



# **PIC24FJ64GA004 Family Data Sheet**

28/44-Pin General Purpose,  
16-Bit Flash Microcontrollers

---

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

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

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

---

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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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

**Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, rPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Octopus, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICKit, PICTail, PIC<sup>32</sup> logo, REAL ICE, rLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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

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

© 2010, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

ISBN: 978-1-60932-022-5

*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<sup>®</sup> MCUs and dsPIC<sup>®</sup> DSCs, KEELOQ<sup>®</sup> code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*

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



# MICROCHIP

# PIC24FJ64GA004 FAMILY

## 28/44-Pin General Purpose, 16-Bit Flash Microcontrollers

### High-Performance CPU:

- Modified Harvard Architecture
- Up to 16 MIPS Operation @ 32 MHz
- 8 MHz Internal Oscillator with 4x PLL Option and Multiple Divide Options
- 17-Bit by 17-Bit Single-Cycle Hardware Multiplier
- 32-Bit by 16-Bit Hardware Divider
- 16-Bit x 16-Bit Working Register Array
- C Compiler Optimized Instruction Set Architecture:
  - 76 base instructions
  - Flexible addressing modes
- 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
- 5.5V Tolerant Input (digital pins only)
- High-Current Sink/Source (18 mA/18 mA) on All I/O Pins
- Flash Program Memory:
  - 10,000 erase/write
  - 20-year data retention minimum
- Power Management modes:
  - Sleep, Idle, Doze and Alternate Clock modes
  - Operating current 650  $\mu$ A/MIPS typical at 2.0V
  - Sleep current 150 nA typical at 2.0V
- Fail-Safe Clock Monitor Operation:
  - Detects clock failure and switches to on-chip, low-power RC oscillator
- On-Chip, 2.5V Regulator with Tracking mode
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Flexible Watchdog Timer (WDT) with On-Chip, Low-Power RC Oscillator for Reliable Operation
- 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 Converter:
  - 500 ksp/s conversion rate
  - Conversion available during Sleep and Idle
- Dual Analog Comparators with Programmable Input/Output Configuration

### Peripheral Features:

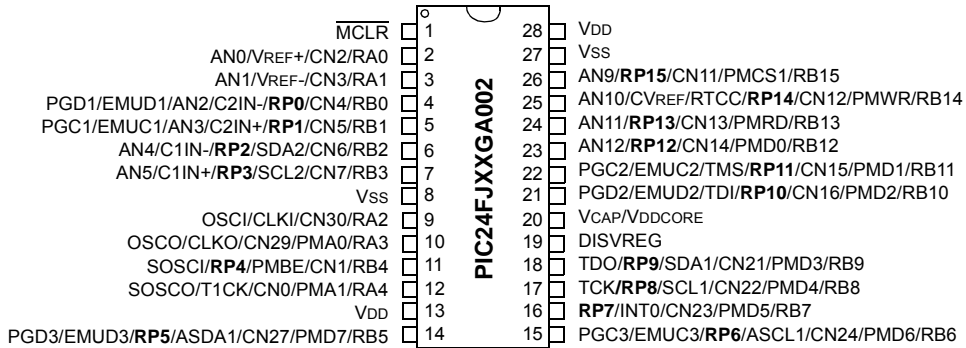
- Peripheral Pin Select:
  - Allows independent I/O mapping of many peripherals
  - Up to 26 available pins (44-pin devices)
  - Continuous hardware integrity checking and safety interlocks prevent unintentional configuration changes
- 8-Bit Parallel Master/Slave Port (PMP/PSP):
  - Up to 16-bit multiplexed addressing, with up to 11 dedicated address pins on 44-pin devices
  - Programmable polarity on control lines
- Hardware Real-Time Clock/Calendar (RTCC):
  - Provides clock, calendar and alarm functions
- Programmable Cyclic Redundancy Check (CRC)
- Two 3-Wire/4-Wire SPI modules (support 4 Frame modes) with 8-Level FIFO Buffer
- Two I<sup>2</sup>C™ modules support Multi-Master/Slave mode and 7-Bit/10-Bit Addressing
- Two UART modules:
  - Supports RS-485, RS-232, and LIN 1.2
  - On-chip hardware encoder/decoder for IrDA®
  - Auto-wake-up on Start bit
  - Auto-Baud Detect
  - 4-level deep FIFO buffer
- Five 16-Bit Timers/Counters with Programmable Prescaler
- Five 16-Bit Capture Inputs
- Five 16-Bit Compare/PWM Outputs
- Configurable Open-Drain Outputs on Digital I/O Pins
- Up to 4 External Interrupt Sources

| PIC24FJ Device | Pins | Program Memory (bytes) | SRAM (bytes) | Remappable Peripherals |               |               |                    |               |     | I <sup>2</sup> C™ | 10-Bit A/D (ch) | Comparators | PMP/PSP | JTAG |
|----------------|------|------------------------|--------------|------------------------|---------------|---------------|--------------------|---------------|-----|-------------------|-----------------|-------------|---------|------|
|                |      |                        |              | Remappable Pins        | Timers 16-Bit | Capture Input | Compare/PWM Output | UART w/ IrDA® | SPI |                   |                 |             |         |      |
| 16GA002        | 28   | 16K                    | 4K           | 16                     | 5             | 5             | 5                  | 2             | 2   | 2                 | 10              | 2           | Y       | Y    |
| 32GA002        | 28   | 32K                    | 8K           | 16                     | 5             | 5             | 5                  | 2             | 2   | 2                 | 10              | 2           | Y       | Y    |
| 48GA002        | 28   | 48K                    | 8K           | 16                     | 5             | 5             | 5                  | 2             | 2   | 2                 | 10              | 2           | Y       | Y    |
| 64GA002        | 28   | 64K                    | 8K           | 16                     | 5             | 5             | 5                  | 2             | 2   | 2                 | 10              | 2           | Y       | Y    |
| 16GA004        | 44   | 16K                    | 4K           | 26                     | 5             | 5             | 5                  | 2             | 2   | 2                 | 13              | 2           | Y       | Y    |
| 32GA004        | 44   | 32K                    | 8K           | 26                     | 5             | 5             | 5                  | 2             | 2   | 2                 | 13              | 2           | Y       | Y    |
| 48GA004        | 44   | 48K                    | 8K           | 26                     | 5             | 5             | 5                  | 2             | 2   | 2                 | 13              | 2           | Y       | Y    |
| 64GA004        | 44   | 64K                    | 8K           | 26                     | 5             | 5             | 5                  | 2             | 2   | 2                 | 13              | 2           | Y       | Y    |

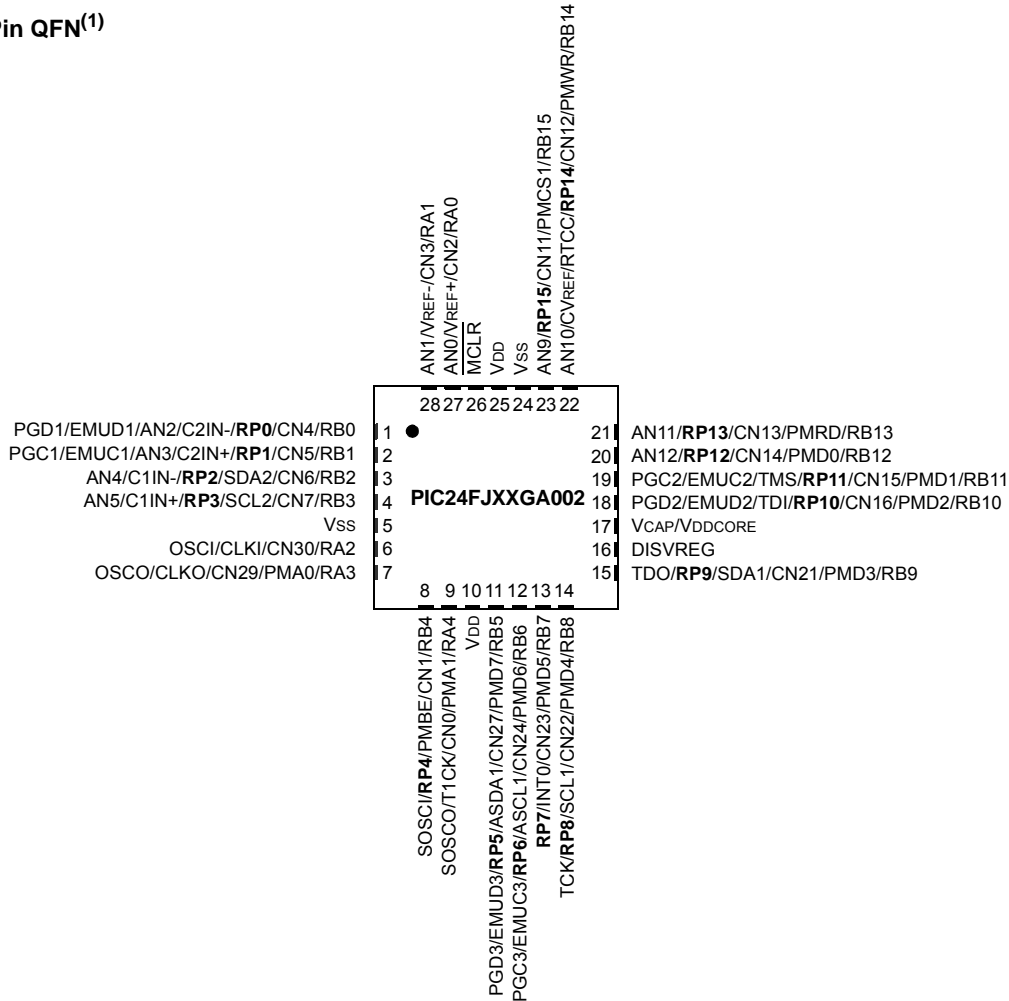
# PIC24FJ64GA004 FAMILY

## Pin Diagrams

### 28-Pin SPDIP, SSOP, SOIC



### 28-Pin QFN<sup>(1)</sup>



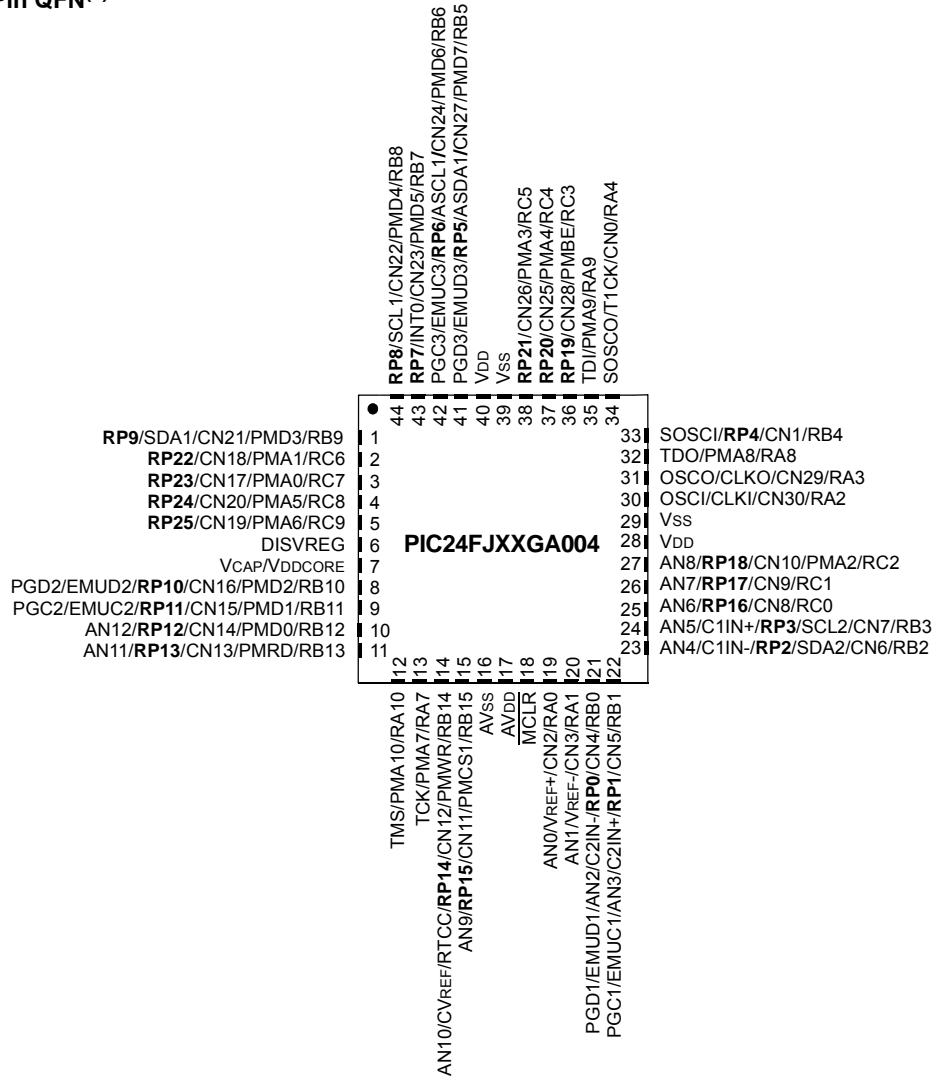
**Legend:** RPn represents remappable peripheral pins.

**Note 1:** Back pad on QFN devices should be connected to Vss.

# PIC24FJ64GA004 FAMILY

## Pin Diagrams (Continued)

44-Pin QFN<sup>(1)</sup>

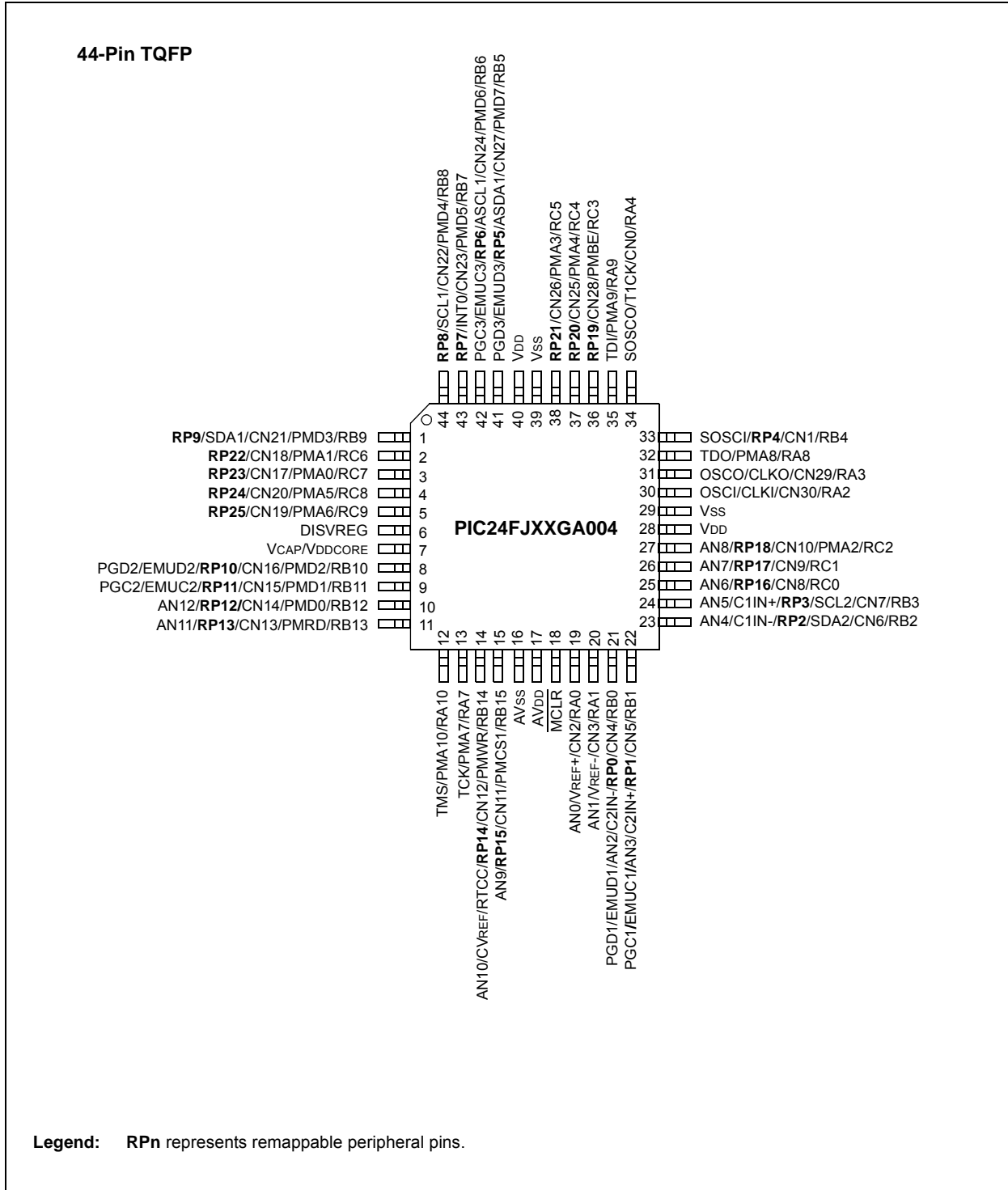


**Legend:** RPN represents remappable peripheral pins.

**Note 1:** Back pad on QFN devices should be connected to Vss.

# PIC24FJ64GA004 FAMILY

## Pin Diagrams (Continued)



# PIC24FJ64GA004 FAMILY

---

## Table of Contents

|      |  |     |
|------|--|-----|
| 1.0  | Device Overview .....  | 9   |
| 2.0  | Guidelines for Getting Started with 16-bit Microcontrollers .....            | 19  |
| 3.0  | CPU .....  | 25  |
| 4.0  | Memory Organization .....  | 31  |
| 5.0  | Flash Program Memory .....   | 49  |
| 6.0  | Resets .....   | 55  |
| 7.0  | Interrupt Controller .....   | 61  |
| 8.0  | Oscillator Configuration .....   | 95  |
| 9.0  | Power-Saving Features .....  | 103 |
| 10.0 | I/O Ports .....  | 105 |
| 11.0 | Timer1 .....   | 125 |
| 12.0 | Timer2/3 and Timer4/5 .....  | 127 |
| 13.0 | Input Capture .....  | 133 |
| 14.0 | Output Compare .....   | 135 |
| 15.0 | Serial Peripheral Interface (SPI) .....                                      | 141 |
| 16.0 | Inter-Integrated Circuit (I <sup>2</sup> C™) .....                           | 151 |
| 17.0 | Universal Asynchronous Receiver Transmitter (UART) .....                     | 159 |
| 18.0 | Parallel Master Port (PMP) .....   | 167 |
| 19.0 | Real-Time Clock And Calendar (RTCC) .....                                    | 177 |
| 20.0 | Programmable Cyclic Redundancy Check (CRC) Generator .....                   | 187 |
| 21.0 | 10-Bit High-Speed A/D Converter .....  | 191 |
| 22.0 | Comparator Module .....  | 201 |
| 23.0 | Comparator Voltage Reference .....   | 205 |
| 24.0 | Special Features .....   | 207 |
| 25.0 | Development Support .....  | 217 |
| 26.0 | Instruction Set Summary .....  | 221 |
| 27.0 | Electrical Characteristics .....   | 229 |
| 28.0 | Packaging Information .....  | 247 |
|      | Appendix A: Revision History .....   | 259 |
|      | Appendix B: Additional Guidance for PIC24FJ64GA004 Family Applications ..... | 260 |
|      | Index .....  | 261 |
|      | The Microchip Web Site .....   | 265 |
|      | Customer Change Notification Service .....                                   | 265 |
|      | Customer Support .....   | 265 |
|      | Reader Response .....  | 266 |
|      | Product Identification System .....  | 267 |

# PIC24FJ64GA004 FAMILY

---

## 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](mailto: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](http://www.microchip.com) to receive the most current information on all of our products.



# PIC24FJ64GA004 FAMILY

---

## 1.0 DEVICE OVERVIEW

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

- PIC24FJ16GA002
- PIC24FJ32GA002
- PIC24FJ48GA002
- PIC24FJ64GA002
- PIC24FJ16GA004
- PIC24FJ32GA004
- PIC24FJ48GA004
- PIC24FJ64GA004

This family introduces a new line of Microchip devices: a 16-bit microcontroller family with a broad peripheral feature set and enhanced computational performance. The PIC24FJ64GA004 family offers a new migration option for those high-performance applications which may be outgrowing their 8-bit platforms, but don't require the numerical processing 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 PIC24FJ64GA004 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 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:** The microcontroller can suspend all operations, or selectively shut down its core while leaving its peripherals active, with a single instruction in software.

#### 1.1.3 OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC24FJ64GA004 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 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.

# PIC24FJ64GA004 FAMILY

---

## 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 to the next larger device. This is true when moving between devices with the same pin count, or even jumping from 28-pin to 44-pin devices.

The PIC24F family is pin-compatible with devices in the dsPIC33 family, and shares some compatibility with the pinout schema for PIC18 and dsPIC30. 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

- **Communications:** The PIC24FJ64GA004 family incorporates a range of serial communication peripherals to handle a range of application requirements. There are two independent I<sup>2</sup>C modules that support both Master and Slave modes of operation. Devices also have, through the peripheral pin select feature, two independent UARTs with built-in IrDA encoder/decoders and two SPI modules.
- **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.
- **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 16 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 use of the core application.
- **10-Bit A/D Converter:** This module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated without waiting for a sampling period, as well as faster sampling speeds.

## 1.3 Details on Individual Family Members

Devices in the PIC24FJ64GA004 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 two ways:

1. Flash program memory (64 Kbytes for PIC24FJ64GA devices, 48 Kbytes for PIC24FJ48GA devices, 32 Kbytes for PIC24FJ32GA devices and 16 Kbytes for PIC24FJ16GA devices).
2. Internal SRAM memory (4k for PIC24FJ16GA devices, 8k for all other devices in the family).
3. Available I/O pins and ports (21 pins on 2 ports for 28-pin devices and 35 pins on 3 ports for 44-pin devices).

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 PIC24FJ64GA004 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 the data sheet. Multiplexed features are sorted by the priority given to a feature, with the highest priority peripheral being listed first.

# PIC24FJ64GA004 FAMILY

**TABLE 1-1: DEVICE FEATURES FOR THE PIC24FJ64GA004 FAMILY**

| Features  | 16GA002   | 32GA002 | 48GA002 | 64GA002 | 16GA004         | 32GA004 | 48GA004 | 64GA004 |
|---|---|---------|---------|---------|-----------------|---------|---------|---------|
| Operating Frequency                                 | DC – 32 MHz   |         |         |         |                 |         |         |         |
| Program Memory (bytes)                              | 16K   | 32K     | 48K     | 64K     | 16K             | 32K     | 48K     | 64K     |
| Program Memory (instructions)                       | 5,504   | 11,008  | 16,512  | 22,016  | 5,504           | 11,008  | 16,512  | 22,016  |
| Data Memory (bytes)                                 | 4096  | 8192    |         |         | 4096            | 8192    |         |         |
| Interrupt Sources<br>(soft vectors/NMI traps)       | 43<br>(39/4)  |         |         |         |                 |         |         |         |
| I/O Ports   | Ports A, B  |         |         |         | Ports A, B, C   |         |         |         |
| Total I/O Pins                                      | 21  |         |         |         | 35              |         |         |         |
| Timers:   |   |         |         |         |                 |         |         |         |
| Total Number (16-bit)                               | 5 <sup>(1)</sup>  |         |         |         |                 |         |         |         |
| 32-Bit (from paired 16-bit timers)                  | 2   |         |         |         |                 |         |         |         |
| Input Capture Channels                              | 5 <sup>(1)</sup>  |         |         |         |                 |         |         |         |
| Output Compare/PWM Channels                         | 5 <sup>(1)</sup>  |         |         |         |                 |         |         |         |
| Input Change Notification Interrupt                 | 21  |         |         |         | 30              |         |         |         |
| Serial Communications:                              |   |         |         |         |                 |         |         |         |
| UART  | 2 <sup>(1)</sup>  |         |         |         |                 |         |         |         |
| SPI (3-wire/4-wire)                                 | 2 <sup>(1)</sup>  |         |         |         |                 |         |         |         |
| I <sup>2</sup> C™                                   | 2   |         |         |         |                 |         |         |         |
| Parallel Communications (PMP/PSP)                   | Yes   |         |         |         |                 |         |         |         |
| JTAG Boundary Scan                                  | Yes   |         |         |         |                 |         |         |         |
| 10-Bit Analog-to-Digital Module<br>(input channels) | 10  |         |         |         | 13              |         |         |         |
| Analog Comparators                                  | 2   |         |         |         |                 |         |         |         |
| Remappable Pins                                     | 16  |         |         |         | 26              |         |         |         |
| Resets (and delays)                                 | POR, BOR, RESET Instruction, MCLR, WDT, Illegal Opcode, REPEAT Instruction, Hardware Traps, Configuration Word Mismatch (PWRT, OST, PLL Lock) |         |         |         |                 |         |         |         |
| Instruction Set                                     | 76 Base Instructions, Multiple Addressing Mode Variations   |         |         |         |                 |         |         |         |
| Packages  | 28-Pin SPDIP/SSOP/SOIC/QFN  |         |         |         | 44-Pin QFN/TQFP |         |         |         |

**Note 1:** Peripherals are accessible through remappable pins.



# PIC24FJ64GA004 FAMILY

**TABLE 1-2: PIC24FJ64GA004 FAMILY PINOUT DESCRIPTIONS**

| Function | Pin Number             |            |                 | I/O | Input Buffer     | Description  |
|----------|------------------------|------------|-----------------|-----|------------------|--|
|          | 28-Pin SPDIP/SSOP/SOIC | 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 <sup>2</sup> C | Alternate I2C1 Synchronous Serial Clock Input/Output. <sup>(1)</sup> |
| ASDA1    | 14                     | 11         | 41              | I/O | I <sup>2</sup> C | Alternate I2C2 Synchronous Serial Clock Input/Output. <sup>(1)</sup> |
| AVDD     | —                      | —          | 17              | P   | —                | Positive Supply for Analog Modules.                                  |
| AVSS     | —                      | —          | 16              | P   | —                | Ground Reference for Analog Modules.                                 |
| C1IN-    | 6                      | 3          | 23              | I   | ANA              | Comparator 1 Negative Input.   |
| C1IN+    | 7                      | 4          | 24              | I   | ANA              | Comparator 1 Positive Input.   |
| C2IN-    | 4                      | 1          | 21              | I   | ANA              | Comparator 2 Negative Input.   |
| C2IN+    | 5                      | 2          | 22              | I   | ANA              | Comparator 2 Positive Input.   |
| CLKI     | 9                      | 6          | 30              | I   | ANA              | Main Clock Input Connection.   |
| CLKO     | 10                     | 7          | 31              | O   | —                | System Clock Output.   |

**Legend:** TTL = TTL input buffer  
ANA = Analog level input/output

ST = Schmitt Trigger input buffer  
I<sup>2</sup>C™ = I<sup>2</sup>C/SMBus input buffer

**Note 1:** Alternative multiplexing when the I2C1SEL Configuration bit is cleared.

# PIC24FJ64GA004 FAMILY

**TABLE 1-2: PIC24FJ64GA004 FAMILY PINOUT DESCRIPTIONS (CONTINUED)**

| Function | Pin Number             |            |                 | I/O | Input Buffer | Description   |
|----------|------------------------|------------|-----------------|-----|--------------|---|
|          | 28-Pin SPDIP/SSOP/SOIC | 28-Pin QFN | 44-Pin QFN/TQFP |     |              |   |
| 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           |   |
| CVREF    | 25                     | 22         | 14              | O   | ANA          | Comparator Voltage Reference Output.  |
| DISVREG  | 19                     | 16         | 6               | I   | ST           | Voltage Regulator Disable.  |
| EMUC1    | 5                      | 2          | 21              | I/O | ST           | In-Circuit Emulator Clock Input/Output.                                       |
| EMUD1    | 4                      | 1          | 22              | I/O | ST           | In-Circuit Emulator Data Input/Output.  |
| EMUC2    | 22                     | 19         | 9               | I/O | ST           | In-Circuit Emulator Clock Input/Output.                                       |
| EMUD2    | 21                     | 18         | 8               | I/O | ST           | In-Circuit Emulator Data Input/Output.  |
| EMUC3    | 15                     | 12         | 42              | I/O | ST           | In-Circuit Emulator Clock Input/Output.                                       |
| EMUD3    | 14                     | 11         | 41              | I/O | ST           | In-Circuit Emulator Data Input/Output.  |
| INT0     | 16                     | 13         | 43              | I   | ST           | External Interrupt Input.   |
| MCLR     | 1                      | 26         | 18              | I   | ST           | Master Clear (device Reset) Input. This line is brought low to cause a Reset. |

**Legend:** TTL = TTL input buffer  
ANA = Analog level input/output

ST = Schmitt Trigger input buffer  
I<sup>2</sup>C™ = I<sup>2</sup>C/SMBus input buffer

**Note 1:** Alternative multiplexing when the I2C1SEL Configuration bit is cleared.

# PIC24FJ64GA004 FAMILY

**TABLE 1-2: PIC24FJ64GA004 FAMILY PINOUT DESCRIPTIONS (CONTINUED)**

| Function | Pin Number             |            |                 | I/O | Input Buffer | Description   |
|----------|------------------------|------------|-----------------|-----|--------------|---|
|          | 28-Pin SPDIP/SSOP/SOIC | 28-Pin QFN | 44-Pin QFN/TQFP |     |              |   |
| OSCI     | 9                      | 6          | 30              | I   | ANA          | Main Oscillator Input Connection.   |
| OSCO     | 10                     | 7          | 31              | O   | ANA          | Main Oscillator Output Connection.  |
| PGC1     | 5                      | 2          | 22              | I/O | ST           | In-Circuit Debugger and ICSP™ Programming Clock   |
| PGD1     | 4                      | 1          | 21              | I/O | ST           | In-Circuit Debugger and ICSP Programming Data.  |
| PGC2     | 22                     | 19         | 9               | I/O | ST           | In-Circuit Debugger and ICSP Programming Clock.   |
| PGD2     | 21                     | 18         | 8               | I/O | ST           | In-Circuit Debugger and ICSP Programming Data.  |
| PGC3     | 14                     | 12         | 42              | I/O | ST           | In-Circuit Debugger and ICSP Programming Clock.   |
| PGD3     | 15                     | 11         | 41              | I/O | ST           | In-Circuit Debugger and ICSP Programming Data.  |
| PMA0     | 10                     | 7          | 3               | I/O | ST/TTL       | Parallel Master Port Address Bit 0 Input (Buffered Slave modes) and Output (Master modes).        |
| PMA1     | 12                     | 9          | 2               | I/O | ST/TTL       | 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   | —            |   |
| PMA11    | —                      | —          | —               | O   | —            |   |
| PMA12    | —                      | —          | —               | O   | —            |   |
| PMA13    | —                      | —          | —               | O   | —            |   |
| PMBE     | 11                     | 8          | 36              | O   | —            | Parallel Master Port Byte Enable Strobe.  |
| PMCS1    | 26                     | 23         | 15              | O   | —            | Parallel Master Port Chip Select 1 Strobe/Address Bit 14.   |
| 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<sup>2</sup>C™ = I<sup>2</sup>C/SMBus input buffer

**Note 1:** Alternative multiplexing when the I2C1SEL Configuration bit is cleared.









# PIC24FJ64GA004 FAMILY

## 2.0 GUIDELINES FOR GETTING STARTED WITH 16-BIT MICROCONTROLLERS

### 2.1 Basic Connection Requirements

Getting started with the PIC24FJ64GA004 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”**)

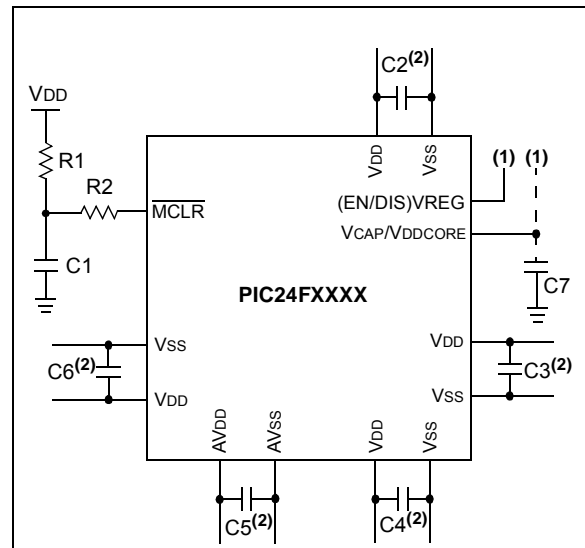
Additionally, the following pins may be required:

- 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  $\mu$ F, 20V ceramic

C7: 10  $\mu$ F, 6.3V or greater, tantalum or ceramic

R1: 10 k $\Omega$

R2: 100 $\Omega$  to 470 $\Omega$

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

**Note 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.

# PIC24FJ64GA004 FAMILY

## 2.2 Power Supply Pins

### 2.2.1 DECOUPLING CAPACITORS

The use of decoupling capacitors on every pair of power supply pins, such as VDD, VSS, AVDD and AVSS is required.

Consider the following criteria when using decoupling capacitors:

- **Value and type of capacitor:** A 0.1  $\mu\text{F}$  (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  $\mu\text{F}$  to 0.001  $\mu\text{F}$ . 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  $\mu\text{F}$  in parallel with 0.001  $\mu\text{F}$ ).
- **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  $\mu\text{F}$  to 47  $\mu\text{F}$ .

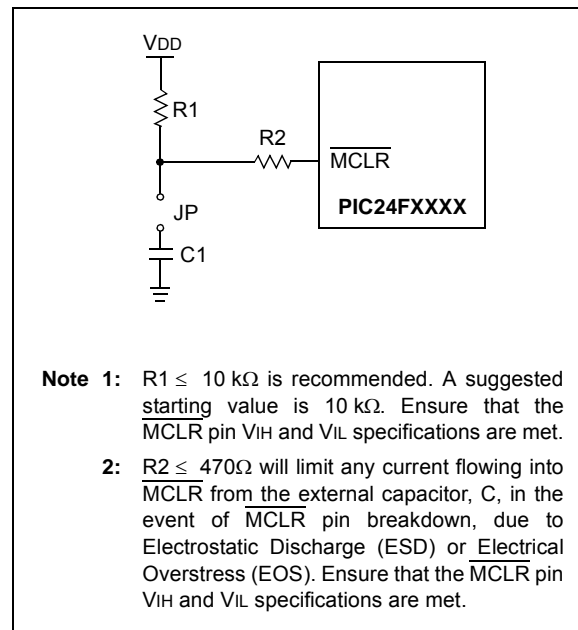
## 2.3 Master Clear ( $\overline{\text{MCLR}}$ ) Pin

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 VDD 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**



# PIC24FJ64GA004 FAMILY

## 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

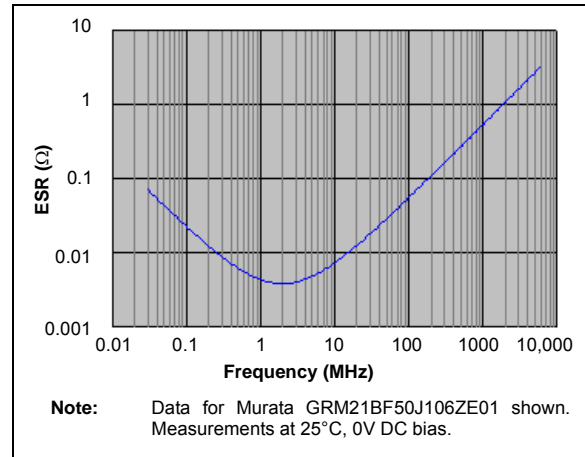
Refer to **Section 24.2 “On-Chip Voltage Regulator”** for details on connecting and using the on-chip regulator.

When the regulator is enabled, a low-ESR (<5 $\Omega$ ) 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  $\mu$ F connected to ground. The type can be ceramic or tantalum. A suitable example is the Murata GRM21BF50J106ZE01 (10  $\mu$ F, 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 27.0 “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 27.0 “Electrical Characteristics”** for information on VDD and VDDCORE.

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



## 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 100 $\Omega$ .

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 25.0 “Development Support”**.

# PIC24FJ64GA004 FAMILY

## 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.

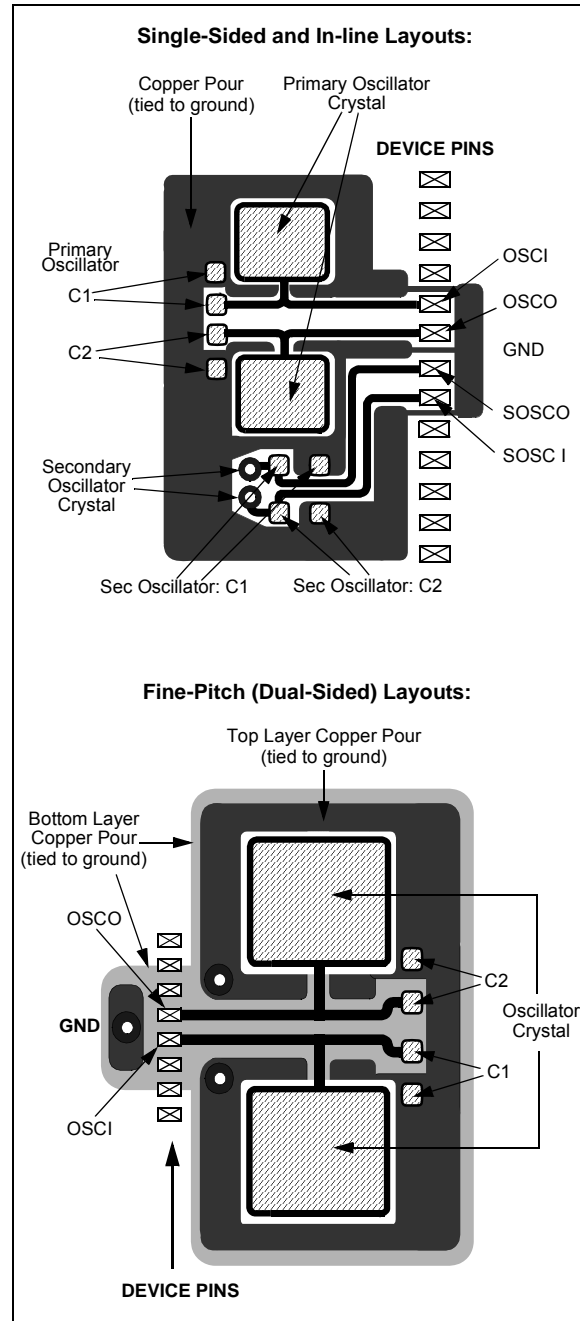
Layout suggestions are shown in Figure 2-4. In-line packages may be handled with a single-sided layout that completely encompasses the oscillator pins. With fine-pitch packages, it is not always possible to completely surround the pins and components. A suitable solution is to tie the broken guard sections 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](http://www.microchip.com)):

- AN826, “Crystal Oscillator Basics and Crystal Selection for rPIC™ 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 k $\Omega$  to 10 k $\Omega$  resistor to Vss on unused pins and drive the output to logic low.

# PIC24FJ64GA004 FAMILY

---

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.

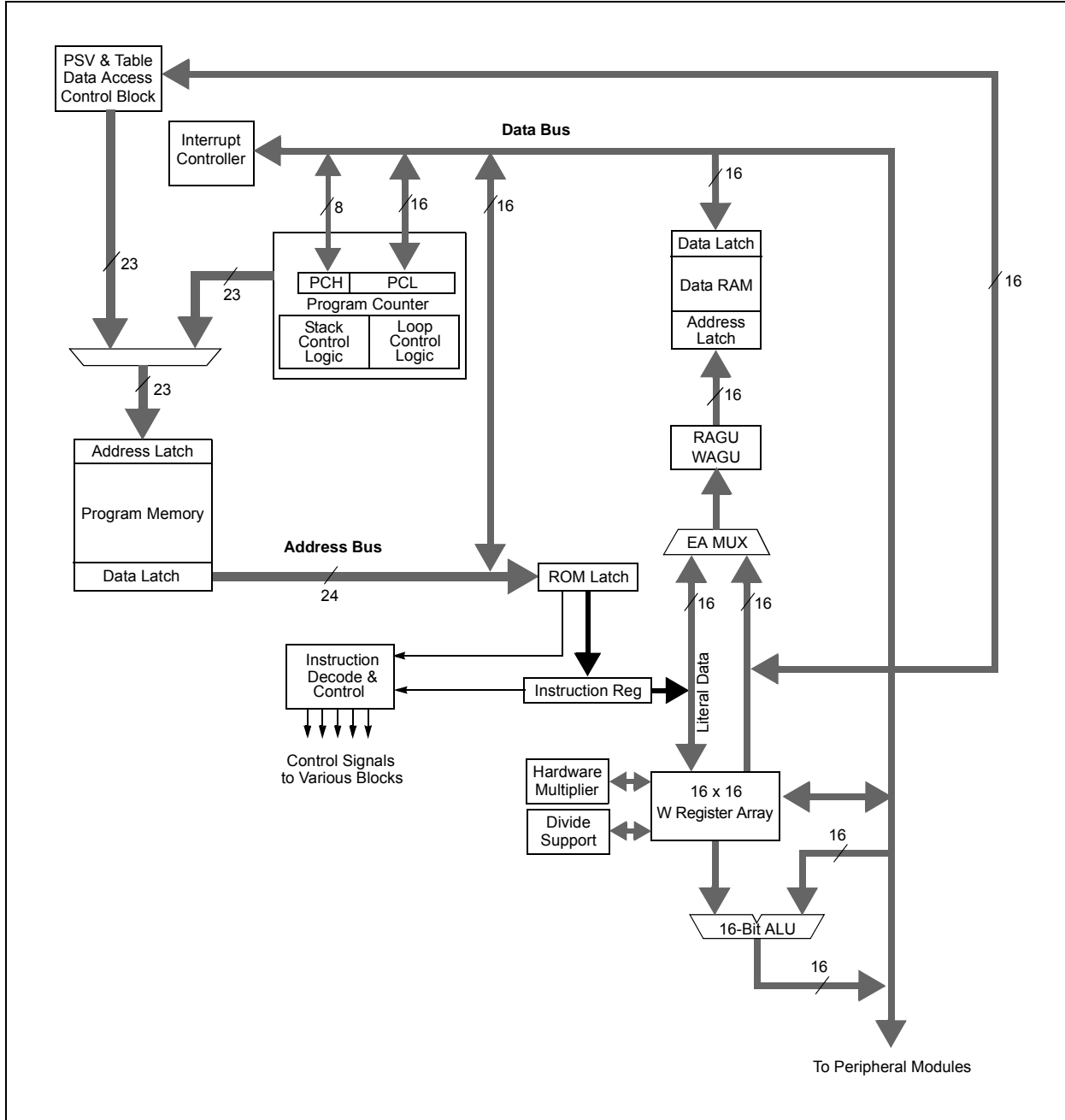
A block diagram of the CPU is shown in Figure 3-1.

### 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.

# PIC24FJ64GA004 FAMILY

FIGURE 3-1: PIC24F CPU CORE BLOCK DIAGRAM

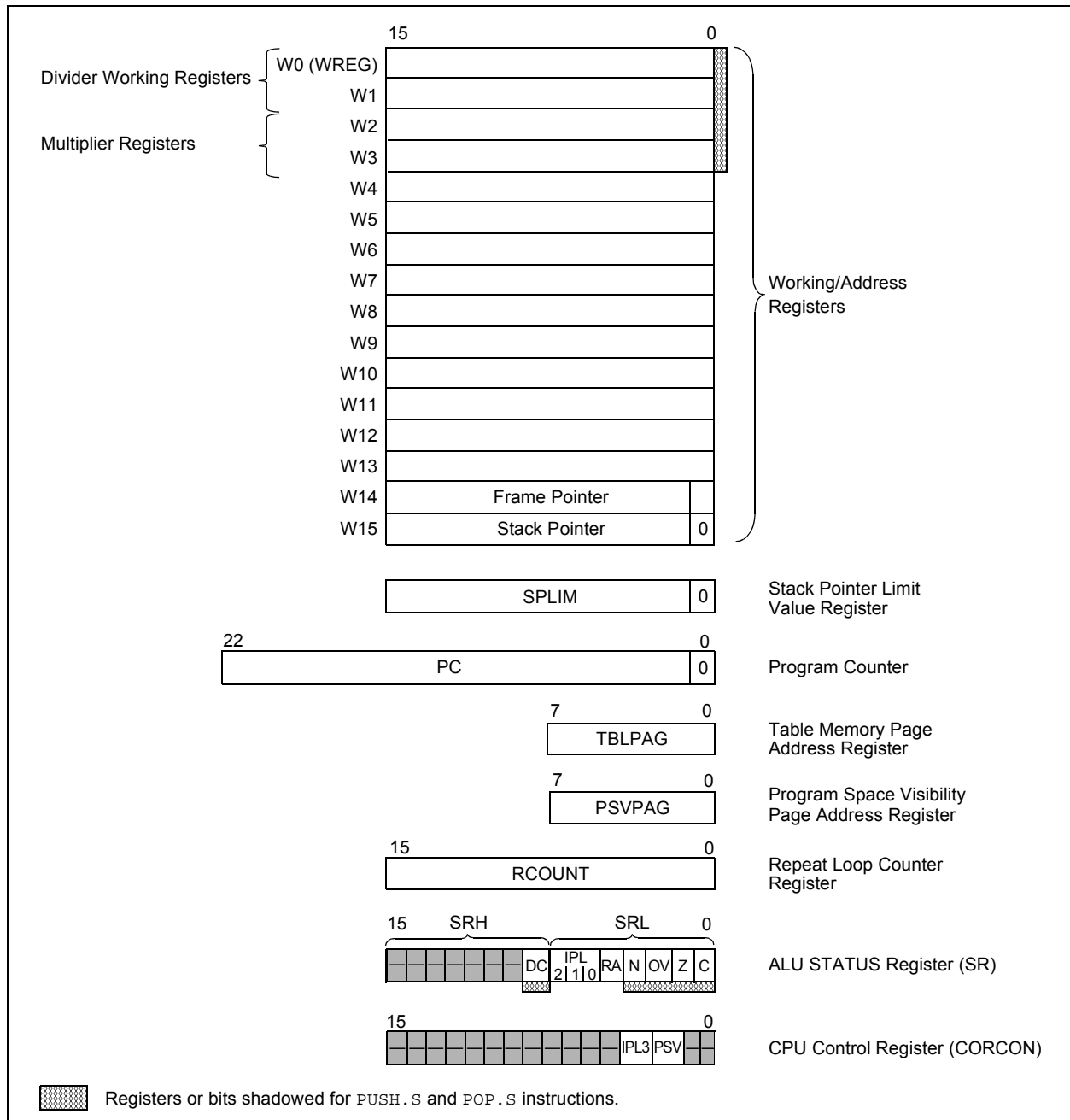


# PIC24FJ64GA004 FAMILY

**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**



# PIC24FJ64GA004 FAMILY

## 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 <sup>(1)</sup> | R/W-0 <sup>(1)</sup> | R/W-0 <sup>(1)</sup> | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| IPL2 <sup>(2)</sup>  | IPL1 <sup>(2)</sup>  | IPL0 <sup>(2)</sup>  | 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      **IPL2:IPL0:** CPU Interrupt Priority Level Status bits<sup>(1,2)</sup>
  - 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.  
**Note 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.

# PIC24FJ64GA004 FAMILY

## 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 <sup>(1)</sup> | PSV   | —   | —     |
| 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 15-4     **Unimplemented:** Read as '0'
- bit 3       **IPL3:** CPU Interrupt Priority Level Status bit<sup>(1)</sup>
  - 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

# PIC24FJ64GA004 FAMILY

---

## 3.3.2 DIVIDER

The divide block supports 32-bit/16-bit and 16-bit/16-bit 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.    |

# PIC24FJ64GA004 FAMILY

## 4.0 MEMORY ORGANIZATION

As Harvard architecture devices, PIC24F micro-controllers 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

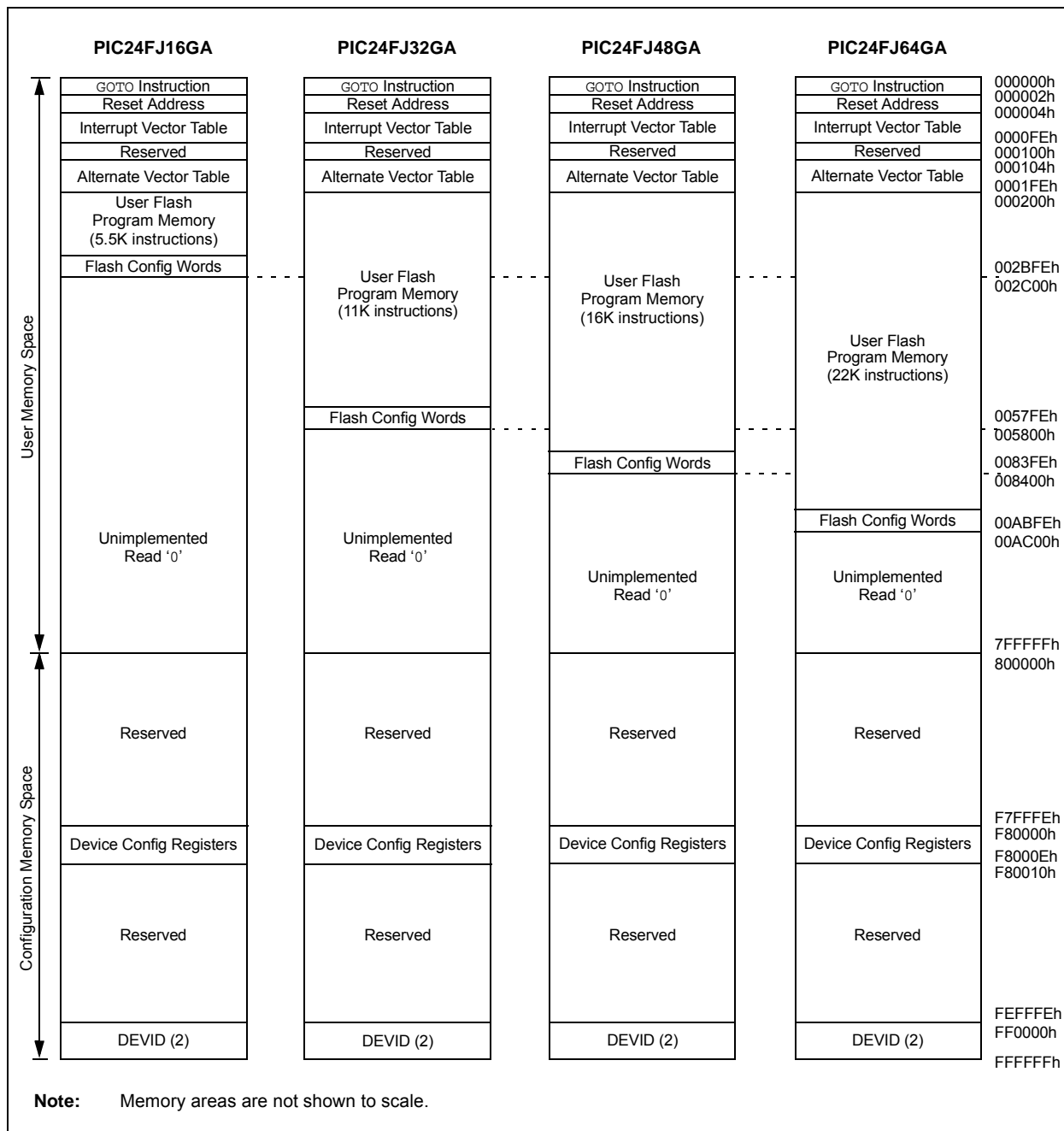
The program address memory space of the PIC24FJ64GA004 family devices is 4M instructions. The space is addressable by a 24-bit value derived

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 7FFFFFFh). 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 PIC24FJ64GA004 family of devices are shown in Figure 4-1.

**FIGURE 4-1: PROGRAM SPACE MEMORY MAP FOR PIC24FJ64GA004 FAMILY DEVICES**



# PIC24FJ64GA004 FAMILY

## 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

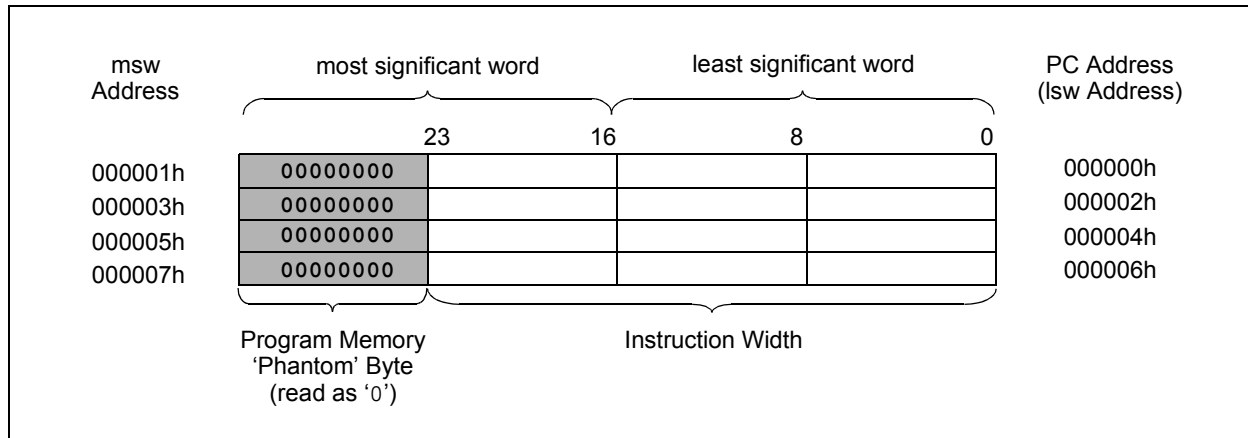
In PIC24FJ64GA004 family devices, the top two words of on-chip program memory are reserved for configuration information. On device Reset, the configuration information is copied into the appropriate Configuration registers. The addresses of the Flash Configuration Word for devices in the PIC24FJ64GA004 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 24.1 “Configuration Bits”**.

**TABLE 4-1: FLASH CONFIGURATION WORDS FOR PIC24FJ64GA004 FAMILY DEVICES**

| Device      | Program Memory (K words) | Configuration Word Addresses |
|-------------|--------------------------|------------------------------|
| PIC24FJ16GA | 5.5                      | 002BFCh:<br>002BFEh          |
| PIC24FJ32GA | 11                       | 0057FCh:<br>0057FEh          |
| PIC24FJ48GA | 16                       | 0083FCh:<br>0083FEh          |
| PIC24FJ64GA | 22                       | 00ABFCh:<br>00ABFEh          |

**FIGURE 4-2: PROGRAM MEMORY ORGANIZATION**





# PIC24FJ64GA004 FAMILY

## 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.

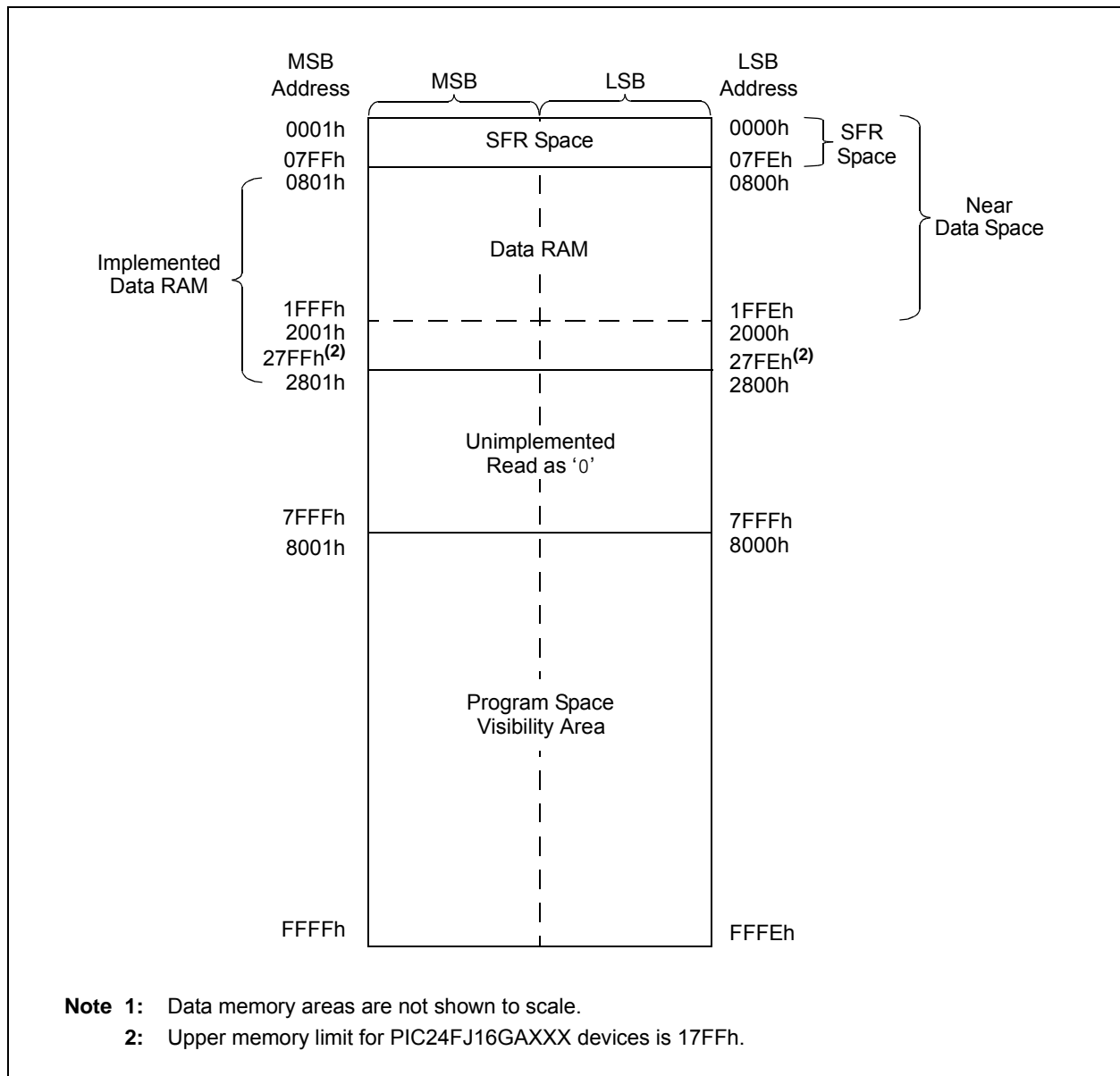
All Effective Addresses (EAs) in the data memory space are 16 bits wide and point to bytes within the data space. This gives a data space address range of 64 Kbytes or 32K words. The lower half of the data memory space (that is, when  $EA<15> = 0$ ) is used for implemented memory addresses, while the 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”).

PIC24FJ64GA family devices implement a total of 8 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 of each word have even addresses, while the Most Significant Bytes have odd addresses.

FIGURE 4-3: DATA SPACE MEMORY MAP FOR PIC24FJ64GA004 FAMILY DEVICES<sup>(1)</sup>



# PIC24FJ64GA004 FAMILY

## 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 (EA) 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 instruction (SE) 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 addressable indirectly. 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 4-24.

**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 <sup>2</sup> C™ | UART     | SPI     |         | —          | —    | I/O  |      |
| 300h              | A/D               |          | —       | —       | —          | —    | —    | —    |
| 400h              | —                 | —        | —       | —       | —          | —    | —    | —    |
| 500h              | —                 | —        | —       | —       | —          | —    | —    | —    |
| 600h              | PMP               | RTC/Comp | CRC     | —       | PPS        |      |      |      |
| 700h              | —                 | —        | System  | 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 Byte 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 <sup>(1)</sup>  | CN9IE <sup>(1)</sup>   | CN8IE <sup>(1)</sup>  | CN7IE   | CN6IE   | CN5IE   | CN4IE                  | CN3IE                  | CN2IE                  | CN1IE                  | CN0IE   | 0000       |
| CNEN2     | 0062 | —       | CN30IE  | CN29IE  | CN28IE <sup>(1)</sup>  | CN27IE  | CN26IE <sup>(1)</sup>  | CN25IE <sup>(1)</sup>  | CN24IE                | CN23IE  | CN22IE  | CN21IE  | CN20IE <sup>(1)</sup>  | CN19IE <sup>(1)</sup>  | CN18IE <sup>(1)</sup>  | CN17IE <sup>(1)</sup>  | CN16IE  | 0000       |
| CNPU1     | 0068 | CN15PUE | CN14PUE | CN13PUE | CN12PUE                | CN11PUE | CN10PUE <sup>(1)</sup> | CN9PUE <sup>(1)</sup>  | CN8PUE <sup>(1)</sup> | CN7PUE  | CN6PUE  | CN5PUE  | CN4PUE                 | CN3PUE                 | CN2PUE                 | CN1PUE                 | CN0PUE  | 0000       |
| CNPU2     | 006A | —       | CN30PUE | CN29PUE | CN28PUE <sup>(1)</sup> | CN27PUE | CN26PUE <sup>(1)</sup> | CN25PUE <sup>(1)</sup> | CN24PUE               | CN23PUE | CN22PUE | CN21PUE | CN20PUE <sup>(1)</sup> | CN19PUE <sup>(1)</sup> | CN18PUE <sup>(1)</sup> | CN17PUE <sup>(1)</sup> | CN16PUE | 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 4-5: 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 | —      | —       | —       | —       | —      | —       | —       | 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 | —      | —       | —       | —       | —      | —       | —       | 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  | —       | —       | —       | —       | 4444       |
| IPC2      | 00A8 | —      | U1RXIP2 | U1RXIP1 | U1RXIP0 | —      | SPI1IP2 | SPI1IP1 | SPI1IP0 | —     | SPF1IP2 | SPF1IP1 | SPF1IP0 | —       | T3IP2   | T3IP1   | T3IP0   | 4444       |
| IPC3      | 00AA | —      | —       | —       | —       | —      | —       | —       | —       | —     | AD1IP2  | AD1IP1  | AD1IP0  | —       | U1TXIP2 | U1TXIP1 | U1TXIP0 | 4444       |
| IPC4      | 00AC | —      | CNIP2   | CNIP1   | CNIP0   | —      | CMIP2   | CMIP1   | CMIP0   | —     | MI2C1P2 | MI2C1P1 | MI2C1P0 | —       | SI2C1P2 | SI2C1P1 | SI2C1P0 | 4444       |
| IPC5      | 00AE | —      | —       | —       | —       | —      | —       | —       | —       | —     | —       | —       | —       | INT1IP2 | INT1IP1 | INT1IP0 | 4444    |            |
| IPC6      | 00B0 | —      | T4IP2   | T4IP1   | T4IP0   | —      | OC4IP2  | OC4IP1  | OC4IP0  | —     | OC3IP2  | OC3IP1  | OC3IP0  | —       | —       | —       | —       | 4444       |
| IPC7      | 00B2 | —      | U2TXIP2 | U2TXIP1 | U2TXIP0 | —      | U2RXIP2 | U2RXIP1 | U2RXIP0 | —     | INT2IP2 | INT2IP1 | INT2IP0 | —       | T5IP2   | T5IP1   | T5IP0   | 4444       |
| IPC8      | 00B4 | —      | —       | —       | —       | —      | —       | —       | —       | —     | SPI2IP2 | SPI2IP1 | SPI2IP0 | —       | SPF2IP2 | SPF2IP1 | SPF2IP0 | 4444       |
| IPC9      | 00B6 | —      | IC5IP2  | IC5IP1  | IC5IP0  | —      | IC4IP2  | IC4IP1  | IC4IP0  | —     | IC3IP2  | IC3IP1  | IC3IP0  | —       | —       | —       | —       | 4444       |
| IPC10     | 00B8 | —      | —       | —       | —       | —      | —       | —       | —       | —     | OC5IP2  | OC5IP1  | OC5IP0  | —       | —       | —       | —       | 4444       |
| IPC11     | 00BA | —      | —       | —       | —       | —      | —       | —       | —       | —     | PMPIP2  | PMPIP1  | PMPIP0  | —       | —       | —       | —       | 4444       |
| IPC12     | 00BC | —      | —       | —       | —       | —      | MI2C2P2 | MI2C2P1 | MI2C2P0 | —     | SI2C2P2 | SI2C2P1 | SI2C2P0 | —       | —       | —       | —       | 4444       |
| IPC15     | 00C2 | —      | —       | —       | —       | —      | RTCIP2  | RTCIP1  | RTCIP0  | —     | —       | —       | —       | —       | —       | —       | —       | 4444       |
| IPC16     | 00C4 | —      | CRCIP2  | CRCIP1  | CRCIP0  | —      | U2ERIP2 | U2ERIP1 | U2ERIP0 | —     | U1ERIP2 | U1ERIP1 | U1ERIP0 | —       | —       | —       | —       | 4444       |
| IPC18     | 00C8 | —      | —       | —       | —       | —      | —       | —       | —       | —     | —       | —       | —       | —       | LVDIP2  | LVDIP1  | LVDIP0  | 4444       |

**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 |
|-----------|------|--------------------------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------------|
| IC1BUF    | 0140 | Input 1 Capture Register |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       | FFFF       |
| IC1CON    | 0142 | —                        | —      | ICSIDL | —      | —      | —      | —     | —     | ICTMR | IC1   | IC10  | ICOV  | ICBNE | ICM2  | ICM1  | ICM0  | 0000       |
| IC2BUF    | 0144 | Input 2 Capture Register |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       | FFFF       |
| IC2CON    | 0146 | —                        | —      | ICSIDL | —      | —      | —      | —     | —     | ICTMR | IC1   | IC10  | ICOV  | ICBNE | ICM2  | ICM1  | ICM0  | 0000       |
| IC3BUF    | 0148 | Input 3 Capture Register |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       | FFFF       |
| IC3CON    | 014A | —                        | —      | ICSIDL | —      | —      | —      | —     | —     | ICTMR | IC1   | IC10  | ICOV  | ICBNE | ICM2  | ICM1  | ICM0  | 0000       |
| IC4BUF    | 014C | Input 4 Capture Register |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       | FFFF       |
| IC4CON    | 014E | —                        | —      | ICSIDL | —      | —      | —      | —     | —     | ICTMR | IC1   | IC10  | ICOV  | ICBNE | ICM2  | ICM1  | ICM0  | 0000       |
| IC5BUF    | 0150 | Input 5 Capture Register |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       | FFFF       |
| IC5CON    | 0152 | —                        | —      | ICSIDL | —      | —      | —      | —     | —     | ICTMR | IC1   | IC10  | ICOV  | ICBNE | ICM2  | ICM1  | ICM0  | 0000       |

**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 |
|-----------|------|-------------------------------------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|--------|-------|-------|-------|------------|
| OC1RS     | 0180 | Output Compare 1 Secondary Register |        |        |        |        |        |       |       |       |       |       |       |        |       |       |       | FFFF       |
| OC1R      | 0182 | Output Compare 1 Register           |        |        |        |        |        |       |       |       |       |       |       |        |       |       |       | FFFF       |
| OC1CON    | 0184 | —                                   | —      | OCSIDL | —      | —      | —      | —     | —     | —     | —     | —     | OCFLT | OCTSEL | OCM2  | OCM1  | OCM0  | 0000       |
| OC2RS     | 0186 | Output Compare 2 Secondary Register |        |        |        |        |        |       |       |       |       |       |       |        |       |       |       | FFFF       |
| OC2R      | 0188 | Output Compare 2 Register           |        |        |        |        |        |       |       |       |       |       |       |        |       |       |       | FFFF       |
| OC2CON    | 018A | —                                   | —      | OCSIDL | —      | —      | —      | —     | —     | —     | —     | —     | OCFLT | OCTSEL | OCM2  | OCM1  | OCM0  | 0000       |
| OC3RS     | 018C | Output Compare 3 Secondary Register |        |        |        |        |        |       |       |       |       |       |       |        |       |       |       | FFFF       |
| OC3R      | 018E | Output Compare 3 Register           |        |        |        |        |        |       |       |       |       |       |       |        |       |       |       | FFFF       |
| OC3CON    | 0190 | —                                   | —      | OCSIDL | —      | —      | —      | —     | —     | —     | —     | —     | OCFLT | OCTSEL | OCM2  | OCM1  | OCM0  | 0000       |
| OC4RS     | 0192 | Output Compare 4 Secondary Register |        |        |        |        |        |       |       |       |       |       |       |        |       |       |       | FFFF       |
| OC4R      | 0194 | Output Compare 4 Register           |        |        |        |        |        |       |       |       |       |       |       |        |       |       |       | FFFF       |
| OC4CON    | 0196 | —                                   | —      | OCSIDL | —      | —      | —      | —     | —     | —     | —     | —     | OCFLT | OCTSEL | OCM2  | OCM1  | OCM0  | 0000       |
| OC5RS     | 0198 | Output Compare 5 Secondary Register |        |        |        |        |        |       |       |       |       |       |       |        |       |       |       | FFFF       |
| OC5R      | 019A | Output Compare 5 Register           |        |        |        |        |        |       |       |       |       |       |       |        |       |       |       | FFFF       |
| OC5CON    | 019C | —                                   | —      | OCSIDL | —      | —      | —      | —     | —     | —     | —     | —     | OCFLT | OCTSEL | OCM2  | OCM1  | OCM0  | 0000       |

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

**TABLE 4-9: I<sup>2</sup>C™ 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 1  |       |             |       |       |             |       |       | 0000       |
| I2C1TRN   | 0202 | —       | —      | —       | —      | —      | —      | —                  | —                              | Transmit Register 1 |       |             |       |       |             |       |       | 00FF       |
| I2C1BRG   | 0204 | —       | —      | —       | —      | —      | —      | —                  | Baud Rate Generator Register 1 |                     |       |             |       |       |             |       | 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 $\bar{A}$ | P     | S     | R $\bar{W}$ | RBF   | TBF   | 0000       |
| I2C1ADD   | 020A | —       | —      | —       | —      | —      | —      | Address Register 1 |                                |                     |       |             |       |       |             | 0000  |       |            |
| I2C1MSK   | 020C | —       | —      | —       | —      | —      | —      | AMSK9              | AMSK8                          | AMSK7               | AMSK6 | AMSK5       | AMSK4 | AMSK3 | AMSK2       | AMSK1 | AMSK0 | 0000       |
| I2C2RCV   | 0210 | —       | —      | —       | —      | —      | —      | —                  | —                              | Receive Register 2  |       |             |       |       |             |       |       | 0000       |
| I2C2TRN   | 0212 | —       | —      | —       | —      | —      | —      | —                  | —                              | Transmit Register 2 |       |             |       |       |             |       |       | 00FF       |
| I2C2BRG   | 0214 | —       | —      | —       | —      | —      | —      | —                  | Baud Rate Generator Register 2 |                     |       |             |       |       |             |       | 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 $\bar{A}$ | P     | S     | R $\bar{W}$ | RBF   | TBF   | 0000       |
| I2C2ADD   | 021A | —       | —      | —       | —      | —      | —      | Address Register 2 |                                |                     |       |             |       |       |             | 0000  |       |            |
| I2C2MSK   | 021C | —       | —      | —       | —      | —      | —      | AMSK9              | AMSK8                          | AMSK7               | AMSK6 | AMSK5       | AMSK4 | AMSK3 | AMSK2       | AMSK1 | AMSK0 | 0000       |

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

**TABLE 4-10: UART 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 |
|-----------|------|--|--------|----------|--------|--------|--------|-------|-------|----------|----------|-------|-------|-------|--------|--------|-------|------------|
| 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 | —                                      | —      | —        | —      | —      | —      | —     | UTX8  | UTX7     | UTX6     | UTX5  | UTX4  | UTX3  | UTX2   | UTX1   | UTX0  | 0000       |
| U1RXREG   | 0226 | —                                      | —      | —        | —      | —      | —      | —     | URX8  | URX7     | URX6     | URX5  | URX4  | URX3  | URX2   | URX1   | URX0  | 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  | URCISEL1 | URCISEL0 | ADDEN | RIDLE | PERR  | FERR   | OERR   | URXDA | 0110       |
| U2TXREG   | 0234 | —                                      | —      | —        | —      | —      | —      | —     | UTX8  | UTX7     | UTX6     | UTX5  | UTX4  | UTX3  | UTX2   | UTX1   | UTX0  | 0000       |
| U2RXREG   | 0236 | —                                      | —      | —        | —      | —      | —      | —     | URX8  | URX7     | URX6     | URX5  | URX4  | URX3  | URX2   | URX1   | URX0  | 0000       |
| U2BRG     | 0238 | Baud Rate Generator Prescaler          |        |          |        |        |        |       |       |          |          |       |       |       |        |        |       | 0000       |

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

**TABLE 4-11: SPI 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 |
|-----------|------|------------------------------|--------|---------|--------|--------|---------|---------|---------|-------|--------|--------|-------|-------|-------|--------|--------|------------|
| SPI1STAT  | 0240 | SPIEN                        | —      | SPIIDL  | —      | —      | SPIBEC2 | SPIBEC1 | SPIBEC0 | SRMPT | SPIROV | SRXMPT | ISEL2 | ISEL1 | ISEL0 | 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 | SPI1 Transmit/Receive Buffer |        |         |        |        |         |         |         |       |        |        |       |       |       |        |        | 0000       |
| SPI2STAT  | 0260 | SPIEN                        | —      | SPIIDL  | —      | —      | SPIBEC2 | SPIBEC1 | SPIBEC0 | SRMPT | SPIROV | SRXMPT | ISEL2 | ISEL1 | ISEL0 | 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 | SPI2 Transmit/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                 | Bit 9                 | Bit 8                 | Bit 7                 | Bit 6 | Bit 5 | Bit 4  | Bit 3                 | Bit 2                 | Bit 1  | Bit 0  | All Resets |
|-----------|------|--------|--------|--------|--------|--------|------------------------|-----------------------|-----------------------|-----------------------|-------|-------|--------|-----------------------|-----------------------|--------|--------|------------|
| TRISA     | 02C0 | —      | —      | —      | —      | —      | TRISA10 <sup>(1)</sup> | TRISA9 <sup>(1)</sup> | TRISA8 <sup>(1)</sup> | TRISA7 <sup>(1)</sup> | —     | —     | TRISA4 | TRISA3 <sup>(2)</sup> | TRISA2 <sup>(3)</sup> | TRISA1 | TRISA0 | 079F       |
| PORTA     | 02C2 | —      | —      | —      | —      | —      | RA10 <sup>(1)</sup>    | RA9 <sup>(1)</sup>    | RA8 <sup>(1)</sup>    | RA7 <sup>(1)</sup>    | —     | —     | RA4    | RA3 <sup>(2)</sup>    | RA2 <sup>(3)</sup>    | RA1    | RA0    | 0000       |
| LATA      | 02C4 | —      | —      | —      | —      | —      | LATA10 <sup>(1)</sup>  | LATA9 <sup>(1)</sup>  | LATA8 <sup>(1)</sup>  | LATA7 <sup>(1)</sup>  | —     | —     | LATA4  | LATA3 <sup>(2)</sup>  | LATA2 <sup>(3)</sup>  | LATA1  | LATA0  | 0000       |
| ODCA      | 02C6 | —      | —      | —      | —      | —      | ODA10 <sup>(1)</sup>   | ODA9 <sup>(1)</sup>   | ODA8 <sup>(1)</sup>   | ODA7 <sup>(1)</sup>   | —     | —     | ODA4   | ODA3 <sup>(2)</sup>   | ODA2 <sup>(3)</sup>   | ODA1   | ODA0   | 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'.

**2:** Bits are available only when the primary oscillator is disabled (POSCMD<1:0> = 00); otherwise read as '0'.

**3:** Bits are available only when the primary oscillator is disabled or EC mode is selected (POSCMD<1:0> = 00 or 11) and CLK0 is disabled (OSCIOFNC = 0); otherwise, 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 | FFFF       |
| PORTB     | 02CA | RB15    | RB14    | RB13    | RB12    | RB11    | RB10    | RB9    | RB8    | RB7    | RB6    | RB5    | RB4    | RB3    | RB2    | RB1    | RB0    | 0000       |
| LATB      | 02CC | LATB15  | LATB14  | LATB13  | LATB12  | LATB11  | LATB10  | LATB9  | LATB8  | LATB7  | LATB6  | LATB5  | LATB4  | LATB3  | LATB2  | LATB1  | LATB0  | 0000       |
| 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  | Bit 8  | Bit 7  | Bit 6  | Bit 5  | Bit 4  | Bit 3  | Bit 2  | Bit 1  | Bit 0  | All Resets |
|----------------------|------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|------------|
| TRISC <sup>(1)</sup> | 02D0 | —      | —      | —      | —      | —      | —      | TRISC9 | TRISC8 | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 03FF       |
| PORTC <sup>(1)</sup> | 02D2 | —      | —      | —      | —      | —      | —      | RC9    | RC8    | RC7    | RC6    | RC5    | RC4    | RC3    | RC2    | RC1    | RC0    | 0000       |
| LATC <sup>(1)</sup>  | 02D4 | —      | —      | —      | —      | —      | —      | LATC9  | LATC8  | LATC7  | LATC6  | LATC5  | LATC4  | LATC3  | LATC2  | LATC1  | LATC0  | 0000       |
| ODCC <sup>(1)</sup>  | 02D6 | —      | —      | —      | —      | —      | —      | ODC9   | OSC8   | ODC7   | ODC6   | ODC5   | ODC4   | ODC3   | ODC2   | ODC1   | ODC0   | 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 4-15: 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 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —     | —     | —     | —     | RTSECSEL | 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                | SSRC1                | SSRC0                | —     | —     | ASAM   | SAMP   | DONE   | 0000   |            |
| AD1CON2   | 0322 | VCFG2              | VCFG1  | VCFG0  | —      | —      | CSCNA  | —      | —                    | BUFS                 | —                    | SMP13 | SMP12 | SMP11  | SMP10  | BUFM   | ALTS   | 0000       |
| AD1CON3   | 0324 | ADRC               | —      | —      | SAMC4  | SAMC3  | SAMC2  | SAMC1  | SAMC0                | ADCS7                | ADCS6                | ADCS5 | ADCS4 | ADCS3  | ADCS2  | ADCS1  | ADCS0  | 0000       |
| AD1CHS    | 0328 | CH0NB              | —      | —      | —      | CH0SB3 | CH0SB2 | CH0SB1 | CH0SB0               | CH0NA                | —                    | —     | —     | CH0SA3 | CH0SA2 | CH0SA1 | CH0SA0 | 0000       |
| AD1PCFG   | 032C | PCFG15             | —      | —      | PCFG12 | PCFG11 | PCFG10 | PCFG9  | PCFG8 <sup>(1)</sup> | PCFG7 <sup>(1)</sup> | PCFG6 <sup>(1)</sup> | PCFG5 | PCFG4 | PCFG3  | PCFG2  | PCFG1  | PCFG0  | 0000       |
| AD1CSSL   | 0330 | CSSL15             | —      | —      | CSSL12 | CSSL11 | CSSL10 | CSSL9  | CSSL8 <sup>(1)</sup> | CSSL7 <sup>(1)</sup> | CSSL6 <sup>(1)</sup> | CSSL5 | CSSL4 | CSSL3  | CSSL2  | CSSL1  | CSSL0  | 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 4-17: 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 | ADDR9  | ADDR8  | ADDR7  | ADDR6  | ADDR5  | ADDR4  | ADDR3  | ADDR2  | 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 | PTEN9  | PTEN8  | PTEN7  | PTEN6  | PTEN5  | PTEN4  | PTEN3  | PTEN2  | 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.

**TABLE 4-18: 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> |        |         |         |         |        |          |          |       |       |       |       |       |       |       |       | xxxxx      |
| 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>   |        |         |         |         |        |          |          |       |       |       |       |       |       |       |       | xxxxx      |
| RCFGCAL   | 0626 | RTCEN   | —      | RTCWREN | RTCSYNC | HALFSEC | RTCOC  | RTCPTR1  | RTCPTR0  | CAL7  | CAL6  | CAL5  | CAL4  | CAL3  | CAL2  | CAL1  | CAL0  | 0000       |

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

**TABLE 4-19: DUAL COMPARATOR 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 |
|-----------|------|--------|--------|--------|--------|--------|--------|---------|---------|-------|-------|-------|-------|-------|-------|-------|-------|------------|
| CMCON     | 0630 | CMIDL  | —      | C2EVT  | C1EVT  | C2EN   | C1EN   | C2OUTEN | C1OUTEN | C2OUT | C1OUT | C2INV | C1INV | C2NEG | C2POS | C1NEG | C1POS | 0000       |
| CVRCON    | 0632 | —      | —      | —      | —      | —      | —      | —       | —       | CVREN | CVROE | CVRR  | CVRSS | CVR3  | CVR2  | CVR1  | CVR0  | 0000       |

**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 |
|-----------|------|-------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|------------|
| CRCCON    | 0640 | —                       | —      | CSIDL  | VWORD4 | VWORD3 | VWORD2 | VWORD1 | VWORD0 | CRCFUL | CRCMPT | —     | CRCGO | PLEN3 | PLEN2 | PLEN1 | PLEN0 | 0040       |
| CRCXOR    | 0642 | X15                     | X14    | X13    | X12    | X11    | X10    | X9     | X8     | X7     | X6     | X5    | X4    | X3    | X2    | X1    | —     | 0000       |
| CRCDAT    | 0644 | CRC Data Input Register |        |        |        |        |        |        |        |        |        |       |       |       |       |       |       | 0000       |
| CRCWDAT   | 0646 | CRC Result Register     |        |        |        |        |        |        |        |        |        |       |       |       |       |       |       | 0000       |

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

**TABLE 4-21: 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                | —     | —     | —     | SD1R4                 | SD1R3                 | SD1R2                 | SD1R1                 | SD1R0                 | 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     | 06D0 | —      | —      | —      | RP17R4 <sup>(1)</sup> | RP17R3 <sup>(1)</sup> | RP17R2 <sup>(1)</sup> | RP17R1 <sup>(1)</sup> | RP17R0 <sup>(1)</sup> | —     | —     | —     | RP16R4 <sup>(1)</sup> | RP16R3 <sup>(1)</sup> | RP16R2 <sup>(1)</sup> | RP16R1 <sup>(1)</sup> | RP16R0 <sup>(1)</sup> | 0000       |
| RPOR9     | 06D2 | —      | —      | —      | RP19R4 <sup>(1)</sup> | RP19R3 <sup>(1)</sup> | RP19R2 <sup>(1)</sup> | RP19R1 <sup>(1)</sup> | RP19R0 <sup>(1)</sup> | —     | —     | —     | RP18R4 <sup>(1)</sup> | RP18R3 <sup>(1)</sup> | RP18R2 <sup>(1)</sup> | RP18R1 <sup>(1)</sup> | RP18R0 <sup>(1)</sup> | 0000       |
| RPOR10    | 06D4 | —      | —      | —      | RP21R4 <sup>(1)</sup> | RP21R3 <sup>(1)</sup> | RP21R2 <sup>(1)</sup> | RP21R1 <sup>(1)</sup> | RP21R0 <sup>(1)</sup> | —     | —     | —     | RP20R4 <sup>(1)</sup> | RP20R3 <sup>(1)</sup> | RP20R2 <sup>(1)</sup> | RP20R1 <sup>(1)</sup> | RP20R0 <sup>(1)</sup> | 0000       |
| RPOR11    | 06D6 | —      | —      | —      | RP23R4 <sup>(1)</sup> | RP23R3 <sup>(1)</sup> | RP23R2 <sup>(1)</sup> | RP23R1 <sup>(1)</sup> | RP23R0 <sup>(1)</sup> | —     | —     | —     | RP22R4 <sup>(1)</sup> | RP22R3 <sup>(1)</sup> | RP22R2 <sup>(1)</sup> | RP22R1 <sup>(1)</sup> | RP22R0 <sup>(1)</sup> | 0000       |
| RPOR12    | 06D8 | —      | —      | —      | RP25R4 <sup>(1)</sup> | RP25R3 <sup>(1)</sup> | RP25R2 <sup>(1)</sup> | RP25R1 <sup>(1)</sup> | RP25R0 <sup>(1)</sup> | —     | —     | —     | RP24R4 <sup>(1)</sup> | RP24R3 <sup>(1)</sup> | RP24R2 <sup>(1)</sup> | RP24R1 <sup>(1)</sup> | RP24R0 <sup>(1)</sup> | 0000       |

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

**Note 1:** Bits are only available on the 44-pin devices; otherwise, they read as '0'.

**TABLE 4-22: CLOCK CONTROL 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 | —      | —      | —      | —      | CM     | VREGS  | EXTR    | SWR    | SWDTEN | WDTO  | SLEEP | IDLE  | BOR    | POR   | (Note 1)   |
| OSCCON    | 0742 | —      | COSC2  | COSC1  | COSC0  | —      | NOSC2  | NOSC1  | NOSC0  | CLKLOCK | IOLOCK | LOCK   | —     | CF    | —     | SOSCEN | OSWEN | (Note 2)   |
| CLKDIV    | 0744 | ROI    | DOZE2  | DOZE1  | DOZE0  | DOZEN  | RCDIV2 | RCDIV1 | RCDIV0 | —       | —      | —      | —     | —     | —     | —      | —     | 3140       |
| OSCTUN    | 0748 | —      | —      | —      | —      | —      | —      | —      | —      | —       | —      | TUN5   | TUN4  | TUN3  | TUN2  | TUN1   | TUN0  | 0000       |

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

- Note 1:** RCON register Reset values are dependent on type of Reset.  
**2:** OSCCON register Reset values are dependent on configuration fuses and by type of Reset.

**TABLE 4-23: 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 <sup>(1)</sup> |
| NVMKEY    | 0766 | —      | —      | —      | —      | —      | —      | —     | —     | —     | —     | —     | —     | NVMKEY<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 4-24: 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 | CRCPMD | —     | —     | —      | —      | —     | I2C2MD | —      | 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 pre-decrements 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.

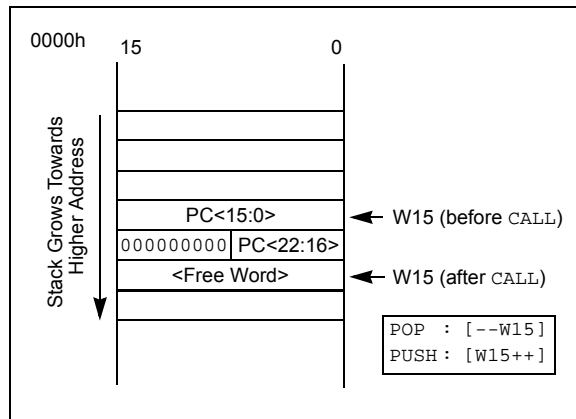
**Note:** A PC push during exception processing will concatenate the SRL register to the MSB of the PC prior to the push.

The Stack Pointer Limit Value register (SPLIM), 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 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 register (TBLPAG) 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 register (PSVPAG) 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 4-25 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.

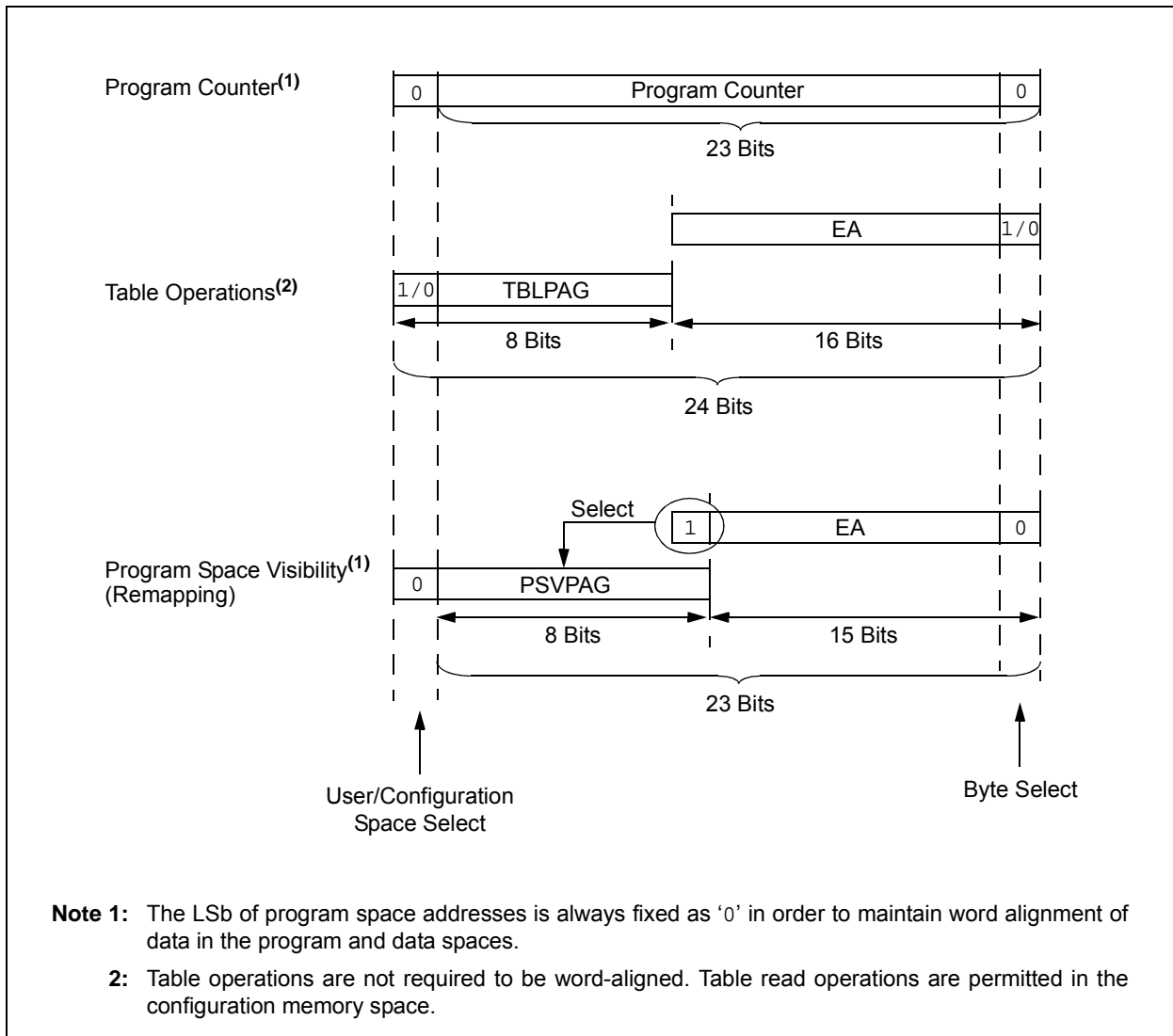
# PIC24FJ64GA004 FAMILY

**TABLE 4-25: PROGRAM SPACE ADDRESS CONSTRUCTION**

| Access Type                                    | Access Space  | Program Space Address                              |             |               |                              |     |
|--|---------------|--|-------------|---------------|------------------------------|-----|
|  |               | <23>   | <22:16>     | <15>          | <14:1>                       | <0> |
| Instruction Access<br>(Code Execution)         | User          | 0  | PC<22:1>    |               |                              | 0   |
|  |               | 0xxx xxxx xxxx xxxx xxxx xxx0                      |             |               |                              |     |
| TBLRD/TBLWT<br>(Byte/Word Read/Write)          | User          | TBLPAG<7:0>  |             | Data EA<15:0> |                              |     |
|  |               | 0xxx xxxx                      xxxx xxxx xxxx xxxx |             |               |                              |     |
|  | Configuration | TBLPAG<7:0>  |             | Data EA<15:0> |                              |     |
|  |               | 1xxx xxxx                      xxxx xxxx xxxx xxxx |             |               |                              |     |
| Program Space Visibility<br>(Block Remap/Read) | User          | 0  | PSVPAG<7:0> |               | Data EA<14:0> <sup>(1)</sup> |     |
|  |               | 0  | xxxx xxxx   |               | xxx xxxx xxxx xxxx           |     |

**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**



## 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 byte select is '1'; the lower byte is selected when it is '0'.

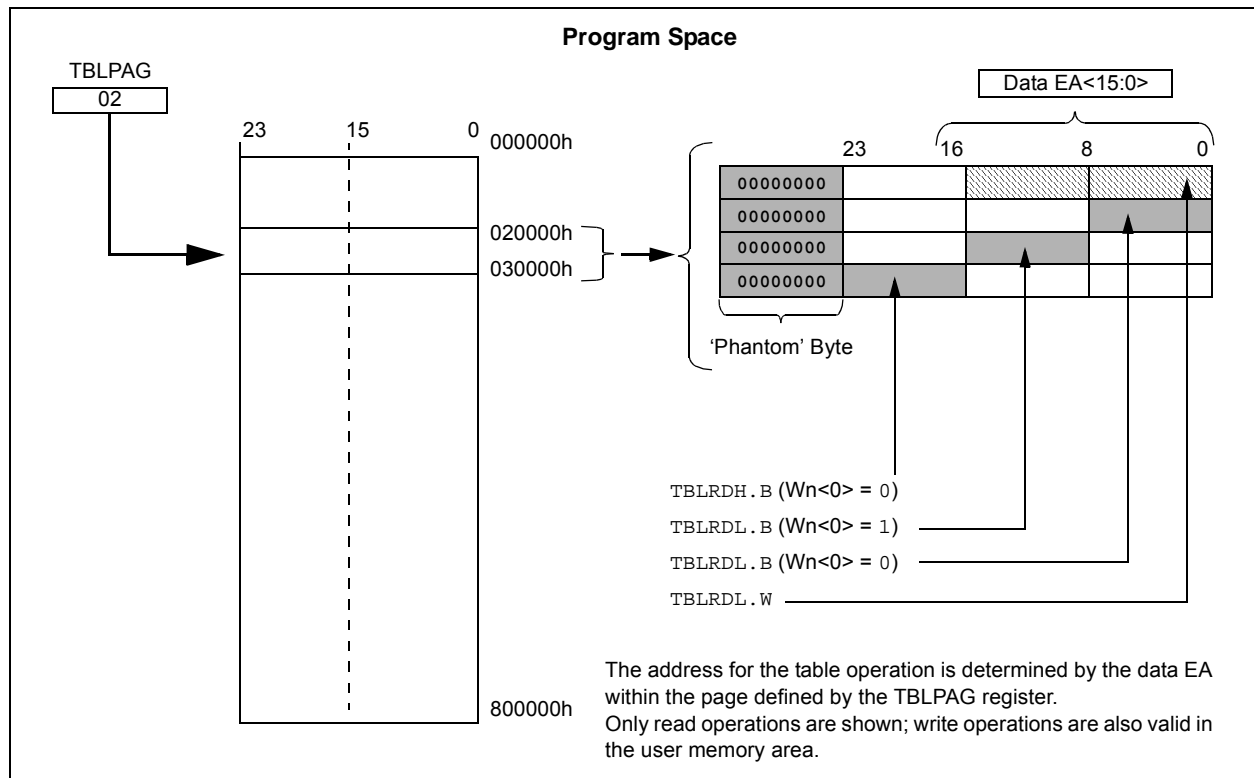
2. `TBLRDH` (Table Read High): In Word mode, it maps the entire upper word of a program address ( $P<23:16>$ ) to a data address. Note that  $D<15:8>$ , the 'phantom' byte, will always be '0'. In Byte mode, it maps the upper or lower byte of the program word to  $D<7:0>$  of the data address, as above. 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**



# PIC24FJ64GA004 FAMILY

## 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 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.

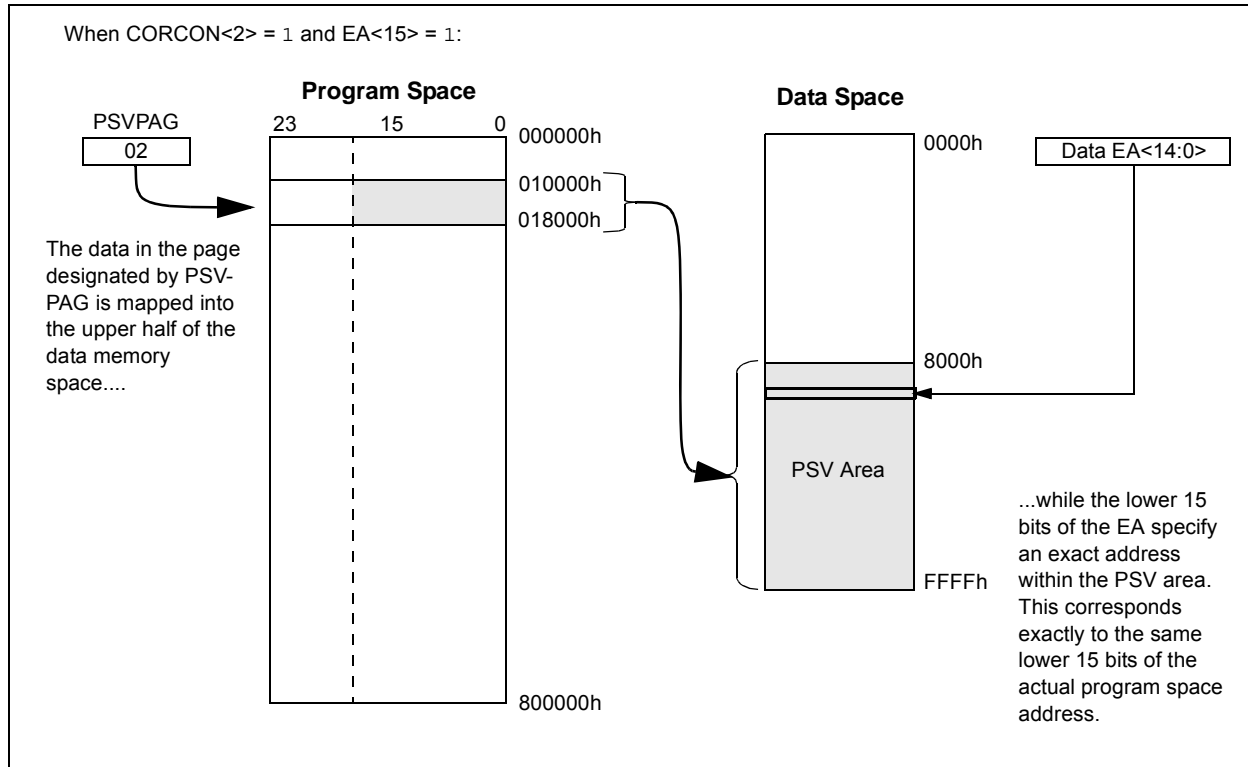
For operations that use PSV and are executed outside a `REPEAT` loop, the `MOV` and `MOV.D` instructions will require one instruction cycle in addition to the specified execution time. All other instructions will require two instruction cycles in addition to the specified execution time.

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**





# PIC24FJ64GA004 FAMILY

## 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 PIC24FJ64GA004 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 2.25V.

Flash memory can be programmed in three ways:

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

ICSP allows a PIC24FJ64GA004 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 PGCx and PGDx, 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.

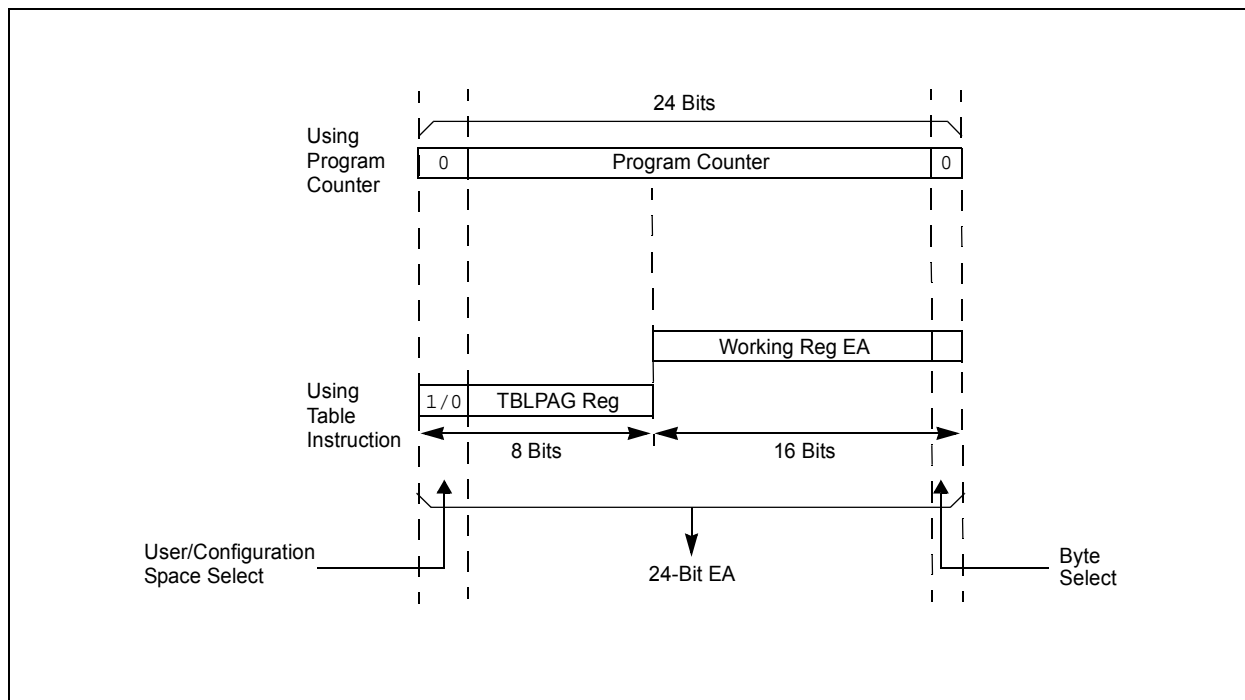
### 5.1 Table Instructions and Flash Programming

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



# PIC24FJ64GA004 FAMILY

---

## 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 FFFFFFFh. 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 it 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 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.

## 5.4 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 5.5 “Programming Operations”** for further details.

## 5.5 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.

Configuration Word values are stored in the last two locations of program memory. Performing a page erase operation on the last page of program memory clears these values and enables code protection. As a result, avoid performing page erase operations on the last page of program memory.

# PIC24FJ64GA004 FAMILY

## REGISTER 5-1: NVMCON: FLASH MEMORY CONTROL REGISTER

|        |       |       |     |     |     |     |       |
|--------|-------|-------|-----|-----|-----|-----|-------|
| R/SO-0 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 | U-0   |
| WR     | WREN  | WRERR | —   | —   | —   | —   | —     |
| bit 15 |       |       |     |     |     |     | bit 8 |

|       |       |     |     |                       |                       |                       |                       |
|-------|-------|-----|-----|-----------------------|-----------------------|-----------------------|-----------------------|
| U-0   | R/W-0 | U-0 | U-0 | R/W-0                 | R/W-0                 | R/W-0                 | R/W-0                 |
| —     | ERASE | —   | —   | NVMOP3 <sup>(1)</sup> | NVMOP2 <sup>(1)</sup> | NVMOP1 <sup>(1)</sup> | NVMOP0 <sup>(1)</sup> |
| bit 7 |       |     |     |                       |                       |                       | bit 0                 |

|                   |                   |                                    |                    |
|-------------------|-------------------|------------------------------------|--------------------|
| <b>Legend:</b>    | SO = Set 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      **WR:** Write Control bit  
 1 = Initiates a Flash memory program or erase operation. The operation is self-timed and the bit is cleared by hardware once operation is complete.  
 0 = Program or erase operation is complete and inactive
- bit 14      **WREN:** Write Enable bit  
 1 = Enable Flash program/erase operations  
 0 = Inhibit Flash program/erase operations
- bit 13      **WRERR:** Write Sequence Error Flag bit  
 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 = Perform the erase operation specified by NVMOP3:NVMOP0 on the next WR command  
 0 = Perform the program operation specified by NVMOP3:NVMOP0 on the next WR command
- bit 5-4     **Unimplemented:** Read as '0'
- bit 3-0     **NVMOP3:NVMOP0:** NVM Operation Select bits<sup>(1)</sup>  
 1111 = Memory bulk erase operation (ERASE = 1) or no operation (ERASE = 0)<sup>(2)</sup>  
 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:** All other combinations of NVMOP3:NVMOP0 are unimplemented.  
**Note 2:** Available in ICSP™ mode only. Refer to device programming specification.

# PIC24FJ64GA004 FAMILY

## 5.5.1 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:

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 5-3.

### EXAMPLE 5-1: ERASING A PROGRAM MEMORY BLOCK

```
; 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
```

# PIC24FJ64GA004 FAMILY

## EXAMPLE 5-2: LOADING THE WRITE BUFFERS

```
; 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 5-3: INITIATING A PROGRAMMING SEQUENCE

```
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     ;                    ; 2 NOPs required after setting WR
NOP     ;
BTSC    NVMCON, #15         ; Wait for the sequence to be completed
BRA     $-2                 ;
```

# PIC24FJ64GA004 FAMILY

---

## 5.5.2 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 5-4).

### EXAMPLE 5-4: PROGRAMMING A SINGLE WORD OF FLASH PROGRAM MEMORY

```
; 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_N, W2             ;
MOV   #HIGH_BYTE_N, 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                                       ; 2 NOPs required after setting WR
NOP                                       ;
```

# PIC24FJ64GA004 FAMILY

## 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,  $\overline{\text{SYSRST}}$ . 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  $\overline{\text{SYSRST}}$  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.

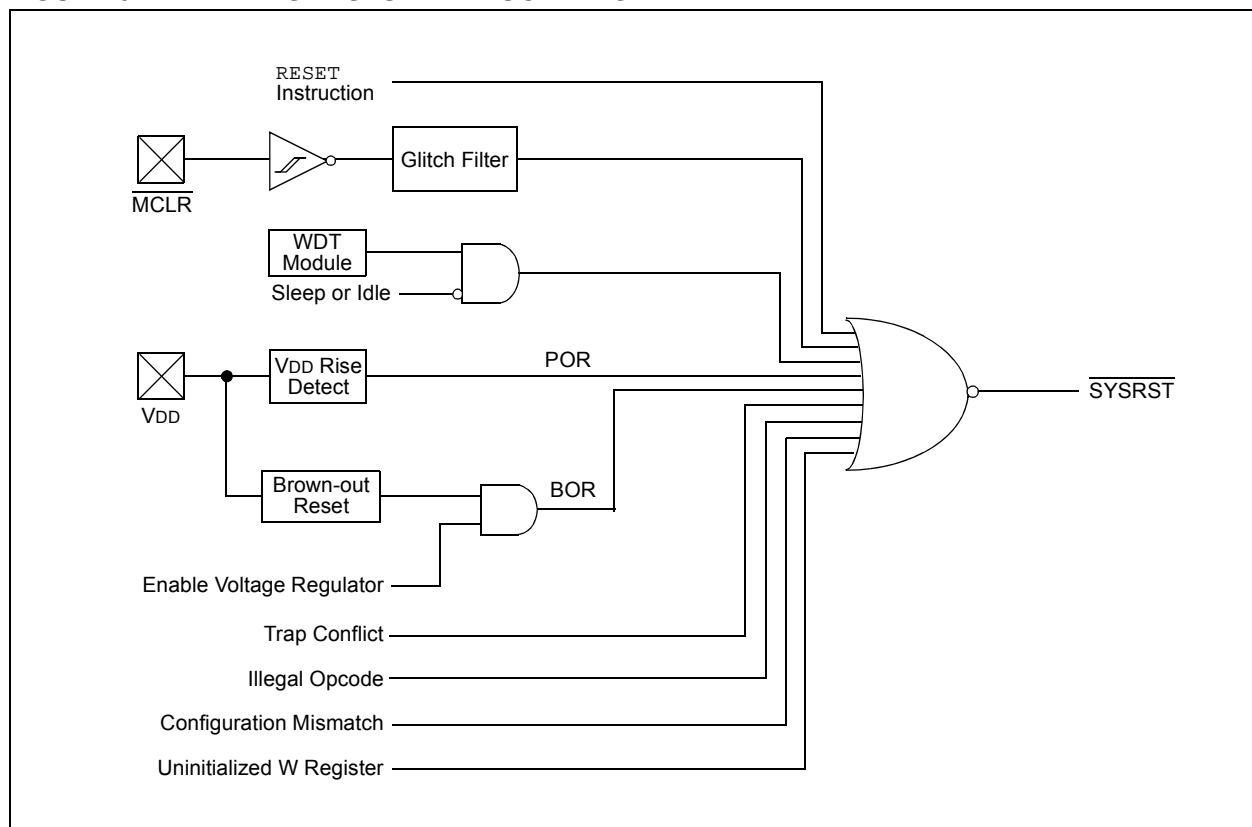
**Note:** Refer to the specific peripheral or CPU section of this manual for register Reset states.

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 manual.

**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**



# PIC24FJ64GA004 FAMILY

## REGISTER 6-1: RCON: RESET CONTROL REGISTER<sup>(1)</sup>

|        |        |     |     |     |     |       |       |
|--------|--------|-----|-----|-----|-----|-------|-------|
| R/W-0  | R/W-0  | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 |
| TRAPR  | IOPUWR | —   | —   | —   | —   | CM    | VREGS |
| 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 <sup>(2)</sup> | WDTO  | SLEEP | IDLE  | BOR   | POR   |
| 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      **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 13-10   **Unimplemented:** Read as '0'
- 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        **VREGS:** Voltage Regulator Standby Enable bit  
 1 = Regulator remains active during Sleep  
 0 = Regulator goes to standby during Sleep
- bit 7        **EXTR:** External Reset (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<sup>(2)</sup>  
 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
- 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-up Reset has occurred  
 0 = A Power-up 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.

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



# PIC24FJ64GA004 FAMILY

**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>)    | $\overline{\text{MCLR}}$ Reset                    | POR                    |
| SWR (RCON<6>)     | RESET Instruction                                 | POR                    |
| WDTO (RCON<4>)    | WDT Time-out                                      | PWRSV Instruction, POR |
| SLEEP (RCON<3>)   | PWRSV #SLEEP Instruction                          | POR                    |
| IDLE (RCON<2>)    | PWRSV #IDLE Instruction                           | POR                    |
| BOR (RCON<1>)     | POR, BOR  | —                      |
| POR (RCON<0>)     | 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                      | FNOS Configuration bits<br>(CW2<10:8>) |
| BOR                      |  |
| $\overline{\text{MCLR}}$ | COSC Control bits<br>(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,  $\overline{\text{SYSRST}}$ , is released after the POR and PWRT delay times expire.

The time that 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.

# PIC24FJ64GA004 FAMILY

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

| Reset Type      | Clock Source          | $\overline{\text{SYSRST}}$ Delay                       | System Clock Delay                 | FSCM Delay | Notes            |
|-----------------|-----------------------|--|------------------------------------|------------|------------------|
| POR             | EC, FRC, FRCDIV, LPRC | $T_{\text{POR}} + T_{\text{STARTUP}} + T_{\text{RST}}$ | —                                  | —          | 1, 2, 3          |
|                 | ECPLL, FRCPLL         | $T_{\text{POR}} + T_{\text{STARTUP}} + T_{\text{RST}}$ | TLOCK                              | TFSCM      | 1, 2, 3, 5, 6    |
|                 | XT, HS, SOSC          | $T_{\text{POR}} + T_{\text{STARTUP}} + T_{\text{RST}}$ | TOST                               | TFSCM      | 1, 2, 3, 4, 6    |
|                 | XTPLL, HSPLL          | $T_{\text{POR}} + T_{\text{STARTUP}} + T_{\text{RST}}$ | $T_{\text{OST}} + T_{\text{LOCK}}$ | TFSCM      | 1, 2, 3, 4, 5, 6 |
| BOR             | EC, FRC, FRCDIV, LPRC | $T_{\text{STARTUP}} + T_{\text{RST}}$                  | —                                  | —          | 2, 3             |
|                 | ECPLL, FRCPLL         | $T_{\text{STARTUP}} + T_{\text{RST}}$                  | TLOCK                              | TFSCM      | 2, 3, 5, 6       |
|                 | XT, HS, SOSC          | $T_{\text{STARTUP}} + T_{\text{RST}}$                  | TOST                               | TFSCM      | 2, 3, 4, 6       |
|                 | XTPLL, HSPLL          | $T_{\text{STARTUP}} + T_{\text{RST}}$                  | $T_{\text{OST}} + T_{\text{LOCK}}$ | TFSCM      | 2, 3, 4, 5, 6    |
| MCLR            | Any Clock             | TRST   | —                                  | —          | 3                |
| WDT             | Any Clock             | TRST   | —                                  | —          | 3                |
| Software        | Any clock             | TRST   | —                                  | —          | 3                |
| Illegal Opcode  | Any Clock             | TRST   | —                                  | —          | 3                |
| Uninitialized W | Any Clock             | TRST   | —                                  | —          | 3                |
| Trap Conflict   | Any Clock             | TRST   | —                                  | —          | 3                |

**Note 1:**  $T_{\text{POR}}$  = Power-on Reset delay (10  $\mu\text{s}$  nominal).

**2:**  $T_{\text{STARTUP}}$  =  $T_{\text{VREG}}$  (10  $\mu\text{s}$  nominal) if on-chip regulator is enabled or  $T_{\text{PWRT}}$  (64 ms nominal) if on-chip regulator is disabled.

**3:**  $T_{\text{RST}}$  = Internal state Reset time.

**4:**  $T_{\text{OST}}$  = Oscillator Start-up Timer. A 10-bit counter counts 1024 oscillator periods before releasing the oscillator clock to the system.

**5:** TLOCK = PLL lock time (2 ms nominal).

**6:** TFSCM = Fail-Safe Clock Monitor delay.

# PIC24FJ64GA004 FAMILY

---

## 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  $\overline{\text{SYSRST}}$  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  $\overline{\text{SYSRST}}$  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.

## 6.2.2.1 FSCM Delay for Crystal and PLL Clock Sources

When the system clock source is provided by a crystal oscillator and/or the PLL, a small delay,  $T_{\text{FSCM}}$ , will automatically be inserted after the POR and PWRT delay times. The FSCM will not begin to monitor the system clock source until this delay expires. The FSCM delay time is nominally 100  $\mu\text{s}$  and provides additional time for the oscillator and/or PLL to stabilize. In most cases, the FSCM delay will prevent an oscillator failure trap at a device Reset when the PWRT is disabled.

## 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 the CW2 register (see Table 6-2). The RCFGAL and NVMCON registers are only affected by a POR.

# PIC24FJ64GA004 FAMILY

---

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.

PIC24FJ64GA004 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.

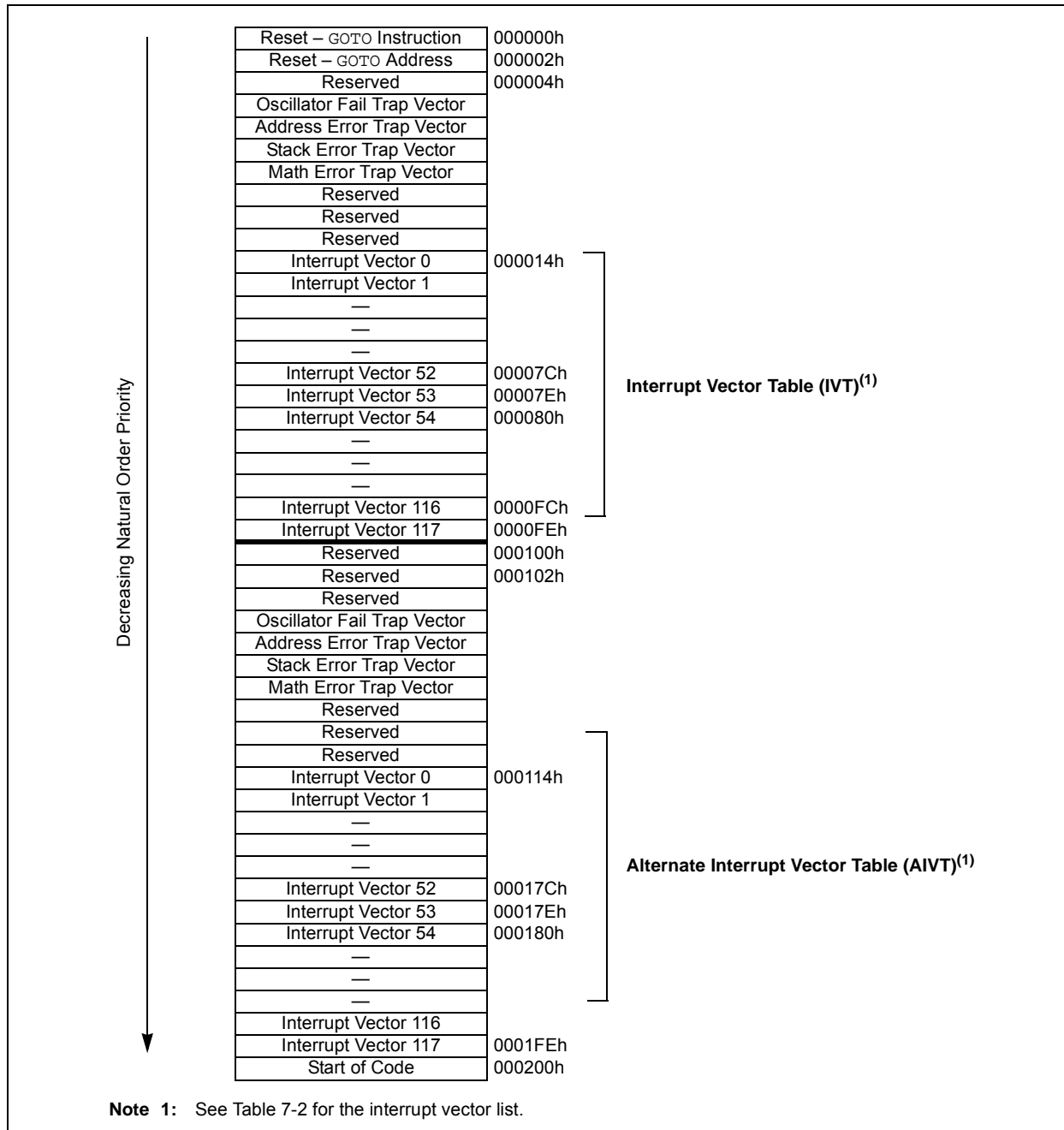
## 7.2 Reset Sequence

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.

# PIC24FJ64GA004 FAMILY

**FIGURE 7-1: PIC24F INTERRUPT VECTOR TABLE**



**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           |

# PIC24FJ64GA004 FAMILY

**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> |
| 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     | 000034h      | 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>  |
| 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<13> | 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>                 | IEC0<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>  |
| LVD Low-Voltage Detect    | 72            | 0000A4h     | 000124h      | IFS4<8>                 | IEC4<8>  | IPC17<2:0>   |

# PIC24FJ64GA004 FAMILY

---

## 7.3 Interrupt Control and Status Registers

The PIC24FJ64GA004 family of devices implements a total of 28 registers for the interrupt controller:

- INTCON1
- INTCON2
- IFS0 through IFS4
- IEC0 through IEC4
- IPC0 through IPC12, IPC15, IPC16 and IPC18

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 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 same sequence that they are listed 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 IPL2:IPL0 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 IPL2:IPL0, also indicates the current CPU priority level. IPL3 is a read-only bit so that trap events cannot be masked by the user software.

All interrupt registers are described in Register 7-1 through Register 7-29, in the following pages.



# PIC24FJ64GA004 FAMILY

## 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 <sup>(1)</sup> |
| 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 <sup>(2,3)</sup> | IPL1 <sup>(2,3)</sup> | IPL0 <sup>(2,3)</sup> | RA <sup>(1)</sup> | N <sup>(1)</sup> | OV <sup>(1)</sup> | Z <sup>(1)</sup> | C <sup>(1)</sup> |
| bit 7                 |                       |                       |                   |                  |                   |                  | bit 0            |

### Legend:

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

bit 7-5      **IPL2:IPL0: CPU Interrupt Priority Level Status bits<sup>(2,3)</sup>**  
 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)

- 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 <sup>(2)</sup> | PSV <sup>(1)</sup> | —   | —     |
| 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<sup>(2)</sup>**  
 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 IPL2:IPL0 bits (SR<7:5>) to form the CPU interrupt priority level.

# PIC24FJ64GA004 FAMILY

## 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'

# PIC24FJ64GA004 FAMILY

## 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

# PIC24FJ64GA004 FAMILY

## 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

# PIC24FJ64GA004 FAMILY

## 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

# PIC24FJ64GA004 FAMILY

## 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

# PIC24FJ64GA004 FAMILY

## 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'

# PIC24FJ64GA004 FAMILY

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

|        |     |     |     |     |     |     |       |
|--------|-----|-----|-----|-----|-----|-----|-------|
| U-0    | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 |
| —      | —   | —   | —   | —   | —   | —   | 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 15-9        **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'



# PIC24FJ64GA004 FAMILY

## 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 <sup>(1)</sup> |
| bit 7 |       |       |     |       |       |       | bit 0                 |

### Legend:

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

- bit 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<sup>(1)</sup>  
                  1 = Interrupt request enabled  
                  0 = Interrupt request not enabled

**Note 1:** If INTxIE = 1, this external interrupt input must be configured to an available RPN pin. See **Section 10.4 "Peripheral Pin Select"** for more information.

# PIC24FJ64GA004 FAMILY

## 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 <sup>(1)</sup> | 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 <sup>(1)</sup> | 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<sup>(1)</sup>  
                   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<sup>(1)</sup>  
                   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 INTxIE = 1, this external interrupt input must be configured to an available RPN pin. See **Section 10.4 "Peripheral Pin Select"** for more information.

# PIC24FJ64GA004 FAMILY

## 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

# PIC24FJ64GA004 FAMILY

## 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'

# PIC24FJ64GA004 FAMILY

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

|        |     |     |     |     |     |     |       |
|--------|-----|-----|-----|-----|-----|-----|-------|
| U-0    | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 |
| —      | —   | —   | —   | —   | —   | —   | 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 15-9     **Unimplemented:** Read as '0'
- bit 8       **LVDIE:** Low-Voltage Detect Interrupt Enable Status 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'

# PIC24FJ64GA004 FAMILY

## 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 **T1IP2:T1IP0:** 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 **OC1IP2:OC1IP0:** 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 **IC1IP2:IC1IP0:** 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 **INT0IP2:INT0IP0:** External Interrupt 0 Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)

•  
•  
•

001 = Interrupt is priority 1

000 = Interrupt source is disabled

# PIC24FJ64GA004 FAMILY

## 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 **T2IP2:T2IP0:** 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 **OC2IP2:OC2IP0:** 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 **IC2IP2:IC2IP0:** 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'

# PIC24FJ64GA004 FAMILY

## 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       **U1RXIP2:U1RXIP0:** 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        **SPI1IP2:SPI1IP0:** 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         **SPF1IP2:SPF1IP0:** 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         **T3IP2:T3IP0:** Timer3 Interrupt Priority bits
  - 111 = Interrupt is priority 7 (highest priority interrupt)
  - 
  - 
  - 
  - 001 = Interrupt is priority 1
  - 000 = Interrupt source is disabled



# PIC24FJ64GA004 FAMILY

## 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 **AD1IP2:AD1IP0:** 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 **U1TXIP2:U1TXIP0:** UART1 Transmitter Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)

•

•

•

001 = Interrupt is priority 1

000 = Interrupt source is disabled

# PIC24FJ64GA004 FAMILY

## 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   |
| —     | MI2C1P2 | MI2C1P1 | MI2C1P0 | —   | SI2C1P2 | SI2C1P1 | SI2C1P0 |
| 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      **CNIP2:CNIP0:** 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        **CMIP2:CMIP0:** 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         **MI2C1P2:MI2C1P0:** 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         **SI2C1P2:SI2C1P0:** Slave I2C1 Event Interrupt Priority bits
  - 111 = Interrupt is priority 7 (highest priority interrupt)
  - 
  - 
  - 
  - 001 = Interrupt is priority 1
  - 000 = Interrupt source is disabled

# PIC24FJ64GA004 FAMILY

## 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     **INT1IP2:INT1IP0:** External Interrupt 1 Priority bits  
 111 = Interrupt is priority 7 (highest priority interrupt)  
 •  
 •  
 •  
 001 = Interrupt is priority 1  
 000 = Interrupt source is disabled

# PIC24FJ64GA004 FAMILY

## 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 **T4IP2:T4IP0:** 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 **OC4IP2:OC4IP0:** 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 **OC3IP2:OC3IP0:** 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'

# PIC24FJ64GA004 FAMILY

## 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    **U2TXIP2:U2TXIP0:** 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    **U2RXIP2:U2RXIP0:** 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     **INT2IP2:INT2IP0:** 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     **T5IP2:T5IP0:** Timer5 Interrupt Priority bits  
 111 = Interrupt is priority 7 (highest priority interrupt)  
 •  
 •  
 •  
 001 = Interrupt is priority 1  
 000 = Interrupt source is disabled

# PIC24FJ64GA004 FAMILY

## 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

**SPI2IP2:SPI2IP0:** 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

**SPF2IP2:SPF2IP0:** SPI2 Fault Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)

•  
•  
•

001 = Interrupt is priority 1

000 = Interrupt source is disabled

# PIC24FJ64GA004 FAMILY

## 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   **IC5IP2:IC5IP0:** 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    **IC4IP2:IC4IP0:** 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    **IC3IP2:IC3IP0:** 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'

# PIC24FJ64GA004 FAMILY

## 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        **OC5IP2:OC5IP0:** 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   |
| —     | PMPPI2 | PMPPI1 | PMPPI0 | —   | —   | —   | —     |
| 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        **PMPPI2:PMPPI0:** 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'



# PIC24FJ64GA004 FAMILY

## 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   |
| —      | —   | —   | —   | —   | MI2C2P2 | MI2C2P1 | MI2C2P0 |
| bit 15 |     |     |     |     |         | bit 8   |         |

|       |         |         |         |     |     |       |     |
|-------|---------|---------|---------|-----|-----|-------|-----|
| U-0   | R/W-1   | R/W-0   | R/W-0   | U-0 | U-0 | U-0   | U-0 |
| —     | SI2C2P2 | SI2C2P1 | SI2C2P0 | —   | —   | —     | —   |
| 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 **MI2C2P2:MI2C2P0:** 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 **SI2C2P2:SI2C2P0:** 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'

# PIC24FJ64GA004 FAMILY

## 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      **RTCIP2:RTCIP0:** 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'

# PIC24FJ64GA004 FAMILY

## 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      **CRCIP2:CRCIP0:** 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        **U2ERIP2:U2ERIP0:** 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         **U1ERIP2:U1ERIP0:** 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'

# PIC24FJ64GA004 FAMILY

## 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      **LVDIP2:LVDIP0:** Low-Voltage Detect Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)

•

•

•

001 = Interrupt is priority 1

000 = Interrupt source is disabled

## 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.

# PIC24FJ64GA004 FAMILY

---

NOTES:

# PIC24FJ64GA004 FAMILY

## 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).

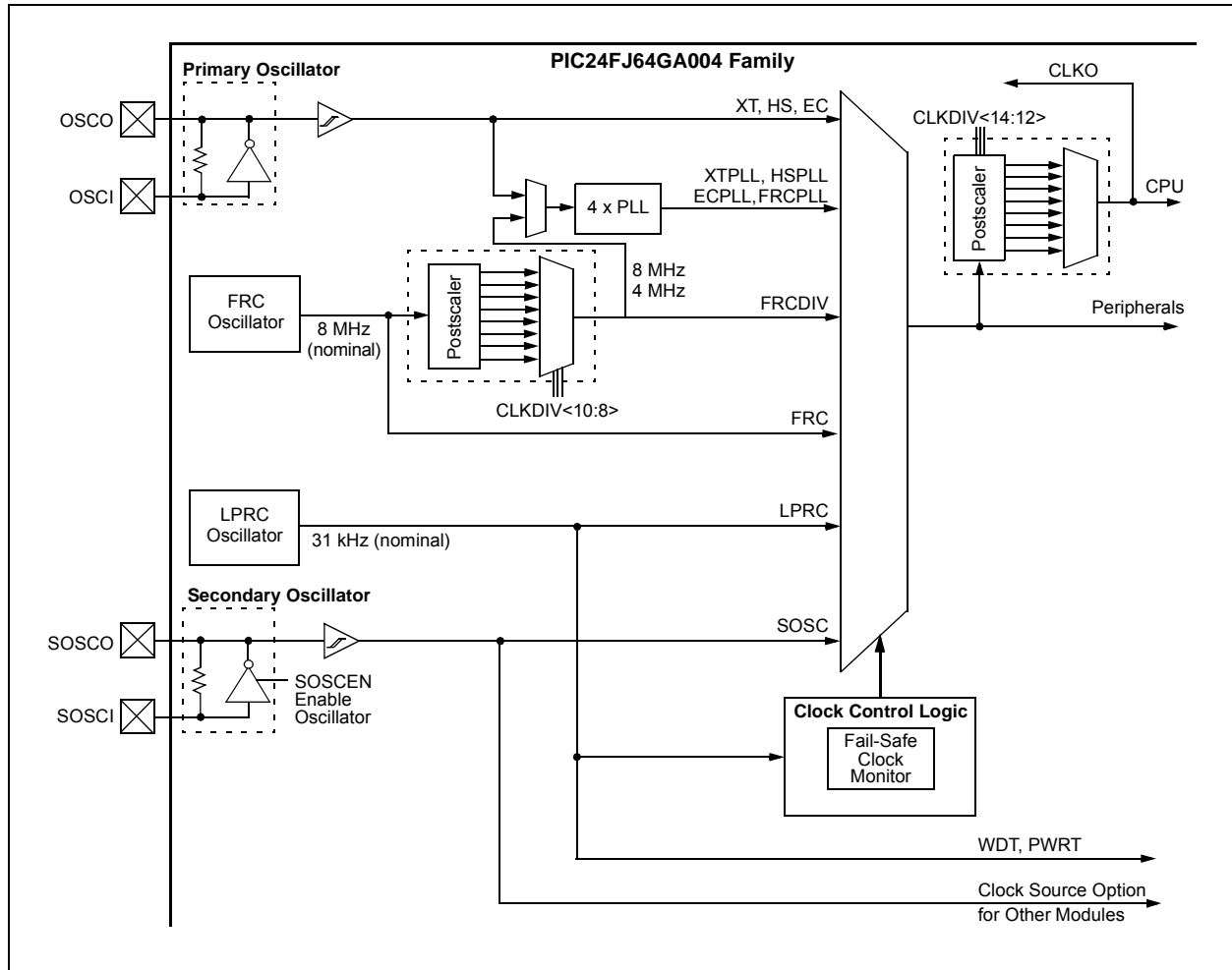
- Software-controllable switching between various clock sources
- Software-controllable postscaler for selective clocking of CPU for system power savings
- A Fail-Safe Clock Monitor (FSCM) that detects clock failure and permits safe application recovery or shutdown

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

The oscillator system for PIC24FJ64GA004 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

**FIGURE 8-1: PIC24FJ64GA004 FAMILY CLOCK DIAGRAM**



# PIC24FJ64GA004 FAMILY

## 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 24.1 “Configuration Bits”** for further details). The Primary Oscillator Configuration bits, POSCMD1:POSCMD0 (Configuration Word 2<1:0>), and the Initial Oscillator Select Configuration bits, FNOSC2:FNOSC0 (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 FCKSM1:FCKSM0 are both programmed ('00').

**TABLE 8-1: CONFIGURATION BIT VALUES FOR CLOCK SELECTION**

| Oscillator Mode                                 | Oscillator Source | POSCMD1:<br>POSCMD0 | FNOSC2:<br>FNOSC0 | Note        |
|---|-------------------|---------------------|-------------------|-------------|
| Fast RC Oscillator with Postscaler (FRCDIV)     | Internal          | 11                  | 111               | <b>1, 2</b> |
| (Reserved)                                      | Internal          | xx                  | 110               | <b>1</b>    |
| Low-Power RC Oscillator (LPRC)                  | Internal          | 11                  | 101               | <b>1</b>    |
| Secondary (Timer1) Oscillator (SOSC)            | Secondary         | 00                  | 100               | <b>1</b>    |
| 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               | <b>1</b>    |
| Fast RC Oscillator (FRC)                        | Internal          | 11                  | 000               | <b>1</b>    |

**Note 1:** OSCO pin function is determined by the OSCIOFCN Configuration bit.

**2:** This is the default oscillator mode for an unprogrammed (erased) device.



# PIC24FJ64GA004 FAMILY

## 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 Clock Divider register (Register 8-2) controls the features associated with Doze mode, as well as the postscaler for the FRC oscillator.

The FRC Oscillator Tune register (Register 8-3) allows the user to fine tune the FRC oscillator over a range of approximately  $\pm 12\%$ . 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 <sup>(1)</sup> | R/W-x <sup>(1)</sup> | R/W-x <sup>(1)</sup> |
| —      | COSC2 | COSC1 | COSC0 | —     | NOSC2                | NOSC1                | NOSC0                |
| bit 15 |       |       |       | bit 8 |                      |                      |                      |

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

|                   |                     |                                    |
|-------------------|---------------------|------------------------------------|
| <b>Legend:</b>    | CO = Clear Only bit | SO = Set 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 **COSC2:COSC0:** 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 **NOSC2:NOSC0:** New Oscillator Selection bits<sup>(1)</sup>

- 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.

# PIC24FJ64GA004 FAMILY

---

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

|       |   |
|-------|---|
| bit 7 | <b>CLKLOCK:</b> Clock Selection Lock Enabled bit<br><u>If FSCM is enabled (FCKSM1 = 1):</u><br>1 = Clock and PLL selections are locked<br>0 = Clock and PLL selections are not locked and may be modified by setting the OSWEN bit<br><u>If FSCM is disabled (FCKSM1 = 0):</u><br>Clock and PLL selections are never locked and may be modified by setting the OSWEN bit. |
| bit 6 | <b>IOLOCK:</b> I/O Lock Enable bit <sup>(2)</sup><br>1 = I/O lock is active<br>0 = I/O lock is not active   |
| bit 5 | <b>LOCK:</b> PLL Lock Status bit <sup>(3)</sup><br>1 = PLL module is in lock or PLL module start-up timer is satisfied<br>0 = PLL module is out of lock, PLL start-up timer is running or PLL is disabled   |
| bit 4 | <b>Unimplemented:</b> Read as '0'   |
| bit 3 | <b>CF:</b> Clock Fail Detect bit<br>1 = FSCM has detected a clock failure<br>0 = No clock failure has been detected   |
| bit 2 | <b>Unimplemented:</b> Read as '0'   |
| bit 1 | <b>SOSCEN:</b> 32 kHz Secondary Oscillator (SOSC) Enable bit<br>1 = Enable secondary oscillator<br>0 = Disable secondary oscillator   |
| bit 0 | <b>OSWEN:</b> Oscillator Switch Enable bit<br>1 = Initiate an oscillator switch to clock source specified by NOSC2:NOSC0 bits<br>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.

# PIC24FJ64GA004 FAMILY

## REGISTER 8-2: CLKDIV: CLOCK DIVIDER REGISTER

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

|       |     |     |     |     |     |     |       |
|-------|-----|-----|-----|-----|-----|-----|-------|
| U-0   | U-1 | 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    **DOZE2:DOZE0:** 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<sup>(1)</sup>  
 1 = DOZE2:DOZE0 bits specify the CPU peripheral clock ratio  
 0 = CPU peripheral clock ratio set to 1:1
- bit 10-8    **RCDIV2:RCDIV0:** 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      **Unimplemented:** Read as '0'
- bit 6      **Unimplemented:** Read as '1'
- bit 5-0    **Unimplemented:** Read as '0'

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

# PIC24FJ64GA004 FAMILY

**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 <sup>(1)</sup> | TUN4 <sup>(1)</sup> | TUN3 <sup>(1)</sup> | TUN2 <sup>(1)</sup> | TUN1 <sup>(1)</sup> | TUN0 <sup>(1)</sup> |
| bit 7 |     |                     |                     |                     |                     |                     | bit 0               |

**Legend:**

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

bit 15-6      **Unimplemented:** Read as '0'

bit 5-0      **TUN5:TUN0:** FRC Oscillator Tuning bits

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 TUN5:TUN0 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 FCKSM1 Configuration bit in Flash Configuration Word 2 must be programmed to '0'. (Refer to **Section 24.1 "Configuration Bits"** for further details.) If the FCKSM1 Configuration bit is unprogrammed ('1'), 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.

# PIC24FJ64GA004 FAMILY

## 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 SOSSEN 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
```

# PIC24FJ64GA004 FAMILY

---

## 8.4.3 SECONDARY OSCILLATOR LOW-POWER OPERATION

**Note:** This feature is implemented only on PIC24FJ64GA004 family devices with a major silicon revision level of B or later (DEVREV register value is 3042h or greater).

The Secondary Oscillator (SOSC) can operate in two distinct levels of power consumption based on device configuration. In Low-Power mode, the oscillator operates in a low-gain, low-power state. By default, the oscillator uses a higher gain setting, and therefore, requires more power. The Secondary Oscillator Mode Selection bits, SOSCSEL<1:0> (CW2<12:11>), determine the oscillator's power mode.

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

## 8.4.4 OSCILLATOR LAYOUT

On low pin count devices, such as those in the PIC24FJ64GA004 family, due to pinout limitations, the SOSC is more susceptible to noise than other PIC24F devices. Unless proper care is taken in the design and layout of the SOSC circuit, it is possible for inaccuracies to be introduced 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 detailed information on crystal circuit design, please refer to the "*PIC24F Family Reference Manual*", **Section 6. "Oscillator"** (DS39700) and Microchip Application Notes: *AN826, "Crystal Oscillator Basics and Crystal Selection for rPIC® and PICmicro® Devices"* (DS00826) and *AN849, "Basic PICmicro® Oscillator Design"* (DS00849).

## 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 10. Power-Saving Features” (DS39698). Additional power-saving tips can also be found in Appendix B: “Additional Guidance for PIC24FJ64GA004 Family Applications” of this document.

The PIC24FJ64GA004 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 and Idle 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. The assembly syntax of the PWRSAV instruction is shown in Example 9-1.

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”.

**Note:** SLEEP\_MODE and IDLE\_MODE are constants defined in the assembler include file for the selected device.

#### 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 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 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 the 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
```

# PIC24FJ64GA004 FAMILY

---

## 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 `PWRSV` 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 `PWRSV` instruction will be held off until entry into Sleep or Idle mode has completed. The device will then wake-up from Sleep or Idle mode.

## 9.3 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 `DOZE2:DOZE0` bits (`CLKDIV<14:12>`). There are eight possible configurations, from 1:1 to 1:256, 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. Power consumption is reduced, but not by as much as the PMD bit does. Most peripheral modules have an enable bit; exceptions include capture, 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.



# PIC24FJ64GA004 FAMILY

## 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  $V_{DD}$ ,  $V_{SS}$ ,  $\overline{MCLR}$  and  $OSC1/CLK1$ ) 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.

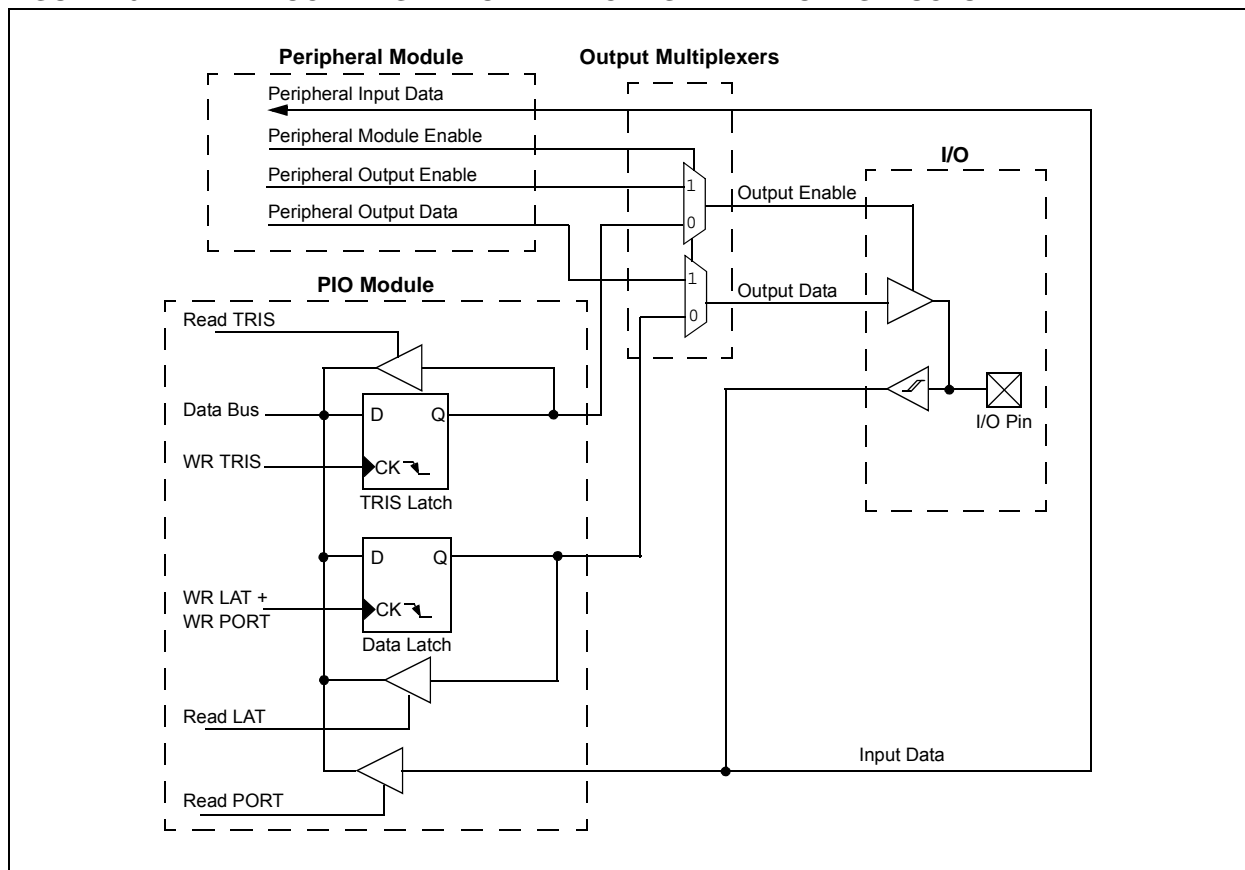
When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled. The I/O pin may be read, but the output driver for the parallel port bit will be disabled. If a peripheral is enabled, but the peripheral is not actively 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/O. The Data Direction register (TRISx) 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 (LATx), read the latch. Writes to the latch, write the latch. Reads from the port (PORTx), 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 LATx and TRISx 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, nevertheless, 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**



# PIC24FJ64GA004 FAMILY

## 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 use of the AD1PCFG and TRIS registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bit set (input). 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.

### 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 are always to be avoided. Table 10-1 summarizes the input capabilities. Refer to **Section 27.1 "DC Characteristics"** for more details.

### EXAMPLE 10-1: PORT WRITE/READ EXAMPLE

```
MOV    0xFF00, W0          ; Configure PORTB<15:8> as inputs
MOV    W0, TRISBB         ; and PORTB<7:0> as outputs
NOP                                ; Delay 1 cycle
BTSS   PORTB, #13         ; Next Instruction
```

TABLE 10-1: INPUT VOLTAGE LEVELS

| Port or Pin                | Tolerated Input | Description   |
|----------------------------|-----------------|---|
| PORTA<4:0>                 | VDD             | Only VDD input levels tolerated.                                  |
| PORTB<15:12>               |                 |   |
| PORTB<4:0>                 |                 |   |
| PORTC<2:0> <sup>(1)</sup>  |                 |   |
| PORTA<10:7> <sup>(1)</sup> | 5.5V            | Tolerates input levels above VDD, useful for most standard logic. |
| PORTB<11:5>                |                 |   |
| PORTC<9:3> <sup>(1)</sup>  |                 |   |

**Note 1:** Unavailable on 28-pin devices.

## 10.3 Input Change Notification

The input change notification function of the I/O ports allows the PIC24FJ64GA004 family of devices to generate interrupt requests to the processor in response to a change of state 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 22 external signals that may be selected (enabled) for generating an interrupt request on a change of state.

There are four control registers associated with the CN module. The CNEN1 and CNEN2 registers 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 also has a weak pull-up connected to it. The pull-ups act as a current source that is connected to the pin, and eliminate the need for external resistors when push button or keypad devices are connected. The pull-ups are enabled separately using the CNPU1 and CNPU2 registers, which contain the control bits for each of the CN pins. Setting any of the control bits enables the weak pull-ups for the corresponding pins.

When the internal pull-up is selected, the pin pulls up to VDD – 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

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. The challenge is even greater on low pin count devices similar to the PIC24FJ64GA family. In an application that needs to use more than one peripheral multiplexed on single pin, inconvenient workarounds 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 26 pins; the number of available pins is dependent on the particular device and its pincount. Pins that support the peripheral pin select feature include the designation "RPn" in their full pin designation, where "RP" designates a remappable peripheral and "n" is the remappable pin number. See Table 1-2 for 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.

The peripheral pin select module is not applied to I<sup>2</sup>C™, 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

When a pin selectable peripheral is active on a given I/O pin, it takes priority over all other digital I/O and digital communication peripherals associated with the pin. Priority is given regardless of the type of peripheral that is mapped. Pin select peripherals never take priority over any analog functions associated with the pin.

### 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 RPNRx registers are used to configure peripheral input mapping (see Register 10-1 through Register 10-14). Each register contains 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 5-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 selections supported by the device.

# PIC24FJ64GA004 FAMILY

**TABLE 10-2: SELECTABLE INPUT SOURCES (MAPS INPUT TO FUNCTION)<sup>(1)</sup>**

| Input Name              | Function Name      | Register | Configuration Bits |
|-------------------------|--------------------|----------|--------------------|
| External Interrupt 1    | INT1               | RPINR0   | INTR1<4:0>         |
| External Interrupt 2    | INT2               | RPINR1   | INTR2R<4:0>        |
| Timer2 External Clock   | T2CK               | RPINR3   | T2CKR<4:0>         |
| Timer3 External Clock   | T3CK               | RPINR3   | T3CKR<4:0>         |
| Timer4 External Clock   | T4CK               | RPINR4   | T4CKR<4:0>         |
| Timer5 External Clock   | T5CK               | RPINR4   | T5CKR<4:0>         |
| Input Capture 1         | IC1                | RPINR7   | IC1R<4:0>          |
| Input Capture 2         | IC2                | RPINR7   | IC2R<4:0>          |
| Input Capture 3         | IC3                | RPINR8   | IC3R<4:0>          |
| Input Capture 4         | IC4                | RPINR8   | IC4R<4:0>          |
| Input Capture 5         | IC5                | RPINR9   | IC5R<4:0>          |
| Output Compare Fault A  | OCFA               | RPINR11  | OCFAR<4:0>         |
| Output Compare Fault B  | OCFB               | RPINR11  | OCFBR<4:0>         |
| UART1 Receive           | U1RX               | RPINR18  | U1RXR<4:0>         |
| UART1 Clear To Send     | $\overline{U1CTS}$ | RPINR18  | U1CTSR<4:0>        |
| UART2 Receive           | U2RX               | RPINR19  | U2RXR<4:0>         |
| UART2 Clear To Send     | $\overline{U2CTS}$ | RPINR19  | U2CTSR<4:0>        |
| SPI1 Data Input         | SDI1               | RPINR20  | SDI1R<4:0>         |
| SPI1 Clock Input        | SCK1IN             | RPINR20  | SCK1R<4:0>         |
| SPI1 Slave Select Input | SS1IN              | RPINR21  | SS1R<4:0>          |
| SPI2 Data Input         | SDI2               | RPINR22  | SDI2R<4:0>         |
| SPI2 Clock Input        | SCK2IN             | RPINR22  | SCK2R<4:0>         |
| SPI2 Slave Select Input | SS2IN              | RPINR23  | SS2R<4:0>          |

**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. Like the RPINRx registers, each register contains two 5-bit fields; 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 '00000'. This permits any given pin to remain disconnected from the output of any of the pin selectable peripherals.

# PIC24FJ64GA004 FAMILY

**TABLE 10-3: SELECTABLE OUTPUT SOURCES (MAPS FUNCTION TO OUTPUT)**

| Function             | Output Function Number <sup>(1)</sup> | Output Name              |
|----------------------|---------------------------------------|--------------------------|
| NULL <sup>(2)</sup>  | 0                                     | NULL                     |
| C1OUT                | 1                                     | Comparator 1 Output      |
| C2OUT                | 2                                     | Comparator 2 Output      |
| U1TX                 | 3                                     | UART1 Transmit           |
| U1RTS <sup>(3)</sup> | 4                                     | UART1 Request To Send    |
| U2TX                 | 5                                     | UART2 Transmit           |
| U2RTS <sup>(3)</sup> | 6                                     | UART2 Request To Send    |
| SDO1                 | 7                                     | SPI1 Data Output         |
| SCK1OUT              | 8                                     | SPI1 Clock Output        |
| SS1OUT               | 9                                     | SPI1 Slave Select Output |
| SDO2                 | 10                                    | SPI2 Data Output         |
| SCK2OUT              | 11                                    | SPI2 Clock Output        |
| SS2OUT               | 12                                    | SPI2 Slave Select Output |
| OC1                  | 18                                    | Output Compare 1         |
| OC2                  | 19                                    | Output Compare 2         |
| OC3                  | 20                                    | Output Compare 3         |
| OC4                  | 21                                    | Output Compare 4         |
| OC5                  | 22                                    | Output Compare 5         |

- Note 1:** Value assigned to the RPn<4:0> pins corresponds to the peripheral output function number.
- 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.

## 10.4.4 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.

# PIC24FJ64GA004 FAMILY

## 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 RP31 and all peripheral pin select outputs are disconnected.

**Note:** In tying peripheral pin select inputs to RP31, RP31 does not have to exist on a device for the registers to be reset 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 inline 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,  $\overline{U1CTS}$
- Output Functions: U1TX,  $\overline{U1RTS}$

### EXAMPLE 10-2: CONFIGURING UART1 INPUT AND OUTPUT FUNCTIONS

```

//*****
// Unlock Registers
//*****
asm volatile ( "MOV    #OSCCON, w1  \n"
              "MOV    #0x46, w2    \n"
              "MOV    #0x57, w3    \n"
              "MOV.b  w2, [w1]     \n"
              "MOV.b  w3, [w1]     \n"
              "BCLR  OSCCON, #6" );

//*****
// Configure Input Functions
// (See Table 10-2)
//*****
//*****
// Assign U1RX To Pin RP0
//*****
RPINR18bits.U1RXR = 0;

//*****
// Assign U1CTS To Pin RP1
//*****
RPINR18bits.U1CTSR = 1;

//*****
// Configure Output Functions
// (See Table 10-3)
//*****
//*****
// Assign U1TX To Pin RP2
//*****
RPOR1bits.RP2R = 3;

//*****
// Assign U1RTS To Pin RP3
//*****
RPOR1bits.RP3R = 4;

//*****
// Lock Registers
//*****
asm volatile ( "MOV    #OSCCON, w1  \n"
              "MOV    #0x46, w2    \n"
              "MOV    #0x57, w3    \n"
              "MOV.b  w2, [w1]     \n"
              "MOV.b  w3, [w1]     \n"
              "BSET  OSCCON, #6" );
```

# PIC24FJ64GA004 FAMILY

## 10.5 Peripheral Pin Select Registers

The PIC24FJ64GA004 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 `OSCCON<IOLOCK> = 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     **INT1R4:INT1R0:** Assign External Interrupt 1 (INT1) to the Corresponding RPn 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     **INT2R4:INT2R0:** Assign External Interrupt 2 (INT2) to the Corresponding RPn Pin bits

# PIC24FJ64GA004 FAMILY

## 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      **T3CKR4:T3CKR0:** Assign Timer3 External Clock (T3CK) to the Corresponding RPn Pin bits
- bit 7-5        **Unimplemented:** Read as '0'
- bit 4-0        **T2CKR4:T2CKR0:** Assign Timer2 External Clock (T2CK) to the Corresponding RPn 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      **T5CKR4:T5CKR0:** Assign Timer5 External Clock (T5CK) to the Corresponding RPn Pin bits
- bit 7-5        **Unimplemented:** Read as '0'
- bit 4-0        **T4CKR4:T4CKR0:** Assign Timer4 External Clock (T4CK) to the Corresponding RPn Pin bits



# PIC24FJ64GA004 FAMILY

## 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      **IC2R4:IC2R0:** Assign Input Capture 2 (IC2) to the Corresponding RPn Pin bits
- bit 7-5        **Unimplemented:** Read as '0'
- bit 4-0        **IC1R4:IC1R0:** Assign Input Capture 1 (IC1) to the Corresponding RPn 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      **IC4R4:IC4R0:** Assign Input Capture 4 (IC4) to the Corresponding RPn Pin bits
- bit 7-5        **Unimplemented:** Read as '0'
- bit 4-0        **IC3R4:IC3R0:** Assign Input Capture 3 (IC3) to the Corresponding RPn Pin bits

# PIC24FJ64GA004 FAMILY

## 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      **IC5R4:IC5R0:** Assign Input Capture 5 (IC5) to the Corresponding RPn 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      **OCFBR4:OCFBR0:** Assign Output Compare Fault B (OCFB) to the Corresponding RPn Pin bits  
 bit 7-5      **Unimplemented:** Read as '0'  
 bit 4-0      **OCFAR4:OCFAR0:** Assign Output Compare Fault A (OCFA) to the Corresponding RPn Pin bits

# PIC24FJ64GA004 FAMILY

## 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      **U1CTSR4:U1CTSR0:** Assign UART1 Clear to Send ( $\overline{\text{U1CTS}}$ ) to the Corresponding RPn Pin bits
- bit 7-5        **Unimplemented:** Read as '0'
- bit 4-0        **U1RXR4:U1RXR0:** Assign UART1 Receive (U1RX) to the Corresponding RPn 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      **U2CTSR4:U2CTSR0:** Assign UART2 Clear to Send ( $\overline{\text{U2CTS}}$ ) to the Corresponding RPn Pin bits
- bit 7-5        **Unimplemented:** Read as '0'
- bit 4-0        **U2RXR4:U2RXR0:** Assign UART2 Receive (U2RX) to the Corresponding RPn Pin bits

# PIC24FJ64GA004 FAMILY

## 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      **SCK1R4:SCK1R0:** Assign SPI1 Clock Input (SCK1IN) to the Corresponding RPn Pin bits
- bit 7-5      **Unimplemented:** Read as '0'
- bit 4-0      **SDI1R4:SDI1R0:** Assign SPI1 Data Input (SDI1) to the Corresponding RPn 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      **SS1R4:SS1R0:** Assign SPI1 Slave Select Input (SS1IN) to the Corresponding RPn Pin bits

# PIC24FJ64GA004 FAMILY

## 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      **SCK2R4:SCK2R0:** Assign SPI2 Clock Input (SCK2IN) to the Corresponding RPn Pin bits
- bit 7-5      **Unimplemented:** Read as '0'
- bit 4-0      **SDI2R4:SDI2R0:** Assign SPI2 Data Input (SDI2) to the Corresponding RPn 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      **SS2R4:SS2R0:** Assign SPI2 Slave Select Input (SS2IN) to the Corresponding RPn Pin bits

# PIC24FJ64GA004 FAMILY

## 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      **RP1R4:RP1R0:** Peripheral Output Function is Assigned to RP1 Output Pin bits  
 (see Table 10-3 for peripheral function numbers)
- bit 7-5        **Unimplemented:** Read as '0'
- bit 4-0        **RP0R4:RP0R0:** Peripheral Output Function is Assigned to RP0 Output Pin bits  
 (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      **RP3R4:RP3R0:** Peripheral Output Function is Assigned to RP3 Output Pin bits  
 (see Table 10-3 for peripheral function numbers)
- bit 7-5        **Unimplemented:** Read as '0'
- bit 4-0        **RP2R4:RP2R0:** Peripheral Output Function is Assigned to RP2 Output Pin bits  
 (see Table 10-3 for peripheral function numbers)

# PIC24FJ64GA004 FAMILY

## 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     **RP5R4:RP5R0:** Peripheral Output Function is Assigned to RP5 Output Pin bits  
(see Table 10-3 for peripheral function numbers)
- bit 7-5      **Unimplemented:** Read as '0'
- bit 4-0      **RP4R4:RP4R0:** Peripheral Output Function is Assigned to RP4 Output Pin bits  
(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     **RP7R4:RP7R0:** Peripheral Output Function is Assigned to RP7 Output Pin bits  
(see Table 10-3 for peripheral function numbers)
- bit 7-5      **Unimplemented:** Read as '0'
- bit 4-0      **RP6R4:RP6R0:** Peripheral Output Function is Assigned to RP6 Output Pin bits  
(see Table 10-3 for peripheral function numbers)

# PIC24FJ64GA004 FAMILY

## 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      **RP9R4:RP9R0:** Peripheral Output Function is Assigned to RP9 Output Pin bits  
 (see Table 10-3 for peripheral function numbers)
- bit 7-5        **Unimplemented:** Read as '0'
- bit 4-0        **RP8R4:RP8R0:** Peripheral Output Function is Assigned to RP8 Output Pin bits  
 (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      **RP11R4:RP11R0:** Peripheral Output Function is Assigned to RP11 Output Pin bits  
 (see Table 10-3 for peripheral function numbers)
- bit 7-5        **Unimplemented:** Read as '0'
- bit 4-0        **RP10R4:RP10R0:** Peripheral Output Function is Assigned to RP10 Output Pin bits  
 (see Table 10-3 for peripheral function numbers)



# PIC24FJ64GA004 FAMILY

## 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     **RP13R4:RP13R0:** Peripheral Output Function is Assigned to RP13 Output Pin bits  
(see Table 10-3 for peripheral function numbers)
- bit 7-5      **Unimplemented:** Read as '0'
- bit 4-0      **RP12R4:RP12R0:** Peripheral Output Function is Assigned to RP12 Output Pin bits  
(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     **RP15R4:RP15R0:** Peripheral Output Function is Assigned to RP15 Output Pin bits  
(see Table 10-3 for peripheral function numbers)
- bit 7-5      **Unimplemented:** Read as '0'
- bit 4-0      **RP14R4:RP14R0:** Peripheral Output Function is Assigned to RP14 Output Pin bits  
(see Table 10-3 for peripheral function numbers)

# PIC24FJ64GA004 FAMILY

## REGISTER 10-23: RPOR8: PERIPHERAL PIN SELECT OUTPUT REGISTER 8

|        |     |     |                       |                       |                       |                       |                       |
|--------|-----|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| U-0    | U-0 | U-0 | R/W-0                 | R/W-0                 | R/W-0                 | R/W-0                 | R/W-0                 |
| —      | —   | —   | RP17R4 <sup>(1)</sup> | RP17R3 <sup>(1)</sup> | RP17R2 <sup>(1)</sup> | RP17R1 <sup>(1)</sup> | RP17R0 <sup>(1)</sup> |
| 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 <sup>(1)</sup> | RP16R3 <sup>(1)</sup> | RP16R2 <sup>(1)</sup> | RP16R1 <sup>(1)</sup> | RP16R0 <sup>(1)</sup> |
| bit 7 |     |     |                       |                       |                       |                       | bit 0                 |

### Legend:

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

- bit 15-13      **Unimplemented:** Read as '0'
- bit 12-8      **RP17R4:RP17R0:** Peripheral Output Function is Assigned to RP17 Output Pin bits<sup>(1)</sup>  
                   (see Table 10-3 for peripheral function numbers)
- bit 7-5        **Unimplemented:** Read as '0'
- bit 4-0        **RP16R4:RP16R0:** Peripheral Output Function is Assigned to RP16 Output Pin bits<sup>(1)</sup>  
                   (see Table 10-3 for peripheral function numbers)

**Note 1:** Bits are only available on the 44-pin devices; otherwise, they read as '0'.

## REGISTER 10-24: RPOR9: PERIPHERAL PIN SELECT OUTPUT REGISTER 9

|        |     |     |        |        |        |        |        |
|--------|-----|-----|--------|--------|--------|--------|--------|
| 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      **RP19R4:RP19R0:** Peripheral Output Function is Assigned to RP19 Output Pin bits<sup>(1)</sup>  
                   (see Table 10-3 for peripheral function numbers)
- bit 7-5        **Unimplemented:** Read as '0'
- bit 4-0        **RP18R4:RP18R0:** Peripheral Output Function is Assigned to RP18 Output Pin bits<sup>(1)</sup>  
                   (see Table 10-3 for peripheral function numbers)

**Note 1:** Bits are only available on the 44-pin devices; otherwise, they read as '0'.

# PIC24FJ64GA004 FAMILY

## REGISTER 10-25: RPOR10: PERIPHERAL PIN SELECT OUTPUT REGISTER 10

|        |     |     |                       |                       |                       |                       |                       |
|--------|-----|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| U-0    | U-0 | U-0 | R/W-0                 | R/W-0                 | R/W-0                 | R/W-0                 | R/W-0                 |
| —      | —   | —   | RP21R4 <sup>(1)</sup> | RP21R3 <sup>(1)</sup> | RP21R2 <sup>(1)</sup> | RP21R1 <sup>(1)</sup> | RP21R0 <sup>(1)</sup> |
| 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 <sup>(1)</sup> | RP20R3 <sup>(1)</sup> | RP20R2 <sup>(1)</sup> | RP20R1 <sup>(1)</sup> | RP20R0 <sup>(1)</sup> |
| bit 7 |     |     |                       |                       |                       |                       | bit 0                 |

### Legend:

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

- bit 15-13     **Unimplemented:** Read as '0'
- bit 12-8     **RP21R4:RP21R0:** Peripheral Output Function is Assigned to RP21 Output Pin bits<sup>(1)</sup>  
(see Table 10-3 for peripheral function numbers)
- bit 7-5      **Unimplemented:** Read as '0'
- bit 4-0      **RP20R4:RP20R0:** Peripheral Output Function is Assigned to RP20 Output Pin bits<sup>(1)</sup>  
(see Table 10-3 for peripheral function numbers)

**Note 1:** Bits are only available on the 44-pin devices; otherwise, they read as '0'.

## REGISTER 10-26: RPOR11: PERIPHERAL PIN SELECT OUTPUT REGISTER 11

|        |     |     |                       |                       |                       |                       |                       |
|--------|-----|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| U-0    | U-0 | U-0 | R/W-0                 | R/W-0                 | R/W-0                 | R/W-0                 | R/W-0                 |
| —      | —   | —   | RP23R4 <sup>(1)</sup> | RP23R3 <sup>(1)</sup> | RP23R2 <sup>(1)</sup> | RP23R1 <sup>(1)</sup> | RP23R0 <sup>(1)</sup> |
| 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 <sup>(1)</sup> | RP22R3 <sup>(1)</sup> | RP22R2 <sup>(1)</sup> | RP22R1 <sup>(1)</sup> | RP22R0 <sup>(1)</sup> |
| bit 7 |     |     |                       |                       |                       |                       | bit 0                 |

### Legend:

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

- bit 15-13     **Unimplemented:** Read as '0'
- bit 12-8     **RP23R4:RP23R0:** Peripheral Output Function is Assigned to RP23 Output Pin bits<sup>(1)</sup>  
(see Table 10-3 for peripheral function numbers)
- bit 7-5      **Unimplemented:** Read as '0'
- bit 4-0      **RP22R4:RP22R0:** Peripheral Output Function is Assigned to RP22 Output Pin bits<sup>(1)</sup>  
(see Table 10-3 for peripheral function numbers)

**Note 1:** Bits are only available on the 44-pin devices; otherwise, they read as '0'.

# PIC24FJ64GA004 FAMILY

## REGISTER 10-27: RPOR12: PERIPHERAL PIN SELECT OUTPUT REGISTER 12

|        |     |     |                       |                       |                       |                       |                       |
|--------|-----|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| U-0    | U-0 | U-0 | R/W-0                 | R/W-0                 | R/W-0                 | R/W-0                 | R/W-0                 |
| —      | —   | —   | RP25R4 <sup>(1)</sup> | RP25R3 <sup>(1)</sup> | RP25R2 <sup>(1)</sup> | RP25R1 <sup>(1)</sup> | RP25R0 <sup>(1)</sup> |
| 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 <sup>(1)</sup> | RP24R3 <sup>(1)</sup> | RP24R2 <sup>(1)</sup> | RP24R1 <sup>(1)</sup> | RP24R0 <sup>(1)</sup> |
| bit 7 |     |     |                       |                       |                       |                       | bit 0                 |

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 15-13 **Unimplemented:** Read as '0'

bit 12-8 **RP25R4:RP25R0:** Peripheral Output Function is Assigned to RP25 Output Pin bits<sup>(1)</sup>  
(see Table 10-3 for peripheral function numbers)

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

bit 4-0 **RP24R4:RP24R0:** Peripheral Output Function is Assigned to RP24 Output Pin bits<sup>(1)</sup>  
(see Table 10-3 for peripheral function numbers)

**Note 1:** Bits are only available on the 44-pin devices; otherwise, they read as '0'.

# PIC24FJ64GA004 FAMILY

## 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:

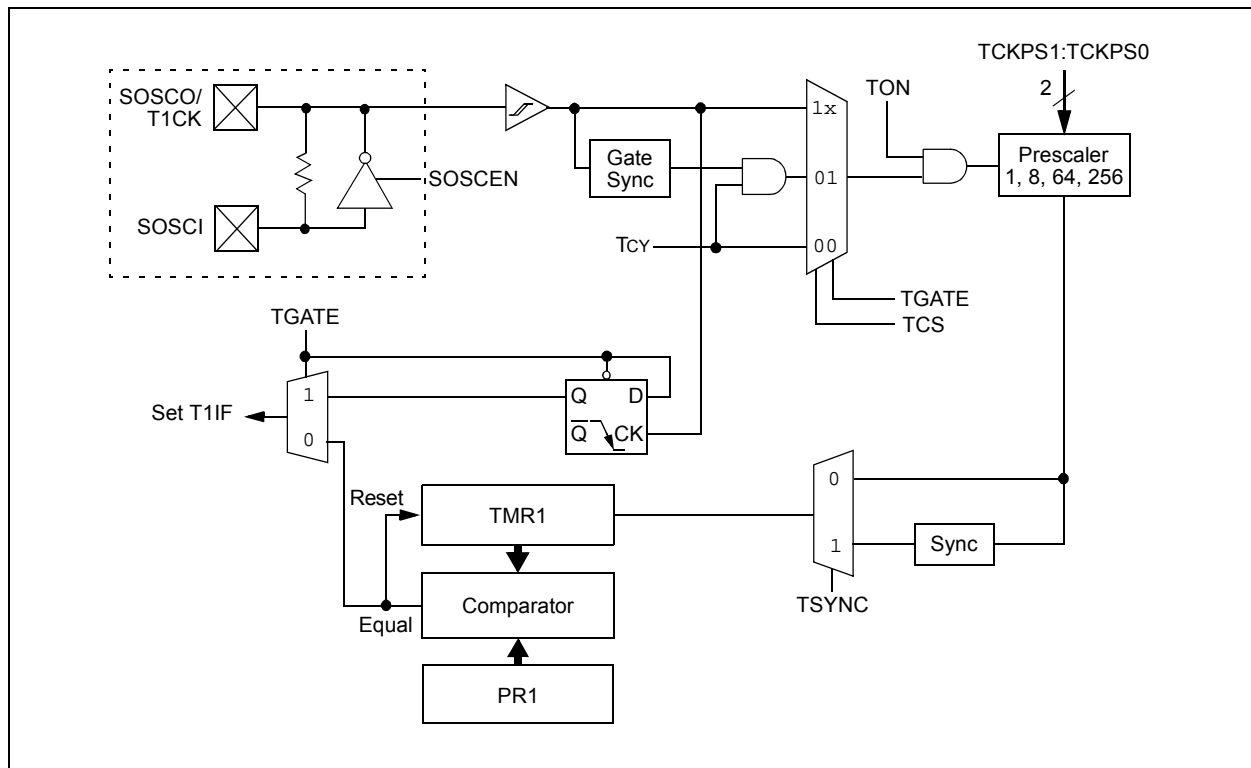
- Timer Gate Operation
- Selectable Prescaler Settings
- Timer Operation during CPU Idle and Sleep modes
- Interrupt on 16-Bit Period Register Match or Falling Edge of External Gate Signal

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 TCKPS1:TCKPS0 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, T1IP2:T1IP0, to set the interrupt priority.

**FIGURE 11-1: 16-BIT TIMER1 MODULE BLOCK DIAGRAM**



# PIC24FJ64GA004 FAMILY

## REGISTER 11-1: T1CON: TIMER1 CONTROL REGISTER

|        |     |       |     |     |     |       |     |
|--------|-----|-------|-----|-----|-----|-------|-----|
| 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     **TCKPS1:TCKPS0:** 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'

## 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 a 32-bit timer, Timer2/3 and Timer4/5 operate in three modes:

- Two independent 16-bit timers (Timer2 and Timer3) 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 TCKPS1:TCKPS0 bits.
3. Set the Clock and Gating modes using the TCS and TGATE bits. If TCS is set to external clock, RPINRx (TxCK) must be configured to an available RPn pin. See **Section 10.4 “Peripheral Pin Select”** 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, T3IP2:T3IP0 or T5IP2:T5IP0, 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).

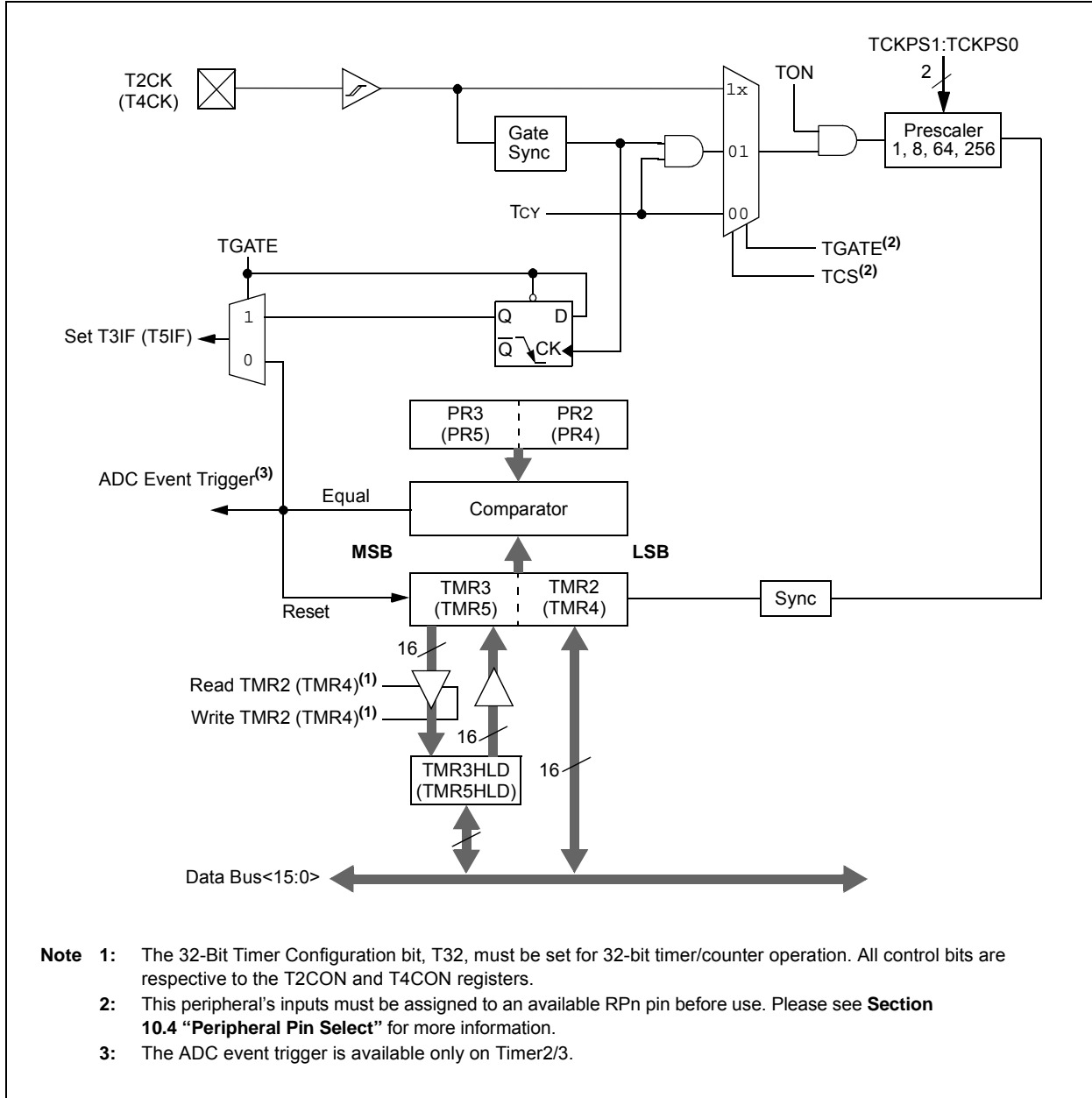
The timer value, at any point, is stored in the register pair, TMR3:TMR2 (or TMR5:TMR4). TMR3 (TMR5) always contains the most significant word of the count, while TMR2 (TMR4) contains the least significant word.

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 TCKPS1:TCKPS0 bits.
3. Set the Clock and Gating modes using the TCS and TGATE bits. See **Section 10.4 “Peripheral Pin Select”** 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, TxIP2:TxIP0, to set the interrupt priority.
6. Set the TON bit (TxCON<15> = 1).

# PIC24FJ64GA004 FAMILY

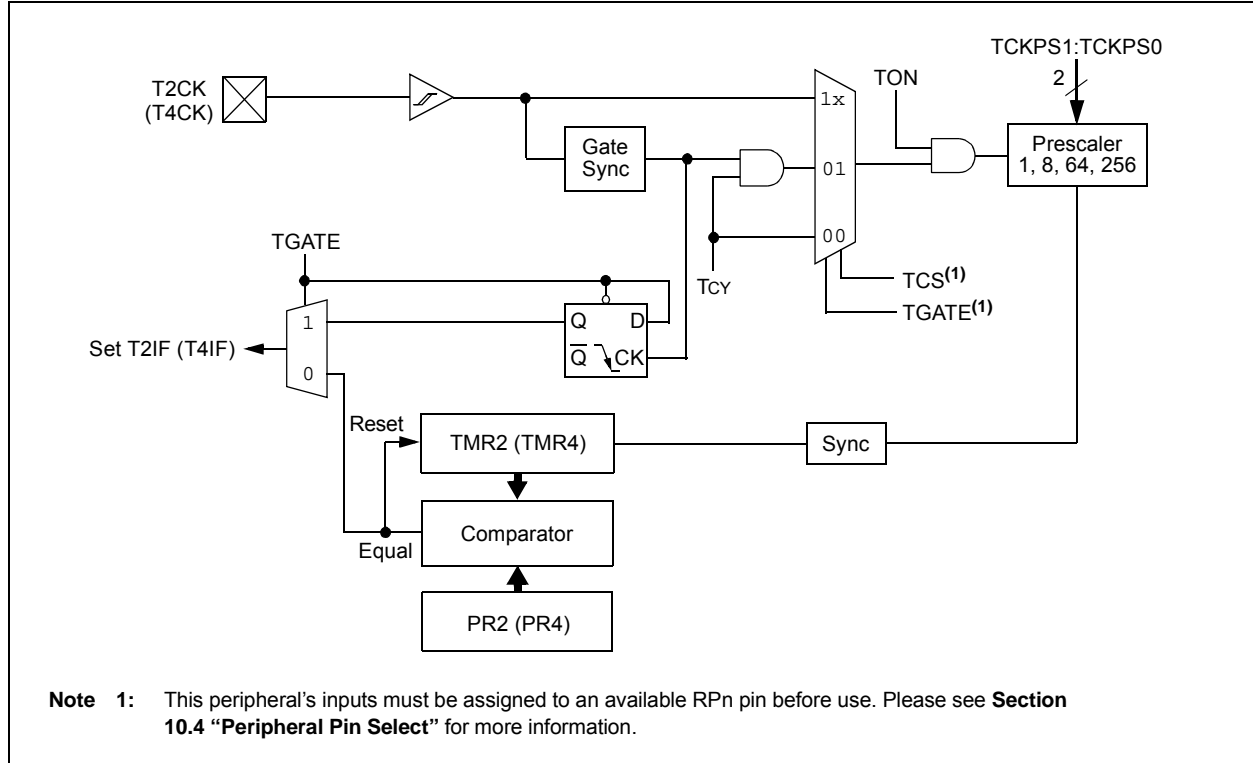
**FIGURE 12-1: TIMER2/3 AND TIMER4/5 (32-BIT) BLOCK DIAGRAM**



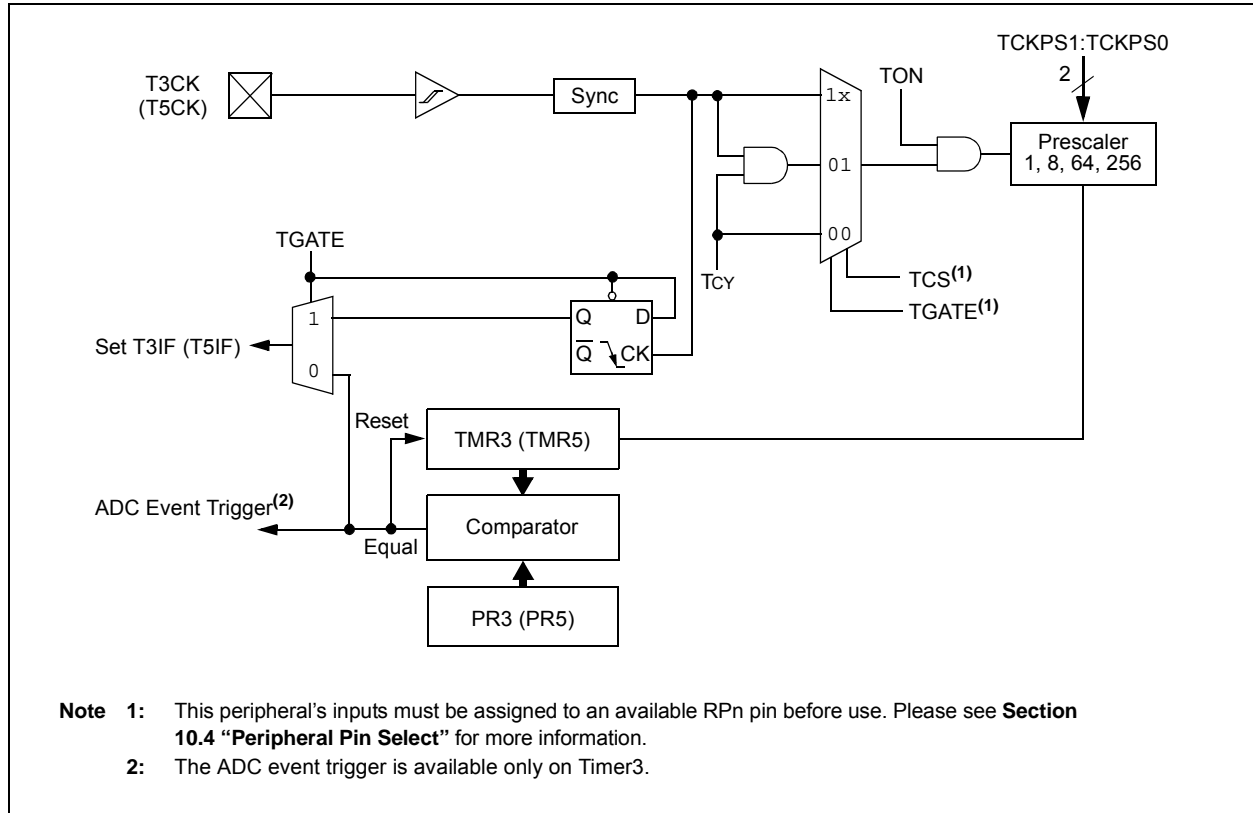


# PIC24FJ64GA004 FAMILY

**FIGURE 12-2: TIMER2 AND TIMER4 (16-BIT SYNCHRONOUS) BLOCK DIAGRAM**



**FIGURE 12-3: TIMER3 AND TIMER5 (16-BIT SYNCHRONOUS) BLOCK DIAGRAM**



# PIC24FJ64GA004 FAMILY

**REGISTER 12-1: TxCON: TIMER2 AND TIMER4 CONTROL REGISTER**

|        |     |       |     |     |     |     |       |
|--------|-----|-------|-----|-----|-----|-----|-------|
| 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 <sup>(1)</sup> | —   | TCS <sup>(2)</sup> | —     |
| bit 7 |       |        |        |                    |     |                    | bit 0 |

**Legend:**

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

- bit 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 enabled  
                   0 = Gated time accumulation disabled
- bit 5-4        **TCKPS1:TCKPS0:** 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<sup>(1)</sup>  
                   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<sup>(2)</sup>  
                   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.  
**Note 2:** If TCS = 1, RPINRx (TxCK) must be configured to an available RPN pin. For more information, see **Section 10.4 "Peripheral Pin Select"**.

# PIC24FJ64GA004 FAMILY

## REGISTER 12-2: TyCON: TIMER3 AND TIMER5 CONTROL REGISTER

|                    |     |                      |     |     |     |       |     |
|--------------------|-----|----------------------|-----|-----|-----|-------|-----|
| R/W-0              | U-0 | R/W-0                | U-0 | U-0 | U-0 | U-0   | U-0 |
| TON <sup>(1)</sup> | —   | TSIDL <sup>(1)</sup> | —   | —   | —   | —     | —   |
| 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 <sup>(1)</sup> | TCKPS1 <sup>(1)</sup> | TCKPS0 <sup>(1)</sup> | —   | —   | TCS <sup>(1,2)</sup> | —   |
| bit 7 |                      |                       |                       |     |     | bit 0                |     |

### Legend:

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

- bit 15        **TON:** Timery On bit<sup>(1)</sup>  
                  1 = Starts 16-bit Timery  
                  0 = Stops 16-bit Timery
- bit 14        **Unimplemented:** Read as '0'
- bit 13        **TSIDL:** Stop in Idle Mode bit<sup>(1)</sup>  
                  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<sup>(1)</sup>  
                  When TCS = 1:  
                  This bit is ignored.  
                  When TCS = 0:  
                  1 = Gated time accumulation enabled  
                  0 = Gated time accumulation disabled
- bit 5-4     **TCKPS1:TCKPS0:** Timery Input Clock Prescale Select bits<sup>(1)</sup>  
                  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<sup>(1,2)</sup>  
                  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”** for more information.

# PIC24FJ64GA004 FAMILY

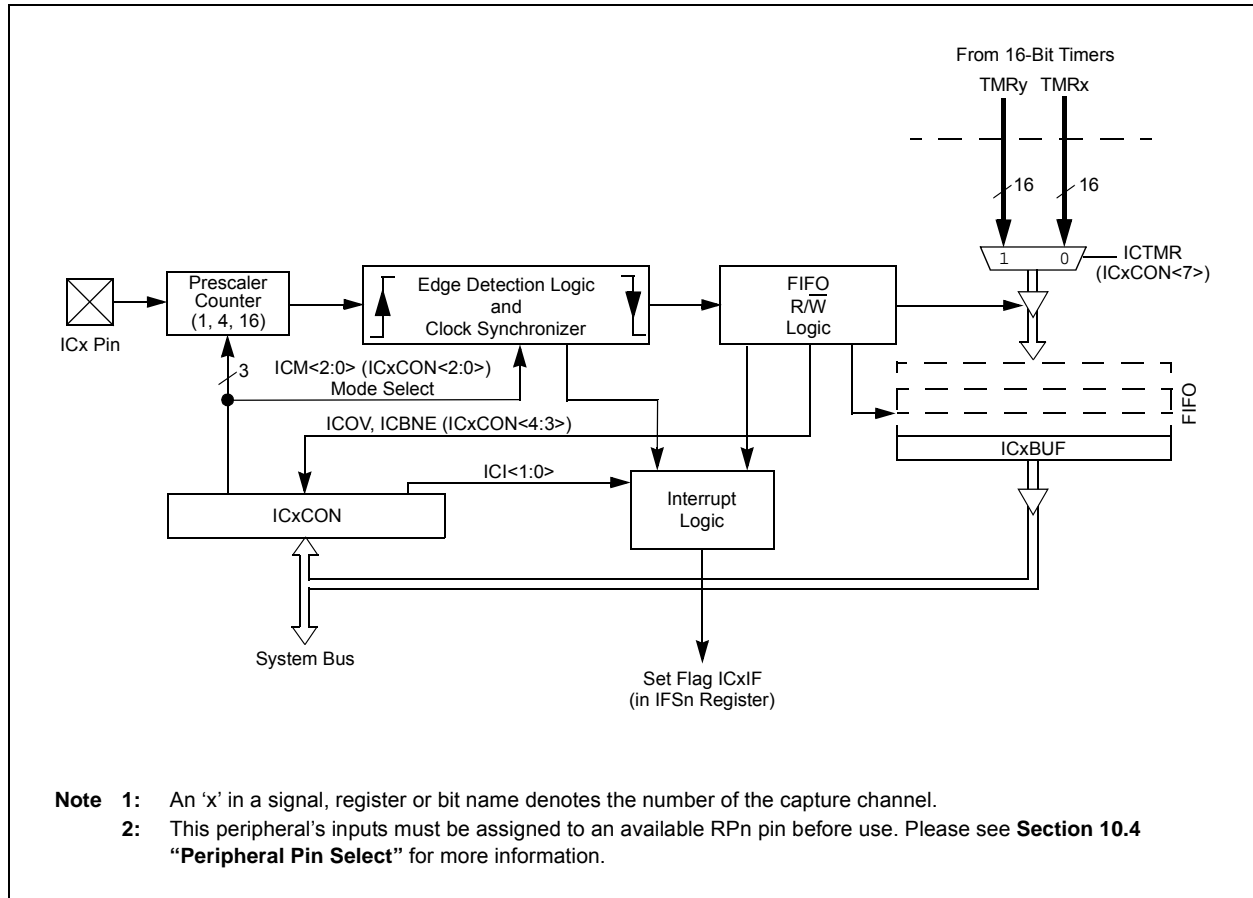
---

NOTES:

## 13.0 INPUT CAPTURE

**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 15. Input Capture” (DS39701).

**FIGURE 13-1: INPUT CAPTURE BLOCK DIAGRAM**



# PIC24FJ64GA004 FAMILY

## 13.1 Input Capture Registers

**REGISTER 13-1: ICxCON: INPUT CAPTURE x CONTROL REGISTER**

|        |       |        |         |         |                     |                     |                     |
|--------|-------|--------|---------|---------|---------------------|---------------------|---------------------|
| U-0    | U-0   | R/W-0  | U-0     | U-0     | U-0                 | U-0                 | U-0                 |
| —      | —     | ICSIDL | —       | —       | —                   | —                   | —                   |
| bit 15 |       |        |         |         |                     |                     | bit 8               |
| R/W-0  | R/W-0 | R/W-0  | R-0, HC | R-0, HC | R/W-0               | R/W-0               | R/W-0               |
| ICTMR  | ICI1  | ICIO   | ICOV    | ICBNE   | ICM2 <sup>(1)</sup> | ICM1 <sup>(1)</sup> | ICM0 <sup>(1)</sup> |
| bit 7  |       |        |         |         |                     |                     | bit 0               |

|                   |                                    |
|-------------------|------------------------------------|
| <b>Legend:</b>    | HC = Hardware Clearable bit        |
| R = Readable bit  | W = Writable bit                   |
| -n = Value at POR | '1' = Bit is set                   |
|                   | U = Unimplemented bit, read as '0' |
|                   | '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 will halt in CPU Idle mode  
               0 = Input capture module will continue to operate in CPU Idle mode
- bit 12-8     **Unimplemented:** Read as '0'
- bit 7         **ICTMR:** Input Capture x Timer Select bit  
               1 = TMR2 contents are captured on capture event  
               0 = TMR3 contents are captured on capture event
- bit 6-5      **ICI1:ICIO:** 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      **ICM2:ICM0:** Input Capture x Mode Select bits<sup>(1)</sup>  
               111 = 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 = Capture mode, every 16th rising edge  
               100 = Capture mode, every 4th rising edge  
               011 = Capture mode, every rising edge  
               010 = Capture mode, every falling edge  
               001 = Capture mode, 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:** RPNRx (ICxRx) must be configured to an available RPN pin. For more information, see **Section 10.4** "Peripheral Pin Select".

## 14.0 OUTPUT COMPARE

**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 16. Output Compare** (DS39706).

### 14.1 Setup for Single Output Pulse Generation

When the OCM control bits (OCxCON<2:0>) are set to ‘100’, the selected output compare channel initializes the OCx pin to the low state and generates a single output pulse.

To generate a single output pulse, the following steps are required (these steps assume the timer source is initially turned off, but this is not a requirement for the module operation):

1. 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.
2. Calculate time to the rising edge of the output pulse relative to the TMRy start value (0000h).
3. 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.
4. Write the values computed in steps 2 and 3 above into the Output Compare x register, OCxR, and the Output Compare x Secondary register, OCxRS, respectively.
5. Set Timer Period register, PRy, to value equal to or greater than value in OCxRS, the Output Compare x Secondary register.
6. Set the OCM bits to ‘100’ and the OCTSEL (OCxCON<3>) bit to the desired timer source. The OCx pin state will now be driven low.
7. Set the TON (TyCON<15>) bit to ‘1’, which enables the compare time base to count.
8. Upon the first match between TMRy and OCxR, the OCx pin will be driven high.
9. When the incrementing timer, TMRy, matches the Output Compare x Secondary register, OCxRS, the second and trailing edge (high-to-low) of the pulse is driven onto the OCx pin. No additional pulses are driven onto the OCx pin and it remains at low. As a result of the second compare match event, the OCxIF interrupt flag bit is set, which will result in an interrupt if it is enabled, by setting the OCxIE bit. For further information on peripheral interrupts, refer to **Section 7.0 “Interrupt Controller”**.

10. To initiate another single pulse output, change the Timer and Compare register settings, if needed, and then issue a write to set the OCM bits to ‘100’. Disabling and re-enabling of the timer and clearing the TMRy register are not required, but may be advantageous for defining a pulse from a known event time boundary.

The output compare module does not have to be disabled after the falling edge of the output pulse. Another pulse can be initiated by rewriting the value of the OCxCON register.

### 14.2 Setup for Continuous Output Pulse Generation

When the OCM control bits (OCxCON<2:0>) are set to ‘101’, the selected output compare channel initializes the OCx pin to the low state and generates output pulses on each and every compare match event.

For the user to configure the module for the generation of a continuous stream of output pulses, the following steps are required (these steps assume the timer source is initially turned off, but this is not a requirement for the module operation):

1. 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.
2. Calculate time to the rising edge of the output pulse relative to the TMRy start value (0000h).
3. 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.
4. Write the values computed in step 2 and 3 above into the Output Compare x register, OCxR, and the Output Compare x Secondary register, OCxRS, respectively.
5. Set Timer Period register, PRy, to value equal to or greater than value in OCxRS.
6. Set the OCM bits to ‘101’ and the OCTSEL bit to the desired timer source. The OCx pin state will now be driven low.
7. Enable the compare time base by setting the TON (TyCON<15>) bit to ‘1’.
8. Upon the first match between TMRy and OCxR, the OCx pin will be driven high.
9. When the compare time base, TMRy, matches the OCxRS, the second and trailing edge (high-to-low) of the pulse is driven onto the OCx pin.
10. As a result of the second compare match event, the OCxIF interrupt flag bit set.
11. When the compare time base and the value in its respective Timer Period register match, the TMRy register resets to 0x0000 and resumes counting.
12. Steps 8 through 11 are repeated and a continuous stream of pulses is generated indefinitely. The OCxIF flag is set on each OCxRS/TMRy compare match event.

# PIC24FJ64GA004 FAMILY

## 14.3 Pulse-Width Modulation Mode

**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”** for more information.

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

1. Set the PWM period by writing to the selected Timer Period register (PRy).
2. Set the PWM duty cycle by writing to the OCxRS register.
3. Write the OCxR register with the initial duty cycle.
4. Enable interrupts, if required, for the timer and output compare modules. The output compare interrupt is required for PWM Fault pin utilization.
5. Configure the output compare module for one of two PWM Operation modes by writing to the Output Compare Mode bits, OCM<2:0> (OCxCON<2:0>).
6. Set the TMRy prescale value and enable the time base by setting TON (TxCON<15>) = 1.

**Note:** The OCxR register should be initialized before the output compare module is first enabled. The OCxR register becomes a Read-Only Duty Cycle register when the module is operated in the PWM modes. The value held in OCxR will become the PWM duty cycle for the first PWM period. The contents of the Output Compare x Secondary register, OCxRS, will not be transferred into OCxR until a time base period match occurs.

### 14.3.1 PWM PERIOD

The PWM period is specified by writing to PRy, the Timer Period register. The PWM period can be calculated using Equation 14-1.

### EQUATION 14-1: CALCULATING THE PWM PERIOD<sup>(1)</sup>

$$\text{PWM Period} = [(PRy) + 1] \cdot T_{CY} \cdot (\text{Timer Prescale Value})$$

where:

$$\text{PWM Frequency} = 1/[\text{PWM Period}]$$

**Note 1:** Based on  $T_{CY} = 2 \cdot T_{OSC}$ , Doze mode and PLL are disabled.

**Note:** A PRy value of N will produce a PWM period of N + 1 time base count cycles. For example, a value of 7 written into the PRy register will yield a period consisting of 8 time base cycles.

### 14.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the OCxRS register. The OCxRS register can be written to at any time, but the duty cycle value is not latched into OCxR until a match between PRy and TMRy occurs (i.e., the period is complete). This provides a double buffer for the PWM duty cycle and is essential for glitchless PWM operation. In the PWM mode, OCxR is a read-only register.

Some important boundary parameters of the PWM duty cycle include:

- If the Output Compare x register, OCxR, is loaded with 0000h, the OCx pin will remain low (0% duty cycle).
- If OCxR is greater than PRy (Timer Period register), the pin will remain high (100% duty cycle).
- If OCxR is equal to PRy, the OCx pin will be low for one time base count value and high for all other count values.

See Example 14-1 for PWM mode timing details. Table 14-1 shows example PWM frequencies and resolutions for a device operating at 10 MIPS.

### EQUATION 14-2: CALCULATION FOR MAXIMUM PWM RESOLUTION<sup>(1)</sup>

$$\text{Maximum PWM Resolution (bits)} = \frac{\log_{10}\left(\frac{F_{CY}}{F_{PWM} \cdot (\text{Timer Prescale Value})}\right)}{\log_{10}(2)} \text{ bits}$$

**Note 1:** Based on  $F_{CY} = F_{OSC}/2$ , Doze mode and PLL are disabled.



# PIC24FJ64GA004 FAMILY

## EXAMPLE 14-1: PWM PERIOD AND DUTY CYCLE CALCULATIONS<sup>(1)</sup>

- Find the Timer Period register value for a desired PWM frequency of 52.08 kHz, where FOSC = 8 MHz with PLL (32 MHz device clock rate) and a Timer2 prescaler setting of 1:1.
  - $TCY = 2 * T_{osc} = 62.5 \text{ ns}$
  - PWM Period =  $1/\text{PWM Frequency} = 1/52.08 \text{ kHz} = 19.2 \mu\text{s}$
  - PWM Period =  $(PR2 + 1) \cdot TCY \cdot (\text{Timer 2 Prescale Value})$
  - $19.2 \mu\text{s} = (PR2 + 1) \cdot 62.5 \text{ ns} \cdot 1$
  - PR2 = 306
- 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:
  - PWM Resolution =  $\log_{10}(F_{CY}/F_{PWM})/\log_{10}(2)$  bits
  - =  $(\log_{10}(16 \text{ MHz}/52.08 \text{ kHz})/\log_{10}(2))$  bits
  - = 8.3 bits

**Note 1:** Based on  $TCY = 2 * T_{OSC}$ , Doze mode and PLL are disabled.

**TABLE 14-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 4 MIPS (FCY = 4 MHz)<sup>(1)</sup>**

| PWM Frequency         | 7.6 Hz | 61 Hz | 122 Hz | 977 Hz | 3.9 kHz | 31.3 kHz | 125 kHz |
|-----------------------|--------|-------|--------|--------|---------|----------|---------|
| Timer Prescaler Ratio | 8      | 1     | 1      | 1      | 1       | 1        | 1       |
| Period Register Value | FFFFh  | FFFFh | 7FFFh  | 0FFFh  | 03FFh   | 007Fh    | 001Fh   |
| Resolution (bits)     | 16     | 16    | 15     | 12     | 10      | 7        | 5       |

**Note 1:** Based on  $F_{CY} = F_{OSC}/2$ , Doze mode and PLL are disabled.

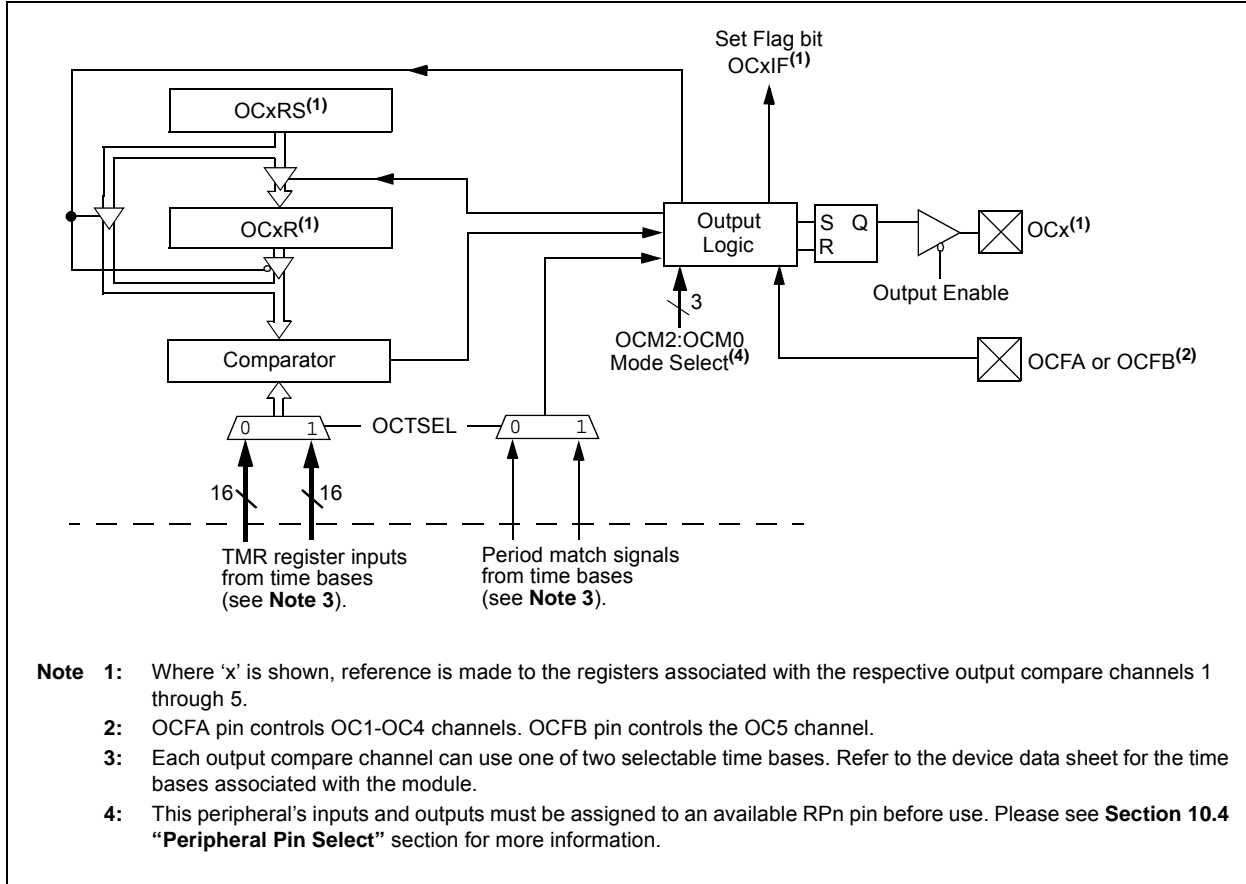
**TABLE 14-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 16 MIPS (FCY = 16 MHz)<sup>(1)</sup>**

| PWM Frequency         | 30.5 Hz | 244 Hz | 488 Hz | 3.9 kHz | 15.6 kHz | 125 kHz | 500 kHz |
|-----------------------|---------|--------|--------|---------|----------|---------|---------|
| Timer Prescaler Ratio | 8       | 1      | 1      | 1       | 1        | 1       | 1       |
| Period Register Value | FFFFh   | FFFFh  | 7FFFh  | 0FFFh   | 03FFh    | 007Fh   | 001Fh   |
| Resolution (bits)     | 16      | 16     | 15     | 12      | 10       | 7       | 5       |

**Note 1:** Based on  $F_{CY} = F_{OSC}/2$ , Doze mode and PLL are disabled.

# PIC24FJ64GA004 FAMILY

FIGURE 14-1: OUTPUT COMPARE MODULE BLOCK DIAGRAM



# PIC24FJ64GA004 FAMILY

## 14.4 Output Compare Register

**REGISTER 14-1: OCxCON: OUTPUT COMPARE x CONTROL REGISTER**

|        |     |        |     |     |     |       |     |
|--------|-----|--------|-----|-----|-----|-------|-----|
| U-0    | U-0 | R/W-0  | U-0 | U-0 | U-0 | U-0   | U-0 |
| —      | —   | OCSIDL | —   | —   | —   | —     | —   |
| bit 15 |     |        |     |     |     | bit 8 |     |

|       |     |     |         |        |                     |                     |                     |
|-------|-----|-----|---------|--------|---------------------|---------------------|---------------------|
| U-0   | U-0 | U-0 | R-0, HC | R/W-0  | R/W-0               | R/W-0               | R/W-0               |
| —     | —   | —   | OCFLT   | OCTSEL | OCM2 <sup>(1)</sup> | OCM1 <sup>(1)</sup> | OCM0 <sup>(1)</sup> |
| bit 7 |     |     |         |        |                     | bit 0               |                     |

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

- bit 15-14    **Unimplemented:** Read as '0'
- bit 13      **OCSIDL:** Stop Output Compare x in Idle Mode Control bit
  - 1 = Output Compare x will halt in CPU Idle mode
  - 0 = Output Compare x will continue to operate in CPU Idle mode
- bit 12-5    **Unimplemented:** Read as '0'
- bit 4        **OCFLT:** PWM Fault Condition Status bit
  - 1 = PWM Fault condition has occurred (cleared in HW only)
  - 0 = No PWM Fault condition has occurred (this bit is only used when OCM<2:0> = 111)
- bit 3        **OCTSEL:** Output Compare x Timer Select bit
  - 1 = Timer3 is the clock source for Output Compare x
  - 0 = Timer2 is the clock source for Output Compare x
 Refer to the device data sheet for specific time bases available to the output compare module.
- bit 2-0     **OCM2:OCM0:** Output Compare x Mode Select bits<sup>(1)</sup>
  - 111 = PWM mode on OCx, Fault pin, OCFx, enabled<sup>(2)</sup>
  - 110 = PWM mode on OCx, Fault pin, OCFx, disabled<sup>(2)</sup>
  - 101 = Initialize OCx pin low, generate continuous output pulses on OCx pin
  - 100 = Initialize OCx pin low, generate single output pulse on OCx pin
  - 011 = Compare event toggles OCx pin
  - 010 = Initialize OCx pin high, compare event forces OCx pin low
  - 001 = Initialize OCx pin low, compare event forces OCx pin high
  - 000 = Output compare channel is disabled

**Note 1:** RPORx (OCx) must be configured to an available RPN pin. For more information, see **Section 10.4 "Peripheral Pin Select"**.

**2:** OCFA pin controls OC1-OC4 channels. OCFB pin controls the OC5 channel.

# PIC24FJ64GA004 FAMILY

---

NOTES:

## 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's SPI and SIOP interfaces.

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.

Depending on the pin count, devices of the PIC24FJ64GA004 family offer one or two SPI modules on a single device.

**Note:** In this section, the SPI modules are referred to together as SPIx or separately as SPI1 and SPI2. Special Function Registers will follow a similar notation. For example, SPIxCON1 or SPIxCON2 refers to the control register for the SPI1 or SPI2 module.

# PIC24FJ64GA004 FAMILY

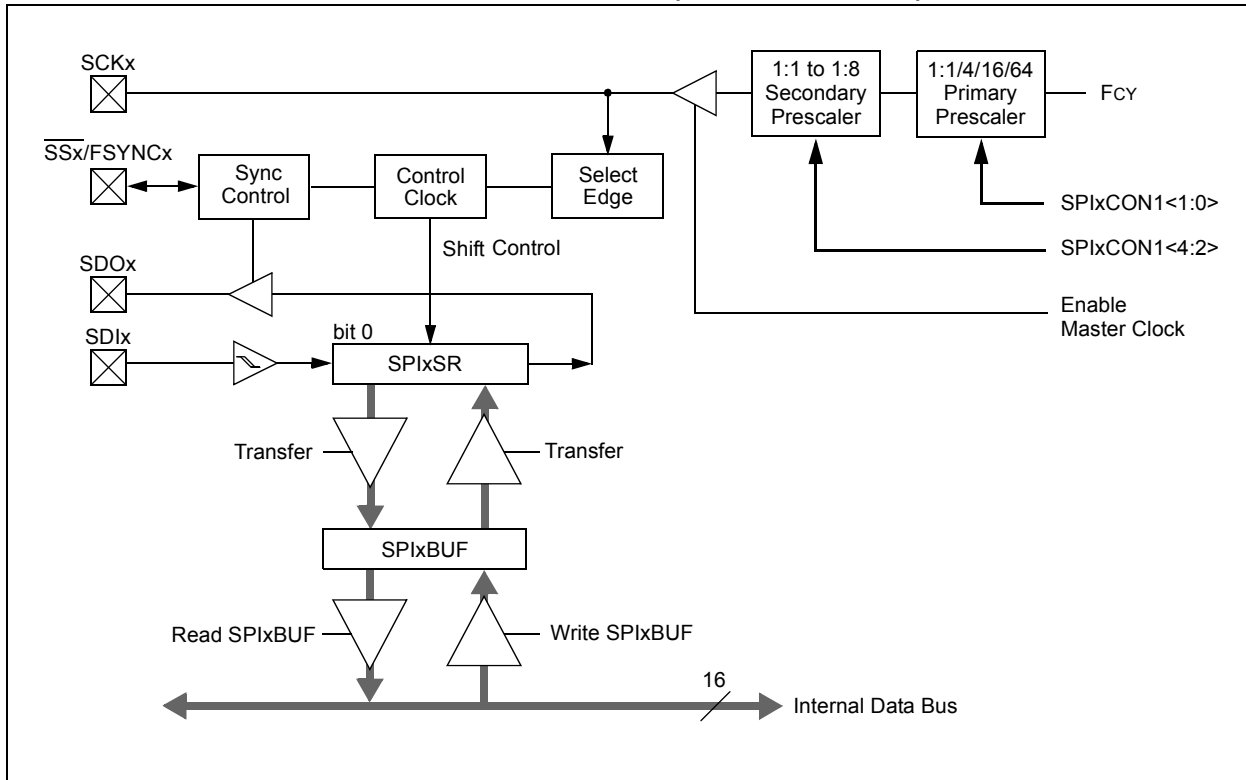
To set up the SPI module for the Standard Master mode of operation:

1. If using interrupts:
  - a) Clear the SPIxIF bit in the respective IFSx register.
  - b) Set the SPIxIE bit in the respective IECx register.
  - c) Write the SPIxIP bits in the respective IPCx 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 IFSx register.
  - b) Set the SPIxIE bit in the respective IECx register.
  - c) Write the SPIxIP bits in the respective IPCx 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 (SPIxCON1<7>) must be set to enable the  $\overline{SS}$  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)**





# PIC24FJ64GA004 FAMILY

## 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 <sup>(1)</sup> | —   | SPISIDL | —   | —   | SPIBEC2 | SPIBEC1 | SPIBEC0 |
| bit 15               |     |         |     |     |         |         | bit 8   |

|       |        |        |        |        |        |        |        |
|-------|--------|--------|--------|--------|--------|--------|--------|
| R-0   | R/C-0  | 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  |

|                                    |                      |
|------------------------------------|----------------------|
| <b>Legend:</b>                     | 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      **SPIEN:** SPIx Enable bit<sup>(1)</sup>  
 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 = Discontinues module operation when device enters Idle mode  
 0 = Continues module operation in Idle mode
- bit 12-11   **Unimplemented:** Read as '0'
- bit 10-8    **SPIBEC2:SPIBEC0:** SPIx Buffer Element Count bits (valid in Enhanced Buffer mode)  
Master mode:  
 Number of SPI transfers pending.  
Slave mode:  
 Number of SPI transfers 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     **SISEL2:SISEL0:** 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 set)  
 010 = Interrupt when SPIx receive buffer is 3/4 or more full  
 001 = Interrupt when data is available in 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 is set)

**Note 1:** If SPIEN = 1, these functions must be assigned to available RPN pins before use. See **Section 10.4** "Peripheral Pin Select" for more information.



# PIC24FJ64GA004 FAMILY

---

## 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 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” for more information.

# PIC24FJ64GA004 FAMILY

## 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 <sup>(1)</sup> | DISSDO <sup>(2)</sup> | MODE16 | SMP   | CKE <sup>(3)</sup> |
| 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 <sup>(4)</sup> | 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:** Disables SCKx pin bit (SPI Master modes only)<sup>(1)</sup>  
                  1 = Internal SPI clock is disabled; pin functions as I/O  
                  0 = Internal SPI clock is enabled
- bit 11      **DISSDO:** Disables SDOx pin bit<sup>(2)</sup>  
                  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 sampled at end of data output time  
                  0 = Input data sampled at 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<sup>(3)</sup>  
                  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 bit (Slave mode)<sup>(4)</sup>  
                  1 =  $\overline{SSx}$  pin used for Slave mode  
                  0 =  $\overline{SSx}$  pin not used by module; pin 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”** for more information.
- Note 2:** If DISSDO = 0, SDOx must be configured to an available RPN pin. See **Section 10.4 “Peripheral Pin Select”** for more information.
- Note 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).
- Note 4:** If SSEN = 1,  $\overline{SSx}$  must be configured to an available RPN pin. See **Section 10.4 “Peripheral Pin Select”** for more information.

# PIC24FJ64GA004 FAMILY

## REGISTER 15-2: SPIxCON1: SPIx CONTROL REGISTER 1 (CONTINUED)

bit 4-2 **SPRE2:SPRE0:** Secondary Prescale bits (Master mode)

111 = Secondary prescale 1:1  
 110 = Secondary prescale 2:1  
 ...  
 000 = Secondary prescale 8:1

bit 1-0 **PPRE1:PPRE0:** 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”** for more information.
- 2:** If DISSDO = 0, SDOx must be configured to an available RPN pin. See **Section 10.4 “Peripheral Pin Select”** 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”** 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 enabled  
 0 = Framed SPIx support disabled

bit 14 **SPIFSD:** Frame Sync Pulse Direction Control on  $\overline{SSx}$  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

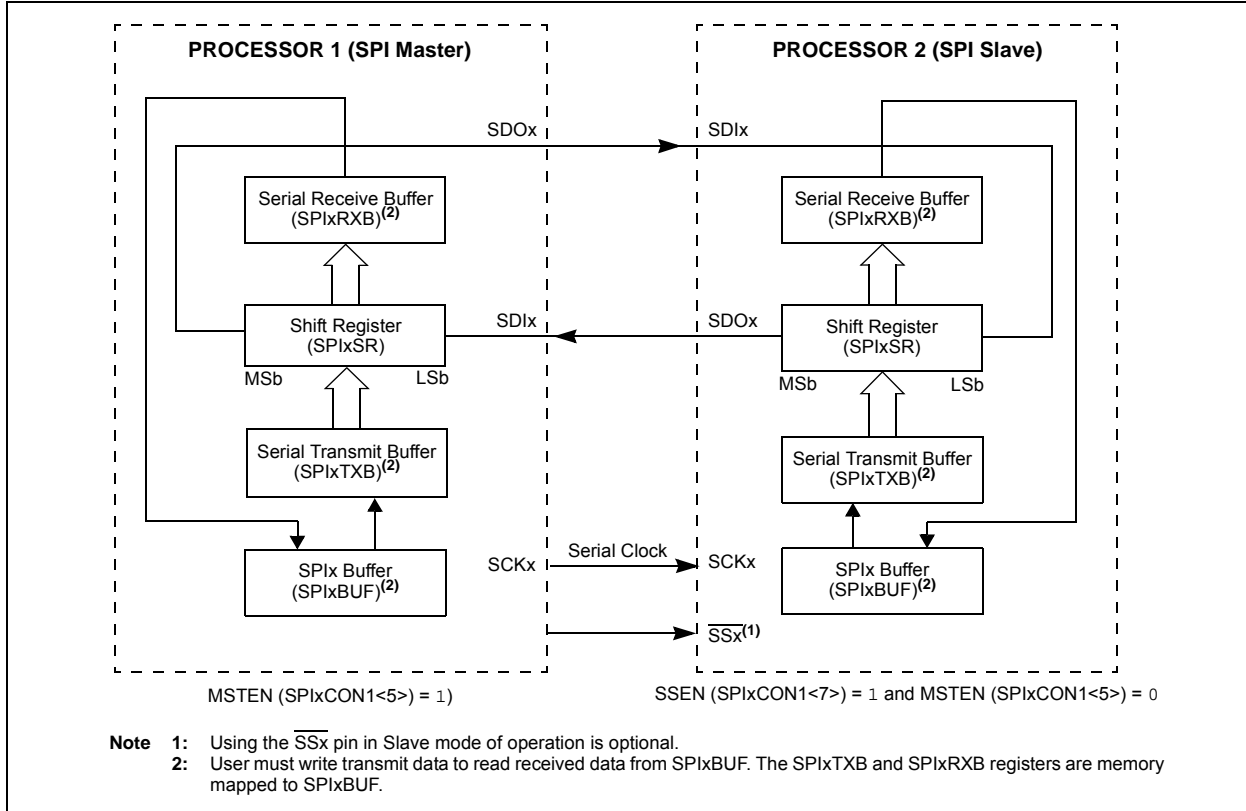
1 = Frame sync pulse coincides with first bit clock  
 0 = Frame sync pulse precedes first bit clock

bit 0 **SPIBEN:** Enhanced Buffer Enable bit

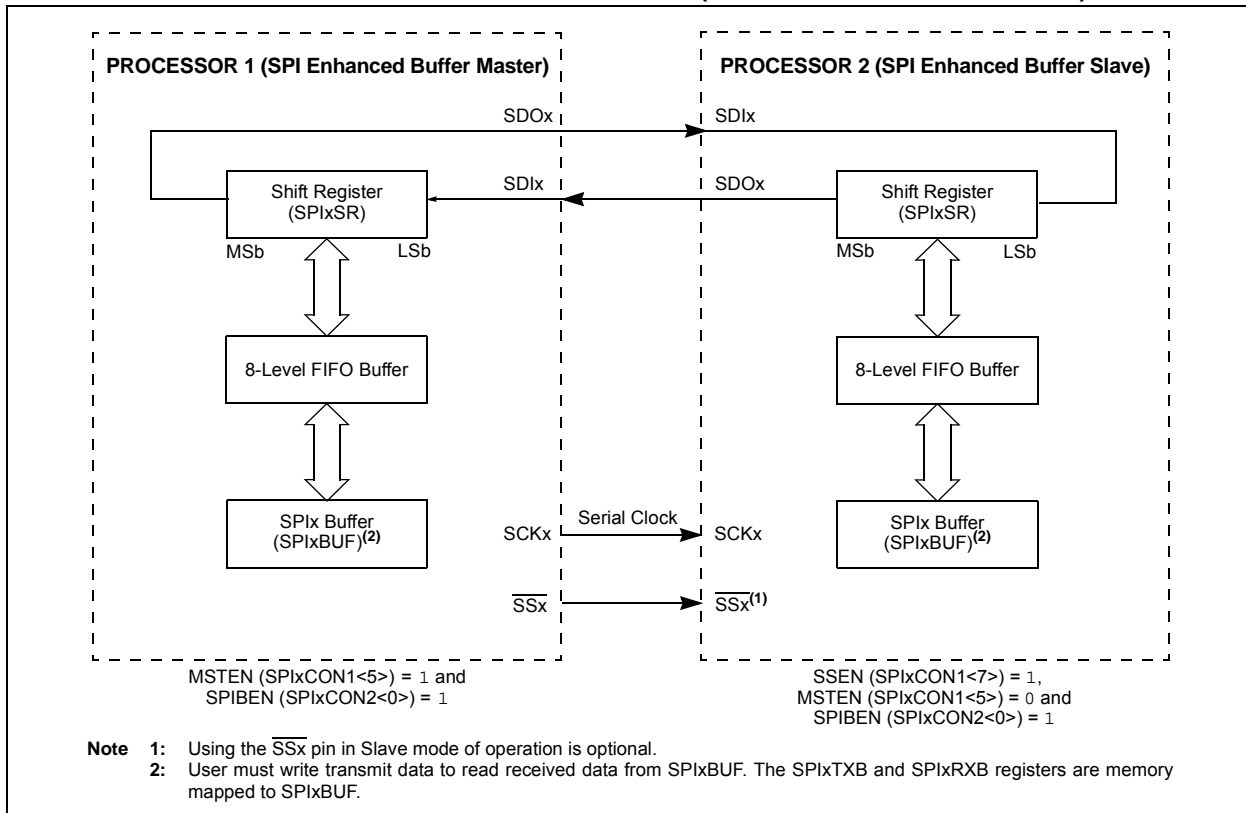
1 = Enhanced Buffer enabled  
 0 = Enhanced Buffer disabled (Legacy mode)

# PIC24FJ64GA004 FAMILY

**FIGURE 15-3: SPI MASTER/SLAVE CONNECTION (STANDARD MODE)**

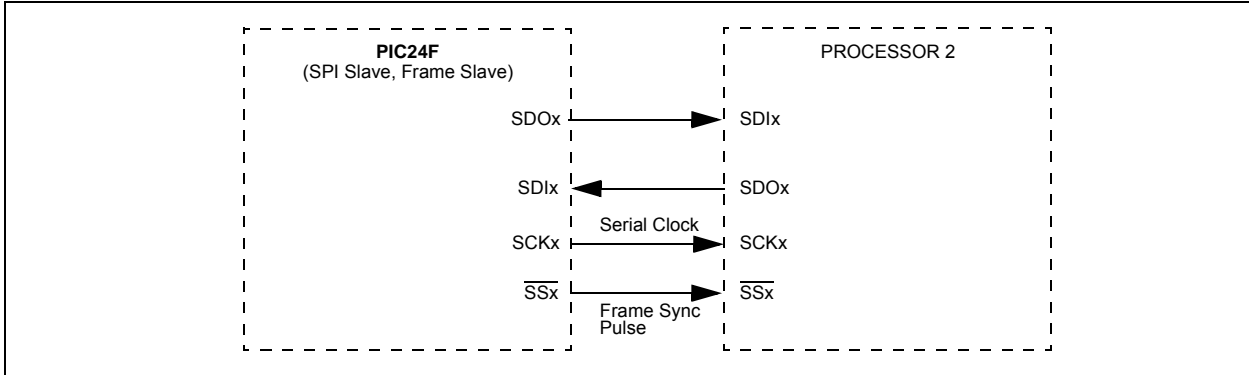


**FIGURE 15-4: SPI MASTER/SLAVE CONNECTION (ENHANCED BUFFER MODES)**

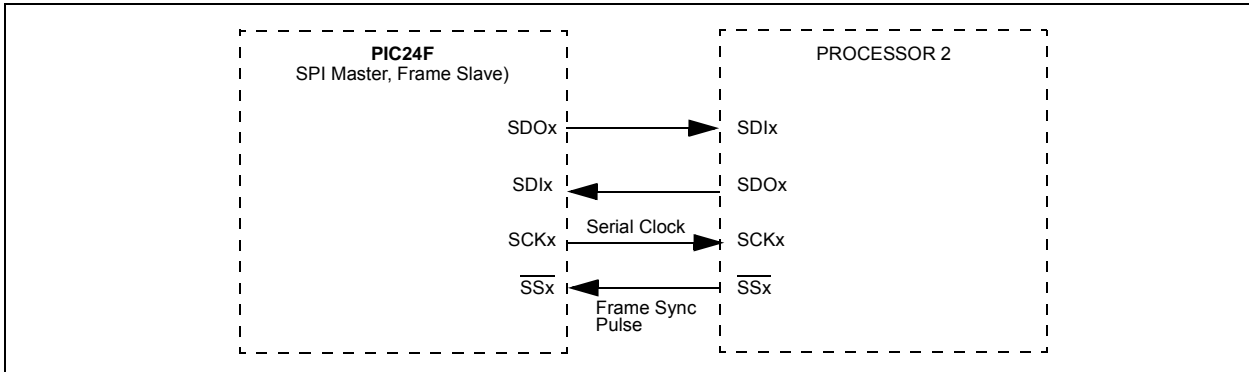


# PIC24FJ64GA004 FAMILY

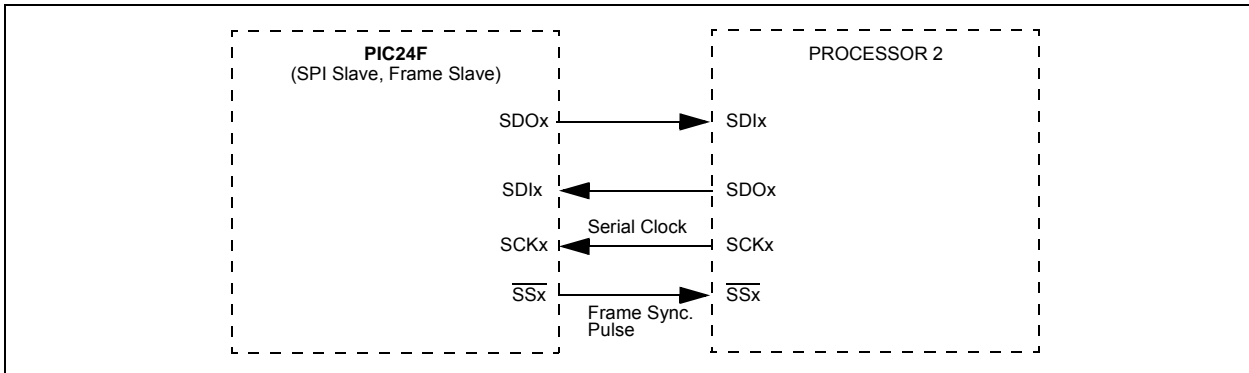
**FIGURE 15-5: SPI MASTER, FRAME MASTER CONNECTION DIAGRAM**



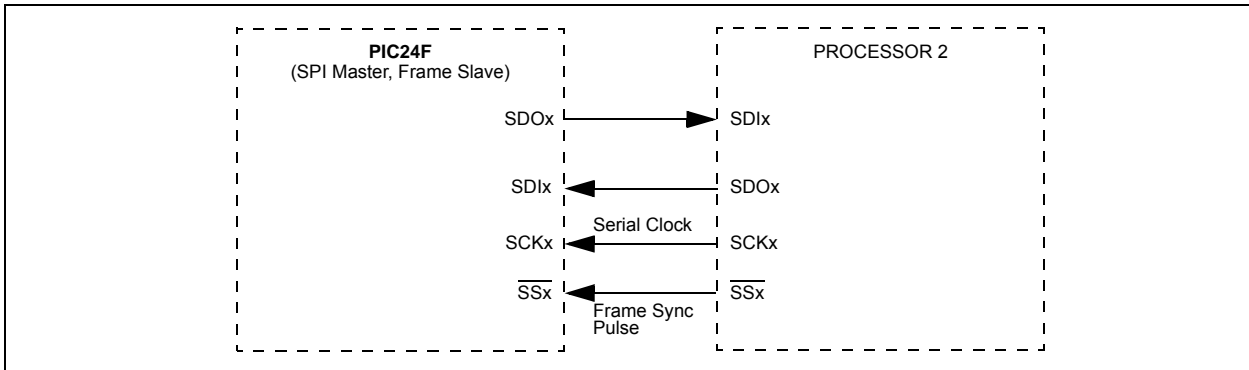
**FIGURE 15-6: SPI MASTER, FRAME SLAVE CONNECTION DIAGRAM**



**FIGURE 15-7: SPI SLAVE, FRAME MASTER CONNECTION DIAGRAM**



**FIGURE 15-8: SPI SLAVE, FRAME SLAVE CONNECTION DIAGRAM**



# PIC24FJ64GA004 FAMILY

**EQUATION 15-1: RELATIONSHIP BETWEEN DEVICE AND SPI CLOCK SPEED<sup>(1)</sup>**

$$F_{SCK} = \frac{F_{CY}}{\text{Primary Prescaler} * \text{Secondary Prescaler}}$$

**Note 1:** Based on  $F_{CY} = F_{OSC}/2$ ; Doze mode and PLL are disabled.

**TABLE 15-1: SAMPLE SCK FREQUENCIES<sup>(1,2)</sup>**

| F <sub>CY</sub> = 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   |
| F <sub>CY</sub> = 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  $F_{CY} = F_{OSC}/2$ ; Doze mode and PLL are disabled.

**Note 2:** SCKx frequencies shown in kHz.

## 16.0 INTER-INTEGRATED CIRCUIT (I<sup>2</sup>C™)

**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<sup>2</sup>C™)” (DS39702).

The Inter-Integrated Circuit™ (I<sup>2</sup>C™) 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<sup>2</sup>C module supports these features:

- Independent master and slave logic
- 7-bit and 10-bit device addresses
- General call address, as defined in the I<sup>2</sup>C 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 Peripheral Remapping Options

The I<sup>2</sup>C modules are tied to fixed pin assignments, and cannot be reassigned to alternate pins using peripheral pin select. To allow some flexibility with peripheral multiplexing, the I2C1 module in all devices, can be reassigned to the alternate pins, designated as ASCL1 and ASDA1 during device configuration.

Pin assignment is controlled by the I2C1SEL Configuration bit; programming this bit (= 0) multiplexes the module to the ASCL1 and ASDA1 pins.

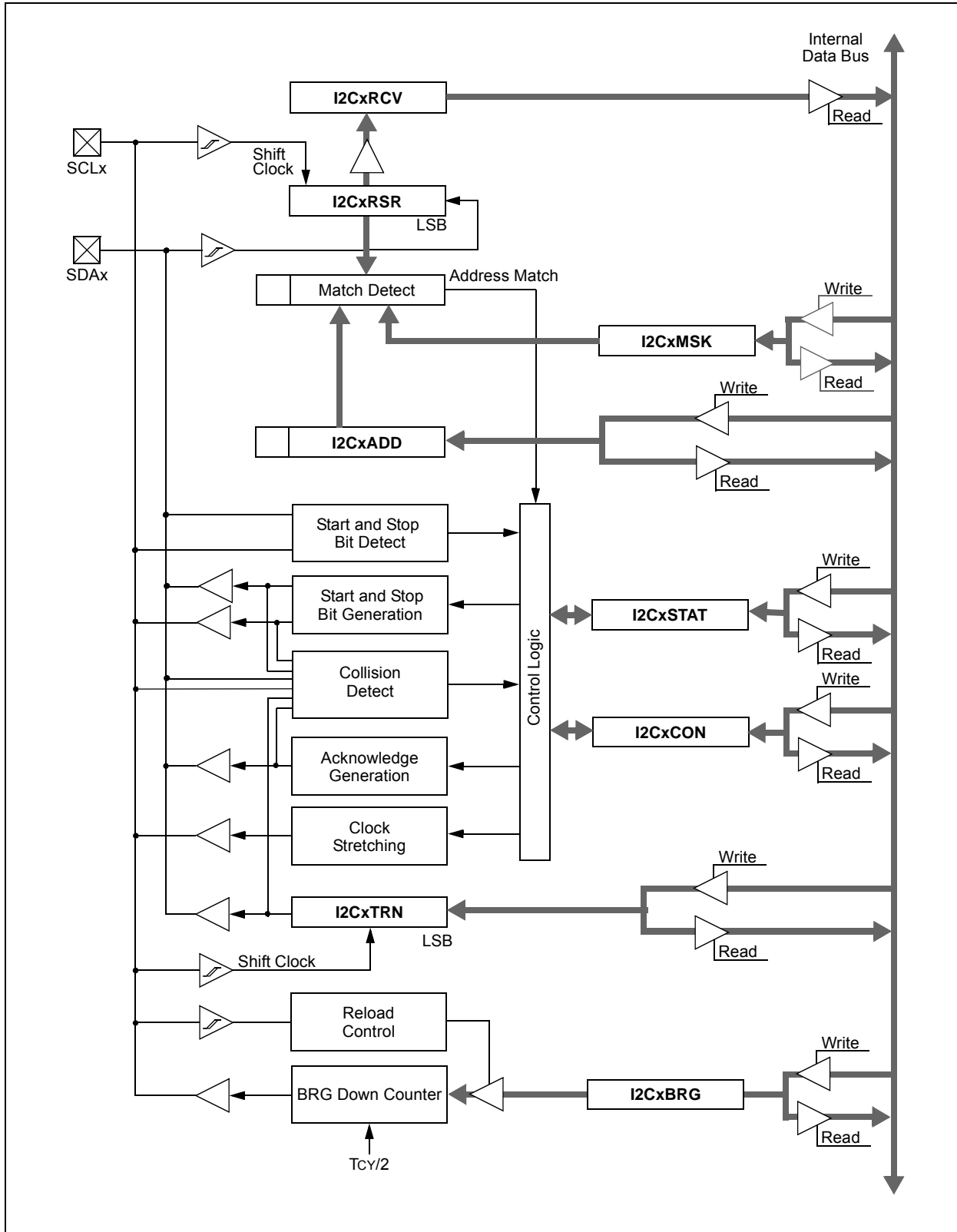
## 16.2 Communicating as a Master in a Single Master Environment

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<sup>2</sup>C 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.

# PIC24FJ64GA004 FAMILY

FIGURE 16-1: I<sup>2</sup>C™ BLOCK DIAGRAM





# PIC24FJ64GA004 FAMILY

## 16.3 Setting Baud Rate When Operating as a Bus Master

To compute the Baud Rate Generator reload value, use Equation 16-1.

**EQUATION 16-1: COMPUTING BAUD RATE RELOAD VALUE<sup>(1)</sup>**

$$F_{SCL} = \frac{F_{CY}}{I2CxBRG + 1 + \frac{F_{CY}}{10,000,000}}$$

or

$$I2CxBRG = \left( \frac{F_{CY}}{F_{SCL}} - \frac{F_{CY}}{10,000,000} \right) - 1$$

**Note 1:** Based on  $F_{CY} = F_{OSC}/2$ ; Doze mode and PLL are disabled.

## 16.4 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, ‘00000000’ and ‘00100000’.

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<sup>2</sup>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<sup>2</sup>C™ CLOCK RATES<sup>(1)</sup>**

| Required System F <sub>SCL</sub> | F <sub>CY</sub> | I2CxBRG Value |               | Actual F <sub>SCL</sub> |
|----------------------------------|-----------------|---------------|---------------|-------------------------|
|                                  |                 | (Decimal)     | (Hexadecimal) |                         |
| 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  $F_{CY} = F_{OSC}/2$ , Doze mode and PLL are disabled.

**TABLE 16-2: I<sup>2</sup>C™ RESERVED ADDRESSES<sup>(1)</sup>**

| Slave Address | R/W Bit | Description                            |
|---------------|---------|--|
| 0000 000      | 0       | General Call Address <sup>(2)</sup>    |
| 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 <sup>(3)</sup> |

**Note 1:** The address bits listed here will never cause an address match, independent of the address mask settings.

**2:** Address will be Acknowledged only if GCEN = 1.

**3:** Match on this address can only occur on the upper byte in 10-Bit Addressing mode.

# PIC24FJ64GA004 FAMILY

## 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     |

|                   |                             |
|-------------------|-----------------------------|
| <b>Legend:</b>    | HC = Hardware Clearable bit |
| R = Readable bit  | W = Writable bit            |
| -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 I2Cx module. All I<sup>2</sup>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<sup>2</sup>C Slave)  
1 = Releases SCLx clock  
0 = Holds SCLx clock low (clock stretch)  
If STREN = 1:  
Bit is R/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 Peripheral 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 disabled  
0 = Slew rate control enabled
- bit 8      **SMEN:** SMBus Input Levels bit  
1 = Enables I/O pin thresholds compliant with SMBus specification  
0 = Disables SMBus input thresholds
- bit 7      **GCEN:** General Call Enable bit (when operating as I<sup>2</sup>C slave)  
1 = Enables interrupt when a general call address is received in the I2CxRSR (module is enabled for reception)  
0 = General call address disabled
- bit 6      **STREN:** SCLx Clock Stretch Enable bit (when operating as I<sup>2</sup>C slave)  
Used in conjunction with SCLREL bit.  
1 = Enables software or receive clock stretching  
0 = Disables software or receive clock stretching

# PIC24FJ64GA004 FAMILY

---

## REGISTER 16-1: I2CxCON: I2Cx CONTROL REGISTER (CONTINUED)

- bit 5      **ACKDT:** Acknowledge Data bit (When operating as I<sup>2</sup>C 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<sup>2</sup>C 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 not in progress
- bit 3      **RCEN:** Receive Enable bit (when operating as I<sup>2</sup>C master)  
1 = Enables Receive mode for I<sup>2</sup>C. Hardware clear at end of eighth bit of master receive data byte.  
0 = Receives sequence not in progress
- bit 2      **PEN:** Stop Condition Enable bit (when operating as I<sup>2</sup>C master)  
1 = Initiates Stop condition on SDAx and SCLx pins. Hardware clear at end of master Stop sequence.  
0 = Stop condition not in progress
- bit 1      **RSEN:** Repeated Start Condition Enabled bit (when operating as I<sup>2</sup>C master)  
1 = Initiates Repeated Start condition on SDAx and SCLx pins. Hardware clear at end of master Repeated Start sequence.  
0 = Repeated Start condition not in progress
- bit 0      **SEN:** Start Condition Enabled bit (when operating as I<sup>2</sup>C master)  
1 = Initiates Start condition on SDAx and SCLx pins. Hardware clear at end of master Start sequence.  
0 = Start condition not in progress

# PIC24FJ64GA004 FAMILY

## 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 $\bar{A}$ | P          | S          | R $\bar{W}$ | RBF      | TBF      |
| bit 7     |           |             |            |            |             |          | bit 0    |

|                   |                   |                                    |  |
|-------------------|-------------------|------------------------------------|--|
| <b>Legend:</b>    | 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<sup>2</sup>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 beginning of master transmission. Hardware clear at 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 address matches 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 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 the I2CxTRN register failed because the I<sup>2</sup>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 is still holding the previous byte  
 0 = No overflow  
 Hardware set at attempt to transfer I2CxRSR to I2CxRCV (cleared by software).
- bit 5        **D $\bar{A}$ :** Data/Address bit (when operating as I<sup>2</sup>C slave)  
 1 = Indicates that the last byte received was data  
 0 = Indicates that the last byte received was device address  
 Hardware clear at device address match. Hardware set by write to I2CxTRN or by reception of slave byte.

# PIC24FJ64GA004 FAMILY

---

## 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 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 detected.
- bit 2      **R/W:** Read/Write Information bit (when operating as I<sup>2</sup>C slave)  
1 = Read – indicates data transfer is output from slave  
0 = Write – indicates data transfer is input to slave  
Hardware set or clear after reception of I<sup>2</sup>C device address byte.
- bit 1      **RBF:** Receive Buffer Full Status bit  
1 = Receive complete, I2CxRCV is full  
0 = Receive 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 in progress, I2CxTRN is full  
0 = Transmit complete, I2CxTRN is empty  
Hardware set when software writes I2CxTRN. Hardware clear at completion of data transmission.

# PIC24FJ64GA004 FAMILY

## 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 **AMSK9:AMSK0:** Mask for Address Bit x Select bits

1 = Enable masking for bit x of incoming message address; bit match not required in this position

0 = Disable masking for bit x; bit match required in this position

## 16.5 Acknowledge Status

In both Master and Slave modes, the ACKSTAT bit is only updated when transmitting data resulting in the reception of an ACK or NACK from another device. Do not check the state of ACKSTAT when receiving data, either as a Slave or a Master. Reading ACKSTAT after receiving address or data bytes returns an invalid result.

# PIC24FJ64GA004 FAMILY

## 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, 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:

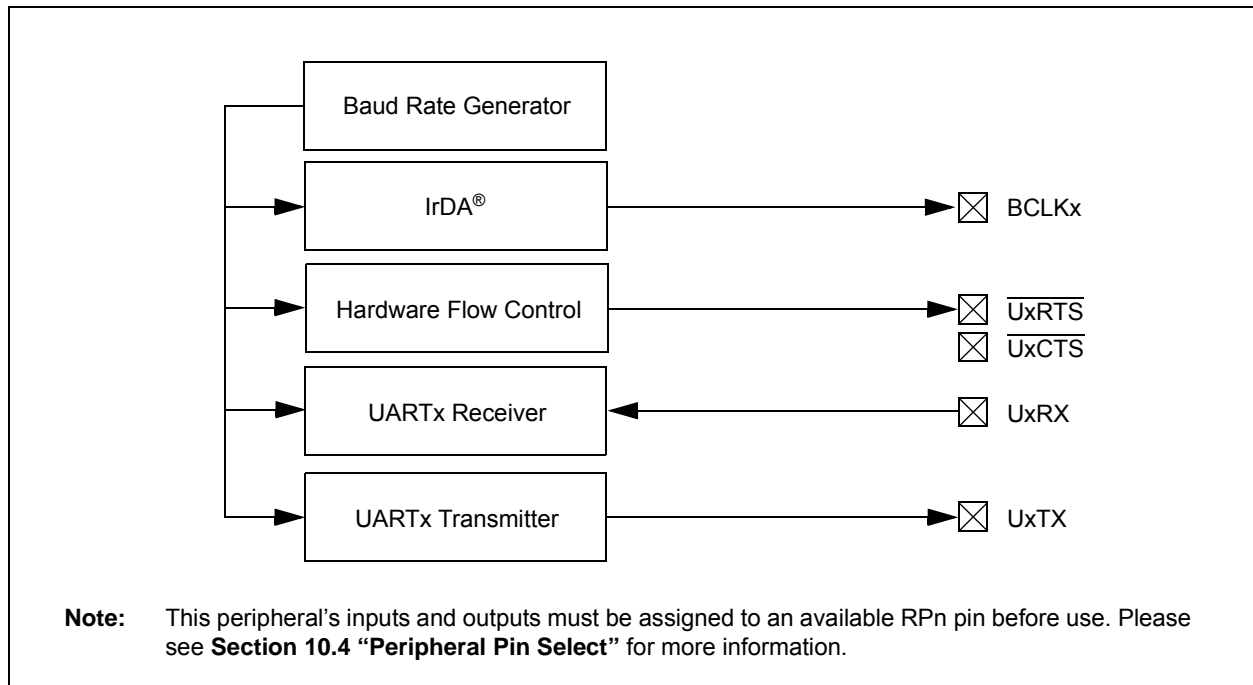
- Full-Duplex, 8 or 9-Bit Data Transmission through the UxTX and UxRX Pins
- Even, Odd or No Parity Options (for 8-bit data)
- One or Two Stop bits
- Hardware Flow Control Option with UxCTS and UxRTS Pins

- 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**



# PIC24FJ64GA004 FAMILY

## 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<sup>(1)</sup>

$$\text{Baud Rate} = \frac{\text{FCY}}{16 \cdot (\text{UxBRG} + 1)}$$
$$\text{UxBRG} = \frac{\text{FCY}}{16 \cdot \text{Baud Rate}} - 1$$

**Note 1:** 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

### EXAMPLE 17-1: BAUD RATE ERROR CALCULATION (BRGH = 0)<sup>(1)</sup>

$$\text{Desired Baud Rate} = \text{FCY}/(16 (\text{UxBRG} + 1))$$

Solving for UxBRG value:

$$\text{UxBRG} = ((\text{FCY}/\text{Desired Baud Rate})/16) - 1$$

$$\text{UxBRG} = ((4000000/9600)/16) - 1$$

$$\text{UxBRG} = 25$$

$$\begin{aligned} \text{Calculated Baud Rate} &= 4000000/(16 (25 + 1)) \\ &= 9615 \end{aligned}$$

$$\begin{aligned} \text{Error} &= (\text{Calculated Baud Rate} - \text{Desired Baud Rate}) \\ &\quad \text{Desired Baud Rate} \\ &= (9615 - 9600)/9600 \\ &= 0.16\% \end{aligned}$$

**Note 1:** Based on FCY = FOSC/2; Doze mode and PLL are disabled.

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<sup>(1)</sup>

$$\text{Baud Rate} = \frac{\text{FCY}}{4 \cdot (\text{UxBRG} + 1)}$$
$$\text{UxBRG} = \frac{\text{FCY}}{4 \cdot \text{Baud Rate}} - 1$$

**Note 1:** 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.



## 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 2 cycles after being set).
4. Write data byte to lower byte of UxTXREG word. The value will be immediately transferred to the Transmit Shift Register (TSR), and the serial bit stream will start shifting out with 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 2 cycles after being set).
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. 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 – sets up the Break character.
3. Load the UxTXREG with a dummy character to initiate transmission (value is ignored).
4. Write ‘55h’ to UxTXREG – 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 mode. 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 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 EXTERNAL IrDA SUPPORT – IrDA CLOCK OUTPUT

To support external IrDA encoder and decoder devices, the BCLKx pin (same as the UxRTS pin) can be configured to generate the 16x baud clock. With 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.

# PIC24FJ64GA004 FAMILY

## 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 <sup>(3)</sup> | R/W-0 <sup>(3)</sup> |
| UARTEN <sup>(1)</sup> | —   | USIDL | IREN <sup>(2)</sup> | RTSMD | —   | UEN1                 | UEN0                 |
| bit 15                |     |       |                     |       |     | bit 8                |                      |

|           |        |           |       |       |        |        |       |
|-----------|--------|-----------|-------|-------|--------|--------|-------|
| R/C-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  |       |

|                   |                   |  |
|-------------------|-------------------|--|
| <b>Legend:</b>    | 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      **UARTEN:** UARTx Enable bit<sup>(1)</sup>  
 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 device enters Idle mode  
 0 = Continue module operation in Idle mode
- bit 12      **IREN:** IrDA<sup>®</sup> Encoder and Decoder Enable bit<sup>(2)</sup>  
 1 = IrDA encoder and decoder enabled  
 0 = IrDA encoder and decoder disabled
- bit 11      **RTSMD:** Mode Selection for  $\overline{\text{UxRTS}}$  Pin bit  
 1 =  $\overline{\text{UxRTS}}$  pin in Simplex mode  
 0 =  $\overline{\text{UxRTS}}$  pin in Flow Control mode
- bit 10      **Unimplemented:** Read as '0'
- bit 9-8      **UEN1:UEN0:** UARTx Enable bits<sup>(3)</sup>  
 11 =  $\overline{\text{UxTX}}$ ,  $\overline{\text{UxRX}}$  and  $\overline{\text{BCLKx}}$  pins are enabled and used;  $\overline{\text{UxCTS}}$  pin controlled by PORT latches  
 10 =  $\overline{\text{UxTX}}$ ,  $\overline{\text{UxRX}}$ ,  $\overline{\text{UxCTS}}$  and  $\overline{\text{UxRTS}}$  pins are enabled and used  
 01 =  $\overline{\text{UxTX}}$ ,  $\overline{\text{UxRX}}$  and  $\overline{\text{UxRTS}}$  pins are enabled and used;  $\overline{\text{UxCTS}}$  pin controlled by PORT latches  
 00 =  $\overline{\text{UxTX}}$  and  $\overline{\text{UxRX}}$  pins are enabled and used;  $\overline{\text{UxCTS}}$  and  $\overline{\text{UxRTS/BCLKx}}$  pins 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  $\overline{\text{UxRX}}$  pin; interrupt generated on falling edge, bit cleared in hardware on following rising edge  
 0 = No wake-up 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 disabled or completed

**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”** for more information.

**2:** This feature is only available for the 16x BRG mode (BRGH = 0).

**3:** Bit availability depends on pin availability.

# PIC24FJ64GA004 FAMILY

---

## REGISTER 17-1: UxMODE: UARTx MODE REGISTER (CONTINUED)

|         |  |
|---------|--|
| bit 4   | <b>RXINV:</b> Receive Polarity Inversion bit<br>1 = UxRX Idle state is '0'<br>0 = UxRX Idle state is '1'   |
| bit 3   | <b>BRGH:</b> High Baud Rate Enable bit<br>1 = BRG generates 4 clocks per bit period (4x baud clock, High-Speed mode)<br>0 = BRG generates 16 clocks per bit period (16x baud clock, Standard mode) |
| bit 2-1 | <b>PDSEL1:PDSEL0:</b> Parity and Data Selection bits<br>11 = 9-bit data, no parity<br>10 = 8-bit data, odd parity<br>01 = 8-bit data, even parity<br>00 = 8-bit data, no parity                    |
| bit 0   | <b>STSEL:</b> Stop Bit Selection bit<br>1 = Two Stop bits<br>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"** for more information.
- 2:** This feature is only available for the 16x BRG mode (`BRGH = 0`).
- 3:** Bit availability depends on pin availability.

# PIC24FJ64GA004 FAMILY

## 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 | UTXISEL0 | —   | UTXBRK    | UTXEN <sup>(1)</sup> | 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 |

|                   |                   |  |
|-------------------|-------------------|--|
| <b>Legend:</b>    | 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      **UTXISEL1:UTXISEL0:** 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<sup>®</sup> Encoder Transmit Polarity Inversion bit  
If IREN = 0:  
 1 = UxTX Idle '0'  
 0 = UxTX Idle '1'  
If 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 disabled or completed
- bit 10      **UTXEN:** Transmit Enable bit<sup>(1)</sup>  
 1 = Transmit enabled, UxTX pin controlled by UARTx  
 0 = Transmit disabled, any pending transmission is aborted and buffer is reset. UxTX pin controlled by the PORT register.
- 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      **URXISEL1:URXISEL0:** 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:** If UARTEN = 1, the peripheral inputs and outputs must be configured to an available RPN pin. See **Section 10.4 “Peripheral Pin Select”** for more information.

# PIC24FJ64GA004 FAMILY

---

## 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 enabled. If 9-bit mode is not selected, this does not take effect.  
0 = Address Detect mode 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:** If **UARTEN** = 1, the peripheral inputs and outputs must be configured to an available RPN pin. See **Section 10.4 “Peripheral Pin Select”** for more information.

# PIC24FJ64GA004 FAMILY

## REGISTER 17-3: UxTXREG: UARTx TRANSMIT REGISTER

|        |     |     |     |     |     |     |       |
|--------|-----|-----|-----|-----|-----|-----|-------|
| U-x    | U-x | U-x | U-x | U-x | U-x | U-x | W-x   |
| —      | —   | —   | —   | —   | —   | —   | UTX8  |
| bit 15 |     |     |     |     |     |     | bit 8 |

|       |      |      |      |      |      |      |       |
|-------|------|------|------|------|------|------|-------|
| W-x   | W-x  | W-x  | W-x  | W-x  | W-x  | W-x  | W-x   |
| UTX7  | UTX6 | UTX5 | UTX4 | UTX3 | UTX2 | UTX1 | UTX0  |
| 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      **UTX8:** Data of the Transmitted Character bit (in 9-bit mode)  
 bit 7-0      **UTX7:UTX0:** Data of the Transmitted Character bits

## REGISTER 17-4: UxRXREG: UARTx RECEIVE REGISTER

|        |     |     |     |     |     |     |       |
|--------|-----|-----|-----|-----|-----|-----|-------|
| U-0    | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R-0   |
| —      | —   | —   | —   | —   | —   | —   | URX8  |
| bit 15 |     |     |     |     |     |     | bit 8 |

|       |      |      |      |      |      |      |       |
|-------|------|------|------|------|------|------|-------|
| R-0   | R-0  | R-0  | R-0  | R-0  | R-0  | R-0  | R-0   |
| URX7  | URX6 | URX5 | URX4 | URX3 | URX2 | URX1 | URX0  |
| 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      **URX8:** Data of the Received Character bit (in 9-bit mode)  
 bit 7-0      **URX7:URX0:** Data of the Received Character bits

# PIC24FJ64GA004 FAMILY

## 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).

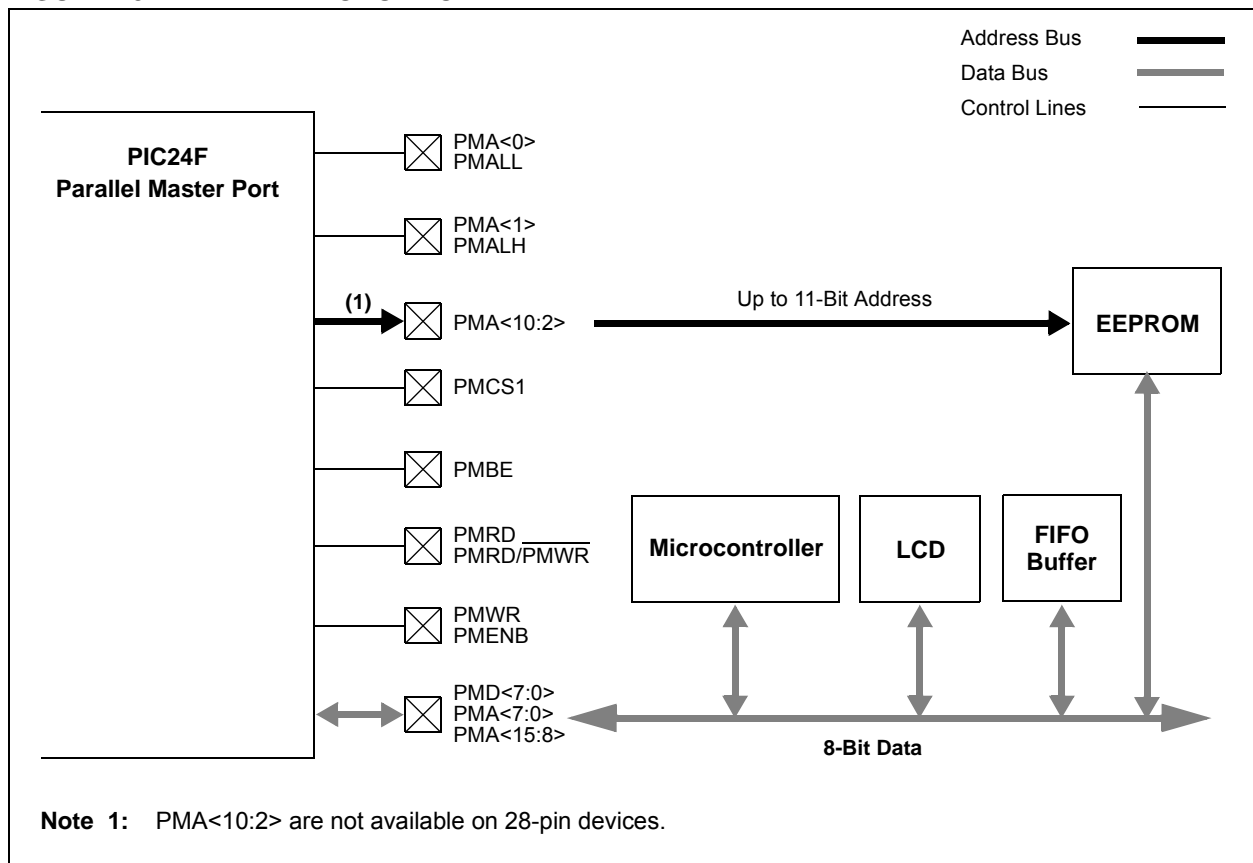
The Parallel Master Port (PMP) module is a parallel 8-bit I/O module, specifically designed to communicate with a wide variety of parallel devices, such as communication peripherals, LCDs, external memory devices and microcontrollers. Because the interface to parallel peripherals varies significantly, the PMP is highly configurable.

**Note:** A number of the pins for the PMP are not present on PIC24FJ64GA004 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/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



# PIC24FJ64GA004 FAMILY

## 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 | ADMUX1 <sup>(1)</sup> | ADMUX0 <sup>(1)</sup> | PTBEEN | PTWREN | PTRDEN |
| bit 15 |     |       |                       |                       |        |        | bit 8  |

|       |       |                      |     |                      |       |       |       |
|-------|-------|----------------------|-----|----------------------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 <sup>(2)</sup> | U-0 | R/W-0 <sup>(2)</sup> | 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 enabled  
                   0 = PMP 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        **ADMUX1:ADMUX0:** Address/Data Multiplexing Selection bits<sup>(1)</sup>  
                   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 enabled  
                   0 = PMBE port disabled
- bit 9             **PTWREN:** Write Enable Strobe Port Enable bit  
                   1 = PMWR/PMENB port enabled  
                   0 = PMWR/PMENB port disabled
- bit 8             **PTRDEN:** Read/Write Strobe Port Enable bit  
                   1 = PMRD/ $\overline{\text{PMWR}}$  port enabled  
                   0 = PMRD/ $\overline{\text{PMWR}}$  port disabled
- bit 7-6          **CSF1:CSF0:** Chip Select Function bits  
                   11 = Reserved  
                   10 = PMCS1 functions as chip set  
                   01 = Reserved  
                   00 = Reserved
- bit 5             **ALP:** Address Latch Polarity bit<sup>(2)</sup>  
                   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<sup>(2)</sup>  
                   1 = Active-high (PMCS1/PMCS1)  
                   0 = Active-low (PMCS1/PMCS1)

**Note 1:** PMA<10:2> are not available on 28-pin devices.

**2:** These bits have no effect when their corresponding pins are used as address lines.



# PIC24FJ64GA004 FAMILY

---

## 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 (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 (PMWR)  
For Master Mode 1 (PMMODE<9:8> = 11):  
1 = Enable strobe active-high (PMENB)  
0 = Enable strobe active-low (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 (PMRD)  
For Master Mode 1 (PMMODE<9:8> = 11):  
1 = Read/write strobe active-high (PMRD/PMWR)  
0 = Read/write strobe active-low (PMRD/PMWR)

**Note 1:** PMA<10:2> are not available on 28-pin devices.

**2:** These bits have no effect when their corresponding pins are used as address lines.

# PIC24FJ64GA004 FAMILY

## 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 <sup>(1)</sup> | WAITB0 <sup>(1)</sup> | WAITM3 | WAITM2 | WAITM1 | WAITM0 | WAITE1 <sup>(1)</sup> | WAITE0 <sup>(1)</sup> |
| bit 7                 |                       |        |        |        |        |                       | bit 0                 |

### Legend:

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

- bit 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        **IRQM1:IRQM0:** Interrupt Request Mode bits  
 11 = Interrupt generated when Read Buffer 3 is read or Write Buffer 3 is written (Buffered PSP mode) or on a read or write operation when PMA<1:0> = 11 (Addressable PSP mode only)  
 10 = No interrupt generated, processor stall activated  
 01 = Interrupt generated at the end of the read/write cycle  
 00 = No interrupt generated
- bit 12-11        **INCM1:INCM0:** 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           **MODE1:MODE0:** 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           **WAITB1:WAITB0:** Data Setup to Read/Write Wait State Configuration bits<sup>(1)</sup>  
 11 = Data wait of 4 T<sub>cy</sub>; multiplexed address phase of 4 T<sub>cy</sub>  
 10 = Data wait of 3 T<sub>cy</sub>; multiplexed address phase of 3 T<sub>cy</sub>  
 01 = Data wait of 2 T<sub>cy</sub>; multiplexed address phase of 2 T<sub>cy</sub>  
 00 = Data wait of 1 T<sub>cy</sub>; multiplexed address phase of 1 T<sub>cy</sub>
- bit 5-2           **WAITM3:WAITM0:** Read to Byte Enable Strobe Wait State Configuration bits  
 1111 = Wait of additional 15 T<sub>cy</sub>  
 ...  
 0001 = Wait of additional 1 T<sub>cy</sub>  
 0000 = No additional wait cycles (operation forced into one T<sub>cy</sub>)
- bit 1-0           **WAITE1:WAITE0:** Data Hold After Strobe Wait State Configuration bits<sup>(1)</sup>  
 11 = Wait of 4 T<sub>cy</sub>  
 10 = Wait of 3 T<sub>cy</sub>  
 01 = Wait of 2 T<sub>cy</sub>  
 00 = Wait of 1 T<sub>cy</sub>

**Note 1:** WAITB and WAITE bits are ignored whenever WAITM3:WAITM0 = 0000.

# PIC24FJ64GA004 FAMILY

## 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   | —     | —     | —     | ADDR<10:8> <sup>(1)</sup> |       |       |
| 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 |
| ADDR<7:0> <sup>(1)</sup> |       |       |       |       |                           |       |       |
| bit 7                    |       |       |       |       |                           |       | bit 0 |

### Legend:

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

- bit 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        **ADDR10:ADDR0:** Parallel Port Destination Address bits<sup>(1)</sup>

**Note 1:** PMA<10:2> 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 <sup>(1)</sup> | PTEN9 <sup>(1)</sup> | PTEN8 <sup>(1)</sup> |
| 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 <sup>(1)</sup> | PTEN6 <sup>(1)</sup> | PTEN5 <sup>(1)</sup> | PTEN4 <sup>(1)</sup> | PTEN3 <sup>(1)</sup> | PTEN2 <sup>(1)</sup>  | 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        **PTEN10:PTEN2:** PMP Address Port Enable bits<sup>(1)</sup>  
                     1 = PMA<10:2> function as PMP address lines  
                     0 = PMA<10:2> function as port I/O
- bit 1-0         **PTEN1:PTEN0:** PMALH/PMALL Strobe Enable bits  
                     1 = PMA1 and PMA0 function as either PMA<1:0> or PMALH and PMALL  
                     0 = PMA1 and PMA0 pads functions as port I/O

**Note 1:** PMA<10:2> are not available on 28-pin devices.

# PIC24FJ64GA004 FAMILY

## 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 |

|                   |                                    |
|-------------------|------------------------------------|
| <b>Legend:</b>    | HS = Hardware Set bit              |
| R = Readable bit  | W = Writable bit                   |
| -n = Value at POR | '1' = Bit is set                   |
|                   | U = Unimplemented bit, read as '0' |
|                   | '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

# PIC24FJ64GA004 FAMILY

## 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 | U-0 | R/W-0                   | R/W-0  |
| —     | —   | —   | —   | —   | —   | RTSECSEL <sup>(1)</sup> | 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 15-2 **Unimplemented:** Read as '0'

bit 1 **RTSECSEL:** RTCC Seconds Clock Output Select bit<sup>(1)</sup>

1 = RTCC seconds clock is selected for the RTCC pin

0 = RTCC alarm pulse is selected for the RTCC pin

bit 0 **PMPTTL:** PMP Module TTL Input Buffer Select bit

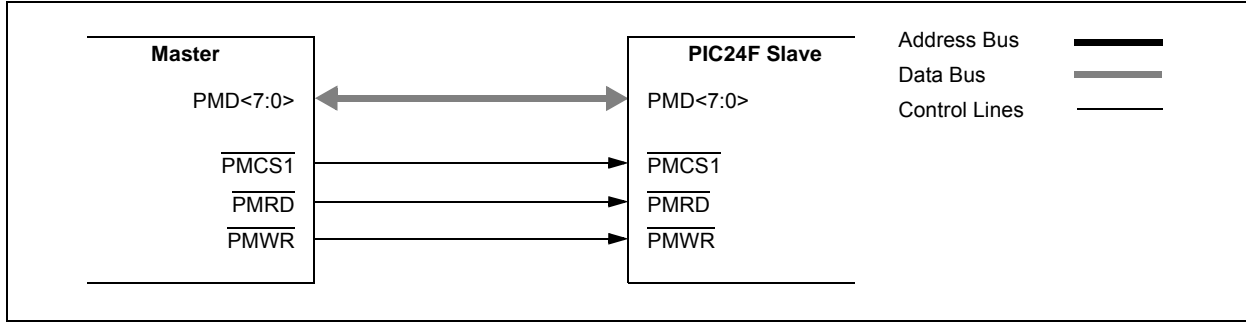
1 = PMP module uses TTL input buffers

0 = PMP module uses Schmitt Trigger input buffers

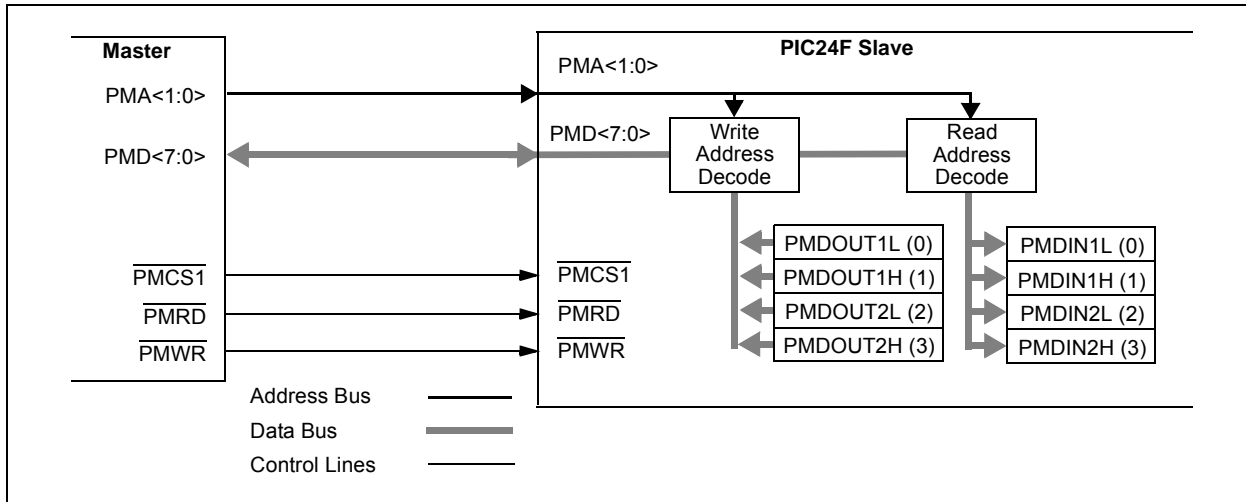
**Note 1:** To enable the actual RTCC output, the RTCOE (RCFGCAL) bit needs to be set.

# PIC24FJ64GA004 FAMILY

**FIGURE 18-2: LEGACY PARALLEL SLAVE PORT EXAMPLE**



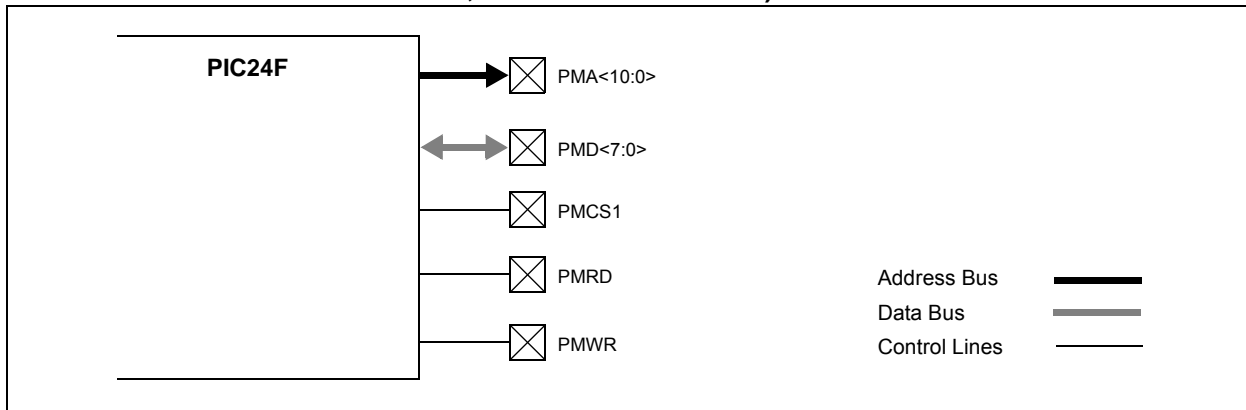
**FIGURE 18-3: ADDRESSABLE PARALLEL SLAVE PORT EXAMPLE**



**TABLE 18-1: SLAVE MODE ADDRESS RESOLUTION**

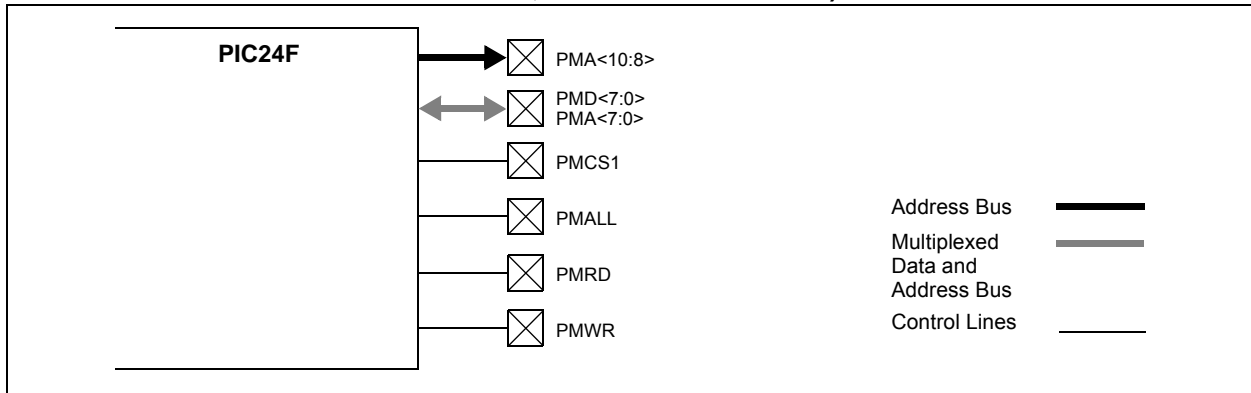
| PMA<1:0> | Output Register (Buffer) | Input Register (Buffer) |
|----------|--------------------------|-------------------------|
| 00       | PMDOUT1<7:0> (0)         | PMDIN1<7:0> (0)         |
| 01       | PMDOUT1<15:8> (1)        | PMDIN1<15:8> (1)        |
| 10       | PMDOUT2<7:0> (2)         | 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)**

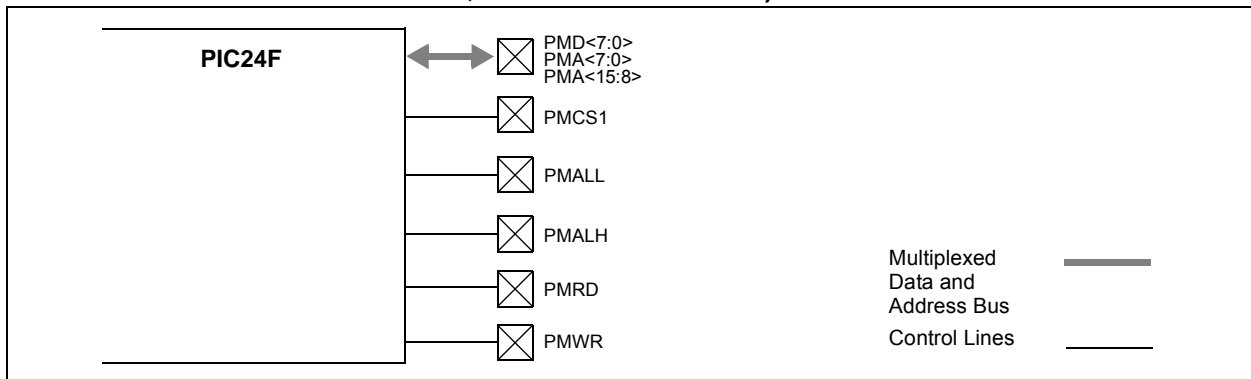


# PIC24FJ64GA004 FAMILY

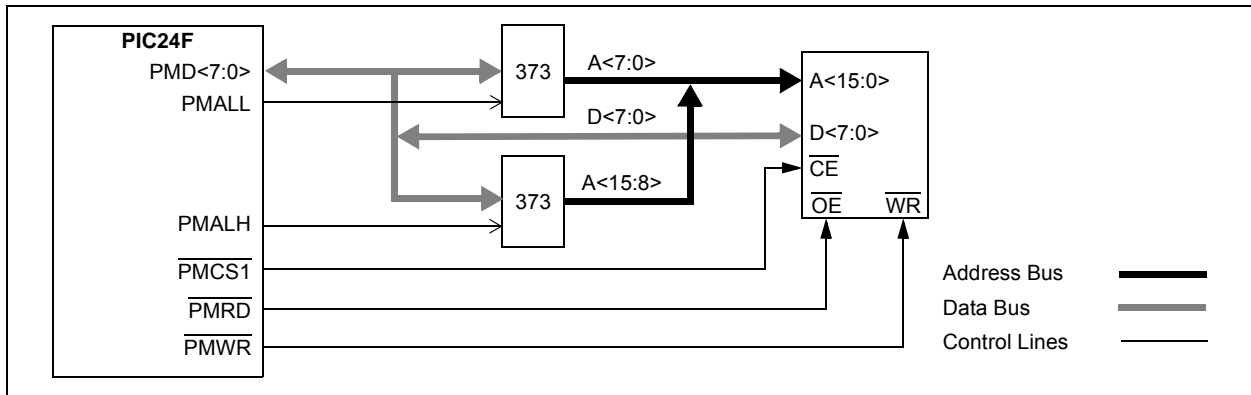
**FIGURE 18-5: MASTER MODE, PARTIALLY MULTIPLEXED ADDRESSING (SEPARATE READ AND WRITE STROBES, SINGLE CHIP SELECT)**



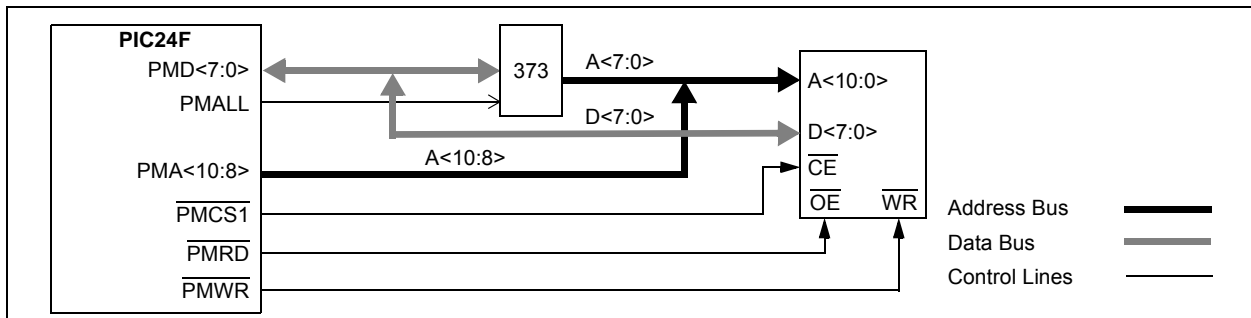
**FIGURE 18-6: MASTER MODE, FULLY MULTIPLEXED ADDRESSING (SEPARATE READ AND WRITE STROBES, SINGLE CHIP SELECT)**



**FIGURE 18-7: EXAMPLE OF A MULTIPLEXED ADDRESSING APPLICATION**

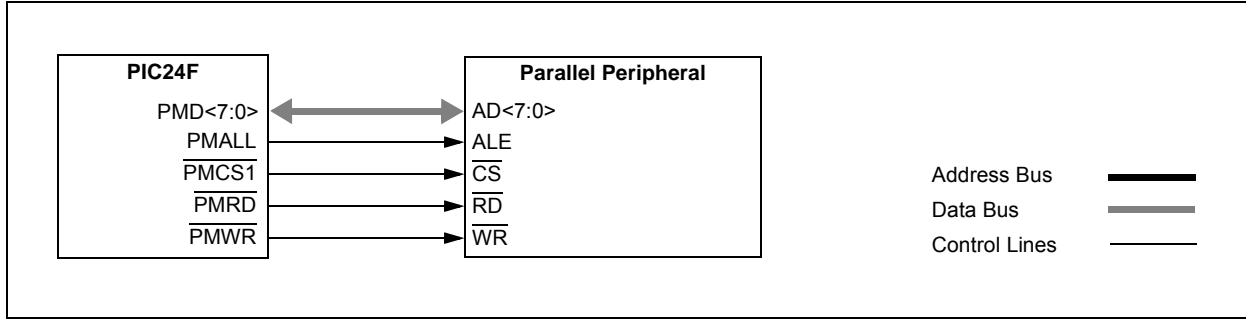


**FIGURE 18-8: EXAMPLE OF A PARTIALLY MULTIPLEXED ADDRESSING APPLICATION**

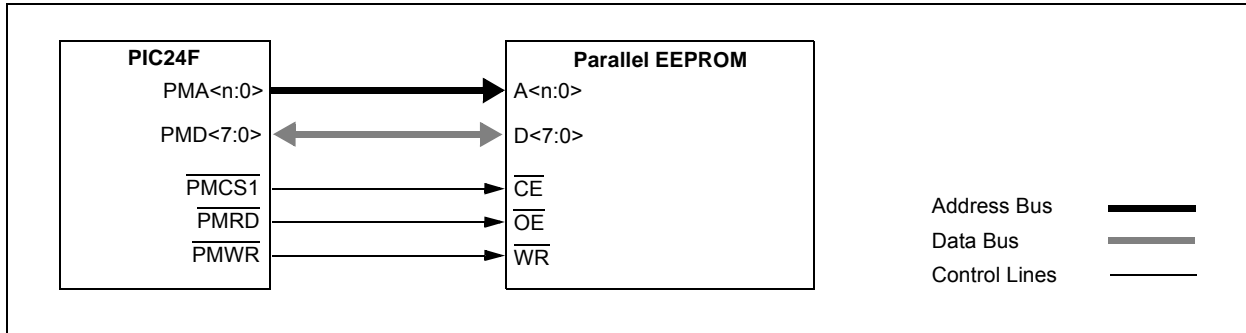


# PIC24FJ64GA004 FAMILY

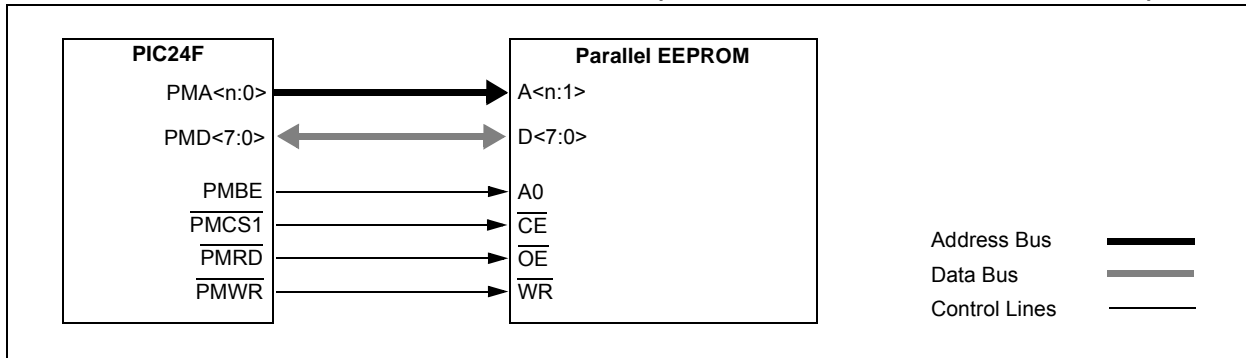
**FIGURE 18-9: EXAMPLE OF AN 8-BIT MULTIPLEXED ADDRESS AND DATA APPLICATION**



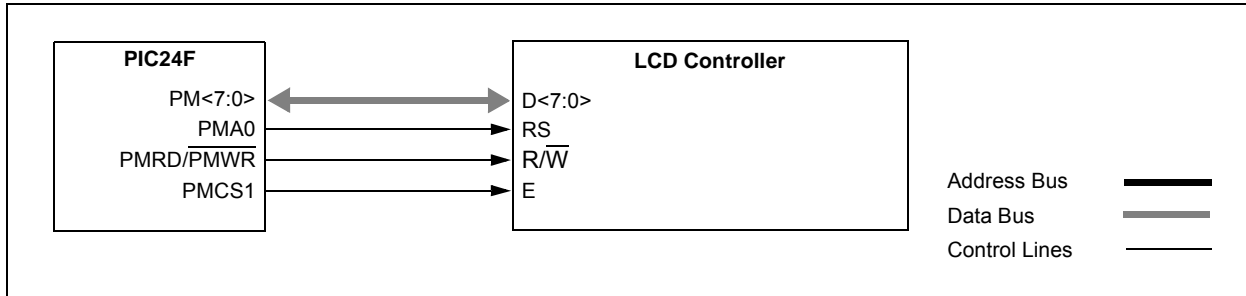
**FIGURE 18-10: PARALLEL EEPROM EXAMPLE (UP TO 11-BIT ADDRESS, 8-BIT DATA)**



**FIGURE 18-11: PARALLEL EEPROM EXAMPLE (UP TO 11-BIT ADDRESS, 16-BIT DATA)**



**FIGURE 18-12: LCD CONTROL EXAMPLE (BYTE MODE OPERATION)**

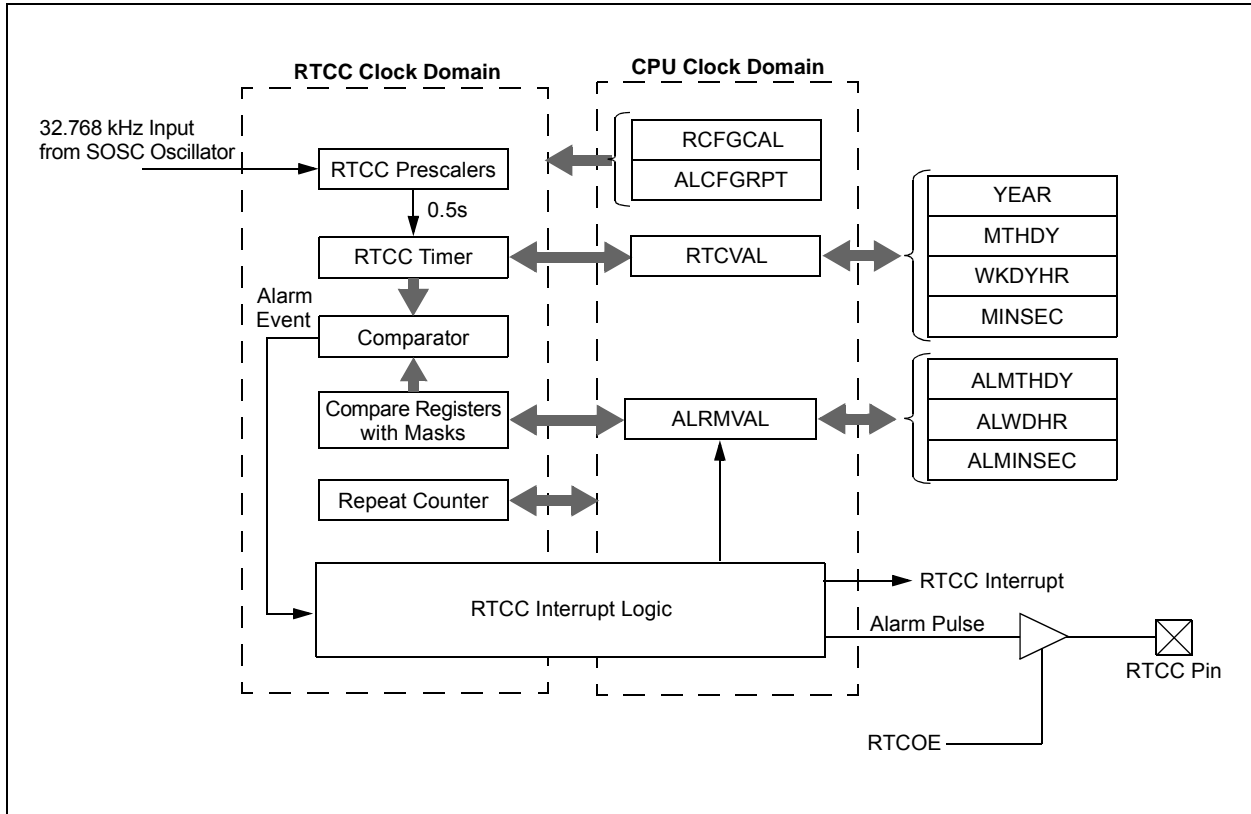




## 19.0 REAL-TIME CLOCK AND CALENDAR (RTCC)

**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).

**FIGURE 19-1: RTCC BLOCK DIAGRAM**



# PIC24FJ64GA004 FAMILY

## 19.1 RTCC Module Registers

The RTCC module registers are organized into three categories:

- RTCC Control Registers
- RTCC Value Registers
- Alarm Value Registers

### 19.1.1 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 the RTCVALH byte, the RTCC Pointer value, RTCPTR<1:0> bits, decrement 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<br><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 the ALRMVALH byte, the Alarm Pointer value, ALRMPTR<1:0> bits, decrement 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.

### 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");
```

**TABLE 19-2: ALRMVAL REGISTER MAPPING**

| ALRMPTR<br><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 will decrement the ALRMPTR<1:0> value. 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.

### 19.1.2 WRITE LOCK

In order 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 1 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.

# PIC24FJ64GA004 FAMILY

## 19.1.3 RTCC CONTROL REGISTERS

**REGISTER 19-1: RCFGAL: RTCC CALIBRATION AND CONFIGURATION REGISTER<sup>(1)</sup>**

|                      |     |         |         |                        |       |         |         |
|----------------------|-----|---------|---------|------------------------|-------|---------|---------|
| R/W-0                | U-0 | R/W-0   | R-0     | R-0                    | R/W-0 | R/W-0   | R/W-0   |
| RTCEN <sup>(2)</sup> | —   | RTCWREN | RTCSYNC | HALFSEC <sup>(3)</sup> | 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:**

|                   |                  |                                    |
|-------------------|------------------|------------------------------------|
| R = Readable bit  | W = 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<sup>(2)</sup>  
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<sup>(3)</sup>  
1 = Second half period of a second  
0 = First half period of a second
- bit 10      **RTCOE:** RTCC Output Enable bit  
1 = RTCC output enabled  
0 = RTCC output disabled
- bit 9-8     **RTCPTR1:RTCPTR0:** 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 RCFGAL register is only affected by a POR.  
**Note 2:** A write to the RTCEN bit is only allowed when RTCWREN = 1.  
**Note 3:** This bit is read-only. It is cleared to '0' on a write to the lower half of the MINSEC register.

# PIC24FJ64GA004 FAMILY

## REGISTER 19-1: RCFGAL: RTCC CALIBRATION AND CONFIGURATION REGISTER<sup>(1)</sup>

bit 7-0      **CAL7:CAL0:** 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 RCFGAL register is only affected by a POR.  
**Note 2:** A write to the RTCEN bit is only allowed when RTCWREN = 1.  
**Note 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 | U-0 | R/W-0                   | R/W-0  |
| —     | —   | —   | —   | —   | —   | RTSECSEL <sup>(1)</sup> | 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 15-2      **Unimplemented:** Read as '0'  
 bit 1      **RTSECSEL:** RTCC Seconds Clock Output Select bit<sup>(1)</sup>  
 1 = RTCC seconds clock is selected for the RTCC pin  
 0 = 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 (RCFGAL) bit needs to be set.

# PIC24FJ64GA004 FAMILY

## 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    **AMASK3:AMASK0:** 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        **ALRMPTR1:ALRMPTR0:** Alarm Value Register Window Pointer bits  
 Points to the corresponding Alarm Value registers when reading 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        **ARPT7:ARPT0:** Alarm Repeat Counter Value bits  
 11111111 = Alarm will repeat 255 more times  
 ...  
 00000000 = Alarm will not repeat  
 The counter decrements on any alarm event. The counter is prevented from rolling over from 00h to FFh unless CHIME = 1.

# PIC24FJ64GA004 FAMILY

## 19.1.4 RTCVAL REGISTER MAPPINGS

### REGISTER 19-4: YEAR: YEAR VALUE REGISTER<sup>(1)</sup>

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

|        |        |        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|--------|--------|
| R/W-x  | R/W-x  | R/W-x  | R/W-x  | R/W-x  | R/W-x  | R/W-x  | R/W-x  |
| YRTEN3 | YRTEN2 | YRTEN1 | YRTEN0 | YRONE3 | YRONE2 | YRONE1 | YRONE0 |
| 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 15-8            **Unimplemented:** Read as '0'
- bit 7-4            **YRTEN3:YRTEN0:** Binary Coded Decimal Value of Year's Tens Digit; Contains a value from 0 to 9
- bit 3-0            **YRONE3:YRONE0:** Binary Coded Decimal Value of Year's Ones Digit; 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<sup>(1)</sup>

|        |     |     |         |         |         |         |         |
|--------|-----|-----|---------|---------|---------|---------|---------|
| U-0    | U-0 | U-0 | R-x     | R-x     | R-x     | R-x     | R-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; Contains a value of '0' or '1'
- bit 11-8            **MTHONE3:MTHONE0:** Binary Coded Decimal Value of Month's Ones Digit; Contains a value from 0 to 9
- bit 7-6            **Unimplemented:** Read as '0'
- bit 5-4            **DAYTEN1:DAYTEN0:** Binary Coded Decimal Value of Day's Tens Digit; Contains a value from 0 to 3
- bit 3-0            **DAYONE3:DAYONE0:** Binary Coded Decimal Value of Day's Ones Digit; Contains a value from 0 to 9

**Note 1:** A write to this register is only allowed when RTCWREN = 1.

# PIC24FJ64GA004 FAMILY

## REGISTER 19-6: WKDYHR: WEEKDAY AND HOURS VALUE REGISTER<sup>(1)</sup>

|        |     |     |     |     |       |       |       |
|--------|-----|-----|-----|-----|-------|-------|-------|
| 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      **WDAY2:WDAY0:** Binary Coded Decimal Value of Weekday Digit; Contains a value from 0 to 6
- bit 7-6        **Unimplemented:** Read as '0'
- bit 5-4        **HRTEN1:HRTEN0:** Binary Coded Decimal Value of Hour's Tens Digit; Contains a value from 0 to 2
- bit 3-0        **HRONE3:HRONE0:** Binary Coded Decimal Value of Hour's Ones Digit; 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    | 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 | SECCONE3 | SECCONE2 | SECCONE1 | SECCONE0 |
| 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    **MINTEN2:MINTEN0:** Binary Coded Decimal Value of Minute's Tens Digit; Contains a value from 0 to 5
- bit 11-8     **MINONE3:MINONE0:** Binary Coded Decimal Value of Minute's Ones Digit; Contains a value from 0 to 9
- bit 7         **Unimplemented:** Read as '0'
- bit 6-4      **SECTEN2:SECTEN0:** Binary Coded Decimal Value of Second's Tens Digit; Contains a value from 0 to 5
- bit 3-0      **SECCONE3:SECCONE0:** Binary Coded Decimal Value of Second's Ones Digit; Contains a value from 0 to 9

# PIC24FJ64GA004 FAMILY

## 19.1.5 ALRMVAL REGISTER MAPPINGS

### REGISTER 19-8: ALMTHDY: ALARM MONTH AND DAY VALUE REGISTER<sup>(1)</sup>

|        |     |     |         |         |         |         |         |
|--------|-----|-----|---------|---------|---------|---------|---------|
| U-0    | U-0 | U-0 | R/W-x   | R/W-x   | R/W-x   | R/W-x   | R/W-x   |
| —      | —   | —   | MHTTEN0 | 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        **MHTTEN0:** Binary Coded Decimal Value of Month's Tens Digit; Contains a value of '0' or '1'
- bit 11-8      **MTHONE3:MTHONE0:** Binary Coded Decimal Value of Month's Ones Digit; Contains a value from 0 to 9
- bit 7-6       **Unimplemented:** Read as '0'
- bit 5-4       **DAYTEN1:DAYTEN0:** Binary Coded Decimal Value of Day's Tens Digit; Contains a value from 0 to 3
- bit 3-0       **DAYONE3:DAYONE0:** Binary Coded Decimal Value of Day's Ones Digit; 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<sup>(1)</sup>

|        |     |     |     |     |       |       |       |
|--------|-----|-----|-----|-----|-------|-------|-------|
| 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      **WDAY2:WDAY0:** Binary Coded Decimal Value of Weekday Digit; Contains a value from 0 to 6
- bit 7-6       **Unimplemented:** Read as '0'
- bit 5-4       **HRTEN1:HRTEN0:** Binary Coded Decimal Value of Hour's Tens Digit; Contains a value from 0 to 2
- bit 3-0       **HRONE3:HRONE0:** Binary Coded Decimal Value of Hour's Ones Digit; Contains a value from 0 to 9

**Note 1:** A write to this register is only allowed when RTCWREN = 1.



# PIC24FJ64GA004 FAMILY

## 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                **MINTEN2:MINTEN0:** Binary Coded Decimal Value of Minute's Tens Digit; Contains a value from 0 to 5
- bit 11-8                 **MINONE3:MINONE0:** Binary Coded Decimal Value of Minute's Ones Digit; Contains a value from 0 to 9
- bit 7                      **Unimplemented:** Read as '0'
- bit 6-4                  **SECTEN2:SECTEN0:** Binary Coded Decimal Value of Second's Tens Digit; Contains a value from 0 to 5
- bit 3-0                  **SECONE3:SECONE0:** Binary Coded Decimal Value of Second's Ones Digit; Contains a value from 0 to 9

## 19.2 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 RCFGAL register. The 8-bit signed value loaded into the lower half of RCFGAL is multiplied by four and will be either 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.

### EQUATION 19-1:

$$(\text{Ideal Frequency}^\dagger - \text{Measured Frequency}) * 60 = \text{Clocks per Minute}$$

$$^\dagger \text{ Ideal frequency} = 32,768 \text{ Hz}$$

3. a) If the oscillator is faster than ideal (negative result from step 2), the RCFGAL register value needs to 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 RCFGAL register value needs to be positive. This causes the specified number of clock pulses to be subtracted from the timer counter once every minute.
4. Divide the number of error clocks per minute by 4 to get the correct CAL value and load the RCFGAL register with the correct value.  
 (Each 1-bit increment in CAL adds or subtracts 4 pulses).

Writes to the lower half of the RCFGAL 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.

# PIC24FJ64GA004 FAMILY

## 19.3 Alarm

- Configurable from half second to one year
- Enabled using the ALRMEN bit (ALCFGRPT<15>, Register 19-3)
- One-time alarm and repeat alarm options available

### 19.3.1 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 shown 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 bits, ARPT7:ARPT0 (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 ARPT7:ARPT0 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.

### 19.3.2 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 RCFGAL 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**

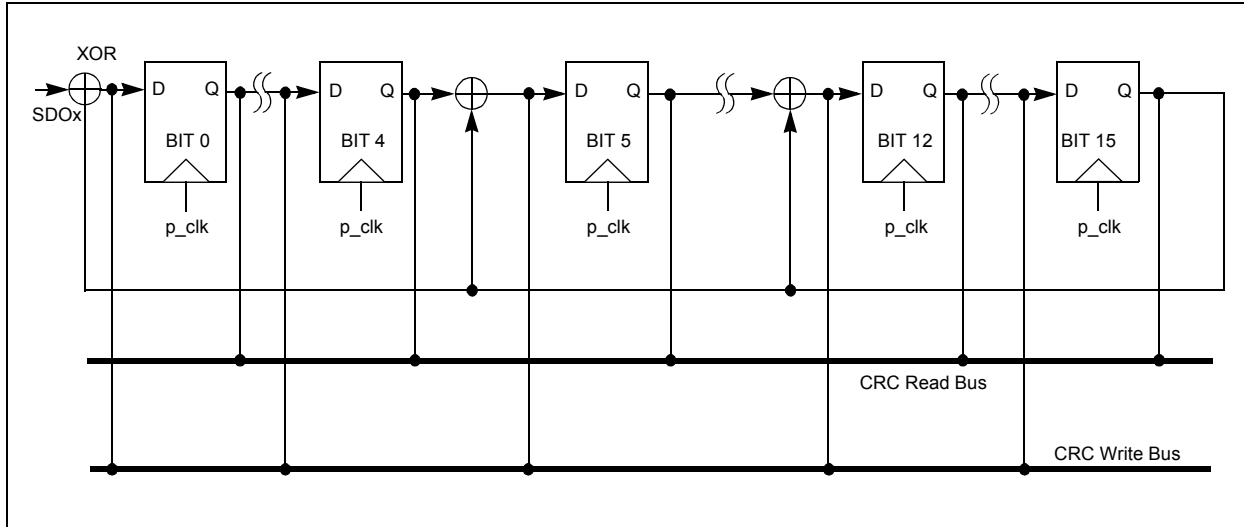
| Alarm Mask Setting<br>(AMASK3:AMASK0)           | Day of<br>the<br>Week      | Month   | Day   | Hours   | Minutes   | Seconds   |
|---|----------------------------|---|---|---|---|---|
| 0000 – Every half second<br>0001 – Every second | <input type="checkbox"/>   | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> <input type="checkbox"/>     |
| 0010 – Every 10 seconds                         | <input type="checkbox"/>   | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> s                            |
| 0011 – Every minute                             | <input type="checkbox"/>   | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> s <input type="checkbox"/> s |
| 0100 – Every 10 minutes                         | <input type="checkbox"/>   | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> m                            | <input type="checkbox"/> s <input type="checkbox"/> s |
| 0101 – Every hour                               | <input type="checkbox"/>   | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> m <input type="checkbox"/> m | <input type="checkbox"/> s <input type="checkbox"/> s |
| 0110 – Every day                                | <input type="checkbox"/>   | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> h <input type="checkbox"/> h | <input type="checkbox"/> m <input type="checkbox"/> m | <input type="checkbox"/> s <input type="checkbox"/> s |
| 0111 – Every week                               | <input type="checkbox"/> d | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> h <input type="checkbox"/> h | <input type="checkbox"/> m <input type="checkbox"/> m | <input type="checkbox"/> s <input type="checkbox"/> s |
| 1000 – Every month                              | <input type="checkbox"/>   | <input type="checkbox"/> <input type="checkbox"/>     | <input type="checkbox"/> d <input type="checkbox"/> d | <input type="checkbox"/> h <input type="checkbox"/> h | <input type="checkbox"/> m <input type="checkbox"/> m | <input type="checkbox"/> s <input type="checkbox"/> s |
| 1001 – Every year <sup>(1)</sup>                | <input type="checkbox"/>   | <input type="checkbox"/> m <input type="checkbox"/> m | <input type="checkbox"/> d <input type="checkbox"/> d | <input type="checkbox"/> h <input type="checkbox"/> h | <input type="checkbox"/> m <input type="checkbox"/> m | <input type="checkbox"/> s <input type="checkbox"/> s |

**Note 1:** Annually, except when configured for February 29.



# PIC24FJ64GA004 FAMILY

FIGURE 20-2: CRC GENERATOR RECONFIGURED FOR  $x^{16} + x^{12} + x^5 + 1$



## 20.1 User Interface

### 20.1.1 DATA INTERFACE

To start serial shifting, a '1' must be written to the CRCGO bit.

The module incorporates a FIFO that is 8 deep when  $PLEN (PLEN < 3:0 >) > 7$ , and 16 deep, otherwise. The data for which the CRC is to be calculated must first be written into the FIFO. The smallest data element that can be written into the FIFO is one byte. For example, if  $PLEN = 5$ , then the size of the data is  $PLEN + 1 = 6$ . The data must be written as follows:

```
data[5:0] = crc_input[5:0]
data[7:6] = 'bxxx'
```

Once data is written into the CRCWDAT MSb (as defined by  $PLEN$ ), the value of the  $VWORD$  bits ( $CRCCON < 12:8 >$ ) increments by one. The serial shifter starts shifting data into the CRC engine when  $CRCGO = 1$  and  $VWORD > 0$ . When the MSb is shifted out,  $VWORD$  decrements by one. The serial shifter continues shifting until the  $VWORD$  reaches 0. Therefore, for a given value of  $PLEN$ , it will take  $(PLEN + 1) * VWORD$  number of clock cycles to complete the CRC calculations.

When  $VWORD$  reaches 8 (or 16), the  $CRCFUL$  bit will be set. When  $VWORD$  reaches 0, the  $CRCMPT$  bit will be set.

To continually feed data into the CRC engine, the recommended mode of operation is to initially "prime" the FIFO with a sufficient number of words so no interrupt is generated before the next word can be written. Once that is done, start the CRC by setting the  $CRCGO$  bit to '1'. From that point onward, the  $VWORD$  bits should be polled. If they read less than 8 or 16, another word can be written into the FIFO.

To empty words already written into a FIFO, the  $CRCGO$  bit must be set to '1' and the CRC shifter allowed to run until the  $CRCMPT$  bit is set.

Also, to get the correct CRC reading, it will be necessary to wait for the  $CRCMPT$  bit to go high before reading the  $CRCWDAT$  register.

If a word is written when the  $CRCFUL$  bit is set, the  $VWORD$  Pointer will roll over to 0. The hardware will then behave as if the FIFO is empty. However, the condition to generate an interrupt will not be met; therefore, no interrupt will be generated (See **Section 20.1.2 "Interrupt Operation"**).

At least one instruction cycle must pass after a write to  $CRCWDAT$  before a read of the  $VWORD$  bits is done.

### 20.1.2 INTERRUPT OPERATION

When the  $VWORD4:VWORD0$  bits make a transition from a value of '1' to '0', an interrupt will be generated.

## 20.2 Operation in Power Save Modes

### 20.2.1 SLEEP MODE

If Sleep mode is entered while the module is operating, the module will be suspended in its current state until clock execution resumes.

### 20.2.2 IDLE MODE

To continue full module operation in Idle mode, the  $CSIDL$  bit must be cleared prior to entry into the mode.

If  $CSIDL = 1$ , the module will behave the same way as it does in Sleep mode; pending interrupt events will be passed on, even though the module clocks are not available.

# PIC24FJ64GA004 FAMILY

## 20.3 Registers

There are four registers used to control programmable CRC operation:

- CRCCON
- CRCXOR
- CRCDAT
- CRCWDAT

### REGISTER 20-1: CRCCON: CRC CONTROL REGISTER

|        |     |       |        |        |        |        |        |
|--------|-----|-------|--------|--------|--------|--------|--------|
| U-0    | U-0 | R/W-0 | R-0    | R-0    | R-0    | R-0    | R-0    |
| —      | —   | CSIDL | VWORD4 | VWORD3 | VWORD2 | VWORD1 | VWORD0 |
| bit 15 |     |       |        |        |        |        | bit 8  |

|        |        |     |       |       |       |       |       |
|--------|--------|-----|-------|-------|-------|-------|-------|
| R-0    | R-1    | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| CRCFUL | CRCMPT | —   | CRCGO | PLEN3 | PLEN2 | PLEN1 | PLEN0 |
| bit 7  |        |     |       |       |       |       | bit 0 |

#### Legend:

R = Readable bit  
-n = Value at POR

W = Writable bit  
'1' = Bit is set

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

bit 15-14 **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 **VWORD4:VWORD0:** Pointer Value bits  
Indicates the number of valid words in the FIFO. Has a maximum value of 8 when PLEN3:PLEN0 > 7, or 16 when PLEN3:PLEN0 ≤ 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 **Unimplemented:** Read as '0'

bit 4 **CRCGO:** Start CRC bit  
1 = Start CRC serial shifter  
0 = CRC serial shifter turned off

bit 3-0 **PLEN3:PLEN0:** Polynomial Length bits  
Denotes the length of the polynomial to be generated minus 1.

# PIC24FJ64GA004 FAMILY

## REGISTER 20-2: CRCXOR: CRC XOR POLYNOMIAL 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 |
| 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      **X15:X1:** XOR of Polynomial Term  $X^n$  Enable bits

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

## 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
- Up to 13 analog input pins
- External voltage reference input pins
- 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

Depending on the particular device pinout, the 10-bit A/D Converter can have up to three analog input pins, designated AN0 through AN12. In addition, there are two analog input pins for external voltage reference connections. These voltage reference inputs may be shared with other analog input pins. The actual number of analog input pins and external voltage reference input configuration will depend on the specific device.

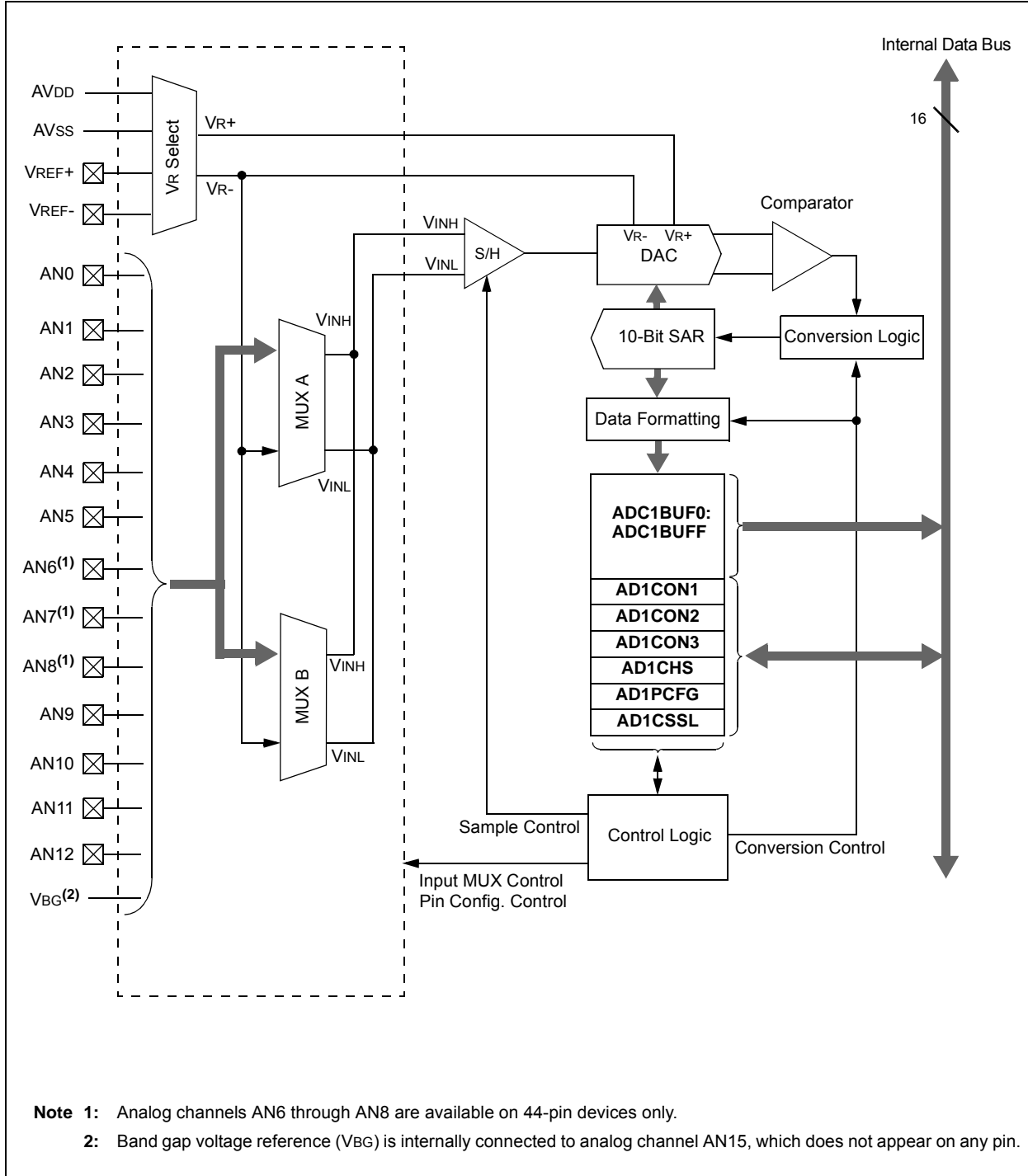
A block diagram of the A/D Converter is shown in Figure 21-1.

To perform an A/D conversion:

1. Configure the A/D module:
  - a) Select port pins as analog inputs (AD1PCFG<15:0>).
  - b) Select voltage reference source to match expected range on analog inputs (AD1CON2<15:13>).
  - c) Select the analog conversion clock to match desired data rate with 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 A/D interrupt (if required):
  - a) Clear the AD1IF bit.
  - b) Select A/D interrupt priority.

# PIC24FJ64GA004 FAMILY

**FIGURE 21-1: 10-BIT HIGH-SPEED A/D CONVERTER BLOCK DIAGRAM**





# PIC24FJ64GA004 FAMILY

## REGISTER 21-1: AD1CON1: A/D CONTROL REGISTER 1

|        |     |        |     |     |     |       |       |
|--------|-----|--------|-----|-----|-----|-------|-------|
| R/W-0  | U-0 | R/C-0  | U-0 | U-0 | U-0 | R/W-0 | R/W-0 |
| ADON   | —   | 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/W-0, HCS |
| SSRC2 | SSRC1 | SSRC0 | —   | —   | ASAM  | SAMP       | DONE       |
| bit 7 |       |       |     |     |       | bit 0      |            |

|                   |                   |                                       |
|-------------------|-------------------|---------------------------------------|
| <b>Legend:</b>    | 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 = 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     **FORM1:FORM0:** 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     **SSRC2:SSRC0:** Conversion Trigger Source Select bits  
           111 = Internal counter ends sampling and starts conversion (auto-convert)  
           110 = Reserved  
           10x = Reserved  
           011 = Reserved  
           010 = Timer3 compare ends sampling and starts conversion  
           001 = Active transition on INT0 pin ends sampling and starts conversion  
           000 = Clearing 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 last conversion completes. SAMP bit is auto-set.  
           0 = Sampling begins when 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

# PIC24FJ64GA004 FAMILY

## REGISTER 21-2: AD1CON2: A/D CONTROL REGISTER 2

|        |       |       |       |     |       |       |     |
|--------|-------|-------|-------|-----|-------|-------|-----|
| R/W-0  | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | U-0   | U-0 |
| VCFG2  | VCFG1 | VCFG0 | —     | —   | 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 = Readable bit                      W = 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      **VCFG2:VCFG0:** Voltage Reference Configuration bits

| VCFG2:VCFG0 | 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*              |

\* AVDD and AVSS inputs are tied to VDD and VSS on 28-pin devices.

bit 12-11      **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      **SMPI3:SMPI0:** 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 configured as two 8-word buffers (ADC1BUFn<15:8> and ADC1BUFn<7:0>)  
0 = Buffer 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

# PIC24FJ64GA004 FAMILY

## REGISTER 21-3: AD1CON3: A/D CONTROL REGISTER 3

|        |     |     |       |       |       |       |       |
|--------|-----|-----|-------|-------|-------|-------|-------|
| R/W-0  | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| ADRC   | —   | —   | 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 |

|                   |                  |                                    |                    |
|-------------------|------------------|------------------------------------|--------------------|
| <b>Legend:</b>    |                  |                                    |                    |
| R = Readable bit  | W = 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    **Unimplemented:** Read as '0'

bit 12-8     **SAMC4:SAMC0:** Auto-Sample Time bits  
             11111 = 31 TAD  
             .....  
             00001 = 1 TAD  
             00000 = 0 TAD (not recommended)

bit 7-0      **ADCS7:ADCS0:** A/D Conversion Clock Select bits  
             11111111  
             ..... = Reserved  
             01000000  
             00111111 = 64 • Tcy  
             .....  
             00000001 = 2 • Tcy  
             00000000 = Tcy

# PIC24FJ64GA004 FAMILY

## REGISTER 21-4: AD1CHS: A/D INPUT SELECT REGISTER

|        |     |     |     |                         |                         |                         |                         |
|--------|-----|-----|-----|-------------------------|-------------------------|-------------------------|-------------------------|
| R/W-0  | U-0 | U-0 | U-0 | R/W-0                   | R/W-0                   | R/W-0                   | R/W-0                   |
| CH0NB  | —   | —   | —   | CH0SB3 <sup>(1,2)</sup> | CH0SB2 <sup>(1,2)</sup> | CH0SB1 <sup>(1,2)</sup> | CH0SB0 <sup>(1,2)</sup> |
| bit 15 |     |     |     |                         |                         |                         | bit 8                   |

|       |     |     |     |                         |                         |                         |                         |
|-------|-----|-----|-----|-------------------------|-------------------------|-------------------------|-------------------------|
| R/W-0 | U-0 | U-0 | U-0 | R/W-0                   | R/W-0                   | R/W-0                   | R/W-0                   |
| CH0NA | —   | —   | —   | CH0SA3 <sup>(1,2)</sup> | CH0SA2 <sup>(1,2)</sup> | CH0SA1 <sup>(1,2)</sup> | CH0SA0 <sup>(1,2)</sup> |
| bit 7 |     |     |     |                         |                         |                         | bit 0                   |

### Legend:

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

- bit 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 VR-
- bit 14-12        **Unimplemented:** Read as '0'
- bit 11-8        **CH0SB3:CH0SB0:** Channel 0 Positive Input Select for MUX B Multiplexer Setting bits<sup>(1,2)</sup>  
                   1111 = Channel 0 positive input is AN15 (band gap voltage reference)  
                   1100 = Channel 0 positive input is AN12  
                   1011 = Channel 0 positive input is AN11  
                   .....  
                   0001 = Channel 0 positive input is AN1  
                   0000 = 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 VR-
- bit 6-4        **Unimplemented:** Read as '0'
- bit 3-0        **CH0SA3:CH0SA0:** Channel 0 Positive Input Select for MUX A Multiplexer Setting bits<sup>(1,2)</sup>  
                   1111 = Channel 0 positive input is AN15 (band gap voltage reference)  
                   1100 = Channel 0 positive input is AN12  
                   1011 = Channel 0 positive input is AN11  
                   .....  
                   0001 = Channel 0 positive input is AN1  
                   0000 = Channel 0 positive input is AN0

- Note 1:** Combinations '1101' and '1110' are unimplemented; do not use.  
**Note 2:** Analog channels AN6, AN7 and AN8 are unavailable on 28-pin devices; do not use.

# PIC24FJ64GA004 FAMILY

## REGISTER 21-5: AD1PCFG: A/D PORT CONFIGURATION REGISTER

|        |     |     |        |        |        |       |                      |
|--------|-----|-----|--------|--------|--------|-------|----------------------|
| R/W-0  | U-0 | U-0 | R/W-0  | R/W-0  | R/W-0  | R/W-0 | R/W-0                |
| PCFG15 | —   | —   | PCFG12 | PCFG11 | PCFG10 | PCFG9 | PCFG8 <sup>(1)</sup> |
| 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 |
| PCFG7 <sup>(1)</sup> | PCFG6 <sup>(1)</sup> | 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:** Analog Input Pin Configuration Control bits

1 = Band gap voltage reference is disabled  
 0 = Band gap voltage reference enabled

bit 14-13                      **Unimplemented:** Read as '0'

bit 12-0                      **PCFG12:PCFG0:** Analog Input Pin Configuration Control bits<sup>(1)</sup>

1 = Pin for corresponding analog channel is configured in Digital mode; I/O port read enabled  
 0 = Pin configured in Analog mode; I/O port read disabled, A/D samples pin voltage

**Note 1:** Analog channels AN6, AN7 and AN8 are unavailable on 28-pin devices; leave these corresponding bits set.

## REGISTER 21-6: AD1CSSL: A/D INPUT SCAN SELECT REGISTER

|        |     |     |        |        |        |       |                      |
|--------|-----|-----|--------|--------|--------|-------|----------------------|
| R/W-0  | U-0 | U-0 | R/W-0  | R/W-0  | R/W-0  | R/W-0 | R/W-0                |
| CSSL15 | —   | —   | CSSL12 | CSSL11 | CSSL10 | CSSL9 | CSSL8 <sup>(1)</sup> |
| 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 <sup>(1)</sup> | CSSL6 <sup>(1)</sup> | 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:** Band Gap Reference Input Pin Scan Selection bits

1 = Band gap voltage reference channel selected for input scan  
 0 = Band gap voltage reference channel omitted from input scan

bit 14-13                      **Unimplemented:** Read as '0'

bit 12-0                      **CSSL12:CSSL0:** A/D Input Pin Scan Selection bits<sup>(1)</sup>

1 = Corresponding analog channel selected for input scan  
 0 = Analog channel omitted from input scan

**Note 1:** Analog channels AN6, AN7 and AN8 are unavailable on 28-pin devices; leave these corresponding bits cleared.

# PIC24FJ64GA004 FAMILY

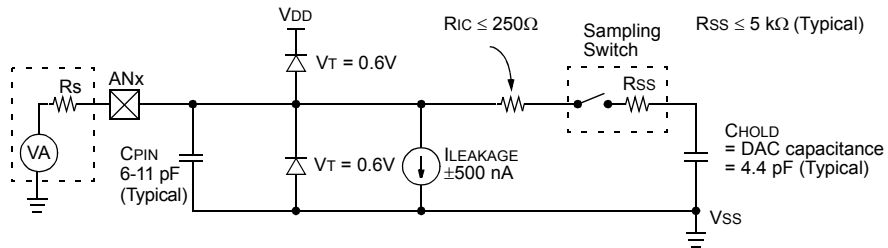
**EQUATION 21-1: A/D CONVERSION CLOCK PERIOD<sup>(1)</sup>**

$$T_{AD} = T_{CY} \cdot (ADCS + 1)$$

$$ADCS = \frac{T_{AD}}{T_{CY}} - 1$$

**Note 1:** Based on  $T_{CY} = 2 * T_{OSC}$ ; Doze mode and PLL are disabled.

**FIGURE 21-2: 10-BIT A/D CONVERTER ANALOG INPUT MODEL**



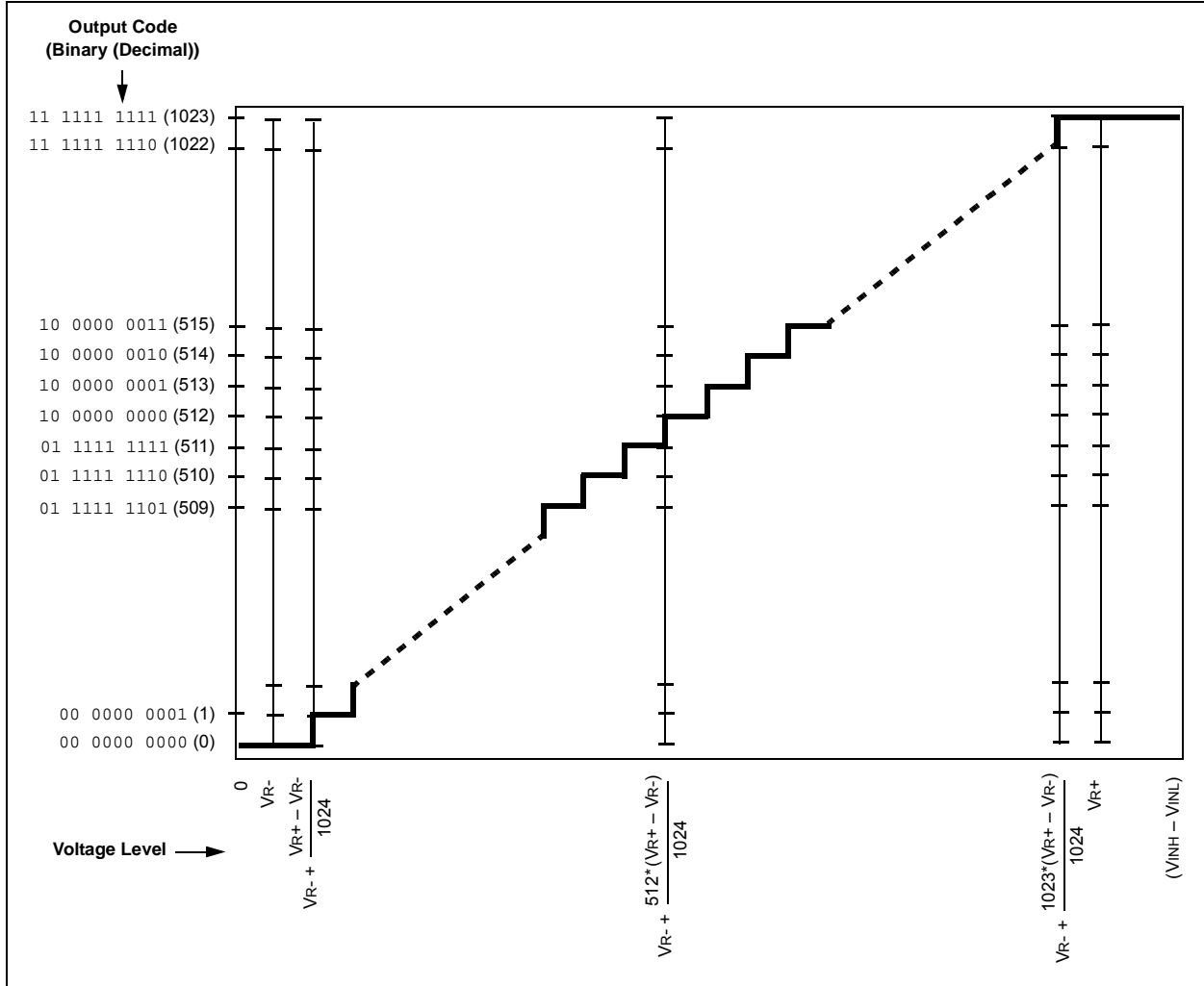
**Legend:**

- CPIN = Input Capacitance
- VT = Threshold Voltage
- ILEAKAGE = Leakage Current at the pin due to various junctions
- RIC = Interconnect Resistance
- RSS = Sampling Switch Resistance
- CHOLD = Sample/Hold Capacitance (from DAC)

**Note:** CPIN value depends on device package and is not tested. Effect of CPIN negligible if  $R_s \leq 5 \text{ k}\Omega$ .

# PIC24FJ64GA004 FAMILY

**FIGURE 21-3: A/D TRANSFER FUNCTION**



# PIC24FJ64GA004 FAMILY

---

NOTES:

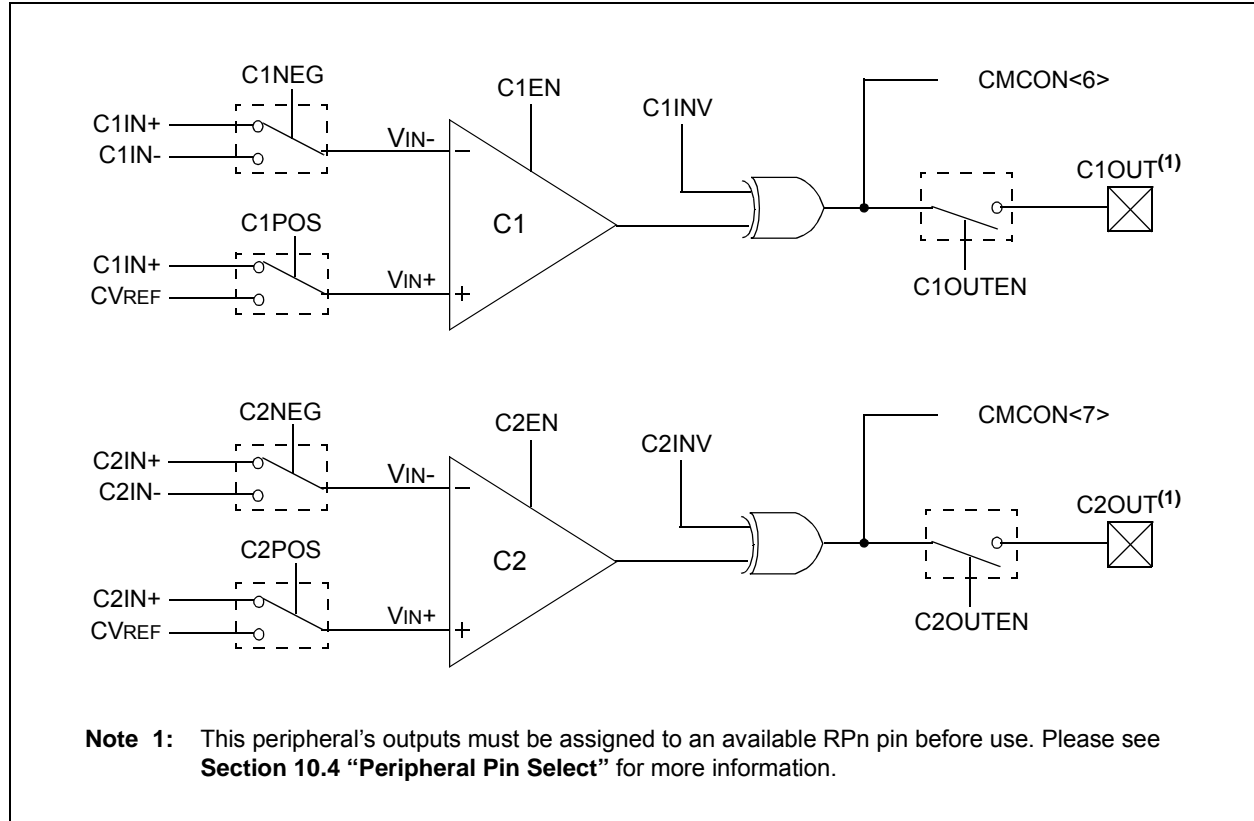


# PIC24FJ64GA004 FAMILY

## 22.0 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 "PIC24F Family Reference Manual", "Section 16. Output Compare" (DS39706).

FIGURE 22-1: COMPARATOR I/O OPERATING MODES



# PIC24FJ64GA004 FAMILY

## REGISTER 22-1: CMCON: COMPARATOR CONTROL REGISTER

|        |     |       |       |       |       |                        |                        |
|--------|-----|-------|-------|-------|-------|------------------------|------------------------|
| R/W-0  | U-0 | R/C-0 | R/C-0 | R/W-0 | R/W-0 | R/W-0                  | R/W-0                  |
| CMIDL  | —   | C2EVT | C1EVT | C2EN  | C1EN  | C2OUTEN <sup>(1)</sup> | C1OUTEN <sup>(2)</sup> |
| bit 15 |     |       |       |       |       | bit 8                  |                        |

|       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|
| R-0   | R-0   | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| C2OUT | C1OUT | C2INV | C1INV | C2NEG | C2POS | C1NEG | C1POS |
| 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:** Stop in Idle Mode bit  
 1 = When device enters Idle mode, module does not generate interrupts; module is still enabled  
 0 = Continue normal module operation in Idle mode
- bit 14      **Unimplemented:** Read as '0'
- bit 13      **C2EVT:** Comparator 2 Event  
 1 = Comparator output changed states  
 0 = Comparator output did not change states
- bit 12      **C1EVT:** Comparator 1 Event  
 1 = Comparator output changed states  
 0 = Comparator output did not change states
- bit 11      **C2EN:** Comparator 2 Enable  
 1 = Comparator is enabled  
 0 = Comparator is disabled
- bit 10      **C1EN:** Comparator 1 Enable  
 1 = Comparator is enabled  
 0 = Comparator is disabled
- bit 9        **C2OUTEN:** Comparator 2 Output Enable<sup>(1)</sup>  
 1 = Comparator output is driven on the output pad  
 0 = Comparator output is not driven on the output pad
- bit 8        **C1OUTEN:** Comparator 1 Output Enable<sup>(2)</sup>  
 1 = Comparator output is driven on the output pad  
 0 = Comparator output is not driven on the output pad
- bit 7        **C2OUT:** Comparator 2 Output bit  
When C2INV = 0:  
 1 = C2 VIN+ > C2 VIN-  
 0 = C2 VIN+ < C2 VIN-  
When C2INV = 1:  
 0 = C2 VIN+ > C2 VIN-  
 1 = C2 VIN+ < C2 VIN-
- bit 6        **C1OUT:** Comparator 1 Output bit  
When C1INV = 0:  
 1 = C1 VIN+ > C1 VIN-  
 0 = C1 VIN+ < C1 VIN-  
When C1INV = 1:  
 0 = C1 VIN+ > C1 VIN-  
 1 = C1 VIN+ < C1 VIN-

# PIC24FJ64GA004 FAMILY

---

## REGISTER 22-1: CMCON: COMPARATOR CONTROL REGISTER (CONTINUED)

|       |   |
|-------|---|
| bit 5 | <b>C2INV:</b> Comparator 2 Output Inversion bit<br>1 = C2 output inverted<br>0 = C2 output not inverted   |
| bit 4 | <b>C1INV:</b> Comparator 1 Output Inversion bit<br>1 = C1 output inverted<br>0 = C1 output not inverted   |
| bit 3 | <b>C2NEG:</b> Comparator 2 Negative Input Configure bit<br>1 = Input is connected to VIN+<br>0 = Input is connected to VIN-<br>See Figure 22-1 for the Comparator modes.  |
| bit 2 | <b>C2POS:</b> Comparator 2 Positive Input Configure bit<br>1 = Input is connected to VIN+<br>0 = Input is connected to CVREF<br>See Figure 22-1 for the Comparator modes. |
| bit 1 | <b>C1NEG:</b> Comparator 1 Negative Input Configure bit<br>1 = Input is connected to VIN+<br>0 = Input is connected to VIN-<br>See Figure 22-1 for the Comparator modes.  |
| bit 0 | <b>C1POS:</b> Comparator 1 Positive Input Configure bit<br>1 = Input is connected to VIN+<br>0 = Input is connected to CVREF<br>See Figure 22-1 for the Comparator modes. |

- Note 1:** If C2OUTEN = 1, the C2OUT peripheral output must be configured to an available RPn pin. See **Section 10.4 “Peripheral Pin Select”** for more information.
- 2:** If C1OUTEN = 1, the C1OUT peripheral output must be configured to an available RPn pin. See **Section 10.4 “Peripheral Pin Select”** for more information.

# PIC24FJ64GA004 FAMILY

---

NOTES:

# PIC24FJ64GA004 FAMILY

## 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

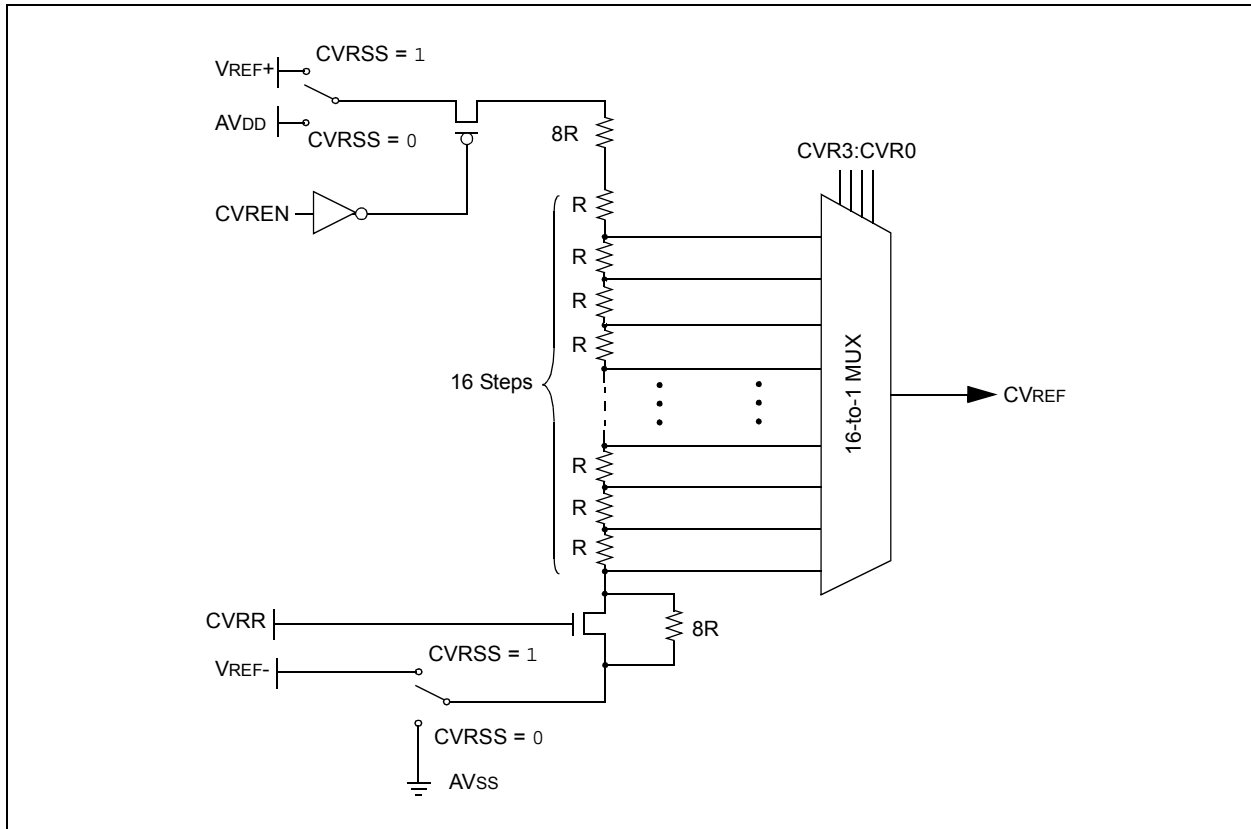
The voltage reference module is controlled through the CVRCON register (Register 23-1). The comparator voltage reference provides two ranges of output

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 (CVR3:CVR0), 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



# PIC24FJ64GA004 FAMILY

## REGISTER 23-1: CVRCON: COMPARATOR VOLTAGE REFERENCE CONTROL REGISTER

|        |     |     |     |     |     |       |     |
|--------|-----|-----|-----|-----|-----|-------|-----|
| U-0    | U-0 | U-0 | U-0 | U-0 | U-0 | U-0   | U-0 |
| —      | —   | —   | —   | —   | —   | —     | —   |
| 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 15-8 **Unimplemented:** Read as '0'

bit 7 **CVREN:** Comparator Voltage Reference Enable bit

1 = CVREF circuit powered on

0 = CVREF circuit powered down

bit 6 **CVROE:** Comparator VREF Output Enable bit

1 = CVREF voltage level is output on 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 **CVR3:CVR0:** Comparator VREF Value Selection  $0 \leq \text{CVR3:CVR0} \leq 15$  bits

When CVRR = 1:

$$\text{CVREF} = (\text{CVR}\langle 3:0 \rangle / 24) \cdot (\text{CVRSRC})$$

When CVRR = 0:

$$\text{CVREF} = 1/4 \cdot (\text{CVRSRC}) + (\text{CVR}\langle 3:0 \rangle / 32) \cdot (\text{CVRSRC})$$

# PIC24FJ64GA004 FAMILY

## 24.0 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)

PIC24FJ64GA004 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

### 24.1 Configuration Bits

The Configuration bits can be programmed (read as ‘0’), or left unprogrammed (read as ‘1’), to select various device configurations. These bits are mapped starting at program memory location F80000h. A complete list is shown in Table 24-1. A detailed explanation of the various bit functions is provided in Register 24-1 through Register 24-4.

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.

### 24.1.1 CONSIDERATIONS FOR CONFIGURING PIC24FJ64GA004 FAMILY DEVICES

In PIC24FJ64GA004 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 two words at the top of the on-chip program memory space, known as the Flash Configuration Words. Their specific locations are shown in Table 24-1. These are packed representations of the actual device Configuration bits, whose actual locations are distributed among five 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.

**TABLE 24-1: FLASH CONFIGURATION WORD LOCATIONS FOR PIC24FJ64GA004 FAMILY DEVICES**

| Device      | Configuration Word Addresses |         |
|-------------|------------------------------|---------|
|             | 1                            | 2       |
| PIC24FJ16GA | 002BFEh                      | 002BFCh |
| PIC24FJ32GA | 0057FEh                      | 0057FCh |
| PIC24FJ48GA | 0083FEh                      | 0083FCh |
| PIC24FJ64GA | 00ABFEh                      | 00ABFCh |

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 Configuration bits are reloaded from the Flash Configuration Word on any device Reset.

The upper byte of both 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.

# PIC24FJ64GA004 FAMILY

## REGISTER 24-1: 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 | r-1 | R/PO-1 | R/PO-1 |
| r      | JTAGEN | GCP    | GWRP   | DEBUG  | r   | 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  |

|  |                                    |
|--|------------------------------------|
| <b>Legend:</b>                         | r = Reserved bit                   |
| R = Readable bit                       | PO = Program Once bit              |
| -n = Value when device is unprogrammed | U = Unimplemented bit, read as '0' |
|  | '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 = 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 **DEBUG:** Background Debugger Enable bit
  - 1 = Device resets into Operational mode
  - 0 = Device resets into Debug mode
- bit 10 **Reserved:** Always maintain as '1'
- bit 9-8 **ICS1:ICS0:** Emulator Pin Placement Select bits
  - 11 = Emulator EMUC1/EMUD1 pins are shared with PGC1/PGD1
  - 10 = Emulator EMUC2/EMUD2 pins are shared with PGC2/PGD2
  - 01 = Emulator EMUC3/EMUD3 pins are shared with PGC3/PGD3
  - 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 enabled
  - 0 = Windowed Watchdog Timer 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



# PIC24FJ64GA004 FAMILY

---

## REGISTER 24-1: CW1: FLASH CONFIGURATION WORD 1 (CONTINUED)

bit 3-0      **WDTPS3:WDTPS0:** 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

# PIC24FJ64GA004 FAMILY

## REGISTER 24-2: 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 | R/PO-1                 | R/PO-1                 | R/PO-1                  | R/PO-1                  | R/PO-1 | R/PO-1 | R/PO-1 |
| IESO   | WUTSEL1 <sup>(1)</sup> | WUTSEL0 <sup>(1)</sup> | SOSCSEL1 <sup>(1)</sup> | SOSCSEL0 <sup>(1)</sup> | 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   |

|  |                                    |
|--|------------------------------------|
| <b>Legend:</b>                         | r = Reserved bit                   |
| R = Readable bit                       | PO = Program Once bit              |
| -n = Value when device is unprogrammed | U = Unimplemented bit, read as '0' |
|  | '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) enabled  
 0 = IESO mode (Two-Speed Start-up) disabled
- bit 14-13 **WUTSEL1:WUTSEL0:** Voltage Regulator Standby Mode Wake-up Time Select bits<sup>(1)</sup>  
 11 = Default regulator start-up time used  
 01 = Fast regulator start-up time used  
 x0 = Reserved; do not use
- bit 12-11 **SOSCSEL1:SOSCSEL0:** Secondary Oscillator Power Mode Select bits<sup>(1)</sup>  
 11 = Default (High Drive Strength) mode  
 01 = Low-Power (Low Drive Strength) mode  
 x0 = Reserved; do not use
- bit 10-8 **FNOSC2:FNOSC0:** 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 **FCKSM1:FCKSM0:** 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 POSCMD1:POSCMD0 = 11 or 00:  
 1 = OSCO/CLKO/RA3 functions as CLKO (Fosc/2)  
 0 = OSCO/CLKO/RA3 functions as port I/O (RA3)  
If POSCMD1:POSCMD0 = 10 or 01:  
 OSCIOFCN has no effect on OSCO/CLKO/RA3.
- bit 4 **IOL1WAY:** IOLOCK One-Way Set Enable bit  
 1 = The OSCCON<IOLOCK> bit 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 OSCCON<IOLOCK> bit can be set and cleared as needed, provided the unlock sequence has been completed
- bit 3 **Unimplemented:** Read as '1'

# PIC24FJ64GA004 FAMILY

## REGISTER 24-2: CW2: FLASH CONFIGURATION WORD 2 (CONTINUED)

- bit 2      **I2C1SEL:** I2C1 Pin Select bit  
           1 = Use default SCL1/SDA1 pins  
           0 = Use alternate SCL1/SDA1 pins
- bit 1-0    **POSCMD1:POSCMD0:** Primary Oscillator Configuration bits  
           11 = Primary oscillator disabled  
           10 = HS Oscillator mode selected  
           01 = XT Oscillator mode selected  
           00 = EC Oscillator mode selected

**Note 1:** These bits are implemented only in devices with a major silicon revision level of B or later (DEVREV register value is 3042h or greater). Refer to **Section 28.0 “Packaging Information”** in the device data sheet for the location and interpretation of product date codes.

## REGISTER 24-3: DEVID: DEVICE ID REGISTER

|        |   |   |   |   |   |   |        |
|--------|---|---|---|---|---|---|--------|
| U      | U | U | U | U | U | U | U      |
| —      | — | — | — | — | — | — | —      |
| bit 23 |   |   |   |   |   |   | bit 16 |

|        |   |        |        |        |        |        |        |
|--------|---|--------|--------|--------|--------|--------|--------|
| U      | U | R      | R      | R      | R      | R      | R      |
| —      | — | FAMID7 | FAMID6 | FAMID5 | FAMID4 | FAMID3 | FAMID2 |
| bit 15 |   |        |        |        |        |        | bit 8  |

|        |        |      |      |      |      |      |       |
|--------|--------|------|------|------|------|------|-------|
| R      | R      | R    | R    | R    | R    | R    | R     |
| FAMID1 | FAMID0 | DEV5 | DEV4 | DEV3 | DEV2 | DEV1 | DEV0  |
| bit 7  |        |      |      |      |      |      | bit 0 |

**Legend:** R = Read-only bit      U = Unimplemented bit

- bit 23-14    **Unimplemented:** Read as ‘1’
- bit 13-6    **FAMID7:FAMID0:** Device Family Identifier bits  
           00010001 = PIC24FJ64GA004 family
- bit 5-0     **DEV5:DEV0:** Individual Device Identifier bits  
           000100 = PIC24FJ16GA002  
           000101 = PIC24FJ32GA002  
           000110 = PIC24FJ48GA002  
           000111 = PIC24FJ64GA002  
           001100 = PIC24FJ16GA004  
           001101 = PIC24FJ32GA004  
           001110 = PIC24FJ48GA004  
           001111 = PIC24FJ64GA004

# PIC24FJ64GA004 FAMILY

## REGISTER 24-4: DEVREV: DEVICE REVISION REGISTER

|        |   |   |   |   |   |   |        |
|--------|---|---|---|---|---|---|--------|
| U      | U | U | U | U | U | U | U      |
| —      | — | — | — | — | — | — | —      |
| bit 23 |   |   |   |   |   |   | bit 16 |

|        |   |   |   |   |   |   |        |
|--------|---|---|---|---|---|---|--------|
| U      | U | U | U | U | U | U | R      |
| —      | — | — | — | — | — | — | MAJRV2 |
| bit 15 |   |   |   |   |   |   | bit 8  |

|        |        |   |   |   |      |      |       |
|--------|--------|---|---|---|------|------|-------|
| R      | R      | U | U | U | R    | R    | R     |
| MAJRV1 | MAJRV0 | — | — | — | DOT2 | DOT1 | DOT0  |
| bit 7  |        |   |   |   |      |      | bit 0 |

|                                  |                       |
|----------------------------------|-----------------------|
| <b>Legend:</b> R = Read-only bit | U = Unimplemented bit |
|----------------------------------|-----------------------|

- bit 23-9      **Unimplemented:** Read as '0'
- bit 8-6      **MAJRV2:MAJRV0:** Major Revision Identifier bits
- bit 5-3      **Unimplemented:** Read as '0'
- bit 2-0      **DOT2:DOT0:** Minor Revision Identifier bits

# PIC24FJ64GA004 FAMILY

## 24.2 On-Chip Voltage Regulator

All of the PIC24FJ64GA004 family of 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 PIC24FJ64GA004 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 24-1). This helps to maintain the stability of the regulator. The recommended value for the filter capacitor is provided in **Section 27.1 “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 24-1 for possible configurations.

### 24.2.1 VOLTAGE REGULATOR TRACKING MODE AND LOW-VOLTAGE DETECTION

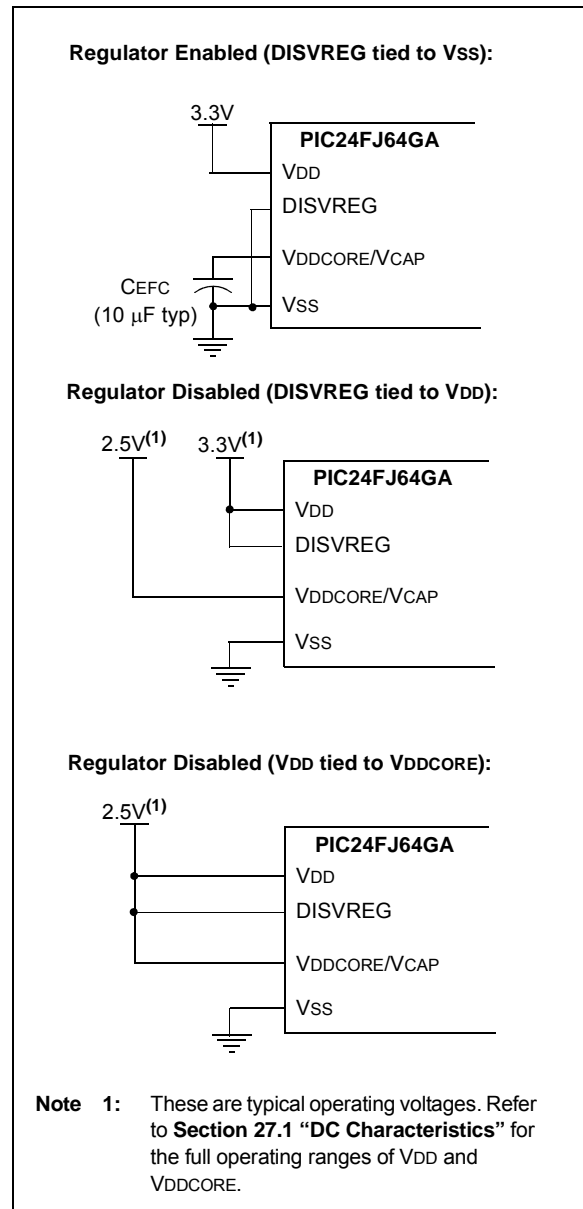
When it is enabled, the on-chip regulator provides a constant voltage of 2.5V nominal to the digital core logic.

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 24-1: CONNECTIONS FOR THE ON-CHIP REGULATOR



### 24.2.2 ON-CHIP REGULATOR AND POR

When the voltage regulator is enabled, it takes approximately 20 µs for it to generate output. During this time, designated as TSTARTUP, code execution is disabled. TSTARTUP is applied every time the device resumes operation after any power-down, including Sleep mode.

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.

# PIC24FJ64GA004 FAMILY

## 24.2.3 ON-CHIP REGULATOR AND BOR

When the on-chip regulator is enabled, PIC24FJ64GA004 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 levels are specified in **Section 27.1 “DC Characteristics”**.

## 24.2.4 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 27.0 “Electrical Characteristics”**.

## 24.2.5 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. This feature is controlled by the VREGS bit (RCON<8>). By default, this bit is cleared, which enables Standby mode.

For select PIC24FJ64GA004 family devices, the time required for regulator wake-up from Standby mode is controlled by the WUTSEL<1:0> Configuration bits (CW2<14:13>). The default wake-up time for all devices is 190  $\mu$ s. Where the WUTSEL Configuration bits are implemented, a fast wake-up option is also available. When WUTSEL<1:0> = 01, the regulator wake-up time is 25  $\mu$ s.

**Note:** This feature is implemented only on PIC24FJ64GA004 family devices with a major silicon revision level of B or later (DEVREV register value is 3042h or greater).

When the regulator’s Standby mode is turned off (VREGS = 1), Flash program memory stays powered in Sleep mode and the device can wake-up in less than 10  $\mu$ s. When VREGS is set, the power consumption while in Sleep mode will be approximately 40  $\mu$ A higher than power consumption when the regulator is allowed to enter Standby mode.

## 24.3 Watchdog Timer (WDT)

For PIC24FJ64GA004 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 WDTPS3:WDTPS0 Configuration bits (Flash Configuration Word 1<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 ranging 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.

# PIC24FJ64GA004 FAMILY

## 24.3.1 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 executed before that window causes a WDT Reset, similar to a WDT time-out.

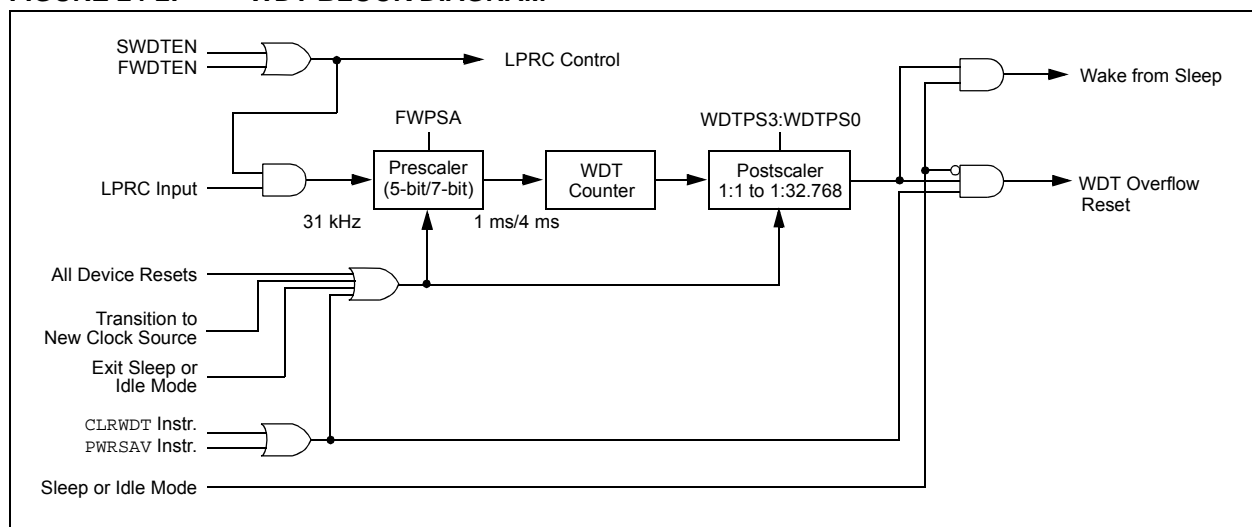
Windowed WDT mode is enabled by programming the `WINDIS` Configuration bit (`CW1<6>`) to '0'.

## 24.3.2 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 software WDT 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 24-2: WDT BLOCK DIAGRAM**



## 24.4 JTAG Interface

PIC24FJ64GA004 family devices implement a JTAG interface, which supports boundary scan device testing.

## 24.5 Program Verification and Code Protection

For all devices in the PIC24FJ64GA004 family of devices, the on-chip program memory space is treated as a single block. Code protection for this block is controlled by one Configuration bit, `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.

## 24.5.1 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.

# PIC24FJ64GA004 FAMILY

---

## 24.6 In-Circuit Serial Programming

PIC24FJ64GA004 family microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock (PGCx) and data (PGDx) and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

## 24.7 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 EMUCx (Emulation/Debug Clock) and EMUDx (Emulation/Debug Data) pins.

To use the in-circuit debugger function of the device, the design must implement ICSP connections to  $\overline{MCLR}$ ,  $V_{DD}$ ,  $V_{SS}$ , PGCx, PGDx and the EMUDx/EMUCx pin pair. 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.



# PIC24FJ64GA004 FAMILY

---

## 25.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C18 and MPLAB C30 C Compilers
  - MPLINK™ Object Linker/  
MPLIB™ Object Librarian
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PICSTART® Plus Development Programmer
  - MPLAB PM3 Device Programmer
  - PICKit™ 2 Development Programmer
- Low-Cost Demonstration and Development Boards and Evaluation Kits

## 25.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Visual device initializer for easy register initialization
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as HI-TECH Software C Compilers and IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PIC MCU emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (assembly or C)
  - Mixed assembly and C
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

# PIC24FJ64GA004 FAMILY

---

## 25.2 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for all PIC MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

## 25.3 MPLAB C18 and MPLAB C30 C Compilers

The MPLAB C18 and MPLAB C30 Code Development Systems are complete ANSI C compilers for Microchip's PIC18 and PIC24 families of microcontrollers and the dsPIC30 and dsPIC33 family of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use not found with other compilers.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

## 25.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

## 25.5 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 Assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire dsPIC30F instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

## 25.6 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C18 and MPLAB C30 C Compilers, and the MPASM and MPLAB ASM30 Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

## 25.7 MPLAB ICE 2000 High-Performance In-Circuit Emulator

The MPLAB ICE 2000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers. Software control of the MPLAB ICE 2000 In-Circuit Emulator is advanced by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The architecture of the MPLAB ICE 2000 In-Circuit Emulator allows expansion to support new PIC microcontrollers.

The MPLAB ICE 2000 In-Circuit Emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft® Windows® 32-bit operating system were chosen to best make these features available in a simple, unified application.

## 25.8 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 MPLAB REAL ICE probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with the popular MPLAB ICD 2 system (RJ11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

MPLAB REAL ICE is field upgradeable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added, such as software breakpoints and assembly code trace. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, real-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

## 25.9 MPLAB ICD 2 In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD 2, is a powerful, low-cost, run-time development tool, connecting to the host PC via an RS-232 or high-speed USB interface. This tool is based on the Flash PIC MCUs and can be used to develop for these and other PIC MCUs and dsPIC DSCs. The MPLAB ICD 2 utilizes the in-circuit debugging capability built into the Flash devices. This feature, along with Microchip's In-Circuit Serial Programming™ (ICSP™) protocol, offers cost-effective, in-circuit Flash debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by setting breakpoints, single stepping and watching variables, and CPU status and peripheral registers. Running at full speed enables testing hardware and applications in real time. MPLAB ICD 2 also serves as a development programmer for selected PIC devices.

## 25.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an SD/MMC card for file storage and secure data applications.

# PIC24FJ64GA004 FAMILY

---

## 25.11 PICSTART Plus Development Programmer

The PICSTART Plus Development Programmer is an easy-to-use, low-cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. The PICSTART Plus Development Programmer supports most PIC devices in DIP packages up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus Development Programmer is CE compliant.

## 25.12 PICkit 2 Development Programmer

The PICkit™ 2 Development Programmer is a low-cost programmer and selected Flash device debugger with an easy-to-use interface for programming many of Microchip's baseline, mid-range and PIC18F families of Flash memory microcontrollers. The PICkit 2 Starter Kit includes a prototyping development board, twelve sequential lessons, software and HI-TECH's PICC™ Lite C compiler, and is designed to help get up to speed quickly using PIC® microcontrollers. The kit provides everything needed to program, evaluate and develop applications using Microchip's powerful, mid-range Flash memory family of microcontrollers.

## 25.13 Demonstration, Development and Evaluation Boards

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Check the Microchip web page ([www.microchip.com](http://www.microchip.com)) for the complete list of demonstration, development and evaluation kits.

## 26.0 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<sup>®</sup> 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 26-1 shows the general symbols used in describing the instructions. The PIC24F instruction set summary in Table 26-2 lists all 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 WO 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.

# PIC24FJ64GA004 FAMILY

**TABLE 26-1: 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) $\in \{0...15\}$            |
| 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 $\in \{0000h...1FFFh\}$   |
| lit1            | 1-bit unsigned literal $\in \{0,1\}$  |
| lit4            | 4-bit unsigned literal $\in \{0...15\}$   |
| lit5            | 5-bit unsigned literal $\in \{0...31\}$   |
| lit8            | 8-bit unsigned literal $\in \{0...255\}$  |
| lit10           | 10-bit unsigned literal $\in \{0...255\}$ for Byte mode, $\{0:1023\}$ for Word mode         |
| lit14           | 14-bit unsigned literal $\in \{0...16384\}$   |
| lit16           | 16-bit unsigned literal $\in \{0...65535\}$   |
| lit23           | 23-bit unsigned literal $\in \{0...8388608\}$ ; LSB must be '0'                             |
| None            | Field does not require an entry, may be blank   |
| PC              | Program Counter   |
| Slit10          | 10-bit signed literal $\in \{-512...511\}$  |
| Slit16          | 16-bit signed literal $\in \{-32768...32767\}$  |
| Slit6           | 6-bit signed literal $\in \{-16...16\}$   |
| Wb              | Base W register $\in \{W0..W15\}$   |
| Wd              | Destination W register $\in \{Wd, [Wd], [Wd++] , [Wd--], [++Wd], [--Wd] \}$                 |
| Wdo             | Destination W register $\in \{Wnd, [Wnd], [Wnd++] , [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb] \}$ |
| Wm,Wn           | Dividend, Divisor working register pair (direct addressing)                                 |
| Wn              | One of 16 working registers $\in \{W0..W15\}$   |
| Wnd             | One of 16 destination working registers $\in \{W0..W15\}$                                   |
| Wns             | One of 16 source working registers $\in \{W0..W15\}$  |
| WREG            | W0 (working register used in file register instructions)                                    |
| Ws              | Source W register $\in \{Ws, [Ws], [Ws++] , [Ws--], [++Ws], [--Ws] \}$                      |
| Wso             | Source W register $\in \{Wns, [Wns], [Wns++] , [Wns--], [++Wns], [--Wns], [Wns+Wb] \}$      |

# PIC24FJ64GA004 FAMILY

**TABLE 26-2: 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<br>(2 or 3) | None                  |
|                   | BTSC Ws, #bit4     | Bit Test Ws, Skip if Clear               | 1          | 1<br>(2 or 3) | None                  |

# PIC24FJ64GA004 FAMILY

**TABLE 26-2: 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<br>(2 or 3) | None                  |
|                   | BTSS $Ws, \#bit4$    | Bit Test $Ws$ , Skip if Set                            | 1          | 1<br>(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 $Ws, Wd$         | $Wd = \bar{Ws}$  | 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 ( $Wb - Ws - C$ ) | 1          | 1             | C, DC, N, OV, Z       |
| CPSEQ             | CPSEQ $Wb, Wn$       | Compare $Wb$ with $Wn$ , Skip if =                     | 1          | 1<br>(2 or 3) | None                  |
| CPSGT             | CPSGT $Wb, Wn$       | Compare $Wb$ with $Wn$ , Skip if >                     | 1          | 1<br>(2 or 3) | None                  |
| CPSLT             | CPSLT $Wb, Wn$       | Compare $Wb$ with $Wn$ , Skip if <                     | 1          | 1<br>(2 or 3) | None                  |
| CPSNE             | CPSNE $Wb, Wn$       | Compare $Wb$ with $Wn$ , Skip if $\neq$                | 1          | 1<br>(2 or 3) | None                  |
| DAW               | DAW $Wn$             | $Wn =$ 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                     |



# PIC24FJ64GA004 FAMILY

**TABLE 26-2: 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                     |
| 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                 | $f = \bar{f} + 1$                            | 1                                      | 1           | C, DC, N, OV, Z       |
|                       | NEG f, WREG           | WREG = $\bar{f} + 1$                         | 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                  |

# PIC24FJ64GA004 FAMILY

**TABLE 26-2: INSTRUCTION SET OVERVIEW (CONTINUED)**

| Assembly Mnemonic | Assembly Syntax     | Description                             | # of Words | # of Cycles | Status Flags Affected |
|-------------------|---------------------|---|------------|-------------|-----------------------|
| PWRSVAV           | PWRSVAV #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                  |
| TBLRDH            | TBLRDH Ws, Wd       | Read Prog<23:16> to Wd<7:0>             | 1          | 2           | None                  |

# PIC24FJ64GA004 FAMILY

**TABLE 26-2: INSTRUCTION SET OVERVIEW (CONTINUED)**

| Assembly Mnemonic | Assembly Syntax          | Description                  | # of Words | # of Cycles | Status Flags Affected |
|-------------------|--------------------------|------------------------------|------------|-------------|-----------------------|
| TBLRDL            | TBLRDL <i>Ws, Wd</i>     | Read Prog<15:0> to Wd        | 1          | 2           | None                  |
| TBLWTH            | TBLWTH <i>Ws, Wd</i>     | Write Ws<7:0> to Prog<23:16> | 1          | 2           | None                  |
| TBLWTL            | TBLWTL <i>Ws, Wd</i>     | Write Ws to Prog<15:0>       | 1          | 2           | None                  |
| ULNK              | ULNK                     | Unlink Frame Pointer         | 1          | 1           | None                  |
| XOR               | XOR <i>f</i>             | $f = f .XOR. WREG$           | 1          | 1           | N, Z                  |
|                   | XOR <i>f, WREG</i>       | $WREG = f .XOR. WREG$        | 1          | 1           | N, Z                  |
|                   | XOR <i>#lit10, Wn</i>    | $Wd = lit10 .XOR. Wd$        | 1          | 1           | N, Z                  |
|                   | XOR <i>Wb, Ws, Wd</i>    | $Wd = Wb .XOR. Ws$           | 1          | 1           | N, Z                  |
|                   | XOR <i>Wb, #lit5, Wd</i> | $Wd = Wb .XOR. lit5$         | 1          | 1           | N, Z                  |
| ZE                | ZE <i>Ws, Wnd</i>        | $Wnd = Zero-Extend Ws$       | 1          | 1           | C, Z, N               |

# PIC24FJ64GA004 FAMILY

---

NOTES:

# PIC24FJ64GA004 FAMILY

## 27.0 ELECTRICAL CHARACTERISTICS

This section provides an overview of the PIC24FJ64GA004 family electrical characteristics. Additional information will be provided in future revisions of this document as it becomes available.

Absolute maximum ratings for the PIC24FJ64GA004 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<sup>(†)</sup>

|  |                       |
|--|-----------------------|
| 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 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 ( <b>Note 1</b> ).....                                | 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 ( <b>Note 1</b> ).....                        | 200 mA                |

**Note 1:** Maximum allowable current is a function of device maximum power dissipation (see Table 27-1).

†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.

# PIC24FJ64GA004 FAMILY

## 27.1 DC Characteristics

FIGURE 27-1: PIC24FJ64GA004 FAMILY VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)

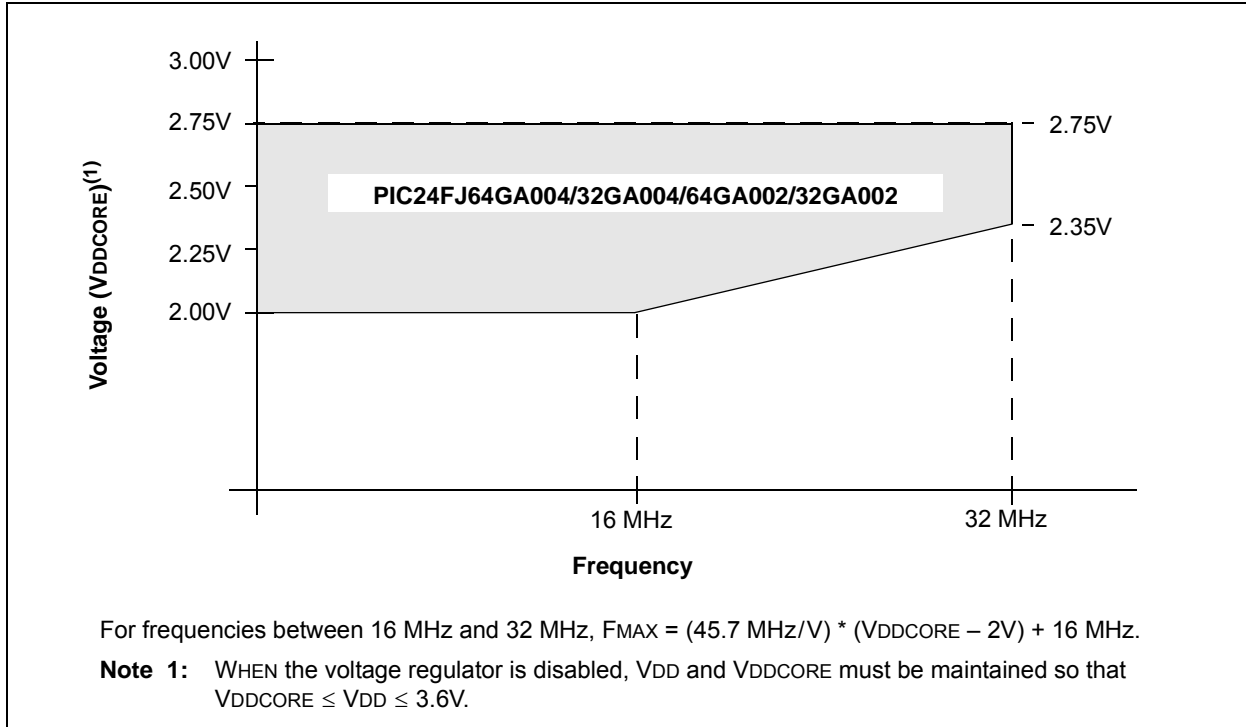
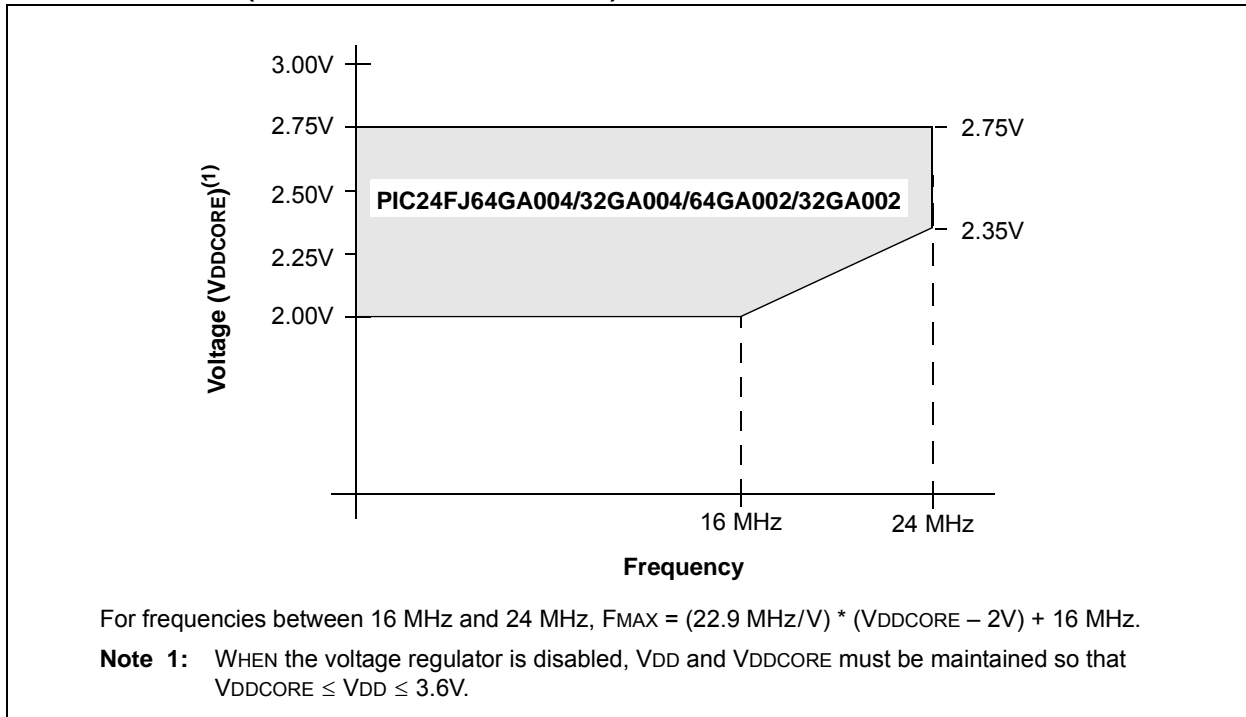


FIGURE 27-2: PIC24FJ64GA004 FAMILY VOLTAGE-FREQUENCY GRAPH (EXTENDED TEMPERATURE)



# PIC24FJ64GA004 FAMILY

**TABLE 27-1: THERMAL OPERATING CONDITIONS**

| Rating   | Symbol             | Min                                 | Typ | Max  | Unit |
|--|--------------------|-------------------------------------|-----|------|------|
| PIC24FJ64GA004 Family:   |                    |                                     |     |      |      |
| Operating Junction Temperature Range   | T <sub>J</sub>     | -40                                 | —   | +140 | °C   |
| Operating Ambient Temperature Range  | T <sub>A</sub>     | -40                                 | —   | +125 | °C   |
| Power Dissipation:<br>Internal Chip Power Dissipation:<br>$P_{INT} = V_{DD} \times (I_{DD} - \sum I_{OH})$<br>I/O Pin Power Dissipation:<br>$P_{I/O} = \sum ((V_{DD} - V_{OH}) \times I_{OH}) + \sum (V_{OL} \times I_{OL})$ | P <sub>D</sub>     | P <sub>INT</sub> + P <sub>I/O</sub> |     |      | W    |
| Maximum Allowed Power Dissipation  | P <sub>D</sub> MAX | $(T_J - T_A)/\theta_{JA}$           |     |      | W    |

**TABLE 27-2: THERMAL PACKAGING CHARACTERISTICS**

| Characteristic                              | Symbol        | Typ  | Max | Unit | Notes    |
|---|---------------|------|-----|------|----------|
| Package Thermal Resistance, 300 mil SOIC    | $\theta_{JA}$ | 49   | —   | °C/W | (Note 1) |
| Package Thermal Resistance, 6x6x0.9 mm QFN  | $\theta_{JA}$ | 33.7 | —   | °C/W | (Note 1) |
| Package Thermal Resistance, 8x8x1 mm QFN    | $\theta_{JA}$ | 28   | —   | °C/W | (Note 1) |
| Package Thermal Resistance, 10x10x1 mm TQFP | $\theta_{JA}$ | 39.3 | —   | °C/W | (Note 1) |

**Note 1:** Junction to ambient thermal resistance, Theta-JA ( $\theta_{JA}$ ) numbers are achieved by package simulations.

# PIC24FJ64GA004 FAMILY

**TABLE 27-3: 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<br>-40°C ≤ TA ≤ +125°C for Extended |                    |      |       |                                   |
| Param No.                | Symbol                | Characteristic   | Min   | Typ <sup>(1)</sup> | Max  | Units | Conditions                        |
| <b>Operating Voltage</b> |                       |  |   |                    |      |       |                                   |
| DC10                     | <b>Supply Voltage</b> |  |   |                    |      |       |                                   |
|                          | 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                   | <b>RAM Data Retention Voltage<sup>(2)</sup></b>                      | 1.5   | —                  | —    | V     |                                   |
| DC16                     | VPOR                  | <b>VDD Start Voltage</b><br>to ensure internal Power-on Reset signal | —   | VSS                | —    | V     |                                   |
| DC17                     | SVDD                  | <b>VDD Rise Rate</b><br>to ensure internal Power-on Reset signal     | 0.05  | —                  | —    | V/ms  | 0-3.3V in 0.1s<br>0-2.5V in 60 ms |

**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.



# PIC24FJ64GA004 FAMILY

**TABLE 27-4: 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<br>-40°C ≤ TA ≤ +125°C for Extended |
| Parameter No.  | Typical <sup>(1)</sup> | Max   | Units   | Conditions  |
| <b>Operating Current (IDD): PMD Bits are Set<sup>(2)</sup></b> |                        |       |   |   |
| DC20   | 0.650                  | 0.850 | mA  | -40°C   |
| DC20a  | 0.650                  | 0.850 | mA  | +25°C   |
| DC20b  | 0.650                  | 0.850 | mA  | +85°C   |
| DC20c  | 0.650                  | 0.850 | mA  | +125°C  |
| DC20d  | 1.2                    | 1.6   | mA  | -40°C   |
| DC20e  | 1.2                    | 1.6   | mA  | +25°C   |
| DC20f  | 1.2                    | 1.6   | mA  | +85°C   |
| DC20g  | 1.2                    | 1.6   | mA  | +125°C  |
| DC23   | 2.6                    | 3.4   | mA  | -40°C   |
| DC23a  | 2.6                    | 3.4   | mA  | +25°C   |
| DC23b  | 2.6                    | 3.4   | mA  | +85°C   |
| DC23c  | 2.6                    | 3.4   | mA  | +125°C  |
| DC23d  | 4.1                    | 5.4   | mA  | -40°C   |
| DC23e  | 4.1                    | 5.4   | mA  | +25°C   |
| DC23f  | 4.1                    | 5.4   | mA  | +85°C   |
| DC23g  | 4.1                    | 5.4   | mA  | +125°C  |
| DC24   | 13.5                   | 17.6  | mA  | -40°C   |
| DC24a  | 13.5                   | 17.6  | mA  | +25°C   |
| DC24b  | 13.5                   | 17.6  | mA  | +85°C   |
| DC24c  | 13.5                   | 17.6  | mA  | +125°C  |
| DC24d  | 15                     | 20    | mA  | -40°C   |
| DC24e  | 15                     | 20    | mA  | +25°C   |
| DC24f  | 15                     | 20    | mA  | +85°C   |
| DC24g  | 15                     | 20    | mA  | +125°C  |
| DC31   | 13                     | 17    | μA  | -40°C   |
| DC31a  | 13                     | 17    | μA  | +25°C   |
| DC31b  | 20                     | 26    | μA  | +85°C   |
| DC31c  | 40                     | 50    | μA  | +125°C  |
| DC31d  | 54                     | 70    | μA  | -40°C   |
| DC31e  | 54                     | 70    | μA  | +25°C   |
| DC31f  | 95                     | 124   | μA  | +85°C   |
| DC31g  | 120                    | 260   | 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. 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 disabled (DISVREG tied to VDD).

**4:** On-chip voltage regulator enabled (DISVREG tied to VSS). Low-Voltage Detect (LVD) and Brown-out Detect (BOD) are enabled.

# PIC24FJ64GA004 FAMILY

**TABLE 27-5: DC CHARACTERISTICS: IDLE CURRENT (I<sub>IDLE</sub>)**

| DC CHARACTERISTICS  |                        |      | Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated)   |            |
|---|------------------------|------|---|------------|
|   |                        |      | Operating temperature -40°C ≤ T <sub>A</sub> ≤ +85°C for Industrial<br>-40°C ≤ T <sub>A</sub> ≤ +125°C for Extended |            |
| Parameter No.   | Typical <sup>(1)</sup> | Max  | Units   | Conditions |
| <b>Idle Current (I<sub>IDLE</sub>): Core Off, Clock On Base Current, PMD Bits are Set<sup>(2)</sup></b> |                        |      |   |            |
| DC40  | 150                    | 200  | μA  | -40°C      |
| DC40a   | 150                    | 200  | μA  | +25°C      |
| DC40b   | 150                    | 200  | μA  | +85°C      |
| DC40c   | 165                    | 220  | μA  | +125°C     |
| DC40d   | 250                    | 325  | μA  | -40°C      |
| DC40e   | 250                    | 325  | μA  | +25°C      |
| DC40f   | 250                    | 325  | μA  | +85°C      |
| DC40g   | 275                    | 360  | μA  | +125°C     |
| DC43  | 0.55                   | 0.72 | mA  | -40°C      |
| DC43a   | 0.55                   | 0.72 | mA  | +25°C      |
| DC43b   | 0.55                   | 0.72 | mA  | +85°C      |
| DC43c   | 0.60                   | 0.8  | mA  | +125°C     |
| DC43d   | 0.82                   | 1.1  | mA  | -40°C      |
| DC43e   | 0.82                   | 1.1  | mA  | +25°C      |
| DC43f   | 0.82                   | 1.1  | mA  | +85°C      |
| DC43g   | 0.91                   | 1.2  | mA  | +125°C     |
| DC47  | 3                      | 4    | mA  | -40°C      |
| DC47a   | 3                      | 4    | mA  | +25°C      |
| DC47b   | 3                      | 4    | mA  | +85°C      |
| DC47c   | 3.3                    | 4.4  | mA  | +125°C     |
| DC47d   | 3.5                    | 4.6  | mA  | -40°C      |
| DC47e   | 3.5                    | 4.6  | mA  | +25°C      |
| DC47f   | 3.5                    | 4.6  | mA  | +85°C      |
| DC47g   | 3.9                    | 5.1  | mA  | +125°C     |
| DC50  | 0.85                   | 1.1  | mA  | -40°C      |
| DC50a   | 0.85                   | 1.1  | mA  | +25°C      |
| DC50b   | 0.85                   | 1.1  | mA  | +85°C      |
| DC50c   | 0.94                   | 1.2  | mA  | +125°C     |
| DC50d   | 1.2                    | 1.6  | mA  | -40°C      |
| DC50e   | 1.2                    | 1.6  | mA  | +25°C      |
| DC50f   | 1.2                    | 1.6  | mA  | +85°C      |
| DC50g   | 1.3                    | 1.8  | 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 test conditions for all I<sub>IDLE</sub> measurements are as follows: OSC1 driven with external square wave from rail to rail. All I/O pins are configured as inputs and pulled to V<sub>DD</sub>. MCLR = V<sub>DD</sub>; 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 disabled (DISVREG tied to V<sub>DD</sub>).

**4:** On-chip voltage regulator enabled (DISVREG tied to V<sub>SS</sub>). Low-Voltage Detect (LVD) and Brown-out Detect (BOD) are enabled.

# PIC24FJ64GA004 FAMILY

**TABLE 27-5: DC CHARACTERISTICS: IDLE CURRENT (I<sub>IDLE</sub>) (CONTINUED)**

| DC CHARACTERISTICS  |                        |     | Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated)                       |            |
|---|------------------------|-----|---|------------|
|   |                        |     | Operating temperature -40°C ≤ TA ≤ +85°C for Industrial<br>-40°C ≤ TA ≤ +125°C for Extended |            |
| Parameter No.   | Typical <sup>(1)</sup> | Max | Units   | Conditions |
| <b>Idle Current (I<sub>IDLE</sub>): Core Off, Clock On Base Current, PMD Bits are Set<sup>(2)</sup></b> |                        |     |   |            |
| DC51  | 4                      | 6   | μA  | -40°C      |
| DC51a   | 4                      | 6   | μA  | +25°C      |
| DC51b   | 8                      | 16  | μA  | +85°C      |
| DC51c   | 20                     | 50  | μA  | +125°C     |
| DC51d   | 42                     | 55  | μA  | -40°C      |
| DC51e   | 42                     | 55  | μA  | +25°C      |
| DC51f   | 70                     | 91  | μA  | +85°C      |
| DC51g   | 100                    | 180 | 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 test conditions for all I<sub>IDLE</sub> measurements are as follows:  $\overline{OSCI}$  driven with external square wave from rail to rail. All I/O pins are configured as inputs and pulled to V<sub>DD</sub>.  $\overline{MCLR} = V_{DD}$ ; 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 disabled (DISVREG tied to V<sub>DD</sub>).
- 4:** On-chip voltage regulator enabled (DISVREG tied to V<sub>SS</sub>). Low-Voltage Detect (LVD) and Brown-out Detect (BOD) are enabled.

# PIC24FJ64GA004 FAMILY

**TABLE 27-6: DC CHARACTERISTICS: POWER-DOWN 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 <sup>(1)</sup> | Max | Units   | Conditions |  |
| <b>Power-Down Current (IPD): PMD Bits are Set, VREGS Bit is '0'<sup>(2)</sup></b> |                        |     |   |            |  |
| DC60  | 0.1                    | 1   | μA  | -40°C      | Base Power-Down Current <sup>(5)</sup>                   |
| DC60a   | 0.15                   | 1   | μA  | +25°C      |  |
| DC60m   | 2.2                    | 7.4 | μA  | +60°C      |  |
| DC60b   | 3.7                    | 12  | μA  | +85°C      |  |
| DC60j   | 15                     | 50  | μA  | +125°C     |  |
| DC60c   | 0.2                    | 1   | μA  | -40°C      |  |
| DC60d   | 0.25                   | 1   | μA  | +25°C      |  |
| DC60n   | 2.6                    | 15  | μA  | +60°C      |  |
| DC60e   | 4.2                    | 25  | μA  | +85°C      |  |
| DC60k   | 16                     | 100 | μA  | +125°C     |  |
| DC60f   | 3.3                    | 9   | μA  | -40°C      |  |
| DC60g   | 3.5                    | 10  | μA  | +25°C      |  |
| DC60o   | 6.7                    | 22  | μA  | +60°C      |  |
| DC60h   | 9                      | 30  | μA  | +85°C      |  |
| DC60l   | 36                     | 120 | μA  | +125°C     |  |
| DC61  | 1.75                   | 3   | μA  | -40°C      | Watchdog Timer Current: ΔI <sub>WDT</sub> <sup>(5)</sup> |
| DC61a   | 1.75                   | 3   | μA  | +25°C      |  |
| DC61m   | 1.75                   | 3   | μA  | +60°C      |  |
| DC61b   | 1.75                   | 3   | μA  | +85°C      |  |
| DC61j   | 3.5                    | 6   | μA  | +125°C     |  |
| DC61c   | 2.4                    | 4   | μA  | -40°C      |  |
| DC61d   | 2.4                    | 4   | μA  | +25°C      |  |
| DC61n   | 2.4                    | 4   | μA  | +60°C      |  |
| DC61e   | 2.4                    | 4   | μA  | +85°C      |  |
| DC61k   | 4.8                    | 8   | μA  | +125°C     |  |
| DC61f   | 2.8                    | 5   | μA  | -40°C      |  |
| DC61g   | 2.8                    | 5   | μA  | +25°C      |  |
| DC61o   | 2.8                    | 5   | μA  | +60°C      |  |
| DC61h   | 2.8                    | 5   | μA  | +85°C      |  |
| DC61l   | 5.6                    | 10  | 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.
- Note 2:** Base IPD is measured with all peripherals and clocks shut down. All I/Os are configured as inputs and pulled high. WDT, etc., are all switched off.
- Note 3:** On-chip voltage regulator disabled (DISVREG tied to VDD).
- Note 4:** On-chip voltage regulator enabled (DISVREG tied to VSS). Low-Voltage Detect (LVD) and Brown-out Detect (BOD) are enabled.
- Note 5:** The Δ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

# PIC24FJ64GA004 FAMILY

**TABLE 27-6: DC CHARACTERISTICS: POWER-DOWN 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<br>-40°C ≤ TA ≤ +125°C for Extended |            |   |
| Parameter No.   | Typical <sup>(1)</sup> | Max | Units   | Conditions |   |
| <b>Power-Down Current (IPD): PMD Bits are Set, VREGS Bit is '0'<sup>(2)</sup></b> |                        |     |   |            |   |
| DC62  | 8                      | 16  | μA  | -40°C      | 2.0V <sup>(3)</sup><br><br>RTCC + Timer1 w/32 kHz Crystal:<br>ΔRTCC ΔI <sub>TI32</sub> <sup>(5)</sup>                                 |
| DC62a   | 12                     | 16  | μA  | +25°C      |   |
| DC62m   | 12                     | 16  | μA  | +60°C      |   |
| DC62b   | 12                     | 16  | μA  | +85°C      |   |
| DC62j   | 18                     | 23  | μA  | +125°C     |   |
| DC62c   | 9                      | 16  | μA  | -40°C      |   |
| DC62d   | 12                     | 16  | μA  | +25°C      |   |
| DC62n   | 12                     | 16  | μA  | +60°C      |   |
| DC62e   | 12.5                   | 16  | μA  | +85°C      |   |
| DC62k   | 20                     | 25  | μA  | +125°C     |   |
| DC62f   | 10.3                   | 18  | μA  | -40°C      |   |
| DC62g   | 13.4                   | 18  | μA  | +25°C      |   |
| DC62o   | 14.0                   | 18  | μA  | +60°C      |   |
| DC62h   | 14.2                   | 18  | μA  | +85°C      |   |
| DC62l   | 23                     | 28  | μA  | +125°C     |   |
| DC63  | 2                      | —   | μA  | -40°C      | 2.0V <sup>(3)</sup><br><br>RTCC + Timer1 w/Low-Power<br>32 kHz Crystal (SOCSEL<1:0> =<br>01): ΔRTCC ΔI <sub>TI32</sub> <sup>(5)</sup> |
| DC63a   | 2                      | —   | μA  | +25°C      |   |
| DC63b   | 6                      | —   | μA  | +85°C      |   |
| DC63c   | 2                      | —   | μA  | -40°C      |   |
| DC63d   | 2                      | —   | μA  | +25°C      |   |
| DC63e   | 7                      | —   | μA  | +85°C      |   |
| DC63f   | 2                      | —   | μA  | -40°C      |   |
| DC63g   | 3                      | —   | μA  | +25°C      |   |
| DC63h   | 7                      | —   | μA  | +85°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.
- Note 2:** Base IPD is measured with all peripherals and clocks shut down. All I/Os are configured as inputs and pulled high. WDT, etc., are all switched off.
- Note 3:** On-chip voltage regulator disabled (DISVREG tied to VDD).
- Note 4:** On-chip voltage regulator enabled (DISVREG tied to VSS). Low-Voltage Detect (LVD) and Brown-out Detect (BOD) are enabled.
- Note 5:** The Δ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

# PIC24FJ64GA004 FAMILY

**TABLE 27-7: DC CHARACTERISTICS: I/O PIN INPUT SPECIFICATIONS**

| DC CHARACTERISTICS |                 |   | Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for Industrial<br>-40°C ≤ TA ≤ +125°C for Extended |                    |                      |       |   |   |
|--------------------|-----------------|---|--|--------------------|----------------------|-------|---|---|
| Param No.          | Sym             | Characteristic  | Min  | Typ <sup>(1)</sup> | Max                  | Units | Conditions  |   |
| DI10               | V <sub>IL</sub> | <b>Input Low Voltage<sup>(4)</sup></b><br>I/O Pins                            | V <sub>SS</sub>  | —                  | 0.2 V <sub>DD</sub>  | V     | PMPTTL = 1  |   |
| DI11               |                 | PMP Pins  | V <sub>SS</sub>  | —                  | 0.15 V <sub>DD</sub> | V     |   |   |
| DI15               |                 | $\overline{\text{MCLR}}$  | V <sub>SS</sub>  | —                  | 0.2 V <sub>DD</sub>  | V     |   |   |
| DI16               |                 | OSCI (XT mode)  | V <sub>SS</sub>  | —                  | 0.2 V <sub>DD</sub>  | V     |   |   |
| DI17               |                 | OSCI (HS mode)  | V <sub>SS</sub>  | —                  | 0.2 V <sub>DD</sub>  | V     |   |   |
| DI18               |                 | I/O Pins with I <sup>2</sup> C™ Buffer  | V <sub>SS</sub>  | —                  | 0.3 V <sub>DD</sub>  | V     |   | SMBus disabled                            |
| DI19               |                 | I/O Pins with SMBus Buffer  | V <sub>SS</sub>  | —                  | 0.8                  | V     |   | SMBus enabled                             |
| DI20               | V <sub>IH</sub> | <b>Input High Voltage<sup>(4)</sup></b><br>I/O Pins:<br>with Analog Functions | 0.8 V <sub>DD</sub>  | —                  | V <sub>DD</sub>      | V     | PMPTTL = 1  |   |
|                    |                 | Digital Only  | 0.8 V <sub>DD</sub>  | —                  | 5.5                  | V     |   |   |
| DI21               |                 | PMP Pins:<br>with Analog Functions  | 0.25 V <sub>DD</sub> + 0.8   | —                  | V <sub>DD</sub>      | V     |   |   |
|                    |                 | Digital Only  | 0.25 V <sub>DD</sub> + 0.8   | —                  | 5.5                  | V     |   |   |
| DI25               |                 | $\overline{\text{MCLR}}$  | 0.8 V <sub>DD</sub>  | —                  | V <sub>DD</sub>      | V     |   |   |
| DI26               |                 | OSCI (XT mode)  | 0.7 V <sub>DD</sub>  | —                  | V <sub>DD</sub>      | V     |   |   |
| DI27               |                 | OSCI (HS mode)  | 0.7 V <sub>DD</sub>  | —                  | V <sub>DD</sub>      | V     |   |   |
| DI28               |                 | I/O Pins with I <sup>2</sup> C Buffer:<br>with Analog Functions               | 0.7 V <sub>DD</sub>  | —                  | V <sub>DD</sub>      | V     |   |   |
|                    |                 | Digital Only  | 0.7 V <sub>DD</sub>  | —                  | 5.5                  | V     |   |   |
| DI29               |                 | I/O Pins with SMBus Buffer:<br>with Analog Functions                          | 2.1  | —                  | V <sub>DD</sub>      | V     |   | 2.5V ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> |
|                    | Digital Only    | 2.1   | —  | 5.5                | v                    |       |   |   |
| DI30               | ICNPU           | <b>CNxx Pull-up Current</b>   | 50   | 250                | 400                  | μA    | V <sub>DD</sub> = 3.3V, V <sub>PIN</sub> = V <sub>SS</sub>                      |   |
| DI50               | I <sub>IL</sub> | <b>Input Leakage Current<sup>(2,3)</sup></b><br>I/O Ports                     | —  | —                  | ±1                   | μA    | V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> ,<br>Pin at high-impedance |   |
| DI51               |                 | Analog Input Pins   | —  | —                  | ±1                   | μA    | V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> ,<br>Pin at high-impedance |   |
| DI55               |                 | $\overline{\text{MCLR}}$  | —  | —                  | ±1                   | μA    | V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub>                            |   |
| DI56               |                 | OSCI  | —  | —                  | ±1                   | μA    | V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> ,<br>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 pin buffer types.

# PIC24FJ64GA004 FAMILY

**TABLE 27-8: 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<br>-40°C ≤ TA ≤ +125°C for Extended |                    |     |       |                                  |
| Param No.          | Sym | Characteristic                             | Min   | Typ <sup>(1)</sup> | Max | Units | Conditions                       |
| DO10               | VOL | <b>Output Low Voltage</b><br>All I/O pins  | —   | —                  | 0.4 | V     | IOL = 8.5 mA, VDD = 3.6V         |
|                    |     |  | —   | —                  | 0.4 | V     | IOL = 5.0 mA, VDD = 2.0V         |
| DO16               |     | All I/O pins                               | —   | —                  | 0.4 | V     | IOL = 8.0 mA, VDD = 3.6V, 125°C  |
|                    |     |  | —   | —                  | 0.4 | V     | IOL = 4.5 mA, VDD = 2.0V, 125°C  |
| DO20               | VOH | <b>Output High Voltage</b><br>All I/O pins | 3   | —                  | —   | V     | IOH = -3.0 mA, VDD = 3.6V        |
|                    |     |  | 1.65  | —                  | —   | V     | IOH = -1.0 mA, VDD = 2.0V        |
| DO26               |     | All I/O pins                               | 3   | —                  | —   | 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 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

**TABLE 27-9: 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<br>-40°C ≤ TA ≤ +125°C for Extended |                    |      |       |   |
| Param No.                   | Sym   | Characteristic                    | Min   | Typ <sup>(1)</sup> | Max  | Units | Conditions                                    |
| <b>Program Flash Memory</b> |       |                                   |   |                    |      |       |   |
| D130                        | EP    | Cell Endurance                    | 10000   | —                  | —    | E/W   | -40°C to +125°C                               |
| D131                        | VPR   | VDD for Read                      | VMIN  | —                  | 3.6  | V     | VMIN = Minimum operating voltage              |
| D132B                       | VPEW  | VDDCORE for Self-Timed Write      | 2.25  | —                  | 2.75 | V     |   |
| D133A                       | TIW   | Self-Timed Write Cycle Time       | —   | 3                  | —    | 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.

# PIC24FJ64GA004 FAMILY

**TABLE 27-10: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS**

| Operating Conditions: $-40^{\circ}\text{C} < T_A < +125^{\circ}\text{C}$ (unless otherwise stated) |        |                                 |     |      |     |               |  |
|--|--------|---------------------------------|-----|------|-----|---------------|--|
| Param No.  | Symbol | Characteristics                 | Min | Typ  | Max | Units         | Comments   |
|  | VRGOUT | Regulator Output Voltage        | —   | 2.5  | —   | V             |  |
|  | VBG    | Band Gap Reference Voltage      | —   | 1.23 | —   | V             |  |
|  | CEFC   | External Filter Capacitor Value | 4.7 | 10   | —   | $\mu\text{F}$ | Series resistance < 3 Ohm recommended; < 5 Ohm required. |
|  | TVREG  | Voltage Regulator Start-up Time | —   | 10   | —   | $\mu\text{s}$ | POR, BOR or when VREGS = 1                               |
|  |        |                                 | —   | 25   | —   | $\mu\text{s}$ | VREGS = 0, WUTSEL<1:0> = 01 <sup>(1)</sup>               |
|  |        |                                 | —   | 190  | —   | $\mu\text{s}$ | VREGS = 0, WUTSEL<1:0> = 11 <sup>(2)</sup>               |
|  | TPWRT  |                                 | —   | 64   | —   | ms            | DISVREG = VDD  |

**Note 1:** Available only in devices with a major silicon revision level of B or later (DEVREV register value is 3042h or greater).

**2:** WUTSEL Configuration bits setting is applicable only in devices with a major silicon revision level of B or later. This specification also applies to all devices prior to revision level B whenever VREGS = 0.



# PIC24FJ64GA004 FAMILY

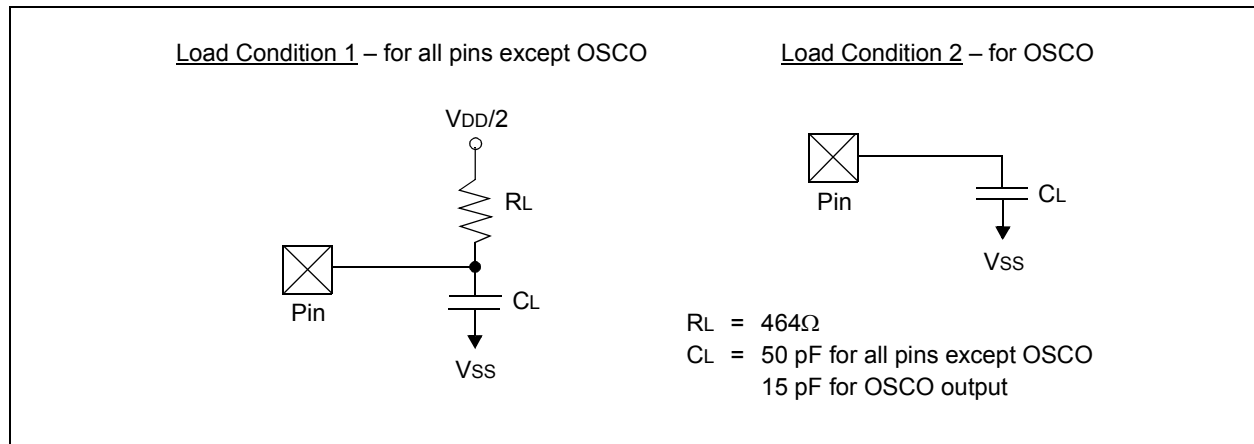
## 27.2 AC Characteristics and Timing Parameters

The information contained in this section defines the PIC24FJ64GA004 family AC characteristics and timing parameters.

**TABLE 27-11: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC**

|                           |   |
|---------------------------|---|
| <b>AC CHARACTERISTICS</b> | <b>Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated)</b>  |
|                           | Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial<br>$-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended |
|                           | Operating voltage $V_{DD}$ range as described in <b>Section 27.1 “DC Characteristics”</b> .   |

**FIGURE 27-3: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS**



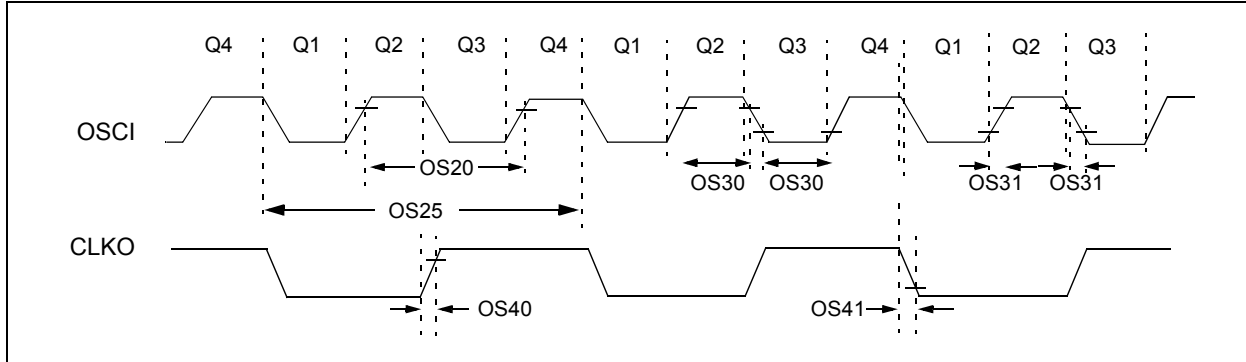
**TABLE 27-12: CAPACITIVE LOADING REQUIREMENTS ON OUTPUT PINS**

| Param No. | Symbol | Characteristic        | Min | Typ <sup>(1)</sup> | Max | Units | Conditions  |
|-----------|--------|-----------------------|-----|--------------------|-----|-------|---|
| DO50      | Cosc2  | OSCO/CLKO pin         | —   | —                  | 15  | pF    | In XT and HS modes when external clock is used to drive OSC1. |
| DO56      | Cio    | All I/O Pins and OSCO | —   | —                  | 50  | pF    | EC mode.  |
| DO58      | Cb     | SCLx, SDAx            | —   | —                  | 400 | pF    | In I <sup>2</sup> C™ 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.

# PIC24FJ64GA004 FAMILY

**FIGURE 27-4: EXTERNAL CLOCK TIMING**



**TABLE 27-13: EXTERNAL CLOCK TIMING REQUIREMENTS**

| AC CHARACTERISTICS |               |  | Standard Operating Conditions: 2.0 to 3.6V (unless otherwise stated)<br>Operating temperature -40°C ≤ Ta ≤ +85°C for Industrial<br>-40°C ≤ Ta ≤ +125°C for Extended |                    |     |       |   |
|--------------------|---------------|--|---|--------------------|-----|-------|---|
| Param No.          | Sym           | Characteristic   | Min   | Typ <sup>(1)</sup> | Max | Units | Conditions  |
| OS10               | Fosc          | External CLKI Frequency<br>(External clocks allowed only in EC mode) | DC  | —                  | 32  | MHz   | EC, -40°C ≤ Ta ≤ +85°C<br>ECPLL, -40°C ≤ Ta ≤ +85°C                                     |
|                    |               |  | 4   | —                  | 8   | MHz   |   |
|                    |               | Oscillator Frequency   | DC  | —                  | 24  | MHz   | EC, -40°C ≤ Ta ≤ +125°C<br>ECPLL, -40°C ≤ Ta ≤ +125°C                                   |
|                    |               |  | 4   | —                  | 6   | MHz   |   |
|                    | XT            | XTPLL, -40°C ≤ Ta ≤ +85°C  | 3   | —                  | 10  | MHz   | HS, -40°C ≤ Ta ≤ +85°C<br>SOSC<br>XTPLL, -40°C ≤ Ta ≤ +125°C<br>HS, -40°C ≤ Ta ≤ +125°C |
|                    |               |  | 3   | —                  | 8   | MHz   |   |
| 10                 |               |  | —   | 32                 | MHz |       |   |
| 31                 |               |  | —   | 33                 | kHz |       |   |
| 3                  | 6             | 24   | —   | —                  | —   | —     |   |
|                    |               |  |   |                    |     |       | 10  |
| OS20               | Tosc          | Tosc = 1/Fosc  | —   | —                  | —   | —     | See parameter OS10 for Fosc value   |
| OS25               | Tcy           | Instruction Cycle Time <sup>(2)</sup>                                | 62.5  | —                  | DC  | ns    |   |
| OS30               | TosL,<br>TosH | External Clock in (OSCI)<br>High or Low Time                         | 0.45 x Tosc   | —                  | —   | ns    | EC  |
| OS31               | TosR,<br>TosF | External Clock in (OSCI)<br>Rise or Fall Time                        | —   | —                  | 20  | ns    | EC  |
| OS40               | TckR          | CLKO Rise Time <sup>(3)</sup>  | —   | 6                  | 10  | ns    |   |
| OS41               | TckF          | CLKO Fall Time <sup>(3)</sup>  | —   | 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).

# PIC24FJ64GA004 FAMILY

**TABLE 27-14: PLL CLOCK TIMING SPECIFICATIONS (V<sub>DD</sub> = 2.0V TO 3.6V)**

| AC CHARACTERISTICS |       |                               | Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated)<br>Operating temperature -40°C ≤ T <sub>A</sub> ≤ +85°C for Industrial<br>-40°C ≤ T <sub>A</sub> ≤ +125°C for Extended |                    |     |       |  |
|--------------------|-------|-------------------------------|--|--------------------|-----|-------|--|
| Param No.          | Sym   | Characteristic <sup>(1)</sup> | Min  | Typ <sup>(2)</sup> | Max | Units | Conditions   |
| OS50               | FPLLI | PLL Input Frequency Range     | 3  | —                  | 8   | MHz   | ECPLL, HSPLL, XTPLL modes, -40°C ≤ T <sub>A</sub> ≤ +85°C  |
|                    |       |                               | 3  | —                  | 6   | MHz   | ECPLL, HSPLL, XTPLL modes, -40°C ≤ T <sub>A</sub> ≤ +125°C |
| OS51               | FSYS  | PLL Output Frequency Range    | 8  | —                  | 32  | MHz   | -40°C ≤ T <sub>A</sub> ≤ +85°C                             |
|                    |       |                               | 8  | —                  | 24  | MHz   | -40°C ≤ T <sub>A</sub> ≤ +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.

**Note 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 27-15: AC CHARACTERISTICS: INTERNAL RC ACCURACY**

| AC CHARACTERISTICS |  | Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated)<br>Operating temperature -40°C ≤ T <sub>A</sub> ≤ +85°C for Industrial<br>-40°C ≤ T <sub>A</sub> ≤ +125°C for Extended |     |     |       |                                 |
|--------------------|--|--|-----|-----|-------|---------------------------------|
| Param No.          | Characteristic                               | Min  | Typ | Max | Units | Conditions                      |
| F20                | Internal FRC Accuracy @ 8 MHz <sup>(1)</sup> |  |     |     |       |                                 |
|                    | FRC  | -2   | —   | 2   | %     | 25°C                            |
|                    |  | -5   | —   | 5   | %     | -40°C ≤ T <sub>A</sub> ≤ +125°C |
|                    |  |  |     |     |       | 3.0V ≤ V <sub>DD</sub> ≤ 3.6V   |

**Note 1:** Frequency calibrated at 25°C and 3.3V. OSCTUN bits can be used to compensate for temperature drift.

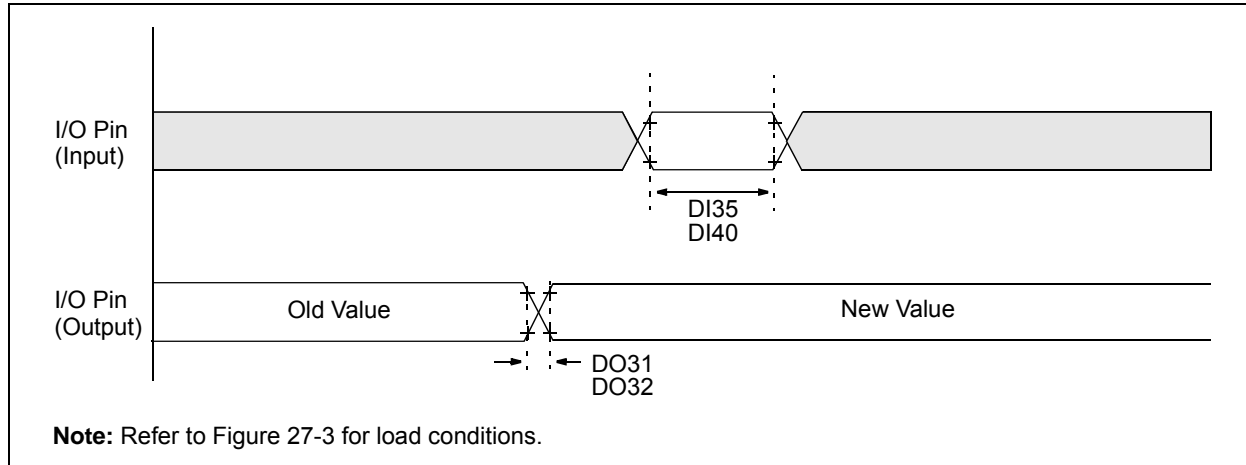
**TABLE 27-16: INTERNAL RC ACCURACY**

| AC CHARACTERISTICS |                              | Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated)<br>Operating temperature -40°C ≤ T <sub>A</sub> ≤ +85°C for Industrial<br>-40°C ≤ T <sub>A</sub> ≤ +125°C for Extended |     |     |       |                                |
|--------------------|------------------------------|--|-----|-----|-------|--------------------------------|
| Param No.          | Characteristic               | Min  | Typ | Max | Units | Conditions                     |
| F21                | LPRC @ 31 kHz <sup>(1)</sup> |  |     |     |       |                                |
|                    |                              | -15  | —   | 15  | %     | 25°C                           |
|                    |                              | -15  | —   | 15  | %     | -40°C ≤ T <sub>A</sub> ≤ +85°C |
|                    |                              | -20  | —   | 20  | %     | 125°C                          |
|                    |                              |  |     |     |       | 3.0V ≤ V <sub>DD</sub> ≤ 3.6V  |

**Note 1:** Change of LPRC frequency as V<sub>DD</sub> changes.

# PIC24FJ64GA004 FAMILY

**FIGURE 27-5: CLKO AND I/O TIMING CHARACTERISTICS**



**TABLE 27-17: CLKO AND I/O TIMING REQUIREMENTS**

| AC CHARACTERISTICS |      |                                    | Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated)   |                    |     |       |            |
|--------------------|------|------------------------------------|---|--------------------|-----|-------|------------|
|                    |      |                                    | Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial<br>$-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended |                    |     |       |            |
| Param No.          | Sym  | Characteristic                     | Min   | Typ <sup>(1)</sup> | 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.

# PIC24FJ64GA004 FAMILY

**TABLE 27-18: ADC MODULE SPECIFICATIONS**

| AC CHARACTERISTICS      |           |  | Standard Operating Conditions: 2.0V to 3.6V<br>(unless otherwise stated)<br>Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial<br>$-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended |         |                                  |          |  |
|-------------------------|-----------|--|---|---------|----------------------------------|----------|--|
| Param No.               | Symbol    | Characteristic                                 | Min.  | Typ     | Max.                             | Units    | Conditions                                     |
| <b>Device Supply</b>    |           |  |   |         |                                  |          |  |
| AD01                    | AVDD      | Module VDD Supply                              | Greater of<br>VDD – 0.3<br>or 2.0   | —       | Lesser of<br>VDD + 0.3<br>or 3.6 | V        |  |
| AD02                    | AVSS      | Module Vss Supply                              | Vss – 0.3   | —       | Vss + 0.3                        | V        |  |
| <b>Reference Inputs</b> |           |  |   |         |                                  |          |  |
| 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        |  |
| <b>Analog Input</b>     |           |  |   |         |                                  |          |  |
| 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        |  |
| AD17                    | RIN       | Recommended Impedance of Analog Voltage Source | —   | —       | 2.5K                             | $\Omega$ | 10-bit   |
| <b>ADC Accuracy</b>     |           |  |   |         |                                  |          |  |
| AD20b                   | Nr        | Resolution                                     | —   | 10      | —                                | bits     |  |
| AD21b                   | INL       | Integral Nonlinearity                          | —   | $\pm 1$ | $< \pm 2$                        | LSb      | VINL = AVSS = VREFL = 0V,<br>AVDD = VREFH = 3V |
| AD22b                   | DNL       | Differential Nonlinearity                      | —   | $\pm 1$ | $< \pm 1.25$                     | LSb      | VINL = AVSS = VREFL = 0V,<br>AVDD = VREFH = 3V |
| AD23b                   | GERR      | Gain Error                                     | —   | $\pm 1$ | $\pm 3$                          | LSb      | VINL = AVSS = VREFL = 0V,<br>AVDD = VREFH = 3V |
| AD24b                   | EOFF      | Offset Error                                   | —   | $\pm 1$ | $\pm 2$                          | LSb      | VINL = AVSS = VREFL = 0V,<br>AVDD = VREFH = 3V |
| AD25b                   | —         | Monotonicity <sup>(1)</sup>                    | —   | —       | —                                | —        | 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- used as the ADC voltage reference.

# PIC24FJ64GA004 FAMILY

**TABLE 27-19: ADC CONVERSION TIMING REQUIREMENTS<sup>(1)</sup>**

| AC CHARACTERISTICS      |                   |   | Standard Operating Conditions: 2.0V to 3.6V<br>(unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for Industrial<br>-40°C ≤ TA ≤ +125°C for Extended |     |      |       |   |
|-------------------------|-------------------|---|---|-----|------|-------|---|
| Param No.               | Symbol            | Characteristic                                    | Min.  | Typ | Max. | Units | Conditions  |
| <b>Clock Parameters</b> |                   |   |   |     |      |       |   |
| AD50                    | TAD               | ADC Clock Period                                  | 75  | —   | —    | ns    | T <sub>cy</sub> = 75 ns, AD1CON3 in default state |
| AD51                    | t <sub>RC</sub>   | ADC Internal RC Oscillator Period                 | —   | 250 | —    | ns    |   |
| <b>Conversion Rate</b>  |                   |   |   |     |      |       |   |
| AD55                    | t <sub>CONV</sub> | Conversion Time                                   | —   | 12  | —    | TAD   |   |
| AD56                    | FCNV              | Throughput Rate                                   | —   | —   | 500  | ksps  | AV <sub>DD</sub> ≥ 2.7V                           |
| AD57                    | t <sub>SAMP</sub> | Sample Time                                       | —   | 1   | —    | TAD   |   |
| <b>Clock Parameters</b> |                   |   |   |     |      |       |   |
| AD61                    | t <sub>PSS</sub>  | Sample Start Delay from setting Sample bit (SAMP) | 2   | —   | 3    | TAD   |   |

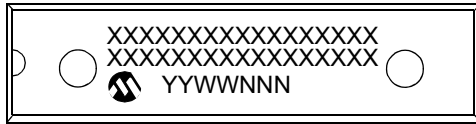
**Note 1:** Because the sample caps will eventually lose charge, clock rates below 10 kHz can affect linearity performance, especially at elevated temperatures.

# PIC24FJ64GA004 FAMILY

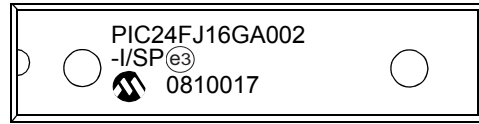
## 28.0 PACKAGING INFORMATION

### 28.1 Package Marking Information

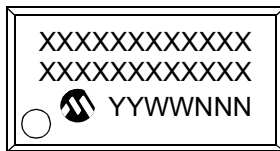
28-Lead SPDIP



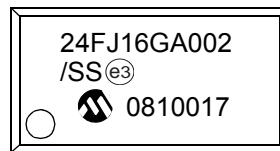
Example



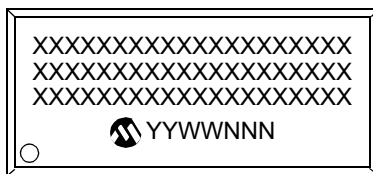
28-Lead SSOP



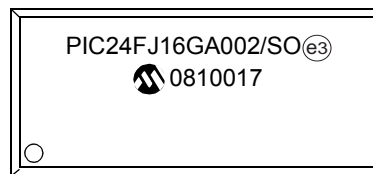
Example



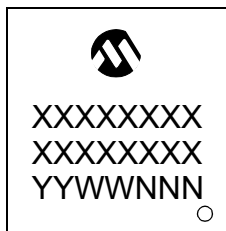
28-Lead SOIC (.300")



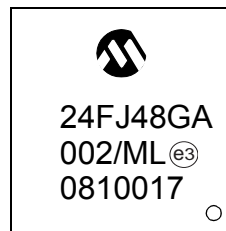
Example



28-Lead QFN



Example

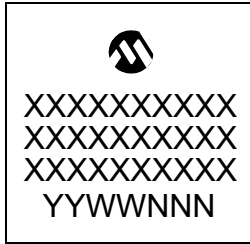


|                |   |   |
|----------------|---|---|
| <b>Legend:</b> | XX...X  | Customer-specific information   |
|                | Y   | Year code (last digit of calendar year)   |
|                | YY  | Year code (last 2 digits of calendar year)  |
|                | WW  | Week code (week of January 1 is week '01')  |
|                | NNN   | Alphanumeric traceability code  |
|                | *   | Pb-free JEDEC designator for Matte Tin (Sn)<br>This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package. |
| <b>Note:</b>   | 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. |   |

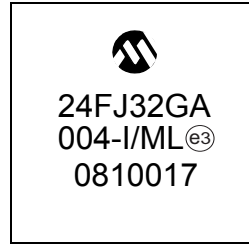
# PIC24FJ64GA004 FAMILY

---

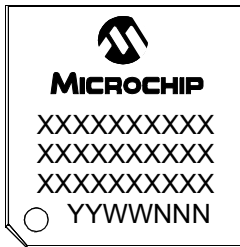
44-Lead QFN



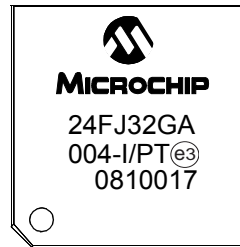
Example



44-Lead TQFP



Example





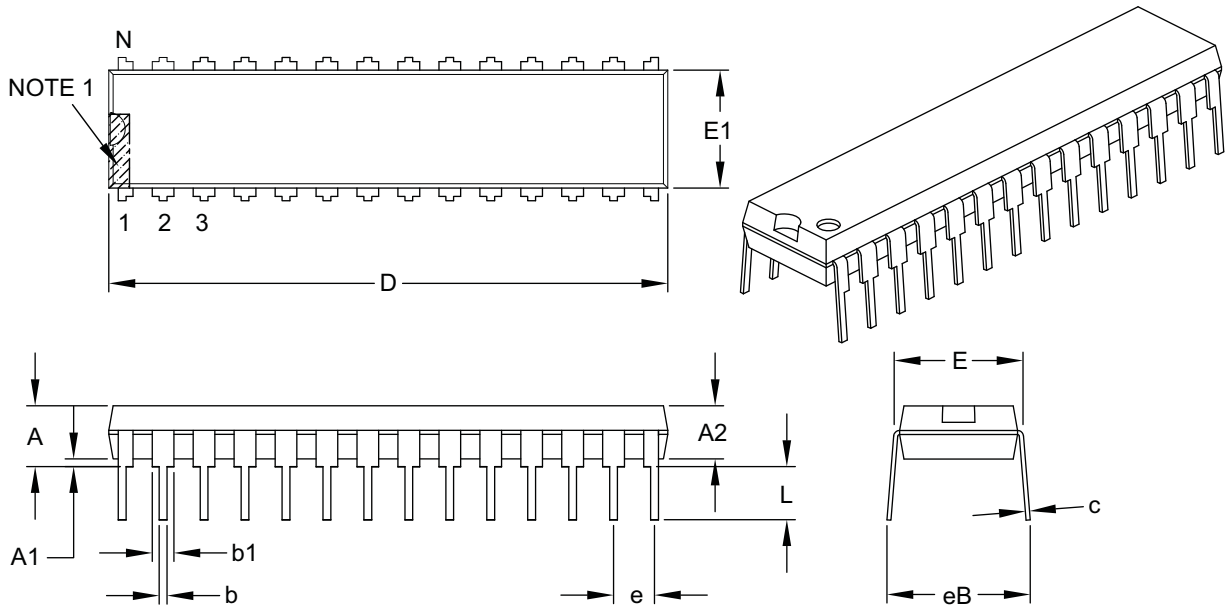
# PIC24FJ64GA004 FAMILY

## 28.2 Package Details

The following sections give the technical details of the packages.

### 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>



| Dimension Limits           | Units | INCHES   |       |       |
|----------------------------|-------|----------|-------|-------|
|                            |       | 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.

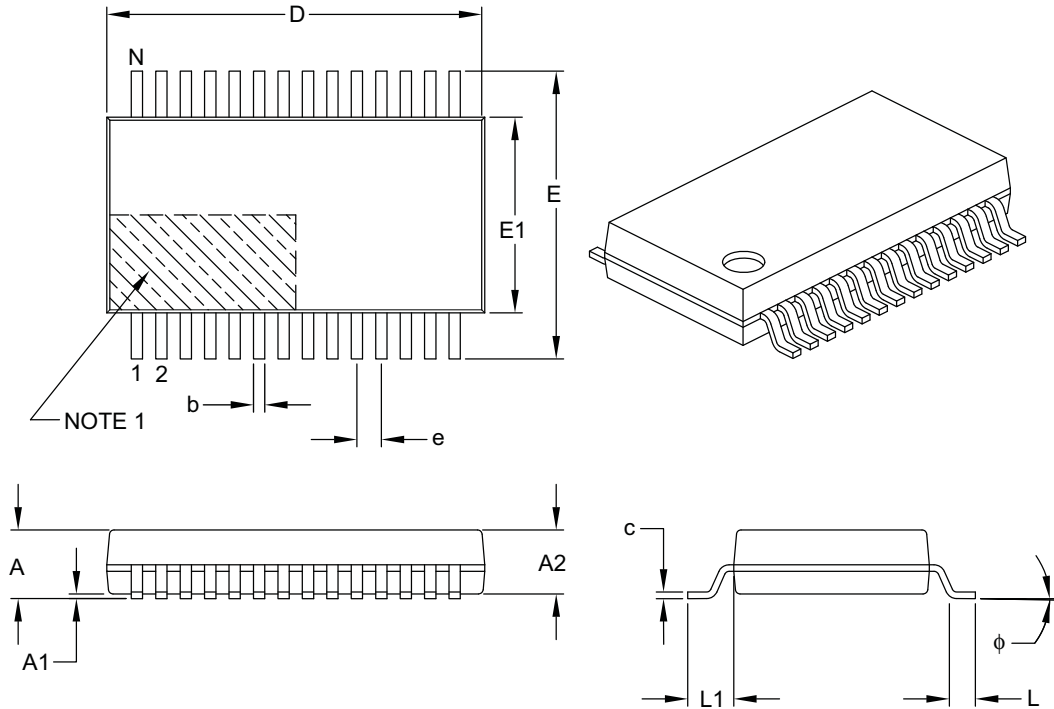
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-070B

# PIC24FJ64GA004 FAMILY

## 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               | $\phi$ |       | 0°          | 4°    | 8°    |
| Lead Width               | b      |       | 0.22        | –     | 0.38  |

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

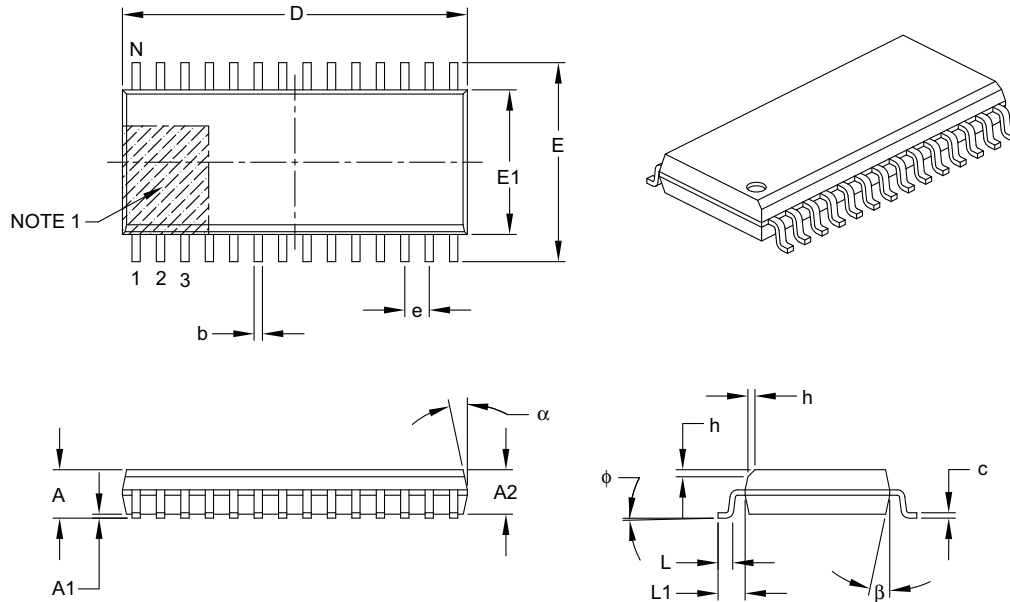
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B

# PIC24FJ64GA004 FAMILY

## 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>



| Dimension Limits         | Units    | MILLIMETERS |     |      |
|--------------------------|----------|-------------|-----|------|
|                          |          | 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           | $\phi$   | 0°          | –   | 8°   |
| Lead Thickness           | c        | 0.18        | –   | 0.33 |
| Lead Width               | b        | 0.31        | –   | 0.51 |
| Mold Draft Angle Top     | $\alpha$ | 5°          | –   | 15°  |
| Mold Draft Angle Bottom  | $\beta$  | 5°          | –   | 15°  |

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

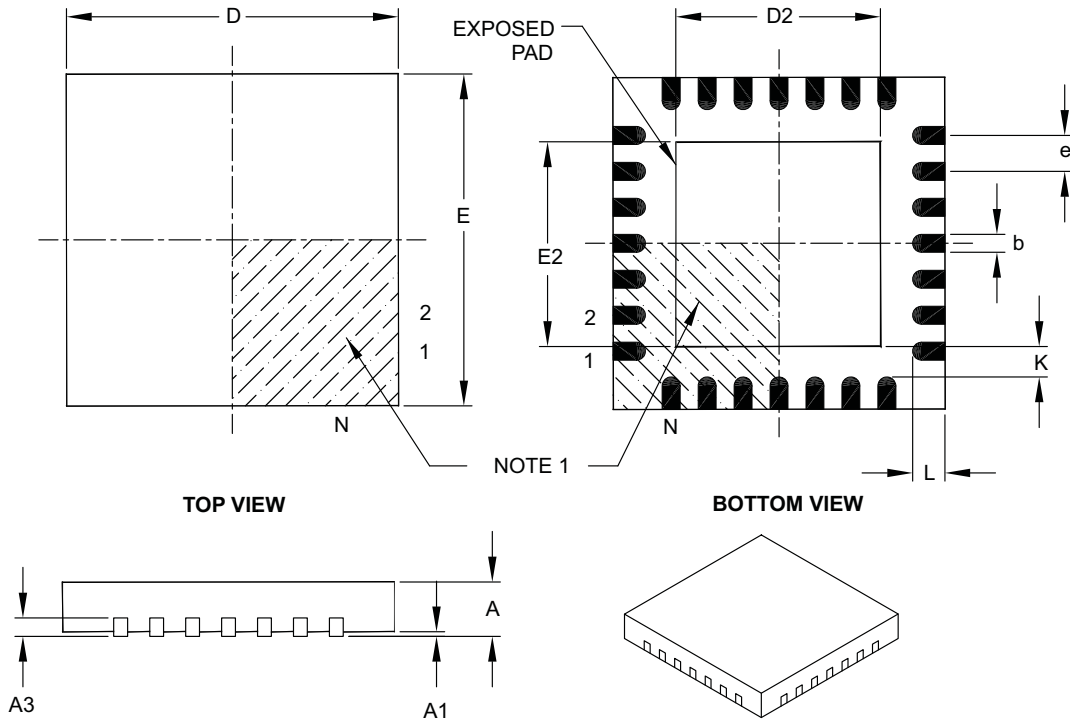
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-052B

# PIC24FJ64GA004 FAMILY

## 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/packageing>



| Dimension Limits       | Units | MILLIMETERS |      |      |
|------------------------|-------|-------------|------|------|
|                        |       | 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:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

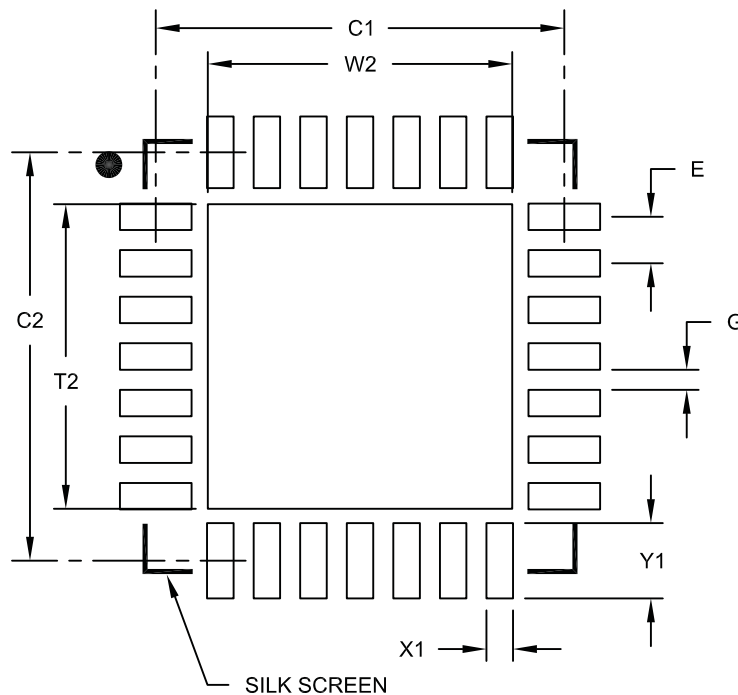
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-105B

# PIC24FJ64GA004 FAMILY

## 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

| Dimension Limits           | Units | MILLIMETERS |      |      |
|----------------------------|-------|-------------|------|------|
|                            |       | 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

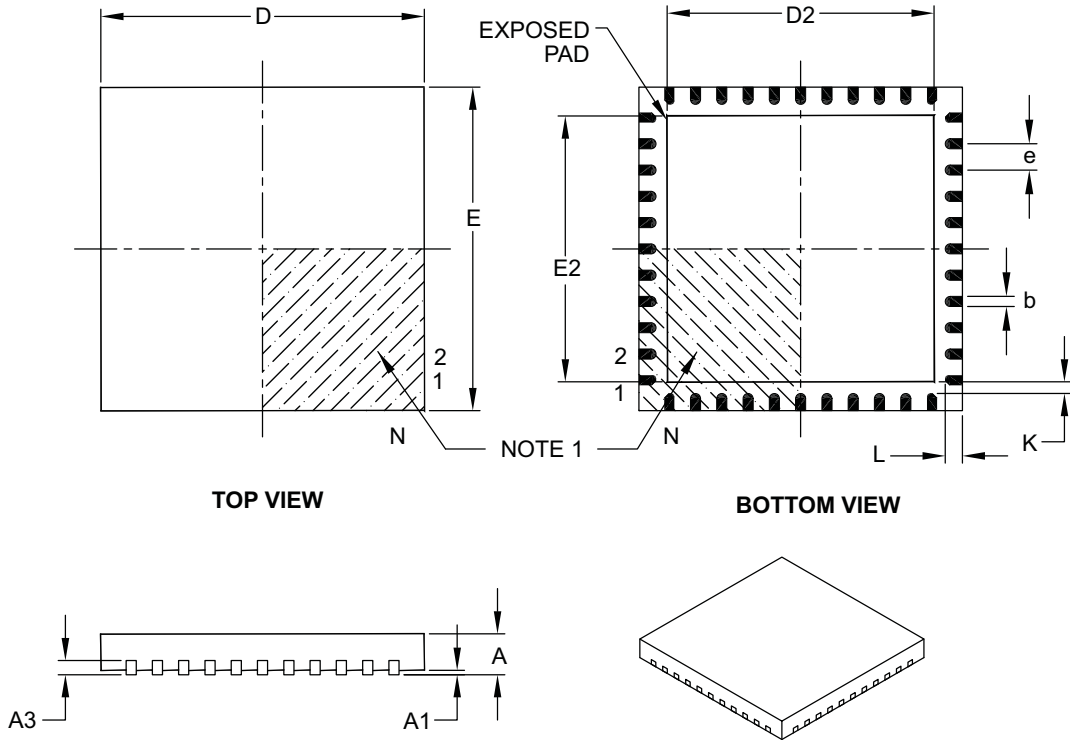
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2105A

# PIC24FJ64GA004 FAMILY

## 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>



| Dimension Limits       | Units | MILLIMETERS |      |      |
|------------------------|-------|-------------|------|------|
|                        |       | 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:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

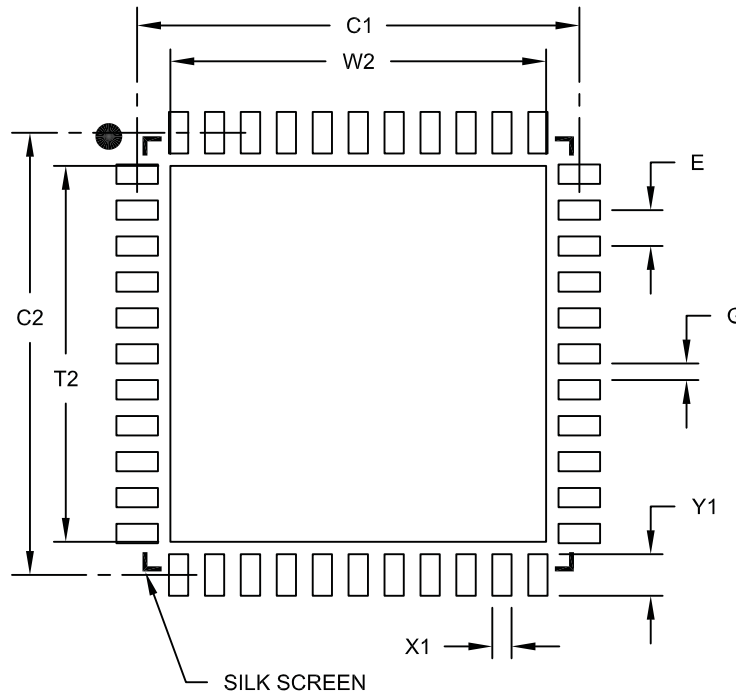
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-103B

# PIC24FJ64GA004 FAMILY

## 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

| Dimension Limits           | Units | MILLIMETERS |      |      |
|----------------------------|-------|-------------|------|------|
|                            |       | 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

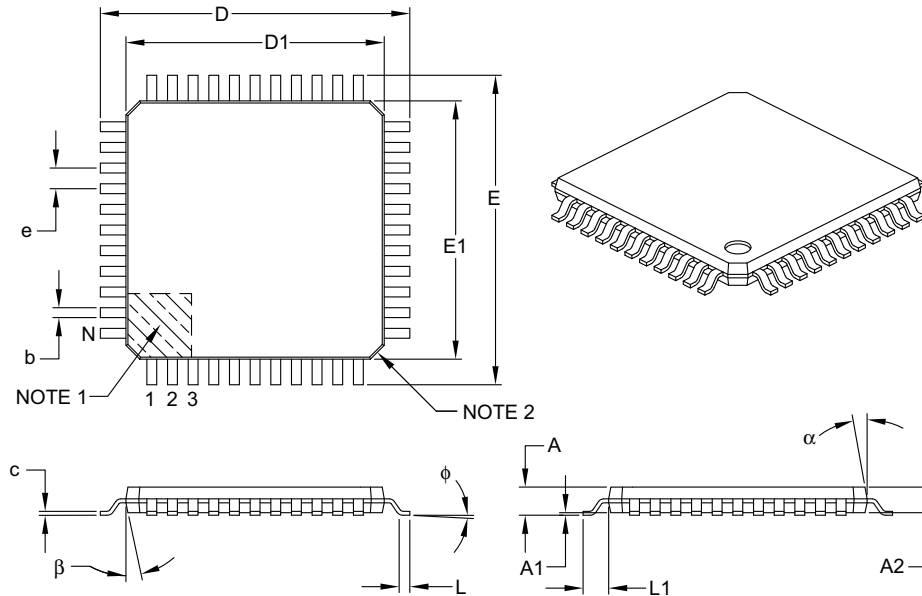
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2103A

# PIC24FJ64GA004 FAMILY

## 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>



| Dimension Limits         | Units    | MILLIMETERS |      |      |
|--------------------------|----------|-------------|------|------|
|                          |          | 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               | $\phi$   | 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     | $\alpha$ | 11°         | 12°  | 13°  |
| Mold Draft Angle Bottom  | $\beta$  | 11°         | 12°  | 13°  |

### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Chamfers at corners are optional; size may vary.
- Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
- 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.

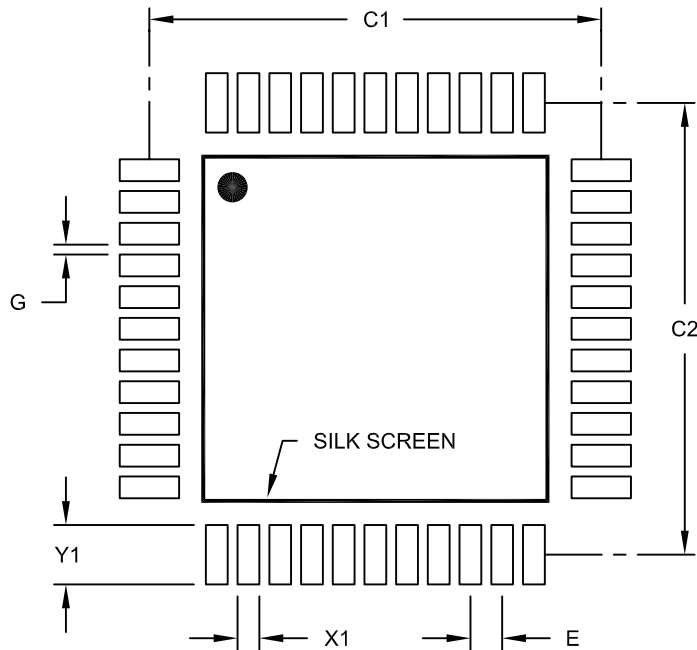
Microchip Technology Drawing C04-076B



# PIC24FJ64GA004 FAMILY

## 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

| Dimension Limits         | Units | MILLIMETERS |       |      |
|--------------------------|-------|-------------|-------|------|
|                          |       | 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

# PIC24FJ64GA004 FAMILY

---

NOTES:

## APPENDIX A: REVISION HISTORY

### Revision A (March 2007)

Original data sheet for the PIC24FJ64GA004 family of devices.

### Revision B (March 2007)

Changes to Table 26-8; packaging diagrams updated.

### Revision C (January 2008)

- Update of electrical specifications to include DC characteristics for Extended Temperature devices.
- Update for A/D converter chapter to include information on internal band gap voltage reference.
- Added "**Appendix B: "Additional Guidance for PIC24FJ64GA004 Family Applications"**".
- General revisions to incorporate corrections included in document errata to date (DS80333).

### Revision D (January 2010)

- Update of electrical specifications to include 60°C specifications for power-down current to DC characteristics.
- Removes references to JTAG programming throughout the document.
- Other minor typographic corrections throughout.

# PIC24FJ64GA004 FAMILY

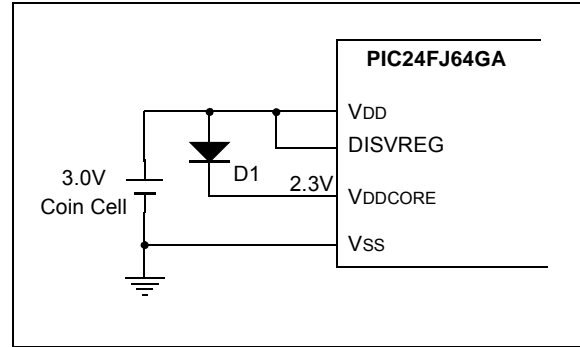
## APPENDIX B: ADDITIONAL GUIDANCE FOR PIC24FJ64GA004 FAMILY APPLICATIONS

### B.1 Additional Methods for Power Reduction

Devices in the PIC24FJ64GA004 family include a number of core features to significantly reduce the application's power requirements. For truly power-sensitive applications, it is possible to further reduce the application's power demands by taking advantage of the device's regulator architecture. These methods help decrease power in two ways: by disabling the internal voltage regulator to eliminate its power consumption, and by reducing the voltage on VDDCORE to lower the device's dynamic current requirements. Using these methods, it is possible to reduce Sleep currents (IPD) from 3.5  $\mu$ A to 250 nA (typical values, refer to specifications DC60d and DC60g in Table 27-6). For dynamic power consumption, the reduction in VDDCORE from 2.5V, provided by the regulator, to 2.0V can provide a power reduction of about 30%.

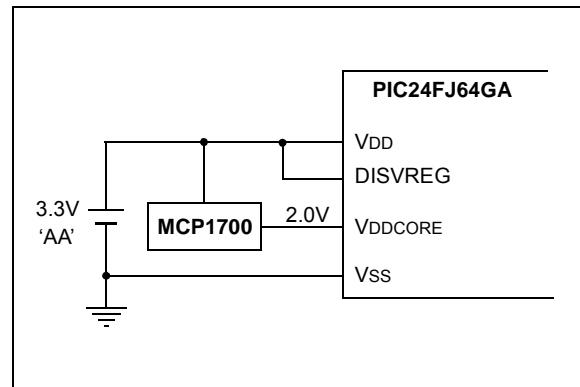
When using a regulated power source or a battery with a constant output voltage, it is possible to decrease power consumption by disabling the regulator. In this case (Figure B-1), a simple diode can be used to reduce the voltage from 3V or greater to the 2V-2.5V required for VDDCORE. This method is only advised on power supplies, such as Lithium Coin cells, which maintain a constant voltage over the life of the battery.

FIGURE B-1: POWER REDUCTION EXAMPLE FOR CONSTANT VOLTAGE SUPPLIES



A similar method can be used for non-regulated sources (Figure B-2). In this case, it can be beneficial to use a low quiescent current external voltage regulator. Devices such as the MCP1700 consume only 1  $\mu$ A to regulate to 2V or 2.5V, which is lower than the current required to power the internal voltage regulator.

FIGURE B-2: POWER REDUCTION EXAMPLE FOR NON-REGULATED SUPPLIES



# PIC24FJ64GA004 FAMILY

## INDEX

### A

|   |     |
|---|-----|
| A/D Converter                                     |     |
| Analog Input Model .....                          | 198 |
| Transfer Function .....                           | 199 |
| Additional Guidance for Family Applications ..... | 260 |
| Assembler   |     |
| MPASM Assembler .....                             | 218 |

### B

|   |         |
|---|---------|
| Block Diagrams                                    |         |
| 10-Bit High-Speed A/D Converter .....             | 192     |
| Accessing Program Memory with                     |         |
| Table Instructions .....                          | 47      |
| Addressable PMP Example .....                     | 174     |
| CALL Stack Frame .....                            | 45      |
| Comparator Operating Modes .....                  | 201     |
| Comparator Voltage Reference .....                | 205     |
| CPU Programmer's Model .....                      | 27      |
| CRC Reconfigured for Polynomial .....             | 188     |
| CRC Shifter Details .....                         | 187     |
| Data Access From Program Space                    |         |
| Address Generation .....                          | 46      |
| I <sup>2</sup> C Module .....                     | 152     |
| Input Capture .....                               | 133     |
| Legacy PMP Example .....                          | 174     |
| On-Chip Regulator Connections .....               | 212     |
| Output Compare .....                              | 138     |
| PIC24F CPU Core .....                             | 26      |
| PIC24FJ64GA004 Family (General) .....             | 12      |
| PMP   |         |
| Master Port Examples .....                        | 174–176 |
| PMP Module Overview .....                         | 167     |
| PSV Operation .....                               | 48      |
| Reset System .....                                | 55      |
| RTCC .....  | 177     |
| Shared I/O Port Structure .....                   | 105     |
| Simplified UART .....                             | 159     |
| SPI Master/Frame Master Connection .....          | 149     |
| SPI Master/Frame Slave Connection .....           | 149     |
| SPI Master/Slave Connection (Enhanced             |         |
| Buffer Mode) .....                                | 148     |
| SPI Master/Slave Connection (Standard Mode) ..... | 148     |
| SPI Slave/Frame Master Connection .....           | 149     |
| SPI Slave/Frame Slave Connection .....            | 149     |
| SPIx Module (Enhanced Mode) .....                 | 143     |
| SPIx Module (Standard Mode) .....                 | 142     |
| System Clock Diagram .....                        | 95      |
| Timer1 .....                                      | 125     |
| Timer2 and Timer4 (16-Bit Modes) .....            | 129     |
| Timer2/3 and Timer4/5 (32-Bit Mode) .....         | 128     |
| Timer3 and Timer5 (16-Bit Modes) .....            | 129     |
| Watchdog Timer (WDT) .....                        | 214     |

### C

|  |     |
|--|-----|
| C Compilers                                |     |
| MPLAB C18 .....                            | 218 |
| MPLAB C30 .....                            | 218 |
| Code Examples                              |     |
| Basic Clock Switching Example .....        | 101 |
| Configuring UART 1 Input and Output        |     |
| Functions (PPS) .....                      | 110 |
| Erasing a Program Memory Block .....       | 52  |
| I/O Port Read/Write .....                  | 106 |
| Initiating a Programming Sequence .....    | 53  |
| Loading the Write Buffers .....            | 53  |
| Single-Word Flash Programming .....        | 54  |
| Code Protection .....                      | 214 |
| Configuration Bits .....                   | 207 |
| Core Features .....                        | 9   |
| CPU  |     |
| ALU .....                                  | 29  |
| Control Registers .....                    | 28  |
| Core Registers .....                       | 27  |
| Programmer's Model .....                   | 25  |
| CRC  |     |
| CRCXOR Register .....                      | 190 |
| Operation in Power Save Modes .....        | 188 |
| User Interface .....                       | 188 |
| Customer Change Notification Service ..... | 265 |
| Customer Notification Service .....        | 265 |
| Customer Support .....                     | 265 |

### D

|                                 |     |
|---------------------------------|-----|
| Data Memory                     |     |
| Address Space .....             | 33  |
| Memory Map .....                | 33  |
| Near Data Space .....           | 34  |
| Organization .....              | 34  |
| SFR Space .....                 | 34  |
| Software Stack .....            | 45  |
| Development Support .....       | 217 |
| Device Features (Summary) ..... | 11  |
| DISVREG Pin .....               | 212 |
| Doze Mode .....                 | 104 |

### E

|  |         |
|--|---------|
| Electrical Characteristics             |         |
| A/D Specifications .....               | 244     |
| Absolute Maximum Ratings .....         | 229     |
| Current Specifications .....           | 233–237 |
| I/O Pin Specifications .....           | 238–239 |
| Internal Clock Specifications .....    | 242     |
| Load Conditions and Requirements for   |         |
| AC Characteristics .....               | 240     |
| Program Memory Specifications .....    | 239     |
| Thermal Operating Conditions .....     | 231     |
| V/F Graphs .....                       | 230     |
| Voltage Ratings .....                  | 232     |
| Voltage Regulator Specifications ..... | 239     |

# PIC24FJ64GA004 FAMILY

|   |             |  |
|---|-------------|--|
| Equations   |             |  |
| A/D Clock Conversion Period .....                           | 198         |  |
| Baud Rate Reload Calculation .....                          | 153         |  |
| Calculating the PWM Period .....                            | 136         |  |
| Calculation for Maximum PWM Resolution .....                | 136         |  |
| Device and SPI Clock Speed Relationship .....               | 150         |  |
| UART Baud Rate with BRGH = 0 .....                          | 160         |  |
| UART Baud Rate with BRGH = 1 .....                          | 160         |  |
| Errata .....  | 8           |  |
| <b>F</b>  |             |  |
| Flash Configuration Words .....                             | 32, 207–210 |  |
| Flash Program Memory  |             |  |
| and Table Instructions .....                                | 49          |  |
| Enhanced ICSP Operation .....                               | 50          |  |
| Programming Algorithm .....                                 | 52          |  |
| RTSP Operation .....  | 50          |  |
| Single-Word Programming .....                               | 54          |  |
| <b>I</b>  |             |  |
| I/O Ports   |             |  |
| Analog Port Configuration .....                             | 106         |  |
| Input Change Notification .....                             | 106         |  |
| Open-Drain Configuration .....                              | 106         |  |
| Parallel (PIO) .....  | 105         |  |
| Peripheral Pin Select .....                                 | 107         |  |
| Pull-ups .....  | 106         |  |
| I <sup>2</sup> C  |             |  |
| Clock Rates .....   | 153         |  |
| Peripheral Remapping Options .....                          | 151         |  |
| Reserved Addresses .....                                    | 153         |  |
| Slave Address Masking .....                                 | 153         |  |
| Idle Mode .....   | 104         |  |
| Instruction Set   |             |  |
| Overview .....  | 223         |  |
| Summary .....   | 221         |  |
| Instruction-Based Power-Saving Modes .....                  | 103         |  |
| Inter-Integrated Circuit. <i>See</i> I <sup>2</sup> C ..... | 151         |  |
| Internet Address .....                                      | 265         |  |
| Interrupts  |             |  |
| Alternate Interrupt Vector Table (AIVT) .....               | 61          |  |
| and Reset Sequence .....                                    | 61          |  |
| Implemented Vectors .....                                   | 63          |  |
| Interrupt Vector Table (IVT) .....                          | 61          |  |
| Registers .....   | 64–92       |  |
| Setup and Service Procedures .....                          | 93          |  |
| Trap Vectors .....  | 62          |  |
| Vector Table .....  | 62          |  |
| <b>J</b>  |             |  |
| JTAG Interface .....  | 214         |  |
| <b>M</b>  |             |  |
| Microchip Internet Web Site .....                           | 265         |  |
| MPLAB ASM30 Assembler, Linker, Librarian .....              | 218         |  |
| MPLAB ICD 2 In-Circuit Debugger .....                       | 219         |  |
| MPLAB ICE 2000 High-Performance Universal                   |             |  |
| In-Circuit Emulator .....                                   | 219         |  |
| MPLAB Integrated Development Environment                    |             |  |
| Software .....  | 217         |  |
| MPLAB PM3 Device Programmer .....                           | 219         |  |
| MPLAB REAL ICE In-Circuit Emulator System .....             | 219         |  |
| MPLINK Object Linker/MPLIB Object Librarian .....           | 218         |  |
| <b>N</b>  |             |  |
| Near Data Space .....                                       | 34          |  |
| <b>O</b>  |             |  |
| Oscillator Configuration                                    |             |  |
| Clock Switching .....                                       | 100         |  |
| Sequence .....  | 101         |  |
| Initial Configuration on POR .....                          | 96          |  |
| Oscillator Modes .....                                      | 96          |  |
| Output Compare  |             |  |
| PWM Mode .....  | 136         |  |
| Period and Duty Cycle Calculation .....                     | 137         |  |
| Single Output Pulse Generation .....                        | 135         |  |
| <b>P</b>  |             |  |
| Packaging   |             |  |
| Details .....   | 249         |  |
| Marking .....   | 247         |  |
| Parallel Master Port. <i>See</i> PMP .....                  | 167         |  |
| Peripheral Enable Bits .....                                | 104         |  |
| Peripheral Module Disable (PMD) bits .....                  | 104         |  |
| Peripheral Pin Select (PPS) .....                           | 107         |  |
| Available Peripherals and Pins .....                        | 107         |  |
| Configuration Control .....                                 | 109         |  |
| Considerations for Use .....                                | 110         |  |
| Input Mapping .....   | 107         |  |
| Mapping Exceptions .....                                    | 109         |  |
| Output Mapping .....  | 108         |  |
| Peripheral Priority .....                                   | 107         |  |
| Registers .....   | 111–124     |  |
| PICSTART Plus Development Programmer .....                  | 220         |  |
| Pinout Descriptions .....                                   | 13–18       |  |
| PMP   |             |  |
| Master Port Examples .....                                  | 174–176     |  |
| Power-Saving Features .....                                 | 103         |  |
| Power-up Requirements .....                                 | 213         |  |
| Product Identification System .....                         | 267         |  |
| Program Memory  |             |  |
| Access Using Table Instructions .....                       | 47          |  |
| Address Construction .....                                  | 45          |  |
| Address Space .....   | 31          |  |
| Flash Configuration Words .....                             | 32          |  |
| Memory Map .....  | 31          |  |
| Organization .....  | 32          |  |
| Program Space Visibility .....                              | 48          |  |
| Pulse-Width Modulation. <i>See</i> PWM .....                | 136         |  |
| <b>R</b>  |             |  |
| Reader Response .....                                       | 266         |  |
| Register Maps   |             |  |
| A/D Converter (ADC) .....                                   | 41          |  |
| Clock Control .....   | 44          |  |
| CPU .....   | 35          |  |
| CRC .....   | 42          |  |
| Dual Comparator .....                                       | 42          |  |
| I <sup>2</sup> C .....                                      | 38          |  |
| ICN .....   | 35          |  |
| Input Capture .....   | 37          |  |
| Interrupt Controller .....                                  | 36          |  |
| NVM .....   | 44          |  |
| Output Compare .....  | 38          |  |
| Pad Configuration .....                                     | 40          |  |
| Parallel Master/Slave Port .....                            | 41          |  |
| Peripheral Pin Select .....                                 | 43          |  |
| PMD .....   | 44          |  |

# PIC24FJ64GA004 FAMILY

|  |          |  |     |
|--|----------|--|-----|
| PORTA.....   | 40       | UxRXREG (UARTx Receive) .....                    | 166 |
| PORTB.....   | 40       | UxSTA (UARTx Status and Control) .....           | 164 |
| PORTC .....  | 40       | UxTXREG (UARTx Transmit).....                    | 166 |
| Real-Time Clock and Calendar (RTCC) .....              | 42       | WKDYHR (RTCC Weekday and Hours Value) .....      | 183 |
| SPI .....  | 39       | YEAR (RTCC Year Value).....                      | 182 |
| Timers.....  | 37       | <b>Resets</b>                                    |     |
| UART .....   | 39       | Clock Source Selection .....                     | 57  |
| <b>Registers</b>                                       |          | Delay Times.....                                 | 58  |
| AD1CHS (A/D Input Select).....                         | 196      | RCON Flags Operation .....                       | 57  |
| AD1CON1 (A/D Control 1).....                           | 193      | SFR States .....                                 | 59  |
| AD1CON2 (A/D Control 2).....                           | 194      | <b>Revision History</b> .....                    | 259 |
| AD1CON3 (A/D Control 3).....                           | 195      | <b>RTCC</b>                                      |     |
| AD1CSSL (A/D Input Scan Select).....                   | 197      | Alarm Configuration.....                         | 186 |
| AD1PCFG (A/D Port Configuration).....                  | 197      | Calibration .....                                | 185 |
| ALCFGRPT (Alarm Configuration).....                    | 181      | Register Mapping .....                           | 178 |
| ALMINSEC (Alarm Minutes and<br>Seconds Value) .....    | 185      | <b>S</b>   |     |
| ALMTHDY (Alarm Month and Day Value) .....              | 184      | Serial Peripheral Interface. <i>See</i> SPI..... | 141 |
| ALWDHR (Alarm Weekday and Hours Value).....            | 184      | SFR Space .....                                  | 34  |
| CLKDIV (Clock Divider) .....                           | 99       | Slective Peripheral Power Control.....           | 104 |
| CMCON (Comparator Control) .....                       | 202      | Sleep Mode .....                                 | 103 |
| CORCON (Core Control) .....                            | 65       | Software Simulator (MPLAB SIM) .....             | 218 |
| CORCON (CPU Control) .....                             | 29       | Software Stack .....                             | 45  |
| CRCCON (CRC Control) .....                             | 189      | <b>T</b>   |     |
| CRCXOR (CRC XOR Polynomial).....                       | 190      | Timer1 .....                                     | 125 |
| CVRCON (Comparator Voltage<br>Reference Control) ..... | 206      | Timer2/3 and Timer4/5 .....                      | 127 |
| CW1 (Flash Configuration Word 1).....                  | 208      | Timing Diagrams                                  |     |
| CW2 (Flash Configuration Word 2).....                  | 210      | CLKO and I/O Timing .....                        | 243 |
| DEVID (Device ID).....                                 | 211      | External Clock Timing.....                       | 241 |
| DEVREV (Device Revision).....                          | 211      | <b>U</b>   |     |
| I2CxCON (I2Cx Control) .....                           | 154      | <b>UART</b>                                      |     |
| I2CxMSK (I2Cx Slave Mode Address Mask) .....           | 158      | Baud Rate Generator (BRG) .....                  | 160 |
| I2CxSTAT (I2Cx Status) .....                           | 156      | Break and Sync Sequence .....                    | 161 |
| ICxCON (Input Capture x Control).....                  | 134      | IrDA Support.....                                | 161 |
| IECn (Interrupt Enable Control 0-4) .....              | 73-77    | Operation of UxCTS and UxRTS Control Pins.....   | 161 |
| IFSn (Interrupt Flag Status 0-4) .....                 | 68-72    | Receiving.....                                   | 161 |
| INTCON1 (Interrupt Control 1).....                     | 66       | Transmitting.....                                | 161 |
| INTCON2 (Interrupt Control 2).....                     | 67       | <b>V</b>   |     |
| IPCn (Interrupt Priority Control 0-18) .....           | 78-92    | VDDCORE/VCAP pin.....                            | 212 |
| MINSEC (RTCC Minutes and Seconds Value).....           | 183      | Voltage Regulator (On-Chip) .....                | 212 |
| MTHDY (RTCC Month and Day Value) .....                 | 182      | and BOR.....                                     | 213 |
| NVMCON (Flash Memory Control) .....                    | 51       | and POR.....                                     | 212 |
| OCxCON (Output Compare x Control) .....                | 139      | Standby Mode .....                               | 213 |
| OSCCON (Oscillator Control) .....                      | 97       | Tracking Mode.....                               | 212 |
| OSCTUN (FRC Oscillator Tune).....                      | 100      | <b>W</b>   |     |
| PADCFG1 (Pad Configuration Control) .....              | 173, 180 | Watchdog Timer (WDT).....                        | 213 |
| PMADDR (PMP Address).....                              | 171      | Winowed Operation .....                          | 214 |
| PMAEN (PMP Enable).....                                | 171      | WWW Address .....                                | 265 |
| PMCON (PMP Control).....                               | 168      | WWW, On-Line Support .....                       | 8   |
| PMMODE (PMP Mode).....                                 | 170      |  |     |
| PMPSTAT (PMP Status).....                              | 172      |  |     |
| RCFGCAL (RTCC Calibration<br>and Configuration) .....  | 179      |  |     |
| RCON (Reset Control) .....                             | 56       |  |     |
| RPINRn (PPS Input Mapping 0-23) .....                  | 111-117  |  |     |
| RPORn (PPS Output Mapping 0-12) .....                  | 118-124  |  |     |
| SPIxCON1 (SPIx Control 1).....                         | 146      |  |     |
| SPIxCON2 (SPIx Control 2).....                         | 147      |  |     |
| SPIxSTAT (SPIx Status and Control) .....               | 144      |  |     |
| SR (ALU STATUS) .....                                  | 28, 65   |  |     |
| T1CON (Timer1 Control).....                            | 126      |  |     |
| TxCON (Timer2 and Timer4 Control).....                 | 130      |  |     |
| TyCON (Timer3 and Timer5 Control).....                 | 131      |  |     |
| UxMODE (UARTx Mode).....                               | 162      |  |     |

# PIC24FJ64GA004 FAMILY

---

NOTES:



# PIC24FJ64GA004 FAMILY

---

## THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at [www.microchip.com](http://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](http://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>**

# PIC24FJ64GA004 FAMILY

---

---

## READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager  
RE: Reader Response  
Total Pages Sent \_\_\_\_\_

From: Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City / State / ZIP / Country \_\_\_\_\_  
Telephone: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_ FAX: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_

Application (optional):

Would you like a reply? \_\_\_Y \_\_\_N

Device: PIC24FJ64GA004 Family Literature Number: DS39881D

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?

---

---

# PIC24FJ64GA004 FAMILY

## 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    | GA0   | 04    | T     | -     | I     | /     | PT    | -     | XXX   |
|---|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>Microchip Trademark</b>                | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| <b>Architecture</b>                       | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| <b>Flash Memory Family</b>                | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| <b>Program Memory Size (KB)</b>           | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| <b>Product Group</b>                      | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| <b>Pin Count</b>                          | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| <b>Tape and Reel Flag (if applicable)</b> | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| <b>Temperature Range</b>                  | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| <b>Package</b>                            | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| <b>Pattern</b>                            | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |

|                            |     |   |
|----------------------------|-----|---|
| <b>Architecture</b>        | 24  | = 16-bit modified Harvard without DSP                                 |
| <b>Flash Memory Family</b> | FJ  | = Flash program memory  |
| <b>Product Group</b>       | GA0 | = General purpose microcontrollers                                    |
| <b>Pin Count</b>           | 02  | = 28-pin  |
|                            | 04  | = 44-pin  |
| <b>Temperature Range</b>   | E   | = -40°C to +125°C (Extended)  |
|                            | I   | = -40°C to +85°C (Industrial)   |
| <b>Package</b>             | SP  | = SPDIP   |
|                            | SO  | = SOIC  |
|                            | SS  | = SSOP  |
|                            | ML  | = QFN   |
|                            | PT  | = TQFP  |
| <b>Pattern</b>             |     | Three-digit QTP, SQTP, Code or Special Requirements (blank otherwise) |
|                            | ES  | = Engineering Sample  |

**Examples:**

- PIC24FJ32GA002-I/ML:  
General purpose PIC24F, 32-Kbyte program memory, 28-pin, Industrial temp., QFN package.
- PIC24FJ64GA004-E/PT:  
General purpose PIC24F, 64-Kbyte program memory, 44-pin, Extended temp., TQFP package.



---

---

## 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](http://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: 886-7-536-4818  
Fax: 886-7-536-4803

**Taiwan - Taipei**  
Tel: 886-2-2500-6610  
Fax: 886-2-2508-0102

**Thailand - Bangkok**  
Tel: 66-2-694-1351  
Fax: 66-2-694-1350

### EUROPE

**Austria - Wels**  
Tel: 43-7242-2244-39  
Fax: 43-7242-2244-393

**Denmark - Copenhagen**  
Tel: 45-4450-2828  
Fax: 45-4485-2829

**France - Paris**  
Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

**Germany - Munich**  
Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

**Italy - Milan**  
Tel: 39-0331-742611  
Fax: 39-0331-466781

**Netherlands - Drunen**  
Tel: 31-416-690399  
Fax: 31-416-690340

**Spain - Madrid**  
Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91

**UK - Wokingham**  
Tel: 44-118-921-5869  
Fax: 44-118-921-5820

Компания «Океан Электроники» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Поставка оригинальных импортных электронных компонентов напрямую с производств Америки, Европы и Азии, а так же с крупнейших складов мира;
- Широкая линейка поставок активных и пассивных импортных электронных компонентов (более 30 млн. наименований);
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Помощь Конструкторского Отдела и консультации квалифицированных инженеров;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Поставка электронных компонентов под контролем ВП;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- При необходимости вся продукция военного и аэрокосмического назначения проходит испытания и сертификацию в лаборатории (по согласованию с заказчиком);
- Поставка специализированных компонентов военного и аэрокосмического уровня качества (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Actel, Aeroflex, Peregrine, VPT, Syfer, Eurofarad, Texas Instruments, MS Kennedy, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits, General Dynamics и др.);

Компания «Океан Электроники» является официальным дистрибьютором и эксклюзивным представителем в России одного из крупнейших производителей разъемов военного и аэрокосмического назначения «JONHON», а так же официальным дистрибьютором и эксклюзивным представителем в России производителя высокотехнологичных и надежных решений для передачи СВЧ сигналов «FORSTAR».



## JONHON

«JONHON» (основан в 1970 г.)

Разъемы специального, военного и аэрокосмического назначения:

(Применяются в военной, авиационной, аэрокосмической, морской, железнодорожной, горно- и нефтедобывающей отраслях промышленности)

«FORSTAR» (основан в 1998 г.)

ВЧ соединители, коаксиальные кабели,  
кабельные сборки и микроволновые компоненты:

(Применяются в телекоммуникациях гражданского и специального назначения, в средствах связи, РЛС, а так же военной, авиационной и аэрокосмической отраслях промышленности).



Телефон: 8 (812) 309-75-97 (многоканальный)

Факс: 8 (812) 320-03-32

Электронная почта: [ocean@oceanchips.ru](mailto:ocean@oceanchips.ru)

Web: <http://oceanchips.ru/>

Адрес: 198099, г. Санкт-Петербург, ул. Калинина, д. 2, корп. 4, лит. А