

# PIC18F2455/2550/4455/4550 Data Sheet

28/40/44-Pin, High-Performance, Enhanced Flash, USB 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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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 WAR-RANTIES 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, 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, Migratable Memory, MXDEV, MXLAB, PICMASTER, 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, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Linear Active Thermistor, MPASM, MPLIB, MPLINK, MPSIM, 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.

 $\ensuremath{\mathsf{SQTP}}$  is a service mark of Microchip Technology Incorporated in the U.S.A.

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

© 2006, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

QUALITY MANAGEMENT SYSTEM

CERTIFIED BY DNV

ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOO® 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.



# 28/40/44-Pin, High-Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology

### **Universal Serial Bus Features:**

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 Endpoints (16 bidirectional)
- · 1-Kbyte Dual Access RAM for USB
- On-Chip USB Transceiver with On-Chip Voltage Regulator
- · Interface for Off-Chip USB Transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

### **Power-Managed Modes:**

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- · Sleep: CPU off, peripherals off
- Idle mode currents down to 5.8 μA typical
- Sleep mode currents down to 0.1 μA typical
- Timer1 Oscillator: 1.1 μA typical, 32 kHz, 2V
- Watchdog Timer: 2.1 μA typical
- Two-Speed Oscillator Start-up

### **Flexible Oscillator Structure:**

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
  - 8 user-selectable frequencies, from 31 kHz to 8 MHz
  - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
  - Allows for safe shutdown if any clock stops

### Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
- Capture is 16-bit, max. resolution 5.2 ns (Tcy/16)
- Compare is 16-bit, max. resolution 83.3 ns (Tcy)
- PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
  - Multiple output modes
  - Selectable polarity
  - Programmable dead time
  - Auto-shutdown and auto-restart
- Enhanced USART module:
  - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I<sup>2</sup>C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

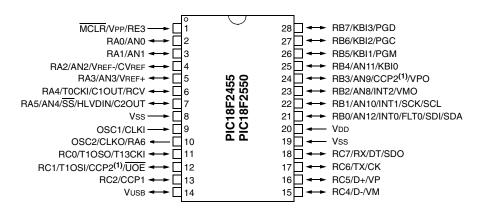
### **Special Microcontroller Features:**

- C Compiler Optimized Architecture with optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory typical
- Flash/Data EEPROM Retention: > 40 years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Optional dedicated ICD/ICSP port (44-pin devices only)
- Wide Operating Voltage Range (2.0V to 5.5V)

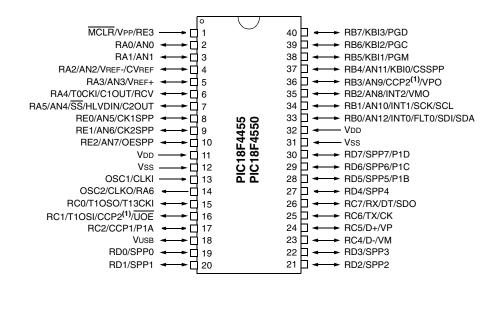
	Prog	ram Memory	Data	Data Memory					MSSP		RT	ators	
Device	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)	<b>I/O</b>	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	SPI	Master I <sup>2</sup> C™	EAUSAI	Comparat	Timers 8/16-Bit
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Υ	Υ	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Υ	Υ	1	2	1/3
PIC18F4455	24K	12288	2048	256	35	13	1/1	Yes	Υ	Υ	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Υ	Υ	1	2	1/3

### **Pin Diagrams**

### 28-Pin PDIP, SOIC



### 40-Pin PDIP



Note 1: RB3 is the alternate pin for CCP2 multiplexing.

#### Pin Diagrams (Continued) 44-Pin TQFP RC5/D+/VP RC4/D-/VM RD3/SPP3 RD2/SPP2 RD1/SPP1 RD0/SPP0 NC/ICRST(2)/ICVPP(2) RC7/RX/DT/SDO RC0/T10S0/T13CKI RD4/SPP4 32 Ш ◄ RD5/SPP5/P1B 31 ----OSC2/CLKO/RA6 RD6/SPP6/P1C 30 - ← OSC1/CLKI RD7/SPP7/P1D PIC18F4455 Vss – VDD Vss PIC18F4550 27 ₩₩₩ RE2/AN7/OESPP VDD RB0/AN12/INT0/FLT0/SDI/SDA 26 ---> RE1/AN6/CK2SPP 25 RE0/AN5/CK1SPP 24 RA5/AN4/SS/HLVDIN/C2OUT RB1/AN10/INT1/SCK/SCL RB2/AN8/INT2/VMO <del>----</del>□□□ 10 23 → RA4/T0CKI/C1OUT/RCV RB3/AN9/CCP2<sup>(1)</sup>/VPO NC/ICCK<sup>(2</sup>)/ICPGC<sup>(2)</sup> + NC/ICDT<sup>(2)</sup>/ICPGD<sup>(2)</sup> + RB4/AN11/KBI0/CSSPP + RB5/KBI1/PGM + RB5/KBI1/PGM + RB7/KBI3/PGD + MCLR/VPP/RE3 -RA2/AN2/VREF-/CVREF RC2/CCP1/P1A RC1/T1OSI/CCP2<sup>(1)</sup>/<u>UOE</u> RC0/T10S0/T13CKI 44-Pin QFN RC5/D+/VP RC4/D-/VM RD3/SPP3 RD2/SPP2 RD1/SPP1 RD1/SPP1 RC7/RX/DT/SDO OSC2/CLKO/RA6 RD4/SPP4 32 OSC1/CLKI RD5/SPP5/P1B 31 Vss 30 RD6/SPP6/P1C Vss 29 Vnn RD7/SPP7/P1D PIC18F4455 28 VDD Vss PIC18F4550 27 RE2/AN7/OESPP VDD 26 RE1/AN6/CK2SPP VDD RE0/AN5/<u>CK</u>1SPP RA5/AN4/SS/HLVDIN/C2OUT 25 RB0/AN12/INT0/FLT0/SDI/SDA 24 RB1/AN10/INT1/SCK/SCL 10 RA4/T0CKI/C1OUT/RCV 23 RB2/AN8/INT2/VMO NC RB4/AN11/KBI0/CSSPP + RB5/KBI1/PGM + RB6/KBI2/PGC + RB7/KBI3/PGD + MCLR/VPP/RE3 -RA0/AN0 + RB3/AN9/CCP2(1)/VPO RA2/AN2/VREF-/CVREF RA3/AN3/VREF+ RB3 is the alternate pin for CCP2 multiplexing. Note 1: Special ICPORTS features available in select circumstances. See Section 25.9 "Special ICPORT Features (Designated Packages Only)" for more information.

### **Table of Contents**

1.0	Device Overview	7
2.0	Oscillator Configurations	
3.0	Power-Managed Modes	35
4.0	Reset	43
5.0	Memory Organization	57
6.0	Flash Program Memory	79
7.0	Data EEPROM Memory	89
8.0	8 x 8 Hardware Multiplier	95
9.0	Interrupts	97
10.0	I/O Ports	111
11.0	Timer0 Module	125
12.0	Timer1 Module	129
13.0	Timer2 Module	135
14.0	Timer3 Module	137
15.0	Capture/Compare/PWM (CCP) Modules	141
16.0	Enhanced Capture/Compare/PWM (ECCP) Module	149
	Universal Serial Bus (USB)	
18.0	Streaming Parallel Port	187
19.0	Master Synchronous Serial Port (MSSP) Module	193
20.0	Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART)	237
21.0	10-Bit Analog-to-Digital Converter (A/D) Module	259
22.0	Comparator Module	269
	Comparator Voltage Reference Module	
24.0	High/Low-Voltage Detect (HLVD)	279
25.0	Special Features of the CPU	285
26.0	Instruction Set Summary	307
27.0	Development Support	357
28.0	Electrical Characteristics	361
29.0	DC and AC Characteristics Graphs and Tables	399
30.0	Packaging Information	401
Appe	endix A: Revision History	409
	endix B: Device Differences	
Appe	endix C: Conversion Considerations	410
Appe	endix D: Migration From Baseline to Enhanced Devices	410
	endix E: Migration From Mid-Range to Enhanced Devices	
Appe	endix F: Migration From High-End to Enhanced Devices	411
Index	x	413
The I	Microchip Web Site	425
Custo	omer Change Notification Service	425
Custo	omer Support	425
Read	der Response	426
DIC1	19E245E/2550/4455/4550 Product Identification System	497

### 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.

NOTES:

### 1.0 DEVICE OVERVIEW

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

PIC18F2455
 PIC18F2550
 PIC18F2550
 PIC18F4455
 PIC18F4550
 PIC18F4550
 PIC18LF4550

This family of devices offers the advantages of all PIC18 microcontrollers – namely, high computational performance at an economical price – with the addition of high endurance, Enhanced Flash program memory. In addition to these features, the PIC18F2455/2550/4455/4550 family introduces design enhancements that make these microcontrollers a logical choice for many high-performance, power sensitive applications.

### 1.1 New Core Features

#### 1.1.1 nanoWatt TECHNOLOGY

All of the devices in the PIC18F2455/2550/4455/4550 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 oscillator block, 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.
- Low Consumption in Key Modules: The power requirements for both Timer1 and the Watchdog Timer are minimized. See Section 28.0 "Electrical Characteristics" for values.

### 1.1.2 UNIVERSAL SERIAL BUS (USB)

Devices in the PIC18F2455/2550/4455/4550 family incorporate a fully featured Universal Serial Bus communications module that is compliant with the USB Specification Revision 2.0. The module supports both low-speed and full-speed communication for all supported data transfer types. It also incorporates its own on-chip transceiver and 3.3V regulator and supports the use of external transceivers and voltage regulators.

# 1.1.3 MULTIPLE OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC18F2455/2550/4455/4550 family offer twelve different oscillator options, allowing users a wide range of choices in developing application hardware. These include:

- Four Crystal modes using crystals or ceramic resonators.
- Four External Clock modes, offering the option of using two pins (oscillator input and a divide-by-4 clock output) or one pin (oscillator input, with the second pin reassigned as general I/O).
- An internal oscillator block which provides an 8 MHz clock (±2% accuracy) 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 both the High-Speed Crystal and External Oscillator modes, which allows a wide range of clock speeds from 4 MHz to 48 MHz.
- Asynchronous dual clock operation, allowing the USB module to run from a high-frequency oscillator while the rest of the microcontroller is clocked from an internal low-power oscillator.

Besides its availability as a clock source, 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 block, 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.

### 1.2 Other Special Features

- Memory Endurance: The Enhanced Flash cells for both program memory and data EEPROM are rated to last for many thousands of erase/write cycles – up to 100,000 for program memory and 1,000,000 for EEPROM. Data retention without refresh is conservatively estimated to be greater than 40 years.
- Self-Programmability: These devices can write to their own program memory spaces under internal software control. By using a bootloader routine, located in the protected Boot Block at the top of program memory, it becomes possible to create an application that can update itself in the field.
- Extended Instruction Set: The PIC18F2455/2550/4455/4550 family introduces an optional extension to the PIC18 instruction set, which adds 8 new instructions and an Indexed Literal Offset Addressing mode. This extension, enabled as a device configuration option, has been specifically designed to optimize re-entrant application code originally developed in high-level languages such as C.
- Enhanced CCP Module: In PWM mode, this
  module provides 1, 2 or 4 modulated outputs for
  controlling half-bridge and full-bridge drivers.
  Other features include auto-shutdown for
  disabling PWM outputs on interrupt or other select
  conditions and auto-restart to reactivate outputs
  once the condition has cleared.
- Enhanced Addressable USART: This serial
  communication module is capable of standard
  RS-232 operation and provides support for the LIN
  bus protocol. Other enhancements include
  Automatic Baud Rate Detection and a 16-bit Baud
  Rate Generator for improved resolution. When the
  microcontroller is using the internal oscillator
  block, the EUSART provides stable operation for
  applications that talk to the outside world without
  using an external crystal (or its accompanying
  power requirement).
- 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.
- Dedicated ICD/ICSP Port: These devices introduce the use of debugger and programming pins that are not multiplexed with other microcontroller features. Offered as an option in select packages, this feature allows users to develop I/O intensive applications while retaining the ability to program and debug in the circuit.

# 1.3 Details on Individual Family Members

Devices in the PIC18F2455/2550/4455/4550 family are available in 28-pin and 40/44-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 six ways:

- 1. Flash program memory (24 Kbytes for PIC18FX455 devices, 32 Kbytes for PIC18FX550).
- 2. A/D channels (10 for 28-pin devices, 13 for 40/44-pin devices).
- 3. I/O ports (3 bidirectional ports and 1 input only port on 28-pin devices, 5 bidirectional ports on 40/44-pin devices).
- CCP and Enhanced CCP implementation (28-pin devices have two standard CCP modules, 40/44-pin devices have one standard CCP module and one ECCP module).
- 5. Streaming Parallel Port (present only on 40/44-pin devices).

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

The pinouts for all devices are listed in Table 1-2 and Table 1-3.

Like all Microchip PIC18 devices, members of the PIC18F2455/2550/4455/4550 family are available as both standard and low-voltage devices. Standard devices with Enhanced Flash memory, designated with an "F" in the part number (such as PIC18F2550), accommodate an operating VDD range of 4.2V to 5.5V. Low-voltage parts, designated by "LF" (such as PIC18LF2550), function over an extended VDD range of 2.0V to 5.5V.

TABLE 1-1: DEVICE FEATURES

Features	PIC18F2455	PIC18F2550	PIC18F4455	PIC18F4550	
Operating Frequency	DC – 48 MHz				
Program Memory (Bytes)	24576	32768	24576	32768	
Program Memory (Instructions)	12288	16384	12288	16384	
Data Memory (Bytes)	2048	2048	2048	2048	
Data EEPROM Memory (Bytes)	256	256	256	256	
Interrupt Sources	19	19	20	20	
I/O Ports	Ports A, B, C, (E)	Ports A, B, C, (E)	Ports A, B, C, D, E	Ports A, B, C, D, E	
Timers	4	4	4	4	
Capture/Compare/PWM Modules	2	2	1	1	
Enhanced Capture/ Compare/PWM Modules	0	0	1	1	
Serial Communications	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART	
Universal Serial Bus (USB) Module	1	1	1	1	
Streaming Parallel Port (SPP)	No	No	Yes	Yes	
10-Bit Analog-to-Digital Module	10 Input Channels	10 Input Channels	13 Input Channels	13 Input Channels	
Comparators	2	2	2	2	
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	
Programmable Low-Voltage Detect	Yes	Yes	Yes	Yes	
Programmable Brown-out Reset	Yes	Yes	Yes	Yes	
Instruction Set	75 Instructions; 83 with Extended Instruction Set enabled				
Packages	28-pin PDIP 28-pin SOIC	28-pin PDIP 28-pin SOIC	40-pin PDIP 44-pin QFN 44-pin TQFP	40-pin PDIP 44-pin QFN 44-pin TQFP	

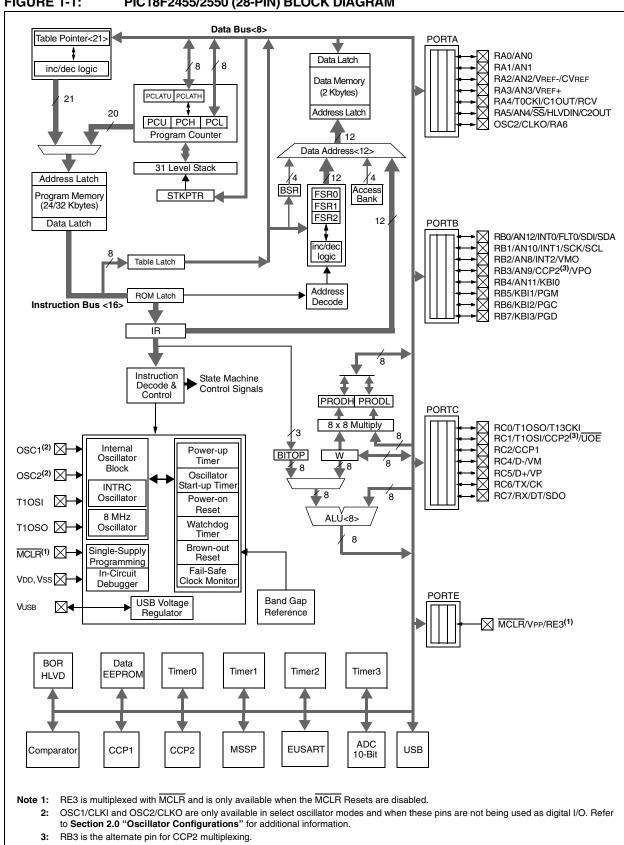


FIGURE 1-1: PIC18F2455/2550 (28-PIN) BLOCK DIAGRAM

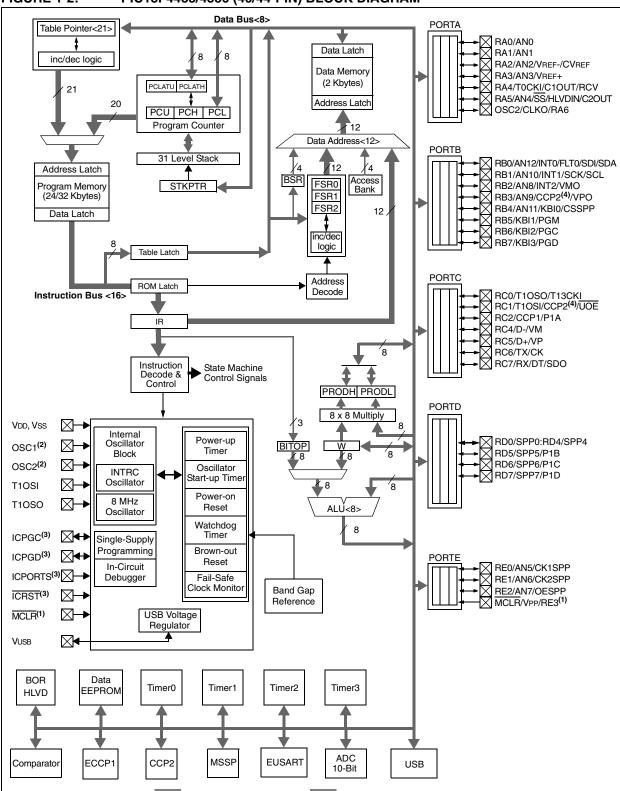


FIGURE 1-2: PIC18F4455/4550 (40/44-PIN) BLOCK DIAGRAM

- **Note 1:** RE3 is multiplexed with  $\overline{MCLR}$  and is only available when the  $\overline{MCLR}$  Resets are disabled.
  - 2: OSC1/CLKI and OSC2/CLKO are only available in select oscillator modes and when these pins are not being used as digital I/O. Refer to Section 2.0 "Oscillator Configurations" for additional information.
  - 3: These pins are only available on 44-pin TQFP packages under certain conditions. Refer to Section 25.9 "Special ICPORT Features (Designated Packages Only)" for additional information.
  - 4: RB3 is the alternate pin for CCP2 multiplexing.

TABLE 1-2: PIC18F2455/2550 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number	Pin	Buffer	Description				
Fill Name	PDIP, SOIC	Туре	Туре	Description				
MCLR/VPP/RE3 MCLR	1	I	ST	Master Clear (input) or programming voltage (input).  Master Clear (Reset) input. This pin is an active-low Reset to the device.				
VPP		Р		Programming voltage input.				
RE3		I	ST	Digital input.				
OSC1/CLKI	9			Oscillator crystal or external clock input.				
OSC1		- 1	Analog	Oscillator crystal input or external clock source input.				
CLKI		I	Analog	External clock source input. Always associated with pin function OSC1. (See OSC2/CLKO pin.)				
OSC2/CLKO/RA6	10			Oscillator crystal or clock output.				
OSC2		0	_	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.				
CLKO		0	_	In select 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.				

**Legend:** TTL = TTL compatible input

CMOS = CMOS compatible input or output

ST = Schmitt Trigger input with CMOS levels
O = Output

I = Input P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

TABLE 1-2: PIC18F2455/2550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Buffer		Description				
	PDIP, SOIC	Туре	туре	·				
				PORTA is a bidirectional I/O port.				
RA0/AN0	2							
RA0		I/O	TTL	Digital I/O.				
AN0		I	Analog	Analog input 0.				
RA1/AN1	3							
RA1		I/O	TTL	Digital I/O.				
AN1		I	Analog	Analog input 1.				
RA2/AN2/VREF-/CVREF RA2	4	I/O	TTL	Digital I/O.				
AN2		1/0	Analog	Analog input 2.				
VREF-		i	Analog	A/D reference voltage (low) input.				
CVREF		0	Analog	Analog comparator reference output.				
RA3/AN3/VREF+	5							
RA3		I/O	TTL	Digital I/O.				
AN3		1	Analog	Analog input 3.				
VREF+		ı	Analog	A/D reference voltage (high) input.				
RA4/T0CKI/C1OUT/RCV	6			5. 1. 1. (2				
RA4 T0CKI		I/O	ST ST	Digital I/O.				
C1OUT		0	51	Timer0 external clock input.  Comparator 1 output.				
RCV		Î	TTL	External USB transceiver RCV input.				
RA5/AN4/SS/	7			·				
HLVDIN/C2OUT	<b>'</b>							
RA5		I/O	TTL	Digital I/O.				
<u>AN</u> 4		1	Analog	Analog input 4.				
SS		!	TTL	SPI slave select input.				
HLVDIN C2OUT		0	Analog	High/Low-Voltage Detect input. Comparator 2 output.				
		U	_	· · · · · · · · · · · · · · · · · · ·				
RA6	_	_	_	See the OSC2/CLKO/RA6 pin.				

**Legend:** TTL = TTL compatible input

CMOS = CMOS compatible input or output

ST = Schmitt Trigger input with CMOS levels
O = Output

I = Input P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

TABLE 1-2: PIC18F2455/2550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Din Nama	Pin Number	Pin	Buffer	Decarinting
Pin Name	PDIP, SOIC	Туре	Туре	Description
				PORTB is a bidirectional I/O port. PORTB can be software
				programmed for internal weak pull-ups on all inputs.
RB0/AN12/INT0/FLT0/	21			
SDI/SDA RB0		I/O	TTL	Digital I/O.
AN12		"	Analog	Analog input 12.
INTO		i	ST	External interrupt 0.
FLT0		ı	ST	PWM Fault input (CCP1 module).
SDI		I	ST	SPI data in.
SDA		I/O	ST	I <sup>2</sup> C™ data I/O.
RB1/AN10/INT1/SCK/ SCL	22			
RB1		I/O	TTL	Digital I/O.
AN10		I	Analog	Analog input 10.
INT1		l I	ST	External interrupt 1.
SCK SCL		I/O I/O	ST ST	Synchronous serial clock input/output for SPI mode.  Synchronous serial clock input/output for I <sup>2</sup> C mode.
		1/0	51	Synchronous serial clock input/output for FC mode.
RB2/AN8/INT2/VMO RB2	23	I/O	TTL	Digital I/O.
AN8		1/0	Analog	Analog input 8.
INT2		;	ST	External interrupt 2.
VMO		Ö	_	External USB transceiver VMO output.
RB3/AN9/CCP2/VPO	24			·
RB3		I/O	TTL	Digital I/O.
AN9		I	Analog	Analog input 9.
CCP2 <sup>(1)</sup>		I/O	ST	Capture 2 input/Compare 2 output/PWM 2 output.
VPO		0	_	External USB transceiver VPO output.
RB4/AN11/KBI0	25			
RB4		I/O	TTL	Digital I/O.
AN11			Analog	Analog input 11.
KBI0		I	TTL	Interrupt-on-change pin.
RB5/KBI1/PGM	26	1/0	TT1	Digital I/O
RB5 KBI1		I/O I	TTL TTL	Digital I/O. Interrupt-on-change pin.
PGM		1/0	ST	Low-Voltage ICSP <sup>TM</sup> Programming enable pin.
RB6/KBI2/PGC	27	","		2011 Voltago 1001 - 1 Togramming enable pin.
RB6	21	I/O	TTL	Digital I/O.
KBI2		1	TTL	Interrupt-on-change pin.
PGC		I/O	ST	In-Circuit Debugger and ICSP programming clock pin.
RB7/KBI3/PGD	28			
RB7		I/O	TTL	Digital I/O.
KBI3		I	TTL	Interrupt-on-change pin.
PGD		I/O	ST	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

= Input

O = Output

P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

TABLE 1-2: PIC18F2455/2550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number PDIP,	Pin Type	Buffer Type	Description
	SOIC			
				PORTC is a bidirectional I/O port.
RC0/T1OSO/T13CKI	11		0.7	D: :: 11/0
RC0 T1OSO		I/O O	ST	Digital I/O. Timer1 oscillator output.
T13CKI		ı	ST	Timer1/Timer3 external clock input.
RC1/T1OSI/CCP2/UOE	12			'
RC1		I/O	ST	Digital I/O.
T1OSI		1	CMOS	Timer1 oscillator input.
CCP2 <sup>(2)</sup> UOE		I/O	ST	Capture 2 input/Compare 2 output/PWM 2 output.  External USB transceiver OE output.
RC2/CCP1	13	_	_	External OSB transceiver OE output.
RC2	13	I/O	ST	Digital I/O.
CCP1		I/O	ST	Capture 1 input/Compare 1 output/PWM 1 output.
RC4/D-/VM	15			
RC4		I	TTL	Digital input.
D- VM		I/O I	— TTL	USB differential minus line (input/output).
RC5/D+/VP	16	Į.	111	External USB transceiver VM input.
RC5	10	1	TTL	Digital input.
D+		I/O	_	USB differential plus line (input/output).
VP		0	TTL	External USB transceiver VP input.
RC6/TX/CK	17			
RC6		I/O O	ST	Digital I/O.
CK		1/0	ST	EUSART asynchronous transmit. EUSART synchronous clock (see RX/DT).
RC7/RX/DT/SDO	18	1, 0	01	200, 111 3y110111011040 01001 (000 110 0 11).
RC7	.0	I/O	ST	Digital I/O.
RX		- 1	ST	EUSART asynchronous receive.
DT		I/O	ST	EUSART synchronous data (see TX/CK).
SDO		0		SPI data out.
RE3	_	_		See MCLR/VPP/RE3 pin.
Vusb	14	0		Internal USB 3.3V voltage regulator.
Vss	8, 19	Р		Ground reference for logic and I/O pins.
VDD	20	Р	_	Positive supply for logic and I/O pins.

**Legend:** TTL = TTL compatible input

CMOS = CMOS compatible input or output

ST = Schmitt Trigger input with CMOS levels

I = Input

O = Output

P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS

Pin Name	Pi	n Numl	ber	Pin	Buffer	Description		
Pin Name	PDIP	QFN	TQFP	Туре	Туре	Description		
MCLR/VPP/RE3 MCLR	1	18	18	1	ST	Master Clear (input) or programming voltage (input).  Master Clear (Reset) input. This pin is an active-low Reset to the device.		
VPP RE3				P I	ST	Programming voltage input. Digital input.		
OSC1/CLKI OSC1 CLKI	13	32	30	 	Analog Analog	, ,		
OSC2/CLKO/RA6 OSC2 CLKO	14	33	31	0	_	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO which has 1/4		
RA6				1/0	TTL	the frequency of OSC1 and denotes the instruction cycle rate.  General purpose I/O pin.		

**Legend:** TTL = TTL compatible input

CMOS = CMOS compatible input or output

ST = Schmitt Trigger input with CMOS levels
O = Output

I = Input P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

2: Default assignment for CCP2 when CCP2MX Configuration bit is set.

3: These pins are No Connect unless the ICPRT Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the DEBUG Configuration bit is cleared.

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pi	n Numl	ber	Pin Buffer		Description		
Pin Name	PDIP	QFN	TQFP	Туре	Туре	Description		
RA0/AN0 RA0 AN0	2	19	19	I/O I	TTL	PORTA is a bidirectional I/O port.  Digital I/O.		
RA1/AN1 RA1 AN1	3	20	20	I/O   I	Analog TTL Analog	Analog input 0.  Digital I/O.  Analog input 1.		
RA2/AN2/VREF-/ CVREF RA2 AN2 VREF- CVREF	4	21	21	I/O I I O	TTL Analog Analog Analog	Digital I/O. Analog input 2. A/D reference voltage (low) input. Analog comparator reference output.		
RA3/AN3/VREF+ RA3 AN3 VREF+	5	22	22	I/O I I	TTL Analog Analog	Digital I/O. Analog input 3. A/D reference voltage (high) input.		
RA4/T0CKI/C1OUT/ RCV RA4 T0CKI C1OUT RCV	6	23	23	I/O I O I	ST ST — TTL	Digital I/O. Timer0 external clock input. Comparator 1 output. External USB transceiver RCV input.		
RA5/AN4/SS/ HLVDIN/C2OUT RA5 AN4 SS HLVDIN C2OUT	7	24	24	I/O I I I O	TTL Analog TTL Analog —	Digital I/O. Analog input 4. SPI slave select input. High/Low-Voltage Detect input. Comparator 2 output. See the OSC2/CLKO/RA6 pin.		

**Legend:** TTL = TTL compatible input

CMOS = CMOS compatible input or output

ST = Schmitt Trigger input with CMOS levels I = Input
O = Output P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

2: Default assignment for CCP2 when CCP2MX Configuration bit is set.

3: These pins are No Connect unless the ICPRT Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the DEBUG Configuration bit is cleared.

© 2006 Microchip Technology Inc. Preliminary DS39632C-page 17

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Din Name	Pin Number			Pin	Buffer	Description.
Pin Name	PDIP	QFN	TQFP	Туре	Туре	Description
RB0/AN12/INT0/ FLT0/SDI/SDA RB0	33	9	8	1/0	TTL	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.  Digital I/O.
AN12 INT0 FLT0 SDI SDA RB1/AN10/INT1/SCK/	34	10	9	        /O	Analog ST ST ST ST	Analog input 12. External interrupt 0. Enhanced PWM Fault input (ECCP1 module). SPI data in. I <sup>2</sup> C™ data I/O.
SCL RB1 AN10 INT1 SCK SCL	54	10	9	I/O I I I/O I/O	TTL Analog ST ST ST	Digital I/O. Analog input 10. External interrupt 1. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I <sup>2</sup> C mode.
RB2/AN8/INT2/VMO RB2 AN8 INT2 VMO	35	11	10	I/O         	TTL Analog ST —	Digital I/O. Analog input 8. External interrupt 2. External USB transceiver VMO output.
RB3/AN9/CCP2/VPO RB3 AN9 CCP2 <sup>(1)</sup> VPO	36	12	11	I/O I I/O O	TTL Analog ST —	Digital I/O. Analog input 9. Capture 2 input/Compare 2 output/PWM 2 output. External USB transceiver VPO output.
RB4/AN11/KBI0/CSSPP RB4 AN11 KBI0 CSSPP	37	14	14	I/O I I O	TTL Analog TTL —	Digital I/O. Analog input 11. Interrupt-on-change pin. SPP chip select control output.
RB5/KBI1/PGM RB5 KBI1 PGM	38	15	15	I/O I I/O	TTL TTL ST	Digital I/O. Interrupt-on-change pin. Low-Voltage ICSP™ Programming enable pin.
RB6/KBI2/PGC RB6 KBI2 PGC	39	16	16	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	40	17	17	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
O = Output

I = Input P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

- 2: Default assignment for CCP2 when CCP2MX Configuration bit is set.
- 3: These pins are No Connect unless the ICPRT Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the DEBUG Configuration bit is cleared.

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pi	n Numl	ber	Pin	Buffer	Description
Pin Name	PDIP	QFN	TQFP	Туре	Туре	Description
						PORTC is a bidirectional I/O port.
RC0/T10S0/T13CKI RC0 T10S0 T13CKI	15	34	32	I/O O I	ST — ST	Digital I/O. Timer1 oscillator output. Timer1/Timer3 external clock input.
RC1/T1OSI/CCP2/ UOE	16	35	35			
RC1 T1OSI CCP2 <sup>(2)</sup> UOE				I/O I I/O O	ST CMOS ST	Digital I/O. Timer1 oscillator input. Capture 2 input/Compare <u>2 o</u> utput/PWM 2 output. External USB transceiver OE output.
RC2/CCP1/P1A RC2 CCP1 P1A	17	36	36	I/O I/O O	ST ST TTL	Digital I/O. Capture 1 input/Compare 1 output/PWM 1 output. Enhanced CCP1 PWM output, channel A.
RC4/D-/VM RC4 D- VM	23	42	42	   I/O   I	TTL — TTL	Digital input. USB differential minus line (input/output). External USB transceiver VM input.
RC5/D+/VP RC5 D+ VP	24	43	43	I I/O I	TTL	Digital input. USB differential plus line (input/output). External USB transceiver VP input.
RC6/TX/CK RC6 TX CK	25	44	44	I/O O I/O	ST — ST	Digital I/O. EUSART asynchronous transmit. EUSART synchronous clock (see RX/DT).
RC7/RX/DT/SDO RC7 RX DT SDO	26	1	1	I/O I I/O O	ST ST ST —	Digital I/O. EUSART asynchronous receive. EUSART synchronous data (see TX/CK). SPI data out.

**Legend:** TTL = TTL compatible input

CMOS = CMOS compatible input or output

ST = Schmitt Trigger input with CMOS levels

l = Input

O = Output

P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

2: Default assignment for CCP2 when CCP2MX Configuration bit is set.

3: These pins are No Connect unless the ICPRT Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the DEBUG Configuration bit is cleared.

**TABLE 1-3:** PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pi	n Numl	ber	Pin	Buffer	Description		
Pili Naille	PDIP	QFN	TQFP	Туре	Туре	Description		
						PORTD is a bidirectional I/O port or a Streaming Parallel Port (SPP). These pins have TTL input buffers when the SPP module is enabled.		
RD0/SPP0 RD0 SPP0	19	38	38	I/O I/O	ST TTL	Digital I/O. Streaming Parallel Port data.		
RD1/SPP1 RD1 SPP1	20	39	39	I/O I/O	ST TTL	Digital I/O. Streaming Parallel Port data.		
RD2/SPP2 RD2 SPP2	21	40	40	I/O I/O	ST TTL	Digital I/O. Streaming Parallel Port data.		
RD3/SPP3 RD3 SPP3	22	41	41	I/O I/O	ST TTL	Digital I/O. Streaming Parallel Port data.		
RD4/SPP4 RD4 SPP4	27	2	2	I/O I/O	ST TTL	Digital I/O. Streaming Parallel Port data.		
RD5/SPP5/P1B RD5 SPP5 P1B	28	3	3	I/O I/O O	ST TTL	Digital I/O. Streaming Parallel Port data. Enhanced CCP1 PWM output, channel B.		
RD6/SPP6/P1C RD6 SPP6 P1C	29	4	4	I/O I/O O	ST TTL	Digital I/O. Streaming Parallel Port data. Enhanced CCP1 PWM output, channel C.		
RD7/SPP7/P1D RD7 SPP7 P1D	30	5	5	I/O I/O O	ST TTL —	Digital I/O. Streaming Parallel Port data. Enhanced CCP1 PWM output, channel D.		

**Legend:** TTL = TTL compatible input

CMOS = CMOS compatible input or output

ST = Schmitt Trigger input with CMOS levels

= Input

= Output

= Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

- 2: Default assignment for CCP2 when CCP2MX Configuration bit is set.
- 3: These pins are No Connect unless the ICPRT Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the DEBUG Configuration bit is cleared.

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number		Pin Buffer	Description		
Pili Name	PDIP	QFN	TQFP	Туре	Туре	Description
RE0/AN5/CK1SPP	8	25	25			PORTE is a bidirectional I/O port.
RE0 AN5 CK1SPP				I/O       	ST Analog —	Digital I/O. Analog input 5. SPP clock 1 output.
RE1/AN6/CK2SPP RE1 AN6 CK2SPP	9	26	26	I/O I O	ST Analog	Digital I/O. Analog input 6. SPP clock 2 output.
RE2/AN7/OESPP RE2 AN7 OESPP	10	27	27	I/O I O	ST Analog —	Digital I/O.
RE3	_	_	_	_	_	See MCLR/VPP/RE3 pin.
Vss	12, 31	6, 30, 31	6, 29	Р	_	Ground reference for logic and I/O pins.
VDD	11, 32	7, 8, 28, 29	7, 28	Р	_	Positive supply for logic and I/O pins.
Vusb	18	37	37	0	_	Internal USB 3.3V voltage regulator output.
NC/ICCK/ICPGC <sup>(3)</sup> ICCK ICPGC	_	_	12	I/O I/O	ST ST	No Connect or dedicated ICD/ICSP™ port clock. In-Circuit Debugger clock. ICSP programming clock.
NC/ICDT/ICPGD <sup>(3)</sup> ICDT ICPGD	_	_	13	I/O I/O	ST ST	No Connect or dedicated ICD/ICSP port clock. In-Circuit Debugger data. ICSP programming data.
NC/ICRST/ICVPP <sup>(3)</sup> ICRST ICVPP	_	_	33	I P	_	No Connect or dedicated ICD/ICSP port Reset.  Master Clear (Reset) input.  Programming voltage input.
NC/ICPORTS <sup>(3)</sup> ICPORTS	_	_	34	Р	_	No Connect or 28-pin device emulation.  Enable 28-pin device emulation when connected to Vss.
NC	_	13				No Connect.

**Legend:** TTL = TTL compatible input

CMOS = CMOS compatible input or output

ST = Schmitt Trigger input with CMOS levels
O = Output

I = Input P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

- 2: Default assignment for CCP2 when CCP2MX Configuration bit is set.
- 3: These pins are No Connect unless the ICPRT Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the DEBUG Configuration bit is cleared.

© 2006 Microchip Technology Inc. Preliminary DS39632C-page 21

NOTES:

# 2.0 OSCILLATOR CONFIGURATIONS

### 2.1 Overview

Devices in the PIC18F2455/2550/4455/4550 family incorporate a different oscillator and microcontroller clock system than previous PIC18F devices. The addition of the USB module, with its unique requirements for a stable clock source, make it necessary to provide a separate clock source that is compliant with both USB low-speed and full-speed specifications.

To accommodate these requirements, PIC18F2455/2550/4455/4550 devices include a new clock branch to provide a 48 MHz clock for full-speed USB operation. Since it is driven from the primary clock source, an additional system of prescalers and postscalers has been added to accommodate a wide range of oscillator frequencies. An overview of the oscillator structure is shown in Figure 2-1.

Other oscillator features used in PIC18 enhanced microcontrollers, such as the internal oscillator block and clock switching, remain the same. They are discussed later in this chapter.

### 2.1.1 OSCILLATOR CONTROL

The operation of the oscillator in PIC18F2455/2550/4455/4550 devices is controlled through two Configuration registers and two control registers. Configuration registers, CONFIG1L and CONFIG1H, select the oscillator mode and USB prescaler/postscaler options. As Configuration bits, these are set when the device is programmed and left in that configuration until the device is reprogrammed.

The OSCCON register (Register 2-2) selects the Active Clock mode; it is primarily used in controlling clock switching in power-managed modes. Its use is discussed in **Section 2.4.1** "Oscillator Control Register".

The OSCTUNE register (Register 2-1) is used to trim the INTRC frequency source, as well as select the low-frequency clock source that drives several special features. Its use is described in **Section 2.2.5.2** "OSCTUNE Register".

### 2.2 Oscillator Types

PIC18F2455/2550/4455/4550 devices can be operated in twelve distinct oscillator modes. In contrast with previous PIC18 enhanced microcontrollers, four of these modes involve the use of two oscillator types at once. Users can program the FOSC3:FOSC0 Configuration bits to select one of these modes:

- 1. XT Crvstal/Resonator
- 2. XTPLL Crystal/Resonator with PLL enabled
- 3. HS High-Speed Crystal/Resonator
- 4. HSPLL High-Speed Crystal/Resonator with PLL enabled
- 5. EC External Clock with Fosc/4 output
- 6. ECIO External Clock with I/O on RA6
- 7. ECPLL External Clock with PLL enabled and Fosc/4 output on RA6
- 8. ECPIO External Clock with PLL enabled, I/O on RA6
- INTHS Internal Oscillator used as
   microcontroller clock source, HS
   Oscillator used as USB clock source
- 10. INTXT Internal Oscillator used as microcontroller clock source, XT Oscillator used as USB clock source
- 11. INTIO Internal Oscillator used as microcontroller clock source, EC Oscillator used as USB clock source, digital I/O on RA6
- 12. INTCKO Internal Oscillator used as microcontroller clock source, EC Oscillator used as USB clock source, Fosc/4 output on RA6

# 2.2.1 OSCILLATOR MODES AND USB OPERATION

Because of the unique requirements of the USB module, a different approach to clock operation is necessary. In previous PICmicro® devices, all core and peripheral clocks were driven by a single oscillator source; the usual sources were primary, secondary or the internal oscillator. With PIC18F2455/2550/4455/4550 devices, the primary oscillator becomes part of the USB module and cannot be associated to any other clock source. Thus, the USB module must be clocked from the primary clock source; however, the microcontroller core and other peripherals can be separately clocked from the secondary or internal oscillators as before.

Because of the timing requirements imposed by USB, an internal clock of either 6 MHz or 48 MHz is required while the USB module is enabled. Fortunately, the microcontroller and other peripherals are not required to run at this clock speed when using the primary oscillator. There are numerous options to achieve the USB module clock requirement and still provide flexibility for clocking the rest of the device from the primary oscillator source. These are detailed in **Section 2.3** "Oscillator Settings for USB".

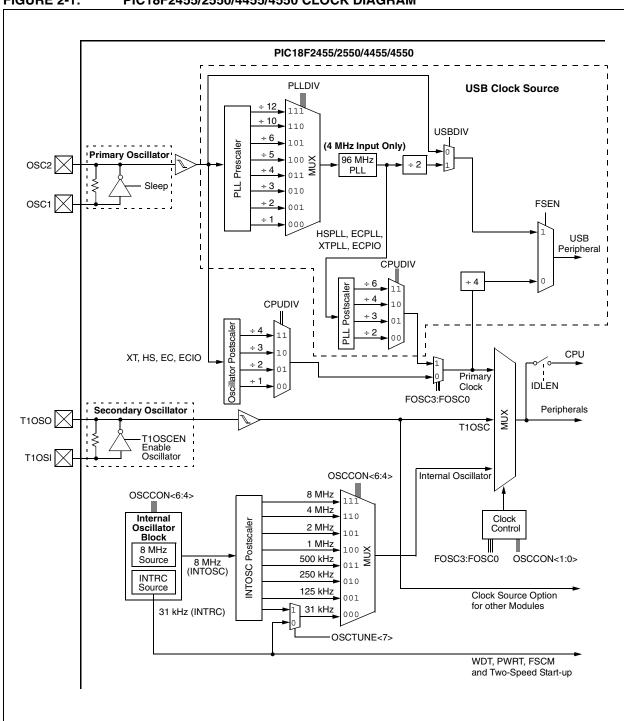


FIGURE 2-1: PIC18F2455/2550/4455/4550 CLOCK DIAGRAM

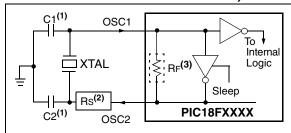
# 2.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In HS, HSPLL, XT and XTPLL 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 parallel cut crystal.

**Note:** Use of a series cut crystal may give a frequency out of the crystal manufacturer's specifications.

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



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

- 2: A series resistor (Rs) may be required for AT strip cut crystals.
- 3: RF varies with the oscillator mode chosen.

TABLE 2-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Typical Capacitor Values Used:					
Mode	Freq	OSC1	OSC2		
XT	4.0 MHz	33 pF	33 pF		
HS	8.0 MHz 16.0 MHz	27 pF 22 pF	27 pF 22 pF		

### Capacitor values are for design guidance only.

These capacitors were tested with the resonators listed below for basic start-up and operation. **These values are not optimized**.

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.

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

Resonators Used:				
4.0 MHz				
8.0 MHz				
16.0 MHz				

TABLE 2-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Crystal Freq	Typical Capacitor Value Tested:		
	rieq	C1	C2	
XT	4 MHz	27 pF	27 pF	
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.

These capacitors were tested with the crystals listed below for basic start-up and operation. **These values are not optimized.** 

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.

See the notes following this table for additional information.

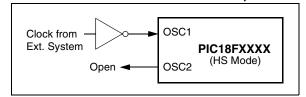
Crystals Used:					
4 MHz					
8 MHz					
20 MHz					

- **Note 1:** Higher capacitance increases the stability of oscillator but also increases the start-up time.
  - 2: When operating below 3V VDD, or when using certain ceramic resonators at any voltage, it may be necessary to use the HS mode or switch to a crystal oscillator.
  - 3: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.
  - **4:** Rs may be required to avoid overdriving crystals with low drive level specification.
  - **5:** Always verify oscillator performance over the VDD and temperature range that is expected for the application.

An internal postscaler allows users to select a clock frequency other than that of the crystal or resonator. Frequency division is determined by the CPUDIV Configuration bits. Users may select a clock frequency of the oscillator frequency, or 1/2, 1/3 or 1/4 of the frequency.

An external clock may also be used when the micro-controller is in HS Oscillator mode. In this case, the OSC2/CLKO pin is left open (Figure 2-3).

# FIGURE 2-3: EXTERNAL CLOCK INPUT OPERATION (HS OSC CONFIGURATION)

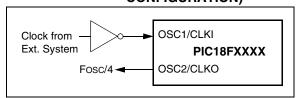


### 2.2.3 EXTERNAL CLOCK INPUT

The EC, ECIO, ECPLL and ECPIO 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.

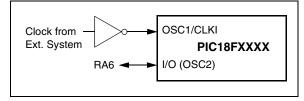
In the EC and ECPLL Oscillator modes, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 2-4 shows the pin connections for the EC Oscillator mode.

FIGURE 2-4: EXTERNAL CLOCK INPUT OPERATION (EC AND ECPLL CONFIGURATION)



The ECIO and ECPIO Oscillator modes function like the EC and ECPLL modes, except that the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6). Figure 2-5 shows the pin connections for the ECIO Oscillator mode.

FIGURE 2-5: EXTERNAL CLOCK INPUT OPERATION (ECIO AND ECPIO CONFIGURATION)



The internal postscaler for reducing clock frequency in XT and HS modes is also available in EC and ECIO modes.

### 2.2.4 PLL FREQUENCY MULTIPLIER

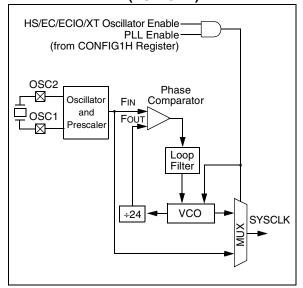
PIC18F2455/2550/4255/4550 devices include a Phase Locked Loop (PLL) circuit. This is provided specifically for USB applications with lower speed oscillators and can also be used as a microcontroller clock source.

The PLL is enabled in HSPLL, XTPLL, ECPLL and ECPIO Oscillator modes. It is designed to produce a fixed 96 MHz reference clock from a fixed 4 MHz input. The output can then be divided and used for both the USB and the microcontroller core clock. Because the PLL has a fixed frequency input and output, there are eight prescaling options to match the oscillator input frequency to the PLL.

There is also a separate postscaler option for deriving the microcontroller clock from the PLL. This allows the USB peripheral and microcontroller to use the same oscillator input and still operate at different clock speeds. In contrast to the postscaler for XT, HS and EC modes, the available options are 1/2, 1/3, 1/4 and 1/6 of the PLL output.

The HSPLL, ECPLL and ECPIO modes make use of the HS mode oscillator for frequencies up to 48 MHz. The prescaler divides the oscillator input by up to 12 to produce the 4 MHz drive for the PLL. The XTPLL mode can only use an input frequency of 4 MHz which drives the PLL directly.

FIGURE 2-6: PLL BLOCK DIAGRAM (HS MODE)



### 2.2.5 INTERNAL OSCILLATOR BLOCK

The PIC18F2455/2550/4455/4550 devices include an internal oscillator block which generates two different clock signals; either can be used as the microcontroller's clock source. If the USB peripheral is not used, the internal oscillator may eliminate the need for external oscillator circuits on the OSC1 and/or OSC2 pins.

The main output (INTOSC) is an 8 MHz clock source which can be used to directly drive the device clock. It also drives the INTOSC postscaler which can provide a range of clock frequencies from 31 kHz to 4 MHz. The INTOSC output is enabled when a clock frequency from 125 kHz to 8 MHz is selected.

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 25.0 "Special Features of the CPU".

The clock source frequency (INTOSC direct, INTRC direct or INTOSC postscaler) is selected by configuring the IRCF bits of the OSCCON register (page 32).

### 2.2.5.1 Internal Oscillator Modes

When the internal oscillator is used as the micro-controller clock source, one of the other oscillator modes (External Clock or External Crystal/Resonator) must be used as the USB clock source. The choice of the USB clock source is determined by the particular internal oscillator mode.

There are four distinct modes available:

- INTHS mode: The USB clock is provided by the oscillator in HS mode.
- INTXT mode: The USB clock is provided by the oscillator in XT mode.
- INTCKO mode: The USB clock is provided by an external clock input on OSC1/CLKI; the OSC2/ CLKO pin outputs Fosc/4.
- 4. INTIO mode: The USB clock is provided by an external clock input on OSC1/CLKI; the OSC2/CLKO pin functions as a digital I/O (RA6).

Of these four modes, only INTIO mode frees up an additional pin (OSC2/CLKO/RA6) for port I/O use.

### 2.2.5.2 OSCTUNE Register

The internal oscillator's output has been calibrated at the factory but can be adjusted in the user's application. This is done by writing to the OSCTUNE register (Register 2-1). The tuning sensitivity is constant throughout the tuning range.

When the OSCTUNE register is modified, the INTOSC and INTRC frequencies will begin shifting to the new frequency. The INTRC clock will reach the new frequency within 8 clock cycles (approximately,  $8*32~\mu s=256~\mu s$ ). The INTOSC clock will stabilize within 1 ms. Code execution continues during this shift. There is no indication that the shift has occurred.

The OSCTUNE register also contains the INTSRC bit. The INTSRC bit allows users to select which internal oscillator provides the clock source when the 31 kHz frequency option is selected. This is covered in greater detail in Section 2.4.1 "Oscillator Control Register".

# 2.2.5.3 Internal Oscillator Output Frequency and Drift

The internal oscillator block is calibrated at the factory to produce an INTOSC output frequency of 8.0 MHz. However, this frequency may drift as VDD or temperature changes, which can affect the controller operation in a variety of ways.

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

### REGISTER 2-1: OSCTUNE: OSCILLATOR TUNING REGISTER

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTSRC	_	_	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 directly from INTRC internal oscillator

bit 6-5 **Unimplemented:** Read as '0'

bit 4-0 TUN4:TUN0: Frequency Tuning bits

01111 = Maximum frequency

. .

00001

00000 = Center frequency. Oscillator module is running at the calibrated frequency.

11111

•

10000 = Minimum frequency

### 2.2.5.4 Compensating for INTOSC Drift

It is possible to adjust the INTOSC frequency by modifying the value in the OSCTUNE register. This has no effect on the INTRC clock source frequency.

Tuning the INTOSC source requires knowing when to make the adjustment, in which direction it should be made and in some cases, how large a change is needed. When using the EUSART, for example, an adjustment may be required when it 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.

It is also possible to verify device clock speed against a 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 greater than expected, then the internal oscillator block is running too fast. To adjust for this, decrement the OSCTUNE register.

Finally, 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.

### 2.3 Oscillator Settings for USB

When the PIC18F4550 is used for USB connectivity, it must have either a 6 MHz or 48 MHz clock for USB operation, depending on whether Low-Speed or Full-Speed mode is being used. This may require some forethought in selecting an oscillator frequency and programming the device.

The full range of possible oscillator configurations compatible with USB operation is shown in Table 2-3.

### 2.3.1 LOW-SPEED OPERATION

The USB clock for Low-Speed mode is derived from the primary oscillator chain and not directly from the PLL. It is divided by 4 to produce the actual 6 MHz clock. Because of this, the microcontroller can only use a clock frequency of 24 MHz when the USB module is active and the controller clock source is one of the primary oscillator modes (XT, HS or EC, with or without the PLL).

This restriction does not apply if the microcontroller clock source is the secondary oscillator or internal oscillator block.

# 2.3.2 RUNNING DIFFERENT USB AND MICROCONTROLLER CLOCKS

The USB module, in either mode, can run asynchronously with respect to the microcontroller core and other peripherals. This means that applications can use the primary oscillator for the USB clock while the microcontroller runs from a separate clock source at a lower speed. If it is necessary to run the entire application from only one clock source, full-speed operation provides a greater selection of microcontroller clock frequencies.

TABLE 2-3: OSCILLATOR CONFIGURATION OPTIONS FOR USB OPERATION

Input Oscillator Frequency	PLL Division (PLLDIV2:PLLDIV0)	Clock Mode (FOSC3:FOSC0)	MCU Clock Division (CPUDIV1:CPUDIV0)	Microcontroller Clock Frequency
48 MHz	N/A <sup>(1)</sup>	EC, ECIO	None (00)	48 MHz
			÷2 (01)	24 MHz
			÷3 (10)	16 MHz
			÷4 (11)	12 MHz
48 MHz	÷12 (111)	EC, ECIO	None (00)	48 MHz
			÷2 (01)	24 MHz
			÷3 (10)	16 MHz
			÷4 (11)	12 MHz
		ECPLL, ECPIO	÷2 (00)	48 MHz
			÷3 (01)	32 MHz
			÷4 (10)	24 MHz
			÷6 (11)	16 MHz
40 MHz	÷10 (110)	EC, ECIO	None (00)	40 MHz
			÷2 (01)	20 MHz
			÷3 (10)	13.33 MHz
			÷4 (11)	10 MHz
		ECPLL, ECPIO	÷2 (00)	48 MHz
			÷3 (01)	32 MHz
			÷4 (10)	24 MHz
			÷6 (11)	16 MHz
24 MHz	÷6 (101)	HS, EC, ECIO	None (00)	24 MHz
			÷2 (01)	12 MHz
			÷3 (10)	8 MHz
			÷4 (11)	6 MHz
		HSPLL, ECPLL, ECPIO	÷2 (00)	48 MHz
			÷3 (01)	32 MHz
			÷4 (10)	24 MHz
			÷6 (11)	16 MHz

Legend: All clock frequencies, except 24 MHz, are exclusively associated with full-speed USB operation (USB clock of 48 MHz).

Bold is used to highlight clock selections that are compatible with low-speed USB operation (system clock of 24 MHz, USB clock of 6 MHz).

**Note 1:** Only valid when the USBDIV Configuration bit is cleared.

TABLE 2-3: OSCILLATOR CONFIGURATION OPTIONS FOR USB OPERATION (CONTINUED)

Input Oscillator Frequency	PLL Division (PLLDIV2:PLLDIV0)	Clock Mode (FOSC3:FOSC0)	MCU Clock Division (CPUDIV1:CPUDIV0)	Microcontroller Clock Frequency
20 MHz	÷5 (100)	HS, EC, ECIO	None (00)	20 MHz
			÷2 (01)	10 MHz
			÷3 (10)	6.67 MHz
			÷4 (11)	5 MHz
		HSPLL, ECPLL, ECPIO	÷2 (00)	48 MHz
			÷3 (01)	32 MHz
			÷4 (10)	24 MHz
			÷6 (11)	16 MHz
16 MHz	÷4 (011)	HS, EC, ECIO	None (00)	16 MHz
			÷2 (01)	8 MHz
			÷3 (10)	5.33 MHz
			÷4 (11)	4 MHz
		HSPLL, ECPLL, ECPIO	÷2 (00)	48 MHz
			÷3 (01)	32 MHz
			÷4 (10)	24 MHz
			÷6 (11)	16 MHz
12 MHz	÷3 (010)	HS, EC, ECIO	None (00)	12 MHz
			÷2 (01)	6 MHz
			÷3 (10)	4 MHz
			÷4 (11)	3 MHz
		HSPLL, ECPLL, ECPIO	÷2 (00)	48 MHz
			÷3 (01)	32 MHz
			÷4 (10)	24 MHz
			÷6 (11)	16 MHz
8 MHz	÷2 (001)	HS, EC, ECIO	None (00)	8 MHz
			÷2 (01)	4 MHz
			÷3 (10)	2.67 MHz
			÷4 (11)	2 MHz
		HSPLL, ECPLL, ECPIO	÷2 (00)	48 MHz
			÷3 (01)	32 MHz
			÷4 (10)	24 MHz
			÷6 (11)	16 MHz
4 MHz	÷1 (000)	XT, HS, EC, ECIO	None (00)	4 MHz
			÷2 (01)	2 MHz
			÷3 (10)	1.33 MHz
			÷4 (11)	1 MHz
		HSPLL, ECPLL, XTPLL,	÷2 (00)	48 MHz
		ECPIO	÷3 (01)	32 MHz
			÷4 (10)	24 MHz
			÷6 (11)	16 MHz

Legend: All clock frequencies, except 24 MHz, are exclusively associated with full-speed USB operation (USB clock of 48 MHz).

Bold is used to highlight clock selections that are compatible with low-speed USB operation (system clock of 24 MHz, USB clock of 6 MHz).

Note 1: Only valid when the USBDIV Configuration bit is cleared.

### 2.4 Clock Sources and Oscillator Switching

Like previous PIC18 enhanced devices, the PIC18F2455/2550/4455/4550 family includes a feature that allows the device clock source to be switched from the main oscillator to an alternate low-frequency clock source. PIC18F2455/2550/4455/4550 devices offer two alternate clock sources. When an alternate clock source is enabled, the various power-managed operating modes are available.

Essentially, there are three clock sources for these devices:

- Primary oscillators
- · Secondary oscillators
- Internal oscillator block

The **primary oscillators** include the External Crystal and Resonator modes, the External Clock modes and the internal oscillator block. The particular mode is defined by the FOSC3:FOSC0 Configuration bits. The details of these modes are covered earlier in this chapter.

The **secondary oscillators** are those external sources 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.

PIC18F2455/2550/4455/4550 devices offer the Timer1 oscillator as a secondary oscillator. This oscillator, in all power-managed modes, is often the time base for functions such as a Real-Time Clock. Most often, a 32.768 kHz watch crystal is connected between the RC0/T10S0/T13CKI and RC1/T10SI/UOE pins. Like the XT and HS oscillator mode circuits, loading capacitors are also connected from each pin to ground. The Timer1 oscillator is discussed in greater detail in Section 12.3 "Timer1 Oscillator".

In addition to being a primary clock source, the **internal oscillator block** 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.

### 2.4.1 OSCILLATOR CONTROL REGISTER

The OSCCON register (Register 2-2) controls several aspects of the device clock's operation, both in full power operation and in power-managed modes.

The System Clock Select bits, SCS1:SCS0, select the clock source. The available clock sources are the primary clock (defined by the FOSC3:FOSC0 Configuration bits), the secondary clock (Timer1 oscillator) and the internal oscillator block. The clock source changes immediately after one or more of the bits is written to, following a brief clock transition interval. The SCS bits are cleared on all forms of Reset.

The Internal Oscillator Frequency Select bits, IRCF2:IRCF0, select the frequency output of the internal oscillator block to drive the device clock. The choices are the INTRC source, the INTOSC source (8 MHz) or one of the frequencies derived from the INTOSC postscaler (31 kHz to 4 MHz). If the internal oscillator block is supplying the device clock, changing the states of these bits will have an immediate change on the internal oscillator's output. On device Resets, the default output frequency of the internal oscillator block is set at 1 MHz.

When an output frequency of 31 kHz is selected (IRCF2:IRCF0 = 000), users may choose which internal oscillator acts as the source. This is done with the INTSRC bit in the OSCTUNE register (OSCTUNE<7>). Setting this bit selects INTOSC as a 31.25 kHz clock source by enabling the divide-by-256 output of the INTOSC postscaler. Clearing INTSRC selects INTRC (nominally 31 kHz) as the clock source.

This option allows users to select the tunable and more precise INTOSC as a clock source, while maintaining power savings with a very low clock speed. Regardless of the setting of INTSRC, INTRC always remains the clock source for features such as the Watchdog Timer and the Fail-Safe Clock Monitor.

The OSTS, IOFS and T1RUN bits indicate which clock source is currently providing the device clock. The OSTS bit indicates that the Oscillator Start-up Timer has timed out and the primary clock is providing the device clock in primary clock modes. The IOFS bit indicates when the internal oscillator block has stabilized and is providing the device clock in RC Clock modes. The T1RUN bit (T1CON<6>) indicates when the Timer1 oscillator is providing the device clock in secondary clock modes. In power-managed modes, only one of these three bits will be set at any time. If none of these bits are set, the INTRC is providing the clock or the internal oscillator block 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 will be ignored.
  - 2: It is recommended that the Timer1 oscillator be operating and stable prior to switching to it as the clock source; otherwise, a very long delay may occur while the Timer1 oscillator starts.

### 2.4.2 OSCILLATOR TRANSITIONS

PIC18F2455/2550/4455/4550 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".

### REGISTER 2-2: OSCCON: OSCILLATOR CONTROL REGISTER

R/W-0	R/W-1	R/W-0	R/W-0	R <sup>(1)</sup>	R-0	R/W-0	R/W-0
IDLEN	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0
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 Idle mode on SLEEP instruction

0 = Device enters Sleep mode on SLEEP instruction

bit 6-4 IRCF2:IRCF0: Internal Oscillator Frequency Select bits

111 = 8 MHz (INTOSC drives clock directly)

110 = 4 MHz

101 = 2 MHz

100 = 1 MHz<sup>(3)</sup>

011 = 500 kHz

010 = 250 kHz

001 = 125 kHz

000 = 31 kHz (from either INTOSC/256 or INTRC directly)(2)

bit 3 OSTS: Oscillator Start-up Time-out Status bit<sup>(1)</sup>

1 = Oscillator Start-up Timer time-out has expired; primary oscillator is running

0 = Oscillator Start-up Timer time-out is running; primary oscillator is not ready

bit 2 IOFS: INTOSC Frequency Stable bit

1 = INTOSC frequency is stable

0 = INTOSC frequency is not stable

bit 1-0 SCS1:SCS0: System Clock Select bits

1x = Internal oscillator

01 = Timer1 oscillator

00 = Primary oscillator

Note 1: Depends on the state of the IESO Configuration bit.

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

3: Default output frequency of INTOSC on Reset.

# 2.5 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. Unless the USB module is enabled, 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 internal oscillator modes (RC\_RUN and RC\_IDLE), the internal oscillator block 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 25.2 "Watchdog Timer (WDT)", Section 25.3 "Two-Speed Start-up" and Section 25.4 "Fail-Safe Clock Monitor" for more information on WDT, Fail-Safe Clock Monitor and Two-Speed Start-up). The INTOSC output at 8 MHz may be used directly to clock the device or may be divided down by the postscaler. The INTOSC output is disabled if the clock is provided directly from the INTRC output.

Regardless of the Run or Idle mode selected, the USB clock source will continue to operate. If the device is operating from a crystal or resonator-based oscillator, that oscillator will continue to clock the USB module. The core and all other modules will switch to the new clock source.

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).

Sleep mode should never be invoked while the USB module is operating and connected. The only exception is when the device has been issued a "Suspend"

command over the USB. Once the module has suspended operation and shifted to a low-power state, the microcontroller may be safely put into Sleep mode.

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. Other features may be operating that do not require a device clock source (i.e., MSSP slave, PSP, INTn pins and others). Peripherals that may add significant current consumption are listed in Section 28.2 "DC Characteristics: Power-Down and Supply Current".

### 2.6 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 4.5 "Device Reset Timers"**.

The first timer is the Power-up Timer (PWRT), which provides a fixed delay on power-up (parameter 33, Table 28-12). It is enabled by clearing (= 0) the PWRTEN Configuration bit.

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

When the HSPLL Oscillator mode is selected, the device is kept in Reset for an additional 2 ms following the HS mode OST delay, so the PLL can lock to the incoming clock frequency.

There is a delay of interval, TCSD (parameter 38, Table 28-12), following POR, while the controller becomes ready to execute instructions. This delay runs concurrently with any other delays. This may be the only delay that occurs when any of the EC or internal oscillator modes are used as the primary clock source.

TABLE 2-4: OSC1 AND OSC2 PIN STATES IN SLEEP MODE

Oscillator Mode	OSC1 Pin	OSC2 Pin	
INTCKO	Floating, pulled by external clock	At logic low (clock/4 output)	
INTIO	Floating, pulled by external clock	Configured as PORTA, bit 6	
ECIO, ECPIO	Floating, pulled by external clock	Configured as PORTA, bit 6	
EC	Floating, pulled by external clock	At logic low (clock/4 output)	
XT and HS	Feedback inverter disabled at quiescent voltage level	Feedback inverter disabled at quiescent voltage level	

Note: See Table 4-2 in Section 4.0 "Reset" for time-outs due to Sleep and MCLR Reset.

NOTES:

### 3.0 POWER-MANAGED MODES

PIC18F2455/2550/4455/4550 devices offer a total of seven operating modes for more efficient power management. These modes provide a variety of options for selective power conservation in applications where resources may be limited (i.e., battery-powered devices).

There are three categories of power-managed modes:

- · Run modes
- · Idle modes
- · Sleep mode

These categories define which portions of the device are clocked and sometimes, 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 PICmicro® 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 PICmicro 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 the selection of a clock source. 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.

#### 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 FOSC3:FOSC0 Configuration bits
- The secondary clock (the Timer1 oscillator)
- The internal oscillator block (for RC modes)

# 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.

TABLE 3-1: POWER-MANAGED MODES

Mada	oso	CON Bits	Modul	e Clocking	Aveilable Cleak and Occillator Course							
Mode	IDLEN <sup>(1)</sup>	SCS1:SCS0	CPU	Peripherals	Available Clock and Oscillator Source							
Sleep	0	N/A	Off	Off	None – all clocks are disabled							
PRI_RUN	N/A	00	Clocked	Clocked	Primary – all oscillator modes. This is the normal full power execution mode.							
SEC_RUN	N/A	01	Clocked	Clocked	Secondary - Timer1 oscillator							
RC_RUN	N/A	1x	Clocked	Clocked	Internal oscillator block <sup>(2)</sup>							
PRI_IDLE	1	00	Off	Clocked	Primary – all oscillator modes							
SEC_IDLE	1	01	Off	Clocked	Secondary - Timer1 oscillator							
RC_IDLE	1	1x	Off	Clocked	Internal oscillator block <sup>(2)</sup>							

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

2: Includes INTOSC and INTOSC postscaler, as well as the INTRC source.

# 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.

Three bits indicate the current clock source and its status. They are:

- OSTS (OSCCON<3>)
- IOFS (OSCCON<2>)
- 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 IOFS bit is set, the INTOSC output is providing a stable 8 MHz clock source to a divider that actually drives the device clock. When the T1RUN bit is set, the Timer1 oscillator is providing the clock. If none of these bits are set, then either the INTRC clock source is clocking the device, or the INTOSC source is not yet stable.

If the internal oscillator block is configured as the primary clock source by the FOSC3:FOSC0 Configuration bits, then both the OSTS and IOFS bits may be set when in PRI\_RUN or PRI\_IDLE modes. This indicates that the primary clock (INTOSC output) is generating a stable 8 MHz output. Entering another RC power-managed mode at the same frequency would clear the OSTS bit.

- Note 1: Caution should be used when modifying a single IRCF bit. If VDD is less than 3V, it is possible to select a higher clock speed than is supported by the low VDD. Improper device operation may result if the VDD/Fosc specifications are violated.
  - 2: 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 25.3** "Two-Speed Start-up" for details). In this mode, the OSTS bit is set. The IOFS bit may be set if the internal oscillator block is the primary clock source (see **Section 2.4.1** "Oscillator Control Register").

### 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.

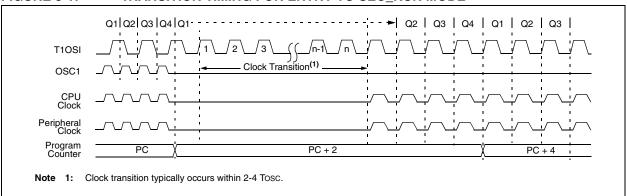
SEC\_RUN mode is entered by setting the SCS1:SCS0 bits to '01'. The device clock source is switched to the Timer1 oscillator (see Figure 3-1), the primary oscillator is shut down, the T1RUN bit (T1CON<6>) is set and the OSTS bit is cleared.

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

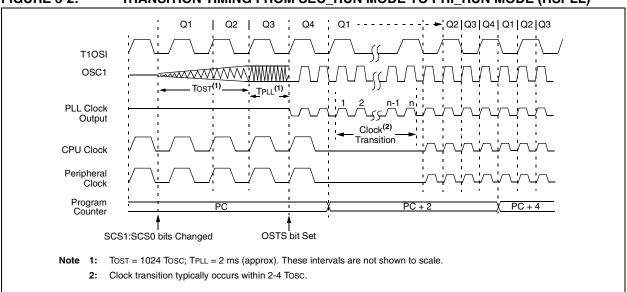
On transitions from SEC\_RUN mode to PRI\_RUN, the peripherals and CPU continue to be clocked from the Timer1 oscillator while the primary clock is started. When the primary clock becomes ready, a clock switch back to the primary clock occurs (see Figure 3-2). When the clock switch is complete, the T1RUN 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.

may result.

### 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)



### 3.2.3 RC\_RUN MODE

In RC\_RUN mode, the CPU and peripherals are clocked from the internal oscillator block using the INTOSC multiplexer; the primary clock is shut down. When using the INTRC source, 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.

If the primary clock source is the internal oscillator block (either INTRC or INTOSC), there are no distinguishable differences between the PRI\_RUN and RC\_RUN modes during execution. However, a clock switch delay will occur during entry to and exit from RC\_RUN mode. Therefore, if the primary clock source is the internal oscillator block, the use of RC\_RUN mode is not recommended.

This mode is entered by setting SCS1 to '1'. Although it is ignored, it is recommended that SCS0 also be cleared; this is to maintain software compatibility with future devices. When the clock source is switched to the INTOSC multiplexer (see Figure 3-3), the primary oscillator is shut down and the OSTS bit is cleared. The IRCF bits may be modified at any time to immediately change the clock speed.

Note: Caution should be used when modifying a single IRCF bit. If VDD is less than 3V, it is possible to select a higher clock speed than is supported by the low VDD. Improper device operation may result if the VDD/Fosc specifications are violated.

© 2006 Microchip Technology Inc. Preliminary DS39632C-page 37

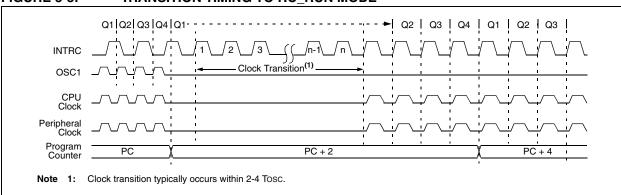
If the IRCF bits and the INTSRC bit are all clear, the INTOSC output is not enabled and the IOFS bit will remain clear; there will be no indication of the current clock source. The INTRC source is providing the device clocks.

If the IRCF bits are changed from all clear (thus, enabling the INTOSC output), or if INTSRC is set, the IOFS bit becomes set after the INTOSC output becomes stable. Clocks to the device continue while the INTOSC source stabilizes after an interval of TIOBST.

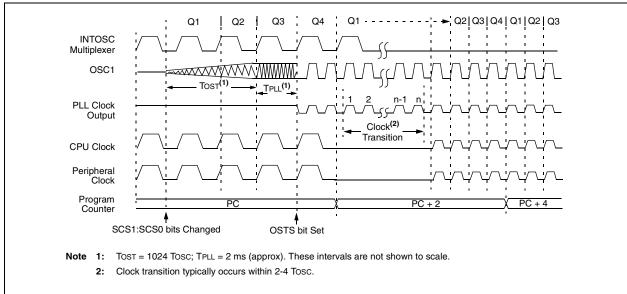
If the IRCF bits were previously at a non-zero value or if INTSRC was set before setting SCS1 and the INTOSC source was already stable, the IOFS bit will remain set.

On transitions from RC\_RUN mode to PRI\_RUN mode, the device continues to be clocked from the INTOSC multiplexer 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 IOFS bit is cleared, the OSTS 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 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 in the PIC18F2455/2550/4455/4550 devices is identical to the legacy Sleep mode offered in all other PICmicro 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 block if either the Two-Speed Start-up or the Fail-Safe Clock Monitor are enabled (see Section 25.0 "Special Features of the CPU"). In either case, the OSTS 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 TCSD (parameter 38, Table 28-12) 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 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

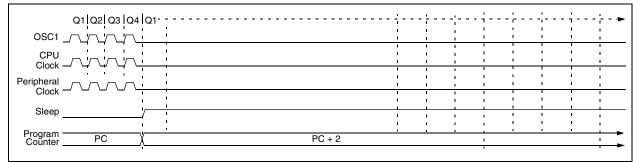
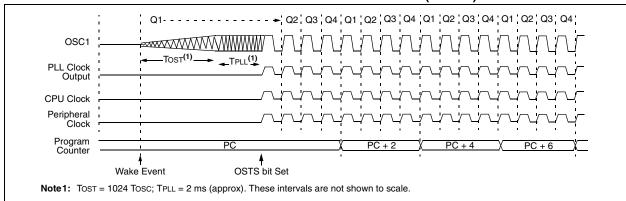


FIGURE 3-6: TRANSITION TIMING FOR WAKE FROM SLEEP (HSPLL)



### 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 clear the SCS bits and execute SLEEP. Although the CPU is disabled, the peripherals continue to be clocked from the primary clock source specified by the FOSC3: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.



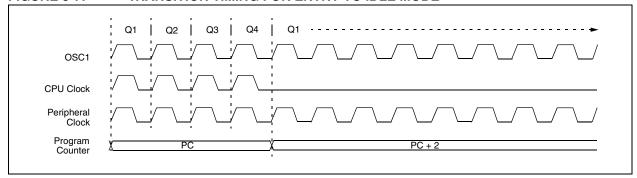
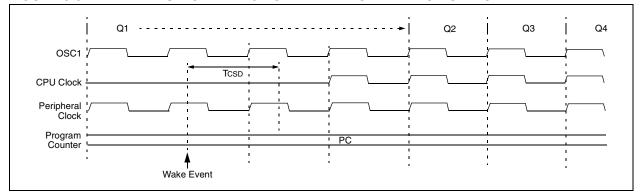


FIGURE 3-8: TRANSITION TIMING FOR WAKE FROM IDLE TO RUN MODE



### 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 using the INTOSC multiplexer. 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 set the SCS1 bit and execute SLEEP. Although its value is ignored, it is recommended that SCS0 also be cleared; this is to maintain software compatibility with future devices. The INTOSC multiplexer may be used to select a higher clock frequency by modifying the IRCF bits before executing the SLEEP instruction. When the clock source is switched to the INTOSC multiplexer, the primary oscillator is shut down and the OSTS bit is cleared.

If the IRCF bits are set to any non-zero value, or the INTSRC bit is set, the INTOSC output is enabled. The IOFS bit becomes set after the INTOSC output becomes stable, after an interval of TIOBST (parameter 39, Table 28-12). Clocks to the peripherals continue while the INTOSC source stabilizes. If the IRCF bits were previously at a non-zero value, or INTSRC was set before the SLEEP instruction was executed and the INTOSC source was already stable, the IOFS bit will remain set. If the IRCF bits and INTSRC are all clear, the INTOSC output will not be enabled, the IOFS bit will remain clear and there will be no indication of the current clock source.

When a wake event occurs, the peripherals continue to be clocked from the INTOSC multiplexer. After a delay of TCSD following the wake event, the CPU begins executing code being clocked by the INTOSC multiplexer. 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 (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 25.2 "Watchdog Timer (WDT)").

The WDT timer and postscaler are cleared by executing a SLEEP or CLRWDT instruction, the loss of a currently selected clock source (if the Fail-Safe Clock Monitor is enabled) and modifying the IRCF bits in the OSCCON register if the internal oscillator block is the device clock source.

#### 3.5.3 EXIT BY RESET

Normally, the device is held in Reset by the Oscillator Start-up Timer (OST) until the primary clock becomes ready. At that time, the OSTS bit is set and the device begins executing code. If the internal oscillator block is the new clock source, the IOFS bit is set instead.

The exit delay time from Reset to the start of code execution depends on both the clock sources before and after the wake-up and the type of oscillator if the new clock source is the primary clock. Exit delays are summarized in Table 3-2.

Code execution can begin before the primary clock becomes ready. If either the Two-Speed Start-up (see Section 25.3 "Two-Speed Start-up") or Fail-Safe Clock Monitor (see Section 25.4 "Fail-Safe Clock Monitor") is enabled, the device may begin execution as soon as the Reset source has cleared. Execution is clocked by the INTOSC multiplexer driven by the internal oscillator block. Execution is clocked by the internal oscillator block until either the primary clock becomes ready or a power-managed mode is entered before the primary clock becomes ready; the primary clock is then shut down.

# 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 not any of the XT or HS modes.

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 and any internal oscillator modes). 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.

TABLE 3-2: EXIT DELAY ON WAKE-UP BY RESET FROM SLEEP MODE OR ANY IDLE MODE (BY CLOCK SOURCES)

Microcontrolle	r Clock Source	Evit Delev	Clock Ready Status	
Before Wake-up	After Wake-up	Exit Delay	Bit (OSCCON)	
	XT, HS			
Primary Device Clock	XTPLL, HSPLL	None	OSTS	
(PRI_IDLE mode)	EC	None		
	INTOSC <sup>(3)</sup>		IOFS	
	XT, HS	Tost <sup>(4)</sup>		
T1OSC or INTRC <sup>(1)</sup>	XTPLL, HSPLL	Tost + t <sub>rc</sub> (4)	OSTS	
1103C OF INTACY	EC	Tcsp <sup>(2)</sup>		
	INTOSC <sup>(3)</sup>	Tiobst <sup>(5)</sup>	IOFS	
	XT, HS	Tost <sup>(4)</sup>		
INTOSC <sup>(3)</sup>	XTPLL, HSPLL	Tost + t <sub>rc</sub> (4)	OSTS	
INTOSC <sup>(*)</sup>	EC	Tcsp <sup>(2)</sup>		
	INTOSC <sup>(3)</sup>	None	IOFS	
	XT, HS	Tost <sup>(4)</sup>		
None	XTPLL, HSPLL	Tost + t <sub>rc</sub> <sup>(4)</sup>	OSTS	
(Sleep mode)	EC	Tcsp <sup>(2)</sup>		
	INTOSC <sup>(3)</sup>	Tiobst <sup>(5)</sup>	IOFS	

- Note 1: In this instance, refers specifically to the 31 kHz INTRC clock source.
  - 2: TCSD (parameter 38, Table 28-12) is a required delay when waking from Sleep and all Idle modes and runs concurrently with any other required delays (see Section 3.4 "Idle Modes").
  - 3: Includes both the INTOSC 8 MHz source and postscaler derived frequencies.
  - **4:** Tost is the Oscillator Start-up Timer period (parameter 32, Table 28-12). t<sub>rc</sub> is the PLL lock time-out (parameter F12, Table 28-9); it is also designated as TPLL.
  - 5: Execution continues during TIOBST (parameter 39, Table 28-12), the INTOSC stabilization period.

#### 4.0 RESET

The PIC18F2455/2550/4455/4550 devices differentiate between various kinds of Reset:

- Power-on Reset (POR)
- MCLR Reset during normal operation b)
- c) MCLR Reset during power-managed modes
- Watchdog Timer (WDT) Reset (during execution)
- e) Programmable Brown-out Reset (BOR)
- f) RESET Instruction
- Stack Full Reset
- 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 5.1.2.4 "Stack Full and Underflow Resets". WDT Resets are covered in Section 25.2 "Watchdog Timer (WDT)".

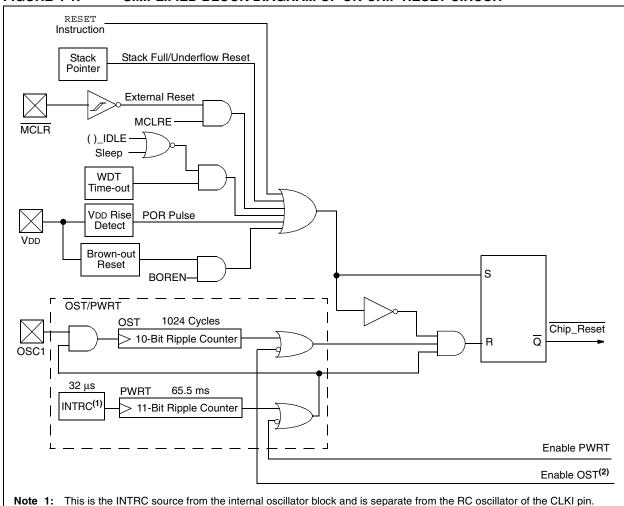
A simplified block diagram of the on-chip Reset circuit is shown in Figure 4-1.

#### 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 cleared by the event and must be set 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 4.6 "Reset State of Registers".

The RCON register also has control bits for setting interrupt priority (IPEN) and software control of the BOR (SBOREN). Interrupt priority is discussed in Section 9.0 "Interrupts". BOR is covered in Section 4.4 "Brown-out Reset (BOR)".

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



2: See Table 4-2 for time-out situations.

### REGISTER 4-1: RCON: RESET CONTROL REGISTER

R/W-0	R/W-1 <sup>(1)</sup>	U-0	R/W-1	R-1	R-1	R/W-0 <sup>(2)</sup>	R/W-0
IPEN	SBOREN	_	RI	TO	PD	POR	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 SBOREN: BOR Software Enable bit<sup>(1)</sup>

If BOREN1:BOREN0 = 01:

1 = BOR is enabled 0 = BOR is disabled

If BOREN1:BOREN0 = 00, 10 or 11:

Bit is disabled and read as '0'.

bit 5 Unimplemented: Read as '0'

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<sup>(2)</sup>

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: If SBOREN is enabled, its Reset state is '1'; otherwise, it is '0'.

2: The actual Reset value of POR is determined by the type of device Reset. See the notes following this register and Section 4.6 "Reset State of Registers" for additional information.

**Note 1:** It is recommended that the  $\overline{POR}$  bit be set after a Power-on Reset has been detected so that subsequent Power-on Resets may be detected.

2: Brown-out Reset is said to have occurred when  $\overline{\mathsf{BOR}}$  is '0' and  $\overline{\mathsf{POR}}$  is '1' (assuming that  $\overline{\mathsf{POR}}$  was set to '1' by software immediately after POR).

### 4.2 Master Clear Reset (MCLR)

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

The  $\overline{\text{MCLR}}$  pin is not driven low by any internal Resets, including the WDT.

In PIC18F2455/2550/4455/4550 devices, the MCLR input can <u>be disabled</u> with the MCLRE Configuration bit. When MCLR is disabled, the pin becomes a digital input. See **Section 10.5 "PORTE, TRISE and LATE Registers"** for more information.

### 4.3 Power-on Reset (POR)

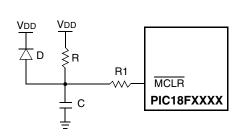
A Power-on Reset pulse is generated on-chip whenever VDD rises above a certain threshold. This allows the device to start in the initialized state when VDD is adequate for operation.

To take advantage of the POR circuitry, tie the  $\overline{MCLR}$  pin through a resistor (1 k $\Omega$  to 10 k $\Omega$ ) 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, **Section 28.1 "DC Characteristics"**). 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.

POR events are captured by the POR bit (RCON<1>). The state of the bit is set to '0' whenever a POR 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 POR.

# 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:  $R < 40 \text{ k}\Omega$  is recommended to make sure that the voltage drop across R does not violate the device's electrical specification.
  - 3:  $R1 \ge 1$  k $\Omega$  will limit any current flowing into  $\overline{MCLR}$  from external capacitor C, in the event of  $\overline{MCLR}$ /VPP pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

© 2006 Microchip Technology Inc. Preliminary DS39632C-page 45

### 4.4 Brown-out Reset (BOR)

PIC18F2455/2550/4455/4550 devices implement a BOR circuit that provides the user with a number of configuration and power-saving options. The BOR is controlled by the BORV1:BORV0 and BOREN1:BOREN0 Configuration bits. There are a total of four BOR configurations which are summarized in Table 4-1.

The BOR threshold is set by the BORV1:BORV0 bits. If BOR is enabled (any values of BOREN1:BOREN0 except '00'), any drop of VDD below VBOR (parameter D005, **Section 28.1 "DC Characteristics"**) for greater than TBOR (parameter 35, Table 28-12) 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.

If the Power-up Timer is enabled, it will be invoked after VDD rises above VBOR; it then will keep the chip in Reset for an additional time delay, TPWRT (parameter 33, Table 28-12). 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.

BOR and the Power-on Timer (PWRT) are independently configured. Enabling BOR Reset does not automatically enable the PWRT.

### 4.4.1 SOFTWARE ENABLED BOR

When BOREN1:BOREN0 = 01, the BOR can be enabled or disabled by the user in software. This is done with the control bit, SBOREN (RCON<6>). Setting SBOREN enables the BOR to function as previously described. Clearing SBOREN disables the BOR entirely. The SBOREN bit operates only in this mode; otherwise, it is read as '0'.

Placing the BOR under software control gives the user the additional flexibility of tailoring the application to its environment without having to reprogram the device to change BOR configuration. It also allows the user to tailor device power consumption in software by eliminating the incremental current that the BOR consumes. While the BOR current is typically very small, it may have some impact in low-power applications.

Note: Even when BOR is under software control, the BOR Reset voltage level is still set by the BORV1:BORV0 Configuration bits. It cannot be changed in software.

### 4.4.2 DETECTING BOR

When BOR is enabled, the  $\overline{\text{BOR}}$  bit always resets to '0' on any BOR or POR event. This makes it difficult to determine if a BOR event has occurred just by reading the state of  $\overline{\text{BOR}}$  alone. A more reliable method is to simultaneously check the state of both  $\overline{\text{POR}}$  and  $\overline{\text{BOR}}$ . This assumes that the  $\overline{\text{POR}}$  bit is reset to '1' in software immediately after any POR event. IF  $\overline{\text{BOR}}$  is '0' while  $\overline{\text{POR}}$  is '1', it can be reliably assumed that a BOR event has occurred.

### 4.4.3 DISABLING BOR IN SLEEP MODE

When BOREN1:BOREN0 = 10, the BOR remains under hardware control and operates as previously described. Whenever the device enters Sleep mode, however, the BOR is automatically disabled. When the device returns to any other operating mode, BOR is automatically re-enabled.

This mode allows for applications to recover from brown-out situations, while actively executing code, when the device requires BOR protection the most. At the same time, it saves additional power in Sleep mode by eliminating the small incremental BOR current.

TABLE 4-1: BOR CONFIGURATIONS

BOR Con	BOR Configuration					
BOREN1	BOREN0	SBOREN (RCON<6>)	BOR Operation			
0	0	Unavailable	BOR disabled; must be enabled by reprogramming the Configuration bits.			
0	1	Available	ble BOR enabled in software; operation controlled by SBOREN.			
1	0	Unavailable	BOR enabled in hardware in Run and Idle modes, disabled during Sleep mode.			
1	1	Unavailable	BOR enabled in hardware; must be disabled by reprogramming the Configuration bits.			

### 4.5 Device Reset Timers

PIC18F2455/2550/4455/4550 devices incorporate three separate on-chip timers that help regulate the Power-on Reset process. Their main function is to ensure that the device clock is stable before code is executed. These timers are:

- Power-up Timer (PWRT)
- Oscillator Start-up Timer (OST)
- · PLL Lock Time-out

### 4.5.1 POWER-UP TIMER (PWRT)

The Power-up Timer (PWRT) of the PIC18F2455/2550/4455/4550 devices is an 11-bit counter which uses the INTRC source as the clock input. This yields an approximate time interval of 2048 x 32  $\mu$ s = 65.6 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 (Table 28-12) for details.

The PWRT is enabled by clearing the PWRTEN Configuration bit.

# 4.5.2 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over (parameter 33, Table 28-12). This ensures that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, HS and HSPLL modes and only on Power-on Reset or on exit from most power-managed modes.

#### 4.5.3 PLL LOCK TIME-OUT

With the PLL enabled in its PLL mode, the time-out sequence following a Power-on Reset is slightly different from other oscillator modes. A separate timer is used to provide a fixed time-out that is sufficient for the PLL to lock to the main oscillator frequency. This PLL lock time-out (TPLL) is typically 2 ms and follows the oscillator start-up time-out.

### 4.5.4 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows:

- After the POR condition has cleared, PWRT time-out is invoked (if enabled).
- 2. Then, the OST is activated.

The total time-out will vary based on oscillator configuration and the status of the PWRT. Figure 4-3, Figure 4-4, Figure 4-5, Figure 4-6 and Figure 4-7 all depict time-out sequences on power-up, with the Power-up Timer enabled and the device operating in HS Oscillator mode. Figures 4-3 through 4-6 also apply to devices operating in XT mode. For devices in RC mode and with the PWRT disabled, on the other hand, there will be no time-out at all.

Since the time-outs occur from the POR pulse, if MCLR is kept low long enough, all time-outs 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.

TABLE 4-2: TIME-OUT IN VARIOUS SITUATIONS

Oscillator	Power-up <sup>(2)</sup> aı	Power-up <sup>(2)</sup> and Brown-out							
Configuration	PWRTEN = 0	PWRTEN = 1	Power-Managed Mode						
HS, XT	66 ms <sup>(1)</sup> + 1024 Tosc	1024 Tosc	1024 Tosc						
HSPLL, XTPLL	66 ms <sup>(1)</sup> + 1024 Tosc + 2 ms <sup>(2)</sup>	1024 Tosc + 2 ms <sup>(2)</sup>	1024 Tosc + 2 ms <sup>(2)</sup>						
EC, ECIO	66 ms <sup>(1)</sup>	_	_						
ECPLL, ECPIO	66 ms <sup>(1)</sup> + 2 ms <sup>(2)</sup>	2 ms <sup>(2)</sup>	2 ms <sup>(2)</sup>						
INTIO, INTCKO	66 ms <sup>(1)</sup>	_	_						
INTHS, INTXT	66 ms <sup>(1)</sup> + 1024 Tosc	1024 Tosc	1024 Tosc						

Note 1: 66 ms (65.5 ms) is the nominal Power-up Timer (PWRT) delay.

2: 2 ms is the nominal time required for the PLL to lock.

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

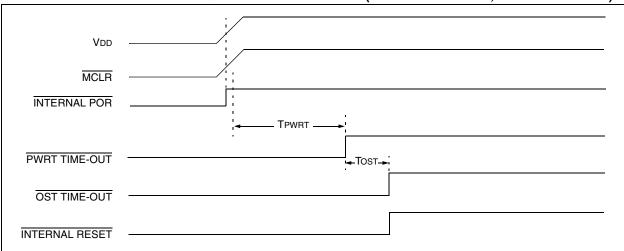


FIGURE 4-4: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 1

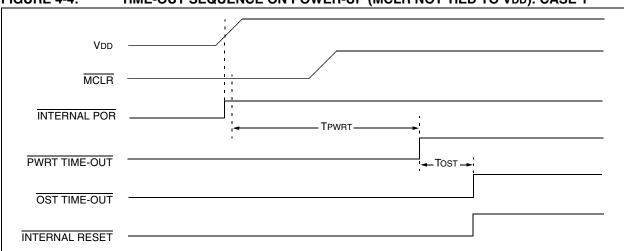
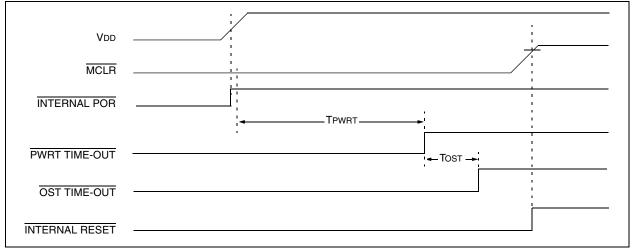
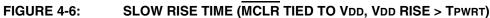
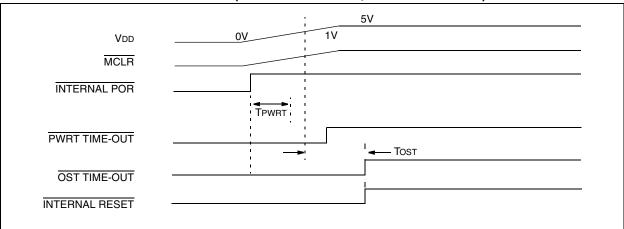


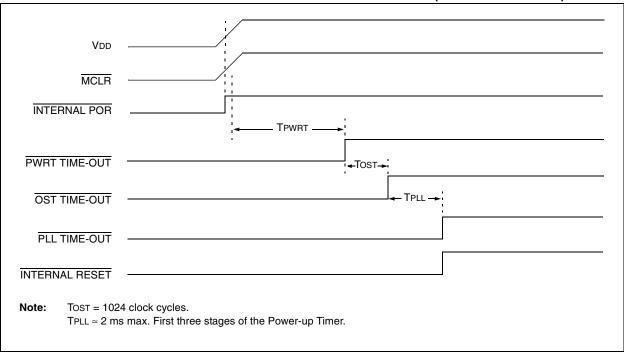
FIGURE 4-5: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 2











### 4.6 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, RI, TO, PD, POR and BOR, are set or cleared differently in different Reset situations as indicated in Table 4-3. These bits are used in software to determine the nature of the Reset.

Table 4-4 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 4-3: STATUS BITS, THEIR SIGNIFICANCE AND THE INITIALIZATION CONDITION FOR RCON REGISTER

Condition	Program		RCC		STKPTR Register				
Condition	Counter	SBOREN	RI	то	PD	POR	BOR	STKFUL	STKUNF
Power-on Reset	0000h	1	1	1	1	0	0	0	0
RESET Instruction	0000h	u <b>(2)</b>	0	u	u	u	u	u	u
Brown-out	0000h	u <b>(2)</b>	1	1	1	u	0	u	u
MCLR during Power-Managed Run modes	0000h	<b>u</b> (2)	u	1	u	u	u	u	u
MCLR during Power-Managed Idle modes and Sleep mode	0000h	<b>u</b> (2)	u	1	0	u	u	u	u
WDT Time-out during Full Power or Power-Managed Run modes	0000h	u <b>(2)</b>	u	0	u	u	u	u	u
MCLR during Full Power Execution	0000h	<b>u</b> (2)	u	u	u	u	u	u	u
Stack Full Reset (STVREN = 1)	0000h	u <b>(2)</b>	u	u	u	u	u	1	u
Stack Underflow Reset (STVREN = 1)	0000h	<b>u</b> (2)	u	u	u	u	u	u	1
Stack Underflow Error (not an actual Reset, STVREN = 0)	0000h	u <b>(2)</b>	u	u	u	u	u	u	1
WDT Time-out during Power-Managed Idle or Sleep modes	PC + 2	u <b>(2)</b>	u	0	0	u	u	u	u
Interrupt Exit from Power-Managed modes	PC + 2 <sup>(1)</sup>	u <b>(2)</b>	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 bits are set, the PC is loaded with the interrupt vector (008h or 0018h).

2: Reset state is '1' for POR and unchanged for all other Resets when software BOR is enabled (BOREN1:BOREN0 Configuration bits = 01 and SBOREN = 1); otherwise, the Reset state is '0'.

TABLE 4-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt		
TOSU	2455	2550	4455	4550	0 0000	0 0000	0 uuuu <sup>(1)</sup>
TOSH	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu <sup>(1)</sup>
TOSL	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu <sup>(1)</sup>
STKPTR	2455	2550	4455	4550	00-0 0000	uu-0 0000	uu-u uuuu <sup>(1)</sup>
PCLATU	2455	2550	4455	4550	0 0000	0 0000	u uuuu
PCLATH	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
PCL	2455	2550	4455	4550	0000 0000	0000 0000	PC + 2 <sup>(3)</sup>
TBLPTRU	2455	2550	4455	4550	00 0000	00 0000	uu uuuu
TBLPTRH	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
TBLPTRL	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
TABLAT	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
PRODH	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODL	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
INTCON	2455	2550	4455	4550	0000 000x	0000 000u	uuuu uuuu <sup>(2)</sup>
INTCON2	2455	2550	4455	4550	1111 -1-1	1111 -1-1	uuuu -u-u <b>(2)</b>
INTCON3	2455	2550	4455	4550	11-0 0-00	11-0 0-00	uu-u u-uu <b>(2)</b>
INDF0	2455	2550	4455	4550	N/A	N/A	N/A
POSTINC0	2455	2550	4455	4550	N/A	N/A	N/A
POSTDEC0	2455	2550	4455	4550	N/A	N/A	N/A
PREINC0	2455	2550	4455	4550	N/A	N/A	N/A
PLUSW0	2455	2550	4455	4550	N/A	N/A	N/A
FSR0H	2455	2550	4455	4550	0000	0000	uuuu
FSR0L	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF1	2455	2550	4455	4550	N/A	N/A	N/A
POSTINC1	2455	2550	4455	4550	N/A	N/A	N/A
POSTDEC1	2455	2550	4455	4550	N/A	N/A	N/A
PREINC1	2455	2550	4455	4550	N/A	N/A	N/A
PLUSW1	2455	2550	4455	4550	N/A	N/A	N/A
FSR1H	2455	2550	4455	4550	0000	0000	uuuu
FSR1L	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
BSR	2455	2550	4455	4550	0000	0000	uuuu

- **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: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).
  - 3: 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).
  - 4: See Table 4-3 for Reset value for specific condition.
  - **5:** PORTA<6>, LATA<6> and TRISA<6> are enabled depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

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

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt		
INDF2	2455	2550	4455	4550	N/A	N/A	N/A
POSTINC2	2455	2550	4455	4550	N/A	N/A	N/A
POSTDEC2	2455	2550	4455	4550	N/A	N/A	N/A
PREINC2	2455	2550	4455	4550	N/A	N/A	N/A
PLUSW2	2455	2550	4455	4550	N/A	N/A	N/A
FSR2H	2455	2550	4455	4550	0000	0000	uuuu
FSR2L	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	2455	2550	4455	4550	x xxxx	u uuuu	u uuuu
TMR0H	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
TMR0L	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	2455	2550	4455	4550	1111 1111	1111 1111	uuuu uuuu
OSCCON	2455	2550	4455	4550	0100 q000	0100 00q0	uuuu uuqu
HLVDCON	2455	2550	4455	4550	0 0-00 0101 0-00 0101		u-uu uuuu
WDTCON	2455	2550	4455	4550	50		u
RCON <sup>(4)</sup>	2455	2550	4455	4550	0q-1 11q0	0q-1 11q0	
TMR1H	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1L	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	2455	2550	4455	4550	0000 0000	u0uu uuuu	uuuu uuuu
TMR2	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
PR2	2455	2550	4455	4550	1111 1111	1111 1111	1111 1111
T2CON	2455	2550	4455	4550	-000 0000	-000 0000	-uuu uuuu
SSPBUF	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPADD	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
SSPCON1	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
SSPCON2	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
ADRESH	2455	2550	4455	4550	50 xxxx xxxx uuuu uuuu u		uuuu uuuu
ADRESL	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	2455	2550	4455	4550	00 0000	00 0000	uu uuuu
ADCON1	2455	2550	4455	4550	00 0qqq	00 0qqq	uu uuuu
ADCON2	2455	2550	4455	4550	0-00 0000	0-00 0000	u-uu uuuu

 $\label{eq:unchanged} \begin{tabular}{ll} $u$ = unchanged, $x$ = unknown, $-$ = unimplemented bit, read as `0', $q$ = value depends on condition. \\ Shaded cells indicate conditions do not apply for the designated device. \\ \end{tabular}$ 

- 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: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).
  - 3: 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).
  - 4: See Table 4-3 for Reset value for specific condition.
  - **5:** PORTA<6>, LATA<6> and TRISA<6> are enabled depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

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

Register	Applicable Devices				Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt
CCPR1H	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	2455	2550	4455	4550	00 0000	00 0000	uu uuuu
	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
CCPR2H	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2L	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
BAUDCON	2455	2550	4455	4550	0100 0-00	0100 0-00	uuuu u-uu
ECCP1DEL	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
ECCP1AS	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
CVRCON	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
CMCON	2455	2550	4455	4550	0000 0111	0000 0111	uuuu uuuu
TMR3H	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3L	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
T3CON	2455	2550	4455	4550	0000 0000	uuuu uuuu	uuuu uuuu
SPBRGH	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
SPBRG	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
RCREG	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
TXREG	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
TXSTA	2455	2550	4455	4550	0000 0010	0000 0010	uuuu uuuu
RCSTA	2455	2550	4455	4550	0000 000x	0000 000x	uuuu uuuu
EEADR	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
EEDATA	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
EECON2	2455	2550	4455	4550	0000 0000	0000 0000	0000 0000
EECON1	2455	2550	4455	4550	xx-0 x000	uu-0 u000	uu-0 u000

- 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: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).
  - 3: 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).
  - 4: See Table 4-3 for Reset value for specific condition.
  - **5:** PORTA<6>, LATA<6> and TRISA<6> are enabled depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

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

Register	Applicable Devices				Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt
IPR2	2455	2550	4455	4550	1111 1111	1111 1111	uuuu uuuu
PIR2	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu <sup>(2)</sup>
PIE2	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
IPR1	2455	2550	4455	4550	1111 1111	1111 1111	uuuu uuuu
	2455	2550	4455	4550	-111 1111	-111 1111	-uuu uuuu
PIR1	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu <b>(2)</b>
	2455	2550	4455	4550	-000 0000	-000 0000	-uuu uuuu
PIE1	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
	2455	2550	4455	4550	-000 0000	-000 0000	-uuu uuuu
OSCTUNE	2455	2550	4455	4550	00 0000	00 0000	uu uuuu
TRISE	2455	2550	4455	4550	111	111	uuu
TRISD	2455	2550	4455	4550	1111 1111	1111 1111	uuuu uuuu
TRISC	2455	2550	4455	4550	11111	11111	uuuuu
TRISB	2455	2550	4455	4550	1111 1111	1111 1111	uuuu uuuu
TRISA <sup>(5)</sup>	2455	2550	4455	4550	-111 1111 <sup>(5)</sup>	-111 1111 <sup>(5)</sup>	-uuu uuuu( <b>5)</b>
LATE	2455	2550	4455	4550	xxx	uuu	uuu
LATD	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATC	2455	2550	4455	4550	xxxxx	uuuuu	uuuuu
LATB	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATA <sup>(5)</sup>	2455	2550	4455	4550	-xxx xxxx(5)	-uuu uuuu <b>(5)</b>	-uuu uuuu <b>(5)</b>
PORTE	2455	2550	4455	4550	0 x000	0 x000	u uuuu
PORTD	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	2455	2550	4455	4550	xxxx -xxx	uuuu -uuu	uuuu -uuu
PORTB	2455	2550	4455	4550	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA <sup>(5)</sup>	2455	2550	4455	4550	-x0x 0000 <sup>(5)</sup>	-u0u 0000 <sup>(5)</sup>	-uuu uuuu <sup>(5)</sup>

- **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: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).
  - 3: 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).
  - **4:** See Table 4-3 for Reset value for specific condition.
  - **5:** PORTA<6>, LATA<6> and TRISA<6> are enabled depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

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

Register	Applicable Devices				Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt
UEP15	2455	2550	4455	4550	0 0000	0 0000	u uuuu
UEP14	2455	2550	4455	4550	0 0000	0 0000	u uuuu
UEP13	2455	2550	4455	4550	0 0000	0 0000	u uuuu
UEP12	2455	2550	4455	4550	0 0000	0 0000	u uuuu
UEP11	2455	2550	4455	4550	0 0000	0 0000	u uuuu
UEP10	2455	2550	4455	4550	0 0000	0 0000	u uuuu
UEP9	2455	2550	4455	4550	0 0000	0 0000	u uuuu
UEP8	2455	2550	4455	4550	0 0000	0 0000	u uuuu
UEP7	2455	2550	4455	4550	0 0000	0 0000	u uuuu
UEP6	2455	2550	4455	4550	0 0000	0 0000	u uuuu
UEP5	2455	2550	4455	4550	0 0000	0 0000	u uuuu
UEP4	2455	2550	4455	4550	0 0000	0 0000	u uuuu
UEP3	2455	2550	4455	4550	0 0000	0 0000	u uuuu
UEP2	2455	2550	4455	4550	0 0000	0 0000	u uuuu
UEP1	2455	2550	4455	4550	0 0000	0 0000	u uuuu
UEP0	2455	2550	4455	4550	0 0000	0 0000	u uuuu
UCFG	2455	2550	4455	4550	00-0 0000	00-0 0000	uu-u uuuu
UADDR	2455	2550	4455	4550	-000 0000	-000 0000	-uuu uuuu
UCON	2455	2550	4455	4550	-0x0 000-	-0x0 000-	-uuu uuu-
USTAT	2455	2550	4455	4550	-xxx xxx-	-xxx xxx-	-uuu uuu-
UEIE	2455	2550	4455	4550	00 0000	00 0000	uu uuuu
UEIR	2455	2550	4455	4550	00 0000	00 0000	uu uuuu
UIE	2455	2550	4455	4550	-000 0000	-000 0000	-uuu uuuu
UIR	2455	2550	4455	4550	-000 0000	-000 0000	-uuu uuuu
UFRMH	2455	2550	4455	4550	xxx	xxx	uuu
UFRML	2455	2550	4455	4550	xxxx xxxx	xxxx xxxx	uuuu uuuu
SPPCON	2455	2550	4455	4550	00	00	uu
SPPEPS	2455	2550	4455	4550	00-0 0000	00-0 0000	uu-u uuuu
SPPCFG	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu
SPPDATA	2455	2550	4455	4550	0000 0000	0000 0000	uuuu uuuu

- 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: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).
  - 3: 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).
  - 4: See Table 4-3 for Reset value for specific condition.
  - 5: PORTA<6>, LATA<6> and TRISA<6> are enabled depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

NOTES:

### 5.0 MEMORY ORGANIZATION

There are three types of memory in PIC18 enhanced microcontroller devices:

- · Program Memory
- Data RAM
- Data EEPROM

As Harvard architecture devices, the data and program memories use separate busses; this allows for concurrent access of the two memory spaces. The data EEPROM, for practical purposes, can be regarded as a peripheral device, since it is addressed and accessed through a set of control registers.

Additional detailed information on the operation of the Flash program memory is provided in **Section 6.0** "Flash Program Memory". Data EEPROM is discussed separately in **Section 7.0** "Data EEPROM Memory".

### 5.1 Program Memory Organization

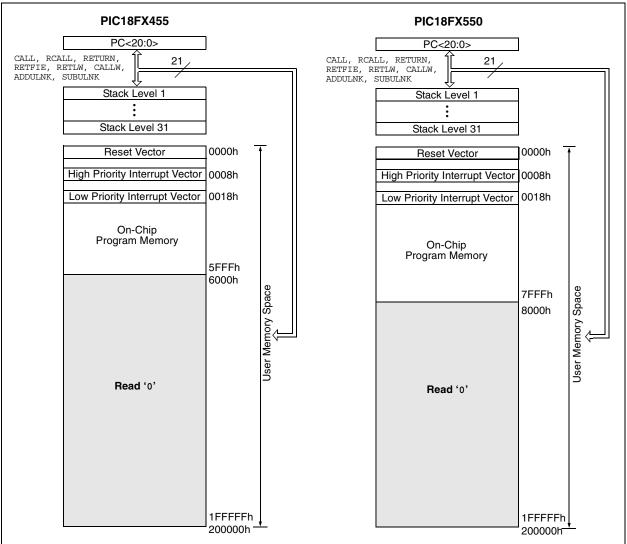
PIC18 microcontrollers implement a 21-bit program counter which is capable of addressing a 2-Mbyte program memory space. Accessing a location between the upper boundary of the physically implemented memory and the 2-Mbyte address will return all '0's (a NOP instruction).

The PIC18F2455 and PIC18F4455 each have 24 Kbytes of Flash memory and can store up to 12,288 single-word instructions. The PIC18F2550 and PIC18F4550 each have 32 Kbytes of Flash memory and can store up to 16,384 single-word instructions.

PIC18 devices have two interrupt vectors. The Reset vector address is at 0000h and the interrupt vector addresses are at 0008h and 0018h.

The program memory maps for PIC18FX455 and PIC18FX550 devices are shown in Figure 5-1.

FIGURE 5-1: PROGRAM MEMORY MAP AND STACK FOR PIC18F2455/2550/4455/4550 DEVICES



#### 5.1.1 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 5.1.4.1 "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 and GOTO 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.

#### 5.1.2 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. 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.

#### 5.1.2.1 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 5-2). This allows users to implement a software stack if necessary. After a CALL, RCALL or interrupt, 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.

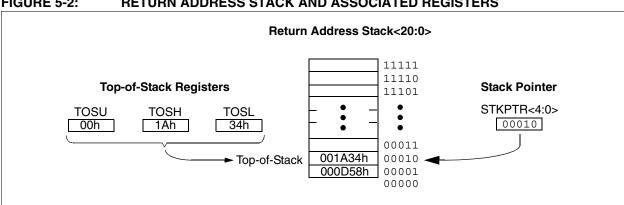


FIGURE 5-2: RETURN ADDRESS STACK AND ASSOCIATED REGISTERS

### 5.1.2.2 Return Stack Pointer (STKPTR)

The STKPTR register (Register 5-1) contains the Stack Pointer value, the STKFUL (Stack Full) status bit and the STKUNF (Stack Underflow) status bit. 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 25.1 "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 sets 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.

### 5.1.2.3 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 5-1: 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 <sup>(1)</sup>	STKUNF <sup>(1)</sup>	_	SP4	SP3	SP2	SP1	SP0
bit 7							bit 0

Legend:	C = Clearable 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 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<sup>(1)</sup>

1 = Stack underflow occurred0 = 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.

#### 5.1.2.4 Stack Full and Underflow Resets

Device Resets on stack overflow and stack underflow conditions are enabled by setting the STVREN bit in Configuration Register 4L. 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 user software or a Power-on Reset.

### 5.1.3 FAST REGISTER STACK

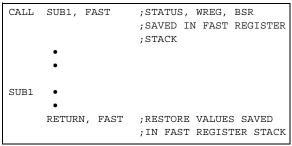
A Fast Register Stack is provided for the STATUS, WREG and BSR registers to provide a "fast return" option for interrupts. Each stack is only one level deep and is neither readable nor writable. It is loaded with the current value of the corresponding register when the processor vectors for an interrupt. All interrupt sources will push values into the stack registers. The values in the registers are then loaded back into their associated 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



# 5.1.4 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

### 5.1.4.1 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  $\,\mathrm{nn}$  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  $\,\mathrm{nn}$  instructions that returns the value ' $\,\mathrm{nn}$ ' 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

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

### 5.1.4.2 Table Reads and Table Writes

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 by using table reads and writes. The Table Pointer (TBLPTR) register specifies the byte address and the Table Latch (TABLAT) register contains the data that is read from or written to program memory. Data is transferred to or from program memory one byte at a time.

Table read and table write operations are discussed further in Section 6.1 "Table Reads and Table Writes".

### 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 5-3.

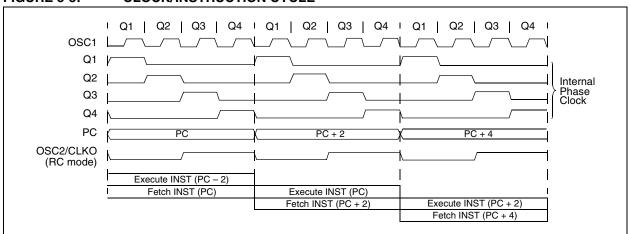
#### 5.2.2 INSTRUCTION FLOW/PIPELINING

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).

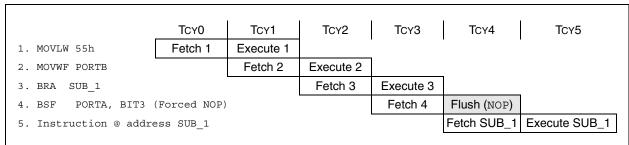
A fetch cycle begins with the Program Counter (PC) incrementing in Q1.

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).





### **EXAMPLE 5-3: INSTRUCTION PIPELINE FLOW**



**Note:** All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

© 2006 Microchip Technology Inc. Preliminary DS39632C-page 61

# 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 5.1.1** "**Program Counter**").

Figure 5-4 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 5-4 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 26.0 "Instruction Set Summary" provides further details of the instruction set.

FIGURE 5-4: INSTRUCTIONS IN PROGRAM MEMORY

FIGURE 3-4:	INSTRUCTIONS IN PROGRAM MEMORY										
				LSB = 1	<b>LSB</b> = 0	Word Address ↓					
		Program M	•			000000h					
		Byte Locat	ions $ ightarrow$			000002h					
						000004h					
						000006h					
	Instruction 1:	MOVLW	055h	0Fh	55h	000008h					
	Instruction 2:	GOTO	0006h	EFh	03h	00000Ah					
				F0h	00h	00000Ch					
	Instruction 3:	MOVFF	123h, 456h	C1h	23h	00000Eh					
				F4h	56h	000010h					
						000012h					
						000014h					

### 5.2.4 TWO-WORD INSTRUCTIONS

The standard PIC18 instruction set has four two-word instructions: CALL, MOVFF, GOTO and LSFR. 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 instruction 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. PIC18F2455/2550/4455/4550 devices implement eight complete banks, for a total of 2048 bytes. Figure 5-5 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 subsection.

To ensure that commonly used registers (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 SFRs and the lower portion of GPR Bank 0 without using the BSR. **Section 5.3.3** "Access Bank" provides a detailed description of the Access RAM.

### 5.3.1 USB RAM

Banks 4 through 7 of the data memory are actually mapped to special dual port RAM. When the USB module is disabled, the GPRs in these banks are used like any other GPR in the data memory space.

When the USB module is enabled, the memory in these banks is allocated as buffer RAM for USB operation. This area is shared between the microcontroller core and the USB Serial Interface Engine (SIE) and is used to transfer data directly between the two.

It is theoretically possible to use the areas of USB RAM that are not allocated as USB buffers for normal scratchpad memory or other variable storage. In practice, the dynamic nature of buffer allocation makes this risky at best. Additionally, Bank 4 is used for USB buffer management when the module is enabled and should not be used for any other purposes during that time.

Additional information on USB RAM and buffer operation is provided in **Section 17.0 "Universal Serial Bus (USB)"**.

### 5.3.2 BANK SELECT REGISTER (BSR)

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.

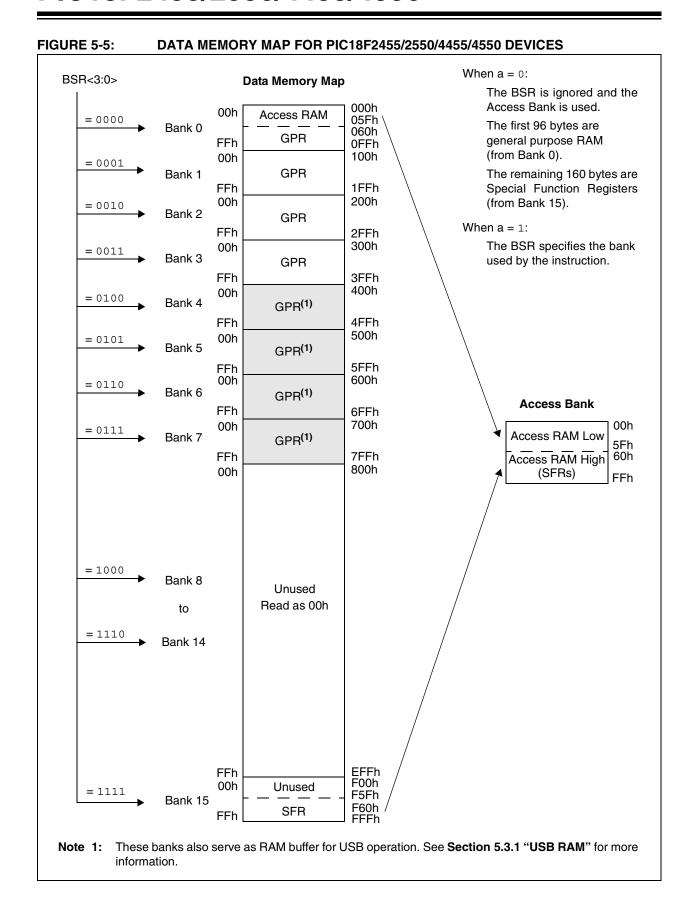
Most instructions in the PIC18 instruction set make use of the Bank Pointer, known as the Bank Select Register (BSR). This SFR holds the 4 Most Significant bits of a location's address; the instruction itself includes the eight Least Significant bits. Only the four lower bits of the BSR are implemented (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 eight 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 5-6.

Since up to sixteen 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.

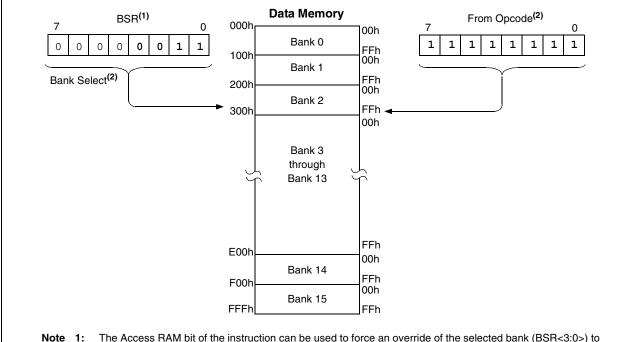
While any bank can be selected, only those banks that are actually implemented can be read or written to. Writes to unimplemented banks are ignored, while reads from unimplemented banks will return '0's. Even so, the STATUS register will still be affected as if the operation was successful. The data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



**Preliminary** 





Note 1: The Access RAM bit of the instruction can be used to force an override of the selected bank (BSR<3:0>) to the registers of the Access Bank.

2: The MOVFF instruction embeds the entire 12-bit address in the instruction.

### 5.3.3 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 Block 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 5-5).

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".

# 5.3.4 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.

#### 5.3.5 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 in the data memory space. SFRs start at the top of data memory and extend downward to occupy the top segment of Bank 15, from F60h to FFFh. A list of these registers is given in Table 5-1 and Table 5-2.

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 a peripheral feature 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 'o's.

TABLE 5-1: SPECIAL FUNCTION REGISTER MAP FOR PIC18F2455/2550/4455/4550 DEVICES

Address	Name	Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2 <sup>(1)</sup>	FBFh	CCPR1H	F9Fh	IPR1	F7Fh	UEP15
FFEh	TOSH	FDEh	POSTINC2 <sup>(1)</sup>	FBEh	CCPR1L	F9Eh	PIR1	F7Eh	UEP14
FFDh	TOSL	FDDh	POSTDEC2 <sup>(1)</sup>	FBDh	CCP1CON	F9Dh	PIE1	F7Dh	UEP13
FFCh	STKPTR	FDCh	PREINC2 <sup>(1)</sup>	FBCh	CCPR2H	F9Ch	(2)	F7Ch	UEP12
FFBh	PCLATU	FDBh	PLUSW2 <sup>(1)</sup>	FBBh	CCPR2L	F9Bh	OSCTUNE	F7Bh	UEP11
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	(2)	F7Ah	UEP10
FF9h	PCL	FD9h	FSR2L	FB9h	(2)	F99h	(2)	F79h	UEP9
FF8h	TBLPTRU	FD8h	STATUS	FB8h	BAUDCON	F98h	(2)	F78h	UEP8
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	ECCP1DEL	F97h	(2)	F77h	UEP7
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	ECCP1AS	F96h	TRISE <sup>(3)</sup>	F76h	UEP6
FF5h	TABLAT	FD5h	T0CON	FB5h	CVRCON	F95h	TRISD <sup>(3)</sup>	F75h	UEP5
FF4h	PRODH	FD4h	(2)	FB4h	CMCON	F94h	TRISC	F74h	UEP4
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB	F73h	UEP3
FF2h	INTCON	FD2h	HLVDCON	FB2h	TMR3L	F92h	TRISA	F72h	UEP2
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	(2)	F71h	UEP1
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRGH	F90h	(2)	F70h	UEP0
FEFh	INDF0 <sup>(1)</sup>	FCFh	TMR1H	FAFh	SPBRG	F8Fh	(2)	F6Fh	UCFG
FEEh	POSTINC0 <sup>(1)</sup>	FCEh	TMR1L	FAEh	RCREG	F8Eh	(2)	F6Eh	UADDR
FEDh	POSTDEC0 <sup>(1)</sup>	FCDh	T1CON	FADh	TXREG	F8Dh	LATE <sup>(3)</sup>	F6Dh	UCON
FECh	PREINC0 <sup>(1)</sup>	FCCh	TMR2	FACh	TXSTA	F8Ch	LATD <sup>(3)</sup>	F6Ch	USTAT
FEBh	PLUSW0 <sup>(1)</sup>	FCBh	PR2	FABh	RCSTA	F8Bh	LATC	F6Bh	UEIE
FEAh	FSR0H	FCAh	T2CON	FAAh	(2)	F8Ah	LATB	F6Ah	UEIR
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA	F69h	UIE
FE8h	WREG	FC8h	SSPADD	FA8h	EEDATA	F88h	(2)	F68h	UIR
FE7h	INDF1 <sup>(1)</sup>	FC7h	SSPSTAT	FA7h	EECON2 <sup>(1)</sup>	F87h	(2)	F67h	UFRMH
FE6h	POSTINC1 <sup>(1)</sup>	FC6h	SSPCON1	FA6h	EECON1	F86h	(2)	F66h	UFRML
FE5h	POSTDEC1 <sup>(1)</sup>	FC5h	SSPCON2	FA5h	(2)	F85h	(2)	F65h	SPPCON <sup>(3)</sup>
FE4h	PREINC1 <sup>(1)</sup>	FC4h	ADRESH	FA4h	(2)	F84h	PORTE	F64h	SPPEPS <sup>(3)</sup>
FE3h	PLUSW1 <sup>(1)</sup>	FC3h	ADRESL	FA3h	(2)	F83h	PORTD <sup>(3)</sup>	F63h	SPPCFG <sup>(3)</sup>
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC	F62h	SPPDATA <sup>(3)</sup>
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB	F61h	(2)
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F80h	PORTA	F60h	(2)

Note 1: Not a physical register.

- 2: Unimplemented registers are read as '0'.
- 3: These registers are implemented only on 40/44-pin devices.

TABLE 5-2: REGISTER FILE SUMMARY (PIC18F2455/2550/4455/4550)

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	_	1	1	Top-of-Stack	Upper Byte (T	OS<20:16>)			0 0000	51, 58
TOSH	Top-of-Stack	High Byte (TO	S<15:8>)						0000 0000	51, 58
TOSL	Top-of-Stack	Low Byte (TO	S<7:0>)						0000 0000	51, 58
STKPTR	STKFUL	STKUNF	_	SP4	SP3	SP2	SP1	SP0	00-0 0000	51, 59
PCLATU	_	1	-	Holding Regis	ster for PC<20	:16>		•	0 0000	51, 58
PCLATH	Holding Regis	ster for PC<15	:8>	•					0000 0000	51, 58
PCL	PC Low Byte (PC<7:0>)									51, 58
TBLPTRU	_	_	bit 21 <sup>(1)</sup>	Program Men	nory Table Poi	nter Upper By	te (TBLPTR<2	20:16>)	00 0000	51, 82
TBLPTRH	Program Men	nory Table Poi	nter High Byte	(TBLPTR<15	:8>)				0000 0000	51, 82
TBLPTRL	Program Men	nory Table Poi	nter Low Byte	(TBLPTR<7:0	>)				0000 0000	51, 82
TABLAT	Program Men	nory Table Lat	ch						0000 0000	51, 82
PRODH	Product Regis	ster High Byte							xxxx xxxx	51, 95
PRODL	Product Regis	ster Low Byte							xxxx xxxx	51, 95
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INTOIF	RBIF	0000 000x	51, 99
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	_	TMR0IP	_	RBIP	1111 -1-1	51, 100
INTCON3	INT2IP	INT1IP	_	INT2IE	INT1IE	_	INT2IF	INT1IF	11-0 0-00	51, 101
INDF0	Uses contents	s of FSR0 to a	ddress data m	nemory – value	of FSR0 not	changed (not a	a physical regi	ster)	N/A	51, 73
POSTINC0				nemory – value					N/A	51, 74
POSTDEC0				nemory – value			• • •		N/A	51, 74
PREINC0									N/A	51, 74
PLUSW0	Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register)  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	51, 74
FSR0H	_	_	_	_	Indirect Data	Memory Addr	ess Pointer 0	High Byte	0000	51, 73
FSR0L	Indirect Data	Memory Addre	ess Pointer 0 L	ow Byte	•				xxxx xxxx	51, 73
WREG	Working Regi	ster							xxxx xxxx	51
INDF1	Uses contents	s of FSR1 to a	ddress data m	emory – value	of FSR1 not	changed (not	a physical regi	ster)	N/A	51, 73
POSTINC1	Uses contents	s of FSR1 to a	ddress data m	nemory – value	of FSR1 pos	t-incremented	(not a physica	l register)	N/A	51, 74
POSTDEC1	Uses contents	s of FSR1 to a	ddress data m	nemory – value	of FSR1 pos	t-decremented	(not a physic	al register)	N/A	51, 74
PREINC1	Uses contents	s of FSR1 to a	ddress data m	nemory – value	of FSR1 pre-	incremented (	not a physical	register)	N/A	51, 74
PLUSW1	Uses contents value of FSR		ddress data m	nemory – value	e of FSR1 pre-	incremented (	not a physical	register) –	N/A	51, 74
FSR1H	_	1	-	_	Indirect Data	Memory Addr	ess Pointer 1	High Byte	0000	51, 73
FSR1L	Indirect Data	Memory Addre	ess Pointer 1 L	ow Byte					xxxx xxxx	51, 73
BSR	_	_	_	_	Bank Select I	Register			0000	52, 63
INDF2	Uses contents	s of FSR2 to a	ddress data m	nemory – value	of FSR2 not	changed (not	a physical regi	ster)	N/A	52, 73
POSTINC2	Uses contents	s of FSR2 to a	ddress data m	nemory – value	of FSR2 pos	t-incremented	(not a physica	l register)	N/A	52, 74
POSTDEC2	Uses contents	s of FSR2 to a	ddress data m	nemory – value	of FSR2 pos	t-decremented	(not a physic	al register)	N/A	52, 74
PREINC2				nemory – value					N/A	52, 74
PLUSW2	Uses contents value of FSR2		ddress data m	nemory – value	of FSR2 pre-	incremented (	not a physical	register) –	N/A	52, 74
FSR2H	_	_	_	_	Indirect Data	Memory Addr	ess Pointer 2	High Byte	0000	52, 73
FSR2L	Indirect Data	Memory Addre	ess Pointer 2 L	ow Byte					xxxx xxxx	52, 73
STATUS	_	_	_	N	OV	Z	DC	С	x xxxx	52, 71
TMR0H	Timer0 Regis	ter High Byte		ı	ı	ı	ı	1	0000 0000	52, 127
TMR0L	Timer0 Regis								xxxx xxxx	52, 127
TOCON	TMR0ON	T08BIT	TOCS	TOSE	PSA	T0PS2	T0PS1	T0PS0	1111 1111	52, 125
						on condition S	l .		1	

 $\textbf{Legend:} \qquad \textbf{x} = \textbf{unknown}, \ \textbf{u} = \textbf{unchanged}, \ \textbf{-} = \textbf{unimplemented}, \ \textbf{q} = \textbf{value depends on condition}. \ \textbf{Shaded cells are unimplemented}, \ \textbf{read as '0'}.$ 

- 2: The SBOREN bit is only available when BOREN<1:0> = 01; otherwise, the bit reads as '0'.
- 3: These registers and/or bits are not implemented on 28-pin devices and are read as '0'. Reset values are shown for 40/44-pin devices; individual unimplemented bits should be interpreted as '-'.
- 4: RA6 is configured as a port pin based on various primary oscillator modes. When the port pin is disabled, all of the associated bits read '0'.
- 5: RE3 is only available as a port pin when the MCLRE Configuration bit is clear; otherwise, the bit reads as '0'.
- **6:** RC5 and RC4 are only available as port pins when the USB module is disabled (UCON<3>=0).
- 7: I<sup>2</sup>C Slave mode only.

Note 1: Bit 21 of the TBLPTRU allows access to the device Configuration bits.

TABLE 5-2: REGISTER FILE SUMMARY (PIC18F2455/2550/4455/4550) (CONTINUED)

IABLE 5-2	. nec	ISIEN FI	LE SUMM	IANT (FIC	, 10FZ433/ 	2330/443	3/4330) (C			ı		
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		
OSCCON	IDLEN	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0	0100 q000	52, 32		
HLVDCON	VDIRMAG	_	IRVST	HLVDEN	HLVDL3	HLVDL2	HLVDL1	HLVDL0	0-00 0101	52, 279		
WDTCON	_	_	_		_	_	_	SWDTEN	0	52, 298		
RCON	IPEN	SBOREN <sup>(2)</sup>	_	RI	TO	PD	POR	BOR	0q-1 11q0	52, 44		
TMR1H	Timer1 Regis	Timer1 Register High Byte										
TMR1L	Timer1 Regis	xxxx xxxx	52, 133									
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T10SCEN	T1SYNC	TMR1CS	TMR10N	0000 0000	52, 129		
TMR2	Timer2 Regis	imer2 Register										
PR2	Timer2 Period	d Register							1111 1111	52, 136		
T2CON	_	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	52, 135		
SSPBUF	MSSP Receiv	ve Buffer/Trans	smit Register						xxxx xxxx	52, 194, 202		
SSPADD	MSSP Addre	ss Register in	I <sup>2</sup> C™ Slave m	ode. MSSP B	aud Rate Relo	ad Register in	I <sup>2</sup> C™ Master	mode.	0000 0000	52, 202		
SSPSTAT	SMP	CKE	D/A	Р	S	R/W	UA	BF	0000 0000	52, 194, 203		
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	52, 195, 204		
SSPCON2	GCEN	ACKSTAT	ACKDT/ ADMSK5 <sup>(7)</sup>	ACKEN/ ADMSK4 <sup>(7)</sup>	RCEN/ ADMSK3 <sup>(7)</sup>	PEN/ ADMSK2 <sup>(7)</sup>	RSEN/ ADMSK1 <sup>(7)</sup>	SEN	0000 0000	52, 205		
ADRESH	A/D Result R	egister High B	yte						xxxx xxxx	52, 268		
ADRESL	A/D Result R	egister Low By	/te						xxxx xxxx	52, 268		
ADCON0	_	_	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	00 0000	52, 259		
ADCON1	_	_	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	00 0qqq	52, 260		
ADCON2	ADFM	_	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0-00 0000	52, 261		
CCPR1H	Capture/Com	pare/PWM Re	gister 1 High E	Byte					xxxx xxxx	53, 142		
CCPR1L	Capture/Com	pare/PWM Re	gister 1 Low E	yte					xxxx xxxx	53, 142		
CCP1CON	P1M1 <sup>(3)</sup>	P1M0 <sup>(3)</sup>	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	53, 141, 149		
CCPR2H	Capture/Com	pare/PWM Re	gister 2 High E	Byte					xxxx xxxx	53, 142		
CCPR2L	Capture/Com	pare/PWM Re	gister 2 Low E	lyte					xxxx xxxx	53, 142		
CCP2CON	_	_	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	00 0000	53, 141		
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	_	WUE	ABDEN	0100 0-00	53, 240		
ECCP1DEL	PRSEN	PDC6 <sup>(3)</sup>	PDC5 <sup>(3)</sup>	PDC4 <sup>(3)</sup>	PDC3 <sup>(3)</sup>	PDC2 <sup>(3)</sup>	PDC1 <sup>(3)</sup>	PDC0 <sup>(3)</sup>	0000 0000	53, 158		
ECCP1AS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1 <sup>(3)</sup>	PSSBD0 <sup>(3)</sup>	0000 0000	53, 159		
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000 0000	53, 275		
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111	53, 269		
TMR3H	Timer3 Regis	ter High Byte							xxxx xxxx	53, 139		
TMR3L	Timer3 Regis		T		1	T	Т		xxxx xxxx	53, 139		
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON	0000 0000	53, 137		
SPBRGH	EUSART Bau	ud Rate Gener	ator Register I	High Byte					0000 0000	53, 241		
SPBRG	EUSART Bau	ud Rate Gener	ator Register L	ow Byte					0000 0000	53, 241		
RCREG	EUSART Red	ceive Register							0000 0000	53, 250		
TXREG	EUSART Tra	nsmit Register			1	Γ	Γ		0000 0000	53, 247		
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	53, 238		
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	53, 239		

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

Note 1: Bit 21 of the TBLPTRU allows access to the device Configuration bits.

- 2: The SBOREN bit is only available when BOREN<1:0> = 01; otherwise, the bit reads as '0'.
- 3: These registers and/or bits are not implemented on 28-pin devices and are read as '0'. Reset values are shown for 40/44-pin devices; individual unimplemented bits should be interpreted as '-'.
- 1: RA6 is configured as a port pin based on various primary oscillator modes. When the port pin is disabled, all of the associated bits read '0'.
- 5: RE3 is only available as a port pin when the MCLRE Configuration bit is clear; otherwise, the bit reads as '0'.
- 6: RC5 and RC4 are only available as port pins when the USB module is disabled (UCON<3> = 0).
- 7: I<sup>2</sup>C Slave mode only.

TABLE 5-2: REGISTER FILE SUMMARY (PIC18F2455/2550/4455/4550) (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
EEADR	EEPROM Ad	dress Register							0000 0000	53, 89
EEDATA	EEPROM Da	ta Register							0000 0000	53, 89
EECON2	EEPROM Co	ntrol Register	2 (not a physic	cal register)					0000 0000	53, 80
EECON1	EEPGD	CFGS	_	FREE	WRERR	WREN	WR	RD	xx-0 x000	53, 81
IPR2	OSCFIP	CMIP	USBIP	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	1111 1111	54, 107
PIR2	OSCFIF	CMIF	USBIF	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	0000 0000	54, 103
PIE2	OSCFIE	CMIE	USBIE	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	0000 0000	54, 105
IPR1	SPPIP <sup>(3)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	54, 106
PIR1	SPPIF <sup>(3)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	54, 102
PIE1	SPPIE <sup>(3)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	54, 104
OSCTUNE	INTSRC	-	_	TUN4	TUN3	TUN2	TUN1	TUN0	00 0000	54, 28
TRISE <sup>(3)</sup>	_		_	_	_	TRISE2	TRISE1	TRISE0	111	54, 124
TRISD(3)	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	54, 122
TRISC	TRISC7	TRISC6	_	_	_	TRISC2	TRISC1	TRISC0	11111	54, 119
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	54, 116
TRISA	_	TRISA6 <sup>(4)</sup>	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	-111 1111	54, 113
LATE <sup>(3)</sup>	_	_	_	_	_	LATE2	LATE1	LATE0	xxx	54, 124
LATD <sup>(3)</sup>	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	xxxx xxxx	54, 122
LATC	LATC7	LATC6	_	_	_	LATC2	LATC1	LATC0	xxxxx	54, 119
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx xxxx	54, 116
LATA	_	LATA6 <sup>(4)</sup>	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	-xxx xxxx	54, 113
PORTE	RDPU <sup>(3)</sup>	_	_	_	RE3 <sup>(5)</sup>	RE2 <sup>(3)</sup>	RE1 <sup>(3)</sup>	RE0 <sup>(3)</sup>	0 x000	54, 123
PORTD <sup>(3)</sup>	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	54, 122
PORTC	RC7	RC6	RC5 <sup>(6)</sup>	RC4 <sup>(6)</sup>	_	RC2	RC1	RC0	xxxx -xxx	54, 119
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	54, 116
PORTA	_	RA6 <sup>(4)</sup>	RA5	RA4	RA3	RA2	RA1	RA0	-x0x 0000	54, 113
UEP15	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	0 0000	55, 169
UEP14	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	0 0000	55, 169
UEP13	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	0 0000	55, 169
UEP12	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	0 0000	55, 169
UEP11	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	0 0000	55, 169
UEP10	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	0 0000	55, 169
UEP9	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	0 0000	55, 169
UEP8	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	0 0000	55, 169
UEP7	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	0 0000	55, 169
UEP6	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	0 0000	55, 169
UEP5	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	0 0000	55, 169
UEP4	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	0 0000	55, 169
UEP3	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	0 0000	55, 169
UEP2	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	0 0000	55, 169
UEP1	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	0 0000	55, 169
	_	_	_			EPOUTEN	EPINEN			55, 169
UEP0	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	0 0000	55, 1

 $\textbf{Legend:} \qquad \text{$x$ = unknown, $u$ = unchanged, $-$ = unimplemented, $q$ = value depends on condition. Shaded cells are unimplemented, read as '0'.}$ 

Note 1: Bit 21 of the TBLPTRU allows access to the device Configuration bits.

<sup>2:</sup> The SBOREN bit is only available when BOREN<1:0> = 01; otherwise, the bit reads as '0'.

<sup>3:</sup> These registers and/or bits are not implemented on 28-pin devices and are read as '0'. Reset values are shown for 40/44-pin devices; individual unimplemented bits should be interpreted as '-'.

<sup>4:</sup> RA6 is configured as a port pin based on various primary oscillator modes. When the port pin is disabled, all of the associated bits read '0'.

<sup>5:</sup> RE3 is only available as a port pin when the MCLRE Configuration bit is clear; otherwise, the bit reads as '0'.

**<sup>6:</sup>** RC5 and RC4 are only available as port pins when the USB module is disabled (UCON<3>=0).

<sup>7:</sup> I<sup>2</sup>C Slave mode only.

TABLE 5-2: REGISTER FILE SUMMARY (PIC18F2455/2550/4455/4550) (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
UCFG	UTEYE	UOEMON	_	UPUEN	UTRDIS	FSEN	PPB1	PPB0	00-0 0000	55, 166
UADDR	_	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0	-000 0000	55, 170
UCON	_	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	_	-0x0 000-	55, 164
USTAT	_	ENDP3	ENDP2	ENDP1	ENDP0	DIR	PPBI	_	-xxx xxx-	55, 168
UEIE	BTSEE	_	_	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE	00 0000	55, 182
UEIR	BTSEF	_	_	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF	00 0000	55, 181
UIE	_	SOFIE	STALLIE	IDLEIE	TRNIE	ACTVIE	UERRIE	URSTIE	-000 0000	55, 180
UIR	_	SOFIF	STALLIF	IDLEIF	TRNIF	ACTVIF	UERRIF	URSTIF	-000 0000	55, 178
UFRMH	_	_	_	_	_	FRM10	FRM9	FRM8	xxx	55, 170
UFRML	FRM7	FRM6	FRM5	FRM4	FRM3	FRM2	FRM1	FRM0	xxxx xxxx	55, 170
SPPCON <sup>(3)</sup>	_	_	_	_	_	_	SPPOWN	SPPEN	00	55, 187
SPPEPS <sup>(3)</sup>	RDSPP	WRSPP	_	SPPBUSY	ADDR3	ADDR2	ADDR1	ADDR0	00-0 0000	55, 191
SPPCFG <sup>(3)</sup>	CLKCFG1	CLKCFG0	CSEN	CLK1EN	WS3	WS2	WS1	WS0	0000 0000	55, 188
SPPDATA <sup>(3)</sup>	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	0000 0000	55, 192

 $\textbf{Legend:} \qquad \textbf{x} = \textbf{unknown}, \ \textbf{u} = \textbf{unchanged}, \ \textbf{-} = \textbf{unimplemented}, \ \textbf{q} = \textbf{value} \ \textbf{depends} \ \textbf{on condition}. \ \textbf{Shaded} \ \textbf{cells} \ \textbf{are unimplemented}, \ \textbf{read as '0'}.$ 

Note 1: Bit 21 of the TBLPTRU allows access to the device Configuration bits.

- 2: The SBOREN bit is only available when BOREN<1:0> = 01; otherwise, the bit reads as '0'.
- 3: These registers and/or bits are not implemented on 28-pin devices and are read as '0'. Reset values are shown for 40/44-pin devices; individual unimplemented bits should be interpreted as '-'.
- 4: RA6 is configured as a port pin based on various primary oscillator modes. When the port pin is disabled, all of the associated bits read '0'.
- 5: RE3 is only available as a port pin when the MCLRE Configuration bit is clear; otherwise, the bit reads as '0'.
- 6: RC5 and RC4 are only available as port pins when the USB module is disabled (UCON<3> = 0).
- 7: I<sup>2</sup>C Slave mode only.

#### 5.3.6 STATUS REGISTER

The STATUS register, shown in Register 5-2, contains the arithmetic status of the ALU. As with any other SFR, it can be the operand for any instruction.

If the STATUS register is the destination for an instruction that affects the Z, DC, C, OV or N bits, the results of the instruction are not written; instead, the STATUS register is updated according to the instruction performed. Therefore, the result of an instruction with the STATUS register as its destination may be different than intended. As an example, CLRF STATUS will set the Z bit and leave the remaining Status bits unchanged ('000u uluu').

It is recommended 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 that do not affect Status bits, see the instruction set summaries in Table 26-2 and Table 26-3.

Note: The C and DC bits operate as the Borrow and Digit Borrow bits, respectively, in subtraction.

#### REGISTER 5-2: 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 <sup>(1)</sup>	C <sup>(2)</sup>
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  $b\bar{t}$  is used for signed arithmetic (2's complement). It indicates whether the result was negative (ALU MSB = 1).

1 = Result was negative0 = 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 of the result) 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<sup>(1)</sup>

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<sup>(2)</sup>

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 mode 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 5.3.4 "General"

Purpose Register File") or a location in the Access Bank (Section 5.3.3 "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 5.3.2 "Bank Select Register (BSR)") 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.

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

	LFSR	FSR0, 100h	;	
NEXT	CLRF	POSTINC0	;	Clear INDF
			;	register then
			;	inc pointer
	BTFSS	FSROH, 1	;	All done with
			;	Bank1?
	BRA	NEXT	;	NO, clear next
CONTINU	E		;	YES, continue

## 5.4.3.1 FSR Registers and the INDF Operand

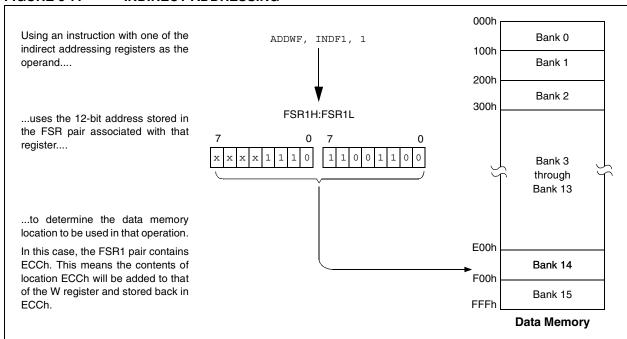
At the core of Indirect Addressing are three sets of registers: FSR0, FSR1 and FSR2. Each represents a pair of 8-bit registers: FSRnH and FSRnL. The four upper bits of the FSRnH register are not used, so each FSR pair holds a 12-bit value. This represents a value that can address the entire range of 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 5-7: INDIRECT ADDRESSING



© 2006 Microchip Technology Inc. Preliminary DS39632C-page 73

## 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 it 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 that 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, rollovers 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 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 eight additional two-word commands to the existing PIC18 instruction set: ADDFSR, ADDULNK, CALLW, MOVSF, MOVSS, PUSHL, SUBFSR and SUBULNK. 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);
- 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 in shown in Figure 5-8.

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 26.2.1** "Extended Instruction Syntax".

## FIGURE 5-8: COMPARING ADDRESSING OPTIONS FOR BIT-ORIENTED AND BYTE-ORIENTED INSTRUCTIONS (EXTENDED INSTRUCTION SET ENABLED)

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

#### When a = 0 and $f \ge 60h$ :

The instruction executes in Direct Forced mode. 'f' is interpreted as a location in the Access RAM between 060h and 0FFh. This is the same as the SFRs or locations F60h to 0FFh (Bank 15) of data memory.

Locations below 60h are not available in this addressing mode.

#### When a = 0 and $f \le 5Fh$ :

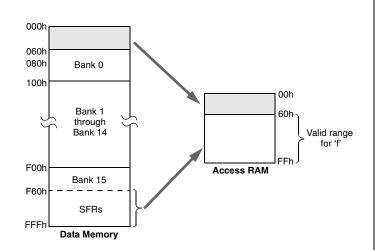
The instruction executes in Indexed Literal Offset mode. 'f' is interpreted as an offset to the address value in FSR2. The two are added together to obtain the address of the target register for the instruction. The address can be anywhere in the data memory space.

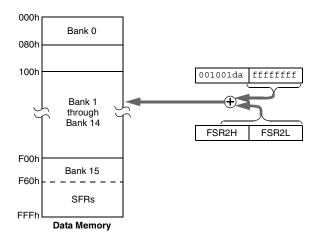
Note that in this mode, the correct syntax is now:  $\label{eq:add_power} \text{ADDWF } \left[ k \right], \text{ d}$ 

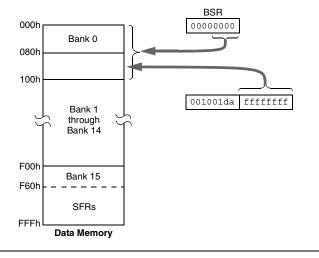
where 'k' is the same as 'f'.

#### 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.







### 5.6.3 MAPPING THE ACCESS BANK IN INDEXED LITERAL OFFSET MODE

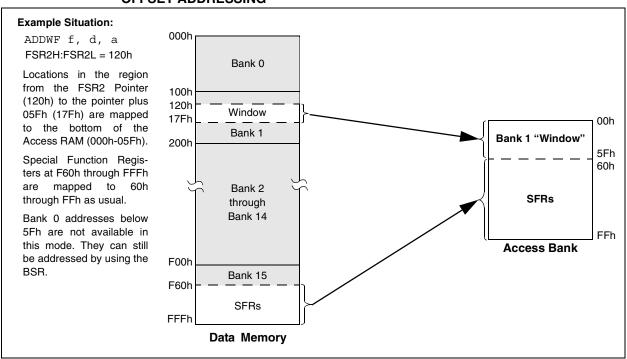
The use of Indexed Literal Offset Addressing mode effectively changes how the lower portion of Access RAM (00h to 5Fh) is mapped. Rather than containing just the contents of the bottom half of Bank 0, this mode maps the contents from Bank 0 and a user-defined "window" that can be located anywhere in 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 5.3.3 "Access Bank"). An example of Access Bank remapping in this addressing mode is shown in Figure 5-9.

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 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 5-9: REMAPPING THE ACCESS BANK WITH INDEXED LITERAL OFFSET ADDRESSING



© 2006 Microchip Technology Inc. Preliminary DS39632C-page 77

NOTES:

#### 6.0 FLASH PROGRAM MEMORY

The Flash program memory is readable, writable and erasable, during normal operation over the entire VDD

A read from program memory is executed on one byte at a time. A write to program memory is executed on blocks of 32 bytes at a time. Program memory is erased in blocks of 64 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

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.

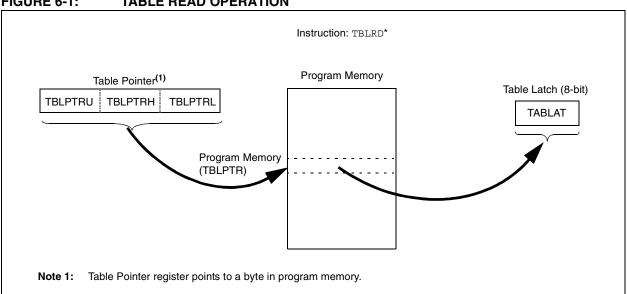
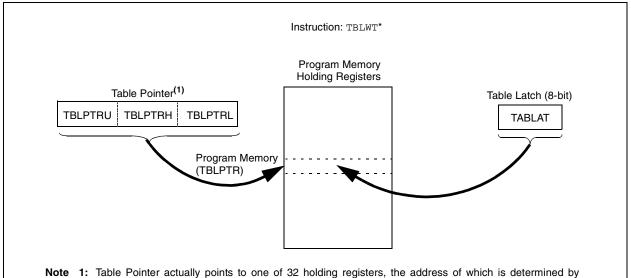


FIGURE 6-1: **TABLE READ OPERATION** 

**Preliminary** © 2006 Microchip Technology Inc. DS39632C-page 79

#### FIGURE 6-2: TABLE WRITE OPERATION



Note 1: Table Pointer actually points to one of 32 holding registers, the address of which is determined by TBLPTRL<4:0>. The process for physically writing data to the program memory array is discussed in Section 6.5 "Writing to Flash Program Memory".

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

The CFGS control bit determines if the access will be to the Configuration/Calibration registers or to program memory/data EEPROM memory. When set, subsequent operations will operate on Configuration registers regardless of EEPGD (see Section 25.0 "Special Features of the CPU"). When clear, memory selection access is determined by EEPGD.

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 WREN 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.

**Note:** The EEIF interrupt flag bit (PIR2<4>) is set when the write is complete. It must be cleared in software.

#### REGISTER 6-1: EECON1: DATA EEPROM CONTROL REGISTER 1

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	_	FREE	WRERR <sup>(1)</sup>	WREN	WR	RD
bit 7							bit 0

Legend:	S = Settable 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 **EEPGD:** Flash Program or Data EEPROM Memory Select bit

1 = Access Flash program memory

0 = Access data EEPROM memory

bit 6 CFGS: Flash Program/Data EEPROM or Configuration Select bit

1 = Access Configuration registers

0 = Access Flash program or data EEPROM memory

bit 5 Unimplemented: Read as '0'

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/Data EEPROM Error Flag bit<sup>(1)</sup>

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/Data EEPROM Write Enable bit

1 = Allows write cycles to Flash program/data EEPROM

0 = Inhibits write cycles to Flash program/data EEPROM

bit 1 WR: Write Control bit

1 = Initiates a data EEPROM erase/write cycle or 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 to the EEPROM is complete

bit 0 RD: Read Control bit

1 = Initiates an EEPROM read (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software. RD bit cannot be set when EEPGD = 1 or CFGS = 1.)

0 = Does not initiate an EEPROM read

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

#### 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, 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 five LSbs of the Table Pointer register (TBLPTR<4:0>) determine which of the 32 program memory holding registers is written to. When the timed write to program memory begins (via the WR bit), the 16 MSbs of the TBLPTR (TBLPTR<21:6>) determine which program memory block of 32 bytes is written to. For more detail, see Section 6.5 "Writing to Flash Program Memory".

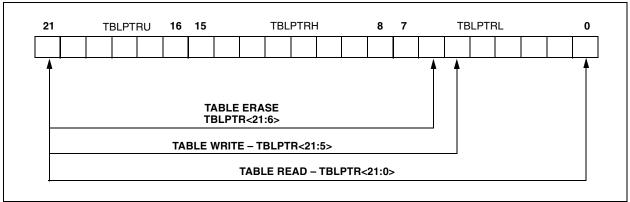
When an erase of program memory is executed, the 16 MSbs of the Table Pointer register (TBLPTR<21:6>) point to the 64-byte block that will be erased. The Least Significant bits (TBLPTR<5:0>) are ignored.

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

TABLE C.4.	TABLE BOILTED	ODEDATIONS WITH	AND INCTRUCTIONS
IABLE 6-1.	IARIFPUNIER	OPERATIONS WITH TRURD	AND TRIMT 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





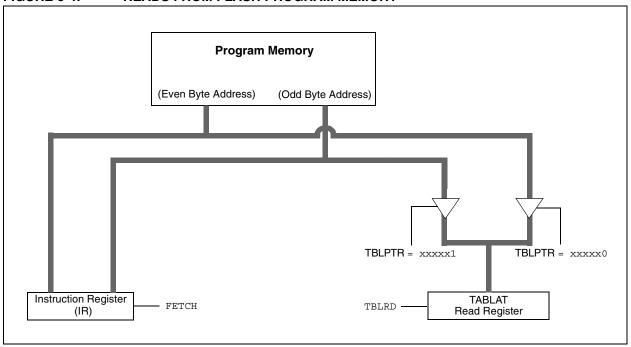
## 6.3 Reading the Flash Program Memory

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 MOVWF	CODE_ADDR_UPPER TBLPTRU	; Load TBLPTR with the base ; address of the word
	MOVLW	CODE ADDR HIGH	,
	MOVWF	TBLPTRH	
	MOVLW	CODE_ADDR_LOW	
	MOVWF	TBLPTRL	
READ_WORD			
	TBLRD*+		; read into TABLAT and increment
	MOVF	TABLAT, W	; get data
	MOVWF	WORD_EVEN	
	TBLRD*+		; read into TABLAT and increment
	MOVF	TABLAT, W	; get data
	MOVF	WORD_ODD	

© 2006 Microchip Technology Inc. Preliminary DS39632C-page 83

#### 6.4 Erasing Flash Program Memory

The minimum erase block is 32 words or 64 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 micro-controller itself, a block of 64 bytes of program memory is erased. The Most Significant 16 bits of the TBLPTR<21:6> point to the block being erased. TBLPTR<5:0> are ignored.

The EECON1 register commands the erase operation. The EEPGD bit must be set to point to the Flash program memory. 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 is:

- Load Table Pointer register with address of row being erased.
- 2. Set the EECON1 register for the erase operation:
  - set EEPGD bit to point to program memory;
  - clear the CFGS bit to access program memory;
  - set WREN bit to enable writes;
  - set FREE bit to enable the erase.
- 3. Disable interrupts.
- 4. Write 55h to EECON2.
- 5. Write 0AAh to EECON2.
- Set the WR bit. This will begin the Row Erase cycle.
- 7. The CPU will stall for duration of the erase (about 2 ms using internal timer).
- 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, EEPGD	; point to Flash program memory
	BCF	EECON1, CFGS	; access Flash program memory
	BSF	EECON1, WREN	; enable write to memory
	BSF	EECON1, FREE	; enable Row Erase operation
	BCF	INTCON, GIE	; disable interrupts
Required	MOVLW	55h	
Sequence	MOVWF	EECON2	; write 55h
	MOVLW	0AAh	
	MOVWF	EECON2	; write OAAh
	BSF	EECON1, WR	; start erase (CPU stall)
	BSF	INTCON, GIE	; re-enable interrupts

#### 6.5 Writing to Flash Program Memory

The minimum programming block is 16 words or 32 bytes. Word or byte programming is not supported.

Table writes are used internally to load the holding registers needed to program the Flash memory. There are 32 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 32 times for each programming operation. All of the table write operations will essentially be short writes because only the holding registers are written. At the end of updating the 32 holding registers, the EECON1 register must be written to in order to start the programming operation with a long write.

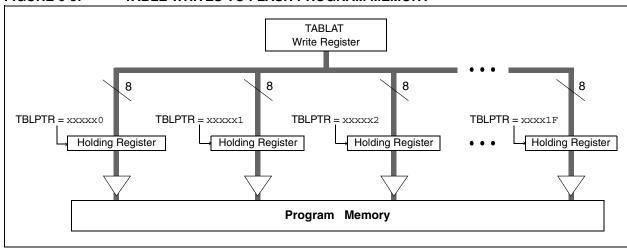
The long write is necessary for programming the internal Flash. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer.

The EEPROM 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: The default value of the holding registers on device Resets and after write operations is FFh. A write of FFh to a holding register does not modify that byte. This means that individual bytes of program memory may be modified, provided that the change does not attempt to change any bit from a '0' to a '1'. When modifying individual bytes, it is not necessary to load all 32 holding registers

before executing a write operation.

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:

- Read 64 bytes into RAM.
- Update data values in RAM as necessary.
- 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.
- Write 32 bytes into the holding registers with auto-increment.
- 7. Set the EECON1 register for the write operation:
  - set EEPGD bit to point to program memory;
  - · clear the CFGS bit to access program memory;
  - set WREN 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 (about 2 ms using internal timer).
- 13. Re-enable interrupts.
- Repeat steps 6 through 14 once more to write 64 bytes.
- 15. Verify the memory (table read).

This procedure will require about 8 ms to update one row of 64 bytes of memory. An example of the required code is given in Example 6-3.

Note: Before setting the WR bit, the Table Pointer address needs to be within the intended address range of the 32 bytes in the holding register.

#### **EXAMPLE 6-3: WRITING TO FLASH PROGRAM MEMORY**

EXAMPLE 6-3:	WRITING	G TO FLASH PROGRA	M MEMORY
	MOVLW	D'64'	; number of bytes in erase block
	MOVWF	COUNTER	,
	MOVLW	BUFFER ADDR HIGH	; point to buffer
	MOVWF	FSR0H	, point to barrer
	MOVLW	BUFFER ADDR LOW	
	MOVWF	FSR0L	
			I and mpipmp with the have
	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	
READ_BLOCK			
	TBLRD*+		; read into TABLAT, and inc
	MOVF	TABLAT, W	; get data
	MOVWF	POSTINCO	; store data
	DECFSZ	COUNTER	; done?
	BRA	READ_BLOCK	; repeat
MODIFY_WORD			
	MOVLW	DATA_ADDR_HIGH	; point to buffer
	MOVWF	FSROH	
	MOVLW	DATA ADDR LOW	
	MOVWF	FSROL	
	MOVLW	NEW DATA LOW	; update buffer word
	MOVWF	POSTINCO	•
	MOVLW	NEW DATA HIGH	
	MOVWF	INDF0	
ERASE BLOCK	110 V W1	INDIO	
DICADI_DIOCK	MOVLW	CODE ADDR UPPER	; load TBLPTR with the base
	MOVUE	TBLPTRU	; address of the memory block
	MOVLW	CODE_ADDR_HIGH	
	MOVWF	TBLPTRH	
	MOVLW	CODE_ADDR_LOW	
	MOVWF	TBLPTRL	
	BSF	EECON1, EEPGD	; point to Flash program memory
	BCF	EECON1, CFGS	; access Flash program memory
	BSF	EECON1, WREN	; enable write to memory
	BSF	EECON1, FREE	; enable Row Erase operation
	BCF	INTCON, GIE	; disable interrupts
	MOVLW	55h	
Required	MOVWF	EECON2	; write 55h
Sequence	MOVLW	0AAh	
	MOVWF	EECON2	; write OAAh
	BSF	EECON1, WR	; start erase (CPU stall)
	BSF	INTCON, GIE	; re-enable interrupts
	TBLRD*-	•	; dummy read decrement
	MOVLW	BUFFER ADDR HIGH	; point to buffer
	MOVWF	FSR0H	, <u>F</u>
	MOVLW	BUFFER ADDR LOW	
	MOVWF	FSR0L	
	MOVLW	D'2'	
	MOVLW		
שמתתום קייום ארם		COUNTER1	
WRITE_BUFFER_BACK		D1221	mumber of backer to 1-1-1-1
	MOVIN	D'32'	; number of bytes in holding register
MD TOD STORE ST.	MOVWF	COUNTER	
WRITE_BYTE_TO_HRE			
	MOVF	POSTINCO, W	; get low byte of buffer data
	MOVWF	TABLAT	; present data to table latch
	TBLWT+*		; write data, perform a short write
			; to internal TBLWT holding register.
	DECFSZ	COUNTER	; loop until buffers are full
	BRA	WRITE_WORD_TO_HREGS	

#### **EXAMPLE 6-3: WRITING TO FLASH PROGRAM MEMORY (CONTINUED)**

PROGRAM_MEMORY			
	BSF	EECON1, EEPGD	; point to Flash program memory
	BCF	EECON1, CFGS	; access Flash program memory
	BSF	EECON1, WREN	; enable write to memory
	BCF	INTCON, GIE	; disable interrupts
	MOVLW	55h	
Required	MOVWF	EECON2	; write 55h
Sequence	MOVLW	0AAh	
	MOVWF	EECON2	; write OAAh
	BSF	EECON1, WR	; start program (CPU stall)
	DECFSZ	COUNTER1	
	BRA	WRITE_BUFFER_BACK	
	BSF	INTCON, GIE	; re-enable interrupts
	BCF	EECON1, WREN	; disable write to memory

#### 6.5.2 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.

### 6.5.3 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.

### 6.5.4 PROTECTION AGAINST SPURIOUS WRITES

To protect against spurious writes to Flash program memory, the write initiate sequence must also be followed. See Section 25.0 "Special Features of the CPU" for more detail.

## 6.6 Flash Program Operation During Code Protection

See Section 25.5 "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 <sup>(1)</sup>	Program Me	emory Table F	Pointer Uppe	r Byte (TBLP	TR<20:16>)	51
TBLPTRH	Program M	emory Table	Pointer H	igh Byte (TB	LPTR<15:8	>)			51
TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								51
TABLAT	Program Memory Table Latch								51
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INTOIE	RBIE	TMR0IF	INTOIF	RBIF	51
EECON2	EEPROM (	Control Regi	ster 2 (not	a physical re	egister)				53
EECON1	EEPGD	CFGS	_	FREE	WRERR	WREN	WR	RD	53
IPR2	OSCFIP	CMIP	USBIP	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	54
PIR2	OSCFIF	CMIF	USBIF	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	54
PIE2	OSCFIE	CMIE	USBIE	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	54

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used during Flash/EEPROM access.

Note 1: Bit 21 of the TBLPTRU allows access to the device Configuration bits.

NOTES:

#### 7.0 DATA EEPROM MEMORY

The data EEPROM is a nonvolatile memory array, separate from the data RAM and program memory, that is used for long-term storage of program data. It is not directly mapped in either the register file or program memory space, but is indirectly addressed through the Special Function Registers (SFRs). The EEPROM is readable and writable during normal operation over the entire VDD range.

Four SFRs are used to read and write to the data EEPROM as well as the program memory. They are:

- EECON1
- EECON2
- EEDATA
- EEADR

The data EEPROM allows byte read and write. When interfacing to the data memory block, EEDATA holds the 8-bit data for read/write and the EEADR register holds the address of the EEPROM location being accessed.

The EEPROM data memory is rated for high erase/write cycle endurance. A byte write automatically erases the location and writes the new data (erase-before-write). The write time is controlled by an on-chip timer; it will vary with voltage and temperature as well as from chip to chip. Please refer to parameter D122 (Table 28-1 in Section 28.0 "Electrical Characteristics") for exact limits.

#### 7.1 EECON1 and EECON2 Registers

Access to the data EEPROM is controlled by two registers: EECON1 and EECON2. These are the same registers which control access to the program memory and are used in a similar manner for the data EEPROM.

The EECON1 register (Register 7-1) is the control register for data and program memory access. Control bit, EEPGD, determines if the access will be to program or data EEPROM memory. When clear, operations will access the data EEPROM memory. When set, program memory is accessed.

Control bit, CFGS, determines if the access will be to the Configuration registers or to program memory/data EEPROM memory. When set, subsequent operations access Configuration registers. When CFGS is clear, the EEPGD bit selects either program Flash or data EEPROM memory.

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 WREN 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.

Note: The EEIF interrupt flag bit (PIR2<4>) is set when the write is complete. It must be cleared in software.

Control bits, RD and WR, start read and erase/write operations, respectively. These bits are set by firmware and cleared by hardware at the completion of the operation.

The RD bit cannot be set when accessing program memory (EEPGD = 1). Program memory is read using table read instructions. See **Section 6.1** "**Table Reads and Table Writes**" regarding table reads.

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.

#### REGISTER 7-1: EECON1: DATA EEPROM CONTROL REGISTER 1

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	_	FREE	WRERR <sup>(1)</sup>	WREN	WR	RD
bit 7							bit 0

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

bit 7 **EEPGD:** Flash Program or Data EEPROM Memory Select bit

1 = Access Flash program memory

0 = Access data EEPROM memory

bit 6 CFGS: Flash Program/Data EEPROM or Configuration Select bit

1 = Access Configuration registers

0 = Access Flash program or data EEPROM memory

bit 5 Unimplemented: Read as '0'

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/Data EEPROM 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/Data EEPROM Write Enable bit

1 = Allows write cycles to Flash program/data EEPROM

0 = Inhibits write cycles to Flash program/data EEPROM

bit 1 WR: Write Control bit

1 = Initiates a data EEPROM erase/write cycle or 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 to the EEPROM is complete

bit 0 RD: Read Control bit

1 = Initiates an EEPROM read (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software. RD bit cannot be set when EEPGD = 1 or CFGS = 1.)

0 = Does not initiate an EEPROM read

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

## 7.2 Reading the Data EEPROM Memory

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

The basic process is shown in Example 7-1.

## 7.3 Writing to the Data EEPROM Memory

To write an EEPROM data location, the address must first be written to the EEADR register and the data written to the EEDATA register. The sequence in Example 7-2 must be followed to initiate the write cycle.

The write will not begin if this sequence is not exactly followed (write 55h to EECON2, write 0AAh to EECON2, then set WR bit) for each byte. It is strongly recommended that interrupts be disabled during this code segment.

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

After a write sequence has been initiated, EECON1, EEADR and EEDATA cannot be modified. The WR bit will be inhibited from being set unless the WREN bit is set. The WREN bit must be set on a previous instruction. Both WR and WREN cannot be set with the same instruction.

At the completion of the write cycle, the WR bit is cleared in hardware and the EEPROM Interrupt Flag bit (EEIF) is set. The user may either enable this interrupt, or poll this bit. EEIF must be cleared by software.

#### 7.4 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.

#### **EXAMPLE 7-1: DATA EEPROM READ**

```
MOVLW
       DATA EE ADDR
MOVWF
                          ; Lower bits of Data Memory Address to read
       EEADR
       EECON1, EEPGD
BCF
                         ; Point to DATA memory
BCF
       EECON1, CFGS
                          ; Access EEPROM
       EECON1, RD
                          ; EEPROM Read
BSF
MOVF
       EEDATA, W
                          ; W = EEDATA
```

#### **EXAMPLE 7-2: DATA EEPROM WRITE**

```
MOVLW
                  DATA_EE_ADDR
          MOVWF
                  EEADR
                                     ; Lower bits of Data Memory Address to write
          MOVLW
                  DATA EE DATA
                                     ;
           MOVWF
                                     ; Data Memory Value to write
                  EEDATA
          BCF
                  EECON1, EPGD
                                     ; Point to DATA memory
                  EECON1, CFGS
          BCF
                                     ; Access EEPROM
          BSF
                  EECON1, WREN
                                     ; Enable writes
                  INTCON, GIE
           BCF
                                     ; Disable Interrupts
          MOVLW
                  55h
                                     ; Write 55h
Required
          MOVWF
                  EECON2
          MOVLW
                  0AAh
Sequence
          MOVWF EECON2
                                     ; Write OAAh
           BSF
                  EECON1, WR
                                    ; Set WR bit to begin write
          BSF
                  INTCON, GIE
                                     ; Enable Interrupts
                                     : User code execution
           BCF
                  EECON1, WREN
                                     ; Disable writes on write complete (EEIF set)
```

#### 7.5 Operation During Code-Protect

Data EEPROM memory has its own code-protect bits in Configuration Words. External read and write operations are disabled if code protection is enabled.

The microcontroller itself can both read and write to the internal data EEPROM regardless of the state of the code-protect Configuration bit. Refer to Section 25.0 "Special Features of the CPU" for additional information.

#### 7.6 Protection Against Spurious Write

There are conditions when the device may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been implemented. On power-up, the WREN bit is cleared. In addition, writes to the EEPROM are blocked during the Power-up Timer period (TPWRT, parameter 33, Table 28-12).

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

#### 7.7 Using the Data EEPROM

The data EEPROM is a high endurance, byte addressable array that has been optimized for the storage of frequently changing information (e.g., program variables or other data that are updated often). Frequently changing values will typically be updated more often than specification D124 or D124A. If this is not the case, an array refresh must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory.

A simple data EEPROM refresh routine is shown in Example 7-3.

**Note:** If data EEPROM is only used to store constants and/or data that changes rarely, an array refresh is likely not required. See specification D124 or D124A.

#### **EXAMPLE 7-3: DATA EEPROM REFRESH ROUTINE**

	CLRF	EEADR	; Start at address 0
	BCF	EECON1, CFGS	; Set for memory
	BCF	EECON1, EEPGD	; Set for Data EEPROM
	BCF	INTCON, GIE	; Disable interrupts
	BSF	EECON1, WREN	; Enable writes
Loop			; Loop to refresh array
	BSF	EECON1, RD	; Read current address
	MOVLW	55h	i
Required	MOVWF	EECON2	; Write 55h
Sequence	MOVLW	0AAh	i
	MOVWF	EECON2	; Write OAAh
	BSF	EECON1, WR	; Set WR bit to begin write
	BTFSC	EECON1, WR	; Wait for write to complete
	BRA	\$-2	
	INCFSZ	EEADR, F	; Increment address
	BRA	LOOP	; Not zero, do it again
	BCF	EECON1, WREN	; Disable writes
	BSF	INTCON, GIE	; Enable interrupts

TABLE 7-1: REGISTERS ASSOCIATED WITH DATA EEPROM MEMORY

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	51
EEADR	EEPROM A	ddress Regis	ster						53
EEDATA	EEPROM D	ata Register							53
EECON2	EEPROM C	ontrol Regist	ter 2 (not a p	ohysical reg	ister)				53
EECON1	EEPGD	CFGS	_	FREE	WRERR	WREN	WR	RD	53
IPR2	OSCFIP	CMIP	USBIP	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	54
PIR2	OSCFIF	CMIF	USBIF	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	54
PIE2	OSCFIE	CMIE	USBIE	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	54

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used during Flash/EEPROM access.

© 2006 Microchip Technology Inc. Preliminary DS39632C-page 93

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  $8 \times 8$  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
BTFSC	ARG2, SB	; Test Sign Bit
SUBWF	PRODH, F	; PRODH = PRODH
		; - ARG1
MOVF	ARG2, W	
BTFSC	ARG1, SB	; Test Sign Bit
SUBWF	PRODH, F	; PRODH = PRODH
		; - ARG2

TABLE 8-1: PERFORMANCE COMPARISON FOR VARIOUS MULTIPLY OPERATIONS

<b>D</b> .:.	M. 102 - 1 - M 1 - 1	Program	Cycles	Time			
Routine	Multiply Method	Memory (Words)	(Max)	@ 40 MHz	@ 10 MHz	@ 4 MHz	
8 x 8 unsigned	Without hardware multiply	13	69	6.9 μs	27.6 μs	69 μs	
o x o unsigned	Hardware multiply	1	1	100 ns	400 ns	1 μs	
9 v 9 signed	Without hardware multiply	33	91	9.1 μs	36.4 μs	91 μs	
8 x 8 signed	Hardware multiply	6	6	600 ns	2.4 μs	6 μs	
16 v 16 upoignod	Without hardware multiply	21	242	24.2 μs	96.8 μs	242 μs	
16 x 16 unsigned	Hardware multiply	28	28	2.8 μs	11.2 μs	28 μs	
16 v 16 signed	Without hardware multiply	52	254	25.4 μs	102.6 μs	254 μs	
16 x 16 signed	Hardware multiply	35	40	4.0 μs	16.0 μs	40 μs	

Example 8-3 shows the sequence to do a 16 x 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

```
RES3:RES0 = ARG1H:ARG1L \bullet ARG2H:ARG2L

= (ARG1H \bullet ARG2H \bullet 2<sup>16</sup>) +

(ARG1H \bullet ARG2L \bullet 2<sup>8</sup>) +

(ARG1L \bullet ARG2H \bullet 2<sup>8</sup>) +

(ARG1L \bullet ARG2L)
```

## EXAMPLE 8-3: 16 x 16 UNSIGNED MULTIPLY ROUTINE

```
MOVF
       ARG1L, W
MULWF
       ARG2L
                      ; ARG1L * ARG2L->
                      ; PRODH: PRODL
MOVFF
       PRODH, RES1
                      ;
MOVFF
       PRODL, RESO
MOVF
       ARG1H, W
MULWF
       ARG2H
                      ; ARG1H * ARG2H->
                      ; PRODH:PRODL
       PRODH, RES3
MOVFF
MOVFF
       PRODL, RES2
MOVF
       ARG1L, W
                      ; ARG1L * ARG2H->
MULWF
       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
                      ; ARG1H * ARG2L->
MULWF
       ARG2L
                      ; PRODH:PRODL
MOVF
       PRODL, W
                      ; Add cross
ADDWF
       RES1, F
       PRODH, W
MOVF
                      ; products
ADDWFC RES2, F
CLRF
       WREG
                      ;
ADDWFC RES3, F
                      ;
```

Example 8-4 shows the sequence to do a 16 x 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

```
RES3:RES0 = ARG1H:ARG1L • ARG2H:ARG2L

= (ARG1H • ARG2H • 2^{16}) + (ARG1H • ARG2L • 2^{8}) + (ARG1L • ARG2H • 2^{8}) + (ARG1L • ARG2L) + (-1 • ARG2H < -2 * ARG1H:ARG1L • 2^{16}) + (-1 • ARG1H < -7 * • ARG2H:ARG2L • 2^{16})
```

## EXAMPLE 8-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```
MOVF
           ARG1L, W
                        ; ARG1L * ARG2L ->
   MULWF
          ARG2L
                        ; PRODH:PRODL
   MOVFF
           PRODH, RES1
           PRODL, RESO
   MOVFF
   MOVF
           ARG1H, W
                        ; ARG1H * ARG2H ->
   MULWF
          ARG2H
                        ; PRODH: PRODL
   MOVFF
           PRODH, RES3
   MOVFF
           PRODL, RES2
   MOVF
           ARG1L.W
                        ; ARG1L * ARG2H ->
   MULWF
          ARG2H
                        ; PRODH: PRODL
   MOVF
           PRODL, W
                       ; Add cross
          RES1, F
   ADDWF
          PRODH, W
                        ; products
   MOVF
   ADDWFC RES2, F
   CLRF
           WREG
   ADDWFC RES3, F
   MOVF
           ARG1H, W
                       ; ARG1H * ARG2L ->
   MULWF
          ARG2L
                        ; PRODH: PRODL
           PRODL, W
   MOVF
                        ; Add cross
   ADDWF
          RES1, F
                        ; products
   MOVF
          PRODH, W
   ADDWFC RES2, F
   CLRF
          WREG
   ADDWFC RES3, F
   BTFSS
          ARG2H, 7
                        ; ARG2H:ARG2L neg?
   BRA
           SIGN ARG1
                       ; no, check ARG1
   MOVF
          ARG1L, W
                        ;
   SUBWF
          RES2
   MOVF
          ARG1H. W
   SUBWFB RES3
SIGN_ARG1
   BTFSS
          ARG1H, 7
                       ; ARG1H:ARG1L neg?
           CONT_CODE
   BRA
                       ; no, done
   MOVF
           ARG2L, W
   SUBWF
          RES2
   MOVF
           ARG2H, W
   SUBWFB RES3
CONT_CODE
```

#### 9.0 INTERRUPTS

The PIC18F2455/2550/4455/4550 devices have multiple interrupt sources and an interrupt priority feature that allows each interrupt source to be assigned a high priority level or a low priority level. The high priority interrupt vector is at 000008h and the low priority interrupt vector is at 000018h. High priority interrupt events will interrupt any low priority interrupts that may be in progress.

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

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

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.

Each interrupt source has three bits to control its operation. The functions of these bits 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 000008h or 000018h, 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 PICmicro® 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 000008h 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 (000008h or 000018h). 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 INT 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.

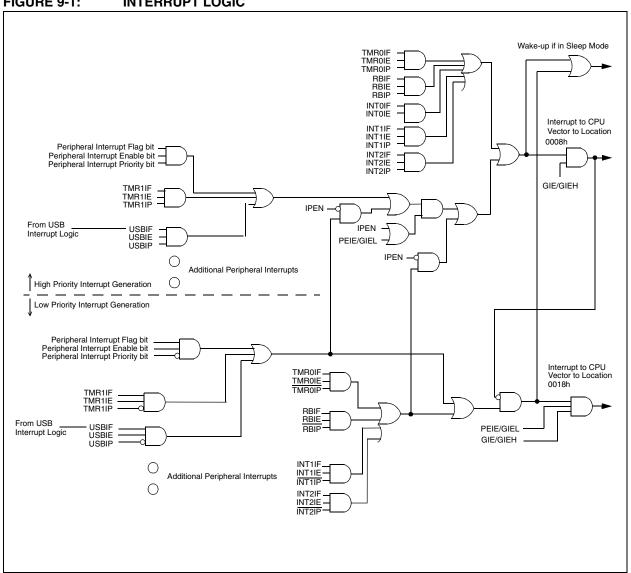
#### 9.1 USB Interrupts

Unlike other peripherals, the USB module is capable of generating a wide range of interrupts for many types of events. These include several types of normal communication and status events and several module level error events.

To handle these events, the USB module is equipped with its own interrupt logic. The logic functions in a manner similar to the microcontroller level interrupt funnel, with each interrupt source having separate flag and enable bits. All events are funneled to a single device level interrupt, USBIF (PIR2<5>). Unlike the device level interrupt logic, the individual USB interrupt events cannot be individually assigned their own priority. This is determined at the device level interrupt funnel for all USB events by the USBIP bit.

For additional details on USB interrupt logic, refer to **Section 17.5 "USB Interrupts"**.

FIGURE 9-1: **INTERRUPT LOGIC** 



Note:

#### 9.2 INTCON Registers

The INTCON registers are readable and writable registers which contain various enable, priority and flag bits.

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 <sup>(1)</sup>
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

GIE/GIEH: Global Interrupt Enable bit

When IPEN = 0:
1 = Enables all unmasked interrupts

When IPEN = 1:

1 = Enables all high priority interrupts

0 = Disables all high priority interrupts

bit 6 PEIE/GIEL: Peripheral Interrupt Enable bit

0 = Disables all interrupts

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 interrupt0 = Disables the RB port change interrupt

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<sup>(1)</sup>

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.

bit 2

#### REGISTER 9-2: INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
RBPU	INTEDG0	INTEDG1	INTEDG2	_	TMR0IP	_	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 edge0 = Interrupt on falling edge

bit 5 INTEDG1: External Interrupt 1 Edge Select bit

1 = Interrupt on rising edge0 = 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 **Unimplemented:** Read as '0'

bit 2 TMR0IP: TMR0 Overflow Interrupt Priority bit

1 = High priority0 = Low priority

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

bit 0 RBIP: RB Port Change Interrupt Priority bit

1 = High priority0 = 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	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
INT2IP	INT1IP	_	INT2IE	INT1IE	_	INT2IF	INT1IF
bit 7							bit 0

R = Readable bit U = Unimplemented bit, read as '0' W = Writable bit -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

Legend:

bit 6 INT1IP: INT1 External Interrupt Priority bit

> 1 = High priority 0 = Low priority

bit 5 Unimplemented: Read as '0'

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 Unimplemented: Read as '0'

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

Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding Note: 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.

**Preliminary** © 2006 Microchip Technology Inc. DS39632C-page 101

#### 9.3 PIR Registers

The PIR registers contain the individual flag bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are two Peripheral Interrupt Request (Flag) registers (PIR1 and PIR2).

- 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>).
  - 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
SPPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

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

bit 7 SPPIF: Streaming Parallel Port Read/Write Interrupt Flag bit<sup>(1)</sup>

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 RCIF: EUSART Receive Interrupt Flag bit

1 = The EUSART receive buffer, RCREG, is full (cleared when RCREG is read)

0 = The EUSART receive buffer is empty

bit 4 **TXIF:** EUSART Transmit Interrupt Flag bit

1 = The EUSART transmit buffer, TXREG, is empty (cleared when TXREG is written)

0 = The EUSART transmit buffer is full

bit 3 SSPIF: Master Synchronous Serial Port Interrupt Flag bit

1 = The transmission/reception is complete (must be cleared in software)

0 = Waiting to transmit/receive

bit 2 **CCP1IF:** CCP1 Interrupt Flag bit

#### Capture mode:

1 = A TMR1 register capture occurred (must be cleared in software)

0 = No TMR1 register capture occurred

#### Compare mode:

1 = A TMR1 register compare match occurred (must be cleared in software)

0 = No TMR1 register compare match occurred

#### PWM mode:

Unused in this mode.

bit 1 TMR2IF: TMR2 to PR2 Match Interrupt Flag bit

1 = TMR2 to PR2 match occurred (must be cleared in software)

0 = No TMR2 to PR2 match occurred

bit 0 TMR1IF: TMR1 Overflow Interrupt Flag bit

1 = TMR1 register overflowed (must be cleared in software)

0 = TMR1 register did not overflow

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

#### REGISTER 9-5: PIR2: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIF	CMIF	USBIF	EEIF	BCLIF	HLVDIF	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 = System oscillator failed, clock input has changed to INTOSC (must be cleared in software)

0 = System clock operating

bit 6 CMIF: Comparator Interrupt Flag bit

1 = Comparator input has changed (must be cleared in software)

0 = Comparator input has not changed

bit 5 USB Interrupt Flag bit

1 = USB has requested an interrupt (must be cleared in software)

0 = No USB interrupt request

bit 4 **EEIF:** Data EEPROM/Flash Write Operation Interrupt Flag bit

1 = The write operation is complete (must be cleared in software)

0 = The write operation is not complete or has not been started

bit 3 **BCLIF:** Bus Collision Interrupt Flag bit

1 = A bus collision has occurred (must be cleared in software)

0 = No bus collision occurred

bit 2 **HLVDIF:** High/Low-Voltage Detect Interrupt Flag bit

1 = A high/low-voltage condition occurred (must be cleared in software)

0 = No high/low-voltage event has occurred

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: CCP2 Interrupt Flag bit

Capture mode:

1 = A TMR1 or TMR3 register capture occurred (must be cleared in software)

0 = No TMR1 or TMR3 register capture occurred

Compare mode:

1 = A TMR1 or TMR3 register compare match occurred (must be cleared in software)

0 = No TMR1 or TMR3 register compare match occurred

PWM mode:

Unused in this mode.

#### 9.4 PIE Registers

Legend:

R = Readable bit

The PIE registers contain the individual enable bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are two Peripheral Interrupt Enable registers (PIE1 and PIE2). When IPEN = 0, the PEIE bit must be set to enable any of these peripheral interrupts.

#### REGISTER 9-6: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

W = Writable bit

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

U = Unimplemented bit, read as '0'

x = Bit is unknown

-n = Value at Po	OR '1' = Bit is set	'0' = Bit is cleared
bit 7	SPPIE: Streaming Parallel Port Read/Write  1 = Enables the SPP read/write interrupt	e Interrupt Enable bit <sup>(1)</sup>
bit 6	<ul> <li>0 = Disables the SPP read/write interrupt</li> <li>ADIE: A/D Converter Interrupt Enable bit</li> <li>1 = Enables the A/D interrupt</li> <li>0 = Disables the A/D interrupt</li> </ul>	
bit 5	RCIE: EUSART Receive Interrupt Enable 1 = Enables the EUSART receive interrupt 0 = Disables the EUSART receive interrupt 1 = Disables the EUSART receive interrupt 2 = Disables the EUSART receive interrupt 3 = Disables the EUSART receive int	t
bit 4	<b>TXIE:</b> EUSART Transmit Interrupt Enable 1 = Enables the EUSART transmit interrupt 0 = Disables the EUSART transmit interrupt 1 = Disables the EUSART transmit interrupt 2 = Disables the EUSART transmit 2 = Disables the EUSART transm	bit ot
bit 3	<b>SSPIE:</b> Master Synchronous Serial Port In 1 = Enables the MSSP interrupt 0 = Disables the MSSP interrupt	terrupt Enable bit
bit 2	CCP1IE: CCP1 Interrupt Enable bit  1 = Enables the CCP1 interrupt  0 = Disables the CCP1 interrupt	
bit 1	<b>TMR2IE:</b> TMR2 to PR2 Match Interrupt En 1 = Enables the TMR2 to PR2 match inter 0 = Disables the TMR2 to PR2 match inter	rupt
bit 0	<b>TMR1IE:</b> TMR1 Overflow Interrupt Enable 1 = Enables the TMR1 overflow interrupt 0 = Disables the TMR1 overflow interrupt	bit

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

#### REGISTER 9-7: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIE	CMIE	USBIE	EEIE	BCLIE	HLVDIE	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 CMIE: Comparator Interrupt Enable bit

1 = Enabled
0 = Disabled

bit 5 USBIE: USB Interrupt Enable bit

1 = Enabled
0 = Disabled

bit 4 **EEIE:** Data EEPROM/Flash Write Operation Interrupt Enable bit

1 = Enabled
0 = Disabled

bit 3 BCLIE: Bus Collision Interrupt Enable bit

1 = Enabled
0 = Disabled

bit 2 **HLVDIE:** High/Low-Voltage Detect Interrupt Enable bit

1 = Enabled0 = Disabled

bit 1 TMR3IE: TMR3 Overflow Interrupt Enable bit

1 = Enabled
0 = Disabled

bit 0 CCP2IE: CCP2 Interrupt Enable bit

1 =Enabled 0 =Disabled

#### 9.5 IPR Registers

The IPR registers contain the individual priority bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are two Peripheral Interrupt Priority registers (IPR1 and IPR2). Using the priority bits requires that the Interrupt Priority Enable (IPEN) bit be set.

#### REGISTER 9-8: 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
SPPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP
bit 7							bit 0

Legend:						
R = Readable bit -n = Value at POR		W = Writable bit	U = Unimplemented bit,	read as '0'		
		'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown		
bit 7 SPPIP: Streaming Parallel Port Read/W  1 = High priority  0 = Low priority		I/Write Interrupt Priority bit <sup>(1)</sup>				
bit 6 ADIP: A/D		Converter Interrupt Priorit	y bit			
	1 = High priority 0 = Low priority					

bit 5	RCIP: EUSART Receive Interrupt Priority bit
	1 = High priority
	0 = Low priority
bit 4	TXIP: EUSART Transmit Interrupt Priority bit
	. 12.1 2.9

1 = High priority 0 = Low priority

bit 3 SSPIP: Master Synchronous Serial Port Interrupt Priority bit

1 = High priority0 = Low priority

bit 2 CCP1IP: CCP1 Interrupt Priority bit

1 = High priority0 = Low priority

bit 1 TMR2IP: TMR2 to PR2 Match Interrupt Priority bit

1 = High priority0 = Low priority

bit 0 TMR1IP: TMR1 Overflow Interrupt Priority bit

1 = High priority0 = Low priority

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

#### REGISTER 9-9: IPR2: PERIPHERAL INTERRUPT PRIORITY 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
OSCFIP	CMIP	USBIP	EEIP	BCLIP	HLVDIP	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 priority0 = Low priority

bit 6 CMIP: Comparator Interrupt Priority bit

1 = High priority0 = Low priority

bit 5 USBIP: USB Interrupt Priority bit

1 = High priority0 = Low priority

bit 4 **EEIP:** Data EEPROM/Flash Write Operation Interrupt Priority bit

1 = High priority0 = Low priority

bit 3 **BCLIP:** Bus Collision Interrupt Priority bit

1 = High priority0 = Low priority

bit 2 **HLVDIP:** High/Low-Voltage Detect Interrupt Priority bit

1 = High priority0 = Low priority

bit 1 TMR3IP: TMR3 Overflow Interrupt Priority bit

1 = High priority0 = Low priority

bit 0 CCP2IP: CCP2 Interrupt Priority bit

1 = High priority0 = Low priority

#### 9.6 RCON Register

The RCON register contains flag bits which are used to determine the cause of the last Reset or wake-up from Idle or Sleep modes. RCON also contains the IPEN bit which enables interrupt priorities.

#### REGISTER 9-10: RCON: RESET CONTROL REGISTER

R/W-0	R/W-1 <sup>(1)</sup>	U-0	R/W-1	R-1	R-1	R/W-0 <sup>(2)</sup>	R/W-0
IPEN	SBOREN	_	RI	TO	PD	POR	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	SBOREN: BOR Software Enable bit <sup>(1)</sup>
	For details of bit operation, see Register 4-1.
bit 5	Unimplemented: Read as '0'
bit 4	RI: RESET Instruction Flag bit
	For details of bit operation, see Register 4-1.
bit 3	TO: Watchdog 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 <sup>(2)</sup>
	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.

- **Note 1:** If SBOREN is enabled, its Reset state is '1'; otherwise, it is '0'. See Register 4-1 for additional information.
  - 2: The actual Reset value of POR is determined by the type of device Reset. See Register 4-1 for additional information.

#### 9.7 INTn Pin Interrupts

External interrupts on the RB0/AN12/INT0/FLT0/SDI/SDA, RB1/AN10/INT1/SCK/SCL and RB2/AN8/INT2/VMO pins are edge-triggered. If the corresponding INTEDGx 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 RBx/INTx pin, the corresponding flag bit, INTxIF, is set. This interrupt can be disabled by clearing the corresponding enable bit, INTxIE. Flag bit, INTxIF, must be cleared in software in the Interrupt Service Routine before re-enabling the interrupt.

All external interrupts (INT0, INT1 and INT2) can wakeup the processor from the power-managed modes if bit, INTxIE, 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 and INT2 is determined by the value contained in the interrupt priority bits, INT1IP (INTCON3<6>) and INT2IP (INTCON3<7>). There is no priority bit associated with INT0. It is always a high priority interrupt source.

#### 9.8 TMR0 Interrupt

In 8-bit mode (which is the default), an overflow in the TMR0 register (FFh  $\rightarrow$  00h) will set flag bit, TMR0IF. In 16-bit mode, an overflow in the TMR0H:TMR0L register pair (FFFFh  $\rightarrow$  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 11.0** "Timer0 Module" for further details on the Timer0 module.

#### 9.9 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>).

#### 9.10 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
                                         ; Restore BSR
         BSR_TEMP, BSR
MOVF
         W TEMP, W
                                         : Restore WREG
         STATUS TEMP, STATUS
                                         ; Restore STATUS
```

NOTES:

#### 10.0 I/O PORTS

Depending on the device selected and features enabled, there are up to five 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.

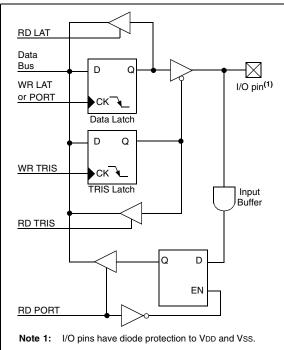
Each port has three registers for its operation. These registers are:

- TRIS register (data direction register)
- PORT register (reads the levels on the pins of the device)
- · LAT register (output latch)

The Data Latch register (LATA) is useful for read-modify-write operations on the value driven by the I/O pins.

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 PORTA, TRISA and LATA Registers

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

Reading the PORTA register reads the status of the pins; writing to it will write to the port latch.

The Data Latch register (LATA) is also memory mapped. Read-modify-write operations on the LATA register read and write the latched output value for PORTA.

The RA4 pin is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA6 pin is multiplexed with the main oscillator pin; it is enabled as an oscillator or I/O pin by the selection of the main oscillator in Configuration Register 1H (see Section 25.1 "Configuration Bits" for details). When not used as a port pin, RA6 and its associated TRIS and LAT bits are read as '0'.

RA4 is also multiplexed with the USB module; it serves as a receiver input from an external USB transceiver. For details on configuration of the USB module, see Section 17.2 "USB Status and Control".

Several PORTA pins are multiplexed with analog inputs, the analog VREF+ and VREF- inputs and the comparator voltage reference output. The operation of pins RA5 and RA3:RA0 as A/D converter inputs is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register 1).

Note: On a Power-on Reset, RA5 and RA3:RA0 are configured as analog inputs and read as '0'. RA4 is configured as a digital input.

All other PORTA pins have TTL input levels and full CMOS output drivers.

The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

#### **EXAMPLE 10-1: INITIALIZING PORTA**

LAMINIF	LL 10-1.		INITIALIZING FORTA
CLRF	PORTA	;	Initialize PORTA by
		;	clearing output
		;	data latches
CLRF	LATA	;	Alternate method
		;	to clear output
		;	data latches
MOVLW	0Fh	;	Configure A/D
MOVWF	ADCON1	;	for digital inputs
MOVLW	07h	;	Configure comparators
MOVWF	CMCON	;	for digital input
MOVLW	0CFh	;	Value used to
		;	initialize data
		;	direction
MOVWF	TRISA	;	Set RA<3:0> as inputs
		;	RA<5:4> as outputs
			_

TABLE 10-1: PORTA I/O SUMMARY

Pin	Function	TRIS Setting	I/O	I/O Type	Description		
RA0/AN0	RA0	0	OUT	DIG	LATA<0> data output; not affected by analog input.		
		1	IN	TTL	PORTA<0> data input; disabled when analog input enabled.		
	AN0	1	IN	ANA	A/D input channel 0 and Comparator C1- input. Default configuration on POR; does not affect digital output.		
RA1/AN1	RA1	0	OUT	DIG	LATA<1> data output; not affected by analog input.		
		1	IN	TTL	PORTA<1> data input; reads '0' on POR.		
	AN1	1	IN	ANA	A/D input channel 1 and Comparator C2- input. Default configuration on POR; does not affect digital output.		
RA2/AN2/ VREF-/CVREF	RA2	0	OUT	DIG	LATA<2> data output; not affected by analog input. Disabled when CVREF output enabled.		
		1	IN	TTL	PORTA<2> data input. Disabled when analog functions enabled; disabled when CVREF output enabled.		
	AN2	1	IN	ANA	A/D input channel 2 and Comparator C2+ input. Default configuration on POR; not affected by analog output.		
	VREF-	1	IN	ANA	A/D and comparator voltage reference low input.		
	CVREF	х	OUT	ANA	Comparator voltage reference output. Enabling this feature disables digital I/O.		
RA3/AN3/	RA3	0	OUT	DIG	LATA<3> data output; not affected by analog input.		
VREF+		1	IN	TTL	PORTA<3> data input; disabled when analog input enabled.		
	AN3	1	IN	ANA	A/D input channel 3 and Comparator C1+ input. Default configuration on POR.		
	VREF+	1	IN	ANA	A/D and comparator voltage reference high input.		
RA4/T0CKI/	RA4	0	OUT	DIG	LATA<4> data output; not affected by analog input.		
C1OUT/RCV		1	IN	ST	PORTA<4> data input; disabled when analog input enabled.		
	T0CKI	1	IN	ST	Timer0 clock input.		
	C1OUT	0	OUT	DIG	Comparator 1 output; takes priority over port data.		
	RCV	х	IN	TTL	External USB transceiver RCV input.		
RA5/AN4/SS/	RA5	0	OUT	DIG	LATA<5> data output; not affected by analog input.		
HLVDIN/C2OUT		1	IN	TTL	PORTA<5> data input; disabled when analog input enabled.		
	AN4	1	IN	ANA	A/D input channel 4. Default configuration on POR.		
	SS	1	IN	TTL	Slave select input for SSP (MSSP module).		
	HLVDIN	1	IN	ANA	High/Low-Voltage Detect external trip point input.		
	C2OUT	0	OUT	DIG	Comparator 2 output; takes priority over port data.		
OSC2/CLKO/	OSC2	х	OUT	ANA	Main oscillator feedback output connection (all XT and HS modes).		
RA6	CLKO	х	OUT	DIG	System cycle clock output (Fosc/4); available in EC, ECPLL and INTCKO modes.		
	RA6	0	OUT	DIG	LATA<6> data output. Available only in ECIO, ECPIO and INTIO modes; otherwise, reads as '0'.		
		1	IN	TTL	PORTA<6> data input. Available only in ECIO, ECPIO and INTIO modes; otherwise, reads as '0'.		

Legend: OUT = Output, IN = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input,

 $TTL = TTL \ Buffer \ Input, \ x = Don't \ care \ (TRIS \ bit \ does \ not \ affect \ port \ direction \ or \ is \ overridden \ for \ this \ option)$ 

TABLE 10-2: 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	_	RA6 <sup>(1)</sup>	RA5	RA4	RA3	RA2	RA1	RA0	54
LATA	_	LATA6 <sup>(1)</sup>	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	54
TRISA	1	TRISA6 <sup>(1)</sup>	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	54
ADCON1	_	_	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	52
CMCON	C2OUT	C10UT	C2INV	C1INV	CIS	CM2	CM1	CM0	53
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	53
UCON	_	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND		55

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by PORTA.

**Note 1:** RA6 and its associated latch and data direction bits are enabled as I/O pins based on oscillator configuration; otherwise, they are read as '0'.

© 2006 Microchip Technology Inc. Preliminary DS39632C-page 113

# 10.2 PORTB, TRISB and LATB Registers

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

The Data Latch register (LATB) is also memory mapped. Read-modify-write operations on the LATB register read and write the latched output value for PORTB.

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.

Note: On a Power-on Reset, RB4:RB0 are configured as analog inputs by default and read as '0'; RB7:RB5 are configured as digital inputs.

By programming the Configuration bit, PBADEN (CONFIG3H<1>), RB4:RB0 will alternatively be configured as digital inputs on POR.

Four of the PORTB pins (RB7:RB4) have an interrupton-change feature. Only pins configured as inputs can cause this interrupt to occur. Any RB7:RB4 pin configured as an output is excluded from the interrupton-change comparison. The pins 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>).

The interrupt-on-change can be used to wake the device from Sleep. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- Any read or write of PORTB (except with the MOVFF (ANY), PORTB instruction). This will end the mismatch condition.
- b) Clear flag bit, RBIF.

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

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

Pins, RB2 and RB3, are multiplexed with the USB peripheral and serve as the differential signal outputs for an external USB transceiver (TRIS configuration). Refer to **Section 17.2.2.2 "External Transceiver"** for additional information on configuring the USB module for operation with an external transceiver.

RB4 is multiplexed with CSSPP, the chip select function for the Streaming Parallel Port (SPP) – TRIS setting. Details of its operation are discussed in **Section 18.0** "Streaming Parallel Port".

#### **EXAMPLE 10-2: INITIALIZING PORTB**

CLRF	PORTB	; Initialize PORTB by
		; clearing output
		; data latches
CLRF	LATB	; Alternate method
		; to clear output
		; data latches
MOVLW	0Eh	; Set RB<4:0> as
MOVWF	ADCON1	; digital I/O pins
		; (required if config bit
		; PBADEN is set)
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

**TABLE 10-3: PORTB I/O SUMMARY** 

Pin	Function	TRIS Setting	I/O	I/O Type	Description				
RB0/AN12/	RB0	0	OUT	DIG	LATB<0> data output; not affected by analog input.				
INT0/FLT0/ SDI/SDA		1	IN	TTL	PORTB<0> data input; weak pull-up when RBPU bit is cleared. Disabled when analog input enabled.(1)				
	AN12	1	IN	ANA	A/D input channel 12. <sup>(1)</sup>				
	INT0	1	IN	ST	External interrupt 0 input.				
	FLT0	1	IN	ST	Enhanced PWM Fault input (ECCP1 module); enabled in software.				
	SDI	1	IN	ST	SPI data input (MSSP module).				
	SDA	1	OUT	DIG	I <sup>2</sup> C™ data output (MSSP module); takes priority over port data.				
		1	IN	I <sup>2</sup> C/SMB	1 <sup>2</sup> C data input (MSSP module); input type depends on module setting.				
RB1/AN10/	RB1	0	OUT	DIG	LATB<1> data output; not affected by analog input.				
INT1/SCK/ SCL		1	IN	TTL	PORTB<1> data input; weak pull-up when RBPU bit is cleared. Disabled when analog input enabled.(1)				
	AN10	1	IN	ANA	A/D input channel 10. <sup>(1)</sup>				
	INT1	1	IN	ST	External interrupt 1 input.				
	SCK	0	OUT	DIG	SPI clock output (MSSP module); takes priority over port data.				
		1	IN	ST	SPI clock input (MSSP module).				
	SCL	CL 0 OUT DIG I <sup>2</sup> C clock output (MSSP module); takes pri		I <sup>2</sup> C clock output (MSSP module); takes priority over port data.					
		1	IN	I <sup>2</sup> C/SMB	I <sup>2</sup> C clock input (MSSP module); input type depends on module setting.				
RB2/AN8/	RB2	RB2 0 OU		DIG	LATB<2> data output; not affected by analog input.				
NT2/VMO		1	IN	TTL	PORTB<2> data input; weak pull-up when RBPU bit is cleared. Disabled when analog input enabled.(1)				
	AN8	1	IN	ANA	A/D input channel 8. <sup>(1)</sup>				
	INT2	1	IN	ST	External interrupt 2 input.				
	VMO	0	OUT	DIG	External USB transceiver VMO data output.				
RB3/AN9/	RB3	0	OUT	DIG	LATB<3> data output; not affected by analog input.				
CCP2/VPO		1	IN	TTL	PORTB<3> data input; weak pull-up when RBPU bit is cleared. Disabled when analog input enabled.(1)				
	AN9	1	IN	ANA	A/D input channel 9. <sup>(1)</sup>				
	CCP2 <sup>(2)</sup>	0	OUT	DIG	CCP2 Compare and PWM output.				
		1	IN	ST	CCP2 Capture input.				
	VPO	0	OUT	DIG	External USB transceiver VPO data output.				
RB4/AN11/	RB4	0	OUT	DIG	LATB<4> data output; not affected by analog input.				
KBI0/CSSPP		1	IN	TTL	PORTB<4> data input; weak pull-up when RBPU bit is cleared. Disabled when analog input enabled.(1)				
	AN11	1	IN	ANA	A/D input channel 11. <sup>(1)</sup>				
	KBI0	1	IN	TTL	Interrupt-on-pin change.				
	CSSPP <sup>(4)</sup>	0	OUT	DIG	SPP chip select control output.				
RB5/KBI1/	RB5	0	OUT	DIG	LATB<5> data output.				
PGM		1	IN	TTL	PORTB<5> data input; weak pull-up when RBPU bit is cleared.				
	KBI1	1	IN	TTL	Interrupt-on-pin change.				
	PGM	х	IN	ST	Single-Supply Programming mode entry (ICSP™). Enabled by LVP Configuration bit; all other pin functions disabled.				

**Legend:** OUT = Output, IN = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input, I<sup>2</sup>C/SMB = I<sup>2</sup>C/SMBus input buffer, TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option)

Note 1: Configuration on POR is determined by PBADEN Configuration bit. Pins are configured as analog inputs when PBADEN is set and digital inputs when PBADEN is cleared.

- 2: Alternate pin assignment for CCP2 when CCP2MX = 0. Default assignment is RC1.
- **3:** All other pin functions are disabled when ICSP™ or ICD operation is enabled.
- 4: 40/44-pin devices only.

TABLE 10-3: PORTB I/O SUMMARY (CONTINUED)

Pin	Function	TRIS Setting	I/O	I/O Type	Description		
RB6/KBI2/	RB6	0	OUT	DIG	LATB<6> data output.		
PGC		1	IN	TTL	PORTB<6> data input; weak pull-up when RBPU bit is cleared.		
	KBI2	1	IN	TTL	Interrupt-on-pin change.		
	PGC	х	IN	ST	Serial execution (ICSP™) clock input for ICSP and ICD operation. (3)		
RB7/KBI3/	RB7	0	OUT	DIG	LATB<7> data output.		
PGD		1	IN	TTL	PORTB<7> data input; weak pull-up when RBPU bit is cleared.		
	KBI3	1	IN	TTL	Interrupt-on-pin change.		
	PGD	х	OUT	DIG	Serial execution data output for ICSP and ICD operation. (3)		
		х	IN	ST	Serial execution data input for ICSP and ICD operation. (3)		

**Legend:** OUT = Output, IN = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input,  $I^2C/SMB = I^2C/SMBus$  input buffer, TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is

overridden for this option)

**Note** 1: Configuration on POR is determined by PBADEN Configuration bit. Pins are configured as analog inputs when PBADEN is set and digital inputs when PBADEN is cleared.

- 2: Alternate pin assignment for CCP2 when CCP2MX = 0. Default assignment is RC1.
- **3:** All other pin functions are disabled when ICSP™ or ICD operation is enabled.
- 4: 40/44-pin devices only.

TABLE 10-4: 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	54
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	54
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	54
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INTOIE	RBIE	TMR0IF	INT0IF	RBIF	51
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	-	TMR0IP	_	RBIP	51
INTCON3	INT2IP	INT1IP	_	INT2IE	INT1IE	_	INT2IF	INT1IF	51
ADCON1	_	_	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	52
SPPCON <sup>(1)</sup>	_	_	_	_	_	_	SPPOWN	SPPEN	55
SPPCFG <sup>(1)</sup>	CLKCFG1	CLKCFG0	CSEN	CLK1EN	WS3	WS2	WS1	WS0	55
UCON	_	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	_	55

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by PORTB.

Note 1: These registers are unimplemented on 28-pin devices.

# 10.3 PORTC, TRISC and LATC Registers

PORTC is a 7-bit wide, bidirectional port. The corresponding data direction register is TRISC. Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., put the contents of the output latch on the selected pin).

In PIC18F2455/2550/4455/4550 devices, the RC3 pin is not implemented.

The Data Latch register (LATC) is also memory mapped. Read-modify-write operations on the LATC register read and write the latched output value for PORTC.

PORTC is primarily multiplexed with serial communication modules, including the EUSART, MSSP module and the USB module (Table 10-5). Except for RC4 and RC5, PORTC uses Schmitt Trigger input buffers.

Pins RC4 and RC5 are multiplexed with the USB module. Depending on the configuration of the module, they can serve as the differential data lines for the onchip USB transceiver, or the data inputs from an external USB transceiver. Both RC4 and RC5 have TTL input buffers instead of the Schmitt Trigger buffers on the other pins.

Unlike other PORTC pins, RC4 and RC5 do not have TRISC bits associated with them. As digital ports, they can only function as digital inputs. When configured for USB operation, the data direction is determined by the configuration and status of the USB module at a given time. If an external transceiver is used, RC4 and RC5 always function as inputs from the transceiver. If the on-chip transceiver is used, the data direction is determined by the operation being performed by the module at that time.

When the external transceiver is enabled, RC2 also serves as the output enable control to the transceiver. Additional information on configuring USB options is provided in **Section 17.2.2.2** "External Transceiver".

When enabling peripheral functions on PORTC pins other than RC4 and RC5, care should be taken in defining the TRIS bits. 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: On a Power-on Reset, these pins, except RC4 and RC5, are configured as digital inputs. To use pins RC4 and RC5 as digital inputs, the USB module must be disabled (UCON<3> = 0) and the on-chip USB transceiver must be disabled (UCFG<3> = 1).

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	07h	; Value used to
		; initialize data
		; direction
MOVWF	TRISC	; RC<5:0> as outputs
		; RC<7:6> as inputs

TABLE 10-5: PORTC I/O SUMMARY

Pin	Function	TRIS Setting	I/O	I/O Type	Description		
RC0/T1OSO/	RC0	0	OUT	DIG	LATC<0> data output.		
T13CKI		1	IN	ST	PORTC<0> data input.		
	T10S0	х	OUT	ANA	Timer1 oscillator output; enabled when Timer1 oscillator enabled. Disables digital I/O.		
	T13CKI	1	IN	ST	Timer1/Timer3 counter input.		
RC1/T1OSI/	RC1	0	OUT	DIG	LATC<1> data output.		
CCP2/UOE		1	IN	ST	PORTC<1> data input.		
	T1OSI	х	IN	ANA	NA Timer1 oscillator input; enabled when Timer1 oscillator enabled. Disables digital I/O.		
	CCP2 <sup>(1)</sup>	0	OUT	DIG	CCP2 Compare and PWM output; takes priority over port data.		
		1	IN	ST	CCP2 Capture input.		
	UOE	0	OUT	DIG	External USB transceiver OE output.		
RC2/CCP1/	RC2	0	OUT	DIG	LATC<2> data output.		
P1A 1 IN ST		ST	PORTC<2> data input.				
	CCP1	0	OUT	DIG	ECCP1 Compare and PWM output; takes priority over port data.		
-		1	IN	ST	ECCP1 Capture input.		
			OUT	DIG	ECCP1 Enhanced PWM output, channel A; takes priority over port data. May be configured for tri-state during Enhanced PWM shutdown events.		
RC4/D-/VM	RC4	(2)	IN	TTL	PORTC<4> data input; disabled when USB module or on-chip transceiver are enabled.		
	D-	(2)	OUT	XCVR	USB bus differential minus line output (internal transceiver).		
		(2)	IN	XCVR	USB bus differential minus line input (internal transceiver).		
	VM	(2)	IN	TTL	External USB transceiver VM input.		
RC5/D+/VP	RC5	(2)	IN	TTL	PORTC<5> data input; disabled when USB module or on-chip transceiver are enabled.		
	D+	(2)	OUT	XCVR	USB bus differential plus line output (internal transceiver).		
		(2)	IN	XCVR	USB bus differential plus line input (internal transceiver).		
	VP	(2)	IN	TTL	External USB transceiver VP input.		
RC6/TX/CK	RC6	0	OUT	DIG	LATC<6> data output.		
		1	IN	ST	PORTC<6> data input.		
	TX	0	OUT	DIG	Asynchronous serial transmit data output (EUSART module); takes priority over port data. User must configure as output.		
	CK	0	OUT	DIG	Synchronous serial clock output (EUSART module); takes priority over port data.		
		1	IN	ST	Synchronous serial clock input (EUSART module).		

Legend: OUT = Output, IN = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input,

TTL = TTL Buffer Input, XCVR = USB transceiver, x = Don't care (TRIS bit does not affect port direction or is overridden for this option)

Note 1: Default pin assignment. Alternate pin assignment is RB3 (when CCP2MX = 0).

2: RC4 and RC5 do not have corresponding TRISC bits. In Port mode, these pins are input only. USB data direction is determined by the USB configuration.

3: 40/44-pin devices only.

TABLE 10-5: PORTC I/O SUMMARY (CONTINUED)

Pin	Function	TRIS Setting	I/O	I/O Type	Description	
RC7/RX/DT/	RC7	0	OUT	DIG	LATC<7> data output.	
SDO		1	IN	ST PORTC<7> data input.		
	RX	1	IN	ST	Asynchronous serial receive data input (EUSART module).	
	DT	1	OUT	DIG	Synchronous serial data output (EUSART module); takes priority over SPI and port data.	
		1	IN	ST	Synchronous serial data input (EUSART module). User must configure as an input.	
	SDO	0	OUT	DIG	SPI data output (MSSP module); takes priority over port data.	

Legend: OUT = Output, IN = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input,

TTL = TTL Buffer Input, XCVR = USB transceiver, x = Don't care (TRIS bit does not affect port direction or is overridden for this option)

- Note 1: Default pin assignment. Alternate pin assignment is RB3 (when CCP2MX = 0).
  - 2: RC4 and RC5 do not have corresponding TRISC bits. In Port mode, these pins are input only. USB data direction is determined by the USB configuration.
  - 3: 40/44-pin devices only.

TABLE 10-6: 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 <sup>(1)</sup>	RC4 <sup>(1)</sup>	_	RC2	RC1	RC0	54
LATC	LATC7	LATC6	_	_	_	LATC2	LATC1	LATC0	54
TRISC	TRISC7	TRISC6	_	_	_	TRISC2	TRISC1	TRISC0	54
UCON	_	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	_	55

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by PORTC.

Note 1: RC5 and RC4 are only available as port pins when the USB module is disabled (UCON<3> = 0).

© 2006 Microchip Technology Inc. Preliminary DS39632C-page 119

# 10.4 PORTD, TRISD and LATD Registers

**Note:** PORTD is only available on 40/44-pin devices.

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

The Data Latch register (LATD) is also memory mapped. Read-modify-write operations on the LATD register read and write the latched output value for PORTD.

All pins on PORTD are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

Each of the PORTD pins has a weak internal pull-up. A single control bit, RDPU (PORTE<7>), can turn on all the pull-ups. This is performed by setting RDPU. The weak pull-up is automatically turned off when the port pin is configured as a digital output or as one of the other multiplexed peripherals. The pull-ups are disabled on a Power-on Reset. The PORTE register is shown in **Section 10.5 "PORTE**, **TRISE and LATE Registers"**.

Three of the PORTD pins are multiplexed with outputs, P1B, P1C and P1D, of the Enhanced CCP module. The operation of these additional PWM output pins is covered in greater detail in **Section 16.0 "Enhanced Capture/Compare/PWM (ECCP) Module"**.

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

PORTD can also be configured as an 8-bit wide Streaming Parallel Port (SPP). In this mode, the input buffers are TTL. For additional information on configuration and uses of the SPP, see **Section 18.0** "Streaming Parallel Port".

Note: When the Enhanced PWM mode is used with either dual or quad outputs, the MSSP functions of PORTD are automatically disabled.

#### **EXAMPLE 10-4: INITIALIZING PORTD**

CLRF	PORTD	; Initialize PORTD by ; clearing output
CLRF	LATD	<pre>; data latches ; Alternate method</pre>
CLRF	LAID	; 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

TABLE 10-7: PORTD I/O SUMMARY

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RD0/SPP0	RD0	0	OUT	DIG	LATD<0> data output.
		1	IN	ST	PORTD<0> data input.
İ	SPP0	1	OUT	DIG	SPP<0> output data; takes priority over port data.
		1	IN	TTL	SPP<0> input data.
RD1/SPP1	RD1	0	OUT	DIG	LATD<1> data output.
		1	IN	ST	PORTD<1> data input.
	SPP1	1	OUT	DIG	SPP<1> output data; takes priority over port data.
		1	IN	TTL	SPP<1> input data.
RD2/SPP2	RD2	0	OUT	DIG	LATD<2> data output.
		1	IN	ST	PORTD<2> data input.
1	SPP2	1	OUT	DIG	SPP<2> output data; takes priority over port data.
		1	IN	TTL	SPP<2> input data.
RD3/SPP3	RD3	0	OUT	DIG	LATD<3> data output.
		1	IN	ST	PORTD<3> data input.
1	SPP3	1	OUT	DIG	SPP<3> output data; takes priority over port data.
		1	IN	TTL	SPP<3> input data.
RD4/SPP4	RD4	0	OUT	DIG	LATD<4> data output.
		1	IN	ST	PORTD<4> data input.
	SPP4	1	OUT	DIG	SPP<4> output data; takes priority over port data.
		1	IN	TTL	SPP<4> input data.
RD5/SPP5/P1B	RD5	0	OUT	DIG	LATD<5> data output
		1	IN	ST	PORTD<5> data input
İ	SPP5	1	OUT	DIG	SPP<5> output data; takes priority over port data.
		1	IN	TTL	SPP<5> input data.
·	P1B	0	OUT	DIG	ECCP1 Enhanced PWM output, channel B; takes priority over port and SPP data. (1)
RD6/SPP6/P1C	RD6	0	OUT	DIG	LATD<6> data output.
		1	IN	ST	PORTD<6> data input.
	SPP6	1	OUT	DIG	SPP<6> output data; takes priority over port data.
		1	IN	TTL	SPP<6> input data.
·	P1C	0	OUT DIG ECCP1 Enhanced PV		ECCP1 Enhanced PWM output, channel C; takes priority over port and SPP data. (1)
RD7/SPP7/P1D	RD7	0	OUT	DIG	LATD<7> data output.
		1	IN	ST	PORTD<7> data input.
	SPP7	1	OUT	DIG	SPP<7> output data; takes priority over port data.
		1	IN	TTL	SPP<7> input data.
	P1D	0	OUT	DIG	ECCP1 Enhanced PWM output, channel D; takes priority over port and SPP data. <sup>(1)</sup>

Legend: OUT = Output, IN = Input, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input

Note 1: May be configured for tri-state during Enhanced PWM shutdown events.

TABLE 10-8: 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 <sup>(3)</sup>	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	54
LATD <sup>(3)</sup>	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	54
TRISD <sup>(3)</sup>	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	54
PORTE	RDPU <sup>(3)</sup>	_	_	_	RE3 <sup>(1,2)</sup>	RE2 <sup>(3)</sup>	RE1 <sup>(3)</sup>	RE0 <sup>(3)</sup>	54
CCP1CON	P1M1 <sup>(3)</sup>	P1M0 <sup>(3)</sup>	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	53
SPPCON <sup>(3)</sup>	_	_	_	_	_	_	SPPOWN	SPPEN	55

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by PORTD.

**Note 1:** Implemented only when Master Clear functionality is disabled (MCLRE Configuration bit = 0).

2: RE3 is the only PORTE bit implemented on both 28-pin and 40/44-pin devices. All other bits are implemented only when PORTE is implemented (i.e., 40/44-pin devices).

3: These registers and/or bits are unimplemented on 28-pin devices.

# 10.5 PORTE, TRISE and LATE Registers

Depending on the particular PIC18F2455/2550/4455/4550 device selected, PORTE is implemented in two different ways.

For 40/44-pin devices, PORTE is a 4-bit wide port. Three pins (RE0/AN5/CK1SPP, RE1/AN6/CK2SPP and RE2/AN7/OESPP) are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers. When selected as an analog input, these pins will read as '0's.

The corresponding data direction register is TRISE. Setting a TRISE bit (=1) will make the corresponding PORTE pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISE bit (=0) will make the corresponding PORTE pin an output (i.e., put the contents of the output latch on the selected pin).

In addition to port data, the PORTE register (Register 10-1) also contains the RDPU control bit (PORTE<7>); this enables or disables the weak pull-ups on PORTD.

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

**Note:** On a Power-on Reset, RE2:RE0 are configured as analog inputs.

The Data Latch register (LATE) is also memory mapped. Read-modify-write operations on the LATE register read and write the latched output value for PORTE.

The fourth pin of PORTE (MCLR/VPP/RE3) is an input only pin. Its operation is controlled by the MCLRE Configuration bit. When selected as a port pin (MCLRE = 0), it functions as a digital input only pin; as such, it does not have TRIS or LAT bits associated with its operation. Otherwise, it functions as the device's Master Clear input. In either configuration, RE3 also functions as the programming voltage input during programming.

**Note:** On a Power-on Reset, RE3 is enabled as a digital input only if Master Clear functionality is disabled.

#### **EXAMPLE 10-5: INITIALIZING PORTE**

CLRF	PORTE	;	Initialize PORTE by
		;	clearing output
		;	data latches
CLRF	LATE	;	Alternate method
		;	to clear output
		;	data latches
MOVLW	0Ah	;	Configure A/D
MOVWF	ADCON1	;	for digital inputs
MOVLW	03h	;	Value used to
		;	initialize data
		;	direction
MOVLW	07h	;	Turn off
MOVWF	CMCON	;	comparators
MOVWF	TRISC	;	Set RE<0> as inputs
		;	RE<1> as outputs
		;	RE<2> as inputs

#### 10.5.1 PORTE IN 28-PIN DEVICES

For 28-pin devices, PORTE is only available when Master Clear functionality is disabled (MCLRE = 0). In these cases, PORTE is a single bit, input only port comprised of RE3 only. The pin operates as previously described.

#### REGISTER 10-1: PORTE REGISTER

R/W-0	U-0	U-0	U-0	R/W-x	R/W-0	R/W-0	R/W-0
RDPU <sup>(3)</sup>	_	_	_	RE3 <sup>(1,2)</sup>	RE2 <sup>(3)</sup>	RE1 <sup>(3)</sup>	RE0 <sup>(3)</sup>
bit 7							bit 0

Legend:R = Readable bitW = Writable bitU = Unimplemented bit, read as '0'-n = Value at POR'1' = Bit is set'0' = Bit is clearedx = Bit is unknown

bit 7 RDPU: PORTD Pull-up Enable bit

1 = PORTD pull-ups are enabled by individual port latch values

0 = All PORTD pull-ups are disabled

bit 6-4 Unimplemented: Read as '0'

bit 3-0 **RE3:RE0:** PORTE Data Input bits<sup>(1,2,3)</sup>

- Note 1: implemented only when Master Clear functionality is disabled (MCLRE Configuration bit = 0); otherwise, read as '0'.
  - **2:** RE3 is the only PORTE bit implemented on both 28-pin and 40/44-pin devices. All other bits are implemented only when PORTE is implemented (i.e., 40/44-pin devices).
  - 3: Unimplemented in 28-pin devices; read as '0'.

TABLE 10-9: PORTE I/O SUMMARY

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RE0/AN5/	RE0	0	OUT	DIG	LATE<0> data output; not affected by analog input.
CK1SPP		1	IN	ST	PORTE<0> data input; disabled when analog input enabled.
	AN5	1	IN	ANA	A/D input channel 5; default configuration on POR.
	CK1SPP	0	OUT	DIG	SPP clock 1 output (SPP enabled).
RE1/AN6/	RE1	0	OUT	DIG	LATE<1> data output; not affected by analog input.
CK2SPP		1	IN	ST	PORTE<1> data input; disabled when analog input enabled.
	AN6	1	IN	ANA	A/D input channel 6; default configuration on POR.
	CK2SPP	0	OUT	DIG	SPP clock 2 output (SPP enabled).
RE2/AN7/	RE2	0	OUT	DIG	LATE<2> data output; not affected by analog input.
OESPP		1	IN	ST	PORTE<2> data input; disabled when analog input enabled.
	AN7	1	IN	ANA	A/D input channel 7; default configuration on POR.
	OESPP	0	OUT	DIG	SPP enable output (SPP enabled).
		ST	External Master Clear input; enabled when MCLRE Configuration bit is set.		
	VPP	(1)	IN	ANA	High-voltage detection, used for ICSP™ mode entry detection. Always available regardless of pin mode.
	RE3	(1)	IN	ST	PORTE<3> data input; enabled when MCLRE Configuration bit is clear.

Legend: OUT = Output, IN = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input
 Note 1: RE3 does not have a corresponding TRISE<3> bit. This pin is always an input regardless of mode.

TABLE 10-10: 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	RDPU <sup>(3)</sup>	_	_	_	RE3 <sup>(1,2)</sup>	RE2 <sup>(3)</sup>	RE1 <sup>(3)</sup>	RE0 <sup>(3)</sup>	54
LATE <sup>(3)</sup>	_	_	_	_	_	LATE2	LATE1	LATE0	54
TRISE(3)	_	_	_	_	_	TRISE2	TRISE1	TRISE0	54
ADCON1	_	_	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	52
CMCON	C2OUT	C10UT	C2INV	C1INV	CIS	CM2	CM1	CM0	53
SPPCON <sup>(3)</sup>		_	_			-	SPPOWN	SPPEN	55
SPPCFG <sup>(3)</sup>	CLKCFG1	CLKCFG0	CSEN	CLK1EN	WS3	WS2	WS1	WS0	55

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by PORTE.

Note 1: Implemented only when Master Clear functionality is disabled (MCLRE Configuration bit = 0).

2: RE3 is the only PORTE bit implemented on both 28-pin and 40/44-pin devices. All other bits are implemented only when PORTE is implemented (i.e., 40/44-pin devices).

3: These registers or bits are unimplemented on 28-pin devices.

#### 11.0 TIMERO 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 11-1) 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 11-1. Figure 11-2 shows a simplified block diagram of the Timer0 module in 16-bit mode.

#### REGISTER 11-1: TOCON: TIMERO 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	l 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 TOCKI 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
	011 = 1.16 Prescale value 010 = 1:8 Prescale value
	001 = 1:4 Prescale value
	000 = 1:2 Prescale value

#### 11.1 Timer0 Operation

Timer0 can operate as either a timer or a counter; the mode is selected by clearing the ToCS bit (ToCON<5>). In Timer mode, the module increments on every clock by default unless a different prescaler value is selected (see **Section 11.3 "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 Counter 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

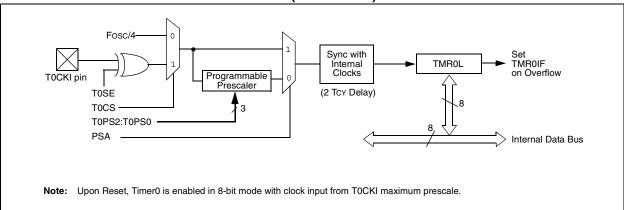
internal phase clock (Tosc). There is a delay between synchronization and the onset of incrementing the timer/counter.

## 11.2 Timer0 Reads and Writes in 16-Bit Mode

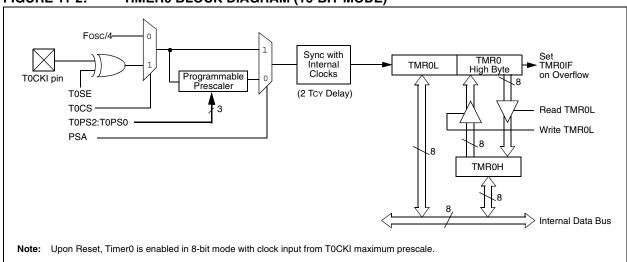
TMR0H is not the actual high byte of Timer0 in 16-bit mode. It is actually a buffered version of the real high byte of Timer0 which is not directly readable nor writable (refer to Figure 11-2). TMR0H is updated with the contents of the high byte of Timer0 during a read of TMR0L. This provides the ability 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 11-1: TIMERO BLOCK DIAGRAM (8-BIT MODE)



#### FIGURE 11-2: TIMERO BLOCK DIAGRAM (16-BIT MODE)



#### 11.3 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.

## 11.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control and can be changed "on-the-fly" during program execution.

#### 11.4 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 reenabling 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 11-1: REGISTERS ASSOCIATED WITH TIMERO

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page	
TMR0L	TMR0L Timer0 Register Low Byte									
TMR0H	Timer0 Register High Byte									
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	51	
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	_	TMR0IP	_	RBIP	51	
T0CON	TMR00N	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	52	
TRISA	_	TRISA6 <sup>(1)</sup>	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	54	

**Legend:** — = unimplemented locations, read as '0'. Shaded cells are not used by Timer0.

**Note 1:** RA6 is configured as a port pin based on various primary oscillator modes. When the port pin is disabled, all of the associated bits read '0'.

NOTES:

#### 12.0 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
- Module Reset on CCP Special Event Trigger
- · Device clock status flag (T1RUN)

A simplified block diagram of the Timer1 module is shown in Figure 12-1. A block diagram of the module's operation in Read/Write mode is shown in Figure 12-2.

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 12-1). 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 12-1: T1CON: TIMER1 CONTROL REGISTER

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	TMR10N
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 oscillator0 = 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 T10SCEN: 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

This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 TMR1CS: Timer1 Clock Source Select bit

1 = External clock from RC0/T10S0/T13CKI pin (on the rising edge)

0 = Internal clock (Fosc/4)

bit 0 TMR1ON: Timer1 On bit

1 = Enables Timer1

0 = Stops Timer1

#### 12.1 Timer1 Operation

Timer1 can operate in one of these modes:

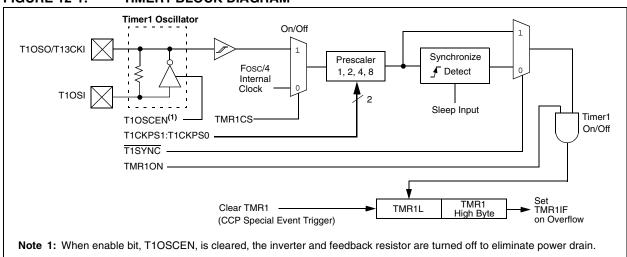
- Timer
- · Synchronous Counter
- · Asynchronous Counter

The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>). When TMR1CS is cleared (= 0), Timer1 increments on every internal instruction

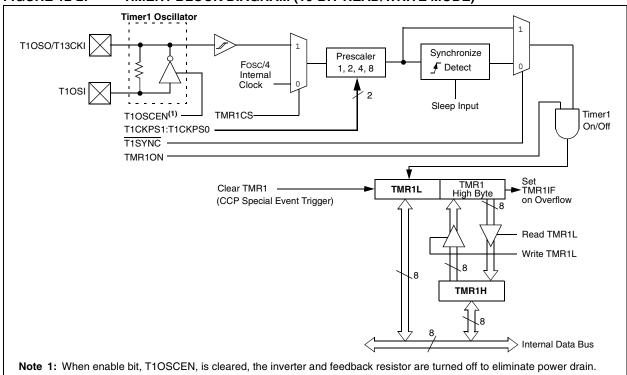
cycle (Fosc/4). When the bit is set, Timer1 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled.

When Timer1 is enabled, the RC1/T1OSI/UOE 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 12-1: TIMER1 BLOCK DIAGRAM



#### FIGURE 12-2: TIMER1 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)



#### 12.2 Timer1 16-Bit Read/Write Mode

Timer1 can be configured for 16-bit reads and writes (see Figure 12-2). 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. 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.

#### 12.3 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 12-3. Table 12-1 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 12-3: EXTERNAL
COMPONENTS FOR THE
TIMER1 LP OSCILLATOR

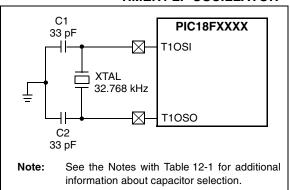


TABLE 12-1: CAPACITOR SELECTION FOR THETIMEROSCILLATOR<sup>(2,3,4)</sup>

Osc Type	Freq	C1	C2
LP	32 kHz	27 pF <sup>(1)</sup>	27 pF <sup>(1)</sup>

- 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.

## 12.3.1 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.

#### 12.3.2 LOW-POWER TIMER1 OPTION

The Timer1 oscillator can operate at two distinct levels of power consumption based on device configuration. When the LPT1OSC Configuration bit is set, the Timer1 oscillator operates in a low-power mode. When LPT1OSC is not set, Timer1 operates at a higher power level. Power consumption for a particular mode is relatively constant, regardless of the device's operating mode. The default Timer1 configuration is the higher power mode.

As the low-power Timer1 mode tends to be more sensitive to interference, high noise environments may cause some oscillator instability. The low-power option is, therefore, best suited for low noise applications where power conservation is an important design consideration.

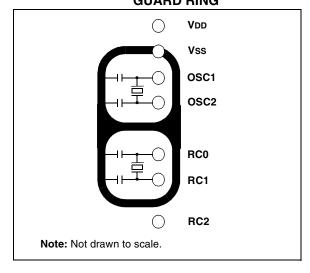
## 12.3.3 TIMER1 OSCILLATOR LAYOUT CONSIDERATIONS

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 12-3, 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.

If a high-speed circuit must be located near the oscillator (such as the CCP1 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 12-4, may be helpful when used on a single-sided PCB or in addition to a ground plane.

FIGURE 12-4: OSCILLATOR CIRCUIT
WITH GROUNDED
GUARD RING



#### 12.4 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>).

# 12.5 Resetting Timer1 Using the CCP Special Event Trigger

If either of the CCP modules is configured in Compare mode to generate a Special Event Trigger (CCP1M3:CCP1M0 or CCP2M3:CCP2M0 = 1011), this signal will reset Timer1. The trigger from CCP2 will also start an A/D conversion if the A/D module is enabled (see Section 15.3.4 "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 CCPRH:CCPRL 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 CCP2 module will not set the TMR1IF interrupt flag bit (PIR1<0>).

# 12.6 Using Timer1 as a Real-Time Clock

Adding an external LP oscillator to Timer1 (such as the one described in **Section 12.3 "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 12-1, 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.

#### EXAMPLE 12-1: IMPLEMENTING A REAL-TIME CLOCK USING A TIMER1 INTERRUPT SERVICE

```
RTCinit
          MOVLW
                                 ; Preload TMR1 register pair
          MOVWF TMR1H
                                ; for 1 second overflow
          CLRF
                TMR1L
          MOVLW b'00001111'
                                ; Configure for external clock,
                                ; Asynchronous operation, external oscillator
          MOVWF T10SC
                                ; Initialize timekeeping registers
          CLRF
                 secs
          CLRF
                 mins
         MOVLW
                 d'12'
         MOVWF hours
          BSF
                PIE1, TMR1IE
                                ; Enable Timer1 interrupt
         RETURN
RTCisr
                TMR1H, 7
         BSF
                               ; Preload for 1 sec overflow
                PIR1, TMR1IF
                                ; Clear interrupt flag
          BCF
                 secs, F
                                ; Increment seconds
          INCF
                                ; 60 seconds elapsed?
          MOVLW
                 d'59'
          CPFSGT secs
          RETURN
                                ; No, done
                                ; Clear seconds
          CLRF
                 secs
                                ; Increment minutes
          INCF mins, F
          MOVLW d'59'
                                ; 60 minutes elapsed?
          CPFSGT mins
         RETURN
                                ; No, done
                                ; clear minutes
          CLRF mins
                                ; Increment hours
          INCF
                 hours, F
         MOVLW
                 d'23'
                                ; 24 hours elapsed?
          CPFSGT hours
          RETURN
                                 ; No, done
          MOVLW
                 d'01'
                                 ; Reset hours to 1
          MOVWF hours
          RETURN
                                 ; Done
```

#### TABLE 12-2: 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	51
PIR1	SPPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54
PIE1	SPPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54
IPR1	SPPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54
TMR1L	Timer1 Reg	gister Low By	/te						52
TMR1H	TImer1 Register High Byte								
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T10SCEN	T1SYNC	TMR1CS	TMR10N	52

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by the Timer1 module.

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

NOTES:

#### 13.0 TIMER2 MODULE

The Timer2 module timer 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 postscaler (1:1 through 1:16)
- · Interrupt on TMR2 to PR2 match
- Optional use as the shift clock for the MSSP module

The module is controlled through the T2CON register (Register 13-1) which enables or disables the timer and configures the prescaler and postscaler. 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 13-1.

#### 13.1 Timer2 Operation

In normal operation, TMR2 is incremented from 00h on each clock (Fosc/4). A 2-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/postscaler (see Section 13.2 "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 postscaler 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 13-1: 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 T20UTPS3:T20UTPS0: 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

#### 13.2 Timer2 Interrupt

Timer2 also can 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>).

#### 13.3 TMR2 Output

The unscaled output of TMR2 is available primarily to the CCP 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 module operating in SPI mode. Additional information is provided in Section 19.0 "Master Synchronous Serial Port (MSSP) Module".

FIGURE 13-1: TIMER2 BLOCK DIAGRAM

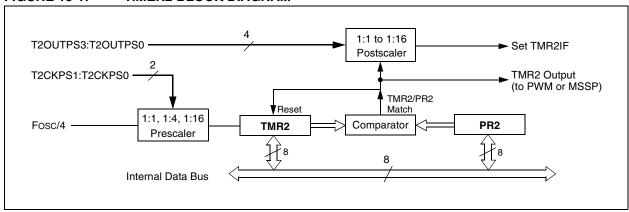


TABLE 13-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

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	51
PIR1	SPPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54
PIE1	SPPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54
IPR1	SPPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54
TMR2	Timer2 Reg	jister							52
T2CON	_	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	52
PR2	Timer2 Peri	iod Register		•		•	•		52

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by the Timer2 module.

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

#### 14.0 TIMER3 MODULE

The Timer3 module timer/counter 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 CCP Special Event Trigger

A simplified block diagram of the Timer3 module is shown in Figure 14-1. A block diagram of the module's operation in Read/Write mode is shown in Figure 14-2.

The Timer3 module is controlled through the T3CON register (Register 14-1). It also selects the clock source options for the CCP modules (see **Section 15.1.1** "CCP Modules and Timer Resources" for more information).

#### REGISTER 14-1: 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 CCPx Enable bits

1x = Timer3 is the capture/compare clock source for both CCP modules

01 = Timer3 is the capture/compare clock source for CCP2; Timer1 is the capture/compare clock source for CCP1

00 = Timer1 is the capture/compare clock source for both CCP 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

#### 14.1 Timer3 Operation

Timer3 can operate in one of three modes:

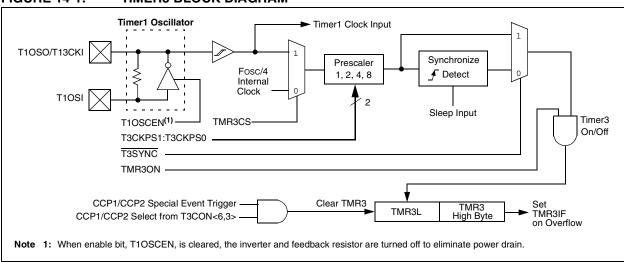
- Timer
- · Synchronous Counter
- · Asynchronous Counter

The operating mode is determined by the clock select bit, TMR3CS (T3CON<1>). When TMR3CS is cleared (= 0), Timer3 increments on every internal instruction

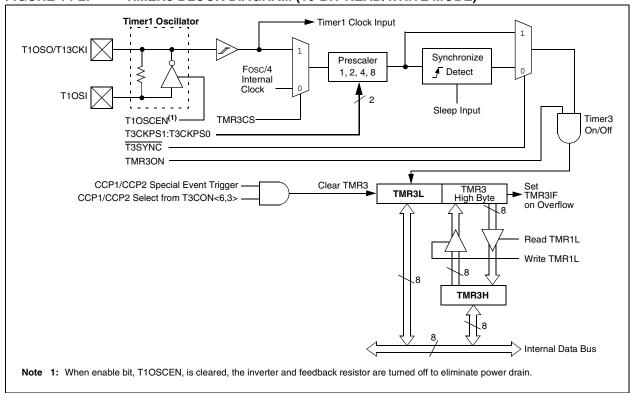
cycle (Fosc/4). When the bit is set, Timer3 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled.

As with Timer1, the RC1/T1OSI/\overline{IOE} 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 14-1: TIMER3 BLOCK DIAGRAM



#### FIGURE 14-2: TIMER3 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)



#### 14.2 Timer3 16-Bit Read/Write Mode

Timer3 can be configured for 16-bit reads and writes (see Figure 14-2). 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. 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.

## 14.3 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 12.0 "Timer1 Module".

#### 14.4 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>).

#### 14.5 Resetting Timer3 Using the CCP Special Event Trigger

If the CCP2 module is configured to generate a Special Event Trigger in Compare mode (CCP2M3:CCP2M0 = 1011), this signal will reset Timer3. It will also start an A/D conversion if the A/D module is enabled (see **Section 15.3.4 "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 CCPR2H:CCPR2L register pair effectively becomes a period register 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 a CCP module, the write will take precedence.

**Note:** The Special Event Triggers from the CCP2 module will not set the TMR3IF interrupt flag bit (PIR2<1>).

TABLE 14-1: 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	51
PIR2	OSCFIF	CMIF	USBIF	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	54
PIE2	OSCFIE	CMIE	USBIE	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	54
IPR2	OSCFIP	CMIP	USBIP	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	54
TMR3L	Timer3 Reg	gister Low Byt	е						53
TMR3H	Timer3 Reg	gister High By	te						53
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T10SCEN	T1SYNC	TMR1CS	TMR10N	52
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON	53

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by the Timer3 module.

NOTES:

# 15.0 CAPTURE/COMPARE/PWM (CCP) MODULES

PIC18F2455/2550/4455/4550 devices all have two CCP (Capture/Compare/PWM) modules. Each 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.

In 28-pin devices, the two standard CCP modules (CCP1 and CCP2) operate as described in this chapter. In 40/44-pin devices, CCP1 is implemented as an Enhanced CCP module, with standard Capture and Compare modes and Enhanced PWM modes. The ECCP implementation is discussed in **Section 16.0** "Enhanced Capture/Compare/PWM (ECCP) Module".

The Capture and Compare operations described in this chapter apply to all standard and Enhanced CCP modules.

Note: Throughout this section and Section 16.0 "Enhanced Capture/Compare/PWM (ECCP) Module", references to the register and bit names for CCP modules 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 CCP1, CCP2 or ECCP1. "CCPxCON" is used throughout these sections to refer to the module control register regardless of whether the CCP module is a standard or Enhanced implementation.

#### REGISTER 15-1: CCPxCON: STANDARD CCPx CONTROL REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
(1)	(1)	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0

I ea	end	•
LCG	CITA	

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'(1)

bit 5-4 DCxB1:DCxB0: PWM Duty Cycle Bit 1 and Bit 0 for CCPx Module

Capture mode:

Unused.

Compare mode:

Unused.

PWM mode:

These bits are the two LSbs (bit 1 and bit 0) of the 10-bit PWM duty cycle. The eight MSbs of the duty cycle are found in CCPR1L.

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 CCP2 match (CCPxIF bit is set)

11xx = PWM mode

Note 1: These bits are not implemented on 28-pin devices and are read as '0'.

#### 15.1 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.

## 15.1.1 CCP MODULES AND TIMER RESOURCES

The CCP modules utilize Timers 1, 2 or 3, depending on the mode selected. Timer1 and Timer3 are available to modules in Capture or Compare modes, while Timer2 is available for modules in PWM mode.

TABLE 15-1: CCP MODE – TIMER RESOURCE

CCP/ECCP Mode	Timer Resource
Capture	Timer1 or Timer3
Compare	Timer1 or Timer3
PWM	Timer2

The assignment of a particular timer to a module is determined by the Timer to CCP enable bits in the T3CON register (Register 14-1). Both modules may be active at any given time and may share the same timer resource if they are configured to operate in the same mode (Capture/Compare or PWM) at the same time. The interactions between the two modules are summarized in Figure 15-2. In Timer1 in Asynchronous Counter mode, the capture operation will not work.

#### 15.1.2 CCP2 PIN ASSIGNMENT

The pin assignment for CCP2 (Capture input, Compare and PWM output) can change, based on device configuration. The CCP2MX Configuration bit determines which pin CCP2 is multiplexed to. By default, it is assigned to RC1 (CCP2MX = 1). If the Configuration bit is cleared, CCP2 is multiplexed with RB3.

Changing the pin assignment of CCP2 does not automatically change any requirements for configuring the port pin. Users must always verify that the appropriate TRIS register is configured correctly for CCP2 operation, regardless of where it is located.

TABLE 15-2: INTERACTIONS BETWEEN CCP1 AND CCP2 FOR TIMER RESOURCES

CCP1 Mode	CCP2 Mode	Interaction
Capture	Capture	Each module can use TMR1 or TMR3 as the time base. The time base can be different for each CCP.
Capture	Compare	CCP2 can be configured for the Special Event Trigger to reset TMR1 or TMR3 (depending upon which time base is used). Automatic A/D conversions on trigger event can also be done. Operation of CCP1 could be affected if it is using the same timer as a time base.
Compare	Capture	CCP1 be configured for the Special Event Trigger to reset TMR1 or TMR3 (depending upon which time base is used). Operation of CCP2 could be affected if it is using the same timer as a time base.
Compare	Compare	Either module can be configured for the Special Event Trigger to reset the time base. Automatic A/D conversions on CCP2 trigger event can be done. Conflicts may occur if both modules are using the same time base.
Capture	PWM <sup>(1)</sup>	None
Compare	PWM <sup>(1)</sup>	None
PWM <sup>(1)</sup>	Capture	None
PWM <sup>(1)</sup>	Compare	None
PWM <sup>(1)</sup>	PWM	Both PWMs will have the same frequency and update rate (TMR2 interrupt).

Note 1: Includes standard and Enhanced PWM operation.

## 15.2 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 CCPx 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.

#### 15.2.1 CCP PIN CONFIGURATION

In Capture mode, the appropriate CCPx pin should be configured as an input by setting the corresponding TRIS direction bit.

**Note:** If RB3/CCP2 or RC1/CCP2 is configured as an output, a write to the port can cause a capture condition.

#### 15.2.2 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 15.1.1 "CCP Modules and Timer Resources").

#### 15.2.3 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.

## 15.2.4 CCP PRESCALER

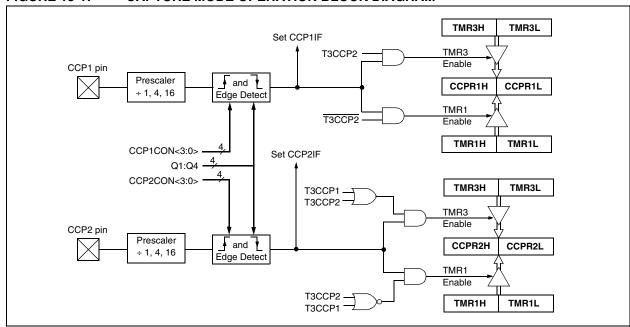
There are four prescaler settings in Capture mode. They are specified as part of the operating mode selected by the mode select bits (CCPxM3:CCPxM0). Whenever the CCP module is turned off or Capture mode is disabled, the prescaler counter is cleared. This means that any Reset will clear the prescaler 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 15-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the "false" interrupt.

# EXAMPLE 15-1: CHANGING BETWEEN CAPTURE PRESCALERS (CCP2 SHOWN)

```
CLRF CCP2CON ; Turn CCP module off
MOVLW NEW_CAPT_PS ; Load WREG with the
; new prescaler mode
; value and CCP ON
MOVWF CCP2CON ; Load CCP2CON with
; this value
```

## FIGURE 15-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



## 15.3 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 CCPx pin can be:

- · driven high
- · driven low
- toggled (high-to-low or low-to-high)
- remain 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.

#### 15.3.1 CCP PIN CONFIGURATION

The user must configure the CCPx pin as an output by clearing the appropriate TRIS bit.

Note:

Clearing the CCP2CON register will force the RB3 or RC1 compare output latch (depending on device configuration) to the default low level. This is not the PORTB or PORTC I/O data latch.

#### 15.3.2 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.

## 15.3.3 SOFTWARE INTERRUPT MODE

When the Generate Software Interrupt mode is chosen (CCPxM3:CCPxM0 = 1010), the corresponding CCPx pin is not affected. Only a CCP interrupt is generated, if enabled, and the CCPxIE bit is set.

## 15.3.4 SPECIAL EVENT TRIGGER

Both CCP modules are equipped with a Special Event Trigger. This is an internal hardware signal generated in Compare mode to trigger actions by other modules. The Special Event Trigger is enabled by selecting the Compare Special Event Trigger mode (CCPxM3:CCPxM0 = 1011).

For either CCP module, the Special Event Trigger resets the Timer register pair for whichever timer resource is currently assigned as the module's time base. This allows the CCPRx registers to serve as a programmable period register for either timer.

The Special Event Trigger for CCP2 can also start an A/D conversion. In order to do this, the A/D converter must already be enabled.

FIGURE 15-2: COMPARE MODE OPERATION BLOCK DIAGRAM

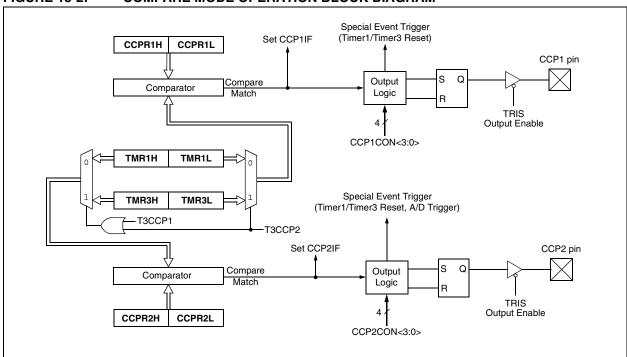


TABLE 15-3: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, TIMER1 AND TIMER3

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	51
RCON	IPEN	SBOREN <sup>(1)</sup>	1	RI	TO	PD	POR	BOR	52
PIR1	SPPIF <sup>(2)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54
PIE1	SPPIE <sup>(2)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54
IPR1	SPPIP <sup>(2)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54
PIR2	OSCFIF	CMIF	USBIF	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	54
PIE2	OSCFIE	CMIE	USBIE	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	54
IPR2	OSCFIP	CMIP	USBIP	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	54
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	54
TRISC	TRISC7	TRISC6			_	TRISC2	TRISC1	TRISC0	54
TMR1L	Timer1 Reg	gister Low By	⁄te						52
TMR1H	Timer1 Reg	gister High B	yte						52
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T10SCEN	T1SYNC	TMR1CS	TMR10N	52
TMR3H	Timer3 Reg	gister High B	yte						53
TMR3L	Timer3 Reg	gister Low By	⁄te						53
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON	53
CCPR1L	Capture/Co	mpare/PWN	l Register 1	Low Byte					53
CCPR1H	Capture/Co	mpare/PWN	l Register 1	High Byte					53
CCP1CON	P1M1 <sup>(2)</sup>	P1M0 <sup>(2)</sup>	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	53
CCPR2L	Capture/Co	mpare/PWN	l Register 2	Low Byte					53
CCPR2H	Capture/Co	mpare/PWN	l Register 2	High Byte					53
CCP2CON	_		DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	53

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by Capture/Compare, Timer1 or Timer3.

**Note 1:** The SBOREN bit is only available when BOREN<1:0> = 01; otherwise, the bit reads as '0'.

<sup>2:</sup> These bits are unimplemented on 28-pin devices; always maintain these bits clear.

#### 15.4 PWM Mode

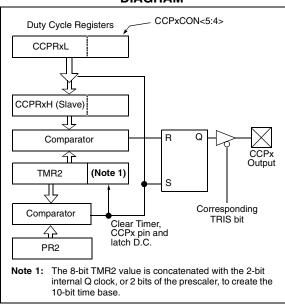
In Pulse-Width Modulation (PWM) mode, the CCPx pin produces up to a 10-bit resolution PWM output. Since the CCP2 pin is multiplexed with a PORTB or PORTC data latch, the appropriate TRIS bit must be cleared to make the CCP2 pin an output.

Note: Clearing the CCP2CON register will force the RB3 or RC1 output latch (depending on device configuration) to the default low level. This is not the PORTB or PORTC I/O data latch.

Figure 15-3 shows a simplified block diagram of the CCP module in PWM mode.

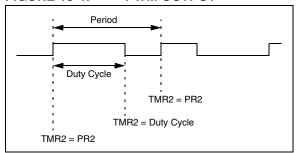
For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 15.4.4** "Setup for PWM Operation".

## FIGURE 15-3: SIMPLIFIED PWM BLOCK DIAGRAM



A PWM output (Figure 15-4) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

## FIGURE 15-4: PWM OUTPUT



#### 15.4.1 PWM PERIOD

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

#### **EQUATION 15-1:**

```
PWM Period = [(PR2) + 1] • 4 • TOSC • (TMR2 Prescale Value)
```

PWM frequency is defined as 1/[PWM period].

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

- · TMR2 is cleared
- The CCPx pin is set (exception: if PWM duty cycle = 0%, the CCPx pin will not be set)
- The PWM duty cycle is latched from CCPRxL into CCPRxH

Note: The Timer2 postscalers (see Section 13.0 "Timer2 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.

#### 15.4.2 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> bits contain the two LSbs. This 10-bit value is represented by CCPRxL:CCPxCON<5:4>. The following equation is used to calculate the PWM duty cycle in time:

## **EQUATION 15-2:**

PWM Duty Cycle = (CCPRxL:CCPxCON<5:4>) • Tosc • (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 and TMR2 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, concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCPx pin is cleared.

The maximum PWM resolution (bits) for a given PWM frequency is given by the equation:

#### **EQUATION 15-3:**

PWM Resolution (max) = 
$$\frac{\log\left(\frac{Fosc}{FPWM}\right)}{\log(2)}$$
 bits

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

TABLE 15-4: 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

## 15.4.3 PWM AUTO-SHUTDOWN (CCP1 ONLY)

The PWM auto-shutdown features of the Enhanced CCP module are also available to CCP1 in 28-pin devices. The operation of this feature is discussed in detail in **Section 16.4.7** "Enhanced PWM Auto-Shutdown".

Auto-shutdown features are not available for CCP2.

#### 15.4.4 SETUP FOR PWM OPERATION

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

- Set the PWM period by writing to the PR2 register.
- Set the PWM duty cycle by writing to the CCPRxL register and CCPxCON<5:4> bits.
- 3. Make the CCPx pin an output by clearing the appropriate TRIS bit.
- 4. Set the TMR2 prescale value, then enable Timer2 by writing to T2CON.
- 5. Configure the CCPx module for PWM operation.

TABLE 15-5: REGISTERS ASSOCIATED WITH PWM AND TIMER2

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	51	
RCON	IPEN	SBOREN <sup>(1)</sup>	_	RI	TO	PD	POR	BOR	52	
PIR1	SPPIF <sup>(2)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54	
PIE1	SPPIE <sup>(2)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54	
IPR1	SPPIP <sup>(2)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54	
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	54	
TRISC	TRISC7	TRISC6	_	_	_	TRISC2	TRISC1	TRISC0	54	
TMR2	Timer2 Register									
PR2	Timer2 Per	iod Register							52	
T2CON	_	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	52	
CCPR1L	Capture/Co	mpare/PWM	Register 1 I	_ow Byte					53	
CCPR1H	Capture/Co	mpare/PWM	Register 1 I	ligh Byte					53	
CCP1CON	P1M1 <sup>(2)</sup>	P1M0 <sup>(2)</sup>	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	53	
CCPR2L	Capture/Co	mpare/PWM	Register 2 I	_ow Byte					53	
CCPR2H	Capture/Co	mpare/PWM	Register 2 I	High Byte					53	
CCP2CON	_	_	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	53	
ECCP1AS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1 <sup>(2)</sup>	PSSBD0 <sup>(2)</sup>	53	
ECCP1DEL	PRSEN	PDC6 <sup>(2)</sup>	PDC5 <sup>(2)</sup>	PDC4 <sup>(2)</sup>	PDC3 <sup>(2)</sup>	PDC2 <sup>(2)</sup>	PDC1 <sup>(2)</sup>	PDC0 <sup>(2)</sup>	53	

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by PWM or Timer2.

**Note 1:** The SBOREN bit is only available when BOREN<1:0> = 01; otherwise, the bit reads as '0'.

2: These bits are unimplemented on 28-pin devices; always maintain these bits clear.

# 16.0 ENHANCED CAPTURE/COMPARE/PWM (ECCP) MODULE

**Note:** The ECCP module is implemented only in 40/44-pin devices.

In PIC18F4455/4550 devices, CCP1 is implemented as a standard CCP module 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 16.4 "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 16-1. It differs from the CCPxCON registers in PIC18F2255/2550 devices in that the two Most Significant bits are implemented to control PWM functionality.

## REGISTER 16-1: CCP1CON: ECCP CONTROL REGISTER (40/44-PIN DEVICES)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
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 P1M1:P1M0: Enhanced PWM Output Configuration bits

If CCP1M3:CCP1M2 = 00, 01, 10:

xx = P1A assigned as Capture/Compare input/output; P1B, P1C, P1D assigned as port pins

## If CCP1M3:CCP1M2 = 11:

00 = Single output: P1A modulated; P1B, P1C, P1D 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 DC1B1:DC1B0: PWM Duty Cycle Bit 1 and Bit 0

Capture mode:

Unused.

Compare mode:

Unused.

PWM mode:

These bits are the two LSbs of the 10-bit PWM duty cycle. The eight MSbs of the duty cycle are found in CCPR1L.

## bit 3-0 CCP1M3:CCP1M0: Enhanced CCP Mode Select bits

0000 = Capture/Compare/PWM off (resets ECCP 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 CCP1 pin low, set output on compare match (set CCP1IF)

1001 = Compare mode, initialize CCP1 pin high, clear output on compare match (set CCP1IF)

1010 = Compare mode, generate software interrupt only, CCP1 pin reverts to I/O state

1011 = Compare mode, trigger special event (CCP1 resets TMR1 or TMR3, sets CCP1IF bit)

1100 = PWM mode: P1A, P1C active-high; P1B, P1D active-high

1101 = PWM mode: P1A, P1C active-high; P1B, P1D active-low

1110 = PWM mode: P1A, P1C active-low; P1B, P1D active-high

1111 = PWM mode: P1A, P1C active-low; P1B, P1D active-low

In addition to the expanded range of modes available through the CCP1CON register, the ECCP module has two additional registers associated with Enhanced PWM operation and auto-shutdown features. They are:

- ECCP1DEL (Dead-Band Delay)
- ECCP1AS (Auto-Shutdown Configuration)

## 16.1 ECCP Outputs and Configuration

The Enhanced CCP module may have up to four PWM outputs, depending on the selected operating mode. These outputs, designated P1A through P1D, are multiplexed with I/O pins on PORTC and PORTD. The outputs that are active depend on the CCP operating mode selected. The pin assignments are summarized in Table 16-1.

To configure the I/O pins as PWM outputs, the proper PWM mode must be selected by setting the P1M1:P1M0 and CCP1M3:CCP1M0 bits. The appropriate TRISC and TRISD direction bits for the port pins must also be set as outputs.

## 16.1.1 ECCP MODULES AND TIMER RESOURCES

Like the standard CCP modules, the ECCP module can utilize Timers 1, 2 or 3, depending on the mode selected. Timer1 and Timer3 are available for modules in Capture or Compare modes, while Timer2 is available for modules in PWM mode. Interactions between the standard and Enhanced CCP modules are identical to those described for standard CCP modules. Additional details on timer resources are provided in Section 15.1.1 "CCP Modules and Timer Resources".

## 16.2 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 CCP. These are discussed in detail in Section 15.2 "Capture Mode" and Section 15.3 "Compare Mode".

#### 16.2.1 SPECIAL EVENT TRIGGER

The Special Event Trigger output of ECCP resets the TMR1 or TMR3 register pair, depending on which timer resource is currently selected. This allows the CCPR1H:CCPR1L registers to effectively be a 16-bit programmable period register for Timer1 or Timer3.

## 16.3 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 15.4** "**PWM Mode**". This is also sometimes referred to as "Compatible CCP" mode, as in Table 16-1.

Note: When setting up single output PWM operations, users are free to use either of the processes described in Section 15.4.4 "Setup for PWM Operation" or Section 16.4.9 "Setup for PWM Operation". The latter is more generic but will work for either single or multi-output PWM.

TABLE 16-1: PIN ASSIGNMENTS FOR VARIOUS ECCP1 MODES

ECCP Mode	CCP1CON Configuration	RC2 RD5		RD6	RD7					
All PIC18F4455/4550 devices:										
Compatible CCP	00xx 11xx	CCP1	RD5/SPP5	RD6/SPP6	RD7/SPP7					
Dual PWM	10xx 11xx	P1A	P1B	RD6/SPP6	RD7/SPP7					
Quad PWM	x1xx 11xx	P1A	P1B	P1C	P1D					

**Legend:** x = Don't care. Shaded cells indicate pin assignments not used by ECCP in a given mode.

#### 16.4 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 P1A through P1D. 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 P1M1:P1M0 and CCP1M3:CCP1M0 bits of the CCP1CON register.

Figure 16-1 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 PWM Dead-Band Delay register, ECCP1DEL, 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.

#### 16.4.1 PWM PERIOD

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

#### **EQUATION 16-1:**

PWM Period = 
$$[(PR2) + 1] \cdot 4 \cdot TOSC \cdot$$
  
(TMR2 Prescale Value)

PWM frequency is defined as 1/[PWM period]. When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

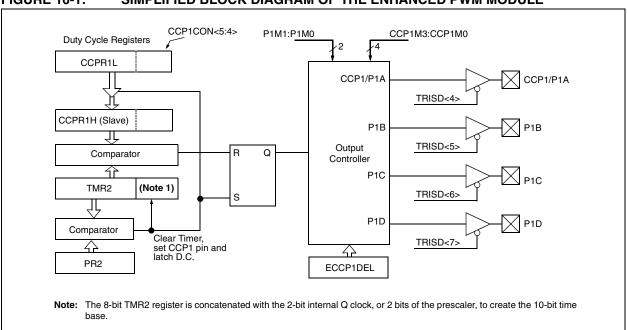
· TMR2 is cleared

Note:

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

The Timer2 postscaler (see Section 13.0 "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 16-1: SIMPLIFIED BLOCK DIAGRAM OF THE ENHANCED PWM MODULE



© 2006 Microchip Technology Inc. Preliminary DS39632C-page 151

#### 16.4.2 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 16-2:**

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 CCP1 pin is cleared. The maximum PWM resolution (bits) for a given PWM frequency is given by the following equation.

#### **EQUATION 16-3:**

PWM Resolution (max) = 
$$\frac{\log\left(\frac{FOSC}{FPWM}\right)}{\log(2)}$$
 bits

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

## 16.4.3 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

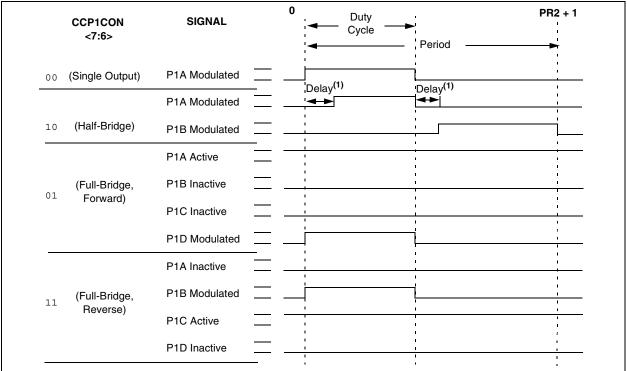
The Single Output mode is the standard PWM mode discussed in **Section 16.4** "**Enhanced PWM Mode**". The Half-Bridge and Full-Bridge Output modes are covered in detail in the sections that follow.

The general relationship of the outputs in all configurations is summarized in Figure 16-2 and Figure 16-3.

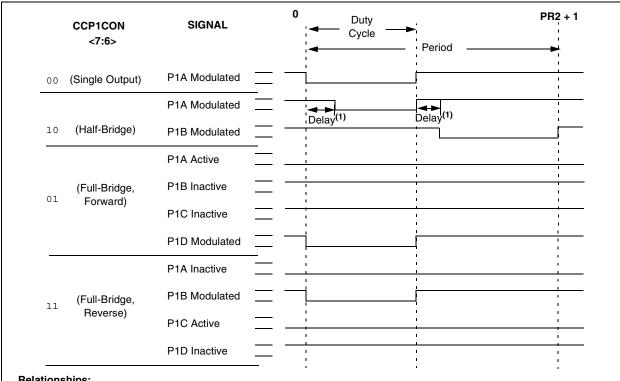
TABLE 16-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz

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 16-3: PWM OUTPUT RELATIONSHIPS (ACTIVE-LOW STATE)** 



## Relationships:

- Period = 4 \* Tosc \* (PR2 + 1) \* (TMR2 Prescale Value)
- Duty Cycle = Tosc \* (CCPR1L<7:0>:CCP1CON<5:4>) \* (TMR2 Prescale Value)
- Delay = 4 \* Tosc \* (ECCP1DEL<6:0>)

Note 1: Dead-band delay is programmed using the ECCP1DEL register (Section 16.4.6 "Programmable Dead-Band Delay").

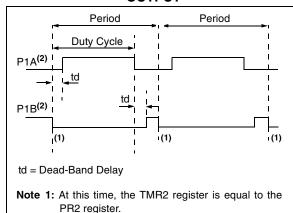
#### 16.4.4 HALF-BRIDGE MODE

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 16-4). This mode can be used for half-bridge applications, as shown in Figure 16-5, 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 PDC6:PDC0 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 16.4.6** "**Programmable Dead-Band Delay**" for more details of the dead-band delay operations.

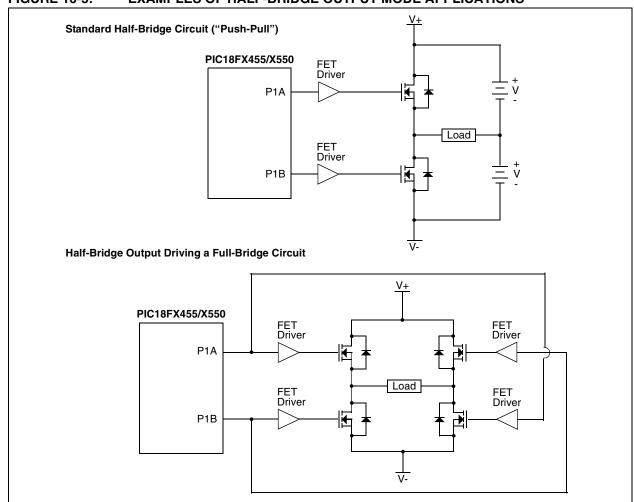
Since the P1A and P1B outputs are multiplexed with the PORTC<2> and PORTD<5> data latches, the TRISC<2> and TRISD<5> bits must be cleared to configure P1A and P1B as outputs.

FIGURE 16-4: HALF-BRIDGE PWM OUTPUT



2: Output signals are shown as active-high.

FIGURE 16-5: EXAMPLES OF HALF-BRIDGE OUTPUT MODE APPLICATIONS

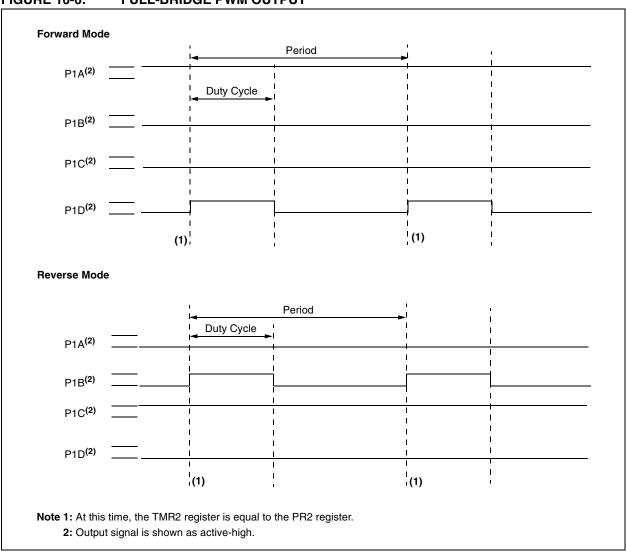


## 16.4.5 FULL-BRIDGE MODE

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 16-6.

P1A, P1B, P1C and P1D outputs are multiplexed with the PORTC<2>, PORTD<5>, PORTD<6> and PORTD<7> data latches. The TRISC<2>, TRISD<5>, TRISD<6> and TRISD<7> bits must be cleared to make the P1A, P1B, P1C and P1D pins outputs.

FIGURE 16-6: FULL-BRIDGE PWM OUTPUT



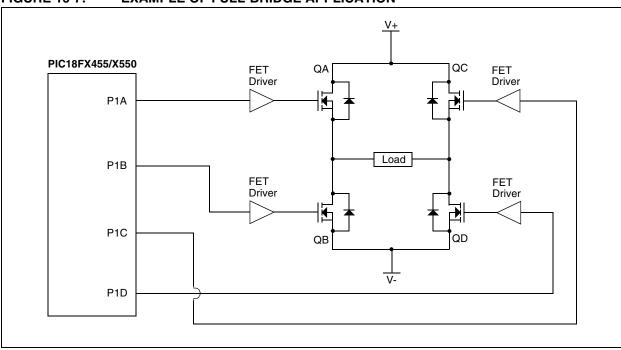


FIGURE 16-7: EXAMPLE OF FULL-BRIDGE APPLICATION

## 16.4.5.1 Direction Change in Full-Bridge Mode

In the Full-Bridge Output mode, the P1M1 bit in the CCP1CON register allows the user 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 Tosc \* (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 T2CKPS1:T2CKPS0 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 16-8.

Note that in the Full-Bridge Output mode, the ECCP 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:

- The direction of the PWM output changes when the duty cycle of the output is at or near 100%.
- The turn-off time of the power switch, including the power device and driver circuit, is greater than the turn-on time.

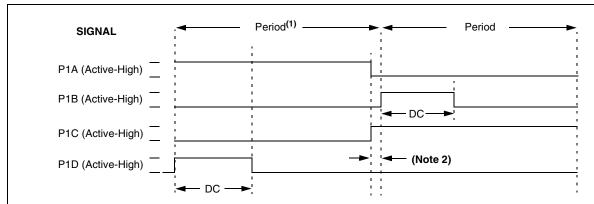
Figure 16-9 shows an example where the PWM direction changes from forward to reverse at a near 100% duty cycle. At time t1, the outputs, P1A and P1D, become inactive, while output P1C becomes active. In this example, since the turn-off time of the power devices is longer than the turn-on time, a shoot-through current may flow through power devices, QC and QD, (see Figure 16-7) 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.
- 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.

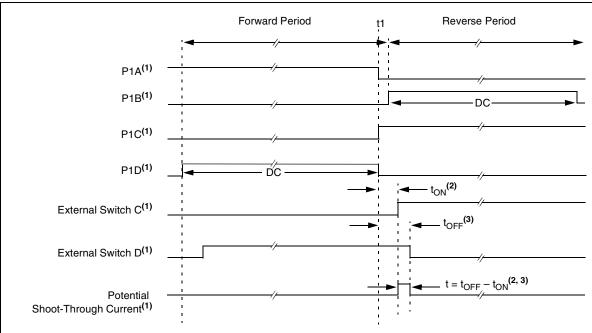
FIGURE 16-8: PWM DIRECTION CHANGE



Note 1: The direction bit in the CCP1 Control register (CCP1CON<7>) is written any time during the PWM cycle.

2: When changing directions, the P1A and P1C signals switch before the end of the current PWM cycle at intervals of 4 Tosc, 16 Tosc or 64 Tosc, depending on the Timer2 prescaler value. The modulated P1B and P1D signals are inactive at this time.

FIGURE 16-9: PWM DIRECTION CHANGE AT NEAR 100% DUTY CYCLE



Note 1: All signals are shown as active-high.

2: t<sub>ON</sub> is the turn-on delay of power switch QC and its driver.

**3:** t<sub>OFF</sub> is the turn-off delay of power switch QD and its driver.

© 2006 Microchip Technology Inc. Preliminary DS39632C-page 157

#### 16.4.6 PROGRAMMABLE DEAD-BAND **DELAY**

Note: Programmable dead-band delay is not implemented in 28-pin devices with standard CCP modules.

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 16-4 for illustration. Bits PDC6:PDC0 of the ECCP1DEL register (Register 16-2) set the delay period in terms of microcontroller instruction cycles (Tcy or 4 Tosc). These bits are not available on 28-pin devices, as the standard CCP module does not support half-bridge operation.

#### 16.4.7 ENHANCED PWM AUTO-SHUTDOWN

When ECCP 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 comparator modules, a low level on the RB0/AN12/INT0/FLT0/SDI/SDA 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 digital signal on the INTO 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 ECCPAS2:ECCPAS0 bits (bits<6:4> of the ECCP1AS register).

When a shutdown occurs, the output pins are asynchronously placed in their shutdown states, specified bν the PSSAC1:PSSAC0 PSSBD1:PSSBD0 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 ECCPASE bit (ECCP1AS<7>) is also set to hold the Enhanced PWM outputs in their shutdown states.

The ECCPASE bit is set by hardware when a shutdown event occurs. If automatic restarts are not enabled, the ECCPASE bit is cleared by firmware when the cause of the shutdown clears. If automatic restarts are enabled, the ECCPASE bit is automatically cleared when the cause of the auto-shutdown has cleared.

If the ECCPASE bit is set when a PWM period begins, the PWM outputs remain in their shutdown state for that entire PWM period. When the ECCPASE bit is cleared, the PWM outputs will return to normal operation at the beginning of the next PWM period.

Writing to the ECCPASE bit is disabled Note: while a shutdown condition is active.

#### REGISTER 16-2: ECCP1DEL: PWM DEAD-BAND 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
PRSEN	PDC6 <sup>(1)</sup>	PDC5 <sup>(1)</sup>	PDC4 <sup>(1)</sup>	PDC3 <sup>(1)</sup>	PDC2 <sup>(1)</sup>	PDC1 <sup>(1)</sup>	PDC0 <sup>(1)</sup>
bit 7							bit 0

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

bit 7 PRSEN: PWM Restart Enable bit

- 1 = Upon auto-shutdown, the ECCPASE bit clears automatically once the shutdown event goes away; the PWM restarts automatically
- 0 = Upon auto-shutdown, ECCPASE must be cleared in software to restart the PWM)

PDC6:PDC0: PWM Delay Count bits<sup>(1)</sup> bit 6-0

> Delay time, in number of Fosc/4 (4 \* Tosc) cycles, between the scheduled and actual time for a PWM signal to transition to active.

Note 1: Reserved on 28-pin devices; maintain these bits clear.

## REGISTER 16-3: ECCP1AS: ENHANCED CAPTURE/COMPARE/PWM 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
ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1 <sup>(1)</sup>	PSSBD0 <sup>(1)</sup>
bit 7							bit 0

Legend:W = Writable bitU = Unimplemented bit, read as '0'-n = Value at POR'1' = Bit is set'0' = Bit is clearedx = Bit is unknown

bit 7 ECCPASE: ECCP Auto-Shutdown Event Status bit

1 = A shutdown event has occurred; ECCP outputs are in shutdown state

0 = ECCP outputs are operating

bit 6-4 ECCPAS2:ECCPAS0: ECCP Auto-Shutdown Source Select bits

111 = FLT0 or Comparator 1 or Comparator 2

110 = FLT0 or Comparator 2

101 = FLT0 or Comparator 1

100 = FLT0

011 = Either Comparator 1 or 2

010 = Comparator 2 output

001 = Comparator 1 output

000 = Auto-shutdown is disabled

bit 3-2 PSSAC1:PSSAC0: Pins A and C Shutdown State Control bits

1x = Pins A and C tri-state (40/44-pin devices)

01 = Drive Pins A and C to '1'

00 = Drive Pins A and C to '0'

bit 1-0 PSSBD1:PSSBD0: Pins B and D Shutdown State Control bits<sup>(1)</sup>

1x = Pins B and D tri-state

01 = Drive Pins B and D to '1'

00 = Drive Pins B and D to '0'

Note 1: Reserved on 28-pin devices; maintain these bits clear.

#### 16.4.7.1 Auto-Shutdown and Auto-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 PRSEN bit of the ECCP1DEL register (ECCP1DEL<7>).

In Shutdown mode with PRSEN = 1 (Figure 16-10), the ECCPASE 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 PRSEN = 0 (Figure 16-11), once a shutdown condition occurs, the ECCPASE bit will remain set until it is cleared by firmware. Once ECCPASE is cleared, the Enhanced PWM will resume at the beginning of the next PWM period.

**Note:** Writing to the ECCPASE bit is disabled while a shutdown condition is active.

Independent of the PRSEN bit setting, if the auto-shutdown source is one of the comparators, the shutdown condition is a level. The ECCPASE 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 ECCPASE bit.

#### 16.4.8 START-UP CONSIDERATIONS

When the ECCP 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).

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 ECCP module may cause damage to the application circuit. The ECCP 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.



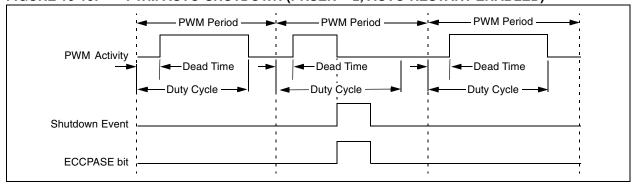
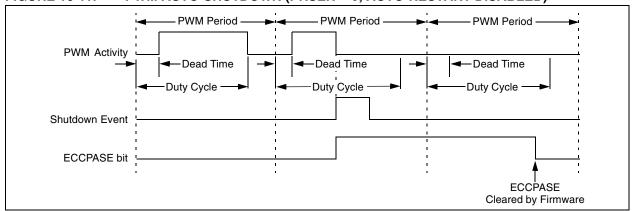


FIGURE 16-11: PWM AUTO-SHUTDOWN (PRSEN = 0, AUTO-RESTART DISABLED)



#### 16.4.9 SETUP FOR PWM OPERATION

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

- Configure the PWM pins, P1A and P1B (and P1C and P1D, if used), as inputs by setting the corresponding TRIS bits.
- 2. Set the PWM period by loading the PR2 register.
- 3. If Auto-Shutdown is required do the following:
  - Disable Auto-Shutdown (ECCPASE = 0)
  - Configure source (FLT0, Comparator 1 or Comparator 2)
  - · Wait for non-shutdown condition
- Configure the ECCP module for the desired PWM mode and configuration by loading the CCP1CON register with the appropriate values:
  - Select one of the available output configurations and direction with the P1M1:P1M0 bits.
  - Select the polarities of the PWM output signals with the CCP1M3:CCP1M0 bits.
- Set the PWM duty cycle by loading the CCPR1L register and CCP1CON<5:4> bits.
- For Half-Bridge Output mode, set the dead-band delay by loading ECCP1DEL<6:0> with the appropriate value.
- If auto-shutdown operation is required, load the ECCP1AS register:
  - Select the auto-shutdown sources using the ECCPAS2:ECCPAS0 bits.
  - Select the shutdown states of the PWM output pins using the PSSAC1:PSSAC0 and PSSBD1:PSSBD0 bits.
  - Set the ECCPASE bit (ECCP1AS<7>).
  - Configure the comparators using the CMCON register.
  - Configure the comparator inputs as analog inputs.
- 8. If auto-restart operation is required, set the PRSEN bit (ECCP1DEL<7>).
- 9. Configure and start TMR2:
  - Clear the TMR2 interrupt flag bit by clearing the TMR2IF bit (PIR1<1>).
  - Set the TMR2 prescale value by loading the T2CKPS bits (T2CON<1:0>).
  - Enable Timer2 by setting the TMR2ON bit (T2CON<2>).
- Enable PWM outputs after a new PWM cycle has started:
  - Wait until TMRn overflows (TMRnIF bit is set).
  - Enable the CCP1/P1A, P1B, P1C and/or P1D pin outputs by clearing the respective TRIS bits.
  - Clear the ECCPASE bit (ECCP1AS<7>).

## 16.4.10 OPERATION IN POWER-MANAGED MODES

In Sleep mode, all clock sources are disabled. Timer2 will not increment and the state of the module will not change. If the ECCP pin is driving a value, it will continue to drive that value. When the device wakes up, it will continue from this state. If Two-Speed Start-ups are enabled, the initial start-up frequency from INTOSC and the postscaler may not be stable immediately.

In PRI\_IDLE mode, the primary clock will continue to clock the ECCP module without change. In all other power-managed modes, the selected power-managed mode clock will clock Timer2. Other power-managed mode clocks will most likely be different than the primary clock frequency.

## 16.4.10.1 Operation with Fail-Safe Clock Monitor

If the Fail-Safe Clock Monitor is enabled, a clock failure will force the device into the power-managed RC\_RUN mode and the OSCFIF bit (PIR2<7>) will be set. The ECCP will then be clocked from the internal oscillator clock source, which may have a different clock frequency than the primary clock.

See the previous section for additional details.

#### 16.4.11 EFFECTS OF A RESET

Both Power-on Reset and subsequent Resets will force all ports to Input mode and the CCP registers to their Reset states.

This forces the Enhanced CCP module to reset to a state compatible with the standard CCP module.

TABLE 16-3: REGISTERS ASSOCIATED WITH ECCP MODULE AND TIMER1 TO 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	INTOIF	RBIF	51
RCON	IPEN	SBOREN <sup>(1)</sup>	_	RI	TO	PD	POR	BOR	52
IPR1	SPPIP <sup>(2)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54
PIR1	SPPIF <sup>(2)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54
PIE1	SPPIE <sup>(2)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54
IPR2	OSCFIP	CMIP	USBIP	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	54
PIR2	OSCFIF	CMIF	USBIF	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	54
PIE2	OSCFIE	CMIE	USBIE	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	54
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	54
TRISC	TRISC7	TRISC6	_	_	_	TRISC2	TRISC1	TRISC0	54
TRISD <sup>(2)</sup>	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	54
TMR1L	Timer1 Register Low Byte								
TMR1H	Timer1 Reg	jister High Byt	е						52
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR10N	52
TMR2	Timer2 Mod	dule Register							52
T2CON	_	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	52
PR2	Timer2 Per	iod Register							52
TMR3L	Timer3 Reg	ister Low Byte	9						53
TMR3H	Timer3 Reg	jister High Byt	е						53
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON	53
CCPR1L	Capture/Co	mpare/PWM I	Register 1 (LS	SB)			•	•	53
CCPR1H	Capture/Co	mpare/PWM I	Register 1 (MS	SB)					53
CCP1CON	P1M1 <sup>(2)</sup>	P1M0 <sup>(2)</sup>	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	53
ECCP1AS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1 <sup>(2)</sup>	PSSBD0 <sup>(2)</sup>	53
ECCP1DEL	PRSEN	PDC6 <sup>(2)</sup>	PDC5 <sup>(2)</sup>	PDC4 <sup>(2)</sup>	PDC3 <sup>(2)</sup>	PDC2 <sup>(2)</sup>	PDC1 <sup>(2)</sup>	PDC0 <sup>(2)</sup>	53

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used during ECCP operation.

Note 1: The SBOREN bit is only available when BOREN<1:0> = 01; otherwise, the bit reads as '0'.

<sup>2:</sup> These bits or registers are unimplemented in 28-pin devices; always maintain these bits clear.

## 17.0 UNIVERSAL SERIAL BUS (USB)

This section describes the details of the USB peripheral. Because of the very specific nature of the module, knowledge of USB is expected. Some high-level USB information is provided in **Section 17.10 "Overview of USB"** only for application design reference. Designers are encouraged to refer to the official specification published by the USB Implementers Forum (USB-IF) for the latest information. USB Specification Revision 2.0 is the most current specification at the time of publication of this document.

## 17.1 Overview of the USB Peripheral

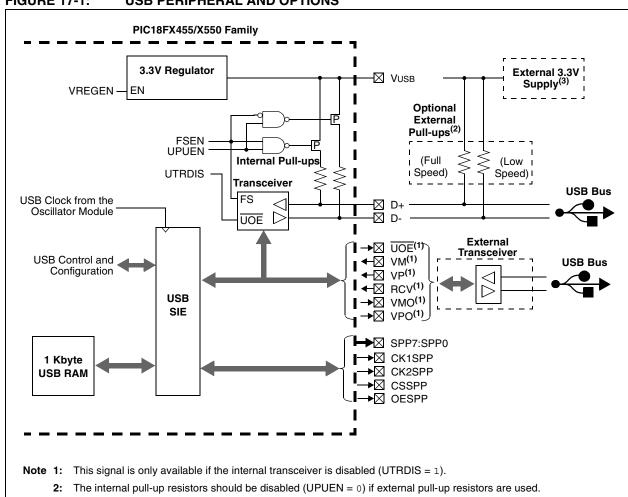
The PIC18FX455/X550 device family contains a full-speed and low-speed compatible USB Serial Interface Engine (SIE) that allows fast communication between any USB host and the PIC® microcontroller.

The SIE can be interfaced directly to the USB, utilizing the internal transceiver, or it can be connected through an external transceiver. An internal 3.3V regulator is also available to power the internal transceiver in 5V applications.

Some special hardware features have been included to improve performance. Dual port memory in the device's data memory space (USB RAM) has been supplied to share direct memory access between the microcontroller core and the SIE. Buffer descriptors are also provided, allowing users to freely program endpoint memory usage within the USB RAM space. A Streaming Parallel Port has been provided to support the uninterrupted transfer of large volumes of data, such as isochronous data, to external memory buffers.

Figure 17-1 presents a general overview of the USB peripheral and its features.

FIGURE 17-1: USB PERIPHERAL AND OPTIONS



3: Do not enable the internal regulator when using an external 3.3V supply.

#### 17.2 USB Status and Control

The operation of the USB module is configured and managed through three control registers. In addition, a total of 22 registers are used to manage the actual USB transactions. The registers are:

- USB Control register (UCON)
- USB Configuration register (UCFG)
- USB Transfer Status register (USTAT)
- USB Device Address register (UADDR)
- Frame Number registers (UFRMH:UFRML)
- Endpoint Enable registers 0 through 15 (UEPn)

## 17.2.1 USB CONTROL REGISTER (UCON)

The USB Control register (Register 17-1) contains bits needed to control the module behavior during transfers. The register contains bits that control the following:

- Main USB Peripheral Enable
- Ping-Pong Buffer Pointer Reset
- · Control of the Suspend mode
- · Packet Transfer Disable

hit 7

bit 1

bit 0

In addition, the USB Control register contains a status bit, SE0 (UCON<5>), which is used to indicate the occurrence of a single-ended zero on the bus. When the USB module is enabled, this bit should be monitored to determine whether the differential data lines have come out of a single-ended zero condition. This helps to differentiate the initial power-up state from the USB Reset signal.

The overall operation of the USB module is controlled by the USBEN bit (UCON<3>). Setting this bit activates the module and resets all of the PPBI bits in the Buffer Descriptor Table to '0'. This bit also activates the on-chip voltage regulator and connects internal pull-up resistors, if they are enabled. Thus, this bit can be used as a soft attach/detach to the USB. Although all status and control bits are ignored when this bit is clear, the module needs to be fully preconfigured prior to setting this bit

#### REGISTER 17-1: UCON: USB CONTROL REGISTER

Unimplemented, Dood on to

0 = Resume signaling disabledSUSPND: Suspend USB bit

Unimplemented: Read as '0'

U-0	R/W-0	R-x	R/C-0	R/W-0	R/W-0	R/W-0	U-0
_	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	_
bit 7							bit 0

Legend:	C = Clearable 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 /	Unimplemented: Read as '0'
bit 6	PPBRST: Ping-Pong Buffers Reset bit
	<ul><li>1 = Reset all Ping-Pong Buffer Pointers to the Even Buffer Descriptor (BD) banks</li><li>0 = Ping-Pong Buffer Pointers not being reset</li></ul>
bit 5	SE0: Live Single-Ended Zero Flag bit
	<ul><li>1 = Single-ended zero active on the USB bus</li><li>0 = No single-ended zero detected</li></ul>
bit 4	PKTDIS: Packet Transfer Disable bit
	<ul> <li>1 = SIE token and packet processing disabled, automatically set when a SETUP token is received</li> <li>0 = SIE token and packet processing enabled</li> </ul>
bit 3	USBEN: USB Module Enable bit
	<ul><li>1 = USB module and supporting circuitry enabled (device attached)</li><li>0 = USB module and supporting circuitry disabled (device detached)</li></ul>
bit 2	RESUME: Resume Signaling Enable bit
	1 = Resume signaling activated

1 = USB module and supporting circuitry in Power Conserve mode, SIE clock inactive

**Preliminary** 

0 = USB module and supporting circuitry in normal operation, SIE clock clocked at the configured rate

The PPBRST bit (UCON<6>) controls the Reset status when Double-Buffering mode (ping-pong buffering) is used. When the PPBRST bit is set, all Ping-Pong Buffer Pointers are set to the Even buffers. PPBRST has to be cleared by firmware. This bit is ignored in buffering modes not using ping-pong buffering.

The PKTDIS bit (UCON<4>) is a flag indicating that the SIE has disabled packet transmission and reception. This bit is set by the SIE when a SETUP token is received to allow setup processing. This bit cannot be set by the microcontroller, only cleared; clearing it allows the SIE to continue transmission and/or reception. Any pending events within the Buffer Descriptor Table will still be available, indicated within the USTAT register's FIFO buffer.

The RESUME bit (UCON<2>) allows the peripheral to perform a remote wake-up by executing Resume signaling. To generate a valid remote wake-up, firmware must set RESUME for 10 ms and then clear the bit. For more information on Resume signaling, see Sections 7.1.7.5, 11.4.4 and 11.9 in the USB 2.0 specification.

The SUSPND bit (UCON<1>) places the module and supporting circuitry (i.e., voltage regulator) in a low-power mode. The input clock to the SIE is also disabled. This bit should be set by the software in response to an IDLEIF interrupt. It should be reset by the microcontroller firmware after an ACTVIF interrupt is observed. When this bit is active, the device remains attached to the bus but the transceiver outputs remain Idle. The voltage on the VUSB pin may vary depending on the value of this bit. Setting this bit before a IDLEIF request will result in unpredictable bus behavior.

Note: While in Suspend mode, a typical bus powered USB device is limited to  $500~\mu A$  of current. This is the complete current drawn by the PICmicro device and its supporting circuitry. Care should be taken to assure minimum current draw when the device enters Suspend mode.

## 17.2.2 USB CONFIGURATION REGISTER (UCFG)

Prior to communicating over USB, the module's associated internal and/or external hardware must be configured. Most of the configuration is performed with the UCFG register (Register 17-2). The separate USB voltage regulator (see **Section 17.2.2.8 "Internal Regulator"**) is controlled through the Configuration registers.

The UFCG register contains most of the bits that control the system level behavior of the USB module. These include:

- Bus Speed (full speed versus low speed)
- On-Chip Pull-up Resistor Enable
- · On-Chip Transceiver Enable
- · Ping-Pong Buffer Usage

The UCFG register also contains two bits which aid in module testing, debugging and USB certifications. These bits control output enable state monitoring and eye pattern generation.

Note: The USB speed, transceiver and pull-up should only be configured during the module setup phase. It is not recommended to switch these settings while the module is enabled.

## 17.2.2.1 Internal Transceiver

The USB peripheral has a built-in, USB 2.0, full-speed and low-speed compliant transceiver, internally connected to the SIE. This feature is useful for low-cost single chip applications. The UTRDIS bit (UCFG<3>) controls the transceiver; it is enabled by default (UTRDIS = 0). The FSEN bit (UCFG<2>) controls the transceiver speed; setting the bit enables full-speed operation.

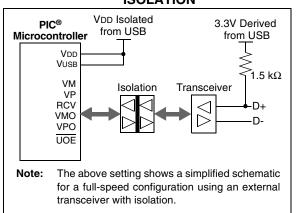
The on-chip USB pull-up resistors are controlled by the UPUEN bit (UCFG<4>). They can only be selected when the on-chip transceiver is enabled.

The USB specification requires 3.3V operation for communications; however, the rest of the chip may be running at a higher voltage. Thus, the transceiver is supplied power from a separate source, VUSB.

## 17.2.2.2 External Transceiver

This module provides support for use with an off-chip transceiver. The off-chip transceiver is intended for applications where physical conditions dictate the location of the transceiver to be away from the SIE. For example, applications that require isolation from the USB could use an external transceiver through some isolation to the microcontroller's SIE (Figure 17-2). External transceiver operation is enabled by setting the UTRDIS bit.

# FIGURE 17-2: TYPICAL EXTERNAL TRANSCEIVER WITH ISOLATION



## REGISTER 17-2: UCFG: USB CONFIGURATION REGISTER

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UTEYE	UOEMON <sup>(1)</sup>	_	UPUEN <sup>(2,3)</sup>	UTRDIS <sup>(2)</sup>	FSEN <sup>(2)</sup>	PPB1	PPB0
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 UTEYE: USB Eye Pattern Test Enable bit

1 = Eye pattern test enabled0 = Eye pattern test disabled

bit 6 **UOEMON:** USB OE Monitor Enable bit<sup>(1)</sup>

1 = UOE signal active; it indicates intervals during which the D+/D- lines are driving

 $0 = \overline{\mathsf{UOE}}$  signal inactive

bit 5 Unimplemented: Read as '0'

bit 4 **UPUEN:** USB On-Chip Pull-up Enable bit (2,3)

1 = On-chip pull-up enabled (pull-up on D+ with FSEN = 1 or D- with FSEN = 0)

0 = On-chip pull-up disabled

bit 3 **UTRDIS:** On-Chip Transceiver Disable bit<sup>(2)</sup>

1 = On-chip transceiver disabled; digital transceiver interface enabled

0 = On-chip transceiver active

bit 2 FSEN: Full-Speed Enable bit<sup>(2)</sup>

1 = Full-speed device: controls transceiver edge rates; requires input clock at 48 MHz

0 = Low-speed device: controls transceiver edge rates; requires input clock at 6 MHz

bit 1-0 PPB1:PPB0: Ping-Pong Buffers Configuration bits

11 = Even/Odd ping-pong buffers enabled for Endpoints 1 to 15

10 = Even/Odd ping-pong buffers enabled for all endpoints

01 = Even/Odd ping-pong buffer enabled for OUT Endpoint 0

00 = Even/Odd ping-pong buffers disabled

Note 1: If UTRDIS is set, the UOE signal will be active independent of the UOEMON bit setting.

2: The UPUEN, UTRDIS and FSEN bits should never be changed while the USB module is enabled. These values must be preconfigured prior to enabling the module.

3: This bit is only valid when the on-chip transceiver is active (UTRDIS = 0); otherwise, it is ignored.

There are 6 signals from the module to communicate with and control an external transceiver:

- VM: Input from the single-ended D- line
- VP: Input from the single-ended D+ line
- RCV: Input from the differential receiver
- VMO: Output to the differential line driver
- . VPO: Output to the differential line driver
- UOE: Output enable

The VPO and VMO signals are outputs from the SIE to the external transceiver. The RCV signal is the output from the external transceiver to the SIE; it represents the differential signals from the serial bus translated into a single pulse train. The VM and VP signals are used to report conditions on the serial bus to the SIE that can't be captured with the RCV signal. The combinations of states of these signals and their interpretation are listed in Table 17-1 and Table 17-2.

TABLE 17-1: DIFFERENTIAL OUTPUTS TO TRANSCEIVER

	VPO	VMO	Bus State
	0	0	Single-Ended Zero
	0	1	Differential '0'
1 0 Differential '1'		Differential '1'	
	1	1	Illegal Condition

TABLE 17-2: SINGLE-ENDED INPUTS FROM TRANSCEIVER

VP	VM	Bus State
0	0	Single-Ended Zero
0	1	Low Speed
1	0	High Speed
1	1	Error

The UOE signal toggles the state of the external transceiver. This line is pulled low by the device to enable the transmission of data from the SIE to an external device.

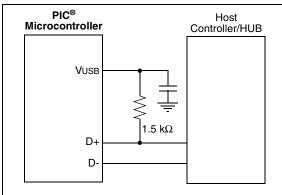
## 17.2.2.3 Internal Pull-up Resistors

The PIC18FX455/X550 devices have built-in pull-up resistors designed to meet the requirements for low-speed and full-speed USB. The UPUEN bit (UCFG<4>) enables the internal pull-ups. Figure 17-1 shows the pull-ups and their control.

## 17.2.2.4 External Pull-up Resistors

External pull-up may also be used. The VUSB pin may be used to pull up D+ or D-. The pull-up resistor must be 1.5 k $\Omega$  ( $\pm5\%$ ) as required by the USB specifications. Figure 17-3 shows an example.

FIGURE 17-3: EXTERNAL CIRCUITRY



Note: The above setting shows a typical connection for a full-speed configuration using an on-chip regulator and an external pull-up resistor.

## 17.2.2.5 Ping-Pong Buffer Configuration

The usage of ping-pong buffers is configured using the PPB1:PPB0 bits. Refer to **Section 17.4.4 "Ping-Pong Buffering"** for a complete explanation of the ping-pong buffers.

## 17.2.2.6 USB Output Enable Monitor

The USB  $\overline{\text{OE}}$  monitor provides indication as to whether the SIE is listening to the bus or actively driving the bus. This is enabled by default when using an external transceiver or when UCFG<6> = 1.

The USB  $\overline{\text{OE}}$  monitoring is useful for initial system debugging, as well as scope triggering during eye pattern generation tests.

## 17.2.2.7 Eye Pattern Test Enable

An automatic eye pattern test can be generated by the module when the UCFG<7> bit is set. The eye pattern output will be observable based on module settings, meaning that the user is first responsible for configuring the SIE clock settings, pull-up resistor and Transceiver mode. In addition, the module has to be enabled.

Once UTEYE is set, the module emulates a switch from a receive to transmit state and will start transmitting a J-K-J-K bit sequence (K-J-K-J for full speed). The sequence will be repeated indefinitely while the Eye Pattern Test mode is enabled.

Note that this bit should never be set while the module is connected to an actual USB system. This test mode is intended for board verification to aid with USB certification tests. It is intended to show a system developer the noise integrity of the USB signals which can be affected by board traces, impedance mismatches and proximity to other system components. It does not properly test the transition from a receive to a transmit state. Although the eye pattern is not meant to replace the more complex USB certification test, it should aid during first order system debugging.

## 17.2.2.8 Internal Regulator

The PIC18FX455/X550 devices have a built-in 3.3V regulator to provide power to the internal transceiver and provide a source for the internal/external pull-ups. An external 220 nF (±20%) capacitor is required for stability.

**Note:** The drive from VUSB is sufficient to only drive an external pull-up in addition to the internal transceiver.

The regulator is enabled by default and can be disabled through the VREGEN Configuration bit. When enabled, the voltage is visible on pin VUSB. When the regulator is disabled, a 3.3V source must be provided through the VUSB pin for the internal transceiver. If the internal transceiver is disabled, VUSB is not used.

- **Note 1:** Do not enable the internal regulator if an external regulator is connected to VUSB.
  - 2: VDD must be greater than VUSB at all times, even with the regulator disabled.

## 17.2.3 USB STATUS REGISTER (USTAT)

The USB Status register reports the transaction status within the SIE. When the SIE issues a USB transfer complete interrupt, USTAT should be read to determine the status of the transfer. USTAT contains the transfer endpoint number, direction and Ping-Pong Buffer Pointer value (if used).

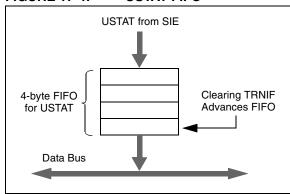
**Note:** The data in the USB Status register is valid only when the TRNIF interrupt flag is asserted.

The USTAT register is actually a read window into a four-byte status FIFO, maintained by the SIE. It allows the microcontroller to process one transfer while the SIE processes additional endpoints (Figure 17-4). When the SIE completes using a buffer for reading or writing data, it updates the USTAT register. If another USB transfer is performed before a transaction complete interrupt is serviced, the SIE will store the status of the next transfer into the status FIFO.

Clearing the transfer complete flag bit, TRNIF, causes the SIE to advance the FIFO. If the next data in the FIFO holding register is valid, the SIE will immediately reassert the interrupt. If no additional data is present, TRNIF will remain clear; USTAT data will no longer be reliable.

Note: If an endpoint request is received while the USTAT FIFO is full, the SIE will automatically issue a NAK back to the host.

FIGURE 17-4: USTAT FIFO



#### REGISTER 17-3: USTAT: USB STATUS REGISTER

U-0	R-x	R-x	R-x	R-x	R-x	R-x	U-0
_	ENDP3	ENDP2	ENDP1	ENDP0	DIR	PPBI <sup>(1)</sup>	_
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 **ENDP3:ENDP0:** Encoded Number of Last Endpoint Activity bits

(represents the number of the BDT updated by the last USB transfer)

1111 = Endpoint 15 1110 = Endpoint 14

...

0001 = Endpoint 1 0000 = Endpoint 0

bit 2 DIR: Last BD Direction Indicator bit

1 = The last transaction was an IN token

0 = The last transaction was an OUT or SETUP token

bit 1 **PPBI:** Ping-Pong BD Pointer Indicator bit<sup>(1)</sup>

1 = The last transaction was to the Odd BD bank 0 = The last transaction was to the Even BD bank

bit 0 Unimplemented: Read as '0'

Note 1: This bit is only valid for endpoints with available Even and Odd BD registers.

#### 17.2.4 USB ENDPOINT CONTROL

Each of the 16 possible bidirectional endpoints has its own independent control register, UEPn (where 'n' represents the endpoint number). Each register has an identical complement of control bits. The prototype is shown in Register 17-4.

The EPHSHK bit (UEPn<4>) controls handshaking for the endpoint; setting this bit enables USB handshaking. Typically, this bit is always set except when using isochronous endpoints.

The EPCONDIS bit (UEPn<3>) is used to enable or disable USB control operations (SETUP) through the endpoint. Clearing this bit enables SETUP transactions. Note that the corresponding EPINEN and EPOUTEN bits must be set to enable IN and OUT

transactions. For Endpoint 0, this bit should always be cleared since the USB specifications identify Endpoint 0 as the default control endpoint.

The EPOUTEN bit (UEPn<2>) is used to enable or disable USB OUT transactions from the host. Setting this bit enables OUT transactions. Similarly, the EPINEN bit (UEPn<1>) enables or disables USB IN transactions from the host.

The EPSTALL bit (UEPn<0>) is used to indicate a STALL condition for the endpoint. If a STALL is issued on a particular endpoint, the EPSTALL bit for that endpoint pair will be set by the SIE. This bit remains set until it is cleared through firmware, or until the SIE is reset.

## REGISTER 17-4: UEPn: USB ENDPOINT n CONTROL REGISTER (UEP0 THROUGH UEP15)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	— — EPHSHK EPCONDIS EPOUTE		EPOUTEN	EPINEN	EPSTALL <sup>(1)</sup>		
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 4	EPHSHK: Endpoint Handshake Enable bit				
	1 = Endpoint handshake enabled				

Unimplemented: Read as '0'

bit 7-5

0 = Endpoint handshake disabled (typically used for isochronous endpoints)

bit 3 EPCONDIS: Bidirectional Endpoint Control bit

If EPOUTEN = 1 and EPINEN = 1:

1 = Disable Endpoint n from control transfers; only IN and OUT transfers allowed
 0 = Enable Endpoint n for control (SETUP) transfers; IN and OUT transfers also allowed

bit 2 **EPOUTEN:** Endpoint Output Enable bit

1 = Endpoint n output enabled0 = Endpoint n output disabled

bit 1 **EPINEN:** Endpoint Input Enable bit

1 = Endpoint n input enabled0 = Endpoint n input disabled

bit 0 **EPSTALL:** Endpoint Stall Enable bit<sup>(1)</sup>

1 = Endpoint n is stalled0 = Endpoint n is not stalled

Note 1: Valid only if Endpoint n is enabled; otherwise, the bit is ignored.

## 17.2.5 USB ADDRESS REGISTER (UADDR)

The USB Address register contains the unique USB address that the peripheral will decode when active. UADDR is reset to 00h when a USB Reset is received, indicated by URSTIF, or when a Reset is received from the microcontroller. The USB address must be written by the microcontroller during the USB setup phase (enumeration) as part of the Microchip USB firmware support.

## 17.2.6 USB FRAME NUMBER REGISTERS (UFRMH:UFRML)

The Frame Number registers contain the 11-bit frame number. The low-order byte is contained in UFRML, while the three high-order bits are contained in UFRMH. The register pair is updated with the current frame number whenever a SOF token is received. For the microcontroller, these registers are read-only. The Frame Number register is primarily used for isochronous transfers.

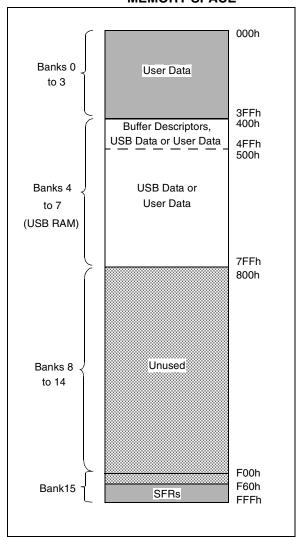
#### 17.3 USB RAM

USB data moves between the microcontroller core and the SIE through a memory space known as the USB RAM. This is a special dual port memory that is mapped into the normal data memory space in Banks 4 through 7 (400h to 7FFh) for a total of 1 Kbyte (Figure 17-5).

Bank 4 (400h through 4FFh) is used specifically for endpoint buffer control, while Banks 5 through 7 are available for USB data. Depending on the type of buffering being used, all but 8 bytes of Bank 4 may also be available for use as USB buffer space.

Although USB RAM is available to the microcontroller as data memory, the sections that are being accessed by the SIE should not be accessed by the microcontroller. A semaphore mechanism is used to determine the access to a particular buffer at any given time. This is discussed in **Section 17.4.1.1 "Buffer Ownership**".

# FIGURE 17-5: IMPLEMENTATION OF USB RAM IN DATA MEMORY SPACE



## 17.4 Buffer Descriptors and the Buffer Descriptor Table

The registers in Bank 4 are used specifically for endpoint buffer control in a structure known as the Buffer Descriptor Table (BDT). This provides a flexible method for users to construct and control endpoint buffers of various lengths and configuration.

The BDT is composed of Buffer Descriptors (BD) which are used to define and control the actual buffers in the USB RAM space. Each BD, in turn, consists of four registers, where n represents one of the 64 possible BDs (range of 0 to 63):

BDnSTAT: BD Status register
BDnCNT: BD Byte Count register
BDnADRL: BD Address Low register
BDnADRH: BD Address High register

BDs always occur as a four-byte block in the sequence, BDnSTAT:BDnCNT:BDnADRL:BDnADRH. The address of BDnSTAT is always an offset of (4n-1) (in hexadecimal) from 400h, with n being the buffer descriptor number.

Depending on the buffering configuration used (Section 17.4.4 "Ping-Pong Buffering"), there are up to 32, 33 or 64 sets of buffer descriptors. At a minimum, the BDT must be at least 8 bytes long. This is because the USB specification mandates that every device must have Endpoint 0 with both input and output for initial setup. Depending on the endpoint and buffering configuration, the BDT can be as long as 256 bytes.

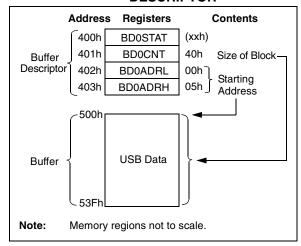
Although they can be thought of as Special Function Registers, the Buffer Descriptor Status and Address registers are not hardware mapped, as conventional microcontroller SFRs in Bank 15 are. If the endpoint corresponding to a particular BD is not enabled, its registers are not used. Instead of appearing as unimplemented addresses, however, they appear as available RAM. Only when an endpoint is enabled by setting the UEPn<1> bit does the memory at those addresses become functional as BD registers. As with any address in the data memory space, the BD registers have an indeterminate value on any device Reset.

An example of a BD for a 64-byte buffer, starting at 500h, is shown in Figure 17-6. A particular set of BD registers is only valid if the corresponding endpoint has been enabled using the UEPn register. All BD registers are available in USB RAM. The BD for each endpoint should be set up prior to enabling the endpoint.

#### 17.4.1 BD STATUS AND CONFIGURATION

Buffer descriptors not only define the size of an endpoint buffer, but also determine its configuration and control. Most of the configuration is done with the BD Status register, BDnSTAT. Each BD has its own unique and correspondingly numbered BDnSTAT register.

FIGURE 17-6: EXAMPLE OF A BUFFER DESCRIPTOR



Unlike other control registers, the bit configuration for the BDnSTAT register is context sensitive. There are two distinct configurations, depending on whether the microcontroller or the USB module is modifying the BD and buffer at a particular time. Only three bit definitions are shared between the two.

## 17.4.1.1 Buffer Ownership

Because the buffers and their BDs are shared between the CPU and the USB module, a simple semaphore mechanism is used to distinguish which is allowed to update the BD and associated buffers in memory.

This is done by using the UOWN bit (BDnSTAT<7>) as a semaphore to distinguish which is allowed to update the BD and associated buffers in memory. UOWN is the only bit that is shared between the two configurations of BDnSTAT.

When UOWN is clear, the BD entry is "owned" by the microcontroller core. When the UOWN bit is set, the BD entry and the buffer memory are "owned" by the USB peripheral. The core should not modify the BD or its corresponding data buffer during this time. Note that the microcontroller core can still read BDnSTAT while the SIE owns the buffer and vice versa.

The buffer descriptors have a different meaning based on the source of the register update. Prior to placing ownership with the USB peripheral, the user can configure the basic operation of the peripheral through the BDnSTAT bits. During this time, the byte count and buffer location registers can also be set.

When UOWN is set, the user can no longer depend on the values that were written to the BDs. From this point, the SIE updates the BDs as necessary, overwriting the original BD values. The BDnSTAT register is updated by the SIE with the token PID and the transfer count, BDnCNT, is updated.

The BDnSTAT byte of the BDT should always be the last byte updated when preparing to arm an endpoint. The SIE will clear the UOWN bit when a transaction has completed. The only exception to this is when KEN is enabled and/or BSTALL is enabled.

No hardware mechanism exists to block access when the UOWN bit is set. Thus, unexpected behavior can occur if the microcontroller attempts to modify memory when the SIE owns it. Similarly, reading such memory may produce inaccurate data until the USB peripheral returns ownership to the microcontroller.

## 17.4.1.2 BDnSTAT Register (CPU Mode)

When UOWN = 0, the microcontroller core owns the BD. At this point, the other seven bits of the register take on control functions.

The Keep Enable bit, KEN (BDnSTAT<5>), determines if a BD stays enabled. If the bit is set, once the UOWN bit is set, it will remain owned by the SIE independent of the endpoint activity. This prevents the USTAT FIFO from being updated, as well as the transaction complete interrupt from being set for the endpoint. This feature should only be enabled when the Streaming Parallel Port is selected as the data I/O channel instead of USB RAM.

The Address Increment Disable bit, INCDIS (BDnSTAT<4>), controls the SIE's automatic address increment function. Setting INCDIS disables the auto-increment of the buffer address by the SIE for each byte transmitted or received. This feature should only be enabled when using the Streaming Parallel Port, where each data byte is processed to or from the same memory location.

The Data Toggle Sync Enable bit, DTSEN (BDnSTAT<3>), controls data toggle parity checking. Setting DTSEN enables data toggle synchronization by

the SIE. When enabled, it checks the data packet's parity against the value of DTS (BDnSTAT<6>). If a packet arrives with an incorrect synchronization, the data will essentially be ignored. It will not be written to the USB RAM and the USB transfer complete interrupt flag will not be set. The SIE will send an ACK token back to the host to Acknowledge receipt, however. The effects of the DTSEN bit on the SIE are summarized in Table 17-3.

The Buffer Stall bit, BSTALL (BDnSTAT<2>), provides support for control transfers, usually one-time stalls on Endpoint 0. It also provides support for the SET\_FEATURE/CLEAR\_FEATURE commands specified in Chapter 9 of the USB specification; typically, continuous STALLs to any endpoint other than the default control endpoint.

The BSTALL bit enables buffer stalls. Setting BSTALL causes the SIE to return a STALL token to the host if a received token would use the BD in that location. The EPSTALL bit in the corresponding UEPn control register is set and a STALL interrupt is generated when a STALL is issued to the host. The UOWN bit remains set and the BDs are not changed unless a SETUP token is received. In this case, the STALL condition is cleared and the ownership of the BD is returned to the microcontroller core.

The BD9:BD8 bits (BDnSTAT<1:0>) store the two most significant digits of the SIE byte count; the lower 8 digits are stored in the corresponding BDnCNT register. See Section 17.4.2 "BD Byte Count" for more information.

TABLE 17-3: EFFECT OF DTSEN BIT ON ODD/EVEN (DATA0/DATA1) PACKET RECEPTION

OUT Packet	BDnSTAT Settings		Device Response after Receiving Packet			
from Host	DTSEN	DTS	Handshake	UOWN	TRNIF	BDnSTAT and USTAT Status
DATA0	1	0	ACK	0	1	Updated
DATA1	1	0	ACK	1	0	Not Updated
DATA0	1	1	ACK	0	1	Updated
DATA1	1	1	ACK	1	0	Not Updated
Either	0	х	ACK	0	1	Updated
Either, with error	х	х	NAK	1	0	Not Updated

**Legend:** x = don't care

## REGISTER 17-5: BDnSTAT: BUFFER DESCRIPTOR n STATUS REGISTER (BD0STAT THROUGH BD63STAT), CPU MODE (DATA IS WRITTEN TO THE SIDE)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
UOWN <sup>(1)</sup>	DTS <sup>(2)</sup>	KEN	INCDIS	DTSEN	BSTALL	BC9	BC8
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 **UOWN:** USB Own bit<sup>(1)</sup>

0 = The microcontroller core owns the BD and its corresponding buffer

bit 6 DTS: Data Toggle Synchronization bit (2)

1 = Data 1 packet0 = Data 0 packet

bit 5 KEN: BD Keep Enable bit

1 = USB will keep the BD indefinitely once UOWN is set (required for SPP endpoint configuration)

0 = USB will hand back the BD once a token has been processed

bit 4 INCDIS: Address Increment Disable bit

1 = Address increment disabled (required for SPP endpoint configuration)

0 = Address increment enabled

bit 3 DTSEN: Data Toggle Synchronization Enable bit

1 = Data toggle synchronization is enabled; data packets with incorrect Sync value will be ignored except for a SETUP transaction, which is accepted even if the data toggle bits do not match

0 = No data toggle synchronization is performed

bit 2 BSTALL: Buffer Stall Enable bit

1 = Buffer stall enabled; STALL handshake issued if a token is received that would use the BD in the given location (UOWN bit remains set, BD value is unchanged)

0 = Buffer stall disabled

bit 1-0 BC9:BC8: Byte Count 9 and 8 bits

The byte count bits represent the number of bytes that will be transmitted for an IN token or received during an OUT token. Together with BC<7:0>, the valid byte counts are 0-1023.

Note 1: This bit must be initialized by the user to the desired value prior to enabling the USB module.

2: This bit is ignored unless DTSEN = 1.

## 17.4.1.3 BDnSTAT Register (SIE Mode)

When the BD and its buffer are owned by the SIE, most of the bits in BDnSTAT take on a different meaning. The configuration is shown in Register 17-6. Once UOWN is set, any data or control settings previously written there by the user will be overwritten with data from the SIE.

The BDnSTAT register is updated by the SIE with the token Packet Identifier (PID) which is stored in BDnSTAT<5:3>. The transfer count in the corresponding BDnCNT register is updated. Values that overflow the 8-bit register carry over to the two most significant digits of the count, stored in BDnSTAT<1:0>.

#### 17.4.2 BD BYTE COUNT

The byte count represents the total number of bytes that will be transmitted during an IN transfer. After an IN transfer, the SIE will return the number of bytes sent to the host.

For an OUT transfer, the byte count represents the maximum number of bytes that can be received and stored in USB RAM. After an OUT transfer, the SIE will return the actual number of bytes received. If the number of bytes received exceeds the corresponding byte count, the data packet will be rejected and a NAK handshake will be generated. When this happens, the byte count will not be updated.

The 10-bit byte count is distributed over two registers. The lower 8 bits of the count reside in the BDnCNT register. The upper two bits reside in BDnSTAT<1:0>. This represents a valid byte range of 0 to 1023.

#### 17.4.3 BD ADDRESS VALIDATION

The BD Address register pair contains the starting RAM address location for the corresponding endpoint buffer. For an endpoint starting location to be valid, it must fall in the range of the USB RAM, 400h to 7FFh. No mechanism is available in hardware to validate the BD address.

If the value of the BD address does not point to an address in the USB RAM, or if it points to an address within another endpoint's buffer, data is likely to be lost or overwritten. Similarly, overlapping a receive buffer (OUT endpoint) with a BD location in use can yield unexpected results. When developing USB applications, the user may want to consider the inclusion of software-based address validation in their code.

# REGISTER 17-6: BDnSTAT: BUFFER DESCRIPTOR n STATUS REGISTER (BD0STAT THROUGH BD63STAT), SIE MODE (DATA RETURNED BY THE SIDE TO THE MICROCONTROLLER)

R/W-x	U-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
UOWN	_	PID3	PID2	PID1	PID0	BC9	BC8
bit 7							bit 0

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

bit 7 UOWN: USB Own bit

1 = The SIE owns the BD and its corresponding buffer

bit 6 Reserved: Not written by the SIE bit 5-2 PID3:PID0: Packet Identifier bits

The received token PID value of the last transfer (IN, OUT or SETUP transactions only).

bit 1-0 BC9:BC8: Byte Count 9 and 8 bits

These bits are updated by the SIE to reflect the actual number of bytes received on an OUT transfer

and the actual number of bytes transmitted on an IN transfer.

#### 17.4.4 PING-PONG BUFFERING

An endpoint is defined to have a ping-pong buffer when it has two sets of BD entries: one set for an Even transfer and one set for an Odd transfer. This allows the CPU to process one BD while the SIE is processing the other BD. Double-buffering BDs in this way allows for maximum throughput to/from the USB.

The USB module supports four modes of operation:

- No ping-pong support
- Ping-pong buffer support for OUT Endpoint 0 only
- · Ping-pong buffer support for all endpoints
- Ping-pong buffer support for all other Endpoints except Endpoint 0

The ping-pong buffer settings are configured using the PPB1:PPB0 bits in the UCFG register.

The USB module keeps track of the Ping-Pong Pointer individually for each endpoint. All pointers are initially reset to the Even BD when the module is enabled. After

the completion of a transaction (UOWN cleared by the SIE), the pointer is toggled to the Odd BD. After the completion of the next transaction, the pointer is toggled back to the Even BD and so on.

The Even/Odd status of the last transaction is stored in the PPBI bit of the USTAT register. The user can reset all Ping-Pong Pointers to Even using the PPBRST bit.

Figure 17-7 shows the four different modes of operation and how USB RAM is filled with the BDs.

BDs have a fixed relationship to a particular endpoint, depending on the buffering configuration. The mapping of BDs to endpoints is detailed in Table 17-4. This relationship also means that gaps may occur in the BDT if endpoints are not enabled contiguously. This theoretically means that the BDs for disabled endpoints could be used as buffer space. In practice, users should avoid using such spaces in the BDT unless a method of validating BD addresses is implemented.

FIGURE 17-7: BUFFER DESCRIPTOR TABLE MAPPING FOR BUFFERING MODES

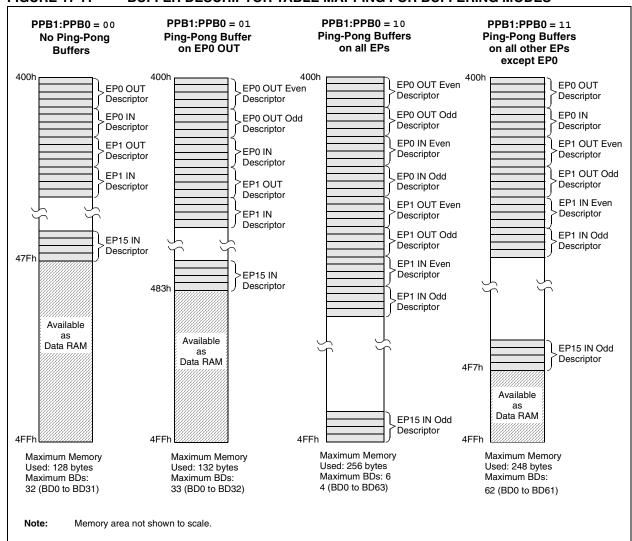


TABLE 17-4: ASSIGNMENT OF BUFFER DESCRIPTORS FOR THE DIFFERENT BUFFERING MODES

		BDs Assigned to Endpoint									
Endpoint	Mode 0 (No Ping-Pong)		Mode 1 (Ping-Pong on EP0 OUT)		Mode 2 (Ping-Pong on all EPs)		Mode 3 (Ping-Pong on all other EPs, except EP0)				
	Out	In	Out	In	Out	In	Out	In			
0	0	1	0 (E), 1 (O)	2	0 (E), 1 (O)	2 (E), 3 (O)	0	1			
1	2	3	3	4	4 (E), 5 (O)	6 (E), 7 (O)	2 (E), 3 (O)	4 (E), 5 (O)			
2	4	5	5	6	8 (E), 9 (O)	10 (E), 11 (O)	6 (E), 7 (O)	8 (E), 9 (O)			
3	6	7	7	8	12 (E), 13 (O)	14 (E), 15 (O)	10 (E), 11 (O)	12 (E), 13 (O)			
4	8	9	9	10	16 (E), 17 (O)	18 (E), 19 (O)	14 (E), 15 (O)	16 (E), 17 (O)			
5	10	11	11	12	20 (E), 21 (O)	22 (E), 23 (O)	18 (E), 19 (O)	20 (E), 21 (O)			
6	12	13	13	14	24 (E), 25 (O)	26 (E), 27 (O)	22 (E), 23 (O)	24 (E), 25 (O)			
7	14	15	15	16	28 (E), 29 (O)	30 (E), 31 (O)	26 (E), 27 (O)	28 (E), 29 (O)			
8	16	17	17	18	32 (E), 33 (O)	34 (E), 35 (O)	30 (E), 31 (O)	32 (E), 33 (O)			
9	18	19	19	20	36 (E), 37 (O)	38 (E), 39 (O)	34 (E), 35 (O)	36 (E), 37 (O)			
10	20	21	21	22	40 (E), 41 (O)	42 (E), 43 (O)	38 (E), 39 (O)	40 (E), 41 (O)			
11	22	23	23	24	44 (E), 45 (O)	46 (E), 47 (O)	42 (E), 43 (O)	44 (E), 45 (O)			
12	24	25	25	26	48 (E), 49 (O)	50 (E), 51 (O)	46 (E), 47 (O)	48 (E), 49 (O)			
13	26	27	27	28	52 (E), 53 (O)	54 (E), 55 (O)	50 (E), 51 (O)	52 (E), 53 (O)			
14	28	29	29	30	56 (E), 57 (O)	58 (E), 59 (O)	54 (E), 55 (O)	56 (E), 57 (O)			
15	30	31	31	32	60 (E), 61 (O)	62 (E), 63 (O)	58 (E), 59 (O)	60 (E), 61 (O)			

**Legend:** (E) = Even transaction buffer, (O) = Odd transaction buffer

TABLE 17-5: SUMMARY OF USB BUFFER DESCRIPTOR TABLE REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BDnSTAT <sup>(1)</sup>	UOWN	DTS <sup>(4)</sup>	PID3 <sup>(2)</sup> KEN <sup>(3)</sup>	PID2 <sup>(2)</sup> INCDIS <sup>(3)</sup>	PID1 <sup>(2)</sup> DTSEN <sup>(3)</sup>	PID0 <sup>(2)</sup> BSTALL <sup>(3)</sup>	BC9	BC8
BDnCNT <sup>(1)</sup>	Byte Count	Byte Count						
BDnADRL <sup>(1)</sup>	Buffer Address Low							
BDnADRH <sup>(1)</sup>	Buffer Address High							

- **Note 1:** For buffer descriptor registers, n may have a value of 0 to 63. For the sake of brevity, all 64 registers are shown as one generic prototype. All registers have indeterminate Reset values (xxxx xxxx).
  - 2: Bits 5 through 2 of the BDnSTAT register are used by the SIE to return PID3:PID0 values once the register is turned over to the SIE (UOWN bit is set). Once the registers have been under SIE control, the values written for KEN, INCDIS, DTSEN and BSTALL are no longer valid.
  - **3:** Prior to turning the buffer descriptor over to the SIE (UOWN bit is cleared), bits 5 through 2 of the BDnSTAT register are used to configure the KEN, INCDIS, DTSEN and BSTALL settings.
  - 4: This bit is ignored unless DTSEN = 1.

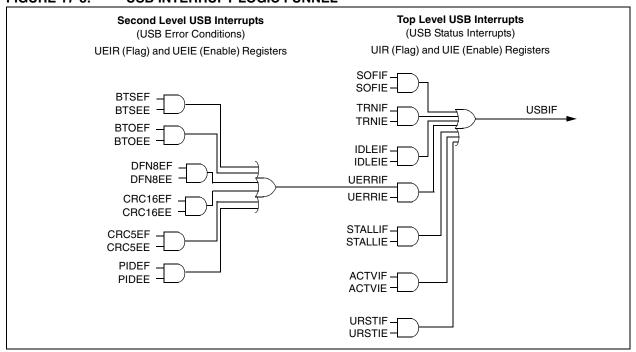
## 17.5 USB Interrupts

The USB module can generate multiple interrupt conditions. To accommodate all of these interrupt sources, the module is provided with its own interrupt logic structure, similar to that of the microcontroller. USB interrupts are enabled with one set of control registers and trapped with a separate set of flag registers. All sources are funneled into a single USB interrupt request, USBIF (PIR2<5>), in the microcontroller's interrupt logic.

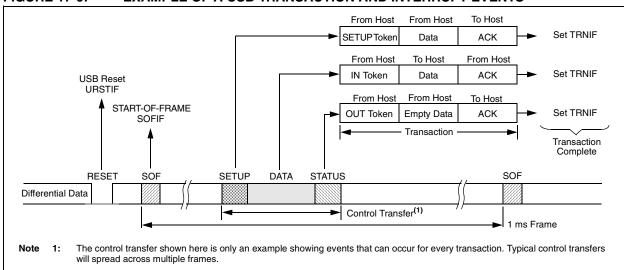
Figure 17-8 shows the interrupt logic for the USB module. There are two layers of interrupt registers in the USB module. The top level consists of overall USB status interrupts; these are enabled and flagged in the UIE and UIR registers, respectively. The second level consists of USB error conditions, which are enabled and flagged in the UEIR and UEIE registers. An interrupt condition in any of these triggers a USB Error Interrupt Flag (UERRIF) in the top level.

Interrupts may be used to trap routine events in a USB transaction. Figure 17-9 shows some common events within a USB frame and their corresponding interrupts.

FIGURE 17-8: USB INTERRUPT LOGIC FUNNEL



### FIGURE 17-9: EXAMPLE OF A USB TRANSACTION AND INTERRUPT EVENTS



## 17.5.1 USB INTERRUPT STATUS REGISTER (UIR)

Legend:

R = Readable bit

The USB Interrupt Status register (Register 17-7) contains the flag bits for each of the USB status interrupt sources. Each of these sources has a corresponding interrupt enable bit in the UIE register. All of the USB status flags are ORed together to generate the USBIF interrupt flag for the microcontroller's interrupt funnel.

Once an interrupt bit has been set by the SIE, it must be cleared by software by writing a '0'. The flag bits can also be set in software which can aid in firmware debugging.

## REGISTER 17-7: UIR: USB INTERRUPT STATUS REGISTER

W = Writable bit

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0
_	SOFIF	STALLIF	IDLEIF <sup>(1)</sup>	TRNIF <sup>(2)</sup>	ACTVIF <sup>(3)</sup>	UERRIF <sup>(4)</sup>	URSTIF
bit 7							bit 0

U = Unimplemented bit, read as '0'

-n = Value at F	POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown					
bit 7	Unimplemented: Read as '0'								
bit 6	SOFIF: START-OF-FRAME Token Interrupt bit								
		T-OF-FRAME token red RT-OF-FRAME token re	,						
bit 5	STALLIF: A	STALL Handshake Inte	errupt bit						
		_ handshake was sent l _ handshake has not be	,						
bit 4	IDLEIF: Idle	IDLEIF: Idle Detect Interrupt bit <sup>(1)</sup>							
		dition detected (constar condition detected	nt Idle state of 3 ms or more)						
bit 3	TRNIF: Tran	saction Complete Interr	rupt bit <sup>(2)</sup>						
			ion is complete; read USTAT region is not complete or no transac						
bit 2	ACTVIF: Bus	Activity Detect Interru	pt bit <sup>(3)</sup>						
		on the D+/D- lines was ity detected on the D+/							
bit 1	<b>UERRIF:</b> US	B Error Condition Inter	rupt bit <sup>(4)</sup>						
		asked error condition hasked error condition h							
bit 0	URSTIF: US	B Reset Interrupt bit							
		B Reset occurred; 00h Reset has occurred	is loaded into UADDR register						

- Note 1: Once an Idle state is detected, the user may want to place the USB module in Suspend mode.
  - 2: Clearing this bit will cause the USTAT FIFO to advance (valid only for IN, OUT and SETUP tokens).
  - **3:** This bit is typically unmasked only following the detection of a UIDLE interrupt event.
  - **4:** Only error conditions enabled through the UEIE register will set this bit. This bit is a status bit only and cannot be set or cleared by the user.

# 17.5.1.1 Bus Activity Detect Interrupt Bit (ACTVIF)

The ACTVIF bit cannot be cleared immediately after the USB module wakes up from Suspend or while the USB module is suspended. A few clock cycles are required to synchronize the internal hardware state machine before the ACTVIF bit can be cleared by firmware. Clearing the ACTVIF bit before the internal

hardware is synchronized may not have an effect on the value of ACTVIF. Additionally, if the USB module uses the clock from the 96 MHz PLL source, then after clearing the SUSPND bit, the USB module may not be immediately operational while waiting for the 96 MHz PLL to lock. The application code should clear the ACTVIF flag as shown in Example 17-1.

## **EXAMPLE 17-1: CLEARING ACTVIF BIT (UIR<2>)**

```
Assembly:
    BCF
            UCON, SUSPND
LOOP:
   BTFSS
           UIR, ACTVIF
   BRA
            DONE
   BCF
            UIR, ACTVIF
    BRA
            LOOP
DONE:
C:
UCONbits.SUSPND = 0;
while (UIRbits.ACTVIF) { UIRbits.ACTVIF = 0; }
```

# 17.5.2 USB INTERRUPT ENABLE REGISTER (UIE)

Legend:

R = Readable bit

-n = Value at POR

The USB Interrupt Enable register (Register 17-8) contains the enable bits for the USB status interrupt sources. Setting any of these bits will enable the respective interrupt source in the UIR register.

The values in this register only affect the propagation of an interrupt condition to the microcontroller's interrupt logic. The flag bits are still set by their interrupt conditions, allowing them to be polled and serviced without actually generating an interrupt.

x = Bit is unknown

U = Unimplemented bit, read as '0'

'0' = Bit is cleared

## REGISTER 17-8: UIE: USB INTERRUPT ENABLE REGISTER

W = Writable bit

'1' = Bit is set

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	SOFIE	STALLIE	IDLEIE	TRNIE	ACTVIE	UERRIE	URSTIE
bit 7							bit 0

II = Value at I	011 1 - 511 13 301	0 = Dit is ci
bit 7	Unimplemented: Read as '0'	
bit 6	SOFIE: START-OF-FRAME Token Interrup	ot Enable bit
	1 = START-OF-FRAME token interrupt en 0 = START-OF-FRAME token interrupt dis	
bit 5	STALLIE: STALL Handshake Interrupt En	able bit
	<ul><li>1 = STALL interrupt enabled</li><li>0 = STALL interrupt disabled</li></ul>	
bit 4	IDLEIE: Idle Detect Interrupt Enable bit	
	<ul><li>1 = Idle detect interrupt enabled</li><li>0 = Idle detect interrupt disabled</li></ul>	
bit 3	TRNIE: Transaction Complete Interrupt Er	nable bit
	1 = Transaction interrupt enabled	
	0 = Transaction interrupt disabled	
bit 2	ACTVIE: Bus Activity Detect Interrupt Ena	ble bit
	<ul><li>1 = Bus activity detect interrupt enabled</li><li>0 = Bus activity detect interrupt disabled</li></ul>	
bit 1	<b>UERRIE:</b> USB Error Interrupt Enable bit	
	<ul><li>1 = USB error interrupt enabled</li><li>0 = USB error interrupt disabled</li></ul>	
bit 0	URSTIE: USB Reset Interrupt Enable bit	
	1 = USB Reset interrupt enabled	
	0 = USB Reset interrupt disabled	

# 17.5.3 USB ERROR INTERRUPT STATUS REGISTER (UEIR)

The USB Error Interrupt Status register (Register 17-9) contains the flag bits for each of the error sources within the USB peripheral. Each of these sources is controlled by a corresponding interrupt enable bit in the UEIE register. All of the USB error flags are ORed together to generate the USB Error Interrupt Flag (UERRIF) at the top level of the interrupt logic.

Each error bit is set as soon as the error condition is detected. Thus, the interrupt will typically not correspond with the end of a token being processed.

Once an interrupt bit has been set by the SIE, it must be cleared by software by writing a '0'.

## REGISTER 17-9: UEIR: USB ERROR INTERRUPT STATUS REGISTER

R/C-0	U-0	U-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
BTSEF	_	_	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF
bit 7							bit 0

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

bit 7 BTSEF: Bit Stuff Error Flag bit

1 = A bit stuff error has been detected

0 = No bit stuff error

bit 6-5 Unimplemented: Read as '0'

bit 4 BTOEF: Bus Turnaround Time-out Error Flag bit

1 = Bus turnaround time-out has occurred (more than 16 bit times of Idle from previous EOP elapsed)

0 = No bus turnaround time-out

bit 3 **DFN8EF:** Data Field Size Error Flag bit

1 = The data field was not an integral number of bytes0 = The data field was an integral number of bytes

bit 2 CRC16EF: CRC16 Failure Flag bit

1 = The CRC16 failed0 = The CRC16 passed

bit 1 CRC5EF: CRC5 Host Error Flag bit

bit 0

1 = The token packet was rejected due to a CRC5 error

0 = The token packet was accepted

PIDEF: PID Check Failure Flag bit

1 = PID check failed0 = PID check passed

# 17.5.4 USB ERROR INTERRUPT ENABLE REGISTER (UEIE)

The USB Error Interrupt Enable register (Register 17-10) contains the enable bits for each of the USB error interrupt sources. Setting any of these bits will enable the respective error interrupt source in the UEIR register to propagate into the UERR bit at the top level of the interrupt logic.

Legend:

R = Readable bit

-n = Value at POR

As with the UIE register, the enable bits only affect the propagation of an interrupt condition to the microcontroller's interrupt logic. The flag bits are still set by their interrupt conditions, allowing them to be polled and serviced without actually generating an interrupt.

x = Bit is unknown

## REGISTER 17-10: UEIE: USB ERROR INTERRUPT ENABLE REGISTER

W = Writable bit

'1' = Bit is set

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BTSEE	_	_	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE
bit 7							bit 0

U = Unimplemented bit, read as '0'

'0' = Bit is cleared

bit 7	BTSEE: Bit Stuff Error Interrupt Enable bit
	1 = Bit stuff error interrupt enabled
	0 = Bit stuff error interrupt disabled
bit 6-5	Unimplemented: Read as '0'
bit 4	<b>BTOEE</b> : Bus Turnaround Time-out Error Interrupt Enable bit
	<ul><li>1 = Bus turnaround time-out error interrupt enabled</li><li>0 = Bus turnaround time-out error interrupt disabled</li></ul>
bit 3	<b>DFN8EE:</b> Data Field Size Error Interrupt Enable bit
	1 = Data field size error interrupt enabled
	0 = Data field size error interrupt disabled
bit 2	CRC16EE: CRC16 Failure Interrupt Enable bit
	1 = CRC16 failure interrupt enabled
	0 = CRC16 failure interrupt disabled
bit 1	CRC5EE: CRC5 Host Error Interrupt Enable bit
	1 = CRC5 host error interrupt enabled
	0 = CRC5 host error interrupt disabled
bit 0	PIDEE: PID Check Failure Interrupt Enable bit
	1 = PID check failure interrupt enabled
	0 = PID check failure interrupt disabled

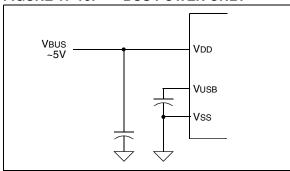
### 17.6 USB Power Modes

Many USB applications will likely have several different sets of power requirements and configuration. The most common power modes encountered are Bus Power Only, Self-Power Only and Dual Power with Self-Power Dominance. The most common cases are presented here.

## 17.6.1 BUS POWER ONLY

In Bus Power Only mode, all power for the application is drawn from the USB (Figure 17-10). This is effectively the simplest power method for the device.

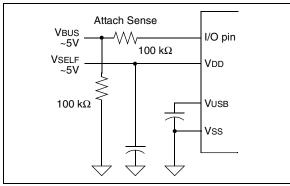
FIGURE 17-10: BUS POWER ONLY



## 17.6.2 SELF-POWER ONLY

In Self-Power Only mode, the USB application provides its own power, with very little power being pulled from the USB. Figure 17-11 shows an example. Note that an attach indication is added to indicate when the USB has been connected.

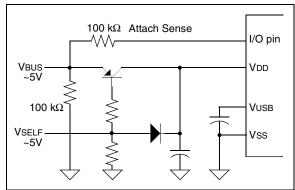
FIGURE 17-11: SELF-POWER ONLY



# 17.6.3 DUAL POWER WITH SELF-POWER DOMINANCE

Some applications may require a dual power option. This allows the application to use internal power primarily, but switch to power from the USB when no internal power is available. Figure 17-12 shows a simple Dual Power with Self-Power Dominance example, which automatically switches between Self-Power Only and USB Bus Power Only modes.

FIGURE 17-12: DUAL POWER EXAMPLE



Note: Users should keep in mind the limits for devices drawing power from the USB.

According to USB Specification 2.0, this cannot exceed 100 mA per low-power device or 500 mA per high-power device.

## 17.7 Streaming Parallel Port

The Streaming Parallel Port (SPP) is an alternate route option for data besides USB RAM. Using the SPP, an endpoint can be configured to send data to or receive data directly from external hardware.

This methodology presents design possibilities where the microcontroller acts as a data manager, allowing the SPP to pass large blocks of data without the microcontroller actually processing it. An application example might include a data acquisition system, where data is streamed from an external FIFO through USB to the host computer. In this case, endpoint control is managed by the microcontroller and raw data movement is processed externally.

The SPP is enabled as a USB endpoint port through the associated endpoint buffer descriptor. The endpoint must be enabled as follows:

- 1. Set BDnADRL:BDnADRH to point to FFFFh.
- Set the KEN bit (BDnSTAT<5>) to let SIE keep control of the buffer.
- Set the INCDIS bit (BDnSTAT<4>) to disable automatic address increment.

Refer to Section 18.0 "Streaming Parallel Port" for more information about the SPP.

- Note 1: If an endpoint is configured to use the SPP, the SPP module must also be configured to use the USB module.

  Otherwise, unexpected operation may occur.
  - 2: In addition, if an endpoint is configured to use the SPP, the data transfer type of that endpoint must be isochronous only.

## 17.8 Oscillator

The USB module has specific clock requirements. For full-speed operation, the clock source must be 48 MHz. Even so, the microcontroller core and other peripherals are not required to run at that clock speed or even from the same clock source. Available clocking options are described in detail in **Section 2.3 "Oscillator Settings for USB"**.

## 17.9 USB Firmware and Drivers

Microchip provides a number of application specific resources, such as USB firmware and driver support. Refer to www.microchip.com for the latest firmware and driver support.

TABLE 17-6: REGISTERS ASSOCIATED WITH USB MODULE OPERATION(1)

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Details on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	51
IPR2	OSCFIP	CMIP	USBIP	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	54
PIR2	OSCFIF	CMIF	USBIF	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	54
PIE2	OSCFIE	CMIE	USBIE	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	54
UCON	_	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	_	55
UCFG	UTEYE	UOEMON	_	UPUEN	UTRDIS	FSEN	PPB1	PPB0	55
USTAT		ENDP3	ENDP2	ENDP1	ENDP0	DIR	PPBI	ı	55
UADDR	_	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0	55
UFRML	FRM7	FRM6	FRM5	FRM4	FRM3	FRM2	FRM1	FRM0	55
UFRMH	_	-	_	_	_	FRM10	FRM9	FRM8	55
UIR	_	SOFIF	STALLIF	IDLEIF	TRNIF	ACTVIF	UERRIF	URSTIF	55
UIE	_	SOFIE	STALLIE	IDLEIE	TRNIE	ACTVIE	UERRIE	URSTIE	55
UEIR	BTSEF	-	-	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF	55
UEIE	BTSEE	_	_	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE	55
UEP0	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	55
UEP1	_	-	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	55
UEP2	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	55
UEP3	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	55
UEP4	_	-	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	55
UEP5	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	55
UEP6	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	55
UEP7	_	-	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	55
UEP8	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	55
UEP9	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	55
UEP10	_	-	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	55
UEP11		_		EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	55
UEP12	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	55
UEP13		_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	55
UEP14	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	55
UEP15	_	_	_	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	55

Legend:

— = unimplemented, read as '0'. Shaded cells are not used by the USB module.

**Note 1:** This table includes only those hardware mapped SFRs located in Bank 15 of the data memory space. The Buffer Descriptor registers, which are mapped into Bank 4 and are not true SFRs, are listed separately in Table 17-5.

### 17.10 Overview of USB

This section presents some of the basic USB concepts and useful information necessary to design a USB device. Although much information is provided in this section, there is a plethora of information provided within the USB specifications and class specifications. Thus, the reader is encouraged to refer to the USB specifications for more information (www.usb.org). If you are very familiar with the details of USB, then this section serves as a basic, high-level refresher of USB.

### 17.10.1 LAYERED FRAMEWORK

USB device functionality is structured into a layered framework graphically shown in Figure 17-13. Each level is associated with a functional level within the device. The highest layer, other than the device, is the configuration. A device may have multiple configurations. For example, a particular device may have multiple power requirements based on Self-Power Only or Bus Power Only modes.

For each configuration, there may be multiple interfaces. Each interface could support a particular mode of that configuration.

Below the interface is the endpoint(s). Data is directly moved at this level. There can be as many as 16 bidirectional endpoints. Endpoint 0 is always a control endpoint and by default, when the device is on the bus, Endpoint 0 must be available to configure the device.

## 17.10.2 FRAMES

Information communicated on the bus is grouped into 1 ms time slots, referred to as frames. Each frame can contain many transactions to various devices and endpoints. Figure 17-9 shows an example of a transaction within a frame.

### 17.10.3 TRANSFERS

There are four transfer types defined in the USB specification.

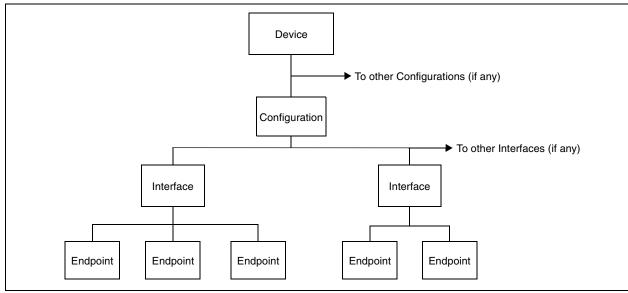
- Isochronous: This type provides a transfer method for large amounts of data (up to 1023 bytes) with timely delivery ensured; however, the data integrity is not ensured. This is good for streaming applications where small data loss is not critical, such as audio.
- Bulk: This type of transfer method allows for large amounts of data to be transferred with ensured data integrity; however, the delivery timeliness is not ensured.
- Interrupt: This type of transfer provides for ensured timely delivery for small blocks of data, plus data integrity is ensured.
- Control: This type provides for device setup control.

While full-speed devices support all transfer types, low-speed devices are limited to interrupt and control transfers only.

## 17.10.4 POWER

Power is available from the Universal Serial Bus. The USB specification defines the bus power requirements. Devices may either be self-powered or bus powered. Self-powered devices draw power from an external source, while bus powered devices use power supplied from the bus.





The USB specification limits the power taken from the bus. Each device is ensured 100 mA at approximately 5V (one unit load). Additional power may be requested, up to a maximum of 500 mA. Note that power above one unit load is a request and the host or hub is not obligated to provide the extra current. Thus, a device capable of consuming more than one unit load must be able to maintain a low-power configuration of a one unit load or less, if necessary.

The USB specification also defines a Suspend mode. In this situation, current must be limited to  $500~\mu\text{A}$ , averaged over 1 second. A device must enter a Suspend state after 3 ms of inactivity (i.e., no SOF tokens for 3 ms). A device entering Suspend mode must drop current consumption within 10 ms after Suspend. Likewise, when signaling a wake-up, the device must signal a wake-up within 10 ms of drawing current above the Suspend limit.

### 17.10.5 ENUMERATION

When the device is initially attached to the bus, the host enters an enumeration process in an attempt to identify the device. Essentially, the host interrogates the device, gathering information such as power consumption, data rates and sizes, protocol and other descriptive information; descriptors contain this information. A typical enumeration process would be as follows:

- USB Reset: Reset the device. Thus, the device is not configured and does not have an address (address 0).
- 2. Get Device Descriptor: The host requests a small portion of the device descriptor.
- 3. USB Reset: Reset the device again.
- Set Address: The host assigns an address to the device.
- Get Device Descriptor: The host retrieves the device descriptor, gathering info such as manufacturer, type of device, maximum control packet size.
- 6. Get configuration descriptors.
- 7. Get any other descriptors.
- 8. Set a configuration.

The exact enumeration process depends on the host.

## 17.10.6 DESCRIPTORS

There are eight different standard descriptor types of which five are most important for this device.

## 17.10.6.1 Device Descriptor

The device descriptor provides general information, such as manufacturer, product number, serial number, the class of the device and the number of configurations. There is only one device descriptor.

## 17.10.6.2 Configuration Descriptor

The configuration descriptor provides information on the power requirements of the device and how many different interfaces are supported when in this configuration. There may be more than one configuration for a device (i.e., low-power and high-power configurations).

## 17.10.6.3 Interface Descriptor

The interface descriptor details the number of endpoints used in this interface, as well as the class of the interface. There may be more than one interface for a configuration.

## 17.10.6.4 Endpoint Descriptor

The endpoint descriptor identifies the transfer type (Section 17.10.3 "Transfers") and direction, as well as some other specifics for the endpoint. There may be many endpoints in a device and endpoints may be shared in different configurations.

## 17.10.6.5 String Descriptor

Many of the previous descriptors reference one or more string descriptors. String descriptors provide human readable information about the layer (Section 17.10.1 "Layered Framework") they describe. Often these strings show up in the host to help the user identify the device. String descriptors are generally optional to save memory and are encoded in a unicode format.

## 17.10.7 BUS SPEED

Each USB device must indicate its bus presence and speed to the host. This is accomplished through a 1.5 k $\Omega$  resistor which is connected to the bus at the time of the attachment event.

Depending on the speed of the device, the resistor either pulls up the D+ or D- line to 3.3V. For a low-speed device, the pull-up resistor is connected to the D- line. For a full-speed device, the pull-up resistor is connected to the D+ line.

# 17.10.8 CLASS SPECIFICATIONS AND DRIVERS

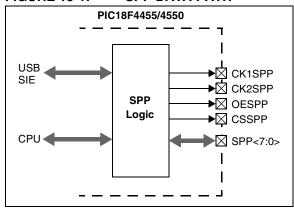
USB specifications include class specifications which operating system vendors optionally support. Examples of classes include Audio, Mass Storage, Communications and Human Interface (HID). In most cases, a driver is required at the host side to 'talk' to the USB device. In custom applications, a driver may need to be developed. Fortunately, drivers are available for most common host systems for the most common classes of devices. Thus, these drivers can be reused.

## 18.0 STREAMING PARALLEL PORT

**Note:** The Streaming Parallel Port is only available on 40/44-pin devices.

PIC18F4455/4550 USB devices provide a Streaming Parallel Port as a high-speed interface for moving data to and from an external system. This parallel port operates as a master port, complete with chip select and clock outputs to control the movement of data to slave devices. Data can be channelled either directly to the USB SIE or to the microprocessor core. Figure 18-1 shows a block view of the SPP data path.

FIGURE 18-1: SPP DATA PATH



In addition, the SPP can provide time multiplexed addressing information along with the data by using the second strobe output. Thus, the USB endpoint number can be written in conjunction with the data for that endpoint.

## 18.1 SPP Configuration

The operation of the SPP is controlled by two registers: SPPCON and SPPCFG. The SPPCON register (Register 18-1) controls the overall operation of the parallel port and determines if it operates under USB or microcontroller control. The SPPCFG register (Register 18-2) controls timing configuration and pin outputs.

## 18.1.1 ENABLING THE SPP

To enable the SPP, set the SPPEN bit (SPPCON<0>). In addition, the TRIS bits for the corresponding SPP pins must be properly configured. At a minimum:

- Bits TRISD<7:0> must be set (= 1)
- Bits TRISE<2:1> must be cleared (= 0)

If CK1SPP is to be used:

• Bit TRISE<0> must be cleared (= 0)

If CSPP is to be used:

• Bit TRISB<4> must be cleared (= 0)

## REGISTER 18-1: SPPCON: SPP CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
_	_	_	_	_	_	SPPOWN	SPPEN
bit 7							bit 0

bit 7-2 **Unimplemented:** Read as '0' bit 1 **SPPOWN:** SPP Ownership bit

1 = USB peripheral controls the SPP

0 = Microcontroller directly controls the SPP

bit 0 SPPEN: SPP Enable bit

1 = SPP is enabled 0 = SPP is disabled

#### REGISTER 18-2: SPPCFG: SPP CONFIGURATION 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
CLKCFG1	CLKCFG0	CSEN	CLK1EN	WS3	WS2	WS1	WS0
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 CLKCFG1:CLKCFG0: SPP Clock Configuration bits

1x = CLK1 toggles on read or write of an Odd endpoint address; CLK2 toggles on read or write of an Even endpoint address

01 = CLK1 toggles on write; CLK2 toggles on read

00 = CLK1 toggles only on endpoint address write; CLK2 toggles on data read or write

bit 5 CSEN: SPP Chip Select Pin Enable bit

1 = RB4 pin is controlled by the SPP module and functions as SPP CS output

0 = RB4 functions as a digital I/O port

bit 4 CLK1EN: SPP CLK1 Pin Enable bit

1 = RE0 pin is controlled by the SPP module and functions as SPP CLK1 output

0 = RE0 functions as a digital I/O port

bit 3-0 WS3:WS0: SPP Wait States bits

1111 = 30 additional wait states 1110 = 28 additional wait states

•

0001 = 2 additional wait states 0000 = 0 additional wait states

## 18.1.2 CLOCKING DATA

The SPP has four control outputs:

- Two separate clock outputs (CK1SPP and CK2SPP)
- Output enable (OESPP)
- Chip select (CSSPP)

Together, they allow for several different configurations for controlling the flow of data to slave devices. When all control outputs are used, the three main options are:

- CLK1 clocks endpoint address information while CLK2 clocks data
- CLK1 clocks write operations while CLK2 clocks reads
- CLK1 clocks Odd address data while CLK2 clocks Even address data

Additional control options are derived by disabling the CK1SPP and CSSPP outputs. These are enabled or disabled with the CLK1EN and CSEN bits, respectively, located in Register 18-2.

## 18.1.3 WAIT STATES

The SPP is designed with the capability of adding wait states to read and write operations. This allows access to parallel devices that require extra time for access.

Wait state clocking is based on the data source clock. If the SPP is configured to operate as a USB endpoint, then wait states are based on the USB clock. Likewise, if the SPP is configured to operate from the microcontroller, then wait states are based on the instruction rate (Fosc/4).

The WS3:WS0 bits set the wait states used by the SPP, with a range of no wait states to 30 wait states, in multiples of two. The wait states are added symmetrically to all transactions, with one-half added following each of the two clock cycles normally required for the transaction. Figure 18-3 and Figure 18-4 show signalling examples with 4 wait states added to each transaction.

## 18.1.4 SPP PULL-UPS

The SPP data lines (SPP<7:0>) are equipped with internal pull-ups for applications that may leave the port in a high-impedance condition. The pull-ups are enabled using the control bit, RDPU (PORTE<7>).

FIGURE 18-2: TIMING FOR MICROCONTROLLER WRITE ADDRESS, WRITE DATA AND READ DATA (NO WAIT STATES)

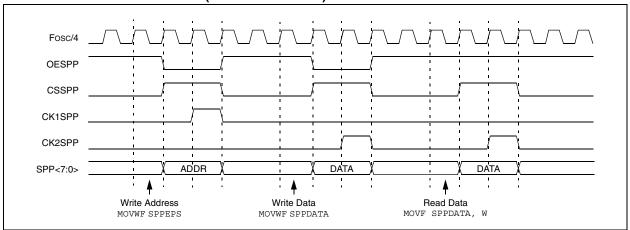


FIGURE 18-3: TIMING FOR USB WRITE ADDRESS AND DATA (4 WAIT STATES)

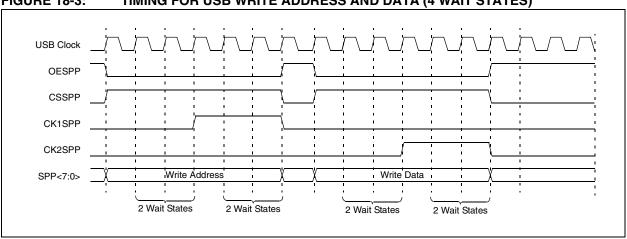
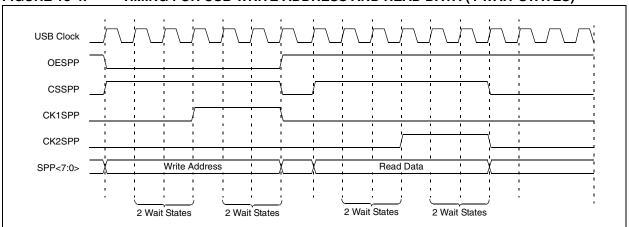


FIGURE 18-4: TIMING FOR USB WRITE ADDRESS AND READ DATA (4 WAIT STATES)



© 2006 Microchip Technology Inc. Preliminary DS39632C-page 189

## 18.2 Setup for USB Control

When the SPP is configured for USB operation, data can be clocked directly to and from the USB peripheral without intervention of the microcontroller; thus, no process time is required. Data is clocked into or out from the SPP with endpoint (address) information first, followed by one or more bytes of data, as shown in Figure 18-5. This is ideal for applications that require isochronous, large volume data movement.

The following steps are required to set up the SPP for USB control:

- Configure the SPP as desired, including wait states and clocks.
- 2. Set the SPPOWN bit for USB ownership.
- 3. Set the buffer descriptor starting address (BDnADRL:BDnADRH) to FFFFh.
- 4. Set the KEN bit (BDnSTAT<5>) so the buffer descriptor is kept indefinitely by the SIE.
- Set the INCDIS bit (BDnSTAT<4>) to disable automatic buffer address increment.
- Set the SPPEN bit to enable the module.

**Note:** If a USB endpoint is configured to use the SPP, the data transfer type of that endpoint must be isochronous only.

## **18.3** Setup for Microcontroller Control

The SPP can also act as a parallel port for the microcontroller. In this mode, the SPPEPS register (Register 18-3) provides status and address write control. Data is written to and read from the SPPDATA register. When the SPP is owned by the microcontroller, the SPP clock is driven by the instruction clock (FOSC/4).

The following steps are required to set up the SPP for microcontroller operation:

- Configure the SPP as desired, including wait states and clocks.
- 2. Clear the SPPOWN bit.
- 3. Set SPPEN to enable the module.

#### 18.3.1 SPP INTERRUPTS

When owned by the microcontroller core, control can generate an interrupt to notify the application when each read and write operation is completed. The interrupt flag bit is SPPIF (PIR1<7>) and is enabled by the SPPIE bit (PIE1<7>). Like all other microcontroller level interrupts, it can be set to a low or high priority. This is done with the SPPIP bit (IPR1<7>).

### 18.3.2 WRITING TO THE SPP

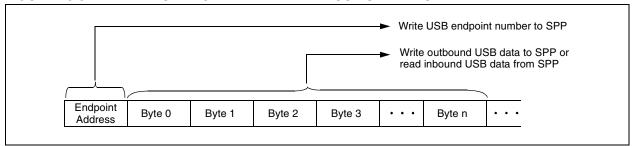
Once configured, writing to the SPP is performed by writing to the SPPEPS and SPPDATA registers. If the SPP is configured to clock out endpoint address information with the data, writing to the SPPEPS register initiates the address write cycle. Otherwise, the write is started by writing the data to the SPPDATA register. The SPPBUSY bit indicates the status of the address and the data write cycles.

The following is an example write sequence:

- Write the 4-bit address to the SPPEPS register.
   The SPP automatically starts writing the address. If address write is not used, then skip to step 3.
- Monitor the SPPBUSY bit to determine when the address has been sent. The duration depends on the wait states.
- 3. Write the data to the SPPDATA register. The SPP automatically starts writing the data.
- Monitor the SPPBUSY bit to determine when the data has been sent. The duration depends on the wait states.
- Go back to steps 1 or 3 to write a new address or data.

Note: The SPPBUSY bit should be polled to make certain that successive writes to the SPPEPS or SPPDATA registers do not overrun the wait time due to the wait state setting.

## FIGURE 18-5: TRANSFER OF DATA BETWEEN USB SIE AND SPP



#### 18.3.3 READING FROM THE SPP

Reading from the SPP involves reading the SPPDATA register. Reading the register the first time initiates the read operation. When the read is finished, indicated by the SPPBUSY bit, the SPPDATA will be loaded with the current data.

The following is an example read sequence:

- Write the 4-bit address to the SPPEPS register.
   The SPP automatically starts writing the address. If address write is not used then skip to step 3.
- Monitor the SPPBUSY bit to determine when the address has been sent. The duration depends on the wait states.

- Read the data from the SPPDATA register; the data from the previous read operation is returned. The SPP automatically starts the read cycle for the next read.
- 4. Monitor the SPPBUSY bit to determine when the data has been read. The duration depends on the wait states.
- 5. Go back to step 3 to read the current byte from the SPP and start the next read cycle.

## REGISTER 18-3: SPPEPS: SPP ENDPOINT ADDRESS AND STATUS REGISTER

R-0	R-0	U-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
RDSPP	WRSPP	_	SPPBUSY	ADDR3	ADDR2	ADDR1	ADDR0
bit 7							bit 0

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

bit 7 RDSPP: SPP Read Status bit (Valid when SPPCON<SPPOWN> = 1, USB)

1 = The last transaction was a read from the SPP

0 = The last transaction was not a read from the SPP

bit 6 WRSPP: SPP Write Status bit (Valid when SPPCON<SPPOWN> = 1, USB)

1 = The last transaction was a write to the SPP

0 = The last transaction was not a write to the SPP

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

bit 4 SPPBUSY: SPP Handshaking Override bit

1 = The SPP is busy

0 = The SPP is ready to accept another read or write request

bit 3-0 ADDR3:ADDR0: SPP Endpoint Address bits

1111 = Endpoint Address 15

•

0001

0000 = Endpoint Address 0

TABLE 18-1: REGISTERS ASSOCIATED WITH THE STREAMING PARALLEL PORT

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
SPPCON <sup>(3)</sup>	_	_		_		_	SPPOWN	SPPEN	55
SPPCFG <sup>(3)</sup>	CLKCFG1	CLKCFG0	CSEN	CLK1EN	WS3	WS2	WS1	WS0	55
SPPEPS <sup>(3)</sup>	RDSPP	WRSPP	_	SPPBUSY	ADDR3	ADDR2	ADDR1	ADDR0	55
SPPDATA <sup>(3)</sup>	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	55
PIR1	SPPIF <sup>(3)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54
PIE1	SPPIE <sup>(3)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54
IPR1	SPPIP <sup>(3)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54
PORTE	RDPU <sup>(3)</sup>	_	_	_	RE3 <sup>(1,2)</sup>	RE2 <sup>(3)</sup>	RE1 <sup>(3)</sup>	RE0 <sup>(3)</sup>	54

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used for the Streaming Parallel Port.

**Note 1:** Implemented only when Master Clear functionality is disabled (MCLRE Configuration bit = 0).

- 2: RE3 is the only PORTE bit implemented on both 28-pin and 40/44-pin devices. All other bits are implemented only when PORTE is implemented (i.e., 40/44-pin devices).
- 3: These registers and/or bits are unimplemented on 28-pin devices.

## 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<sup>2</sup>C<sup>™</sup>)
  - Full Master mode
  - Slave mode (with general address call)

The I<sup>2</sup>C interface supports the following modes in hardware:

- · Master mode
- · Multi-Master mode
- Slave mode

## 19.2 Control Registers

The MSSP module has three associated control registers. These include a status register (SSPSTAT) and two control registers (SSPCON1 and SSPCON2). The use of these registers and their individual Configuration bits differ significantly depending on whether the MSSP module is operated in SPI or I<sup>2</sup>C mode.

Additional details are provided under the individual sections.

### 19.3 SPI Mode

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

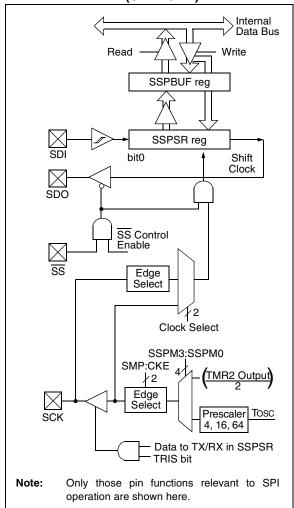
- Serial Data Out (SDO) RC7/RX/DT/SDO
- Serial Data In (SDI) RB0/AN12/INT0/FLT0/SDI/SDA
- Serial Clock (SCK) RB1/AN10/INT1/SCK/SCL

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

• Slave Select (SS) - RA5/AN4/SS/HLVDIN/C2OUT

Figure 19-1 shows the block diagram of the MSSP module when operating in SPI mode.

FIGURE 19-1: MSSP BLOCK DIAGRAM (SPI MODE)



#### 19.3.1 REGISTERS

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

- MSSP Control Register 1 (SSPCON1)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer Register (SSPBUF)
- MSSP Shift Register (SSPSR) Not directly accessible

SSPCON1 and SSPSTAT are the control and status registers in SPI mode operation. The SSPCON1 register is readable and writable. The lower six bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

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

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

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

## REGISTER 19-1: SSPSTAT: MSSP STATUS REGISTER (SPI MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE <sup>(1)</sup>	D/A	Р	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

SPI Master mode:

1 = Input data sampled at end of data output time0 = Input data sampled at middle of data output time

SPI Slave mode:

SMP must be cleared when SPI is used in Slave mode.

bit 6 CKE: SPI Clock Select bit<sup>(1)</sup>

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<sup>2</sup>C mode only.

bit 4 **P:** Stop bit

Used in I<sup>2</sup>C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.

bit 3 S: Start bit

Used in I<sup>2</sup>C mode only.

bit 2 R/W: Read/Write Information bit

Used in I<sup>2</sup>C mode only.

bit 1 **UA:** Update Address bit

Used in I<sup>2</sup>C mode only.

bit 0 **BF:** Buffer Full Status bit (Receive mode only)

1 = Receive complete, SSPBUF is full

0 = Receive not complete, SSPBUF is empty

Note 1: Polarity of clock state is set by the CKP bit (SSPCON1<4>).

## REGISTER 19-2: SSPCON1: MSSP 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 <sup>(1)</sup>	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
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 (Transmit mode only)

1 = The SSPBUF register is written while it is still transmitting the previous word

(must be cleared in software) 0 = No collision

bit 6 SSPOV: Receive Overflow Indicator bit<sup>(1)</sup>

SPI Slave mode:

1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow (must be cleared in software).

0 = No overflow

bit 5 SSPEN: Master Synchronous Serial Port Enable bit

1 = Enables serial port and configures SCK, SDO, SDI and  $\overline{SS}$  as serial port pins<sup>(2)</sup>

0 = Disables serial port and configures these pins as I/O port pins(2)

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

0101 = SPI Slave mode, clock = SCK pin,  $\overline{SS}$  pin control disabled,  $\overline{SS}$  can be used as I/O pin<sup>(3)</sup>

0100 = SPI Slave mode, clock = SCK pin,  $\overline{SS}$  pin control enabled<sup>(3)</sup>

0011 = SPI Master mode, clock = TMR2 output/2(3)

0010 = SPI Master mode, clock = Fosc/64<sup>(3)</sup>

0001 = SPI Master mode, clock = Fosc/16<sup>(3)</sup>

0000 = SPI Master mode, clock = Fosc/4<sup>(3)</sup>

**Note 1:** In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF 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 I<sup>2</sup>C™ mode only.

#### 19.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON1<5:0> and SSPSTAT<7:6>). These control bits allow the following to be specified:

- · Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- · Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

The MSSP module consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the eight bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full detect bit, BF (SSPSTAT<0>) and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before

reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the Write Collision detect bit, WCOL (SSPCON1<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

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

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

## **EXAMPLE 19-1: LOADING THE SSPBUF (SSPSR) REGISTER**

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

#### 19.3.3 ENABLING SPI I/O

To enable the serial port, MSSP Enable bit, SSPEN (SSPCON1<5>), must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, reinitialize the SSPCON registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and SS pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed as follows:

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<7> bit cleared
- SCK (Master mode) must have TRISB<1> bit cleared
- SCK (Slave mode) must have TRISB<1> bit set
- SS must have TRISA<5> bit set

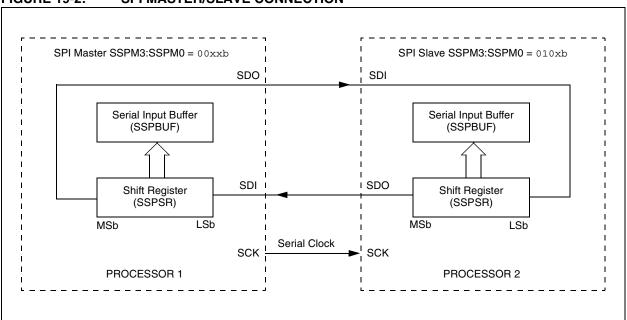
Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

## 19.3.4 TYPICAL CONNECTION

Figure 19-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

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





© 2006 Microchip Technology Inc. Preliminary DS39632C-page 197

#### 19.3.5 MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2, Figure 19-2) is to broadcast data by the software protocol.

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

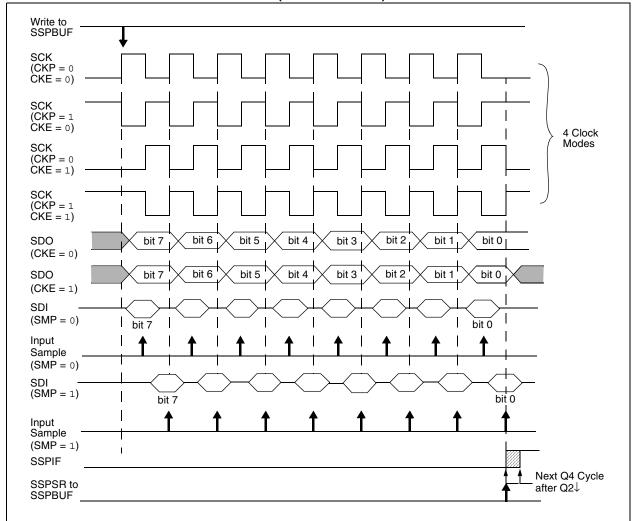
The clock polarity is selected by appropriately programming the CKP bit (SSPCON1<4>). This then, would give waveforms for SPI communication as shown in Figure 19-3, Figure 19-5 and Figure 19-6, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- Fosc/4 (or Tcy)
- Fosc/16 (or 4 Tcy)
- Fosc/64 (or 16 Tcy)
- Timer2 output/2

This allows a maximum data rate (at 48 MHz) of 2.00 Mbps.

Figure 19-3 shows the waveforms for Master mode. When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPBUF is loaded with the received data is shown.





## 19.3.6 SLAVE MODE

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

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

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

# 19.3.7 SLAVE SELECT SYNCHRONIZATION

The  $\overline{SS}$  pin allows a Synchronous Slave mode. The SPI must be in Slave mode with the  $\overline{SS}$  pin control enabled (SSPCON1<3:0> = 04h). When the  $\overline{SS}$  pin is low, transmission and reception are enabled and the SDO pin is driven. When the  $\overline{SS}$  pin goes high, the SDO pin is no longer driven, even if in the middle of a

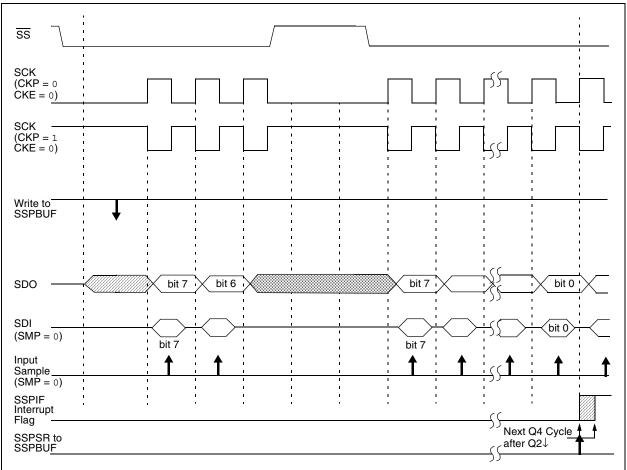
transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

- Note 1: When the SPI module is in Slave mode with  $\overline{SS}$  pin control enabled (SSPCON1<3:0> = 0100), the SPI module will reset if the  $\overline{SS}$  pin is set to VDD.
  - 2: If the SPI is used in Slave mode with CKE set, then the SS pin control must be enabled.

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

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





© 2006 Microchip Technology Inc. Preliminary DS39632C-page 199

FIGURE 19-5: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

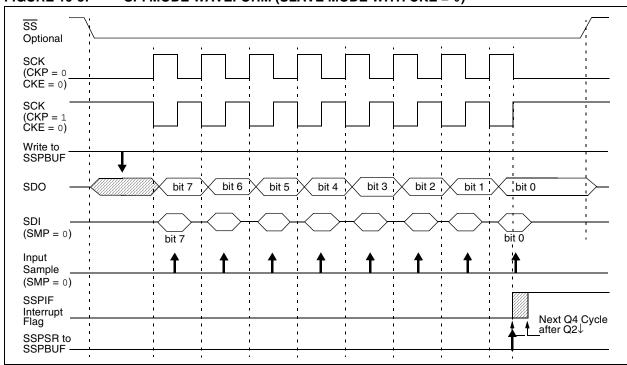
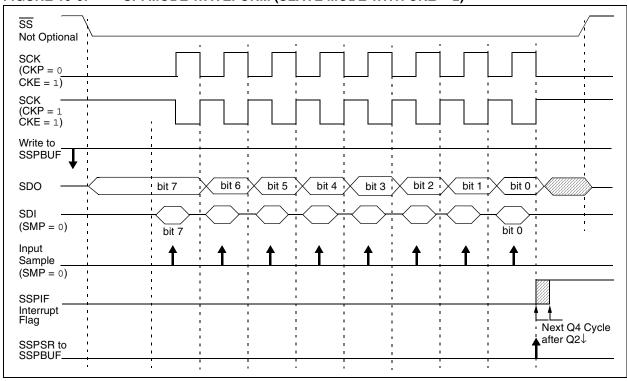


FIGURE 19-6: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



# 19.3.8 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 most idle modes, a clock is provided to the peripherals. That clock should be from the primary clock source, the secondary clock (Timer1 oscillator) or the INTOSC source. See **Section 2.4 "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 devices 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 eight bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device.

#### 19.3.9 EFFECTS OF A RESET

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

### 19.3.10 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	Control Bits State				
Terminology	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.

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	51
PIR1	SPPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54
PIE1	SPPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54
IPR1	SPPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54
TRISA	_	TRISA6 <sup>(2)</sup>	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	54
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	54
TRISC	TRISC7	TRISC6	_	_	_	TRISC2	TRISC1	TRISC0	54
SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	52
SSPSTAT	SMP	CKE	D/ <del>A</del>	Р	S	R/W	UA	BF	52

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by the MSSP in SPI mode.

Note 1: These bits are unimplemented in 28-pin devices; always maintain these bits clear.

2: RA6 is configured as a port pin based on various primary oscillator modes. When the port pin is disabled, all of the associated bits read '0'.

## 19.4 I<sup>2</sup>C Mode

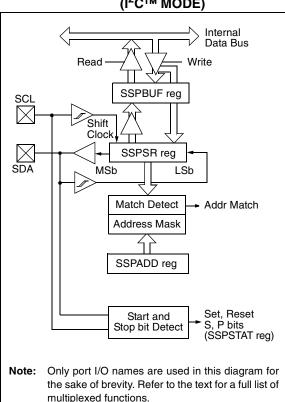
The MSSP module in I<sup>2</sup>C mode fully implements all master and slave functions (including general call support) and provides interrupts on Start and Stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit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

Two pins are used for data transfer:

- Serial clock (SCL) RB1/AN10/INT1/SCK/SCL
- Serial data (SDA) RB0/AN12/INT0/FLT0/SDI/SDA

The user must configure these pins as inputs by setting the associated TRIS bits.

FIGURE 19-7: MSSP BLOCK DIAGRAM (I<sup>2</sup>C™ MODE)



#### 19.4.1 REGISTERS

The MSSP module has six registers for I<sup>2</sup>C operation. These are:

- MSSP Control Register 1 (SSPCON1)
- MSSP Control Register 2 (SSPCON2)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer Register (SSPBUF)
- MSSP Shift Register (SSPSR) Not directly accessible
- MSSP Address Register (SSPADD)

SSPCON1, SSPCON2 and SSPSTAT are the control and status registers in I<sup>2</sup>C mode operation. The SSPCON1 and SSPCON2 registers are readable and writable. The lower six bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

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

SSPADD register holds the slave device address when the MSSP is configured in I<sup>2</sup>C Slave mode. When the MSSP is configured in Master mode, the lower seven bits of SSPADD act as the Baud Rate Generator reload value.

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

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

## REGISTER 19-3: SSPSTAT: MSSP STATUS REGISTER (I<sup>2</sup>C™ MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P <sup>(1)</sup>	S <sup>(1)</sup>	R/W(2,3)	UA	BF
bit 7	•		•	•			bit 0

Legend:

bit 5

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 inputs0 = Disable SMBus specific inputs

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<sup>(1)</sup>

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<sup>(2,3)</sup>

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 SSPADD register

0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit

In Transmit mode:

1 = SSPBUF is full

0 = SSPBUF is empty

In Receive mode:

1 = SSPBUF is full (does not include the  $\overline{ACK}$  and Stop bits)

0 = SSPBUF is empty (does not include the  $\overline{ACK}$  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 MSSP is in Active mode.

DS39632C-page 203

## REGISTER 19-4: SSPCON1: MSSP CONTROL REGISTER 1 (I<sup>2</sup>C™ 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	CKP	SSPM3	SSPM2	SSPM1	SSPM0
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 SSPBUF register was attempted while the  $I^2C$  conditions were not valid for a transmission to be started (must be cleared in software)
- 0 = No collision

### In Slave Transmit mode:

- 1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
- 0 = No collision

## In Receive mode (Master or Slave modes):

This is a "don't care" bit.

## bit 6 SSPOV: Receive Overflow Indicator bit

#### In Receive mode:

- 1 = A byte is received while the SSPBUF register is still holding the previous byte (must be cleared in software)
- 0 = No overflow

## In Transmit mode:

This is a "don't care" bit in Transmit mode.

- bit 5 SSPEN: Master Synchronous Serial Port Enable bit
  - 1 = Enables the serial port and configures the SDA and SCL pins as the serial port pins<sup>(1)</sup>
  - 0 = Disables serial port and configures these pins as I/O port pins<sup>(1)</sup>
- bit 4 CKP: SCK Release Control bit

## In Slave mode:

- 1 = Release clock
- 0 = Holds clock low (clock stretch), used to ensure data setup time

## In Master mode:

Unused in this mode.

### bit 3-0 SSPM3:SSPM0: Master Synchronous Serial Port Mode Select bits

- $1111 = I^2C$  Slave mode, 10-bit address with Start and Stop bit interrupts enabled (2)
- 1110 =  $I^2C$  Slave mode, 7-bit address with Start and Stop bit interrupts enabled<sup>(2)</sup>
- 1011 = I<sup>2</sup>C Firmware Controlled Master mode (slave Idle)<sup>(2)</sup>
- 1000 =  $I^2C$  Master mode, clock = Fosc/ $(4 * (SSPADD + 1))^{(2)}$
- $0111 = I^2C$  Slave mode, 10-bit address<sup>(2)</sup>
- $0110 = I^2C$  Slave mode, 7-bit address<sup>(2)</sup>

## Note 1: When enabled, the SDA and SCL pins must be properly configured as input or output.

2: Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.

## REGISTER 19-5: SSPCON2: MSSP CONTROL REGISTER 2 (I<sup>2</sup>C™ 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 <sup>(1)</sup>	ACKEN <sup>(2)</sup>	RCEN <sup>(2)</sup>	PEN <sup>(2)</sup>	RSEN <sup>(2)</sup>	SEN <sup>(2)</sup>
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 (Slave mode only)

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)<sup>(1)</sup>

1 = Not Acknowledge

0 = Acknowledge

bit 4 ACKEN: Acknowledge Sequence Enable bit<sup>(2)</sup>

1 = Initiate Acknowledge sequence on SDA and SCL pins and transmit ACKDT data bit. Automatically cleared by hardware.

0 = Acknowledge sequence Idle

bit 3 RCEN: Receive Enable bit (Master Receive mode only)(2)

1 = Enables Receive mode for I<sup>2</sup>C

0 = Receive Idle

bit 2 **PEN:** Stop Condition Enable bit<sup>(2)</sup>

1 = Initiate Stop condition on SDA and SCL pins. Automatically cleared by hardware.

0 = Stop condition Idle

bit 1 RSEN: Repeated Start Condition Enable bit<sup>(2)</sup>

1 = Initiate Repeated Start condition on SDA and SCL pins. Automatically cleared by hardware.

0 = Repeated Start condition Idle

bit 0 SEN: Start Condition Enable/Stretch Enable bit(2)

1 = Initiate Start condition on SDA and SCL 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 I<sup>2</sup>C module is active, these bits may not be set (no spooling) and the SSPBUF may not be written (or writes to the SSPBUF are disabled).

## REGISTER 19-6: SSPCON2: MSSP CONTROL REGISTER 2 (I<sup>2</sup>C™ 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 <sup>(1)</sup>
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 (Slave mode only)

1 = Enable interrupt when a general call address (0000h) is received in the SSPSR

0 = General call address disabled

bit 6 ACKSTAT: Acknowledge Status bit

Unused in Slave mode.

bit 5-2 ADMSK5:ADMSK2: Slave Address Mask Select bits

1 = Masking of corresponding bits of SSPADD enabled 0 = Masking of corresponding bits of SSPADD disabled

bit 1 ADMSK1: Slave Address Mask Select bit

In 7-Bit Address mode:

1 = Masking of SPADD<1> only enabled

0 = Masking of SPADD<1> only disabled

In 10-Bit Address mode:

1 = Masking of SSPADD < 1:0 > enabled

0 = Masking of SSPADD<1:0> disabled

bit 0 SEN: Stretch Enable bit<sup>(1)</sup>

1 = Clock stretching is enabled for both slave transmit and slave receive (stretch enabled)

0 = Clock stretching is disabled

**Note 1:** If the I<sup>2</sup>C 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).

#### 19.4.2 OPERATION

The MSSP module functions are enabled by setting MSSP Enable bit, SSPEN (SSPCON1<5>).

The SSPCON1 register allows control of the I<sup>2</sup>C operation. Four mode selection bits (SSPCON1<3:0>) allow one of the following I<sup>2</sup>C modes to be selected:

- I<sup>2</sup>C Master mode, clock
- I<sup>2</sup>C Slave mode (7-bit address)
- I<sup>2</sup>C Slave mode (10-bit address)
- I<sup>2</sup>C Slave mode (7-bit address) with Start and Stop bit interrupts enabled
- I<sup>2</sup>C Slave mode (10-bit address) with Start and Stop bit interrupts enabled
- I<sup>2</sup>C Firmware Controlled Master mode, slave is Idle

Selection of any I<sup>2</sup>C mode with the SSPEN bit set forces the SCL and SDA 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 SCL and SDA pins.

## 19.4.3 SLAVE MODE

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

The I<sup>2</sup>C 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 SSPBUF register with the received value currently in the SSPSR register.

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

- The Buffer Full bit, BF (SSPSTAT<0>), was set before the transfer was received.
- The overflow bit, SSPOV (SSPCON1<6>), was set before the transfer was received.

In this case, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF is set. The BF bit is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the  $I^2C$  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 SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match and the BF and SSPOV bits are clear, the following events occur:

- The SSPSR register value is loaded into the SSPBUF register.
- 2. The Buffer Full bit, BF, is set.
- 3. An ACK pulse is generated.
- The MSSP Interrupt Flag bit, SSPIF, is set (and interrupt is generated, if enabled) on the falling edge of the ninth SCL pulse.

In 10-Bit Address mode, two address bytes need to be received by the slave. The five Most Significant bits (MSbs) of the first address byte specify if this is a 10-bit address. Bit R/W (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '11110 A9 A8 0', where 'A9' and 'A8' are the two MSbs of the address. The sequence of events for 10-bit address is as follows, with steps 7 through 9 for the slave-transmitter:

- Receive first (high) byte of address (bits SSPIF, BF and UA (SSPSTAT<1>) are set on address match).
- 2. Update the SSPADD register with second (low) byte of address (clears bit UA and releases the SCL line).
- Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
- Receive second (low) byte of address (bits SSPIF, BF and UA are set).
- Update the SSPADD register with the first (high) byte of address. If match releases SCL line, this will clear bit UA.
- 6. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
- 7. Receive Repeated Start condition.
- Receive first (high) byte of address (bits SSPIF and BF are set).
- Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

## 19.4.3.2 Address Masking

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 makes it possible to Acknowledge up to 31 addresses in 7-bit mode and up to 63 addresses in 10-bit mode (see Example 19-2).

The I<sup>2</sup>C Slave behaves the same way whether address masking is used or not. However, when address masking is used, the I<sup>2</sup>C slave can Acknowledge multiple addresses and cause interrupts. When this occurs, it is necessary to determine which address caused the interrupt by checking SSPBUF.

In 7-Bit Address mode, address mask bits ADMSK<5:1> (SSPCON2<5:1>) mask the corresponding address bits in the SSPADD register. For any ADMSK bits that are set (ADMSK<n> = 1), the corresponding address bit is ignored (SSPADD<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 mode, bits ADMSK<5:2> mask the corresponding address bits in the SSPADD register. In addition, ADMSK1 simultaneously masks the two LSbs of the address (SSPADD<1:0>). For any ADMSK bits that are active (ADMSK<n> = 1), the corresponding address bit is ignored (SSPADD<n> = x). Also note that although in 10-Bit Addressing mode, the upper address bits reuse part of the SSPADD 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**

## 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 MSbs 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

## 19.4.3.3 Reception

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

When the address byte overflow condition exists, then the no Acknowledge (ACK) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set, or bit SSPOV (SSPCON1<6>) is set.

An MSSP interrupt is generated for each data transfer byte. The Interrupt Flag bit, SSPIF, must be cleared in software. The SSPSTAT register is used to determine the status of the byte.

If SEN is enabled (SSPCON2<0> = 1), RB1/AN10/INT1/SCK/SCL will be held low (clock stretch) following each data transfer. The clock must be released by setting bit, CKP (SSPCON1<4>). See **Section 19.4.4** "Clock Stretching" for more detail.

#### 19.4.3.4 Transmission

When the R/W bit of the incoming address byte is set and an address match occurs, the R/W bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The ACK pulse will be sent on the ninth bit and pin RB1/AN10/INT1/SCK/ SCL is held low regardless of SEN (see Section 19.4.4 "Clock Stretching" for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data. The transmit data must be loaded into the SSPBUF register which also loads the SSPSR register. Then pin RB1/AN10/INT1/SCK/SCL should be enabled by setting bit, CKP (SSPCON1<4>). The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 19-10).

The  $\overline{ACK}$  pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line is high (not  $\overline{ACK}$ ), then the data transfer is complete. In this case, when the  $\overline{ACK}$  is latched by the slave, the slave logic is reset (resets SSPSTAT register) and the slave monitors for another occurrence of the Start bit. If the SDA line was low  $(\overline{ACK})$ , the next transmit data must be loaded into the SSPBUF register. Again, pin RB1/AN10/INT1/SCK/SCL must be enabled by setting bit CKP (SSPCON1<4>).

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

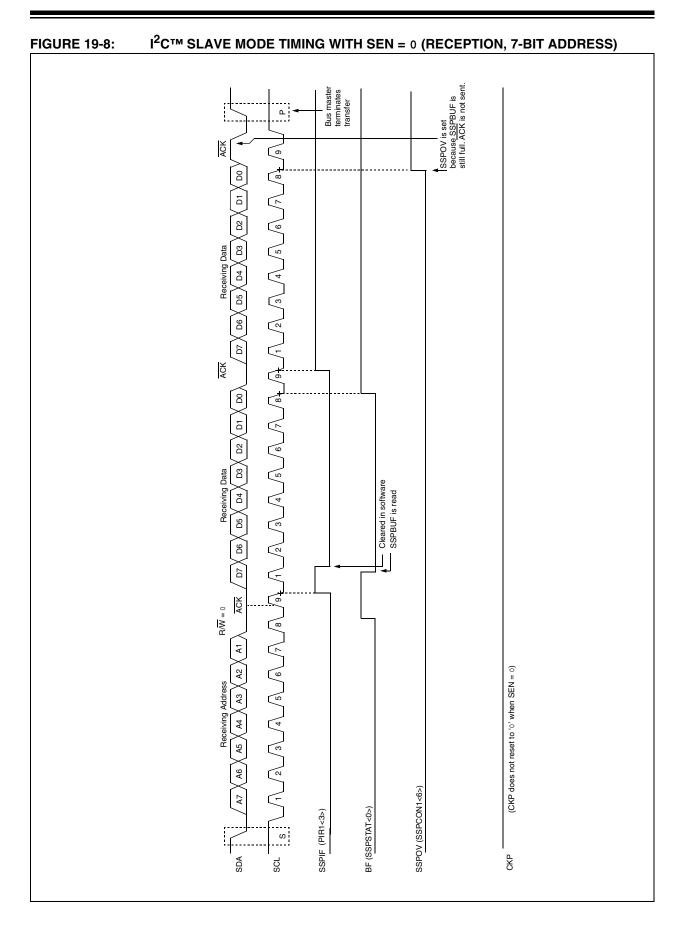
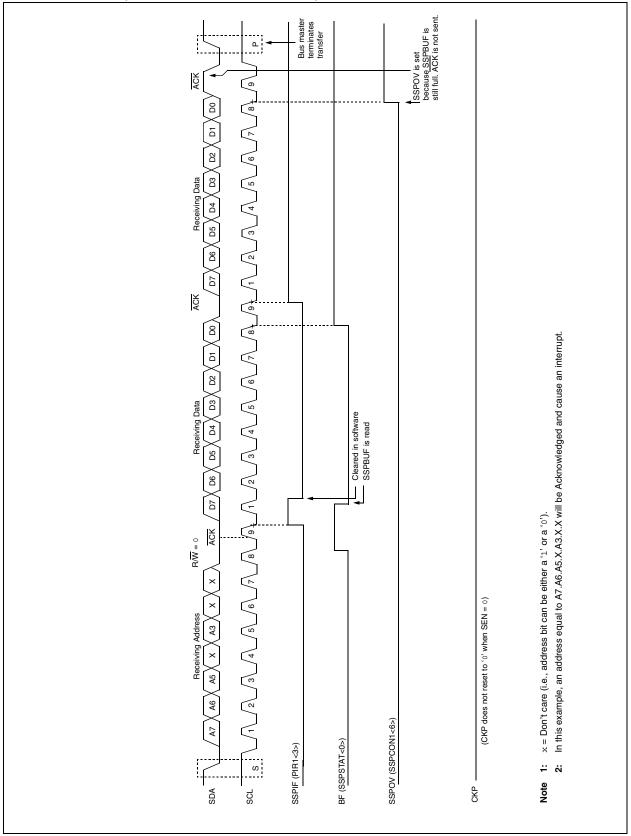
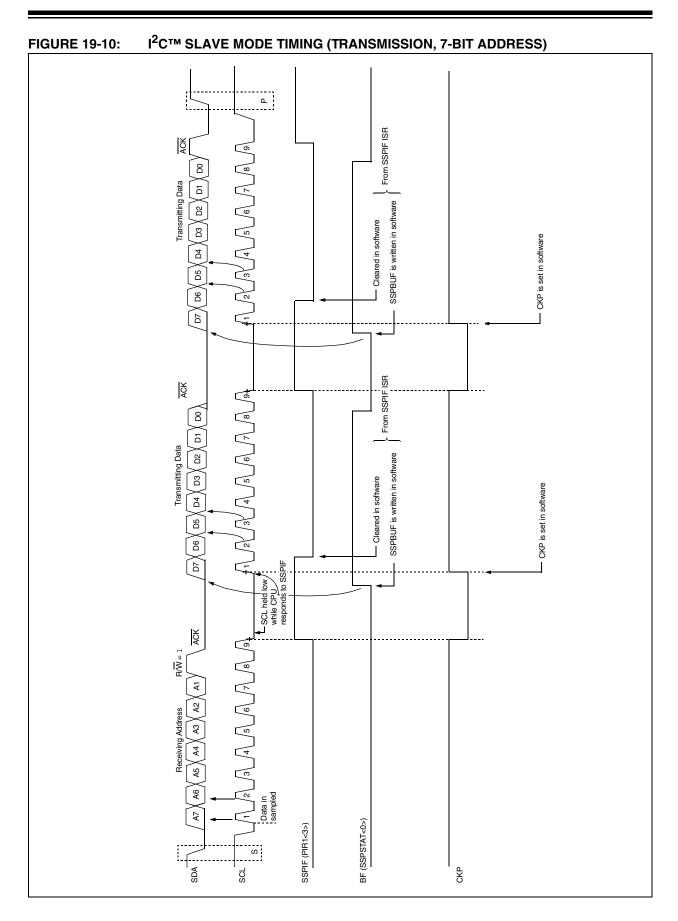


FIGURE 19-9: I<sup>2</sup>C<sup>TM</sup> SLAVE MODE TIMING WITH SEN = 0 AND ADMSK<5:1> = 01011 (RECEPTION, 7-BIT ADDRESS)





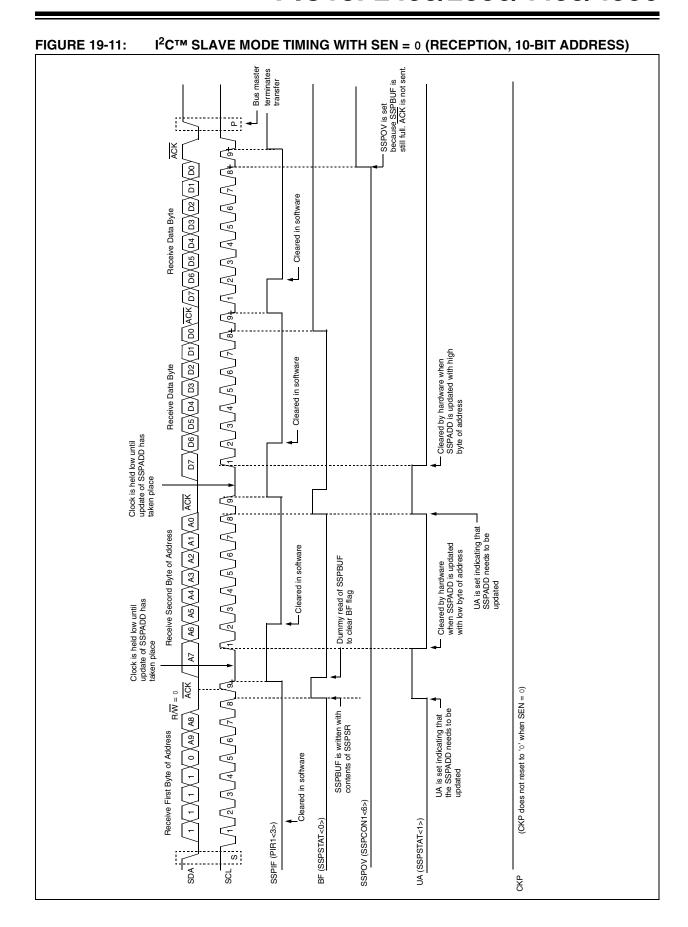
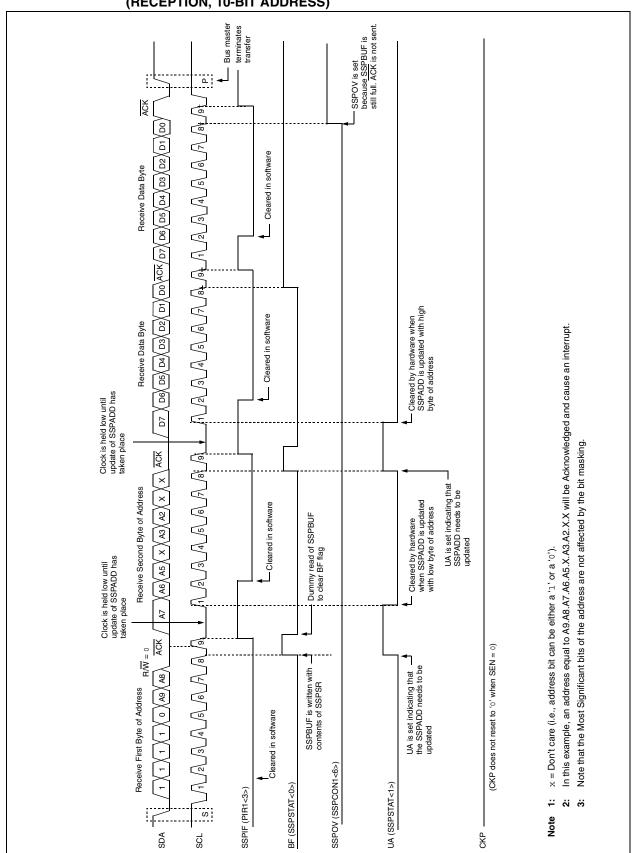
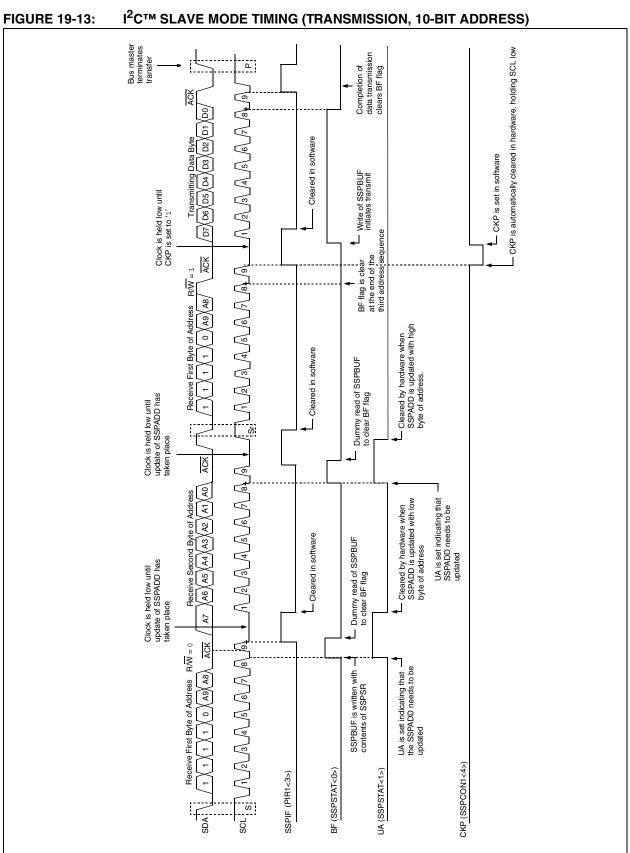


FIGURE 19-12: I<sup>2</sup>C<sup>™</sup> SLAVE MODE TIMING WITH SEN = 0 AND ADMSK<5:1> = 01001 (RECEPTION, 10-BIT ADDRESS)





#### 19.4.4 CLOCK STRETCHING

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

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

# 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  $\overline{ACK}$  sequence if the BF bit is set, the CKP bit in the SSPCON1 register is automatically cleared, forcing the SCL output to be held low. The CKP bit being cleared to '0' will assert the SCL line low. The CKP bit must be set in the user's ISR before reception is allowed to continue. By holding the SCL line low, the user has time to service the ISR and read the contents of the SSPBUF before the master device can initiate another receive sequence. This will prevent buffer overruns from occurring (see Figure 19-15).

- Note 1: If the user reads the contents of the SSPBUF before the falling edge of the ninth clock, thus clearing the BF bit, the CKP bit will not be cleared and clock stretching will not occur.
  - 2: The CKP bit can be set in software regardless of the state of the BF bit. The user should be careful to clear the BF bit in the ISR before the next receive sequence in order to prevent an overflow condition.

## 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  $R/\overline{W}$  bit cleared to 'o'. The release of the clock line occurs upon updating SSPADD. Clock stretching will occur on each data receive sequence as described in 7-bit mode.

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

## 19.4.4.3 Clock Stretching for 7-Bit Slave Transmit Mode

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

The user's ISR must set the CKP bit before transmission is allowed to continue. By holding the SCL line low, the user has time to service the ISR and load the contents of the SSPBUF before the master device can initiate another transmit sequence (see Figure 19-10).

- Note 1: If the user loads the contents of SSPBUF, setting the BF bit before the falling edge of the ninth clock, the CKP bit will not be cleared and clock stretching will not occur.
  - **2:** The CKP bit can be set in software regardless of the state of the BF bit.

# 19.4.4.4 Clock Stretching for 10-Bit Slave Transmit Mode

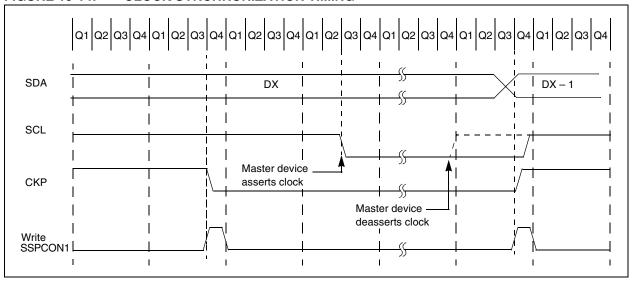
In 10-Bit Slave Transmit mode, clock stretching is controlled during the first two address sequences by the state of the UA bit, just as it is in 10-Bit Slave Receive mode. The first two addresses are followed by a third address sequence which contains the highorder bits of the 10-bit address and the 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).

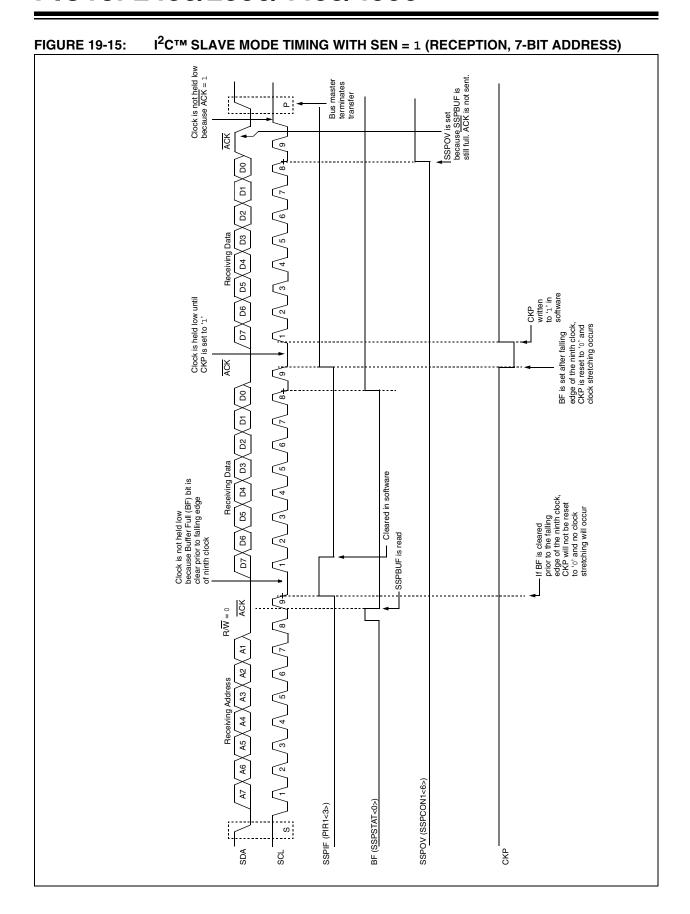
# 19.4.4.5 Clock Synchronization and the CKP bit

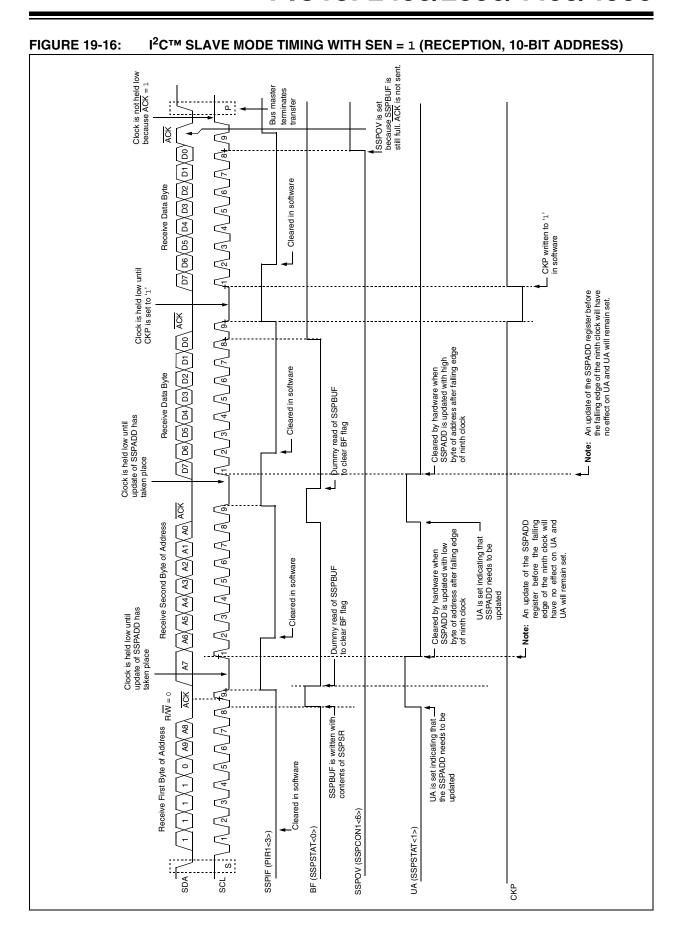
When the CKP bit is cleared, the SCL output is forced to '0'. However, clearing the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external I<sup>2</sup>C master device has

already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I<sup>2</sup>C bus have deasserted SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 19-14).









# 19.4.5 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I<sup>2</sup>C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an Acknowledge.

The general call address is one of eight addresses reserved for specific purposes by the  $I^2C$  protocol. It consists of all '0's with  $R/\overline{W} = 0$ .

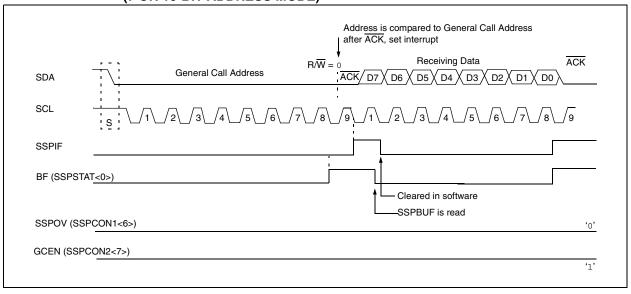
The general call address is recognized when the General Call Enable (GCEN) bit is enabled (SSPCON2<7> set). Following a Start bit detect, 8 bits are shifted into the SSPSR and the address is compared against the SSPADD. It is also compared to the general call address and fixed in hardware.

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

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

In 10-bit mode, the SSPADD is required to be updated for the second half of the address to match and the UA bit is set (SSPSTAT<1>). If the general call address is sampled when the GCEN bit is set, while the slave is configured in 10-Bit Address mode, then the second half of the address is not necessary, the UA bit will not be set and the slave will begin receiving data after the Acknowledge (Figure 19-17).

FIGURE 19-17: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE (7 OR 10-BIT ADDRESS MODE)



#### 19.4.6 MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPM bits in SSPCON1 and by setting the SSPEN bit. In Master mode, the SCL and SDA lines are manipulated by the MSSP hardware if the TRIS bits are set.

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

In Firmware Controlled Master mode, user code conducts all  $\rm I^2C$  bus operations based on Start and Stop bit conditions.

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

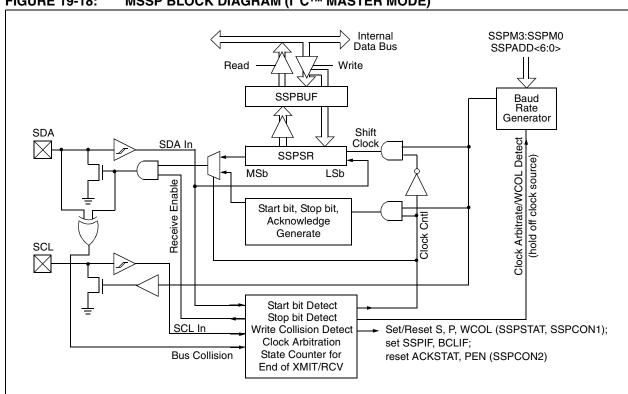
- 1. Assert a Start condition on SDA and SCL.
- Assert a Repeated Start condition on SDA and SCL.
- Write to the SSPBUF register initiating transmission of data/address.
- 4. Configure the I<sup>2</sup>C port to receive data.
- 5. Generate an Acknowledge condition at the end of a received byte of data.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

Note: The MSSP module, when configured in I<sup>2</sup>C Master mode, does not allow queueing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPBUF register to initiate transmission before the Start condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur.

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

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

FIGURE 19-18: MSSP BLOCK DIAGRAM (I<sup>2</sup>C™ MASTER MODE)



## 19.4.6.1 I<sup>2</sup>C Master Mode Operation

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I<sup>2</sup>C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (seven bits) and the Read/Write (R/ $\overline{W}$ ) bit. In this case, the R/ $\overline{W}$  bit will be logic 'o'. Serial data is transmitted eight bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

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

The Baud Rate Generator used for the SPI mode operation is used to set the SCL clock frequency for either 100 kHz, 400 kHz or 1 MHz  $I^2$ C operation. See **Section 19.4.7 "Baud Rate"** for more detail.

A typical transmit sequence would go as follows:

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

#### 19.4.7 BAUD RATE

In I<sup>2</sup>C Master mode, the Baud Rate Generator (BRG) reload value is placed in the lower seven bits of the SSPADD register (Figure 19-19). When a write occurs to SSPBUF, the Baud Rate Generator will automatically begin counting. The BRG counts down to '0' and stops until another reload has taken place. The BRG count is decremented twice per instruction cycle (TCY) on the Q2 and Q4 clocks. In I<sup>2</sup>C Master mode, the BRG is reloaded automatically.

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

Table 19-3 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPADD.

FIGURE 19-19: BAUD RATE GENERATOR BLOCK DIAGRAM

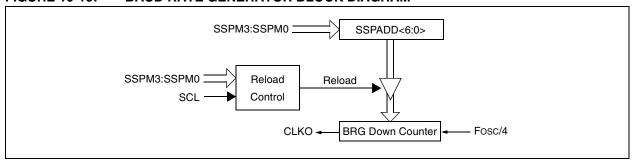


TABLE 19-3: I<sup>2</sup>C™ CLOCK RATE W/BRG

FcY	Fcy * 2	BRG Value	FSCL (2 Rollovers of BRG)
10 MHz	20 MHz	18h	400 kHz <sup>(1)</sup>
10 MHz	20 MHz	1Fh	312.5 kHz
10 MHz	20 MHz	63h	100 kHz
4 MHz	8 MHz	09h	400 kHz <sup>(1)</sup>
4 MHz	8 MHz	0Ch	308 kHz
4 MHz	8 MHz	27h	100 kHz
1 MHz	2 MHz	02h	333 kHz <sup>(1)</sup>
1 MHz	2 MHz	09h	100 kHz
1 MHz	2 MHz	00h	1 MHz <sup>(1)</sup>

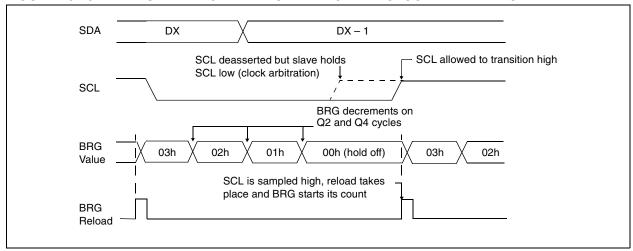
Note 1: The I<sup>2</sup>C™ interface does not conform to the 400 kHz I<sup>2</sup>C specification (which applies to rates greater than 100 kHz) in all details, but may be used with care where higher rates are required by the application.

#### 19.4.7.1 Clock Arbitration

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, deasserts the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the

SCL pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 19-20).

## FIGURE 19-20: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION



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

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

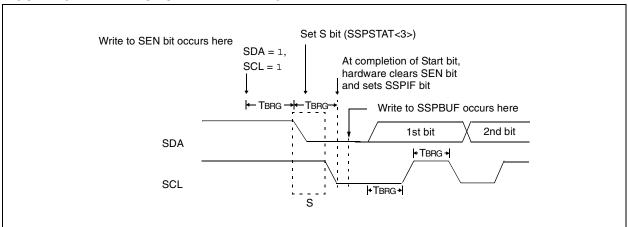
Note: If at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLIF, is set, the Start condition is aborted and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

## 19.4.8.1 WCOL Status Flag

If the user writes the SSPBUF 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).

te: Because queueing of events is not allowed, writing to the lower five bits of SSPCON2 is disabled until the Start condition is complete.

FIGURE 19-21: FIRST START BIT TIMING



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

A Repeated Start condition occurs when the RSEN bit (SSPCON2<1>) is programmed high and the I<sup>2</sup>C logic module is in the Idle state. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded with the contents of SSPADD<5:0> and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. Following this, the RSEN bit (SSPCON2<1>) will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit (SSPSTAT<3>) will be set. The SSPIF bit will not be set until the Baud Rate Generator has timed out.

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

- **2:** A bus collision during the Repeated Start condition occurs if:
  - SDA is sampled low when SCL goes from low-to-high.
  - SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data '1'.

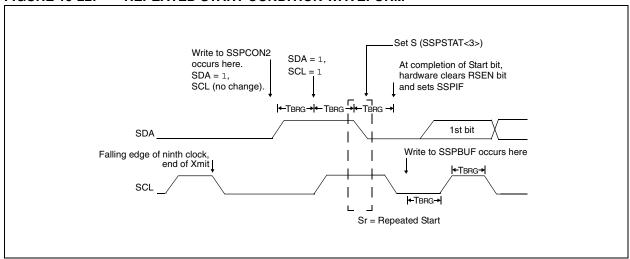
Immediately following the SSPIF bit getting set, the user may write the SSPBUF with the 7-bit address in 7-bit mode or the default first address in 10-bit mode. After the first eight bits are transmitted and an ACK is received, the user may then transmit an additional eight bits of address (10-bit mode)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

## 19.4.9.1 WCOL Status Flag

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

FIGURE 19-22: REPEATED START CONDITION WAVEFORM



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

Transmission of a data byte, a 7-bit address, or the other half of a 10-bit address is accomplished by simply writing a value to the SSPBUF register. This action will set the Buffer Full flag bit, BF and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted (see data hold time specification parameter 106). SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high (see data setup time specification parameter 107). When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an ACK bit during the ninth bit time if an address match occurred, or if data was received properly. The status of ACK is written into the ACKDT bit on the falling edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSPIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 19-23).

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

## 19.4.10.1 BF Status Flag

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

## 19.4.10.2 WCOL Status Flag

If the user writes the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur) after 2 Tcy after the SSPBUF write. If SSPBUF is rewritten within 2 Tcy, the WCOL bit is set and SSPBUF is updated. This may result in a corrupted transfer.

The user should verify that the WCOL is clear after each write to SSPBUF 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 (SSPCON2<6>) is cleared when the slave has sent an Acknowledge  $(\overline{ACK} = 0)$  and is set when the slave does not Acknowledge  $(\overline{ACK} = 1)$ . A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

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

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

**Note:** The MSSP module must be in an Idle state before the RCEN bit is set or the RCEN bit will be disregarded.

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

## 19.4.11.1 BF Status Flag

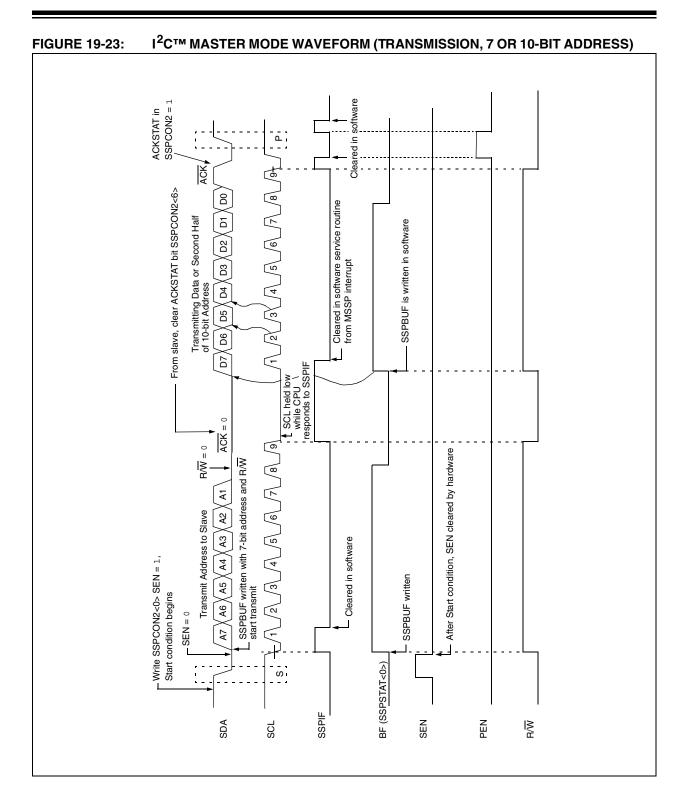
In receive operation, the BF bit is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when the SSPBUF register is read.

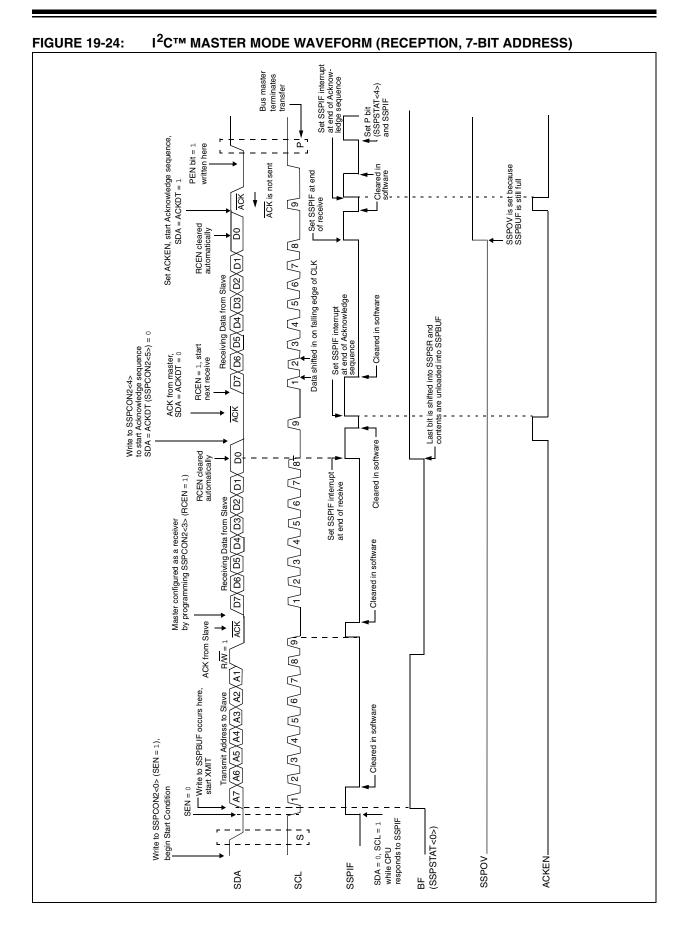
#### 19.4.11.2 SSPOV Status Flag

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

## 19.4.11.3 WCOL Status Flag

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





# 19.4.12 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit. **ACKEN** (SSPCON2<4>). When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into an inactive state (Figure 19-25).

## 19.4.12.1 WCOL Status Flag

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

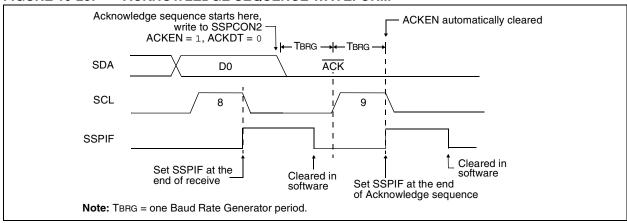
#### 19.4.13 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Enable bit, PEN (SSPCON2<2>). At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit (SSPSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 19-26).

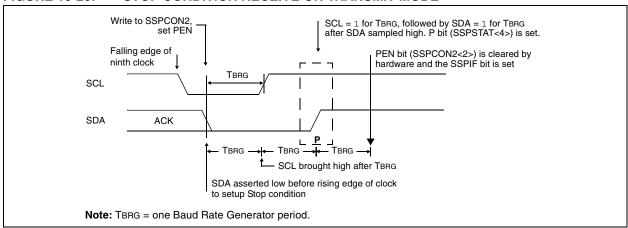
## 19.4.13.1 WCOL Status Flag

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

#### FIGURE 19-25: ACKNOWLEDGE SEQUENCE WAVEFORM



#### FIGURE 19-26: STOP CONDITION RECEIVE OR TRANSMIT MODE



#### 19.4.14 SLEEP OPERATION

While in Sleep mode, the I<sup>2</sup>C module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the 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 I<sup>2</sup>C bus may be taken when the P bit (SSPSTAT<4>) is set, or the bus is Idle, with both the S and P bits clear. When the bus is busy, enabling the MSSP interrupt will generate the interrupt when the Stop condition occurs.

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

The states where arbitration can be lost are:

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

## 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 SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin = 0, then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLIF and reset the  $I^2$ C 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 SDA and SCL lines are deasserted and the SSPBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the  $\rm I^2C$  bus is free, the user can resume communication by asserting a Start condition.

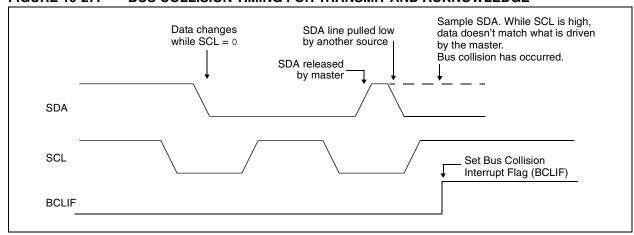
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the  $I^2C$  bus is free, the user can resume communication by asserting a Start condition.

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

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

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the  $I^2C$  bus can be taken when the P bit is set in the SSPSTAT register, or the bus is Idle and the S and P bits are cleared.





# 19.4.17.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

- a) SDA or SCL are sampled low at the beginning of the Start condition (Figure 19-28).
- SCL is sampled low before SDA is asserted low (Figure 19-29).

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

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

- · the Start condition is aborted,
- · the BCLIF flag is set and
- the MSSP module is reset to its inactive state (Figure 19-28).

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

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 19-30). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to '0'. If the SCL pin is sampled as '0', during this time a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

Note:

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

Start or Stop conditions.



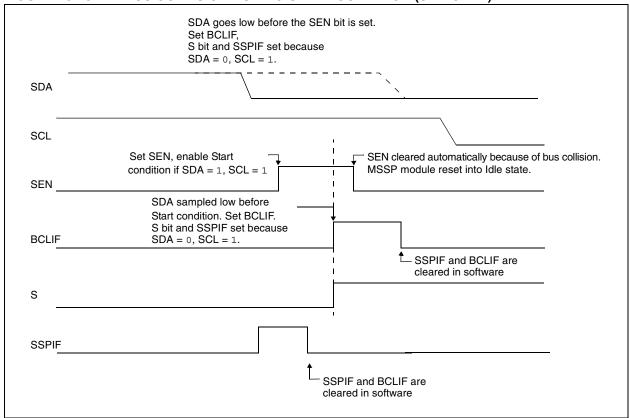


FIGURE 19-29: BUS COLLISION DURING START CONDITION (SCL = 0)

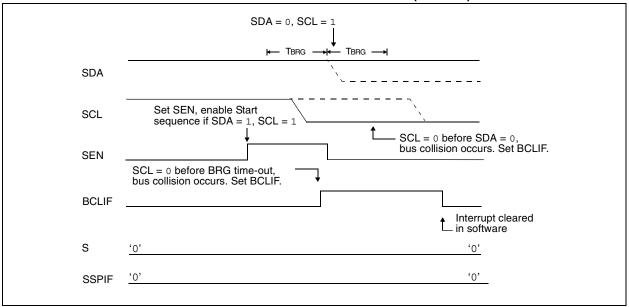
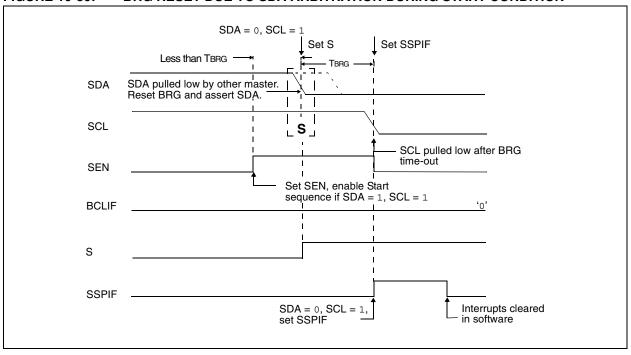


FIGURE 19-30: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION



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

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

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', see Figure 19-31). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time

If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition (see Figure 19-32).

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

FIGURE 19-31: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

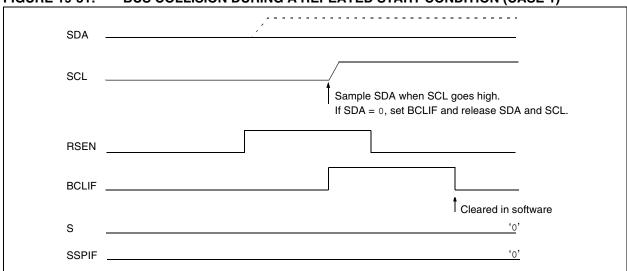
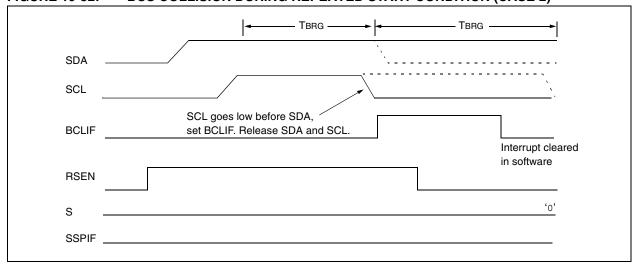


FIGURE 19-32: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)



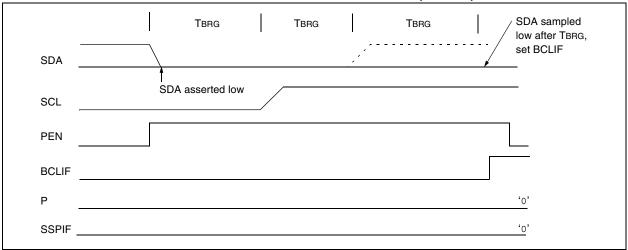
# 19.4.17.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

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

The Stop condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPADD<6:0> and counts down to '0'. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 19-33). If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 19-34).

FIGURE 19-33: BUS COLLISION DURING A STOP CONDITION (CASE 1)





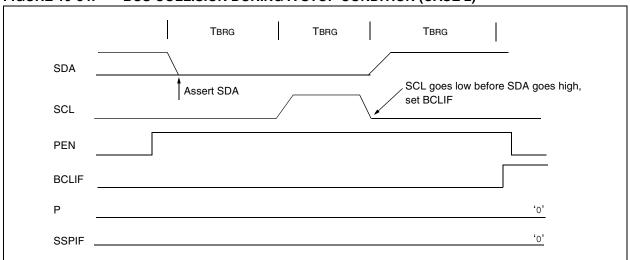


TABLE 19-4: REGISTERS ASSOCIATED WITH I<sup>2</sup>C™ OPERATION

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	51
PIR1	SPPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54
PIE1	SPPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54
IPR1	SPPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54
PIR2	OSCFIF	CMIF	USBIF	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	54
PIE2	OSCFIE	CMIE	USBIE	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	54
IPR2	OSCFIP	CMIP	USBIP	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	54
TRISC	TRISC7	TRISC6	_	_	_	TRISC2	TRISC1	TRISC0	54
TRISD <sup>(1)</sup>	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	54
SSPBUF	MSSP Rec	eive Buffer/T	ransmit Reg	jister					52
SSPADD		ress Registe d Rate Reloa			r mode.				52
TMR2	Timer2 Reg	gister							52
PR2	Timer2 Per	iod Register							52
SSPCON1	WCOL	SSPOV	POV SSPEN CKP SSPM3 SSPM2 SSPM1 SSPM0						
SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	52
SSPSTAT	SMP	CKE	$D/\overline{A}$	Р	S	R/W	UA	BF	52

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by the MSSP in  $I^2C^{TM}$  mode.

Note 1: These registers or bits are not implemented in 28-pin devices.

# 20.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of the two serial I/O modules. (Generically, the USART 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.

The EUSART can be configured in the following modes:

- Asynchronous (full-duplex) with:
  - Auto-wake-up on Break signal
  - 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 the Enhanced USART are multiplexed with PORTC. In order to configure RC6/TX/CK and RC7/RX/DT/SDO as an EUSART:

- bit SPEN (RCSTA<7>) must be set (= 1)
- bit TRISC<7> must be set (= 1)
- bit TRISC<6> must be set (= 1)

**Note:** The EUSART control will automatically reconfigure the pin from input to output as needed.

The operation of the Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCON)

These are detailed on the following pages in Register 20-1, Register 20-2 and Register 20-3, respectively.

## REGISTER 20-1: TXSTA: 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	RC TX9 TXEN <sup>(1)</sup> S		SYNC	SENDB	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 transmission0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit<sup>(1)</sup>

1 = Transmit enabled0 = Transmit disabled

bit 4 SYNC: EUSART Mode Select bit

1 = Synchronous mode0 = Asynchronous mode

bit 3 SENDB: 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 with the exception that SREN has no effect in Synchronous Slave mode.

## REGISTER 20-2: RCSTA: 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	PEN RX9 SREN C		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 RX/DT and TX/CK pins as serial port pins)

0 = Serial port disabled (held in Reset)

bit 6 RX9: 9-Bit Receive Enable bit

1 = Selects 9-bit reception0 = Selects 8-bit reception

bit 5 SREN: Single Receive Enable bit

Asynchronous mode:

Don't care.

Synchronous mode - Master:

1 = Enables single receive

0 = Disables single receive

This bit is cleared after reception is complete.

Synchronous mode - Slave:

Don't care.

bit 4 CREN: Continuous Receive Enable bit

Asynchronous mode:

1 = Enables receiver

0 = Disables receiver

Synchronous mode:

1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)

0 = Disables continuous receive

bit 3 ADDEN: Address Detect Enable bit

Asynchronous mode 9-bit (RX9 = 1):

1 = Enables address detection, enables interrupt and 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

Asynchronous mode 9-bit (RX9 = 0):

Don't care.

bit 2 FERR: Framing Error bit

1 = Framing error (can be updated by reading RCREG 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.

#### REGISTER 20-3: BAUDCON: 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	RXDTP	TXCKP	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 bit

1 = 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 bit

1 = Receive operation is Idle0 = Receive operation is active

bit 5 RXDTP: Received Data Polarity Select bit

Asynchronous mode:

1 = RX data is inverted

0 = RX data received is not inverted

Synchronous modes:

1 = CK clocks are inverted

0 = CK clocks are not inverted

bit 4 TXCKP: Clock and Data Polarity Select bit

Asynchronous mode:

1 = TX data is inverted

0 = TX data is not inverted

Synchronous modes:

1 = CK clocks are inverted

0 = CK clocks are not inverted

bit 3 BRG16: 16-Bit Baud Rate Register Enable bit

1 = 16-bit Baud Rate Generator - SPBRGH and SPBRG

0 = 8-bit Baud Rate Generator - SPBRG only (Compatible mode), SPBRGH value ignored

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

bit 1 WUE: Wake-up Enable bit

Asynchronous mode:

1 = EUSART will continue to sample the RX pin – interrupt generated on falling edge; bit cleared in hardware on following rising edge

0 = RX pin not monitored or rising edge detected

Synchronous mode:

Unused in this mode.

bit 0 ABDEN: Auto-Baud Detect Enable bit

Asynchronous 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 completed

Synchronous 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 (BAUDCON<3>) selects 16-bit mode.

The SPBRGH:SPBRG register pair controls the period of a free-running timer. In Asynchronous mode, bits BRGH (TXSTA<2>) and BRG16 (BAUDCON<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 SPBRGH:SPBRG 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 SPBRGH:SPBRG 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 SPBRG register pair.

#### 20.1.2 SAMPLING

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

TABLE 20-1: BAUD RATE FORMULAS

C	onfiguration B	its	BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH	BRG/EUSART Mode	Baud Hate Formula
0	0	0	8-bit/Asynchronous	Fosc/[64 (n + 1)]
0	0	1	8-bit/Asynchronous	F000/[16 (n + 1)]
0	1	0	16-bit/Asynchronous	Fosc/[16 (n + 1)]
0	1	1	16-bit/Asynchronous	
1	0	х	8-bit/Synchronous	Fosc/[4 (n + 1)]
1	1 1 x		16-bit/Synchronous	

**Legend:** x = Don't care, n = value of SPBRGH:SPBRG register pair

## **EXAMPLE 20-1: CALCULATING BAUD RATE ERROR**

For a device with Fosc of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

Desired Baud Rate = Fosc/(64 ([SPBRGH:SPBRG] + 1))

Solving for SPBRGH:SPBRG:

X = ((Fosc/Desired Baud Rate)/64) - 1

= ((16000000/9600)/64) - 1

= [25.042] = 25

Calculated Baud Rate = 16000000/(64 (25 + 1))

= 9615

Error = (Calculated Baud Rate – Desired Baud Rate)/Desired Baud Rate

= (9615 - 9600)/9600 = 0.16%

## 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	
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	53	
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	53	
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	_	WUE	ABDEN	53	
SPBRGH	EUSART B	EUSART Baud Rate Generator Register High Byte								
SPBRG	EUSART Baud Rate Generator Register Low Byte									

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by the BRG.

TABLE 20-3: BAUD RATES FOR ASYNCHRONOUS MODES

					SYNC	= 0, BRG	l = 0, BRG	<b>316</b> = 0				
BAUD RATE	Fosc	= 40.000	= 40.000 MHz F		osc = 20.000 MHz		Fosc = 10.000 MHz			Fosc = 8.000 MHz		
(K)	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	1201	-0.16	103
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2403	-0.16	51
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9615	-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	-	_	_

		SYNC = 0, BRGH = 0, BRG16 = 0												
BAUD	Fos	c = 4.000	MHz	Fos	c = 2.000	MHz	Fos	Fosc = 1.000 MHz						
RATE (K)	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	300	-0.16	103	300	-0.16	51					
1.2	1.202	0.16	51	1201	-0.16	25	1201	-0.16	12					
2.4	2.404	0.16	25	2403	-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	_	_	_	_	_	_					

					SYNC	= 0, BRG	l = 1, BRG	i <b>16 =</b> 0					
BAUD	Fosc	= 40.000	) MHz	Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fos	Fosc = 8.000 MHz		
RATE (K)	Actual Rate (K)	ate		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	2403	-0.16	207	
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9615	-0.16	51	
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19230	-0.16	25	
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55555	3.55	8	
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	_	_	_	

			s'	YNC = 0, E	BRGH = 1	., BRG16 =	0			
BAUD RATE	Fosc	= 4.000	MHz	Fos	c = 2.000	MHz	Fos	Fosc = 1.000 MHz		
(K)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	
0.3	_	_	_	-	_	_	300	-0.16	207	
1.2	1.202	0.16	207	1201	-0.16	103	1201	-0.16	51	
2.4	2.404	0.16	103	2403	-0.16	51	2403	-0.16	25	
9.6	9.615	0.16	25	9615	-0.16	12	_	_	_	
19.2	19.231	0.16	12	_	_	_	_	_	_	
57.6	62.500	8.51	3	_	_	_	_	_	_	
115.2	125.000	8.51	1		_	_	_	_	_	

TABLE 20-3: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

					SYNC	= 0, BRGI	H = 0, BRG	116 = 1					
BAUD RATE	Fosc	= 40.000	) MHz	Fosc = 20.000 MHz			Fosc	Fosc = 10.000 MHz			Fosc = 8.000 MHz		
(K)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% val	SPBRG value (decimal)	
0.3	0.300	0.00	8332	0.300	0.02	4165	0.300	0.02	2082	300	-0.04	1665	
1.2	1.200	0.02	2082	1.200	-0.03	1041	1.200	-0.03	520	1201	-0.16	415	
2.4	2.402	0.06	1040	2.399	-0.03	520	2.404	0.16	259	2403	-0.16	207	
9.6	9.615	0.16	259	9.615	0.16	129	9.615	0.16	64	9615	-0.16	51	
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19230	-0.16	25	
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55555	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			Fos	c = 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)			
0.3	0.300	0.04	832	300	-0.16	415	300	-0.16	207			
1.2	1.202	0.16	207	1201	-0.16	103	1201	-0.16	51			
2.4	2.404	0.16	103	2403	-0.16	51	2403	-0.16	25			
9.6	9.615	0.16	25	9615	-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	300	-0.01	6665	
1.2	1.200	0.00	8332	1.200	0.02	4165	1.200	0.02	2082	1200	-0.04	1665	
2.4	2.400	0.02	4165	2.400	0.02	2082	2.402	0.06	1040	2400	-0.04	832	
9.6	9.606	0.06	1040	9.596	-0.03	520	9.615	0.16	259	9615	-0.16	207	
19.2	19.193	-0.03	520	19.231	0.16	259	19.231	0.16	129	19230	-0.16	103	
57.6	57.803	0.35	172	57.471	-0.22	86	58.140	0.94	42	57142	0.79	34	
115.2	114.943	-0.22	86	116.279	0.94	42	113.636	-1.36	21	117647	-2.12	16	

		SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1												
BAUD RATE (K)	Fos	c = 4.000	MHz	Fos	c = <b>2.000</b>	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)					
0.3	0.300	0.01	3332	300	-0.04	1665	300	-0.04	832					
1.2	1.200	0.04	832	1201	-0.16	415	1201	-0.16	207					
2.4	2.404	0.16	415	2403	-0.16	207	2403	-0.16	103					
9.6	9.615	0.16	103	9615	-0.16	51	9615	-0.16	25					
19.2	19.231	0.16	51	19230	-0.16	25	19230	-0.16	12					
57.6	58.824	2.12	16	55555	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 RX signal, the RX 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 SPBRG begins counting up, using the preselected clock source on the first rising edge of RX. After eight bits on the RX pin, or the fifth rising edge, an accumulated value totalling the proper BRG period is left in the SPBRGH:SPBRG 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 (BAUDCON<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 SPBRG and SPBRGH 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 SPBRGH register. Refer to Table 20-4 for counter clock rates to the BRG.

While the ABD sequence takes place, the EUSART state machine is held in Idle. The RCIF interrupt is set once the fifth rising edge on RX is detected. The value in the RCREG needs to be read to clear the RCIF interrupt. The contents of RCREG 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

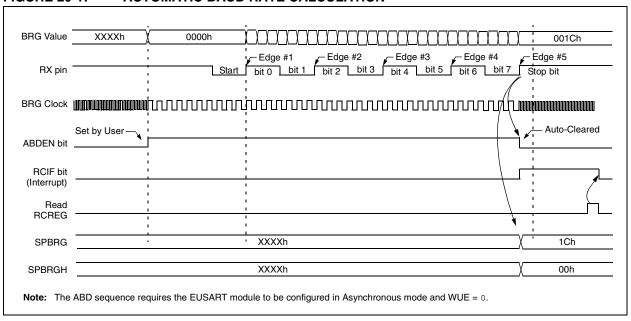
BRG16	BRGH	BRG Counter Clock
0	0	Fosc/512
0	1	Fosc/128
1	0	Fosc/128
1	1	Fosc/32

Note: During the ABD sequence, SPBRG and SPBRGH are both used as a 16-bit counter, independent of the BRG16 setting.

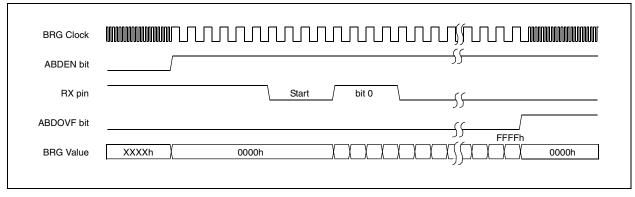
#### 20.1.3.1 ABD and EUSART Transmission

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, TXREG 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.

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 (TXSTA<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 (TXSTA<2> and BAUDCON<3>). Parity is not supported by the hardware but can be implemented in software and stored as the 9th data bit.

The TXCKP (BAUDCON<4>) and RXDTP (BAUDCON<5>) bits allow the TX and RX signals to be inverted (polarity reversed). Devices that buffer signals between TTL and RS-232 levels also invert the signal. Setting the TXCKP and RXDTP bits allows for the use of circuits that provide buffering without inverting the signal.

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 Break signal
- 12-Bit Break Character Transmit
- · Auto-Baud Rate Detection
- · Pin State Polarity

# 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, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXREG register (if available).

Once the TXREG register transfers the data to the TSR register (occurs in one Tcy), the TXREG register is empty and the TXIF flag bit (PIR1<4>) is set. This interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TXIE (PIE1<4>). TXIF will be set regardless of the state of TXIE; it cannot be cleared in software. TXIF is also not cleared immediately upon loading TXREG, but becomes valid in the second instruction cycle following the load instruction. Polling TXIF immediately following a load of TXREG will return invalid results.

While TXIF indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR 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.

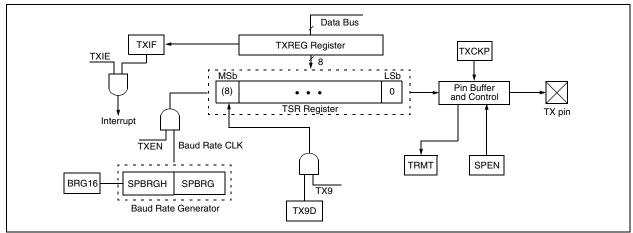
The TXCKP bit (BAUDCON<4>) allows the TX signal to be inverted (polarity reversed). Devices that buffer signals from TTL to RS-232 levels also invert the signal (when TTL = 1, RS-232 = negative). Inverting the polarity of the TX pin data by setting the TXCKP bit allows for use of circuits that provide buffering without inverting the signal.

- **Note 1:** The TSR register is not mapped in data memory so it is not available to the user.
  - 2: Flag bit TXIF is set when enable bit TXEN is set.

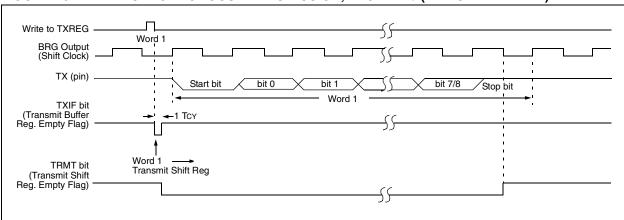
To set up an Asynchronous Transmission:

- Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
- Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
- If the signal from the TX pin is to be inverted, set the TXCKP bit.
- 4. If interrupts are desired, set enable bit TXIE.
- If 9-bit transmission is desired, set transmit bit TX9. Can be used as address/data bit.
- 6. Enable the transmission by setting bit TXEN which will also set bit TXIF.
- If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
- Load data to the TXREG register (starts transmission).
- 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



## FIGURE 20-4: ASYNCHRONOUS TRANSMISSION, TXCKP = 0 (TX NOT INVERTED)



# FIGURE 20-5: ASYNCHRONOUS TRANSMISSION (BACK TO BACK), TXCKP = 0 (TX NOT INVERTED)

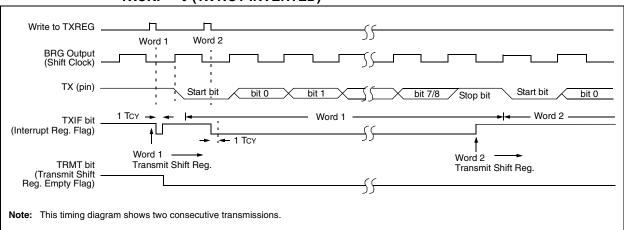


TABLE 20-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

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	51	
PIR1	SPPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54	
PIE1	SPPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54	
IPR1	SPPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54	
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	53	
TXREG	EUSART Transmit Register									
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	53	
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	_	WUE	ABDEN	53	
SPBRGH	EUSART Baud Rate Generator Register High Byte									
SPBRG	EUSART E	Baud Rate G	enerator Re	gister Low E	Byte	•		•	53	

**Legend:** — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.

Note 1: Reserved in 28-pin devices; always maintain these bits clear.

# 20.2.2 EUSART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 20-6. The data is received on the RX 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.

The RXDTP bit (BAUDCON<5>) allows the RX signal to be inverted (polarity reversed). Devices that buffer signals from RS-232 to TTL levels also perform an inversion of the signal (when RS-232 = positive, TTL = 0). Inverting the polarity of the RX pin data by setting the RXDTP bit allows for the use of circuits that provide buffering without inverting the signal.

To set up an Asynchronous Reception:

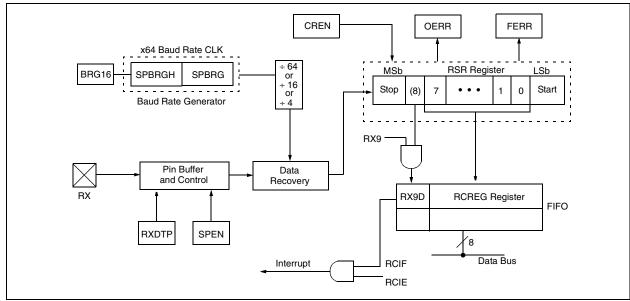
- Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
- Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
- If the signal at the RX pin is to be inverted, set the RXDTP bit.
- 4. If interrupts are desired, set enable bit RCIE.
- 5. If 9-bit reception is desired, set bit RX9.
- 6. Enable the reception by setting bit CREN.
- Flag bit, RCIF, will be set when reception is complete and an interrupt will be generated if enable bit, RCIE, was set.
- 8. Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
- Read the 8-bit received data by reading the RCREG register.
- If any error occurred, clear the error by clearing enable bit CREN.
- If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

# 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:

- Initialize the SPBRGH:SPBRG 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.
- If the signal at the RX pin is to be inverted, set the RXDTP bit. If the signal from the TX pin is to be inverted, set the TXCKP bit.
- If interrupts are required, set the RCEN bit and select the desired priority level with the RCIP bit.
- 5. Set the RX9 bit to enable 9-bit reception.
- 6. Set the ADDEN bit to enable address detect.
- 7. Enable reception by setting the CREN bit.
- The RCIF bit will be set when reception is complete. The interrupt will be Acknowledged if the RCIE and GIE bits are set.
- 9. Read the RCSTA register to determine if any error occurred during reception, as well as read bit 9 of data (if applicable).
- Read RCREG to determine if the device is being addressed.
- 11. If any error occurred, clear the CREN bit.
- If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and interrupt the CPU.

#### FIGURE 20-6: EUSART RECEIVE BLOCK DIAGRAM



## FIGURE 20-7: ASYNCHRONOUS RECEPTION, RXDTP = 0 (RX NOT INVERTED)

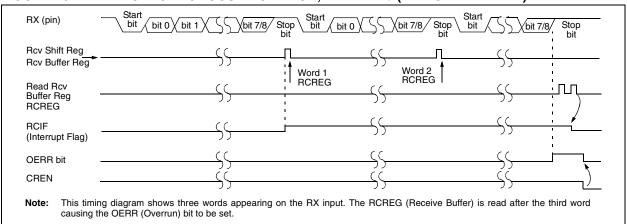


TABLE 20-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

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	51	
PIR1	SPPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54	
PIE1	SPPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54	
IPR1	SPPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54	
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	53	
RCREG	EUSART F	Receive Regis	ster						53	
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	53	
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	_	WUE	ABDEN	53	
SPBRGH	EUSART E	EUSART Baud Rate Generator Register High Byte								
SPBRG	EUSART E	Baud Rate Ge	enerator Re	gister Low E	Byte		•		53	

**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 RX/DT line while the EUSART is operating in Asynchronous mode.

The auto-wake-up feature is enabled by setting the WUE bit (BAUDCON<1>). Once set, the typical receive sequence on RX/DT 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 RX/DT 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 RCIF 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 RCREG register.

The WUE bit is automatically cleared once a low-tohigh transition is observed on the RX 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 RX/DT, information with any state changes before the Stop bit may signal a false End-of-

Character 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., XT or HS 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.

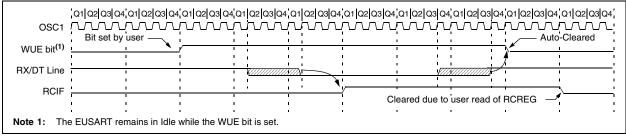
# 20.2.4.2 Special Considerations Using the WUE Bit

The timing of WUE and RCIF 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 RCIF bit. The WUE bit is cleared after this when a rising edge is seen on RX/DT. The interrupt condition is then cleared by reading the RCREG register. Ordinarily, the data in RCREG will be dummy data and should be discarded.

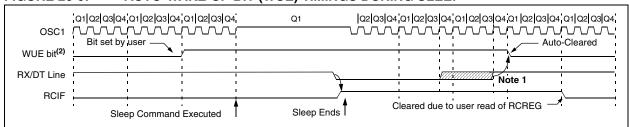
The fact that the WUE bit has been cleared (or is still set) and the RCIF flag is set should not be used as an indicator of the integrity of the data in RCREG. Users should consider implementing a parallel method in firmware to verify received data integrity.

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



Note 1: If the wake-up event requires long oscillator warm-up time, the auto-clear of the WUE bit can occur before the oscillator is ready. This sequence should not depend on the presence of Q clocks.

2: The EUSART remains in Idle while the WUE bit is set.

#### 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 (TXSTA<3> and TXSTA<5>) are set while the Transmit Shift register is loaded with data. Note that the value of data written to TXREG 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 TXREG 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.
- Set the TXEN and SENDB bits to set up the Break character.
- 3. Load the TXREG with a dummy character to initiate transmission (the value is ignored).
- Write '55h' to TXREG to load the Sync character into the transmit FIFO buffer.
- After the Break has been sent, the SENDB bit is reset by hardware. The Sync character now transmits in the preconfigured mode.

When the TXREG becomes empty, as indicated by the TXIF, the next data byte can be written to TXREG.

#### 20.2.6 RECEIVING A BREAK CHARACTER

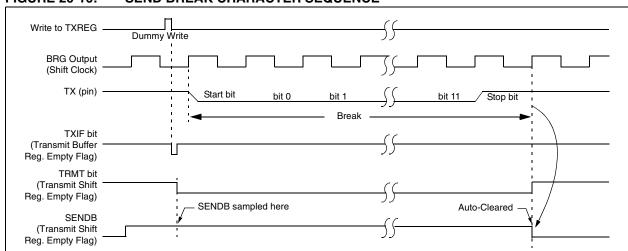
The Enhanced USART module can receive a Break character in two ways.

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 RX/DT, cause an RCIF 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 ABD bit once the TXIF interrupt is observed.





© 2006 Microchip Technology Inc. Preliminary DS39632C-page 253

## 20.3 EUSART Synchronous Master Mode

The Synchronous Master mode is entered by setting the CSRC bit (TXSTA<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 (TXSTA<4>). In addition, enable bit, SPEN (RCSTA<7>), is set in order to configure the TX and RX pins to CK (clock) and DT (data) lines, respectively.

The Master mode indicates that the processor transmits the master clock on the CK line.

Clock polarity (CK) is selected with the TXCKP bit (BAUDCON<4>). Setting TXCKP sets the Idle state on CK as high, while clearing the bit sets the Idle state as low. Data polarity (DT) is selected with the RXDTP bit (BAUDCON<5>). Setting RXDTP sets the Idle state on DT as high, while clearing the bit sets the Idle state as low. DT is sampled when CK returns to its idle state. 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, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available).

Once the TXREG register transfers the data to the TSR register (occurs in one TCY), the TXREG is empty and the TXIF flag bit (PIR1<4>) is set. The interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TXIE (PIE1<4>). TXIF is set regardless of the state of enable bit TXIE; it cannot be cleared in software. It will reset only when new data is loaded into the TXREG register.

While flag bit, TXIF, indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR is empty. No interrupt logic is tied to this bit so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory so it is not available to the user.

To set up a Synchronous Master Transmission:

- Initialize the SPBRGH:SPBRG 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.
- If the signal from the CK pin is to be inverted, set the TXCKP bit. If the signal from the DT pin is to be inverted, set the RXDTP bit.
- 4. If interrupts are desired, set enable bit TXIE.
- 5. If 9-bit transmission is desired, set bit TX9.
- 6. Enable the transmission by setting bit TXEN.
- If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
- Start transmission by loading data to the TXREG register.
- If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.



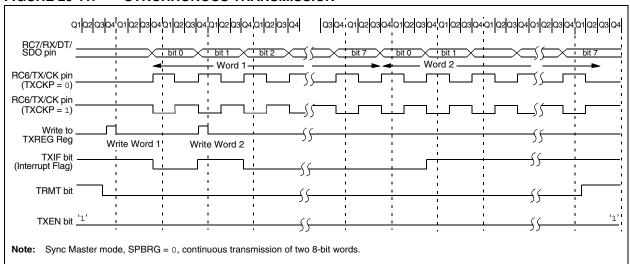


FIGURE 20-12: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)

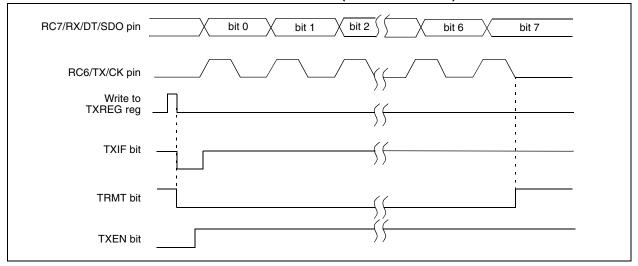


TABLE 20-7: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

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	51	
PIR1	SPPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54	
PIE1	SPPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54	
IPR1	SPPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54	
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	53	
TXREG	EUSART T	ransmit Reg	ister						53	
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	53	
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	_	WUE	ABDEN	53	
SPBRGH	EUSART E	EUSART Baud Rate Generator Register High Byte								
SPBRG	EUSART E	Baud Rate G	enerator Re	gister Low	Byte				53	

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.

Note 1: Reserved in 28-pin devices; always maintain these bits clear.

© 2006 Microchip Technology Inc. Preliminary DS39632C-page 255

# 20.3.2 EUSART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either the Single Receive Enable bit, SREN (RCSTA<5>), or the Continuous Receive Enable bit, CREN (RCSTA<4>). Data is sampled on the RX pin on the falling edge of the clock.

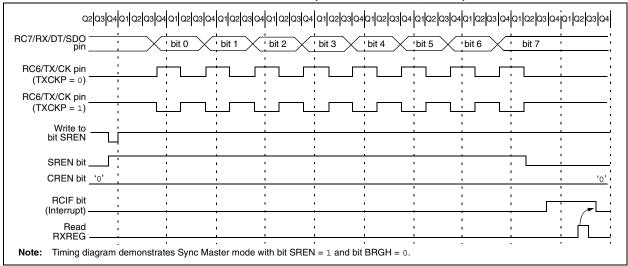
If enable bit SREN is set, only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set, then CREN takes precedence.

To set up a Synchronous Master Reception:

- Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRG16 bit, as required, to achieve the desired baud rate.
- Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
- 3. Ensure bits CREN and SREN are clear.

- If the signal from the CK pin is to be inverted, set the TXCKP bit. If the signal from the DT pin is to be inverted, set the RXDTP bit.
- 5. If interrupts are desired, set enable bit RCIE.
- 6. If 9-bit reception is desired, set bit RX9.
- 7. If a single reception is required, set bit SREN. For continuous reception, set bit CREN.
- Interrupt flag bit, RCIF, will be set when reception is complete and an interrupt will be generated if the enable bit, RCIE, was set.
- Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
- Read the 8-bit received data by reading the RCREG register.
- 11. If any error occurred, clear the error by clearing bit CREN.
- 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	51
PIR1	SPPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54
PIE1	SPPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54
IPR1	SPPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	53
RCREG	EUSART R	eceive Regi	ster						53
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	53
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	_	WUE	ABDEN	53
SPBRGH	EUSART B	EUSART Baud Rate Generator Register High Byte							
SPBRG	EUSART B	aud Rate Ge	enerator R	egister Lov	v Byte	•	•		53

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used for synchronous master reception.

# 20.4 EUSART Synchronous Slave Mode

Synchronous Slave mode is entered by clearing bit, CSRC (TXSTA<7>). This mode differs from the Synchronous Master mode in that the shift clock is supplied externally at the CK pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in any power-managed mode.

# 20.4.1 EUSART SYNCHRONOUS SLAVE TRANSMISSION

The operation of the Synchronous Master and Slave modes are identical, except in the case of the Sleep mode.

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

- a) The first word will immediately transfer to the TSR register and transmit.
- The second word will remain in the TXREG register.
- c) Flag bit, TXIF, will not be set.
- d) When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and flag bit, TXIF, will now be set.
- If enable bit TXIE 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.
- 3. If interrupts are desired, set enable bit TXIE.
- 4. If the signal from the CK pin is to be inverted, set the TXCKP bit. If the signal from the DT pin is to be inverted, set the RXDTP bit.
- 5. If 9-bit transmission is desired, set bit TX9.
- Enable the transmission by setting enable bit TXEN.
- 7. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
- 8. 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	51	
PIR1	SPPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54	
PIE1	SPPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54	
IPR1	SPPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54	
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	53	
TXREG	EUSART T	ransmit Reg	ster						53	
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	53	
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	_	WUE	ABDEN	53	
SPBRGH	EUSART E	EUSART Baud Rate Generator Register High Byte								
SPBRG	EUSART E	EUSART Baud Rate Generator Register Low Byte								

**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 RCREG register. If the RCIE 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:

- Enable the synchronous master serial port by setting bits SYNC and SPEN and clearing bit CSRC.
- 2. If interrupts are desired, set enable bit RCIE.
- If the signal from the CK pin is to be inverted, set the TXCKP bit. If the signal from the DT pin is to be inverted, set the RXDTP bit.
- 4. If 9-bit reception is desired, set bit RX9.
- 5. To enable reception, set enable bit CREN.
- Flag bit, RCIF, will be set when reception is complete. An interrupt will be generated if enable bit, RCIE, was set.
- Read the RCSTA 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 RCREG register.
- If any error occurred, clear the error by clearing bit CREN.
- 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	51
PIR1	SPPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54
PIE1	SPPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54
IPR1	SPPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	53
RCREG	EUSART F	Receive Regi	ster						53
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	53
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	_	WUE	ABDEN	53
SPBRGH	EUSART Baud Rate Generator Register High Byte								
SPBRG	EUSART Baud Rate Generator Register Low Byte								

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used for synchronous slave reception.

# 21.0 10-BIT ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) converter module has 10 inputs for the 28-pin devices and 13 for the 40/44-pin devices. This module allows conversion of an analog input signal to a corresponding 10-bit digital number.

The module has five registers:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)
- A/D Control Register 2 (ADCON2)

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 functions of the port pins. The ADCON2 register, shown in Register 21-3, configures the A/D clock source, programmed acquisition time and justification.

#### REGISTER 21-1: ADCON0: A/D CONTROL REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	_	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 Unimplemented: Read as '0'

bit 5-2 CHS3:CHS0: Analog Channel Select bits

0000 = Channel 0 (AN0)

0001 = Channel 1 (AN1)

0010 = Channel 2 (AN2)

0011 = Channel 3 (AN3)

0100 = Channel 4 (AN4)

0101 = Channel 5 (AN5) $^{(1,2)}$ 

0110 = Channel 6 (AN6) $^{(1,2)}$ 

0111 = Channel 7 (AN7)(1,2)

1000 = Channel 8 (AN8)

1001 = Channel 9 (AN9)

1010 = Channel 10 (AN10)

1011 = Channel 11 (AN11)

1100 = Channel 12 (AN12)

1101 = Unimplemented<sup>(2)</sup>

1110 = Unimplemented<sup>(2)</sup>

1111 = Unimplemented<sup>(2)</sup>

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: These channels are not implemented on 28-pin devices.

2: Performing a conversion on unimplemented channels will return a floating input measurement.

#### REGISTER 21-2: ADCON1: A/D CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0 <sup>(1)</sup>	R/W <sup>(1)</sup>	R/W <sup>(1)</sup>	R/W <sup>(1)</sup>
_	_	VCFG0	VCFG0	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 **Unimplemented:** Read as '0'

bit 5 VCFG0: Voltage Reference Configuration bit (VREF- source)

1 = VREF- (AN2)

0 = Vss

bit 4 **VCFG0:** Voltage Reference Configuration bit (VREF+ source)

1 = VREF+ (AN3)

0 = VDD

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

- OI GO.1													
PCFG3: PCFG0	AN12	AN11	AN10	6NA	AN8	AN7 <sup>(2)</sup>	AN6 <sup>(2)</sup>	AN5 <sup>(2)</sup>	AN4	AN3	AN2	AN1	ANO
0000(1)	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α
0001	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α
0010	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α
0011	D	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α
0100	D	D	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α
0101	D	D	D	Α	Α	Α	Α	Α	Α	Α	Α	Α	Α
0110	D	D	D	D	Α	Α	Α	Α	Α	Α	Α	Α	Α
0111(1)	D	D	D	D	D	Α	Α	Α	Α	Α	Α	Α	Α
1000	D	D	D	D	D	D	Α	Α	Α	Α	Α	Α	Α
1001	D	D	D	D	D	D	D	Α	Α	Α	Α	Α	Α
1010	D	D	D	D	D	D	D	D	Α	Α	Α	Α	Α
1011	D	D	D	D	D	D	D	D	D	Α	Α	Α	Α
1100	D	D	D	D	D	D	D	D	D	D	Α	Α	Α
1101	D	D	D	D	D	D	D	D	D	D	D	Α	Α
1110	D	D	D	D	D	D	D	D	D	D	D	D	Α
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

A = Analog input

D = Digital I/O

**Note 1:** The POR value of the PCFG bits depends on the value of the PBADEN Configuration bit. When PBADEN = 1, PCFG<3:0> = 0000; when PBADEN = 0, PCFG<3:0> = 0111.

2: AN5 through AN7 are available only on 40/44-pin devices.

#### REGISTER 21-3: ADCON2: A/D CONTROL REGISTER 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	_	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 Unimplemented: Read as '0'

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 T**AD

010 = 4 TAD

001 = 2 TAD 000 = 0 TAD<sup>(1)</sup>

bit 2-0 ADCS2:ADCS0: A/D Conversion Clock Select bits

111 = FRC (clock derived from A/D RC oscillator)(1)

110 = Fosc/64

101 = Fosc/16

100 = Fosc/4

011 = FRC (clock derived from A/D RC oscillator)(1)

010 = Fosc/32

001 = Fosc/8

000 = Fosc/2

**Note 1:** 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 analog reference voltage is software selectable to either the device's positive and negative supply voltage (VDD and VSS) or the voltage level on the RA3/AN3/VREF+ and RA2/AN2/VREF-/CVREF pins.

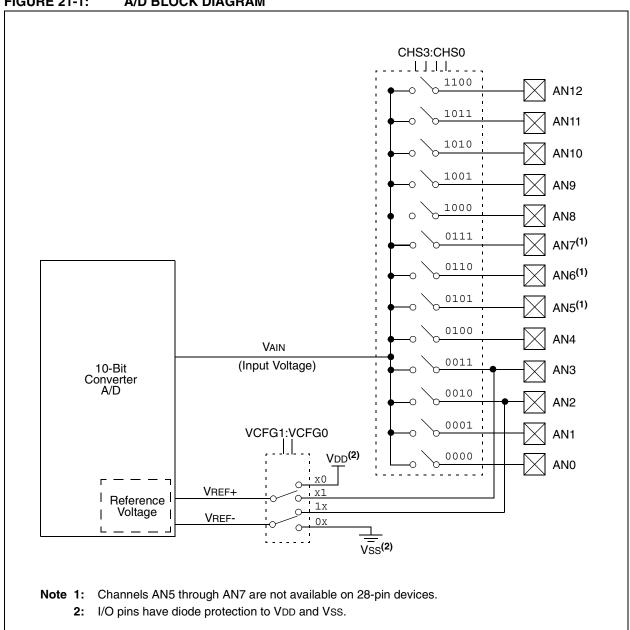
The A/D converter has a unique feature of being able to operate while the device is in Sleep mode. To operate in Sleep, the A/D conversion clock must be derived from the A/D's internal RC oscillator.

The output of the sample and hold is the input into the converter, which generates the result via successive approximation.

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.

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 register) is cleared and A/D Interrupt Flag bit, ADIF, is set. The block diagram of the A/D module is shown in Figure 21-1.

**FIGURE 21-1:** A/D BLOCK DIAGRAM



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

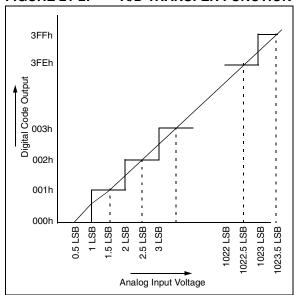
After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as an input. To determine acquisition time, see **Section 21.1** "A/D Acquisition Requirements". After 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 perform an A/D conversion:

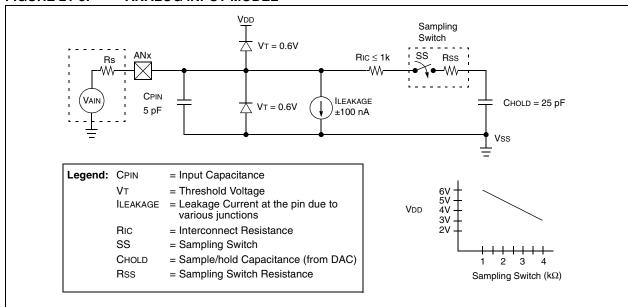
- 1. Configure the A/D module:
  - Configure analog pins, voltage reference and digital I/O (ADCON1)
  - Select A/D input channel (ADCON0)
  - Select A/D acquisition time (ADCON2)
  - Select A/D conversion clock (ADCON2)
  - Turn on A/D module (ADCON0)
- 2. Configure A/D interrupt (if desired):
  - · 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 register)

- 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
- 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 3 TAD is required before the next acquisition starts.









### 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 21-3. 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 k $\Omega$ . 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.

Example 21-3 shows the calculation of the minimum required acquisition time TACQ. This calculation is based on the following application system assumptions:

 $\begin{array}{lll} \text{CHOLD} & = & 25 \text{ pF} \\ \text{Rs} & = & 2.5 \text{ k}\Omega \\ \text{Conversion Error} & \leq & 1/2 \text{ LSb} \end{array}$ 

VDD =  $5V \rightarrow RSS = 2 k\Omega$ Temperature =  $85^{\circ}C$  (system max.)

#### **EQUATION 21-1: ACQUISITION TIME**

```
TACQ = Amplifier Settling Time + Holding Capacitor Charging Time + Temperature Coefficient

= TAMP + TC + TCOFF
```

### **EQUATION 21-2: A/D MINIMUM CHARGING TIME**

```
\begin{array}{lll} V_{HOLD} & = & (V_{REF} - (V_{REF}/2048)) \bullet (1 - e^{(-T_{C}/C_{HOLD}(R_{IC} + R_{SS} + R_{S}))}) \\ \text{or} \\ T_{C} & = & -(C_{HOLD})(R_{IC} + R_{SS} + R_{S}) \ln(1/2048) \end{array}
```

#### **EQUATION 21-3: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME**

```
TACO
                     TAMP + TC + TCOFF
TAMP
                     0.2 \mu s
TCOFF
                     (Temp - 25^{\circ}C)(0.02 \mu s/^{\circ}C)
                     (85^{\circ}C - 25^{\circ}C)(0.02 \,\mu\text{s}/^{\circ}C)
                     1.2 \mu s
Temperature coefficient is only required for temperatures > 25°C. Below 25°C, TCOFF = 0 ms.
TC
                     -(CHOLD)(RIC + RSS + RS) ln(1/2048) \mu s
                     -(25 \text{ pF}) (1 \text{ k}\Omega + 2 \text{ k}\Omega + 2.5 \text{ k}\Omega) \ln(0.0004883) \text{ }\mu\text{s}
                     1.05 \mu s
TACQ
                     0.2 \mu s + 1.05 \mu s + 1.2 \mu s
                     2.45 \mu s
```

# 21.2 Selecting and Configuring Acquisition Time

The ADCON2 register allows the user to select an acquisition time that occurs each time the GO/DONE bit is set. It also gives users the option to use an automatically determined acquisition time.

Acquisition time may be set with the ACQT2:ACQT0 bits (ADCON2<5:3>) which provide a range of 2 to 20 TAD. 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.

Manual acquisition is selected when ACQT2:ACQT0 = 000. When the GO/ $\overline{DONE}$  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  $\overline{GO/DONE}$  bit. This option is also the default Reset state of the ACQT2:ACQT0 bits and is compatible with devices that do not offer programmable acquisition times.

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 28-29 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 So	ource (TAD)	Maximum Dev	ice Frequency
Operation	ADCS2:ADCS0	PIC18FXXXX	PIC18LFXXXX <sup>(4)</sup>
2 Tosc	000	2.86 MHz	1.43 MHz
4 Tosc	100	5.71 MHz	2.86 MHz
8 Tosc	001	11.43 MHz	5.72 MHz
16 Tosc	101	22.86 MHz	11.43 MHz
32 Tosc	010	45.71 MHz	22.86 MHz
64 Tosc	110	48.0 MHz	45.71 MHz
RC <sup>(3)</sup>	x11	1.00 MHz <sup>(1)</sup>	1.00 MHz <sup>(2)</sup>

- Note 1: The RC source has a typical TAD time of 4 ms.
  - 2: The RC source has a typical TAD time of 6 ms.
  - 3: For device frequencies above 1 MHz, the device must be in Sleep for the entire conversion or the A/D accuracy may be out of specification.
  - 4: Low-power devices only.

# 21.4 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 ADCON2 should be updated in accordance with the clock source to be used in that mode. After entering the mode, an A/D acquisition or conversion may be started. Once started, the device should continue to be clocked by the same clock source until the conversion has been completed.

If desired, the device may be placed into the corresponding Idle mode during the conversion. If the device 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 FRC 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 bit (OSCCON<7>) must have already been cleared prior to starting the conversion.

## 21.5 Configuring Analog Port Pins

The ADCON1, TRISA, TRISB and TRISE registers all configure 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 as analog inputs. 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.
  - **3:** The PBADEN bit in Configuration Register 3H configures PORTB pins to reset as analog or digital pins by controlling how the PCFG0 bits in ADCON1 are reset.

#### 21.6 A/D Conversions

Figure 21-4 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 21-5 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).

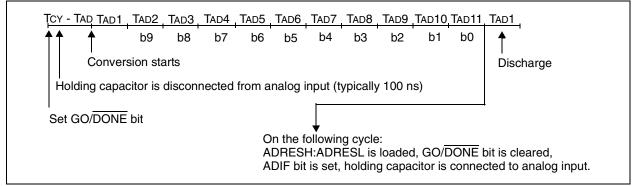
After the A/D conversion is completed or aborted, a 2 TAD wait is required before the next acquisition can be started. After this wait, acquisition on the selected channel is automatically started.

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

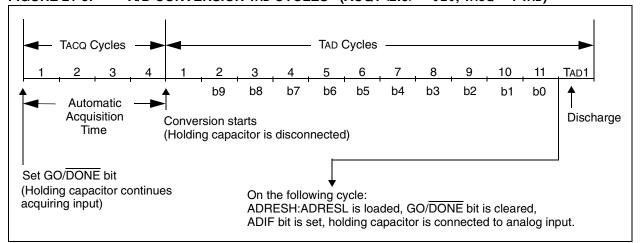
### 21.7 Discharge

The discharge phase is used to initialize the value of the capacitor array. The array is discharged before every sample. This feature helps to optimize the unity-gain amplifier as the circuit always needs to charge the capacitor array, rather than charge/discharge based on previous measurement values.





### FIGURE 21-5: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 010, TACQ = 4 TAD)



### 21.8 Use of the CCP2 Trigger

An A/D conversion can be started by the Special Event Trigger of the CCP2 module. This requires that the CCP2M3:CCP2M0 bits (CCP2CON<3:0>) be programmed as '1011' and that the A/D module is enabled (ADON bit is set). When the trigger occurs, the GO/DONE bit will be set, starting the A/D acquisition and conversion and the Timer1 (or Timer3) counter will be reset to zero. Timer1 (or Timer3) is reset to automatically repeat the A/D acquisition period with minimal software overhead (moving ADRESH:ADRESL to the

desired location). The appropriate analog input channel must be selected and the minimum acquisition period is either timed by the user, or an appropriate TACQ time selected before the Special Event Trigger sets the GO/DONE bit (starts a conversion).

If the A/D module is not enabled (ADON is cleared), the Special Event Trigger will be ignored by the A/D module but will still reset the Timer1 (or Timer3) counter.

TABLE 21-2: REGISTERS ASSOCIATED WITH A/D 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	51
PIR1	SPPIF <sup>(4)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54
PIE1	SPPIE <sup>(4)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54
IPR1	SPPIP <sup>(4)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54
PIR2	OSCFIF	CMIF	USBIF	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	54
PIE2	OSCFIE	CMIE	USBIE	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	54
IPR2	OSCFIP	CMIP	USBIP	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	54
ADRESH	A/D Result Register High Byte							52	
ADRESL	A/D Result	Register Lov	w Byte						52
ADCON0	_		CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	52
ADCON1	_	_	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	52
ADCON2	ADFM	_	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	52
PORTA	_	RA6 <sup>(2)</sup>	RA5	RA4	RA3	RA2	RA1	RA0	54
TRISA	_	TRISA6 <sup>(2)</sup>	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	54
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	54
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	54
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	54
PORTE	RDPU <sup>(4)</sup>	_	_	_	RE3 <sup>(1,3)</sup>	RE2 <sup>(4)</sup>	RE1 <sup>(4)</sup>	RE0 <sup>(4)</sup>	54
TRISE <sup>(4)</sup>	_	_	_	_	_	TRISE2	TRISE1	TRISE0	54
LATE <sup>(4)</sup>	_		_	_	_	LATE2	LATE1	LATE0	54

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.

Note 1: Implemented only when Master Clear functionality is disabled (MCLRE Configuration bit = 0).

- **2:** RA6 and its associated latch and data direction bits are enabled as I/O pins based on oscillator configuration; otherwise, they are read as '0'.
- **3:** RE3 port bit is available only as an input pin when the MCLRE Configuration bit is '0'.
- **4:** These registers and/or bits are not implemented on 28-pin devices.

### 22.0 COMPARATOR MODULE

The analog comparator module contains two comparators that can be configured in a variety of ways. The inputs can be selected from the analog inputs multiplexed with pins RA0 through RA5, as well as the on-chip voltage reference (see **Section 23.0 "Comparator Voltage Reference Module"**). The digital outputs (normal or inverted) are available at the pin level and can also be read through the control register.

The CMCON register (Register 22-1) selects the comparator input and output configuration. Block diagrams of the various comparator configurations are shown in Figure 22-1.

#### REGISTER 22-1: CMCON: COMPARATOR CONTROL REGISTER

R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1
C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
bit 7							bit 0

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

bit 7 **C2OUT**: Comparator 2 Output bit

When C2INV = 0: 1 = C2 VIN+ > C2 VIN-0 = C2 VIN+ < C2 VIN-When C2INV = 1:

1 = C2 VIN+ < C2 VIN-0 = C2 VIN+ > C2 VIN-

bit 6 C10UT: Comparator 1 Output bit

When C1INV = 0: 1 = C1 VIN+ > C1 VIN-0 = C1 VIN+ < C1 VIN-When C1INV = 1: 1 = C1 VIN+ < C1 VIN-0 = C1 VIN+ > C1 VIN-

bit 5 **C2INV**: Comparator 2 Output Inversion bit

1 = C2 output inverted0 = C2 output not inverted

bit 4 C1INV: Comparator 1 Output Inversion bit

1 = C1 output inverted0 = C1 output not inverted

bit 3 CIS: Comparator Input Switch bit

When CM2:CM0 = 110:

1 = C1 Vin- connects to RA3/AN3/VREF+
C2 Vin- connects to RA2/AN2/VREF-/CVREF

0 = C1 VIN- connects to RA0/AN0 C2 VIN- connects to RA1/AN1

bit 2-0 CM2:CM0: Comparator Mode bits

Figure 22-1 shows the Comparator modes and the CM2:CM0 bit settings.

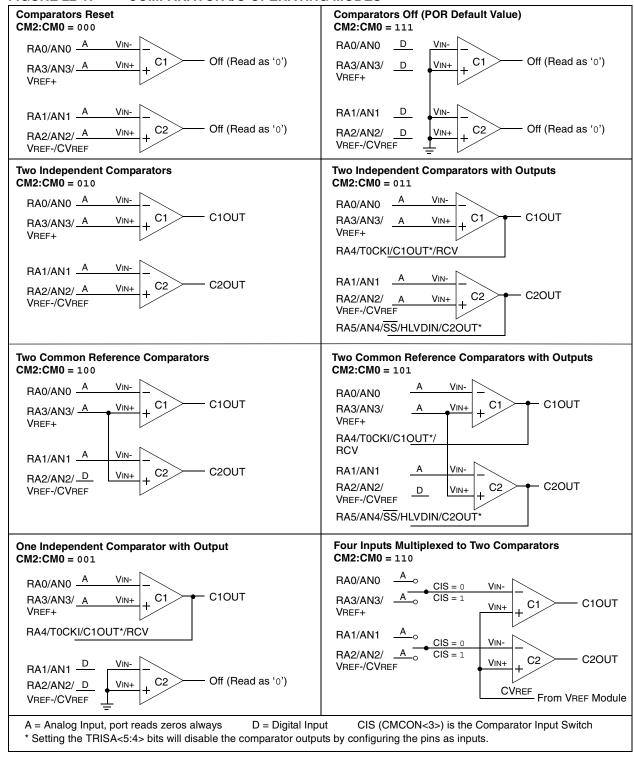
## 22.1 Comparator Configuration

There are eight modes of operation for the comparators, shown in Figure 22-1. Bits CM2:CM0 of the CMCON register are used to select these modes. The TRISA register controls the data direction of the comparator pins for each mode. If the Comparator

mode is changed, the comparator output level may not be valid for the specified mode change delay shown in **Section 28.0 "Electrical Characteristics"**.

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

#### FIGURE 22-1: COMPARATOR I/O OPERATING MODES



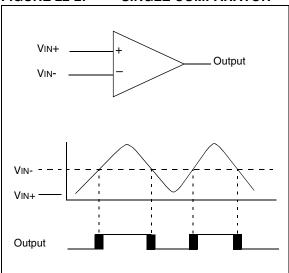
### 22.2 Comparator Operation

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 VIN+ is less than the analog input VIN-, the output of the comparator is a digital low level. When the analog input at VIN+ is greater than the analog input VIN-, 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.

### 22.3 Comparator Reference

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 (Figure 22-2).

FIGURE 22-2: SINGLE COMPARATOR



#### 22.3.1 EXTERNAL REFERENCE SIGNAL

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

#### 22.3.2 INTERNAL REFERENCE SIGNAL

The comparator module also allows the selection of an internally generated voltage reference from the comparator voltage reference module. This module is described in more detail in **Section 23.0 "Comparator Voltage Reference Module"**.

The internal reference is only available in the mode where four inputs are multiplexed to two comparators (CM2:CM0 = 110). In this mode, the internal voltage reference is applied to the Vin+ pin of both comparators.

### 22.4 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output has a valid level. If the internal reference is changed, the maximum delay of the internal voltage reference must be considered when using the comparator outputs. Otherwise, the maximum delay of the comparators should be used (see **Section 28.0** "**Electrical Characteristics**").

## 22.5 Comparator Outputs

The comparator outputs are read through the CMCON register. These bits are read-only. The comparator outputs may also be directly output to the RA4 and RA5 I/O pins. When enabled, multiplexors in the output path of the RA4 and RA5 pins will switch and the output of each pin will be the unsynchronized output of the comparator. The uncertainty of each of the comparators is related to the input offset voltage and the response time given in the specifications. Figure 22-3 shows the comparator output block diagram.

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

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

- Note 1: When reading the PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert an analog input according to the Schmitt Trigger input specification.
  - 2: Analog levels on any pin defined as a digital input may cause the input buffer to consume more current than is specified.

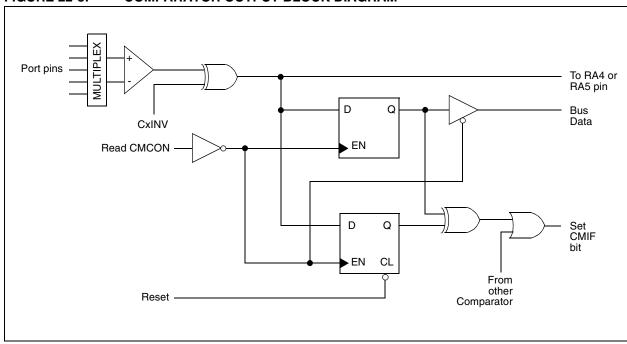


FIGURE 22-3: COMPARATOR OUTPUT BLOCK DIAGRAM

### 22.6 Comparator Interrupts

The comparator interrupt flag is set whenever there is a change in the output value of either comparator. Software will need to maintain information about the status of the output bits, as read from CMCON<7:6>, to determine the actual change that occurred. The CMIF bit (PIR2<6>) is the Comparator Interrupt Flag. The CMIF bit must be reset by clearing it. Since it is also possible to write a '1' to this register, a simulated interrupt may be initiated.

Both the CMIE bit (PIE2<6>) 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 CMIF bit will still be set if an interrupt condition occurs.

Note: If a change in the CMCON register (C1OUT or C2OUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CMIF (PIR2<6>) interrupt flag may not get set.

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

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

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

# 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, as shown in the comparator specifications. To minimize power consumption while in Sleep mode, turn off the comparators (CM2:CM0 = 111) before entering Sleep. If the device wakes up from Sleep, the contents of the CMCON register are not affected.

### 22.8 Effects of a Reset

A device Reset forces the CMCON register to its Reset state, causing the comparator modules to be turned off (CM2:CM0 = 111). However, the input pins (RA0 through RA3) are configured as analog inputs by default on device Reset. The I/O configuration for these pins is determined by the setting of the PCFG3:PCFG0 bits (ADCON1<3:0>). Therefore, device current is minimized when analog inputs are present at Reset time.

# 22.9 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 22-4. Since the analog pins are connected to a digital output, they have reverse biased diodes to VDD and Vss. The analog input, therefore, must be between Vss and VDD. If the input voltage deviates from this

range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up condition may occur. A maximum source impedance of 10  $k\Omega$  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 22-4: COMPARATOR ANALOG INPUT MODEL

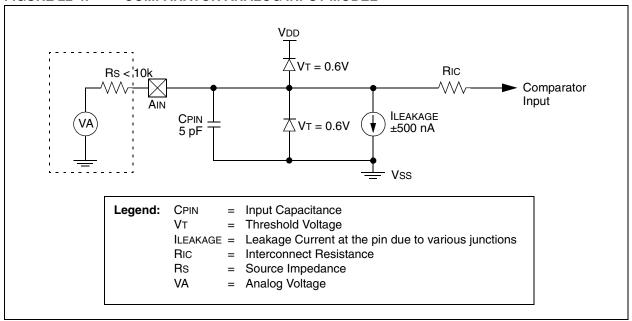


TABLE 22-1: REGISTERS ASSOCIATED WITH COMPARATOR MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	53
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	53
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	51
PIR2	OSCFIF	CMIF	USBIF	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	54
PIE2	OSCFIE	CMIE	USBIE	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	54
IPR2	OSCFIP	CMIP	USBIP	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	54
PORTA	_	RA6 <sup>(1)</sup>	RA5	RA4	RA3	RA2	RA1	RA0	54
LATA	_	LATA6 <sup>(1)</sup>	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	54
TRISA	_	TRISA6 <sup>(1)</sup>	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	54

**Legend:** — = unimplemented, read as '0'. Shaded cells are unused by the comparator module.

**Note 1:** PORTA<6> and its direction and latch bits are individually configured as port pins based on various oscillator modes. When disabled, these bits read as '0'.

© 2006 Microchip Technology Inc. Preliminary DS39632C-page 273

NOTES:

# 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.

# 23.1 Configuring the Comparator Voltage Reference

The 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) x CVRSRC
If CVRR = 0:
CVREF = (CVRSRC/4) + (((CVR3:CVR0)/32) x
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 28-3 in Section 28.0 "Electrical Characteristics").

#### 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 <sup>(1)</sup>	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit	t, 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
              CVROE: Comparator VREF Output Enable bit(1)
bit 6
              1 = CVREF voltage level is also output on the RA2/AN2/VREF-/CVREF pin
              0 = CVREF voltage is disconnected from the RA2/AN2/VREF-/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 = VDD - VSS
bit 3-0
              CVR3:CVR0: Comparator VREF Value Selection bits (0 \le (CVR3:CVR0) \le 15)
              When CVRR = 1:
              CVREF = ((CVR3:CVR0)/24) • (CVRSRC)
              When CVRR = 0:
              CVREF = (CVRSRC/4) + ((CVR3:CVR0)/32) • (CVRSRC)
```

Note 1: CVROE overrides the TRISA<2> bit setting.

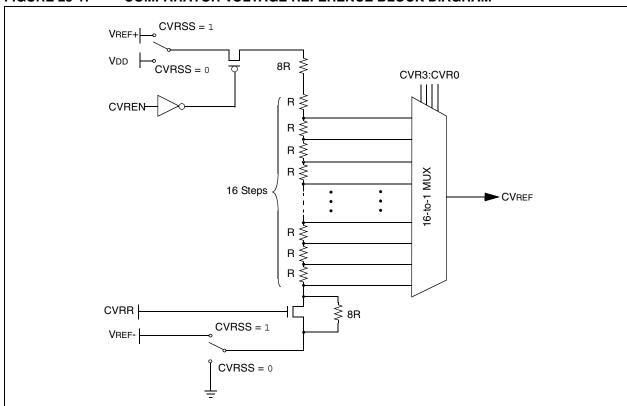


FIGURE 23-1: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM

### 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 28.0** "Electrical Characteristics".

### 23.3 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.

#### 23.4 Effects of a Reset

A device Reset disables the voltage reference by clearing bit, CVREN (CVRCON<7>). This Reset also disconnects the reference from the RA2 pin by clearing bit, CVROE (CVRCON<6>) and selects the high-voltage range by clearing bit, CVRR (CVRCON<5>). The CVR value select bits are also cleared.

#### 23.5 Connection Considerations

The voltage reference module operates independently of the comparator module. The output of the reference generator may be connected to the RA2 pin if the TRISA<2> bit and the CVROE bit are both set. Enabling the voltage reference output onto RA2 when it is configured as a digital input will increase current consumption. Connecting RA2 as a digital output with CVRSS enabled will also increase current consumption.

The RA2 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.

### FIGURE 23-2: COMPARATOR VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE

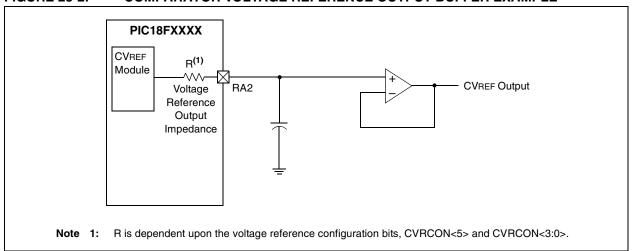


TABLE 23-1: REGISTERS ASSOCIATED WITH COMPARATOR VOLTAGE REFERENCE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	53
CMCON	C2OUT	C10UT	C2INV	C1INV	CIS	CM2	CM1	CM0	53
TRISA	_	TRISA6 <sup>(1)</sup>	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	54

**Legend:** Shaded cells are not used with the comparator voltage reference.

**Note 1:** PORTA<6> and its direction and latch bits are individually configured as port pins based on various oscillator modes. When disabled, these bits read as '0'.

© 2006 Microchip Technology Inc. Preliminary DS39632C-page 277

NOTES:

U = Unimplemented bit, read as '0'

# 24.0 HIGH/LOW-VOLTAGE DETECT (HLVD)

PIC18F2455/2550/4455/4550 devices have a High/Low-Voltage Detect module (HLVD). This is a programmable circuit that allows the user to specify both a device voltage trip point and the direction of change from that point. If the device experiences an excursion past the trip point in that direction, an interrupt flag is set. If the interrupt is enabled, the program execution will branch to the interrupt vector address and the software can then respond to the interrupt.

Legend:

R = Readable bit

The High/Low-Voltage Detect Control register (Register 24-1) completely controls the operation of the HLVD module. This allows the circuitry to be "turned off" by the user under software control which minimizes the current consumption for the device.

The block diagram for the HLVD module is shown in Figure 24-1.

#### REGISTER 24-1: HLVDCON: HIGH/LOW-VOLTAGE DETECT CONTROL REGISTER

W = Writable bit

R/W-0	U-0	R-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1
VDIRMAG	_	IRVST	HLVDEN	HLVDL3 <sup>(1)</sup>	HLVDL2 <sup>(1)</sup>	HLVDL1 <sup>(1)</sup>	HLVDL0 <sup>(1)</sup>
bit 7							bit 0

-n = Value at F	POR "	1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown
bit 7	VDIRMAG: Vol	tage Direction Magn	nitude Select bit	
			als or exceeds trip point (HLVDI als or falls below trip point (HLV	
bit 6	Unimplemente	d: Read as '0'		
bit 5	IRVST: Internal	Reference Voltage	Stable Flag bit	
	0 = Indicates th	nat the voltage dete		flag at the specified voltage range errupt flag at the specified voltage
bit 4	HLVDEN: High	Low-Voltage Detect	Power Enable bit	
	<ul><li>1 = HLVD enak</li><li>0 = HLVD disal</li></ul>			
bit 3-0	HLVDL3:HLVD	L0: Voltage Detection	on Limit bits <sup>(1)</sup>	
	1111 = Externa 1110 = Maximu	• .	ed (input comes from the HLVDII	N pin)
	•			
	•			
	0000 = Minimu	m setting		

Note 1: See Table 28-6 in Section 28.0 "Electrical Characteristics" for specifications.

The module is enabled by setting the HLVDEN bit. Each time that the HLVD module is enabled, the circuitry requires some time to stabilize. The IRVST bit is a read-only bit and is used to indicate when the circuit is stable. The module can only generate an interrupt after the circuit is stable and IRVST is set.

The VDIRMAG bit determines the overall operation of the module. When VDIRMAG is cleared, the module monitors for drops in VDD below a predetermined set point. When the bit is set, the module monitors for rises in VDD above the set point.

### 24.1 Operation

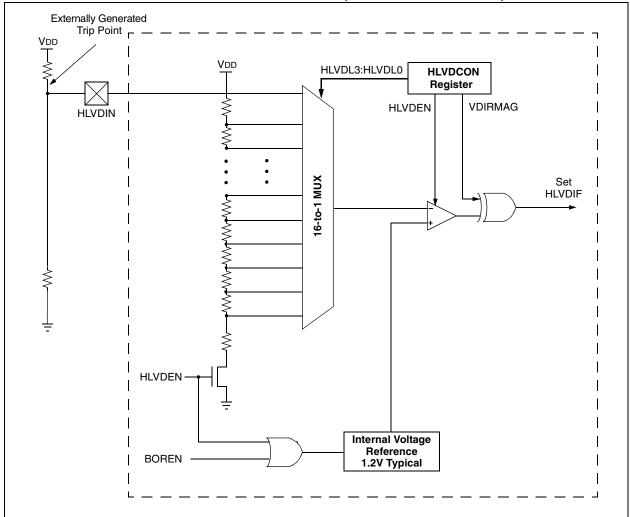
When the HLVD module is enabled, a comparator uses an internally generated reference voltage as the set point. The set point is compared with the trip point, where each node in the resistor divider represents a trip point voltage. The "trip point" voltage is the voltage level at which the device detects a high or low-voltage

event, depending on the configuration of the module. When the supply voltage is equal to the trip point, the voltage tapped off of the resistor array is equal to the internal reference voltage generated by the voltage reference module. The comparator then generates an interrupt signal by setting the HLVDIF bit.

The trip point voltage is software programmable to any one of 16 values. The trip point is selected by programming the HLVDL3:HLVDL0 bits (HLVDCON<3:0>).

The HLVD module has an additional feature that allows the user to supply the trip voltage to the module from an external source. This mode is enabled when bits, HLVDL3:HLVDL0, are set to '1111'. In this state, the comparator input is multiplexed from the external input pin, HLVDIN. This gives users flexibility because it allows them to configure the High/Low-Voltage Detect interrupt to occur at any voltage in the valid operating range.

FIGURE 24-1: HLVD MODULE BLOCK DIAGRAM (WITH EXTERNAL INPUT)



#### 24.2 HLVD Setup

The following steps are needed to set up the HLVD module:

- Disable the module by clearing the HLVDEN bit (HLVDCON<4>).
- 2. Write the value to the HLVDL3:HLVDL0 bits that selects the desired HLVD trip point.
- Set the VDIRMAG bit to detect high voltage (VDIRMAG = 1) or low voltage (VDIRMAG = 0).
- 4. Enable the HLVD module by setting the HLVDEN bit.
- Clear the HLVD Interrupt Flag, HLVDIF (PIR2<2>), which may have been set from a previous interrupt.
- Enable the HLVD interrupt, if interrupts are desired, by setting the HLVDIE and GIE/GIEH bits (PIE2<2> and INTCON<7>). An interrupt will not be generated until the IRVST bit is set.

### 24.3 Current Consumption

When the module is enabled, the HLVD comparator and voltage divider are enabled and will consume static current. The total current consumption, when enabled, is specified in electrical specification parameter D022 (Section 28.2 "DC Characteristics").

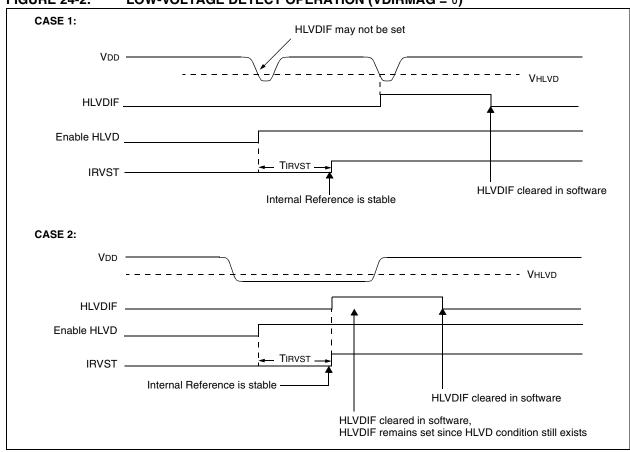
Depending on the application, the HLVD module does not need to be operating constantly. To decrease the current requirements, the HLVD circuitry may only need to be enabled for short periods where the voltage is checked. After doing the check, the HLVD module may be disabled.

### 24.4 HLVD Start-up Time

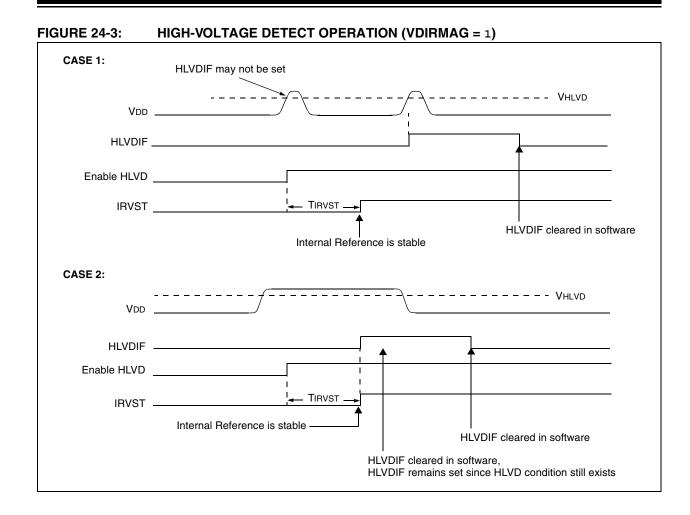
The internal reference voltage of the HLVD module, specified in electrical specification parameter D420 (see Table 28-6 in **Section 28.0 "Electrical Characteristics"**), may be used by other internal circuitry, such as the Programmable Brown-out Reset. If the HLVD or other circuits using the voltage reference are disabled to lower the device's current consumption, the reference voltage circuit will require time to become stable before a low or high-voltage condition can be reliably detected. This start-up time, TIRVST, is an interval that is independent of device clock speed. It is specified in electrical specification parameter 36 (Table 28-12).

The HLVD interrupt flag is not enabled until TIRVST has expired and a stable reference voltage is reached. For this reason, brief excursions beyond the set point may not be detected during this interval. Refer to Figure 24-2 or Figure 24-3.

FIGURE 24-2: LOW-VOLTAGE DETECT OPERATION (VDIRMAG = 0)



© 2006 Microchip Technology Inc. Preliminary DS39632C-page 281

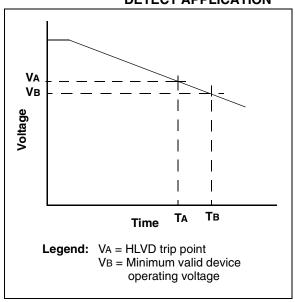


## 24.5 Applications

In many applications, the ability to detect a drop below or rise above a particular threshold is desirable. For example, the HLVD module could be periodically enabled to detect Universal Serial Bus (USB) attach or detach. This assumes the device is powered by a lower voltage source than the USB when detached. An attach would indicate a high-voltage detect from, for example, 3.3V to 5V (the voltage on USB) and vice versa for a detach. This feature could save a design a few extra components and an attach signal (input pin).

For general battery applications, Figure 24-4 shows a possible voltage curve. Over time, the device voltage decreases. When the device voltage reaches voltage, VA, the HLVD logic generates an interrupt at time, TA. The interrupt could cause the execution of an ISR, which would allow the application to perform "house-keeping tasks" and perform a controlled shutdown before the device voltage exits the valid operating range at TB. The HLVD, thus, would give the application a time window, represented by the difference between TA and TB, to safely exit.

FIGURE 24-4: TYPICAL
HIGH/LOW-VOLTAGE
DETECT APPLICATION



## 24.6 Operation During Sleep

When enabled, the HLVD circuitry continues to operate during Sleep. If the device voltage crosses the trip point, the HLVDIF bit will be set and the device will wake-up from Sleep. Device execution will continue from the interrupt vector address if interrupts have been globally enabled.

#### 24.7 Effects of a Reset

A device Reset forces all registers to their Reset state. This forces the HLVD module to be turned off.

TABLE 24-1: REGISTERS ASSOCIATED WITH HIGH/LOW-VOLTAGE DETECT MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
HLVDCON	VDIRMAG	_	IRVST	HLVDEN	HLVDL3	HLVDL2	HLVDL1	HLVDL0	52
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	51
PIR2	OSCFIF	CMIF	USBIF	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	54
PIE2	OSCFIE	CMIE	USBIE	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	54
IPR2	OSCFIP	CMIP	USBIP	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	54

**Legend:** — = unimplemented, read as '0'. Shaded cells are unused by the HLVD module.

NOTES:

# 25.0 SPECIAL FEATURES OF THE CPU

PIC18F2455/2550/4455/4550 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
- · ID Locations
- · 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, PIC18F2455/2550/4455/4550 devices have a Watchdog Timer, which is either permanently enabled via the Configuration bits or software controlled (if configured as disabled).

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.

### 25.1 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.

The user will note that address 300000h is beyond the user program memory space. In fact, it belongs to the configuration memory space (300000h-3FFFFFh), which can only be accessed using table reads and table writes.

Programming the Configuration registers is done in a manner similar to programming the Flash memory. The WR bit in the EECON1 register starts a self-timed write to the Configuration register. In normal operation mode, a TBLWT instruction, with the TBLPTR pointing to the Configuration register, sets up the address and the data for the Configuration register write. Setting the WR bit starts a long write to the Configuration register. The Configuration registers are written a byte at a time. To write or erase a configuration cell, a TBLWT instruction can write a '1' or a '0' into the cell. For additional details on Flash programming, refer to Section 6.5 "Writing to Flash Program Memory".

TABLE 25-1: 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
300000h	CONFIG1L	_	_	USBDIV	CPUDIV1	CPUDIV0	PLLDIV2	PLLDIV1	PLLDIV0	00 0000
300001h	CONFIG1H	IESO	FCMEN	_	-	FOSC3	FOSC2	FOSC1	FOSC0	00 0101
300002h	CONFIG2L	_	_	VREGEN	BORV1	BORV0	BOREN1	BOREN0	PWRTEN	01 1111
300003h	CONFIG2H	_	_	_	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN	1 1111
300005h	CONFIG3H	MCLRE	_	_	_	_	LPT1OSC	PBADEN	CCP2MX	1011
300006h	CONFIG4L	DEBUG	XINST	ICPRT <sup>(3)</sup>	_	_	LVP	_	STVREN	1001-1
300008h	CONFIG5L	_	_	_	_	CP3 <sup>(1)</sup>	CP2	CP1	CP0	1111
300009h	CONFIG5H	CPD	СРВ	_	_	_	_	_	_	11
30000Ah	CONFIG6L	_	_	_	_	WRT3 <sup>(1)</sup>	WRT2	WRT1	WRT0	1111
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	_	_	_	_	_	111
30000Ch	CONFIG7L		_	_	_	EBTR3 <sup>(1)</sup>	EBTR2	EBTR1	EBTR0	1111
30000Dh	CONFIG7H	-	EBTRB	_	_	_	_			-1
3FFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	XXXX XXXX <sup>(2)</sup>
3FFFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	0001 0010 <sup>(2)</sup>

**Legend:** x = unknown, u = unchanged, - = unimplemented. Shaded cells are unimplemented, read as '0'.

Note 1: Unimplemented in PIC18FX455 devices; maintain this bit set.

<sup>2:</sup> See Register 25-13 and Register 25-14 for DEVID values. DEVID registers are read-only and cannot be programmed by the user.

<sup>3:</sup> Available only on PIC18F4455/4550 devices in 44-pin TQFP packages. Always leave this bit clear in all other devices.

### REGISTER 25-1: CONFIG1L: CONFIGURATION REGISTER 1 LOW (BYTE ADDRESS 300000h)

U-0	U-0	R/P-0	R/P-0	R/P-0	R/P-0	R/P-0	R/P-0
_	_	USBDIV	CPUDIV1	CPUDIV0	PLLDIV2	PLLDIV1	PLLDIV0
bit 7							bit 0

Legend:

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

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

bit 5 USBDIV: USB Clock Selection bit (used in Full-Speed USB mode only; UCFG:FSEN = 1)

1 = USB clock source comes from the 96 MHz PLL divided by 2

0 = USB clock source comes directly from the primary oscillator block with no postscale

bit 4-3 CPUDIV1:CPUDIV0: System Clock Postscaler Selection bits

For XT, HS, EC and ECIO Oscillator modes:

11 = Primary oscillator divided by 4 to derive system clock

10 = Primary oscillator divided by 3 to derive system clock

01 = Primary oscillator divided by 2 to derive system clock

00 = Primary oscillator used directly for system clock (no postscaler)

For XTPLL, HSPLL, ECPLL and ECPIO Oscillator modes:

11 = 96 MHz PLL divided by 6 to derive system clock

10 = 96 MHz PLL divided by 4 to derive system clock

01 = 96 MHz PLL divided by 3 to derive system clock

00 = 96 MHz PLL divided by 2 to derive system clock

bit 2-0 PLLDIV2:PLLDIV0: PLL Prescaler Selection bits

111 = Divide by 12 (48 MHz oscillator input)

110 = Divide by 10 (40 MHz oscillator input)

101 = Divide by 6 (24 MHz oscillator input)

100 = Divide by 5 (20 MHz oscillator input)

011 = Divide by 4 (16 MHz oscillator input)

010 = Divide by 3 (12 MHz oscillator input)

001 = Divide by 2 (8 MHz oscillator input)

000 = No prescale (4 MHz oscillator input drives PLL directly)

### REGISTER 25-2: CONFIG1H: CONFIGURATION REGISTER 1 HIGH (BYTE ADDRESS 300001h)

R/P-0	R/P-0	U-0	U-0	R/P-0	R/P-1	R/P-0	R/P-1
IESO	FCMEN	_	_	FOSC3 <sup>(1)</sup>	FOSC2 <sup>(1)</sup>	FOSC1 <sup>(1)</sup>	FOSC0 <sup>(1)</sup>
bit 7							bit 0

Legend:

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

bit 7 IESO: Internal/External Oscillator Switchover bit

1 = Oscillator Switchover mode enabled

0 = Oscillator Switchover mode disabled

bit 6 FCMEN: Fail-Safe Clock Monitor Enable bit

1 = Fail-Safe Clock Monitor enabled0 = Fail-Safe Clock Monitor disabled

bit 5-4 Unimplemented: Read as '0'

bit 3-0 **FOSC3:FOSC0:** Oscillator Selection bits<sup>(1)</sup>

111x = HS oscillator, PLL enabled (HSPLL)

110x = HS oscillator (HS)

1011 = Internal oscillator, HS oscillator used by USB (INTHS)

1010 = Internal oscillator, XT used by USB (INTXT)

1001 = Internal oscillator, CLKO function on RA6, EC used by USB (INTCKO)

1000 = Internal oscillator, port function on RA6, EC used by USB (INTIO)

0111 = EC oscillator, PLL enabled, CLKO function on RA6 (ECPLL)

0110 = EC oscillator, PLL enabled, port function on RA6 (ECPIO)

0101 = EC oscillator, CLKO function on RA6 (EC)

0100 = EC oscillator, port function on RA6 (ECIO)

001x = XT oscillator, PLL enabled (XTPLL)

000x = XT oscillator (XT)

Note 1: The microcontroller and USB module both use the selected oscillator as their clock source in XT, HS and EC modes. The USB module uses the indicated XT, HS or EC oscillator as its clock source whenever the microcontroller uses the internal oscillator.

### REGISTER 25-3: CONFIG2L: CONFIGURATION REGISTER 2 LOW (BYTE ADDRESS 300002h)

U-0	U-0	R/P-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
_	_	VREGEN	BORV1 <sup>(1)</sup>	BORV0 <sup>(1)</sup>	BOREN1 <sup>(2)</sup>	BORENO <sup>(2)</sup>	PWRTEN <sup>(2)</sup>
bit 7							bit 0

Legend:

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

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

bit 5 VREGEN: USB Internal Voltage Regulator Enable bit

1 = USB voltage regulator enabled0 = USB voltage regulator disabled

bit 4-3 BORV1:BORV0: Brown-out Reset Voltage bits<sup>(1)</sup>

11 = Minimum setting

•

00 = Maximum setting

bit 2-1 **BOREN1:BOREN0:** Brown-out Reset Enable bits<sup>(2)</sup>

11 = Brown-out Reset enabled in hardware only (SBOREN is disabled)

10 = Brown-out Reset enabled in hardware only and disabled in Sleep mode (SBOREN is disabled)

01 = Brown-out Reset enabled and controlled by software (SBOREN is enabled)

00 = Brown-out Reset disabled in hardware and software

bit 0 **PWRTEN:** Power-up Timer Enable bit<sup>(2)</sup>

1 = PWRT disabled

0 = PWRT enabled

Note 1: See Section 28.0 "Electrical Characteristics" for the specifications.

The Power-up Timer is decoupled from Brown-out Reset, allowing these features to be independently controlled.

## REGISTER 25-4: CONFIG2H: CONFIGURATION REGISTER 2 HIGH (BYTE ADDRESS 300003h)

U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
_	_	_	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN
bit 7							bit 0

Legend:

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

bit 7-5 **Unimplemented:** Read as '0'

bit 4-1 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 = 1:40001 = 1:2

0000 = 1:1

bit 0 WDTEN: Watchdog Timer Enable bit

1 = WDT enabled

0 = WDT disabled (control is placed on the SWDTEN bit)

### REGISTER 25-5: CONFIG3H: CONFIGURATION REGISTER 3 HIGH (BYTE ADDRESS 300005h)

R/P-1	U-0	U-0	U-0	U-0	R/P-0	R/P-1	R/P-1
MCLRE	_	_	_	_	LPT1OSC	PBADEN	CCP2MX
bit 7							bit 0

Legend:

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

bit 7 MCLRE: MCLR Pin Enable bit

 $1 = \overline{\text{MCLR}}$  pin enabled, RE3 input pin disabled  $0 = \overline{\text{RE3}}$  input pin enabled, MCLR pin disabled

bit 6-3 Unimplemented: Read as '0'

bit 2 LPT10SC: Low-Power Timer1 Oscillator Enable bit

1 = Timer1 configured for low-power operation0 = Timer1 configured for higher power operation

bit 1 PBADEN: PORTB A/D Enable bit

(Affects ADCON1 Reset state. ADCON1 controls PORTB<4:0> pin configuration.)

1 = PORTB<4:0> pins are configured as analog input channels on Reset

0 = PORTB<4:0> pins are configured as digital I/O on Reset

bit 0 CCP2MX: CCP2 MUX bit

1 = CCP2 input/output is multiplexed with RC10 = CCP2 input/output is multiplexed with RB3

### REGISTER 25-6: CONFIG4L: CONFIGURATION REGISTER 4 LOW (BYTE ADDRESS 300006h)

R/P-1	R/P-0	R/P-0	U-0	U-0	R/P-1	U-0	R/P-1
DEBUG	XINST	ICPRT <sup>(1)</sup>	-	_	LVP	_	STVREN
bit 7							bit 0

-n = Value when device is unprogrammed

Legend:

R = Readable bit

U = Unimplemented bit, read as '0'

u = Unchanged from programmed state

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 ICPRT: Dedicated In-Circuit Debug/Programming Port (ICPORT) Enable bit<sup>(1)</sup>

P = Programmable bit

1 = ICPORT enabled

0 = ICPORT disabled

bit 4-3 **Unimplemented:** Read as '0'

bit 2 LVP: Single-Supply ICSP™ Enable bit

1 = Single-Supply ICSP enabled0 = Single-Supply ICSP disabled

bit 1 Unimplemented: Read as '0'

bit 0 STVREN: Stack Full/Underflow Reset Enable bit

1 = Stack full/underflow will cause Reset

0 = Stack full/underflow will not cause Reset

**Note 1:** Available only on PIC18F4455/4550 devices in 44-pin TQFP packages. Always leave this bit clear in all other devices.

## REGISTER 25-7: CONFIG5L: CONFIGURATION REGISTER 5 LOW (BYTE ADDRESS 300008h)

U-0	U-0	U-0	U-0	R/C-1	R/C-1	R/C-1	R/C-1
_	_	_	-	CP3 <sup>(1)</sup>	CP2	CP1	CP0
bit 7							bit 0

Legend:

R = Readable bit C = Clearable bit U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed u = Unchanged from programmed state

bit 7-4 Unimplemented: Read as '0' bit 3 CP3: Code Protection bit<sup>(1)</sup>

1 = Block 3 (006000-007FFFh) is not code-protected 0 = Block 3 (006000-007FFFh) is code-protected

bit 2 CP2: Code Protection bit

1 = Block 2 (004000-005FFFh) is not code-protected 0 = Block 2 (004000-005FFFh) is code-protected

bit 1 CP1: Code Protection bit

1 = Block 1 (002000-003FFFh) is not code-protected 0 = Block 1 (002000-003FFFh) is code-protected

bit 0 **CP0:** Code Protection bit

1 = Block 0 (000800-001FFFh) is not code-protected 0 = Block 0 (000800-001FFFh) is code-protected

Note 1: Unimplemented in PIC18FX455 devices; maintain this bit set.

### REGISTER 25-8: CONFIG5H: CONFIGURATION REGISTER 5 HIGH (BYTE ADDRESS 300009h)

R/C-1	R/C-1	U-0	U-0	U-0	U-0	U-0	U-0
CPD	СРВ	_	_	_	_	_	_
bit 7							bit 0

Legend:

 $R = Readable \ bit$   $C = Clearable \ bit$   $U = Unimplemented \ bit, read \ as '0'$ -n = Value when device is unprogrammed  $u = Unchanged \ from \ programmed \ state$ 

bit 7 CPD: Data EEPROM Code Protection bit

1 = Data EEPROM is not code-protected 0 = Data EEPROM is code-protected

bit 6 CPB: Boot Block Code Protection bit

1 = Boot block (000000-0007FFh) is not code-protected 0 = Boot block (000000-0007FFh) is code-protected

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

### REGISTER 25-9: CONFIG6L: CONFIGURATION REGISTER 6 LOW (BYTE ADDRESS 30000Ah)

U-0	U-0	U-0	U-0	R/C-1	R/C-1	R/C-1	R/C-1
_	_	_	_	WRT3 <sup>(1)</sup>	WRT2	WRT1	WRT0
bit 7							bit 0

Legend:

R = Readable bit C = Clearable bit U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed u = Unchanged from programmed state

bit 7-4 **Unimplemented:** Read as '0' bit 3 **WRT3:** Write Protection bit<sup>(1)</sup>

1 = Block 3 (006000-007FFFh) is not write-protected 0 = Block 3 (006000-007FFFh) is write-protected

bit 2 WRT2: Write Protection bit

1 = Block 2 (004000-005FFFh) is not write-protected 0 = Block 2 (004000-005FFFh) is write-protected

bit 1 WRT1: Write Protection bit

1 = Block 1 (002000-003FFFh) is not write-protected 0 = Block 1 (002000-003FFFh) is write-protected

bit 0 WRT0: Write Protection bit

1 = Block 0 (000800-001FFFh) or (001000-001FFFh) is not write-protected 0 = Block 0 (000800-001FFFh) or (001000-001FFFh) is write-protected

Note 1: Unimplemented in PIC18FX455 devices; maintain this bit set.

## REGISTER 25-10: CONFIG6H: CONFIGURATION REGISTER 6 HIGH (BYTE ADDRESS 30000Bh)

R/C-1	R/C-1	R-1	U-0	U-0	U-0	U-0	U-0
WRTD	WRTB	WRTC <sup>(1)</sup>	_	_	_	_	_
bit 7							bit 0

Legend:

R = Readable bit C = Clearable bit U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed u = Unchanged from programmed state

bit 7 WRTD: Data EEPROM Write Protection bit

1 = Data EEPROM is not write-protected

0 = Data EEPROM is write-protected

bit 6 WRTB: Boot Block Write Protection bit

1 = Boot block (000000-0007FFh) is not write-protected 0 = Boot block (000000-0007FFh) is write-protected

bit 5 **WRTC:** Configuration Register Write Protection bit<sup>(1)</sup>

1 = Configuration registers (300000-3000FFh) are not write-protected 0 = Configuration registers (300000-3000FFh) are write-protected

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

Note 1: This bit is read-only in normal execution mode; it can be written only in Program mode.

### REGISTER 25-11: CONFIG7L: CONFIGURATION REGISTER 7 LOW (BYTE ADDRESS 30000Ch)

U-0	U-0	U-0	U-0	R/C-1	R/C-1	R/C-1	R/C-1
_	_	_	_	EBTR3 <sup>(1)</sup>	EBTR2	EBTR1	EBTR0
bit 7							bit 0

Legend:

R = Readable bit C = Clearable bit U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed u = Unchanged from programmed state

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **EBTR3:** Table Read Protection bit<sup>(1)</sup>

1 =Block 3 (006000-007FFFh) not protected from table reads executed in other blocks 0 =Block 3 (006000-007FFFh) protected from table reads executed in other blocks

bit 2 EBTR2: Table Read Protection bit

1 = Block 2 (004000-005FFFh) not protected from table reads executed in other blocks 0 = Block 2 (004000-005FFFh) protected from table reads executed in other blocks

bit 1 EBTR1: Table Read Protection bit

1 =Block 1 (002000-003FFFh) is not protected from table reads executed in other blocks 0 =Block 1 (002000-003FFFh) is protected from table reads executed in other blocks

bit 0 **EBTR0:** Table Read Protection bit

1 = Block 0 (000800-001FFFh) is not protected from table reads executed in other blocks 0 = Block 0 (000800-001FFFh) is protected from table reads executed in other blocks

Note 1: Unimplemented in PIC18FX455 devices; maintain this bit set.

## REGISTER 25-12: CONFIG7H: CONFIGURATION REGISTER 7 HIGH (BYTE ADDRESS 30000Dh)

U-0	R/C-1	U-0	U-0	U-0	U-0	U-0	U-0
_	EBTRB	_	_	_	_	_	_
bit 7							bit 0

Legend:

 $R = Readable \ bit$   $C = Clearable \ bit$   $U = Unimplemented \ bit, read \ as '0'$ -n = Value when device is unprogrammed  $u = Unchanged \ from \ programmed \ state$ 

bit 7 Unimplemented: Read as '0'

bit 6 EBTRB: Boot Block Table Read Protection bit

1 = Boot block (000000-0007FFh) is not protected from table reads executed in other blocks 0 = Boot block (000000-0007FFh) is protected from table reads executed in other blocks

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

### REGISTER 25-13: DEVID1: DEVICE ID REGISTER 1 FOR PIC18F2455/2550/4455/4550 DEVICES

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
bit 7							bit 0

Legend:

R = Read-only bit P = Programmable bit U = Unimplemented bit, read as '0' -n = Value when device is unprogrammed <math>u = Unchanged from programmed state

bit 7-5 **DEV2:DEV0:** Device ID bits

011 = PIC18F2455 010 = PIC18F2550 001 = PIC18F4455 000 = PIC18F4550

bit 4-0 REV3:REV0: Revision ID bits

These bits are used to indicate the device revision.

#### REGISTER 25-14: DEVID2: DEVICE ID REGISTER 2 FOR PIC18F2455/2550/4455/4550 DEVICES

R	R	R	R	R	R	R	R
DEV10 <sup>(1)</sup>	DEV9 <sup>(1)</sup>	DEV8 <sup>(1)</sup>	DEV7 <sup>(1)</sup>	DEV6 <sup>(1)</sup>	DEV5 <sup>(1)</sup>	DEV4 <sup>(1)</sup>	DEV3 <sup>(1)</sup>
bit 7							bit 0

Legend:

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

bit 7-0 **DEV10:DEV3:** Device ID bits<sup>(1)</sup>

These bits are used with the DEV2:DEV0 bits in the Device ID Register 1 to identify the part number. 0001 0010 = PIC18F2455/2550/4455/4550 devices

**Note 1:** These values for DEV10:DEV3 may be shared with other devices. The specific device is always identified by using the entire DEV10:DEV0 bit sequence.

### 25.2 Watchdog Timer (WDT)

For PIC18F2455/2550/4455/4550 devices, the WDT is driven by the INTRC source. 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.

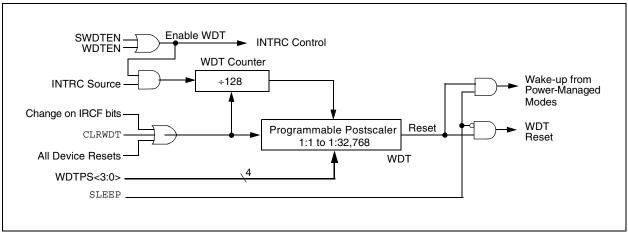
The 4 ms period of the WDT is multiplied by a 16-bit postscaler. Any output of the WDT postscaler is selected by a multiplexer, controlled by bits in Configuration Register 2H. Available periods range from 4 ms to 131.072 seconds (2.18 minutes). The WDT and postscaler are cleared when any of the following events occur: a SLEEP or CLRWDT instruction is executed, the IRCF bits (OSCCON<6:4>) are changed or a clock failure has occurred.

- Note 1: The CLRWDT and SLEEP instructions clear the WDT and postscaler counts when executed.
  - 2: Changing the setting of the IRCF bits (OSCCON<6:4>) clears the WDT and postscaler counts.
  - When a CLRWDT instruction is executed, the postscaler count will be cleared.

#### 25.2.1 CONTROL REGISTER

Register 25-15 shows the WDTCON register. This is a readable and writable register which contains a control bit that allows software to override the WDT enable Configuration bit, but only if the Configuration bit has disabled the WDT.

FIGURE 25-1: WDT BLOCK DIAGRAM



#### REGISTER 25-15: WDTCON: WATCHDOG TIMER CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
_	_	_	_	_	_	_	SWDTEN <sup>(1)</sup>
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 **SWDTEN:** Software Controlled Watchdog Timer Enable bit<sup>(1)</sup>

1 = Watchdog Timer is on0 = Watchdog Timer is off

Note 1: This bit has no effect if the Configuration bit, WDTEN, is enabled.

### TABLE 25-2: 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	SBOREN <sup>(1)</sup>	-	RI	TO	PD	POR	BOR	52
WDTCON	1	_		_	_	_	_	SWDTEN	52

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by the Watchdog Timer.

Note 1: The SBOREN bit is only available when BOREN<1:0> = 01; otherwise, the bit reads as '0'.

### 25.3 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 INTRC 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 XT, HS, XTPLL or HSPLL (Crystal-based modes). Other sources do not require an OST start-up delay; for these, 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.

Because the OSCCON register is cleared on Reset events, the INTOSC (or postscaler) clock source is not initially available after a Reset event; the INTRC clock is used directly at its base frequency. To use a higher clock speed on wake-up, the INTOSC or postscaler clock sources can be selected to provide a higher clock speed by setting bits, IRCF2:IRCF0, immediately after

Reset. For wake-ups from Sleep, the INTOSC or postscaler clock sources can be selected by setting IRCF2:IRCF0 prior to entering Sleep mode.

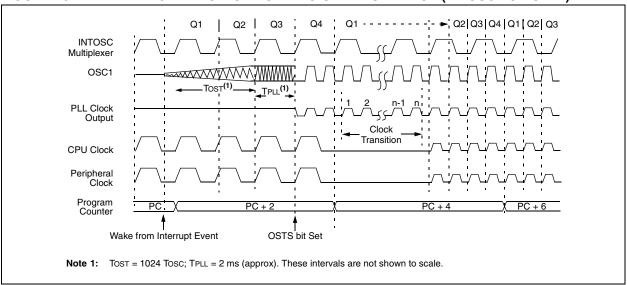
In all other power-managed modes, Two-Speed Start-up is not used. The device will be clocked by the currently selected clock source until the primary clock source becomes available. The setting of the IESO bit is ignored.

# 25.3.1 SPECIAL CONSIDERATIONS FOR USING TWO-SPEED START-UP

While using the INTRC 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.



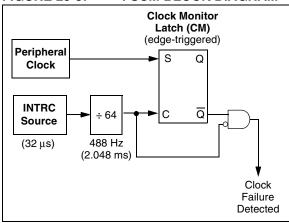


#### 25.4 Fail-Safe Clock Monitor

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 INTRC 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 25-3) is accomplished by creating a sample clock signal, which is the INTRC 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 latch (CM). The CM is set on the falling edge of the device clock source, but cleared on the rising edge of the sample clock.

FIGURE 25-3: 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 25-4). 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 25.3.1 "Special Considerations for Using Two-Speed Start-up" for more details.

To use a higher clock speed on wake-up, the INTOSC or postscaler clock sources can be selected to provide a higher clock speed by setting bits IRCF2:IRCF0 immediately after Reset. For wake-ups from Sleep, the INTOSC or postscaler clock sources can be selected by setting IRCF2:IRCF0 prior to entering Sleep mode.

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.

#### 25.4.1 FSCM AND THE WATCHDOG TIMER

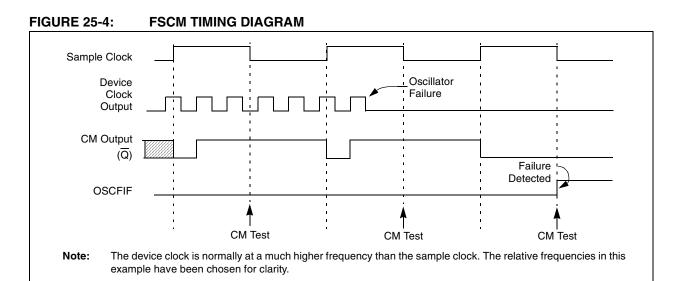
Both the FSCM and the WDT are clocked by the INTRC oscillator. Since the WDT operates with a separate divider and counter, disabling the WDT has no effect on the operation of the INTRC oscillator when the FSCM is enabled.

As already noted, the clock source is switched to the INTOSC clock when a clock failure is detected. Depending on the frequency selected by the IRCF2:IRCF0 bits, 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 Monitor 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.

### 25.4.2 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 1H (with any start-up delays that are required for the oscillator mode, such as OST or PLL timer). The INTOSC multiplexer 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 INTOSC multiplexer. The OSCCON register will remain in its Reset state until a power-managed mode is entered.



# 25.4.3 FSCM INTERRUPTS IN POWER-MANAGED MODES

By entering a power-managed mode, the clock multiplexer 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 (OSCFIF = 1), code execution will be clocked by the INTOSC multiplexer. 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 INTOSC source.

#### 25.4.4 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 EC or INTRC, monitoring can begin immediately following these events.

For oscillator modes involving a crystal or resonator (HS, HSPLL or XT), the situation is somewhat different. Since the oscillator may require a start-up time considerably longer than the FCSM 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 INTRC 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 25.3.1 "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.

# 25.5 Program Verification and Code Protection

The overall structure of the code protection on the PIC18 Flash devices differs significantly from other PICmicro<sup>®</sup> devices.

The user program memory is divided into five blocks. One of these is a boot block of 2 Kbytes. The remainder of the memory is divided into four blocks on binary boundaries.

Each of the five blocks has three code protection bits associated with them. They are:

- Code-Protect bit (CPn)
- Write-Protect bit (WRTn)
- External Block Table Read bit (EBTRn)

Figure 25-5 shows the program memory organization for 24 and 32-Kbyte devices and the specific code protection bit associated with each block. The actual locations of the bits are summarized in Table 25-3.

FIGURE 25-5: CODE-PROTECTED PROGRAM MEMORY FOR PIC18F2455/2550/4455/4550

MEMORY S	ZE/DEVICE		Diank Cada Bratastian	
24 Kbytes (PIC18F2455/2555)	32 Kbytes (PIC18F2550/4550)	Address Range	Block Code Protection Controlled By:	
Boot Block	Boot Block	000000h 0007FFh	CPB, WRTB, EBTRB	
Block 0	Block 0	000800h 001FFFh	CP0, WRT0, EBTR0	
Block 1	Block 1	002000h 003FFFh	CP1, WRT1, EBTR1	
Block 2	Block 2	004000h 005FFFh	CP2, WRT2, EBTR2	
Unimplemented Read '0's	Block 3	006000h 007FFFh	CP3, WRT3, EBTR3	
Unimplemented Read 'o's	Unimplemented Read '0's	008000h	(Unimplemented Memory Space)	
		1FFFFFh		

## TABLE 25-3: SUMMARY OF CODE PROTECTION REGISTERS

				0 0					
File	Name	ame Bit 7 Bit 6 Bit 5 Bit 4 Bit 3		Bit 3	Bit 2	Bit 1	Bit 0		
300008h	CONFIG5L	_	_	_	_	CP3 <sup>(1)</sup>	CP2	CP1	CP0
300009h	CONFIG5H	CPD	СРВ	_	_	_	_	_	_
30000Ah	CONFIG6L	_	_	_	_	WRT3 <sup>(1)</sup>	WRT2	WRT1	WRT0
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	_	_	_	_	_
30000Ch	CONFIG7L	_	_	_	_	EBTR3 <sup>(1)</sup>	EBTR2	EBTR1	EBTR0
30000Dh	CONFIG7H	_	EBTRB	_	_	_	_	_	-

Legend: Shaded cells are unimplemented.

Note 1: Unimplemented in PIC18FX455 devices; maintain this bit set.

# 25.5.1 PROGRAM MEMORY CODE PROTECTION

The program memory may be read to or written from any location using the table read and table write instructions. The device ID may be read with table reads. The Configuration registers may be read and written with the table read and table write instructions.

In normal execution mode, the CPn bits have no direct effect. CPn bits inhibit external reads and writes. A block of user memory may be protected from table writes if the WRTn Configuration bit is '0'. The EBTRn bits control table reads. For a block of user memory with the EBTRn bit set to '0', a table read instruction that executes from within that block is allowed to read.

A table read instruction that executes from a location outside of that block is not allowed to read and will result in reading '0's. Figures 25-6 through 25-8 illustrate table write and table read protection.

Code protection bits may only be written to a '0' from a '1' state. It is not possible to write a '1' to a bit in the '0' state. Code protection bits are only set to '1' by a full Chip Erase or Block Erase function. The full Chip Erase and Block Erase functions can only be initiated via ICSP operation or an external programmer.

FIGURE 25-6: TABLE WRITE (WRTn) DISALLOWED

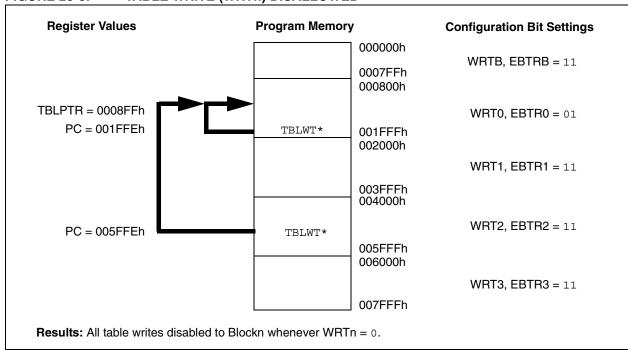


FIGURE 25-7: EXTERNAL BLOCK TABLE READ (EBTRn) DISALLOWED

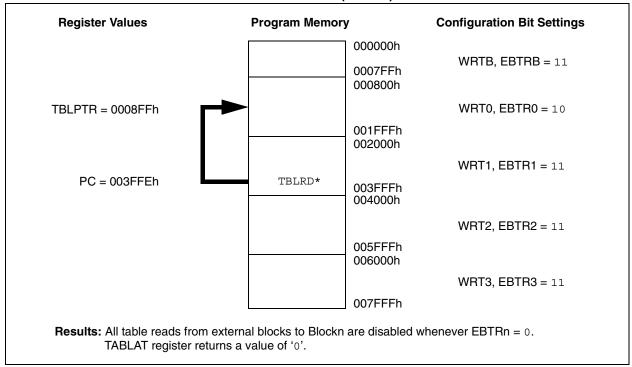
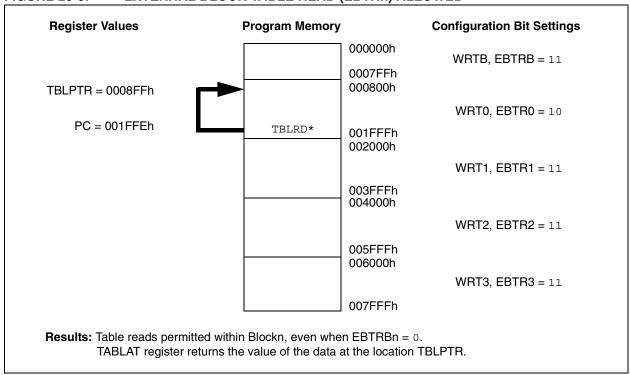


FIGURE 25-8: EXTERNAL BLOCK TABLE READ (EBTRn) ALLOWED



# 25.5.2 DATA EEPROM CODE PROTECTION

The entire data EEPROM is protected from external reads and writes by two bits: CPD and WRTD. CPD inhibits external reads and writes of data EEPROM. WRTD inhibits internal and external writes to data EEPROM. The CPU can continue to read and write data EEPROM regardless of the protection bit settings.

# 25.5.3 CONFIGURATION REGISTER PROTECTION

The Configuration registers can be write-protected. The WRTC bit controls protection of the Configuration registers. In normal execution mode, the WRTC bit is readable only. WRTC can only be written via ICSP operation or an external programmer.

#### 25.6 ID Locations

Eight memory locations (200000h-200007h) are designated as ID locations, where the user can store checksum or other code identification numbers. These locations are both readable and writable during normal execution through the TBLRD and TBLWT instructions or during program/verify. The ID locations can be read when the device is code-protected.

## 25.7 In-Circuit Serial Programming

PIC18F2455/2550/4455/4550 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.

### 25.8 In-Circuit Debugger

When the DEBUG Configuration bit is programmed to a 'o', 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 25-4 shows which resources are required by the background debugger.

TABLE 25-4: DEBUGGER RESOURCES

I/O pins:	RB6, RB7
Stack:	2 levels
Program Memory:	512 bytes
Data Memory:	10 bytes

To use the In-Circuit Debugger function of the microcontroller, the design must implement In-Circuit Serial Programming connections to MCLR/VPP/RE3, VDD, Vss, RB7 and RB6. This will interface to the In-Circuit Debugger module available from Microchip or one of the third party development tool companies.

# 25.9 Special ICPORT Features (Designated Packages Only)

Under specific circumstances, the No Connect (NC) pins of PIC18F4455/4550 devices in 44-pin TQFP packages can provide additional functionality. These features are controlled by device Configuration bits and are available only in this package type and pin count.

#### 25.9.1 DEDICATED ICD/ICSP PORT

The 44-pin TQFP devices can use NC pins to provide an alternate port for In-Circuit Debugging (ICD) and In-Circuit Serial Programming (ICSP). These pins are collectively known as the dedicated ICSP/ICD port, since they are not shared with any other function of the device.

When implemented, the dedicated port activates three NC pins to provide an alternate device Reset, data and clock ports. None of these ports overlap with standard I/O pins, making the I/O pins available to the user's application.

The dedicated ICSP/ICD port is enabled by setting the ICPRT Configuration bit. The port functions the same way as the legacy ICSP/ICD port on RB6/RB7. Table 25-5 identifies the functionally equivalent pins for ICSP and ICD purposes.

TABLE 25-5: EQUIVALENT PINS FOR LEGACY AND DEDICATED ICD/ICSP™ PORTS

Pin N	Name	Pin			
Legacy Port	Dedicated Port	Туре	Pin Function		
MCLR/VPP/ RE3	NC/ICRST/ ICVPP	Р	Device Reset and Programming Enable		
RB6/KBI2/ PGC	NC/ICCK/ ICPGC	I	Serial Clock		
RB7/KBI3/ PGD	NC/ICDT/ ICPGD	I/O	Serial Data		

**Legend:** I = Input, O = Output, P = Power

Even when the dedicated port is enabled, the ICSP and ICD functions remain available through the legacy port. When VIH is seen on the MCLR/VPP/RE3 pin, the state of the ICRST/ICVPP pin is ignored.

- **Note 1:** The ICPRT Configuration bit can only be programmed through the default ICSP port.
  - 2: The ICPRT Configuration bit must be maintained clear for all 28-pin and 40-pin devices; otherwise, unexpected operation may occur.

#### 25.9.2 28-PIN EMULATION

PIC18F4455/4550 devices in 44-pin TQFP packages also have the ability to change their configuration under external control for debugging purposes. This allows the device to behave as if it were a PIC18F2455/2550 28-pin device.

This 28-pin Configuration mode is controlled through a single pin, NC/ICPORTS. Connecting this pin to Vss forces the device to function as a 28-pin device. Features normally associated with the 40/44-pin devices are disabled along with their corresponding control registers and bits. This includes PORTD and PORTE, the SPP and the Enhanced PWM functionality of CCP1. On the other hand, connecting the pin to VDD forces the device to function in its default configuration.

The configuration option is only available when background debugging and the dedicated ICD/ICSP port are both enabled (DEBUG Configuration bit is clear and ICPRT Configuration bit is set). When disabled, NC/ICPORTS is a No Connect pin.

#### 25.10 Single-Supply ICSP Programming

The LVP Configuration bit enables Single-Supply ICSP Programming (formerly known as Low-Voltage ICSP Programming or LVP). When Single-Supply Programming is enabled, the microcontroller can be programmed without requiring high voltage being applied to the MCLR/VPP/RE3 pin, but the RB5/KBI1/PGM pin is then dedicated to controlling Program mode entry and is not available as a general purpose I/O pin.

While programming using <u>Single-Supply Programming</u>, VDD is applied to the  $\overline{MCLR}/VPP/RE3$  pin as in normal execution mode. To enter Programming mode, VDD is applied to the PGM pin.

- Note 1: High-Voltage Programming is always available, regardless of the state of the LVP bit, by applying VIHH to the MCLR pin.
  - 2: While in Low-Voltage ICSP Programming mode, the RB5 pin can no longer be used as a general purpose I/O pin and should be held low during normal operation.
  - **3:** When using Low-Voltage ICSP Programming (LVP) and the pull-ups on PORTB are enabled, bit 5 in the TRISB register must be cleared to disable the pull-up on RB5 and ensure the proper operation of the device.
  - **4:** If the device Master Clear is disabled, verify that either of the following is done to ensure proper entry into ICSP mode:
    - a) disable Low-Voltage Programming (CONFIG4L<2> = 0); or
    - b) make certain that RB5/KBI1/PGM is held low during entry into ICSP.

If Single-Supply ICSP Programming mode will not be used, the LVP bit can be cleared. RB5/KBI1/PGM then becomes available as the digital I/O pin, RB5. The LVP bit may be set or cleared only when using standard high-voltage programming (VIHH applied to the MCLR/VPP/RE3 pin). Once LVP has been disabled, only the standard high-voltage programming is available and must be used to program the device.

Memory that is not code-protected can be erased using either a Block Erase, or erased row by row, then written at any specified VDD. If code-protected memory is to be erased, a Block Erase is required. If a Block Erase is to be performed when using Low-Voltage Programming, the device must be supplied with VDD of 4.5V to 5.5V.

### 26.0 INSTRUCTION SET SUMMARY

PIC18F2455/2550/4455/4550 devices incorporate the standard set of 75 PIC18 core instructions, as well as an extended set of eight 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.

#### 26.1 Standard Instruction Set

The standard PIC18 instruction set adds many enhancements to the previous PICmicro instruction sets, while maintaining an easy migration from these PICmicro 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 26-2 lists **byte-oriented**, **bit-oriented**, **literal** and **control** operations. Table 26-1 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 zero, the result is placed in the WREG register. If 'd' is one, 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  $\mu$ s. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2  $\mu$ s. Two-word branch instructions (if true) would take 3  $\mu$ s.

Figure 26-1 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 26-2, lists the standard instructions recognized by the Microchip MPASM $^{\text{TM}}$  Assembler.

**Section 26.1.1 "Standard Instruction Set"** provides a description of each instruction.

## TABLE 26-1: 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: Carry, Digit Carry, Zero, Overflow, Negative.
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 <sub>s</sub>	12-bit register file address (000h to FFFh). This is the source address.
	12-bit register file address (000h to FFFh). This is the destination address.
f <sub>d</sub> 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
11	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 <sub>S</sub>	7-bit offset value for indirect addressing of register files (source).
z <sub>d</sub>	7-bit offset value for indirect addressing of register files (destination).
{ }	Optional argument.
[text]	Indicates an indexed address.
(text)	The contents of text.
[expr] <n></n>	Specifies bit n of the register indicated by the pointer expr.
$\rightarrow$	Assigned to.
< >	Register bit field.
€	In the set of.
italics	User-defined term (font is Courier).

### FIGURE 26-1: GENERAL FORMAT FOR INSTRUCTIONS

#### Byte-oriented file register operations **Example Instruction** 15 10 ADDWF MYREG, W, B OPCODE d а f (FILE #) 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 Byte to Byte move operations (2-word) 12 11 OPCODE f (Source FILE #) MOVFF MYREG1, MYREG2 15 12 11 0 1111 f (Destination FILE #) f = 12-bit file register address Bit-oriented file register operations OPCODE b (BIT #) f (FILE #) а BSF MYREG, bit, B 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 Literal operations 8 7 15 0 **OPCODE** k (literal) MOVLW 7Fh k = 8-bit immediate value **Control** operations CALL, GOTO and Branch operations 15 0 OPCODE n<7:0> (literal) GOTO Label 15 12 11 0 1111 n<19:8> (literal) n = 20-bit immediate value 15 CALL MYFUNC **OPCODE** n<7:0> (literal) 15 12 11 0 n<19:8> (literal) 1111 S = Fast bit 15 11 10 **OPCODE** n<10:0> (literal) BRA MYFUNC 15 8 7 OPCODE BC MYFUNC n<7:0> (literal)

TABLE 26-2: PIC18FXXXX INSTRUCTION SET

Mnemo	onic,	Description	Cycles	16-1	Bit Instr	uction W	/ord	Status	Notes
Opera	nds	Description	Cycles	MSb			LSb	Affected	Notes
BYTE-ORII	ENTED (	PERATIONS							
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
IORWF	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 <sub>s</sub> (source) to 1st word	2	1100	ffff	ffff	ffff	None	
	o u	f <sub>d</sub> (destination) 2nd word		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	li	0101	10da	ffff	ffff	C, DC, Z, OV, N	- , =
	.,,	borrow		3201				2, 20, 2, 00, 11	
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 an input and is driven low by an external device, the data will be written back with a '0'.

<sup>2:</sup> If this instruction is executed on the TMR0 register (and where applicable, 'd' = 1), the prescaler will be cleared if assigned.

<sup>3:</sup> 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.

**<sup>4:</sup>** 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.

TABLE 26-2: PIC18FXXXX INSTRUCTION SET (CONTINUED)

Mnem	onic,	Description	Cycles	16-	Bit Instr	uction V	/ord	Status	Notes
Opera	inds	Description	Cycles	MSb			LSb	Affected	Notes
BIT-ORIEN	ITED OP	ERATIONS							
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
BTFSC	f, b, a	Bit Test f, Skip if Clear	1 (2 or 3)	1011	bbba	ffff	ffff	None	3, 4
BTFSS	f, b, a	Bit Test f, Skip if Set	1 (2 or 3)	1010	bbba	ffff	ffff	None	3, 4
BTG	f, d, a	Bit Toggle f	1	0111	bbba	ffff	ffff	None	1, 2
CONTROL	OPERA	TIONS						1	1.
ВС	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	2	1110	110s	kkkk	kkkk	None	
		2nd word		1111	kkkk	kkkk	kkkk		
CLRWDT	_	Clear Watchdog Timer	1	0000	0000	0000	0100	TO, PD	
DAW	_	Decimal Adjust WREG	1	0000	0000	0000	0111	С	
GOTO	n	Go to address 1st word	2	1110	1111	kkkk	kkkk	None	
		2nd word		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 an input and is driven low by an external device, the data will be written back with a '0'.

<sup>2:</sup> If this instruction is executed on the TMR0 register (and where applicable, 'd' = 1), the prescaler will be cleared if assigned.

**<sup>3:</sup>** 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.

**<sup>4:</sup>** 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.

TABLE 26-2: PIC18FXXXX INSTRUCTION SET (CONTINUED)

Mnemo	onic,	Description	Cycles	16-	Bit Inst	ruction	Word	Status	Notes
Opera	nds	Description	Cycles	MSb			LSb	Affected	Notes
LITERAL (	ONS								
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	2	1110	1110	OOff	kkkk	None	
		to FSR(f) 1st word		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 MEN	MORY ↔	PROGRAM MEMORY OPERATI	ONS						
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 an input and is driven low by an external device, the data will be written back with a '0'.

<sup>2:</sup> If this instruction is executed on the TMR0 register (and where applicable, 'd' = 1), the prescaler will be cleared if assigned.

**<sup>3:</sup>** 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.

<sup>4:</sup> 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.

#### 26.1.1 STANDARD INSTRUCTION SET

ADD	LW	ADD Literal to W					
Synta	ax:	ADDLW	k				
Oper	ands:	$0 \le k \le 255$	$0 \le k \le 255$				
Oper	ation:	$(W) + k \rightarrow V$	W				
Statu	s Affected:	N, OV, C, E	)C, Z				
Enco	ding:	0000	1111	kkkk	kkkk		
Desc	ription:	The conten 8-bit literal W.					
Word	ls:	1	1				
Cycle	es:	1	1				
QC	ycle Activity:						
	Q1	Q2	Q3		Q4		
	Decode	Read literal 'k'	Proce Data		/rite to W		

Example: ADDLW 15h

 $\begin{array}{rcl} \text{Before Instruction} & & \\ W & = & 10\text{h} \\ \text{After Instruction} & & \\ W & = & 25\text{h} \end{array}$ 

ADDWF	ADD W to f				
Syntax:	ADDWF	f {,d {,a}	}		
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$	d ∈ [0,1]			
Operation:	$(W) + (f) \to$	dest			
Status Affected:	N, OV, C, D	)C, Z			
Encoding:	0010	01da	fff	f	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 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.				
Words:	1				
Cycles:	1				
Q Cycle Activity:					
Q1	Q2	Q3	3		Q4
Decode	Read register 'f'	Proce Data			rite to tination

Example: ADDWF REG, 0, 0

Before Instruction

W = 17h REG = 0C2hAfter Instruction

W = 0D9hREG = 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).

ADD	WFC	ADD W an	d Carry I	bit to	f	
Synta	ax:	ADDWFC	f {,d {,	a}}		
Oper	ands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$				
Oper	ation:	(W) + (f) +	$(C) \rightarrow de$	est		
Statu	s Affected:	N, OV, C, I	OC, Z			
Enco	ding:	0010	00da	fff	f	ffff
Desc	ription:	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 ≤ 95 (5Fh). See Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.				
Word	ls:	1				
Cycle	es:	1				
QC	ycle Activity:					
	Q1	Q2	Q3			Q4
	Decode	Read	Proce	ss	W	/rite to

ANDLW	AND Litera	al with W		
Syntax:	ANDLW	k		
Operands:	$0 \le k \le 255$	;		
Operation:	(W) .AND.	$k \rightarrow W$		
Status Affected	i: N, Z			
Encoding:	0000	1011 k	kkk	kkkk
Description:		nts of W are A		
Words:	1			
Cycles:	1			
Q Cycle Activ	ity:			
Q1	Q2	Q3		Q4
Decod	e Read literal 'k'	Process Data	Wr	rite to W
Example:	ANDLW	05Fh		

Before Instruction W A3h After Instruction W 03h

register 'f'

Data

destination

AND	WF	AND W wi	th f			
Synta	ax:	ANDWF	ANDWF f {,d {,a}}			
Oper	ands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$	d ∈ [0,1]			
Operation: (W) .AND. (f) $\rightarrow$ dest						
Statu	s Affected:	N, Z				
Enco	ding:	0001 01da ffff ffff				
Desc	Encoding:    0001   01da   ffff   ffff     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).   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 operate in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See   Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.			sult is stored is stored back is selected. It is select the dinstruction ion operates ddressing h). See ented and in Indexed		
Word	ls:	1				
Cycle	es:	1				
QC	ycle Activity:					
	Q1	Q2	Q3		Q4	
	Decode	Read register 'f'	Proce Data		Write to destination	

ΑN	IDWF	REG,	Ο,	0	
on					
=	17h C2h				
=	CZII				
=	02h				
=	C2h				

Example:

Before Instruction W REG After Instruction

W REG

BC	Branch if	Carry			
Syntax:	BC n				
Operands:	-128 ≤ n ≤	127			
Operation:	•	if Carry bit is '1' $(PC) + 2 + 2n \rightarrow PC$			
Status Affected:	None				
Encoding:	1110	0010	nnnn nnn		
Description:	will branch The 2's co added to th incremente instruction	mplement ne PC. Since ed to fetch the new acon. This instruction.	umber '2n' is e the PC will ha he next ddress will be ruction is then		
Words:	1				
Cycles:	1(2)				
Q Cycle Activity If Jump:	r:				
. 01	Q2	Q3	Q4		
Q1					
Decode	Read literal 'n'	Process Data	Write to P		
			Write to P		
Decode	ʻn' No	Data	No		
Decode	ʻn' No	Data No	No		
Decode  No operation	ʻn' No	Data No	No		
Decode  No operation  If No Jump:	'n' No operation  Q2  Read literal	Data  No operation  Q3  Process	No operation  Q4  No		
Decode  No operation  If No Jump:  Q1	'n' No operation	Data No operation Q3	No operation		
Decode  No operation  If No Jump:  Q1	'n' No operation  Q2  Read literal	Data  No operation  Q3  Process	No operation  Q4  No		
Decode  No operation  If No Jump: Q1  Decode	'n' No operation  Q2 Read literal 'n'  HERE	Data No operation Q3 Process Data	No operation  Q4  No operation		

1; address (HERE + 12) 0; address (HERE + 2)

If Carry PC If Carry PC

BCF		Bit Clear f			
Synta	ax:	BCF f, b	{,a}		
Oper	ands:	$0 \le f \le 255$ $0 \le b \le 7$ $a \in [0,1]$			
Oper	ation:	$0 \to f {<} b {>}$			
Statu	s Affected:	None			
Enco	ding:	1001	bbba ff	ff ffff	
Desc	ription:	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 ≤ 95 (5Fh). See  Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.			
Word	ls:	1			
Cycle	es:	1			
Q C	ycle Activity:				
1	Q1	Q2	Q3	Q4	
	Decode	Read	Process	Write	

Example: FLAG\_REG, 7, 0

register 'f'

Data

register 'f'

Before Instruction

C7h FLAG\_REG =

After Instruction

FLAG\_REG = 47h

BN	Branch if I	Negative		
Syntax:	BN n			
Operands:	-128 ≤ n ≤	127		
Operation:	if Negative (PC) + 2 +		;	
Status Affected:	None			
Encoding:	1110	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 ha incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then two-cycle instruction.				er '2n' is PC will have ext ss will be
Words:	1			
Cycles:	1(2)			
Q Cycle Activity: If Jump:				
Q1	Q2	Q3	;	Q4
Decode	Read literal	Proce	ss \	Write to PC

Q1	Q2	Q3	Q4
Decode	Read literal	Process	Write to PC
	'n'	Data	
No	No	No	No
operation	operation	operation	operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal	Process	No
	'n'	Data	operation

Jump Example: HERE BN

Before Instruction

PC address (HERE)

After Instruction

If Negative PC

1; address (Jump)

address (HERE + 2)

вис		Branch if	Branch if Not Carry				
Synta	ax:	BNC n					
Oper	ands:	-128 ≤ n ≤	127				
Oper	ation:	,	if Carry bit is '0' $(PC) + 2 + 2n \rightarrow PC$				
Statu	s Affected:	None					
Enco	ding:	1110	0011	nnn	n	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 hav incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.				2n' is will have			
Word	ls:	1					
Cycle	es:	1(2)	1(2)				
Q C	ycle Activity: mp:						
	Q1	Q2	Q3	3		Q4	
	Decode	Read literal	Proce	ess	Wri	te to PC	

	Q1	Q2	Q3	Q4	
	Decode	Read literal	Process	Write to PC	
		'n'	Data		
	No	No	No	No	
	operation	operation	operation	operation	
. '		•	•	•	

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal	Process	No
	ʻn'	Data	operation

Example:	HERE	BNC	Jump	
Before Instruction PC After Instruction	=	address	(HERE)	
If Carry PC If Carry	= = =	0; address 1;	(Jump)	

BNN		Branch if N	Not Nega	itive		
Synta	ax:	BNN n	BNN n			
Oper	ands:	-128 ≤ n ≤ °	-128 ≤ n ≤ 127			
Oper	ation:	•	if Negative bit is '0' $(PC) + 2 + 2n \rightarrow PC$			
Status Affected: None						
Enco	ding:	1110	0111	nnnn	nnnn	
Desc	ription:	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.				
Word	ls:	1	1			
Cycle	es:	1(2)	1(2)			
	Q Cycle Activity: If Jump:					
	Q1	Q2	Q3	1	Q4	
	Decode	Read literal 'n'	Proce Data	-	Vrite to PC	
	No	No	No		No	
	operation	operation	operat	ion	operation	

	Q1	Q2	Q3	Q4
	Decode	Read literal	Process	No
		'n'	Data	operation
Exan	nple:	HERE	BNN Jump	

Before Instruction			
PC	=	address	(HERE)
After Instruction			
If Negative	=	0;	
PC	=	address	(Jump)
If Negative	=	1;	
PC.	_	address	(HEBE + 2)

If No Jump:

BNO	v	Branch if I	Not Over	flow	
Synta	ax:	BNOV n			
Oper	ands:	-128 ≤ n ≤	-128 ≤ n ≤ 127		
Oper	ation:	if Overflow (PC) + 2 +	<b></b>	;	
Statu	s Affected:	None			
Enco	ding:	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 two-cycle instruction.		2n' is Cwill have		
Word		1			
Cycles: Q Cycle Activity: If Jump:		1(2)			
	Q1	Q2	Q3	;	Q4
	Decode	Read literal 'n'	Proce Data		ite to PC
	No operation	No operation	No operat	ion op	No peration

If No	Jump:			
	Q1	Q2	Q3	Q4
	Decode	Read literal	Process	No
		ʻn'	Data	operation

Example:	HERE	BNOV	Jump
Before Instruction	on		
PC	=	address	(HERE)
After Instruction			
If Overflow	=	0;	
PC	=	address	(Jump)
If Overflow	=	1;	
PC	=	address	(HERE + 2)

BNZ		Branch if N	lot Zero			
Synta	ax:	BNZ n				
Oper	ands:	-128 ≤ n ≤ °	127			
Oper	ation:	if Zero bit is (PC) + 2 + 3				
Statu	s Affected:	None				
Enco	ding:	1110	0001	nnnn	nnnn	
Desc	ription:	will branch. The 2's con added to the incremente instruction, PC + 2 + 2	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.			
Word	ls:	1	1			
Cycle	es:	1(2)	1(2)			
Q Cycle Activity: If Jump:						
	Q1	Q2	Q3		Q4	
	Decode	Read literal 'n'	Proce Data		rite to PC	
	No	No	No		No	
	operation	operation	operat	ion c	peration	
If No	Jump:					

Example:	HERE	BNZ Jump
Before Instruction PC After Instruction	=	address (HERE)
If Zero PC If Zero PC	= = = =	0; address (Jump) 1; address (HERE + 2

Q2

Read literal

'n'

Q1

Decode

Q3

Process

Data

Q4

No

operation

BRA	Unconditional Branch				
Syntax:	BRA n				
Operands:	-1024 ≤ n ≤ 1023				
Operation:	$(PC) + 2 + 2n \rightarrow PC$				
Status Affected:	None				
Encoding:	1101 Onnn 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				
Cualant	0				

Words: 1
Cycles: 2
Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal	Process	Write to PC
	ʻn'	Data	
No	No	No	No
operation	operation	operation	operation

<u>Example:</u> HERE BRA Jump

Before Instruction

PC = address (HERE)

After Instruction

PC = address (Jump)

BSF		Bit Set f			
Synta	ax:	BSF f, b {	,a}		
Oper	rands:	$0 \le f \le 255$ $0 \le b \le 7$ $a \in [0,1]$			
Oper	ration:	$1 \to f {<} b {>}$	$1 \rightarrow f < b >$		
Statu	ıs Affected:	None			
Enco	oding:	1000	bbba	ffff	ffff
Desc	ription:	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 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.			
Word	ds:	1			
Cycles:		1			
Q Cycle Activity:					
	Q1	Q2	Q3	3	Q4
	Decode	Read register 'f'	Proce Data		Write egister 'f'

Example: BSF FLAG\_REG, 7, 1

Before Instruction

FLAG\_REG = 0Ah

After Instruction

FLAG\_REG = 8Ah

BTFSC	Bit Test File	, Skip if Clear	•	BTFS	SS	Bit Test File	, Skip if Set	
Syntax:	BTFSC f, b	{,a}		Synta	ax:	BTFSS f, b	{,a}	
Operands:	$0 \le f \le 255$			Oper	ands:	$0 \le f \le 255$		
	$0 \le b \le 7$ $a \in [0,1]$					$0 \le b < 7$ $a \in [0,1]$		
Operation:	skip if (f <b>)</b>	= 0		Oper	ation:	skip if (f <b>)</b>	= 1	
Status Affected:	. , ,	None			s Affected:	None	- <u>-</u>	
Encoding:	1011	bbba ff	ff ffff	Enco		1010	bbba ff:	ff ffff
Description:		gister 'f' is '0',			ription:		gister 'f' is '1', t	
•	instruction is the next inst	skipped. If bit ruction fetched	'b' is '0', then I during the			instruction is the next inst	skipped. If bit ruction fetched	'b' is '1', then during the
		uction execution execution executed inst	on is discarded				uction execution executions executed instanta	
		cle instruction					cle instruction.	_
			k is selected. If				e Access Bank	
	'a' is '1', the GPR bank (	BSR is used to	select the			'a' is '1', the GPR bank (	BSR is used to	select the
	,	d the extended	d instruction			,	d the extended	dinstruction
			ion operates in				ed, this instructi	•
		ral Offset Addr ever f≤95 (5Fl	-				iteral Offset Ad ever f ≤ 95 (5Fh	-
		•	Oriented and				26.2.3 "Byte-	,
	Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.						d Instructions et Mode" for d	
Words:	1			Word	s:	1		
Cycles:	1(2)			Cycle	es:	1(2)		
•		cles if skip and		•			cles if skip and	
	by a	2-word instruc	ction.			by a	2-word instruc	ction.
Q Cycle Activity:	00	00	04	Q C	ycle Activity:	00	00	04
Q1 Decode	Q2 Read	Q3 Process	Q4 No		Q1 Decode	Q2 Read	Q3 Process	Q4 No
Decode	register 'f'	Data	operation		Decode	register 'f'	Data	operation
If skip:				lf sk	ip:		•	•
Q1	Q2	Q3	Q4		Q1	Q2	Q3	Q4
No	No	No	No		No	No	No	No
operation  If skip and followe	operation	operation	operation	If ok	operation	operation by 2-word ins	operation	operation
Q1	Q2	Q3	Q4	11 31	Q1	Q2	Q3	Q4
No	No	No	No		No	No	No	No
operation	operation	operation	operation		operation	operation	operation	operation
No 	No	No 	No 		No 	No 	No 	No 
operation	operation	operation	operation		operation	operation	operation	operation
Example:	FALSE :	TFSC FLAG	B, 1, 0	<u>Exam</u>	nple:	FALSE :	TFSS FLAG	, 1, 0
Potovo Inoterra	TRUE :				Dafava locator :	TRUE :		
Before Instruction PC = address (HERE)				Before Instruct PC		ress (HERE)		
After Instruction		,			After Instructio			
If FLAG< PC		ress (TRUE)			If FLAG<		ress (FALSE)	
If FLAG< PC	<1> = 1;	ress (FALSE)	)		PC If FLAG< PC	1> = 1;	ress (FALSE)	

BTG		Bit Toggle f				
Synta	ax:	BTG f, b {	BTG f, b {,a}			
Oper	ands:	$0 \le f \le 255$ $0 \le b < 7$ $a \in [0,1]$	* = * ''			
Oper	ation:	$(\overline{f < b >}) \rightarrow f <$	<b></b>			
Statu	s Affected:	None				
Enco	ding:	0111	bbba	fff	f f	fff
Description:  Bit 'b' in data memory location 'f' is inverted.  If 'a' is '0', the Access Bank is select If 'a' is '1', the BSR is used to select GPR bank (default).  If 'a' is '0' and the extended instruction set is enabled, this instruction opera in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See  Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.			ected. ect the ection erates ng			
Word	ls:	1				
Cycle	es:	1				
QC	ycle Activity:					
	Q1	Q2	Q3		Q4	1
	Decode	Read register 'f'	Proces Data		Writ registe	-

Example: BTG PORTC, 4, 0

Before Instruction:

PORTC = 0111 0101 [75h]

After Instruction:

PORTC = 0110 0101 [65h]

BOV	Branch if Overfl	ow	
Syntax:	BOV n		
Operands:	$\text{-}128 \leq n \leq 127$		
Operation:	if Overflow bit is '1' $(PC) + 2 + 2n \rightarrow PC$		
Status Affected:	None		
Encoding:	1110 010	nnnn	nnnn
Description:	If the Overflow 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:

Q1	Q2	Q3	Q4
Decode	Read literal	Process	Write to PC
	ʻn'	Data	
No	No	No	No
operation	operation	operation	operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal	Process	No
	'n'	Data	operation

Example: HERE BOV Jump

Before Instruction

PC address (HERE)

After Instruction If Overflow PC If Overflow PC

= = = 1; address (Jump)

address (HERE + 2)

BZ	Branch if Zero		
Syntax:	BZ n		
Operands:	$-128 \le n \le 127$		
Operation:	if Zero bit is '1' (PC) + 2 + 2n $\rightarrow$ PC		
Status Affected:	None		
Encoding:	1110 0000 nnnn nnnn		
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		

Words: 1 Cycles: 1(2)

Q Cycle Activity: If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal	Process	Write to PC
	'n'	Data	
No	No	No	No
operation	operation	operation	operation

two-cycle instruction.

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal	Process	No
	ʻn'	Data	operation

Example: HERE  $_{\mathrm{BZ}}$ Jump

Before Instruction

PC address (HERE)

After Instruction

If Zero PC address (Jump)

If Zero

РC address (HERE + 2)

CALL	Subroutine Call				
Syntax:	CALL k {	CALL k {,s}			
Operands:	$0 \le k \le 1048575$ $s \in [0,1]$				
Operation:	$(PC) + 4 \rightarrow TOS,$ $k \rightarrow PC < 20:1>,$ if $s = 1$ $(W) \rightarrow WS,$ $(STATUS) \rightarrow STATUSS,$ $(BSR) \rightarrow BSRS$				
Status Affected:	None				
Encoding: 1st word (k<7:0>) 2nd word(k<19:8>)	1110 1111	110s k <sub>19</sub> kkk	k <sub>7</sub> kkk kkkk	kkkk <sub>0</sub> kkkk <sub>8</sub>	

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

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: 2 2 Cycles:

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'<7:0>,	Push PC to stack	Read literal 'k'<19:8>,
			Write to PC
No	No	No	No
operation	operation	operation	operation

Example: HERE CALL THERE, 1

Before Instruction

PC address (HERE)

After Instruction

PC TOS address (THERE) address (HERE + 4) WS =

**BSRS BSR** BSRS = STATUSS = STATUS

**CLRWDT** 

Operands:

Operation:

Encoding:

Description:

Status Affected:

Syntax:

CLRF	Clear f	Clear f			
Syntax:	CLRF f{	,a}			
Operands:	$0 \le f \le 255$ $a \in [0,1]$	$0 \le f \le 255$ $a \in [0,1]$			
Operation:	$\begin{array}{l} 000h \rightarrow f, \\ 1 \rightarrow Z \end{array}$	•			
Status Affected:	Z				
Encoding:	0110	101a	ffff	ffff	
Description:	register.  If 'a' is '0',  If 'a' is '1',  GPR bank  If 'a' is '0' a  set is enab  in Indexed  mode whe  Section 20  Bit-Orient	Clears the contents of the specified register.  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 26.2.3 "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	Process	,	Mrita	

	post <u>sca</u> ler of the WDT. Status bits, TO and PD, are set.			
Word	ls:	1		
Cycle	es:	1		
QC	ycle Activity:			
	Q1	Q2	Q3	Q4
	Decode	No	Process	No
		operation	Data	operation
Example:		CLRWDT		

**Clear Watchdog Timer** 

 $000h \to WDT \ postscaler,$ 

0000

CLRWDT instruction resets the Watchdog Timer. It also resets the

0000

0100

CLRWDT

 $1 \to \frac{\overline{\mathsf{TO}},}{\mathsf{PD}}$ 

TO, PD

 $000h \rightarrow WDT, \\$ 

None

	Decode	Read register 'f'	Process Data	Write register 'f'
<u>Exan</u>	nple:	CLRF	FLAG_REG,	1

Before Instruction

FLAG\_REG = 5Ah

After Instruction

FLAG\_REG = 00h

Example: CLRWI	TC	
Before Instruction		
WDT Counter	=	?
After Instruction		
WDT Counter	=	00h
WDT Postscaler	=	0
TO	=	1
DD	_	1

COM	IF	Complement f			
Synta	ax:	COMF f	{,d {,a}}		
Oper	ands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$			
Oper	ation:	$(\overline{f}) \rightarrow dest$			
Statu	s Affected:	N, Z			
Enco	ding:	0001	11da	ffff	ffff
Desc	ription:	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 ≤ 95 (5Fh). See Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.			
Word	ls:	1			
Cycle	es:	1			
QC	ycle Activity:				
	Q1	Q2	Q3	3	Q4
	Decode	Read register 'f'	Proce Data		Write to estination

Example:	CO	MF	REG,	0, 0
Before Instruc REG After Instruction	=	13h		
REG W	=	13h ECh		

CPF	SEQ	Compare f	with W, Skip	if f = W	
Synta	ax:	CPFSEQ	f {,a}		
Oper	ands:	$0 \le f \le 255$ a $\in [0,1]$			
Oper	ation:	(f) - (W),			
		skip if (f) =	(W)		
		(unsigned o	comparison)		
Statu	is Affected:	None			
Enco	oding:	0110	001a fff	f ffff	
Desc	cription:		the contents of		
			o the contents an unsigned s		
			en the fetched		
		•	nd a NOP is ex		
			king this a two	-cycle	
		instruction.  If 'a' is '0', the Access Bank is selected.			
			he BSR is use		
		GPR bank			
			nd the extende		
			ed, this instruc Literal Offset A		
			ever f ≤ 95 (5F	-	
			.2.3 "Byte-Ori		
			d Instruction		
			set Mode" for	details.	
Word		1			
Cycle	es:	1(2) <b>Note:</b> 3 c	ycles if skip an	d followed	
			a 2-word instru		
QC	ycle Activity:	·			
	Q1	Q2	Q3	Q4	
	Decode	Read	Process	No	
lf ala	in.	register 'f'	Data	operation	
If sk	up: Q1	Q2	Q3	Q4	
	No	No	No	No	
	operation	operation	operation	operation	
If sk	ip and followed	,		_	
	Q1	Q2	Q3	Q4	
	No operation	No operation	No operation	No operation	
	No	No	No	No	
	operation	operation	operation	operation	
Exan	nple:	HERE	CPFSEQ REG	. 0	
		NEQUAL	:	, -	
		EQUAL	:		
Before Instruction					
	PC Addre		RE		
	vv REG	= ? = ?			
	After Instruction				
	If REG PC	= W; = Ad	dress (EQUA	r.)	
	If REG	≠ W;	. ~		
	PC	= Ad	dress (NEQUA	AL)	

CPFS	GT	Compare f	with W, Skip	if f > W	CPFSLT
Synta	iX:	CPFSGT	f {,a}		Syntax:
Opera	ands:	$0 \le f \le 255$ $a \in [0,1]$			Operands:
Opera	ation:	(f) - (W), skip if $(f) > (unsigned constant)$	(W) comparison)		Operation:
Status	s Affected:	None			Status Affected:
Enco	ding:	0110	010a ff	ff ffff	
Desc	ription:	Compares t	the contents o	f data memory	Encoding:
			o the contents	•	Description:
Word	s:	If the contents of instruction in executed in two-cycle in If 'a' is '0', till f'a' is '1', till GPR bank of If 'a' is '0' and set is enable in Indexed of	WREG, then a discarded a stead, making astruction. The Access Bathe BSR is use (default).	eater than the the fetched and a NOP is this a like is selected. It is select the like is select the like is select the like is select the like is select the like is select the like is select the like is select the like is select the like is select the like is select the like is select the like is select the like is select the like is select than the like is select than the like is select than the like is select than the like is selected.	Words: Cycles:  Q Cycle Activity: Q1
					Decode
Cycle	s:	1(2)	cles if skip an	d followed	200000
			2-word instru		If skip:
Q Cy	cle Activity:				Q1
·	Q1	Q2	Q3	Q4	No
	Decode	Read	Process	No	operation
		register 'f'	Data	operation	If skip and followe
If ski		00	00	04	Q1
Ī	Q1 No	Q2 No	Q3 No	Q4 No	No
	operation	operation	operation	operation	operation
lf ski	-	d by 2-word in:		operation	No
11 014	Q1	Q2	Q3	Q4	operation
ſ	No	No	No	No	
	operation	operation	operation	operation	Example:
İ	No	No	No	No	
	operation	operation	operation	operation	<b>5</b> ( ) .
Exam	ıple:	HERE NGREATER	CPFSGT RE	EG, 0	Before Instruc PC W

ntax:	CPFSLT	f {,a}		
perands:	$0 \le f \le 255$ $a \in [0,1]$			
peration:	(f) – (W), skip if (f) < (unsigned		on)	
atus Affected:	None			
coding:	0110	000a	ffff	ffff
escription:	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).			
ords:	1			
vcles:		•	kip and fol I instruction	
Cycle Activity:				
Q1	Q2	Q3	}	Q4

Compare f with W, Skip if f < W

	Q1	Q2	Q3	Q4
	Decode	Read	Process	No
		register 'f'	Data	operation
ماہ	in:			<u> </u>

Q1	Q2	Q3	Q4
No	No	No	No
operation	operation	operation	operation

#### red by 2-word instruction:

Q1	Q2	Q3	Q4
No	No	No	No
operation	operation	operation	operation
No	No	No	No
operation	operation	operation	operation

CPFSLT REG, 1 HERE

NLESS LESS

uction

Address (HERE)

After Instruction

If REG PC If REG Address (LESS) W;

Address (NLESS)

Before Instruction PC W

After Instruction

If REG PC If REG PC

GREATER

Address (HERE) ?

Address (GREATER) W; Address (NGREATER)

DAW	1	Decimal A	Decimal Adjust W Register					
Synta	ax:	DAW	DAW					
Oper	ands:	None	None					
Oper	ration:	(W<3:0>) + else	If [W<3:0>> 9] or [DC = 1] then (W<3:0>) + 6 $\rightarrow$ W<3:0>; else (W<3:0>) $\rightarrow$ W<3:0>					
		If $[W<7:4> + DC > 9]$ or $[C = 1]$ then $(W<7:4>) + 6 + DC \rightarrow W<7:4>$ ; else $(W<7:4>) + DC \rightarrow W<7:4>$						
Statu	s Affected:	С	С					
Enco	oding:	0000	0000 00	00 011	1			
Desc	eription:	resulting fro variables (e	s the eight-bi om the earlier each in packe es a correct p	addition of t	two at)			
Word	ds:	1						
Cycle	es:	1	1					
QC	ycle Activity:							
	Q1	Q2 Q3 Q4						
	Decode	Read register W	Process Data	Write W				

Example 1:	DA	AM.
Before Inst	ruction	
W	=	A5h
С	=	0
DC	=	0
After Instru	ction	
W	=	05h
С	=	1
DC	=	0
Example 2:		
Before Inst	ruction	
W	=	CEh
С	=	0
DC	=	0
After Instru	ction	
W	=	34h
С	=	1
DC	_	Λ

DEC	F	Decrement	tf					
Synta	ax:	DECF f {,d	DECF f {,d {,a}}					
Oper	rands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$	d ∈ [0,1]					
Oper	ation:	$(f) - 1 \rightarrow de$	est					
Statu	s Affected:	C, DC, N, C	OV, Z					
Enco	oding:	0000	01da ffi	ff ffff				
Desc	oription:	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). 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 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.						
Word	ds:	1						
Cycle	es:	1						
Q Cycle Activity:								
	Q1	Q2	Q3	Q4				
	Decode	Read	Process	Write to				

Example:	D	ECF	CNT,	1,	0
Before Instru		041			
CNT Z	=	01h 0			
After Instruc	tion				
CNT	=	00h			

register 'f'

Data

destination

DECFS	SZ	Decrement	t f, Skip if 0		DCF	SNZ	Decrement	t f, Skip if Not	0
Syntax	:	DECFSZ f {,d {,a}}		Synt	ax:	DCFSNZ	f {,d {,a}}		
Operar	nds:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$			Ope	rands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$		
Operat	ion:	(f) $-1 \rightarrow de$ skip if resul	-		Ope	ration:	(f) $-1 \rightarrow de$ skip if resul	•	
Status	Affected:	None		Statu	us Affected:	None			
Encodi	ing:	0010	11da ff:	ff ffff	Enco	oding:	0100	11da fff	f ffff
Descrip	otion:	decremented placed in William placed back of the result which is alroad a NOP in it a two-cyclif 'a' is '0', the second of the s	le instruction. he Access Ba he BSR is use (default). nd the extend	the result is ne result is (default). It instruction, is discarded stead, making the instruction of the instruction operates addressing the instruction operates and instru	Desc	cription:	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 ≤ 95 (5Fh). See Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed		the result is e result is (default). In ext dy fetched, is decuted yele with its selected. It is select the ed instruction operates addressing Fh). See ented and in Indexed
Words:	:	1					Literal Offs	set Mode" for	details.
Cycles	:		cles if skip ar 2-word instru		Word Cycl			cycles if skip a	
Q Cyc	cle Activity: Q1	Q2	Q3	Q4	QC	Cycle Activity:	by	a 2-word instr	uction.
	Decode	Read	Process	Write to		Q1	Q2	Q3	Q4
ماداه کا		register 'f'	Data	destination	J	Decode	Read register 'f'	Process Data	Write to destination
If skip	: Q1	Q2	Q3	Q4	If sk	kin:	register i	Dala	destination
	No	No	No No	No No	s. 1	Q1	Q2	Q3	Q4
	operation	operation	operation	operation		No	No	No	No
	•	d by 2-word in		000.000	J	operation	operation	operation	operation
	Q1	Q2	Q3	Q4	If sl	kip and followe	d by 2-word in	struction:	
	No	No	No	No	]	Q1	Q2	Q3	Q4
	operation	operation	operation	operation		No	No	No	No
	No	No	No	No	]	operation	operation	operation	operation
	operation	operation	operation	operation		No	No	No	No
Examp	ole:	HERE	DECFSZ	CNT, 1, 1		operation	operation	operation	operation
		CONTINUE	GOTO	LOOP	<u>Exar</u>	mple:	ZERO	DCFSNZ TEM : :	IP, 1, 0
В	efore Instruc		, (HEDE.)			Before Instruc		•	
At	PC fter Instruction CNT If CNT PC	on = CNT - · = 0;	S (HERE)  1  S (CONTINUE	₹)		TEMP After Instruction TEMP If TEMP	=	? TEMP – 1, 0;	
	If CNT PC	<b>≠</b> 0;	S (HERE + 2			PC If TEMP PC	= ≠ =	Address (2 0; Address (1	

GOTO	Unconditional Branch					
Syntax:	GOTO k	GOTO k				
Operands:	$0 \le k \le 10$	$0 \le k \le 1048575$				
Operation:	$k \rightarrow PC < 20:1 >$					
Status Affected:	None					
Encoding: 1st word (k<7:0>) 2nd word(k<19:8>)	1110 1111	1111 k <sub>19</sub> kkk	k <sub>7</sub> kkk kkkk	kkkk <sub>0</sub> kkkk <sub>8</sub>		
Description:	GOTO allows an unconditional branch anywhere within the entire 2-Mbyte memory range. The 20-bit value 'k' is loaded into PC<20:1>. GOTO is always a two-cycle instruction.					

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)

INCF	•	Increment	f					
Synta	ax:	INCF f {,c	INCF f {,d {,a}}					
Oper	ands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$	d ∈ [0,1]					
Oper	ation:	$(f) + 1 \rightarrow de$	est					
Statu	s Affected:	C, DC, N,	OV, Z					
Enco	ding:	0010	10da	fff	f	ffff		
	ription:	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). 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 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.						
Word	ls:	1						
Cycle	Cycles: 1							
QC	Q Cycle Activity:							
	Q1	Q2	Q3			Q4		
	Decode	Read	Proces	ss	V	/rite to		

II	NCF	CNT,	1,	0
ction				
= = = =	FFh 0 ? ?			
on				
= = = =	00h 1 1 1			
	etion = = = = = on	= FFh = 0 = ? = ?	etion	etion

register 'f'

Data

destination

INCF	SZ	Increment	f, Skip if 0		INF	SNZ	Increment	f, Skip if Not	0
Synta	ax:	INCFSZ f	{,d {,a}}		Syn	tax:	INFSNZ f	{,d {,a}}	
Oper	ands:	0 ≤ f ≤ 255			Оре	erands:	$0 \le f \le 255$		
		$d \in [0,1]$					d ∈ [0,1]		
		$a \in [0,1]$					a ∈ [0,1]		
Oper	ation:	$(f) + 1 \rightarrow de$	est,		Оре	eration:	$(f) + 1 \rightarrow de$	-	
		skip if resul	t = 0		01-1	۸ # +	skip if resul	<b>l</b> ≠ 0	
Statu	s Affected:	None				us Affected:	None		
Enco	oding:	0011	11da ff	ff ffff		oding:	0100	10da ff:	
Desc	ription:	The conten	ts of register "	r' are	Des	cription:		ts of register 'fd. If 'd' is '0', t	
	•		d. If 'd' is 'o', t					d. If 'd' is '1', th	
		•	/. If 'd' is '1', th				•	k in register 'f'	
		•	k in register 'f'.					is not '0', the	
			is '0', the nex eady fetched,					which is alrea	•
				stead, making				and a NOP is ex aking it a two-c	
		it a two-cyc	le instruction.				instruction.	iking it a two-c	yolo
		•		nk is selected.			If 'a' is '0', t	he Access Ba	nk is selected.
		GPR bank		d to select the				he BSR is use	d to select the
			nd the extend	ed instruction			GPR bank	(default). .nd the extend	ad instruction
		set is enabl	ed, this instru	ction operates				led, this instru	
			Literal Offset A	-				Literal Offset A	•
			never f ≤ 95 (5	,				never $f \le 95$ (5)	,
			.2.3 "Byte-Ored Instruction					.2.3 "Byte-Or	
			set Mode" for					ed Instruction set Mode" for	
Word	ds:	1			Wor	rds:	1	101	dotailo.
Cycle	es:	1(2)			Сус		1(2)		
,		` '	cycles if skip and followed		-,-		` '	cycles if skip a	nd followed
		by a	2-word instru	ction.			by	a 2-word inst	ruction.
QC	ycle Activity:				Q (	Cycle Activity:			
	Q1	Q2	Q3	Q4		Q1	Q2	Q3	Q4
	Decode	Read	Process	Write to		Decode	Read	Process	Write to
		register 'f'	Data	destination			register 'f'	Data	destination
If sk	•	0.0	00	0.4	If s	kip:		_	_
	Q1	Q2	Q3	Q4		Q1	Q2	Q3	Q4
	No operation	No operation	No operation	No operation		No operation	No operation	No operation	No operation
lf sk		d by 2-word in:		operation	If c	kip and followe			operation
	Q1	Q2	Q3	Q4	11 3	Q1	Q2	Q3	Q4
	No	No	No	No.		No	No	No	No No
	operation	operation	operation	operation		operation	operation	operation	operation
	No	No	No	No		No	No	No	No
	operation	operation	operation	operation		operation	operation	operation	operation
Exan	nple:	NZERO	INCFSZ CN : :	JT, 1, 0	<u>Exa</u>	mple:	HERE ZERO NZERO	INFSNZ REG	3, 1, 0
	Before Instruc	tion				Before Instruc	tion		
	PC	= Address	(HERE)			PC	= Address	S (HERE)	
	After Instruction					After Instruction			
	CNT If CNT	= CNT + 7 = 0;	1			REG If REG	= REG + ≠ 0;	I	
	PC	= Address	S (ZERO)			PC	= Address	s (NZERO)	
	If CNT PC	≠ 0; = Address	(NZERO)			If REG PC	= 0; = Address	s (ZERO)	
	. 0	, , , , , , , , , , , , , , , , , , , ,	(1121110)			. •			

Data

IORLW Inclusive OR Literal with W									
Synta	ax:	IORLW k	IORLW k						
Oper	ands:	$0 \le k \le 255$	$0 \le k \le 255$						
Oper	ation:	(W) .OR. $k \rightarrow W$							
Status Affected: N, Z									
Enco	ding:	0000	1001	kkkk	kkkk				
Desc	ription:	The contereight-bit lite			d with the t is placed in				
Word	ls:	1	1						
Cycle	es:	1	1						
QC	ycle Activity:								
	Q1	Q2	Q2 Q3 Q4						
	Decode	Read	Proce	ess	Write to W				

Example: IORLW 35h

literal 'k'

Before Instruction W = 9Ah After Instruction

W = BFh

IORV	VF	Inclusive OR W with f						
Synta	ax:	IORWF f	IORWF f {,d {,a}}					
Oper	ands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$	d ∈ [0,1]					
Oper	ation:	(W) .OR. (f)	$\rightarrow$ dest					
Statu	s Affected:	N, Z						
Enco	ding:	0001	00da	ffff	ffff			
Desc	ription:	'0', the result is (default). If 'a' is '0', t If 'a' is '1', t GPR bank If 'a' is '0' a set is enabl in Indexed mode wher Section 26	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 ≤ 95 (5Fh). See Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed					
Word	ls:	1	1					
Cycle	es:	1						
QC	ycle Activity:							
	Q1	Q2	Q3		Q4			
	Decode	Read register 'f'	Process Data	-	rite to stination			

LFSR	Load FSF	₹			
Syntax:	LFSR f,	k			
Operands:	$0 \le f \le 2$ $0 \le k \le 40$	95			
Operation:	$k \to FSRf$				
Status Affected:	None				
Encoding:	1110 1111	1110 0000	00ff k <sub>7</sub> kkk	k <sub>11</sub> 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:

Q1	Q2	Q3	Q4
Decode	Read literal	Process	Write
	'k' MSB	Data	literal 'k' MSB
			to FSRfH
Decode	Read literal	Process	Write literal 'k'
	'k' LSB	Data	to FSRfL

Example: LFSR 2, 3ABh

After Instruction

FSR2H = 03h FSR2L = ABh

MOVF	Move f
Syntax:	MOVF f {,d {,a}}
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	$f \to dest$
Status Affected:	N, Z
Encoding:	0101 00da ffff ffff
	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 '0', 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 26.2.3 "Byte-Oriented and

Words: 1
Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read	Process	Write W
	register 'f'	Data	

**Bit-Oriented Instructions in Indexed** 

Literal Offset Mode" for details.

Example: MOVF REG, 0, 0

Before Instruction

REG = 22h W = FFh After Instruction REG = 22h W = 22h

MOVFF	Move f to	f			
Syntax:	MOVFF 1	f <sub>s</sub> ,f <sub>d</sub>			
Operands:	$0 \le f_s \le 4095$ $0 \le f_d \le 4095$				
Operation:	$(f_s) \to f_d$				
Status Affected:	None				
Encoding: 1st word (source) 2nd word (destin.)	1100 1111	ffff ffff	ffff ffff	ffff <sub>s</sub> ffff <sub>d</sub>	
Description:	The contents of source register 'f <sub>s</sub> ' are				

moved to destination 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.

Either source or destination can be W (a useful special situation).

MOVFF is particularly useful for transferring a data memory location to a peripheral register (such as the transmit

buffer or an I/O port).
The MOVFF instruction cannot use the PCL, TOSU, TOSH or TOSL as the

destination register.

Words: 2 Cycles: 2

Q Cycle Activity:

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

 $\begin{array}{lll} \mathsf{REG1} & = & 33\mathsf{h} \\ \mathsf{REG2} & = & 11\mathsf{h} \end{array}$ 

After Instruction

REG1 = 33h REG2 = 33h

MOV	'LB	Move Literal to Low Nibble in BSR				
Synta	ax:	MOVLW F	Κ.			
Oper	rands:	$0 \le k \le 255$	;			
Oper	ration:	$k\to BSR$				
Statu	us Affected:	None				
Enco	oding:	0000	0001	kkkk	kkkk	
Desc	cription:	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 <sub>7</sub> :k <sub>4</sub> .				
Word	ds:	1				
Cycle	es:	1				
QC	cycle Activity:					
	Q1	Q2	Q3	}	Q4	
ļ	Decode	Read literal 'k'	Proce Data		ite literal to BSR	

Example: MOVLB 5

Before Instruction

BSR Register = 02h

After Instruction

BSR Register = 05h

MOVLW	Move Lite	ral to W		
Syntax:	MOVLW	k		
Operands:	$0 \le k \le 25$	5		
Operation:	$k\toW$			
Status Affected:	None			
Encoding:	0000	1110	kkkk	kkkk
Description:	The eight-	bit literal '	k' is loade	d 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:	$0 \le f \le 255$ $a \in [0,1]$
Operation:	$(W) \rightarrow f$
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 26.2.3 "Byte-Oriented and

Words: 1
Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read	Process	Write
	register 'f'	Data	register 'f'

**Bit-Oriented Instructions in Indexed** 

Literal Offset Mode" for details.

Example: MOVWF REG, 0

Before Instruction

W = 4Fh REG = FFh

After Instruction

W = 4FhREG = 4Fh

MUL	LW	Multiply L	iteral with	ı W				
Synta	ax:	MULLW	MULLW k					
Oper	ands:	$0 \le k \le 255$	;					
Oper	ation:	(W) x k $\rightarrow$	PRODH:P	RODL				
Statu	s Affected:	None						
Enco	ding:	0000	1101	kkkk	kkkk			
Desc	ription:	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.						
Word	ls:	1						
Cycle	es:	1						
QC	ycle Activity:							
	Q1	Q2	Q3		Q4			
	Decode	Read literal 'k'	Proces Data	re PI	Write gisters RODH:			

Example:	MULLW	0C4h	
Before Instruction	า		
W PRODH	= =	E2h ?	
PRODL After Instruction	=	?	
W PRODH PRODL	= = =	E2h ADh 08h	

MUL	WF	Multiply W with f					
Synta	ax:	MULWF	f {,a}				
Oper	ands:	$0 \le f \le 255$	5				
		a ∈ [0,1]					
Oper	ation:	(W) x (f) —	→ PRODH:PI	RODL			
Statu	s Affected:	None	None				
Enco	oding:	0000	001a f	fff ffff			
Desc	ription:	out between register filter result is storegister partially high byte. Unchanged None of the Note that repossible in result is possible in result is possible in to select the feat is '0', selected. It to select the feat is enable operates in Addressing f≤95 (5FF)	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. 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 26.2.3 "Byte-Oriented and Bit-Oriented				
Word	ds:	1					
Cycle	es:	1	1				
QC	ycle Activity:						
	Q1	Q2	Q3	Q4			
	Decode	Read register 'f'	Process Data	Write registers PRODH: PRODL			

Example:	MULWF	REG,	1
Before Instruction	า		
W	=	C4h	
REG	=	B5h	
PRODH	=	?	
PRODL	=	?	
After Instruction			
W	=	C4h	
REG	=	B5h	
PRODH	=	8Ah	
PRODL	=	94h	

NEGF	Negate f			
Syntax:	NEGF f	{,a}		
Operands:	$0 \le f \le 255$ $a \in [0,1]$			
Operation:	$(\bar{f}) + 1 \to f$			
Status Affected:	N, OV, C, DC, Z			
Encoding:	0110	110a	ffff	ffff
Description:	Location 'f' is negated using two's complement. The result is placed in the 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 ≤ 95 (5Fh). See  Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.			
Manala.	4			

Words: 1
Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read	Process	Write
	register 'f'	Data	register 'f'

Example: NEGF REG, 1

Before Instruction

REG = 0011 1010 [3Ah]

After Instruction

REG = 1100 0110 [C6h]

NOP		No Operation				
Synta	ax:	NOP				
Oper	ands:	None				
Oper	ation:	No operation				
Statu	Status Affected: None					
Enco	oding:	0000 1111	0000 xxxx	000 xxx	-	0000 xxxx
Desc	ription:	No operati	on.			
Word	ls:	1				
Cycle	es:	1				
QC	ycle Activity:					
	Q1	Q2	Q	3		Q4
	Decode	No	No			No
		operation	operat	tion	ор	eration

Example:

None.

POP	Pop Top of Return Stack				
Syntax:	POP				
Operands:	None				
Operation:	$(TOS) \to b$	it bucket			
Status Affected:	None				
Encoding:	0000 0000 0000 0110				
Description:  The TOS value is pulled off the return stack and is discarded. The TOS value the previous value the 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.			OS value lue that ack. enable ne return		
Words:	1				
Cycles:	1				
Q Cycle Activity:					
Q1	Q2	Q3		Q4	
Decode	No operation	Pop TOS value		No eration	

Example: POP GOTO NEW

Before Instruction

TOS = 0031A2h Stack (1 level down) = 014332h

After Instruction

TOS = 014332h PC = NEW

PUS	Н	Push Top of Return Stack				
Synta	ax:	PUSH				
Oper	ands:	None				
Oper	ation:	(PC + 2) →	$(PC + 2) \rightarrow TOS$			
Statu	s Affected:	None	None			
Enco	ding:	0000 0000 0000 0101				
Desc	ription:	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.				
Word	ls:	1				
Cycle	es:	1				
QC	ycle Activity:					
	Q1	Q2	Q3	i	Q4	
	Decode	Push PC + 2 onto return stack	No operat	ion	No operation	

After Instruction

PC = 0126h
TOS = 0126h
Stack (1 level down) = 345Ah

RCALL	Relative Call			
Syntax:	RCALL n			
Operands:	$-1024 \le n \le 1023$			
Operation:	$ (PC) + 2 \rightarrow TOS, $ $ (PC) + 2 + 2n \rightarrow PC $			
Status Affected:	None			
Encoding:	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			
O Cycle Activity				

Q Cycle Activity:

	Q1	Q2	Q3	Q4
	Decode	Read literal 'n'	Process Data	Write to PC
		Push PC to stack		
Ī	No	No	No	No
	operation	operation	operation	operation

Example: HERE RCALL Jump

Before Instruction

PC = Address (HERE)

After Instruction

PC = TOS = Address (Jump) Address (HERE + 2)

RES	ET	Reset				
Synta	ax:	RESET				
Oper	ands:	None				
Oper	ation:		Reset all registers and flags that are affected by a MCLR Reset.			
Statu	s Affected:	All				
Enco	ding:	0000	0000	111	.1	1111
Desc	ription:	This instruction				•
Word	ls:	1				
Cycle	es:	1				
QC	ycle Activity:					
	Q1	Q2	Q3	3		Q4
	Decode	Start	No			No
		Reset	onerat	ion	οn	eration

RESET Example:

After Instruction

Registers = Flags\* = Reset Value Reset Value

RETFIE	Return from Interrupt			
Syntax:	RETFIE {s}			
Operands:	$s \in [0,1]$			
Operation:	$ \begin{split} &(TOS) \to PC, \\ &1 \to GIE/GIEH \text{ or } PEIE/GIEL, \\ &\text{if } s = 1 \\ &(WS) \to W, \\ &(STATUSS) \to STATUS, \\ &(BSRS) \to BSR, \\ &PCLATU,  PCLATH \text{ are unchanged} \end{split} $			
Status Affected:	GIE/GIEH, PEI	E/GIE	EL.	
Encoding:	0000 000	00	0001	000s
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, STATUSS and BSRS are loaded into their corresponding registers, W, STATUS and BSR. If 's' = 0, no update of these registers occurs (default).			

Words: 1 Cycles: 2

#### Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	Pop PC from stack
			Set GIEH or GIEL
No	No	No	No
operation	operation	operation	operation

Example: RETFIE 1

After Interrupt

RETLW	Return Literal to W		
Syntax:	RETLW k		

 $\begin{array}{ll} \text{Operands:} & 0 \leq k \leq 255 \\ \text{Operation:} & k \rightarrow W, \\ & (\text{TOS}) \rightarrow \text{PC}, \end{array}$ 

PCLATU, PCLATH are unchanged

Status Affected: None

Encoding: 0000 1100 kkkk kkkk

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 Cycles: 2

#### Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read	Process	Pop PC from
	literal 'k'	Data	stack, Write
			to W
No	No	No	No
operation	operation	operation	operation

#### Example:

```
CALL TABLE ; W contains table ; offset value ; W now has ; table value : 
TABLE ADDWF PCL ; W = offset RETLW k0 ; Begin table RETLW k1 ; : :
```

; End of table

Before Instruction W = 07h

RETLW kn

After Instruction W = value of kn

RET	JRN	Return from Subroutine		
Synta	ax:	RETURN	{s}	
Oper	ands:	$s\in \left[ 0,1\right]$		
Oper	ation:	$(TOS) \rightarrow PC,$ if s = 1 $(WS) \rightarrow W,$ $(STATUSS) \rightarrow STATUS,$ $(BSRS) \rightarrow BSR,$ PCLATU, PCLATH are unchanged		
Statu	s Affected:	None		
Enco	coding: 0000 0000 0001 001s			
Desc	ription:	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, STATUSS and BSRS are loaded into their corresponding registers, W, STATUS and BSR. If 's' = 0, no update of these registers occurs (default).		
Word	ls:	1		
Cycle	es:	2		
Q Cycle Activity:				
	Q1	Q2	Q3	Q4
	Decode	No	Process	Pop PC
		operation	Data	from stack

Example: RETURN

No

operation

No

operation

No

operation

No

operation

After Instruction: PC = TOS

RLCF		Rotate Left f through Carry				
Syntax	x:	RLCF f	{,d {,a}}			
Opera	ınds:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$	. , .			
Opera	tion:	$(f<7>) \rightarrow C$	$(f) \rightarrow dest,  (f<7>) \rightarrow C,  (C) \rightarrow dest<0>$			
Status	Affected:	C, N, Z				
Encod	ling:	0011	01da	ffff	ffff	
Descri	iption:	one bit to the flag. If 'd' is W. If 'd' is in register If 'a' is 'o', selected. If select the left 'a' is 'o' as set is enable operates in Addressing f ≤ 95 (5Fh "Byte-Orie Instruction	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 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.			
Words	s:	1				
Cycles	s:	1				
Q Cy	cle Activity:					
_	Q1	Q2	Q	-	Q4	
	Decode	Read register 'f'	Proce Data		Write to estination	

RLNCF	Rotate Lef	t f (No Carry)	)	RRCF	Rotate Rig	ht f through	Carry
Syntax:	RLNCF	f {,d {,a}}		Syntax:	RRCF f{	,d {,a}}	
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$			Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$		
Operation:	$(f< n>) \rightarrow de$ $(f< 7>) \rightarrow de$	•		Operation:	$(f<0>) \rightarrow C$	•	
Status Affecte	d: N, Z		1	Ctatus Affactado	(C) → dest	<1>	
Encoding:	0100	01da ff	ff ffff	Status Affected:	C, N, Z		
Description:		-		Encoding:	0011		ff ffff
Waste	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 ≤ 95 (5Fh). See  Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.  register f		Description:	The contents of register 'f' are roone bit to the right through the C flag. If 'd' is '0', the result is place If 'd' is '1', the result is placed by register 'f' (default).  If 'a' is '0', the Access Bank is set If 'a' is '1', the BSR is used to set GPR bank (default).  If 'a' is '0' and the extended instract is enabled, this instruction op in Indexed Literal Offset Address mode whenever f ≤ 95 (5Fh). Se Section 26.2.3 "Byte-Oriented Bit-Oriented Instructions in Indicated Literal Offset Mode" for details.		gh the Carry is placed in W. laced back in ank is selected. ed to select the ded instruction uction operates Addressing 5Fh). See riented and ns in Indexed r details.	
Words:	1				C	registe	ert
Cycles:	1			Words:	1		
Q Cycle Activ	•	00	0.4	Cycles:	1		
Q1 Decod	Q2 de Read	Q3 Process	Q4 Write to	Q Cycle Activity:			
Decoc	register 'f'	Data	destination	Q1	Q2	Q3	Q4
				Decode	Read	Process	Write to
Example:	RLNCF	REG, 1,	0		register 'f'	Data	destination
Before Instruction  REG = 1010 1011		Example:	RRCF	REG, 0,	0		
After Instruction			Before Instru	uction			
REG	G = 0101 0	111		REG C	= 1110 = 0	0110	
				After Instruc REG W C	tion = 1110 = 0111 = 0		

RRN	CF	Rotate Rig	ht f (No	Carry	)	
Synta	ax:	RRNCF 1	{,d {,a}}			
Oper	rands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$				
Oper	ration:	` '	$(f) \rightarrow dest,$ $(f<0>) \rightarrow dest<7>$			
Statu	is Affected:	N, Z				
Enco	oding:	0100	00da	fff	f ffff	
Desc	oription:	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). 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). 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 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.				
Word	ds:	1				
Cycle	es:	1				
QC	ycle Activity:					
	Q1	Q2	Q3	3	Q4	
	Decode	Read	Proce	SS	Write to	

SETI	F	Set f					
Synta	ax:	SETF f {,	a}				
Oper	ands:	$0 \le f \le 255$ $a \in [0,1]$	* - : * *				
Oper	ation:	$FFh \to f$					
Statu	s Affected:	None					
Enco	oding:	0110	100a	fff	f fff	f	
Desc	ription:	are set to F If 'a' is '0', 1 If 'a' is '1', 1 GPR bank If 'a' is '0' a set is enab in Indexed mode wher Section 26 Bit-Oriente	The contents of the specified register are set to FFh.  If 'a' is 'o', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).  If 'a' is 'o' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See  Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.				
Word	ds:	1					
Cycle	es:	1					
QC	ycle Activity:						
	Q1	Q2	Q3	1	Q4		
	Decode	Read	Proce		Write		
		register 'f'	Data	a	register	'f'	
<u>Exan</u>	nple:	SETF	RE	G,1			

Before Instruction REG 5Ah After Instruction REG FFh

register 'f'

RRNCF

RRNCF

=

1101 0111

1110 1011

Example 1:

Example 2:

Before Instruction REG

After Instruction REG

Before Instruction

Data

REG, 1, 0

REG, 0, 0

destination

SLEE	ΞP	Enter Sleep mode				
Synta	ax:	SLEEP				
Oper	ands:	None				
Oper	ration:		$0 \to \overline{PD}$			
Status Affected: TO, PD						
Enco	ding:	0000 0000 0000 0011				0011
Desc	cription:	cleared. Ti is set. Wat postscaler The proce	The Power-Down status bit (\$\overline{PD}\$) is cleared. The Time-out status bit (\$\overline{TO}\$) is set. Watchdog Timer and its postscaler are cleared. The processor is put into Sleep mode with the oscillator stopped.			
Word	is:	1				
Cycle	es:	1	1			
Q C	ycle Activity:					
	Q1	Q2	Q3			Q4
	Decode	No operation	Proces Data			Go to Sleep

Example:		SLEEP
Before Ins	truction	
TO	= ?	
PD	= ?	
After Instru	uction	
TO	= 1	t
PD	= 0	

† If WDT causes wake-up, this bit is cleared.

SUBFWB	Subtract 1	from W with	Borrow	
Syntax:	SUBFWB	f {,d {,a}}		
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$			
Operation:	(W) - (f) -	$(\overline{C}) \to \text{dest}$		
Status Affected:	N, OV, C,	DC, Z		
Encoding:	0101	01da ffi	f 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 ≤ 95 (5Fh). See Section 26.2.3 "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 Instruct REG W C	tion = 3 = 2 = 1	1125, 1, 6		
After Instructio REG W C Z N	= FF = 2 = 0 = 0	sult is negative	e	
Example 2:	SUBFWB	REG, 0, 0		
Before Instruct REG W C	tion = 2 = 5 = 1			
After Instruction REG W C Z N	= 2 = 3 = 1 = 0	sult is positive		
Example 3:	SUBFWB	REG, 1, 0		
Before Instruc	tion	. , -		
REG W C	= 1 = 2 = 0			
After Instructio REG W C Z	on = 0 = 2 = 1			

© 2006 Microchip Technology Inc.

SUBL	w	Subtract W from Literal				
Syntax	x:	SUBLW I	SUBLW k			
Opera	nds:	$0 \le k \le 25$	$0 \le k \le 255$			
Opera	tion:	$k-(W)\rightarrow$	$k - (W) \rightarrow W$			
Status	Affected:	N, OV, C, DC, Z				
Encod	ling:	0000 1000 kkkk kkkk				
Descri	iption	W is subtracted from the eight-bit literal 'k'. The result is placed in W.				
Words	s:	1				
Cycles	s:	1				
Q Cy	cle Activity:					
_	Q1	Q2	Q3		Q4	
	Decode	Read literal 'k'	Proces Data		Vrite to W	
Exam	ole 1:	SUBLW (	02h			

Before Instruct	tion		
W	=	01h	
С	=	?	
After Instructio	n		
W	=	01h	
С	=	1	; result is positive
Z	=	0	
N	=	0	
Example 2:	S	UBLW	02h
Before Instruct	tion		
W	=	02h	
С	=	?	
A 44   44 44 -	_		

After Instruction
W =
C =
Z =
N = 00h 1 1 0 ; result is zero

Example 3:

SUBLW 02h Before Instruction 03h ? W C After Instruction

W = C = Z = N = N FFh ; (2's complement) 0 ; result is negative 0 = =

SUBWF	Subtract \	N from f	
Syntax:	SUBWF	f {,d {,a}}	
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$		
Operation:	(f) - (W) -	o dest	
Status Affected:	N, OV, C,	DC, Z	
Encoding:	0101	11da ffi	ff 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).  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 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details.			
Words:	1	uctans.	
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 Instruc REG W C After Instructio REG W	= 3 = 2 = ? on = 1 = 2		
C Z	= 0	esult is positive	<b>)</b>
N Example 2:	= 0	DEC 0 0	
Example 2: Before Instruc	SUBWF tion	REG, 0, 0	
REG W C After Instructio REG W C Z	= 2 = 2 = ? on = 2 = 0	esult is zero	
Example 3:	= U SUBWF	REG, 1, 0	
Before Instruction REG W C After Instruction REG W C Z N	tion = 1 = 2 = ? on = FFh;(2	REG, 1, 0 's complemenesult is negative	,

SUBWFB	Subtract	W from f with	Borrow
Syntax:	SUBWFB	f {,d {,a}}	
Operands:	$0 \le f \le 255$	5	
	$d \in [0,1]$		
0	$a \in [0,1]$	( <del>0</del> )	
Operation:		$(\overline{C}) \rightarrow \text{dest}$	
Status Affected:	N, OV, C,	ı	
Encoding:	0101		ff ffff
Description:  Words:	from regis method). I in W. If 'd' in register If 'a' is '0', If 'a' is '1', GPR bank If 'a' is '0' set is enal in Indexed mode whe Section 2 Bit-Orient	'f' (default). the Access Ba the BSR is use (default). and the extend	olement esult is stored t is stored back ank is selected. ed to select the led instruction action operates Addressing iFh). See riented and as in Indexed
Cycles:	1		
Q Cycle Activity:	00	02	04
Q1 Decode	Q2 Read	Q3 Process	Q4 Write to
Decode	register 'f'	Data	destination
Example 1:	SUBWFB	REG, 1, 0	
Before Instruct			
REG W C	= 19h = 0Dh = 1	(0001 10 (0000 11	001) 101)
After Instructio REG	n = 0Ch	(0000 10	)11)
W	= 0Dh	(0000 10	•
C Z	= 1 = 0		
N	= 0	; result is p	oositive
Example 2:	SUBWFB	REG, 0, 0	
Before Instruct REG	tion = 1Bh	(0001 10	)11)
W	= 1Ah		10)
C After Instructio	= 0		
REG	= 1Bh	(0001 10	)11)
W C	= 00h = 1		
Z	= 1	; result is z	zero
N	= 0		
Example 3:	SUBWFB	REG, 1, 0	
Before Instruct REG W	= 03h = 0Eh	(0000 00 (0000 11	•
C After Instructio	= 1 n		
REG	= F5h	(1111 01	
W	= 0Eh	; [2's comp	
Č Z	= 0	(0000 11	/
	= 0		

Synta	ax:	SWAPF f	{.d {.a}}				
•	rands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$	$d \in [0,1]$ $a \in [0,1]$				
Oper	ration:		$(f<3:0>) \to dest<7:4>,  (f<7:4>) \to dest<3:0>$				
Statu	us Affected:	None					
Enco	oding:	0011	10da	ffff	ffff		
		GPR bank If 'a' is '0' a set is enab in Indexed mode wher Section 26 Bit-Oriente	egister 'f' (d the Access he BSR is u (default). and the exte led, this ins Literal Offsonever $f \le 95$	efault). Bank is used to ended in truction et Addr (5Fh). Oriento	selected select the estruction operates essing See ed and Indexed		
Word	de.	1	set would	ioi dete	uio.		
Cycle		1					
-	cycle Activity:	•					
u o	Q1	Q2	Q3		Q4		
	Decode	Read register 'f'	Process Data		Vrite to stination		
Exar	nple:	SWAPF I	REG, 1, 0	)			

TBL	RD	Table Read					
Synta	ax:	TBLRD ( *;	*+; *	-; +*)			
Oper	ands:	None					
Oper	ation:	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					
Statu	s Affected:	None					
Enco	ding:	0000	0	000	000	00	10nn nn=0 * =1 *+ =2 *- =3 +*
2000	ription:	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.  The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range.  TBLPTR[0] = 0: Least Significant Byte of Program Memory Word  TBLPTR[0] = 1: Most Significant Byte of Program Memory Word  The TBLRD instruction can modify the value of TBLPTR as follows:					
		<ul> <li>no chang</li> </ul>		ot.			
		<ul><li>post-incre</li><li>post-decr</li></ul>					
		pre-increment					
Word	ls:	1					
Cycle	es:	2					
	ycle Activity:	:					
_ ,	Q1	Q2		C	)3		Q4
	Decode	No		N	0 ation		No

operation

No operation

(Read Program Memory)

No

operation

operation

No

operation

operation

No operation

(Write TABLAT)

TBLRD	Table Read (	Cont	inued)
Example 1:	TBLRD *+	;	
Before Instructi TABLAT TBLPTR MEMORY After Instructior TABLAT TBLPTR	(00A356h)	= = = =	55h 00A356h 34h 34h 00A357h
Example 2:	TBLRD +*	;	
Before Instructi TABLAT TBLPTR MEMORY MEMORY After Instructior TABLAT TBLPTR	(01A357h) (01A358h)	= = = = =	AAh 01A357h 12h 34h 34h 01A358h

TBLWT	Table Writ	te		
Syntax:	TBLWT (*	; *+; *-; +*)	)	
Operands:	None			
Operation:	if TBLWT*			
	(TABLAT)	-		
	TBLPTR – if TBLWT*-		je;	
	(TABLAT)	$\rightarrow$ Holding		
	(TBLPTR)	$+ 1 \rightarrow TBI$		
	if TBLWT*-		iotor	
	(TABLAT) (TBLPTR)			
	if TBLWT+	-*		
	(TBLPTR)			
3.55	(TABLAT)	→ Holding	, Register	
Status Affected:	None	· · · · ·	·	<del> </del>
Encoding:	0000	0000	0000	11nn nn=0 *
		1		nn=0 * =1 *+
				=2 *-
			<u> </u> !	=3 +*
Description:				s of TBLPTR
	to determine 8 holding r			is written to.
	The holdin	ng registers	s are used	to
	program th	ne contents	s of Progra	am
	Memory (F			
	"Flash Prodetails on			
	The TBLP	TR (a 21-b	oit pointer)	points to
	each byte	in the prog	gram memo	ory. TBLPTR
				The LSb of
	the TBLPT program m		•	
	. •	PTR[0] = 0:	: Least Si	ignificant
		-	Byte of I	Program / Word
	TBLP	PTR[0] = 1:	Most Sig	gnificant Program
			Byte of I Memory	Program Word
	The TBLW		on can mó	
	value of TE		follows:	
	<ul><li>no chan</li><li>post-inc</li></ul>	-		
	•	crement		
	pre-incre			
Words:	1			
Cycles:	2			
Q Cycle Activity:				
-	Q1	Q2	Q3	Q4
	Decode	No	No	No
		· .	operation	operation
	No	No	No operation	No
	operation	operation (Read	operation	operation (Write to
		TABLAT)		Holding
		1		Register)

TBLWT	Table Write(C	ontir	nued)
Example 1:	TBLWT *+;		
(00A356 After Instructio TABLAT TBLPTR	G REGISTER Sh) ons (table write	= = comp = =	55h 00A356h FFh oletion) 55h 00A357h
(00A356	Sh)	=	55h
Before Instruction TABLAT TBLPTR HOLDING (01389) HOLDING (01389B) After Instruction TABLAT TBLPTR HOLDING (01389A)	G REGISTER Ah) G REGISTER Bh) on (table write of G REGISTER th) G REGISTER	= = = = = = = = = = = = = = = = = = =	34h 01389Ah FFh FFh etion) 34h 01389Bh FFh 34h

Register)

TSTFSZ	Test f, Skip if 0				
Syntax:	TSTFSZ f	TSTFSZ f {,a}			
Operands:	$0 \le f \le 255$ $a \in [0,1]$	$0 \le f \le 255$ $a \in [0,1]$			
Operation:	skip if f = 0	skip if $f = 0$			
Status Affected:	None	None			
Encoding:	0110	011a	ffff	ffff	
Description:	If 'f' = 0, the next instruction fetched during the current instruction execution is discarded and a NOR is executed.				

If 'f' = 0, the next instruction fetched during the current instruction execution is discarded and a  $\mathtt{NOP}$  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 'o' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f  $\leq$  95 (5Fh). See Section 26.2.3 "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	Process	No
	register 'f'	Data	operation

#### If skip:

Q1	Q2	Q3	Q4
No	No	No	No
operation	operation	operation	operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No	No	No	No
operation	operation	operation	operation
No	No	No	No
operation	operation	operation	operation

Example: HERE TSTFSZ CNT, 1

NZERO : ZERO :

Before Instruction

PC = Address (HERE)

After Instruction

If CNT = 00h,

PC = Address (ZERO)

 $f CNT \neq 00h$ ,

PC = Address (NZERO)

XOR	LW	Exclusive OR Literal with W				
Synta	ax:	XORLW	k			
Operands:		$0 \le k \le 255$	$0 \le k \le 255$			
Oper	ation:	(W) .XOR.	(W) .XOR. $k \rightarrow W$			
Statu	s Affected:	N, Z				
Enco	ding:	0000	1010	kkkk	kkkk	
Desc	ription:	The content the 8-bit lit in W.				
Word	ls:	1	1			
Cycle	es:	1				
Q Cycle Activity:						
	Q1	Q2	Q3		Q4	
	Decode	Read literal 'k'	Proces Data		rite to W	

Example: XORLW 0AFh

Before Instruction

W = B5h

After Instruction

W = 1Ah

XORWF	Exclusive OR	W	with 1	f

Syntax: XORWF  $f \{ d \{,a\} \}$ 

Operands:  $0 \le f \le 255$  $d \in [0,1]$ 

 $a \in [0,1]$ 

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 ≤ 95 (5Fh). See Section 26.2.3 "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	Process	Write to
	register 'f'	Data	destination

Example: XORWF REG, 1, 0

Before Instruction

REG = AFh W = B5h After Instruction

REG = 1Ah W = B5h

#### 26.2 Extended Instruction Set

In addition to the standard 75 instructions of the PIC18 instruction set, PIC18F2455/2550/4455/4550 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 mode for many of the standard PIC18 instructions.

The additional features of the extended instruction set are disabled by default. To enable them, users must set the XINST Configuration bit.

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 26-3. Detailed descriptions are provided in **Section 26.2.2** "Extended Instruction Set". The opcode field descriptions in Table 26-1 (page 308) 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.

#### 26.2.1 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<sup>TM</sup> 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 26.2.3.1 "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 26-3: EXTENSIONS TO THE PIC18 INSTRUCTION SET

Mnemonic,		Description	Cycles	16-E	16-Bit Instruction Word		Status	
Operar	nds	Description	Cycles	MSb			LSb	Affected
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 <sub>s</sub> (source) to 1st word	2	1110	1011	0zzz	ZZZZ	None
		f <sub>d</sub> (destination) 2nd word		1111	ffff	ffff	ffff	
MOVSS	$z_s, z_d$	Move z <sub>s</sub> (source) to 1st word	2	1110	1011	1zzz	ZZZZ	None
		z <sub>d</sub> (destination) 2nd word		1111	xxxx	XZZZ	ZZZZ	
PUSHL	k	Store literal at FSR2,	1	1110	1010	kkkk	kkkk	None
		decrement FSR2						
SUBFSR	f, k	Subtract literal from FSR	1	1110	1001	ffkk	kkkk	None
SUBULNK	k	Subtract literal from FSR2 and	2	1110	1001	11kk	kkkk	None
		return						

#### **EXTENDED INSTRUCTION SET** 26.2.2

ADDFSR	Add Literal to FSR				
Syntax:	ADDFSF	ADDFSR f, k			
Operands:	$0 \le k \le 63$				
	$f \in [0, 1, 2]$				
Operation:	$FSR(f) + k \rightarrow FSR(f)$				
Status Affected:	None				
Encoding:	1110	1000	ffkk	kkkk	
Description:	The 6-bit literal 'k' is added to the contents of the FSR specified by 'f'.				
Words:	1				
0	4				

Cycles:

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read	Process	Write to
	literal 'k'	Data	FSR

Example: ADDFSR 2, 23h

Before Instruction

FSR2 03FFh =

After Instruction

FSR2 0422h

ADDULNK	Add Literal to FSR2 and Return				
Syntax:	ADDULN	K k			
Operands:	$0 \le k \le 63$				
Operation:	$FSR2 + k \to FSR2,$				
	$(TOS) \to PC$				
Status Affected:	None				
Encoding:	1110	1000	11kk	kkkk	
Description:	The 6-bit literal 'k' is added to the contents of 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 ADDFSR instruction, where f = 3 (binary '11'); it operates only on FSR2.				
Words:	1				

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read	Process	Write to
	literal 'k'	Data	FSR
No	No	No	No
Operation	Operation	Operation	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 syntax then becomes: {label} instruction argument(s).

CALLW	Subroutine Call Using WREG			
Syntax:	CALLW			
Operands:	None			
Operation:	$(PC + 2) \rightarrow TOS,$ $(W) \rightarrow PCL,$ $(PCLATH) \rightarrow PCH,$ $(PCLATU) \rightarrow PCU$			
Status Affected:	None			
Encoding:	0000	0000	0001	0100
Description	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. Unlike CALL, there is no option to update W, STATUS or BSR.			

Cycles:

Q Cycle Activity:

Words:

Q1	Q2	Q3	Q4
Decode	Read	Push PC to	No
	WREG	stack	operation
No	No	No	No
operation	operation	operation	operation

Example:	HERE	CALLW

1

2

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 Inde	Move Indexed to f			
Syntax:	MOVSF	MOVSF [z <sub>s</sub> ], f <sub>d</sub>			
Operands:	$0 \le z_s \le 127$ $0 \le f_d \le 4095$				
Operation:	$((FSR2) + z_s) \rightarrow f_d$				
Status Affected:	None				
Encoding: 1st word (source) 2nd word (destin.)	1110 1111	1011 ffff	0zzz ffff	zzzz <sub>s</sub> ffff <sub>d</sub>	
Descriptions	The cente	nto of the		-iotor oro	

Description:

The contents of the source register are moved to destination register 'f<sub>d</sub>'. The actual address of the source register is determined by adding the 7-bit literal offset 'z<sub>s</sub>' in the first word to the value of FSR2. The address of the destination register is specified by the 12-bit literal 'f<sub>d</sub>' 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
ſ	Decode	Determine	Determine	Read
L		source addr	source addr	source reg
Ī	Decode	No operation	No operation	Write register 'f'
		No dummy read		(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

MOVSS	Move Indexed to Indexed			
Syntax:	MOVSS [z <sub>s</sub> ], [z <sub>d</sub> ]			
Operands:	$0 \le z_s \le 127$ $0 \le z_d \le 127$			
Operation:	$((FSR2) + z_s) \rightarrow ((FSR2) + z_d)$			
Status Affected:	None			
Encoding: 1st word (source) 2nd word (dest.)	1110 1011 1zzz zzzzs 1111 xxxx xzzz zzzzd			
Description	The contents of the source register are			

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<sub>s</sub>' or 'z<sub>d</sub>',

respectively, to the value of FSR2. Both registers can be located anywhere in the 4096-byte data memory space (000h to FFFh).

The MOVSS 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. If the resultant destination address points to an indirect addressing register, the instruction will execute as a NOP.

Words: 2 Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Determine	Determine	Read
	source addr	source addr	source reg
Decode	Determine	Determine	Write
	dest addr	dest addr	to dest reg

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

PUSI	НL	S	tore Litera	l at I	FSR2,	Decrem	ent	FSR2
Synta	ax:	Ρ	USHL k					
Oper	ands:	0	≤ k ≤ 255					
Oper	ation:	$k \rightarrow (FSR2),$ FSR2 – 1 $\rightarrow$ FSR2						
Status	s Affected:	Ν	one					
Enco	ding:		1111	10	010	kkkk		kkkk
Desc	ription:	The 8-bit literal 'k' is written to the data memory address specified by FSR2. FS is decremented by '1' after the operation This instruction allows users to push valuents a software stack.			R2. FSR2 peration.			
Word	s:	1						
Cycle	es:	1						
Q Cycle Activity:								
	Q1		Q2		(	Q3		Q4
	Decode		Read 'l	ς'		cess		Write to

Example: PUSHL 08h

Before Instruction

 $\begin{array}{lll} \text{FSR2H:FSR2L} & = & \text{01ECh} \\ \text{Memory (01ECh)} & = & \text{00h} \end{array}$ 

After Instruction

 $\begin{array}{lll} \text{FSR2H:FSR2L} & = & \text{01EBh} \\ \text{Memory (01ECh)} & = & \text{08h} \end{array}$ 

SUBFSR	Subtract Literal from FSR			
Syntax:	SUBFSR f, k			
Operands:	$0 \le k \le 63$			
	$f \in [0, 1, 2]$			
Operation:	$FSRf - k \to FSRf$			
Status Affected:	None			
Encoding:	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			

Cycles:

Q Cycle Activity:

	Q1	Q2	Q3	Q4
	Decode	Read	Process	Write to
L		register 'f'	Data	destination

Example: SUBFSR 2, 23h

Before Instruction

03FFh FSR2

After Instruction

FSR2 03DCh

SUBULNK	Subtract Li	teral from F	SR2 and Re	eturn
Syntax:	SUBULNK	k		
Operands:	$0 \le k \le 63$			
Operation:	FSR2 - k -	→ FSR2		
	$(TOS) \to P$	С		
Status Affected:	None			
Encoding:	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			

Q Cycle Activity: Q1 Q2 Q3 Q4 Decode Read Process Write to register 'f' Data destination No No No No

Operation

Operation

Operation

Example: SUBULNK 23h

2

Before Instruction

Operation

Cycles:

FSR2 03FFh РС 0100h

After Instruction

03DCh FSR2 РС (TOS)

# 26.2.3 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 mode (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 bitoriented 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 26.2.3.1 "Extended Instruction Syntax with Standard PIC18 Commands").

Although the Indexed Literal Offset Addressing 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 Addressing 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.

### 26.2.3.1 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 brackets, will generate an error in the MPASM Assembler.

If the index argument is properly bracketed for Indexed Literal Offset Addressing mode, 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.

# 26.2.4 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 PIC18F2455/2550/4455/4550, 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:  $0 \le k \le 95$   $d \in [0,1]$ 

Operation:  $(W) + ((FSR2) + k) \rightarrow dest$ 

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:

Q1	Q2	Q3	Q4
Decode	Read 'k'	Process	Write to
		Data	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:  $0 \le f \le 95$  $0 \le b \le 7$ 

Operation:  $1 \rightarrow ((FSR2) + k) < b >$ 

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:

Q1	Q2	Q3	Q4
Decode	Read	Process	Write to
	register 'f'	Data	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:  $0 \le k \le 95$ 

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:

	Q1	Q2	Q3	Q4
I	Decode	Decode Read 'k'		Write
			Data	register

Example: SETF [OFST]

Before Instruction

OFST = 2Ch FSR2 = 0A00h Contents of 0A2Ch = 00h

After Instruction Contents

of 0A2Ch = FFh

### 26.2.5 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 of the PIC18F2455/2550/4455/4550 family of devices. 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 mode. 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.

#### 27.0 DEVELOPMENT SUPPORT

The PICmicro<sup>®</sup> 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<sup>™</sup> Object Linker/ MPLIB<sup>™</sup> Object Librarian
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB ICE 4000 In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PICSTART® Plus Development Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration and Development Boards and Evaluation Kits

# 27.1 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 PICmicro 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.

#### 27.2 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for all PICmicro 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

# 27.3 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 family of microcontrollers and dsPIC30F 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.

#### 27.4 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

# 27.5 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

#### 27.6 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PICmicro 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, as well as 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 laboratory environment, making it an excellent, economical software development tool.

# 27.7 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 PICmicro 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 PICmicro 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.

#### 27.8 MPLAB ICE 4000 High-Performance In-Circuit Emulator

The MPLAB ICE 4000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for high-end PICmicro MCUs and dsPIC DSCs. Software control of the MPLAB ICE 4000 In-Circuit Emulator is provided by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 4000 is a premium emulator system, providing the features of MPLAB ICE 2000, but with increased emulation memory and high-speed performance for dsPIC30F and PIC18XXXX devices. Its advanced emulator features include complex triggering and timing, and up to 2 Mb of emulation memory.

The MPLAB ICE 4000 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.

#### 27.9 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 PICmicro MCUs and can be used to develop for these and other PICmicro 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 PICmicro devices.

#### 27.10 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 PICmicro 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.

# 27.11 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 PICmicro 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.

# 27.12 Demonstration, Development and Evaluation Boards

A wide variety of demonstration, development and evaluation boards for various PICmicro 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.

#### 28.0 ELECTRICAL CHARACTERISTICS

#### Absolute Maximum Ratings(†)

Ambient temperature under bias	40°C to +85°C
Storage temperature	65°C to +150°C
Voltage on any pin with respect to Vss (except VDD, MCLR and RA4)	0.3V to (VDD + 0.3V)
Voltage on VDD with respect to Vss	-0.3V to +7.5V
Voltage on MCLR with respect to Vss (Note 2)	0V to +13.25V
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, IiK (VI < 0 or VI > VDD)	±20 mA
Output clamp current, loκ (Vo < 0 or Vo > VDD)	±20 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by all ports	200 mA
Maximum current sourced by all ports	200 mA

- **Note 1:** Power dissipation is calculated as follows: Pdis = VDD x {IDD  $\sum$  IOH} +  $\sum$  {(VDD VOH) x IOH} +  $\sum$ (VOL x IOL)
  - 2: Voltage spikes below Vss at the MCLR/VPP/RE3 pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the MCLR/VPP/RE3 pin, rather than pulling this pin directly to Vss.

† **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.

FIGURE 28-1: PIC18F2455/2550/4455/4550 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)

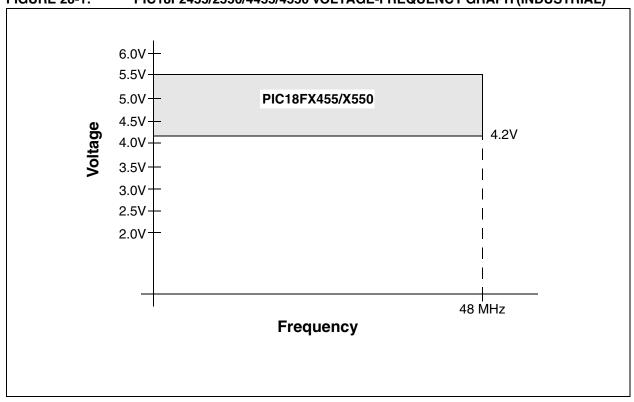
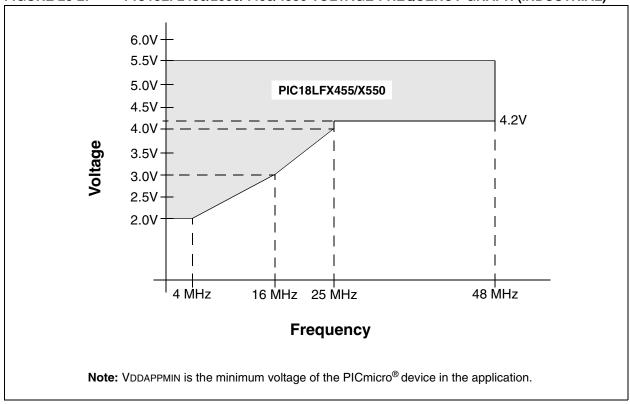


FIGURE 28-2: PIC18LF2455/2550/4455/4550 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)



28.1 DC Characteristics: Supply Voltage

PIC18F2455/2550/4455/4550 (Industrial) PIC18LF2455/2550/4455/4550 (Industrial)

PIC18LF2 (Indus		4455/4550		ard Op		_	ditions (unless otherwise stated) -40°C ≤ TA ≤ +85°C for industrial			
	PIC18F2455/2550/4455/4550 (Industrial)			Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \le \text{Ta} \le +85^{\circ}\text{C}$ for industrial						
Param No.	Symbol	Characteristic	acteristic Min Typ Max Units Condition				Conditions			
D001	VDD	Supply Voltage	2.0	_	5.5	V	EC, HS, XT and Internal Oscillator modes			
			3.0	_	5.5	V	HSPLL, XTPLL, ECPIO and ECPLL Oscillator modes			
D002	VDR	RAM Data Retention Voltage <sup>(1)</sup>	1.5	_	_	V				
D003	VPOR	VDD Start Voltage to ensure internal Power-on Reset signal		_	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	•	•	•	•				
		BORV1:BORV0 = 11	2.00	2.05	2.16	V				
		BORV1:BORV0 = 10	2.65	2.79	2.93	V				
		BORV1:BORV0 = 01	4.11	4.33	4.55	V				
		BORV1:BORV0 = 00	4.36	4.59	4.82	V				

**Legend:** Shading of rows is to assist in readability of the table.

Note 1: This is the limit to which VDD can be lowered in Sleep mode, or during a device Reset, without losing RAM data.

28.2 DC Characteristics: Power-Down and Supply Current

PIC18F2455/2550/4455/4550 (Industrial) PIC18LF2455/2550/4455/4550 (Industrial)

	PIC18LF2455/2550/4455/4550 (Industrial)			Standard Operating Conditions (unless otherwise stated)  Operating temperature $-40^{\circ}\text{C} \le \text{Ta} \le +85^{\circ}\text{C}$ for industrial								
PIC18F24	<b>55/2550/4455/4550</b> trial)		Standard Operating Conditions (unless otherwise stated)  Operating temperature $-40^{\circ}\text{C} \le \text{Ta} \le +85^{\circ}\text{C}$ for industrial									
Param No.	Device	Тур	Typ Max Units Conditions									
	Power-Down Current (IPD) <sup>(1)</sup>											
	PIC18LFX455/X550	0.1	0.95	μΑ	-40°C	.,						
		0.1	1.0	μΑ	+25°C	VDD = 2.0V ( <b>Sleep</b> mode)						
		0.2	5	μΑ	+85°C	(Olcep mode)						
	PIC18LFX455/X550	0.1	1.4	μΑ	-40°C	.,						
		0.1	2	μΑ	+25°C	VDD = 3.0V ( <b>Sleep</b> mode)						
		0.3	8	μΑ	+85°C	(Olcep mode)						
	All devices	0.1	1.9	μΑ	-40°C							
		0.1	2.0	μΑ	+25°C	VDD = 5.0V ( <b>Sleep</b> mode)						
		0.4	15	μΑ	+85°C	(Sicep mode)						

**Legend:** TBD = To Be Determined. Shading of rows is to assist in readability of the table.

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:

- 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: BOR and HLVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.

#### 28.2 DC Characteristics: Power-Down and Supply Current

PIC18F2455/2550/4455/4550 (Industrial)

PIC18LF2455/2550/4455/4550 (Industrial) (Continued)

PIC18LF2 (Indus	2455/2550/4455/4550 trial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \le \text{Ta} \le +85^{\circ}\text{C}$ for industrial  Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \le \text{Ta} \le +85^{\circ}\text{C}$ for industrial								
PIC18F24 (Indus	<b>55/2550/4455/4550</b> trial)										
Param No.	Device	Тур	Max	ns							
	Supply Current (IDD) <sup>(2)</sup>										
	PIC18LFX455/X550	15	32	μΑ	-40°C						
		15	30	μΑ	+25°C	VDD = 2.0V					
		15	29	μΑ	+85°C						
	PIC18LFX455/X550	40	63	μΑ	-40°C		Fosc = 31 kHz				
		35	60	μΑ	+25°C	VDD = 3.0V	(RC_RUN mode,				
		30	57	μΑ	+85°C		INTRC source)				
	All devices	105	168	μΑ	-40°C	_					
		90	160	μΑ	+25°C	VDD = 5.0V					
		80	152	μΑ	+85°C						
	PIC18LFX455/X550	0.33	1	mA	-40°C						
		0.33	1	mA	+25°C	VDD = 2.0V	Fosc = 1 MHz				
		0.33	1	mA	+85°C						
	PIC18LFX455/X550	0.6	1.3	mA	-40°C						
		0.6	1.2	mA	+25°C	VDD = 3.0V	(RC_RUN mode,				
		0.6	1.1	mA	+85°C		INTOSC source)				
	All devices	1.1	2.3	mA	-40°C						
		1.1	2.2	mA	+25°C	VDD = 5.0V					
		1.0	2.1	mA	+85°C						
	PIC18LFX455/X550	0.8	2.1	mA	-40°C						
		0.8	2.0	mA	+25°C	VDD = 2.0V					
		0.8	1.9	mA	+85°C						
	PIC18LFX455/X550	1.3	3.0	mA	-40°C		Fosc = 4 MHz				
		1.3	3.0	mA	+25°C	VDD = 3.0V	(RC_RUN mode,				
		1.3	3.0	mA	+85°C		INTOSC source)				
	All devices	2.5	5.3	mA	-40°C						
		2.5	5.0	mA	+25°C	VDD = 5.0V					
		2.5	4.8	mA	+85°C						

**Legend:** TBD = To Be Determined. Shading of rows is to assist in readability of the table.

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:

- 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: BOR and HLVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.

28.2 DC Characteristics: Power-Down and Supply Current

PIC18F2455/2550/4455/4550 (Industrial)

PIC18LF2455/2550/4455/4550 (Industrial) (Continued)

PIC18LF2 (Indus	<b>455/2550/4455/4550</b> trial)	Standard Operating Conditions (unless otherwise stated)  Operating temperature $-40^{\circ}\text{C} \le \text{Ta} \le +85^{\circ}\text{C}$ for industrial								
PIC18F24 (Indus	<b>55/2550/4455/4550</b> trial)	Standard Operating Conditions (unless otherwise stated)  Operating temperature $-40^{\circ}\text{C} \le \text{TA} \le +85^{\circ}\text{C}$ for industrial								
Param No.	Device	Тур	Max	Units		Condition	าร			
	Supply Current (IDD) <sup>(2)</sup>									
	PIC18LFX455/X550	2.9	8	μΑ	-40°C					
		3.1	8	μΑ	+25°C	VDD = 2.0V				
		3.6	11	μΑ	+85°C					
	PIC18LFX455/X550	4.5	11	μΑ	-40°C		Fosc = 31 kHz			
		4.8	11	μΑ	+25°C	VDD = 3.0V	(RC_IDLE mode,			
		5.8	15	μΑ	+85°C		INTRC source)			
	All devices	9.2	16	μΑ	-40°C					
		9.8	16	μΑ	+25°C	VDD = 5.0V				
		11.4	36	μΑ	+85°C					
	PIC18LFX455/X550	165	350	μΑ	-40°C		Fosc = 1 MHz			
		175	350	μΑ	+25°C	VDD = 2.0V				
		190	350	μΑ	+85°C					
	PIC18LFX455/X550	250	500	μΑ	-40°C					
		270	500	μΑ	+25°C	VDD = 3.0V	(RC_IDLE mode,			
		290	500	μΑ	+85°C		INTOSC source)			
	All devices	0.50	1	mA	-40°C					
		0.52	1	mA	+25°C	VDD = 5.0V				
		0.55	1	mA	+85°C					
	PIC18LFX455/X550	340	500	μΑ	-40°C					
		350	500	μΑ	+25°C	VDD = 2.0V				
		360	500	μΑ	+85°C					
	PIC18LFX455/X550	520	900	μΑ	-40°C	_	Fosc = 4 MHz			
		540	900	μΑ	+25°C	VDD = 3.0V	(RC_IDLE mode,			
		580	900	μΑ	+85°C		INTOSC source)			
	All devices	1.0	1.6	mA	-40°C					
		1.1	1.5	mA	+25°C	VDD = 5.0V				
		1.1	1.4	mA	+85°C					

**Legend:** TBD = To Be Determined. Shading of rows is to assist in readability of the table.

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:

- 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: BOR and HLVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.

#### 28.2 DC Characteristics: Power-Down and Supply Current

PIC18F2455/2550/4455/4550 (Industrial)

PIC18LF2455/2550/4455/4550 (Industrial) (Continued)

PIC18LF2 (Indust	<b>455/2550/4455/4550</b> trial)		Standard Operating Conditions (unless otherwise stated)  Operating temperature $-40^{\circ}\text{C} \le \text{TA} \le +85^{\circ}\text{C}$ for industrial								
PIC18F24 (Indust	<b>55/2550/4455/4550</b> trial)	Standard Operating Conditions (unless otherwise stated)  Operating temperature $-40^{\circ}\text{C} \le \text{TA} \le +85^{\circ}\text{C}$ for industrial									
Param No.	Device	Тур	Max	ns							
	Supply Current (IDD) <sup>(2)</sup>										
	PIC18LFX455/X550	250	500	μΑ	-40°C						
		250	500	μΑ	+25°C	VDD = 2.0V					
		250	500	μΑ	+85°C						
	PIC18LFX455/X550	550	650	μΑ	-40°C		Fosc = 1 MHz				
		480	650	μΑ	+25°C	VDD = 3.0V	(PRI_RUN,				
		460	650	μΑ	+85°C		EC oscillator)				
	All devices	1.2	1.6	mA	-40°C						
		1.1	1.5	mA	+25°C	VDD = 5.0V					
		1.0	1.4	mA	+85°C						
	PIC18LFX455/X550	0.74	2.0	mA	-40°C						
		0.74	2.0	mA	+25°C	VDD = 2.0V					
		0.74	2.0	mA	+85°C						
	PIC18LFX455/X550	1.3	3.0	mA	-40°C		Fosc = 4 MHz				
		1.3	3.0	mA	+25°C	VDD = 3.0V	(PRI_RUN,				
		1.3	3.0	mA	+85°C		EC oscillator)				
	All devices	2.7	6.0	mA	-40°C						
		2.6	6.0	mA	+25°C	VDD = 5.0V					
		2.5	6.0	mA	+85°C						
	All devices	15	35	mA	-40°C						
		16	35	mA	+25°C	VDD = 4.2V					
		16	35	mA	+85°C		Fosc = 40 MHz				
	All devices	21	40	mA	-40°C		( <b>PRI_RUN</b> , EC oscillator)				
		21	40	mA	+25°C	VDD = 5.0V	,				
		21	40	mA	+85°C						
	All devices	20	40	mA	-40°C						
		20	40	mA	+25°C	VDD = 4.2V					
		20	40	mA	+85°C	1	FOSC = 48 MHz ( <b>PRI_RUN</b> , EC oscillator)				
	All devices	25	50	mA	-40°C						
		25	50	mA	+25°C	VDD = 5.0V					
		25	50	mA	+85°C						

**Legend:** TBD = To Be Determined. Shading of rows is to assist in readability of the table.

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:

- 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: BOR and HLVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.

28.2 DC Characteristics: Power-Down and Supply Current

PIC18F2455/2550/4455/4550 (Industrial)

PIC18LF2455/2550/4455/4550 (Industrial) (Continued)

PIC18LF2455/2550/4455/4550 (Industrial)				rating ( perature	•	ess otherwise star $A \le +85^{\circ}C$ for indus	,			
PIC18F24 (Indus	<b>55/2550/4455/4550</b> Itrial)	Standard Operating Conditions (unless otherwise stated)  Operating temperature -40°C ≤ Ta ≤ +85°C for industrial								
Param No.	Device	Тур	Max	าร						
	Supply Current (IDD) <sup>(2)</sup>									
	PIC18LFX455/X550	65	130	μΑ	-40°C					
		65	120	μΑ	+25°C	VDD = 2.0V				
		70	115	μΑ	+85°C					
	PIC18LFX455/X550	120	270	μΑ	-40°C		Fosc = 1 MHz			
		120	250	μΑ	+25°C	VDD = 3.0V	(PRI_IDLE mode,			
		130	240	μΑ	+85°C		EC oscillator)			
	All devices	230	480	μΑ	-40°C					
		240	450	μΑ	+25°C	VDD = 5.0V				
		250	430	μΑ	+85°C					
	PIC18LFX455/X550	255	475	μΑ	-40°C					
		260	450	μΑ	+25°C	VDD = 2.0V				
		270	430	μΑ	+85°C		Fosc = 4 MHz			
	PIC18LFX455/X550	420	900	μΑ	-40°C					
		430	850	μΑ	+25°C	VDD = 3.0V	(PRI_IDLE mode,			
		450	810	μΑ	+85°C		EC oscillator)			
	All devices	0.9	1.5	mA	-40°C					
		0.9	1.4	mA	+25°C	VDD = 5.0V				
		0.9	1.3	mA	+85°C					
	All devices	6.0	16	mA	-40°C					
		6.2	16	mA	+25°C	VDD = 4.2V				
		6.6	16	mA	+85°C		FOSC = 40 MHz			
	All devices	8.1	18	mA	-40°C		( <b>PRI_IDLE</b> mode, EC oscillator)			
		8.3	18	mA	+25°C	VDD = 5.0V				
		9.0	18	mA	+85°C					
	All devices	8.0	18	mA	-40°C					
		8.1	18	mA	+25°C	VDD = 4.2V				
		8.2	18	mA	+85°C	7	FOSC = 48 MHz (PRI_IDLE mode, EC oscillator)			
	All devices	9.8	21	mA	-40°C					
		10.0	21	mA	+25°C	VDD = 5.0V				
		10.5	21	mA	+85°C	7				

**Legend:** TBD = To Be Determined. Shading of rows is to assist in readability of the table.

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:

- 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:** BOR and HLVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.

#### 28.2 DC Characteristics: Power-Down and Supply Current

PIC18F2455/2550/4455/4550 (Industrial)

PIC18LF2455/2550/4455/4550 (Industrial) (Continued)

	PIC18LF2455/2550/4455/4550 (Industrial)			Standard Operating Conditions (unless otherwise stated)  Operating temperature $-40^{\circ}\text{C} \le \text{Ta} \le +85^{\circ}\text{C}$ for industrial								
PIC18F24 (Indus	<b>55/2550/4455/4550</b> trial)		Standard Operating Conditions (unless otherwise stated)  Operating temperature $-40^{\circ}\text{C} \le \text{TA} \le +85^{\circ}\text{C}$ for industrial									
Param No.	Device	Тур	Max	Units	Conditions							
	Supply Current (IDD) <sup>(2)</sup>											
	PIC18LFX455/X550	14	40	μΑ	-40°C							
		15	40	μΑ	+25°C	VDD = 2.0V	FOSC = 32 kHz <sup>(3)</sup> ( <b>SEC_RUN</b> mode, Timer1 as clock)					
		16	40	μΑ	+85°C							
	PIC18LFX455/X550	40	74	μΑ	-40°C							
		35	70	μΑ	+25°C	VDD = 3.0V						
		31	67	μΑ	+85°C							
	All devices	99	150	μΑ	-40°C							
		81	150	μΑ	+25°C	VDD = 5.0V						
		75	150	μΑ	+85°C							
	PIC18LFX455/X550	2.5	12	μΑ	-40°C							
		3.7	12	μΑ	+25°C	VDD = 2.0V						
		4.5	12	μΑ	+85°C							
	PIC18LFX455/X550	5.0	15	μΑ	-40°C		Fosc = 32 kHz <sup>(3)</sup>					
		5.4	15	μΑ	+25°C	VDD = 3.0V	(SEC_IDLE mode,					
		6.3	15	μΑ	+85°C		Timer1 as clock)					
	All devices	8.5	25	μΑ	-40°C							
		9.0	25	μΑ	+25°C	VDD = 5.0V						
		10.5	36	μΑ	+85°C							

**Legend:** TBD = To Be Determined. Shading of rows is to assist in readability of the table.

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:

- 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: BOR and HLVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.

28.2 DC Characteristics: Power-Down and Supply Current

PIC18F2455/2550/4455/4550 (Industrial)

PIC18LF2455/2550/4455/4550 (Industrial) (Continued)

PIC18I F2	455/2550/4455/4550				Conditions (unles		·		
(Indust			-	perature	•	≤ +85°C for indu			
PIC18F24	<b>55/2550/4455/4550</b> trial)			rating (	Conditions (unles e -40°C ≤ TA	ss otherwise sta ≤ +85°C for indu	•		
Param No.	Device	Тур	Max	Units	Conditions				
D022	Module Differential Currer	its (∆lw	DT, ∆lbo	or, ∆llv	D, $\triangle$ IOSCB, $\triangle$ IAD)				
(∆IWDT)	Watchdog Timer	1.3	3.8	μΑ	-40°C				
		1.4	3.8	μΑ	+25°C	VDD = 2.0V			
		2.0	3.8	μΑ	+85°C				
		1.9	4.6	μΑ	-40°C				
		2.0	4.6	μΑ	+25°C	VDD = 3.0V			
		2.8	4.6	μΑ	+85°C				
		4.0	10	μΑ	-40°C				
		5.5	10	μА	+25°C	VDD = 5.0V			
		5.6	10	μΑ	+85°C				
D022A	Brown-out Reset <sup>(4)</sup>	35	40	μΑ	-40°C to +85°C	VDD = 3.0V			
(∆lbor)		40	45	μΑ	-40°C to +85°C				
		0	2	μΑ	-40°C to +85°C	VDD = 5.0V	Sleep mode, BOREN1:BOREN0 = 10		
D022B	High/Low-Voltage	22	38	μΑ	-40°C to +85°C	VDD = 2.0V			
(∆llvd)	Detect <sup>(4)</sup>	25	40	μΑ	-40°C to +85°C	VDD = 3.0V			
		29	45	μΑ	-40°C to +85°C	VDD = 5.0V			
D025	Timer1 Oscillator	2.1	4.5	μΑ	-40°C				
(∆loscb)		1.8	4.5	μΑ	+25°C	VDD = 2.0V	32 kHz on Timer1 <sup>(3)</sup>		
		2.1	4.5	μΑ	+85°C				
		2.2	6.0	μΑ	-40°C				
		2.6	6.0	μΑ	+25°C	VDD = 3.0V	32 kHz on Timer1 <sup>(3)</sup>		
		2.9	6.0	μΑ	+85°C				
		3.0	8.0	μΑ	-40°C				
		3.2	8.0	μΑ	+25°C	VDD = 5.0V	32 kHz on Timer1 <sup>(3)</sup>		
		3.4	8.0	μΑ	+85°C				
D026	A/D Converter	1.0	2.0	μΑ	-40°C to +85°C	VDD = 2.0V			
(∆IAD)		1.0	2.0	μА	-40°C to +85°C	VDD = 3.0V	A/D on, not converting		
		1.0	2.0	μА	-40°C to +85°C	VDD = 5.0V			

**Legend:** TBD = To Be Determined. Shading of rows is to assist in readability of the table.

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:

- 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:** BOR and HLVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.

#### 28.2 DC Characteristics: Power-Down and Supply Current

PIC18F2455/2550/4455/4550 (Industrial)

PIC18LF2455/2550/4455/4550 (Industrial) (Continued)

							<u> </u>			
PIC18LF2 (Indus	<b>455/2550/4455/4550</b> trial)	Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \le \text{TA} \le +85^{\circ}\text{C}$ for industrial								
PIC18F24 (Indus	<b>55/2550/4455/4550</b> trial)	Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \le \text{Ta} \le +85^{\circ}\text{C}$ for industrial								
Param No.	Device	Тур	Max	Units	Conditions					
	USB and Related Module	Differer	tial Cu	rrents (	ΔIUSBX, ΔIPLL, ΔΙ	UREG)				
ΔIUSBX	USB Module	_	TBD	mA	+25°C	VDD = 3.3V				
	with On-Chip Transceiver	TBD	TBD	mA	+25°C	VDD = 5.0V				
$\Delta$ IPLL	96 MHz PLL	–	TBD	mA	+25°C	VDD = 3.3V				
	(Oscillator Module)	TBD	TBD	TBD	+25°C	VDD = 5.0V				
ΔIUREG	USB Internal Voltage Regulator		TBD	μА	+25°C	VDD = 5.0V				

**Legend:** TBD = To Be Determined. Shading of rows is to assist in readability of the table.

- 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:

- 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:** BOR and HLVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.

# 28.3 DC Characteristics: PIC18F2455/2550/4455/4550 (Industrial) PIC18LF2455/2550/4455/4550 (Industrial)

DC CHA	ARACTE	RISTICS				(unless otherwise stated) ∆ ≤ +85°C for industrial
Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
	VIL	Input Low Voltage				
		I/O ports (except RC4/RC5 in USB mode):				
D030		with TTL buffer	Vss	0.15 VDD	V	VDD < 4.5V
D030A			_	0.8	V	$4.5V \le VDD \le 5.5V$
D031		with Schmitt Trigger buffer RC3 and RC4	Vss Vss	0.2 VDD 0.3 VDD	V V	
D032		MCLR	Vss	0.2 VDD	V	
D032A		OSC1 and T1OSI	Vss	0.3 VDD	V	XT, HS, HSPLL modes <sup>(1)</sup>
D033		OSC1	Vss	0.2 VDD	V	EC mode <sup>(1)</sup>
	VILU	D+/D- input	_	0.8	V	VDD = 4.35V, USB suspended <sup>(5)</sup>
	VIH	Input High Voltage				
		I/O ports (except RC4/RC5 in USB mode):				
D040		with TTL buffer	0.25 VDD + 0.8V	VDD	V	VDD < 4.5V
D040A			2.0	VDD	V	$4.5V \le VDD \le 5.5V$
D041		with Schmitt Trigger buffer RC3 and RC4	0.8 VDD 0.7 VDD	Vdd Vdd	V	
D042		MCLR	0.8 VDD	VDD	V	
D042A		OSC1 and T1OSI	0.7 VDD	VDD	V	XT, HS, HSPLL modes <sup>(1)</sup>
D043		OSC1	0.8 VDD	VDD	V	EC mode <sup>(1)</sup>
	VIHU	D+/D- input	2.4	_	V	VDD = 4.35V, USB suspended <sup>(5)</sup>
	lı∟	Input Leakage Current <sup>(2,3)</sup>				
D060		I/O ports	_	±1	μΑ	Vss ≤ VPIN ≤ VDD, Pin at high-impedance
D061		MCLR	_	±5	μΑ	Vss ≤ VPIN ≤ VDD
D063		OSC1	_	±5	μΑ	Vss ≤ VPIN ≤ VDD
	IPU	Weak Pull-up Current				
D070	IPURB	PORTB weak pull-up current	50	400	μΑ	VDD = 5V, VPIN = VSS

**Note 1:** In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PICmicro® device be driven with an external clock while in RC mode.

- 3: Negative current is defined as current sourced by the pin.
- 4: Parameter is characterized but not tested.
- 5: D+ parameters per USB Specification 2.0.

<sup>2:</sup> The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

# 28.3 DC Characteristics: PIC18F2455/2550/4455/4550 (Industrial) PIC18LF2455/2550/4455/4550 (Industrial) (Continued)

DC CH	ARACTE	RISTICS				(unless otherwise stated) ∆ ≤ +85°C for industrial
Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
	Vol	Output Low Voltage				
D080		I/O ports (except RC4/RC5 in USB mode)	_	0.6	V	IOL = $8.5 \text{ mA}$ , VDD = $4.5 \text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$
D083		OSC2/CLKO (EC, ECIO modes)	_	0.6	V	IOL = 1.6 mA, VDD = 4.5V, -40°C to +85°C
	Volu	D+/D- out	_	0.3		VDD = 4.35V, USB suspended <sup>(5)</sup>
	Vон	Output High Voltage <sup>(3)</sup>				
D090		I/O ports (except RC4/RC5 in USB mode)	VDD - 0.7	_	V	IOH = $-3.0$ mA, VDD = $4.5$ V, $-40$ °C to $+85$ °C
D092		OSC2/CLKO (EC, ECIO, ECPIO modes)	VDD - 0.7	_	V	IOH = $-1.3$ mA, VDD = $4.5$ V, $-40$ °C to $+85$ °C
	Vони	D+/D- out	2.8	3.6	V	VDD = 4.35V, USB suspended <sup>(5)</sup>
		Capacitive Loading Specs on Output Pins				
D100 <sup>(4)</sup>	Cosc2	OSC2 pin	_	15	pF	In XT and HS modes when external clock is used to drive OSC1
D101	Сю	All I/O pins and OSC2 (in RC mode)	_	50	pF	To meet the AC Timing Specifications
D102	Св	SCL, SDA	_	400	pF	I <sup>2</sup> C™ Specification

**Note 1:** In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PICmicro® device be driven with an external clock while in RC mode.

- 2: The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3: Negative current is defined as current sourced by the pin.
- 4: Parameter is characterized but not tested.
- 5: D+ parameters per USB Specification 2.0.

TABLE 28-1: MEMORY PROGRAMMING REQUIREMENTS

DC Cha	racteris	stics					unless otherwise stated) ∆ ≤ +85°C for industrial
Param No.	Sym	Characteristic	Min	Тур†	Max	Units	Conditions
		Internal Program Memory Programming Specifications <sup>(1)</sup>					
D110	VPP	Voltage on MCLR/VPP/RE3 pin	9.00	_	13.25	V	(Note 3)
D113	IDDP	Supply Current during Programming	_	_	10	mA	
		Data EEPROM Memory					
D120	ED	Byte Endurance	100K	1M	_	E/W	-40°C to +85°C
D121	VDRW	VDD for Read/Write	VMIN	_	5.5	V	Using EECON to read/write VMIN = Minimum operating voltage
D122	TDEW	Erase/Write Cycle Time	_	4	_	ms	
D123	TRETD	Characteristic Retention	40	_	_	Year	Provided no other specifications are violated
D124	TREF	Number of Total Erase/Write Cycles before Refresh <sup>(2)</sup>	1M	10M	_	E/W	-40°C to +85°C
		Program Flash Memory					
D130	EP	Cell Endurance	10K	100K	_	E/W	-40°C to +85°C
D131	VPR	VDD for Read	VMIN	_	5.5	V	VMIN = Minimum operating voltage
D132	VIE	VDD for Block Erase	4.5	_	5.5	V	Using ICSP™ port
D132A	Viw	VDD for Externally Timed Erase or Write	4.5	_	5.5	V	Using ICSP port
D132B	VPEW	VDD for Self-Timed Write	VMIN	_	5.5	V	VMIN = Minimum operating voltage
D133	TIE	ICSP Block Erase Cycle Time	_	4	_	ms	VDD > 4.5V
D133A	Tıw	ICSP Erase or Write Cycle Time (externally timed)	1	_	_	ms	VDD > 4.5V
D133A	Tıw	Self-Timed Write Cycle Time	_	2	_	ms	
D134	TRETD	Characteristic Retention	40	100	_	Year	Provided no other specifications are violated

<sup>†</sup> Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** These specifications are for programming the on-chip program memory through the use of table write instructions.

<sup>2:</sup> Refer to **Section 7.7** "**Using the Data EEPROM**" for a more detailed discussion on data EEPROM endurance.

**<sup>3:</sup>** Required only if Single-Supply Programming is disabled.

#### **TABLE 28-2: COMPARATOR SPECIFICATIONS**

**Operating Conditions:** 3.0V < VDD < 5.5V,  $-40^{\circ}C < TA < +85^{\circ}C$  (unless otherwise stated) **Param Characteristics** Min Units Comments Sym Тур Max No. D300 Vioff Input Offset Voltage ±5.0 ±10 mV D301 VICM Input Common Mode Voltage\* 0 VDD - 1.5V D302 CMRR Common Mode Rejection Ratio\* 55 dΒ TRESP Response Time\*(1) 300 150 400 PIC18FXXXX ns PIC18LFXXXX, 300A 150 600 ns VDD = 2.0V301 Тмс2оv Comparator Mode Change to 10 μs Output Valid\*

**Note 1:** Response time measured with one comparator input at (VDD – 1.5)/2, while the other input transitions from Vss to VDD.

#### **TABLE 28-3: VOLTAGE REFERENCE SPECIFICATIONS**

Operating	<b>Operating Conditions:</b> 3.0V < VDD < 5.5V, -40°C < TA < +85°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	_	1/4	1 1/2	LSb LSb	Low Range (CVRR = 1) High Range (CVRR = 0)					
D312	VRur	Unit Resistor Value (R)*	_	2k	_	Ω						
310	TSET	Settling Time*(1)	_	_	10	μs						

<sup>\*</sup> These parameters are characterized but not tested.

Note 1: Settling time measured while CVRR = 1 and CVR3:CVR0 transitions from '0000' to '1111'.

<sup>\*</sup> These parameters are characterized but not tested.

#### TABLE 28-4: USB MODULE SPECIFICATIONS

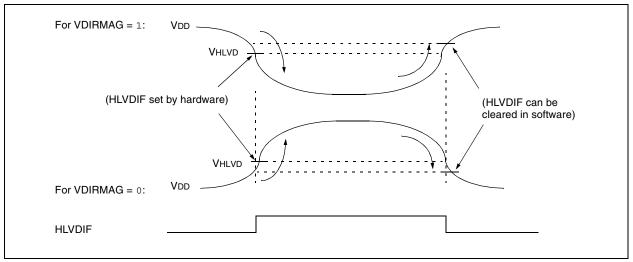
Operatin	g Condit	cions: -40°C < TA < $+85$ °C (unle	ess otherv	vise state	d).		
Param No.	Sym	Characteristic	Min	Тур	Max	Units	Comments
D313	VUSB	USB Voltage	3.0	_	3.6	V	Voltage on bus must be in this range for proper USB operation
D314	lı∟	Input Leakage on Pin	_	_	±1	μА	Vss ≤ VPAD ≤ VDD; pin at high impedance
D315	VILUSB	Input Low Voltage for USB Buffer	_	_	0.8	V	For Vusb range
D316	VIHUSB	Input High Voltage for USB Buffer	2.0	_	_	V	For Vusb range
D317	VCRS	Crossover Voltage	1.3		2.0	V	Voltage range for pad_dp and pad_dm crossover to occur
D318	VDIFS	Differential Input Sensitivity	_	_	0.2	V	The difference between D+ and D- must exceed this value while VCM is met
D319	Vсм	Differential Common Mode Range	0.8	_	2.5	V	
D320	Zout	Driver Output Impedance	28	_	44	Ω	
D321	Vol	Voltage Output Low	0.0		0.3	V	1.5 k $\Omega$ load connected to 3.6V
D322	Vон	Voltage Output High	2.8	_	3.6	V	15 $k\Omega$ load connected to ground

#### TABLE 28-5: USB INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

Operatin	Operating Conditions: -40°C < TA < +85°C (unless otherwise stated).										
Param No.	Sym	Characteristics	Min	Тур	Max	Units	Comments				
D323	VUSBANA	Regulator Output Voltage*	3.0	_	3.6	V					
D324	CUSB	External Filter Capacitor Value*	220	_		nF	Must hold sufficient charge for peak load with minimal voltage drop				

<sup>\*</sup> These parameters are characterized but not tested. Parameter numbers not yet assigned for these specifications.

#### FIGURE 28-3: HIGH/LOW-VOLTAGE DETECT CHARACTERISTICS



#### TABLE 28-6: HIGH/LOW-VOLTAGE DETECT CHARACTERISTICS

# Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for industrial

Param No.	Symbol	Charac	teristic	Min	Тур	Max	Units	Conditions
D420		HLVD Voltage on VDD		2.06	2.17	2.28	٧	
		Transition High-to-Low	HLVDL<3:0> = 0001	2.12	2.23	2.34	٧	
			HLVDL<3:0> = 0010	2.24	2.36	2.48	٧	
			HLVDL<3:0> = 0011	2.32	2.44	2.56	٧	
			HLVDL<3:0> = 0100	2.47	2.60	2.73	٧	
			HLVDL<3:0> = 0101	2.65	2.79	2.93	٧	
			HLVDL<3:0> = 0110	2.74	2.89	3.04	٧	
			HLVDL<3:0> = 0111	2.96	3.12	3.28	٧	
			HLVDL<3:0> = 1000	3.22	3.39	3.56	٧	
			HLVDL<3:0> = 1001	3.37	3.55	3.73	٧	
			HLVDL<3:0> = 1010	3.52	3.71	3.90	٧	
			HLVDL<3:0> = 1011	3.70	3.90	4.10	٧	
			HLVDL<3:0> = 1100	3.90	4.11	4.32	V	
			HLVDL<3:0> = 1101	4.11	4.33	4.55	V	
			HLVDL<3:0> = 1110	4.36	4.59	4.82	V	

#### 28.4 AC (Timing) Characteristics

#### 28.4.1 TIMING PARAMETER SYMBOLOGY

The timing parameter symbols have been created using one of the following formats:

1. Tpps	S2ppS	3. Tcc:st	(I <sup>2</sup> C specifications only)
2. Tpps	S	4. Ts	(I <sup>2</sup> C specifications only)
Т			
F	Frequency	Т	Time
Lowerd	case letters (pp) and their meanings:		
pp			
ad	SPP address write	mc	MCLR
СС	CCP1	osc	OSC1
ck	CLKO	rd	RD
cs	<del>CS</del>	rw	RD or WR
da	SPP data write	sc	SCK
di	SDI	ss	SS
do	SDO	tO	T0CKI
dt	Data in	t1	T13CKI
io	I/O port	wr	WR
Upperd	case letters and their meanings:		
S			
F	Fall	Р	Period
Н	High	R	Rise
- 1	Invalid (High-Impedance)	V	Valid
L	Low	Z	High-Impedance
I <sup>2</sup> C onl	ly		
AA	output access	High	High
BU	JF Bus free	Low	Low
Tcc:st	(I <sup>2</sup> C specifications only)		
CC			
HD	) Hold	SU	Setup
ST			
DA	AT DATA input hold	STO	Stop condition
ST	A Start condition		

#### 28.4.2 TIMING CONDITIONS

The temperature and voltages specified in Table 28-7 apply to all timing specifications unless otherwise noted. Figure 28-4 specifies the load conditions for the timing specifications.

Because of space limitations, the generic terms "PIC18FXXXX" and "PIC18LFXXXX" are used throughout this section to refer to the PIC18F2455/2550/4455/4550 and PIC18LF2455/2550/4455/4550 families of devices specifically and only those devices.

#### TABLE 28-7: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

Standard Operating Conditions (unless otherwise stated)

Note:

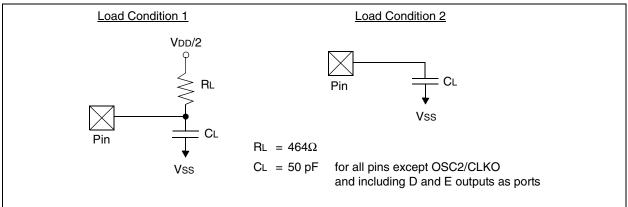
Operating temperature  $-40^{\circ}\text{C} \le \text{TA} \le +85^{\circ}\text{C}$  for industrial

AC CHARACTERISTICS Operating voltage VDD range as described in DC spec Section 28.1 and

Section 28.3.

LF parts operate for industrial temperatures only.

#### FIGURE 28-4: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS



#### 28.4.3 TIMING DIAGRAMS AND SPECIFICATIONS

#### FIGURE 28-5: EXTERNAL CLOCK TIMING (ALL MODES EXCEPT PLL)

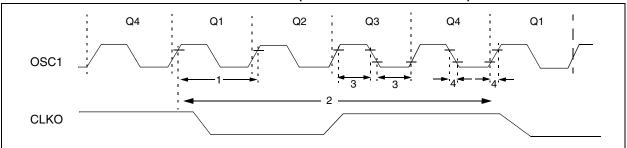


TABLE 28-8: EXTERNAL CLOCK TIMING REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
1A	Fosc	External CLKI Frequency(1)	DC	48	MHz	EC, ECIO Oscillator mode
		Oscillator Frequency <sup>(1)</sup>	0.2	1	MHz	XT, XTPLL Oscillator mode
			4	25	MHz	HS Oscillator mode
			4	25	MHz	HSPLL Oscillator mode
1	Tosc	External CLKI Period <sup>(1)</sup>	20.8	_	ns	EC, ECIO Oscillator mode
		Oscillator Period <sup>(1)</sup>	1000	5000	ns	XT Oscillator mode
			40	250	ns	HS Oscillator mode
			40	250	ns	HSPLL Oscillator mode
2	TCY	Instruction Cycle Time <sup>(1)</sup>	83.3	_	ns	Tcy = 4/Fosc
3	TosL,	External Clock in (OSC1)	30	_	ns	XT Oscillator mode
	TosH	High or Low Time	10	_	ns	HS Oscillator mode
4	TosR,	External Clock in (OSC1)	_	20	ns	XT Oscillator mode
	TosF	Rise or Fall Time	_	7.5	ns	HS Oscillator mode

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.

#### TABLE 28-9: PLL CLOCK TIMING SPECIFICATIONS (VDD = 3.0V TO 5.5V)

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
F10	Fosc	Oscillator Frequency Range	4	_	48	MHz	
F11	Fsys	On-Chip VCO System Frequency	_	96	_	MHz	
F12	t <sub>rc</sub>	PLL Start-up Time (Lock Time)	_	_	2	ms	
F13	$\Delta$ CLK	CLKO Stability (Jitter)	-0.25	_	+0.25	%	

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

TABLE 28-10: AC CHARACTERISTICS: INTERNAL RC ACCURACY
PIC18F2455/2550/4455/4550 (INDUSTRIAL)
PIC18LF2455/2550/4455/4550 (INDUSTRIAL)

	<b>F2455/2550/4455/4550</b> istrial)	Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \le \text{Ta} \le +85^{\circ}\text{C}$ for industrial								
PIC18F2 (Indu		Standard Operating Conditions (unless otherwise stated)  Operating temperature $-40^{\circ}\text{C} \le \text{TA} \le +85^{\circ}\text{C}$ for industrial								
Param No.	Device	Min	Тур	Max	Units	Conc	litions			
	INTOSC Accuracy @ Freq = 8 N	1Hz, 4 M	lz, 4 MHz, 2 MHz, 1 MHz, 500 kHz, 250 kHz, 125 kHz <sup>(1)</sup>							
	PIC18LF2455/2550/4455/4550	-2	+/-1	2	%	+25°C	VDD = 2.7-3.3V			
		-5	_	5	%	-10°C to +85°C	VDD = 2.7-3.3V			
		-10	+/-1	10	%	-40°C to +85°C	VDD = 2.7-3.3V			
	PIC18F2455/2550/4455/4550	-2	+/-1	2	%	+25°C	VDD = 4.5-5.5V			
		<b>-</b> 5	1	5	%	-10°C to +85°C	VDD = 4.5-5.5V			
		-10	+/-1	10	%	-40°C to +85°C	VDD = 4.5-5.5V			
	INTRC Accuracy @ Freq = 31 k	Hz <sup>(2)</sup>				·				
	PIC18LF2455/2550/4455/4550	26.562	-	35.938	kHz	-40°C to +85°C	VDD = 2.7-3.3V			
	PIC18F2455/2550/4455/4550	26.562		35.938	kHz	-40°C to +85°C	VDD = 4.5-5.5V			

**Legend:** Shading of rows is to assist in readability of the table.

Note 1: Frequency calibrated at 25°C. OSCTUNE register can be used to compensate for temperature drift.

- 2: INTRC frequency after calibration.
- 3: Change of INTRC frequency as VDD changes.

FIGURE 28-6: CLKO AND I/O TIMING

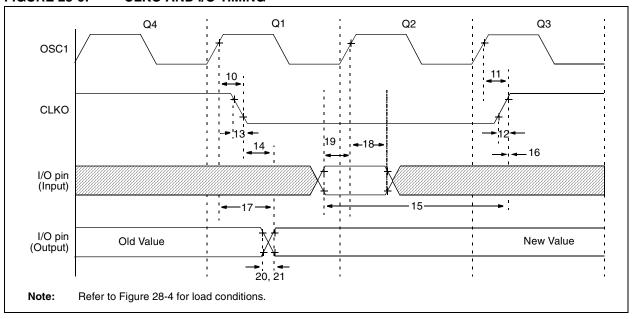


TABLE 28-11: CLKO AND I/O TIMING REQUIREMENTS

Param No.	Symbol	Characteri	stic	Min	Тур	Max	Units	Conditions
10	TosH2ckL	OSC1 ↑ to CLKO ↓		_	75	200	ns	(Note 1)
11	TosH2ckH	OSC1 ↑ to CLKO ↑		_	75	200	ns	(Note 1)
12	TckR	CLKO Rise Time		_	35	100	ns	(Note 1)
13	TckF	CLKO Fall Time		_	35	100	ns	(Note 1)
14	TckL2ioV	CLKO ↓ to Port Out Valid		_	_	0.5 Tcy + 20	ns	(Note 1)
15	TioV2ckH	Port In Valid before CLKC	) ↑	0.25 Tcy + 25	_	_	ns	(Note 1)
16	TckH2iol	Port In Hold after CLKO		0	_	_	ns	(Note 1)
17	TosH2ioV	OSC1 ↑ (Q1 cycle) to Po	rt Out Valid	_	50	150	ns	
18	TosH2ioI	OSC1 ↑ (Q2 cycle) to	PIC18FXXXX	100	_	_	ns	
18A		Port Input Invalid (I/O in hold time)	PIC18 <b>LF</b> XXXX	200	_	_	ns	VDD = 2.0V
19	TioV2osH	Port Input Valid to OSC1 1	(I/O in setup time)	0	_	_	ns	
20	TioR	Port Output Rise Time	PIC18FXXXX	_	10	25	ns	
20A			PIC18 <b>LF</b> XXXX	_	_	60	ns	VDD = 2.0V
21	TioF	Port Output Fall Time	PIC18FXXXX	_	10	25	ns	
21A			PIC18 <b>LF</b> XXXX	_	_	60	ns	VDD = 2.0V
22†	TINP	INT pin High or Low Time	High or Low Time		_	_	ns	
23†	TRBP	RB7:RB4 Change INT Hi	gh or Low Time	Tcy	_	_	ns	

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

Note 1: Measurements are taken in RC mode, where CLKO output is 4 x Tosc.

FIGURE 28-7: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

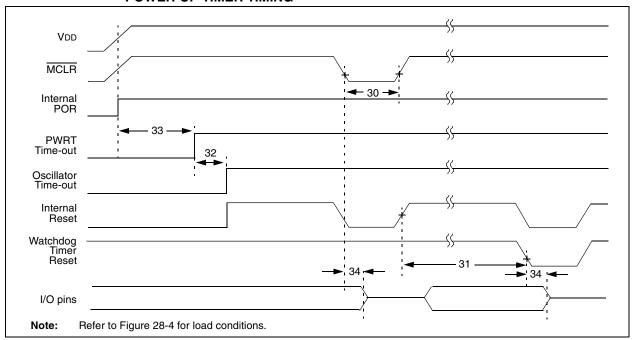


FIGURE 28-8: BROWN-OUT RESET TIMING

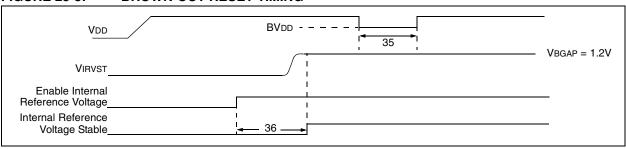


TABLE 28-12: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Тур	Max	Units	Conditions
30	TmcL	MCLR Pulse Width (low)	2			μs	
31	TWDT	Watchdog Timer Time-out Period (no postscaler)	_	4.00	TBD	ms	
32	Tost	Oscillator Start-up Timer Period	1024 Tosc	_	1024 Tosc	_	Tosc = OSC1 period
33	TPWRT	Power-up Timer Period	_	65.5	TBD	ms	
34	Tioz	I/O High-Impedance from MCLR Low or Watchdog Timer Reset	_	2	_	μs	
35	TBOR	Brown-out Reset Pulse Width	200	_	_	μs	V <sub>DD</sub> ≤ B <sub>VDD</sub> (see D005)
36	TIRVST	Time for Internal Reference Voltage to become Stable	_	20	50	μs	
37	TLVD	Low-Voltage Detect Pulse Width	200		_	μs	$VDD \le VLVD$
38	TCSD	CPU Start-up Time	5	_	10	μs	
39	TIOBST	Time for INTOSC to Stabilize	_	1	_	ms	

**Legend:** TBD = To Be Determined

FIGURE 28-9: TIMERO AND TIMER1 EXTERNAL CLOCK TIMINGS

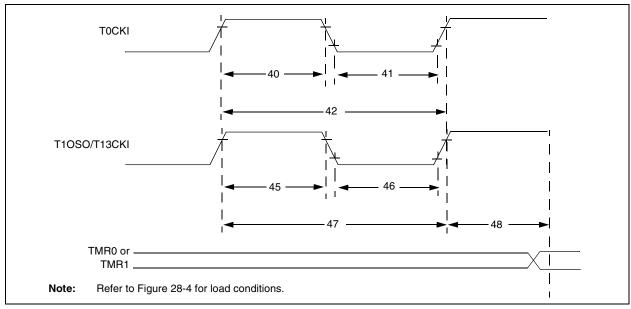


TABLE 28-13: TIMERO AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Symbol		Characteristic		Min	Max	Units	Conditions
40	Tt0H	T0CKI High Pu	lse Width	No prescaler	0.5 Tcy + 20	_	ns	
				With prescaler	10	_	ns	
41	Tt0L	T0CKI Low Pu	lse Width	No prescaler	0.5 Tcy + 20	_	ns	
				With prescaler	10	_	ns	
42	Tt0P	T0CKI Period		No prescaler	Tcy + 10		ns	
				With prescaler	Greater of: 20 ns or (Tcy + 40)/N		ns	N = prescale value (1, 2, 4,, 256)
45	Tt1H	T13CKI High	Synchronous, no	prescaler	0.5 Tcy + 20	_	ns	
		Time	Synchronous,	PIC18FXXXX	10	_	ns	
			with prescaler	PIC18 <b>LF</b> XXXX	25	_	ns	VDD = 2.0V
			Asynchronous	PIC18FXXXX	30		ns	
				PIC18 <b>LF</b> XXXX	50		ns	VDD = 2.0V
46	Tt1L	T13CKI Low	Synchronous, no	prescaler	0.5 Tcy + 5		ns	
		Time	Synchronous,	PIC18FXXXX	10		ns	
			with prescaler	PIC18 <b>LF</b> XXXX	25		ns	VDD = 2.0V
			Asynchronous	PIC18FXXXX	30		ns	
				PIC18 <b>LF</b> XXXX	50		ns	VDD = 2.0V
47	Tt1P	T13CKI Input Period	Synchronous		Greater of: 20 ns or (TCY + 40)/N		ns	N = prescale value (1, 2, 4, 8)
			Asynchronous		60		ns	
	Ft1	T13CKI Oscilla	tor Input Frequency Range		DC	50	kHz	
48	Tcke2tmrl	Delay from Ext Increment	ernal T13CKI Cloc	k Edge to Timer	2 Tosc	7 Tosc	_	

FIGURE 28-10: CAPTURE/COMPARE/PWM TIMINGS (ALL CCP MODULES)

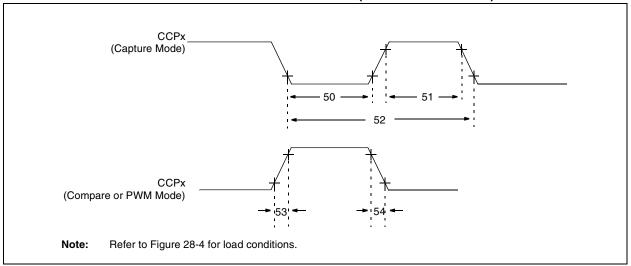


TABLE 28-14: CAPTURE/COMPARE/PWM REQUIREMENTS (ALL CCP MODULES)

Param No.	Symbol	С	haracteristi	С	Min	Max	Units	Conditions
50	TccL	CCPx Input Low	No prescale	er	0.5 Tcy + 20	_	ns	
		Time	With	PIC18FXXXX	10	_	ns	
			prescaler	PIC18 <b>LF</b> XXXX	20	_	ns	VDD = 2.0V
51	TccH			er	0.5 Tcy + 20	_	ns	
		High Time	With	PIC18FXXXX	10	_	ns	
			prescaler	PIC18 <b>LF</b> XXXX	20	_	ns	VDD = 2.0V
52	TccP	CCPx Input Perio	od		<u>3 Tcy + 40</u> N	_	ns	N = prescale value (1, 4 or 16)
53	TccR	CCPx Output Fal	II Time	PIC18FXXXX	_	25	ns	
				PIC18 <b>LF</b> XXXX	_	45	ns	VDD = 2.0V
54	TccF	CCPx Output Fal	CPx Output Fall Time PIC1			25	ns	
				PIC18 <b>LF</b> XXXX	_	45	ns	VDD = 2.0V

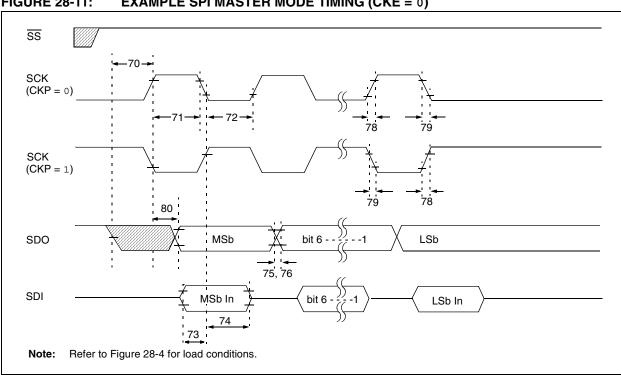


FIGURE 28-11: **EXAMPLE SPI MASTER MODE TIMING (CKE = 0)** 

TABLE 28-15: EXAMPLE SPI MODE REQUIREMENTS (MASTER MODE, CKE = 0)

Param No.	Symbol	Characteristi	Characteristic		Max	Units	Conditions
70	TssL2scH, TssL2scL	SS ↓ to SCK ↓ or SCK ↑ Input		Tcy	1	ns	
71	TscH	SCK Input High Time	Continuous	1.25 Tcy + 30	_	ns	
71A		(Slave mode)	Single Byte	40	_	ns	(Note 1)
72	TscL	SCK Input Low Time	Continuous	1.25 Tcy + 30	_	ns	
72A		(Slave mode)	Single Byte	40	_	ns	(Note 1)
73	TdiV2scH, TdiV2scL	Setup Time of SDI Data Input to SCK Edge		100	_	ns	
73A	Tb2b	Last Clock Edge of Byte 1 to the 1st Clock Edge of Byte 2		1.5 Tcy + 40	_	ns	(Note 2)
74	TscH2diL, TscL2diL	Hold Time of SDI Data Input to	SCK Edge	100	_	ns	
75	TdoR	SDO Data Output Rise Time	PIC18FXXXX	_	25	ns	
			PIC18 <b>LF</b> XXXX	_	45	ns	VDD = 2.0V
76	TdoF	SDO Data Output Fall Time		_	25	ns	
78	TscR	SCK Output Rise Time	PIC18FXXXX	_	25	ns	
		(Master mode)	PIC18 <b>LF</b> XXXX	_	45	ns	VDD = 2.0V
79	TscF	SCK Output Fall Time (Master mode)			25	ns	
80	TscH2doV,	SDO Data Output Valid after	PIC18FXXXX		50	ns	
	TscL2doV	SCK Edge	PIC18 <b>LF</b> XXXX	_	100	ns	VDD = 2.0V

Note 1: Requires the use of Parameter 73A.

2: Only if Parameter 71A and 72A are used.

FIGURE 28-12: EXAMPLE SPI MASTER MODE TIMING (CKE = 1)

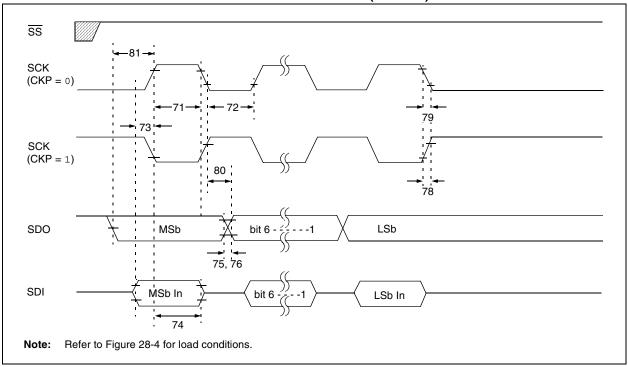


TABLE 28-16: EXAMPLE SPI MODE REQUIREMENTS (MASTER MODE, CKE = 1)

Param. No.	Symbol	Characterist	ic	Min	Max	Units	Conditions
71	TscH	SCK Input High Time	Continuous	1.25 Tcy + 30	_	ns	
71A		(Slave mode)	Single Byte	40	_	ns	(Note 1)
72	TscL	SCK Input Low Time	Continuous	1.25 Tcy + 30	_	ns	
72A		(Slave mode)	Single Byte	40	_	ns	(Note 1)
73	TdiV2scH, TdiV2scL	Setup Time of SDI Data Input	up Time of SDI Data Input to SCK Edge		_	ns	
73A	Tb2b	Last Clock Edge of Byte 1 to to f Byte 2	1.5 TCY + 40	_	ns	(Note 2)	
74	TscH2diL, TscL2diL	Hold Time of SDI Data Input to SCK Edge		100	_	ns	
75	TdoR	SDO Data Output Rise Time	PIC18FXXXX	_	25	ns	
			PIC18 <b>LF</b> XXXX	_	45	ns	VDD = 2.0V
76	TdoF	SDO Data Output Fall Time		_	25	ns	
78	TscR	SCK Output Rise Time	PIC18FXXXX	_	25	ns	
		(Master mode)	PIC18 <b>LF</b> XXXX	_	45	ns	VDD = 2.0V
79	TscF	SCK Output Fall Time (Maste	r mode)	_	25	ns	
80	TscH2doV,	SDO Data Output Valid after	PIC18FXXXX	_	50	ns	
	TscL2doV	SCK Edge	PIC18 <b>LF</b> XXXX		100	ns	VDD = 2.0V
81	TdoV2scH, TdoV2scL	SDO Data Output Setup to SO	CK Edge	Tcy	_	ns	

Note 1: Requires the use of Parameter 73A.

2: Only if Parameter 71A and 72A are used.

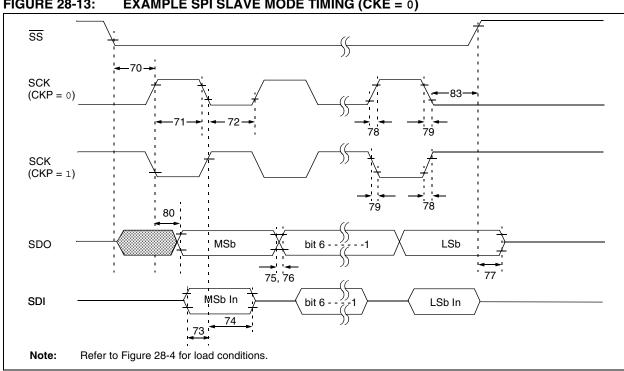


FIGURE 28-13: **EXAMPLE SPI SLAVE MODE TIMING (CKE = 0)** 

TABLE 28-17: EXAMPLE SPI MODE REQUIREMENTS (SLAVE MODE TIMING, CKE = 0)

Param No.	Symbol	Characteristic	Characteristic		Max	Units	Conditions
70	TssL2scH, TssL2scL	SS ↓ to SCK ↓ or SCK ↑ Input	SCK ↑ Input		ı	ns	
71	TscH	SCK Input High Time	Continuous	1.25 Tcy + 30	1	ns	
71A		(Slave mode)	Single Byte	40	1	ns	(Note 1)
72	TscL	SCK Input Low Time	Continuous	1.25 Tcy + 30		ns	
72A		(Slave mode)	Single Byte	40	_	ns	(Note 1)
73	TdiV2scH, TdiV2scL	Setup Time of SDI Data Input to SCK Ed	Time of SDI Data Input to SCK Edge		1	ns	
73A	Tb2b	Last Clock Edge of Byte 1 to the First Cloc	1.5 Tcy + 40	_	ns	(Note 2)	
74	TscH2diL, TscL2diL	Hold Time of SDI Data Input to SCK Edge		100	_	ns	
75	TdoR	SDO Data Output Rise Time	PIC18FXXXX	_	25	ns	
			PIC18 <b>LF</b> XXXX	_	45	ns	VDD = 2.0V
76	TdoF	SDO Data Output Fall Time			25	ns	
77	TssH2doZ	SS ↑ to SDO Output High-Impedance		10	50	ns	
78	TscR	SCK Output Rise Time (Master mode)	PIC18FXXXX	_	25	ns	
			PIC18 <b>LF</b> XXXX	_	45	ns	VDD = 2.0V
79	TscF	SCK Output Fall Time (Master mode)		_	25	ns	
80	TscH2doV,	SDO Data Output Valid after SCK Edge	PIC18FXXXX	_	50	ns	
	TscL2doV		PIC18 <b>LF</b> XXXX		100	ns	VDD = 2.0V
83	TscH2ssH, TscL2ssH	SS ↑ after SCK edge		1.5 Tcy + 40	_	ns	

Requires the use of Parameter 73A. Note 1:

2: Only if Parameter 71A and 72A are used.

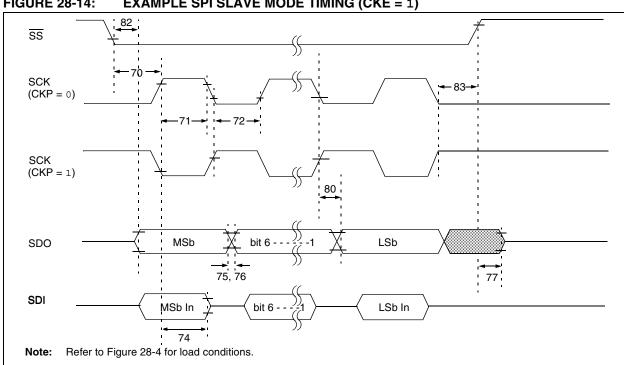


FIGURE 28-14: **EXAMPLE SPI SLAVE MODE TIMING (CKE = 1)** 

TABLE 28-18: EXAMPLE SPI SLAVE MODE REQUIREMENTS (CKE = 1)

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
70	TssL2scH, TssL2scL	$\overline{\text{SS}}\downarrow$ to SCK $\downarrow$ or SCK $\uparrow$ Input		Tcy	_	ns	
71	TscH	SCK Input High Time	Continuous	1.25 Tcy + 30	_	ns	
71A		(Slave mode)	Single Byte	40	_	ns	(Note 1)
72	TscL	SCK Input Low Time	Continuous	1.25 Tcy + 30	_	ns	
72A		(Slave mode)	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 SDI Data Input to SCK	d Time of SDI Data Input to SCK Edge			ns	
75	TdoR	SDO Data Output Rise Time	PIC18FXXXX	_	25	ns	
			PIC18 <b>LF</b> XXXX	_	45	ns	VDD = 2.0V
76	TdoF	SDO Data Output Fall Time		_	25	ns	
77	TssH2doZ	SS ↑ to SDO Output High-Impedance	е	10	50	ns	
78	TscR	SCK Output Rise Time	PIC18FXXXX	_	25	ns	
		(Master mode)	PIC18 <b>LF</b> XXXX	_	45	ns	VDD = 2.0V
79	TscF	SCK Output Fall Time (Master mode	e)	_	25	ns	
80	TscH2doV,	SDO Data Output Valid after SCK	PIC18FXXXX	_	50	ns	
	TscL2doV	Edge	PIC18 <b>LF</b> XXXX	_	100	ns	VDD = 2.0V
82	TssL2doV	SDO Data Output Valid after SS ↓	PIC18FXXXX	_	50	ns	
		Edge	PIC18 <b>LF</b> XXXX	_	100	ns	VDD = 2.0V
83	TscH2ssH, TscL2ssH	SS ↑ after SCK Edge		1.5 Tcy + 40	_	ns	

Note 1: Requires the use of Parameter 73A.

Only if Parameter 71A and 72A are used.

FIGURE 28-15: I<sup>2</sup>CTM BUS START/STOP BITS TIMING

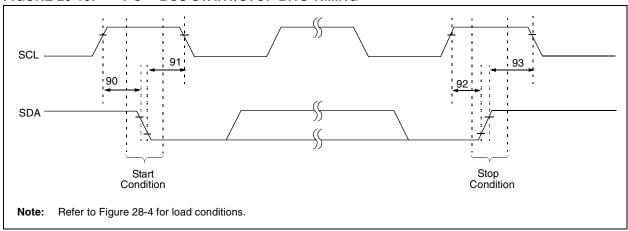
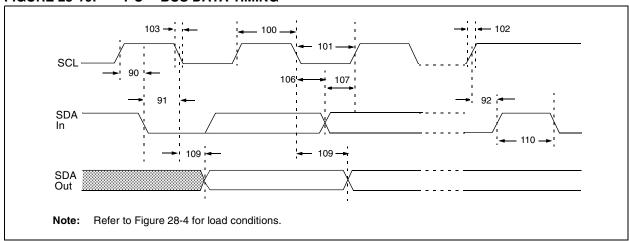


TABLE 28-19: I<sup>2</sup>C™ BUS START/STOP BITS REQUIREMENTS (SLAVE MODE)

Param. No.	Symbol	Characte	ristic	Min	Max	Units	Conditions
90	Tsu:sta	Start Condition	100 kHz mode	4700	_	ns	Only relevant for Repeated
		Setup Time	400 kHz mode	600	_		Start condition
91	THD:STA	Start Condition	100 kHz mode	4000	_	ns	After this period, the first
		Hold Time	400 kHz mode	600	_		clock pulse is generated
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 28-16: I<sup>2</sup>C™ BUS DATA TIMING



#### TABLE 28-20: I<sup>2</sup>C™ BUS DATA REQUIREMENTS (SLAVE MODE)

Param. No.	Symbol	Characte	eristic	Min	Max	Units	Conditions
100	THIGH	Clock High Time	100 kHz mode	4.0	_	μs	PIC18FXXXX must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	_	μs	PIC18FXXXX must operate at a minimum of 10 MHz
			MSSP Module	1.5 Tcy	_		
101	TLOW	Clock Low Time	100 kHz mode	4.7		μs	PIC18FXXXX must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3		μs	PIC18FXXXX must operate at a minimum of 10 MHz
			MSSP Module	1.5 Tcy	_		
102	TR	SDA and SCL Rise	100 kHz mode	_	1000	ns	
		Time	400 kHz mode	20 + 0.1 CB	300	ns	CB is specified to be from 10 to 400 pF
103	TF	SDA and SCL Fall	100 kHz mode	_	300	ns	
		Time	400 kHz mode	20 + 0.1 CB	300	ns	CB is specified to be from 10 to 400 pF
90	Tsu:sta	Start Condition	100 kHz mode	4.7	_	μs	Only relevant for Repeated
		Setup Time	400 kHz mode	0.6	_	μs	Start condition
91	THD:STA	Start Condition	100 kHz mode	4.0	_	μs	After this period, the first
		Hold Time	400 kHz mode	0.6	_	μs	clock pulse is generated
106	THD:DAT	Data Input Hold	100 kHz mode	0	_	ns	
		Time	400 kHz mode	0	0.9	μs	
107	TSU:DAT	Data Input Setup	100 kHz mode	250	_	ns	(Note 2)
		Time	400 kHz mode	100	_	ns	
92	Tsu:sto	Stop Condition	100 kHz mode	4.7	_	μs	
		Setup Time	400 kHz mode	0.6	_	μs	
109	TAA	Output Valid from	100 kHz mode	_	3500	ns	(Note 1)
		Clock	400 kHz mode	_	_	ns	
110	TBUF	Bus Free Time	100 kHz mode	4.7	_	μs	Time the bus must be free
			400 kHz mode	1.3	_	μs	before a new transmission can start
D102	Св	Bus Capacitive Load		<u> </u>	400	pF	

- **Note 1:** As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.
  - 2: A Fast mode 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 SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line, TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

FIGURE 28-17: MASTER SSP I<sup>2</sup>C™ BUS START/STOP BITS TIMING WAVEFORMS

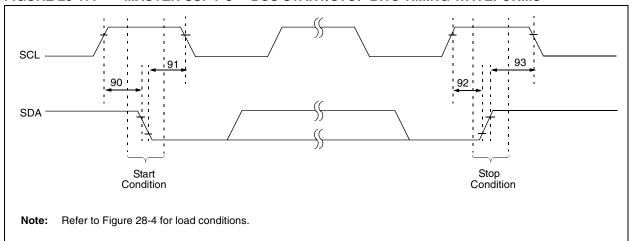


TABLE 28-21: MASTER SSP I<sup>2</sup>C™ BUS START/STOP BITS REQUIREMENTS

Param. No.	Symbol	Characte	eristic	Min	Max	Units	Conditions
90	Tsu:sta	Start Condition	100 kHz mode	2(Tosc)(BRG + 1)	_	ns	Only relevant for
		Setup Time	400 kHz mode	2(Tosc)(BRG + 1)	_		Repeated Start
			1 MHz mode <sup>(1)</sup>	2(Tosc)(BRG + 1)	_		condition
91	THD:STA	Start Condition	100 kHz mode	2(Tosc)(BRG + 1)	_	ns	After this period, the
		Hold Time	400 kHz mode	2(Tosc)(BRG + 1)	_		first clock pulse is
			1 MHz mode <sup>(1)</sup>	2(Tosc)(BRG + 1)	_		generated
92	Tsu:sto	Stop Condition	100 kHz mode	2(Tosc)(BRG + 1)	_	ns	
		Setup Time	400 kHz mode	2(Tosc)(BRG + 1)	_		
			1 MHz mode <sup>(1)</sup>	2(Tosc)(BRG + 1)	_		
93	THD:STO	Stop Condition	100 kHz mode	2(Tosc)(BRG + 1)	_	ns	
		Hold Time	400 kHz mode	2(Tosc)(BRG + 1)	_		
			1 MHz mode <sup>(1)</sup>	2(Tosc)(BRG + 1)	_		

**Note 1:** Maximum pin capacitance = 10 pF for all  $I^2C^{TM}$  pins.

FIGURE 28-18: MASTER SSP I<sup>2</sup>C™ BUS DATA TIMING

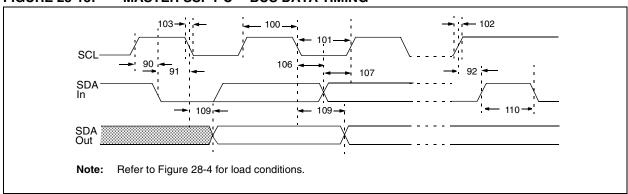


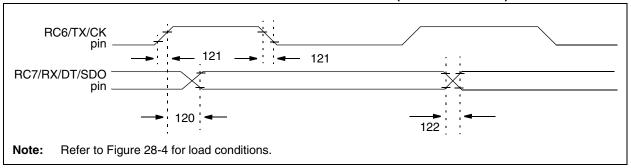
TABLE 28-22: MASTER SSP I<sup>2</sup>C™ BUS DATA REQUIREMENTS

Param. No.	Symbol	Charac	teristic	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 <sup>(1)</sup>	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 <sup>(1)</sup>	2(Tosc)(BRG + 1)	_	ms		
102	TR	SDA and SCL	100 kHz mode	_	1000	ns	CB is specified to be from	
		Rise Time	400 kHz mode	20 + 0.1 CB	300	ns	10 to 400 pF	
			1 MHz mode <sup>(1)</sup>	_	300	ns		
103	TF	SDA and SCL	100 kHz mode	_	300	ns	CB is specified to be from	
		Fall Time	400 kHz mode	20 + 0.1 CB	300	ns	10 to 400 pF	
			1 MHz mode <sup>(1)</sup>	_	100	ns		
90	Tsu:sta	Start Condition	100 kHz mode	2(Tosc)(BRG + 1)	_	ms	Only relevant for	
		Setup Time	400 kHz mode	2(Tosc)(BRG + 1)	_	ms	Repeated Start	
			1 MHz mode <sup>(1)</sup>	2(Tosc)(BRG + 1)	_	ms	condition	
91	THD:STA	Start Condition	100 kHz mode	2(Tosc)(BRG + 1)	_	ms	After this period, the first	
		Hold Time	400 kHz mode	2(Tosc)(BRG + 1)	_	ms	clock pulse is generated	
			1 MHz mode <sup>(1)</sup>	2(Tosc)(BRG + 1)	_	ms		
106	THD:DAT	Data Input	100 kHz mode	0	_	ns		
		Hold Time	400 kHz mode	0	0.9	ms		
107	TSU:DAT	Data Input	100 kHz mode	250	_	ns	(Note 2)	
		Setup Time	400 kHz mode	100	_	ns		
92	Tsu:sto	Stop Condition	100 kHz mode	2(Tosc)(BRG + 1)	_	ms		
		Setup Time	400 kHz mode	2(Tosc)(BRG + 1)	_	ms		
			1 MHz mode <sup>(1)</sup>	2(Tosc)(BRG + 1)	_	ms		
109	TAA	Output Valid	100 kHz mode	_	3500	ns		
		from Clock	400 kHz mode	_	1000	ns		
			1 MHz mode <sup>(1)</sup>	_	_	ns		
110	TBUF	Bus Free Time	100 kHz mode	4.7	_	ms	Time the bus must be free	
			400 kHz mode	1.3	_	ms	before a new transmission can start	
D102	Св	Bus Capacitive Lo	oading	_	400	pF		

**Note 1:** Maximum pin capacitance = 10 pF for all  $I^2C^{TM}$  pins.

2: A Fast mode I<sup>2</sup>C bus device can be used in a Standard mode I<sup>2</sup>C bus system but parameter #107 ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line, parameter #102 + parameter #107 = 1000 + 250 = 1250 ns (for 100 kHz mode), before the SCL line is released.

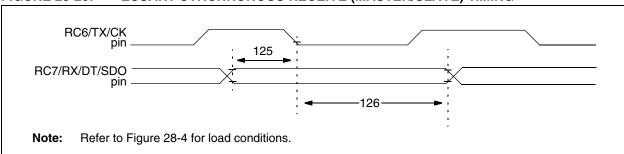
#### FIGURE 28-19: EUSART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING



#### TABLE 28-23: EUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock High to Data Out Valid	PIC18 <b>F</b> XXXX	_	40	ns	
			PIC18 <b>LF</b> XXXX	_	100	ns	VDD = 2.0V
121	Tckrf	Clock Out Rise Time and Fall Time	PIC18FXXXX	_	20	ns	
		(Master mode)	PIC18 <b>LF</b> XXXX	_	50	ns	VDD = 2.0V
122	Tdtrf	Data Out Rise Time and Fall Time	PIC18FXXXX	_	20	ns	
			PIC18 <b>LF</b> XXXX	_	50	ns	VDD = 2.0V

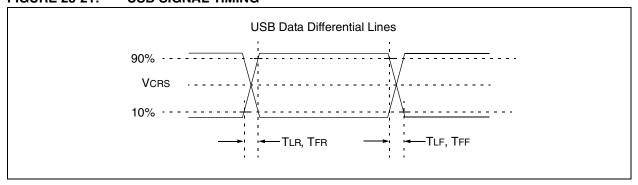
#### FIGURE 28-20: EUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



#### TABLE 28-24: EUSART SYNCHRONOUS RECEIVE REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
125	TDTV2CKL	SYNC RCV (MASTER & SLAVE) Data Hold before CK ↓ (DT hold time)	10		ns	
126	TCKL2DTL	Data Hold after CK ↓ (DT hold time)	15		ns	

### FIGURE 28-21: USB SIGNAL TIMING



### **TABLE 28-25: USB LOW-SPEED TIMING REQUIREMENTS**

Param No.	Symbol	Characteristic	Min	Тур	Max	Units	Conditions
	TLR	Transition Rise Time	75	_	300	ns	CL = 200 to 600 pF
	TLF	Transition Fall Time	75		300	ns	CL = 200 to 600 pF
	TLRFM	Rise/Fall Time Matching	80		125	%	

### TABLE 28-26: USB FULL-SPEED REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Тур	Max	Units	Conditions
	TFR	Transition Rise Time	4	_	20	ns	CL = 50 pF
	TFF	Transition Fall Time	4	_	20	ns	CL = 50 pF
	TFRFM	Rise/Fall Time Matching	90	_	111.1	%	

© 2006 Microchip Technology Inc. Preliminary DS39632C-page 395

FIGURE 28-22: STREAMING PARALLEL PORT TIMING (PIC18F4455/4550)

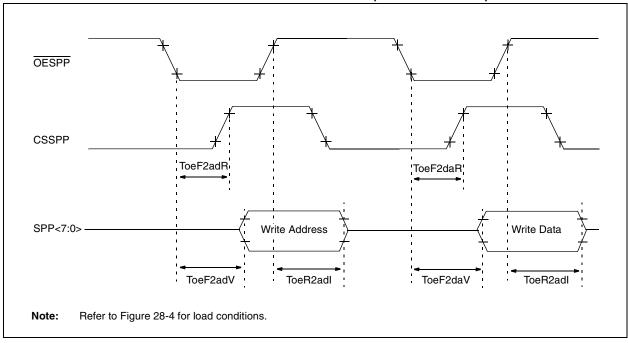


TABLE 28-27: STREAMING PARALLEL PORT REQUIREMENTS (PIC18F4455/4550)

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
	ToeF2adR	OESPP Falling Edge to CSSPP Rising Edge, Address Out	0	5	ns	
	ToeF2adV	OESPP Falling Edge to Address Out Valid	0	5	ns	
	ToeR2adl	OESPP Rising Edge to Address Out Invalid	0	5	ns	
	ToeF2daR	OESPP Falling Edge to CSSPP Rising Edge, Data Out	0	5	ns	
	ToeF2daV	OESPP Falling Edge to Address Out Valid	0	5	ns	
	ToeR2dal	OESPP Rising Edge to Data Out Invalid	0	5	ns	

TABLE 28-28: A/D CONVERTER CHARACTERISTICS: PIC18F2455/2550/4455/4550 (INDUSTRIAL) PIC18LF2455/2550/4455/4550 (INDUSTRIAL)

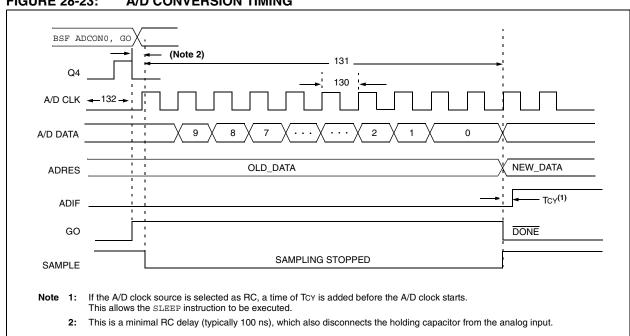
Param No.	Symbol	Characteristic	Min	Тур	Max	Units	Conditions
A01	NR	Resolution	_		10	bit	$\Delta V$ REF $\geq 3.0V$
A03	EIL	Integral Linearity Error	_	_	<±1	LSb	$\Delta V$ REF $\geq 3.0V$
A04	EDL	Differential Linearity Error	_	_	<±1	LSb	ΔVREF ≥ 3.0V
A06	Eoff	Offset Error	_	_	<±1.5	LSb	$\Delta V$ REF $\geq 3.0V$
A07	Egn	Gain Error	_	_	<±1	LSb	$\Delta V$ REF $\geq 3.0V$
A10	_	Monotonicity	Guaranteed <sup>(1)</sup>			_	VSS ≤ VAIN ≤ VREF
A20	$\Delta V$ REF	Reference Voltage Range (VREFH – VREFL)	1.8 3		_	V	VDD < 3.0V VDD ≥ 3.0V
A21	VREFH	Reference Voltage High	Vss		VREFH	V	
A22	VREFL	Reference Voltage Low	Vss - 0.3V		VDD - 3.0V	V	
A25	VAIN	Analog Input Voltage	VREFL	_	VREFH	V	
A30	ZAIN	Recommended Impedance of Analog Voltage Source	_		2.5	kΩ	
A50	IREF	VREF Input Current <sup>(2)</sup>		_ _	5 150	μ <b>Α</b> μ <b>Α</b>	During VAIN 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: VREFH current is from RA3/AN3/VREF+ pin or VDD, whichever is selected as the VREFH source.

VREFL current is from RA2/AN2/VREF-/CVREF pin or VSS, whichever is selected as the VREFL source.

#### FIGURE 28-23: A/D CONVERSION TIMING



© 2006 Microchip Technology Inc. Preliminary DS39632C-page 397

#### **TABLE 28-29: A/D CONVERSION REQUIREMENTS**

Param No.	Symbol	Characte	eristic	Min	Max	Units	Conditions
130	TAD	A/D Clock Period	PIC18FXXXX	0.7	25.0 <sup>(1)</sup>	μs	Tosc based, VREF ≥ 3.0V
			PIC18 <b>LF</b> XXXX	1.4	25.0 <sup>(1)</sup>	μs	VDD = 2.0V, TOSC based, VREF full range
			PIC18FXXXX	TBD	1	μs	A/D RC mode
			PIC18 <b>LF</b> XXXX	TBD	3	μs	V <sub>DD</sub> = 2.0V, A/D RC mode
131	TCNV	Conversion Time (not including acquisition	Conversion Time (not including acquisition time) <sup>(2)</sup>			TAD	
132	TACQ	Acquisition Time <sup>(3)</sup>		1.4	_	μs	-40°C to +85°C
				TBD	_	μs	$0^{\circ}C \le to \le +85^{\circ}C$
135	Tswc	Switching Time from C	onvert $\rightarrow$ Sample		(Note 4)		
137	TDIS	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 Tcy 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.

# 29.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

Graphs and tables are not available at this time.

NOTES:

### 30.0 PACKAGING INFORMATION

### 30.1 Package Marking Information

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



### Example



#### 28-Lead SOIC



### Example



#### 40-Lead PDIP



### 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)

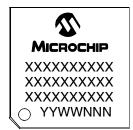
This package is Pb-free. The Pb-free JEDEC designator (e3)

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.

### **Package Marking Information (Continued)**

44-Lead TQFP



44-Lead QFN



Example



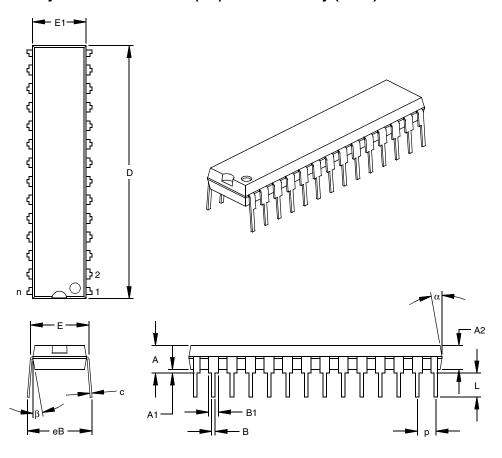
Example



#### **Package Details** 30.2

The following sections give the technical details of the packages.

### 28-Lead Skinny Plastic Dual In-line (SP) – 300 mil Body (PDIP)



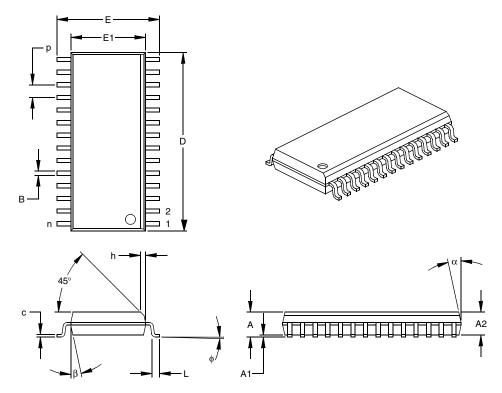
		Units		INCHES*		MILLIMETERS			
Dimen	sion L	imits	MIN	NOM	MAX	MIN	NOM	MAX	
Number of Pins		n		28			28		
Pitch		р		.100			2.54		
Top to Seating Plane		Α	.140	.150	.160	3.56	3.81	4.06	
Molded Package Thickness		A2	.125	.130	.135	3.18	3.30	3.43	
Base to Seating Plane		A1	.015			0.38			
Shoulder to Shoulder Width		E	.300	.310	.325	7.62	7.87	8.26	
Molded Package Width		E1	.275	.285	.295	6.99	7.24	7.49	
Overall Length		D	1.345	1.365	1.385	34.16	34.67	35.18	
Tip to Seating Plane		L	.125	.130	.135	3.18	3.30	3.43	
Lead Thickness		С	.008	.012	.015	0.20	0.29	0.38	
Upper Lead Width		B1	.040	.053	.065	1.02	1.33	1.65	
Lower Lead Width		В	.016	.019	.022	0.41	0.48	0.56	
Overall Row Spacing	§	eВ	.320	.350	.430	8.13	8.89	10.92	
Mold Draft Angle Top		α	5	10	15	5	10	15	
Mold Draft Angle Bottom		β	5	10	15	5	10	15	
* Controlling Parameter § Significant Characteristic Notes: Dimension D and E1 do not include	lo mo	ld floob (	or protrucione	Mold floob or r	vratruojana aha	Il not avoord (	010" (0.254mm	) por sido	

Dimension D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MO-095

Drawing No. C04-070

### 28-Lead Plastic Small Outline (SO) - Wide, 300 mil Body (SOIC)



	Units		INCHES*		N	MILLIMETERS		
Dimension	Limits	MIN	NOM	MAX	MIN	NOM	MAX	
Number of Pins	n		28			28		
Pitch	р		.050			1.27		
Overall Height	Α	.093	.099	.104	2.36	2.50	2.64	
Molded Package Thickness	A2	.088	.091	.094	2.24	2.31	2.39	
Standoff §	A1	.004	.008	.012	0.10	0.20	0.30	
Overall Width	Е	.394	.407	.420	10.01	10.34	10.67	
Molded Package Width	E1	.288	.295	.299	7.32	7.49	7.59	
Overall Length	D	.695	.704	.712	17.65	17.87	18.08	
Chamfer Distance	h	.010	.020	.029	0.25	0.50	0.74	
Foot Length	L	.016	.033	.050	0.41	0.84	1.27	
Foot Angle Top	ф	0	4	8	0	4	8	
Lead Thickness	С	.009	.011	.013	0.23	0.28	0.33	
Lead Width	В	.014	.017	.020	0.36	0.42	0.51	
Mold Draft Angle Top	α	0	12	15	0	12	15	
Mold Draft Angle Bottom	β	0	12	15	0	12	15	

<sup>\*</sup> Controlling Parameter

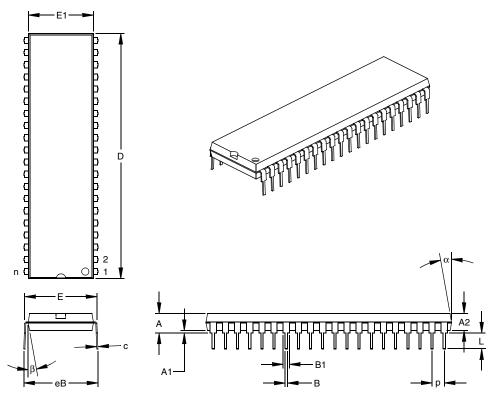
#### Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side. JEDEC Equivalent: MS-013

Drawing No. C04-052

<sup>§</sup> Significant Characteristic

### 40-Lead Plastic Dual In-line (P) - 600 mil Body (PDIP)



	Units		INCHES*		MILLIMETERS			
Dimensio	n Limits	MIN	NOM	MAX	MIN	NOM	MAX	
Number of Pins	n	40				40		
Pitch	р		.100			2.54		
Top to Seating Plane	Α	.160	.175	.190	4.06	4.45	4.83	
Molded Package Thickness	A2	.140	.150	.160	3.56	3.81	4.06	
Base to Seating Plane	A1	.015			0.38			
Shoulder to Shoulder Width	Е	.595	.600	.625	15.11	15.24	15.88	
Molded Package Width	E1	.530	.545	.560	13.46	13.84	14.22	
Overall Length	D	2.045	2.058	2.065	51.94	52.26	52.45	
Tip to Seating Plane	L	.120	.130	.135	3.05	3.30	3.43	
Lead Thickness	С	.008	.012	.015	0.20	0.29	0.38	
Upper Lead Width	B1	.030	.050	.070	0.76	1.27	1.78	
Lower Lead Width	В	.014	.018	.022	0.36	0.46	0.56	
Overall Row Spacing §	eВ	.620	.650	.680	15.75	16.51	17.27	
Mold Draft Angle Top	α	5	10	15	5	10	15	
Mold Draft Angle Bottom	β	5	10	15	5	10	15	

<sup>\*</sup> Controlling Parameter

#### Notes:

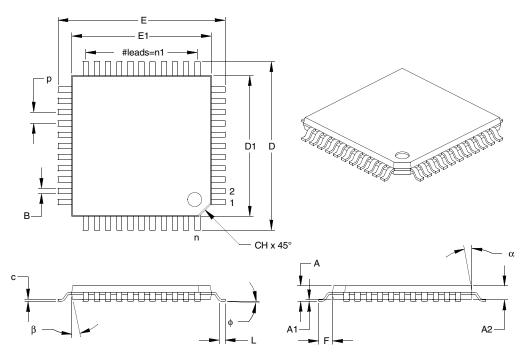
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MO-011 Drawing No. C04-016

© 2006 Microchip Technology Inc. Preliminary DS39632C-page 405

<sup>§</sup> Significant Characteristic

### 44-Lead Plastic Thin-Quad Flatpack (PT) 10x10x1 mm Body, 1.0/0.10 mm Lead Form (TQFP)



	Units		INCHES		М	MILLIMETERS*		
Dimension Lin	nits	MIN	NOM	MAX	MIN	MON	MAX	
Number of Pins	n		44			4	4	
Pitch	р		.031			0.8	80	
Pins per Side	n1	11				1	1	
Overall Height	Α	.039	.043	.047	1.00	1.10	1.20	
Molded Package Thickness	A2	.037	.039	.041	0.95	1.00	1.05	
Standoff	A1	.002	.004	.006	0.05	0.10	0.15	
Foot Length	L	.018	.024	.030	0.45	0.60	0.75	
Footprint (Reference)	F	.039 REF.				1.00 REF.		
Foot Angle	ф	0	3.5	7	0	3.5	7	
Overall Width	Е	.463	.472	.482	11.75	12.00	12.25	
Overall Length	D	.463	.472	.482	11.75	12.00	12.25	
Molded Package Width	E1	.390	.394	.398	9.90	10.00	10.10	
Molded Package Length	D1	.390	.394	.398	9.90	10.00	10.10	
Lead Thickness	С	.004	.006	.008	0.09	0.15	0.20	
Lead Width	В	.012	.015	.017	0.30	0.38	0.44	
Pin 1 Corner Chamfer	CH	.025	.035	.045	0.64	0.89	1.14	
Mold Draft Angle Top	α	5	10	15	5	10	15	
Mold Draft Angle Bottom	β	5	10	15	5	10	15	

<sup>\*</sup> Controlling Parameter

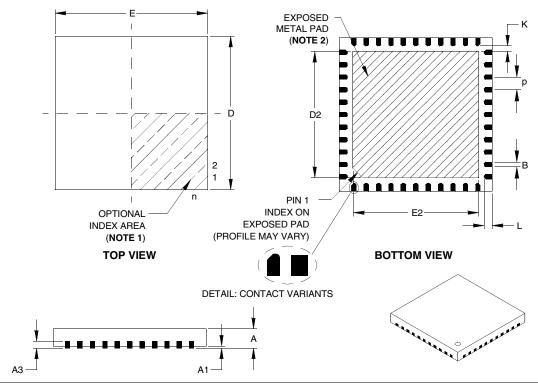
#### Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side. REF: Reference Dimension, usually without tolerance, for information purposes only.

See ASME Y14.5M JEDEC Equivalent: MS-026 Drawing No. C04-076

Revised 07-22-05

### 44-Lead Plastic Quad Flat No Lead Package (ML) 8x8 mm Body (QFN)



	Units		INCHES		МІ	MILLIMETERS*		
Dimension Li	mits	MIN	NOM	MAX	MIN	NOM	MAX	
Number of Contacts	n		44			44		
Pitch	р	.026 BSC				0.65 BSC		
Overall Height	Α	.031	.035	.039	0.80	0.90	1.00	
Standoff	A1	.000	.001	.002	0	0.02	0.05	
Base Thickness	А3	.010 REF				0.25 REF		
Overall Width	Е	.309	.315	.321	7.85	8.00	8.15	
Exposed Pad Width	E2	.236	.258	.260	5.99	6.55	6.60	
Overall Length	D	.309	.315	.321	7.85	8.00	8.15	
Exposed Pad Length	D2	.236	.258	.260	5.99	6.55	6.60	
Contact Width	В	.008	.013	.013	0.20	0.33	0.35	
Contact Length §	L	.014	.016	.019	0.35	0.40	0.48	
Contact-to-Exposed-Pad §	K	.014	-	-	0.20	-	-	

<sup>\*</sup> Controlling Parameter

§ Significant Characteristic

#### Notes

 $\textbf{1.} \ \textbf{Pin 1} \ \textbf{visual index feature may vary, but must be located within the hatched area.}$ 

2. Exposed pad varies according to die attach paddle size.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

See ASME Y14.5M

REF: Reference Dimension, usually without tolerance, for information purposes only.

See ASME Y14.5M

JEDEC equivalent: M0-220

Drawing No. C04-103

Revised 09-12-05

NOTES:

### APPENDIX A: REVISION HISTORY

### Revision A (May 2004)

Original data sheet for PIC18F2455/2550/4455/4550 devices.

### **Revision B (October 2004)**

This revision includes updates to the Electrical Specifications in **Section 28.0** "Electrical Characteristics" and includes minor corrections to the data sheet text.

### **Revision C (February 2006)**

This revision includes updates to Section 19.0 "Master Synchronous Serial Port (MSSP) Module", Section 20.0 "Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART)" and the Electrical Specifications in Section 28.0 "Electrical Characteristics" and includes minor corrections to the data sheet text.

### 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

Features	PIC18F2455	PIC18F2550	PIC18F4455	PIC18F4550
Program Memory (Bytes)	24576	32768	24576	32768
Program Memory (Instructions)	12288	16384	12288	16384
Interrupt Sources	19	19	20	20
I/O Ports	Ports A, B, C, (E)	Ports A, B, C, (E)	Ports A, B, C, D, E	Ports A, B, C, D, E
Capture/Compare/PWM Modules	2	2	1	1
Enhanced Capture/Compare/ PWM Modules	0	0	1	1
Parallel Communications (SPP)	No	No	Yes	Yes
10-Bit Analog-to-Digital Module	10 input channels	10 input channels	13 input channels	13 input channels
Packages	28-pin PDIP 28-pin SOIC	28-pin PDIP 28-pin SOIC	40-pin PDIP 44-pin TQFP 44-pin QFN	40-pin PDIP 44-pin TQFP 44-pin QFN

## APPENDIX C: CONVERSION CONSIDERATIONS

This appendix discusses the considerations for converting from previous versions of a device to the ones listed in this data sheet. Typically, these changes are due to the differences in the process technology used. An example of this type of conversion is from a PIC16C74A to a PIC16C74B.

**Not Applicable** 

# APPENDIX D: MIGRATION FROM BASELINE TO ENHANCED DEVICES

This section discusses how to migrate from a Baseline device (i.e., PIC16C5X) to an Enhanced MCU device (i.e., PIC18FXXX).

The following are the list of modifications over the PIC16C5X microcontroller family:

**Not Currently Available** 

# APPENDIX E: MIGRATION FROM MID-RANGE TO ENHANCED DEVICES

A detailed discussion of the differences between the mid-range MCU devices (i.e., PIC16CXXX) and the enhanced devices (i.e., PIC18FXXX) is provided in *AN716, "Migrating Designs from PIC16C74A/74B to PIC18C442"*. The changes discussed, while device specific, are generally applicable to all mid-range to enhanced device migrations.

This Application Note is available as Literature Number DS00716.

APPENDIX F: MIGRATION FROM HIGH-END TO ENHANCED DEVICES

A detailed discussion of the migration pathway and differences between the high-end MCU devices (i.e., PIC17CXXX) and the enhanced devices (i.e., PIC18FXXX) is provided in *AN726*, "*PIC17CXXX to PIC18CXXX Migration*". This Application Note is available as Literature Number DS00726.

NOTES:

INDEX		A/D	269
A		Analog Input Model	
	050	Baud Rate Generator	
A/D		Capture Mode Operation	
Acquisition Requirements		Comparator Analog Input Model	
ADCON0 Register		Comparator I/O Operating Modes	
ADCON1 Register		Comparator Output	
ADCON2 Register		Comparator Voltage Reference	
ADRESH Register		Comparator Voltage Reference Output	
ADRESL Register		Buffer Example	27
Analog Port Pins, Configuring		Compare Mode Operation	
Associated Registers		Device Clock	
Configuring the Module		Enhanced PWM	
Conversion Clock (TAD)		EUSART Receive	
Conversion Requirements		EUSART Transmit	248
Conversion Status (GO/DONE Bit)		External Power-on Reset Circuit	
Conversions		(Slow VDD Power-up)	4
Converter Characteristics		Fail-Safe Clock Monitor	
Converter Interrupt, Configuring		Generic I/O Port	
Discharge		High/Low-Voltage Detect with External Input	
Operation in Power-Managed Modes	266	MSSP (I <sup>2</sup> C Master Mode)	
Selecting and Configuring		MSSP (I <sup>2</sup> C Mode)	
Acquisition Time		MSSP (SPI Mode)	
Special Event Trigger (CCP2)		On-Chip Reset Circuit	
Special Event Trigger (ECCP)		PIC18F2455/2550	
Use of the CCP2 Trigger		PIC18F4455/4550	
Absolute Maximum Ratings		PLL (HS Mode)	
AC (Timing) Characteristics	378	PWM Operation (Simplified)	
Load Conditions for Device Timing		Reads from Flash Program Memory	
Specifications		Single Comparator	
Parameter Symbology		SPP Data Path	
Temperature and Voltage Specifications		Table Read Operation	
Timing Conditions	379	Table Write Operation	
AC Characteristics		Table Writes to Flash Program Memory	
Internal RC Accuracy	381	Timer0 in 16-Bit Mode	
Access Bank		Timer0 in 8-Bit Mode	
Mapping with Indexed Literal Offset Mode		Timer1	
ACKSTAT		Timer1 (16-Bit Read/Write Mode)	
ACKSTAT Status Flag		Timer2	
ADCON0 Register		Timer3	
GO/DONE Bit		Timer3 (16-Bit Read/Write Mode)	
ADCON1 Register		USB Interrupt Logic	
ADCON2 Register		USB Peripheral and Options	
ADDFSR		Watchdog Timer	
ADDLW		BN	
ADDULNK		BNC	
ADDWF		BNN	
ADDWFC		BNOV	
ADRESH Register		BNZ	_
ADRESL Register25	9, 262	BOR. See Brown-out Reset.	010
Analog-to-Digital Converter. See A/D.		BOV	30
and BSR		BRA	
ANDLW	-	Break Character (12-Bit) Transmit and Receive	
ANDWF	315	BRG. See Baud Rate Generator.	23
Assembler		Brown-out Reset (BOR)	11
MPASM Assembler	358	Detecting	
В		Disabling in Sleep Mode	
		Software Enabled	
Baud Rate Generator			
BC		BSF	
BCF		BTFSC	
BF		BTFSS	
BF Status Flag	227	BTG BZ	32
		127	3/2

C		Operation	27 <sup>-</sup>
C Compilers		Operation During Sleep	
MPLAB C18	358	Outputs	
MPLAB C30		Reference	27
CALL		External Signal	27 <sup>-</sup>
CALLW		Internal Signal	27 <sup>-</sup>
Capture (CCP Module)		Response Time	27 <sup>-</sup>
CCP Pin Configuration		Comparator Specifications	37
<u>-</u>		Comparator Voltage Reference	
CCPRxH:CCPRxL Registers		Accuracy and Error	
Prescaler		Associated Registers	
Software Interrupt		Configuring	
Timer1/Timer3 Mode Selection		Connection Considerations	
Capture (ECCP Module)	150	Effects of a Reset	
Capture/Compare (CCP Module)		Operation During Sleep	
Associated Registers		Compare (CCP Module)	
Capture/Compare/PWM (CCP)	141	CCP Pin Configuration	
Capture Mode. See Capture.		CCPRx Register	
CCP Mode and Timer Resources		Software Interrupt	
CCP2 Pin Assignment		Special Event Trigger 139, 1	
CCPRxH Register		Timer1/Timer3 Mode Selection	
CCPRxL Register	142		
Compare Mode. See Compare.		Compare (ECCP Module)	
Interaction of Two CCP Modules for		Special Event Trigger	
Timer Resources	142	Configuration Bits	
Module Configuration	142	Configuration Register Protection	
Clock Sources		Context Saving During Interrupts	
Effects of Power-Managed Modes		Conversion Considerations	
Selecting the 31 kHz Source		CPFSEQ	324
Selection Using OSCCON Register		CPFSGT	325
CLRF		CPFSLT	325
CLRWDT		Crystal Oscillator/Ceramic Resonator	2
	323	Customer Change Notification Service	423
Code Examples	00	Customer Notification Service	423
16 x 16 Signed Multiply Routine		Customer Support	
16 x 16 Unsigned Multiply Routine			
8 x 8 Signed Multiply Routine		D	
8 x 8 Unsigned Multiply Routine		Data Addressing Modes	72
Changing Between Capture Prescalers		Comparing Addressing Modes with the	
Computed GOTO Using an Offset Value		Extended Instruction Set Enabled	76
Data EEPROM Read		Direct	
Data EEPROM Refresh Routine	92	Indexed Literal Offset	
Data EEPROM Write		Indirect	
Erasing a Flash Program Memory Row	84	Inherent and Literal	
Fast Register Stack	60	Data EEPROM	/ 2
How to Clear RAM (Bank 1) Using			001
Indirect Addressing	72	Code Protection	
Implementing a Real-Time Clock Using a		Data EEPROM Memory	
Timer1 Interrupt Service	133	Associated Registers	
Initializing PORTA		EECON1 and EECON2 Registers	
Initializing PORTB		Operation During Code-Protect	
Initializing PORTC		Protection Against Spurious Write	
Initializing PORTD		Reading	
Initializing PORTE		Using	92
Loading the SSPBUF (SSPSR) Register		Write Verify	9 <sup>.</sup>
, , ,		Writing	9 <sup>.</sup>
Reading a Flash Program Memory Word	03	Data Memory	63
Saving STATUS, WREG and BSR	100	Access Bank	
Registers in RAM		and the Extended Instruction Set	
Writing to Flash Program Memory		Bank Select Register (BSR)	
Code Protection		General Purpose Registers	
COMF		Map for PIC18F2455/2550/4455/4550 Devices	
Comparator		Special Function Registers	
Analog Input Connection Considerations	273	Map	
Associated Registers	273	USB RAM	
Configuration			
Effects of a Reset		DAW	32t
Interrupts	272	DC and AC Characteristics	001
•		Graphs and Tables	399

DC Characteristics372
Power-Down and Supply Current364
Supply Voltage363
DCFSNZ327
DECF
DECFSZ
Dedicated ICD/ICSP Port305
Development Support357
Device Differences
Device Overview7
Features (table)9
New Core Features7
Other Special Features8
Device Reset Timers47
Oscillator Start-up Timer (OST)47
PLL Lock Time-out47
Power-up Timer (PWRT)47
Direct Addressing73
E
_
Effect on Standard PIC Instructions
Effect on Standard PIC MCU Instructions
Electrical Characteristics
Enhanced Capture/Compare/PWM (ECCP)149
Associated Registers
Capture and Compare Modes150
Capture Mode. See Capture (ECCP Module).
Outputs and Configuration
Pin Configurations for ECCP1150
PWM Mode. See PWM (ECCP Module).
Standard PWM Mode
Timer Resources150
Enhanced Universal Synchronous Asynchronous
Receiver Transmitter (EUSART). See EUSART.
Receiver Transmitter (EUSART). See EUSART. Equations
Receiver Transmitter (EUSART). See EUSART. Equations A/D Acquisition Time264
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.         Equations         A/D Acquisition Time       264         A/D Minimum Charging Time       264         Calculating the Minimum Required       264         A/D Acquisition Time       264         Errata       5         EUSART       4         Asynchronous Mode       247         12-Bit Break Transmit and Receive       253         Associated Registers, Receive       251         Associated Registers, Transmit       249         Auto-Wake-up on Sync Break Character       252         Receiver       250         Setting up 9-Bit Mode with       240         Address Detect       250         Transmitter       247         Baud Rate Generator       Operation in Power-Managed Modes       241         Baud Rate Generator (BRG)       241         Associated Registers       242         Auto-Baud Rate Detect       245
Receiver Transmitter (EUSART). See EUSART.           Equations           A/D Acquisition Time         264           A/D Minimum Charging Time         264           Calculating the Minimum Required         264           A/D Acquisition Time         264           Errata         5           EUSART         4           Asynchronous Mode         247           12-Bit Break Transmit and Receive         253           Associated Registers, Receive         251           Associated Registers, Transmit         249           Auto-Wake-up on Sync Break Character         252           Receiver         250           Setting up 9-Bit Mode with         240           Address Detect         250           Transmitter         247           Baud Rate Generator         Operation in Power-Managed Modes         241           Baud Rate Generator (BRG)         241           Associated Registers         242           Auto-Baud Rate Detect         245           Baud Rate Error, Calculating         242
Receiver Transmitter (EUSART). See EUSART.           Equations           A/D Acquisition Time         264           A/D Minimum Charging Time         264           Calculating the Minimum Required         264           A/D Acquisition Time         264           Errata         5           EUSART         4           Asynchronous Mode         247           12-Bit Break Transmit and Receive         253           Associated Registers, Receive         251           Associated Registers, Transmit         249           Auto-Wake-up on Sync Break Character         252           Receiver         250           Setting up 9-Bit Mode with         240           Address Detect         250           Transmitter         247           Baud Rate Generator         Operation in Power-Managed Modes         241           Baud Rate Generator (BRG)         241           Associated Registers         242           Auto-Baud Rate Detect         245           Baud Rate Error, Calculating         242           Baud Rates, Asynchronous Modes         243
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.           Equations           A/D Acquisition Time         264           A/D Minimum Charging Time         264           Calculating the Minimum Required         264           A/D Acquisition Time         264           Errata         5           EUSART         247           Asynchronous Mode         247           12-Bit Break Transmit and Receive         253           Associated Registers, Receive         251           Associated Registers, Transmit         249           Auto-Wake-up on Sync Break Character         252           Receiver         250           Setting up 9-Bit Mode with         240           Address Detect         250           Transmitter         247           Baud Rate Generator         20           Operation in Power-Managed Modes         241           Baud Rate Generator (BRG)         241           Associated Registers         242           Auto-Baud Rate Detect         245           Baud Rate Error, Calculating         242           Baud Rates, Asynchronous Modes         243           High Baud Rate Select (BRGH Bit)         241           Sampling         241 </td
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time
Receiver Transmitter (EUSART). See EUSART.  Equations  A/D Acquisition Time

Synchronous Slave Mode	
Associated Registers, Receive	
Associated Registers, Transmit	
Reception	
Transmission	
Extended Instruction Set	
ADDFSR	350
ADDULNK	
and Using MPLAB IDE Tools	356
CALLW	
Considerations for Use	354
MOVSF	351
MOVSS	352
PUSHL	
SUBFSR	353
SUBULNK	353
Syntax	349
External Clock Input	26
F	
•	
Fail-Safe Clock Monitor	
Exiting the Operation	
Interrupts in Power-Managed Modes	
POR or Wake-up from Sleep	
WDT During Oscillator Failure	
Fast Register Stack	
Firmware Instructions	
Flash Program Memory	
Associated Registers	
Control Registers	
EECON1 and EECON2	
TABLAT (Table Latch) Register	
TBLPTR (Table Pointer) Register	
Erase Sequence	
Erasing	
Operation During Code-Protect	87
Protection Against Spurious Writes	
Reading	83
Table Pointer	
Boundaries Based on Operation	
Table Pointer Boundaries	
Table Reads and Table Writes	
Unexpected Termination of Write	
Write Sequence	
Write Verify	
Writing To	85
FSCM. See Fail-Safe Clock Monitor.	
G	
<del></del>	000
GOTO	328
Н	
Hardwara Multipliar	0.5
Hardware Multiplier	
Operation	
Performance Comparison	95

High/Low-Voltage Detect	279	Instruction Cycle	61
Applications		Clocking Scheme	
Associated Registers		Flow/Pipelining	
Characteristics	377	Instruction Set	
Current Consumption		ADDLW	
Effects of a Reset		ADDWF	313
Operation		ADDWF (Indexed Literal Offset mode)	
During Sleep		ADDWFC	
Setup		ANDLW	
Start-up Time		ANDWF	_
Typical Application		BC	
HLVD. See High/Low-Voltage Detect.	202	BCF	
TIEVD. See High/Low-Voltage Detect.		BN	
I		BNC	
I/O Ports	111	BNN	_
I <sup>2</sup> C Mode (MSSP)		BNOV	_
	220		
Acknowledge Sequence Timing		BNZ	
Associated Registers		BOV	_
Baud Rate Generator	223	BRA	
Bus Collision		BSF	
During a Repeated Start Condition		BSF (Indexed Literal Offset mode)	
During a Stop Condition		BTFSC	320
Clock Arbitration		BTFSS	320
Clock Stretching	216	BTG	321
10-Bit Slave Receive Mode (SEN = 1)	216	BZ	322
10-Bit Slave Transmit Mode	216	CALL	322
7-Bit Slave Receive Mode (SEN = 1)	216	CLRF	323
7-Bit Slave Transmit Mode		CLRWDT	
Clock Synchronization and the CKP Bit		COMF	
Effect of a Reset		CPFSEQ	_
General Call Address Support	_	CPFSGT	_
I <sup>2</sup> C Clock Rate w/BRG			
		CPFSLT	
Master Mode		DAW	
Operation		DCFSNZ	_
Reception		DECF	
Repeated Start Condition Timing		DECFSZ	
Start Condition Timing		General Format	309
Transmission		GOTO	
Transmit Sequence	222	INCF	328
Multi-Master Communication, Bus Collision		INCFSZ	329
and Arbitration	231	INFSNZ	329
Multi-Master Mode	231	IORLW	330
Operation	207	IORWF	330
Read/Write Bit Information (R/W Bit)	207, 209	LFSR	
Registers		MOVF	
Serial Clock (RB1/AN10/INT1/SCK/SCL)		MOVFF	
Slave Mode		MOVLB	
Addressing		MOVLW	
Addressing Masking		MOVWF	
Reception			
		MULLW	
Transmission		MULWF	
Sleep Operation		NEGF	
Stop Condition Timing		NOP	
ID Locations		Opcode Field Descriptions	
Idle Modes		POP	336
INCF		PUSH	336
INCFSZ		RCALL	337
In-Circuit Debugger		RESET	337
In-Circuit Serial Programming (ICSP)	285, 305	RETFIE	338
Indexed Literal Offset Addressing		RETLW	
and Standard PIC18 Instructions	354	RETURN	
Indexed Literal Offset Mode		RLCF	
Indirect Addressing		RLNCF	
INFSNZ		RRCF	
Initialization Conditions for all Registers		RRNCF	

SETF341
SETF (Indexed Literal Offset mode)355
SLEEP342
Standard Instructions
SUBFWB
SUBLW343
SUBWF
SUBWFB
SWAPF
TBLRD
TBLWT
TSTFSZ
XORLW
XORWF
INTCON Register RBIF Bit114
INTCON Registers
Inter-Integrated Circuit. See I <sup>2</sup> C.
Internal Oscillator Block
Adjustment27
INTHS, INTXT, INTCKO and INTIO Modes27
OSCTUNE Register27
Internal RC Oscillator
Use with WDT297
Internet Address423
Interrupt Sources
A/D Conversion Complete263
Capture Complete (CCP)143
Compare Complete (CCP)144
Interrupt-on-Change (RB7:RB4)114
INTn Pin109
PORTB, Interrupt-on-Change109
TMR0109
TMR0 Overflow127
TMR1 Overflow129
TMR2 to PR2 Match (PWM)146, 151
TMR3 Overflow137, 139
Interrupts97
Logic (diagram)98
USB97
Interrupts, Flag Bits
Interrupt-on-Change (RB7:RB4) Flag
(RBIF Bit)
INTOSC Frequency Drift
INTOSC, INTRC. See Internal Oscillator Block.
IORLW
IORWF
IPR Registers
ii ii ii legisteis 100
L
LFSR
Low-Voltage ICSP Programming. See Single-Supply
ICSP Programming.
103F Flogramming.
M
Master Clear Reset (MCLR)45
Master Synchronous Serial Port (MSSP). See MSSP.
Memory Organization
Data Memory
Program Memory
Memory Programming Requirements
Microchip Internet Web Site
Migration from Baseline to Enhanced Devices410
Migration from High-End to Enhanced Devices

	331
MOVFF	
MOVLB	
MOVLW	
MOVSF	
MOVSS	352
MOVWF	333
MPLAB ASM30 Assembler, Linker, Librarian	
MPLAB ICD 2 In-Circuit Debugger	359
MPLAB ICE 2000 High-Performance	
Universal In-Circuit Emulator	359
MPLAB ICE 4000 High-Performance	
Universal In-Circuit Emulator	359
MPLAB Integrated Development	
Environment Software	357
MPLAB PM3 Device Programmer	
MPLINK Object Linker/MPLIB Object Librarian	358
MSSP	
ACK Pulse 207,	209
Control Registers (general)	
I <sup>2</sup> C Mode. See I <sup>2</sup> C Mode.	
Module Overview	193
SPI Master/Slave Connection	
SPI Mode. See SPI Mode.	
SSPBUF	198
SSPSR	
MULLW	
MULWF	
	••
N	
NEGF	335
NOP	
0	
Oscillator Configuration	23
Oscillator Configuration	
EC	23
ECECIO	23 23
ECECIOECPIO	23 23 23
ECECIOECPIOECPIL	23 23 23
EC	23 23 23 23
EC	23 23 23 23 23
EC	23 23 23 23 23
EC	23 23 23 23 23 23
EC	23 23 23 23 23 23
EC	23 23 23 23 23 23 23
EC	23 23 23 23 23 23 23 23
EC	23 23 23 23 23 23 23 23 23
EC	23 23 23 23 23 25 25 25 25
EC	23 23 23 23 23 25 25 25 25 25
EC	20 20
EC	23 23 23 23 25 25 25 25 25 25 25
EC	23 23 23 23 23 23 23 23 23 23 23 23
EC	23 23 23 23 23 23 23 23 23 23 23 23 23 23 23
EC	23 24 25
EC	23 24 25
EC	23 24 25
EC	23 24 25
EC	23 24 25 
EC	23 24 24 25 
EC	23 24 24 25 
EC	23 24 24 25 
EC	23 23 23 25 31 32 35 31 32 35 31 32 35 31 35 35 35 36 36 36

Pin Functions		PORTB	
MCLR/VPP/RE3	12, 16	Associated Registers	116
NC/ICCK/ICPGC	21	I/O Summary	
NC/ICDT/ICPGD	21	LATB Register	114
NC/ICPORTS		PORTB Register	
NC/ICRST/ICVPP	21	RB1/AN10/INT1/SCK/SCL Pin	
OSC1/CLKI		RB7:RB4 Interrupt-on-Change Flag (RBIF Bit)	
OSC2/CLKO/RA6		TRISB Register	
RAO/ANO	•	PORTC	
RA1/AN1	•	Associated Registers	110
RA2/AN2/VREF-/CVREF		I/O Summary	
RA3/AN3/VREF+	•	LATC Register	
RA4/T0CKI/C1OUT/RCV	·	PORTC Register	
RA5/AN4/SS/HLVDIN/C2OUT		TRISC Register	
RB0/AN12/INT0/FLT0/SDI/SDA		PORTD	117
	·		100
RB1/AN10/INT1/SCK/SCL	•	Associated Registers	
RB2/AN8/INT2/VMO	·	I/O Summary	
RB3/AN9/CCP2/VPO	·	LATD Register	
RB4/AN11/KBI0		PORTD Register	
RB4/AN11/KBI0/CSSPP		TRISD Register	120
RB5/KBI1/PGM	·	PORTE	
RB6/KBI2/PGC	•	Associated Registers	
RB7/KBI3/PGD	*	I/O Summary	
RC0/T10S0/T13CKI		LATE Register	
RC1/T1OSI/CCP2/UOE	15, 19	PORTE Register	123
RC2/CCP1	15	TRISE Register	123
RC2/CCP1/P1A	19	Postscaler, WDT	
RC4/D-/VM	15, 19	Assignment (PSA Bit)	127
RC5/D+/VP	15, 19	Rate Select (T0PS2:T0PS0 Bits)	127
RC6/TX/CK	15, 19	Power-Managed Modes	35
RC7/RX/DT/SDO	15, 19	and Multiple Sleep Commands	
RD0/SPP0	20	and PWM Operation	
RD1/SPP1	20	Clock Sources	
RD2/SPP2	20	Clock Transitions and Status Indicators	
RD3/SPP3		Entering	
RD4/SPP4		Exiting Idle and Sleep Modes	
RD5/SPP5/P1B		by Interrupt	
RD6/SPP6/P1C		by Reset	
RD7/SPP7/P1D		by WDT Time-out	
RE0/AN5/CK1SPP		Without an Oscillator Start-up Delay	
RE1/AN6/CK2SPP		Idle	
RE2/AN7/OESPP		Idle Modes	08
VDD		PRI_IDLE	40
	- ,	_	
Vss	,	RC_IDLE	
VUSB	15, 21	SEC_IDLE	
Pinout I/O Descriptions	40	Run Modes	
PIC18F2455/2550		PRI_RUN	
PIC18F4455/4550		RC_RUN	
PIR Registers		SEC_RUN	
PLL Frequency Multiplier	26	Selecting	
HSPLL, XTPLL, ECPLL and ECPIO		Sleep	
Oscillator Modes	26	Summary (table)	
PLL Lock Time-out	47	Power-on Reset (POR)	
POP	336	Oscillator Start-up Timer (OST)	47
POR. See Power-on Reset.		Power-up Timer (PWRT)	
PORTA		Time-out Sequence	
Associated Registers	113	Power-up Delays	33
I/O Summary		Power-up Timer (PWRT)	
LATA Register	111	Prescaler	
PORTA Register		Timer2	152
TRISA Register		Prescaler, Timer0	
•		Assignment (PSA Bit)	
		Rate Select (T0PS2:T0PS0 Bits)	

Prescaler, Timer2	147	Reader Response	424
PRI_IDLE Mode	40	Register File	6
PRI_RUN Mode	36	Register File Summary	
Program Counter		Registers	
PCL, PCH and PCU Registers		ADCON0 (A/D Control 0)	259
PCLATH and PCLATU Registers		ADCON1 (A/D Control 1)	
Program Memory		ADCON2 (A/D Control 2)	
and the Extended Instruction Set	75	BAUDCON (Baud Rate Control)	
Code Protection		BDnSTAT (Buffer Descriptor n Status,	
Instructions		CPU Mode)	17:
Two-Word	_	BDnSTAT (Buffer Descriptor n Status,	
Interrupt Vector		SIE Mode)	174
Look-up Tables		CCP1CON (ECCP Control)	
Map and Stack (diagram)		CCPxCON (Standard CCPx Control)	
Reset Vector		CMCON (Comparator Control)	
Program Verification and Code Protection		CONFIG1H (Configuration 1 High)	
Associated Registers		CONFIG1L (Configuration 1 Low)	
Programming, Device Instructions		CONFIG2H (Configuration 2 High)	
	307		
Pulse-Width Modulation. See PWM (CCP Module)		CONFIG2L (Configuration 2 Low)	
and PWM (ECCP Module).	000	CONFIGAL (Configuration 3 High)	
PUSH		CONFIGAL (Configuration 4 Low)	
PUSH and POP Instructions		CONFIG5H (Configuration 5 High)	
PUSHL	352	CONFIG5L (Configuration 5 Low)	
PWM (CCP Module)		CONFIG6H (Configuration 6 High)	
Associated Registers		CONFIG6L (Configuration 6 Low)	
Auto-Shutdown (CCP1 Only)		CONFIG7H (Configuration 7 High)	
Duty Cycle	146	CONFIG7L (Configuration 7 Low)	29
Example Frequencies/Resolutions	147	CVRCON (Comparator Voltage	
Period	146	Reference Control)	27
Setup for PWM Operation	147	DEVID1 (Device ID 1)	
TMR2 to PR2 Match		DEVID2 (Device ID 2)	
PWM (ECCP Module)	151	ECCP1AS (Enhanced Capture/Compare/PWM	
CCPR1H:CCPR1L Registers		Auto-Shutdown Control)	159
Direction Change in Full-Bridge		ECCP1DEL (PWM Dead-Band Delay)	
Output Mode	156	EECON1 (Data EEPROM Control 1)	
Duty Cycle		HLVDCON (High/Low-Voltage	. 01, 0
Effects of a Reset		Detect Control)	270
Enhanced PWM Auto-Shutdown			
Enhanced PWM Mode		INTCON (Interrupt Control)	
		INTCON2 (Interrupt Control 2)	
Example Frequencies/Resolutions		INTCON3 (Interrupt Control 3)	
Full-Bridge Application Example		IPR1 (Peripheral Interrupt Priority 1)	
Full-Bridge Mode		IPR2 (Peripheral Interrupt Priority 2)	
Half-Bridge Mode	154	OSCCON (Oscillator Control)	
Half-Bridge Output Mode		OSCTUNE (Oscillator Tuning)	
Applications Example		PIE1 (Peripheral Interrupt Enable 1)	
Operation in Power-Managed Modes	161	PIE2 (Peripheral Interrupt Enable 2)	10
Operation with Fail-Safe Clock Monitor		PIR1 (Peripheral Interrupt Request	
Output Configurations	152	(Flag) 1)	102
Output Relationships (Active-High)	153	PIR2 (Peripheral Interrupt Request	
Output Relationships (Active-Low)	153	(Flag) 2)	10
Period	151	PORTE	12
Programmable Dead-Band Delay	158	RCON (Reset Control)	44, 108
Setup for PWM Operation		RCSTA (Receive Status and Control)	239
Start-up Considerations		SPPCFG (SPP Configuration)	
TMR2 to PR2 Match		SPPCON (SPP Control)	
		SPPEPS (SPP Endpoint Address	
Q		and Status)	19
Q Clock	147, 152	SSPCON1 (MSSP Control 1, I <sup>2</sup> C Mode)	
	,	SSPCON1 (MSSP Control 1, SPI Mode)	
R		SSPCON1 (MSSP Control 2, I <sup>2</sup> C	13
RAM. See Data Memory.		Master Mode)	201
RC_IDLE Mode	<b>41</b>	CCDCONO (MCCD Control O 120	20:
RC_RUN Mode		SSPCON2 (MSSP Control 2, I <sup>2</sup> C	000
RCALL		Slave Mode)	206
RCON Register		SSPSTAT (MSSP Status, I <sup>2</sup> C Mode)	
· · · · · · · · · · · · · · · · · · ·	<b>5</b> 0	SSPSTAT (MSSP Status, SPI Mode)	194
Bit Status During Initialization			

STATUS		Serial Data In	
STKPTR (Stack Pointer)	59	Serial Data Out	193
T0CON (Timer0 Control)	125	Slave Mode	199
T1CON (Timer1 Control)	129	Slave Select	193
T2CON (Timer2 Control)	135	Slave Select Synchronization	199
T3CON (Timer3 Control)	137	SPI Clock	198
TXSTA (Transmit Status and Control)	238	Typical Connection	197
UCFG (USB Configuration)		SPP. See Streaming Parallel Port.	
UCON (USB Control)		SS	193
UEIE (USB Error Interrupt Enable)		SSPOV	
UEIR (USB Error Interrupt Status)		SSPOV Status Flag	227
UEPn (USB Endpoint n Control)		SSPSTAT Register	
UIE (USB Interrupt Enable)		R/W Bit	209
UIR (USB Interrupt Status)		SSPxSTAT Register	
USTAT (USB Status)		R/W Bit	207
WDTCON (Watchdog Timer Control)		Stack Full/Underflow Resets	
RESET		STATUS Register	
Reset State of Registers		Streaming Parallel Port	
Resets		Associated Registers	
Brown-out Reset (BOR)	·	Clocking Data	
` ,		<del>_</del>	
Oscillator Start-up Timer (OST)		Configuration	
Power-on Reset (POR)		Internal Pull-ups	
Power-up Timer (PWRT)		Interrupts	
RETFIE		Microcontroller Control Setup	
RETLW		Reading from (Microcontroller Mode)	191
RETURN		Transfer of Data Between USB SIE	
Return Address Stack		and SPP (diagram)	
and Associated Registers		USB Control Setup	
Return Stack Pointer (STKPTR)		Wait States	
Revision History		Writing to (Microcontroller Mode)	
RLCF	339	SUBFSR	353
RLNCF		SUBFWB	_
RRCF	340	SUBLW	343
RRNCF	341	SUBULNK	
	341	SUBULNKSUBWF	
RRNCF	341		343
		SUBWF	343 344
S	193	SUBWFSUBWFBSWAPF	343 344
<b>S</b> SCK	193 193	SUBWFSUBWFB	343 344
<b>S</b> SCKSDI	193 193 193	SUBWFSUBWFBSWAPF	343 344
<b>S</b> SCKSDISDO	193 193 193 40	SUBWFSUBWFBSWAPFSWAPF	343 344 344
SCK		SUBWFSUBWFBSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPF	343 344 344
\$ SCK		SUBWFSUBWFBSWAPF	343 344 344 127
\$ SCK		SUBWFSUBWFBSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPF	343 344 344 127 126 127
SCK SDI SDO SEC_IDLE Mode SEC_RUN Mode Serial Clock, SCK Serial Data In (SDI)		SUBWFSUBWFBSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPF	343 344 127 126 127 126
S SCK		SUBWFSUBWFBSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPF	343 344 127 126 127 126 126
SCK		SUBWFSUBWFBSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPFSWAPF	343 344 127 126 127 126 82
\$  \$CK		SUBWF	343 344 127 126 126 126 82 60
\$  \$CK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPS0 Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT	343 344 127 126 126 126 82 60 345
\$  \$CK		SUBWF	343 344 127 126 127 126 82 60 345 346
SCK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPS0 Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT Time-out in Various Situations (table)	343 344 127 126 126 126 
SCK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPS0 Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT Time-out in Various Situations (table) Timer0  16-Bit Mode Timer Reads and Writes	343 344 127 126 126 82 60 345 346 47 125
SCK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPS0 Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT Time-out in Various Situations (table) Timer0  16-Bit Mode Timer Reads and Writes Associated Registers	343 344 127 126 127 126 82 
SCK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPS0 Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT Time-out in Various Situations (table) Timer0  16-Bit Mode Timer Reads and Writes Associated Registers Clock Source Edge Select (TOSE Bit)	343344127126126
SCK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPS0 Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT Time-out in Various Situations (table) Timer0  16-Bit Mode Timer Reads and Writes Associated Registers Clock Source Edge Select (TOSE Bit) Clock Source Select (TOCS Bit)	343344127126126
SCK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPSO Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT Time-out in Various Situations (table) Timer0  16-Bit Mode Timer Reads and Writes Associated Registers Clock Source Edge Select (TOSE Bit) Operation	343344127126126
SCK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPS0 Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT Time-out in Various Situations (table) Timer0  16-Bit Mode Timer Reads and Writes Associated Registers Clock Source Edge Select (TOSE Bit) Operation Overflow Interrupt	34334412712612634534647125126126126126126126126
SCK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPSO Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT Time-out in Various Situations (table) Timer0  16-Bit Mode Timer Reads and Writes Associated Registers Clock Source Edge Select (TOSE Bit) Operation Overflow Interrupt Prescaler	343344127126126
SCK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPS0 Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT Time-out in Various Situations (table) Timer0  16-Bit Mode Timer Reads and Writes Associated Registers Clock Source Edge Select (TOSE Bit) Operation Overflow Interrupt Prescaler Switching Assignment	343344127126126
SCK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPSO Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT Time-out in Various Situations (table) Timer0  16-Bit Mode Timer Reads and Writes Associated Registers Clock Source Edge Select (TOSE Bit) Clock Source Select (TOCS Bit) Operation Overflow Interrupt Prescaler Switching Assignment Prescaler. See Prescaler, Timer0.	343344127126126
SCK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPSO Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT Time-out in Various Situations (table) Timer0  16-Bit Mode Timer Reads and Writes Associated Registers Clock Source Edge Select (TOSE Bit) Clock Source Select (TOCS Bit) Operation Overflow Interrupt Prescaler Switching Assignment Prescaler. See Prescaler, Timer0. Timer1	
SCK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPS0 Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT Time-out in Various Situations (table) Timer0  16-Bit Mode Timer Reads and Writes Associated Registers Clock Source Edge Select (TOSE Bit) Clock Source Select (TOCS Bit) Operation Overflow Interrupt Prescaler Switching Assignment Prescaler. See Prescaler, Timer0. Timer1  16-Bit Read/Write Mode	
SCK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPSO Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT Time-out in Various Situations (table) Timer0  16-Bit Mode Timer Reads and Writes Associated Registers Clock Source Edge Select (TOSE Bit) Clock Source Select (TOCS Bit) Operation Overflow Interrupt Prescaler Switching Assignment Prescaler. See Prescaler, Timer0. Timer1  16-Bit Read/Write Mode Associated Registers	
SCK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPS0 Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT Time-out in Various Situations (table) Timer0  16-Bit Mode Timer Reads and Writes Associated Registers Clock Source Edge Select (TOSE Bit) Clock Source Select (TOCS Bit) Operation Overflow Interrupt Prescaler Switching Assignment Prescaler. See Prescaler, Timer0. Timer1  16-Bit Read/Write Mode	
SCK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPSO Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT Time-out in Various Situations (table) Timer0  16-Bit Mode Timer Reads and Writes Associated Registers Clock Source Edge Select (TOSE Bit) Clock Source Select (TOCS Bit) Operation Overflow Interrupt Prescaler Switching Assignment Prescaler. See Prescaler, Timer0. Timer1  16-Bit Read/Write Mode Associated Registers	
SCK		SUBWF SUBWFB SWAPF  T  TOCON Register PSA Bit TOCS Bit TOPS2:TOPS0 Bits TOSE Bit Table Pointer Operations (table) Table Reads/Table Writes TBLRD TBLWT Time-out in Various Situations (table) Timer0  16-Bit Mode Timer Reads and Writes Associated Registers Clock Source Edge Select (TOSE Bit) Clock Source Select (TOCS Bit) Operation Overflow Interrupt Prescaler Switching Assignment Prescaler. See Prescaler, Timer0.  Timer1  16-Bit Read/Write Mode Associated Registers Interrupt	

Oscillator	129, 131	Clock/Instruction (
Layout Considerations	·	EUSART Synchro
Low-Power Option		(Master/Slav
Using Timer1 as a Clock Source		EUSART Synchro
Overflow Interrupt		(Master/Slav
Resetting, Using a Special Event	120	Example SPI Mas
	122	Example SPI Mas
Trigger Output (CCP)		•
Special Event Trigger (ECCP)		Example SPI Slav
TMR1H Register		Example SPI Slav
TMR1L Register		External Clock (Al
Use as a Real-Time Clock		Fail-Safe Clock M
Timer2		First Start Bit Timi
Associated Registers		Full-Bridge PWM
Interrupt		Half-Bridge PWM
Operation		High/Low-Voltage
Output		High-Voltage Dete
PR2 Register		I <sup>2</sup> C Bus Data
TMR2 to PR2 Match Interrupt		I <sup>2</sup> C Bus Start/Stop
Timer3		I <sup>2</sup> C Master Mode
16-Bit Read/Write Mode		I <sup>2</sup> C Master Mode
Associated Registers		I <sup>2</sup> C Slave Mode (1
Operation		I <sup>2</sup> C Slave Mode (1
Oscillator	137, 139	SEN = 0, AD
Overflow Interrupt		I <sup>2</sup> C Slave Mode (1
Special Event Trigger (CCP)	139	I <sup>2</sup> C Slave Mode (1
TMR3H Register		I <sup>2</sup> C Slave Mode (7
TMR3L Register	137	I <sup>2</sup> C Slave Mode (7
Timing Diagrams		ADMSK = 01
A/D Conversion	397	I <sup>2</sup> C Slave Mode (7
Acknowledge Sequence	230	I <sup>2</sup> C Slave Mode (7
Asynchronous Reception (TXCKP = 0,		I <sup>2</sup> C Slave Mode G
TX Not Inverted)	251	Sequence (7
Asynchronous Transmission (TXCKP = 0,		Low-Voltage Dete
TX Not Inverted)		Master SSP I <sup>2</sup> C B
Asynchronous Transmission, Back to Back		Master SSP I <sup>2</sup> C B
(TXCKP = 0, TX Not Inverted)		PWM Auto-Shutde
Automatic Baud Rate Calculation		Auto-Restart
Auto-Wake-up Bit (WUE) During		PWM Auto-Shutde
Normal Operation	252	Auto-Restart
Auto-Wake-up Bit (WUE) During Sleep		PWM Direction Cl
Baud Rate Generator with Clock Arbitratio		PWM Direction Ch
BRG Overflow Sequence		100% Duty C
BRG Reset Due to SDA Arbitration		PWM Output
During Start Condition	222	Repeated Start Co
		_ `
Brown-out Reset (BOR) Bus Collision During a Repeated Start		Reset, Watchdog Start-up Tim
	004	·
Condition (Case 1)	234	Timer (PWR
Bus Collision During a Repeated Start	004	Send Break Chara
Condition (Case 2)	234	Slave Synchroniza
Bus Collision During a Start	000	Slow Rise Time (N
Condition (SCL = 0)	233	VDD Rise > T
Bus Collision During a Start	000	SPI Mode (Master
Condition (SDA only)	232	SPI Mode (Slave
Bus Collision During a Stop		SPI Mode (Slave
Condition (Case 1)	235	SPP Write Addres
Bus Collision During a Stop		(4 Wait State
Condition (Case 2)	235	SPP Write Addres
Bus Collision for Transmit and		(4 Wait State
Acknowledge	231	SPP Write Addres
Capture/Compare/PWM		(No Wait Sta
(All CCP Modules)		Stop Condition Re
CLKO and I/O		Streaming Paralle
Clock Synchronization	217	Synchronous Rec
		(Master Mod

Clock/Instruction Cycle	61
EUSART Synchronous Receive	
(Master/Slave)	394
EUSART Synchronous Transmission	00.4
(Master/Slave)	394
Example SPI Master Mode (CKE = 0) Example SPI Master Mode (CKE = 1)	
Example SPI Slave Mode (CKE = 0)	
Example SPI Slave Mode (CKE = 1) External Clock (All Modes Except PLL)	
Fail-Safe Clock Monitor	
First Start Bit Timing	
Full-Bridge PWM Output	
Half-Bridge PWM Output	
High/Low-Voltage Detect Characteristics	
High-Voltage Detect (VDIRMAG = 1)	
I <sup>2</sup> C Bus Data	
I <sup>2</sup> C Bus Start/Stop Bits	
I <sup>2</sup> C Master Mode (7 or 10-Bit Transmission)	
I <sup>2</sup> C Master Mode (7-Bit Reception)	229
I <sup>2</sup> C Slave Mode (10-Bit Reception, SEN = 0)	213
I <sup>2</sup> C Slave Mode (10-Bit Reception,	
SEN = 0, ADMSK 01001)	214
I <sup>2</sup> C Slave Mode (10-Bit Reception, SEN = 1)	
I <sup>2</sup> C Slave Mode (10-Bit Transmission)	
I <sup>2</sup> C Slave Mode (7-Bit Reception, SEN = 0)	
$I^2C$ Slave Mode (7-bit Reception, SEN = 0,	
ADMSK = 01011)	211
I <sup>2</sup> C Slave Mode (7-Bit Reception, SEN = 1)	218
I <sup>2</sup> C Slave Mode (7-Bit Transmission)	212
I <sup>2</sup> C Slave Mode General Call Address	
Sequence (7 or 10-Bit Address Mode)	220
Low-Voltage Detect (VDIRMAG = 0)	281
Master SSP I <sup>2</sup> C Bus Data	392
Master SSP I <sup>2</sup> C Bus Start/Stop Bits	392
PWM Auto-Shutdown (PRSEN = 0,	
Auto-Restart Disabled)	160
PWM Auto-Shutdown (PRSEN = 1,	400
Auto-Restart Enabled)	
PWM Direction Change	157
PWM Direction Change at Near 100% Duty Cycle	157
PWM Output	
Repeated Start Condition	
Reset, Watchdog Timer (WDT), Oscillator	220
Start-up Timer (OST) and Power-up	
Timer (PWRT)	383
Send Break Character Sequence	
Slave Synchronization	
Slow Rise Time (MCLR Tied to VDD,	
VDD Rise > TPWRT)	49
SPI Mode (Master Mode)	
SPI Mode (Slave Mode with CKE = 0)	
SPI Mode (Slave Mode with CKE = 1)	
SPP Write Address and Data for USB	
(4 Wait States)	189
SPP Write Address and Read Data for USB	
(4 Wait States)	189
SPP Write Address, Write and Read Data	
(No Wait States)	
Stop Condition Receive or Transmit Mode	
Streaming Parallel Port (PIC18F4455/4550)	396
Synchronous Reception	_
(Master Mode, SREN)	256

Synchronous Transmission254
Synchronous Transmission (Through TXEN)255
Time-out Sequence on POR w/PLL Enabled
(MCLR Tied to VDD)49
Time-out Sequence on Power-up
(MCLR Not Tied to VDD), Case 148
Time-out Sequence on Power-up
(MCLR Not Tied to VDD), Case 248
Time-out Sequence on Power-up
(MCLR Tied to VDD, VDD Rise TPWRT)48
Timer0 and Timer1 External Clock384
Transition for Entry to Idle Mode40
Transition for Entry to SEC_RUN Mode37
Transition for Entry to Sleep Mode39
Transition for Two-Speed Start-up
(INTOSC to HSPLL)299
Transition for Wake from Idle to Run Mode40
Transition for Wake from Sleep (HSPLL)39
Transition From RC_RUN Mode to
PRI_RUN Mode38
Transition from SEC_RUN Mode to
PRI_RUN Mode (HSPLL)37
Transition to RC_RUN Mode
USB Signal395
Timing Diagrams and Specifications
Capture/Compare/PWM Requirements
(All CCP Modules)
CLKO and I/O Requirements
EUSART Synchronous Receive
Requirements
EUSART Synchronous Transmission
Requirements
Example SPI Mode Requirements
(Master Mode, CKE = 0)386
Example SPI Mode Requirements
(Master Mode, CKE = 1)387
Example SPI Mode Requirements
(Slave Mode, CKE = 0)
Example SPI Mode Requirements
(Slave Mode, CKE = 1)389
External Clock Requirements380
I <sup>2</sup> C Bus Data Requirements (Slave Mode)391
I <sup>2</sup> C Bus Start/Stop Bits Requirements390
Master SSP I <sup>2</sup> C Bus Data Requirements
Master SSP I <sup>2</sup> C Bus Start/Stop Bits
Requirements392
PLL Clock
Reset, Watchdog Timer, Oscillator Start-up
Timer, Power-up Timer and Brown-out
Reset Requirements383
Streaming Parallel Port Requirements
(PIC18F4455/4550)396
Timer0 and Timer1 External Clock
Requirements
USB Full-Speed Requirements395
USB Low-Speed Requirements395
Top-of-Stack Access
TQFP Packages and Special Features
TSTFSZ
Two-Speed Start-up
Two-Word Instructions
Example Cases
TXSTA Register
BRGH Bit241

### U

Universal Serial Bus	63
Address Register (UADDR)	
and Streaming Parallel Port	
Associated Registers	
Buffer Descriptor Table	171
Buffer Descriptors	
Address Validation	174
Assignment in Different	
Buffering Modes	176
BDnSTAT Register (CPU Mode)	172
BDnSTAT Register (SIE Mode)	174
Byte Count	
Example	
Memory Map	175
Ownership	
Ping-Pong Buffering	
Register Summary	176
Status and Configuration	171
Class Specifications and Drivers	
Descriptors	
Endpoint Control	
Enumeration	
External Pull-up Resistors	167
External Transceiver	
Eye Pattern Test Enable	
Firmware and Drivers	184
Frame Number Registers	
Frames	
Internal Pull-up Resistors	167
Internal Transceiver	165
Internal Voltage Regulator	167
Interrupts	177
and USB Transactions	177
Layered Framework	
Oscillator Requirements	184
Output Enable Monitor	167
Overview	163, 185
Ping-Pong Buffer Configuration	167
Power	185
Power Modes	183
Bus Power Only	183
Dual Power with Self-Power	
Dominance	183
Self-Power Only	183
RAM	
Memory Map	170
Speed	
Status and Control	
Transfer Types	185
UFRMH:UFRML Registers	170
USB. See Universal Serial Bus.	

V	
Voltage Reference Specifications	375
W	
Watchdog Timer (WDT)	285, 297
Associated Registers	298
Control Register	297
During Oscillator Failure	300
Programming Considerations	
WCOL	225, 226, 227, 230
WCOL Status Flag	225, 226, 227, 230
WWW Address	
WWW, On-Line Support	5
X	
XORLW	347
XORWF	348

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

### **READER RESPONSE**

то.

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.

RE		Reader Response	Total Pages Sent					
Fro	om:							
Apı	plica	cation (optional):						
Wo	ould	d you like a reply?YN						
De	vice	e: PIC18F2455/2550/4455/4550 Lit	erature Number: DS39632C					
Qu	estic	tions:						
1.	Wh	/hat are the best features of this documen	t?					
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.	Ho	ow would you improve this document?						

### PIC18F2455/2550/4455/4550 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.	<u>x</u> T	/XX XXX	Examples	S:
Device	Temperature Range	Package Pattern		BLF4550-I/P 301 = Industrial temp., PDIP age, Extended VDD limits, QTP pattern
Device	PIC18F2455/25 VDD range 4.2 PIC18LF2455/25	550 <sup>(1)</sup> , PIC18LF4455/4550 <sup>(1)</sup> , 550T <sup>(2)</sup> , PIC18LF4455/4550T <sup>(2)</sup> ;	c) PIC18	ge, Extended VDD limits.  BF4455-I/P = Industrial temp., PDIP age, normal VDD limits.
Temperature Range		C to +85°C (Industrial) C to +125°C (Extended)		
Package	SO = SOI	nny Plastic DIP P		F = Standard Voltage Range LF = Wide Voltage Range T = in tape and reel TQFP packages only.
Pattern	QTP, SQTP, Coo (blank otherwise	de or Special Requirements )		

© 2006 Microchip Technology Inc. **Preliminary** DS39632C-page 427



### 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

Alpharetta, GA Tel: 770-640-0034 Fax: 770-640-0307

**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

San Jose

Mountain View, CA Tel: 650-215-1444 Fax: 650-961-0286

**Toronto** 

Mississauga, Ontario,

Canada

Tel: 905-673-0699 Fax: 905-673-6509

#### ASIA/PACIFIC

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: 86-28-8676-6200 Fax: 86-28-8676-6599

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: 91-80-2229-0061 Fax: 91-80-2229-0062

India - New Delhi

Tel: 91-11-5160-8631 Fax: 91-11-5160-8632

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: 43-7242-2244-399 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

10/31/05

### **Mouser Electronics**

**Authorized Distributor** 

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

### Microchip:

PIC18F4455-I/P PIC18LF4550T-I/PT PIC18F2550T-I/SO PIC18LF4550T-I/ML PIC18F4455T-I/ML PIC18LF4455T-I/ML PIC18LF4455T-I/ML PIC18LF4550T-I/PT PIC18LF4550-I/PT PIC18LF4550-I/PT PIC18LF4550-I/PT PIC18F4550-I/PT PIC18F4550-I/PT PIC18F4550-I/PT PIC18F4550-I/PD PIC18F4550-I/PD PIC18F4550-I/PD PIC18F4550-I/PD PIC18F4550-I/PD PIC18F4550-I/PD PIC18F4550-I/PD PIC18F4550-I/PD PIC18F4550-I/PD PIC18F4550-I/PD PIC18F4550-I/PD PIC18F4550-I/PD PIC18F4550-I/PD PIC18F4550-I/PD PIC18F4455T-I/PD PIC18F4550-I/PD 