT2:ACQT0 bits are cleared. A conversion is started after the following instruction to allow entry into Sleep mode before the conversion begins.

Figure 21-4 shows the operation of the A/D Converter after the GO/DONE bit has been set, the ACQT2:ACQT0 bits are set to '010' and selecting a 4 TAD acquisition time before the conversion starts.

Clearing the GO/DONE bit during a conversion will abort the current conversion. The A/D Result register pair will NOT be updated with the partially completed A/D conversion sample. This means 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 completed or aborted, a 2 TAD wait is required before the next acquisition can be started. After this wait, acquisition on the selected channel is automatically started.

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

FIGURE 21-3: A/D CONVERSION TAD CYCLES (ACQT2:ACQT0 = 000, TACQ = 0)

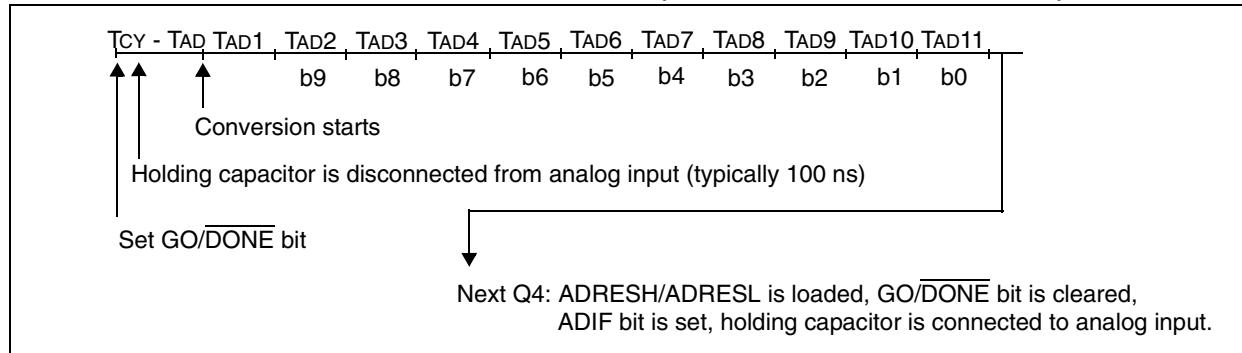
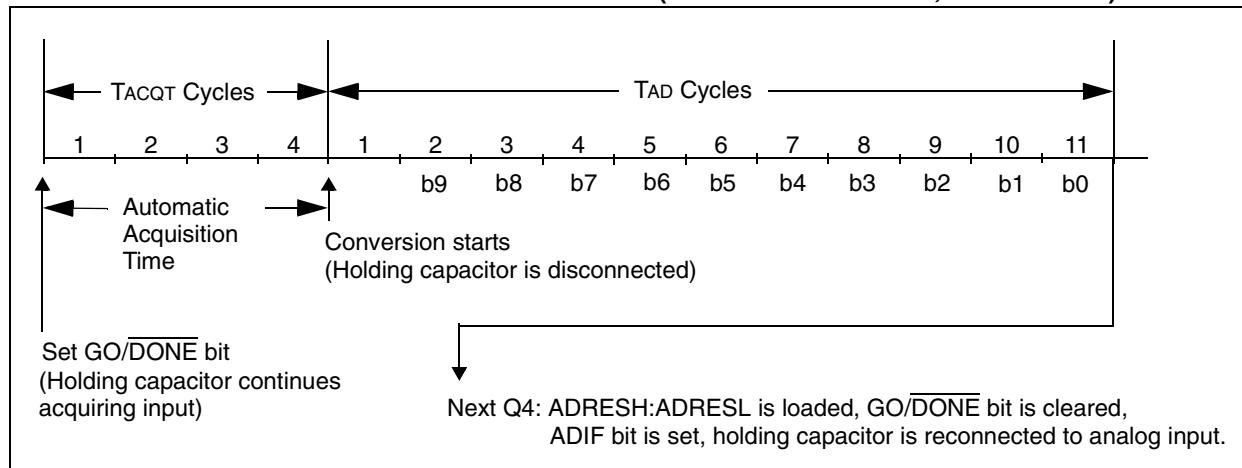


FIGURE 21-4: A/D CONVERSION TAD CYCLES (ACQT2:ACQT0 = 010, TACQ = 4 TAD)



21.7 A/D Converter Calibration

The A/D Converter in the PIC18F87J11 family of devices includes a self-calibration feature which compensates for any offset generated within the module. The calibration process is automated and is initiated by setting the ADCAL bit (ADCON1<6>). The next time the GO/DONE bit is set, the module will perform a “dummy” conversion (that is, with reading none of the input channels) and store the resulting value internally to compensate for the offset. Thus, subsequent offsets will be compensated. An example of a calibration routine is shown in Example 21-1.

The calibration process assumes that the device is in a relatively steady-state operating condition. If A/D calibration is used, it should be performed after each device Reset or if there are other major changes in operating conditions.

21.8 Operation in Power-Managed Modes

The selection of the automatic acquisition time and A/D conversion clock is determined in part by the clock source and frequency while in a power-managed mode.

If the A/D is expected to operate while the device is in a power-managed mode, the ACQT2:ACQT0 and ADCS2:ADCS0 bits in ADCON1 should be updated in accordance with the power-managed mode clock that will be used. After the power-managed mode is entered (either of the power-managed Run modes), an A/D acquisition or conversion may be started. Once an acquisition or conversion is started, the device should continue to be clocked by the same power-managed mode clock source until the conversion has been completed. If desired, the device may be placed into the corresponding power-managed Idle mode during the conversion.

If the power-managed mode clock frequency is less than 1 MHz, the A/D RC clock source should be selected.

Operation in the Sleep mode requires the A/D RC clock to be selected. If bits, ACQT2:ACQT0, are set to ‘000’ and a conversion is started, the conversion will be delayed one instruction cycle to allow execution of the SLEEP instruction and entry to Sleep mode. The IDLEN and SCS bits in the OSCCON register must have already been cleared prior to starting the conversion.

EXAMPLE 21-1: SAMPLE A/D CALIBRATION ROUTINE

BSF	WDTCON,ADSHR	;Enable write/read to the shared SFR
BCF	ANCON0,PCFG0	;Make Channel 0 analog
BCF	WDTCON,ADSHR	;Disable write/read to the shared SFR
BSF	ADCON0,ADON	;Enable A/D module
BSF	ADCON1,ADCAL	;Enable Calibration
BSF	ADCON0,GO	;Start a dummy A/D conversion
CALIBRATION		;
BTFSC	ADCON0,GO	;Wait for the dummy conversion to finish
BRA	CALIBRATION	;
BCF	ADCON1,ADCAL	;Calibration done, turn off calibration enable
		;Proceed with the actual A/D conversion

PIC18F87J11 FAMILY

TABLE 21-2: SUMMARY OF A/D REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	58
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	58
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	58
ADRESH	A/D Result Register High Byte								57
ADRESL	A/D Result Register Low Byte								57
ADC0N0 ⁽²⁾	VCFG1	VCFG0	CHS3	CHS3	CHS1	CHS0	GO/DONE	ADON	57
ANCON0 ⁽³⁾	PCFG7	PCFG6	—	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	57
ADC0N1 ⁽²⁾	ADFM	ADCAL	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	57
ANCON1 ⁽³⁾	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	57
CCP2CON	P2M1	P2M0	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	57
PORTA	RA7 ⁽⁴⁾	RA6 ⁽⁴⁾	RA5	RA4	RA3	RA2	RA1	RA0	59
TRISA	TRISA7 ⁽⁴⁾	TRISA6 ⁽⁴⁾	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	58
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	—	59
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	58
PORTH ⁽¹⁾	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0	59
TRISH ⁽¹⁾	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	58

Legend: — = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.

Note 1: This register is not implemented on 64-pin devices.

2: Default (legacy) SFR at this address, available when WDTCON<4> = 0.

3: Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.

4: These bits are only available in select oscillator modes (FOSC2 Configuration bit = 0); otherwise, they are unimplemented.

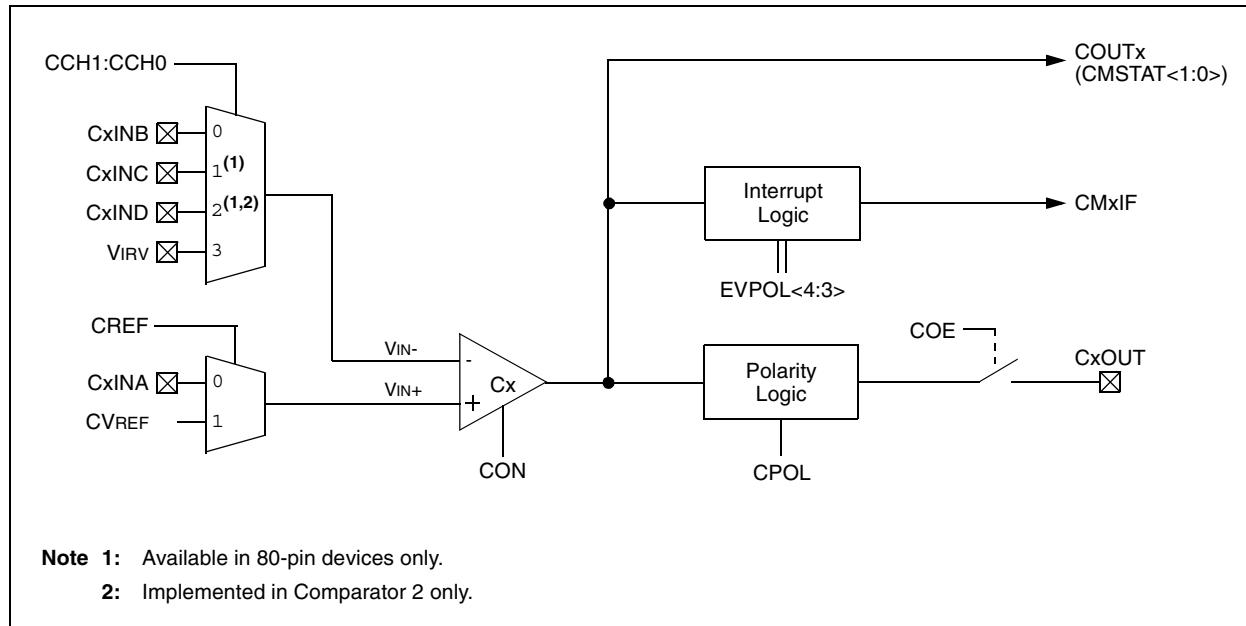
22.0 COMPARATOR MODULE

The analog comparator module contains two comparators that can be independently configured in a variety of ways. The inputs can be selected from the analog inputs and two internal voltage references. The digital outputs are available at the pin level and can also be read through the control register. Multiple output and interrupt event generation are also available. A generic single comparator from the module is shown in Figure 22-1.

Key features of the module includes:

- Independent comparator control
- Programmable input configuration
- Output to both pin and register levels
- Programmable output polarity
- Independent interrupt generation for each comparator with configurable interrupt-on-change

FIGURE 22-1: COMPARATOR SIMPLIFIED BLOCK DIAGRAM



22.1 Registers

The CMxCON registers (Register 22-1) select the input and output configuration for each comparator, as well as the settings for interrupt generation.

The CMSTAT register (Register 22-2) provides the output results of the comparators. The bits in this register are read-only.

PIC18F87J11 FAMILY

REGISTER 22-1: CMxCON: COMPARATORx CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
CON	COE	CPOL	EVPOL1	EVPOLO	CREF	CCH1	CCH0
bit 7	bit 0						

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

- bit 7 **CON:** Comparator Enable bit
1 = Comparator is enabled
0 = Comparator is disabled
- bit 6 **COE:** Comparator Output Enable bit
1 = Comparator output is present on the CxOUT pin
0 = Comparator output is internal only
- bit 5 **CPOL:** Comparator Output Polarity Select bit
1 = Comparator output is inverted
0 = Comparator output is not inverted
- bit 4-3 **EVPOL1:EVPOLO:** Interrupt Polarity Select bits
11 = Interrupt generation on any change of the output⁽¹⁾
10 = Interrupt generation only on high-to-low transition of the output
01 = Interrupt generation only on low-to-high transition of the output
00 = Interrupt generation is disabled
- bit 2 **CREF:** Comparator Reference Select bit (non-inverting input)
1 = Non-inverting input connects to internal CVREF voltage
0 = Non-inverting input connects to CxINA pin
- bit 1-0 **CCH1:CCH0:** Comparator Channel Select bits
11 = Inverting input of comparator connects to VIRV
10 = Inverting input of comparator connects to CxIND pin⁽²⁾
01 = Inverting input of comparator connects to CxINC pin⁽²⁾
00 = Inverting input of comparator connects to CxINB pin

Note 1: The CMxIF is automatically set any time this mode is selected and must be cleared by the application after the initial configuration.

2: Available in 80-pin devices only.

PIC18F87J11 FAMILY

REGISTER 22-2: CMSTAT: COMPARATOR OUTPUT STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R-1	R-1
—	—	—	—	—	—	COUT2	COUT1
bit 7	bit 0						

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

bit 7-2 **Unimplemented:** Read as '0'bit 1-0 **COUT2:COUT1:** Comparator x Status bitsIf CPOL = 0 (non-inverted polarity):

1 = Comparator's VIN+ > VIN-

0 = Comparator's VIN+ < VIN-

If CPOL = 1 (inverted polarity):

1 = Comparator VIN+ < VIN-

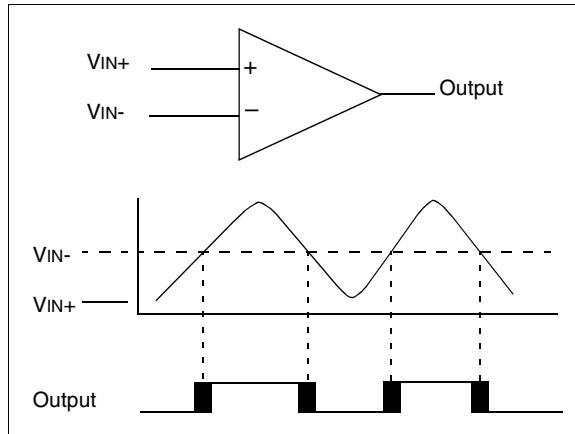
0 = Comparator VIN+ > VIN-

PIC18F87J11 FAMILY

22.2 Comparator Operation

A single comparator is shown in Figure 22-2, 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 22-2 represent the uncertainty due to input offsets and response time.

FIGURE 22-2: SINGLE COMPARATOR



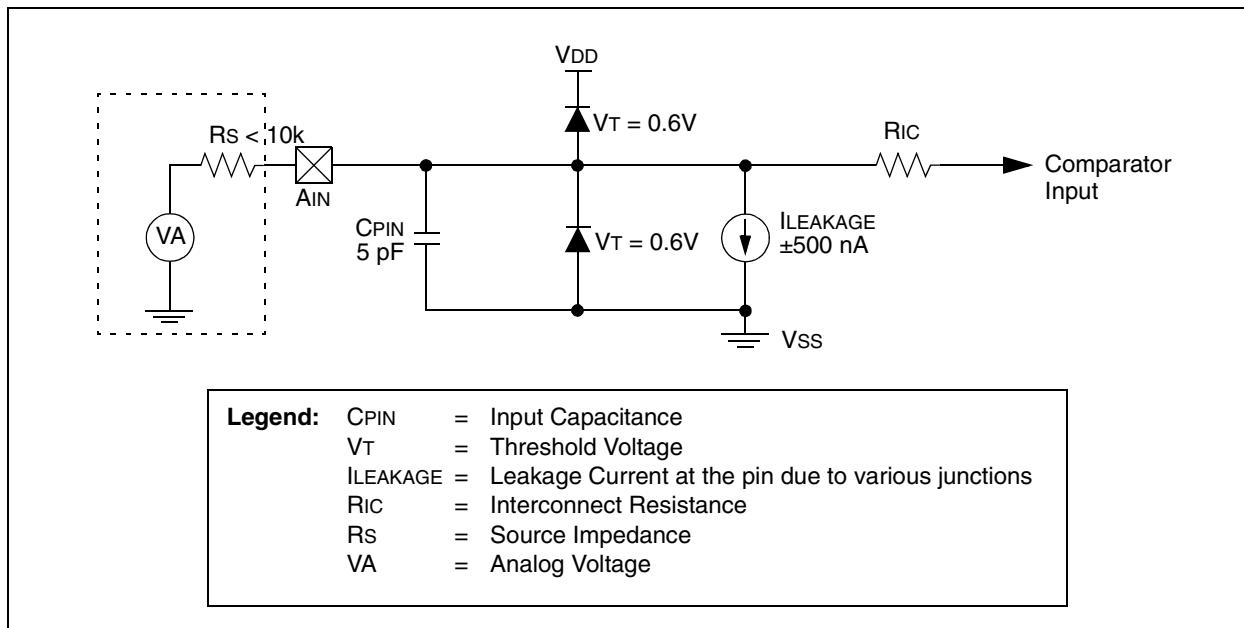
22.3 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output has a valid level. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response to a comparator input change. Otherwise, the maximum delay of the comparators should be used (see Section 27.0 "Electrical Characteristics").

22.4 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 22-3. Since the analog pins are connected to a digital output, they have reverse biased diodes to V_{DD} and V_{SS} . The analog input, therefore, must be between V_{SS} and V_{DD} . 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 22-3: COMPARATOR ANALOG INPUT MODEL



22.5 Comparator Control and Configuration

Each comparator has up to eight possible combinations of inputs: up to four external analog inputs, and one of two internal voltage references.

Both comparators allow a selection of the signal from pin, CxINA, or the voltage from the comparator reference (CVREF) on the non-inverting channel. This is compared to either CxINB, CxINC, CXIND or the microcontroller's fixed internal reference voltage (VIRV, 1.2V nominal) on the inverting channel. The comparator inputs and outputs are tied to fixed I/O pins, defined in Table 22-1. The available configurations and their corresponding bit settings are shown in Figure 22-1.

TABLE 22-1: COMPARATOR INPUTS AND OUTPUTS

Comparator	Input or Output	I/O Pin
1	C1INA (VIN+)	RF6
	C1INB (VIN-)	RF5
	C1INC (VIN-) ⁽¹⁾	RH6 ⁽¹⁾
	C1OUT	RF2
2	C2INA(VIN+)	RF4
	C2INB(VIN-)	RF3
	C2INC(VIN-) ⁽¹⁾	RH4 ⁽¹⁾
	C2IND(VIN-) ⁽¹⁾	RH5 ⁽¹⁾
	C2OUT	RF1

Note 1: Available in 80-pin devices only.

22.5.1 COMPARATOR ENABLE AND INPUT SELECTION

Setting the CON bit of the CMxCON register (CMxCON<7>) enables the comparator for operation. Clearing the CON bit disables the comparator resulting in minimum current consumption.

The CCH1:CCH0 bits in the CMxCON register (CMxCON<1:0>) direct either one of three analog input pins, or the Internal Reference Voltage (VIRV), to the comparator VIN-. Depending on the comparator operating mode, either an external or internal voltage reference may be used. The analog signal present at VIN- is compared to the signal at VIN+ and the digital output of the comparator is adjusted accordingly.

The external reference is used when CREF = 0 (CMxCON<2>) and VIN+ is connected to the CxINA pin. When external voltage references are used, the comparator module can be configured to have the reference sources externally. The reference signal must be between VSS and VDD, and can be applied to either pin of the comparator.

The comparator module also allows the selection of an internally generated voltage reference (CVREF) from the comparator voltage reference module. This module is described in more detail in **Section 23.0 “Comparator Voltage Reference Module”**. The reference from the Comparator Voltage Reference module is only available when CREF = 1. In this mode, the internal voltage reference is applied to the comparator's VIN+ pin.

Note: The comparator input pin selected by CCH1:CH0 must be configured as an input by setting both the corresponding TRISF or TRISH bit, and the corresponding PCFG bit in the ANCON1 register.

22.5.1.1 Comparator Configurations in 64-Pin and 80-Pin Devices

In PIC18F87J11 family devices, the C and D input channels for both comparators are linked to pins in PORTH and cannot be reassigned to alternate analog inputs. Because of this, 64-pin devices offer a total of 4 different configurations for each comparator. In contrast, 80-pin devices offer a choice of 6 configurations for Comparator 1, and 8 configurations for Comparator 2. The configurations shown in Figure 22-1 are footnoted to indicate where they are not available.

22.5.2 COMPARATOR ENABLE AND OUTPUT SELECTION

The comparator outputs are read through the CMSTAT register. The CMSTAT<0> reads the Comparator 1 output and CMSTAT<1> reads the Comparator 2 output. These bits are read-only.

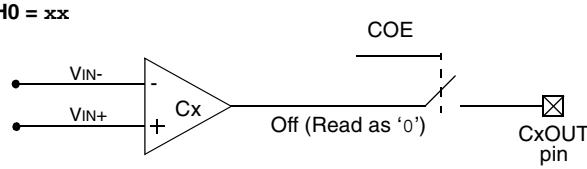
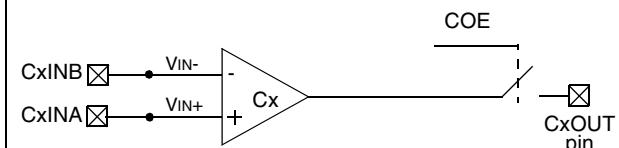
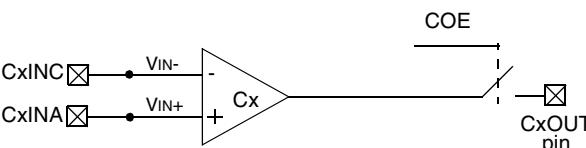
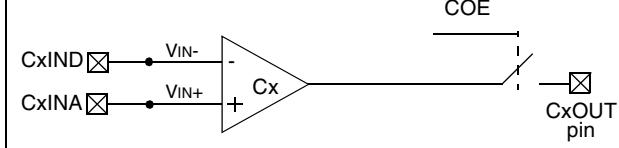
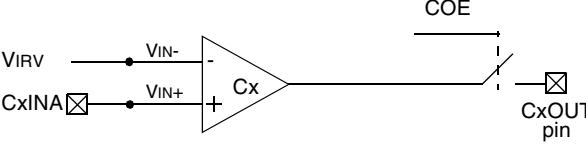
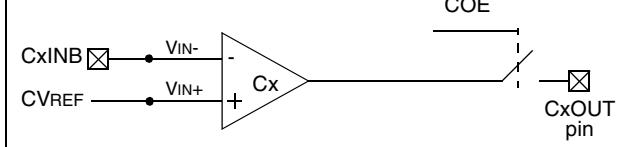
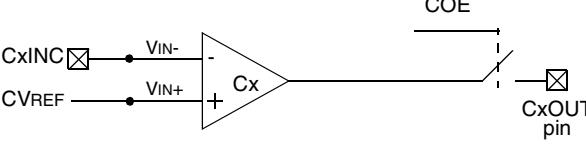
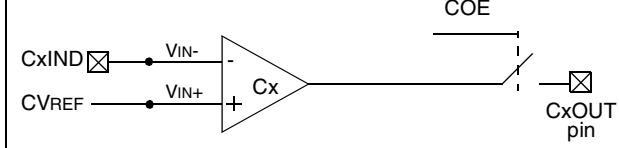
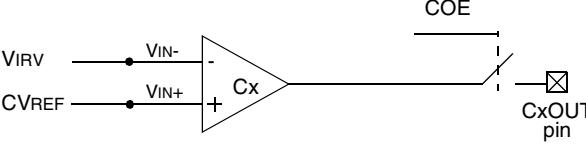
The comparator outputs may also be directly output to the RF1 and RF2 I/O pins by setting the COE bit (CMxCON<6>). When enabled, multiplexors in the output path of the pins switch to the output of the comparator. The TRISF<1:2> bits still function as the digital output enable for the RF1 and RF2 pins while in this mode.

By default, the comparator's output is at logic high whenever the voltage on VIN+ is greater than on VIN-. The polarity of the comparator outputs can be inverted using the CPOL bit (CMxCON<5>).

The uncertainty of each of the comparators is related to the input offset voltage and the response time given in the specifications, as discussed in **Section 22.2 “Comparator Operation”**.

PIC18F87J11 FAMILY

FIGURE 22-4: COMPARATOR I/O CONFIGURATIONS

Comparator Off CON = 0, CREF = x, CCH1:CCH0 = xx	
	
Comparator CxINB > CxINA Compare CON = 1, CREF = 0, CCH1:CCH0 = 00	Comparator CxINC > CxINA Compare⁽¹⁾ CON = 1, CREF = 0, CCH1:CCH0 = 01
	
Comparator CxIND > CxINA Compare^(1,2) CON = 1, CREF = 0, CCH1:CCH0 = 10	Comparator VIRV > CxINA Compare CON = 1, CREF = 0, CCH1:CCH0 = 11
	
Comparator CxINB > CVREF Compare CON = 1, CREF = 1, CCH1:CCH0 = 00	Comparator CxINC > CVREF Compare⁽¹⁾ CON = 1, CREF = 1, CCH1:CCH0 = 01
	
Comparator CxIND > CVREF Compare^(1,2) CON = 1, CREF = 1, CCH1:CCH0 = 10	Comparator VIRV > CVREF Compare CON = 1, CREF = 1, CCH1:CCH0 = 11
	
Legend: VIRV = Fixed Interval Reference Voltage (1.2V nominal), CVREF = Comparator Voltage Reference module output. Configurations are available on both Comparators 1 and 2 in all package sizes unless otherwise noted.	
Note 1: Configuration is available in 80-pin devices only. 2: Configuration is available in Comparator 2 only (80-pin devices).	

22.6 Comparator Interrupts

The comparator interrupt flag is set whenever any of the following occurs:

- Low-to-high transition of the comparator output
- High-to-low transition of the comparator output
- Any change in the comparator output

The comparator interrupt selection is done by the EVPOL1:EVPOL0 bits in the CMxCON register (CMxCON<4:3>).

In order to provide maximum flexibility, the output of the comparator may be inverted using the CPOL bit in the CMxCON register (CMxCON<5>). This is functionally identical to reversing the inverting and non-inverting inputs of the comparator for a particular mode.

An interrupt is generated on the low-to-high or high-to-low transition of the comparator output. This mode of interrupt generation is dependent on EVPOL<1:0> in the CMxCON register. If EVPOL<1:0> = 01 or 10, the interrupt is generated on a low-to-high or high-to-low

transition of the comparator output. Once the interrupt is generated, it is required to clear the interrupt flag by software.

When EVPOL<1:0> = 11, 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 CMSTAT<1:0>, to determine the actual change that occurred. The CMxIF bits (PIR2<6:5>) are the Comparator Interrupt Flags. The CMxIF bits must be reset by clearing them. Since it is also possible to write a '1' to this register, a simulated interrupt may be initiated. Table 22-2 shows the interrupt generation with respect to comparator input voltages and EVPOL bit settings.

Both the CMxIE bits (PIE2<6:5>) and the PEIE bit (INTCON<6>) must be set to enable the interrupt. In addition, the GIE bit (INTCON<7>) must also be set. If any of these bits are clear, the interrupt is not enabled, though the CMxIF bits will still be set if an interrupt condition occurs.

TABLE 22-2: COMPARATOR INTERRUPT GENERATION

CPOL	EVPOL<1:0>	Comparator Input Change	COUTx Transition	Interrupt Generated
0	00	VIN+ > VIN-	Low-to-High	No
		VIN+ < VIN-	High-to-Low	No
	01	VIN+ > VIN-	Low-to-High	Yes
		VIN+ < VIN-	High-to-Low	No
	10	VIN+ > VIN-	Low-to-High	No
		VIN+ < VIN-	High-to-Low	Yes
	11	VIN+ > VIN-	Low-to-High	Yes
		VIN+ < VIN-	High-to-Low	Yes
1	00	VIN+ > VIN-	High-to-Low	No
		VIN+ < VIN-	Low-to-High	No
	01	VIN+ > VIN-	High-to-Low	No
		VIN+ < VIN-	Low-to-High	Yes
	10	VIN+ > VIN-	High-to-Low	Yes
		VIN+ < VIN-	Low-to-High	No
	11	VIN+ > VIN-	High-to-Low	Yes
		VIN+ < VIN-	Low-to-High	Yes

PIC18F87J11 FAMILY

22.7 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. Each operational comparator will consume additional current. To minimize power consumption while in Sleep mode, turn off the comparators (CON = 0) before entering Sleep. If the device wakes up from Sleep, the contents of the CMxCON register are not affected.

22.8 Effects of a Reset

A device Reset forces the CMxCON registers to their Reset state. This forces both comparators and the voltage reference to the OFF state.

TABLE 22-3: REGISTERS ASSOCIATED WITH COMPARATOR MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	58
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	58
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	58
CM1CON	CON	COE	CPOL	EVPOL1	EVPOLO	CREF	CCH1	CCH0	56
CM2CON	CON	COE	CPOL	EVPOL1	EVPOLO	CREF	CCH1	CCH0	56
CMSTAT	—	—	—	—	—	—	COUT2	COUT1	56
CVRCON ⁽²⁾	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	59
ANCON1 ⁽²⁾	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	57
ANCON0 ⁽²⁾	PCFG7	PCFG6	—	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	57
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	—	59
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	—	58
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	58
PORTH ⁽¹⁾	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0	59
TRISH ⁽¹⁾	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	58

Legend: — = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.

Note 1: These registers are not implemented on 64-pin devices.

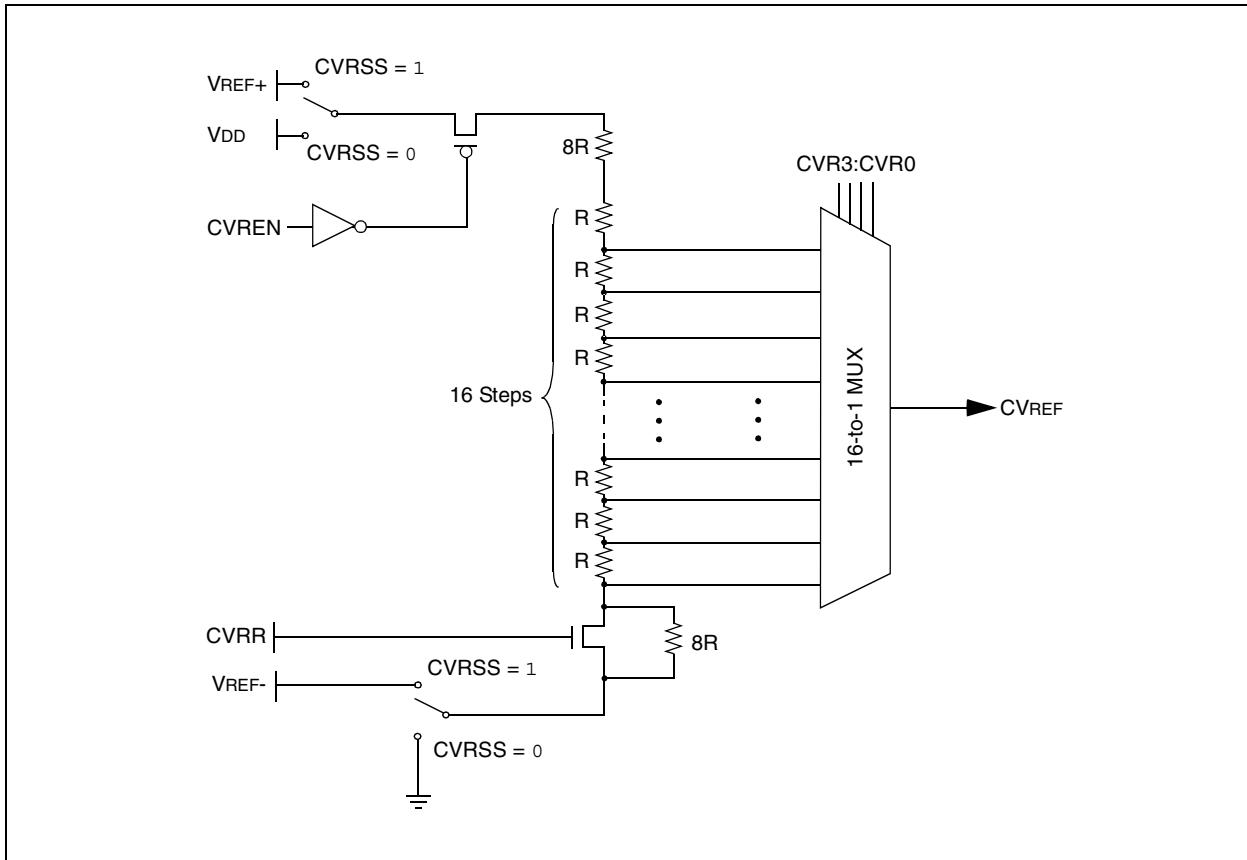
2: Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.

23.0 COMPARATOR VOLTAGE REFERENCE MODULE

The comparator voltage reference is a 16-tap resistor ladder network that provides a selectable reference voltage. Although its primary purpose is to provide a reference for the analog comparators, it may also be used independently of them.

A block diagram of the module is shown in Figure 23-1. 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 module's supply reference can be provided from either device VDD/VSS or an external voltage reference.

FIGURE 23-1: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM



PIC18F87J11 FAMILY

23.1 Configuring the Comparator Voltage Reference

The comparator voltage reference module is controlled through the CVRCON register (Register 23-1). The comparator voltage reference provides two ranges of output voltage, each with 16 distinct levels. The range to be used is selected by the CVRR bit (CVRCON<5>). The primary difference between the ranges is the size of the steps selected by the CVREF Selection bits (CVR3:CVR0), with one range offering finer resolution. The equations used to calculate the output of the comparator voltage reference are as follows:

If CVRR = 1:

$$CVREF = ((CVR3:CVR0)/24) \times (CVRSRC)$$

If CVRR = 0:

$$CVREF = (CVRSRC/4) + ((CVR3:CVR0)/32) \times (CVRSRC)$$

The comparator reference supply voltage can come from either VDD and Vss, or the external VREF+ and VREF- that are multiplexed with RA2 and RA3. The voltage source is selected by the CVRSS bit (CVRCON<4>).

The settling time of the comparator voltage reference must be considered when changing the CVREF output (see Table 27-3 in **Section 27.0 “Electrical Characteristics”**).

The CVRCON register is a shared address SFR and uses the same address as the PR4 register. The CVRCON register is accessed by setting the ADSHR bit (WDTCON<4>).

REGISTER 23-1: CVRCON: COMPARATOR VOLTAGE REFERENCE CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CVREN	CVROE ⁽¹⁾	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
bit 7							bit 0

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

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⁾

1 = CVREF voltage level is also output on the RF5/AN10/C1INB/CVREF pin

0 = CVREF voltage is disconnected from the RF5/AN10/C1INB/CVREF pin

bit 5 **CVRR:** Comparator VREF Range Selection bit

1 = 0 to 0.667 CVRSRC, with CVRSRC/24 step size (low range)

0 = 0.25 CVRSRC to 0.75 CVRSRC, with CVRSRC/32 step size (high range)

bit 4 **CVRSS:** Comparator VREF Source Selection bit

1 = Comparator reference source, CVRSRC = (VREF+) – (VREF-)

0 = Comparator reference source, CVRSRC = AVDD – AVSS

bit 3-0 **CVR3:CVR0:** Comparator VREF Value Selection bits ($0 \leq (CVR3:CVR0) \leq 15$)

When CVRR = 1:

$$CVREF = ((CVR3:CVR0)/24) \bullet (CVRSRC)$$

When CVRR = 0:

$$CVREF = (CVRSRC/4) + ((CVR3:CVR0)/32) \bullet (CVRSRC)$$

Note 1: CVROE overrides the TRISF<5> bit setting.

23.2 Voltage Reference Accuracy/Error

The full range of voltage reference cannot be realized due to the construction of the module. The transistors on the top and bottom of the resistor ladder network (Figure 23-1) keep CVREF from approaching the reference source rails. The voltage reference is derived from the reference source; therefore, the CVREF output changes with fluctuations in that source. The tested absolute accuracy of the voltage reference can be found in **Section 27.0 “Electrical Characteristics”**.

23.3 Connection Considerations

The voltage reference module operates independently of the comparator module. The output of the reference generator may be connected to the RF5 pin if the CVROE bit is set. Enabling the voltage reference output onto RA2 when it is configured as a digital input will increase current consumption. Connecting RF5 as a digital output with CVRSS enabled will also increase current consumption.

The RF5 pin can be used as a simple D/A output with limited drive capability. Due to the limited current drive capability, a buffer must be used on the voltage reference output for external connections to VREF. Figure 23-2 shows an example buffering technique.

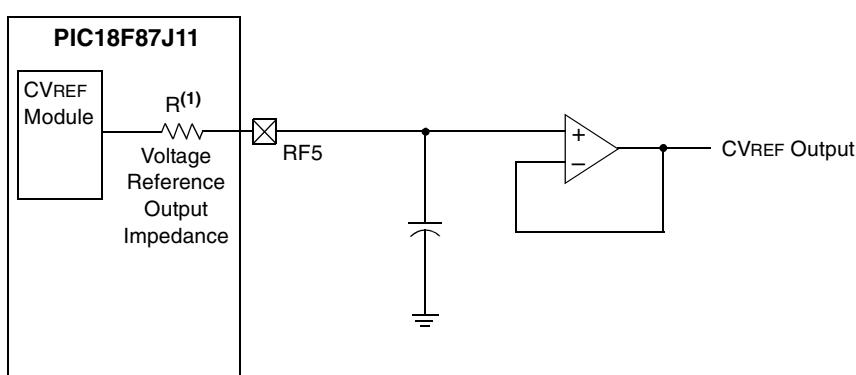
23.4 Operation During Sleep

When the device wakes up from Sleep through an interrupt or a Watchdog Timer time-out, the contents of the CVRCON register are not affected. To minimize current consumption in Sleep mode, the voltage reference should be disabled.

23.5 Effects of a Reset

A device Reset disables the voltage reference by clearing CVREN (CVRCON<7>). This Reset also disconnects the reference from the RA2 pin by clearing CVROE, and selects the high-voltage range by clearing CVRR. The CVR value select bits are also cleared.

FIGURE 23-2: COMPARATOR VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE



Note 1: R is dependent upon the comparator voltage reference configuration bits, CVRCON<5> and CVRCON<3:0>.

TABLE 23-1: REGISTERS ASSOCIATED WITH COMPARATOR VOLTAGE REFERENCE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
CVRCON ⁽²⁾	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	59
CM1CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	56
CM2CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	56
TRISA	TRISA7 ⁽¹⁾	TRISA6 ⁽¹⁾	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	58
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	58
ANCON0 ⁽²⁾	PCFG7	PCFG6	—	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	57
ANCON1 ⁽²⁾	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	57

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used with the comparator voltage reference.

Note 1: These bits are only available in select oscillator modes (FOSC2 Configuration bit = 0); otherwise, they are unimplemented.

2: Configuration SFR, overlaps with default SFR at this address; available only when WDTCON<4> = 1.

PIC18F87J11 FAMILY

NOTES:

24.0 SPECIAL FEATURES OF THE CPU

PIC18F87J11 Family devices include several features intended to maximize reliability and minimize cost through elimination of external components. These are:

- Oscillator Selection
- Resets:
 - Power-on Reset (POR)
 - Power-up Timer (PWRT)
 - Oscillator Start-up Timer (OST)
 - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor
- Two-Speed Start-up
- Code Protection
- In-Circuit Serial Programming

The oscillator can be configured for the application depending on frequency, power, accuracy and cost. All of the options are discussed in detail in **Section 2.0 “Oscillator Configurations”**.

A complete discussion of device Resets and interrupts is available in previous sections of this data sheet. In addition to their Power-up and Oscillator Start-up Timers provided for Resets, the PIC18F87J11 Family of devices have a configurable Watchdog Timer which is controlled in software.

The inclusion of an internal RC oscillator also provides the additional benefits of a Fail-Safe Clock Monitor (FSCM) and Two-Speed Start-up. FSCM provides for background monitoring of the peripheral clock and automatic switchover in the event of its failure. Two-Speed Start-up enables code to be executed almost immediately on start-up, while the primary clock source completes its start-up delays.

All of these features are enabled and configured by setting the appropriate Configuration register bits.

24.1 Configuration Bits

The Configuration bits can be programmed (read as ‘0’) or left unprogrammed (read as ‘1’) to select various device configurations. These bits are mapped starting at program memory location 300000h. A complete list is shown in Table 24-2. A detailed explanation of the various bit functions is provided in Register 24-1 through Register 24-6.

24.1.1 CONSIDERATIONS FOR CONFIGURING THE PIC18F87J11 FAMILY DEVICES

Unlike previous PIC18 microcontrollers, devices of the PIC18F87J11 Family do not use persistent memory registers to store configuration information. The configuration bytes are implemented as volatile memory which means that configuration data must be programmed each time the device is powered up.

Configuration data is stored in the four words at the top of the on-chip program memory space, known as the Flash Configuration Words. It is stored in program memory in the same order shown in Table 24-2, with CONFIG1L at the lowest address and CONFIG3H at the highest. The data is automatically loaded in the proper Configuration registers during device power-up.

When creating applications for these devices, users should always specifically allocate the location of the Flash Configuration Word for configuration data. This is to make certain that program code is not stored in this address when the code is compiled.

The volatile memory cells used for the Configuration bits always reset to ‘1’ on Power-on Resets. For all other type of Reset events, the previously programmed values are maintained and used without reloading from program memory.

The four Most Significant bits of CONFIG1H, CONFIG2H and CONFIG3H in program memory should also be ‘1111’. This makes these Configuration Words appear to be NOP instructions in the remote event that their locations are ever executed by accident. Since Configuration bits are not implemented in the corresponding locations, writing ‘1’s to these locations has no effect on device operation.

To prevent inadvertent configuration changes during code execution, all programmable Configuration bits are write-once. After a bit is initially programmed during a power cycle, it cannot be written to again. Changing a device configuration requires that power to the device be cycled.

PIC18F87J11 FAMILY

TABLE 24-1: MAPPING OF THE FLASH CONFIGURATION WORDS TO THE CONFIGURATION REGISTERS

Configuration Byte	Code Space Address	Configuration Register Address
CONFIG1L	XXXF8h	300000h
CONFIG1H	XXXF9h	300001h
CONFIG2L	XXXFAh	300002h
CONFIG2H	XXXF Bh	300003h
CONFIG3L	XXXFCh	300004h
CONFIG3H	XXXF Dh	300005h
CONFIG4L ⁽¹⁾	XXXF Eh	300006h
CONFIG4H ⁽¹⁾	XXXF Fh	300007h

Note 1: Unimplemented in PIC18F87J11 Family devices.

TABLE 24-2: CONFIGURATION BITS AND DEVICE IDs

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/Unprogrammed Value ⁽¹⁾
300000h	CONFIG1L	DEBUG	XINST	STVREN	—	—	—	WDTEN	111- ---1
300001h	CONFIG1H	__(2)	__(2)	__(2)	__(2)	—	CP0	—	— 1111 0111
300002h	CONFIG2L	IESO	FCMEN	—	—	—	FOSC2	FOSC1	FOSC0 11-- -111
300003h	CONFIG2H	__(2)	__(2)	__(2)	__(2)	WDTPS3	WDTPS2	WDTPS1	WDTPS0 1111 1111
300004h	CONFIG3L	WAIT ⁽³⁾	BW ⁽³⁾	EMB1 ⁽³⁾	EMB0 ⁽³⁾	EASHFT ⁽³⁾	—	—	— 1111 1---
300005h	CONFIG3H	__(2)	__(2)	__(2)	__(2)	MSSPMSK	PMPMX ⁽³⁾	ECCPMX ⁽³⁾	CCP2MX 1111 1111
3FFFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0 xxxx0 0000 ⁽⁴⁾
3FFFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3 0100 00xx ⁽⁴⁾

Legend: x = unknown, u = unchanged, - = unimplemented. Shaded cells are unimplemented, read as '0'.

Note 1: Values reflect the unprogrammed state as received from the factory and following Power-on Resets. In all other Reset states, the configuration bytes maintain their previously programmed states.

- 2:** The value of these bits in program memory should always be '1'. This ensures that the location is executed as a NOP if it is accidentally executed.
- 3:** Implemented in 80-pin devices only.
- 4:** See Register 24-7 and Register 24-8 for DEVID values. These registers are read-only and cannot be programmed by the user.

PIC18F87J11 FAMILY

REGISTER 24-1: CONFIG1L: CONFIGURATION REGISTER 1 LOW (BYTE ADDRESS 300000h)

R/WO-1	R/WO-1	R/WO-1	U-0	U-0	U-0	U-0	R/WO-1
DEBUG	XINST	STVREN	—	—	—	—	WDTEN
bit 7	bit 0						

Legend:

R = Readable bit

WO = Write-Once bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7	DEBUG: Background Debugger Enable bit 1 = Background debugger disabled; RB6 and RB7 configured as general purpose I/O pins 0 = Background debugger enabled; RB6 and RB7 are dedicated to In-Circuit Debug
bit 6	XINST: Extended Instruction Set Enable bit 1 = Instruction set extension and Indexed Addressing mode enabled 0 = Instruction set extension and Indexed Addressing mode disabled (Legacy mode)
bit 5	STVREN: Stack Overflow/Underflow Reset Enable bit 1 = Reset on stack overflow/underflow enabled 0 = Reset on stack overflow/underflow disabled
bit 4-1	Unimplemented: Read as '0'
bit 0	WDTEN: Watchdog Timer Enable bit 1 = WDT enabled 0 = WDT disabled (control is placed on SWDTEN bit)

REGISTER 24-2: CONFIG1H: CONFIGURATION REGISTER 1 HIGH (BYTE ADDRESS 300001h)

U-1	U-1	U-1	U-1	U-0	R/WO-1	U-1	U-1
—	—	—	—	—	CP0	—	—
bit 7	bit 0						

Legend:

R = Readable bit

WO = Write-Once bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-3	Unimplemented: Read as '0'
bit 2	CP0: Code Protection bit 1 = Program memory is not code-protected 0 = Program memory is code-protected
bit 1-0	Unimplemented: Read as '0'

PIC18F87J11 FAMILY

REGISTER 24-3: CONFIG2L: CONFIGURATION REGISTER 2 LOW (BYTE ADDRESS 300002h)

R/WO-1	R/WO-1	U-0	U-0	U-0	R/WO-1	R/WO-1	R/WO-1
IESO	FCMEN	—	—	—	FOSC2	FOSC1	FOSC0
bit 7							bit 0

Legend:

R = Readable bit

WO = Write-Once bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7 **IESO:** Two-Speed Start-up (Internal/External Oscillator Switchover) Control bit

1 = Two-Speed Start-up enabled

0 = Two-Speed Start-up disabled

bit 6 **FCMEN:** Fail-Safe Clock Monitor Enable bit

1 = Fail-Safe Clock Monitor enabled

0 = Fail-Safe Clock Monitor disabled

bit 5-3 **Unimplemented:** Read as '0'

bit 2-0 **FOSC2:FOSC0:** Oscillator Selection bits

111 = EC oscillator with PLL enabled; CLKO on RA6 (ECPLL)

110 = EC oscillator; CLKO on RA6 (EC)

101 = HS oscillator with PLL enabled (HSPLL)

100 = HS oscillator (HS)

011 = Internal oscillator with PLL enabled; CLKO on RA6, port function on RA7 (INTPLL1)

010 = Internal oscillator with PLL enabled; port function on RA6 and RA7 (INTPLL2)

001 = Internal oscillator block; CLKO on RA6, port function on RA7 (INTIO1)

000 = Internal oscillator block ; port function on RA6 and RA7 (INTIO2)

PIC18F87J11 FAMILY

REGISTER 24-4: CONFIG2H: CONFIGURATION REGISTER 2 HIGH (BYTE ADDRESS 300003h)

U-1	U-1	U-1	U-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1
—	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7							bit 0

Legend:

R = Readable bit

WO = Write-Once bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-4

Unimplemented: Read as '0'

bit 3-0

WDTPS3:WDTPS0: Watchdog Timer Postscale Select bits

1111 = 1:32,768

1110 = 1:16,384

1101 = 1:8,192

1100 = 1:4,096

1011 = 1:2,048

1010 = 1:1,024

1001 = 1:512

1000 = 1:256

0111 = 1:128

0110 = 1:64

0101 = 1:32

0100 = 1:16

0011 = 1:8

0010 = 1:4

0001 = 1:2

0000 = 1:1

PIC18F87J11 FAMILY

REGISTER 24-5: CONFIG3L: CONFIGURATION REGISTER 3 LOW (BYTE ADDRESS 300004h)

R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	U-0	U-0	U-0
WAIT ⁽¹⁾	BW ⁽¹⁾	EMB1 ⁽¹⁾	EMB0 ⁽¹⁾	EASHFT ⁽¹⁾	—	—	—
bit 7	bit 0						

Legend:

R = Readable bit

WO = Write-Once bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7 **WAIT:** External Bus Wait Enable bit⁽¹⁾

1 = Wait states on the external bus are disabled

0 = Wait states on the external bus are enabled and selected by MEMCON<5:4>

bit 6 **BW:** Data Bus Width Select bit⁽¹⁾

1 = 16-Bit Data Width modes

0 = 8-Bit Data Width modes

bit 5-4 **EMB1:EMB0:** External Memory Bus Configuration bits⁽¹⁾

11 = Microcontroller mode, external bus disabled

10 = Extended Microcontroller mode, 12-bit address width for external bus

01 = Extended Microcontroller mode, 16-bit address width for external bus

00 = Extended Microcontroller mode, 20-bit address width for external bus

bit 3 **EASHFT:** External Address Bus Shift Enable bit⁽¹⁾

1 = Address shifting enabled – external address bus is shifted to start at 000000h

0 = Address shifting disabled – external address bus reflects the PC value

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

Note 1: Implemented on 80-pin devices only.

REGISTER 24-6: CONFIG3H: CONFIGURATION REGISTER 3 HIGH (BYTE ADDRESS 300005h)

U-1	U-1	U-1	U-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1
—	—	—	—	MSSPMSK	PMPMX ⁽¹⁾	ECCPMX ⁽¹⁾	CCP2MX
bit 7	bit 0						

Legend:

R = Readable bit

WO = Write-Once bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-4	Unimplemented: Read as '0'
bit 3	MSSPMSK: MSSP Address Masking Mode Select bit 1 = 7-Bit Address Masking mode enabled 0 = 5-Bit Address Masking mode enable
bit 2	PMPMX: PMP Pin Multiplex bit ⁽¹⁾ 1 = PMP data and control multiplexed to same pins as external memory bus (PORTD and PORTE) 0 = PMP data and control multiplexed to alternate pin assignments (PORTA, PORTF and PORTH)
bit 1	ECCPMX: ECCPx MUX bit ⁽¹⁾ 1 = ECCP1 outputs (P1B/P1C) are multiplexed with RE6 and RE5; ECCP3 outputs (P3B/P3C) are multiplexed with RE4 and RE3 0 = ECCP1 outputs (P1B/P1C) are multiplexed with RH7 and RH6; ECCP3 outputs (P3B/P3C) are multiplexed with RH5 and RH4
bit 0	CCP2MX: CCP2 MUX bit 1 = CCP2/P2A is multiplexed with RC1 0 = CCP2/P2A is multiplexed with RE7 in Microcontroller mode (all devices) or with RB3 in Extended Microcontroller mode (80-pin devices only)

Note 1: Implemented on 80-pin devices only.

PIC18F87J11 FAMILY

REGISTER 24-7: DEVID1: DEVICE ID REGISTER 1 FOR PIC18F87J11 FAMILY DEVICES

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
bit 7	bit 0						

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

bit 7-5 **DEV2:DEV0:** Device ID bits

See Register 24-8 for a complete listing.

bit 4-0 **REV4:REV0:** Revision ID bits

These bits are used to indicate the device revision.

REGISTER 24-8: DEVID2: DEVICE ID REGISTER 2 FOR PIC18F87J11 FAMILY DEVICES

R	R	R	R	R	R	R	R
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3
bit 7	bit 0						

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

bit 7-0 **DEV10:DEV3:** Device ID bits:

DEV10:DEV3 (DEVID2<7:0>)	DEV2:DEV0 (DEVID1<7:5>)	Device
0100 0100	010	PIC18F66J11
0100 0100	011	PIC18F66J16
0100 0100	100	PIC18F67J11
0100 0100	111	PIC18F86J11
0100 0101	000	PIC18F86J16
0100 0101	001	PIC18F87J11

24.2 Watchdog Timer (WDT)

For PIC18F87J11 Family devices, the WDT is driven by the INTRC oscillator. When the WDT is enabled, the clock source is also enabled. The nominal WDT period is 4 ms and has the same stability as the INTRC oscillator.

The 4 ms period of the WDT is multiplied by a 16-bit postscaler. Any output of the WDT postscaler is selected by a multiplexor, controlled by the WDTPS bits in Configuration Register 2H. Available periods range from about 4 ms to 135 seconds (2.25 minutes depending on voltage, temperature and WDT postscaler). The WDT and postscaler are cleared whenever a SLEEP or CLRWDW instruction is executed, or a clock failure (primary or Timer1 oscillator) has occurred.

Note 1: The CLRWDW and SLEEP instructions clear the WDT and postscaler counts when executed.

2: When a CLRWDW instruction is executed, the postscaler count will be cleared.

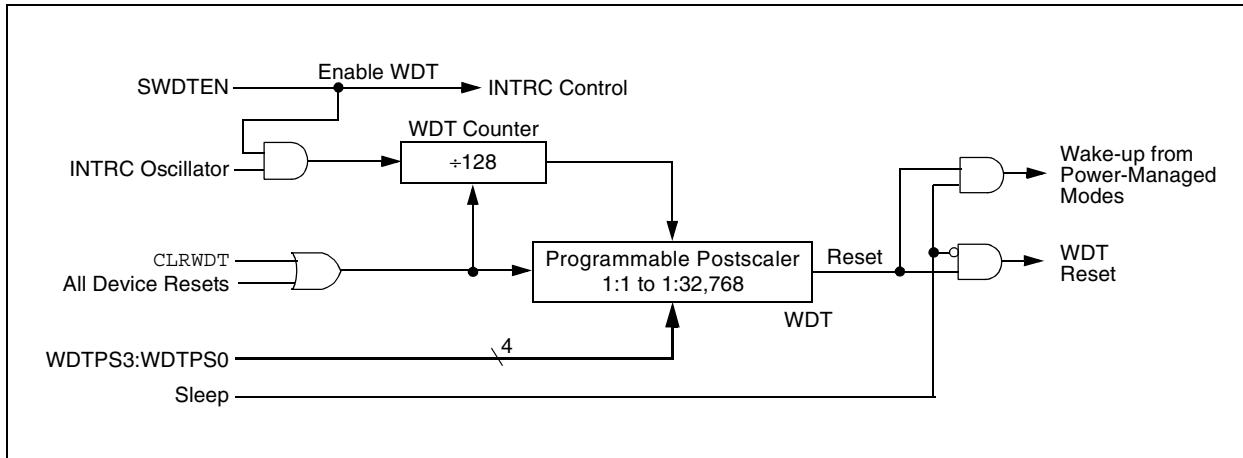
24.2.1 CONTROL REGISTER

The WDTCON register (Register 24-9) is a readable and writable register. The SWDTEN bit enables or disables WDT operation. This allows software to override the WDTEN Configuration bit and enable the WDT only if it has been disabled by the Configuration bit.

The ADSHR bit selects which SFRs are currently selected and accessible. See **Section 5.3.4.1 “Shared Address SFRs”** for additional details.

The LVDSTAT is a read-only status bit which is continuously updated and provides information about the current level of VDDCORE. This bit is only valid when the on-chip voltage regulator is enabled.

FIGURE 24-1: WDT BLOCK DIAGRAM



PIC18F87J11 FAMILY

REGISTER 24-9: WDTCON: WATCHDOG TIMER CONTROL REGISTER

R/W-0	R-x	U-0	R/W-0	U-0	U-0	U-0	U-0
REGSLP	LVDSTAT	—	ADSHR	—	—	—	SWDTEN ⁽¹⁾
bit 7	bit 0						

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

bit 7 **REGSLP:** Voltage Regulator Low-Power Operation Enable bit

1 = On-chip regulator enters low-power operation when device enters Sleep mode

0 = On-chip regulator is active, even in Sleep mode

bit 6 **LVDSTAT:** LVD Status bit

1 = VDDCORE > 2.45V

0 = VDDCORE < 2.45V

bit 5 **Unimplemented:** Read as '0'bit 4 **ADSHR:** Shared Address SFR Select bit

For details of bit operation, see Register 5-3.

bit 3-1 **Unimplemented:** Read as '0'bit 0 **SWDTEN:** Software Controlled Watchdog Timer Enable bit⁽¹⁾

1 = Watchdog Timer is on

0 = Watchdog Timer is off

Note 1: This bit has no effect if the Configuration bit, WDTEN, is enabled.**TABLE 24-3: SUMMARY OF WATCHDOG TIMER REGISTERS**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
RCON	IPEN	—	CM	RI	TO	PD	POR	BOR	56
WDTCON	REGSLP	LVDSTAT	—	ADSHR	—	—	—	SWDTEN	57

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the Watchdog Timer.

24.3 On-Chip Voltage Regulator

All of the PIC18F87J11 family devices power their core digital logic at a nominal 2.5V. For designs that are required to operate at a higher typical voltage, such as 3.3V, all devices in the PIC18F87J11 family incorporate an on-chip regulator that allows the device to run its core logic from VDD.

The regulator is controlled by the ENVREG pin. Tying VDD to the pin enables the regulator, which in turn, provides power to the core from the other VDD pins. When the regulator is enabled, a low-ESR filter capacitor must be connected to the VDDCORE/VCAP pin (Figure 24-2). This helps to maintain the stability of the regulator. The recommended value for the filter capacitor is provided in **Section 27.3 “DC Characteristics: PIC18F87J11 Family (Industrial)”**.

If ENVREG is tied to Vss, the regulator is disabled. In this case, separate power for the core logic at a nominal 2.5V must be supplied to the device on the VDDCORE/VCAP pin to run the I/O pins at higher voltage levels, typically 3.3V. Alternatively, the VDDCORE/VCAP and VDD pins can be tied together to operate at a lower nominal voltage. Refer to Figure 24-2 for possible configurations.

24.3.1 VOLTAGE REGULATOR TRACKING MODE AND LOW-VOLTAGE DETECTION

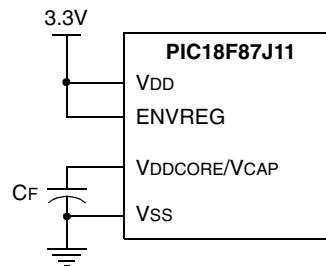
When it is enabled, the on-chip regulator provides a constant voltage of 2.5V nominal to the digital core logic. The regulator can provide this level from a VDD of about 2.5V, all the way up to the device's VDDMAX. It does not have the capability to boost VDD levels below 2.5V. In order to prevent “brown-out” conditions, when the voltage drops too low for the regulator, the regulator enters Tracking mode. In Tracking mode, the regulator output follows VDD, with a typical voltage drop of 100 mV.

The on-chip regulator includes a simple, Low-Voltage Detect (LVD) circuit. If VDD drops too low to maintain approximately 2.45V on VDDCORE, the circuit sets the Low-Voltage Detect Interrupt Flag, LVDIF (PIR2<2>). This can be used to generate an interrupt and put the application into a low-power operational mode, or trigger an orderly shutdown. Low-Voltage Detection is only available when the regulator is enabled.

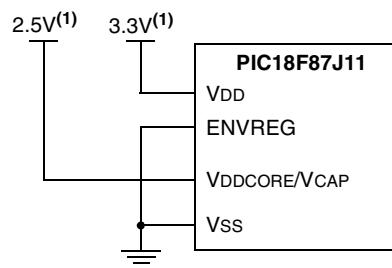
The Low-Voltage Detect interrupt is edge-sensitive. The interrupt flag will only be set once per falling edge of VDDCORE. Firmware can clear the interrupt flag, but a new interrupt will not be generated until VDDCORE rises back above, and then falls below, the 2.45 threshold. Upon device Resets, the interrupt flag will reset to ‘0’, even if VDDCORE is less than 2.45V. When the regulator is enabled, the LVDSTAT bit in the WDTCON register can be polled to determine the current level of VDDCORE.

FIGURE 24-2: CONNECTIONS FOR THE ON-CHIP REGULATOR

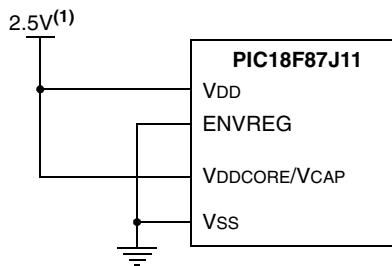
Regulator Enabled (ENVREG tied to VDD):



Regulator Disabled (ENVREG tied to ground):



Regulator Disabled (VDD tied to VDDCORE):



Note 1: These are typical operating voltages. Refer to **Section 27.1 “DC Characteristics: Supply Voltage”** for the full operating ranges of VDD and VDDCORE.

PIC18F87J11 FAMILY

24.3.2 ON-CHIP REGULATOR AND BOR

When the on-chip regulator is enabled, PIC18F87J11 family devices also have a simple brown-out capability. If the voltage supplied to the regulator is inadequate to maintain a regulated level, the regulator Reset circuitry will generate a Brown-out Reset. This event is captured by the BOR flag bit (RCON<0>).

The operation of the Brown-out Reset is described in more detail in **Section 4.4 “Brown-out Reset (BOR)”** and **Section 4.4.1 “Detecting BOR”**. The brown-out voltage levels are specific in **Section 27.1 “DC Characteristics: Supply Voltage PIC18F87J11 Family (Industrial)”**.

24.3.3 POWER-UP REQUIREMENTS

The on-chip regulator is designed to meet the power-up requirements for the device. If the application does not use the regulator, then strict power-up conditions must be adhered to. While powering up, VDDCORE must never exceed VDD by 0.3 volts.

24.3.4 OPERATION IN SLEEP MODE

When enabled, the on-chip regulator always consumes a small incremental amount of current over IDD. This includes when the device is in Sleep mode, even though the core digital logic does not require power. To provide additional savings in applications where power resources are critical, the regulator can be configured to automatically disable itself whenever the device goes into Sleep mode. This feature is controlled by the REGSLP bit (WDTCON<7>, Register 24-9). Setting this bit disables the regulator in Sleep mode and reduces its current consumption to a minimum.

Substantial Sleep mode power savings can be obtained by setting the REGSLP bit, but device wake-up time will increase in order to insure the regulator has enough time to stabilize. The REGSLP bit is automatically cleared by hardware when a Low-Voltage Detect condition occurs.

24.4 Two-Speed Start-up

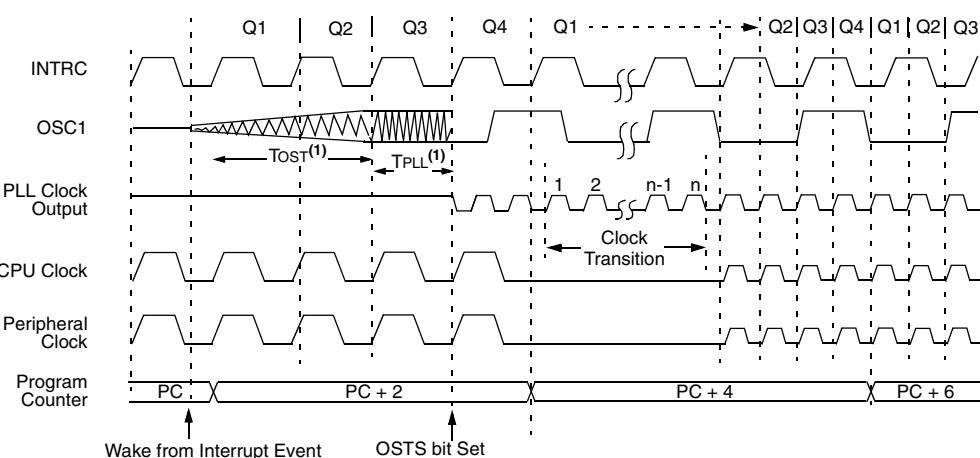
The Two-Speed Start-up feature helps to minimize the latency period, from oscillator start-up to code execution, by allowing the microcontroller to use the INT RC oscillator as a clock source until the primary clock source is available. It is enabled by setting the IESO Configuration bit.

Two-Speed Start-up should be enabled only if the primary oscillator mode is HS or HSPLL (Crystal-Based) modes. Since the EC and ECPLL modes do not require an Oscillator Start-up Timer delay, Two-Speed Start-up should be disabled.

When enabled, Resets and wake-ups from Sleep mode cause the device to configure itself to run from the internal oscillator block as the clock source, following the time-out of the Power-up Timer after a Power-on Reset is enabled. This allows almost immediate code execution while the primary oscillator starts and the OST is running. Once the OST times out, the device automatically switches to PRI_RUN mode.

In all other power-managed modes, Two-Speed Start-up is not used. The device will be clocked by the currently selected clock source until the primary clock source becomes available. The setting of the IESO bit is ignored.

FIGURE 24-3: TIMING TRANSITION FOR TWO-SPEED START-UP (INTRC TO HSPLL)



24.4.1 SPECIAL CONSIDERATIONS FOR USING TWO-SPEED START-UP

While using the INT RC oscillator in Two-Speed Start-up, the device still obeys the normal command sequences for entering power-managed modes, including serial SLEEP instructions (refer to **Section 3.1.4 “Multiple Sleep Commands”**). In practice, this means that user code can change the SCS1:SCS0 bit settings or issue SLEEP instructions before the OST times out. This would allow an application to briefly wake-up, perform routine “housekeeping” tasks and return to Sleep before the device starts to operate from the primary oscillator.

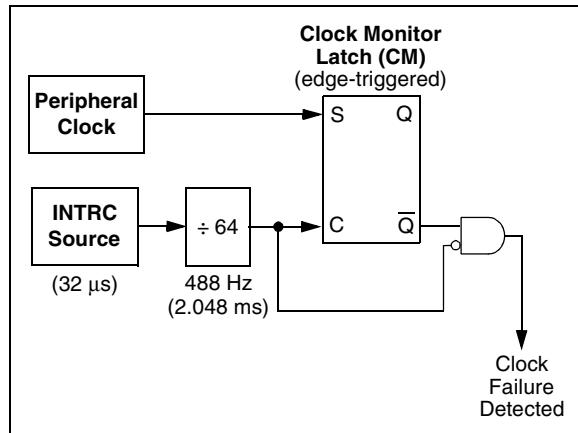
User code can also check if the primary clock source is currently providing the device clocking by checking the status of the OSTS bit (OSCCON<3>). If the bit is set, the primary oscillator is providing the clock. Otherwise, the internal oscillator block is providing the clock during wake-up from Reset or Sleep mode.

24.5 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the microcontroller to continue operation in the event of an external oscillator failure by automatically switching the device clock to the internal oscillator block. The FSCM function is enabled by setting the FCMEN Configuration bit.

When FSCM is enabled, the INT RC oscillator runs at all times to monitor clocks to peripherals and provide a backup clock in the event of a clock failure. Clock monitoring (shown in Figure 24-4) is accomplished by creating a sample clock signal which is the INT RC output divided by 64. This allows ample time between FSCM sample clocks for a peripheral clock edge to occur. The peripheral device clock and the sample clock are presented as inputs to the Clock Monitor (CM) latch. The CM is set on the falling edge of the device clock source but cleared on the rising edge of the sample clock.

FIGURE 24-4: FSCM BLOCK DIAGRAM



Clock failure is tested for on the falling edge of the sample clock. If a sample clock falling edge occurs while CM is still set, a clock failure has been detected (Figure 24-5). This causes the following:

- the FSCM generates an oscillator fail interrupt by setting bit OSCFIF (PIR2<7>);
- the device clock source is switched to the internal oscillator block (OSCCON is not updated to show the current clock source – this is the fail-safe condition); and
- the WDT is reset.

During switchover, the postscaler frequency from the internal oscillator block may not be sufficiently stable for timing sensitive applications. In these cases, it may be desirable to select another clock configuration and enter an alternate power-managed mode. This can be done to attempt a partial recovery or execute a controlled shutdown. See **Section 3.1.4 “Multiple Sleep Commands”** and **Section 24.4.1 “Special Considerations for Using Two-Speed Start-up”** for more details.

The FSCM will detect failures of the primary or secondary clock sources only. If the internal oscillator block fails, no failure would be detected, nor would any action be possible.

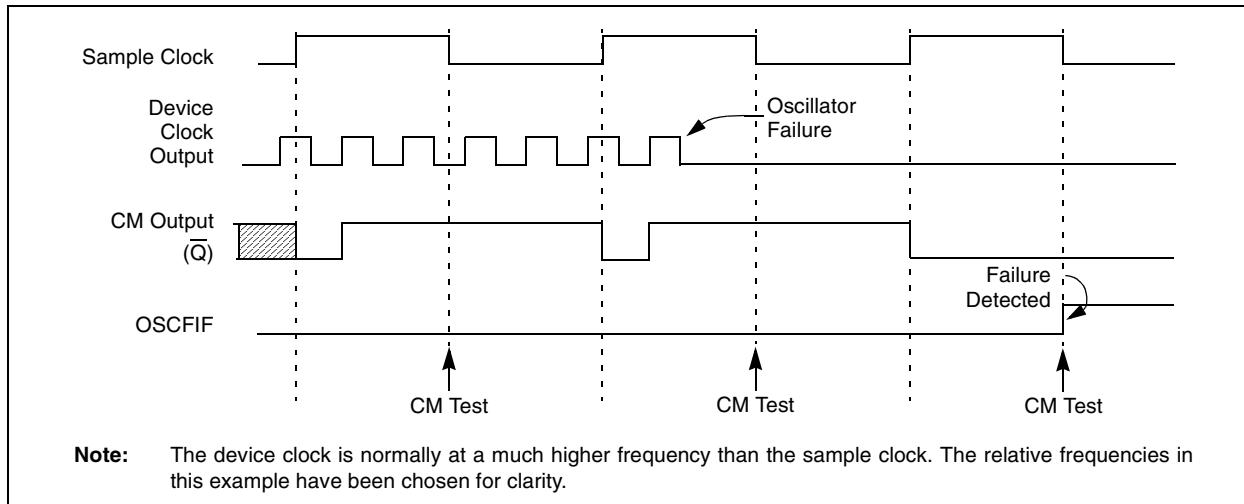
24.5.1 FSCM AND THE WATCHDOG TIMER

Both the FSCM and the WDT are clocked by the INT RC oscillator. Since the WDT operates with a separate divider and counter, disabling the WDT has no effect on the operation of the INT RC oscillator when the FSCM is enabled.

As already noted, the clock source is switched to the INT RC clock when a clock failure is detected; this may mean a substantial change in the speed of code execution. If the WDT is enabled with a small prescale value, a decrease in clock speed allows a WDT time-out to occur and a subsequent device Reset. For this reason, fail-safe clock events also reset the WDT and postscaler, allowing it to start timing from when execution speed was changed and decreasing the likelihood of an erroneous time-out.

PIC18F87J11 FAMILY

FIGURE 24-5: FSCM TIMING DIAGRAM



24.5.2 EXITING FAIL-SAFE OPERATION

The fail-safe condition is terminated by either a device Reset or by entering a power-managed mode. On Reset, the controller starts the primary clock source specified in Configuration Register 2H (with any required start-up delays that are required for the oscillator mode, such as OST or PLL timer). The INT RC oscillator provides the device clock until the primary clock source becomes ready (similar to a Two-Speed Start-up). The clock source is then switched to the primary clock (indicated by the OSTS bit in the OSCCON register becoming set). The Fail-Safe Clock Monitor then resumes monitoring the peripheral clock.

The primary clock source may never become ready during start-up. In this case, operation is clocked by the INT RC oscillator. The OSCCON register will remain in its Reset state until a power-managed mode is entered.

24.5.3 FSCM INTERRUPTS IN POWER-MANAGED MODES

By entering a power-managed mode, the clock multiplexor selects the clock source selected by the OSCCON register. Fail-Safe Clock Monitoring of the power-managed clock source resumes in the power-managed mode.

If an oscillator failure occurs during power-managed operation, the subsequent events depend on whether or not the oscillator failure interrupt is enabled. If enabled ($\text{OSCFIF} = 1$), code execution will be clocked by the INT RC multiplexor. An automatic transition back to the failed clock source will not occur.

If the interrupt is disabled, subsequent interrupts while in Idle mode will cause the CPU to begin executing instructions while being clocked by the INT RC source.

24.5.4 POR OR WAKE-UP FROM SLEEP

The FSCM is designed to detect oscillator failure at any point after the device has exited Power-on Reset (POR) or low-power Sleep mode. When the primary device clock is either the EC or INT RC modes, monitoring can begin immediately following these events.

For HS or HSPLL modes, the situation is somewhat different. Since the oscillator may require a start-up time considerably longer than the FSCM sample clock time, a false clock failure may be detected. To prevent this, the internal oscillator block is automatically configured as the device clock and functions until the primary clock is stable (the OST and PLL timers have timed out). This is identical to Two-Speed Start-up mode. Once the primary clock is stable, the INT RC returns to its role as the FSCM source.

Note: The same logic that prevents false oscillator failure interrupts on POR, or wake from Sleep, will also prevent the detection of the oscillator's failure to start at all following these events. This can be avoided by monitoring the OSTS bit and using a timing routine to determine if the oscillator is taking too long to start. Even so, no oscillator failure interrupt will be flagged.

As noted in **Section 24.4.1 “Special Considerations for Using Two-Speed Start-up”**, it is also possible to select another clock configuration and enter an alternate power-managed mode while waiting for the primary clock to become stable. When the new power-managed mode is selected, the primary clock is disabled.

24.6 Program Verification and Code Protection

For all devices in the PIC18F87J11 Family of devices, the on-chip program memory space is treated as a single block. Code protection for this block is controlled by one Configuration bit, CP0. This bit inhibits external reads and writes to the program memory space. It has no direct effect in normal execution mode.

24.6.1 CONFIGURATION REGISTER PROTECTION

The Configuration registers are protected against untoward changes or reads in two ways. The primary protection is the write-once feature of the Configuration bits which prevents reconfiguration once the bit has been programmed during a power cycle. To safeguard against unpredictable events, Configuration bit changes resulting from individual cell level disruptions (such as ESD events) will cause a parity error and trigger a device Reset. This is seen by the user as a Configuration Match Reset.

The data for the Configuration registers is derived from the Flash Configuration Words in program memory. When the CP0 bit set, the source data for device configuration is also protected as a consequence.

24.7 In-Circuit Serial Programming

PIC18F87J11 Family 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.

24.8 In-Circuit Debugger

When the DEBUG Configuration bit is programmed to a '0', the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB® IDE. When the microcontroller has this feature enabled, some resources are not available for general use. Table 24-4 shows which resources are required by the background debugger.

TABLE 24-4: DEBUGGER RESOURCES

I/O pins:	RB6, RB7
Stack:	2 levels
Program Memory:	512 bytes
Data Memory:	10 bytes

PIC18F87J11 FAMILY

NOTES:

25.0 INSTRUCTION SET SUMMARY

The PIC18F87J11 Family of devices incorporate the standard set of 75 PIC18 core instructions, as well as an extended set of 8 new instructions for the optimization of code that is recursive or that utilizes a software stack. The extended set is discussed later in this section.

25.1 Standard Instruction Set

The standard PIC18 instruction set adds many enhancements to the previous PIC® instruction sets, while maintaining an easy migration from these instruction sets. Most instructions are a single program memory word (16 bits), but there are four instructions that require two program memory locations.

Each single-word instruction is a 16-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 instruction set is highly orthogonal and is grouped into four basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal** operations
- **Control** operations

The PIC18 instruction set summary in Table 25-2 lists **byte-oriented**, **bit-oriented**, **literal** and **control** operations. Table 25-1 shows the opcode field descriptions.

Most **byte-oriented** instructions have three operands:

1. The file register (specified by 'f')
2. The destination of the result (specified by 'd')
3. The accessed memory (specified by 'a')

The file register designator, 'f', specifies which file register is to be used by the instruction. The destination designator, 'd', specifies where the result of the operation is to be placed. If 'd' is '0', the result is placed in the WREG register. If 'd' is '1', the result is placed in the file register specified in the instruction.

All **bit-oriented** instructions have three operands:

1. The file register (specified by 'f')
2. The bit in the file register (specified by 'b')
3. The accessed memory (specified by 'a')

The bit field designator 'b' selects the number of the bit affected by the operation, while the file register designator, 'f', represents the number of the file in which the bit is located.

The **literal** instructions may use some of the following operands:

- A literal value to be loaded into a file register (specified by 'K')
- The desired FSR register to load the literal value into (specified by 'f')
- No operand required (specified by '—')

The **control** instructions may use some of the following operands:

- A program memory address (specified by 'n')
- The mode of the CALL or RETURN instructions (specified by 's')
- The mode of the table read and table write instructions (specified by 'm')
- No operand required (specified by '—')

All instructions are a single word, except for four double-word instructions. These instructions were made double-word to contain the required information in 32 bits. In the second word, the 4 MSbs are '1's. If this second word is executed as an instruction (by itself), it will execute as a NOP.

All single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles with the additional instruction cycle(s) executed as a NOP.

The double-word instructions execute in two instruction cycles.

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μ s. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 μ s. Two-word branch instructions (if true) would take 3 μ s.

Figure 25-1 shows the general formats that the instructions can have. All examples use the convention 'nnh' to represent a hexadecimal number.

The instruction set summary, shown in Table 25-2, lists the standard instructions recognized by the Microchip MPASM™ Assembler.

Section 25.1.1 “Standard Instruction Set” provides a description of each instruction.

PIC18F87J11 FAMILY

TABLE 25-1: OPCODE FIELD DESCRIPTIONS

Field	Description
a	RAM access bit: a = 0: RAM location in Access RAM (BSR register is ignored) a = 1: RAM bank is specified by BSR register
bbb	Bit address within an 8-bit file register (0 to 7).
BSR	Bank Select Register. Used to select the current RAM bank.
C, DC, Z, OV, N	ALU Status bits: C arry, D igit C arry, Z ero, O verflow, N egative.
d	Destination select bit: d = 0: store result in WREG d = 1: store result in file register f
dest	Destination: either the WREG register or the specified register file location.
f	8-bit register file address (00h to FFh), or 2-bit FSR designator (0h to 3h).
f _s	12-bit register file address (000h to FFFh). This is the source address.
f _d	12-bit register file address (000h to FFFh). This is the destination address.
GIE	Global Interrupt Enable bit.
k	Literal field, constant data or label (may be either an 8-bit, 12-bit or a 20-bit value).
label	Label name.
mm	The mode of the TBLPTR register for the table read and table write instructions. Only used with table read and table write instructions: * No Change to register (such as TBLPTR with table reads and writes) *+ Post-Increment register (such as TBLPTR with table reads and writes) *- Post-Decrement register (such as TBLPTR with table reads and writes) +* Pre-Increment register (such as TBLPTR with table reads and writes)
n	The relative address (2's complement number) for relative branch instructions or the direct address for Call/Branch and Return instructions.
PC	Program Counter.
PCL	Program Counter Low Byte.
PCH	Program Counter High Byte.
PCLATH	Program Counter High Byte Latch.
PCLATU	Program Counter Upper Byte Latch.
PD	Power-Down bit.
PRODH	Product of Multiply High Byte.
PRODL	Product of Multiply Low Byte.
s	Fast Call/Return mode select bit: s = 0: do not update into/from shadow registers s = 1: certain registers loaded into/from shadow registers (Fast mode)
TBLPTR	21-bit Table Pointer (points to a program memory location).
TABLAT	8-bit Table Latch.
TO	Time-out bit.
TOS	Top-of-Stack.
u	Unused or Unchanged.
WDT	Watchdog Timer.
WREG	Working register (accumulator).
x	Don't care ('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.
z _s	7-bit offset value for Indirect Addressing of register files (source).
z _d	7-bit offset value for Indirect Addressing of register files (destination).
{ }	Optional argument.
[text]	Indicates Indexed Addressing.
(text)	The contents of text.
[expr] <n>	Specifies bit n of the register indicated by the pointer, expr.
→	Assigned to.
< >	Register bit field.
ε	In the set of.
italics	User-defined term (font is Courier New).

FIGURE 25-1: GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations	Example Instruction																	
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 15px;">15</td><td style="width: 10px;">10</td><td style="width: 9px;">9</td><td style="width: 8px;">8</td><td style="width: 7px;">7</td><td style="width: 0px;">0</td></tr> <tr> <td>OPCODE</td><td>d</td><td>a</td><td colspan="3">f (FILE #)</td></tr> </table> <p> d = 0 for result destination to be WREG register d = 1 for result destination to be file register (f) a = 0 to force Access Bank a = 1 for BSR to select bank f = 8-bit file register address </p>	15	10	9	8	7	0	OPCODE	d	a	f (FILE #)			ADDWF MYREG, W, B					
15	10	9	8	7	0													
OPCODE	d	a	f (FILE #)															
Byte to Byte move operations (2-word)																		
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 15px;">15</td><td style="width: 12px;">12</td><td style="width: 11px;">11</td><td style="width: 0px;">0</td></tr> <tr> <td>OPCODE</td><td colspan="3">f (Source FILE #)</td></tr> <tr> <td style="width: 15px;">15</td><td style="width: 12px;">12</td><td style="width: 11px;">11</td><td style="width: 0px;">0</td></tr> <tr> <td>1111</td><td colspan="3">f (Destination FILE #)</td></tr> </table> <p>f = 12-bit file register address</p>	15	12	11	0	OPCODE	f (Source FILE #)			15	12	11	0	1111	f (Destination FILE #)			MOVFF MYREG1, MYREG2	
15	12	11	0															
OPCODE	f (Source FILE #)																	
15	12	11	0															
1111	f (Destination FILE #)																	
Bit-oriented file register operations																		
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 15px;">15</td><td style="width: 12px;">12</td><td style="width: 11px;">11</td><td style="width: 9px;">9</td><td style="width: 8px;">8</td><td style="width: 7px;">7</td><td style="width: 0px;">0</td></tr> <tr> <td>OPCODE</td><td>b (BIT #)</td><td>a</td><td colspan="3">f (FILE #)</td></tr> </table> <p> b = 3-bit position of bit in file register (f) a = 0 to force Access Bank a = 1 for BSR to select bank f = 8-bit file register address </p>	15	12	11	9	8	7	0	OPCODE	b (BIT #)	a	f (FILE #)			BSF MYREG, bit, B				
15	12	11	9	8	7	0												
OPCODE	b (BIT #)	a	f (FILE #)															
Literal operations																		
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 15px;">15</td><td style="width: 8px;">8</td><td style="width: 7px;">7</td><td style="width: 0px;">0</td></tr> <tr> <td>OPCODE</td><td colspan="3">k (literal)</td></tr> </table> <p>k = 8-bit immediate value</p>	15	8	7	0	OPCODE	k (literal)			MOVLW 7Fh									
15	8	7	0															
OPCODE	k (literal)																	
Control operations																		
CALL, GOTO and Branch operations																		
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 15px;">15</td><td style="width: 8px;">8</td><td style="width: 7px;">7</td><td style="width: 0px;">0</td></tr> <tr> <td>OPCODE</td><td colspan="3">n<7:0> (literal)</td></tr> <tr> <td style="width: 15px;">15</td><td style="width: 12px;">12</td><td style="width: 11px;">11</td><td style="width: 0px;">0</td></tr> <tr> <td>1111</td><td colspan="3">n<19:8> (literal)</td></tr> </table> <p>n = 20-bit immediate value</p>	15	8	7	0	OPCODE	n<7:0> (literal)			15	12	11	0	1111	n<19:8> (literal)			GOTO Label	
15	8	7	0															
OPCODE	n<7:0> (literal)																	
15	12	11	0															
1111	n<19:8> (literal)																	
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 15px;">15</td><td style="width: 8px;">8</td><td style="width: 7px;">7</td><td style="width: 0px;">0</td></tr> <tr> <td>OPCODE</td><td>S</td><td colspan="3">n<7:0> (literal)</td></tr> <tr> <td style="width: 15px;">15</td><td style="width: 12px;">12</td><td style="width: 11px;">11</td><td style="width: 0px;">0</td></tr> <tr> <td>1111</td><td colspan="3">n<19:8> (literal)</td></tr> </table> <p>S = Fast bit</p>	15	8	7	0	OPCODE	S	n<7:0> (literal)			15	12	11	0	1111	n<19:8> (literal)			CALL MYFUNC
15	8	7	0															
OPCODE	S	n<7:0> (literal)																
15	12	11	0															
1111	n<19:8> (literal)																	
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 15px;">15</td><td style="width: 11px;">11</td><td style="width: 10px;">10</td><td style="width: 0px;">0</td></tr> <tr> <td>OPCODE</td><td colspan="3">n<10:0> (literal)</td></tr> </table>	15	11	10	0	OPCODE	n<10:0> (literal)			BRA MYFUNC									
15	11	10	0															
OPCODE	n<10:0> (literal)																	
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 15px;">15</td><td style="width: 8px;">8</td><td style="width: 7px;">7</td><td style="width: 0px;">0</td></tr> <tr> <td>OPCODE</td><td colspan="3">n<7:0> (literal)</td></tr> </table>	15	8	7	0	OPCODE	n<7:0> (literal)			BC MYFUNC									
15	8	7	0															
OPCODE	n<7:0> (literal)																	

PIC18F87J11 FAMILY

TABLE 25-2: PIC18F87J11 FAMILY INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word				Status Affected	Notes	
			MSb	LSb					
BYTE-ORIENTED OPERATIONS									
ADDWF f, d, a	Add WREG and f	1	0010 01da	ffff	ffff	C, DC, Z, OV, N		1, 2	
ADDWFC f, d, a	Add WREG and Carry bit to f	1	0010 00da	ffff	ffff	C, DC, Z, OV, N		1, 2	
ANDWF f, d, a	AND WREG with f	1	0001 01da	ffff	ffff	Z, N		1,2	
CLRF f, a	Clear f	1	0110 101a	ffff	ffff	Z		2	
COMF f, d, a	Complement f	1	0001 11da	ffff	ffff	Z, N		1, 2	
CPFSEQ f, a	Compare f with WREG, Skip =	1 (2 or 3)	0110 001a	ffff	ffff	None		4	
CPFSGT f, a	Compare f with WREG, Skip >	1 (2 or 3)	0110 010a	ffff	ffff	None		4	
CPFSLT f, a	Compare f with WREG, Skip <	1 (2 or 3)	0110 000a	ffff	ffff	None		1, 2	
DECF f, d, a	Decrement f	1	0000 01da	ffff	ffff	C, DC, Z, OV, N		1, 2, 3, 4	
DECFSZ f, d, a	Decrement f, Skip if 0	1 (2 or 3)	0010 11da	ffff	ffff	None		1, 2, 3, 4	
DCFSNZ f, d, a	Decrement f, Skip if Not 0	1 (2 or 3)	0100 11da	ffff	ffff	None		1, 2	
INCF f, d, a	Increment f	1	0010 10da	ffff	ffff	C, DC, Z, OV, N		1, 2, 3, 4	
INCFSZ f, d, a	Increment f, Skip if 0	1 (2 or 3)	0011 11da	ffff	ffff	None		4	
INFSNZ f, d, a	Increment f, Skip if Not 0	1 (2 or 3)	0100 10da	ffff	ffff	None		1, 2	
IOWWF f, d, a	Inclusive OR WREG with f	1	0001 00da	ffff	ffff	Z, N		1, 2	
MOVF f, d, a	Move f	1	0101 00da	ffff	ffff	Z, N		1	
MOVFF f _s , f _d	Move f _s (source) to 1st word f _d (destination) 2nd word	2	1100 ffff	ffff	ffff	None			
			1111 ffff	ffff	ffff				
MOVWF f, a	Move WREG to f	1	0110 111a	ffff	ffff	None			
MULWF f, a	Multiply WREG with f	1	0000 001a	ffff	ffff	None		1, 2	
NEGF f, a	Negate f	1	0110 110a	ffff	ffff	C, DC, Z, OV, N			
RLCF f, d, a	Rotate Left f through Carry	1	0011 01da	ffff	ffff	C, Z, N		1, 2	
RLNCF f, d, a	Rotate Left f (No Carry)	1	0100 01da	ffff	ffff	Z, N			
RRCF f, d, a	Rotate Right f through Carry	1	0011 00da	ffff	ffff	C, Z, N			
RRNCF f, d, a	Rotate Right f (No Carry)	1	0100 00da	ffff	ffff	Z, N			
SETF f, a	Set f	1	0110 100a	ffff	ffff	None		1, 2	
SUBFWB f, d, a	Subtract f from WREG with Borrow	1	0101 01da	ffff	ffff	C, DC, Z, OV, N			
SUBWF f, d, a	Subtract WREG from f	1	0101 11da	ffff	ffff	C, DC, Z, OV, N		1, 2	
SUBWFB f, d, a	Subtract WREG from f with Borrow	1	0101 10da	ffff	ffff	C, DC, Z, OV, N			
SWAPF f, d, a	Swap Nibbles in f	1	0011 10da	ffff	ffff	None		4	
TSTFSZ f, a	Test f, Skip if 0	1 (2 or 3)	0110 011a	ffff	ffff	None		1, 2	
XORWF f, d, a	Exclusive OR WREG with f	1	0001 10da	ffff	ffff	Z, N			

- Note 1:** When a PORT register is modified as a function of itself (e.g., MOVF PORTB, 1, 0), 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.
- 3:** If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.
- 4:** Some instructions are two-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

PIC18F87J11 FAMILY

TABLE 25-2: PIC18F87J11 FAMILY INSTRUCTION SET (CONTINUED)

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word				Status Affected	Notes
			MSb	Lsb				
BIT-ORIENTED OPERATIONS								
BCF f, b, a	Bit Clear f	1	1001 bbba	ffff ffff			None	1, 2
BSF f, b, a	Bit Set f	1	1000 bbba	ffff ffff			None	1, 2
BTFS C f, b, a	Bit Test f, Skip if Clear	1 (2 or 3)	1011 bbba	ffff ffff			None	3, 4
BTFS S f, b, a	Bit Test f, Skip if Set	1 (2 or 3)	1010 bbba	ffff ffff			None	3, 4
BTG f, b, a	Bit Toggle f	1	0111 bbba	ffff ffff			None	1, 2
CONTROL OPERATIONS								
BC n	Branch if Carry	1 (2)	1110 0010	nnnn nnnn			None	
BN n	Branch if Negative	1 (2)	1110 0110	nnnn nnnn			None	
BNC n	Branch if Not Carry	1 (2)	1110 0011	nnnn nnnn			None	
BNN n	Branch if Not Negative	1 (2)	1110 0111	nnnn nnnn			None	
BNOV n	Branch if Not Overflow	1 (2)	1110 0101	nnnn nnnn			None	
BNZ n	Branch if Not Zero	1 (2)	1110 0001	nnnn nnnn			None	
BOV n	Branch if Overflow	1 (2)	1110 0100	nnnn nnnn			None	
BRA n	Branch Unconditionally	2	1101 0nnn	nnnn nnnn			None	
BZ n	Branch if Zero	1 (2)	1110 0000	nnnn nnnn			None	
CALL n, s	Call Subroutine 1st word 2nd word	2	1110 110s	kkkk kkkk			None	
			1111 kkkk	kkkk kkkk				
CLRWDT —	Clear Watchdog Timer	1	0000 0000	0000 0100			TO, PD	
DAW —	Decimal Adjust WREG	1	0000 0000	0000 0111			C	
GOTO n	Go to Address 1st word 2nd word	2	1110 1111	kkkk kkkk			None	
			1111 kkkk	kkkk kkkk				
NOP —	No Operation	1	0000 0000	0000 0000			None	
NOP —	No Operation	1	1111 xxxx	xxxx xxxx			None	4
POP —	Pop Top of Return Stack (TOS)	1	0000 0000	0000 0110			None	
PUSH —	Push Top of Return Stack (TOS)	1	0000 0000	0000 0101			None	
RCALL n	Relative Call	2	1101 1nnn	nnnn nnnn			None	
RESET	Software Device Reset	1	0000 0000	1111 1111			All	
RETFIE s	Return from Interrupt Enable	2	0000 0000	0001 000s			GIE/GIEH, PEIE/GIEL	
RETLW k	Return with Literal in WREG	2	0000 1100	kkkk kkkk			None	
RETURN s	Return from Subroutine	2	0000 0000	0001 001s			None	
SLEEP —	Go into Standby mode	1	0000 0000	0000 0011			TO, PD	

Note 1: When a PORT register is modified as a function of itself (e.g., MOVF PORTB, 1, 0), 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.
- 3:** If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.
- 4:** Some instructions are two-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

PIC18F87J11 FAMILY

TABLE 25-2: PIC18F87J11 FAMILY INSTRUCTION SET (CONTINUED)

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word				Status Affected	Notes	
			MSb		LSb				
LITERAL OPERATIONS									
ADDLW k	Add Literal and WREG	1	0000	1111	kkkk	kkkk	C, DC, Z, OV, N		
ANDLW k	AND Literal with WREG	1	0000	1011	kkkk	kkkk	Z, N		
IORLW k	Inclusive OR Literal with WREG	1	0000	1001	kkkk	kkkk	Z, N		
LFSR f, k	Move Literal (12-bit) 2nd word to FSR (f) 1st word	2	1110	1110	0fff	kkkk	None		
			1111	0000	kkkk	kkkk			
MOVLB k	Move Literal to BSR<3:0>	1	0000	0001	0000	kkkk	None		
MOVLW k	Move Literal to WREG	1	0000	1110	kkkk	kkkk	None		
MULLW k	Multiply Literal with WREG	1	0000	1101	kkkk	kkkk	None		
RETLW k	Return with Literal in WREG	2	0000	1100	kkkk	kkkk	None		
SUBLW k	Subtract WREG from Literal	1	0000	1000	kkkk	kkkk	C, DC, Z, OV, N		
XORLW k	Exclusive OR Literal with WREG	1	0000	1010	kkkk	kkkk	Z, N		
DATA MEMORY ↔ PROGRAM MEMORY OPERATIONS									
TBLRD*	Table Read	2	0000	0000	0000	1000	None		
TBLRD*+	Table Read with Post-Increment		0000	0000	0000	1001	None		
TBLRD*-	Table Read with Post-Decrement		0000	0000	0000	1010	None		
TBLRD+*	Table Read with Pre-Increment		0000	0000	0000	1011	None		
TBLWT*	Table Write	2	0000	0000	0000	1100	None		
TBLWT*+	Table Write with Post-Increment		0000	0000	0000	1101	None		
TBLWT*-	Table Write with Post-Decrement		0000	0000	0000	1110	None		
TBLWT+*	Table Write with Pre-Increment		0000	0000	0000	1111	None		

- Note 1:** When a PORT register is modified as a function of itself (e.g., MOVF PORTB, 1, 0), 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.
- 3:** If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.
- 4:** Some instructions are two-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

25.1.1 STANDARD INSTRUCTION SET

ADDLW	ADD Literal to W											
Syntax:	ADDLW k											
Operands:	0 ≤ k ≤ 255											
Operation:	(W) + k → W											
Status Affected:	N, OV, C, DC, Z											
Encoding:	0000	1111	kkkk	kkkk								
Description:	The contents of W are added to the 8-bit literal 'k' and the result is placed in W.											
Words:	1											
Cycles:	1											
Q Cycle Activity:	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>Decode</td><td>Read literal 'k'</td><td>Process Data</td><td>Write to W</td></tr> </table>				Q1	Q2	Q3	Q4	Decode	Read literal 'k'	Process Data	Write to W
Q1	Q2	Q3	Q4									
Decode	Read literal 'k'	Process Data	Write to W									

Example: ADDLW 15h

Before Instruction

W = 10h

After Instruction

W = 25h

ADDWF	ADD W to f											
Syntax:	ADDWF f {,d {,a}}											
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]											
Operation:	(W) + (f) → dest											
Status Affected:	N, OV, C, DC, Z											
Encoding:	0010	01da	ffff	ffff								
Description:	Add W to register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).											
	If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.											
Words:	1											
Cycles:	1											
Q Cycle Activity:	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write to destination</td></tr> </table>				Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write to destination
Q1	Q2	Q3	Q4									
Decode	Read register 'f'	Process Data	Write to destination									

Example: ADDWF REG, 0, 0

Before Instruction

W = 17h

REG = 0C2h

After Instruction

W = 0D9h

REG = 0C2h

Note: All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction format then becomes: {label} instruction argument(s).

PIC18F87J11 FAMILY

ADDWFC	ADD W and Carry bit to f								
Syntax:	ADDWFC f {,d {,a}}								
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$								
Operation:	$(W) + (f) + (C) \rightarrow \text{dest}$								
Status Affected:	N, OV, C, DC, Z								
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>0010</td> <td>00da</td> <td>ffff</td> <td>ffff</td> </tr> </table>	0010	00da	ffff	ffff				
0010	00da	ffff	ffff						
Description:	Add W, the Carry flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process Data</td> <td>Write to destination</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write to destination						

Example: ADDWFC REG, 0, 1

Before Instruction

Carry bit = 1
REG = 02h
W = 4Dh

After Instruction

Carry bit = 0
REG = 02h
W = 50h

ANDLW	AND Literal with W								
Syntax:	ANDLW k								
Operands:	$0 \leq k \leq 255$								
Operation:	$(W) .AND. k \rightarrow W$								
Status Affected:	N, Z								
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>0000</td> <td>1011</td> <td>kkkk</td> <td>kkkk</td> </tr> </table>	0000	1011	kkkk	kkkk				
0000	1011	kkkk	kkkk						
Description:	The contents of W are ANDed with the 8-bit literal 'k'. The result is placed in W.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> <tr> <td>Decode</td> <td>Read literal 'k'</td> <td>Process Data</td> <td>Write to W</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'k'	Process Data	Write to W
Q1	Q2	Q3	Q4						
Decode	Read literal 'k'	Process Data	Write to W						

Example: ANDLW 05Fh

Before Instruction

W = A3h

After Instruction

W = 03h

ANDWF	AND W with f								
Syntax:	ANDWF f {,d {,a}}								
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$								
Operation:	$(W) .AND. (f) \rightarrow \text{dest}$								
Status Affected:	N, Z								
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0001</td><td>01da</td><td>ffff</td><td>ffff</td></tr> </table>	0001	01da	ffff	ffff				
0001	01da	ffff	ffff						
Description:	<p>The contents of W are ANDed with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).</p> <p>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.</p>								
Words:	1								
Cycles:	1								
Q Cycle Activity:									
	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr><td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write to destination</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write to destination						

Example: ANDWF REG, 0, 0

Before Instruction

W = 17h
REG = C2h

After Instruction

W = 02h
REG = C2h

BC	Branch if Carry												
Syntax:	BC n												
Operands:	$-128 \leq n \leq 127$												
Operation:	if Carry bit is '1', $(PC) + 2 + 2n \rightarrow PC$												
Status Affected:	None												
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>1110</td><td>0010</td><td>nnnn</td><td>nnnn</td></tr> </table>	1110	0010	nnnn	nnnn								
1110	0010	nnnn	nnnn										
Description:	If the Carry bit is '1', then the program will branch.												
	The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 2 + 2n$. This instruction is then a two-cycle instruction.												
Words:	1												
Cycles:	1(2)												
Q Cycle Activity:													
If Jump:													
	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr><td>Decode</td><td>Read literal 'n'</td><td>Process Data</td><td>Write to PC</td></tr> <tr><td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'n'	Process Data	Write to PC	No operation	No operation	No operation	No operation
Q1	Q2	Q3	Q4										
Decode	Read literal 'n'	Process Data	Write to PC										
No operation	No operation	No operation	No operation										
If No Jump:													
	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr><td>Decode</td><td>Read literal 'n'</td><td>Process Data</td><td>No operation</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'n'	Process Data	No operation				
Q1	Q2	Q3	Q4										
Decode	Read literal 'n'	Process Data	No operation										

Example: HERE BC 5

Before Instruction

PC = address (HERE)

After Instruction

If Carry PC = 1;
If Carry PC = address (HERE + 12)
If Carry PC = 0;
If Carry PC = address (HERE + 2)

PIC18F87J11 FAMILY

BCF Bit Clear f									
Syntax:	<code>BCF f, b {,a}</code>								
Operands:	$0 \leq f \leq 255$ $0 \leq b \leq 7$ $a \in [0,1]$								
Operation:	$0 \rightarrow f $								
Status Affected:	None								
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>1001</td> <td>bbba</td> <td>ffff</td> <td>ffff</td> </tr> </table>	1001	bbba	ffff	ffff				
1001	bbba	ffff	ffff						
Description:	Bit 'b' in register 'f' is cleared. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process Data</td> <td>Write register 'f'</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write register 'f'
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write register 'f'						

Example: `BCF FLAG_REG, 7, 0`

Before Instruction
`FLAG_REG = C7h`

After Instruction
`FLAG_REG = 47h`

BN Branch if Negative													
Syntax:	<code>BN n</code>												
Operands:	$-128 \leq n \leq 127$												
Operation:	if Negative bit is '1', $(PC) + 2 + 2n \rightarrow PC$												
Status Affected:	None												
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>1110</td> <td>0110</td> <td>nnnn</td> <td>nnnn</td> </tr> </table>	1110	0110	nnnn	nnnn								
1110	0110	nnnn	nnnn										
Description:	If the Negative bit is '1', then the program will branch.												
	The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 2 + 2n$. This instruction is then a two-cycle instruction.												
Words:	1												
Cycles:	1(2)												
Q Cycle Activity:													
If Jump:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> <tr> <td>Decode</td> <td>Read literal 'n'</td> <td>Process Data</td> <td>Write to PC</td> </tr> <tr> <td>No operation</td> <td>No operation</td> <td>No operation</td> <td>No operation</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'n'	Process Data	Write to PC	No operation	No operation	No operation	No operation
Q1	Q2	Q3	Q4										
Decode	Read literal 'n'	Process Data	Write to PC										
No operation	No operation	No operation	No operation										
If No Jump:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> <tr> <td>Decode</td> <td>Read literal 'n'</td> <td>Process Data</td> <td>No operation</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'n'	Process Data	No operation				
Q1	Q2	Q3	Q4										
Decode	Read literal 'n'	Process Data	No operation										

Example: `HERE BN Jump`

Before Instruction
`PC = address (HERE)`

After Instruction
`If Negative PC = 1; address (Jump)`
`If Negative PC = 0; address (HERE + 2)`

PIC18F87J11 FAMILY

BNC	Branch if Not Carry			
Syntax:	BNC n			
Operands:	$-128 \leq n \leq 127$			
Operation:	if Carry bit is '0', $(PC) + 2 + 2n \rightarrow PC$			
Status Affected:	None			
Encoding:	1110	0011	nnnn	nnnn
Description:	If the Carry bit is '0', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 2 + 2n$. This instruction is then a two-cycle instruction.			
Words:	1			
Cycles:	1(2)			
Q Cycle Activity:				
If Jump:				
	Q1	Q2	Q3	Q4
	Decode	Read literal 'n'	Process Data	Write to PC
	No operation	No operation	No operation	No operation
If No Jump:				
	Q1	Q2	Q3	Q4
	Decode	Read literal 'n'	Process Data	No operation

Example: HERE BNC Jump

Before Instruction
 PC = address (HERE)

After Instruction
 If Carry PC = 0;
 If Carry PC = address (Jump)
 If Carry PC = 1;
 If Carry PC = address (HERE + 2)

BNN	Branch if Not Negative			
Syntax:	BNN n			
Operands:	$-128 \leq n \leq 127$			
Operation:	if Negative bit is '0', $(PC) + 2 + 2n \rightarrow PC$			
Status Affected:	None			
Encoding:	1110	0111	nnnn	nnnn
Description:	If the Negative bit is '0', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 2 + 2n$. This instruction is then a two-cycle instruction.			
Words:	1			
Cycles:	1(2)			
Q Cycle Activity:				
If Jump:				
	Q1	Q2	Q3	Q4
	Decode	Read literal 'n'	Process Data	Write to PC
	No operation	No operation	No operation	No operation
If No Jump:				
	Q1	Q2	Q3	Q4
	Decode	Read literal 'n'	Process Data	No operation

Example: HERE BNN Jump

Before Instruction
 PC = address (HERE)

After Instruction
 If Negative PC = 0;
 If Negative PC = address (Jump)
 If Negative PC = 1;
 If Negative PC = address (HERE + 2)

PIC18F87J11 FAMILY

BNOV Branch if Not Overflow													
Syntax:	BNOV n												
Operands:	-128 ≤ n ≤ 127												
Operation:	if Overflow bit is '0', (PC) + 2 + 2n → PC												
Status Affected:	None												
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1110</td><td>0101</td><td>nnnn</td><td>nnnn</td></tr></table>	1110	0101	nnnn	nnnn								
1110	0101	nnnn	nnnn										
Description:	If the Overflow bit is '0', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.												
Words:	1												
Cycles:	1(2)												
Q Cycle Activity:													
If Jump:	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>Decode</td><td>Read literal 'n'</td><td>Process Data</td><td>Write to PC</td></tr><tr><td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr></table>	Q1	Q2	Q3	Q4	Decode	Read literal 'n'	Process Data	Write to PC	No operation	No operation	No operation	No operation
Q1	Q2	Q3	Q4										
Decode	Read literal 'n'	Process Data	Write to PC										
No operation	No operation	No operation	No operation										
If No Jump:	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>Decode</td><td>Read literal 'n'</td><td>Process Data</td><td>No operation</td></tr></table>	Q1	Q2	Q3	Q4	Decode	Read literal 'n'	Process Data	No operation				
Q1	Q2	Q3	Q4										
Decode	Read literal 'n'	Process Data	No operation										

Example:	HERE	BNOV	Jump
Before Instruction			
PC	=	address (HERE)	
After Instruction			
If Overflow	=	0;	
PC	=	address (Jump)	
If Overflow	=	1;	
PC	=	address (HERE + 2)	

BNZ Branch if Not Zero													
Syntax:	BNZ n												
Operands:	-128 ≤ n ≤ 127												
Operation:	if Zero bit is '0', (PC) + 2 + 2n → PC												
Status Affected:	None												
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1110</td><td>0001</td><td>nnnn</td><td>nnnn</td></tr></table>	1110	0001	nnnn	nnnn								
1110	0001	nnnn	nnnn										
Description:	If the Zero bit is '0', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.												
Words:	1												
Cycles:	1(2)												
Q Cycle Activity:													
If Jump:	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>Decode</td><td>Read literal 'n'</td><td>Process Data</td><td>Write to PC</td></tr><tr><td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr></table>	Q1	Q2	Q3	Q4	Decode	Read literal 'n'	Process Data	Write to PC	No operation	No operation	No operation	No operation
Q1	Q2	Q3	Q4										
Decode	Read literal 'n'	Process Data	Write to PC										
No operation	No operation	No operation	No operation										
If No Jump:	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>Decode</td><td>Read literal 'n'</td><td>Process Data</td><td>No operation</td></tr></table>	Q1	Q2	Q3	Q4	Decode	Read literal 'n'	Process Data	No operation				
Q1	Q2	Q3	Q4										
Decode	Read literal 'n'	Process Data	No operation										

Example:	HERE	BNZ	Jump
Before Instruction			
PC	=	address (HERE)	
After Instruction			
If Zero	=	0;	
PC	=	address (Jump)	
If Zero	=	1;	
PC	=	address (HERE + 2)	

PIC18F87J11 FAMILY

BRA	Unconditional Branch												
Syntax:	BRA n												
Operands:	-1024 ≤ n ≤ 1023												
Operation:	(PC) + 2 + 2n → PC												
Status Affected:	None												
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>1101</td><td>0nnn</td><td>nnnn</td><td>nnnn</td></tr> </table>	1101	0nnn	nnnn	nnnn								
1101	0nnn	nnnn	nnnn										
Description:	Add the 2's complement number '2n' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is a two-cycle instruction.												
Words:	1												
Cycles:	2												
Q Cycle Activity:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Q1</th> <th style="text-align: center;">Q2</th> <th style="text-align: center;">Q3</th> <th style="text-align: center;">Q4</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Decode</td> <td style="text-align: center;">Read literal 'n'</td> <td style="text-align: center;">Process Data</td> <td style="text-align: center;">Write to PC</td> </tr> <tr> <td style="text-align: center;">No operation</td> <td style="text-align: center;">No operation</td> <td style="text-align: center;">No operation</td> <td style="text-align: center;">No operation</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'n'	Process Data	Write to PC	No operation	No operation	No operation	No operation
Q1	Q2	Q3	Q4										
Decode	Read literal 'n'	Process Data	Write to PC										
No operation	No operation	No operation	No operation										

Example: HERE BRA Jump

Before Instruction

PC = address (HERE)

After Instruction

PC = address (Jump)

BSF	Bit Set f				
Syntax:	BSF f, b {,a}				
Operands:	0 ≤ f ≤ 255 0 ≤ b ≤ 7 a ∈ [0,1]				
Operation:	1 → f				
Status Affected:	None				
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>1000</td><td>bbba</td><td>ffff</td><td>ffff</td></tr> </table>	1000	bbba	ffff	ffff
1000	bbba	ffff	ffff		
Description:	Bit 'b' in register 'f' is set. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).				
	If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.				

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f'

Example: BSF FLAG_REG, 7, 1

Before Instruction

FLAG_REG = 0Ah

After Instruction

FLAG_REG = 8Ah

PIC18F87J11 FAMILY

BTFSC	Bit Test File, Skip if Clear				
Syntax:	BTFSC f, b {,a}				
Operands:	$0 \leq f \leq 255$ $0 \leq b \leq 7$ $a \in [0,1]$				
Operation:	skip if $(f < b) = 0$				
Status Affected:	None				
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>1011</td><td>bbba</td><td>ffff</td><td>ffff</td></tr> </table>	1011	bbba	ffff	ffff
1011	bbba	ffff	ffff		
Description:	<p>If bit 'b' in register 'f' is '0', then the next instruction is skipped. If bit 'b' is '0', then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this a two-cycle instruction.</p> <p>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.</p>				
Words:	1				
Cycles:	1(2)				
Note:	3 cycles if skip and followed by a 2-word instruction.				

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example: HERE BTFSC FLAG, 1, 0
 FALSE :
 TRUE :

Before Instruction

PC = address (HERE)

After Instruction

If FLAG<1> = 0;

PC = address (TRUE)

If FLAG<1> = 1;

PC = address (FALSE)

BTFS	Bit Test File, Skip if Set				
Syntax:	BTFS f, b {,a}				
Operands:	$0 \leq f \leq 255$ $0 \leq b < 7$ $a \in [0,1]$				
Operation:	skip if $(f < b) = 1$				
Status Affected:	None				
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>1010</td><td>bbba</td><td>ffff</td><td>ffff</td></tr> </table>	1010	bbba	ffff	ffff
1010	bbba	ffff	ffff		
Description:	<p>If bit 'b' in register 'f' is '1', then the next instruction is skipped. If bit 'b' is '1', then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this a two-cycle instruction.</p> <p>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.</p>				

Words:

1

Cycles:

1(2)

Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example: HERE BTFS FLAG, 1, 0
 FALSE :
 TRUE :

Before Instruction

PC = address (HERE)

After Instruction

If FLAG<1> = 0;

PC = address (FALSE)

If FLAG<1> = 1;

PC = address (TRUE)

BTG	Bit Toggle f								
Syntax:	BTG f, b {,a}								
Operands:	0 ≤ f ≤ 255 0 ≤ b < 7 a ∈ [0,1]								
Operation:	$(\overline{f}) \rightarrow f$								
Status Affected:	None								
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0111</td><td>bbba</td><td>ffff</td><td>ffff</td></tr> </table>	0111	bbba	ffff	ffff				
0111	bbba	ffff	ffff						
Description:	<p>Bit 'b' in data memory location 'f' is inverted.</p> <p>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.</p>								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write register 'f'</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write register 'f'
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write register 'f'						

Example: BTG PORTC, 4, 0

Before Instruction:

PORTC = 0111 0101 [75h]

After Instruction:

PORTC = 0110 0101 [65h]

BOV	Branch if Overflow												
Syntax:	BOV n												
Operands:	-128 ≤ n ≤ 127												
Operation:	if Overflow bit is '1', (PC) + 2 + 2n → PC												
Status Affected:	None												
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>1110</td><td>0100</td><td>nnnn</td><td>nnnn</td></tr> </table>	1110	0100	nnnn	nnnn								
1110	0100	nnnn	nnnn										
Description:	<p>If the Overflow bit is '1', then the program will branch.</p> <p>The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.</p>												
Words:	1												
Cycles:	1(2)												
Q Cycle Activity:	<p>If Jump:</p> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>Decode</td><td>Read literal 'n'</td><td>Process Data</td><td>Write to PC</td></tr> <tr> <td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'n'	Process Data	Write to PC	No operation	No operation	No operation	No operation
Q1	Q2	Q3	Q4										
Decode	Read literal 'n'	Process Data	Write to PC										
No operation	No operation	No operation	No operation										
If No Jump:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>Decode</td><td>Read literal 'n'</td><td>Process Data</td><td>No operation</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'n'	Process Data	No operation				
Q1	Q2	Q3	Q4										
Decode	Read literal 'n'	Process Data	No operation										

Example: HERE BOV Jump

Before Instruction
 $\text{PC} = \text{address (HERE)}$

After Instruction
 $\begin{array}{ll} \text{If Overflow} & = 1; \\ \text{PC} & = \text{address (Jump)} \\ \text{If Overflow} & = 0; \\ \text{PC} & = \text{address (HERE + 2)} \end{array}$

PIC18F87J11 FAMILY

BZ	Branch if Zero	CALL	Subroutine Call																				
Syntax:	BZ n	Syntax:	CALL k {s}																				
Operands:	-128 ≤ n ≤ 127	Operands:	0 ≤ k ≤ 1048575 s ∈ [0,1]																				
Operation:	if Zero bit is '1', (PC) + 2 + 2n → PC	Operation:	(PC) + 4 → TOS, k → PC<20:1>; if s = 1, (W) → WS, (STATUS) → STATUSS, (BSR) → BSRS																				
Status Affected:	None	Status Affected:	None																				
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>1110</td><td>0000</td><td>nnnn</td><td>nnnn</td></tr> </table>	1110	0000	nnnn	nnnn	Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>1110</td><td>110s</td><td>k₇kkk</td><td>kkk₀</td></tr> <tr><td>1111</td><td>k₁₉kkk</td><td>kkkk</td><td>kkkk₈</td></tr> </table>	1110	110s	k ₇ kkk	kkk ₀	1111	k ₁₉ kkk	kkkk	kkkk ₈								
1110	0000	nnnn	nnnn																				
1110	110s	k ₇ kkk	kkk ₀																				
1111	k ₁₉ kkk	kkkk	kkkk ₈																				
Description:	If the Zero bit is '1', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.	Description:	Subroutine call of entire 2-Mbyte memory range. First, return address (PC + 4) is pushed onto the return stack. If 's' = 1, the W, STATUS and BSR registers are also pushed into their respective shadow registers, WS, STATUSS and BSRS. If 's' = 0, no update occurs (default). Then, the 20-bit value 'k' is loaded into PC<20:1>. CALL is a two-cycle instruction.																				
Words:	1	Words:	2																				
Cycles:	1(2)	Cycles:	2																				
Q Cycle Activity:		Q Cycle Activity:																					
If Jump:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr><td>Decode</td><td>Read literal 'n'</td><td>Process Data</td><td>Write to PC</td></tr> <tr><td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'n'	Process Data	Write to PC	No operation	No operation	No operation	No operation	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr><td>Decode</td><td>Read literal 'n'</td><td>Process Data</td><td>No operation</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'n'	Process Data	No operation	
Q1	Q2	Q3	Q4																				
Decode	Read literal 'n'	Process Data	Write to PC																				
No operation	No operation	No operation	No operation																				
Q1	Q2	Q3	Q4																				
Decode	Read literal 'n'	Process Data	No operation																				
If No Jump:		<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr><td>Decode</td><td>Read literal 'n'</td><td>Process Data</td><td>No operation</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'n'	Process Data	No operation	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr><td>Decode</td><td>Read literal 'k'<7:0>,</td><td>Push PC to stack</td><td>Read literal 'k'<19:8>, Write to PC</td></tr> <tr><td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'k'<7:0>,	Push PC to stack	Read literal 'k'<19:8>, Write to PC	No operation	No operation	No operation	No operation
Q1	Q2	Q3	Q4																				
Decode	Read literal 'n'	Process Data	No operation																				
Q1	Q2	Q3	Q4																				
Decode	Read literal 'k'<7:0>,	Push PC to stack	Read literal 'k'<19:8>, Write to PC																				
No operation	No operation	No operation	No operation																				
Example:	HERE BZ Jump	Example:	HERE CALL THERE, 1																				
Before Instruction		Before Instruction																					
PC = address (HERE)		PC = address (HERE)																					
After Instruction		After Instruction																					
If Zero = 1; PC = address (Jump)		PC = address (THERE) TOS = address (HERE + 4) WS = W BSRS = BSR STATUSS = STATUS																					
If Zero = 0; PC = address (HERE + 2)																							

CLRF	Clear f								
Syntax:	CLRF f {,a}								
Operands:	$0 \leq f \leq 255$ $a \in [0,1]$								
Operation:	$000h \rightarrow f,$ $1 \rightarrow Z$								
Status Affected:	Z								
Encoding:	<table border="1"><tr><td>0110</td><td>101a</td><td>ffff</td><td>ffff</td></tr></table>	0110	101a	ffff	ffff				
0110	101a	ffff	ffff						
Description:	<p>Clears the contents of the specified register.</p> <p>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.</p>								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write register 'f'</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write register 'f'
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write register 'f'						

Example: CLRF FLAG_REG, 1

Before Instruction
FLAG_REG = 5Ah
After Instruction
FLAG_REG = 00h

CLRWD	Clear Watchdog Timer								
Syntax:	CLRWD								
Operands:	None								
Operation:	$000h \rightarrow WDT,$ $000h \rightarrow WDT postscaler,$ $1 \rightarrow \overline{TO},$ $1 \rightarrow \overline{PD}$								
Status Affected:	$\overline{TO}, \overline{PD}$								
Encoding:	<table border="1"><tr><td>0000</td><td>0000</td><td>0000</td><td>0100</td></tr></table>	0000	0000	0000	0100				
0000	0000	0000	0100						
Description:	<p>CLRWD instruction resets the Watchdog Timer. It also resets the postscaler of the WDT. Status bits, TO and PD, are set.</p>								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>Decode</td><td>No operation</td><td>Process Data</td><td>No operation</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	No operation	Process Data	No operation
Q1	Q2	Q3	Q4						
Decode	No operation	Process Data	No operation						

Example: CLRWD

Before Instruction		
WDT Counter	=	?
After Instruction		
WDT Counter	=	00h
WDT Postscaler	=	0
TO	=	1
PD	=	1

PIC18F87J11 FAMILY

COMF	Complement f								
Syntax:	COMF f {,d {,a}}								
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$								
Operation:	$f \rightarrow \text{dest}$								
Status Affected:	N, Z								
Encoding:	0001 11da ffff ffff								
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' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write to destination</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write to destination						

Example: COMF REG, 0, 0

Before Instruction
REG = 13h
After Instruction
REG = 13h
W = EC_h

CPFSEQ	Compare f with W, Skip if f = W
Syntax:	CPFSEQ f {,a}
Operands:	$0 \leq f \leq 255$ $a \in [0,1]$
Operation:	$(f) - (W)$, skip if $(f) = (W)$ (unsigned comparison)
Status Affected:	None
Encoding:	0110 001a ffff ffff
Description:	Compares the contents of data memory location 'f' to the contents of W by performing an unsigned subtraction. If 'f' = W, then the fetched instruction is discarded and a NOP is executed instead, making this a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.
Words:	1
Cycles:	1(2) Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example: HERE CPFSEQ REG, 0
NEQUAL :
EQUAL :

Before Instruction

PC Address = HERE
W = ?
REG = ?

After Instruction

If REG = W;
PC = Address (EQUAL)
If REG ≠ W;
PC = Address (NEQUAL)

CPFSGT	Compare f with W, Skip if f > W																	
Syntax:	CPFSGT f {,a}																	
Operands:	$0 \leq f \leq 255$ $a \in [0,1]$																	
Operation:	$(f) - (W)$, skip if $(f) > (W)$ (unsigned comparison)																	
Status Affected:	None																	
Encoding:	0110	010a	ffff	ffff														
Description:	Compares the contents of data memory location 'f' to the contents of the W by performing an unsigned subtraction. If the contents of 'f' are greater than the contents of WREG, then the fetched instruction is discarded and a NOP is executed instead, making this a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.																	
Words:	1																	
Cycles:	1(2) Note: 3 cycles if skip and followed by a 2-word instruction.																	
Q Cycle Activity:	<table border="1"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>No operation</td></tr> </table>				Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	No operation						
Q1	Q2	Q3	Q4															
Decode	Read register 'f'	Process Data	No operation															
If skip:	<table border="1"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr> </table>				Q1	Q2	Q3	Q4	No operation	No operation	No operation	No operation						
Q1	Q2	Q3	Q4															
No operation	No operation	No operation	No operation															
If skip and followed by 2-word instruction:	<table border="1"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr> <tr> <td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr> </table>				Q1	Q2	Q3	Q4	No operation	No operation	No operation	No operation	No operation	No operation	No operation	No operation		
Q1	Q2	Q3	Q4															
No operation	No operation	No operation	No operation															
No operation	No operation	No operation	No operation															
Example:	<p>HERE CPFSGT REG, 0 NGREATER : GREATER :</p> <p>Before Instruction</p> <table> <tr> <td>PC</td><td>=</td><td>Address (HERE)</td></tr> <tr> <td>W</td><td>=</td><td>?</td></tr> </table> <p>After Instruction</p> <table> <tr> <td>If REG ></td><td>W;</td></tr> <tr> <td>PC</td><td>= Address (GREATER)</td></tr> <tr> <td>If REG \leq</td><td>W;</td></tr> <tr> <td>PC</td><td>= Address (NGREATER)</td></tr> </table>				PC	=	Address (HERE)	W	=	?	If REG >	W;	PC	= Address (GREATER)	If REG \leq	W;	PC	= Address (NGREATER)
PC	=	Address (HERE)																
W	=	?																
If REG >	W;																	
PC	= Address (GREATER)																	
If REG \leq	W;																	
PC	= Address (NGREATER)																	

CPFSLT	Compare f with W, Skip if f < W															
Syntax:	CPFSLT f {,a}															
Operands:	$0 \leq f \leq 255$ $a \in [0,1]$															
Operation:	$(f) - (W)$, skip if $(f) < (W)$ (unsigned comparison)															
Status Affected:	None															
Encoding:	0110	000a	ffff	ffff												
Description:	Compares the contents of data memory location 'f' to the contents of W by performing an unsigned subtraction. If the contents of 'f' are less than the contents of W, then the fetched instruction is discarded and a NOP is executed instead, making this a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).															
Words:	1															
Cycles:	1(2) Note: 3 cycles if skip and followed by a 2-word instruction.															
Q Cycle Activity:	<table border="1"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>No operation</td></tr> </table>				Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	No operation				
Q1	Q2	Q3	Q4													
Decode	Read register 'f'	Process Data	No operation													
If skip:	<table border="1"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr> </table>				Q1	Q2	Q3	Q4	No operation	No operation	No operation	No operation				
Q1	Q2	Q3	Q4													
No operation	No operation	No operation	No operation													
If skip and followed by 2-word instruction:	<table border="1"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr> <tr> <td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr> </table>				Q1	Q2	Q3	Q4	No operation	No operation	No operation	No operation	No operation	No operation	No operation	No operation
Q1	Q2	Q3	Q4													
No operation	No operation	No operation	No operation													
No operation	No operation	No operation	No operation													

Example: HERE CPFSLT REG, 1
NLESS :
LESS :

Before Instruction
PC = Address (HERE)
W = ?

After Instruction
If REG < W;
PC = Address (LESS)
If REG \geq W;
PC = Address (NLESS)

PIC18F87J11 FAMILY

DAW	Decimal Adjust W Register	DECF	Decrement f																
Syntax:	DAW	Syntax:	DECF f {,d {,a}}																
Operands:	None	Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]																
Operation:	If [W<3:0> > 9] or [DC = 1] then, (W<3:0>) + 6 → W<3:0>; else, (W<3:0>) → W<3:0> If [W<7:4> > 9] or [C = 1] then, (W<7:4>) + 6 → W<7:4>, C = 1; else, (W<7:4>) → W<7:4>	Operation:	(f) – 1 → dest																
Status Affected:	C	Status Affected:	C, DC, N, OV, Z																
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0000</td><td>0000</td><td>0000</td><td>0111</td></tr> </table>	0000	0000	0000	0111	Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0000</td><td>01da</td><td>ffff</td><td>ffff</td></tr> </table>	0000	01da	ffff	ffff								
0000	0000	0000	0111																
0000	01da	ffff	ffff																
Description:	DAW adjusts the eight-bit value in W, resulting from the earlier addition of two variables (each in packed BCD format) and produces a correct packed BCD result.	Description:	Decrement register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).																
Words:	1	Words:	1																
Cycles:	1	Cycles:	1																
Q Cycle Activity:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr><td>Decode</td><td>Read register W</td><td>Process Data</td><td>Write W</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register W	Process Data	Write W	Q Cycle Activity:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr><td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write to destination</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write to destination
Q1	Q2	Q3	Q4																
Decode	Read register W	Process Data	Write W																
Q1	Q2	Q3	Q4																
Decode	Read register 'f'	Process Data	Write to destination																

Example 1: DAW

Before Instruction

W	=	A5h
C	=	0
DC	=	0

After Instruction

W	=	05h
C	=	1
DC	=	0

Example 2:

Before Instruction

W	=	CEh
C	=	0
DC	=	0

After Instruction

W	=	34h
C	=	1
DC	=	0

Words:
1
Cycles:
1
Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: DECF CNT, 1, 0

Before Instruction

CNT	=	01h
Z	=	0

After Instruction

CNT	=	00h
Z	=	1

DECFSZ	Decrement f, Skip if 0
Syntax:	DECFSZ f {,d {,a}}
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	$(f) - 1 \rightarrow \text{dest}$, skip if result = 0
Status Affected:	None
Encoding:	0010 11da ffff ffff
Description:	The contents of register 'f' are decremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). If the result is '0', the next instruction which is already fetched is discarded and a NOP is executed instead, making it a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.
Words:	1
Cycles:	1(2)
Note:	3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

HERE DECFSZ CNT, 1

GOTO LOOP

CONTINUE

Before Instruction

PC = Address (HERE)

After Instruction

CNT = CNT - 1

If CNT = 0;

PC = Address (CONTINUE)

If CNT ≠ 0;

PC = Address (HERE + 2)

DCFSNZ	Decrement f, Skip if not 0
Syntax:	DCFSNZ f {,d {,a}}
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	$(f) - 1 \rightarrow \text{dest}$, skip if result ≠ 0
Status Affected:	None
Encoding:	0100 11da ffff ffff
Description:	The contents of register 'f' are decremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). If the result is not '0', the next instruction which is already fetched is discarded and a NOP is executed instead, making it a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.
Words:	1
Cycles:	1(2)
Note:	3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

HERE DCFSNZ TEMP, 1, 0

ZERO :

NZERO :

Before Instruction

TEMP = ?

After Instruction

TEMP = TEMP - 1,

If TEMP = 0;

PC = Address (ZERO)

If TEMP ≠ 0;

PC = Address (NZERO)

PIC18F87J11 FAMILY

GOTO	Unconditional Branch	INCF	Increment f	
Syntax:	GOTO k	Syntax:	INCF f {,d {,a}}	
Operands:	$0 \leq k \leq 1048575$	Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$	
Operation:	$k \rightarrow PC<20:1>$	Operation:	$(f) + 1 \rightarrow dest$	
Status Affected:	None	Status Affected:	C, DC, N, OV, Z	
Encoding:		Encoding:		
1st word (k<7:0>)	1110 1111	k ₁₉ kkk	k ₇ kkk	
2nd word(k<19:8>)			k ₈ kkk ₀	
Description:	GOTO allows an unconditional branch anywhere within entire 2-Mbyte memory range. The 20-bit value 'k' is loaded into PC<20:1>. GOTO is always a two-cycle instruction.	Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).	
Words:	2			
Cycles:	2			
Q Cycle Activity:				
	Q1 Q2 Q3 Q4			
	Decode	Read literal 'k'<7:0>,	No operation	Read literal 'k'<19:8>, Write to PC
	No operation	No operation	No operation	No operation

Example: GOTO THERE

After Instruction
PC = Address (THERE)

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: INCF CNT, 1, 0

Before Instruction

CNT	=	FFh
Z	=	0
C	=	?
DC	=	?

After Instruction

CNT	=	00h
Z	=	1
C	=	1
DC	=	1

INCSFZ	Increment f, Skip if 0
Syntax:	INCSFZ f {,d {,a}}
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	$(f) + 1 \rightarrow \text{dest}$, skip if result = 0
Status Affected:	None
Encoding:	0011 11da ffff ffff
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f'. (default) If the result is '0', the next instruction which is already fetched is discarded and a NOP is executed instead, making it a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.
Words:	1
Cycles:	1(2)
Note:	3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

```
HERE    INCSFZ   CNT, 1, 0
NZERO   :
ZERO    :
```

Before Instruction

PC = Address (HERE)

After Instruction

```
CNT = CNT + 1
If CNT = 0;
PC = Address (ZERO)
If CNT ≠ 0;
PC = Address (NZERO)
```

INFSNZ	Increment f, Skip if not 0
Syntax:	INFSNZ f {,d {,a}}
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	$(f) + 1 \rightarrow \text{dest}$, skip if result ≠ 0
Status Affected:	None
Encoding:	0100 10da ffff ffff
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). If the result is not '0', the next instruction which is already fetched is discarded and a NOP is executed instead, making it a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.
Words:	1
Cycles:	1(2)
Note:	3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

```
HERE    INFSNZ   REG, 1, 0
ZERO   :
NZERO  :
```

Before Instruction

PC = Address (HERE)

After Instruction

```
REG = REG + 1
If REG ≠ 0;
PC = Address (NZERO)
If REG = 0;
PC = Address (ZERO)
```

PIC18F87J11 FAMILY

IORLW	Inclusive OR Literal with W								
Syntax:	IORLW k								
Operands:	$0 \leq k \leq 255$								
Operation:	$(W) .OR. k \rightarrow W$								
Status Affected:	N, Z								
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>0000</td> <td>1001</td> <td>kkkk</td> <td>kkkk</td> </tr> </table>	0000	1001	kkkk	kkkk				
0000	1001	kkkk	kkkk						
Description:	The contents of W are ORed with the eight-bit literal 'k'. The result is placed in W.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center;">Q1</th> <th style="text-align: center;">Q2</th> <th style="text-align: center;">Q3</th> <th style="text-align: center;">Q4</th> </tr> <tr> <td style="text-align: center;">Decode</td> <td style="text-align: center;">Read literal 'k'</td> <td style="text-align: center;">Process Data</td> <td style="text-align: center;">Write to W</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'k'	Process Data	Write to W
Q1	Q2	Q3	Q4						
Decode	Read literal 'k'	Process Data	Write to W						

Example: IORLW 35h

Before Instruction
 $W = 9Ah$
After Instruction
 $W = BFh$

IORWF	Inclusive OR W with f				
Syntax:	IORWF f {,d {,a}}				
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$				
Operation:	$(W) .OR. (f) \rightarrow dest$				
Status Affected:	N, Z				
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>0001</td> <td>00da</td> <td>ffff</td> <td>ffff</td> </tr> </table>	0001	00da	ffff	ffff
0001	00da	ffff	ffff		
Description:	Inclusive OR W with register 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).				

If 'a' is '0', the Access Bank is selected.
If 'a' is '1', the BSR is used to select the GPR bank (default).
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1
Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: IORWF RESULT, 0, 1

Before Instruction
 $RESULT = 13h$
 $W = 91h$
After Instruction
 $RESULT = 13h$
 $W = 93h$

PIC18F87J11 FAMILY

LFSR	Load FSR															
Syntax:	LFSR f, k															
Operands:	0 ≤ f ≤ 2 0 ≤ k ≤ 4095															
Operation:	$k \rightarrow \text{FSR}_f$															
Status Affected:	None															
Encoding:	1110 1111	1110 0000	00ff k ₇ kkk	k ₁₁ kkk kkkk												
Description:	The 12-bit literal 'k' is loaded into the file select register pointed to by 'f'.															
Words:	2															
Cycles:	2															
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> </thead> <tbody> <tr> <td>Decode</td><td>Read literal 'k' MSB</td><td>Process Data</td><td>Write literal 'k' MSB to FSR_{fH}</td></tr> <tr> <td>Decode</td><td>Read literal 'k' LSB</td><td>Process Data</td><td>Write literal 'k' to FSR_{fL}</td></tr> </tbody> </table>				Q1	Q2	Q3	Q4	Decode	Read literal 'k' MSB	Process Data	Write literal 'k' MSB to FSR _{fH}	Decode	Read literal 'k' LSB	Process Data	Write literal 'k' to FSR _{fL}
Q1	Q2	Q3	Q4													
Decode	Read literal 'k' MSB	Process Data	Write literal 'k' MSB to FSR _{fH}													
Decode	Read literal 'k' LSB	Process Data	Write literal 'k' to FSR _{fL}													

Example: LFSR 2, 3ABh

After Instruction

FSR2H	=	03h
FSR2L	=	ABh

MOVF	Move f											
Syntax:	MOVF f {,d {,a}}											
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]											
Operation:	$f \rightarrow \text{dest}$											
Status Affected:	N, Z											
Encoding:	0101	00da	ffff	ffff								
Description:	The contents of register 'f' are moved to a destination dependent upon the status of 'd'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). Location 'f' can be anywhere in the 256-byte bank.											
	If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).											
	If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.											
Words:	1											
Cycles:	1											
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> </thead> <tbody> <tr> <td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write W</td></tr> </tbody> </table>				Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write W
Q1	Q2	Q3	Q4									
Decode	Read register 'f'	Process Data	Write W									

Example: MOVF REG, 0, 0

Before Instruction

REG	=	22h
W	=	FFh

After Instruction

REG	=	22h
W	=	22h

PIC18F87J11 FAMILY

MOVFF	Move f to f												
Syntax:	MOVFF f _s ,f _d												
Operands:	0 ≤ f _s ≤ 4095 0 ≤ f _d ≤ 4095												
Operation:	(f _s) → f _d												
Status Affected:	None												
Encoding:													
1st word (source)	1100 ffff ffff ffff _s												
2nd word (destin.)	1111 ffff ffff ffff _d												
Description:	<p>The contents of source register 'f_s' are moved to destination register 'f_d'. Location of source 'f_s' can be anywhere in the 4096-byte data space (000h to FFFh) and location of destination 'f_d' can also be anywhere from 000h to FFFh.</p> <p>Either source or destination can be W (a useful special situation).</p> <p>MOVFF is particularly useful for transferring a data memory location to a peripheral register (such as the transmit buffer or an I/O port).</p> <p>The MOVFF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register</p>												
Words:	2												
Cycles:	2												
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read register 'f' (src)</td> <td>Process Data</td> <td>No operation</td> </tr> <tr> <td>Decode</td> <td>No operation No dummy read</td> <td>No operation</td> <td>Write register 'f' (dest)</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f' (src)	Process Data	No operation	Decode	No operation No dummy read	No operation	Write register 'f' (dest)
Q1	Q2	Q3	Q4										
Decode	Read register 'f' (src)	Process Data	No operation										
Decode	No operation No dummy read	No operation	Write register 'f' (dest)										

Example: MOVFF REG1, REG2

Before Instruction

REG1 = 33h
REG2 = 11h

After Instruction

REG1 = 33h
REG2 = 33h

MOVLB	Move Literal to Low Nibble in BSR								
Syntax:	MOVLW k								
Operands:	0 ≤ k ≤ 255								
Operation:	k → BSR								
Status Affected:	None								
Encoding:									
0000 0001 kkkk kkkk									
Description:	The eight-bit literal 'k' is loaded into the Bank Select Register (BSR). The value of BSR<7:4> always remains '0' regardless of the value of k _{7:k₄} .								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read literal 'k'</td> <td>Process Data</td> <td>Write literal 'k' to BSR</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'k'	Process Data	Write literal 'k' to BSR
Q1	Q2	Q3	Q4						
Decode	Read literal 'k'	Process Data	Write literal 'k' to BSR						

Example: MOVLB 5

Before Instruction
BSR Register = 02h
After Instruction
BSR Register = 05h

PIC18F87J11 FAMILY

MOVLW	Move Literal to W		
Syntax:	MOVLW k		
Operands:	0 ≤ k ≤ 255		
Operation:	k → W		
Status Affected:	None		
Encoding:	0000 1110 kkkk kkkk		
Description:	The eight-bit literal 'k' is loaded into W.		
Words:	1		
Cycles:	1		
Q Cycle Activity:			
	Q1 Q2 Q3 Q4		
Decode	Read literal 'k'	Process Data	Write to W

Example: MOVLW 5Ah

After Instruction

W = 5Ah

MOVWF	Move W to f		
Syntax:	MOVWF f {,a}		
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]		
Operation:	(W) → f		
Status Affected:	None		
Encoding:	0110 111a ffff ffff		
Description:	Move data from W to register 'f'. Location 'f' can be anywhere in the 256-byte bank.		
	If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).		
	If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.		
Words:	1		
Cycles:	1		
Q Cycle Activity:			
	Q1 Q2 Q3 Q4		
Decode	Read register 'f'	Process Data	Write register 'f'

Example: MOVWF REG, 0

Before Instruction

W = 4Fh
REG = FFh

After Instruction

W = 4Fh
REG = 4Fh

PIC18F87J11 FAMILY

MULLW	Multiply Literal with W								
Syntax:	MULLW k								
Operands:	$0 \leq k \leq 255$								
Operation:	$(W) \times k \rightarrow PRODH:PRODL$								
Status Affected:	None								
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0000</td><td>1101</td><td>kkkk</td><td>kkkk</td></tr> </table>	0000	1101	kkkk	kkkk				
0000	1101	kkkk	kkkk						
Description:	An unsigned multiplication is carried out between the contents of W and the 8-bit literal 'k'. The 16-bit result is placed in PRODH:PRODL register pair. PRODH contains the high byte. W is unchanged. None of the Status flags are affected. Note that neither Overflow nor Carry is possible in this operation. A Zero result is possible but not detected.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Q1</th><th style="text-align: center;">Q2</th><th style="text-align: center;">Q3</th><th style="text-align: center;">Q4</th></tr> </thead> <tbody> <tr> <td style="text-align: center;">Decode</td><td style="text-align: center;">Read literal 'k'</td><td style="text-align: center;">Process Data</td><td style="text-align: center;">Write registers PRODH: PRODL</td></tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'k'	Process Data	Write registers PRODH: PRODL
Q1	Q2	Q3	Q4						
Decode	Read literal 'k'	Process Data	Write registers PRODH: PRODL						

Example: MULLW 0C4h

Before Instruction

W	=	E2h
PRODH	=	?
PRODL	=	?

After Instruction

W	=	E2h
PRODH	=	ADh
PRODL	=	08h

MULWF	Multiply W with f								
Syntax:	MULWF f {,a}								
Operands:	$0 \leq f \leq 255$ $a \in [0,1]$								
Operation:	$(W) \times (f) \rightarrow PRODH:PRODL$								
Status Affected:	None								
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0000</td><td>001a</td><td>ffff</td><td>ffff</td></tr> </table>	0000	001a	ffff	ffff				
0000	001a	ffff	ffff						
Description:	An unsigned multiplication is carried out between the contents of W and the register file location 'f'. The 16-bit result is stored in the PRODH:PRODL register pair. PRODH contains the high byte. Both W and 'f' are unchanged. None of the Status flags are affected. Note that neither Overflow nor Carry is possible in this operation. A Zero result is possible but not detected.								
Words:	If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).								
Cycles:	If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.								
Q Cycle Activity:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Q1</th><th style="text-align: center;">Q2</th><th style="text-align: center;">Q3</th><th style="text-align: center;">Q4</th></tr> </thead> <tbody> <tr> <td style="text-align: center;">Decode</td><td style="text-align: center;">Read register 'f'</td><td style="text-align: center;">Process Data</td><td style="text-align: center;">Write registers PRODH: PRODL</td></tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write registers PRODH: PRODL
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write registers PRODH: PRODL						

Example: MULWF REG, 1

Before Instruction

W	=	C4h
REG	=	B5h
PRODH	=	?
PRODL	=	?

After Instruction

W	=	C4h
REG	=	B5h
PRODH	=	8Ah
PRODL	=	94h

NEGF	Negate f								
Syntax:	NEGF f {,a}								
Operands:	$0 \leq f \leq 255$ $a \in [0,1]$								
Operation:	$(\bar{f}) + 1 \rightarrow f$								
Status Affected:	N, OV, C, DC, Z								
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0110</td><td>110a</td><td>ffff</td><td>ffff</td></tr> </table>	0110	110a	ffff	ffff				
0110	110a	ffff	ffff						
Description:	<p>Location 'f' is negated using two's complement. The result is placed in the data memory location 'f'.</p> <p>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.</p>								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr><td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write register 'f'</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write register 'f'
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write register 'f'						

Example: NEGF REG, 1

Before Instruction

REG = 0011 1010 [3Ah]

After Instruction

REG = 1100 0110 [C6h]

NOP	No Operation								
Syntax:	NOP								
Operands:	None								
Operation:	No operation								
Status Affected:	None								
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0000</td><td>0000</td><td>0000</td><td>0000</td></tr> <tr><td>1111</td><td>xxxx</td><td>xxxx</td><td>xxxx</td></tr> </table>	0000	0000	0000	0000	1111	xxxx	xxxx	xxxx
0000	0000	0000	0000						
1111	xxxx	xxxx	xxxx						
Description:	No operation.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr><td>Decode</td><td>No operation</td><td>No operation</td><td>No operation</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	No operation	No operation	No operation
Q1	Q2	Q3	Q4						
Decode	No operation	No operation	No operation						

Example:

None.

PIC18F87J11 FAMILY

POP	Pop Top of Return Stack	PUSH	Push Top of Return Stack
Syntax:	POP	Syntax:	PUSH
Operands:	None	Operands:	None
Operation:	(TOS) → bit bucket	Operation:	(PC + 2) → TOS
Status Affected:	None	Status Affected:	None
Encoding:	0000 0000 0000 0110	Encoding:	0000 0000 0000 0101
Description:	The TOS value is pulled off the return stack and is discarded. The TOS value then becomes the previous value that was pushed onto the return stack. This instruction is provided to enable the user to properly manage the return stack to incorporate a software stack.	Description:	The PC + 2 is pushed onto the top of the return stack. The previous TOS value is pushed down on the stack. This instruction allows implementing a software stack by modifying TOS and then pushing it onto the return stack.
Words:	1	Words:	1
Cycles:	1	Cycles:	1
Q Cycle Activity:		Q Cycle Activity:	
Q1	Q2	Q3	Q4
Decode	No operation	POP TOS value	No operation

<u>Example:</u>	POP	
	GOTO	NEW
Before Instruction		
TOS	= 0031A2h	
Stack (1 level down)	= 014332h	
After Instruction		
TOS	= 014332h	
PC	= NEW	

<u>Example:</u>	PUSH	
Before Instruction		
TOS	= 345Ah	
PC	= 0124h	
After Instruction		
PC	= 0126h	
TOS	= 0126h	
Stack (1 level down)	= 345Ah	

PIC18F87J11 FAMILY

RCALL	Relative Call												
Syntax:	RCALL n												
Operands:	$-1024 \leq n \leq 1023$												
Operation:	$(PC) + 2 \rightarrow TOS,$ $(PC) + 2 + 2n \rightarrow PC$												
Status Affected:	None												
Encoding:	<table border="1"><tr><td>1101</td><td>1nnn</td><td>nnnn</td><td>nnnn</td></tr></table>	1101	1nnn	nnnn	nnnn								
1101	1nnn	nnnn	nnnn										
Description:	Subroutine call with a jump up to 1K from the current location. First, return address (PC + 2) is pushed onto the stack. Then, add the 2's complement number '2n' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 2 + 2n$. This instruction is a two-cycle instruction.												
Words:	1												
Cycles:	2												
Q Cycle Activity:	<table border="1"><thead><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr></thead><tbody><tr><td>Decode</td><td>Read literal 'n' PUSH PC to stack</td><td>Process Data</td><td>Write to PC</td></tr><tr><td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr></tbody></table>	Q1	Q2	Q3	Q4	Decode	Read literal 'n' PUSH PC to stack	Process Data	Write to PC	No operation	No operation	No operation	No operation
Q1	Q2	Q3	Q4										
Decode	Read literal 'n' PUSH PC to stack	Process Data	Write to PC										
No operation	No operation	No operation	No operation										

Example: HERE RCALL Jump

Before Instruction
 PC = Address (HERE)
 After Instruction
 PC = Address (Jump)
 TOS = Address (HERE + 2)

RESET	Reset								
Syntax:	RESET								
Operands:	None								
Operation:	Reset all registers and flags that are affected by a MCLR Reset.								
Status Affected:	All								
Encoding:	<table border="1"><tr><td>0000</td><td>0000</td><td>1111</td><td>1111</td></tr></table>	0000	0000	1111	1111				
0000	0000	1111	1111						
Description:	This instruction provides a way to execute a MCLR Reset in software.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"><thead><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr></thead><tbody><tr><td>Decode</td><td>Start reset</td><td>No operation</td><td>No operation</td></tr></tbody></table>	Q1	Q2	Q3	Q4	Decode	Start reset	No operation	No operation
Q1	Q2	Q3	Q4						
Decode	Start reset	No operation	No operation						

Example: RESET

After Instruction
 Registers = Reset Value
 Flags* = Reset Value

PIC18F87J11 FAMILY

RETFIE	Return from Interrupt	RETLW	Return Literal to W																								
Syntax:	RETFIE {s}	Syntax:	RETLW k																								
Operands:	$s \in [0,1]$	Operands:	$0 \leq k \leq 255$																								
Operation:	(TOS) \rightarrow PC, 1 \rightarrow GIE/GIEH or PEIE/GIEL; if $s = 1$, (WS) \rightarrow W, (STATUS) \rightarrow STATUS, (BSRS) \rightarrow BSR, PCLATU, PCLATH are unchanged	Operation:	$k \rightarrow W$, (TOS) \rightarrow PC, PCLATU, PCLATH are unchanged																								
Status Affected:	GIE/GIEH, PEIE/GIEL.	Status Affected:	None																								
Encoding:	<table border="1"><tr><td>0000</td><td>0000</td><td>0001</td><td>000s</td></tr></table>	0000	0000	0001	000s	Encoding:	<table border="1"><tr><td>0000</td><td>1100</td><td>kkkk</td><td>kkkk</td></tr></table>	0000	1100	kkkk	kkkk																
0000	0000	0001	000s																								
0000	1100	kkkk	kkkk																								
Description:	Return from interrupt. Stack is popped and Top-of-Stack (TOS) is loaded into the PC. Interrupts are enabled by setting either the high or low-priority global interrupt enable bit. If 's' = 1, the contents of the shadow registers WS, STATUS and BSRS are loaded into their corresponding registers W, STATUS and BSR. If 's' = 0, no update of these registers occurs (default).	Description:	W is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). The high address latch (PCLATH) remains unchanged.																								
Words:	1	Words:	1																								
Cycles:	2	Cycles:	2																								
Q Cycle Activity:	<table border="1"><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>Decode</td><td>Read literal 'k'</td><td>Process Data</td><td>POP PC from stack, write to W</td></tr><tr><td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr></table>	Q1	Q2	Q3	Q4	Decode	Read literal 'k'	Process Data	POP PC from stack, write to W	No operation	No operation	No operation	No operation	Q Cycle Activity:	<table border="1"><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>Decode</td><td>Read literal 'k'</td><td>Process Data</td><td>POP PC from stack, write to W</td></tr><tr><td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr></table>	Q1	Q2	Q3	Q4	Decode	Read literal 'k'	Process Data	POP PC from stack, write to W	No operation	No operation	No operation	No operation
Q1	Q2	Q3	Q4																								
Decode	Read literal 'k'	Process Data	POP PC from stack, write to W																								
No operation	No operation	No operation	No operation																								
Q1	Q2	Q3	Q4																								
Decode	Read literal 'k'	Process Data	POP PC from stack, write to W																								
No operation	No operation	No operation	No operation																								

Example: RETFIE 1

After Interrupt

PC	=	TOS
W	=	WS
BSR	=	BSRS
STATUS	=	STATUS
GIE/GIEH, PEIE/GIEL	=	1

Example:

```

CALL TABLE ; W contains table
            ; offset value
            ; W now has
            ; table value
:
TABLE
    ADDWF PCL ; W = offset
    RETLW k0 ; Begin table
    RETLW k1 ;
:
:
RETLW kn ; End of table

Before Instruction
    W      = 07h
After Instruction
    W      = value of kn

```

RETURN	Return from Subroutine	RLCF	Rotate Left f through Carry								
Syntax:	RETURN {s}	Syntax:	RLCF f {,d {,a}}								
Operands:	s ∈ [0,1]	Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]								
Operation:	(TOS) → PC; if s = 1, (WS) → W, (STATUS) → STATUS, (BSRS) → BSR, PCLATU, PCLATH are unchanged	Operation:	(f<n>) → dest<n + 1>, (f<7>) → C, (C) → dest<0>								
Status Affected:	None	Status Affected:	C, N, Z								
Encoding:	<table border="1"><tr><td>0000</td><td>0000</td><td>0001</td><td>001s</td></tr></table>	0000	0000	0001	001s	Encoding:	<table border="1"><tr><td>0011</td><td>01da</td><td>ffff</td><td>ffff</td></tr></table>	0011	01da	ffff	ffff
0000	0000	0001	001s								
0011	01da	ffff	ffff								
Description:	Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter. If 's'= 1, the contents of the shadow registers WS, STATUS and BSRS are loaded into their corresponding registers W, STATUS and BSR. If 's' = 0, no update of these registers occurs (default).	Description:	The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.								
Words:	1	Words:	1								
Cycles:	2	Cycles:	1								
Q Cycle Activity:		Q Cycle Activity:									
	Q1 Q2 Q3 Q4		Q1 Q2 Q3 Q4								
	Decode No operation Process Data POP PC from stack		Decode Read register 'f' Process Data Write to destination								
	No operation No operation No operation No operation										

Example: RETURN

After Instruction:
PC = TOS

Words: 1
Cycles: 1
Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: RLCF REG, 0, 0

Before Instruction
REG = 1110 0110
C = 0
After Instruction
REG = 1110 0110
W = 1100 1100
C = 1

PIC18F87J11 FAMILY

RLNCF	Rotate Left f (No Carry)								
Syntax:	RLNCF f {,d {,a}}								
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$								
Operation:	$(f<_{n>} \rightarrow \text{dest}<_{n+1>},$ $(f<_7>) \rightarrow \text{dest}<_0>$								
Status Affected:	N, Z								
Encoding:	0100 01da ffff ffff								
Description:	The contents of register 'f' are rotated one bit to the left. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> </thead> <tbody> <tr> <td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write to destination</td></tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write to destination						

Example: RLNCF REG, 1, 0

Before Instruction

REG = 1010 1011

After Instruction

REG = 0101 0111

RRCF	Rotate Right f through Carry								
Syntax:	RRCF f {,d {,a}}								
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$								
Operation:	$(f<_{n>} \rightarrow \text{dest}<_{n-1>},$ $(f<_0>) \rightarrow C,$ $(C) \rightarrow \text{dest}<_7>$								
Status Affected:	C, N, Z								
Encoding:	0011 00da ffff ffff								
Description:	The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> </thead> <tbody> <tr> <td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write to destination</td></tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write to destination						

Example: RRCF REG, 0, 0

Before Instruction

REG = 1110 0110

C = 0

After Instruction

REG = 1110 0110

W = 0111 0011

C = 0

RRNCF	Rotate Right f (No Carry)								
Syntax:	RRNCF f {,d ,a}								
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$								
Operation:	$(f< n >) \rightarrow \text{dest}<n - 1>$, $(f<0>) \rightarrow \text{dest}<7>$								
Status Affected:	N, Z								
Encoding:	0100 00da ffff ffff								
Description:	<p>The contents of register 'f' are rotated one bit to the right. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).</p> <p>If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' is '1', then the bank will be selected as per the BSR value (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.</p> 								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process Data</td> <td>Write register 'f'</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write register 'f'
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write register 'f'						

Example 1: RRNCF REG, 1, 0

Before Instruction
REG = 1101 0111
After Instruction
REG = 1110 1011

Example 2: RRNCF REG, 0, 0

Before Instruction
W = ?
REG = 1101 0111
After Instruction
W = 1110 1011
REG = 1101 0111

SETF	Set f								
Syntax:	SETF f {,a}								
Operands:	$0 \leq f \leq 255$ $a \in [0,1]$								
Operation:	$\text{FFh} \rightarrow f$								
Status Affected:	None								
Encoding:	0110 100a ffff ffff								
Description:	<p>The contents of the specified register are set to FFh.</p> <p>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.</p>								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process Data</td> <td>Write register 'f'</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write register 'f'
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write register 'f'						

Example: SETF REG, 1

Before Instruction		
REG	=	5Ah
After Instruction		
REG	=	FFh

PIC18F87J11 FAMILY

SLEEP	Enter Sleep Mode								
Syntax:	SLEEP								
Operands:	None								
Operation:	00h → WDT, 0 → WDT postscaler, 1 → $\overline{\text{TO}}$, 0 → $\overline{\text{PD}}$								
Status Affected:	$\overline{\text{TO}}$, $\overline{\text{PD}}$								
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0000</td><td>0000</td><td>0000</td><td>0011</td></tr> </table>	0000	0000	0000	0011				
0000	0000	0000	0011						
Description:	The Power-Down status bit ($\overline{\text{PD}}$) is cleared. The Time-out status bit ($\overline{\text{TO}}$) is set. The Watchdog Timer and its postscaler are cleared. The processor is put into Sleep mode with the oscillator stopped.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>Decode</td><td>No operation</td><td>Process Data</td><td>Go to Sleep</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	No operation	Process Data	Go to Sleep
Q1	Q2	Q3	Q4						
Decode	No operation	Process Data	Go to Sleep						

Example: SLEEP

Before Instruction

$$\begin{array}{l} \overline{\text{TO}} = ? \\ \overline{\text{PD}} = ? \end{array}$$

After Instruction

$$\begin{array}{l} \overline{\text{TO}} = 1 \dagger \\ \overline{\text{PD}} = 0 \end{array}$$

† If WDT causes wake-up, this bit is cleared.

SUBFWB	Subtract f from W with Borrow
--------	-------------------------------

Syntax:	SUBFWB f {,d {,a}}				
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$				
Operation:	$(W) - (f) - (\overline{C}) \rightarrow \text{dest}$				
Status Affected:	N, OV, C, DC, Z				
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0101</td><td>01da</td><td>ffff</td><td>ffff</td></tr> </table>	0101	01da	ffff	ffff
0101	01da	ffff	ffff		
Description:	Subtract register 'f' and Carry flag (borrow) from W (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored in register 'f' (default).				

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See

Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example 1: SUBFWB REG, 1, 0

Before Instruction

$$\begin{array}{l} \text{REG} = 3 \\ \text{W} = 2 \\ \text{C} = 1 \end{array}$$

After Instruction

$$\begin{array}{l} \text{REG} = \text{FF} \\ \text{W} = 2 \\ \text{C} = 0 \\ \text{Z} = 0 \\ \text{N} = 1 ; \text{result is negative} \end{array}$$

Example 2: SUBFWB REG, 0, 0

Before Instruction

$$\begin{array}{l} \text{REG} = 2 \\ \text{W} = 5 \\ \text{C} = 1 \end{array}$$

After Instruction

$$\begin{array}{l} \text{REG} = 2 \\ \text{W} = 3 \\ \text{C} = 1 \\ \text{Z} = 0 \\ \text{N} = 0 ; \text{result is positive} \end{array}$$

Example 3: SUBFWB REG, 1, 0

Before Instruction

$$\begin{array}{l} \text{REG} = 1 \\ \text{W} = 2 \\ \text{C} = 0 \end{array}$$

After Instruction

$$\begin{array}{l} \text{REG} = 0 \\ \text{W} = 2 \\ \text{C} = 1 \\ \text{Z} = 1 ; \text{result is zero} \\ \text{N} = 0 \end{array}$$

SUBLW	Subtract W from Literal								
Syntax:	SUBLW k								
Operands:	0 ≤ k ≤ 255								
Operation:	k – (W) → W								
Status Affected:	N, OV, C, DC, Z								
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0000</td><td>1000</td><td>kkkk</td><td>kkkk</td></tr> </table>	0000	1000	kkkk	kkkk				
0000	1000	kkkk	kkkk						
Description:	W is subtracted from the eight-bit literal 'k'. The result is placed in W.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>Decode</td><td>Read literal 'k'</td><td>Process Data</td><td>Write to W</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'k'	Process Data	Write to W
Q1	Q2	Q3	Q4						
Decode	Read literal 'k'	Process Data	Write to W						

Example 1: SUBLW 02h

Before Instruction

W = 01h
C = ?

After Instruction

W = 01h
C = 1 ; result is positive
Z = 0
N = 0

Example 2: SUBLW 02h

Before Instruction

W = 02h
C = ?

After Instruction

W = 00h
C = 1 ; result is zero
Z = 1
N = 0

Example 3: SUBLW 02h

Before Instruction

W = 03h
C = ?

After Instruction

W = FFh ; (2's complement)
C = 0 ; result is negative
Z = 0
N = 1

SUBWF	Subtract W from f				
Syntax:	SUBWF f {,d {,a}}				
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]				
Operation:	(f) – (W) → dest				
Status Affected:	N, OV, C, DC, Z				
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0101</td><td>11da</td><td>ffff</td><td>ffff</td></tr> </table>	0101	11da	ffff	ffff
0101	11da	ffff	ffff		

Description:	Subtract W from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write to destination</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write to destination						

If 'a' is '0', the Access Bank is selected.
If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words:
1

Cycles:
1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example 1: SUBWF REG, 1, 0

Before Instruction

REG = 3
W = 2
C = ?

After Instruction

REG = 1
W = 2
C = 1 ; result is positive
Z = 0
N = 0

Example 2: SUBWF REG, 0, 0

Before Instruction

REG = 2
W = 2
C = ?

After Instruction

REG = 2
W = 0
C = 1 ; result is zero
Z = 1
N = 0

Example 3: SUBWF REG, 1, 0

Before Instruction

REG = 1
W = 2
C = ?

After Instruction

REG = FFh ; (2's complement)
W = 2
C = 0 ; result is negative
Z = 0
N = 1

PIC18F87J11 FAMILY

SUBWFB	Subtract W from f with Borrow								
Syntax:	SUBWFB f {,d {,a}}								
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]								
Operation:	(f) – (W) – (C) → dest								
Status Affected:	N, OV, C, DC, Z								
Encoding:	<table border="1"><tr><td>0101</td><td>10da</td><td>ffff</td><td>ffff</td></tr></table>	0101	10da	ffff	ffff				
0101	10da	ffff	ffff						
Description:	Subtract W and the Carry flag (borrow) from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write to destination</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write to destination						

Example 1: SUBWFB REG, 1, 0

Before Instruction

REG = 19h (0001 1001)
W = 0Dh (0000 1101)
C = 1

After Instruction

REG = 0Ch (0000 1011)
W = 0Dh (0000 1101)
C = 1
Z = 0
N = 0 ; result is positive

Example 2: SUBWFB REG, 0, 0

Before Instruction

REG = 1Bh (0001 1011)
W = 1Ah (0001 1010)
C = 0

After Instruction

REG = 1Bh (0001 1011)
W = 00h
C = 1
Z = 1 ; result is zero
N = 0

Example 3: SUBWFB REG, 1, 0

Before Instruction

REG = 03h (0000 0011)
W = 0Eh (0000 1101)
C = 1

After Instruction

REG = F5h (1111 0100)
; [2's comp]
W = 0Eh (0000 1101)
C = 0
Z = 0
N = 1 ; result is negative

SWAPF	Swap f								
Syntax:	SWAPF f {,d {,a}}								
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]								
Operation:	(f<3:0>) → dest<7:4>, (f<7:4>) → dest<3:0>								
Status Affected:	None								
Encoding:	<table border="1"><tr><td>0011</td><td>10da</td><td>ffff</td><td>ffff</td></tr></table>	0011	10da	ffff	ffff				
0011	10da	ffff	ffff						
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write to destination</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write to destination						

Example: SWAPF REG, 1, 0

Before Instruction

REG = 53h

After Instruction

REG = 35h

TBLRD	Table Read				
Syntax:	TBLRD (*, *+; *-; +*)				
Operands:	None				
Operation:	if TBLRD *, (Prog Mem (TBLPTR)) → TABLAT, TBLPTR – No Change; if TBLRD *+, (Prog Mem (TBLPTR)) → TABLAT, (TBLPTR) + 1 → TBLPTR; if TBLRD *-, (Prog Mem (TBLPTR)) → TABLAT, (TBLPTR) - 1 → TBLPTR; if TBLRD +*, (TBLPTR) + 1 → TBLPTR, (Prog Mem (TBLPTR)) → TABLAT				
Status Affected:	None				
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>0000</td> <td>0000</td> <td>0000</td> <td> 10nn nn=0 * =1 *+ =2 *- =3 +* </td> </tr> </table>	0000	0000	0000	10nn nn=0 * =1 *+ =2 *- =3 +*
0000	0000	0000	10nn nn=0 * =1 *+ =2 *- =3 +*		
Description:	<p>This instruction is used to read the contents of Program Memory (P.M.). To address the program memory, a pointer called Table Pointer (TBLPTR) is used.</p> <p>The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range.</p> <p>TBLPTR<0> = 0:Least Significant Byte of Program Memory Word TBLPTR<0> = 1:Most Significant Byte of Program Memory Word</p> <p>The TBLRD instruction can modify the value of TBLPTR as follows:</p> <ul style="list-style-type: none"> • no change • post-increment • post-decrement • pre-increment 				
Words:	1				
Cycles:	2				
Q Cycle Activity:					

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Read Program Memory)	No operation	No operation (Write TABLAT)

TBLRD	Table Read (Continued)
Example 1:	TBLRD *+ ;
Before Instruction	
TABLAT	= 55h
TBLPTR	= 00A356h
MEMORY(00A356h)	= 34h
After Instruction	
TABLAT	= 34h
TBLPTR	= 00A357h
Example 2:	TBLRD +* ;
Before Instruction	
TABLAT	= AAh
TBLPTR	= 01A357h
MEMORY(01A357h)	= 12h
MEMORY(01A358h)	= 34h
After Instruction	
TABLAT	= 34h
TBLPTR	= 01A358h

PIC18F87J11 FAMILY

TBLWT	Table Write				
Syntax:	TBLWT (*; *+; *-; +*)				
Operands:	None				
Operation:	if TBLWT*, (TABLAT) → Holding Register, TBLPTR – No Change; if TBLWT*+, (TABLAT) → Holding Register, (TBLPTR) + 1 → TBLPTR; if TBLWT*-, (TABLAT) → Holding Register, (TBLPTR) - 1 → TBLPTR; if TBLWT+*, (TBLPTR) + 1 → TBLPTR, (TABLAT) → Holding Register				
Status Affected:	None				
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0000</td><td>0000</td><td>0000</td><td>11nn nn=0 * =1 *+ =2 *- =3 +*</td></tr> </table>	0000	0000	0000	11nn nn=0 * =1 *+ =2 *- =3 +*
0000	0000	0000	11nn nn=0 * =1 *+ =2 *- =3 +*		
Description:	<p>This instruction uses the 3 LSBs of TBLPTR to determine which of the 8 holding registers the TABLAT is written to. The holding registers are used to program the contents of Program Memory (P.M.). (Refer to Section 5.0 “Memory Organization” for additional details on programming Flash memory.)</p> <p>The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range. The LSb of the TBLPTR selects which byte of the program memory location to access.</p> <p style="margin-left: 40px;">TBLPTR<0> = 0:Least Significant Byte of Program Memory Word</p> <p style="margin-left: 40px;">TBLPTR<0> = 1:Most Significant Byte of Program Memory Word</p> <p>The TBLWT instruction can modify the value of TBLPTR as follows:</p> <ul style="list-style-type: none"> • no change • post-increment • post-decrement • pre-increment 				
Words:	1				
Cycles:	2				
Q Cycle Activity:					

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Read TABLAT)	No operation	No operation (Write to Holding Register)

TBLWT	Table Write (Continued)
Example 1:	TBLWT *+;
Before Instruction	
TABLAT	= 55h
TBLPTR	= 00A356h
HOLDING REGISTER (00A356h)	= FFh
After Instructions (table write completion)	
TABLAT	= 55h
TBLPTR	= 00A357h
HOLDING REGISTER (00A356h)	= 55h
Example 2:	TBLWT +*;
Before Instruction	
TABLAT	= 34h
TBLPTR	= 01389Ah
HOLDING REGISTER (01389Ah)	= FFh
HOLDING REGISTER (01389Bh)	= FFh
After Instruction (table write completion)	
TABLAT	= 34h
TBLPTR	= 01389Bh
HOLDING REGISTER (01389Ah)	= FFh
HOLDING REGISTER (01389Bh)	= 34h

TSTFSZ	Test f, Skip if 0								
Syntax:	TSTFSZ f {,a}								
Operands:	$0 \leq f \leq 255$ $a \in [0,1]$								
Operation:	skip if $f = 0$								
Status Affected:	None								
Encoding:	<table border="1"><tr><td>0110</td><td>011a</td><td>ffff</td><td>ffff</td></tr></table>	0110	011a	ffff	ffff				
0110	011a	ffff	ffff						
Description:	If 'f' = 0, the next instruction fetched during the current instruction execution is discarded and a NOP is executed, making this a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.								
Words:	1								
Cycles:	1(2) Note: 3 cycles if skip and followed by a 2-word instruction.								
Q Cycle Activity:	<table border="1"><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>No operation</td></tr></table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	No operation
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	No operation						

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example: HERE TSTFSZ CNT, 1
NZERO :
ZERO :

Before Instruction
PC = Address (HERE)
After Instruction
If CNT = 00h,
PC = Address (ZERO)
If CNT \neq 00h,
PC = Address (NZERO)

XORLW	Exclusive OR Literal with W								
Syntax:	XORLW k								
Operands:	$0 \leq k \leq 255$								
Operation:	(W) .XOR. k \rightarrow W								
Status Affected:	N, Z								
Encoding:	<table border="1"><tr><td>0000</td><td>1010</td><td>kkkk</td><td>kkkk</td></tr></table>	0000	1010	kkkk	kkkk				
0000	1010	kkkk	kkkk						
Description:	The contents of W are XORed with the 8-bit literal 'k'. The result is placed in W.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>Decode</td><td>Read literal 'k'</td><td>Process Data</td><td>Write to W</td></tr></table>	Q1	Q2	Q3	Q4	Decode	Read literal 'k'	Process Data	Write to W
Q1	Q2	Q3	Q4						
Decode	Read literal 'k'	Process Data	Write to W						

Example: XORLW 0AFh

Before Instruction
W = B5h
After Instruction
W = 1Ah

PIC18F87J11 FAMILY

XORWF Exclusive OR W with f

Syntax: XORWF f {,d {,a}}

Operands: $0 \leq f \leq 255$

$d \in [0,1]$

$a \in [0,1]$

Operation: (W) .XOR. (f) \rightarrow dest

Status Affected: N, Z

Encoding:

0001	10da	ffff	ffff
------	------	------	------

Description: Exclusive OR the contents of W with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in the register 'f' (default).

If 'a' is '0', the Access Bank is selected.
If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: XORWF REG, 1, 0

Before Instruction

REG = AFh
W = B5h

After Instruction

REG = 1Ah
W = B5h

25.2 Extended Instruction Set

In addition to the standard 75 instructions of the PIC18 instruction set, the PIC18F87J11 Family family of devices also provide an optional extension to the core CPU functionality. The added features include eight additional instructions that augment Indirect and Indexed Addressing operations and the implementation of Indexed Literal Offset Addressing for many of the standard PIC18 instructions.

The additional features of the extended instruction set are enabled by default on unprogrammed devices. Users must properly set or clear the XINST Configuration bit during programming to enable or disable these features.

The instructions in the extended set can all be classified as literal operations, which either manipulate the File Select Registers, or use them for Indexed Addressing. Two of the instructions, ADDFSR and SUBFSR, each have an additional special instantiation for using FSR2. These versions (ADDULNK and SUBULNK) allow for automatic return after execution.

The extended instructions are specifically implemented to optimize re-entrant program code (that is, code that is recursive or that uses a software stack) written in high-level languages, particularly C. Among other things, they allow users working in high-level languages to perform certain operations on data structures more efficiently. These include:

- dynamic allocation and deallocation of software stack space when entering and leaving subroutines
- function pointer invocation
- software Stack Pointer manipulation
- manipulation of variables located in a software stack

A summary of the instructions in the extended instruction set is provided in Table 25-3. Detailed descriptions are provided in **Section 25.2.2 “Extended Instruction Set”**. The opcode field descriptions in Table 25-1 (page 330) apply to both the standard and extended PIC18 instruction sets.

Note: The instruction set extension and the Indexed Literal Offset Addressing mode were designed for optimizing applications written in C; the user may likely never use these instructions directly in assembler. The syntax for these commands is provided as a reference for users who may be reviewing code that has been generated by a compiler.

25.2.1 EXTENDED INSTRUCTION SYNTAX

Most of the extended instructions use indexed arguments, using one of the File Select Registers and some offset to specify a source or destination register. When an argument for an instruction serves as part of Indexed Addressing, it is enclosed in square brackets ("[]"). This is done to indicate that the argument is used as an index or offset. The MPASM™ Assembler will flag an error if it determines that an index or offset value is not bracketed.

When the extended instruction set is enabled, brackets are also used to indicate index arguments in byte-oriented and bit-oriented instructions. This is in addition to other changes in their syntax. For more details, see **Section 25.2.3.1 “Extended Instruction Syntax with Standard PIC18 Commands”**.

Note: In the past, square brackets have been used to denote optional arguments in the PIC18 and earlier instruction sets. In this text and going forward, optional arguments are denoted by braces ("{ }").

TABLE 25-3: EXTENSIONS TO THE PIC18 INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word				Status Affected
			MSb	LSb			
ADDFSR f, k	Add Literal to FSR	1	1110	1000	ffkk	kkkk	None
ADDULNK k	Add Literal to FSR2 and Return	2	1110	1000	11kk	kkkk	None
CALLW	Call Subroutine using WREG	2	0000	0000	0001	0100	None
MOVSF z _s , f _d	Move z _s (source) to 1st word f _d (destination) 2nd word	2	1110	1011	0zzz	zzzz	None
MOVSS z _s , z _d	Move z _s (source) to 1st word z _d (destination) 2nd word	2	1111	ffff	ffff	ffff	None
PUSHL k	Store Literal at FSR2, Decrement FSR2	1	1110	1011	1zzz	zzzz	None
SUBFSR f, k	Subtract Literal from FSR	1	1110	1001	ffkk	kkkk	None
SUBULNK k	Subtract Literal from FSR2 and Return	2	1110	1001	11kk	kkkk	None

PIC18F87J11 FAMILY

25.2.2 EXTENDED INSTRUCTION SET

ADDFSR	Add Literal to FSR				ADDULNK	Add Literal to FSR2 and Return			
Syntax:	ADDFSR f, k				Syntax:	ADDULNK k			
Operands:	0 ≤ k ≤ 63				Operands:	0 ≤ k ≤ 63			
	f ∈ [0, 1, 2]				Operation:	FSR2 + k → FSR2, (TOS) → PC			
Operation:	FSR(f) + k → FSR(f)				Status Affected:	None			
Status Affected:	None				Encoding:	1110 1000 ffkk kkkk			
Encoding:	1110 1000 ffkk kkkk				Description:	The 6-bit literal 'k' is added to the contents of the FSR specified by 'f'.			
Description:	The 6-bit literal 'k' is added to the contents of the FSR specified by 'f'.					The 6-bit literal 'k' is added to the contents of FSR2. A RETURN is then executed by loading the PC with the TOS.			
Words:	1								
Cycles:	1								
Q Cycle Activity:									
	Q1	Q2	Q3	Q4					
	Decode	Read literal 'k'	Process Data	Write to FSR					

Example: ADDFSR 2, 23h

Before Instruction
FSR2 = 03FFh
After Instruction
FSR2 = 0422h

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to FSR
No Operation	No Operation	No Operation	No Operation

Example: ADDULNK 23h

Before Instruction
FSR2 = 03FFh
PC = 0100h
After Instruction
FSR2 = 0422h
PC = (TOS)

Note: All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction format then becomes: {label} instruction argument(s).

CALLW	Subroutine Call using WREG												
Syntax:	CALLW												
Operands:	None												
Operation:	(PC + 2) → TOS, (W) → PCL, (PCLATH) → PCH, (PCLATU) → PCU												
Status Affected:	None												
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0000</td><td>0000</td><td>0001</td><td>0100</td></tr> </table>	0000	0000	0001	0100								
0000	0000	0001	0100										
Description	<p>First, the return address (PC + 2) is pushed onto the return stack. Next, the contents of W are written to PCL; the existing value is discarded. Then, the contents of PCLATH and PCLATU are latched into PCH and PCU, respectively. The second cycle is executed as a NOP instruction while the new next instruction is fetched.</p> <p>Unlike CALL, there is no option to update W, STATUS or BSR.</p>												
Words:	1												
Cycles:	2												
Q Cycle Activity:	<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> </thead> <tbody> <tr> <td>Decode</td><td>Read WREG</td><td>Push PC to stack</td><td>No operation</td></tr> <tr> <td>No operation</td><td>No operation</td><td>No operation</td><td>No operation</td></tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read WREG	Push PC to stack	No operation				
Q1	Q2	Q3	Q4										
Decode	Read WREG	Push PC to stack	No operation										
No operation	No operation	No operation	No operation										

Example: HERE CALLW

Before Instruction

```

PC      = address (HERE)
PCLATH = 10h
PCLATU = 00h
W       = 06h

```

After Instruction

```

PC      = 001006h
TOS    = address (HERE + 2)
PCLATH = 10h
PCLATU = 00h
W       = 06h

```

MOVSF	Move Indexed to f
--------------	--------------------------

Syntax: MOVSF [z_s], f_d

Operands: 0 ≤ z_s ≤ 127
0 ≤ f_d ≤ 4095

Operation: ((FSR2) + z_s) → f_d

Status Affected: None

Encoding:
1st word (source)
2nd word (destin.)

1110	1011	0zzz	zzzz _s
1111	ffff	ffff	fffff _d

Description: The contents of the source register are moved to destination register 'f_d'. The actual address of the source register is determined by adding the 7-bit literal offset 'z_s', in the first word, to the value of FSR2. The address of the destination register is specified by the 12-bit literal 'f_d' in the second word. Both addresses can be anywhere in the 4096-byte data space (000h to FFFh).

The MOVSF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

If the resultant source address points to an Indirect Addressing register, the value returned will be 00h.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Determine source addr	Determine source addr	Read source reg
Decode	No operation No dummy read	No operation	Write register 'f' (dest)

Example: MOVSF [05h], REG2

Before Instruction

```

FSR2      = 80h
Contents of 85h = 33h
REG2      = 11h

```

After Instruction

```

FSR2      = 80h
Contents of 85h = 33h
REG2      = 33h

```

PIC18F87J11 FAMILY

MOVSS	Move Indexed to Indexed	PUSHL	Store Literal at FSR2, Decrement FSR2																
Syntax:	MOVSS [z _s], [z _d]	Syntax:	PUSHL k																
Operands:	0 ≤ z _s ≤ 127 0 ≤ z _d ≤ 127	Operands:	0 ≤ k ≤ 255																
Operation:	((FSR2) + z _s) → ((FSR2) + z _d)	Operation:	k → (FSR2), FSR2 - 1 → FSR2																
Status Affected:	None	Status Affected:	None																
Encoding:		Encoding:																	
1st word (source)	1110 1011 1zzz zzzz _s	1111 1010 kkkk kkkk																	
2nd word (dest.)	1111 xxxx xzzz zzzz _d																		
Description	<p>The contents of the source register are moved to the destination register. The addresses of the source and destination registers are determined by adding the 7-bit literal offsets 'z_s' or 'z_d', respectively, to the value of FSR2. Both registers can be located anywhere in the 4096-byte data memory space (000h to FFFh).</p> <p>The MOVSS instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.</p> <p>If the resultant source address points to an Indirect Addressing register, the value returned will be 00h. If the resultant destination address points to an Indirect Addressing register, the instruction will execute as a NOP.</p>	<p>Words:</p> <p>Cycles:</p> <p>Q Cycle Activity:</p> <table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read 'k'</td> <td>Process data</td> <td>Write to destination</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read 'k'	Process data	Write to destination	<p>The 8-bit literal 'k' is written to the data memory address specified by FSR2. FSR2 is decremented by 1 after the operation.</p> <p>This instruction allows users to push values onto a software stack.</p> <p>Words: 1</p> <p>Cycles: 1</p> <p>Q Cycle Activity:</p> <table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read 'k'</td> <td>Process data</td> <td>Write to destination</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read 'k'	Process data	Write to destination
Q1	Q2	Q3	Q4																
Decode	Read 'k'	Process data	Write to destination																
Q1	Q2	Q3	Q4																
Decode	Read 'k'	Process data	Write to destination																

Example: MOVSS [05h], [06h]

Before Instruction

FSR2	=	80h
Contents of 85h	=	33h
Contents of 86h	=	11h

After Instruction

FSR2	=	80h
Contents of 85h	=	33h
Contents of 86h	=	33h

Example: PUSHL 08h

Before Instruction

FSR2H:FSR2L	=	01ECh
Memory (01ECh)	=	00h

After Instruction

FSR2H:FSR2L	=	01EBh
Memory (01ECh)	=	08h

PIC18F87J11 FAMILY

SUBFSR	Subtract Literal from FSR								
Syntax:	SUBFSR f, k								
Operands:	0 ≤ k ≤ 63 $f \in [0, 1, 2]$								
Operation:	FSRf – k → FSRf								
Status Affected:	None								
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1110</td><td>1001</td><td>ffkk</td><td>kkkk</td></tr></table>	1110	1001	ffkk	kkkk				
1110	1001	ffkk	kkkk						
Description:	The 6-bit literal 'k' is subtracted from the contents of the FSR specified by 'f'.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1" style="width: 100%; text-align: center;"> <tr> <th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr> <tr> <td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write to destination</td></tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write to destination						

Example: SUBFSR 2, 23h

Before Instruction
FSR2 = 03FFh
After Instruction
FSR2 = 03DCh

SUBULNK	Subtract Literal from FSR2 and Return				
Syntax:	SUBULNK k				
Operands:	0 ≤ k ≤ 63				
Operation:	FSR2 – k → FSR2, (TOS) → PC				
Status Affected:	None				
Encoding:	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1110</td><td>1001</td><td>11kk</td><td>kkkk</td></tr></table>	1110	1001	11kk	kkkk
1110	1001	11kk	kkkk		
Description:	The 6-bit literal 'k' is subtracted from the contents of the FSR2. A RETURN is then executed by loading the PC with the TOS.				

The instruction takes two cycles to execute; a NOP is performed during the second cycle.

This may be thought of as a special case of the SUBFSR instruction, where f = 3 (binary '11'); it operates only on FSR2.

Words: 1
Cycles: 2
Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination
No Operation	No Operation	No Operation	No Operation

Example: SUBULNK 23h

Before Instruction
FSR2 = 03FFh
PC = 0100h
After Instruction
FSR2 = 03DCh
PC = (TOS)

PIC18F87J11 FAMILY

25.2.3 BYTE-ORIENTED AND BIT-ORIENTED INSTRUCTIONS IN INDEXED LITERAL OFFSET MODE

Note: Enabling the PIC18 instruction set extension may cause legacy applications to behave erratically or fail entirely.

In addition to eight new commands in the extended set, enabling the extended instruction set also enables Indexed Literal Offset Addressing (**Section 5.6.1 “Indexed Addressing with Literal Offset”**). This has a significant impact on the way that many commands of the standard PIC18 instruction set are interpreted.

When the extended set is disabled, addresses embedded in opcodes are treated as literal memory locations: either as a location in the Access Bank ($a = 0$) or in a GPR bank designated by the BSR ($a = 1$). When the extended instruction set is enabled and $a = 0$, however, a file register argument of 5Fh or less is interpreted as an offset from the pointer value in FSR2 and not as a literal address. For practical purposes, this means that all instructions that use the Access RAM bit as an argument – that is, all byte-oriented and bit-oriented instructions, or almost half of the core PIC18 instructions – may behave differently when the extended instruction set is enabled.

When the content of FSR2 is 00h, the boundaries of the Access RAM are essentially remapped to their original values. This may be useful in creating backward-compatible code. If this technique is used, it may be necessary to save the value of FSR2 and restore it when moving back and forth between C and assembly routines in order to preserve the Stack Pointer. Users must also keep in mind the syntax requirements of the extended instruction set (see **Section 25.2.3.1 “Extended Instruction Syntax with Standard PIC18 Commands”**).

Although the Indexed Literal Offset mode can be very useful for dynamic stack and pointer manipulation, it can also be very annoying if a simple arithmetic operation is carried out on the wrong register. Users who are accustomed to the PIC18 programming must keep in mind that, when the extended instruction set is enabled, register addresses of 5Fh or less are used for Indexed Literal Offset Addressing.

Representative examples of typical byte-oriented and bit-oriented instructions in the Indexed Literal Offset mode are provided on the following page to show how execution is affected. The operand conditions shown in the examples are applicable to all instructions of these types.

25.2.3.1 Extended Instruction Syntax with Standard PIC18 Commands

When the extended instruction set is enabled, the file register argument ‘f’ in the standard byte-oriented and bit-oriented commands is replaced with the literal offset value ‘k’. As already noted, this occurs only when ‘f’ is less than or equal to 5Fh. When an offset value is used, it must be indicated by square brackets (“[]”). As with the extended instructions, the use of brackets indicates to the compiler that the value is to be interpreted as an index or an offset. Omitting the brackets, or using a value greater than 5Fh within the brackets, will generate an error in the MPASM Assembler.

If the index argument is properly bracketed for Indexed Literal Offset Addressing, the Access RAM argument is never specified; it will automatically be assumed to be ‘0’. This is in contrast to standard operation (extended instruction set disabled), when ‘a’ is set on the basis of the target address. Declaring the Access RAM bit in this mode will also generate an error in the MPASM Assembler.

The destination argument ‘d’ functions as before.

In the latest versions of the MPASM Assembler, language support for the extended instruction set must be explicitly invoked. This is done with either the command line option, /y, or the PE directive in the source listing.

25.2.4 CONSIDERATIONS WHEN ENABLING THE EXTENDED INSTRUCTION SET

It is important to note that the extensions to the instruction set may not be beneficial to all users. In particular, users who are not writing code that uses a software stack may not benefit from using the extensions to the instruction set.

Additionally, the Indexed Literal Offset Addressing mode may create issues with legacy applications written to the PIC18 assembler. This is because instructions in the legacy code may attempt to address registers in the Access Bank below 5Fh. Since these addresses are interpreted as literal offsets to FSR2 when the instruction set extension is enabled, the application may read or write to the wrong data addresses.

When porting an application to the PIC18F87J11 Family family, it is very important to consider the type of code. A large, re-entrant application that is written in C and would benefit from efficient compilation will do well when using the instruction set extensions. Legacy applications that heavily use the Access Bank will most likely not benefit from using the extended instruction set.

ADDWF	ADD W to Indexed (Indexed Literal Offset mode)								
Syntax:	ADDWF [k] {,d}								
Operands:	$0 \leq k \leq 95$ $d \in [0,1]$								
Operation:	$(W) + ((FSR2) + k) \rightarrow \text{dest}$								
Status Affected:	N, OV, C, DC, Z								
Encoding:	0010 01d0 kkkk kkkk								
Description:	The contents of W are added to the contents of the register indicated by FSR2, offset by the value 'k'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read 'k'</td> <td>Process Data</td> <td>Write to destination</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read 'k'	Process Data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read 'k'	Process Data	Write to destination						

Example: ADDWF [OFST], 0

Before Instruction	
W	= 17h
OFST	= 2Ch
FSR2	= 0A00h
Contents of 0A2Ch	= 20h
After Instruction	
W	= 37h
Contents of 0A2Ch	= 20h

BSF	Bit Set Indexed (Indexed Literal Offset mode)								
Syntax:	BSF [k], b								
Operands:	$0 \leq f \leq 95$ $0 \leq b \leq 7$								
Operation:	$1 \rightarrow ((FSR2) + k)$								
Status Affected:	None								
Encoding:	1000 bbb0 kkkk kkkk								
Description:	Bit 'b' of the register indicated by FSR2, offset by the value 'k', is set.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process Data</td> <td>Write to destination</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process Data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process Data	Write to destination						

Example: BSF [FLAG_OFST], 7

Before Instruction	
FLAG_OFST	= 0Ah
FSR2	= 0A00h
Contents of 0A0Ah	= 55h
After Instruction	
Contents of 0A0Ah	= D5h

SETF	Set Indexed (Indexed Literal Offset mode)								
Syntax:	SETF [k]								
Operands:	$0 \leq k \leq 95$								
Operation:	$FFh \rightarrow ((FSR2) + k)$								
Status Affected:	None								
Encoding:	0110 1000 kkkk kkkk								
Description:	The contents of the register indicated by FSR2, offset by 'k', are set to FFh.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read 'k'</td> <td>Process Data</td> <td>Write register</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read 'k'	Process Data	Write register
Q1	Q2	Q3	Q4						
Decode	Read 'k'	Process Data	Write register						

Example: SETF [OFST]

Before Instruction	
OFST	= 2Ch
FSR2	= 0A00h
Contents of 0A2Ch	= 00h
After Instruction	
Contents of 0A2Ch	= FFh

PIC18F87J11 FAMILY

25.2.5 SPECIAL CONSIDERATIONS WITH MICROCHIP MPLAB® IDE TOOLS

The latest versions of Microchip's software tools have been designed to fully support the extended instruction set for the PIC18F87J11 Family family. This includes the MPLAB C18 C Compiler, MPASM assembly language and MPLAB Integrated Development Environment (IDE).

When selecting a target device for software development, MPLAB IDE will automatically set default Configuration bits for that device. The default setting for the XINST Configuration bit is '0', disabling the extended instruction set and Indexed Literal Offset Addressing. For proper execution of applications developed to take advantage of the extended instruction set, XINST must be set during programming.

To develop software for the extended instruction set, the user must enable support for the instructions and the Indexed Addressing mode in their language tool(s). Depending on the environment being used, this may be done in several ways:

- A menu option or dialog box within the environment that allows the user to configure the language tool and its settings for the project
- A command line option
- A directive in the source code

These options vary between different compilers, assemblers and development environments. Users are encouraged to review the documentation accompanying their development systems for the appropriate information.

26.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
 - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
 - MPASM™ Assembler
 - MPLAB C18 and MPLAB C30 C Compilers
 - MPLINK™ Object Linker/
MPLIB™ Object Librarian
 - MPLAB ASM30 Assembler/Linker/Library
- Simulators
 - MPLAB SIM Software Simulator
- Emulators
 - MPLAB ICE 2000 In-Circuit Emulator
 - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debugger
 - MPLAB ICD 2
- Device Programmers
 - PICSTART® Plus Development Programmer
 - MPLAB PM3 Device Programmer
 - PICkit™ 2 Development Programmer
- Low-Cost Demonstration and Development Boards and Evaluation Kits

26.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
 - Simulator
 - Programmer (sold separately)
 - Emulator (sold separately)
 - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Visual device initializer for easy register initialization
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as HI-TECH Software C Compilers and IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PIC MCU emulator and simulator tools (automatically updates all project information)
- Debug using:
 - Source files (assembly or C)
 - Mixed assembly and C
 - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

PIC18F87J11 FAMILY

26.2 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for all PIC MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

26.3 MPLAB C18 and MPLAB C30 C Compilers

The MPLAB C18 and MPLAB C30 Code Development Systems are complete ANSI C compilers for Microchip's PIC18 and PIC24 families of microcontrollers and the dsPIC30 and dsPIC33 family of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use not found with other compilers.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

26.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

26.5 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 Assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire dsPIC30F instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

26.6 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C18 and MPLAB C30 C Compilers, and the MPASM and MPLAB ASM30 Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

26.7 MPLAB ICE 2000 High-Performance In-Circuit Emulator

The MPLAB ICE 2000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers. Software control of the MPLAB ICE 2000 In-Circuit Emulator is advanced by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

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

The MPLAB ICE 2000 In-Circuit Emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft® Windows® 32-bit operating system were chosen to best make these features available in a simple, unified application.

26.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC® and MCU devices. It debugs and programs PIC® and dsPIC® Flash microcontrollers with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The MPLAB REAL ICE probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with the popular MPLAB ICD 2 system (RJ11) or with the new high speed, noise tolerant, low-voltage differential signal (LVDS) interconnection (CAT5).

MPLAB REAL ICE is field upgradeable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added, such as software breakpoints and assembly code trace. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, real-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

26.9 MPLAB ICD 2 In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD 2, is a powerful, low-cost, run-time development tool, connecting to the host PC via an RS-232 or high-speed USB interface. This tool is based on the Flash PIC MCUs and can be used to develop for these and other PIC MCUs and dsPIC DSCs. The MPLAB ICD 2 utilizes the in-circuit debugging capability built into the Flash devices. This feature, along with Microchip's In-Circuit Serial Programming™ (ICSP™) protocol, offers cost-effective, in-circuit Flash debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by setting breakpoints, single stepping and watching variables, and CPU status and peripheral registers. Running at full speed enables testing hardware and applications in real time. MPLAB ICD 2 also serves as a development programmer for selected PIC devices.

26.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an SD/MMC card for file storage and secure data applications.

PIC18F87J11 FAMILY

26.11 PICSTART Plus Development Programmer

The PICSTART Plus Development Programmer is an easy-to-use, low-cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. The PICSTART Plus Development Programmer supports most PIC devices in DIP packages up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus Development Programmer is CE compliant.

26.12 PICkit 2 Development Programmer

The PICkit™ 2 Development Programmer is a low-cost programmer and selected Flash device debugger with an easy-to-use interface for programming many of Microchip's baseline, mid-range and PIC18F families of Flash memory microcontrollers. The PICkit 2 Starter Kit includes a prototyping development board, twelve sequential lessons, software and HI-TECH's PICC™ Lite C compiler, and is designed to help get up to speed quickly using PIC® microcontrollers. The kit provides everything needed to program, evaluate and develop applications using Microchip's powerful, mid-range Flash memory family of microcontrollers.

26.13 Demonstration, Development and Evaluation Boards

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart® battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Check the Microchip web page (www.microchip.com) and the latest "Product Selector Guide" (DS00148) for the complete list of demonstration, development and evaluation kits.

27.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings^(†)

Ambient temperature under bias.....	-40°C to +100°C
Storage temperature	-65°C to +150°C
Voltage on any digital only input pin or <u>MCLR</u> with respect to Vss (except VDD)	-0.3V to 6.0V
Voltage on any combined digital and analog pin with respect to Vss	-0.3V to (VDD + 0.3V)
Voltage on VDDCORE with respect to Vss.....	-0.3V to 2.75V
Voltage on VDD with respect to Vss	-0.3V to 4.0V
Total power dissipation (Note 1)	1.0W
Maximum current out of Vss pin	300 mA
Maximum current into VDD pin	250 mA
Input clamp current, I _{IK} (V _I < 0 or V _I > V _{DD}) (Note 2).....	±0 mA
Output clamp current, I _{OK} (V _O < 0 or V _O > V _{DD}) (Note 2).....	±0 mA
Maximum output current sunk by any PORTB and PORTC I/O pins.....	25 mA
Maximum output current sunk by any PORTD, PORTE and PORTJ I/O pins	8 mA
Maximum output current sunk by any PORTA, PORTF, PORTG and PORTH I/O pins.....	2 mA
Maximum output current sourced by any PORTB and PORTC I/O pins.....	25 mA
Maximum output current sourced by any PORTD, PORTE and PORTJ I/O pins	8 mA
Maximum output current sourced by any PORTA, PORTF, PORTG and PORTH I/O pins	2 mA
Maximum current sunk by all ports combined.....	200 mA
Maximum current sourced by all ports combined.....	200 mA

Note 1: Power dissipation is calculated as follows:

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

2: No clamping diodes are present.

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

PIC18F87J11 FAMILY

FIGURE 27-1: PIC18F87J11 FAMILY VOLTAGE-FREQUENCY GRAPH, REGULATOR ENABLED (INDUSTRIAL)

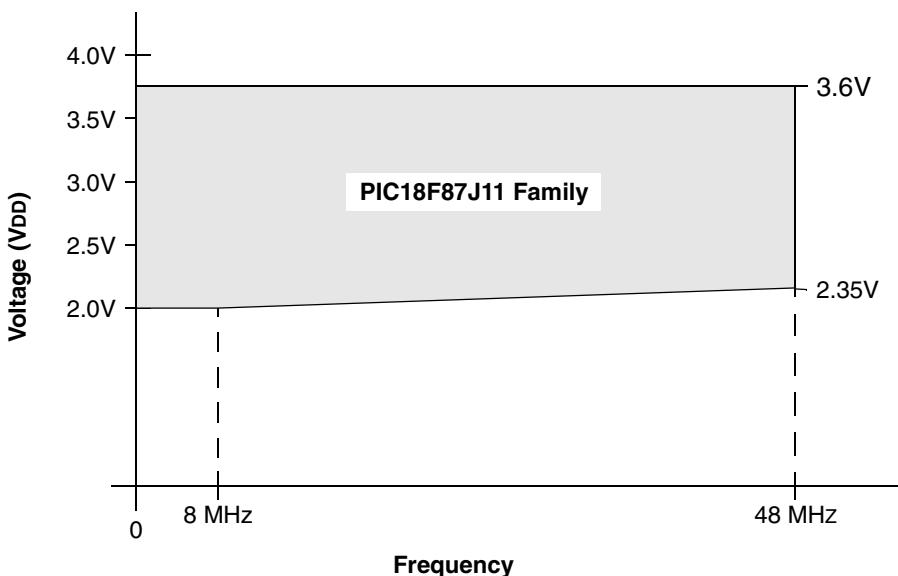
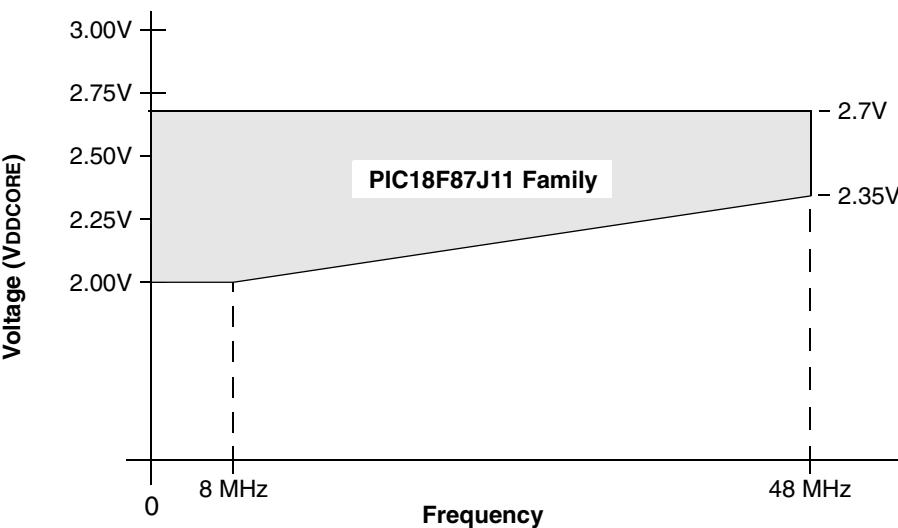


FIGURE 27-2: PIC18F87J11 FAMILY VOLTAGE-FREQUENCY GRAPH, REGULATOR DISABLED (INDUSTRIAL)⁽¹⁾



Note 1: When the on-chip voltage regulator is disabled, V_{DD} and V_{DDCORE} must be maintained so that V_{DDCORE} \leq V_{DD} \leq 3.6V.

27.1 DC Characteristics: Supply Voltage PIC18F87J11 Family (Industrial)

PIC18F87J11 Family Family (Industrial)			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
D001	VDD	Supply Voltage	VDDCORE 2.0	—	3.6 3.6	V V	ENVREG tied to VSS ENVREG tied to VDD
D001B	VDDCORE	External Supply for Microcontroller Core	2.0	—	2.7	V	ENVREG tied to Vss
D001C	AVDD	Analog Supply Voltage	VDD – 0.3	—	VDD + 0.3	V	
D001D	AVSS	Analog Ground Potential	VSS – 0.3	—	VSS + 0.3	V	
D002	VDR	RAM Data Retention Voltage⁽¹⁾	1.5	—	—	V	
D003	VPOR	VDD Power-on Reset Voltage	—	—	0.7	V	See Section 4.3 “Power-on Reset (POR)” for details
D004	SVDD	VDD Rise Rate to ensure internal Power-on Reset signal	0.05	—	—	V/ms	See Section 4.3 “Power-on Reset (POR)” for details
D005	VBOR	Brown-out Reset Voltage	—	1.8	—	V	

Note 1: This is the limit to which VDD can be lowered in Sleep mode, or during a device Reset, without losing RAM data.

PIC18F87J11 FAMILY

27.2 DC Characteristics: Power-Down and Supply Current PIC18F87J11 Family (Industrial)

PIC18F87J11 Family (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
Param No.	Device	Typ	Max	Units	Conditions	
Power-Down Current (IPD)⁽¹⁾						
All devices	0.5	1.4	μA	-40°C	VDD = 2.0V ⁽⁴⁾ (Sleep mode)	
	0.5	1.4	μA	+25°C		
	5.5	10.2	μA	+85°C		
All devices	0.6	1.5	μA	-40°C	VDD = 2.5V ⁽⁴⁾ (Sleep mode)	
	0.6	1.5	μA	+25°C		
	6.8	12.6	μA	+85°C		
All devices	2.9	7	μA	-40°C	VDD = 3.3V ⁽⁵⁾ (Sleep mode)	
	3.6	7	μA	+25°C		
	9.6	19	μA	+85°C		

Legend: TBD = To Be Determined

- Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or Vss and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).
- 2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.

The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
MCLR = VDD; WDT enabled/disabled as specified.

- 3:** Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.
- 4:** Voltage regulator disabled (ENVREG = 0, tied to Vss).
- 5:** Voltage regulator enabled (ENVREG = 1, tied to VDD, REGSLP = 1).

27.2 DC Characteristics: Power-Down and Supply Current PIC18F87J11 Family (Industrial) (Continued)

PIC18F87J11 Family (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial						
Param No.	Device	Typ	Max	Units	Conditions			
Supply Current (IDD) ^(2,3)	All devices						FOSC = 31 kHz (RC_RUN mode, internal oscillator source)	
	All devices	5	14.2	µA	-40°C	VDD = 2.0V, VDDCORE = 2.0V ⁽⁴⁾		
		5.5	14.2	µA	+25°C			
		10	19.0	µA	+85°C			
	All devices	6.8	16.5	µA	-40°C	VDD = 2.5V, VDDCORE = 2.5V ⁽⁴⁾		
		7.6	16.5	µA	+25°C			
		14	22.4	µA	+85°C			
	All devices	37	84	µA	-40°C	VDD = 3.3V ⁽⁵⁾		
		51	84	µA	+25°C			
		72	108	µA	+85°C			
	All devices	0.43	0.82	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V ⁽⁴⁾	FOSC = 1 MHz (RC_RUN mode, internal oscillator source)	
		0.47	0.82	mA	+25°C			
		0.52	0.95	mA	+85°C			
	All devices	0.52	0.98	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V ⁽⁴⁾		
		0.57	0.98	mA	+25°C			
		0.63	1.10	mA	+85°C			
	All devices	0.59	0.96	mA	-40°C	VDD = 3.3V ⁽⁵⁾		
		0.65	0.96	mA	+25°C			
		0.72	1.18	mA	+85°C			
	All devices	0.88	1.45	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V ⁽⁴⁾	FOSC = 4 MHz (RC_RUN mode, internal oscillator source)	
		1	1.45	mA	+25°C			
		1.1	1.58	mA	+85°C			
	All devices	1.2	1.72	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V ⁽⁴⁾		
		1.3	1.72	mA	+25°C			
		1.4	1.85	mA	+85°C			
	All devices	1.3	1.87	mA	-40°C	VDD = 3.3V ⁽⁵⁾		
		1.4	1.87	mA	+25°C			
		1.5	1.96	mA	+85°C			

Legend: TBD = To Be Determined

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.

The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
MCLR = VDD; WDT enabled/disabled as specified.

3: Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

4: Voltage regulator disabled (ENVREG = 0, tied to Vss).

5: Voltage regulator enabled (ENVREG = 1, tied to VDD, REGSLP = 1).

PIC18F87J11 FAMILY

27.2 DC Characteristics: Power-Down and Supply Current PIC18F87J11 Family (Industrial) (Continued)

PIC18F87J11 Family (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
Param No.	Device	Typ	Max	Units	Conditions	
Supply Current (IDD) Cont.^(2,3)						
All devices	3	9.4	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V ⁽⁴⁾	FOSC = 31 kHz (RC_IDLE mode, internal oscillator source)
	3.3	9.4	μA	+25°C		
	8.5	17.2	μA	+85°C		
All devices	4	10.5	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V ⁽⁴⁾	FOSC = 1 MHz (RC_IDLE mode, internal oscillator source)
	4.3	10.5	μA	+25°C		
	10.3	19.5	μA	+85°C		
All devices	34	82	μA	-40°C	VDD = 3.3V ⁽⁵⁾	FOSC = 4 MHz (RC_IDLE mode, internal oscillator source)
	48	82	μA	+25°C		
	69	105	μA	+85°C		
All devices	0.33	0.75	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V ⁽⁴⁾	FOSC = 1 MHz (RC_IDLE mode, internal oscillator source)
	0.37	0.75	mA	+25°C		
	0.41	0.84	mA	+85°C		
All devices	0.39	0.78	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V ⁽⁴⁾	FOSC = 1 MHz (RC_IDLE mode, internal oscillator source)
	0.42	0.78	mA	+25°C		
	0.47	0.91	mA	+85°C		
All devices	0.43	0.82	mA	-40°C	VDD = 3.3V ⁽⁵⁾	FOSC = 4 MHz (RC_IDLE mode, internal oscillator source)
	0.48	0.82	mA	+25°C		
	0.54	0.95	mA	+85°C		
All devices	0.53	0.98	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V ⁽⁴⁾	FOSC = 4 MHz (RC_IDLE mode, internal oscillator source)
	0.57	0.98	mA	+25°C		
	0.61	1.12	mA	+85°C		
All devices	0.63	1.14	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V ⁽⁴⁾	FOSC = 4 MHz (RC_IDLE mode, internal oscillator source)
	0.67	1.14	mA	+25°C		
	0.72	1.25	mA	+85°C		
All devices	0.7	1.27	mA	-40°C	VDD = 3.3V ⁽⁵⁾	FOSC = 4 MHz (RC_IDLE mode, internal oscillator source)
	0.76	1.27	mA	+25°C		
	0.82	1.45	mA	+85°C		

Legend: TBD = To Be Determined

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.

The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
MCLR = VDD; WDT enabled/disabled as specified.

3: Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

4: Voltage regulator disabled (ENVREG = 0, tied to VSS).

5: Voltage regulator enabled (ENVREG = 1, tied to VDD, REGSLP = 1).

27.2 DC Characteristics: Power-Down and Supply Current PIC18F87J11 Family (Industrial) (Continued)

PIC18F87J11 Family (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	Device	Typ	Max	Units	Conditions		
Supply Current (IDD) Cont. ^(2,3)							VDD = 2.0V, VDDCORE = 2.0V ⁽⁴⁾ VDD = 2.5V, VDDCORE = 2.5V ⁽⁴⁾ VDD = 3.3V ⁽⁵⁾
All devices		0.17	0.35	mA	-40°C		
		0.18	0.35	mA	+25°C		
		0.20	0.42	mA	+85°C		
All devices		0.29	0.52	mA	-40°C		
		0.31	0.52	mA	+25°C		
		0.34	0.61	mA	+85°C		
All devices		0.59	1.1	mA	-40°C		
		0.44	0.85	mA	+25°C		
		0.42	0.85	mA	+85°C		
All devices		0.70	1.25	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V ⁽⁴⁾	
		0.75	1.25	mA	+25°C		
		0.79	1.36	mA	+85°C		
All devices		1.10	1.7	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V ⁽⁴⁾	
		1.10	1.7	mA	+25°C		
		1.12	1.82	mA	+85°C		
All devices		1.55	1.95	mA	-40°C	VDD = 3.3V ⁽⁵⁾	
		1.47	1.89	mA	+25°C		
		1.54	1.92	mA	+85°C		
All devices		9.9	14.8	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V ⁽⁴⁾	
		9.5	14.8	mA	+25°C		
		10.1	15.2	mA	+85°C		
All devices		13.3	19.2	mA	-40°C	VDD = 3.3V ⁽⁵⁾	
		12.2	18.7	mA	+25°C		
		12.1	18.7	mA	+85°C		

Legend: TBD = To Be Determined

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.

The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
MCLR = VDD; WDT enabled/disabled as specified.

3: Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

4: Voltage regulator disabled (ENVREG = 0, tied to VSS).

5: Voltage regulator enabled (ENVREG = 1, tied to VDD, REGSLP = 1).

PIC18F87J11 FAMILY

27.2 DC Characteristics: Power-Down and Supply Current PIC18F87J11 Family (Industrial) (Continued)

PIC18F87J11 Family (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial							
Param No.	Device	Typ	Max	Units	Conditions				
Supply Current (IDD) Cont.^(2,3)									
All devices	2.8	3.8	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V ⁽⁴⁾	FOSC = 4 MHz. 16 MHz internal (PRI_RUN HSPLL mode)			
	3.0	3.8	mA	+25°C					
	3.2	4.1	mA	+85°C					
All devices	4.5	5.2	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V ⁽⁴⁾				
	4.4	5.2	mA	+25°C					
	4.5	5.2	mA	+85°C					
All devices	5.7	6.7	mA	-40°C	VDD = 3.3V ⁽⁵⁾				
	5.5	6.3	mA	+25°C					
	5.3	6.3	mA	+85°C					
All devices	10.8	13.5	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V ⁽⁴⁾	FOSC = 12 MHz, 48 MHz internal (PRI_RUN HSPLL mode)			
	10.8	13.5	mA	+25°C					
	9.9	13.0	mA	+85°C					
All devices	13.4	17.1	mA	-40°C	VDD = 3.3V ⁽⁵⁾				
	12.3	16.2	mA	+25°C					
	11.2	15.5	mA	+85°C					

Legend: TBD = To Be Determined

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or Vss and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.

The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
MCLR = VDD; WDT enabled/disabled as specified.

3: Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

4: Voltage regulator disabled (ENVREG = 0, tied to Vss).

5: Voltage regulator enabled (ENVREG = 1, tied to VDD, REGSLP = 1).

27.2 DC Characteristics: Power-Down and Supply Current PIC18F87J11 Family (Industrial) (Continued)

PIC18F87J11 Family (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	Device	Typ	Max	Units	Conditions		
Supply Current (IDD) Cont. ^(2,3)							VDD = 2.0V, VDDCORE = 2.0V ⁽⁴⁾ VDD = 2.5V, VDDCORE = 2.5V ⁽⁴⁾ VDD = 3.3V ⁽⁵⁾
All devices		0.10	0.26	mA	-40°C		
		0.07	0.18	mA	+25°C		
		0.09	0.22	mA	+85°C		
All devices		0.25	0.48	mA	-40°C		
		0.13	0.30	mA	+25°C		
		0.10	0.26	mA	+85°C		
All devices		0.45	0.68	mA	-40°C		
		0.26	0.45	mA	+25°C		
		0.30	0.54	mA	+85°C		
All devices		0.36	0.60	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V ⁽⁴⁾	
		0.33	0.56	mA	+25°C		
		0.35	0.56	mA	+85°C		
All devices		0.52	0.81	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V ⁽⁴⁾	
		0.45	0.70	mA	+25°C		
		0.46	0.70	mA	+85°C		
All devices		0.80	1.15	mA	-40°C	VDD = 3.3V ⁽⁵⁾	
		0.66	0.98	mA	+25°C		
		0.65	0.98	mA	+85°C		
All devices		5.2	6.5	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V ⁽⁴⁾	
		4.9	5.9	mA	+25°C		
		3.4	4.5	mA	+85°C		
All devices		6.2	8.4	mA	-40°C	VDD = 3.3V ⁽⁵⁾	
		5.9	7.5	mA	+25°C		
		5.8	7.5	mA	+85°C		

Legend: TBD = To Be Determined

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.

The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
MCLR = VDD; WDT enabled/disabled as specified.

3: Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

4: Voltage regulator disabled (ENVREG = 0, tied to VSS).

5: Voltage regulator enabled (ENVREG = 1, tied to VDD, REGSLP = 1).

PIC18F87J11 FAMILY

27.2 DC Characteristics: Power-Down and Supply Current PIC18F87J11 Family (Industrial) (Continued)

PIC18F87J11 Family (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	Device	Typ	Max	Units	Conditions		
Supply Current (IDD) Cont.^(2,3)							
All devices	18	35	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V ⁽⁴⁾	FOSC = 32 kHz ⁽³⁾ (SEC_RUN mode, Timer1 as clock)	
	19	35	μA	+25°C			
	28	49	μA	+85°C			
All devices	20	45	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V ⁽⁴⁾		
	21	45	μA	+25°C			
	32	61	μA	+85°C			
All devices	0.06	0.11	mA	-40°C	VDD = 3.3V ⁽⁵⁾	FOSC = 32 kHz ⁽³⁾ (SEC_IDLE mode, Timer1 as clock)	
	0.07	0.11	mA	+25°C			
	0.09	0.15	mA	+85°C			
All devices	14	28	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V ⁽⁴⁾		
	15	28	μA	+25°C			
	24	43	μA	+85°C			
All devices	15	31	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V ⁽⁴⁾		
	16	31	μA	+25°C			
	27	50	μA	+85°C			
All devices	0.05	0.10	mA	-40°C	VDD = 3.3V ⁽⁵⁾		
	0.06	0.10	mA	+25°C			
	0.08	0.14	mA	+85°C			

Legend: TBD = To Be Determined

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.

The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
MCLR = VDD; WDT enabled/disabled as specified.

3: Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

4: Voltage regulator disabled (ENVREG = 0, tied to VSS).

5: Voltage regulator enabled (ENVREG = 1, tied to VDD, REGSLP = 1).

27.2 DC Characteristics: Power-Down and Supply Current PIC18F87J11 Family (Industrial) (Continued)

PIC18F87J11 Family (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq \text{TA} \leq +85^{\circ}\text{C}$ for industrial					
Param No.	Device	Typ	Max	Units	Conditions		
D022	Module Differential Currents (ΔIWDT, ΔIOSCB, ΔIAD)						
	Watchdog Timer	2.1	7.0	μA	-40°C	$\text{VDD} = 2.0\text{V}$, $\text{VDDCORE} = 2.0\text{V}^{(4)}$	
		2.2	7.0	μA	$+25^{\circ}\text{C}$		
		4.3	9.5	μA	$+85^{\circ}\text{C}$		
		3.0	8.0	μA	-40°C		
		3.1	8.0	μA	$+25^{\circ}\text{C}$	$\text{VDD} = 2.5\text{V}$, $\text{VDDCORE} = 2.5\text{V}^{(4)}$	
		5.5	10.4	μA	$+85^{\circ}\text{C}$		
		5.9	12.1	μA	-40°C		
		6.2	12.1	μA	$+25^{\circ}\text{C}$	$\text{VDD} = 3.3\text{V}$	
		6.9	13.6	μA	$+85^{\circ}\text{C}$		
D025 (ΔIOSCB)	Timer1 Oscillator	14	24	μA	-40°C	$\text{VDD} = 2.0\text{V}$, $\text{VDDCORE} = 2.0\text{V}^{(4)}$	32 kHz on Timer1 ⁽³⁾
		15	24	μA	$+25^{\circ}\text{C}$		
		23	36	μA	$+85^{\circ}\text{C}$		
		17	26	μA	-40°C	$\text{VDD} = 2.5\text{V}$, $\text{VDDCORE} = 2.5\text{V}^{(4)}$	32 kHz on Timer1 ⁽³⁾
		18	26	μA	$+25^{\circ}\text{C}$		
		25	38	μA	$+85^{\circ}\text{C}$		
		19	35	μA	-40°C	$\text{VDD} = 3.3\text{V}$	32 kHz on Timer1 ⁽³⁾
		21	35	μA	$+25^{\circ}\text{C}$		
		28	44	μA	$+85^{\circ}\text{C}$		
D026 (ΔIAD)	A/D Converter	3.0	10.0	μA	-40°C to $+85^{\circ}\text{C}$	$\text{VDD} = 2.0\text{V}$, $\text{VDDCORE} = 2.0\text{V}^{(4)}$	A/D on, not converting
		3.0	10.0	μA	-40°C to $+85^{\circ}\text{C}$		
		3.2	11.0	μA	-40°C to $+85^{\circ}\text{C}$	$\text{VDD} = 3.3\text{V}$	

Legend: TBD = To Be Determined

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or Vss and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.

The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
MCLR = VDD; WDT enabled/disabled as specified.

3: Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to $+70^{\circ}\text{C}$. Extended temperature crystals are available at a much higher cost.

4: Voltage regulator disabled (ENVREG = 0, tied to Vss).

5: Voltage regulator enabled (ENVREG = 1, tied to VDD, REGSLP = 1).

PIC18F87J11 FAMILY

27.3 DC Characteristics: PIC18F87J11 Family (Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial			
Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
D030 D031 D032 D033 D033A D034	V _{IL}	Input Low Voltage All I/O ports: with TTL buffer with Schmitt Trigger buffer	V _{SS}	0.15 V _{DD}	V	
		MCLR	V _{SS}	0.2 V _{DD}	V	
		OSC1	V _{SS}	0.3 V _{DD}	V	HS, HSPLL modes
		OSC1	V _{SS}	0.2 V _{DD}	V	EC, ECPLL modes
		T13CKI	V _{SS}	0.3	V	
D040 D041 D042 D043 D043A D044	V _{IH}	Input High Voltage I/O ports with analog functions: with TTL buffer with Schmitt Trigger buffer	0.25 V _{DD} + 0.8V	V _{DD}	V	V _{DD} < 3.3V
		Digital-only I/O ports: with TTL buffer	0.8 V _{DD}	V _{DD}	V	
		with Schmitt Trigger buffer	0.25 V _{DD} + 0.8V 2.0	5.5 5.5	V	V _{DD} < 3.3V 3.3V ≤ V _{DD} ≤ 3.6V
		MCLR	0.8 V _{DD}	V _{DD}	V	
		OSC1	0.7 V _{DD}	V _{DD}	V	HS, HSPLL modes
		OSC1	0.8 V _{DD}	V _{DD}	V	EC, ECPLL modes
		T13CKI	1.6	V _{DD}	V	
		Input Leakage Current^(1,2) I/O ports	—	±1	µA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , Pin at high-impedance
		MCLR	—	±1	µA	V _{SS} ≤ V _{PIN} ≤ V _{DD}
		OSC1	—	±5	µA	V _{SS} ≤ V _{PIN} ≤ V _{DD}
D070	I _{PU} IPURB	Weak Pull-up Current PORTB weak pull-up current	30	240	µA	V _{DD} = 3.3V, V _{PIN} = V _{SS}

Note 1: Negative current is defined as current sourced by the pin.

PIC18F87J11 FAMILY

27.3 DC Characteristics: PIC18F87J11 Family (Industrial) (Continued)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq \text{TA} \leq +85^{\circ}\text{C}$ for industrial			
Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
D080	VOL	Output Low Voltage I/O ports: PORTA, PORTF, PORTG, PORTH	—	0.4	V	I _{OL} = 2 mA, V _{DD} = 3.3V, -40°C to $+85^{\circ}\text{C}$
		PORTD, PORTE, PORTJ	—	0.4	V	I _{OL} = 3.4 mA, V _{DD} = 3.3V, -40°C to $+85^{\circ}\text{C}$
		PORTB, PORTC	—	0.4	V	I _{OL} = 3.4 mA, V _{DD} = 3.3V, -40°C to $+85^{\circ}\text{C}$
D083		OSC2/CLKO (EC, ECPLL modes)	—	0.4	V	I _{OL} = 1.6 mA, V _{DD} = 3.3V, -40°C to $+85^{\circ}\text{C}$
D090	VOH	Output High Voltage ⁽¹⁾ I/O ports: PORTA, PORTF, PORTG, PORTH	2.4	—	V	I _{OH} = -2 mA, V _{DD} = 3.3V, -40°C to $+85^{\circ}\text{C}$
		PORTD, PORTE, PORTJ	2.4	—	V	I _{OH} = -2 mA, V _{DD} = 3.3V, -40°C to $+85^{\circ}\text{C}$
		PORTB, PORTC	2.4	—	V	I _{OH} = -2 mA, V _{DD} = 3.3V, -40°C to $+85^{\circ}\text{C}$
D092		OSC2/CLKO (INTOSC, EC, ECPLL modes)	2.4	—	V	I _{OH} = -1 mA, V _{DD} = 3.3V, -40°C to $+85^{\circ}\text{C}$
(4)	COSC2	Capacitive Loading Specs on Output Pins	—	15	pF	In HS mode when external clock is used to drive OSC1
		OSC2 pin	—	50	pF	To meet the AC Timing Specifications
		All I/O pins and OSC2	—	400	pF	I ² C™ Specification
D102	CB	SCLx, SDAx	—			

Note 1: Negative current is defined as current sourced by the pin.

PIC18F87J11 FAMILY

TABLE 27-1: MEMORY PROGRAMMING REQUIREMENTS

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq \text{TA} \leq +85^{\circ}\text{C}$ for industrial				
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
Program Flash Memory							
D130	EP	Cell Endurance	1K	10K	—	E/W	-40°C to $+85^{\circ}\text{C}$
D131	VPR	VDD for Read	V _{MIN}	—	3.6	V	V _{MIN} = Minimum operating voltage
D132B	VPEW	VDD for Self-Timed Write	V _{MIN}	—	3.6	V	V _{MIN} = Minimum operating voltage
D133A	TIW	Self-Timed Write Cycle Time	—	2.8	—	ms	
D134	TRETD	Characteristic Retention	20	—	—	Year	Provided no other specifications are violated
D135	IDDP	Supply Current during Programming	—	10	14	mA	
D1xxx	TWE	Writes per Erase Cycle	—	—	1		

† Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

PIC18F87J11 FAMILY

TABLE 27-2: COMPARATOR SPECIFICATIONS

Operating Conditions: $3.0V < VDD < 3.6V$, $-40^{\circ}C < TA < +85^{\circ}C$ (unless otherwise stated)							
Param No.	Sym	Characteristics	Min	Typ	Max	Units	Comments
D300	VIOFF	Input Offset Voltage	—	± 5.0	± 10	mV	
D301	VICM	Input Common Mode Voltage*	0	—	$AVDD - 1.5$	V	
	VIRV	Internal Reference Voltage	—	$\pm 1.2^{(2)}$	—	V	$\pm 1.2\%$
D302	CMRR	Common Mode Rejection Ratio*	55	—	—	dB	
300	TRESP	Response Time ^{(1)*}	—	150	400	ns	
301	TMC2OV	Comparator Mode Change to Output Valid*	—	—	10	μs	

* These parameters are characterized but not tested.

Note 1: Response time measured with one comparator input at $(VDD - 1.5)/2$, while the other input transitions from Vss to VDD.

2: Tolerance is $\pm 1.2\%$.

TABLE 27-3: VOLTAGE REFERENCE SPECIFICATIONS

Operating Conditions: $3.0V < VDD < 3.6V$, $-40^{\circ}C < TA < +85^{\circ}C$ (unless otherwise stated)							
Param No.	Sym	Characteristics	Min	Typ	Max	Units	Comments
D310	VRES	Resolution	$VDD/24$	—	$VDD/32$	LSb	
D311	VRAA	Absolute Accuracy	—	—	$1/2$	LSb	
D312	VRUR	Unit Resistor Value (R)	—	2k	—	Ω	
310	TSET	Settling Time ⁽¹⁾	—	—	10	μs	

Note 1: Settling time measured while CVRR = 1 and the CVR3:CVR0 bits transition from '0000' to '1111'.

TABLE 27-4: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

Operating Conditions: $-40^{\circ}C < TA < +85^{\circ}C$ (unless otherwise stated)							
Param No.	Sym	Characteristics	Min	Typ	Max	Units	Comments
	VRGOUT	Regulator Output Voltage*	—	2.5	—	V	
	CF	External Filter Capacitor Value*	4.7	10	—	μF	Capacitor must be low-ESR

* These parameters are characterized but not tested. Parameter numbers not yet assigned for these specifications.

PIC18F87J11 FAMILY

27.4 AC (Timing) Characteristics

27.4.1 TIMING PARAMETER SYMOLOGY

The timing parameter symbols have been created following one of the following formats:

- | | | |
|-------------|-----------|--|
| 1. TppS2ppS | 3. Tcc:st | (I ² C specifications only) |
| 2. TppS | 4. Ts | (I ² C specifications only) |

T	F	Frequency	T	Time
---	---	-----------	---	------

Lowercase letters (pp) and their meanings:

pp			
cc	CCP1	osc	OSC1
ck	CLKO	rd	<u>RD</u>
cs	<u>CS</u>	rw	<u>RD</u> or <u>WR</u>
di	SDI	sc	SCK
do	SDO	ss	<u>SS</u>
dt	Data in	t0	T0CKI
io	I/O port	t1	T13CKI
mc	MCLR	wr	<u>WR</u>

Uppercase letters and their meanings:

S		P	Period
F	Fall	R	Rise
H	High	V	Valid
I	Invalid (High-impedance)	Z	High-impedance
L	Low	High	High
I ² C only		Low	Low
AA	output access		
BUF	Bus free		

Tcc:ST (I²C specifications only)

CC		SU	Setup
HD	Hold		
ST		STO	Stop condition
DAT	DATA input hold		
STA	Start condition		

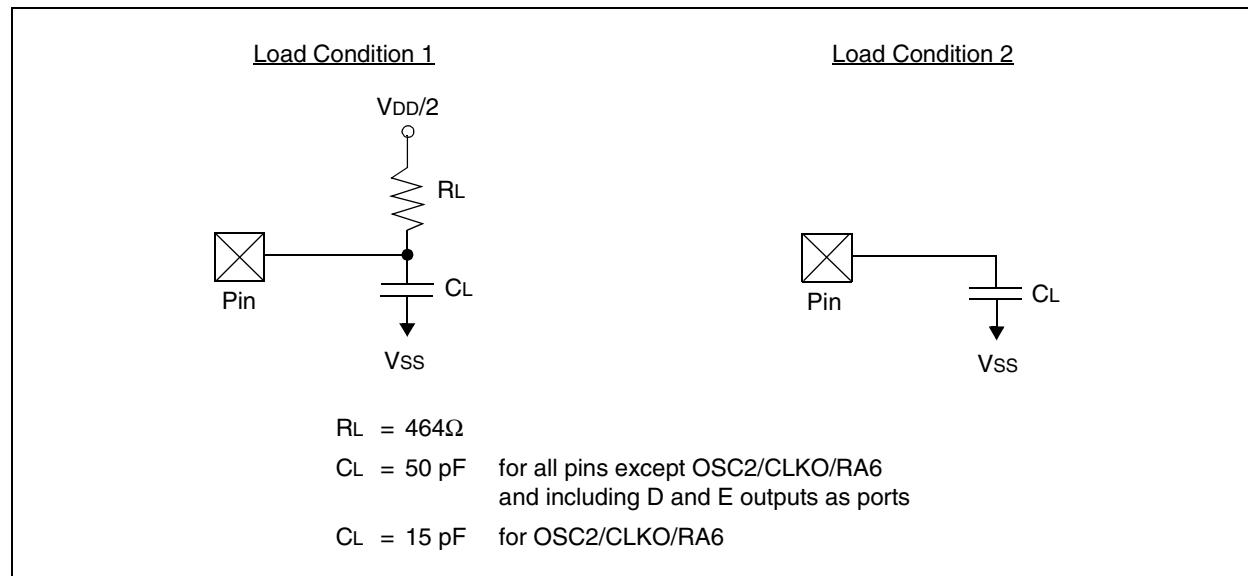
27.4.2 TIMING CONDITIONS

The temperature and voltages specified in Table 27-5 apply to all timing specifications unless otherwise noted. Figure 27-3 specifies the load conditions for the timing specifications.

TABLE 27-5: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

AC CHARACTERISTICS	Standard Operating Conditions (unless otherwise stated)
	Operating temperature $-40^{\circ}\text{C} \leq \text{TA} \leq +85^{\circ}\text{C}$ for industrial
	Operating voltage VDD range as described in Section 27.1 and Section 27.3 .

FIGURE 27-3: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS



PIC18F87J11 FAMILY

27.4.3 TIMING DIAGRAMS AND SPECIFICATIONS

FIGURE 27-4: EXTERNAL CLOCK TIMING

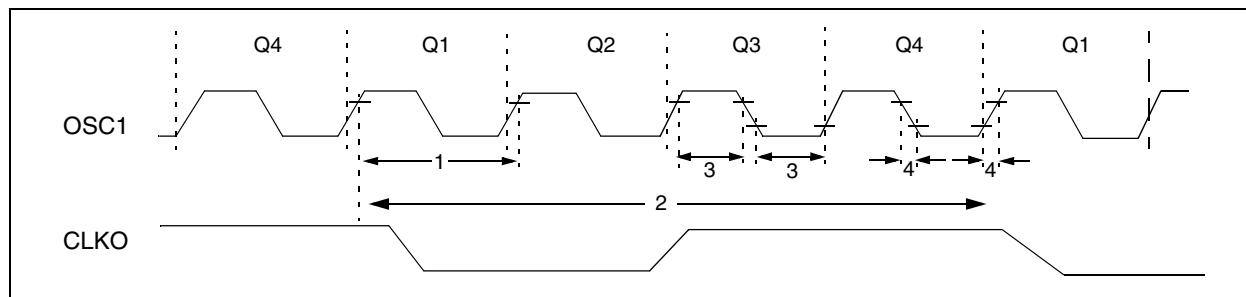


TABLE 27-6: EXTERNAL CLOCK TIMING REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
1A	Fosc	External CLKI Frequency ⁽¹⁾	DC	48	MHz	EC Oscillator mode
		Oscillator Frequency ⁽¹⁾	DC	12	MHz	ECPLL Oscillator mode
	Tosc	External CLKI Period ⁽¹⁾	4	25	MHz	HS Oscillator mode
		Oscillator Period ⁽¹⁾	4	12	MHz	HSPLL Oscillator mode
1	Tosc	External CLKI Period ⁽¹⁾	20.8	—	ns	EC Oscillator mode
		Oscillator Period ⁽¹⁾	83.3	—	ns	ECPLL Oscillator mode
2	Tcy	Instruction Cycle Time ⁽¹⁾	40.0	250	ns	HS Oscillator mode
3	TosL, TosH	External Clock in (OSC1) High or Low Time	83.3	250	ns	HSPLL Oscillator mode
4	TosR, TosF	External Clock in (OSC1) Rise or Fall Time	—	7.5	ns	TCY = 4/Fosc, Industrial

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period for all configurations except PLL. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1/CLKI pin. When an external clock input is used, the "max." cycle time limit is "DC" (no clock) for all devices.

PIC18F87J11 FAMILY

TABLE 27-7: PLL CLOCK TIMING SPECIFICATIONS (VDD = 2.15V TO 3.6V)

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
F10	Fosc	Oscillator Frequency Range	4	—	12	MHz	
F11	FSYS	On-Chip VCO System Frequency	16	—	48	MHz	
F12	t _{rc}	PLL Start-up Time (lock time)	—	—	2	ms	
F13	ΔCLK	CLKO Stability (jitter)	-2	—	+2	%	

† Data in "Typ" column is at 3.3V, 25°C, unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 27-8: INTERNAL RC ACCURACY (INTOSC AND INTRC SOURCES)

Param No.	Device	Min	Typ	Max	Units	Conditions	
INTOSC Accuracy @ Freq = 8 MHz, 4 MHz, 2 MHz, 1 MHz, 500 kHz, 250 kHz, 125 kHz, 31 kHz ⁽¹⁾							
All Devices	-2	+/-1	2	%	+25°C	VDD = 2.7-3.3V	
	-5	—	5	%	-10°C to +85°C	VDD = 2.0-3.3V	
	-10	+/-1	10	%	-40°C to +85°C	VDD = 2.0-3.3V	
INTRC Accuracy @ Freq = 31 kHz ⁽¹⁾							
All Devices	21.7	—	40.3	kHz			

Note 1: The accuracy specification of the 31 kHz clock is determined by which source is providing it at a given time. When INTSRC (OSCTUNE<7>) is '1', use the INTOSC accuracy specification. When INTSRC is '0', use the INTRC accuracy specification.

PIC18F87J11 FAMILY

FIGURE 27-5: CLKO AND I/O TIMING

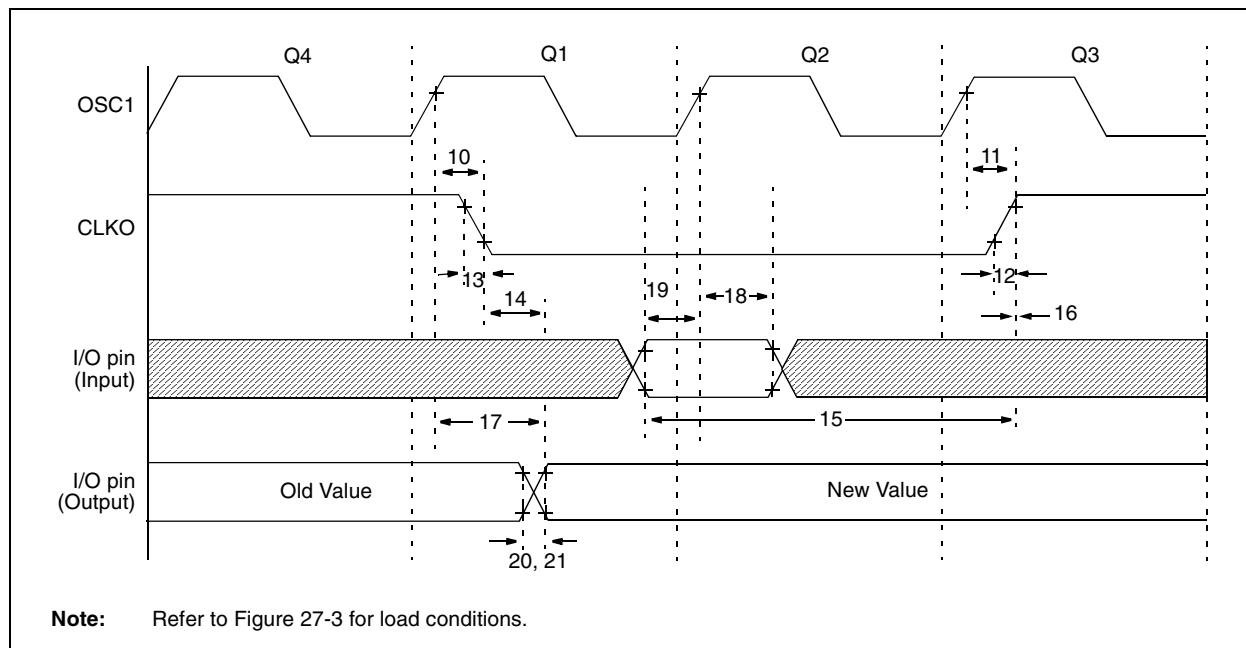


TABLE 27-9: CLKO AND I/O TIMING REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
10	TosH2ckL	OSC1 \uparrow to CLKO \downarrow	—	75	200	ns	(Note 1)
11	TosH2ckH	OSC1 \uparrow to CLKO \uparrow	—	75	200	ns	(Note 1)
12	TckR	CLKO Rise Time	—	15	30	ns	(Note 1)
13	TckF	CLKO Fall Time	—	15	30	ns	(Note 1)
14	TckL2ioV	CLKO \downarrow to Port Out Valid	—	—	0.5 TCY + 20	ns	
15	TioV2ckH	Port In Valid before CLKO \uparrow	0.25 TCY + 25	—	—	ns	
16	TckH2iol	Port In Hold after CLKO \uparrow	0	—	—	ns	
17	TosH2ioV	OSC1 \uparrow (Q1 cycle) to Port Out Valid	—	50	150	ns	
18	TosH2iol	OSC1 \uparrow (Q2 cycle) to Port Input Invalid (I/O in hold time)	100	—	—	ns	
19	TioV2osH	Port Input Valid to OSC1 \uparrow (I/O in setup time)	0	—	—	ns	
20	TioR	Port Output Rise Time	—	—	6	ns	
21	TioF	Port Output Fall Time	—	—	5	ns	
22†	TINP	INTx pin High or Low Time	TCY	—	—	ns	
23†	TRBP	RB7:RB4 Change INTx High or Low Time	TCY	—	—	ns	

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

Note 1: Measurements are taken in EC mode, where CLKO output is 4 x Tosc.

FIGURE 27-6: PROGRAM MEMORY READ TIMING DIAGRAM

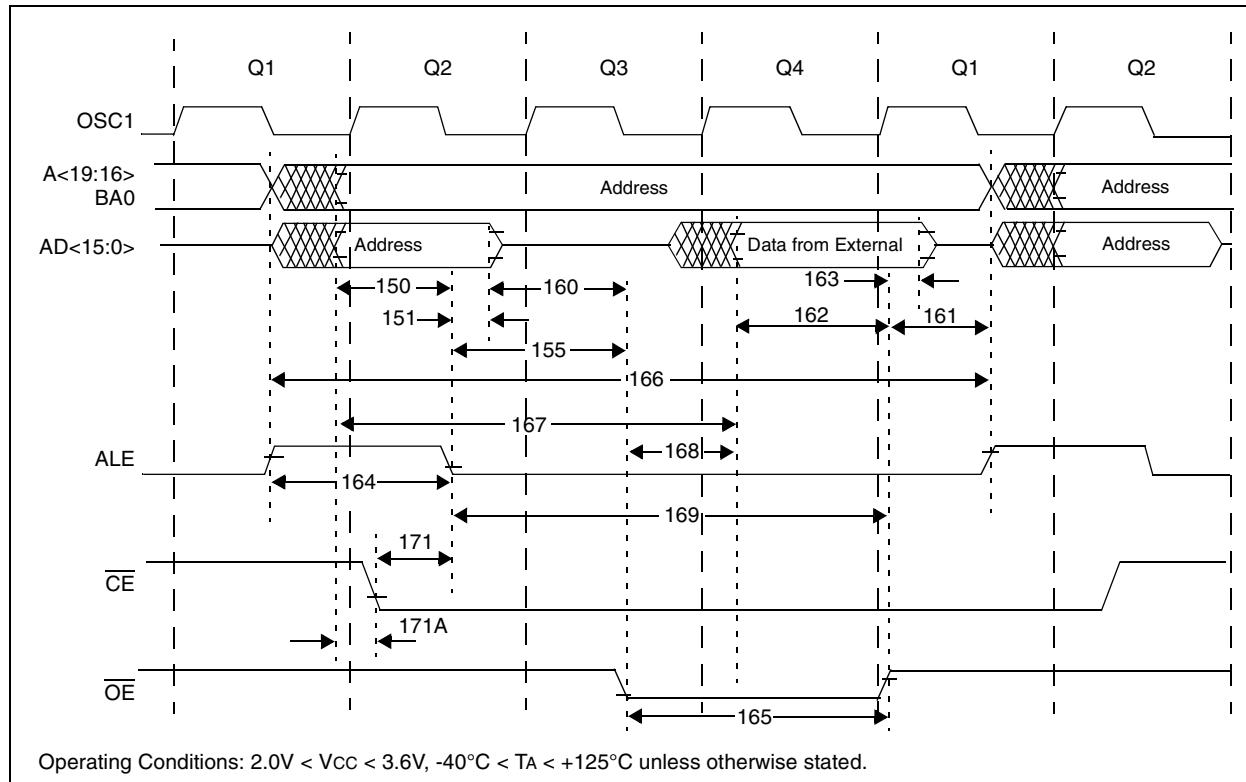


TABLE 27-10: CLKO AND I/O TIMING REQUIREMENTS

Param. No	Symbol	Characteristics	Min	Typ	Max	Units
150	TadV2all	Address Out Valid to ALE ↓ (address setup time)	0.25 TCY - 10	—	—	ns
151	TalL2adl	ALE ↓ to Address Out Invalid (address hold time)	5	—	—	ns
155	TalL2oeL	ALE ↓ to OE ↓	10	0.125 TCY	—	ns
160	TadZ2oeL	AD high-Z to OE ↓ (bus release to OE)	0	—	—	ns
161	ToeH2adD	OE ↑ to AD Driven	0.125 TCY - 5	—	—	ns
162	TadV2oeH	Least Significant Data Valid before OE ↑ (data setup time)	20	—	—	ns
163	ToeH2adl	OE ↑ to Data In Invalid (data hold time)	0	—	—	ns
164	TalH2all	ALE Pulse Width	—	0.25 TCY	—	ns
165	ToeL2oeH	OE Pulse Width	0.5 TCY - 5	0.5 TCY	—	ns
166	TalH2alH	ALE ↑ to ALE ↑ (cycle time)	—	TCY	—	ns
167	Tacc	Address Valid to Data Valid	0.75 TCY - 25	—	—	ns
168	Toe	OE ↓ to Data Valid	—	—	0.5 TCY - 25	ns
169	TalL2oeH	ALE ↓ to OE ↑	0.625 TCY - 10	—	0.625 TCY + 10	ns
171	TalH2csL	Chip Enable Active to ALE ↓	0.25 TCY - 20	—	—	ns
171A	TubL2oeH	AD Valid to Chip Enable Active	—	—	10	ns

PIC18F87J11 FAMILY

FIGURE 27-7: PROGRAM MEMORY WRITE TIMING DIAGRAM

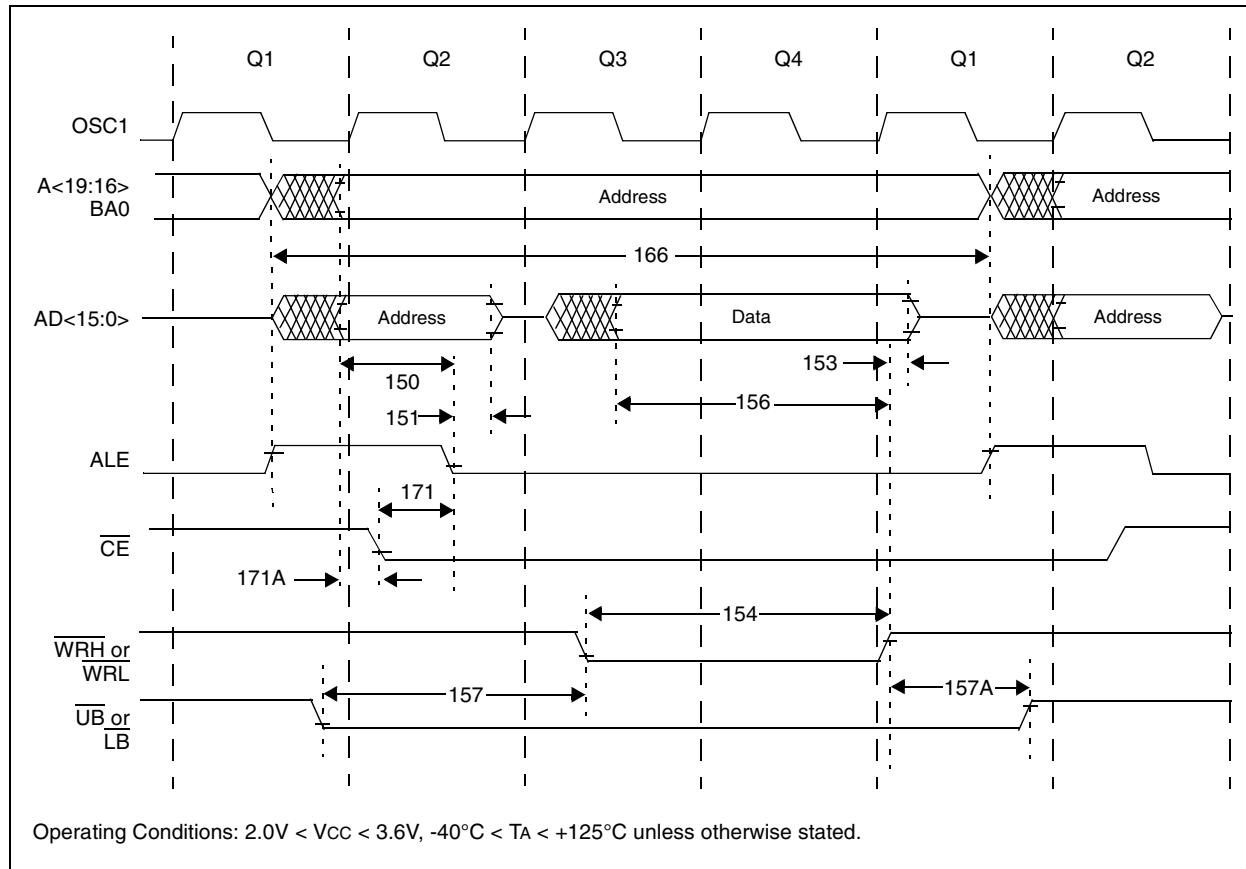


TABLE 27-11: PROGRAM MEMORY WRITE TIMING REQUIREMENTS

Param. No	Symbol	Characteristics	Min	Typ	Max	Units
150	TadV2all	Address Out Valid to ALE ↓ (address setup time)	0.25 TCY – 10	—	—	ns
151	TalH2adl	ALE ↓ to Address Out Invalid (address hold time)	5	—	—	ns
153	TwrH2adl	WRn ↑ to Data Out Invalid (data hold time)	5	—	—	ns
154	TwrL	WRn Pulse Width	0.5 TCY – 5	0.5 TCY	—	ns
156	TadV2wrH	Data Valid before WRn ↑ (data setup time)	0.5 TCY – 10	—	—	ns
157	TbsV2wrL	Byte Select Valid before WRn ↓ (byte select setup time)	0.25 TCY	—	—	ns
157A	TwrH2bsl	WRn ↑ to Byte Select Invalid (byte select hold time)	0.125 TCY – 5	—	—	ns
166	TalH2alH	ALE ↑ to ALE ↑ (cycle time)	—	TCY	—	ns
171	TalH2csL	Chip Enable Active to ALE ↓	0.25 TCY – 20	—	—	ns
171A	TubL2oeH	AD Valid to Chip Enable Active	—	—	10	ns

FIGURE 27-8: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

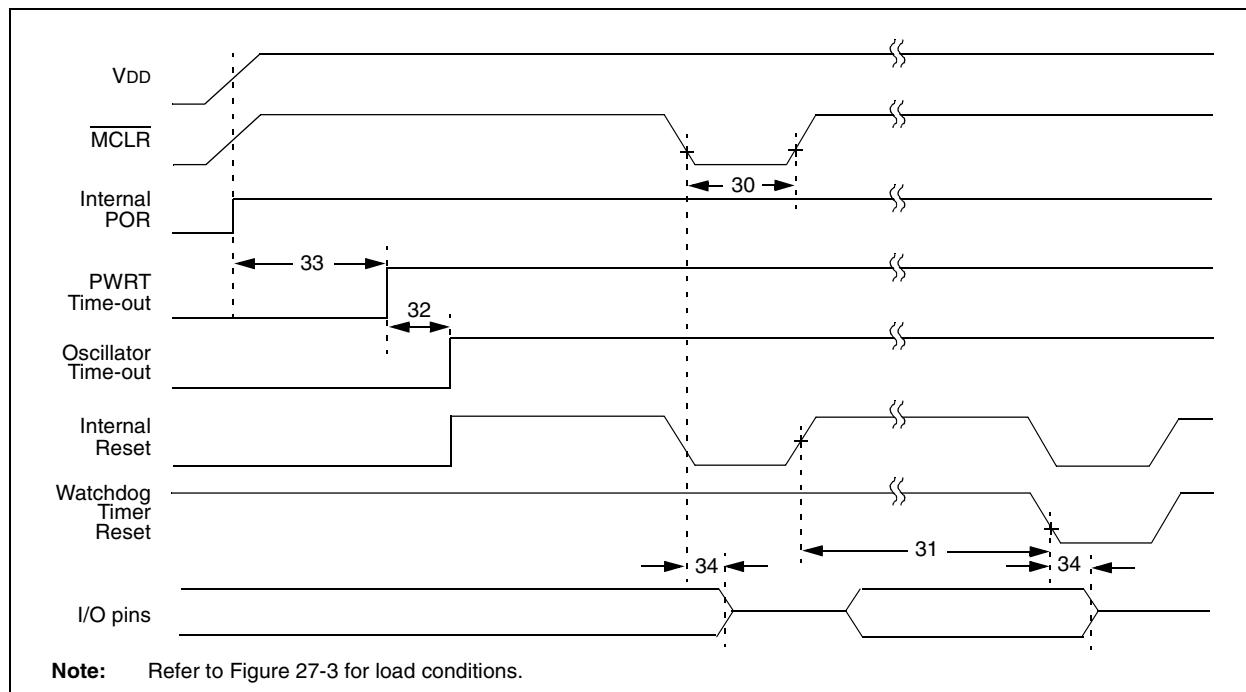


TABLE 27-12: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
30	T _{MCL}	MCLR Pulse Width (low)	2	—	—	TCY	(Note 1)
31	T _{WDT}	Watchdog Timer Time-out Period (no postscaler)	3.4	4.0	4.6	ms	
32	T _{OST}	Oscillator Start-up Timer Period	1024 T _{osc}	—	1024 T _{osc}	—	T _{osc} = OSC1 period
33	T _{PWRT}	Power-up Timer Period	45.8	65.5	85.2	ms	
34	T _{IOZ}	I/O High-Impedance from MCLR Low or Watchdog Timer Reset	—	2	—	μs	
38	T _{CSD}	CPU Start-up Time	—	200	—	μs	

Note 1: To ensure device reset, MCLR must be low for at least 2 TCY or 400 μs, whichever is lower.

PIC18F87J11 FAMILY

TABLE 27-13: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

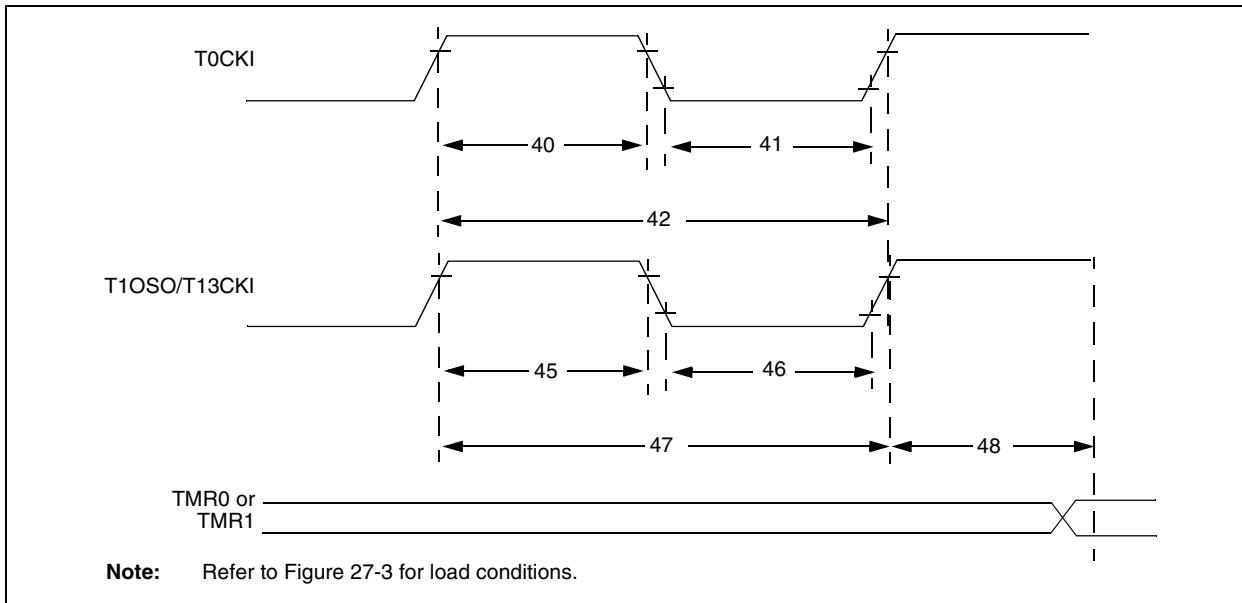


TABLE 27-14: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
40	T _{T0H}	T0CKI High Pulse Width	No prescaler	0.5 TCY + 20	—	ns	
			With prescaler	10	—	ns	
41	T _{T0L}	T0CKI Low Pulse Width	No prescaler	0.5 TCY + 20	—	ns	
			With prescaler	10	—	ns	
42	T _{T0P}	T0CKI Period	No prescaler	TCY + 10	—	ns	N = prescale value (1, 2, 4,..., 256)
			With prescaler	Greater of: 20 ns or (TCY + 40)/N	—	ns	
45	T _{T1H}	T13CKI High Time	Synchronous, no prescaler	0.5 TCY + 20	—	ns	
			Synchronous, with prescaler	10	—	ns	
			Asynchronous	30	—	ns	
46	T _{T1L}	T13CKI Low Time	Synchronous, no prescaler	0.5 TCY + 5	—	ns	
			Synchronous, with prescaler	10	—	ns	
			Asynchronous	30	—	ns	
47	T _{T1P}	T13CKI Input Period	Synchronous	Greater of: 20 ns or (TCY + 40)/N	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	60	—	ns	
	F _{T1}	T13CKI Oscillator Input Frequency Range		DC	50	kHz	
48	TCKE2TMRI	Delay from External T13CKI Clock Edge to Timer Increment		2 Tosc	7 Tosc	—	

FIGURE 27-9: PARALLEL SLAVE PORT TIMING

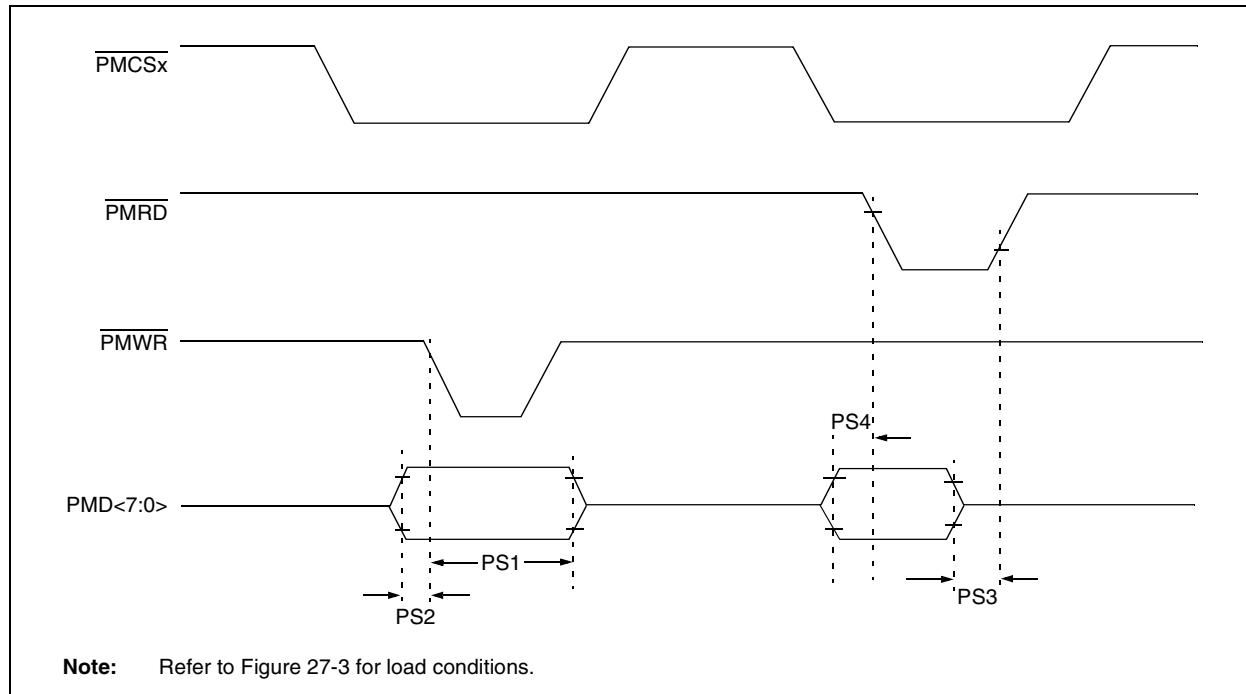


TABLE 27-15: PARALLEL SLAVE PORT REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
PS1	TdtV2wrH	Data In Valid before PMWR or PMCSx Inactive (setup time)	20	—	ns	
PS2	TwrH2dtI	PMWR or PMCSx Inactive to Data-In Invalid (hold time)	20	—	ns	
PS3	TrdL2dtV	PMRD and PMCSx Active to Data-Out Valid	—	80	ns	
PS4	TrdH2dtI	PMRD Active or PMCSx Inactive to Data-Out Invalid	10	30	ns	

PIC18F87J11 FAMILY

FIGURE 27-10: PARALLEL MASTER PORT READ TIMING DIAGRAM

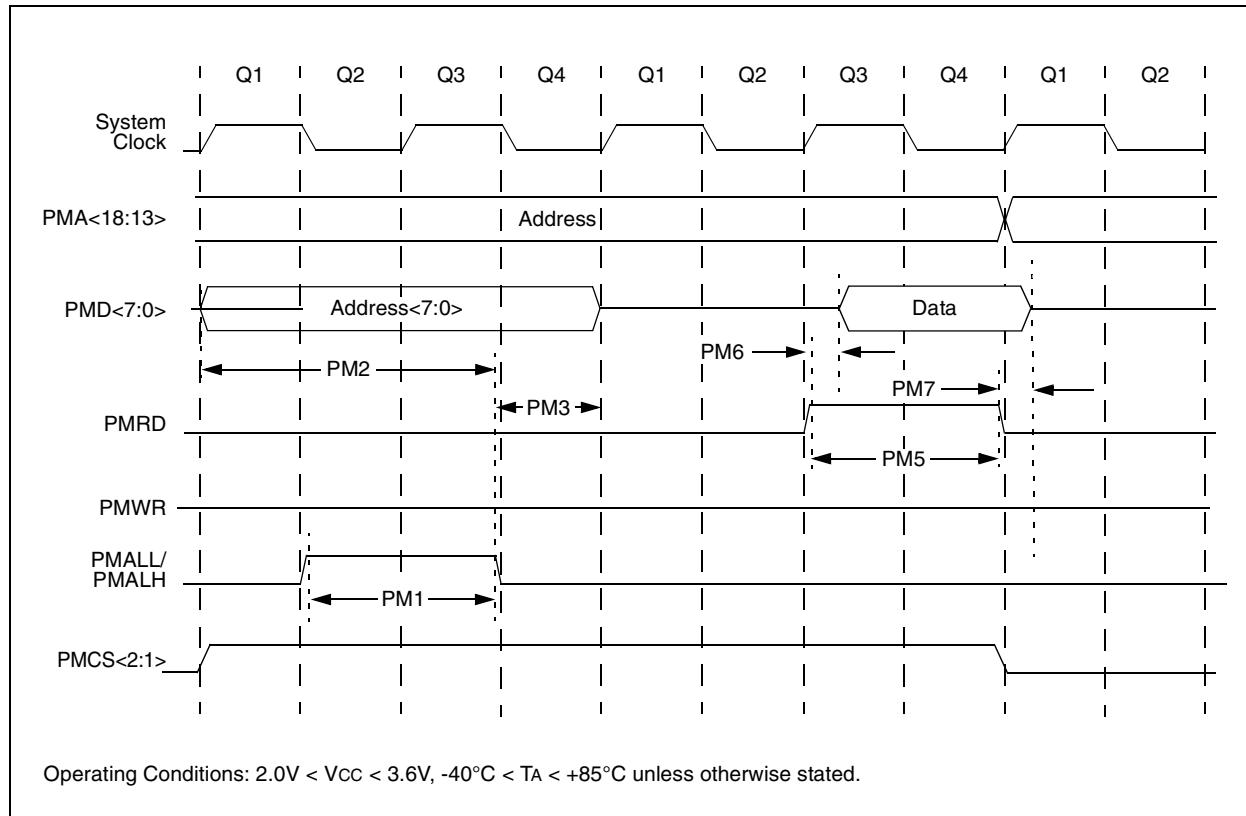


TABLE 27-16: PARALLEL MASTER PORT READ TIMING REQUIREMENTS

Param. No	Symbol	Characteristics	Min	Typ	Max	Units
PM1		PMALL/PMALH Pulse Width	—	0.5 T _{CY}	—	ns
PM2		Address out valid to PMALL/PMALH Invalid (address setup time)	—	0.75 T _{CY}	—	ns
PM3		PMALL/PMALH Invalid to Address Out Invalid (address hold time)	—	0.25 T _{CY}	—	ns
PM5		PMRD Pulse Width	—	0.5 T _{CY}	—	ns
PM6		PMRD or PMENB Active to Data In Valid (data setup time)	—	—	—	ns
PM7		PMRD or PMENB Inactive to Data In Invalid (data hold time)	—	—	—	ns

FIGURE 27-11: PARALLEL MASTER PORT WRITE TIMING DIAGRAM

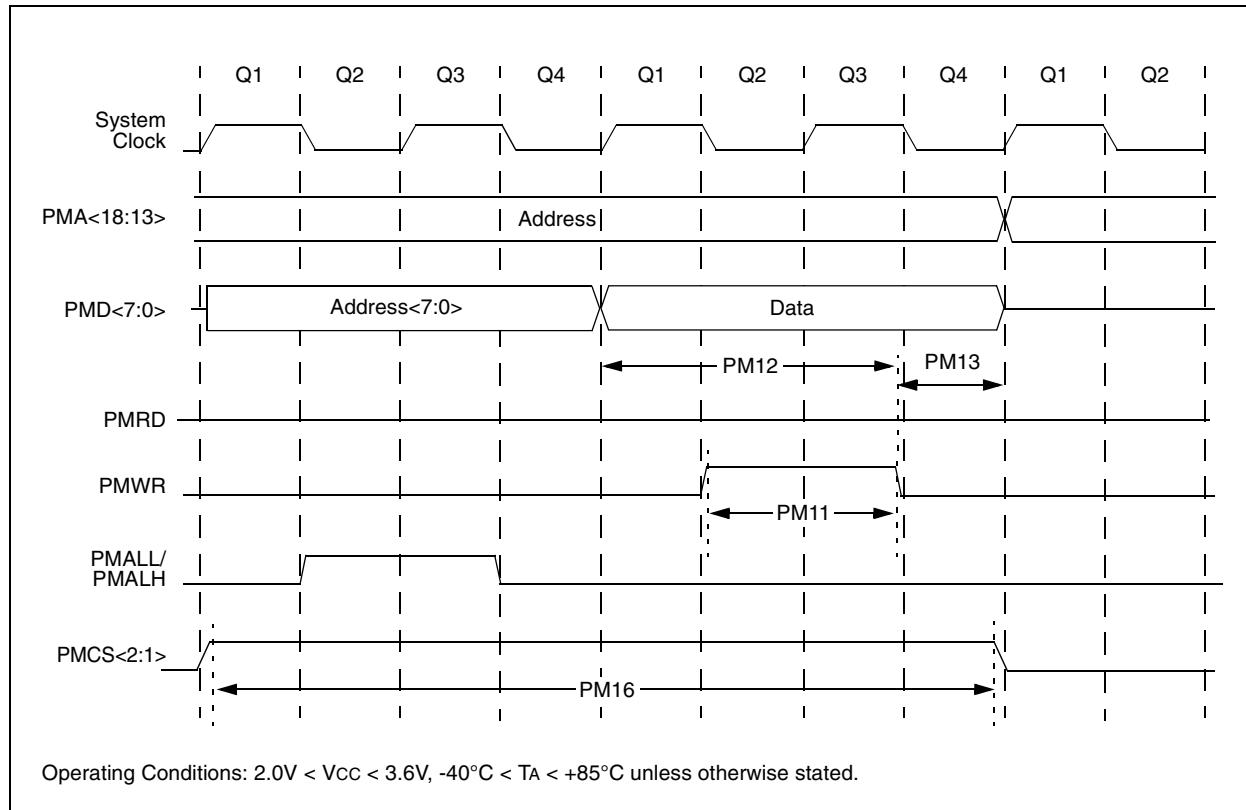
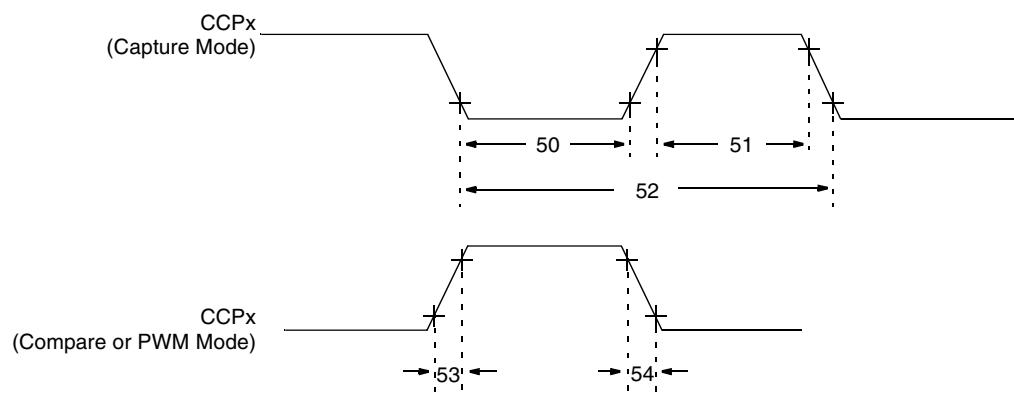


TABLE 27-17: PARALLEL MASTER PORT WRITE TIMING REQUIREMENTS

Param. No	Symbol	Characteristics	Min	Typ	Max	Units
PM11		PMWR Pulse Width	—	0.5 TCY	—	ns
PM12		Data Out Valid before PMWR or PMENB Goes Inactive (data setup time)	—	—	—	ns
PM13		PMWR or PMEMB Invalid to Data Out Invalid (data hold time)	—	—	—	ns
PM16		PMCSx Pulse Width	TCY – 5	—	—	ns

PIC18F87J11 FAMILY

FIGURE 27-12: CAPTURE/COMPARE/PWM TIMINGS (INCLUDING ECCP MODULES)



Note: Refer to Figure 27-3 for load conditions.

TABLE 27-18: CAPTURE/COMPARE/PWM REQUIREMENTS (INCLUDING ECCP MODULES)

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
50	TccL	CCPx Input Low Time	No prescaler	0.5 T _{CY} + 20	—	ns	
			With prescaler	10	—	ns	
51	TccH	CCPx Input High Time	No prescaler	0.5 T _{CY} + 20	—	ns	
			With prescaler	10	—	ns	
52	TccP	CCPx Input Period		$\frac{3 \text{ T}_{\text{CY}} + 40}{N}$	—	ns	N = prescale value (1, 4 or 16)
53	TccR	CCPx Output Fall Time		—	25	ns	
54	TccF	CCPx Output Fall Time		—	25	ns	

FIGURE 27-13: EXAMPLE SPI MASTER MODE TIMING (CKE = 0)

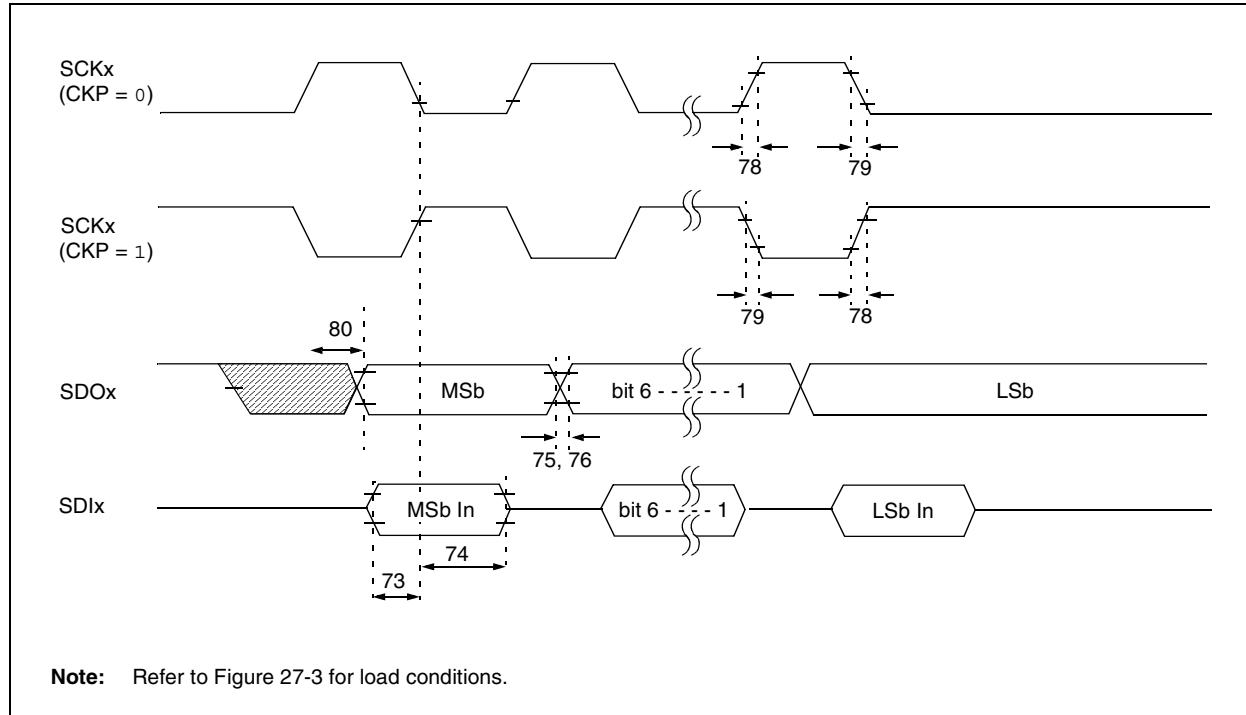
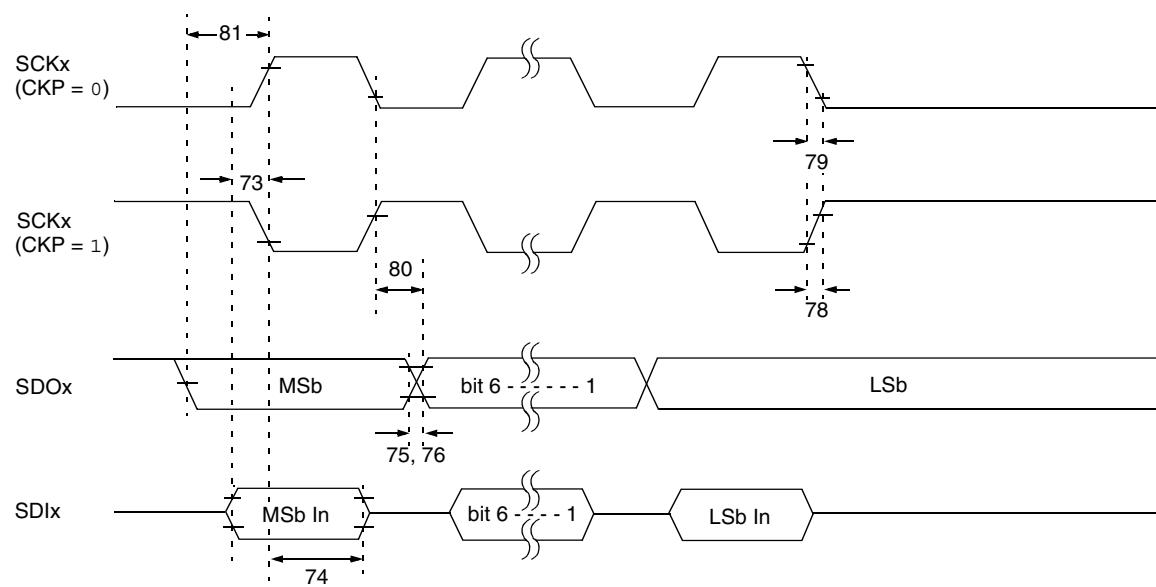


TABLE 27-19: EXAMPLE SPI MODE REQUIREMENTS (MASTER MODE, CKE = 0)

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
73	TdIV2sCH, TdIV2sCL	Setup Time of SDIx Data Input to SCKx Edge	100	—	ns	
73A	Tb2B	Last Clock Edge of Byte 1 to the 1st Clock Edge of Byte 2	1.5 TCY + 40	—	ns	
75	TdoR	SDOx Data Output Rise Time	—	25	ns	
76	TdoF	SDOx Data Output Fall Time	—	25	ns	
78	TscR	SCKx Output Rise Time	—	25	ns	
79	TscF	SCKx Output Fall Time	—	25	ns	
80	Tsch2dov, Tscl2dov	SDOx Data Output Valid after SCKx Edge	—	50	ns	

PIC18F87J11 FAMILY

FIGURE 27-14: EXAMPLE SPI MASTER MODE TIMING (CKE = 1)



Note: Refer to Figure 27-3 for load conditions.

TABLE 27-20: EXAMPLE SPI MODE REQUIREMENTS (MASTER MODE, CKE = 1)

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
73	T _{DIV2SCH} , T _{DIV2SCL}	Setup Time of SDIx Data Input to SCKx Edge	100	—	ns	
74	T _{sch2dil} , T _{scl2dil}	Hold Time of SDIx Data Input to SCKx Edge	100	—	ns	
75	T _{doR}	SDOx Data Output Rise Time	—	25	ns	
76	T _{doF}	SDOx Data Output Fall Time	—	25	ns	
78	T _{scR}	SCKx Output Rise Time	—	25	ns	
79	T _{scF}	SCKx Output Fall Time	—	25	ns	
80	T _{sch2dov} , T _{scl2dov}	SDOx Data Output Valid after SCKx Edge	—	50	ns	
81	T _{dov2sch} , T _{dov2scl}	SDOx Data Output Setup to SCKx Edge	T _{CY}	—	ns	

FIGURE 27-15: EXAMPLE SPI SLAVE MODE TIMING (CKE = 0)

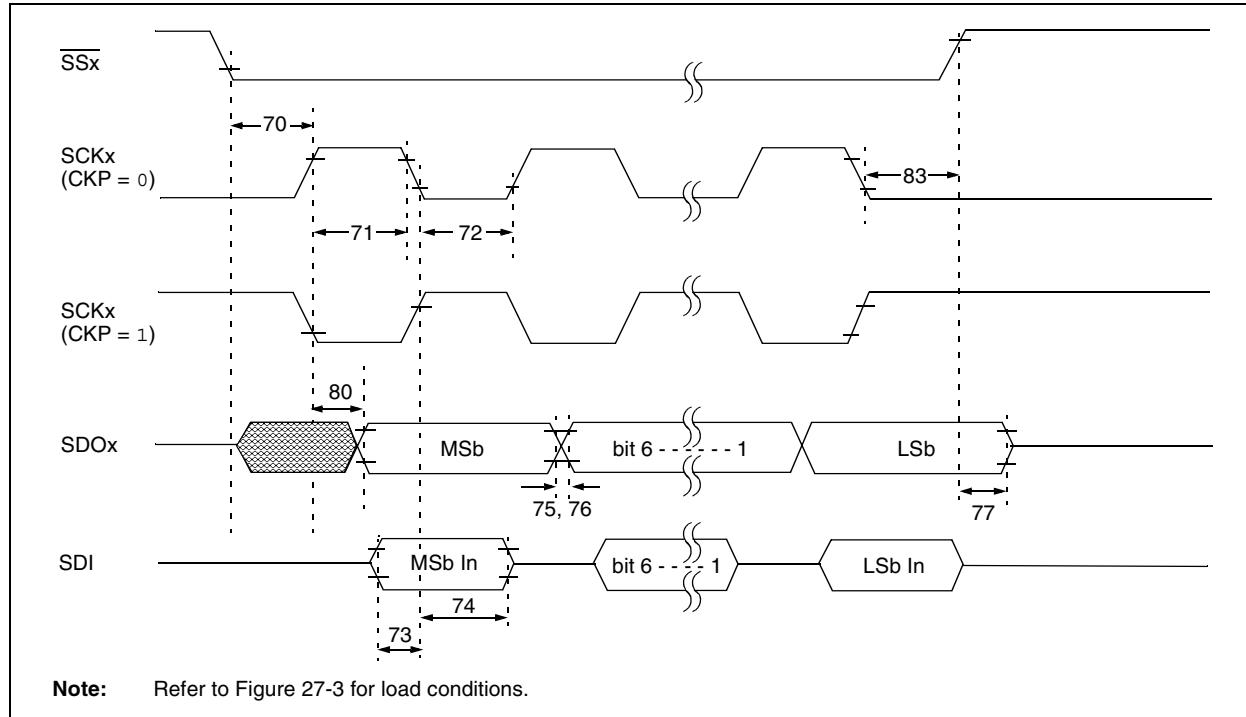


TABLE 27-21: EXAMPLE SPI MODE REQUIREMENTS (SLAVE MODE TIMING, CKE = 0)

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
70	TssL2sCH, TssL2sCL	$\overline{SSx} \downarrow$ to SCKx \downarrow or SCKx \uparrow Input	3 TCY	—	ns	
70A	TssL2WB	$\overline{SSx} \downarrow$ to write to SSPxBUF	3 TCY	—	ns	
71	TscH	SCKx Input High Time	Continuous	1.25 TCY + 30	—	ns
71A			Single byte	40	—	ns (Note 1)
72	TscL	SCKx Input Low Time	Continuous	1.25 TCY + 30	—	ns
72A			Single byte	40	—	ns (Note 1)
73	TdIV2sCH, TdIV2sCL	Setup Time of SDIx Data Input to SCKx Edge	100	—	ns	
73A	TB2B	Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2	1.5 TCY + 40	—	ns	(Note 2)
74	TsCH2DIL, TsCL2DIL	Hold Time of SDIx Data Input to SCKx Edge	100	—	ns	
75	TDoR	SDOx Data Output Rise Time	—	25	ns	
76	TDoF	SDOx Data Output Fall Time	—	25	ns	
77	TsH2D0Z	$\overline{SSx} \uparrow$ to SDOx Output High-Impedance	10	50	ns	
80	TscH2D0V, TscL2D0V	SDOx Data Output Valid after SCKx Edge	—	50	ns	
83	TscH2ssH, TscL2ssH	$\overline{SSx} \uparrow$ after SCKx Edge	1.5 TCY + 40	—	ns	

Note 1: Requires the use of Parameter #73A.

2: Only if Parameter #71A and #72A are used.

PIC18F87J11 FAMILY

FIGURE 27-16: EXAMPLE SPI SLAVE MODE TIMING (CKE = 1)

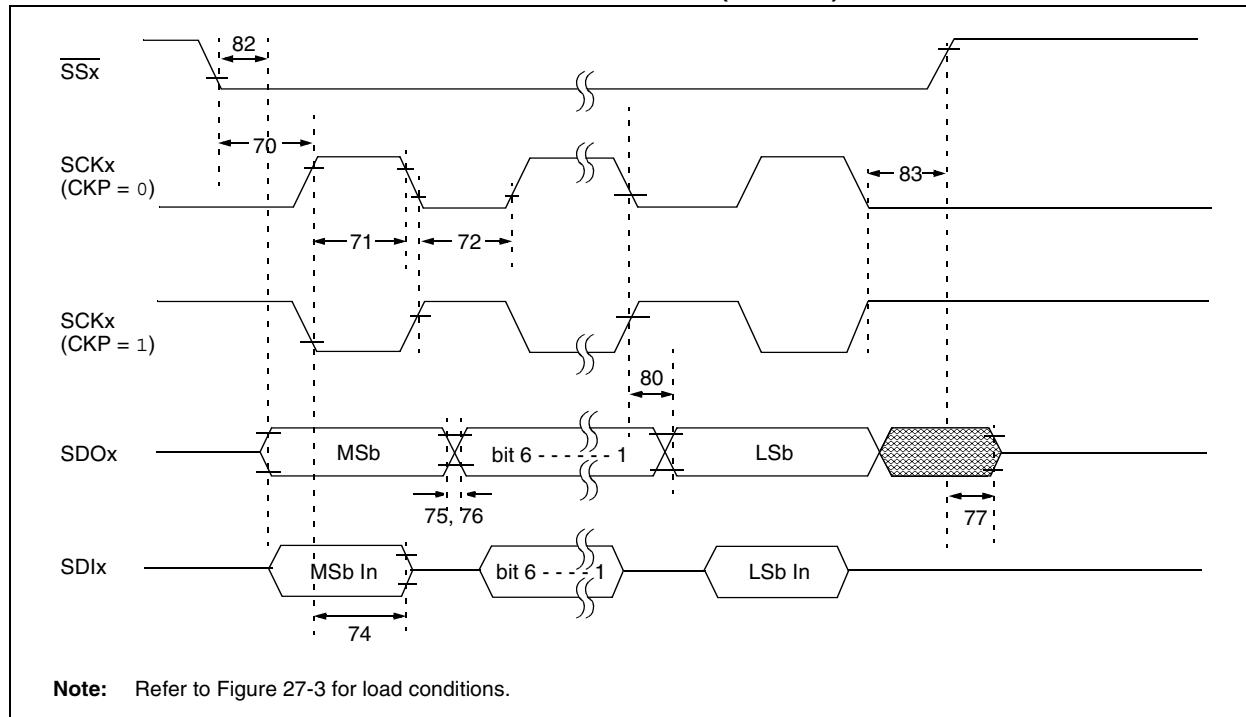


TABLE 27-22: EXAMPLE SPI SLAVE MODE REQUIREMENTS (CKE = 1)

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
70	TssL2sCH, TssL2sCL	SSx ↓ to SCKx ↓ or SCKx ↑ Input	3 TCY	—	ns	
70A	TssL2WB	SSx ↓ to write to SSPxBUF	3 TCY	—	ns	
71	TsCH	SCKx Input High Time	Continuous	1.25 TCY + 30	—	ns
		Single byte	40	—	ns	(Note 1)
72	TsCL	SCKx Input Low Time	Continuous	1.25 TCY + 30	—	ns
		Single byte	40	—	ns	(Note 1)
73A	TB2B	Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2	1.5 TCY + 40	—	ns	(Note 2)
74	TsCH2DIL, TsCL2DIL	Hold Time of SDIx Data Input to SCKx Edge	100	—	ns	
75	TDOFR	SDOx Data Output Rise Time	—	25	ns	
76	TDOF	SDOx Data Output Fall Time	—	25	ns	
77	TSSH2doZ	SSx ↑ to SDOx Output High-Impedance	10	50	ns	
80	TsCH2doV, TsCL2doV	SDOx Data Output Valid after SCKx Edge	—	50	ns	
82	TssL2doV	SDOx Data Output Valid after SSx ↓ Edge	—	50	ns	
83	TsCH2ssH, TsCL2ssH	SSx ↑ after SCKx Edge	1.5 TCY + 40	—	ns	

Note 1: Requires the use of Parameter #73A.

2: Only if Parameter #71A and #72A are used.

FIGURE 27-17: I²CTM BUS START/STOP BITS TIMING

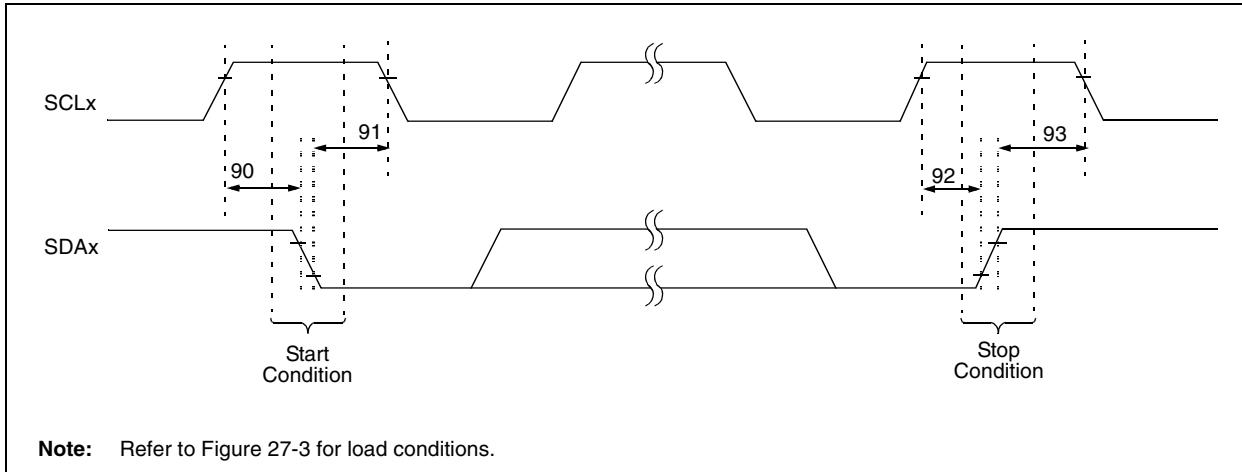
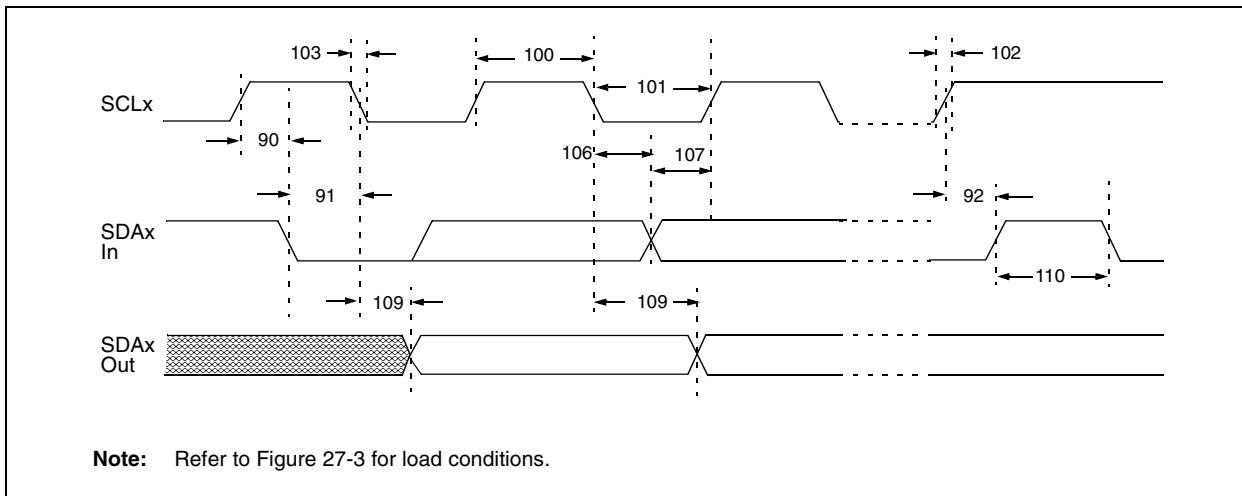


TABLE 27-23: I²CTM BUS START/STOP BITS REQUIREMENTS (SLAVE MODE)

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
90	TSU:STA	Start Condition	100 kHz mode	4700	—	ns Only relevant for Repeated Start condition
		Setup Time	400 kHz mode	600	—	
91	THD:STA	Start Condition	100 kHz mode	4000	—	ns After this period, the first clock pulse is generated
		Hold Time	400 kHz mode	600	—	
92	TSU:STO	Stop Condition	100 kHz mode	4700	—	ns
		Setup Time	400 kHz mode	600	—	
93	THD:STO	Stop Condition	100 kHz mode	4000	—	ns
		Hold Time	400 kHz mode	600	—	

FIGURE 27-18: I²CTM BUS DATA TIMING



PIC18F87J11 FAMILY

TABLE 27-24: I²C™ BUS DATA REQUIREMENTS (SLAVE MODE)

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
100	THIGH	Clock High Time	100 kHz mode	4.0	—	μs	
			400 kHz mode	0.6	—	μs	
			MSSP modules	1.5 T _{CY}	—		
101	TLOW	Clock Low Time	100 kHz mode	4.7	—	μs	
			400 kHz mode	1.3	—	μs	
			MSSP modules	1.5 T _{CY}	—		
102	TR	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1 C _B	300	ns	C _B is specified to be from 10 to 400 pF
103	TF	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1 C _B	300	ns	C _B is specified to be from 10 to 400 pF
90	TSU:STA	Start Condition Setup Time	100 kHz mode	4.7	—	μs	Only relevant for Repeated Start condition
			400 kHz mode	0.6	—	μs	
91	THD:STA	Start Condition Hold Time	100 kHz mode	4.0	—	μs	After this period, the first clock pulse is generated
			400 kHz mode	0.6	—	μs	
106	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
107	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	(Note 2)
			400 kHz mode	100	—	ns	
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	4.7	—	μs	
			400 kHz mode	0.6	—	μs	
109	TAA	Output Valid from Clock	100 kHz mode	—	3500	ns	(Note 1)
			400 kHz mode	—	—	ns	
110	TBUF	Bus Free Time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
D102	CB	Bus Capacitive Loading		—	400	pF	

- Note 1:** As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCLx to avoid unintended generation of Start or Stop conditions.
- 2:** A Fast mode I²C™ bus device can be used in a Standard mode I²C bus system, but the requirement, TSU:DAT ≥ 250 ns, must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCLx signal. If such a device does stretch the LOW period of the SCLx signal, it must output the next data bit to the SDAx line, TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCLx line is released.

FIGURE 27-19: MSSP I²C™ BUS START/STOP BITS TIMING WAVEFORMS

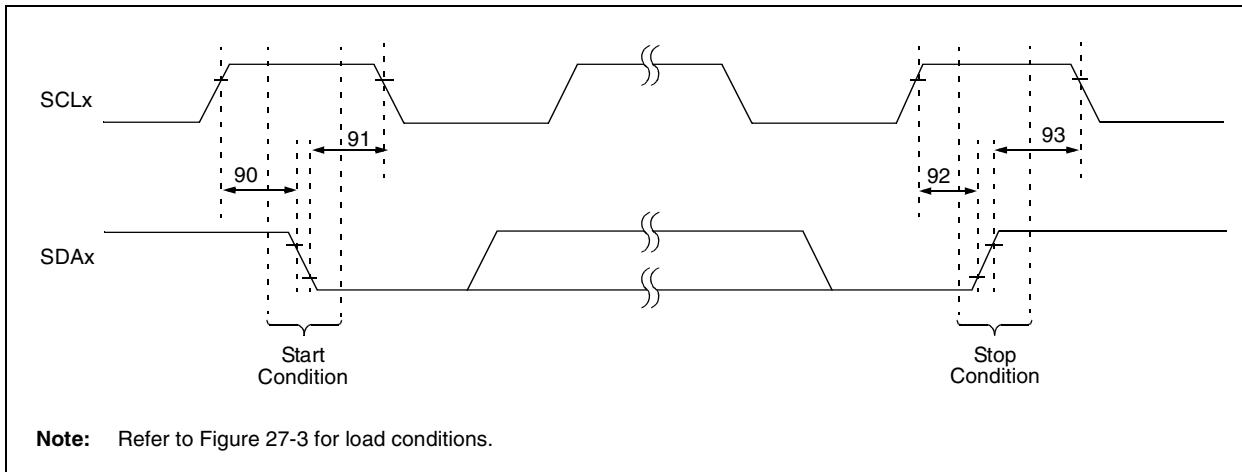
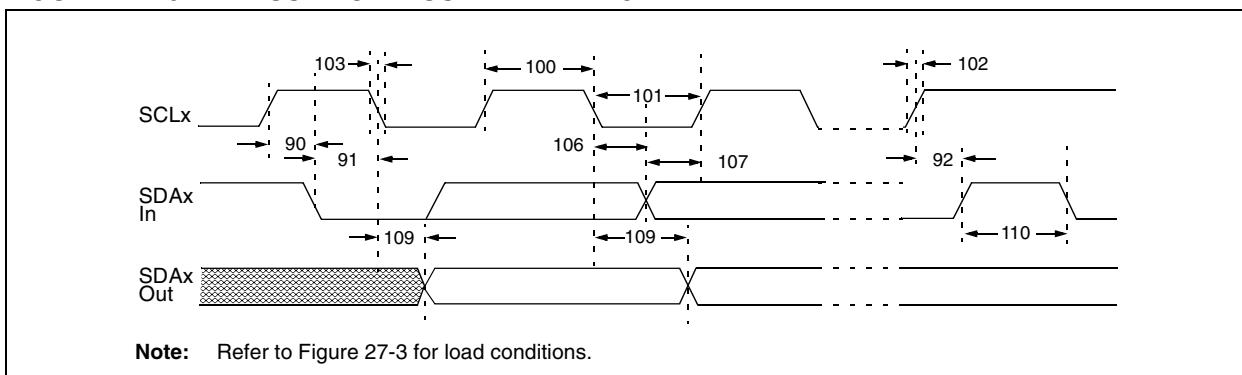


TABLE 27-25: MSSP I²C™ BUS START/STOP BITS REQUIREMENTS

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
90	TSU:STA	Start Condition Setup Time	100 kHz mode	2(Tosc)(BRG + 1)	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	2(Tosc)(BRG + 1)	—		
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—		
91	THD:STA	Start Condition Hold Time	100 kHz mode	2(Tosc)(BRG + 1)	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	2(Tosc)(BRG + 1)	—		
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—		
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	2(Tosc)(BRG + 1)	—	ns	
			400 kHz mode	2(Tosc)(BRG + 1)	—		
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—		
93	THD:STO	Stop Condition Hold Time	100 kHz mode	2(Tosc)(BRG + 1)	—	ns	
			400 kHz mode	2(Tosc)(BRG + 1)	—		
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—		

Note 1: Maximum pin capacitance = 10 pF for all I²C™ pins.

FIGURE 27-20: MSSP I²C™ BUS DATA TIMING



PIC18F87J11 FAMILY

TABLE 27-26: MSSP I²C™ BUS DATA REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
100	THIGH	Clock High Time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	ms
101	TLOW	Clock Low Time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	ms
102	TR	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns
			400 kHz mode	20 + 0.1 CB	300	ns
			1 MHz mode ⁽¹⁾	—	300	ns
103	TF	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns
			400 kHz mode	20 + 0.1 CB	300	ns
			1 MHz mode ⁽¹⁾	—	100	ns
90	TSU:STA	Start Condition Setup Time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	ms
91	THD:STA	Start Condition Hold Time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	ms
106	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns
			400 kHz mode	0	0.9	ms
			1 MHz mode ⁽¹⁾	TBD	—	ns
107	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns
			400 kHz mode	100	—	ns
			1 MHz mode ⁽¹⁾	TBD	—	ns
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	ms
109	TAA	Output Valid from Clock	100 kHz mode	—	3500	ns
			400 kHz mode	—	1000	ns
			1 MHz mode ⁽¹⁾	—	—	ns
110	TBUF	Bus Free Time	100 kHz mode	4.7	—	ms
			400 kHz mode	1.3	—	ms
			1 MHz mode ⁽¹⁾	TBD	—	ms
D102	CB	Bus Capacitive Loading	—	400	pF	

Legend: TBD = To Be Determined

Note 1: Maximum pin capacitance = 10 pF for all I²C™ pins.

- 2:** A Fast mode I²C bus device can be used in a Standard mode I²C bus system, but parameter #107 \geq 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCLx signal. If such a device does stretch the LOW period of the SCLx signal, it must output the next data bit to the SDAx line, parameter #102 + parameter #107 = 1000 + 250 = 1250 ns (for 100 kHz mode), before the SCLx line is released.

FIGURE 27-21: EUSART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

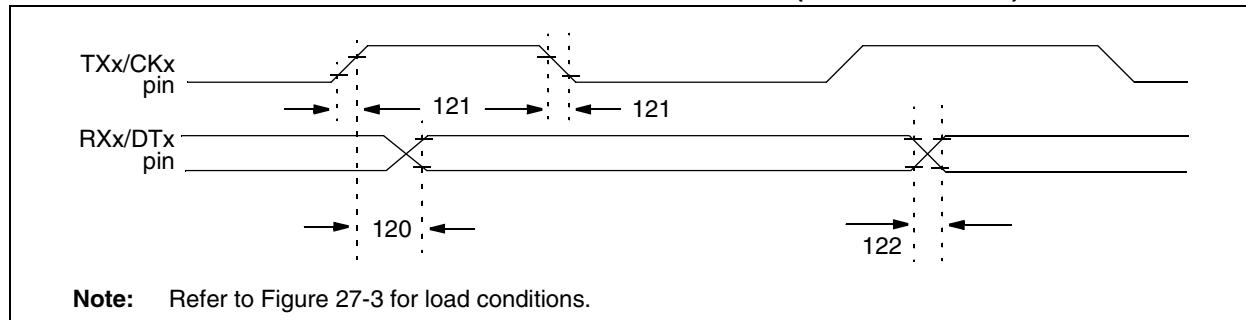


TABLE 27-27: EUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
120	TCKH2DTV	<u>SYNC XMIT (MASTER and SLAVE)</u> Clock High to Data Out Valid	—	40	ns	
121	TCKRF	Clock Out Rise Time and Fall Time (Master mode)	—	20	ns	
122	TDTRF	Data Out Rise Time and Fall Time	—	20	ns	

FIGURE 27-22: EUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING

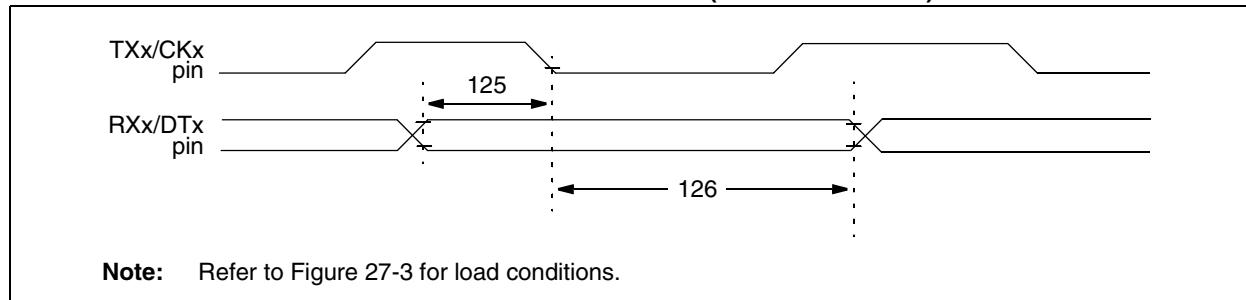


TABLE 27-28: EUSART SYNCHRONOUS RECEIVE REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
125	TDTV2CKL	<u>SYNC RCV (MASTER and SLAVE)</u> Data Hold before CKx ↓ (DTx hold time)	10	—	ns	
126	TCKL2DTL	Data Hold after CKx ↓ (DTx hold time)	15	—	ns	

PIC18F87J11 FAMILY

TABLE 27-29: A/D CONVERTER CHARACTERISTICS: PIC18F87J11 FAMILY (INDUSTRIAL)

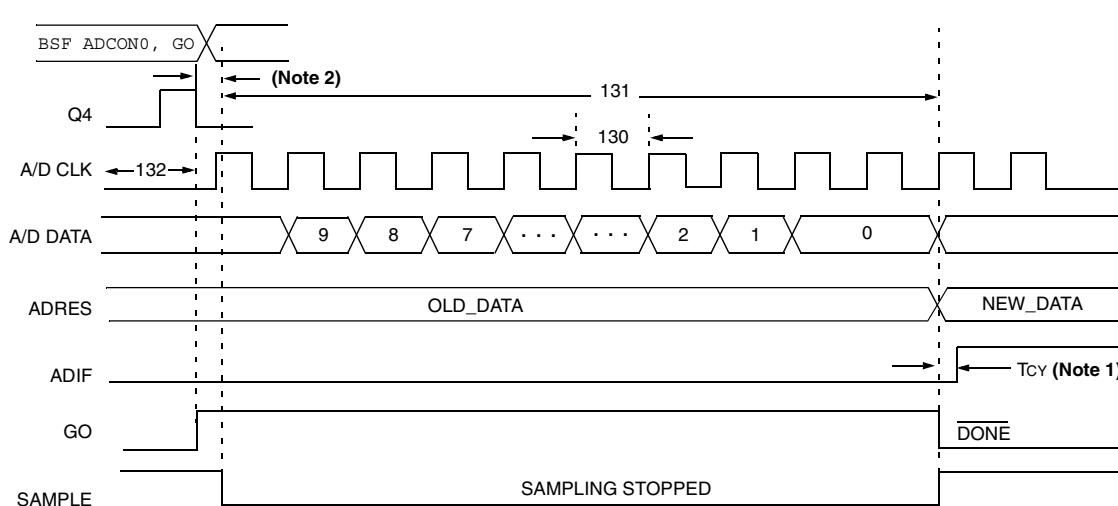
Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
A01	NR	Resolution	—	—	10	bit	$\Delta V_{REF} \geq 3.0V$
A03	EIL	Integral Linearity Error	—	—	< ± 1	LSb	$\Delta V_{REF} \geq 3.0V$
A04	EDL	Differential Linearity Error	—	—	< ± 1	LSb	$\Delta V_{REF} \geq 3.0V$
A06	EOFF	Offset Error	—	—	< ± 3	LSb	$\Delta V_{REF} \geq 3.0V$
A07	EGN	Gain Error	—	—	< ± 3	LSb	$\Delta V_{REF} \geq 3.0V$
A10	—	Monotonicity	Guaranteed ⁽¹⁾			—	$V_{SS} \leq V_{AIN} \leq V_{REF}$
A20	ΔV_{REF}	Reference Voltage Range ($V_{REFH} - V_{REFL}$)	2.0 3	—	—	V	$V_{DD} < 3.0V$ $V_{DD} \geq 3.0V$
A21	V_{REFH}	Reference Voltage High	V_{SS}	—	V_{REFH}	V	
A22	V_{REFL}	Reference Voltage Low	$V_{SS} - 0.3V$	—	$V_{DD} - 3.0V$	V	
A25	V_{AIN}	Analog Input Voltage	V_{REFL}	—	V_{REFH}	V	
A30	Z_{AIN}	Recommended Impedance of Analog Voltage Source	—	—	2.5	k Ω	
A50	I _{REF}	V _{REF} Input Current ⁽²⁾	—	—	5 150	μA	During V_{AIN} acquisition. During A/D conversion cycle.

Note 1: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

2: V_{REFH} current is from RA3/AN3/V_{REF+} pin or V_{DD}, whichever is selected as the V_{REFH} source.

V_{REFL} current is from RA2/AN2/V_{REF-} pin or V_{SS}, whichever is selected as the V_{REFL} source.

FIGURE 27-23: A/D CONVERSION TIMING



Note 1: If the A/D clock source is selected as RC, a time of T_{CY} is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

2: This is a minimal RC delay (typically 100 ns), which also disconnects the holding capacitor from the analog input.

PIC18F87J11 FAMILY

TABLE 27-30: A/D CONVERSION REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
130	TAD	A/D Clock Period	0.7	25.0 ⁽¹⁾	μs	Tosc based, VREF ≥ 3.0V
			TBD	1	μs	A/D RC mode
131	T _{CNV}	Conversion Time (not including acquisition time) (Note 2)	11	12	TAD	
132	T _{ACQ}	Acquisition Time (Note 3)	1.4	—	μs	-40°C to +85°C
135	T _{SWC}	Switching Time from Convert → Sample	—	(Note 4)		
TBD	T _{DIS}	Discharge Time	0.2	—	μs	

Legend: TBD = To Be Determined

Note 1: The time of the A/D clock period is dependent on the device frequency and the TAD clock divider.

2: ADRES registers may be read on the following T_{CY} cycle.

3: The time for the holding capacitor to acquire the “New” input voltage when the voltage changes full scale after the conversion (VDD to Vss or Vss to VDD). The source impedance (Rs) on the input channels is 50Ω.

4: On the following cycle of the device clock.

PIC18F87J11 FAMILY

NOTES:

28.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

Graphs and tables are not available at this time.

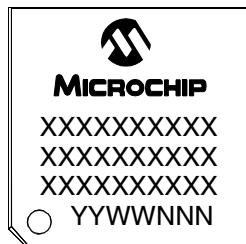
PIC18F87J11 FAMILY

NOTES:

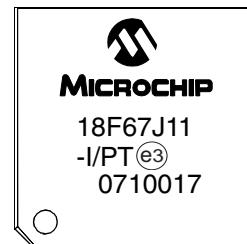
29.0 PACKAGING INFORMATION

29.1 Package Marking Information

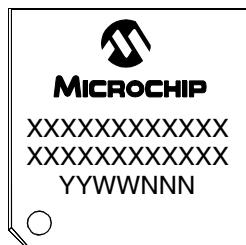
64-Lead TQFP



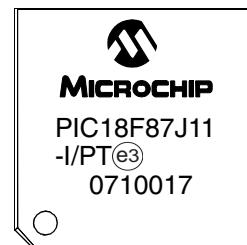
Example



80-Lead TQFP



Example



Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
*		This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

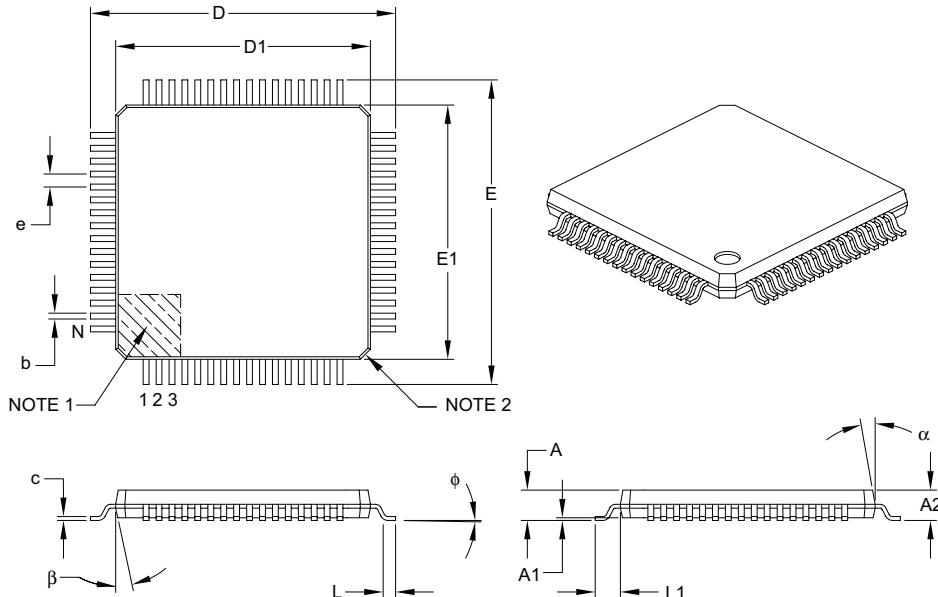
PIC18F87J11 FAMILY

29.2 Package Details

The following sections give the technical details of the packages.

64-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm Footprint [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Dimension Limits		MILLIMETERS		
		MIN	NOM	MAX
Number of Leads	N	64		
Lead Pitch	e	0.50	BSC	
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	–	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Foot Angle	ϕ	0°	3.5°	7°
Overall Width	E	12.00 BSC		
Overall Length	D	12.00 BSC		
Molded Package Width	E1	10.00 BSC		
Molded Package Length	D1	10.00 BSC		
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

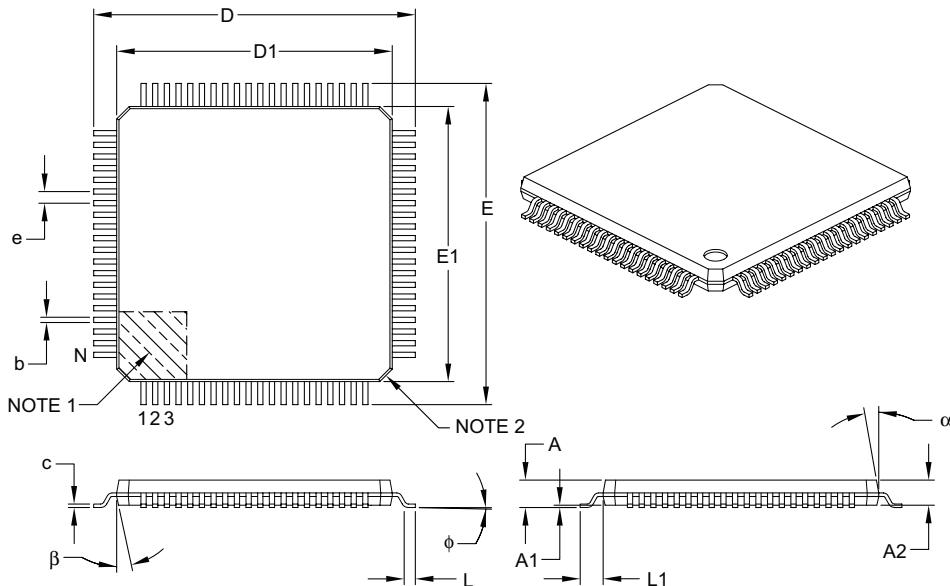
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-085B

80-Lead Plastic Thin Quad Flatpack (PT) – 12x12x1 mm Body, 2.00 mm Footprint [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads	N	80		
Lead Pitch	e	0.50	BSC	
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	–	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Foot Angle	φ	0°	3.5°	7°
Overall Width	E	14.00 BSC		
Overall Length	D	14.00 BSC		
Molded Package Width	E1	12.00 BSC		
Molded Package Length	D1	12.00 BSC		
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-092B

PIC18F87J11 FAMILY

NOTES:

PIC18F87J11 FAMILY

APPENDIX A: REVISION HISTORY

Revision A (January 2007)

Original data sheet for the PIC18F87J11 Family of devices.

Revision B (February 2007)

Updated values in Power-Down and Supply Current table in "DC Characteristics" section.

APPENDIX B: DEVICE DIFFERENCES

The differences between the devices listed in this data sheet are shown in Table B-1.

TABLE B-1: DEVICE DIFFERENCES BETWEEN PIC18F87J11 FAMILY MEMBERS

Features	PIC18F66J11	PIC18F66J16	PIC18F67J11	PIC18F86J11	PIC18F86J16	PIC18F87J11
Program memory	64K	96K	128K	64K	96K	128K
Program Memory (Instructions)	32764	49148	65532	32764	49148	65532
I/O Ports	Ports A, B, C, D, E, F, G				Ports A, B, C, D, E, F, G, H, J	
EMB	No				Yes	
10-Bit ADC module	11 Input Channels				15 Input Channels	
Packages	64-Pin TQFP				80-Pin TQFP	

PIC18F87J11 FAMILY

NOTES:

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com, click on Customer Change Notification and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: <http://support.microchip.com>

PIC18F87J11 FAMILY

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager

Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? Y N

Device: PIC18F87J11 Family

Literature Number: DS39778B

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

INDEX

A

A/D	291
A/D Converter Interrupt, Configuring	295
Acquisition Requirements	296
ADCAL Bit	299
ADRESH Register	294
Analog Port Pins, Configuring	297
Associated Registers	300
Automatic Acquisition Time	297
Calibration	299
Configuring the Module	295
Conversion Clock (TAD)	297
Conversion Requirements	421
Conversion Status (GO/DONE Bit)	294
Conversions	298
Converter Characteristics	420
Operation in Power-Managed Modes	299
Special Event Trigger (ECCP)	210, 298
Use of the ECCP2 Trigger	298
Absolute Maximum Ratings	383
AC (Timing) Characteristics	398
Load Conditions for Device Timing	
Specifications	399
Parameter Symbology	398
Temperature and Voltage Specifications	399
Timing Conditions	399
ACKSTAT	259
ACKSTAT Status Flag	259
ADCAL Bit	299
ADCON0 Register	
GO/DONE Bit	294
ADDFSR	372
ADDLW	335
ADDULNK	372
ADDWF	335
ADDWFC	336
ADRESL Register	294
Analog-to-Digital Converter. See A/D.	
ANDLW	336
ANDWF	337
Assembler	
MPASM Assembler	380
Auto-Wake-up on Sync Break Character	282

B

Baud Rate Generator	255
BC	337
BCF	338
BF	259
BF Status Flag	259
Block Diagrams	
16-Bit Byte Select Mode	103
16-Bit Byte Write Mode	101
16-Bit Word Write Mode	102
8-Bit Multiplexed Address and	
Data Application	178
8-Bit Multiplexed Mode Example	105
A/D	294
Analog Input Model	295
Baud Rate Generator	255
Capture Mode Operation	201
Comparator	301
Comparator Analog Input Model	304

Comparator I/O Configurations	306
Comparator Voltage Reference	309
Comparator Voltage Reference	
Output Buffer Example	311
Compare Mode Operation	202
Connections for On-Chip Voltage Regulator	323
Demultiplexed Addressing Mode	171
Device Clock	31
Enhanced PWM	211
EUSART Receive	281
EUSART Transmit	279
External Power-on Reset Circuit	
(Slow VDD Power-up)	51
Fail-Safe Clock Monitor	325
Fully Multiplexed Addressing Mode	171
Generic I/O Port Operation	127
Interrupt Logic	112
LCD Control	179
Legacy Parallel Slave Port	165
MSSP (SPI Mode)	223
MSSPx (I ² C Master Mode)	253
MSSPx (I ² C Mode)	233
Multiplexed Addressing Application	178
On-Chip Reset Circuit	49
Parallel EEPROM (Up to 15-Bit Address,	
16-Bit Data)	179
Parallel EEPROM (Up to 15-Bit Address,	
8-Bit Data)	179
Parallel Master/Slave Connection	
Addressed Buffer	168
Parallel Master/Slave Connection Buffered	167
Partially Multiplexed Addressing Application	178
Partially Multiplexed Addressing Mode	171
PIC18F6XJ1X (64-Pin)	10
PIC18F8XJ1X (80-Pin)	11
PLL	36
PMP Module	157
PWM Operation (Simplified)	204
Reads From Flash Program Memory	91
Table Read Operation	87
Table Write Operation	88
Table Writes to Flash Program Memory	93
Timer0 in 16-Bit Mode	182
Timer0 in 8-Bit Mode	182
Timer1	186
Timer1 (16-Bit Read/Write Mode)	186
Timer2	192
Timer3	194
Timer3 (16-Bit Read/Write Mode)	194
Timer4	198
Watchdog Timer	321
BN	338
BNC	339
BNN	339
BNOV	340
BNZ	340
BOR. See Brown-out Reset.	
BOV	343
BRA	341
Break Character (12-Bit) Transmit and Receive	284
BRG. See Baud Rate Generator.	

PIC18F87J11 FAMILY

Brown-out Reset (BOR)	51	Single-Word Write to Flash	
and On-Chip Voltage Regulator	324	Program Memory	95
Detecting	51	Writing to Flash Program Memory	94
Disabling in Sleep Mode	51	Code Protection	313
BSF	341	COMF	346
BTFSC	342	Comparator	301
BTFSS	342	Analog Input Connection Considerations	304
BTG	343	Associated Registers	308
BZ	344	Configuration	305
C		Control	305
C Compilers		Effects of a Reset	308
MPLAB C18	380	Enable, Input Selection	305
MPLAB C30	380	Enable, Output Selection	305
Calibration (A/D Converter)	299	Interrupts	307
CALL	344	Operation	304
CALLW	373	Operation During Sleep	308
Capture (CCP Module)	201	Reference	
Associated Registers	203	Response Time	304
CCP Pin Configuration	201	Single Comparator	304
CCPRxH:CCPRxL Registers	201	Comparator Specifications	397
Prescaler	201	Comparator Voltage Reference	309
Software Interrupt	201	Accuracy and Error	311
Timer1/Timer3 Mode Selection	201	Associated Registers	311
Capture (ECCP Module)	210	Configuring	310
Capture/Compare/PWM (CCP)	199	Connection Considerations	311
Capture Mode. <i>See</i> Capture.		Effects of a Reset	311
CCP Mode and Timer Resources	200	Operation During Sleep	311
CCPRxH Register	200	Compare (CCP Module)	202
CCPRxL Register	200	Associated Registers	203
Compare Mode. <i>See</i> Compare.		CCPRx Register	202
Module Configuration	200	Pin Configuration	202
Timer Interconnect Configurations	200	Software Interrupt	202
Clock Sources	33	Timer1/Timer3 Mode Selection	202
Default System Clock on Reset	34	Compare (ECCP Module)	210
Selection Using OSCCON Register	34	Special Event Trigger	210, 298
CLRF	345	Compare (ECCPx Modules)	
CLRWD	345	Special Event Trigger	195
Code Examples		Computed GOTO	67
16 x 16 Signed Multiply Routine	110	Configuration Bits	313
16 x 16 Unsigned Multiply Routine	110	Configuration Mismatch Reset (CM)	51
8 x 8 Signed Multiply Routine	109	Configuration Register Protection	327
8 x 8 Unsigned Multiply Routine	109	Core Features	
A/D Calibration Routine	299	Easy Migration	8
Changing Between Capture Prescalers	201	Expanded Memory	7
Computed GOTO Using an Offset Value	67	Extended Instruction Set	7
Erasing a Flash Program Memory Row	92	External Memory Bus	7
Fast Register Stack	67	nanoWatt Technology	7
How to Clear RAM (Bank 1) Using		Oscillator Options and Features	7
Indirect Addressing	81	CPFSEQ	346
Implementing a Real-Time Clock Using		CPFGT	347
a Timer1 Interrupt Service	189	CPFSLT	347
Initializing PORTA	130	Crystal Oscillator/Ceramic Resonator	35
Initializing PORTB	133	Customer Change Notification Service	431
Initializing PORTC	136	Customer Notification Service	431
Initializing PORTD	139	Customer Support	431
Initializing PORTE	142	D	
Initializing PORTF	145	Data Addressing Modes	81
Initializing PORTG	148	Comparing Addressing Modes with the	
Initializing PORTH	151	Extended Instruction Set Enabled	85
Initializing PORTJ	154	Direct	81
Loading the SSP1BUF (SSP1SR) Register	226	Indexed Literal Offset	84
Reading a Flash Program Memory Word	91	BSR	86
Saving STATUS, WREG and BSR		Instructions Affected	84
Registers in RAM	126	Mapping Access Bank	86

Indirect	81	EUSART	
Inherent and Literal	81	Asynchronous Mode	279
Data Memory	70	12-Bit Break Transmit and Receive	284
Access Bank	72	Associated Registers, Receive	282
Bank Select Register (BSR).....	70	Associated Registers, Transmit	280
Extended Instruction Set.....	84	Auto-Wake-up on Sync Break	282
General Purpose Registers.....	72	Receiver	281
Memory Map	71	Setting Up 9-Bit Mode with Address Detect	281
Memory Maps		Transmitter	279
Special Function Registers	73	Baud Rate Generator	
Special Function Registers	73	Operation in Power-Managed Mode.....	273
Context Defined SFRs	74	Baud Rate Generator (BRG)	273
Shared Address	74	Associated Registers.....	274
DAW	348	Auto-Baud Rate Detect.....	277
DC and AC Characteristics		Baud Rate Error, Calculating	274
Graphs and Tables	423	Baud Rates, Asynchronous Modes	275
DC Characteristics	394	High Baud Rate Select (BRGH Bit)	273
Power-Down and Supply Current	386	Sampling	273
Supply Voltage	385	Synchronous Master Mode.....	285
DCFSNZ	349	Associated Registers, Receive	287
DECFSZ	348	Associated Registers, Transmit	286
DECF	349	Reception	287
Default System Clock.....	34	Transmission	285
Development Support	379	Synchronous Slave Mode.....	288
Device Differences	429	Associated Registers, Receive	289
Device Overview	7	Associated Registers, Transmit	288
Details on Individual Family Members	8	Reception	289
Features (64-Pin Devices)	9	Transmission	288
Features (80-Pin Devices)	9	Extended Instruction Set	
Direct Addressing.....	82	ADDFSR	372
E		ADDULNK	372
ECCP		CALLW	373
Associated Registers	222	MOVSF	373
Capture and Compare Modes.....	210	MOVSS	374
Enhanced PWM Mode	211	PUSHL	374
Standard PWM Mode.....	210	SUBFSR	375
Effect on Standard PIC Instructions	376	SUBULNK	375
Effects of Power-Managed Modes on		External Memory Bus	97
Various Clock Sources.....	40	16-Bit Byte Select Mode	103
Electrical Characteristics	383	16-Bit Byte Write Mode	101
Enhanced Capture/Compare/PWM (ECCP)	207	16-Bit Data Width Modes	100
Capture Mode. <i>See</i> Capture (ECCP Module).		16-Bit Mode Timing	104
ECCP1/ECCP3 Outputs and		16-Bit Word Write Mode	102
Program Memory Mode	208	8-Bit Data Width Mode	105
ECCP2 Outputs and Program Memory Modes	208	8-Bit Mode Timing	106
Outputs and Configuration	208	Address and Data Line Usage (table)	99
Pin Configurations for ECCP1	209	Address and Data Width	99
Pin Configurations for ECCP2	209	Address Shifting	99
Pin Configurations for ECCP3	210	Control	98
PWM Mode. <i>See</i> PWM (ECCP Module).		I/O Port Functions	97
Timer Resources	208	Operation in Power-Managed Modes	107
Use of CCP4/CCP5 with ECCP1/ECCP3	208	Program Memory Modes	100
Enhanced Universal Synchronous Asynchronous		Extended Microcontroller	100
Receiver Transmitter (EUSART). <i>See</i> EUSART.		Microcontroller	100
ENVREG pin	323	Wait States	100
Equations		Weak Pull-ups on Port Pins	100
A/D Acquisition Time.....	296	External Oscillator Modes	
A/D Minimum Charging Time	296	Clock Input (EC Modes)	36
Calculating the Minimum Required		HS	35
Acquisition Time	296		
Errata	5		

PIC18F87J11 FAMILY

F

Fail-Safe Clock Monitor.....	313, 325
Exiting	326
Interrupts in Power-Managed Modes.....	326
POR or Wake-up From Sleep	326
WDT During Oscillator Failure	325
Fast Register Stack.....	67
Firmware Instructions.....	329
Flash Configuration Words.....	313
Flash Program Memory.....	87
Associated Registers	96
Control Registers	88
EECON1 and EECON2	88
TABLAT (Table Latch) Register	90
TBLPTR (Table Pointer) Register	90
Erase Sequence	92
Erasing.....	92
Operation During Code-Protect	96
Reading.....	91
Table Pointer	
Boundaries Based on Operation.....	90
Table Pointer Boundaries	90
Table Reads and Table Writes	87
Write Sequence	93
Write Sequence (Word Programming)	95
Writing.....	93
Unexpected Termination.....	96
Write Verify	96
FSCM. See Fail-Safe Clock Monitor.	

G

GOTO.....	350
-----------	-----

H

Hardware Multiplier	109
8 x 8 Multiplication Algorithms	109
Operation	109
Performance Comparison (table).....	109

I

I/O Ports	127
Input Pull-up Configuration	128
Open-Drain Outputs.....	128
Pin Capabilities	127
I ² C Mode (MSSP)	
Acknowledge Sequence Timing.....	262
Associated Registers	268
Baud Rate Generator.....	255
Bus Collision	
During a Repeated Start Condition	266
During a Stop Condition	267
Clock Arbitration.....	256
Clock Stretching.....	248
10-Bit Slave Receive Mode (SEN = 1).....	248
10-Bit Slave Transmit Mode.....	248
7-Bit Slave Receive Mode (SEN = 1).....	248
7-Bit Slave Transmit Mode.....	248
Clock Synchronization and the CKP bit	249
Effects of a Reset.....	263
General Call Address Support	252
I ² C Clock Rate w/BRG	255

Master Mode.....	253
Operation.....	254
Reception	259
Repeated Start Condition Timing	258
Start Condition Timing	257
Transmission	259
Multi-Master Communication, Bus Collision and Arbitration	263
Multi-Master Mode	263
Operation.....	238
Read/Write Bit Information (R/W Bit)	238, 241
Registers	233
Serial Clock (RC3/SCKx/SCLx)	241
Slave Mode.....	238
Address Masking Modes	
5-Bit	239
7-Bit	240
Addressing.....	238
Reception	241
Transmission	241
Sleep Operation.....	263
Stop Condition Timing	262
INCF	350
INCFSZ.....	351
In-Circuit Debugger.....	327
In-Circuit Serial Programming (ICSP).....	313, 327
Indexed Literal Offset Addressing	
and Standard PIC18 Instructions.....	376
Indexed Literal Offset Mode.....	376
Indirect Addressing	82
INFSNZ.....	351
Initialization Conditions for all Registers	55–60
Instruction Cycle	68
Clocking Scheme.....	68
Flow/Pipelining.....	68
Instruction Set.....	329
ADDLW	335
ADDWF	335
ADDWF (Indexed Literal Offset Mode)	377
ADDWFC	336
ANDLW	336
ANDWF	337
BC	337
BCF	338
BN	338
BNC	339
BNN	339
BNOV	340
BNZ	340
BOV	343
BRA	341
BSF	341
BSF (Indexed Literal Offset Mode)	377
BTFSC	342
BTFSS	342
BTG	343
BZ	344
CALL	344
CLRF	345
CLRWD	345
COMF	346
CPFSEQ	346

CPFSGT	347
CPFSLT	347
DAW.....	348
DCFSNZ	349
DEC F	348
DECFSZ.....	349
Extended Instructions	371
Considerations when Enabling	376
Syntax	371
Use with MPLAB IDE Tools	378
General Format.....	331
GOTO	350
INCF.....	350
INCFSZ.....	351
INFSNZ.....	351
IORLW	352
IORWF	352
LFSR.....	353
MOV F.....	353
MOVFF	354
MOVLB	354
MOV LW	355
MOV WF	355
MULLW.....	356
MULWF.....	356
NEGF	357
NOP	357
Opcode Field Descriptions.....	330
POP	358
PUSH	358
RCALL	359
RESET	359
RETFIE	360
RETLW	360
RETURN	361
RLCF.....	361
RLNCF	362
RRCF	362
RRNCF	363
SETF	363
SETF (Indexed Literal Offset Mode)	377
SLEEP	364
Standard Instructions	329
SUBFWB.....	364
SUBLW	365
SUBWF	365
SUBWFB.....	366
SWAPF	366
TBLRD	367
TBLWT	368
TSTFSZ	369
XORLW	369
XORWF	370
INTCON Register	113
RBIF Bit.....	133
INTCON Registers	113
Inter-Integrated Circuit. <i>See</i> I ² C.	
Internal Oscillator Block	37
Adjustment.....	38
INTIO Modes.....	37
INTOSC Frequency Drift.....	38
INTOSC Output Frequency.....	38
INTPLL Modes	37
Internal RC Block	
Use with WDT	321
Internal Voltage Reference Specifications.....	397
Internet Address	431
Interrupt Sources	313
A/D Conversion Complete	295
Capture Complete (CCP)	201
Compare Complete (CCP)	202
Interrupt-on-Change (RB7:RB4).....	133
TMR0 Overflow.....	183
TMR2 to PR2 Match (PWM).....	211
TMR3 Overflow.....	193, 195
TMR4 to PR4 Match	198
TMR4 to PR4 Match (PWM).....	197
Interrupts	111
During, Context Saving.....	126
INTx Pin.....	126
PORTB, Interrupt-on-Change.....	126
TMR0	126
Interrupts, Flag Bits	
Interrupt-on-Change (RB7:RB4) Flag (RBIF Bit).....	133
INTOSC, INTRC. <i>See</i> Internal Oscillator Block.	
IORLW	352
IORWF	352
IPR Registers.....	122
L	
LFSR	353
M	
Master Clear (MCLR).....	51
Master Synchronous Serial Port (MSSP). <i>See</i> MSSP.	
Memory Organization	61
Data Memory	70
Program Memory.....	61
Memory Programming Requirements.....	396
Microchip Internet Web Site.....	431
MOV F.....	353
MOVFF	354
MOVLB	354
MOV LW	355
MOVSF	373
MOVSS.....	374
MOVWF	355
MPLAB ASM30 Assembler, Linker, Librarian	380
MPLAB ICD 2 In-Circuit Debugger	381
MPLAB ICE 2000 High-Performance Universal In-Circuit Emulator.....	381
MPLAB Integrated Development Environment Software	379
MPLAB PM3 Device Programmer	381
MPLAB REAL ICE In-Circuit Emulator System	381
MPLINK Object Linker/MPLIB Object Librarian	380
MSSP	
ACK Pulse	238, 241
I ² C Mode. <i>See</i> I ² C Mode.	
Module Overview	223
SPI Master/Slave Connection.....	227
MULLW.....	356
MULWF.....	356
N	
NEGF	357
NOP	357

PIC18F87J11 FAMILY

O

Open-Drain Outputs	128
Oscillator Configuration	31
EC	31
ECPLL	31
HS	31
HSPLL	31
Internal Oscillator Block	37
INTIO1	31
INTIO2	31
INTPLL1	31
INTPLL2	31
Oscillator Selection	313
Oscillator Start-up Timer (OST)	40
Oscillator Switching	33
Oscillator Transitions	34
Oscillator, Timer1	185, 195
Oscillator, Timer3	193

P

Packaging	425
Details	426
Marking	425
Parallel Master Port (PMP)	157
Application Examples	178
Associated Registers	180
Control Registers	158
Data Registers	164
Master Port Modes	170
Slave Port Modes	165
PICSTART Plus Development Programmer	382
PIE Registers	119
Pin Functions	
AVDD	29
AVDD	19
AVss	29
AVss	19
ENVREG	19, 29
MCLR	12, 20
OSC1/CLK1/RA7	12, 20
OSC2/CLK0/RA6	12, 20
RA0/AN0	13, 21
RA1/AN1	13, 21
RA2/AN2/VREF-	13, 21
RA3/AN3/VREF+	13, 21
RA4/PMD5/T0CKI	21
RA4/T0CKI	13
RA5/AN4	13
RA5/PMD4/AN4	21
RA6	13, 21
RA7	13, 21
RB0/FLT0/INT0	14, 22
RB1/INT1/PMA4	14, 22
RB2/INT2/PMA3	14, 22
RB3/INT3//PMA2/ECCP2/P2A	22
RB3/INT3/PMA2	14
RB4/KB10/PMA1	14, 22
RB5/KB11/PMA0	14, 22
RB6/KB12/PGC	14, 22
RB7/KB13/PGD	14, 22
RC0/T1OSO/T13CKI	15, 23
RC1/T1OSI/ECCP1/P2A	15, 23
RC2/ECCP1/P1A	15, 23
RC3/SCK1/SCL1	15, 23
RC4/SDI1/SDA1	15, 23

RC5/SDO1	15, 23
RC6/TX1/CK1	15, 23
RC7/RX1/DT1	15, 23
RD0/AD0/PMD0	24
RD0/PMD0	16
RD1/AD1/PMD1	24
RD1/PMD1	16
RD2/AD2/PMD2	24
RD2/PMD2	16
RD3/AD3/PMD3	24
RD3/PMD3	16
RD4/AD4/PMD4/SDO2	24
RD4/PMD4/SDO2	16
RD5/AD5/PMD5/SDI2/SDA2	24
RD5/PMD5/SDI2/SDA2	16
RD6/AD6/PMD6/SCK2/SCL2	24
RD6/PMD6/SCK2/SCL2	16
RD7/AD7/PMD7/SS2	24
RD7/PMD7/SS2	16
RE0/AD8/PMRD/P2D	25
RE0/PMRD/P2D	17
RE1/AD9/PMWR/P2C	25
RE1/PMWR/P2C	17
RE2/AD10/PMBE/P2B	25
RE2/PMBE/P2B	17
RE3/AD11/PMA13/P3C/REFO	25
RE3/PMA13/P3C/REFO	17
RE4/AD12/PMA12/P3B	25
RE4/PMA12/P3B	17
RE5/AD13/PMA11/P1C	25
RE5/PMA11/P1C	17
RE6/AD14/PMA10/P1B	25
RE6/PMA10/P1B	17
RE7/AD15/PMA9/ECCP2/P2A	25
RE7/PMA9/ECCP2/P2A	17
RF1/AN6/C2OUT	18, 26
RF2/PMA5/AN7/C1OUT	18, 26
RF3/AN8/C2INB	18, 26
RF4/AN9/C2INA	18, 26
RF5/AN10/C1INB/CVREF	18
RF5/PMD2/AN10/C1INB/CVREF	26
RF6/AN11/C1INA	18
RF6/PMD1/AN11/C1INA	26
RF7/PMD0/SS1	26
RF7/SS1	18
RG0/PMA8/ECCP3/P3A	19, 27
RG1/PMA7/TX2/CK2	19, 27
RG2/PMA6/RX2/DT2	19, 27
RG3/PMCS1/CCP4/P3D	19, 27
RG4/PMCS2/CCP5/P1D	19, 27
RH0/A16	28
RH1/A17	28
RH2/A18/PMD7	28
RH3/A19/PMD6	28
RH4/PMD3/AN12/P3C/C2INC	28
RH5/PMBE/AN13/P3B/C2IND	28
RH6/PMRD/AN14/P1C/C1INC	28
RH7/PMWR/AN15/P1B	28
RJ0/ALE	29
RJ1/OE	29
RJ2/WRL	29
RJ3/WRH	29
RJ4/BA0	29
RJ5/CE	29
RJ6/LB	29

RJ7/UB.....	29
VDD	29
VDD	19
VDDCORE/VCAP.....	19, 29
Vss.....	29
Vss.....	19
Pinout I/O Descriptions	
PIC18F6XJ1X (64-TQFP)	12
PIC18F8XJ1X (80-TQFP)	20
PIR Registers	116
PLL.....	36
HSPLL and ECPLL Oscillator Modes	36
Use with INTOSC.....	36
POP	358
POR. <i>See</i> Power-on Reset.	
PORTA	
Associated Registers	132
LATA Register.....	130
PORTA Register	130
TRISA Register	130
PORTB	
Associated Registers	135
LATB Register.....	133
PORTB Register	133
RB7:RB4 Interrupt-on-Change Flag (RBIF Bit)	133
TRISB Register.....	133
PORTC	
Associated Registers	138
LATC Register	136
PORTC Register	136
RC3/SCKx/SCLx Pin.....	241
TRISC Register.....	136
PORTD	
Associated Registers	141
LATD Register	139
PORTD Register	139
TRISD Register.....	139
PORTE	
Associated Registers	144
LATE Register.....	142
PORTE Register	142
TRISE Register	142
PORTF	
Associated Registers	147
LATF Register.....	145
PORTF Register	145
TRISF Register	145
PORTG	
Associated Registers	150
LATG Register	148
PORTG Register	148
TRISG Register.....	148
PORTH	
Associated Registers	153
LATH Register	151
PORTH Register	151
TRISH Register.....	151
PORTJ	
Associated Registers	155
LATJ Register	154
PORTJ Register	154
TRISJ Register.....	154
Power-Managed Modes.....	41
and EUSART Operation	273
and SPI Operation	231
Clock Sources	41
Clock Transitions and Status Indicators	42
Entering	41
Exiting Idle and Sleep Modes	47
By Interrupt.....	47
By Reset.....	47
By WDT Time-out.....	47
Without an Oscillator Start-up Delay	47
Idle Modes	45
PRI_IDLE	46
RC_IDLE	47
SEC_IDLE	46
Multiple Sleep Commands.....	42
Run Modes	42
PRI_RUN.....	42
RC_RUN.....	44
SEC_RUN	42
Selecting.....	41
Sleep Mode	45
OSC1 and OSC2 Pin States.....	40
Summary (table)	41
Power-on Reset (POR).....	51
Power-up Delays	40
Power-up Timer (PWRT)	40, 52
Time-out Sequence	52
Prescaler	
Timer2	212
Prescaler, Timer0	183
Prescaler, Timer2 (Timer4).....	205
PRI_IDLE Mode.....	46
PRI_RUN Mode	42
Program Counter	65
PCL, PCH and PCU Registers	65
PCLATH and PCLATU Registers	65
Program Memory	
ALU	
Status	80
Extended Instruction Set	84
Flash Configuration Words	62
Hard Memory Vectors.....	62
Instructions	69
Two-Word	69
Interrupt Vector.....	62
Look-up Tables.....	67
Memory Maps	61
Hard Vectors and Configuration Words.....	62
Modes.....	64
Modes.....	63
Extended Microcontroller	63
Extended Microcontroller (Address Shifting).....	64
Memory Access (table).....	64
Microcontroller.....	63
Reset Vector	62
Program Verification and Code Protection	327
Programming, Device Instructions	329
Pull-up Configuration	128
Pulse-Width Modulation. <i>See</i> PWM (CCP Module) and PWM (ECCP Module).	
PUSH.....	358

PIC18F87J11 FAMILY

PUSH and POP Instructions	66
PUSHL	374
PWM (CCP Module)	
Associated Registers	206
Duty Cycle.....	204
Example Frequencies/Resolutions	205
Operation Setup	205
Period.....	204
PR2/PR4 Registers.....	204
TMR2 (TMR4) to PR2 (PR4) Match.....	204
TMR2 to PR2 Match	211
TMR4 to PR4 Match	197
PWM (ECCP Module)	211
CCPR1H:CCPR1L Registers.....	211
Direction Change in Full-Bridge	
Output Mode	216
Duty Cycle.....	212
Effects of a Reset.....	221
Enhanced PWM Auto-Shutdown	218
Example Frequencies/Resolutions	212
Full-Bridge Mode.....	215
Full-Bridge Output Application Example	216
Half-Bridge Mode	214
Half-Bridge Output Mode	
Applications Example.....	214
Output Configurations	212
Output Relationships (Active-High).....	213
Output Relationships (Active-Low).....	213
Period.....	211
Programmable Dead-Band Delay	218
Setup for PWM Operation.....	221
Start-up Considerations	219
Q	
Q Clock	205, 212
R	
RAM. See Data Memory.	
RC_IDLE Mode	47
RC_RUN Mode	44
RCALL.....	359
RCON Register	
Bit Status During Initialization	54
Reader Response	432
Reference Clock Output.....	38
Register File	72
Register File Summary.....	75–79
Registers	
ADCON0 (A/D Control 0)	291
ADCON0 (A/D Control 1)	292
ANCON0 (A/D Port Configuration 2).....	293
ANCON1 (A/D Port Configuration 1).....	293
BAUDCONx (Baud Rate Control)	272
CCPxCON (Capture/Compare/PWM	
Control)	199
CCPxCON (ECCPx Control)	207
CMSTAT (Comparator Output Status)	303
CMxCON (Comparatortx Control).....	302
CONFIG1H (Configuration 1 High)	315
CONFIG1L (Configuration 1 Low)	315
CONFIG2H (Configuration 2 High)	317
CONFIG3H (Configuration 3 High)	319
CONFIG3L (Configuration 3 Low).....	63, 318
CVRCON (Comparator Voltage	
Reference Control).....	310
DEVID1 (Device ID 1)	320
DEVID2 (Device ID 2).....	320
ECCPxAS (ECCPx Auto-Shutdown Control).....	219
ECCPxDEL (ECCPx PWM Delay).....	218
EECON1 (EEPROM Control 1)	89
INTCON (Interrupt Control).....	113
INTCON2 (Interrupt Control 2).....	114
INTCON3 (Interrupt Control 3).....	115
IPR1 (Peripheral Interrupt Priority 1)	122
IPR2 (Peripheral Interrupt Priority 2)	123
IPR3 (Peripheral Interrupt Priority 3)	124
MEMCON (External Memory Bus Control)	98
ODCON1 (Peripheral Open-Drain Control 1)	129
ODCON2 (Peripheral Open-Drain Control 2)	129
ODCON3 (Peripheral Open-Drain Control 3)	129
OSCCON (Oscillator Control)	32
OSCTUNE (Oscillator Tuning)	33
PADCFG1 (I/O Pad Configuration Control)	130
PIE1 (Peripheral Interrupt Enable 1).....	119
PIE2 (Peripheral Interrupt Enable 2).....	120
PIE3 (Peripheral Interrupt Enable 3).....	121
PIR1 (Peripheral Interrupt Request (Flag) 1)	116
PIR2 (Peripheral Interrupt Request (Flag) 2)	117
PIR3 (Peripheral Interrupt Request (Flag) 3)	118
PMADDRH (Parallel Port Address High	
Byte, Master Mode Only)	164
PMCONH (Parallel Port Control High Byte)	158
PMCONL (Parallel Port Control Low Byte)	159
PMEH (Parallel Port Enable High Byte)	161
PMEL (Parallel Port Enable Low Byte)	162
PMMODEH (Parallel Port Mode High Byte)	160
PMMODEL (Parallel Port Mode Low Byte)	161
PMSTAT (Parallel Port Status High Byte)	162
PMSTAT (Parallel Port Status Low Byte)	163
RCON (Reset Control)	50, 125
RCSTAx (Receive Status and Control)	271
REFOCON (Reference Oscillator Control)	39
SSPCON2 (MSSPx Control 2, I ² C	
Master Mode)	236
SSPCON2 (MSSPx Control 2, I ² C	
Slave Mode)	237
SSPxCON1 (MSSPx Control 1, I ² C Mode)	235
SSPxCON1 (MSSPx Control 1, SPI Mode)	225
SSPxMSK (I ² C Slave Address Mask)	237
SSPxSTAT (MSSPx Status, I ² C Mode)	234
SSPxSTAT (MSSPx Status, SPI Mode)	224
STATUS	80
STKPTR (Stack Pointer)	66
T0CON (Timer0 Control)	181
T1CON (Timer1 Control)	185
T2CON (Timer2 Control)	191
T3CON (Timer3 Control)	193
T4CON (Timer4 Control)	197
TXSTAx (Transmit Status and Control)	270
WDTCON (Watchdog Timer Control)	74, 322
RESET	359
Reset	49
Brown-out Reset (BOR)	49
Configuration Mismatch (CM)	49
MCLR Reset, During Power-Managed Modes	49
MCLR Reset, Normal Operation	49
Power-on Reset (POR)	49
RESET Instruction	49
Stack Full Reset	49
Stack Underflow Reset	49
Watchdog Timer (WDT) Reset	49

Resets	313
Brown-out Reset (BOR)	313
Oscillator Start-up Timer (OST)	313
Power-on Reset (POR)	313
Power-up Timer (PWRT)	313
RETFIE	360
RETLW	360
RETURN	361
Revision History	429
RLCF	361
RLNCF	362
RRCF	362
RRNCF	363
S	
SCKx	223
SDIx	223
SDOx	223
SEC_IDLE Mode	46
SEC_RUN Mode	42
Serial Clock, SCKx	223
Serial Data In (SDIx)	223
Serial Data Out (SDOx)	223
Serial Peripheral Interface. <i>See SPI Mode.</i>	
SETF	363
Slave Select (\overline{SS}_x)	223
SLEEP	364
Software Simulator (MPLAB SIM)	380
Special Event Trigger. <i>See Compare (ECCP Module).</i>	
Special Features of the CPU	313
Special Function Registers	
Shared Registers	74
SPI Mode (MSSP)	223
Associated Registers	232
Bus Mode Compatibility	231
Clock Speed, Interactions	231
Effects of a Reset	231
Enabling SPI I/O	227
Master Mode	228
Master/Slave Connection	227
Operation	226
Operation in Power-Managed Modes	231
Serial Clock	223
Serial Data In	223
Serial Data Out	223
Slave Mode	229
Slave Select	223
Slave Select Synchronization	229
SPI Clock	228
SSPxBUF Register	228
SSPxSR Register	228
Typical Connection	227
SSPOV	259
SSPOV Status Flag	259
SSPxSTAT Register	
R/W Bit	238, 241
SSx	223
Stack Full/Underflow Resets	67
SUBFSR	375
SUBFWB	364
SUBLW	365
SUBLNK	375
SUBWF	365
SUBWFB	366
SWAPF	366

T	
Table Pointer Operations (table)	90
Table Reads/Table Writes	67
TBLRD	367
TBLWT	368
Timer0	181
Associated Registers	183
Operation	182
Overflow Interrupt	183
Prescaler	183
Switching Assignment	183
Prescaler Assignment (PSA Bit)	183
Prescaler Select (TOPS2:T0PS0 Bits)	183
Prescaler. <i>See Prescaler, Timer0.</i>	
Reads and Writes in 16-Bit Mode	182
Source Edge Select (T0SE Bit)	182
Source Select (T0CS Bit)	182
Timer1	185
16-Bit Read/Write Mode	187
Associated Registers	189
Considerations in Asynchronous	
Counter Mode	188
Interrupt	188
Operation	186
Oscillator	185, 187
Layout Considerations	187
Oscillator, as Secondary Clock	33
Resetting, Using the ECCPx	
Special Event Trigger	188
Special Event Trigger (ECCP)	210
TMR1H Register	185
TMR1L Register	185
Use as a Clock Source	187
Use as a Real-Time Clock	188
Timer2	191
Associated Registers	192
Interrupt	192
Operation	191
Output	192
PR2 Register	211
TMR2 to PR2 Match Interrupt	211
Timer3	193
16-Bit Read/Write Mode	195
Associated Registers	195
Operation	194
Oscillator	193, 195
Overflow Interrupt	193, 195
Special Event Trigger (ECCPx)	195
TMR3H Register	193
TMR3L Register	193
Timer4	197
Associated Registers	198
Operation	197
Output	198
Postscaler. <i>See Postscaler, Timer4.</i>	
PR4 Register	197
Prescaler. <i>See Prescaler, Timer4.</i>	
TMR4 Register	197
TMR4 to PR4 Match Interrupt	197, 198

PIC18F87J11 FAMILY

Timing Diagrams	
A/D Conversion	420
Asynchronous Reception	282
Asynchronous Transmission	280
Asynchronous Transmission (Back to Back)	280
Automatic Baud Rate Calculation	278
Auto-Wake-up Bit (WUE) During	
Normal Operation	283
Auto-Wake-up Bit (WUE) During Sleep	283
Baud Rate Generator with Clock Arbitration	256
BRG Overflow Sequence	278
BRG Reset Due to SDAX Arbitration During	
Start Condition	265
Bus Collision During a Repeated Start Condition (Case 1)	266
Bus Collision During a Repeated Start Condition (Case 2)	266
Bus Collision During a Start Condition (SCLx = 0)	265
Bus Collision During a Stop Condition (Case 1)	267
Bus Collision During a Stop Condition (Case 2)	267
Bus Collision During Start Condition (SDAX Only)	264
Bus Collision for Transmit and Acknowledge	263
Capture/Compare/PWM (Including	
ECCP Modules)	410
CLKO and I/O	402
Clock Synchronization	249
Clock/Instruction Cycle	68
EUSART Synchronous Receive	
(Master/Slave)	419
EUSART Synchronous Transmission	
(Master/Slave)	419
Example SPI Master Mode (CKE = 0)	411
Example SPI Master Mode (CKE = 1)	412
Example SPI Slave Mode (CKE = 0)	413
Example SPI Slave Mode (CKE = 1)	414
External Clock (All Modes Except PLL)	400
External Memory Bus for Sleep (Extended Microcontroller Mode)	104, 106
External Memory Bus for TBLRD (Extended Microcontroller Mode)	104, 106
Fail-Safe Clock Monitor	326
First Start Bit Timing	257
Full-Bridge PWM Output	215
Half-Bridge PWM Output	214
I ² C Acknowledge Sequence	262
I ² C Bus Data	415
I ² C Bus Start/Stop Bits	415
I ² C Master Mode (7 or 10-Bit Transmission)	260
I ² C Master Mode (7-Bit Reception)	261
I ² C Slave Mode (10-Bit Reception, SEN = 0)	246
I ² C Slave Mode (10-Bit Reception, SEN = 0, ADMSK = 01001)	245
I ² C Slave Mode (10-Bit Reception, SEN = 1)	251
I ² C Slave Mode (10-Bit Transmission)	247
I ² C Slave Mode (7-Bit Reception, SEN = 0)	242
I ² C Slave Mode (7-bit Reception, SEN = 0, ADMSK = 01011)	243
I ² C Slave Mode (7-Bit Reception, SEN = 1)	250
I ² C Slave Mode (7-Bit Transmission)	244
I ² C Slave Mode General Call Address Sequence (7 or 10-Bit Addressing Mode)	252
I ² C Stop Condition Receive or Transmit Mode	262
MSSP I ² C Bus Data	417
MSSP I ² C Bus Start/Stop Bits	417
Parallel Master Port Read	408
Parallel Master Port Write	409
Parallel Slave Port	407
Parallel Slave Port Read	166, 169
Parallel Slave Port Write	166, 169
Program Memory Read	403
Program Memory Write	404
PWM Auto-Shutdown (P1RSEN = 0, Auto-Restart Disabled)	220
PWM Auto-Shutdown (P1RSEN = 1, Auto-Restart Enabled)	220
PWM Direction Change	217
PWM Direction Change at Near 100% Duty Cycle	217
PWM Output	204
Read and Write, 8-Bit Data, Demultiplexed Address	173
Read, 16-Bit Data, Demultiplexed Address	176
Read, 16-Bit Multiplexed Data, Fully Multiplexed 16-Bit Address	177
Read, 16-Bit Multiplexed Data, Partially Multiplexed Address	176
Read, 8-Bit Data, Fully Multiplexed 16-Bit Address	175
Read, 8-Bit Data, Partially Multiplexed Address	173
Read, 8-Bit Data, Partially Multiplexed Address, Enable Strobe	174
Read, 8-Bit Data, Wait States Enabled, Partially Multiplexed Address	173
Repeated Start Condition	258
Reset, Watchdog Timer (WDT), Oscillator Start-up Timer (OST) and Power-up Timer (PWRT)	405
Send Break Character Sequence	284
Slave Synchronization	229
Slow Rise Time (MCLR Tied to VDD, VDD Rise > TPWRT)	53
SPI Mode (Master Mode)	228
SPI Mode (Slave Mode, CKE = 0)	230
SPI Mode (Slave Mode, CKE = 1)	230
Synchronous Reception (Master Mode, SREN)	287
Synchronous Transmission	285
Synchronous Transmission (Through TXEN)	286
Time-out Sequence on Power-up (MCLR Not Tied to VDD), Case 1	52
Time-out Sequence on Power-up (MCLR Not Tied to VDD), Case 2	53
Time-out Sequence on Power-up (MCLR Tied to VDD, VDD Rise < TPWRT)	52
Timer0 and Timer1 External Clock	406
Transition for Entry to Idle Mode	46
Transition for Entry to SEC_RUN Mode	43
Transition for Entry to Sleep Mode	45
Transition for Two-Speed Start-up (INTRC to HSPLL)	324
Transition for Wake From Idle to Run Mode	46
Transition for Wake From Sleep (HSPLL)	45
Transition From RC_RUN Mode to PRI_RUN Mode	44
Transition From SEC_RUN Mode to PRI_RUN Mode (HSPLL)	43
Transition to RC_RUN Mode	44

Write, 16-Bit Multiplexed Data, Fully Multiplexed 16-Bit Address	177
Write, 16-Bit Multiplexed Data, Partially Multiplexed Address	177
Write, 8-Bit Data, Demultiplexed Address.....	176
Write, 8-Bit Data, Fully Multiplexed 16-Bit Address	175
Write, 8-Bit Data, Partially Multiplexed Address	174
Write, 8-Bit Data, Partially Multiplexed Address, Enable Strobe	175
Write, 8-Bit Data, Wait States Enabled, Partially Multiplexed Address.....	174
Timing Diagrams and Specifications	
Capture/Compare/PWM Requirements (Including ECCP Modules)	410
CLKO and I/O Requirements	402, 403
EUSART Synchronous Receive Requirements	419
EUSART Synchronous Transmission Requirements	419
Example SPI Mode Requirements (Master Mode, CKE = 0)	411
Example SPI Mode Requirements (Master Mode, CKE = 1)	412
Example SPI Mode Requirements (Slave Mode, CKE = 0)	413
Example SPI Slave Mode Requirements (CKE = 1)	414
External Clock Requirements	400
I ² C Bus Data Requirements (Slave Mode)	416
I ² C Bus Start/Stop Bits Requirements (Slave Mode)	415
Internal RC Accuracy (INTOSC, INTRC Sources)	401
MSSP I ² C Bus Data Requirements	418
MSSP I ² C Bus Start/Stop Bits Requirements	417
Parallel Master Port Read Requirements	408
Parallel Master Port Write	409
Parallel Slave Port Requirements	407
PLL Clock.....	401
Program Memory Write Requirements	404
Reset, Watchdog Timer (WDT), Oscillator Start-up Timer (OST), Power-up Timer (PWRT) and Brown-out Reset	405
Timer0 and Timer1 External Clock Requirements	406
TSTFSZ	369
Two-Speed Start-up.....	313, 324
Two-Word Instructions	
Example Cases.....	69
TXSTAx Register	
BRGH Bit	273

V

VDDCORE/VCAP Pin	323
Voltage Reference Specifications.....	397
Voltage Regulator (On-Chip)	323
Operation in Sleep Mode.....	324
Power-up Requirements.....	324

W

Watchdog Timer (WDT).....	313, 321
Associated Registers.....	322
Control Register.....	321
During Oscillator Failure	325
Programming Considerations	321
WCOL	257, 258, 259, 262
WCOL Status Flag.....	257, 258, 259, 262
WWW Address	431
WWW, On-Line Support	5

X

XORLW	369
XORWF	370

PIC18F87J11 FAMILY

NOTES:

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	X	XX	XXX	Examples:
Device	Temperature Range	Package	Pattern	
Device	PIC18F66J11/66J16/67J11 ⁽¹⁾ , PIC18F86J11/86J16/87J11 ⁽¹⁾ , PIC18F66J11/66J16/67J11T ⁽²⁾ , PIC18F86J11/86J16/87J11T ⁽²⁾			a) PIC18F87J11-I/PT_301 = Industrial temp., TQFP package, QTP pattern #301. b) PIC18F66J16T-I/PT = Tape and reel, Industrial temp., TQFP package.
Temperature Range	I = -40°C to +85°C (Industrial)			
Package	PT = TQFP (Thin Quad Flatpack)			
Pattern	QTP, SQTP, Code or Special Requirements (blank otherwise)			Note 1: F = Standard Voltage Range 2: T = in tape and reel



MICROCHIP

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
<http://support.microchip.com>
Web Address:
www.microchip.com

Atlanta

Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston

Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago

Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas

Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit

Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo

Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles

Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara

Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto

Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Habour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Fuzhou
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Shunde
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7250
Fax: 86-29-8833-7256

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471-6166
Fax: 81-45-471-6122

Korea - Gumi
Tel: 82-54-473-4301
Fax: 82-54-473-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Penang
Tel: 60-4-646-8870
Fax: 60-4-646-5086

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

12/08/06