查询PIC24FJ64GA104供应商

PIC24FJ64GA104 Family Data Sheet

28/44-Pin, 16-Bit General Purpose Flash Microcontrollers with nanoWatt XLP 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.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Trademarks

Printed on recycled paper.

ISBN:978-1-60932-440-7

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

MICROCHIP

PIC24FJ64GA104 FAMILY

28/44-Pin, 16-Bit General Purpose Flash Microcontrollers with nanoWatt XLP Technology

Power Management Modes:

- Selectable Power Management modes with nanoWatt XLP Technology for Extremely Low Power:
  - Deep Sleep mode allows near total power-down (20 nA typical and 500 nA with RTCC or WDT), along with the ability to wake-up on external triggers, or self-wake on programmable WDT or RTCC alarm
  - Extreme low-power DSBOR for Deep Sleep, LPBOR for all other modes
  - Sleep mode shuts down peripherals and core for substantial power reduction, fast wake-up
  - Idle mode shuts down the CPU and peripherals for significant power reduction, down to 4.5 μA typical
  - Doze mode enables CPU clock to run slower than peripherals
  - Alternate Clock modes allow on-the-fly switching to a lower clock speed for selective power reduction during Run mode, down to 15 μA typical

High-Performance CPU:

- Modified Harvard Architecture
- Up to 16 MIPS Operation @ 32 MHz
- 8 MHz Internal Oscillator with:
  - 4x PLL option
  - Multiple divide options
- 17-Bit x 17-Bit Single-Cycle Hardware Fractional/integer Multiplier
- 32-Bit by 16-Bit Hardware Divider
- 16 x 16-Bit Working Register Array
- C Compiler Optimized Instruction Set Architecture:
  - 76 base instructions
  - Flexible addressing modes
- Linear Program Memory Addressing, up to 12 Mbytes
- Linear Data Memory Addressing, up to 64 Kbytes
- Two Address Generation Units for Separate Read and Write Addressing of Data Memory

Special Microcontroller Features:

- Operating Voltage Range of 2.0V to 3.6V
- Self-Reprogrammable under Software Control
- 5.5V Tolerant Input (digital pins only)
- High-Current Sink/Source (18 mA/18 mA) on All I/O pins

Special Microcontroller Features (continued):

- Flash Program Memory:
  - 10,000 erase/write cycle endurance (minimum)
  - 20-year data retention minimum
  - Selectable write protection boundary
- Fail-Safe Clock Monitor Operation:
  - Detects clock failure and switches to on-chip FRC Oscillator
- On-Chip 2.5V Regulator
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Two Flexible Watchdog Timers (WDT) for Reliable Operation:
  - Standard programmable WDT for normal operation
  - Extreme low-power WDT with programmable period of 2 ms to 26 days for Deep Sleep mode
- In-Circuit Serial Programming™ (ICSP™) and In-Circuit Debug (ICD) via 2 Pins
- JTAG Boundary Scan Support

Analog Features:

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter:
  - 500 ksps conversion rate
  - Conversion available during Sleep and Idle
- Three Analog Comparators with Programmable Input/Output Configuration
- Charge Time Measurement Unit (CTMU):
  - Supports capacitive touch sensing for touch screens and capacitive switches
  - Provides high-resolution time measurement and simple temperature sensing

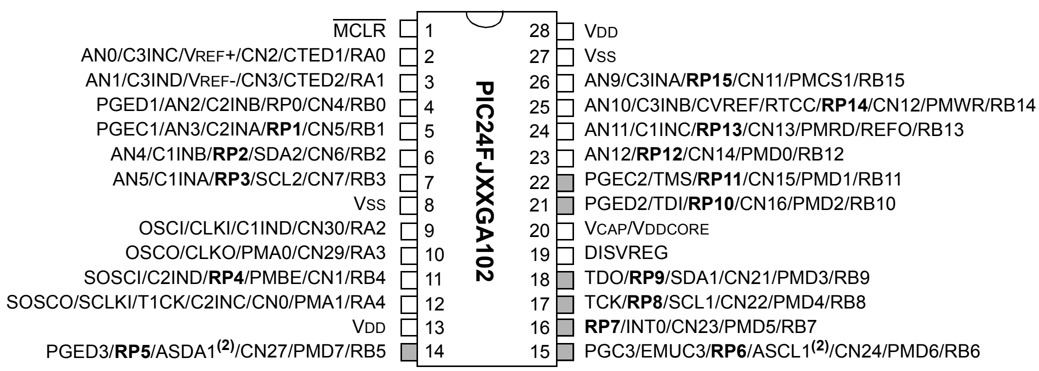
PIC24FJ Device	Pins	Program Memory (Bytes)	SRAM (Bytes)	Remappable Peripherals						I²C™	10-Bit A/D (ch)	Comparators	PMP/PSP	RTCC	CTMU
				Remappable Pins	Timers 16-Bit	Capture Input	Compare/PWM Output	UART w/ IrDA®	SPI						
32GA102	28	32K	8K	16	5	5	5	2	2	2	10	3	Y	Y	Y
64GA102	28	64K	8K	16	5	5	5	2	2	2	10	3	Y	Y	Y
32GA104	44	32K	8K	26	5	5	5	2	2	2	13	3	Y	Y	Y
64GA104	44	64K	8K	26	5	5	5	2	2	2	13	3	Y	Y	Y

Peripheral Features:

- Peripheral Pin Select:
  - Allows independent I/O mapping of many peripherals
  - Up to 26 available pins (44-pin devices)
  - Continuous hardware integrity checking and safety interlocks prevent unintentional configuration changes
- 8-Bit Parallel Master Port (PMP/PSP):
  - Up to 16-bit multiplexed addressing, with up to 11 dedicated address pins on 44-pin devices
  - Programmable polarity on control lines
  - Supports legacy Parallel Slave Port
- Hardware Real-Time Clock/Calendar (RTCC):
  - Provides clock, calendar and alarm functions
  - Functions even in Deep Sleep mode
- Two 3-Wire/4-Wire SPI modules (support 4 Frame modes) with 8-Level FIFO Buffer
- Two I^{2}C™ modules support Multi-Master/Slave mode and 7-Bit/10-Bit Addressing
- Two UART modules:
  - Supports RS-485, RS-232 and LIN/J2602
  - On-chip hardware encoder/decoder for IrDA®
  - Auto-wake-up on Start bit
  - Auto-Baud Detect (ABD)
  - 4-level deep FIFO buffer
- Five 16-Bit Timers/Counters with Programmable Prescaler
- Five 16-Bit Capture Inputs, each with a Dedicated Time Base
- Five 16-Bit Compare/PWM Outputs, each with a Dedicated Time Base
- Programmable, 32-Bit Cyclic Redundancy Check (CRC) Generator
- Configurable Open-Drain Outputs on Digital I/O Pins
- Up to 3 External Interrupt Sources

Pin Diagrams

28-Pin SPDIP, SOIC, SSOP(1)



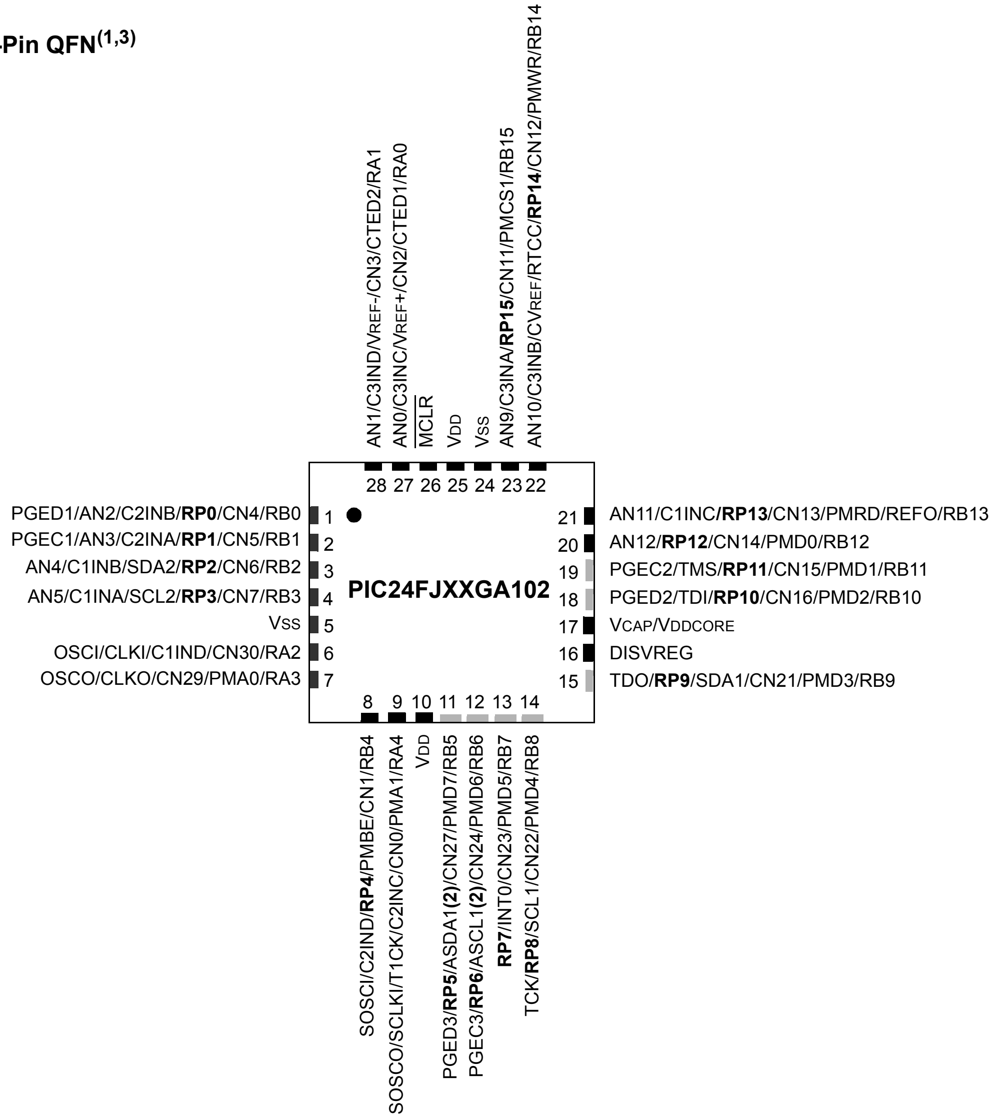
Legend: **RPn** represents remappable peripheral pins.

Note 1: Gray shading indicates 5.5V tolerant input pins.

2: Alternative multiplexing for SDA1 and SCL1 when the I2C1SEL bit is set.

Pin Diagrams

28-Pin QFN(1,3)



Legend: **RPn** represents remappable peripheral pins.

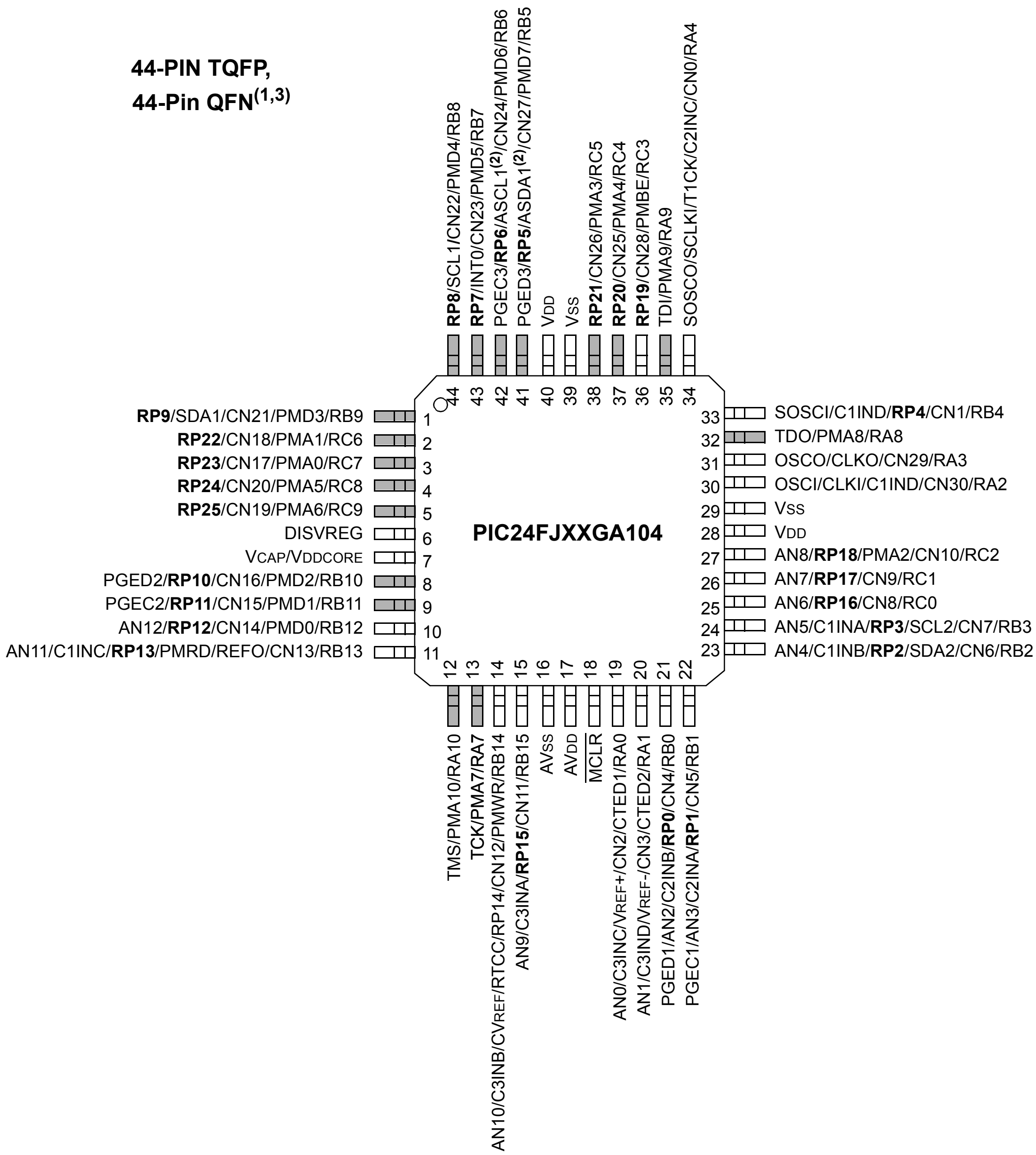
Note 1: Gray shading indicates 5.5V tolerant input pins.

2: Alternative multiplexing for SDA1 and SCL1 when the I2C1SEL bit is set.

3: The back pad on QFN devices should be connected to VSS.

查询PIC24FJ64GA104供应商

Pin Diagrams



Legend: **RPn** represents remappable peripheral pins.

Note 1: Gray shading indicates 5.5V tolerant input pins.

2: Alternative multiplexing for SDA1 and SCL1 when the I2C1SEL bit is set.

3: The back pad on QFN devices should be connected to VSS.

Table of Contents

TO OUR VALUED CUSTOMERS

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

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

Most Current Data Sheet

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

http://www.microchip.com

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

Errata

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

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

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

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

Customer Notification System

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

1.0 DEVICE OVERVIEW

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

- PIC24FJ32GA102
- PIC24FJ64GA102
- PIC24FJ32GA104
- PIC24FJ64GA104

The PIC24FJ64GA104 family provides an expanded peripheral feature set and a new option for high-performance applications which may need more than an 8-bit platform, but do not require the power of a digital signal processor.

1.1 Core Features

1.1.1 16-BIT ARCHITECTURE

Central to all PIC24F devices is the 16-bit modified Harvard architecture, first introduced with Microchip's dsPIC® digital signal controllers. The PIC24F CPU core offers a wide range of enhancements, such as:

- 16-bit data and 24-bit address paths with the ability to move information between data and memory spaces
- Linear addressing of up to 12 Mbytes (program space) and 64 Kbytes (data)
- A 16-element working register array with built-in software stack support
- A 17 x 17 hardware multiplier with support for integer math
- Hardware support for 32 by 16-bit division
- An instruction set that supports multiple addressing modes and is optimized for high-level languages, such as 'C'
- Operational performance up to 16 MIPS

1.1.2 POWER-SAVING TECHNOLOGY

All of the devices in the PIC24FJ64GA104 family incorporate a range of features that can significantly reduce power consumption during operation. Key items include:

- **On-the-Fly Clock Switching:** The device clock can be changed under software control to the Timer1 source or the internal, Low-Power Internal RC Oscillator during operation, allowing the user to incorporate power-saving ideas into their software designs.
- **Doze Mode Operation:** When timing-sensitive applications, such as serial communications, require the uninterrupted operation of peripherals, the CPU clock speed can be selectively reduced, allowing incremental power savings without missing a beat.
- **Instruction-Based Power-Saving Modes:** There are three instruction-based power-saving modes:
  - Idle Mode – The core is shut down while leaving the peripherals active.
  - Sleep Mode – The core and peripherals that require the system clock are shut down, leaving the peripherals active that use their own clock or the clock from other devices.
  - Deep Sleep Mode – The core, peripherals (except RTCC and DSWDT), Flash and SRAM are shut down for optimal current savings to extend battery life for portable applications.

1.1.3 OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC24FJ64GA104 family offer five different oscillator options, allowing users a range of choices in developing application hardware. These include:

- Two Crystal modes using crystals or ceramic resonators.
- Two External Clock modes offering the option of a divide-by-2 clock output.
- A Fast Internal Oscillator (FRC) with a nominal 8 MHz output, which can also be divided under software control to provide clock speeds as low as 31 kHz.
- A Phase Lock Loop (PLL) frequency multiplier available to the external oscillator modes and the FRC Oscillator, which allows clock speeds of up to 32 MHz.
- A separate Low-Power Internal RC Oscillator (LPRC) with a fixed 31 kHz output, which provides a low-power option for timing-insensitive applications.

The internal oscillator block also provides a stable reference source for the Fail-Safe Clock Monitor. This option constantly monitors the main clock source against a reference signal provided by the internal oscillator and enables the controller to switch to the internal oscillator, allowing for continued low-speed operation or a safe application shutdown.

1.1.4 EASY MIGRATION

Regardless of the memory size, all devices share the same rich set of peripherals, allowing for a smooth migration path as applications grow and evolve. The consistent pinout scheme used throughout the entire family also aids in migrating from one device to the next larger device.

The PIC24F family is pin-compatible with devices in the dsPIC33 family, and shares some compatibility with the pinout schema for PIC18 and dsPIC30 devices. This extends the ability of applications to grow from the relatively simple, to the powerful and complex, yet still selecting a Microchip device.

1.2 Other Special Features

- **Peripheral Pin Select:** The Peripheral Pin Select feature allows most digital peripherals to be mapped over a fixed set of digital I/O pins. Users may independently map the input and/or output of any one of the many digital peripherals to any one of the I/O pins.
- **Communications:** The PIC24FJ64GA104 family incorporates a range of serial communication peripherals to handle a range of application requirements. There are two independent I^{2}C™ modules that support both Master and Slave modes of operation. Devices also have, through the Peripheral Pin Select (PPS) feature, two independent UARTs with built-in IrDA® encoder/decoders and two SPI modules.
- **Analog Features:** All members of the PIC24FJ64GA104 family include a 10-bit A/D Converter module and a triple comparator module. The A/D module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated without waiting for a sampling period, as well as faster sampling speeds. The comparator module includes three analog comparators that are configurable for a wide range of operations.
- **CTMU Interface:** This module provides a convenient method for precision time measurement and pulse generation, and can serve as an interface for capacitive sensors.
- **Parallel Master/Enhanced Parallel Slave Port:** One of the general purpose I/O ports can be reconfigured for enhanced parallel data communications. In this mode, the port can be configured for both master and slave operations, and supports 8-bit and 16-bit data transfers with up to 12 external address lines in Master modes.
- **Real-Time Clock/Calendar:** This module implements a full-featured clock and calendar with alarm functions in hardware, freeing up timer resources and program memory space for the use of the core application.

1.3 Details on Individual Family Members

Devices in the PIC24FJ64GA104 family are available in 28-pin and 44-pin packages. The general block diagram for all devices is shown in Figure 1-1.

The devices are differentiated from each other in several ways:

- Flash Program Memory:
  - PIC24FJ32GA1 devices – 32 Kbytes
  - PIC24FJ64GA1 devices – 64 Kbytes
- Available I/O Pins and Ports:
  - 28-pin devices – 21 pins on two ports
  - 44-pin devices – 35 pins on three ports
- Available Interrupt-on-Change Notification (ICN) Inputs:
  - 28-pin devices – 21
  - 44-pin devices – 31
- Available Remappable Pins:
  - 28-pin devices – 16 pins
  - 44-pin devices – 26 pins
- Available PMP Address Pins:
  - 28-pin devices – 3 pins
  - 44-pin devices – 12 pins
- Available A/D Input Channels:
  - 28-pin devices – 10 pins
  - 44-pin devices – 13 pins

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

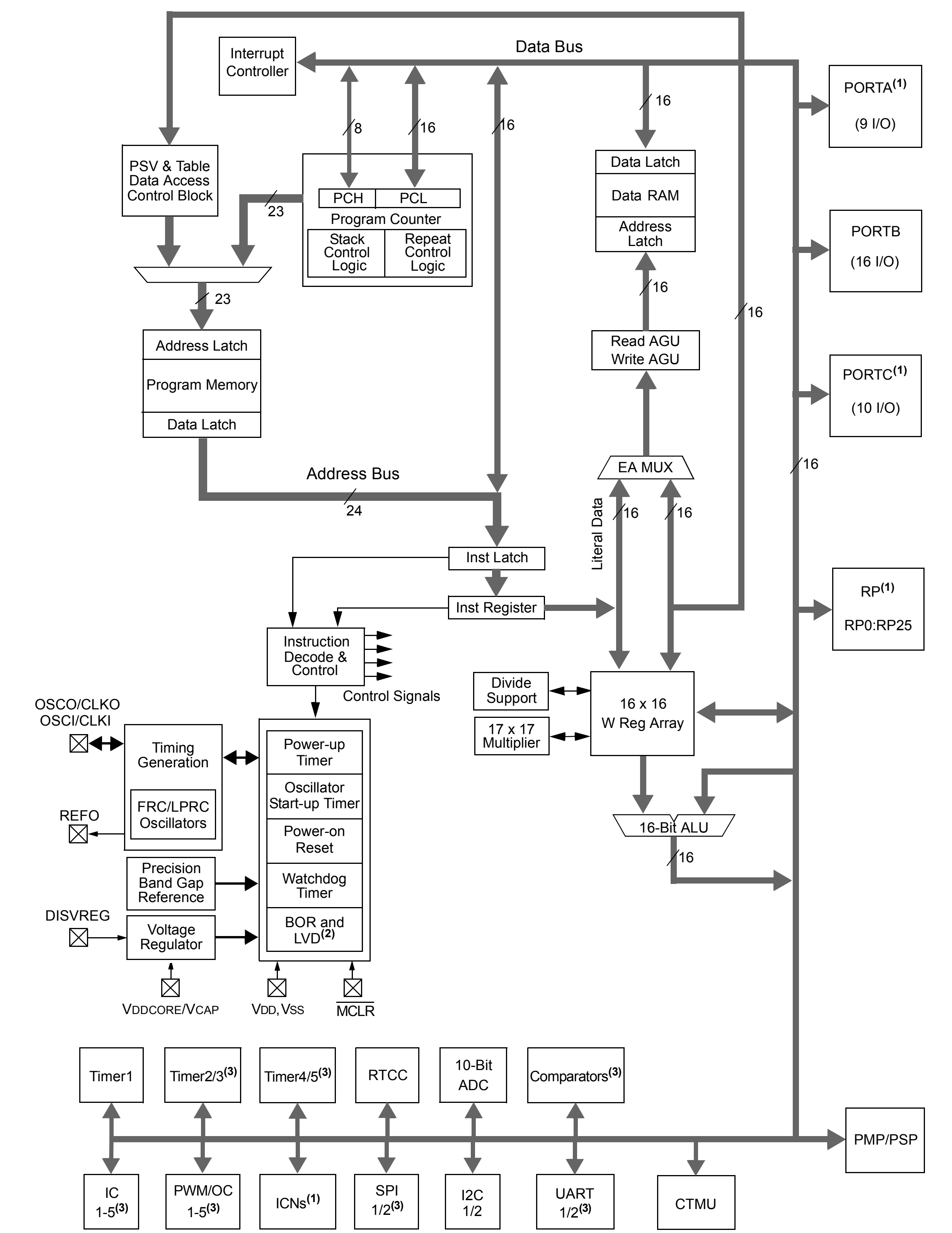
A list of the pin features available on the PIC24FJ64GA104 family devices, sorted by function, is shown in Table 1-2. Note that this table shows the pin location of individual peripheral features and not how they are multiplexed on the same pin. This information is provided in the pinout diagrams in the beginning of this data sheet. Multiplexed features are sorted by the priority given to a feature, with the highest priority peripheral being listed first.

TABLE 1-1: DEVICE FEATURES FOR THE PIC24FJ64GA104 FAMILY

<table>
<tr><th>Features</th><th>PIC24FJ32GA102</th><th>PIC24FJ64GA102</th><th>PIC24FJ32GA104</th><th>PIC24FJ64GA104</th></tr>
<tr><td>Operating Frequency</td><td colspan="4">DC – 32 MHz</td></tr>
<tr><td>Program Memory (bytes)</td><td>32K</td><td>64K</td><td>32K</td><td>64K</td></tr>
<tr><td>Program Memory (instructions)</td><td>11,008</td><td>22,016</td><td>11,008</td><td>22,016</td></tr>
<tr><td>Data Memory (bytes)</td><td colspan="4">8,192</td></tr>
<tr><td>Interrupt Sources (soft vectors/ NMI traps)</td><td colspan="4">45 (41/4)</td></tr>
<tr><td>I/O Ports</td><td colspan="2">Ports A and B</td><td colspan="2">Ports A, B, C</td></tr>
<tr><td>Total I/O Pins</td><td colspan="2">21</td><td colspan="2">35</td></tr>
<tr><td>Remappable Pins</td><td colspan="2">16</td><td colspan="2">26</td></tr>
<tr><td>Timers:
Total Number (16-bit)</td><td colspan="4">5[1]</td></tr>
<tr><td>32-Bit (from paired 16-bit timers)</td><td colspan="4">2</td></tr>
<tr><td>Input Capture Channels</td><td colspan="4">5[1]</td></tr>
<tr><td>Output Compare/PWM Channels</td><td colspan="4">5[1]</td></tr>
<tr><td>Input Change Notification Interrupt</td><td colspan="2">21</td><td colspan="2">31</td></tr>
<tr><td>Serial Communications:
UART</td><td colspan="4">2[1]</td></tr>
<tr><td>SPI (3-wire/4-wire)</td><td colspan="4">2[1]</td></tr>
<tr><td>I^{2}C™</td><td colspan="4">2</td></tr>
<tr><td>Parallel Communications (PMP/PSP)</td><td colspan="4">Yes</td></tr>
<tr><td>JTAG Boundary Scan</td><td colspan="4">Yes</td></tr>
<tr><td>10-Bit Analog-to-Digital Module (input channels)</td><td colspan="2">10</td><td colspan="2">13</td></tr>
<tr><td>Analog Comparators</td><td colspan="4">3</td></tr>
<tr><td>CTMU Interface</td><td colspan="4">Yes</td></tr>
<tr><td>Resets (and delays)</td><td colspan="4">POR, BOR, RESET Instruction, $\overline{\text{MCLR}}$, WDT; Illegal Opcode, REPEAT Instruction, Hardware Traps, Configuration Word Mismatch (PWRT, OST, PLL Lock)</td></tr>
<tr><td>Instruction Set</td><td colspan="4">76 Base Instructions, Multiple Addressing Mode Variations</td></tr>
<tr><td>Packages</td><td colspan="2">28-Pin QFN, SOIC, SSOP and SPDIP</td><td colspan="2">44-Pin QFN and TQFP</td></tr>
</table>

Note 1: Peripherals are accessible through remappable pins.

FIGURE 1-1: PIC24FJ64GA104 FAMILY GENERAL BLOCK DIAGRAM



Note 1: Not all I/O pins or features are implemented on all device pinout configurations. See Table 1-2 for specific implementations by pin count.
2: BOR functionality is provided when the on-board voltage regulator is enabled.
3: These peripheral I/Os are only accessible through remappable pins.

TABLE 1-2: PIC24FJ64GA104 FAMILY PINOUT DESCRIPTIONS

Function	Pin Number			I/O	Input Buffer	Description
	28-Pin SPDIP/ SOIC/SSOP	28-Pin QFN	44-Pin QFN/ TQFP			
AN0	2	27	19	I	ANA	A/D Analog Inputs.
AN1	3	28	20	I	ANA	
AN2	4	1	21	I	ANA	
AN3	5	2	22	I	ANA	
AN4	6	3	23	I	ANA	
AN5	7	4	24	I	ANA	
AN6	—	—	25	I	ANA	
AN7	—	—	26	I	ANA	
AN8	—	—	27	I	ANA	
AN9	26	23	15	I	ANA	
AN10	25	22	14	I	ANA	
AN11	24	21	11	I	ANA	
AN12	23	20	10	I	ANA	
ASCL1	15	12	42	I/O	I^2C	Alternate I2C1 Synchronous Serial Clock Input/Output.
ASDA1	14	11	41	I/O	I^2C	Alternate I2C1 Synchronous Serial Data Input/Output.
AVDD	—	—	17	P	—	Positive Supply for Analog modules.
AVSS	—	—	16	P	—	Ground Reference for Analog modules.
C1INA	7	4	24	I	ANA	Comparator 1 Input A.
C1INB	6	3	23	I	ANA	Comparator 1 Input B.
C1INC	24	21	11	I	ANA	Comparator 1 Input C.
C1IND	9	6	30	I	ANA	Comparator 1 Input D.
C2INA	5	2	22	I	ANA	Comparator 2 Input A.
C2INB	4	1	21	I	ANA	Comparator 2 Input B.
C2INC	12	9	34	I	ANA	Comparator 2 Input C.
C2IND	11	8	33	I	ANA	Comparator 2 Input D.
C3INA	26	23	15	I	ANA	Comparator 3 Input A.
C3INB	25	22	14	I	ANA	Comparator 3 Input B.
C3INC	2	27	19	I	ANA	Comparator 3 Input C.
C3IND	3	28	20	I	ANA	Comparator 3 Input D.
CLKI	9	6	30	I	ANA	Main Clock Input Connection.
CLKO	10	7	31	O	—	System Clock Output.

Legend: TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; I^2C™ = I^2C/SMBus input buffer

TABLE 1-2: PIC24FJ64GA104 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number 28-Pin SPDIP/ SOIC/SSOP	28-Pin QFN	44-Pin QFN/ TQFP	I/O	Input Buffer	Description
CN0	12	9	34	I	ST	Interrupt-on-Change Inputs.
CN1	11	8	33	I	ST	
CN2	2	27	19	I	ST	
CN3	3	28	20	I	ST	
CN4	4	1	21	I	ST	
CN5	5	2	22	I	ST	
CN6	6	3	23	I	ST	
CN7	7	4	24	I	ST	
CN8	—	—	25	I	ST	
CN9	—	—	26	I	ST	
CN10	—	—	27	I	ST	
CN11	26	23	15	I	ST	
CN12	25	22	14	I	ST	
CN13	24	21	11	I	ST	
CN14	23	20	10	I	ST	
CN15	22	19	9	I	ST	
CN16	21	18	8	I	ST	
CN17	—	—	3	I	ST	
CN18	—	—	2	I	ST	
CN19	—	—	5	I	ST	
CN20	—	—	4	I	ST	
CN21	18	15	1	I	ST	
CN22	17	14	44	I	ST	
CN23	16	13	43	I	ST	
CN24	15	12	42	I	ST	
CN25	—	—	37	I	ST	
CN26	—	—	38	I	ST	
CN27	14	11	41	I	ST	
CN28	—	—	36	I	ST	
CN29	10	7	31	I	ST	
CN30	9	6	30	I	ST	
CTED1	2	27	19	I	ANA	CTMU External Edge Input 1.
CTED2	3	28	20	I	ANA	CTMU External Edge Input 2.
CVREF	25	22	14	O	—	Comparator Voltage Reference Output.
DISVREG	19	16	6	I	ST	Voltage Regulator Disable.

Legend: TTL = TTL input buffer; ANA = Analog level input/output; ST = Schmitt Trigger input buffer; I^2C™ = I^2C/SMBus input buffer

TABLE 1-2: PIC24FJ64GA104 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	28-Pin SPDIP/ SOIC/SSOP	28-Pin QFN	44-Pin QFN/ TQFP			
INT0	16	13	43	I	ST	External Interrupt Input.
$\overline{MCLR}$	1	26	18	I	ST	Master Clear (device Reset) Input. This line is brought low to cause a Reset.
OSCI	9	6	30	I	ANA	Main Oscillator Input Connection.
OSCO	10	7	31	O	ANA	Main Oscillator Output Connection.
PGEC1	5	2	22	I/O	ST	In-Circuit Debugger/Emulator/ICSP™ Programming Clock.
PGED1	4	1	21	I/O	ST	In-Circuit Debugger/Emulator/ICSP Programming Data.
PGEC2	22	19	9	I/O	ST	In-Circuit Debugger/Emulator/ICSP Programming Clock.
PGED2	21	18	8	I/O	ST	In-Circuit Debugger/Emulator/ICSP Programming Data.
PGEC3	15	12	42	I/O	ST	In-Circuit Debugger/Emulator/ICSP Programming Clock.
PGED3	14	11	41	I/O	ST	In-Circuit Debugger/Emulator/ICSP Programming Data.
PMA0	10	7	3	I/O	ST	Parallel Master Port Address Bit 0 Input (Buffered Slave modes) and Output (Master modes).
PMA1	12	9	2	I/O	ST	Parallel Master Port Address Bit 1 Input (Buffered Slave modes) and Output (Master modes).
PMA2	—	—	27	O	—	Parallel Master Port Address (Demultiplexed Master modes).
PMA3	—	—	38	O	—	
PMA4	—	—	37	O	—	
PMA5	—	—	4	O	—	
PMA6	—	—	5	O	—	
PMA7	—	—	13	O	—	
PMA8	—	—	32	O	—	
PMA9	—	—	35	O	—	
PMA10	—	—	12	O	—	
PMCS1	26	23	15	I/O	ST/TTL	Parallel Master Port Chip Select 1 Strobe/Address Bit 15.
PMBE	11	8	36	O	—	Parallel Master Port Byte Enable Strobe.
PMD0	23	20	10	I/O	ST/TTL	Parallel Master Port Data (Demultiplexed Master mode) or Address/Data (Multiplexed Master modes).
PMD1	22	19	9	I/O	ST/TTL	
PMD2	21	18	8	I/O	ST/TTL	
PMD3	18	15	1	I/O	ST/TTL	
PMD4	17	14	44	I/O	ST/TTL	
PMD5	16	13	43	I/O	ST/TTL	
PMD6	15	12	42	I/O	ST/TTL	
PMD7	14	11	41	I/O	ST/TTL	
PMRD	24	21	11	O	—	Parallel Master Port Read Strobe.
PMWR	25	22	14	O	—	Parallel Master Port Write Strobe.

Legend: TTL = TTL input buffer; ANA = Analog level input/output; ST = Schmitt Trigger input buffer; I^2C™ = I^2C/SMBus input buffer

TABLE 1-2: PIC24FJ64GA104 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	28-Pin SPDIP/SOIC/SSOP	28-Pin QFN	44-Pin QFN/TQFP			
RA0	2	27	19	I/O	ST	PORTA Digital I/O.
RA1	3	28	20	I/O	ST	
RA2	9	6	30	I/O	ST	
RA3	10	7	31	I/O	ST	
RA4	12	9	34	I/O	ST	
RA7	—	—	13	I/O	ST	
RA8	—	—	32	I/O	ST	
RA9	—	—	35	I/O	ST	
RA10	—	—	12	I/O	ST	
RB0	4	1	21	I/O	ST	PORTB Digital I/O.
RB1	5	2	22	I/O	ST	
RB2	6	3	23	I/O	ST	
RB3	7	4	24	I/O	ST	
RB4	11	8	33	I/O	ST	
RB5	14	11	41	I/O	ST	
RB6	15	12	42	I/O	ST	
RB7	16	13	43	I/O	ST	
RB8	17	14	44	I/O	ST	
RB9	18	15	1	I/O	ST	
RB10	21	18	8	I/O	ST	
RB11	22	19	9	I/O	ST	
RB12	23	20	10	I/O	ST	
RB13	24	21	11	I/O	ST	
RB14	25	22	14	I/O	ST	
RB15	26	23	15	I/O	ST	
RC0	—	—	25	I/O	ST	PORTC Digital I/O.
RC1	—	—	26	I/O	ST	
RC2	—	—	27	I/O	ST	
RC3	—	—	36	I/O	ST	
RC4	—	—	37	I/O	ST	
RC5	—	—	38	I/O	ST	
RC6	—	—	2	I/O	ST	
RC7	—	—	3	I/O	ST	
RC8	—	—	4	I/O	ST	
RC9	—	—	5	I/O	ST	
REFO	24	21	11	O	—	Reference Clock Output.

Legend: TTL = TTL input buffer; ST = Schmitt Trigger input buffer
ANA = Analog level input/output; I^2C™ = I^2C/SMBus input buffer

TABLE 1-2: PIC24FJ64GA104 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	28-Pin SPDIP/ SOIC/SSOP	28-Pin QFN	44-Pin QFN/ TQFP			
RP0	4	1	21	I/O	ST	Remappable Peripheral (input or output).
RP1	5	2	22	I/O	ST	
RP2	6	3	23	I/O	ST	
RP3	7	4	24	I/O	ST	
RP4	11	8	33	I/O	ST	
RP5	14	11	41	I/O	ST	
RP6	15	12	42	I/O	ST	
RP7	16	13	43	I/O	ST	
RP8	17	14	44	I/O	ST	
RP9	18	15	1	I/O	ST	
RP10	21	18	8	I/O	ST	
RP11	22	19	9	I/O	ST	
RP12	23	20	10	I/O	ST	
RP13	24	21	11	I/O	ST	
RP14	25	22	14	I/O	ST	
RP15	26	23	15	I/O	ST	
RP16	—	—	25	I/O	ST	
RP17	—	—	26	I/O	ST	
RP18	—	—	27	I/O	ST	
RP19	—	—	36	I/O	ST	
RP20	—	—	37	I/O	ST	
RP21	—	—	38	I/O	ST	
RP22	—	—	2	I/O	ST	
RP23	—	—	3	I/O	ST	
RP24	—	—	4	I/O	ST	
RP25	—	—	5	I/O	ST	
RTCC	25	22	14	O	—	Real-Time Clock Alarm/Seconds Pulse Output.
SCL1	17	14	44	I/O	I^2C	I2C1 Synchronous Serial Clock Input/Output.
SCL2	7	4	24	I/O	I^2C	I2C2 Synchronous Serial Clock Input/Output.
SDA1	18	15	1	I/O	I^2C	I2C1 Data Input/Output.
SDA2	6	3	23	I/O	I^2C	I2C2 Data Input/Output.
SOSCI	11	8	33	I	ANA	Secondary Oscillator/Timer1 Clock Input.
SOSCO	12	9	34	O	ANA	Secondary Oscillator/Timer1 Clock Output.
T1CK	12	9	34	I	ST	Timer1 Clock Input.
TCK	17	14	13	I	ST	JTAG Test Clock Input.
TDI	21	18	35	I	ST	JTAG Test Data Input.
TDO	18	15	32	O	—	JTAG Test Data Output.
TMS	22	19	12	I	ST	JTAG Test Mode Select Input.

Legend: TTL = TTL input buffer; ANA = Analog level input/output; ST = Schmitt Trigger input buffer; I^2C™ = I^2C/SMBus input buffer

TABLE 1-2: PIC24FJ64GA104 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	28-Pin SPDIP/ SOIC/SSOP	28-Pin QFN	44-Pin QFN/ TQFP			
VCAP	20	17	7	P	—	External Filter Capacitor Connection (regulator enabled).
VDD	13, 28	10, 25	28, 40	P	—	Positive Supply for Peripheral Digital Logic and I/O Pins.
VDDCORE	20	17	7	P	—	Positive Supply for Microcontroller Core Logic (regulator disabled).
VREF-	3	28	20	I	ANA	A/D and Comparator Reference Voltage (low) Input.
VREF+	2	27	19	I	ANA	A/D and Comparator Reference Voltage (high) Input.
VSS	8, 27	5, 24	29, 39	P	—	Ground Reference for Logic and I/O Pins.

Legend: TTL = TTL input buffer
ANA = Analog level input/output
ST = Schmitt Trigger input buffer
I^2C™ = I^2C/SMBus input buffer

2.0 GUIDELINES FOR GETTING STARTED WITH 16-BIT MICROCONTROLLERS

2.1 Basic Connection Requirements

Getting started with the PIC24FJ64GA104 family of 16-bit microcontrollers requires attention to a minimal set of device pin connections before proceeding with development.

The following pins must always be connected:

- All VDD and VSS pins (see **Section 2.2 “Power Supply Pins”**)
- All AVDD and AVSS pins, regardless of whether or not the analog device features are used (see **Section 2.2 “Power Supply Pins”**)
- MCLR pin (see **Section 2.3 “Master Clear (MCLR) Pin”**)
- ENVREG/DISVREG and VCAP/VDDCORE pins (PIC24FJ devices only) (see **Section 2.4 “Voltage Regulator Pins (ENVREG/DISVREG and VCAP/VDDCORE)”**)

These pins must also be connected if they are being used in the end application:

- PGECx/PGEDx pins used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes (see **Section 2.5 “ICSP Pins”**)
- OSCI and OSCO pins when an external oscillator source is used (see **Section 2.6 “External Oscillator Pins”**)

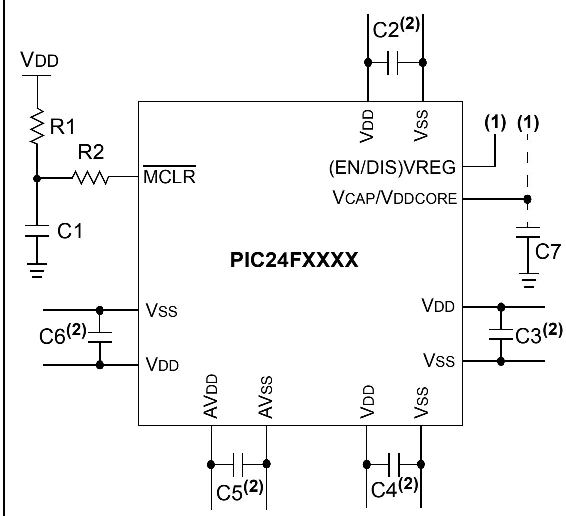
Additionally, the following pins may be required:

- VREF+/VREF- pins used when external voltage reference for analog modules is implemented

Note:	The AVDD and AVSS pins must always be connected, regardless of whether any of the analog modules are being used.

The minimum mandatory connections are shown in Figure 2-1.

FIGURE 2-1: RECOMMENDED MINIMUM CONNECTIONS



Key (all values are recommendations):

C1 through C6: 0.1 µF, 20V ceramic

C7: 10 µF, 6.3V or greater, tantalum or ceramic

R1: 10 kΩ

R2: 100Ω to 470Ω

Note 1: See **Section 2.4 “Voltage Regulator Pins (ENVREG/DISVREG and VCAP/VDDCORE)”** for explanation of ENVREG/DISVREG pin connections.

2: The example shown is for a PIC24F device with five VDD/VSS and AVDD/AVSS pairs. Other devices may have more or less pairs; adjust the number of decoupling capacitors appropriately.

2.2 Power Supply Pins

2.2.1 DECOUPLING CAPACITORS

The use of decoupling capacitors on every pair of power supply pins, such as V_{DD}, V_{SS}, AV_{DD} and AV_{SS} is required.

Consider the following criteria when using decoupling capacitors:

- **Value and type of capacitor:** A 0.1 µF (100 nF), 10-20V capacitor is recommended. The capacitor should be a low-ESR device with a resonance frequency in the range of 200 MHz and higher. Ceramic capacitors are recommended.
- **Placement on the printed circuit board:** The decoupling capacitors should be placed as close to the pins as possible. It is recommended to place the capacitors on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer on the PCB using a via; however, ensure that the trace length from the pin to the capacitor is no greater than 0.25 inch (6 mm).
- **Handling high-frequency noise:** If the board is experiencing high-frequency noise (upward of tens of MHz), add a second ceramic type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 µF to 0.001 µF. Place this second capacitor next to each primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible (e.g., 0.1 µF in parallel with 0.001 µF).
- **Maximizing performance:** On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum, thereby reducing PCB trace inductance.

2.2.2 TANK CAPACITORS

On boards with power traces running longer than six inches in length, it is suggested to use a tank capacitor for integrated circuits including microcontrollers to supply a local power source. The value of the tank capacitor should be determined based on the trace resistance that connects the power supply source to the device, and the maximum current drawn by the device in the application. In other words, select the tank capacitor so that it meets the acceptable voltage sag at the device. Typical values range from 4.7 µF to 47 µF.

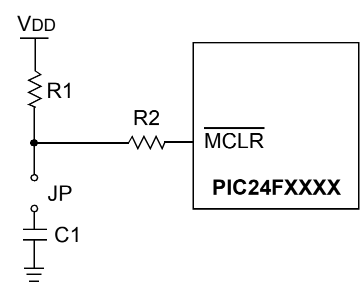
2.3 Master Clear ($\overline{\text{MCLR}}$) Pin

The $\overline{\text{MCLR}}$ pin provides two specific device functions: device Reset, and device programming and debugging. If programming and debugging are not required in the end application, a direct connection to V_{DD} may be all that is required. The addition of other components, to help increase the application's resistance to spurious Resets from voltage sags, may be beneficial. A typical configuration is shown in Figure 2-1. Other circuit designs may be implemented, depending on the application's requirements.

During programming and debugging, the resistance and capacitance that can be added to the pin must be considered. Device programmers and debuggers drive the $\overline{\text{MCLR}}$ pin. Consequently, specific voltage levels (V_{IH} and V_{IL}) and fast signal transitions must not be adversely affected. Therefore, specific values of R1 and C1 will need to be adjusted based on the application and PCB requirements. For example, it is recommended that the capacitor, C1, be isolated from the $\overline{\text{MCLR}}$ pin during programming and debugging operations by using a jumper (Figure 2-2). The jumper is replaced for normal run-time operations.

Any components associated with the $\overline{\text{MCLR}}$ pin should be placed within 0.25 inch (6 mm) of the pin.

FIGURE 2-2: EXAMPLE OF $\overline{\text{MCLR}}$ PIN CONNECTIONS



Note 1: R1 ≤ 10 kΩ is recommended. A suggested starting value is 10 kΩ. Ensure that the $\overline{\text{MCLR}}$ pin V_{IH} and V_{IL} specifications are met.

2: R2 ≤ 470Ω will limit any current flowing into $\overline{\text{MCLR}}$ from the external capacitor, C, in the event of $\overline{\text{MCLR}}$ pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS). Ensure that the $\overline{\text{MCLR}}$ pin V_{IH} and V_{IL} specifications are met.

2.4 Voltage Regulator Pins (ENVREG/DISVREG and VCAP/VDDCORE)

Note:	This section applies only to PIC24FJ devices with an on-chip voltage regulator.

The on-chip voltage regulator enable/disable pin (ENVREG or DISVREG, depending on the device family) must always be connected directly to either a supply voltage or to ground. The particular connection is determined by whether or not the regulator is to be used:

- For ENVREG, tie to VDD to enable the regulator, or to ground to disable the regulator
- For DISVREG, tie to ground to enable the regulator or to VDD to disable the regulator

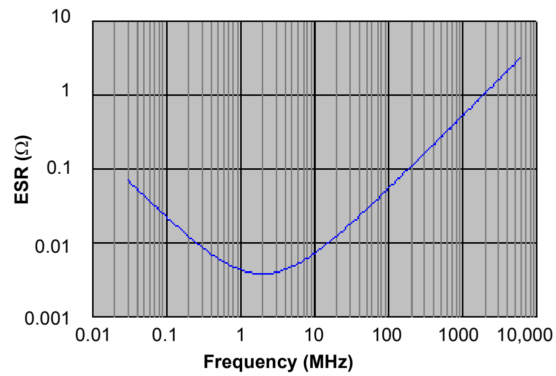
Refer to **Section 25.2 "On-Chip Voltage Regulator"** for details on connecting and using the on-chip regulator.

When the regulator is enabled, a low-ESR (<5Ω) capacitor is required on the VCAP/VDDCORE pin to stabilize the voltage regulator output voltage. The VCAP/VDDCORE pin must not be connected to VDD, and must use a capacitor of 10 µF connected to ground. The type can be ceramic or tantalum. A suitable example is the Murata GRM21BF50J106ZE01 (10 µF, 6.3V) or equivalent. Designers may use Figure 2-3 to evaluate ESR equivalence of candidate devices.

The placement of this capacitor should be close to VCAP/VDDCORE. It is recommended that the trace length not exceed 0.25 inch (6 mm). Refer to **Section 28.0 "Electrical Characteristics"** for additional information.

When the regulator is disabled, the VCAP/VDDCORE pin must be tied to a voltage supply at the VDDCORE level. Refer to **Section 28.0 "Electrical Characteristics"** for information on VDD and VDDCORE.

FIGURE 2-3: FREQUENCY vs. ESR PERFORMANCE FOR SUGGESTED VCAP



Note: Data for Murata GRM21BF50J106ZE01 shown. Measurements at 25°C, 0V DC bias.

2.5 ICSP Pins

The PGECx and PGEDx pins are used for In-Circuit Serial Programming (ICSP) and debugging purposes. It is recommended to keep the trace length between the ICSP connector and the ICSP pins on the device as short as possible. If the ICSP connector is expected to experience an ESD event, a series resistor is recommended, with the value in the range of a few tens of ohms, not to exceed 100Ω.

Pull-up resistors, series diodes and capacitors on the PGECx and PGEDx pins are not recommended as they will interfere with the programmer/debugger communications to the device. If such discrete components are an application requirement, they should be removed from the circuit during programming and debugging. Alternatively, refer to the AC/DC characteristics and timing requirements information in the respective device Flash programming specification for information on capacitive loading limits and pin input voltage high (VIH) and input low (VIL) requirements.

For device emulation, ensure that the "Communication Channel Select" (i.e., PGECx/PGEDx pins) programmed into the device matches the physical connections for the ICSP to the Microchip debugger/emulator tool.

For more information on available Microchip development tools connection requirements, refer to **Section 26.0 "Development Support"**.

2.6 External Oscillator Pins

Many microcontrollers have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency Secondary Oscillator (refer to **Section 8.0 "Oscillator Configuration"** for details).

The oscillator circuit should be placed on the same side of the board as the device. Place the oscillator circuit close to the respective oscillator pins with no more than 0.5 inch (12 mm) between the circuit components and the pins. The load capacitors should be placed next to the oscillator itself, on the same side of the board.

Use a grounded copper pour around the oscillator circuit to isolate it from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed.

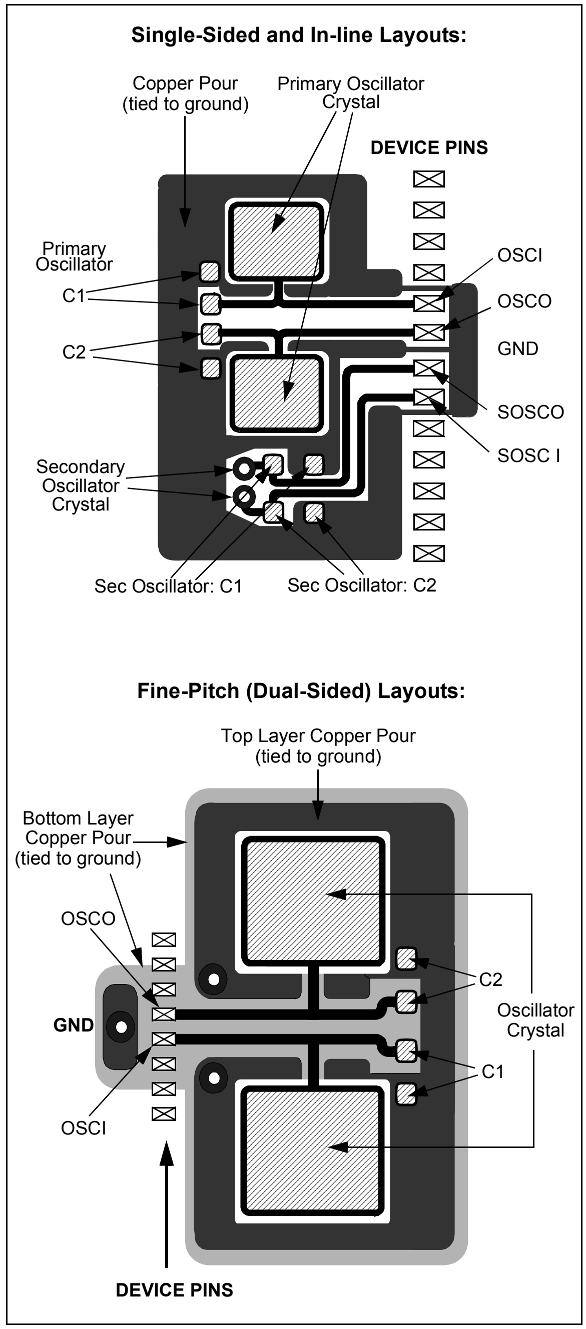
Layout suggestions are shown in Figure 2-4. In-line packages may be handled with a single-sided layout that completely encompasses the oscillator pins. With fine-pitch packages, it is not always possible to completely surround the pins and components. A suitable solution is to tie the broken guard sections to a mirrored ground layer. In all cases, the guard trace(s) must be returned to ground.

In planning the application's routing and I/O assignments, ensure that adjacent port pins and other signals in close proximity to the oscillator are benign (i.e., free of high frequencies, short rise and fall times and other similar noise).

For additional information and design guidance on oscillator circuits, please refer to these Microchip Application Notes, available at the corporate web site (www.microchip.com):

- *AN826, "Crystal Oscillator Basics and Crystal Selection for rfPIC™ and PICmicro® Devices"*
- *AN849, "Basic PICmicro® Oscillator Design"*
- *AN943, "Practical PICmicro® Oscillator Analysis and Design"*
- *AN949, "Making Your Oscillator Work"*

FIGURE 2-4: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT



2.7 Configuration of Analog and Digital Pins During ICSP Operations

If an ICSP compliant emulator is selected as a debugger, it automatically initializes all of the A/D input pins (ANx) as “digital” pins. Depending on the particular device, this is done by setting all bits in the ADnPCFG register(s), or clearing all bit in the ANSx registers.

All PIC24F devices will have either one or more ADnPCFG registers or several ANSx registers (one for each port); no device will have both. Refer to **Section 21.0 “10-Bit High-Speed A/D Converter”**) for more specific information.

The bits in these registers that correspond to the A/D pins that initialized the emulator must not be changed by the user application firmware; otherwise, communication errors will result between the debugger and the device.

If your application needs to use certain A/D pins as analog input pins during the debug session, the user application must modify the appropriate bits during initialization of the ADC module, as follows:

- For devices with an ADnPCFG register, clear the bits corresponding to the pin(s) to be configured as analog. Do not change any other bits, particularly those corresponding to the PGECx/PGEDx pair, at any time.
- For devices with ANSx registers, set the bits corresponding to the pin(s) to be configured as analog. Do not change any other bits, particularly those corresponding to the PGECx/PGEDx pair, at any time.

When a Microchip debugger/emulator is used as a programmer, the user application firmware must correctly configure the ADnPCFG or ANSx registers. Automatic initialization of this register is only done during debugger operation. Failure to correctly configure the register(s) will result in all A/D pins being recognized as analog input pins, resulting in the port value being read as a logic '0', which may affect user application functionality.

2.8 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic low state. Alternatively, connect a 1 kΩ to 10 kΩ resistor to VSS on unused pins and drive the output to logic low.

NOTES:

3.0 CPU

> **Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 2. "CPU"** (DS39703).

The PIC24F CPU has a 16-bit (data), modified Harvard architecture with an enhanced instruction set and a 24-bit instruction word with a variable length opcode field. The Program Counter (PC) is 23 bits wide and addresses up to 4M instructions of user program memory space. A single-cycle instruction prefetch mechanism is used to help maintain throughput and provides predictable execution. All instructions execute in a single cycle, with the exception of instructions that change the program flow, the double-word move (`MOV.D`) instruction and the table instructions. Overhead-free program loop constructs are supported using the `REPEAT` instructions, which are interruptible at any point.

PIC24F devices have sixteen, 16-bit working registers in the programmer's model. Each of the working registers can act as a data, address or address offset register. The 16th working register (W15) operates as a Software Stack Pointer for interrupts and calls.

The upper 32 Kbytes of the data space memory map can optionally be mapped into program space at any 16K word boundary defined by the 8-bit Program Space Visibility Page Address (PSVPAG) register. The program to data space mapping feature lets any instruction access program space as if it were data space.

The Instruction Set Architecture (ISA) has been significantly enhanced beyond that of the PIC18, but maintains an acceptable level of backward compatibility. All PIC18 instructions and addressing modes are supported either directly or through simple macros. Many of the ISA enhancements have been driven by compiler efficiency needs.

The core supports Inherent (no operand), Relative, Literal, Memory Direct and three groups of addressing modes. All modes support Register Direct and various Register Indirect modes. Each group offers up to seven addressing modes. Instructions are associated with predefined addressing modes depending upon their functional requirements.

For most instructions, the core is capable of executing a data (or program data) memory read, a working register (data) read, a data memory write and a program (instruction) memory read per instruction cycle. As a result, three parameter instructions can be supported, allowing trinary operations (that is, A + B = C) to be executed in a single cycle.

A high-speed, 17-bit by 17-bit multiplier has been included to significantly enhance the core arithmetic capability and throughput. The multiplier supports Signed, Unsigned and Mixed mode, 16-bit by 16-bit or 8-bit by 8-bit integer multiplication. All multiply instructions execute in a single cycle.

The 16-bit ALU has been enhanced with integer divide assist hardware that supports an iterative non-restoring divide algorithm. It operates in conjunction with the `REPEAT` instruction looping mechanism and a selection of iterative divide instructions to support 32-bit (or 16-bit), divided by 16-bit, integer signed and unsigned division. All divide operations require 19 cycles to complete, but are interruptible at any cycle boundary.

The PIC24F has a vectored exception scheme with up to 8 sources of non-maskable traps and up to 118 interrupt sources. Each interrupt source can be assigned to one of seven priority levels.

A block diagram of the CPU is shown in Figure 3-1.

3.1 Programmer's Model

The programmer's model for the PIC24F is shown in Figure 3-2. All registers in the programmer's model are memory mapped and can be manipulated directly by instructions. A description of each register is provided in Table 3-1. All registers associated with the programmer's model are memory mapped.

FIGURE 3-1: PIC24F CPU CORE BLOCK DIAGRAM

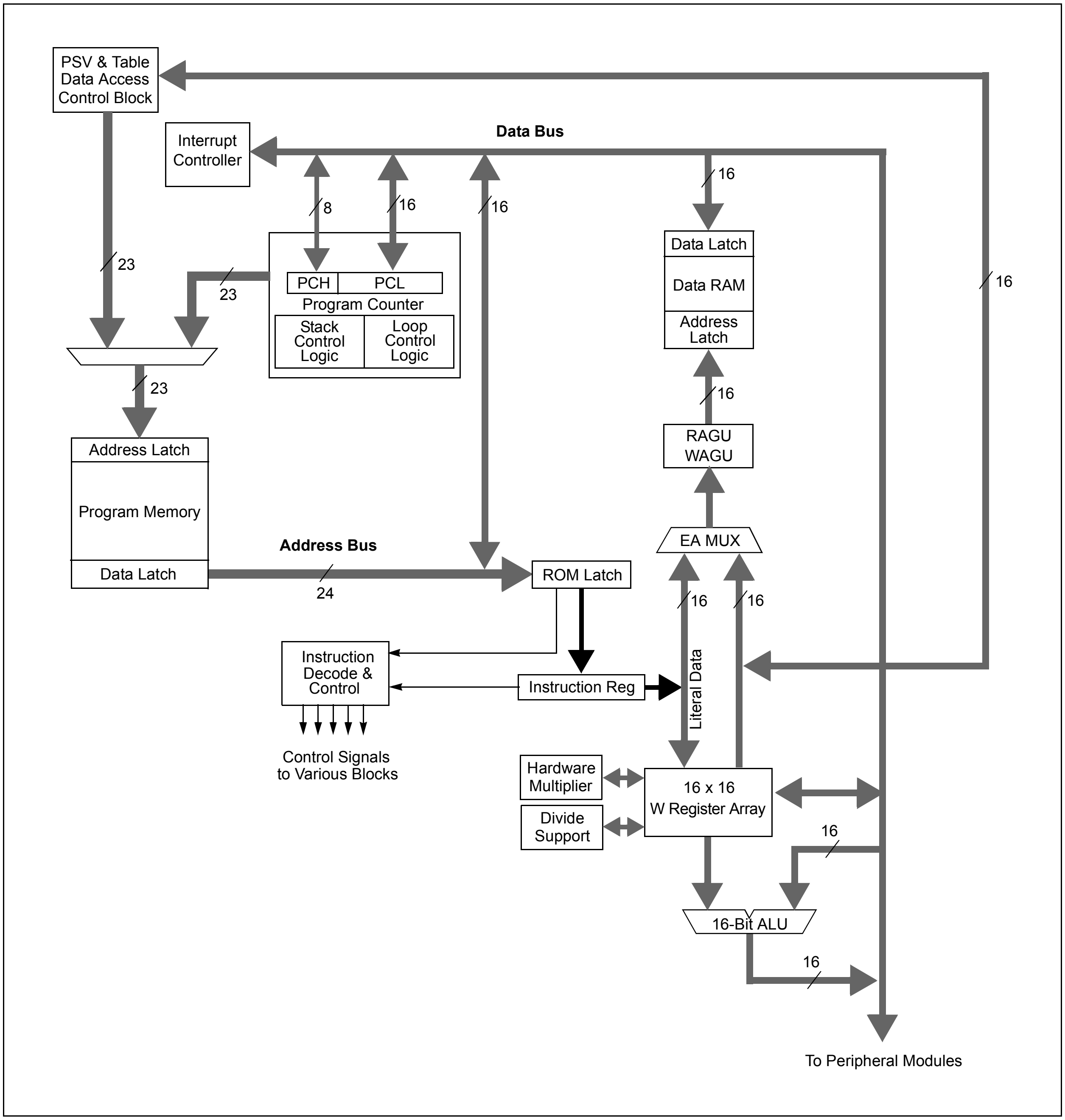
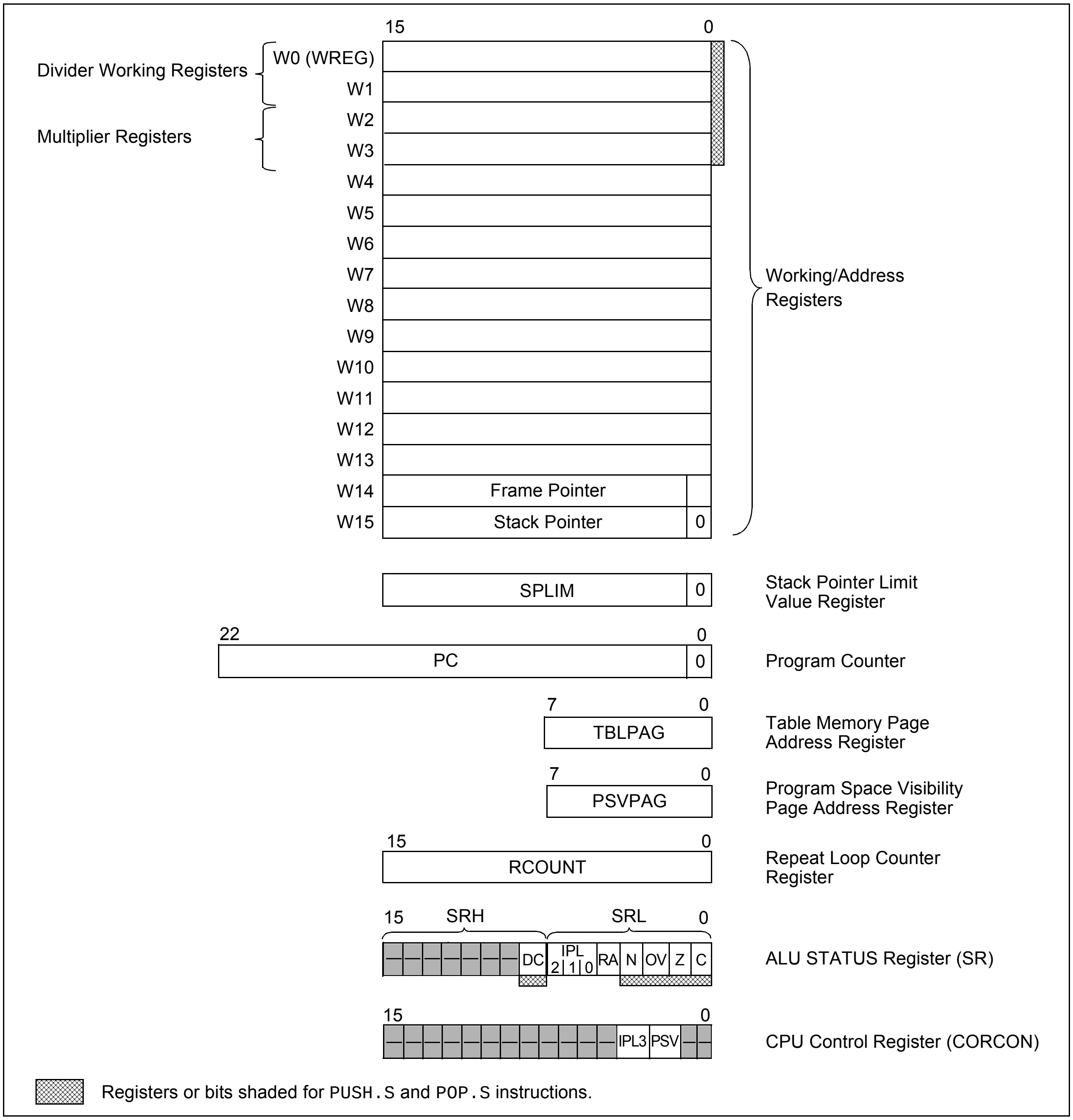


TABLE 3-1: CPU CORE REGISTERS

Register(s) Name	Description
W0 through W15	Working Register Array
PC	23-Bit Program Counter
SR	ALU STATUS Register
SPLIM	Stack Pointer Limit Value Register
TBLPAG	Table Memory Page Address Register
PSVPAG	Program Space Visibility Page Address Register
RCOUNT	Repeat Loop Counter Register
CORCON	CPU Control Register

FIGURE 3-2: PROGRAMMER'S MODEL



3.2 CPU Control Registers

REGISTER 3-1: SR: ALU STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	DC
bit 15							bit 8

R/W-0[1]	R/W-0[1]	R/W-0[1]	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2[2]	IPL1[2]	IPL0[2]	RA	N	OV	Z	C
bit 7							bit 0

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

bit 15-9 **Unimplemented:** Read as ‘0’

bit 8 **DC:** ALU Half Carry/Borrow bit
1 = A carry out from the 4th low-order bit (for byte-sized data) or 8th low-order bit (for word-sized data) of the result occurred
0 = No carry out from the 4th or 8th low-order bit of the result has occurred

bit 7-5 **IPL<2:0>:** CPU Interrupt Priority Level Status bits[1,2]
111 = CPU interrupt priority level is 7 (15); user interrupts disabled
110 = CPU interrupt priority level is 6 (14)
101 = CPU interrupt priority level is 5 (13)
100 = CPU interrupt priority level is 4 (12)
011 = CPU interrupt priority level is 3 (11)
010 = CPU interrupt priority level is 2 (10)
001 = CPU interrupt priority level is 1 (9)
000 = CPU interrupt priority level is 0 (8)

bit 4 **RA:** REPEAT Loop Active bit
1 = REPEAT loop in progress
0 = REPEAT loop not in progress

bit 3 **N:** ALU Negative bit
1 = Result was negative
0 = Result was non-negative (zero or positive)

bit 2 **OV:** ALU Overflow bit
1 = Overflow occurred for signed (2’s complement) arithmetic in this arithmetic operation
0 = No overflow has occurred

bit 1 **Z:** ALU Zero bit
1 = An operation which effects the Z bit has set it at some time in the past
0 = The most recent operation which effects the Z bit has cleared it (i.e., a non-zero result)

bit 0 **C:** ALU Carry/Borrow bit
1 = A carry out from the Most Significant bit of the result occurred
0 = No carry out from the Most Significant bit of the result occurred

Note 1: The IPL Status bits are read-only when NSTDIS (INTCON1<15>) = 1.
2: The IPL Status bits are concatenated with the IPL3 bit (CORCON<3>) to form the CPU Interrupt Priority Level (IPL). The value in parentheses indicates the IPL when IPL3 = 1.

REGISTER 3-2: CORCON: CPU CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/C-0	R/W-0	U-0	U-0
—	—	—	—	IPL3(1)	PSV	—	—
bit 7							bit 0

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

bit 15-4 **Unimplemented:** Read as ‘0’

bit 3 **IPL3:** CPU Interrupt Priority Level Status bit(1)
1 = CPU interrupt priority level is greater than 7
0 = CPU interrupt priority level is 7 or less

bit 2 **PSV:** Program Space Visibility in Data Space Enable bit
1 = Program space visible in data space
0 = Program space not visible in data space

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

Note 1: User interrupts are disabled when IPL3 = 1.

3.3 Arithmetic Logic Unit (ALU)

The PIC24F ALU is 16 bits wide and is capable of addition, subtraction, bit shifts and logic operations. Unless otherwise mentioned, arithmetic operations are 2’s complement in nature. Depending on the operation, the ALU may affect the values of the Carry (C), Zero (Z), Negative (N), Overflow (OV) and Digit Carry (DC) Status bits in the SR register. The C and DC Status bits operate as Borrow and Digit Borrow bits, respectively, for subtraction operations.

The ALU can perform 8-bit or 16-bit operations, depending on the mode of the instruction that is used. Data for the ALU operation can come from the W register array, or data memory, depending on the addressing mode of the instruction. Likewise, output data from the ALU can be written to the W register array or a data memory location.

The PIC24F CPU incorporates hardware support for both multiplication and division. This includes a dedicated hardware multiplier and support hardware for 16-bit divisor division.

3.3.1 MULTIPLIER

The ALU contains a high-speed, 17-bit x 17-bit multiplier. It supports unsigned, signed or mixed sign operation in several multiplication modes:

1. 16-bit x 16-bit signed
2. 16-bit x 16-bit unsigned
3. 16-bit signed x 5-bit (literal) unsigned
4. 16-bit unsigned x 16-bit unsigned
5. 16-bit unsigned x 5-bit (literal) unsigned
6. 16-bit unsigned x 16-bit signed
7. 8-bit unsigned x 8-bit unsigned

3.3.2 DIVIDER

The divide block supports signed and unsigned integer divide operations with the following data sizes:

1. 32-bit signed/16-bit signed divide
2. 32-bit unsigned/16-bit unsigned divide
3. 16-bit signed/16-bit signed divide
4. 16-bit unsigned/16-bit unsigned divide

The quotient for all divide instructions ends up in W0 and the remainder in W1. Sixteen-bit signed and unsigned `DIV` instructions can specify any W register for both the 16-bit divisor (Wn), and any W register (aligned) pair (W(m + 1):Wm) for the 32-bit dividend. The divide algorithm takes one cycle per bit of divisor, so both 32-bit/16-bit and 16-bit/16-bit instructions take the same number of cycles to execute.

3.3.3 MULTI-BIT SHIFT SUPPORT

The PIC24F ALU supports both single bit and single-cycle, multi-bit arithmetic and logic shifts. Multi-bit shifts are implemented using a shifter block, capable of performing up to a 15-bit arithmetic right shift, or up to a 15-bit left shift, in a single cycle. All multi-bit shift instructions only support Register Direct Addressing for both the operand source and result destination.

A full summary of instructions that use the shift operation is provided below in Table 3-2.

TABLE 3-2: INSTRUCTIONS THAT USE THE SINGLE AND MULTI-BIT SHIFT OPERATION

Instruction	Description
ASR	Arithmetic shift right source register by one or more bits.
SL	Shift left source register by one or more bits.
LSR	Logical shift right source register by one or more bits.

4.0 MEMORY ORGANIZATION

As Harvard architecture devices, PIC24F microcontrollers feature separate program and data memory spaces and busses. This architecture also allows the direct access of program memory from the data space during code execution.

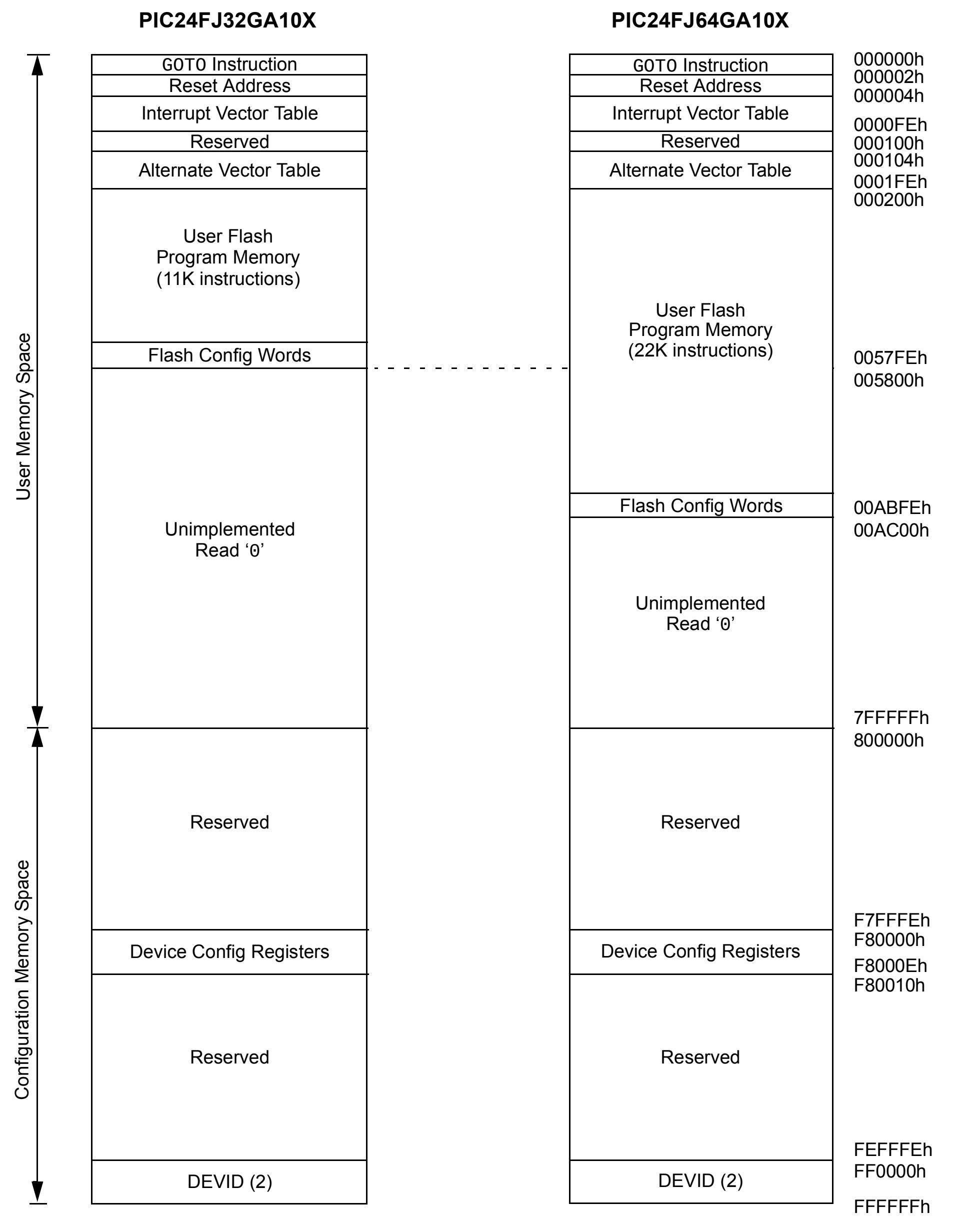
4.1 Program Address Space

The program address memory space of the PIC24FJ64GA104 family devices is 4M instructions. The space is addressable by a 24-bit value derived from either the 23-bit Program Counter (PC) during program execution, or from table operation or data space remapping, as described in **Section 4.3 "Interfacing Program and Data Memory Spaces"**.

User access to the program memory space is restricted to the lower half of the address range (000000h to 7FFFFFh). The exception is the use of TBLRD/TBLWT operations which use TBLPAG<7> to permit access to the Configuration bits and Device ID sections of the configuration memory space.

Memory maps for the PIC24FJ64GA104 family of devices are shown in Figure 4-1.

FIGURE 4-1: PROGRAM SPACE MEMORY MAP FOR PIC24FJ64GA104 FAMILY DEVICES



Note: Memory areas are not shown to scale.

4.1.1 PROGRAM MEMORY ORGANIZATION

The program memory space is organized in word-addressable blocks. Although it is treated as 24 bits wide, it is more appropriate to think of each address of the program memory as a lower and upper word, with the upper byte of the upper word being unimplemented. The lower word always has an even address, while the upper word has an odd address (Figure 4-2).

Program memory addresses are always word-aligned on the lower word and addresses are incremented or decremented by two during code execution. This arrangement also provides compatibility with data memory space addressing and makes it possible to access data in the program memory space.

4.1.2 HARD MEMORY VECTORS

All PIC24F devices reserve the addresses between 00000h and 000200h for hard coded program execution vectors. A hardware Reset vector is provided to redirect code execution from the default value of the PC on device Reset to the actual start of code. A `GOTO` instruction is programmed by the user at 000000h with the actual address for the start of code at 000002h.

PIC24F devices also have two interrupt vector tables, located from 000004h to 0000FFh and 000100h to 0001FFh. These vector tables allow each of the many device interrupt sources to be handled by separate ISRs. A more detailed discussion of the interrupt vector tables is provided in **Section 7.1 "Interrupt Vector Table"**.

4.1.3 FLASH CONFIGURATION WORDS

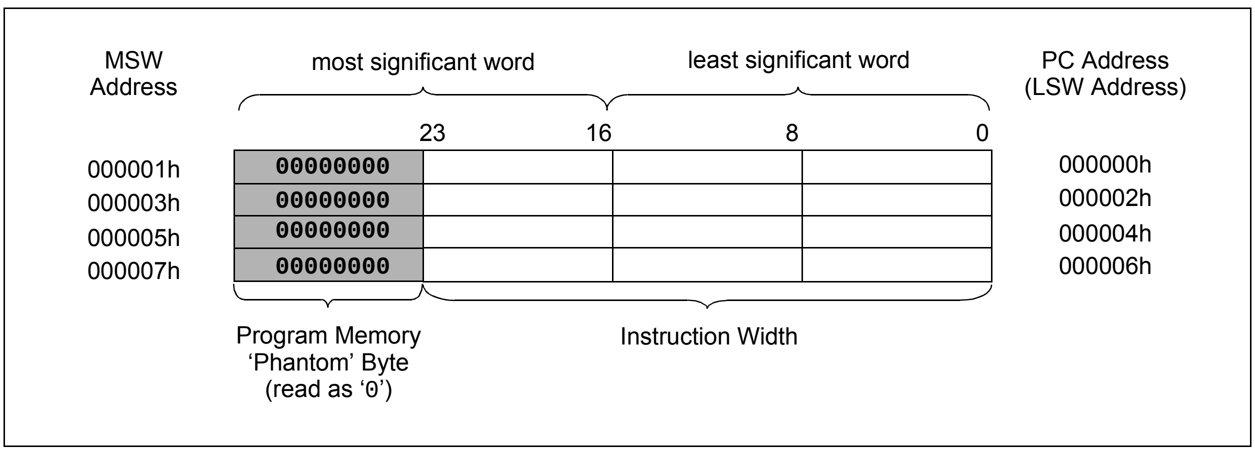
In PIC24FJ64GA104 family devices, the top four words of on-chip program memory are reserved for configuration information. On device Reset, the configuration information is copied into the appropriate Configuration registers. The addresses of the Flash Configuration Word for devices in the PIC24FJ64GA104 family are shown in Table 4-1. Their location in the memory map is shown with the other memory vectors in Figure 4-1.

The Configuration Words in program memory are a compact format. The actual Configuration bits are mapped in several different registers in the configuration memory space. Their order in the Flash Configuration Words do not reflect a corresponding arrangement in the configuration space. Additional details on the device Configuration Words are provided in **Section 25.1 "Configuration Bits"**.

TABLE 4-1: FLASH CONFIGURATION WORDS FOR PIC24FJ64GA104 FAMILY DEVICES

Device	Program Memory (Words)	Configuration Word Addresses
PIC24FJ32GA1	11,008	0057F8h: 0057FEh
PIC24FJ64GA1	22,016	00ABF8h: 00ABFEh

FIGURE 4-2: PROGRAM MEMORY ORGANIZATION



4.2 Data Address Space

The PIC24F core has a separate, 16-bit wide data memory space, addressable as a single linear range. The data space is accessed using two Address Generation Units (AGUs), one each for read and write operations. The data space memory map is shown in Figure 4-3.

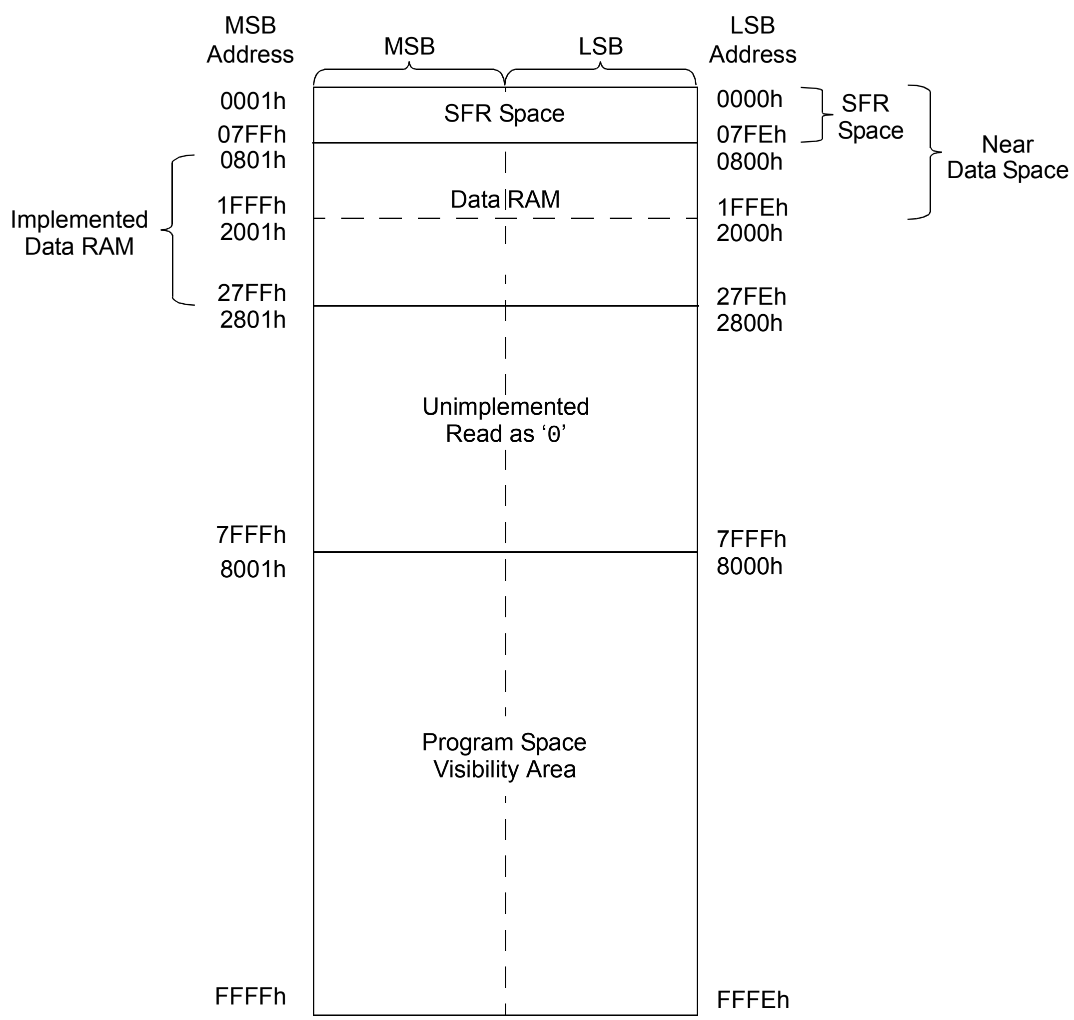
All Effective Addresses (EAs) in the data memory space are 16 bits wide and point to bytes within the data space. This gives a data space address range of 64 Kbytes or 32K words. The lower half of the data memory space (that is, when EA<15> = `0`) is used for implemented memory addresses, while the upper half (EA<15> = `1`) is reserved for the program space visibility area (see **Section 4.3.3 "Reading Data from Program Memory Using Program Space Visibility"**).

PIC24FJ64GA104 family devices implement a total of 16 Kbytes of data memory. Should an EA point to a location outside of this area, an all zero word or byte will be returned.

4.2.1 DATA SPACE WIDTH

The data memory space is organized in byte-addressable, 16-bit wide blocks. Data is aligned in data memory and registers as 16-bit words, but all data space EAs resolve to bytes. The Least Significant Bytes (LSBs) of each word have even addresses, while the Most Significant Bytes (MSBs) have odd addresses.

FIGURE 4-3: DATA SPACE MEMORY MAP FOR PIC24FJ64GA104 FAMILY DEVICES



Note: Data memory areas are not shown to scale.

4.2.2 DATA MEMORY ORGANIZATION AND ALIGNMENT

To maintain backward compatibility with PIC® devices and improve data space memory usage efficiency, the PIC24F instruction set supports both word and byte operations. As a consequence of byte accessibility, all Effective Address calculations are internally scaled to step through word-aligned memory. For example, the core recognizes that Post-Modified Register Indirect Addressing mode [Ws++] will result in a value of Ws + 1 for byte operations and Ws + 2 for word operations.

Data byte reads will read the complete word which contains the byte using the LSb of any EA to determine which byte to select. The selected byte is placed onto the LSB of the data path. That is, data memory and registers are organized as two parallel, byte-wide entities with shared (word) address decode, but separate write lines. Data byte writes only write to the corresponding side of the array or register which matches the byte address.

All word accesses must be aligned to an even address. Misaligned word data fetches are not supported, so care must be taken when mixing byte and word operations or translating from 8-bit MCU code. If a misaligned read or write is attempted, an address error trap will be generated. If the error occurred on a read, the instruction underway is completed; if it occurred on a write, the instruction will be executed but the write will not occur. In either case, a trap is then executed, allowing the system and/or user to examine the machine state prior to execution of the address Fault.

All byte loads into any W register are loaded into the Least Significant Byte. The Most Significant Byte is not modified.

A Sign-Extend (`SE`) instruction is provided to allow users to translate 8-bit signed data to 16-bit signed values. Alternatively, for 16-bit unsigned data, users can clear the MSB of any W register by executing a Zero-Extend (`ZE`) instruction on the appropriate address.

Although most instructions are capable of operating on word or byte data sizes, it should be noted that some instructions operate only on words.

4.2.3 NEAR DATA SPACE

The 8-Kbyte area between 0000h and 1FFFh is referred to as the near data space. Locations in this space are directly addressable via a 13-bit absolute address field within all memory direct instructions. The remainder of the data space is indirectly addressable. Additionally, the whole data space is addressable using `MOV` instructions, which support Memory Direct Addressing with a 16-bit address field.

4.2.4 SFR SPACE

The first 2 Kbytes of the near data space, from 0000h to 07FFh, are primarily occupied with Special Function Registers (SFRs). These are used by the PIC24F core and peripheral modules for controlling the operation of the device.

SFRs are distributed among the modules that they control and are generally grouped together by module. Much of the SFR space contains unused addresses; these are read as '0'. A diagram of the SFR space, showing where SFRs are actually implemented, is shown in Table 4-2. Each implemented area indicates a 32-byte region where at least one address is implemented as an SFR. A complete listing of implemented SFRs, including their addresses, is shown in Tables 4-3 through 4-26.

TABLE 4-2: IMPLEMENTED REGIONS OF SFR DATA SPACE

	SFR Space Address							
	xx00	**xx20**	**xx40**	**xx60**	**xx80**	**xxA0**	**xxC0**	**xxE0**
000h	Core			ICN	Interrupts			—
100h	Timers		Capture		Compare			—
200h	I^{2}C™	UART	SPI	—	—	—	I/O	
300h	A/D	A/D/CTMU	—	—	—	—	—	—
400h	—	—	—	—	—	—	—	—
500h	—	—	—	—	—	—	—	—
600h	PMP	RTCC	CRC/Comp	Comparators	PPS			—
700h	—	—	System/DS	NVM/PMD	—	—	—	—

Legend: — = No implemented SFRs in this block

TABLE 4-3: CPU CORE REGISTERS MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
WREG0	0000	Working Register 0																0000
WREG1	0002	Working Register 1																0000
WREG2	0004	Working Register 2																0000
WREG3	0006	Working Register 3																0000
WREG4	0008	Working Register 4																0000
WREG5	000A	Working Register 5																0000
WREG6	000C	Working Register 6																0000
WREG7	000E	Working Register 7																0000
WREG8	0010	Working Register 8																0000
WREG9	0012	Working Register 9																0000
WREG10	0014	Working Register 10																0000
WREG11	0016	Working Register 11																0000
WREG12	0018	Working Register 12																0000
WREG13	001A	Working Register 13																0000
WREG14	001C	Working Register 14																0000
WREG15	001E	Working Register 15																0800
SPLIM	0020	Stack Pointer Limit Value Register																xxxx
PCL	002E	Program Counter Low Word Register																0000
PCH	0030	—	—	—	—	—	—	—	—	Program Counter Register High Byte								0000
TBLPAG	0032	—	—	—	—	—	—	—	—	Table Memory Page Address Register								0000
PSVPAG	0034	—	—	—	—	—	—	—	—	Program Space Visibility Page Address Register								0000
RCOUNT	0036	Repeat Loop Counter Register																xxxx
SR	0042	—	—	—	—	—	—	—	DC	IPL2	IPL1	IPL0	RA	N	OV	Z	C	0000
CORCON	0044	—	—	—	—	—	—	—	—	—	—	—	—	IPL3	PSV	—	—	0000
DISICNT	0052	—	—	Disable Interrupts Counter Register														xxxx

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-4: ICN REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
CNEN1	0060	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE(1)	CN9IE(1)	CN8IE(1)	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNEN2	0062	—	CN30IE	CN29IE	CN28IE(1)	CN27IE	CN26IE(1)	CN25IE(1)	CN24IE	CN23IE	CN22IE	CN21IE	CN20IE(1)	CN19IE(1)	CN18IE(1)	CN17IE(1)	CN16IE	0000
CNPU1	0068	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE(1)	CN9PUE(1)	CN8PUE(1)	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	006A	—	CN30PUE	CN29PUE	CN28PUE(1)	CN27PUE	CN26PUE(1)	CN25PUE(1)	CN24PUE	CN23PUE	CN22PUE	CN21PUE	CN20PUE(1)	CN19PUE(1)	CN18PUE(1)	CN17PUE(1)	CN16PUE	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: Unimplemented in 28-pin devices; read as '0'.

TABLE 4-5: INTERRUPT CONTROLLER REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
INTCON1	0080	NSTDIS	—	—	—	—	—	—	—	—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000
IFS0	0084	—	—	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPF1IF	T3IF	T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0086	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	—	—	—	—	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF	0000
IFS2	0088	—	—	PMPIF	—	—	—	OC5IF	—	IC5IF	IC4IF	IC3IF	—	—	—	SPI2IF	SPF2IF	0000
IFS3	008A	—	RTCIF	—	—	—	—	—	—	—	—	—	—	—	MI2C2IF	SI2C2IF	—	0000
IFS4	008C	—	—	CTMUIF	—	—	—	—	LVDIF	—	—	—	—	CRCIF	U2ERIF	U1ERIF	—	0000
IEC0	0094	—	—	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPF1IE	T3IE	T2IE	OC2IE	IC2IE	—	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0096	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	—	—	—	—	INT1IE	CNIE	CMIE	MI2C1IE	SI2C1IE	0000
IEC2	0098	—	—	PMPIE	—	—	—	OC5IE	—	IC5IE	IC4IE	IC3IE	—	—	—	SPI2IE	SPF2IE	0000
IEC3	009A	—	RTCIE	—	—	—	—	—	—	—	—	—	—	—	MI2C2IE	SI2C2IE	—	0000
IEC4	009C	—	—	CTMUIE	—	—	—	—	LVDIE	—	—	—	—	CRCIE	U2ERIE	U1ERIE	—	0000
IPC0	00A4	—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0	—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0	4444
IPC1	00A6	—	T2IP2	T2IP1	T2IP0	—	OC2IP2	OC2IP1	OC2IP0	—	IC2IP2	IC2IP1	IC2IP0	—	—	—	—	4440
IPC2	00A8	—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0	—	SPF1IP2	SPF1IP1	SPF1IP0	—	T3IP2	T3IP1	T3IP0	4444
IPC3	00AA	—	—	—	—	—	—	—	—	—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0	0044
IPC4	00AC	—	CNIP2	CNIP1	CNIP0	—	CMIP2	CMIP1	CMIP0	—	MI2C1IP2	MI2C1IP1	MI2C1IP0	—	SI2C1IP2	SI2C1IP1	SI2C1IP0	4444
IPC5	00AE	—	—	—	—	—	—	—	—	—	—	—	—	—	INT1IP2	INT1IP1	INT1IP0	0004
IPC6	00B0	—	T4IP2	T4IP1	T4IP0	—	OC4IP2	OC4IP1	OC4IP0	—	OC3IP2	OC3IP1	OC3IP0	—	—	—	—	4440
IPC7	00B2	—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0	—	INT2IP2	INT2IP1	INT2IP0	—	T5IP2	T5IP1	T5IP0	4444
IPC8	00B4	—	—	—	—	—	—	—	—	—	SPI2IP2	SPI2IP1	SPI2IP0	—	SPF2IP2	SPF2IP1	SPF2IP0	0044
IPC9	00B6	—	IC5IP2	IC5IP1	IC5IP0	—	IC4IP2	IC4IP1	IC4IP0	—	IC3IP2	IC3IP1	IC3IP0	—	—	—	—	4440
IPC10	00B8	—	—	—	—	—	—	—	—	—	OC5IP2	OC5IP1	OC5IP0	—	—	—	—	0040
IPC11	00BA	—	—	—	—	—	—	—	—	—	PMPIP2	PMPIP1	PMPIP0	—	—	—	—	0040
IPC12	00BC	—	—	—	—	—	MI2C2IP2	MI2C2IP1	MI2C2IP0	—	SI2C2IP2	SI2C2IP1	SI2C2IP0	—	—	—	—	0440
IPC15	00C2	—	—	—	—	—	RTCIP2	RTCIP1	RTCIP0	—	—	—	—	—	—	—	—	0400
IPC16	00C4	—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0	—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—	4440
IPC18	00C8	—	—	—	—	—	—	—	—	—	—	—	—	—	LVDIP2	LVDIP1	LVDIP0	0004
IPC19	00CA	—	—	—	—	—	—	—	—	—	CTMUIP2	CTMUIP1	CTMUIP0	—	—	—	—	0040
INTTREG	00E0	CPUIRQ	—	VHOLD	—	ILR3	ILR2	ILR1	ILR0	—	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-6: TIMER REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TMR1	0100	Timer1 Register																0000
PR1	0102	Timer1 Period Register																FFFF
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000
TMR2	0106	Timer2 Register																0000
TMR3HLD	0108	Timer3 Holding Register (for 32-bit timer operations only)																0000
TMR3	010A	Timer3 Register																0000
PR2	010C	Timer2 Period Register																FFFF
PR3	010E	Timer3 Period Register																FFFF
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000
TMR4	0114	Timer4 Register																0000
TMR5HLD	0116	Timer5 Holding Register (for 32-bit operations only)																0000
TMR5	0118	Timer5 Register																0000
PR4	011A	Timer4 Period Register																FFFF
PR5	011C	Timer5 Period Register																FFFF
T4CON	011E	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T5CON	0120	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-7: INPUT CAPTURE REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
IC1CON1	0140	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC1CON2	0142	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC1BUF	0144	Input Capture 1 Buffer Register																0000
IC1TMR	0146	Timer Value 1 Register																xxxx
IC2CON1	0148	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC2CON2	014A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC2BUF	014C	Input Capture 2 Buffer Register																0000
IC2TMR	014E	Timer Value 2 Register																xxxx
IC3CON1	0150	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC3CON2	0152	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC3BUF	0154	Input Capture 3 Buffer Register																0000
IC3TMR	0156	Timer Value 3 Register																xxxx
IC4CON1	0158	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC4CON2	015A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC4BUF	015C	Input Capture 4 Buffer Register																0000
IC4TMR	015E	Timer Value 4 Register																xxxx
IC5CON1	0160	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC5CON2	0162	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC5BUF	0164	Input Capture 5 Buffer Register																0000
IC5TMR	0166	Timer Value 5 Register																xxxx

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-8: OUTPUT COMPARE REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
OC1CON1	0190	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC1CON2	0192	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC1RS	0194	Output Compare 1 Secondary Register																0000
OC1R	0196	Output Compare 1 Register																0000
OC1TMR	0198	Timer Value 1 Register																xxxx
OC2CON1	019A	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC2CON2	019C	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC2RS	019E	Output Compare 2 Secondary Register																0000
OC2R	01A0	Output Compare 2 Register																0000
OC2TMR	01A2	Timer Value 2 Register																xxxx
OC3CON1	01A4	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC3CON2	01A6	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC3RS	01A8	Output Compare 3 Secondary Register																0000
OC3R	01AA	Output Compare 3 Register																0000
OC3TMR	01AC	Timer Value 3 Register																xxxx
OC4CON1	01AE	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC4CON2	01B0	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC4RS	01B2	Output Compare 4 Secondary Register																0000
OC4R	01B4	Output Compare 4 Register																0000
OC4TMR	01B6	Timer Value 4 Register																xxxx
OC5CON1	01B8	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC5CON2	01BA	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC5RS	01BC	Output Compare 5 Secondary Register																0000
OC5R	01BE	Output Compare 5 Register																0000
OC5TMR	01C0	Timer Value 5 Register																xxxx

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-9: I²C™ REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
I2C1RCV	0200	—	—	—	—	—	—	—	—	Receive Register								0000
I2C1TRN	0202	—	—	—	—	—	—	—	—	Transmit Register								00FF
I2C1BRG	0204	—	—	—	—	—	—	—	Baud Rate Generator Register									0000
I2C1CON	0206	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C1STAT	0208	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/$\overline{A}$	P	S	R/$\overline{W}$	RBF	TBF	0000
I2C1ADD	020A	—	—	—	—	—	—	Address Register										0000
I2C1MSK	020C	—	—	—	—	—	—	Address Mask Register										0000
I2C2RCV	0210	—	—	—	—	—	—	—	—	Receive Register								0000
I2C2TRN	0212	—	—	—	—	—	—	—	—	Transmit Register								00FF
I2C2BRG	0214	—	—	—	—	—	—	—	Baud Rate Generator Register									0000
I2C2CON	0216	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C2STAT	0218	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/$\overline{A}$	P	S	R/$\overline{W}$	RBF	TBF	0000
I2C2ADD	021A	—	—	—	—	—	—	Address Register										0000
I2C2MSK	021C	—	—	—	—	—	—	Address Mask Register										0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-10: UART REGISTER MAPS

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
U1MODE	0220	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U1STA	0222	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U1TXREG	0224	—	—	—	—	—	—	—	Transmit Register									xxxx
U1RXREG	0226	—	—	—	—	—	—	—	Receive Register									0000
U1BRG	0228	Baud Rate Generator Prescaler Register																0000
U2MODE	0230	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U2STA	0232	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U2TXREG	0234	—	—	—	—	—	—	—	Transmit Register									xxxx
U2RXREG	0236	—	—	—	—	—	—	—	Receive Register									0000
U2BRG	0238	Baud Rate Generator Prescaler Register																0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-11: SPI REGISTER MAPS

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
SPI1STAT	0240	SPIEN	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0	SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF	0000
SPI1CON1	0242	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000
SPI1CON2	0244	FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—	—	—	—	—	—	—	SPIFE	SPIBEN	0000
SPI1BUF	0248	Transmit and Receive Buffer																0000
SPI2STAT	0260	SPIEN	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0	SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF	0000
SPI2CON1	0262	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000
SPI2CON2	0264	FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—	—	—	—	—	—	—	SPIFE	SPIBEN	0000
SPI2BUF	0268	Transmit and Receive Buffer																0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-12: PORTA REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10(1)	Bit 9(1)	Bit 8(1)	Bit 7(1)	Bit 6	Bit 5	Bit 4	Bit 3	Bit2	Bit 1	Bit 0	All Resets
TRISA	02C0	—	—	—	—	—	TRISA10	TRISA9	TRISA8	TRISA7	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	079F
PORTA	02C2	—	—	—	—	—	RA10	RA9	RA8	RA7	—	—	RA4	RA3	RA2	RA1	RA0	xxxx
LATA	02C4	—	—	—	—	—	LATA10	LATA9	LATA8	LATA7	—	—	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx
ODCA	02C6	—	—	—	—	—	ODA10	ODA9	ODA8	ODA7	—	—	ODA4	ODA3	ODA2	ODA1	ODA0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal. Reset values shown are for 44-pin devices.
Note 1: Bits are unimplemented in 28-pin devices; read as '0'.

TABLE 4-13: PORTB REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TRISB	02C8	TRISB15	TRISB14	TRISB13	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	EFBF
PORTB	02CA	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx
LATB	02CC	LATB15	LATB14	LATB13	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx
ODCB	02CE	ODB15	ODB14	ODB13	ODB12	ODB11	ODB10	ODB9	ODB8	ODB7	ODB6	ODB5	ODB4	ODB3	ODB2	ODB1	ODB0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-14: PORTC REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9(1)	Bit 8(1)	Bit 7(1)	Bit 6(1)	Bit 5(1)	Bit 4(1)	Bit 3(1)	Bit 2(1)	Bit 1(2(1))	Bit 0(1)	All Resets
TRISC	02D0	—	—	—	—	—	—	TRISC9	TRISC8	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	03FF
PORTC	02D2	—	—	—	—	—	—	RC9	RC8	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx
LATC	02D4	—	—	—	—	—	—	LATC9	LATC8	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx
ODCC	02D6	—	—	—	—	—	—	ODC9	ODC8	ODC7	ODC6	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal. Reset values shown are for 44-pin devices.
Note 1: Bits are unimplemented in 28-pin devices; read as '0'.

TABLE 4-15: PAD CONFIGURATION REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
PADCFG1	02FC	—	—	—	—	—	—	—	—	—	—	—	—	—	RTSECSEL1	RTSECSEL0	PMPTTL	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-16: ADC REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
ADC1BUF0	0300	ADC Data Buffer 0																xxxx
ADC1BUF1	0302	ADC Data Buffer 1																xxxx
ADC1BUF2	0304	ADC Data Buffer 2																xxxx
ADC1BUF3	0306	ADC Data Buffer 3																xxxx
ADC1BUF4	0308	ADC Data Buffer 4																xxxx
ADC1BUF5	030A	ADC Data Buffer 5																xxxx
ADC1BUF6	030C	ADC Data Buffer 6																xxxx
ADC1BUF7	030E	ADC Data Buffer 7																xxxx
ADC1BUF8	0310	ADC Data Buffer 8																xxxx
ADC1BUF9	0312	ADC Data Buffer 9																xxxx
ADC1BUFA	0314	ADC Data Buffer 10																xxxx
ADC1BUFB	0316	ADC Data Buffer 11																xxxx
ADC1BUFC	0318	ADC Data Buffer 12																xxxx
ADC1BUFD	031A	ADC Data Buffer 13																xxxx
ADC1BUFE	031C	ADC Data Buffer 14																xxxx
ADC1BUFF	031E	ADC Data Buffer 15																xxxx
AD1CON1	0320	ADON	—	ADSIDL	—	—	—	FORM1	FORM0	SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE	0000
AD1CON2	0322	VCFG2	VCFG1	VCFG0	r	—	CSCNA	—	—	BUFS	—	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS	0000
AD1CON3	0324	ADRC	r	r	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0000
AD1CHS	0328	CH0NB	—	—	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0	CH0NA	—	—	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0	0000
AD1PCFG	032C	PCFG15	PCFG14	PCFG13	PCFG12(1)	PCFG11	PCFG10	PCFG9	PCFG8(1)	PCFG7(1)	PCFG6(1)	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
AD1CSSL	0330	CSSL15	CSSL14	CSSL13	CSSL12(1)	CSSL11	CSSL10	CSSL9	CSSL8(1)	CSSL7(1)	CSSL6(1)	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000

Legend: — = unimplemented, read as '0', r = reserved, maintain as '0'. Reset values are shown in hexadecimal.

Note 1: Bits are not available on 28-pin devices; read as '0'.

TABLE 4-17: CTMU REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
CTMUCON	033C	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT	0000
CTMUICON	033E	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	—	—	—	—	—	—	—	—	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-18: PARALLEL MASTER/SLAVE PORT REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
PMCON	0600	PMPEN	—	PSIDL	ADRMUX1	ADRMUX0	PTBEEN	PTWREN	PTRDEN	CSF1	CSF0	ALP	—	CS1P	BEP	WRSP	RDSP	0000
PMMODE	0602	BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0	WAITB1	WAITB0	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1	WAITE0	0000
PMADDR	0604	—	CS1	—	—	—	ADDR10(1)	ADDR9(1)	ADDR8(1)	ADDR7(1)	ADDR6(1)	ADDR5(1)	ADDR4(1)	ADDR3(1)	ADDR2(1)	ADDR1	ADDR0	0000
PMDOUT1		Parallel Port Data Out Register 1 (Buffers 0 and 1)																0000
PMDOUT2	0606	Parallel Port Data Out Register 2 (Buffers 2 and 3)																0000
PMDIN1	0608	Parallel Port Data In Register 1 (Buffers 0 and 1)																0000
PMDIN2	060A	Parallel Port Data In Register 2 (Buffers 2 and 3)																0000
PMAEN	060C	—	PTEN14	—	—	—	PTEN10(1)	PTEN9(1)	PTEN8(1)	PTEN7(1)	PTEN6(1)	PTEN5(1)	PTEN4(1)	PTEN3(1)	PTEN2(1)	PTEN1	PTEN0	0000
PMSTAT	060E	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: Bits are not available on 28-pin devices; read as '0'.

TABLE 4-19: REAL-TIME CLOCK AND CALENDAR REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
ALRMVAL	0620	Alarm Value Register Window Based on ALRMPTR<1:0>																xxxx
ALCFGRPT	0622	ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0	ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0	0000
RTCVAL	0624	RTCC Value Register Window Based on RTCPTR<1:0>																xxxx
RCFGCAL	0626	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTR0	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	xxxx

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-20: CRC REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
CRCCON1	0640	CRCEN	—	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0	CRCFUL	CRCMPT	CRCISEL	CRCGO	LENDIAN	—	—	—	0000
CRCCON2	0642	—	—	—	DWIDTH4	DWIDTH3	DWIDTH2	DWIDTH1	DWIDTH0	—	—	—	PLEN4	PLEN3	PLEN2	PLEN1	PLEN0	0000
CRCXORL	0644	X15	X14	X13	X12	X11	X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	—	0000
CRCXORH	0646	X31	X30	X29	X28	X27	X26	X25	X24	X23	X22	X21	X20	X19	X19	X17	X16	0000
CRCDATL	0648	CRC Data Input Register Low Word																xxxx
CRCDATH	064A	CRC Data Input Register High Word																xxxx
CRCWDATL	064C	CRC Result Register Low Word																xxxx
CRCWDATH	064E	CRC Result Register High Word																xxxx

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-21: COMPARATORS REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
CMSTAT	0650	CMIDL	—	—	—	—	C3EVT	C2EVT	C1EVT	—	—	—	—	—	C3OUT	C2OUT	C1OUT	0000
CVRCON	0652	—	—	—	—	—	CVREFP	CVREFM1	CVREFM0	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000
CM1CON	0654	CEN	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0	0000
CM2CON	065C	CEN	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0	0000
CM3CON	0664	CEN	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-22: PERIPHERAL PIN SELECT REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
RPINR0	0680	—	—	—	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0	—	—	—	—	—	—	—	—	1F00
RPINR1	0682	—	—	—	—	—	—	—	—	—	—	—	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0	001F
RPINR3	0686	—	—	—	T3CKR4	T3CKR3	T3CKR2	T3CKR1	T3CKR0	—	—	—	T2CKR4	T2CKR3	T2CKR2	T2CKR1	T2CKR0	1F1F
RPINR4	0688	—	—	—	T5CKR4	T5CKR3	T5CKR2	T5CKR1	T5CKR0	—	—	—	T4CKR4	T4CKR3	T4CKR2	T4CKR1	T4CKR0	1F1F
RPINR7	068E	—	—	—	IC2R4	IC2R3	IC2R2	IC2R1	IC2R0	—	—	—	IC1R4	IC1R3	IC1R2	IC1R1	IC1R0	1F1F
RPINR8	0690	—	—	—	IC4R4	IC4R3	IC4R2	IC4R1	IC4R0	—	—	—	IC3R4	IC3R3	IC3R2	IC3R1	IC3R0	1F1F
RPINR9	0692	—	—	—	—	—	—	—	—	—	—	—	IC5R4	IC5R3	IC5R2	IC5R1	IC5R0	001F
RPINR11	0696	—	—	—	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0	—	—	—	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0	1F1F
RPINR18	06A4	—	—	—	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0	—	—	—	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0	1F1F
RPINR19	06A6	—	—	—	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0	—	—	—	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0	1F1F
RPINR20	06A8	—	—	—	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0	—	—	—	SDI1R4	SDI1R3	SDI1R2	SDI1R1	SDI1R0	1F1F
RPINR21	06AA	—	—	—	—	—	—	—	—	—	—	—	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0	001F
RPINR22	06AC	—	—	—	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0	—	—	—	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0	1F1F
RPINR23	06AE	—	—	—	—	—	—	—	—	—	—	—	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0	001F
RPOR0	06C0	—	—	—	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0	—	—	—	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0	0000
RPOR1	06C2	—	—	—	RP3R4	RP3R3	RP3R2	RP3R1	RP3R0	—	—	—	RP2R4	RP2R3	RP2R2	RP2R1	RP2R0	0000
RPOR2	06C4	—	—	—	RP5R4	RP5R3	RP5R2	RP5R1	RP5R0	—	—	—	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0	0000
RPOR3	06C6	—	—	—	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0	—	—	—	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0	0000
RPOR4	06C8	—	—	—	RP9R4	RP9R3	RP9R2	RP9R1	RP9R0	—	—	—	RP8R4	RP8R3	RP8R2	RP8R1	RP8R0	0000
RPOR5	06CA	—	—	—	RP11R4	RP11R3	RP11R2	RP11R1	RP11R0	—	—	—	RP10R4	RP10R3	RP10R2	RP10R1	RP10R0	0000
RPOR6	06CC	—	—	—	RP13R4	RP13R3	RP13R2	RP13R1	RP13R0	—	—	—	RP12R4	RP12R3	RP12R2	RP12R1	RP12R0	0000
RPOR7	06CE	—	—	—	RP15R4	RP15R3	RP15R2	RP15R1	RP15R0	—	—	—	RP14R4	RP14R3	RP14R2	RP14R1	RP14R0	0000
RPOR8[1]	06D0	—	—	—	RP17R4	RP17R3	RP17R2	RP17R1	RP17R0	—	—	—	RP16R4	RP16R3	RP16R2	RP16R1	RP16R0	0000
RPOR9[1]	06D2	—	—	—	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0	—	—	—	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0	0000
RPOR10[1]	06D4	—	—	—	RP21R4	RP21R3	RP21R2	RP21R1	RP21R0	—	—	—	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0	0000
RPOR11[1]	06D6	—	—	—	RP23R4	RP23R3	RP23R2	RP23R1	RP23R0	—	—	—	RP22R4	RP22R3	RP22R2	RP22R1	RP22R0	0000
RPOR12[1]	06D8	—	—	—	RP25R4	RP25R3	RP25R2	RP25R1	RP25R0	—	—	—	RP24R4	RP24R3	RP24R2	RP24R1	RP24R0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: Registers are unimplemented in 28-pin devices; read as '0'.

查询PIC24FJ64GA104供应商

TABLE 4-23: SYSTEM REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
RCON	0740	TRAPR	IOPUWR	—	—	—	DPSLP	CM	PMSLP	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	**Note 1**
OSCCON	0742	—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0	CLKLOCK	IOLOCK	LOCK	—	CF	POSCEN	SOSCEN	OSWEN	**Note 2**
CLKDIV	0744	ROI	DOZE2	DOZE1	DOZE0	DOZEN	RCDIV2	RCDIV1	RCDIV0	—	—	—	—	—	—	—	—	0100
OSCTUN	0748	—	—	—	—	—	—	—	—	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	0000
REFOCON	074E	ROEN	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0	—	—	—	—	—	—	—	—	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: The Reset value of the RCON register is dependent on the type of Reset event. See **Section 6.0 "Resets"** for more information.
2: The Reset value of the OSCCON register is dependent on both the type of Reset event and the device configuration. See **Section 8.0 "Oscillator Configuration"** for more information.

TABLE 4-24: DEEP SLEEP REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets(1)
DSCON	758	DSEN	—	—	—	—	—	—	—	—	—	—	—	—	—	DSBOR	RELEASE	0000
DSWAKE	075A	—	—	—	—	—	—	—	DSINT0	DSFLT	—	—	DSWDT	DSRTC	DSMCLR	—	DSPOR	0001
DSGPR0	075C	Deep Sleep General Purpose Register 0																0000
DSGPR1	075E	Deep Sleep General Purpose Register 1																0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: The Deep Sleep registers are only reset on a VDD POR event.

TABLE 4-25: NVM REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
NVMCON	0760	WR	WREN	WRERR	—	—	—	—	—	—	ERASE	—	—	NVMOP3	NVMOP2	NVMOP1	NVMOP0	0000(1)
NVMKEY	0766	—	—	—	—	—	—	—	—	NVMKEY Register<7:0>								0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: Reset value shown is for POR only. Value on other Reset states is dependent on the state of memory write or erase operations at the time of Reset.

TABLE 4-26: PMD REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
PMD1	0770	T5MD	T4MD	T3MD	T2MD	T1MD	—	—	—	I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	—	—	ADC1MD	0000
PMD2	0772	—	—	—	IC5MD	IC4MD	IC3MD	IC2MD	IC1MD	—	—	—	OC5MD	OC4MD	OC3MD	OC2MD	OC1MD	0000
PMD3	0774	—	—	—	—	—	CMPMD	RTCCMD	PMPMD	CRCMD	—	—	—	—	—	I2C2MD	—	0000
PMD4	0776	—	—	—	—	—	—	—	—	—	—	—	—	REFOMD	CTMUMD	LVDMD	—	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

4.2.5 SOFTWARE STACK

In addition to its use as a working register, the W15 register in PIC24F devices is also used as a Software Stack Pointer. The pointer always points to the first available free word and grows from lower to higher addresses. It predecrements for stack pops and post-increments for stack pushes, as shown in Figure 4-4. Note that for a PC push during any `CALL` instruction, the MSB of the PC is zero-extended before the push, ensuring that the MSB is always clear.

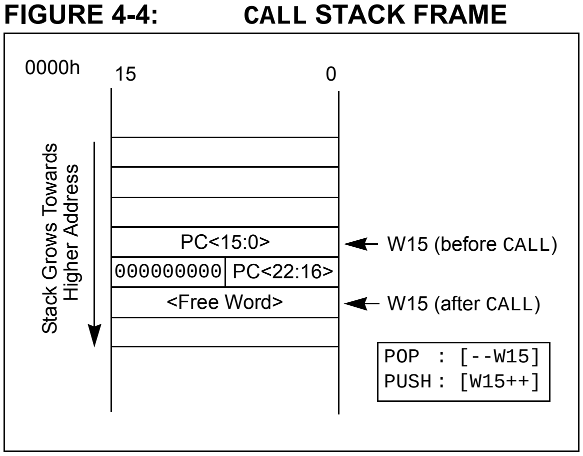
Note:	A PC push during exception processing will concatenate the SRL register to the MSB of the PC prior to the push.

The Stack Pointer Limit Value (SPLIM) register, associated with the Stack Pointer, sets an upper address boundary for the stack. SPLIM is uninitialized at Reset. As is the case for the Stack Pointer, SPLIM<0> is forced to ‘0’ because all stack operations must be word-aligned. Whenever an EA is generated using W15 as a source or destination pointer, the resulting address is compared with the value in SPLIM. If the contents of the Stack Pointer (W15) and the SPLIM register are equal, and a push operation is performed, a stack error trap will not occur. The stack error trap will occur on a subsequent push operation. Thus, for example, if it is desirable to cause a stack error trap when the stack grows beyond address 2000h in RAM, initialize the SPLIM with the value, 1FFEh.

Similarly, a Stack Pointer underflow (stack error) trap is generated when the Stack Pointer address is found to be less than 0800h. This prevents the stack from interfering with the Special Function Register (SFR) space.

A write to the SPLIM register should not be immediately followed by an indirect read operation using W15.

FIGURE 4-4: CALL STACK FRAME



4.3 Interfacing Program and Data Memory Spaces

The PIC24F architecture uses a 24-bit wide program space and a 16-bit wide data space. The architecture is also a modified Harvard scheme, meaning that data can also be present in the program space. To use this data successfully, it must be accessed in a way that preserves the alignment of information in both spaces.

Aside from normal execution, the PIC24F architecture provides two methods by which program space can be accessed during operation:

- Using table instructions to access individual bytes or words anywhere in the program space
- Remapping a portion of the program space into the data space (program space visibility)

Table instructions allow an application to read or write to small areas of the program memory. This makes the method ideal for accessing data tables that need to be updated from time to time. It also allows access to all bytes of the program word. The remapping method allows an application to access a large block of data on a read-only basis, which is ideal for look-ups from a large table of static data; it can only access the least significant word of the program word.

4.3.1 ADDRESSING PROGRAM SPACE

Since the address ranges for the data and program spaces are 16 and 24 bits, respectively, a method is needed to create a 23-bit or 24-bit program address from 16-bit data registers. The solution depends on the interface method to be used.

For table operations, the 8-bit Table Memory Page Address (TBLPAG) register is used to define a 32K word region within the program space. This is concatenated with a 16-bit EA to arrive at a full 24-bit program space address. In this format, the Most Significant bit of TBLPAG is used to determine if the operation occurs in the user memory (TBLPAG<7> = `0`) or the configuration memory (TBLPAG<7> = `1`).

For remapping operations, the 8-bit Program Space Visibility Page Address (PSVPAG) register is used to define a 16K word page in the program space. When the Most Significant bit of the EA is ‘1’, PSVPAG is concatenated with the lower 15 bits of the EA to form a 23-bit program space address. Unlike table operations, this limits remapping operations strictly to the user memory area.

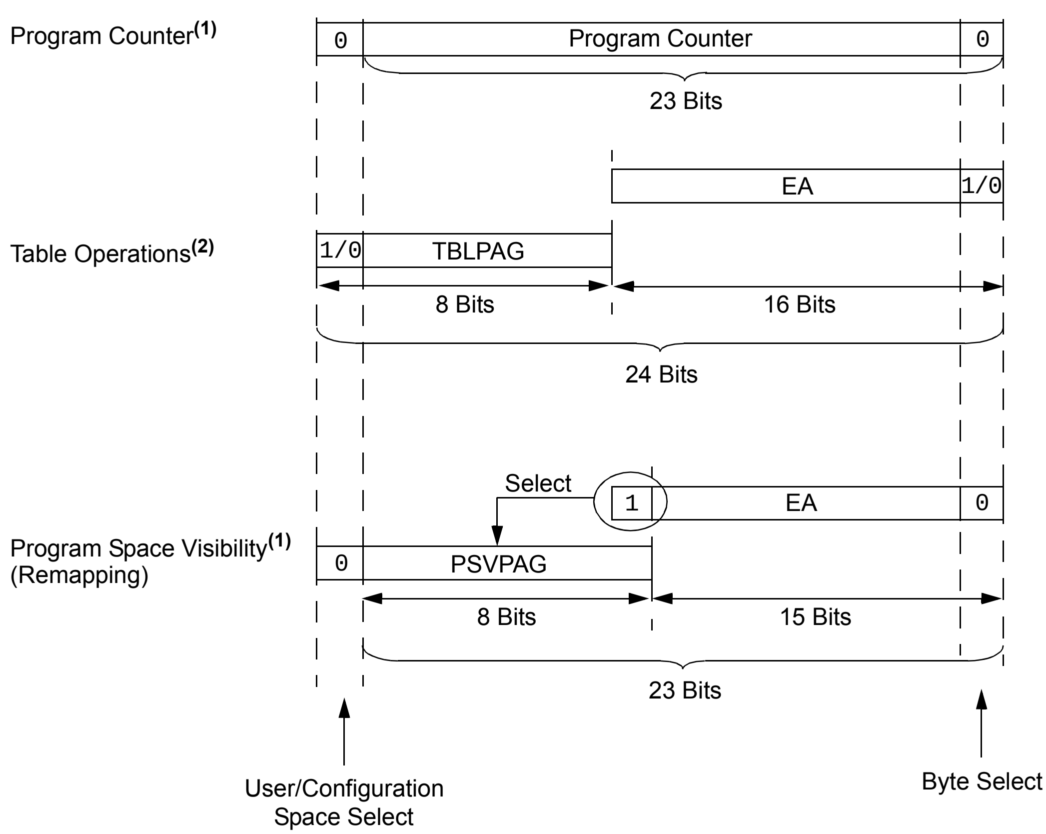
Table 4-27 and Figure 4-5 show how the program EA is created for table operations and remapping accesses from the data EA. Here, P<23:0> refers to a program space word, whereas D<15:0> refers to a data space word.

TABLE 4-27: PROGRAM SPACE ADDRESS CONSTRUCTION

<table>
<tr><th rowspan="2">Access Type</th><th rowspan="2">Access Space</th><th colspan="5">Program Space Address</th></tr>
<tr><th><23></th><th><22:16></th><th><15></th><th><14:1></th><th><0></th></tr>
<tr><td rowspan="2">Instruction Access (Code Execution)</td><td rowspan="2">User</td><td>0</td><td colspan="3">PC<22:1></td><td>0</td></tr>
<tr><td colspan="5">0xx xxxx xxxx xxxx xxxx xxx0</td></tr>
<tr><td rowspan="4">TBLRD/TBLWT (Byte/Word Read/Write)</td><td rowspan="2">User</td><td colspan="2">TBLPAG<7:0></td><td colspan="3">Data EA<15:0></td></tr>
<tr><td colspan="2">0xxx xxxx</td><td colspan="3">xxxx xxxx xxxx xxxx</td></tr>
<tr><td rowspan="2">Configuration</td><td colspan="2">TBLPAG<7:0></td><td colspan="3">Data EA<15:0></td></tr>
<tr><td colspan="2">1xxx xxxx</td><td colspan="3">xxxx xxxx xxxx xxxx</td></tr>
<tr><td rowspan="2">Program Space Visibility (Block Remap/Read)</td><td rowspan="2">User</td><td>0</td><td colspan="2">PSVPAG<7:0></td><td colspan="2">Data EA<14:0>(1)</td></tr>
<tr><td>0</td><td colspan="2">xxxx xxxx</td><td colspan="2">xxx xxxx xxxx xxxx</td></tr>
</table>

Note 1: Data EA<15> is always '1' in this case, but is not used in calculating the program space address. Bit 15 of the address is PSVPAG<0>.

FIGURE 4-5: DATA ACCESS FROM PROGRAM SPACE ADDRESS GENERATION



Note 1: The LSb of program space addresses is always fixed as '0' in order to maintain word alignment of data in the program and data spaces.

2: Table operations are not required to be word-aligned. Table read operations are permitted in the configuration memory space.

4.3.2 DATA ACCESS FROM PROGRAM MEMORY USING TABLE INSTRUCTIONS

The `TBLRDL` and `TBLWTL` instructions offer a direct method of reading or writing the lower word of any address within the program space without going through data space. The `TBLRDH` and `TBLWTH` instructions are the only method to read or write the upper 8 bits of a program space word as data.

The PC is incremented by two for each successive 24-bit program word. This allows program memory addresses to directly map to data space addresses. Program memory can thus be regarded as two, 16-bit word-wide address spaces, residing side by side, each with the same address range. `TBLRDL` and `TBLWTL` access the space which contains the least significant data word, and `TBLRDH` and `TBLWTH` access the space which contains the upper data byte.

Two table instructions are provided to move byte or word-sized (16-bit) data to and from program space. Both function as either byte or word operations.

1. `TBLRDL` (Table Read Low): In Word mode, it maps the lower word of the program space location (P<15:0>) to a data address (D<15:0>).

   In Byte mode, either the upper or lower byte of the lower program word is mapped to the lower byte of a data address. The upper byte is selected when the byte select is '`1`'; the lower byte is selected when it is '`0`'.

2. `TBLRDH` (Table Read High): In Word mode, it maps the entire upper word of a program address (P<23:16>) to a data address. Note that D<15:8>, the 'phantom' byte, will always be '`0`'.

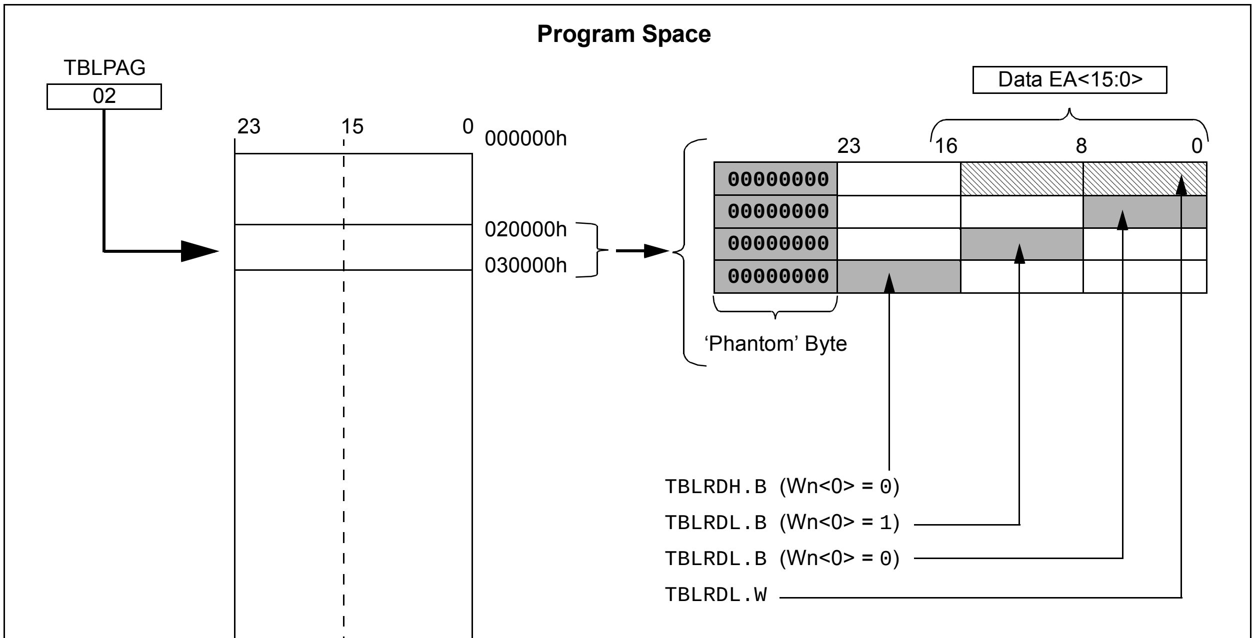
   In Byte mode, it maps the upper or lower byte of the program word to D<7:0> of the data address, as above. Note that the data will always be '`0`' when the upper 'phantom' byte is selected (byte select = `1`).

In a similar fashion, two table instructions, `TBLWTH` and `TBLWTL`, are used to write individual bytes or words to a program space address. The details of their operation are explained in **Section 5.0 "Flash Program Memory"**.

For all table operations, the area of program memory space to be accessed is determined by the Table Memory Page Address register (TBLPAG). TBLPAG covers the entire program memory space of the device, including user and configuration spaces. When TBLPAG<7> = `0`, the table page is located in the user memory space. When TBLPAG<7> = `1`, the page is located in configuration space.

Note:	Only table read operations will execute in the configuration memory space, and only then, in implemented areas, such as the Device ID. Table write operations are not allowed.

FIGURE 4-6: ACCESSING PROGRAM MEMORY WITH TABLE INSTRUCTIONS



The address for the table operation is determined by the data EA within the page defined by the TBLPAG register.
Only read operations are shown; write operations are also valid in the user memory area.

4.3.3 READING DATA FROM PROGRAM MEMORY USING PROGRAM SPACE VISIBILITY

The upper 32 Kbytes of data space may optionally be mapped into any 16K word page of the program space. This provides transparent access of stored constant data from the data space without the need to use special instructions (i.e., `TBLRDL/H`).

Program space access through the data space occurs if the Most Significant bit (MSb) of the data space EA is '`1`' and program space visibility is enabled by setting the PSV bit in the CPU Control register (CORCON<2>). The location of the program memory space to be mapped into the data space is determined by the Program Space Visibility Page Address register (PSVPAG). This 8-bit register defines any one of 256 possible pages of 16K words in program space. In effect, PSVPAG functions as the upper 8 bits of the program memory address, with the 15 bits of the EA functioning as the lower bits. Note that by incrementing the PC by 2 for each program memory word, the lower 15 bits of data space addresses directly map to the lower 15 bits in the corresponding program space addresses.

Data reads to this area add an additional cycle to the instruction being executed, since two program memory fetches are required.

Although each data space address, 8000h and higher, maps directly into a corresponding program memory address (see Figure 4-7), only the lower 16 bits of the 24-bit program word are used to contain the data. The upper 8 bits of any program space locations used as data should be programmed with '`1111 1111`' or '`0000 0000`' to force a `NOP`. This prevents possible issues should the area of code ever be accidentally executed.

Note:	PSV access is temporarily disabled during table reads/writes.

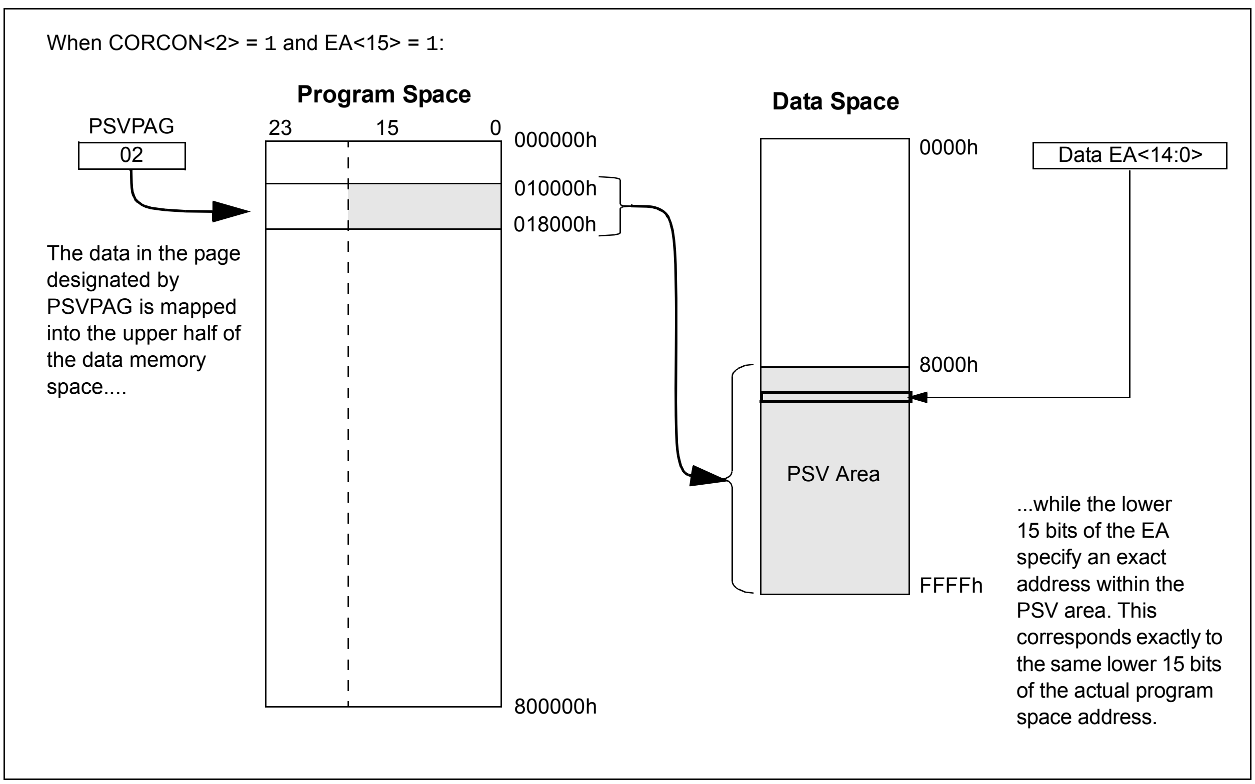
For operations that use PSV and are executed outside a `REPEAT` loop, the `MOV` and `MOV.D` instructions will require one instruction cycle in addition to the specified execution time. All other instructions will require two instruction cycles in addition to the specified execution time.

For operations that use PSV which are executed inside a `REPEAT` loop, there will be some instances that require two instruction cycles in addition to the specified execution time of the instruction:

- Execution in the first iteration
- Execution in the last iteration
- Execution prior to exiting the loop due to an interrupt
- Execution upon re-entering the loop after an interrupt is serviced

Any other iteration of the `REPEAT` loop will allow the instruction accessing data, using PSV, to execute in a single cycle.

FIGURE 4-7: PROGRAM SPACE VISIBILITY OPERATION



5.0 FLASH PROGRAM MEMORY

> **Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 4. "Program Memory"** (DS39715).

The PIC24FJ64GA104 family of devices contains internal Flash program memory for storing and executing application code. The memory is readable, writable and erasable when operating with VDD over 2.35V. (If the regulator is disabled, VDDCORE must be over 2.25V.)

It can be programmed in four ways:

- In-Circuit Serial Programming™ (ICSP™)
- Run-Time Self-Programming (RTSP)
- Enhanced In-Circuit Serial Programming (Enhanced ICSP)

ICSP allows a PIC24FJ64GA104 family device to be serially programmed while in the end application circuit. This is simply done with two lines for the programming clock and programming data (which are named PGECx and PGEDx, respectively), and three other lines for power (VDD), ground (VSS) and Master Clear (MCLR). 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.

RTSP is accomplished using `TBLRD` (table read) and `TBLWT` (table write) instructions. With RTSP, the user may write program memory data in blocks of 64 instructions (192 bytes) at a time and erase program memory in blocks of 512 instructions (1536 bytes) at a time.

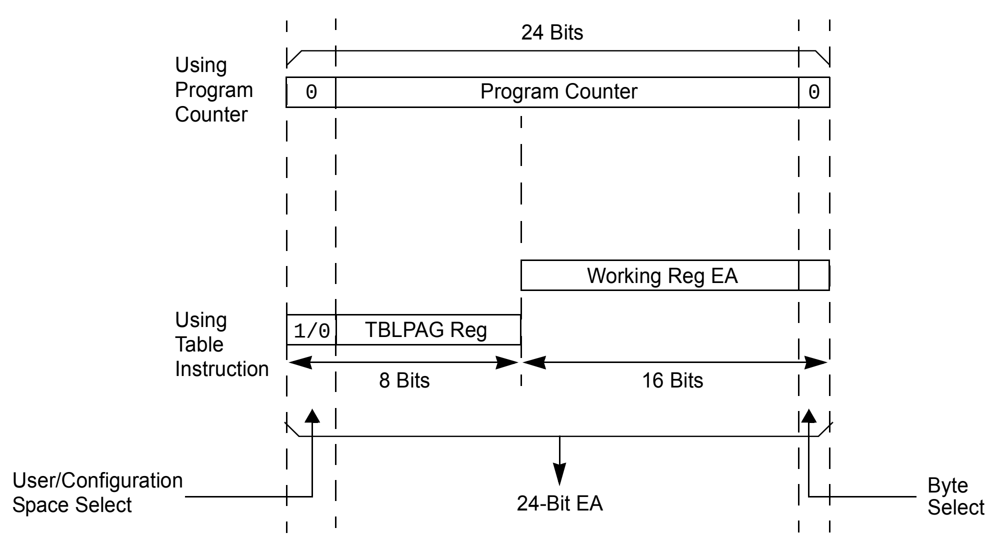
5.1 Table Instructions and Flash Programming

Regardless of the method used, all programming of Flash memory is done with the table read and table write instructions. These allow direct read and write access to the program memory space from the data memory while the device is in normal operating mode. The 24-bit target address in the program memory is formed using the TBLPAG<7:0> bits and the Effective Address (EA) from a W register specified in the table instruction, as shown in Figure 5-1.

The `TBLRDL` and the `TBLWTL` instructions are used to read or write to bits<15:0> of program memory. `TBLRDL` and `TBLWTL` can access program memory in both Word and Byte modes.

The `TBLRDH` and `TBLWTH` instructions are used to read or write to bits<23:16> of program memory. `TBLRDH` and `TBLWTH` can also access program memory in Word or Byte mode.

FIGURE 5-1: ADDRESSING FOR TABLE REGISTERS



5.2 RTSP Operation

The PIC24F Flash program memory array is organized into rows of 64 instructions or 192 bytes. RTSP allows the user to erase blocks of eight rows (512 instructions) at a time and to program one row at a time. It is also possible to program single words.

The 8-row erase blocks and single row write blocks are edge-aligned, from the beginning of program memory, on boundaries of 1536 bytes and 192 bytes, respectively.

When data is written to program memory using `TBLWT` instructions, the data is not written directly to memory. Instead, data written using table writes is stored in holding latches until the programming sequence is executed.

Any number of `TBLWT` instructions can be executed and a write will be successfully performed. However, 64 `TBLWT` instructions are required to write the full row of memory.

To ensure that no data is corrupted during a write, any unused addresses should be programmed with FFFFFFh. This is because the holding latches reset to an unknown state, so if the addresses are left in the Reset state, they may overwrite the locations on rows which were not rewritten.

The basic sequence for RTSP programming is to set up a Table Pointer, then do a series of `TBLWT` instructions to load the buffers. Programming is performed by setting the control bits in the NVMCON register.

Data can be loaded in any order and the holding registers can be written to multiple times before performing a write operation. Subsequent writes, however, will wipe out any previous writes.

Note:	Writing to a location multiple times without erasing is *not* recommended.

All of the table write operations are single-word writes (2 instruction cycles), because only the buffers are written. A programming cycle is required for programming each row.

5.3 JTAG Operation

The PIC24F family supports JTAG boundary scan. Boundary scan can improve the manufacturing process by verifying pin to PCB connectivity.

5.4 Enhanced In-Circuit Serial Programming

Enhanced In-Circuit Serial Programming uses an on-board bootloader, known as the program executive, to manage the programming process. Using an SPI data frame format, the program executive can erase, program and verify program memory. For more information on Enhanced ICSP, see the device programming specification.

5.5 Control Registers

There are two SFRs used to read and write the program Flash memory: NVMCON and NVMKEY.

The NVMCON register (Register 5-1) controls which blocks are to be erased, which memory type is to be programmed and when the programming cycle starts.

NVMKEY is a write-only register that is used for write protection. To start a programming or erase sequence, the user must consecutively write 55h and AAh to the NVMKEY register. Refer to **Section 5.6 "Programming Operations"** for further details.

5.6 Programming Operations

A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. During a programming or erase operation, the processor stalls (waits) until the operation is finished. Setting the WR bit (NVMCON<15>) starts the operation and the WR bit is automatically cleared when the operation is finished.

REGISTER 5-1: NVMCON: FLASH MEMORY CONTROL REGISTER

R/SO-0, HC(1)	R/W-0(1)	R/W-0, HS(1)	U-0	U-0	U-0	U-0	U-0
WR	WREN	WRERR	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0(1)	U-0	U-0	R/W-0(1)	R/W-0(1)	R/W-0(1)	R/W-0(1)
—	ERASE	—	—	NVMOP3(2)	NVMOP2(2)	NVMOP1(2)	NVMOP0(2)
bit 7							bit 0

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

bit 15 **WR:** Write Control bit(1)
- `1` = Initiates a Flash memory program or erase operation. The operation is self-timed and the bit is cleared by hardware once the operation is complete.
- `0` = Program or erase operation is complete and inactive

bit 14 **WREN:** Write Enable bit(1)
- `1` = Enable Flash program/erase operations
- `0` = Inhibit Flash program/erase operations

bit 13 **WRERR:** Write Sequence Error Flag bit(1)
- `1` = An improper program or erase sequence attempt, or termination has occurred (bit is set automatically on any set attempt of the WR bit)
- `0` = The program or erase operation completed normally

bit 12-7 **Unimplemented:** Read as ‘`0`’

bit 6 **ERASE:** Erase/Program Enable bit(1)
- `1` = Perform the erase operation specified by NVMOP<3:0> on the next WR command
- `0` = Perform the program operation specified by NVMOP<3:0> on the next WR command

bit 5-4 **Unimplemented:** Read as ‘`0`’

bit 3-0 **NVMOP<3:0>:** NVM Operation Select bits(1,2)
- `1111` = Memory bulk erase operation (ERASE = `1`) or no operation (ERASE = `0`)(3)
- `0011` = Memory word program operation (ERASE = `0`) or no operation (ERASE = `1`)
- `0010` = Memory page erase operation (ERASE = `1`) or no operation (ERASE = `0`)
- `0001` = Memory row program operation (ERASE = `0`) or no operation (ERASE = `1`)

Note 1: These bits can only be reset on POR.
2: All other combinations of NVMOP<3:0> are unimplemented.
3: Available in ICSP™ mode only. Refer to device programming specification.

5.6.1 PROGRAMMING ALGORITHM FOR FLASH PROGRAM MEMORY

The user can program one row of Flash program memory at a time. To do this, it is necessary to erase the 8-row erase block containing the desired row. The general process is as follows:

1. Read eight rows of program memory (512 instructions) and store in data RAM.
2. Update the program data in RAM with the desired new data.
3. Erase the block (see Example 5-1):
   a) Set the NVMOP bits (NVMCON<3:0>) to ‘0010’ to configure for block erase. Set the ERASE (NVMCON<6>) and WREN (NVMCON<14>) bits.
   b) Write the starting address of the block to be erased into the TBLPAG and W registers.
   c) Write 55h to NVMKEY.
   d) Write AAh to NVMKEY.
   e) Set the WR bit (NVMCON<15>). The erase cycle begins and the CPU stalls for the duration of the erase cycle. When the erase is done, the WR bit is cleared automatically.
4. Write the first 64 instructions from data RAM into the program memory buffers (see Example 5-1).
5. Write the program block to Flash memory:
   a) Set the NVMOP bits to ‘0001’ to configure for row programming. Clear the ERASE bit and set the WREN bit.
   b) Write 55h to NVMKEY.
   c) Write AAh to NVMKEY.
   d) Set the WR bit. The programming cycle begins and the CPU stalls for the duration of the write cycle. When the write to Flash memory is done, the WR bit is cleared automatically.
6. Repeat steps 4 and 5, using the next available 64 instructions from the block in data RAM by incrementing the value in TBLPAG, until all 512 instructions are written back to Flash memory.

For protection against accidental operations, the write initiate sequence for NVMKEY must be used to allow any erase or program operation to proceed. After the programming command has been executed, the user must wait for the programming time until programming is complete. The two instructions following the start of the programming sequence should be NOPs, as shown in Example 5-5.

EXAMPLE 5-1: ERASING A PROGRAM MEMORY BLOCK (ASSEMBLY LANGUAGE CODE)

```
; Set up NVMCON for block erase operation
       MOV     #0x4042, W0                ;
       MOV     W0, NVMCON                 ; Initialize NVMCON
; Init pointer to row to be ERASED
       MOV     #tblpage(PROG_ADDR), W0    ;
       MOV     W0, TBLPAG                 ; Initialize PM Page Boundary SFR
       MOV     #tbloffset(PROG_ADDR), W0  ; Initialize in-page EA[15:0] pointer
       TBLWTL  W0, [W0]                   ; Set base address of erase block
       DISI    #5                         ; Block all interrupts with priority <7
                                          ; for next 5 instructions
       MOV     #0x55, W0
       MOV     W0, NVMKEY                 ; Write the 55 key
       MOV     #0xAA, W1                  ;
       MOV     W1, NVMKEY                 ; Write the AA key
       BSET    NVMCON, #WR                ; Start the erase sequence
       NOP                                ; Insert two NOPs after the erase
       NOP                                ; command is asserted
```

EXAMPLE 5-2: ERASING A PROGRAM MEMORY BLOCK (C LANGUAGE CODE)

```
// C example using MPLAB C30
    unsigned long progAddr = 0xXXXXXX;        // Address of row to write
    unsigned int offset;

//Set up pointer to the first memory location to be written
    TBLPAG = progAddr>>16;                    // Initialize PM Page Boundary SFR
    offset = progAddr & 0xFFFF;               // Initialize lower word of address

    __builtin_tblwtl(offset, 0x0000);         // Set base address of erase block
                                              // with dummy latch write

    NVMCON = 0x4042;                          // Initialize NVMCON

    asm("DISI #5");                           // Block all interrupts with priority <7
                                              // for next 5 instructions
    __builtin_write_NVM();                    // C30 function to perform unlock
                                              // sequence and set WR
```

EXAMPLE 5-3: LOADING THE WRITE BUFFERS (ASSEMBLY LANGUAGE CODE)

```
; Set up NVMCON for row programming operations
        MOV     #0x4001, W0                  ;
        MOV     W0, NVMCON                   ; Initialize NVMCON
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
        MOV     #0x0000, W0                  ;
        MOV     W0, TBLPAG                   ; Initialize PM Page Boundary SFR
        MOV     #0x6000, W0                  ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
        MOV     #LOW_WORD_0, W2              ;
        MOV     #HIGH_BYTE_0, W3             ;
        TBLWTL  W2, [W0]                     ; Write PM low word into program latch
        TBLWTH  W3, [W0++]                   ; Write PM high byte into program latch
; 1st_program_word
        MOV     #LOW_WORD_1, W2              ;
        MOV     #HIGH_BYTE_1, W3             ;
        TBLWTL  W2, [W0]                     ; Write PM low word into program latch
        TBLWTH  W3, [W0++]                   ; Write PM high byte into program latch
;  2nd_program_word
        MOV     #LOW_WORD_2, W2              ;
        MOV     #HIGH_BYTE_2, W3             ;
        TBLWTL  W2, [W0]                     ; Write PM low word into program latch
        TBLWTH  W3, [W0++]                   ; Write PM high byte into program latch
        •
        •
        •
; 63rd_program_word
        MOV     #LOW_WORD_31, W2             ;
        MOV     #HIGH_BYTE_31, W3            ;
        TBLWTL  W2, [W0]                     ; Write PM low word into program latch
        TBLWTH  W3, [W0]                     ; Write PM high byte into program latch
```

EXAMPLE 5-4: LOADING THE WRITE BUFFERS (C LANGUAGE CODE)

```
// C example using MPLAB C30

    #define NUM_INSTRUCTION_PER_ROW 64
    unsigned int offset;
    unsigned int i;
    unsigned long progAddr = 0xXXXXXX;              // Address of row to write
    unsigned int progData[2*NUM_INSTRUCTION_PER_ROW];   // Buffer of data to write

//Set up NVMCON for row programming
    NVMCON = 0x4001;                                // Initialize NVMCON

//Set up pointer to the first memory location to be written
    TBLPAG = progAddr>>16;                          // Initialize PM Page Boundary SFR
    offset = progAddr & 0xFFFF;                     // Initialize lower word of address

//Perform TBLWT instructions to write necessary number of latches
for(i=0; i < 2*NUM_INSTRUCTION_PER_ROW; i++)
    {
        __builtin_tblwtl(offset, progData[i++]);    // Write to address low word
        __builtin_tblwth(offset, progData[i]);      // Write to upper byte
        offset = offset + 2;                        // Increment address
    }
```

EXAMPLE 5-5: INITIATING A PROGRAMMING SEQUENCE (ASSEMBLY LANGUAGE CODE)

```
        DISI    #5                          ; Block all interrupts with priority <7
                                            ; for next 5 instructions
        MOV     #0x55, W0
        MOV     W0, NVMKEY                  ; Write the 55 key
        MOV     #0xAA, W1                   ;
        MOV     W1, NVMKEY                  ; Write the AA key
        BSET    NVMCON, #WR                 ; Start the erase sequence
        NOP                                 ;
        NOP                                 ;
        BTSC    NVMCON, #15                 ; and wait for it to be
        BRA     $-2                         ; completed
```

EXAMPLE 5-6: INITIATING A PROGRAMMING SEQUENCE (C LANGUAGE CODE)

```
// C example using MPLAB C30

asm("DISI #5");           // Block all interrupts with priority < 7
                          // for next 5 instructions

__builtin_write_NVM();    // Perform unlock sequence and set WR
```


5.6.2 PROGRAMMING A SINGLE WORD OF FLASH PROGRAM MEMORY

If a Flash location has been erased, it can be programmed using table write instructions to write an instruction word (24-bit) into the write latch. The TBLPAG register is loaded with the 8 Most Significant Bytes of the Flash address. The `TBLWTL` and `TBLWTH` instructions write the desired data into the write latches and specify the lower 16 bits of the program memory address to write to. To configure the NVMCON register for a word write, set the NVMOP bits (NVMCON<3:0>) to ‘`0011`’. The write is performed by executing the unlock sequence and setting the WR bit (see Example 5-7).

EXAMPLE 5-7: PROGRAMMING A SINGLE WORD OF FLASH PROGRAM MEMORY (ASSEMBLY LANGUAGE CODE)

```
; Setup a pointer to data Program Memory
    MOV     #tblpage(PROG_ADDR), W0       ;
    MOV     W0, TBLPAG                    ;Initialize PM Page Boundary SFR
    MOV     #tbloffset(PROG_ADDR), W0     ;Initialize a register with program memory address

    MOV     #LOW_WORD, W2                 ;
    MOV     #HIGH_BYTE, W3                ;
    TBLWTL  W2, [W0]                      ; Write PM low word into program latch
    TBLWTH  W3, [W0++]                    ; Write PM high byte into program latch

; Setup NVMCON for programming one word to data Program Memory
    MOV     #0x4003, W0                   ;
    MOV     W0, NVMCON                    ; Set NVMOP bits to 0011

    DISI    #5                            ; Disable interrupts while the KEY sequence is written
    MOV     #0x55, W0                     ; Write the key sequence
    MOV     W0, NVMKEY
    MOV     #0xAA, W0
    MOV     W0, NVMKEY
    BSET    NVMCON, #WR                   ; Start the write cycle
    NOP                                   ; Insert two NOPs after the erase
    NOP                                   ; Command is asserted
```

EXAMPLE 5-8: PROGRAMMING A SINGLE WORD OF FLASH PROGRAM MEMORY (C LANGUAGE CODE)

```
// C example using MPLAB C30

    unsigned int offset;
    unsigned long progAddr = 0xXXXXXX;          // Address of word to program
    unsigned int progDataL = 0xXXXX;            // Data to program lower word
    unsigned char progDataH = 0xXX;             // Data to program upper byte

//Set up NVMCON for word programming
    NVMCON = 0x4003;                            // Initialize NVMCON

//Set up pointer to the first memory location to be written
    TBLPAG = progAddr>>16;                      // Initialize PM Page Boundary SFR
    offset = progAddr & 0xFFFF;                 // Initialize lower word of address

//Perform TBLWT instructions to write latches
        __builtin_tblwtl(offset, progDataL);    // Write to address low word
        __builtin_tblwth(offset, progDataH);    // Write to upper byte
        asm(“DISI #5”);                         // Block interrupts with priority < 7
                                                // for next 5 instructions
        __builtin_write_NVM();                  // C30 function to perform unlock
                                                // sequence and set WR
```

NOTES:

6.0 RESETS

> **Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 7. "Reset"** (DS39712).

The Reset module combines all Reset sources and controls the device Master Reset Signal, SYSRST. The following is a list of device Reset sources:

- POR: Power-on Reset
- MCLR: Pin Reset
- SWR: RESET Instruction
- WDT: Watchdog Timer Reset
- BOR: Brown-out Reset
- CM: Configuration Mismatch Reset
- TRAPR: Trap Conflict Reset
- IOPUWR: Illegal Opcode Reset
- UWR: Uninitialized W Register Reset

A simplified block diagram of the Reset module is shown in Figure 6-1.

Any active source of Reset will make the SYSRST signal active. Many registers associated with the CPU and peripherals are forced to a known Reset state. Most registers are unaffected by a Reset; their status is unknown on POR and unchanged by all other Resets.

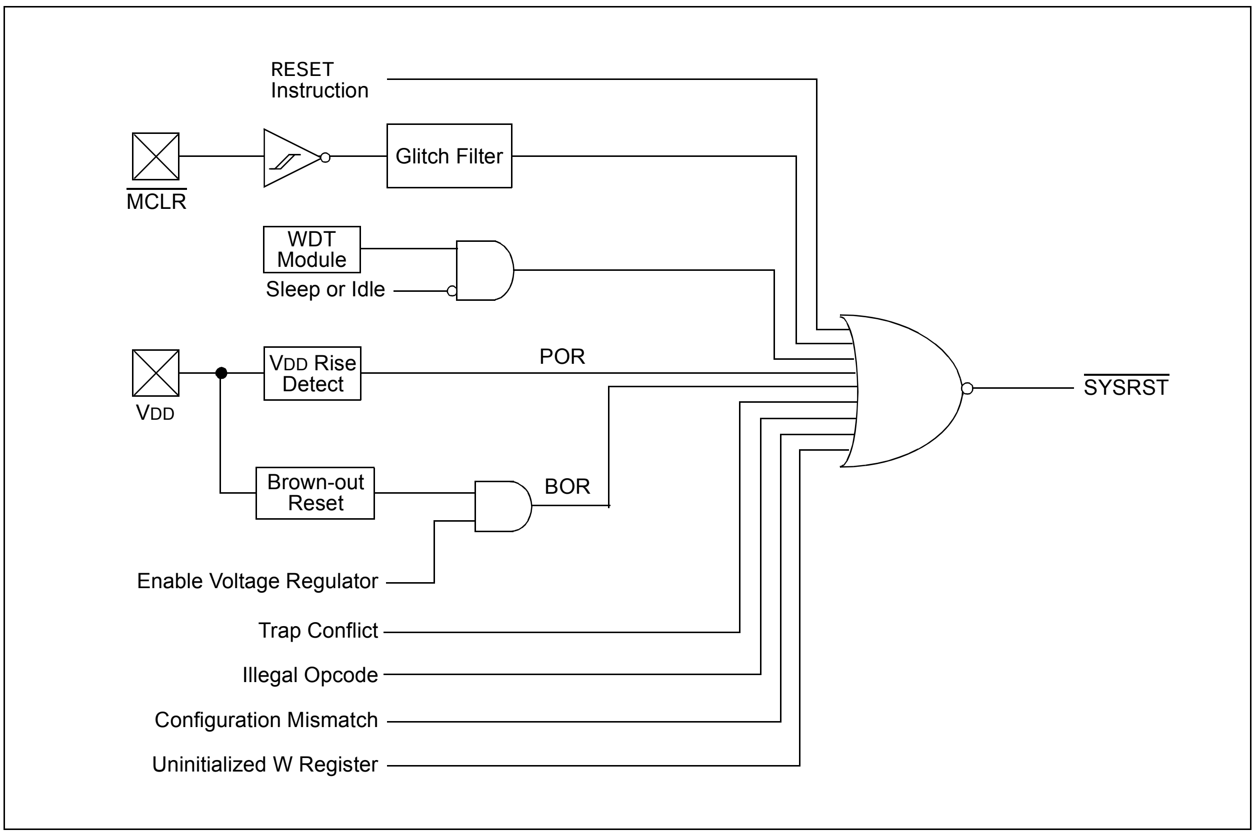
> **Note:** Refer to the specific peripheral or CPU section of this manual for register Reset states.

All types of device Reset will set a corresponding status bit in the RCON register to indicate the type of Reset (see Register 6-1). A Power-on Reset will clear all bits, except for the BOR and POR bits (RCON<1:0>), which are set. The user may set or clear any bit at any time during code execution. The RCON bits only serve as status bits. Setting a particular Reset status bit in software will not cause a device Reset to occur.

The RCON register also has other bits associated with the Watchdog Timer and device power-saving states. The function of these bits is discussed in other sections of this data sheet.

> **Note:** The status bits in the RCON register should be cleared after they are read so that the next RCON register value after a device Reset will be meaningful.

FIGURE 6-1: RESET SYSTEM BLOCK DIAGRAM



REGISTER 6-1: RCON: RESET CONTROL REGISTER(1)

R/W-0	R/W-0	U-0	U-0	U-0	R/CO-0, HS	R/W-0	R/W-0
TRAPR	IOPUWR	—	—	—	DPSLP	CM	PMSLP
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0,	R/W-0	R/W-1	R/W-1
EXTR	SWR	SWDTEN(2)	WDTO	SLEEP	IDLE	BOR	POR
bit 7							bit 0

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

bit 15 **TRAPR:** Trap Reset Flag bit
1 = A Trap Conflict Reset has occurred
0 = A Trap Conflict Reset has not occurred

bit 14 **IOPUWR:** Illegal Opcode or Uninitialized W Access Reset Flag bit
1 = An illegal opcode detection, an illegal address mode or uninitialized W register used as an Address Pointer caused a Reset
0 = An illegal opcode or uninitialized W Reset has not occurred

bit 13-11 **Unimplemented:** Read as '0'

bit 10 **DPSLP:** Deep Sleep Mode Flag bit
1 = Deep Sleep has occurred
0 = Deep Sleep has not occurred

bit 9 **CM:** Configuration Word Mismatch Reset Flag bit
1 = A Configuration Word Mismatch Reset has occurred
0 = A Configuration Word Mismatch Reset has not occurred

bit 8 **PMSLP:** Program Memory Power During Sleep bit
1 = Program memory bias voltage remains powered during Sleep
0 = Program memory bias voltage is powered down during Sleep and voltage regulator enters Standby mode

bit 7 **EXTR:** External Reset ($\overline{MCLR}$) Pin bit
1 = A Master Clear (pin) Reset has occurred
0 = A Master Clear (pin) Reset has not occurred

bit 6 **SWR:** Software Reset (Instruction) Flag bit
1 = A RESET instruction has been executed
0 = A RESET instruction has not been executed

bit 5 **SWDTEN:** Software Enable/Disable of WDT bit(2)
1 = WDT is enabled
0 = WDT is disabled

bit 4 **WDTO:** Watchdog Timer Time-out Flag bit
1 = WDT time-out has occurred
0 = WDT time-out has not occurred

bit 3 **SLEEP:** Wake From Sleep Flag bit
1 = Device has been in Sleep mode
0 = Device has not been in Sleep mode

bit 2 **IDLE:** Wake-up From Idle Flag bit
1 = Device has been in Idle mode
0 = Device has not been in Idle mode

Note 1: All of the Reset status bits may be set or cleared in software. Setting one of these bits in software does not cause a device Reset.

2: If the FWDTEN Configuration bit is '1' (unprogrammed), the WDT is always enabled, regardless of the SWDTEN bit setting.

REGISTER 6-1: RCON: RESET CONTROL REGISTER[1] (CONTINUED)

bit 1 **BOR:** Brown-out Reset Flag bit
`1` = A Brown-out Reset has occurred. Note that BOR is also set after a Power-on Reset.
`0` = A Brown-out Reset has not occurred

bit 0 **POR:** Power-on Reset Flag bit
`1` = A Power-on Reset has occurred
`0` = A Power-on Reset has not occurred

Note 1: All of the Reset status bits may be set or cleared in software. Setting one of these bits in software does not cause a device Reset.

2: If the FWDTEN Configuration bit is ‘1’ (unprogrammed), the WDT is always enabled, regardless of the SWDTEN bit setting.

TABLE 6-1: RESET FLAG BIT OPERATION

Flag Bit	Setting Event	Clearing Event
TRAPR (RCON<15>)	Trap Conflict Event	POR
IOPUWR (RCON<14>)	Illegal Opcode or Uninitialized W Register Access	POR
CM (RCON<9>)	Configuration Mismatch Reset	POR
EXTR (RCON<7>)	MCLR Reset	POR
SWR (RCON<6>)	`RESET` Instruction	POR
WDTO (RCON<4>)	WDT Time-out	`PWRSAV` Instruction, POR
SLEEP (RCON<3>)	`PWRSAV #SLEEP` Instruction	POR
IDLE (RCON<2>)	`PWRSAV #IDLE` Instruction	POR
BOR (RCON<1>)	POR, BOR	—
POR (RCON<0>)	POR	—
DPSLP (RCON<10>)	`PWRSAV #SLEEP` instruction with DSCON <DSEN> set	POR

Note: All Reset flag bits may be set or cleared by the user software.

6.1 Clock Source Selection at Reset

If clock switching is enabled, the system clock source at device Reset is chosen as shown in Table 6-2. If clock switching is disabled, the system clock source is always selected according to the oscillator Configuration bits. Refer to **Section 8.0 “Oscillator Configuration”** for further details.

TABLE 6-2: OSCILLATOR SELECTION vs. TYPE OF RESET (CLOCK SWITCHING ENABLED)

Reset Type	Clock Source Determinant
POR	FNOSC Configuration bits (CW2<10:8>)
BOR	
MCLR	COSC Control bits (OSCCON<14:12>)
WDTO	
SWR	

6.2 Device Reset Times

The Reset times for various types of device Reset are summarized in Table 6-3. Note that the System Reset signal, SYSRST, is released after the POR and PWRT delay times expire.

The time at which the device actually begins to execute code will also depend on the system oscillator delays, which include the Oscillator Start-up Timer (OST) and the PLL lock time. The OST and PLL lock times occur in parallel with the applicable SYSRST delay times.

The FSCM delay determines the time at which the FSCM begins to monitor the system clock source after the SYSRST signal is released.

TABLE 6-3: RESET DELAY TIMES FOR VARIOUS DEVICE RESETS

Reset Type	Clock Source	$\overline{\text{SYSRST}}$ Delay	System Clock Delay	Notes
POR(6)	EC	TPOR + TRST + TPWRT	—	**1, 2, 3, 8**
	FRC, FRCDIV	TPOR + TRST + TPWRT	TFRC	**1, 2, 3, 4, 7, 8**
	LPRC	TPOR + TRST + TPWRT	TLPRC	**1, 2, 3, 4, 8**
	ECPLL	TPOR + TRST + TPWRT	TLOCK	**1, 2, 3, 5, 8**
	FRCPLL	TPOR + TRST + TPWRT	TFRC + TLOCK	**1, 2, 3, 4, 5, 7, 8**
	XT, HS, SOSC	TPOR+ TRST + TPWRT	TOST	**1, 2, 3, 6, 8**
	XTPLL, HSPLL	TPOR + TRST + TPWRT	TOST + TLOCK	**1, 2, 3, 5, 6, 8**
BOR	EC	TRST + TPWRT	—	**2, 3, 8**
	FRC, FRCDIV	TRST + TPWRT	TFRC	**2, 3, 4, 7, 8**
	LPRC	TRST + TPWRT	TLPRC	**2, 3, 4, 8**
	ECPLL	TRST + TPWRT	TLOCK	**2, 3, 5, 8**
	FRCPLL	TRST + TPWRT	TFRC + TLOCK	**2, 3, 4, 5, 7, 8**
	XT, HS, SOSC	TRST + TPWRT	TOST	**2, 3, 6, 8**
	XTPLL, HSPLL	TRST + TPWRT	TFRC + TLOCK	**2, 3, 4, 5, 8**
All Others	Any Clock	TRST	—	**2, 8**

Note 1: TPOR = Power-on Reset delay.
2: TRST = Internal State Reset time.
3: TPWRT = 64 ms nominal if regulator is disabled (DISVREG tied to VDD).
4: TFRC and TLPRC = RC Oscillator start-up times.
5: TLOCK = PLL lock time.
6: TOST = Oscillator Start-up Timer (OST). A 10-bit counter waits 1024 oscillator periods before releasing the oscillator clock to the system.
7: If Two-Speed Start-up is enabled, regardless of the Primary Oscillator selected, the device starts with FRC, and in such cases, FRC start-up time is valid.
8: TRST = Configuration setup time.

Note:	For detailed operating frequency and timing specifications, see **Section 28.0 "Electrical Characteristics"**.

6.2.1 POR AND LONG OSCILLATOR START-UP TIMES

The oscillator start-up circuitry and its associated delay timers are not linked to the device Reset delays that occur at power-up. Some crystal circuits (especially low-frequency crystals) will have a relatively long start-up time. Therefore, one or more of the following conditions is possible after SYSRST is released:

- The oscillator circuit has not begun to oscillate.
- The Oscillator Start-up Timer has not expired (if a crystal oscillator is used).
- The PLL has not achieved a lock (if PLL is used).

The device will not begin to execute code until a valid clock source has been released to the system. Therefore, the oscillator and PLL start-up delays must be considered when the Reset delay time must be known.

6.2.2 FAIL-SAFE CLOCK MONITOR (FSCM) AND DEVICE RESETS

If the FSCM is enabled, it will begin to monitor the system clock source when SYSRST is released. If a valid clock source is not available at this time, the device will automatically switch to the FRC Oscillator and the user can switch to the desired crystal oscillator in the Trap Service Routine (TSR).

6.3 Special Function Register Reset States

Most of the Special Function Registers (SFRs) associated with the PIC24F CPU and peripherals are reset to a particular value at a device Reset. The SFRs are grouped by their peripheral or CPU function and their Reset values are specified in each section of this manual.

The Reset value for each SFR does not depend on the type of Reset with the exception of four registers. The Reset value for the Reset Control register, RCON, will depend on the type of device Reset. The Reset value for the Oscillator Control register, OSCCON, will depend on the type of Reset and the programmed values of the FNOSC bits in Flash Configuration Word 2 (CW2); see Table 6-2. The RCFGCAL and NVMCON registers are only affected by a POR.

6.4 Deep Sleep BOR (DSBOR)

Deep Sleep BOR is a very low-power BOR circuitry, used when the device is in Deep Sleep mode. Due to low-current consumption, accuracy may vary.

The DSBOR trip point is around 2.0V. DSBOR is enabled by configuring CW4 (DSBOREN) = 1. DSBOR will re-arm the POR to ensure the device will reset if VDD drops below the POR threshold.

NOTES:

7.0 INTERRUPT CONTROLLER

> **Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 8. "Interrupts"** (DS39707).

The PIC24F interrupt controller reduces the numerous peripheral interrupt request signals to a single interrupt request signal to the PIC24F CPU. It has the following features:

- Up to 8 processor exceptions and software traps
- 7 user-selectable priority levels
- Interrupt Vector Table (IVT) with up to 118 vectors
- A unique vector for each interrupt or exception source
- Fixed priority within a specified user priority level
- Alternate Interrupt Vector Table (AIVT) for debug support
- Fixed interrupt entry and return latencies

7.1 Interrupt Vector Table

The Interrupt Vector Table (IVT) is shown in Figure 7-1. The IVT resides in program memory, starting at location 000004h. The IVT contains 126 vectors, consisting of 8 non-maskable trap vectors, plus up to 118 sources of interrupt. In general, each interrupt source has its own vector. Each interrupt vector contains a 24-bit wide address. The value programmed into each interrupt vector location is the starting address of the associated Interrupt Service Routine (ISR).

Interrupt vectors are prioritized in terms of their natural priority; this is linked to their position in the vector table. All other things being equal, lower addresses have a higher natural priority. For example, the interrupt associated with vector 0 will take priority over interrupts at any other vector address.

PIC24FJ64GA104 family devices implement non-maskable traps and unique interrupts. These are summarized in Table 7-1 and Table 7-2.

7.1.1 ALTERNATE INTERRUPT VECTOR TABLE

The Alternate Interrupt Vector Table (AIVT) is located after the IVT, as shown in Figure 7-1. Access to the AIVT is provided by the ALTIVT control bit (INTCON2<15>). If the ALTIVT bit is set, all interrupt and exception processes will use the alternate vectors instead of the default vectors. The alternate vectors are organized in the same manner as the default vectors.

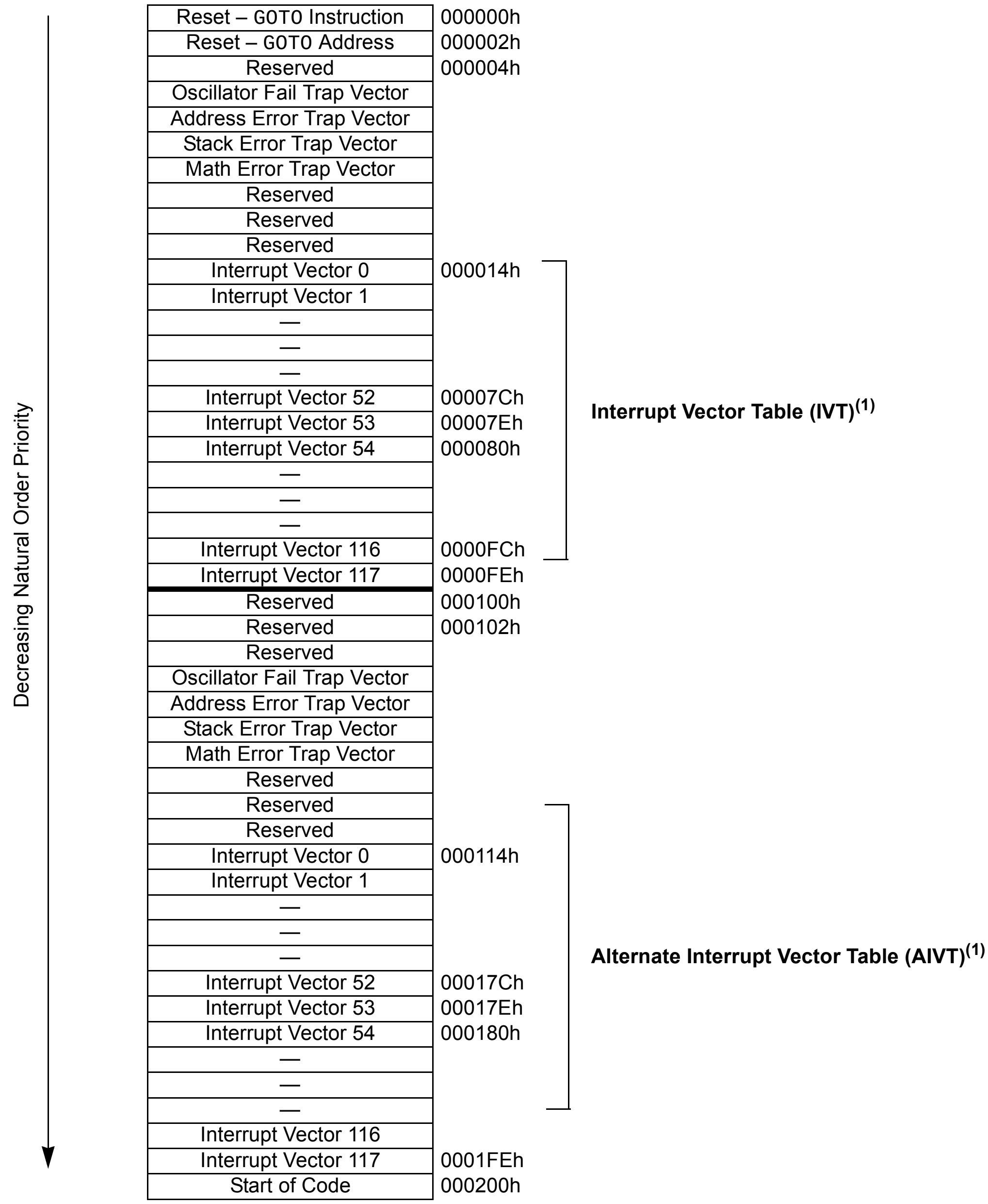
The AIVT supports emulation and debugging efforts by providing a means to switch between an application and a support environment without requiring the interrupt vectors to be reprogrammed. This feature also enables switching between applications for evaluation of different software algorithms at run time. If the AIVT is not needed, the AIVT should be programmed with the same addresses used in the IVT.

7.2 Reset Sequence

A device Reset is not a true exception because the interrupt controller is not involved in the Reset process. The PIC24F devices clear their registers in response to a Reset which forces the PC to zero. The microcontroller then begins program execution at location 000000h. The user programs a `GOTO` instruction at the Reset address, which redirects program execution to the appropriate start-up routine.

> **Note:** Any unimplemented or unused vector locations in the IVT and AIVT should be programmed with the address of a default interrupt handler routine that contains a `RESET` instruction.

FIGURE 7-1: PIC24F INTERRUPT VECTOR TABLE



Note 1: See Table 7-2 for the interrupt vector list.

TABLE 7-1: TRAP VECTOR DETAILS

Vector Number	IVT Address	AIVT Address	Trap Source
0	000004h	000104h	Reserved
1	000006h	000106h	Oscillator Failure
2	000008h	000108h	Address Error
3	00000Ah	00010Ah	Stack Error
4	00000Ch	00010Ch	Math Error
5	00000Eh	00010Eh	Reserved
6	000010h	000110h	Reserved
7	000012h	000112h	Reserved

TABLE 7-2: IMPLEMENTED INTERRUPT VECTORS

Interrupt Source	Vector Number	IVT Address	AIVT Address	Interrupt Bit Locations		
				Flag	Enable	Priority
ADC1 Conversion Done	13	00002Eh	00012Eh	IFS0<13>	IEC0<13>	IPC3<6:4>
Comparator Event	18	000038h	000138h	IFS1<2>	IEC1<2>	IPC4<10:8>
CRC Generator	67	00009Ah	00019Ah	IFS4<3>	IEC4<3>	IPC16<14:12>
CTMU Event	77	0000AEh	0001AEh	IFS4<13>	IEC4<13>	IPC19<6:4>
External Interrupt 0	0	000014h	000114h	IFS0<0>	IEC0<0>	IPC0<2:0>
External Interrupt 1	20	00003Ch	00013Ch	IFS1<4>	IEC1<4>	IPC5<2:0>
External Interrupt 2	29	00004Eh	00014Eh	IFS1<13>	IEC1<13>	IPC7<6:4>
I2C1 Master Event	17	000036h	000136h	IFS1<1>	IEC1<1>	IPC4<6:4>
I2C1 Slave Event	16	000034h	000134h	IFS1<0>	IEC1<0>	IPC4<2:0>
I2C2 Master Event	50	000078h	000178h	IFS3<2>	IEC3<2>	IPC12<10:8>
I2C2 Slave Event	49	000076h	000176h	IFS3<1>	IEC3<1>	IPC12<6:4>
Input Capture 1	1	000016h	000116h	IFS0<1>	IEC0<1>	IPC0<6:4>
Input Capture 2	5	00001Eh	00011Eh	IFS0<5>	IEC0<5>	IPC1<6:4>
Input Capture 3	37	00005Eh	00015Eh	IFS2<5>	IEC2<5>	IPC9<6:4>
Input Capture 4	38	000060h	000160h	IFS2<6>	IEC2<6>	IPC9<10:8>
Input Capture 5	39	000062h	000162h	IFS2<7>	IEC2<7>	IPC9<14:12>
Input Change Notification	19	00003Ah	00013Ah	IFS1<3>	IEC1<3>	IPC4<14:12>
LVD Low-Voltage Detect	72	0000A4h	0001A4h	IFS4<8>	IEC4<8>	IPC18<2:0>
Output Compare 1	2	000018h	000118h	IFS0<2>	IEC0<2>	IPC0<10:8>
Output Compare 2	6	000020h	000120h	IFS0<6>	IEC0<6>	IPC1<10:8>
Output Compare 3	25	000046h	000146h	IFS1<9>	IEC1<9>	IPC6<6:4>
Output Compare 4	26	000048h	000148h	IFS1<10>	IEC1<10>	IPC6<10:8>
Output Compare 5	41	000066h	000166h	IFS2<9>	IEC2<9>	IPC10<6:4>
Parallel Master Port	45	00006Eh	00016Eh	IFS2<13>	IEC2<13>	IPC11<6:4>
Real-Time Clock/Calendar	62	000090h	000190h	IFS3<14>	IEC3<14>	IPC15<10:8>
SPI1 Error	9	000026h	000126h	IFS0<9>	IEC0<9>	IPC2<6:4>
SPI1 Event	10	000028h	000128h	IFS0<10>	IEC0<10>	IPC2<10:8>
SPI2 Error	32	000054h	000154h	IFS2<0>	IEC2<0>	IPC8<2:0>
SPI2 Event	33	000056h	000156h	IFS2<1>	IEC2<1>	IPC8<6:4>
Timer1	3	00001Ah	00011Ah	IFS0<3>	IEC0<3>	IPC0<14:12>
Timer2	7	000022h	000122h	IFS0<7>	IEC0<7>	IPC1<14:12>
Timer3	8	000024h	000124h	IFS0<8>	IEC0<8>	IPC2<2:0>
Timer4	27	00004Ah	00014Ah	IFS1<11>	IEC1<11>	IPC6<14:12>
Timer5	28	00004Ch	00014Ch	IFS1<12>	IEC1<12>	IPC7<2:0>
UART1 Error	65	000096h	000196h	IFS4<1>	IEC4<1>	IPC16<6:4>
UART1 Receiver	11	00002Ah	00012Ah	IFS0<11>	IEC0<11>	IPC2<14:12>
UART1 Transmitter	12	00002Ch	00012Ch	IFS0<12>	IEC0<12>	IPC3<2:0>
UART2 Error	66	000098h	000198h	IFS4<2>	IEC4<2>	IPC16<10:8>
UART2 Receiver	30	000050h	000150h	IFS1<14>	IEC1<14>	IPC7<10:8>
UART2 Transmitter	31	000052h	000152h	IFS1<15>	IEC1<15>	IPC7<14:12>

7.3 Interrupt Control and Status Registers

The PIC24FJ64GA104 family of devices implements the following registers for the interrupt controller:

- INTCON1
- INTCON2
- IFS0 through IFS4
- IEC0 through IEC4
- IPC0 through IPC20 (except IPC13, IPC14 and IPC17)
- INTTREG

Global interrupt control functions are controlled from INTCON1 and INTCON2. INTCON1 contains the Interrupt Nesting Disable (NSTDIS) bit, as well as the control and status flags for the processor trap sources. The INTCON2 register controls the external interrupt request signal behavior and the use of the Alternate Interrupt Vector Table.

The IFSx registers maintain all of the interrupt request flags. Each source of interrupt has a status bit which is set by the respective peripherals, or an external signal, and is cleared via software.

The IECx registers maintain all of the interrupt enable bits. These control bits are used to individually enable interrupts from the peripherals or external signals.

The IPCx registers are used to set the interrupt priority level for each source of interrupt. Each user interrupt source can be assigned to one of eight priority levels.

The interrupt sources are assigned to the IFSx, IECx and IPCx registers in the order of their vector numbers, as shown in Table 7-2. For example, the INT0 (External Interrupt 0) is shown as having a vector number and a natural order priority of 0. Thus, the INT0IF status bit is found in IFS0<0>, the INT0IE enable bit in IEC0<0> and the INT0IP<2:0> priority bits in the first position of IPC0 (IPC0<2:0>).

Although they are not specifically part of the interrupt control hardware, two of the CPU control registers contain bits that control interrupt functionality. The ALU STATUS Register (SR) contains the IPL<2:0> bits (SR<7:5>); these indicate the current CPU interrupt priority level. The user may change the current CPU priority level by writing to the IPL bits.

The CORCON register contains the IPL3 bit, which, together with IPL<2:0>, indicates the current CPU priority level. IPL3 is a read-only bit so that trap events cannot be masked by the user software.

The interrupt controller has the Interrupt Controller Test Register (INTTREG) that displays the status of the interrupt controller. When an interrupt request occurs, its associated vector number and the new interrupt priority level are latched into INTTREG.

This information can be used to determine a specific interrupt source if a generic ISR is used for multiple vectors – such as when ISR remapping is used in bootloader applications. It also could be used to check if another interrupt is pending while in an ISR.

All interrupt registers are described in Register 7-1 through Register 7-32, on the following pages.

REGISTER 7-1: SR: ALU STATUS REGISTER (IN CPU)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R-0
—	—	—	—	—	—	—	DC[1]
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2[2,3]	IPL1[2,3]	IPL0[2,3]	RA[1]	N[1]	OV[1]	Z[1]	C[1]
bit 7							bit 0

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

bit 7-5 **IPL<2:0>:** CPU Interrupt Priority Level Status bits[2,3]
- 111 = CPU interrupt priority level is 7 (15). User interrupts are disabled.
- 110 = CPU interrupt priority level is 6 (14)
- 101 = CPU interrupt priority level is 5 (13)
- 100 = CPU interrupt priority level is 4 (12)
- 011 = CPU interrupt priority level is 3 (11)
- 010 = CPU interrupt priority level is 2 (10)
- 001 = CPU interrupt priority level is 1 (9)
- 000 = CPU interrupt priority level is 0 (8)

Note 1: See Register 3-1 for the description of the remaining bit(s) that are not dedicated to interrupt control functions.

2: The IPL bits are concatenated with the IPL3 bit (CORCON<3>) to form the CPU interrupt priority level. The value in parentheses indicates the interrupt priority level if IPL3 = 1.

3: The IPL Status bits are read-only when NSTDIS (INTCON1<15>) = 1.

REGISTER 7-2: CORCON: CPU CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/C-0	R/W-0	U-0	U-0
—	—	—	—	IPL3[2]	PSV[1]	—	—
bit 7							bit 0

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

bit 3 **IPL3:** CPU Interrupt Priority Level Status bit[2]
- 1 = CPU interrupt priority level is greater than 7
- 0 = CPU interrupt priority level is 7 or less

Note 1: See Register 3-2 for the description of the remaining bit(s) that are not dedicated to interrupt control functions.

2: The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU interrupt priority level.

REGISTER 7-3: INTCON1: INTERRUPT CONTROL REGISTER 1

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
NSTDIS	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—
bit 7							bit 0

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

bit 15 **NSTDIS:** Interrupt Nesting Disable bit
1 = Interrupt nesting is disabled
0 = Interrupt nesting is enabled

bit 14-5 **Unimplemented:** Read as ‘0’

bit 4 **MATHERR:** Arithmetic Error Trap Status bit
1 = Overflow trap has occurred
0 = Overflow trap has not occurred

bit 3 **ADDRERR:** Address Error Trap Status bit
1 = Address error trap has occurred
0 = Address error trap has not occurred

bit 2 **STKERR:** Stack Error Trap Status bit
1 = Stack error trap has occurred
0 = Stack error trap has not occurred

bit 1 **OSCFAIL:** Oscillator Failure Trap Status bit
1 = Oscillator failure trap has occurred
0 = Oscillator failure trap has not occurred

bit 0 **Unimplemented:** Read as ‘0’

REGISTER 7-4: INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-0	R-0	U-0	U-0	U-0	U-0	U-0	U-0
ALTIVT	DISI	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	INT2EP	INT1EP	INT0EP
bit 7							bit 0

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

bit 15 **ALTIVT:** Enable Alternate Interrupt Vector Table bit
1 = Use Alternate Interrupt Vector Table
0 = Use standard (default) vector table

bit 14 **DISI:** DISI Instruction Status bit
1 = DISI instruction is active
0 = DISI instruction is not active

bit 13-3 **Unimplemented:** Read as ‘0’

bit 2 **INT2EP:** External Interrupt 2 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 1 **INT1EP:** External Interrupt 1 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 0 **INT0EP:** External Interrupt 0 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

REGISTER 7-5: IFS0: INTERRUPT FLAG STATUS REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPF1IF	T3IF
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT0IF
bit 7							bit 0

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

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13 **AD1IF:** A/D Conversion Complete Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12 **U1TXIF:** UART1 Transmitter Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 11 **U1RXIF:** UART1 Receiver Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 10 **SPI1IF:** SPI1 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 9 **SPF1IF:** SPI1 Fault Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 8 **T3IF:** Timer3 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 7 **T2IF:** Timer2 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6 **OC2IF:** Output Compare Channel 2 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **IC2IF:** Input Capture Channel 2 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4 **Unimplemented:** Read as ‘0’

bit 3 **T1IF:** Timer1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 2 **OC1IF:** Output Compare Channel 1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **IC1IF:** Input Capture Channel 1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **INT0IF:** External Interrupt 0 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 7-6: IFS1: INTERRUPT FLAG STATUS REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF
bit 7							bit 0

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

bit 15 **U2TXIF:** UART2 Transmitter Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 14 **U2RXIF:** UART2 Receiver Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 13 **INT2IF:** External Interrupt 2 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12 **T5IF:** Timer5 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 11 **T4IF:** Timer4 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 10 **OC4IF:** Output Compare Channel 4 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 9 **OC3IF:** Output Compare Channel 3 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 8-5 **Unimplemented:** Read as ‘0’

bit 4 **INT1IF:** External Interrupt 1 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 3 **CNIF:** Input Change Notification Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 2 **CMIF:** Comparator Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **MI2C1IF:** Master I2C1 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **SI2C1IF:** Slave I2C1 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 7-7: IFS2: INTERRUPT FLAG STATUS REGISTER 2

U-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	U-0
—	—	PMPIF	—	—	—	OC5IF	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
IC5IF	IC4IF	IC3IF	—	—	—	SPI2IF	SPF2IF
bit 7							bit 0

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

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13 **PMPIF:** Parallel Master Port Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12-10 **Unimplemented:** Read as ‘0’

bit 9 **OC5IF:** Output Compare Channel 5 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 8 **Unimplemented:** Read as ‘0’

bit 7 **IC5IF:** Input Capture Channel 5 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6 **IC4IF:** Input Capture Channel 4 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **IC3IF:** Input Capture Channel 3 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4-2 **Unimplemented:** Read as ‘0’

bit 1 **SPI2IF:** SPI2 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **SPF2IF:** SPI2 Fault Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 7-8: IFS3: INTERRUPT FLAG STATUS REGISTER 3

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	RTCIF	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0,	R/W-0	U-0
—	—	—	—	—	MI2C2IF	SI2C2IF	—
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as ‘0’

bit 14 **RTCIF:** Real-Time Clock/Calendar Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 13-3 **Unimplemented:** Read as ‘0’

bit 2 **MI2C2IF:** Master I2C2 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **SI2C2IF:** Slave I2C2 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **Unimplemented:** Read as ‘0’

REGISTER 7-9: IFS4: INTERRUPT FLAG STATUS REGISTER 4

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
—	—	CTMUIF	—	—	—	—	LVDIF
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	—	CRCIF	U2ERIF	U1ERIF	—
bit 7							bit 0

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

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **CTMUIF:** CTMU Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12-9 **Unimplemented:** Read as '0'

bit 8 **LVDIF:** Low-Voltage Detect Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **CRCIF:** CRC Generator Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 2 **U2ERIF:** UART2 Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **U1ERIF:** UART1 Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

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

REGISTER 7-10: IEC0: INTERRUPT ENABLE CONTROL REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPF1IE	T3IE
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
T2IE	OC2IE	IC2IE	—	T1IE	OC1IE	IC1IE	INT0IE
bit 7							bit 0

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

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13 **AD1IE:** A/D Conversion Complete Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 12 **U1TXIE:** UART1 Transmitter Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 11 **U1RXIE:** UART1 Receiver Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 10 **SPI1IE:** SPI1 Transfer Complete Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 9 **SPF1IE:** SPI1 Fault Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 8 **T3IE:** Timer3 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 7 **T2IE:** Timer2 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 6 **OC2IE:** Output Compare Channel 2 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 5 **IC2IE:** Input Capture Channel 2 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 4 **Unimplemented:** Read as ‘0’

bit 3 **T1IE:** Timer1 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 2 **OC1IE:** Output Compare Channel 1 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 1 **IC1IE:** Input Capture Channel 1 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 0 **INT0IE:** External Interrupt 0 Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

REGISTER 7-11: IEC1: INTERRUPT ENABLE CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
U2TXIE	U2RXIE	INT2IE[(1)]	T5IE	T4IE	OC4IE	OC3IE	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT1IE[(1)]	CNIE	CMIE	MI2C1IE	SI2C1IE
bit 7							bit 0

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

bit 15	**U2TXIE:** UART2 Transmitter Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 14	**U2RXIE:** UART2 Receiver Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 13	**INT2IE:** External Interrupt 2 Enable bit[(1)] 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 12	**T5IE:** Timer5 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 11	**T4IE:** Timer4 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 10	**OC4IE:** Output Compare Channel 4 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 9	**OC3IE:** Output Compare Channel 3 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 8-5	**Unimplemented:** Read as ‘0’
bit 4	**INT1IE:** External Interrupt 1 Enable bit[(1)] 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 3	**CNIE:** Input Change Notification Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 2	**CMIE:** Comparator Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 1	**MI2C1IE:** Master I2C1 Event Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 0	**SI2C1IE:** Slave I2C1 Event Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled

Note 1: If an external interrupt is enabled, the interrupt input must also be configured to an available RPn or PRIx pin. See **Section 10.4 “Peripheral Pin Select (PPS)”** for more information.

REGISTER 7-12: IEC2: INTERRUPT ENABLE CONTROL REGISTER 2

U-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	U-0
—	—	PMPIE	—	—	—	OC5IE	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
IC5IE	IC4IE	IC3IE	—	—	—	SPI2IE	SPF2IE
bit 7							bit 0

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

bit 15-14	**Unimplemented:** Read as ‘0’
bit 13	**PMPIE:** Parallel Master Port Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 12-10	**Unimplemented:** Read as ‘0’
bit 9	**OC5IE:** Output Compare Channel 5 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 8	**Unimplemented:** Read as ‘0’
bit 7	**IC5IE:** Input Capture Channel 5 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 6	**IC4IE:** Input Capture Channel 4 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 5	**IC3IE:** Input Capture Channel 3 Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 4-2	**Unimplemented:** Read as ‘0’
bit 1	**SPI2IE:** SPI2 Event Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled
bit 0	**SPF2IE:** SPI2 Fault Interrupt Enable bit 1 = Interrupt request enabled 0 = Interrupt request not enabled

REGISTER 7-13: IEC3: INTERRUPT ENABLE CONTROL REGISTER 3

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	RTCIE	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0
—	—	—	—	—	MI2C2IE	SI2C2IE	—
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as ‘0’

bit 14 **RTCIE:** Real-Time Clock/Calendar Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 13-3 **Unimplemented:** Read as ‘0’

bit 2 **MI2C2IE:** Master I2C2 Event Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 1 **SI2C2IE:** Slave I2C2 Event Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 0 **Unimplemented:** Read as ‘0’

REGISTER 7-14: IEC4: INTERRUPT ENABLE CONTROL REGISTER 4

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
—	—	CTMUIE	—	—	—	—	LVDIE
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	—	CRCIE	U2ERIE	U1ERIE	—
bit 7							bit 0

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

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13 **CTMUIE:** CTMU Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 12-9 **Unimplemented:** Read as ‘0’

bit 8 **LVDIE:** Low-Voltage Detect Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3 **CRCIE:** CRC Generator Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 2 **U2ERIE:** UART2 Error Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 1 **U1ERIE:** UART1 Error Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 0 **Unimplemented:** Read as ‘0’

REGISTER 7-15: IPC0: INTERRUPT PRIORITY CONTROL REGISTER 0

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **T1IP<2:0>:** Timer1 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **OC1IP<2:0>:** Output Compare Channel 1 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

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

bit 6-4 **IC1IP<2:0>:** Input Capture Channel 1 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **INT0IP<2:0>:** External Interrupt 0 Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-16: IPC1: INTERRUPT PRIORITY CONTROL REGISTER 1

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T2IP2	T2IP1	T2IP0	—	OC2IP2	OC2IP1	OC2IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	IC2IP2	IC2IP1	IC2IP0	—	—	—	—
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **T2IP<2:0>:** Timer2 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **OC2IP<2:0>:** Output Compare Channel 2 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

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

bit 6-4 **IC2IP<2:0>:** Input Capture Channel 2 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3-0 **Unimplemented:** Read as '0'

REGISTER 7-17: IPC2: INTERRUPT PRIORITY CONTROL REGISTER 2

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPF1IP2	SPF1IP1	SPF1IP0	—	T3IP2	T3IP1	T3IP0
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **U1RXIP<2:0>:** UART1 Receiver Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **SPI1IP<2:0>:** SPI1 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

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

bit 6-4 **SPF1IP<2:0>:** SPI1 Fault Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **T3IP<2:0>:** Timer3 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-18: IPC3: INTERRUPT PRIORITY CONTROL REGISTER 3

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0
bit 7							bit 0

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

bit 15-7 **Unimplemented:** Read as ‘0’

bit 6-4 **AD1IP<2:0>:** A/D Conversion Complete Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **U1TXIP<2:0>:** UART1 Transmitter Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-19: IPC4: INTERRUPT PRIORITY CONTROL REGISTER 4

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	CNIP2	CNIP1	CNIP0	—	CMIP2	CMIP1	CMIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	MI2C1IP2	MI2C1IP1	MI2C1IP0	—	SI2C1IP2	SI2C1IP1	SI2C1IP0
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **CNIP<2:0>:** Input Change Notification Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **CMIP<2:0>:** Comparator Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **MI2C1IP<2:0>:** Master I2C1 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **SI2C1IP<2:0>:** Slave I2C1 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-20: IPC5: INTERRUPT PRIORITY CONTROL REGISTER 5

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	INT1IP2	INT1IP1	INT1IP0
bit 7							bit 0

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

bit 15-3 **Unimplemented:** Read as ‘0’

bit 2-0 **INT1IP<2:0>:** External Interrupt 1 Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-21: IPC6: INTERRUPT PRIORITY CONTROL REGISTER 6

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T4IP2	T4IP1	T4IP0	—	OC4IP2	OC4IP1	OC4IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	OC3IP2	OC3IP1	OC3IP0	—	—	—	—
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **T4IP<2:0>:** Timer4 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **OC4IP<2:0>:** Output Compare Channel 4 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **OC3IP<2:0>:** Output Compare Channel 3 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

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

REGISTER 7-22: IPC7: INTERRUPT PRIORITY CONTROL REGISTER 7

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	INT2IP2	INT2IP1	INT2IP0	—	T5IP2	T5IP1	T5IP0
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **U2TXIP<2:0>:** UART2 Transmitter Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **U2RXIP<2:0>:** UART2 Receiver Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **INT2IP<2:0>:** External Interrupt 2 Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **T5IP<2:0>:** Timer5 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-23: IPC8: INTERRUPT PRIORITY CONTROL REGISTER 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPI2IP2	SPI2IP1	SPI2IP0	—	SPF2IP2	SPF2IP1	SPF2IP0
bit 7							bit 0

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

bit 15-7 **Unimplemented:** Read as '0'

bit 6-4 **SPI2IP<2:0>:** SPI2 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **SPF2IP<2:0>:** SPI2 Fault Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-24: IPC9: INTERRUPT PRIORITY CONTROL REGISTER 9

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC5IP2	IC5IP1	IC5IP0	—	IC4IP2	IC4IP1	IC4IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	IC3IP2	IC3IP1	IC3IP0	—	—	—	—
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **IC5IP<2:0>:** Input Capture Channel 5 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **IC4IP<2:0>:** Input Capture Channel 4 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **IC3IP<2:0>:** Input Capture Channel 3 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

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

REGISTER 7-25: IPC10: INTERRUPT PRIORITY CONTROL REGISTER 10

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	OC5IP2	OC5IP1	OC5IP0	—	—	—	—
bit 7							bit 0

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

bit 15-7 **Unimplemented:** Read as '0'

bit 6-4 **OC5IP<2:0>:** Output Compare Channel 5 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3-0 **Unimplemented:** Read as '0'

REGISTER 7-26: IPC11: INTERRUPT PRIORITY CONTROL REGISTER 11

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	PMPIP2	PMPIP1	PMPIP0	—	—	—	—
bit 7							bit 0

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

bit 15-7 **Unimplemented:** Read as ‘0’

bit 6-4 **PMPIP<2:0>:** Parallel Master Port Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

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

REGISTER 7-27: IPC12: INTERRUPT PRIORITY CONTROL REGISTER 12

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	MI2C2IP2	MI2C2IP1	MI2C2IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	SI2C2IP2	SI2C2IP1	SI2C2IP0	—	—	—	—
bit 7							bit 0

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

bit 15-11 **Unimplemented:** Read as ‘0’

bit 10-8 **MI2C2IP<2:0>:** Master I2C2 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **SI2C2IP<2:0>:** Slave I2C2 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

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

REGISTER 7-28: IPC15: INTERRUPT PRIORITY CONTROL REGISTER 15

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	RTCIP2	RTCIP1	RTCIP0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 15-11 **Unimplemented:** Read as '0'

bit 10-8 **RTCIP<2:0>:** Real-Time Clock/Calendar Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

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

REGISTER 7-29: IPC16: INTERRUPT PRIORITY CONTROL REGISTER 16

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **CRCIP<2:0>:** CRC Generator Error Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **U2ERIP<2:0>:** UART2 Error Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **U1ERIP<2:0>:** UART1 Error Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

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

REGISTER 7-30: IPC18: INTERRUPT PRIORITY CONTROL REGISTER 18

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	LVDIP2	LVDIP1	LVDIP0
bit 7							bit 0

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

bit 15-3 **Unimplemented:** Read as '0'

bit 2-0 **LVDIP<2:0>:** Low-Voltage Detect Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-31: IPC19: INTERRUPT PRIORITY CONTROL REGISTER 19

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	CTMUIP2	CTMUIP1	CTMUIP0	—	—	—	—
bit 7							bit 0

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

bit 15-7 **Unimplemented:** Read as '0'

bit 6-4 **CTMUIP<2:0>:** CTMU Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3-0 **Unimplemented:** Read as '0'

REGISTER 7-32: INTTREG: INTERRUPT CONTROL AND STATUS REGISTER

R-0	U-0	R/W-0	U-0	R-0	R-0	R-0	R-0
CPUIRQ	—	VHOLD	—	ILR3	ILR2	ILR1	ILR0
bit 15							bit 8

U-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
—	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0
bit 7							bit 0

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

bit 15 **CPUIRQ:** Interrupt Request from Interrupt Controller CPU bit
1 = An interrupt request has occurred but has not yet been Acknowledged by the CPU; this happens when the CPU priority is higher than the interrupt priority
0 = No interrupt request is unacknowledged

bit 14 Unimplemented: Read as '0'

bit 13 **VHOLD:** Vector Number Capture Configuration bit
1 = The VECNUM bits contain the value of the highest priority pending interrupt
0 = The VECNUM bits contain the value of the last Acknowledged interrupt (i.e., the last interrupt that has occurred with higher priority than the CPU, even if other interrupts are pending)

bit 12 Unimplemented: Read as '0'

bit 11-8 **ILR<3:0>:** New CPU Interrupt Priority Level bits
1111 = CPU Interrupt Priority Level is 15
•
•
•
0001 = CPU Interrupt Priority Level is 1
0000 = CPU Interrupt Priority Level is 0

bit 7 Unimplemented: Read as '0'

bit 6-0 **VECNUM<6:0>:** Pending Interrupt Vector ID bits (pending vector number is VECNUM + 8)
0111111 = Interrupt Vector pending is number 135
•
•
•
0000001 = Interrupt Vector pending is number 9
0000000 = Interrupt Vector pending is number 8

7.4 Interrupt Setup Procedures

7.4.1 INITIALIZATION

To configure an interrupt source:

1. Set the NSTDIS control bit (INTCON1<15>) if nested interrupts are not desired.
2. Select the user-assigned priority level for the interrupt source by writing the control bits in the appropriate IPCx register. The priority level will depend on the specific application and type of interrupt source. If multiple priority levels are not desired, the IPCx register control bits for all enabled interrupt sources may be programmed to the same non-zero value.

Note:	At a device Reset, the IPCx registers are initialized, such that all user interrupt sources are assigned to priority level 4.

3. Clear the interrupt flag status bit associated with the peripheral in the associated IFSx register.
4. Enable the interrupt source by setting the interrupt enable control bit associated with the source in the appropriate IECx register.

7.4.2 INTERRUPT SERVICE ROUTINE

The method that is used to declare an ISR and initialize the IVT with the correct vector address will depend on the programming language (i.e., 'C' or assembler) and the language development toolsuite that is used to develop the application. In general, the user must clear the interrupt flag in the appropriate IFSx register for the source of the interrupt that the ISR handles. Otherwise, the ISR will be re-entered immediately after exiting the routine. If the ISR is coded in assembly language, it must be terminated using a `RETFIE` instruction to unstack the saved PC value, SRL value and old CPU priority level.

7.4.3 TRAP SERVICE ROUTINE

A Trap Service Routine (TSR) is coded like an ISR, except that the appropriate trap status flag in the INTCON1 register must be cleared to avoid re-entry into the TSR.

7.4.4 INTERRUPT DISABLE

All user interrupts can be disabled using the following procedure:

1. Push the current SR value onto the software stack using the `PUSH` instruction.
2. Force the CPU to priority level 7 by inclusive ORing the value OEh with SRL.

To enable user interrupts, the `POP` instruction may be used to restore the previous SR value.

Note that only user interrupts with a priority level of 7 or less can be disabled. Trap sources (level 8-15) cannot be disabled.

The `DISI` instruction provides a convenient way to disable interrupts of priority levels 1-6 for a fixed period of time. Level 7 interrupt sources are not disabled by the `DISI` instruction.

NOTES:

8.0 OSCILLATOR CONFIGURATION

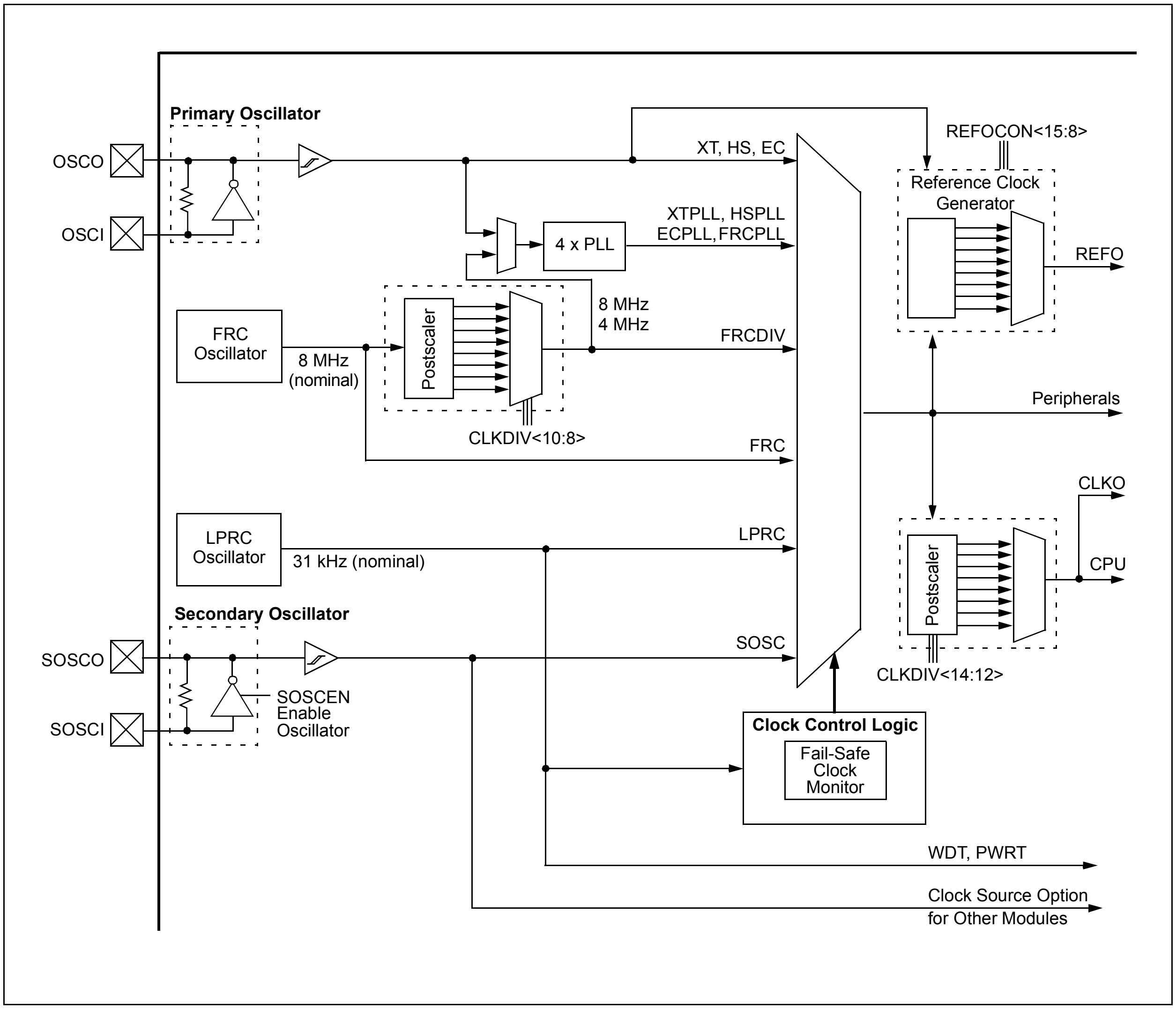
Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **"Section 6. Oscillator"** (DS39700).

The oscillator system for PIC24FJ64GA104 family devices has the following features:

- A total of four external and internal oscillator options as clock sources, providing 11 different clock modes
- On-chip 4x PLL to boost internal operating frequency on select internal and external oscillator sources
- Software-controllable switching between various clock sources
- Software-controllable postscaler for selective clocking of CPU for system power savings
- A Fail-Safe Clock Monitor (FSCM) that detects clock failure and permits safe application recovery or shutdown
- A separate and independently configurable system clock output for synchronizing external hardware

A simplified diagram of the oscillator system is shown in Figure 8-1.

FIGURE 8-1: PIC24FJ64GA104 FAMILY CLOCK DIAGRAM



8.1 CPU Clocking Scheme

The system clock source can be provided by one of four sources:

- Primary Oscillator (POSC) on the OSCI and OSCO pins
- Secondary Oscillator (SOSC) on the SOSCI and SOSCO pins
- Fast Internal RC (FRC) Oscillator
- Low-Power Internal RC (LPRC) Oscillator

The Primary Oscillator and FRC sources have the option of using the internal 4x PLL. The frequency of the FRC clock source can optionally be reduced by the programmable clock divider. The selected clock source generates the processor and peripheral clock sources.

The processor clock source is divided by two to produce the internal instruction cycle clock, FCY. In this document, the instruction cycle clock is also denoted by FOSC/2. The internal instruction cycle clock, FOSC/2, can be provided on the OSCO I/O pin for some operating modes of the Primary Oscillator.

8.2 Initial Configuration on POR

The oscillator source (and operating mode) that is used at a device Power-on Reset event is selected using Configuration bit settings. The oscillator Configuration bit settings are located in the Configuration registers in the program memory (refer to **Section 25.1 "Configuration Bits"** for further details). The Primary Oscillator Configuration bits, POSCMD<1:0> (Configuration Word 2<1:0>), and the Initial Oscillator Select Configuration bits, FNOSC<2:0> (Configuration Word 2<10:8>), select the oscillator source that is used at a Power-on Reset. The FRC Primary Oscillator with postscaler (FRCDIV) is the default (unprogrammed) selection. The Secondary Oscillator, or one of the internal oscillators, may be chosen by programming these bit locations.

The Configuration bits allow users to choose between the various clock modes, shown in Table 8-1.

8.2.1 CLOCK SWITCHING MODE CONFIGURATION BITS

The FCKSM Configuration bits (Configuration Word 2<7:6>) are used to jointly configure device clock switching and the Fail-Safe Clock Monitor (FSCM). Clock switching is enabled only when FCKSM1 is programmed ('0'). The FSCM is enabled only when the FCKSM<1:0> bits are both programmed ('00').

TABLE 8-1: CONFIGURATION BIT VALUES FOR CLOCK SELECTION

Oscillator Mode	Oscillator Source	POSCMD<1:0>	FNOSC<2:0>	Note
Fast RC Oscillator with Postscaler (FRCDIV)	Internal	11	111	**1, 2**
(Reserved)	Internal	xx	110	**1**
Low-Power RC Oscillator (LPRC)	Internal	11	101	**1**
Secondary (Timer1) Oscillator (SOSC)	Secondary	11	100	**1**
Primary Oscillator (XT) with PLL Module (XTPLL)	Primary	01	011	
Primary Oscillator (EC) with PLL Module (ECPLL)	Primary	00	011	
Primary Oscillator (HS)	Primary	10	010	
Primary Oscillator (XT)	Primary	01	010	
Primary Oscillator (EC)	Primary	00	010	
Fast RC Oscillator with PLL Module (FRCPLL)	Internal	11	001	**1**
Fast RC Oscillator (FRC)	Internal	11	000	**1**

Note 1: OSCO pin function is determined by the OSCIOFCN Configuration bit.
2: This is the default oscillator mode for an unprogrammed (erased) device.

8.3 Control Registers

The operation of the oscillator is controlled by three Special Function Registers:

- OSCCON
- CLKDIV
- OSCTUN

The OSCCON register (Register 8-1) is the main control register for the oscillator. It controls clock source switching and allows the monitoring of clock sources.

The CLKDIV register (Register 8-2) controls the features associated with Doze mode, as well as the postscaler for the FRC Oscillator.

The OSCTUN register (Register 8-3) allows the user to fine tune the FRC Oscillator over a range of approximately ±12%. Each bit increment or decrement changes the factory calibrated frequency of the FRC Oscillator by a fixed amount.

REGISTER 8-1: OSCCON: OSCILLATOR CONTROL REGISTER

U-0	R-0	R-0	R-0	U-0	R/W-x(1)	R/W-x(1)	R/W-x(1)
—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0
bit 15							bit 8

R/SO-0	R/W-0	R-0(3)	U-0	R/CO-0	R/W-0	R/W-0	R/W-0
CLKLOCK	IOLOCK(2)	LOCK	—	CF	POSCEN	SOSCEN	OSWEN
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **COSC<2:0>:** Current Oscillator Selection bits
- 111 = Fast RC Oscillator with Postscaler (FRCDIV)
- 110 = Reserved
- 101 = Low-Power RC Oscillator (LPRC)
- 100 = Secondary Oscillator (SOSC)
- 011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL)
- 010 = Primary Oscillator (XT, HS, EC)
- 001 = Fast RC Oscillator with Postscaler and PLL module (FRCPLL)
- 000 = Fast RC Oscillator (FRC)

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **NOSC<2:0>:** New Oscillator Selection bits(1)
- 111 = Fast RC Oscillator with Postscaler (FRCDIV)
- 110 = Reserved
- 101 = Low-Power RC Oscillator (LPRC)
- 100 = Secondary Oscillator (SOSC)
- 011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL)
- 010 = Primary Oscillator (XT, HS, EC)
- 001 = Fast RC Oscillator with Postscaler and PLL module (FRCPLL)
- 000 = Fast RC Oscillator (FRC)

Note 1: Reset values for these bits are determined by the FNOSC Configuration bits.
2: The state of the IOLOCK bit can only be changed once an unlocking sequence has been executed. In addition, if the IOL1WAY Configuration bit is '1', once the IOLOCK bit is set, it cannot be cleared.
3: Also resets to '0' during any valid clock switch or whenever a non-PLL clock mode is selected.

REGISTER 8-1: OSCCON: OSCILLATOR CONTROL REGISTER (CONTINUED)

bit 7	**CLKLOCK:** Clock Selection Lock Enabled bit If FSCM is enabled (FCKSM1 = `1`): `1` = Clock and PLL selections are locked `0` = Clock and PLL selections are not locked and may be modified by setting the OSWEN bit If FSCM is disabled (FCKSM1 = `0`): Clock and PLL selections are never locked and may be modified by setting the OSWEN bit.
bit 6	**IOLOCK:** I/O Lock Enable bit(2) `1` = I/O lock is active `0` = I/O lock is not active
bit 5	**LOCK:** PLL Lock Status bit(3) `1` = PLL module is in lock or PLL module start-up timer is satisfied `0` = PLL module is out of lock, PLL start-up timer is running or PLL is disabled
bit 4	**Unimplemented:** Read as '`0`'
bit 3	**CF:** Clock Fail Detect bit `1` = FSCM has detected a clock failure `0` = No clock failure has been detected
bit 2	**POSCEN:** Primary Oscillator Sleep Enable bit `1` = Primary Oscillator continues to operate during Sleep mode `0` = Primary Oscillator disabled during Sleep mode
bit 1	**SOSCEN:** 32 kHz Secondary Oscillator (SOSC) Enable bit `1` = Enable Secondary Oscillator `0` = Disable Secondary Oscillator
bit 0	**OSWEN:** Oscillator Switch Enable bit `1` = Initiate an oscillator switch to clock source specified by NOSC<2:0> bits `0` = Oscillator switch is complete

Note 1: Reset values for these bits are determined by the FNOSC Configuration bits.

2: The state of the IOLOCK bit can only be changed once an unlocking sequence has been executed. In addition, if the IOL1WAY Configuration bit is '`1`', once the IOLOCK bit is set, it cannot be cleared.

3: Also resets to '`0`' during any valid clock switch or whenever a non-PLL clock mode is selected.

REGISTER 8-2: CLKDIV: CLOCK DIVIDER REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
ROI	DOZE2	DOZE1	DOZE0	DOZEN(1)	RCDIV2	RCDIV1	RCDIV0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 15 **ROI:** Recover on Interrupt bit
- 1 = Interrupts clear the DOZEN bit and reset the CPU peripheral clock ratio to 1:1
- 0 = Interrupts have no effect on the DOZEN bit

bit 14-12 **DOZE<2:0>:** CPU Peripheral Clock Ratio Select bits
- 111 = 1:128
- 110 = 1:64
- 101 = 1:32
- 100 = 1:16
- 011 = 1:8
- 010 = 1:4
- 001 = 1:2
- 000 = 1:1

bit 11 **DOZEN:** DOZE Enable bit(1)
- 1 = DOZE<2:0> bits specify the CPU peripheral clock ratio
- 0 = CPU peripheral clock ratio set to 1:1

bit 10-8 **RCDIV<2:0>:** FRC Postscaler Select bits
- 111 = 31.25 kHz (divide-by-256)
- 110 = 125 kHz (divide-by-64)
- 101 = 250 kHz (divide-by-32)
- 100 = 500 kHz (divide-by-16)
- 011 = 1 MHz (divide-by-8)
- 010 = 2 MHz (divide-by-4)
- 001 = 4 MHz (divide-by-2)
- 000 = 8 MHz (divide-by-1)

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

Note 1: This bit is automatically cleared when the ROI bit is set and an interrupt occurs.

REGISTER 8-3: OSCTUN: FRC OSCILLATOR TUNE REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN5[(1)]	TUN4[(1)]	TUN3[(1)]	TUN2[(1)]	TUN1[(1)]	TUN0[(1)]
bit 7							bit 0

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

bit 15-6 **Unimplemented:** Read as ‘0’

bit 5-0 **TUN<5:0>:** FRC Oscillator Tuning bits[(1)]

011111 = Maximum frequency deviation
011110 =
•
•
•
000001 =
000000 = Center frequency, oscillator is running at factory calibrated frequency
111111 =
•
•
•
100001 =
100000 = Minimum frequency deviation

Note 1: Increments or decrements of TUN<5:0> may not change the FRC frequency in equal steps over the FRC tuning range and may not be monotonic.

8.4 Clock Switching Operation

With few limitations, applications are free to switch between any of the four clock sources (POSC, SOSC, FRC and LPRC) under software control and at any time. To limit the possible side effects that could result from this flexibility, PIC24F devices have a safeguard lock built into the switching process.

Note:	The Primary Oscillator mode has three different submodes (XT, HS and EC) which are determined by the POSCMDx Configuration bits. While an application can switch to and from Primary Oscillator mode in software, it cannot switch between the different primary submodes without reprogramming the device.

8.4.1 ENABLING CLOCK SWITCHING

To enable clock switching, the FCKSM Configuration bits in CW2 must be programmed to ‘00’. (Refer to **Section 25.1 “Configuration Bits”** for further details.) If the FCKSM Configuration bits are unprogrammed (‘1x’), the clock switching function and Fail-Safe Clock Monitor function are disabled. This is the default setting.

The NOSCx control bits (OSCCON<10:8>) do not control the clock selection when clock switching is disabled. However, the COSCx bits (OSCCON<14:12>) will reflect the clock source selected by the FNOSCx Configuration bits.

The OSWEN control bit (OSCCON<0>) has no effect when clock switching is disabled. It is held at ‘0’ at all times.

8.4.2 OSCILLATOR SWITCHING SEQUENCE

At a minimum, performing a clock switch requires this basic sequence:

1. If desired, read the COSCx bits (OSCCON<14:12>), to determine the current oscillator source.
2. Perform the unlock sequence to allow a write to the OSCCON register high byte.
3. Write the appropriate value to the NOSCx bits (OSCCON<10:8>) for the new oscillator source.
4. Perform the unlock sequence to allow a write to the OSCCON register low byte.
5. Set the OSWEN bit to initiate the oscillator switch.

Once the basic sequence is completed, the system clock hardware responds automatically as follows:

1. The clock switching hardware compares the COSCx bits with the new value of the NOSCx bits. If they are the same, then the clock switch is a redundant operation. In this case, the OSWEN bit is cleared automatically and the clock switch is aborted.
2. If a valid clock switch has been initiated, the LOCK (OSCCON<5>) and CF (OSCCON<3>) bits are cleared.
3. The new oscillator is turned on by the hardware if it is not currently running. If a crystal oscillator must be turned on, the hardware will wait until the OST expires. If the new source is using the PLL, then the hardware waits until a PLL lock is detected (LOCK = `1`).
4. The hardware waits for 10 clock cycles from the new clock source and then performs the clock switch.
5. The hardware clears the OSWEN bit to indicate a successful clock transition. In addition, the NOSCx bit values are transferred to the COSCx bits.
6. The old clock source is turned off at this time, with the exception of LPRC (if WDT or FSCM are enabled) or SOSC (if SOSCEN remains set).

> **Note 1:** The processor will continue to execute code throughout the clock switching sequence. Timing sensitive code should not be executed during this time.
>
> **2:** Direct clock switches between any Primary Oscillator mode with PLL and FRCPLL mode are not permitted. This applies to clock switches in either direction. In these instances, the application must switch to FRC mode as a transition clock source between the two PLL modes.

A recommended code sequence for a clock switch includes the following:

1. Disable interrupts during the OSCCON register unlock and write sequence.
2. Execute the unlock sequence for the OSCCON high byte by writing 78h and 9Ah to OSCCON<15:8> in two back-to-back instructions.
3. Write new oscillator source to the NOSCx bits in the instruction immediately following the unlock sequence.
4. Execute the unlock sequence for the OSCCON low byte by writing 46h and 57h to OSCCON<7:0> in two back-to-back instructions.
5. Set the OSWEN bit in the instruction immediately following the unlock sequence.
6. Continue to execute code that is not clock sensitive (optional).
7. Invoke an appropriate amount of software delay (cycle counting) to allow the selected oscillator and/or PLL to start and stabilize.
8. Check to see if OSWEN is '`0`'. If it is, the switch was successful. If OSWEN is still set, then check the LOCK bit to determine the cause of failure.

The core sequence for unlocking the OSCCON register and initiating a clock switch is shown in Example 8-1.

EXAMPLE 8-1: BASIC CODE SEQUENCE FOR CLOCK SWITCHING

```
;Place the new oscillator selection in W0
;OSCCONH (high byte) Unlock Sequence
MOV         #OSCCONH, w1
MOV         #0x78, w2
MOV         #0x9A, w3
MOV.b       w2, [w1]
MOV.b       w3, [w1]
;Set new oscillator selection
MOV.b       WREG, OSCCONH
;OSCCONL (low byte) unlock sequence
MOV         #OSCCONL, w1
MOV         #0x46, w2
MOV         #0x57, w3
MOV.b       w2, [w1]
MOV.b       w3, [w1]
;Start oscillator switch operation
BSET        OSCCON,#0
```

8.5 Secondary Oscillator (SOSC)

8.5.1 BASIC SOSC OPERATION

PIC24FJ64GA104 family devices do not have to set the SOSCEN bit to use the Secondary Oscillator. Any module requiring the SOSC (such as RTCC, Timer1 or DSWDT) will automatically turn on the SOSC when the clock signal is needed. The SOSC, however, has a long start-up time. To avoid delays for peripheral start-up, the SOSC can be manually started using the SOSCEN bit.

To use the Secondary Oscillator, the SOSCSEL<1:0> bits (CW3<9:8>) must be configured in an oscillator mode – either '11' or '01'. Setting SOSCSEL to '00' configures the SOSC pins for Digital mode, enabling digital I/O functionality on the pins. Digital functionality will not be available if the SOSC is configured in either of the oscillator modes.

8.5.2 LOW-POWER SOSC OPERATION

The Secondary Oscillator can operate in two distinct levels of power consumption based on device configuration. In Low-Power mode, the oscillator operates in a low drive strength, low-power state. By default, the oscillator uses a higher drive strength, and therefore, requires more power. The Secondary Oscillator Mode Configuration bits, SOSCSEL<1:0> (CW3<9:8>), determine the oscillator's power mode. Programming the SOSCSEL bits to '01' selects low-power operation.

The lower drive strength of this mode makes the SOSC more sensitive to noise and requires a longer start-up time. When Low-Power mode is used, care must be taken in the design and layout of the SOSC circuit to ensure that the oscillator starts up and oscillates properly.

8.5.3 EXTERNAL (DIGITAL) CLOCK MODE (SCLKI)

The SOSC can also be configured to run from an external 32 kHz clock source, rather than the internal oscillator. In this mode, also referred to as Digital mode, the clock source provided on the SCLKI pin is used to clock any modules that are configured to use the Secondary Oscillator. In this mode, the crystal driving circuit is disabled and the SOSCEN bit (OSCCON<1>) has no effect.

8.5.4 SOSC LAYOUT CONSIDERATIONS

The pinout limitations on low pin count devices, such as those in the PIC24FJ64GA104 family, may make the SOSC more susceptible to noise than other PIC24F devices. Unless proper care is taken in the design and layout of the SOSC circuit, this external noise may introduce inaccuracies into the oscillator's period.

In general, the crystal circuit connections should be as short as possible. It is also good practice to surround the crystal circuit with a ground loop or ground plane. For more information on crystal circuit design, please refer to **Section 6 "Oscillator"** (DS39700) of the "*PIC24F Family Reference Manual*". Additional information is also available in these Microchip Application Notes:

- *AN826, "Crystal Oscillator Basics and Crystal Selection for rfPIC® and PICmicro® Devices"* (DS00826)
- *AN849, "Basic PICmicro® Oscillator Design"* (DS00849).

8.6 Reference Clock Output

In addition to the CLKO output (FOSC/2) available in certain oscillator modes, the device clock in the PIC24FJ64GA104 family devices can also be configured to provide a reference clock output signal to a port pin. This feature is available in all oscillator configurations and allows the user to select a greater range of clock submultiples to drive external devices in the application.

This reference clock output is controlled by the REFOCON register (Register 8-4). Setting the ROEN bit (REFOCON<15>) makes the clock signal available on the REFO pin. The RODIV bits (REFOCON<11:8>) enable the selection of 16 different clock divider options.

The ROSSLP and ROSEL bits (REFOCON<13:12>) control the availability of the reference output during Sleep mode. The ROSEL bit determines if the oscillator on OSC1 and OSC2, or the current system clock source, is used for the reference clock output. The ROSSLP bit determines if the reference source is available on REFO when the device is in Sleep mode.

To use the reference clock output in Sleep mode, both the ROSSLP and ROSEL bits must be set. The device clock must also be configured for one of the primary modes (EC, HS or XT); otherwise, if the POSCEN bit is not also set, the oscillator on OSC1 and OSC2 will be powered down when the device enters Sleep mode. Clearing the ROSEL bit allows the reference output frequency to change as the system clock changes during any clock switches.

REGISTER 8-4: REFOCON: REFERENCE OSCILLATOR CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROEN	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 15 **ROEN:** Reference Oscillator Output Enable bit
1 = Reference oscillator is enabled on REFO pin
0 = Reference oscillator is disabled

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **ROSSLP:** Reference Oscillator Output Stop in Sleep bit
1 = Reference oscillator continues to run in Sleep
0 = Reference oscillator is disabled in Sleep

bit 12 **ROSEL:** Reference Oscillator Source Select bit
1 = Primary Oscillator is used as the base clock. Note that the crystal oscillator must be enabled using the FOSC<2:0> bits; the crystal maintains the operation in Sleep mode.
0 = System clock is used as the base clock; base clock reflects any clock switching of the device

bit 11-8 **RODIV<3:0>:** Reference Oscillator Divisor Select bits
1111 = Base clock value divided by 32,768
1110 = Base clock value divided by 16,384
1101 = Base clock value divided by 8,192
1100 = Base clock value divided by 4,096
1011 = Base clock value divided by 2,048
1010 = Base clock value divided by 1,024
1001 = Base clock value divided by 512
1000 = Base clock value divided by 256
0111 = Base clock value divided by 128
0110 = Base clock value divided by 64
0101 = Base clock value divided by 32
0100 = Base clock value divided by 16
0011 = Base clock value divided by 8
0010 = Base clock value divided by 4
0001 = Base clock value divided by 2
0000 = Base clock value

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

NOTES:

9.0 POWER-SAVING FEATURES

> **Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 39. "Power-Saving Features with Deep Sleep"** (DS39727).

The PIC24FJ64GA104 family of devices provides the ability to manage power consumption by selectively managing clocking to the CPU and the peripherals. In general, a lower clock frequency and a reduction in the number of circuits being clocked constitutes lower consumed power. All PIC24F devices manage power consumption in four different ways:

- Clock Frequency
- Instruction-Based Sleep, Idle and Deep Sleep modes
- Software Controlled Doze mode
- Selective Peripheral Control in Software

Combinations of these methods can be used to selectively tailor an application's power consumption, while still maintaining critical application features, such as timing-sensitive communications.

9.1 Clock Frequency and Clock Switching

PIC24F devices allow for a wide range of clock frequencies to be selected under application control. If the system clock configuration is not locked, users can choose low-power or high-precision oscillators by simply changing the NOSC bits. The process of changing a system clock during operation, as well as limitations to the process, are discussed in more detail in **Section 8.0 "Oscillator Configuration"**.

9.2 Instruction-Based Power-Saving Modes

PIC24F devices have two special power-saving modes that are entered through the execution of a special PWRSAV instruction. Sleep mode stops clock operation and halts all code execution; Idle mode halts the CPU and code execution, but allows peripheral modules to continue operation. Deep Sleep mode stops clock operation, code execution and all peripherals except RTCC and DSWDT. It also freezes I/O states and removes power to SRAM and Flash memory.

The assembly syntax of the PWRSAV instruction is shown in Example 9-1.

> **Note:** SLEEP_MODE and IDLE_MODE are constants defined in the assembler include file for the selected device.

Sleep and Idle modes can be exited as a result of an enabled interrupt, WDT time-out or a device Reset. When the device exits these modes, it is said to "wake-up".

9.2.1 SLEEP MODE

Sleep mode has these features:

- The system clock source is shut down. If an on-chip oscillator is used, it is turned off.
- The device current consumption will be reduced to a minimum provided that no I/O pin is sourcing current.
- The I/O pin directions and states are frozen.
- The Fail-Safe Clock Monitor does not operate during Sleep mode since the system clock source is disabled.
- The LPRC clock will continue to run in Sleep mode if the WDT or RTCC with LPRC as clock source is enabled.
- The WDT, if enabled, is automatically cleared prior to entering Sleep mode.
- Some device features or peripherals may continue to operate in Sleep mode. This includes items, such as the input change notification on the I/O ports, or peripherals that use an external clock input. Any peripheral that requires the system clock source for its operation will be disabled in Sleep mode.

The device will wake-up from Sleep mode on any of these events:

- On any interrupt source that is individually enabled
- On any form of device Reset
- On a WDT time-out

On wake-up from Sleep, the processor will restart with the same clock source that was active when Sleep mode was entered.

EXAMPLE 9-1: PWRSAV INSTRUCTION SYNTAX

```
PWRSAV      #SLEEP_MODE    ; Put the device into SLEEP mode
PWRSAV      #IDLE_MODE     ; Put the device into IDLE mode
BSET        DSCON, #DSEN   ; Enable Deep Sleep
PWRSAV      #SLEEP_MODE    ; Put the device into Deep SLEEP mode
```

9.2.2 IDLE MODE

Idle mode has these features:

- The CPU will stop executing instructions.
- The WDT is automatically cleared.
- The system clock source remains active. By default, all peripheral modules continue to operate normally from the system clock source, but can also be selectively disabled (see **Section 9.4 "Selective Peripheral Module Control"**).
- If the WDT or FSCM is enabled, the LPRC will also remain active.

The device will wake from Idle mode on any of these events:

- Any interrupt that is individually enabled
- Any device Reset
- A WDT time-out

On wake-up from Idle, the clock is reapplied to the CPU and instruction execution begins immediately, starting with the instruction following the PWRSAV instruction or the first instruction in the ISR.

9.2.3 INTERRUPTS COINCIDENT WITH POWER SAVE INSTRUCTIONS

Any interrupt that coincides with the execution of a PWRSAV instruction (except for Deep Sleep) will be held off until entry into Sleep or Idle mode has completed. The device will then wake-up from Sleep or Idle mode.

9.2.4 DEEP SLEEP MODE

In PIC24FJ64GA104 family devices, Deep Sleep mode is intended to provide the lowest levels of power consumption available, without requiring the use of external switches to completely remove all power from the device. Entry into Deep Sleep mode is completely under software control. Exit from Deep Sleep mode can be triggered from any of the following events:

- POR event
- $\overline{MCLR}$ event
- RTCC alarm (If the RTCC is present)
- External Interrupt 0
- Deep Sleep Watchdog Timer (DSWDT) time-out

In Deep Sleep mode, it is possible to keep the device Real-Time Clock and Calendar (RTCC) running without the loss of clock cycles.

The device has a dedicated Deep Sleep Brown-out Reset (DSBOR) and a Deep Sleep Watchdog Timer Reset (DSWDT) for monitoring voltage and time-out events. The DSBOR and DSWDT are independent of the standard BOR and WDT used with other power-managed modes (Sleep, Idle and Doze).

Note:	Since Deep Sleep mode powers down the microcontroller by turning off the on-chip VDDCORE voltage regulator, Deep Sleep capability is available only when operating with the internal regulator enabled.

9.2.4.1 Entering Deep Sleep Mode

Deep Sleep mode is entered by setting the DSEN bit in the DSCON register, and then executing a SLEEP instruction (PWRSAV #SLEEP_MODE) within one to three instruction cycles to minimize the chance that Deep Sleep will be spuriously entered.

If the PWRSAV command is not given within three instruction cycles, the DSEN bit will be cleared by the hardware and must be set again by the software before entering Deep Sleep mode. The DSEN bit is also automatically cleared when exiting the Deep Sleep mode.

Note:	To re-enter Deep Sleep after a Deep Sleep wake-up, allow a delay of at least 3 TCY after clearing the RELEASE bit.

The sequence to enter Deep Sleep mode is:

1. If the application requires the Deep Sleep WDT, enable it and configure its clock source (see **Section 9.2.4.7 "Deep Sleep WDT"** for details).
2. If the application requires Deep Sleep BOR, enable it by programming the DSBOREN Configuration bit (CW4<6>).
3. If the application requires wake-up from Deep Sleep on RTCC alarm, enable and configure the RTCC module (see **Section 19.0 "Real-Time Clock and Calendar (RTCC)"** for more information).
4. If needed, save any critical application context data by writing it to the DSGPR0 and DSGPR1 registers (optional).
5. Enable Deep Sleep mode by setting the DSEN bit (DSCON<15>).
6. Enter Deep Sleep mode by immediately issuing a PWRSAV #0 instruction.

Any time the DSEN bit is set, all bits in the DSWAKE register will be automatically cleared.

9.2.4.2 Special Cases when Entering Deep Sleep Mode

When entering Deep Sleep mode, there are certain circumstances that require a delay between setting the DSEN bit and executing the PWRSAV instruction. These can be generally reduced to three scenarios:

1. Scenario (1): use an external wake-up source (INT0) or the RTCC is used
2. Scenario (2): with application-level interrupts that can be temporarily disabled
3. Scenario (3): with interrupts that must be monitored

In the first scenario, the application requires a wake-up from Deep Sleep on the assertion of the INT0 pin or the RTCC interrupt. In this case, three NOP instructions must be inserted to properly synchronize the detection of an asynchronous INT0 interrupt after the device enters Deep Sleep mode. If the application does not use wake-up on INT0 or RTCC, the NOP instructions are optional.

In the second scenario, the application also uses interrupts which can be briefly ignored. With these applications, an interrupt event during the execution of the NOP instructions may cause an ISR to be executed. This means that more than three instruction cycles will elapse before returning to the code and that the DSEN bit will be cleared. To prevent the missed entry into Deep Sleep, temporarily disable interrupts prior to entering Deep Sleep mode. Invoking the DISI instruction for four cycles is sufficient to prevent interrupts from disrupting Deep Sleep entry.

In the third scenario, interrupts cannot be ignored even briefly; constant interrupt detection is required, even during the interval between setting DSEN and executing the PWRSAV instruction. For these cases, it is possible to disable interrupts and test for an interrupt condition, skipping the PWRSAV instruction if necessary. Testing for interrupts can be accomplished by checking the status of the CPUIRQ bit (INTTREG<15>). If an unserviced interrupt is pending, this bit will be set. If CPUIRQ is set prior to executing the PWRSAV instruction, the instruction is skipped. At this point, the DISI instruction has expired (being more than 4 instructions from when it was executed) and the application vectors to the appropriate ISR. When the application returns, it can either attempt to re-enter Deep Sleep mode or perform some other system function. In either case, the application must have some functional code located, following the PWRSAV instruction, in the event that the PWRSAV instruction is skipped and the device does not enter Deep Sleep mode.

Examples for implementing these cases are shown in Example 9-2. It is recommended that an assembler, or in-line C routine be used in these cases, to ensure that the code executes in the number of cycles required.

EXAMPLE 9-2: IMPLEMENTING THE SPECIAL CASES FOR ENTERING DEEP SLEEP

```
// Case 1: simplest delay scenario
//
asm("bset  DSCON, #15");
asm("nop");
asm("nop");
asm("nop");
asm("pwrsav #0");
//
// Case 2: interrupts disabled
//
asm("disi #4");
asm("bset  DSCON, #15");
asm("nop");
asm("nop");
asm("nop");
asm("pwrsav #0");
//
// Case 3: interrupts disabled with
// interrupt testing
//
asm("disi #4");
asm("bset  DSCON, #15");
asm("nop");
asm("nop");
asm("btss  INTTREG, #15");
asm("pwrsav #0");
// continue with application code here
//
```

9.2.4.3 Exiting Deep Sleep Mode

Deep Sleep mode exits on any one of the following events:

- POR event on VDD supply. If there is no DSBOR circuit to re-arm the VDD supply POR circuit, the external VDD supply must be lowered to the natural arming voltage of the POR circuit.
- DSWDT time-out. When the DSWDT timer times out, the device exits Deep Sleep.
- RTCC alarm (if RTCEN = 1).
- Assertion ('0') of the $\overline{\text{MCLR}}$ pin.
- Assertion of the INT0 pin (if the interrupt was enabled before Deep Sleep mode was entered). The polarity configuration is used to determine the assertion level ('0' or '1') of the pin that will cause an exit from Deep Sleep mode. Exiting from Deep Sleep mode requires a change on the INT0 pin while in Deep Sleep mode.

Note:	Any interrupt pending when entering Deep Sleep mode is cleared.

Exiting Deep Sleep mode generally does not retain the state of the device and is equivalent to a Power-on Reset (POR) of the device. Exceptions to this include the RTCC (if present), which remains operational through the wake-up, the DSGPRx registers and the DSWDT bit.

Wake-up events that occur from the time Deep Sleep exits, until the time that the POR sequence completes, are ignored, and are not captured in the DSWAKE register.

The sequence for exiting Deep Sleep mode is:

1. After a wake-up event, the device exits Deep Sleep and performs a POR. The DSEN bit is cleared automatically. Code execution resumes at the Reset vector.
2. To determine if the device exited Deep Sleep, read the Deep Sleep bit, DPSLP (RCON<10>). This bit will be set if there was an exit from Deep Sleep mode. If the bit is set, clear it.
3. Determine the wake-up source by reading the DSWAKE register.
4. Determine if a DSBOR event occurred during Deep Sleep mode by reading the DSBOR bit (DSCON<1>).
5. If application context data has been saved, read it back from the DSGPR0 and DSGPR1 registers.
6. Clear the RELEASE bit (DSCON<0>).

9.2.4.4 Deep Sleep Wake-up Time

Since wake-up from Deep Sleep results in a POR, the wake-up time from Deep Sleep is the same as the device POR time. Also, because the internal regulator is turned off, the voltage on VCAP may drop depending on how long the device is asleep. If VCAP has dropped below 2V, then there will be additional wake-up time while the regulator charges VCAP.

Deep Sleep wake-up time is specified in **Section 28.0 "Electrical Characteristics"** as TDSWU. This specification indicates the worst-case wake-up time, including the full POR Reset time (including TPOR and TRST), as well as the time to fully charge a 10 μF capacitor on VCAP which has discharged to 0V. Wake-up may be significantly faster if VCAP has not discharged.

9.2.4.5 Saving Context Data with the DSGPR0/DSGPR1 Registers

As exiting Deep Sleep mode causes a POR, most Special Function Registers reset to their default POR values. In addition, because VDDCORE power is not supplied in Deep Sleep mode, information in data RAM may be lost when exiting this mode.

Applications which require critical data to be saved prior to Deep Sleep may use the Deep Sleep General Purpose registers, DSGPR0 and DSGPR1, or data EEPROM (if available). Unlike other SFRs, the contents of these registers are preserved while the device is in Deep Sleep mode. After exiting Deep Sleep, software can restore the data by reading the registers and clearing the RELEASE bit (DSCON<0>).

9.2.4.6 I/O Pins During Deep Sleep

During Deep Sleep, the general purpose I/O pins retain their previous states and the Secondary Oscillator (SOSC) will remain running, if enabled. Pins that are configured as inputs (TRIS bit is set) prior to entry into Deep Sleep remain high-impedance during Deep Sleep. Pins that are configured as outputs (TRIS bit is clear) prior to entry into Deep Sleep remain as output pins during Deep Sleep. While in this mode, they continue to drive the output level determined by their corresponding LAT bit at the time of entry into Deep Sleep.

Once the device wakes back up, all I/O pins continue to maintain their previous states, even after the device has finished the POR sequence and is executing application code again. Pins configured as inputs during Deep Sleep remain high-impedance and pins configured as outputs continue to drive their previous value. After waking up, the TRIS and LAT registers, and the SOSCEN bit (OSCCON<1>) are reset. If firmware modifies any of these bits or registers, the I/O will not immediately go to the newly configured states. Once the firmware clears the RELEASE bit (DSCON<0>) the I/O pins are "released". This causes the I/O pins to take the states configured by their respective TRIS and LAT bit values.

This means that keeping the SOSC running after waking up requires the SOSCEN bit to be set before clearing RELEASE.

If the Deep Sleep BOR (DSBOR) is enabled, and a DSBOR or a true POR event occurs during Deep Sleep, the I/O pins will be immediately released similar to clearing the RELEASE bit. All previous state information will be lost, including the general purpose DSGPR0 and DSGPR1 contents.

If a $\overline{\text{MCLR}}$ Reset event occurs during Deep Sleep, the DSGPRx, DSCON and DSWAKE registers will remain valid and the RELEASE bit will remain set. The state of the SOSC will also be retained. The I/O pins, however, will be reset to their $\overline{\text{MCLR}}$ Reset state. Since RELEASE is still set, changes to the SOSCEN bit (OSCCON<1>) cannot take effect until the RELEASE bit is cleared.

In all other Deep Sleep wake-up cases, application firmware must clear the RELEASE bit in order to reconfigure the I/O pins.

9.2.4.7 Deep Sleep WDT

To enable the DSWDT in Deep Sleep mode, program the Configuration bit, DSWDTEN (CW4<7>). The device Watchdog Timer (WDT) need not be enabled for the DSWDT to function. Entry into Deep Sleep mode automatically resets the DSWDT.

The DSWDT clock source is selected by the DSWDTOSC Configuration bit (CW4<4>). The postscaler options are programmed by the DSWDTPS<3:0> Configuration bits (CW4<3:0>). The minimum time-out period that can be achieved is 2.1 ms and the maximum is 25.7 days. For more details on the CW4 Configuration register and DSWDT configuration options, refer to **Section 25.0 "Special Features"**.

9.2.4.8 Switching Clocks in Deep Sleep Mode

Both the RTCC and the DSWDT may run from either SOSC or the LPRC clock source. This allows both the RTCC and DSWDT to run without requiring both the LPRC and SOSC to be enabled together, reducing power consumption.

Running the RTCC from LPRC will result in a loss of accuracy in the RTCC of approximately 5 to 10%. If an accurate RTCC is required, it must be run from the SOSC clock source. The RTCC clock source is selected with the RTCOSC Configuration bit (CW4<5>).

Under certain circumstances, it is possible for the DSWDT clock source to be off when entering Deep Sleep mode. In this case, the clock source is turned on automatically (if DSWDT is enabled), without the need for software intervention. However, this can cause a delay in the start of the DSWDT counters. In order to avoid this delay when using SOSC as a clock source, the application can activate SOSC prior to entering Deep Sleep mode.

9.2.4.9 Checking and Clearing the Status of Deep Sleep

Upon entry into Deep Sleep mode, the status bit, DPSLP (RCON<10>), becomes set and must be cleared by software.

On power-up, the software should read this status bit to determine if the Reset was due to an exit from Deep Sleep mode and clear the bit if it is set. Of the four possible combinations of DPSLP and POR bit states, three cases can be considered:

- Both the DPSLP and POR bits are cleared. In this case, the Reset was due to some event other than a Deep Sleep mode exit.
- The DPSLP bit is clear, but the POR bit is set. This is a normal Power-on Reset.
- Both the DPSLP and POR bits are set. This means that Deep Sleep mode was entered, the device was powered down and Deep Sleep mode was exited.

9.2.4.10 Power-on Resets (PORs)

VDD voltage is monitored to produce PORs. Since exiting from Deep Sleep functionally looks like a POR, the technique described in **Section 9.2.4.9 “Checking and Clearing the Status of Deep Sleep”** should be used to distinguish between Deep Sleep and a true POR event.

When a true POR occurs, the entire device, including all Deep Sleep logic (Deep Sleep registers, RTCC, DSWDT, etc.) is reset.

9.2.4.11 Summary of Deep Sleep Sequence

To review, these are the necessary steps involved in invoking and exiting Deep Sleep mode:

1. Device exits Reset and begins to execute its application code.
2. If DSWDT functionality is required, program the appropriate Configuration bit.
3. Select the appropriate clock(s) for the DSWDT and RTCC (optional).
4. Enable and configure the RTCC (optional).
5. Write context data to the DSGPRx registers (optional).
6. Enable the INT0 interrupt (optional).
7. Set the DSEN bit in the DSCON register.
8. Enter Deep Sleep by issuing a `PWRSV #SLEEP_MODE` command.
9. Device exits Deep Sleep when a wake-up event occurs.
10. The DSEN bit is automatically cleared.
11. Read and clear the DPSLP status bit in RCON, and the DSWAKE status bits.
12. Read the DSGPRx registers (optional).
13. Once all state related configurations are complete, clear the RELEASE bit.
14. Application resumes normal operation.

REGISTER 9-1: DSCON: DEEP SLEEP CONTROL REGISTER

R/W-0, HC	U-0	U-0	U-0	U-0	U-0	U-0	U-0
DSEN[(1)]	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0, HCS	R/C-0, HS
—	—	—	—	—	—	DSBOR[(1,2,3)]	RELEASE[(1,2)]
bit 7							bit 0

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

bit 15 **DSEN:** Deep Sleep Enable bit[(1)]
`1` = Device enters Deep Sleep when `PWRSAV #0` is executed in the next instruction
`0` = Device enters normal Sleep when `PWRSAV #0` is executed

bit 14-2 **Unimplemented:** Read as '`0`'

bit 1 **DSBOR:** Deep Sleep BOR Event Status bit[(1,2,3)]
`1` = The DSBOR was active and a BOR event was detected during Deep Sleep
`0` = The DSBOR was disabled or was active and did not detect a BOR event during Deep Sleep

bit 0 **RELEASE:** I/O Pin State Deep Sleep Release bit[(1,2)]
`1` = I/O pins and SOSC maintain their states following exit from Deep Sleep, regardless of their LAT and TRIS configuration
`0` = I/O pins and SOSC are released from their Deep Sleep states. The pin state is controlled by the LAT and TRIS configurations, and the SOSCEN bit.

Note 1: These bits are reset only in the case of a POR event outside of Deep Sleep mode.
2: Reset value is '`0`' for initial power-on POR only and '`1`' for Deep Sleep POR.
3: This is a status bit only; a DSBOR event will NOT cause a wake-up from Deep Sleep.

REGISTER 9-2: DSWAKE: DEEP SLEEP WAKE-UP SOURCE REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0, HS
—	—	—	—	—	—	—	DSINT0(1)
bit 15							bit 8

R/W-0, HS	U-0	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	U-0	R/W-0, HS
DSFLT(1)	—	—	DSWDT(1)	DSRTC(1)	DSMCLR(1)	—	DSPOR(2)
bit 7							bit 0

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

bit 15-9 **Unimplemented:** Read as ‘0’

bit 8 **DSINT0:** Interrupt-on-Change bit(1)
1 = External Interrupt 0 was asserted during Deep Sleep
0 = External Interrupt 0 was not asserted during Deep Sleep

bit 7 **DSFLT:** Deep Sleep Fault Detected bit(1)
1 = A Fault occurred during Deep Sleep and some Deep Sleep configuration settings may have been corrupted
0 = No Fault was detected during Deep Sleep

bit 6-5 **Unimplemented:** Read as ‘0’

bit 4 **DSWDT:** Deep Sleep Watchdog Timer Time-out bit(1)
1 = The Deep Sleep Watchdog Timer timed out during Deep Sleep
0 = The Deep Sleep Watchdog Timer did not time out during Deep Sleep

bit 3 **DSRTC:** Real-Time Clock and Calendar Alarm bit(1)
1 = The Real-Time Clock and Calendar triggered an alarm during Deep Sleep
0 = The Real-Time Clock and Calendar did not trigger an alarm during Deep Sleep

bit 2 **DSMCLR:** Deep Sleep $\overline{\text{MCLR}}$ Event bit(1)
1 = The $\overline{\text{MCLR}}$ pin was asserted during Deep Sleep
0 = The $\overline{\text{MCLR}}$ pin was not asserted during Deep Sleep

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **DSPOR:** Power-on Reset Event bit(2)
1 = The VDD supply POR circuit was active and a POR event was detected
0 = The VDD supply POR circuit was not active, or was active, but did not detect a POR event

Note 1: This bit can only be set while the device is in Deep Sleep mode.
2: This bit can be set outside of Deep Sleep.

9.3 Doze Mode

Generally, changing clock speed and invoking one of the power-saving modes are the preferred strategies for reducing power consumption. There may be circumstances, however, where this is not practical. For example, it may be necessary for an application to maintain uninterrupted synchronous communication, even while it is doing nothing else. Reducing system clock speed may introduce communication errors, while using a power-saving mode may stop communications completely.

Doze mode is a simple and effective alternative method to reduce power consumption while the device is still executing code. In this mode, the system clock continues to operate from the same source and at the same speed. Peripheral modules continue to be clocked at the same speed while the CPU clock speed is reduced. Synchronization between the two clock domains is maintained, allowing the peripherals to access the SFRs while the CPU executes code at a slower rate.

Doze mode is enabled by setting the DOZEN bit (CLKDIV<11>). The ratio between peripheral and core clock speed is determined by the DOZE<2:0> bits (CLKDIV<14:12>). There are eight possible configurations, from 1:1 to 1:128, with 1:1 being the default.

It is also possible to use Doze mode to selectively reduce power consumption in event driven applications. This allows clock-sensitive functions, such as synchronous communications, to continue without interruption while the CPU Idles, waiting for something to invoke an interrupt routine. Enabling the automatic return to full-speed CPU operation on interrupts is enabled by setting the ROI bit (CLKDIV<15>). By default, interrupt events have no effect on Doze mode operation.

9.4 Selective Peripheral Module Control

Idle and Doze modes allow users to substantially reduce power consumption by slowing or stopping the CPU clock. Even so, peripheral modules still remain clocked, and thus, consume power. There may be cases where the application needs what these modes do not provide: the allocation of power resources to CPU processing with minimal power consumption from the peripherals.

PIC24F devices address this requirement by allowing peripheral modules to be selectively disabled, reducing or eliminating their power consumption. This can be done with two control bits:

- The Peripheral Enable bit, generically named "XXXEN", located in the module's main control SFR.
- The Peripheral Module Disable (PMD) bit, generically named "XXXMD", located in one of the PMD Control registers.

Both bits have similar functions in enabling or disabling its associated module. Setting the PMD bit for a module disables all clock sources to that module, reducing its power consumption to an absolute minimum. In this state, the control and status registers associated with the peripheral will also be disabled, so writes to those registers will have no effect and read values will be invalid. Many peripheral modules have a corresponding PMD bit.

In contrast, disabling a module by clearing its XXXEN bit disables its functionality, but leaves its registers available to be read and written to. This reduces power consumption, but not by as much as setting the PMD bit does. Most peripheral modules have an enable bit; exceptions include input capture, output compare and RTCC.

To achieve more selective power savings, peripheral modules can also be selectively disabled when the device enters Idle mode. This is done through the control bit of the generic name format, "XXXIDL". By default, all modules that can operate during Idle mode will do so. Using the disable on Idle feature allows further reduction of power consumption during Idle mode, enhancing power savings for extremely critical power applications.

NOTES:

10.0 I/O PORTS

> **Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 12. "I/O Ports with Peripheral Pin Select (PPS)"** (DS39711).

All of the device pins (except VDD, VSS, MCLR and OSCI/CLKI) are shared between the peripherals and the parallel I/O ports. All I/O input ports feature Schmitt Trigger inputs for improved noise immunity.

10.1 Parallel I/O (PIO) Ports

A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral. The peripheral's output buffer data and control signals are provided to a pair of multiplexers. The multiplexers select whether the peripheral or the associated port has ownership of the output data and control signals of the I/O pin. The logic also prevents "loop through", in which a port's digital output can drive the input of a peripheral that shares the same pin. Figure 10-1 shows how ports are shared with other peripherals and the associated I/O pin to which they are connected.

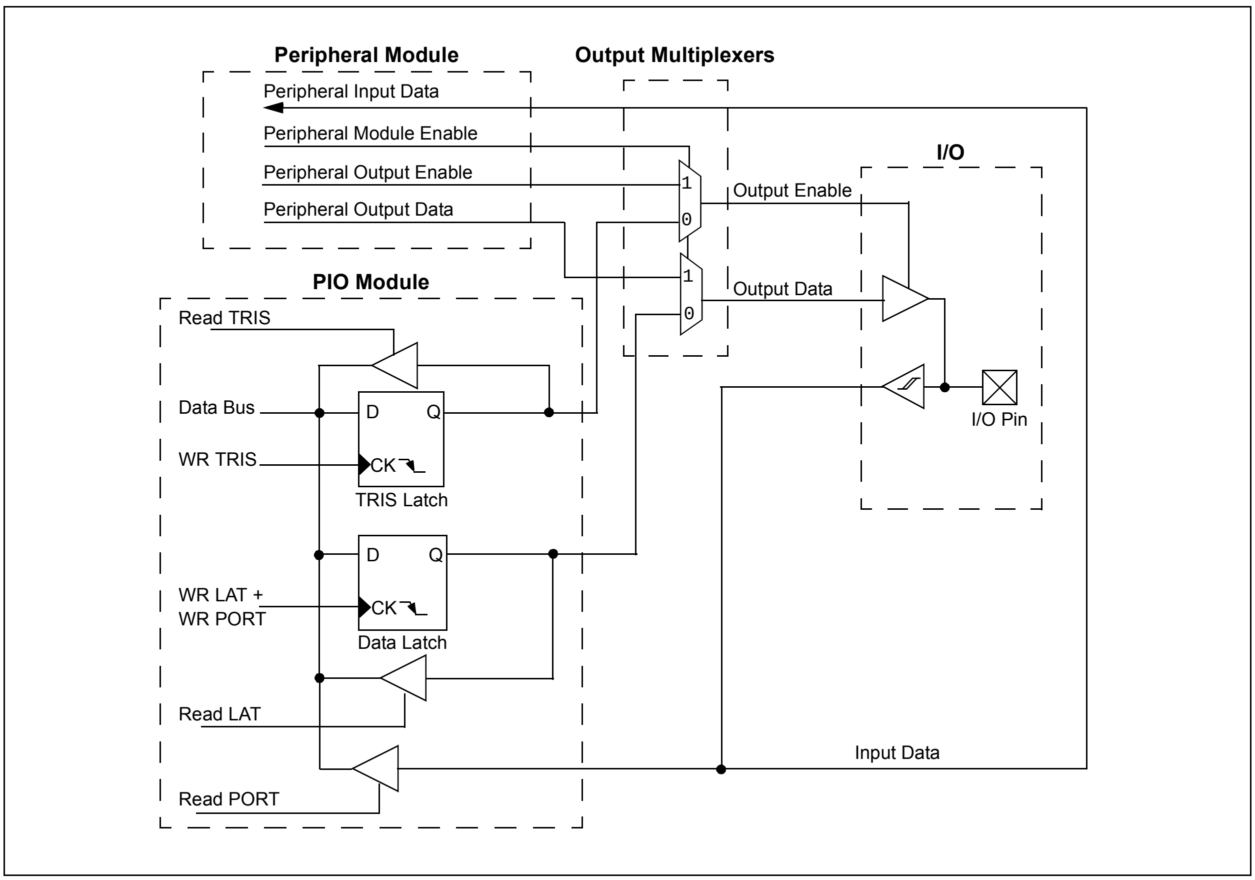
When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled. The I/O pin may be read, but the output driver for the parallel port bit will be disabled. If a peripheral is enabled, but the peripheral is not actively driving a pin, that pin may be driven by a port.

All port pins have three registers directly associated with their operation as digital I/Os. The Data Direction register (TRIS) determines whether the pin is an input or an output. If the data direction bit is a '1', then the pin is an input. All port pins are defined as inputs after a Reset. Reads from the Output Latch register (LAT), read the latch. Writes to the Output Latch register, write the latch. Reads from the port (PORT), read the port pins, while writes to the port pins, write the latch.

Any bit and its associated data and control registers that are not valid for a particular device will be disabled. That means the corresponding LAT and TRIS registers, and the port pin will read as zeros.

When a pin is shared with another peripheral or function that is defined as an input only, it is regarded as a dedicated port because there is no other competing source of outputs.

FIGURE 10-1: BLOCK DIAGRAM OF A TYPICAL SHARED PORT STRUCTURE



10.1.1 OPEN-DRAIN CONFIGURATION

In addition to the PORT, LAT and TRIS registers for data control, each port pin can also be individually configured for either digital or open-drain output. This is controlled by the Open-Drain Control register, ODCx, associated with each port. Setting any of the bits configures the corresponding pin to act as an open-drain output.

The open-drain feature allows the generation of outputs higher than VDD (e.g., 5V) on any desired digital only pins by using external pull-up resistors. The maximum open-drain voltage allowed is the same as the maximum VIH specification.

10.2 Configuring Analog Port Pins

The AD1PCFGL and TRIS registers control the operation of the A/D port pins. Setting a port pin as an analog input also requires that the corresponding TRIS bit be set. If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

When reading the PORT register, all pins configured as analog input channels will read as cleared (a low level).

Pins configured as digital inputs will not convert an analog input. Analog levels on any pin that is defined as a digital input (including the ANx pins) may cause the input buffer to consume current that exceeds the device specifications.

10.2.1 I/O PORT WRITE/READ TIMING

One instruction cycle is required between a port direction change or port write operation and a read operation of the same port. Typically, this instruction would be a NOP (Example 10-1).

10.2.2 ANALOG INPUT PINS AND VOLTAGE CONSIDERATIONS

The voltage tolerance of pins used as device inputs is dependent on the pin's input function. Pins that are used as digital only inputs are able to handle DC voltages up to 5.5V, a level typical for digital logic circuits. In contrast, pins that also have analog input functions of any kind can only tolerate voltages up to VDD. Voltage excursions beyond VDD on these pins should be avoided.

Table 10-1 summarizes the input voltage capabilities. Refer to **Section 28.0 "Electrical Characteristics"** for more details.

TABLE 10-1: INPUT VOLTAGE TOLERANCE

Port or Pin	Tolerated Input	Description
PORTA<4:0>	VDD	Only VDD input levels tolerated.
PORTB<15:12>		
PORTB<4:0>		
PORTC<3:0>(1)		
PORTA<10:7>(1)	5.5V	Tolerates input levels above VDD, useful for most standard logic.
PORTB<11:7>		
PORTB<6:5>		
PORTC<9:4>(1)		

Note 1: Not available on 28-pin devices.

EXAMPLE 10-1: PORT WRITE/READ EXAMPLE

```
MOV    0xFF00, W0          ; Configure PORTB<15:8> as inputs
MOV    W0, TRISB           ; and PORTB<7:0> as outputs
NOP                        ; Delay 1 cycle
BTSS   PORTB, #13          ; Next Instruction
```


10.3 Input Change Notification

The input change notification function of the I/O ports allows the PIC24FJ64GA104 family of devices to generate interrupt requests to the processor in response to a Change-of-State (COS) on selected input pins. This feature is capable of detecting input Change-of-States even in Sleep mode, when the clocks are disabled. Depending on the device pin count, there are up to 31 external inputs that may be selected (enabled) for generating an interrupt request on a Change-of-State.

Registers, CNEN1 and CNEN2, contain the interrupt enable control bits for each of the CN input pins. Setting any of these bits enables a CN interrupt for the corresponding pins.

Each CN pin has a weak pull-up connected to it. The pull-up acts as a current source that is connected to the pin. This eliminates the need for external resistors when push button or keypad devices are connected. The pull-ups are separately enabled using the CNPU1 and CNPU2 registers (for pull-ups). Each CN pin has individual control bits for its pull-up. Setting a control bit enables the weak pull-up for the corresponding pin.

When the internal pull-up is selected, the pin pulls up to V_{DD} – 0.7V (typical). Make sure that there is no external pull-up source when the internal pull-ups are enabled, as the voltage difference can cause a current path.

Note:	Pull-ups on change notification pins should always be disabled whenever the port pin is configured as a digital output.

10.4 Peripheral Pin Select (PPS)

A major challenge in general purpose devices is providing the largest possible set of peripheral features while minimizing the conflict of features on I/O pins. In an application that needs to use more than one peripheral multiplexed on a single pin, inconvenient work arounds in application code or a complete redesign may be the only option.

The Peripheral Pin Select feature provides an alternative to these choices by enabling the user's peripheral set selection and their placement on a wide range of I/O pins. By increasing the pinout options available on a particular device, users can better tailor the microcontroller to their entire application, rather than trimming the application to fit the device.

The Peripheral Pin Select feature operates over a fixed subset of digital I/O pins. Users may independently map the input and/or output of any one of many digital peripherals to any one of these I/O pins. Peripheral Pin Select is performed in software and generally does not require the device to be reprogrammed. Hardware safeguards are included that prevent accidental or spurious changes to the peripheral mapping once it has been established.

10.4.1 AVAILABLE PINS

The Peripheral Pin Select feature is used with a range of up to 25 pins, depending on the particular device and its pin count. Pins that support the Peripheral Pin Select feature include the designation "RPn" in their full pin designation, where "n" is the remappable pin number.

See Table 1-2 for a summary of pinout options in each package offering.

10.4.2 AVAILABLE PERIPHERALS

The peripherals managed by the Peripheral Pin Select are all digital only peripherals. These include general serial communications (UART and SPI), general purpose timer clock inputs, timer related peripherals (input capture and output compare) and external interrupt inputs. Also included are the outputs of the comparator module, since these are discrete digital signals.

Peripheral Pin Select is not available for I^2C™ change notification inputs, RTCC alarm outputs or peripherals with analog inputs.

A key difference between pin select and non pin select peripherals is that pin select peripherals are not associated with a default I/O pin. The peripheral must always be assigned to a specific I/O pin before it can be used. In contrast, non pin select peripherals are always available on a default pin, assuming that the peripheral is active and not conflicting with another peripheral.

10.4.2.1 Peripheral Pin Select Function Priority

Pin-selectable peripheral outputs (for example, OC and UART transmit) take priority over any general purpose digital functions permanently tied to that pin, such as PMP and port I/O. Specialized digital outputs, such as USB functionality, take priority over PPS outputs on the same pin. The pin diagrams at the beginning of this data sheet list peripheral outputs in order of priority. Refer to them for priority concerns on a particular pin.

Unlike devices with fixed peripherals, pin-selectable peripheral inputs never take ownership of a pin. The pin's output buffer is controlled by the pin's TRIS bit setting, or by a fixed peripheral on the pin. If the pin is configured in Digital mode, then the PPS input will operate correctly, reading the input. If an analog function is enabled on the same pin, the pin-selectable input will be disabled.

10.4.3 CONTROLLING PERIPHERAL PIN SELECT

Peripheral Pin Select features are controlled through two sets of Special Function Registers: one to map peripheral inputs and one to map outputs. Because they are separately controlled, a particular peripheral's input and output (if the peripheral has both) can be placed on any selectable function pin without constraint.

The association of a peripheral to a peripheral-selectable pin is handled in two different ways, depending on if an input or an output is being mapped.

10.4.3.1 Input Mapping

The inputs of the Peripheral Pin Select options are mapped on the basis of the peripheral; that is, a control register associated with a peripheral dictates the pin it will be mapped to. The RPINRx registers are used to configure peripheral input mapping (see Register 10-1 through Register 10-14). Each register contains up to two sets of 5-bit fields, with each set associated with one of the pin-selectable peripherals. Programming a given peripheral's bit field with an appropriate 6-bit value maps the RPn pin with that value to that peripheral. For any given device, the valid range of values for any of the bit fields corresponds to the maximum number of Peripheral Pin Select options supported by the device.

TABLE 10-2: SELECTABLE INPUT SOURCES (MAPS INPUT TO FUNCTION)(1)

Input Name	Function Name	Register	Function Mapping Bits
External Interrupt 1	INT1	RPINR0	INT1R<5:0>
External Interrupt 2	INT2	RPINR1	INT2R<5:0>
Input Capture 1	IC1	RPINR7	IC1R<5:0>
Input Capture 2	IC2	RPINR7	IC2R<5:0>
Input Capture 3	IC3	RPINR8	IC3R<5:0>
Input Capture 4	IC4	RPINR8	IC4R<5:0>
Input Capture 5	IC5	RPINR9	IC5R<5:0>
Output Compare Fault A	OCFA	RPINR11	OCFAR<5:0>
Output Compare Fault B	OCFB	RPINR11	OCFBR<5:0>
SPI1 Clock Input	SCK1IN	RPINR20	SCK1R<5:0>
SPI1 Data Input	SDI1	RPINR20	SDI1R<5:0>
SPI1 Slave Select Input	SS1IN	RPINR21	SS1R<5:0>
SPI2 Clock Input	SCK2IN	RPINR22	SCK2R<5:0>
SPI2 Data Input	SDI2	RPINR22	SDI2R<5:0>
SPI2 Slave Select Input	SS2IN	RPINR23	SS2R<5:0>
Timer2 External Clock	T2CK	RPINR3	T2CKR<5:0>
Timer3 External Clock	T3CK	RPINR3	T3CKR<5:0>
Timer4 External Clock	T4CK	RPINR4	T4CKR<5:0>
Timer5 External Clock	T5CK	RPINR4	T5CKR<5:0>
UART1 Clear To Send	$\overline{\text{U1CTS}}$	RPINR18	U1CTSR<5:0>
UART1 Receive	U1RX	RPINR18	U1RXR<5:0>
UART2 Clear To Send	$\overline{\text{U2CTS}}$	RPINR19	U2CTSR<5:0>
UART2 Receive	U2RX	RPINR19	U2RXR<5:0>

Note 1: Unless otherwise noted, all inputs use the Schmitt Trigger input buffers.

10.4.3.2 Output Mapping

In contrast to inputs, the outputs of the Peripheral Pin Select options are mapped on the basis of the pin. In this case, a control register associated with a particular pin dictates the peripheral output to be mapped. The RPORx registers are used to control output mapping. Each register contains up to two 5-bit fields, with each field being associated with one RPn pin (see Register 10-15 through Register 10-27). The value of the bit field corresponds to one of the peripherals and that peripheral's output is mapped to the pin (see Table 10-3).

Because of the mapping technique, the list of peripherals for output mapping also includes a null value of '000000'. This permits any given pin to remain disconnected from the output of any of the pin-selectable peripherals.

TABLE 10-3: SELECTABLE OUTPUT SOURCES (MAPS FUNCTION TO OUTPUT)

Output Function Number(1)	Function	Output Name
0	NULL(2)	Null
1	C1OUT	Comparator 1 Output
2	C2OUT	Comparator 2 Output
3	U1TX	UART1 Transmit
4	$\overline{\text{U1RTS}}$(3)	UART1 Request To Send
5	U2TX	UART2 Transmit
6	$\overline{\text{U2RTS}}$(3)	UART2 Request To Send
7	SDO1	SPI1 Data Output
8	SCK1OUT	SPI1 Clock Output
9	SS1OUT	SPI1 Slave Select Output
10	SDO2	SPI2 Data Output
11	SCK2OUT	SPI2 Clock Output
12	SS2OUT	SPI2 Slave Select Output
18	OC1	Output Compare 1
19	OC2	Output Compare 2
20	OC3	Output Compare 3
21	OC4	Output Compare 4
22	OC5	Output Compare 5
23-28	(unused)	NC
29	CTPLS	CTMU Output Pulse
30	C3OUT	Comparator 3 Output
31	(unused)	NC

Note 1: Setting the RPORx register with the listed value assigns that output function to the associated RPn pin.
2: The NULL function is assigned to all RPn outputs at device Reset and disables the RPn output function.
3: IrDA® BCLK functionality uses this output.

10.4.3.3 Mapping Limitations

The control schema of the Peripheral Pin Select is extremely flexible. Other than systematic blocks that prevent signal contention caused by two physical pins being configured as the same functional input, or two functional outputs configured as the same pin, there are no hardware enforced lock outs. The flexibility extends to the point of allowing a single input to drive multiple peripherals or a single functional output to drive multiple output pins.

10.4.3.4 PPS Mapping Exceptions for PIC24FJ64GA1 Family Devices

Although the PPS registers allow for up to 32 remappable pins, a maximum of 26 pins are implemented in 44-pin devices (RP0 through RP25). In 28-pin devices, none of the remappable pins above RP15 are implemented.

10.4.4 CONTROLLING CONFIGURATION CHANGES

Because peripheral remapping can be changed during run time, some restrictions on peripheral remapping are needed to prevent accidental configuration changes. PIC24F devices include three features to prevent alterations to the peripheral map:

- Control register lock sequence
- Continuous state monitoring
- Configuration bit remapping lock

10.4.4.1 Control Register Lock

Under normal operation, writes to the RPINRx and RPORx registers are not allowed. Attempted writes will appear to execute normally, but the contents of the registers will remain unchanged. To change these registers, they must be unlocked in hardware. The register lock is controlled by the IOLOCK bit (OSCCON<6>). Setting IOLOCK prevents writes to the control registers; clearing IOLOCK allows writes.

To set or clear IOLOCK, a specific command sequence must be executed:

1. Write 46h to OSCCON<7:0>.
2. Write 57h to OSCCON<7:0>.
3. Clear (or set) IOLOCK as a single operation.

Unlike the similar sequence with the oscillator's LOCK bit, IOLOCK remains in one state until changed. This allows all of the Peripheral Pin Selects to be configured with a single unlock sequence, followed by an update to all control registers, then locked with a second lock sequence.

10.4.4.2 Continuous State Monitoring

In addition to being protected from direct writes, the contents of the RPINRx and RPORx registers are constantly monitored in hardware by shadow registers. If an unexpected change in any of the registers occurs (such as cell disturbances caused by ESD or other external events), a Configuration Mismatch Reset will be triggered.

10.4.4.3 Configuration Bit Pin Select Lock

As an additional level of safety, the device can be configured to prevent more than one write session to the RPINRx and RPORx registers. The IOL1WAY (CW2<4>) Configuration bit blocks the IOLOCK bit from being cleared after it has been set once. If IOLOCK remains set, the register unlock procedure will not execute and the Peripheral Pin Select Control registers cannot be written to. The only way to clear the bit and re-enable peripheral remapping is to perform a device Reset.

In the default (unprogrammed) state, IOL1WAY is set, restricting users to one write session. Programming IOL1WAY allows users unlimited access (with the proper use of the unlock sequence) to the Peripheral Pin Select registers.

10.4.5 CONSIDERATIONS FOR PERIPHERAL PIN SELECTION

The ability to control Peripheral Pin Selection introduces several considerations into application design that could be overlooked. This is particularly true for several common peripherals that are available only as remappable peripherals.

The main consideration is that the Peripheral Pin Selects are not available on default pins in the device's default (Reset) state. Since all RPINRx registers reset to '11111' and all RPORx registers reset to '00000', all Peripheral Pin Select inputs are tied to VSS and all Peripheral Pin Select outputs are disconnected.

Note:	RP31 does not have to exist on a device for the registers to be reset to it, or for peripheral pin outputs to be tied to it.

This situation requires the user to initialize the device with the proper peripheral configuration before any other application code is executed. Since the IOLOCK bit resets in the unlocked state, it is not necessary to execute the unlock sequence after the device has come out of Reset. For application safety, however, it is best to set IOLOCK and lock the configuration after writing to the control registers.

Because the unlock sequence is timing-critical, it must be executed as an assembly language routine in the same manner as changes to the oscillator configuration. If the bulk of the application is written in C or another high-level language, the unlock sequence should be performed by writing in-line assembly.

Choosing the configuration requires the review of all Peripheral Pin Selects and their pin assignments, especially those that will not be used in the application. In all cases, unused pin-selectable peripherals should be disabled completely. Unused peripherals should have their inputs assigned to an unused RPn pin function. I/O pins with unused RPn functions should be configured with the null peripheral output.

The assignment of a peripheral to a particular pin does not automatically perform any other configuration of the pin's I/O circuitry. In theory, this means adding a pin-selectable output to a pin may mean inadvertently driving an existing peripheral input when the output is driven. Users must be familiar with the behavior of other fixed peripherals that share a remappable pin and know when to enable or disable them. To be safe, fixed digital peripherals that share the same pin should be disabled when not in use.

Along these lines, configuring a remappable pin for a specific peripheral does not automatically turn that feature on. The peripheral must be specifically configured for operation and enabled, as if it were tied to a fixed pin. Where this happens in the application code (immediately following device Reset and peripheral configuration or inside the main application routine) depends on the peripheral and its use in the application.

A final consideration is that Peripheral Pin Select functions neither override analog inputs, nor reconfigure pins with analog functions for digital I/O. If a pin is configured as an analog input on device Reset, it must be explicitly reconfigured as digital I/O when used with a Peripheral Pin Select.

Example 10-2 shows a configuration for bidirectional communication with flow control using UART1. The following input and output functions are used:

- Input Functions: U1RX, $\overline{\text{U1CTS}}$
- Output Functions: U1TX, $\overline{\text{U1RTS}}$

EXAMPLE 10-2: CONFIGURING UART1 INPUT AND OUTPUT FUNCTIONS IN ASSEMBLY CODE

```
;unlock     registers
push        w1;
push        w2;
push        w3;
mov         #OSCCON, w1;
mov         #0x46, w2;
mov         #0x57, w3;
mov.b       w2, [w1];
mov.b       w3, [w1];
bclr        OSCCON, #6;

; Configure Input Functions (Table10-2)
; Assign U1CTS To Pin RP1, U1RX To Pin RP0
mov         #0x0100, w1;
mov         w1,RPINR18;

; Configure Output Functions (Table 10-3)
; Assign U1RTS To Pin RP3, U1TX To Pin RP2
mov         #0x0403, w1;
mov         w1, RPOR1;

;lock       registers
mov         #OSCCON, w1;
mov         #0x46, w2;
mov         #0x57, w3;
mov.b       w2, [w1];
mov.b       w3, [w1];
bset        OSCCON, #6;
pop         w3;
pop         w2;
pop         w1;
```

EXAMPLE 10-3: CONFIGURING UART1 INPUT AND OUTPUT FUNCTIONS IN C

```
//unlock registers
__builtin_write_OSCCONL(OSCCON & 0xBF);

// Configure Input Functions (Table 9-1)
// Assign U1RX To Pin RP0
RPINR18bits.U1RXR = 0;

// Assign U1CTS To Pin RP1
RPINR18bits.U1CTSR = 1;

// Configure Output Functions (Table 9-2)
// Assign U1TX To Pin RP2
RPOR1bits.RP2R = 3;

// Assign U1RTS To Pin RP3
RPOR1bits.RP3R = 4;

//lock registers
__builtin_write_OSCCONL(OSCCON | 0x40);
```

10.4.6 PERIPHERAL PIN SELECT REGISTERS

The PIC24FJ64GA104 family of devices implements a total of 27 registers for remappable peripheral configuration:

- Input Remappable Peripheral Registers (14)
- Output Remappable Peripheral Registers (13)

> **Note:** Input and output register values can only be changed if IOLOCK (OSCCON<6>) = `0`. See **Section 10.4.4.1 “Control Register Lock”** for a specific command sequence.

REGISTER 10-1: RPINR0: PERIPHERAL PIN SELECT INPUT REGISTER 0

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 15-13	**Unimplemented:** Read as ‘0’
bit 12-8	**INT1R<4:0>:** Assign External Interrupt 1 (INT1) to Corresponding RPn or RPIn Pin bits
bit 7-0	**Unimplemented:** Read as ‘0’

REGISTER 10-2: RPINR1: PERIPHERAL PIN SELECT INPUT REGISTER 1

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0
bit 7							bit 0

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

bit 15-5	**Unimplemented:** Read as ‘0’
bit 4-0	**INT2R<4:0>:** Assign External Interrupt 2 (INT2) to Corresponding RPn or RPIn pin bits

REGISTER 10-3: RPINR3: PERIPHERAL PIN SELECT INPUT REGISTER 3

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	T3CKR4	T3CKR3	T3CKR2	T3CKR1	T3CKR0
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	T2CKR4	T2CKR3	T2CKR2	T2CKR1	T2CKR0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **T3CKR<4:0>:** Assign Timer3 External Clock (T3CK) to Corresponding RPn or RPIn Pin bits

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **T2CKR<4:0>:** Assign Timer2 External Clock (T2CK) to Corresponding RPn or RPIn Pin bits

REGISTER 10-4: RPINR4: PERIPHERAL PIN SELECT INPUT REGISTER 4

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	T5CKR4	T5CKR3	T5CKR2	T5CKR1	T5CKR0
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	T4CKR4	T4CKR3	T4CKR2	T4CKR1	T4CKR0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **T5CKR<4:0>:** Assign Timer5 External Clock (T5CK) to Corresponding RPn or RPIn Pin bits

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **T4CKR<4:0>:** Assign Timer4 External Clock (T4CK) to Corresponding RPn or RPIn Pin bits

REGISTER 10-5: RPINR7: PERIPHERAL PIN SELECT INPUT REGISTER 7

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC2R4	IC2R3	IC2R2	IC2R1	IC2R0
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC1R4	IC1R3	IC1R2	IC1R1	IC1R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **IC2R<4:0>:** Assign Input Capture 2 (IC2) to Corresponding RPn or RPIn Pin bits

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **IC1R<4:0>:** Assign Input Capture 1 (IC1) to Corresponding RPn or RPIn Pin bits

REGISTER 10-6: RPINR8: PERIPHERAL PIN SELECT INPUT REGISTER 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC4R4	IC4R3	IC4R2	IC4R1	IC4R0
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC3R4	IC3R3	IC3R2	IC3R1	IC3R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **IC4R<4:0>:** Assign Input Capture 4 (IC4) to Corresponding RPn or RPIn Pin bits

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **IC3R<4:0>:** Assign Input Capture 3 (IC3) to Corresponding RPn or RPIn Pin bits

REGISTER 10-7: RPINR9: PERIPHERAL PIN SELECT INPUT REGISTER 9

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC5R4	IC5R3	IC5R2	IC5R1	IC5R0
bit 7							bit 0

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

bit 15-5 **Unimplemented:** Read as ‘0’

bit 4-0 **IC5R<4:0>:** Assign Input Capture 5 (IC5) to Corresponding RPn or RPIn Pin bits

REGISTER 10-8: RPINR11: PERIPHERAL PIN SELECT INPUT REGISTER 11

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **OCFBR<4:0>:** Assign Output Compare Fault B (OCFB) to Corresponding RPn or RPIn Pin bits

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **OCFAR<4:0>:** Assign Output Compare Fault A (OCFA) to Corresponding RPn or RPIn Pin bits

REGISTER 10-9: RPINR18: PERIPHERAL PIN SELECT INPUT REGISTER 18

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0
bit 7							bit 0

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

bit 15-13	**Unimplemented:** Read as ‘0’
bit 12-8	**U1CTSR<4:0>:** Assign UART1 Clear to Send ($\overline{\text{U1CTS}}$) to Corresponding RPn or RPIn Pin bits
bit 7-5	**Unimplemented:** Read as ‘0’
bit 4-0	**U1RXR<4:0>:** Assign UART1 Receive (U1RX) to Corresponding RPn or RPIn Pin bits

REGISTER 10-10: RPINR19: PERIPHERAL PIN SELECT INPUT REGISTER 19

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0
bit 7							bit 0

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

bit 15-13	**Unimplemented:** Read as ‘0’
bit 12-8	**U2CTSR<4:0>:** Assign UART2 Clear to Send ($\overline{\text{U2CTS}}$) to Corresponding RPn or RPIn Pin bits
bit 7-5	**Unimplemented:** Read as ‘0’
bit 4-0	**U2RXR<4:0>:** Assign UART2 Receive (U2RX) to Corresponding RPn or RPIn Pin bits

REGISTER 10-11: RPINR20: PERIPHERAL PIN SELECT INPUT REGISTER 20

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SDI1R4	SDI1R3	SDI1R2	SDI1R1	SDI1R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **SCK1R<4:0>:** Assign SPI1 Clock Input (SCK1IN) to Corresponding RPn or RPIn Pin bits

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **SDI1R<4:0>:** Assign SPI1 Data Input (SDI1) to Corresponding RPn or RPIn Pin bits

REGISTER 10-12: RPINR21: PERIPHERAL PIN SELECT INPUT REGISTER 21

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0
bit 7							bit 0

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

bit 15-5 **Unimplemented:** Read as ‘0’

bit 4-0 **SS1R<4:0>:** Assign SPI1 Slave Select Input (SS1IN) to Corresponding RPn or RPIn Pin bits

REGISTER 10-13: RPINR22: PERIPHERAL PIN SELECT INPUT REGISTER 22

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **SCK2R<4:0>:** Assign SPI2 Clock Input (SCK2IN) to Corresponding RPn or RPIn Pin bits

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **SDI2R<4:0>:** Assign SPI2 Data Input (SDI2) to Corresponding RPn or RPIn Pin bits

REGISTER 10-14: RPINR23: PERIPHERAL PIN SELECT INPUT REGISTER 23

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0
bit 7							bit 0

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

bit 15-5 **Unimplemented:** Read as ‘0’

bit 4-0 **SS2R<4:0>:** Assign SPI2 Slave Select Input (SS2IN) to Corresponding RPn or RPIn Pin bits

REGISTER 10-15: RPOR0: PERIPHERAL PIN SELECT OUTPUT REGISTER 0

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **RP1R<4:0<:** RP1 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP1 (see Table 10-3 for peripheral function numbers).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RP0R<4:0>:** RP0 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP0 (see Table 10-3 for peripheral function numbers).

REGISTER 10-16: RPOR1: PERIPHERAL PIN SELECT OUTPUT REGISTER 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP3R4	RP3R3	RP3R2	RP3R1	RP3R0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP2R4	RP2R3	RP2R2	RP2R1	RP2R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **RP3R<4:0>:** RP3 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP3 (see Table 10-3 for peripheral function numbers).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RP2R<4:0>:** RP2 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP2 (see Table 10-3 for peripheral function numbers).

REGISTER 10-17: RPOR2: PERIPHERAL PIN SELECT OUTPUT REGISTER 2

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP5R4	RP5R3	RP5R2	RP5R1	RP5R0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **RP5R<4:0>:** RP5 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP5 (see Table 10-3 for peripheral function numbers).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RP4R<4:0>:** RP4 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP4 (see Table 10-3 for peripheral function numbers).

REGISTER 10-18: RPOR3: PERIPHERAL PIN SELECT OUTPUT REGISTER 3

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **RP7R<4:0>:** RP7 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP7 (see Table 10-3 for peripheral function numbers).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RP6R<4:0>:** RP6 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP6 (see Table 10-3 for peripheral function numbers).

REGISTER 10-19: RPOR4: PERIPHERAL PIN SELECT OUTPUT REGISTER 4

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP9R4	RP9R3	RP9R2	RP9R1	RP9R0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP8R4	RP8R3	RP8R2	RP8R1	RP8R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **RP9R<4:0>:** RP9 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP9 (see Table 10-3 for peripheral function numbers).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RP8R<4:0>:** RP8 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP8 (see Table 10-3 for peripheral function numbers).

REGISTER 10-20: RPOR5: PERIPHERAL PIN SELECT OUTPUT REGISTER 5

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP11R4	RP11R3	RP11R2	RP11R1	RP11R0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP10R4	RP10R3	RP10R2	RP10R1	RP10R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **RP11R<4:0>:** RP11 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP11 (see Table 10-3 for peripheral function numbers).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RP10R<4:0>:** RP10 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP10 (see Table 10-3 for peripheral function numbers).

REGISTER 10-21: RPOR6: PERIPHERAL PIN SELECT OUTPUT REGISTER 6

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP13R4	RP13R3	RP13R2	RP13R1	RP13R0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP12R4	RP12R3	RP12R2	RP12R1	RP12R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **RP13R<4:0>:** RP13 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP13 (see Table 10-3 for peripheral function numbers).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RP12R<4:0>:** RP12 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP12 (see Table 10-3 for peripheral function numbers).

REGISTER 10-22: RPOR7: PERIPHERAL PIN SELECT OUTPUT REGISTER 7

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP15R4	RP15R3	RP15R2	RP15R1	RP15R0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP14R4	RP14R3	RP14R2	RP14R1	RP14R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **RP15R<4:0>:** RP15 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP0 (see Table 10-3 for peripheral function numbers).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RP14R<4:0>:** RP14 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP14 (see Table 10-3 for peripheral function numbers).

REGISTER 10-23: RPOR8: PERIPHERAL PIN SELECT OUTPUT REGISTER 8(1)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP17R4	RP17R3	RP17R2	RP17R1	RP17R0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP16R4	RP16R3	RP16R2	RP16R1	RP16R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as '0'

bit 12-8 **RP17R<4:0>:** RP17 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP17 (see Table 10-3 for peripheral function numbers).

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

bit 4-0 **RP16R<4:0>:** RP16 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP16 (see Table 10-3 for peripheral function numbers).

Note 1: This register is unimplemented in 28-pin devices; all bits read as '0'.

REGISTER 10-24: RPOR9: PERIPHERAL PIN SELECT OUTPUT REGISTER 9(1)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as '0'

bit 12-8 **RP19R<4:0>:** RP19 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP19 (see Table 10-3 for peripheral function numbers).

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

bit 4-0 **RP18R<4:0>:** RP18 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP18 (see Table 10-3 for peripheral function numbers).

Note 1: This register is unimplemented in 28-pin devices; all bits read as '0'.

REGISTER 10-25: RPOR10: PERIPHERAL PIN SELECT OUTPUT REGISTER 10(1)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP21R4	RP21R3	RP21R2	RP21R1	RP21R0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **RP21R<4:0>:** RP21 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP21 (see Table 10-3 for peripheral function numbers).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RP20R<4:0>:** RP20 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP20 (see Table 10-3 for peripheral function numbers).

Note 1: This register is unimplemented in 28-pin devices; all bits read as ‘0’.

REGISTER 10-26: RPOR11: PERIPHERAL PIN SELECT OUTPUT REGISTER 11(1)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP23R4	RP23R3	RP23R2	RP23R1	RP23R0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP22R4	RP22R3	RP22R2	RP22R1	RP22R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **RP23R<4:0>:** RP23 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP23 (see Table 10-3 for peripheral function numbers).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RP22R<4:0>:** RP22 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP22 (see Table 10-3 for peripheral function numbers).

Note 1: This register is unimplemented in 28-pin devices; all bits read as ‘0’.

REGISTER 10-27: RPOR12: PERIPHERAL PIN SELECT OUTPUT REGISTER 12(1)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP25R4	RP25R3	RP25R2	RP25R1	RP25R0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP24R4	RP24R3	RP24R2	RP24R1	RP24R0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **RP25R<5:0>:** RP25 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP25 (see Table 10-3 for peripheral function numbers).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RP24R<5:0>:** RP24 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP24 (see Table 10-3 for peripheral function numbers).

Note 1: This register is unimplemented in 28-pin devices; all bits read as ‘0’.

11.0 TIMER1

Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 14. "Timers"** (DS39704).

The Timer1 module is a 16-bit timer which can serve as the time counter for the Real-Time Clock (RTC) or operate as a free-running, interval timer/counter. Timer1 can operate in three modes:

- 16-Bit Timer
- 16-Bit Synchronous Counter
- 16-Bit Asynchronous Counter

Timer1 also supports these features:

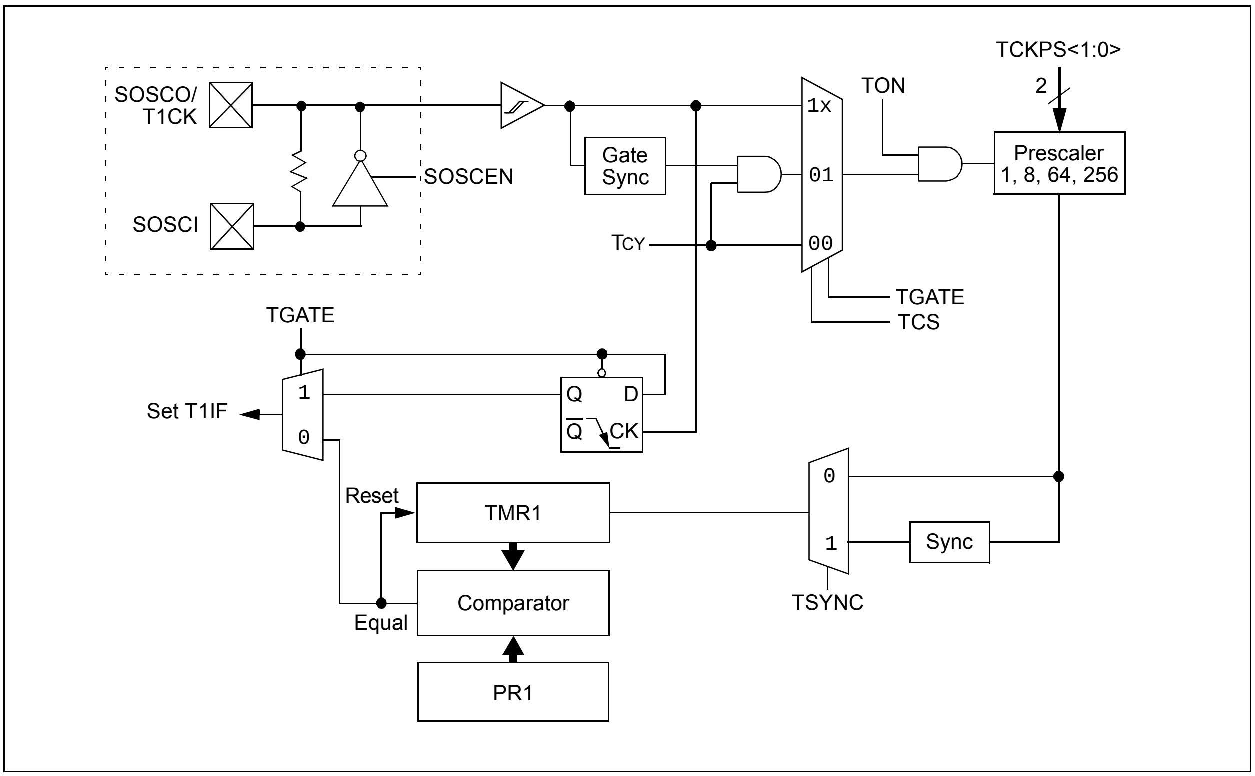
- Timer Gate Operation
- Selectable Prescaler Settings
- Timer Operation during CPU Idle and Sleep modes
- Interrupt on 16-Bit Period Register Match or Falling Edge of External Gate Signal

Figure 11-1 presents a block diagram of the 16-bit timer module.

To configure Timer1 for operation:

1. Set the TON bit (= 1).
2. Select the timer prescaler ratio using the TCKPS<1:0> bits.
3. Set the Clock and Gating modes using the TCS and TGATE bits.
4. Set or clear the TSYNC bit to configure synchronous or asynchronous operation.
5. Load the timer period value into the PR1 register.
6. If interrupts are required, set the interrupt enable bit, T1IE. Use the priority bits, T1IP<2:0>, to set the interrupt priority.

FIGURE 11-1: 16-BIT TIMER1 MODULE BLOCK DIAGRAM



REGISTER 11-1: T1CON: TIMER1 CONTROL REGISTER(1)

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—
bit 7							bit 0

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

bit 15 **TON:** Timer1 On bit
1 = Starts 16-bit Timer1
0 = Stops 16-bit Timer1

bit 14 **Unimplemented:** Read as '0'

bit 13 **TSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **TGATE:** Timer1 Gated Time Accumulation Enable bit
When TCS = 1:
This bit is ignored.
When TCS = 0:
1 = Gated time accumulation enabled
0 = Gated time accumulation disabled

bit 5-4 **TCKPS<1:0>:** Timer1 Input Clock Prescale Select bits
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3 **Unimplemented:** Read as '0'

bit 2 **TSYNC:** Timer1 External Clock Input Synchronization Select bit
When TCS = 1:
1 = Synchronize external clock input
0 = Do not synchronize external clock input
When TCS = 0:
This bit is ignored.

bit 1 **TCS:** Timer1 Clock Source Select bit
1 = External clock from T1CK pin (on the rising edge)
0 = Internal clock (FOSC/2)

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

Note 1: Changing the value of TxCON while the timer is running (TON = 1) causes the timer prescale counter to reset and is not recommended.

12.0 TIMER2/3 AND TIMER4/5

Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 14. "Timers"** (DS39704).

The Timer2/3 and Timer4/5 modules are 32-bit timers, which can also be configured as four independent 16-bit timers with selectable operating modes.

As 32-bit timers, Timer2/3 and Timer4/5 can each operate in three modes:

- Two Independent 16-Bit Timers with All 16-Bit Operating modes (except Asynchronous Counter mode)
- Single 32-Bit Timer
- Single 32-Bit Synchronous Counter

They also support these features:

- Timer Gate Operation
- Selectable Prescaler Settings
- Timer Operation during Idle and Sleep modes
- Interrupt on a 32-Bit Period Register Match
- ADC Event Trigger (Timer4/5 only)

Individually, all four of the 16-bit timers can function as synchronous timers or counters. They also offer the features listed above, except for the ADC event trigger; this is implemented only with Timer5. The operating modes and enabled features are determined by setting the appropriate bit(s) in the T2CON, T3CON, T4CON and T5CON registers. T2CON and T4CON are shown in generic form in Register 12-1; T3CON and T5CON are shown in Register 12-2.

For 32-bit timer/counter operation, Timer2 and Timer4 are the least significant word; Timer3 and Timer4 are the most significant word of the 32-bit timers.

Note:	For 32-bit operation, T3CON and T5CON control bits are ignored. Only T2CON and T4CON control bits are used for setup and control. Timer2 and Timer4 clock and gate inputs are utilized for the 32-bit timer modules, but an interrupt is generated with the Timer3 or Timer5 interrupt flags.

To configure Timer2/3 or Timer4/5 for 32-bit operation:

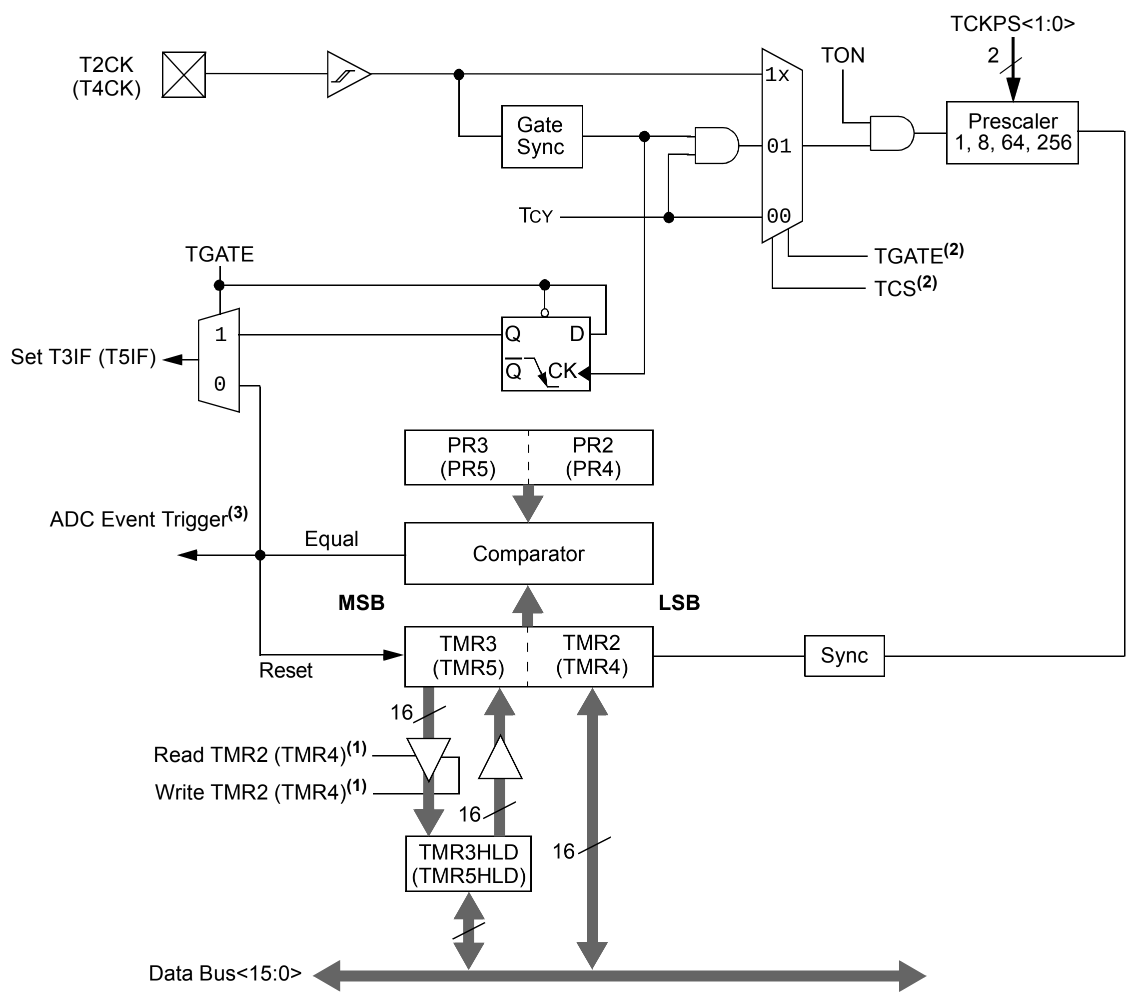
1. Set the T32 bit (T2CON<3> or T4CON<3> = 1).
2. Select the prescaler ratio for Timer2 or Timer4 using the TCKPS<1:0> bits.
3. Set the Clock and Gating modes using the TCS and TGATE bits. If TCS is set to an external clock, RPINRx (TxCK) must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.
4. Load the timer period value. PR3 (or PR5) will contain the most significant word of the value while PR2 (or PR4) contains the least significant word.
5. If interrupts are required, set the interrupt enable bit, T3IE or T5IE; use the priority bits, T3IP<2:0> or T5IP<2:0>, to set the interrupt priority. Note that while Timer2 or Timer4 controls the timer, the interrupt appears as a Timer3 or Timer5 interrupt.
6. Set the TON bit (= 1).

The timer value, at any point, is stored in the register pair, TMR3:TMR2 (or TMR5:TMR4). TMR3 (TMR5) always contains the most significant word of the count, while TMR2 (TMR4) contains the least significant word.

To configure any of the timers for individual 16-bit operation:

1. Clear the T32 bit corresponding to that timer (T2CON<3> for Timer2 and Timer3 or T4CON<3> for Timer4 and Timer5).
2. Select the timer prescaler ratio using the TCKPS<1:0> bits.
3. Set the Clock and Gating modes using the TCS and TGATE bits. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.
4. Load the timer period value into the PRx register.
5. If interrupts are required, set the interrupt enable bit, TxIE; use the priority bits, TxIP<2:0>, to set the interrupt priority.
6. Set the TON bit (TxCON<15> = 1).

FIGURE 12-1: TIMER2/3 AND TIMER4/5 (32-BIT) BLOCK DIAGRAM

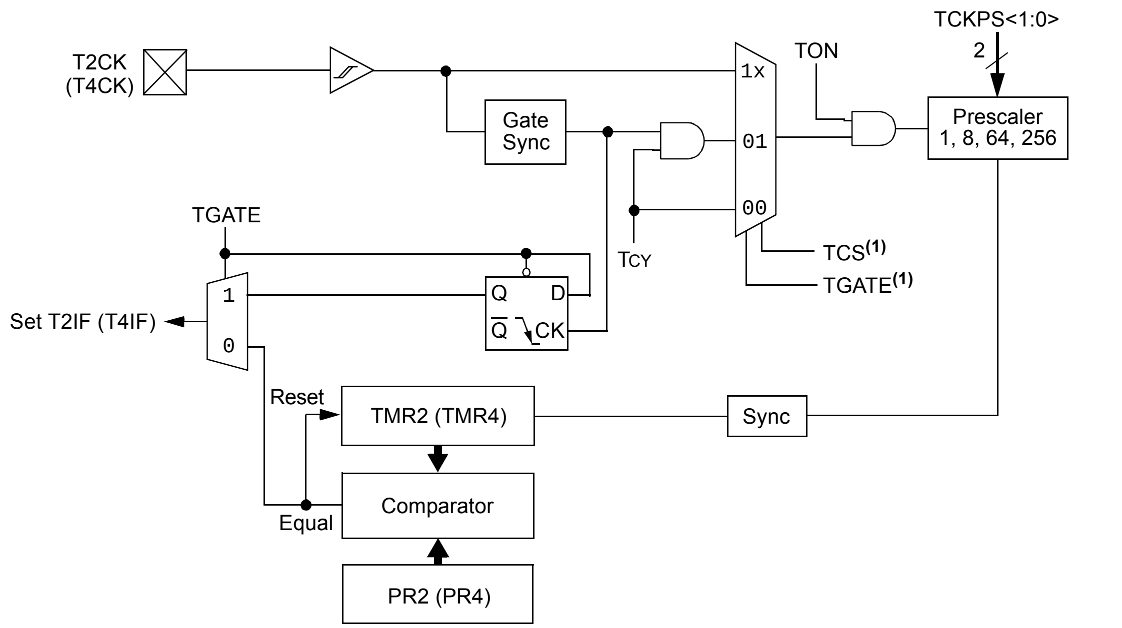


Note 1: The 32-Bit Timer Configuration bit, T32, must be set for 32-bit timer/counter operation. All control bits are respective to the T2CON and T4CON registers.

2: The timer clock input must be assigned to an available RPn pin before use. Please see **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

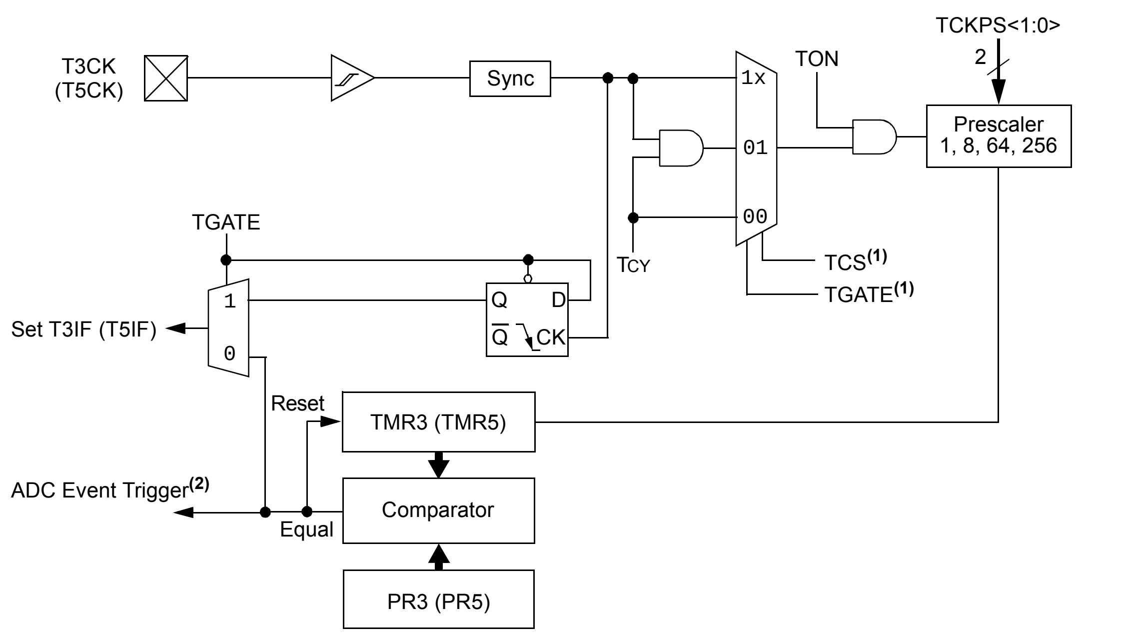
3: The ADC event trigger is available only on Timer 2/3 in 32-bit mode and Timer 3 in 16-bit mode.

FIGURE 12-2: TIMER2 AND TIMER4 (16-BIT SYNCHRONOUS) BLOCK DIAGRAM



Note 1: The timer clock input must be assigned to an available RPn pin before use. Please see **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

FIGURE 12-3: TIMER3 AND TIMER5 (16-BIT ASYNCHRONOUS) BLOCK DIAGRAM



Note 1: The timer clock input must be assigned to an available RPn pin before use. Please see **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

2: The ADC event trigger is available only on Timer3.

REGISTER 12-1: TxCON: TIMER2 AND TIMER4 CONTROL REGISTER[3]

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	T32[1]	—	TCS[2]	—
bit 7							bit 0

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

bit 15 **TON:** Timerx On bit
When TxCON<3> = 1:
1 = Starts 32-bit Timerx/y
0 = Stops 32-bit Timerx/y
When TxCON<3> = 0:
1 = Starts 16-bit Timerx
0 = Stops 16-bit Timerx

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **TSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-7 **Unimplemented:** Read as ‘0’

bit 6 **TGATE:** Timerx Gated Time Accumulation Enable bit
When TCS = 1:
This bit is ignored.
When TCS = 0:
1 = Gated time accumulation is enabled
0 = Gated time accumulation is disabled

bit 5-4 **TCKPS<1:0>:** Timerx Input Clock Prescale Select bits
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3 **T32:** 32-Bit Timer Mode Select bit[1]
1 = Timerx and Timery form a single 32-bit timer
0 = Timerx and Timery act as two 16-bit timers
In 32-bit mode, T3CON control bits do not affect 32-bit timer operation.

bit 2 **Unimplemented:** Read as ‘0’

bit 1 **TCS:** Timerx Clock Source Select bit[2]
1 = External clock from pin, TxCK (on the rising edge)
0 = Internal clock (FOSC/2)

bit 0 **Unimplemented:** Read as ‘0’

Note 1: In 32-bit mode, the T3CON or T5CON control bits do not affect 32-bit timer operation.
2: If TCS = 1, RPINRx (TxCK) must be configured to an available RPn pin. For more information, see **Section 10.4 “Peripheral Pin Select (PPS)”**.
3: Changing the value of TxCON while the timer is running (TON = 1) causes the timer prescale counter to reset and is not recommended.

REGISTER 12-2: TyCON: TIMER3 AND TIMER5 CONTROL REGISTER[(3)]

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON[(1)]	—	TSIDL[(1)]	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
—	TGATE[(1)]	TCKPS1[(1)]	TCKPS0[(1)]	—	—	TCS[(1,2)]	—
bit 7							bit 0

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

bit 15 **TON:** Timery On bit[(1)]
1 = Starts 16-bit Timery
0 = Stops 16-bit Timery

bit 14 **Unimplemented:** Read as '0'

bit 13 **TSIDL:** Stop in Idle Mode bit[(1)]
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **TGATE:** Timery Gated Time Accumulation Enable bit[(1)]
When TCS = 1:
This bit is ignored.
When TCS = 0:
1 = Gated time accumulation is enabled
0 = Gated time accumulation is disabled

bit 5-4 **TCKPS<1:0>:** Timery Input Clock Prescale Select bits[(1)]
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **TCS:** Timery Clock Source Select bit[(1,2)]
1 = External clock from pin TyCK (on the rising edge)
0 = Internal clock (FOSC/2)

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

Note 1: When 32-bit operation is enabled (T2CON<3> or T4CON<3> = 1), these bits have no effect on Timery operation; all timer functions are set through T2CON and T4CON.

2: If TCS = 1, RPINRx (TxCK) must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

3: Changing the value of TyCON while the timer is running (TON = 1) causes the timer prescale counter to reset and is not recommended.

NOTES:

13.0 INPUT CAPTURE WITH DEDICATED TIMERS

Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 34. "Input Capture with Dedicated Timer"** (DS39722).

Devices in the PIC24FJ64GA104 family all feature 5 independent input capture modules. Each of the modules offers a wide range of configuration and operating options for capturing external pulse events and generating interrupts.

Key features of the input capture module include:

- Hardware-configurable for 32-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with up to 20 user-selectable trigger/sync sources available
- A 4-level FIFO buffer for capturing and holding timer values for several events
- Configurable interrupt generation
- Up to 6 clock sources available for each module, driving a separate internal 16-bit counter

The module is controlled through two registers: ICxCON1 (Register 13-1) and ICxCON2 (Register 13-2). A general block diagram of the module is shown in Figure 13-1.

13.1 General Operating Modes

13.1.1 SYNCHRONOUS AND TRIGGER MODES

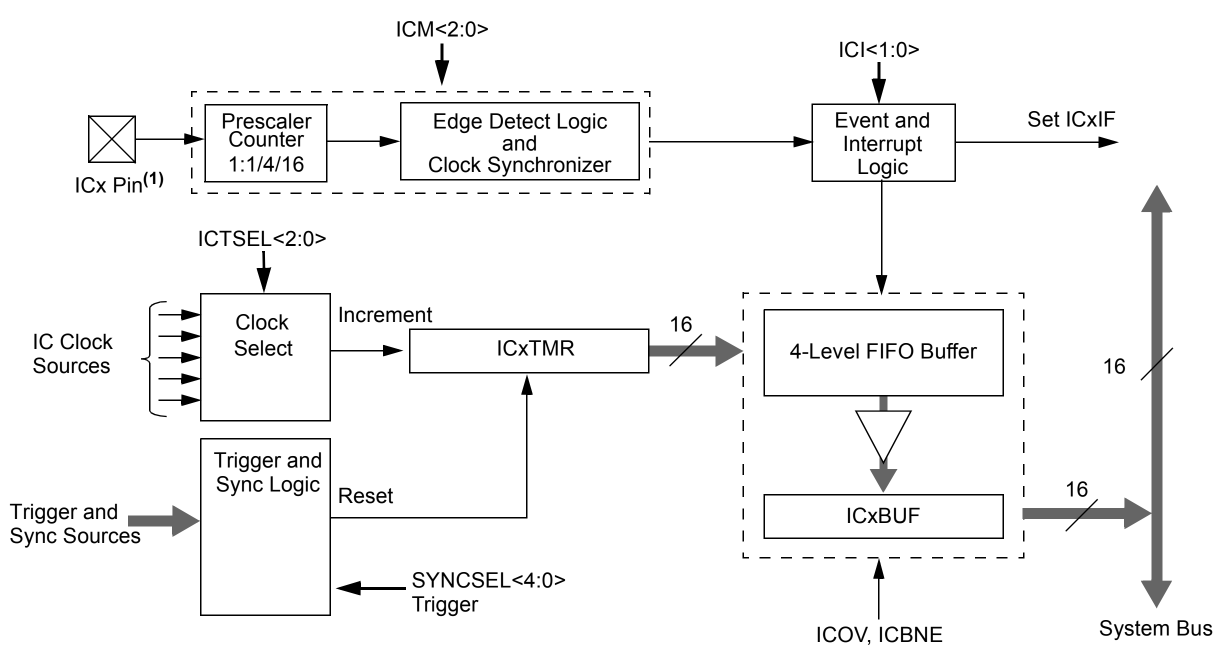
By default, the input capture module operates in a free-running mode. The internal 16-bit counter ICxTMR counts up continuously, wrapping around from FFFFh to 0000h on each overflow, with its period synchronized to the selected external clock source. When a capture event occurs, the current 16-bit value of the internal counter is written to the FIFO buffer.

In Synchronous mode, the module begins capturing events on the ICx pin as soon as its selected clock source is enabled. Whenever an event occurs on the selected sync source, the internal counter is reset. In Trigger mode, the module waits for a Sync event from another internal module to occur before allowing the internal counter to run.

Standard, free-running operation is selected by setting the SYNCSEL bits to '`00000`' and clearing the ICTRIG bit (ICxCON2<7>). Synchronous and Trigger modes are selected any time the SYNCSEL bits are set to any value except '`00000`'. The ICTRIG bit selects either Synchronous or Trigger mode; setting the bit selects Trigger mode operation. In both modes, the SYNCSEL bits determine the sync/trigger source.

When the SYNCSEL bits are set to '`00000`' and ICTRIG is set, the module operates in Software Trigger mode. In this case, capture operations are started by manually setting the TRIGSTAT bit (ICxCON2<6>).

FIGURE 13-1: INPUT CAPTURE BLOCK DIAGRAM



Note 1: The ICx inputs must be assigned to an available RPn pin before use. Please see **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

13.1.2 CASCADED (32-BIT) MODE

By default, each module operates independently with its own 16-bit timer. To increase resolution, adjacent even and odd modules can be configured to function as a single 32-bit module. (For example, modules 1 and 2 are paired, as are modules 3 and 4, and so on.) The odd-numbered module (ICx) provides the Least Significant 16 bits of the 32-bit register pairs, and the even module (ICy) provides the Most Significant 16 bits. Wrap-arounds of the ICx registers cause an increment of their corresponding ICy registers.

Cascaded operation is configured in hardware by setting the IC32 bits (ICxCON2<8>) for both modules.

13.2 Capture Operations

The input capture module can be configured to capture timer values and generate interrupts on rising edges on ICx, or all transitions on ICx. Captures can be configured to occur on all rising edges or just some (every 4th or 16th). Interrupts can be independently configured to generate on each event or a subset of events.

To set up the module for capture operations:

1. Configure the ICx input for one of the available Peripheral Pin Select pins.
2. If Synchronous mode is to be used, disable the sync source before proceeding.
3. Make sure that any previous data has been removed from the FIFO by reading ICxBUF until the ICBNE bit (ICxCON1<3>) is cleared.
4. Set the SYNCSEL bits (ICxCON2<4:0>) to the desired sync/trigger source.
5. Set the ICTSEL bits (ICxCON1<12:10>) for the desired clock source. If the desired clock source is running, set the ICTSEL bits before the Input Capture module is enabled for proper synchronization with the desired clock source.
6. Set the ICI bits (ICxCON1<6:5>) to the desired interrupt frequency.
7. Select Synchronous or Trigger mode operation:
   a) Check that the SYNCSEL bits are not set to '00000'.
   b) For Synchronous mode, clear the ICTRIG bit (ICxCON2<7>).
   c) For Trigger mode, set ICTRIG and clear the TRIGSTAT bit (ICxCON2<6>).
8. Set the ICM bits (ICxCON1<2:0>) to the desired operational mode.
9. Enable the selected trigger/sync source.

For 32-bit cascaded operations, the setup procedure is slightly different:

1. Set the IC32 bits for both modules (ICyCON2<8> and (ICxCON2<8>), enabling the even-numbered module first. This ensures the modules will start functioning in unison.
2. Set the ICTSEL and SYNCSEL bits for both modules to select the same sync/trigger and time base source. Set the even module first, then the odd module. Both modules must use the same ICTSEL and SYNCSEL settings.
3. Clear the ICTRIG bit of the even module (ICyCON2<7>); this forces the module to run in Synchronous mode with the odd module, regardless of its trigger setting.
4. Use the odd module's ICI bits (ICxCON1<6:5>) to the desired interrupt frequency.
5. Use the ICTRIG bit of the odd module (ICxCON2<7>) to configure Trigger or Synchronous mode operation.

> **Note:** For Synchronous mode operation, enable the sync source as the last step. Both input capture modules are held in Reset until the sync source is enabled.

6. Use the ICM bits of the odd module (ICxCON1<2:0>) to set the desired capture mode.

The module is ready to capture events when the time base and the trigger/sync source are enabled. When the ICBNE bit (ICxCON1<3>) becomes set, at least one capture value is available in the FIFO. Read input capture values from the FIFO until the ICBNE clears to '0'.

For 32-bit operation, read both the ICxBUF and ICyBUF for the full 32-bit timer value (ICxBUF for the lsw, ICyBUF for the msw). At least one capture value is available in the FIFO buffer when the odd module's ICBNE bit (ICxCON1<3>) becomes set. Continue to read the buffer registers until ICBNE is cleared (perform automatically by hardware).

REGISTER 13-1: ICxCON1: INPUT CAPTURE x CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-0, HCS	R-0, HCS	R/W-0	R/W-0	R/W-0
—	ICI1	ICI0	ICOV	ICBNE	ICM2[(1)]	ICM1[(1)]	ICM0[(1)]
bit 7							bit 0

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

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **ICSIDL:** Input Capture x Module Stop in Idle Control bit
1 = Input capture module halts in CPU Idle mode
0 = Input capture module continues to operate in CPU Idle mode

bit 12-10 **ICTSEL<2:0>:** Input Capture Timer Select bits
111 = System clock (Fosc/2)
110 = Reserved
101 = Reserved
100 = Timer1
011 = Timer5
010 = Timer4
001 = Timer2
000 = Timer3

bit 9-7 **Unimplemented:** Read as '0'

bit 6-5 **ICI<1:0>:** Select Number of Captures per Interrupt bits
11 = Interrupt on every fourth capture event
10 = Interrupt on every third capture event
01 = Interrupt on every second capture event
00 = Interrupt on every capture event

bit 4 **ICOV:** Input Capture x Overflow Status Flag bit (read-only)
1 = Input capture overflow occurred
0 = No input capture overflow occurred

bit 3 **ICBNE:** Input Capture x Buffer Empty Status bit (read-only)
1 = Input capture buffer is not empty, at least one more capture value can be read
0 = Input capture buffer is empty

bit 2-0 **ICM<2:0>:** Input Capture Mode Select bits[(1)]
111 = Interrupt mode: input capture functions as interrupt pin only when device is in Sleep or Idle mode (rising edge detect only, all other control bits are not applicable)
110 = Unused (module disabled)
101 = Prescaler Capture mode: capture on every 16th rising edge
100 = Prescaler Capture mode: capture on every 4th rising edge
011 = Simple Capture mode: capture on every rising edge
010 = Simple Capture mode: capture on every falling edge
001 = Edge Detect Capture mode: capture on every edge (rising and falling); ICI<1:0 bits do not control interrupt generation for this mode
000 = Input capture module turned off

Note 1: The ICx input must also be configured to an available RPn pin. For more information, see **Section 10.4 "Peripheral Pin Select (PPS)"**.

REGISTER 13-2: ICxCON2: INPUT CAPTURE x CONTROL REGISTER 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	IC32
bit 15							bit 8

R/W-0	R/W-0, HS	U-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-1
ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7							bit 0

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

bit 15-9 **Unimplemented:** Read as ‘0’

bit 8 **IC32:** Cascade Two IC Modules Enable bit (32-bit operation)
1 = ICx and ICy operate in cascade as a 32-bit module (this bit must be set in both modules)
0 = ICx functions independently as a 16-bit module

bit 7 **ICTRIG:** ICx Trigger/Sync Select bit
1 = Trigger ICx from source designated by SYNCSELx bits
0 = Synchronize ICx with source designated by SYNCSELx bits

bit 6 **TRIGSTAT:** Timer Trigger Status bit
1 = Timer source has been triggered and is running (set in hardware, can be set in software)
0 = Timer source has not been triggered and is being held clear

bit 5 **Unimplemented:** Read as ‘0’

bit 4-0 **SYNCSEL<4:0>:** Trigger/Synchronization Source Selection bits
11111 = Reserved
11110 = Reserved
11101 = Reserved
11100 = CTMU(1)
11011 = A/D(1)
11010 = Comparator 3(1)
11001 = Comparator 2(1)
11000 = Comparator 1(1)
10111 = Input Capture 4
10110 = Input Capture 3
10101 = Input Capture 2
10100 = Input Capture 1
10011 = Reserved
10010 = Reserved
1000x = Reserved
01111 = Timer5
01110 = Timer4
01101 = Timer3
01100 = Timer2
01011 = Timer1
01010 = Input Capture 5
01001 = Reserved
01000 = Reserved
00111 = Reserved
00110 = Reserved
00101 = Output Compare 5
00100 = Output Compare 4
00011 = Output Compare 3
00010 = Output Compare 2
00001 = Output Compare 1
00000 = Not synchronized to any other module

Note 1: Use these inputs as trigger sources only and never as sync sources.

14.0 OUTPUT COMPARE WITH DEDICATED TIMERS

Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 35. "Output Capture with Dedicated Timer"** (DS39723).

All devices in the PIC24FJ64GA104 family features 5 independent output compare modules. Each of these modules offers a wide range of configuration and operating options for generating pulse trains on internal device events, and can produce Pulse-Width Modulated (PWM) waveforms for driving power applications.

Key features of the output compare module include:

- Hardware-configurable for 32-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with up to 21 user-selectable trigger/sync sources available
- Two separate Period registers (a main register, OCxR, and a secondary register, OCxRS) for greater flexibility in generating pulses of varying widths
- Configurable for single pulse or continuous pulse generation on an output event or continuous PWM waveform generation
- Up to 6 clock sources available for each module, driving a separate internal 16-bit counter

14.1 General Operating Modes

14.1.1 SYNCHRONOUS AND TRIGGER MODES

By default, the output compare module operates in a Free-Running mode. The internal 16-bit counter, OCxTMR, runs counts up continuously, wrapping around from FFFFh to 0000h on each overflow with its period synchronized to the selected external clock source. Compare or PWM events are generated each time a match between the internal counter and one of the Period registers occurs.

In Synchronous mode, the module begins performing its compare or PWM operation as soon as its selected clock source is enabled. Whenever an event occurs on the selected sync source, the module's internal counter is reset. In Trigger mode, the module waits for a sync event from another internal module to occur before allowing the counter to run.

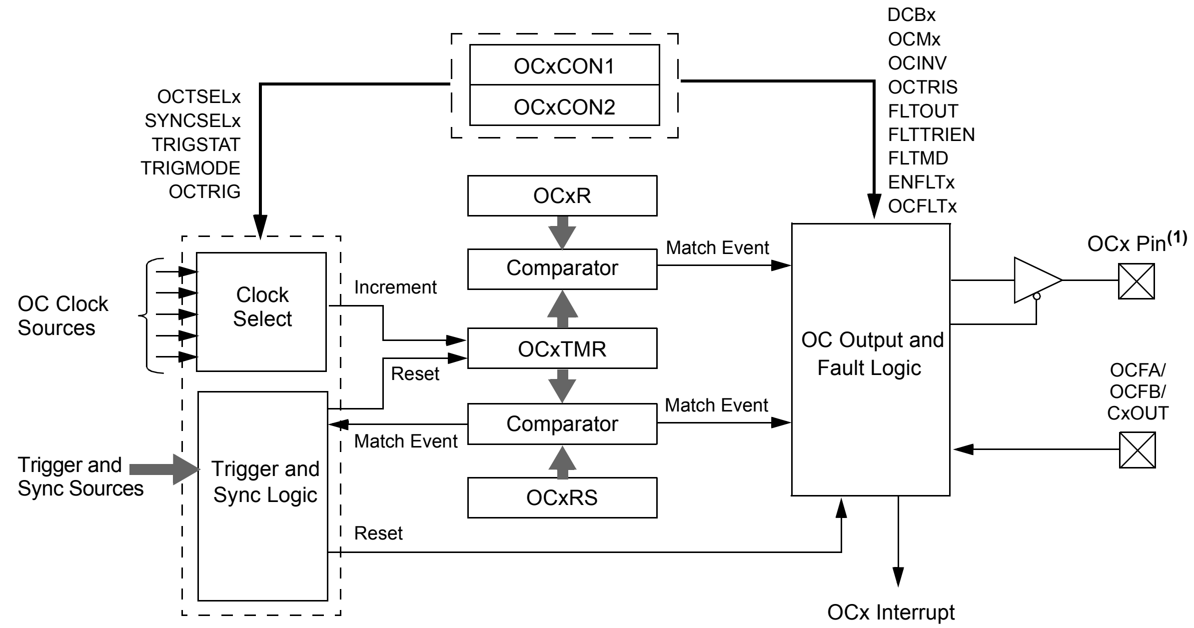
Free-Running mode is selected by default or any time that the SYNCSEL bits (OCxCON2<4:0>) are set to '00000'. Synchronous or Trigger modes are selected any time the SYNCSEL bits are set to any value except '00000'. The OCTRIG bit (OCxCON2<7>) selects either Synchronous or Trigger mode; setting the bit selects Trigger mode operation. In both modes, the SYNCSEL bits determine the sync/trigger source.

14.1.2 CASCADED (32-BIT) MODE

By default, each module operates independently with its own set of 16-bit Timer and Duty Cycle registers. To increase the range, adjacent even and odd modules can be configured to function as a single 32-bit module. (For example, Modules 1 and 2 are paired, as are Modules 3 and 4, and so on.) The odd-numbered module (OCx) provides the Least Significant 16 bits of the 32-bit register pairs and the even-numbered module (OCy) provides the Most Significant 16 bits. Wrap-arounds of the OCx registers cause an increment of their corresponding OCy registers.

Cascaded operation is configured in hardware by setting the OC32 bit (OCxCON2<8>) for both modules.

FIGURE 14-1: OUTPUT COMPARE BLOCK DIAGRAM (16-BIT MODE)



Note 1: The OCx outputs must be assigned to an available RPn pin before use. Please see **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

14.2 Compare Operations

In Compare mode (Figure 14-1), the output compare module can be configured for single-shot or continuous pulse generation; it can also repeatedly toggle an output pin on each timer event.

To set up the module for compare operations:

1. Configure the OCx output for one of the available Peripheral Pin Select pins.
2. Calculate the required values for the OCxR and (for Double Compare modes) OCxRS Duty Cycle registers:
   a) Determine the instruction clock cycle time. Take into account the frequency of the external clock to the timer source (if one is used) and the timer prescaler settings.
   b) Calculate time to the rising edge of the output pulse relative to the timer start value (0000h).
   c) Calculate the time to the falling edge of the pulse based on the desired pulse width and the time to the rising edge of the pulse.
3. Write the rising edge value to OCxR and the falling edge value to OCxRS.
4. For Trigger mode operations, set OCTRIG to enable Trigger mode. Set or clear TRIGMODE to configure trigger operation and TRIGSTAT to select a hardware or software trigger. For Synchronous mode, clear OCTRIG.
5. Set the SYNCSEL<4:0> bits to configure the trigger or synchronization source. If free-running timer operation is required, set the SYNCSEL bits to '00000' (no sync/trigger source).
6. Select the time base source with the OCTSEL<2:0> bits. If the desired clock source is running, set the OCTSEL<2:0> bits before the output compare module is enabled for proper synchronization with the desired clock source. If necessary, set the TON bit for the selected timer which enables the compare time base to count. Synchronous mode operation starts as soon as the synchronization source is enabled. Trigger mode operation starts after a trigger source event occurs.
7. Set the OCM<2:0> bits for the appropriate compare operation (= 0xx).

For 32-bit cascaded operation, these steps are also necessary:

1. Set the OC32 bits for both registers (OCyCON2<8> and (OCxCON2<8>). Enable the even-numbered module first to ensure the modules will start functioning in unison.
2. Clear the OCTRIG bit of the even module (OCyCON2), so the module will run in Synchronous mode.
3. Configure the desired output and Fault settings for OCy.
4. Force the output pin for OCx to the output state by clearing the OCTRIS bit.
5. If Trigger mode operation is required, configure the trigger options in OCx by using the OCTRIG (OCxCON2<7>), TRIGSTAT (OCxCON2<6>) and SYNCSEL (OCxCON2<4:0>) bits.
6. Configure the desired Compare or PWM mode of operation (OCM<2:0>) for OCy first, then for OCx.

Depending on the output mode selected, the module holds the OCx pin in its default state and forces a transition to the opposite state when OCxR matches the timer. In Double Compare modes, OCx is forced back to its default state when a match with OCxRS occurs. The OCxIF interrupt flag is set after an OCxR match in Single Compare modes and after each OCxRS match in Double Compare modes.

Single-shot pulse events only occur once, but may be repeated by simply rewriting the value of the OCxCON1 register. Continuous pulse events continue indefinitely until terminated.

14.3 Pulse-Width Modulation (PWM) Mode

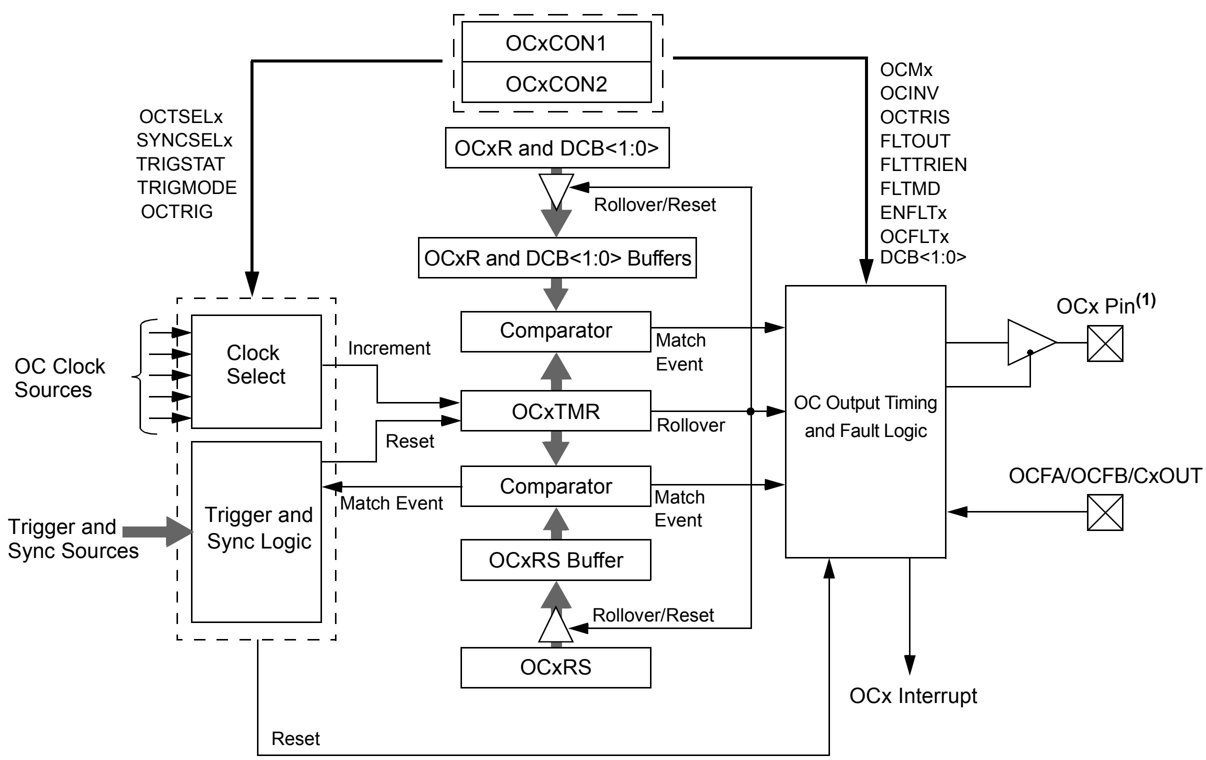
In PWM mode, the output compare module can be configured for edge-aligned or center-aligned pulse waveform generation. All PWM operations are double-buffered (buffer registers are internal to the module and are not mapped into SFR space).

To configure the output compare module for edge-aligned PWM operation:

1. Configure the OCx output for one of the available Peripheral Pin Select pins.
2. Calculate the desired on-time and load it into the OCxR register.
3. Calculate the desired period and load it into the OCxRS register.
4. Select the current OCx as the synchronization source by writing 0x1F to SYNCSEL<4:0> (OCxCON2<4:0>) and '0' to OCTRIG (OCxCON2<7>).
5. Select a clock source by writing to the OCTSEL2<2:0> (OCxCON1<12:10>) bits.
6. Enable interrupts, if required, for the timer and output compare modules. The output compare interrupt is required for PWM Fault pin utilization.
7. Select the desired PWM mode in the OCM<2:0> (OCxCON1<2:0>) bits.
8. If a timer is selected as a clock source, set the TMRy prescale value and enable the time base by setting the TON (TxCON<15>) bit.

> **Note:** This peripheral contains input and output functions that may need to be configured by the Peripheral Pin Select. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

FIGURE 14-2: OUTPUT COMPARE BLOCK DIAGRAM (DOUBLE-BUFFERED, 16-BIT PWM MODE)



Note 1: The OCx outputs must be assigned to an available RPn pin before use. Please see **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

14.3.1 PWM PERIOD

In edge aligned PWM mode, the period is specified by the value of OCxRS register. In center aligned PWM mode, the period of the synchronization source such as Timer's PRy specifies the period. The period in both cases can be calculated using Equation 14-1.

EQUATION 14-1: CALCULATING THE PWM PERIOD(1)

PWM Period = [Value + 1] x TCY x (Prescaler Value)

Where: Value = OCxRS in Edge-Aligned PWM mode and can be PRy in Center-Aligned PWM mode (If TMRy is the sync source).

Note 1: Based on TCY = TOSC * 2; Doze mode and PLL are disabled.

14.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the OCxRS and OCxR registers. The OCxRS and OCxR registers can be written to at any time, but the duty cycle value is not latched until a period is complete. This provides a double buffer for the PWM duty cycle and is essential for glitchless PWM operation.

Some important boundary parameters of the PWM duty cycle include:

- Edge-Aligned PWM
  - If OCxR and OCxRS are loaded with 0000h, the OCx pin will remain low (0% duty cycle).
  - If OCxRS is greater than OCxR, the pin will remain high (100% duty cycle).
- Center-Aligned PWM (with TMRy as the sync source)
  - If OCxR, OCxRS and PRy are all loaded with 0000h, the OCx pin will remain low (0% duty cycle).
  - If OCxRS is greater than PRy, the pin will go high (100% duty cycle).

See Example 14-1 for PWM mode timing details. Table 14-1 and Table 14-2 show example PWM frequencies and resolutions for a device operating at 4 MIPS and 10 MIPS, respectively.

EQUATION 14-2: CALCULATION FOR MAXIMUM PWM RESOLUTION(1)

$$\text{Maximum PWM Resolution (bits)} = \frac{\log_{10}\left(\frac{F_{CY}}{F_{PWM} \cdot (\text{Prescale Value})}\right)}{\log_{10}(2)} \text{ bits}$$

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

EXAMPLE 14-1: PWM PERIOD AND DUTY CYCLE CALCULATIONS(1)

1. Find the OCxRS register value for a desired PWM frequency of 52.08 kHz, where FOSC = 8 MHz with PLL (32 MHz device clock rate) and a prescaler setting of 1:1 using Edge-Aligned PWM mode.

   TCY = 2 * TOSC = 62.5 ns
   PWM Period = 1/PWM Frequency = 1/52.08 kHz = 19.2 µs
   PWM Period = (OCxRS + 1) • TCY • (OCx Prescale Value)
   19.2 µs = (OCxRS + 1) • 62.5 ns • 1
   OCxRS = 306

2. Find the maximum resolution of the duty cycle that can be used with a 52.08 kHz frequency and a 32 MHz device clock rate:

   $\text{PWM Resolution} = \log_{10}(F_{CY}/F_{PWM})/\log_{10}2) \text{ bits}$
   $= (\log_{10}(16 \text{ MHz}/52.08 \text{ kHz})/\log_{10}2) \text{ bits}$
   $= 8.3 \text{ bits}$

Note 1: Based on TCY = 2 * TOSC; Doze mode and PLL are disabled.

14.4 Subcycle Resolution

The DCB bits (OCxCON2<10:9>) provide for resolution better than one instruction cycle. When used, they delay the falling edge generated by a match event by a portion of an instruction cycle.

For example, setting DCB<1:0> = `10` causes the falling edge to occur half way through the instruction cycle in which the match event occurs, instead of at the beginning. These bits cannot be used when OCM<2:0> = `001`. When operating the module in PWM mode (OCM<2:0> = `110` or `111`), the DCB bits will be double-buffered.

The DCB bits are intended for use with a clock source identical to the system clock. When an OCx module with enabled prescaler is used, the falling edge delay caused by the DCB bits will be referenced to the system clock period, rather than the OCx module's period.

TABLE 14-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 4 MIPS (FCY = 4 MHz)(1)

PWM Frequency	7.6 Hz	61 Hz	122 Hz	977 Hz	3.9 kHz	31.3 kHz	125 kHz
Prescaler Ratio	8	1	1	1	1	1	1
Period Value	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
Resolution (bits)	16	16	15	12	10	7	5

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

TABLE 14-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 16 MIPS (FCY = 16 MHz)(1)

PWM Frequency	30.5 Hz	244 Hz	488 Hz	3.9 kHz	15.6 kHz	125 kHz	500 kHz
Prescaler Ratio	8	1	1	1	1	1	1
Period Value	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
Resolution (bits)	16	16	15	12	10	7	5

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

REGISTER 14-1: OCxCON1: OUTPUT COMPARE x CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2[2]	ENFLT1
bit 15							bit 8

R/W-0	R/W-0, HCS	R/W-0, HCS	R/W-0, HCS	R/W-0	R/W-0	R/W-0	R/W-0
ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2[1]	OCM1[1]	OCM0[1]
bit 7							bit 0

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

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **OCSIDL:** Stop Output Compare x in Idle Mode Control bit
1 = Output compare x halts in CPU Idle mode
0 = Output compare x continues to operate in CPU Idle mode

bit 12-10 **OCTSEL<2:0>:** Output Compare x Timer Select bits
111 = System clock
110 = Reserved
101 = Reserved
100 = Timer1
011 = Timer5
010 = Timer4
001 = Timer3
000 = Timer2

bit 9 **ENFLT2:** Comparator Fault Input Enable bit[2]
1 = Comparator Fault input is enabled
0 = Comparator Fault input is disabled

bit 8 **ENFLT1:** OCFB Fault Input Enable bit
1 = OCFB Fault input is enabled
0 = OCFB Fault input is disabled

bit 7 **ENFLT0:** OCFA Fault Input Enable bit
1 = OCFA Fault input is enabled
0 = OCFA Fault input is disabled

bit 6 **OCFLT2:** PWM Comparator Fault Condition Status bit[2]
1 = PWM comparator Fault condition has occurred (this is cleared in hardware only)
0 = PWM comparator Fault condition has not occurred (this bit is used only when OCM<2:0> = 111)

bit 5 **OCFLT1:** PWM OCFB Fault Input Enable bit
1 = PWM OCFB Fault condition has occurred (this is cleared in hardware only)
0 = PWM OCFB Fault condition has not occurred (this bit is used only when OCM<2:0> = 111)

bit 4 **OCFLT0:** PWM OCFA Fault Condition Status bit
1 = PWM OCFA Fault condition has occurred (this is cleared in hardware only)
0 = PWM OCFA Fault condition has not occurred (this bit is used only when OCM<2:0> = 111)

bit 3 **TRIGMODE:** Trigger Status Mode Select bit
1 = TRIGSTAT (OCxCON2<6>) is cleared when OCxRS = OCxTMR or in software
0 = TRIGSTAT is only cleared by software

Note 1: The OCx output must also be configured to an available RPn pin. For more information, see **Section 10.4 "Peripheral Pin Select (PPS)"**.

2: The comparator module used for Fault input varies with the OCx module. OC1 and OC2 use Comparator 1; OC3 and OC4 use Comparator 2; OC5 uses Comparator 3.

REGISTER 14-1: OCxCON1: OUTPUT COMPARE x CONTROL REGISTER 1 (CONTINUED)

bit 2-0 **OCM<2:0>:** Output Compare x Mode Select bits(1)

111 = Center-Aligned PWM mode on OCx
110 = Edge-Aligned PWM mode on OCx
101 = Double Compare Continuous Pulse mode: initialize OCx pin low, toggle OCx state continuously on alternate matches of OCxR and OCxRS
100 = Double Compare Single-Shot mode: initialize OCx pin low, toggle OCx state on matches of OCxR and OCxRS for one cycle
011 = Single Compare Continuous Pulse mode: compare events continuously toggle OCx pin
010 = Single Compare Single-Shot mode: initialize OCx pin high, compare event forces OCx pin low
001 = Single Compare Single-Shot mode: initialize OCx pin low, compare event forces OCx pin high
000 = Output compare channel is disabled

Note 1: The OCx output must also be configured to an available RPn pin. For more information, see **Section 10.4 "Peripheral Pin Select (PPS)"**.

2: The comparator module used for Fault input varies with the OCx module. OC1 and OC2 use Comparator 1; OC3 and OC4 use Comparator 2; OC5 uses Comparator 3.

REGISTER 14-2: OCxCON2: OUTPUT COMPARE x CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1(3)	DCB0(3)	OC32
bit 15							bit 8

R/W-0	R/W-0, HS	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0
OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7							bit 0

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

bit 15 **FLTMD:** Fault Mode Select bit
1 = Fault mode is maintained until the Fault source is removed and the corresponding OCFLT0 bit is cleared in software
0 = Fault mode is maintained until the Fault source is removed and a new PWM period starts

bit 14 **FLTOUT:** Fault Out bit
1 = PWM output is driven high on a Fault
0 = PWM output is driven low on a Fault

bit 13 **FLTTRIEN:** Fault Output State Select bit
1 = Pin is forced to an output on a Fault condition
0 = Pin I/O condition is unaffected by a Fault

bit 12 **OCINV:** OCMP Invert bit
1 = OCx output is inverted
0 = OCx output is not inverted

bit 11 **Unimplemented:** Read as ‘0’

bit 10-9 **DCB<1:0>:** OC Pulse-Width Least Significant bits(3)
11 = Delay OCx falling edge by 3/4 of the instruction cycle
10 = Delay OCx falling edge by 1/2 of the instruction cycle
01 = Delay OCx falling edge by 1/4 of the instruction cycle
00 = OCx falling edge occurs at start of the instruction cycle

bit 8 **OC32:** Cascade Two OC Modules Enable bit (32-bit operation)
1 = Cascade module operation enabled
0 = Cascade module operation disabled

bit 7 **OCTRIG:** OCx Trigger/Sync Select bit
1 = Trigger OCx from source designated by SYNCSELx bits
0 = Synchronize OCx with source designated by SYNCSELx bits

bit 6 **TRIGSTAT:** Timer Trigger Status bit
1 = Timer source has been triggered and is running
0 = Timer source has not been triggered and is being held clear

bit 5 **OCTRIS:** OCx Output Pin Direction Select bit
1 = OCx pin is tri-stated
0 = Output compare peripheral x connected to OCx pin

Note 1: Do not use an OC module as its own trigger source, either by selecting this mode or another equivalent SYNCSEL setting.
2: Use these inputs as trigger sources only and never as sync sources.
3: These bits affect the rising edge when OCINV = 1. The bits have no effect when the OCM bits (OCxCON1<1:0>) = 001.

REGISTER 14-2: OCxCON2: OUTPUT COMPARE x CONTROL REGISTER 2 (CONTINUED)

bit 4-0 **SYNCSEL<4:0>:** Trigger/Synchronization Source Selection bits

11111 = This OC module(1)
11110 = Reserved
11101 = Reserved
11100 = CTMU(2)
11011 = A/D(2)
11010 = Comparator 3(2)
11001 = Comparator 2(2)
11000 = Comparator 1(2)
10111 = Input Capture 4(2)
10110 = Input Capture 3(2)
10101 = Input Capture 2(2)
10100 = Input Capture 1(2)
100xx = Reserved
01111 = Timer5
01110 = Timer4
01101 = Timer3
01100 = Timer2
01011 = Timer1
01010 = Input Capture 5(2)
01001 = Reserved
01000 = Reserved
00111 = Reserved
00110 = Reserved
00101 = Output Compare 5(1)
00100 = Output Compare 4(1)
00011 = Output Compare 3(1)
00010 = Output Compare 2(1)
00001 = Output Compare 1(1)
00000 = Not synchronized to any other module

Note 1: Do not use an OC module as its own trigger source, either by selecting this mode or another equivalent SYNCSEL setting.

2: Use these inputs as trigger sources only and never as sync sources.

3: These bits affect the rising edge when OCINV = 1. The bits have no effect when the OCM bits (OCxCON1<1:0>) = 001.

15.0 SERIAL PERIPHERAL INTERFACE (SPI)

> **Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 23. "Serial Peripheral Interface (SPI)"** (DS39699).

The Serial Peripheral Interface (SPI) module is a synchronous 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 SPI module is compatible with Motorola® SPI and SIOP interfaces. All devices of the PIC24FJ64GA104 family include three SPI modules

The module supports operation in two buffer modes. In Standard mode, data is shifted through a single serial buffer. In Enhanced Buffer mode, data is shifted through an 8-level FIFO buffer.

> **Note:** Do not perform read-modify-write operations (such as bit-oriented instructions) on the SPIxBUF register in either Standard or Enhanced Buffer mode.

The module also supports a basic framed SPI protocol while operating in either Master or Slave mode. A total of four framed SPI configurations are supported.

The SPI serial interface consists of four pins:

- SDIx: Serial Data Input
- SDOx: Serial Data Output
- SCKx: Shift Clock Input or Output
- $\overline{\text{SSx}}$: Active-Low Slave Select or Frame Synchronization I/O Pulse

The SPI module can be configured to operate using 2, 3 or 4 pins. In the 3-pin mode, $\overline{\text{SSx}}$ is not used. In the 2-pin mode, both SDOx and $\overline{\text{SSx}}$ are not used.

Block diagrams of the module in Standard and Enhanced modes are shown in Figure 15-1 and Figure 15-2.

> **Note:** In this section, the SPI modules are referred to together as SPIx or separately as SPI1, SPI2 or SPI3. Special Function Registers will follow a similar notation. For example, SPIxCON1 and SPIxCON2 refer to the control registers for any of the 3 SPI modules.

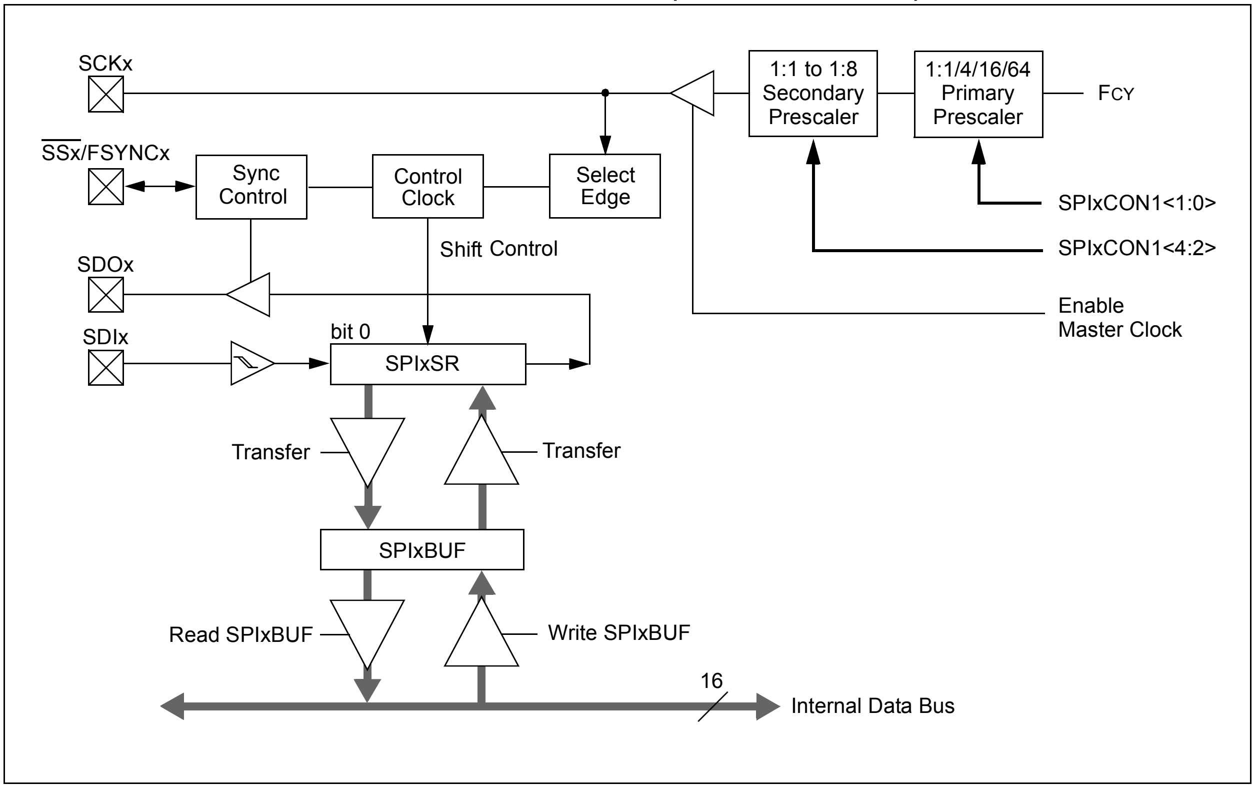
To set up the SPI module for the Standard Master mode of operation:

1. If using interrupts:
   a) Clear the SPIxIF bit in the respective IFS register.
   b) Set the SPIxIE bit in the respective IEC register.
   c) Write the SPIxIP bits in the respective IPC register to set the interrupt priority.
2. Write the desired settings to the SPIxCON1 and SPIxCON2 registers with MSTEN (SPIxCON1<5>) = 1.
3. Clear the SPIROV bit (SPIxSTAT<6>).
4. Enable SPI operation by setting the SPIEN bit (SPIxSTAT<15>).
5. Write the data to be transmitted to the SPIxBUF register. Transmission (and reception) will start as soon as data is written to the SPIxBUF register.

To set up the SPI module for the Standard Slave mode of operation:

1. Clear the SPIxBUF register.
2. If using interrupts:
   a) Clear the SPIxIF bit in the respective IFS register.
   b) Set the SPIxIE bit in the respective IEC register.
   c) Write the SPIxIP bits in the respective IPC register to set the interrupt priority.
3. Write the desired settings to the SPIxCON1 and SPIxCON2 registers with MSTEN (SPIxCON1<5>) = 0.
4. Clear the SMP bit.
5. If the CKE bit (SPIxCON1<8>) is set, then the SSEN bit (SPIxCON1<7>) must be set to enable the $\overline{SSx}$ pin.
6. Clear the SPIROV bit (SPIxSTAT<6>).
7. Enable SPI operation by setting the SPIEN bit (SPIxSTAT<15>).

FIGURE 15-1: SPIx MODULE BLOCK DIAGRAM (STANDARD MODE)



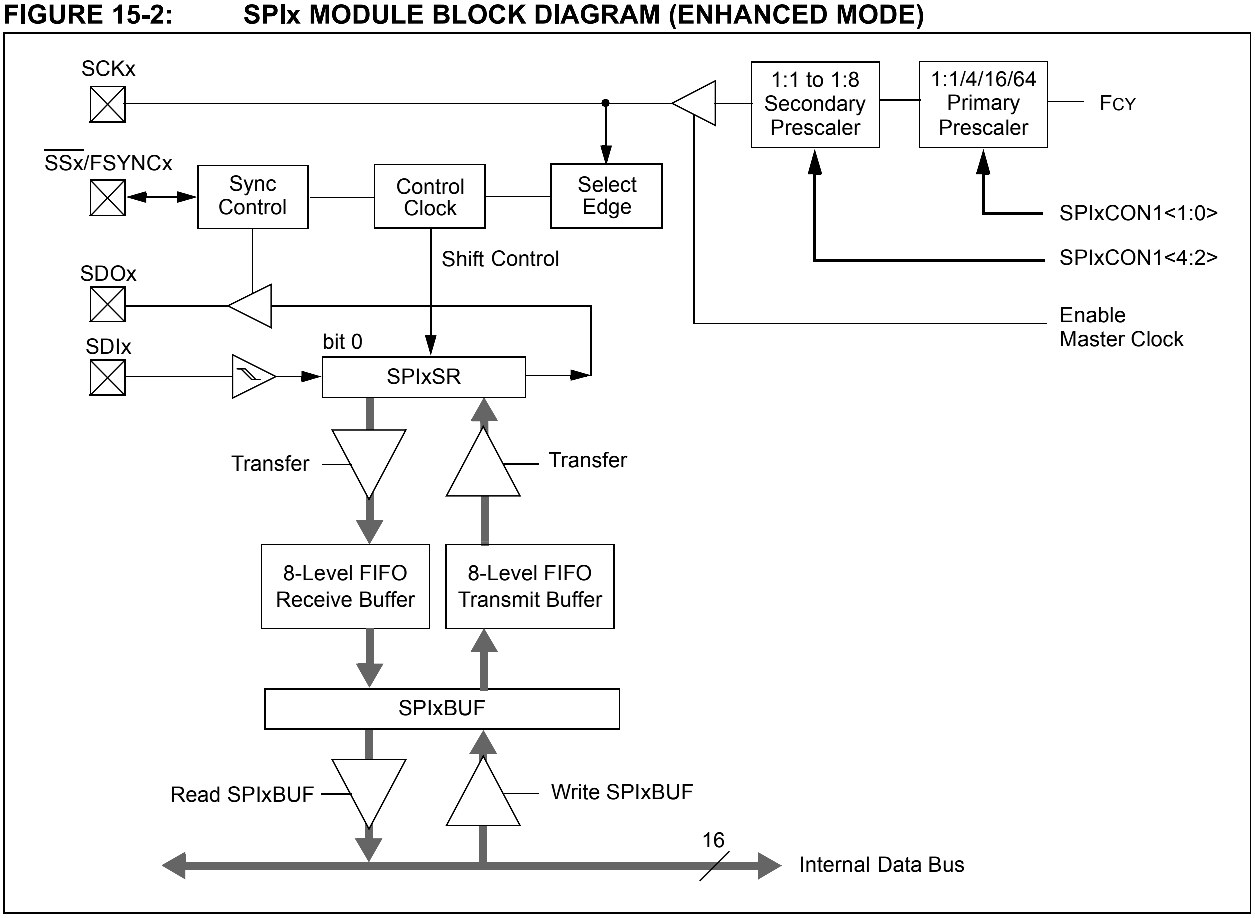
To set up the SPI module for the Enhanced Buffer Master mode of operation:

1. If using interrupts:
   a) Clear the SPIxIF bit in the respective IFS register.
   b) Set the SPIxIE bit in the respective IEC register.
   c) Write the SPIxIP bits in the respective IPC register.
2. Write the desired settings to the SPIxCON1 and SPIxCON2 registers with MSTEN (SPIxCON1<5>) = `1`.
3. Clear the SPIROV bit (SPIxSTAT<6>).
4. Select Enhanced Buffer mode by setting the SPIBEN bit (SPIxCON2<0>).
5. Enable SPI operation by setting the SPIEN bit (SPIxSTAT<15>).
6. Write the data to be transmitted to the SPIxBUF register. Transmission (and reception) will start as soon as data is written to the SPIxBUF register.

To set up the SPI module for the Enhanced Buffer Slave mode of operation:

1. Clear the SPIxBUF register.
2. If using interrupts:
   a) Clear the SPIxIF bit in the respective IFS register.
   b) Set the SPIxIE bit in the respective IEC register.
   c) Write the SPIxIP bits in the respective IPC register to set the interrupt priority.
3. Write the desired settings to the SPIxCON1 and SPIxCON2 registers with MSTEN (SPIxCON1<5>) = `0`.
4. Clear the SMP bit.
5. If the CKE bit is set, then the SSEN bit must be set, thus enabling the $\overline{SSx}$ pin.
6. Clear the SPIROV bit (SPIxSTAT<6>).
7. Select Enhanced Buffer mode by setting the SPIBEN bit (SPIxCON2<0>).
8. Enable SPI operation by setting the SPIEN bit (SPIxSTAT<15>).

FIGURE 15-2: SPIx MODULE BLOCK DIAGRAM (ENHANCED MODE)



REGISTER 15-1: SPIxSTAT: SPIx STATUS AND CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	R-0	R-0	R-0
SPIEN(1)	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0
bit 15							bit 8

R-0	R/C-0, HS	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0
SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF
bit 7							bit 0

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

bit 15 **SPIEN:** SPIx Enable bit(1)
1 = Enables module and configures SCKx, SDOx, SDIx and $\overline{SSx}$ as serial port pins
0 = Disables module

bit 14 **Unimplemented:** Read as '0'

bit 13 **SPISIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-11 **Unimplemented:** Read as '0'

bit 10-8 **SPIBEC<2:0>:** SPIx Buffer Element Count bits (valid in Enhanced Buffer mode)
Master mode:
Number of SPI transfers that are pending.
Slave mode:
Number of SPI transfers that are unread.

bit 7 **SRMPT:** Shift Register (SPIxSR) Empty bit (valid in Enhanced Buffer mode)
1 = SPIx Shift register is empty and ready to send or receive
0 = SPIx Shift register is not empty

bit 6 **SPIROV:** Receive Overflow Flag bit
1 = A new byte/word is completely received and discarded. The user software has not read the previous data in the SPIxBUF register.
0 = No overflow has occurred

bit 5 **SRXMPT:** Receive FIFO Empty bit (valid in Enhanced Buffer mode)
1 = Receive FIFO is empty
0 = Receive FIFO is not empty

bit 4-2 **SISEL<2:0>:** SPIx Buffer Interrupt Mode bits (valid in Enhanced Buffer mode)
111 = Interrupt when SPIx transmit buffer is full (SPITBF bit is set)
110 = Interrupt when last bit is shifted into SPIxSR; as a result, the TX FIFO is empty
101 = Interrupt when the last bit is shifted out of SPIxSR; now the transmit is complete
100 = Interrupt when one data is shifted into the SPIxSR; as a result, the TX FIFO has one open spot
011 = Interrupt when SPIx receive buffer is full (SPIRBF bit is set)
010 = Interrupt when SPIx receive buffer is 3/4 or more full
001 = Interrupt when data is available in the receive buffer (SRMPT bit is set)
000 = Interrupt when the last data in the receive buffer is read; as a result, the buffer is empty (SRXMPT bit set)

Note 1: If SPIEN = 1, these functions must be assigned to available RPn pins before use. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

REGISTER 15-1: SPIxSTAT: SPIx STATUS AND CONTROL REGISTER (CONTINUED)

bit 1 **SPITBF:** SPIx Transmit Buffer Full Status bit
1 = Transmit not yet started; SPIxTXB is full
0 = Transmit started; SPIxTXB is empty
In Standard Buffer mode:
Automatically set in hardware when CPU writes SPIxBUF location, loading SPIxTXB. Automatically cleared in hardware when SPIx module transfers data from SPIxTXB to SPIxSR.
In Enhanced Buffer mode:
Automatically set in hardware when CPU writes SPIxBUF location, loading the last available buffer location. Automatically cleared in hardware when a buffer location is available for a CPU write.

bit 0 **SPIRBF:** SPIx Receive Buffer Full Status bit
1 = Receive is complete, SPIxRXB is full
0 = Receive is not complete, SPIxRXB is empty
In Standard Buffer mode:
Automatically set in hardware when SPIx transfers data from SPIxSR to SPIxRXB. Automatically cleared in hardware when core reads SPIxBUF location, reading SPIxRXB.
In Enhanced Buffer mode:
Automatically set in hardware when SPIx transfers data from SPIxSR to buffer, filling the last unread buffer location. Automatically cleared in hardware when a buffer location is available for a transfer from SPIxSR.

Note 1: If SPIEN = 1, these functions must be assigned to available RPn pins before use. See **Section 10.4 “Peripheral Pin Select (PPS)”** for more information.

REGISTER 15-2: SPIxCON1: SPIx CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DISSCK[(1)]	DISSDO[(2)]	MODE16	SMP	CKE[(3)]
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSEN[(4)]	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as '0'

bit 12 **DISSCK:** Disable SCKx pin bit (SPI Master modes only)[(1)]
1 = Internal SPI clock is disabled; pin functions as I/O
0 = Internal SPI clock is enabled

bit 11 **DISSDO:** Disable SDOx pin bit[(2)]
1 = SDOx pin is not used by module; pin functions as I/O
0 = SDOx pin is controlled by the module

bit 10 **MODE16:** Word/Byte Communication Select bit
1 = Communication is word-wide (16 bits)
0 = Communication is byte-wide (8 bits)

bit 9 **SMP:** SPIx Data Input Sample Phase bit
Master mode:
1 = Input data is sampled at the end of data output time
0 = Input data is sampled at the middle of data output time
Slave mode:
SMP must be cleared when SPIx is used in Slave mode.

bit 8 **CKE:** SPIx Clock Edge Select bit[(3)]
1 = Serial output data changes on transition from active clock state to Idle clock state (see bit 6)
0 = Serial output data changes on transition from Idle clock state to active clock state (see bit 6)

bit 7 **SSEN:** Slave Select Enable (Slave mode) bit[(4)]
1 = $\overline{SSx}$ pin is used for Slave mode
0 = $\overline{SSx}$ pin is not used by module; pin is controlled by port function

bit 6 **CKP:** Clock Polarity Select bit
1 = Idle state for clock is a high level; active state is a low level
0 = Idle state for clock is a low level; active state is a high level

bit 5 **MSTEN:** Master Mode Enable bit
1 = Master mode
0 = Slave mode

Note 1: If DISSCK = 0, SCKx must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

2: If DISSDO = 0, SDOx must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

3: The CKE bit is not used in the Framed SPI modes. The user should program this bit to '0' for the Framed SPI modes (FRMEN = 1).

4: If SSEN = 1, $\overline{SSx}$ must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

REGISTER 15-2: SPIxCON1: SPIx CONTROL REGISTER 1 (CONTINUED)

bit 4-2 **SPRE<2:0>:** Secondary Prescale bits (Master mode)
111 = Secondary prescale 1:1
110 = Secondary prescale 2:1
. . .
000 = Secondary prescale 8:1

bit 1-0 **PPRE<1:0>:** Primary Prescale bits (Master mode)
11 = Primary prescale 1:1
10 = Primary prescale 4:1
01 = Primary prescale 16:1
00 = Primary prescale 64:1

Note 1: If DISSCK = 0, SCKx must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.
2: If DISSDO = 0, SDOx must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.
3: The CKE bit is not used in the Framed SPI modes. The user should program this bit to '0' for the Framed SPI modes (FRMEN = 1).
4: If SSEN = 1, $\overline{\text{SSx}}$ must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

REGISTER 15-3: SPIxCON2: SPIx CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	SPIFE	SPIBEN
bit 7							bit 0

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

bit 15 **FRMEN:** Framed SPIx Support bit
1 = Framed SPIx support is enabled
0 = Framed SPIx support is disabled

bit 14 **SPIFSD:** Frame Sync Pulse Direction Control on $\overline{\text{SSx}}$ Pin bit
1 = Frame sync pulse input (slave)
0 = Frame sync pulse output (master)

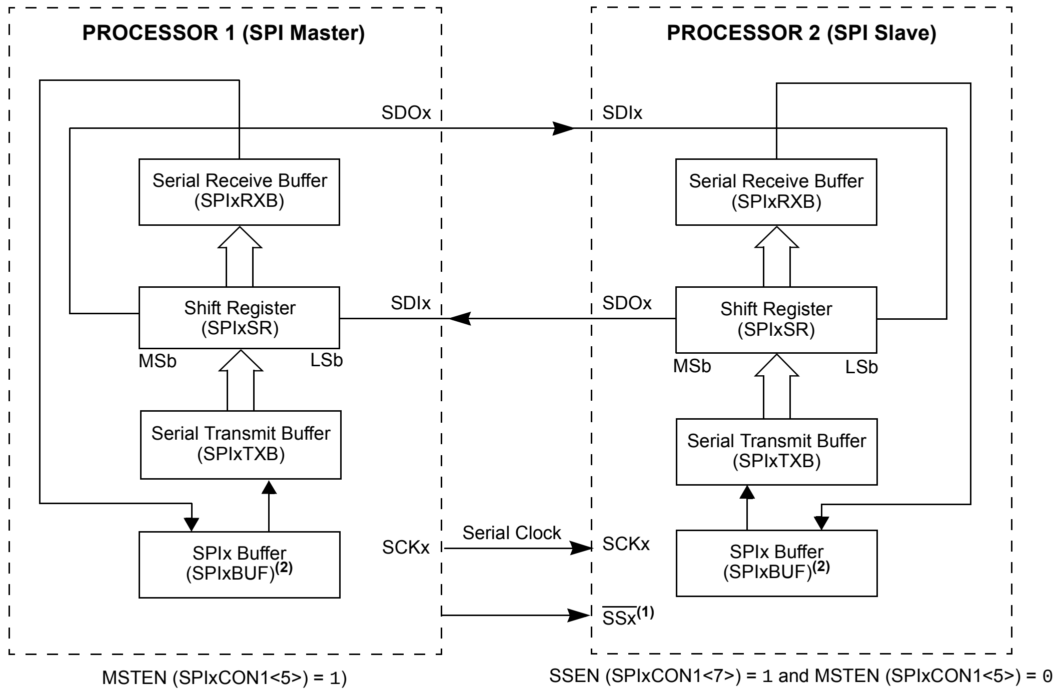
bit 13 **SPIFPOL:** Frame Sync Pulse Polarity bit (Frame mode only)
1 = Frame sync pulse is active-high
0 = Frame sync pulse is active-low

bit 12-2 **Unimplemented:** Read as '0'

bit 1 **SPIFE:** Frame Sync Pulse Edge Select bit
1 = Frame sync pulse coincides with the first bit clock
0 = Frame sync pulse precedes the first bit clock

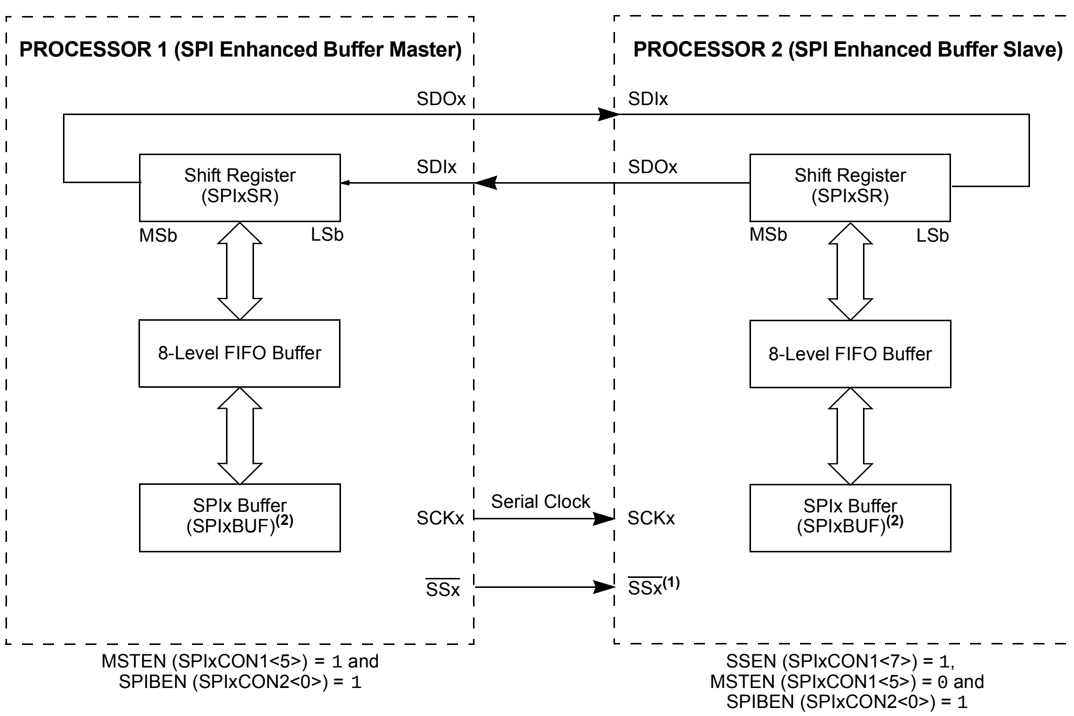
bit 0 **SPIBEN:** Enhanced Buffer Enable bit
1 = Enhanced buffer is enabled
0 = Enhanced buffer is disabled (Legacy mode)

FIGURE 15-3: SPI MASTER/SLAVE CONNECTION (STANDARD MODE)



Note 1: Using the $\overline{\text{SSx}}$ pin in Slave mode of operation is optional.
2: User must write transmit data to read received data from SPIxBUF. The SPIxTXB and SPIxRXB registers are memory mapped to SPIxBUF.

FIGURE 15-4: SPI MASTER/SLAVE CONNECTION (ENHANCED BUFFER MODES)



Note 1: Using the $\overline{\text{SSx}}$ pin in Slave mode of operation is optional.
2: User must write transmit data to read received data from SPIxBUF. The SPIxTXB and SPIxRXB registers are memory mapped to SPIxBUF.

FIGURE 15-5: SPI MASTER, FRAME MASTER CONNECTION DIAGRAM

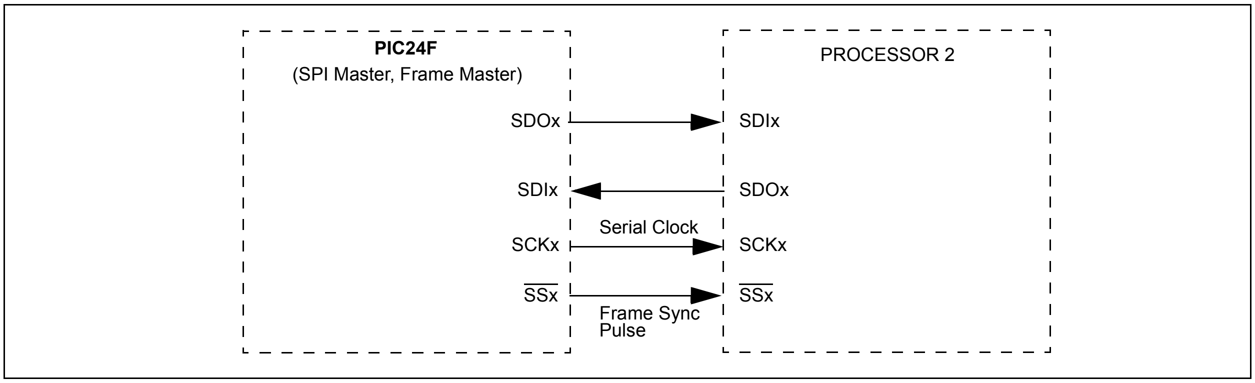


FIGURE 15-6: SPI MASTER, FRAME SLAVE CONNECTION DIAGRAM

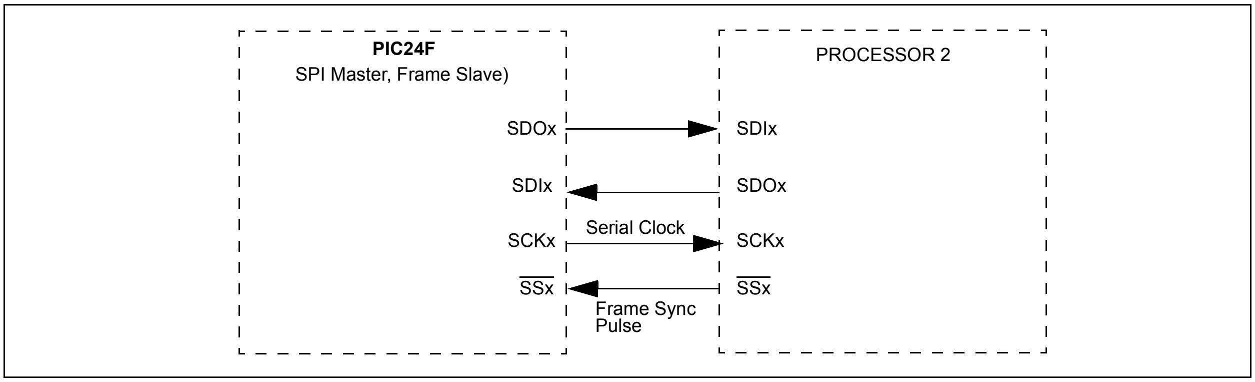


FIGURE 15-7: SPI SLAVE, FRAME MASTER CONNECTION DIAGRAM

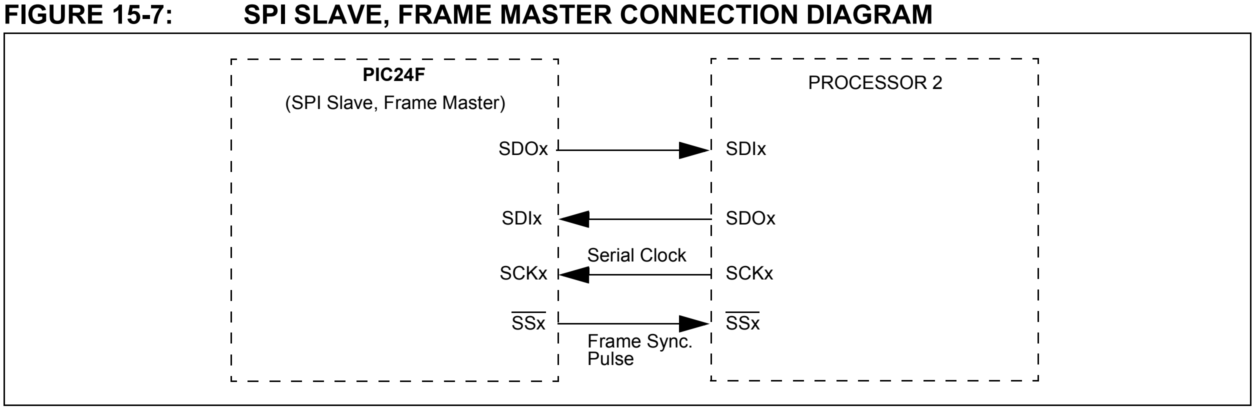
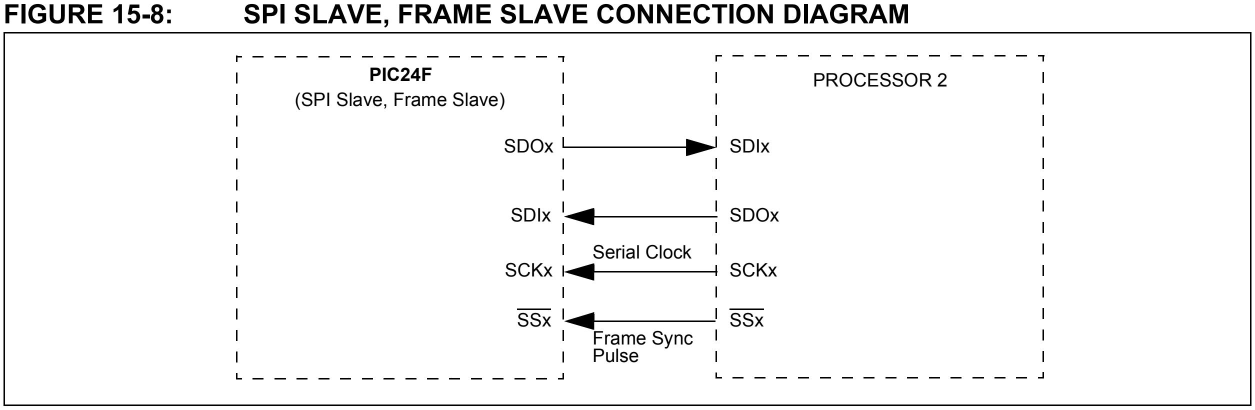


FIGURE 15-8: SPI SLAVE, FRAME SLAVE CONNECTION DIAGRAM



EQUATION 15-1: RELATIONSHIP BETWEEN DEVICE AND SPI CLOCK SPEED[1]

$$F_{SCK} = \frac{F_{CY}}{\text{Primary Prescaler * Secondary Prescaler}}$$

Note 1: Based on FCY = FOSC/2, Doze mode and PLL are disabled.

TABLE 15-1: SAMPLE SCK FREQUENCIES[1,2]

FCY = 16 MHz		Secondary Prescaler Settings				
		1:1	2:1	4:1	6:1	8:1
Primary Prescaler Settings	1:1	Invalid	8000	4000	2667	2000
	4:1	4000	2000	1000	667	500
	16:1	1000	500	250	167	125
	64:1	250	125	63	42	31
FCY = 5 MHz						
Primary Prescaler Settings	1:1	5000	2500	1250	833	625
	4:1	1250	625	313	208	156
	16:1	313	156	78	52	39
	64:1	78	39	20	13	10

Note 1: Based on FCY = FOSC/2, Doze mode and PLL are disabled.
2: SCKx frequencies are shown in kHz.

16.0 INTER-INTEGRATED CIRCUIT (I^2C™)

> **Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 24. "Inter-Integrated Circuit™ (I^2C™)"** (DS39702).

The Inter-Integrated Circuit (I^2C) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, display drivers, A/D Converters, etc.

The I^2C module supports these features:

- Independent master and slave logic
- 7-bit and 10-bit device addresses
- General call address as defined in the I^2C protocol
- Clock stretching to provide delays for the processor to respond to a slave data request
- Both 100 kHz and 400 kHz bus specifications.
- Configurable address masking
- Multi-Master modes to prevent loss of messages in arbitration
- Bus Repeater mode, allowing the acceptance of all messages as a slave regardless of the address
- Automatic SCL

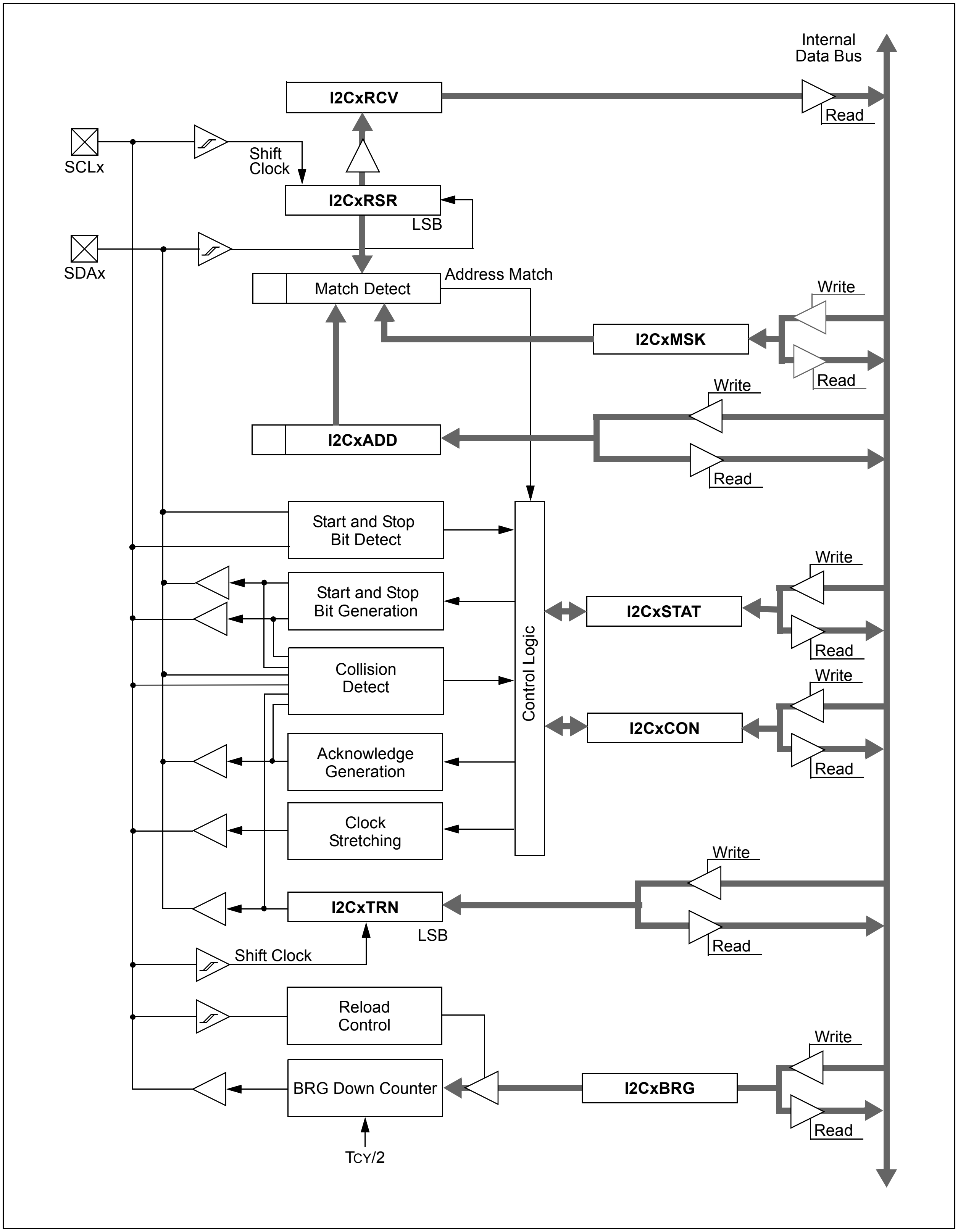
A block diagram of the module is shown in Figure 16-1.

16.1 Communicating as a Master in a Single Master Environment

The details of sending a message in Master mode depends on the communications protocol for the device being communicated with. Typically, the sequence of events is as follows:

1. Assert a Start condition on SDAx and SCLx.
2. Send the I^2C device address byte to the slave with a write indication.
3. Wait for and verify an Acknowledge from the slave.
4. Send the first data byte (sometimes known as the command) to the slave.
5. Wait for and verify an Acknowledge from the slave.
6. Send the serial memory address low byte to the slave.
7. Repeat steps 4 and 5 until all data bytes are sent.
8. Assert a Repeated Start condition on SDAx and SCLx.
9. Send the device address byte to the slave with a read indication.
10. Wait for and verify an Acknowledge from the slave.
11. Enable master reception to receive serial memory data.
12. Generate an ACK or NACK condition at the end of a received byte of data.
13. Generate a Stop condition on SDAx and SCLx.

FIGURE 16-1: I2C™ BLOCK DIAGRAM



16.2 Setting Baud Rate When Operating as a Bus Master

To compute the Baud Rate Generator (BRG) reload value, use Equation 16-1.

EQUATION 16-1: COMPUTING BAUD RATE RELOAD VALUE(1,2)

$$F_{SCL} = \frac{F_{CY}}{I2CxBRG + 1 + \frac{F_{CY}}{10,000,000}}$$

or

$$I2CxBRG = \left(\frac{F_{CY}}{F_{SCL}} - \frac{F_{CY}}{10,000,000}\right) - 1$$

Note 1: Based on FCY = FOSC/2, Doze mode and PLL are disabled.

2: These clock rate values are for guidance only. The actual clock rate can be affected by various system level parameters. The actual clock rate should be measured in its intended application.

16.3 Slave Address Masking

The I2CxMSK register (Register 16-3) designates address bit positions as "don't care" for both 7-Bit and 10-Bit Addressing modes. Setting a particular bit location (= `1`) in the I2CxMSK register causes the slave module to respond whether the corresponding address bit value is a '`0`' or a '`1`'. For example, when I2CxMSK is set to '`00100000`', the slave module will detect both addresses: '`0000000`' and '`0100000`'.

To enable address masking, the IPMI (Intelligent Peripheral Management Interface) must be disabled by clearing the IPMIEN bit (I2CxCON<11>).

Note: As a result of changes in the I^{2}C™ protocol, the addresses in Table 16-2 are reserved and will not be Acknowledged in Slave mode. This includes any address mask settings that include any of these addresses.

TABLE 16-1: I^{2}C™ CLOCK RATES(1,2)

Required System FSCL	FCY	I2CxBRG Value (Decimal)	I2CxBRG Value (Hexadecimal)	Actual FSCL
100 kHz	16 MHz	157	9D	100 kHz
100 kHz	8 MHz	78	4E	100 kHz
100 kHz	4 MHz	39	27	99 kHz
400 kHz	16 MHz	37	25	404 kHz
400 kHz	8 MHz	18	12	404 kHz
400 kHz	4 MHz	9	9	385 kHz
400 kHz	2 MHz	4	4	385 kHz
1 MHz	16 MHz	13	D	1.026 MHz
1 MHz	8 MHz	6	6	1.026 MHz
1 MHz	4 MHz	3	3	0.909 MHz

Note 1: Based on FCY = FOSC/2, Doze mode and PLL are disabled.

2: These clock rate values are for guidance only. The actual clock rate can be affected by various system level parameters. The actual clock rate should be measured in its intended application.

TABLE 16-2: I^{2}C™ RESERVED ADDRESSES(1)

Slave Address	R/W Bit	Description
`0000 000`	`0`	General Call Address(2)
`0000 000`	`1`	Start Byte
`0000 001`	`x`	Cbus Address
`0000 010`	`x`	Reserved
`0000 011`	`x`	Reserved
`0000 1xx`	`x`	HS Mode Master Code
`1111 1xx`	`x`	Reserved
`1111 0xx`	`x`	10-Bit Slave Upper Byte(3)

Note 1: The address bits listed here will never cause an address match, independent of address mask settings.

2: The address will be Acknowledged only if GCEN = `1`.

3: A match on this address can only occur on the upper byte in 10-Bit Addressing mode.

REGISTER 16-1: I2CxCON: I2Cx CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-1, HC	R/W-0	R/W-0	R/W-0	R/W-0
I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0, HC	R/W-0, HC	R/W-0, HC	R/W-0, HC	R/W-0, HC
GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

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

bit 15 **I2CEN:** I2Cx Enable bit
1 = Enables the I2Cx module, and configures the SDAx and SCLx pins as serial port pins
0 = Disables the I2Cx module. All I^{2}C pins are controlled by port functions.

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **I2CSIDL:** Stop in Idle Mode bit
1 = Discontinues module operation when device enters an Idle mode
0 = Continues module operation in Idle mode

bit 12 **SCLREL:** SCLx Release Control bit (when operating as I^{2}C Slave)
1 = Releases SCLx clock
0 = Holds SCLx clock low (clock stretch)
If STREN = 1:
Bit is R/$\overline{W}$ (i.e., software may write ‘0’ to initiate stretch and write ‘1’ to release clock). Hardware clear at beginning of slave transmission. Hardware clear at end of slave reception.
If STREN = 0:
Bit is R/S (i.e., software may only write ‘1’ to release clock). Hardware clear at beginning of slave transmission.

bit 11 **IPMIEN:** Intelligent Platform Management Interface (IPMI) Enable bit
1 = IPMI Support mode is enabled; all addresses Acknowledged
0 = IPMI mode is disabled

bit 10 **A10M:** 10-Bit Slave Addressing bit
1 = I2CxADD is a 10-bit slave address
0 = I2CxADD is a 7-bit slave address

bit 9 **DISSLW:** Disable Slew Rate Control bit
1 = Slew rate control is disabled
0 = Slew rate control is enabled

bit 8 **SMEN:** SMBus Input Levels bit
1 = Enables I/O pin thresholds compliant with the SMBus specification
0 = Disables the SMBus input thresholds

bit 7 **GCEN:** General Call Enable bit (when operating as I^{2}C slave)
1 = Enables interrupt when a general call address is received in the I2CxRSR (module is enabled for reception)
0 = General call address is disabled

bit 6 **STREN:** SCLx Clock Stretch Enable bit (when operating as I^{2}C slave)
Used in conjunction with the SCLREL bit.
1 = Enables software or receive clock stretching
0 = Disables software or receive clock stretching

REGISTER 16-1: I2CxCON: I2Cx CONTROL REGISTER (CONTINUED)

bit 5	**ACKDT:** Acknowledge Data bit (When operating as I^2C master. Applicable during master receive.) Value that will be transmitted when the software initiates an Acknowledge sequence. `1` = Sends NACK during Acknowledge `0` = Sends ACK during Acknowledge
bit 4	**ACKEN:** Acknowledge Sequence Enable bit (When operating as I^2C master. Applicable during master receive.) `1` = Initiates Acknowledge sequence on SDAx and SCLx pins and transmits ACKDT data bit. Hardware clear at end of master Acknowledge sequence. `0` = Acknowledge sequence is not in progress
bit 3	**RCEN:** Receive Enable bit (when operating as I^2C master) `1` = Enables Receive mode for I^2C. Hardware clear at end of eighth bit of master receive data byte. `0` = Receive sequence is not in progress
bit 2	**PEN:** Stop Condition Enable bit (when operating as I^2C master) `1` = Initiates Stop condition on SDAx and SCLx pins. Hardware clear at end of master Stop sequence. `0` = Stop condition is not in progress
bit 1	**RSEN:** Repeated Start Condition Enabled bit (when operating as I^2C master) `1` = Initiates Repeated Start condition on SDAx and SCLx pins. Hardware clear at end of master Repeated Start sequence. `0` = Repeated Start condition is not in progress
bit 0	**SEN:** Start Condition Enabled bit (when operating as I^2C master) `1` = Initiates Start condition on SDAx and SCLx pins. Hardware clear at end of master Start sequence. `0` = Start condition is not in progress

REGISTER 16-2: I2CxSTAT: I2Cx STATUS REGISTER

R-0, HSC	R-0, HSC	U-0	U-0	U-0	R/C-0, HS	R-0, HSC	R-0, HSC
ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10
bit 15							bit 8

R/C-0, HS	R/C-0, HS	R-0, HSC	R/C-0, HSC	R/C-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
IWCOL	I2COV	D/$\overline{A}$	P	S	R/$\overline{W}$	RBF	TBF
bit 7							bit 0

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

bit 15 **ACKSTAT:** Acknowledge Status bit
1 = NACK was detected last
0 = ACK was detected last
Hardware set or clear at end of Acknowledge.

bit 14 **TRSTAT:** Transmit Status bit
(When operating as I^{2}C master. Applicable to master transmit operation.)
1 = Master transmit is in progress (8 bits + ACK)
0 = Master transmit is not in progress
Hardware set at the beginning of master transmission. Hardware clear at the end of slave Acknowledge.

bit 13-11 **Unimplemented:** Read as ‘0’

bit 10 **BCL:** Master Bus Collision Detect bit
1 = A bus collision has been detected during a master operation
0 = No collision
Hardware set at detection of bus collision.

bit 9 **GCSTAT:** General Call Status bit
1 = General call address was received
0 = General call address was not received
Hardware set when the address matches the general call address. Hardware clear at Stop detection.

bit 8 **ADD10:** 10-Bit Address Status bit
1 = 10-bit address was matched
0 = 10-bit address was not matched
Hardware set at the match of 2nd byte of matched 10-bit address. Hardware clear at Stop detection.

bit 7 **IWCOL:** Write Collision Detect bit
1 = An attempt to write to the I2CxTRN register failed because the I^{2}C module is busy
0 = No collision
Hardware set at occurrence of write to I2CxTRN while busy (cleared by software).

bit 6 **I2COV:** Receive Overflow Flag bit
1 = A byte was received while the I2CxRCV register was still holding the previous byte
0 = No overflow
Hardware set at attempt to transfer I2CxRSR to I2CxRCV (cleared by software).

bit 5 **D/$\overline{A}$:** Data/$\overline{\text{Address}}$ bit (when operating as I^{2}C slave)
1 = Indicates that the last byte received was data
0 = Indicates that the last byte received was the the device address
Hardware clear occurs at device address match. Hardware set after a transmission finishes or at reception of a slave byte.

REGISTER 16-2: I2CxSTAT: I2Cx STATUS REGISTER (CONTINUED)

bit 4 **P:** Stop bit
`1` = Indicates that a Stop bit has been detected last
`0` = Stop bit was not detected last
Hardware set or clear when Start, Repeated Start or Stop is detected.

bit 3 **S:** Start bit
`1` = Indicates that a Start (or Repeated Start) bit has been detected last
`0` = Start bit was not detected last
Hardware set or clear when Start, Repeated Start or Stop is detected.

bit 2 **R/$\overline{\text{W}}$:** Read/$\overline{\text{Write}}$ Information bit (when operating as I^2C slave)
`1` = Read – indicates data transfer is output from the slave
`0` = Write – indicates data transfer is input to the slave
Hardware set or clear after reception of I^2C device address byte.

bit 1 **RBF:** Receive Buffer Full Status bit
`1` = Receive is complete, I2CxRCV is full
`0` = Receive is not complete, I2CxRCV is empty
Hardware set when I2CxRCV is written with received byte. Hardware clear when software reads I2CxRCV.

bit 0 **TBF:** Transmit Buffer Full Status bit
`1` = Transmit is in progress, I2CxTRN is full
`0` = Transmit is complete, I2CxTRN is empty
Hardware set when software writes I2CxTRN. Hardware clear at completion of data transmission.

REGISTER 16-3: I2CxMSK: I2Cx SLAVE MODE ADDRESS MASK REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	AMSK9	AMSK8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
AMSK7	AMSK6	AMSK5	AMSK4	AMSK3	AMSK2	AMSK1	AMSK0
bit 7							bit 0

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

bit 15-10 **Unimplemented:** Read as ‘0’

bit 9-0 **AMSK<9:0>:** Mask for Address Bit x Select bits

1 = Enable masking for bit x of incoming message address; bit match is not required in this position
0 = Disable masking for bit x; bit match is required in this position

17.0 UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 21. "UART"** (DS39708).

The Universal Asynchronous Receiver Transmitter (UART) module is one of the serial I/O modules available in the PIC24F device family. The UART is a full-duplex, asynchronous system that can communicate with peripheral devices, such as personal computers, LIN/J2602, RS-232 and RS-485 interfaces. The module also supports a hardware flow control option with the UxCTS and UxRTS pins, and also includes an IrDA® encoder and decoder.

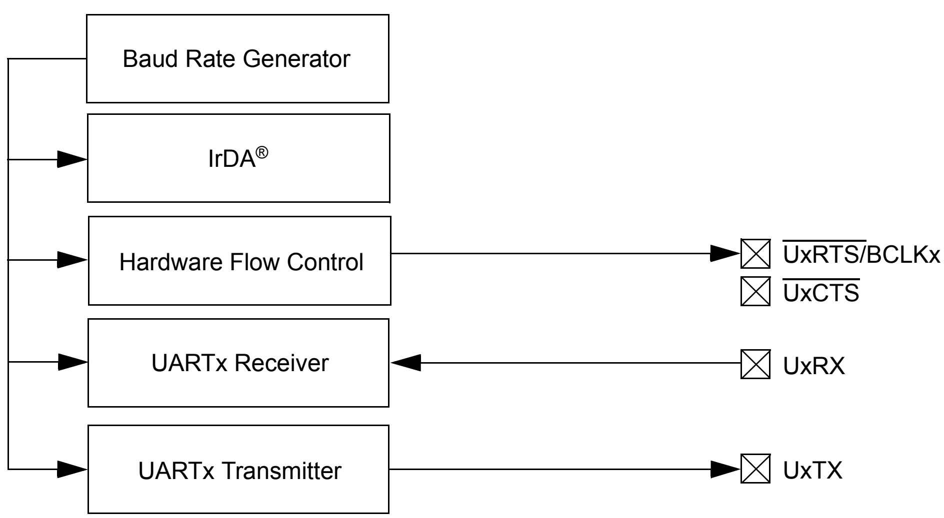
The primary features of the UART module are:

- Full-Duplex, 8 or 9-Bit Data Transmission through the UxTX and UxRX pins
- Even, Odd or No Parity Options (for 8-bit data)
- One or Two Stop bits
- Hardware Flow Control Option with UxCTS and UxRTS pins
- Fully Integrated Baud Rate Generator with 16-Bit Prescaler
- Baud Rates Ranging from 1 Mbps to 15 bps at 16 MIPS
- 4-Deep, First-In-First-Out (FIFO) Transmit Data Buffer
- 4-Deep FIFO Receive Data Buffer
- Parity, Framing and Buffer Overrun Error Detection
- Support for 9-Bit mode with Address Detect (9th bit = 1)
- Transmit and Receive Interrupts
- Loopback mode for Diagnostic Support
- Support for Sync and Break Characters
- Supports Automatic Baud Rate Detection
- IrDA Encoder and Decoder Logic
- 16x Baud Clock Output for IrDA Support

A simplified block diagram of the UART is shown in Figure 17-1. The UART module consists of these key important hardware elements:

- Baud Rate Generator
- Asynchronous Transmitter
- Asynchronous Receiver

FIGURE 17-1: UART SIMPLIFIED BLOCK DIAGRAM



Note: The UART inputs and outputs must all be assigned to available RPn pins before use. Please see **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

17.1 UART Baud Rate Generator (BRG)

The UART module includes a dedicated 16-bit Baud Rate Generator. The UxBRG register controls the period of a free-running, 16-bit timer. Equation 17-1 shows the formula for computation of the baud rate with BRGH = 0.

EQUATION 17-1: UART BAUD RATE WITH BRGH = 0[1,2]

$$\text{Baud Rate} = \frac{F_{CY}}{16 \cdot (\text{UxBRG} + 1)}$$

$$\text{UxBRG} = \frac{F_{CY}}{16 \cdot \text{Baud Rate}} - 1$$

Note 1: FCY denotes the instruction cycle clock frequency (FOSC/2).

2: Based on FCY = FOSC/2, Doze mode and PLL are disabled.

Example 17-1 shows the calculation of the baud rate error for the following conditions:

- FCY = 4 MHz
- Desired Baud Rate = 9600

The maximum baud rate (BRGH = 0) possible is FCY/16 (for UxBRG = 0) and the minimum baud rate possible is FCY/(16 * 65536).

Equation 17-2 shows the formula for computation of the baud rate with BRGH = 1.

EQUATION 17-2: UART BAUD RATE WITH BRGH = 1[1,2]

$$\text{Baud Rate} = \frac{F_{CY}}{4 \cdot (\text{UxBRG} + 1)}$$

$$\text{UxBRG} = \frac{F_{CY}}{4 \cdot \text{Baud Rate}} - 1$$

Note 1: FCY denotes the instruction cycle clock frequency.

2: Based on FCY = FOSC/2, Doze mode and PLL are disabled.

The maximum baud rate (BRGH = 1) possible is FCY/4 (for UxBRG = 0) and the minimum baud rate possible is FCY/(4 * 65536).

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

EXAMPLE 17-1: BAUD RATE ERROR CALCULATION (BRGH = 0)[1]

Desired Baud Rate = FCY/(16 (UxBRG + 1))

Solving for UxBRG Value:

UxBRG = ((FCY/Desired Baud Rate)/16) – 1
UxBRG = ((4000000/9600)/16) – 1
UxBRG = 25

Calculated Baud Rate = 4000000/(16 (25 + 1))
= 9615

Error = (Calculated Baud Rate – Desired Baud Rate)
Desired Baud Rate
= (9615 – 9600)/9600
= 0.16%

Note 1: Based on FCY = FOSC/2, Doze mode and PLL are disabled.

17.2 Transmitting in 8-Bit Data Mode

1. Set up the UART:
   a) Write appropriate values for data, parity and Stop bits.
   b) Write appropriate baud rate value to the UxBRG register.
   c) Set up transmit and receive interrupt enable and priority bits.
2. Enable the UART.
3. Set the UTXEN bit (causes a transmit interrupt two cycles after being set).
4. Write data byte to the lower byte of the UxTXREG word. The value will be immediately transferred to the Transmit Shift Register (TSR) and the serial bit stream will start shifting out with the next rising edge of the baud clock.
5. Alternately, the data byte may be transferred while UTXEN = `0`, and then the user may set UTXEN. This will cause the serial bit stream to begin immediately because the baud clock will start from a cleared state.
6. A transmit interrupt will be generated as per interrupt control bit, UTXISELx.

17.3 Transmitting in 9-Bit Data Mode

1. Set up the UART (as described in **Section 17.2 "Transmitting in 8-Bit Data Mode"**).
2. Enable the UART.
3. Set the UTXEN bit (causes a transmit interrupt).
4. Write UxTXREG as a 16-bit value only.
5. A word write to UxTXREG triggers the transfer of the 9-bit data to the TSR. The serial bit stream will start shifting out with the first rising edge of the baud clock.
6. A transmit interrupt will be generated as per the setting of control bit, UTXISELx.

17.4 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.

1. Configure the UART for the desired mode.
2. Set UTXEN and UTXBRK to set up the Break character.
3. Load the UxTXREG with a dummy character to initiate transmission (value is ignored).
4. Write '55h' to UxTXREG; this loads the Sync character into the transmit FIFO.
5. After the Break has been sent, the UTXBRK bit is reset by hardware. The Sync character now transmits.

17.5 Receiving in 8-Bit or 9-Bit Data Mode

1. Set up the UART (as described in **Section 17.2 "Transmitting in 8-Bit Data Mode"**).
2. Enable the UART.
3. A receive interrupt will be generated when one or more data characters have been received as per interrupt control bit, URXISELx.
4. Read the OERR bit to determine if an overrun error has occurred. The OERR bit must be reset in software.
5. Read UxRXREG.

The act of reading the UxRXREG character will move the next character to the top of the receive FIFO, including a new set of PERR and FERR values.

17.6 Operation of $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ Control Pins

UARTx Clear to Send ($\overline{\text{UxCTS}}$) and Request to Send ($\overline{\text{UxRTS}}$) are the two hardware-controlled pins that are associated with the UART module. These two pins allow the UART to operate in Simplex and Flow Control modes. They are implemented to control the transmission and reception between the Data Terminal Equipment (DTE). The UEN<1:0> bits in the UxMODE register configure these pins.

17.7 Infrared Support

The UART module provides two types of infrared UART support: one is the IrDA clock output to support the external IrDA encoder and decoder device (legacy module support), and the other is the full implementation of the IrDA encoder and decoder. Note that because the IrDA modes require a 16x baud clock, they will only work when the BRGH bit (UxMODE<3>) is '`0`'.

17.7.1 IRDA CLOCK OUTPUT FOR EXTERNAL IRDA SUPPORT

To support external IrDA encoder and decoder devices, the BCLKx pin (same as the $\overline{\text{UxRTS}}$ pin) can be configured to generate the 16x baud clock. When UEN<1:0> = `11`, the BCLKx pin will output the 16x baud clock if the UART module is enabled. It can be used to support the IrDA codec chip.

17.7.2 BUILT-IN IRDA ENCODER AND DECODER

The UART has full implementation of the IrDA encoder and decoder as part of the UART module. The built-in IrDA encoder and decoder functionality is enabled using the IREN bit (UxMODE<12>). When enabled (IREN = `1`), the receive pin (UxRX) acts as the input from the infrared receiver. The transmit pin (UxTX) acts as the output to the infrared transmitter.

REGISTER 17-1: UxMODE: UARTx MODE REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
UARTEN(1)	—	USIDL	IREN(2)	RTSMD	—	UEN1	UEN0
bit 15							bit 8

R/W-0, HC	R/W-0	R/W-0, HC	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL
bit 7							bit 0

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

bit 15 **UARTEN:** UARTx Enable bit(1)
- 1 = UARTx is enabled; all UARTx pins are controlled by UARTx as defined by UEN<1:0>
- 0 = UARTx is disabled; all UARTx pins are controlled by port latches; UARTx power consumption is minimal

bit 14 **Unimplemented:** Read as '0'

bit 13 **USIDL:** Stop in Idle Mode bit
- 1 = Discontinue module operation when the device enters Idle mode
- 0 = Continue module operation in Idle mode

bit 12 **IREN:** IrDA® Encoder and Decoder Enable bit(2)
- 1 = IrDA encoder and decoder are enabled
- 0 = IrDA encoder and decoder are disabled

bit 11 **RTSMD:** Mode Selection for $\overline{\text{UxRTS}}$ Pin bit
- 1 = $\overline{\text{UxRTS}}$ pin is in Simplex mode
- 0 = $\overline{\text{UxRTS}}$ pin is in Flow Control mode

bit 10 **Unimplemented:** Read as '0'

bit 9-8 **UEN<1:0>:** UARTx Enable bits
- 11 = UxTX, UxRX and BCLKx pins are enabled and used; $\overline{\text{UxCTS}}$ pin is controlled by port latches
- 10 = UxTX, UxRX, $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ pins are enabled and used
- 01 = UxTX, UxRX and $\overline{\text{UxRTS}}$ pins are enabled and used; $\overline{\text{UxCTS}}$ pin is controlled by port latches
- 00 = UxTX and UxRX pins are enabled and used; $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$/BCLKx pins are controlled by port latches

bit 7 **WAKE:** Wake-up on Start Bit Detect During Sleep Mode Enable bit
- 1 = UARTx will continue to sample the UxRX pin; interrupt generated on falling edge; bit cleared in hardware on following rising edge
- 0 = No wake-up is enabled

bit 6 **LPBACK:** UARTx Loopback Mode Select bit
- 1 = Enable Loopback mode
- 0 = Loopback mode is disabled

bit 5 **ABAUD:** Auto-Baud Enable bit
- 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 is disabled or completed

bit 4 **RXINV:** Receive Polarity Inversion bit
- 1 = UxRX Idle state is '0'
- 0 = UxRX Idle state is '1'

Note 1: If UARTEN = 1, the peripheral inputs and outputs must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

2: This feature is only available for the 16x BRG mode (BRGH = 0).

REGISTER 17-1: UxMODE: UARTx MODE REGISTER (CONTINUED)

bit 3 **BRGH:** High Baud Rate Enable bit
1 = High-Speed mode (four BRG clock cycles per bit)
0 = Standard mode (16 BRG clock cycles per bit)

bit 2-1 **PDSEL<1:0>:** Parity and Data Selection bits
11 = 9-bit data, no parity
10 = 8-bit data, odd parity
01 = 8-bit data, even parity
00 = 8-bit data, no parity

bit 0 **STSEL:** Stop Bit Selection bit
1 = Two Stop bits
0 = One Stop bit

Note 1: If UARTEN = 1, the peripheral inputs and outputs must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

2: This feature is only available for the 16x BRG mode (BRGH = 0).

REGISTER 17-2: UxSTA: UARTx STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	U-0	R/W-0, HC	R/W-0	R-0	R-1
UTXISEL1	UTXINV(1)	UTXISEL0	—	UTXBRK	UTXEN(2)	UTXBF	TRMT
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-1	R-0	R-0	R/C-0	R-0
URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

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

bit 15,13 **UTXISEL<1:0>:** Transmission Interrupt Mode Selection bits

11 = Reserved; do not use
10 = Interrupt when a character is transferred to the Transmit Shift Register (TSR), and as a result, the transmit buffer becomes empty
01 = Interrupt when the last character is shifted out of the Transmit Shift Register; all transmit operations are completed
00 = Interrupt when a character is transferred to the Transmit Shift Register (this implies there is at least one character open in the transmit buffer)

bit 14 **UTXINV:** IrDA® Encoder Transmit Polarity Inversion bit(1)

IREN = 0:
1 = UxTX Idle '0'
0 = UxTX Idle '1'

IREN = 1:
1 = UxTX Idle '1'
0 = UxTX Idle '0'

bit 12 **Unimplemented:** Read as '0'

bit 11 **UTXBRK:** Transmit Break bit

1 = Send Sync Break on next transmission – Start bit, followed by twelve '0' bits, followed by Stop bit; cleared by hardware upon completion
0 = Sync Break transmission is disabled or completed

bit 10 **UTXEN:** Transmit Enable bit(2)

1 = Transmit is enabled, UxTX pin is controlled by UARTx
0 = Transmit is disabled, any pending transmission is aborted and the buffer is reset; UxTX pin is controlled by port

bit 9 **UTXBF:** Transmit Buffer Full Status bit (read-only)

1 = Transmit buffer is full
0 = Transmit buffer is not full; at least one more character can be written

bit 8 **TRMT:** Transmit Shift Register Empty bit (read-only)

1 = Transmit Shift Register is empty and transmit buffer is empty (the last transmission has completed)
0 = Transmit Shift Register is not empty, a transmission is in progress or queued

bit 7-6 **URXISEL<1:0>:** Receive Interrupt Mode Selection bits

11 = Interrupt is set on RSR transfer, making the receive buffer full (i.e., has 4 data characters)
10 = Interrupt is set on RSR transfer, making the receive buffer 3/4 full (i.e., has 3 data characters)
0x = Interrupt is set when any character is received and transferred from the RSR to the receive buffer; receive buffer has one or more characters

Note 1: Value of bit only affects the transmit properties of the module when the IrDA encoder is enabled (IREN = 1).

2: If UARTEN = 1, the peripheral inputs and outputs must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

REGISTER 17-2: UxSTA: UARTx STATUS AND CONTROL REGISTER (CONTINUED)

bit 5 **ADDEN:** Address Character Detect bit (bit 8 of received data = 1)
1 = Address Detect mode is enabled. If 9-bit mode is not selected, this does not take effect.
0 = Address Detect mode is disabled

bit 4 **RIDLE:** Receiver Idle bit (read-only)
1 = Receiver is Idle
0 = Receiver is active

bit 3 **PERR:** Parity Error Status bit (read-only)
1 = Parity error has been detected for the current character (character at the top of the receive FIFO)
0 = Parity error has not been detected

bit 2 **FERR:** Framing Error Status bit (read-only)
1 = Framing error has been detected for the current character (character at the top of the receive FIFO)
0 = Framing error has not been detected

bit 1 **OERR:** Receive Buffer Overrun Error Status bit (clear/read-only)
1 = Receive buffer has overflowed
0 = Receive buffer has not overflowed (clearing a previously set OERR bit (1 → 0 transition) will reset the receiver buffer and the RSR to the empty state

bit 0 **URXDA:** Receive Buffer Data Available bit (read-only)
1 = Receive buffer has data, at least one more character can be read
0 = Receive buffer is empty

Note 1: Value of bit only affects the transmit properties of the module when the IrDA encoder is enabled (IREN = 1).

2: If UARTEN = 1, the peripheral inputs and outputs must be configured to an available RPn pin. See **Section 10.4 “Peripheral Pin Select (PPS)”** for more information.

NOTES:

18.0 PARALLEL MASTER PORT (PMP)

> **Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 13. "Parallel Master Port (PMP)"** (DS39713).

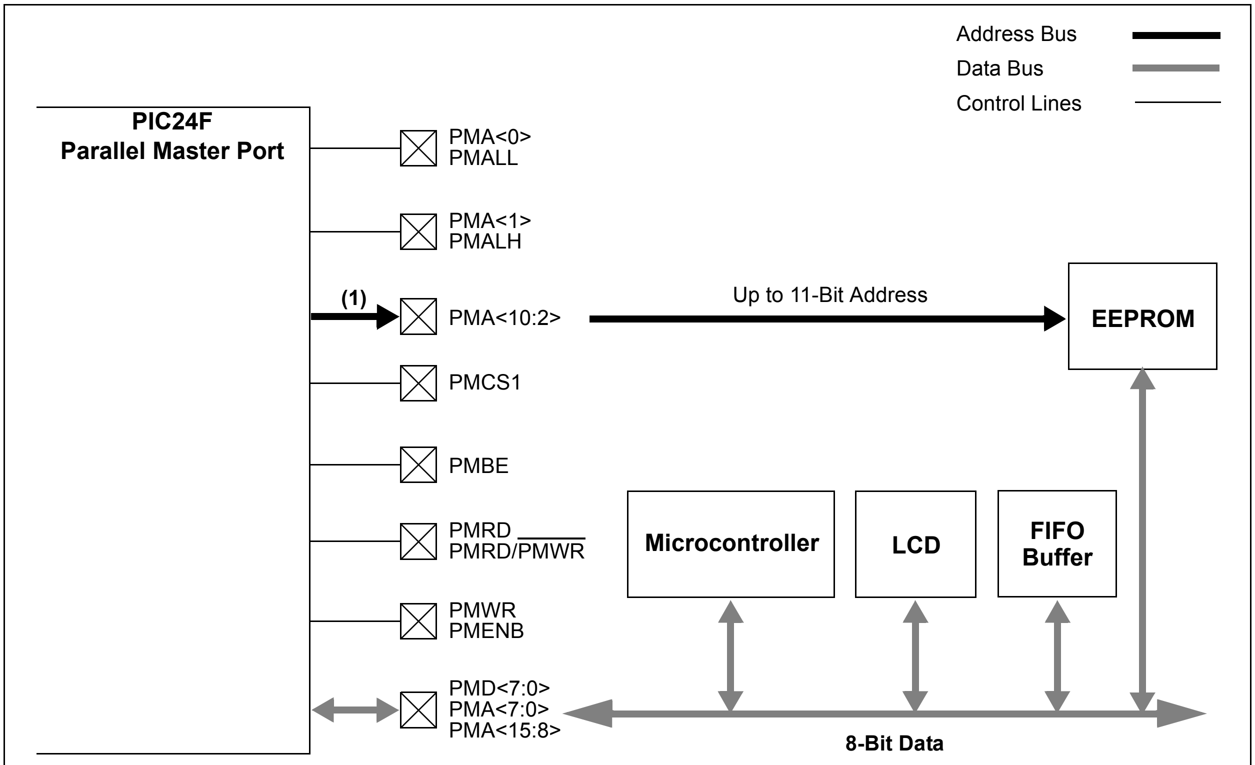
The Parallel Master Port (PMP) module is a parallel, 8-bit I/O module, specifically designed to communicate with a wide variety of parallel devices, such as communication peripherals, LCDs, external memory devices and microcontrollers. Because the interface to parallel peripherals varies significantly, the PMP is highly configurable.

> **Note:** A number of the pins for the PMP are not present on PIC24FJ64GA1 family devices. Refer to the specific device's pinout to determine which pins are available.

Key features of the PMP module include:

- Up to 16 Programmable Address Lines
- One Chip Select Line
- Programmable Strobe Options:
  - Individual Read and Write Strobes or;
  - Read/$\overline{\text{Write}}$ Strobe with Enable Strobe
- Address Auto-Increment/Auto-Decrement
- Programmable Address/Data Multiplexing
- Programmable Polarity on Control Signals
- Legacy Parallel Slave Port Support
- Enhanced Parallel Slave Support:
  - Address Support
  - 4-Byte Deep Auto-Incrementing Buffer
- Programmable Wait States
- Selectable Input Voltage Levels

FIGURE 18-1: PMP MODULE OVERVIEW



Note 1: PMA<10:2> bits are not available on 28-pin devices.

REGISTER 18-1: PMCON: PARALLEL PORT CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMPEN	—	PSIDL	ADRMUX1(1)	ADRMUX0(1)	PTBEEN	PTWREN	PTRDEN
bit 15							bit 8

R/W-0	R/W-0	R/W-0(2)	U-0	R/W-0(2)	R/W-0	R/W-0	R/W-0
CSF1	CSF0	ALP	—	CS1P	BEP	WRSP	RDSP
bit 7							bit 0

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

bit 15 **PMPEN:** Parallel Master Port Enable bit
- 1 = PMP is enabled
- 0 = PMP is disabled, no off-chip access performed

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **PSIDL:** Stop in Idle Mode bit
- 1 = Discontinue module operation when device enters Idle mode
- 0 = Continue module operation in Idle mode

bit 12-11 **ADRMUX<1:0>:** Address/Data Multiplexing Selection bits(1)
- 11 = Reserved
- 10 = All 16 bits of address are multiplexed on PMD<7:0> pins
- 01 = Lower 8 bits of address are multiplexed on PMD<7:0> pins; upper 3 bits are multiplexed on PMA<10:8>
- 00 = Address and data appear on separate pins

bit 10 **PTBEEN:** Byte Enable Port Enable bit (16-Bit Master mode)
- 1 = PMBE port is enabled
- 0 = PMBE port is disabled

bit 9 **PTWREN:** Write Enable Strobe Port Enable bit
- 1 = PMWR/PMENB port is enabled
- 0 = PMWR/PMENB port is disabled

bit 8 **PTRDEN:** Read/Write Strobe Port Enable bit
- 1 = PMRD/PMWR port is enabled
- 0 = PMRD/PMWR port is disabled

bit 7-6 **CSF<1:0>:** Chip Select Function bits
- 11 = Reserved
- 10 = PMCS1 functions as chip set
- 01 = Reserved
- 00 = Reserved

bit 5 **ALP:** Address Latch Polarity bit(2)
- 1 = Active-high (PMALL and PMALH)
- 0 = Active-low (PMALL and PMALH)

bit 4 **Unimplemented:** Read as ‘0’

bit 3 **CS1P:** Chip Select 1 Polarity bit(2)
- 1 = Active-high (PMCS1/PMCS1)
- 0 = Active-low (PMCS1/PMCS1)

Note 1: PMA<10:2> bits are not available on 28-pin devices.
2: These bits have no effect when their corresponding pins are used as address lines.

REGISTER 18-1: PMCON: PARALLEL PORT CONTROL REGISTER (CONTINUED)

bit 2 **BEP:** Byte Enable Polarity bit

`1` = Byte enable active-high (PMBE)
`0` = Byte enable active-low ($\overline{\text{PMBE}}$)

bit 1 **WRSP:** Write Strobe Polarity bit

For Slave modes and Master Mode 2 (PMMODE<9:8> = `00,01,10`):
`1` = Write strobe active-high (PMWR)
`0` = Write strobe active-low ($\overline{\text{PMWR}}$)

For Master Mode 1 (PMMODE<9:8> = `11`):
`1` = Enable strobe active-high (PMENB)
`0` = Enable strobe active-low ($\overline{\text{PMENB}}$)

bit 0 **RDSP:** Read Strobe Polarity bit

For Slave modes and Master Mode 2 (PMMODE<9:8> = `00,01,10`):
`1` = Read strobe active-high (PMRD)
`0` = Read strobe active-low ($\overline{\text{PMRD}}$)

For Master Mode 1 (PMMODE<9:8> = `11`):
`1` = Read/write strobe active-high (PMRD/$\overline{\text{PMWR}}$)
`0` = Read/write strobe active-low ($\overline{\text{PMRD}}$/PMWR)

Note 1: PMA<10:2> bits are not available on 28-pin devices.
2: These bits have no effect when their corresponding pins are used as address lines.

REGISTER 18-2: PMMODE: PARALLEL PORT MODE REGISTER

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WAITB1[1]	WAITB0[1]	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1[1]	WAITE0[1]
bit 7							bit 0

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

bit 15 **BUSY:** Busy bit (Master mode only)
1 = Port is busy (not useful when the processor stall is active)
0 = Port is not busy

bit 14-13 **IRQM<1:0>:** Interrupt Request Mode bits
11 = Interrupt generated when Read Buffer 3 is read or Write Buffer 3 is written (Buffered PSP mode), or on a read or write operation when PMA<1:0> = 11 (Addressable PSP mode only)
10 = No interrupt is generated, processor stall activated
01 = Interrupt is generated at the end of the read/write cycle
00 = No interrupt is generated

bit 12-11 **INCM<1:0>:** Increment Mode bits
11 = PSP read and write buffers auto-increment (Legacy PSP mode only)
10 = Decrement ADDR<10:0> by 1 every read/write cycle
01 = Increment ADDR<10:0> by 1 every read/write cycle
00 = No increment or decrement of address

bit 10 **MODE16:** 8/16-Bit Mode bit
1 = 16-bit mode: Data register is 16 bits; a read or write to the Data register invokes two 8-bit transfers
0 = 8-bit mode: Data register is 8 bits; a read or write to the Data register invokes one 8-bit transfer

bit 9-8 **MODE<1:0>:** Parallel Port Mode Select bits
11 = Master Mode 1 (PMCS1, PMRD/PMWR, PMENB, PMBE, PMA<x:0> and PMD<7:0>)
10 = Master Mode 2 (PMCS1, PMRD, PMWR, PMBE, PMA<x:0> and PMD<7:0>)
01 = Enhanced PSP control signals (PMRD, PMWR, PMCS1, PMD<7:0> and PMA<1:0>)
00 = Legacy Parallel Slave Port control signals (PMRD, PMWR, PMCS1 and PMD<7:0>)

bit 7-6 **WAITB<1:0>:** Data Setup to Read/Write Wait State Configuration bits[1]
11 = Data wait of 4 TCY; multiplexed address phase of 4 TCY
10 = Data wait of 3 TCY; multiplexed address phase of 3 TCY
01 = Data wait of 2 TCY; multiplexed address phase of 2 TCY
00 = Data wait of 1 TCY; multiplexed address phase of 1 TCY

bit 5-2 **WAITM<3:0>:** Read to Byte Enable Strobe Wait State Configuration bits
1111 = Wait of additional 15 TCY
...
0001 = Wait of additional 1 TCY
0000 = No additional wait cycles (operation forced into one TCY)

bit 1-0 **WAITE<1:0>:** Data Hold After Strobe Wait State Configuration bits[1]
11 = Wait of 4 TCY
10 = Wait of 3 TCY
01 = Wait of 2 TCY
00 = Wait of 1 TCY

Note 1: WAITB and WAITE bits are ignored whenever WAITM<3:0> = 0000.

REGISTER 18-3: PMADDR: PARALLEL PORT ADDRESS REGISTER

U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	CS1	—	—	—	ADDR10(1)	ADDR9(1)	ADDR8(1)
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADDR7(1)	ADDR6(1)	ADDR5(1)	ADDR4(1)	ADDR3(1)	ADDR2(1)	ADDR1(1)	ADDR0(1)
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as '0'

bit 14 **CS1:** Chip Select 1 bit
1 = Chip Select 1 is active
0 = Chip Select 1 is inactive

bit 13-11 **Unimplemented:** Read as '0'

bit 10-0 **ADDR<10:0>:** Parallel Port Destination Address bits(1)

Note 1: PMA<10:2> bits are not available on 28-pin devices.

REGISTER 18-4: PMAEN: PARALLEL PORT ENABLE REGISTER

U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	PTEN14	—	—	—	PTEN10(1)	PTEN9(1)	PTEN8(1)
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN7(1)	PTEN6(1)	PTEN5(1)	PTEN4(1)	PTEN3(1)	PTEN2(1)	PTEN1	PTEN0
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as '0'

bit 14 **PTEN14:** PMCS1 Strobe Enable bit
1 = PMCS1 functions as chip select
0 = PMCS1 pin functions as port I/O

bit 13-11 **Unimplemented:** Read as '0'

bit 10-2 **PTEN<10:2>:** PMP Address Port Enable bits(1)
1 = PMA<10:2> function as PMP address lines
0 = PMA<10:2> function as port I/O

bit 1-0 **PTEN<1:0>:** PMALH/PMALL Strobe Enable bits
1 = PMA1 and PMA0 function as either PMA<1:0> or PMALH and PMALL
0 = PMA1 and PMA0 pads function as port I/O

Note 1: PMA<10:2> bits are not available on 28-pin devices.

REGISTER 18-5: PMSTAT: PARALLEL PORT STATUS REGISTER

R-0	R/W-0, HS	U-0	U-0	R-0	R-0	R-0	R-0
IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F
bit 15							bit 8

R-1	R/W-0, HS	U-0	U-0	R-1	R-1	R-1	R-1
OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E
bit 7							bit 0

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

bit 15 **IBF:** Input Buffer Full Status bit
1 = All writable input buffer registers are full
0 = Some or all of the writable input buffer registers are empty

bit 14 **IBOV:** Input Buffer Overflow Status bit
1 = A write attempt to a full input byte register occurred (must be cleared in software)
0 = No overflow occurred

bit 13-12 **Unimplemented:** Read as ‘0’

bit 11-8 **IB3F:IB0F** Input Buffer x Status Full bits
1 = Input buffer contains data that has not been read (reading buffer will clear this bit)
0 = Input buffer does not contain any unread data

bit 7 **OBE:** Output Buffer Empty Status bit
1 = All readable output buffer registers are empty
0 = Some or all of the readable output buffer registers are full

bit 6 **OBUF:** Output Buffer Underflow Status bits
1 = A read occurred from an empty output byte register (must be cleared in software)
0 = No underflow occurred

bit 5-4 **Unimplemented:** Read as ‘0’

bit 3-0 **OB3E:OB0E** Output Buffer x Status Empty bits
1 = Output buffer is empty (writing data to the buffer will clear this bit)
0 = Output buffer contains data that has not been transmitted

REGISTER 18-6: PADCFG1: PAD CONFIGURATION CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	RTSECSEL1[(1)]	RTSECSEL0[(1)]	PMPTTL
bit 7							bit 0

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

bit 15-3 **Unimplemented:** Read as ‘0’

bit 2-1 **RTSECSEL<1:0>:** RTCC Seconds Clock Output Select bits[(1)]

- 11 = Reserved; do not use
- 10 = RTCC source clock is selected for the RTCC pin (clock can be LPRC or SOSC, depending on the setting of the Flash Configuration bit, RTCOSC (CW4<5>))
- 01 = RTCC seconds clock is selected for the RTCC pin
- 00 = RTCC alarm pulse is selected for the RTCC pin

bit 0 **PMPTTL:** PMP Module TTL Input Buffer Select bit

- 1 = PMP module uses TTL input buffers
- 0 = PMP module uses Schmitt Trigger input buffers

Note 1: To enable the actual RTCC output, the RTCOE (RCFGCAL<10>) bit needs to be set.

FIGURE 18-2: LEGACY PARALLEL SLAVE PORT EXAMPLE

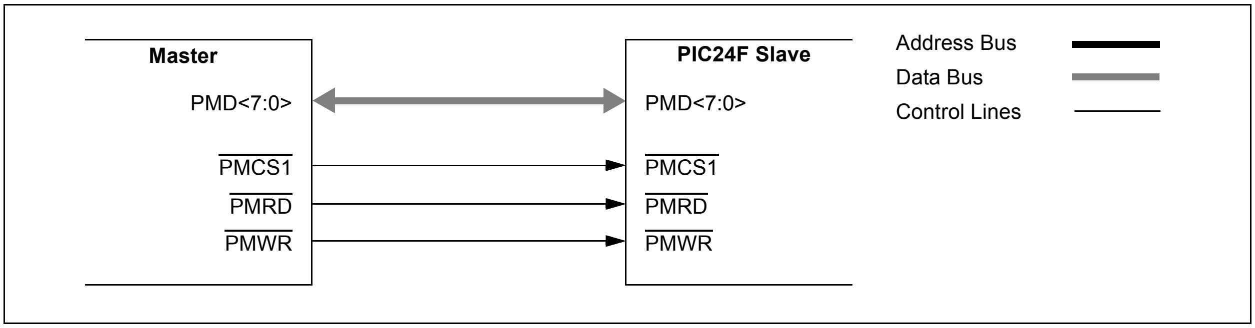


FIGURE 18-3: ADDRESSABLE PARALLEL SLAVE PORT EXAMPLE

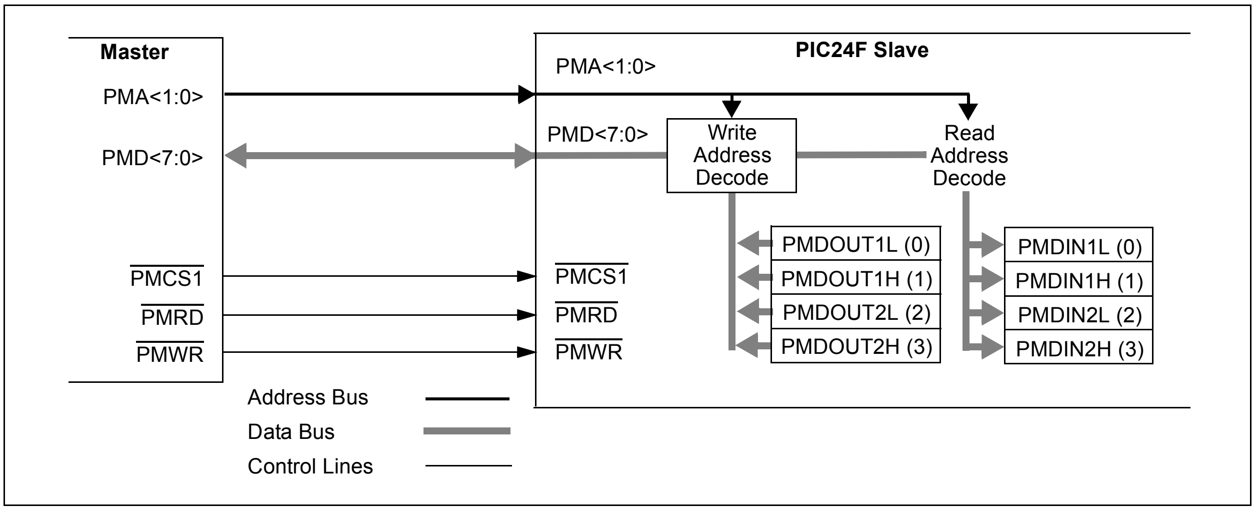


TABLE 18-1: SLAVE MODE ADDRESS RESOLUTION

PMA<1:0>	Output Register (Buffer)	Input Register (Buffer)
00	PMDOUT1<7:0> (0)	PMDIN1<7:0> (0)
01	PMDOUT1<15:8> (1)	PMDIN1<15:8> (1)
10	PMDOUT2<7:0> (2)	PMDIN2<7:0> (2)
11	PMDOUT2<15:8> (3)	PMDIN2<15:8> (3)

FIGURE 18-4: MASTER MODE, DEMULTIPLEXED ADDRESSING (SEPARATE READ AND WRITE STROBES, SINGLE CHIP SELECT)

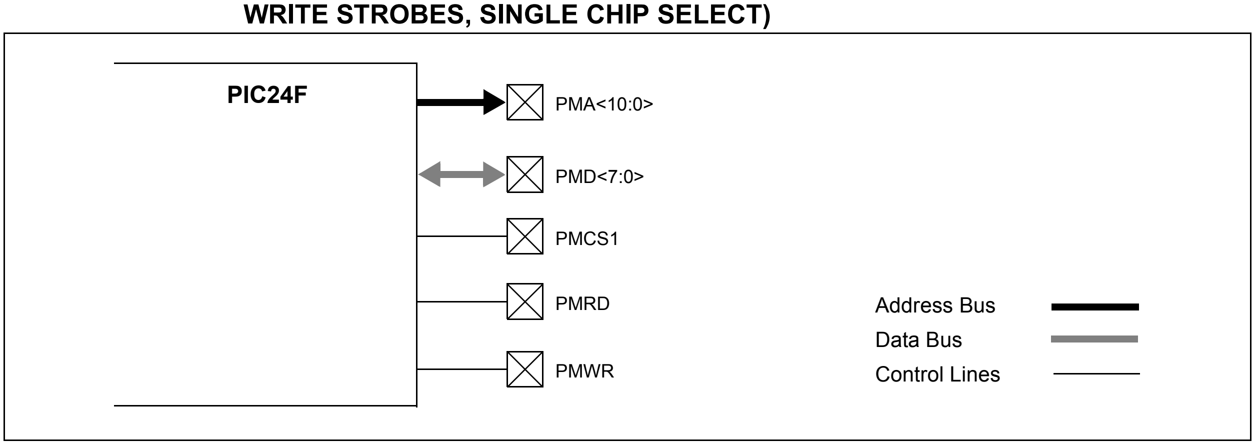


FIGURE 18-5: MASTER MODE, PARTIALLY MULTIPLEXED ADDRESSING (SEPARATE READ AND WRITE STROBES, SINGLE CHIP SELECT)

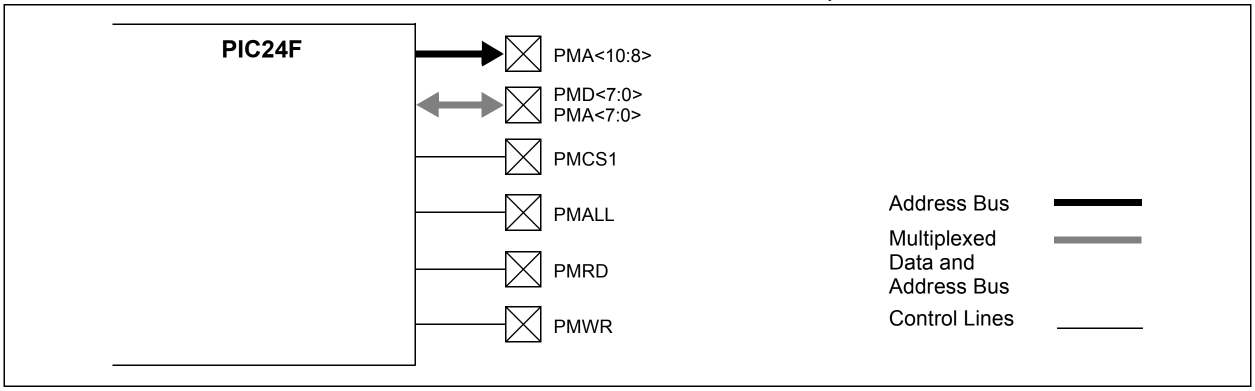


FIGURE 18-6: MASTER MODE, FULLY MULTIPLEXED ADDRESSING (SEPARATE READ AND WRITE STROBES, SINGLE CHIP SELECT)

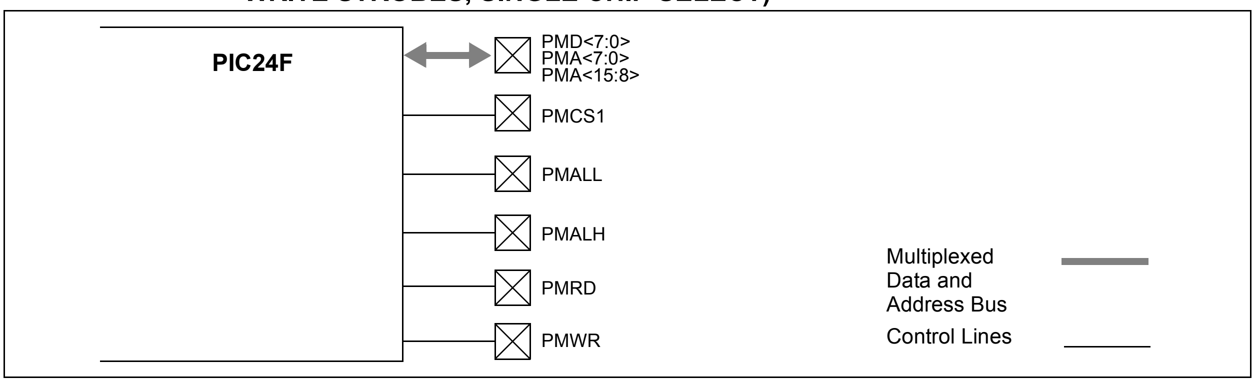


FIGURE 18-7: EXAMPLE OF A MULTIPLEXED ADDRESSING APPLICATION

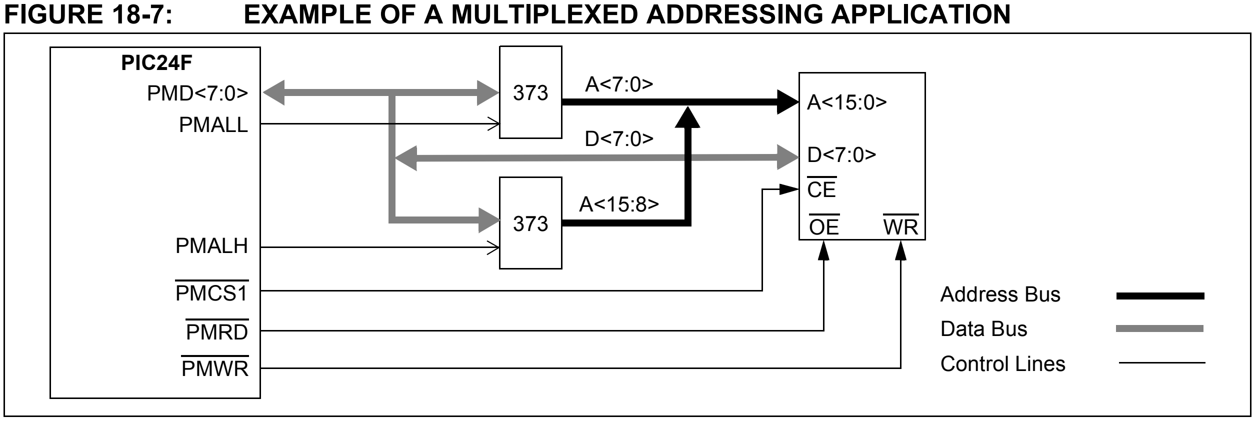


FIGURE 18-8: EXAMPLE OF A PARTIALLY MULTIPLEXED ADDRESSING APPLICATION

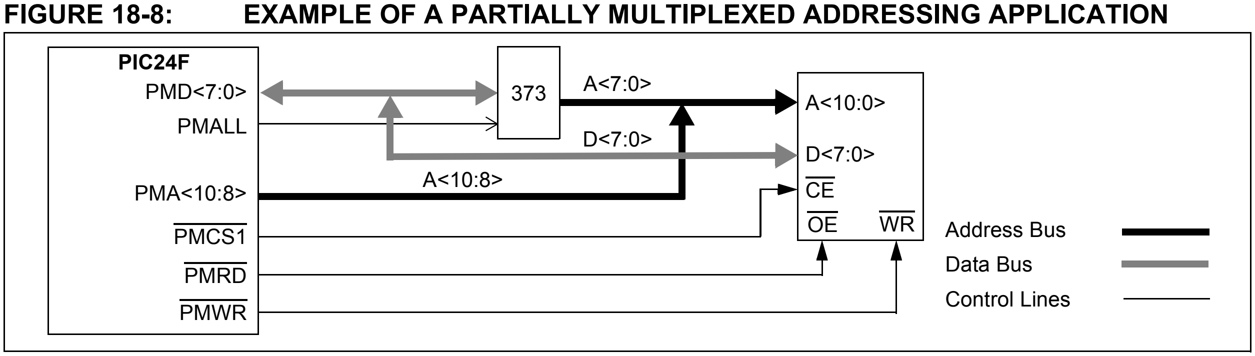


FIGURE 18-9: EXAMPLE OF AN 8-BIT MULTIPLEXED ADDRESS AND DATA APPLICATION

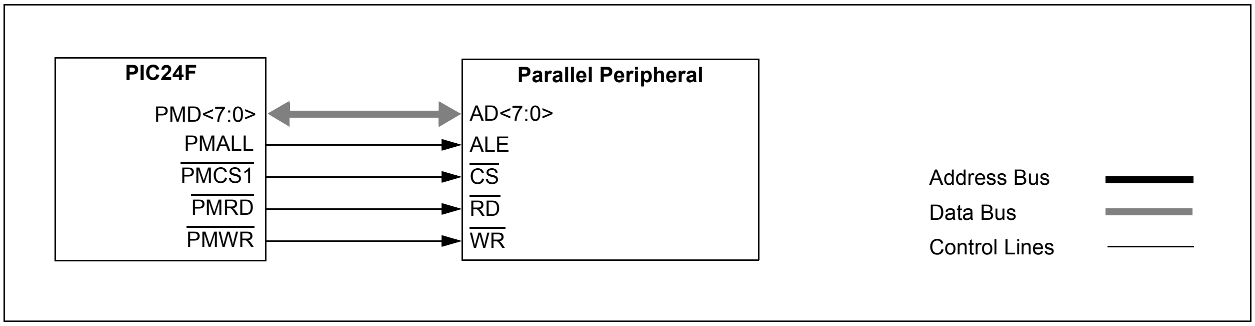


FIGURE 18-10: PARALLEL EEPROM EXAMPLE (UP TO 11-BIT ADDRESS, 8-BIT DATA)

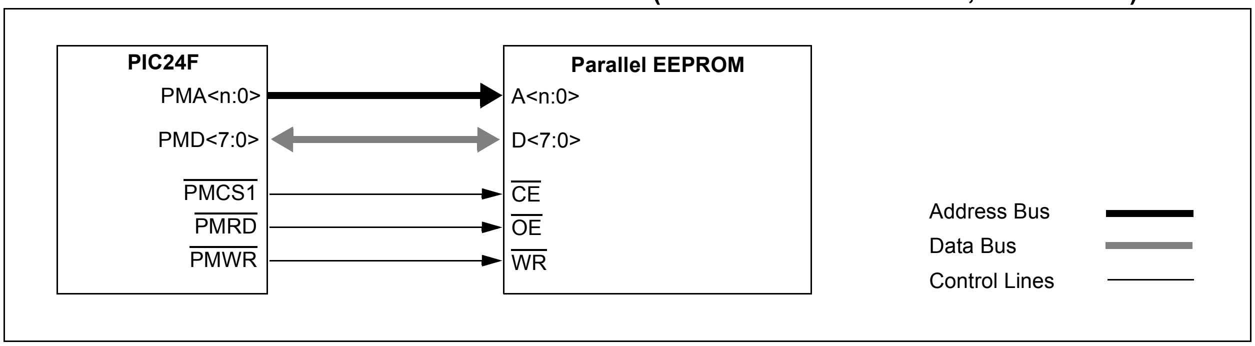


FIGURE 18-11: PARALLEL EEPROM EXAMPLE (UP TO 11-BIT ADDRESS, 16-BIT DATA)

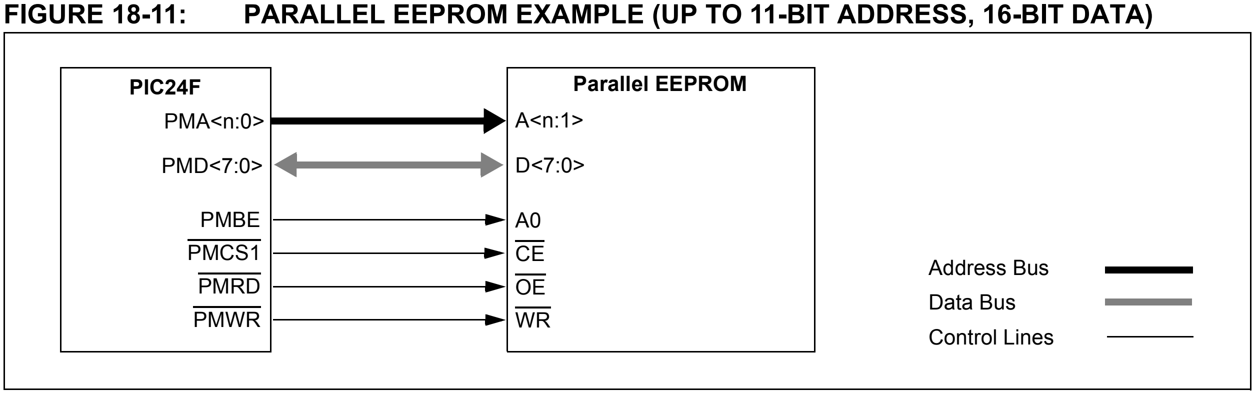
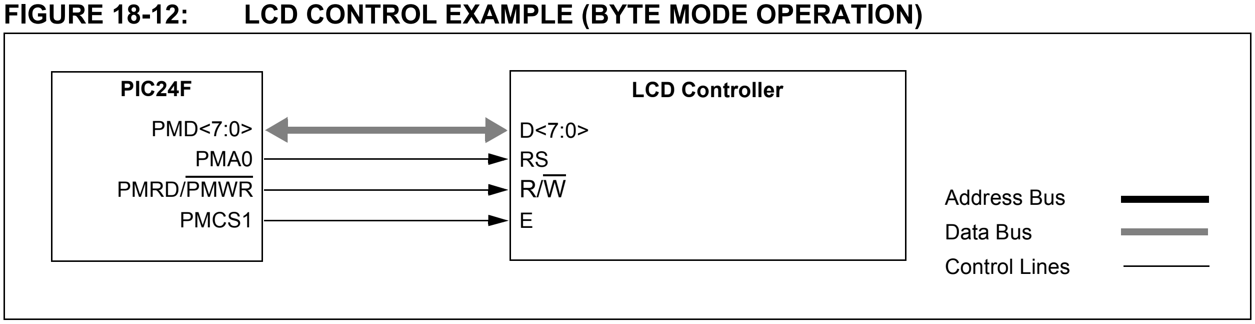


FIGURE 18-12: LCD CONTROL EXAMPLE (BYTE MODE OPERATION)



19.0 REAL-TIME CLOCK AND CALENDAR (RTCC)

Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 29. "Real-Time Clock and Calendar (RTCC)"** (DS39696).

The RTCC provides the user with a Real-Time Clock and Calendar (RTCC) function that can be calibrated.

Key features of the RTCC module are:

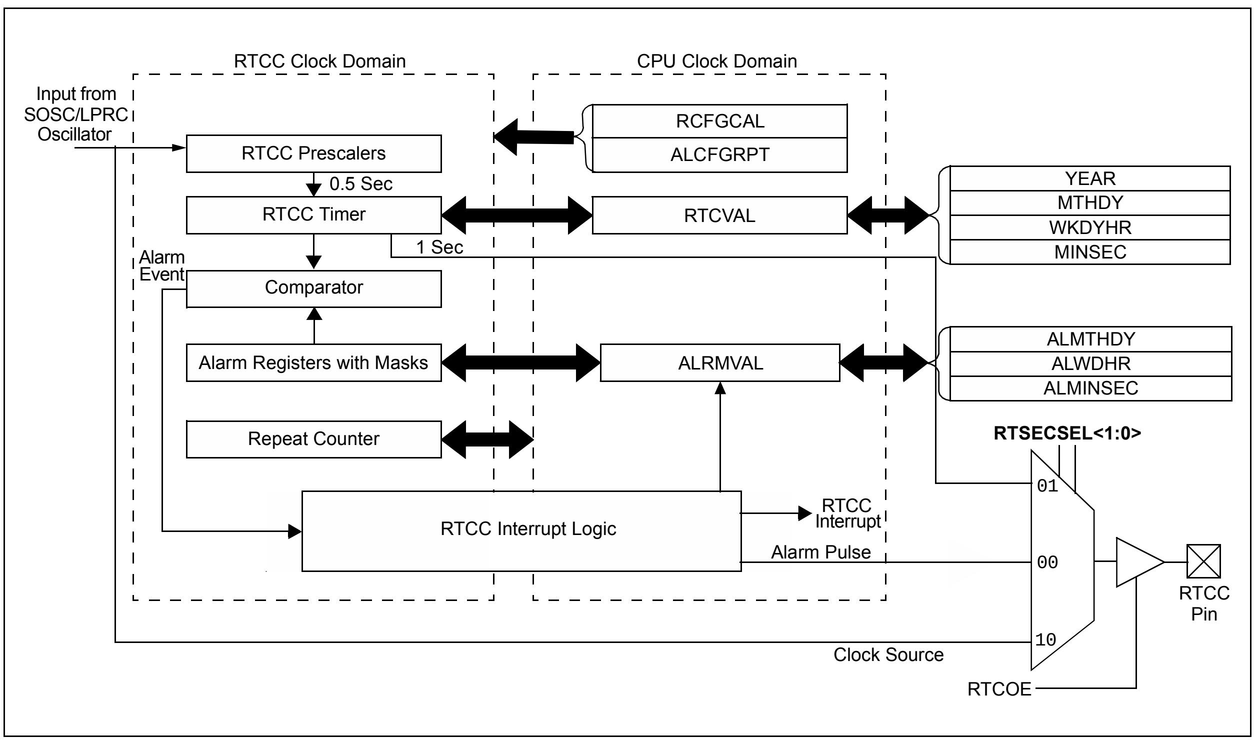
- Operates in Deep Sleep mode
- Selectable clock source
- Provides hours, minutes and seconds using 24-hour format
- Visibility of one half second period
- Provides calendar – weekday, date, month and year
- Alarm-configurable for half a second, one second, 10 seconds, one minute, 10 minutes, one hour, one day, one week, one month or one year
- Alarm repeat with decrementing counter
- Alarm with indefinite repeat chime
- Year 2000 to 2099 leap year correction
- BCD format for smaller software overhead
- Optimized for long-term battery operation
- User calibration of the 32.768 kHz clock crystal/32K INTRC frequency with periodic auto-adjust

19.1 RTCC Source Clock

The user can select between the SOSC crystal oscillator or the LPRC Low-Power Internal Oscillator as the clock reference for the RTCC module. This is configured using the RTCOSC (CW4<5>) Configuration bit. This gives the user an option to trade off system cost, accuracy and power consumption, based on the overall system needs.

The SOSC and RTCC will both remain running while the device is held in Reset with $\overline{\text{MCLR}}$ and will continue running after $\overline{\text{MCLR}}$ is released.

FIGURE 19-1: RTCC BLOCK DIAGRAM



19.2 RTCC Module Registers

The RTCC module registers are organized into three categories:

- RTCC Control Registers
- RTCC Value Registers
- Alarm Value Registers

19.2.1 REGISTER MAPPING

To limit the register interface, the RTCC Timer and Alarm Time registers are accessed through corresponding register pointers. The RTCC Value register window (RTCVALH and RTCVALL) uses the RTCPTR bits (RCFGCAL<9:8>) to select the desired Timer register pair (see Table 19-1).

By writing to the RTCVALH byte, the RTCC Pointer value (the RTCPTR<1:0> bits) decrements by one until they reach '00'. Once they reach '00', the MINUTES and SECONDS value will be accessible through RTCVALH and RTCVALL until the pointer value is manually changed.

TABLE 19-1: RTCVAL REGISTER MAPPING

RTCPTR<1:0>	RTCC Value Register Window	
	RTCVAL<15:8>	RTCVAL<7:0>
00	MINUTES	SECONDS
01	WEEKDAY	HOURS
10	MONTH	DAY
11	—	YEAR

The Alarm Value register window (ALRMVALH and ALRMVALL) uses the ALRMPTR bits (ALCFGRPT<9:8>) to select the desired Alarm register pair (see Table 19-2).

By writing to the ALRMVALH byte, the Alarm Pointer value (ALRMPTR<1:0> bits) decrements by one until they reach '00'. Once they reach '00', the ALRMMIN and ALRMSEC value will be accessible through ALRMVALH and ALRMVALL until the pointer value is manually changed.

TABLE 19-2: ALRMVAL REGISTER MAPPING

ALRMPTR <1:0>	Alarm Value Register Window	
	ALRMVAL<15:8>	ALRMVAL<7:0>
00	ALRMMIN	ALRMSEC
01	ALRMWD	ALRMHR
10	ALRMMNTH	ALRMDAY
11	—	—

Considering that the 16-bit core does not distinguish between 8-bit and 16-bit read operations, the user must be aware that when reading either the ALRMVALH or ALRMVALL bytes, the ALRMPTR<1:0> value will be decremented. The same applies to the RTCVALH or RTCVALL bytes with the RTCPTR<1:0> being decremented.

> **Note:** This only applies to read operations and not write operations.

19.2.2 WRITE LOCK

To perform a write to any of the RTCC Timer registers, the RTCWREN bit (RCFGCAL<13>) must be set (refer to Example 19-1).

> **Note:** To avoid accidental writes to the timer, it is recommended that the RTCWREN bit (RCFGCAL<13>) is kept clear at any other time. For the RTCWREN bit to be set, there is only one instruction cycle time window allowed between the 55h/AA sequence and the setting of RTCWREN; therefore, it is recommended that code follow the procedure in Example 19-1.

19.2.3 SELECTING RTCC CLOCK SOURCE

The clock source for the RTCC module can be selected using the Flash Configuration bit, RTCOSC (CW4<5>). When the bit is set to '1', the Secondary Oscillator (SOSC) is used as the reference clock, and when the bit is '0', LPRC is used as the reference clock.

EXAMPLE 19-1: SETTING THE RTCWREN BIT

```
asm volatile("push w7");
asm volatile("push w8");
asm volatile("disi #5");
asm volatile("mov #0x55, w7");
asm volatile("mov w7, _NVMKEY");
asm volatile("mov #0xAA, w8");
asm volatile("mov w8, _NVMKEY");
asm volatile("bset _RCFGCAL, #13");   //set the RTCWREN bit
asm volatile("pop w8");
asm volatile("pop w7");
```


19.2.4 RTCC CONTROL REGISTERS

REGISTER 19-1: RCFGCAL: RTCC CALIBRATION AND CONFIGURATION REGISTER(1)

R/W-0	U-0	R/W-0	R-0, HSC	R-0, HSC	R/W-0	R/W-0	R/W-0
RTCEN(2)	—	RTCWREN	RTCSYNC	HALFSEC(3)	RTCOE	RTCPTR1	RTCPTR0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
bit 7							bit 0

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

bit 15 **RTCEN:** RTCC Enable bit(2)
1 = RTCC module is enabled
0 = RTCC module is disabled

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **RTCWREN:** RTCC Value Registers Write Enable bit
1 = RTCVALH and RTCVALL registers can be written to by the user
0 = RTCVALH and RTCVALL registers are locked out from being written to by the user

bit 12 **RTCSYNC:** RTCC Value Registers Read Synchronization bit
1 = RTCVALH, RTCVALL and ALCFGRPT registers can change while reading due to a rollover ripple resulting in an invalid data read. If the register is read twice and results in the same data, the data can be assumed to be valid.
0 = RTCVALH, RTCVALL or ALCFGRPT registers can be read without concern over a rollover ripple

bit 11 **HALFSEC:** Half Second Status bit(3)
1 = Second half period of a second
0 = First half period of a second

bit 10 **RTCOE:** RTCC Output Enable bit
1 = RTCC output is enabled
0 = RTCC output is disabled

bit 9-8 **RTCPTR<1:0>:** RTCC Value Register Window Pointer bits
Points to the corresponding RTCC Value registers when reading the RTCVALH and RTCVALL registers. The RTCPTR<1:0> value decrements on every read or write of RTCVALH until it reaches ‘00’.

RTCVAL<15:8>:
00 = MINUTES
01 = WEEKDAY
10 = MONTH
11 = Reserved

RTCVAL<7:0>:
00 = SECONDS
01 = HOURS
10 = DAY
11 = YEAR

Note 1: The RCFGCAL register is only affected by a POR.
2: A write to the RTCEN bit is only allowed when RTCWREN = 1.
3: This bit is read-only; it is cleared to ‘0’ on a write to the lower half of the MINSEC register.

REGISTER 19-1: RCFGCAL: RTCC CALIBRATION AND CONFIGURATION REGISTER[1] (CONTINUED)

bit 7-0 **CAL<7:0>:** RTC Drift Calibration bits

`01111111` = Maximum positive adjustment; adds 508 RTC clock pulses every one minute
.
.
.
`01111111` = Minimum positive adjustment; adds 4 RTC clock pulses every one minute
`00000000` = No adjustment
`11111111` = Minimum negative adjustment; subtracts 4 RTC clock pulses every one minute
.
.
.
`10000000` = Maximum negative adjustment; subtracts 512 RTC clock pulses every one minute

Note 1: The RCFGCAL register is only affected by a POR.
2: A write to the RTCEN bit is only allowed when RTCWREN = `1`.
3: This bit is read-only; it is cleared to '`0`' on a write to the lower half of the MINSEC register.

REGISTER 19-2: PADCFG1: PAD CONFIGURATION CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	RTSECSEL1[1]	RTSECSEL0[1]	PMPTTL
bit 7							bit 0

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

bit 15-3 **Unimplemented:** Read as '`0`'

bit 2-1 **RTSECSEL<1:0>:** RTCC Seconds Clock Output Select bits[1]

`11` = Reserved; do not use
`10` = RTCC source clock is selected for the RTCC pin (clock can be LPRC or SOSC, depending on the setting of the RTCOSC bit (CW4<5>))
`01` = RTCC seconds clock is selected for the RTCC pin
`00` = RTCC alarm pulse is selected for the RTCC pin

bit 0 **PMPTTL:** PMP Module TTL Input Buffer Select bit

`1` = PMP module uses TTL input buffers
`0` = PMP module uses Schmitt Trigger input buffers

Note 1: To enable the actual RTCC output, the RTCOE (RCFGCAL<10>) bit needs to be set.

REGISTER 19-3: ALCFGRPT: ALARM 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
ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0
bit 7							bit 0

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

bit 15 **ALRMEN:** Alarm Enable bit

1 = Alarm is enabled (cleared automatically after an alarm event whenever ARPT<7:0> = 00h and CHIME = 0)
0 = Alarm is disabled

bit 14 **CHIME:** Chime Enable bit

1 = Chime is enabled; ARPT<7:0> bits are allowed to roll over from 00h to FFh
0 = Chime is disabled; ARPT<7:0> bits stop once they reach 00h

bit 13-10 **AMASK<3:0>:** Alarm Mask Configuration bits

0000 = Every half second
0001 = Every second
0010 = Every 10 seconds
0011 = Every minute
0100 = Every 10 minutes
0101 = Every hour
0110 = Once a day
0111 = Once a week
1000 = Once a month
1001 = Once a year (except when configured for February 29th, once every 4 years)
101x = Reserved; do not use
11xx = Reserved; do not use

bit 9-8 **ALRMPTR<1:0>:** Alarm Value Register Window Pointer bits

Points to the corresponding Alarm Value registers when reading the ALRMVALH and ALRMVALL registers. The ALRMPTR<1:0> value decrements on every read or write of ALRMVALH until it reaches ‘00’.

ALRMVAL<15:8>:
00 = ALRMMIN
01 = ALRMWD
10 = ALRMMNTH
11 = Unimplemented

ALRMVAL<7:0>:
00 = ALRMSEC
01 = ALRMHR
10 = ALRMDAY
11 = Unimplemented

bit 7-0 **ARPT<7:0>:** Alarm Repeat Counter Value bits

11111111 = Alarm will repeat 255 more times
.
.
.
00000000 = Alarm will not repeat

The counter decrements on any alarm event; it is prevented from rolling over from 00h to FFh unless CHIME = 1.

19.2.5 RTCVAL REGISTER MAPPINGS

REGISTER 19-4: YEAR: YEAR VALUE REGISTER(1)

U-0, HSC	U-0, HSC	U-0, HSC	U-0, HSC	U-0, HSC	U-0, HSC	U-0, HSC	U-0, HSC
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
YRTEN3	YRTEN2	YRTEN1	YRTEN0	YRONE3	YRONE2	YRONE1	YRONE0
bit 7							bit 0

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

bit 15-8 **Unimplemented:** Read as '0'

bit 7-4 **YRTEN<3:0>:** Binary Coded Decimal Value of Year's Tens Digit bits
Contains a value from 0 to 9.

bit 3-0 **YRONE<3:0>:** Binary Coded Decimal Value of Year's Ones Digit bits
Contains a value from 0 to 9.

Note 1: A write to the YEAR register is only allowed when RTCWREN = 1.

REGISTER 19-5: MTHDY: MONTH AND DAY VALUE REGISTER(1)

U-0, HSC	U-0, HSC	U-0, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15							bit 8

U-0, HSC	U-0, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as '0'

bit 12 **MTHTEN0:** Binary Coded Decimal Value of Month's Tens Digit bit
Contains a value of 0 or 1.

bit 11-8 **MTHONE<3:0>:** Binary Coded Decimal Value of Month's Ones Digit bits
Contains a value from 0 to 9.

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

bit 5-4 **DAYTEN<1:0>:** Binary Coded Decimal Value of Day's Tens Digit bits
Contains a value from 0 to 3.

bit 3-0 **DAYONE<3:0>:** Binary Coded Decimal Value of Day's Ones Digit bits
Contains a value from 0 to 9.

Note 1: A write to this register is only allowed when RTCWREN = 1.

REGISTER 19-6: WKDYHR: WEEKDAY AND HOURS VALUE REGISTER[1]

U-0, HSC	U-0, HSC	U-0, HSC	U-0, HSC	U-0, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 15							bit 8

U-0, HSC	U-0, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0

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

bit 15-11 **Unimplemented:** Read as ‘0’

bit 10-8 **WDAY<2:0>:** Binary Coded Decimal Value of Weekday Digit bits
Contains a value from 0 to 6.

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

bit 5-4 **HRTEN<1:0>:** Binary Coded Decimal Value of Hour’s Tens Digit bits
Contains a value from 0 to 2.

bit 3-0 **HRONE<3:0>:** Binary Coded Decimal Value of Hour’s Ones Digit bits
Contains a value from 0 to 9.

Note 1: A write to this register is only allowed when RTCWREN = 1.

REGISTER 19-7: MINSEC: MINUTES AND SECONDS VALUE REGISTER

U-0, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15							bit 8

U-0, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **MINTEN<2:0>:** Binary Coded Decimal Value of Minute’s Tens Digit bits
Contains a value from 0 to 5.

bit 11-8 **MINONE<3:0>:** Binary Coded Decimal Value of Minute’s Ones Digit bits
Contains a value from 0 to 9.

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **SECTEN<2:0>:** Binary Coded Decimal Value of Second’s Tens Digit bits
Contains a value from 0 to 5.

bit 3-0 **SECONE<3:0>:** Binary Coded Decimal Value of Second’s Ones Digit bits
Contains a value from 0 to 9.

19.2.6 ALRMVAL REGISTER MAPPINGS

REGISTER 19-8: ALMTHDY: ALARM MONTH AND DAY VALUE REGISTER(1)

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12 **MTHTEN0:** Binary Coded Decimal Value of Month’s Tens Digit bit
Contains a value of 0 or 1.

bit 11-8 **MTHONE<3:0>:** Binary Coded Decimal Value of Month’s Ones Digit bits
Contains a value from 0 to 9.

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

bit 5-4 **DAYTEN<1:0>:** Binary Coded Decimal Value of Day’s Tens Digit bits
Contains a value from 0 to 3.

bit 3-0 **DAYONE<3:0>:** Binary Coded Decimal Value of Day’s Ones Digit bits
Contains a value from 0 to 9.

Note 1: A write to this register is only allowed when RTCWREN = 1.

REGISTER 19-9: ALWDHR: ALARM WEEKDAY AND HOURS VALUE REGISTER(1)

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0

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

bit 15-11 **Unimplemented:** Read as ‘0’

bit 10-8 **WDAY<2:0>:** Binary Coded Decimal Value of Weekday Digit bits
Contains a value from 0 to 6.

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

bit 5-4 **HRTEN<1:0>:** Binary Coded Decimal Value of Hour’s Tens Digit bits
Contains a value from 0 to 2.

bit 3-0 **HRONE<3:0>:** Binary Coded Decimal Value of Hour’s Ones Digit bits
Contains a value from 0 to 9.

Note 1: A write to this register is only allowed when RTCWREN = 1.

REGISTER 19-10: ALMINSEC: ALARM MINUTES AND SECONDS VALUE REGISTER

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15							bit 8

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **MINTEN<2:0>:** Binary Coded Decimal Value of Minute’s Tens Digit bits
Contains a value from 0 to 5.

bit 11-8 **MINONE<3:0>:** Binary Coded Decimal Value of Minute’s Ones Digit bits
Contains a value from 0 to 9.

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **SECTEN<2:0>:** Binary Coded Decimal Value of Second’s Tens Digit bits
Contains a value from 0 to 5.

bit 3-0 **SECONE<3:0>:** Binary Coded Decimal Value of Second’s Ones Digit bits
Contains a value from 0 to 9.

19.3 Calibration

The real-time crystal input can be calibrated using the periodic auto-adjust feature. When properly calibrated, the RTCC can provide an error of less than 3 seconds per month. This is accomplished by finding the number of error clock pulses and storing the value into the lower half of the RCFGCAL register. The 8-bit signed value loaded into the lower half of RCFGCAL is multiplied by four and will either be added or subtracted from the RTCC timer, once every minute. Refer to the steps below for RTCC calibration:

1. Using another timer resource on the device; the user must find the error of the 32.768 kHz crystal.
2. Once the error is known, it must be converted to the number of error clock pulses per minute.
3. a) If the oscillator is faster than ideal (negative result from step 2), the RCFGCAL register value must be negative. This causes the specified number of clock pulses to be subtracted from the timer counter, once every minute.

   b) If the oscillator is slower than ideal (positive result from step 2), the RCFGCAL register value must be positive. This causes the specified number of clock pulses to be subtracted from the timer counter, once every minute.

Divide the number of error clocks per minute by 4 to get the correct calibration value and load the RCFGCAL register with the correct value. (Each 1-bit increment in the calibration adds or subtracts 4 pulses.)

EQUATION 19-1:

(Ideal Frequency† – Measured Frequency) * 60 = Clocks per Minute

† Ideal Frequency = 32,768 Hz

Writes to the lower half of the RCFGCAL register should only occur when the timer is turned off or immediately after the rising edge of the seconds pulse.

> **Note:** It is up to the user to include, in the error value, the initial error of the crystal drift due to temperature and drift due to crystal aging.

19.4 Alarm

- Configurable from half second to one year
- Enabled using the ALRMEN bit (ALCFGRPT<15>)
- One-time alarm and repeat alarm options are available

19.4.1 CONFIGURING THE ALARM

The alarm feature is enabled using the ALRMEN bit. This bit is cleared when an alarm is issued. Writes to ALRMVAL should only take place when ALRMEN = `0`.

As displayed in Figure 19-2, the interval selection of the alarm is configured through the AMASK bits (ALCFGRPT<13:10>). These bits determine which and how many digits of the alarm must match the clock value for the alarm to occur.

The alarm can also be configured to repeat based on a preconfigured interval. The amount of times this occurs, once the alarm is enabled, is stored in the ARPT<7:0> bits (ALCFGRPT<7:0>). When the value of the ARPT bits equals 00h and the CHIME bit (ALCFGRPT<14>) is cleared, the repeat function is disabled and only a single alarm will occur. The alarm can be repeated up to 255 times by loading ARPT<7:0> with FFh.

After each alarm is issued, the value of the ARPT bits is decremented by one. Once the value has reached 00h, the alarm will be issued one last time, after which, the ALRMEN bit will be cleared automatically and the alarm will turn off.

Indefinite repetition of the alarm can occur if the CHIME bit = `1`. Instead of the alarm being disabled when the value of the ARPT bits reaches 00h, it rolls over to FFh and continues counting indefinitely while CHIME is set.

19.4.2 ALARM INTERRUPT

At every alarm event, an interrupt is generated. In addition, an alarm pulse output is provided that operates at half the frequency of the alarm. This output is completely synchronous to the RTCC clock and can be used as a trigger clock to other peripherals.

> **Note:** Changing any of the registers, other than the RCFGCAL and ALCFGRPT registers, and the CHIME bit while the alarm is enabled (ALRMEN = `1`), can result in a false alarm event leading to a false alarm interrupt. To avoid a false alarm event, the timer and alarm values should only be changed while the alarm is disabled (ALRMEN = `0`). It is recommended that the ALCFGRPT register and CHIME bit be changed when RTCSYNC = `0`.

FIGURE 19-2: ALARM MASK SETTINGS

Alarm Mask Setting (AMASK<3:0>)	Day of the Week	Month	Day	Hours	Minutes	Seconds
0000 - Every half second 0001 - Every second						
0010 - Every 10 seconds						s
0011 - Every minute						s s
0100 - Every 10 minutes					m	s s
0101 - Every hour					m m	s s
0110 - Every day				h h	m m	s s
0111 - Every week	d			h h	m m	s s
1000 - Every month			d d	h h	m m	s s
1001 - Every year[(1)]		m m	d d	h h	m m	s s

Note 1: Annually, except when configured for February 29.

NOTES:

20.0 32-BIT PROGRAMMABLE CYCLIC REDUNDANCY CHECK (CRC) GENERATOR

Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 41. "32-Bit Programmable Cyclic Redundancy Check (CRC)"** (DS39729).

The programmable CRC generator provides a hardware-implemented method of quickly generating checksums for various networking and security applications. It offers the following features:

- User-programmable CRC polynomial equation, up to 32 bits
- Programmable shift direction (little or big-endian)
- Independent data and polynomial lengths
- Configurable interrupt output
- Data FIFO

A simplified block diagram of the CRC generator is shown in Figure 20-1. A simple version of the CRC shift engine is shown in Figure 20-2.

FIGURE 20-1: CRC BLOCK DIAGRAM

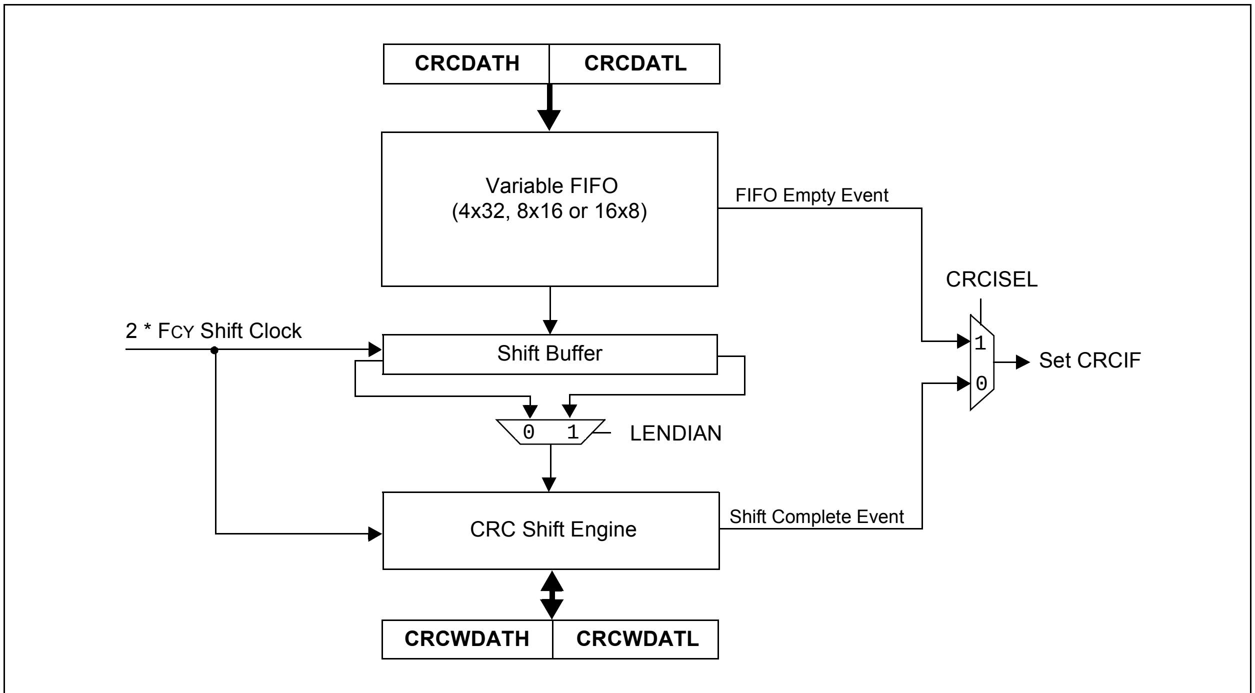
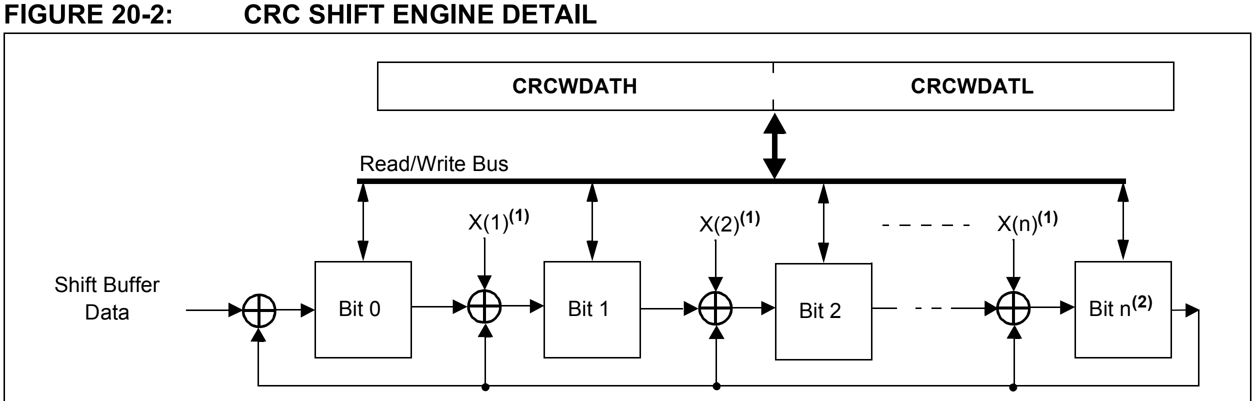


FIGURE 20-2: CRC SHIFT ENGINE DETAIL



Note 1: Each XOR stage of the shift engine is programmable. See text for details.

2: Polynomial length n is determined by ([PLEN<3:0>] + 1)

20.1 User Interface

20.1.1 POLYNOMIAL INTERFACE

The CRC module can be programmed for CRC polynomials of up to the 32nd order, using up to 32 bits. Polynomial length, which reflects the highest exponent in the equation, is selected by the PLEN<4:0> bits (CRCCON2<4:0>).

The CRCXORL and CRCXORH registers control which exponent terms are included in the equation. Setting a particular bit includes that exponent term in the equation; functionally, this includes an XOR operation on the corresponding bit in the CRC engine. Clearing the bit disables the XOR.

For example, consider two CRC polynomials, one a 16-bit equation and the other, a 32-bit equation:

$$x16 + x12 + x5 + 1$$

and

$$x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1$$

To program these polynomials into the CRC generator, set the register bits as shown in Table 20-1.

Note that the appropriate positions are set to '1' to indicate that they are used in the equation (for example, X26 and X23). The 0 bit required by the equation is always XORed; thus, X0 is a don't care. For a polynomial of length N, it is assumed that the Nth bit will always be used, regardless of the bit setting. Therefore, for a polynomial length of 32, there is no 32nd bit in the CRCxOR register.

20.1.2 DATA INTERFACE

The module incorporates a FIFO that works with a variable data width. Input data width can be configured to any value between one and 32 bits using the DWIDTH<4:0> bits (CRCCON2<12:8>). When the data width is greater than 15, the FIFO is four words deep. When the DWIDTH value is between 15 and 8, the FIFO is 8 words deep. When the DWIDTH value is less than 8, the FIFO is 16 words deep.

The data for which the CRC is to be calculated must first be written into the FIFO. Even if the data width is less than 8, the smallest data element that can be written into the FIFO is one byte. For example, if the DWIDTH value is five, then the size of the data is DWIDTH + 1, or six. The data is written as a whole byte; the two unused upper bits are ignored by the module.

Once data is written into the MSb of the CRCDAT registers (that is, MSb as defined by the data width), the value of the VWORD<4:0> bits (CRCCON1<12:8>) increments by one. For example, if the DWIDTH value is 24, the VWORD bits will increment when bit 7 of CRCDATH is written. Therefore, CRCDATL must always be written before CRCDATH.

The CRC engine starts shifting data when the CRCGO bit is set and the value of VWORD is greater than zero. Each word is copied out of the FIFO into a buffer register, which decrements VWORD. The data is then shifted out of the buffer. The CRC engine continues shifting at a rate of two bits per instruction cycle, until the VWORD value reaches zero. This means that for a given data width, it takes half that number of instructions for each word to complete the calculation. For example, it takes 16 cycles to calculate the CRC for a single word of 32-bit data.

When the VWORD value reaches the maximum value for the configured value of DWIDTH (4, 8 or 16), the CRCFUL bit becomes set. When the VWORD value reaches zero, the CRCMPT bit becomes set. The FIFO is emptied and the VWORD<4:0> bits are set to '00000' whenever CRCEN is '0'.

At least one instruction cycle must pass, after a write to CRCDAT, before a read of the VWORD bits is done.

TABLE 20-1: CRC SETUP EXAMPLES FOR 16 AND 32-BIT POLYNOMIAL

CRC Control Bits	Bit Values	
	16-Bit Polynomial	32-Bit Polynomial
PLEN<4:0>	01111	11111
X<31:16>	0000 0000 0000 000x	0000 0100 1100 0001
X<15:0>	0001 0000 0010 000x	0001 1101 1011 011x

20.1.3 DATA SHIFT DIRECTION

The LENDIAN bit (CRCCON1<3>) is used to control the shift direction. By default, the CRC will shift data through the engine, MSb first. Setting LENDIAN (= 1) causes the CRC to shift data, LSb first. This setting allows better integration with various communication schemes and removes the overhead of reversing the bit order in software. Note that this only changes the direction of the data that is shifted into the engine. The result of the CRC calculation will still be a normal CRC result, not a reverse CRC result.

20.1.4 INTERRUPT OPERATION

The module generates an interrupt that is configurable by the user for either of two conditions.

If CRCISEL is '0', an interrupt is generated when the VWORD<4:0> bits make a transition from a value of '1' to '0'. If CRCISEL is '1', an interrupt will be generated after the CRC operation finishes and the module sets the CRCGO bit to '0'. Manually setting CRCGO to '0' will not generate an interrupt.

20.1.5 TYPICAL OPERATION

To use the module for a typical CRC calculation:

1. Set the CRCEN bit to enable the module.
2. Configure the module for the desired operation:
    d) Program the desired polynomial using the CRCXORL and CRCXORH registers, and the PLEN<4:0> bits
    e) Configure the data width and shift direction using the DWIDTH and LENDIAN bits
    f) Select the desired interrupt mode using the CRCISEL bit
3. Preload the FIFO by writing to the CRCDATL and CRCDATH registers until the CRCFUL bit is set or no data is left
4. Clear old results by writing 00h to CRCWDATL and CRCWDATH. CRCWDAT can also be left unchanged to resume a previously halted calculation.
5. Set the CRCGO bit to start calculation.
6. Write remaining data into the FIFO as space becomes available.
7. When the calculation completes, CRCGO is automatically cleared. An interrupt will be generated if CRCISEL = 1.
8. Read CRCWDATL and CRCWDATH for the result of the calculation.

20.2 Registers

There are eight registers associated with the module:

- CRCCON1
- CRCCON2
- CRCXORL
- CRCXORH
- CRCDATL
- CRCDATH
- CRCWDATL
- CRCWDATH

The CRCCON1 and CRCCON2 registers (Register 20-1 and Register 20-2) control the operation of the module, and configure the various settings. The CRCXOR registers (Register 20-3 and Register 20-4) select the polynomial terms to be used in the CRC equation. The CRCDAT and CRCWDAT registers are each register pairs that serve as buffers for the double-word, input data and CRC processed output, respectively.

REGISTER 20-1: CRCCON1: CRC CONTROL REGISTER 1

R/W-0	U-0	R/W-0	R-0	R-0	R-0	R-0	R-0
CRCEN	—	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0
bit 15							bit 8

R-0, HCS	R-1, HCS	R/W-0	R/W-0, HC	R/W-0	U-0	U-0	U-0
CRCFUL	CRCMPT	CRCISEL	CRCGO	LENDIAN	—	—	—
bit 7							bit 0

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

bit 15 **CRCEN**: CRC Enable bit
1 = Module is enabled
0 = Module is enabled. All state machines, pointers and CRCWDAT/CRCDAT are reset; other SFRs are NOT reset.

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **CSIDL:** CRC Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-8 **VWORD<4:0>:** Pointer Value bits
Indicates the number of valid words in the FIFO. Has a maximum value of 8 when PLEN<3:0> > 7, or 16 when PLEN<3:0> ≤ 7.

bit 7 **CRCFUL**: FIFO Full bit
1 = FIFO is full
0 = FIFO is not full

bit 6 **CRCMPT:** FIFO Empty Bit
1 = FIFO is empty
0 = FIFO is not empty

bit 5 **CRCISEL:** CRC interrupt Selection bit
1 = Interrupt on FIFO is empty; CRC calculation is not complete
0 = Interrupt on shift is complete and CRCWDAT result is ready

bit 4 **CRCGO:** Start CRC bit
1 = Start CRC serial shifter
0 = CRC serial shifter is turned off

bit 3 **LENDIAN:** Data Shift Direction Select bit
1 = Data word is shifted into the CRC starting with the LSb (little endian)
0 = Data word is shifted into the CRC starting with the MSb (big endian)

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

REGISTER 20-2: CRCCON2: CRC CONTROL REGISTER 2

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DWIDTH4	DWIDTH3	DWIDTH2	DWIDTH1	DWIDTH0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	PLEN4	PLEN3	PLEN2	PLEN1	PLEN0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **DWIDTH<4:0>:** Data Width Select bits
Defines the width of the data word (Data Word Width = (DWIDTH<4:0>) + 1).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **PLEN<4:0>:** Polynomial Length Select bits
Defines the length of the CRC polynomial (Polynomial Length = (PLEN<4:0>) + 1).

REGISTER 20-3: CRCXORL: CRC XOR POLYNOMIAL REGISTER, LOW BYTE

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X15	X14	X13	X12	X11	X10	X9	X8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
X7	X6	X5	X4	X3	X2	X1	—
bit 7							bit 0

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

bit 15-1 **X<15:1>:** XOR of Polynomial Term X^n Enable bits

bit 0 **Unimplemented:** Read as ‘0’

REGISTER 20-4: CRCXORH: CRC XOR POLYNOMIAL REGISTER, HIGH BYTE

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X31	X30	X29	X28	X27	X26	X25	X24
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X23	X22	X21	X20	X19	X18	X17	X16
bit 7							bit 0

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

bit 15-0 **X<31:16>:** XOR of Polynomial Term X^n Enable bits

21.0 10-BIT HIGH-SPEED A/D CONVERTER

Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 17. "10-Bit A/D Converter"** (DS39705).

The 10-bit A/D Converter has the following key features:

- Successive Approximation (SAR) conversion
- Conversion speeds of up to 500 ksps
- 13 analog input pins
- External voltage reference input pins
- Internal band gap reference inputs
- Automatic Channel Scan mode
- Selectable conversion trigger source
- 16-word conversion result buffer
- Selectable Buffer Fill modes
- Four result alignment options
- Operation during CPU Sleep and Idle modes

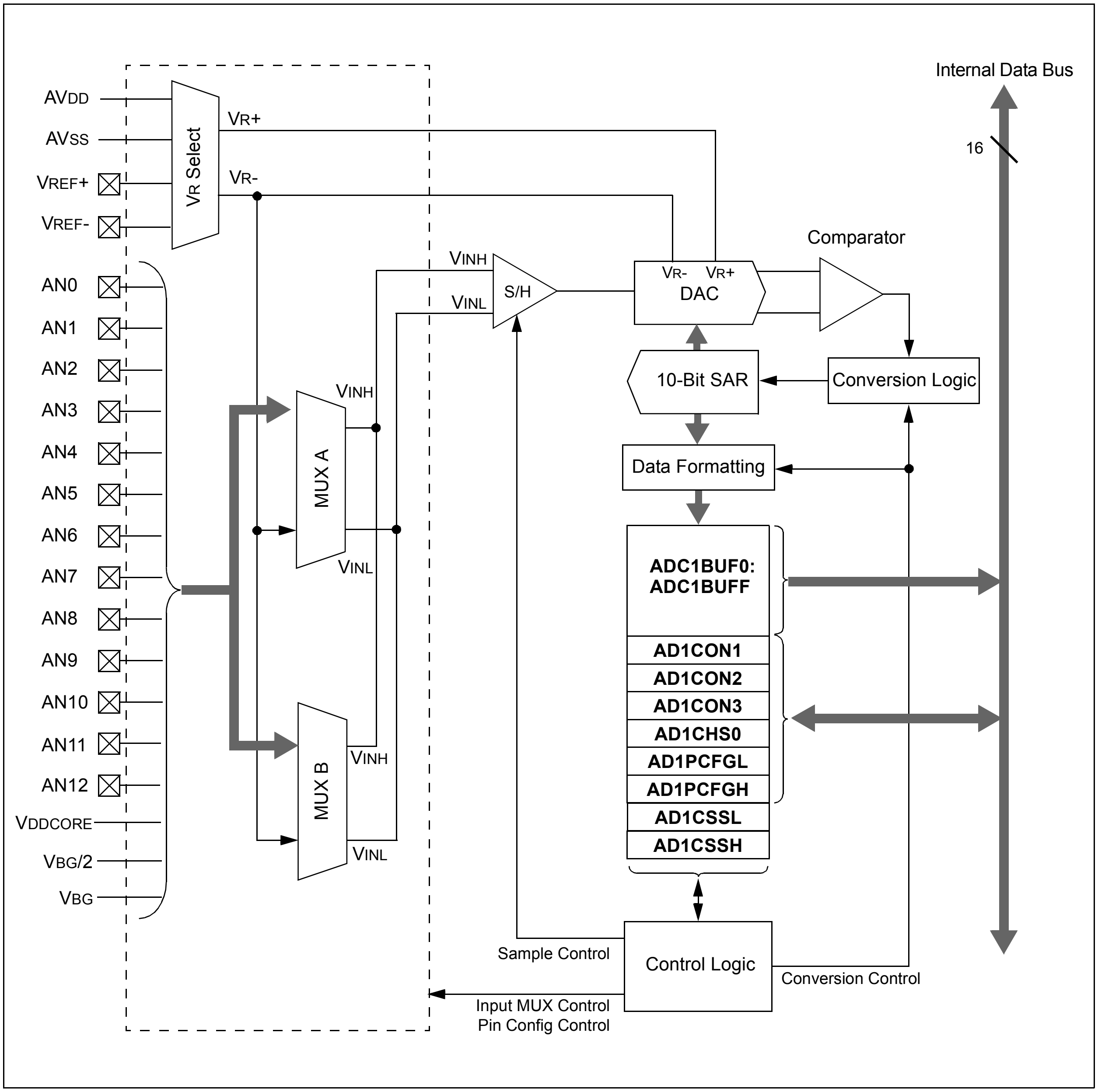
On all PIC24FJ64GA104 family devices, the 10-bit A/D Converter has 13 analog input pins, designated AN0 through AN12. In addition, there are two analog input pins for external voltage reference connections (VREF+ and VREF-). These voltage reference inputs may be shared with other analog input pins.

A block diagram of the A/D Converter is shown in Figure 21-1.

To perform an A/D conversion:

1. Configure the A/D module:
   a) Configure port pins as analog inputs and/or select band gap reference inputs (AD1PCFGL<15:0> and AD1PCFGH<1:0>).
   b) Select voltage reference source to match expected range on analog inputs (AD1CON2<15:13>).
   c) Select the analog conversion clock to match the desired data rate with the processor clock (AD1CON3<7:0>).
   d) Select the appropriate sample/conversion sequence (AD1CON1<7:5> and AD1CON3<12:8>).
   e) Select how conversion results are presented in the buffer (AD1CON1<9:8>).
   f) Select interrupt rate (AD1CON2<5:2>).
   g) Turn on A/D module (AD1CON1<15>).
2. Configure the A/D interrupt (if required):
   a) Clear the AD1IF bit.
   b) Select A/D interrupt priority.

FIGURE 21-1: 10-BIT HIGH-SPEED A/D CONVERTER BLOCK DIAGRAM



REGISTER 21-1: AD1CON1: A/D CONTROL REGISTER 1

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
ADON[(1)]	—	ADSIDL	—	—	—	FORM1	FORM0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0, HCS	R/C-0, HCS
SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE
bit 7							bit 0

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

bit 15 **ADON:** A/D Operating Mode bit[(1)]
1 = A/D Converter module is operating
0 = A/D Converter is off

bit 14 **Unimplemented:** Read as '0'

bit 13 **ADSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-10 **Unimplemented:** Read as '0'

bit 9-8 **FORM<1:0>:** Data Output Format bits
11 = Signed fractional (sddd dddd dd00 0000)
10 = Fractional (dddd dddd dd00 0000)
01 = Signed integer (ssss sssd dddd dddd)
00 = Integer (0000 00dd dddd dddd)

bit 7-5 **SSRC<2:0>:** Conversion Trigger Source Select bits
111 = Internal counter ends sampling and starts conversion (auto-convert)
110 = CTMU event ends sampling and starts conversion
101 = Reserved
100 = Timer5 compare ends sampling and starts conversion
011 = Reserved
010 = Timer3 compare ends sampling and starts conversion
001 = Active transition on INT0 pin ends sampling and starts conversion
000 = Clearing the SAMP bit ends sampling and starts conversion

bit 4-3 **Unimplemented:** Read as '0'

bit 2 **ASAM:** A/D Sample Auto-Start bit
1 = Sampling begins immediately after the last conversion completes; SAMP bit is auto-set
0 = Sampling begins when the SAMP bit is set

bit 1 **SAMP:** A/D Sample Enable bit
1 = A/D sample/hold amplifier is sampling input
0 = A/D sample/hold amplifier is holding

bit 0 **DONE:** A/D Conversion Status bit
1 = A/D conversion is done
0 = A/D conversion is NOT done

Note 1: Values of ADC1BUFx registers will not retain their values once the ADON bit is cleared. Read out the conversion values from the buffer before disabling the module.

REGISTER 21-2: AD1CON2: A/D CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	r-0	U-0	R/W-0	U-0	U-0
VCFG2	VCFG1	VCFG0	r	—	CSCNA	—	—
bit 15							bit 8

R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUFS	—	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS
bit 7							bit 0

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

bit 15-13 **VCFG<2:0>:** Voltage Reference Configuration bits

VCFG<2:0>	VR+	VR-
000	AVDD	AVSS
001	External VREF+ pin	AVSS
010	AVDD	External VREF- pin
011	External VREF+ pin	External VREF- pin
1xx	AVDD	AVSS

bit 12 **Reserved:** Maintain as ‘0’

bit 11 **Unimplemented:** Read as ‘0’

bit 10 **CSCNA:** Scan Input Selections for CH0+ S/H Input for MUX A Input Multiplexer Setting bit
1 = Scan inputs
0 = Do not scan inputs

bit 9-8 **Unimplemented:** Read as ‘0’

bit 7 **BUFS:** Buffer Fill Status bit (valid only when BUFM = 1)
1 = A/D is currently filling buffer 08-0F; user should access data in 00-07
0 = A/D is currently filling buffer 00-07; user should access data in 08-0F

bit 6 **Unimplemented:** Read as ‘0’

bit 5-2 **SMPI<3:0>:** Sample/Convert Sequences Per Interrupt Selection bits
1111 = Interrupts at the completion of conversion for each 16th sample/convert sequence
1110 = Interrupts at the completion of conversion for each 15th sample/convert sequence
.....
0001 = Interrupts at the completion of conversion for each 2nd sample/convert sequence
0000 = Interrupts at the completion of conversion for each sample/convert sequence

bit 1 **BUFM:** Buffer Mode Select bit
1 = Buffer is configured as two 8-word buffers (ADC1BUFn<15:8> and ADC1BUFn<7:0>)
0 = Buffer is configured as one 16-word buffer (ADC1BUFn<15:0>)

bit 0 **ALTS:** Alternate Input Sample Mode Select bit
1 = Uses MUX A input multiplexer settings for first sample, then alternates between MUX B and MUX A input multiplexer settings for all subsequent samples
0 = Always uses MUX A input multiplexer settings

REGISTER 21-3: AD1CON3: A/D CONTROL REGISTER 3

R/W-0	r-0	r-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	r	r	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0
bit 7							bit 0

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

bit 15 **ADRC:** A/D Conversion Clock Source bit
1 = A/D internal RC clock
0 = Clock derived from system clock

bit 14-13 **Reserved**: Maintain as '0'

bit 12-8 **SAMC<4:0>:** Auto-Sample Time bits
11111 = 31 TAD
.....
00001 = 1 TAD
00000 = 0 TAD (not recommended)

bit 7-0 **ADCS<7:0>:** A/D Conversion Clock Select bits
11111111 to 01000000 = Reserved
......
00111111 = 64 • TCY
......
00000001 = 2 • TCY
00000000 = TCY

REGISTER 21-4: AD1CHS: A/D INPUT SELECT REGISTER

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NB	—	—	CH0SB4[1,2]	CH0SB3[1,2]	CH0SB2[1,2]	CH0SB1[1,2]	CH0SB0[1,2]
bit 15							bit 8

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NA	—	—	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0
bit 7							bit 0

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

bit 15 **CH0NB:** Channel 0 Negative Input Select for MUX B Multiplexer Setting bit
1 = Channel 0 negative input is AN1
0 = Channel 0 negative input is V_{R-}

bit 14-13 **Unimplemented:** Read as ‘0’

bit 12-8 **CH0SB<4:0>:** Channel 0 Positive Input Select for MUX B Multiplexer Setting bits[1,2]
11111 = Channel 0 positive input is reserved for CTMU use only[3]
1xxxx = Unimplemented; do not use.
01111 = Channel 0 positive input is internal band gap reference (V_{BG})
01110 = Channel 0 positive input is $V_{BG}/2$
01101 = Channel 0 positive input is voltage regulator output (V_{DDCORE})
01100 = Channel 0 positive input is AN12
01011 = Channel 0 positive input is AN11
01010 = Channel 0 positive input is AN10
01001 = Channel 0 positive input is AN9
01000 = Channel 0 positive input is AN8
00111 = Channel 0 positive input is AN7
00110 = Channel 0 positive input is AN6
00101 = Channel 0 positive input is AN5
00100 = Channel 0 positive input is AN4
00011 = Channel 0 positive input is AN3
00010 = Channel 0 positive input is AN2
00001 = Channel 0 positive input is AN1
00000 = Channel 0 positive input is AN0

bit 7 **CH0NA:** Channel 0 Negative Input Select for MUX A Multiplexer Setting bit
1 = Channel 0 negative input is AN1
0 = Channel 0 negative input is V_{R-}

bit 6-5 **Unimplemented:** Read as ‘0’

bit 4-0 **CH0SA<4:0>:** Channel 0 Positive Input Select for MUX A Multiplexer Setting bits
Implemented combinations are identical to those for CH0SB<4:0> (above).

Note 1: Combinations not shown here are unimplemented; do not use.
2: Analog channels, AN6, AN7, AN8 and AN12, are unavailable on 28-pin devices; do not use.
3: Selecting this internal channel allows the CTMU module to utilize the A/D Converter sample and hold capacitor (C_{AD}) for the smallest time measurements.

REGISTER 21-5: AD1PCFG: A/D PORT CONFIGURATION REGISTER

R/W-0	R/W-0	R/W-0	R/W-0(1)	R/W-0	R/W-0	R/W-0	R/W-0(1)
PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
bit 15							bit 8

R/W-0(1)	R/W-0(1)	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

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

bit 15 **PCFG15:** A/D Input Band Gap Reference Enable bit
1 = Internal band gap (V_{BG}) reference channel is disabled
0 = Internal band gap reference channel is enabled

bit 14 **PCFG14:** A/D Input Half Band Gap Reference Enable bit
1 = Internal half band gap ($V_{BG}/2$) reference channel is disabled
0 = Internal half band gap reference channel is enabled

bit 13 **PCFG13:** A/D Input Voltage Regulator Output Reference Enable bit
1 = Internal voltage regulator output (V_{DDCORE}) reference channel is disabled
0 = Internal voltage regulator output reference channel is enabled

bit 12-0 **PCFG<12:0>:** Analog Input Pin Configuration Control bits(1)
1 = Pin for corresponding analog channel is configured in Digital mode; I/O port read is enabled
0 = Pin is configured in Analog mode; I/O port read is disabled, A/D samples pin voltage

Note 1: Analog channels, AN6, AN7, AN8 and AN12, are unavailable on 28-pin devices; leave these corresponding bits set.

REGISTER 21-6: AD1CSSL: A/D INPUT SCAN SELECT REGISTER

R/W-0	R/W-0	R/W-0	R/W-0[1]	R/W-0	R/W-0	R/W-0	R/W-0
CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8[1]
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0
bit 7							bit 0

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

bit 15 **CSSL15:** A/D Input Band Gap Scan Enable bit
1 = Internal band gap (V_{BG}) channel is enabled for input scan
0 = Analog channel is disabled from input scan

bit 14 **CSSL14:** A/D Input Half Band Gap Scan Enable bit
1 = Internal half band gap ($V_{BG}/2$) channel is enabled for input scan
0 = Analog channel is disabled from input scan

bit 13 **CSSL13:** A/D Input Voltage Regulator Output Scan Enable bit
1 = Internal voltage regulator output (V_{DDCORE}) is enabled for input scan
0 = Analog channel is disabled from input scan

bit 12-0 **CSSL<12:0>:** A/D Input Pin Scan Selection bits[1]
1 = Corresponding analog channel is selected for input scan
0 = Analog channel is omitted from input scan

Note 1: Analog channels, AN6, AN7, AN8 and AN12, are unavailable on 28-pin devices; leave these corresponding bits cleared.

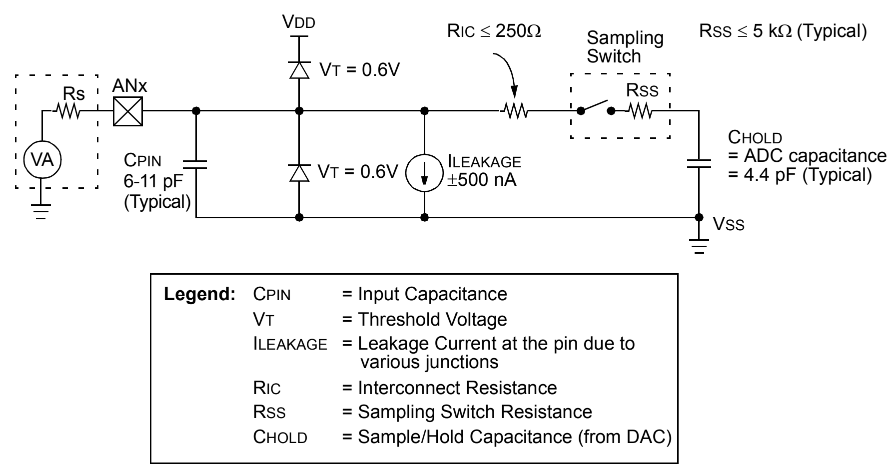
EQUATION 21-1: A/D CONVERSION CLOCK PERIOD(1)

$$ADCS = \frac{T_{AD}}{T_{CY}} - 1$$

$$T_{AD} = T_{CY} \bullet (ADCS + 1)$$

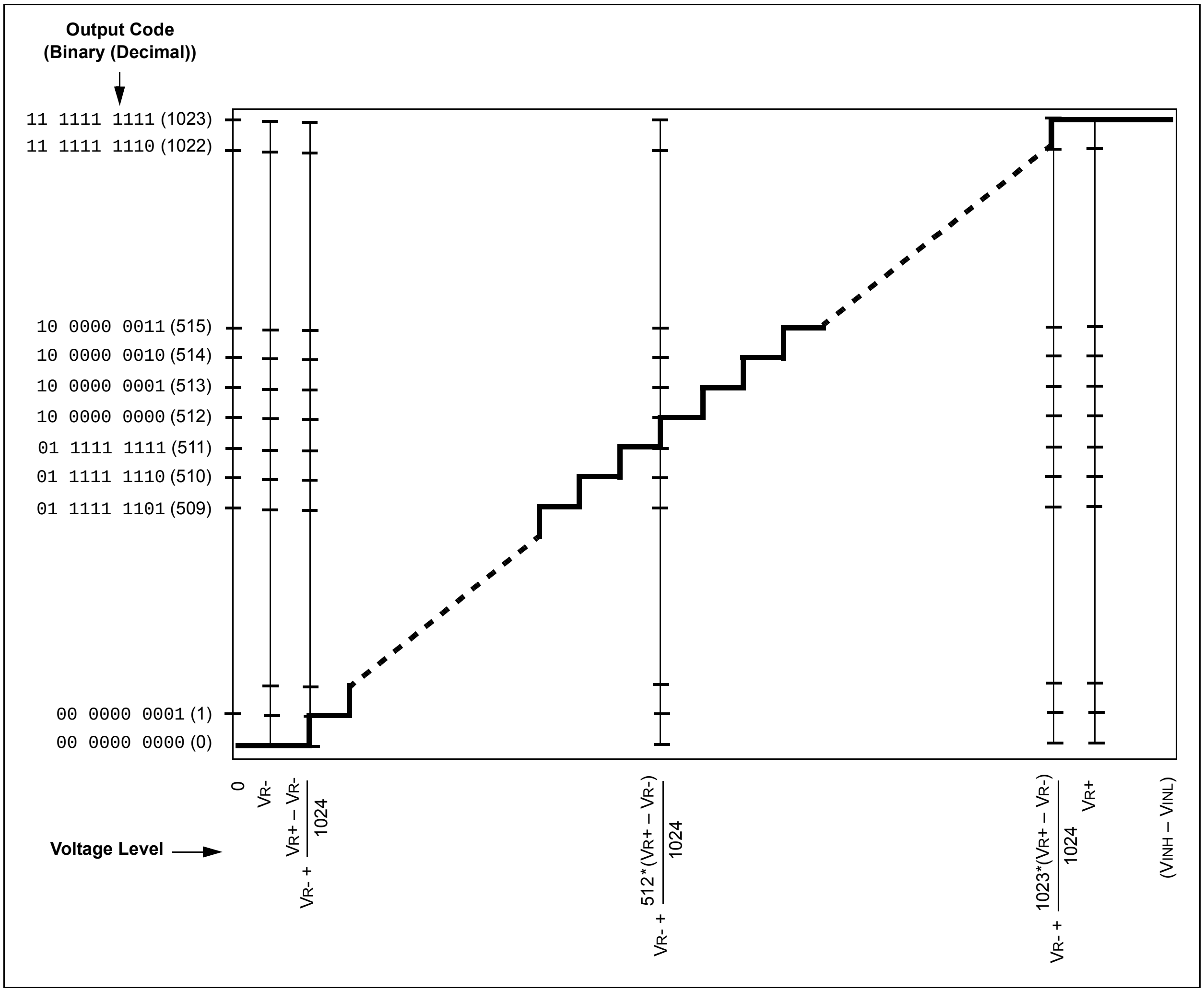
Note 1: Based on TCY = 2 * TOSC, Doze mode and PLL are disabled.

FIGURE 21-2: 10-BIT A/D CONVERTER ANALOG INPUT MODEL



Note: CPIN value depends on the device package and is not tested. The effect of CPIN is negligible if Rs ≤ 5 kΩ.

FIGURE 21-3: A/D TRANSFER FUNCTION



22.0 TRIPLE COMPARATOR MODULE

> **Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the associated *"PIC24F Family Reference Manual"*, **Section 46. "Scalable Comparator Module"** (DS39734)

The triple comparator module provides three dual input comparators. The inputs to the comparator can be configured to use any one of four external analog inputs, as well as voltage reference inputs from the voltage reference generator and band gap reference.

The comparator outputs may be directly connected to the CxOUT pins. When the respective COE equals '1', the I/O pad logic makes the unsynchronized output of the comparator available on the pin.

A simplified block diagram of the module in shown in Figure 22-1. Diagrams of the possible individual comparator configurations are shown in Figure 22-2.

Each comparator has its own control register, CMxCON (Register 22-1), for enabling and configuring its operation. The output and event status of all three comparators are provided in the CMSTAT register (Register 22-2).

FIGURE 22-1: TRIPLE COMPARATOR MODULE BLOCK DIAGRAM

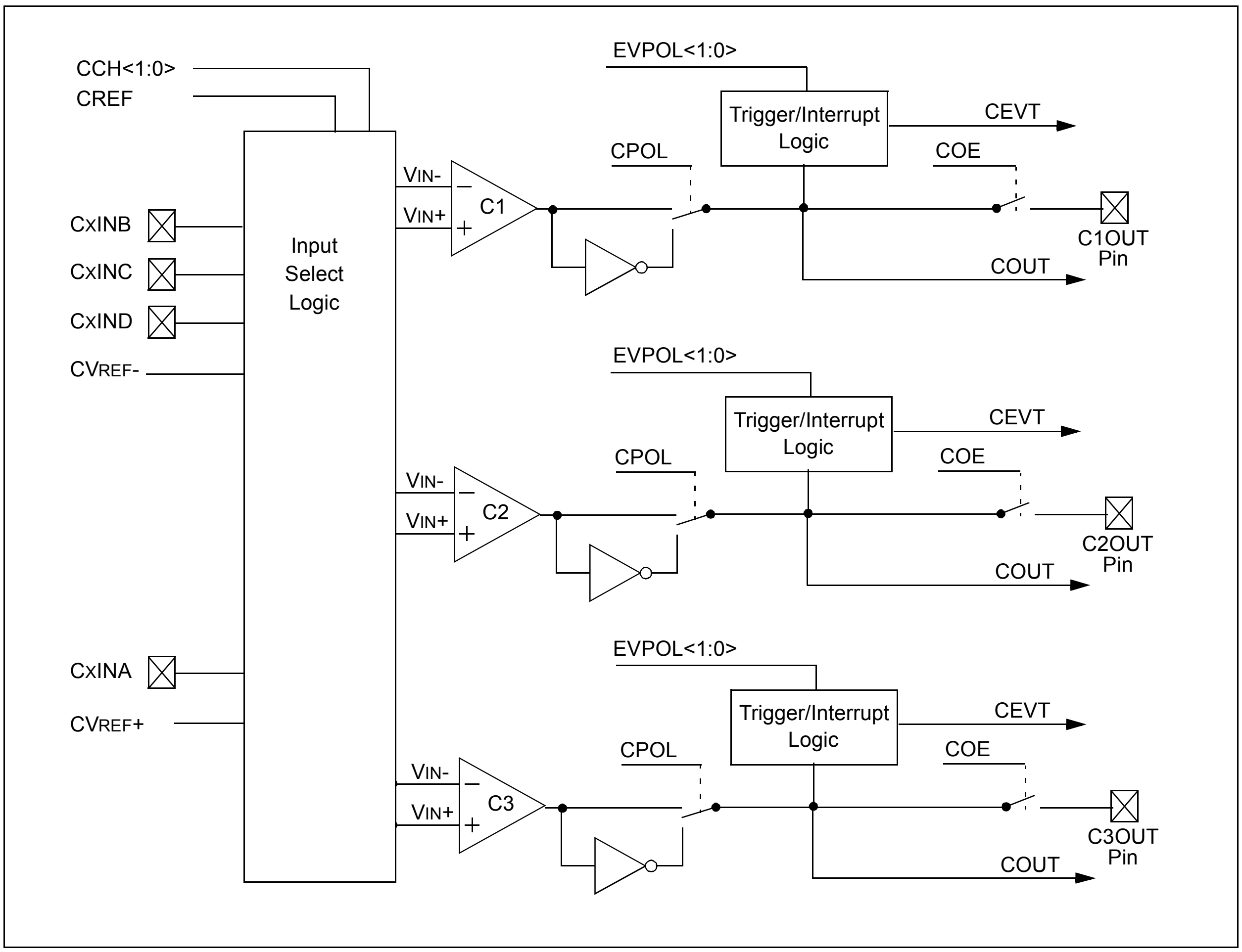
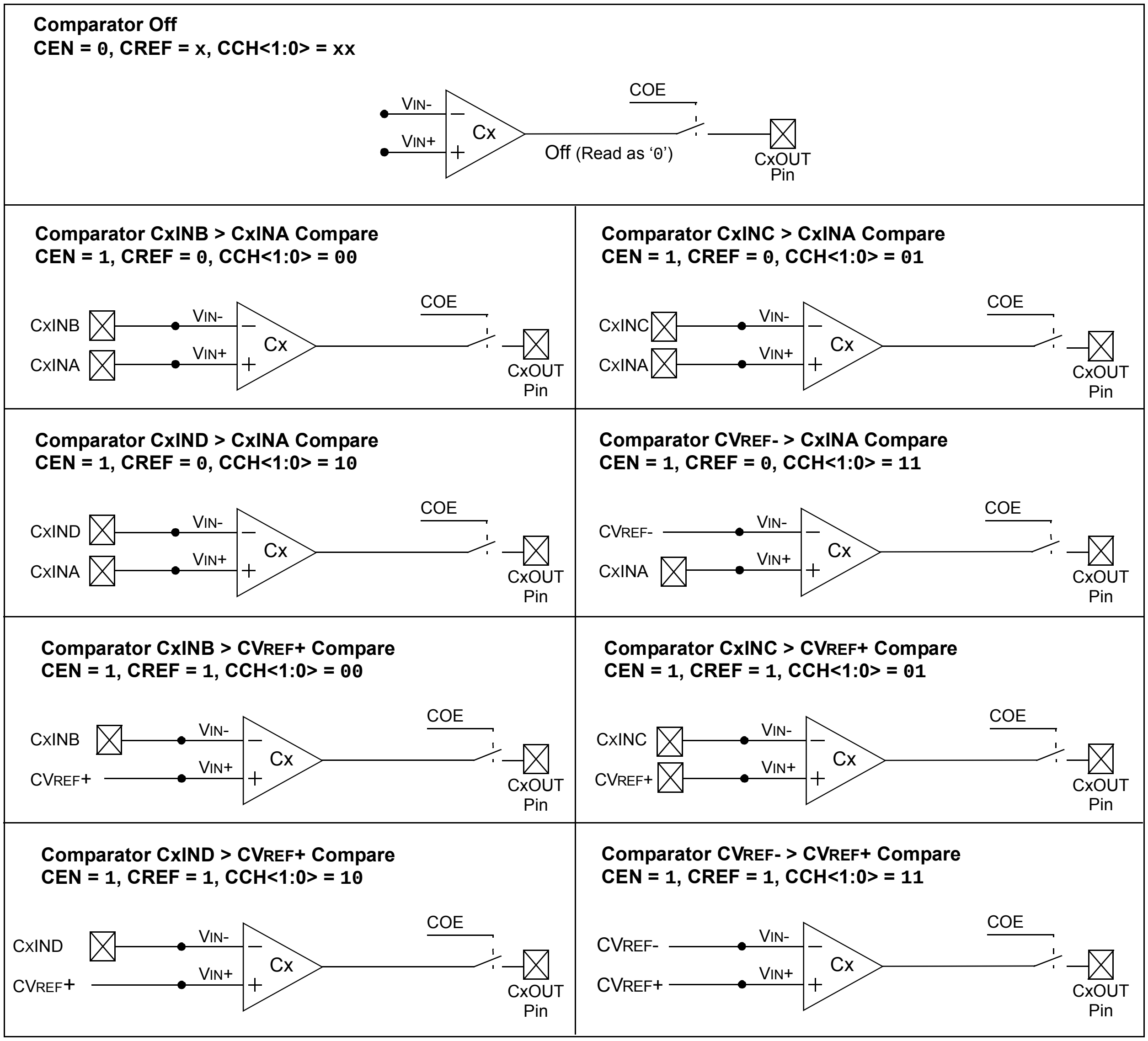


FIGURE 22-2: INDIVIDUAL COMPARATOR CONFIGURATIONS



REGISTER 22-1: CMxCON: COMPARATOR x CONTROL REGISTERS (COMPARATORS 1 THROUGH 3)

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R-0
CEN	COE	CPOL	—	—	—	CEVT	COUT
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0
bit 7							bit 0

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

bit 15 **CEN:** Comparator Enable bit
- 1 = Comparator is enabled
- 0 = Comparator is disabled

bit 14 **COE:** Comparator Output Enable bit
- 1 = Comparator output is present on the CxOUT pin.
- 0 = Comparator output is internal only

bit 13 **CPOL:** Comparator Output Polarity Select bit
- 1 = Comparator output is inverted
- 0 = Comparator output is not inverted

bit 12-10 **Unimplemented:** Read as ‘0’

bit 9 **CEVT:** Comparator Event bit
- 1 = Comparator event defined by EVPOL<1:0> has occurred; subsequent triggers and interrupts are disabled until the bit is cleared
- 0 = Comparator event has not occurred

bit 8 **COUT:** Comparator Output bit

When CPOL = 0:
- 1 = $V_{IN+} > V_{IN-}$
- 0 = $V_{IN+} < V_{IN-}$

When CPOL = 1:
- 1 = $V_{IN+} < V_{IN-}$
- 0 = $V_{IN+} > V_{IN-}$

bit 7-6 **EVPOL<1:0>:** Trigger/Event/Interrupt Polarity Select bits
- 11 = Trigger/event/interrupt generated on any change of the comparator output (while CEVT = 0)
- 10 = Trigger/event/interrupt generated on transition of the comparator output:
  - If CPOL = 0 (non-inverted polarity): High-to-low transition only.
  - If CPOL = 1 (inverted polarity): Low-to-high transition only.
- 01 = Trigger/event/interrupt generated on transition of comparator output:
  - If CPOL = 0 (non-inverted polarity): Low-to-high transition only.
  - If CPOL = 1 (inverted polarity): High-to-low transition only.
- 00 = Trigger/event/interrupt generation is disabled

bit 5 **Unimplemented:** Read as ‘0’

REGISTER 22-1: CMxCON: COMPARATOR x CONTROL REGISTERS (COMPARATORS 1 THROUGH 3) (CONTINUED)

bit 4 **CREF:** Comparator Reference Select bits (non-inverting input)
1 = Non-inverting input connects to internal CVREF+ input reference voltage
0 = Non-inverting input connects to CxINA pin

bit 3-2 **Unimplemented:** Read as '0'

bit 1-0 **CCH<1:0>:** Comparator Channel Select bits
11 = Inverting input of comparator connects to CVREF- input reference voltage
10 = Inverting input of comparator connects to CxIND pin
01 = Inverting input of comparator connects to CxINC pin
00 = Inverting input of comparator connects to CxINB pin

REGISTER 22-2: CMSTAT: COMPARATOR MODULE STATUS REGISTER

R/W-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
CMIDL	—	—	—	—	C3EVT	C2EVT	C1EVT
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
—	—	—	—	—	C3OUT	C2OUT	C1OUT
bit 7							bit 0

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

bit 15 **CMIDL:** Comparator Stop in Idle Mode bit
1 = Discontinue operation of all comparators when device enters Idle mode
0 = Continue operation of all enabled comparators in Idle mode

bit 14-11 **Unimplemented:** Read as '0'

bit 10 **C3EVT:** Comparator 3 Event Status bit (read-only)
Shows the current event status of Comparator 3 (CM3CON<9>).

bit 9 **C2EVT:** Comparator 2 Event Status bit (read-only)
Shows the current event status of Comparator 2 (CM2CON<9>).

bit 8 **C1EVT:** Comparator 1 Event Status bit (read-only)
Shows the current event status of Comparator 1 (CM1CON<9>).

bit 7-3 **Unimplemented:** Read as '0'

bit 2 **C3OUT:** Comparator 3 Output Status bit (read-only)
Shows the current output of Comparator 3 (CM3CON<8>).

bit 1 **C2OUT:** Comparator 2 Output Status bit (read-only)
Shows the current output of Comparator 2 (CM2CON<8>).

bit 0 **C1OUT:** Comparator 1 Output Status bit (read-only)
Shows the current output of Comparator 1 (CM1CON<8>).

23.0 COMPARATOR VOLTAGE REFERENCE

> **Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 20. "Comparator Voltage Reference Module"** (DS39709).

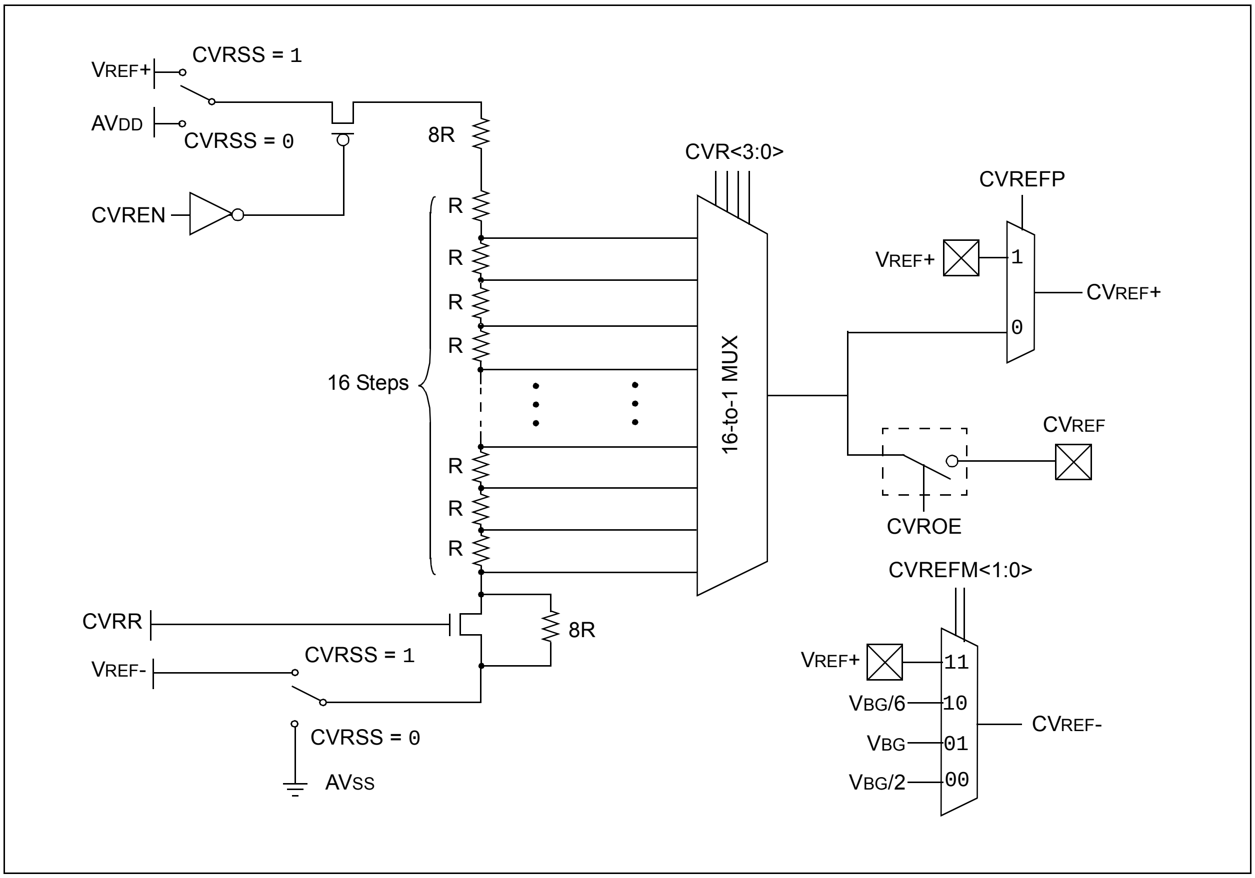
23.1 Configuring the Comparator Voltage Reference

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 (CVR<3:0>), with one range offering finer resolution.

The comparator reference supply voltage can come from either VDD and VSS, or the external VREF+ and VREF-. 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.

FIGURE 23-1: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM



REGISTER 23-1: CVRCON: COMPARATOR VOLTAGE REFERENCE CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	CVREFP	CVREFM1	CVREFM0
bit 15							bit 8

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	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
bit 7							bit 0

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

bit 15-11 **Unimplemented:** Read as ‘0’

bit 10 **CVREFP:** CVREF+ Reference Output Select bit
- 1 = Use VREF+ input pin as CVREF+ reference output to comparators
- 0 = Use comparator voltage reference module’s generated output as CVREF+ reference output to comparators

bit 9-8 **CVREFM<1:0>:** CVREF- Reference Output Select bits
- 11 = Use VREF+ input pin as CVREF- reference output to comparators
- 10 = Use VBG/6 as CVREF- reference output to comparators
- 01 = Use VBG as CVREF- reference output to comparators
- 00 = Use VBG/2 as CVREF- reference output to comparators

bit 7 **CVREN:** Comparator Voltage Reference Enable bit
- 1 = CVREF circuit is powered on
- 0 = CVREF circuit is powered down

bit 6 **CVROE:** Comparator VREF Output Enable bit
- 1 = CVREF voltage level is output on CVREF pin
- 0 = CVREF voltage level is disconnected from CVREF pin

bit 5 **CVRR:** Comparator VREF Range Selection bit
- 1 = CVRSRC range should be 0 to 0.625 CVRSRC with CVRSRC/24 step size
- 0 = CVRSRC range should be 0.25 to 0.719 CVRSRC with CVRSRC/32 step size

bit 4 **CVRSS:** Comparator VREF Source Selection bit
- 1 = Comparator reference source, CVRSRC = VREF+ – VREF-
- 0 = Comparator reference source, CVRSRC = AVDD – AVSS

bit 3-0 **CVR<3:0>:** Comparator VREF Value Selection (0 ≤ CVR<3:0> ≤ 15) bits

<u>When CVRR = 1:</u>
CVREF = (CVR<3:0>/24) • (CVRSRC)

<u>When CVRR = 0:</u>
CVREF = 1/4 • (CVRSRC) + (CVR<3:0>/32) • (CVRSRC)

24.0 CHARGE TIME MEASUREMENT UNIT (CTMU)

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the associated *"PIC24F Family Reference Manual"*, **Section 11. "Charge Time Measurement Unit (CTMU)"** (DS39724).

The Charge Time Measurement Unit is a flexible analog module that provides accurate differential time measurement between pulse sources, as well as asynchronous pulse generation. Its key features include:

- Four edge input trigger sources
- Polarity control for each edge source
- Control of edge sequence
- Control of response to edges
- Time measurement resolution of 1 nanosecond
- Accurate current source suitable for capacitive measurement

Together with other on-chip analog modules, the CTMU can be used to precisely measure time, measure capacitance, measure relative changes in capacitance or generate output pulses that are independent of the system clock. The CTMU module is ideal for interfacing with capacitive-based sensors.

The CTMU is controlled through two registers: CTMUCON and CTMUICON. CTMUCON enables the module and controls edge source selection, edge source polarity selection and edge sequencing. The CTMUICON register controls the selection and trim of the current source.

24.1 Measuring Capacitance

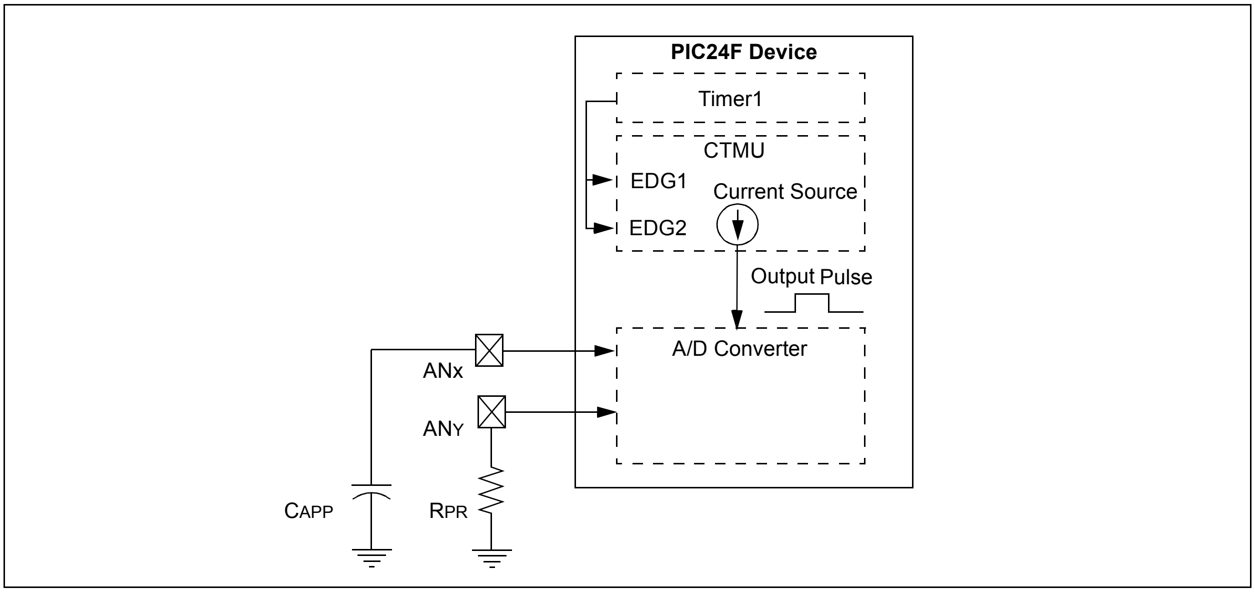
The CTMU module measures capacitance by generating an output pulse, with a width equal to the time between edge events, on two separate input channels. The pulse edge events to both input channels can be selected from four sources: two internal peripheral modules (OC1 and Timer1) and two external pins (CTEDG1 and CTEDG2). This pulse is used with the module's precision current source to calculate capacitance according to the relationship:

$$i = C \cdot \frac{dV}{dT}$$

For capacitance measurements, the A/D Converter samples an external capacitor (C_{APP}) on one of its input channels after the CTMU output's pulse. A Precision Resistor (R_{PR}) provides current source calibration on a second A/D channel. After the pulse ends, the converter determines the voltage on the capacitor. The actual calculation of capacitance is performed in software by the application.

Figure 24-1 shows the external connections used for capacitance measurements, and how the CTMU and A/D modules are related in this application. This example also shows the edge events coming from Timer1, but other configurations using external edge sources are possible. A detailed discussion on measuring capacitance and time with the CTMU module is provided in the *"PIC24F Family Reference Manual"*.

FIGURE 24-1: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR CAPACITANCE MEASUREMENT



24.2 Measuring Time

Time measurements on the pulse width can be similarly performed using the A/D module's internal capacitor (CAD) and a precision resistor for current calibration. Figure 24-2 shows the external connections used for time measurements, and how the CTMU and A/D modules are related in this application. This example also shows both edge events coming from the external CTEDG pins, but other configurations using internal edge sources are possible. For the smallest time measurements, select the internal A/D Channel 31, CH0Sx <4:0>= `11111`. This minimizes any stray capacitance that may otherwise be associated with using an input pin, thus keeping the total capacitance to that of the A/D Converter itself (4-5 pF). A detailed discussion on measuring capacitance and time with the CTMU module is provided in the "*PIC24F Family Reference Manual*".

24.3 Pulse Generation and Delay

The CTMU module can also generate an output pulse with edges that are not synchronous with the device's system clock. More specifically, it can generate a pulse with a programmable delay from an edge event input to the module.

When the module is configured for pulse generation delay by setting the TGEN bit (CTMUCON<12>), the internal current source is connected to the B input of Comparator 2. A capacitor (CDELAY) is connected to the Comparator 2 pin, C2INB, and the comparator voltage reference, CVREF, is connected to C2INA. CVREF is then configured for a specific trip point. The module begins to charge CDELAY when an edge event is detected. When CDELAY charges above the CVREF trip point, a pulse is output on CTPLS. The length of the pulse delay is determined by the value of CDELAY and the CVREF trip point.

Figure 24-3 shows the external connections for pulse generation, as well as the relationship of the different analog modules required. While CTEDG1 is shown as the input pulse source, other options are available. A detailed discussion on pulse generation with the CTMU module is provided in the "*PIC24F Family Reference Manual*".

FIGURE 24-2: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR TIME MEASUREMENT

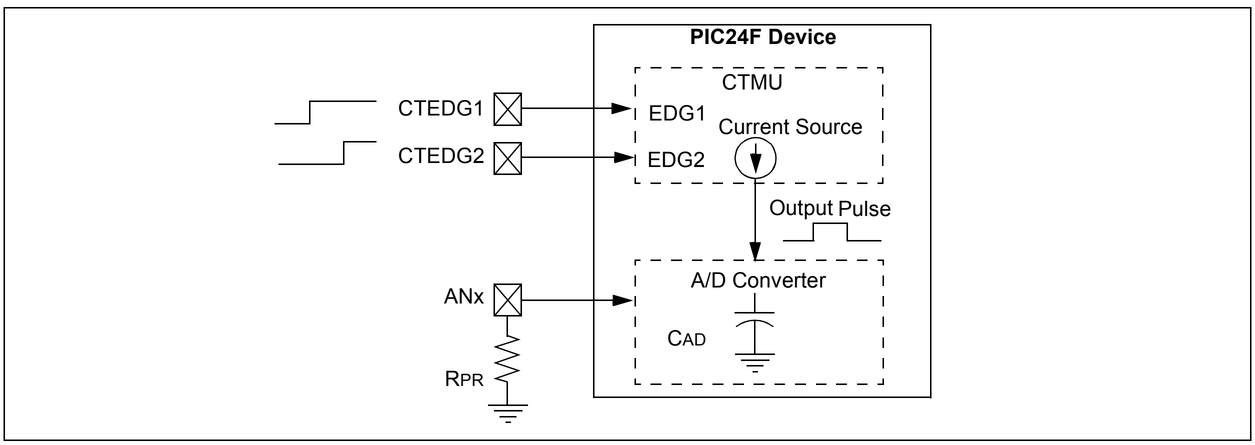
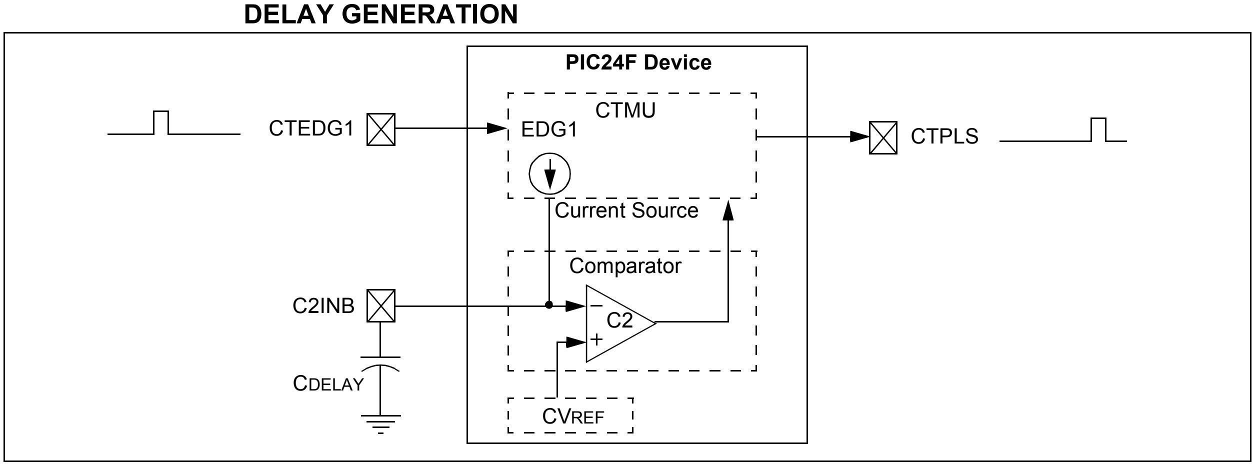


FIGURE 24-3: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR PULSE DELAY GENERATION



REGISTER 24-1: CTMUCON: CTMU CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEN	—	CTMUSIDL	TGEN[(1)]	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT
bit 7							bit 0

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

bit 15 **CTMUEN:** CTMU Enable bit
1 = Module is enabled
0 = Module is disabled

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **CTMUSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12 **TGEN:** Time Generation Enable bit[(1)]
1 = Enables edge delay generation
0 = Disables edge delay generation

bit 11 **EDGEN:** Edge Enable bit
1 = Edges are not blocked
0 = Edges are blocked

bit 10 **EDGSEQEN:** Edge Sequence Enable bit
1 = Edge 1 event must occur before Edge 2 event can occur
0 = No edge sequence is needed

bit 9 **IDISSEN:** Analog Current Source Control bit
1 = Analog current source output is grounded
0 = Analog current source output is not grounded

bit 8 **CTTRIG:** Trigger Control bit
1 = Trigger output is enabled
0 = Trigger output is disabled

bit 7 **EDG2POL:** Edge 2 Polarity Select bit
1 = Edge 2 is programmed for a positive edge response
0 = Edge 2 is programmed for a negative edge response

bit 6-5 **EDG2SEL<1:0>:** Edge 2 Source Select bits
11 = CTED1 pin
10 = CTED2 pin
01 = OC1 module
00 = Timer1 module

bit 4 **EDG1POL:** Edge 1 Polarity Select bit
1 = Edge 1 is programmed for a positive edge response
0 = Edge 1 is programmed for a negative edge response

Note 1: If TGEN = 1, the peripheral inputs and outputs must be configured to an available RPn pin. For more information, see **Section 10.4 “Peripheral Pin Select (PPS)”**.

REGISTER 24-1: CTMUCON: CTMU CONTROL REGISTER (CONTINUED)

bit 3-2 **EDG1SEL<1:0>:** Edge 1 Source Select bits
`11` = CTED1 pin
`10` = CTED2 pin
`01` = OC1 module
`00` = Timer1 module

bit 1 **EDG2STAT:** Edge 2 Status bit
`1` = Edge 2 event has occurred
`0` = Edge 2 event has not occurred

bit 0 **EDG1STAT:** Edge 1 Status bit
`1` = Edge 1 event has occurred
`0` = Edge 1 event has not occurred

Note 1: If TGEN = `1`, the peripheral inputs and outputs must be configured to an available RPn pin. For more information, see **Section 10.4 "Peripheral Pin Select (PPS)"**.

REGISTER 24-2: CTMUICON: CTMU CURRENT 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
ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 15-10 **ITRIM<5:0>:** Current Source Trim bits
`011111` = Maximum positive change from nominal current
`011110`
`.....`
`000001` = Minimum positive change from nominal current
`000000` = Nominal current output specified by IRNG<1:0>
`111111` = Minimum negative change from nominal current
`.....`
`100010`
`100001` = Maximum negative change from nominal current

bit 9-8 **IRNG<1:0>:** Current Source Range Select bits
`11` = 100 × Base Current
`10` = 10 × Base Current
`01` = Base current level (0.55 μA nominal)
`00` = Current source is disabled

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

25.0 SPECIAL FEATURES

> **Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the following sections of the *"PIC24F Family Reference Manual"*:
>
> - **Section 9. "Watchdog Timer (WDT)"** (DS39697)
> - **Section 32. "High-Level Device Integration"** (DS39719)
> - **Section 33. "Programming and Diagnostics"** (DS39716)

PIC24FJ64GA104 family devices include several features intended to maximize application flexibility and reliability, and minimize cost through elimination of external components. These are:

- Flexible Configuration
- Watchdog Timer (WDT)
- Code Protection
- JTAG Boundary Scan Interface
- In-Circuit Serial Programming
- In-Circuit Emulation

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 F80000h. A detailed explanation of the various bit functions is provided in Register 25-1 through Register 25-6.

Note that address F80000h is beyond the user program memory space. In fact, it belongs to the configuration memory space (800000h-FFFFFFh) which can only be accessed using table reads and table writes.

25.1.1 CONSIDERATIONS FOR CONFIGURING PIC24FJ64GA104 FAMILY DEVICES

In PIC24FJ64GA104 family devices, the configuration bytes are implemented as volatile memory. This means that configuration data must be programmed each time the device is powered up. Configuration data is stored in the three words at the top of the on-chip program memory space, known as the Flash Configuration Words. Their specific locations are shown in Table 25-1. These are packed representations of the actual device Configuration bits, whose actual locations are distributed among several locations in configuration space. The configuration data is automatically loaded from the Flash Configuration Words to the proper Configuration registers during device Resets.

> **Note:** Configuration data is reloaded on all types of device Resets.

When creating applications for these devices, users should always specifically allocate the location of the Flash Configuration Word for configuration data. This is to make certain that program code is not stored in this address when the code is compiled.

The upper byte of all Flash Configuration Words in program memory should always be '1111 1111'. This makes them appear to be NOP instructions in the remote event that their locations are ever executed by accident. Since Configuration bits are not implemented in the corresponding locations, writing '1's to these locations has no effect on device operation.

> **Note:** Performing a page erase operation on the last page of program memory clears the Flash Configuration Words, enabling code protection as a result. Therefore, users should avoid performing page erase operations on the last page of program memory.

TABLE 25-1: FLASH CONFIGURATION WORD LOCATIONS FOR PIC24FJ64GA104 FAMILY DEVICES

Device	Configuration Word Addresses			
	1	2	3	4
PIC24FJ32GA10x	57FEh	57FCh	57FAh	57F8h
PIC24FJ64GA10x	ABFEh	ABFCh	ABFAh	ABF8h

REGISTER 25-1: CW1: FLASH CONFIGURATION WORD 1

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

r-x	R/PO-1	R/PO-1	R/PO-1	R/PO-1	U-1	R/PO-1	R/PO-1
r	JTAGEN[1]	GCP	GWRP	$\overline{\text{DEBUG}}$	—	ICS1	ICS0
bit 15							bit 8

R/PO-1	R/PO-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
FWDTEN	WINDIS	—	FWPSA	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7							bit 0

Legend:	r = Reserved bit		
R = Readable bit	PO = Program Once bit	U = Unimplemented bit, read as '0'	
-n = Value when device is unprogrammed		'1' = Bit is set	'0' = Bit is cleared

bit 23-16 **Unimplemented:** Read as '1'

bit 15 **Reserved:** The value is unknown; program as '0'

bit 14 **JTAGEN:** JTAG Port Enable bit[1]
1 = JTAG port is enabled
0 = JTAG port is disabled

bit 13 **GCP:** General Segment Program Memory Code Protection bit
1 = Code protection is disabled
0 = Code protection is enabled for the entire program memory space

bit 12 **GWRP:** General Segment Code Flash Write Protection bit
1 = Writes to program memory are allowed
0 = Writes to program memory are disabled

bit 11 **$\overline{\text{DEBUG}}$:** Background Debugger Enable bit
1 = Device resets into Operational mode
0 = Device resets into Debug mode

bit 10 **Unimplemented:** Read as '1'

bit 9-8 **ICS<1:0>:** Emulator Pin Placement Select bits
11 = Emulator functions are shared with PGEC1/PGED1
10 = Emulator functions are shared with PGEC2/PGED2
01 = Emulator functions are shared with PGEC3/PGED3
00 = Reserved; do not use

bit 7 **FWDTEN:** Watchdog Timer Enable bit
1 = Watchdog Timer is enabled
0 = Watchdog Timer is disabled

bit 6 **WINDIS:** Windowed Watchdog Timer Disable bit
1 = Standard Watchdog Timer is enabled
0 = Windowed Watchdog Timer is enabled; FWDTEN must be '1'

bit 5 **Unimplemented:** Read as '1'

bit 4 **FWPSA:** WDT Prescaler Ratio Select bit
1 = Prescaler ratio of 1:128
0 = Prescaler ratio of 1:32

Note 1: The JTAGEN bit can only be modified using In-Circuit Serial Programming™ (ICSP™). It cannot be modified while connected through the JTAG interface.

REGISTER 25-1: CW1: FLASH CONFIGURATION WORD 1 (CONTINUED)

bit 3-0 **WDTPS<3:0>:** Watchdog Timer Postscaler 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

Note 1: The JTAGEN bit can only be modified using In-Circuit Serial Programming™ (ICSP™). It cannot be modified while connected through the JTAG interface.

REGISTER 25-2: CW2: FLASH CONFIGURATION WORD 2

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	U-1	U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1
IESO	—	—	—	—	FNOSC2	FNOSC1	FNOSC0
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	U-1	R/PO-1	R/PO-1	R/PO-1
FCKSM1	FCKSM0	OSCIOFCN	IOL1WAY	—	I2C1SEL	POSCMD1	POSCMD0
bit 7							bit 0

Legend:		
R = Readable bit	PO = Program Once bit	U = Unimplemented bit, read as ‘0’
-n = Value when device is unprogrammed	‘1’ = Bit is set	‘0’ = Bit is cleared

bit 23-16 **Unimplemented:** Read as ‘1’

bit 15 **IESO:** Internal External Switchover bit
1 = IESO mode (Two-Speed Start-up) is enabled
0 = IESO mode (Two-Speed Start-up) is disabled

bit 14-11 **Unimplemented:** Read as ‘1’

bit 10-8 **FNOSC<2:0>:** Initial Oscillator Select bits
111 = Fast RC Oscillator with Postscaler (FRCDIV)
110 = Reserved
101 = Low-Power RC Oscillator (LPRC)
100 = Secondary Oscillator (SOSC)
011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL)
010 = Primary Oscillator (XT, HS, EC)
001 = Fast RC Oscillator with Postscaler and PLL module (FRCPLL)
000 = Fast RC Oscillator (FRC)

bit 7-6 **FCKSM<1:0>:** Clock Switching and Fail-Safe Clock Monitor Configuration bits
1x = Clock switching and Fail-Safe Clock Monitor are disabled
01 = Clock switching is enabled, Fail-Safe Clock Monitor is disabled
00 = Clock switching is enabled, Fail-Safe Clock Monitor is enabled

bit 5 **OSCIOFCN:** OSCO Pin Configuration bit
If POSCMD<1:0> = 11 or 00:
1 = OSCO/CLKO/RA3 functions as CLKO (FOSC/2)
0 = OSCO/CLKO/RA3 functions as port I/O (RC15)
If POSCMD<1:0> = 10 or 01:
OSCIOFCN has no effect on OSCO/CLKO/RA3.

bit 4 **IOL1WAY:** IOLOCK One-Way Set Enable bit
1 = The IOLOCK bit (OSCCON<6>) can be set once, provided the unlock sequence has been completed. Once set, the Peripheral Pin Select registers cannot be written to a second time.
0 = The IOLOCK bit can be set and cleared as needed, provided the unlock sequence has been completed

bit 3 **Unimplemented:** Read as ‘1’

bit 2 **I2C1SEL:** I2C1 Pin Select bit
1 = Use default SCL1/SDA1 pins
0 = Use alternate SCL1/SDA1 pins

bit 1-0 **POSCMD<1:0>:** Primary Oscillator Configuration bits
11 = Primary Oscillator is disabled
10 = HS Oscillator mode is selected
01 = XT Oscillator mode is selected
00 = EC Oscillator mode is selected

REGISTER 25-3: CW3: FLASH CONFIGURATION WORD 3

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	R/PO-1	R/PO-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
WPEND	WPCFG	WPDIS	—	WUTSEL1	WUTSEL0	SOSCSEL1[1]	SOSCSEL0[1]
bit 15							bit 8

U-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
—	—	WPFP5	WPFP4	WPFP3	WPFP2	WPFP1	WPFP0
bit 7							bit 0

Legend:		
R = Readable bit	PO = Program Once bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed	'1' = Bit is set	'0' = Bit is cleared

bit 23-16 **Unimplemented:** Read as '1'

bit 15 **WPEND:** Segment Write Protection End Page Select bit
1 = Protected code segment lower boundary is at the bottom of program memory (000000h); upper boundary is the code page specified by WPFP<8:0>
0 = Protected code segment upper boundary is at the last page of program memory; lower boundary is the code page specified by WPFP<8:0>

bit 14 **WPCFG:** Configuration Word Code Page Protection Select bit
1 = Last page (at the top of program memory) and Flash Configuration Words are not protected
0 = Last page and Flash Configuration Words are code-protected

bit 13 **WPDIS:** Segment Write Protection Disable bit
1 = Segmented code protection is disabled
0 = Segmented code protection is enabled; protected segment defined by WPEND, WPCFG and WPFPx Configuration bits

bit 12 **Unimplemented:** Read as '1'

bit 11-10 **WUTSEL<1:0>:** Voltage Regulator Standby Mode Wake-up Time Select bits
11 = Default regulator start-up time used
01 = Fast regulator start-up time used
x0 = Reserved; do not use

bit 9-8 **SOSCSEL<1:0>:** Secondary Oscillator Power Mode Select bits[1]
11 = SOSC pins are in default (high drive strength) oscillator mode
01 = SOSC pins are in Low-Power (low drive strength) Oscillator mode
00 = SOSC pins have digital I/O functions (RA4, RB4); SCLKI can be used
10 = Reserved

bit 7-6 **Unimplemented:** Read as '1'

bit 5-0 **WPFP5:WPFP0:** Protected Code Segment Boundary Page bits
Designates the 512 instruction page that is the boundary of the protected code segment, starting with Page 9 at the bottom of program memory.
If WPEND = 1:
Last address of designated code page is the upper boundary of the segment.
If WPEND = 0:
First address of designated code page is the lower boundary of the segment.

Note 1: Digital functions on the SOSCI and SOSCO pins are only available when configured in Digital I/O mode ('00').

REGISTER 25-4: CW4: FLASH CONFIGURATION WORD 4

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DSWDTEN	DSBOREN	RTCOSC	DSWDTOSC	DSWDTPS3	DSWDTPS2	DSWDTPS1	DSWDTPS0
bit 7							bit 0

Legend:		
R = Readable bit	PO = Program Once bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed	'1' = Bit is set	'0' = Bit is cleared

bit 23-8 **Unimplemented:** Read as '1'

bit 7 **DSWDTEN:** Deep Sleep Watchdog Timer Enable bit
1 = DSWDT is enabled
0 = DSWDT is disabled

bit 6 **DSBOREN:** Deep Sleep BOR Enable bit
1 = BOR is enabled in Deep Sleep
0 = BOR is disabled in Deep Sleep (does not affect Sleep mode)

bit 5 **RTCOSC:** RTCC Reference Clock Select bit
1 = RTCC uses SOSC as reference clock
0 = RTCC uses LPRC as reference clock

bit 4 **DSWDTOSC:** DSWDT Reference Clock Select bit
1 = DSWDT uses LPRC as reference clock
0 = DSWDT uses SOSC as reference clock

bit 3-0 **DSWDTPS<3:0>:** DSWDT Postscale select bits
The DSWDT prescaler is 32; this creates an approximate base time unit of 1 ms.
1111 = 1:2,147,483,648 (25.7 days)
1110 = 1:536,870,912 (6.4 days)
1101 = 1:134,217,728 (38.5 hours)
1100 = 1:33,554,432 (9.6 hours)
1011 = 1:8,388,608 (2.4 hours)
1010 = 1:2,097,152 (36 minutes)
1001 = 1:524,288 (9 minutes)
1000 = 1:131,072 (135 seconds)
0111 = 1:32,768 (34 seconds)
0110 = 1:8,192 (8.5 seconds)
0101 = 1:2,048 (2.1 seconds)
0100 = 1:512 (528 ms)
0011 = 1:128 (132 ms)
0010 = 1:32 (33 ms)
0001 = 1:8 (8.3 ms)
0000 = 1:2 (2.1 ms)

REGISTER 25-5: DEVID: DEVICE ID REGISTER

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 23							bit 16

R	R	R	R	R	R	R	R
FAMID7	FAMID6	FAMID5	FAMID4	FAMID3	FAMID2	FAMID1	FAMID0
bit 15							bit 8

R	R	R	R	R	R	R	R
DEV7	DEV6	DEV5	DEV4	DEV3	DEV2	DEV1	DEV0
bit 7							bit 0

Legend: R = Read-Only bit	U = Unimplemented bit

bit 23-16 **Unimplemented:** Read as '1'

bit 15-8 **FAMID<7:0>:** Device Family Identifier bits
`01000010` = PIC24FJ64GA104 family

bit 7-0 **DEV<7:0>:** Individual Device Identifier bits
`00000010` = PIC24FJ32GA102
`00000110` = PIC24FJ64GA102
`00001010` = PIC24FJ32GA104
`00001110` = PIC24FJ64GA104

REGISTER 25-6: DEVREV: DEVICE REVISION REGISTER

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 23							bit 16

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 15							bit 8

U	U	U	U	R	R	R	R
—	—	—	—	REV3	REV2	REV1	REV0
bit 7							bit 0

Legend: R = Read-only bit	U = Unimplemented bit

bit 23-4 **Unimplemented:** Read as '0'

bit 3-0 **REV<3:0>:** Minor Revision Identifier bits
Encodes revision number of the device (sequential number only; no major/minor fields).

25.2 On-Chip Voltage Regulator

All PIC24FJ64GA104 family devices power their core digital logic at a nominal 2.5V. This may create an issue for designs that are required to operate at a higher typical voltage, such as 3.3V. To simplify system design, all devices in the PIC24FJ64GA104 family incorporate an on-chip regulator that allows the device to run its core logic from VDD.

The regulator is controlled by the DISVREG pin. Tying VSS to the pin enables the regulator, which in turn, provides power to the core from the other VDD pins. When the regulator is enabled, a low-ESR capacitor (such as ceramic) must be connected to the VDDCORE/VCAP pin (Figure 25-1). This helps to maintain the stability of the regulator. The recommended value for the Filter Capacitor (CEFC) is provided in **Section 28.1 "DC Characteristics"**.

If DISVREG is tied to VDD, the regulator is disabled. In this case, separate power for the core logic, at a nominal 2.5V, must be supplied to the device on the VDDCORE/VCAP pin to run the I/O pins at higher voltage levels, typically 3.3V. Alternatively, the VDDCORE/VCAP and VDD pins can be tied together to operate at a lower nominal voltage. Refer to Figure 25-1 for possible configurations.

25.2.1 VOLTAGE REGULATOR TRACKING MODE AND LOW-VOLTAGE DETECTION

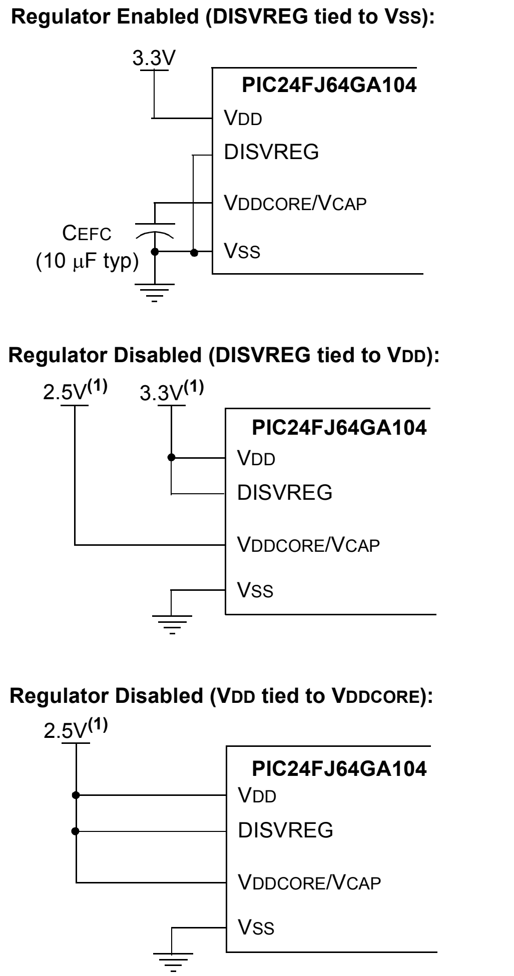
When it is enabled, the on-chip regulator provides a constant voltage of 2.5V nominal to the digital core logic.

The regulator can provide this level from a VDD of about 2.5V, all the way up to the device's VDDMAX. It does not have the capability to boost VDD levels below 2.5V. In order to prevent "brown-out" conditions when the voltage drops too low for the regulator, the regulator enters Tracking mode. In Tracking mode, the regulator output follows VDD with a typical voltage drop of 100 mV.

When the device enters Tracking mode, it is no longer possible to operate at full speed. To provide information about when the device enters Tracking mode, the on-chip regulator includes a simple, Low-Voltage Detect circuit. When VDD drops below full-speed operating voltage, the circuit sets the Low-Voltage Detect Interrupt Flag, LVDIF (IFS4<8>). This can be used to generate an interrupt and put the application into a Low-Power Operational mode or trigger an orderly shutdown.

Low-Voltage Detection is only available when the regulator is enabled.

FIGURE 25-1: CONNECTIONS FOR THE ON-CHIP REGULATOR



Note 1: These are typical operating voltages. Refer to **Section 28.1 "DC Characteristics"** for the full operating ranges of VDD and VDDCORE.

25.2.2 ON-CHIP REGULATOR AND POR

When the voltage regulator is enabled, it takes approximately 10 µs for it to generate output. During this time, designated as TPM, code execution is disabled. TPM is applied every time the device resumes operation after any power-down, including Sleep mode. TPM is determined by the setting of the PMSLP bit (RCON<8>) and the WUTSEL Configuration bits (CW3<11:10>).

Note: For more information on TPM, see **Section 28.0 "Electrical Characteristics"**.

If the regulator is disabled, a separate Power-up Timer (PWRT) is automatically enabled. The PWRT adds a fixed delay of 64 ms nominal delay at device start-up (POR or BOR only).

When waking up from Sleep with the regulator disabled, TPM is used to determine the wake-up time. To decrease the device wake-up time when operating with the regulator disabled, the PMSLP bit can be set.

25.2.3 ON-CHIP REGULATOR AND BOR

When the on-chip regulator is enabled, PIC24FJ64GA104 family devices also have a simple brown-out capability. If the voltage supplied to the regulator is inadequate to maintain the tracking level, the regulator Reset circuitry will generate a Brown-out Reset. This event is captured by the BOR flag bit (RCON<1>). The brown-out voltage specifications are provided in **Section 28.0 "Electrical Characteristics"**.

25.2.4 POWER-UP REQUIREMENTS

The on-chip regulator is designed to meet the power-up requirements for the device. If the application does not use the regulator, then strict power-up conditions must be adhered to. While powering up, VDDCORE must never exceed VDD by 0.3 volts.

Note:	For more information, see **Section 28.0 "Electrical Characteristics"**.

25.2.5 VOLTAGE REGULATOR STANDBY MODE

When enabled, the on-chip regulator always consumes a small incremental amount of current over IDD/IPD, including when the device is in Sleep mode, even though the core digital logic does not require power. To provide additional savings in applications where power resources are critical, the regulator automatically places itself into Standby mode whenever the device goes into Sleep mode by removing power from the Flash program memory. This feature is controlled by the PMSLP bit (RCON<8>). By default, this bit is cleared, which enables Standby mode.

For PIC24FJ64GA104 family devices, the time required for regulator wake-up from Standby mode is controlled by the WUTSEL<1:0> Configuration bits (CW3<11:10>). The default wake-up time for all devices is 190 µs, which is a Legacy mode provided to match older PIC24F device wake-up times.

Implementing the WUTSEL Configuration bits provides a fast wake-up option. When WUTSEL<1:0> = `01`, the regulator wake-up time is TPM, 10 µs.

When the regulator's Standby mode is turned off (PMSLP = `1`), Flash program memory stays powered in Sleep mode. That enables device wake-up without waiting for TPM. With PMSLP set, however, the power consumption, while in Sleep mode, will be approximately 40 µA higher than what it would be if the regulator was allowed to enter Standby mode.

25.3 Watchdog Timer (WDT)

For PIC24FJ64GA104 family devices, the WDT is driven by the LPRC Oscillator. When the WDT is enabled, the clock source is also enabled.

The nominal WDT clock source from LPRC is 31 kHz. This feeds a prescaler that can be configured for either 5-bit (divide-by-32) or 7-bit (divide-by-128) operation. The prescaler is set by the FWPSA Configuration bit. With a 31 kHz input, the prescaler yields a nominal WDT time-out period (TWDT) of 1 ms in 5-bit mode, or 4 ms in 7-bit mode.

A variable postscaler divides down the WDT prescaler output and allows for a wide range of time-out periods. The postscaler is controlled by the WDTPS<3:0> Configuration bits (CW1<3:0>), which allow the selection of a total of 16 settings, from 1:1 to 1:32,768. Using the prescaler and postscaler time-out periods, ranges from 1 ms to 131 seconds can be achieved.

The WDT, prescaler and postscaler are reset:

- On any device Reset
- On the completion of a clock switch, whether invoked by software (i.e., setting the OSWEN bit after changing the NOSC bits) or by hardware (i.e., Fail-Safe Clock Monitor)
- When a `PWRSAV` instruction is executed (i.e., Sleep or Idle mode is entered)
- When the device exits Sleep or Idle mode to resume normal operation
- By a `CLRWDT` instruction during normal execution

If the WDT is enabled, it will continue to run during Sleep or Idle modes. When the WDT time-out occurs, the device will wake the device and code execution will continue from where the `PWRSAV` instruction was executed. The corresponding SLEEP or IDLE bits (RCON<3:2>) will need to be cleared in software after the device wakes up.

The WDT Flag bit, WDTO (RCON<4>), is not automatically cleared following a WDT time-out. To detect subsequent WDT events, the flag must be cleared in software.

Note:	The `CLRWDT` and `PWRSAV` instructions clear the prescaler and postscaler counts when executed.

25.3.1 WINDOWED OPERATION

The Watchdog Timer has an optional Fixed Window mode of operation. In this Windowed mode, `CLRWDT` instructions can only reset the WDT during the last 1/4 of the programmed WDT period. A `CLRWDT` instruction is executed before that window causes a WDT Reset; this is similar to a WDT time-out.

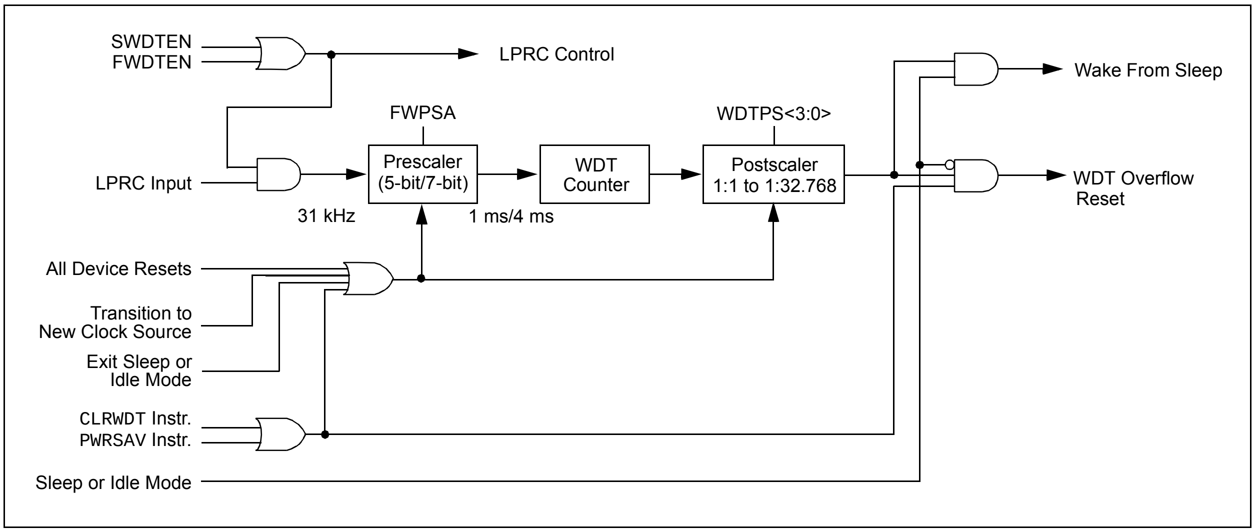
Windowed WDT mode is enabled by programming the WINDIS Configuration bit (CW1<6>) to '0'.

25.3.2 CONTROL REGISTER

The WDT is enabled or disabled by the FWDTEN Configuration bit. When the FWDTEN Configuration bit is set, the WDT is always enabled.

The WDT can be optionally controlled in software when the FWDTEN Configuration bit has been programmed to '0'. The WDT is enabled in software by setting the SWDTEN control bit (RCON<5>). The SWDTEN control bit is cleared on any device Reset. The WDT software option allows the user to enable the WDT for critical code segments, and disable the WDT during non-critical segments, for maximum power savings.

FIGURE 25-2: WDT BLOCK DIAGRAM



25.4 Deep Sleep Watchdog Timer (DSWDT)

PIC24FJ64GA104 family devices have both a WDT module and a DSWDT module. The latter runs, if enabled, when a device is in Deep Sleep and is driven by either the SOSC or LPRC Oscillator. The clock source is selected by the DSWDTOSC (CW4<4>) Configuration bit.

The DSWDT can be configured to generate a time-out at 2.1 ms to 25.7 days by selecting the respective postscaler.The postscaler can be selected by the Configuration bits, DSWDTPS<3:0> (CW4<3:0>). When the DSWDT is enabled, the clock source is also enabled. DSWDT is one of the sources that can wake the device from Deep Sleep mode.

25.5 Program Verification and Code Protection

PIC24FJ64GA104 family devices provide two complimentary methods to protect application code from overwrites and erasures. These also help to protect the device from inadvertent configuration changes during run time.

25.5.1 GENERAL SEGMENT PROTECTION

For all devices in the PIC24FJ64GA104 family, the on-chip program memory space is treated as a single block, known as the General Segment (GS). Code protection for this block is controlled by one Configuration bit, GCP. This bit inhibits external reads and writes to the program memory space. It has no direct effect in normal execution mode.

Write protection is controlled by the GWRP bit in the Configuration Word. When GWRP is programmed to '0', internal write and erase operations to program memory are blocked.

25.5.2 CODE SEGMENT PROTECTION

In addition to global General Segment protection, a separate subrange of the program memory space can be individually protected against writes and erases. This area can be used for many purposes where a separate block of erase and write-protected code is needed, such as bootloader applications. Unlike common boot block implementations, the specially protected segment in the PIC24FJ64GA104 family devices can be located by the user anywhere in the program space and configured in a wide range of sizes.

Code segment protection provides an added level of protection to a designated area of program memory, by disabling the NVM safety interlock, whenever a write or erase address falls within a specified range. It does not override General Segment protection controlled by the GCP or GWRP bits. For example, if GCP and GWRP are enabled, enabling segmented code protection for the bottom half of program memory does not undo General Segment protection for the top half.

The size and type of protection for the segmented code range are configured by the WPFPx, WPEND, WPCFG and WPDIS bits in Configuration Word 3. Code segment protection is enabled by programming the WPDIS bit (= 0). The WPFP bits specify the size of the segment to be protected by specifying the 512-word code page that is the start or end of the protected segment. The specified region is inclusive, therefore, this page will also be protected.

The WPEND bit determines if the protected segment uses the top or bottom of the program space as a boundary. Programming WPEND (= 0) sets the bottom of program memory (000000h) as the lower boundary of the protected segment. Leaving WPEND unprogrammed (= 1) protects the specified page through the last page of implemented program memory, including the Configuration Word locations.

A separate bit, WPCFG, is used to independently protect the last page of program space, including the Flash Configuration Words. Programming WPCFG (= 0) protects the last page, regardless of the other bit settings. This may be useful in circumstances where write protection is needed for both a code segment in the bottom of memory, as well as the Flash Configuration Words.

The various options for segment code protection are shown in Table 25-2.

25.5.3 CONFIGURATION REGISTER PROTECTION

The Configuration registers are protected against inadvertent or unwanted changes, or reads in two ways. The primary protection method is the same as that of the RP registers – shadow registers contain a complimentary value which is constantly compared with the actual value.

To safeguard against unpredictable events, Configuration bit changes resulting from individual cell level disruptions (such as ESD events) will cause a parity error and trigger a device Reset.

The data for the Configuration registers is derived from the Flash Configuration Words in program memory. When the GCP bit is set, the source data for device configuration is also protected as a consequence. Even if General Segment protection is not enabled, the device configuration can be protected by using the appropriate code cement protection setting.

TABLE 25-2: SEGMENT CODE PROTECTION CONFIGURATION OPTIONS

Segment Configuration Bits			Write/Erase Protection of Code Segment
WPDIS	WPEND	WPCFG	
1	x	1	No additional protection enabled; all program memory protection is configured by GCP and GWRP
1	x	0	Last code page protected, including Flash Configuration Words
0	1	0	Addresses from the first address of code page are defined by WPFP<5:0> through the end of implemented program memory (inclusive) are protected, including Flash Configuration Words
0	0	0	Address, 000000h, through the last address of code page, defined by WPFP<5:0> (inclusive) is protected
0	1	1	Addresses from first address of code page, defined by WPFP<5:0> through the end of implemented program memory (inclusive), are protected, including Flash Configuration Words
0	0	1	Addresses from first address of code page, defined by WPFP<5:0> through the end of implemented program memory (inclusive), are protected

25.6 JTAG Interface

PIC24FJ64GA104 family devices implement a JTAG interface, which supports boundary scan device testing.

25.7 In-Circuit Serial Programming

PIC24FJ64GA104 family microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock (PGECx) and data (PGEDx), 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 MPLAB® ICD 2 is selected as a debugger, the in-circuit debugging functionality is enabled. This function allows simple debugging functions when used with MPLAB IDE. Debugging functionality is controlled through the PGECx (Emulation/Debug Clock) and PGEDx (Emulation/Debug Data) pins.

To use the in-circuit debugger function of the device, the design must implement ICSP connections to $\overline{\text{MCLR}}$, VDD, VSS and the PGECx/PGEDx pin pair designated by the ICS Configuration bits. In addition, when the feature is enabled, some of the resources are not available for general use. These resources include the first 80 bytes of data RAM and two I/O pins.

26.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers and dsPIC® digital signal controllers are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB C Compiler for Various Device Families
  - HI-TECH C for Various Device Families
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
  - MPLAB ICD 3
  - PICkit™ 3 Debug Express
- Device Programmers
  - PICkit™ 2 Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

26.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - In-Circuit 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
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (C or assembly)
  - Mixed C and assembly
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

26.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

26.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

The compilers include a macro assembler, linker, preprocessor, and one-step driver, and can run on multiple platforms.

26.4 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

26.5 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

26.6 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC devices. MPLAB C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

26.7 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

26.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC® Flash MCUs and dsPIC® Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

26.9 MPLAB ICD 3 In-Circuit Debugger System

MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost effective high-speed hardware debugger/programmer for Microchip Flash Digital Signal Controller (DSC) and microcontroller (MCU) devices. It debugs and programs PIC® Flash microcontrollers and dsPIC® DSCs with the powerful, yet easy-to-use graphical user interface of MPLAB Integrated Development Environment (IDE).

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

26.10 PICkit 3 In-Circuit Debugger/ Programmer and PICkit 3 Debug Express

The MPLAB PICkit 3 allows debugging and programming of PIC® and dsPIC® Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB Integrated Development Environment (IDE). The MPLAB PICkit 3 is connected to the design engineer's PC using a full speed USB interface and can be connected to the target via an Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the reset line to implement in-circuit debugging and In-Circuit Serial Programming™.

The PICkit 3 Debug Express include the PICkit 3, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

26.11 PICkit 2 Development Programmer/Debugger and PICkit 2 Debug Express

The PICkit™ 2 Development Programmer/Debugger is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcontrollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

The PICkit 2 Debug Express include the PICkit 2, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

26.12 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an MMC card for file storage and data applications.

26.13 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

27.0 INSTRUCTION SET SUMMARY

> **Note:** This chapter is a brief summary of the PIC24F instruction set architecture, and is not intended to be a comprehensive reference source.

The PIC24F instruction set adds many enhancements to the previous PIC® MCU instruction sets, while maintaining an easy migration from previous PIC MCU instruction sets. Most instructions are a single program memory word. Only three instructions require two program memory locations.

Each single-word instruction is a 24-bit word divided into an 8-bit 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:

- Word or byte-oriented operations
- Bit-oriented operations
- Literal operations
- Control operations

Table 27-1 shows the general symbols used in describing the instructions. The PIC24F instruction set summary in Table 27-2 lists all of the instructions, along with the status flags affected by each instruction.

Most word or byte-oriented W register instructions (including barrel shift instructions) have three operands:

- The first source operand, which is typically a register 'Wb' without any address modifier
- The second source operand, which is typically a register 'Ws' with or without an address modifier
- The destination of the result, which is typically a register 'Wd' with or without an address modifier

However, word or byte-oriented file register instructions have two operands:

- The file register specified by the value, 'f'
- The destination, which could either be the file register, 'f', or the W0 register, which is denoted as 'WREG'

Most bit-oriented instructions (including simple rotate/shift instructions) have two operands:

- The W register (with or without an address modifier) or file register (specified by the value of 'Ws' or 'f')
- The bit in the W register or file register (specified by a literal value or indirectly by the contents of register, 'Wb')

The literal instructions that involve data movement may use some of the following operands:

- A literal value to be loaded into a W register or file register (specified by the value of 'k')
- The W register or file register where the literal value is to be loaded (specified by 'Wb' or 'f')

However, literal instructions that involve arithmetic or logical operations use some of the following operands:

- The first source operand, which is a register 'Wb' without any address modifier
- The second source operand, which is a literal value
- The destination of the result (only if not the same as the first source operand), which is typically a register 'Wd' with or without an address modifier

The control instructions may use some of the following operands:

- A program memory address
- The mode of the table read and table write instructions

All instructions are a single word, except for certain double-word instructions, which were made double-word instructions so that all the required information is available in these 48 bits. In the second word, the 8 MSbs are '0's. If this second word is executed as an instruction (by itself), it will execute as a NOP.

Most 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. Notable exceptions are the BRA (unconditional/computed branch), indirect CALL/GOTO, all table reads and writes, and RETURN/RETFIE instructions, which are single-word instructions but take two or three cycles.

Certain instructions that involve skipping over the subsequent instruction require either two or three cycles if the skip is performed, depending on whether the instruction being skipped is a single-word or two-word instruction. Moreover, double-word moves require two cycles. The double-word instructions execute in two instruction cycles.

TABLE 27-1: SYMBOLS USED IN OPCODE DESCRIPTIONS

Field	Description
#text	Means literal defined by "text"
(text)	Means "content of text"
[text]	Means "the location addressed by text"
{ }	Optional field or operation
<n:m>	Register bit field
.b	Byte mode selection
.d	Double-Word mode selection
.S	Shadow register select
.w	Word mode selection (default)
bit4	4-bit bit selection field (used in word addressed instructions) ∈ {0...15}
C, DC, N, OV, Z	MCU Status bits: Carry, Digit Carry, Negative, Overflow, Sticky Zero
Expr	Absolute address, label or expression (resolved by the linker)
f	File register address ∈ {0000h...1FFFh}
lit1	1-bit unsigned literal ∈ {0,1}
lit4	4-bit unsigned literal ∈ {0...15}
lit5	5-bit unsigned literal ∈ {0...31}
lit8	8-bit unsigned literal ∈ {0...255}
lit10	10-bit unsigned literal ∈ {0...255} for Byte mode, {0:1023} for Word mode
lit14	14-bit unsigned literal ∈ {0...16383}
lit16	16-bit unsigned literal ∈ {0...65535}
lit23	23-bit unsigned literal ∈ {0...8388607}; LSB must be '0'
None	Field does not require an entry, may be blank
PC	Program Counter
Slit10	10-bit signed literal ∈ {-512...511}
Slit16	16-bit signed literal ∈ {-32768...32767}
Slit6	6-bit signed literal ∈ {-16...16}
Wb	Base W register ∈ {W0..W15}
Wd	Destination W register ∈ { Wd, [Wd], [Wd++], [Wd--], [++Wd], [--Wd] }
Wdo	Destination W register ∈ { Wnd, [Wnd], [Wnd++], [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb] }
Wm,Wn	Dividend, Divisor working register pair (direct addressing)
Wn	One of 16 working registers ∈ {W0..W15}
Wnd	One of 16 destination working registers ∈ {W0..W15}
Wns	One of 16 source working registers ∈ {W0..W15}
WREG	W0 (working register used in file register instructions)
Ws	Source W register ∈ { Ws, [Ws], [Ws++], [Ws--], [++Ws], [--Ws] }
Wso	Source W register ∈ { Wns, [Wns], [Wns++], [Wns--], [++Wns], [--Wns], [Wns+Wb] }

TABLE 27-2: INSTRUCTION SET OVERVIEW

Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
ADD	ADD	f	f = f + WREG	1	1	C, DC, N, OV, Z
	ADD	f,WREG	WREG = f + WREG	1	1	C, DC, N, OV, Z
	ADD	#lit10,Wn	Wd = lit10 + Wd	1	1	C, DC, N, OV, Z
	ADD	Wb,Ws,Wd	Wd = Wb + Ws	1	1	C, DC, N, OV, Z
	ADD	Wb,#lit5,Wd	Wd = Wb + lit5	1	1	C, DC, N, OV, Z
ADDC	ADDC	f	f = f + WREG + (C)	1	1	C, DC, N, OV, Z
	ADDC	f,WREG	WREG = f + WREG + (C)	1	1	C, DC, N, OV, Z
	ADDC	#lit10,Wn	Wd = lit10 + Wd + (C)	1	1	C, DC, N, OV, Z
	ADDC	Wb,Ws,Wd	Wd = Wb + Ws + (C)	1	1	C, DC, N, OV, Z
	ADDC	Wb,#lit5,Wd	Wd = Wb + lit5 + (C)	1	1	C, DC, N, OV, Z
AND	AND	f	f = f .AND. WREG	1	1	N, Z
	AND	f,WREG	WREG = f .AND. WREG	1	1	N, Z
	AND	#lit10,Wn	Wd = lit10 .AND. Wd	1	1	N, Z
	AND	Wb,Ws,Wd	Wd = Wb .AND. Ws	1	1	N, Z
	AND	Wb,#lit5,Wd	Wd = Wb .AND. lit5	1	1	N, Z
ASR	ASR	f	f = Arithmetic Right Shift f	1	1	C, N, OV, Z
	ASR	f,WREG	WREG = Arithmetic Right Shift f	1	1	C, N, OV, Z
	ASR	Ws,Wd	Wd = Arithmetic Right Shift Ws	1	1	C, N, OV, Z
	ASR	Wb,Wns,Wnd	Wnd = Arithmetic Right Shift Wb by Wns	1	1	N, Z
	ASR	Wb,#lit5,Wnd	Wnd = Arithmetic Right Shift Wb by lit5	1	1	N, Z
BCLR	BCLR	f,#bit4	Bit Clear f	1	1	None
	BCLR	Ws,#bit4	Bit Clear Ws	1	1	None
BRA	BRA	C,Expr	Branch if Carry	1	1 (2)	None
	BRA	GE,Expr	Branch if Greater than or Equal	1	1 (2)	None
	BRA	GEU,Expr	Branch if Unsigned Greater than or Equal	1	1 (2)	None
	BRA	GT,Expr	Branch if Greater than	1	1 (2)	None
	BRA	GTU,Expr	Branch if Unsigned Greater than	1	1 (2)	None
	BRA	LE,Expr	Branch if Less than or Equal	1	1 (2)	None
	BRA	LEU,Expr	Branch if Unsigned Less than or Equal	1	1 (2)	None
	BRA	LT,Expr	Branch if Less than	1	1 (2)	None
	BRA	LTU,Expr	Branch if Unsigned Less than	1	1 (2)	None
	BRA	N,Expr	Branch if Negative	1	1 (2)	None
	BRA	NC,Expr	Branch if Not Carry	1	1 (2)	None
	BRA	NN,Expr	Branch if Not Negative	1	1 (2)	None
	BRA	NOV,Expr	Branch if Not Overflow	1	1 (2)	None
	BRA	NZ,Expr	Branch if Not Zero	1	1 (2)	None
	BRA	OV,Expr	Branch if Overflow	1	1 (2)	None
	BRA	Expr	Branch Unconditionally	1	2	None
	BRA	Z,Expr	Branch if Zero	1	1 (2)	None
	BRA	Wn	Computed Branch	1	2	None
BSET	BSET	f,#bit4	Bit Set f	1	1	None
	BSET	Ws,#bit4	Bit Set Ws	1	1	None
BSW	BSW.C	Ws,Wb	Write C bit to Ws<Wb>	1	1	None
	BSW.Z	Ws,Wb	Write Z bit to Ws<Wb>	1	1	None
BTG	BTG	f,#bit4	Bit Toggle f	1	1	None
	BTG	Ws,#bit4	Bit Toggle Ws	1	1	None
BTSC	BTSC	f,#bit4	Bit Test f, Skip if Clear	1	1 (2 or 3)	None
	BTSC	Ws,#bit4	Bit Test Ws, Skip if Clear	1	1 (2 or 3)	None

TABLE 27-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
BTSS	BTSS	f,#bit4	Bit Test f, Skip if Set	1	1 (2 or 3)	None
	BTSS	Ws,#bit4	Bit Test Ws, Skip if Set	1	1 (2 or 3)	None
BTST	BTST	f,#bit4	Bit Test f	1	1	Z
	BTST.C	Ws,#bit4	Bit Test Ws to C	1	1	C
	BTST.Z	Ws,#bit4	Bit Test Ws to Z	1	1	Z
	BTST.C	Ws,Wb	Bit Test Ws<Wb> to C	1	1	C
	BTST.Z	Ws,Wb	Bit Test Ws<Wb> to Z	1	1	Z
BTSTS	BTSTS	f,#bit4	Bit Test then Set f	1	1	Z
	BTSTS.C	Ws,#bit4	Bit Test Ws to C, then Set	1	1	C
	BTSTS.Z	Ws,#bit4	Bit Test Ws to Z, then Set	1	1	Z
CALL	CALL	lit23	Call Subroutine	2	2	None
	CALL	Wn	Call Indirect Subroutine	1	2	None
CLR	CLR	f	f = 0x0000	1	1	None
	CLR	WREG	WREG = 0x0000	1	1	None
	CLR	Ws	Ws = 0x0000	1	1	None
CLRWDT	CLRWDT		Clear Watchdog Timer	1	1	WDTO, Sleep
COM	COM	f	f = $\bar{f}$	1	1	N, Z
	COM	f,WREG	WREG = $\bar{f}$	1	1	N, Z
	COM	Ws,Wd	Wd = $\overline{Ws}$	1	1	N, Z
CP	CP	f	Compare f with WREG	1	1	C, DC, N, OV, Z
	CP	Wb,#lit5	Compare Wb with lit5	1	1	C, DC, N, OV, Z
	CP	Wb,Ws	Compare Wb with Ws (Wb – Ws)	1	1	C, DC, N, OV, Z
CP0	CP0	f	Compare f with 0x0000	1	1	C, DC, N, OV, Z
	CP0	Ws	Compare Ws with 0x0000	1	1	C, DC, N, OV, Z
CPB	CPB	f	Compare f with WREG, with Borrow	1	1	C, DC, N, OV, Z
	CPB	Wb,#lit5	Compare Wb with lit5, with Borrow	1	1	C, DC, N, OV, Z
	CPB	Wb,Ws	Compare Wb with Ws, with Borrow (Wb – Ws – $\overline{C}$)	1	1	C, DC, N, OV, Z
CPSEQ	CPSEQ	Wb,Wn	Compare Wb with Wn, Skip if =	1	1 (2 or 3)	None
CPSGT	CPSGT	Wb,Wn	Compare Wb with Wn, Skip if >	1	1 (2 or 3)	None
CPSLT	CPSLT	Wb,Wn	Compare Wb with Wn, Skip if <	1	1 (2 or 3)	None
CPSNE	CPSNE	Wb,Wn	Compare Wb with Wn, Skip if ≠	1	1 (2 or 3)	None
DAW	DAW.B	Wn	Wn = Decimal Adjust Wn	1	1	C
DEC	DEC	f	f = f – 1	1	1	C, DC, N, OV, Z
	DEC	f,WREG	WREG = f – 1	1	1	C, DC, N, OV, Z
	DEC	Ws,Wd	Wd = Ws – 1	1	1	C, DC, N, OV, Z
DEC2	DEC2	f	f = f – 2	1	1	C, DC, N, OV, Z
	DEC2	f,WREG	WREG = f – 2	1	1	C, DC, N, OV, Z
	DEC2	Ws,Wd	Wd = Ws – 2	1	1	C, DC, N, OV, Z
DISI	DISI	#lit14	Disable Interrupts for k Instruction Cycles	1	1	None
DIV	DIV.SW	Wm,Wn	Signed 16/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.SD	Wm,Wn	Signed 32/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.UW	Wm,Wn	Unsigned 16/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.UD	Wm,Wn	Unsigned 32/16-bit Integer Divide	1	18	N, Z, C, OV
EXCH	EXCH	Wns,Wnd	Swap Wns with Wnd	1	1	None
FF1L	FF1L	Ws,Wnd	Find First One from Left (MSb) Side	1	1	C
FF1R	FF1R	Ws,Wnd	Find First One from Right (LSb) Side	1	1	C

TABLE 27-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
GOTO	GOTO	Expr	Go to Address	2	2	None
	GOTO	Wn	Go to Indirect	1	2	None
INC	INC	f	f = f + 1	1	1	C, DC, N, OV, Z
	INC	f,WREG	WREG = f + 1	1	1	C, DC, N, OV, Z
	INC	Ws,Wd	Wd = Ws + 1	1	1	C, DC, N, OV, Z
INC2	INC2	f	f = f + 2	1	1	C, DC, N, OV, Z
	INC2	f,WREG	WREG = f + 2	1	1	C, DC, N, OV, Z
	INC2	Ws,Wd	Wd = Ws + 2	1	1	C, DC, N, OV, Z
IOR	IOR	f	f = f .IOR. WREG	1	1	N, Z
	IOR	f,WREG	WREG = f .IOR. WREG	1	1	N, Z
	IOR	#lit10,Wn	Wd = lit10 .IOR. Wd	1	1	N, Z
	IOR	Wb,Ws,Wd	Wd = Wb .IOR. Ws	1	1	N, Z
	IOR	Wb,#lit5,Wd	Wd = Wb .IOR. lit5	1	1	N, Z
LNK	LNK	#lit14	Link Frame Pointer	1	1	None
LSR	LSR	f	f = Logical Right Shift f	1	1	C, N, OV, Z
	LSR	f,WREG	WREG = Logical Right Shift f	1	1	C, N, OV, Z
	LSR	Ws,Wd	Wd = Logical Right Shift Ws	1	1	C, N, OV, Z
	LSR	Wb,Wns,Wnd	Wnd = Logical Right Shift Wb by Wns	1	1	N, Z
	LSR	Wb,#lit5,Wnd	Wnd = Logical Right Shift Wb by lit5	1	1	N, Z
MOV	MOV	f,Wn	Move f to Wn	1	1	None
	MOV	[Wns+Slit10],Wnd	Move [Wns + Slit10] to Wnd	1	1	None
	MOV	f	Move f to f	1	1	N, Z
	MOV	f,WREG	Move f to WREG	1	1	N, Z
	MOV	#lit16,Wn	Move 16-bit Literal to Wn	1	1	None
	MOV.b	#lit8,Wn	Move 8-bit Literal to Wn	1	1	None
	MOV	Wn,f	Move Wn to f	1	1	None
	MOV	Wns,[Wns+Slit10]	Move Wns to [Wns + Slit10]	1	1	
	MOV	Wso,Wdo	Move Ws to Wd	1	1	None
	MOV	WREG,f	Move WREG to f	1	1	N, Z
	MOV.D	Wns,Wd	Move Double from W(ns):W(ns + 1) to Wd	1	2	None
	MOV.D	Ws,Wnd	Move Double from Ws to W(nd + 1):W(nd)	1	2	None
MUL	MUL.SS	Wb,Ws,Wnd	{Wnd + 1, Wnd} = Signed(Wb) * Signed(Ws)	1	1	None
	MUL.SU	Wb,Ws,Wnd	{Wnd + 1, Wnd} = Signed(Wb) * Unsigned(Ws)	1	1	None
	MUL.US	Wb,Ws,Wnd	{Wnd + 1, Wnd} = Unsigned(Wb) * Signed(Ws)	1	1	None
	MUL.UU	Wb,Ws,Wnd	{Wnd + 1, Wnd} = Unsigned(Wb) * Unsigned(Ws)	1	1	None
	MUL.SU	Wb,#lit5,Wnd	{Wnd + 1, Wnd} = Signed(Wb) * Unsigned(lit5)	1	1	None
	MUL.UU	Wb,#lit5,Wnd	{Wnd + 1, Wnd} = Unsigned(Wb) * Unsigned(lit5)	1	1	None
	MUL	f	W3:W2 = f * WREG	1	1	None
NEG	NEG	f	f = $\overline{f}$ + 1	1	1	C, DC, N, OV, Z
	NEG	f,WREG	WREG = $\overline{f}$ + 1	1	1	C, DC, N, OV, Z
	NEG	Ws,Wd	Wd = $\overline{Ws}$ + 1	1	1	C, DC, N, OV, Z
NOP	NOP		No Operation	1	1	None
	NOPR		No Operation	1	1	None
POP	POP	f	Pop f from Top-of-Stack (TOS)	1	1	None
	POP	Wdo	Pop from Top-of-Stack (TOS) to Wdo	1	1	None
	POP.D	Wnd	Pop from Top-of-Stack (TOS) to W(nd):W(nd + 1)	1	2	None
	POP.S		Pop Shadow Registers	1	1	All
PUSH	PUSH	f	Push f to Top-of-Stack (TOS)	1	1	None
	PUSH	Wso	Push Wso to Top-of-Stack (TOS)	1	1	None
	PUSH.D	Wns	Push W(ns):W(ns + 1) to Top-of-Stack (TOS)	1	2	None
	PUSH.S		Push Shadow Registers	1	1	None

TABLE 27-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
PWRSAV	PWRSAV #lit1	Go into Sleep or Idle mode	1	1	WDTO, Sleep
RCALL	RCALL Expr	Relative Call	1	2	None
	RCALL Wn	Computed Call	1	2	None
REPEAT	REPEAT #lit14	Repeat Next Instruction lit14 + 1 times	1	1	None
	REPEAT Wn	Repeat Next Instruction (Wn) + 1 times	1	1	None
RESET	RESET	Software Device Reset	1	1	None
RETFIE	RETFIE	Return from Interrupt	1	3 (2)	None
RETLW	RETLW #lit10,Wn	Return with Literal in Wn	1	3 (2)	None
RETURN	RETURN	Return from Subroutine	1	3 (2)	None
RLC	RLC f	f = Rotate Left through Carry f	1	1	C, N, Z
	RLC f,WREG	WREG = Rotate Left through Carry f	1	1	C, N, Z
	RLC Ws,Wd	Wd = Rotate Left through Carry Ws	1	1	C, N, Z
RLNC	RLNC f	f = Rotate Left (No Carry) f	1	1	N, Z
	RLNC f,WREG	WREG = Rotate Left (No Carry) f	1	1	N, Z
	RLNC Ws,Wd	Wd = Rotate Left (No Carry) Ws	1	1	N, Z
RRC	RRC f	f = Rotate Right through Carry f	1	1	C, N, Z
	RRC f,WREG	WREG = Rotate Right through Carry f	1	1	C, N, Z
	RRC Ws,Wd	Wd = Rotate Right through Carry Ws	1	1	C, N, Z
RRNC	RRNC f	f = Rotate Right (No Carry) f	1	1	N, Z
	RRNC f,WREG	WREG = Rotate Right (No Carry) f	1	1	N, Z
	RRNC Ws,Wd	Wd = Rotate Right (No Carry) Ws	1	1	N, Z
SE	SE Ws,Wnd	Wnd = Sign-Extended Ws	1	1	C, N, Z
SETM	SETM f	f = FFFFh	1	1	None
	SETM WREG	WREG = FFFFh	1	1	None
	SETM Ws	Ws = FFFFh	1	1	None
SL	SL f	f = Left Shift f	1	1	C, N, OV, Z
	SL f,WREG	WREG = Left Shift f	1	1	C, N, OV, Z
	SL Ws,Wd	Wd = Left Shift Ws	1	1	C, N, OV, Z
	SL Wb,Wns,Wnd	Wnd = Left Shift Wb by Wns	1	1	N, Z
	SL Wb,#lit5,Wnd	Wnd = Left Shift Wb by lit5	1	1	N, Z
SUB	SUB f	f = f – WREG	1	1	C, DC, N, OV, Z
	SUB f,WREG	WREG = f – WREG	1	1	C, DC, N, OV, Z
	SUB #lit10,Wn	Wn = Wn – lit10	1	1	C, DC, N, OV, Z
	SUB Wb,Ws,Wd	Wd = Wb – Ws	1	1	C, DC, N, OV, Z
	SUB Wb,#lit5,Wd	Wd = Wb – lit5	1	1	C, DC, N, OV, Z
SUBB	SUBB f	f = f – WREG – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB f,WREG	WREG = f – WREG – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB #lit10,Wn	Wn = Wn – lit10 – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB Wb,Ws,Wd	Wd = Wb – Ws – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB Wb,#lit5,Wd	Wd = Wb – lit5 – ($\overline{C}$)	1	1	C, DC, N, OV, Z
SUBR	SUBR f	f = WREG – f	1	1	C, DC, N, OV, Z
	SUBR f,WREG	WREG = WREG – f	1	1	C, DC, N, OV, Z
	SUBR Wb,Ws,Wd	Wd = Ws – Wb	1	1	C, DC, N, OV, Z
	SUBR Wb,#lit5,Wd	Wd = lit5 – Wb	1	1	C, DC, N, OV, Z
SUBBR	SUBBR f	f = WREG – f – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBBR f,WREG	WREG = WREG – f – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBBR Wb,Ws,Wd	Wd = Ws – Wb – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBBR Wb,#lit5,Wd	Wd = lit5 – Wb – ($\overline{C}$)	1	1	C, DC, N, OV, Z
SWAP	SWAP.b Wn	Wn = Nibble Swap Wn	1	1	None
	SWAP Wn	Wn = Byte Swap Wn	1	1	None

TABLE 27-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
TBLRDH	TBLRDH	Ws,Wd	Read Prog<23:16> to Wd<7:0>	1	2	None
TBLRDL	TBLRDL	Ws,Wd	Read Prog<15:0> to Wd	1	2	None
TBLWTH	TBLWTH	Ws,Wd	Write Ws<7:0> to Prog<23:16>	1	2	None
TBLWTL	TBLWTL	Ws,Wd	Write Ws to Prog<15:0>	1	2	None
ULNK	ULNK		Unlink Frame Pointer	1	1	None
XOR	XOR	f	f = f .XOR. WREG	1	1	N, Z
	XOR	f,WREG	WREG = f .XOR. WREG	1	1	N, Z
	XOR	#lit10,Wn	Wd = lit10 .XOR. Wd	1	1	N, Z
	XOR	Wb,Ws,Wd	Wd = Wb .XOR. Ws	1	1	N, Z
	XOR	Wb,#lit5,Wd	Wd = Wb .XOR. lit5	1	1	N, Z
ZE	ZE	Ws,Wnd	Wnd = Zero-Extend Ws	1	1	C, Z, N

NOTES:

28.0 ELECTRICAL CHARACTERISTICS

This section provides an overview of the PIC24FJ64GA104 family electrical characteristics. Additional information will be provided in future revisions of this document as it becomes available.

Absolute maximum ratings for the PIC24FJ64GA104 family are listed below. Exposure to these maximum rating conditions for extended periods may affect device reliability. Functional operation of the device at these, or any other conditions above the parameters indicated in the operation listings of this specification, is not implied.

Absolute Maximum Ratings(†)

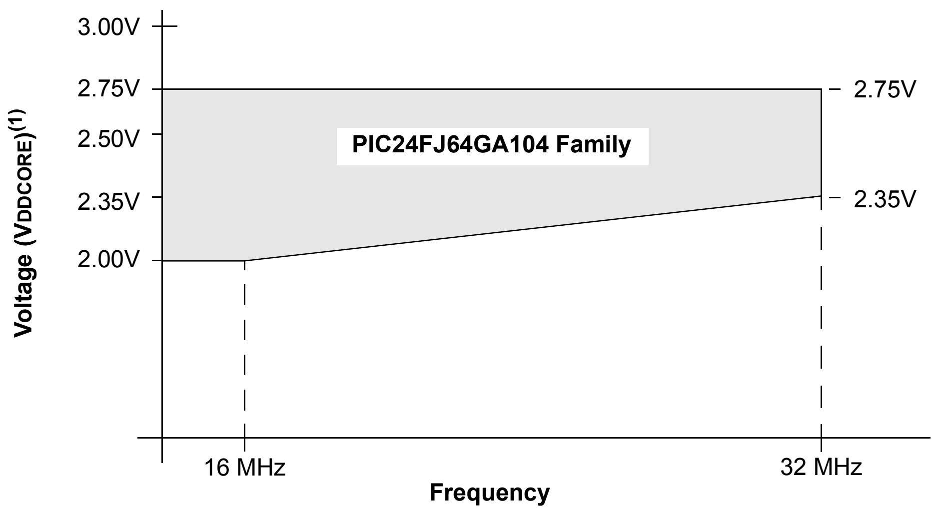
Ambient temperature under bias	-40°C to +135°C
Storage temperature	-65°C to +150°C
Voltage on VDD with respect to VSS	-0.3V to +4.0V
Voltage on any combined analog and digital pin, and $\overline{MCLR}$, with respect to VSS	-0.3V to (VDD + 0.3V)
Voltage on any digital only pin with respect to VSS	-0.3V to +6.0V
Voltage on VDDCORE with respect to VSS	-0.3V to +3.0V
Maximum current out of VSS pin	300 mA
Maximum current into VDD pin **(Note 1)**	250 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 **(Note 1)**	200 mA

Note 1: Maximum allowable current is a function of device maximum power dissipation (see Table 28-1).

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.

28.1 DC Characteristics

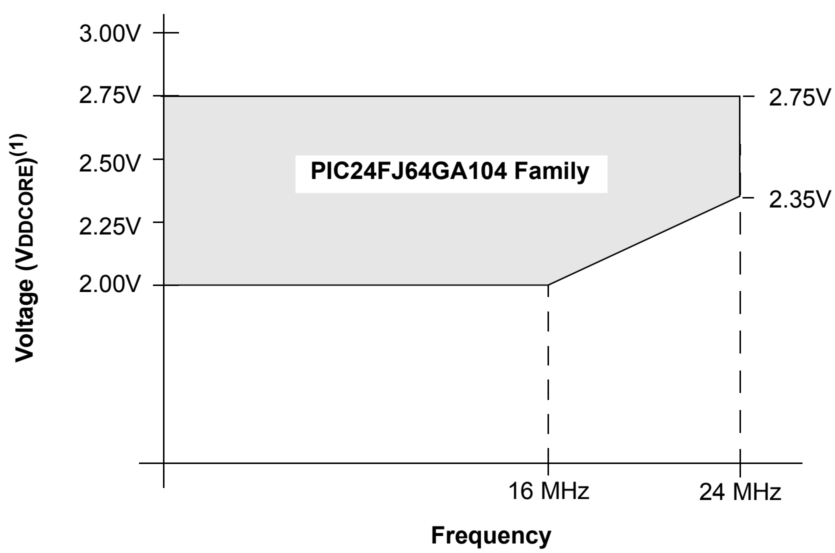
FIGURE 28-1: PIC24FJ64GA104 FAMILY VOLTAGE/FREQUENCY GRAPH (INDUSTRIAL)



For frequencies between 16 MHz and 32 MHz, F_{MAX} = (45.7 MHz/V) * (V_{DDCORE} – 2V) + 16 MHz.

Note 1: When the voltage regulator is disabled, V_{DD} and V_{DDCORE} must be maintained so that $V_{DDCORE} \le V_{DD} \le 3.6V$.

FIGURE 28-2: PIC24FJ64GA104 FAMILY VOLTAGE/FREQUENCY GRAPH (EXTENDED TEMPERATURE)



For frequencies between 16 MHz and 24 MHz, F_{MAX} = (22.9 MHz/V) * (V_{DDCORE} – 2V) + 16 MHz.

Note 1: When the voltage regulator is disabled, V_{DD} and V_{DDCORE} must be maintained so that $V_{DDCORE} \le V_{DD} \le 3.6V$.

TABLE 28-1: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min	Typ	Max	Unit
PIC24FJ64GA104 Family:					
Operating Junction Temperature Range	T_J	-40	—	+140	°C
Operating Ambient Temperature Range	T_A	-40	—	+125	°C
Power Dissipation: Internal Chip Power Dissipation: $P_{INT} = V_{DD} \times (I_{DD} - \Sigma I_{OH})$ I/O Pin Power Dissipation: $P_{I/O} = \Sigma (\{V_{DD} - V_{OH}\} \times I_{OH}) + \Sigma (V_{OL} \times I_{OL})$	P_D	$P_{INT} + P_{I/O}$			W
Maximum Allowed Power Dissipation	P_{DMAX}	$(T_J - T_A)/\theta_{JA}$			W

TABLE 28-2: THERMAL PACKAGING CHARACTERISTICS

Characteristic	Symbol	Typ	Max	Unit	Notes
Package Thermal Resistance, 300 mil SOIC	θ_{JA}	49	—	°C/W	**(Note 1)**
Package Thermal Resistance, 6x6x0.9 mm QFN	θ_{JA}	33.7	—	°C/W	**(Note 1)**
Package Thermal Resistance, 8x8x1 mm QFN	θ_{JA}	28	—	°C/W	**(Note 1)**
Package Thermal Resistance, 10x10x1 mm TQFP	θ_{JA}	39.3	—	°C/W	**(Note 1)**

Note 1: Junction to ambient thermal resistance; Theta-JA (θ_{JA}) numbers are achieved by package simulations.

TABLE 28-3: DC CHARACTERISTICS: TEMPERATURE AND VOLTAGE SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
Operating Voltage							
DC10	**Supply Voltage**						
	VDD		2.2	—	3.6	V	Regulator enabled
	VDD		VDDCORE	—	3.6	V	Regulator disabled
	VDDCORE		2.0	—	2.75	V	Regulator disabled
DC12	VDR	**RAM Data Retention Voltage(2)**	1.5	—	—	V	
DC16	VPOR	**VDD Start Voltage** to Ensure Internal Power-on Reset Signal	VSS	—	—	V	
DC17	SVDD	**VDD Rise Rate** to Ensure Internal Power-on Reset Signal	0.05	—	—	V/ms	0-3.3V in 0.1s 0-2.5V in 60 ms
DC18	VBOR	**Brown-out Reset Voltage**	—	2.05	—	V	

Note 1: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: This is the limit to which VDD can be lowered without losing RAM data.

TABLE 28-4: DC CHARACTERISTICS: OPERATING CURRENT (IDD)

DC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Parameter No.	**Typical[1]**	**Max**	**Units**	**Conditions**		
Operating Current (IDD)[2]						
DC21	0.24	0.395	mA	-40°C	2.0V[3]	0.5 MIPS
DC21a	0.25	0.395	mA	+25°C		
DC21b	0.25	0.395	mA	+85°C		
DC21f	0.3	0.395	mA	+125°C		
DC21c	0.44	0.78	mA	-40°C	3.3V[4]	
DC21d	0.41	0.78	mA	+25°C		
DC21e	0.41	0.78	mA	+85°C		
DC21g	0.6	0.78	mA	+125°C		
DC20	0.5	0.75	mA	-40°C	2.0V[3]	1 MIPS
DC20a	0.5	0.75	mA	+25°C		
DC20b	0.5	0.75	mA	+85°C		
DC20c	0.6	0.75	mA	+125°C		
DC20d	0.75	1.4	mA	-40°C	3.3V[4]	
DC20e	0.75	1.4	mA	+25°C		
DC20f	0.75	1.4	mA	+85°C		
DC20g	1.0	1.4	mA	+125°C		
DC23	2.0	3.0	mA	-40°C	2.0V[3]	4 MIPS
DC23a	2.0	3.0	mA	+25°C		
DC23b	2.0	3.0	mA	+85°C		
DC23c	2.4	3.0	mA	+125°C		
DC23d	2.9	4.2	mA	-40°C	3.3V[4]	
DC23e	2.9	4.2	mA	+25°C		
DC23f	2.9	4.2	mA	+85°C		
DC23g	3.5	4.2	mA	+125°C		

Note 1: Data in “Typical” column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption. The test conditions for all IDD measurements are as follows: OSCI driven with external square wave from rail to rail. All I/O pins are configured as inputs and pulled to VDD. $\overline{MCLR}$ = VDD; WDT and FSCM are disabled. CPU, SRAM, program memory and data memory are operational. No peripheral modules are operating and all of the Peripheral Module Disable (PMD) bits are set.

3: On-chip voltage regulator is disabled (DISVREG is tied to VDD).

4: On-chip voltage regulator is enabled (DISVREG is tied to VSS). Low-Voltage Detect (LVD) and Brown-out Detect (BOD) are enabled.

TABLE 28-4: DC CHARACTERISTICS: OPERATING CURRENT (IDD) (CONTINUED)

DC CHARACTERISTICS				Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended		
Parameter No.	Typical(1)	Max	Units	Conditions		
Operating Current (IDD)(2)						
DC24	10.5	15.5	mA	-40°C	2.5V(3)	16 MIPS
DC24a	10.5	15.5	mA	+25°C		
DC24b	10.5	15.5	mA	+85°C		
DC24c	11.3	15.5	mA	+125°C		
DC24d	11.3	15.5	mA	-40°C	3.3V(4)	
DC24e	11.3	15.5	mA	+25°C		
DC24f	11.3	15.5	mA	+85°C		
DC24g	11.3	15.5	mA	+125°C		
DC31	15.0	18.0	μA	-40°C	2.0V(3)	LPRC (31 kHz)
DC31a	15.0	19.0	μA	+25°C		
DC31b	20.0	36.0	μA	+85°C		
DC31c	42.0	55.0	μA	+125°C		
DC31d	57.0	120.0	μA	-40°C	3.3V(4)	
DC31e	57.0	125.0	μA	+25°C		
DC31f	95.0	160.0	μA	+85°C		
DC31g	114.0	180.0	μA	+125°C		

Note 1: Data in "Typical" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption. The test conditions for all IDD measurements are as follows: OSCI driven with external square wave from rail to rail. All I/O pins are configured as inputs and pulled to VDD. $\overline{\text{MCLR}}$ = VDD; WDT and FSCM are disabled. CPU, SRAM, program memory and data memory are operational. No peripheral modules are operating and all of the Peripheral Module Disable (PMD) bits are set.

3: On-chip voltage regulator is disabled (DISVREG is tied to VDD).

4: On-chip voltage regulator is enabled (DISVREG is tied to VSS). Low-Voltage Detect (LVD) and Brown-out Detect (BOD) are enabled.

TABLE 28-5: DC CHARACTERISTICS: IDLE CURRENT (IIDLE)

DC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Parameter No.	**Typical[1]**	**Max**	**Units**	**Conditions**		
Idle Current (IIDLE)[2]						
DC41	67	100	µA	-40°C	2.0V[3]	0.5 MIPS
DC41a	68	100	µA	+25°C		
DC41b	74	100	µA	+85°C		
DC41f	102	120	µA	+125°C		
DC41c	166	265	µA	-40°C	3.3V[4]	
DC41d	167	265	µA	+25°C		
DC41e	177	265	µA	+85°C		
DC41g	225	285	µA	+125°C		
DC40	125	180	µA	-40°C	2.0V[3]	1 MIPS
DC40a	125	180	µA	+25°C		
DC40b	125	180	µA	+85°C		
DC40c	167	200	µA	+125°C		
DC40d	210	350	µA	-40°C	3.3V[4]	
DC40e	210	350	µA	+25°C		
DC40f	210	350	µA	+85°C		
DC40g	305	370	µA	+125°C		
DC43	0.5	0.6	mA	-40°C	2.0V[3]	4 MIPS
DC43a	0.5	0.6	mA	+25°C		
DC43b	0.5	0.6	mA	+85°C		
DC43c	0.54	0.62	mA	+125°C		
DC43d	0.75	0.95	mA	-40°C	3.3V[4]	
DC43e	0.75	0.95	mA	+25°C		
DC43f	0.75	0.95	mA	+85°C		
DC43g	0.8	0.97	mA	+125°C		
DC47	2.6	3.3	mA	-40°C	2.5V[3]	16 MIPS
DC47a	2.6	3.3	mA	+25°C		
DC47b	2.6	3.3	mA	+85°C		
DC47f	2.7	3.4	mA	+125°C		
DC47c	2.9	3.5	mA	-40°C	3.3V[4]	
DC47d	2.9	3.5	mA	+25°C		
DC47e	2.9	3.5	mA	+85°C		
DC47g	3.0	3.6	mA	+125°C		

Note 1: Data in "Typical" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Base IIDLE current is measured with the core off, OSCI driven with external square wave from rail to rail. All I/O pins are configured as inputs and pulled to VDD. MCLR = VDD; WDT and FSCM are disabled. No peripheral modules are operating and all of the Peripheral Module Disable (PMD) bits are set.

3: On-chip voltage regulator is disabled (DISVREG is tied to VDD).

4: On-chip voltage regulator is enabled (DISVREG is tied to VSS). Low-Voltage Detect (LVD) and Brown-out Detect (BOD) are enabled.

TABLE 28-5: DC CHARACTERISTICS: IDLE CURRENT (IIDLE) (CONTINUED)

DC CHARACTERISTICS				Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended		
Parameter No.	**Typical(1)**	**Max**	**Units**	**Conditions**		
Idle Current (IIDLE)(2)						
DC50	0.8	1.0	mA	-40°C	2.0V(3)	FRC (4 MIPS)
DC50a	0.8	1.0	mA	+25°C		
DC50b	0.8	1.0	mA	+85°C		
DC50c	0.9	1.1	mA	+125°C		
DC50d	1.1	1.3	mA	-40°C	3.3V(4)	
DC50e	1.1	1.3	mA	+25°C		
DC50f	1.1	1.3	mA	+85°C		
DC50g	1.2	1.4	mA	+125°C		
DC51	2.4	8.0	μA	-40°C	2.0V(3)	LPRC (31 kHz)
DC51a	2.2	8.0	μA	+25°C		
DC51b	7.2	21	μA	+85°C		
DC51c	35	50	μA	+125°C		
DC51d	38	55	μA	-40°C	3.3V(4)	
DC51e	44	60	μA	+25°C		
DC51f	70	100	μA	+85°C		
DC51g	96	150	μA	+125°C		

Note 1: Data in "Typical" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Base IIDLE current is measured with the core off, OSCI driven with external square wave from rail to rail. All I/O pins are configured as inputs and pulled to VDD. MCLR = VDD; WDT and FSCM are disabled. No peripheral modules are operating and all of the Peripheral Module Disable (PMD) bits are set.

3: On-chip voltage regulator is disabled (DISVREG is tied to VDD).

4: On-chip voltage regulator is enabled (DISVREG is tied to VSS). Low-Voltage Detect (LVD) and Brown-out Detect (BOD) are enabled.

TABLE 28-6: DC CHARACTERISTICS: POWER-DOWN BASE CURRENT (IPD)

DC CHARACTERISTICS				Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended		
Parameter No.	**Typical(1)**	**Max**	**Units**	**Conditions**		
Power-Down Current (IPD)(2)						
DC60	0.05	1.0	μA	-40°C	2.0V(3)	Base Power-Down Current(5)
DC60a	0.2	1.0	μA	+25°C		
DC60i	2.0	6.5	μA	+60°C		
DC60b	3.5	12	μA	+85°C		
DC60m	29.9	50	μA	+125°C		
DC60c	0.1	1.0	μA	-40°C	2.5V(3)	
DC60d	0.4	1.0	μA	+25°C		
DC60j	2.5	15	μA	+60°C		
DC60e	4.2	25	μA	+85°C		
DC60n	36.2	75	μA	+125°C		
DC60f	3.3	9.0	μA	-40°C	3.3V(4)	
DC60g	3.3	10	μA	+25°C		
DC60k	5.0	20	μA	+60°C		
DC60h	7.0	30	μA	+85°C		
DC60p	39.2	80	μA	+125°C		
DC70c	0.003	0.2	μA	-40°C	2.5V(4)	Base Deep Sleep Current
DC70d	0.02	0.2	μA	+25°C		
DC70j	0.2	0.35	μA	+60°C		
DC70e	0.51	1.5	μA	+85°C		
DC70a	6.1	12	μA	+125°C		
DC70f	0.01	0.3	μA	-40°C	3.3V(4)	
DC70g	0.04	0.3	μA	+25°C		
DC70k	0.2	0.5	μA	+60°C		
DC70h	0.71	2.0	μA	+85°C		
DC70b	7.2	16	μA	+125°C		

Note 1: Data in the Typical column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Base IPD is measured with the device in Sleep mode (all peripherals and clocks shut down). All I/Os are configured as inputs and pulled high. WDT, etc., are all switched off, PMSLP bit is clear and the Peripheral Module Disable (PMD) bits for all unused peripherals are set.

3: On-chip voltage regulator is disabled (DISVREG is tied to VDD).

4: On-chip voltage regulator is enabled (DISVREG is tied to VSS). Low-Voltage Detect (LVD) and Brown-out Detect (BOD) are enabled.

5: The Δ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

TABLE 28-7: DC CHARACTERISTICS: POWER-DOWN PERIPHERAL MODULE Δ CURRENT (IPD)

DC CHARACTERISTICS				Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended		
Parameter No.	**Typical(1)**	**Max**	**Units**	**Conditions**		
Δ Power-Down Current (IPD): PMD Bits are Set, PMSLP Bit is '0'(2)						
DC61	0.2	0.7	μA	-40°C	2.0V(3)	31 kHz LPRC Oscillator with RTCC, WDT, DSWDT or Timer 1: ΔILPRC(5)
DC61a	0.2	0.7	μA	+25°C		
DC61i	0.2	0.7	μA	+60°C		
DC61b	0.23	0.7	μA	+85°C		
DC61m	0.3	1.0	μA	+125°C		
DC61c	0.25	0.9	μA	-40°C	2.5V(3)	
DC61d	0.25	0.9	μA	+25°C		
DC61j	0.25	0.9	μA	+60°C		
DC61e	0.28	0.9	μA	+85°C		
DC61p	0.5	1.2	μA	+125°C		
DC61f	0.6	1.5	μA	-40°C	3.3V(4)	
DC61g	0.6	1.5	μA	+25°C		
DC61k	0.6	1.5	μA	+60°C		
DC61h	0.8	1.5	μA	+85°C		
DC61n	1.0	1.7	μA	+125°C		
DC62	0.5	1.0	μA	-40°C	2.0V(3)	Low drive strength, 32 kHz Crystal with RTCC, DSWDT or Timer1: ΔISOSC; SOSCSEL = 01
DC62a	0.5	1.0	μA	+25°C		
DC62i	0.5	1.0	μA	+60°C		
DC62b	0.5	1.3	μA	+85°C		
DC62m	0.6	1.6	μA	+125°C		
DC62c	0.7	1.5	μA	-40°C	2.5V(3)	
DC62d	0.7	1.5	μA	+25°C		
DC62j	0.7	1.5	μA	+60°C		
DC62e	0.7	1.8	μA	+85°C		
DC62n	0.8	2.1	μA	+125°C		
DC62f	1.5	2.0	μA	-40°C	3.3V(4)	
DC62g	1.5	2.0	μA	+25°C		
DC62k	1.5	2.0	μA	+60°C		
DC62h	1.5	2.5	μA	+85°C		
DC62p	1.9	3.0	μA	+125°C		

Note 1: Data in the Typical column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Peripheral IPD deltas are measured with the device in Sleep mode (all peripherals and clocks shut down). All I/Os are configured as inputs and pulled high. Only the peripheral or clock being measured is enabled. PMSLP bit is clear and the Peripheral Module Disable bits (PMD) for all unused peripherals are set.

3: On-chip voltage regulator is disabled (DISVREG is tied to VDD).

4: On-chip voltage regulator is enabled (DISVREG is tied to VSS). Low-Voltage Detect (LVD) and Brown-out Detect (BOD) are enabled.

5: The Δ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

TABLE 28-7: DC CHARACTERISTICS: POWER-DOWN PERIPHERAL MODULE Δ CURRENT (IPD) (CONTINUED)

DC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Parameter No.	**Typical[1]**	**Max**	**Units**	**Conditions**		
Δ Power-Down Current (IPD): PMD Bits are Set, PMSLP Bit is '0'[2]						
DC63	1.8	2.3	μA	-40°C	2.0V[3]	32 kHz Crystal with RTCC, DSWDT or Timer1: ΔISOSC; SOSCSEL = 11[5]
DC63a	1.8	2.7	μA	+25°C		
DC63i	1.8	3.0	μA	+60°C		
DC63b	1.8	3.0	μA	+85°C		
DC63m	2.2	3.3	μA	+125°C		
DC63c	2	2.7	μA	-40°C	2.5V[3]	
DC63d	2	2.9	μA	+25°C		
DC63j	2	3.2	μA	+60°C		
DC63e	2	3.5	μA	+85°C		
DC63n	2.5	3.8	μA	+125°C		
DC63f	2.25	3.0	μA	-40°C	3.3V[4]	
DC63g	2.25	3.0	μA	+25°C		
DC63k	2.25	3.3	μA	+60°C		
DC63h	2.25	3.5	μA	+85°C		
DC63p	2.8	4.0	μA	+125°C		
DC71c	0.001	0.25	μA	-40°C	2.5V[4]	Deep Sleep BOR: ΔIDSBOR
DC71d	0.03	0.25	μA	+25°C		
DC71j	0.05	0.60	μA	+60°C		
DC71e	0.08	2.0	μA	+85°C		
DC71a	3.9	10	μA	+125°C		
DC71f	0.001	0.50	μA	-40°C	3.3V[4]	
DC71g	0.03	0.50	μA	+25°C		
DC71k	0.05	0.75	μA	+60°C		
DC71h	0.08	2.5	μA	+85°C		
DC71b	3.9	12.5	μA	+125°C		

Note 1: Data in the Typical column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Peripheral IPD deltas are measured with the device in Sleep mode (all peripherals and clocks shut down). All I/Os are configured as inputs and pulled high. Only the peripheral or clock being measured is enabled. PMSLP bit is clear and the Peripheral Module Disable bits (PMD) for all unused peripherals are set.

3: On-chip voltage regulator is disabled (DISVREG is tied to VDD).

4: On-chip voltage regulator is enabled (DISVREG is tied to VSS). Low-Voltage Detect (LVD) and Brown-out Detect (BOD) are enabled.

5: The Δ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

TABLE 28-8: DC CHARACTERISTICS: I/O PIN INPUT SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ[1]**	**Max**	**Units**	**Conditions**
	VIL	**Input Low Voltage[4]**					
DI10		I/O Pins with ST Buffer	VSS	—	0.2 VDD	V	
DI11		I/O Pins with TTL Buffer	VSS	—	0.15 VDD	V	
DI15		$\overline{\text{MCLR}}$	VSS	—	0.2 VDD	V	
DI16		OSC1 (XT mode)	VSS	—	0.2 VDD	V	
DI17		OSC1 (HS mode)	VSS	—	0.2 VDD	V	
DI18		I/O Pins with I^2C™ Buffer:	VSS	—	0.3 VDD	V	
DI19		I/O Pins with SMBus Buffer:	VSS	—	0.8	V	SMBus enabled
	VIH	**Input High Voltage[4]**					
DI20		I/O Pins with ST Buffer: with Analog Functions, Digital Only	 0.8 VDD 0.8 VDD	 — —	 VDD 5.5	 V V	
DI21		I/O Pins with TTL Buffer: with Analog Functions, Digital Only	 0.25 VDD + 0.8 0.25 VDD + 0.8	 — —	 VDD 5.5	 V V	
DI25		$\overline{\text{MCLR}}$	0.8 VDD	—	VDD	V	
DI26		OSC1 (XT mode)	0.7 VDD	—	VDD	V	
DI27		OSC1 (HS mode)	0.7 VDD	—	VDD	V	
DI28		I/O Pins with I^2C Buffer: with Analog Functions, Digital Only	 0.7 VDD 0.7 VDD	 — —	 VDD 5.5	 V V	
DI29		I/O Pins with SMBus Buffer: with Analog Functions, Digital Only	 2.1 2.1		 VDD 5.5	 V V	2.5V ≤ VPIN ≤ VDD
DI30	ICNPU	**CNx Pull-up Current**	50	250	400	μA	VDD = 3.3V, VPIN = VSS
	IIL	**Input Leakage Current[2,3]**					
DI50		I/O Ports	—	—	±50	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance
DI51		Analog Input Pins	—	—	±50	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance
DI55		$\overline{\text{MCLR}}$	—	—	±50	nA	VSS ≤ VPIN ≤ VDD
DI56		OSC1	—	—	±50	nA	VSS ≤ VPIN ≤ VDD, XT and HS modes

Note 1: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as current sourced by the pin.

4: Refer to Table 1-2 for I/O pins buffer types.

TABLE 28-9: DC CHARACTERISTICS: I/O PIN OUTPUT SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
	VOL	**Output Low Voltage**					
DO10		I/O Ports	—	—	0.4	V	IOL = 8.5 mA, VDD = 3.6V
			—	—	0.4	V	IOL = 5.0 mA, VDD = 2.0V
DO16		I/O Ports	—	—	0.4	V	IOL = 8.0 mA, VDD = 3.6V, 125°C
			—	—	0.4	V	IOL = 4.5 mA, VDD = 2.0V, 125°C
	VOH	**Output High Voltage**					
DO20		I/O Ports	3.0	—	—	V	IOH = -3.0 mA, VDD = 3.6V
			2.4	—	—	V	IOH = -6.0 mA, VDD = 3.6V
			1.65	—	—	V	IOH = -1.0 mA, VDD = 2.0V
			1.4	—	—	V	IOH = -3.0 mA, VDD = 2.0V
DO26		I/O Ports	3.0	—	—	V	IOH = -2.5 mA, VDD = 3.6V, 125°C
			1.65	—	—	V	IOH = -0.5 mA, VDD = 2.0V, 125°C

Note 1: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

TABLE 28-10: DC CHARACTERISTICS: PROGRAM MEMORY

DC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
D130	EP	Cell Endurance	10,000	—	—	E/W	-40°C to +85°C
D131	VPR	VDD for Read	VMIN	—	3.6	V	VMIN = Minimum operating voltage
	VPEW	Supply Voltage for Self-Timed Writes					
D132A		VDDCORE	2.25	—	3.6	V	
D132B		VDD	2.35	—	3.6	V	
D133A	TIW	Self-Timed Write Cycle Time	—	3	—	ms	
D133B	TIE	Self-Timed Page Erase Time	40	—	—	ms	
D134	TRETD	Characteristic Retention	20	—	—	Year	Provided no other specifications are violated
D135	IDDP	Supply Current during Programming	—	7	—	mA	

Note 1: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated.

TABLE 28-11: COMPARATOR SPECIFICATIONS

Operating Conditions: 2.0V < VDD < 3.6V, -40°C < TA < +85°C (unless otherwise stated)							
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Comments**
D300	VIOFF	Input Offset Voltage*	—	20	40	mV	
D301	VICM	Input Common Mode Voltage*	0	—	VDD	V	
D302	CMRR	Common Mode Rejection Ratio*	55	—	—	dB	
300	TRESP	Response Time*(1)	—	150	400	ns	
301	TMC2OV	Comparator Mode Change to Output Valid*	—	—	10	µs	

* Parameters are characterized but not tested.

Note 1: Response time measured with one comparator input at (VDD – 1.5)/2, while the other input transitions from VSS to VDD.

TABLE 28-12: COMPARATOR VOLTAGE REFERENCE SPECIFICATIONS

Operating Conditions: 2.0V < VDD < 3.6V, -40°C < TA < +85°C (unless otherwise stated)							
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Comments**
VRD310	CVRES	Resolution	VDD/24	—	VDD/32	LSb	
VRD311	CVRAA	Absolute Accuracy	—	—	AVDD – 1.5	LSb	
VRD312	CVRUR	Unit Resistor Value (R)	—	2k	—	Ω	
VR310	TSET	Settling Time(1)	—	—	10	µs	

Note 1: Settling time measured while CVRR = 1 and CVR<3:0> bits transition from '0000' to '1111'.

TABLE 28-13: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

Operating Conditions: -40°C < TA < +85°C (unless otherwise stated)							
Param No.	**Symbol**	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
	VBG	Band Gap Reference Voltage	1.14	1.2	1.26	V	
	TBG	Band Gap Reference Start-up Time	—	1	—	ms	
	VRGOUT	Regulator Output Voltage	2.35	2.5	2.75	V	
	CEFC	External Filter Capacitor Value	4.7	10	—	µF	Series resistance < 3 Ohm recommended; < 5 Ohm required.

28.2 AC Characteristics and Timing Parameters

The information contained in this section defines the PIC24FJ64GA104 family AC characteristics and timing parameters.

TABLE 28-14: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

AC CHARACTERISTICS	**Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial and -40°C ≤ TA ≤ +125°C for Extended Operating voltage VDD range as described in **Section 28.1 "DC Characteristics"**.

FIGURE 28-3: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS

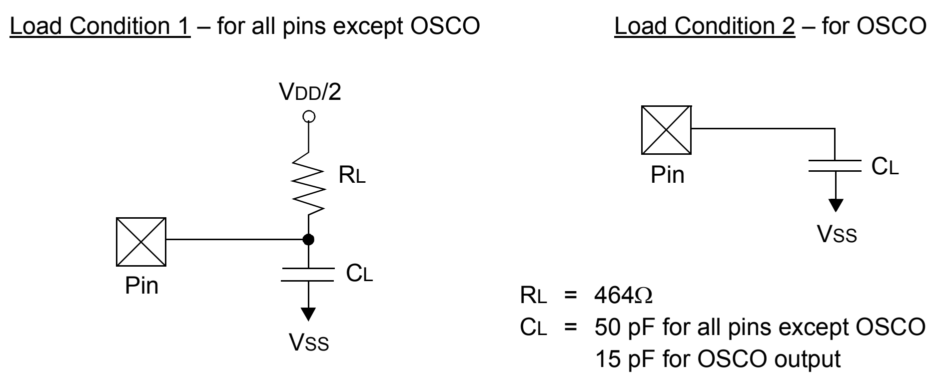


TABLE 28-15: CAPACITIVE LOADING REQUIREMENTS ON OUTPUT PINS

Param No.	Symbol	Characteristic	Min	Typ(1)	Max	Units	Conditions
DO50	COSC2	OSCO/CLKO Pin	—	—	15	pF	In XT and HS modes when external clock is used to drive OSCI.
DO56	CIO	All I/O Pins and OSCO	—	—	50	pF	EC mode.
DO58	CB	SCLx, SDAx	—	—	400	pF	In I^2C™ mode.

Note 1: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

FIGURE 28-4: EXTERNAL CLOCK TIMING

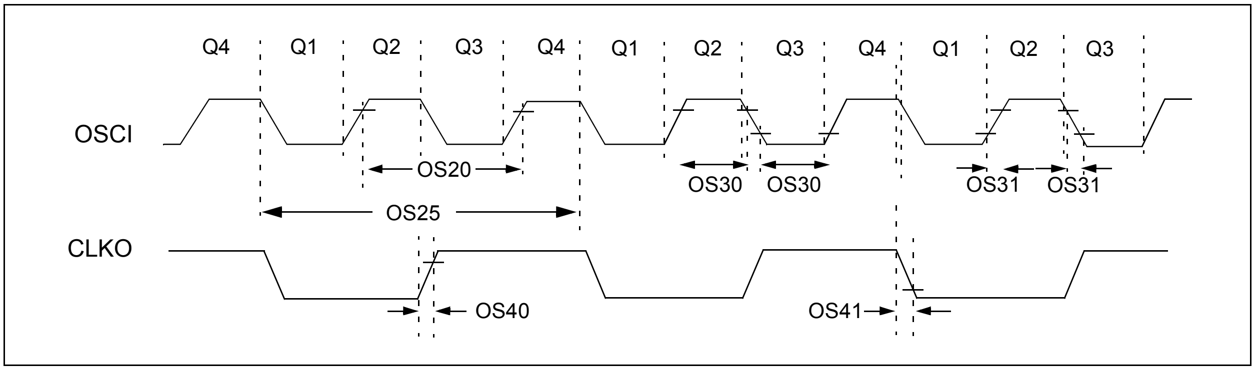


TABLE 28-16: EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.50 to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
OS10	FOSC	External CLKI Frequency (External clocks allowed only in EC mode)	DC 4 DC 4	— — — —	32 8 24 6	MHz MHz MHz MHz	EC, -40°C ≤ TA ≤ +85°C ECPLL, -40°C ≤ TA ≤ +85°C EC, -40°C ≤ TA ≤ +125°C ECPLL, -40°C ≤ TA ≤ +125°C
		Oscillator Frequency	3 3 10 31 3 10	— — — — — —	10 8 32 33 6 24	MHz MHz MHz kHz MHz MHz	XT XTPLL, -40°C ≤ TA ≤ +85°C HS, -40°C ≤ TA ≤ +85°C SOSC XTPLL, -40°C ≤ TA ≤ +125°C HS, -40°C ≤ TA ≤ +125°C
OS20	TOSC	TOSC = 1/FOSC	—	—	—	—	See parameter OS10 for FOSC value
OS25	TCY	Instruction Cycle Time(2)	62.5	—	DC	ns	
OS30	TosL, TosH	External Clock in (OSCI) High or Low Time	0.45 x TOSC	—	—	ns	EC
OS31	TosR, TosF	External Clock in (OSCI) Rise or Fall Time	—	—	20	ns	EC
OS40	TckR	CLKO Rise Time(3)	—	6	10	ns	
OS41	TckF	CLKO Fall Time(3)	—	6	10	ns	

Note 1: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Instruction cycle period (TCY) equals two times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “Min.” values with an external clock applied to the OSCI/CLKI pin. When an external clock input is used, the “Max.” cycle time limit is “DC” (no clock) for all devices.

3: Measurements are taken in EC mode. The CLKO signal is measured on the OSCO pin. CLKO is low for the Q1-Q2 period (1/2 TCY) and high for the Q3-Q4 period (1/2 TCY).

TABLE 28-17: PLL CLOCK TIMING SPECIFICATIONS (VDD = 2.0V TO 3.6V)

AC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Sym	Characteristic(1)	Min	Typ(2)	Max	Units	Conditions
OS50	FPLLI	PLL Input Frequency Range	3	—	8	MHz	ECPLL, HSPLL, XTPLL modes, -40°C ≤ TA ≤ +85°C
			3	—	6	MHz	ECPLL, HSPLL, XTPLL modes, -40°C ≤ TA ≤ +125°C
OS51	FSYS	PLL Output Frequency Range	8	—	32	MHz	-40°C ≤ TA ≤ +85°C
			8	—	24	MHz	-40°C ≤ TA ≤ +125°C
OS52	TLOCK	PLL Start-up Time (Lock Time)	—	—	2	ms	
OS53	DCLK	CLKO Stability (Jitter)	-2	1	2	%	Measured over 100 ms period

Note 1: These parameters are characterized but not tested in manufacturing.

2: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

TABLE 28-18: INTERNAL RC OSCILLATOR SPECIFICATIONS

AC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Sym	Characteristic(1)	Min	Typ	Max	Units	Conditions
	TFRC	FRC Start-up Time	—	15	—	μs	
	TLPRC	LPRC Start-up Time	—	500	—	μs	

TABLE 28-19: INTERNAL RC OSCILLATOR ACCURACY

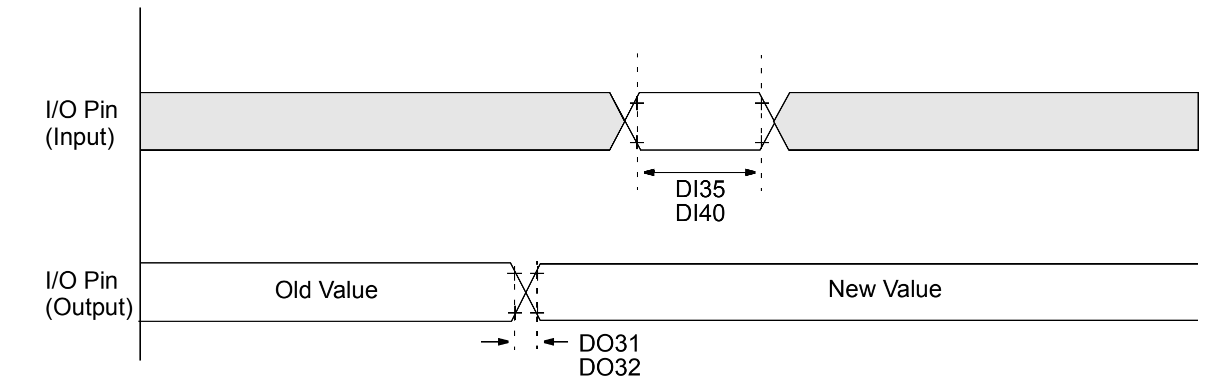
AC CHARACTERISTICS		Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Characteristic	Min	Typ	Max	Units	Conditions
F20	FRC Accuracy @ 8 MHz(1,3)	-1.25	±0.25	1.0	%	-40°C ≤ TA ≤ +85°C, 3.0V ≤ VDD ≤ 3.6V
F21	LPRC Accuracy @ 31 kHz(2)	-15	—	15	%	-40°C ≤ TA ≤ +85°C, 3.0V ≤ VDD ≤ 3.6V

Note 1: Frequency calibrated at 25°C and 3.3V. OSCTUN bits can be used to compensate for temperature drift.

2: Change of LPRC frequency as VDD changes.

3: To achieve this accuracy, physical stress applied to the microcontroller package (ex: by flexing the PCB) must be kept to a minimum.

FIGURE 28-5: CLKO AND I/O TIMING CHARACTERISTICS



Note: Refer to Figure 28-3 for load conditions.

TABLE 28-20: CLKO AND I/O TIMING REQUIREMENTS

AC CHARACTERISTICS			**Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
DO31	TIOR	Port Output Rise Time	—	10	25	ns	
DO32	TIOF	Port Output Fall Time	—	10	25	ns	
DI35	TINP	INTx pin High or Low Time (output)	20	—	—	ns	
DI40	TRBP	CNx High or Low Time (input)	2	—	—	TCY	

Note 1: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated.

TABLE 28-21: RESET, POWER-UP TIMER AND BROWN-OUT RESET TIMING REQUIREMENTS

AC CHARACTERISTICS			**Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ(1)**	**Max.**	**Units**	**Conditions**
SY10	TmcL	MCLR Pulse Width (low)	2	—	—	μs	
SY11	TPWRT	Power-up Timer Period	—	64	—	ms	
SY12	TPOR	Power-on Reset Delay	—	2	—	μs	
SY13	TIOZ	I/O High-Impedance from MCLR Low or Watchdog Timer Reset	—	—	100	ns	
SY25	TBOR	Brown-out Reset Pulse Width	1	—	—	μs	VDD ≤ VBOR
	TRST	Internal State Reset Time	—	50	—	μs	
	TDSWU	Wake-up from Deep Sleep Time	—	200	—	μs	Based on full discharge of 10 μF capacitor on VCAP. Includes TPOR and TRST.
	TPM		— —	10 190	— —	μs μs	Sleep wake-up with PMSLP = 0 and WUTSEL<1:0> = 11

Note 1: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated.

TABLE 28-22: ADC MODULE SPECIFICATIONS

AC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic	Min.	Typ	Max.	Units	Conditions
Device Supply							
AD01	AVDD	Module VDD Supply	Greater of VDD – 0.3 or 2.0	—	Lesser of VDD + 0.3 or 3.6	V	
AD02	AVSS	Module VSS Supply	VSS – 0.3	—	VSS + 0.3	V	
Reference Inputs							
AD05	VREFH	Reference Voltage High	AVSS + 1.7	—	AVDD	V	
AD06	VREFL	Reference Voltage Low	AVSS	—	AVDD – 1.7	V	
AD07	VREF	Absolute Reference Voltage	AVSS – 0.3	—	AVDD + 0.3	V	
AD08	IVREF	Reference Voltage Input Current	—	—	1.25	mA	**(Note 3)**
AD09	ZVREF	Reference Input Impedance	—	10K	—	Ω	**(Note 4)**
Analog Input							
AD10	VINH-VINL	Full-Scale Input Span	VREFL	—	VREFH	V	**(Note 2)**
AD11	VIN	Absolute Input Voltage	AVSS – 0.3	—	AVDD + 0.3	V	
AD12	VINL	Absolute VINL Input Voltage	AVSS – 0.3	—	AVDD/2	V	
AD13	—	Leakage Current	—	±0.001	±0.610	µA	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V, Source Impedance = 2.5 kΩ
AD17	RIN	Recommended Impedance of Analog Voltage Source	—	—	2.5K	Ω	10-bit
ADC Accuracy							
AD20b	NR	Resolution	—	10	—	bits	
AD21b	INL	Integral Nonlinearity	—	±1	<±2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD22b	DNL	Differential Nonlinearity	—	±0.5	<±1.25	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD23b	GERR	Gain Error	—	±1	±3	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD24b	EOFF	Offset Error	—	±1	±2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD25b	—	Monotonicity(1)	—	—	—	—	Guaranteed

Note 1: The ADC conversion result never decreases with an increase in the input voltage and has no missing codes.
2: Measurements taken with external VREF+ and VREF- are used as the ADC voltage reference.
3: External reference voltage is applied to the VREF+/- pins. IVREF is current during conversion at 3.3V, 25°C. Parameter is for design guidance only and is not tested.
4: Impedance during sampling at 3.3V, 25°C. Parameter is for design guidance only and is not tested.

TABLE 28-23: ADC CONVERSION TIMING REQUIREMENTS(1)

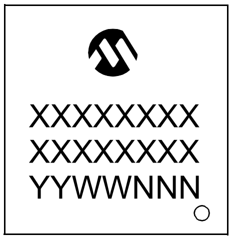
AC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
Clock Parameters							
AD50	TAD	ADC Clock Period	75	—	—	ns	TCY = 75 ns, AD1CON3 in default state
AD51	tRC	ADC Internal RC Oscillator Period	—	250	—	ns	
Conversion Rate							
AD55	tCONV	Conversion Time	—	12	—	TAD	
AD56	FCNV	Throughput Rate	—	—	500	ksps	AVDD > 2.7V
AD57	tSAMP	Sample Time	—	1	—	TAD	
Clock Parameters							
AD61	tPSS	Sample Start Delay from setting Sample bit (SAMP)	2	—	3	TAD	

Note 1: Because the sample capacitors will eventually lose charge, clock rates below 10 kHz can affect linearity performance, especially at elevated temperatures.

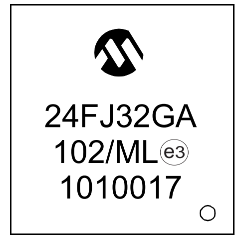
29.0 PACKAGING INFORMATION

29.1 Package Marking Information

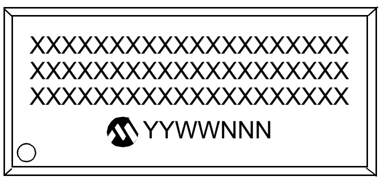
28-Lead QFN



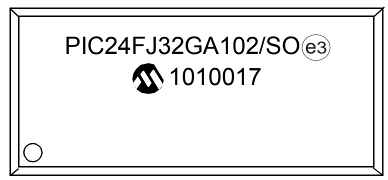
Example



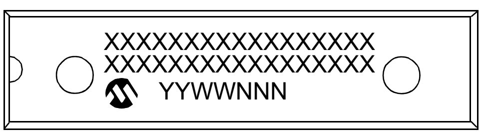
28-Lead SOIC (.300")



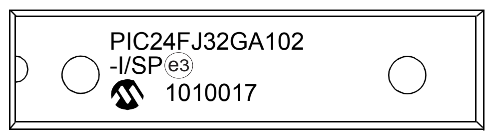
Example



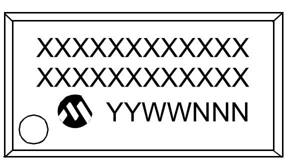
28-Lead SPDIP



Example



28-Lead SSOP

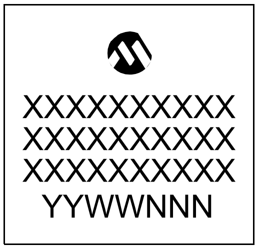


Example

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
		Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

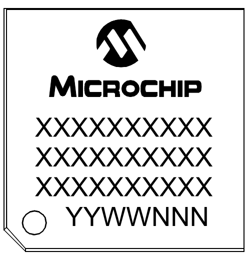
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

44-Lead QFN

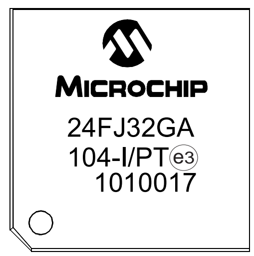


Example

44-Lead TQFP



Example

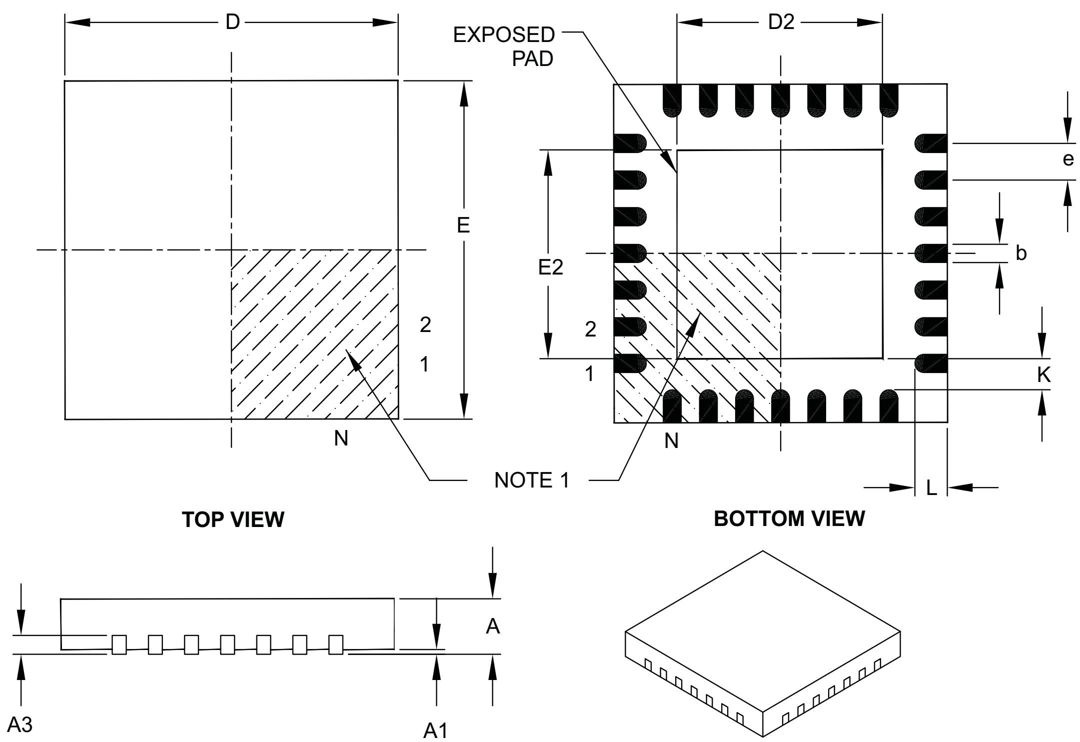


29.2 Package Details

The following sections give the technical details of the packages.

28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.65 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.20 REF	
Overall Width	E		6.00 BSC	
Exposed Pad Width	E2	3.65	3.70	4.20
Overall Length	D		6.00 BSC	
Exposed Pad Length	D2	3.65	3.70	4.20
Contact Width	b	0.23	0.30	0.35
Contact Length	L	0.50	0.55	0.70
Contact-to-Exposed Pad	K	0.20	–	–

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.

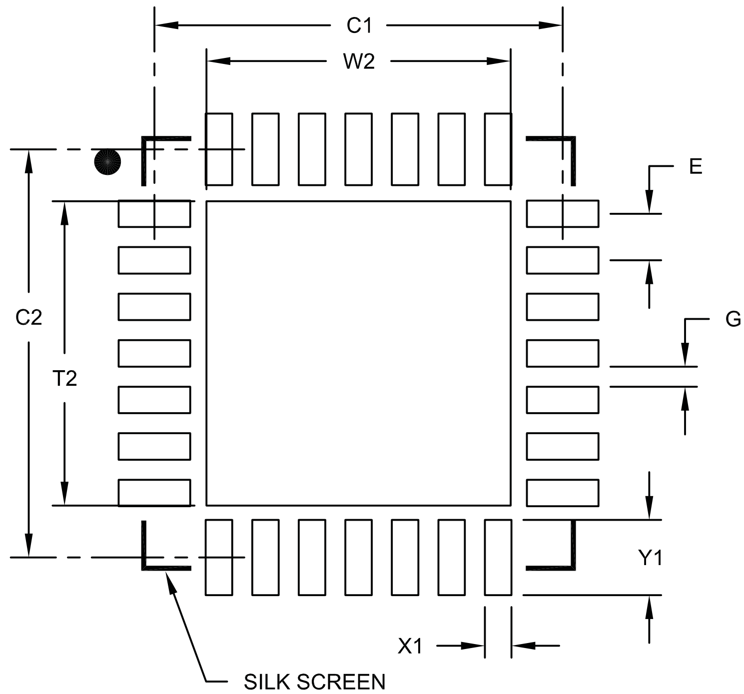
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-105B

28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Optional Center Pad Width	W2			4.25
Optional Center Pad Length	T2			4.25
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.37
Contact Pad Length (X28)	Y1			1.00
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

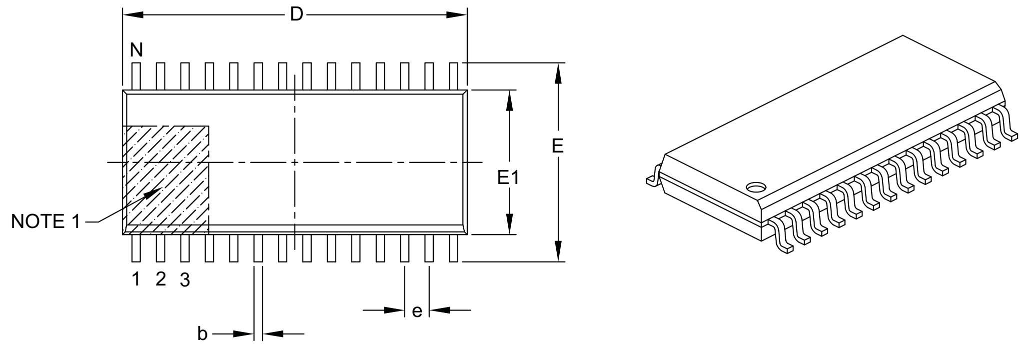
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

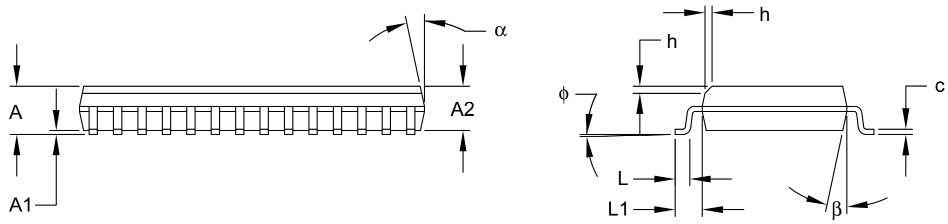
Microchip Technology Drawing No. C04-2105A

查询PIC24FJ64GA104供应商

28-Lead Plastic Small Outline (SO) – Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging





		Units	MILLIMETERS		
		Dimension Limits	MIN	NOM	MAX
Number of Pins	N		28		
Pitch	e		1.27 BSC		
Overall Height	A		–	–	2.65
Molded Package Thickness	A2		2.05	–	–
Standoff §	A1		0.10	–	0.30
Overall Width	E		10.30 BSC		
Molded Package Width	E1		7.50 BSC		
Overall Length	D		17.90 BSC		
Chamfer (optional)	h		0.25	–	0.75
Foot Length	L		0.40	–	1.27
Footprint	L1		1.40 REF		
Foot Angle Top	φ		0°	–	8°
Lead Thickness	c		0.18	–	0.33
Lead Width	b		0.31	–	0.51
Mold Draft Angle Top	α		5°	–	15°
Mold Draft Angle Bottom	β		5°	–	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

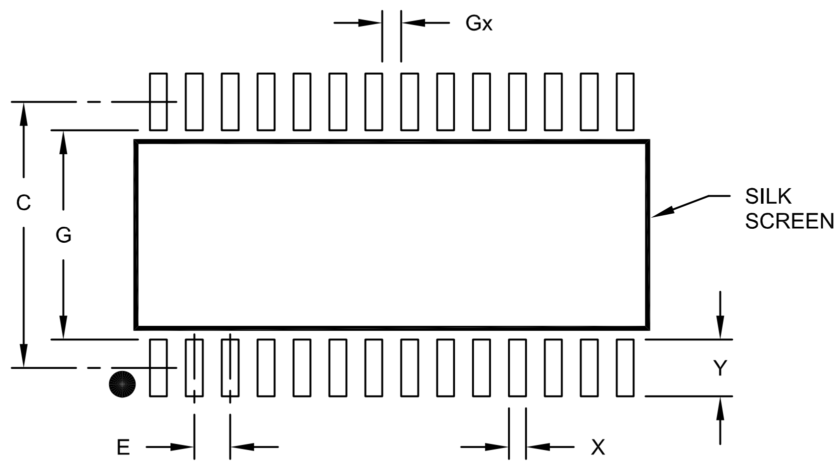
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-052B

查询PIC24FJ64GA104供应商

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	C		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

Notes:

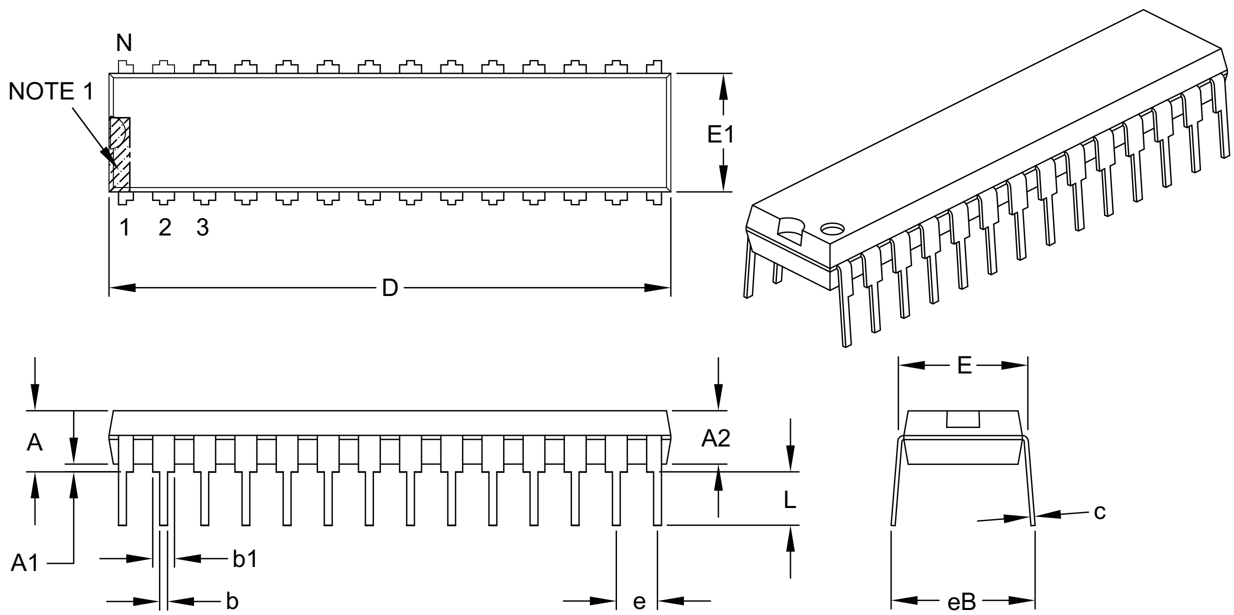
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2052A

28-Lead Skinny Plastic Dual In-Line (SP) – 300 mil Body [SPDIP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

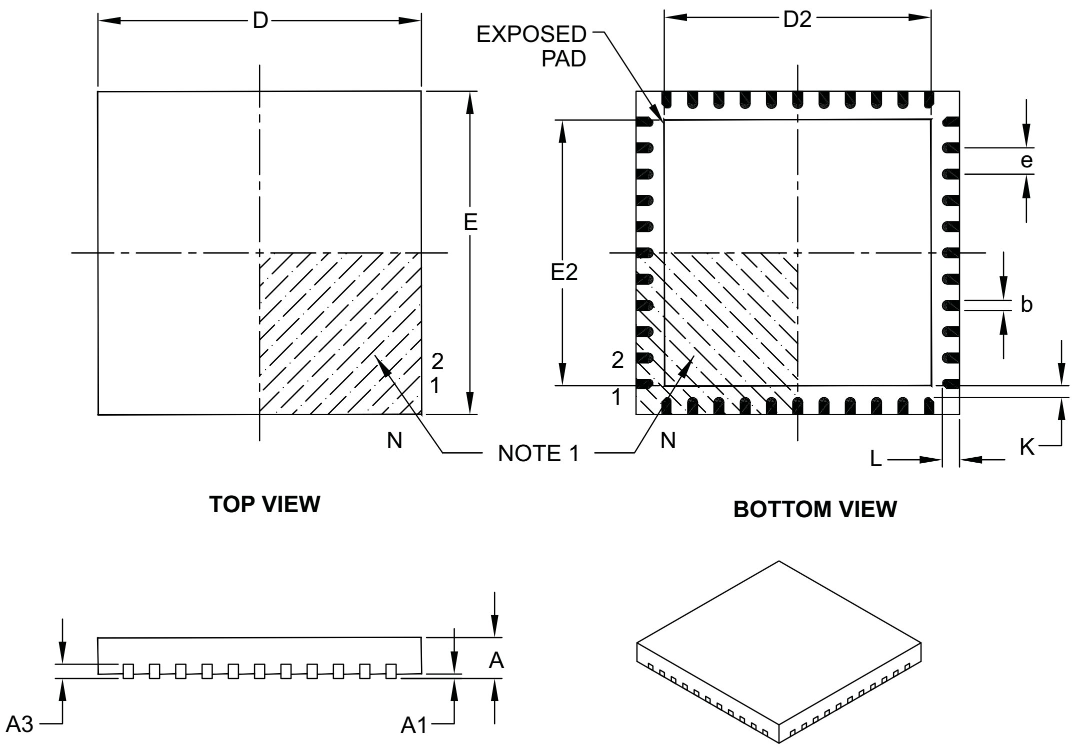
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-070B

44-Lead Plastic Quad Flat, No Lead Package (ML) – 8x8 mm Body [QFN]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



		Units	MILLIMETERS		
		Dimension Limits	MIN	NOM	MAX
Number of Pins	N			44	
Pitch	e			0.65 BSC	
Overall Height	A		0.80	0.90	1.00
Standoff	A1		0.00	0.02	0.05
Contact Thickness	A3			0.20 REF	
Overall Width	E			8.00 BSC	
Exposed Pad Width	E2		6.30	6.45	6.80
Overall Length	D			8.00 BSC	
Exposed Pad Length	D2		6.30	6.45	6.80
Contact Width	b		0.25	0.30	0.38
Contact Length	L		0.30	0.40	0.50
Contact-to-Exposed Pad	K		0.20	–	–

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.

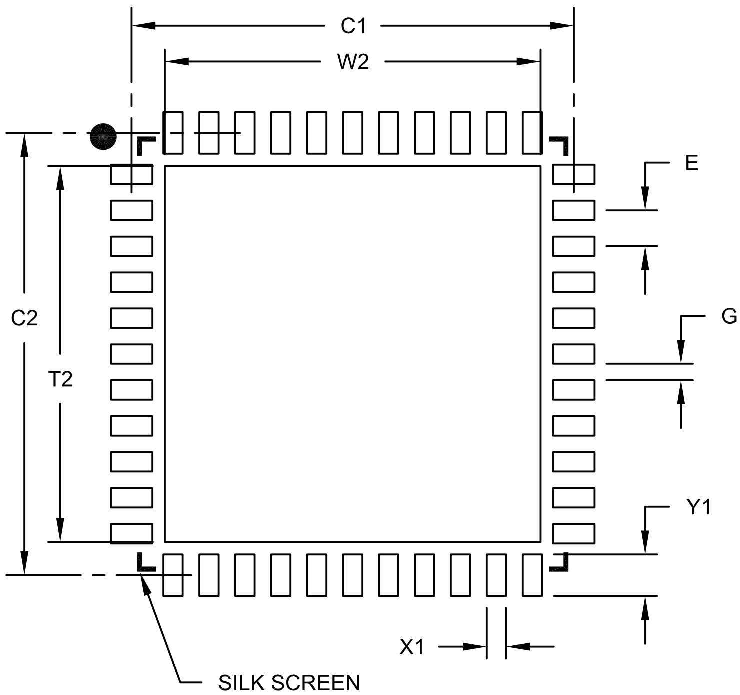
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-103B

44-Lead Plastic Quad Flat, No Lead Package (ML) – 8x8 mm Body [QFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Optional Center Pad Width	W2			6.80
Optional Center Pad Length	T2			6.80
Contact Pad Spacing	C1		8.00	
Contact Pad Spacing	C2		8.00	
Contact Pad Width (X44)	X1			0.35
Contact Pad Length (X44)	Y1			0.80
Distance Between Pads	G	0.25		

Notes:

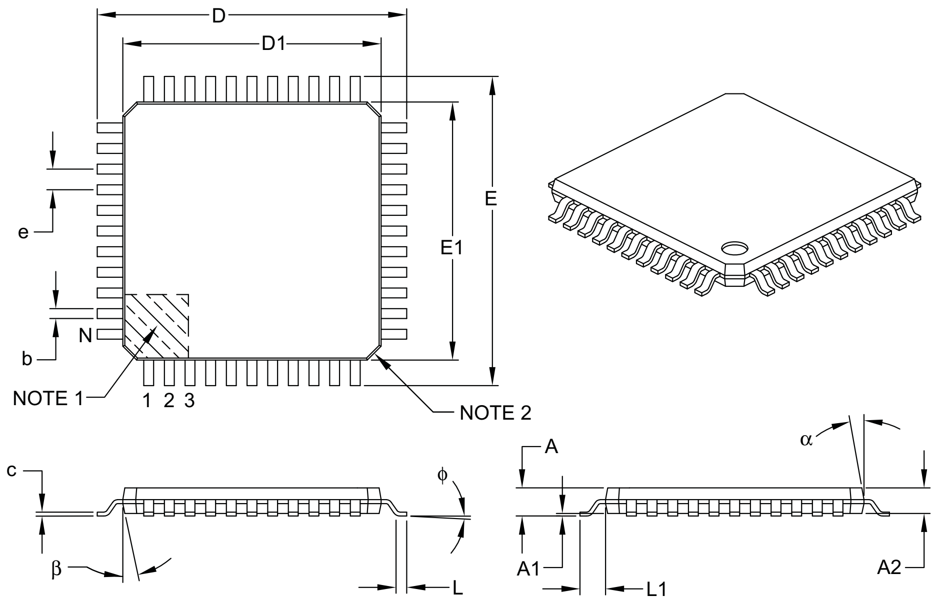
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2103A

44-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads	N		44	
Lead Pitch	e		0.80 BSC	
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	–	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Foot Angle	φ	0°	3.5°	7°
Overall Width	E		12.00 BSC	
Overall Length	D		12.00 BSC	
Molded Package Width	E1		10.00 BSC	
Molded Package Length	D1		10.00 BSC	
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.30	0.37	0.45
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

Notes:

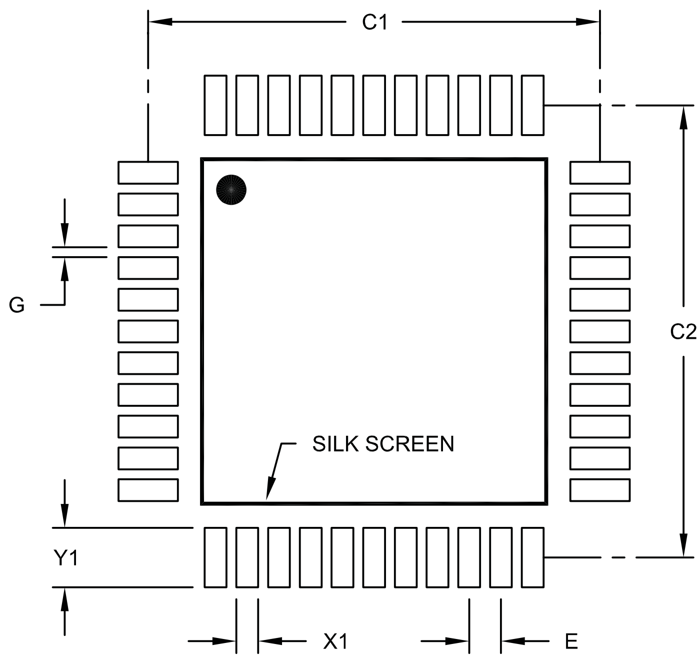
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-076B

44-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

		Units	MILLIMETERS	
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.80 BSC	
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X44)	X1			0.55
Contact Pad Length (X44)	Y1			1.50
Distance Between Pads	G	0.25		

Notes:

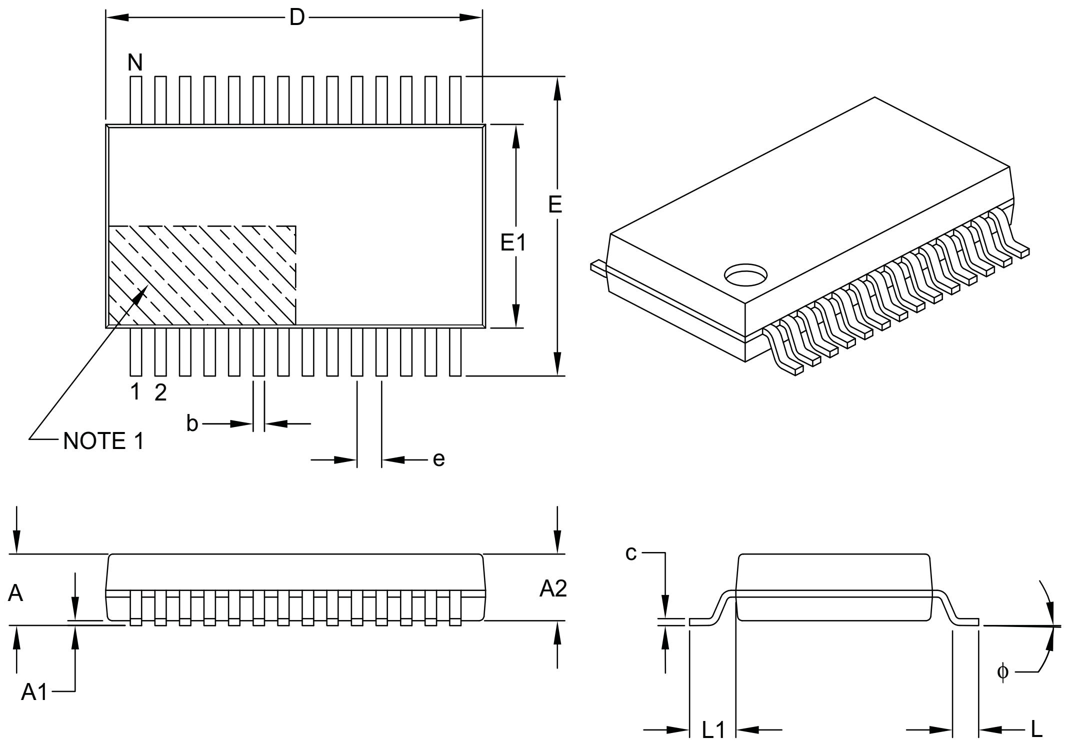
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2076A

28-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	9.90	10.20	10.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1		1.25 REF	
Lead Thickness	c	0.09	–	0.25
Foot Angle	ϕ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

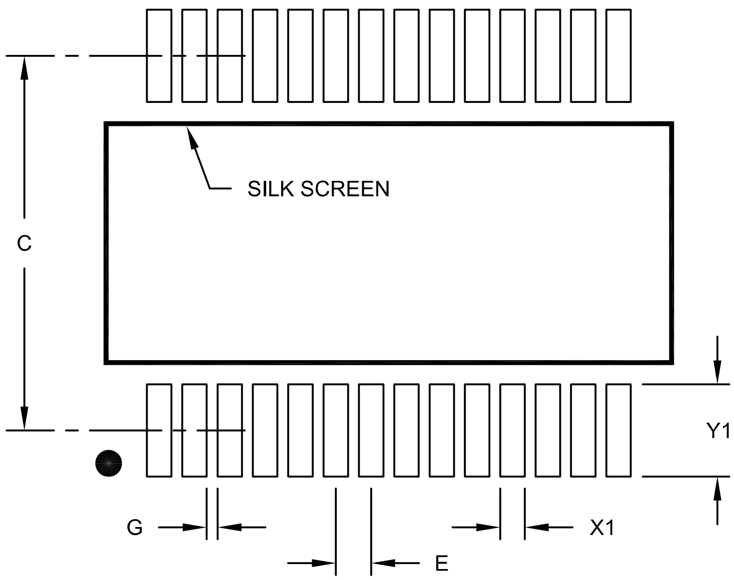
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B

28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

		Units	MILLIMETERS		
		Dimension Limits	MIN	NOM	MAX
Contact Pitch	E		0.65 BSC		
Contact Pad Spacing	C			7.20	
Contact Pad Width (X28)	X1				0.45
Contact Pad Length (X28)	Y1				1.75
Distance Between Pads	G		0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2073A

NOTES:

APPENDIX A: REVISION HISTORY

Revision A (August 2009)

Original data sheet for the PIC24FJ64GA104 family of devices.

Revision B (October 2009)

Corrected **Section 10.3 "Input Change Notification"** regarding the number of ICN inputs and the availability of pull-downs.

Updated **Section 10.4.2 "Available Peripherals"** by removing the Timer 1 clock input from Table 10-2.

Updated **Section 28.1 "DC Characteristics"** as follows:

- Added new specifications to Tables 29-4 and 29-5 for I_{DD} and I_{IDLE} at 0.5 MIPS operation.
- Updated Table 29-4 with revised maximum I_{DD} specifications for 1 MIP and 4 MIPS.
- Renumbered the parameters for the delta I_{PD} current (32 kHz, SOSCEL = `11`) from DC62*n* to DC63*n*.

Revision C (August 2010)

This revision includes the following updates:

Pin Diagrams

- Updated Pin 7 and Pin 14 in 28-Pin SPDIP, SOIC.
- Updated the device name, Pin13 and Pin 23, in 28-Pin QFN.

Removed IEC5, IFS5 and IPC21 rows from Table 4-5.

Updated CLKDIV bit details in Table 4-23.

Removed JTAG from Flash programming list in **Section 5.0 "Flash Program Memory"**.

Updated **Section 10.4.5 "Considerations for Peripheral Pin Selection"** as follows:

- Replaced the code in Example 10-2.
- Added the new code as Example 10-3.

Updated shaded note in **Section 20.0 "32-Bit Programmable Cyclic Redundancy Check (CRC) Generator"** and **Section 22.0 "Triple Comparator Module"**.

Updated **Section 28.1 "DC Characteristics"** as follows:

- Updated the device name in Table 28-1.
- Added the "125°C data" in Table 28-4,Table 28-5,Table 28-6 and Table 28-7.
- Updated Min and Typ columns of DC16 in Table 28-3.
- Added rows, AD08 and AD09, in Table 28-22.
- Added Figure 28-2.

Added the 28-pin SSOP package to **Section 29.0 "Packaging Information"**.

NOTES:

INDEX

V

W

NOTES:

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com, click on Customer Change Notification and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://support.microchip.com

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager   Total Pages Sent ________

RE: Reader Response

From: Name ____________________

Company ____________________

Address ____________________

City / State / ZIP / Country ____________________

Telephone: (_______) _________ - _________   FAX: (______) _________ - _________

Application (optional):

Would you like a reply?___Y ___N

Device: PIC24FJ64GA104 Family   Literature Number: DS39951C

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?

____

____

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PIC 24 FJ 64 GA1 04 T - I / PT - XXX

- **Microchip Trademark** — PIC
- **Architecture** — 24
- **Flash Memory Family** — FJ
- **Program Memory Size (KB)** — 64
- **Product Group** — GA1
- **Pin Count** — 04
- **Tape and Reel Flag (if applicable)** — T
- **Temperature Range** — I
- **Package** — PT
- **Pattern** — XXX

Architecture	24	= 16-bit modified Harvard without DSP
Flash Memory Family	FJ	= Flash program memory
Product Group	GA1	= General purpose microcontrollers
Pin Count	02 04	= 28-pin = 44-pin
Temperature Range	I E	= -40°C to +85°C (Industrial) = -40°C to +125°C (Extended)
Package	ML PT SO SP SS	= 28-lead (6x6 mm) or 44-lead (8x8 mm) QFN (Quad Flat) = 44-lead (10x10x1 mm) TQFP (Thin Quad Flatpack) = 28-lead (7.50 mm wide) SOIC (Small Outline) = 28-lead (300 mil) SPDIP (Skinny Plastic Dual In-Line) = 28-lead (530 mm) SSOP (Plastic Shrink Small)
Pattern	Three-digit QTP, SQTP, Code or Special Requirements (blank otherwise) ES	 = Engineering Sample

Examples:

a) PIC24FJ64GA104-I/PT:
PIC24F device with, 64-Kbyte program memory, 44-pin, Industrial temp., TQFP package.

b) PIC24FJ32GA102-I/ML:
PIC24F device with32-Kbyte program memory, 28-pin, Industrial temp.,QFN package.

查询PIC24FJ64GA104供应商

MICROCHIP

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-6578-300
Fax: 886-3-6578-370

Taiwan - Kaohsiung
Tel: 886-7-213-7830
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

07/15/10