



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contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, 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, 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, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICKit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, rLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

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

© 2004, 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, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.



MICROCHIP

PIC18F2455/2550/4455/4550

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-board 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 μ A typical
- Sleep mode currents down to 0.1 μ A typical
- Timer1 oscillator: 1.1 μ A typical, 32 kHz, 2V
- Watchdog Timer: 2.1 μ A 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 6.25 ns ($T_{CY}/16$)
 - Compare is 16-bit, max resolution 100 ns (T_{CY})
 - 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 I²C™ Master and Slave Modes
- 10-bit, up to 13-channels Analog-to-Digital Converter module (A/D) with programmable acquisition time
- Dual analog comparators with input multiplexing

Special Microcontroller Features:

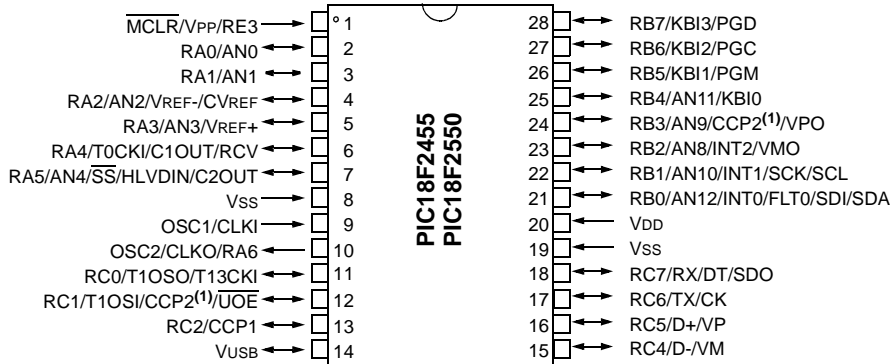
- C compiler optimized architecture with optional extended instruction set
- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Flash/Data EEPROM Retention: > 40 years
- Self-programmable under software control
- Priority levels for interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
 - Programmable period from 41 ms to 131s
 - 2% stability over VDD and temperature
- 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)

| Device | Program Memory | | Data Memory | | I/O | 10-bit A/D (ch) | CCP/ECCP (PWM) | SPP | MSSP | | EAUSART | Comparators | Timers 8/16-bit |
|------------|----------------|----------------------------|--------------|----------------|-----|-----------------|----------------|-----|------|-------------------------|---------|-------------|-----------------|
| | FLASH (bytes) | # Single-Word Instructions | SRAM (bytes) | EEPROM (bytes) | | | | | SPI | Master I ² C | | | |
| PIC18F2455 | 24 K | 12288 | 2048 | 256 | 24 | 10 | 2/0 | No | Y | Y | 1 | 2 | 1/3 |
| PIC18F2550 | 32 K | 16384 | 2048 | 256 | 24 | 10 | 2/0 | No | Y | Y | 1 | 2 | 1/3 |
| PIC18F4455 | 24 K | 12288 | 2048 | 256 | 35 | 13 | 1/1 | Yes | Y | Y | 1 | 2 | 1/3 |
| PIC18F4550 | 32 K | 16384 | 2048 | 256 | 35 | 13 | 1/1 | Yes | Y | Y | 1 | 2 | 1/3 |

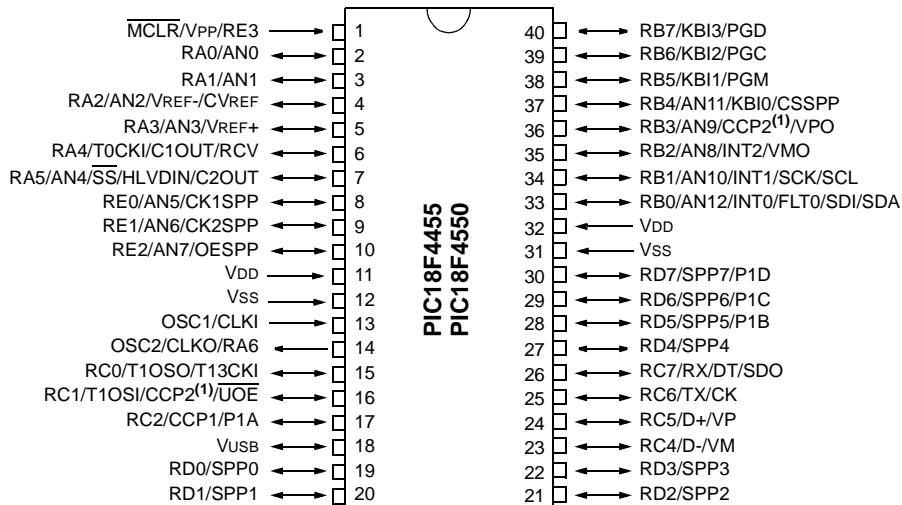
PIC18F2455/2550/4455/4550

Pin Diagrams

28-Pin PDIP, SOIC



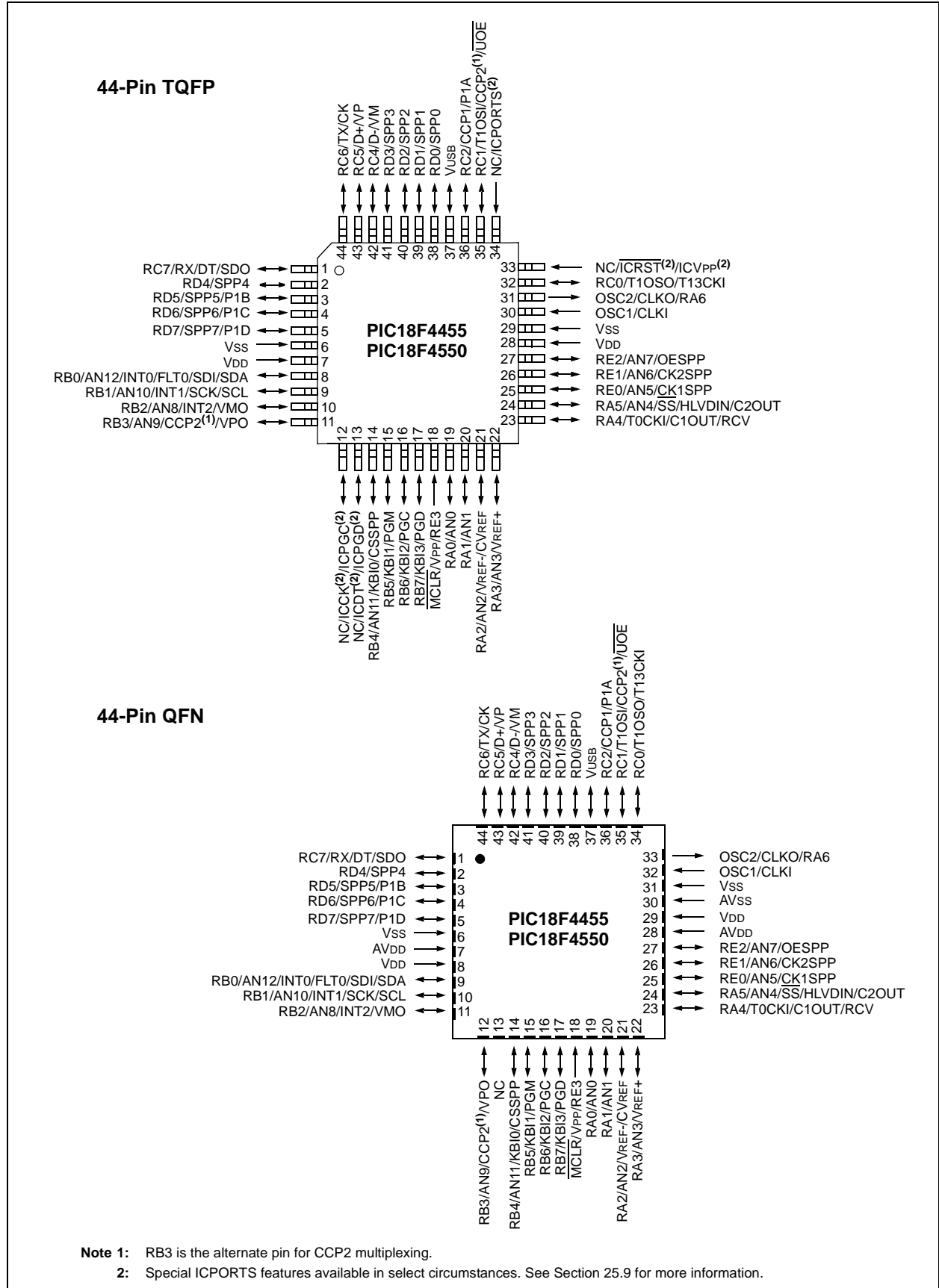
40-Pin PDIP



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

PIC18F2455/2550/4455/4550

Pin Diagrams (Continued)



PIC18F2455/2550/4455/4550

Table of Contents

| | | |
|------|--|-----|
| 1.0 | Device Overview | 7 |
| 2.0 | Oscillator Configurations | 23 |
| 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 |
| 17.0 | Universal Serial Bus (USB) | 163 |
| 18.0 | Streaming Parallel Port | 187 |
| 19.0 | Master Synchronous Serial Port (MSSP) Module | 193 |
| 20.0 | Enhanced Universal Synchronous Receiver Transmitter (EUSART) | 233 |
| 21.0 | 10-Bit Analog-to-Digital Converter (A/D) Module | 253 |
| 22.0 | Comparator Module | 263 |
| 23.0 | Comparator Voltage Reference Module | 269 |
| 24.0 | High/Low-Voltage Detect (HLVD) | 273 |
| 25.0 | Special Features of the CPU | 279 |
| 26.0 | Instruction Set Summary | 301 |
| 27.0 | Development Support | 351 |
| 28.0 | Electrical Characteristics | 357 |
| 29.0 | DC and AC Characteristics Graphs and Tables | 395 |
| 30.0 | Packaging Information | 397 |
| | Appendix A: Revision History | 405 |
| | Appendix B: Device Differences | 405 |
| | Appendix C: Conversion Considerations | 406 |
| | Appendix D: Migration from Baseline to Enhanced Devices | 406 |
| | Appendix E: Migration from Mid-Range to Enhanced Devices | 407 |
| | Appendix F: Migration from High-End to Enhanced Devices | 407 |
| | Index | 409 |
| | On-Line Support | 421 |
| | Systems Information and Upgrade Hot Line | 421 |
| | Reader Response | 422 |
| | PIC18F2455/2550/4455/4550 Product Identification System | 423 |

TO OUR VALUED CUSTOMERS

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

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

Most Current Data Sheet

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

<http://www.microchip.com>

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

Errata

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

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

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

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

Customer Notification System

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

PIC18F2455/2550/4455/4550

NOTES:

PIC18F2455/2550/4455/4550

1.0 DEVICE OVERVIEW

This document contains device specific information for the following devices:

- PIC18F2455
- PIC18F2550
- PIC18F4455
- PIC18F4550
- PIC18LF2455
- PIC18LF2550
- PIC18LF4455
- 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.
- **Lower Consumption in Key Modules:** The power requirements for both Timer1 and the Watchdog Timer have been reduced by up to 80%, with typical values of 1.1 μ A and 2.1 μ A, respectively.

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 ($\pm 2\%$ 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 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.

PIC18F2455/2550/4455/4550

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).
4. CCP and Enhanced CCP implementation (28-pin devices have 2 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 V_{DD} range of 4.2V to 5.5V. Low-voltage parts, designated by “LF” (such as PIC18LF2550), function over an extended V_{DD} range of 2.0V to 5.5V.

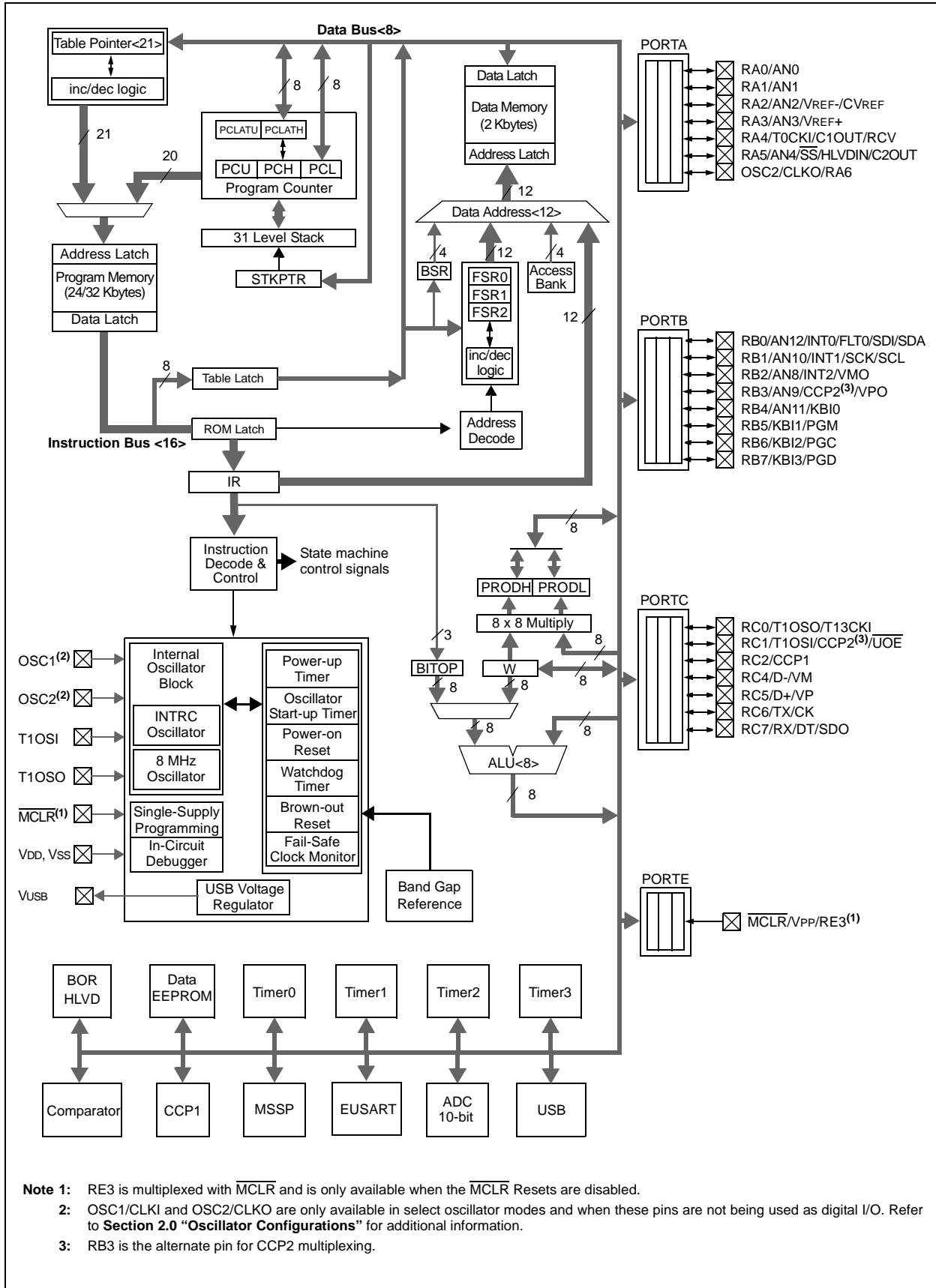
PIC18F2455/2550/4455/4550

TABLE 1-1: DEVICE FEATURES

| Features | PIC18F2455 | PIC18F2550 | PIC18F4455 | PIC18F4550 |
|--|--|--|--|--|
| Operating Frequency | DC – 48 MHz | DC – 48 MHz | DC – 48 MHz | 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 | 1 | 1 | 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 | 75 Instructions; 83 with Extended Instruction Set enabled | 75 Instructions; 83 with Extended Instruction Set enabled | 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 |

PIC18F2455/2550/4455/4550

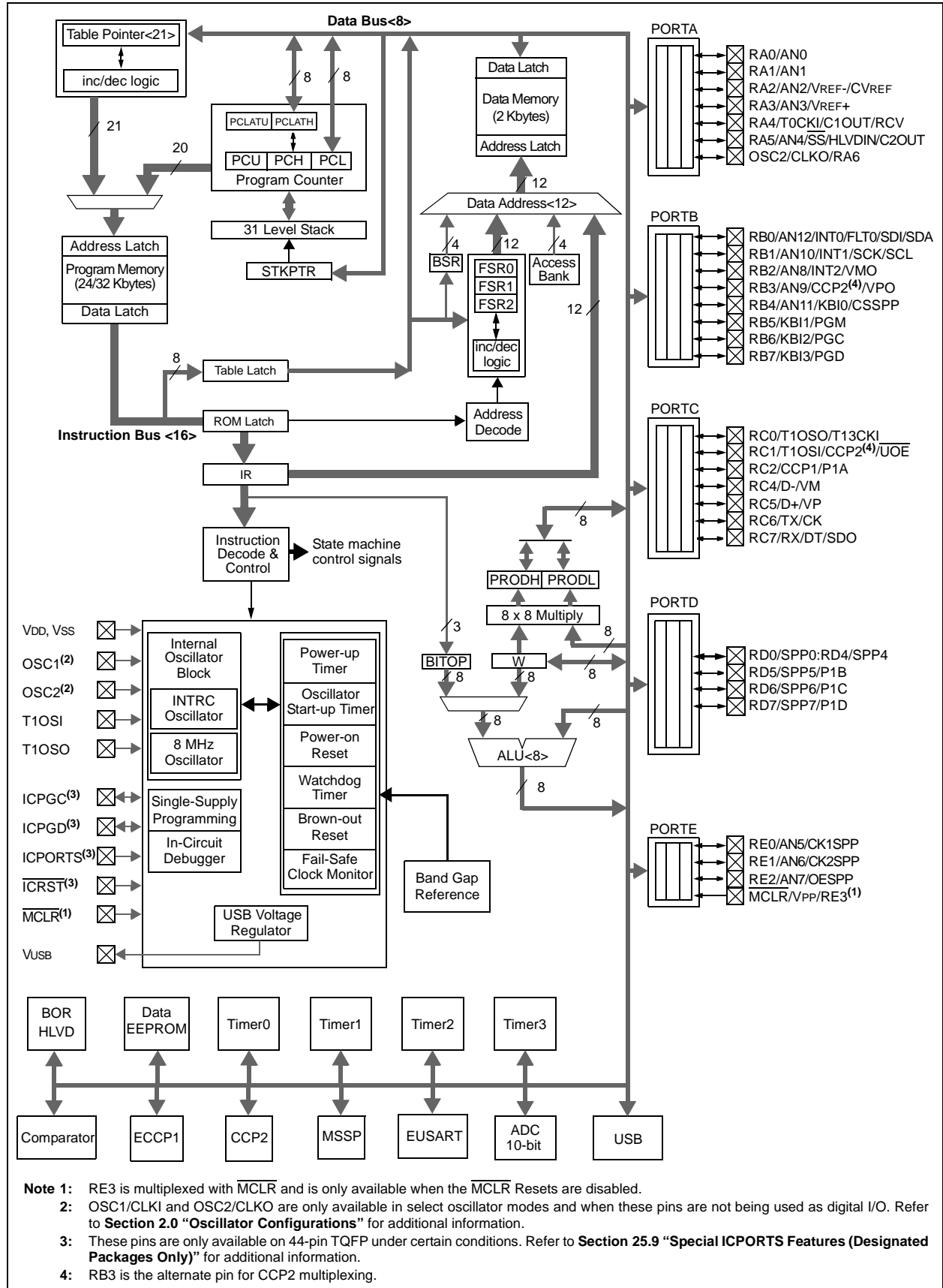
FIGURE 1-1: PIC18F2445/2550 (28-PIN) BLOCK DIAGRAM



- Note 1:** RE3 is multiplexed with $\overline{\text{MCLR}}$ and is only available when the $\overline{\text{MCLR}}$ Resets are disabled.
- Note 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.
- Note 3:** RB3 is the alternate pin for CCP2 multiplexing.

PIC18F2455/2550/4455/4550

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



PIC18F2455/2550/4455/4550

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

| Pin Name | Pin Number | Pin Type | Buffer Type | Description |
|--|------------|-------------------|-------------------|--|
| | PDIP, SOIC | | | |
| MCLR/VPP/RE3 MCLR VPP RE3 | 1 | I P I | ST ST | Master Clear (input) or programming voltage (input). Master Clear (Reset) input. This pin is an active-low Reset to the device. Programming voltage input. Digital input. |
| OSC1/CLKI OSC1 CLKI | 9 | I I | Analog Analog | Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. External clock source input. Always associated with pin function OSC1. (See OSC2/CLKO pins.) |
| OSC2/CLKO/RA6 OSC2 CLKO RA6 | 10 | O O I/O | — — TTL | Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In select modes, OSC2 pin outputs CLKO which has 1/4 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 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.

PIC18F2455/2550/4455/4550

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

| Pin Name | Pin Number | Pin Type | Buffer Type | Description |
|--|------------|-------------------------|-------------------------------------|--|
| | PDIP, SOIC | | | |
| RA0/AN0 RA0 AN0 | 2 | I/O I | TTL Analog | PORTA is a bidirectional I/O port. Digital I/O. Analog input 0. |
| RA1/AN1 RA1 AN1 | 3 | I/O I | TTL Analog | Digital I/O. Analog input 1. |
| RA2/AN2/VREF-/CVREF RA2 AN2 VREF- CVREF | 4 | I/O I I O | TTL 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 | 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 | I/O I O I | TTL 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 | 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. |
| RA6 | — | — | — | See the OSC2/CLKO/RA6 pin. |

Legend: TTL = TTL compatible input
 ST = Schmitt Trigger input with CMOS levels
 O = Output

CMOS = CMOS compatible input or 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.

PIC18F2455/2550/4455/4550

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

| Pin Name | Pin Number | Pin Type | Buffer Type | Description |
|---|------------|----------|-------------|---|
| | PDIP, SOIC | | | |
| PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs. | | | | |
| RB0/AN12/INT0/FLT0/SDI/SDA | 21 | | | |
| RB0 | | I/O | TTL | Digital I/O. |
| AN12 | | I | Analog | Analog input 12. |
| INT0 | | I | ST | External interrupt 0. |
| FLT0 | | I | ST | PWM Fault input (CCP1 module). |
| SDI | | I | ST | SPI data in. |
| SDA | | I/O | ST | I ² C data I/O. |
| RB1/AN10/INT1/SCK/SCL | 22 | | | |
| RB1 | | I/O | TTL | Digital I/O. |
| AN10 | | I | Analog | Analog input 10. |
| INT1 | | I | ST | External interrupt 1. |
| SCK | | I/O | ST | Synchronous serial clock input/output for SPI™ mode. |
| SCL | | I/O | ST | Synchronous serial clock input/output for I ² C™ mode. |
| RB2/AN8/INT2/VMO | 23 | | | |
| RB2 | | I/O | TTL | Digital I/O. |
| AN8 | | I | Analog | Analog input 8. |
| INT2 | | I | ST | External interrupt 2. |
| VMO | | O | — | External USB transceiver VMO output. |
| RB3/AN9/CCP2/VPO | 24 | | | |
| RB3 | | I/O | TTL | Digital I/O. |
| AN9 | | I | TTL | Analog input 9. |
| CCP2 ⁽¹⁾ | | I/O | ST | Capture 2 input/Compare 2 output/PWM 2 output. |
| VPO | | O | — | External USB transceiver VPO output. |
| RB4/AN11/KBI0 | 25 | | | |
| RB4 | | I/O | TTL | Digital I/O. |
| AN11 | | I | Analog | Analog input 11. |
| KBI0 | | I | TTL | Interrupt-on-change pin. |
| RB5/KBI1/PGM | 26 | | | |
| RB5 | | I/O | TTL | Digital I/O. |
| KBI1 | | I | TTL | Interrupt-on-change pin. |
| PGM | | I/O | ST | Low-Voltage ICSP™ Programming enable pin. |
| RB6/KBI2/PGC | 27 | | | |
| RB6 | | I/O | TTL | Digital I/O. |
| KBI2 | | I | 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
ST = Schmitt Trigger input with CMOS levels
O = Output
CMOS = CMOS compatible input or 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.

PIC18F2455/2550/4455/4550

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 1L and 1H 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 Crystal/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
9. 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

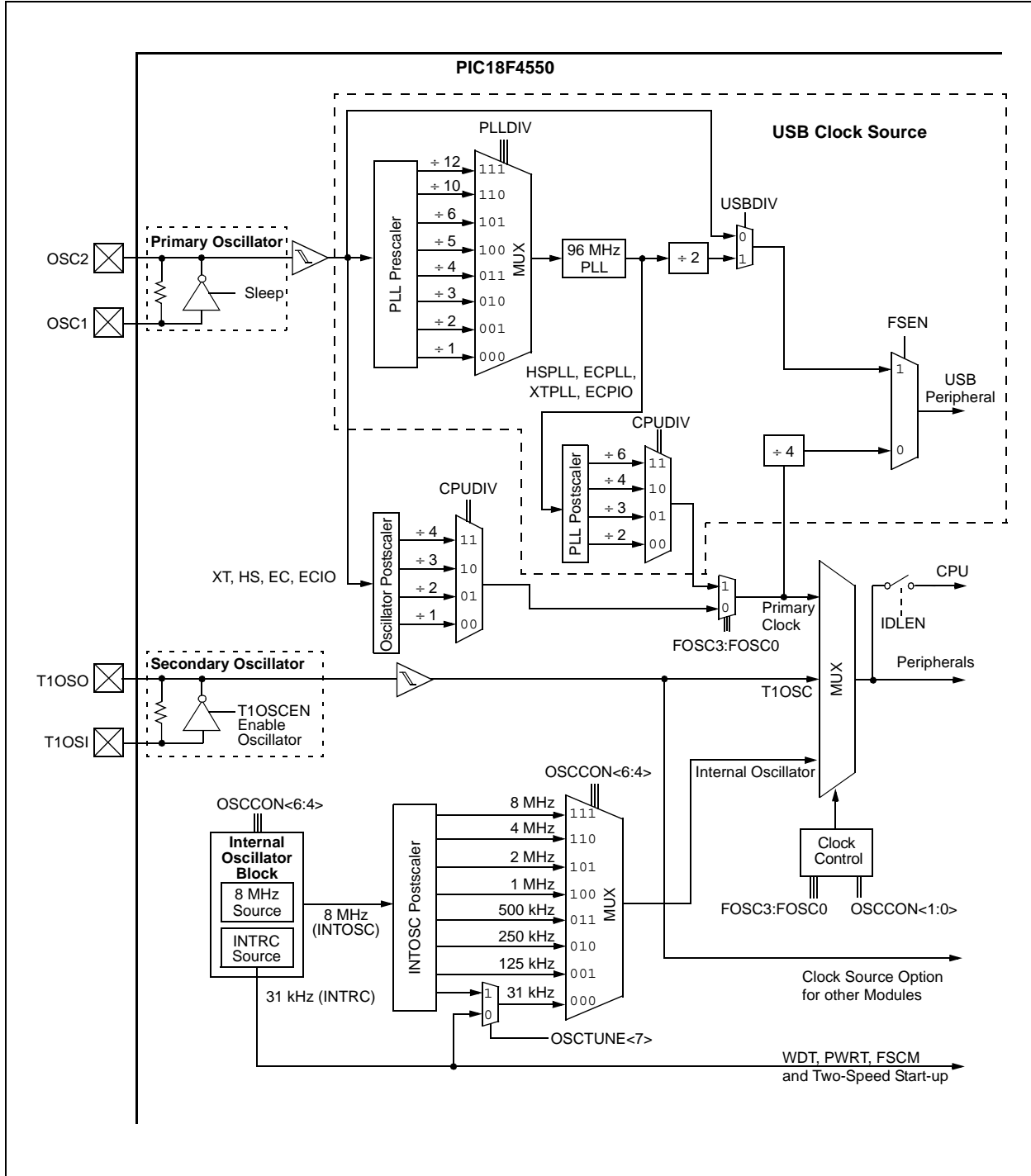
2.2.1 OSCILLATOR MODES AND USB OPERATION

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”**.

PIC18F2455/2550/4455/4550

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)

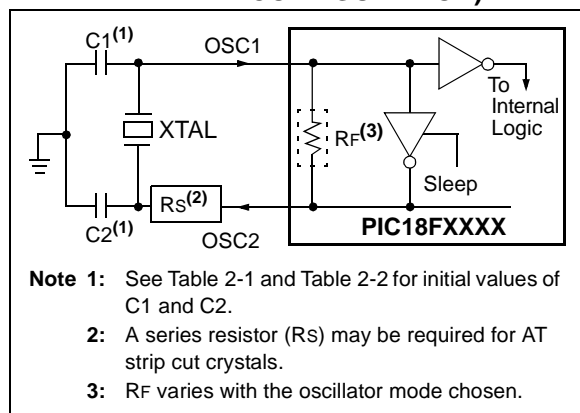


TABLE 2-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

| Typical Capacitor Values Used: | | | |
|--------------------------------|----------|-------|-------|
| Mode | Freq | OSC1 | OSC2 |
| XT | 4.0 MHz | 33 pF | 33 pF |
| HS | 8.0 MHz | 27 pF | 27 pF |
| | 16.0 MHz | 22 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 on page 25 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 Values Tested: | |
|----------|--------------|----------------------------------|-------|
| | | 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 |

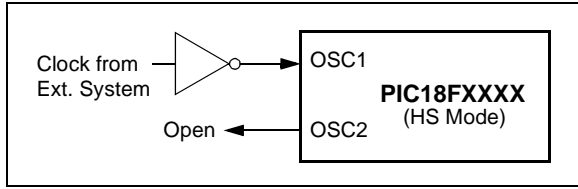
- Note 1:** Higher capacitance increases the stability of oscillator but also increases the start-up time.
- 2:** When operating below 3V VDD, or when using certain ceramic resonators at any voltage, it may be necessary to use the HS mode or switch to a crystal oscillator.
- 3:** Since each resonator/crystal 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 microcontroller is in HS Oscillator mode. In this case, the OSC2/CLKO pin is left open (Figure 2-3).

PIC18F2455/2550/4455/4550

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

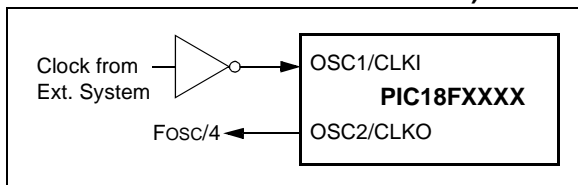


2.2.3 EXTERNAL CLOCK INPUT

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.

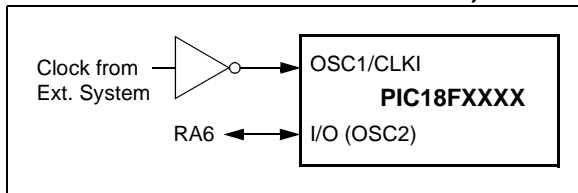
In the EC and ECPLL Oscillator modes, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 2-4 shows the pin connections for the EC Oscillator mode.

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

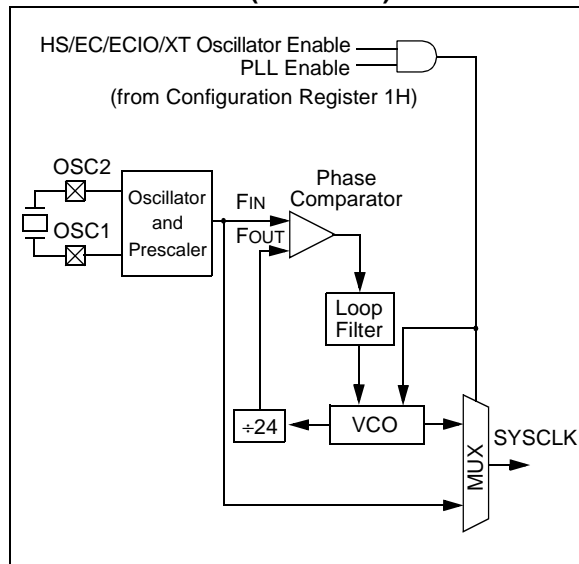
PIC18F2455/2550/4255/4550 devices include a Phase Locked Loop (PLL) circuit. This is provided specifically for USB applications with lower speed oscillators and can also be used as a microcontroller clock source.

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 microcontroller 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 USB clock source is determined by the particular internal oscillator mode.

There are four distinct modes available:

1. INTHS mode: The USB clock is provided by the oscillator in HS mode.
2. INTXT mode: The USB clock is provided by the oscillator in XT mode.
3. INTCKO mode: The USB clock is provided by an external clock input on OSC1/CLKI; the OSC2/CLKO pin outputs $F_{osc}/4$.
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 V_{DD} 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.

PIC18F2455/2550/4455/4550

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 |

- 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

| 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 |

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.

PIC18F2455/2550/4455/4550

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 ⁽¹⁾ | 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.

PIC18F2455/2550/4455/4550

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 |
| | | | +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/T1OSO/T13CKI and RC1/T1OSI pins. Like the XT and HS mode oscillator 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.

PIC18F2455/2550/4455/4550

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 REGISTER

| R/W-0 | R/W-1 | R/W-0 | R/W-0 | R ⁽¹⁾ | R-0 | R/W-0 | R/W-0 |
|-------|-------|-------|-------|------------------|------|-------|-------|
| IDLEN | IRCF2 | IRCF1 | IRCF0 | OSTS | IOFS | SCS1 | SCS0 |
| bit 7 | | | | | | | bit 0 |

- 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⁽³⁾
 - 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⁽¹⁾
 - 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 block
 - 01 = Timer1 oscillator
 - 00 = Primary oscillator
- Note 1:** Depends on 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.

| | | | |
|-------------------|------------------|------------------------------------|--------------------|
| 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 |

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” com-

mand 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., SSP 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 $\overline{\text{MCLR}}$ Reset.

PIC18F2455/2550/4455/4550

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

| Mode | OSCCON Bits | | Module Clocking | | Available Clock and Oscillator Source |
|----------|-------------------------|----------------|-----------------|-------------|--|
| | IDLEN<7> ⁽¹⁾ | SCS1:SCS0<1:0> | CPU | Peripherals | |
| 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 ⁽²⁾ |
| 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 ⁽²⁾ |

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

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

PIC18F2455/2550/4455/4550

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 may result.

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.

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

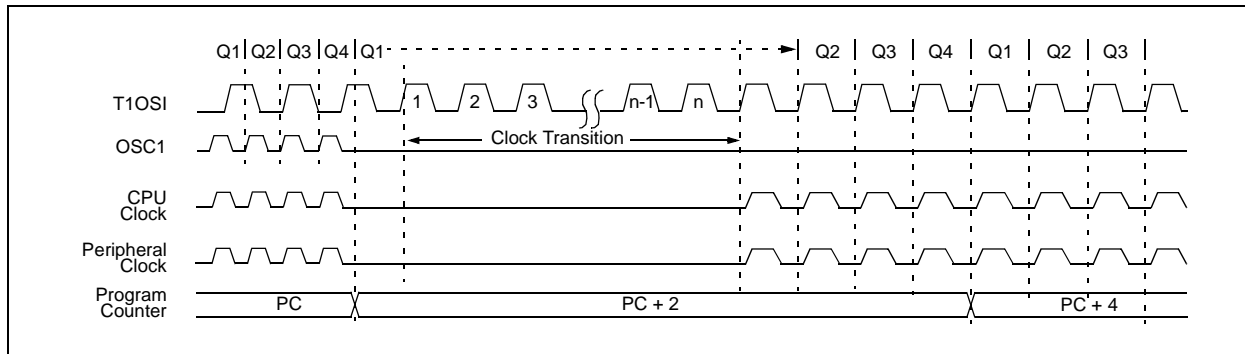
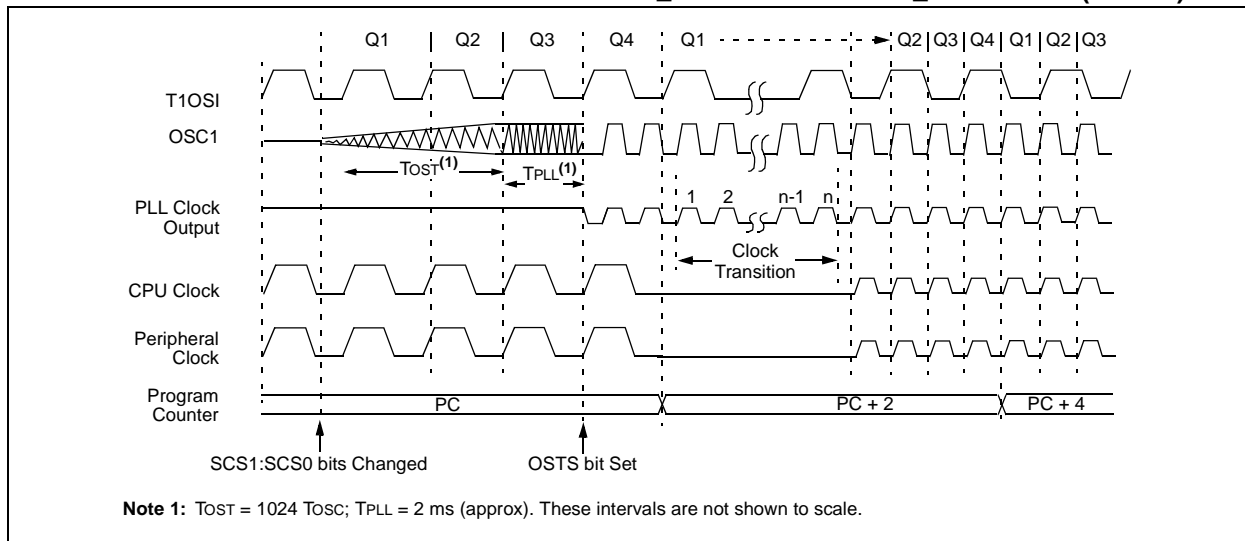


FIGURE 3-2: TRANSITION TIMING FROM SEC_RUN MODE TO PRI_RUN MODE (HSPLL)



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 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.

PIC18F2455/2550/4455/4550

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

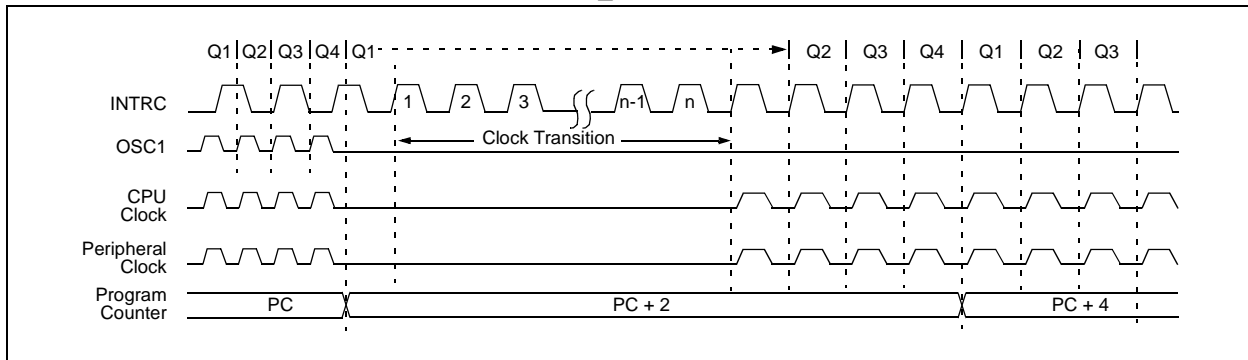
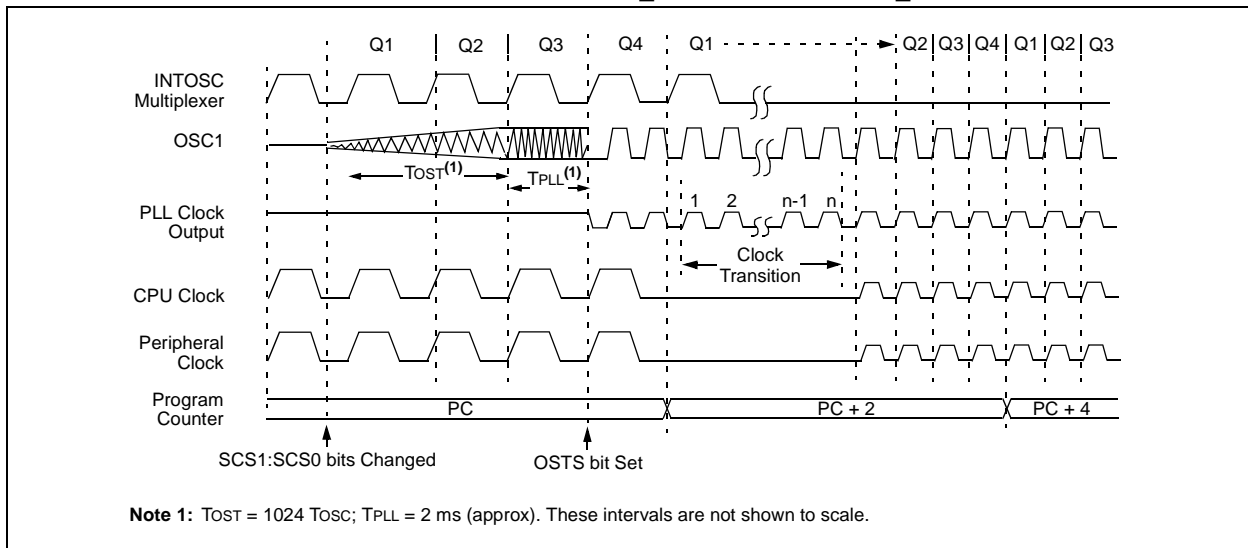


FIGURE 3-4: TRANSITION TIMING FROM RC_RUN MODE TO PRI_RUN MODE



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

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

Since the CPU is not executing instructions, the only exits from any of the Idle modes are by interrupt, WDT time-out or a Reset. When a wake event occurs, CPU execution is delayed by an interval of T_{CSD} (parameter 38, Table 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 the Sleep mode, a WDT time-out will result in a WDT wake-up to the Run mode currently specified by the SCS1:SCS0 bits.

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

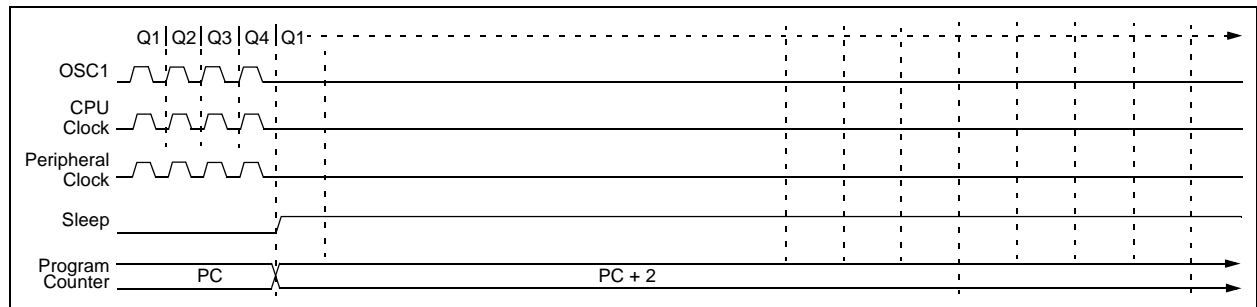
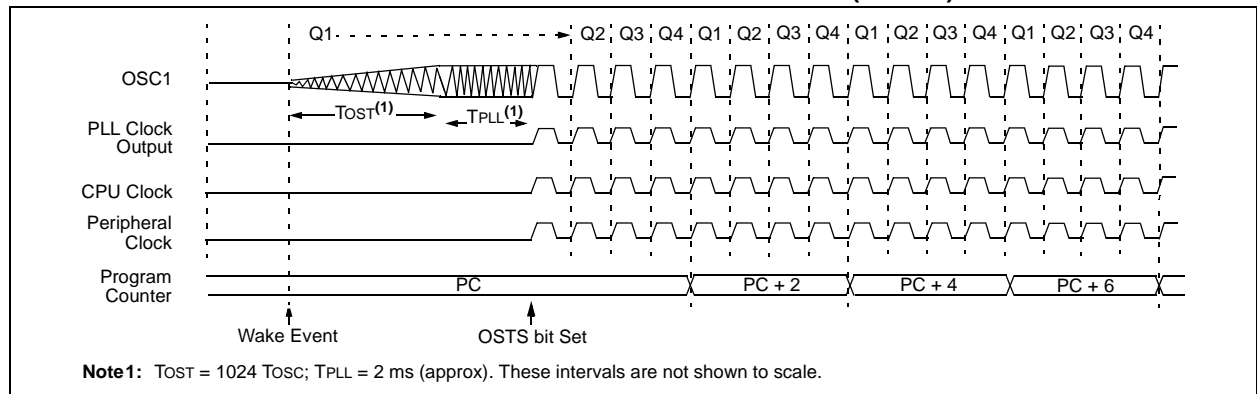


FIGURE 3-6: TRANSITION TIMING FOR WAKE FROM SLEEP (HSPLL)



PIC18F2455/2550/4455/4550

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 T_{CSD} 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 T_{CSD} following the wake event, the CPU begins executing code being clocked by the Timer1 oscillator. The IDLEN and SCS bits are not affected by the wake-up; the Timer1 oscillator continues to run (see Figure 3-8).

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

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

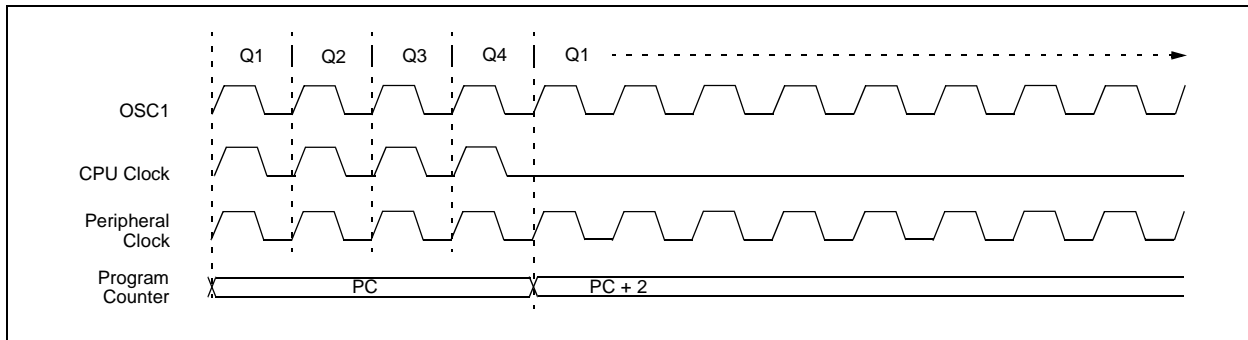
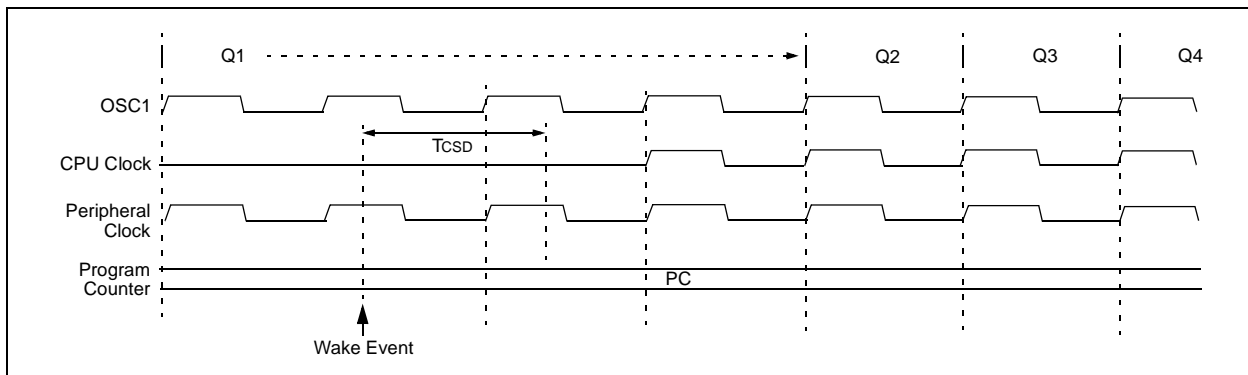


FIGURE 3-8: TRANSITION TIMING FOR WAKE FROM IDLE TO RUN MODE



3.4.3 RC_IDLE MODE

In RC_IDLE mode, the CPU is disabled but the peripherals continue to be clocked from the internal oscillator block 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.

PIC18F2455/2550/4455/4550

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)

| Microcontroller Clock Source | | Exit Delay | Clock Ready Status Bit (OSCCON) |
|--------------------------------------|-----------------------|---------------------------------------|---------------------------------|
| Before Wake-up | After Wake-up | | |
| Primary Device Clock (PRI_IDLE mode) | XT, HS | None | OSTS |
| | XTPLL, HSPLL | | |
| | EC | | IOFS |
| | INTOSC ⁽³⁾ | | |
| T1OSC or INTRC ⁽¹⁾ | XT, HS | TOST ⁽⁴⁾ | OSTS |
| | XTPLL, HSPLL | TOST + t _{rc} ⁽⁴⁾ | |
| | EC | TcSD ⁽²⁾ | IOFS |
| | INTOSC ⁽²⁾ | TIOBST ⁽⁵⁾ | |
| INTOSC ⁽³⁾ | XT, HS | TOST ⁽⁵⁾ | OSTS |
| | XTPLL, HSPLL | TOST + t _{rc} ⁽⁴⁾ | |
| | EC | TcSD ⁽²⁾ | IOFS |
| | INTOSC ⁽²⁾ | None | |
| None (Sleep mode) | XT, HS | TOST ⁽⁴⁾ | OSTS |
| | XTPLL, HSPLL | TOST + t _{rc} ⁽⁴⁾ | |
| | EC | TcSD ⁽²⁾ | IOFS |
| | INTOSC ⁽²⁾ | TIOBST ⁽⁵⁾ | |

Note 1: In this instance, refers specifically to the 31 kHz INTRC clock source.

2: TcSD (parameter 38) 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 (parameter 32). t_{rc} is the PLL Lock-out Timer (parameter F12); it is also designated as TPLL.

5: Execution continues during TIOBST (parameter 39), the INTOSC stabilization period.

PIC18F2455/2550/4455/4550

4.0 RESET

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

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ Reset during normal operation
- $\overline{\text{MCLR}}$ Reset during power managed modes
- Watchdog Timer (WDT) Reset (during execution)
- Programmable Brown-out Reset (BOR)
- RESET Instruction
- Stack Full Reset
- Stack Underflow Reset

This section discusses Resets generated by $\overline{\text{MCLR}}$, 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)”**.

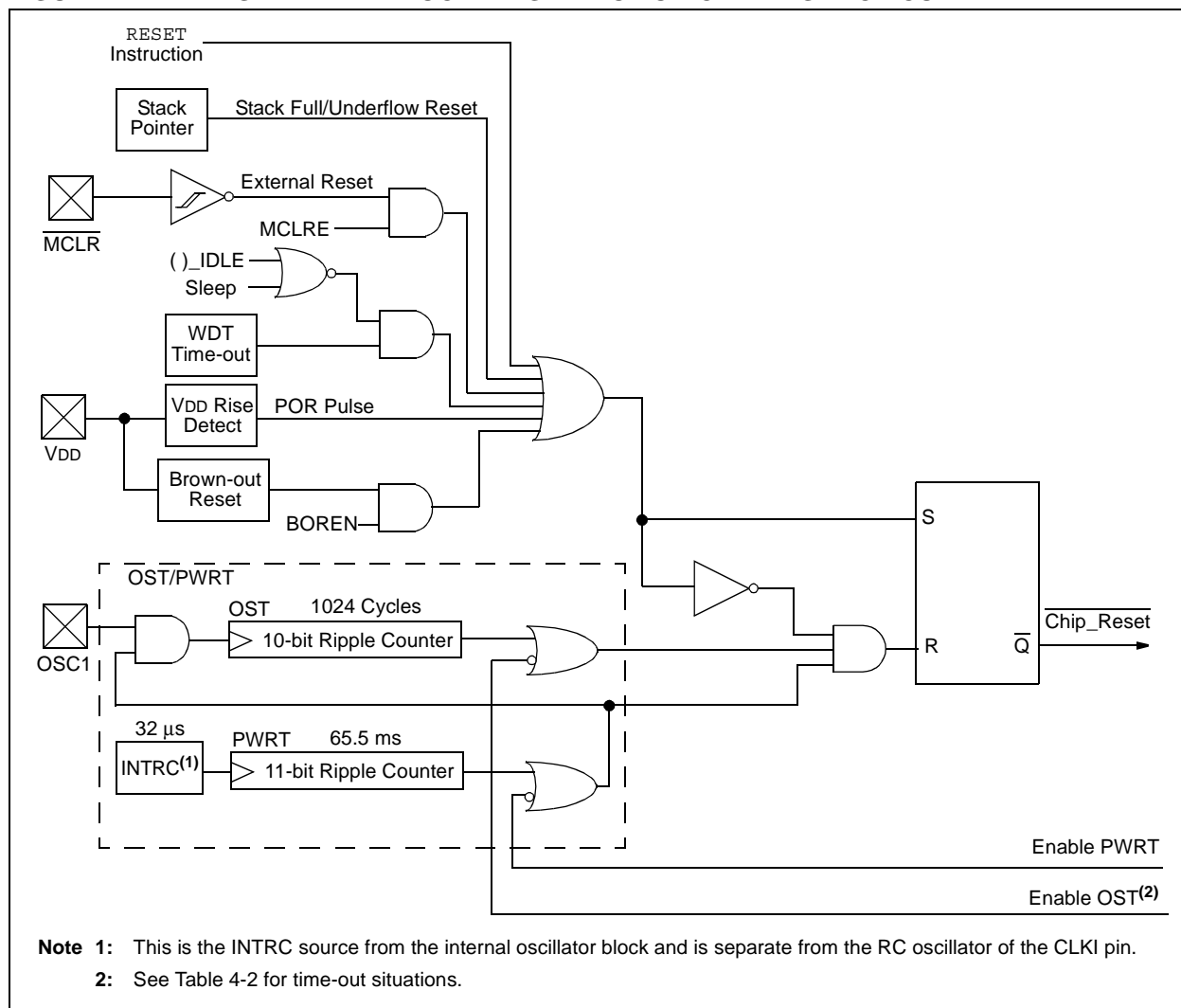
A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 4-1.

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



PIC18F2455/2550/4455/4550

REGISTER 4-1: RCON REGISTER

| R/W-0 | R/W-1 ⁽¹⁾ | U-0 | R/W-1 | R-1 | R-1 | R/W-0 ⁽²⁾ | R/W-0 | |
|-------|----------------------|-----|-----------------|-----------------|-----------------|----------------------|------------------|-------|
| IPEN | SBOREN | — | \overline{RI} | \overline{TO} | \overline{PD} | \overline{POR} | \overline{BOR} | |
| bit 7 | | | | | | | | bit 0 |

- 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⁽¹⁾
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 **\overline{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 **\overline{TO} :** Watchdog Time-out Flag bit
 1 = Set by power-up, CLRWDT instruction or SLEEP instruction
 0 = A WDT time-out occurred
- bit 2 **\overline{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 **\overline{POR} :** Power-on Reset Status bit⁽²⁾
 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 **\overline{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 \overline{POR} is determined by the type of device Reset. See the notes following this table and **Section 4.6 "Reset State of Registers"** for additional information.

Legend:

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

Note 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{BOR} is '0' and \overline{POR} is '1' (assuming that \overline{POR} was set to '1' by software immediately after POR).

4.2 Master Clear Reset ($\overline{\text{MCLR}}$)

The $\overline{\text{MCLR}}$ 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 $\overline{\text{MCLR}}$ 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 $\overline{\text{MCLR}}$ input can be disabled with the MCLRE configuration bit. When $\overline{\text{MCLR}}$ 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)

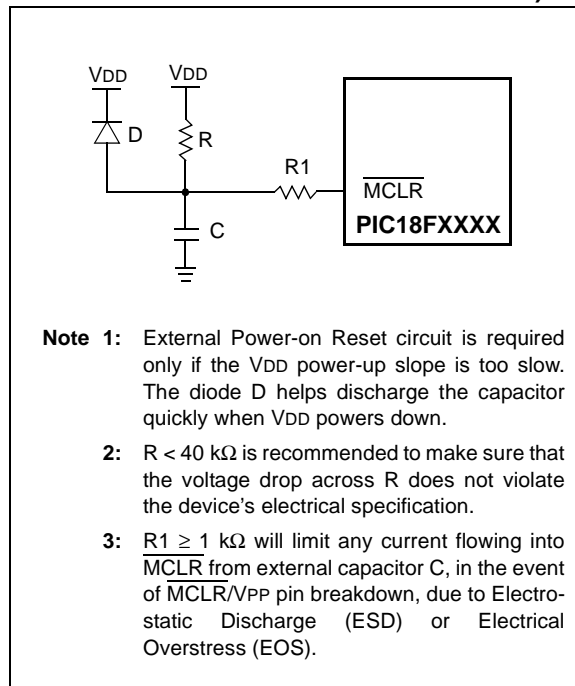
A Power-on Reset pulse is generated on-chip whenever VDD rises above a certain threshold. This allows the device to start in the initialized state when VDD is adequate for operation.

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

When the device starts normal operation (i.e., exits the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

POR events are captured by the $\overline{\text{POR}}$ 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. $\overline{\text{POR}}$ 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)



PIC18F2455/2550/4455/4550

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) for greater than TBOR (parameter 35) will reset the device. A Reset may or may not occur if VDD falls below VBOR for less than TBOR. The chip will remain in Brown-out Reset until VDD rises above VBOR.

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). 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 Configuration | | Status of SBOREN (RCON<6>) | BOR Operation |
|-------------------|--------|----------------------------|--|
| BOREN1 | BOREN0 | | |
| 0 | 0 | Unavailable | BOR disabled; must be enabled by reprogramming the configuration bits. |
| 0 | 1 | Available | 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. |

PIC18F2455/2550/4455/4550

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 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 $2048 \times 32 \mu\text{s} = 65.6 \text{ ms}$. While the PWRT is counting, the device is held in Reset.

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

The PWRT is enabled by clearing the $\overline{\text{PWRTEN}}$ 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). This ensures that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP, 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:

1. After the POR pulse 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 or LP modes. 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 $\overline{\text{MCLR}}$ is kept low long enough, all time-outs will expire. Bringing $\overline{\text{MCLR}}$ 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 Configuration | Power-up ⁽²⁾ and Brown-out | | Exit from Power Managed Mode |
|--------------------------|--|--|--|
| | $\overline{\text{PWRTEN}} = 0$ | $\overline{\text{PWRTEN}} = 1$ | |
| HS, XT | $66 \text{ ms}^{(1)} + 1024 \text{ TOSC}$ | 1024 TOSC | 1024 TOSC |
| HSPLL, XTPLL | $66 \text{ ms}^{(1)} + 1024 \text{ TOSC} + 2 \text{ ms}^{(2)}$ | $1024 \text{ TOSC} + 2 \text{ ms}^{(2)}$ | $1024 \text{ TOSC} + 2 \text{ ms}^{(2)}$ |
| EC, ECIO | $66 \text{ ms}^{(1)}$ | — | — |
| ECPLL, ECPIO | $66 \text{ ms}^{(1)} + 2 \text{ ms}^{(2)}$ | $2 \text{ ms}^{(2)}$ | $2 \text{ ms}^{(2)}$ |
| INTIO, INTCKO | $66 \text{ ms}^{(1)}$ | — | — |
| INTHS, INTXL | $66 \text{ ms}^{(1)} + 1024 \text{ TOSC}$ | 1024 TOSC | 1024 TOSC |

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

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

PIC18F2455/2550/4455/4550

FIGURE 4-3: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ TIED TO V_{DD} , V_{DD} RISE < T_{PWRT})

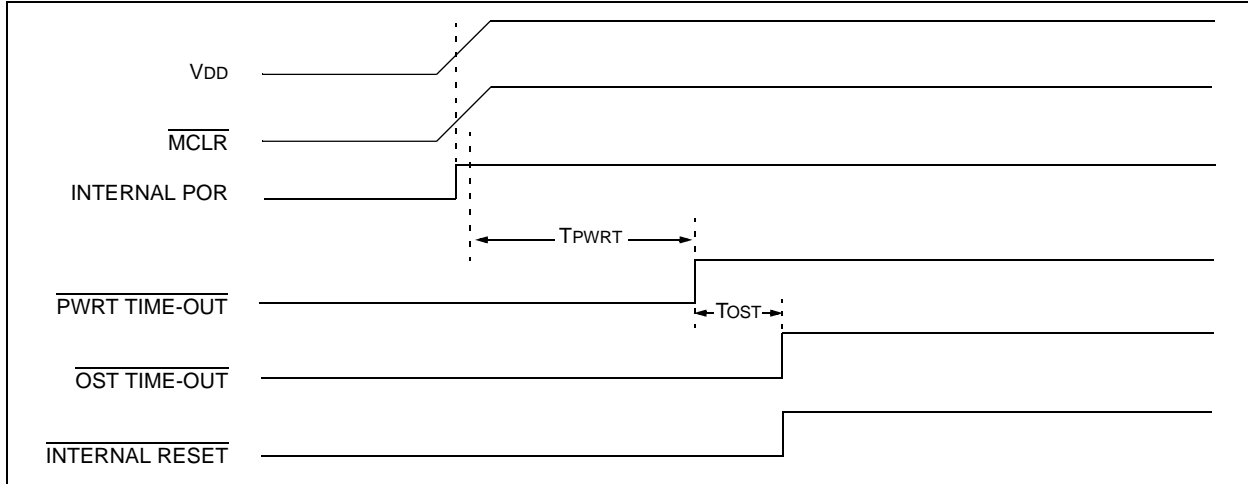


FIGURE 4-4: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD}): CASE 1

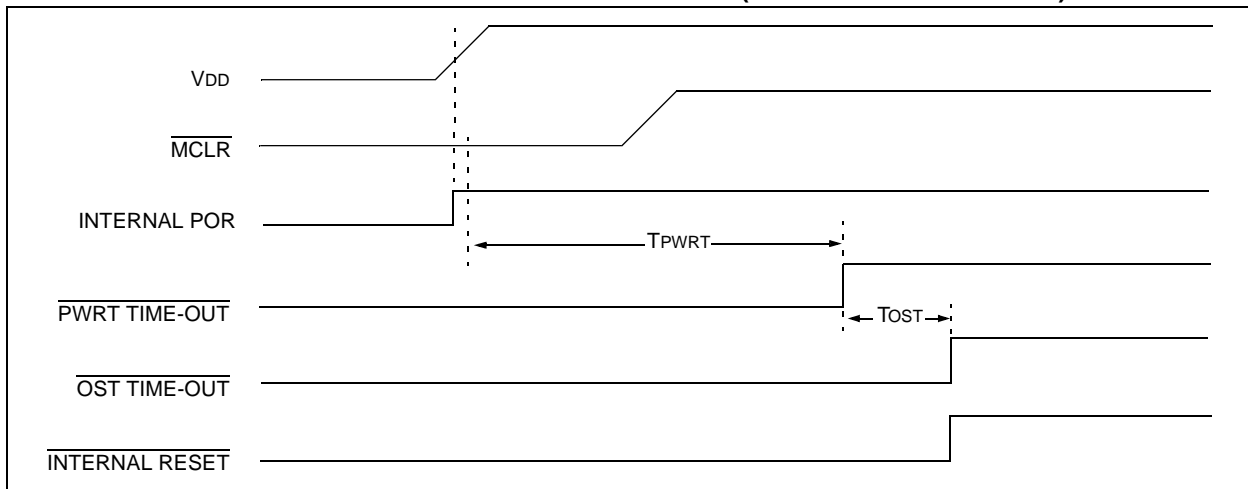
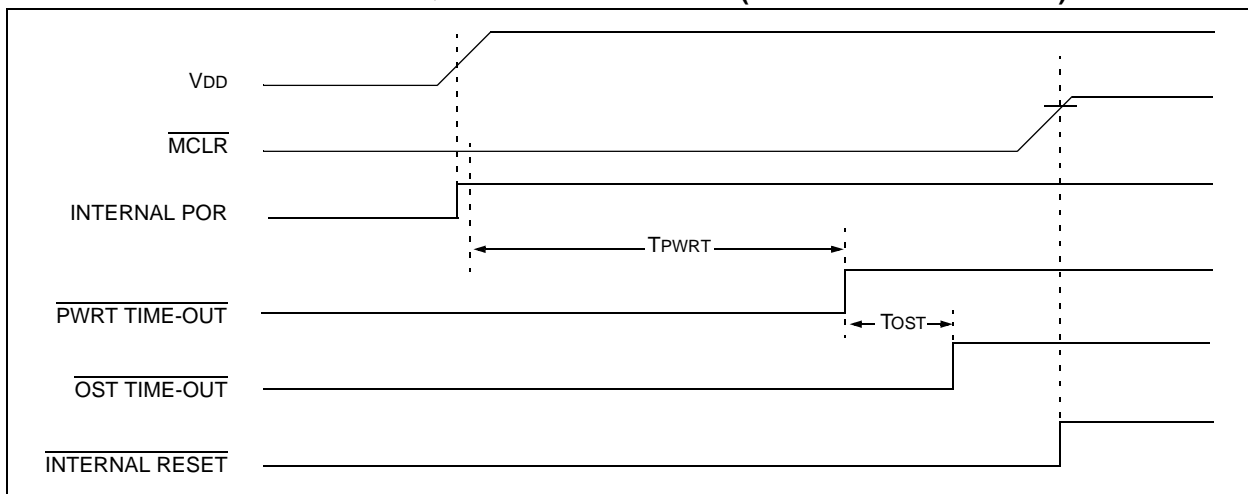


FIGURE 4-5: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD}): CASE 2



PIC18F2455/2550/4455/4550

FIGURE 4-6: SLOW RISE TIME ($\overline{\text{MCLR}}$ TIED TO V_{DD} , V_{DD} RISE $>$ T_{PWRT})

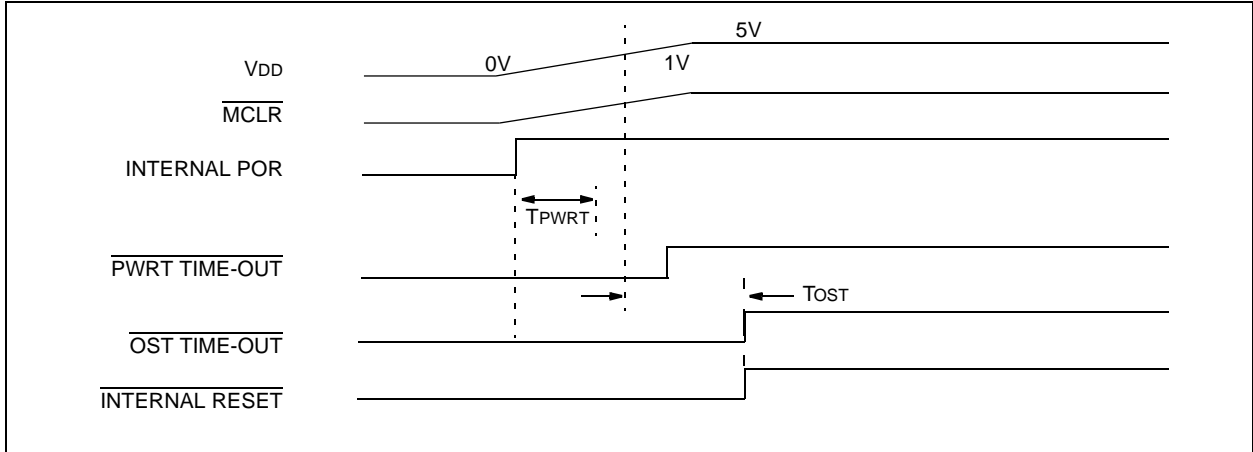
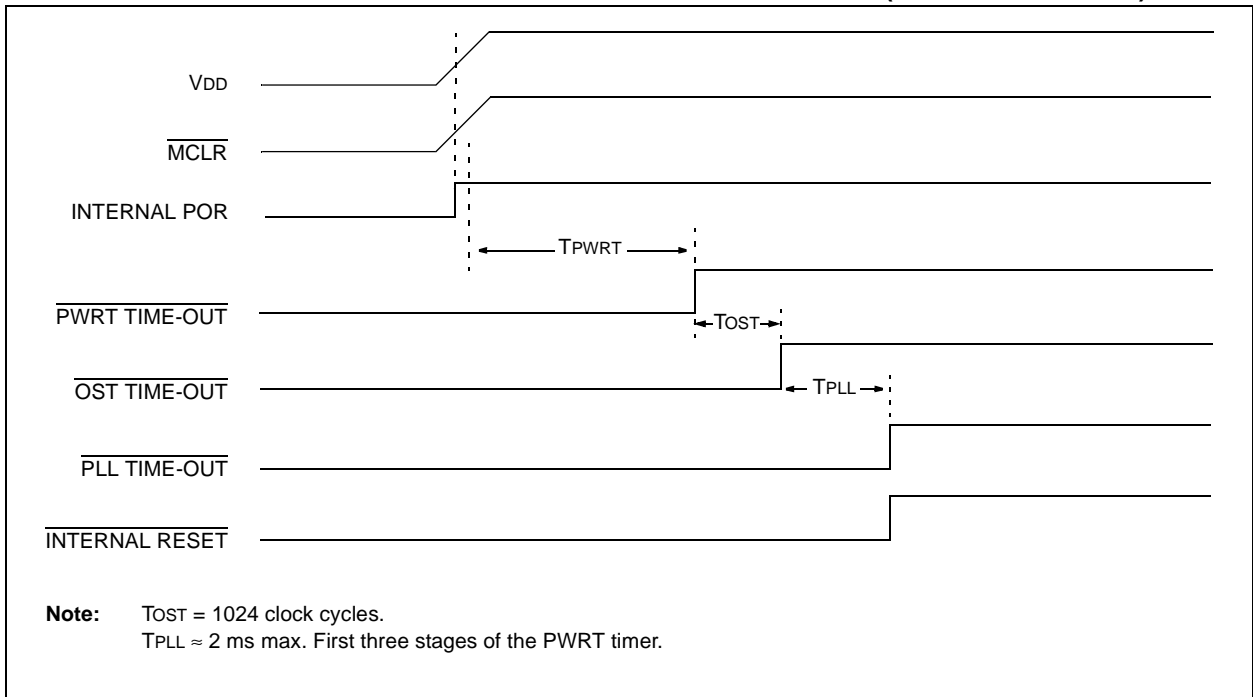


FIGURE 4-7: TIME-OUT SEQUENCE ON POR W/ PLL ENABLED ($\overline{\text{MCLR}}$ TIED TO V_{DD})



PIC18F2455/2550/4455/4550

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, \overline{RI} , \overline{TO} , \overline{PD} , \overline{POR} and \overline{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 Counter | RCON Register | | | | | | STKPTR Register | |
|--|-----------------|------------------|-----------------|-----------------|-----------------|------------------|------------------|-----------------|--------|
| | | SBOREN | \overline{RI} | \overline{TO} | \overline{PD} | \overline{POR} | \overline{BOR} | STKFUL | STKUNF |
| Power-on Reset | 0000h | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| RESET Instruction | 0000h | u ⁽²⁾ | 0 | u | u | u | u | u | u |
| Brown-out | 0000h | u ⁽²⁾ | 1 | 1 | 1 | u | 0 | u | u |
| \overline{MCLR} during Power Managed Run modes | 0000h | u ⁽²⁾ | u | 1 | u | u | u | u | u |
| \overline{MCLR} during Power Managed Idle modes and Sleep mode | 0000h | u ⁽²⁾ | u | 1 | 0 | u | u | u | u |
| WDT Time-out during Full Power or Power Managed Run modes | 0000h | u ⁽²⁾ | u | 0 | u | u | u | u | u |
| \overline{MCLR} during Full Power Execution | 0000h | u ⁽²⁾ | u | u | u | u | u | u | u |
| Stack Full Reset (STVREN = 1) | 0000h | u ⁽²⁾ | u | u | u | u | u | 1 | u |
| Stack Underflow Reset (STVREN = 1) | 0000h | u ⁽²⁾ | u | u | u | u | u | u | 1 |
| Stack Underflow Error (not an actual Reset, STVREN = 0) | 0000h | u ⁽²⁾ | u | u | u | u | u | u | 1 |
| WDT Time-out during Power Managed Idle or Sleep modes | PC + 2 | u ⁽²⁾ | u | 0 | 0 | u | u | u | u |
| Interrupt Exit from Power Managed modes | PC + 2 | u ⁽²⁾ | 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 \overline{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'.

PIC18F2455/2550/4455/4550

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 ⁽¹⁾ |
| TOSH | 2455 | 2550 | 4455 | 4550 | 0000 0000 | 0000 0000 | uuuu uuuu ⁽¹⁾ |
| TOSL | 2455 | 2550 | 4455 | 4550 | 0000 0000 | 0000 0000 | uuuu uuuu ⁽¹⁾ |
| STKPTR | 2455 | 2550 | 4455 | 4550 | 00-0 0000 | uu-0 0000 | uu-u uuuu ⁽¹⁾ |
| 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 ⁽²⁾ |
| 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 ⁽³⁾ |
| INTCON2 | 2455 | 2550 | 4455 | 4550 | 1111 -1-1 | 1111 -1-1 | uuuu -u-u ⁽³⁾ |
| INTCON3 | 2455 | 2550 | 4455 | 4550 | 11-0 0-00 | 11-0 0-00 | uu-u u-uu ⁽³⁾ |
| 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 |

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

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

2: 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'.

PIC18F2455/2550/4455/4550

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 |
| TOCON | 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 | 0-00 0101 | 0-00 0101 | u-uu uuuu |
| WDTCON | 2455 | 2550 | 4455 | 4550 | ---- --0 | ---- --0 | ---- --u |
| RCON ⁽⁴⁾ | 2455 | 2550 | 4455 | 4550 | 0q-1 11q0 | 0q-q qquu | uq-u qquu |
| 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 | xxxx xxxx | uuuu uuuu | 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 |

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

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

2: 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'.

PIC18F2455/2550/4455/4550

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 | 01-0 0-00 | 01-0 0-00 | uu-u 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 |

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

- Note 1:** When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.
- 2: 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'.

PIC18F2455/2550/4455/4550

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 |
| | 2455 | 2550 | 4455 | 4550 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PIR2 | 2455 | 2550 | 4455 | 4550 | 0000 0000 | 0000 0000 | uuuu uuuu ⁽³⁾ |
| | 2455 | 2550 | 4455 | 4550 | 0000 0000 | 0000 0000 | uuuu uuuu ⁽³⁾ |
| PIE2 | 2455 | 2550 | 4455 | 4550 | 0000 0000 | 0000 0000 | uuuu uuuu |
| | 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 ⁽³⁾ |
| | 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 | 0--0 0000 | 0--0 0000 | u--u uuuu |
| TRISE | 2455 | 2550 | 4455 | 4550 | ---- -111 | ---- -111 | uuuu -uuu |
| TRISD | 2455 | 2550 | 4455 | 4550 | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISC | 2455 | 2550 | 4455 | 4550 | 11-- -111 | 11-- -111 | uu-- -uuu |
| TRISB | 2455 | 2550 | 4455 | 4550 | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISA ⁽⁵⁾ | 2455 | 2550 | 4455 | 4550 | -111 1111 ⁽⁵⁾ | -111 1111 ⁽⁵⁾ | -uuu uuuu ⁽⁵⁾ |
| LATE | 2455 | 2550 | 4455 | 4550 | ---- -xxx | ---- -uuu | ---- -uuu |
| LATD | 2455 | 2550 | 4455 | 4550 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| LATC | 2455 | 2550 | 4455 | 4550 | xx-- -xxx | uu-- -uuu | uu-- -uuu |
| LATB | 2455 | 2550 | 4455 | 4550 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| LATA ⁽⁵⁾ | 2455 | 2550 | 4455 | 4550 | -xxx xxxx ⁽⁵⁾ | -uuu uuuu ⁽⁵⁾ | -uuu uuuu ⁽⁵⁾ |
| 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 ⁽⁵⁾ | 2455 | 2550 | 4455 | 4550 | -x0x 0000 ⁽⁵⁾ | -u0u 0000 ⁽⁵⁾ | -uuu uuuu ⁽⁵⁾ |

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

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

2: 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'.

PIC18F2455/2550/4455/4550

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 uu- |
| USTAT | 2455 | 2550 | 4455 | 4550 | -xxx xxx- | -xxx xxx- | -uuu uu- |
| UEIE | 2455 | 2550 | 4455 | 4550 | 0--0 0000 | 0--0 0000 | u--u uuuu |
| UEIR | 2455 | 2550 | 4455 | 4550 | 0--0 0000 | 0--0 0000 | u--u 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 |

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

- Note 1:** When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.
- 2:** 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'.

PIC18F2455/2550/4455/4550

NOTES:

PIC18F2455/2550/4455/4550

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

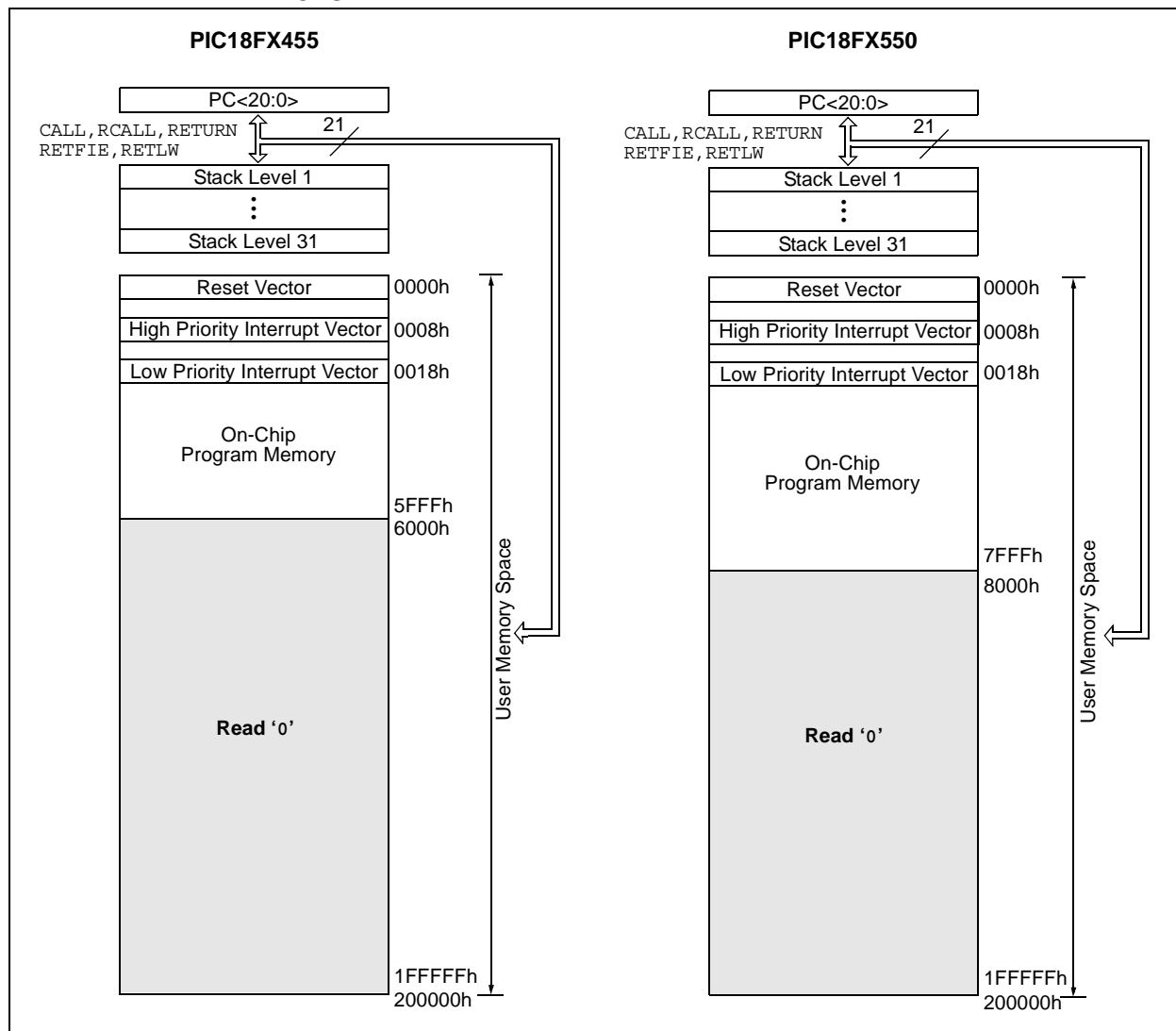
PIC18 microcontrollers implement a 21-bit program counter, which is capable of addressing a 2-Mbyte program memory space. Accessing a location between the upper boundary of the physically implemented memory and the 2-Mbyte address will return all '0's (a NOP instruction).

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



PIC18F2455/2550/4455/4550

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 File 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.

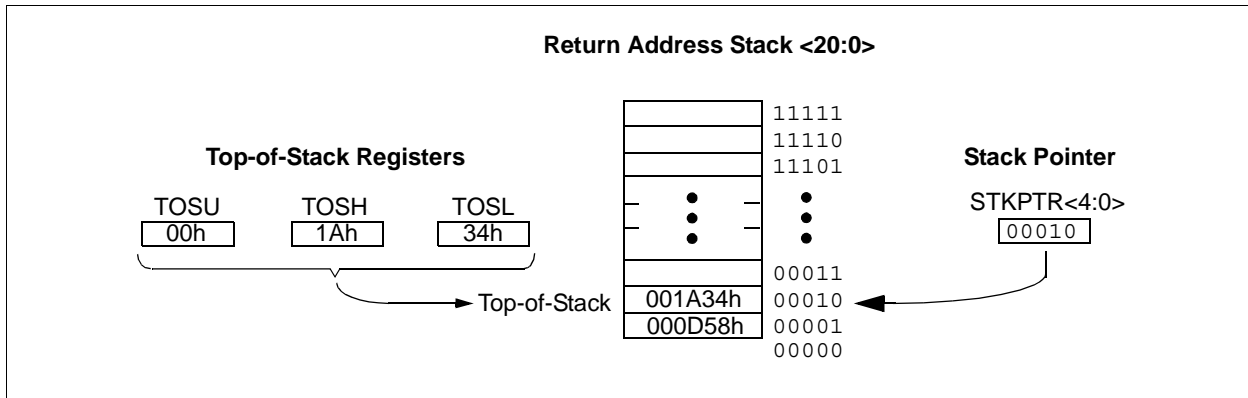
The Stack Pointer is initialized to ‘00000’ after all Resets. There is no RAM associated with the location corresponding to a Stack Pointer value of ‘00000’; this is only a Reset value. Status bits indicate if the stack is full or has overflowed or has underflowed.

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



PIC18F2455/2550/4455/4550

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 bits. The value of the Stack Pointer can be 0 through 31. The Stack Pointer increments before values are pushed onto the stack and decrements after values are popped off the stack. On Reset, the Stack Pointer value will be zero. The user may read and write the Stack Pointer value. This feature can be used by a Real-Time Operating System 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 REGISTER

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

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

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

Legend:

| | | | |
|-------------------|------------------|----------------------|------------------------|
| R = Readable bit | W = Writable bit | U = Unimplemented | C = Clearable only bit |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

PIC18F2455/2550/4455/4550

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 will set the appropriate STKFUL or STKUNF bit and then cause a device Reset. When STVREN is cleared, a full or underflow condition will set the appropriate STKFUL or STKUNF bit but not cause a device Reset. The STKFUL or STKUNF bits are cleared by the user software or a Power-on Reset.

5.1.3 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

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

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

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

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

EXAMPLE 5-2: COMPUTED GOTO USING AN OFFSET VALUE

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

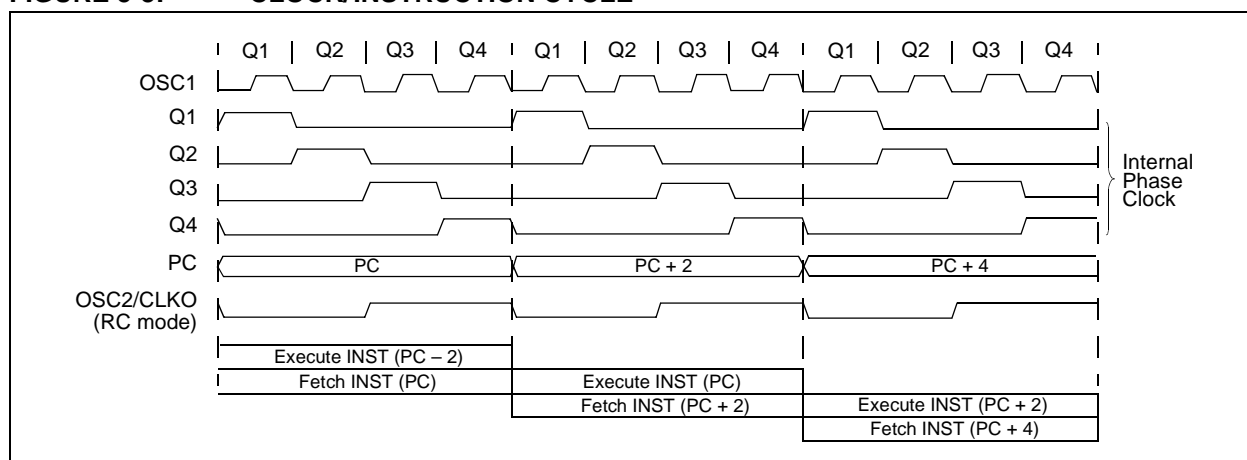
5.2.2 INSTRUCTION FLOW/PIPELINING

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).

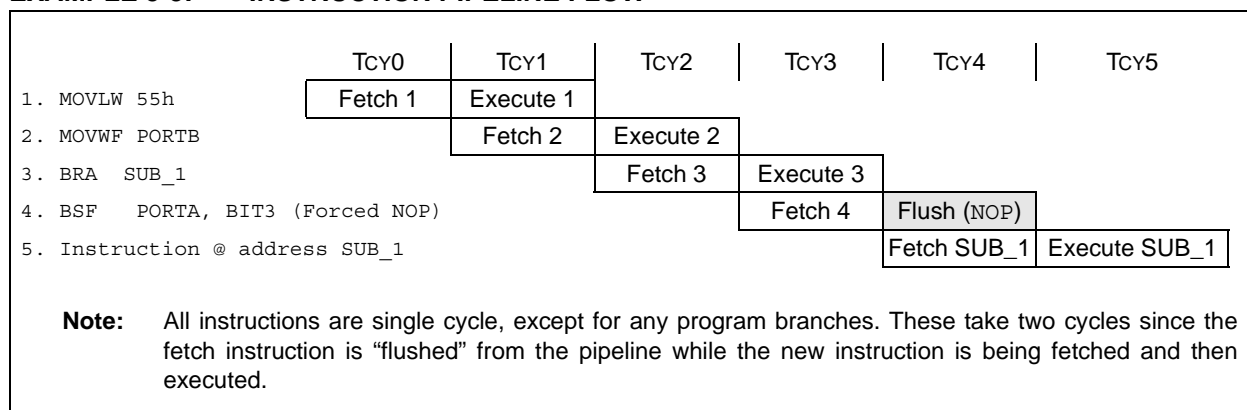
A fetch cycle begins with the Program Counter (PC) incrementing in Q1.

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).

FIGURE 5-3: CLOCK/INSTRUCTION CYCLE



EXAMPLE 5-3: INSTRUCTION PIPELINE FLOW



PIC18F2455/2550/4455/4550

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

| Program Memory Byte Locations → | | | Word Address | | |
|------------------------------------|-------|------------|--------------|---------|---------|
| | | | LSB = 1 | LSB = 0 | |
| | | | | 000000h | |
| | | | | 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 8 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 scratch pad 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 8 Least Significant bits. Only the four lower bits of the BSR are implemented (BSR3:BSR0). The upper four bits are unused; they will always read '0' and cannot be written to. The BSR can be loaded directly by using the `MOVLB` instruction.

The value of the BSR indicates the bank in data memory; the 8 bits in the instruction show the location in the bank and can be thought of as an offset from the bank's lower boundary. The relationship between the BSR's value and the bank division in data memory is shown in Figure 5-6.

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

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

PIC18F2455/2550/4455/4550

FIGURE 5-5: DATA MEMORY MAP FOR PIC18F2455/2550/4455/4550 DEVICES

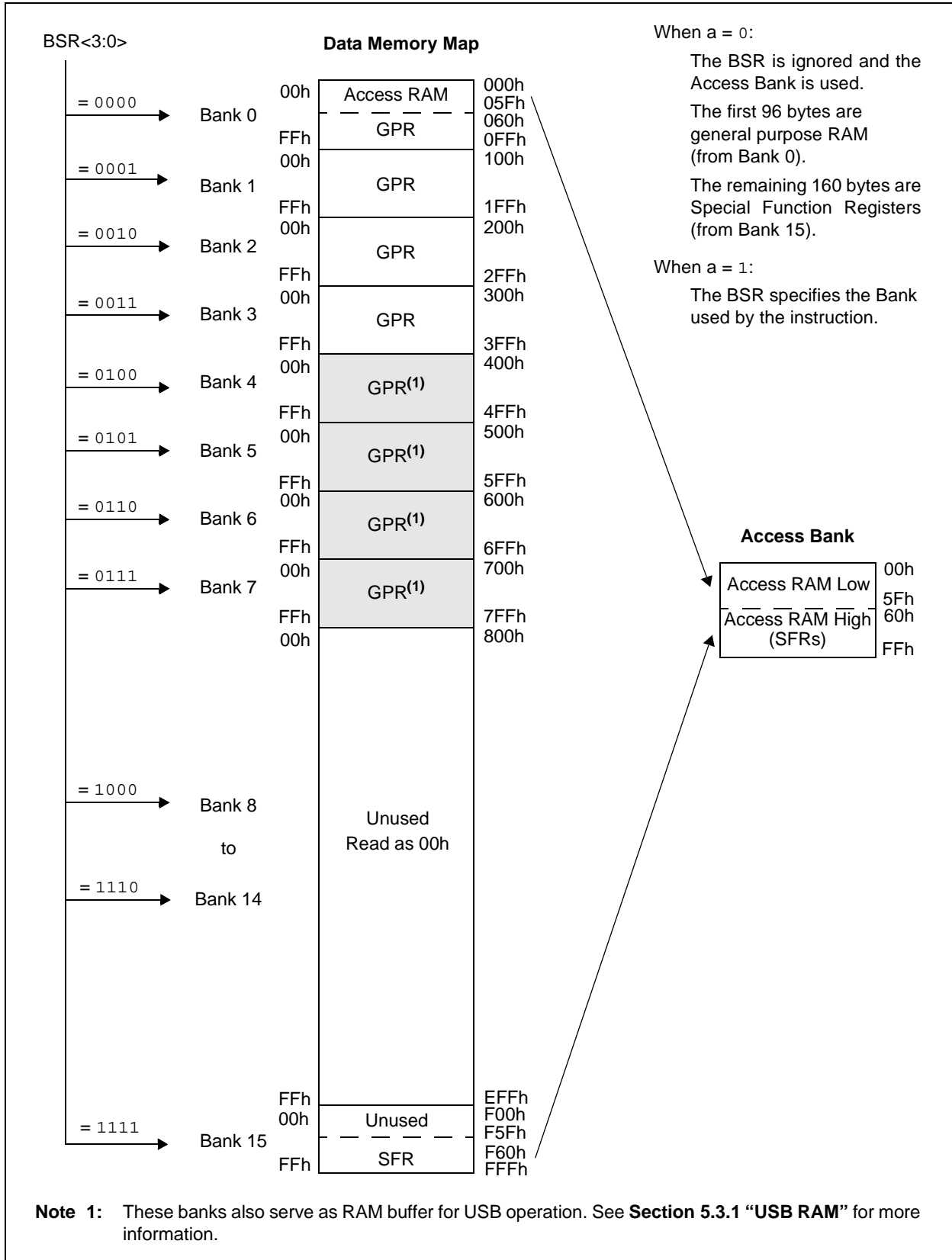
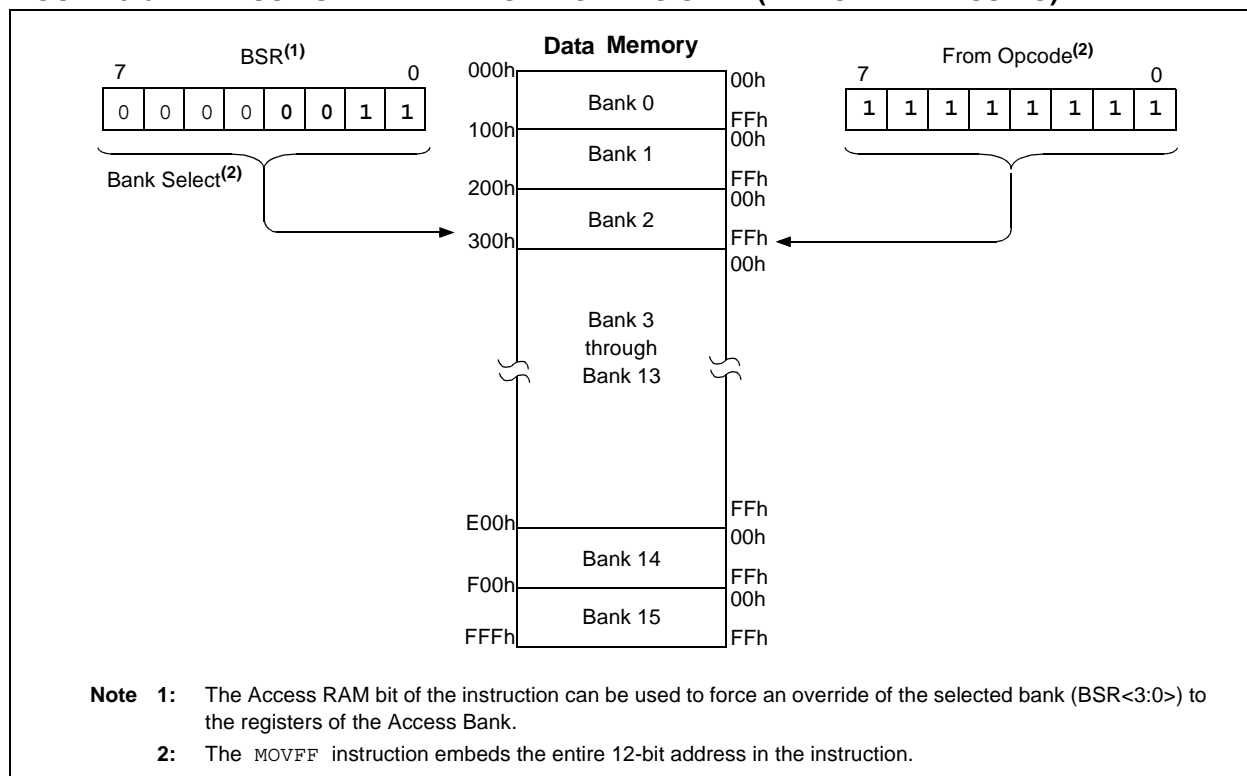


FIGURE 5-6: USE OF THE BANK SELECT REGISTER (DIRECT ADDRESSING)



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. This upper half is also 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.

PIC18F2455/2550/4455/4550

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 features are described in the chapter for that peripheral.

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

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 ⁽¹⁾ | FBFh | CCPR1H | F9Fh | IPR1 | F7Fh | UEP15 |
| FFEh | TOSH | FDEh | POSTINC2 ⁽¹⁾ | FBEh | CCPR1L | F9Eh | PIR1 | F7Eh | UEP14 |
| FFDh | TOSL | FDDh | POSTDEC2 ⁽¹⁾ | FBDh | CCP1CON | F9Dh | PIE1 | F7Dh | UEP13 |
| FFCh | STKPTR | FDCh | PREINC2 ⁽¹⁾ | FBCh | CCPR2H | F9Ch | — ⁽²⁾ | F7Ch | UEP12 |
| FFBh | PCLATU | FDBh | PLUSW2 ⁽¹⁾ | FBHh | 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 ⁽³⁾ | F96h | TRISE ⁽³⁾ | F76h | UEP6 |
| FF5h | TABLAT | FD5h | T0CON | FB5h | CVRCON | F95h | TRISD ⁽³⁾ | 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 | LVDCON | 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 ⁽¹⁾ | FCFh | TMR1H | FAFh | SPBRG | F8Fh | — ⁽²⁾ | F6Fh | UCFG |
| FEHh | POSTINC0 ⁽¹⁾ | FCEh | TMR1L | FAEh | RCREG | F8Eh | — ⁽²⁾ | F6Eh | UADDR |
| FEDh | POSTDEC0 ⁽¹⁾ | FCDh | T1CON | FADh | TXREG | F8Dh | LATE ⁽³⁾ | F6Dh | UCON |
| FECh | PREINC0 ⁽¹⁾ | FCCh | TMR2 | FACH | TXSTA | F8Ch | LATD ⁽³⁾ | F6Ch | USTAT |
| FEBh | PLUSW0 ⁽¹⁾ | FCBh | PR2 | FABh | RCSTA | F8Bh | LATC | F6Bh | UEIE |
| FEAh | FSR0H | FCAh | T2CON | FAAh | EEADRH | 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 ⁽¹⁾ | FC7h | SSPSTAT | FA7h | EECON2 ⁽¹⁾ | F87h | — ⁽²⁾ | F67h | UFRMH |
| FE6h | POSTINC1 ⁽¹⁾ | FC6h | SSPCON1 | FA6h | EECON1 | F86h | — ⁽²⁾ | F66h | UFRML |
| FE5h | POSTDEC1 ⁽¹⁾ | FC5h | SSPCON2 | FA5h | — ⁽²⁾ | F85h | — ⁽²⁾ | F65h | SPPCON ⁽³⁾ |
| FE4h | PREINC1 ⁽¹⁾ | FC4h | ADRESH | FA4h | — ⁽²⁾ | F84h | PORTE | F64h | SPPEPS ⁽³⁾ |
| FE3h | PLUSW1 ⁽¹⁾ | FC3h | ADRESL | FA3h | — ⁽²⁾ | F83h | PORTD ⁽³⁾ | F63h | SPPCFG ⁽³⁾ |
| FE2h | FSR1H | FC2h | ADCON0 | FA2h | IPR2 | F82h | PORTC | F62h | SPPDATA ⁽³⁾ |
| 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.
Note 2: Unimplemented registers are read as '0'.
Note 3: Registers are implemented only on 40/44-pin devices.

PIC18F2455/2550/4455/4550

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 | — | — | — | Top-of-Stack Upper Byte (TOS<20:16>) | | | | | ---0 0000 | 51, 58 |
| TOSH | Top-of-Stack High Byte (TOS<15:8>) | | | | | | | | 0000 0000 | 51, 58 |
| TOSL | Top-of-Stack Low Byte (TOS<7:0>) | | | | | | | | 0000 0000 | 51, 58 |
| STKPTR | STKOVF | STKUNF | — | Return Stack Pointer | | | | | 00-0 0000 | 51, 59 |
| PCLATU | — | — | bit 21 ⁽¹⁾ | Holding Register for PC<20:16> | | | | | ---0 0000 | 51, 58 |
| PCLATH | Holding Register for PC<15:8> | | | | | | | | 0000 0000 | 51, 58 |
| PCL | PC Low Byte (PC<7:0>) | | | | | | | | 0000 0000 | 51, 58 |
| TBLPTRU | — | — | bit 21 | Program Memory Table Pointer Upper Byte (TBLPTR<20:16>) | | | | | --00 0000 | 51, 82 |
| TBLPTRH | Program Memory Table Pointer High Byte (TBLPTR<15:8>) | | | | | | | | 0000 0000 | 51, 82 |
| TBLPTRL | Program Memory Table Pointer Low Byte (TBLPTR<7:0>) | | | | | | | | 0000 0000 | 51, 82 |
| TABLAT | Program Memory Table Latch | | | | | | | | 0000 0000 | 51, 82 |
| PRODH | Product Register High Byte | | | | | | | | xxxx xxxx | 51, 95 |
| PRODL | Product Register Low Byte | | | | | | | | xxxx xxxx | 51, 95 |
| INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 0000 000x | 51, 99 |
| INTCON2 | RBPV | 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 of FSR0 to address data memory – value of FSR0 not changed (not a physical register) | | | | | | | | N/A | 51, 73 |
| POSTINC0 | Uses contents of FSR0 to address data memory – value of FSR0 post-incremented (not a physical register) | | | | | | | | N/A | 51, 74 |
| POSTDEC0 | Uses contents of FSR0 to address data memory – value of FSR0 post-decremented (not a physical register) | | | | | | | | N/A | 51, 74 |
| PREINC0 | Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register) | | | | | | | | N/A | 51, 74 |
| PLUSW0 | Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register) – value of FSR0 offset by W | | | | | | | | N/A | 51, 74 |
| FSR0H | — | — | — | — | Indirect Data Memory Address Pointer 0 High | | | | ---- 0000 | 51, 73 |
| FSR0L | Indirect Data Memory Address Pointer 0 Low Byte | | | | | | | | xxxx xxxx | 51, 73 |
| WREG | Working Register | | | | | | | | xxxx xxxx | 51 |
| INDF1 | Uses contents of FSR1 to address data memory – value of FSR1 not changed (not a physical register) | | | | | | | | N/A | 51, 73 |
| POSTINC1 | Uses contents of FSR1 to address data memory – value of FSR1 post-incremented (not a physical register) | | | | | | | | N/A | 51, 74 |
| POSTDEC1 | Uses contents of FSR1 to address data memory – value of FSR1 post-decremented (not a physical register) | | | | | | | | N/A | 51, 74 |
| PREINC1 | Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register) | | | | | | | | N/A | 51, 74 |
| PLUSW1 | Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register) – value of FSR1 offset by W | | | | | | | | N/A | 51, 74 |
| FSR1H | — | — | — | — | Indirect Data Memory Address Pointer 1 High | | | | ---- 0000 | 51, 73 |
| FSR1L | Indirect Data Memory Address Pointer 1 Low Byte | | | | | | | | xxxx xxxx | 51, 73 |
| BSR | — | — | — | — | Bank Select Register | | | | ---- 0000 | 52, 63 |
| INDF2 | Uses contents of FSR2 to address data memory – value of FSR2 not changed (not a physical register) | | | | | | | | N/A | 52, 73 |
| POSTINC2 | Uses contents of FSR2 to address data memory – value of FSR2 post-incremented (not a physical register) | | | | | | | | N/A | 52, 74 |
| POSTDEC2 | Uses contents of FSR2 to address data memory – value of FSR2 post-decremented (not a physical register) | | | | | | | | N/A | 52, 74 |
| PREINC2 | Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register) | | | | | | | | N/A | 52, 74 |
| PLUSW2 | Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register) – value of FSR2 offset by W | | | | | | | | N/A | 52, 74 |
| FSR2H | — | — | — | — | Indirect Data Memory Address Pointer 2 High | | | | ---- 0000 | 52, 73 |
| FSR2L | Indirect Data Memory Address Pointer 2 Low Byte | | | | | | | | xxxx xxxx | 52, 73 |
| STATUS | — | — | — | N | OV | Z | DC | C | ---x xxxxx | 52, 71 |
| TMR0H | Timer0 Register High Byte | | | | | | | | 0000 0000 | 52, 127 |
| TMR0L | Timer0 Register Low Byte | | | | | | | | xxxx xxxx | 52, 127 |
| T0CON | TMR0ON | T08BIT | T0CS | T0SE | T0PS3 | T0PS2 | T0PS1 | T0PS0 | 1111 1111 | 52, 125 |

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

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

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

Note 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 '-'.
Note 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'.

Note 5: RE3 is only available as a port pin when the MCLRE configuration bit is clear; otherwise, the bit reads as '0'.

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

PIC18F2455/2550/4455/4550

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 |
|-----------|--|-----------------------|---------------------|---------------------|---------------------|---------------------|-----------------------|-----------------------|-------------------|-----------------|
| OSCCON | IDLEN | IRCF2 | IRCF1 | IRCF0 | OSTS | FLTS | SCS1 | SCS0 | 0100 q000 | 52, 32 |
| HLVDCON | VDIRMAG | — | IRVST | LVDEN | LVDL3 | LVDL2 | LVDL1 | LVDL0 | 0-00 0101 | 52, 273 |
| WDTCON | — | — | — | — | — | — | — | SWDTEN | --- ---0 | 52, 292 |
| RCON | IPEN | SBOREN ⁽²⁾ | — | RI | TO | PD | POR | BOR | 0q-1 11q0 | 52, 44 |
| TMR1H | Timer1 Register High Byte | | | | | | | | xxxx xxxx | 52, 133 |
| TMR1L | Timer1 Register Low Bytes | | | | | | | | xxxx xxxx | 52, 133 |
| T1CON | RD16 | T1RUN | T1CKPS1 | T1CKPS0 | T1OSCEN | T1SYNC | TMR1CS | TMR1ON | 0000 0000 | 52, 129 |
| TMR2 | Timer2 Register | | | | | | | | 0000 0000 | 52, 136 |
| PR2 | Timer2 Period Register | | | | | | | | 1111 1111 | 52, 136 |
| T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | 52, 135 |
| SSPBUF | SSP Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | 52, 194, 202 |
| SSPADD | SSP Address Register in I ² C™ Slave Mode. SSP Baud Rate Reload Register in I ² C Master Mode. | | | | | | | | 0000 0000 | 52, 202 |
| SSPSTAT | SMP | CKE | D/A | P | 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 | ACKEN | RCEN | PEN | RSEN | SEN | 0000 0000 | 52, 205 |
| ADRESH | A/D Result Register High Byte | | | | | | | | xxxx xxxx | 52, 262 |
| ADRESL | A/D Result Register Low Byte | | | | | | | | xxxx xxxx | 52, 262 |
| ADCON0 | — | — | CHS3 | CHS2 | CHS1 | CHS0 | GO/DONE | ADON | --00 0000 | 52, 253 |
| ADCON1 | — | — | VCFG1 | VCFG0 | PCFG3 | PCFG2 | PCFG1 | PCFG0 | --00 0qqq | 52, 254 |
| ADCON2 | ADFM | — | ACQT2 | ACQT1 | ACQT0 | ADCS2 | ADCS1 | ADCS0 | 0-00 0000 | 52, 255 |
| CCPR1H | Capture/Compare/PWM Register1 High Byte | | | | | | | | xxxx xxxx | 53, 142 |
| CCPR1L | Capture/Compare/PWM Register1 Low Byte | | | | | | | | xxxx xxxx | 53, 142 |
| CCP1CON | P1M1 ⁽³⁾ | P1M0 ⁽³⁾ | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | 0000 0000 | 53, 141, 149 |
| CCPR2H | Capture/Compare/PWM Register2 High Byte | | | | | | | | xxxx xxxx | 53, 142 |
| CCPR2L | Capture/Compare/PWM Register2 Low Byte | | | | | | | | xxxx xxxx | 53, 142 |
| CCP2CON | — | — | DC2B1 | DC2B0 | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | 0000 0000 | 53, 141 |
| BAUDCON | ABDOVF | RCMT | — | SCKP | BRG16 | — | WUE | ABDEN | 01-0 0-00 | 53, 236 |
| ECCP1DEL | PRSEN | PDC6 ⁽³⁾ | PDC5 ⁽³⁾ | PDC4 ⁽³⁾ | PDC3 ⁽³⁾ | PDC2 ⁽³⁾ | PDC1 ⁽³⁾ | PDC0 ⁽³⁾ | 0000 0000 | 53, 158 |
| ECCP1AS | ECCPASE | ECCPAS2 | ECCPAS1 | ECCPAS0 | PSSAC1 | PSSAC0 | PSSBD1 ⁽³⁾ | PSSBD0 ⁽³⁾ | 0000 0000 | 53, 159 |
| CVRCON | CVREN | CVROE | CVRR | CVRSS | CVR3 | CVR2 | CVR1 | CVR0 | 0000 0000 | 53, 269 |
| CMCON | C2OUT | C1OUT | C2INV | C1INV | CIS | CM2 | CM1 | CM0 | 0000 0111 | 53, 263 |
| TMR3H | Timer3 Register High Byte | | | | | | | | xxxx xxxx | 53, 139 |
| TMR3L | Timer3 Register Low Byte | | | | | | | | xxxx xxxx | 53, 139 |
| T3CON | RD16 | T3CCP2 | T3CKPS1 | T3CKPS0 | T3CCP1 | T3SYNC | TMR3CS | TMR3ON | 0000 0000 | 53, 137 |
| SPBRGH | USART Baud Rate Generator High Byte | | | | | | | | 0000 0000 | 53, 237 |
| SPBRG | USART Baud Rate Generator | | | | | | | | 0000 0000 | 53, 237 |
| RCREG | USART Receive Register | | | | | | | | 0000 0000 | 53, 244 |
| TXREG | USART Transmit Register | | | | | | | | 0000 0000 | 53, 242 |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 0000 0010 | 53, 234 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADEN | FERR | OERR | RX9D | 0000 000x | 53, 235 |

Legend:

x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

Note

- 1: Bit 21 of the PC is only available in Serial Programming mode.
- 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).

PIC18F2455/2550/4455/4550

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 Address Register | | | | | | | | 0000 0000 | 53, 89 |
| EEDATA | EEPROM Data Register | | | | | | | | 0000 0000 | 53, 89 |
| EECON2 | EEPROM Control Register 2 (not a physical register) | | | | | | | | 0000 0000 | 53, 80 |
| EECON1 | EEPGD | CFG5 | — | FREE | WRERR | WREN | WR | RD | xx-0 x000 | 53, 81 |
| IPR2 | OSCFIP | CMIP | USBIP | EEIP | BCLIP | LVDIP | TMR3IP | CCP2IP | 1111 1111 | 54, 107 |
| PIR2 | OSCFIF | CMIF | USBIF | EEIF | BCLIF | LVDIF | TMR3IF | CCP2IF | 0000 0000 | 54, 103 |
| PIE2 | OSCFIE | CMIE | USBIE | EEIE | BCLIE | LVDIE | TMR3IE | CCP2IE | 0000 0000 | 54, 105 |
| IPR1 | SPPIF ⁽³⁾ | ADIP | RCIP | TXIP | SSPIF | CCP1IP | TMR2IP | TMR1IP | 1111 1111 | 54, 106 |
| PIR1 | SPPIF ⁽³⁾ | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 54, 102 |
| PIE1 | SPPIE ⁽³⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 54, 104 |
| OSCTUNE | INTSRC | — | — | TUN4 | TUN3 | TUN2 | TUN1 | TUN0 | 0--0 0000 | 54, 28 |
| TRISE ⁽³⁾ | — | — | — | — | — | TRISE2 | TRISE1 | TRISE0 | 0000 -111 | 54, 124 |
| TRISD ⁽³⁾ | TRISD7 | TRISD6 | TRISD5 | TRISD4 | TRISD3 | TRISD2 | TRISD1 | TRISD0 | 1111 1111 | 54, 122 |
| TRISC | TRISC7 | TRISC6 | — | — | — | TRISC2 | TRISC1 | TRISC0 | 11-- -111 | 54, 119 |
| TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 1111 1111 | 54, 116 |
| TRISA | — | TRISA6 ⁽⁴⁾ | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | -111 1111 | 54, 113 |
| LATE ⁽³⁾ | — | — | — | — | — | LATE2 | LATE1 | LATE0 | --- -xxx | 54, 124 |
| LATD ⁽³⁾ | LATD7 | LATD6 | LATD5 | LATD4 | LATD3 | LATD2 | LATD1 | LATD0 | xxxx xxxx | 54, 122 |
| LATC | LATC7 | LATC6 | — | — | — | LATC2 | LATC1 | LATC0 | xx-- -xxx | 54, 119 |
| LATB | LATB7 | LATB6 | LATB5 | LATB4 | LATB3 | LATB2 | LATB1 | LATB0 | xxxx xxxx | 54, 116 |
| LATA | — | LATA6 ⁽⁴⁾ | LATA5 | LATA4 | LATA3 | LATA2 | LATA1 | LATA0 | xxxx xxxx | 54, 113 |
| PORTE | RDPU ⁽³⁾ | — | — | — | RE3 ⁽⁵⁾ | RE2 ⁽³⁾ | RE1 ⁽³⁾ | RE0 ⁽³⁾ | 0--- xxxx | 54, 123 |
| PORTD ⁽³⁾ | RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | RD0 | xxxx xxxx | 54, 122 |
| PORTC | RC7 | RC6 | RC5 ⁽⁶⁾ | RC4 ⁽⁶⁾ | — | RC2 | RC1 | RC0 | xxxx -xxx | 54, 119 |
| PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | 54, 116 |
| PORTA | — | RA6 ⁽⁴⁾ | 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 |
| UEP0 | — | — | — | EPHSHK | EPCONDIS | EPOUTEN | EPINEN | EPSTALL | ---0 0000 | 55, 169 |

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

- Note**
- 1: Bit 21 of the PC is only available in Serial Programming mode.
 - 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).

PIC18F2455/2550/4455/4550

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 | UPP1 | UPP0 | 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 | 0--0 0000 | 55, 181 |
| UEIR | BTSEF | — | — | BTOEF | DFN8EF | CRC16EF | CRC5EF | PIDEF | 0--0 0000 | 55, 180 |
| UIE | — | SOFIE | STALLIE | IDLEIE | TRNIE | ACTIVIE | UERRIE | URSTIE | -000 0000 | 55, 179 |
| UIR | — | SOFIF | STALLIF | IDLEIF | TRNIF | ACTIVIF | UERRIF | URSTIF | -000 0000 | 55, 178 |
| UFRMH | — | — | — | — | — | FRM10 | FRM9 | FRM8 | ---- -xxx | 55, 170 |
| UFRML | FRM7 | FRM6 | FRM5 | FRM4 | FRM3 | FRM2 | FRM1 | FRM0 | xxxx xxx | 55, 170 |
| SPPCON ⁽³⁾ | — | — | — | — | — | — | SPPOWN | SPPEN | ---- --00 | 55, 187 |
| SPPEPS ⁽³⁾ | RDSPP | WRSP | — | BUSY | ADDR3 | ADDR2 | ADDR1 | ADDR0 | 00-0 0000 | 55, 191 |
| SPPCFG ⁽³⁾ | CLKCFG1 | CLKCFG0 | CSEN | CLK1EN | WS3 | WS2 | WS1 | WS0 | 0000 0000 | 55, 188 |
| SPPDATA ⁽³⁾ | DATA7 | DATA6 | DATA5 | DATA4 | DATA3 | DATA2 | DATA1 | DATA0 | 0000 0000 | 55, 192 |

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

- Note**
- 1: Bit 21 of the PC is only available in Serial Programming mode.
 - 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).

PIC18F2455/2550/4455/4550

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 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 u1uu').

It is recommended that only `BCF`, `BSF`, `SWAPF`, `MOVWF` 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 | C |
| bit 7 | | | | | | | | bit 0 |

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **N:** Negative bit

This bit is used for signed arithmetic (2's complement). It indicates whether the result was negative (ALU MSB = 1).

1 = Result was negative

0 = Result was positive

bit 3 **OV:** Overflow bit

This bit is used for signed arithmetic (2's complement). It indicates an overflow of the 7-bit magnitude which causes the sign bit (bit 7) to change state.

1 = Overflow occurred for signed arithmetic (in this arithmetic operation)

0 = No overflow occurred

bit 2 **Z:** Zero bit

1 = The result of an arithmetic or logic operation is zero

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

bit 1 **DC:** Digit carry/borrow bit

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

Note: For borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either bit 4 or bit 3 of the source register.

bit 0 **C:** Carry/borrow bit

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

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

PIC18F2455/2550/4455/4550

5.4 Data Addressing Modes

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

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

The addressing modes are:

- Inherent
- Literal
- Direct
- Indirect

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

5.4.1 INHERENT AND LITERAL ADDRESSING

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

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

5.4.2 DIRECT ADDRESSING

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

In the core PIC18 instruction set, bit-oriented and byte-oriented instructions use some version of direct addressing by default. All of these instructions include some 8-bit literal address as their Least Significant Byte. This address specifies either a register address in one of the banks of data RAM (**Section 5.3.4 “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 File 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 FSR0H, 1   ; All done with
      ; Bank1?
      BRA   NEXT       ; NO, clear next
CONTINUE                                ; YES, continue
```

5.4.3.1 FSR Registers and the INDF Operand

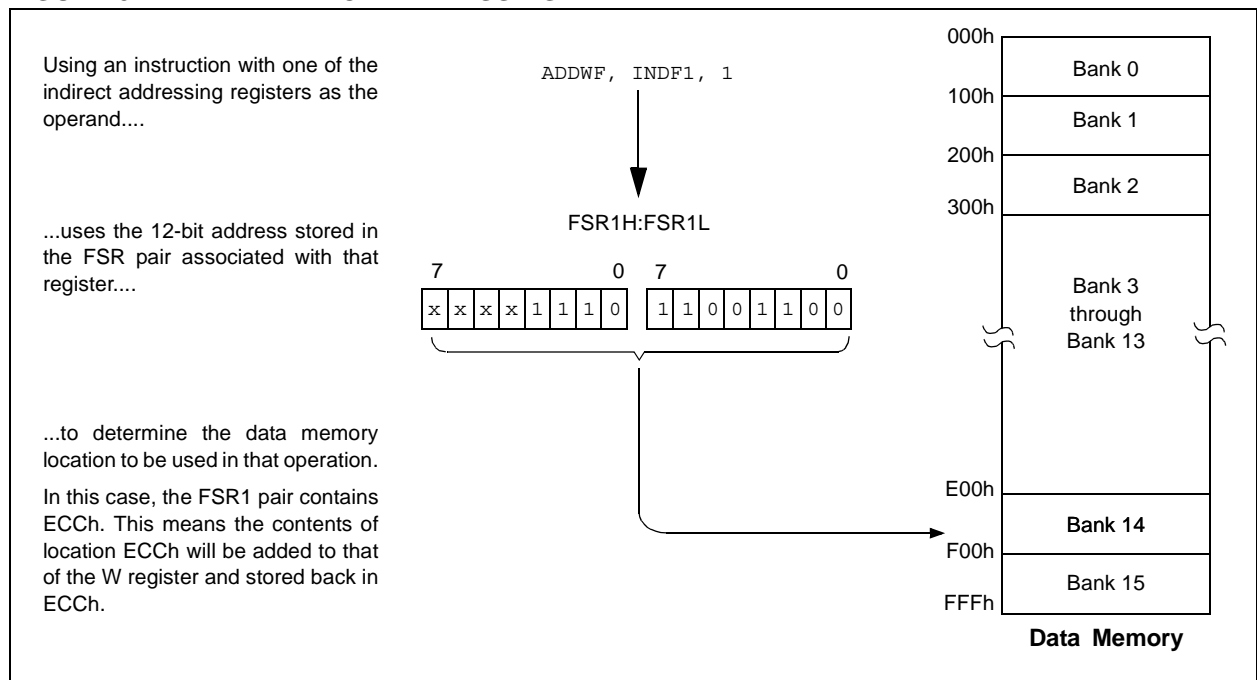
At the core of indirect addressing are three sets of registers: FSR0, FSR1 and FSR2. Each represents a pair of 8-bit registers, FSRnH and FSRnL. The four upper bits of the FSRnH register are not used, so each FSR pair holds a 12-bit value. This represents a value that can address the entire range of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



PIC18F2455/2550/4455/4550

5.4.3.2 FSR Registers and POSTINC, POSTDEC, PREINC and PLUSW

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

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

In this context, accessing an INDF register uses the value in the FSR registers without changing them. Similarly, accessing a PLUSW register gives the FSR value offset by 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, roll-overs of the FSRnL register from FFh to 00h carry over to the FSRnH register. On the other hand, results of these operations do not change the value of any flags in the Status register (e.g., Z, N, OV, etc.).

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

5.4.3.3 Operations by FSRs on FSRs

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

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

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

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

5.5 Program Memory and the Extended Instruction Set

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

Enabling the extended instruction set adds eight additional two-word commands to the existing PIC18 instruction set: `ADDFSR`, `ADDLNLK`, `CALLW`, `MOVSF`, `MOVSS`, `PUSHL`, `SUBFSR` and `SUBLNLK`. 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 remain unchanged.

5.6.1 INDEXED ADDRESSING WITH LITERAL OFFSET

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

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

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

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

5.6.2 INSTRUCTIONS AFFECTED BY INDEXED LITERAL OFFSET MODE

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

Additionally, byte-oriented and bit-oriented instructions are not affected if they use the Access Bank (Access RAM bit is `'1'`), or include a file address of `60h` or above. Instructions meeting these criteria will continue to execute as before. A comparison of the different possible addressing modes when the extended instruction set is enabled is shown in Figure 5-8.

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”**.

PIC18F2455/2550/4455/4550

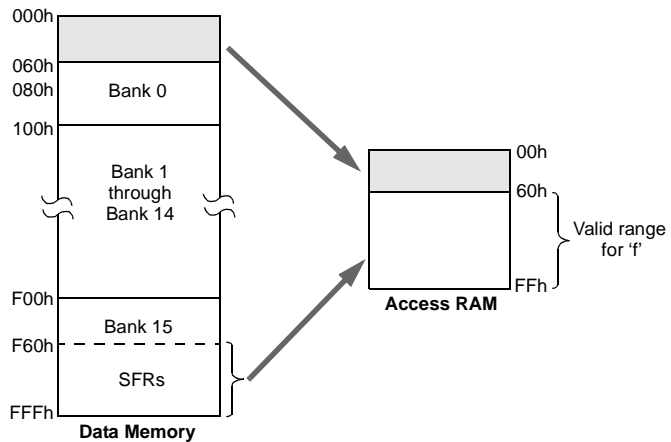
FIGURE 5-8: COMPARING ADDRESSING OPTIONS FOR BIT-ORIENTED AND BYTE-ORIENTED INSTRUCTIONS (EXTENDED INSTRUCTION SET ENABLED)

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

When a = 0 and f ≥ 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.

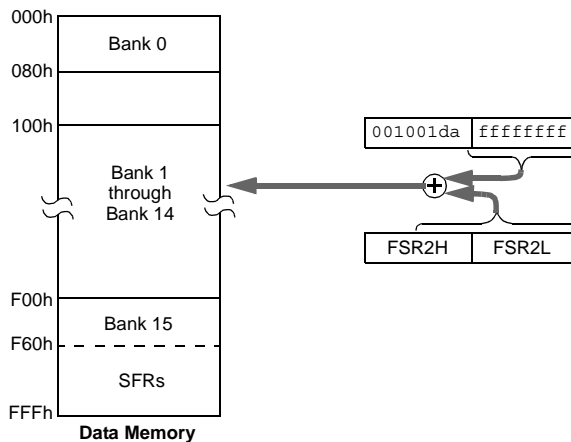


When a = 0 and f ≤ 5Fh:

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

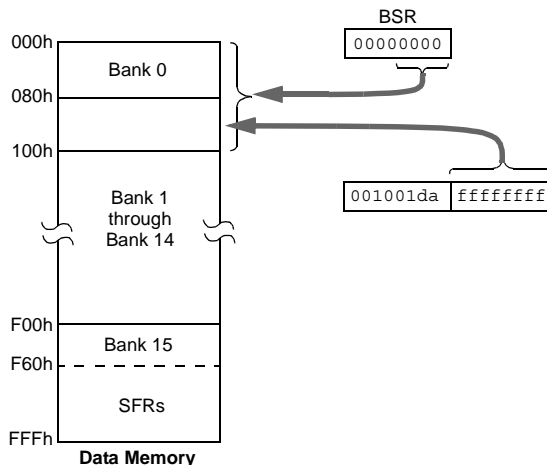
Note that in this mode, the correct syntax is now:

ADDWF [k], d
where 'k' is the same as 'f'.



When a = 1 (all values of f):

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



5.6.3 MAPPING THE ACCESS BANK IN INDEXED LITERAL OFFSET MODE

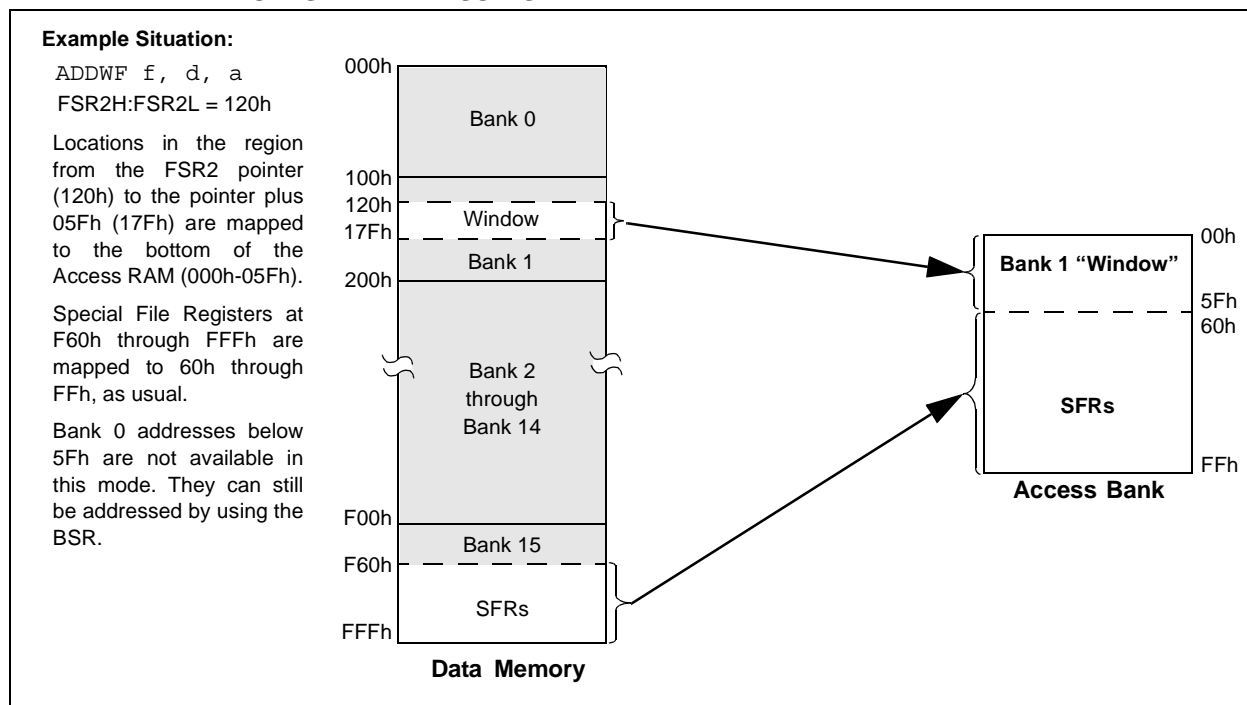
The use of Indexed Literal Offset Addressing mode effectively changes how the lower portion of Access RAM (00h to 5Fh) is mapped. Rather than containing just the contents of the bottom half of Bank 0, this mode maps the contents from Bank 0 and a user defined “window” that can be located anywhere in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



PIC18F2455/2550/4455/4550

NOTES:

6.0 FLASH PROGRAM MEMORY

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

A read from program memory is executed on one byte at a time. A write to program memory is executed on blocks of 64 bytes at a time. 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 and erases.

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

6.1 Table Reads and Table Writes

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

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

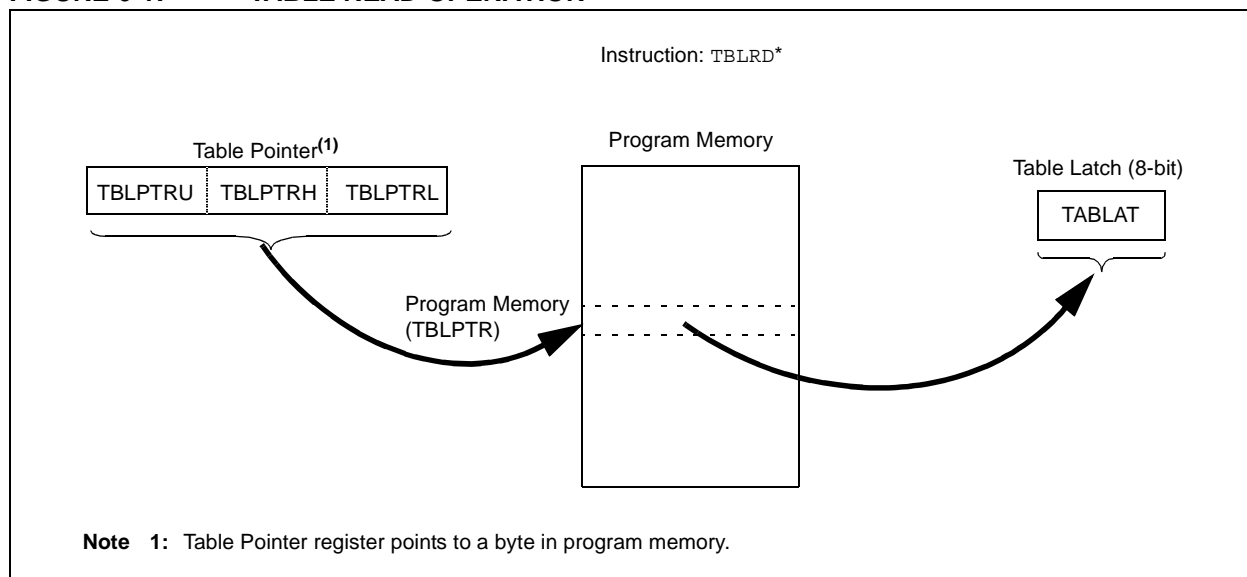
The program memory space is 16 bits wide, while the data RAM space is 8 bits wide. Table reads and table writes move data between these two memory spaces through an 8-bit register (TABLAT).

Table read operations retrieve data from program memory and places 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.

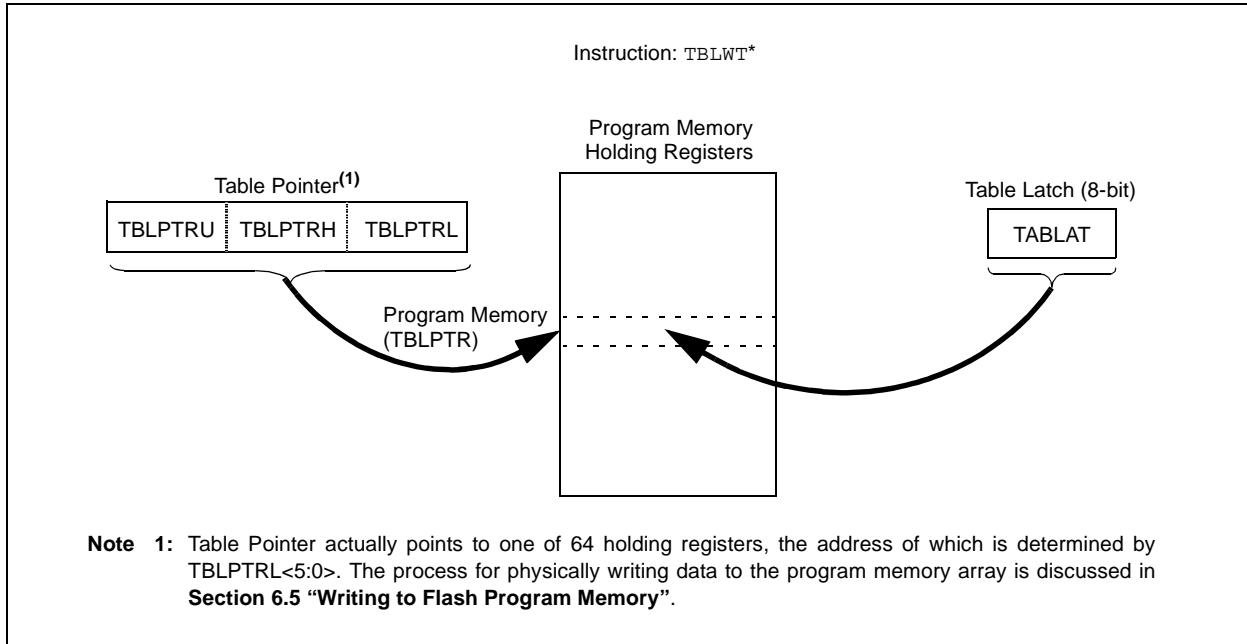
Table operations work with byte entities. A table block containing data, rather than program instructions, is not required to be word aligned. Therefore, a table block can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions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

FIGURE 6-1: TABLE READ OPERATION



PIC18F2455/2550/4455/4550

FIGURE 6-2: TABLE WRITE OPERATION



6.2 Control Registers

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

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

6.2.1 EECON1 AND EECON2 REGISTERS

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

The 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.

PIC18F2455/2550/4455/4550

REGISTER 6-1: EECOM1 REGISTER

| 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 | WREN | WR | RD |
| bit 7 | | | | | | bit 0 | |

- 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 = 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
Note: When a WRERR occurs, the EEGPD and CFGS bits are not cleared. This allows tracing of the error condition.
- 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 EEGPD = 1 or CFGS = 1.)
 0 = Does not initiate an EEPROM read

Legend:

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

PIC18F2455/2550/4455/4550

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 six LSbs of the Table Pointer register (TBLPTR<5:0>) determine which of the 64 program memory holding registers is written to. When the timed write to program memory begins (via the WR bit), the 16 MSBs of the TBLPTR (TBLPTR<21:6>) determine which program memory block of 64 bytes is written to. For more detail, see **Section 6.5 “Writing to Flash Program Memory”**.

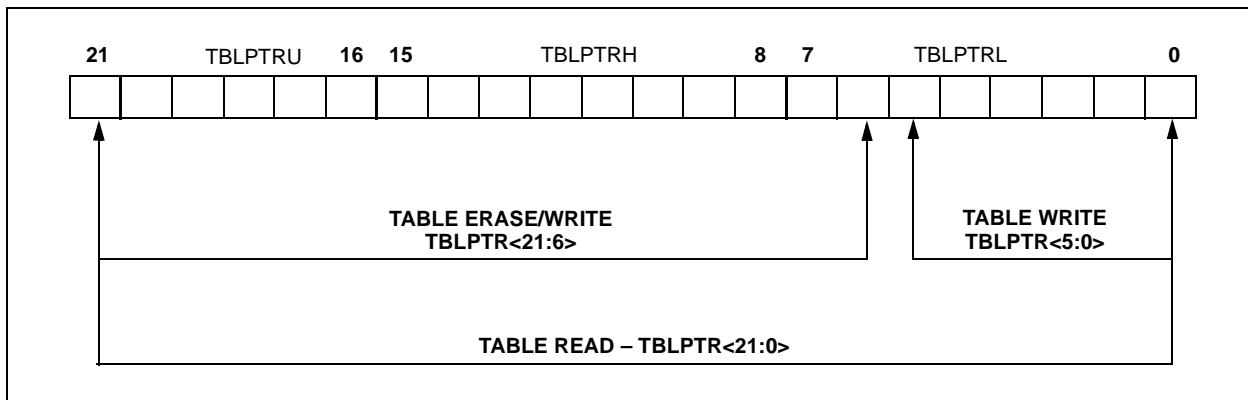
When an erase of program memory is executed, the 16 MSBs of the Table Pointer register (TBLPTR<21:6>) point to the 64-byte block that will be erased. The Least Significant bits (TBLPTR<5:0>) are ignored.

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

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

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

FIGURE 6-3: TABLE POINTER BOUNDARIES BASED ON OPERATION



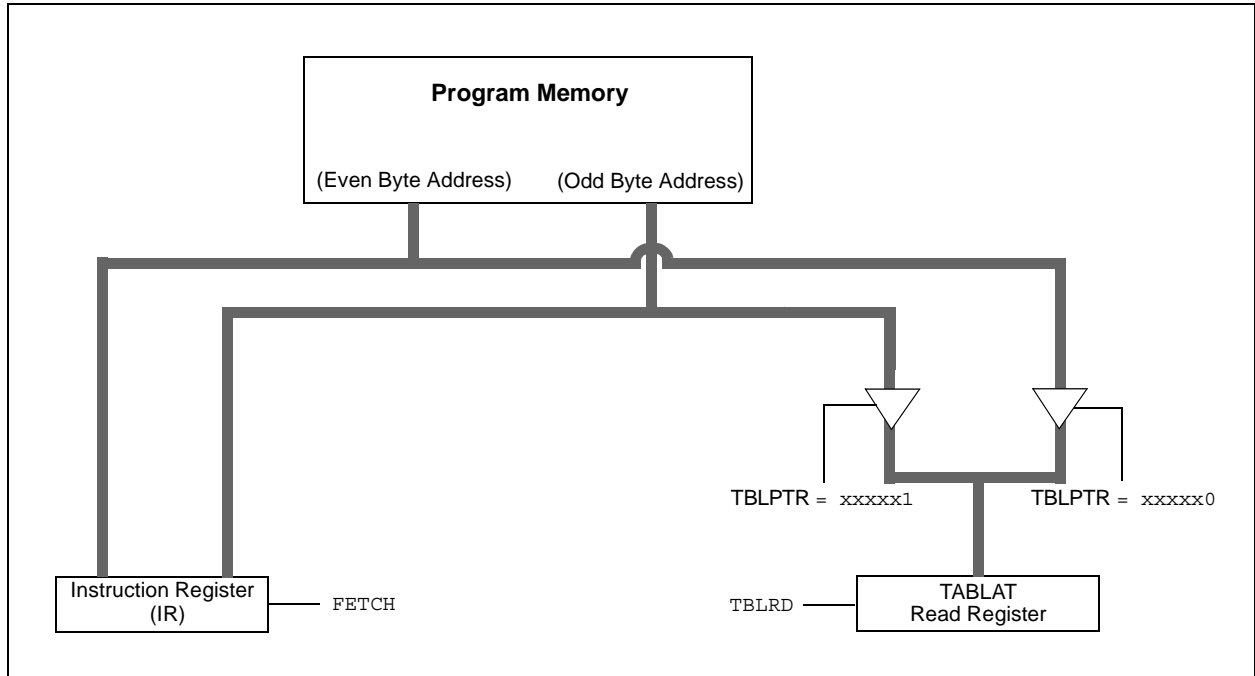
6.3 Reading the Flash Program Memory

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

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

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

FIGURE 6-4: READS FROM FLASH PROGRAM MEMORY



EXAMPLE 6-1: READING A FLASH PROGRAM MEMORY WORD

```

        MOVLW    CODE_ADDR_UPPER        ; Load TBLPTR with the base
        MOVWF   TBLPTRU                 ; address of the word
        MOVLW    CODE_ADDR_HIGH
        MOVWF   TBLPTRH
        MOVLW    CODE_ADDR_LOW
        MOVWF   TBLPTRL
READ_WORD
        TBLRD*+                          ; 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
    
```

PIC18F2455/2550/4455/4550

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 microcontroller 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 location is:

1. 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.
6. 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 Sequence | MOVLW | 55h | | |
| | MOVWF | EECON2 | | ; write 55h |
| | MOVLW | 0AAh | | |
| | MOVWF | EECON2 | | ; write 0AAh |
| | BSF | EECON1, WR | | ; start erase (CPU stall) |
| | BSF | INTCON, GIE | | ; re-enable interrupts |

6.5 Writing to Flash Program Memory

The minimum programming block is 32 words or 64 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 64 holding registers used by the table writes for programming.

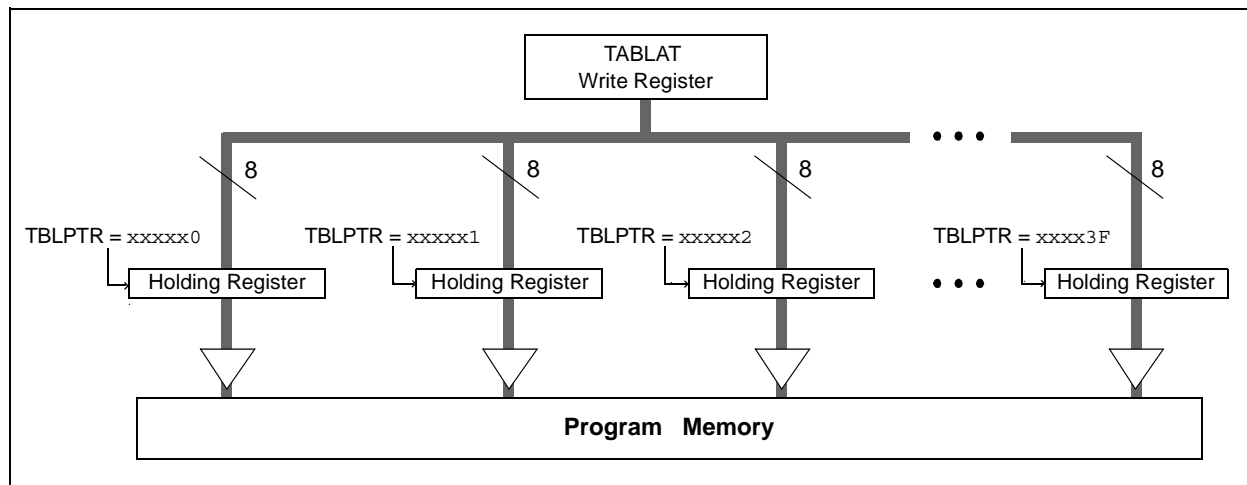
Since the Table Latch (TABLAT) is only a single byte, the TBLWT instruction may need to be executed 64 times for each programming operation. All of the table write operations will essentially be short writes because only the holding registers are written. At the end of updating the 64 holding registers,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

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

The 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 64 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:

1. Read 64 bytes into RAM.
2. Update data values in RAM as necessary.
3. Load Table Pointer register with address being erased.
4. Execute the Row Erase procedure.
5. Load Table Pointer register with address of first byte being written.
6. Write the 64 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.
14. Verify the memory (table read).

This procedure will require about 18 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 64 bytes in the holding register.

PIC18F2455/2550/4455/4550

EXAMPLE 6-3: WRITING TO FLASH PROGRAM MEMORY

```

MOVWL D'64 ; number of bytes in erase block
MOVWF COUNTER
MOVLW BUFFER_ADDR_HIGH ; point to buffer
MOVWF FSR0H
MOVLW BUFFER_ADDR_LOW
MOVWF FSR0L
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 POSTINC0 ; store data
DECFSZ COUNTER ; done?
BRA READ_BLOCK ; repeat

MODIFY_WORD
MOVLW DATA_ADDR_HIGH ; point to buffer
MOVWF FSR0H
MOVLW DATA_ADDR_LOW
MOVWF FSR0L
MOVLW NEW_DATA_LOW ; update buffer word
MOVWF POSTINC0
MOVLW NEW_DATA_HIGH
MOVWF INDF0

ERASE_BLOCK
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
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
Sequence
MOVLW 55h
MOVWF EECON2 ; write 55h
MOVLW 0AAh
MOVWF EECON2 ; write 0AAh
BSF EECON1, WR ; start erase (CPU stall)
BSF INTCON, GIE ; re-enable interrupts
TBLRD*- ; dummy read decrement
MOVLW BUFFER_ADDR_HIGH ; point to buffer
MOVWF FSR0H
MOVLW BUFFER_ADDR_LOW
MOVWF FSR0L

WRITE_BUFFER_BACK
MOVLW D'64 ; number of bytes in holding register
MOVWF COUNTER

WRITE_BYTE_TO_HREGS
MOVWF POSTINC0, 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

```


PIC18F2455/2550/4455/4550

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 Sequence | MOVWF | EECON2 | ; write 55h |
| | MOVLW | 0AAh | |
| | MOVWF | EECON2 | ; write 0AAh |
| | BSF | EECON1, WR | ; start program (CPU stall) |
| | 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 | Program Memory Table Pointer Upper Byte (TBLPTR<20:16>) | | | | | 51 |
| TBPLTRH | Program Memory Table Pointer High Byte (TBLPTR<15:8>) | | | | | | | | 51 |
| TBLPTRL | Program Memory Table Pointer High Byte (TBLPTR<7:0>) | | | | | | | | 51 |
| TABLAT | Program Memory Table Latch | | | | | | | | 51 |
| INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INTE | RBIE | TMR0IF | INTF | RBIF | 51 |
| EECON2 | EEPROM Control Register 2 (not a physical register) | | | | | | | | 53 |
| EECON1 | EEPGD | CFGFS | — | FREE | WRERR | WREN | WR | RD | 53 |
| IPR2 | OSCFIP | CMIP | USBIP | EEIP | BCLIP | LVDIP | TMR3IP | CCP2IP | 54 |
| PIR2 | OSCFIF | CMIF | USBIF | EEIF | BCLIF | LVDIF | TMR3IF | CCP2IF | 54 |
| PIE2 | OSCFIE | CMIE | USBIE | EEIE | BCLIE | LVDIE | TMR3IE | CCP2IE | 54 |

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

PIC18F2455/2550/4455/4550

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.

PIC18F2455/2550/4455/4550

REGISTER 7-1: EECOM1 REGISTER

| 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 | WREN | WR | RD | |
| bit 7 | | | | | | | | bit 0 |

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

| | | | |
|-------------------|------------------|------------------------------------|--------------------|
| 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 |

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  EEADR              ; Lower bits of Data Memory Address to read
BCF    EECON1, EEPGD      ; Point to DATA memory
BCF    EECON1, CFGS       ; Access EEPROM
BSF    EECON1, RD         ; EEPROM Read
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  EEDATA             ; Data Memory Value to write
BCF    EECON1, EPGD       ; Point to DATA memory
BCF    EECON1, CFGS       ; Access EEPROM
BSF    EECON1, WREN       ; Enable writes

BCF    INTCON, GIE        ; Disable Interrupts
MOVLW  55h                ;
Required MOVWF  EECON2      ; Write 55h
Sequence MOVLW  0AAh       ;
MOVWF  EECON2             ; Write 0AAh
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)
    
```

PIC18F2455/2550/4455/4550

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).

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 |
| Required Sequence | MOVLW | 55h | | ; |
| | MOVWF | EECON2 | | ; Write 55h |
| | MOVLW | 0AAh | | ; |
| | MOVWF | EECON2 | | ; Write 0AAh |
| | BSF | EECON1, WR | | ; Set WR bit to begin write |
| | BTFSZ | 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 |

PIC18F2455/2550/4455/4550

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 | INTE | RBIE | TMR0IF | INTF | RBIF | 51 |
| EEADR | EEPROM Address Register | | | | | | | | 53 |
| EEDATA | EEPROM Data Register | | | | | | | | 53 |
| EECON2 | EEPROM Control Register 2 (not a physical register) | | | | | | | | 53 |
| EECON1 | EEPGD | CFGS | — | FREE | WRERR | WREN | WR | RD | 53 |
| IPR2 | OSCFIP | CMIP | USBIP | EEIP | BCLIP | LVDIP | TMR3IP | CCP2IP | 54 |
| PIR2 | OSCFIF | CMIF | USBIF | EEIF | BCLIF | LVDIF | TMR3IF | CCP2IF | 54 |
| PIE2 | OSCFIE | CMIE | USBIE | EEIE | BCLIE | LVDIE | TMR3IE | CCP2IE | 54 |

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

PIC18F2455/2550/4455/4550

NOTES:

PIC18F2455/2550/4455/4550

8.0 8 x 8 HARDWARE MULTIPLIER

8.1 Introduction

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

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

8.2 Operation

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

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

EXAMPLE 8-1: 8 x 8 UNSIGNED MULTIPLY ROUTINE

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

EXAMPLE 8-2: 8 x 8 SIGNED MULTIPLY ROUTINE

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

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

| Routine | Multiply Method | Program Memory (Words) | Cycles (Max) | Time | | |
|------------------|---------------------------|------------------------|--------------|--------------|---------------|-------------|
| | | | | @ 40 MHz | @ 10 MHz | @ 4 MHz |
| 8 x 8 unsigned | Without hardware multiply | 13 | 69 | 6.9 μ s | 27.6 μ s | 69 μ s |
| | Hardware multiply | 1 | 1 | 100 ns | 400 ns | 1 μ s |
| 8 x 8 signed | Without hardware multiply | 33 | 91 | 9.1 μ s | 36.4 μ s | 91 μ s |
| | Hardware multiply | 6 | 6 | 600 ns | 2.4 μ s | 6 μ s |
| 16 x 16 unsigned | Without hardware multiply | 21 | 242 | 24.2 μ s | 96.8 μ s | 242 μ s |
| | Hardware multiply | 28 | 28 | 2.8 μ s | 11.2 μ s | 28 μ s |
| 16 x 16 signed | Without hardware multiply | 52 | 254 | 25.4 μ s | 102.6 μ s | 254 μ s |
| | Hardware multiply | 35 | 40 | 4.0 μ s | 16.0 μ s | 40 μ s |

PIC18F2455/2550/4455/4550

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

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

EXAMPLE 8-3: 16 x 16 UNSIGNED MULTIPLY ROUTINE

```

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

MOVFF PRODH, RES1   ;
MOVFF PRODL, RES0   ;
;

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

MOVFF PRODH, RES3   ;
MOVFF PRODL, RES2   ;
;

MOVF ARG1L, W
MULWF ARG2H          ; ARG1L * ARG2H ->
                      ; PRODH:PRODL

MOVF PRODL, W
ADDWF RES1, F        ; Add cross
MOVF PRODH, W        ; products
ADDWFC RES2, F       ;
CLRF WREG             ;
ADDWFC RES3, F       ;
;

MOVF ARG1H, W
MULWF ARG2L          ; ARG1H * ARG2L ->
                      ; PRODH:PRODL

MOVF PRODL, W
ADDWF RES1, F        ; Add cross
MOVF PRODH, W        ; products
ADDWFC RES2, F       ;
CLRF WREG             ;
ADDWFC RES3, F       ;

```

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

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

EXAMPLE 8-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```

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

MOVFF PRODH, RES1   ;
MOVFF PRODL, RES0   ;
;

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

MOVFF PRODH, RES3   ;
MOVFF PRODL, RES2   ;
;

MOVF ARG1L, W
MULWF ARG2H          ; ARG1L * ARG2H ->
                      ; PRODH:PRODL

MOVF PRODL, W
ADDWF RES1, F        ; Add cross
MOVF PRODH, W        ; products
ADDWFC RES2, F       ;
CLRF WREG             ;
ADDWFC RES3, F       ;
;

MOVF ARG1H, W
MULWF ARG2L          ; ARG1H * ARG2L ->
                      ; PRODH:PRODL

MOVF PRODL, W
ADDWF RES1, F        ; Add cross
MOVF PRODH, W        ; products
ADDWFC RES2, F       ;
CLRF WREG             ;
ADDWFC RES3, F       ;
;

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

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

CONT_CODE
;

```

9.0 INTERRUPTS

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

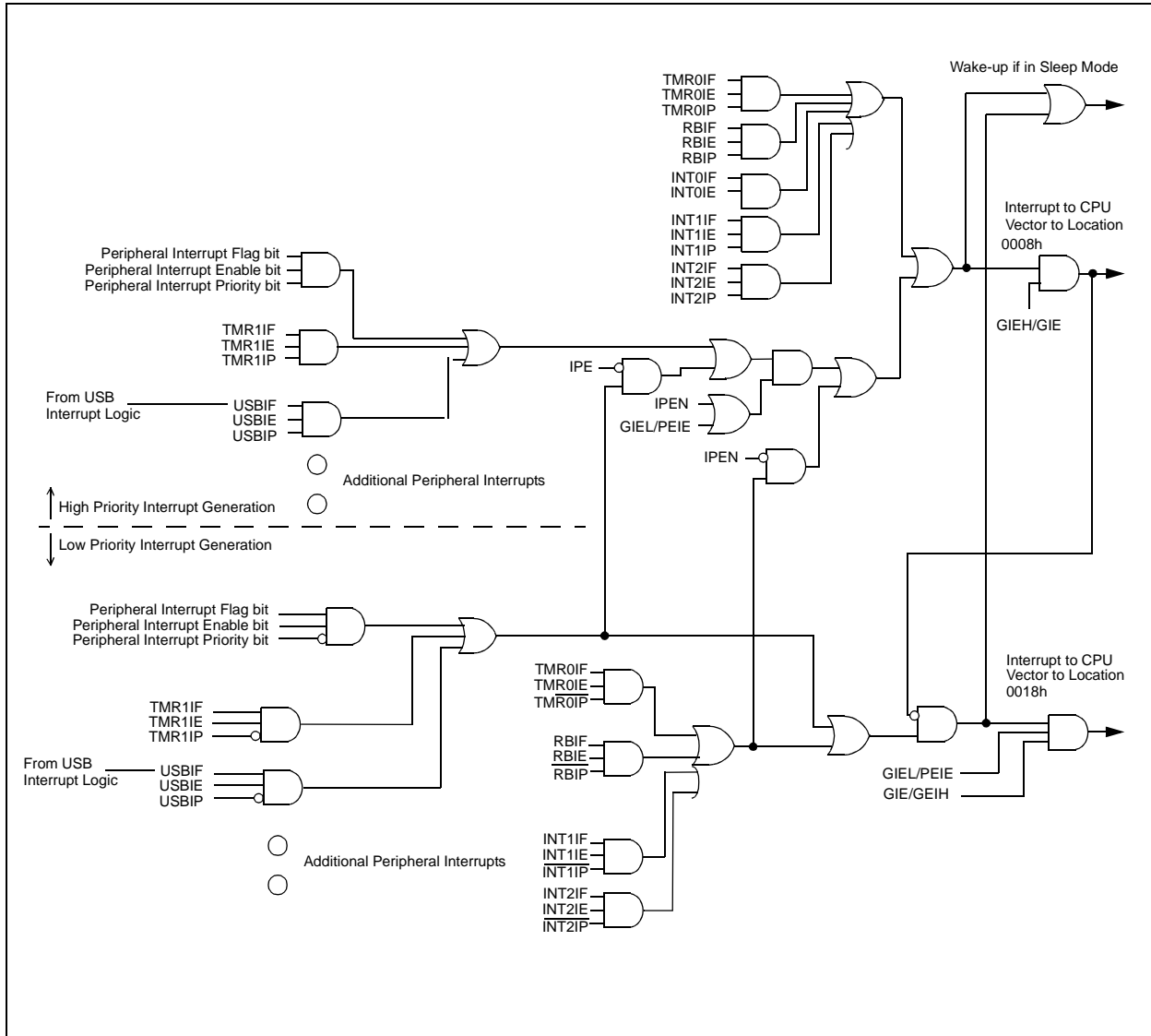
Unlike other peripherals, the USB module is capable of generating a wide range of interrupts for many types of events. These include several types of normal communication and status events and several module level error events.

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”**.

PIC18F2455/2550/4455/4550

FIGURE 9-1: INTERRUPT LOGIC



PIC18F2455/2550/4455/4550

9.2 INTCON Registers

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

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global interrupt enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

REGISTER 9-1: INTCON 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 |
| bit 7 | | | | | | | bit 0 |

- bit 7 **GIE/GIEH:** Global Interrupt Enable bit
When IPEN = 0:
 1 = Enables all unmasked interrupts
 0 = Disables all interrupts
When IPEN = 1:
 1 = Enables all high priority interrupts
 0 = Disables all high priority interrupts
- bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit
When IPEN = 0:
 1 = Enables all unmasked peripheral interrupts
 0 = Disables all peripheral interrupts
When IPEN = 1:
 1 = Enables all low priority peripheral interrupts
 0 = Disables all low priority peripheral interrupts
- bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
 1 = Enables the TMR0 overflow interrupt
 0 = Disables the TMR0 overflow interrupt
- bit 4 **INT0IE:** INT0 External Interrupt Enable bit
 1 = Enables the INT0 external interrupt
 0 = Disables the INT0 external interrupt
- bit 3 **RBIE:** RB Port Change Interrupt Enable bit
 1 = Enables the RB port change interrupt
 0 = Disables the RB port change interrupt
- bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
 1 = TMR0 register has overflowed (must be cleared in software)
 0 = TMR0 register did not overflow
- bit 1 **INT0IF:** INT0 External Interrupt Flag bit
 1 = The INT0 external interrupt occurred (must be cleared in software)
 0 = The INT0 external interrupt did not occur
- bit 0 **RBIF:** RB Port Change Interrupt Flag bit
 1 = 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: A mismatch condition will continue to set this bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared.

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 |

PIC18F2455/2550/4455/4550

REGISTER 9-2: INTCON2 REGISTER

| | | | | | | | |
|--------------------------|---------|---------|---------|-----|--------|-----|-------|
| R/W-1 | R/W-1 | R/W-1 | R/W-1 | U-0 | R/W-1 | U-0 | R/W-1 |
| $\overline{\text{RBPU}}$ | INTEDG0 | INTEDG1 | INTEDG2 | — | TMR0IP | — | RBIP |
| bit 7 | | | | | | | bit 0 |

- bit 7 **$\overline{\text{RBPU}}$** : PORTB Pull-up Enable bit
 1 = All PORTB pull-ups are disabled
 0 = PORTB pull-ups are enabled by individual port latch values
- bit 6 **INTEDG0**: External Interrupt 0 Edge Select bit
 1 = Interrupt on rising edge
 0 = Interrupt on falling edge
- bit 5 **INTEDG1**: External Interrupt 1 Edge Select bit
 1 = Interrupt on rising edge
 0 = Interrupt on falling edge
- bit 4 **INTEDG2**: External Interrupt 2 Edge Select bit
 1 = Interrupt on rising edge
 0 = Interrupt on falling edge
- bit 3 **Unimplemented**: Read as '0'
- bit 2 **TMR0IP**: TMR0 Overflow Interrupt Priority bit
 1 = High priority
 0 = Low priority
- bit 1 **Unimplemented**: Read as '0'
- bit 0 **RBIP**: RB Port Change Interrupt Priority bit
 1 = High priority
 0 = Low priority

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

Note: 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.

PIC18F2455/2550/4455/4550

REGISTER 9-3: INTCON3 REGISTER

| 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 | |

- bit 7 **INT2IP:** INT2 External Interrupt Priority bit
 1 = High priority
 0 = Low priority
- bit 6 **INT1IP:** INT1 External Interrupt Priority bit
 1 = High priority
 0 = Low priority
- bit 5 **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

Legend:

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

Note: 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.

PIC18F2455/2550/4455/4550

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>).

2: User software should ensure the appropriate interrupt flag bits are cleared prior to enabling an interrupt and after servicing that interrupt.

REGISTER 9-4: PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1

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

- bit 7 **SPPIF:** Streaming Parallel Port Read/Write Interrupt Flag bit⁽¹⁾
 1 = A read or a write operation has taken place (must be cleared in software)
 0 = No read or write has occurred
Note 1: This bit is reserved on 28-pin devices; always maintain this bit clear.
- 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 = MR1 register did not overflow

Legend:

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

PIC18F2455/2550/4455/4550

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 | LVDIF | TMR3IF | CCP2IF |
| bit 7 | | | | | | bit 0 | |

- 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 **USBIF:** 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 **LVDIF:** 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 **ECCP1IF:** CCPx Interrupt Flag bit⁽¹⁾
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.

Note 1: This bit is available in 40/44-pin devices only.

| | | | |
|-----------------------|------------------|------------------------------------|--------------------|
| Legend: 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 |

PIC18F2455/2550/4455/4550

9.4 PIE Registers

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

| 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 ⁽¹⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE |
| bit 7 | | | | | | | bit 0 |

- bit 7 **SPPIE:** Streaming Parallel Port Read/Write Interrupt Enable bit⁽¹⁾
1 = Enables the SPP read/write interrupt
0 = Disables the SPP read/write interrupt
Note 1: This bit is reserved on 28-pin devices; always maintain this bit clear.
- bit 6 **ADIE:** A/D Converter Interrupt Enable bit
1 = Enables the A/D interrupt
0 = Disables the A/D interrupt
- bit 5 **RCIE:** EUSART Receive Interrupt Enable bit
1 = Enables the EUSART receive interrupt
0 = Disables the EUSART receive interrupt
- bit 4 **TXIE:** EUSART Transmit Interrupt Enable bit
1 = Enables the EUSART transmit interrupt
0 = Disables the EUSART transmit interrupt
- bit 3 **SSPIE:** Master Synchronous Serial Port Interrupt Enable bit
1 = Enables the MSSP interrupt
0 = Disables the MSSP interrupt
- bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt
- bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt
- bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

Legend:

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

PIC18F2455/2550/4455/4550

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 | LVDIE | TMR3IE | CCP2IE |
| bit 7 | | | | | | bit 0 | |

- bit 7 **OSCFIE:** Oscillator Fail Interrupt Enable bit
1 = Enabled
0 = Disabled
- bit 6 **CMIE:** Comparator Interrupt Enable bit⁽¹⁾
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 **LVDIE:** High/Low-Voltage Detect Interrupt Enable bit
1 = Enabled
0 = Disabled
- bit 1 **TMR3IE:** TMR3 Overflow Interrupt Enable bit
1 = Enabled
0 = Disabled
- bit 0 **CCP2IE:** CCP2 Interrupt Enable bit
1 = Enabled
0 = Disabled

| | | | |
|-------------------|------------------|------------------------------------|--------------------|
| 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 |

PIC18F2455/2550/4455/4550

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 ⁽¹⁾ | ADIP | RCIP | TXIP | SSPIP | CCP1IP | TMR2IP | TMR1IP |
| bit 7 | | | | | | | bit 0 |

- bit 7 **SPPIP:** Streaming Parallel Port Read/Write Interrupt Priority bit⁽¹⁾
1 = High priority
0 = Low priority
Note 1: This bit is reserved on 28-pin devices; always maintain this bit set.
- bit 6 **ADIP:** A/D Converter Interrupt Priority 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
1 = High priority
0 = Low priority
- bit 3 **SSPIP:** Master Synchronous Serial Port Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 2 **CCP1IP:** CCP1 Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 1 **TMR2IP:** TMR2 to PR2 Match Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 0 **TMR1IP:** TMR1 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

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 |

PIC18F2455/2550/4455/4550

REGISTER 9-9: IPR2: PERIPHERAL INTERRUPT PRIORITY REGISTER 2

| R/W-1 | R/W-1 | R/W-0 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|--------|-------|-------|-------|-------|-------|--------|--------|
| OSCFIP | CMIP | USBIP | EEIP | BCLIP | LVDIP | TMR3IP | CCP2IP |
| bit 7 | | | | | | | bit 0 |

- bit 7 **OSCFIP:** Oscillator Fail Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 6 **CMIP:** Comparator Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 5 **USBIP:** USB Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 4 **EEIP:** Data EEPROM/Flash Write Operation Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 3 **BCLIP:** Bus Collision Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 2 **LVDIP:** High/Low-Voltage Detect Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 1 **TMR3IP:** TMR3 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority
- bit 0 **CCP2IP:** CCP2 Interrupt Priority bit
1 = High priority
0 = Low priority

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 |

PIC18F2455/2550/4455/4550

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 REGISTER

| R/W-0 | R/W-1 | U-0 | R/W-1 | R-1 | R-1 | R/W-0 | R/W-0 |
|-------|--------|-----|------------------------|------------------------|------------------------|-------------------------|-------------------------|
| IPEN | SBOREN | — | $\overline{\text{RI}}$ | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | $\overline{\text{POR}}$ | $\overline{\text{BOR}}$ |
| bit 7 | | | | | | | bit 0 |

- 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:** Software BOR Enable bit
For details of bit operation, see Register 4-1.
- bit 5 **Unimplemented:** Read as '0'
- bit 4 **$\overline{\text{RI}}$:** RESET Instruction Flag bit
For details of bit operation, see Register 4-1.
- bit 3 **$\overline{\text{TO}}$:** Watchdog Time-out Flag bit
For details of bit operation, see Register 4-1.
- bit 2 **$\overline{\text{PD}}$:** Power-Down Detection Flag bit
For details of bit operation, see Register 4-1.
- bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit
For details of bit operation, see Register 4-1.
- bit 0 **$\overline{\text{BOR}}$:** Brown-out Reset Status bit
For details of bit operation, see Register 4-1.

Note 1: Actual Reset values are determined by device configuration and the nature of the device Reset. See Register 4-1 for additional information.

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 |

9.7 INTn Pin Interrupts

External interrupts on the RB0/INT0, RB1/INT1 and RB2/INT2 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, INTxF, is set. This interrupt can be disabled by clearing the corresponding enable bit, INTxE. Flag bit, INTxF, must be cleared in software in the Interrupt Service Routine before re-enabling the interrupt.

All external interrupts (INT0, INT1 and INT2) can wake-up the processor from the power managed modes if bit, INTxE, 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 → 00h) will set flag bit TMR0IF. In 16-bit mode, an overflow in the TMR0H:TMR0L register pair (FFFFh → 0000h) will set TMR0IF. The interrupt can be enabled/disabled by setting/clearing enable bit, TMR0IE (INTCON<5>). Interrupt priority for Timer0 is determined by the value contained in the interrupt priority bit, TMR0IP (INTCON2<2>). See **Section 13.0 “Timer2 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_TEMP located anywhere
;
; USER ISR CODE
;
MOVFF    BSR_TEMP, BSR          ; Restore BSR
MOVFF    W_TEMP, W              ; Restore WREG
MOVFF    STATUS_TEMP, STATUS    ; Restore STATUS
```

PIC18F2455/2550/4455/4550

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.

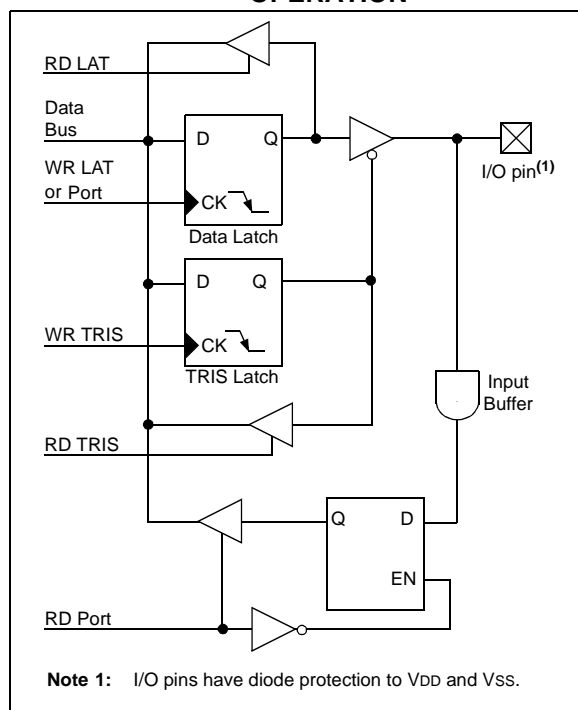
Each port has three registers for its operation. These registers are:

- 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 that the I/O pins are driving.

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 a 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. Pins RA6 and RA7 are multiplexed with the main oscillator pins; they are enabled as oscillator or I/O pins by the selection of the main oscillator in Configuration Register 1H (see **Section 25.1 “Configuration Bits”** for details). When they are not used as port pins, RA6 and RA7 and their 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 RA3:RA0 and RA5 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

```

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
    
```

PIC18F2455/2550/4455/4550

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 | | OUT | ANA | Comparator voltage reference output. Enabling this feature disables digital I/O. |
| RA3/AN3/ VREF+ | RA3 | 0 | OUT | DIG | LATA<3> data output; not affected by analog input. |
| | | 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/ C1OUT/RCV | RA4 | 0 | OUT | DIG | LATA<4> data output; not affected by analog input. |
| | | 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 | | IN | TTL | External USB transceiver RCV input. |
| RA5/AN4/ \overline{SS} / HLVDIN/C2OUT | RA5 | 0 | OUT | DIG | LATA<5> data output; not affected by analog input. |
| | | 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. |
| | \overline{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/ RA6 | 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'. |
| | OSC2 | x | OUT | ANA | Main oscillator feedback output connection (all XT and HS modes). |
| | CLKO | x | OUT | DIG | System cycle clock output (Fosc/4); available in EC, ECPLL and INTCKO modes. |

Legend: OUT = Output, IN = Input, ANA = Analog Signal, DIG = Digital Output, OD = Open-Drain Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input

PIC18F2455/2550/4455/4550

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 ⁽¹⁾ | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | 54 |
| LATA | — | LATA6 ⁽¹⁾ | LATA Data Output Register | | | | | | 54 |
| TRISA | — | TRISA6 ⁽¹⁾ | PORTA Data Direction Register | | | | | | 54 |
| ADCON1 | — | — | VCFG1 | VCFG0 | PCFG3 | PCFG2 | PCFG1 | PCFG0 | 52 |
| CMCON ⁽²⁾ | C2OUT | C1OUT | C2INV | C1INV | CIS | CM2 | CM1 | CM0 | 53 |
| CVRCON ⁽²⁾ | 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'.

2: These registers are unimplemented on 28-pin devices.

PIC18F2455/2550/4455/4550

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 $\overline{\text{RBP}}\text{U}$ (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, P $\overline{\text{B}}\text{ADEN}$ (CONFIG3H<1>), RB4:RB0 will alternatively be configured as digital inputs on POR.

Four of the PORTB pins (RB7:RB4) have an interrupt-on-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 interrupt-on-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.
- 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
              ; P $\overline{\text{B}}\text{ADEN}$  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
```

PIC18F2455/2550/4455/4550

TABLE 10-3: PORTB I/O SUMMARY

| Pin | Function | TRIS Setting | I/O | I/O Type | Description |
|------------------------------------|---------------------|--------------|----------------------|---|---|
| RB0/AN12/ INT0/FLT0/ SDI/SDA | RB0 | 0 | OUT | DIG | LATB<0> data output; not affected by analog input. |
| | | 1 | IN | TTL | PORTB<0> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. Disabled when analog input enabled ⁽¹⁾ . |
| | AN12 | 1 | IN | ANA | A/D input channel 12 ⁽¹⁾ . |
| | 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 ² C™ data output (MSSP module); takes priority over port data. | |
| | 1 | IN | I ² C/SMB | I ² C data input (MSSP module); input type depends on module setting. | |
| RB1/AN10/ INT1/SCK/ SCL | RB1 | 0 | OUT | DIG | LATB<1> data output; not affected by analog input. |
| | | 1 | IN | TTL | PORTB<1> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. Disabled when analog input enabled ⁽¹⁾ . |
| | AN10 | 1 | IN | ANA | A/D input channel 10 ⁽¹⁾ . |
| | 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 | 0 | OUT | DIG | I ² C clock output (MSSP module); takes priority over port data. | |
| | 1 | IN | I ² C/SMB | I ² C clock input (MSSP module); input type depends on module setting. | |
| RB2/AN8/ INT2/VMO | RB2 | 0 | OUT | DIG | LATB<2> data output; not affected by analog input. |
| | | 1 | IN | TTL | PORTB<2> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. Disabled when analog input enabled ⁽¹⁾ . |
| | AN8 | 1 | IN | ANA | A/D input channel 8 ⁽¹⁾ . |
| | INT2 | 1 | IN | ST | External Interrupt 2 input. |
| VMO | | OUT | DIG | External USB transceiver D- data output. | |
| RB3/AN9/ CCP2/VPO | RB3 | 0 | OUT | DIG | LATB<3> data output; not affected by analog input. |
| | | 1 | IN | TTL | PORTB<3> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. Disabled when analog input enabled ⁽¹⁾ . |
| | AN9 | 1 | IN | ANA | A/D input channel 9 ⁽¹⁾ . |
| | CCP2 ⁽²⁾ | 0 | OUT | DIG | CCP2 Compare and PWM output. |
| 1 | | IN | ST | CCP2 Capture input. | |
| VPO | | OUT | DIG | External USB transceiver D+ data output. | |
| RB4/AN11/ KB10/CSSPP | RB4 | 0 | OUT | DIG | LATB<4> data output; not affected by analog input. |
| | | 1 | IN | TTL | PORTB<4> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. Disabled when analog input enabled ⁽¹⁾ . |
| | AN11 | 1 | IN | ANA | A/D input channel 11 ⁽¹⁾ . |
| | KB10 | 1 | IN | TTL | Interrupt on pin change. |
| CSSPP ⁽⁴⁾ | | IN | DIG | SPP chip select control output. | |
| RB5/KB11/ PGM | RB5 | 0 | OUT | DIG | LATB<5> data output. |
| | | 1 | IN | TTL | PORTB<5> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. |
| | KB11 | 1 | IN | TTL | Interrupt on pin change. |
| | PGM | x | IN | ST | Single-Supply Programming mode entry (ICSP). Enabled by LVP configuration bit; all other pin functions disabled. |

Legend: PWR = Power Supply, OUT = Output, IN = Input, ANA = Analog Signal, DIG = Digital Output, OD = Open-Drain Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input

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

2: Alternate pin assignment for CCP2, when CCP2MX = 0. Default assignment is RC3.

3: All other pin functions are disabled when ICSP™ or ICD operation are enabled.

4: 40/44-pin devices only.

PIC18F2455/2550/4455/4550

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

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

Legend: PWR = Power Supply, OUT = Output, IN = Input, ANA = Analog Signal, DIG = Digital Output, OD = Open-Drain Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input

- Note 1:** Configuration on POR is determined by PBDEN configuration bit. Pins are configured as analog inputs when PBDEN is set and digital inputs when PBDEN is cleared.
- 2:** Alternate pin assignment for CCP2, when CCP2MX = 0. Default assignment is RC3.
- 3:** All other pin functions are disabled when ICSP™ or ICD operation are 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 | LATB Data Output Register (Read and Write to Data Latch) | | | | | | | | 54 |
| TRISB | PORTB Data Direction Control Register | | | | | | | | 54 |
| INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 51 |
| INTCON2 | $\overline{\text{RBPU}}$ | INTEDG0 | INTEDG1 | INTEDG2 | — | TMR0IP | — | RBIP | 51 |
| INTCON3 | INT2IP | INT1IP | — | INT2IE | INT1IE | — | INT2IF | INT1IF | 51 |
| ADCON1 | — | — | VCFG1 | VCFG0 | PCFG3 | PCFG2 | PCFG1 | PCFG0 | 52 |
| SPPCON ⁽¹⁾ | — | — | — | — | — | — | SPPOWN | SPPEN | 55 |
| SPPCFG ⁽¹⁾ | 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 communications modules, including the USART, 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 on-chip 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 are configured as digital inputs.

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
```


PIC18F2455/2550/4455/4550

TABLE 10-5: PORTC I/O SUMMARY

| Pin | Function | TRIS Setting | I/O | I/O Type | Description |
|------------------------|---------------------|------------------|-----|--|---|
| RC0/T1OSO/ T13CKI | RC0 | 0 | OUT | DIG | LATC<0> data output. |
| | | 1 | IN | ST | PORTC<0> data input. |
| | T1OSO | x | OUT | ANA | Timer1 oscillator output; enabled when Timer1 oscillator enabled. Disables digital I/O. |
| | T13CKI | 1 | IN | ST | Timer1/Timer3 counter input. |
| RC1/T1OSI/ CCP2/UOE | RC1 | 0 | OUT | DIG | LATC<1> data output. |
| | | 1 | IN | ST | PORTC<1> data input. |
| | T1OSI | x | IN | ANA | Timer1 oscillator input; enabled when Timer1 oscillator enabled. Disables digital I/O. |
| | CCP2 ⁽¹⁾ | 0 | OUT | DIG | CCP2 Compare and PWM output; takes priority over port data. |
| | | 1 | IN | ST | CCP2 Capture input. |
| | \overline{UOE} | | OUT | DIG | External USB transceiver \overline{OE} output. |
| RC2/CCP1/ P1A | RC2 | 0 | OUT | DIG | LATC<2> data output. |
| | | 1 | IN | 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. |
| | P1A ⁽³⁾ | | 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 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 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 | 1 | OUT | DIG | Asynchronous serial transmit data output (USART module); takes priority over port data. User must configure as output. |
| | | 1 | OUT | DIG | Synchronous serial clock output (USART module); takes priority over port data. |
| | CK | 1 | IN | ST | Synchronous serial clock input (USART module). |
| 1 | | IN | ST | Synchronous serial clock input (USART module). | |
| RC7/RX/DT/ SDO | RC7 | 0 | OUT | DIG | LATC<7> data output. |
| | | 1 | IN | ST | PORTC<7> data input. |
| | RX | 1 | IN | ST | Asynchronous serial receive data input (USART module). |
| | DT | 1 | OUT | DIG | Synchronous serial data output (USART module); takes priority over port data. |
| | | 1 | IN | ST | Synchronous serial data input (USART module). User must configure as an input. |
| | SDO | 0 | OUT | DIG | SPI data output (MSSP module). |

Legend: PWR = Power Supply, OUT = Output, IN = Input, ANA = Analog Signal, DIG = Digital Output, OD = Open-Drain Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input, XCVR = USB transceiver.

Note 1: Default pin assignment. Alternate pin assignment is RB2 (when CCP2MX = 0).

Note 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.

Note 3: 40/44-pin devices only.

PIC18F2455/2550/4455/4550

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 | RC4 | — | 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.

PIC18F2455/2550/4455/4550

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 SSP functions of PORTD are automatically disabled.

EXAMPLE 10-4: INITIALIZING PORTD

```
CLRF   PORTD   ; Initialize PORTD by
              ; clearing output
              ; data latches
CLRF   LATD    ; Alternate method
              ; to clear output
              ; data latches
MOVLW  0CFh   ; Value used to
              ; initialize data
              ; direction
MOVWF  TRISD   ; Set RD<3:0> as inputs
              ; RD<5:4> as outputs
              ; RD<7:6> as inputs
```

PIC18F2455/2550/4455/4550

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. |
| | SPP0 | | OUT | DIG | SPP<0> output data; takes priority over port data. |
| | | | IN | TTL | SPP<0> input data. |
| RD1/SPP1 | RD1 | 0 | OUT | DIG | LATD<1> data output. |
| | | 1 | IN | ST | PORTD<1> data input. |
| | SPP1 | | OUT | DIG | SPP<1> output data; takes priority over port data. |
| | | | IN | TTL | SPP<1> input data. |
| RD2/SPP2 | RD2 | 0 | OUT | DIG | LATD<2> data output. |
| | | 1 | IN | ST | PORTD<2> data input. |
| | SPP2 | | OUT | DIG | SPP<2> output data; takes priority over port data. |
| | | | IN | TTL | SPP<2> input data. |
| RD3/SPP3 | RD3 | 0 | OUT | DIG | LATD<3> data output. |
| | | 1 | IN | ST | PORTD<3> data input. |
| | SPP3 | | OUT | DIG | SPP<3> output data; takes priority over port data. |
| | | | IN | TTL | SPP<3> input data. |
| RD4/SPP4 | RD4 | 0 | OUT | DIG | LATD<4> data output. |
| | | 1 | IN | ST | PORTD<4> data input. |
| | SPP4 | | OUT | DIG | SPP<4> output data; takes priority over port data. |
| | | | IN | TTL | SPP<4> input data. |
| RD5/SPP5/P1B | RD5 | 0 | OUT | DIG | LATD<5> data output |
| | | 1 | IN | ST | PORTD<5> data input |
| | SPP5 | | OUT | DIG | SPP<5> output data; takes priority over port data. |
| | | | 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 | | OUT | DIG | SPP<6> output data; takes priority over port data. |
| | | | IN | TTL | SPP<6> input data. |
| | P1C | 0 | OUT | DIG | 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 | | OUT | DIG | SPP<7> output data; takes priority over port data. |
| | | | IN | TTL | SPP<7> input data. |
| | P1D | 0 | OUT | DIG | ECCP1 Enhanced PWM output, channel D; takes priority over port and SPP data ⁽¹⁾ . |

Legend: PWR = Power Supply, OUT = Output, IN = Input, ANA = Analog Signal, DIG = Digital Output, OD = Open-Drain Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input

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

PIC18F2455/2550/4455/4550

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 | RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | RD0 | 54 |
| LATD | LATD Data Output Register | | | | | | | | 54 |
| TRISD | PORTD Data Direction Register | | | | | | | | 54 |
| PORTE | RDPU | — | — | — | RE3 ^(1,2) | RE2 ⁽²⁾ | RE1 ⁽²⁾ | RE0 ⁽²⁾ | 54 |
| CCP1CON | EPWM1M1 | EPWM1M0 | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | 53 |
| SPPCON | — | — | — | — | — | — | 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).

Note 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).

Note 3: These bits are available on 40/44-pin devices only.

PIC18F2455/2550/4455/4550

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 ($\overline{\text{MCLR}}/\text{VPP}/\text{RE3}$) 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-0 | U-0 | U-0 | U-0 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
| | RDPU | — | — | — | RE3 ⁽¹⁾ | RE2 ⁽²⁾ | RE1 ⁽²⁾ | RE0 ⁽²⁾ |
| bit 7 | | | | | | | | bit 0 |

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^(1,2)

Note 1: Implemented only when Master Clear functionality is disabled (MCLRE configuration bit = 0); otherwise, read as '0'.

2: Unimplemented in 28-pin devices; read as '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 |

PIC18F2455/2550/4455/4550

TABLE 10-9: PORTE I/O SUMMARY

| Pin | Function | TRIS Setting | I/O | I/O Type | Description |
|--------------------|----------|--------------|-----|----------|--|
| RE0/AN5/ CK1SPP | RE0 | 0 | OUT | DIG | LATE<0> data output; not affected by analog input. |
| | | 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 | | OUT | DIG | SPP clock 1 output (SPP enabled). |
| RE1/AN6/ CK2SPP | RE1 | 0 | OUT | DIG | LATE<1> data output; not affected by analog input. |
| | | 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 | | OUT | DIG | SPP clock 2 output (SPP enabled). |
| RE2/AN7/ OESPP | RE2 | 0 | OUT | DIG | LATE<2> data output; not affected by analog input. |
| | | 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 | | OUT | DIG | SPP output enable output (SPP enabled). |
| MCLR/VPP/ RE3 | RE3 | — (1) | IN | ST | PORTE<3> data input; enabled when MCLRE configuration bit is clear. |
| | MCLR | —(1) | IN | 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. |

Legend: OUT = Output, IN = Input, ANA = Analog Signal, DIG = Digital Output, OD = Open-Drain Output, ST = Schmitt Buffer Input, TTL = TTL 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 | — | — | — | RE3 ^(1,2) | RE2 ⁽²⁾ | RE1 ⁽²⁾ | RE0 ⁽²⁾ | 54 |
| LATE | — | — | — | — | — | LATE Data Output Register ⁽²⁾ | | | 54 |
| TRISE | — | — | — | — | — | TRISE Data Direction Register ⁽²⁾ | | | 54 |
| ADCON1 | — | — | VCFG1 | VCFG0 | PCFG3 | PCFG2 | PCFG1 | PCFG0 | 52 |
| CMCON | C2OUT | C1OUT | C2INV | C1INV | CIS | CM2 | CM1 | CM0 | 53 |
| SPPCON ⁽³⁾ | — | — | — | — | — | — | SPPOWN | SPPEN | 55 |
| SPPCFG ⁽³⁾ | 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 are unimplemented on 28-pin devices.

PIC18F2455/2550/4455/4550

11.0 TIMER0 MODULE

The Timer0 module incorporates the following features:

- Software selectable operation as a timer or counter in both 8-bit or 16-bit modes
- Readable and writable registers
- Dedicated 8-bit, software programmable prescaler
- Selectable clock source (internal or external)
- Edge select for external clock
- Interrupt-on-overflow

The T0CON register (Register 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 timer module in 16-bit mode.

REGISTER 11-1: T0CON: TIMER0 CONTROL REGISTER

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|--------|--------|-------|-------|-------|-------|-------|-------|
| TMR0ON | T08BIT | T0CS | T0SE | PSA | T0PS2 | T0PS1 | T0PS0 |
| | | | | | | | bit 0 |

- bit 7 **TMR0ON:** Timer0 On/Off Control bit
 1 = Enables Timer0
 0 = Stops Timer0
- bit 6 **T08BIT:** Timer0 8-bit/16-bit Control bit
 1 = Timer0 is configured as an 8-bit timer/counter
 0 = Timer0 is configured as a 16-bit timer/counter
- bit 5 **T0CS:** Timer0 Clock Source Select bit
 1 = Transition on T0CKI pin
 0 = Internal instruction cycle clock (CLKO)
- bit 4 **T0SE:** Timer0 Source Edge Select bit
 1 = Increment on high-to-low transition on T0CKI pin
 0 = Increment on low-to-high transition on T0CKI pin
- bit 3 **PSA:** Timer0 Prescaler Assignment bit
 1 = Timer0 prescaler is NOT assigned. Timer0 clock input bypasses prescaler.
 0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.
- bit 2-0 **T0PS2:T0PS0:** Timer0 Prescaler Select bits
 111 = 1:256 Prescale value
 110 = 1:128 Prescale value
 101 = 1:64 Prescale value
 100 = 1:32 Prescale value
 011 = 1:16 Prescale value
 010 = 1:8 Prescale value
 001 = 1:4 Prescale value
 000 = 1:2 Prescale value

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 |

PIC18F2455/2550/4455/4550

11.1 Timer0 Operation

Timer0 can operate as either a timer or a counter; the mode is selected by clearing the T0CS bit (T0CON<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: TIMER0 BLOCK DIAGRAM (8-BIT MODE)

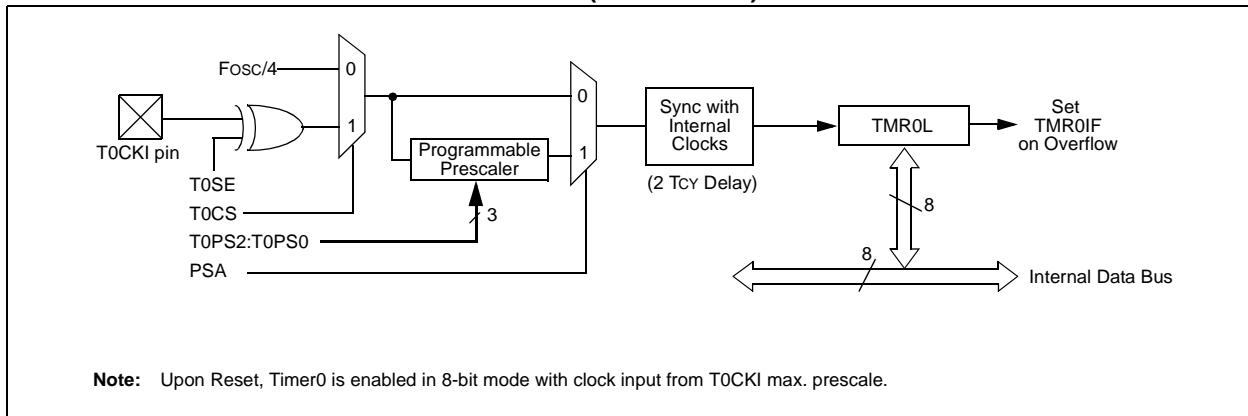
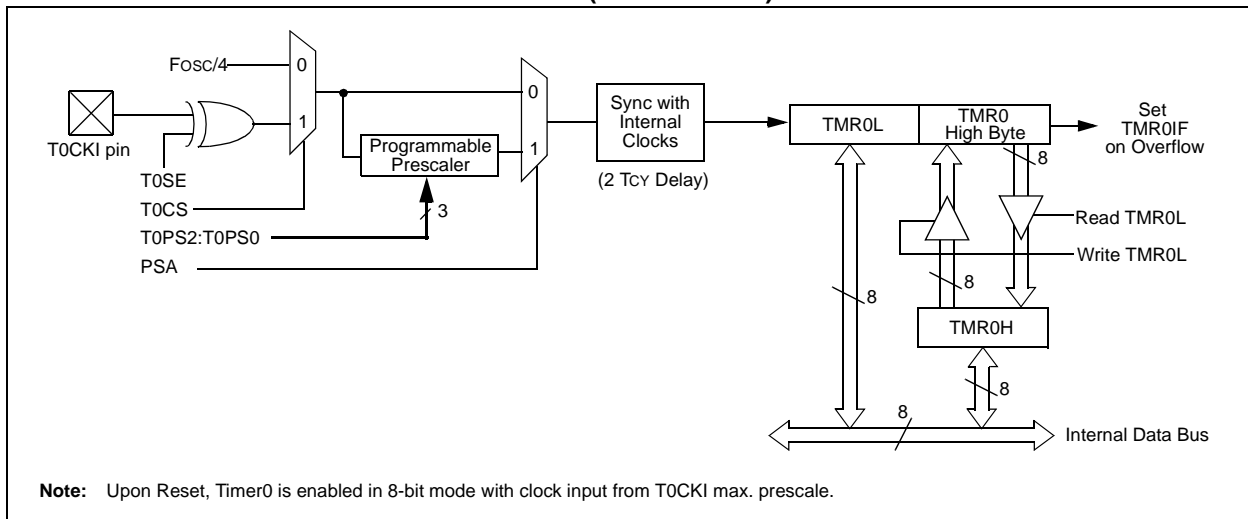


FIGURE 11-2: TIMER0 BLOCK DIAGRAM (16-BIT MODE)



PIC18F2455/2550/4455/4550

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 re-enabling the interrupt, the TMR0IF bit must be cleared in software by the Interrupt Service Routine.

Since Timer0 is shut down in Sleep mode, the TMR0 interrupt cannot awaken the processor from Sleep.

TABLE 11-1: REGISTERS ASSOCIATED WITH TIMER0

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on page |
|--------|----------------------------------|-------------------------------|--------|--------|-------|--------|--------|-------|----------------------|
| TMR0L | Timer0 Module Low Byte Register | | | | | | | | 52 |
| TMR0H | Timer0 Module High Byte Register | | | | | | | | 52 |
| INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 51 |
| T0CON | TMR0ON | T08BIT | T0CS | T0SE | PSA | T0PS2 | T0PS1 | T0PS0 | 52 |
| TRISA | — | PORTA Data Direction Register | | | | | | | 54 |

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

PIC18F2455/2550/4455/4550

NOTES:

PIC18F2455/2550/4455/4550

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 (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
- 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 | TMR1ON |
| bit 7 | | | | | | bit 0 | |

- bit 7 **RD16:** 16-bit Read/Write Mode Enable bit
 1 = Enables register read/write of Timer1 in one 16-bit operation
 0 = Enables register read/write of Timer1 in two 8-bit operations
- bit 6 **T1RUN:** Timer1 System Clock Status bit
 1 = Device clock is derived from Timer1 oscillator
 0 = Device clock is derived from another source
- bit 5-4 **T1CKPS1:T1CKPS0:** Timer1 Input Clock Prescale Select bits
 11 = 1:8 Prescale value
 10 = 1:4 Prescale value
 01 = 1:2 Prescale value
 00 = 1:1 Prescale value
- bit 3 **T1OSCEN:** Timer1 Oscillator Enable bit
 1 = Timer1 oscillator is enabled
 0 = Timer1 oscillator is shut off
 The oscillator inverter and feedback resistor are turned off to eliminate power drain.
- bit 2 **T1SYNC:** Timer1 External Clock Input Synchronization Select bit
 When TMR1CS = 1:
 1 = Do not synchronize external clock input
 0 = Synchronize external clock input
 When TMR1CS = 0:
 This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.
- bit 1 **TMR1CS:** Timer1 Clock Source Select bit
 1 = External clock from pin RC0/T1OSO/T13CKI (on the rising edge)
 0 = Internal clock (FOSC/4)
- bit 0 **TMR1ON:** Timer1 On bit
 1 = Enables Timer1
 0 = Stops Timer1

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 |

PIC18F2455/2550/4455/4550

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, TMR3CS (T3CON<1>). When TMR3CS is cleared (= 0), Timer3 increments on every internal instruction

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

When Timer1 is enabled, the RC1/T1OSI and RC0/T1OSO/T13CKI pins become inputs. This means the values of TRISC<1:0> are ignored and the pins are read as '0'.

FIGURE 12-1: TIMER1 BLOCK DIAGRAM

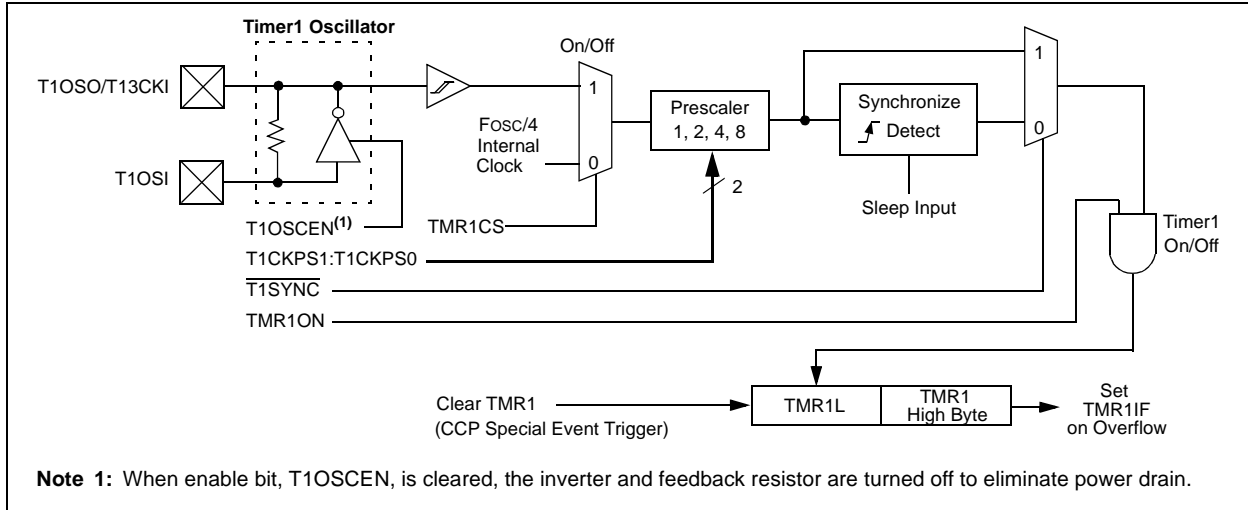
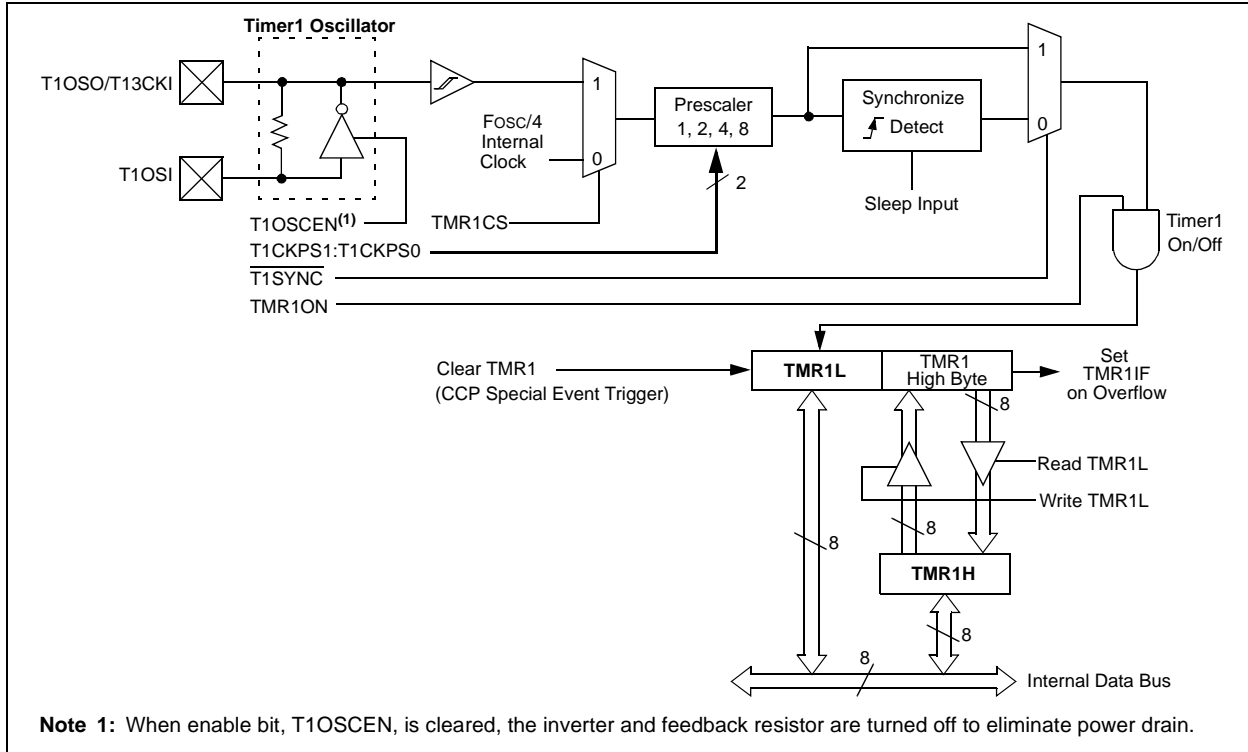


FIGURE 12-2: TIMER1 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)



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

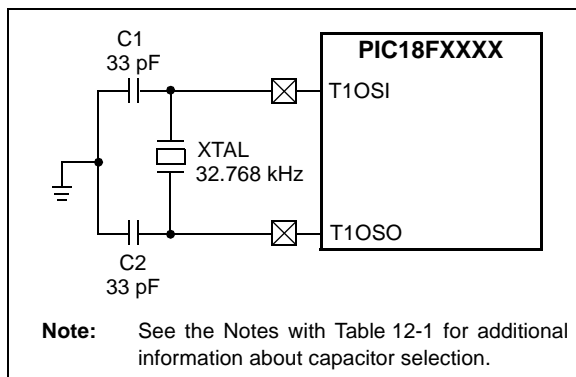


TABLE 12-1: CAPACITOR SELECTION FOR THE TIMER OSCILLATOR

| Osc Type | Freq | C1 | C2 |
|----------|--------|----------------------|----------------------|
| LP | 32 kHz | 27 pF ⁽¹⁾ | 27 pF ⁽¹⁾ |

Note 1: Microchip suggests these values as a starting point in validating the oscillator circuit.

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

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

4: Capacitor values are for design guidance only.

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.

PIC18F2455/2550/4455/4550

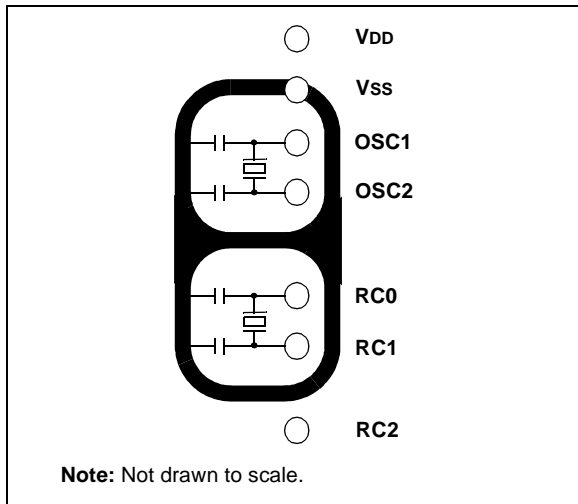
12.3.3 TIMER1 OSCILLATOR LAYOUT CONSIDERATIONS

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”** above), 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 overflow.

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.

PIC18F2455/2550/4455/4550

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

```

RTCinit
    MOVLW    80h                ; Preload TMR1 register pair
    MOVWF   TMR1H              ; for 1 second overflow
    CLRF    TMR1L
    MOVLW   b'00001111'       ; Configure for external clock,
    MOVWF   T1OSC              ; Asynchronous operation, external oscillator
    CLRF    secs               ; Initialize timekeeping registers
    CLRF    mins
    MOVLW   d'12'
    MOVWF   hours
    BSF     PIE1, TMR1IE      ; Enable Timer1 interrupt
    RETURN

RTCisr
    BSF     TMR1H, 7          ; Preload for 1 sec overflow
    BCF     PIR1, TMR1IF     ; Clear interrupt flag
    INCF    secs, F          ; Increment seconds
    MOVLW   d'59'           ; 60 seconds elapsed?
    CPFSGT  secs
    RETURN                    ; No, done
    CLRF    secs            ; Clear seconds
    INCF    mins, F         ; Increment minutes
    MOVLW   d'59'           ; 60 minutes elapsed?
    CPFSGT  mins
    RETURN                    ; No, done
    CLRF    mins           ; clear minutes
    INCF    hours, F        ; Increment hours
    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 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 54 |
| PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 54 |
| IPR1 | PSPIP ⁽¹⁾ | ADIP | RCIP | TXIP | SSPIP | CCP1IP | TMR2IP | TMR1IP | 54 |
| TMR1L | Timer1 Register, Low Byte | | | | | | | | 52 |
| TMR1H | Timer1 Register, High Byte | | | | | | | | 52 |
| T1CON | RD16 | T1RUN | T1CKPS1 | T1CKPS0 | T1OSCEN | T1SYNC | TMR1CS | TMR1ON | 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.

PIC18F2455/2550/4455/4550

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 ($F_{OSC}/4$). 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, \overline{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 |

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

bit 6-3 **T2OUTPS3:T2OUTPS0:** Timer2 Output Postscale Select bits

0000 = 1:1 Postscale

0001 = 1:2 Postscale

-
-
-

1111 = 1:16 Postscale

bit 2 **TMR2ON:** Timer2 On bit

1 = Timer2 is on

0 = Timer2 is off

bit 1-0 **T2CKPS1:T2CKPS0:** Timer2 Clock Prescale Select bits

00 = Prescaler is 1

01 = Prescaler is 4

1x = Prescaler is 16

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

PIC18F2455/2550/4455/4550

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<7:3>).

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

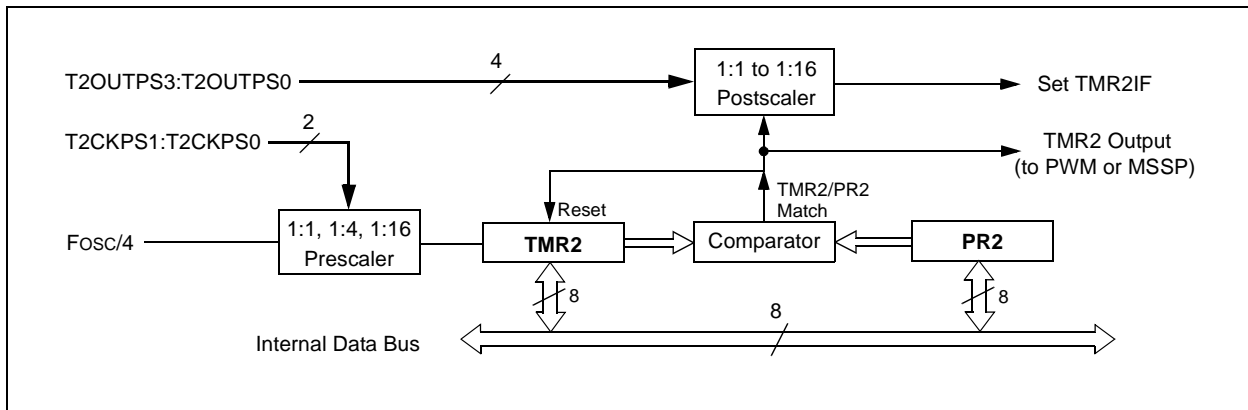


TABLE 13-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

| 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 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 54 |
| PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 54 |
| IPR1 | PSPIP ⁽¹⁾ | ADIP | RCIP | TXIP | SSPIP | CCP1IP | TMR2IP | TMR1IP | 54 |
| TMR2 | Timer2 Register | | | | | | | | 52 |
| T2CON | — | T2OUTPS3 | T2OUTPS2 | T2OUTPS1 | T2OUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | 52 |
| PR2 | Timer2 Period 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.

PIC18F2455/2550/4455/4550

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 | T3ECCP1 ⁽¹⁾ | T3CKPS1 | T3CKPS0 | T3CCP1 ⁽¹⁾ | T3SYNC | TMR3CS | TMR3ON |
| bit 7 | | | | | | | bit 0 |

- 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 **T3ECCP1:T3CCP1:** Timer3 and Timer1 to CCPx Enable bits⁽¹⁾
 1x = Timer3 is the capture/compare clock source for both CCP and ECCP modules
 01 = Timer3 is the capture/compare clock source for ECCP1;
 Timer1 is the capture/compare clock source for CCP1
 00 = Timer1 is the capture/compare clock source for both CCP and ECCP modules
- Note 1:** These bits are available on 40/44-pin devices only.
- 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

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 |

PIC18F2455/2550/4455/4550

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 ($F_{OSC}/4$). When the bit is set, Timer3 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled.

As with Timer1, the RC1/T1OSI and RC0/T1OSO/T13CKI pins become inputs when the Timer1 oscillator is enabled. This means the values of TRISC<1:0> are ignored and the pins are read as '0'.

FIGURE 14-1: TIMER3 BLOCK DIAGRAM

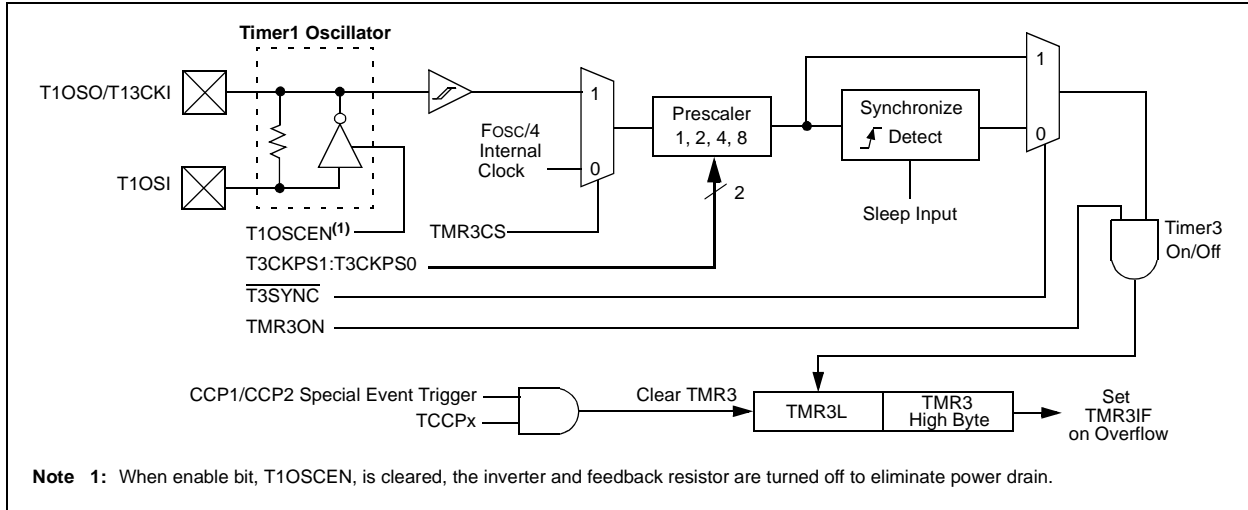
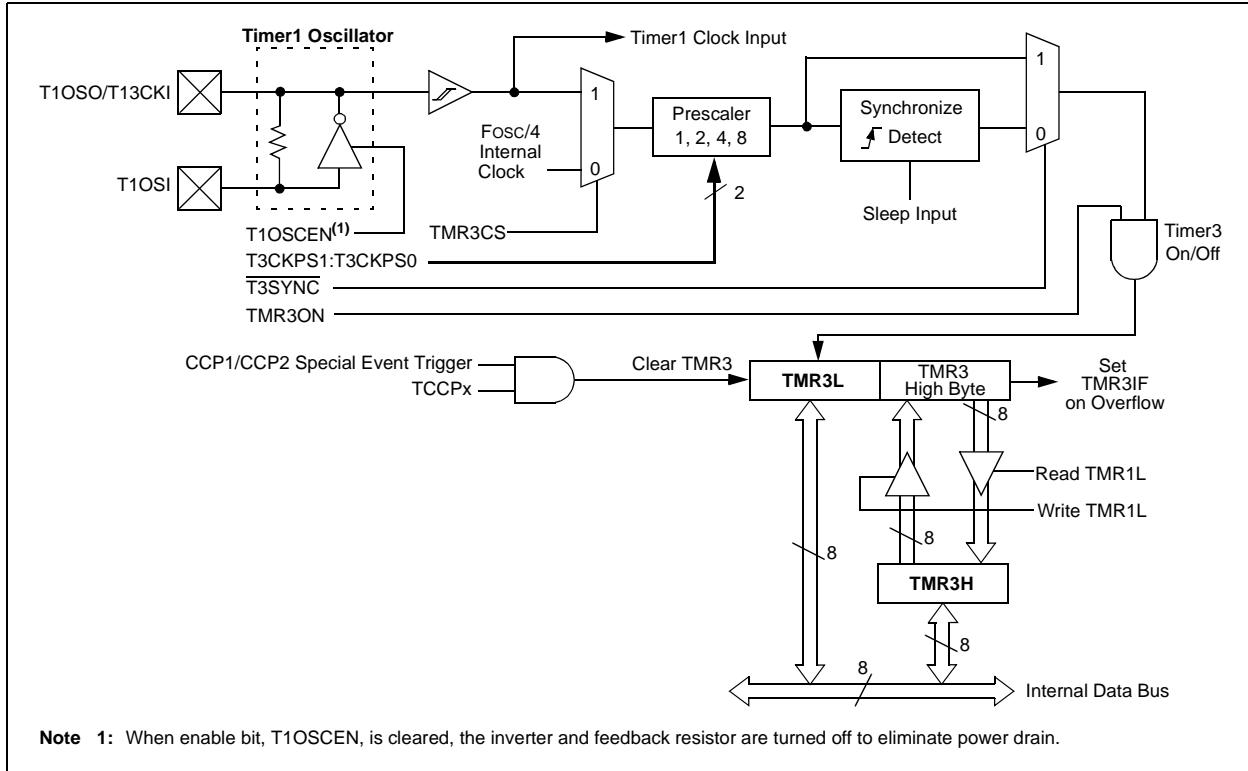


FIGURE 14-2: TIMER3 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)



PIC18F2455/2550/4455/4550

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 Timer3 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 (ECCP1M3:ECCP1M0 = 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 ECCPR2H:ECCPR2L 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 (PIR1<0>).

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 ⁽²⁾ | — | EEIF | BCLIF | LVDIF | TMR3IF | ECCP1IF ⁽²⁾ | 54 |
| PIE2 | OSCFIE | CMIE ⁽²⁾ | — | EEIE | BCLIE | LVDIE | TMR3IE | ECCP1IE ⁽²⁾ | 54 |
| IPR2 | OSCFIP | CMIP ⁽²⁾ | — | EEIP | BCLIP | LVDIP | TMR3IP | ECCP1IP ⁽²⁾ | 54 |
| TMR3L | Timer3 Register Low Byte | | | | | | | | 53 |
| TMR3H | Timer3 Register High Byte | | | | | | | | 53 |
| T1CON | RD16 | T1RUN | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{T1SYNC}$ | TMR1CS | TMR1ON | 52 |
| T3CON | RD16 | T3ECCP1 ⁽¹⁾ | T3CKPS1 | T3CKPS0 | T3CCP1 ⁽¹⁾ | $\overline{T3SYNC}$ | TMR3CS | TMR3ON | 53 |

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

Note 1: These bits are available on 40/44-pin devices only.

2: These bits are available on 40/44-pin devices and reserved in 28-pin devices.

PIC18F2455/2550/4455/4550

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 REGISTER (CCP2 MODULE, CCP1 MODULE IN 28-PIN DEVICES)

| | | | | | | | | |
|-------|-----|-------|-------|--------|--------|--------|--------|-------|
| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | |
| — | — | DCxB1 | DCxB0 | CCPxM3 | CCPxM2 | CCPxM1 | CCPxM0 | |
| bit 7 | | | | | | | | bit 0 |

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-4 **DCxB1:DCxB0:** PWM Duty Cycle bit 1 and bit 0 for CCP Module x

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 (DCx9:DCx2) of the duty cycle are found in CCPRxL.

bit 3-0 **CCPxM3:CCPxM0:** CCP Module x Mode Select bits

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

0001 = Reserved

0010 = Compare mode, toggle output on match (CCPIF 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 CCP pin low; on compare match, force CCP pin high (CCPIF bit is set)

1001 = Compare mode: initialize CCP pin high; on compare match, force CCP pin low (CCPIF bit is set)

1010 = Compare mode: generate software interrupt on compare match (CCPIF bit is set, CCP pin reflects I/O state)

1011 = Compare mode: trigger special event, reset timer, start A/D conversion on CCP2 match (CCPIF bit is set)

11xx = PWM mode

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as ‘0’

-n = Value at POR

‘1’ = Bit is set

‘0’ = Bit is cleared

x = Bit is unknown

PIC18F2455/2550/4455/4550

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 Compare PWM | Timer1 or Timer3 Timer1 or Timer3 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 ⁽¹⁾ | None |
| Compare | PWM ⁽¹⁾ | None |
| PWM ⁽¹⁾ | Capture | None |
| PWM ⁽¹⁾ | Compare | None |
| PWM ⁽¹⁾ | 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 CCPxH: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.

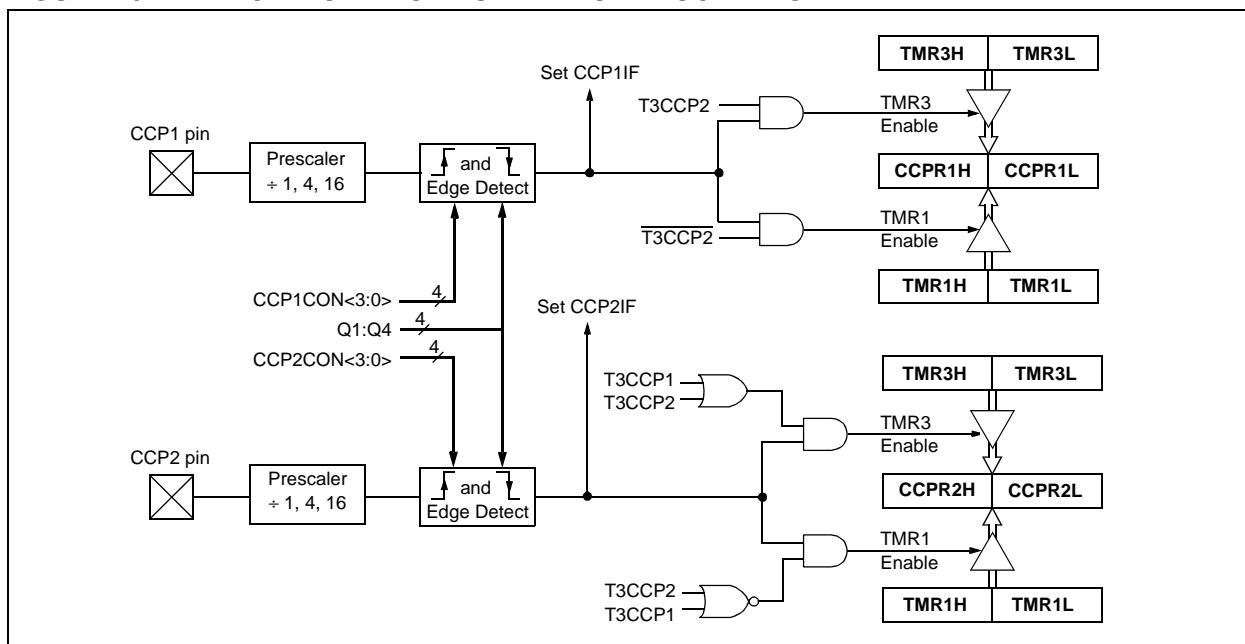
Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared, therefore, the first capture may be from a non-zero prescaler. Example 15-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler 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



PIC18F2455/2550/4455/4550

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.

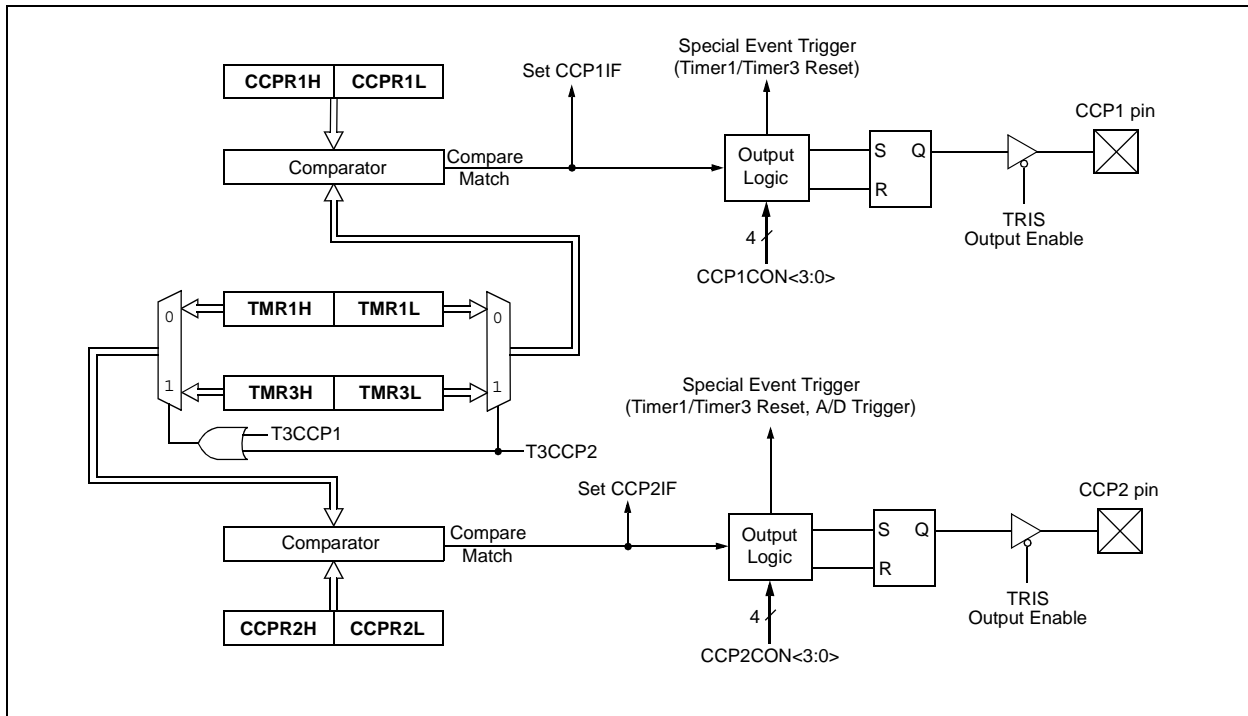
15.3.4 SPECIAL EVENT TRIGGER

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



PIC18F2455/2550/4455/4550

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 | — | \overline{RI} | \overline{TO} | \overline{PD} | \overline{POR} | \overline{BOR} | 52 |
| PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 54 |
| PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 54 |
| IPR1 | PSPIP ⁽¹⁾ | ADIP | RCIP | TXIP | SSPIP | CCP1IP | TMR2IP | TMR1IP | 54 |
| PIR2 | OSCFIE | CMIE | — | EEIE | BCLIF | LVDIF | TMR3IF | CCP2IF | 54 |
| PIE2 | OSCFIF | CMIF | — | EEIF | BCLIE | LVDIE | TMR3IE | CCP2IE | 54 |
| IPR2 | OSCFIP | CMIP | — | EEIP | BCLIP | LVDIP | TMR3IP | CCP2IP | 54 |
| TRISB | PORTB Data Direction Control Register | | | | | | | | 54 |
| TRISC | PORTC Data Direction Control Register | | | | | | | | 54 |
| TMR1L | Timer1 Register Low Byte | | | | | | | | 52 |
| TMR1H | Timer1 Register High Byte | | | | | | | | 52 |
| T1CON | RD16 | T1RUN | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{T1SYNC}$ | TMR1CS | TMR1ON | 52 |
| TMR3H | Timer3 Register High Byte | | | | | | | | 53 |
| TMR3L | Timer3 Register Low Byte | | | | | | | | 53 |
| T3CON | RD16 | T3CCP2 | T3CKPS1 | T3CKPS0 | T3CCP1 | $\overline{T3SYNC}$ | TMR3CS | TMR3ON | 53 |
| CCPR1L | Capture/Compare/PWM Register 1 Low Byte | | | | | | | | 53 |
| CCPR1H | Capture/Compare/PWM Register 1 High Byte | | | | | | | | 53 |
| CCP1CON | P1M1 ⁽¹⁾ | P1M0 ⁽¹⁾ | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | 53 |
| CCPR2L | Capture/Compare/PWM Register 2 Low Byte | | | | | | | | 53 |
| CCPR2H | Capture/Compare/PWM 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: These bits are unimplemented on 28-pin devices; always maintain these bits clear.

PIC18F2455/2550/4455/4550

15.4 PWM Mode

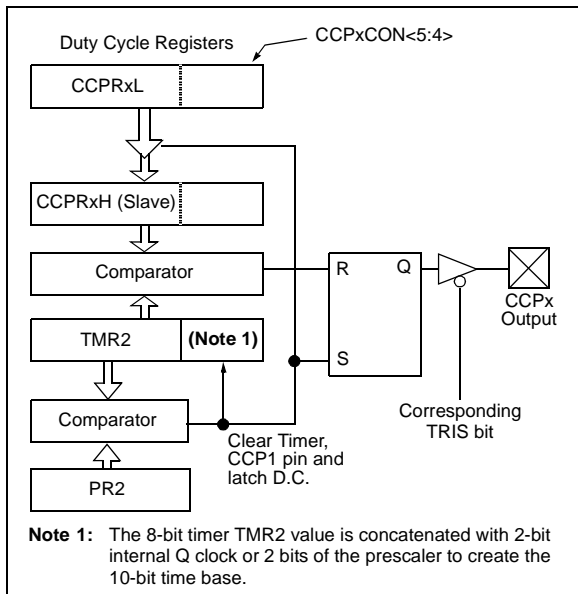
In Pulse-Width Modulation (PWM) mode, the CCPx pin produces up to a 10-bit resolution PWM output. Since the CCP2 pin is multiplexed with a PORTB or PORTC data latch, the appropriate TRIS bit must be cleared to make the CCP2 pin an output.

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.

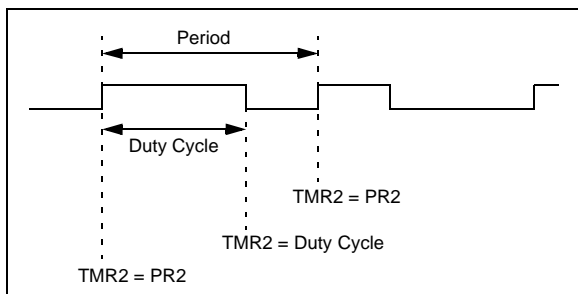
For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 15.4.4 “Setup for PWM Operation”**.

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:

$$\text{PWM Period} = [(PR2) + 1] \cdot 4 \cdot \text{Tosc} \cdot (\text{TMR2 Prescale Value})$$

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 CCP2 pin is set (exception: if PWM duty cycle = 0%, the CCP2 pin will not be set)
- The PWM duty cycle is latched from CCPR2L into CCPR2H

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 CCPR2L register and to the CCPxCON<5:4> bits. Up to 10-bit resolution is available. The CCPRxL contains the eight MSBs and the CCPxCON<5:4> contains the two LSBs. This 10-bit value is represented by CCPRxL:CCPxCON<5:4>. The following equation is used to calculate the PWM duty cycle in time:

EQUATION 15-2:

$$\text{PWM Duty Cycle} = (\text{CCPRxL:CCPxCON<5:4>}) \cdot \text{Tosc} \cdot (\text{TMR2 Prescale Value})$$

CCPRxL and CCPxCON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPR2H until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR2H is a read-only register.

PIC18F2455/2550/4455/4550

The CCPR2H 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 CCP2 pin is cleared.

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

EQUATION 15-3:

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

Note: If the PWM duty cycle value is longer than the PWM period, the CCP2 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:

1. Set the PWM period by writing to the PR2 register.
2. 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.

PIC18F2455/2550/4455/4550

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 | — | \overline{RI} | \overline{TO} | \overline{PD} | \overline{POR} | \overline{BOR} | 52 |
| PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 54 |
| PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 54 |
| IPR1 | PSPIP ⁽¹⁾ | ADIP | RCIP | TXIP | SSPIP | CCP1IP | TMR2IP | TMR1IP | 54 |
| TRISB | PORTB Data Direction Control Register | | | | | | | | 54 |
| TRISC | PORTC Data Direction Control Register | | | | | | | | 54 |
| TMR2 | Timer2 Register | | | | | | | | 52 |
| PR2 | Timer2 Period Register | | | | | | | | 52 |
| T2CON | — | T2OUTPS3 | T2OUTPS2 | T2OUTPS1 | T2OUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | 52 |
| CCPR1L | Capture/Compare/PWM Register 1 Low Byte | | | | | | | | 53 |
| CCPR1H | Capture/Compare/PWM Register 1 High Byte | | | | | | | | 53 |
| CCP1CON | P1M1 ⁽¹⁾ | P1M0 ⁽¹⁾ | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | 53 |
| CCPR2L | Capture/Compare/PWM Register 2 Low Byte | | | | | | | | 53 |
| CCPR2H | Capture/Compare/PWM Register 2 High Byte | | | | | | | | 53 |
| CCP2CON | — | — | DC2B1 | DC2B0 | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | 53 |
| ECCP1AS | ECCPASE | ECCPAS2 | ECCPAS1 | ECCPAS0 | PSSAC1 | PSSAC0 | PSSBD1 ⁽¹⁾ | PSSBD0 ⁽¹⁾ | 53 |
| ECCP1DEL | PRSEN | PDC6 ⁽¹⁾ | PDC5 ⁽¹⁾ | PDC4 ⁽¹⁾ | PDC3 ⁽¹⁾ | PDC2 ⁽¹⁾ | PDC1 ⁽¹⁾ | PDC0 ⁽¹⁾ | 53 |

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

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

PIC18F2455/2550/4455/4550

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 CCPCON registers in PIC18F2255/2550 devices in that the two Most Significant bits are implemented to control PWM functionality.

REGISTER 16-1: CCP1CON REGISTER (ECCP1 MODULE, PIC18F4455/4550 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 |

bit 7-6 **P1M1:P1M0:** Enhanced PWM Output Configuration bits

If ECCP1M3:ECCP1M2 = 00, 01, 10:

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

If ECCP1M3:ECCP1M2 = 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 CCPRxL.

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 ECCP pin low, set output on compare match (set ECCP1IF)

1001 = Compare mode, initialize ECCP pin high, clear output on compare match (set ECCP1IF)

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

1011 = Compare mode, trigger special event (ECCP resets TMR1 or TMR3, sets ECCP1IF 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

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

PIC18F2455/2550/4455/4550

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 EPWM1M1:EPWM1M0 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 ECCP1 module are identical in operation to that of CCP1. 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 ECCP1 resets the TMR1 or TMR3 register pair, depending on which timer resource is currently selected. This allows the ECCP1 register 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 ECCP1 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 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:

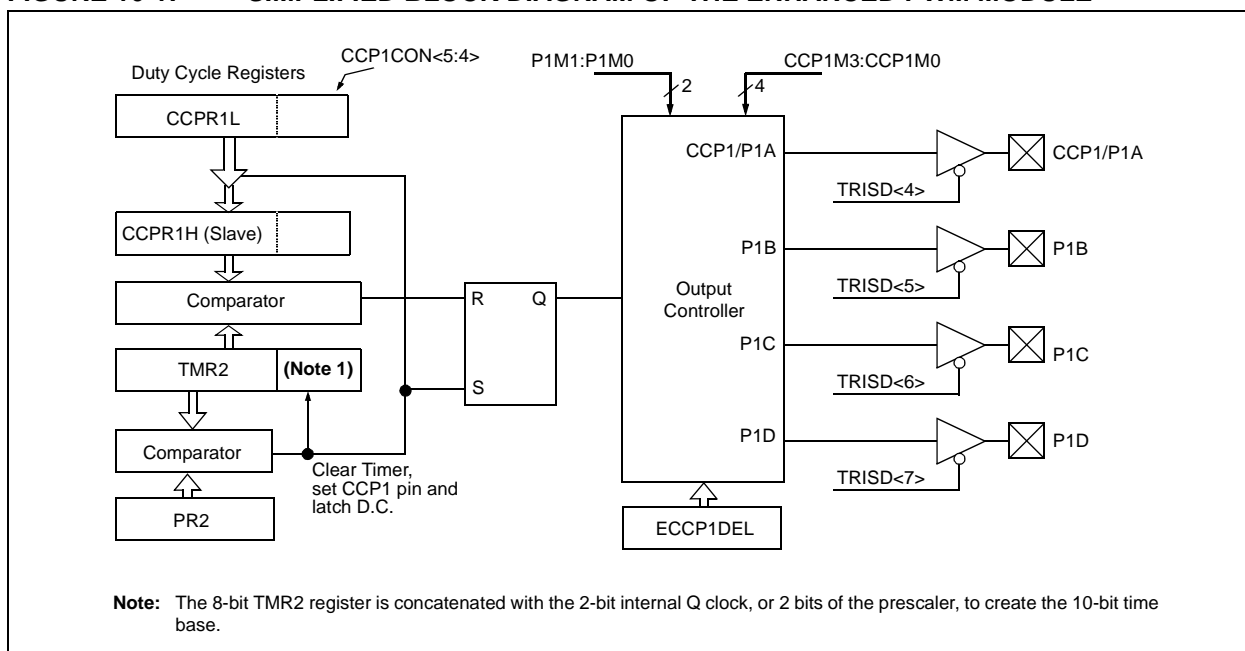
$$\text{PWM Period} = [(\text{PR2}) + 1] \cdot 4 \cdot \text{TOSC} \cdot (\text{TMR2 Prescale Value})$$

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

- TMR2 is cleared
- The CCP1 pin is set (if PWM duty cycle = 0%, the ECP1 pin will not be set)
- The PWM duty cycle is copied from CCPR1L into CCPR1H

Note: 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



PIC18F2455/2550/4455/4550

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:

$$\text{PWM Duty Cycle} = (\text{ECCPR1L:ECCP1CON<5:4>} \cdot \text{Tosc} \cdot (\text{TMR2 Prescale Value}))$$

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

The CCPR1H register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation. When the CCPR1H and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or two bits of the TMR2 prescaler, the CCP1 pin is cleared. The maximum PWM resolution (bits) for a given PWM frequency is given by the following equation.

EQUATION 16-3:

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

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

16.4.3 PWM OUTPUT CONFIGURATIONS

The EPWM1M1:EPWM1M0 bits in the ECCP1CON 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 |

PIC18F2455/2550/4455/4550

FIGURE 16-2: PWM OUTPUT RELATIONSHIPS (ACTIVE-HIGH STATE)

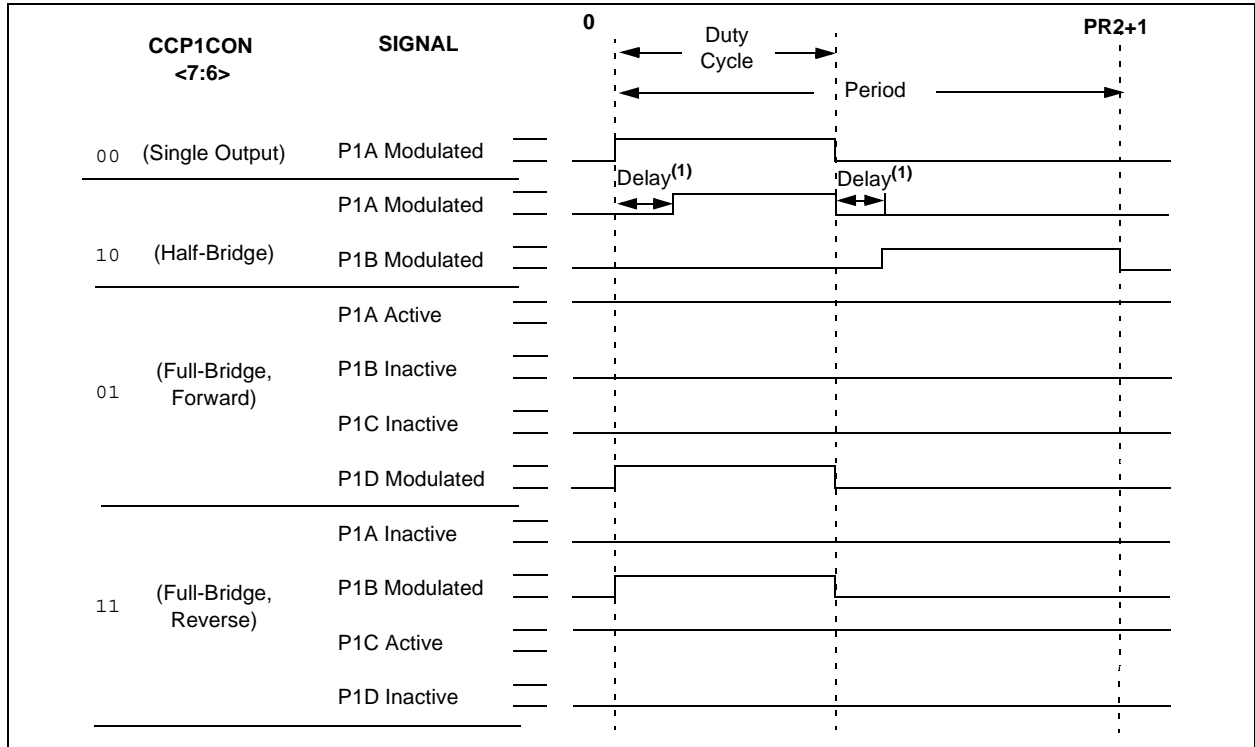
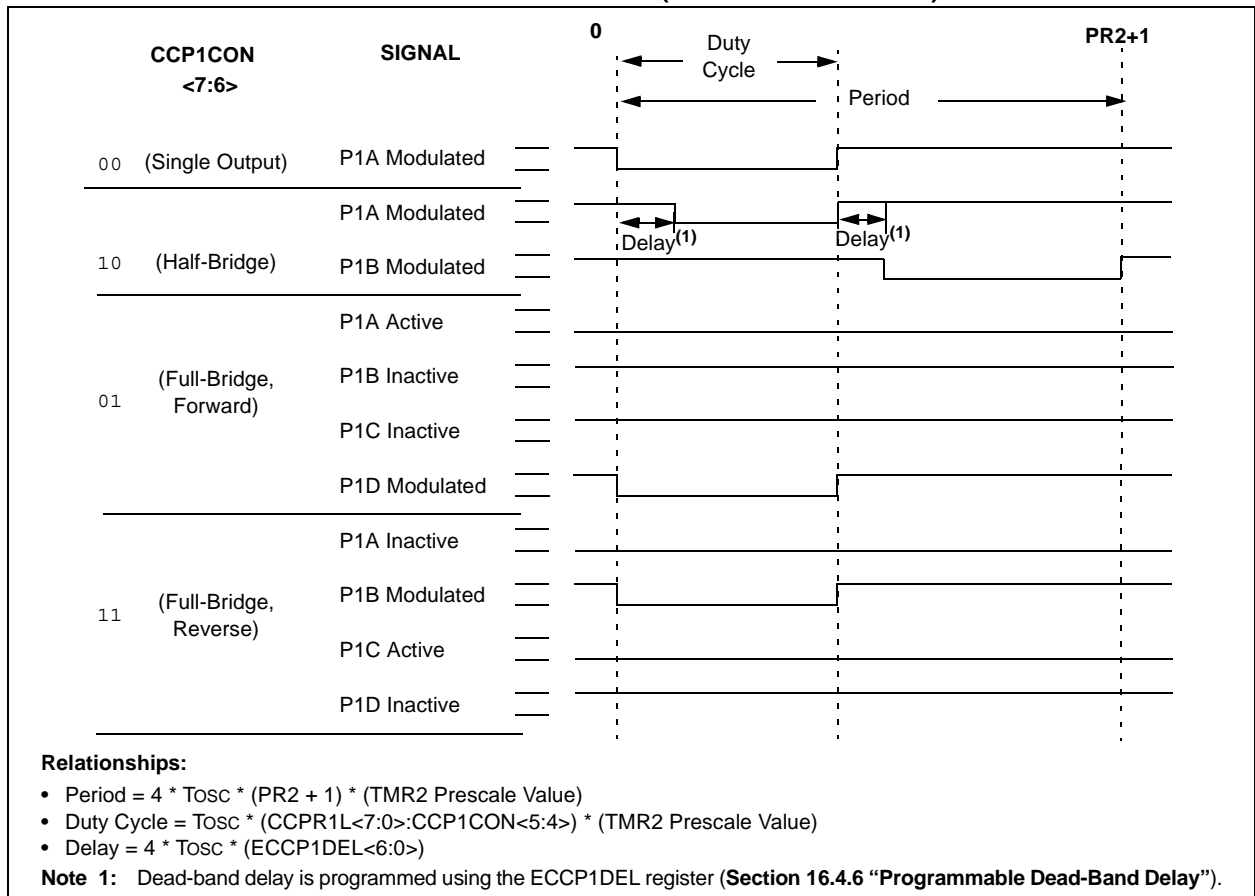


FIGURE 16-3: PWM OUTPUT RELATIONSHIPS (ACTIVE-LOW STATE)



PIC18F2455/2550/4455/4550

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.

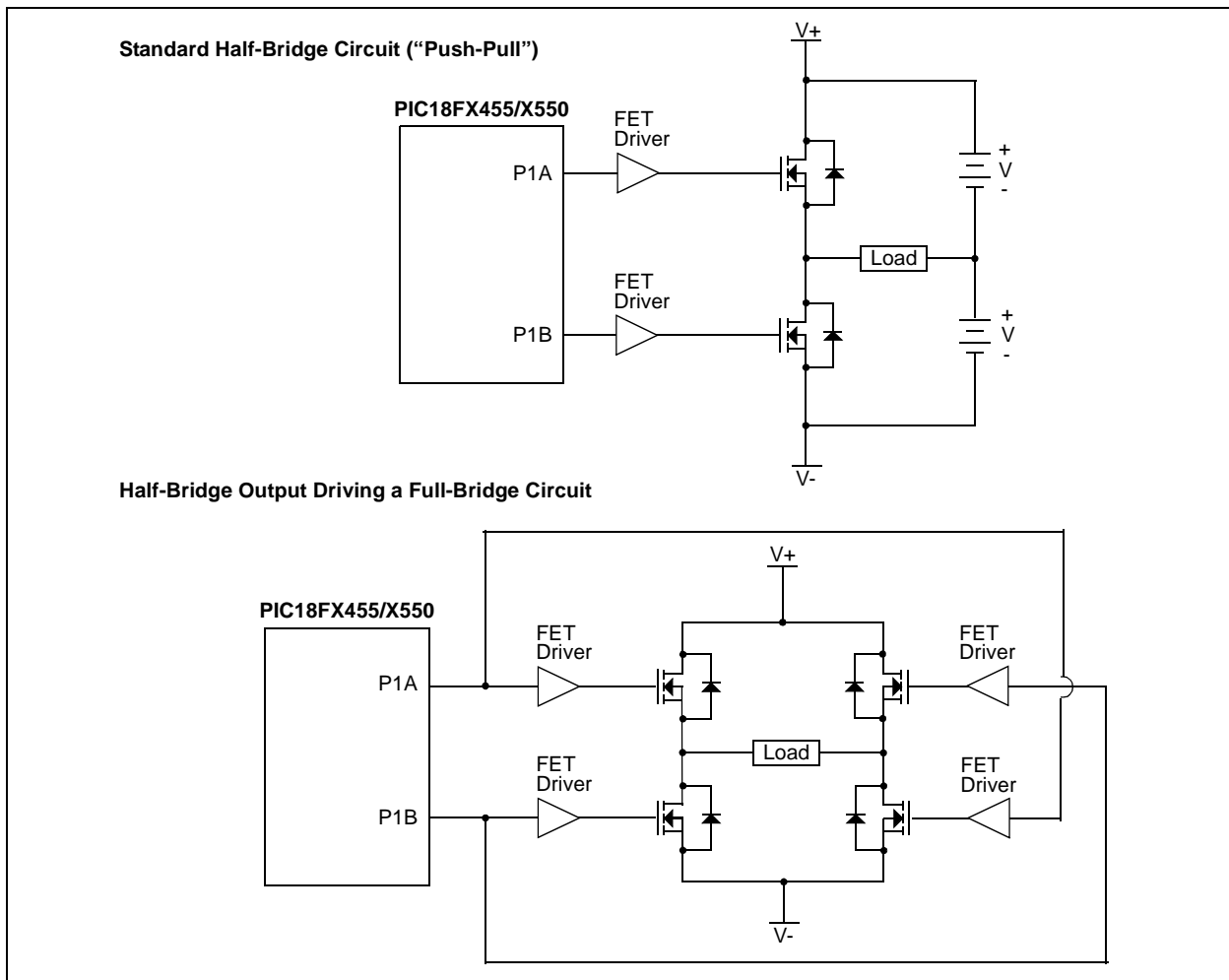
In Half-Bridge Output mode, the programmable dead-band delay can be used to prevent shoot-through current in half-bridge power devices. The value of bits PDC6:PDC0 sets the number of instruction cycles before the output is driven active. If the value is greater than the duty cycle, the corresponding 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



FIGURE 16-5: EXAMPLES OF HALF-BRIDGE OUTPUT MODE APPLICATIONS



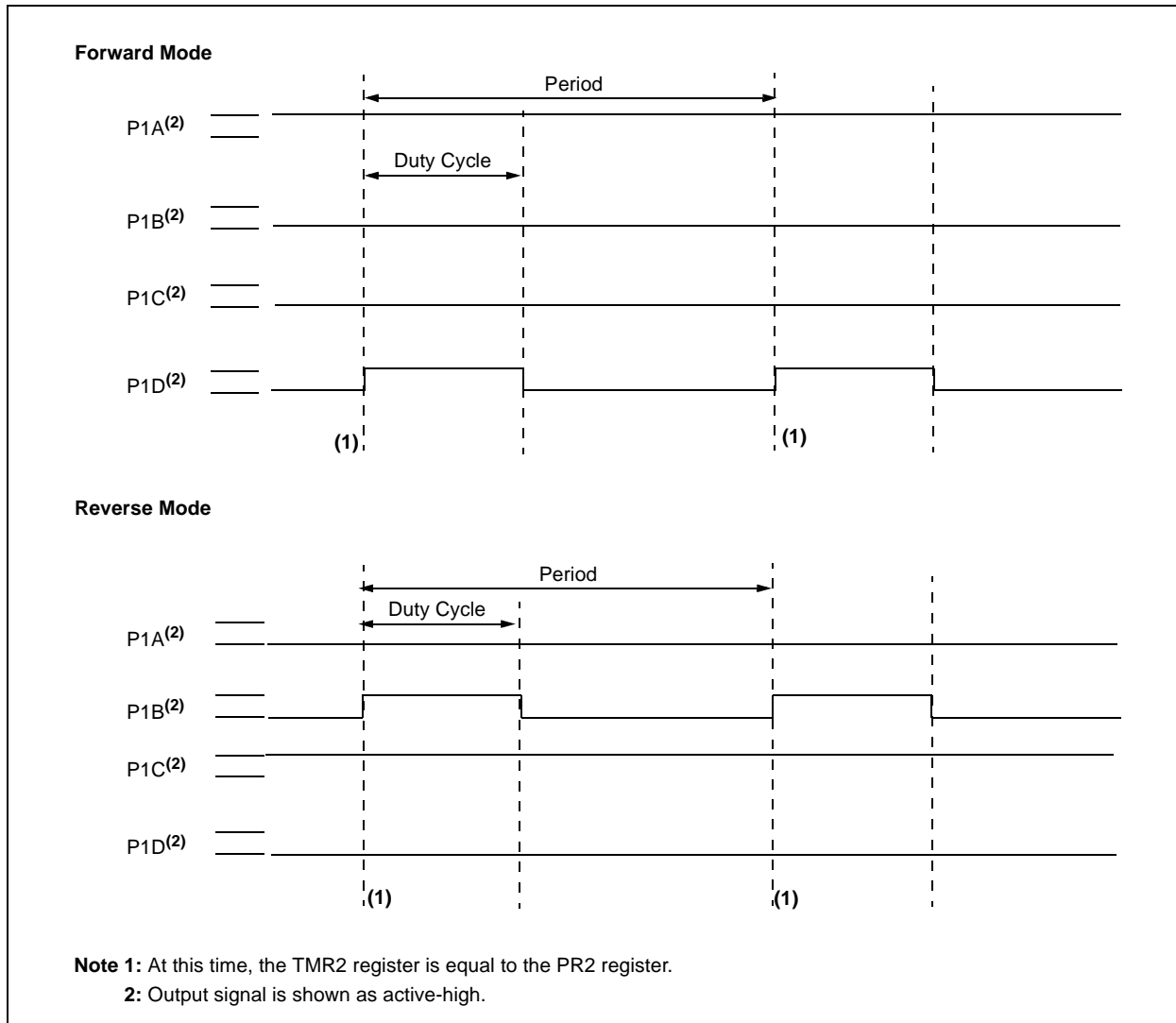
PIC18F2455/2550/4455/4550

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.

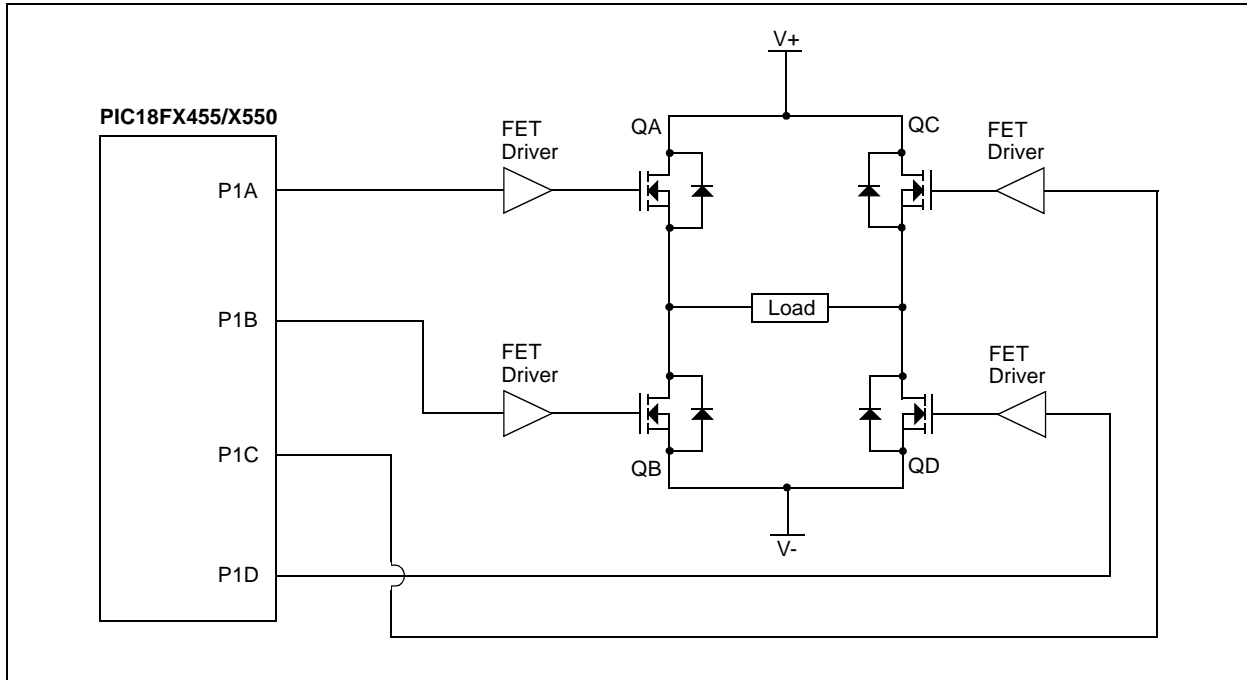
P1A, P1B, P1C and P1D outputs are multiplexed with the PORTC<2>, PORTD<5>, PORTD<6> and PORTD<7> data latches. The TRISC<2>, TRISD<5>, TRISD<6> and TRISD<7> bits must be cleared to make the P1A, P1B, P1C and P1D pins outputs.

FIGURE 16-6: FULL-BRIDGE PWM OUTPUT



PIC18F2455/2550/4455/4550

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 EPWM1M1 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 $(4 T_{OSC} * (\text{Timer2 Prescale Value}))$ before the next PWM period begins. The Timer2 prescaler will be either 1, 4 or 16, depending on the value of the T2CKPS bit (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 CCP1 module does not provide any dead-band delay. In general, since only one output is modulated at all times, dead-band delay is not required. However, there is a situation where a dead-band delay might be required. This situation occurs when both of the following conditions are true:

1. The direction of the PWM output changes when the duty cycle of the output is at or near 100%.
2. The turn-off time of the power switch, including the power device and driver circuit, is greater than the turn-on time.

Figure 16-9 shows an example where the PWM direction changes from forward to reverse at a near 100% duty cycle. At time t_1 , 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.
2. Use switch drivers that can drive the switches off faster than they can drive them on.

Other options to prevent shoot-through current may exist.

FIGURE 16-8: PWM DIRECTION CHANGE

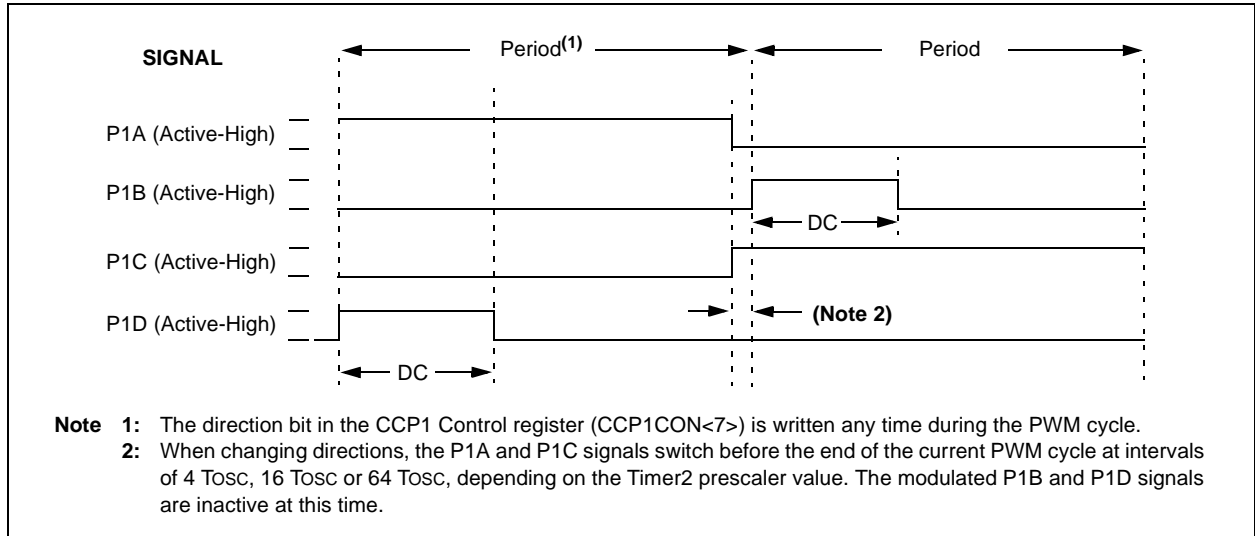
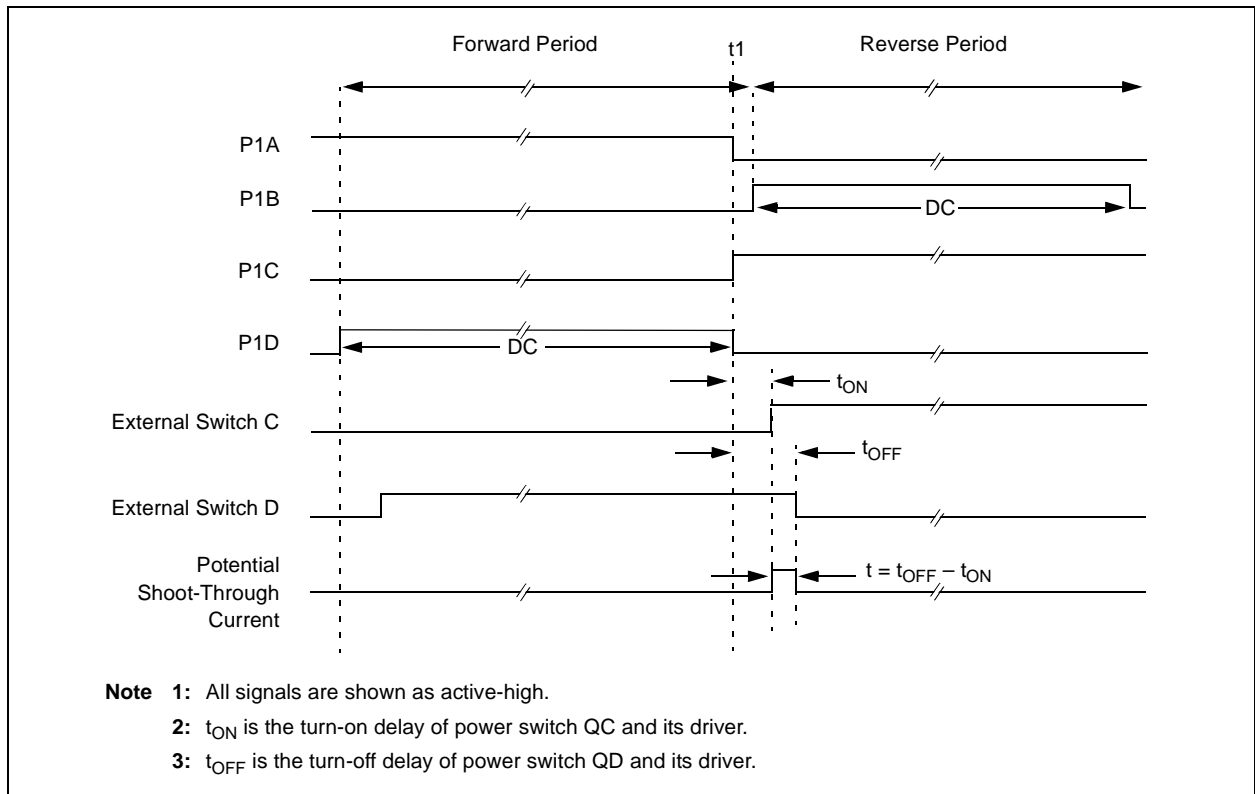


FIGURE 16-9: PWM DIRECTION CHANGE AT NEAR 100% DUTY CYCLE



PIC18F2455/2550/4455/4550

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 CCP1 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 RBOAN12/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 INT0 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 by the PSSAC1:PSSAC0 and PSS1BD1:PSS1BD0 bits (ECCPAS3:ECCPAS0). 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.

Note: Writing to the ECCPASE bit is disabled while a shutdown condition is active.

REGISTER 16-2: ECCP1DEL: PWM 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 |
| PRSEN | PDC6 ⁽¹⁾ | PDC5 ⁽¹⁾ | PDC4 ⁽¹⁾ | PDC3 ⁽¹⁾ | PDC2 ⁽¹⁾ | PDC1 ⁽¹⁾ | PDC0 ⁽¹⁾ |
| bit 7 | | | | | | | bit 0 |

- 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
- bit 6-0 **PDC6:PDC0:** PWM Delay Count bits⁽¹⁾
 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.

| | | | |
|-------------------|------------------|------------------------------------|--------------------|
| 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 |

PIC18F2455/2550/4455/4550

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 ⁽¹⁾ | PSSBD0 ⁽¹⁾ |
| bit 7 | | | | | | bit 0 | |

- 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 = RB0 or Comparator 1 or Comparator 2
 110 = RB0 or Comparator 2
 101 = RB0 or Comparator 1
 100 = RB0
 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⁽¹⁾
 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.

| 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 |

PIC18F2455/2550/4455/4550

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 ECCPASE 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 16-10: PWM AUTO-SHUTDOWN (PRSEN = 1, AUTO-RESTART ENABLED)

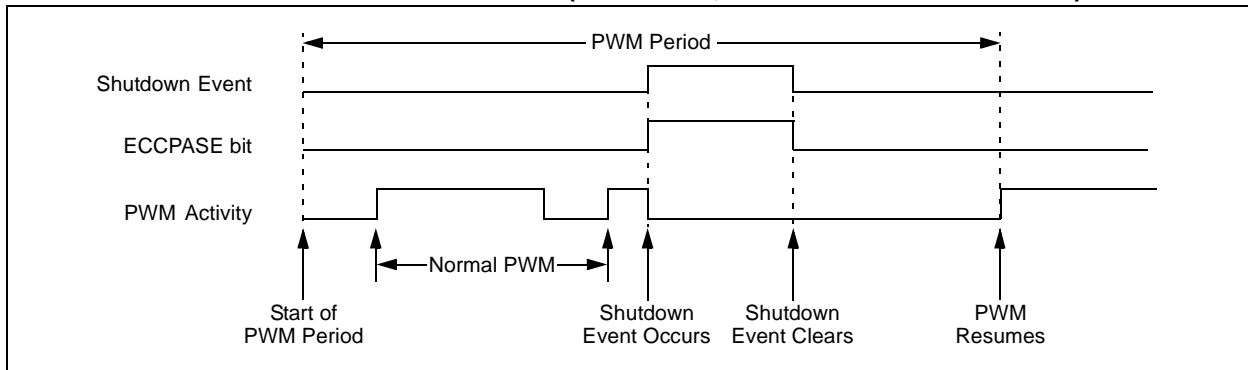
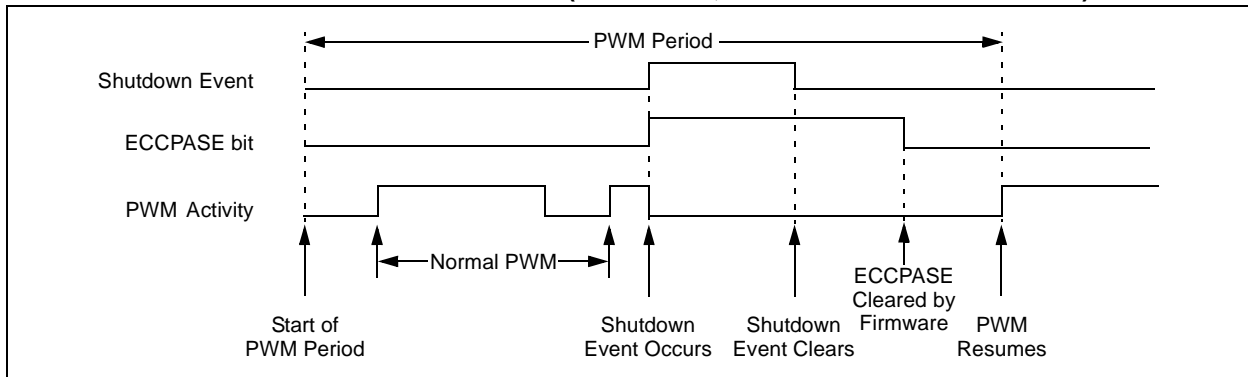


FIGURE 16-11: PWM AUTO-SHUTDOWN (PRSEN = 0, AUTO-RESTART DISABLED)



16.4.9 SETUP FOR PWM OPERATION

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

1. 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. Configure the CCP1 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.
4. Set the PWM duty cycle by loading the CCPR1L register and CCP1CON<5:4> bits.
5. For Half-Bridge Output mode, set the dead-band delay by loading ECCP1DEL<6:0> with the appropriate value.
6. 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 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.
7. If auto-restart operation is required, set the PRSEN bit (ECCP1DEL<7>).
8. 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>).
9. 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 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.

PIC18F2455/2550/4455/4550

TABLE 16-3: REGISTERS ASSOCIATED WITH ECCP1 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 | INT0IF | RBIF | page 51 |
| RCON | IPEN | — | — | \overline{RI} | \overline{TO} | \overline{PD} | \overline{POR} | \overline{BOR} | page 52 |
| IPR1 | PSPIP | ADIP | RCIP | TXIP | SSPIP | CCP1IP | TMR2IP | TMR1IP | page 54 |
| PIR1 | PSPIF | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | page 54 |
| PIE1 | PSPIE | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | page 54 |
| IPR2 | OSCFIP | CMIP | — | EEIP | BCLIP | LVDIP | TMR3IP | CCP2IP | page 54 |
| PIR2 | OSCFIF | CMIF | — | EEIF | BCLIF | LVDIF | TMR3IF | CCP2IF | page 54 |
| PIE2 | OSCFIE | CMIE | — | EEIE | BCLIE | LVDIE | TMR3IE | CCP2IE | page 54 |
| TRISB | PORTB Data Direction Register | | | | | | | | page 54 |
| TRISC | PORTC Data Direction Register | | | | | | | | page 54 |
| TRISD ⁽¹⁾ | PORTD Data Direction Register | | | | | | | | 54 |
| TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | page 52 |
| TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | page 52 |
| T1CON | RD16 | T1RUN | T1CKPS1 | T1CKPS0 | T1OSCEN | T1SYNC | TMR1CS | TMR1ON | page 52 |
| TMR2 | Timer2 Module Register | | | | | | | | page 52 |
| T2CON | — | T2OUTPS3 | T2OUTPS2 | T2OUTPS1 | T2OUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | page 52 |
| PR2 | Timer2 Period Register | | | | | | | | page 52 |
| TMR3L | Holding Register for the Least Significant Byte of the 16-bit TMR3 Register | | | | | | | | page 53 |
| TMR3H | Holding Register for the Most Significant Byte of the 16-bit TMR3 Register | | | | | | | | page 53 |
| T3CON | RD16 | T3ECCP1 ⁽¹⁾ | T3CKPS1 | T3CKPS0 | T3CCP1 ⁽¹⁾ | $\overline{T3SYNC}$ | TMR3CS | TMR3ON | page 53 |
| CCPR1L ⁽²⁾ | Capture/Compare/PWM Register 1 (LSB) | | | | | | | | page 53 |
| CCPR1H ⁽²⁾ | Capture/Compare/PWM Register 1 (MSB) | | | | | | | | page 53 |
| CCP1CON | P1M1 ⁽¹⁾ | P1M0 ⁽¹⁾ | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | page 53 |
| ECCP1AS ⁽²⁾ | ECCPASE | ECCPAS2 | ECCPAS1 | ECCPAS0 | PSSAC1 | PSSAC0 | PSSBD1 ⁽²⁾ | PSSBD0 ⁽²⁾ | page 53 |
| ECCP1DEL | PRSEN | PDC6 ⁽²⁾ | PDC5 ⁽²⁾ | PDC4 ⁽²⁾ | PDC3 ⁽²⁾ | PDC2 ⁽²⁾ | PDC1 ⁽²⁾ | PDC0 ⁽²⁾ | page 53 |

Legend: — = unimplemented, read as '0'. Shaded cells are not used during ECCP operation.

Note 1: These bits are available on 40/44-pin devices only.

2: These bits or registers are unimplemented in 28-pin devices; always maintain these bit clear.

3: These bits are available on 40/44-pin and reserved on 28-pin devices.

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

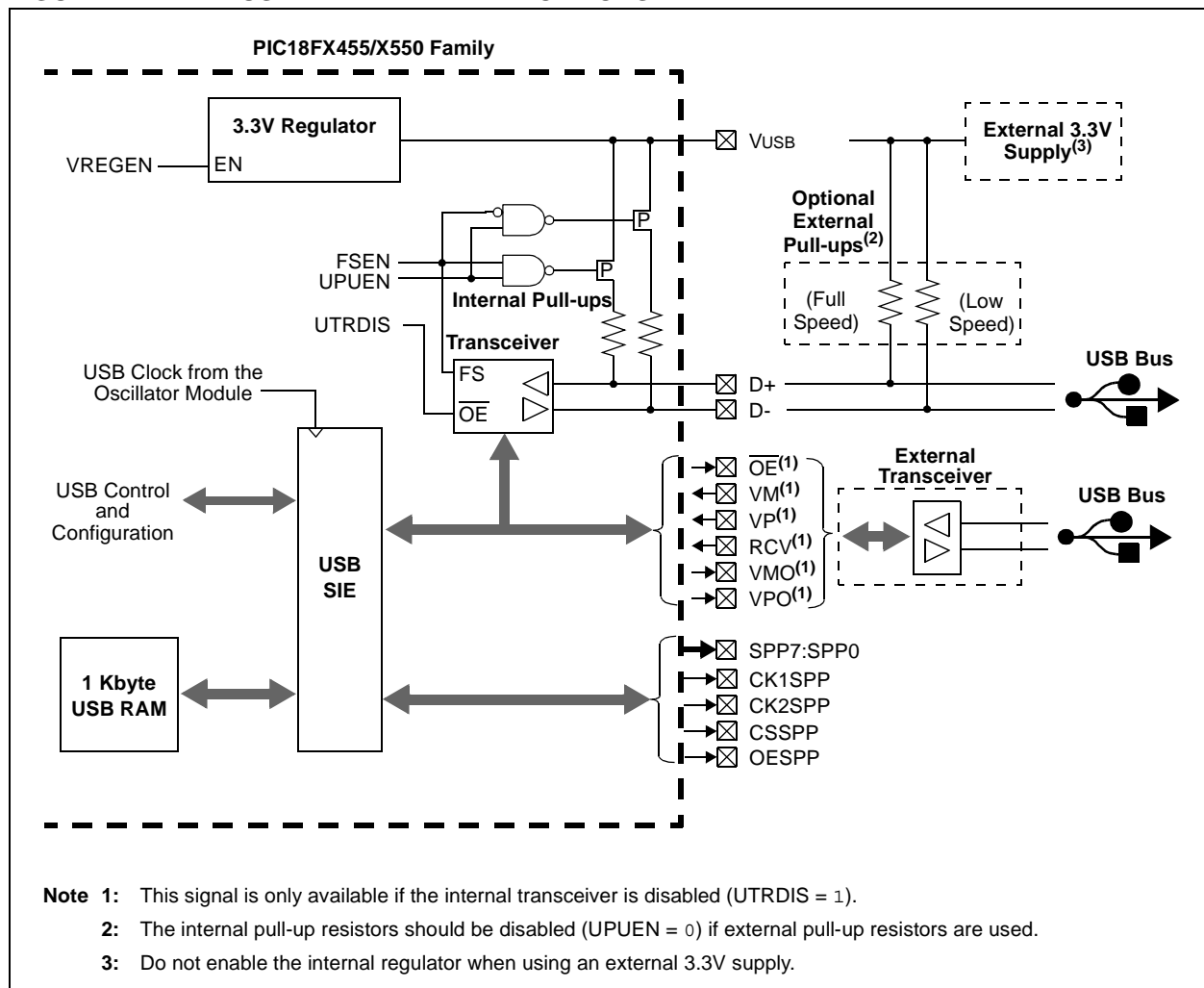
The PIC18FX455/X550 device family contains a full speed and low-speed compatible USB Serial Interface Engine (SIE) that allows fast communications between any USB host and the PIC® microcontroller. The SIE

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



PIC18F2455/2550/4455/4550

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 19 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 configuration bits that control the following:

- Main USB Peripheral Enable
- Ping-Pong Buffer Pointer Reset
- Control of the Suspend mode
- Packet Transfer Disable

In addition, the USB Control register contains a status bit, which is used to indicate the occurrence of a single-ended zero on the bus.

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 configuration 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

| | | | | | | | |
|-------|--------|-----|--------|-------|--------|--------|-------|
| 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 |

- bit 7 **Unimplemented:** Read as '0'
- bit 6 **PPBRST:** Ping-Pong Buffers Reset bit
1 = Reset all ping-pong buffer pointers to the Even BD banks
0 = Ping-pong buffer pointers not being reset
- bit 5 **SE0:** Live Single-Ended Zero Flag bit
1 = Single-ended zero active on the USB bus
0 = No single-ended zero detected
- bit 4 **PKTDIS:** Packet Transfer Disable bit
1 = SIE token and packet processing disabled, automatically set when a SETUP token is received
0 = SIE token and packet processing enabled
- bit 3 **USBEN:** USB Module Enable bit
1 = USB module and supporting circuitry enabled (device attached)
0 = USB module and supporting circuitry disabled (device detached)
- bit 2 **RESUME:** Resume Signaling Enable bit
1 = Resume signaling activated
0 = Resume signaling disabled
- bit 1 **SUSPND:** Suspend USB bit
1 = USB module and supporting circuitry in Power Conserve mode, SIE clock inactive
0 = USB module and supporting circuitry in normal operation, SIE clock clocked at the configured rate
- bit 0 **Unimplemented:** Read as '0'

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

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, 11.9 and 11.4.4 in the USB 2.0 specification.

The SUSPND (UCON<1>) bit 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 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 μ 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 device enters Suspend mode.

17.2.2 USB CONTROL REGISTER (UCON)

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 UCFG register contains most of the bits that control the system level behavior of the USB module. These include:

- Bus speed (full speed vs. 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.

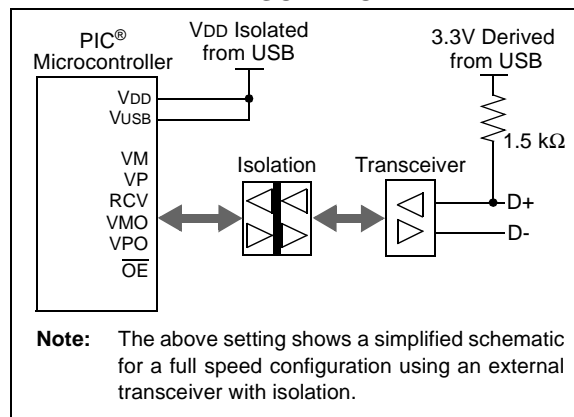
The on-chip USB pull-up resistors are controlled by the UPUEN bit (UCFG<4>). They can only be selected when the on-chip transceiver is enabled.

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



PIC18F2455/2550/4455/4550

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 ⁽¹⁾ | — | UPUEN ^(2,3) | UTRDIS ⁽²⁾ | FSEN ⁽²⁾ | PPB1 | PPB0 |
| bit 7 | | | | | | | bit 0 |

- bit 7 **UTEYE:** USB Eye Pattern Test Enable bit
 1 = Eye pattern test enabled
 0 = Eye pattern test disabled
- bit 6 **UOEMON:** USB \overline{OE} Monitor Enable bit⁽¹⁾
 1 = \overline{OE} signal active; it indicates intervals during which the D+/D- lines are driving
 0 = \overline{OE} 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⁽²⁾
 1 = On-chip transceiver disabled; digital transceiver interface enabled
 0 = On-chip transceiver active
- bit 2 **FSEN:** Full Speed Enable bit⁽²⁾
 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 = Reserved
 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 \overline{OE} signal will be active independent of the UOEMON bit setting.

2: The FSEN, UPUEN and UTRDIS 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.

| 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 |

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
- \overline{OE} : 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' |
| 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 \overline{OE} 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.

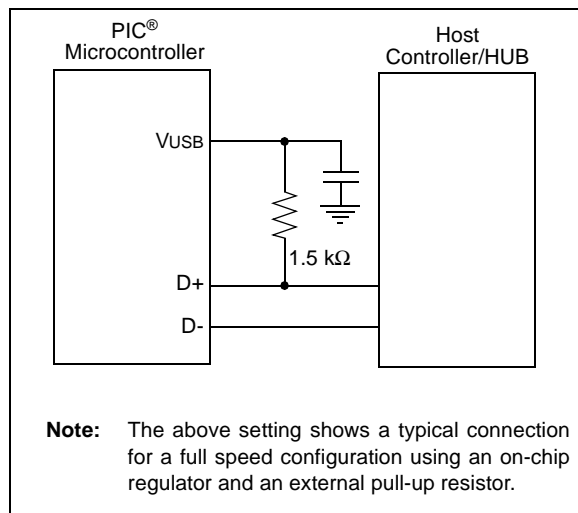
17.2.2.3 Internal Pull-up Resistors

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 Ω ($\pm 5\%$), as required by the USB specifications. Figure 17-3 shows an example.

FIGURE 17-3: EXTERNAL CIRCUITRY



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{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 \overline{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<6> 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 ($\pm 20\%$) 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.
- Note 2:** VDD must be greater than VUSB at all times, even with the regulator disabled.

PIC18F2455/2550/4455/4550

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).

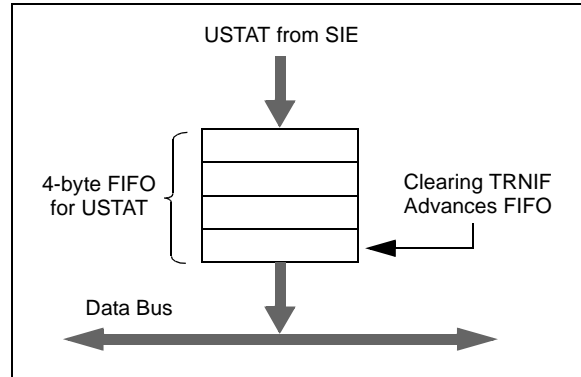
Note: The data in the USB Status register is valid only when the TRNIF interrupt flag is asserted.

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 NACK 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 ⁽¹⁾ | — |
| bit 7 | | | | | | | bit 0 |

- bit 7 **Unimplemented:** Read as '0'
- bit 6-3 **ENDP4:ENDP0:** Encoded number of last endpoint activity (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⁽¹⁾
 - 1 = The last transaction was to the Odd BD bank
 - 0 = The last transaction was to the Even BD bank

Note 1: This bit is only valid for endpoints with available Even and Odd BD registers.
- bit 0 **Unimplemented:** Read as '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 |

PIC18F2455/2550/4455/4550

17.2.4 USB ENDPOINT CONTROL

Each of the 16 possible bidirectional endpoints has its own independent control register, UEP_n (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 (UEP_n<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 (UEP_n<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 bit 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 (UEP_n<2>) is used to enable or disable USB OUT transactions from the host. Setting this bit enables OUT transactions. Similarly, the EPINEN bit (UEP_n<1>) enables or disables USB IN transactions from the host.

The EPSTALL bit (UEP_n<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: UEP_n: 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 | EPOUTEN | EPINEN | EPSTALL ⁽¹⁾ | |
| bit 7 | | | | | | | | bit 0 |

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **EPHSHK:** Endpoint Handshake Enable bit

1 = Endpoint handshake enabled

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 enabled

0 = Endpoint n output disabled

bit 1 **EPINEN:** Endpoint Input Enable bit

1 = Endpoint n input enabled

0 = Endpoint n input disabled

bit 0 **EPSTALL:** Endpoint Stall Enable bit⁽¹⁾

1 = Endpoint n is stalled

0 = Endpoint n is not stalled

Note 1: Valid only if Endpoint n is enabled; otherwise, the bit is ignored.

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

PIC18F2455/2550/4455/4550

17.2.5 USB ADDRESS REGISTER (UADDR)

The 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 Chapter 9 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.

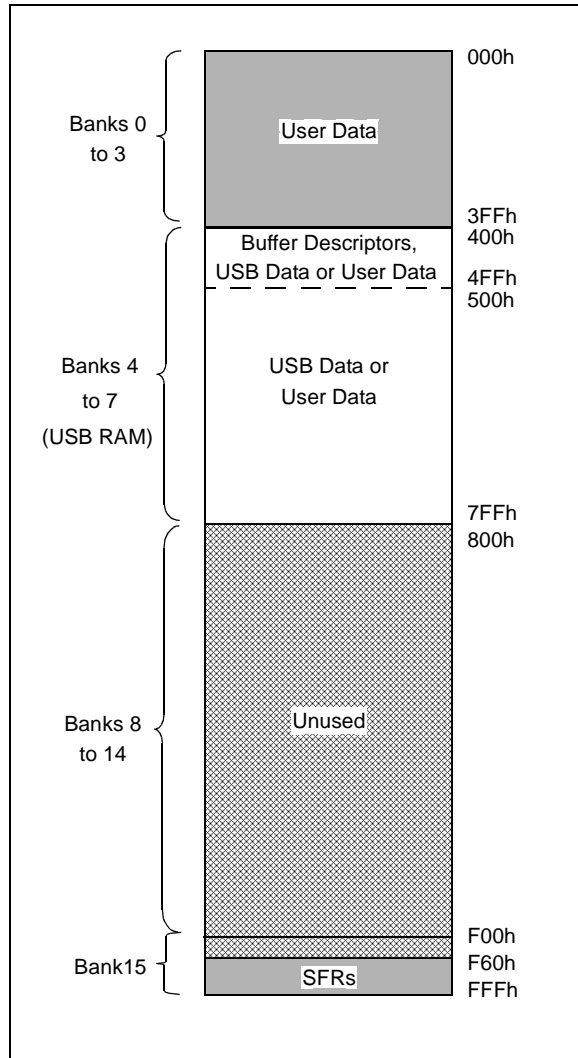
17.3 USB RAM

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.

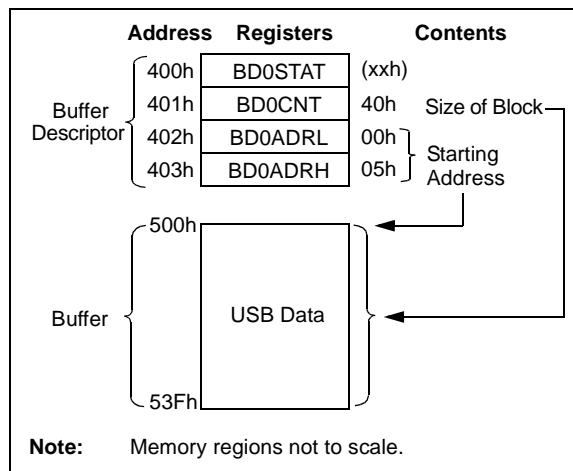
Although they can be thought of as special function control registers, the Buffer Descriptor Status and Address registers are not hardware mapped, as conventional microcontroller SFRs in Bank 15 are. If the endpoint corresponding to a particular BD is not enabled, its registers are not used. Instead 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 40-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 (or more generically, BDSTAT). Each BD has its own unique and correspondingly numbered BDSTAT register.

FIGURE 17-6: EXAMPLE OF A BUFFER DESCRIPTOR



Unlike other control registers, the bit configuration for the BDSTAT 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 (BDSTAT<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 BDSTAT.

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 BDSTAT 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 BDSTAT 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 BDSTAT register is updated by the SIE with the token PID and the transfer count, BDnCNT, is updated.

PIC18F2455/2550/4455/4550

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 BDSTAT 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 (BDSTAT<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 (BDSTAT<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 (BDSTAT<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 (BDSTAT<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 (BDSTAT<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 (BDSTAT<1:0>) store the two most significant digits of the SIE byte count; the lower 8 digits are stored in the corresponding BDCNT 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 from Host | BDnSTAT Settings | | Device Response after Receiving Packet | | | |
|----------------------|------------------|-----|--|------|-------|-------------------------|
| | DTSEN | DTS | Handshake | UOWN | TRNIF | BDSTAT 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 | x | ACK | 0 | 1 | Updated |
| Either, with error | x | x | NACK | 1 | 0 | Not Updated |

Legend: x = don't care

PIC18F2455/2550/4455/4550

REGISTER 17-5: BD_nSTAT: 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 ⁽¹⁾ | DTS ⁽²⁾ | KEN | INCDIS | DTSEN | BSTALL | BC9 | BC8 |
| bit 7 | | | | | | bit 0 | |

- bit 7 **UOWN:** USB Own bit⁽¹⁾
0 = The microcontroller core owns the BD and its corresponding buffer
- bit 6 **DTS:** Data Toggle Synchronization bit⁽²⁾
1 = Data 1 packet
0 = 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
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 bits 9 and 8
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.

| 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 |

PIC18F2455/2550/4455/4550

17.4.1.3 BDSTAT Register (SIE Mode)

When the BD and its buffer are owned by the SIE, most of the bits in BDSTAT 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 BDSTAT register is updated by the SIE with the token Packet Identifier (PID), which is stored in BDSTAT<5:3>. The transfer count in the corresponding BDCNT register is updated; values that overflow the 8-bit register carry over to the two most significant digits of the count, stored in BDSTAT<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 NACK 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 BDCNT register. The upper two bits reside in BDSTAT<1:0>. This represents a valid byte range of 0 to 1023.

17.4.3 BD ADDRESS VALIDATION

The BD Address register pair contain 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 |

- 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 bits 9 and 8
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.

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 |

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 three modes of operation:

- No ping-pong support
- Ping-pong buffer support for OUT Endpoint 0 only
- Ping-pong buffer support for all endpoints

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

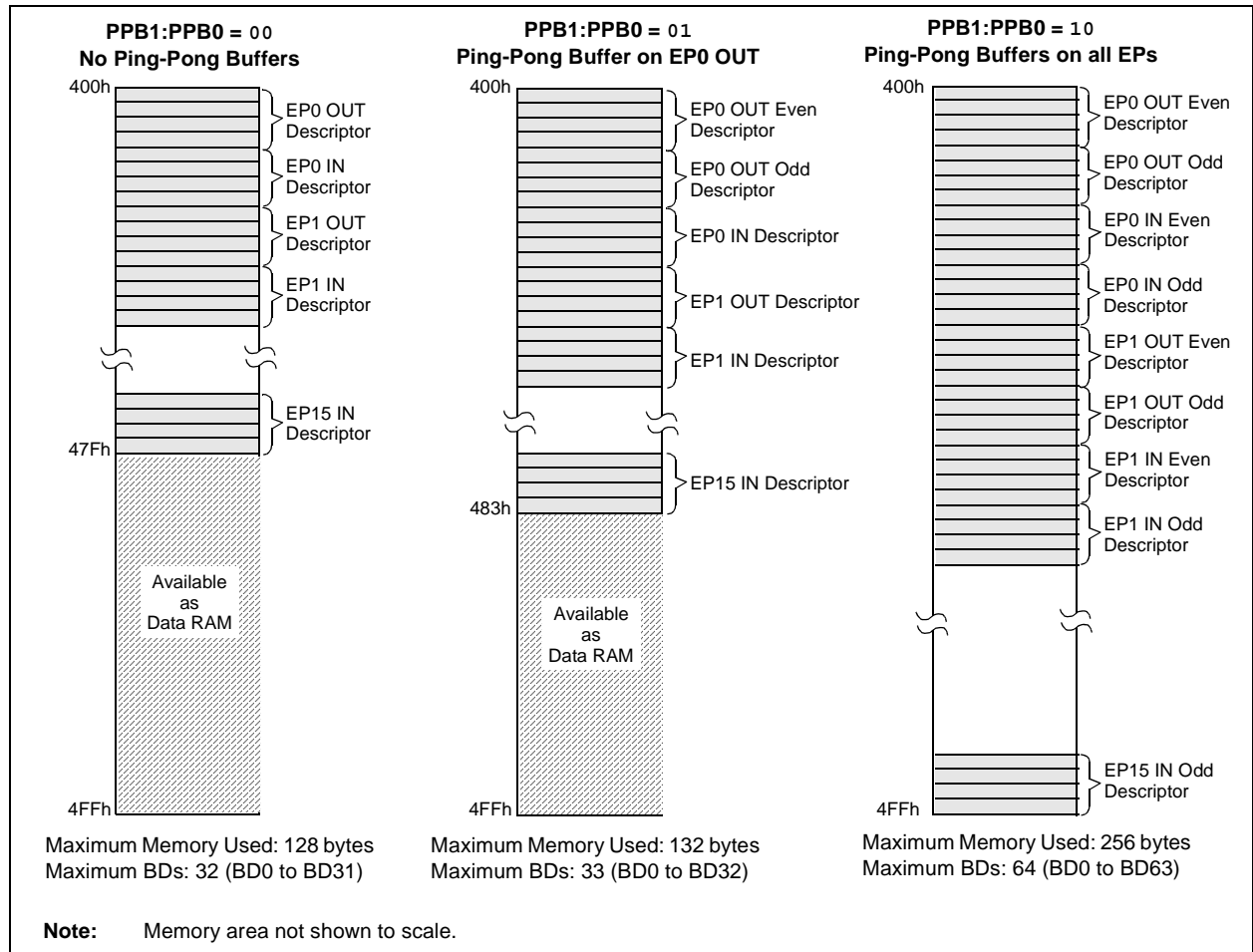
SIE), the pointer is toggled to the Odd BD. After the completion of the next transaction, the pointer is toggled back to the Even BD and so on.

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 three 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



PIC18F2455/2550/4455/4550

TABLE 17-4: ASSIGNMENT OF BUFFER DESCRIPTORS FOR THE DIFFERENT BUFFERING MODES

| Endpoint | BDs Assigned to Endpoint | | | | | |
|----------|--------------------------|----|----------------------------------|----|----------------------------------|----------------|
| | Mode 0 (No Ping-Pong) | | Mode 1 (Ping-Pong on EP0 OUT) | | Mode 2 (Ping-Pong on all EPs) | |
| | Out | In | Out | In | Out | In |
| 0 | 0 | 1 | 0 (E), 1 (O) | 2 | 0 (E), 1 (O) | 2 (E), 3 (O) |
| 1 | 2 | 3 | 3 | 4 | 4 (E), 5 (O) | 6 (E), 7 (O) |
| 2 | 4 | 5 | 5 | 6 | 8 (E), 9 (O) | 10 (E), 11 (O) |
| 3 | 6 | 7 | 7 | 8 | 12 (E), 13 (O) | 14 (E), 15 (O) |
| 4 | 8 | 9 | 9 | 10 | 16 (E), 17 (O) | 18 (E), 19 (O) |
| 5 | 10 | 11 | 11 | 12 | 20 (E), 21 (O) | 22 (E), 23 (O) |
| 6 | 12 | 13 | 13 | 14 | 24 (E), 25 (O) | 26 (E), 27 (O) |
| 7 | 14 | 15 | 15 | 16 | 28 (E), 29 (O) | 30 (E), 31 (O) |
| 8 | 16 | 17 | 17 | 18 | 32 (E), 33 (O) | 34 (E), 35 (O) |
| 9 | 18 | 19 | 19 | 20 | 36 (E), 37 (O) | 38 (E), 39 (O) |
| 10 | 20 | 21 | 21 | 22 | 40 (E), 41 (O) | 42 (E), 43 (O) |
| 11 | 22 | 23 | 23 | 24 | 44 (E), 45 (O) | 46 (E), 47 (O) |
| 12 | 24 | 25 | 25 | 26 | 48 (E), 49 (O) | 50 (E), 51 (O) |
| 13 | 26 | 27 | 27 | 28 | 52 (E), 53 (O) | 54 (E), 55 (O) |
| 14 | 28 | 29 | 29 | 30 | 56 (E), 57 (O) | 58 (E), 59 (O) |
| 15 | 30 | 31 | 31 | 32 | 60 (E), 61 (O) | 62 (E), 63 (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 ⁽¹⁾ | UOWN | DTS | PID3 ⁽²⁾ KEN ⁽³⁾ | PID2 ⁽²⁾ INCDIS ⁽³⁾ | PID1 ⁽²⁾ DTSEN ⁽³⁾ | PID0 ⁽²⁾ BSTALL ⁽³⁾ | BC9 | BC8 |
| BDnCNT ⁽¹⁾ | Byte Count | | | | | | | |
| BDnADRL ⁽¹⁾ | Buffer Address Low | | | | | | | |
| BDnADRH ⁽¹⁾ | 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.

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

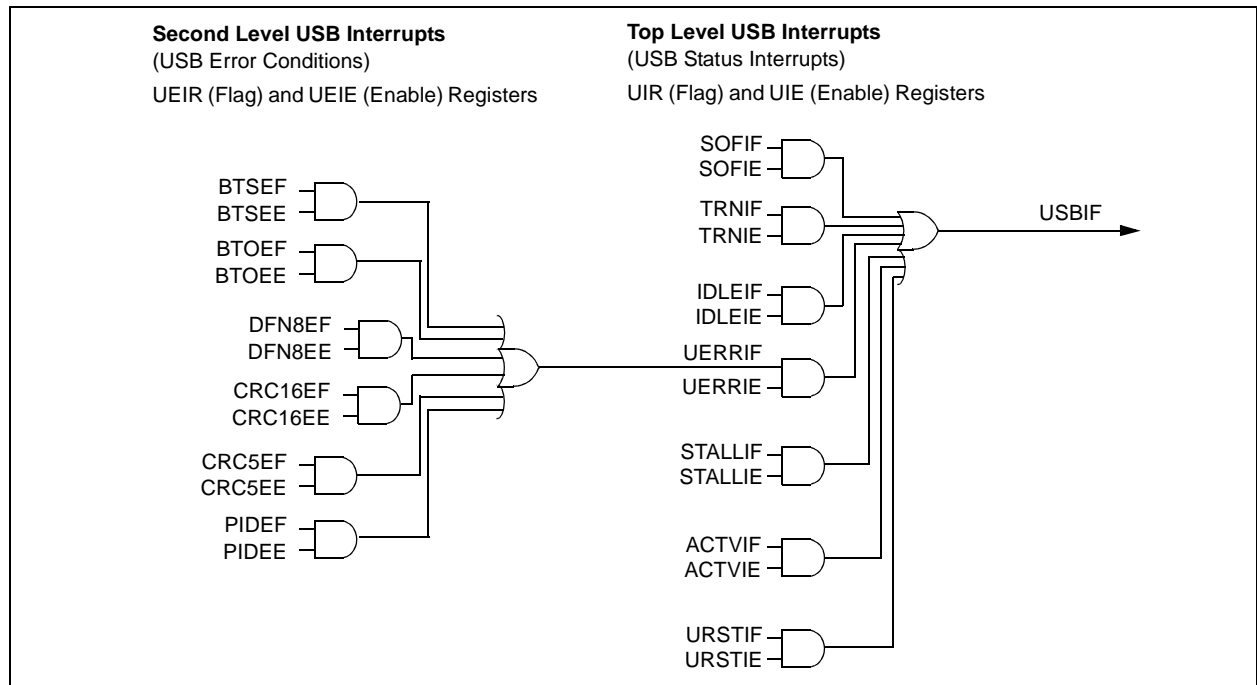
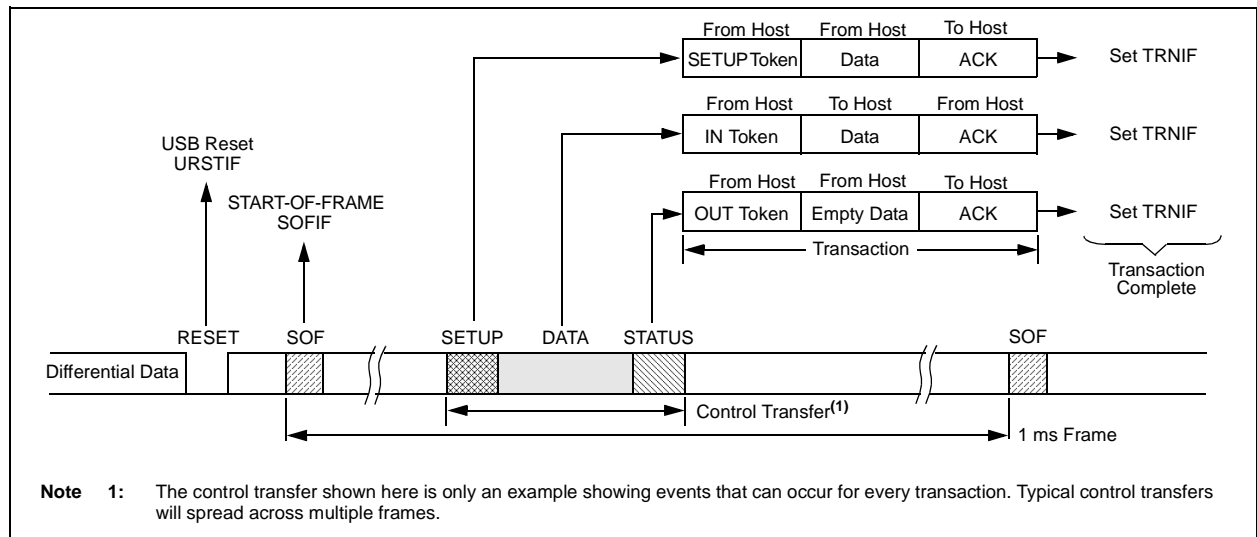


FIGURE 17-9: EXAMPLE OF A USB TRANSACTION AND INTERRUPT EVENTS



PIC18F2455/2550/4455/4550

17.5.1 USB INTERRUPT STATUS REGISTER (UIR)

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

| 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 ⁽¹⁾ | TRNIF ⁽²⁾ | ACTVIF ⁽³⁾ | UERRIF ⁽⁴⁾ | URSTIF |
| bit 7 | | | | | | | bit 0 |

- bit 7 **Unimplemented:** Read as '0'
 - bit 6 **SOFIF:** START-OF-FRAME Token Interrupt bit
1 = A START-OF-FRAME token received by the SIE
0 = No START-OF-FRAME token received by the SIE
 - bit 5 **STALLIF:** A STALL Handshake Interrupt bit
1 = A STALL handshake was sent by the SIE
0 = A STALL handshake has not been sent
 - bit 4 **IDLEIF:** Idle Detect Interrupt bit⁽¹⁾
1 = Idle condition detected (constant Idle state of 3 ms or more)
0 = No Idle condition detected
 - bit 3 **TRNIF:** Transaction Complete Interrupt bit⁽²⁾
1 = Processing of pending transaction is complete; read USTAT register for endpoint information
0 = Processing of pending transaction is not complete or no transaction is pending
 - bit 2 **ACTVIF:** Bus Activity Detect Interrupt bit⁽³⁾
1 = Activity on the D+/D- lines was detected
0 = No activity detected on the D+/D- lines
 - bit 1 **UERRIF:** USB Error Condition Interrupt bit⁽⁴⁾
1 = An unmasked error condition has occurred
0 = No unmasked error condition has occurred
 - bit 0 **URSTIF:** USB Reset Interrupt bit
1 = Valid USB Reset occurred; 00h is loaded into UADDR register
0 = No USB Reset has occurred
- 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. The bit is a status bit only and cannot be set or cleared by the user.

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 |

PIC18F2455/2550/4455/4550

17.5.2 USB INTERRUPT ENABLE REGISTER (UIE)

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.

REGISTER 17-8: UIE: USB INTERRUPT ENABLE REGISTER

| | | | | | | | |
|-------|-------|---------|--------|-------|--------|--------|--------|
| 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 |

- bit 7 **Unimplemented:** Read as '0'
- bit 6 **SOFIE:** START-OF-FRAME Token Interrupt Enable bit
1 = START-OF-FRAME token interrupt enabled
0 = START-OF-FRAME token interrupt disabled
- bit 5 **STALLIE:** STALL Handshake Interrupt Enable bit
1 = STALL interrupt enabled
0 = STALL interrupt disabled
- bit 4 **IDLEIE:** Idle Detect Interrupt Enable bit
1 = Idle detect interrupt enabled
0 = Idle detect interrupt disabled
- bit 3 **TRNIE:** Transaction Complete Interrupt Enable bit
1 = Transaction interrupt enabled
0 = Transaction interrupt disabled
- bit 2 **ACTVIE:** Bus Activity Detect Interrupt Enable bit
1 = Bus activity detect interrupt enabled
0 = Bus activity detect interrupt disabled
- bit 1 **UERRIE:** USB Error Interrupt Enable bit
1 = USB error interrupt enabled
0 = USB error interrupt disabled
- bit 0 **URSTIE:** USB Reset Interrupt Enable bit
1 = USB Reset interrupt enabled
0 = USB Reset interrupt disabled

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 |

PIC18F2455/2550/4455/4550

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 | CRCE5F | PIDEF |
| bit 7 | | | | | | | bit 0 |

- 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 bytes
0 = The data field was an integral number of bytes
- bit 2 **CRC16EF:** CRC16 Failure Flag bit
1 = The CRC16 failed
0 = The CRC16 passed
- bit 1 **CRCE5F:** CRC5 Host Error Flag bit
1 = The token packet was rejected due to a CRC5 error
0 = The token packet was accepted
- bit 0 **PIDEF:** PID Check Failure Flag bit
1 = PID check failed
0 = PID check passed

Legend:

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

PIC18F2455/2550/4455/4550

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.

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 with actually generating an interrupt.

REGISTER 17-10: UEIE: USB ERROR INTERRUPT ENABLE REGISTER

| 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 |

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 **BTOEE:** Bus Turnaround Time-out Error Interrupt Enable bit

1 = Bus turnaround time-out error interrupt enabled
0 = Bus turnaround time-out error interrupt disabled

bit 3 **DFN8EE:** 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

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 |

PIC18F2455/2550/4455/4550

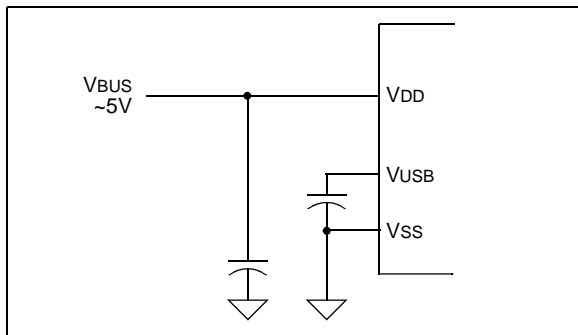
17.6 USB Power Modes

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.

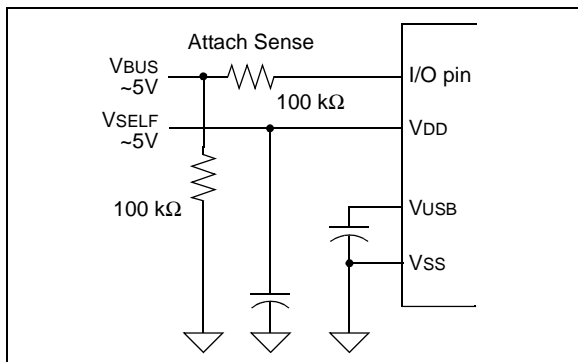
FIGURE 17-10: BUS POWER ONLY



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.

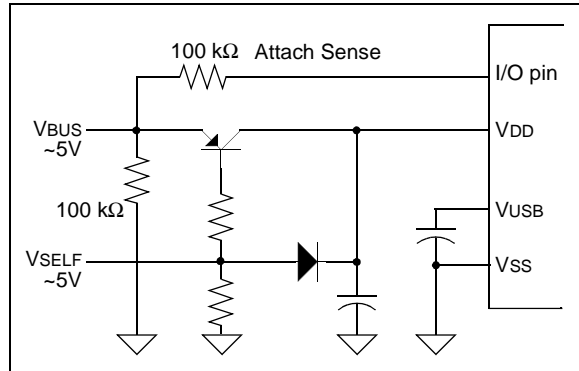
FIGURE 17-11: SELF-POWER ONLY



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.
2. Set the KEN bit (BDnSTAT<5>) to let SIE keep control of the buffer.
3. 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: 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.

PIC18F2455/2550/4455/4550

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 Chapter 9 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 | RBFIF | 51 |
| IPR2 | OSCFIP | CMIP | USBIP | EEIP | BCLIP | LVDIP | TMR3IP | CCP2IP | 54 |
| PIR2 | OSCFIF | CMIF | USBIF | EEIF | BCLIF | LVDIF | TMR3IF | CCP2IF | 54 |
| PIE2 | OSCFIE | CMIE | USBIE | EEIE | BCLIE | LVDIE | 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 | — | 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.

PIC18F2455/2550/4455/4550

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 will be a good, albeit basic, high-level refresher of USB.

17.10.1 LAYERED FRAMEWORK

A 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.

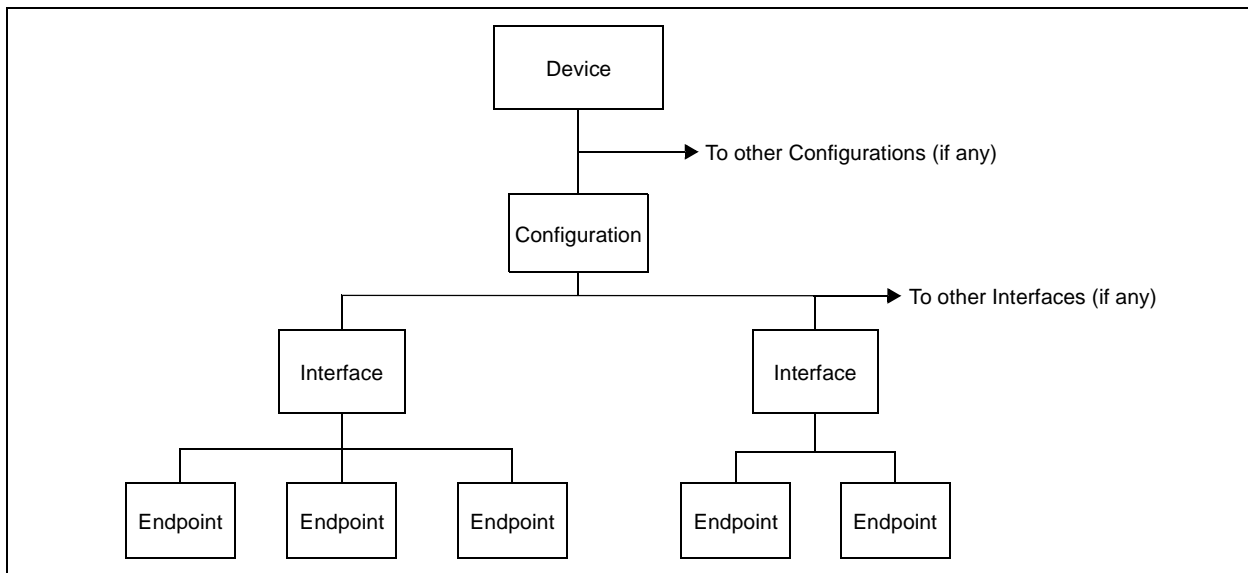
- **Isochronous:** This type provides a transfer method for large amounts of data (up to 1023 bytes) with timely delivery ensured; however, the data integrity is not ensured. This is good for streaming applications where small data loss is not critical, such as audio.
- **Bulk:** This type of transfer 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.

FIGURE 17-13: USB LAYERS



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 one unit load or less, if necessary.

The USB specification also defines a Suspend mode. In this situation, current must be limited to 500 μ 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:

1. 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.
4. Set Address: The host assigns an address to the device.
5. 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 Ω 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.

PIC18F2455/2550/4455/4550

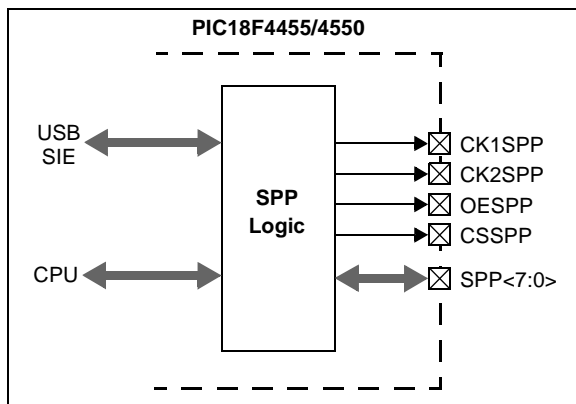
NOTES:

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

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

PIC18F2455/2550/4455/4550

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 | | |

- 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

| | | | |
|-------------------|------------------|------------------------------------|--------------------|
| 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 |

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>).

PIC18F2455/2550/4455/4550

FIGURE 18-2: TIMING FOR MICROCONTROLLER WRITE ADDRESS, WRITE DATA AND READ DATA (NO WAIT STATES)

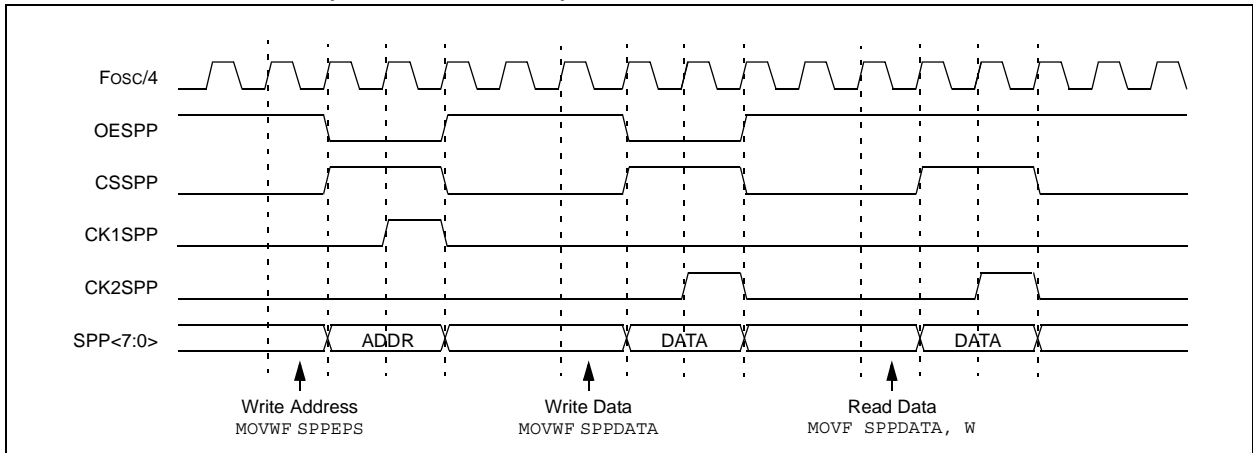


FIGURE 18-3: TIMING FOR USB WRITE ADDRESS AND DATA (4 WAIT STATES)

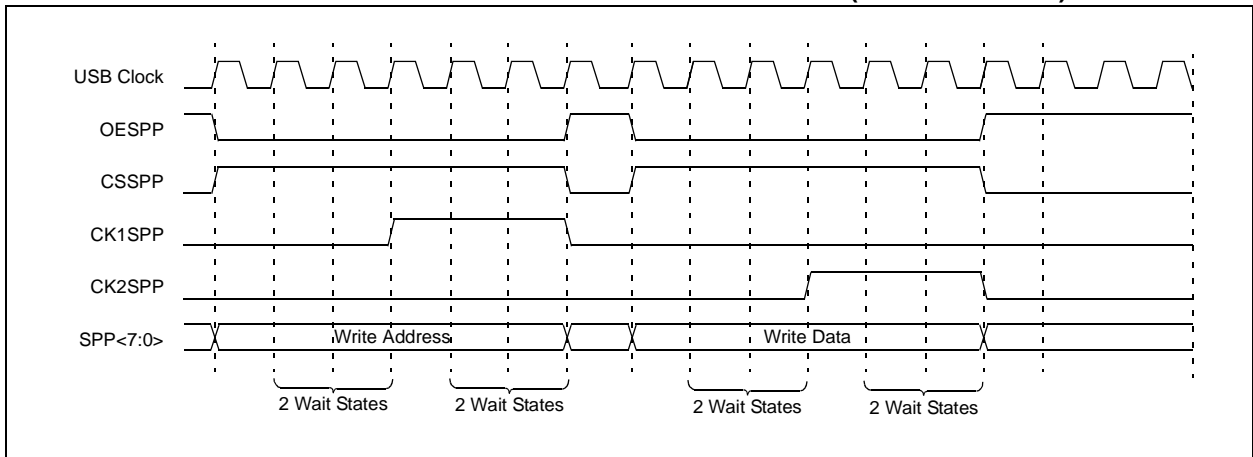
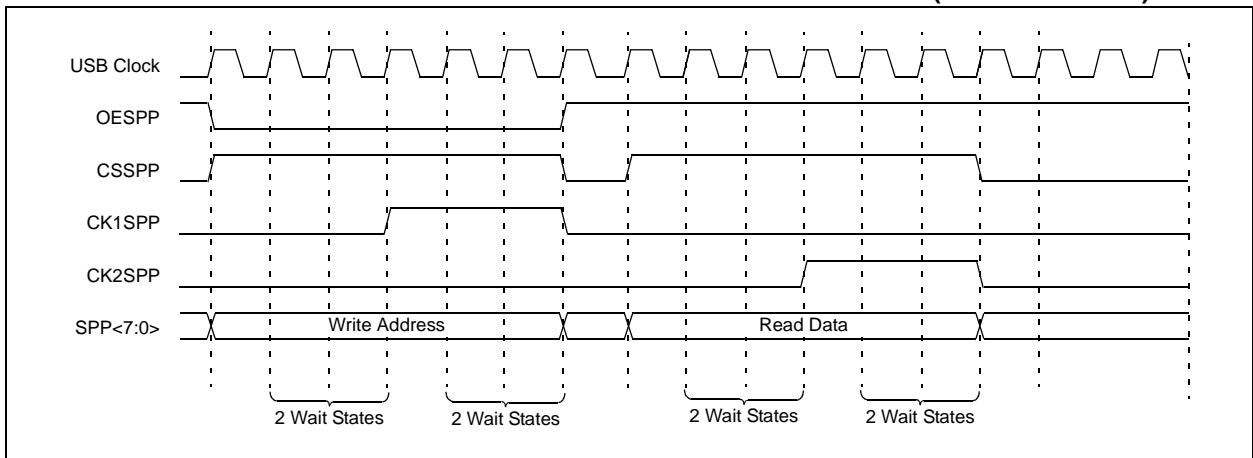


FIGURE 18-4: TIMING FOR USB WRITE ADDRESS AND READ DATA (4 WAIT STATES)



PIC18F2455/2550/4455/4550

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 setup the SPP for USB control:

1. 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.
5. Set the INCDIS bit (BDnSTAT<4>) to disable automatic buffer address increment.
6. Set the SPPEN bit to enable the module.

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 ($F_{OSC}/4$).

The following steps are required to setup the SPP for microcontroller operation:

1. 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 (PIR<7>) and is enabled by the SSPIE bit (PIE1<7>). Like all other microcontroller level interrupts, it can be set to a low or high priority. This is done with the SSPIP bit (IPR<7>).

18.3.2 WRITING TO THE SPP

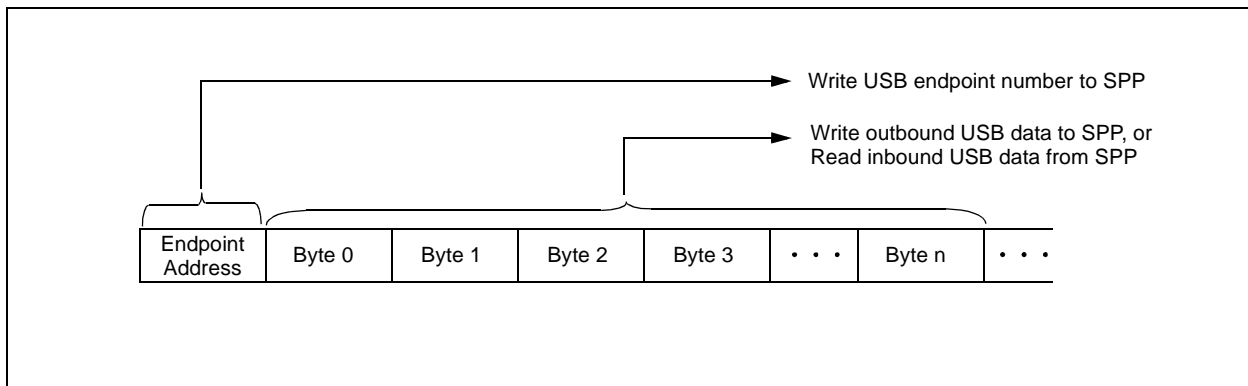
Once configured, writing to the SPP is performed by writing to the SPPEPS and SPPDATA registers. If the SPP is configured to clock out endpoint address information with the data, writing to the SPPEPS register initiates the address write cycle. Otherwise, the write is started by writing the data to the SPPDATA register. The SPPBUSY bit indicates the status of the address and the data write cycles.

The following is an example write sequence:

1. 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.
2. Monitor the BUSY 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.
4. Monitor the BUSY bit to determine when the data has been sent. The duration depends on the wait states.
5. 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



PIC18F2455/2550/4455/4550

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 | — | — | — | — | — | — | SPPOWN | SPPEN | 55 |
| SPPCFG | CLKCFG1 | CLKCFG0 | CSEN | CLK1EN | WS3 | WS2 | WS1 | WS0 | 55 |
| SPPEPS | RDSPP | WRSP | — | SPPBUSY | ADDR3 | ADDR2 | ADDR1 | ADDR0 | 55 |
| SPPDATA | SPP Data Register | | | | | | | | 55 |
| PIR1 | SPPIF | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 54 |
| PIE1 | SPPIE | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 54 |
| IPR1 | SPPIP | ADIP | RCIP | TXIP | SSPIP | CCP1IP | TMR2IP | TMR1IP | 54 |
| PORTE | RDP | — | — | — | RE3 ⁽¹⁾ | RE2 | RE1 | RE0 | 54 |

Legend: — = unimplemented, read as '0'. Shaded cells are not used for the Streaming Parallel Port.

Note 1: RE3 is available when the MCLRE configuration bit is clear; otherwise, it reads as '0'.

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 (I²C)
 - Full Master mode
 - Slave mode (with general address call)

The I²C interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode

19.2 Control Registers

The MSSP module has three associated 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 I²C 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 SPI modes are supported. To accomplish communication, typically three pins are used:

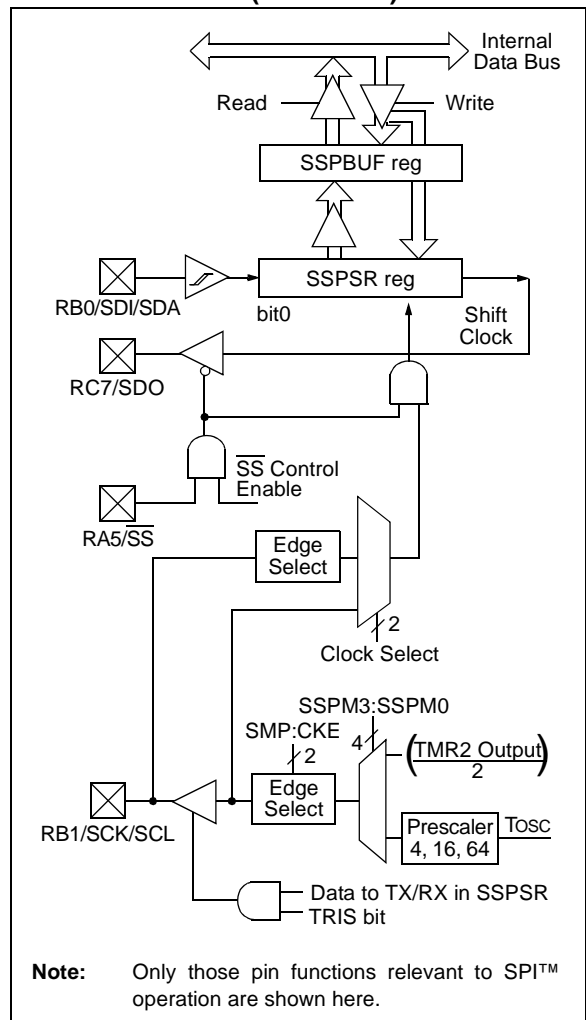
- Serial Data Out (SDO) – RC7/RX/DT/SDO
- Serial Data In (SDI) – RB0/AN12/INT0/FLT0/SDI/SDA
- Serial Clock (SCK) – RB1/AN10/INT1/SCK/SCL

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

- Slave Select (\overline{SS}) – RA5/AN4/ \overline{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)



PIC18F2455/2550/4455/4550

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 6 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 | D/A | P | S | R/W | UA | BF |
| | | | | | | bit 0 | |

- bit 7 **SMP:** Sample bit
SPI Master mode:
 1 = Input data sampled at end of data output time
 0 = Input data sampled at middle of data output time
SPI Slave mode:
 SMP must be cleared when SPI operation is used in Slave mode.
- bit 6 **CKE:** SPI Clock Select bit
 1 = Transmit occurs on transition from active to Idle clock state
 0 = Transmit occurs on transition from Idle to active clock state
 Polarity of clock state is set by the CKP bit (SSPCON1<4>).
- bit 5 **D/A:** Data/Address bit
 Used in I²C mode only.
- bit 4 **P:** Stop bit
 Used in I²C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.
- bit 3 **S:** Start bit
 Used in I²C mode only.
- bit 2 **R/W:** Read/Write bit Information
 Used in I²C mode only.
- bit 1 **UA:** Update Address bit
 Used in I²C mode only.
- bit 0 **BF:** Buffer Full Status bit (Receive mode only)
 1 = Receive complete, SSPBUF is full
 0 = Receive not complete, SSPBUF is empty

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

PIC18F2455/2550/4455/4550

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 | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **WCOL:** Write Collision Detect bit (Transmit mode only)

- 1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
- 0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit

SPI Slave mode:

- 1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow (must be cleared in software).
- 0 = No overflow

Note: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

bit 5 **SSPEN:** Synchronous Serial Port Enable bit

- 1 = Enables serial port and configures SCK, SDO, SDI and \overline{SS} as serial port pins
- 0 = Disables serial port and configures these pins as I/O port pins

Note: When enabled, these pins must be properly configured as input or output.

bit 4 **CKP:** Clock Polarity Select bit

- 1 = Idle state for clock is a high level
- 0 = Idle state for clock is a low level

bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits

- 0101 = SPI Slave mode, clock = SCK pin, \overline{SS} pin control disabled, \overline{SS} can be used as I/O pin
- 0100 = SPI Slave mode, clock = SCK pin, \overline{SS} pin control enabled
- 0011 = SPI Master mode, clock = TMR2 output/2
- 0010 = SPI Master mode, clock = FOSC/64
- 0001 = SPI Master mode, clock = FOSC/16
- 0000 = SPI Master mode, clock = FOSC/4

Note: Bit combinations not specifically listed here are either reserved or implemented in I²C mode only.

Legend:

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

PIC18F2455/2550/4455/4550

19.3.2 OPERATION

When initializing SPI operation, 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 consists of a Transmit/Receive Shift register (SSPSR) and a Buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full detect bit, BF (SSPSTAT<0>) and the interrupt flag bit, SSPIF, 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. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 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 | SSPSTAT, BF | ;Has data been received (transmit complete)? |
| | BRA | LOOP | ;No |
| | MOVF | SSPBUF, W | ;WREG reg = contents of SSPBUF |
| | MOVWF | RXDATA | ;Save in user RAM, if data is meaningful |
| | MOVF | TXDATA, W | ;W reg = contents of TXDATA |
| | MOVWF | SSPBUF | ;New data to xmit |

19.3.3 ENABLING SPI I/O

To enable the serial port, SSP 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 \overline{SS} pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed 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
- \overline{SS} must have TRISA<5> bit set

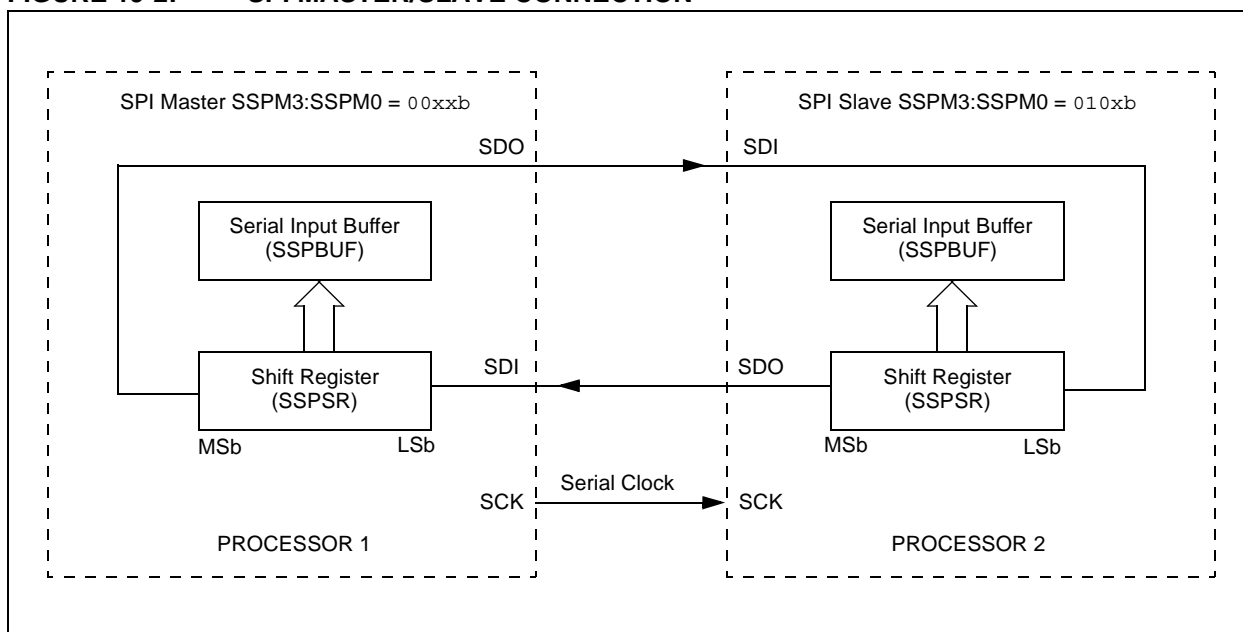
Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

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

FIGURE 19-2: SPI MASTER/SLAVE CONNECTION



PIC18F2455/2550/4455/4550

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 module is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications as a "Line Activity Monitor" mode.

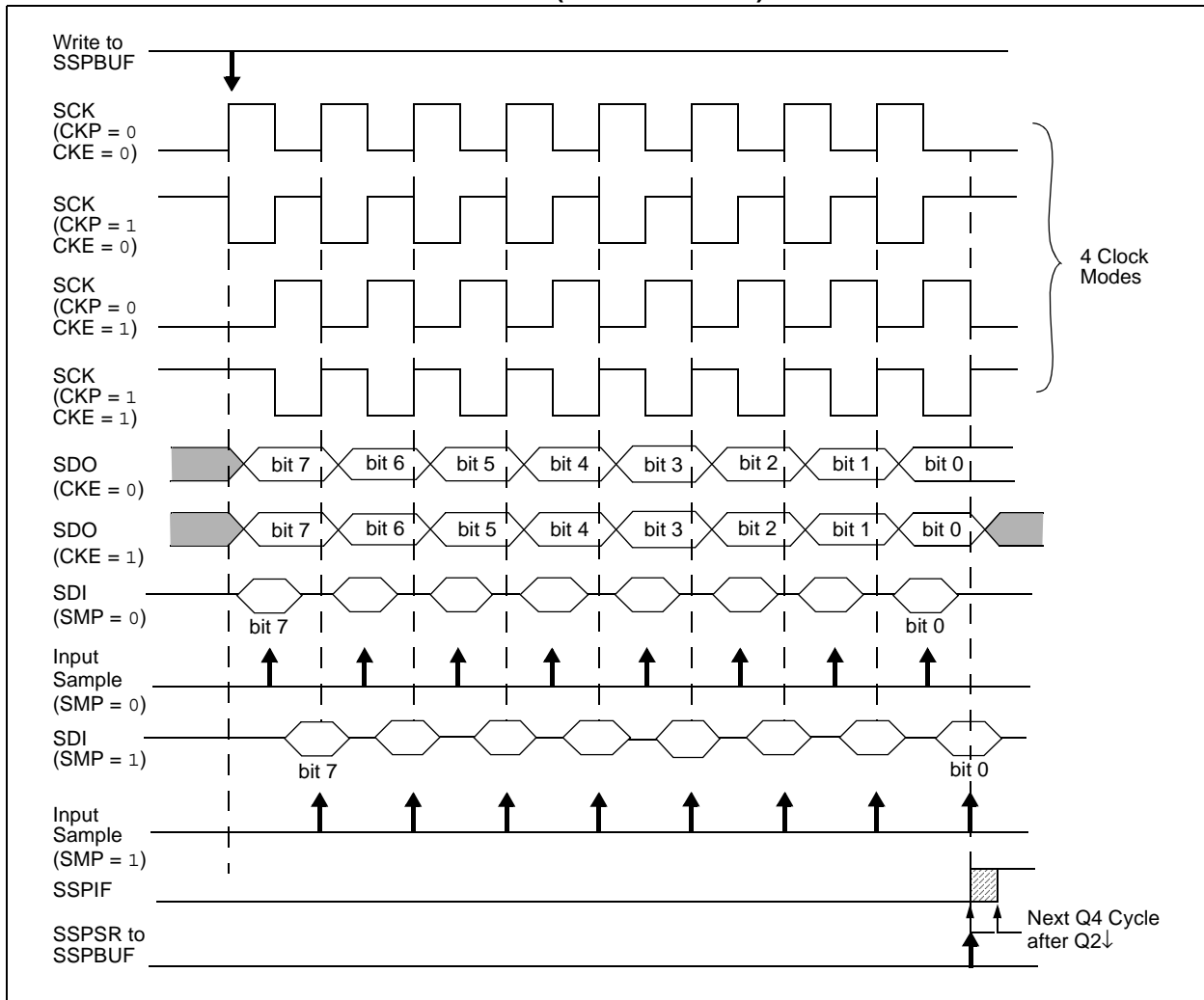
The clock polarity is selected by appropriately programming the CKP bit (SSPCON1<4>). This then, would give waveforms for SPI communication as shown in Figure 19-3, Figure 19-5 and Figure 19-6, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be 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 40 MHz) of 10.00 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.

FIGURE 19-3: SPI™ MODE WAVEFORM (MASTER MODE)



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.

Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCK pin. The Idle state is determined by the CKP bit (SSPCON1<4>).

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

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

19.3.7 SLAVE SELECT SYNCHRONIZATION

The \overline{SS} pin allows a Synchronous Slave mode. The SPI must be in Slave mode with \overline{SS} pin control enabled (SSPCON1<3:0> = 04h). The pin must not be driven low for the \overline{SS} pin to function as an input. The data latch

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

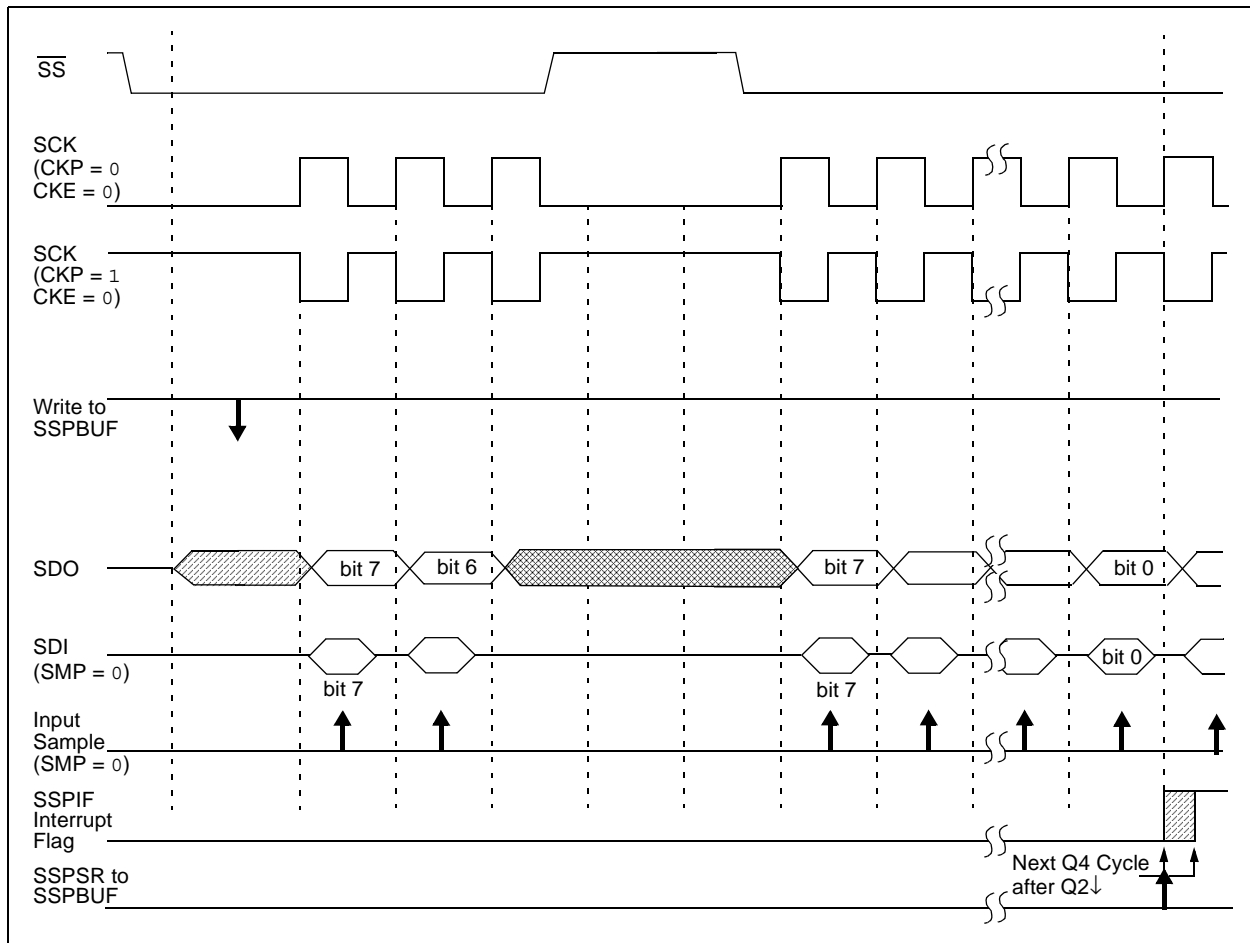
Note 1: When the SPI module is in Slave mode with \overline{SS} pin control enabled (SSPCON<3:0> = 0100), the SPI module will reset if the \overline{SS} pin is set to VDD.

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

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

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

FIGURE 19-4: SLAVE SYNCHRONIZATION WAVEFORM



PIC18F2455/2550/4455/4550

FIGURE 19-5: SPI™ MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

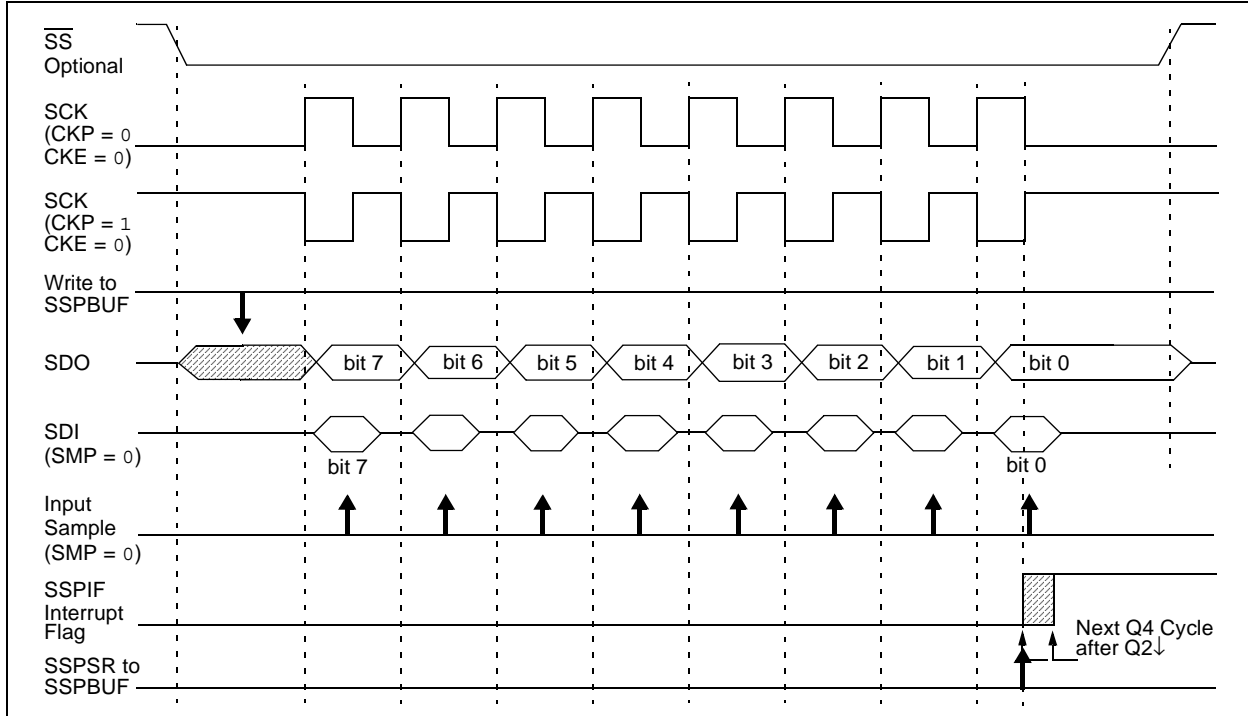
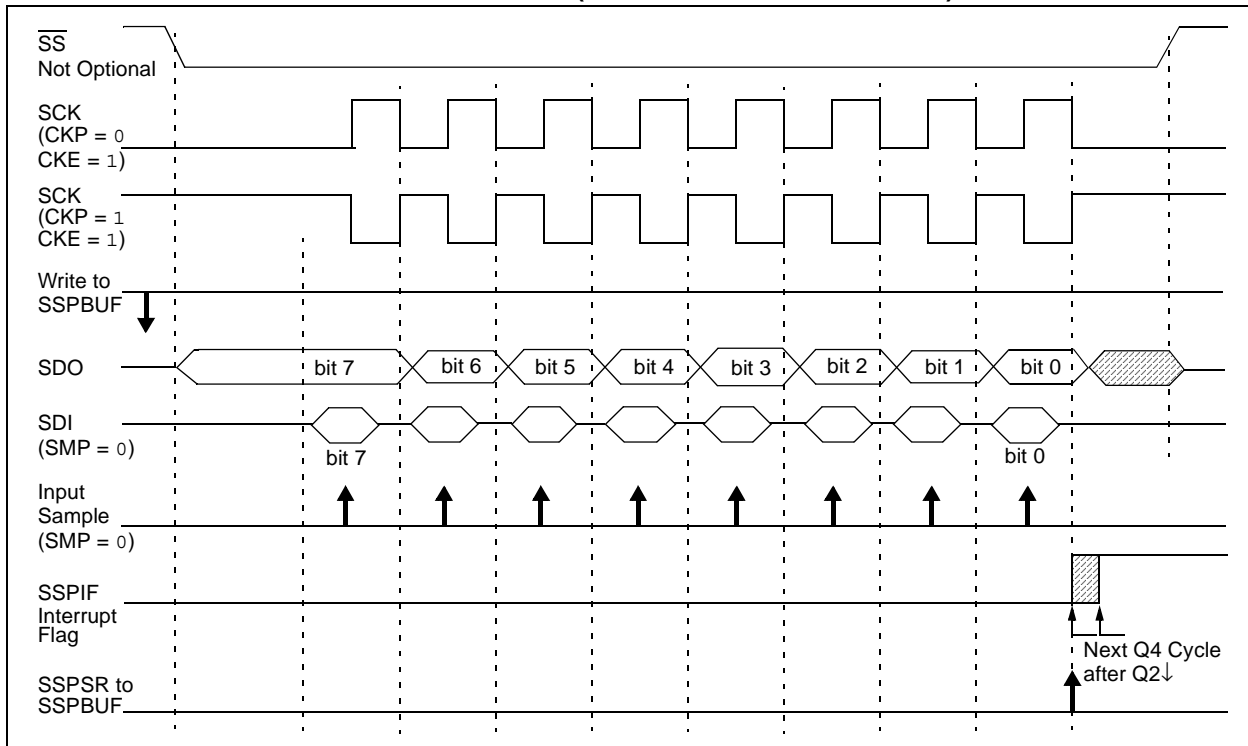


FIGURE 19-6: SPI™ MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



PIC18F2455/2550/4455/4550

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 power managed modes, a clock is provided to the peripherals. That clock should be from the primary clock source, the secondary clock (Timer1 oscillator at 32.768 kHz) 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 device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in any power managed mode and data to be shifted into the SPI Transmit/Receive Shift register. When all 8 bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device.

19.3.9 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 Terminology | Control Bits State | |
|-------------------------------|--------------------|-----|
| | CKP | CKE |
| 0, 0 | 0 | 1 |
| 0, 1 | 0 | 0 |
| 1, 0 | 1 | 1 |
| 1, 1 | 1 | 0 |

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

TABLE 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 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 54 |
| PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 54 |
| IPR1 | PSPIP ⁽¹⁾ | ADIP | RCIP | TXIP | SSPIP | CCP1IP | TMR2IP | TMR1IP | 54 |
| TRISA | PORTA Data Direction Register | | | | | | | | 54 |
| TRISB | PORTB Data Direction Register | | | | | | | | 54 |
| TRISC | PORTC Data Direction Register | | | | | | | | 54 |
| SSPBUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | 52 |
| SSPCON1 | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 52 |
| SSPSTAT | SMP | CKE | D/ \bar{A} | P | S | R/ \bar{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.

PIC18F2455/2550/4455/4550

19.4 I²C Mode

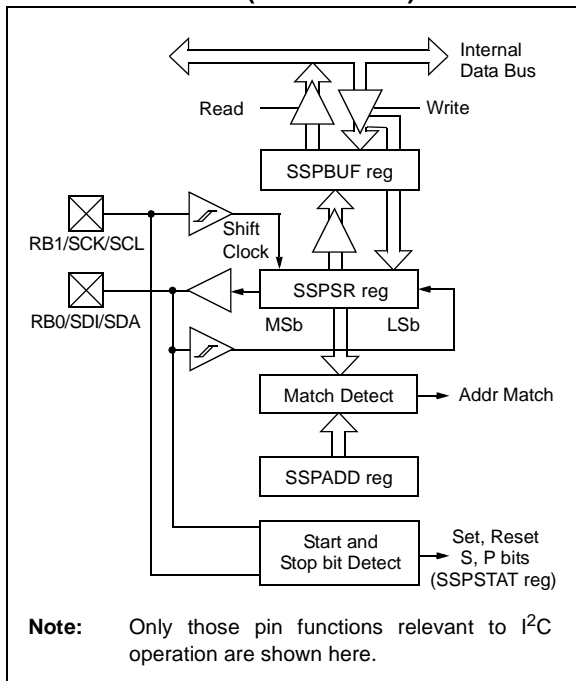
The MSSP module in I²C mode fully implements all master and slave functions (including general call support) and provides interrupts on Start and Stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications as well as 7-bit and 10-bit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

Two pins are used for data transfer:

- Serial clock (SCL) – RB1/AN10/INT1/SCK/SCL
- Serial data (SDA) – RB0/AN12/INT0/FLT0/SDI/SDA

The user must configure these pins as inputs or outputs through the TRISB<1:0> bits.

FIGURE 19-7: MSSP BLOCK DIAGRAM (I²C™ MODE)



19.4.1 REGISTERS

The MSSP module has six registers for I²C 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 I²C mode operation. The SSPCON1 and SSPCON2 registers are readable and writable. The lower 6 bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

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

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

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

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

PIC18F2455/2550/4455/4550

REGISTER 19-3: SSPSTAT: MSSP STATUS REGISTER (I²C MODE)

| | | | | | | | |
|-------|-------|-----|-----|-----|-----|-------|-----|
| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| SMP | CKE | D/A | P | S | R/W | UA | BF |
| bit 7 | | | | | | bit 0 | |

bit 7 **SMP:** Slew Rate Control bit

In Master or Slave mode:

- 1 = Slew rate control disabled for Standard Speed mode (100 kHz and 1 MHz)
- 0 = Slew rate control enabled for High-Speed mode (400 kHz)

bit 6 **CKE:** SMBus Select bit

In Master or Slave mode:

- 1 = Enable SMBus specific inputs
- 0 = Disable SMBus specific inputs

bit 5 **D/A:** Data/Address bit

In Master mode:

Reserved.

In Slave mode:

- 1 = Indicates that the last byte received or transmitted was data
- 0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** Stop bit

- 1 = Indicates that a Stop bit has been detected last
- 0 = Stop bit was not detected last

Note: This bit is cleared on Reset and when SSPEN is cleared.

bit 3 **S:** Start bit

- 1 = Indicates that a Start bit has been detected last
- 0 = Start bit was not detected last

Note: This bit is cleared on Reset and when SSPEN is cleared.

bit 2 **R/W:** Read/Write bit Information (I²C mode only)

In Slave mode:

- 1 = Read
- 0 = Write

Note: This bit holds the R/W bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or not ACK bit.

In Master mode:

- 1 = Transmit is in progress
- 0 = Transmit is not in progress

Note: ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in Idle mode.

bit 1 **UA:** Update Address 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 = Receive complete, SSPBUF is full
- 0 = Receive not complete, SSPBUF is empty

In Receive mode:

- 1 = Data transmit in progress (does not include the ACK and Stop bits), SSPBUF is full
- 0 = Data transmit complete (does not include the ACK and Stop bits), SSPBUF is empty

Legend:

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

PIC18F2455/2550/4455/4550

REGISTER 19-4: SSPCON1: MSSP CONTROL REGISTER 1 (I²C MODE)

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| bit 7 | | | | | | bit 0 | |

bit 7 **WCOL:** Write Collision Detect bit

In Master Transmit mode:

1 = A write to the SSPBUF register was attempted while the I²C conditions were not valid for a transmission to be started (must be cleared in software)

0 = No collision

In Slave Transmit mode:

1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)

0 = No collision

In Receive mode (Master or Slave modes):

This is a "don't care" bit.

bit 6 **SSPOV:** Receive Overflow Indicator bit

In Receive mode:

1 = A byte is received while the SSPBUF register is still holding the previous byte (must be cleared in software)

0 = No overflow

In Transmit mode:

This is a "don't care" bit in Transmit mode.

bit 5 **SSPEN:** Synchronous Serial Port Enable bit

1 = Enables the serial port and configures the SDA and SCL pins as the serial port pins

0 = Disables serial port and configures these pins as I/O port pins

Note: When enabled, the SDA and SCL pins must be properly configured as input or output.

bit 4 **CKP:** SCK Release Control bit

In Slave mode:

1 = Release clock

0 = Holds clock low (clock stretch), used to ensure data setup time

In Master mode:

Unused in this mode.

bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits

1111 = I²C Slave mode, 10-bit address with Start and Stop bit interrupts enabled

1110 = I²C Slave mode, 7-bit address with Start and Stop bit interrupts enabled

1011 = I²C Firmware Controlled Master mode (Slave Idle)

1000 = I²C Master mode, clock = FOSC/(4 * (SSPADD + 1))

0111 = I²C Slave mode, 10-bit address

0110 = I²C Slave mode, 7-bit address

Note: Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

PIC18F2455/2550/4455/4550

REGISTER 19-5: SSPCON2: MSSP CONTROL REGISTER 2 (I²C MODE)

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | |
|-------|---------|-------|----------------------|---------------------|--------------------|---------------------|--------------------|-------|
| GCEN | ACKSTAT | ACKDT | ACKEN ⁽¹⁾ | RCEN ⁽¹⁾ | PEN ⁽¹⁾ | RSEN ⁽¹⁾ | SEN ⁽¹⁾ | |
| bit 7 | | | | | | | | bit 0 |

- bit 7 **GCEN:** General Call Enable bit (Slave mode only)
 1 = Enable interrupt when a general call address (0000h) is received in the SSPSR
 0 = General call address disabled
- bit 6 **ACKSTAT:** Acknowledge Status bit (Master Transmit mode only)
 1 = Acknowledge was not received from slave
 0 = Acknowledge was received from slave
- bit 5 **ACKDT:** Acknowledge Data bit (Master Receive mode only)
 1 = Not Acknowledge
 0 = Acknowledge
- Note:** Value that will be transmitted when the user initiates an Acknowledge sequence at the end of a receive.
- bit 4 **ACKEN:** Acknowledge Sequence Enable bit (Master Receive mode only)⁽¹⁾
 1 = Initiate Acknowledge sequence on SDA and SCL pins and transmit ACKDT data bit. Automatically cleared by hardware.
 0 = Acknowledge sequence Idle
- bit 3 **RCEN:** Receive Enable bit (Master mode only)⁽¹⁾
 1 = Enables Receive mode for I²C
 0 = Receive Idle
- bit 2 **PEN:** Stop Condition Enable bit (Master mode only)⁽¹⁾
 1 = Initiate Stop condition on SDA and SCL pins. Automatically cleared by hardware.
 0 = Stop condition Idle
- bit 1 **RSEN:** Repeated Start Condition Enabled bit (Master mode only)⁽¹⁾
 1 = Initiate Repeated Start condition on SDA and SCL pins. Automatically cleared by hardware.
 0 = Repeated Start condition Idle
- bit 0 **SEN:** Start Condition Enabled/Stretch Enabled bit⁽¹⁾
In Master mode:
 1 = Initiate Start condition on SDA and SCL pins. Automatically cleared by hardware.
 0 = Start condition Idle
In Slave mode:
 1 = Clock stretching is enabled for both slave transmit and slave receive (stretch enabled)
 0 = Clock stretching is disabled
- Note 1:** For bits ACKEN, RCEN, PEN, RSEN, SEN: If the I²C module is not in the Idle mode, these bits may not be set (no spooling) and the SSPBUF may not be written (or writes to the SSPBUF are disabled).

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 |

PIC18F2455/2550/4455/4550

19.4.2 OPERATION

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

The SSPCON1 register allows control of the I²C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I²C modes to be selected:

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

Selection of any I²C mode with the SSPEN bit set, forces the SCL and SDA pins to be open-drain, provided these pins are programmed to inputs by setting the appropriate TRISB 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 (TRISB<1:0> set). The MSSP module will override the input state with the output data when required (slave-transmitter).

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

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

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

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

In this case, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF (PIR1<3>) is set. The BF bit is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the I²C specification, as well as the requirement of the MSSP module, are shown in timing parameter #100 and parameter #101.

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:

1. The SSPSR register value is loaded into the SSPBUF register.
2. The Buffer Full bit, BF, is set.
3. An \overline{ACK} pulse is generated.
4. MSSP Interrupt Flag bit, SSPIF (PIR1<3>), is set (interrupt is generated, if enabled) on the falling edge of the ninth SCL pulse.

In 10-bit Address mode, two address bytes need to be received by the slave. The five Most Significant bits (MSBs) of the first address byte specify if this is a 10-bit address. Bit R/W (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '11110 A9 A8 0', where 'A9' and 'A8' are the two MSBs of the address. The sequence of events for 10-bit address is as follows, with steps 7 through 9 for the slave-transmitter:

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

19.4.3.2 Reception

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

When the address byte overflow condition exists, then the no Acknowledge (\overline{ACK}) pulse is given. An overflow condition is defined as either bit, BF (SSPSTAT<0>), is set, or bit, SSPOV (SSPCON1<6>), is set.

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

If SEN is enabled (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 (SSPCON<4>). See **Section 19.4.4 “Clock Stretching”** for more detail.

19.4.3.3 Transmission

When the $\overline{R/W}$ bit of the incoming address byte is set and an address match occurs, the $\overline{R/W}$ bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The \overline{ACK} pulse will be sent on the ninth bit and pin 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 19-9).

The \overline{ACK} pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line is high (not \overline{ACK}), then the data transfer is complete. In this case, when the \overline{ACK} is latched by the slave, the slave logic is reset (resets SSPSTAT register) and the slave monitors for another occurrence of the Start bit. If the SDA line was low (\overline{ACK}), the next transmit data must be loaded into the SSPBUF register. Again, pin RB1/AN10/INT1/SCK/SCL must be enabled by setting bit CKP.

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

PIC18F2455/2550/4455/4550

FIGURE 19-8: I²C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 7-BIT ADDRESS)

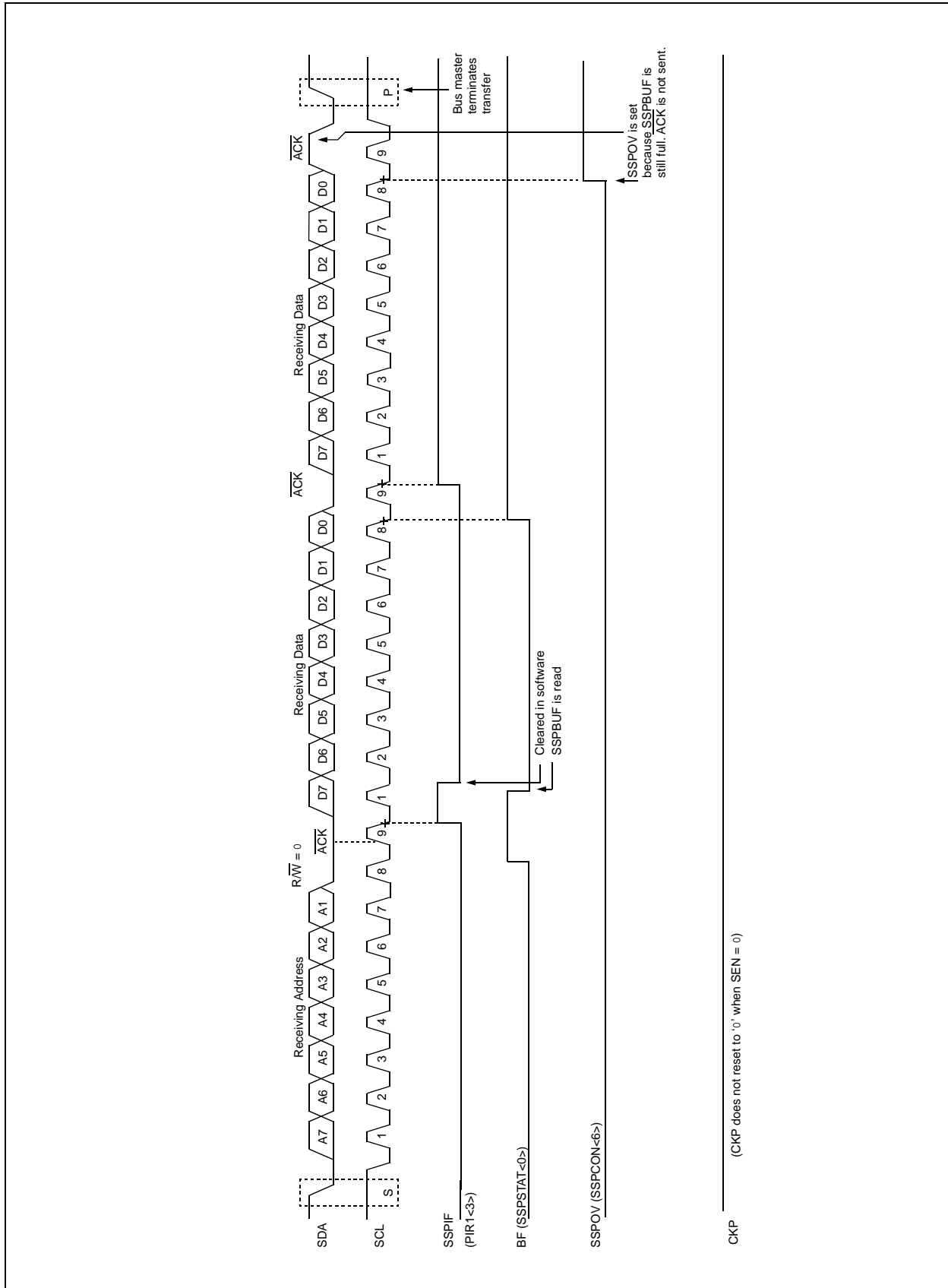
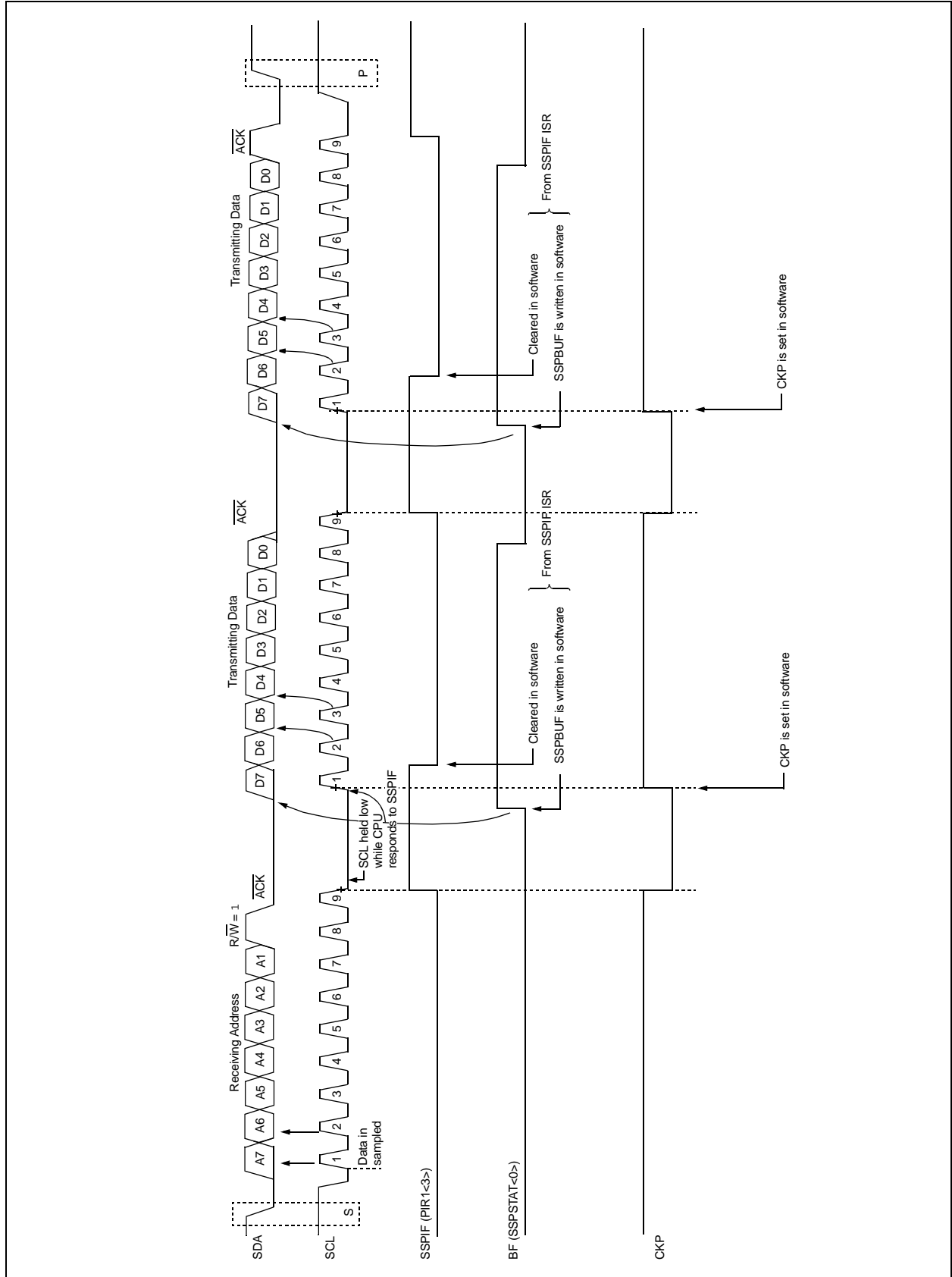


FIGURE 19-9: I²C™ SLAVE MODE TIMING (TRANSMISSION, 7-BIT ADDRESS)



PIC18F2455/2550/4455/4550

FIGURE 19-10: I²C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 10-BIT ADDRESS)

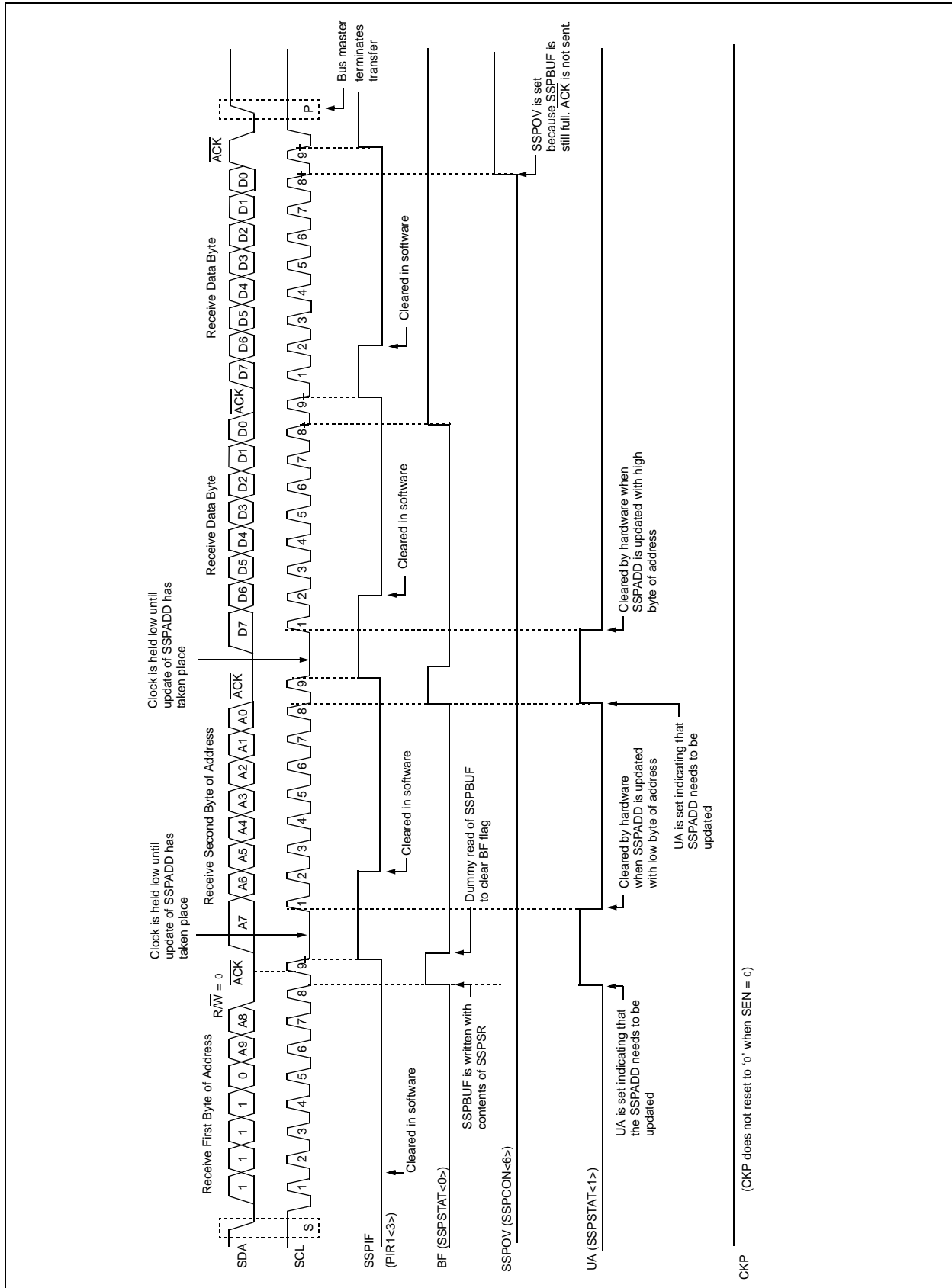
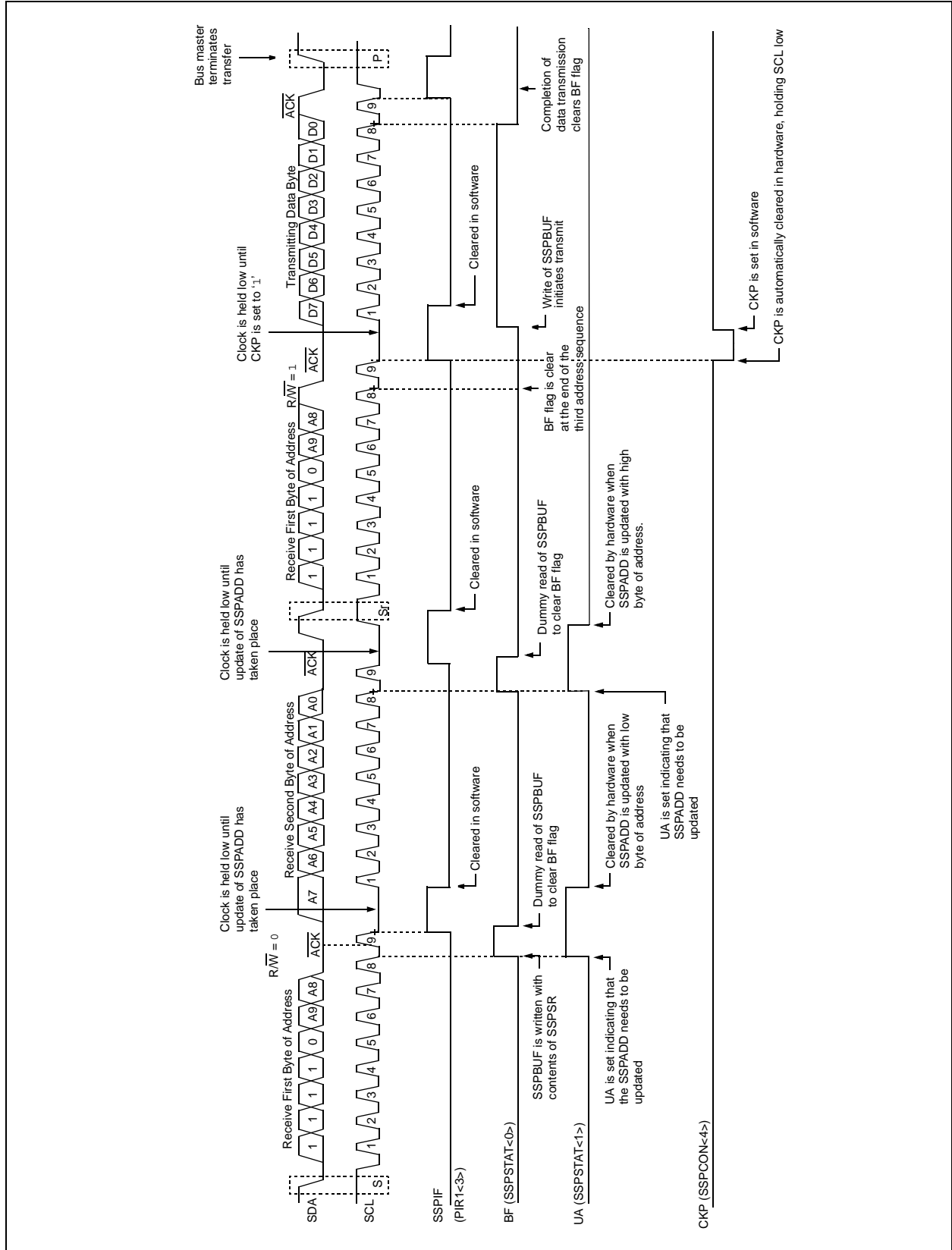


FIGURE 19-11: I²C™ SLAVE MODE TIMING (TRANSMISSION, 10-BIT ADDRESS)



PIC18F2455/2550/4455/4550

19.4.4 CLOCK STRETCHING

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

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

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

In 7-bit Slave Receive mode, on the falling edge of the ninth clock at the end of the ACK sequence if the BF bit is set, the CKP bit in the SSPCON1 register is automatically cleared, forcing the SCL output to be held low. The CKP 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 19-13).

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

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

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

In 10-bit Slave Receive mode during the address sequence, clock stretching automatically takes place but CKP is not cleared. During this time, if the UA bit is set after the ninth clock, clock stretching is initiated. The UA bit is set after receiving the upper byte of the 10-bit address and following the receive of the second byte of the 10-bit address with the R/W bit cleared to '0'. The release of the clock line occurs upon updating 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 19-9).

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

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

19.4.4.4 Clock Stretching for 10-bit Slave Transmit Mode

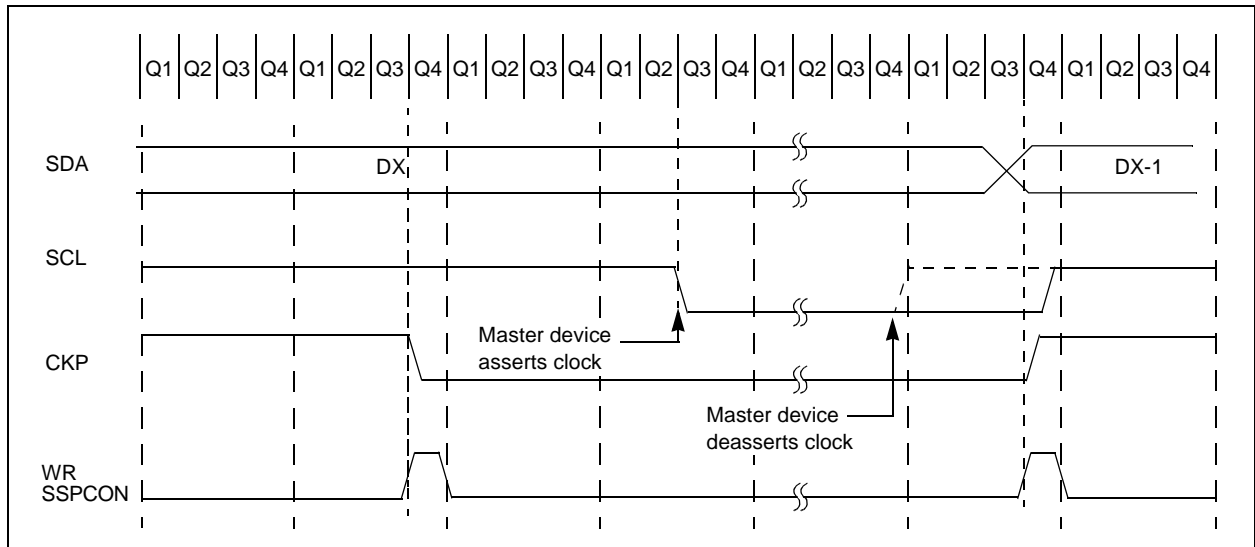
In 10-bit Slave Transmit mode, clock stretching is controlled during the first two address sequences by the state of the UA bit, just as it is in 10-bit Slave Receive mode. The first two addresses are followed by a third address sequence which contains the high-order bits of the 10-bit address and the R/W bit set to '1'. After the third address sequence is performed, the UA bit is not set, the module is now configured in Transmit mode and clock stretching is controlled by the BF flag as in 7-bit Slave Transmit mode (see Figure 19-11).

19.4.4.5 Clock Synchronization and the CKP bit

When the CKP bit is cleared, the SCL output is forced to '0'. However, setting the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external I²C master device has

already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I²C bus have deasserted SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 19-12).

FIGURE 19-12: CLOCK SYNCHRONIZATION TIMING



PIC18F2455/2550/4455/4550

FIGURE 19-13: I²C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 7-BIT ADDRESS)

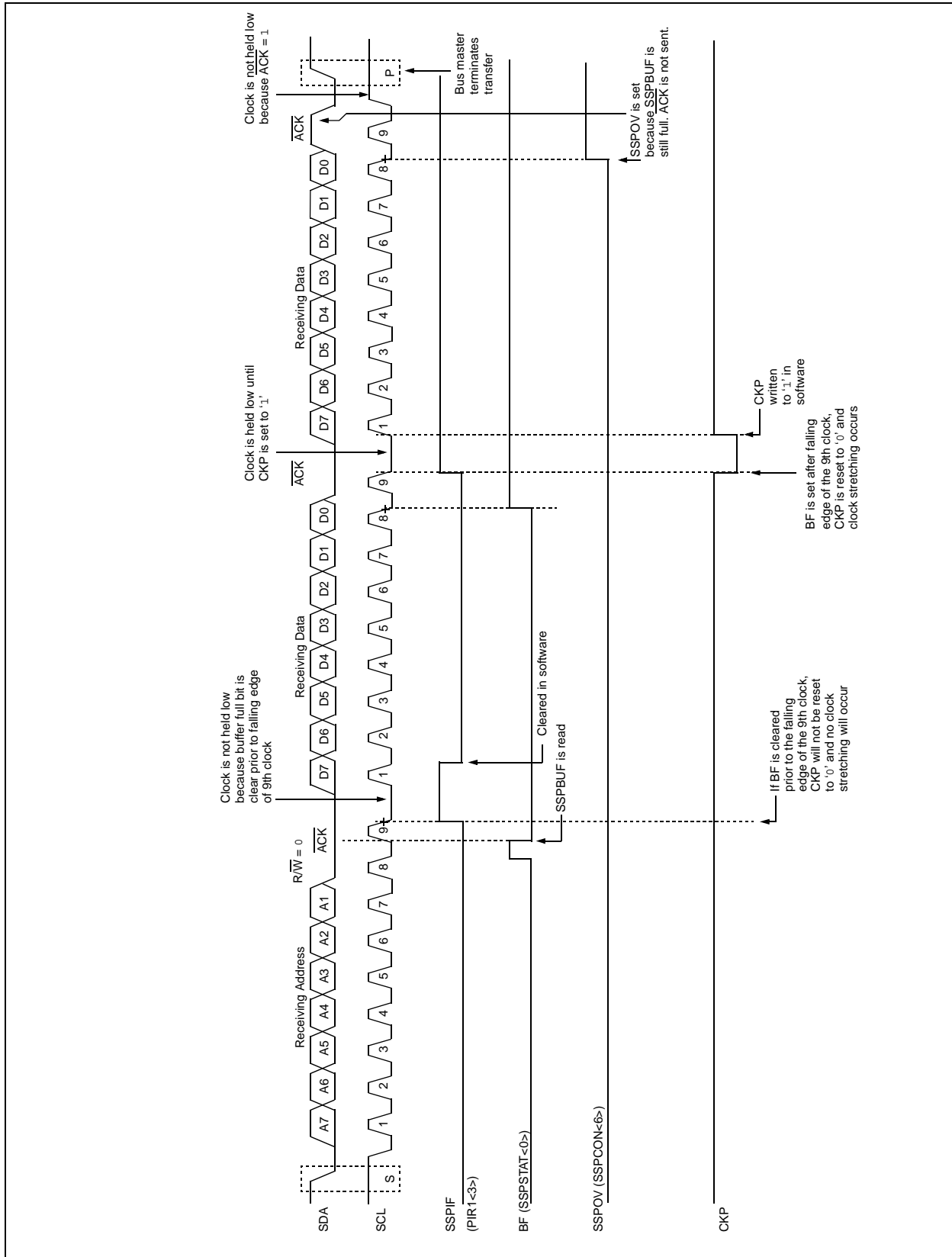
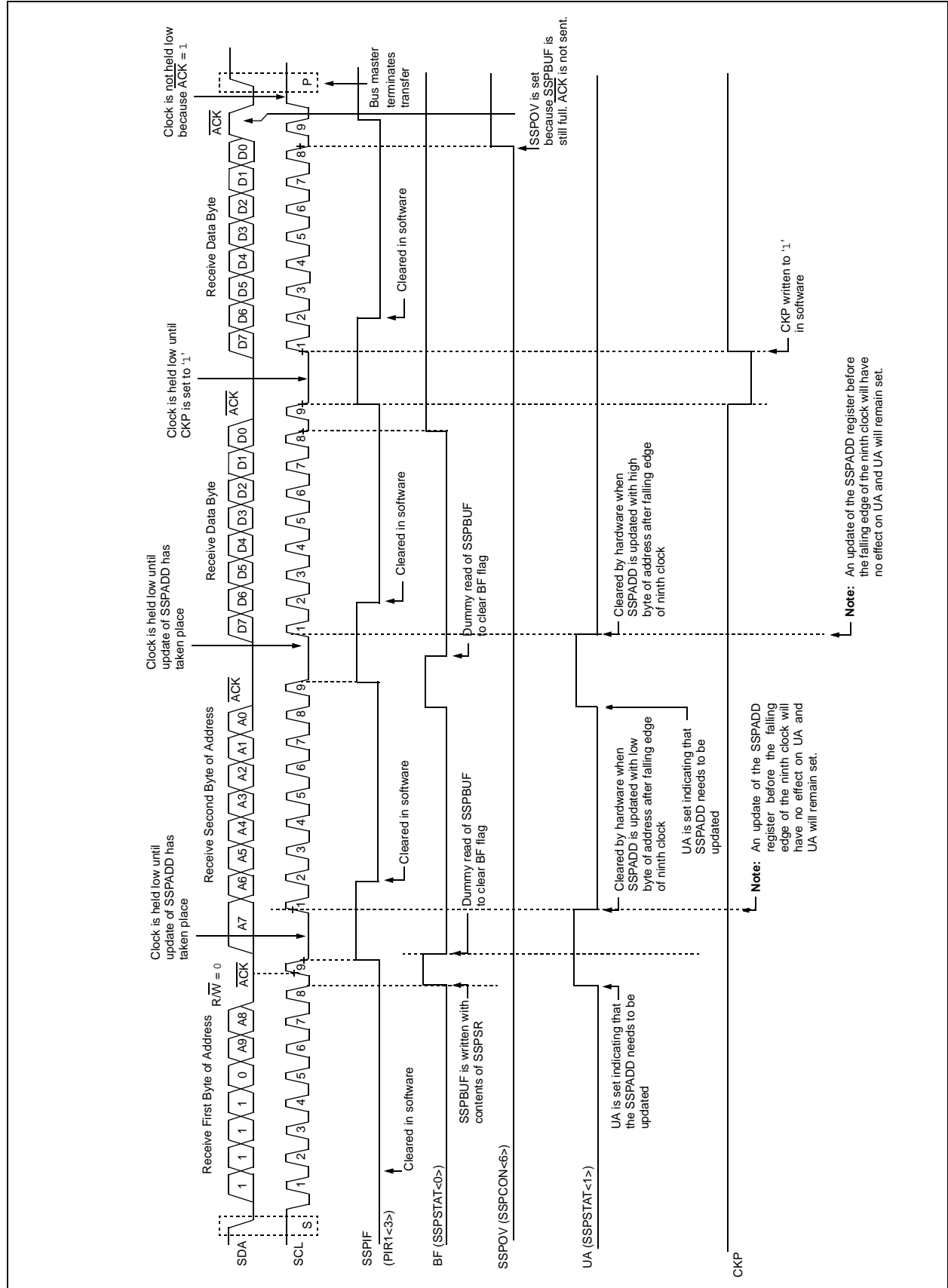


FIGURE 19-14: I²C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 10-BIT ADDRESS)



PIC18F2455/2550/4455/4550

19.4.5 GENERAL CALL ADDRESS SUPPORT

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

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

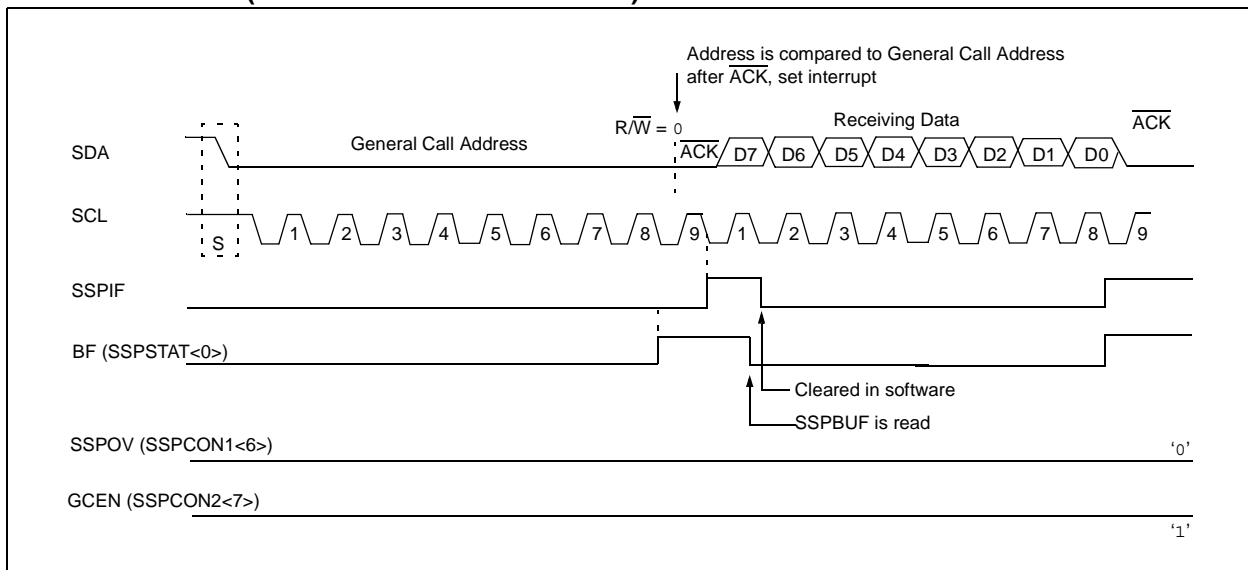
The general call address is recognized when the General Call Enable bit (GCEN) is enabled (SSPCON2<7> set). Following a Start bit detect, 8 bits are shifted into the SSPSR and the address is compared against the SSPADD. It is also compared to the general call address and fixed in hardware.

If the general call address matches, the SSPSR is transferred to the SSPBUF, the BF flag bit is set (eighth bit) and on the falling edge of the ninth bit (\overline{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 19-15).

FIGURE 19-15: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE (7 OR 10-BIT ADDRESS MODE)



PIC18F2455/2550/4455/4550

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.

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

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

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

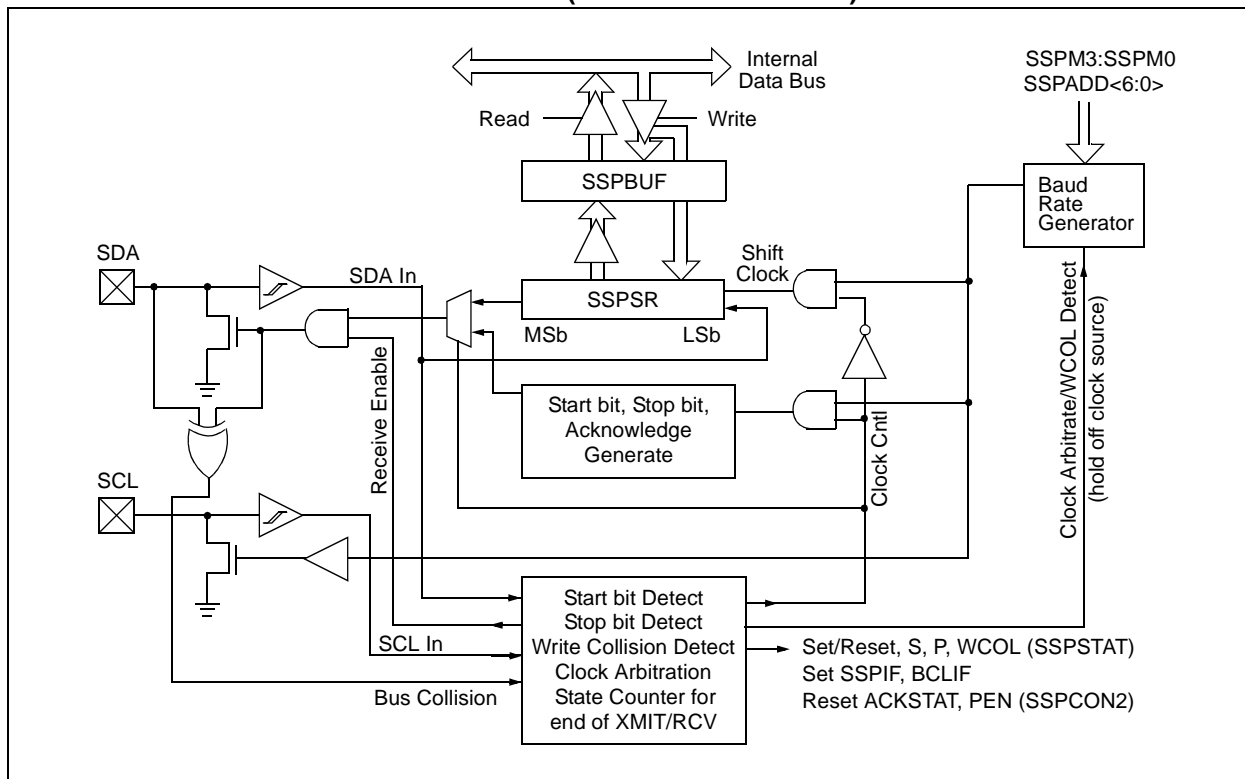
1. Assert a Start condition on SDA and SCL.
2. Assert a Repeated Start condition on SDA and SCL.
3. Write to the SSPBUF register initiating transmission of data/address.
4. Configure the I²C port to receive data.
5. Generate an Acknowledge condition at the end of a received byte of data.
6. Generate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

Note: The MSSP module, when configured in I²C Master mode, does not allow 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 SSP Interrupt Flag bit, SSPIF, to be set (SSP interrupt, if enabled):

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

FIGURE 19-16: MSSP BLOCK DIAGRAM (I²C™ MASTER MODE)



PIC18F2455/2550/4455/4550

19.4.6.1 I²C Master Mode Operation

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

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/W) bit. In this case, the R/W bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/W bit. In this case, the R/W bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

The Baud Rate Generator used for the SPI mode operation is used to set the SCL clock frequency for either 100 kHz, 400 kHz or 1 MHz I²C operation. See **Section 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>).
2. SSPIF is set. The MSSP module will wait the required start time before any other operation takes place.
3. The user loads the SSPBUF with the slave address to transmit.
4. Address is shifted out the SDA pin until all 8 bits are transmitted.
5. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
6. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
7. The user loads the SSPBUF with eight bits of data.
8. Data is shifted out the SDA pin until all 8 bits are transmitted.
9. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
10. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
11. The user generates a Stop condition by setting the Stop Enable bit, PEN (SSPCON2<2>).
12. Interrupt is generated once the Stop condition is complete.

PIC18F2455/2550/4455/4550

19.4.7 BAUD RATE

In I²C Master mode, the Baud Rate Generator (BRG) reload value is placed in the lower 7 bits of the SSPADD register (Figure 19-17). When a write occurs to SSPBUF, the Baud Rate Generator will automatically begin counting. The BRG counts down to '0' and stops until another reload has taken place. The BRG count is decremented twice per instruction cycle (TCY) on the Q2 and Q4 clocks. In I²C Master mode, the BRG is reloaded automatically.

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

Table 19-3 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPADD.

FIGURE 19-17: BAUD RATE GENERATOR BLOCK DIAGRAM

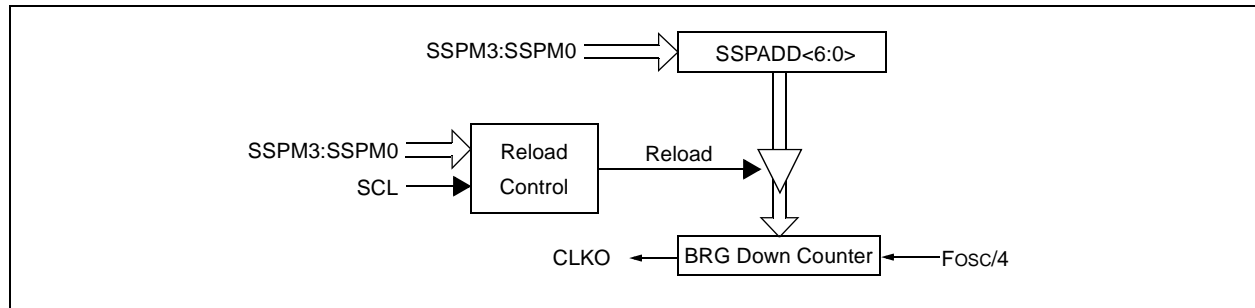


TABLE 19-3: I²C CLOCK RATE W/BRG

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

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

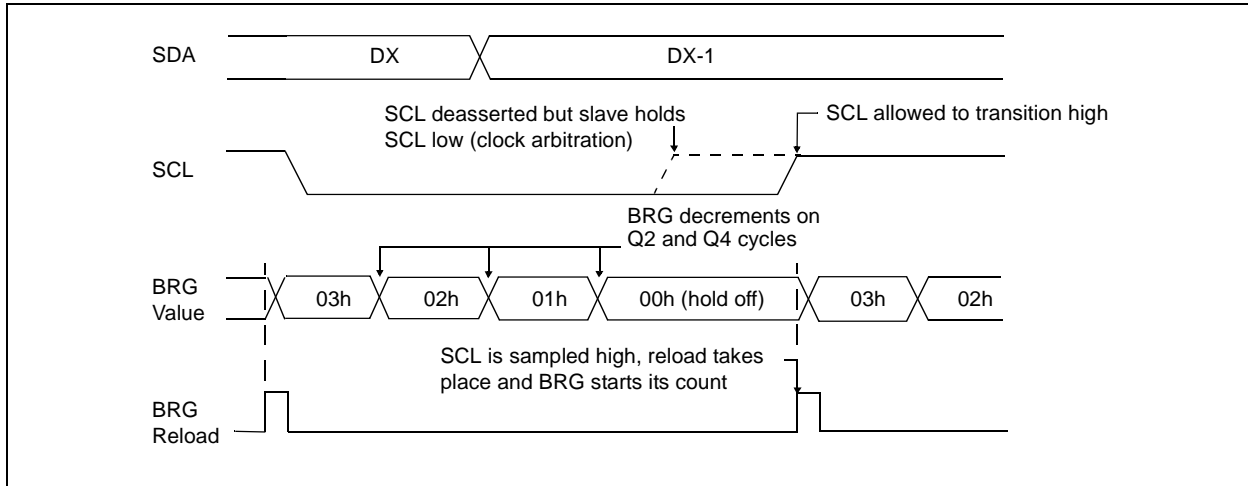
PIC18F2455/2550/4455/4550

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 19-18).

FIGURE 19-18: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION



19.4.8 I²C MASTER MODE START CONDITION TIMING

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

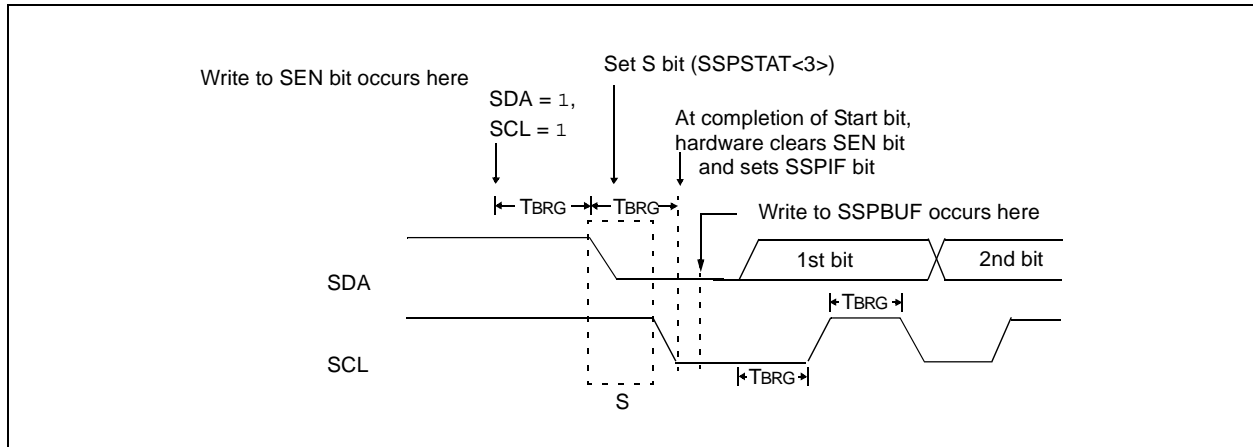
Note: If at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLIF, is set, the Start condition is aborted and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

19.4.8.1 WCOL Status Flag

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

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

FIGURE 19-19: FIRST START BIT TIMING



PIC18F2455/2550/4455/4550

19.4.9 I²C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition occurs when the RSEN bit (SSPCON2<1>) is programmed high and the I²C logic module is in the Idle state. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded with the contents of SSPADD<5:0> and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. Following this, the RSEN bit (SSPCON2<1>) will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit (SSPSTAT<3>) will be set. The SSPIF bit will not be set until the Baud Rate Generator has timed out.

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

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

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

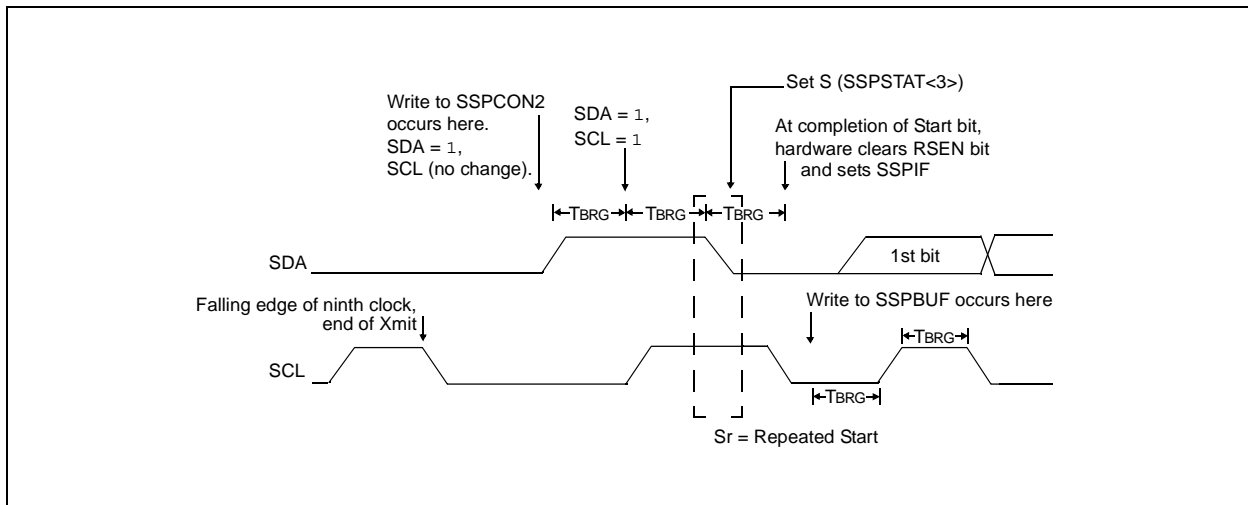
Immediately following the SSPIF bit getting set, the user may write the SSPBUF with the 7-bit address in 7-bit mode or the default first address in 10-bit mode. After the first eight bits are transmitted and an ACK is received, the user may then transmit an additional eight bits of address (10-bit mode)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

19.4.9.1 WCOL Status Flag

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

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

FIGURE 19-20: REPEAT START CONDITION WAVEFORM



19.4.10 I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the 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 19-21).

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

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 8 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 is set and the contents of the buffer are unchanged (the write doesn't occur).

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 (ACK = 0) and is set when the slave does not Acknowledge (ACK = 1). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

19.4.11 I²C MASTER MODE RECEPTION

Master mode reception is enabled by programming the Receive Enable bit, RCEN (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

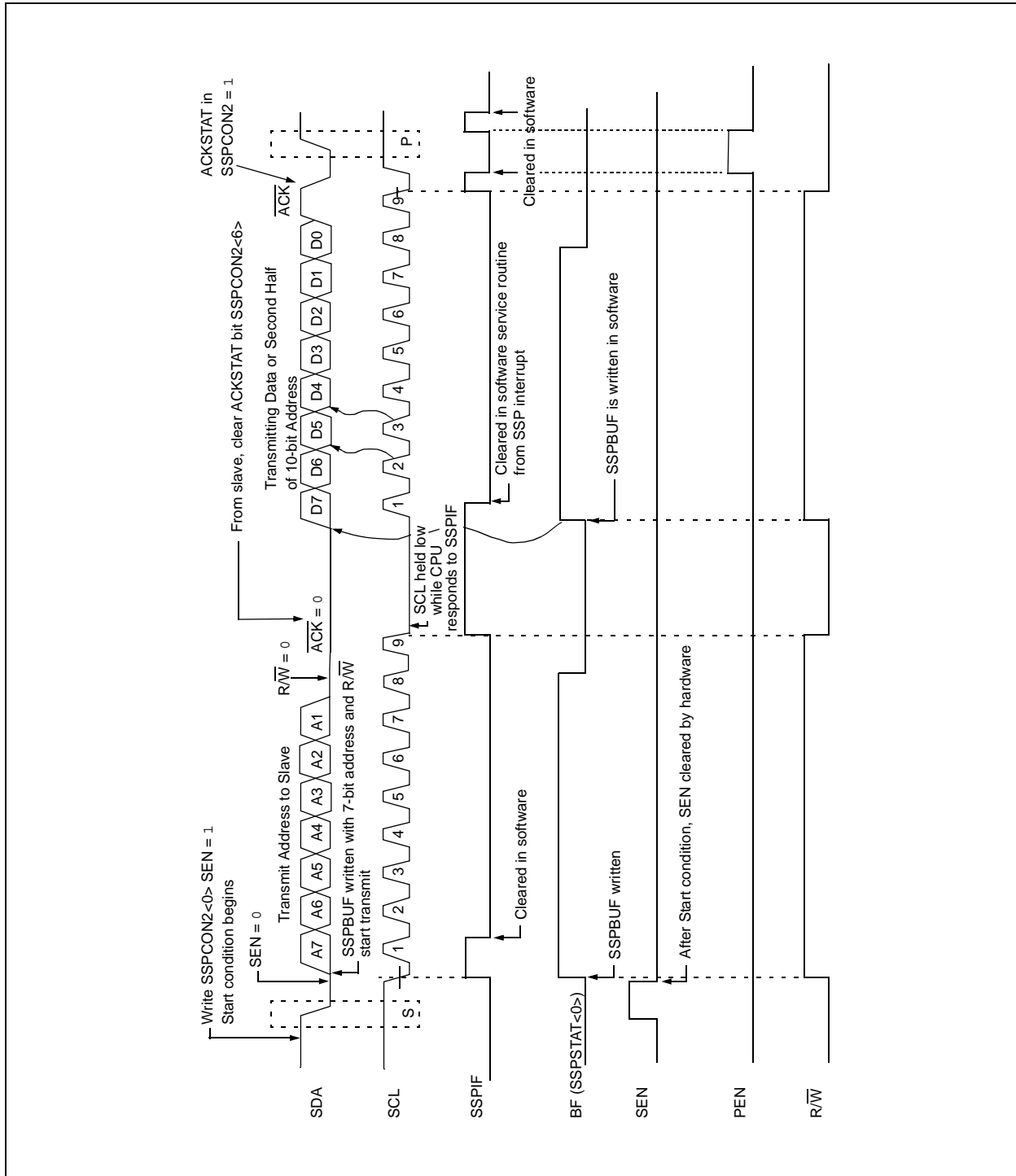
In receive operation, the SSPOV bit is set when 8 bits are received into the SSPSR and the BF flag bit is already set from a previous reception.

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).

PIC18F2455/2550/4455/4550

FIGURE 19-21: I²C™ MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)



PIC18F2455/2550/4455/4550

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 Idle mode (Figure 19-23).

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 19-24).

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 19-23: ACKNOWLEDGE SEQUENCE WAVEFORM

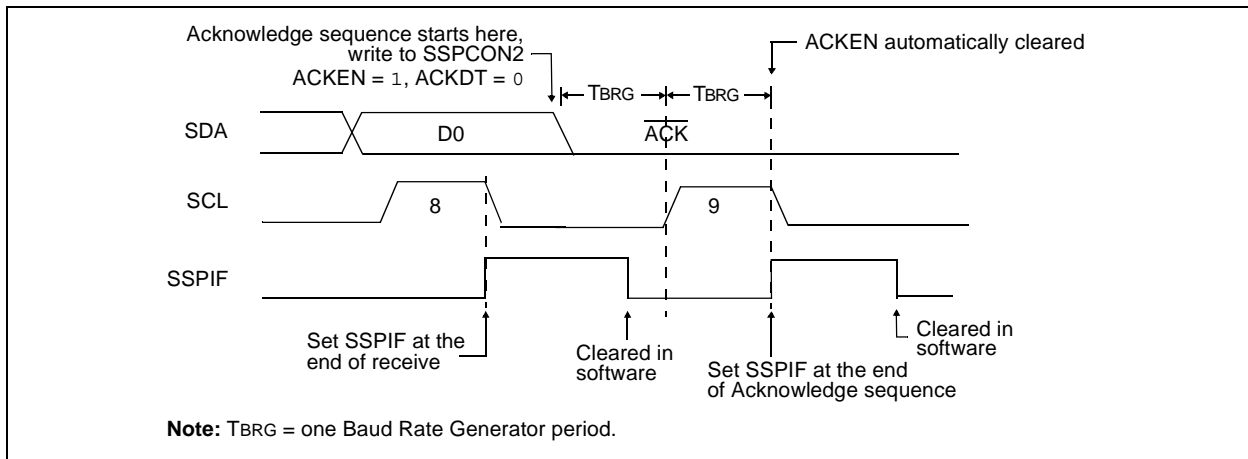
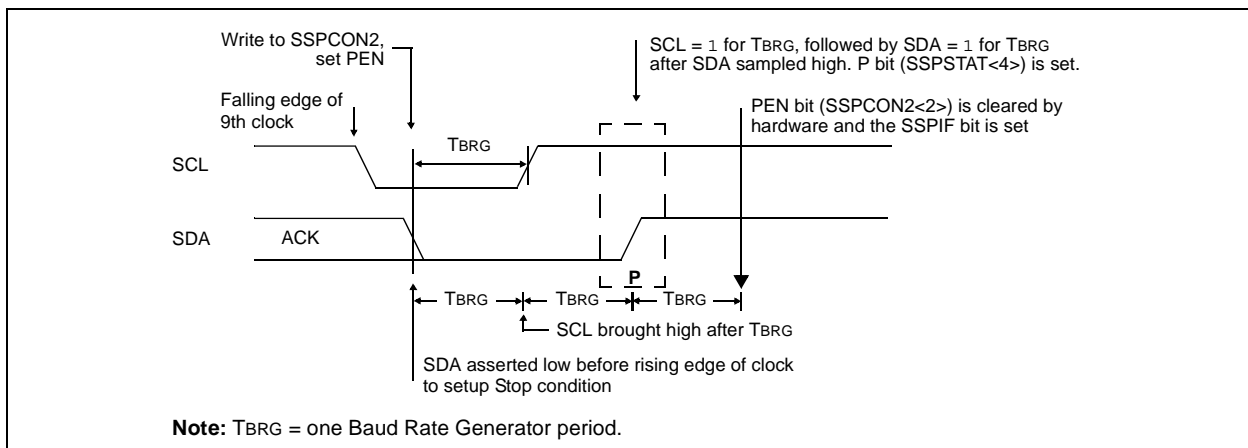


FIGURE 19-24: STOP CONDITION RECEIVE OR TRANSMIT MODE



19.4.14 SLEEP OPERATION

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

19.4.15 EFFECT OF A RESET

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

19.4.16 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I²C bus may be taken when the P bit (SSPSTAT<4>) is set, or the bus is Idle, with both the S and P bits clear. When the bus is busy, enabling the SSP interrupt will generate the interrupt when the Stop condition occurs.

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

The states where arbitration can be lost are:

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

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 I²C port to its Idle state (Figure 19-25).

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

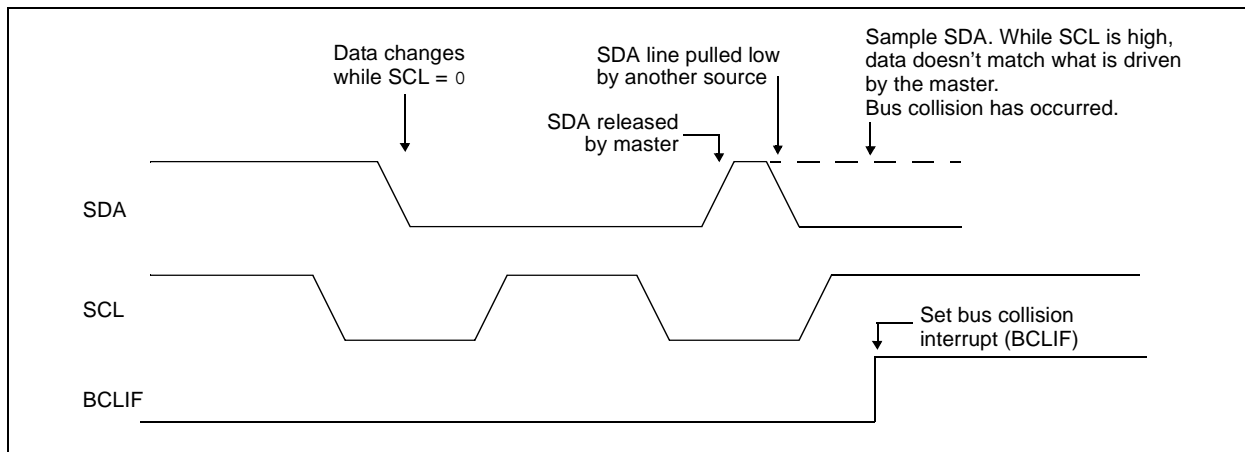
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision Interrupt Service Routine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

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

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

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I²C bus can be taken when the P bit is set in the SSPSTAT register, or the bus is Idle and the S and P bits are cleared.

FIGURE 19-25: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE



PIC18F2455/2550/4455/4550

19.4.17.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

- SDA or SCL are sampled low at the beginning of the Start condition (Figure 19-26).
- SCL is sampled low before SDA is asserted low (Figure 19-27).

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

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

- the Start condition is aborted,
- the BCLIF flag is set and
- the MSSP module is reset to its Idle state (Figure 19-26).

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

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 19-28). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to '0' and during this time, if the SCL pins are sampled as '0', a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

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

FIGURE 19-26: BUS COLLISION DURING START CONDITION (SDA ONLY)

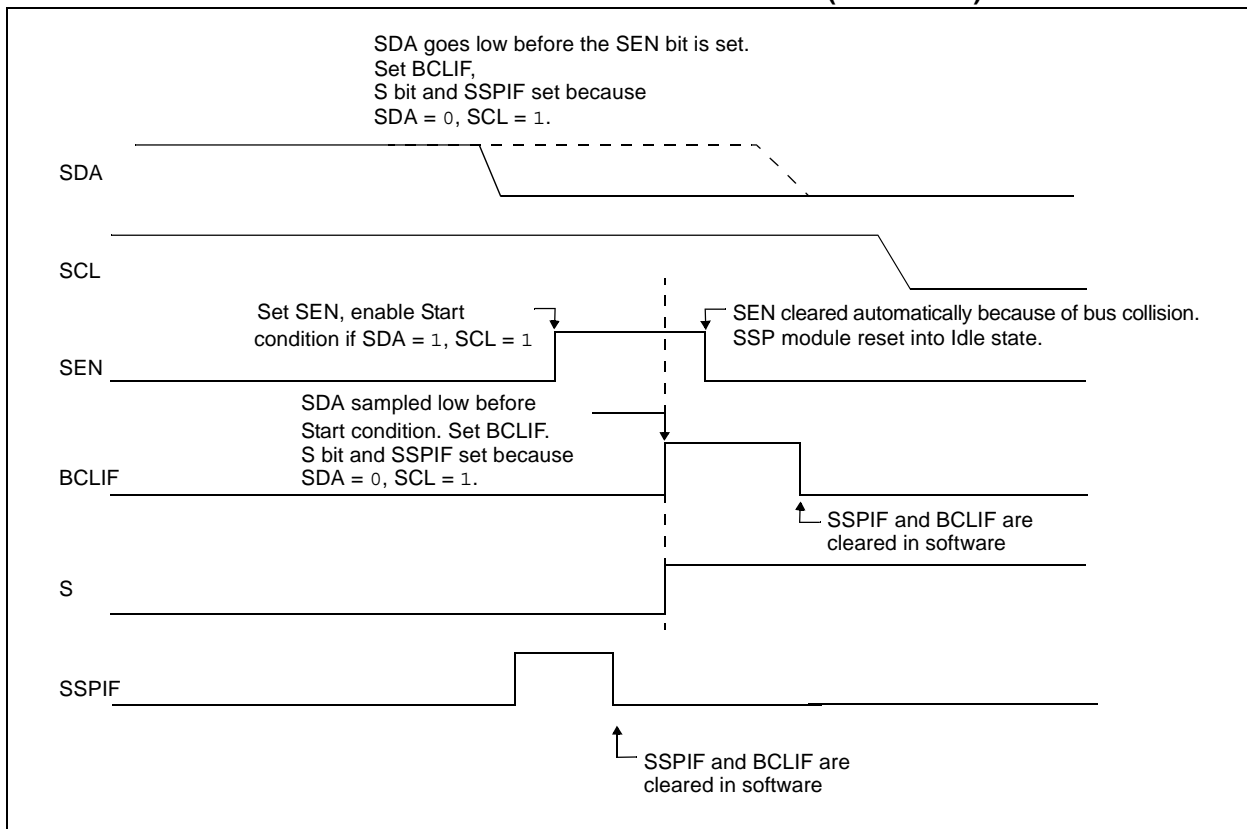


FIGURE 19-27: BUS COLLISION DURING START CONDITION (SCL = 0)

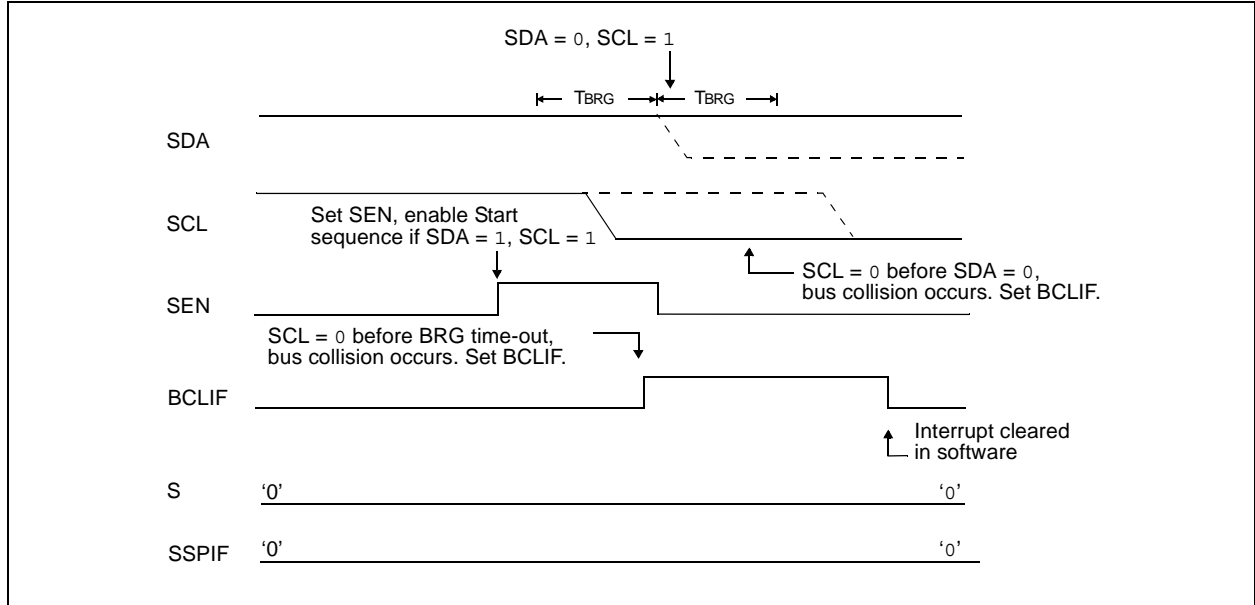
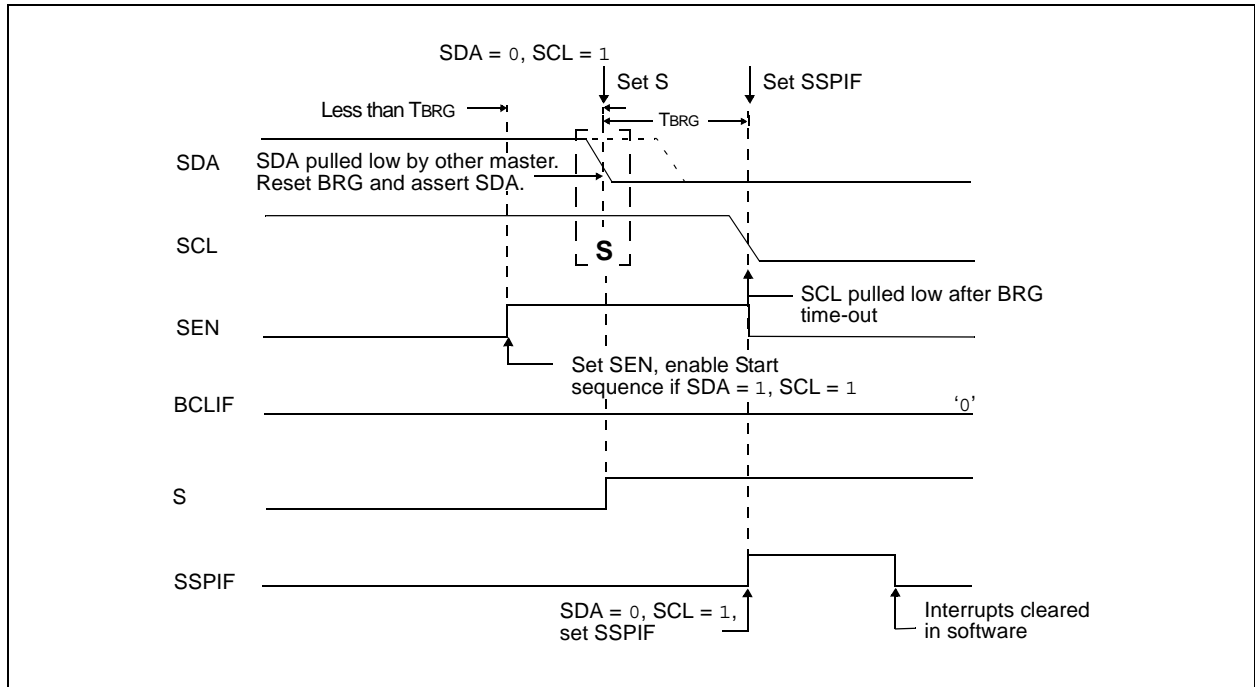


FIGURE 19-28: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION



PIC18F2455/2550/4455/4550

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 19-29). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition (see Figure 19-30).

If, at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

FIGURE 19-29: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

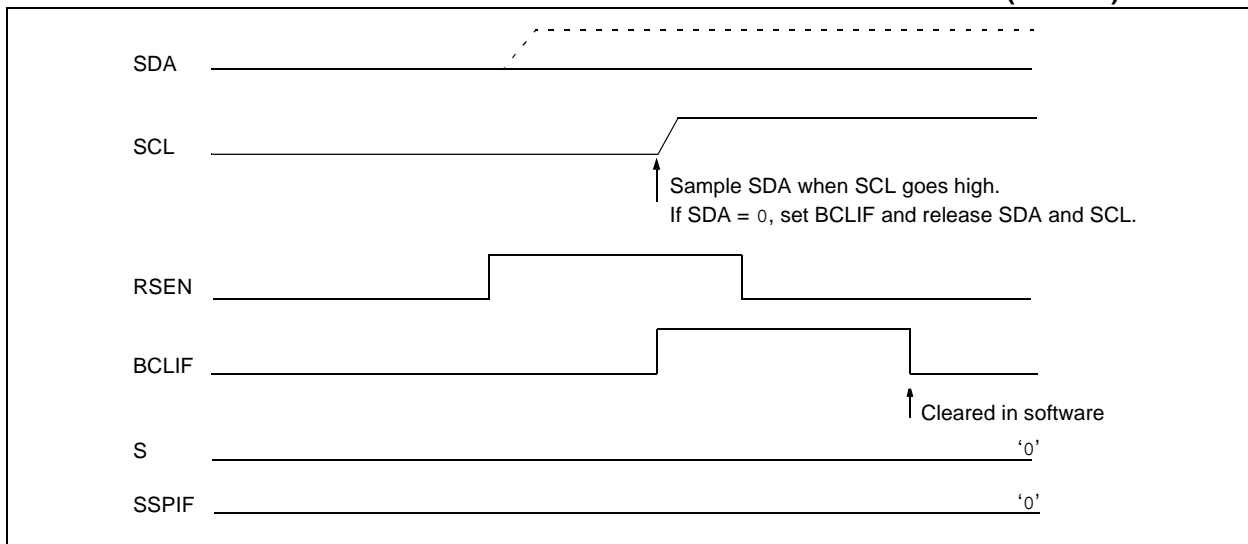
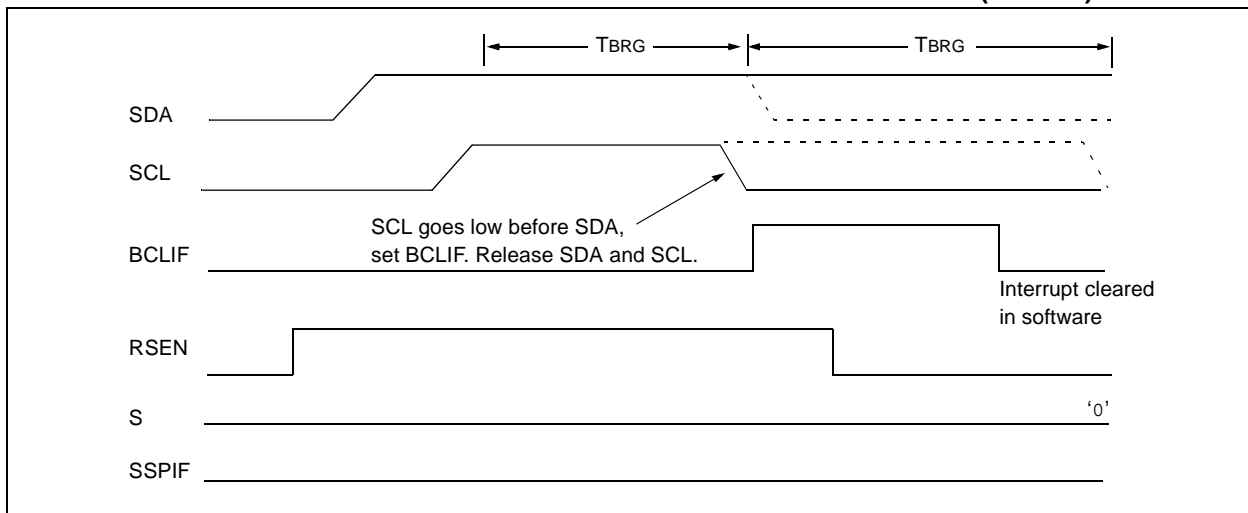


FIGURE 19-30: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)



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.
- 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 19-31). If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 19-32).

FIGURE 19-31: BUS COLLISION DURING A STOP CONDITION (CASE 1)

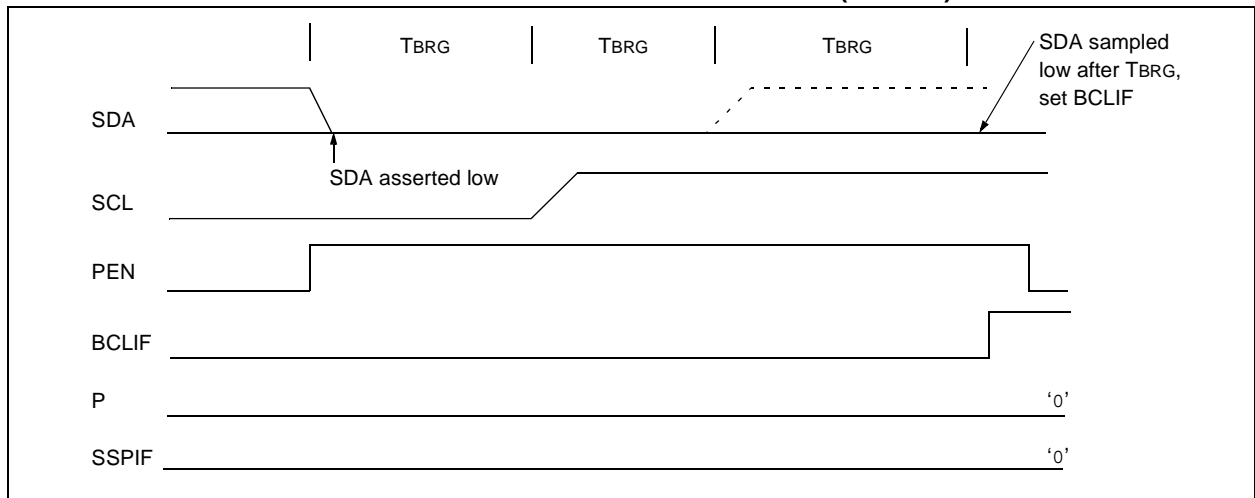
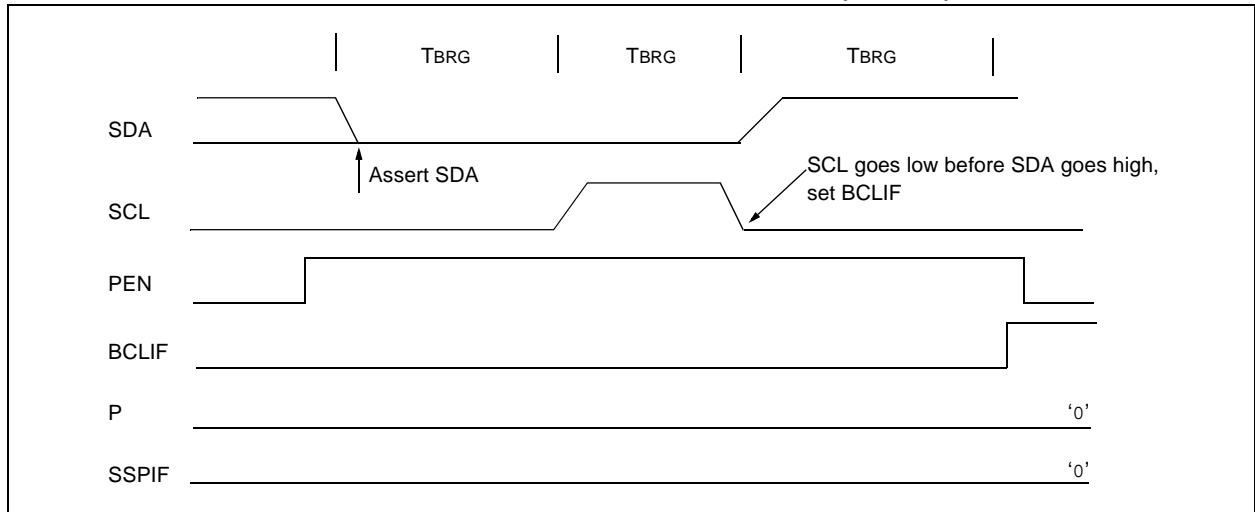


FIGURE 19-32: BUS COLLISION DURING A STOP CONDITION (CASE 2)



PIC18F2455/2550/4455/4550

NOTES:

20.0 ENHANCED UNIVERSAL SYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

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

The EUSART 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 character reception
 - Auto-baud calibration
 - 12-bit Break character transmission
- Synchronous – Master (half-duplex) with selectable clock polarity
- Synchronous – Slave (half-duplex) with selectable clock polarity

The pins of the Enhanced USART are multiplexed with PORTC. In order to configure RC6/TX/CK and RC7/RX/DT/SDO as a USART:

- bit SPEN (RCSTA<7>) must be set (= 1)
- bit TRISC<7> must be set (= 1)
- bit TRISC<6> must be cleared (= 0) for Asynchronous and Synchronous Master modes, or set (= 1) for Synchronous Slave mode

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.

PIC18F2455/2550/4455/4550

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 | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D |

bit 7 bit 0

bit 7 **CSRC:** Clock Source Select bit

Asynchronous mode:

Don't care.

Synchronous mode:

1 = Master mode (clock generated internally from BRG)

0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit

1 = Selects 9-bit transmission

0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit

1 = Transmit enabled

0 = Transmit disabled

Note: SREN/CREN overrides TXEN in Sync mode.

bit 4 **SYNC:** EUSART Mode Select bit

1 = Synchronous mode

0 = 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.

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

PIC18F2455/2550/4455/4550

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 | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D |
| bit 7 | | | | | | bit 0 | |

- 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 reception
 0 = Selects 8-bit reception
- bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
 Don't care.
Synchronous mode – Master:
 1 = Enables single receive
 0 = Disables single receive
 This bit is cleared after reception is complete.
Synchronous mode – Slave:
 Don't care.
- bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
 1 = Enables 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 receive next valid byte)
 0 = No framing error
- bit 1 **OERR:** Overrun Error bit
 1 = Overrun error (can be cleared by clearing bit CREN)
 0 = No overrun error
- bit 0 **RX9D:** 9th bit of Received Data
 This can be address/data bit or a parity bit and must be calculated by user firmware.

Legend:

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

PIC18F2455/2550/4455/4550

REGISTER 20-3: BAUDCON: BAUD RATE CONTROL REGISTER

| R/W-0 | R-1 | U-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | |
|--------|-------|-----|-------|-------|-----|-------|-------|-------|
| ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | |
| bit 7 | | | | | | | | bit 0 |

- 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 Idle
 0 = Receive operation is active
- bit 5 **Unimplemented:** Read as '0'
- bit 4 **SCKP:** Synchronous Clock Polarity Select bit
Asynchronous mode:
 Unused in this mode.
Synchronous mode:
 1 = Idle state for clock (CK) is a high level
 0 = Idle state for clock (CK) is a low level
- 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.

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 |

PIC18F2455/2550/4455/4550

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 advanta-

geous 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

| Configuration Bits | | | BRG/EUSART Mode | Baud Rate Formula |
|--------------------|-------|------|---------------------|------------------------|
| SYNC | BRG16 | BRGH | | |
| 0 | 0 | 0 | 8-bit/Asynchronous | $F_{OSC}/[64 (n + 1)]$ |
| 0 | 0 | 1 | 8-bit/Asynchronous | $F_{OSC}/[16 (n + 1)]$ |
| 0 | 1 | 0 | 16-bit/Asynchronous | |
| 0 | 1 | 1 | 16-bit/Asynchronous | $F_{OSC}/[4 (n + 1)]$ |
| 1 | 0 | x | 8-bit/Synchronous | |
| 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 | = $F_{OSC}/(64 ([SPBRGH:SPBRG] + 1))$ |
| Solving for SPBRGH:SPBRG: | |
| X | = $((F_{OSC}/\text{Desired Baud Rate})/64) - 1$ |
| | = $((16000000/9600)/64) - 1$ |
| | = $[25.042] = 25$ |
| Calculated Baud Rate | = $16000000/(64 (25 + 1))$ |
| | = 9615 |
| Error | = $(\text{Calculated Baud Rate} - \text{Desired Baud Rate})/\text{Desired Baud Rate}$ |
| | = $(9615 - 9600)/9600 = 0.16\%$ |

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 | — | SCKP | BRG16 | — | WUE | ABDEN | 53 |
| SPBRGH | EUSART Baud Rate Generator Register High Byte | | | | | | | | 53 |
| SPBRG | EUSART Baud Rate Generator Register Low Byte | | | | | | | | 53 |

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

PIC18F2455/2550/4455/4550

TABLE 20-3: BAUD RATES FOR ASYNCHRONOUS MODES

| BAUD RATE (K) | SYNC = 0, BRGH = 0, BRG16 = 0 | | | | | | | | | | | |
|---------------|-------------------------------|---------|-----------------------|-------------------|---------|-----------------------|-------------------|---------|-----------------------|------------------|---------|-----------------------|
| | Fosc = 40.000 MHz | | | Fosc = 20.000 MHz | | | Fosc = 10.000 MHz | | | Fosc = 8.000 MHz | | |
| | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) |
| 0.3 | — | — | — | — | — | — | — | — | — | — | — | — |
| 1.2 | — | — | — | 1.221 | 1.73 | 255 | 1.202 | 0.16 | 129 | 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 | — | — | — |

| BAUD RATE (K) | SYNC = 0, BRGH = 0, BRG16 = 0 | | | | | | | | |
|---------------|-------------------------------|---------|-----------------------|------------------|---------|-----------------------|------------------|---------|-----------------------|
| | Fosc = 4.000 MHz | | | Fosc = 2.000 MHz | | | Fosc = 1.000 MHz | | |
| | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) |
| 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 | — | — | — | — | — | — |

| BAUD RATE (K) | SYNC = 0, BRGH = 1, BRG16 = 0 | | | | | | | | | | | |
|---------------|-------------------------------|---------|-----------------------|-------------------|---------|-----------------------|-------------------|---------|-----------------------|------------------|---------|-----------------------|
| | Fosc = 40.000 MHz | | | Fosc = 20.000 MHz | | | Fosc = 10.000 MHz | | | Fosc = 8.000 MHz | | |
| | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) |
| 0.3 | — | — | — | — | — | — | — | — | — | — | — | — |
| 1.2 | — | — | — | — | — | — | — | — | — | — | — | — |
| 2.4 | — | — | — | — | — | — | 2.441 | 1.73 | 255 | 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 | — | — | — |

| BAUD RATE (K) | SYNC = 0, BRGH = 1, BRG16 = 0 | | | | | | | | |
|---------------|-------------------------------|---------|-----------------------|------------------|---------|-----------------------|------------------|---------|-----------------------|
| | Fosc = 4.000 MHz | | | Fosc = 2.000 MHz | | | Fosc = 1.000 MHz | | |
| | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) |
| 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 | — | — | — | — | — | — |

PIC18F2455/2550/4455/4550

TABLE 20-3: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

| BAUD RATE (K) | SYNC = 0, BRGH = 0, BRG16 = 1 | | | | | | | | | | | |
|---------------|-------------------------------|---------|-----------------------|-------------------|---------|-----------------------|-------------------|---------|-----------------------|------------------|---------|-----------------------|
| | Fosc = 40.000 MHz | | | Fosc = 20.000 MHz | | | Fosc = 10.000 MHz | | | Fosc = 8.000 MHz | | |
| | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) |
| 0.3 | 0.300 | 0.00 | 8332 | 0.300 | 0.02 | 4165 | 0.300 | 0.02 | 2082 | 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 | | | Fosc = 2.000 MHz | | | Fosc = 1.000 MHz | | |
| | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) |
| 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 |

| BAUD RATE (K) | SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1 | | | | | | | | |
|---------------|--|---------|-----------------------|------------------|---------|-----------------------|------------------|---------|-----------------------|
| | Fosc = 4.000 MHz | | | Fosc = 2.000 MHz | | | Fosc = 1.000 MHz | | |
| | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) | Actual Rate (K) | % Error | SPBRG value (decimal) |
| 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 | — | — | — | — | — | — |

PIC18F2455/2550/4455/4550

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 Detection 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 totaling 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 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

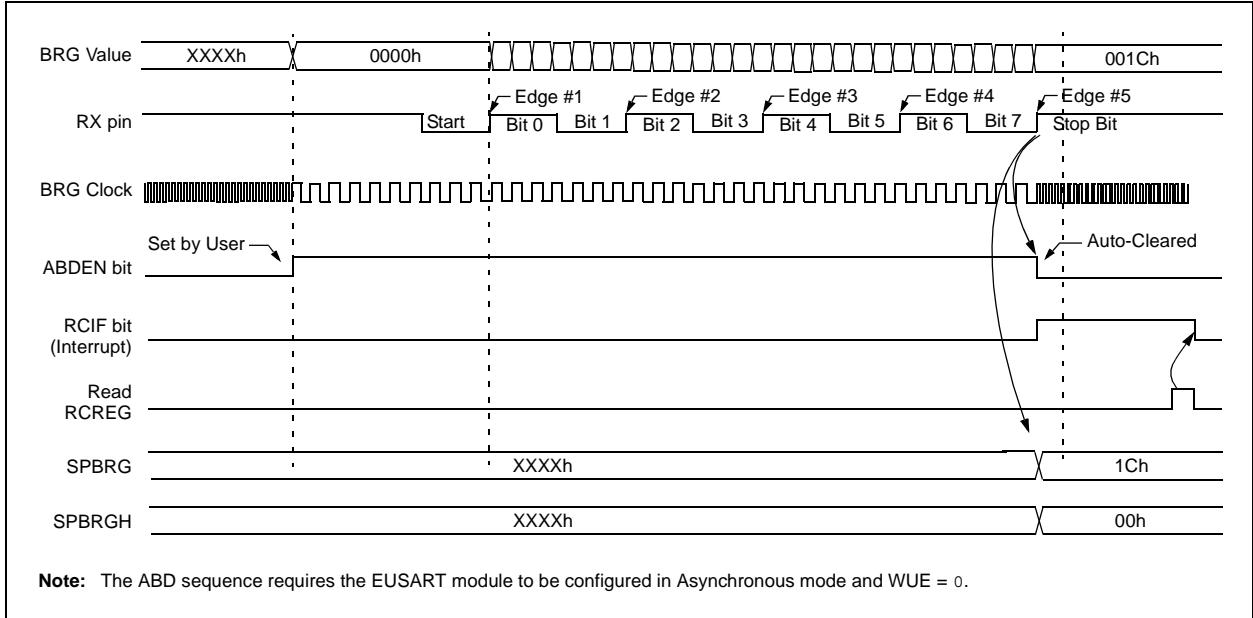
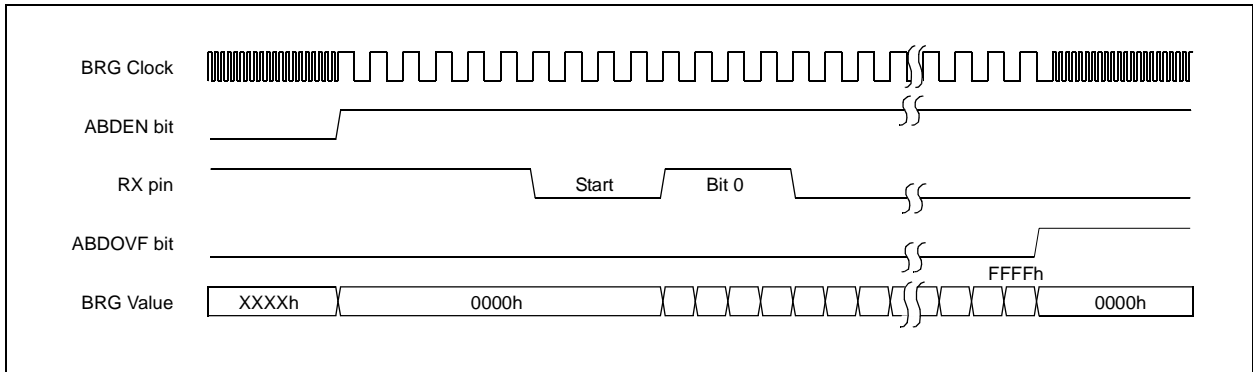


FIGURE 20-2: BRG OVERFLOW SEQUENCE



PIC18F2455/2550/4455/4550

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.

When operating in Asynchronous mode, the EUSART module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver
- Auto-Wake-up on Sync Break Character
- 12-bit Break Character Transmit
- Auto-Baud Rate Detection

20.2.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 20-3. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The Shift register obtains its data from the Read/Write Transmit Buffer register, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXREG register (if available).

Once the TXREG register transfers the data to the TSR register (occurs in one T_{cy}), 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.

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:

1. 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 bit SYNC and setting bit SPEN.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set transmit bit TX9. Can be used as address/data bit.
5. Enable the transmission by setting bit TXEN which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

FIGURE 20-3: EUSART TRANSMIT BLOCK DIAGRAM

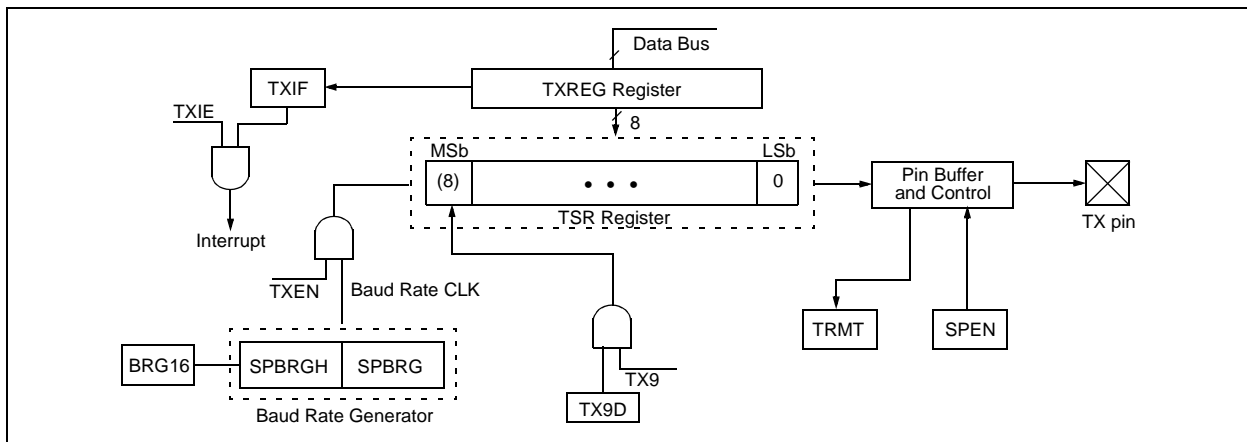


FIGURE 20-4: ASYNCHRONOUS TRANSMISSION

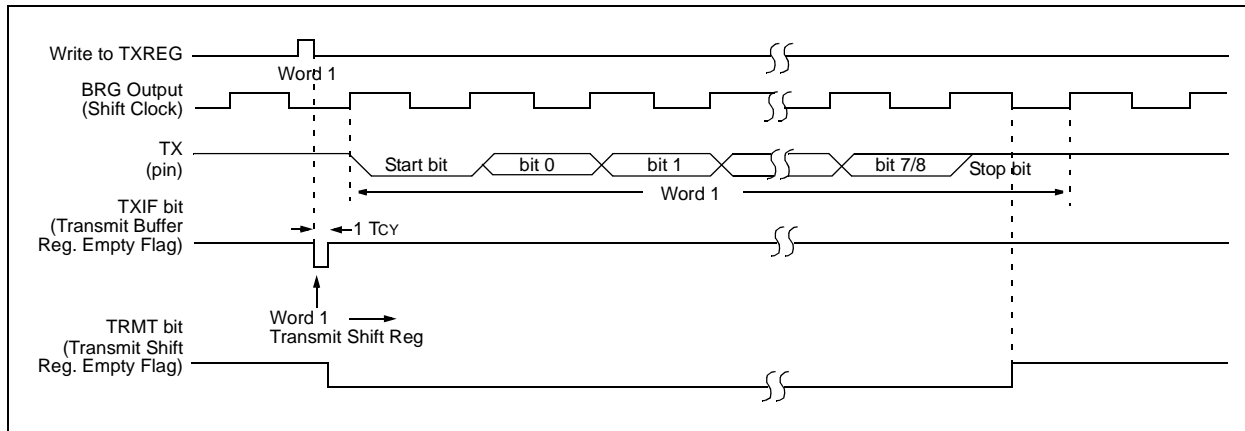


FIGURE 20-5: ASYNCHRONOUS TRANSMISSION (BACK TO BACK)

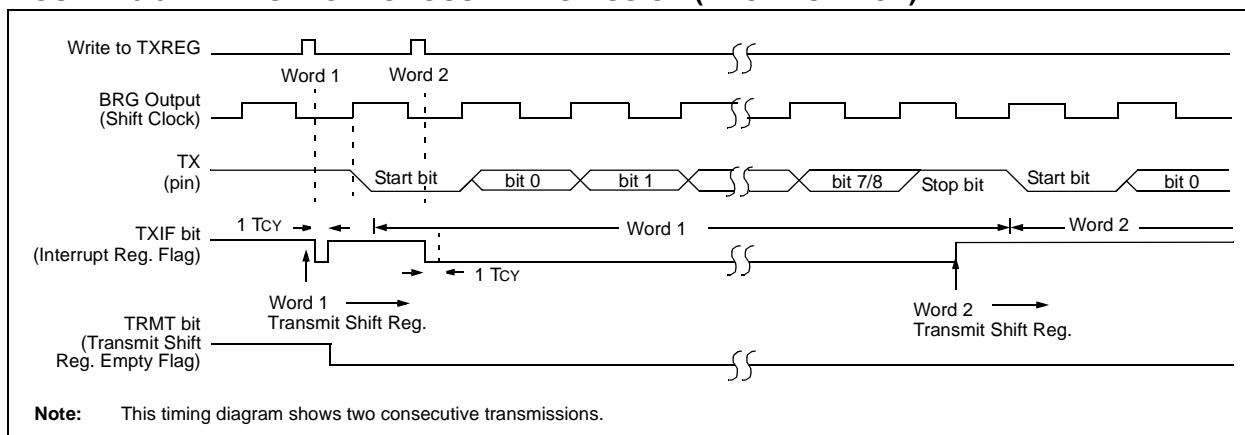


TABLE 20-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on page |
|---------|---|-----------|--------|--------|-------|--------|--------|--------|----------------------|
| INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 51 |
| PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 54 |
| PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 54 |
| IPR1 | PSPIP ⁽¹⁾ | ADIP | RCIP | TXIP | SSPIP | CCP1IP | TMR2IP | TMR1IP | 54 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 53 |
| TXREG | EUSART Transmit Register | | | | | | | | 53 |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 53 |
| BAUDCON | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 53 |
| SPBRGH | EUSART Baud Rate Generator Register High Byte | | | | | | | | 53 |
| SPBRG | EUSART Baud Rate Generator Register Low 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.

PIC18F2455/2550/4455/4550

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.

To set up an Asynchronous Reception:

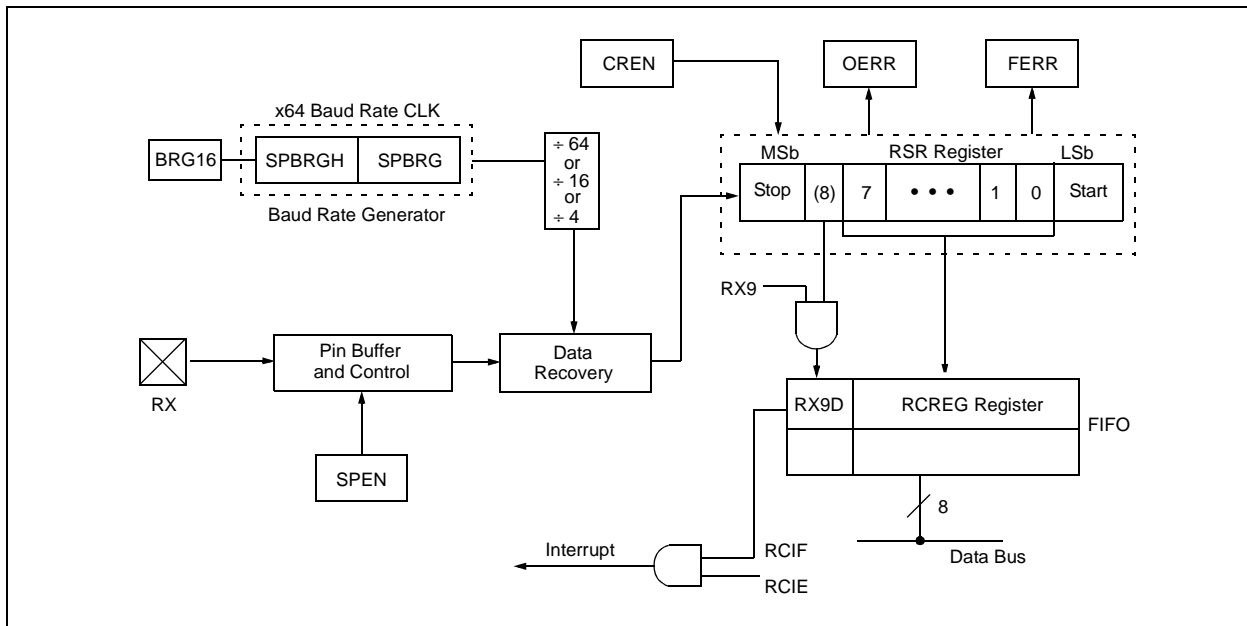
1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, set enable bit RCIE.
4. If 9-bit reception is desired, set bit RX9.
5. Enable the reception by setting bit CREN.
6. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
7. 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.
9. If any error occurred, clear the error by clearing enable bit CREN.
10. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

20.2.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

1. Initialize the 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.
3. If interrupts are required, set the RCEN bit and select the desired priority level with the RCIP bit.
4. Set the RX9 bit to enable 9-bit reception.
5. Set the ADDEN bit to enable address detect.
6. Enable reception by setting the CREN bit.
7. The RCIF bit will be set when reception is complete. The interrupt will be Acknowledged if the RCIE and GIE bits are set.
8. Read the RCSTA register to determine if any error occurred during reception, as well as read bit 9 of data (if applicable).
9. Read RCREG to determine if the device is being addressed.
10. If any error occurred, clear the CREN bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and interrupt the CPU.

FIGURE 20-6: EUSART RECEIVE BLOCK DIAGRAM



PIC18F2455/2550/4455/4550

FIGURE 20-7: ASYNCHRONOUS RECEPTION

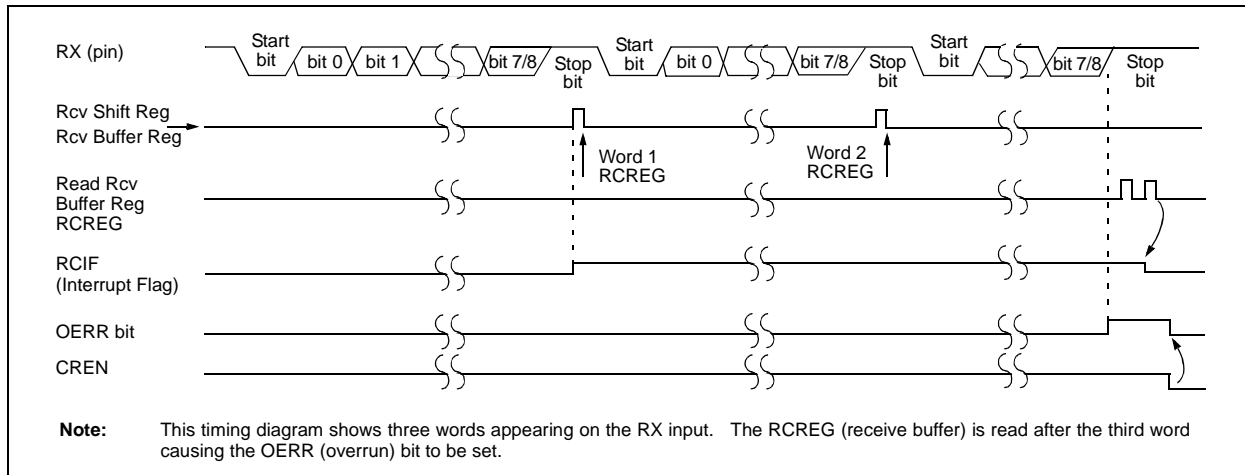


TABLE 20-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on page |
|---------|---|-----------|--------|--------|-------|--------|--------|--------|----------------------|
| INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 51 |
| PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 54 |
| PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 54 |
| IPR1 | PSPIP ⁽¹⁾ | ADIP | RCIP | TXIP | SSPIP | CCP1IP | TMR2IP | TMR1IP | 54 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 53 |
| RCREG | EUSART Receive Register | | | | | | | | 53 |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 53 |
| BAUDCON | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 53 |
| SPBRGH | EUSART Baud Rate Generator Register High Byte | | | | | | | | 53 |
| SPBRG | EUSART Baud Rate Generator Register Low Byte | | | | | | | | 53 |

Legend: — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

Note 1: Reserved in 28-pin devices; always maintain these bits clear.

PIC18F2455/2550/4455/4550

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-to-high 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

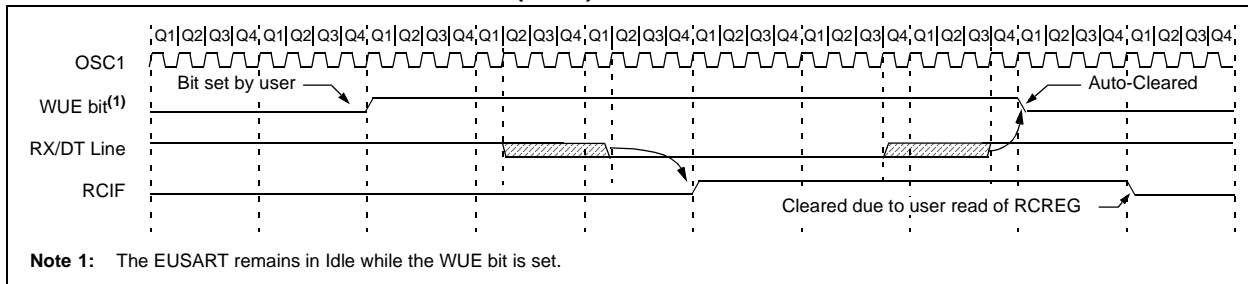
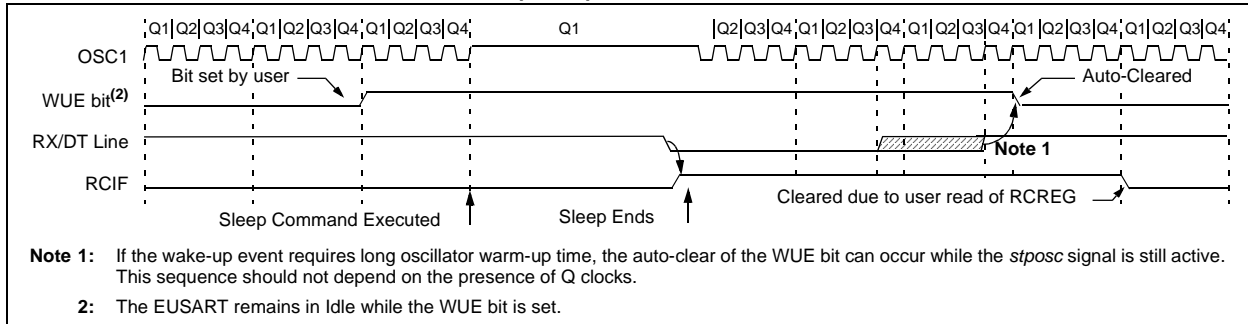


FIGURE 20-9: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP



20.2.5 BREAK CHARACTER SEQUENCE

The Enhanced 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.
2. 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).
4. Write '55h' to TXREG to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware. The Sync character now transmits in the preconfigured mode.

When the TXREG becomes empty, as indicated by the TXIF, the next data byte can be written to TXREG.

20.2.6 RECEIVING A BREAK CHARACTER

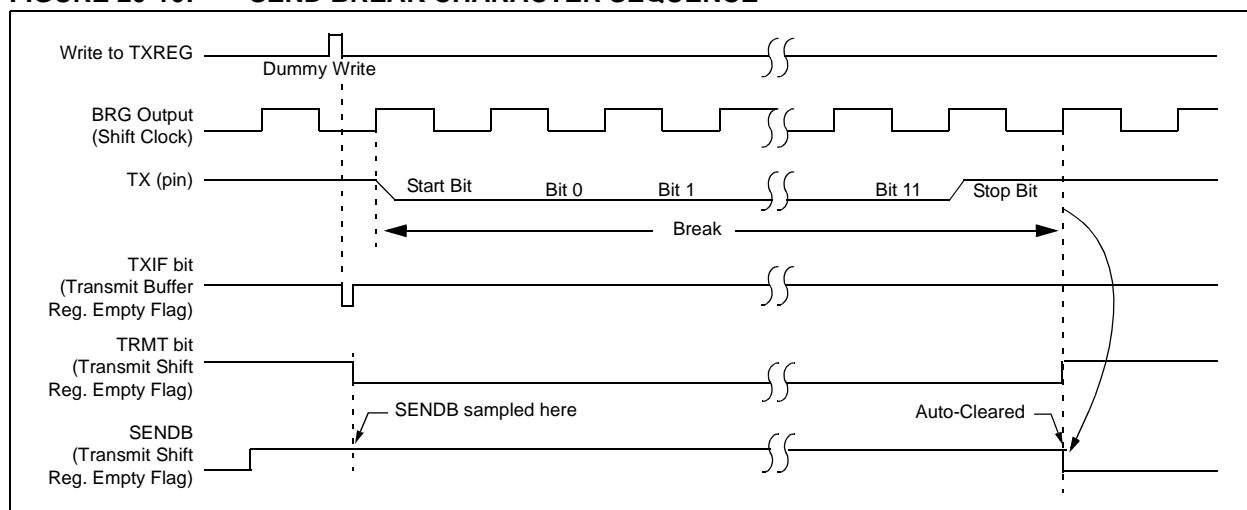
The Enhanced USART module can receive a Break character in two ways.

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.

FIGURE 20-10: SEND BREAK CHARACTER SEQUENCE



PIC18F2455/2550/4455/4550

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 is selected with the SCKP bit (BAUDCON<4>); setting SCKP sets the Idle state on CK as high, while clearing the bit sets the Idle state as low. This option is provided to support Microwire devices with this module.

20.3.1 EUSART SYNCHRONOUS MASTER TRANSMISSION

The EUSART transmitter block diagram is shown in Figure 20-3. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The Shift register obtains its data from the Read/Write Transmit Buffer register, 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).

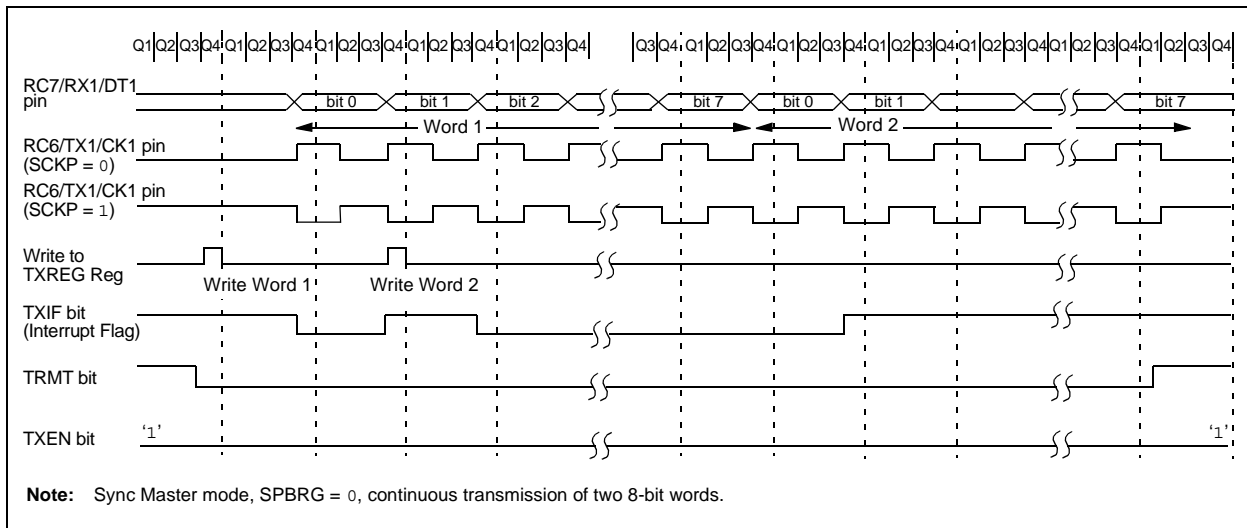
Once the TXREG register transfers the data to the TSR register (occurs in one T_{CYCLE}), the TXREG is empty and the TXIF flag bit (PIR1<4>) is set. The interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TXIE (PIE1<4>). TXIF is set regardless of the state of enable 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:

1. 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.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set bit TX9.
5. Enable the transmission by setting bit TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG register.
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

FIGURE 20-11: SYNCHRONOUS TRANSMISSION



PIC18F2455/2550/4455/4550

FIGURE 20-12: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)

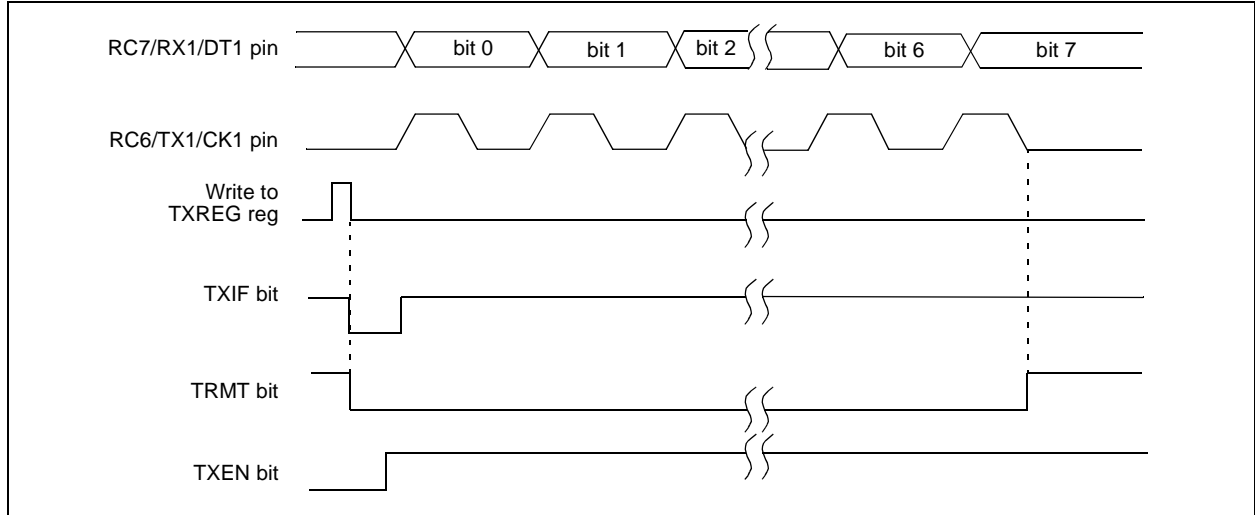


TABLE 20-7: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

| 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 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 54 |
| PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 54 |
| IPR1 | PSPIP ⁽¹⁾ | ADIP | RCIP | TXIP | SSPIP | CCP1IP | TMR2IP | TMR1IP | 54 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 53 |
| TXREG | EUSART Transmit Register | | | | | | | | 53 |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 53 |
| BAUDCON | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 53 |
| SPBRGH | EUSART Baud Rate Generator Register High Byte | | | | | | | | 53 |
| SPBRG | EUSART Baud Rate Generator Register 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.

PIC18F2455/2550/4455/4550

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:

1. 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.
3. Ensure bits CREN and SREN are clear.
4. If interrupts are desired, set enable bit RCIE.
5. If 9-bit reception is desired, set bit RX9.
6. If a single reception is required, set bit SREN. For continuous reception, set bit CREN.
7. Interrupt flag bit RCIF will be set when reception is complete and an interrupt will be generated if the 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.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing bit CREN.
11. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

FIGURE 20-13: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)

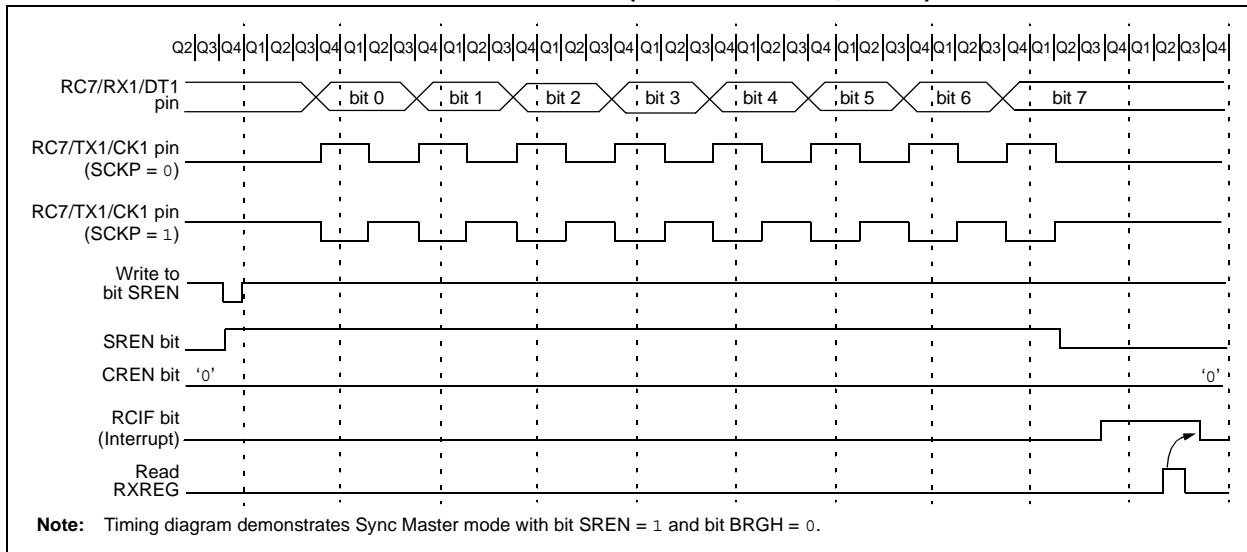


TABLE 20-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

| 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 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 54 |
| PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 54 |
| IPR1 | PSPIP ⁽¹⁾ | ADIP | RCIP | TXIP | SSPIP | CCP1IP | TMR2IP | TMR1IP | 54 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 53 |
| RCREG | EUSART Receive Register | | | | | | | | 53 |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 53 |
| BAUDCON | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 53 |
| SPBRGH | EUSART Baud Rate Generator Register High Byte | | | | | | | | 53 |
| SPBRG | EUSART Baud Rate Generator Register Low Byte | | | | | | | | 53 |

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

Note 1: Reserved in 28-pin devices; always maintain these bits clear.

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 low-power mode.

20.4.1 EUSART SYNCHRONOUS SLAVE TRANSMIT

The operation of the Synchronous Master and Slave modes are identical, except in the case of the Sleep mode.

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

- The first word will immediately transfer to the TSR register and transmit.
- The second word will remain in TXREG register.
- Flag bit TXIF will not be set.
- When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and flag bit TXIF will now be set.
- If enable bit TXIE is set, the interrupt will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Transmission:

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

TABLE 20-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on page |
|---------|---|-----------|--------|--------|-------|--------|--------|--------|----------------------|
| INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 51 |
| PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 54 |
| PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 54 |
| IPR1 | PSPIP ⁽¹⁾ | ADIP | RCIP | TXIP | SSPIP | CCP1IP | TMR2IP | TMR1IP | 54 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 53 |
| TXREG | EUSART Transmit Register | | | | | | | | 53 |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 53 |
| BAUDCON | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 53 |
| SPBRGH | EUSART Baud Rate Generator Register High Byte | | | | | | | | 53 |
| SPBRG | EUSART Baud Rate Generator Register Low Byte | | | | | | | | 53 |

Legend: — = unimplemented, read as '0'. Shaded cells are not used for synchronous slave transmission.

Note 1: Reserved in 28-pin devices; always maintain these bits clear.

PIC18F2455/2550/4455/4550

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:

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

TABLE 20-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on page |
|---------|---|-----------|--------|--------|--------|--------|--------|--------|----------------------|
| INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 51 |
| PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 54 |
| PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 54 |
| IPR1 | PSPIP ⁽¹⁾ | ADIP | RCIP | TXIP | SSPIP | CCP1IP | TMR2IP | TMR1IP | 54 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 53 |
| RCREG | EUSART Receive Register | | | | | | | | 53 |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDER | BRGH | TRMT | TX9D | 53 |
| BAUDCON | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 53 |
| SPBRGH | EUSART Baud Rate Generator Register High Byte | | | | | | | | 53 |
| SPBRG | EUSART Baud Rate Generator Register Low Byte | | | | | | | | 53 |

Legend: — = unimplemented, read as '0'. Shaded cells are not used for synchronous slave reception.

Note 1: Reserved in 28-pin devices; always maintain these bits clear.

PIC18F2455/2550/4455/4550

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, shown in Register 21-3, configures the A/D clock source, programmed acquisition time and justification.

REGISTER 21-1: ADCON0 REGISTER

| | | | | | | | |
|-------|-----|-------|-------|-------|-------|---------|-------|
| 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 | |

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⁽²⁾
 1110 = Unimplemented⁽²⁾
 1111 = Unimplemented⁽²⁾

Note 1: These channels are not implemented on 28-pin devices.

2: Performing a conversion on unimplemented channels will return a floating input measurement.

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

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 |

PIC18F2455/2550/4455/4550

REGISTER 21-2: ADCON1 REGISTER

| | | | | | | | |
|-------|-----|-------|-------|----------------------|--------------------|--------------------|--------------------|
| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 ⁽¹⁾ | R/W ⁽¹⁾ | R/W ⁽¹⁾ | R/W ⁽¹⁾ |
| — | — | VCFG1 | VCFG0 | PCFG3 | PCFG2 | PCFG1 | PCFG0 |
| bit 7 | | | | bit 0 | | | |

- bit 7-6 **Unimplemented:** Read as '0'
- bit 5 **VCFG1:** Voltage Reference Configuration bit (VREF- source)
1 = VREF- (AN2)
0 = AVSS
- bit 4 **VCFG0:** Voltage Reference Configuration bit (VREF+ source)
1 = VREF+ (AN3)
0 = AVDD
- bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits:

| PCFG3: PCFG0 | AN12 | AN11 | AN10 | AN9 | AN8 | AN7 ⁽²⁾ | AN6 ⁽²⁾ | AN5 ⁽²⁾ | AN4 | AN3 | AN2 | AN1 | AN0 |
|---------------------|------|------|------|-----|-----|--------------------|--------------------|--------------------|-----|-----|-----|-----|-----|
| 0000 ⁽¹⁾ | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0001 | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0010 | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0011 | D | A | A | A | A | A | A | A | A | A | A | A | A |
| 0100 | D | D | A | A | A | A | A | A | A | A | A | A | A |
| 0101 | D | D | D | A | A | A | A | A | A | A | A | A | A |
| 0110 | D | D | D | D | A | A | A | A | A | A | A | A | A |
| 0111 ⁽¹⁾ | D | D | D | D | D | A | A | A | A | A | A | A | A |
| 1000 | D | D | D | D | D | D | A | A | A | A | A | A | A |
| 1001 | D | D | D | D | D | D | D | A | A | A | A | A | A |
| 1010 | D | D | D | D | D | D | D | D | A | A | A | A | A |
| 1011 | D | D | D | D | D | D | D | D | D | A | A | A | A |
| 1100 | D | D | D | D | D | D | D | D | D | D | A | A | A |
| 1101 | D | D | D | D | D | D | D | D | D | D | D | A | A |
| 1110 | D | D | D | D | D | D | D | D | D | D | D | D | A |
| 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.

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

PIC18F2455/2550/4455/4550

REGISTER 21-3: ADCON2 REGISTER

| 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 |

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 TAD

010 = 4 TAD

001 = 2 TAD

000 = 0 TAD⁽¹⁾

bit 2-0 **ADCS2:ADCS0:** A/D Conversion Clock Select bits

111 = FRC (clock derived from A/D RC oscillator)⁽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 T_{cy} (instruction cycle) is added before the A/D clock starts. This allows the SLEEP instruction to be executed before starting a conversion.

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

PIC18F2455/2550/4455/4550

The analog reference voltage is software selectable to either the device's positive and negative supply voltage (AVDD and AVSS), or the voltage level on the RA3/AN3/VREF+ and RA2/AN2/VREF-/CVREF pins.

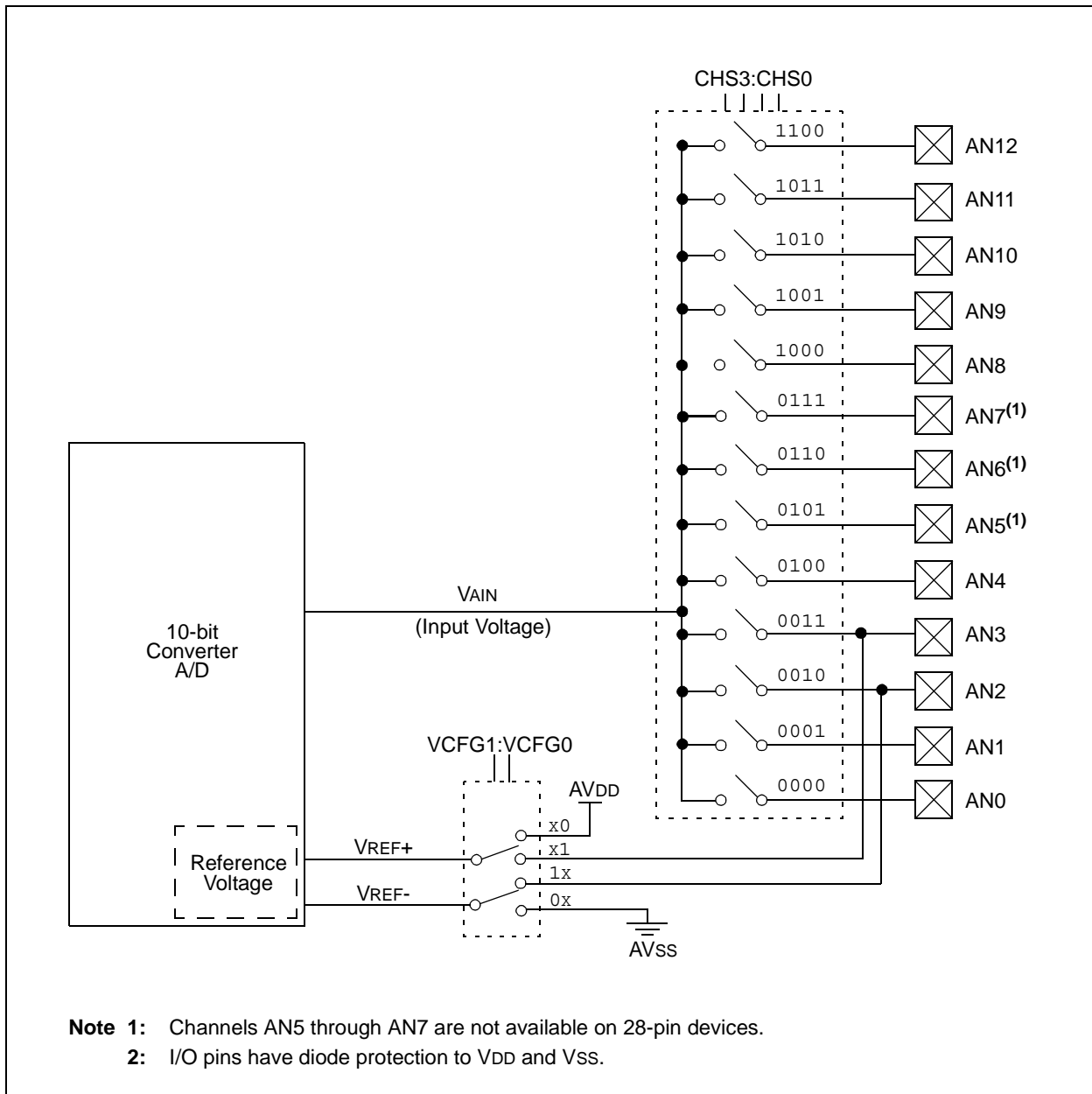
The A/D converter has a unique feature of being able to operate while the device is in Sleep mode. To operate in Sleep, the A/D conversion clock must be derived from the A/D's internal RC oscillator.

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



PIC18F2455/2550/4455/4550

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
6. Read A/D Result registers (ADRESH:ADRESL); clear bit ADIF, if required.
7. For next conversion, go to step 1 or step 2, as required. The A/D conversion time per bit is defined as T_{AD} . A minimum wait of $3 T_{AD}$ is required before the next acquisition starts.

FIGURE 21-2: A/D TRANSFER FUNCTION

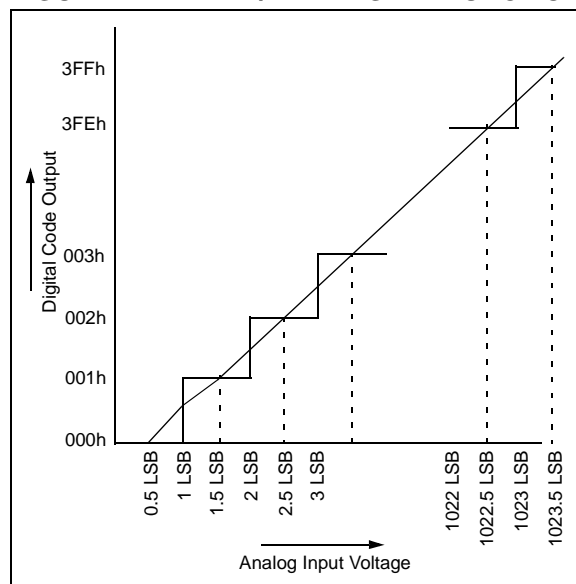
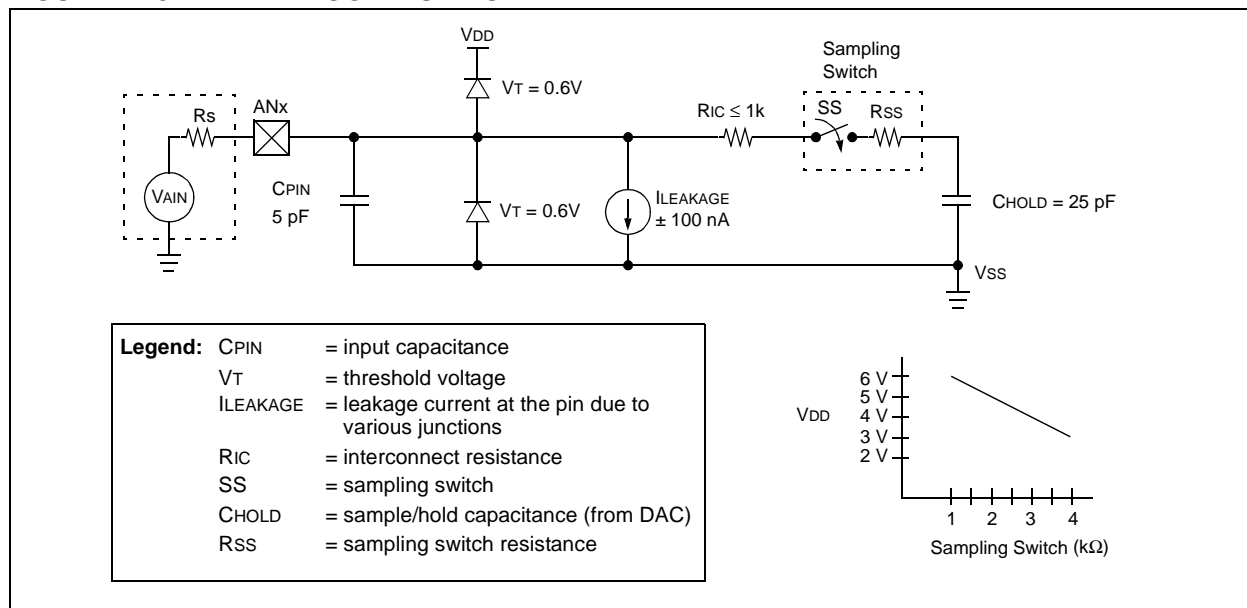


FIGURE 21-3: ANALOG INPUT MODEL



PIC18F2455/2550/4455/4550

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 kΩ.** After the analog input channel is selected (changed), the channel must be sampled for at least the minimum acquisition time before starting a conversion.

Note: When the conversion is started, the holding capacitor is disconnected from the input pin.

To calculate the minimum acquisition time, Equation 21-1 may be used. This equation assumes that 1/2 LSB error is used (1024 steps for the A/D). The 1/2 LSB error is the maximum error allowed for the A/D to meet its specified resolution.

Example 21-3 shows the calculation of the minimum required acquisition time TACQ. This calculation is based on the following application system assumptions:

| | | |
|------------------|---|--------------------|
| CHOLD | = | 25 pF |
| Rs | = | 2.5 kΩ |
| Conversion Error | ≤ | 1/2 LSB |
| VDD | = | 5V → Rss = 2 kΩ |
| Temperature | = | 85°C (system max.) |

EQUATION 21-1: ACQUISITION TIME

$$\begin{aligned} TACQ &= \text{Amplifier Settling Time} + \text{Holding Capacitor Charging Time} + \text{Temperature Coefficient} \\ &= TAMP + TC + TCOFF \end{aligned}$$

EQUATION 21-2: A/D MINIMUM CHARGING TIME

$$\begin{aligned} V_{HOLD} &= (V_{REF} - (V_{REF}/2048)) \cdot (1 - e^{-(TC/CHOLD)(RIC + R_{SS} + R_S)}) \\ \text{or} \\ TC &= -(CHOLD)(RIC + R_{SS} + R_S) \ln(1/2048) \end{aligned}$$

EQUATION 21-3: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME

$$\begin{aligned} TACQ &= TAMP + TC + TCOFF \\ TAMP &= 0.2 \mu\text{s} \\ TCOFF &= \begin{aligned} &(\text{Temp} - 25^\circ\text{C})(0.02 \mu\text{s}/^\circ\text{C}) \\ &(50^\circ\text{C} - 25^\circ\text{C})(0.02 \mu\text{s}/^\circ\text{C}) \\ &1.2 \mu\text{s} \end{aligned} \\ \text{Temperature coefficient is only required for temperatures } &> 25^\circ\text{C}. \text{ Below } 25^\circ\text{C}, TCOFF = 0 \text{ ms.} \\ TC &= \begin{aligned} &-(CHOLD)(RIC + R_{SS} + R_S) \ln(1/2047) \mu\text{s} \\ &-(25 \text{ pF})(1 \text{ k}\Omega + 2 \text{ k}\Omega + 2.5 \text{ k}\Omega) \ln(0.0004883) \mu\text{s} \\ &5.03 \mu\text{s} \end{aligned} \\ TACQ &= \begin{aligned} &0.2 \mu\text{s} + 5 \mu\text{s} + 1.2 \mu\text{s} \\ &6.4 \mu\text{s} \end{aligned} \end{aligned}$$

21.2 Selecting and Configuring Acquisition Time

The ADCON2 register allows the user to select an acquisition time that occurs each time the $\overline{\text{GO/DONE}}$ 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 provides a range of 2 to 20 TAD. When the $\overline{\text{GO/DONE}}$ 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 $\overline{\text{GO/DONE}}$ bit.

Manual acquisition is selected when the ACQT2:ACQT0 = 000. When the $\overline{\text{GO/DONE}}$ bit is set, sampling is stopped and a conversion begins. The user is responsible for ensuring the required acquisition time has passed between selecting the desired input channel and setting the $\overline{\text{GO/DONE}}$ 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 $\overline{\text{GO/DONE}}$ 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 for more information).

Table 21-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

TABLE 21-1: TAD vs. DEVICE OPERATING FREQUENCIES

| AD Clock Source (TAD) | | Maximum Device Frequency | |
|-----------------------|-------------|--------------------------|----------------------------|
| Operation | ADCS2:ADCS0 | PIC18FXXXX | PIC18LFXXXX ⁽⁴⁾ |
| 2 TOSC | 000 | 2.86 MHz | 1.43 kHz |
| 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 | 40.0 MHz | 22.86 MHz |
| 64 TOSC | 110 | 40.0 MHz | 22.86 MHz |
| RC ⁽³⁾ | x11 | 1.00 MHz ⁽¹⁾ | 1.00 MHz ⁽²⁾ |

- 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.

PIC18F2455/2550/4455/4550

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.

- | |
|--|
| <p>Note 1: When reading the Port register, all pins configured as analog input channels will read as cleared (a low level). Pins configured as a digital input will convert as an analog input. Analog levels on a digitally configured input will be accurately converted.</p> <p>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.</p> <p>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.</p> |
|--|

21.6 A/D Conversions

Figure 21-4 shows the operation of the A/D converter after the GO 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 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 measure values.

FIGURE 21-4: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 000, TACQ = 0)

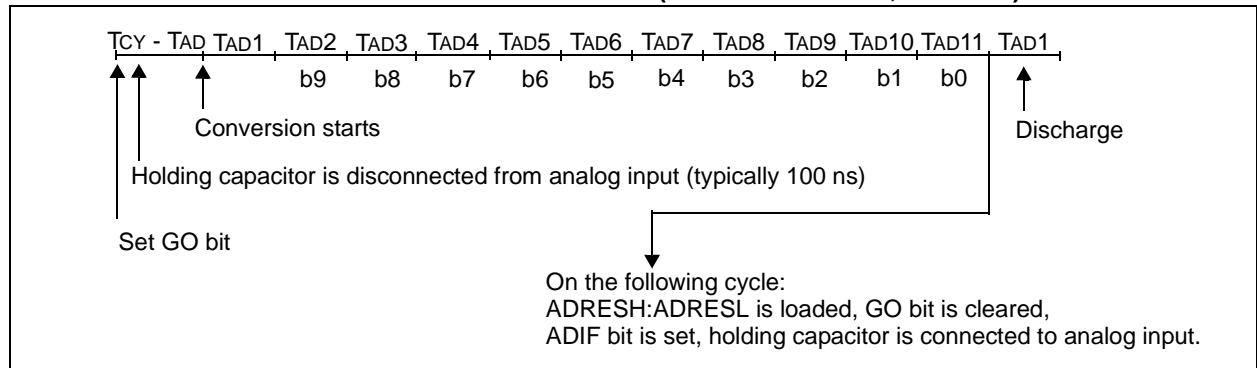
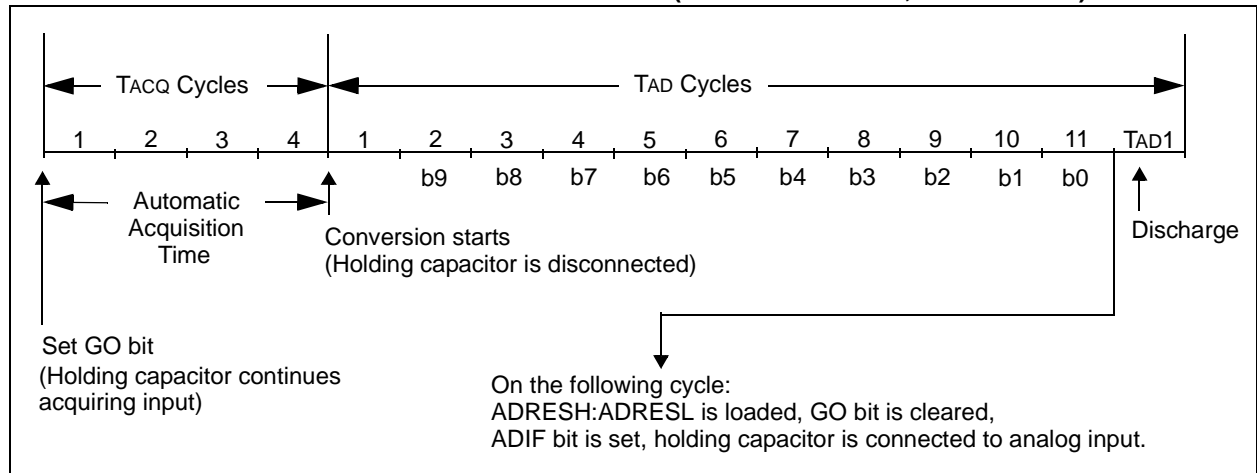


FIGURE 21-5: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 010, TACQ = 4 TAD)



PIC18F2455/2550/4455/4550

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 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 54 |
| PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 54 |
| IPR1 | PSPIP ⁽¹⁾ | ADIP | RCIP | TXIP | SSPIP | CCP1IP | TMR2IP | TMR1IP | 54 |
| PIR2 | OSCFIF | CMIF | — | EEIF | BCLIF | HLVDIF | TMR3IF | CCP2IF | 54 |
| PIE2 | OSCFIE | CMIE | — | EEIE | BCLIE | HLVDIE | TMR3IE | CCP2IE | 54 |
| IPR2 | OSCFIP | CMIP | — | EEIP | BCLIP | HLVDIP | TMR3IP | CCP2IP | 54 |
| ADRESH | A/D Result Register High Byte | | | | | | | | 52 |
| ADRESL | A/D Result Register Low 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 | RA7 ⁽¹⁾ | RA6 ⁽¹⁾ | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | 54 |
| TRISA | TRISA7 ⁽²⁾ | TRISA6 ⁽²⁾ | PORTA Data Direction Control Register | | | | | | 54 |
| PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | 54 |
| TRISB | PORTB Data Direction Control Register | | | | | | | | 54 |
| LATB | PORTB Data Latch Register (Read and Write to Data Latch) | | | | | | | | 54 |
| PORTE ⁽⁴⁾ | — | — | — | — | RE3 ⁽³⁾ | RE2 | RE1 | RE0 | 54 |
| TRISE ⁽⁴⁾ | IBF | OBF | IBOV | PSPMODE | — | TRISE2 | TRISE1 | TRISE0 | 54 |
| LATE ⁽⁴⁾ | — | — | — | — | — | PORTE Data Latch Register | | | 54 |

Legend: — = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.

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

2: PORTA<7:6> and their direction bits are individually configured as port pins based on various primary oscillator modes. When disabled, these bits read as '0'.

3: RE3 port bit is available only as an input pin when the MCLRE configuration bit is '0'.

4: These registers are not implemented on 28-pin devices.

PIC18F2455/2550/4455/4550

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 21.0 “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 REGISTER

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| R-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-1 | R/W-1 | R/W-1 |
| C2OUT | C1OUT | C2INV | C1INV | CIS | CM2 | CM1 | CM0 |
| bit 7 | | | | | | bit 0 | |

bit 7 **C2OUT**: Comparator 2 Output bit

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 **C1OUT**: 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 inverted

0 = C2 output not inverted

bit 4 **C1INV**: Comparator 1 Output Inversion bit

1 = C1 output inverted

0 = C1 output not inverted

bit 3 **CIS**: Comparator Input Switch bit

When CM2:CM0 = 110:

1 = C1 VIN- connects to RA3/AN3

C2 VIN- connects to RA2/AN2

0 = C1 VIN- connects to RA0/AN0

C2 VIN- connects to RA1/AN1

bit 2-0 **CM2:CM0**: Comparator Mode bits

Figure 22-1 shows the Comparator modes and CM2:CM0 bit settings.

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

PIC18F2455/2550/4455/4550

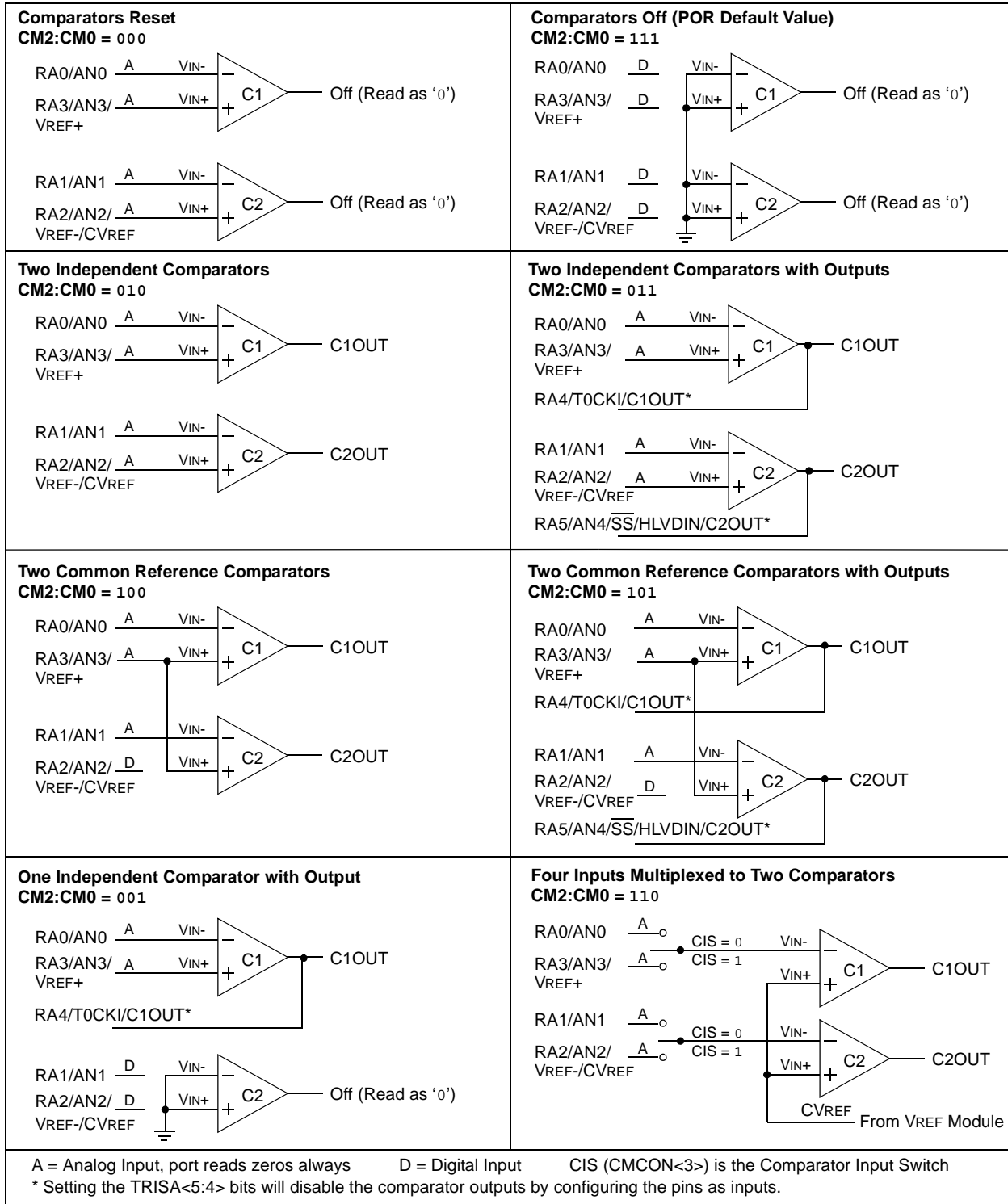
22.1 Comparator Configuration

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



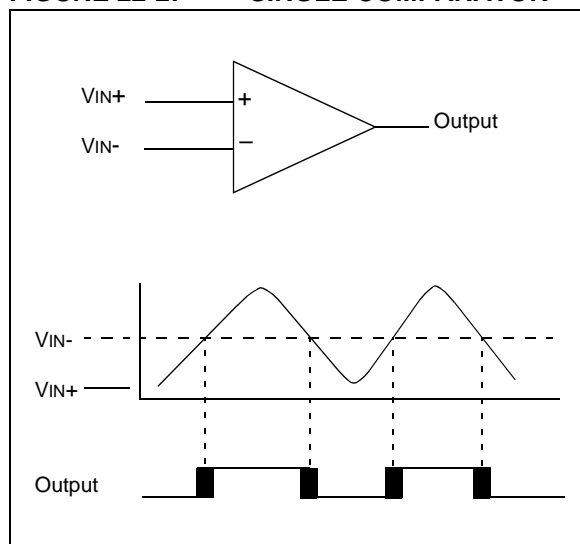
22.2 Comparator Operation

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 V_{IN+} is less than the analog input V_{IN-} , the output of the comparator is a digital low level. When the analog input at V_{IN+} is greater than the analog input V_{IN-} , the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 22-2 represent the uncertainty, due to input offsets and response time.

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 V_{IN-} is compared to the signal at V_{IN+} 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 V_{SS} and V_{DD} 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 21.0 “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 V_{IN+} 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 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.

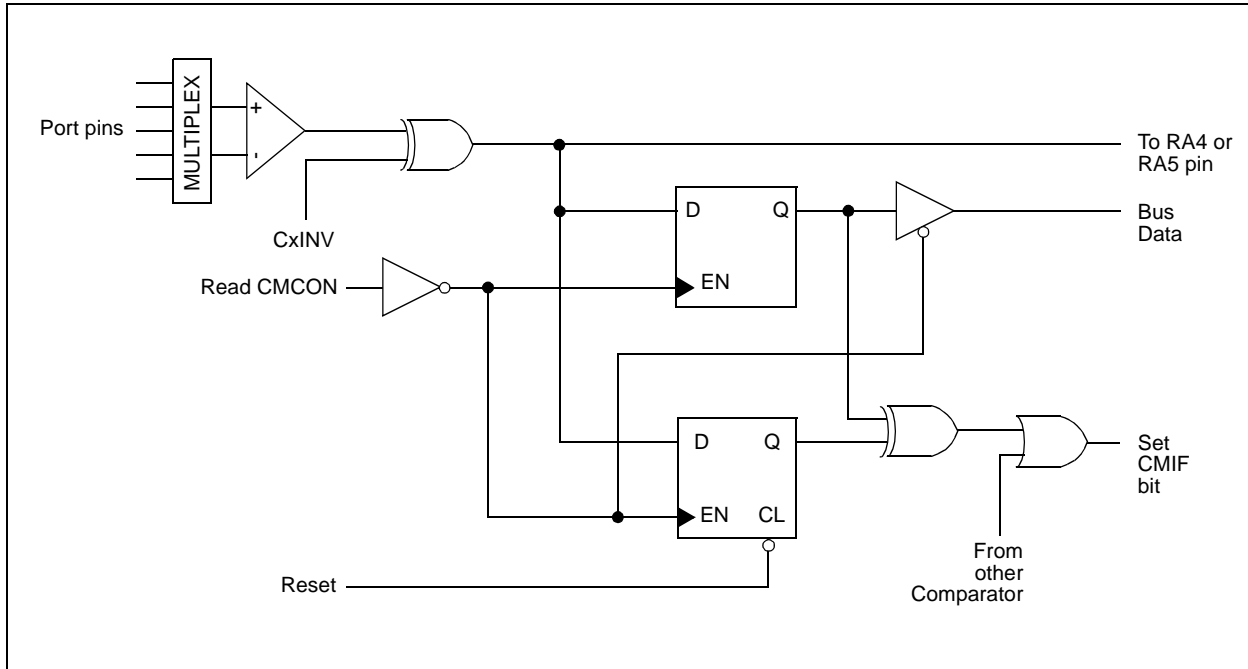
The TRISA bits will still function as an output enable/disable for the RA4 and RA5 pins while in this mode.

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

- Note 1:** When reading the Port register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert an analog input according to the Schmitt Trigger input specification.
- 2:** Analog levels on any pin defined as a digital input may cause the input buffer to consume more current than is specified.

PIC18F2455/2550/4455/4550

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 (PIR registers) interrupt flag may not get set.

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

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

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

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 kΩ 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

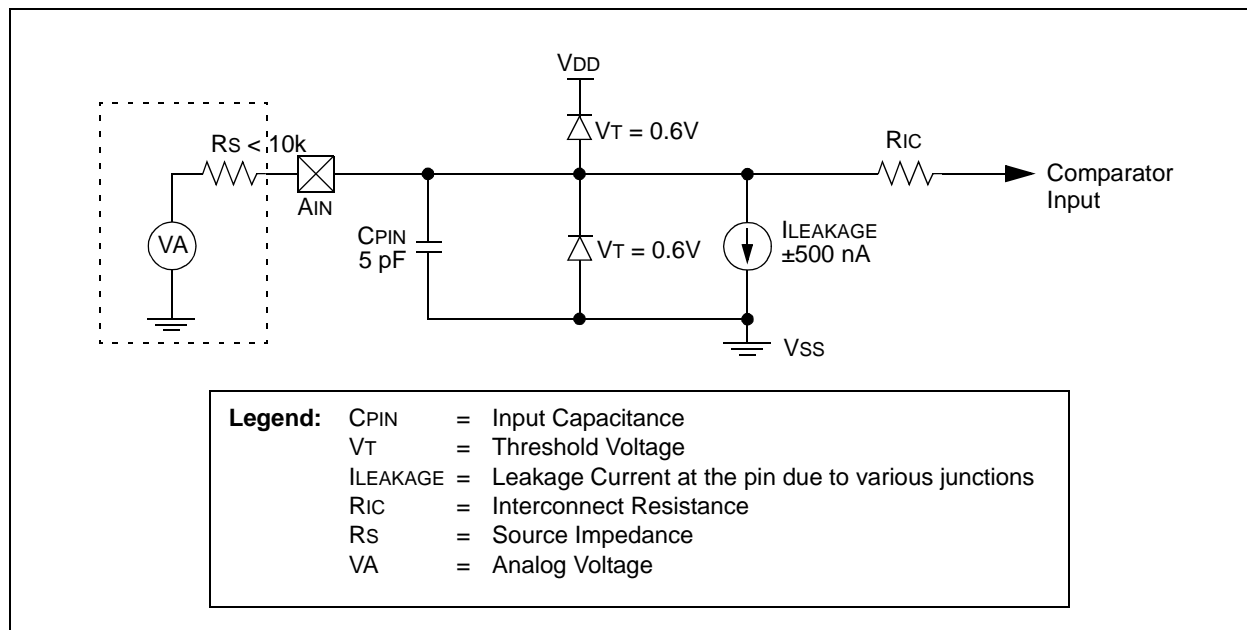


TABLE 22-1: REGISTERS ASSOCIATED WITH COMPARATOR MODULE

| 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 | — | EEIF | BCLIF | LVDIF | TMR3IF | CCP2IF | 54 |
| PIE2 | OCSFIE | CMIE | — | EEIE | BCLIE | LVDIE | TMR3IE | CCP2IE | 54 |
| IPR2 | OSCFIP | CMIP | — | EEIP | BCLIP | LVDIP | TMR3IP | CCP2IP | 54 |
| PORTA | — | RA6 ⁽¹⁾ | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | 54 |
| LATA | — | LATA6 ⁽¹⁾ | PORTA Data Latch Register (Read and Write to Data Latch) | | | | | | 54 |
| TRISA | — | TRISA6 ⁽¹⁾ | PORTA Data Direction Control Register | | | | | | 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'.

PIC18F2455/2550/4455/4550

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:

$$\text{If CVRR} = 1: \\ \text{CVREF} = ((\text{CVR3:CVR0})/24) \times \text{CVRSRC}$$

$$\text{If CVRR} = 0: \\ \text{CVREF} = (\text{CVRSRC} \times 1/4) + (((\text{CVR3:CVR0})/32) \times \text{CVRSRC})$$

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 REGISTER

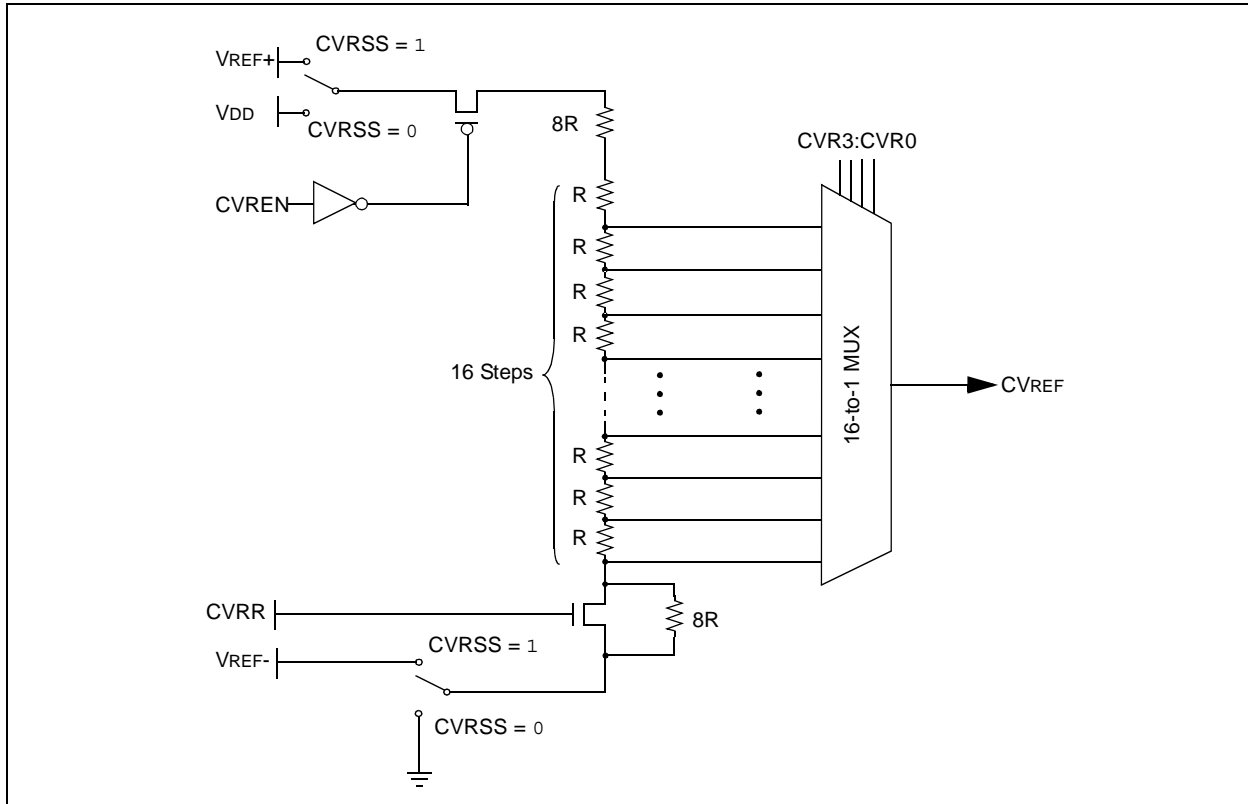
| | | | | | | | |
|-------|----------------------|-------|-------|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| CVREN | CVROE ⁽¹⁾ | CVRR | CVRSS | CVR3 | CVR2 | CVR1 | CVR0 |
| bit 7 | | | | | | | bit 0 |

- bit 7 **CVREN:** Comparator Voltage Reference Enable bit
1 = CVREF circuit powered on
0 = CVREF circuit powered down
- bit 6 **CVROE:** Comparator VREF Output Enable bit⁽¹⁾
1 = CVREF voltage level is also output on the RA2/AN2/VREF-/CVREF pin
0 = CVREF voltage is disconnected from the RA2/AN2/VREF-/CVREF pin
Note 1: CVROE overrides the TRISA<2> bit setting.
- 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 ≤ (CVR3:CVR0) ≤ 15)
When CVRR = 1:
CVREF = ((CVR3:CVR0)/24) • (CVRSRC)
When CVRR = 0:
CVREF = (CVRSRC/4) + ((CVR3:CVR0)/32) • (CVRSRC)

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

PIC18F2455/2550/4455/4550

FIGURE 23-1: 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.

PIC18F2455/2550/4455/4550

FIGURE 23-2: VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE

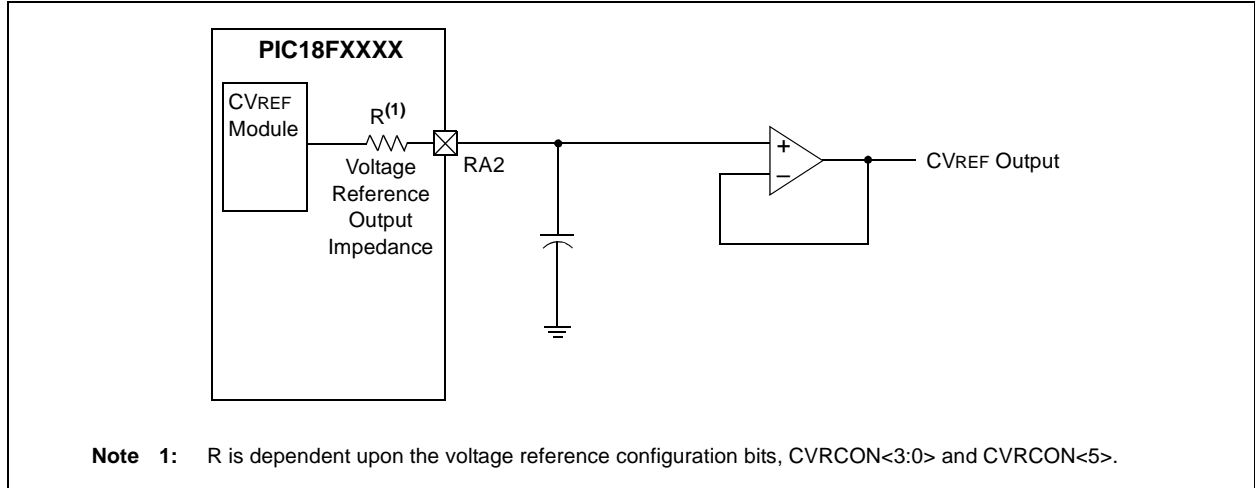


TABLE 23-1: REGISTERS ASSOCIATED WITH COMPARATOR VOLTAGE REFERENCE

| 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 | C1OUT | C2INV | C1INV | CIS | CM2 | CM1 | CM0 | 53 |
| TRISA | TRISA7 ⁽¹⁾ | TRISA6 ⁽¹⁾ | PORTA Data Direction Control Register | | | | | | 54 |

Legend: Shaded cells are not used with the comparator voltage reference.

Note 1: PORTA pins are enabled based on oscillator configuration.

PIC18F2455/2550/4455/4550

NOTES:

PIC18F2455/2550/4455/4550

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.

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

| 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 | HLVDL2 | HLVDL1 | HLVDL0 |
| bit 7 | | | | | | | bit 0 |

- bit 7 **VDIRMAG:** Voltage Direction Magnitude Select bit
 1 = Event occurs when voltage equals or exceeds trip point (HLVDL3:HLVDL0)
 0 = Event occurs when voltage equals or falls below trip point (HLVDL3:HLVDL0)
- bit 6 **Unimplemented:** Read as ‘0’
- bit 5 **IRVST:** Internal Reference Voltage Stable Flag bit
 1 = Indicates that the Voltage Detect logic will generate the interrupt flag at the specified voltage range
 0 = Indicates that the Voltage Detect logic will not generate the interrupt flag at the specified voltage range and the HLVD interrupt should not be enabled
- bit 4 **HLVDEN:** High/Low-Voltage Detect Power Enable bit
 1 = HLVD enabled
 0 = HLVD disabled
- bit 3-0 **HLVDL3:HLVDL0:** Voltage Detection Limit bits
 1111 = External analog input is used (input comes from the HLVDIN pin)
 1110 = 4.41V-4.87V
 1101 = 4.11V-4.55V
 1100 = 3.92V-4.34V
 1011 = 3.72V-4.12V
 1010 = 3.53V-3.91V
 1001 = 3.43V-3.79V
 1000 = 3.24V-3.58V
 0111 = 2.95V-3.26V
 0110 = 2.75V-3.03V
 0101 = 2.64V-2.92V
 0100 = 2.43V-2.69V
 0011 = 2.35V-2.59V
 0010 = 2.16V-2.38V
 0001 = 1.96V-2.16V
 0000 = Reserved

Note: HLVDL3:HLVDL0 modes that result in a trip point below the valid operating voltage of the device are not tested.

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 |

PIC18F2455/2550/4455/4550

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

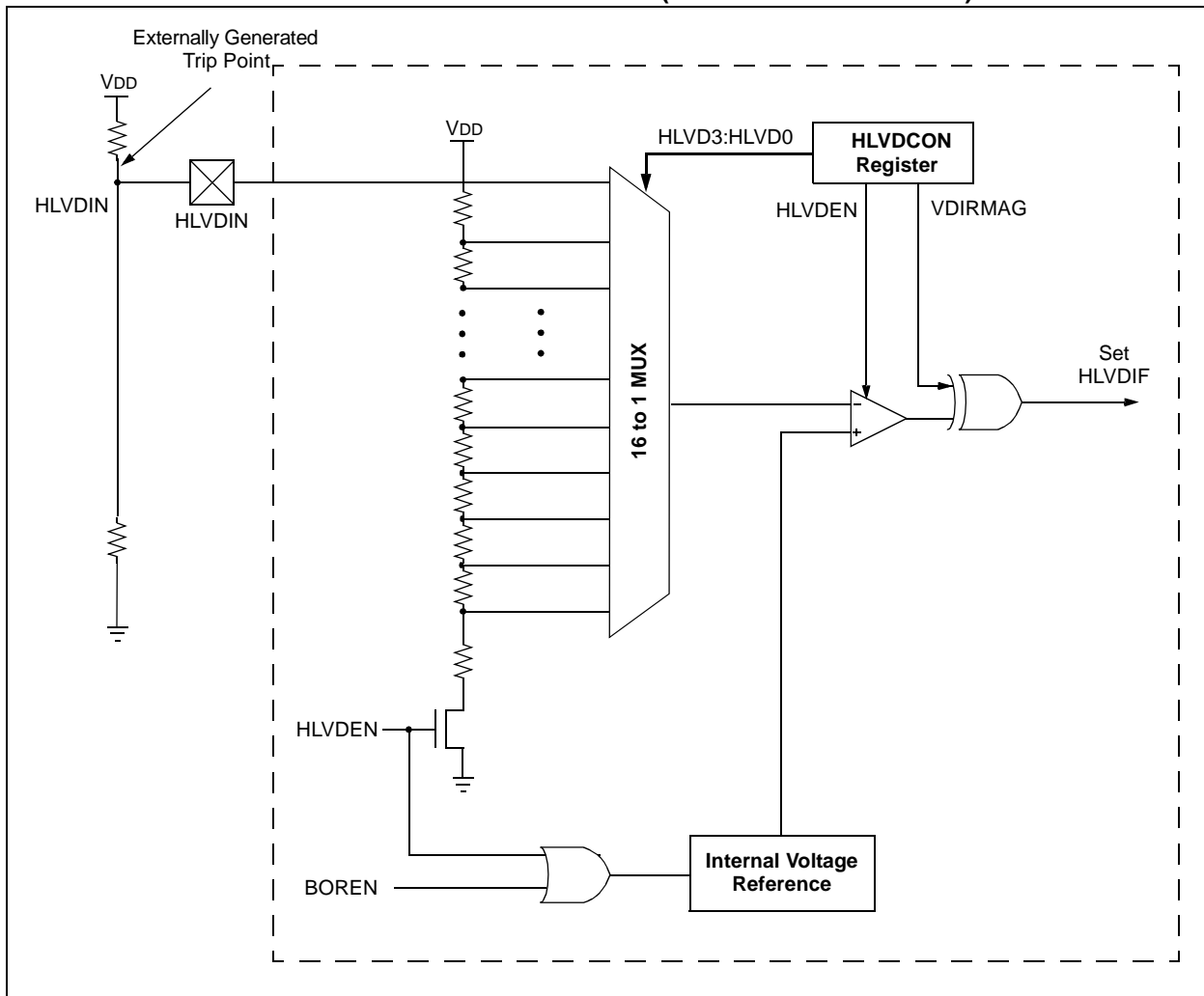
When the HLVD module is enabled, a comparator uses an internally generated reference voltage as the set point. The set point is compared with the trip point, where each node in the resistor divider represents a trip point voltage. The “trip point” voltage is the voltage level at which the device detects 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 sixteen 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:

1. 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.
3. 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.
5. Clear the HLVD interrupt flag (PIR2<2>), which may have been set from a previous interrupt.
6. Enable the HLVD interrupt if interrupts are desired by setting the HLVDIE and GIE bits (PIE<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 #D022B.

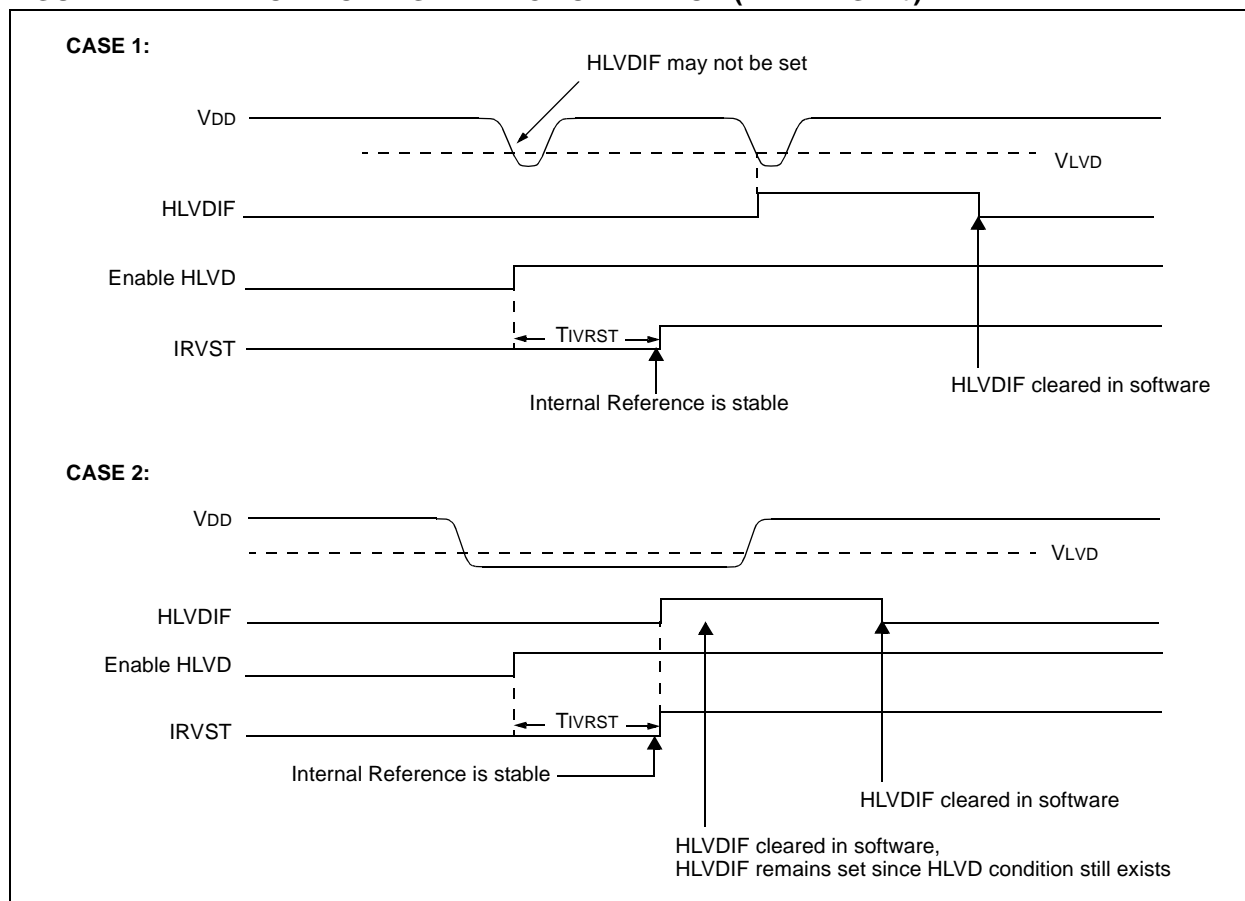
Depending on the application, the HLVD module does not need to be operating constantly. To decrease the current requirements, the HLVD circuitry may only need to be enabled for short periods where the voltage is checked. After doing the check, the HLVD module may be disabled.

24.4 HLVD Start-up Time

The internal reference voltage of the HLVD module, specified in electrical specification parameter #D423, 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, T_{IRVST}, is an interval that is independent of device clock speed. It is specified in electrical specification parameter 36 (Table 28-12).

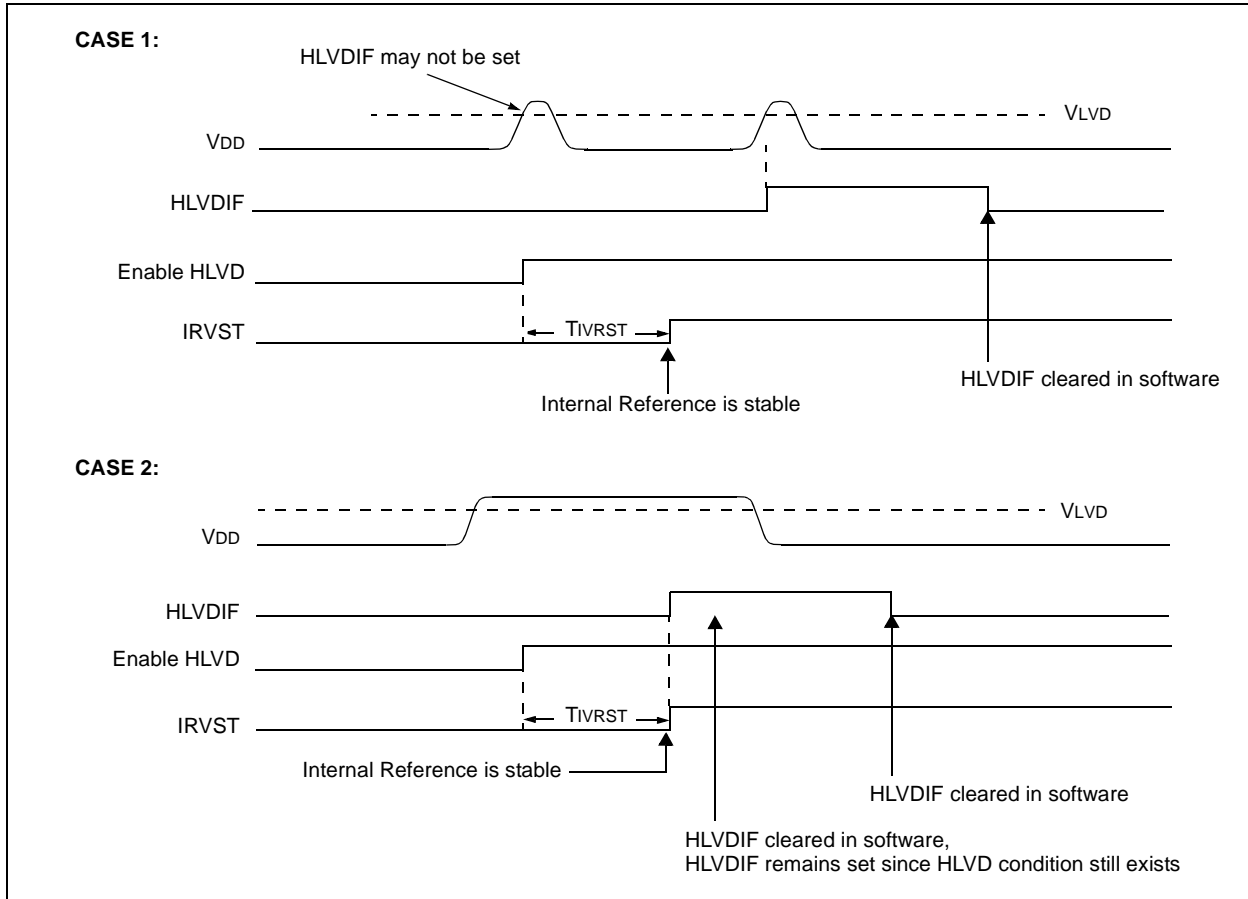
The HLVD interrupt flag is not enabled until T_{IRVST} has expired and a stable reference voltage is reached. For this reason, brief excursions beyond the set point may not be detected during this interval. Refer to Figure 24-2 or Figure 24-3.

FIGURE 24-2: LOW-VOLTAGE DETECT OPERATION (VDIRMAG = 0)



PIC18F2455/2550/4455/4550

FIGURE 24-3: HIGH-VOLTAGE DETECT OPERATION (VDIRMAG = 1)

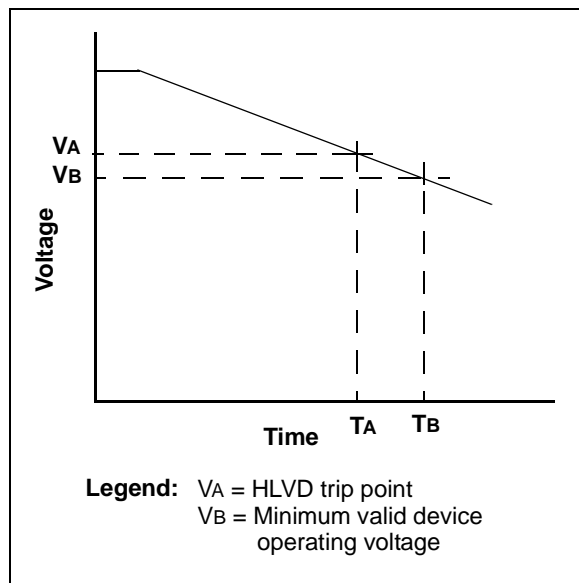


24.5 Applications

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 USB attach or detach. This assumes the device is powered by a lower voltage source than the Universal Serial Bus 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, V_A, the HLVD logic generates an interrupt at time, T_A. 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 T_B. The HLVD, thus, would give the application a time window, represented by the difference between T_A and T_B, to safely exit.

FIGURE 24-4: TYPICAL LOW-VOLTAGE DETECT APPLICATION



PIC18F2455/2550/4455/4550

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 | — | EEIF | BCLIF | HLVDIF | TMR3IF | CCP2IF | 54 |
| PIE2 | OCSFIE | CMIE | — | EEIE | BCLIE | HLVDIE | TMR3IE | CCP2IE | 54 |
| IPR2 | OSCFIP | CMIP | — | EEIP | BCLIP | HLVDIP | TMR3IP | CCP2IP | 54 |

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the HLVD module.

PIC18F2455/2550/4455/4550

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 switch over 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.

PIC18F2455/2550/4455/4550

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-3FFFFFFh), 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 EECON1 register WR bit 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 | — | — | USBPLL | CPUDIV1 | CPUDIV0 | PLLDIV2 | PLLDIV1 | PLLDIV0 | --00 0000 |
| 300001h | CONFIG1H | IESO | FCMEN | — | — | FOSC3 | FOSC2 | FOSC1 | FOSC0 | 00-- 0101 |
| 300002h | CONFIG2L | — | — | VREGEN | BORV1 | BORV0 | BOREN1 | BOREN0 | PWRTE \bar{N} | --01 1111 |
| 300003h | CONFIG2H | — | — | — | WDTPS3 | WDTPS2 | WDTPS1 | WDTPS0 | WDTEN | ---1 1111 |
| 300005h | CONFIG3H | MCLRE | — | — | — | — | LPT1OSC | PBADEN | CCP2MX | 1--- -011 |
| 300006h | CONFIG4L | DEBUG | XINST | ICPRT | — | — | LVP | — | STVREN | 100- -1-1 |
| 300008h | CONFIG5L | — | — | — | — | CP3 ⁽¹⁾ | CP2 | CP1 | CP0 | ---- 1111 |
| 300009h | CONFIG5H | CPD | CPB | — | — | — | — | — | — | 11-- ---- |
| 30000Ah | CONFIG6L | — | — | — | — | WRT3 ⁽¹⁾ | WRT2 | WRT1 | WRT0 | ---- 1111 |
| 30000Bh | CONFIG6H | WRD \bar{D} | WRB \bar{T} | WRT \bar{C} | — | — | — | — | — | 111- ---- |
| 30000Ch | CONFIG7L | — | — | — | — | EBTR3 ⁽¹⁾ | EBTR2 | EBTR1 | EBTR0 | ---- 1111 |
| 30000Dh | CONFIG7H | — | EBTR \bar{B} | — | — | — | — | — | — | -1-- ---- |
| 3FFFFEh | DEVID1 | DEV2 | DEV1 | DEV0 | REV4 | REV3 | REV2 | REV1 | REV0 | xxxx xxxx ⁽²⁾ |
| 3FFFFFFh | DEVID2 | DEV10 | DEV9 | DEV8 | DEV7 | DEV6 | DEV5 | DEV4 | DEV3 | 0000 1011 ⁽²⁾ |

Legend: x = unknown, u = unchanged, - = unimplemented. Shaded cells are unimplemented, read as '0'.

Note 1: Unimplemented in PIC18FX455 devices; maintain this bit set.

2: See Register 25-15 for DEVID1 values. DEVID registers are read-only and cannot be programmed by the user.

PIC18F2455/2550/4455/4550

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 | |
| — | — | USBPLL | CPUDIV1 | CPUDIV0 | PLLDIV2 | PLLDIV1 | PLLDIV0 | |
| bit 7 | | | | | | | | bit 0 |

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

bit 5 **USBPLL:** USB Clock Selection bit

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 = Primary oscillator divided by 6 to derive system clock

10 = Primary oscillator divided by 4 to derive system clock

01 = Primary oscillator divided by 3 to derive system clock

00 = Primary oscillator 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)

Legend:

R = Readable bit

P = Programmable bit U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

PIC18F2455/2550/4455/4550

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 | FOSC2 | FOSC1 | FOSC0 |
| bit 7 | | | | bit 0 | | | |

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 enabled
- 0 = Fail-Safe Clock Monitor disabled

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

bit 3-0 **FOSC3:FOSC0:** Oscillator Selection bits⁽¹⁾

- 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), XT used by USB

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 block.

Legend:

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

PIC18F2455/2550/4455/4550

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 | BORV0 | BOREN1 ⁽¹⁾ | BOREN0 ⁽¹⁾ | PWRTEN ⁽¹⁾ | |
| bit 7 | | | | | | | | bit 0 |

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

bit 5 **VREGEN:** USB Internal Voltage Regulator Enable bit

1 = USB voltage regulator enabled

0 = USB voltage regulator disabled

bit 4-3 **BORV1:BORV0:** Brown-out Reset Voltage bits

11 = VBOR set to 2.1V

10 = VBOR set to 2.8V

01 = VBOR set to 4.3V

00 = VBOR set to 4.6V

bit 2-1 **BOREN1:BOREN0** Brown-out Reset Enable bits⁽¹⁾

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⁽¹⁾

1 = PWRT disabled

0 = PWRT enabled

Note 1: The Power-up Timer is decoupled from Brown-out Reset, allowing these features to be independently controlled.

Legend:

R = Readable bit

P = Programmable bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

u = Unchanged from programmed state

PIC18F2455/2550/4455/4550

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 |

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 = 1:4
 0001 = 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)

Legend:

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

PIC18F2455/2550/4455/4550

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 | |

- bit 7 **MCLRE:** $\overline{\text{MCLR}}$ Pin Enable bit
1 = $\overline{\text{MCLR}}$ pin enabled; RE3 input pin disabled
0 = RE3 input pin enabled; $\overline{\text{MCLR}}$ disabled
- bit 6-3 **Unimplemented:** Read as '0'
- bit 2 **LPT1OSC:** Low-Power Timer 1 Oscillator Enable bit
1 = Timer1 configured for low-power operation
0 = 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 RC1
0 = CCP2 input/output is multiplexed with RB3

Legend:

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

PIC18F2455/2550/4455/4550

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 |
|---------------------------|-------|----------------------|-----|-----|-------|-----|--------|
| $\overline{\text{DEBUG}}$ | XINST | ICPRT ⁽¹⁾ | — | — | LVP | — | STVREN |
| bit 7 | | | | | | | bit 0 |

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⁽¹⁾

- 1 = ICPORT enabled
- 0 = ICPORT disabled

Note 1: Available only on PIC18F4455/4550 devices in 44-pin TQFP packages. Always leave this bit clear in all other devices.

bit 4-3 **Unimplemented:** Read as '0'

bit 2 **LVP:** Single-Supply ICSP™ Enable bit

- 1 = Single-Supply ICSP enabled
- 0 = 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

Legend:

R = Readable bit

C = Clearable bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

u = Unchanged from programmed state

PIC18F2455/2550/4455/4550

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 ⁽¹⁾ | CP2 | CP1 | CP0 |
| bit 7 | | | | bit 0 | | | |

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **CP3:** Code Protection bit⁽¹⁾

1 = Block 3 (006000-007FFFh) not code-protected

0 = Block 3 (006000-007FFFh) code-protected

Note 1: Unimplemented in PIC18FX455 devices; maintain this bit set.

bit 2 **CP2:** Code Protection bit

1 = Block 2 (004000-005FFFh) not code-protected

0 = Block 2 (004000-005FFFh) code-protected

bit 1 **CP1:** Code Protection bit

1 = Block 1 (002000-003FFFh) not code-protected

0 = Block 1 (002000-003FFFh) code-protected

bit 0 **CP0:** Code Protection bit

1 = Block 0 (000800-001FFFh) not code-protected

0 = Block 0 (000800-001FFFh) code-protected

Legend:

R = Readable bit

C = Clearable bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

u = Unchanged from programmed state

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 | CPB | — | — | — | — | — | — |
| bit 7 | | | | bit 0 | | | |

bit 7 **CPD:** Data EEPROM Code Protection bit

1 = Data EEPROM not code-protected

0 = Data EEPROM code-protected

bit 6 **CPB:** Boot Block Code Protection bit

1 = Boot block (000000-0007FFFh) not code-protected

0 = Boot block (000000-0007FFFh) code-protected

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

PIC18F2455/2550/4455/4550

REGISTER 25-9: CONFIG6L: CONFIGURATION REGISTER 6 LOW (BYTE ADDRESS 3000Ah)

| | | | | | | | |
|-------|-----|-----|-----|---------------------|-------|-------|-------|
| U-0 | U-0 | U-0 | U-0 | R/C-1 | R/C-1 | R/C-1 | R/C-1 |
| — | — | — | — | WRT3 ⁽¹⁾ | WRT2 | WRT1 | WRT0 |
| bit 7 | | | | bit 0 | | | |

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **WRT3:** Write Protection bit⁽¹⁾

- 1 = Block 3 (006000-007FFFh) not write-protected
- 0 = Block 3 (006000-007FFFh) write-protected

Note 1: Unimplemented in PIC18FX455 devices; maintain this bit set.

bit 2 **WRT2:** Write Protection bit

- 1 = Block 2 (004000-005FFFh) not write-protected
- 0 = Block 2 (004000-005FFFh) write-protected

bit 1 **WRT1:** Write Protection bit

- 1 = Block 1 (002000-003FFFh) not write-protected
- 0 = Block 1 (002000-003FFFh) write-protected

bit 0 **WRT0:** Write Protection bit

- 1 = Block 0 (000800-001FFFh) not write-protected
- 0 = Block 0 (000800-001FFFh) write-protected

Legend:

R = Readable bit C = Clearable bit U = Unimplemented bit, read as '0'
 -n = Value when device is unprogrammed u = Unchanged from programmed state

REGISTER 25-10: CONFIG6H: CONFIGURATION REGISTER 6 HIGH (BYTE ADDRESS 3000Bh)

| | | | | | | | |
|-------|-------|---------------------|-------|-----|-----|-----|-----|
| R/C-1 | R/C-1 | R-1 | U-0 | U-0 | U-0 | U-0 | U-0 |
| WRTD | WRTB | WRTC ⁽¹⁾ | — | — | — | — | — |
| bit 7 | | | bit 0 | | | | |

bit 7 **WRTD:** Data EEPROM Write Protection bit

- 1 = Data EEPROM not write-protected
- 0 = Data EEPROM write-protected

bit 6 **WRTB:** Boot Block Write Protection bit

- 1 = Boot block (000000-0007FFFh) not write-protected
- 0 = Boot block (000000-0007FFFh) write-protected

bit 5 **WRTC:** Configuration Register Write Protection bit⁽¹⁾

- 1 = Configuration registers (300000-300FFFh) not write-protected
- 0 = Configuration registers (300000-300FFFh) write-protected

Note 1: This bit is read-only in normal execution mode; it can be written only in Program mode.

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

PIC18F2455/2550/4455/4550

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 ⁽¹⁾ | EBTR2 | EBTR1 | EBTR0 |
| bit 7 | | | | bit 0 | | | |

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **EBTR3:** Table Read Protection bit⁽¹⁾

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

Note 1: Unimplemented in PIC18FX455 devices; maintain this bit set.

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) not protected from table reads executed in other blocks

0 = Block 1 (002000-003FFFh) protected from table reads executed in other blocks

bit 0 **EBTR0:** Table Read Protection bit

1 = Block 0 (000800-001FFFh) not protected from table reads executed in other blocks

0 = Block 0 (000800-001FFFh) protected from table reads executed in other blocks

Legend:

R = Readable bit

C = Clearable bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

u = Unchanged from programmed state

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 | | | |

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

bit 6 **EBTRB:** Boot Block Table Read Protection bit

1 = Boot block (000000-0007FFFh) not protected from table reads executed in other blocks

0 = Boot block (000000-0007FFFh) protected from table reads executed in other blocks

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

PIC18F2455/2550/4455/4550

REGISTER 25-13: 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

bit 7-5 **DEV2:DEV0:** Device ID bits

101 = PIC18F2455
011 = PIC18F2550
100 = PIC18F4455
010 = PIC18F4550

bit 4-0 **REV4:REV0:** Revision ID bits

These bits are used to indicate the device revision.

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

REGISTER 25-14: DEVICE ID REGISTER 2 FOR PIC18F2455/2550/4455/4550 DEVICES

| | | | | | | | |
|-------|------|------|------|------|------|------|------|
| R | R | R | R | R | R | R | R |
| DEV10 | DEV9 | DEV8 | DEV7 | DEV6 | DEV5 | DEV4 | DEV3 |

bit 7 bit 0

bit 7-0 **DEV10:DEV3:** Device ID bits

These bits are used with the DEV2:DEV0 bits in the Device ID Register 1 to identify the part number.

0000 1011 = PIC18F2455/2550/4455/4550 devices

Note: 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.

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

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.

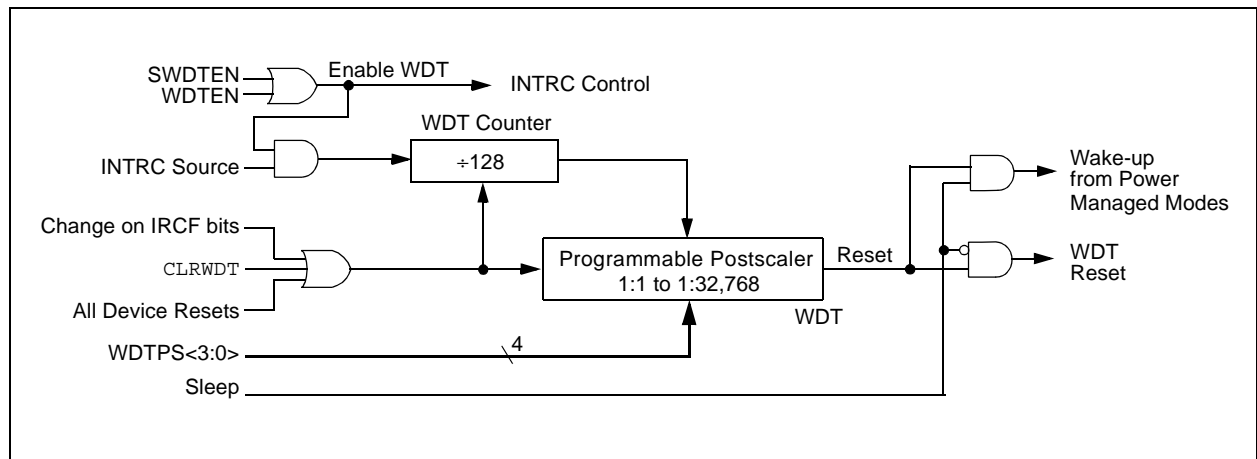
2: Changing the setting of the IRCF bits (OSCCON<6:4>) clears the WDT and postscaler counts.

3: 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



PIC18F2455/2550/4455/4550

REGISTER 25-15: WDTCON REGISTER

| | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|-----|-----------------------|
| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 |
| — | — | — | — | — | — | — | — | SWDTEN ⁽¹⁾ |
| bit 7 | | | | | | | | bit 0 |

bit 7-1 **Unimplemented:** Read as '0'

bit 0 **SWDTEN:** Software Controlled Watchdog Timer Enable bit⁽¹⁾

1 = Watchdog Timer is on

0 = Watchdog Timer is off

Note 1: This bit has no effect if the configuration bit WDTEN is enabled.

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

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 | — | \overline{RI} | \overline{TO} | \overline{PD} | \overline{POR} | \overline{BOR} | 52 |
| WDTCON | — | — | — | — | — | — | — | SWDTEN | 52 |

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the Watchdog Timer.

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.

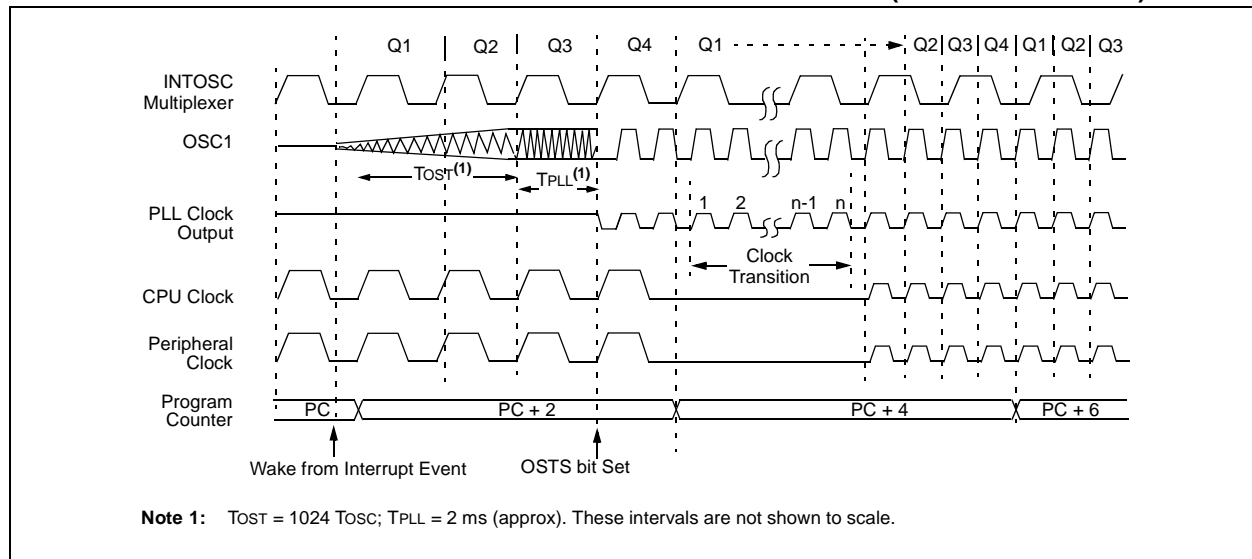
In all other power managed modes, Two-Speed Start-up is not used. The device will be clocked by the currently selected clock source until the primary clock source becomes available. The setting of the IESO bit is ignored.

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.

FIGURE 25-2: TIMING TRANSITION FOR TWO-SPEED START-UP (INTOSC TO HSPLL)



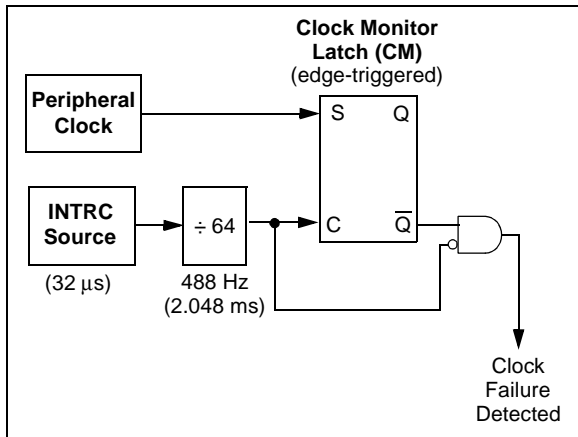
PIC18F2455/2550/4455/4550

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 shut-down. 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.

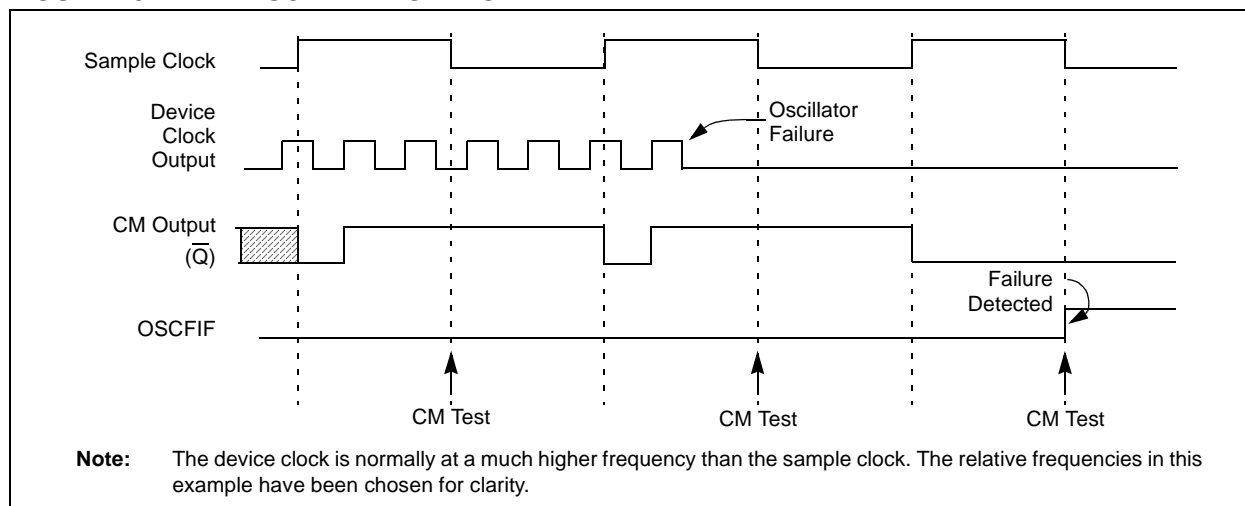
As already noted, the clock source is switched to the INTOSC clock when a clock failure is detected. Depending on the frequency selected by the IRCF2:IRCF0 bits, this may mean a substantial change in the speed of code execution. If the WDT is enabled with a small prescale value, a decrease in clock speed 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 required 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.

FIGURE 25-4: FSCM TIMING DIAGRAM



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 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 modes, monitoring can begin immediately following these events.

For oscillator modes involving a crystal or resonator (HS, HSPLL, LP or XT), the situation is somewhat different. Since the oscillator may require a start-up time considerably longer than the FSCM sample clock time, a false clock failure may be detected. To prevent this, the internal oscillator block is automatically configured as the device clock and functions until the primary clock is stable (the OST and PLL timers have timed out). This is identical to Two-Speed Start-up mode. Once the primary clock is stable, the 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.

PIC18F2455/2550/4455/4550

25.5.1 PROGRAM MEMORY CODE PROTECTION

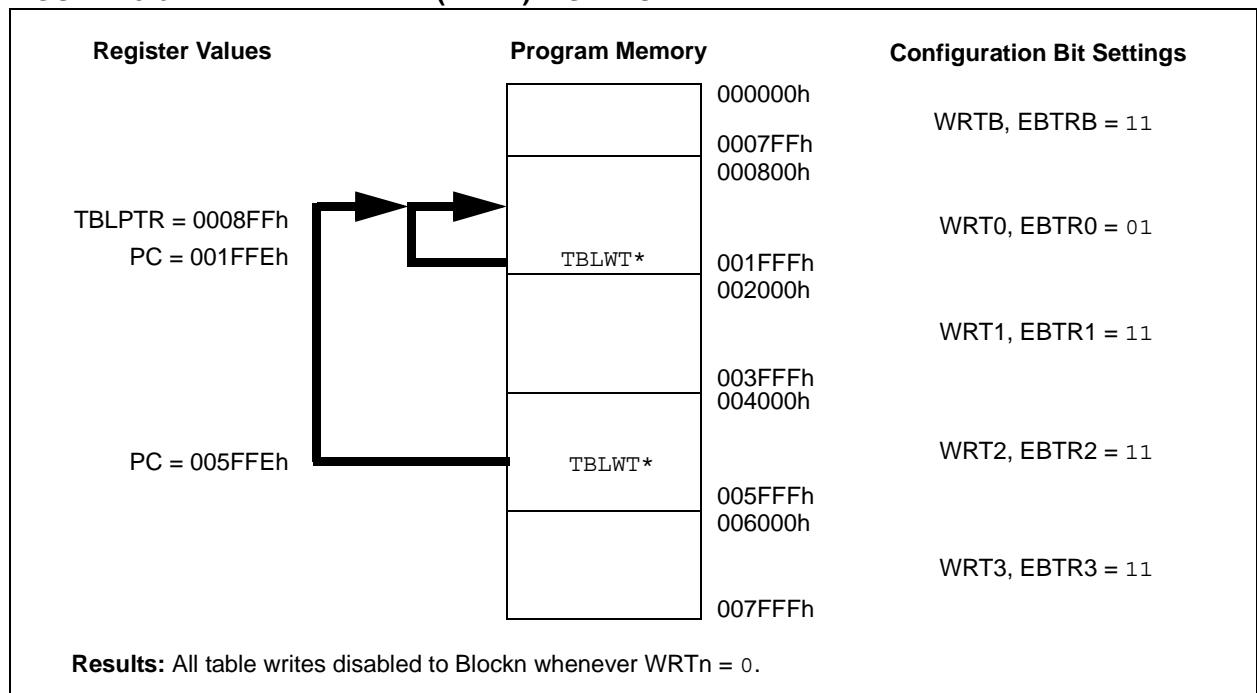
The program memory may be read to or written from any location using the table read and table write instructions. The device ID may be read with table reads. The configuration registers may be read and written with the table read and table write instructions.

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.

Note: 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



PIC18F2455/2550/4455/4550

FIGURE 25-7: EXTERNAL BLOCK TABLE READ (EBTRn) DISALLOWED

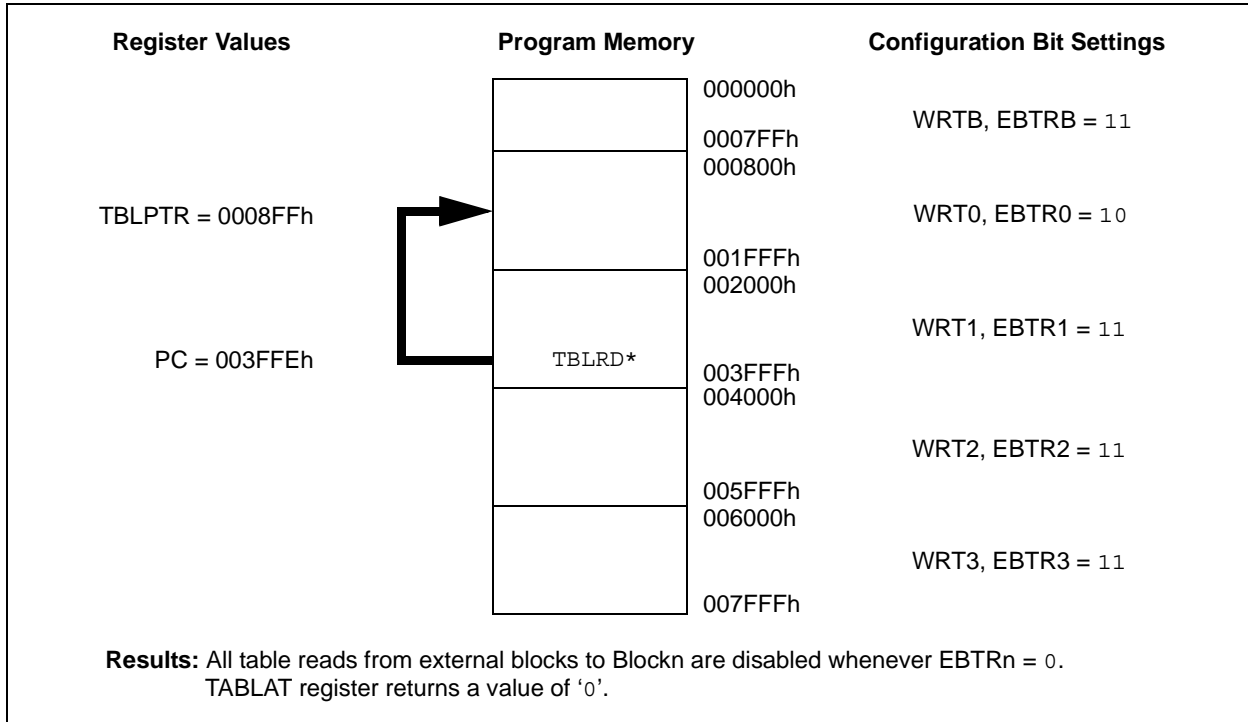
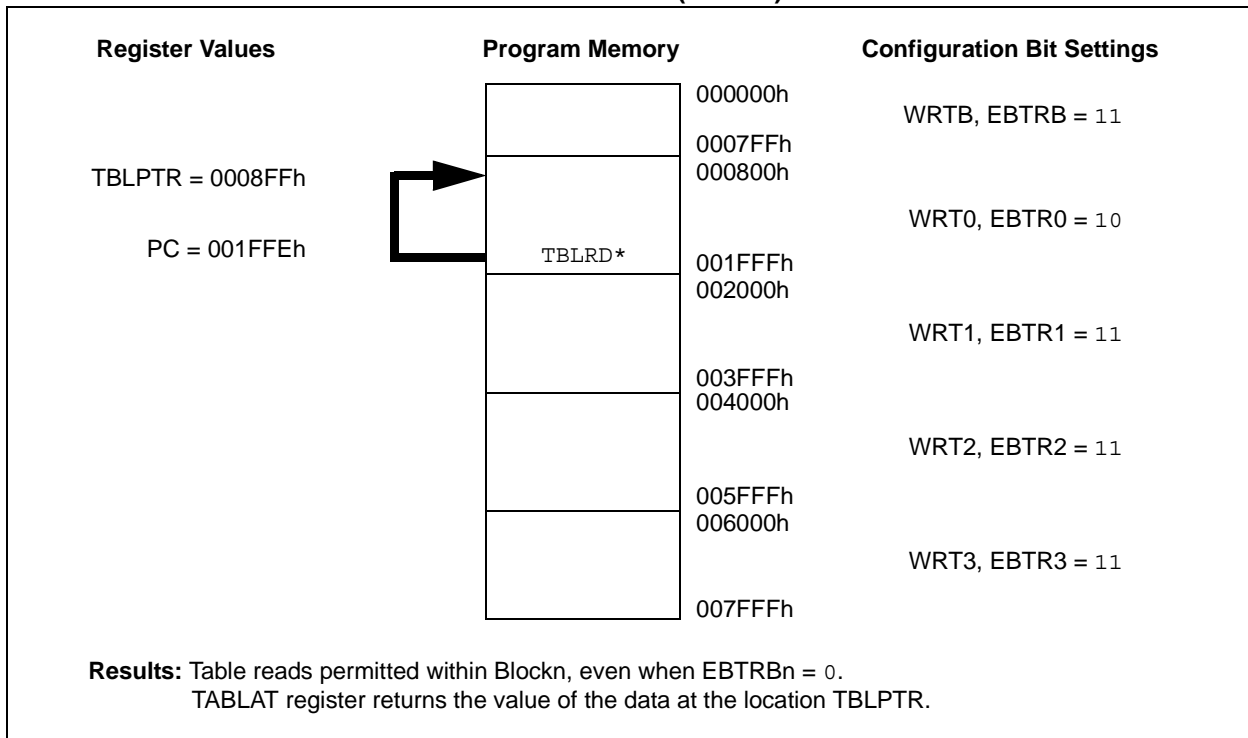


FIGURE 25-8: EXTERNAL BLOCK TABLE READ (EBTRn) ALLOWED



PIC18F2455/2550/4455/4550

25.5.2 DATA EEPROM CODE PROTECTION

The entire data EEPROM is protected from external reads and writes by two bits: CPD and WRD. CPD inhibits external reads and writes of data EEPROM. WRD 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 '0', the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB® IDE. When the microcontroller has this feature enabled, some resources are not available for general use. Table 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 ICSPORTS 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 Name | | Pin Type | Pin Function |
|--------------|----------------|----------|-------------------------------------|
| Legacy Port | Dedicated Port | | |
| MCLR/VPP/RE3 | NC/ICRST/ICVPP | P | 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

PIC18F2455/2550/4455/4550

Even when the dedicated port is enabled, the ICSP and ICD functions remain available through the legacy port. When V_{IH} is seen on the $\overline{MCLR}/V_{PP}/RE3$ pin, the state of the $\overline{ICRST}/ICV_{PP}$ pin is ignored.

Note 1: The ICPORT configuration bit can only be programmed through the default ICSP port.

2: The ICPORT 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, $\overline{NC}/ICPORTS$. Connecting this pin to V_{SS} 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 \overline{PORTD} and \overline{PORTE} , the \overline{SPP} and the Enhanced PWM functionality of $\overline{CCP1}$. On the other hand, connecting the pin to V_{DD} 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 (\overline{DEBUG} configuration bit is clear and \overline{ICPRT} configuration bit is set). When disabled, $\overline{NC}/ICPORTS$ 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 $\overline{MCLR}/V_{PP}/RE3$ pin, but the $\overline{RB5}/\overline{KB11}/\overline{PGM}$ pin is then dedicated to controlling Program mode entry and is not available as a general purpose I/O pin.

While programming using Single-Supply Programming, V_{DD} is applied to the $\overline{MCLR}/V_{PP}/RE3$ pin as in normal execution mode. To enter Programming mode, V_{DD} is applied to the \overline{PGM} pin.

Note 1: High-Voltage Programming is always available, regardless of the state of the LVP bit, by applying V_{IH} to the \overline{MCLR} pin.

2: While in Low-Voltage ICSP Programming mode, the $\overline{RB5}$ 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 \overline{PORTB} are enabled, bit 5 in the \overline{TRISB} register must be cleared to disable the pull-up on $\overline{RB5}$ 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 ($\overline{CONFIG4}<2> = 0$); or

b) make certain that $\overline{RB5}/\overline{KB11}/\overline{PGM}$ is held low during entry into ICSP.

If Single-Supply ICSP Programming mode will not be used, the LVP bit can be cleared. $\overline{RB5}/\overline{KB11}/\overline{PGM}$ then becomes available as the digital I/O pin, $\overline{RB5}$. The LVP bit may be set or cleared only when using standard high-voltage programming (V_{IH} applied to the $\overline{MCLR}/V_{PP}/RE3$ 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 V_{DD} . 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 V_{DD} 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 8 new instructions for the optimization of code that is recursive or that utilizes a software stack. The extended set is discussed later in this section.

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 μ s. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 μ s. Two-word branch instructions (if true) would take 3 μ s.

Figure 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 Assembler (MPASM™).

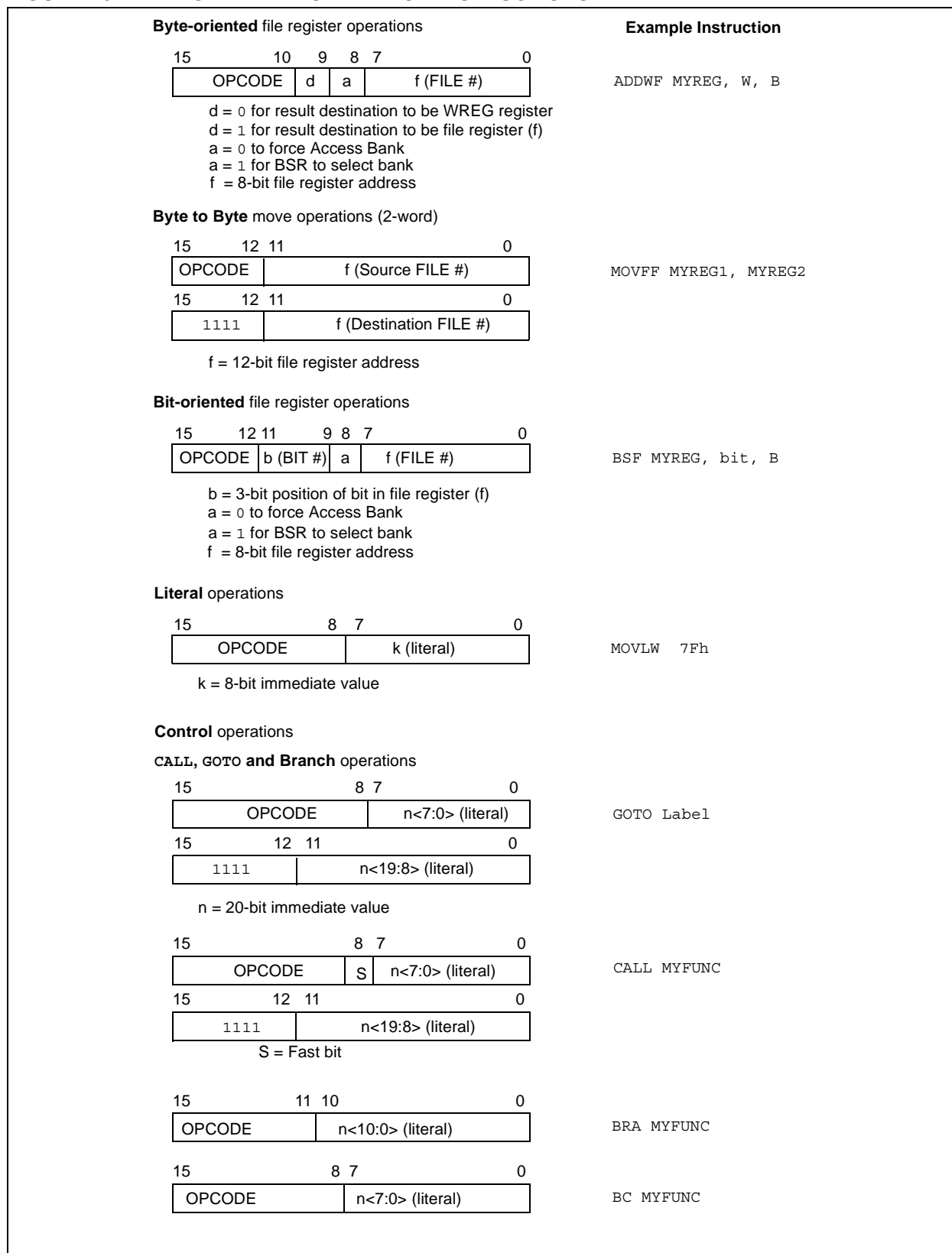
Section 26.1.1 "Standard Instruction Set" provides a description of each instruction.

PIC18F2455/2550/4455/4550

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: C arry, D igit Carry, Z ero, O verflow, N egative. |
| d | Destination select bit d = 0: store result in WREG d = 1: store result in file register f |
| dest | Destination: either the WREG register or the specified register file location. |
| f | 8-bit register file address (00h to FFh) or 2-bit FSR designator (0h to 3h). |
| f _s | 12-bit register file address (000h to FFFh). This is the source address. |
| f _d | 12-bit register file address (000h to FFFh). This is the destination address. |
| GIE | Global Interrupt Enable bit. |
| k | Literal field, constant data or label (may be either an 8-bit, 12-bit or a 20-bit value). |
| label | Label name. |
| mm | The mode of the TBLPTR register for the table read and table write instructions. Only used with table read and table write instructions: |
| * | No change to register (such as TBLPTR with table reads and writes) |
| *+ | Post-Increment register (such as TBLPTR with table reads and writes) |
| *- | Post-Decrement register (such as TBLPTR with table reads and writes) |
| +* | Pre-Increment register (such as TBLPTR with table reads and writes) |
| n | The relative address (2's complement number) for relative branch instructions or the direct address for Call/Branch and Return instructions. |
| PC | Program Counter. |
| PCL | Program Counter Low Byte. |
| PCH | Program Counter High Byte. |
| PCLATH | Program Counter High Byte Latch. |
| PCLATU | Program Counter Upper Byte Latch. |
| \overline{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. |
| \overline{TO} | Time-out bit. |
| TOS | Top-of-Stack. |
| u | Unused or unchanged. |
| WDT | Watchdog Timer. |
| WREG | Working register (accumulator). |
| x | Don't care ('0' or '1'). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools. |
| z _s | 7-bit offset value for indirect addressing of register files (source). |
| z _d | 7-bit offset value for indirect addressing of register files (destination). |
| { } | Optional argument. |
| [text] | Indicates an indexed address. |
| (text) | The contents of text. |
| [expr] <n> | Specifies bit n of the register indicated by the pointer expr. |
| → | Assigned to. |
| < > | Register bit field. |
| ∈ | In the set of. |
| <i>italics</i> | User defined term (font is Courier). |

FIGURE 26-1: GENERAL FORMAT FOR INSTRUCTIONS



PIC18F2455/2550/4455/4550

TABLE 26-2: PIC18FXXX INSTRUCTION SET

| Mnemonic, Operands | Description | Cycles | 16-Bit Instruction Word | | | | Status Affected | Notes | |
|---------------------------------|---------------------------------|--|-------------------------|------|------|------|--------------------|-----------------|------------|
| | | | MSb | | | LSb | | | |
| BYTE-ORIENTED OPERATIONS | | | | | | | | | |
| ADDWF | f, d, a | Add WREG and f | 1 | 0010 | 01da | ffff | ffff | C, DC, Z, OV, N | 1, 2 |
| ADDWFC | f, d, a | Add WREG and Carry bit to f | 1 | 0010 | 00da | ffff | ffff | C, DC, Z, OV, N | 1, 2 |
| ANDWF | f, d, a | AND WREG with f | 1 | 0001 | 01da | ffff | ffff | Z, N | 1, 2 |
| CLRF | f, a | Clear f | 1 | 0110 | 101a | ffff | ffff | Z | 2 |
| COMF | f, d, a | Complement f | 1 | 0001 | 11da | ffff | ffff | Z, N | 1, 2 |
| CPFSEQ | f, a | Compare f with WREG, skip = | 1 (2 or 3) | 0110 | 001a | ffff | ffff | None | 4 |
| CPFSGT | f, a | Compare f with WREG, skip > | 1 (2 or 3) | 0110 | 010a | ffff | ffff | None | 4 |
| CPFSLT | f, a | Compare f with WREG, skip < | 1 (2 or 3) | 0110 | 000a | ffff | ffff | None | 1, 2 |
| DECf | f, d, a | Decrement f | 1 | 0000 | 01da | ffff | ffff | C, DC, Z, OV, N | 1, 2, 3, 4 |
| DECFSZ | f, d, a | Decrement f, Skip if 0 | 1 (2 or 3) | 0010 | 11da | ffff | ffff | None | 1, 2, 3, 4 |
| DCFSNZ | f, d, a | Decrement f, Skip if Not 0 | 1 (2 or 3) | 0100 | 11da | ffff | ffff | None | 1, 2 |
| INCF | f, d, a | Increment f | 1 | 0010 | 10da | ffff | ffff | C, DC, Z, OV, N | 1, 2, 3, 4 |
| INCFSZ | f, d, a | Increment f, Skip if 0 | 1 (2 or 3) | 0011 | 11da | ffff | ffff | None | 4 |
| INFSNZ | f, d, a | Increment f, Skip if Not 0 | 1 (2 or 3) | 0100 | 10da | ffff | ffff | None | 1, 2 |
| 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 _s (source) to f _d (destination) | 2 | 1100 | ffff | ffff | ffff | None | |
| MOVWF | f, a | Move WREG to f | 1 | 0110 | 111a | ffff | ffff | None | |
| MULWF | f, a | Multiply WREG with f | 1 | 0000 | 001a | ffff | ffff | None | 1, 2 |
| NEGF | f, a | Negate f | 1 | 0110 | 110a | ffff | ffff | C, DC, Z, OV, N | |
| RLCF | f, d, a | Rotate Left f through Carry | 1 | 0011 | 01da | ffff | ffff | C, Z, N | 1, 2 |
| RLNCF | f, d, a | Rotate Left f (No Carry) | 1 | 0100 | 01da | ffff | ffff | Z, N | |
| RRCF | f, d, a | Rotate Right f through Carry | 1 | 0011 | 00da | ffff | ffff | C, Z, N | |
| RRNCF | f, d, a | Rotate Right f (No Carry) | 1 | 0100 | 00da | ffff | ffff | Z, N | |
| SETf | f, a | Set f | 1 | 0110 | 100a | ffff | ffff | None | 1, 2 |
| SUBFWB | f, d, a | Subtract f from WREG with borrow | 1 | 0101 | 01da | ffff | ffff | C, DC, Z, OV, N | |
| SUBWF | f, d, a | Subtract WREG from f | 1 | 0101 | 11da | ffff | ffff | C, DC, Z, OV, N | 1, 2 |
| SUBWFB | f, d, a | Subtract WREG from f with borrow | 1 | 0101 | 10da | ffff | ffff | C, DC, Z, OV, N | |
| SWAPF | f, d, a | Swap nibbles in f | 1 | 0011 | 10da | ffff | ffff | None | 4 |
| TSTFSZ | f, a | Test f, skip if 0 | 1 (2 or 3) | 0110 | 011a | ffff | ffff | None | 1, 2 |
| XORWF | f, d, a | Exclusive OR WREG with f | 1 | 0001 | 10da | ffff | ffff | Z, N | |

- Note 1:** When a Port register is modified as a function of itself (e.g., MOVF PORTB, 1, 0), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and where applicable, 'd' = 1), the prescaler will be cleared if assigned.
- 3:** If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.
- 4:** Some instructions are two-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

PIC18F2455/2550/4455/4550

TABLE 26-2: PIC18FXXX INSTRUCTION SET (CONTINUED)

| Mnemonic, Operands | Description | Cycles | 16-Bit Instruction Word | | | | Status Affected | Notes | |
|--------------------------------|-------------|--------------------------------|-------------------------|------|------|------|--------------------|-----------------------------------|------|
| | | | MSb | LSb | | | | | |
| BIT-ORIENTED OPERATIONS | | | | | | | | | |
| BCF | f, b, a | Bit Clear f | 1 | 1001 | bbba | ffff | ffff | None | 1, 2 |
| BSF | f, b, a | Bit Set f | 1 | 1000 | bbba | ffff | ffff | None | 1, 2 |
| 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 OPERATIONS | | | | | | | | | |
| BC | n | Branch if Carry | 1 (2) | 1110 | 0010 | nnnn | nnnn | None | 4 |
| 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 | \overline{TO} , \overline{PD} | |
| DAW | — | Decimal Adjust WREG | 1 | 0000 | 0000 | 0000 | 0111 | C | |
| 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 | |
| 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 | \overline{TO} , \overline{PD} | |

- Note 1:** When a Port register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and where applicable, 'd' = 1), the prescaler will be cleared if assigned.
- 3:** If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.
- 4:** Some instructions are two-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

PIC18F2455/2550/4455/4550

TABLE 26-2: PIC18FXXX INSTRUCTION SET (CONTINUED)

| Mnemonic, Operands | Description | Cycles | 16-Bit Instruction Word | | | | Status Affected | Notes | |
|--|-------------|--|-------------------------|------|------|------|--------------------|-----------------|--|
| | | | MSb | | | LSb | | | |
| LITERAL OPERATIONS | | | | | | | | | |
| ADDLW | k | Add literal and WREG | 1 | 0000 | 1111 | kkkk | kkkk | C, DC, Z, OV, N | |
| ANDLW | k | AND literal with WREG | 1 | 0000 | 1011 | kkkk | kkkk | Z, N | |
| IORLW | k | Inclusive OR literal with WREG | 1 | 0000 | 1001 | kkkk | kkkk | Z, N | |
| LFSR | f, k | Move literal (12-bit) 2nd word to FSR(f) 1st word | 2 | 1110 | 1110 | 00ff | kkkk | None | |
| MOVLB | k | Move literal to BSR<3:0> | 1 | 0000 | 0001 | 0000 | kkkk | None | |
| MOVLW | k | Move literal to WREG | 1 | 0000 | 1110 | kkkk | kkkk | None | |
| MULLW | k | Multiply literal with WREG | 1 | 0000 | 1101 | kkkk | kkkk | None | |
| RETLW | k | Return with literal in WREG | 2 | 0000 | 1100 | kkkk | kkkk | None | |
| SUBLW | k | Subtract WREG from literal | 1 | 0000 | 1000 | kkkk | kkkk | C, DC, Z, OV, N | |
| XORLW | k | Exclusive OR literal with WREG | 1 | 0000 | 1010 | kkkk | kkkk | Z, N | |
| DATA MEMORY ↔ PROGRAM MEMORY OPERATIONS | | | | | | | | | |
| TBLRD* | | Table Read | 2 | 0000 | 0000 | 0000 | 1000 | None | |
| TBLRD*+ | | Table Read with post-increment | | 0000 | 0000 | 0000 | 1001 | None | |
| TBLRD*- | | Table Read with post-decrement | | 0000 | 0000 | 0000 | 1010 | None | |
| TBLRD*+ | | Table Read with pre-increment | | 0000 | 0000 | 0000 | 1011 | None | |
| TBLWT* | | Table Write | 2 | 0000 | 0000 | 0000 | 1100 | None | |
| TBLWT*+ | | Table Write with post-increment | | 0000 | 0000 | 0000 | 1101 | None | |
| TBLWT*- | | Table Write with post-decrement | | 0000 | 0000 | 0000 | 1110 | None | |
| TBLWT*+ | | Table Write with pre-increment | | 0000 | 0000 | 0000 | 1111 | None | |

- Note 1:** When a Port register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and where applicable, 'd' = 1), the prescaler will be cleared if assigned.
- 3:** If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.
- 4:** Some instructions are two-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

PIC18F2455/2550/4455/4550

26.1.1 STANDARD INSTRUCTION SET

ADDLW ADD literal to W

Syntax: ADDLW k
 Operands: $0 \leq k \leq 255$
 Operation: $(W) + k \rightarrow W$
 Status Affected: N, OV, C, DC, Z
 Encoding:

| | | | |
|------|------|------|------|
| 0000 | 1111 | kkkk | kkkk |
|------|------|------|------|

 Description: The contents of W are added to the 8-bit literal 'k' and the result is placed in W.
 Words: 1
 Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|------------------|--------------|------------|
| Decode | Read literal 'k' | Process Data | Write to W |

Example: ADDLW 15h

Before Instruction
 W = 10h
 After Instruction
 W = 25h

ADDWF ADD W to f

Syntax: ADDWF f {,d {,a}}
 Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$
 Operation: $(W) + (f) \rightarrow \text{dest}$
 Status Affected: N, OV, C, DC, Z
 Encoding:

| | | | |
|------|------|------|------|
| 0010 | 01da | ffff | ffff |
|------|------|------|------|

 Description: Add W to register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).
 If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).
 If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1
 Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|-------------------|--------------|----------------------|
| Decode | Read register 'f' | Process Data | Write to destination |

Example: ADDWF REG, 0, 0

Before Instruction
 W = 17h
 REG = 0C2h
 After Instruction
 W = 0D9h
 REG = 0C2h

Note: All PIC18 instructions may take an optional label argument, preceding the instruction mnemonic, for use in symbolic addressing. If a label is used, the instruction format then becomes: {label} instruction argument(s).

PIC18F2455/2550/4455/4550

ADDWFC **ADD W and Carry bit to f**

Syntax: ADDWFC f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: $(W) + (f) + (C) \rightarrow \text{dest}$

Status Affected: N, OV, C, DC, Z

Encoding:

| | | | |
|------|------|------|------|
| 0010 | 00da | ffff | ffff |
|------|------|------|------|

Description: Add W, the Carry flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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: ADDWFC REG, 0, 1

Before Instruction

 Carry bit = 1
 REG = 02h
 W = 4Dh

After Instruction

 Carry bit = 0
 REG = 02h
 W = 50h

ANDLW **AND literal with W**

Syntax: ANDLW k

Operands: $0 \leq k \leq 255$

Operation: $(W) .\text{AND. } k \rightarrow W$

Status Affected: N, Z

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 1011 | kkkk | kkkk |
|------|------|------|------|

Description: The contents of W are ANDed with the 8-bit literal 'k'. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|------------------|--------------|------------|
| Decode | Read literal 'k' | Process Data | Write to W |

Example: ANDLW 05Fh

Before Instruction

 W = A3h

After Instruction

 W = 03h

PIC18F2455/2550/4455/4550

ANDWF AND W with f

Syntax: ANDWF f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: (W) .AND. (f) → dest

Status Affected: N, Z

Encoding:

| | | | |
|------|------|------|------|
| 0001 | 01da | ffff | ffff |
|------|------|------|------|

Description: The contents of W are AND'ed 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 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
Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|-------------------|--------------|----------------------|
| Decode | Read register 'f' | Process Data | Write to destination |

Example: ANDWF REG, 0, 0

Before Instruction
 W = 17h
 REG = C2h
 After Instruction
 W = 02h
 REG = C2h

BC Branch if Carry

Syntax: BC n

Operands: $-128 \leq n \leq 127$

Operation: if Carry bit is '1'
 $(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 1110 | 0010 | nnnn | nnnn |
|------|------|------|------|

Description: If the Carry bit is '1', then the program will branch.
 The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 2 + 2n$. This instruction is then a two-cycle instruction.

Words: 1
Cycles: 1(2)
Q Cycle Activity:
 If Jump:

| Q1 | Q2 | Q3 | Q4 |
|--------------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | Write to PC |
| No operation | No operation | No operation | No operation |

If No Jump:

| Q1 | Q2 | Q3 | Q4 |
|--------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | No operation |

Example: HERE BC 5

Before Instruction
 PC = address (HERE)
 After Instruction
 If Carry = 1;
 PC = address (HERE + 12)
 If Carry = 0;
 PC = address (HERE + 2)

PIC18F2455/2550/4455/4550

BCF Bit Clear f

Syntax: BCF f, b {,a}

Operands: $0 \leq f \leq 255$
 $0 \leq b \leq 7$
 $a \in [0,1]$

Operation: $0 \rightarrow f \leftarrow b$

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 1001 | bbba | ffff | ffff |
|------|------|------|------|

Description: Bit 'b' in register 'f' is cleared.
 If 'a' is '0', the Access Bank is selected.
 If 'a' is '1', the BSR is used to select the GPR bank (default).
 If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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 register 'f' |

Example: BCF FLAG_REG, 7, 0

Before Instruction
 FLAG_REG = C7h

After Instruction
 FLAG_REG = 47h

BN Branch if Negative

Syntax: BN n

Operands: $-128 \leq n \leq 127$

Operation: if Negative bit is '1'
 $(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 1110 | 0110 | nnnn | nnnn |
|------|------|------|------|

Description: If the Negative bit is '1', then the program will branch.
 The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 2 + 2n$. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

If Jump:

| Q1 | Q2 | Q3 | Q4 |
|--------------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | Write to PC |
| No operation | No operation | No operation | No operation |

If No Jump:

| Q1 | Q2 | Q3 | Q4 |
|--------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | No operation |

Example: HERE BN Jump

Before Instruction
 PC = address (HERE)

After Instruction
 If Negative = 1;
 PC = address (Jump)
 If Negative = 0;
 PC = address (HERE + 2)

PIC18F2455/2550/4455/4550

BNC Branch if Not Carry

Syntax: BNC n

Operands: $-128 \leq n \leq 127$

Operation: if Carry bit is '0'
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 1110 | 0011 | nnnn | nnnn |
|------|------|------|------|

Description: If the Carry bit is '0', then the program will branch.
The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

| Q1 | Q2 | Q3 | Q4 |
|--------------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | Write to PC |
| No operation | No operation | No operation | No operation |

If No Jump:

| Q1 | Q2 | Q3 | Q4 |
|--------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | No operation |

Example: HERE BNC Jump

Before Instruction
PC = address (HERE)

After Instruction
If Carry = 0;
PC = address (Jump)

If Carry = 1;
PC = address (HERE + 2)

BNN Branch if Not Negative

Syntax: BNN n

Operands: $-128 \leq n \leq 127$

Operation: if Negative bit is '0'
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 1110 | 0111 | nnnn | nnnn |
|------|------|------|------|

Description: If the Negative bit is '0', then the program will branch.
The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

| Q1 | Q2 | Q3 | Q4 |
|--------------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | Write to PC |
| No operation | No operation | No operation | No operation |

If No Jump:

| Q1 | Q2 | Q3 | Q4 |
|--------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | No operation |

Example: HERE BNN Jump

Before Instruction
PC = address (HERE)

After Instruction
If Negative = 0;
PC = address (Jump)

If Negative = 1;
PC = address (HERE + 2)

PIC18F2455/2550/4455/4550

BNOV Branch if Not Overflow

Syntax: BNOV n

Operands: $-128 \leq n \leq 127$

Operation: if Overflow bit is '0'
 $(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 1110 | 0101 | nnnn | nnnn |
|------|------|------|------|

Description: If the Overflow bit is '0', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 2 + 2n$. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

| Q1 | Q2 | Q3 | Q4 |
|--------------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | Write to PC |
| No operation | No operation | No operation | No operation |

If No Jump:

| Q1 | Q2 | Q3 | Q4 |
|--------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | No operation |

Example: HERE BNOV Jump

Before Instruction
PC = address (HERE)

After Instruction
If Overflow = 0;
PC = address (Jump)
If Overflow = 1;
PC = address (HERE + 2)

BNZ Branch if Not Zero

Syntax: BNZ n

Operands: $-128 \leq n \leq 127$

Operation: if Zero bit is '0'
 $(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 1110 | 0001 | nnnn | nnnn |
|------|------|------|------|

Description: If the Zero bit is '0', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 2 + 2n$. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

| Q1 | Q2 | Q3 | Q4 |
|--------------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | Write to PC |
| No operation | No operation | No operation | No operation |

If No Jump:

| Q1 | Q2 | Q3 | Q4 |
|--------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | No operation |

Example: HERE BNZ Jump

Before Instruction
PC = address (HERE)

After Instruction
If Zero = 0;
PC = address (Jump)
If Zero = 1;
PC = address (HERE + 2)

PIC18F2455/2550/4455/4550

BRA Unconditional Branch

Syntax: BRA n

Operands: $-1024 \leq n \leq 1023$

Operation: $(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 1101 | 0nnn | nnnn | nnnn |
|------|------|------|------|

Description: Add the 2's complement number '2n' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 2 + 2n$. This instruction is a two-cycle instruction.

Words: 1

Cycles: 2

Q Cycle Activity:

| | Q1 | Q2 | Q3 | Q4 |
|--------------|------------------|--------------|--------------|----|
| Decode | Read literal 'n' | Process Data | Write to PC | |
| No operation | No operation | No operation | No operation | |

Example: HERE BRA Jump

Before Instruction
PC = address (HERE)

After Instruction
PC = address (Jump)

BSF Bit Set f

Syntax: BSF f, b {,a}

Operands: $0 \leq f \leq 255$
 $0 \leq b \leq 7$
 $a \in [0,1]$

Operation: $1 \rightarrow f$

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 1000 | bbba | ffff | ffff |
|------|------|------|------|

Description: Bit 'b' in register 'f' is set. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

| | Q1 | Q2 | Q3 | Q4 |
|--------|-------------------|--------------|--------------------|----|
| Decode | Read register 'f' | Process Data | Write register 'f' | |

Example: BSF FLAG_REG, 7, 1

Before Instruction
FLAG_REG = 0Ah

After Instruction
FLAG_REG = 8Ah

PIC18F2455/2550/4455/4550

BTFSK **Bit Test File, Skip if Clear**

Syntax: BTFSK f, b {,a}

Operands: $0 \leq f \leq 255$
 $0 \leq b \leq 7$
 $a \in [0,1]$

Operation: skip if (f) = 0

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 1011 | bbba | ffff | ffff |
|------|------|------|------|

Description: If bit 'b' in register 'f' is '0', then the next instruction is skipped. If bit 'b' is '0', then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this a two-cycle instruction.
 If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).
 If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh).
 See **Section 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 register 'f' | Process Data | No operation |

If skip:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |

If skip and followed by 2-word instruction:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |
| No operation | No operation | No operation | No operation |

Example:

```

HERE   BTFSK   FLAG, 1, 0
FALSE  :
TRUE   :
```

Before Instruction
 PC = address (HERE)

After Instruction
 If FLAG<1> = 0;
 PC = address (TRUE)
 If FLAG<1> = 1;
 PC = address (FALSE)

BTFSK **Bit Test File, Skip if Set**

Syntax: BTFSK f, b {,a}

Operands: $0 \leq f \leq 255$
 $0 \leq b < 7$
 $a \in [0,1]$

Operation: skip if (f) = 1

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 1010 | bbba | ffff | ffff |
|------|------|------|------|

Description: If bit 'b' in register 'f' is '1', then the next instruction is skipped. If bit 'b' is '1', then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this a two-cycle instruction.
 If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).
 If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh).
 See **Section 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 register 'f' | Process Data | No operation |

If skip:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |

If skip and followed by 2-word instruction:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |
| No operation | No operation | No operation | No operation |

Example:

```

HERE   BTFSK   FLAG, 1, 0
FALSE  :
TRUE   :
```

Before Instruction
 PC = address (HERE)

After Instruction
 If FLAG<1> = 0;
 PC = address (FALSE)
 If FLAG<1> = 1;
 PC = address (TRUE)

PIC18F2455/2550/4455/4550

BTG Bit Toggle f

Syntax: BTG f, b {,a}
 Operands: $0 \leq f \leq 255$
 $0 \leq b < 7$
 $a \in [0,1]$
 Operation: $(\bar{f} < b >) \rightarrow f < b >$
 Status Affected: None
 Encoding:

| | | | |
|------|------|------|------|
| 0111 | bbba | ffff | ffff |
|------|------|------|------|

Description: Bit 'b' in data memory location 'f' is inverted.
 If 'a' is '0', the Access Bank is selected.
 If 'a' is '1', the BSR is used to select the GPR bank (default).
 If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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 register 'f' |

Example: BTG PORTC, 4, 0

Before Instruction:
 PORTC = 0111 0101 [75h]
 After Instruction:
 PORTC = 0110 0101 [65h]

BOV Branch if Overflow

Syntax: BOV n
 Operands: $-128 \leq n \leq 127$
 Operation: if Overflow bit is '1'
 $(PC) + 2 + 2n \rightarrow PC$
 Status Affected: None
 Encoding:

| | | | |
|------|------|------|------|
| 1110 | 0100 | 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 'n' | Process Data | Write to PC |
| No operation | No operation | No operation | No operation |

If No Jump:

| Q1 | Q2 | Q3 | Q4 |
|--------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | No operation |

Example: HERE BOV Jump
 Before Instruction
 PC = address (HERE)
 After Instruction
 If Overflow = 1;
 PC = address (Jump)
 If Overflow = 0;
 PC = address (HERE + 2)

PIC18F2455/2550/4455/4550

BZ Branch if Zero

Syntax: BZ n

Operands: $-128 \leq n \leq 127$

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 two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
 If Jump:

| Q1 | Q2 | Q3 | Q4 |
|--------------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | Write to PC |
| No operation | No operation | No operation | No operation |

If No Jump:

| Q1 | Q2 | Q3 | Q4 |
|--------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | No operation |

Example: HERE BZ Jump

Before Instruction
 PC = address (HERE)

After Instruction
 If Zero = 1;
 PC = address (Jump)
 If Zero = 0;
 PC = address (HERE + 2)

CALL Subroutine Call

Syntax: CALL k {,s}

Operands: $0 \leq k \leq 1048575$
 $s \in [0,1]$

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:

| | | | |
|------|-------------|----------|----------|
| 1110 | 110s | k_7kkk | $kkkk_0$ |
| 1111 | $k_{19}kkk$ | $kkkk$ | $kkkk_8$ |

1st word ($k<7:0>$)
 2nd word ($k<19:8>$)

Description: Subroutine call of entire 2-Mbyte memory range. First, return address ($PC + 4$) is pushed onto the return stack. If 's' = 1, the W, Status and BSR registers are also pushed into their respective shadow registers, WS, STATUSS and BSRS. If 's' = 0, no update occurs (default). Then, the 20-bit value 'k' is loaded into $PC<20:1>$. CALL is a two-cycle instruction.

Words: 2

Cycles: 2

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--|------------------|--|
| Decode | Read literal 'k'<7:0>, Push PC to stack | Push PC to stack | Read literal 'k'<19:8>, Write to PC |
| No operation | No operation | No operation | No operation |

Example: HERE CALL THERE, 1

Before Instruction
 PC = address (HERE)

After Instruction
 PC = address (THERE)
 TOS = address (HERE + 4)
 WS = W
 BSRS = BSR
 STATUSS = Status

PIC18F2455/2550/4455/4550

CLRF **Clear f**

Syntax: CLRF f{,a}

Operands: $0 \leq f \leq 255$
 $a \in [0,1]$

Operation: $000h \rightarrow f$
 $1 \rightarrow Z$

Status Affected: Z

Encoding:

| | | | |
|------|------|------|------|
| 0110 | 101a | ffff | ffff |
|------|------|------|------|

Description: 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 $f \leq 95$ (5Fh). See **Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|-------------------|--------------|--------------------|
| Decode | Read register 'f' | Process Data | Write register 'f' |

Example: CLRF FLAG_REG, 1

Before Instruction
 FLAG_REG = 5Ah

After Instruction
 FLAG_REG = 00h

CLRWDT **Clear Watchdog Timer**

Syntax: CLRWDT

Operands: None

Operation: $000h \rightarrow$ WDT,
 $000h \rightarrow$ WDT postscaler,
 $1 \rightarrow \overline{TO}$,
 $1 \rightarrow \overline{PD}$

Status Affected: \overline{TO} , \overline{PD}

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 0000 | 0000 | 0100 |
|------|------|------|------|

Description: CLRWDT instruction resets the Watchdog Timer. It also resets the postscaler of the WDT. Status bits, \overline{TO} and \overline{PD} , are set.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|--------------|--------------|--------------|
| Decode | No operation | Process Data | No operation |

Example: CLRWDT

Before Instruction
 WDT Counter = ?

After Instruction
 WDT Counter = 00h
 WDT Postscaler = 0
 \overline{TO} = 1
 \overline{PD} = 1

PIC18F2455/2550/4455/4550

COMF Complement f

Syntax: COMF f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: $(\bar{f}) \rightarrow \text{dest}$

Status Affected: N, Z

Encoding:

| | | | |
|------|------|------|------|
| 0001 | 11da | ffff | ffff |
|------|------|------|------|

Description: The contents of register 'f' are complemented. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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: COMF REG, 0, 0

Before Instruction
 REG = 13h

After Instruction
 REG = 13h
 W = ECh

CPFSEQ Compare f with W, skip if f = W

Syntax: CPFSEQ f {,a}

Operands: $0 \leq f \leq 255$
 $a \in [0,1]$

Operation: $(f) - (W)$,
 skip if $(f) = (W)$
 (unsigned comparison)

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0110 | 001a | ffff | ffff |
|------|------|------|------|

Description: Compares the contents of data memory location 'f' to the contents of W by performing an unsigned subtraction. If $f = W$, then the fetched instruction is discarded and a NOP is executed instead, making this a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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 register 'f' | Process Data | No operation |

If skip:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |

If skip and followed by 2-word instruction:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |
| No operation | No operation | No operation | No operation |

Example: HERE CPFSEQ REG, 0
 NEQUAL :
 EQUAL :

Before Instruction
 PC Address = HERE
 W = ?
 REG = ?

After Instruction
 If REG = W;
 PC = Address (EQUAL)
 If REG \neq W;
 PC = Address (NEQUAL)

PIC18F2455/2550/4455/4550

CPFSGT Compare f with W, skip if f > W

Syntax: CPFSGT f {,a}
 Operands: $0 \leq f \leq 255$
 $a \in [0,1]$
 Operation: $(f) - (W)$,
 skip if $(f) > (W)$
 (unsigned comparison)
 Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0110 | 010a | ffff | ffff |
|------|------|------|------|

Description: Compares the contents of data memory location 'f' to the contents of the W by performing an unsigned subtraction. If the contents of 'f' are greater than the contents of WREG, then the fetched instruction is discarded and a NOP is executed instead, making this a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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 register 'f' | Process Data | No operation |

If skip:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |

If skip and followed by 2-word instruction:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |
| No operation | No operation | No operation | No operation |

Example: HERE CPFSGT REG, 0
 NGREATER :
 GREATER :

Before Instruction
 PC = Address (HERE)
 W = ?
 After Instruction
 If REG > W;
 PC = Address (GREATER)
 If REG ≤ W;
 PC = Address (NGREATER)

CPFSLT Compare f with W, skip if f < W

Syntax: CPFSLT f {,a}
 Operands: $0 \leq f \leq 255$
 $a \in [0,1]$
 Operation: $(f) - (W)$,
 skip if $(f) < (W)$
 (unsigned comparison)
 Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0110 | 000a | ffff | ffff |
|------|------|------|------|

Description: Compares the contents of data memory location 'f' to the contents of W by performing an unsigned subtraction. If the contents of 'f' are less than the contents of W, then the fetched instruction is discarded and a NOP is executed instead, making this a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

Words: 1
 Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|-------------------|--------------|--------------|
| Decode | Read register 'f' | Process Data | No operation |

If skip:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |

If skip and followed by 2-word instruction:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |
| No operation | No operation | No operation | No operation |

Example: HERE CPFSLT REG, 1
 NLESS :
 LESS :

Before Instruction
 PC = Address (HERE)
 W = ?
 After Instruction
 If REG < W;
 PC = Address (LESS)
 If REG ≥ W;
 PC = Address (NLESS)

PIC18F2455/2550/4455/4550

DAW **Decimal Adjust W Register**

Syntax: DAW

Operands: None

Operation: If [W<3:0> > 9] or [DC = 1] then
(W<3:0>) + 6 → W<3:0>;
else
(W<3:0>) → W<3:0>;

 If [W<7:4> + DC > 9] or [C = 1] then
(W<7:4>) + 6 + DC → W<7:4>;
else
(W<7:4>) + DC → W<7:4>

Status Affected: C

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 0000 | 0000 | 0111 |
|------|------|------|------|

Description: DAW adjusts the eight-bit value in W, resulting from the earlier addition of two variables (each in packed BCD format) and produces a correct packed BCD result.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|-----------------|--------------|---------|
| Decode | Read register W | Process Data | Write W |

Example 1:

DAW

Before Instruction

W = A5h
C = 0
DC = 0

After Instruction

W = 05h
C = 1
DC = 0

Example 2:

Before Instruction

W = CEh
C = 0
DC = 0

After Instruction

W = 34h
C = 1
DC = 0

DECF **Decrement f**

Syntax: DECF f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: $(f) - 1 \rightarrow \text{dest}$

Status Affected: C, DC, N, OV, Z

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 01da | ffff | ffff |
|------|------|------|------|

Description: Decrement register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).
If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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:

DECF CNT, 1, 0

Before Instruction

CNT = 01h
Z = 0

After Instruction

CNT = 00h
Z = 1

PIC18F2455/2550/4455/4550

DECFSZ Decrement f, skip if 0

Syntax: DECFSZ f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: $(f) - 1 \rightarrow \text{dest}$,
skip if result = 0

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0010 | 11da | ffff | ffff |
|------|------|------|------|

Description: The contents of register 'f' are decremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). If the result is '0', the next instruction, which is already fetched, is discarded and a NOP is executed instead, making it a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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 register 'f' | Process Data | Write to destination |

If skip:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |

If skip and followed by 2-word instruction:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |
| No operation | No operation | No operation | No operation |

Example: HERE DECFSZ CNT, 1, 1
 GOTO LOOP
 CONTINUE

Before Instruction
PC = Address (HERE)
After Instruction
CNT = CNT - 1
If CNT = 0;
PC = Address (CONTINUE)
If CNT \neq 0;
PC = Address (HERE + 2)

DCFSNZ Decrement f, skip if not 0

Syntax: DCFSNZ f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: $(f) - 1 \rightarrow \text{dest}$,
skip if result \neq 0

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0100 | 11da | ffff | ffff |
|------|------|------|------|

Description: The contents of register 'f' are decremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). If the result is not '0', the next instruction, which is already fetched, is discarded and a NOP is executed instead, making it a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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 register 'f' | Process Data | Write to destination |

If skip:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |

If skip and followed by 2-word instruction:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |
| No operation | No operation | No operation | No operation |

Example: HERE DCFSNZ TEMP, 1, 0
 ZERO :
 NZERO :

Before Instruction
TEMP = ?
After Instruction
TEMP = TEMP - 1,
If TEMP = 0;
PC = Address (ZERO)
If TEMP \neq 0;
PC = Address (NZERO)

PIC18F2455/2550/4455/4550

GOTO Unconditional Branch

Syntax: GOTO k

Operands: $0 \leq k \leq 1048575$

Operation: $k \rightarrow PC<20:1>$

Status Affected: None

Encoding:

1st word ($k<7:0>$)

2nd word ($k<19:8>$)

| | | | |
|------|-------------|----------|----------|
| 1110 | 1111 | k_7kkk | $kkkk_0$ |
| 1111 | $k_{19}kkk$ | $kkkk$ | $kkkk_8$ |

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

Syntax: INCF f {,d {,a}}

Operands: $0 \leq f \leq 255$

$d \in [0,1]$

$a \in [0,1]$

Operation: $(f) + 1 \rightarrow \text{dest}$

Status Affected: C, DC, N, OV, Z

Encoding:

| | | | |
|------|------|------|------|
| 0010 | 10da | ffff | ffff |
|------|------|------|------|

Description: The contents of register 'f' are incremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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: INCF CNT, 1, 0

Before Instruction

CNT = FFh
Z = 0
C = ?
DC = ?

After Instruction

CNT = 00h
Z = 1
C = 1
DC = 1

PIC18F2455/2550/4455/4550

INCFSZ Increment f, skip if 0

Syntax: INCFSZ f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: $(f) + 1 \rightarrow \text{dest}$,
 skip if result = 0

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0011 | 11da | ffff | ffff |
|------|------|------|------|

Description: The contents of register 'f' are incremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f'. (default)
 If the result is '0', the next instruction, which is already fetched, is discarded and a NOP is executed instead, making it a two-cycle instruction.
 If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).
 If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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 register 'f' | Process Data | Write to destination |

If skip:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |

If skip and followed by 2-word instruction:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |
| No operation | No operation | No operation | No operation |

Example: HERE INCFSZ CNT, 1, 0
 NZERO :
 ZERO :

Before Instruction
 PC = Address (HERE)
 After Instruction
 CNT = CNT + 1
 If CNT = 0;
 PC = Address (ZERO)
 If CNT \neq 0;
 PC = Address (NZERO)

INFSNZ Increment f, skip if not 0

Syntax: INFSNZ f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: $(f) + 1 \rightarrow \text{dest}$,
 skip if result $\neq 0$

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0100 | 10da | ffff | ffff |
|------|------|------|------|

Description: The contents of register 'f' are incremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).
 If the result is not '0', the next instruction, which is already fetched, is discarded and a NOP is executed instead, making it a two-cycle instruction.
 If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).
 If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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 register 'f' | Process Data | Write to destination |

If skip:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |

If skip and followed by 2-word instruction:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |
| No operation | No operation | No operation | No operation |

Example: HERE INFSNZ REG, 1, 0
 ZERO :
 NZERO :

Before Instruction
 PC = Address (HERE)
 After Instruction
 REG = REG + 1
 If REG \neq 0;
 PC = Address (NZERO)
 If REG = 0;
 PC = Address (ZERO)

PIC18F2455/2550/4455/4550

IORLW Inclusive OR literal with W

Syntax: IORLW k

Operands: $0 \leq k \leq 255$

Operation: (W) .OR. k \rightarrow W

Status Affected: N, Z

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 1001 | kkkk | kkkk |
|------|------|------|------|

Description: The contents of W are ORed with the eight-bit literal 'k'. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|---------------------|-----------------|------------|
| Decode | Read literal 'k' | Process Data | Write to W |

Example: IORLW 35h

Before Instruction

W = 9Ah

After Instruction

W = BFh

IORWF Inclusive OR W with f

Syntax: IORWF f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: (W) .OR. (f) \rightarrow dest

Status Affected: N, Z

Encoding:

| | | | |
|------|------|------|------|
| 0001 | 00da | ffff | ffff |
|------|------|------|------|

Description: Inclusive OR W with register 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).
 If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).
 If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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: IORWF RESULT, 0, 1

Before Instruction

RESULT = 13h

W = 91h

After Instruction

RESULT = 13h

W = 93h

PIC18F2455/2550/4455/4550

LFSR Load FSR

Syntax: LFSR f, k

Operands: $0 \leq f \leq 2$
 $0 \leq k \leq 4095$

Operation: $k \rightarrow \text{FSRf}$

Status Affected: None

Encoding:

| | | | |
|------|------|--------------------|---------------------|
| 1110 | 1110 | 00ff | k ₁₁ kkk |
| 1111 | 0000 | k ₇ 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 'k' MSB | Process Data | Write literal 'k' MSB to FSRfH |
| Decode | Read literal 'k' LSB | Process Data | Write literal 'k' to FSRfL |

Example: LFSR 2, 3ABh

After Instruction

FSR2H = 03h
 FSR2L = ABh

MOVF Move f

Syntax: MOVF f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: $f \rightarrow \text{dest}$

Status Affected: N, Z

Encoding:

| | | | |
|------|------|------|------|
| 0101 | 00da | ffff | ffff |
|------|------|------|------|

Description: The contents of register 'f' are moved to a destination dependent upon the status of 'd'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). Location 'f' can be anywhere in the 256-byte bank.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|-------------------|--------------|---------|
| Decode | Read register 'f' | Process Data | Write W |

Example: MOVF REG, 0, 0

Before Instruction

REG = 22h
 W = FFh

After Instruction

REG = 22h
 W = 22h

PIC18F2455/2550/4455/4550

MOVFF Move f to f

Syntax: MOVFF f_s, f_d

Operands: $0 \leq f_s \leq 4095$
 $0 \leq f_d \leq 4095$

Operation: $(f_s) \rightarrow f_d$

Status Affected: None

Encoding:

1st word (source)

2nd word (destin.)

| | | | |
|------|------|------|-------------------|
| 1100 | ffff | ffff | ffff _s |
| 1111 | ffff | ffff | ffff _d |

Description: The contents of source register ' f_s ' are moved to destination register ' f_d '. Location of source ' f_s ' can be anywhere in the 4096-byte data space (000h to FFFh) and location of destination ' f_d ' can also be anywhere from 000h to FFFh.

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 (3)

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

REG1 = 33h

REG2 = 11h

After Instruction

REG1 = 33h

REG2 = 33h

MOVLB Move literal to low nibble in BSR

Syntax: MOVLW k

Operands: $0 \leq k \leq 255$

Operation: $k \rightarrow \text{BSR}$

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 0001 | kkkk | kkkk |
|------|------|------|------|

Description:

The eight-bit literal 'k' is loaded into the Bank Select Register (BSR). The value of BSR<7:4> always remains '0', regardless of the value of $k_7:k_4$.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|------------------|--------------|--------------------------|
| Decode | Read literal 'k' | Process Data | Write literal 'k' to BSR |

Example: MOVLB 5

Before Instruction

BSR Register = 02h

After Instruction

BSR Register = 05h

PIC18F2455/2550/4455/4550

MOVLW Move literal to W

Syntax: MOVLW k

Operands: $0 \leq k \leq 255$

Operation: $k \rightarrow W$

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 1110 | kkkk | kkkk |
|------|------|------|------|

Description: The eight-bit literal 'k' is loaded into W.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|---------------------|-----------------|------------|
| Decode | Read literal 'k' | Process Data | Write to W |

Example: MOVLW 5Ah

After Instruction

W = 5Ah

MOVWF Move W to f

Syntax: MOVWF f{,a}

Operands: $0 \leq f \leq 255$
 $a \in [0,1]$

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 $f \leq 95$ (5Fh). See **Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|----------------------|-----------------|-----------------------|
| Decode | Read register 'f' | Process Data | Write register 'f' |

Example: MOVWF REG, 0

Before Instruction

W = 4Fh
REG = FFh

After Instruction

W = 4Fh
REG = 4Fh

PIC18F2455/2550/4455/4550

MULLW Multiply literal with W

Syntax: MULLW k

Operands: $0 \leq k \leq 255$

Operation: $(W) \times k \rightarrow \text{PRODH}:\text{PRODL}$

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 1101 | kkkk | kkkk |
|------|------|------|------|

Description: An unsigned multiplication is carried out between the contents of W and the 8-bit literal 'k'. The 16-bit result is placed in PRODH:PRODL register pair. PRODH contains the high byte. W is unchanged. None of the Status flags are affected. Note that neither Overflow nor Carry is possible in this operation. A zero result is possible but not detected.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|------------------|--------------|------------------------------|
| Decode | Read literal 'k' | Process Data | Write registers PRODH: PRODL |

Example: MULLW 0C4h

Before Instruction

W = E2h

PRODH = ?

PRODL = ?

After Instruction

W = E2h

PRODH = ADh

PRODL = 08h

MULWF Multiply W with f

Syntax: MULWF f {,a}

Operands: $0 \leq f \leq 255$
 $a \in [0,1]$

Operation: $(W) \times (f) \rightarrow \text{PRODH}:\text{PRODL}$

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 001a | ffff | ffff |
|------|------|------|------|

Description: An unsigned multiplication is carried out between the contents of W and the register file location 'f'. The 16-bit result is stored in the PRODH:PRODL register pair. PRODH contains the high byte. Both W and 'f' are unchanged. None of the Status flags are affected. Note that neither Overflow nor Carry is possible in this operation. A zero result is possible but not detected. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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 registers PRODH: PRODL |

Example: MULWF REG, 1

Before Instruction

W = C4h

REG = B5h

PRODH = ?

PRODL = ?

After Instruction

W = C4h

REG = B5h

PRODH = 8Ah

PRODL = 94h

PIC18F2455/2550/4455/4550

NEGF

Negate f

Syntax: `NEGF f{,a}`

Operands: $0 \leq f \leq 255$
 $a \in [0,1]$

Operation: $(\bar{f}) + 1 \rightarrow f$

Status Affected: N, OV, C, DC, Z

Encoding:

| | | | |
|------|------|------|------|
| 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 $f \leq 95$ (5Fh). See **Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|-------------------|--------------|--------------------|
| Decode | Read register 'f' | Process Data | Write register 'f' |

Example: `NEGF REG, 1`

Before Instruction

REG = 0011 1010 [3Ah]

After Instruction

REG = 1100 0110 [C6h]

NOP

No Operation

Syntax: `NOP`

Operands: None

Operation: No operation

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 0000 | 0000 | 0000 |
| 1111 | xxxx | xxxx | xxxx |

Description: No operation.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|--------------|--------------|--------------|
| Decode | No operation | No operation | No operation |

Example:

None.

PIC18F2455/2550/4455/4550

POP Pop Top of Return Stack

Syntax: POP

Operands: None

Operation: (TOS) → bit 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 then becomes the previous value that was pushed onto the return stack. This instruction is provided to enable the user to properly manage the return stack to incorporate a software stack.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|--------------|---------------|--------------|
| Decode | No operation | Pop TOS value | No operation |

Example:

| | | |
|----------------------|------|---------|
| | POP | |
| | GOTO | NEW |
| Before Instruction | | |
| TOS | = | 0031A2h |
| Stack (1 level down) | = | 014332h |
| After Instruction | | |
| TOS | = | 014332h |
| PC | = | NEW |

PUSH Push Top of Return Stack

Syntax: PUSH

Operands: None

Operation: (PC + 2) → TOS

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 0000 | 0000 | 0101 |
|------|------|------|------|

Description: The PC + 2 is pushed onto the top of the return stack. The previous TOS value is pushed down on the stack. This instruction allows implementing a software stack by modifying TOS and then pushing it onto the return stack.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|-------------------------------|--------------|--------------|
| Decode | Push PC + 2 onto return stack | No operation | No operation |

Example:

| | | |
|----------------------|---|-------|
| Before Instruction | | |
| TOS | = | 345Ah |
| PC | = | 0124h |
| After Instruction | | |
| PC | = | 0126h |
| TOS | = | 0126h |
| Stack (1 level down) | = | 345Ah |

PIC18F2455/2550/4455/4550

RCALL Relative Call

Syntax: RCALL n

Operands: $-1024 \leq n \leq 1023$

Operation: (PC) + 2 → TOS,
(PC) + 2 + 2n → 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

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------------------------------|--------------|--------------|
| Decode | Read literal 'n' Push PC to stack | Process Data | Write to PC |
| No operation | No operation | No operation | No operation |

Example: HERE RCALL Jump

Before Instruction

PC = Address (HERE)

After Instruction

PC = Address (Jump)

TOS = Address (HERE + 2)

RESET Reset

Syntax: RESET

Operands: None

Operation: Reset all registers and flags that are affected by a MCLR Reset.

Status Affected: All

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 0000 | 1111 | 1111 |
|------|------|------|------|

Description: This instruction provides a way to execute a MCLR Reset in software.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|-------------|--------------|--------------|
| Decode | Start Reset | No operation | No operation |

Example: RESET

After Instruction

Registers = Reset Value

Flags* = Reset Value

PIC18F2455/2550/4455/4550

RETFIE Return from Interrupt

Syntax: RETFIE {s}

Operands: s ∈ [0,1]

Operation: (TOS) → PC,
1 → GIE/GIEH or PEIE/GIEL,
if s = 1
(WS) → W,
(STATUS) → Status,
(BSRS) → BSR,
PCLATU, PCLATH are unchanged.

Status Affected: GIE/GIEH, PEIE/GIEL.

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 0000 | 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, STATUS 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 operation | No operation | No operation | No operation |

Example: RETFIE 1

After Interrupt

```

PC          = TOS
W           = WS
BSR        = BSRS
Status     = STATUS
GIE/GIEH, PEIE/GIEL = 1
    
```

RETLW Return literal to W

Syntax: RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → W,
(TOS) → PC,
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 literal 'k' | Process Data | Pop PC from stack, Write to W |
| No operation | No operation | No operation | No 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 ;
:
:
  RETLW kn ; End of table
    
```

Before Instruction

W = 07h

After Instruction

W = value of kn

PIC18F2455/2550/4455/4550

RETURN Return from Subroutine

Syntax: RETURN {s}

Operands: s ∈ [0,1]

Operation: (TOS) → PC,
if s = 1
(WS) → W,
(STATUS) → Status,
(BSRS) → BSR,
PCLATU, PCLATH are unchanged

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 0000 | 0001 | 001s |
|------|------|------|------|

Description: Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter. If 's' = 1, the contents of the shadow registers WS, STATUS and BSRS are loaded into their corresponding registers, W, Status and BSR. If 's' = 0, no update of these registers occurs (default).

Words: 1

Cycles: 2

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|-------------------|
| Decode | No operation | Process Data | Pop PC from stack |
| No operation | No operation | No operation | No operation |

Example: RETURN

After Instruction:
PC = TOS

RLCF Rotate Left f through Carry

Syntax: RLCF f {,d {,a}}

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

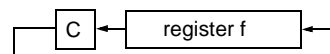
Operation: (f<n>) → dest<n + 1>,
(f<7>) → C,
(C) → dest<0>

Status Affected: C, N, Z

Encoding:

| | | | |
|------|------|------|------|
| 0011 | 01da | ffff | ffff |
|------|------|------|------|

Description: The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 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: RLCF REG, 0, 0

Before Instruction
REG = 1110 0110
C = 0

After Instruction
REG = 1110 0110
W = 1100 1100
C = 1

PIC18F2455/2550/4455/4550

RLNCF Rotate Left f (no carry)

Syntax: RLNCF f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: $(f\langle n \rangle) \rightarrow \text{dest}\langle n + 1 \rangle$,
 $(f\langle 7 \rangle) \rightarrow \text{dest}\langle 0 \rangle$

Status Affected: N, Z

Encoding:

| | | | |
|------|------|------|------|
| 0100 | 01da | ffff | ffff |
|------|------|------|------|

Description: The contents of register 'f' are rotated one bit to the left. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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: RLNCF REG, 1, 0

Before Instruction
REG = 1010 1011
After Instruction
REG = 0101 0111

RRCF Rotate Right f through Carry

Syntax: RRCF f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

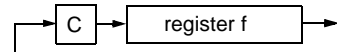
Operation: $(f\langle n \rangle) \rightarrow \text{dest}\langle n - 1 \rangle$,
 $(f\langle 0 \rangle) \rightarrow C$,
 $(C) \rightarrow \text{dest}\langle 7 \rangle$

Status Affected: C, N, Z

Encoding:

| | | | |
|------|------|------|------|
| 0011 | 00da | ffff | ffff |
|------|------|------|------|

Description: The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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: RRCF REG, 0, 0

Before Instruction
REG = 1110 0110
C = 0
After Instruction
REG = 1110 0110
W = 0111 0011
C = 0

PIC18F2455/2550/4455/4550

RRNCF Rotate Right f (no carry)

Syntax: RRNCF f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: $(f \langle n \rangle) \rightarrow \text{dest} \langle n - 1 \rangle$,
 $(f \langle 0 \rangle) \rightarrow \text{dest} \langle 7 \rangle$

Status Affected: N, Z

Encoding:

| | | | |
|------|------|------|------|
| 0100 | 00da | ffff | ffff |
|------|------|------|------|

Description: 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 Offsetting mode whenever $f \leq 95$ (5Fh). See **Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offsetting 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: RRNCF REG, 1, 0

Before Instruction
 REG = 1101 0111
 After Instruction
 REG = 1110 1011

Example 2: RRNCF REG, 0, 0

Before Instruction
 W = ?
 REG = 1101 0111
 After Instruction
 W = 1110 1011
 REG = 1101 0111

SETF Set f

Syntax: SETF f {,a}

Operands: $0 \leq f \leq 255$
 $a \in [0,1]$

Operation: FFh \rightarrow f

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0110 | 100a | ffff | ffff |
|------|------|------|------|

Description: The contents of the specified register are set to FFh. 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 Offsetting mode whenever $f \leq 95$ (5Fh). See **Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offsetting Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|-------------------|--------------|--------------------|
| Decode | Read register 'f' | Process Data | Write register 'f' |

Example: SETF REG, 1

Before Instruction
 REG = 5Ah
 After Instruction
 REG = FFh

PIC18F2455/2550/4455/4550

SLEEP Enter Sleep mode

Syntax: SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT postscaler,
1 → \overline{TO} ,
0 → PD

Status Affected: \overline{TO} , \overline{PD}

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 0000 | 0000 | 0011 |
|------|------|------|------|

Description: 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.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|--------------|--------------|-------------|
| Decode | No operation | Process Data | Go to Sleep |

Example: SLEEP

Before Instruction

\overline{TO} = ?
 \overline{PD} = ?

After Instruction

\overline{TO} = 1 †
 \overline{PD} = 0

† If WDT causes wake-up, this bit is cleared.

SUBFWB Subtract f from W with borrow

Syntax: SUBFWB f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: $(W) - (f) - (\overline{C}) \rightarrow \text{dest}$

Status Affected: N, OV, C, DC, Z

Encoding:

| | | | |
|------|------|------|------|
| 0101 | 01da | ffff | ffff |
|------|------|------|------|

Description: Subtract register 'f' and Carry flag (borrow) from W (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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 Instruction

REG = 3
W = 2
C = 1

After Instruction

REG = FF
W = 2
C = 0
Z = 0
N = 1 ; result is negative

Example 2: SUBFWB REG, 0, 0

Before Instruction

REG = 2
W = 5
C = 1

After Instruction

REG = 2
W = 3
C = 1
Z = 0
N = 0 ; result is positive

Example 3: SUBFWB REG, 1, 0

Before Instruction

REG = 1
W = 2
C = 0

After Instruction

REG = 0
W = 2
C = 1
Z = 1 ; result is zero
N = 0

PIC18F2455/2550/4455/4550

SUBLW Subtract W from literal

Syntax: SUBLW k

Operands: $0 \leq k \leq 255$

Operation: $k - (W) \rightarrow W$

Status Affected: N, OV, C, DC, Z

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 1000 | kkkk | kkkk |
|------|------|------|------|

Description: W is subtracted from the eight-bit literal 'k'. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|------------------|--------------|------------|
| Decode | Read literal 'k' | Process Data | Write to W |

Example 1: SUBLW 02h

Before Instruction
W = 01h
C = ?

After Instruction
W = 01h
C = 1 ; result is positive
Z = 0
N = 0

Example 2: SUBLW 02h

Before Instruction
W = 02h
C = ?

After Instruction
W = 00h
C = 1 ; result is zero
Z = 1
N = 0

Example 3: SUBLW 02h

Before Instruction
W = 03h
C = ?

After Instruction
W = FFh ; (2's complement)
C = 0 ; result is negative
Z = 0
N = 1

SUBWF Subtract W from f

Syntax: SUBWF f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: $(f) - (W) \rightarrow \text{dest}$

Status Affected: N, OV, C, DC, Z

Encoding:

| | | | |
|------|------|------|------|
| 0101 | 11da | ffff | ffff |
|------|------|------|------|

Description: Subtract W from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).
If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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 |

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|-------------------|--------------|----------------------|
| Decode | Read register 'f' | Process Data | Write to destination |

Example 1: SUBWF REG, 1, 0

Before Instruction
REG = 3
W = 2
C = ?

After Instruction
REG = 1
W = 2
C = 1 ; result is positive
Z = 0
N = 0

Example 2: SUBWF REG, 0, 0

Before Instruction
REG = 2
W = 2
C = ?

After Instruction
REG = 2
W = 0
C = 1 ; result is zero
Z = 1
N = 0

Example 3: SUBWF REG, 1, 0

Before Instruction
REG = 1
W = 2
C = ?

After Instruction
REG = FFh ; (2's complement)
W = 2
C = 0 ; result is negative
Z = 0
N = 1

PIC18F2455/2550/4455/4550

SUBWFB Subtract W from f with Borrow

Syntax: SUBWFB f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: $(f) - (W) - (\overline{C}) \rightarrow \text{dest}$

Status Affected: N, OV, C, DC, Z

Encoding:

| | | | |
|------|------|------|------|
| 0101 | 10da | ffff | ffff |
|------|------|------|------|

Description: Subtract W and the Carry flag (borrow) from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1
 Cycles: 1

Q Cycle Activity:

| | Q1 | Q2 | Q3 | Q4 |
|--------|----|-------------------|--------------|----------------------|
| Decode | | Read register 'f' | Process Data | Write to destination |

Example 1: SUBWFB REG, 1, 0

Before Instruction
 REG = 19h (0001 1001)
 W = 0Dh (0000 1101)
 C = 1

After Instruction
 REG = 0Ch (0000 1011)
 W = 0Dh (0000 1101)
 C = 1
 Z = 0
 N = 0 ; result is positive

Example 2: SUBWFB REG, 0, 0

Before Instruction
 REG = 1Bh (0001 1011)
 W = 1Ah (0001 1010)
 C = 0

After Instruction
 REG = 1Bh (0001 1011)
 W = 00h
 C = 1
 Z = 1 ; result is zero
 N = 0

Example 3: SUBWFB REG, 1, 0

Before Instruction
 REG = 03h (0000 0011)
 W = 0Eh (0000 1101)
 C = 1

After Instruction
 REG = F5h (1111 0100)
 ; [2's comp]
 W = 0Eh (0000 1101)
 C = 0
 Z = 0
 N = 1 ; result is negative

SWAPF Swap f

Syntax: SWAPF f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: $(f<3:0>) \rightarrow \text{dest}<7:4>$,
 $(f<7:4>) \rightarrow \text{dest}<3:0>$

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0011 | 10da | ffff | ffff |
|------|------|------|------|

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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: SWAPF REG, 1, 0

Before Instruction
 REG = 53h

After Instruction
 REG = 35h

PIC18F2455/2550/4455/4550

TBLRD Table Read

Syntax: TBLRD (*; *+; *-; +*)

Operands: None

Operation: if TBLRD *,
(Prog Mem (TBLPTR)) → TABLAT;
TBLPTR – No Change;
if TBLRD *+,
(Prog Mem (TBLPTR)) → TABLAT;
(TBLPTR) + 1 → TBLPTR;
if TBLRD *-,
(Prog Mem (TBLPTR)) → TABLAT;
(TBLPTR) – 1 → TBLPTR;
if TBLRD +*,
(TBLPTR) + 1 → TBLPTR;
(Prog Mem (TBLPTR)) → TABLAT;

Status Affected: None

| | | | | |
|-----------|------|------|------|---|
| Encoding: | 0000 | 0000 | 0000 | 10nn nn=0 * =1 *+ =2 *- =3 +* |
|-----------|------|------|------|---|

Description: 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:

- no change
- post-increment
- post-decrement
- pre-increment

Words: 1

Cycles: 2

Q Cycle Activity:

| | Q1 | Q2 | Q3 | Q4 |
|--------------|------------------------------------|--------------|-----------------------------|--------------|
| Decode | No operation | No operation | No operation | No operation |
| No operation | No operation (Read Program Memory) | No operation | No operation (Write TABLAT) | |

TBLRD Table Read (Continued)

Example 1: TBLRD *+ ;

Before Instruction

| | | |
|------------------|---|---------|
| TABLAT | = | 55h |
| TBLPTR | = | 00A356h |
| MEMORY (00A356h) | = | 34h |

After Instruction

| | | |
|--------|---|---------|
| TABLAT | = | 34h |
| TBLPTR | = | 00A357h |

Example 2: TBLRD +* ;

Before Instruction

| | | |
|------------------|---|---------|
| TABLAT | = | AAh |
| TBLPTR | = | 01A357h |
| MEMORY (01A357h) | = | 12h |
| MEMORY (01A358h) | = | 34h |

After Instruction

| | | |
|--------|---|---------|
| TABLAT | = | 34h |
| TBLPTR | = | 01A358h |

PIC18F2455/2550/4455/4550

TBLWT Table Write

Syntax: TBLWT (*, *+, *-, +*)
 Operands: None
 Operation: if TBLWT*,
 (TABLAT) → Holding Register;
 TBLPTR – No Change;
 if TBLWT*+,
 (TABLAT) → Holding Register;
 (TBLPTR) + 1 → TBLPTR;
 if TBLWT*-,
 (TABLAT) → Holding Register;
 (TBLPTR) – 1 → TBLPTR;
 if TBLWT+*,
 (TBLPTR) + 1 → TBLPTR;
 (TABLAT) → Holding Register;

Status Affected: None

Encoding:

| | | | |
|------|------|------|---|
| 0000 | 0000 | 0000 | 11nn nn=0 * =1 *+ =2 *- =3 *+ |
|------|------|------|---|

Description: This instruction uses the 3 LSBs of TBLPTR to determine which of the 8 holding registers the TABLAT is written to. The holding registers are used to program the contents of Program Memory (P.M.). (Refer to **Section 6.0 “Flash Program Memory”** for additional details on programming Flash memory.) The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range. The LSB of the TBLPTR selects which byte of the program memory location to access.

TBLPTR[0] = 0: Least Significant Byte of Program Memory Word
 TBLPTR[0] = 1: Most Significant Byte of Program Memory Word

The TBLWT instruction can modify the value of TBLPTR as follows:

- no change
- post-increment
- post-decrement
- pre-increment

Words: 1
 Cycles: 2
 Q Cycle Activity:

| | Q1 | Q2 | Q3 | Q4 |
|--------------|----------------------------|--------------|--|--------------|
| Decode | No operation | No operation | No operation | No operation |
| No operation | No operation (Read TABLAT) | No operation | No operation (Write to Holding Register) | No operation |

TBLWT Table Write (Continued)

Example 1: TBLWT *+;

Before Instruction
 TABLAT = 55h
 TBLPTR = 00A356h
 HOLDING REGISTER (00A356h) = FFh

After Instructions (table write completion)
 TABLAT = 55h
 TBLPTR = 00A357h
 HOLDING REGISTER (00A356h) = 55h

Example 2: TBLWT *+;

Before Instruction
 TABLAT = 34h
 TBLPTR = 01389Ah
 HOLDING REGISTER (01389Ah) = FFh
 HOLDING REGISTER (01389Bh) = FFh

After Instruction (table write completion)
 TABLAT = 34h
 TBLPTR = 01389Bh
 HOLDING REGISTER (01389Ah) = FFh
 HOLDING REGISTER (01389Bh) = 34h

PIC18F2455/2550/4455/4550

TSTFSZ Test f, skip if 0

Syntax: TSTFSZ f {,a}

Operands: $0 \leq f \leq 255$
 $a \in [0,1]$

Operation: skip if $f = 0$

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0110 | 011a | ffff | ffff |
|------|------|------|------|

Description: If 'f' = 0, the next instruction fetched during the current instruction execution is discarded and a NOP is executed, making this a two-cycle instruction. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default). If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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 register 'f' | Process Data | No operation |

If skip:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |

If skip and followed by 2-word instruction:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|--------------|--------------|
| No operation | No operation | No operation | No operation |
| No operation | No operation | No operation | No operation |

Example:

```

HERE    TSTFSZ  CNT, 1
NZERO   :
ZERO    :
```

Before Instruction
PC = Address (HERE)

After Instruction
If CNT = 00h,
PC = Address (ZERO)
If CNT \neq 00h,
PC = Address (NZERO)

XORLW Exclusive OR literal with W

Syntax: XORLW k

Operands: $0 \leq k \leq 255$

Operation: (W) .XOR. k \rightarrow W

Status Affected: N, Z

Encoding:

| | | | |
|------|------|------|------|
| 0000 | 1010 | kkkk | kkkk |
|------|------|------|------|

Description: The contents of W are XORed with the 8-bit literal 'k'. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|------------------|--------------|------------|
| Decode | Read literal 'k' | Process Data | Write to W |

Example: XORLW 0AFh

Before Instruction
W = B5h

After Instruction
W = 1Ah

PIC18F2455/2550/4455/4550

XORWF Exclusive OR W with f

Syntax: XORWF f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: (W) .XOR. (f) → dest

Status Affected: N, Z

Encoding:

| | | | |
|------|------|------|------|
| 0001 | 10da | ffff | ffff |
|------|------|------|------|

Description: Exclusive OR the contents of W with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in the register 'f' (default).
If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 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: XORWF REG, 1, 0

Before Instruction

REG = AFh
W = B5h

After Instruction

REG = 1Ah
W = B5h

PIC18F2455/2550/4455/4550

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 302) 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. MPASM™ Assembler will flag an error if it determines that an index or offset value is not bracketed.

When the extended instruction set is enabled, brackets are also used to indicate index arguments in byte-oriented and bit-oriented instructions. This is in addition to other changes in their syntax. For more details, see **Section 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, Operands | Description | Cycles | 16-Bit Instruction Word | | | | Status Affected |
|---------------------------------------|---|--------|-------------------------|------|------|------|--------------------|
| | | | MSb | | LSb | | |
| ADDFSR f, k | Add literal to FSR | 1 | 1110 | 1000 | ffkk | kkkk | None |
| ADDULNK k | Add literal to FSR2 and return | 2 | 1110 | 1000 | 11kk | kkkk | None |
| CALLW | Call subroutine using WREG | 2 | 0000 | 0000 | 0001 | 0100 | None |
| MOVSF Z _s , f _d | Move Z _s (source) to 1st word f _d (destination) 2nd word | 2 | 1110 | 1011 | 0zzz | zzzz | None |
| MOVSS Z _s , Z _d | Move Z _s (source) to 1st word Z _d (destination) 2nd word | 2 | 1110 | 1011 | 1zzz | zzzz | None |
| PUSHL k | Store literal at FSR2, decrement FSR2 | 1 | 1110 | 1010 | kkkk | kkkk | None |
| SUBFSR f, k | Subtract literal from FSR | 1 | 1110 | 1001 | ffkk | kkkk | None |
| SUBULNK k | Subtract literal from FSR2 and return | 2 | 1110 | 1001 | 11kk | kkkk | None |

PIC18F2455/2550/4455/4550

26.2.2 EXTENDED INSTRUCTION SET

ADDFSR Add Literal to FSR

Syntax: ADDFSR f, k
 Operands: $0 \leq k \leq 63$
 $f \in [0, 1, 2]$
 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
 Cycles: 1
 Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|------------------|--------------|--------------|
| Decode | Read literal 'k' | Process Data | Write to FSR |

Example: ADDFSR 2, 23h

Before Instruction
 FSR2 = 03FFh
 After Instruction
 FSR2 = 0422h

ADDULNK Add Literal to FSR2 and Return

Syntax: ADDULNK k
 Operands: $0 \leq k \leq 63$
 Operation: $FSR2 + k \rightarrow FSR2,$
 $(TOS) \rightarrow PC$
 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

Words: 1
 Cycles: 2

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------------|------------------|--------------|--------------|
| Decode | Read literal 'k' | Process Data | Write to FSR |
| No Operation | No Operation | No Operation | No Operation |

Example: ADDULNK 23h

Before Instruction
 FSR2 = 03FFh
 PC = 0100h
 After Instruction
 FSR2 = 0422h
 PC = (TOS)

Note: All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction syntax then becomes: {label} instruction argument(s).

PIC18F2455/2550/4455/4550

CALLW Subroutine Call Using WREG

Syntax: CALLW

Operands: None

Operation: (PC + 2) → TOS,
(W) → PCL,
(PCLATH) → PCH,
(PCLATU) → 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.

Words: 1

Cycles: 2

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------------|--------------|------------------|--------------|
| Decode | Read WREG | Push PC to stack | No operation |
| No operation | No operation | No operation | No operation |

Example: HERE CALLW

Before Instruction

PC = address (HERE)
PCLATH = 10h
PCLATU = 00h
W = 06h

After Instruction

PC = 001006h
TOS = address (HERE + 2)
PCLATH = 10h
PCLATU = 00h
W = 06h

MOVSF Move Indexed to f

Syntax: MOVSF [z_s], f_d

Operands: 0 ≤ z_s ≤ 127
0 ≤ f_d ≤ 4095

Operation: ((FSR2) + z_s) → f_d

Status Affected: None

Encoding:

| | | | |
|------|------|------|-------------------|
| 1110 | 1011 | 0zzz | zzzz _s |
| 1111 | ffff | ffff | ffff _d |

Description: The contents of the source register are moved to destination register 'f_d'. The actual address of the source register is determined by adding the 7-bit literal offset 'z_s' in the first word to the value of FSR2. The address of the destination register is specified by the 12-bit literal 'f_d' in the second word. Both addresses can be anywhere in the 4096-byte data space (000h to FFFh). The MOVSF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register. If the resultant source address points to an indirect addressing register, the value returned will be 00h.

Words: 2

Cycles: 2

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|-------------------------------|-----------------------|---------------------------|
| Decode | Determine source addr | Determine source addr | Read source reg |
| Decode | No operation No dummy read | No operation | Write register 'f' (dest) |

Example: MOVSF [05h], REG2

Before Instruction

FSR2 = 80h
Contents of 85h = 33h
REG2 = 11h

After Instruction

FSR2 = 80h
Contents of 85h = 33h
REG2 = 33h

PIC18F2455/2550/4455/4550

MOVSS Move Indexed to Indexed

Syntax: MOVSS [z_s], [z_d]

Operands: 0 ≤ z_s ≤ 127
0 ≤ z_d ≤ 127

Operation: ((FSR2) + z_s) → ((FSR2) + z_d)

Status Affected: None

Encoding:

| | | | |
|------|------|------|-------------------|
| 1110 | 1011 | 1zzz | zzzz _s |
| 1111 | xxxx | xzzz | zzzz _d |

1st word (source)

2nd word (dest.)

Description

The contents of the source register are moved to the destination register. The addresses of the source and destination registers are determined by adding the 7-bit literal offsets 'z_s' or 'z_d', respectively, to the value of FSR2. Both registers can be located anywhere in the 4096-byte data memory space (000h to FFFh).

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 source addr | Determine source addr | Read source reg |
| Decode | Determine dest addr | Determine dest addr | Write 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

PUSHL Store Literal at FSR2, Decrement FSR2

Syntax: PUSHL k

Operands: 0 ≤ k ≤ 255

Operation: k → (FSR2),
FSR2 - 1 → FSR2

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 1111 | 1010 | kkkk | kkkk |
|------|------|------|------|

Description:

The 8-bit literal 'k' is written to the data memory address specified by FSR2. FSR2 is decremented by '1' after the operation. This instruction allows users to push values onto a software stack.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|----------|--------------|----------------------|
| Decode | Read 'k' | Process data | Write to destination |

Example: PUSHL 08h

Before Instruction

FSR2H:FSR2L = 01ECh

Memory (01ECh) = 00h

After Instruction

FSR2H:FSR2L = 01EBh

Memory (01ECh) = 08h

PIC18F2455/2550/4455/4550

SUBFSR Subtract Literal from FSR

Syntax: SUBFSR f, k
 Operands: $0 \leq k \leq 63$
 $f \in [0, 1, 2]$
 Operation: $FSRf - k \rightarrow FSRf$
 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
 Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|-------------------|--------------|----------------------|
| Decode | Read register 'f' | Process Data | Write to destination |

Example: SUBFSR 2, 23h

Before Instruction
 FSR2 = 03FFh
 After Instruction
 FSR2 = 03DCh

SUBULNK Subtract Literal from FSR2 and Return

Syntax: SUBULNK k
 Operands: $0 \leq k \leq 63$
 Operation: $FSR2 - k \rightarrow FSR2$
 $(TOS) \rightarrow PC$
 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
 Cycles: 2
 Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------------|-------------------|--------------|----------------------|
| Decode | Read register 'f' | Process Data | Write to destination |
| No Operation | No Operation | No Operation | No Operation |

Example: SUBULNK 23h

Before Instruction
 FSR2 = 03FFh
 PC = 0100h
 After Instruction
 FSR2 = 03DCh
 PC = (TOS)

PIC18F2455/2550/4455/4550

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 bit-oriented instructions, or almost half of the core PIC18 instructions – may behave differently when the extended instruction set is enabled.

When the content of FSR2 is 00h, the boundaries of the Access RAM are essentially remapped to their original values. This may be useful in creating backward compatible code. If this technique is used, it may be necessary to save the value of FSR2 and restore it when moving back and forth between C and assembly routines in order to preserve the Stack Pointer. Users must also keep in mind the syntax requirements of the extended instruction set (see **Section 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.

PIC18F2455/2550/4455/4550

ADDWF ADD W to Indexed (Indexed Literal Offset mode)

Syntax: ADDWF [k] {,d}

Operands: $0 \leq k \leq 95$
 $d \in [0,1]$

Operation: $(W) + ((FSR2) + k) \rightarrow dest$

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 Data | Write to destination |

Example: ADDWF [OFST] , 0

Before Instruction

W = 17h
 OFST = 2Ch
 FSR2 = 0A00h
 Contents of 0A2Ch = 20h

After Instruction

W = 37h
 Contents of 0A2Ch = 20h

BSF Bit Set Indexed (Indexed Literal Offset mode)

Syntax: BSF [k], b

Operands: $0 \leq f \leq 95$
 $0 \leq b \leq 7$

Operation: $1 \rightarrow ((FSR2) + k) < b >$

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 register 'f' | Process Data | Write to destination |

Example: BSF [FLAG_OFST] , 7

Before Instruction

FLAG_OFST = 0Ah
 FSR2 = 0A00h
 Contents of 0A0Ah = 55h

After Instruction

Contents of 0A0Ah = D5h

SETF Set Indexed (Indexed Literal Offset mode)

Syntax: SETF [k]

Operands: $0 \leq k \leq 95$

Operation: $FFh \rightarrow ((FSR2) + k)$

Status Affected: None

Encoding:

| | | | |
|------|------|------|------|
| 0110 | 1000 | kkkk | kkkk |
|------|------|------|------|

Description: The contents of the register indicated by FSR2, offset by 'k', are set to FFh.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|--------|----------|--------------|----------------|
| Decode | Read 'k' | Process Data | Write register |

Example: SETF [OFST]

Before Instruction

OFST = 2Ch
 FSR2 = 0A00h
 Contents of 0A2Ch = 00h

After Instruction

Contents of 0A2Ch = FFh

PIC18F2455/2550/4455/4550

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® 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 C17 and MPLAB C18 C Compilers
 - MPLINK™ Object Linker/
MPLIB™ Object Librarian
 - MPLAB C30 C Compiler
 - MPLAB ASM30 Assembler/Linker/Library
- Simulators
 - MPLAB SIM Software Simulator
 - MPLAB dsPIC30 Software Simulator
- Emulators
 - MPLAB ICE 2000 In-Circuit Emulator
 - MPLAB ICE 4000 In-Circuit Emulator
- In-Circuit Debugger
 - MPLAB ICD 2
- Device Programmers
 - PRO MATE® II Universal Device Programmer
 - PICSTART® Plus Development Programmer
 - MPLAB PM3 Device Programmer
- Low-Cost Demonstration Boards
 - PICDEM™ 1 Demonstration Board
 - PICDEM.net™ Demonstration Board
 - PICDEM 2 Plus Demonstration Board
 - PICDEM 3 Demonstration Board
 - PICDEM 4 Demonstration Board
 - PICDEM 17 Demonstration Board
 - PICDEM 18R Demonstration Board
 - PICDEM LIN Demonstration Board
 - PICDEM USB Demonstration Board
- Evaluation Kits
 - KEELQ®
 - PICDEM MSC
 - microID®
 - CAN
 - PowerSmart®
 - Analog

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® based application that contains:

- An interface to 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
- Mouse over variable inspection
- Extensive on-line help

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PICmicro 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 increasing 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

PIC18F2455/2550/4455/4550

27.3 MPLAB C17 and MPLAB C18 C Compilers

The MPLAB C17 and MPLAB C18 Code Development Systems are complete ANSI C compilers for Microchip's PIC17CXXX and PIC18CXXX family of microcontrollers. 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 C17 and MPLAB C18 C compilers. 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/librarian 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 C30 C Compiler

The MPLAB C30 C compiler is a full-featured, ANSI compliant, optimizing compiler that translates standard ANSI C programs into dsPIC30F assembly language source. The compiler also supports many command line options and language extensions to take full advantage of the dsPIC30F device hardware capabilities and afford fine control of the compiler code generator.

MPLAB C30 is distributed with a complete ANSI C standard library. All library functions have been validated and conform to the ANSI C library standard. The library includes functions for string manipulation, dynamic memory allocation, data conversion, time-keeping and math functions (trigonometric, exponential and hyperbolic). The compiler provides symbolic information for high-level source debugging with the MPLAB IDE.

27.6 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 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

27.7 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC hosted environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any pin. The execution can be performed in Single-Step, Execute Until Break or Trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and MPLAB C18 C Compilers, as well as the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent, economical software development tool.

27.8 MPLAB SIM30 Software Simulator

The MPLAB SIM30 software simulator allows code development in a PC hosted environment by simulating the dsPIC30F series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any of the pins.

The MPLAB SIM30 simulator fully supports symbolic debugging using the MPLAB C30 C Compiler and MPLAB ASM30 assembler. The simulator runs in either a Command Line mode for automated tasks, or from MPLAB IDE. This high-speed simulator is designed to debug, analyze and optimize time intensive DSP routines.

27.9 MPLAB ICE 2000 High-Performance Universal In-Circuit Emulator

The MPLAB ICE 2000 universal 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 universal architecture of the MPLAB ICE 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.

27.10 MPLAB ICE 4000 High-Performance Universal In-Circuit Emulator

The MPLAB ICE 4000 universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for high-end PICmicro microcontrollers. Software control of the MPLAB ICE 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, up to 2 Mb of emulation memory and the ability to view variables in real-time.

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.

27.11 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 microcontrollers. 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, 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.

27.12 PRO MATE II Universal Device Programmer

The PRO MATE II is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features an LCD display for instructions and error messages and a modular detachable socket assembly to support various package types. In Stand-Alone mode, the PRO MATE II device programmer can read, verify and program PICmicro devices without a PC connection. It can also set code protection in this mode.

27.13 MPLAB PM3 Device Programmer

The MPLAB PM3 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. MPLAB PM3 connects to the host PC via an RS-232 or USB cable. 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.

PIC18F2455/2550/4455/4550

27.14 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 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.

27.15 PICDEM 1 PICmicro Demonstration Board

The PICDEM 1 demonstration board demonstrates the capabilities of the PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The sample microcontrollers provided with the PICDEM 1 demonstration board can be programmed with a PRO MATE II device programmer or a PICSTART Plus development programmer. The PICDEM 1 demonstration board can be connected to the MPLAB ICE in-circuit emulator for testing. A prototype area extends the circuitry for additional application components. Features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs.

27.16 PICDEM.net Internet/Ethernet Demonstration Board

The PICDEM.net demonstration board is an Internet/Ethernet demonstration board using the PIC18F452 microcontroller and TCP/IP firmware. The board supports any 40-pin DIP device that conforms to the standard pinout used by the PIC16F877 or PIC18C452. This kit features a user friendly TCP/IP stack, web server with HTML, a 24L256 Serial EEPROM for Xmodem download to web pages into Serial EEPROM, ICSP/MPLAB ICD 2 interface connector, an Ethernet interface, RS-232 interface and a 16 x 2 LCD display. Also included is the book and CD-ROM "TCP/IP Lean, Web Servers for Embedded Systems," by Jeremy Bentham

27.17 PICDEM 2 Plus Demonstration Board

The PICDEM 2 Plus demonstration board supports many 18, 28 and 40-pin microcontrollers, including PIC16F87X and PIC18FXX2 devices. All the necessary hardware and software is included to run the demonstration programs. The sample microcontrollers provided with the PICDEM 2 demonstration board can be programmed with a PRO MATE II device programmer, PICSTART Plus development programmer, or MPLAB ICD 2 with a Universal Programmer Adapter. The MPLAB ICD 2 and MPLAB ICE in-circuit emulators may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area extends the circuitry for additional application components. Some of the features include an RS-232 interface, a 2 x 16 LCD display, a piezo speaker, an on-board temperature sensor, four LEDs and sample PIC18F452 and PIC16F877 Flash microcontrollers.

27.18 PICDEM 3 PIC16C92X Demonstration Board

The PICDEM 3 demonstration board supports the PIC16C923 and PIC16C924 in the PLCC package. All the necessary hardware and software is included to run the demonstration programs.

27.19 PICDEM 4 8/14/18-Pin Demonstration Board

The PICDEM 4 can be used to demonstrate the capabilities of the 8, 14 and 18-pin PIC16XXXX and PIC18XXXX MCUs, including the PIC16F818/819, PIC16F87/88, PIC16F62XA and the PIC18F1320 family of microcontrollers. PICDEM 4 is intended to showcase the many features of these low pin count parts, including LIN and Motor Control using ECCP. Special provisions are made for low-power operation with the supercapacitor circuit and jumpers allow on-board hardware to be disabled to eliminate current draw in this mode. Included on the demo board are provisions for Crystal, RC or Canned Oscillator modes, a five volt regulator for use with a nine volt wall adapter or battery, DB-9 RS-232 interface, ICD connector for programming via ICSP and development with MPLAB ICD 2, 2 x 16 liquid crystal display, PCB footprints for H-Bridge motor driver, LIN transceiver and EEPROM. Also included are: header for expansion, eight LEDs, four potentiometers, three push buttons and a prototyping area. Included with the kit is a PIC16F627A and a PIC18F1320. Tutorial firmware is included along with the User's Guide.

27.20 PICDEM 17 Demonstration Board

The PICDEM 17 demonstration board is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756A, PIC17C762 and PIC17C766. A programmed sample is included. The PRO MATE II device programmer, or the PICSTART Plus development programmer, can be used to reprogram the device for user tailored application development. The PICDEM 17 demonstration board supports program download and execution from external on-board Flash memory. A generous prototype area is available for user hardware expansion.

27.21 PICDEM 18R PIC18C601/801 Demonstration Board

The PICDEM 18R demonstration board serves to assist development of the PIC18C601/801 family of Microchip microcontrollers. It provides hardware implementation of both 8-bit Multiplexed/Demultiplexed and 16-bit Memory modes. The board includes 2 Mb external Flash memory and 128 Kb SRAM memory, as well as serial EEPROM, allowing access to the wide range of memory types supported by the PIC18C601/801.

27.22 PICDEM LIN PIC16C43X Demonstration Board

The powerful LIN hardware and software kit includes a series of boards and three PICmicro microcontrollers. The small footprint PIC16C432 and PIC16C433 are used as slaves in the LIN communication and feature on-board LIN transceivers. A PIC16F874 Flash microcontroller serves as the master. All three microcontrollers are programmed with firmware to provide LIN bus communication.

27.23 PICkit™ 1 Flash Starter Kit

A complete “development system in a box”, the PICkit Flash Starter Kit includes a convenient multi-section board for programming, evaluation and development of 8/14-pin Flash PIC® microcontrollers. Powered via USB, the board operates under a simple Windows GUI. The PICkit 1 Starter Kit includes the User’s Guide (on CD ROM), PICkit 1 tutorial software and code for various applications. Also included are MPLAB® IDE (Integrated Development Environment) software, software and hardware “Tips 'n Tricks for 8-pin Flash PIC® Microcontrollers” Handbook and a USB interface cable. Supports all current 8/14-pin Flash PIC microcontrollers, as well as many future planned devices.

27.24 PICDEM USB PIC16C7X5 Demonstration Board

The PICDEM USB Demonstration Board shows off the capabilities of the PIC16C745 and PIC16C765 USB microcontrollers. This board provides the basis for future USB products.

27.25 Evaluation and Programming Tools

In addition to the PICDEM series of circuits, Microchip has a line of evaluation kits and demonstration software for these products.

- KEELOQ evaluation and programming tools for Microchip’s HCS Secure Data Products
- CAN developers kit for automotive network applications
- Analog design boards and filter design software
- PowerSmart battery charging evaluation/calibration kits
- IrDA® development kit
- microID development and rLab™ development software
- SEEVAL® designer kit for memory evaluation and endurance calculations
- PICDEM MSC demo boards for Switching mode power supply, high-power IR driver, delta sigma ADC and flow rate sensor

Check the Microchip web page and the latest Product Selector Guide for the complete list of demonstration and evaluation kits.

PIC18F2455/2550/4455/4550

NOTES:

28.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings^(†)

| | |
|---|-----------------------|
| Ambient temperature under bias | -40°C to +125°C |
| Storage temperature | -65°C to +150°C |
| Voltage on any pin with respect to VSS (except VDD, $\overline{\text{MCLR}}$ and RA4) | -0.3V to (VDD + 0.3V) |
| Voltage on VDD with respect to VSS | -0.3V to +7.5V |
| Voltage on $\overline{\text{MCLR}}$ 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, I _{IK} (V _I < 0 or V _I > VDD) | ±20 mA |
| Output clamp current, I _{OK} (V _O < 0 or V _O > 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:

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

- 2:** Voltage spikes below VSS at the $\overline{\text{MCLR}}$ /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 $\overline{\text{MCLR}}$ /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.

PIC18F2455/2550/4455/4550

FIGURE 28-1: PIC18F2455/2550/4455/4550 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)

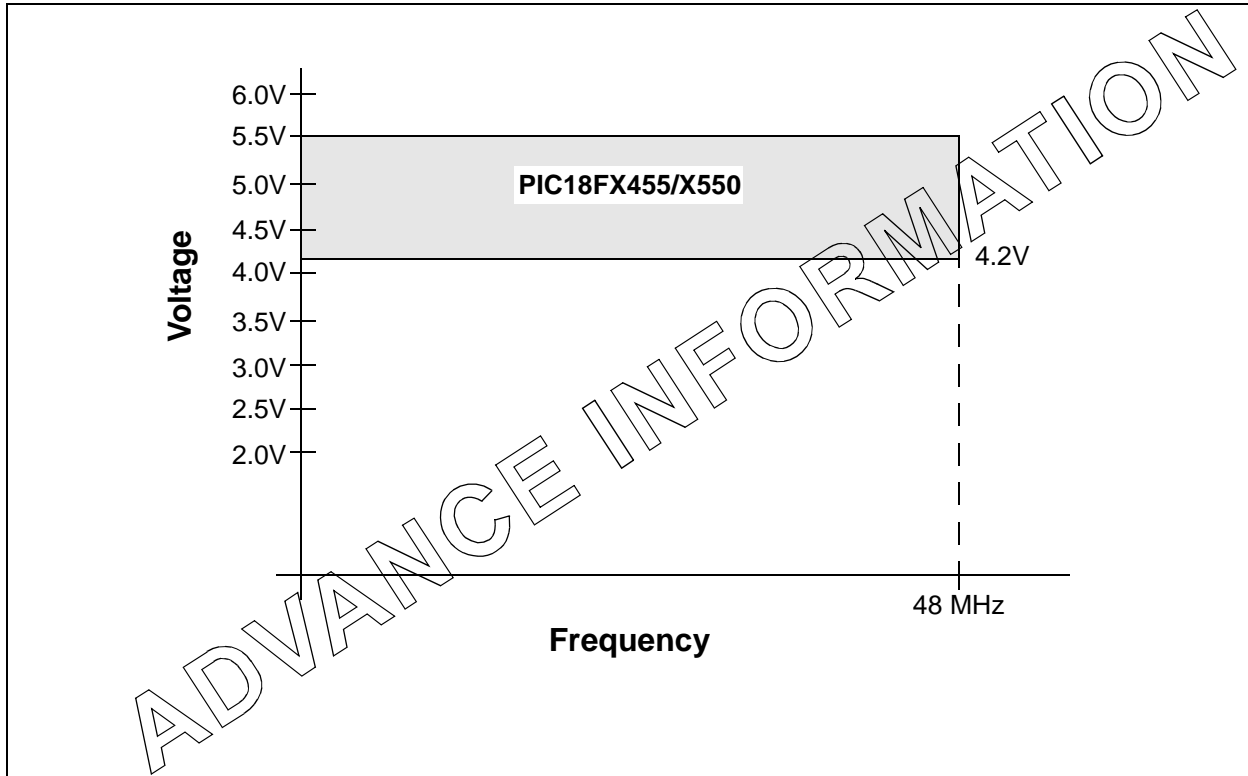
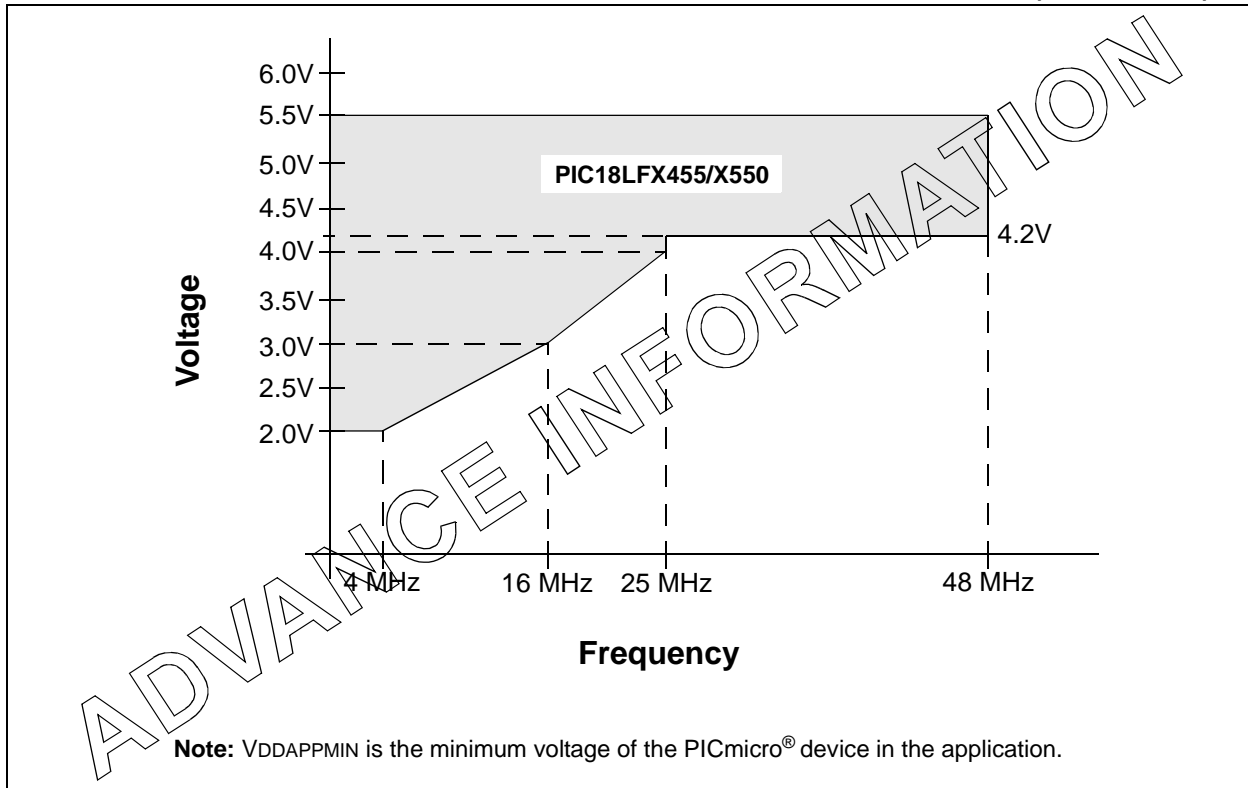


FIGURE 28-2: PIC18LF2455/2550/4455/4550 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)



Note: VDDAPPMIN is the minimum voltage of the PICmicro[®] device in the application.

PIC18F2455/2550/4455/4550

28.1 DC Characteristics: Supply Voltage PIC18F2455/2550/4455/4550 (Industrial) PIC18LF2455/2550/4455/4550 (Industrial)

| PIC18LF2455/2550/4455/4550 (Industrial) | | Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
|--|--------|--|------|------|------|-------|--|
| PIC18F2455/2550/4455/4550 (Industrial) | | Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| Param No. | Symbol | Characteristic | Min | Typ | Max | Units | Conditions |
| D001 | VDD | Supply Voltage | 2.0 | — | 5.5 | V | HS, XT and Internal Oscillator modes |
| | | | 3.0 | — | 5.5 | V | HSPLL, XTPLL, ECPIO and ECPLL modes |
| D002 | VDR | RAM Data Retention Voltage⁽¹⁾ | 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 | 1.96 | 2.06 | 2.16 | V | |
| | | BORV1:BORV0 = 10 | 2.64 | 2.78 | 2.92 | V | |
| | | BORV1:BORV0 = 01 | 4.11 | 4.33 | 4.55 | V | |
| | | BORV1:BORV0 = 00 | 4.41 | 4.64 | 4.87 | 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.

PIC18F2455/2550/4455/4550

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 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial | | | | |
|---|------------------|---|-----|---------------|-----------------------|--|
| PIC18F2455/2550/4455/4550 (Industrial) | | Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial | | | | |
| Param No. | Device | Typ | Max | Units | Conditions | |
| Power-down Current (I_{PD})⁽¹⁾ | | | | | | |
| PIC18LFX455/X550 | PIC18LFX455/X550 | 0.2 | TBD | μA | -40°C | $V_{DD} = 2.0\text{V}$, (Sleep mode) |
| | | 0.2 | TBD | μA | $+25^{\circ}\text{C}$ | |
| | | 0.4 | TBD | μA | $+85^{\circ}\text{C}$ | |
| PIC18LFX455/X550 | PIC18LFX455/X550 | 0.2 | TBD | μA | -40°C | $V_{DD} = 3.0\text{V}$, (Sleep mode) |
| | | 0.2 | TBD | μA | $+25^{\circ}\text{C}$ | |
| | | 0.5 | TBD | μA | $+85^{\circ}\text{C}$ | |
| All devices | All devices | 0.2 | TBD | μA | -40°C | $V_{DD} = 5.0\text{V}$, (Sleep mode) |
| | | 0.2 | TBD | μA | $+25^{\circ}\text{C}$ | |
| | | 1.0 | TBD | μA | $+85^{\circ}\text{C}$ | |

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 V_{DD} or V_{SS} 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 I_{DD} measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{DD} ;
MCLR = V_{DD} ; WDT enabled/disabled as specified.
- 3:** Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to $+70^{\circ}\text{C}$. Extended temperature crystals are available at a much higher cost.

PIC18F2455/2550/4455/4550

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°C ≤ TA ≤ +85°C for industrial | | | | | |
|--|--------|--|-----|-------|------------|------------|---|
| PIC18F2455/2550/4455/4550 (Industrial) | | Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| Param No. | Device | Typ | Max | Units | Conditions | | |
| Supply Current (IDD)⁽²⁾ | | | | | | | |
| PIC18LFX455/X550 | | 11 | TBD | μA | -40°C | VDD = 2.0V | FOSC = 31 kHz (RC_RUN mode, Internal oscillator source) |
| | | 13 | TBD | μA | +25°C | | |
| | | 14 | TBD | μA | +85°C | | |
| PIC18LFX455/X550 | | 34 | TBD | μA | -40°C | VDD = 3.0V | |
| | | 28 | TBD | μA | +25°C | | |
| | | 25 | TBD | μA | +85°C | | |
| All devices | | 77 | TBD | μA | -40°C | VDD = 5.0V | |
| | | 62 | TBD | μA | +25°C | | |
| | | 53 | TBD | μA | +85°C | | |
| PIC18LFX455/X550 | | 100 | TBD | μA | -40°C | VDD = 2.0V | |
| | | 110 | TBD | μA | +25°C | | |
| | | 120 | TBD | μA | +85°C | | |
| PIC18LFX455/X550 | | 180 | TBD | μA | -40°C | VDD = 3.0V | |
| | | 180 | TBD | μA | +25°C | | |
| | | 170 | TBD | μA | +85°C | | |
| All devices | | 340 | TBD | μA | -40°C | VDD = 5.0V | |
| | | 330 | TBD | μA | +25°C | | |
| | | 310 | TBD | μA | +85°C | | |
| PIC18LFX455/X550 | | 359 | TBD | μA | -40°C | VDD = 2.0V | |
| | | 360 | TBD | μA | +25°C | | |
| | | 370 | TBD | μA | +85°C | | |
| PIC18LFX455/X550 | | 0.58 | TBD | mA | -40°C | VDD = 3.0V | |
| | | 0.58 | TBD | mA | +25°C | | |
| | | 0.58 | TBD | mA | +85°C | | |
| All devices | | 1.1 | TBD | mA | -40°C | VDD = 5.0V | |
| | | 1.1 | TBD | mA | +25°C | | |
| | | 1.0 | TBD | 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:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;

MCLR = VDD; WDT enabled/disabled as specified.

3: Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

PIC18F2455/2550/4455/4550

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°C ≤ TA ≤ +85°C for industrial | | | | |
| PIC18F2455/2550/4455/4550 (Industrial) | | Standard Operating Conditions (unless otherwise stated) | | | | |
| | | Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | |
| Param No. | Device | Typ | Max | Units | Conditions | |
| Supply Current (IDD)⁽²⁾ | | | | | | |
| PIC18LFX455/X550 | | 4.7 | TBD | μA | -40°C | VDD = 2.0V VDD = 3.0V VDD = 5.0V Fosc = 31 kHz (RC_IDLE mode, Internal oscillator source) |
| | | 4.6 | TBD | μA | +25°C | |
| | | 5.1 | TBD | μA | +85°C | |
| PIC18LFX455/X550 | | 6.9 | TBD | μA | -40°C | |
| | | 6.3 | TBD | μA | +25°C | |
| | | 6.8 | TBD | μA | +85°C | |
| All devices | | 12 | TBD | μA | -40°C | |
| | | 10 | TBD | μA | +25°C | |
| | | 10 | TBD | μA | +85°C | |
| PIC18LFX455/X550 | | 49 | TBD | μA | -40°C | VDD = 2.0V VDD = 3.0V VDD = 5.0V Fosc = 1 MHz (RC_IDLE mode, Internal oscillator source) |
| | | 52 | TBD | μA | +25°C | |
| | | 56 | TBD | μA | +85°C | |
| PIC18LFX455/X550 | | 73 | TBD | μA | -40°C | |
| | | 77 | TBD | μA | +25°C | |
| | | 77 | TBD | μA | +85°C | |
| All devices | | 130 | TBD | μA | -40°C | |
| | | 130 | TBD | μA | +25°C | |
| | | 130 | TBD | μA | +85°C | |
| PIC18LFX455/X550 | | 140 | TBD | μA | -40°C | VDD = 2.0V VDD = 3.0V VDD = 5.0V Fosc = 4 MHz (RC_IDLE mode, Internal oscillator source) |
| | | 140 | TBD | μA | +25°C | |
| | | 150 | TBD | μA | +85°C | |
| PIC18LFX455/X550 | | 220 | TBD | μA | -40°C | |
| | | 220 | TBD | μA | +25°C | |
| | | 210 | TBD | μA | +85°C | |
| All devices | | 0.39 | TBD | mA | -40°C | |
| | | 0.40 | TBD | mA | +25°C | |
| | | 0.38 | TBD | 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:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;

MCLR = VDD; WDT enabled/disabled as specified.

3: Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

PIC18F2455/2550/4455/4550

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°C ≤ TA ≤ +85°C for industrial | | |
| PIC18F2455/2550/4455/4550 (Industrial) | | Standard Operating Conditions (unless otherwise stated) | | | | |
| | | Operating temperature | | -40°C ≤ TA ≤ +85°C for industrial | | |
| Param No. | Device | Typ | Max | Units | Conditions | |
| Supply Current (I_{DD})⁽²⁾ | | | | | | |
| | PIC18LFX455/X550 | 150 | TBD | μA | -40°C | V _{DD} = 2.0V V _{DD} = 3.0V V _{DD} = 5.0V FOSC = 1 MHz (PRI_RUN, EC oscillator) |
| | | 150 | TBD | μA | +25°C | |
| | | 160 | TBD | μA | +85°C | |
| PIC18LFX455/X550 | 340 | TBD | μA | -40°C | | |
| | 300 | TBD | μA | +25°C | | |
| | 280 | TBD | μA | +85°C | | |
| All devices | 0.72 | TBD | mA | -40°C | | |
| | 0.63 | TBD | mA | +25°C | | |
| | 0.57 | TBD | mA | +85°C | | |
| PIC18LFX455/X550 | 440 | TBD | μA | -40°C | V _{DD} = 2.0V V _{DD} = 3.0V V _{DD} = 5.0V FOSC = 4 MHz (PRI_RUN, EC oscillator) | |
| | 450 | TBD | μA | +25°C | | |
| | 460 | TBD | μA | +85°C | | |
| PIC18LFX455/X550 | 0.80 | TBD | mA | -40°C | | |
| | 0.78 | TBD | mA | +25°C | | |
| | 0.77 | TBD | mA | +85°C | | |
| All devices | 1.6 | TBD | mA | -40°C | | |
| | 1.5 | TBD | mA | +25°C | | |
| | 1.5 | TBD | mA | +85°C | | |
| All devices | 9.9 | TBD | mA | -40°C | V _{DD} = 4.2V V _{DD} = 5.0V FOSC = 40 MHz (PRI_RUN, EC oscillator) | |
| | 9.7 | TBD | mA | +25°C | | |
| | 9.9 | TBD | mA | +85°C | | |
| All devices | 11.9 | TBD | mA | -40°C | | |
| | 12.1 | TBD | mA | +25°C | | |
| | 12.3 | TBD | 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 V_{DD} or V_{SS} 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 I_{DD} measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{DD};

MCLR = V_{DD}; WDT enabled/disabled as specified.

3: Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

PIC18F2455/2550/4455/4550

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 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial | | | | |
| PIC18F2455/2550/4455/4550 (Industrial) | | Standard Operating Conditions (unless otherwise stated) | | | | |
| | | Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial | | | | |
| Param No. | Device | Typ | Max | Units | Conditions | |
| Supply Current (I_{DD})^(2,3) | | | | | | |
| PIC18LFX455/X550 | | 37 | TBD | μA | -40°C | V _{DD} = 2.0V V _{DD} = 3.0V V _{DD} = 5.0V FOSC = 1 MHz (PRI_IDLE mode, EC oscillator) |
| | | 37 | TBD | μA | $+25^{\circ}\text{C}$ | |
| | | 38 | TBD | μA | $+85^{\circ}\text{C}$ | |
| PIC18LFX455/X550 | | 58 | TBD | μA | -40°C | |
| | | 59 | TBD | μA | $+25^{\circ}\text{C}$ | |
| | | 60 | TBD | μA | $+85^{\circ}\text{C}$ | |
| All devices | | 110 | TBD | μA | -40°C | |
| | | 110 | TBD | μA | $+25^{\circ}\text{C}$ | |
| | | 110 | TBD | μA | $+85^{\circ}\text{C}$ | |
| PIC18LFX455/X550 | | 140 | TBD | μA | -40°C | V _{DD} = 2.0V V _{DD} = 3.0V V _{DD} = 5.0V FOSC = 4 MHz (PRI_IDLE mode, EC oscillator) |
| | | 140 | TBD | μA | $+25^{\circ}\text{C}$ | |
| | | 140 | TBD | μA | $+85^{\circ}\text{C}$ | |
| PIC18LFX455/X550 | | 220 | TBD | μA | -40°C | |
| | | 230 | TBD | μA | $+25^{\circ}\text{C}$ | |
| | | 230 | TBD | μA | $+85^{\circ}\text{C}$ | |
| All devices | | 410 | TBD | μA | -40°C | |
| | | 420 | TBD | μA | $+25^{\circ}\text{C}$ | |
| | | 430 | TBD | μA | $+85^{\circ}\text{C}$ | |
| All devices | | 3.7 | TBD | mA | -40°C | V _{DD} = 4.2 V V _{DD} = 5.0V FOSC = 40 MHz (PRI_IDLE mode, EC oscillator) |
| | | 3.2 | TBD | mA | $+25^{\circ}\text{C}$ | |
| | | 3.3 | TBD | mA | $+85^{\circ}\text{C}$ | |
| All devices | | 4.4 | TBD | mA | -40°C | |
| | | 4.6 | TBD | mA | $+25^{\circ}\text{C}$ | |
| | | 4.6 | TBD | mA | $+85^{\circ}\text{C}$ | |

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 V_{DD} or V_{SS} 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 I_{DD} measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{DD};

MCLR = V_{DD}; WDT enabled/disabled as specified.

3: Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to $+70^{\circ}\text{C}$. Extended temperature crystals are available at a much higher cost.

PIC18F2455/2550/4455/4550

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°C ≤ TA ≤ +85°C for industrial | | | | | |
|--|--------|--|-----|-------|------------|------------------------|---|
| PIC18F2455/2550/4455/4550 (Industrial) | | Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| Param No. | Device | Typ | Max | Units | Conditions | | |
| Supply Current (I_{DD})^(2,3) | | | | | | | |
| PIC18LFX455/X550 | | 13 | TBD | μA | -10°C | V _{DD} = 2.0V | F _{OSC} = 32 kHz ⁽³⁾ (SEC_RUN mode, Timer1 as clock) |
| | | 14 | TBD | μA | +25°C | | |
| | | 16 | TBD | μA | +70°C | | |
| PIC18LFX455/X550 | | 34 | TBD | μA | -10°C | V _{DD} = 3.0V | |
| | | 31 | TBD | μA | +25°C | | |
| | | 28 | TBD | μA | +70°C | | |
| All devices | | 72 | TBD | μA | -10°C | V _{DD} = 5.0V | |
| | | 65 | TBD | μA | +25°C | | |
| | | 59 | TBD | μA | +70°C | | |
| PIC18LFX455/X550 | | 5.5 | TBD | μA | -10°C | V _{DD} = 2.0V | F _{OSC} = 32 kHz ⁽³⁾ (SEC_IDLE mode, Timer1 as clock) |
| | | 5.8 | TBD | μA | +25°C | | |
| | | 6.1 | TBD | μA | +70°C | | |
| PIC18LFX455/X550 | | 8.2 | TBD | μA | -10°C | V _{DD} = 3.0V | |
| | | 8.6 | TBD | μA | +25°C | | |
| | | 8.8 | TBD | μA | +70°C | | |
| All devices | | 13 | TBD | μA | -10°C | V _{DD} = 5.0V | |
| | | 13 | TBD | μA | +25°C | | |
| | | 13 | TBD | μA | +70°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 V_{DD} or V_{SS} 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 I_{DD} measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{DD};
MCLR = V_{DD}; WDT enabled/disabled as specified.
- 3:** Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

PIC18F2455/2550/4455/4550

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 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial | | | | | |
|--|-----------------------------|---|---------------|-----------------------|--|------------------------|-----------------------------------|
| PIC18F2455/2550/4455/4550 (Industrial) | | Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial | | | | | |
| Param No. | Device | Typ | Max | Units | Conditions | | |
| D022 (ΔI_{WDT}) | Watchdog Timer | 1.7 | TBD | μA | -40°C | $V_{DD} = 2.0\text{V}$ | |
| | | 2.1 | TBD | μA | $+25^{\circ}\text{C}$ | | |
| | | 2.6 | TBD | μA | $+85^{\circ}\text{C}$ | | |
| | | $V_{DD} = 3.0\text{V}$ | 2.2 | TBD | μA | -40°C | |
| | | | 2.4 | TBD | μA | $+25^{\circ}\text{C}$ | |
| | | | 2.8 | TBD | μA | $+85^{\circ}\text{C}$ | |
| | | | 2.9 | TBD | μA | -40°C | |
| | | | 3.1 | TBD | μA | $+25^{\circ}\text{C}$ | |
| | | | 3.3 | TBD | μA | $+85^{\circ}\text{C}$ | |
| D022A (ΔI_{BOR}) | Brown-out Reset | 17 | TBD | μA | -40°C to $+85^{\circ}\text{C}$ | $V_{DD} = 3.0\text{V}$ | |
| | | 47 | TBD | μA | -40°C to $+85^{\circ}\text{C}$ | $V_{DD} = 5.0\text{V}$ | |
| | | 0 | TBD | μA | -40°C to $+85^{\circ}\text{C}$ | $V_{DD} = 5.0\text{V}$ | Sleep mode, BOREN1:BOREN0 = 10 |
| D022B (ΔI_{LVD}) | Low-Voltage Detect | 14 | TBD | μA | -40°C to $+85^{\circ}\text{C}$ | $V_{DD} = 2.0\text{V}$ | |
| | | 18 | TBD | μA | -40°C to $+85^{\circ}\text{C}$ | $V_{DD} = 3.0\text{V}$ | |
| | | 21 | TBD | μA | -40°C to $+85^{\circ}\text{C}$ | $V_{DD} = 5.0\text{V}$ | |
| D025 (ΔI_{OSCB}) | Timer1 Oscillator | 1.0 | TBD | μA | -10°C | $V_{DD} = 2.0\text{V}$ | 32 kHz on Timer1 ⁽⁴⁾ |
| | | 1.1 | TBD | μA | $+25^{\circ}\text{C}$ | | |
| | | 1.1 | TBD | μA | $+70^{\circ}\text{C}$ | | |
| | | $V_{DD} = 3.0\text{V}$ | 1.2 | TBD | μA | -10°C | |
| | | | 1.3 | TBD | μA | $+25^{\circ}\text{C}$ | |
| | | | 1.2 | TBD | μA | $+70^{\circ}\text{C}$ | |
| | | | 1.8 | TBD | μA | -10°C | |
| | | | 1.9 | TBD | μA | $+25^{\circ}\text{C}$ | |
| $V_{DD} = 5.0\text{V}$ | 1.9 | TBD | μA | $+70^{\circ}\text{C}$ | 32 kHz on Timer1 ⁽⁴⁾ | | |
| | D026 (ΔI_{AD}) | A/D Converter | 1.0 | TBD | μA | | $V_{DD} = 2.0\text{V}$ |
| 1.0 | | | TBD | μA | | $V_{DD} = 3.0\text{V}$ | |
| 1.0 | | | TBD | μA | | $V_{DD} = 5.0\text{V}$ | |

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 V_{DD} or V_{SS} and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).
- 2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{DD} ;
MCLR = V_{DD} ; WDT enabled/disabled as specified.
- 3:** Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to $+70^{\circ}\text{C}$. Extended temperature crystals are available at a much higher cost.

PIC18F2455/2550/4455/4550

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°C ≤ TA ≤ +85°C for industrial | | | | |
|--|---------------------------------------|--|-----|-------|------------|------------|
| PIC18F2455/2550/4455/4550 (Industrial) | | Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | |
| Param No. | Device | Typ | Max | Units | Conditions | |
| USB and Related Module Differential Currents (ΔI_{USB}, ΔI_{PLL}, ΔI_{UREG}, ΔI_{XCVR}) | | | | | | |
| ΔI_{USB} | USB Module | | | | +25°C | VDD = 3.0V |
| | | | | | +25°C | VDD = 5.0V |
| ΔI_{PLL} | 96 MHz USB PLL (Oscillator Module) | | | | +25°C | VDD = 3.0V |
| | | | | | +25°C | VDD = 5.0V |
| ΔI_{UREG} | USB Internal Voltage Regulator | | | | +25°C | VDD = 3.0V |
| | | | | | +25°C | VDD = 5.0V |
| ΔI_{XCVR} | USB On-Chip Transceiver | | | | +25°C | VDD = 3.0V |
| | | | | | +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:
 OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
 MCLR = VDD; WDT enabled/disabled as specified.
- 3:** Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

PIC18F2455/2550/4455/4550

28.3 DC Characteristics: PIC18F2455/2550/4455/4550 (Industrial) PIC18LF2455/2550/4455/4550 (Industrial)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial | | | |
|--|---|--|--|---|--------------------------------------|--|
| Param No. | Symbol | Characteristic | Min | Max | Units | Conditions |
| D030 D030A D031 D032 D032A D033 | V _{IL} V _{ILU} | Input Low Voltage I/O ports (except RC4/RC5 in USB mode): with TTL buffer with Schmitt Trigger buffer RC3 and RC4 <u>MCLR</u> OSC1 and T1OSI OSC1 D+/D- Input | V _{SS} — V _{SS} V _{SS} V _{SS} V _{SS} | 0.15 V _{DD} 0.8 0.2 V _{DD} 0.3 V _{DD} 0.2 V _{DD} 0.3 V _{DD} 0.2 V _{DD} 0.8 | V V V V V V V V | V _{DD} < 4.5V 4.5V ≤ V _{DD} ≤ 5.5V LP, XT, HS, HSPLL modes ⁽¹⁾ EC mode ⁽¹⁾ V _{DD} = 4.35V, USB suspended ⁽⁵⁾ |
| D040 D040A D041 D042 D042A D043 | V _{IH} V _{IHU} | Input High Voltage I/O ports (except RC4/RC5 in USB mode): with TTL buffer with Schmitt Trigger buffer RC3 and RC4 <u>MCLR</u> OSC1 and T1OSI OSC1 D+/D- Input | 0.25 V _{DD} + 0.8V 2.0 0.8 V _{DD} 0.7 V _{DD} 0.8 V _{DD} 0.7 V _{DD} 0.8 V _{DD} 2.4 | V _{DD} V _{DD} V _{DD} V _{DD} V _{DD} V _{DD} V _{DD} — | V V V V V V V V | V _{DD} < 4.5V 4.5V ≤ V _{DD} ≤ 5.5V LP, XT, HS, HSPLL modes ⁽¹⁾ EC mode ⁽¹⁾ V _{DD} = 4.35V, USB suspended ⁽⁵⁾ |
| D060 D061 D063 | I _{IL} | Input Leakage Current^(2,3) I/O ports <u>MCLR</u> OSC1 | — — — | ±1 ±5 ±5 | μA μA μA | V _{SS} ≤ V _{PIN} ≤ V _{DD} , Pin at high-impedance V _{SS} ≤ V _{PIN} ≤ V _{DD} V _{SS} ≤ V _{PIN} ≤ V _{DD} |
| D070 | I _{PU} I _{PURB} | Weak Pull-up Current PORTB weak pull-up current | 50 | 400 | μA | V _{DD} = 5V, V _{PIN} = V _{SS} |

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.

PIC18F2455/2550/4455/4550

28.3 DC Characteristics: PIC18F2455/2550/4455/4550 (Industrial) PIC18LF2455/2550/4455/4550 (Industrial) (Continued)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial | | | |
|---------------------|--------|--|---|-----|-------|--|
| Param No. | Symbol | Characteristic | Min | Max | Units | Conditions |
| D080 | VOL | Output Low Voltage I/O ports (except RC4/RC5 in USB mode) | — | 0.6 | V | $I_{OL} = 8.5 \text{ mA}$, $V_{DD} = 4.5\text{V}$, -40°C to $+85^{\circ}\text{C}$ |
| D083 | | OSC2/CLKO (EC, ECIO modes) | — | 0.6 | V | $I_{OL} = 1.6 \text{ mA}$, $V_{DD} = 4.5\text{V}$, -40°C to $+85^{\circ}\text{C}$ |
| | VOLU | D+/D- Out | — | 0.3 | | $V_{DD} = 4.35\text{V}$, USB suspended ⁽⁵⁾ |
| D090 | VOH | Output High Voltage⁽³⁾ I/O ports (except RC4/RC5 in USB mode) | $V_{DD} - 0.7$ | — | | $I_{OH} = -3.0 \text{ mA}$, $V_{DD} = 4.5\text{V}$, -40°C to $+85^{\circ}\text{C}$ |
| D092 | | OSC2/CLKO (EC, ECIO, ECPIO modes) | $V_{DD} - 0.7$ | — | V | $I_{OH} = -1.3 \text{ mA}$, $V_{DD} = 4.5\text{V}$, -40°C to $+85^{\circ}\text{C}$ |
| | VOHU | D+/D- Out | 2.8 | 3.6 | V | $V_{DD} = 4.35\text{V}$, USB suspended ⁽⁵⁾ |
| | | Capacitive Loading Specs on Output Pins | | | | |
| D100 ⁽⁴⁾ | Cosc2 | OSC2 pin | — | 15 | pF | In XT, HS and LP modes when external clock is used to drive OSC1 |
| D101 | Cio | All I/O pins and OSC2 (in RC mode) | — | 50 | pF | To meet the AC Timing Specifications |
| D102 | Cb | SCL, SDA | — | 400 | pF | I ² 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.

PIC18F2455/2550/4455/4550

TABLE 28-1: MEMORY PROGRAMMING REQUIREMENTS

| DC Characteristics | | | Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial | | | | |
|---|-----------------|--|---|------|-------|-------|---|
| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
| Internal Program Memory Programming Specifications⁽¹⁾ | | | | | | | |
| 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 Using EECON to read/write V _{MIN} = Minimum operating voltage |
| D121 | VDRW | VDD for Read/Write | V _{MIN} | — | 5.5 | V | |
| D122 | TDEW | Erase/Write Cycle Time | — | 4 | — | ms | Provided no other specifications are violated -40°C to +85°C |
| D123 | TRETD | Characteristic Retention | 40 | — | — | Year | |
| D124 | TREF | Number of Total Erase/Write Cycles before Refresh ⁽²⁾ | 1M | 10M | — | E/W | |
| Program Flash Memory | | | | | | | |
| D130 | EP | Cell Endurance | 10K | 100K | — | E/W | -40°C to +85°C |
| D131 | VPR | VDD for Read | V _{MIN} | — | 5.5 | V | V _{MIN} = Minimum operating voltage |
| D132 | VIE | VDD for Block Erase | 4.5 | — | 5.5 | V | Using ICSP™ port |
| D132A | V _{IW} | VDD for Externally Timed Erase or Write | 4.5 | — | 5.5 | V | Using ICSP port |
| D132B | VPEW | VDD for Self-timed Write | V _{MIN} | — | 5.5 | V | V _{MIN} = Minimum operating voltage |
| D133 | TIE | ICSP Block Erase Cycle Time | — | 4 | — | ms | VDD > 4.5V |
| D133A | TIW | ICSP Erase or Write Cycle Time (externally timed) | 1 | — | — | ms | VDD > 4.5V |
| D133A | TIW | Self-timed Write Cycle Time | — | 2 | — | ms | |
| D134 | TRETD | Characteristic Retention | 40 | 100 | — | Year | Provided no other specifications are violated |

† 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.
- 2:** Refer to **Section 7.7 “Using the Data EEPROM”** for a more detailed discussion on data EEPROM endurance.
- 3:** Required only if single-supply programming is disabled.

PIC18F2455/2550/4455/4550

TABLE 28-2: COMPARATOR SPECIFICATIONS

| Operating Conditions: $3.0V < V_{DD} < 5.5V$, $-40^{\circ}C < T_A < +85^{\circ}C$ (unless otherwise stated) | | | | | | | |
|--|--------------------|---|-----|-------|----------------|-------|--|
| Param No. | Sym | Characteristics | Min | Typ | Max | Units | Comments |
| D300 | V _{IOFF} | Input Offset Voltage | — | ± 5.0 | ± 10 | mV | |
| D301 | V _{ICM} | Input Common Mode Voltage* | 0 | — | $V_{DD} - 1.5$ | V | |
| D302 | CMRR | Common Mode Rejection Ratio* | 55 | — | — | dB | |
| 300 | T _{RESP} | Response Time ^{(1)*} | — | 150 | 400 | ns | PIC18FXXXX |
| 300A | | | — | 150 | 600 | ns | PIC18LFXXXX, V _{DD} = 2.0V |
| 301 | T _{MC2OV} | Comparator Mode Change to Output Valid* | — | — | 10 | μs | |

* These parameters are characterized but not tested.

Note 1: Response time measured with one comparator input at $(V_{DD} - 1.5)/2$, while the other input transitions from V_{SS} to V_{DD}.

TABLE 28-3: VOLTAGE REFERENCE SPECIFICATIONS

| Operating Conditions: $3.0V < V_{DD} < 5.5V$, $-40^{\circ}C < T_A < +85^{\circ}C$ (unless otherwise stated) | | | | | | | |
|--|------------------|------------------------------|-------------|-----|-------------|-------|-----------------------|
| Param No. | Sym | Characteristics | Min | Typ | Max | Units | Comments |
| D310 | V _{RES} | Resolution | $V_{DD}/24$ | — | $V_{DD}/32$ | LSb | |
| D311 | V _{RAA} | Absolute Accuracy | — | 1/4 | 1 | LSb | Low Range (CVRR = 1) |
| | | | — | — | 1/2 | LSb | High Range (CVRR = 0) |
| D312 | V _{RUR} | Unit Resistor Value (R)* | — | 2k | — | Ω | |
| 310 | T _{SET} | Settling Time ⁽¹⁾ | — | — | 10 | μs | |

* These parameters are characterized but not tested.

Note 1: Settling time measured while CVRR = 1 and CVR3:CVR0 transitions from '0000' to '1111'.

PIC18F2455/2550/4455/4550

TABLE 28-4: USB MODULE SPECIFICATIONS

| Operating Conditions: $-40^{\circ}\text{C} < T_A < +85^{\circ}\text{C}$ (unless otherwise stated). | | | | | | | |
|--|--------|-----------------------------------|-----|-----|---------|---------------|--|
| Param No. | Sym | Characteristic | Min | Typ | Max | Units | Comments |
| | VUSB | USB Voltage | 3.0 | — | 3.6 | V | Voltage on bus must be in this range for proper USB operation |
| | IIL | Input Leakage on pin | — | — | ± 1 | μA | $V_{SS} \leq V_{PAD} \leq V_{DD}$; Pin at high impedance |
| | VILUSB | Input Low Voltage for USB Buffer | — | — | 0.8 | V | For VUSB range |
| | VIHUSB | Input High Voltage for USB Buffer | 2.0 | — | — | V | For VUSB range |
| | VCRS | Crossover Voltage | 1.3 | — | 2.0 | V | Voltage range for pad_dp and pad_dm crossover to occur |
| | VDIFS | Differential Input Sensitivity | — | — | 0.2 | V | The difference between D+ and D- must exceed this value while VCM is met |
| | VCM | Differential Common Mode Range | 0.8 | — | 2.5 | V | |
| | ZOUT | Driver Output Impedance | 28 | — | 44 | Ω | |
| | VOL | Voltage Output Low | 0.0 | — | 0.3 | V | 1.5 k Ω load connected to 3.6V |
| | VOH | Voltage Output High | 2.8 | — | 3.6 | V | 15 k Ω load connected to ground |

TABLE 28-5: USB INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

| Operating Conditions: $-40^{\circ}\text{C} < T_A < +85^{\circ}\text{C}$ (unless otherwise stated). | | | | | | | |
|--|---------|----------------------------------|-----|-----|-----|-------|---|
| Param No. | Sym | Characteristics | Min | Typ | Max | Units | Comments |
| | VUSBANA | Regulator Output Voltage* | 3.0 | — | 3.6 | V | |
| | CUSB | External Filter Capacitor Value* | — | 220 | — | nF | Must hold sufficient charge for peak load with minimal voltage drop |

* These parameters are characterized but not tested. Parameter numbers not yet assigned for these specifications.

PIC18F2455/2550/4455/4550

FIGURE 28-3: HIGH/LOW-VOLTAGE DETECT CHARACTERISTICS

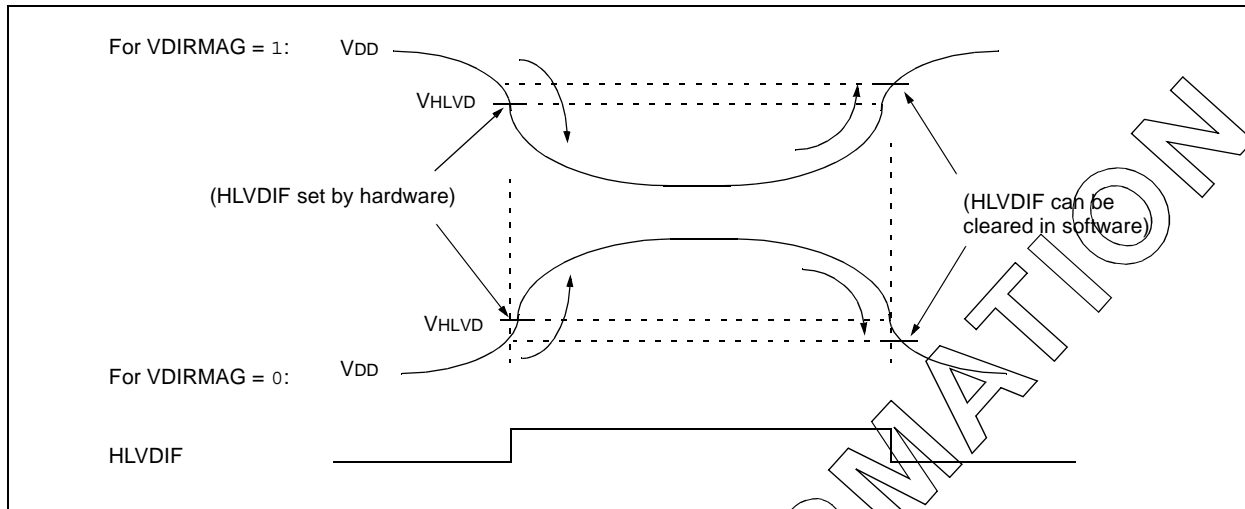


TABLE 28-6: HIGH/LOW-VOLTAGE DETECT CHARACTERISTICS

| | | | Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial | | | | | |
|------------|--------|--------------------------------|---|------|------|-------|------------|----------|
| Param No. | Symbol | Characteristic | Min | Typ† | Max | Units | Conditions | |
| D420 | | HLVD Voltage on VDD Transition | LVV = 0000 | N/A | N/A | N/A | V | Reserved |
| | | | LVV = 0001 | N/A | N/A | N/A | V | Reserved |
| | | | LVV = 0010 | 2.16 | 2.27 | 2.38 | V | |
| | | | LVV = 0011 | 2.35 | 2.47 | 2.59 | V | |
| | | | LVV = 0100 | 2.43 | 2.56 | 2.69 | V | |
| | | | LVV = 0101 | 2.64 | 2.78 | 2.92 | V | |
| | | | LVV = 0110 | 2.75 | 2.89 | 3.03 | V | |
| | | | LVV = 0111 | 2.95 | 3.10 | 3.26 | V | |
| | | | LVV = 1000 | 3.24 | 3.41 | 3.58 | V | |
| | | | LVV = 1001 | 3.43 | 3.61 | 3.79 | V | |
| | | | LVV = 1010 | 3.53 | 3.72 | 3.91 | V | |
| | | | LVV = 1011 | 3.72 | 3.92 | 4.12 | V | |
| | | | LVV = 1100 | 3.92 | 4.13 | 4.34 | V | |
| | | | LVV = 1101 | 4.11 | 4.33 | 4.55 | V | |
| LVV = 1110 | 4.41 | 4.64 | 4.87 | V | | | | |

† Production tested at $T_{AMB} = 25^{\circ}\text{C}$. Specifications over temperature limits ensured by characterization.

PIC18F2455/2550/4455/4550

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. TppS2ppS
2. TppS
3. TCC:ST (I²C specifications only)
4. Ts (I²C specifications only)

| | |
|----------------------------------|----------------------|
| <p>T</p> <p>F Frequency</p> | <p>T</p> <p>Time</p> |
|----------------------------------|----------------------|

Lowercase letters (pp) and their meanings:

| | |
|---|--|
| <p>pp</p> <p>ad SPP address write</p> <p>cc CCP1</p> <p>ck CLKO</p> <p>cs \overline{CS}</p> <p>da SPP data write</p> <p>di SDI</p> <p>do SDO</p> <p>dt Data in</p> <p>io I/O port</p> | <p>mc \overline{MCLR}</p> <p>osc OSC1</p> <p>rd \overline{RD}</p> <p>rw \overline{RD} or \overline{WR}</p> <p>sc \overline{SCK}</p> <p>ss \overline{SS}</p> <p>t0 T0CKI</p> <p>t1 T13CKI</p> <p>wr \overline{WR}</p> |
|---|--|

Uppercase letters and their meanings:

| | |
|--|--|
| <p>S</p> <p>F Fall</p> <p>H High</p> <p>I Invalid (High-impedance)</p> <p>L Low</p> <p>I²C only</p> <p>AA output access</p> <p>BUF Bus free</p> | <p>P Period</p> <p>R Rise</p> <p>V Valid</p> <p>Z High-impedance</p> <p>High High</p> <p>Low Low</p> |
|--|--|

TCC:ST (I²C specifications only)

| | |
|---|---|
| <p>CC</p> <p>HD Hold</p> <p>ST</p> <p>DAT DATA input hold</p> <p>STA Start condition</p> | <p>SU Setup</p> <p>STO Stop condition</p> |
|---|---|

PIC18F2455/2550/4455/4550

28.4.2 TIMING CONDITIONS

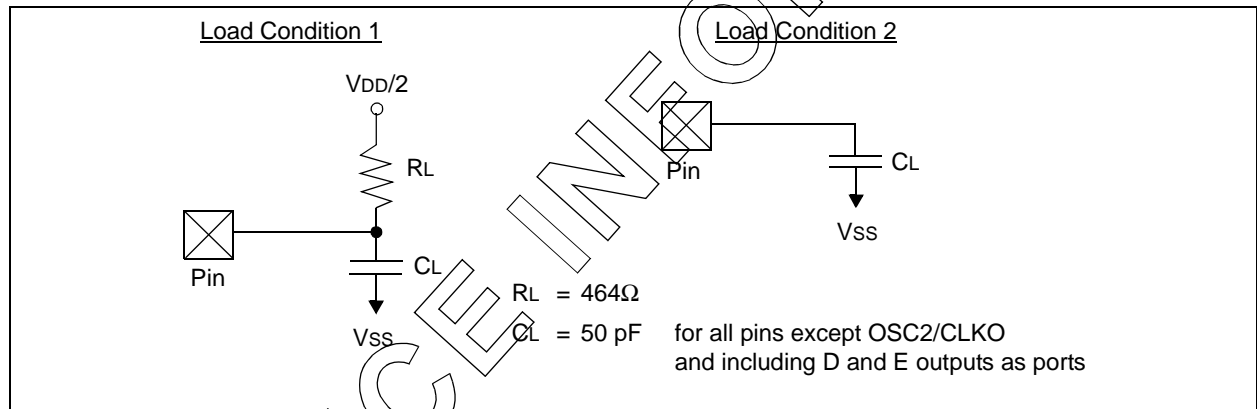
The temperature and voltages specified in Table 28-7 apply to all timing specifications unless otherwise noted. Figure 28-4 specifies the load conditions for the timing specifications.

Note: 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

| | |
|---------------------------|--|
| AC CHARACTERISTICS | Standard Operating Conditions (unless otherwise stated) |
| | Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial |
| | Operating voltage V_{DD} 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



PIC18F2455/2550/4455/4550

28.4.3 TIMING DIAGRAMS AND SPECIFICATIONS

FIGURE 28-5: EXTERNAL CLOCK TIMING (ALL MODES EXCEPT PLL)

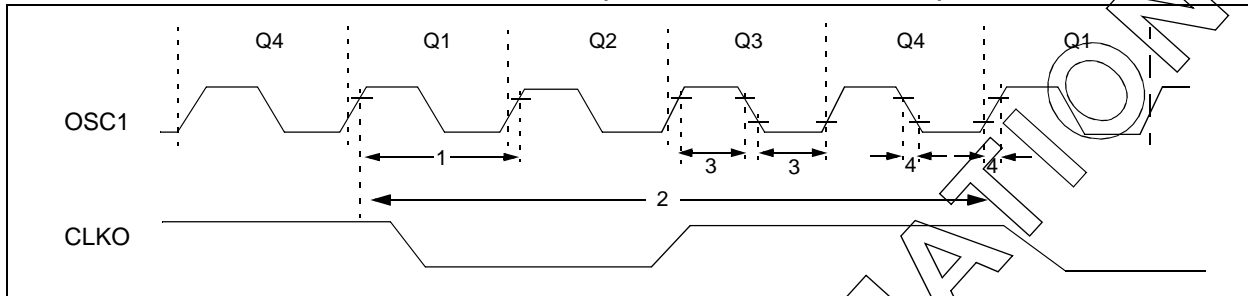


TABLE 28-8: EXTERNAL CLOCK TIMING REQUIREMENTS

| Param. No. | Symbol | Characteristic | Min | Max | Units | Conditions |
|------------|--|---|------|--------|-------|---------------------------|
| 1A | FOSC | External CLKI Frequency ⁽¹⁾ | DC | 48 | MHz | EC, ECIO Oscillator mode |
| | | Oscillator Frequency ⁽¹⁾ | 0.1 | 4 | MHz | XT, XTPLL Oscillator mode |
| | | | 4 | 25 | MHz | HS Oscillator mode |
| | | | 4 | 48 | MHz | HSPLL Oscillator mode |
| 1 | TOSC | External CLKI Period ⁽¹⁾ | 20.8 | — | ns | EC, ECIO Oscillator mode |
| | | Oscillator Period ⁽¹⁾ | 250 | 10,000 | ns | XT Oscillator mode |
| | | | 25 | 250 | ns | HS Oscillator mode |
| | | | 20.8 | 250 | ns | HSPLL Oscillator mode |
| 2 | T _{cy} | Instruction Cycle Time ⁽¹⁾ | 83.3 | — | ns | T _{cy} = 4/FOSC |
| 3 | T _{osL} , T _{osH} | External Clock in (OSC1) High or Low Time | 30 | — | ns | XT Oscillator mode |
| | | | 10 | — | ns | HS Oscillator mode |
| 4 | T _{osR} , T _{osF} | External Clock in (OSC1) Rise or Fall Time | — | 20 | ns | XT Oscillator mode |
| | | | — | 7.5 | ns | HS Oscillator mode |

Note 1: Instruction cycle period (T_{cy}) 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.

PIC18F2455/2550/4455/4550

TABLE 28-9: PLL CLOCK TIMING SPECIFICATIONS (V_{DD} = 4.2V 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 _{rc} | PLL Start-up Time (Lock Time) | — | — | 2 | ms | |
| F13 | ΔCLK | CLKO Stability (Jitter) | -0.25 | — | +0.25 | % | |

† 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)**

| PIC18LF2455/2550/4455/4550 (Industrial) | | Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
|---|--------|--|------|--------|-------|----------------|----------------------------|
| PIC18F2455/2550/4455/4550 (Industrial) | | Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| Param No. | Device | Min | Typ | Max | Units | Conditions | |
| INTOSC Accuracy @ Freq = 8 MHz, 4 MHz, 2 MHz, 1 MHz, 500 kHz, 250 kHz, 125 kHz⁽¹⁾ | | | | | | | |
| PIC18LF2455/2550/4455/4550 | | -2 | +/-1 | 2 | % | +25°C | V _{DD} = 2.7-3.3V |
| | | -5 | — | 5 | % | -10°C to +85°C | V _{DD} = 2.7-3.3V |
| | | -10 | +/-1 | 10 | % | -40°C to +85°C | V _{DD} = 2.7-3.3V |
| PIC18F2455/2550/4455/4550 | | -2 | +/-1 | 2 | % | +25°C | V _{DD} = 4.5-5.5V |
| | | -5 | — | 5 | % | -10°C to +85°C | V _{DD} = 4.5-5.5V |
| | | -10 | +/-1 | 10 | % | -40°C to +85°C | V _{DD} = 4.5-5.5V |
| INTRC Accuracy @ Freq = 31 kHz⁽²⁾ | | | | | | | |
| PIC18LF2455/2550/4455/4550 | | 26.562 | — | 35.938 | kHz | -40°C to +85°C | V _{DD} = 2.7-3.3V |
| PIC18F2455/2550/4455/4550 | | 26.562 | — | 35.938 | kHz | -40°C to +85°C | V _{DD} = 4.5-5.5V |

Legend: Shading of rows is to assist in readability of the table.

- Note 1:** Frequency calibrated at 25°C. OSC_{TUNE} register can be used to compensate for temperature drift.
Note 2: INTRC frequency after calibration.
Note 3: Change of INTRC frequency as V_{DD} changes.

PIC18F2455/2550/4455/4550

FIGURE 28-6: CLKO AND I/O TIMING

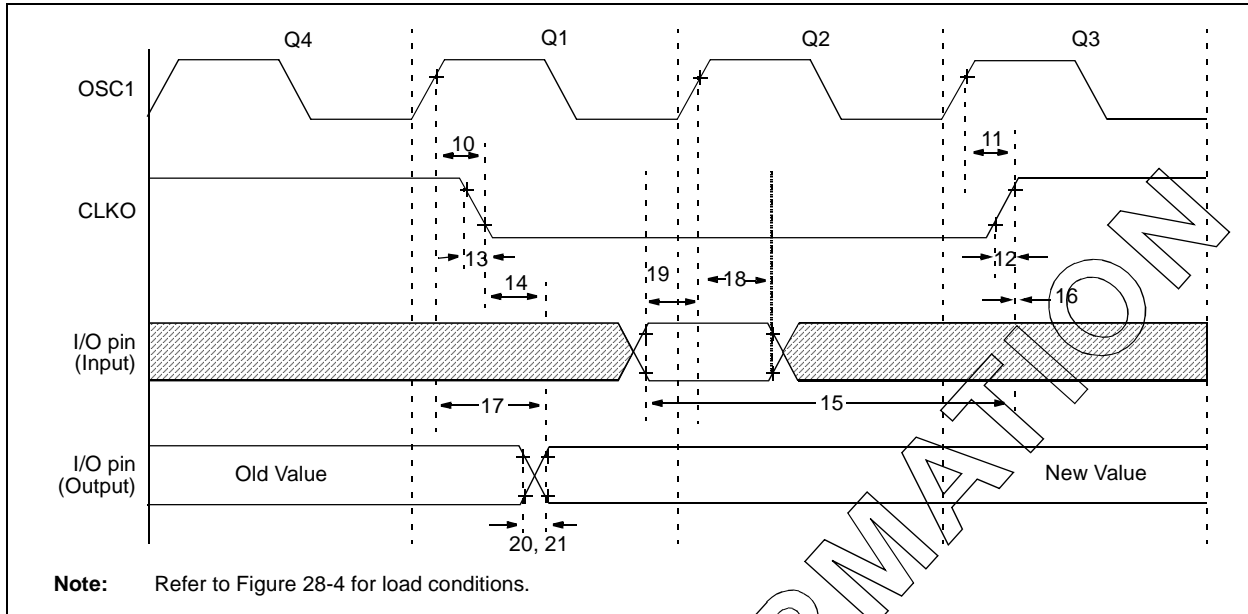


TABLE 28-11: CLKO AND I/O TIMING REQUIREMENTS

| Param No. | Symbol | Characteristic | Min | Typ | 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 T _{CY} + 20 | ns | (Note 1) | |
| 15 | TioV2ckH | Port In Valid before CLKO ↑ | 0.25 T _{CY} + 25 | — | — | ns | (Note 1) | |
| 16 | TckH2ioI | Port In Hold after CLKO ↑ | 0 | — | — | ns | (Note 1) | |
| 17 | TosH2ioV | OSC1 ↑ (Q1 cycle) to Port Out Valid | — | 50 | 150 | ns | | |
| 18 | TosH2ioI | OSC1 ↑ (Q2 cycle) to Port Input Invalid (I/O in hold time) | PIC18FXXXX | 100 | — | — | ns | |
| 18A | | | PIC18LFXXXX | 200 | — | — | ns | V _{DD} = 2.0V |
| 19 | TioV2osH | Port Input Valid to OSC1 ↑ (I/O in setup time) | 0 | — | — | ns | | |
| 20 | TioR | Port Output Rise Time | PIC18FXXXX | — | 10 | 25 | ns | |
| 20A | | | PIC18LFXXXX | — | — | 60 | ns | V _{DD} = 2.0V |
| 21 | TioF | Port Output Fall Time | PIC18FXXXX | — | 10 | 25 | ns | |
| 21A | | | PIC18LFXXXX | — | — | 60 | ns | V _{DD} = 2.0V |
| 22† | TINP | INT pin High or Low Time | T _{CY} | — | — | ns | | |
| 23† | TRBP | RB7:RB4 Change INT High or Low Time | T _{CY} | — | — | ns | | |
| 24† | TRCP | RC7:RC4 Change INT High or Low Time | 20 | | | 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 T_{osc}.

PIC18F2455/2550/4455/4550

FIGURE 28-7: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

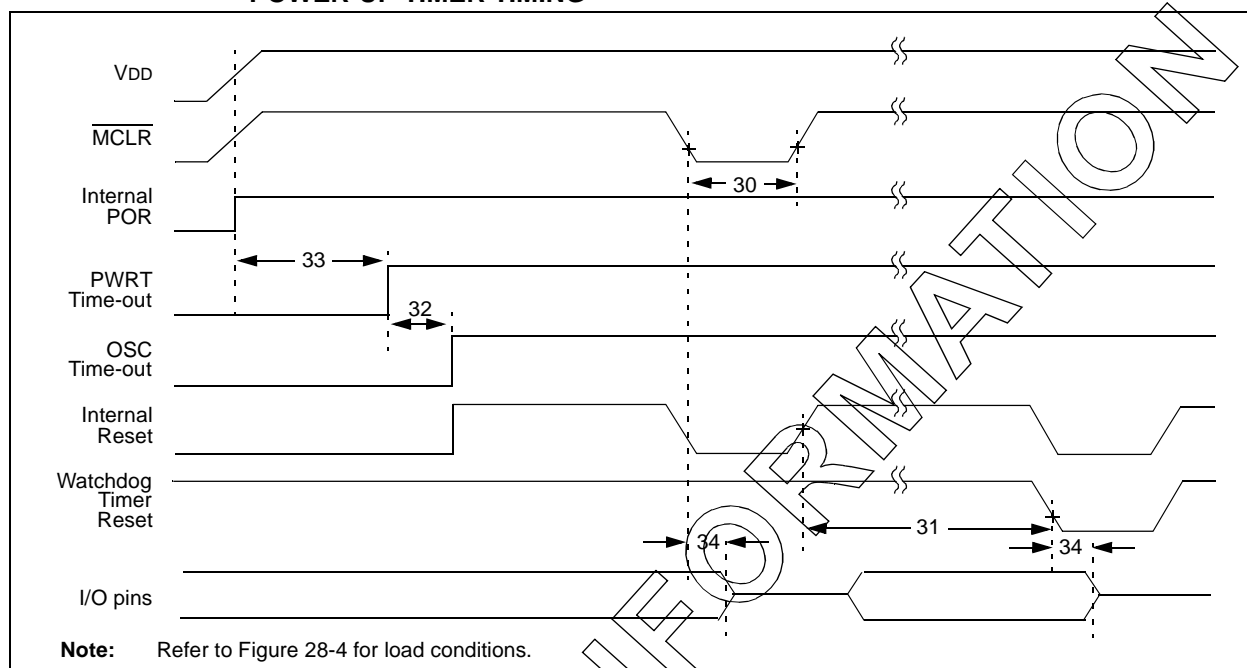


FIGURE 28-8: BROWN-OUT RESET TIMING

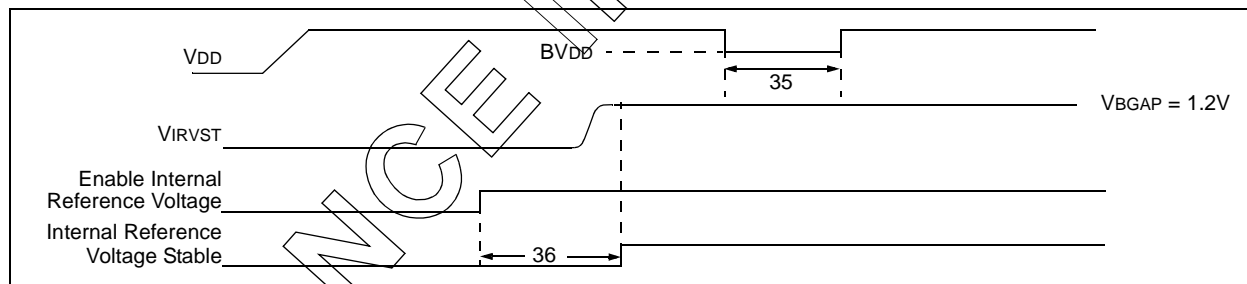


TABLE 28-12: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET REQUIREMENTS

| Param. No. | Symbol | Characteristic | Min | Typ | Max | Units | Conditions |
|------------|--------------------|--|-----------------------|------|-----------------------|-------|--------------------------------|
| 30 | T _{MCLR} | MCLR Pulse Width (low) | 2 | — | — | μs | |
| 31 | T _{WDT} | Watchdog Timer Time-out Period (no postscaler) | — | 4.00 | TBD | ms | |
| 32 | T _{OST} | Oscillation Start-up Timer Period | 1024 T _{osc} | — | 1024 T _{osc} | — | T _{osc} = OSC1 period |
| 33 | T _{PWRT} | Power-up Timer Period | — | 65.5 | TBD | ms | |
| 34 | T _{IOZ} | I/O High-Impedance from MCLR Low or Watchdog Timer Reset | — | 2 | — | μs | |
| 35 | T _{BOR} | Brown-out Reset Pulse Width | 200 | — | — | μs | VDD ≤ BVDD (see D005) |
| 36 | T _{IVRST} | Time for Internal Reference Voltage to become Stable | — | 20 | 50 | μs | |
| 37 | T _{LVDT} | Low-Voltage Detect Pulse Width | 200 | — | — | μs | VDD ≤ VLVD |
| 38 | T _{CSD} | CPU Start-up Time | 5 | — | 10 | μs | |
| 39 | T _{IOBST} | Time for INTOSC to Stabilize | — | 1 | — | ms | |

Legend: TBD = To Be Determined

PIC18F2455/2550/4455/4550

FIGURE 28-9: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

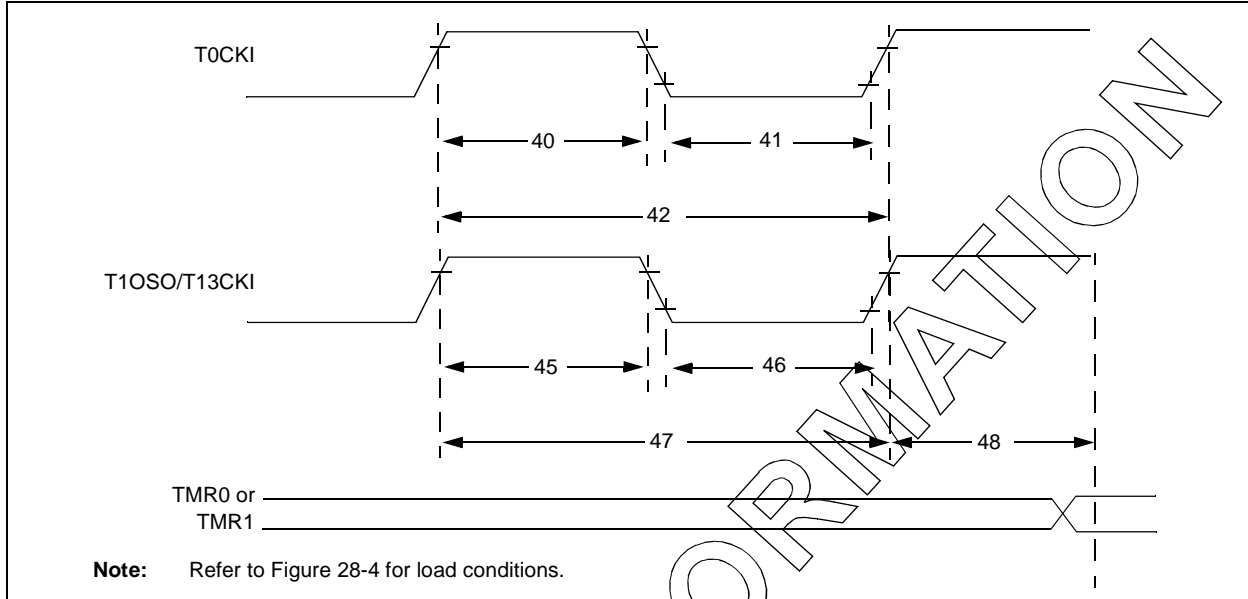


TABLE 28-13: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

| Param No. | Symbol | Characteristic | Min | Max | Units | Conditions | |
|-------------|-----------|--|-----------------------------|--|------------|------------|--|
| 40 | Tt0H | T0CKI High Pulse Width | No prescaler | $0.5 T_{CY} + 20$ | — | ns | |
| | | | With prescaler | 10 | — | ns | |
| 41 | Tt0L | T0CKI Low Pulse Width | No prescaler | $0.5 T_{CY} + 20$ | — | ns | |
| | | | With prescaler | 10 | — | ns | |
| 42 | Tt0P | T0CKI Period | No prescaler | $T_{CY} + 10$ | — | ns | |
| | | | With prescaler | Greater of: 20 ns or $(T_{CY} + 40)/N$ | — | ns | N = prescale value (1, 2, 4, ..., 256) |
| 45 | Tt1H | T13CKI High Time | Synchronous, no prescaler | $0.5 T_{CY} + 20$ | — | ns | |
| | | | Synchronous, with prescaler | PIC18FXXXX | 10 | — | ns |
| | | | | PIC18LFXXXX | 25 | — | ns |
| | | | Asynchronous | PIC18FXXXX | 30 | — | ns |
| PIC18LFXXXX | 50 | — | | ns | VDD = 2.0V | | |
| 46 | Tt1L | T13CKI Low Time | Synchronous, no prescaler | $0.5 T_{CY} + 5$ | — | ns | |
| | | | Synchronous, with prescaler | PIC18FXXXX | 10 | — | ns |
| | | | | PIC18LFXXXX | 25 | — | ns |
| | | | Asynchronous | PIC18FXXXX | 30 | — | ns |
| PIC18LFXXXX | 50 | — | | ns | VDD = 2.0V | | |
| 47 | Tt1P | T13CKI Input Period | Synchronous | Greater of: 20 ns or $(T_{CY} + 40)/N$ | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | Asynchronous | 60 | — | ns | |
| | Ft1 | T13CKI Oscillator Input Frequency Range | DC | 50 | kHz | | |
| 48 | Tcke2tmr1 | Delay from External T13CKI Clock Edge to Timer Increment | 2 TOSC | 7 TOSC | — | | |

PIC18F2455/2550/4455/4550

FIGURE 28-10: CAPTURE/COMPARE/PWM TIMINGS (ALL CCP MODULES)

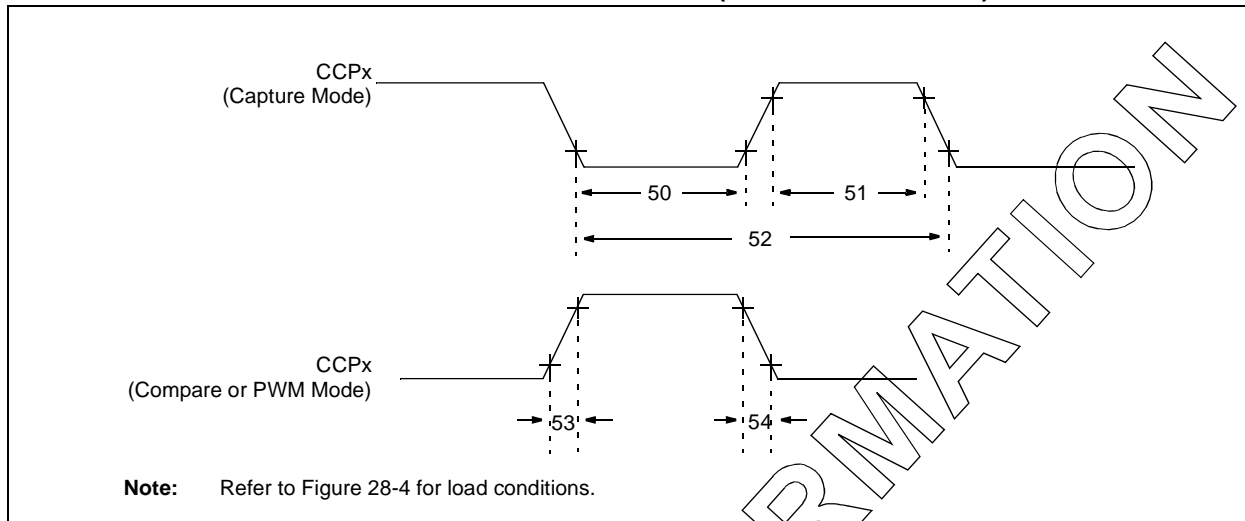


TABLE 28-14: CAPTURE/COMPARE/PWM REQUIREMENTS (ALL CCP MODULES)

| Param No. | Symbol | Characteristic | | Min | Max | Units | Conditions |
|-----------|--------|-----------------------|----------------|-----------------------------|----------|-------|---------------------------------|
| 50 | TccL | CCPx Input Low Time | No prescaler | $0.5 T_{CY} + 20$ | — | ns | |
| | | | With prescaler | PIC18FXXXXX PIC18LFXXXXX | 10 20 | — | |
| 51 | TccH | CCPx Input High Time | No prescaler | $0.5 T_{CY} + 20$ | — | ns | |
| | | | With prescaler | PIC18FXXXXX PIC18LFXXXXX | 10 20 | — | |
| 52 | TccP | CCPx Input Period | | $\frac{3 T_{CY} + 40}{N}$ | — | ns | N = prescale value (1, 4 or 16) |
| 53 | TccR | CCPx Output Fall Time | PIC18FXXXXX | — | 25 | ns | |
| | | | PIC18LFXXXXX | — | 45 | ns | |
| 54 | TccF | CCPx Output Fall Time | PIC18FXXXXX | — | 25 | ns | |
| | | | PIC18LFXXXXX | — | 45 | ns | |

PIC18F2455/2550/4455/4550

FIGURE 28-11: EXAMPLE SPI™ MASTER MODE TIMING (CKE = 0)

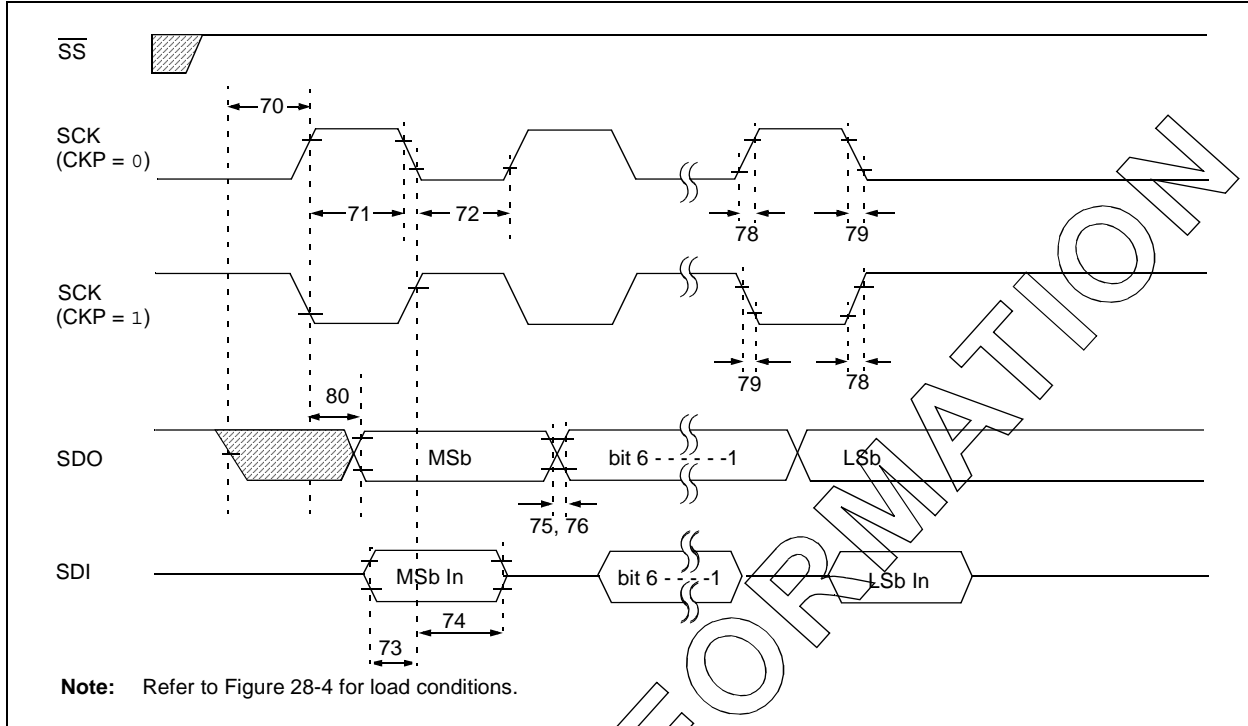


TABLE 28-15: EXAMPLE SPI™ MODE REQUIREMENTS (MASTER MODE, CKE = 0)

| Param No. | Symbol | Characteristic | Min | Max | Units | Conditions |
|-----------|-----------------------|---|--------------------------|---------------------------|-------|---------------------------|
| 70 | TssL2scH, TssL2scL | SS ↓ to SCK ↓ or SCK ↑ Input | T _{cy} | — | ns | |
| 71 | Tsch | SCK Input High Time | Continuous | 1.25 T _{cy} + 30 | — | ns |
| 71A | | (Slave mode) | Single Byte | 40 | — | ns (Note 1) |
| 72 | Tscl | SCK Input Low Time | Continuous | 1.25 T _{cy} + 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 T _{cy} + 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 |
| | | | PIC18LFXXXX | — | 45 | ns V _{DD} = 2.0V |
| 76 | TdoF | SDO Data Output Fall Time | — | 25 | ns | |
| 78 | TscR | SCK Output Rise Time (Master mode) | PIC18FXXXX | — | 25 | ns |
| | | | PIC18LFXXXX | — | 45 | ns V _{DD} = 2.0V |
| 79 | TscF | SCK Output Fall Time (Master mode) | — | 25 | ns | |
| 80 | Tsch2doV, TscL2doV | SDO Data Output Valid after SCK Edge | PIC18FXXXX | — | 50 | ns |
| | | | PIC18LFXXXX | — | 100 | ns V _{DD} = 2.0V |

Note 1: Requires the use of Parameter #73A.

Note 2: Only if Parameter #71A and #72A are used.

PIC18F2455/2550/4455/4550

FIGURE 28-12: EXAMPLE SPI™ MASTER MODE TIMING (CKE = 1)

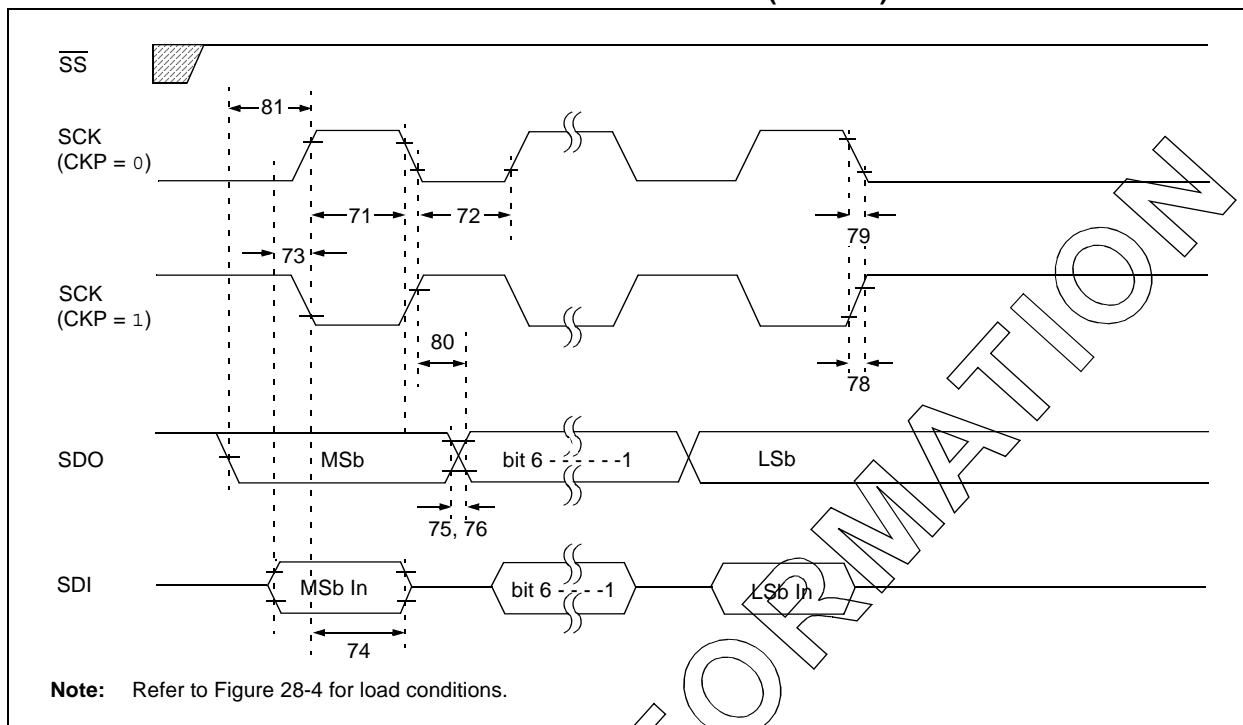


TABLE 28-16: EXAMPLE SPI™ MODE REQUIREMENTS (MASTER MODE, CKE = 1)

| Param. No. | Symbol | Characteristic | Min | Max | Units | Conditions | |
|------------|--------------------|---|--------------------------|---------------------------|-------|------------|------------------------|
| 71 | Tsch | SCK Input High Time (Slave mode) | Continuous | 1.25 T _{CY} + 30 | — | ns | |
| 71A | | Single Byte | 40 | — | ns | (Note 1) | |
| 72 | Tscl | SCK Input Low Time (Slave mode) | Continuous | 1.25 T _{CY} + 30 | — | ns | |
| 72A | | 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 T _{CY} + 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 | |
| | | | PIC18LFXXXX | — | 45 | ns | V _{DD} = 2.0V |
| 76 | TdoF | SDO Data Output Fall Time | — | 25 | ns | | |
| 78 | TsCR | SCK Output Rise Time (Master mode) | PIC18FXXXX | — | 25 | ns | |
| | | | PIC18LFXXXX | — | 45 | ns | V _{DD} = 2.0V |
| 79 | TscF | SCK Output Fall Time (Master mode) | — | 25 | ns | | |
| 80 | Tsch2doV, Tscl2doV | SDO Data Output Valid after SCK Edge | PIC18FXXXX | — | 50 | ns | |
| | | | PIC18LFXXXX | — | 100 | ns | V _{DD} = 2.0V |
| 81 | TdoV2scH, TdoV2scL | SDO Data Output Setup to SCK Edge | T _{CY} | — | ns | | |

Note 1: Requires the use of Parameter #73A.

2: Only if Parameter #71A and #72A are used.

PIC18F2455/2550/4455/4550

FIGURE 28-13: EXAMPLE SPI™ SLAVE MODE TIMING (CKE = 0)

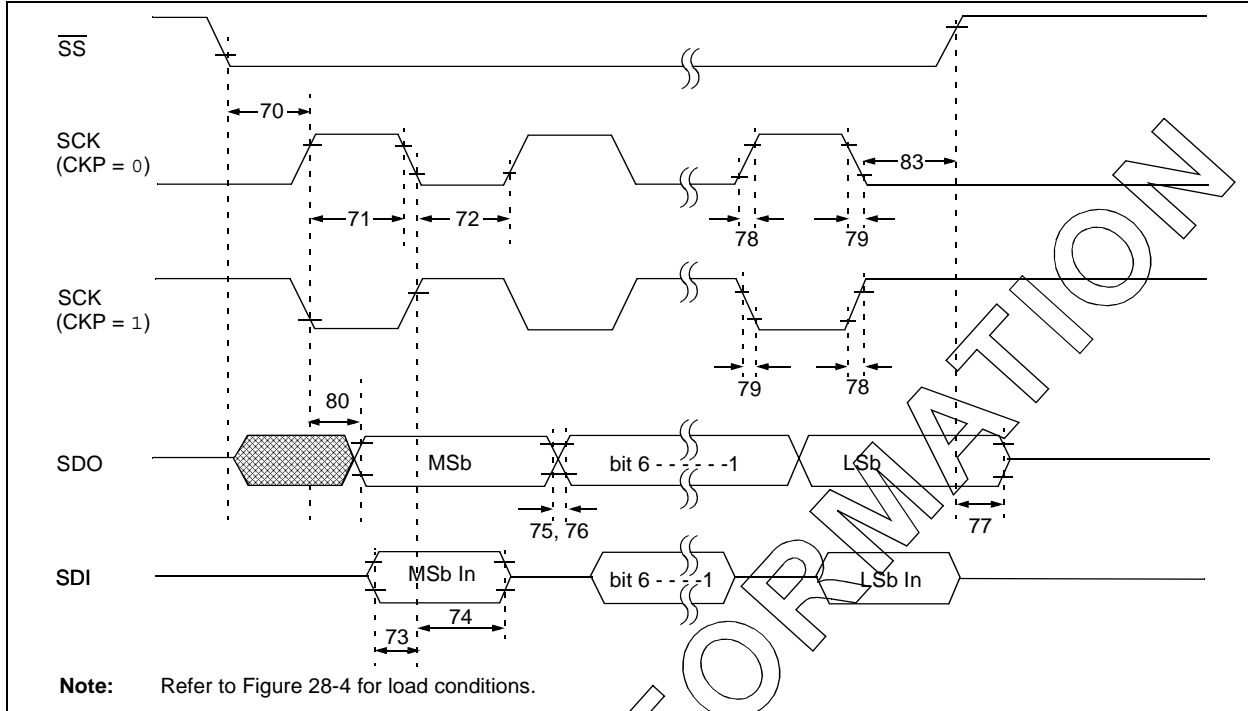


TABLE 28-17: EXAMPLE SPI™ MODE REQUIREMENTS (SLAVE MODE TIMING, CKE = 0)

| Param No. | Symbol | Characteristic | Min | Max | Units | Conditions |
|-----------|--------------------|---|--------------------------|---------------------------|-------|------------|
| 70 | Tssl2scH, Tssl2scL | SS ↓ to SCK ↓ or SCK ↑ Input | T _{CY} | — | ns | |
| 71 | Tsch | SCK Input High Time (Slave mode) | Continuous | 1.25 T _{CY} + 30 | — | ns |
| 71A | | | Single Byte | 40 | — | ns |
| 72 | TscL | SCK Input Low Time (Slave mode) | Continuous | 1.25 T _{CY} + 30 | — | ns |
| 72A | | | Single Byte | 40 | — | ns |
| 73 | TdiV2scH, TdiV2scL | Setup Time of SDI Data Input to SCK Edge | 100 | — | ns | |
| 73A | Tb2b | Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2 | 1.5 T _{CY} + 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 |
| 76 | | | PIC18LFXXXX | — | 45 | ns |
| 76 | TdoF | SDO Data Output Fall Time | — | 25 | ns | |
| 77 | Tssl2doZ | SS ↑ to SDO Output High-Impedance | 10 | 50 | ns | |
| 78 | TscR | SCK Output Rise Time (Master mode) | PIC18FXXXX | — | 25 | ns |
| 79 | | | PIC18LFXXXX | — | 45 | ns |
| 79 | TscF | SCK Output Fall Time (Master mode) | — | 25 | ns | |
| 80 | Tsch2doV, TscL2doV | SDO Data Output Valid after SCK Edge | PIC18FXXXX | — | 50 | ns |
| 83 | | | PIC18LFXXXX | — | 100 | ns |
| 83 | Tsch2ssH, TscL2ssH | SS ↑ after SCK edge | 1.5 T _{CY} + 40 | — | ns | |

Note 1: Requires the use of Parameter #73A.

Note 2: Only if Parameter #71A and #72A are used.

PIC18F2455/2550/4455/4550

FIGURE 28-14: EXAMPLE SPI™ SLAVE MODE TIMING (CKE = 1)

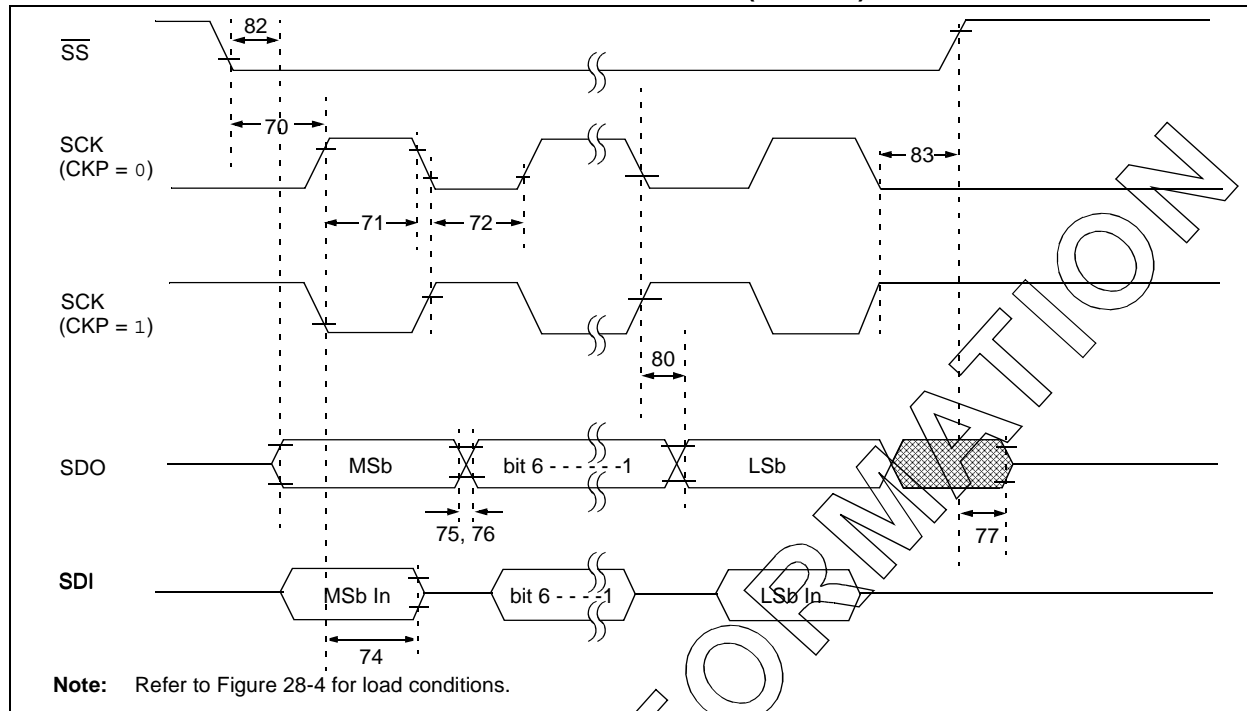


TABLE 28-18: EXAMPLE SPI™ SLAVE MODE REQUIREMENTS (CKE = 1)

| Param No. | Symbol | Characteristic | Min | Max | Units | Conditions |
|-----------|--------------------|---|--------------------------|---------------------------|-------|---------------------------|
| 70 | TssL2sch, TssL2scl | SS ↓ to SCK ↓ or SCK ↑ Input | T _{CY} | — | ns | |
| 71 | Tsch | SCK Input High Time (Slave mode) | Continuous | 1.25 T _{CY} + 30 | — | ns |
| 71A | | | Single Byte | 40 | — | ns (Note 1) |
| 72 | Tscl | SCK Input Low Time (Slave mode) | Continuous | 1.25 T _{CY} + 30 | — | ns |
| 72A | | | Single Byte | 40 | — | ns (Note 1) |
| 73A | Tb2b | Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2 | 1.5 T _{CY} + 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 |
| | | | PIC18LFXXXX | — | 45 | ns V _{DD} = 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 |
| | | | PIC18LFXXXX | — | 45 | ns V _{DD} = 2.0V |
| 79 | TscF | SCK Output Fall Time (Master mode) | — | 25 | ns | |
| 80 | Tsch2doV, TscL2doV | SDO Data Output Valid after SCK Edge | PIC18FXXXX | — | 50 | ns |
| | | | PIC18LFXXXX | — | 100 | ns V _{DD} = 2.0V |
| 82 | TssL2doV | SDO Data Output Valid after SS ↓ Edge | PIC18FXXXX | — | 50 | ns |
| | | | PIC18LFXXXX | — | 100 | ns V _{DD} = 2.0V |
| 83 | Tsch2ssH, TscL2ssH | SS ↑ after SCK Edge | 1.5 T _{CY} + 40 | — | ns | |

Note 1: Requires the use of Parameter #73A.

Note 2: Only if Parameter #71A and #72A are used.

PIC18F2455/2550/4455/4550

FIGURE 28-15: I²C™ BUS START/STOP BITS TIMING

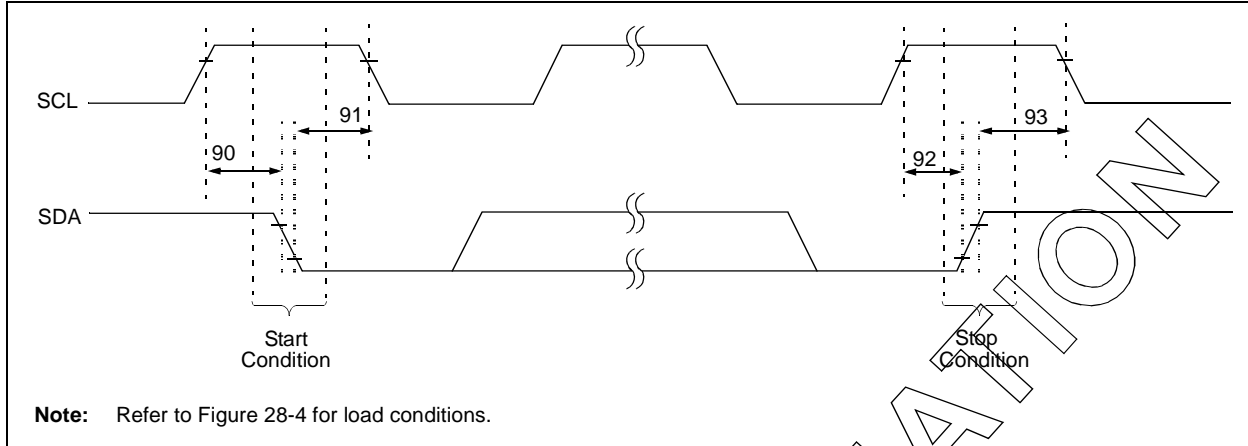
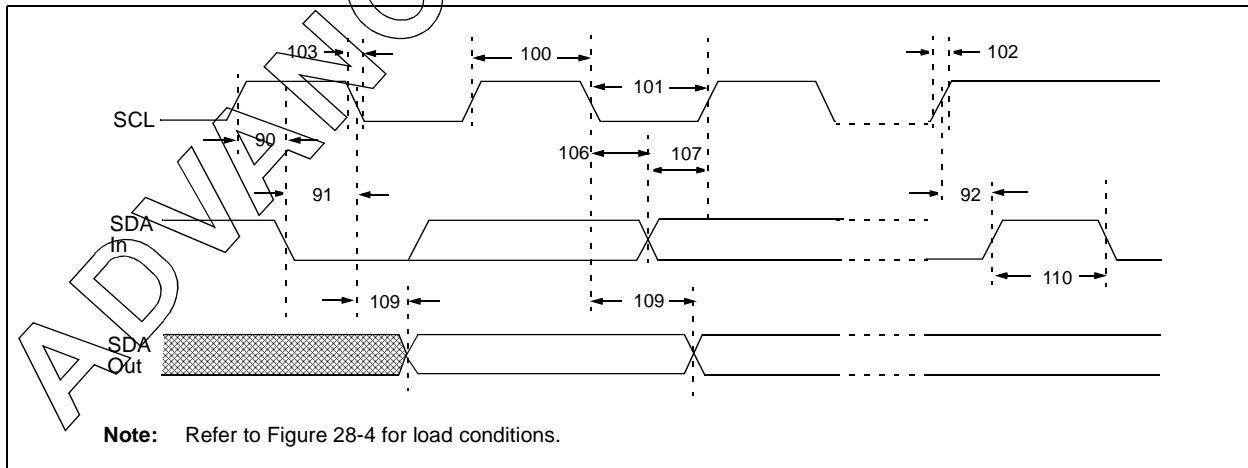


TABLE 28-19: I²C™ BUS START/STOP BITS REQUIREMENTS (SLAVE MODE)

| Param. No. | Symbol | Characteristic | Min | Max | Units | Conditions | |
|------------|---------|----------------------------|--------------|------|-------|------------|---|
| 90 | TSU:STA | Start Condition Setup Time | 100 kHz mode | 4700 | — | ns | Only relevant for Repeated Start condition |
| | | | 400 kHz mode | 600 | — | | |
| 91 | THD:STA | Start Condition Hold Time | 100 kHz mode | 4000 | — | ns | After this period, the first clock pulse is generated |
| | | | 400 kHz mode | 600 | — | | |
| 92 | TSU:STO | Stop Condition Setup Time | 100 kHz mode | 4700 | — | ns | |
| | | | 400 kHz mode | 600 | — | | |
| 93 | THD:STO | Stop Condition Hold Time | 100 kHz mode | 4000 | — | ns | |
| | | | 400 kHz mode | 600 | — | | |

FIGURE 28-16: I²C™ BUS DATA TIMING



PIC18F2455/2550/4455/4550

TABLE 28-20: I²C™ BUS DATA REQUIREMENTS (SLAVE MODE)

| Param. No. | Symbol | Characteristic | | Min | Max | Units | Conditions | |
|------------|----------------|----------------------------|--------------|-------------------------|------|-------|---|--|
| 100 | THIGH | Clock High Time | 100 kHz mode | 4.0 | — | μs | 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 |
| | | | SSP Module | 1.5 T _{CY} | — | | | |
| 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 |
| | | | SSP Module | 1.5 T _{CY} | — | | | |
| 102 | TR | SDA and SCL Rise Time | 100 kHz mode | — | 1000 | ns | C _B is specified to be from 10 to 400 pF | |
| | | | 400 kHz mode | 20 + 0.1 C _B | 300 | ns | | |
| 103 | TF | SDA and SCL Fall Time | 100 kHz mode | — | 300 | ns | C _B is specified to be from 10 to 400 pF | |
| | | | 400 kHz mode | 20 + 0.1 C _B | 300 | ns | | |
| 90 | TSU:STA | Start Condition Setup Time | 100 kHz mode | 4.7 | — | μs | Only relevant for Repeated Start condition | |
| | | | 400 kHz mode | 0.6 | — | μs | | |
| 91 | THD:STA | Start Condition Hold Time | 100 kHz mode | 4.0 | — | μs | After this period, the first clock pulse is generated | |
| | | | 400 kHz mode | 0.6 | — | μs | | |
| 106 | THD:DAT | Data Input Hold Time | 100 kHz mode | 0 | — | ns | | |
| | | | 400 kHz mode | 0 | 0.9 | μs | | |
| 107 | TSU:DAT | Data Input Setup Time | 100 kHz mode | 250 | — | ns | (Note 2) | |
| | | | 400 kHz mode | 100 | — | ns | | |
| 92 | TSU:STO | Stop Condition Setup Time | 100 kHz mode | 4.7 | — | μs | | |
| | | | 400 kHz mode | 0.6 | — | μs | | |
| 109 | TAA | Output Valid from Clock | 100 kHz mode | — | 3500 | ns | (Note 1) | |
| | | | 400 kHz mode | — | — | ns | | |
| 110 | TBUF | Bus Free Time | 100 kHz mode | 4.7 | — | μs | Time the bus must be free before a new transmission can start | |
| | | | 400 kHz mode | 1.3 | — | μs | | |
| D102 | C _B | Bus Capacitive Loading | | — | 400 | pF | | |

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

2: A fast mode 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, T_R max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the standard mode I²C bus specification), before the SCL line is released.

PIC18F2455/2550/4455/4550

FIGURE 28-17: MASTER SSP I²C™ BUS START/STOP BITS TIMING WAVEFORMS

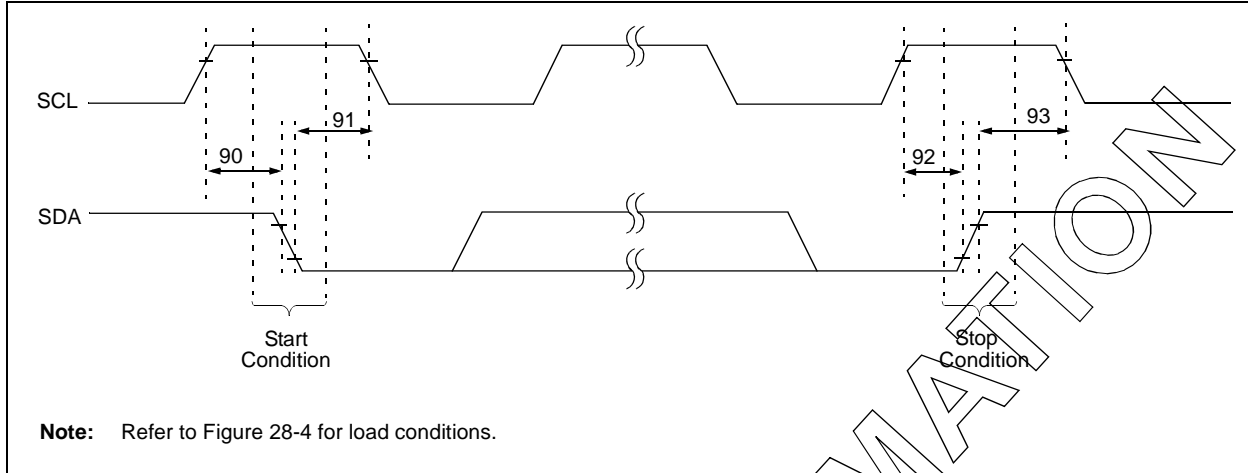
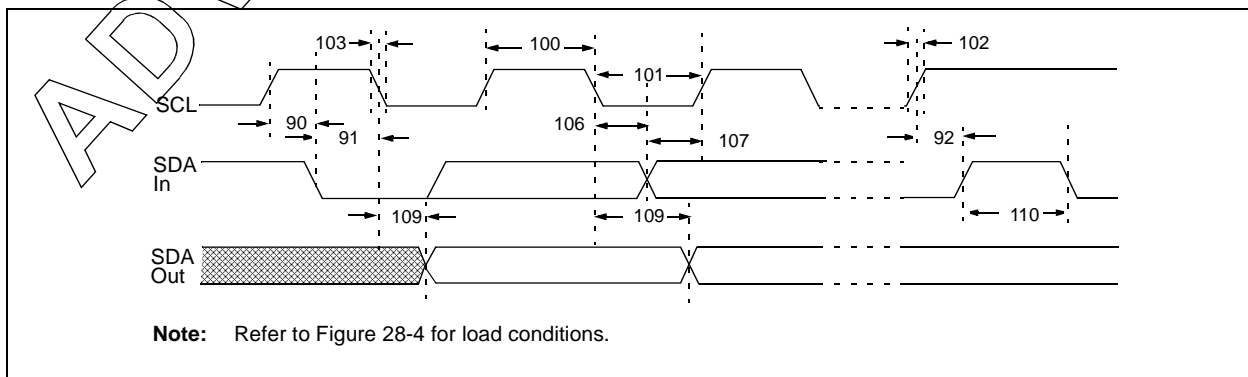


TABLE 28-21: MASTER SSP I²C™ BUS START/STOP BITS REQUIREMENTS

| Param. No. | Symbol | Characteristic | | Min | Max | Units | Conditions |
|------------|---------|----------------------------|---------------------------|-----------------------|-----|-------|---|
| 90 | TSU:STA | Start Condition Setup Time | 100 kHz mode | $2(T_{OSC})(BRG + 1)$ | — | ns | Only relevant for Repeated Start condition |
| | | | 400 kHz mode | $2(T_{OSC})(BRG + 1)$ | — | | |
| | | | 1 MHz mode ⁽¹⁾ | $2(T_{OSC})(BRG + 1)$ | — | | |
| 91 | THD:STA | Start Condition Hold Time | 100 kHz mode | $2(T_{OSC})(BRG + 1)$ | — | ns | After this period, the first clock pulse is generated |
| | | | 400 kHz mode | $2(T_{OSC})(BRG + 1)$ | — | | |
| | | | 1 MHz mode ⁽¹⁾ | $2(T_{OSC})(BRG + 1)$ | — | | |
| 92 | TSU:STO | Stop Condition Setup Time | 100 kHz mode | $2(T_{OSC})(BRG + 1)$ | — | ns | |
| | | | 400 kHz mode | $2(T_{OSC})(BRG + 1)$ | — | | |
| | | | 1 MHz mode ⁽¹⁾ | $2(T_{OSC})(BRG + 1)$ | — | | |
| 93 | THD:STO | Stop Condition Hold Time | 100 kHz mode | $2(T_{OSC})(BRG + 1)$ | — | ns | |
| | | | 400 kHz mode | $2(T_{OSC})(BRG + 1)$ | — | | |
| | | | 1 MHz mode ⁽¹⁾ | $2(T_{OSC})(BRG + 1)$ | — | | |

Note 1: Maximum pin capacitance = 10 pF for all I²C pins.

FIGURE 28-18: MASTER SSP I²C™ BUS DATA TIMING



PIC18F2455/2550/4455/4550

TABLE 28-22: MASTER SSP I²C™ BUS DATA REQUIREMENTS

| Param. No. | Symbol | Characteristic | Min | Max | Units | Conditions |
|------------|---------|----------------------------|---------------------------|-----------------------|-------|------------|
| 100 | THIGH | Clock High Time | 100 kHz mode | $2(T_{osc})(BRG + 1)$ | — | ms |
| | | | 400 kHz mode | $2(T_{osc})(BRG + 1)$ | — | ms |
| | | | 1 MHz mode ⁽¹⁾ | $2(T_{osc})(BRG + 1)$ | — | ms |
| 101 | TLOW | Clock Low Time | 100 kHz mode | $2(T_{osc})(BRG + 1)$ | — | ms |
| | | | 400 kHz mode | $2(T_{osc})(BRG + 1)$ | — | ms |
| | | | 1 MHz mode ⁽¹⁾ | $2(T_{osc})(BRG + 1)$ | — | ms |
| 102 | TR | SDA and SCL Rise Time | 100 kHz mode | — | 1000 | ns |
| | | | 400 kHz mode | $20 + 0.1 C_B$ | 300 | ns |
| | | | 1 MHz mode ⁽¹⁾ | — | 300 | ns |
| 103 | TF | SDA and SCL Fall Time | 100 kHz mode | — | 300 | ns |
| | | | 400 kHz mode | $20 + 0.1 C_B$ | 300 | ns |
| | | | 1 MHz mode ⁽¹⁾ | — | 100 | ns |
| 90 | TSU:STA | Start Condition Setup Time | 100 kHz mode | $2(T_{osc})(BRG + 1)$ | — | ms |
| | | | 400 kHz mode | $2(T_{osc})(BRG + 1)$ | — | ms |
| | | | 1 MHz mode ⁽¹⁾ | $2(T_{osc})(BRG + 1)$ | — | ms |
| 91 | THD:STA | Start Condition Hold Time | 100 kHz mode | $2(T_{osc})(BRG + 1)$ | — | ms |
| | | | 400 kHz mode | $2(T_{osc})(BRG + 1)$ | — | ms |
| | | | 1 MHz mode ⁽¹⁾ | $2(T_{osc})(BRG + 1)$ | — | ms |
| 106 | THD:DAT | Data Input Hold Time | 100 kHz mode | 0 | — | ns |
| | | | 400 kHz mode | 0 | 0.9 | ms |
| 107 | TSU:DAT | Data Input Setup Time | 100 kHz mode | 250 | — | ns |
| | | | 400 kHz mode | 100 | — | ns |
| 92 | TSU:STO | Stop Condition Setup Time | 100 kHz mode | $2(T_{osc})(BRG + 1)$ | — | ms |
| | | | 400 kHz mode | $2(T_{osc})(BRG + 1)$ | — | ms |
| | | | 1 MHz mode ⁽¹⁾ | $2(T_{osc})(BRG + 1)$ | — | ms |
| 109 | TAA | Output Valid from Clock | 100 kHz mode | — | 3500 | ns |
| | | | 400 kHz mode | — | 1000 | ns |
| | | | 1 MHz mode ⁽¹⁾ | — | — | ns |
| 110 | TBUF | Bus Free Time | 100 kHz mode | 4.7 | — | ms |
| | | | 400 kHz mode | 1.3 | — | ms |
| D102 | CB | Bus Capacitive Loading | — | 400 | pF | |

Note 1: Maximum pin capacitance = 10 pF for all I²C pins.

2: A Fast mode I²C bus device can be used in a standard mode I²C bus system but, parameter #107 ≥ 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.

PIC18F2455/2550/4455/4550

FIGURE 28-19: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

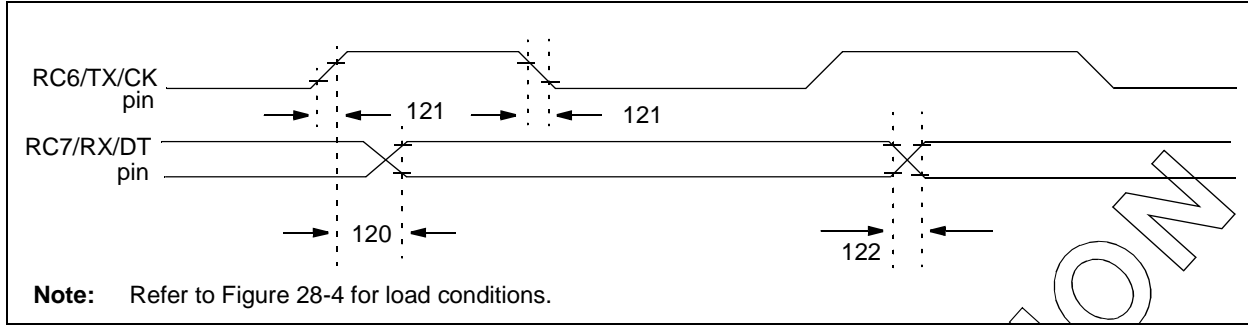


TABLE 28-23: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS

| Param No. | Symbol | Characteristic | Min | Max | Units | Conditions | |
|-----------|----------|--|-------------|-----|-------|------------|------------|
| 120 | TckH2dtV | SYNC XMIT (MASTER & SLAVE) Clock High to Data Out Valid | PIC18FXXXX | — | 40 | ns | VDD = 2.0V |
| | | | PIC18LFXXXX | — | 100 | ns | |
| 121 | Tckrf | Clock Out Rise Time and Fall Time (Master mode) | PIC18FXXXX | — | 20 | ns | VDD = 2.0V |
| | | | PIC18LFXXXX | — | 50 | ns | |
| 122 | Tdtrf | Data Out Rise Time and Fall Time | PIC18FXXXX | — | 20 | ns | VDD = 2.0V |
| | | | PIC18LFXXXX | — | 50 | ns | |

FIGURE 28-20: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING

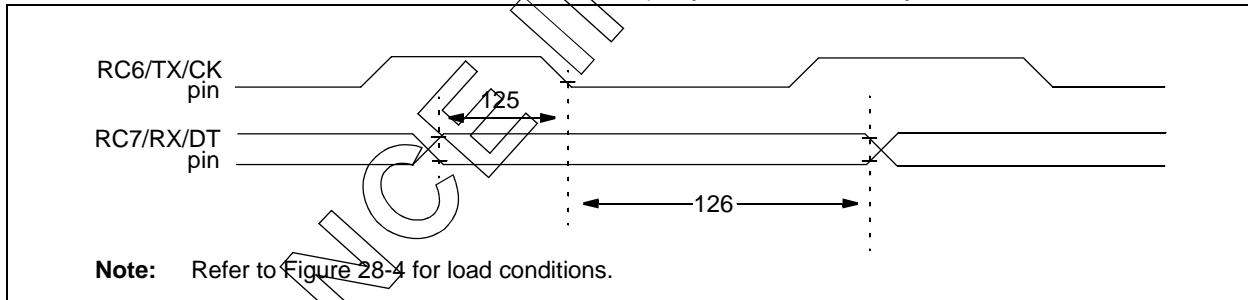


TABLE 28-24: USART SYNCHRONOUS RECEIVE REQUIREMENTS

| 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 | |

PIC18F2455/2550/4455/4550

FIGURE 28-21: USB SIGNAL TIMING

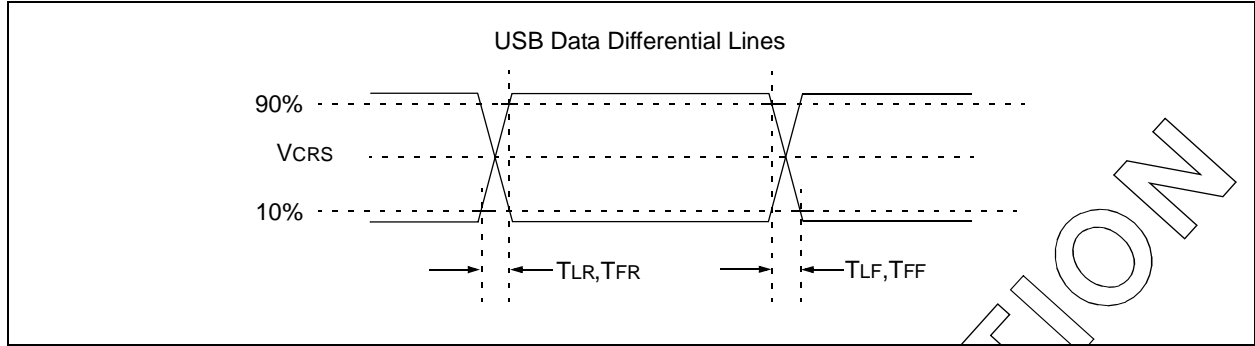


TABLE 28-25: USB LOW-SPEED TIMING REQUIREMENTS

| Param No. | Symbol | Characteristic | Min | Typ | Max | Units | Conditions |
|-----------|-------------------|-------------------------|-----|-----|-----|-------|--------------------|
| | T _{LR} | Transition Rise Time | 75 | — | 300 | ns | CL = 200 to 600 pF |
| | T _{LF} | Transition Fall Time | 75 | — | 300 | ns | CL = 200 to 600 pF |
| | T _{LRFM} | Rise/Fall Time Matching | 80 | — | 125 | % | |

TABLE 28-26: USB FULL SPEED REQUIREMENTS

| Param No. | Symbol | Characteristic | Min | Typ | Max | Units | Conditions |
|-----------|-------------------|-------------------------|-----|-----|-------|-------|------------|
| | T _{FR} | Transition Rise Time | 4 | — | 20 | ns | CL = 50 pF |
| | T _{FF} | Transition Fall Time | 4 | — | 20 | ns | CL = 50 pF |
| | T _{FRFM} | Rise/Fall Time Matching | 90 | — | 111.1 | % | |

PIC18F2455/2550/4455/4550

FIGURE 28-22: STREAMING PARALLEL PORT TIMING (PIC18F4455/4550)

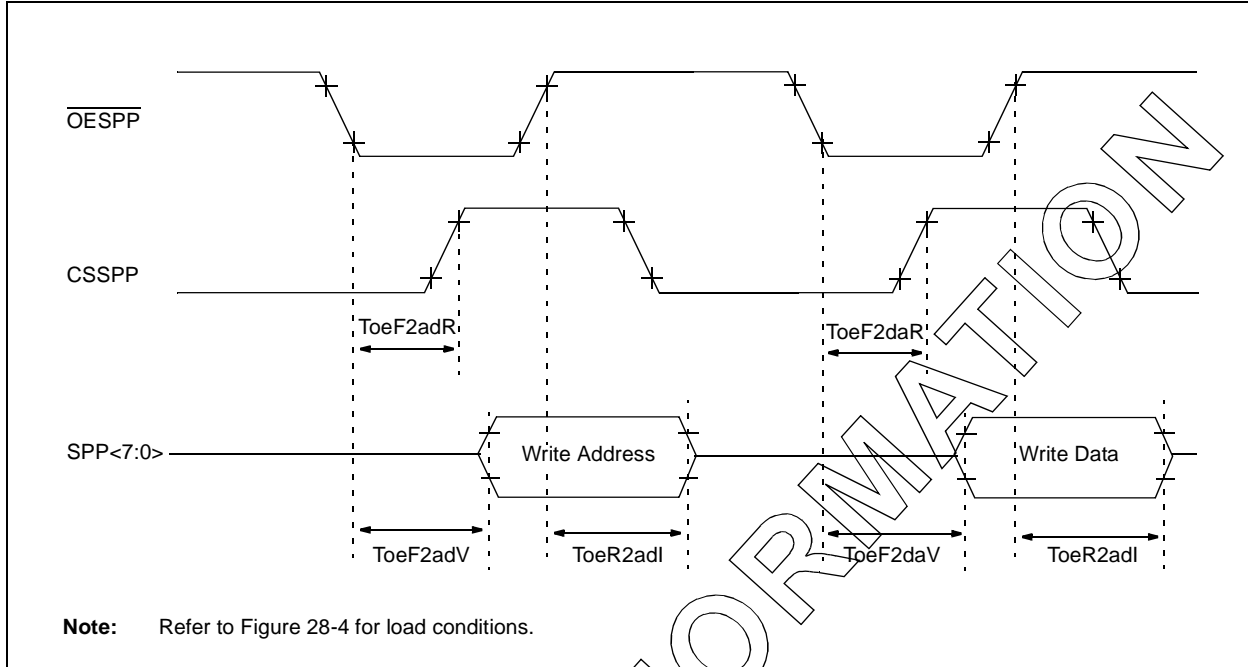


TABLE 28-27: PARALLEL SLAVE PORT REQUIREMENTS (PIC18F4455/4550)

| Param. No. | Symbol | Characteristic | Min | Max | Units | Conditions |
|------------|----------|--|-----|-----|-------|------------|
| | ToeF2adR | $\overline{\text{OESPP}}$ Falling Edge to CSSPP Rising Edge, Address Out | 0 | 5 | ns | |
| | ToeF2adV | $\overline{\text{OESPP}}$ Falling Edge to Address Out Valid | 0 | 5 | ns | |
| | ToeR2adI | $\overline{\text{OESPP}}$ Rising Edge to Address Out Invalid | 0 | 5 | ns | |
| | ToeF2daR | $\overline{\text{OESPP}}$ Falling Edge to CSSPP Rising Edge, Data Out | 0 | 5 | ns | |
| | ToeF2daV | $\overline{\text{OESPP}}$ Falling Edge to Address Out Valid | 0 | 5 | ns | |
| | ToeR2daI | $\overline{\text{OESPP}}$ Rising Edge to Data Out Invalid | 0 | 5 | ns | |

PIC18F2455/2550/4455/4550

**TABLE 28-28: A/D CONVERTER CHARACTERISTICS: PIC18F2455/2550/4455/4550 (INDUSTRIAL)
PIC18LF2455/2550/4455/4550 (INDUSTRIAL)**

| Param No. | Symbol | Characteristic | Min | Typ | Max | Units | Conditions | |
|-----------|-------------------|--|---------------------------|-----|----------------------|--------------------|--|--|
| A01 | NR | Resolution | — | — | 10 | bit | $\Delta V_{REF} \geq 3.0V$ | |
| A03 | EIL | Integral Linearity Error | — | — | $<\pm 1$ | LSb | $\Delta V_{REF} \geq 3.0V$ | |
| A04 | EDL | Differential Linearity Error | — | — | $<\pm 1$ | LSb | $\Delta V_{REF} \geq 3.0V$ | |
| A06 | E _{OFF} | Offset Error | — | — | $<\pm 1$ | LSb | $\Delta V_{REF} \geq 3.0V$ | |
| A07 | E _{GN} | Gain Error | — | — | $<\pm 1$ | LSb | $\Delta V_{REF} \geq 3.0V$ | |
| A10 | — | Monotonicity | Guaranteed ⁽¹⁾ | | | | | |
| A20 | ΔV_{REF} | Reference Voltage Range (V _{REFH} – V _{REFL}) | 3 | — | $AV_{DD} - AV_{SS}$ | V | For 10-bit resolution | |
| A21 | V _{REFH} | Reference Voltage High | $AV_{SS} + 3.0V$ | — | $AV_{DD} + 0.3V$ | V | For 10-bit resolution | |
| A22 | V _{REFL} | Reference Voltage Low | $AV_{SS} - 0.3V$ | — | $AV_{DD} - 3.0V$ | V | For 10-bit resolution | |
| A25 | V _{AIN} | Analog Input Voltage | V _{REFL} | — | V _{REFH} | V | | |
| A28 | AV _{DD} | Analog Supply Voltage | $V_{DD} - 0.3$ | — | $V_{DD} + 0.3$ | V | | |
| A29 | AV _{SS} | Analog Supply Voltage | $V_{SS} - 0.3$ | — | $V_{SS} + 0.3$ | V | | |
| A30 | Z _{AIN} | Recommended Impedance of Analog Voltage Source | — | — | 2.5 | k Ω | | |
| A40 | I _{AD} | A/D Conversion Current (V _{DD}) | PIC18FXXXX | 180 | — | μA | Average current consumption when A/D is on (Note 2) | |
| | | | PIC18LFXXXX | — | 90 | — | μA | V _{DD} = 2.0V; Average current consumption when A/D is on (Note 2) |
| A50 | I _{REF} | V _{REF} Input Current (Note 3) | — | — | ± 5 ± 150 | μA μA | During V _{AIN} acquisition. During A/D conversion cycle. | |

Note 1: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

2: When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.

3: V_{REFH} current is from RA3/AN3/V_{REF+} pin or AV_{DD}, whichever is selected as the V_{REFH} source.
V_{REFL} current is from RA2/AN2/V_{REF-}/CV_{REF} pin or AV_{SS}, whichever is selected as the V_{REFL} source.

PIC18F2455/2550/4455/4550

FIGURE 28-23: A/D CONVERSION TIMING

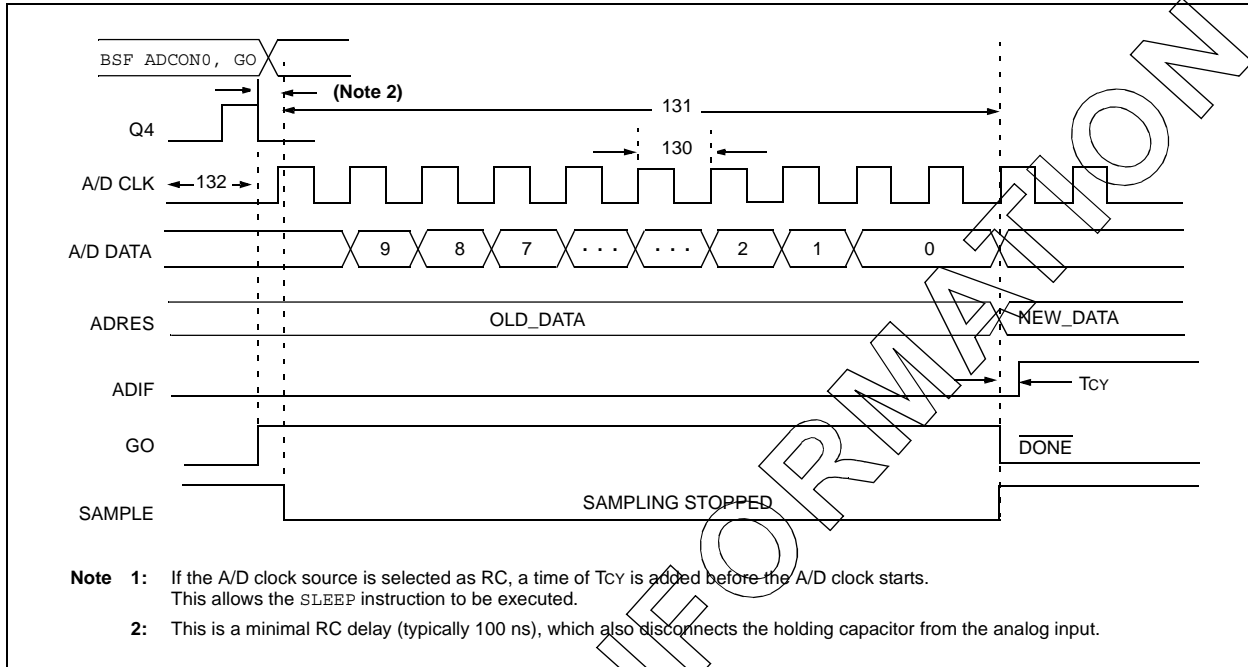


TABLE 28-29: A/D CONVERSION REQUIREMENTS

| Param No. | Symbol | Characteristic | Min | Max | Units | Conditions | |
|-----------|------------------|--|-------------|-----------------|---------------------|---|--|
| 130 | TAD | A/D Clock Period | PIC18FXXXX | 0.7 | 25.0 ⁽¹⁾ | μs | TOSC based, $V_{REF} \geq 3.0\text{V}$ |
| | | | PIC18LFXXXX | TBD | TBD | μs | $V_{DD} = 2.0\text{V}$; TOSC based, V_{REF} full range |
| | | PIC18FXXXX | TBD | TBD | μs | A/D RC mode | |
| | | PIC18LFXXXX | TBD | TBD | μs | $V_{DD} = 2.0\text{V}$; A/D RC mode | |
| 131 | T _{CVN} | Conversion Time (not including acquisition time) (Note 2) | 11 | 12 | TAD | | |
| 132 | T _{ACQ} | Acquisition Time (Note 3) | 1.4 | — | μs | -40°C to +85°C | |
| | | | TBD | — | μs | 0°C ≤ to ≤ +85°C | |
| 135 | T _{SWC} | Switching Time from Convert → Sample | — | (Note 4) | | | |
| TBD | T _{DIS} | Discharge Time | 0.2 | — | μs | | |

- Note 1:** The time of the A/D clock period is dependent on the device frequency and the TAD clock divider.
- Note 2:** ADRES register may be read on the following T_{CY} cycle.
- Note 3:** The time for the holding capacitor to acquire the "New" input voltage when the voltage changes full scale after the conversion (AV_{DD} to AV_{SS} or AV_{SS} to AV_{DD}). The source impedance (R_s) on the input channels is 50 Ω .
- Note 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.

PIC18F2455/2550/4455/4550

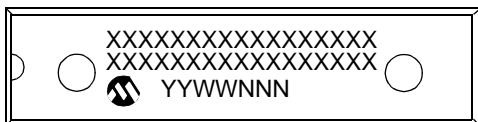
NOTES:

PIC18F2455/2550/4455/4550

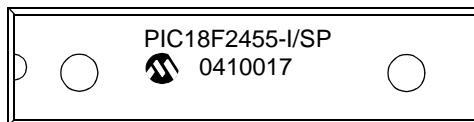
30.0 PACKAGING INFORMATION

30.1 Package Marking Information

28-Lead PDIP (Skinny DIP)



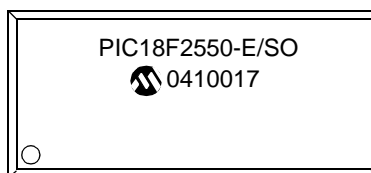
Example



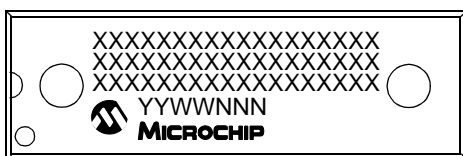
28-Lead SOIC



Example



40-Lead PDIP



Example



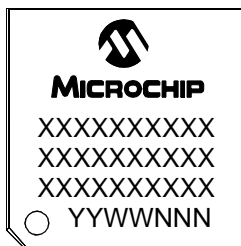
| | | |
|----------------|--|--|
| Legend: | XX...X | Customer specific information* |
| | Y | Year code (last digit of calendar year) |
| | YY | Year code (last 2 digits of calendar year) |
| | WW | Week code (week of January 1 is week '01') |
| | NNN | Alphanumeric traceability code |
| Note: | In 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. | |

* Standard PICmicro device marking consists of Microchip part number, year code, week code, and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

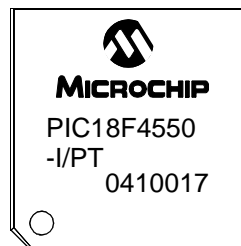
PIC18F2455/2550/4455/4550

Package Marking Information (Continued)

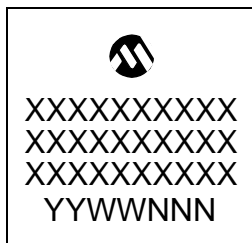
44-Lead TQFP



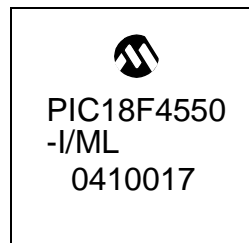
Example



44-Lead QFN



Example

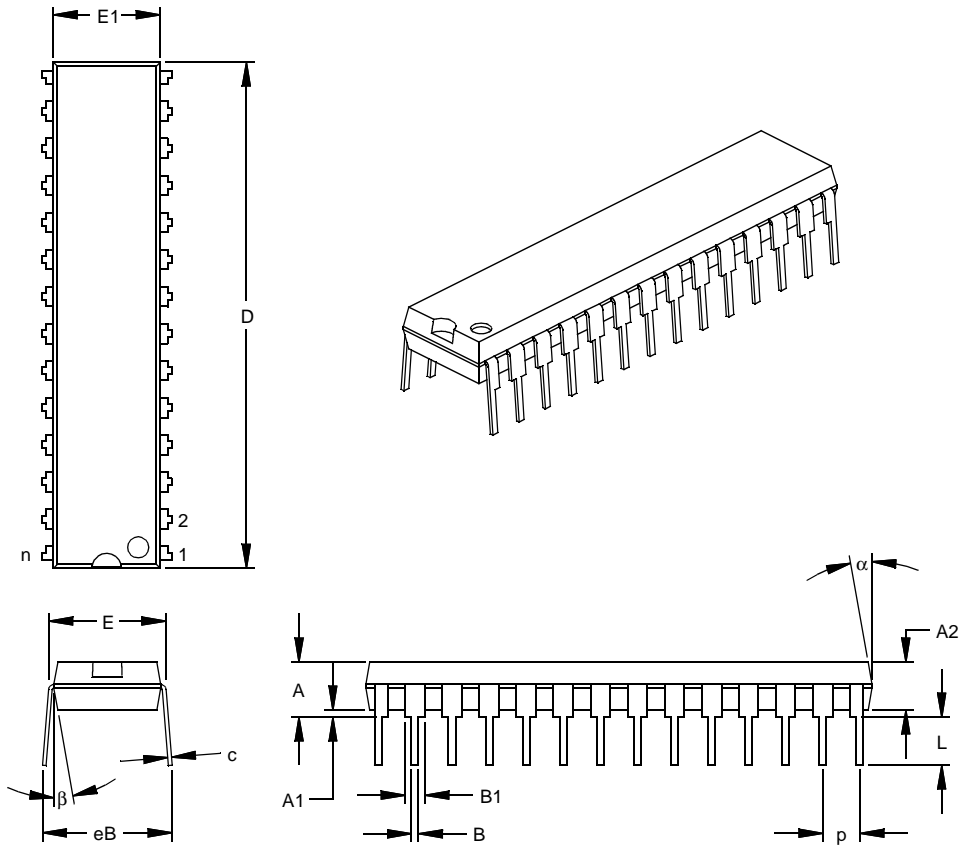


PIC18F2455/2550/4455/4550

30.2 Package Details

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 | | |
|----------------------------|------|-------|---------|-------|-------|-------------|-------|-------|
| Dimension Limits | | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | | 28 | | | 28 | |
| Pitch | p | | | .100 | | | 2.54 | |
| Top to Seating Plane | A | | .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 | c | | .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 | B | | .016 | .019 | .022 | 0.41 | 0.48 | 0.56 |
| Overall Row Spacing | § eB | | .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 mold flash or protrusions. Mold flash or protrusions shall not exceed

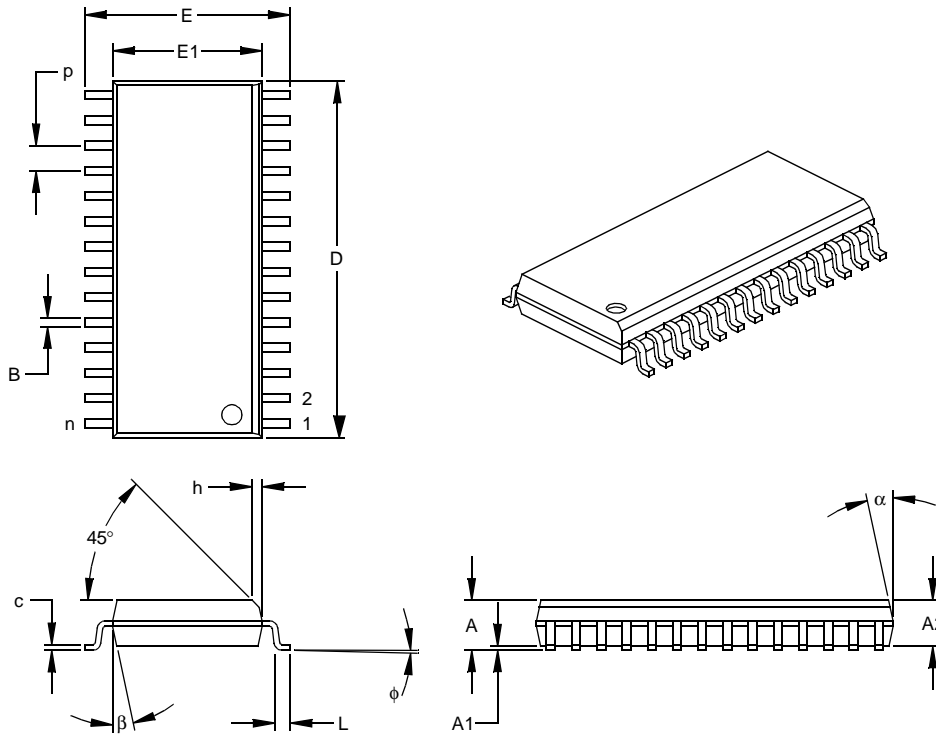
.010" (0.254mm) per side.

JEDEC Equivalent: MO-095

Drawing No. C04-070

PIC18F2455/2550/4455/4550

28-Lead Plastic Small Outline (SO) – Wide, 300 mil Body (SOIC)



| Dimension Limits | Units | INCHES* | | | MILLIMETERS | | |
|--------------------------|-------|---------|------|------|-------------|-------|-------|
| | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 28 | | | 28 | |
| Pitch | p | | .050 | | | 1.27 | |
| Overall Height | A | .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 | E | .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 | φ | 0 | 4 | 8 | 0 | 4 | 8 |
| Lead Thickness | c | .009 | .011 | .013 | 0.23 | 0.28 | 0.33 |
| Lead Width | B | .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 |

* Controlling Parameter
 § Significant Characteristic

Notes:

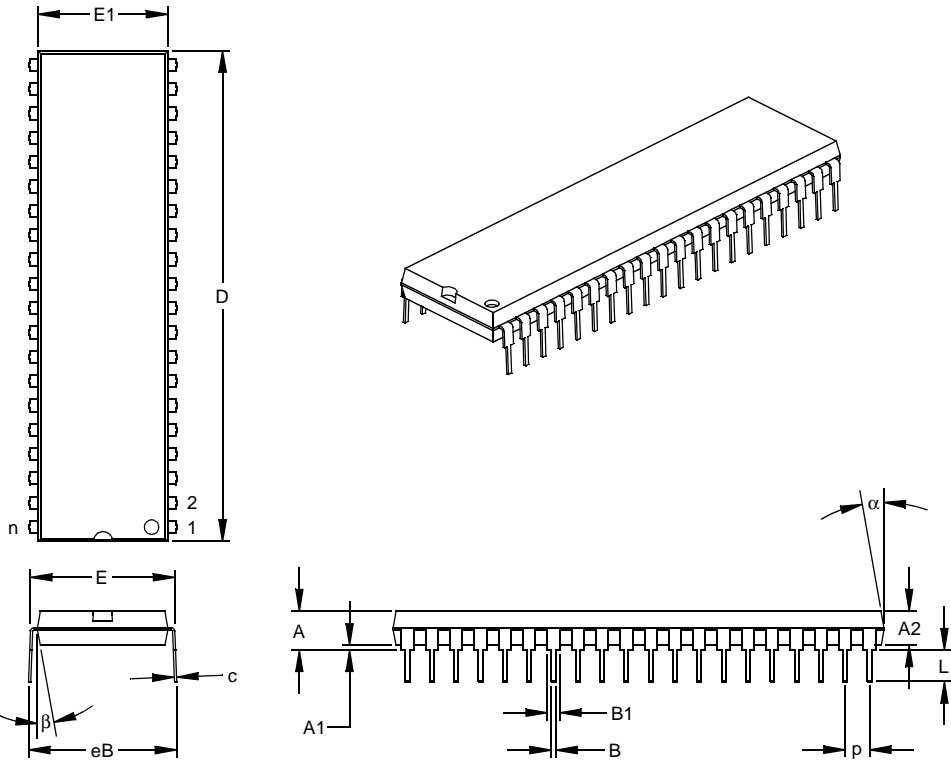
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-013

Drawing No. C04-052

PIC18F2455/2550/4455/4550

40-Lead Plastic Dual In-line (P) – 600 mil Body (PDIP)



| Units | | INCHES* | | | MILLIMETERS | | |
|----------------------------|------|---------|-------|-------|-------------|-------|-------|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 40 | | | 40 | |
| Pitch | p | | .100 | | | 2.54 | |
| Top to Seating Plane | A | .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 | E | .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 | c | .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 | B | .014 | .018 | .022 | 0.36 | 0.46 | 0.56 |
| Overall Row Spacing | § eB | .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 |

* Controlling Parameter

§ Significant Characteristic

Notes:

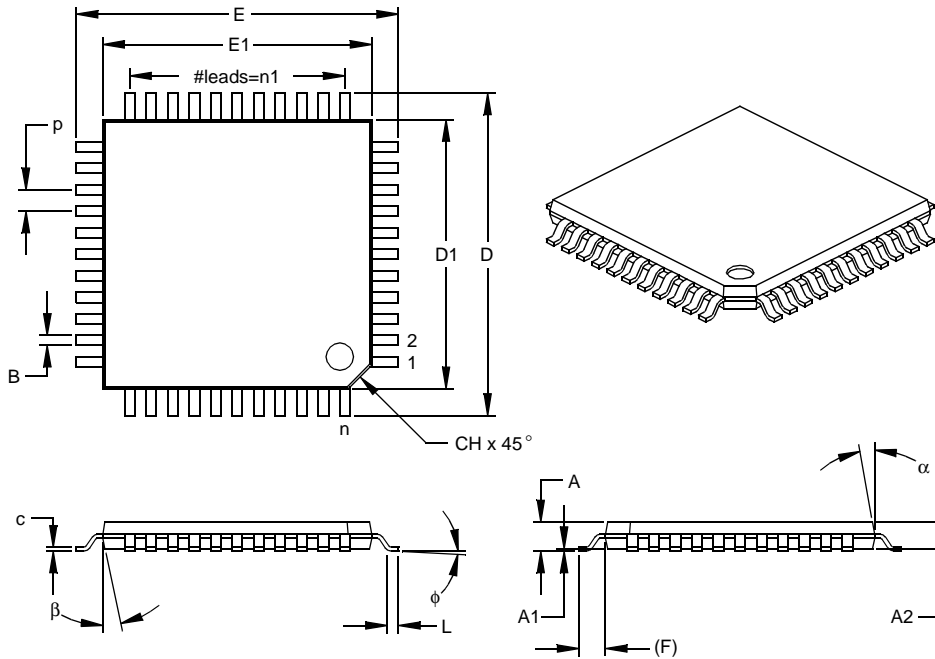
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MO-011

Drawing No. C04-016

PIC18F2455/2550/4455/4550

44-Lead Plastic Thin Quad Flatpack (PT) 10x10x1 mm Body, 1.0/0.10 mm Lead Form (TQFP)



| Dimension Limits | Units | INCHES | | | MILLIMETERS* | | |
|--------------------------|-------|--------|------|------|--------------|-------|-------|
| | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 44 | | | 44 | |
| Pitch | p | | .031 | | | 0.80 | |
| Pins per Side | n1 | | 11 | | | 11 | |
| Overall Height | A | .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 | | 1.00 | | |
| Foot Angle | φ | 0 | 3.5 | 7 | 0 | 3.5 | 7 |
| Overall Width | E | .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 | c | .004 | .006 | .008 | 0.09 | 0.15 | 0.20 |
| Lead Width | B | .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 |

* Controlling Parameter

§ Significant Characteristic

Notes:

Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed

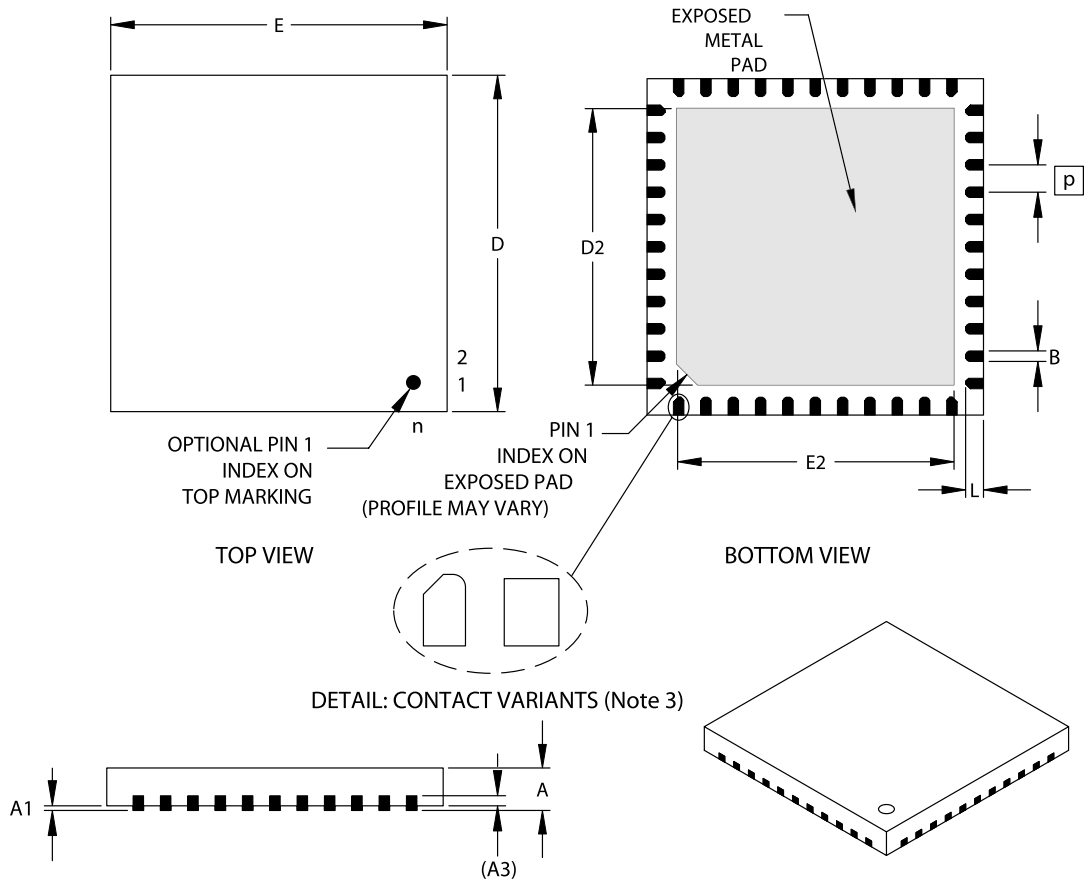
.010" (0.254mm) per side.

JEDEC Equivalent: MS-026

Drawing No. C04-076

PIC18F2455/2550/4455/4550

44-Lead Plastic Quad Flat No Lead Package (ML) 8x8 mm Body (QFN)



| Dimension Limits | Units | INCHES | | | MILLIMETERS* | | |
|--------------------|-------|-----------------------|------|------|-----------------------|------|------|
| | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Contacts | n | | 44 | | | 44 | |
| Pitch | p | .026 BSC ¹ | | | 0.65 BSC ¹ | | |
| Overall Height | A | .031 | .035 | .039 | 0.80 | 0.90 | 1.00 |
| Standoff | A1 | .000 | .001 | .002 | 0 | 0.02 | 0.05 |
| Base Thickness | (A3) | .010 REF ² | | | 0.25 REF ² | | |
| Overall Width | E | .309 | .315 | .321 | 7.85 | 8.00 | 8.15 |
| Exposed Pad Width | E2 | .246 | .268 | .274 | 6.25 | 6.80 | 6.95 |
| Overall Length | D | .309 | .315 | .321 | 7.85 | 8.00 | 8.15 |
| Exposed Pad Length | D2 | .246 | .268 | .274 | 6.25 | 6.80 | 6.95 |
| Contact Width | B | .008 | .013 | .013 | 0.20 | 0.33 | 0.35 |
| Contact Length | L | .014 | .016 | .019 | 0.35 | 0.40 | 0.48 |

*Controlling Parameter

Notes:

1. BSC: Basic Dimension. Theoretically exact value shown without tolerances. See ASME Y14.5M
2. REF: Reference Dimension, usually without tolerance, for information purposes only. See ASME Y14.5M
3. Contact profiles may vary.

JEDEC equivalent: M0-220
Drawing No. C04-103

PIC18F2455/2550/4455/4550

NOTES:

PIC18F2455/2550/4455/4550

APPENDIX A: REVISION HISTORY

Revision A (May 2004)

Original data sheet for PIC18F2455/2550/4455/4550 devices.

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 |

PIC18F2455/2550/4455/4550

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.

PIC18F2455/2550/4455/4550

NOTES:

PIC18F2455/2550/4455/4550

INDEX

A

| | |
|--|----------|
| A/D | 253 |
| A/D Converter Interrupt, Configuring | 257 |
| Acquisition Requirements | 258 |
| ADCON0 Register | 253 |
| ADCON1 Register | 253 |
| ADCON2 Register | 253 |
| ADRESH Register | 253, 256 |
| ADRESL Register | 253 |
| Analog Port Pins, Configuring | 260 |
| Associated Registers | 262 |
| Calculating the Minimum Required | |
| Acquisition Time | 258 |
| Configuring the Module | 257 |
| Conversion Clock (TAD) | 259 |
| Conversion Status (GO/DONE Bit) | 256 |
| Conversions | 261 |
| Converter Characteristics | 393 |
| Discharge | 261 |
| Operation in Power Managed Modes | 260 |
| Selecting and Configuring | |
| Acquisition Time | 259 |
| Special Event Trigger (CCP2) | 262 |
| Special Event Trigger (ECCP) | 150 |
| Absolute Maximum Ratings | 357 |
| AC (Timing) Characteristics | 374 |
| Load Conditions for Device | |
| Timing Specifications | 375 |
| Parameter Symbology | 374 |
| Temperature and Voltage Specifications | 375 |
| Timing Conditions | 375 |
| AC Characteristics | |
| Internal RC Accuracy | 377 |
| Access Bank | 65 |
| Mapping with Indexed Literal | |
| Offset Mode | 77 |
| ACKSTAT | 223 |
| ACKSTAT Status Flag | 223 |
| ADCON0 Register | 253 |
| GO/DONE Bit | 256 |
| ADCON1 Register | 253 |
| ADCON2 Register | 253 |
| ADDFSR | 344 |
| ADDLW | 307 |
| ADDLWLNK | 344 |
| ADDWF | 307 |
| ADDWFC | 308 |
| ADRESH Register | 253 |
| ADRESL Register | 253, 256 |
| Analog-to-Digital Converter. <i>See</i> A/D. | |
| and BSR | 77 |
| ANDLW | 308 |
| ANDWF | 309 |
| Assembler | |
| MPASM Assembler | 351 |
| Auto-Wake-up on Sync Break Character | 246 |

B

| | |
|----------------------------|-----|
| Bank Select Register (BSR) | 63 |
| Baud Rate Generator | 219 |
| Baud Rate Generator (BRG) | 237 |
| BC | 309 |
| BCF | 310 |

| | |
|--------------------------------------|-----|
| BF | 223 |
| BF Status Flag | 223 |
| Block Diagrams | |
| A/D | 256 |
| Analog Input Model | 257 |
| Baud Rate Generator | 219 |
| Capture Mode Operation | 143 |
| Comparator Analog Input Model | 267 |
| Comparator I/O Operating Modes | |
| (Diagram) | 264 |
| Comparator Output | 266 |
| Comparator Voltage Reference | 270 |
| Compare Mode Operation | 144 |
| Device Clock | 24 |
| Enhanced PWM | 151 |
| EUSART Receive | 244 |
| EUSART Transmit | 242 |
| External Power-on Reset Circuit | |
| (Slow V _{DD} Power-up) | 45 |
| Fail-Safe Clock Monitor | 294 |
| Generic I/O Port | 111 |
| High/Low-Voltage Detect with | |
| External Input | 274 |
| MSSP (I ² C Master Mode) | 217 |
| MSSP (I ² C Mode) | 202 |
| MSSP (SPI Mode) | 193 |
| On-Chip Reset Circuit | 43 |
| PIC18F2455/2550 | 10 |
| PIC18F4455/4550 | 11 |
| PLL (HS Mode) | 26 |
| PWM Operation (Simplified) | 146 |
| Reads from Flash Program Memory | 83 |
| Single Comparator | 265 |
| SPP Data Path | 187 |
| Table Read Operation | 79 |
| Table Write Operation | 80 |
| Table Writes to Flash Program Memory | 85 |
| Timer0 in 16-Bit Mode | 126 |
| Timer0 in 8-Bit Mode | 126 |
| Timer1 | 130 |
| Timer1 (16-Bit Read/Write Mode) | 130 |
| Timer2 | 136 |
| Timer3 | 138 |
| Timer3 (16-Bit Read/Write Mode) | 138 |
| USB Interrupt Logic | 177 |
| USB Peripheral and Options | 163 |
| Voltage Reference Output | |
| Buffer Example | 271 |
| Watchdog Timer | 291 |
| BN | 310 |
| BNC | 311 |
| BNN | 311 |
| BNOV | 312 |
| BNZ | 312 |
| BOR. <i>See</i> Brown-out Reset. | |
| BOV | 315 |
| BRA | 313 |
| BRG. <i>See</i> Baud Rate Generator. | |
| Brown-out Reset (BOR) | 46 |
| Disabling in Sleep mode | 46 |
| BSF | 313 |
| BTFSC | 314 |
| BTFSS | 314 |
| BTG | 315 |
| BZ | 316 |

PIC18F2455/2550/4455/4550

C

| | |
|--------------------------------------|---------------|
| C Compilers | |
| MPLAB C17 | 352 |
| MPLAB C18 | 352 |
| MPLAB C30 | 352 |
| CALL | 316 |
| CALLW | 345 |
| Capture (CCP Module) | 143 |
| Associated Registers | 145 |
| CCP Pin Configuration | 143 |
| CCPRxH:CCPRxL Registers | 143 |
| Prescaler | 143 |
| Software Interrupt | 143 |
| Timer1/Timer3 Mode Selection | 143 |
| Capture (ECCP Module) | 150 |
| Capture/Compare/PWM (CCP) | 141 |
| Capture Mode. See Capture. | |
| CCP Mode and Timer Resources | 142 |
| CCPRxH Register | 142 |
| CCPRxL Register | 142 |
| Compare Mode. See Compare. | |
| Interaction of Two CCP Modules for | |
| Timer Resources | 142 |
| Module Configuration | 142 |
| Clock Sources | 31 |
| Selecting the 31 kHz Source | 31 |
| Selection Using OSCCON Register | 31 |
| CLRf | 317 |
| CLRWDt | 317 |
| Code Examples | |
| 16 x 16 Signed Multiply Routine | 96 |
| 16 x 16 Unsigned Multiply Routine | 96 |
| 8 x 8 Signed Multiply Routine | 95 |
| 8 x 8 Unsigned Multiply Routine | 95 |
| Changing Between Capture | |
| Prescalers | 143 |
| Computed GOTO Using an | |
| Offset Value | 60 |
| Data EEPROM Read | 91 |
| Data EEPROM Refresh Routine | 92 |
| Data EEPROM Write | 91 |
| Erasing a Flash Program Memory Row | 84 |
| Fast Register Stack | 60 |
| How to Clear RAM (Bank 1) Using | |
| Indirect Addressing | 72 |
| Implementing a Real-Time Clock Using | |
| a Timer1 Interrupt Service | 133 |
| Initializing PORTA | 111 |
| Initializing PORTB | 114 |
| Initializing PORTC | 117 |
| Initializing PORTD | 120 |
| Initializing PORTE | 123 |
| Loading the SSPBUF (SSPSR) Register | 196 |
| Reading a Flash Program Memory Word | 83 |
| Saving Status, WREG and | |
| BSR Registers in RAM | 109 |
| Writing to Flash Program Memory | 86–87 |
| Code Protection | 279 |
| COMF | 318 |
| Comparator | 263 |
| Analog Input Connection | |
| Considerations | 267 |
| Associated Registers | 267 |
| Configuration | 264 |
| Effects of a Reset | 266 |
| Interrupts | 266 |
| Operation | 265 |
| Operation During Sleep | 266 |
| Outputs | 265 |
| Reference | 265 |
| External Signal | 265 |
| Internal Signal | 265 |
| Response Time | 265 |
| Comparator Specifications | 371 |
| Comparator Voltage Reference | 269 |
| Accuracy and Error | 270 |
| Associated Registers | 271 |
| Configuring | 269 |
| Connection Considerations | 270 |
| Effects of a Reset | 270 |
| Operation During Sleep | 270 |
| Compare (CCP Module) | 144 |
| Associated Registers | 145 |
| CCPRx Register | 144 |
| Pin Configuration | 144 |
| Software Interrupt | 144 |
| Special Event Trigger | 139, 144, 262 |
| Timer1/Timer3 Mode Selection | 144 |
| Compare (ECCP Module) | 150 |
| Special Event Trigger | 150 |
| Configuration Bits | 280 |
| Configuration Register Protection | 299 |
| Context Saving During Interrupts | 109 |
| Conversion Considerations | 406 |
| CPFSEQ | 318 |
| CPFSGT | 319 |
| CPFSLT | 319 |
| Crystal Oscillator/Ceramic Resonator | 25 |
| D | |
| Data Addressing Modes | 72 |
| Comparing Addressing Modes with the | |
| Extended Instruction Set Enabled | 76 |
| Direct | 72 |
| Indexed Literal Offset | 75 |
| Indirect | 72 |
| Inherent and Literal | 72 |
| Data EEPROM Code Protection | 299 |
| Data EEPROM Memory | 89 |
| Associated Registers | 93 |
| EECON1 and EECON2 Registers | 89 |
| Operation During Code-Protect | 92 |
| Protection Against Spurious Write | 92 |
| Reading | 91 |
| Using | 92 |
| Write Verify | 91 |
| Writing | 91 |

PIC18F2455/2550/4455/4550

| | | | |
|---|---------|--|----------|
| Data Memory | 63 | EUSART | |
| Access Bank | 65 | Asynchronous Mode | 242 |
| and the Extended Instruction Set | 75 | Associated Registers, Receive | 245 |
| Bank Select Register (BSR) | 63 | Associated Registers, Transmit | 243 |
| General Purpose Registers | 65 | Auto-Wake-up on Sync Break | 246 |
| Map for PIC18F2455/2550/4455/4550 | | Break Character Sequence | 247 |
| Devices | 64 | Receiver | 244 |
| Special Function Registers | 66 | Setting up 9-Bit Mode with | |
| USB RAM | 63 | Address Detect | 244 |
| DAW | 320 | Transmitter | 242 |
| DC and AC Characteristics | | Baud Rate Generator (BRG) | |
| Graphs and Tables | 395 | Associated Registers | 237 |
| DC Characteristics | 368 | Auto-Baud Rate Detect | 240 |
| Power-Down and Supply Current | 360 | Baud Rate Error, Calculating | 237 |
| Supply Voltage | 359 | Baud Rates, Asynchronous | |
| DCFSNZ | 321 | Modes | 238 |
| DECF | 320 | High Baud Rate Select | |
| DECFSZ | 321 | (BRGH Bit) | 237 |
| Dedicated ICD/ICSP Port | 299 | Operation in Power | |
| Demonstration Boards | | Managed Modes | 237 |
| PICDEM 1 | 354 | Sampling | 237 |
| PICDEM 17 | 355 | Synchronous Master Mode | 248 |
| PICDEM 18R | 355 | Associated Registers, | |
| PICDEM 2 Plus | 354 | Receive | 250 |
| PICDEM 3 | 354 | Associated Registers, | |
| PICDEM 4 | 354 | Transmit | 249 |
| PICDEM LIN | 355 | Reception | 250 |
| PICDEM USB | 355 | Transmission | 248 |
| PICDEM.net Internet/Ethernet | 354 | Synchronous Slave Mode | 251 |
| Development Support | 351 | Associated Registers, | |
| Device Differences | 405 | Receive | 252 |
| Device Overview | 7 | Associated Registers, | |
| Features (table) | 9 | Transmit | 251 |
| New Core Features | 7 | Reception | 252 |
| Other Special Features | 8 | Transmission | 251 |
| Device Reset Timers | 47 | Evaluation and Programming Tools | 355 |
| Oscillator Start-up Timer (OST) | 47 | Extended Instruction Set | 343 |
| PLL Lock Time-out | 47 | ADDFSR | 344 |
| Power-up Timer (PWRT) | 47 | ADDULNK | 344 |
| Direct Addressing | 73 | and Using MPLAB IDE Tools | 350 |
| E | | CALLW | 345 |
| Effect on Standard PIC Instructions | 75, 348 | Considerations for Use | 348 |
| Effects of Power Managed Modes on | | MOVSF | 345 |
| Various Clock Sources | 33 | MOVSS | 346 |
| Electrical Characteristics | 357 | PUSHL | 346 |
| Enhanced Capture/Compare/PWM (ECCP) | 149 | SUBFSR | 347 |
| Associated Registers | 162 | SUBULNK | 347 |
| Capture and Compare Modes | 150 | Syntax | 343 |
| Capture Mode. See Capture | | External Clock Input | 26 |
| (ECCP Module). | | F | |
| Outputs and Configuration | 150 | Fail-Safe Clock Monitor | 279, 294 |
| Pin Configurations for ECCP1 | 150 | Interrupts in Power | |
| PWM Mode. See PWM (ECCP Module). | | Managed Modes | 295 |
| Standard PWM Mode | 150 | POR or Wake-up from Sleep | 295 |
| Timer Resources | 150 | WDT During Oscillator Failure | 294 |
| Enhanced PWM Mode. See PWM (ECCP Module). | | Fast Register Stack | 60 |
| Enhanced Universal Synchronous | | Firmware Instructions | 301 |
| Receiver Transmitter (USART). See EUSART. | | | |
| Equations | | | |
| A/D Acquisition Time | 258 | | |
| A/D Minimum Charging Time | 258 | | |
| Errata | 5 | | |

PIC18F2455/2550/4455/4550

| | | | |
|--|-----|---|-------------|
| Flash Program Memory | 79 | Effect of a Reset | 227 |
| Associated Registers | 87 | General Call Address Support | 216 |
| Control Registers | 80 | I ² C Clock Rate w/BRG | 219 |
| EECON1 and EECON2 | 80 | Master Mode | 217 |
| TABLAT (Table Latch) Register | 82 | Operation | 218 |
| TBLPTR (Table Pointer) Register | 82 | Reception | 223 |
| Erase Sequence | 84 | Repeated Start Condition Timing | 222 |
| Erasing | 84 | Start Condition | 221 |
| Operation During Code-Protect | 87 | Transmission | 223 |
| Reading | 83 | Transmit Sequence | 218 |
| Table Pointer | | Multi-Master Communication, Bus Collision | |
| Boundaries Based on Operation | 82 | and Arbitration | 227 |
| Table Pointer Boundaries | 82 | Multi-Master Mode | 227 |
| Table Reads and Table Writes | 79 | Operation | 206 |
| Write Sequence | 85 | Read/Write Bit Information (R/W Bit) | 206, 207 |
| Writing To | 85 | Registers | 202 |
| Protection Against Spurious Writes | 87 | Serial Clock (RB1/AN10/INT1/SCK/SCL) | 207 |
| Unexpected Termination | 87 | Slave Mode | 206 |
| Write Verify | 87 | Addressing | 206 |
| FSCM. See Fail-Safe Clock Monitor. | | Reception | 207 |
| G | | Transmission | 207 |
| GOTO | 322 | Sleep Operation | 227 |
| H | | Stop Condition Timing | 226 |
| Hardware Multiplier | 95 | ID Locations | 279, 299 |
| Introduction | 95 | INCF | 322 |
| Operation | 95 | INCFSZ | 323 |
| Performance Comparison | 95 | In-Circuit Debugger | 299 |
| High/Low-Voltage Detect | 273 | In-Circuit Serial Programming (ICSP) | 279, 299 |
| Applications | 276 | Indexed Literal Offset Addressing | |
| Associated Registers | 277 | and Standard PIC18 Instructions | 348 |
| Characteristics | 373 | Indexed Literal Offset Mode | 75, 77, 348 |
| Current Consumption | 275 | Indirect Addressing | 73 |
| Effects of a Reset | 277 | INFSNZ | 323 |
| Operation | 274 | Initialization Conditions for all Registers | 51–55 |
| During Sleep | 277 | Instruction Cycle | 61 |
| Start-up Time | 275 | Clocking Scheme | 61 |
| Setup | 275 | Instruction Flow/Pipelining | 61 |
| Typical Application | 276 | Instruction Format | 303 |
| HLVD. See High/Low-Voltage Detect. | | Instruction Set | 301 |
| I | | ADDLW | 307 |
| I/O Ports | 111 | ADDWF | 307 |
| I ² C Mode (MSSP) | | ADDWF (Indexed Literal Offset mode) | 349 |
| Acknowledge Sequence Timing | 226 | ADDWFC | 308 |
| Baud Rate Generator | 219 | ANDLW | 308 |
| Bus Collision | | ANDWF | 309 |
| During a Repeated Start Condition | 230 | BC | 309 |
| During a Stop Condition | 231 | BCF | 310 |
| Clock Arbitration | 220 | BN | 310 |
| Clock Stretching | 212 | BNC | 311 |
| 10-Bit Slave Receive Mode | | BNN | 311 |
| (SEN = 1) | 212 | BNOV | 312 |
| 10-Bit Slave Transmit Mode | 212 | BNZ | 312 |
| 7-Bit Slave Receive Mode | | BOV | 315 |
| (SEN = 1) | 212 | BRA | 313 |
| 7-Bit Slave Transmit Mode | 212 | BSF | 313 |
| Clock Synchronization and the CKP bit | | BSF (Indexed Literal Offset mode) | 349 |
| (SEN = 1) | 213 | BTFSC | 314 |
| | | BTFSS | 314 |
| | | BTG | 315 |
| | | BZ | 316 |

PIC18F2455/2550/4455/4550

| | | | |
|--|-----|---|---------------|
| CALL | 316 | Interrupt Sources | 279 |
| CLRF | 317 | A/D Conversion Complete | 257 |
| CLRWDT | 317 | Capture Complete (CCP) | 143 |
| COMF | 318 | Compare Complete (CCP) | 144 |
| CPFSEQ | 318 | Interrupt-on-Change (RB7:RB4) | 114 |
| CPFSGT | 319 | INTn Pin | 109 |
| CPFSLT | 319 | PORTB, Interrupt-on-Change | 109 |
| DAW | 320 | TMR0 | 109 |
| DCFSNZ | 321 | TMR0 Overflow | 127 |
| DECf | 320 | TMR2 to PR2 Match (PWM) | 146, 151 |
| DECFSZ | 321 | TMR3 Overflow | 129, 137, 139 |
| GOTO | 322 | Interrupts | 97 |
| INCF | 322 | Logic (diagram) | 98 |
| INCFSZ | 323 | USB | 97 |
| INFSNZ | 323 | Interrupts, Flag Bits | |
| IORLW | 324 | Interrupt-on-Change (RB7:RB4) Flag | |
| IORWF | 324 | (RBIF Bit) | 114 |
| LFSR | 325 | INTOSC Frequency Drift | 27 |
| MOVF | 325 | INTOSC, INTRC. <i>See</i> Internal Oscillator Block. | |
| MOVFF | 326 | IORLW | 324 |
| MOVLB | 326 | IORWF | 324 |
| MOVLW | 327 | IPR Registers | 106 |
| MOVWF | 327 | | |
| MULLW | 328 | L | |
| MULWF | 328 | LFSR | 325 |
| NEGF | 329 | Look-up Tables | 60 |
| NOP | 329 | Low-Voltage ICSP Programming | |
| POP | 330 | <i>See</i> Single-Supply ICSP Programming. | |
| PUSH | 330 | | |
| RCALL | 331 | M | |
| RESET | 331 | Master Clear Reset ($\overline{\text{MCLR}}$) | 45 |
| RETFIE | 332 | Master Synchronous Serial Port (MSSP). <i>See</i> MSSP. | |
| RETLW | 332 | Memory Organization | 57 |
| RETURN | 333 | Data Memory | 63 |
| RLCF | 333 | Program Memory | 57 |
| RLNCF | 334 | Memory Programming Requirements | 370 |
| RRCF | 334 | Migration from Baseline to | |
| RRNCF | 335 | Enhanced Devices | 406 |
| SETF | 335 | Migration from High-End to | |
| SETF (Indexed Literal Offset mode) | 349 | Enhanced Devices | 407 |
| SLEEP | 336 | Migration from Mid-Range to | |
| SUBFWB | 336 | Enhanced Devices | 407 |
| SUBLW | 337 | MOVf | 325 |
| SUBWF | 337 | MOVFF | 326 |
| SUBWFB | 338 | MOVLB | 326 |
| SWAPF | 338 | MOVLW | 327 |
| TBLRD | 339 | MOVSF | 345 |
| TBLWT | 340 | MOVSS | 346 |
| TSTFSZ | 341 | MOVWF | 327 |
| XORLW | 341 | MPLAB ASM30 Assembler, | |
| XORWF | 342 | Linker, Librarian | 352 |
| Instructions in Program Memory | 62 | MPLAB ICD 2 In-Circuit Debugger | 353 |
| Two-Word Instructions | 62 | MPLAB ICE 2000 High-Performance | |
| INTCON Register | | Universal In-Circuit Emulator | 353 |
| RBIF Bit | 114 | MPLAB ICE 4000 High-Performance | |
| INTCON Registers | 99 | Universal In-Circuit Emulator | 353 |
| Inter-Integrated Circuit. <i>See</i> I ² C. | | MPLAB Integrated Development | |
| Internal Oscillator Block | 27 | Environment Software | 351 |
| Adjustment | 27 | MPLAB PM3 Device Programmer | 353 |
| INTHS, INTXT, INTCKO and INTIO Modes | 27 | MPLINK Object Linker/ | |
| OSCTUNE Register | 27 | MPLIB Object Librarian | 352 |
| Internal RC Oscillator | | | |
| Use with WDT | 291 | | |

PIC18F2455/2550/4455/4550

| | | | |
|---|----------|--|--------|
| MSSP | | | |
| ACK Pulse | 206, 207 | RA5/AN4/ \overline{SS} /HLVDIN/C2OUT | 13, 17 |
| Control Registers (general) | 193 | RB0/AN12/INT0/FLT0/SDI/SDA | 14, 18 |
| I ² C Mode. See I ² C Mode. | | RB1/AN10/INT1/SCK/SCL | 14, 18 |
| Module Overview | 193 | RB2/AN8/INT2/VMO | 14, 18 |
| SPI Master/Slave Connection | 197 | RB3/AN9/CCP2/VPO | 14, 18 |
| SPI Mode. See SPI Mode. | | RB4/AN11/KBI0 | 14 |
| SSPBUF | 198 | RB4/AN11/KBI0/CSSPP | 18 |
| SSPSR | 198 | RB5/KBI1/PGM | 14, 18 |
| MULLW | 328 | RB6/KBI2/PGC | 14, 18 |
| MULWF | 328 | RB7/KBI3/PGD | 14, 18 |
| N | | RC0/T1OSO/T13CKI | 15, 19 |
| NEGf | 329 | RC1/T1OSI/CCP2/UOE | 15, 19 |
| NOP | 329 | RC2/CCP1 | 15 |
| O | | RC2/CCP1/P1A | 19 |
| Opcode Field Descriptions | 302 | RC4/D-/VM | 19 |
| OPTION_REG Register | | RC5/D+/VP | 15, 19 |
| PSA Bit | 127 | RC6/TX/CK | 15, 19 |
| T0CS Bit | 126 | RC7/RX/DT/SDO | 15, 19 |
| T0PS2:T0PS0 Bits | 127 | RD0/SPP0 | 20 |
| T0SE Bit | 126 | RD1/SPP1 | 20 |
| Oscillator Configuration | 23 | RD2/SPP2 | 20 |
| EC | 23 | RD3/SPP3 | 20 |
| ECIO | 23 | RD4/SPP4 | 20 |
| ECPIO | 23 | RD5/SPP5/P1B | 20 |
| ECPLL | 23 | RD6/SPP6/P1C | 20 |
| HS | 23 | RD7/SPP7/P1D | 20 |
| HSPLL | 23 | RE0/AN5/CK1SPP | 21 |
| INTCKO | 23 | RE1/AN6/CK2SPP | 21 |
| Internal Oscillator Block | 27 | RE2/AN7/OESPP | 21 |
| INTHS | 23 | VDD | 15, 21 |
| INTIO | 23 | VSS | 15, 21 |
| INTXT | 23 | VUSB | 15, 21 |
| Oscillator Modes and USB Operation | 23 | Pinout I/O Descriptions | |
| XT | 23 | PIC18F2455/2550 | 12 |
| XTPLL | 23 | PIC18F4455/4550 | 16 |
| Oscillator Selection | 279 | PIR Registers | 102 |
| Oscillator Settings for USB | 29 | PLL Frequency Multiplier | 26 |
| Oscillator Start-up Timer (OST) | 33, 47 | HSPLL, XTPLL, ECPLL and ECPIO | |
| Oscillator Switching | 31 | Oscillator Modes | 26 |
| Oscillator Transitions | 32 | PLL Lock Time-out | 47 |
| Oscillator, Timer1 | 139 | POP | 330 |
| Oscillator, Timer3 | 129, 137 | POR. See Power-on Reset. | |
| P | | PORTA | |
| Packaging Information | 397 | Associated Registers | 113 |
| Details | 399 | I/O Summary | 112 |
| Marking | 397 | LATA Register | 111 |
| PICkit 1 Flash Starter Kit | 355 | PORTA Register | 111 |
| PICSTART Plus Development Programmer | 354 | TRISA Register | 111 |
| PIE Registers | 104 | PORTB | |
| Pin Functions | | Associated Registers | 116 |
| MCLR/VPP/RE3 | 12, 16 | I/O Summary | 115 |
| NC/ICCK/ICPGC | 21 | LATB Register | 114 |
| NC/ICDT/ICPGD | 21 | PORTB Register | 114 |
| NC/ICPORTS | 21 | RB1/AN10/INT1/SCK/SCL Pin | 207 |
| NC/ICRST/ICVPP | 21 | RB7:RB4 Interrupt-on-Change Flag | |
| OSC1/CLKI | 12, 16 | (RBIF Bit) | 114 |
| OSC2/CLKO/RA6 | 12, 16 | TRISB Register | 114 |
| RA0/AN0 | 13, 17 | PORTC | |
| RA1/AN1 | 13, 17 | Associated Registers | 119 |
| RA2/AN2/VREF-/CVREF | 13, 17 | I/O Summary | 118 |
| RA3/AN3/VREF+ | 13, 17 | LATC Register | 117 |
| RA4/T0CKI/C1OUT/RCV | 13, 17 | PORTC Register | 117 |
| | | TRISC Register | 117 |

PIC18F2455/2550/4455/4550

| | | | |
|--|----------|--|--|
| PORTD | | | |
| Associated Registers | 122 | | |
| I/O Summary | 121 | | |
| LATD Register | 120 | | |
| PORTD Register | 120 | | |
| TRISD Register | 120 | | |
| PORTE | | | |
| Associated Registers | 124 | | |
| I/O Summary | 124 | | |
| LATE Register | 123 | | |
| PORTE Register | 123 | | |
| TRISE Register | 123 | | |
| Postscaler, WDT | | | |
| Assignment (PSA Bit) | 127 | | |
| Rate Select (TOPS2:T0PS0 Bits) | 127 | | |
| Switching Between Timer0 and WDT | 127 | | |
| Power Managed Modes | 35 | | |
| and A/D Operation | 260 | | |
| and EUSART Operation | 237 | | |
| and Multiple Sleep Commands | 36 | | |
| Clock Transitions and Status Indicators | 36 | | |
| Effects on Clock Sources | 33 | | |
| Entering | 35 | | |
| Exiting Idle and Sleep Modes | 41 | | |
| by Interrupt | 41 | | |
| by Reset | 41 | | |
| by WDT Time-out | 41 | | |
| Without an Oscillator Start-up Delay | 42 | | |
| Idle Modes | 39 | | |
| PRI_IDLE | 40 | | |
| RC_IDLE | 41 | | |
| SEC_IDLE | 40 | | |
| Run Modes | 36 | | |
| PRI_RUN | 36 | | |
| RC_RUN | 37 | | |
| SEC_RUN | 36 | | |
| Selecting | 35 | | |
| Sleep Mode | 39 | | |
| Summary (table) | 35 | | |
| Power-on Reset (POR) | 45 | | |
| Oscillator Start-up Timer (OST) | 47 | | |
| Power-up Timer (PWRT) | 47 | | |
| Time-out Sequence | 47 | | |
| Power-up Delays | 33 | | |
| Power-up Timer (PWRT) | 33, 47 | | |
| Prescaler | | | |
| Timer2 | 152 | | |
| Prescaler, Timer0 | 127 | | |
| Assignment (PSA Bit) | 127 | | |
| Rate Select (TOPS2:T0PS0 Bits) | 127 | | |
| Switching Between Timer0 and WDT | 127 | | |
| Prescaler, Timer2 | 147 | | |
| PRI_IDLE Mode | 40 | | |
| PRI_RUN Mode | 36 | | |
| PRO MATE II Universal Device Programmer | 353 | | |
| Program Counter | 58 | | |
| PCL, PCH and PCU Registers | 58 | | |
| PCLATH and PCLATU Registers | 58 | | |
| Program Memory | | | |
| and the Extended Instruction Set | 75 | | |
| Code Protection | 297 | | |
| Interrupt Vector | 57 | | |
| Map and Stack (diagram) | 57 | | |
| Reset Vector | 57 | | |
| Program Verification and Code Protection | 296 | | |
| Associated Registers | 296 | | |
| Programming, Device Instructions | 301 | | |
| Pulse-Width Modulation. See PWM (CCP Module) and PWM (ECCP Module). | | | |
| PUSH | 330 | | |
| PUSH and POP Instructions | 59 | | |
| PUSHL | 346 | | |
| PWM (CCP Module) | | | |
| Associated Registers | 148 | | |
| Auto-Shutdown (CCP1 only) | 147 | | |
| Duty Cycle | 146 | | |
| Example Frequencies/Resolutions | 147 | | |
| Period | 146 | | |
| Setup for PWM Operation | 147 | | |
| TMR2 to PR2 Match | 146, 151 | | |
| PWM (ECCP Module) | 151 | | |
| CCPR1H:CCPR1L Registers | 151 | | |
| Direction Change in Full-Bridge | | | |
| Output Mode | 156 | | |
| Duty Cycle | 152 | | |
| Effects of a Reset | 161 | | |
| Enhanced PWM Auto-Shutdown | 158 | | |
| Example Frequencies/Resolutions | 152 | | |
| Full-Bridge Application Example | 156 | | |
| Full-Bridge Mode | 155 | | |
| Half-Bridge Mode | 154 | | |
| Half-Bridge Output Mode | | | |
| Applications Example | 154 | | |
| Output Configurations | 152 | | |
| Output Relationships (Active-High) | 153 | | |
| Output Relationships (Active-Low) | 153 | | |
| Period | 151 | | |
| Programmable Dead-Band Delay | 158 | | |
| Setup for Operation | 161 | | |
| Start-up Considerations | 160 | | |
| Q | | | |
| Q Clock | 147, 152 | | |
| R | | | |
| RAM. See Data Memory. | | | |
| RC_IDLE Mode | 41 | | |
| RC_RUN Mode | 37 | | |
| RC4/D-VM | 15 | | |
| RCALL | 331 | | |
| RCON Register | | | |
| Bit Status During Initialization | 50 | | |
| Register File | 65 | | |
| Register File Summary | 67–70 | | |
| Registers | | | |
| ADCON0 (A/D Control 0) | 253 | | |
| ADCON1 (A/D Control 1) | 254 | | |
| ADCON2 (A/D Control 2) | 255 | | |
| BAUDCON (Baud Rate Control) | 236 | | |
| BDnSTAT (Buffer Descriptor n Status, CPU Mode) | 173 | | |
| BDnSTAT (Buffer Descriptor n Status, SIE Mode) | 174 | | |
| CCP1CON (Enhanced CCP Control 1) | 149 | | |
| CCPxCON (Standard Capture/Compare/ PWM Control) | 141 | | |
| CMCON (Comparator Control) | 263 | | |
| CONFIG1H (Configuration 1 High) | 282 | | |

PIC18F2455/2550/4455/4550

| | | | |
|---|---------|---|---------|
| CONFIG1L (Configuration 1 Low) | 281 | RESET | 331 |
| CONFIG2H (Configuration 2 High) | 284 | Reset State of Registers | 50 |
| CONFIG2L (Configuration 2 Low) | 283 | Resets | 43, 279 |
| CONFIG3H (Configuration 3 High) | 285 | Brown-out Reset (BOR) | 279 |
| CONFIG4L (Configuration 4 Low) | 286 | Oscillator Start-up Timer (OST) | 279 |
| CONFIG5H (Configuration 5 High) | 287 | Power-on Reset (POR) | 279 |
| CONFIG5L (Configuration 5 Low) | 287 | Power-up Timer (PWRT) | 279 |
| CONFIG6H (Configuration 6 High) | 288 | RETfie | 332 |
| CONFIG6L (Configuration 6 Low) | 288 | RETLW | 332 |
| CONFIG7H (Configuration 7 High) | 289 | RETURN | 333 |
| CONFIG7L (Configuration 7 Low) | 289 | Return Address Stack | 58 |
| CVRCON (Comparator Voltage | | and Associated Registers | 58 |
| Reference Control) | 269 | Return Stack Pointer (STKPTR) | 59 |
| Device ID 1 | 290 | Revision History | 405 |
| Device ID 2 | 290 | RLCF | 333 |
| ECCP1AS (ECCP | | RLNCF | 334 |
| Auto-Shutdown Control) | 159 | RRCF | 334 |
| ECCP1DEL (PWM Configuration) | 158 | RRNCF | 335 |
| ECON1 (Data EEPROM | | S | |
| Control 1) | 81, 90 | SCK | 193 |
| HLVDCON (HLVD Control) | 273 | SDI | 193 |
| INTCON (Interrupt Control) | 99 | SDO | 193 |
| INTCON2 (Interrupt Control 2) | 100 | SEC_IDLE Mode | 40 |
| INTCON3 (Interrupt Control 3) | 101 | SEC_RUN Mode | 36 |
| IPR1 (Peripheral Interrupt Priority 1) | 106 | Serial Clock, SCK | 193 |
| IPR2 (Peripheral Interrupt Priority 2) | 107 | Serial Data In (SDI) | 193 |
| OSCCON (Oscillator Control) | 32 | Serial Data Out (SDO) | 193 |
| OSCTUNE (Oscillator Tuning) | 28 | Serial Peripheral Interface. See SPI Mode. | |
| PIE1 (Peripheral Interrupt Enable 1) | 104 | SETF | 335 |
| PIE2 (Peripheral Interrupt Enable 2) | 105 | Slave Select (\overline{SS}) | 193 |
| PIR1 (Peripheral Interrupt Request | | SLEEP | 336 |
| (Flag 1) | 102 | Sleep | |
| PIR2 (Peripheral Interrupt Request | | OSC1 and OSC2 Pin States | 33 |
| (Flag 2) | 103 | Sleep Mode | 39 |
| PORTE | 123 | Software Enabled BOR | 46 |
| RCON (Reset Control) | 44, 108 | Software Simulator (MPLAB SIM) | 352 |
| RCSTA (Receive Status and Control) | 235 | Software Simulator (MPLAB SIM30) | 352 |
| SPPCFG (SSP Configuration) | 188 | Special Event Trigger. See Compare (ECCP Mode). | |
| SPPCON (SPP Control) | 187 | Special Event Trigger. See Compare (ECCP Module). | |
| SPPEPS (SPP Endpoint Address | | Special Features of the CPU | 279 |
| and Status) | 191 | Special Function Registers | 66 |
| SSPCON1 (MSSP Control 1, I ² C Mode) | 204 | Map | 66 |
| SSPCON1 (MSSP Control 1, SPI Mode) | 195 | Special ICPORT Features | 299 |
| SSPCON2 (MSSP Control 2, I ² C Mode) | 205 | SPI Mode (MSSP) | |
| SSPSTAT (MSSP Status, I ² C Mode) | 203 | Associated Registers | 201 |
| SSPSTAT (MSSP Status, SPI Mode) | 194 | Bus Mode Compatibility | 201 |
| Status | 71 | Effects of a Reset | 201 |
| STKPTR (Stack Pointer) | 59 | Enabling SPI I/O | 197 |
| T0CON (Timer0 Control) | 125 | Master Mode | 198 |
| T1CON (Timer 1 Control) | 129 | Master/Slave Connection | 197 |
| T2CON (Timer2 Control) | 135 | Operation | 196 |
| T3CON (Timer3 Control) | 137 | Operation in Power Managed Modes | 201 |
| TXSTA (Transmit Status and Control) | 234 | Serial Clock | 193 |
| UCFG (USB Configuration) | 166 | Serial Data In | 193 |
| UCON (USB Control) | 164 | Serial Data Out | 193 |
| UEIE (USB Error Interrupt Enable) | 181 | Slave Mode | 199 |
| UEIR (USB Error Interrupt Status) | 180 | Slave Select | 193 |
| UEPn (USB Endpoint n Control) | 169 | Slave Select Synchronization | 199 |
| UIE (USB Interrupt Enable) | 179 | SPI Clock | 198 |
| UIR (USB Interrupt Status) | 178 | Typical Connection | 197 |
| USTAT (USB Status) | 168 | SPP. See Streaming Parallel Port. | 187 |
| WDTCON (Watchdog Timer Control) | 292 | | |

PIC18F2455/2550/4455/4550

| | | | |
|--|----------|--|---------------|
| SS | 193 | Timer3 | 137 |
| SSPOV | 223 | 16-Bit Read/Write Mode | 139 |
| SSPOV Status Flag | 223 | Associated Registers | 139 |
| SSPSTAT Register | | Operation | 138 |
| R/W Bit | 206, 207 | Oscillator | 129, 137, 139 |
| Stack Full/Underflow Resets | 60 | Overflow Interrupt | 129, 137, 139 |
| Standard Instructions | 301 | Special Event Trigger (CCP) | 139 |
| Status Register | 71 | TMR3H Register | 129, 137 |
| Streaming Parallel Port | 187 | TMR3L Register | 129, 137 |
| Associated Registers | 192 | Timing Diagrams | |
| Clocking Data | 188 | A/D Conversion | 394 |
| Configuration | 187 | Acknowledge Sequence | 226 |
| Internal Pull-ups | 188 | Asynchronous Reception | 245 |
| Interrupts | 190 | Asynchronous Transmission | 243 |
| Reading from (Microcontroller Mode) | 191 | Asynchronous Transmission | |
| Setup for Microcontroller Control | 190 | (Back to Back) | 243 |
| Setup for USB Control | 190 | Automatic Baud Rate Calculation | 241 |
| Transfer of Data Between SIE and SPP | 190 | Auto-Wake-up Bit (WUE) During | |
| Wait States | 188 | Normal Operation | 246 |
| Writing to (Microcontroller Mode) | 190 | Auto-Wake-up Bit (WUE) During Sleep | 246 |
| SUBFSR | 347 | Baud Rate Generator with Clock Arbitration | 220 |
| SUBFWB | 336 | BRG Overflow Sequence | 241 |
| SUBLW | 337 | BRG Reset Due to SDA Arbitration | |
| SUBULNK | 347 | During Start Condition | 229 |
| SUBWF | 337 | Brown-out Reset (BOR) | 379 |
| SUBWFB | 338 | Bus Collision During a Repeated | |
| SWAPF | 338 | Start Condition (Case 1) | 230 |
| T | | Bus Collision During a Repeated | |
| Table Pointer Operations (table) | 82 | Start Condition (Case 2) | 230 |
| Table Reads/Table Writes | 60 | Bus Collision During a Start | |
| TBLRD | 339 | Condition (SCL = 0) | 229 |
| TBLWT | 340 | Bus Collision During a Start | |
| Time-out in Various Situations (table) | 47 | Condition (SDA only) | 228 |
| Timer0 | 125 | Bus Collision During a Stop | |
| 16-Bit Mode Timer Reads and Writes | 126 | Condition (Case 1) | 231 |
| Associated Registers | 127 | Bus Collision During a Stop | |
| Clock Source Edge Select (TOSE Bit) | 126 | Condition (Case 2) | 231 |
| Clock Source Select (T0CS Bit) | 126 | Bus Collision for Transmit and | |
| Operation | 126 | Acknowledge | 227 |
| Overflow Interrupt | 127 | Capture/Compare/PWM (CCP) | 381 |
| Prescaler. <i>See</i> Prescaler, Timer0. | | CLKO and I/O | 378 |
| Timer1 | 129 | Clock Synchronization | 213 |
| 16-Bit Read/Write Mode | 131 | Clock/Instruction Cycle | 61 |
| Associated Registers | 133 | Example SPI Master Mode (CKE = 0) | 382 |
| Interrupt | 132 | Example SPI Master Mode (CKE = 1) | 383 |
| Operation | 130 | Example SPI Slave Mode (CKE = 0) | 384 |
| Oscillator | 131 | Example SPI Slave Mode (CKE = 1) | 385 |
| Layout Considerations | 132 | External Clock (All Modes except PLL) | 376 |
| Low-Power Option | 131 | Fail-Safe Clock Monitor | 295 |
| Using Timer1 as a Clock Source | 131 | First Start Bit Timing | 221 |
| Resetting, Using a Special Event | | Full-Bridge PWM Output | 155 |
| Trigger Output (CCP) | 132 | Half-Bridge PWM Output | 154 |
| Special Event Trigger (ECCP) | 150 | High/Low-Voltage Detect Characteristics | 373 |
| Use as a Real-Time Clock | 132 | High-Voltage Detect (VDIRMAG = 1) | 276 |
| Timer2 | 135 | I ² C Bus Data | 386 |
| Associated Registers | 136 | I ² C Bus Start/Stop Bits | 386 |
| Interrupt | 136 | I ² C Master Mode (7 or 10-Bit | |
| Operation | 135 | Transmission) | 224 |
| Output | 136 | I ² C Master Mode (7-Bit Reception) | 225 |
| PR2 Register | 146, 151 | I ² C Slave Mode (10-Bit Reception, | |
| TMR2 to PR2 Match Interrupt | 146, 151 | SEN = 0) | 210 |
| | | I ² C Slave Mode (10-Bit Reception, | |
| | | SEN = 1) | 215 |

PIC18F2455/2550/4455/4550

| | | | |
|--|-----|--|----------|
| I ² C Slave Mode (10-Bit Transmission) | 211 | Transition from RC_RUN Mode to PRI_RUN Mode | 38 |
| I ² C Slave Mode (7-Bit Reception, SEN = 0) | 208 | Transition from SEC_RUN Mode to PRI_RUN Mode (HSPLL) | 37 |
| I ² C Slave Mode (7-Bit Reception, SEN = 1) | 214 | Transition Timing for Entry to Idle Mode | 40 |
| I ² C Slave Mode (7-Bit Transmission) | 209 | Transition Timing for Wake from Idle to Run Mode | 40 |
| I ² C Slave Mode General Call Address Sequence (7 or 10-Bit Address Mode) | 216 | Transition to RC_RUN Mode | 38 |
| Low-Voltage Detect (VDIRMAG = 0) | 275 | USART Synchronous Receive (Master/Slave) | 390 |
| Master SSP I ² C Bus Data | 388 | USART Synchronous Transmission (Master/Slave) | 390 |
| Master SSP I ² C Bus Start/Stop Bits | 388 | Timing Diagrams and Specifications | 376 |
| PWM Auto-Shutdown (PRSEN = 0, Auto-Restart Disabled) | 160 | A/D Conversion Requirements | 394 |
| PWM Auto-Shutdown (PRSEN = 1, Auto-Restart Enabled) | 160 | Capture/Compare/PWM Requirements | 381 |
| PWM Direction Change | 157 | CLKO and I/O Requirements | 378 |
| PWM Direction Change at Near 100% Duty Cycle | 157 | Example SPI Mode Requirements (Master Mode, CKE = 0) | 382 |
| PWM Output | 146 | Example SPI Mode Requirements (Master Mode, CKE = 1) | 383 |
| Repeat Start Condition | 222 | Example SPI Mode Requirements (Slave Mode, CKE = 0) | 384 |
| Reset, Watchdog Timer (WDT), Oscillator Start-up Timer (OST) and Power-up Timer (PWRT) | 379 | Example SPI Mode Requirements (Slave Mode, CKE = 1) | 385 |
| Send Break Character Sequence | 247 | External Clock Requirements | 376 |
| Slave Synchronization | 199 | I ² C Bus Data Requirements (Slave Mode) | 387 |
| Slow Rise Time (MCLR Tied to VDD, VDD Rise > TPWRT) | 49 | I ² C Bus Start/Stop Bits Requirements | 386 |
| SPI Mode (Master Mode) | 198 | Master SSP I ² C Bus Data Requirements | 389 |
| SPI Mode (Slave Mode with CKE = 0) | 200 | Master SSP I ² C Bus Start/Stop Bits Requirements | 388 |
| SPI Mode (Slave Mode with CKE = 1) | 200 | Parallel Slave Port Requirements (PIC18F4455/4550) | 392 |
| SPP Write Address and Data (USB Mode, 4 Wait States) | 189 | PLL Clock | 377 |
| SPP Write Address, Read Data (USB Mode, 4 Wait States) | 189 | Reset, Watchdog Timer, Oscillator Start-up Timer, Power-up Timer and Brown-out Reset Requirements | 379 |
| SPP Write Address, Write and Read Data (No Wait States) | 189 | Timer0 and Timer1 External Clock Requirements | 380 |
| Stop Condition Receive or Transmit Mode | 226 | USART Synchronous Receive Requirements | 390 |
| Streaming Parallel Port (PIC18F4455/4550) | 392 | USART Synchronous Transmission Requirements | 390 |
| Synchronous Reception (Master Mode, SREN) | 250 | USB Full Speed Requirements | 391 |
| Synchronous Transmission | 248 | USB Low-Speed Requirements | 391 |
| Synchronous Transmission (Through TXEN) | 249 | USB Signal Timing | 391 |
| Time-out Sequence on POR w/ PLL Enabled (MCLR Tied to VDD) | 49 | Top-of-Stack Access | 58 |
| Time-out Sequence on Power-up (MCLR Not Tied to VDD), Case 1 | 48 | TQFP Packages and Special Features | 299 |
| Time-out Sequence on Power-up (MCLR Not Tied to VDD), Case 2 | 48 | TSTFSZ | 341 |
| Time-out Sequence on Power-up (MCLR Tied to VDD, VDD Rise TPWRT) | 48 | Two-Speed Start-up | 279, 293 |
| Timer0 and Timer1 External Clock | 380 | Two-Word Instructions Example Cases | 62 |
| Transition for Entry to SEC_RUN Mode | 37 | TXSTA Register BRGH Bit | 237 |
| Transition for Entry to Sleep Mode | 39 | | |
| Transition for Two-Speed Start-up (INTOSC to HSPLL) | 293 | | |
| Transition for Wake from Sleep (HSPLL) | 39 | | |

PIC18F2455/2550/4455/4550

U

| | |
|--|-----|
| Universal Serial Bus | |
| Address Register (UADDR) | 170 |
| and Streaming Parallel Port | 182 |
| Associated Registers | 183 |
| Buffer Descriptor Table | 171 |
| Buffer Descriptors | 171 |
| Address Validation | 174 |
| Assignment in Different | |
| Buffering Modes | 176 |
| BDSTAT Register (CPU Mode) | 172 |
| BDSTAT Register (SIE Mode) | 174 |
| Byte Count | 174 |
| Example | 171 |
| Memory Map | 175 |
| Ownership | 171 |
| Ping-Pong Buffering | 175 |
| Register Summary | 176 |
| Status and Configuration | 171 |
| Bus Power Only Mode | 182 |
| Class Specifications and Drivers | 185 |
| Descriptors | 185 |
| Dual Power with Self-Power | |
| Dominance Mode | 182 |
| Endpoint Control | 169 |
| Enumeration | 185 |
| External Transceiver | 165 |
| Eye Pattern Test Enable | 167 |
| Firmware and Drivers | 183 |
| Frame Number Registers | 170 |
| Frames | 184 |
| Internal Transceiver | 165 |
| Internal Voltage Regulator | 167 |
| Interrupts | 177 |
| and USB Transactions | 177 |
| Interrupt Logic | 177 |
| Layered Framework | 184 |
| Oscillator Requirements | 183 |
| Output Enable Monitor | 167 |
| Overview | 184 |
| Ping-Pong Buffer Configuration | 167 |
| Power | 184 |
| Power Modes | 182 |
| Pull-up Resistors | 167 |
| RAM | 170 |
| Memory Map | 170 |
| Self-Power Only Mode | 182 |
| Speed | 185 |
| Status and Control | 164 |
| Transfer Types | 184 |
| UFRMH:UFRML Registers | 170 |
| USB Memory | 63 |
| USB RAM | 63 |
| USB. See Universal Serial Bus. | |
| Universal Serial Bus | |
| Overview | 163 |

V

| | |
|--|-----|
| Voltage Reference Specifications | 371 |
|--|-----|

W

| | |
|----------------------------------|--------------------|
| Watchdog Timer (WDT) | 279, 291 |
| Associated Registers | 292 |
| Control Register | 291 |
| During Oscillator Failure | 294 |
| Programming Considerations | 291 |
| WCOL | 221, 222, 223, 226 |
| WCOL Status Flag | 221, 222, 223, 226 |
| WWW, On-Line Support | 5 |

X

| | |
|-------------|-----|
| XORLW | 341 |
| XORWF | 342 |

PIC18F2455/2550/4455/4550

NOTES:

ON-LINE SUPPORT

Microchip provides on-line support on the Microchip World Wide Web site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape® or Microsoft® Internet Explorer. Files are also available for FTP download from our FTP site.

Connecting to the Microchip Internet Web Site

The Microchip web site is available at the following URL:

www.microchip.com

The file transfer site is available by using an FTP service to connect to:

<ftp://ftp.microchip.com>

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products
- Conferences for products, Development Systems, technical information and more
- Listing of seminars and events

SYSTEMS INFORMATION AND UPGRADE HOT LINE

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive the most current upgrade kits. The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada and

1-480-792-7302 for the rest of the world.

042003

PIC18F2455/2550/4455/4550

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information and use this outline to provide us with your comments about this document.

To: Technical Publications Manager
RE: Reader Response
Total Pages Sent _____

From: Name _____
Company _____
Address _____
City / State / ZIP / Country _____
Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC18F2455/2550/4455/4550 Literature Number: DS39632A

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

PIC18F2455/2550/4455/4550

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.

| <u>PART NO.</u> | | <u>X</u> | <u>/XX</u> | <u>XXX</u> |
|-------------------|--|----------|-----------------------------|------------|
| Device | Temperature Range | Package | Pattern | |
| Device | PIC18F2455/2550 ⁽¹⁾ , PIC18F4455/4550 ⁽¹⁾ , PIC18F2455/2550T ⁽²⁾ , PIC18F4455/4550T ⁽²⁾ ; V _{DD} range 4.2V to 5.5V PIC18LF2455/2550 ⁽¹⁾ , PIC18LF4455/4550 ⁽¹⁾ , PIC18LF2455/2550T ⁽²⁾ , PIC18LF4455/4550T ⁽²⁾ ; V _{DD} range 2.0V to 5.5V | | | |
| Temperature Range | I | = | -40°C to +85°C (Industrial) | |
| | E | = | -40°C to +125°C (Extended) | |
| Package | PT | = | TQFP (Thin Quad Flatpack) | |
| | SO | = | SOIC | |
| | SP | = | Skinny Plastic DIP | |
| | P | = | PDIP | |
| | M | = | QFN | |
| Pattern | QTP, SQTP, Code or Special Requirements (blank otherwise) | | | |

Examples:

- a) PIC18LF4550-I/P 301 = Industrial temp., PDIP package, Extended V_{DD} limits, QTP pattern #301.
- b) PIC18LF2455-I/SO = Industrial temp., SOIC package, Extended V_{DD} limits.
- c) PIC18F4455-I/P = Industrial temp., PDIP package, normal V_{DD} limits.

Note 1: F = Standard Voltage Range
 LF = Wide Voltage Range

2: T = in tape and reel TQFP packages only.



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: 480-792-7627
Web Address: <http://www.microchip.com>

Atlanta

3780 Mansell Road, Suite 130
Alpharetta, GA 30022
Tel: 770-640-0034
Fax: 770-640-0307

Boston

2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848
Fax: 978-692-3821

Chicago

333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071
Fax: 630-285-0075

Dallas

4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423
Fax: 972-818-2924

Detroit

Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo

2767 S. Albright Road
Kokomo, IN 46902
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles

18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888
Fax: 949-263-1338

San Jose

1300 Terra Bella Avenue
Mountain View, CA 94043
Tel: 650-215-1444
Fax: 650-961-0286

Toronto

6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia

Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing

Unit 706B
Wan Tai Bei Hai Bldg.
No. 6 Chaoyangmen Bei Str.
Beijing, 100027, China
Tel: 86-10-85282100
Fax: 86-10-85282104

China - Chengdu

Rm. 2401-2402, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-86766200
Fax: 86-28-86766599

China - Fuzhou

Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506
Fax: 86-591-7503521

China - Hong Kong SAR

Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Shanghai

Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700
Fax: 86-21-6275-5060

China - Shenzhen

Rm. 1812, 18/F, Building A, United Plaza
No. 5022 Binhe Road, Futian District
Shenzhen 518033, China
Tel: 86-755-82901380
Fax: 86-755-8295-1393

China - Shunde

Room 401, Hongjian Building, No. 2
Fengxiangnan Road, Ronggui Town, Shunde
District, Foshan City, Guangdong 528303, China
Tel: 86-757-28395507 Fax: 86-757-28395571

China - Qingdao

Rm. B505A, Fullhope Plaza,
No. 12 Hong Kong Central Rd.
Qingdao 266071, China
Tel: 86-532-5027355 Fax: 86-532-5027205

India

Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaughnessy Road
Bangalore, 560 025, India
Tel: 91-80-22290061 Fax: 91-80-22290062

Japan

Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea

168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5932 or
82-2-558-5934

Singapore

200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

Taiwan

Kaohsiung Branch
30F - 1 No. 8
Min Chuan 2nd Road
Kaohsiung 806, Taiwan
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan

Taiwan Branch
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Austria

Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark

Regus Business Centre
Lautrup høj 1-3
Ballerup DK-2750 Denmark
Tel: 45-4420-9895 Fax: 45-4420-9910

France

Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - Ier Etage
91300 Massy, France
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany

Steinheilstrasse 10
D-85737 Ismaning, Germany
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy

Via Quasimodo, 12
20025 Legnano (MI)
Milan, Italy
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands

P. A. De Biesbosch 14
NL-5152 SC Drunen, Netherlands
Tel: 31-416-690399
Fax: 31-416-690340

United Kingdom

505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44-118-921-5869
Fax: 44-118-921-5820

02/17/04