



**dsPIC33FJ32MC302/304,
dsPIC33FJ64MCX02/X04, and
dsPIC33FJ128MCX02/X04
Data Sheet**

High-Performance,
16-bit Digital Signal Controllers

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

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

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

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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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

Trademarks

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

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

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, PICkit, PICDEM, PICDEM.net, PICTail, PIC³² logo, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rLAB, Select Mode, Total Endurance, 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.

© 2009, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

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

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



dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, AND dsPIC33FJ128MCX02/X04

High-Performance, 16-bit Digital Signal Controllers

Operating Range:

- Up to 40 MIPS operation (at 3.0V -3.6V):
 - Industrial temperature range (-40°C to +85°C)
 - Extended temperature range (-40°C to +125°C)

High-Performance DSC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set
- 16-bit wide data path
- 24-bit wide instructions
- Linear program memory addressing up to 4M instruction words
- Linear data memory addressing up to 64 Kbytes
- 83 base instructions: mostly 1 word/1 cycle
- Two 40-bit accumulators with rounding and saturation options
- Flexible and powerful addressing modes:
 - Indirect
 - Modulo
 - Bit-Reversed
- Software stack
- 16 x 16 fractional/integer multiply operations
- 32/16 and 16/16 divide operations
- Single-cycle multiply and accumulate:
 - Accumulator write back for DSP operations
 - Dual data fetch
- Up to ± 16 -bit shifts for up to 40-bit data

Direct Memory Access (DMA):

- 8-channel hardware DMA
- Up to 2 Kbytes dual ported DMA buffer area (DMA RAM) to store data transferred via DMA:
 - Allows data transfer between RAM and a peripheral while CPU is executing code (no cycle stealing)
- Most peripherals support DMA

Timers/Capture/Compare/PWM:

- Timer/Counters, up to five 16-bit timers:
 - Can pair up to make two 32-bit timers
 - One timer runs as a Real-Time Clock with an external 32.768 kHz oscillator
 - Programmable prescaler
- Input Capture (up to four channels):
 - Capture on up, down or both edges
 - 16-bit capture input functions
 - 4-deep FIFO on each capture
- Output Compare (up to four channels):
 - Single or Dual 16-bit Compare mode
 - 16-bit Glitchless PWM mode
- Hardware Real-Time Clock/Calendar (RTCC):
 - Provides clock, calendar, and alarm functions

Interrupt Controller:

- 5-cycle latency
- Up to 53 available interrupt sources
- Up to three external interrupts
- Seven programmable priority levels
- Five processor exceptions

Digital I/O:

- Peripheral pin Select functionality
- Up to 35 programmable digital I/O pins
- Wake-up/Interrupt-on-Change for up to 31 pins
- Output pins can drive from 3.0V to 3.6V
- Up to 5V output with open drain configuration
- All digital input pins are 5V tolerant
- 4 mA sink on all I/O pins

On-Chip Flash and SRAM:

- Flash program memory (up to 128 Kbytes)
- Data SRAM (up to 16 Kbytes)
- Boot, Secure, and General Security for program Flash

System Management:

- Flexible clock options:
 - External, crystal, resonator, internal RC
 - Fully integrated Phase-Locked Loop (PLL)
 - Extremely low jitter PLL
- Power-up Timer
- Oscillator Start-up Timer/Stabilizer
- Watchdog Timer with its own RC oscillator
- Fail-Safe Clock Monitor
- Reset by multiple sources

Power Management:

- On-chip 2.5V voltage regulator
- Switch between clock sources in real time
- Idle, Sleep, and Doze modes with fast wake-up

Analog-to-Digital Converters (ADCs):

- 10-bit, 1.1 Msps or 12-bit, 500 Ksps conversion:
 - Two and four simultaneous samples (10-bit ADC)
 - Up to nine input channels with auto-scanning
 - Conversion start can be manual or synchronized with one of four trigger sources
 - Conversion possible in Sleep mode
 - ± 2 LSb max integral nonlinearity
 - ± 1 LSb max differential nonlinearity

Audio Digital-to-Analog Converter (DAC):

- 16-bit Dual Channel DAC module
- 100 Ksps maximum sampling rate
- Second-Order Digital Delta-Sigma Modulator

Comparator Module:

- Two analog comparators with programmable input/output configuration

CMOS Flash Technology:

- Low-power, high-speed Flash technology
- Fully static design
- 3.3V ($\pm 10\%$) operating voltage
- Industrial and Extended temperature
- Low power consumption

Motor Control Peripherals:

- 6-channel 16-bit Motor Control PWM:
 - Three duty cycle generators
 - Independent or Complementary mode
 - Programmable dead-time and output polarity
 - Edge-aligned or center-aligned
 - Manual output override control
 - One Fault input
 - Trigger for ADC conversions
 - PWM frequency for 16-bit resolution (@ 40 MIPS) = 1220 Hz for Edge-Aligned mode, 610 Hz for Center-Aligned mode
 - PWM frequency for 11-bit resolution (@ 40 MIPS) = 39.1 kHz for Edge-Aligned mode, 19.55 kHz for Center-Aligned mode
- 2-channel 16-bit Motor Control PWM:
 - One duty cycle generator
 - Independent or Complementary mode
 - Programmable dead time and output polarity
 - Edge-aligned or center-aligned
 - Manual output override control
 - One Fault input
 - Trigger for ADC conversions
 - PWM frequency for 16-bit resolution (@ 40 MIPS) = 1220 Hz for Edge-Aligned mode, 610 Hz for Center-Aligned mode
 - PWM frequency for 11-bit resolution (@ 40 MIPS) = 39.1 kHz for Edge-Aligned mode, 19.55 kHz for Center-Aligned mode
- 2-Quadrature Encoder Interface module:
 - Phase A, Phase B, and index pulse input
 - 16-bit up/down position counter
 - Count direction status
 - Position Measurement (x2 and x4) mode
 - Programmable digital noise filters on inputs
 - Alternate 16-bit Timer/Counter mode
 - Interrupt on position counter rollover/underflow

Communication Modules:

- 4-wire SPI (up to two modules):
 - Framing supports I/O interface to simple codecs
 - Supports 8-bit and 16-bit data
 - Supports all serial clock formats and sampling modes
- I²C™:
 - Full Multi-Master Slave mode support
 - 7-bit and 10-bit addressing
 - Bus collision detection and arbitration
 - Integrated signal conditioning
 - Slave address masking
- UART (up to two modules):
 - Interrupt on address bit detect
 - Interrupt on UART error
 - Wake-up on Start bit from Sleep mode
 - 4-character TX and RX FIFO buffers
 - LIN bus support
 - IrDA® encoding and decoding in hardware
 - High-Speed Baud mode
 - Hardware Flow Control with CTS and RTS
- Enhanced CAN (ECAN™ module) 2.0B active:
 - Up to eight transmit and up to 32 receive buffers
 - 16 receive filters and three masks
 - Loopback, Listen Only and Listen All
 - Messages modes for diagnostics and bus monitoring
 - Wake-up on CAN message
 - Automatic processing of Remote Transmission Requests
 - FIFO mode using DMA
 - DeviceNet™ addressing support
- Parallel Master Slave Port (PMP/EPSP):
 - Supports 8-bit or 16-bit data
 - Supports 16 address lines
- Programmable Cyclic Redundancy Check (CRC):
 - Programmable bit length for the CRC generator polynomial (up to 16-bit length)
 - 8-deep, 16-bit or 16-deep, 8-bit FIFO for data input

Packaging:

- 28-pin SDIP/SOIC/QFN-S
- 44-pin TQFP/QFN

| |
|---|
| Note: See the device variant table for exact peripheral features per device. |
|---|

dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, AND dsPIC33FJ128MCX02/X04

dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, AND dsPIC33FJ128MCX02/X04 PRODUCT FAMILIES

The device names, pin counts, memory sizes, and peripheral availability of each device are listed below. The following pages show their pinout diagrams.

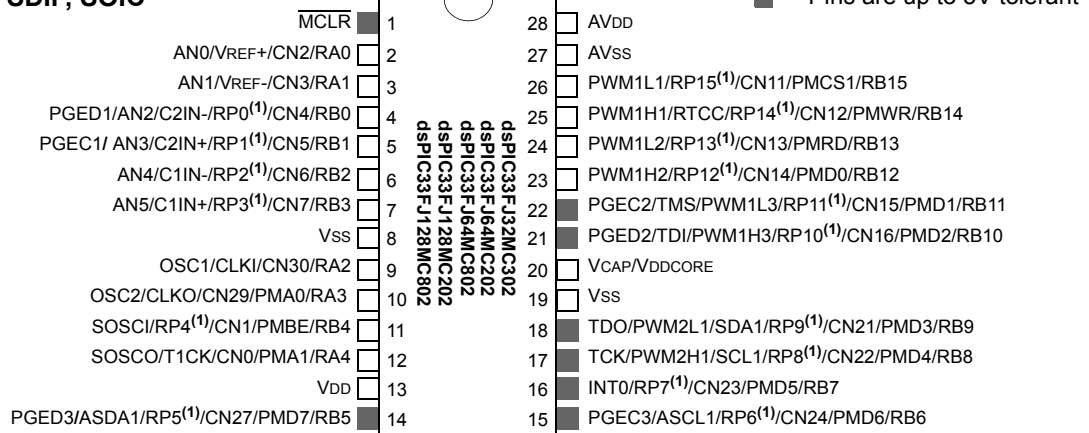
dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 Controller Families

| Device | Pins | Program Flash Memory (Kbyte) | RAM (Kbyte) ⁽¹⁾ | Remappable Peripheral | | | | | | | | | | RTCC | I ² C™ | CRC Generator | 10-bit/12-bit ADC (Channels) | 6-pin 16-bit DAC | Analog Comparator (2 Channels/Voltage Regulator) | 8-bit Parallel Master Port (Address Lines) | I/O Pins | Packages |
|-------------------|------|------------------------------|----------------------------|-----------------------|-----------------------------|---------------|-----------------------------|---|------------------------------|------|-----|-------|------------------------------------|------|-------------------|---------------|------------------------------|------------------|--|--|----------|-----------------|
| | | | | Remappable Pins | 16-bit Timer ⁽²⁾ | Input Capture | Output Compare Standard PWM | Motor Control PWM (Channels) ⁽³⁾ | Quadrature Encoder Interface | UART | SPI | ECAN™ | External Interrupts ⁽⁴⁾ | | | | | | | | | |
| dsPIC33FJ128MC804 | 44 | 128 | 16 | 26 | 5 | 4 | 4 | 6, 2 | 2 | 2 | 2 | 1 | 3 | 1 | 1 | 1 | 9 | 1 | 1/1 | 11 | 35 | QFN TQFP |
| dsPIC33FJ128MC802 | 28 | 128 | 16 | 16 | 5 | 4 | 4 | 6, 2 | 2 | 2 | 2 | 1 | 3 | 1 | 1 | 1 | 6 | 0 | 1/0 | 2 | 21 | SDIP SOIC QFN-S |
| dsPIC33FJ128MC204 | 44 | 128 | 8 | 26 | 5 | 4 | 4 | 6, 2 | 2 | 2 | 2 | 0 | 3 | 1 | 1 | 1 | 9 | 0 | 1/1 | 11 | 35 | QFN TQFP |
| dsPIC33FJ128MC202 | 28 | 128 | 8 | 16 | 5 | 4 | 4 | 6, 2 | 2 | 2 | 2 | 0 | 3 | 1 | 1 | 1 | 6 | 0 | 1/0 | 2 | 21 | SDIP SOIC QFN-S |
| dsPIC33FJ64MC804 | 44 | 64 | 16 | 26 | 5 | 4 | 4 | 6, 2 | 2 | 2 | 2 | 1 | 3 | 1 | 1 | 1 | 9 | 1 | 1/1 | 11 | 35 | QFN TQFP |
| dsPIC33FJ64MC802 | 28 | 64 | 16 | 16 | 5 | 4 | 4 | 6, 2 | 2 | 2 | 2 | 1 | 3 | 1 | 1 | 1 | 6 | 0 | 1/0 | 2 | 21 | SDIP SOIC QFN-S |
| dsPIC33FJ64MC204 | 44 | 64 | 8 | 26 | 5 | 4 | 4 | 6, 2 | 2 | 2 | 2 | 0 | 3 | 1 | 1 | 1 | 9 | 0 | 1/1 | 11 | 35 | QFN TQFP |
| dsPIC33FJ64MC202 | 28 | 64 | 8 | 16 | 5 | 4 | 4 | 6, 2 | 2 | 2 | 2 | 0 | 3 | 1 | 1 | 1 | 6 | 0 | 1/0 | 2 | 21 | SDIP SOIC QFN-S |
| dsPIC33FJ32MC304 | 44 | 32 | 4 | 26 | 5 | 4 | 4 | 6, 2 | 2 | 2 | 2 | 0 | 3 | 1 | 1 | 1 | 9 | 0 | 1/1 | 11 | 35 | QFN TQFP |
| dsPIC33FJ32MC302 | 28 | 32 | 4 | 16 | 5 | 4 | 4 | 6, 2 | 2 | 2 | 2 | 0 | 3 | 1 | 1 | 1 | 6 | 0 | 1/0 | 2 | 21 | SDIP SOIC QFN-S |

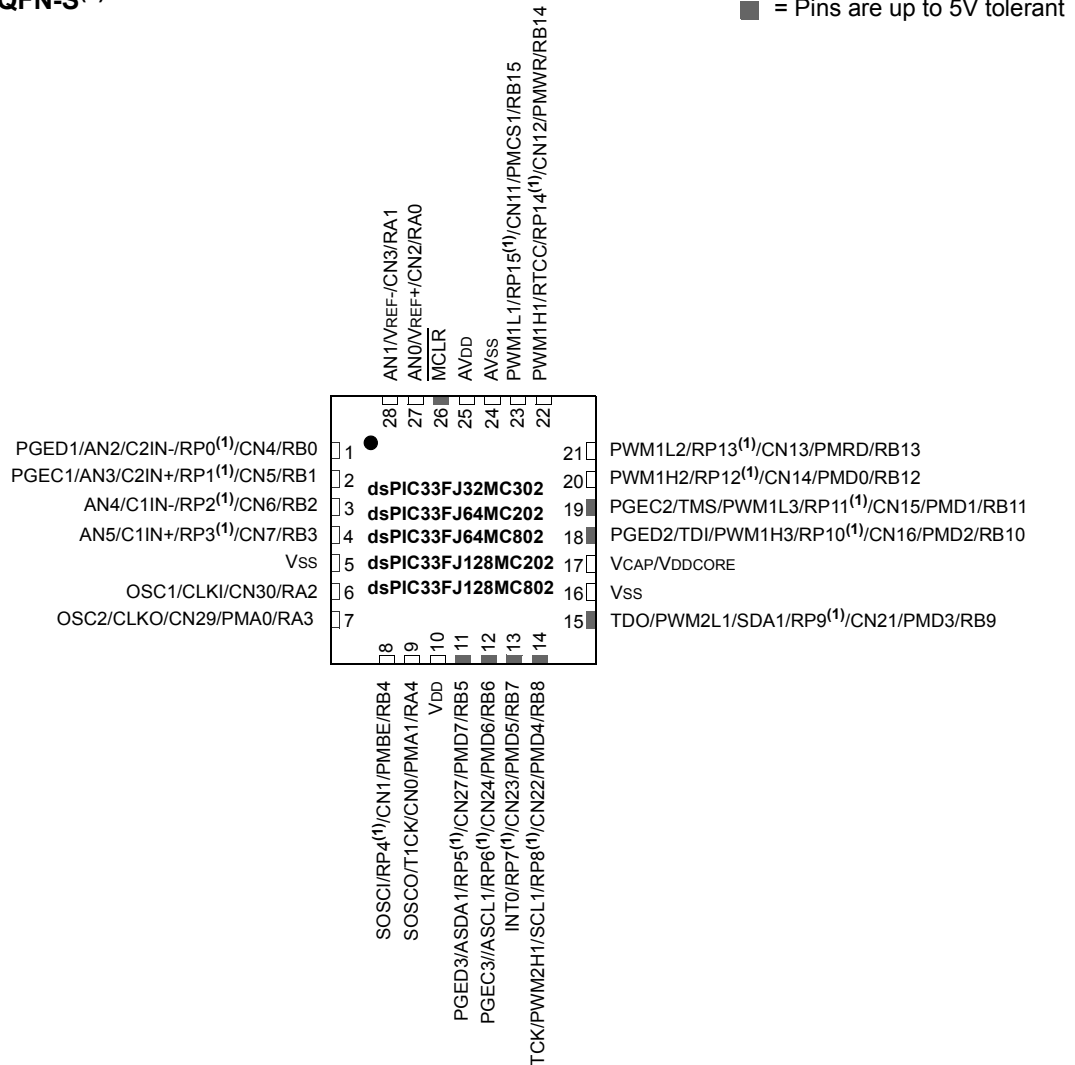
- Note**
- 1: RAM size is inclusive of 2 Kbytes of DMA RAM for all devices except dsPIC33FJ32MC302/304, which include 1 Kbyte of DMA RAM.
 - 2: Only four out of five timers are remappable.
 - 3: Only PWM fault pins are remappable.
 - 4: Only two out of three interrupts are remappable.

Pin Diagrams

28-Pin SDIP, SOIC

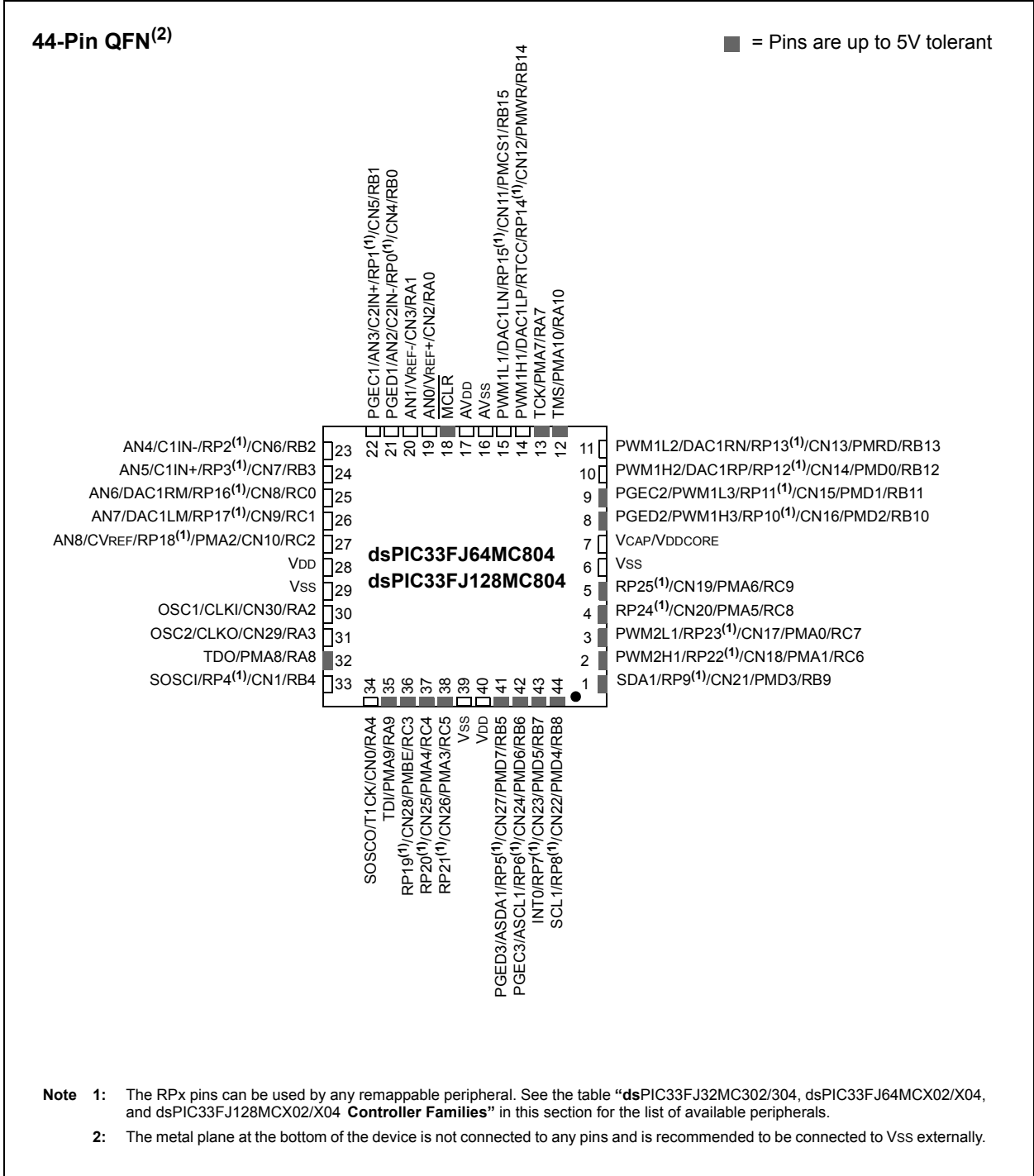


28-Pin QFN-S⁽²⁾

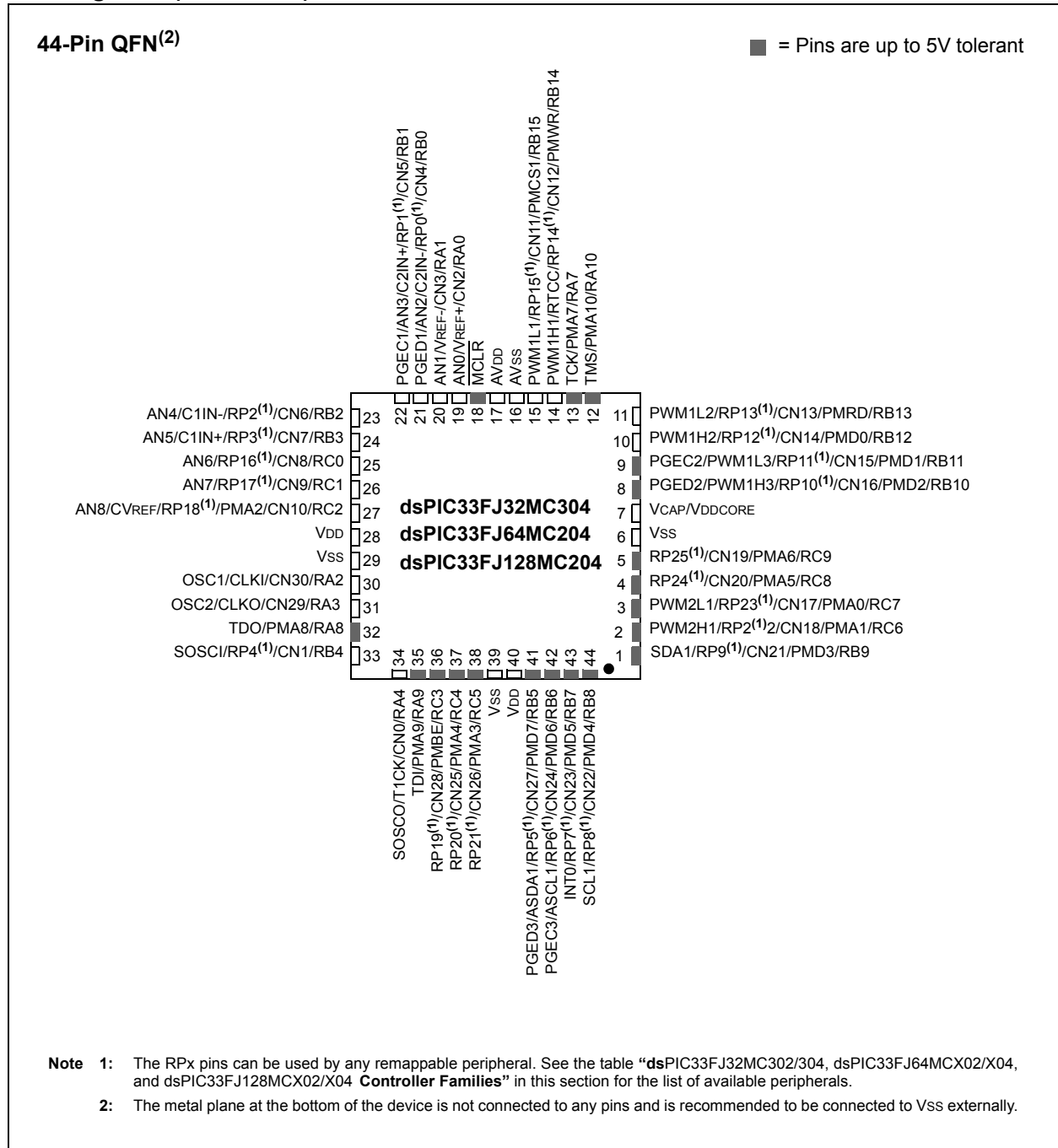


- Note 1:** The RPx pins can be used by any remappable peripheral. See the table “dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 Controller Families” in this section for the list of available peripherals.
- Note 2:** The metal plane at the bottom of the device is not connected to any pins and is recommended to be connected to VSS externally.

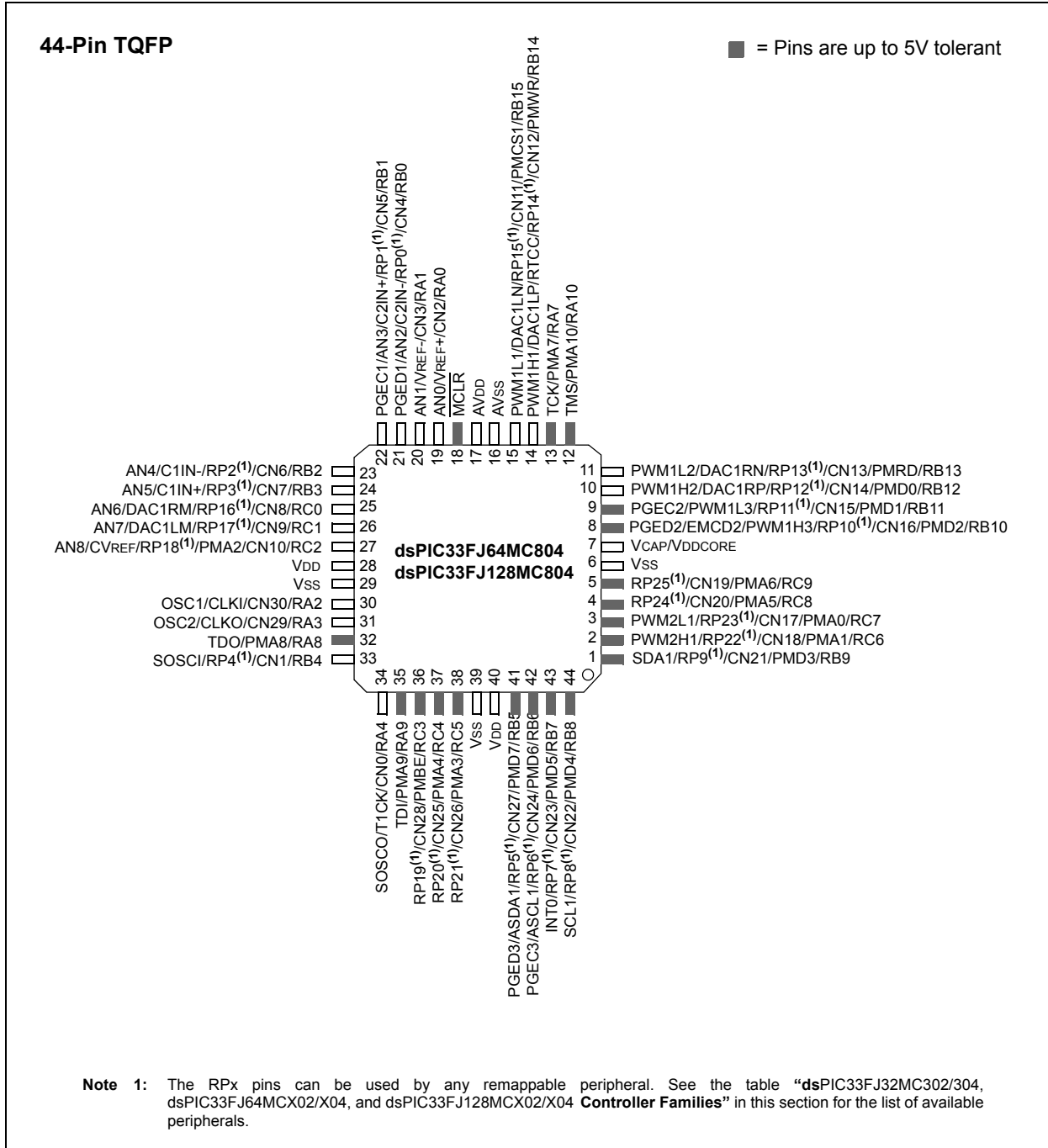
Pin Diagrams (Continued)



Pin Diagrams (Continued)



Pin Diagrams (Continued)



Pin Diagrams (Continued)

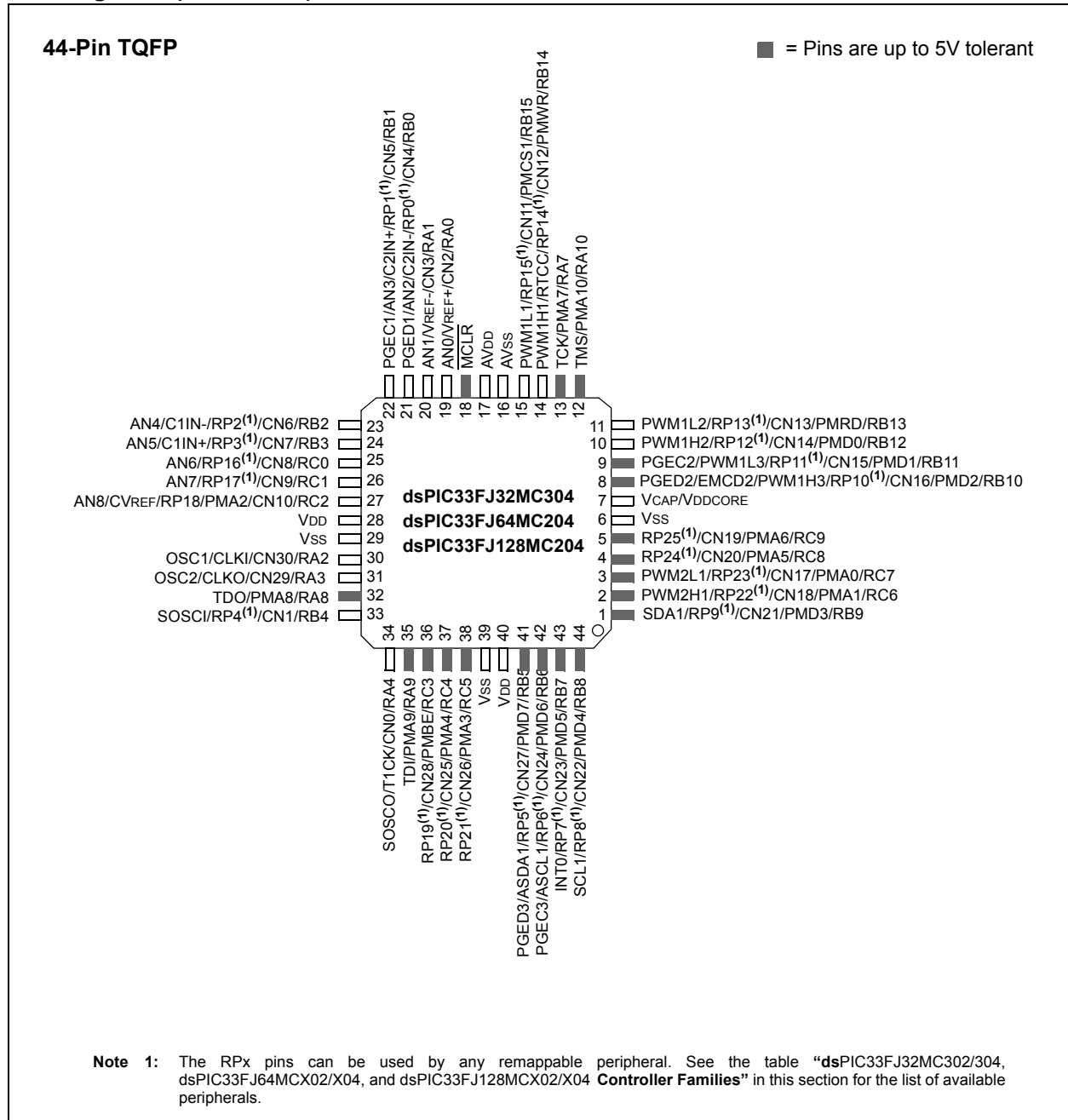


Table of Contents

| | |
|---|-----|
| dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 Product Families..... | 4 |
| 1.0 Device Overview | 13 |
| 2.0 Guidelines for Getting Started with 16-bit Digital Signal Controllers | 19 |
| 3.0 CPU | 23 |
| 4.0 Memory Organization | 35 |
| 5.0 Flash Program Memory | 71 |
| 6.0 Resets | 77 |
| 7.0 Interrupt Controller | 85 |
| 8.0 Direct Memory Access (DMA) | 127 |
| 9.0 Oscillator Configuration | 139 |
| 10.0 Power-Saving Features | 151 |
| 11.0 I/O Ports | 157 |
| 12.0 Timer1 | 189 |
| 13.0 Timer2/3 And Timer4/5 feature | 191 |
| 14.0 Input Capture | 197 |
| 15.0 Output Compare | 199 |
| 16.0 Motor Control PWM Module | 203 |
| 17.0 Quadrature Encoder Interface (QEI) Module | 217 |
| 18.0 Serial Peripheral Interface (SPI)..... | 221 |
| 19.0 Inter-Integrated Circuit (I ² C™) | 227 |
| 20.0 Universal Asynchronous Receiver Transmitter (UART) | 235 |
| 21.0 Enhanced CAN (ECAN™) Module | 241 |
| 22.0 10-bit/12-bit Analog-to-Digital Converter (ADC1)..... | 267 |
| 23.0 Audio Digital-to-Analog Converter (DAC)..... | 281 |
| 24.0 Comparator Module..... | 287 |
| 25.0 Real-Time Clock and Calendar (RTCC) | 293 |
| 26.0 Programmable Cyclic Redundancy Check (CRC) Generator | 303 |
| 27.0 Parallel Master Port (PMP)..... | 307 |
| 28.0 Special Features | 315 |
| 29.0 Instruction Set Summary | 325 |
| 30.0 Development Support..... | 333 |
| 31.0 Electrical Characteristics | 337 |
| 32.0 Packaging Information..... | 383 |
| Appendix A: Revision History..... | 393 |
| Index | 399 |
| The Microchip Web Site | 405 |
| Customer Change Notification Service | 405 |
| Customer Support | 405 |
| Reader Response | 406 |
| Product Identification System..... | 407 |

TO OUR VALUED CUSTOMERS

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

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

Most Current Data Sheet

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

<http://www.microchip.com>

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

Errata

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

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

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

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

Customer Notification System

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

NOTES:

1.0 DEVICE OVERVIEW

Note: This data sheet summarizes the features of dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the related section of the “*dsPIC33F Family Reference Manual*”, which is available from the Microchip website (www.microchip.com)

This document contains device specific information for the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 Digital Signal Controller (DSC) Devices. The dsPIC33F devices contain extensive Digital Signal Processor (DSP) functionality with a high performance 16-bit microcontroller (MCU) architecture.

Figure 1-1 shows a general block diagram of the core and peripheral modules in the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. Table 1-1 lists the functions of the various pins shown in the pinout diagrams.

FIGURE 1-1: dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, AND dsPIC33FJ128MCX02/X04 BLOCK DIAGRAM

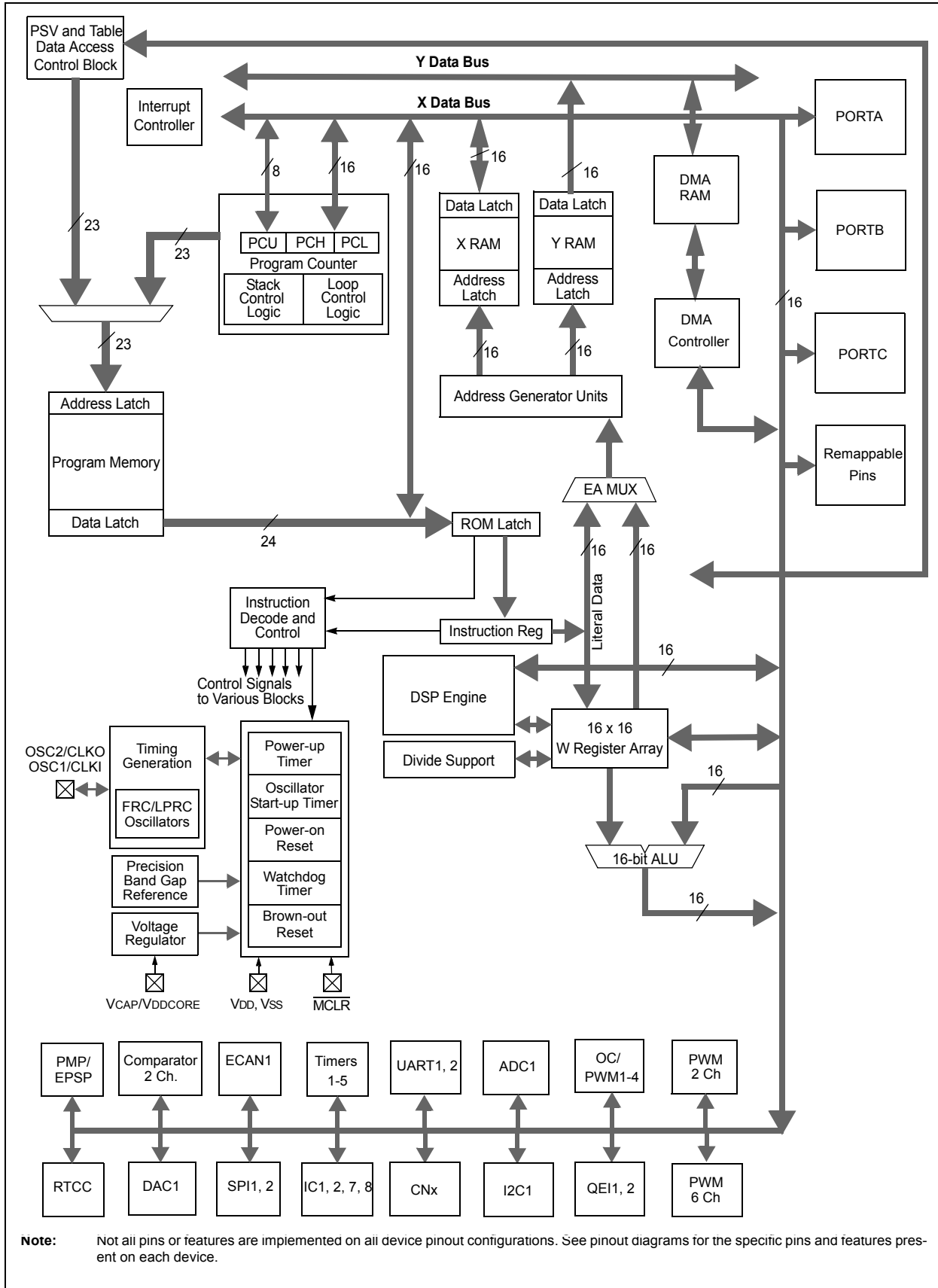


TABLE 1-1: PINOUT I/O DESCRIPTIONS

| Pin Name | Pin Type | Buffer Type | PPS | Description |
|----------|----------|-------------|-----|---|
| AN0-AN8 | I | Analog | No | Analog input channels. |
| CLKI | I | ST/CMOS | No | External clock source input. Always associated with OSC1 pin function. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes. |
| CLKO | O | — | No | Always associated with OSC2 pin function. |
| OSC1 | I | ST/CMOS | No | Oscillator crystal input. ST buffer when configured in RC mode; CMOS otherwise. |
| OSC2 | I/O | — | No | Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes. |
| SOSCI | I | ST/CMOS | No | 32.768 kHz low-power oscillator crystal input; CMOS otherwise. |
| SOSCO | O | — | No | 32.768 kHz low-power oscillator crystal output. |
| CN0-CN30 | I | ST | No | Change notification inputs. Can be software programmed for internal weak pull-ups on all inputs. |
| IC1-IC2 | I | ST | Yes | Capture inputs 1/2. |
| IC7-IC8 | I | ST | Yes | Capture inputs 7/8. |
| OCFA | I | ST | Yes | Compare Fault A input (for Compare Channels 1, 2, 3 and 4). |
| OC1-OC4 | O | — | Yes | Compare outputs 1 through 4. |
| INT0 | I | ST | No | External interrupt 0. |
| INT1 | I | ST | Yes | External interrupt 1. |
| INT2 | I | ST | Yes | External interrupt 2. |
| RA0-RA4 | I/O | ST | No | PORTA is a bidirectional I/O port. |
| RA7-RA10 | I/O | ST | No | PORTA is a bidirectional I/O port. |
| RB0-RB15 | I/O | ST | No | PORTB is a bidirectional I/O port. |
| RC0-RC9 | I/O | ST | No | PORTC is a bidirectional I/O port. |
| T1CK | I | ST | No | Timer1 external clock input. |
| T2CK | I | ST | Yes | Timer2 external clock input. |
| T3CK | I | ST | Yes | Timer3 external clock input. |
| T4CK | I | ST | Yes | Timer4 external clock input. |
| T5CK | I | ST | Yes | Timer5 external clock input. |
| U1CTS | I | ST | Yes | UART1 clear to send. |
| U1RTS | O | — | Yes | UART1 ready to send. |
| U1RX | I | ST | Yes | UART1 receive. |
| U1TX | O | — | Yes | UART1 transmit. |
| U2CTS | I | ST | Yes | UART2 clear to send. |
| U2RTS | O | — | Yes | UART2 ready to send. |
| U2RX | I | ST | Yes | UART2 receive. |
| U2TX | O | — | Yes | UART2 transmit. |
| SCK1 | I/O | ST | Yes | Synchronous serial clock input/output for SPI1. |
| SDI1 | I | ST | Yes | SPI1 data in. |
| SDO1 | O | — | Yes | SPI1 data out. |
| SS1 | I/O | ST | Yes | SPI1 slave synchronization or frame pulse I/O. |
| SCK2 | I/O | ST | Yes | Synchronous serial clock input/output for SPI2. |
| SDI2 | I | ST | Yes | SPI2 data in. |
| SDO2 | O | — | Yes | SPI2 data out. |
| SS2 | I/O | ST | Yes | SPI2 slave synchronization or frame pulse I/O. |
| SCL1 | I/O | ST | No | Synchronous serial clock input/output for I2C1. |
| SDA1 | I/O | ST | No | Synchronous serial data input/output for I2C1. |
| ASCL1 | I/O | ST | No | Alternate synchronous serial clock input/output for I2C1. |
| ASDA1 | I/O | ST | No | Alternate synchronous serial data input/output for I2C1. |

Legend: CMOS = CMOS compatible input or output Analog = Analog input P = Power
 ST = Schmitt Trigger input with CMOS levels O = Output I = Input
 PPS = Peripheral Pin Select TTL = TTL input buffer

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

| Pin Name | Pin Type | Buffer Type | PPS | Description |
|--------------|----------|-------------|-----|---|
| TMS | I | ST | No | JTAG Test mode select pin. |
| TCK | I | ST | No | JTAG test clock input pin. |
| TDI | I | ST | No | JTAG test data input pin. |
| TDO | O | — | No | JTAG test data output pin. |
| INDX1 | I | ST | Yes | Quadrature Encoder Index1 Pulse input. |
| QEA1 | I | ST | Yes | Quadrature Encoder Phase A input in QE11 mode. Auxiliary Timer External Clock/Gate input in Timer mode. |
| QEB1 | I | ST | Yes | Quadrature Encoder Phase A input in QE11 mode. Auxiliary Timer External Clock/Gate input in Timer mode. |
| UPDN1 | O | CMOS | Yes | Position Up/Down Counter Direction State. |
| INDX2 | I | ST | Yes | Quadrature Encoder Index2 Pulse input. |
| QEA2 | I | ST | Yes | Quadrature Encoder Phase A input in QE12 mode. Auxiliary Timer External Clock/Gate input in Timer mode. |
| QEB2 | I | ST | Yes | Quadrature Encoder Phase A input in QE12 mode. Auxiliary Timer External Clock/Gate input in Timer mode. |
| UPDN2 | O | CMOS | Yes | Position Up/Down Counter Direction State. |
| C1RX | I | ST | Yes | ECAN1 bus receive pin. |
| C1TX | O | — | Yes | ECAN1 bus transmit pin. |
| RTCC | O | — | No | Real-Time Clock Alarm Output. |
| CVREF | O | ANA | No | Comparator Voltage Reference Output. |
| C1IN- | I | ANA | No | Comparator 1 Negative Input. |
| C1IN+ | I | ANA | No | Comparator 1 Positive Input. |
| C1OUT | O | — | Yes | Comparator 1 Output. |
| C2IN- | I | ANA | No | Comparator 2 Negative Input. |
| C2IN+ | I | ANA | No | Comparator 2 Positive Input. |
| C2OUT | O | — | Yes | Comparator 2 Output. |
| PMA0 | I/O | TTL/ST | No | Parallel Master Port Address Bit 0 Input (Buffered Slave modes) and Output (Master modes). |
| PMA1 | I/O | TTL/ST | No | Parallel Master Port Address Bit 1 Input (Buffered Slave modes) and Output (Master modes). |
| PMA2 -PMPA10 | O | — | No | Parallel Master Port Address (Demultiplexed Master Modes). |
| PMBE | O | — | No | Parallel Master Port Byte Enable Strobe. |
| PMCS1 | O | — | No | Parallel Master Port Chip Select 1 Strobe. |
| PMD0-PMPD7 | I/O | TTL/ST | No | Parallel Master Port Data (Demultiplexed Master mode) or Address/Data (Multiplexed Master modes). |
| PMRD | O | — | No | Parallel Master Port Read Strobe. |
| PMWR | O | — | No | Parallel Master Port Write Strobe. |
| DAC1RN | O | — | No | DAC1 Negative Output. |
| DAC1RP | O | — | No | DAC1 Positive Output. |
| DAC1RM | O | — | No | DAC1 Output indicating middle point value (typically 1.65V). |
| DAC2RN | O | — | No | DAC2 Negative Output. |
| DAC2RP | O | — | No | DAC2 Positive Output. |
| DAC2RM | O | — | No | DAC2 Output indicating middle point value (typically 1.65V). |

Legend: CMOS = CMOS compatible input or output Analog = Analog input P = Power
 ST = Schmitt Trigger input with CMOS levels O = Output I = Input
 PPS = Peripheral Pin Select TTL = TTL input buffer

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

| Pin Name | Pin Type | Buffer Type | PPS | Description |
|--------------|----------|-------------|-----|--|
| FLTA1 | I | ST | Yes | PWM1 Fault A input. |
| PWM1L1 | O | — | No | PWM1 Low output 1 |
| PWM1H1 | O | — | No | PWM1 High output 1 |
| PWM1L2 | O | — | No | PWM1 Low output 2 |
| PWM1H2 | O | — | No | PWM1 High output 2 |
| PWM1L3 | O | — | No | PWM1 Low output 3 |
| PWM1H3 | O | — | No | PWM1 High output 3 |
| FLTA2 | I | ST | Yes | PWM2 Fault A input. |
| PWM2L1 | O | — | No | PWM2 Low output 1 |
| PWM2H1 | O | — | No | PWM2 High output 1 |
| PGED1 | I/O | ST | No | Data I/O pin for programming/debugging communication channel 1. |
| PGEC1 | I | ST | No | Clock input pin for programming/debugging communication channel 1. |
| PGED2 | I/O | ST | No | Data I/O pin for programming/debugging communication channel 2. |
| PGEC2 | I | ST | No | Clock input pin for programming/debugging communication channel 2. |
| PGED3 | I/O | ST | No | Data I/O pin for programming/debugging communication channel 3. |
| PGEC3 | I | ST | No | Clock input pin for programming/debugging communication channel 3. |
| MCLR | I/P | ST | No | Master Clear (Reset) input. This pin is an active-low Reset to the device. |
| AVDD | P | P | No | Positive supply for analog modules. This pin must be connected at all times. |
| AVSS | P | P | No | Ground reference for analog modules. |
| VDD | P | — | No | Positive supply for peripheral logic and I/O pins. |
| VCAP/VDDCORE | P | — | No | CPU logic filter capacitor connection. |
| VSS | P | — | No | Ground reference for logic and I/O pins. |
| VREF+ | I | Analog | No | Analog voltage reference (high) input. |
| VREF- | I | Analog | No | Analog voltage reference (low) input. |

Legend: CMOS = CMOS compatible input or output Analog = Analog input P = Power
 ST = Schmitt Trigger input with CMOS levels O = Output I = Input
 PPS = Peripheral Pin Select TTL = TTL input buffer

NOTES:

2.0 GUIDELINES FOR GETTING STARTED WITH 16-BIT DIGITAL SIGNAL CONTROLLERS

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the *dsPIC33F Family Reference Manual*, which is available from the Microchip website (www.microchip.com).

2.1 Basic Connection Requirements

Getting started with the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 family of 16-bit Digital Signal Controllers (DSC) requires attention to a minimal set of device pin connections before proceeding with development. The following is a list of pin names, which must always be connected:

- All VDD and VSS pins
(see **Section 2.2 “Decoupling Capacitors”**)
- All AVDD and AVSS pins (regardless if ADC module is not used)
(see **Section 2.2 “Decoupling Capacitors”**)
- VCAP/VDDCORE
(see **Section 2.3 “Capacitor on Internal Voltage Regulator (Vcap/Vddcore)”**)
- MCLR pin
(see **Section 2.4 “Master Clear (MCLR) Pin”**)
- PGECx/PGEDx pins used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes
(see **Section 2.5 “ICSP Pins”**)
- OSC1 and OSC2 pins when 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 ADC module is implemented

Note: The AVDD and AVSS pins must be connected independent of the ADC voltage reference source.

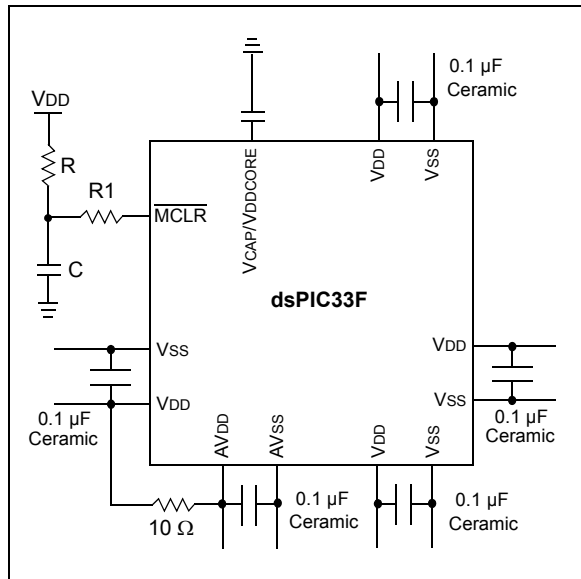
2.2 Decoupling Capacitors

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:** Recommendation of 0.1 μF (100 nF), 10-20V. This capacitor should be a low-ESR and have resonance frequency in the range of 20 MHz and higher. It is recommended that ceramic capacitors be used.
- **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 within one-quarter inch (6 mm) in length.
- **Handling high frequency noise:** If the board is experiencing high frequency noise, upward of tens of MHz, add a second ceramic-type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 μF to 0.001 μF . Place this second capacitor next to the 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. For example, 0.1 μF in parallel with 0.001 μF .
- **Maximizing performance:** On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum thereby reducing PCB track inductance.

FIGURE 2-1: RECOMMENDED MINIMUM CONNECTION



2.2.1 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 DSCs to supply a local power source. The value of the tank capacitor should be determined based on the trace resistance that connects the power supply source to the device, and the maximum current drawn by the device in the application. In other words, select the tank capacitor so that it meets the acceptable voltage sag at the device. Typical values range from 4.7 µF to 47 µF.

2.3 Capacitor on Internal Voltage Regulator (VCAP/VDDCORE)

A low-ESR (< 5 Ohms) capacitor is required on the VCAP/VDDCORE pin, which is used to stabilize the voltage regulator output voltage. The VCAP/VDDCORE pin must not be connected to VDD, and must have a capacitor between 4.7 µF and 10 µF, 16V connected to ground. The type can be ceramic or tantalum. Refer to **Section 31.0 “Electrical Characteristics”** for additional information.

The placement of this capacitor should be close to the VCAP/VDDCORE. It is recommended that the trace length not exceed one-quarter inch (6 mm). Refer to **Section 28.2 “On-Chip Voltage Regulator”** for details.

2.4 Master Clear (MCLR) Pin

The MCLR pin provides for two specific device functions:

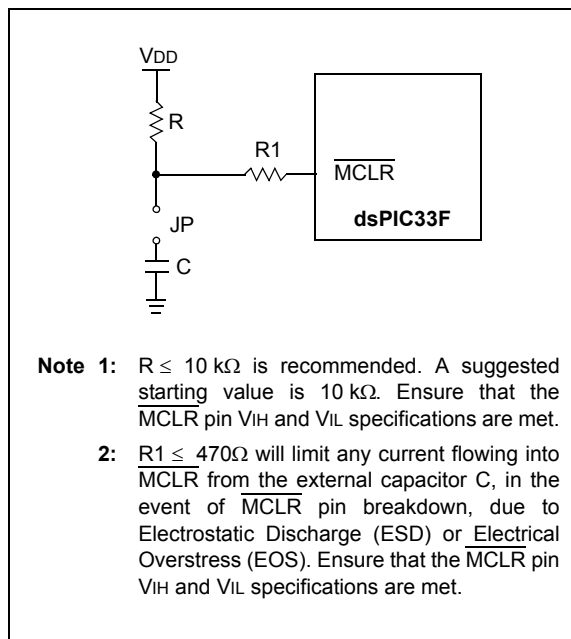
- Device Reset
- Device programming and debugging.

During device programming and debugging, the resistance and capacitance that can be added to the pin must be considered. Device programmers and debuggers drive the MCLR pin. Consequently, specific voltage levels (VIH and VIL) and fast signal transitions must not be adversely affected. Therefore, specific values of R and C will need to be adjusted based on the application and PCB requirements.

For example, as shown in Figure 2-2, it is recommended that the capacitor C, be isolated from the MCLR pin during programming and debugging operations.

Place the components shown in Figure 2-2 within one-quarter inch (6 mm) from the MCLR pin.

FIGURE 2-2: EXAMPLE OF MCLR PIN CONNECTIONS



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 Ohms.

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 (V_{IH}) and input low (V_{IL}) requirements.

Ensure that the “Communication Channel Select” (i.e., PGECx/PGEDx pins) programmed into the device matches the physical connections for the ICSP to MPLAB® ICD 2, MPLAB® ICD 3, or MPLAB® REAL ICE™.

For more information on ICD 2, ICD 3, and REAL ICE connection requirements, refer to the following documents that are available on the Microchip website.

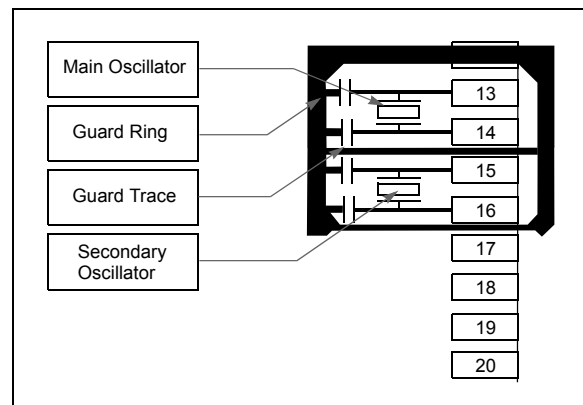
- “MPLAB® ICD 2 In-Circuit Debugger User’s Guide” DS51331
- “Using MPLAB® ICD 2” (poster) DS51265
- “MPLAB® ICD 2 Design Advisory” DS51566
- “Using MPLAB® ICD 3” (poster) DS51765
- “MPLAB® ICD 3 Design Advisory” DS51764
- “MPLAB® REAL ICE™ In-Circuit Debugger User’s Guide” DS51616
- “Using MPLAB® REAL ICE™” (poster) DS51749

2.6 External Oscillator Pins

Many DSCs have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator (refer to **Section 9.0 “Oscillator Configuration”** for details).

The oscillator circuit should be placed on the same side of the board as the device. Also, place the oscillator circuit close to the respective oscillator pins, not exceeding one-half inch (12 mm) distance between them. 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 them 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. A suggested layout is shown in Figure 2-3.

FIGURE 2-3: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT



2.7 Oscillator Value Conditions on Device Start-up

If the PLL of the target device is enabled and configured for the device start-up oscillator, the maximum oscillator source frequency must be limited to $4 \text{ MHz} < F_{\text{IN}} < 8 \text{ MHz}$ to comply with device PLL start-up conditions. This means that if the external oscillator frequency is outside this range, the application must start-up in the FRC mode first. The default PLL settings after a POR with an oscillator frequency outside this range will violate the device operating speed.

Once the device powers up, the application firmware can initialize the PLL SFRs, CLKDIV, and PLLDBF to a suitable value, and then perform a clock switch to the Oscillator + PLL clock source. Note that clock switching must be enabled in the device Configuration word.

2.8 Configuration of Analog and Digital Pins During ICSP Operations

If MPLAB ICD 2, ICD 3, or REAL ICE is selected as a debugger, it automatically initializes all of the A/D input pins (ANx) as “digital” pins, by setting all bits in the AD1PCFGL register.

The bits in this register that correspond to the A/D pins that are initialized by MPLAB ICD 2, ICD 3, or REAL ICE, must not be cleared 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 clear the corresponding bits in the AD1PCFGL register during initialization of the ADC module.

When MPLAB ICD 2, ICD 3, or REAL ICE is used as a programmer, the user application firmware must correctly configure the AD1PCFGL register. 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.

2.9 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic-low state.

Alternatively, connect a 1k to 10k resistor to V_{SS} on unused pins and drive the output to logic low.

3.0 CPU

Note: This data sheet summarizes the features of dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 2. “CPU” (DS70204), which is available from the Microchip website (www.microchip.com).

3.1 Overview

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 CPU module has a 16-bit (data) modified Harvard architecture with an enhanced instruction set, including significant support for DSP. The CPU has a 24-bit instruction word with a variable length opcode field. The Program Counter (PC) is 23 bits wide and addresses up to 4M x 24 bits of user program memory space. The actual amount of program memory implemented varies by device. 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 `DO` and `REPEAT` instructions, both of which are interruptible at any time.

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices have sixteen, 16-bit working registers in the programmer's model. Each of the working registers can serve as a data, address or address offset register. The 16th working register (W15) operates as a software Stack Pointer (SP) for interrupts and calls.

There are two classes of instruction in the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices: MCU and DSP. These two instruction classes are seamlessly integrated into a single CPU. The instruction set includes many addressing modes and is designed for optimum C compiler efficiency. For most instructions, the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 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 $A + B = C$ operations to be executed in a single cycle.

A block diagram of the CPU is shown in Figure 3-1, and the programmer's model for the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 is shown in Figure 3-2.

3.2 Data Addressing Overview

The data space can be addressed as 32K words or 64 Kbytes and is split into two blocks, referred to as X and Y data memory. Each memory block has its own independent Address Generation Unit (AGU). The MCU class of instructions operates solely through the X memory AGU, which accesses the entire memory map as one linear data space. Certain DSP instructions operate through the X and Y AGUs to support dual operand reads, which splits the data address space into two parts. The X and Y data space boundary is device-specific.

Overhead-free circular buffers (Modulo Addressing mode) are supported in both X and Y address spaces. The Modulo Addressing removes the software boundary checking overhead for DSP algorithms. Furthermore, the X AGU circular addressing can be used with any of the MCU class of instructions. The X AGU also supports Bit-Reversed Addressing to greatly simplify input or output data reordering for radix-2 FFT algorithms.

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

3.3 DSP Engine Overview

The DSP engine features a high-speed 17-bit by 17-bit multiplier, a 40-bit ALU, two 40-bit saturating accumulators and a 40-bit bidirectional barrel shifter. The barrel shifter is capable of shifting a 40-bit value up to 16 bits right or left, in a single cycle. The DSP instructions operate seamlessly with all other instructions and have been designed for optimal real-time performance. The `MAC` instruction and other associated instructions can concurrently fetch two data operands from memory while multiplying two W registers and accumulating and optionally saturating the result in the same cycle. This instruction functionality requires that the RAM data space be split for these instructions and linear for all others. Data space partitioning is achieved in a transparent and flexible manner through dedicating certain working registers to each address space.

3.4 Special MCU Features

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 features a 17-bit by 17-bit single-cycle multiplier that is shared by both the MCU ALU and DSP engine. The multiplier can perform signed, unsigned and mixed-sign multiplication. Using a 17-bit by 17-bit multiplier for 16-bit by 16-bit multiplication not only allows you to perform mixed-sign multiplication, it also achieves accurate results for special operations, such as $(-1.0) \times (-1.0)$.

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 supports 16/16 and 32/16 divide operations, both fractional and integer. All divide instructions are iterative operations. They must be executed within a `REPEAT` loop, resulting in a total execution time of 19 instruction cycles. The divide operation can be interrupted during any of those 19 cycles without loss of data.

A 40-bit barrel shifter is used to perform up to a 16-bit left or right shift in a single cycle. The barrel shifter can be used by both MCU and DSP instructions.

FIGURE 3-1: dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, AND dsPIC33FJ128MCX02/X04 CPU CORE BLOCK DIAGRAM

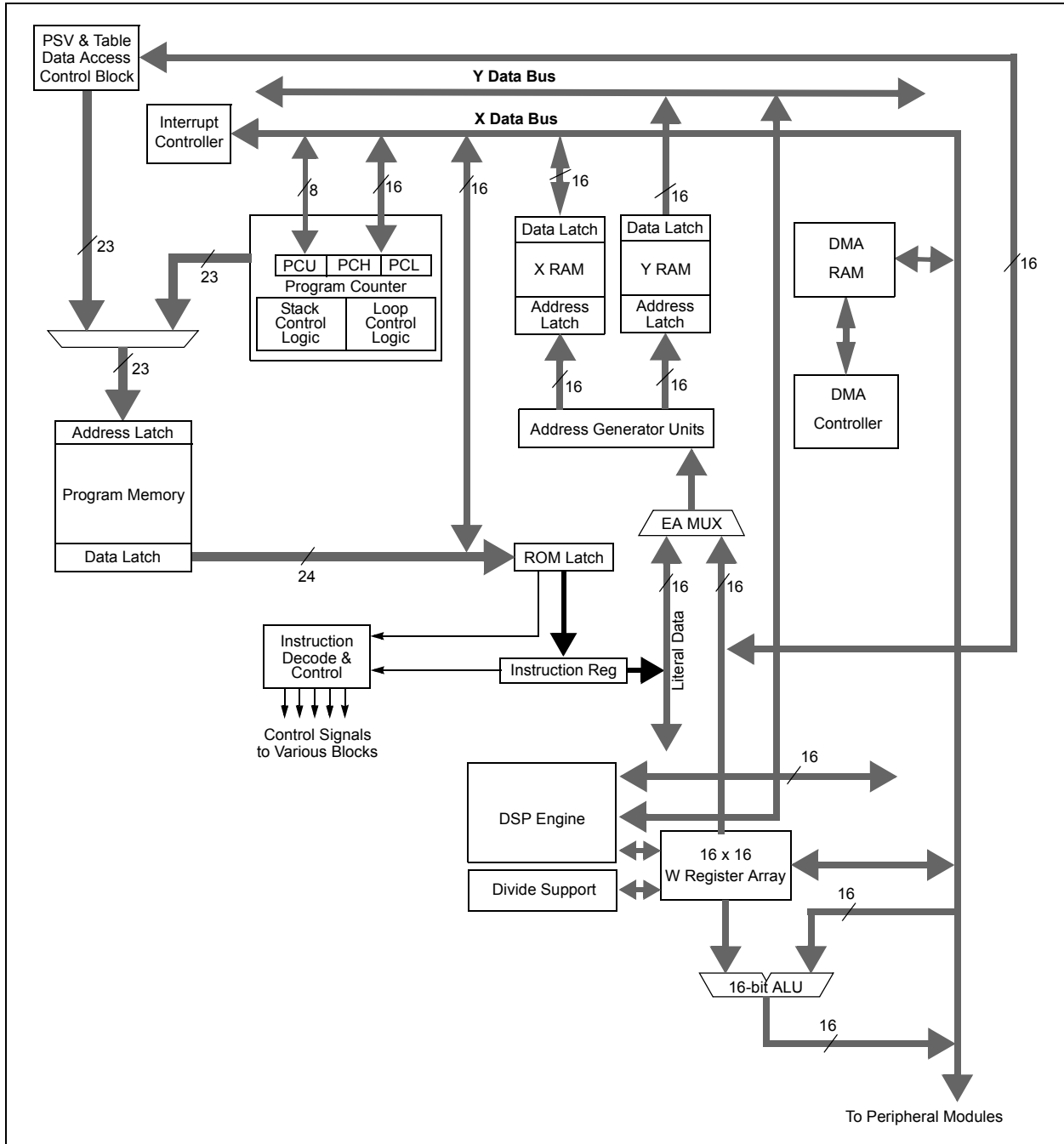
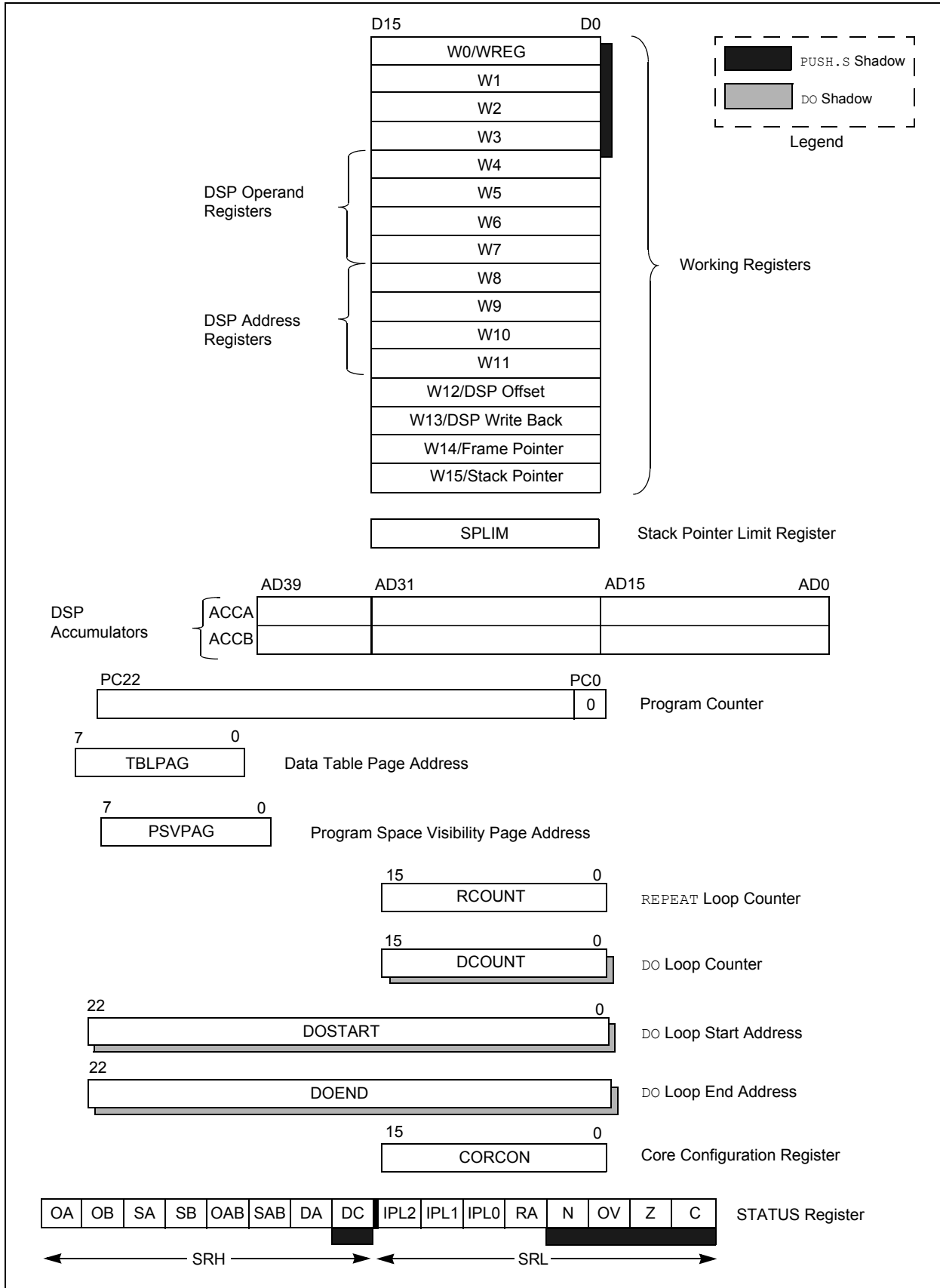


FIGURE 3-2: dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, AND dsPIC33FJ128MCX02/X04 PROGRAMMER'S MODEL



3.5 CPU Control Registers

REGISTER 3-1: SR: CPU STATUS REGISTER

| | | | | | | | |
|--------|-----|-------------------|-------------------|-----|-------|-------|-------|
| R-0 | R-0 | R/C-0 | R/C-0 | R-0 | R/C-0 | R-0 | R/W-0 |
| OA | OB | SA ⁽¹⁾ | SB ⁽¹⁾ | OAB | SAB | DA | DC |
| bit 15 | | | | | | bit 8 | |

| | | | | | | | |
|-------------------------|----------------------|----------------------|-----|-------|-------|-------|-------|
| R/W-0 ⁽³⁾ | R/W-0 ⁽³⁾ | R/W-0 ⁽³⁾ | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| IPL<2:0> ⁽²⁾ | | | RA | N | OV | Z | C |
| bit 7 | | | | | | bit 0 | |

Legend:

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

- bit 15 **OA:** Accumulator A Overflow Status bit
 1 = Accumulator A overflowed
 0 = Accumulator A has not overflowed
- bit 14 **OB:** Accumulator B Overflow Status bit
 1 = Accumulator B overflowed
 0 = Accumulator B has not overflowed
- bit 13 **SA:** Accumulator A Saturation 'Sticky' Status bit⁽¹⁾
 1 = Accumulator A is saturated or has been saturated at some time
 0 = Accumulator A is not saturated
- bit 12 **SB:** Accumulator B Saturation 'Sticky' Status bit⁽¹⁾
 1 = Accumulator B is saturated or has been saturated at some time
 0 = Accumulator B is not saturated
- bit 11 **OAB:** OA || OB Combined Accumulator Overflow Status bit
 1 = Accumulators A or B have overflowed
 0 = Neither Accumulators A or B have overflowed
- bit 10 **SAB:** SA || SB Combined Accumulator (Sticky) Status bit⁽⁴⁾
 1 = Accumulators A or B are saturated or have been saturated at some time in the past
 0 = Neither Accumulator A or B are saturated
- bit 9 **DA:** DO Loop Active bit
 1 = DO loop in progress
 0 = DO loop not in progress
- bit 8 **DC:** MCU 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 low-order bit (for byte-sized data) or 8th low-order bit (for word-sized data) of the result occurred

Note 1: This bit can be read or cleared (not set).

2: The IPL<2:0> bits are concatenated with the IPL<3> bit (CORCON<3>) to form the CPU Interrupt Priority Level. The value in parentheses indicates the IPL if IPL<3> = 1. User interrupts are disabled when IPL<3> = 1.

3: The IPL<2:0> Status bits are read only when NSTDIS = 1 (INTCON1<15>).

4: This bit can be read or cleared (not set). Clearing this bit clears SA and SB.

REGISTER 3-1: SR: CPU STATUS REGISTER (CONTINUED)

- bit 7-5 **IPL<2:0>**: CPU Interrupt Priority Level Status bits⁽²⁾
111 = CPU Interrupt Priority Level is 7 (15), user interrupts disabled
110 = CPU Interrupt Priority Level is 6 (14)
101 = CPU Interrupt Priority Level is 5 (13)
100 = CPU Interrupt Priority Level is 4 (12)
011 = CPU Interrupt Priority Level is 3 (11)
010 = CPU Interrupt Priority Level is 2 (10)
001 = CPU Interrupt Priority Level is 1 (9)
000 = CPU Interrupt Priority Level is 0 (8)
- bit 4 **RA**: REPEAT Loop Active bit
1 = REPEAT loop in progress
0 = REPEAT loop not in progress
- bit 3 **N**: MCU ALU Negative bit
1 = Result was negative
0 = Result was non-negative (zero or positive)
- bit 2 **OV**: MCU ALU Overflow bit
This bit is used for signed arithmetic (two's complement). It indicates an overflow of a magnitude that causes the sign bit to change state.
1 = Overflow occurred for signed arithmetic (in this arithmetic operation)
0 = No overflow occurred
- bit 1 **Z**: MCU ALU Zero bit
1 = An operation that affects the Z bit has set it at some time in the past
0 = The most recent operation that affects the Z bit has cleared it (i.e., a non-zero result)
- bit 0 **C**: MCU 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: This bit can be read or cleared (not set).

2: The IPL<2:0> bits are concatenated with the IPL<3> bit (CORCON<3>) to form the CPU Interrupt Priority Level. The value in parentheses indicates the IPL if IPL<3> = 1. User interrupts are disabled when IPL<3> = 1.

3: The IPL<2:0> Status bits are read only when NSTDIS = 1 (INTCON1<15>).

4: This bit can be read or cleared (not set). Clearing this bit clears SA and SB.

REGISTER 3-2: CORCON: CORE CONTROL REGISTER

| | | | | | | | |
|--------|-----|-----|-------|--------------------|---------|-----|-------|
| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R-0 | R-0 | R-0 |
| — | — | — | US | EDT ⁽¹⁾ | DL<2:0> | | |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------|-------|-------|--------|---------------------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-1 | R/W-0 | R/C-0 | R/W-0 | R/W-0 | R/W-0 |
| SATA | SATB | SATDW | ACCSAT | IPL3 ⁽²⁾ | PSV | RND | IF |
| bit 7 | | | | | | | bit 0 |

| | | | |
|---------------------|----------------------|------------------------------------|------------------|
| Legend: | C = Clear only 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 15-13 **Unimplemented:** Read as '0'
- bit 12 **US:** DSP Multiply Unsigned/Signed Control bit
 1 = DSP engine multiplies are unsigned
 0 = DSP engine multiplies are signed
- bit 11 **EDT:** Early DO Loop Termination Control bit⁽¹⁾
 1 = Terminate executing DO loop at end of current loop iteration
 0 = No effect
- bit 10-8 **DL<2:0>:** DO Loop Nesting Level Status bits
 111 = 7 DO loops active
 •
 •
 •
 001 = 1 DO loop active
 000 = 0 DO loops active
- bit 7 **SATA:** ACCA Saturation Enable bit
 1 = Accumulator A saturation enabled
 0 = Accumulator A saturation disabled
- bit 6 **SATB:** ACCB Saturation Enable bit
 1 = Accumulator B saturation enabled
 0 = Accumulator B saturation disabled
- bit 5 **SATDW:** Data Space Write from DSP Engine Saturation Enable bit
 1 = Data space write saturation enabled
 0 = Data space write saturation disabled
- bit 4 **ACCSAT:** Accumulator Saturation Mode Select bit
 1 = 9.31 saturation (super saturation)
 0 = 1.31 saturation (normal saturation)
- bit 3 **IPL3:** CPU Interrupt Priority Level Status bit 3⁽²⁾
 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

Note 1: This bit is always read as '0'.

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

REGISTER 3-2: CORCON: CORE CONTROL REGISTER (CONTINUED)

- bit 1 **RND:** Rounding Mode Select bit
 1 = Biased (conventional) rounding enabled
 0 = Unbiased (convergent) rounding enabled
- bit 0 **IF:** Integer or Fractional Multiplier Mode Select bit
 1 = Integer mode enabled for DSP multiply ops
 0 = Fractional mode enabled for DSP multiply ops

Note 1: This bit is always read as '0'.

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

3.6 Arithmetic Logic Unit (ALU)

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 ALU is 16 bits wide and is capable of addition, subtraction, bit shifts and logic operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. Depending on the operation, the ALU can 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.

Refer to the “*dsPIC30F/33F Programmer's Reference Manual*” (DS70157) for information on the SR bits affected by each instruction.

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 CPU incorporates hardware support for both multiplication and division. This includes a dedicated hardware multiplier and support hardware for 16-bit-divisor division.

3.6.1 MULTIPLIER

Using the high-speed 17-bit x 17-bit multiplier of the DSP engine, the ALU supports unsigned, signed or mixed-sign operation in several MCU multiplication modes:

- 16-bit x 16-bit signed
- 16-bit x 16-bit unsigned
- 16-bit signed x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit unsigned
- 16-bit unsigned x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit signed
- 8-bit unsigned x 8-bit unsigned

3.6.2 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. 16-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.

3.7 DSP Engine

The DSP engine consists of a high-speed 17-bit x 17-bit multiplier, a barrel shifter and a 40-bit adder/subtractor (with two target accumulators, round and saturation logic).

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 is a single-cycle instruction flow architecture; therefore, concurrent operation of the DSP engine with MCU instruction flow is not possible. However, some MCU ALU and DSP engine resources can be used concurrently by the same instruction (e.g., ED, EDAC).

The DSP engine can also perform inherent accumulator-to-accumulator operations that require no additional data. These instructions are `ADD`, `SUB` and `NEG`.

The DSP engine has options selected through bits in the CPU Core Control register (CORCON), as listed below:

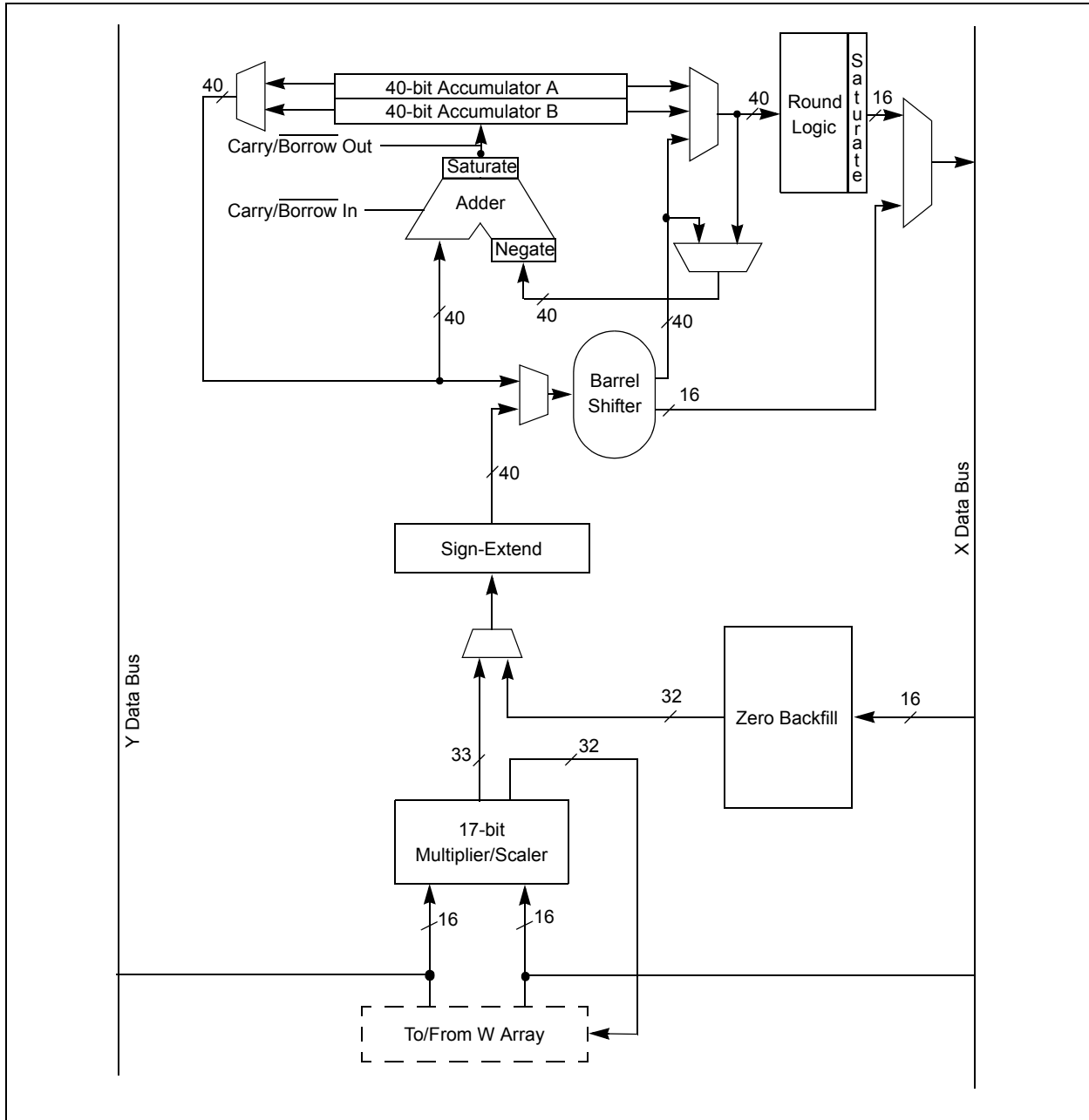
- Fractional or integer DSP multiply (IF)
- Signed or unsigned DSP multiply (US)
- Conventional or convergent rounding (RND)
- Automatic saturation on/off for ACCA (SATA)
- Automatic saturation on/off for ACCB (SATB)
- Automatic saturation on/off for writes to data memory (SATDW)
- Accumulator Saturation mode selection (ACCSAT)

A block diagram of the DSP engine is shown in Figure 3-3.

TABLE 3-1: DSP INSTRUCTIONS SUMMARY

| Instruction | Algebraic Operation | ACC Write Back |
|-------------|-----------------------|----------------|
| CLR | $A = 0$ | Yes |
| ED | $A = (x - y)2$ | No |
| EDAC | $A = A + (x - y)2$ | No |
| MAC | $A = A + (x \cdot y)$ | Yes |
| MAC | $A = A + x2$ | No |
| MOVSAC | No change in A | Yes |
| MPY | $A = x \cdot y$ | No |
| MPY | $A = x2$ | No |
| MPY.N | $A = -x \cdot y$ | No |
| MSC | $A = A - x \cdot y$ | Yes |

FIGURE 3-3: DSP ENGINE BLOCK DIAGRAM



3.7.1 MULTIPLIER

The 17-bit x 17-bit multiplier is capable of signed or unsigned operation and can multiplex its output using a scaler to support either 1.31 fractional (Q31) or 32-bit integer results. Unsigned operands are zero-extended into the 17th bit of the multiplier input value. Signed operands are sign-extended into the 17th bit of the multiplier input value. The output of the 17-bit x 17-bit multiplier/scaler is a 33-bit value that is sign-extended to 40 bits. Integer data is inherently represented as a signed two's complement value, where the Most Significant bit (MSb) is defined as a sign bit. The range of an N-bit two's complement integer is -2^{N-1} to $2^{N-1} - 1$.

- For a 16-bit integer, the data range is -32768 (0x8000) to 32767 (0x7FFF) including 0.
- For a 32-bit integer, the data range is -2,147,483,648 (0x8000 0000) to 2,147,483,647 (0x7FFF FFFF).

When the multiplier is configured for fractional multiplication, the data is represented as a two's complement fraction, where the MSb is defined as a sign bit and the radix point is implied to lie just after the sign bit (QX format). The range of an N-bit two's complement fraction with this implied radix point is -1.0 to $(1 - 2^{1-N})$. For a 16-bit fraction, the Q15 data range is -1.0 (0x8000) to 0.999969482 (0x7FFF) including 0 and has a precision of 3.01518×10^{-5} . In Fractional mode, the 16 x 16 multiply operation generates a 1.31 product that has a precision of 4.65661×10^{-10} .

The same multiplier is used to support the MCU multiply instructions, which include integer 16-bit signed, unsigned and mixed sign multiply operations.

The `MUL` instruction can be directed to use byte or word-sized operands. Byte operands direct a 16-bit result, and word operands direct a 32-bit result to the specified registers in the W array.

3.7.2 DATA ACCUMULATORS AND ADDER/SUBTRACTOR

The data accumulator consists of a 40-bit adder/subtractor with automatic sign extension logic. It can select one of two accumulators (A or B) as its pre-accumulation source and post-accumulation destination. For the `ADD` and `LAC` instructions, the data to be accumulated or loaded can be optionally scaled using the barrel shifter prior to accumulation.

3.7.2.1 Adder/Subtractor, Overflow and Saturation

The adder/subtractor is a 40-bit adder with an optional zero input into one side, and either true or complement data into the other input.

- In the case of addition, the Carry/Borrow input is active-high and the other input is true data (not complemented).
- In the case of subtraction, the Carry/Borrow input is active-low and the other input is complemented.

The adder/subtractor generates Overflow Status bits, SA/SB and OA/OB, which are latched and reflected in the STATUS register:

- Overflow from bit 39: this is a catastrophic overflow in which the sign of the accumulator is destroyed.
- Overflow into guard bits 32 through 39: this is a recoverable overflow. This bit is set whenever all the guard bits are not identical to each other.

The adder has an additional saturation block that controls accumulator data saturation, if selected. It uses the result of the adder, the Overflow Status bits described previously and the `SAT<A:B>` (`CORCON<7:6>`) and `ACCSAT` (`CORCON<4>`) mode control bits to determine when and to what value to saturate.

Six STATUS register bits support saturation and overflow:

- OA: ACCA overflowed into guard bits
- OB: ACCB overflowed into guard bits
- SA: ACCA saturated (bit 31 overflow and saturation)
or
ACCA overflowed into guard bits and saturated (bit 39 overflow and saturation)
- SB: ACCB saturated (bit 31 overflow and saturation)
or
ACCB overflowed into guard bits and saturated (bit 39 overflow and saturation)
- OAB: Logical OR of OA and OB
- SAB: Logical OR of SA and SB

The OA and OB bits are modified each time data passes through the adder/subtractor. When set, they indicate that the most recent operation has overflowed into the accumulator guard bits (bits 32 through 39). The OA and OB bits can also optionally generate an arithmetic warning trap when set and the corresponding Overflow Trap Flag Enable bits (`OVATE`, `OVBTE`) in the `INTCON1` register are set (refer to **Section 7.0 "Interrupt Controller"**). This allows the user application to take immediate action, for example, to correct system gain.

The SA and SB bits are modified each time data passes through the adder/subtractor, but can only be cleared by the user application. When set, they indicate that the accumulator has overflowed its maximum range (bit 31 for 32-bit saturation or bit 39 for 40-bit saturation) and is saturated (if saturation is enabled). When saturation is not enabled, SA and SB default to bit 39 overflow and thus indicate that a catastrophic overflow has occurred. If the `COVTE` bit in the `INTCON1` register is set, the SA and SB bits generate an arithmetic warning trap when saturation is disabled.

The Overflow and Saturation Status bits can optionally be viewed in the STATUS Register (SR) as the logical OR of OA and OB (in bit OAB) and the logical OR of SA and SB (in bit SAB). Programmers can check one bit in the STATUS register to determine if either accumulator has overflowed, or one bit to determine if either accumulator has saturated. This is useful for complex number arithmetic, which typically uses both accumulators.

The device supports three Saturation and Overflow modes:

- **Bit 39 Overflow and Saturation:**
When bit 39 overflow and saturation occurs, the saturation logic loads the maximally positive 9.31 (0x7FFFFFFF) or maximally negative 9.31 value (0x80000000) into the target accumulator. The SA or SB bit is set and remains set until cleared by the user application. This condition is referred to as 'super saturation' and provides protection against erroneous data or unexpected algorithm problems (such as gain calculations).
- **Bit 31 Overflow and Saturation:**
When bit 31 overflow and saturation occurs, the saturation logic then loads the maximally positive 1.31 value (0x007FFFFFFF) or maximally negative 1.31 value (0x00800000) into the target accumulator. The SA or SB bit is set and remains set until cleared by the user application. When this Saturation mode is in effect, the guard bits are not used, so the OA, OB or OAB bits are never set.
- **Bit 39 Catastrophic Overflow:**
The bit 39 Overflow Status bit from the adder is used to set the SA or SB bit, which remains set until cleared by the user application. No saturation operation is performed, and the accumulator is allowed to overflow, destroying its sign. If the COVTE bit in the INTCON1 register is set, a catastrophic overflow can initiate a trap exception.

3.7.3 ACCUMULATOR 'WRITE BACK'

The MAC class of instructions (with the exception of MPY, MPY.N, ED and EDAC) can optionally write a rounded version of the high word (bits 31 through 16) of the accumulator that is not targeted by the instruction into data space memory. The write is performed across the X bus into combined X and Y address space. The following addressing modes are supported:

- **W13, Register Direct:**
The rounded contents of the non-target accumulator are written into W13 as a 1.15 fraction.
- **[W13] += 2, Register Indirect with Post-Increment:**
The rounded contents of the non-target accumulator are written into the address pointed to by W13 as a 1.15 fraction. W13 is then incremented by 2 (for a word write).

3.7.3.1 Round Logic

The round logic is a combinational block that performs a conventional (biased) or convergent (unbiased) round function during an accumulator write (store). The Round mode is determined by the state of the RND bit in the CORCON register. It generates a 16-bit, 1.15 data value that is passed to the data space write saturation logic. If rounding is not indicated by the instruction, a truncated 1.15 data value is stored and the least significant word is simply discarded.

Conventional rounding zero-extends bit 15 of the accumulator and adds it to the ACCxH word (bits 16 through 31 of the accumulator).

- If the ACCxL word (bits 0 through 15 of the accumulator) is between 0x8000 and 0xFFFF (0x8000 included), ACCxH is incremented.
- If ACCxL is between 0x0000 and 0x7FFF, ACCxH is left unchanged.

A consequence of this algorithm is that over a succession of random rounding operations, the value tends to be biased slightly positive.

Convergent (or unbiased) rounding operates in the same manner as conventional rounding, except when ACCxL equals 0x8000. In this case, the Least Significant bit (bit 16 of the accumulator) of ACCxH is examined:

- If it is '1', ACCxH is incremented.
- If it is '0', ACCxH is not modified.

Assuming that bit 16 is effectively random in nature, this scheme removes any rounding bias that may accumulate.

The SAC and SAC.R instructions store either a truncated (SAC), or rounded (SAC.R) version of the contents of the target accumulator to data memory via the X bus, subject to data saturation (see **Section 3.7.3.2 "Data Space Write Saturation"**). For the MAC class of instructions, the accumulator write-back operation functions in the same manner, addressing combined MCU (X and Y) data space through the X bus. For this class of instructions, the data is always subject to rounding.

3.7.3.2 Data Space Write Saturation

In addition to adder/subtractor saturation, writes to data space can also be saturated, but without affecting the contents of the source accumulator. The data space write saturation logic block accepts a 16-bit, 1.15 fractional value from the round logic block as its input, together with overflow status from the original source (accumulator) and the 16-bit round adder. These inputs are combined and used to select the appropriate 1.15 fractional value as output to write to data space memory.

If the SATDW bit in the CORCON register is set, data (after rounding or truncation) is tested for overflow and adjusted accordingly:

- For input data greater than 0x007FFF, data written to memory is forced to the maximum positive 1.15 value, 0x7FFF.
- For input data less than 0xFF8000, data written to memory is forced to the maximum negative 1.15 value, 0x8000.

The Most Significant bit of the source (bit 39) is used to determine the sign of the operand being tested.

If the SATDW bit in the CORCON register is not set, the input data is always passed through unmodified under all conditions.

3.7.4 BARREL SHIFTER

The barrel shifter can perform up to 16-bit arithmetic or logic right shifts, or up to 16-bit left shifts in a single cycle. The source can be either of the two DSP accumulators or the X bus (to support multi-bit shifts of register or memory data).

The shifter requires a signed binary value to determine both the magnitude (number of bits) and direction of the shift operation. A positive value shifts the operand right. A negative value shifts the operand left. A value of '0' does not modify the operand.

The barrel shifter is 40 bits wide, thereby obtaining a 40-bit result for DSP shift operations and a 16-bit result for MCU shift operations. Data from the X bus is presented to the barrel shifter between bit positions 16 and 31 for right shifts, and between bit positions 0 and 16 for left shifts.

4.0 MEMORY ORGANIZATION

Note: This data sheet summarizes the features of dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 4. “Program Memory” (DS70203), which is available from the Microchip website (www.microchip.com).

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 architecture features separate program and data memory spaces and buses. This architecture also allows the direct access of program memory from the data space during code execution.

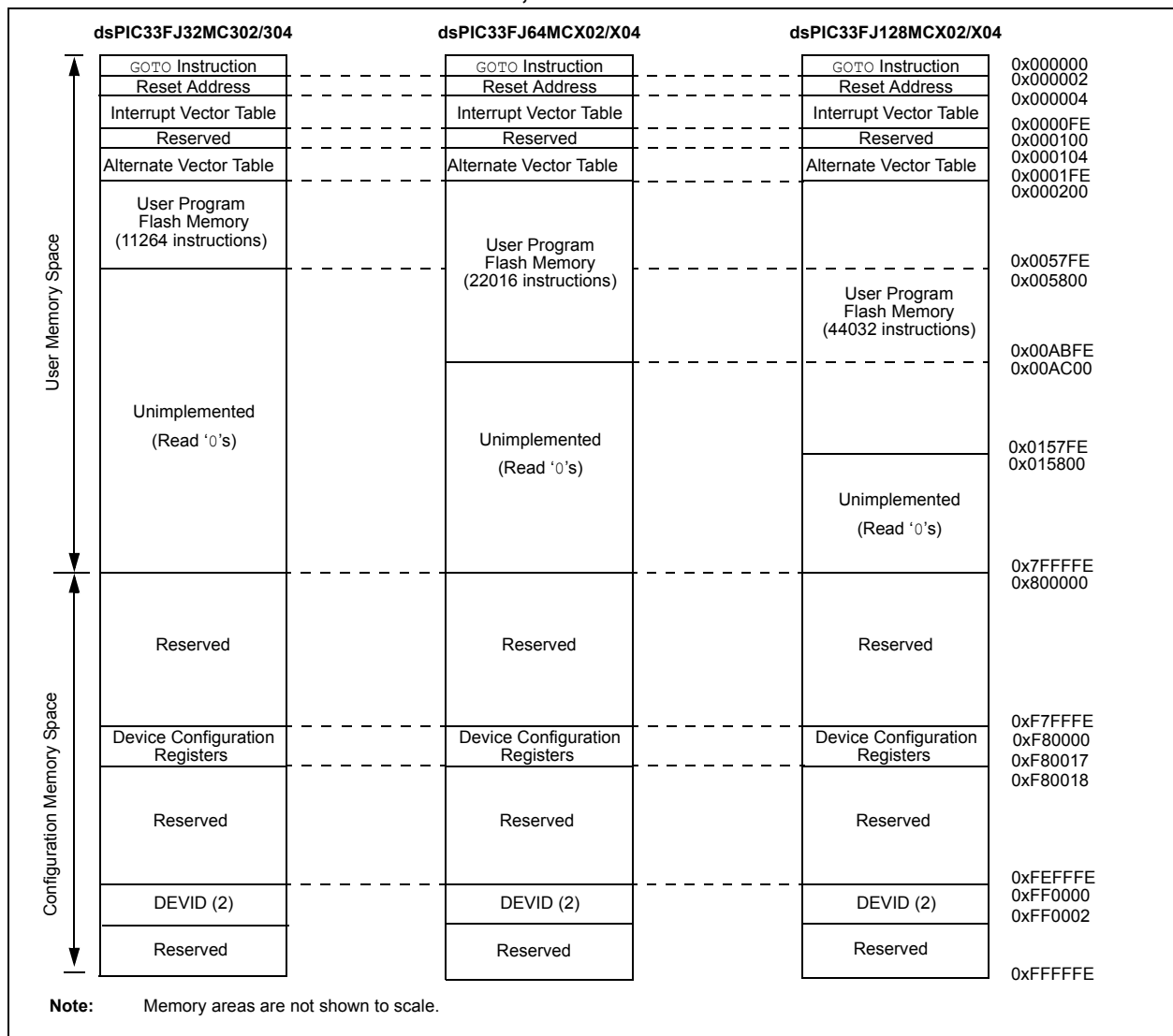
4.1 Program Address Space

The program address memory space of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices is 4M instructions. The space is addressable by a 24-bit value derived either from the 23-bit Program Counter (PC) during program execution, or from table operation or data space remapping as described in Section 4.6 “Interfacing Program and Data Memory Spaces”.

User application access to the program memory space is restricted to the lower half of the address range (0x000000 to 0x7FFFFFFF). 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.

The memory map for the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices is shown in Figure 4-1.

FIGURE 4-1: PROGRAM MEMORY MAP FOR dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, AND dsPIC33FJ128MCX02/X04 DEVICES



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 provides compatibility with data memory space addressing and makes data in the program memory space accessible.

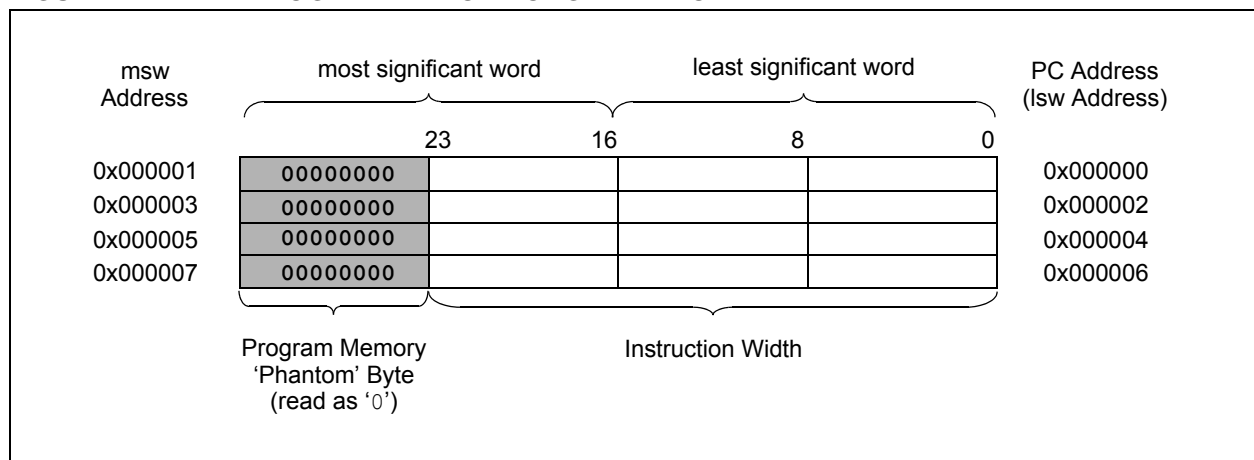
addresses between 0x00000 and 0x000200 for hard-coded program execution vectors. A hardware Reset vector is provided to redirect code execution from the default value of the PC on device Reset to the actual start of code. A GOTO instruction is programmed by the user application at 0x000000, with the actual address for the start of code at 0x000002.

dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices also have two interrupt vector tables, located from 0x000004 to 0x0000FF and 0x000100 to 0x0001FF. These vector tables allow each of the device interrupt sources to be handled by separate Interrupt Service Routines (ISRs). A more detailed discussion of the interrupt vector tables is provided in **Section 7.1 “Interrupt Vector Table”**.

4.1.2 INTERRUPT AND TRAP VECTORS

All dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices reserve the

FIGURE 4-2: PROGRAM MEMORY ORGANIZATION



4.2 Data Address Space

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 CPU has a separate 16-bit-wide data memory space. The data space is accessed using separate Address Generation Units (AGUs) for read and write operations. The data memory maps is shown in Figure 4-4.

All Effective Addresses (EAs) in the data memory space are 16 bits wide and point to bytes within the data space. This arrangement 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 4.6.3 “Reading Data From Program Memory Using Program Space Visibility”**).

dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices implement up to 16 Kbytes of data memory. Should an EA point to a location outside of this area, an all-zero word or byte is returned.

4.2.1 DATA SPACE WIDTH

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

4.2.2 DATA MEMORY ORGANIZATION AND ALIGNMENT

To maintain backward compatibility with PIC[®] MCU devices and improve data space memory usage efficiency, the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 instruction set supports both word and byte operations. As a consequence of byte accessibility, all effective address calculations are internally scaled to step through word-aligned memory. For example, the core recognizes that Post-Modified Register Indirect Addressing mode [$Ws++$] results in a value of $Ws + 1$ for byte operations and $Ws + 2$ for word operations.

A data byte read, reads the complete word that 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 that 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 is generated. If the error occurred on a read, the instruction underway is completed. If the error occurred on a write, the instruction is executed but the write does not occur. In either case, a trap is then executed, allowing the system and/or user application 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 user applications to translate 8-bit signed data to 16-bit signed values. Alternatively, for 16-bit unsigned data, user applications can clear the MSB of any W register by executing a zero-extend (ZE) instruction on the appropriate address.

4.2.3 SFR SPACE

The first 2 Kbytes of the Near Data Space, from 0x0000 to 0x07FF, is primarily occupied by Special Function Registers (SFRs). These are used by the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 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'.

| |
|--|
| Note: The actual set of peripheral features and interrupts varies by the device. Refer to the corresponding device tables and pin-out diagrams for device-specific information. |
|--|

4.2.4 NEAR DATA SPACE

The 8 Kbyte area between 0x0000 and 0x1FFF 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. Additionally, the whole data space is addressable using MOV instructions, which support Memory Direct Addressing mode with a 16-bit address field, or by using Indirect Addressing mode using a working register as an address pointer.

FIGURE 4-3: DATA MEMORY MAP FOR dsPIC33FJ32MC302/304 DEVICES WITH 4 KB RAM

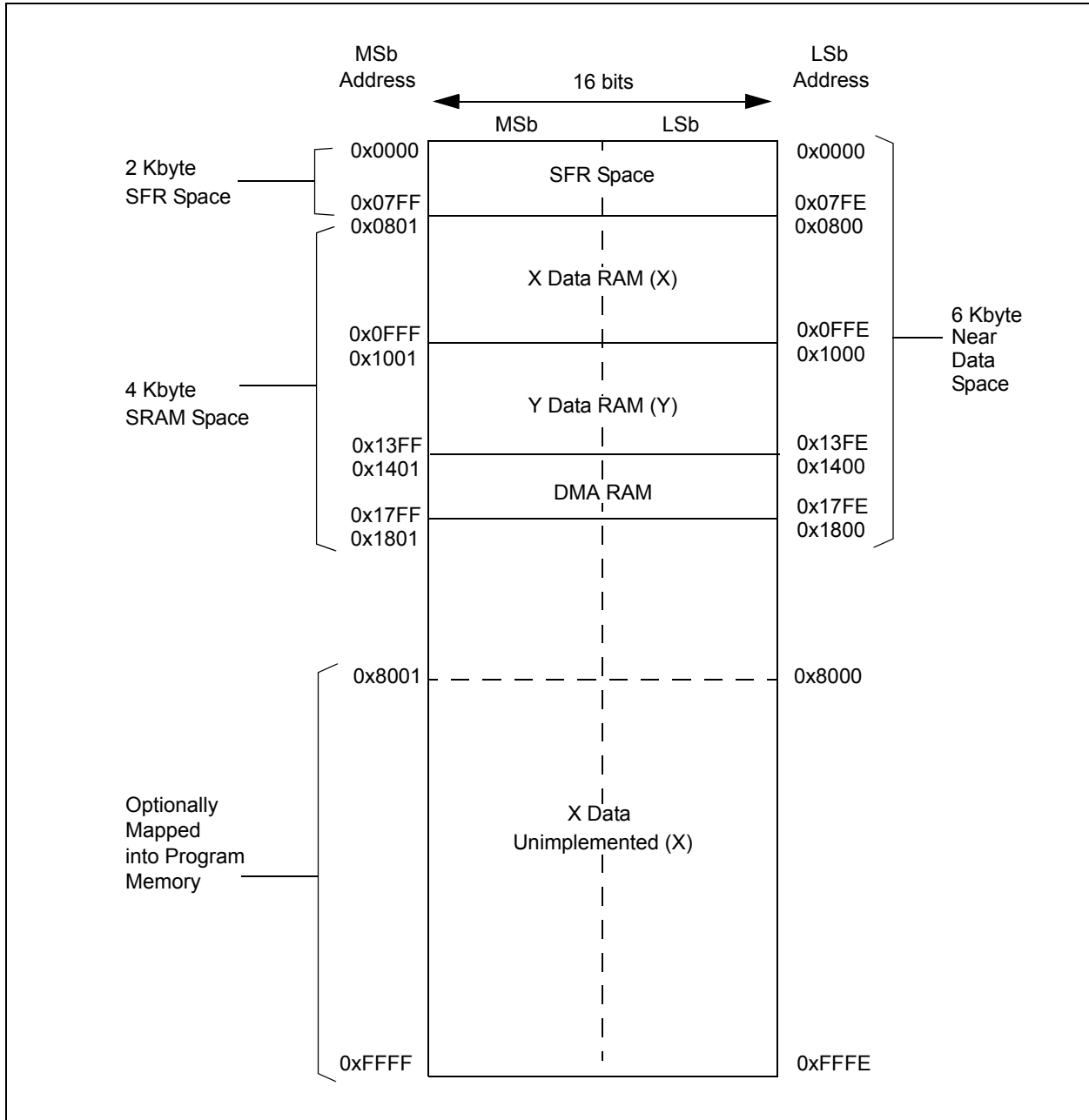


FIGURE 4-4: DATA MEMORY MAP FOR dsPIC33FJ128MC202/204 AND dsPIC33FJ64MC202/204 DEVICES WITH 8 KB RAM

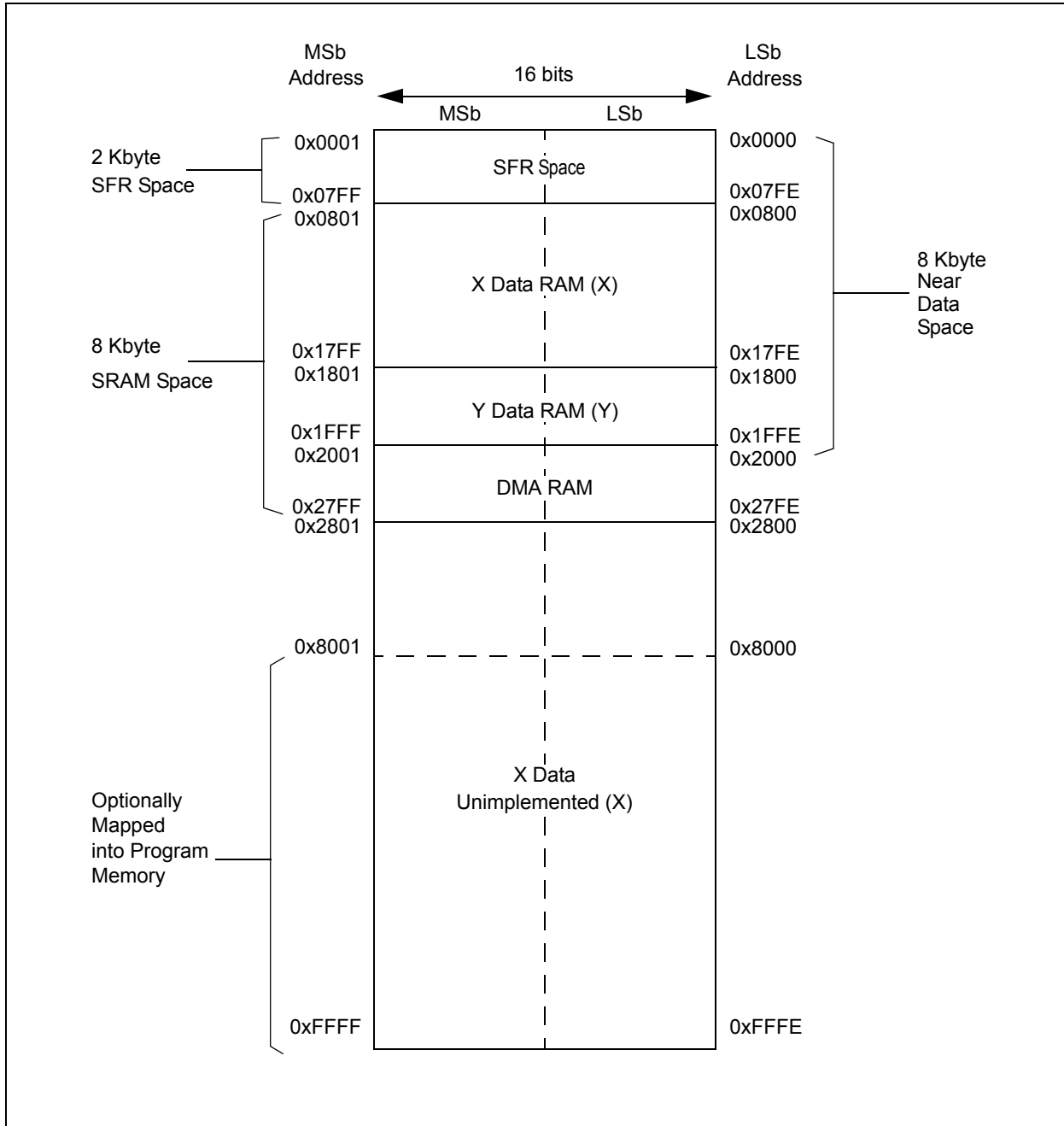
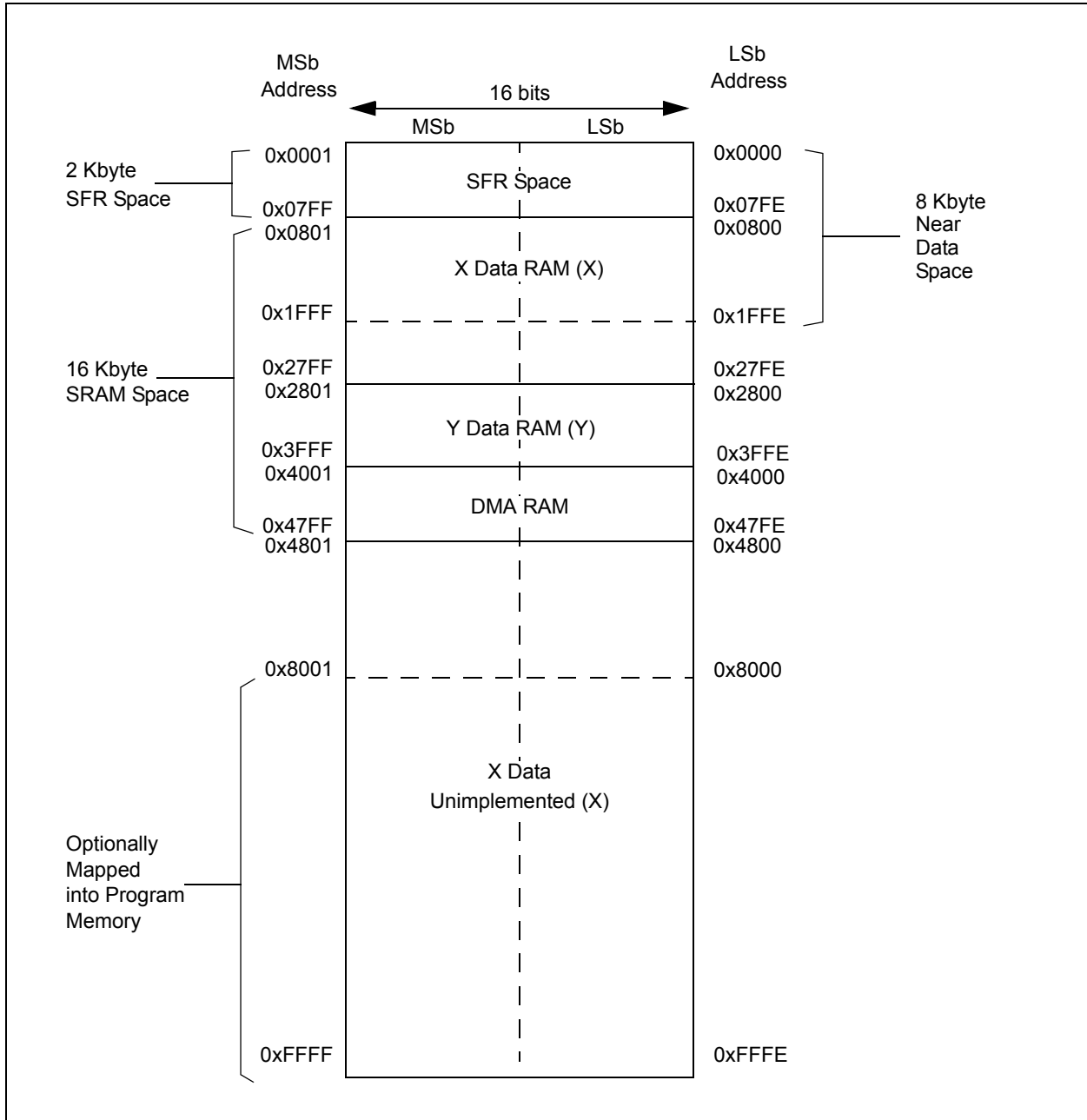


FIGURE 4-5: DATA MEMORY MAP FOR dsPIC33FJ128MC802/804 AND dsPIC33FJ64MC802/804 DEVICES WITH 16 KB RAM



4.2.5 X AND Y DATA SPACES

The core has two data spaces, X and Y. These data spaces can be considered either separate (for some DSP instructions), or as one unified linear address range (for MCU instructions). The data spaces are accessed using two Address Generation Units (AGUs) and separate data paths. This feature allows certain instructions to concurrently fetch two words from RAM, thereby enabling efficient execution of DSP algorithms such as Finite Impulse Response (FIR) filtering and Fast Fourier Transform (FFT).

The X data space is used by all instructions and supports all addressing modes. X data space has separate read and write data buses. The X read data bus is the read data path for all instructions that view data space as combined X and Y address space. It is also the X data prefetch path for the dual operand DSP instructions (MAC class).

The Y data space is used in concert with the X data space by the MAC class of instructions (CLR, ED, EDAC, MAC, MOV SAC, MPY, MPY.N and MSC) to provide two concurrent data read paths.

Both the X and Y data spaces support Modulo Addressing mode for all instructions, subject to addressing mode restrictions. Bit-Reversed Addressing mode is only supported for writes to X data space.

All data memory writes, including in DSP instructions, view data space as combined X and Y address space. The boundary between the X and Y data spaces is device-dependent and is not user-programmable.

All effective addresses are 16 bits wide and point to bytes within the data space. Therefore, the data space address range is 64 Kbytes, or 32K words, though the implemented memory locations vary by device.

4.2.6 DMA RAM

Every dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 device contains up to 2 Kbytes of dual ported DMA RAM located at the end of Y data space, and is a part of Y data space. Memory locations in the DMA RAM space are accessible simultaneously by the CPU and the DMA controller module. DMA RAM is utilized by the DMA controller to store data to be transferred to various peripherals using DMA, as well as data transferred from various peripherals using DMA. The DMA RAM can be accessed by the DMA controller without having to steal cycles from the CPU.

When the CPU and the DMA controller attempt to concurrently write to the same DMA RAM location, the hardware ensures that the CPU is given precedence in accessing the DMA RAM location. Therefore, the DMA RAM provides a reliable means of transferring DMA data without ever having to stall the CPU.

| |
|---|
| <p>Note: DMA RAM can be used for general purpose data storage if the DMA function is not required in an application.</p> |
|---|

TABLE 4-1: CPU CORE REGISTERS MAP

| SFR Name | SFR 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 Register | | | | | | | | | | | | | | | | xxxx | |
| ACCAL | 0022 | ACCAL | | | | | | | | | | | | | | | | xxxx | |
| ACCAH | 0024 | ACCAH | | | | | | | | | | | | | | | | xxxx | |
| ACCAU | 0026 | ACCA<39> | | | | | | | | ACCAU | | | | | | | | xxxx | |
| ACCBL | 0028 | ACCBL | | | | | | | | | | | | | | | | xxxx | |
| ACCBH | 002A | ACCBH | | | | | | | | | | | | | | | | xxxx | |
| ACCBU | 002C | ACCB<39> | | | | | | | | ACCBU | | | | | | | | xxxx | |
| PCL | 002E | Program Counter Low Word Register | | | | | | | | | | | | | | | | xxxx | |
| PCH | 0030 | — | — | — | — | — | — | — | — | Program Counter High Byte Register | | | | | | | | 0000 | |
| TBLPAG | 0032 | — | — | — | — | — | — | — | — | Table Page Address Pointer Register | | | | | | | | 0000 | |
| PSVPAG | 0034 | — | — | — | — | — | — | — | — | Program Memory Visibility Page Address Pointer Register | | | | | | | | 0000 | |
| RCOUNT | 0036 | Repeat Loop Counter Register | | | | | | | | | | | | | | | | xxxx | |
| DCOUNT | 0038 | DCOUNT<15:0> | | | | | | | | | | | | | | | | xxxx | |
| DOSTARTL | 003A | DOSTARTL<15:1> | | | | | | | | | | | | | | | | 0 | xxxx |
| DOSTARTH | 003C | — | — | — | — | — | — | — | — | — | — | DOSTARTH<5:0> | | | | | 00xx | | |
| DOENDL | 003E | DOENDL<15:1> | | | | | | | | | | | | | | | | 0 | xxxx |
| DOENDH | 0040 | — | — | — | — | — | — | — | — | — | DOENDH | | | | | | | 00xx | |
| SR | 0042 | OA | OB | SA | SB | OAB | SAB | DA | DC | IPL2 | IPL1 | IPL0 | RA | N | OV | Z | C | 0000 | |
| CORCON | 0044 | — | — | — | US | EDT | DL<2:0> | | | SATA | SATB | SATDW | ACCSAT | IPL3 | PSV | RND | IF | 0020 | |
| MODCON | 0046 | XMODEN | YMODEN | — | — | BWM<3:0> | | | | | YWM<3:0> | | | XWM<3:0> | | | 0000 | | |

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

TABLE 4-1: CPU CORE REGISTERS MAP (CONTINUED)

| SFR Name | SFR 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 | |
|----------|----------|----------|--------|-------------------------------------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|------------|------|
| XMODSRT | 0048 | XS<15:1> | | | | | | | | | | | | | | | | 0 | xxxx |
| XMODEND | 004A | XE<15:1> | | | | | | | | | | | | | | | | 1 | xxxx |
| YMODSRT | 004C | YS<15:1> | | | | | | | | | | | | | | | | 0 | xxxx |
| YMODEND | 004E | YE<15:1> | | | | | | | | | | | | | | | | 1 | xxxx |
| XBREV | 0050 | BREN | | | | | | | | | | | | | | | XB<14:0> | | xxxx |
| DISICNT | 0052 | — | — | Disable Interrupts Counter Register | | | | | | | | | | | | | | xxxx | |

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

TABLE 4-2: CHANGE NOTIFICATION REGISTER MAP FOR dsPIC33FJ128MC202/802, dsPIC33FJ64MC202/802 AND dsPIC33FJ32MC302

| SFR Name | SFR 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 | — | — | — | CN7IE | CN6IE | CN5IE | CN4IE | CN3IE | CN2IE | CN1IE | CN0IE | 0000 |
| CNEN2 | 00C2 | — | CN30IE | CN29IE | — | CN27IE | — | — | CN24IE | CN23IE | CN22IE | CN21IE | — | — | — | — | CN16IE | 0000 |
| CNPU1 | 0068 | CN15PUE | CN14PUE | CN13PUE | CN12PUE | CN11PUE | — | — | — | CN7PUE | CN6PUE | CN5PUE | CN4PUE | CN3PUE | CN2PUE | CN1PUE | CN0PUE | 0000 |
| CNPU2 | 006A | — | CN30PUE | CN29PUE | — | CN27PUE | — | — | CN24PUE | CN23PUE | CN22PUE | CN21PUE | — | — | — | — | CN16PUE | 0000 |

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

TABLE 4-3: CHANGE NOTIFICATION REGISTER MAP FOR dsPIC33FJ128MC204/804, dsPIC33FJ64MC204/804 AND dsPIC33FJ32MC304

| SFR Name | SFR 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 | CN9IE | CN8IE | CN7IE | CN6IE | CN5IE | CN4IE | CN3IE | CN2IE | CN1IE | CN0IE | 0000 |
| CNEN2 | 00C2 | — | CN30IE | CN29IE | CN28IE | CN27IE | CN26IE | CN25IE | CN24IE | CN23IE | CN22IE | CN21IE | CN20IE | CN19IE | CN18IE | CN17IE | CN16IE | 0000 |
| CNPU1 | 0068 | CN15PUE | CN14PUE | CN13PUE | CN12PUE | CN11PUE | CN10PUE | CN9PUE | CN8PUE | CN7PUE | CN6PUE | CN5PUE | CN4PUE | CN3PUE | CN2PUE | CN1PUE | CN0PUE | 0000 |
| CNPU2 | 006A | — | CN30PUE | CN29PUE | CN28PUE | CN27PUE | CN26PUE | CN25PUE | CN24PUE | CN23PUE | CN22PUE | CN21PUE | CN20PUE | CN19PUE | CN18PUE | CN17PUE | CN16PUE | 0000 |

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

TABLE 4-4: INTERRUPT CONTROLLER REGISTER MAP

| SFR Name | SFR 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 | OVAERR | OVBERR | COVAERR | GOVBERR | OVATE | OVBTE | COVTE | SFTACERR | DIV0ERR | DMACERR | MATHERR | ADDRERR | STKERR | OSCFAIL | — | 0000 | |
| INTCON2 | 0082 | ALTIVT | DISI | — | — | — | — | — | — | — | — | — | — | — | INT2EP | INT1EP | INT0EP | 0000 | |
| IFS0 | 0084 | — | DMA1IF | AD1IF | U1TXIF | U1RXIF | SPI1IF | SPI1EIF | T3IF | T2IF | OC2IF | IC2IF | DMA0IF | T1IF | OC1IF | IC1IF | INT0IF | 0000 | |
| IFS1 | 0086 | U2TXIF | U2RXIF | INT2IF | T5IF | T4IF | OC4IF | OC3IF | DMA2IF | IC8IF | IC7IF | — | INT1IF | CNIF | CMIF | MI2C1IF | SI2C1IF | 0000 | |
| IFS2 | 0088 | — | DMA4IF | PMPIF | — | — | — | — | — | — | — | — | DMA3IF | C1IF ⁽¹⁾ | C1RXIF ⁽¹⁾ | SPI2IF | SPI2EIF | 0000 | |
| IFS3 | 008A | FLTA1IF | RTCIF | DMA5IF | — | — | QE1IF | PWM1IF | — | — | — | — | — | — | — | — | — | 0000 | |
| IFS4 | 008C | DAC1LIF ⁽²⁾ | DAC1RIF ⁽²⁾ | — | — | — | QE2IF | FLTA2IF | PWM2IF | — | — | C1TXIF ⁽¹⁾ | DMA7IF | DMA6IF | CRCIF | U2EIF | U1EIF | — | 0000 |
| IEC0 | 0094 | — | DMA1IE | AD1IE | U1TXIE | U1RXIE | SPI1IE | SPI1EIE | T3IE | T2IE | OC2IE | IC2IE | DMA0IE | T1IE | OC1IE | IC1IE | INT0IE | 0000 | |
| IEC1 | 0096 | U2TXIE | U2RXIE | INT2IE | T5IE | T4IE | OC4IE | OC3IE | DMA2IE | IC8IE | IC7IE | — | INT1IE | CNIE | CMIE | MI2C1IE | SI2C1IE | 0000 | |
| IEC2 | 0098 | — | DMA4IE | PMPIE | — | — | — | — | — | — | — | — | DMA3IE | C1IE ⁽¹⁾ | C1RXIE ⁽¹⁾ | SPI2IE | SPI2EIE | 0000 | |
| IEC3 | 009A | FLTA1IE | RTCIE | DMA5IE | — | — | QE1IE | PWM1IE | — | — | — | — | — | — | — | — | — | 0000 | |
| IEC4 | 009C | DAC1LIE ⁽²⁾ | DAC1RIE ⁽²⁾ | — | — | — | QE2IE | FLTA2IE | PWM2IE | — | — | C1TXIE ⁽¹⁾ | DMA7IE | DMA6IE | CRCIE | U2EIE | U1EIE | — | 0000 |
| IPC0 | 00A4 | — | T1IP<2:0> | | | — | OC1IP<2:0> | | | — | IC1IP<2:0> | | | — | INT0IP<2:0> | | | 4444 | |
| IPC1 | 00A6 | — | T2IP<2:0> | | | — | OC2IP<2:0> | | | — | IC2IP<2:0> | | | — | DMA0IP<2:0> | | | 4444 | |
| IPC2 | 00A8 | — | U1RXIP<2:0> | | | — | SPI1IP<2:0> | | | — | SPI1EIP<2:0> | | | — | T3IP<2:0> | | | 4444 | |
| IPC3 | 00AA | — | — | — | — | — | DMA1IP<2:0> | | | — | AD1IP<2:0> | | | — | U1TXIP<2:0> | | | 0444 | |
| IPC4 | 00AC | — | CNIP<2:0> | | | — | CMIP<2:0> | | | — | MI2C1IP<2:0> | | | — | SI2C1IP<2:0> | | | 4444 | |
| IPC5 | 00AE | — | IC8IP<2:0> | | | — | IC7IP<2:0> | | | — | — | — | — | — | INT1IP<2:0> | | | 4404 | |
| IPC6 | 00B0 | — | T4IP<2:0> | | | — | OC4IP<2:0> | | | — | OC3IP<2:0> | | | — | DMA2IP<2:0> | | | 4444 | |
| IPC7 | 00B2 | — | U2TXIP<2:0> | | | — | U2RXIP<2:0> | | | — | INT2IP<2:0> | | | — | T5IP<2:0> | | | 4444 | |
| IPC8 | 00B4 | — | C1IP<2:0> ⁽¹⁾ | | | — | C1RXIP<2:0> ⁽¹⁾ | | | — | SPI2IP<2:0> | | | — | SPI2EIP<2:0> | | | 4444 | |
| IPC9 | 00B6 | — | — | — | — | — | — | — | — | — | — | — | — | — | DMA3IP<2:0> | | | 0004 | |
| IPC11 | 00BA | — | — | — | — | — | DMA4IP<2:0> | | | — | PMPIP<2:0> | | | — | — | — | — | 0440 | |
| IPC14 | 00C0 | — | — | — | — | — | QE1IP<2:0> | | | — | PWM1IP<2:0> | | | — | — | — | — | 0440 | |
| IPC15 | 00C2 | — | FLTA1IP<2:0> | | | — | RTCIP<2:0> | | | — | DMA5IP<2:0> | | | — | — | — | — | 4440 | |
| IPC16 | 00C4 | — | CRCIP<2:0> | | | — | U2EIP<2:0> | | | — | U1EIP<2:0> | | | — | — | — | — | 4440 | |
| IPC17 | 00C6 | — | — | — | — | — | C1TXIP<2:0> ⁽¹⁾ | | | — | DMA7IP<2:0> | | | — | DMA6IP<2:0> | | | 0444 | |
| IPC18 | 00C8 | — | QE2IP<2:0> | | | — | FLTA2IP<2:0> | | | — | PWM2IP<2:0> | | | — | — | — | — | 4440 | |
| IPC19 | 00CA | — | DAC1LIP<2:0> ⁽²⁾ | | | — | DAC1RIP<2:0> ⁽²⁾ | | | — | — | — | — | — | — | — | — | 4400 | |
| INTTREG | 00E0 | — | — | — | — | ILR<3:0>> | | | — | VECNUM<6:0> | | | | | | 4444 | | | |

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

- Note 1:** Interrupts disabled on devices without ECAN™ modules.
Note 2: Interrupts disabled on devices without DAC.

TABLE 4-5: TIMER REGISTER MAP

| SFR Name | SFR 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 | | | | | | | | | | | | | | | | xxxx |
| PR1 | 0102 | Period Register 1 | | | | | | | | | | | | | | | | FFFF |
| T1CON | 0104 | TON | — | TSIDL | — | — | — | — | — | — | TGATE | TCKPS<1:0> | — | TSYNC | TCS | — | — | 0000 |
| TMR2 | 0106 | Timer2 Register | | | | | | | | | | | | | | | | xxxx |
| TMR3HLD | 0108 | Timer3 Holding Register (for 32-bit timer operations only) | | | | | | | | | | | | | | | | xxxx |
| TMR3 | 010A | Timer3 Register | | | | | | | | | | | | | | | | xxxx |
| PR2 | 010C | Period Register 2 | | | | | | | | | | | | | | | | FFFF |
| PR3 | 010E | Period Register 3 | | | | | | | | | | | | | | | | FFFF |
| T2CON | 0110 | TON | — | TSIDL | — | — | — | — | — | — | TGATE | TCKPS<1:0> | T32 | — | TCS | — | — | 0000 |
| T3CON | 0112 | TON | — | TSIDL | — | — | — | — | — | — | TGATE | TCKPS<1:0> | — | — | TCS | — | — | 0000 |
| TMR4 | 0114 | Timer4 Register | | | | | | | | | | | | | | | | xxxx |
| TMR5HLD | 0116 | Timer5 Holding Register (for 32-bit timer operations only) | | | | | | | | | | | | | | | | xxxx |
| TMR5 | 0118 | Timer5 Register | | | | | | | | | | | | | | | | xxxx |
| PR4 | 011A | Period Register 4 | | | | | | | | | | | | | | | | FFFF |
| PR5 | 011C | Period Register 5 | | | | | | | | | | | | | | | | FFFF |
| T4CON | 011E | TON | — | TSIDL | — | — | — | — | — | — | TGATE | TCKPS<1:0> | T32 | — | TCS | — | — | 0000 |
| T5CON | 0120 | TON | — | TSIDL | — | — | — | — | — | — | TGATE | TCKPS<1:0> | — | — | TCS | — | — | 0000 |

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

TABLE 4-6: INPUT CAPTURE REGISTER MAP

| SFR Name | SFR 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 | | | | | | | | | | | | | | | | xxxx |
| IC1CON | 0142 | — | — | ICSIDL | — | — | — | — | — | ICTMR | ICI<1:0> | ICOV | ICBNE | ICM<2:0> | | | 0000 | |
| IC2BUF | 0144 | Input 2 Capture Register | | | | | | | | | | | | | | | | xxxx |
| IC2CON | 0146 | — | — | ICSIDL | — | — | — | — | — | ICTMR | ICI<1:0> | ICOV | ICBNE | ICM<2:0> | | | 0000 | |
| IC7BUF | 0158 | Input 7 Capture Register | | | | | | | | | | | | | | | | xxxx |
| IC7CON | 015A | — | — | ICSIDL | — | — | — | — | — | ICTMR | ICI<1:0> | ICOV | ICBNE | ICM<2:0> | | | 0000 | |
| IC8BUF | 015C | Input 8 Capture Register | | | | | | | | | | | | | | | | xxxx |
| IC8CON | 015E | — | — | ICSIDL | — | — | — | — | — | ICTMR | ICI<1:0> | ICOV | ICBNE | ICM<2:0> | | | 0000 | |

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

TABLE 4-7: OUTPUT COMPARE REGISTER MAP

| SFR Name | SFR 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 | | | | | | | | | | | | | | | | xxxx |
| OC1R | 0182 | Output Compare 1 Register | | | | | | | | | | | | | | | | xxxx |
| OC1CON | 0184 | — | — | OCSIDL | — | — | — | — | — | — | — | — | OCFLT | OCTSEL | OCM<2:0> | | | 0000 |
| OC2RS | 0186 | Output Compare 2 Secondary Register | | | | | | | | | | | | | | | | xxxx |
| OC2R | 0188 | Output Compare 2 Register | | | | | | | | | | | | | | | | xxxx |
| OC2CON | 018A | — | — | OCSIDL | — | — | — | — | — | — | — | — | OCFLT | OCTSEL | OCM<2:0> | | | 0000 |
| OC3RS | 018C | Output Compare 3 Secondary Register | | | | | | | | | | | | | | | | xxxx |
| OC3R | 018E | Output Compare 3 Register | | | | | | | | | | | | | | | | xxxx |
| OC3CON | 0190 | — | — | OCSIDL | — | — | — | — | — | — | — | — | OCFLT | OCTSEL | OCM<2:0> | | | 0000 |
| OC4RS | 0192 | Output Compare 4 Secondary Register | | | | | | | | | | | | | | | | xxxx |
| OC4R | 0194 | Output Compare 4 Register | | | | | | | | | | | | | | | | xxxx |
| OC4CON | 0196 | — | — | OCSIDL | — | — | — | — | — | — | — | — | OCFLT | OCTSEL | OCM<2:0> | | | 0000 |

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

TABLE 4-8: 6-OUTPUT PWM1 REGISTER MAP

| SFR Name | SFR 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 | Reset State | |
|-----------|----------|----------------------------|------------------------------------|----------|--------|-------------|--------|------------|--------|------------|----------|--------|-------------|--------|------------|--------|--------|-------------|------|
| P1TCON | 01C0 | PTEN | — | PTSIDL | — | — | — | — | — | PTOPS<3:0> | | | PTCKPS<1:0> | | PTMOD<1:0> | | 0000 | | |
| P1TMR | 01C2 | PTDIR | PWM Timer Count Value Register | | | | | | | | | | | | | | | | 0000 |
| P1TPER | 01C4 | — | PWM Time Base Period Register | | | | | | | | | | | | | | | | 0000 |
| P1SECMP | 01C6 | SEVTDIR | PWM Special Event Compare Register | | | | | | | | | | | | | | | | 0000 |
| PWM1CON1 | 01C8 | — | — | — | — | — | PMOD3 | PMOD2 | PMOD1 | — | PEN3H | PEN2H | PEN1H | — | PEN3L | PEN2L | PEN1L | 00FF | |
| PWM1CON2 | 01CA | — | — | — | — | SEVOPS<3:0> | | | — | — | — | — | — | IUE | OSYNC | UDIS | 0000 | | |
| P1DTCON1 | 01CC | DTBPS<1:0> | | DTB<5:0> | | | | DTAPS<1:0> | | | DTA<5:0> | | | | | 0000 | | | |
| P1DTCON2 | 01CE | — | — | — | — | — | — | — | — | — | — | DTS3A | DTS3I | DTS2A | DTS2I | DTS1A | DTS1I | 0000 | |
| P1FLTACON | 01D0 | — | — | FAOV3H | FAOV3L | FAOV2H | FAOV2L | FAOV1H | FAOV1L | FLTAM | — | — | — | FAEN3 | FAEN2 | FAEN1 | 0000 | | |
| P1OVDCON | 01D4 | — | — | POVD3H | POVD3L | POVD2H | POVD2L | POVD1H | POVD1L | — | — | POUT3H | POUT3L | POUT2H | POUT2L | POUT1H | POUT1L | FF00 | |
| P1DC1 | 01D6 | PWM Duty Cycle #1 Register | | | | | | | | | | | | | | | | 0000 | |
| P1DC2 | 01D8 | PWM Duty Cycle #2 Register | | | | | | | | | | | | | | | | 0000 | |
| P1DC3 | 01DA | PWM Duty Cycle #3 Register | | | | | | | | | | | | | | | | 0000 | |

Legend: u = uninitialized bit, — = unimplemented, read as '0'

TABLE 4-9: 2-OUTPUT PWM2 REGISTER MAP

| SFR 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 | Reset State | |
|-----------|-------|----------------------------|------------------------------------|----------|--------|-------------|--------|------------|--------|------------|-------|-------|-------------|------------|-------|--------|--------|-------------|------|
| P2TCON | 05C0 | PTEN | — | PTSIDL | — | — | — | — | — | PTOPS<3:0> | | | PTCKPS<1:0> | PTMOD<1:0> | | | 0000 | | |
| P2TMR | 05C2 | PTDIR | PWM Timer Count Value Register | | | | | | | | | | | | | | | | 0000 |
| P2TPER | 05C4 | — | PWM Time Base Period Register | | | | | | | | | | | | | | | | 0000 |
| P2SECMP | 05C6 | SEVTDIR | PWM Special Event Compare Register | | | | | | | | | | | | | | | | 0000 |
| PWM2CON1 | 05C8 | — | — | — | — | — | — | — | PMOD1 | — | — | — | PEN1H | — | — | — | PEN1L | 00FF | |
| PWM2CON2 | 05CA | — | — | — | — | SEVOPS<3:0> | | | — | — | — | — | — | IUE | OSYNC | UDIS | 0000 | | |
| P2DTCON1 | 05CC | DTBPS<1:0> | | DTB<5:0> | | | | DTAPS<1:0> | | DTA<5:0> | | | | | | | 0000 | | |
| P2DTCON2 | 05CE | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DTS1A | DTS1I | 0000 | |
| P2FLTACON | 05D0 | — | — | — | — | — | — | FAOV1H | FAOV1L | FLTAM | — | — | — | — | — | — | FAEN1 | 0000 | |
| P2OVDCON | 05D4 | — | — | — | — | — | — | POVD1H | POVD1L | — | — | — | — | — | — | POUT1H | POUT1L | FF00 | |
| P2DC1 | 05D6 | PWM Duty Cycle #1 Register | | | | | | | | | | | | | | | | 0000 | |

Legend: u = uninitialized bit, — = unimplemented, read as '0'

TABLE 4-10: QE1 REGISTER MAP

| SFR 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 | Reset State |
|----------|-------|------------------------|--------|---------|--------|--------|-----------|-------|-------|-----------|--------|-------------|-------|--------|-------|----------|-------|-------------|
| QE1CON | 01E0 | CNTERR | — | QEISIDL | INDX | UPDN | QEIM<2:0> | | SWPAB | PCDOUT | TQGATE | TQCKPS<1:0> | | POSRES | TQCS | UPDN_SRC | 0000 | |
| DFLT1CON | 01E2 | — | — | — | — | — | IMV<1:0> | CEID | QEOUT | QECK<2:0> | | | — | — | — | — | 0000 | |
| POS1CNT | 01E4 | Position Counter<15:0> | | | | | | | | | | | | | | | | 0000 |
| MAX1CNT | 01E6 | Maximum Count<15:0> | | | | | | | | | | | | | | | | FFFF |

Legend: u = uninitialized bit, — = unimplemented, read as '0'

TABLE 4-11: QE2 REGISTER MAP

| SFR 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 | Reset State |
|----------|-------|------------------------|--------|---------|--------|--------|-----------|-------|-------|-----------|--------|-------------|-------|--------|-------|----------|-------|-------------|
| QE2CON | 01F0 | CNTERR | — | QEISIDL | INDX | UPDN | QEIM<2:0> | | SWPAB | PCDOUT | TQGATE | TQCKPS<1:0> | | POSRES | TQCS | UPDN_SRC | 0000 | |
| DFLT2CON | 01F2 | — | — | — | — | — | IMV<1:0> | CEID | QEOUT | QECK<2:0> | | | — | — | — | — | 0000 | |
| POS2CNT | 01F4 | Position Counter<15:0> | | | | | | | | | | | | | | | | 0000 |
| MAX2CNT | 01F6 | Maximum Count<15:0> | | | | | | | | | | | | | | | | FFFF |

Legend: u = uninitialized bit, — = unimplemented, read as '0'

TABLE 4-12: I2C1 REGISTER MAP

| SFR Name | SFR Addr | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | All Resets |
|----------|----------|---------|--------|---------|--------|--------|--------|-----------------------|------------------------------|-------------------|-------|-------|-------|-------|-------|-------|-------|------------|
| I2C1RCV | 0200 | — | — | — | — | — | — | — | — | Receive Register | | | | | | | | 0000 |
| I2C1TRN | 0202 | — | — | — | — | — | — | — | — | Transmit Register | | | | | | | | 00FF |
| I2C1BRG | 0204 | — | — | — | — | — | — | — | Baud Rate Generator Register | | | | | | | | 0000 | |
| I2C1CON | 0206 | I2CEN | — | I2CSIDL | SCLREL | IPMIEN | A10M | DISSLW | SMEN | GCEN | STREN | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN | 1000 |
| I2C1STAT | 0208 | ACKSTAT | TRSTAT | — | — | — | BCL | GCSTAT | ADD10 | IWCOL | I2COV | D_A | P | S | R_W | RBF | TBF | 0000 |
| I2C1ADD | 020A | — | — | — | — | — | — | Address Register | | | | | | | | 0000 | | |
| I2C1MSK | 020C | — | — | — | — | — | — | Address Mask Register | | | | | | | | 0000 | | |

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

TABLE 4-13: UART1 REGISTER MAP

| SFR Name | SFR 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 | URXINV | BRGH | PDSEL<1:0> | | STSEL | 0000 |
| U1STA | 0222 | UTXISEL1 | UTXINV | UTXISEL0 | — | UTXBRK | UTXEN | UTXBF | TRMT | URXISEL<1:0> | | ADDEN | RIDLE | PERR | FERR | OERR | URXDA | 0110 |
| U1TXREG | 0224 | — | — | — | — | — | — | — | UTX8 | UART Transmit Register | | | | | | | | xxxxx |
| U1RXREG | 0226 | — | — | — | — | — | — | — | URX8 | UART Received Register | | | | | | | | 0000 |
| U1BRG | 0228 | Baud Rate Generator Prescaler | | | | | | | | | | | | | | | | 0000 |

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

TABLE 4-14: UART2 REGISTER MAP

| SFR Name | SFR 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 |
|----------|----------|-------------------------------|--------|----------|--------|--------|--------|-------|-------|------------------------|--------|-------|--------|-------|------------|-------|-------|------------|
| U2MODE | 0230 | UARTEN | — | USIDL | IREN | RTSMD | — | UEN1 | UEN0 | WAKE | LPBACK | ABAUD | URXINV | BRGH | PDSEL<1:0> | | STSEL | 0000 |
| U2STA | 0232 | UTXISEL1 | UTXINV | UTXISEL0 | — | UTXBRK | UTXEN | UTXBF | TRMT | URXISEL<1:0> | | ADDEN | RIDLE | PERR | FERR | OERR | URXDA | 0110 |
| U2TXREG | 0234 | — | — | — | — | — | — | — | UTX8 | UART Transmit Register | | | | | | | | xxxxx |
| U2RXREG | 0236 | — | — | — | — | — | — | — | URX8 | UART Receive Register | | | | | | | | 0000 |
| U2BRG | 0238 | Baud Rate Generator Prescaler | | | | | | | | | | | | | | | | 0000 |

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

TABLE 4-15: SPI1 REGISTER MAP

| SFR Name | SFR Addr | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | All Resets |
|----------|----------|---|--------|---------|--------|--------|--------|-------|-------|-------|--------|-------|-----------|-------|-------|-----------|--------|------------|
| SPI1STAT | 0240 | SPIEN | — | SPISIDL | — | — | — | — | — | — | SPIROV | — | — | — | — | SPITBF | SPIRBF | 0000 |
| SPI1CON1 | 0242 | — | — | — | DISSCK | DISSDO | MODE16 | SMP | CKE | SSEN | CKP | MSTEN | SPRE<2:0> | | | PPRE<1:0> | | 0000 |
| SPI1CON2 | 0244 | FRMEN | SPIFSD | FRMPOL | — | — | — | — | — | — | — | — | — | — | — | FRMDLY | — | 0000 |
| SPI1BUF | 0248 | SPI1 Transmit and Receive Buffer Register | | | | | | | | | | | | | | | | 0000 |

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

TABLE 4-16: SPI2 REGISTER MAP

| SFR Name | SFR 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 |
|----------|----------|---|--------|---------|--------|--------|--------|-------|-------|-------|--------|-------|-----------|-------|-------|-----------|--------|------------|
| SPI2STAT | 0260 | SPIEN | — | SPISIDL | — | — | — | — | — | — | SPIROV | — | — | — | — | SPITBF | SPIRBF | 0000 |
| SPI2CON1 | 0262 | — | — | — | DISSCK | DISSDO | MODE16 | SMP | CKE | SSEN | CKP | MSTEN | SPRE<2:0> | | | PPRE<1:0> | | 0000 |
| SPI2CON2 | 0264 | FRMEN | SPIFSD | FRMPOL | — | — | — | — | — | — | — | — | — | — | — | FRMDLY | — | 0000 |
| SPI2BUF | 0268 | SPI2 Transmit and Receive Buffer Register | | | | | | | | | | | | | | | | 0000 |

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

TABLE 4-17: ADC1 REGISTER MAP FOR dsPIC33FJ64MC202/802, dsPIC33FJ128MC202/802 AND dsPIC33FJ32MC302

| 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 |
| AD1CON1 | 0320 | ADON | — | ADSIDL | ADDMABM | — | AD12B | FORM<1:0> | | SSRC<2:0> | | | — | SIMSAM | ASAM | SAMP | DONE | 0000 |
| AD1CON2 | 0322 | VCFG<2:0> | | | — | — | CSCNA | CHPS<1:0> | | BUFS | — | SMPI<3:0> | | | | BUFM | ALTS | 0000 |
| AD1CON3 | 0324 | ADRC | — | — | SAMC<4:0> | | | | | ADCS<7:0> | | | | | | | 0000 | |
| AD1CHS123 | 0326 | — | — | — | — | — | CH123NB<1:0> | | CH123SB | — | — | — | — | — | CH123NA<1:0> | | CH123SA | 0000 |
| AD1CHS0 | 0328 | CH0NB | — | — | CH0SB<4:0> | | | | | CH0NA | — | — | CH0SA<4:0> | | | | | 0000 |
| AD1PCFGL | 032C | — | — | — | — | — | — | — | — | — | — | PCFG5 | PCFG4 | PCFG3 | PCFG2 | PCFG1 | PCFG0 | 0000 |
| AD1CSSL | 0330 | — | — | — | — | — | — | — | — | — | — | CSS5 | CSS4 | CSS3 | CSS2 | CSS1 | CSS0 | 0000 |
| AD1CON4 | 0332 | — | — | — | — | — | — | — | — | — | — | — | — | — | DMABL<2:0> | | | 0000 |

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

TABLE 4-18: ADC1 REGISTER MAP FOR dsPIC33FJ64MC204/804, dsPIC33FJ128MC204/804 AND dsPIC33FJ32MC304

| 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 |
| AD1CON1 | 0320 | ADON | — | ADSIDL | ADDMABM | — | AD12B | FORM<1:0> | | SSRC<2:0> | | — | SIMSAM | ASAM | SAMP | DONE | 0000 | |
| AD1CON2 | 0322 | VCFG<2:0> | | — | — | CSCNA | CHPS<1:0> | | BUFS | — | SMPI<3:0> | | | BUFM | ALTS | 0000 | | |
| AD1CON3 | 0324 | ADRC | — | — | SAMC<4:0> | | | | ADCS<7:0> | | | | | | | 0000 | | |
| AD1CHS123 | 0326 | — | — | — | — | — | CH123NB<1:0> | CH123SB | — | — | — | — | — | CH123NA<1:0> | CH123SA | 0000 | | |
| AD1CHS0 | 0328 | CH0NB | — | — | CH0SB<4:0> | | | | CH0NA | — | — | CH0SA<4:0> | | | | | 0000 | |
| AD1PCFGL | 032C | — | — | — | — | — | — | — | PCFG8 | PCFG7 | PCFG6 | PCFG5 | PCFG4 | PCFG3 | PCFG2 | PCFG1 | PCFG0 | 0000 |
| AD1CSSL | 0330 | — | — | — | — | — | — | — | CSS8 | CSS7 | CSS6 | CSS5 | CSS4 | CSS3 | CSS2 | CSS1 | CSS0 | 0000 |
| AD1CON4 | 0332 | — | — | — | — | — | — | — | — | — | — | — | — | — | DMABL<2:0> | | 0000 | |

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

TABLE 4-19: DAC1 REGISTER MAP FOR dsPIC33FJ128MC804 AND dsPIC33FJ64MC804

| SFR Name | SFR 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 |
|----------|----------|----------------|--------|---------|--------|--------|--------|-------|-------|-------|--------------|--------|-------|-------|--------|-------|-------|------------|
| DAC1CON | 03F0 | DACEN | — | DACSIDL | AMPON | — | — | — | FORM | — | DACFDIV<6:0> | | | | | | | 0000 |
| DAC1STAT | 03F2 | LOEN | — | LMVOEN | — | — | LITYPE | LFULL | LEMPY | ROEN | — | RMVOEN | — | — | RITYPE | RFULL | REMPY | 0000 |
| DAC1DFLT | 03F4 | DAC1DFLT<15:0> | | | | | | | | | | | | | | | | 0000 |
| DAC1RDAT | 03F6 | DAC1RDAT<15:0> | | | | | | | | | | | | | | | | 0000 |
| DAC1LDAT | 03F8 | DAC1LDAT<15:0> | | | | | | | | | | | | | | | | 0000 |

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

TABLE 4-20: DMA 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 | |
|-----------|------|-----------|--------|--------|--------|--------|--------|----------|-------|-------|-------------|------------|-------|-------|-------|-----------|-------|------------|------|
| DMA0CON | 0380 | CHEN | SIZE | DIR | HALF | NULLW | — | — | — | — | — | AMODE<1:0> | | — | — | MODE<1:0> | | 0000 | |
| DMA0REQ | 0382 | FORCE | — | — | — | — | — | — | — | — | IRQSEL<6:0> | | | | | | | | 0000 |
| DMA0STA | 0384 | STA<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA0STB | 0386 | STB<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA0PAD | 0388 | PAD<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA0CNT | 038A | — | — | — | — | — | — | CNT<9:0> | | | | | | | | | | 0000 | |
| DMA1CON | 038C | CHEN | SIZE | DIR | HALF | NULLW | — | — | — | — | — | AMODE<1:0> | | — | — | MODE<1:0> | | 0000 | |
| DMA1REQ | 038E | FORCE | — | — | — | — | — | — | — | — | IRQSEL<6:0> | | | | | | | | 0000 |
| DMA1STA | 0390 | STA<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA1STB | 0392 | STB<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA1PAD | 0394 | PAD<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA1CNT | 0396 | — | — | — | — | — | — | CNT<9:0> | | | | | | | | | | 0000 | |
| DMA2CON | 0398 | CHEN | SIZE | DIR | HALF | NULLW | — | — | — | — | — | AMODE<1:0> | | — | — | MODE<1:0> | | 0000 | |
| DMA2REQ | 039A | FORCE | — | — | — | — | — | — | — | — | IRQSEL<6:0> | | | | | | | | 0000 |
| DMA2STA | 039C | STA<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA2STB | 039E | STB<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA2PAD | 03A0 | PAD<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA2CNT | 03A2 | — | — | — | — | — | — | CNT<9:0> | | | | | | | | | | 0000 | |
| DMA3CON | 03A4 | CHEN | SIZE | DIR | HALF | NULLW | — | — | — | — | — | AMODE<1:0> | | — | — | MODE<1:0> | | 0000 | |
| DMA3REQ | 03A6 | FORCE | — | — | — | — | — | — | — | — | IRQSEL<6:0> | | | | | | | | 0000 |
| DMA3STA | 03A8 | STA<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA3STB | 03AA | STB<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA3PAD | 03AC | PAD<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA3CNT | 03AE | — | — | — | — | — | — | CNT<9:0> | | | | | | | | | | 0000 | |
| DMA4CON | 03B0 | CHEN | SIZE | DIR | HALF | NULLW | — | — | — | — | — | AMODE<1:0> | | — | — | MODE<1:0> | | 0000 | |
| DMA4REQ | 03B2 | FORCE | — | — | — | — | — | — | — | — | IRQSEL<6:0> | | | | | | | | 0000 |
| DMA4STA | 03B4 | STA<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA4STB | 03B6 | STB<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA4PAD | 03B8 | PAD<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA4CNT | 03BA | — | — | — | — | — | — | CNT<9:0> | | | | | | | | | | 0000 | |
| DMA5CON | 03BC | CHEN | SIZE | DIR | HALF | NULLW | — | — | — | — | — | AMODE<1:0> | | — | — | MODE<1:0> | | 0000 | |
| DMA5REQ | 03BE | FORCE | — | — | — | — | — | — | — | — | IRQSEL<6:0> | | | | | | | | 0000 |
| DMA5STA | 03C0 | STA<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA5STB | 03C2 | STB<15:0> | | | | | | | | | | | | | | | | 0000 | |

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

TABLE 4-20: DMA REGISTER MAP (CONTINUED)

| 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 | |
|-----------|------|-------------|--------|--------|--------|------------|--------|----------|--------|--------|-------------|------------|--------|--------|--------|-----------|--------|------------|------|
| DMA5PAD | 03C4 | PAD<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA5CNT | 03C6 | — | — | — | — | — | — | CNT<9:0> | | | | | | | | | | 0000 | |
| DMA6CON | 03C8 | CHEN | SIZE | DIR | HALF | NULLW | — | — | — | — | — | AMODE<1:0> | | — | — | MODE<1:0> | | 0000 | |
| DMA6REQ | 03CA | FORCE | — | — | — | — | — | — | — | — | IRQSEL<6:0> | | | | | | | | 0000 |
| DMA6STA | 03CC | STA<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA6STB | 03CE | STB<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA6PAD | 03D0 | PAD<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA6CNT | 03D2 | — | — | — | — | — | — | CNT<9:0> | | | | | | | | | | 0000 | |
| DMA7CON | 03D4 | CHEN | SIZE | DIR | HALF | NULLW | — | — | — | — | — | AMODE<1:0> | | — | — | MODE<1:0> | | 0000 | |
| DMA7REQ | 03D6 | FORCE | — | — | — | — | — | — | — | — | IRQSEL<6:0> | | | | | | | | 0000 |
| DMA7STA | 03D8 | STA<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA7STB | 03DA | STB<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA7PAD | 03DC | PAD<15:0> | | | | | | | | | | | | | | | | 0000 | |
| DMA7CNT | 03DE | — | — | — | — | — | — | CNT<9:0> | | | | | | | | | | 0000 | |
| DMACS0 | 03E0 | PWCOL7 | PWCOL6 | PWCOL5 | PWCOL4 | PWCOL3 | PWCOL2 | PWCOL1 | PWCOL0 | XWCOL7 | XWCOL6 | XWCOL5 | XWCOL4 | XWCOL3 | XWCOL2 | XWCOL1 | XWCOL0 | 0000 | |
| DMACS1 | 03E2 | — | — | — | — | LSTCH<3:0> | | | PPST7 | PPST6 | PPST5 | PPST4 | PPST3 | PPST2 | PPST1 | PPST0 | 0000 | | |
| DSADR | 03E4 | DSADR<15:0> | | | | | | | | | | | | | | | | 0000 | |

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

TABLE 4-21: ECAN1 REGISTER MAP WHEN C1CTRL1.WIN = 0 OR 1 (FOR dsPIC33FJ128MC802/804 AND dsPIC33FJ64MC802/804)

| 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 |
|------------|------|--------------|---------|-------------|-------------|-------------|-------------|--------------|--------|-------------|------------|-------------|------------|------------|------------|------------|--------|------------|
| C1CTRL1 | 0400 | — | — | CSIDL | ABAT | — | REQOP<2:0> | | | OPMODE<2:0> | | | — | CANCAP | — | — | WIN | 0480 |
| C1CTRL2 | 0402 | — | — | — | — | — | — | — | — | — | — | — | DNCNT<4:0> | | | | 0000 | |
| C1VEC | 0404 | — | — | — | FILHIT<4:0> | | | | | — | ICODE<6:0> | | | | | 0000 | | |
| C1FCTRL | 0406 | DMABS<2:0> | | | — | — | — | — | — | — | — | — | FSA<4:0> | | | | | 0000 |
| C1FIFO | 0408 | — | — | FBP<5:0> | | | | | | — | — | FNRB<5:0> | | | | | 0000 | |
| C1INTF | 040A | — | — | TXBO | TXBP | RXBP | TXWAR | RXWAR | EWARN | IVRIF | WAKIF | ERRIF | — | FIFOIF | RBOVIF | RBIF | TBIF | 0000 |
| C1INTE | 040C | — | — | — | — | — | — | — | — | IVRIE | WAKIE | ERRIE | — | FIFOIE | RBOVIE | RBIE | TBIE | 0000 |
| C1EC | 040E | TERRCNT<7:0> | | | | | | RERRCNT<7:0> | | | | | | 0000 | | | | |
| C1CFG1 | 0410 | — | — | — | — | — | — | — | — | SJW<1:0> | | BRP<5:0> | | | | | 0000 | |
| C1CFG2 | 0412 | — | WAKFIL | — | — | — | SEG2PH<2:0> | | | SEG2PHTS | SAM | SEG1PH<2:0> | | | PRSEG<2:0> | | 0000 | |
| C1FEN1 | 0414 | FLTEN15 | FLTEN14 | FLTEN13 | FLTEN12 | FLTEN11 | FLTEN10 | FLTEN9 | FLTEN8 | FLTEN7 | FLTEN6 | FLTEN5 | FLTEN4 | FLTEN3 | FLTEN2 | FLTEN1 | FLTEN0 | FFFF |
| C1FMSKSEL1 | 0418 | F7MSK<1:0> | | F6MSK<1:0> | | F5MSK<1:0> | | F4MSK<1:0> | | F3MSK<1:0> | | F2MSK<1:0> | | F1MSK<1:0> | | F0MSK<1:0> | | 0000 |
| C1FMSKSEL2 | 041A | F15MSK<1:0> | | F14MSK<1:0> | | F13MSK<1:0> | | F12MSK<1:0> | | F11MSK<1:0> | | F10MSK<1:0> | | F9MSK<1:0> | | F8MSK<1:0> | | 0000 |

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

TABLE 4-22: ECAN1 REGISTER MAP WHEN C1CTRL1.WIN = 0 (FOR dsPIC33FJ128MC802/804 AND dsPIC33FJ64MC802/804)

| 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 |
|-----------|-----------|-----------------------------|---------|---------|---------|---------|---------|-------------|---------|---------|---------|---------|---------|---------|---------|-------------|---------|------------|
| | 0400-041E | See definition when WIN = x | | | | | | | | | | | | | | | | |
| C1RXFUL1 | 0420 | RXFUL15 | RXFUL14 | RXFUL13 | RXFUL12 | RXFUL11 | RXFUL10 | RXFUL9 | RXFUL8 | RXFUL7 | RXFUL6 | RXFUL5 | RXFUL4 | RXFUL3 | RXFUL2 | RXFUL1 | RXFUL0 | 0000 |
| C1RXFUL2 | 0422 | RXFUL31 | RXFUL30 | RXFUL29 | RXFUL28 | RXFUL27 | RXFUL26 | RXFUL25 | RXFUL24 | RXFUL23 | RXFUL22 | RXFUL21 | RXFUL20 | RXFUL19 | RXFUL18 | RXFUL17 | RXFUL16 | 0000 |
| C1RXOVF1 | 0428 | RXOVF15 | RXOVF14 | RXOVF13 | RXOVF12 | RXOVF11 | RXOVF10 | RXOVF9 | RXOVF8 | RXOVF7 | RXOVF6 | RXOVF5 | RXOVF4 | RXOVF3 | RXOVF2 | RXOVF1 | RXOVF0 | 0000 |
| C1RXOVF2 | 042A | RXOVF31 | RXOVF30 | RXOVF29 | RXOVF28 | RXOVF27 | RXOVF26 | RXOVF25 | RXOVF24 | RXOVF23 | RXOVF22 | RXOVF21 | RXOVF20 | RXOVF19 | RXOVF18 | RXOVF17 | RXOVF16 | 0000 |
| C1TR01CON | 0430 | TXEN1 | TXABT1 | TXLARB1 | TXERR1 | TXREQ1 | RTREN1 | TX1PRI<1:0> | | TXEN0 | TXABT0 | TXLARB0 | TXERR0 | TXREQ0 | RTREN0 | TX0PRI<1:0> | | 0000 |
| C1TR23CON | 0432 | TXEN3 | TXABT3 | TXLARB3 | TXERR3 | TXREQ3 | RTREN3 | TX3PRI<1:0> | | TXEN2 | TXABT2 | TXLARB2 | TXERR2 | TXREQ2 | RTREN2 | TX2PRI<1:0> | | 0000 |
| C1TR45CON | 0434 | TXEN5 | TXABT5 | TXLARB5 | TXERR5 | TXREQ5 | RTREN5 | TX5PRI<1:0> | | TXEN4 | TXABT4 | TXLARB4 | TXERR4 | TXREQ4 | RTREN4 | TX4PRI<1:0> | | 0000 |
| C1TR67CON | 0436 | TXEN7 | TXABT7 | TXLARB7 | TXERR7 | TXREQ7 | RTREN7 | TX7PRI<1:0> | | TXEN6 | TXABT6 | TXLARB6 | TXERR6 | TXREQ6 | RTREN6 | TX6PRI<1:0> | | 0000 |
| C1RXD | 0440 | Received Data Word | | | | | | | | | | | | | | | | xxxx |
| C1TXD | 0442 | Transmit Data Word | | | | | | | | | | | | | | | | xxxx |

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

TABLE 4-23: ECAN1 REGISTER MAP WHEN C1CTRL1.WIN = 1 (FOR dsPIC33FJ128MC802/804 AND dsPIC33FJ64MC802/804)

| 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 |
|------------|-----------|-----------------------------|--------|--------|------------|--------|----------|------------|-------|-------|------------|------------|-------|-------|-------|-------|-------|------------|
| | 0400-041E | See definition when WIN = x | | | | | | | | | | | | | | | | |
| C1BUFNT1 | 0420 | F3BP<3:0> | | | F2BP<3:0> | | | F1BP<3:0> | | | F0BP<3:0> | | | | | | 0000 | |
| C1BUFNT2 | 0422 | F7BP<3:0> | | | F6BP<3:0> | | | F5BP<3:0> | | | F4BP<3:0> | | | | | | 0000 | |
| C1BUFNT3 | 0424 | F11BP<3:0> | | | F10BP<3:0> | | | F9BP<3:0> | | | F8BP<3:0> | | | | | | 0000 | |
| C1BUFNT4 | 0426 | F15BP<3:0> | | | F14BP<3:0> | | | F13BP<3:0> | | | F12BP<3:0> | | | | | | 0000 | |
| C1RXM0SID | 0430 | SID<10:3> | | | | | SID<2:0> | | — | MIDE | — | EID<17:16> | | | | | xxxx | |
| C1RXM0EID | 0432 | EID<15:8> | | | | | EID<7:0> | | | | | | | | | | xxxx | |
| C1RXM1SID | 0434 | SID<10:3> | | | | | SID<2:0> | | — | MIDE | — | EID<17:16> | | | | | xxxx | |
| C1RXM1EID | 0436 | EID<15:8> | | | | | EID<7:0> | | | | | | | | | | xxxx | |
| C1RXM2SID | 0438 | SID<10:3> | | | | | SID<2:0> | | — | MIDE | — | EID<17:16> | | | | | xxxx | |
| C1RXM2EID | 043A | EID<15:8> | | | | | EID<7:0> | | | | | | | | | | xxxx | |
| C1RXF0SID | 0440 | SID<10:3> | | | | | SID<2:0> | | — | EXIDE | — | EID<17:16> | | | | | xxxx | |
| C1RXF0EID | 0442 | EID<15:8> | | | | | EID<7:0> | | | | | | | | | | xxxx | |
| C1RXF1SID | 0444 | SID<10:3> | | | | | SID<2:0> | | — | EXIDE | — | EID<17:16> | | | | | xxxx | |
| C1RXF1EID | 0446 | EID<15:8> | | | | | EID<7:0> | | | | | | | | | | xxxx | |
| C1RXF2SID | 0448 | SID<10:3> | | | | | SID<2:0> | | — | EXIDE | — | EID<17:16> | | | | | xxxx | |
| C1RXF2EID | 044A | EID<15:8> | | | | | EID<7:0> | | | | | | | | | | xxxx | |
| C1RXF3SID | 044C | SID<10:3> | | | | | SID<2:0> | | — | EXIDE | — | EID<17:16> | | | | | xxxx | |
| C1RXF3EID | 044E | EID<15:8> | | | | | EID<7:0> | | | | | | | | | | xxxx | |
| C1RXF4SID | 0450 | SID<10:3> | | | | | SID<2:0> | | — | EXIDE | — | EID<17:16> | | | | | xxxx | |
| C1RXF4EID | 0452 | EID<15:8> | | | | | EID<7:0> | | | | | | | | | | xxxx | |
| C1RXF5SID | 0454 | SID<10:3> | | | | | SID<2:0> | | — | EXIDE | — | EID<17:16> | | | | | xxxx | |
| C1RXF5EID | 0456 | EID<15:8> | | | | | EID<7:0> | | | | | | | | | | xxxx | |
| C1RXF6SID | 0458 | SID<10:3> | | | | | SID<2:0> | | — | EXIDE | — | EID<17:16> | | | | | xxxx | |
| C1RXF6EID | 045A | EID<15:8> | | | | | EID<7:0> | | | | | | | | | | xxxx | |
| C1RXF7SID | 045C | SID<10:3> | | | | | SID<2:0> | | — | EXIDE | — | EID<17:16> | | | | | xxxx | |
| C1RXF7EID | 045E | EID<15:8> | | | | | EID<7:0> | | | | | | | | | | xxxx | |
| C1RXF8SID | 0460 | SID<10:3> | | | | | SID<2:0> | | — | EXIDE | — | EID<17:16> | | | | | xxxx | |
| C1RXF8EID | 0462 | EID<15:8> | | | | | EID<7:0> | | | | | | | | | | xxxx | |
| C1RXF9SID | 0464 | SID<10:3> | | | | | SID<2:0> | | — | EXIDE | — | EID<17:16> | | | | | xxxx | |
| C1RXF9EID | 0466 | EID<15:8> | | | | | EID<7:0> | | | | | | | | | | xxxx | |
| C1RXF10SID | 0468 | SID<10:3> | | | | | SID<2:0> | | — | EXIDE | — | EID<17:16> | | | | | xxxx | |
| C1RXF10EID | 046A | EID<15:8> | | | | | EID<7:0> | | | | | | | | | | xxxx | |
| C1RXF11SID | 046C | SID<10:3> | | | | | SID<2:0> | | — | EXIDE | — | EID<17:16> | | | | | xxxx | |

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

TABLE 4-23: ECAN1 REGISTER MAP WHEN C1CTRL1.WIN = 1 (FOR dsPIC33FJ128MC802/804 AND dsPIC33FJ64MC802/804) (CONTINUED)

| 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 |
|------------|------|-----------|--------|--------|--------|--------|--------|----------|-------|-------|-------|-------|------------|-------|-------|-------|-------|------------|
| C1RXF11EID | 046E | EID<15:8> | | | | | | EID<7:0> | | | | | | | | xxxx | | |
| C1RXF12SID | 0470 | SID<10:3> | | | | | | SID<2:0> | | — | EXIDE | — | EID<17:16> | | | xxxx | | |
| C1RXF12EID | 0472 | EID<15:8> | | | | | | EID<7:0> | | | | | | | | xxxx | | |
| C1RXF13SID | 0474 | SID<10:3> | | | | | | SID<2:0> | | — | EXIDE | — | EID<17:16> | | | xxxx | | |
| C1RXF13EID | 0476 | EID<15:8> | | | | | | EID<7:0> | | | | | | | | xxxx | | |
| C1RXF14SID | 0478 | SID<10:3> | | | | | | SID<2:0> | | — | EXIDE | — | EID<17:16> | | | xxxx | | |
| C1RXF14EID | 047A | EID<15:8> | | | | | | EID<7:0> | | | | | | | | xxxx | | |
| C1RXF15SID | 047C | SID<10:3> | | | | | | SID<2:0> | | — | EXIDE | — | EID<17:16> | | | xxxx | | |
| C1RXF15EID | 047E | EID<15:8> | | | | | | EID<7:0> | | | | | | | | xxxx | | |

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

TABLE 4-24: PERIPHERAL PIN SELECT INPUT 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 | — | — | — | INT1R<4:0> | | | | — | — | — | — | — | — | — | — | — | — | 1F00 |
| RPINR1 | 0682 | — | — | — | — | — | — | — | — | — | — | — | INT2R<4:0> | | | | 001F | | |
| RPINR3 | 0686 | — | — | — | T3CKR<4:0> | | | | — | — | — | T2CKR<4:0> | | | | 1F1F | | | |
| RPINR4 | 0688 | — | — | — | T5CKR<4:0> | | | | — | — | — | T4CKR<4:0> | | | | 1F1F | | | |
| RPINR7 | 068E | — | — | — | IC2R<4:0> | | | | — | — | — | IC1R<4:0> | | | | 1F1F | | | |
| RPINR10 | 0694 | — | — | — | IC8R<4:0> | | | | — | — | — | IC7R<4:0> | | | | 1F1F | | | |
| RPINR11 | 0696 | — | — | — | — | — | — | — | — | — | — | — | OCFAR<4:0> | | | | 001F | | |
| RPINR12 | 0698 | — | — | — | — | — | — | — | — | — | — | — | FLTA1R<4:0> | | | | 001F | | |
| RPINR13 | 069A | — | — | — | — | — | — | — | — | — | — | — | FLTA2R<4:0> | | | | 001F | | |
| RPINR14 | 069C | — | — | — | QEB1R<4:0> | | | | — | — | — | QEA1R<4:0> | | | | 1F1F | | | |
| RPINR15 | 069E | — | — | — | — | — | — | — | — | — | — | — | INDX1R<4:0> | | | | 001F | | |
| RPINR16 | 06A0 | — | — | — | QEB2R<4:0> | | | | — | — | — | QEA2R<4:0> | | | | 1F1F | | | |
| RPINR17 | 06A2 | — | — | — | — | — | — | — | — | — | — | — | INDX2R<4:0> | | | | 001F | | |
| RPINR18 | 06A4 | — | — | — | U1CTSR<4:0> | | | | — | — | — | U1RXR<4:0> | | | | 1F1F | | | |
| RPINR19 | 06A6 | — | — | — | U2CTSR<4:0> | | | | — | — | — | U2RXR<4:0> | | | | 1F1F | | | |
| RPINR20 | 06A8 | — | — | — | SCK1R<4:0> | | | | — | — | — | SDI1R<4:0> | | | | 1F1F | | | |
| RPINR21 | 06AA | — | — | — | — | — | — | — | — | — | — | — | SS1R<4:0> | | | | 001F | | |
| RPINR22 | 06AC | — | — | — | SCK2R<4:0> | | | | — | — | — | SDI2R<4:0> | | | | 1F1F | | | |
| RPINR23 | 06AE | — | — | — | — | — | — | — | — | — | — | — | SS2R<4:0> | | | | 001F | | |
| RPINR26 ⁽¹⁾ | 06B4 | — | — | — | — | — | — | — | — | — | — | — | C1RXR<4:0> | | | | 001F | | |

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

Note 1: This register is present for dsPIC33FJ128MC802/804 and dsPIC33FJ64MC802/804 devices only.

TABLE 4-25: PERIPHERAL PIN SELECT OUTPUT REGISTER MAP FOR dsPIC33FJ128MC202/802, dsPIC33FJ64MC202/802 AND dsPIC33FJ32MC302

| 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 |
|-----------|------|--------|--------|--------|------------|--------|--------|-------|-------|-------|-------|------------|-------|-------|-------|-------|-------|------------|
| RPOR0 | 06C0 | — | — | — | RP1R<4:0> | | | | — | — | — | RP0R<4:0> | | | | 0000 | | |
| RPOR1 | 06C2 | — | — | — | RP3R<4:0> | | | | — | — | — | RP2R<4:0> | | | | 0000 | | |
| RPOR2 | 06C4 | — | — | — | RP5R<4:0> | | | | — | — | — | RP4R<4:0> | | | | 0000 | | |
| RPOR3 | 06C6 | — | — | — | RP7R<4:0> | | | | — | — | — | RP6R<4:0> | | | | 0000 | | |
| RPOR4 | 06C8 | — | — | — | RP9R<4:0> | | | | — | — | — | RP8R<4:0> | | | | 0000 | | |
| RPOR5 | 06CA | — | — | — | RP11R<4:0> | | | | — | — | — | RP10R<4:0> | | | | 0000 | | |
| RPOR6 | 06CC | — | — | — | RP13R<4:0> | | | | — | — | — | RP12R<4:0> | | | | 0000 | | |
| RPOR7 | 06CE | — | — | — | RP15R<4:0> | | | | — | — | — | RP14R<4:0> | | | | 0000 | | |

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

TABLE 4-26: PERIPHERAL PIN SELECT OUTPUT REGISTER MAP FOR dsPIC33FJ128MC204/804, dsPIC33FJ64MC204/804 AND dsPIC33FJ32MC304

| 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 |
|-----------|------|--------|--------|--------|------------|--------|--------|-------|-------|-------|-------|------------|-------|-------|-------|-------|-------|------------|
| RPOR0 | 06C0 | — | — | — | RP1R<4:0> | | | | — | — | — | RP0R<4:0> | | | | 0000 | | |
| RPOR1 | 06C2 | — | — | — | RP3R<4:0> | | | | — | — | — | RP2R<4:0> | | | | 0000 | | |
| RPOR2 | 06C4 | — | — | — | RP5R<4:0> | | | | — | — | — | RP4R<4:0> | | | | 0000 | | |
| RPOR3 | 06C6 | — | — | — | RP7R<4:0> | | | | — | — | — | RP6R<4:0> | | | | 0000 | | |
| RPOR4 | 06C8 | — | — | — | RP9R<4:0> | | | | — | — | — | RP8R<4:0> | | | | 0000 | | |
| RPOR5 | 06CA | — | — | — | RP11R<4:0> | | | | — | — | — | RP10R<4:0> | | | | 0000 | | |
| RPOR6 | 06CC | — | — | — | RP13R<4:0> | | | | — | — | — | RP12R<4:0> | | | | 0000 | | |
| RPOR7 | 06CE | — | — | — | RP15R<4:0> | | | | — | — | — | RP14R<4:0> | | | | 0000 | | |
| RPOR8 | 06D0 | — | — | — | RP17R<4:0> | | | | — | — | — | RP16R<4:0> | | | | 0000 | | |
| RPOR9 | 06D2 | — | — | — | RP19R<4:0> | | | | — | — | — | RP18R<4:0> | | | | 0000 | | |
| RPOR10 | 06D4 | — | — | — | RP21R<4:0> | | | | — | — | — | RP20R<4:0> | | | | 0000 | | |
| RPOR11 | 06D6 | — | — | — | RP23R<4:0> | | | | — | — | — | RP22R<4:0> | | | | 0000 | | |
| RPOR12 | 06D8 | — | — | — | RP25R<4:0> | | | | — | — | — | RP24R<4:0> | | | | 0000 | | |

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

TABLE 4-27: PARALLEL MASTER/SLAVE PORT REGISTER MAP FOR dsPIC33FJ128MC202/802, dsPIC33FJ64MC202/802 AND dsPIC33FJ32MC302

| 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 | ADRMUX<1:0> | | PTBEEN | PTWREN | PTRDEN | CSF1 | CSF0 | ALP | — | CS1P | BEP | WRSP | RDSP | 0000 | |
| PMMODE | 0602 | BUSY | IRQM<1:0> | | INCM<1:0> | | MODE16 | MODE<1:0> | | WAITB<1:0> | | WAITM<3:0> | | | WAITE<1:0> | | 0000 | | |
| PMADDR | 0604 | ADDR15 | CS1 | ADDR<13:0> | | | | | | | | | | | | | | | 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 |
| PMPDIN2 | 060A | Parallel Port Data In Register 2 (Buffers 2 and 3) | | | | | | | | | | | | | | | | | 0000 |
| PMAEN | 060C | — | PTEN14 | — | — | — | — | — | — | — | — | — | — | — | — | — | PTEN<1:0> | 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 4-28: PARALLEL MASTER/SLAVE PORT REGISTER MAP FOR dsPIC33FJ128MC204/804, dsPIC33FJ64MC204/804 AND dsPIC33FJ32MC304

| 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 | ADRMUX<1:0> | | PTBEEN | PTWREN | PTRDEN | CSF1 | CSF0 | ALP | — | CS1P | BEP | WRSP | RDSP | 0000 | |
| PMMODE | 0602 | BUSY | IRQM<1:0> | | INCM<1:0> | | MODE16 | MODE<1:0> | | WAITB<1:0> | | WAITM<3:0> | | | WAITE<1:0> | | 0000 | | |
| PMADDR | 0604 | ADDR15 | CS1 | ADDR<13:0> | | | | | | | | | | | | | | | 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 |
| PMPDIN2 | 060A | Parallel Port Data In Register 2 (Buffers 2 and 3) | | | | | | | | | | | | | | | | | 0000 |
| PMAEN | 060C | — | PTEN14 | — | — | — | PTEN<10:0> | | | | | | | | | | | 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 4-29: 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 APTR<1:0> | | | | | | | | | | | | | | | | xxxx |
| ALCFGRPT | 0622 | ALRMEN | CHIME | AMASK<3:0> | | | | ALRMPTR<1:0> | | | ARPT<7:0> | | | | | | 0000 | |
| RTCVAL | 0624 | RTCC Value Register Window based on RTCPTR<1:0> | | | | | | | | | | | | | | | | xxxx |
| RCFGCAL | 0626 | RTCEN | — | RTCWREN | RTCSYNC | HALFSEC | RTCOE | RTCPTR<1:0> | | | CAL<7:0> | | | | | | 0000 | |

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

TABLE 4-30: 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 | VWORD<4:0> | | | | CRCFUL | CRCMPT | — | CRCGO | PLEN<3:0> | | | | | 0000 |
| CRCXOR | 0642 | X<15:0> | | | | | | | | | | | | | | | | 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 4-31: 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 | CVR<3:0> | | | 0000 | |

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

TABLE 4-32: PORTA REGISTER MAP FOR dsPIC33FJ128MC202/802, dsPIC33FJ64MC202/802 AND dsPIC33FJ32MC302

| 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 | — | — | — | — | — | — | — | — | — | — | — | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 079F |
| PORTA | 02C2 | — | — | — | — | — | — | — | — | — | — | — | RA4 | RA3 | RA2 | RA1 | RA0 | xxxx |
| LATA | 02C4 | — | — | — | — | — | — | — | — | — | — | — | LATA4 | LATA3 | LATA2 | LATA1 | LATA0 | xxxx |
| ODCA | 02C6 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |

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

TABLE 4-33: PORTA REGISTER MAP FOR dsPIC33FJ128MC204/804, dsPIC33FJ64MC204/804 AND dsPIC33FJ32MC304

| 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 | TRISA9 | TRISA8 | TRISA7 | — | — | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 079F |
| PORTA | 02C2 | — | — | — | — | — | RA10 | RA9 | RA8 | RA7 | — | — | RA4 | RA3 | RA2 | RA1 | RA0 | xxxx |
| LATA | 02C4 | — | — | — | — | — | LATA10 | LATA9 | LATA8 | LATA7 | — | — | LATA4 | LATA3 | LATA2 | LATA1 | LATA0 | xxxx |
| ODCA | 02C6 | — | — | — | — | — | ODCA10 | ODCA9 | ODCA8 | ODCA7 | — | — | — | — | — | — | — | 0000 |

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

TABLE 4-34: 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 | xxxx |
| LATB | 02CC | LATB15 | LATB14 | LATB13 | LATB12 | LATB11 | LATB10 | LATB9 | LATB8 | LATB7 | LATB6 | LATB5 | LATB4 | LATB3 | LATB2 | LATB1 | LATB0 | xxxx |
| ODCB | 02CE | — | — | — | — | ODCB11 | ODCB10 | ODCB9 | ODCB8 | ODCB7 | ODCB6 | ODCB5 | — | — | — | — | — | 0000 |

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

TABLE 4-35: PORTC REGISTER MAP FOR dsPIC33FJ128MC204/804, dsPIC33FJ64MC204/804 AND dsPIC33FJ32MC304

| 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 | 02D0 | — | — | — | — | — | — | TRISC9 | TRISC8 | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 03FF |
| PORTC | 02D2 | — | — | — | — | — | — | RC9 | RC8 | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | xxxx |
| LATC | 02D4 | — | — | — | — | — | — | LATC9 | LATC8 | LATC7 | LATC6 | LATC5 | LATC4 | LATC3 | LATC2 | LATC1 | LATC0 | xxxx |
| ODCC | 02D6 | — | — | — | — | — | — | ODCC9 | ODCC8 | ODCC7 | ODCC6 | ODCC5 | ODCC4 | ODCC3 | — | — | — | 0000 |

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

TABLE 4-36: SYSTEM 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 | xxxx ⁽¹⁾ |
| OSCCON | 0742 | — | — | COSC<2:0> | | | — | NOSC<2:0> | | CLKLOCK | ILOCK | LOCK | — | CF | — | LPOSCEN | OSWEN | 0300 ⁽²⁾ |
| CLKDIV | 0744 | ROI | — | DOZE<2:0> | | DOZEN | FRCDIV<2:0> | | — | PLLPOST<1:0> | | — | PLLPRE<4:0> | | | | | 3040 |
| PLLFBD | 0746 | — | — | — | — | — | — | — | — | — | — | PLLDIV<8:0> | | | | | | 0030 |
| OSCTUN | 0748 | — | — | — | — | — | — | — | — | — | — | TUN<5:0> | | | | | | 0000 |
| ACLKCON | 074A | — | — | SELACLK | AOSCMD<1:0> | | APSTSCLR<2:0> | | ASRCSEL | — | — | — | — | — | — | — | — | 0000 |

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

Note 1: RCON register Reset values dependent on type of Reset.

Note 2: OSCCON register Reset values dependent on the FOSC Configuration bits and by type of Reset.

TABLE 4-37: SECURITY REGISTER MAP FOR dsPIC33FJ128MC204/804 AND dsPIC33FJ64MC204/804 ONLY

| 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 |
|-----------|------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|------------|
| BSRAM | 0750 | — | — | — | — | — | — | — | — | — | — | — | — | — | IW_BSR | IR_BSR | RL_BSR | 0000 |
| SSRAM | 0752 | — | — | — | — | — | — | — | — | — | — | — | — | — | IW_SSR | IR_SSR | RL_SSR | 0000 |

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

TABLE 4-38: 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 | — | — | NVMOP<3:0> | | | 0000 | |
| NVMKEY | 0766 | — | — | — | — | — | — | — | — | NVMKEY<7:0> | | | | | | | 0000 | |

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

TABLE 4-39: 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 | QE11MD | PWM11MD | — | I2C1MD | U2MD | U1MD | SPI2MD | SPI1MD | — | C1MD | AD1MD | 0000 |
| PMD2 | 0772 | IC8MD | IC7MD | — | — | — | — | IC2MD | IC1MD | — | — | — | — | OC4MD | OC3MD | OC2MD | OC1MD | 0000 |
| PMD3 | 0774 | — | — | — | — | — | CMPMD | RTCCMD | PMPMD | CRCMD | DAC1MD | QE12MD | PWM2MD | — | — | — | — | 0000 |

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

4.2.7 SOFTWARE STACK

In addition to its use as a working register, the W15 register in the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices is also used as a software Stack Pointer. The Stack 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 4-6. 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 concatenates the SRL register to the MSb of the PC prior to the push.

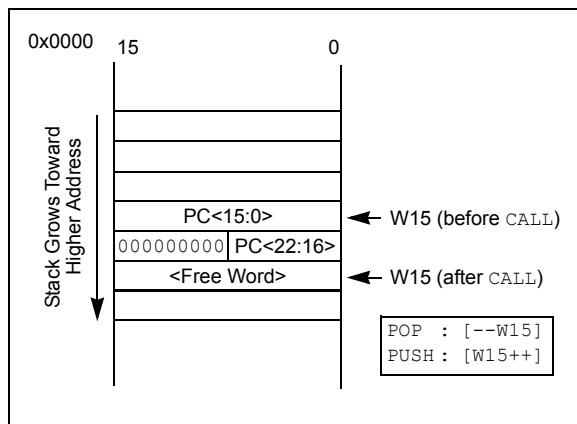
The Stack Pointer Limit register (SPLIM) associated with the Stack Pointer sets an upper address boundary for the stack. SPLIM is uninitialized at Reset. As is the case for the Stack Pointer, SPLIM<0> is forced to '0' because all stack operations must be word aligned.

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 does not occur. The stack error trap occurs on a subsequent push operation. For example, to cause a stack error trap when the stack grows beyond address 0x2000 in RAM, initialize the SPLIM with the value 0x1FFE.

Similarly, a Stack Pointer underflow (stack error) trap is generated when the Stack Pointer address is found to be less than 0x0800. 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 4-6: CALL STACK FRAME



4.2.8 DATA RAM PROTECTION FEATURE

The dsPIC33F product family supports Data RAM protection features that enable segments of RAM to be protected when used in conjunction with Boot and Secure Code Segment Security. BSRAM (Secure RAM segment for BS) is accessible only from the Boot Segment Flash code when enabled. SSRAM (Secure RAM segment for RAM) is accessible only from the Secure Segment Flash code when enabled. See Table 4-1 for an overview of the BSRAM and SSRAM SFRs.

4.3 Instruction Addressing Modes

The addressing modes shown in Table 4-40 form the basis of the addressing modes optimized to support the specific features of individual instructions. The addressing modes provided in the MAC class of instructions differ from those in the other instruction types.

4.3.1 FILE REGISTER INSTRUCTIONS

Most file register instructions use a 13-bit address field (f) to directly address data present in the first 8192 bytes of data memory (near data space). Most file register instructions employ a working register, W0, which is denoted as WREG in these instructions. The destination is typically either the same file register or WREG (with the exception of the MUL instruction), which writes the result to a register or register pair. The MOV instruction allows additional flexibility and can access the entire data space.

4.3.2 MCU INSTRUCTIONS

The three-operand MCU instructions are of the form:

Operand 3 = Operand 1 <function> Operand 2

where Operand 1 is always a working register (that is, the addressing mode can only be register direct), which is referred to as Wb. Operand 2 can be a W register, fetched from data memory, or a 5-bit literal. The result location can be either a W register or a data memory location. The following addressing modes are supported by MCU instructions:

- Register Direct
- Register Indirect
- Register Indirect Post-Modified
- Register Indirect Pre-Modified
- 5-bit or 10-bit Literal

Note: Not all instructions support all the addressing modes given above. Individual instructions can support different subsets of these addressing modes.

TABLE 4-40: FUNDAMENTAL ADDRESSING MODES SUPPORTED

| Addressing Mode | Description |
|---|--|
| File Register Direct | The address of the file register is specified explicitly. |
| Register Direct | The contents of a register are accessed directly. |
| Register Indirect | The contents of Wn forms the Effective Address (EA). |
| Register Indirect Post-Modified | The contents of Wn forms the EA. Wn is post-modified (incremented or decremented) by a constant value. |
| Register Indirect Pre-Modified | Wn is pre-modified (incremented or decremented) by a signed constant value to form the EA. |
| Register Indirect with Register Offset (Register Indexed) | The sum of Wn and Wb forms the EA. |
| Register Indirect with Literal Offset | The sum of Wn and a literal forms the EA. |

4.3.3 MOVE AND ACCUMULATOR INSTRUCTIONS

Move instructions and the DSP accumulator class of instructions provide a greater degree of addressing flexibility than other instructions. In addition to the addressing modes supported by most MCU instructions, move and accumulator instructions also support Register Indirect with Register Offset Addressing mode, also referred to as Register Indexed mode.

Note: For the `MOV` instructions, the addressing mode specified in the instruction can differ for the source and destination EA. However, the 4-bit Wb (Register Offset) field is shared by both source and destination (but typically only used by one).

In summary, the following addressing modes are supported by move and accumulator instructions:

- Register Direct
- Register Indirect
- Register Indirect Post-modified
- Register Indirect Pre-modified
- Register Indirect with Register Offset (Indexed)
- Register Indirect with Literal Offset
- 8-bit Literal
- 16-bit Literal

Note: Not all instructions support all the addressing modes given above. Individual instructions may support different subsets of these addressing modes.

4.3.4 MAC INSTRUCTIONS

The dual source operand DSP instructions (`CLR`, `ED`, `EDAC`, `MAC`, `MPY`, `MPY.N`, `MOVSAC` and `MSC`), also referred to as `MAC` instructions, use a simplified set of addressing modes to allow the user application to effectively manipulate the data pointers through register indirect tables.

The two-source operand prefetch registers must be members of the set {W8, W9, W10, W11}. For data reads, W8 and W9 are always directed to the X RAGU, and W10 and W11 are always directed to the Y AGU. The effective addresses generated (before and after modification) must, therefore, be valid addresses within X data space for W8 and W9 and Y data space for W10 and W11.

Note: Register Indirect with Register Offset Addressing mode is available only for W9 (in X space) and W11 (in Y space).

In summary, the following addressing modes are supported by the `MAC` class of instructions:

- Register Indirect
- Register Indirect Post-Modified by 2
- Register Indirect Post-Modified by 4
- Register Indirect Post-Modified by 6
- Register Indirect with Register Offset (Indexed)

4.3.5 OTHER INSTRUCTIONS

Besides the addressing modes outlined previously, some instructions use literal constants of various sizes. For example, `BRA` (branch) instructions use 16-bit signed literals to specify the branch destination directly, whereas the `DISI` instruction uses a 14-bit unsigned literal field. In some instructions, such as `ADD ACC`, the source of an operand or result is implied by the opcode itself. Certain operations, such as `NOB`, do not have any operands.

4.4 Modulo Addressing

Modulo Addressing mode is a method of providing an automated means to support circular data buffers using hardware. The objective is to remove the need for software to perform data address boundary checks when executing tightly looped code, as is typical in many DSP algorithms.

Modulo Addressing can operate in either data or program space (since the data pointer mechanism is essentially the same for both). One circular buffer can be supported in each of the X (which also provides the pointers into program space) and Y data spaces. Modulo Addressing can operate on any W register pointer. However, it is not advisable to use W14 or W15 for Modulo Addressing since these two registers are used as the Stack Frame Pointer and Stack Pointer, respectively.

In general, any particular circular buffer can be configured to operate in only one direction as there are certain restrictions on the buffer start address (for incrementing buffers), or end address (for decrementing buffers), based upon the direction of the buffer.

The only exception to the usage restrictions is for buffers that have a power-of-two length. As these buffers satisfy the start and end address criteria, they can operate in a bidirectional mode (that is, address boundary checks are performed on both the lower and upper address boundaries).

4.4.1 START AND END ADDRESS

The Modulo Addressing scheme requires that a starting and ending address be specified and loaded into the 16-bit Modulo Buffer Address registers: XMODSRT, XMODEND, YMODSRT and YMODEND (see Table 4-1).

Note: Y space Modulo Addressing EA calculations assume word-sized data (LSb of every EA is always clear).

The length of a circular buffer is not directly specified. It is determined by the difference between the corresponding start and end addresses. The maximum possible length of the circular buffer is 32K words (64 Kbytes).

4.4.2 W ADDRESS REGISTER SELECTION

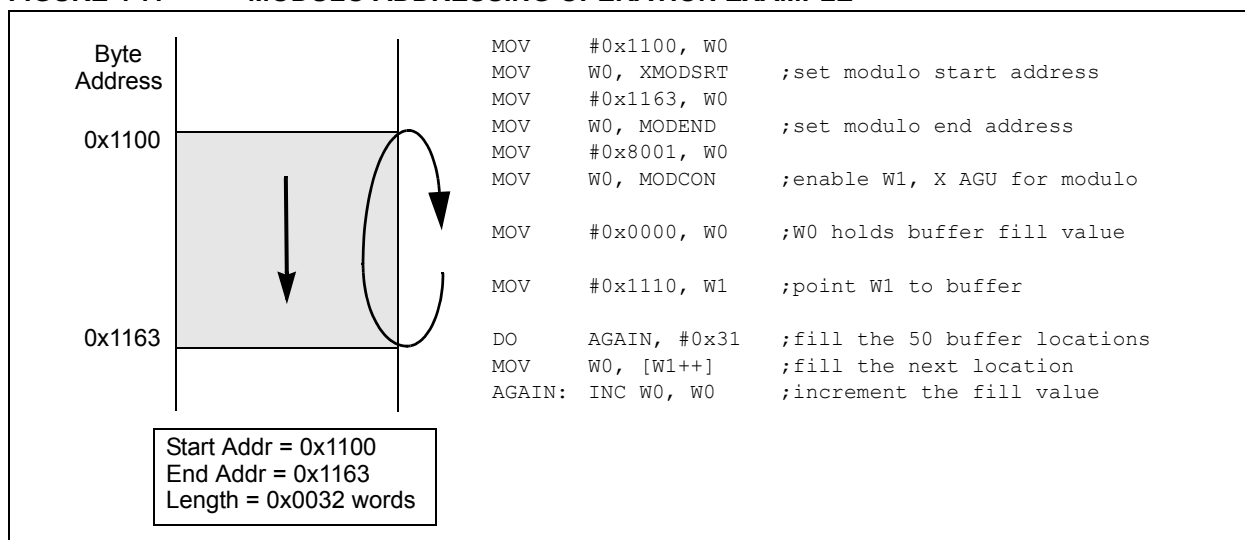
The Modulo and Bit-Reversed Addressing Control register, MODCON<15:0>, contains enable flags as well as a W register field to specify the W Address registers. The XWM and YWM fields select the registers that operate with Modulo Addressing:

- If XWM = 15, X RAGU and X WAGU Modulo Addressing is disabled.
- If YWM = 15, Y AGU Modulo Addressing is disabled.

The X Address Space Pointer W register (XWM), to which Modulo Addressing is to be applied, is stored in MODCON<3:0> (see Table 4-1). Modulo Addressing is enabled for X data space when XWM is set to any value other than '15' and the XMODEN bit is set at MODCON<15>.

The Y Address Space Pointer W register (YWM) to which Modulo Addressing is to be applied is stored in MODCON<7:4>. Modulo Addressing is enabled for Y data space when YWM is set to any value other than '15' and the YMODEN bit is set at MODCON<14>.

FIGURE 4-7: MODULO ADDRESSING OPERATION EXAMPLE



4.4.3 MODULO ADDRESSING APPLICABILITY

Modulo Addressing can be applied to the Effective Address (EA) calculation associated with any W register. Address boundaries check for addresses equal to:

- The upper boundary addresses for incrementing buffers
- The lower boundary addresses for decrementing buffers

It is important to realize that the address boundaries check for addresses less than or greater than the upper (for incrementing buffers) and lower (for decrementing buffers) boundary addresses (not just equal to). Address changes can, therefore, jump beyond boundaries and still be adjusted correctly.

Note: The modulo corrected effective address is written back to the register only when Pre-Modify or Post-Modify Addressing mode is used to compute the effective address. When an address offset (such as [W7 + W2]) is used, Modulo Address correction is performed but the contents of the register remain unchanged.

4.5 Bit-Reversed Addressing

Bit-Reversed Addressing mode is intended to simplify data reordering for radix-2 FFT algorithms. It is supported by the X AGU for data writes only.

The modifier, which can be a constant value or register contents, is regarded as having its bit order reversed. The address source and destination are kept in normal order. Thus, the only operand requiring reversal is the modifier.

4.5.1 BIT-REVERSED ADDRESSING IMPLEMENTATION

Bit-Reversed Addressing mode is enabled in any of these situations:

- BWM bits (W register selection) in the MODCON register are any value other than '15' (the stack cannot be accessed using Bit-Reversed Addressing)
- The BREN bit is set in the XBREV register
- The addressing mode used is Register Indirect with Pre-Increment or Post-Increment

If the length of a bit-reversed buffer is $M = 2^N$ bytes, the last 'N' bits of the data buffer start address must be zeros.

XB<14:0> is the Bit-Reversed Address modifier, or 'pivot point,' which is typically a constant. In the case of an FFT computation, its value is equal to half of the FFT data buffer size.

Note: All bit-reversed EA calculations assume word-sized data (LSb of every EA is always clear). The XB value is scaled accordingly to generate compatible (byte) addresses.

When enabled, Bit-Reversed Addressing is executed only for Register Indirect with Pre-Increment or Post-Increment Addressing and word-sized data writes. It does not function for any other addressing mode or for byte-sized data, and normal addresses are generated instead. When Bit-Reversed Addressing is active, the W Address Pointer is always added to the address modifier (XB), and the offset associated with the Register Indirect Addressing mode is ignored. In addition, as word-sized data is a requirement, the LSb of the EA is ignored (and always clear).

Note: Modulo Addressing and Bit-Reversed Addressing should not be enabled together. If an application attempts to do so, Bit-Reversed Addressing assumes priority when active for the X WAGU and X WAGU, Modulo Addressing is disabled. However, Modulo Addressing continues to function in the X RAGU.

If Bit-Reversed Addressing has already been enabled by setting the BREN (XBREV<15>) bit, a write to the XBREV register should not be immediately followed by an indirect read operation using the W register that has been designated as the bit-reversed pointer.

FIGURE 4-8: BIT-REVERSED ADDRESS EXAMPLE

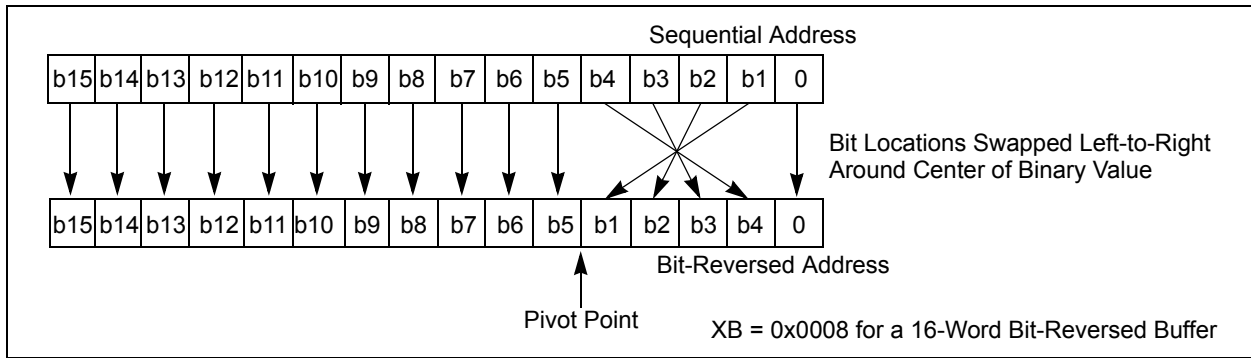


TABLE 4-41: BIT-REVERSED ADDRESS SEQUENCE (16-ENTRY)

| Normal Address | | | | | Bit-Reversed Address | | | | |
|----------------|----|----|----|---------|----------------------|----|----|----|---------|
| A3 | A2 | A1 | A0 | Decimal | A3 | A2 | A1 | A0 | Decimal |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 8 |
| 0 | 0 | 1 | 0 | 2 | 0 | 1 | 0 | 0 | 4 |
| 0 | 0 | 1 | 1 | 3 | 1 | 1 | 0 | 0 | 12 |
| 0 | 1 | 0 | 0 | 4 | 0 | 0 | 1 | 0 | 2 |
| 0 | 1 | 0 | 1 | 5 | 1 | 0 | 1 | 0 | 10 |
| 0 | 1 | 1 | 0 | 6 | 0 | 1 | 1 | 0 | 6 |
| 0 | 1 | 1 | 1 | 7 | 1 | 1 | 1 | 0 | 14 |
| 1 | 0 | 0 | 0 | 8 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 9 | 1 | 0 | 0 | 1 | 9 |
| 1 | 0 | 1 | 0 | 10 | 0 | 1 | 0 | 1 | 5 |
| 1 | 0 | 1 | 1 | 11 | 1 | 1 | 0 | 1 | 13 |
| 1 | 1 | 0 | 0 | 12 | 0 | 0 | 1 | 1 | 3 |
| 1 | 1 | 0 | 1 | 13 | 1 | 0 | 1 | 1 | 11 |
| 1 | 1 | 1 | 0 | 14 | 0 | 1 | 1 | 1 | 7 |
| 1 | 1 | 1 | 1 | 15 | 1 | 1 | 1 | 1 | 15 |

4.6 Interfacing Program and Data Memory Spaces

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 architecture uses a 24-bit-wide program space and a 16-bit-wide data space. The architecture is also a modified Harvard scheme, meaning that data can also be present in the program space. To use this data successfully, it must be accessed in a way that preserves the alignment of information in both spaces.

Aside from normal execution, the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 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 capability makes the method ideal for accessing data tables that need to be updated periodically. 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. The application can only access the least significant word of the program word.

4.6.1 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 Page 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 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.

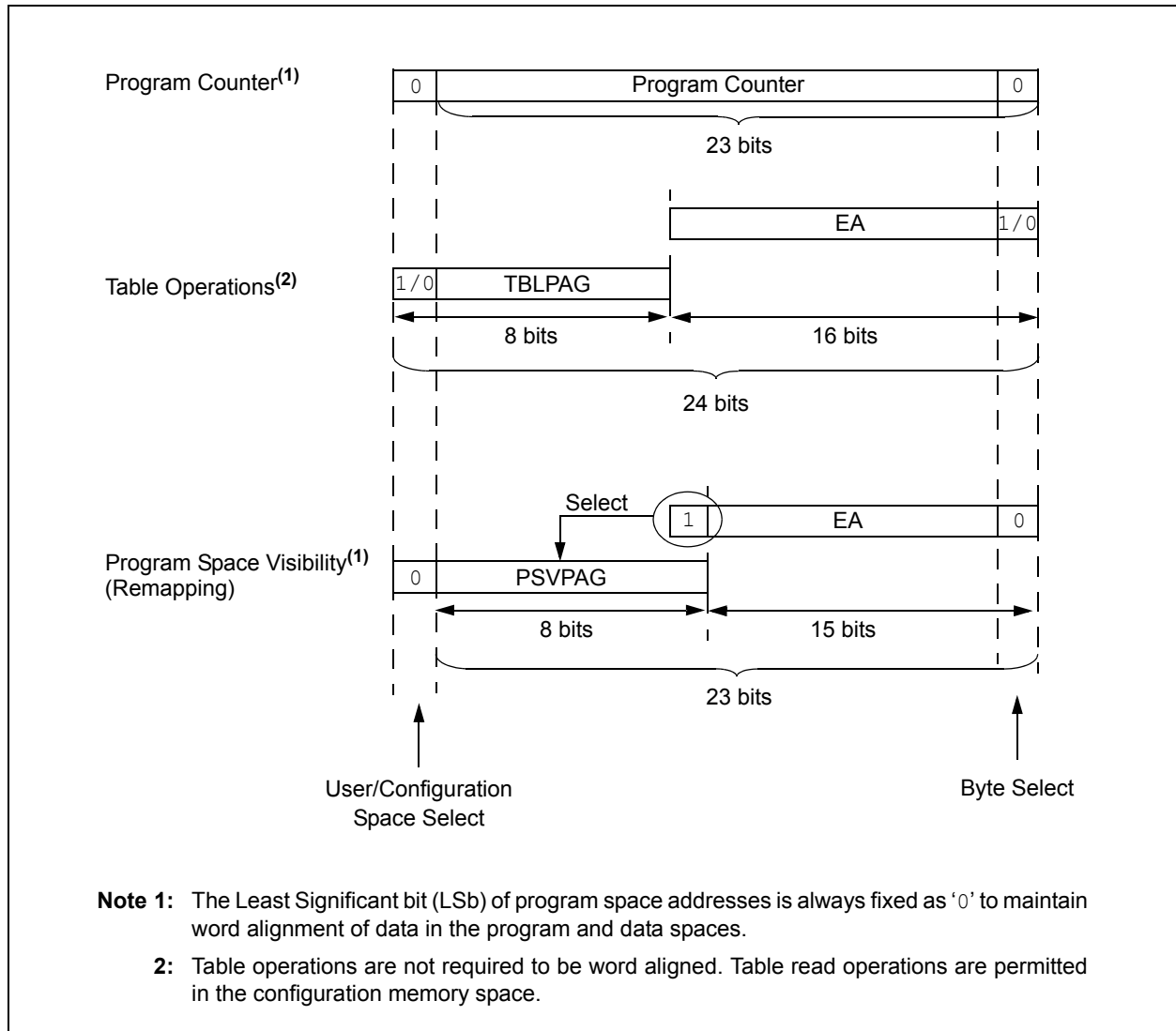
Table 4-42 and Figure 4-9 show how the program EA is created for table operations and remapping accesses from the data EA. Here, P<23:0> refers to a program space word, and D<15:0> refers to a data space word.

TABLE 4-42: PROGRAM SPACE ADDRESS CONSTRUCTION

| Access Type | Access Space | Program Space Address | | | | |
|--|---------------|-------------------------------|-------------|---------------|------------------------------|-----|
| | | <23> | <22:16> | <15> | <14:1> | <0> |
| Instruction Access (Code Execution) | User | 0 | PC<22:1> | | | 0 |
| | | 0xx xxxx xxxx xxxx xxxx xxx0 | | | | |
| TBLRD/TBLWT (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 (Block Remap/Read) | User | 0 | PSVPAG<7:0> | | Data EA<14:0> ⁽¹⁾ | |
| | | 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 4-9: DATA ACCESS FROM PROGRAM SPACE ADDRESS GENERATION



4.6.2 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-wide word address spaces, residing side by side, each with the same address range. TBLRDL and TBLWTL access the space that contains the least significant data word. TBLRDH and TBLWTH access the space that 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.

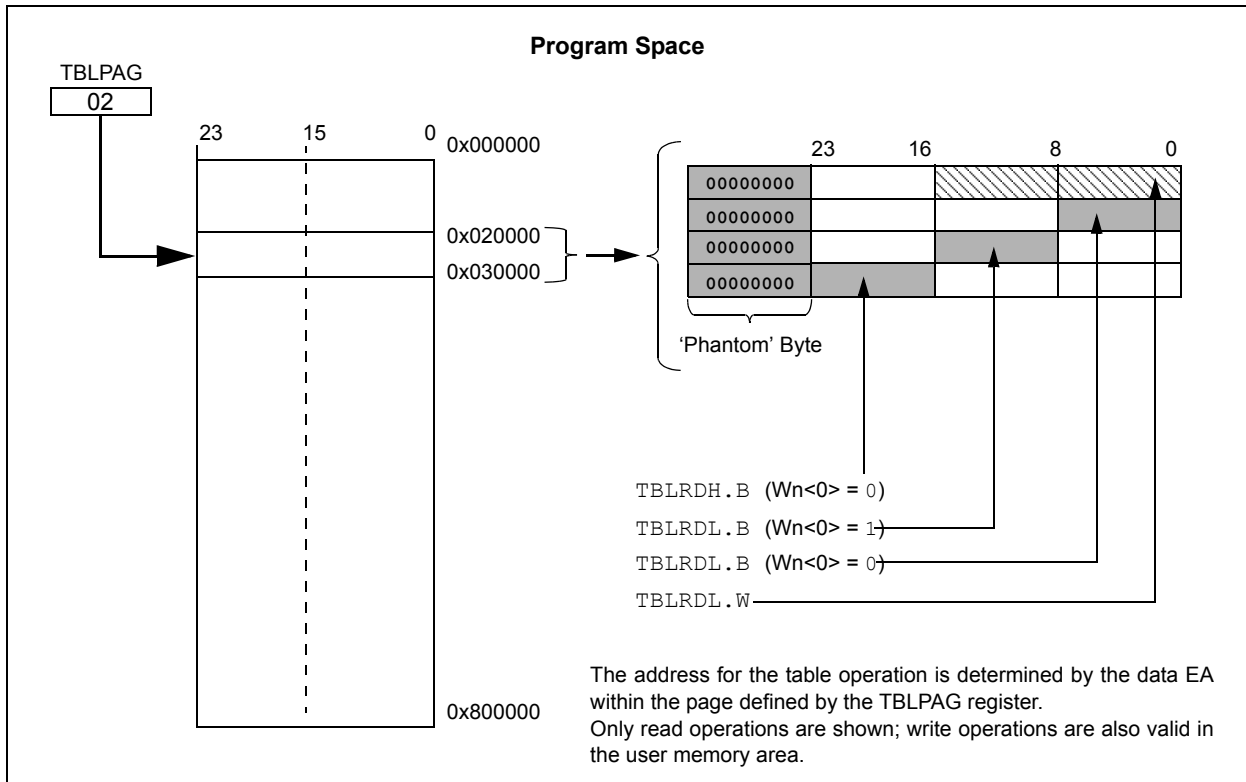
- TBLRDL (Table Read Low):
 - In Word mode, this instruction maps the lower word of the program space location (P<15:0>) to a data address (D<15:0>).

- 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'.
- TBLRDH (Table Read High):
 - In Word mode, this instruction maps the entire upper word of a program address (P<23:16>) to a data address. The 'phantom' byte (D<15:8>), is always '0'.
 - In Byte mode, this instruction maps the upper or lower byte of the program word to D<7:0> of the data address, in the TBLRDL instruction. The data is always '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 Page register (TBLPAG). TBLPAG covers the entire program memory space of the device, including user application 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.

FIGURE 4-10: ACCESSING PROGRAM MEMORY WITH TABLE INSTRUCTIONS



4.6.3 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 option provides transparent access to stored constant data from the data space without the need to use special instructions (such as 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 Core 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 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. 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 a 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 4-11), 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 location 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.

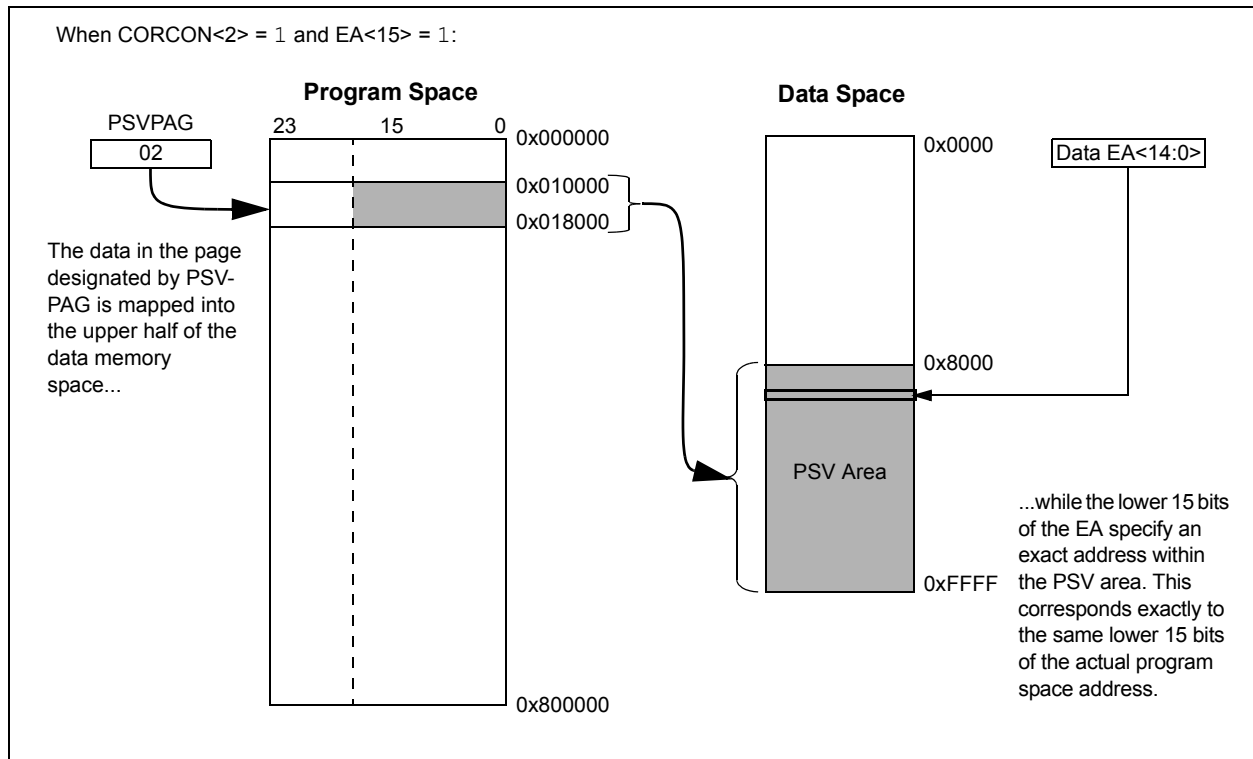
For operations that use PSV and are executed outside a REPEAT loop, the MOV and MOV.D instructions require one instruction cycle in addition to the specified execution time. All other instructions require two instruction cycles in addition to the specified execution time.

For operations that use PSV, and are executed inside a REPEAT loop, these instances 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 allows the instruction using PSV to access data, to execute in a single cycle.

FIGURE 4-11: PROGRAM SPACE VISIBILITY OPERATION



5.0 FLASH PROGRAM MEMORY

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 5. “Flash Programming” (DS70191), which is available from the Microchip website (www.microchip.com).

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices contain internal Flash program memory for storing and executing application code. The memory is readable, writable and erasable during normal operation over the entire VDD range.

Flash memory can be programmed in two ways:

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

ICSP allows a dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 device to be serially programmed while in the end application circuit. This is done with two lines for programming clock and programming data (one of the alternate programming pin pairs: PGECx/PGEDx), 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 digital signal controller 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 application can write program memory data either in blocks or ‘rows’ of 64 instructions (192 bytes) at a time or a single program memory word, and erase program memory in blocks or ‘pages’ of 512 instructions (1536 bytes) at a time.

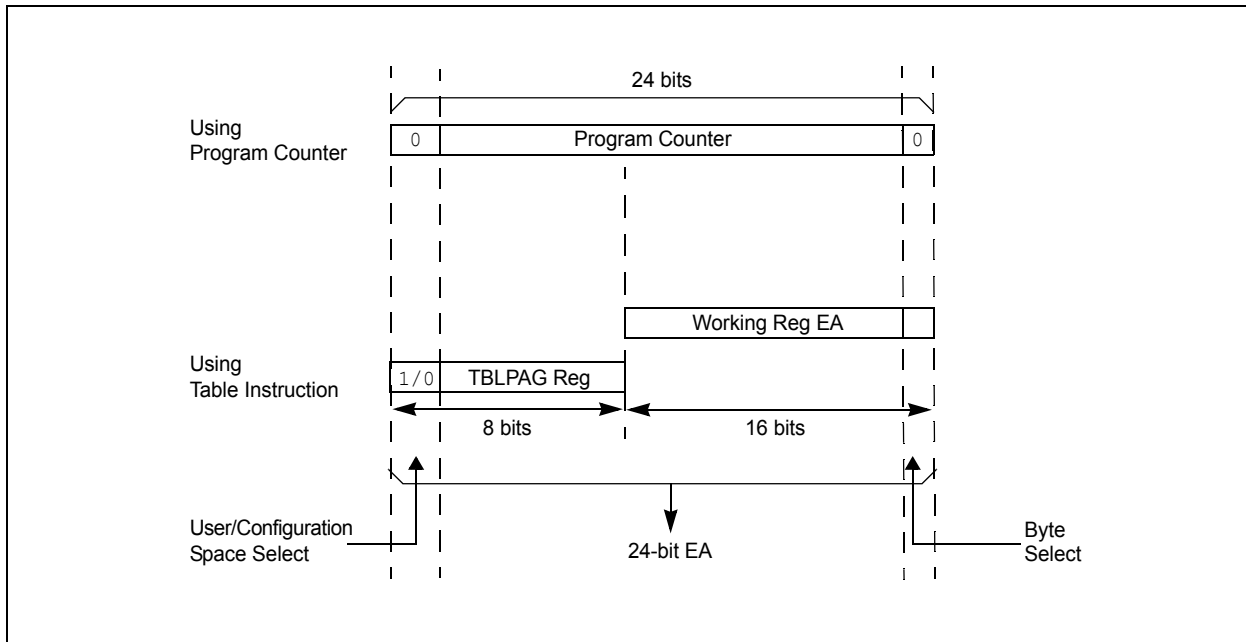
5.1 Table Instructions and Flash Programming

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 bits <7:0> of the TBLPAG register and the Effective Address (EA) from a W register specified in the table instruction, as shown in Figure 5-1.

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

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

FIGURE 5-1: ADDRESSING FOR TABLE REGISTERS



5.2 RTSP Operation

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 Flash program memory array is organized into rows of 64 instructions or 192 bytes. RTSP allows the user application to erase a page of memory, which consists of eight rows (512 instructions) at a time, and to program one row or one word at a time. Table 31-12 shows typical erase and programming times. The 8-row erase pages and single row write rows are edge-aligned from the beginning of program memory, on boundaries of 1536 bytes and 192 bytes, respectively.

The program memory implements holding buffers that can contain 64 instructions of programming data. Prior to the actual programming operation, the write data must be loaded into the buffers sequentially. The instruction words loaded must always be from a group of 64 boundary.

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. A total of 64 `TBLWTL` and `TBLWTH` instructions are required to load the instructions.

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

5.3 Programming Operations

A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. The processor stalls (waits) until the programming operation is finished.

The programming time depends on the FRC accuracy (see Table 31-19) and the value of the FRC Oscillator Tuning register (see Register 9-4). Use the following formula to calculate the minimum and maximum values for the Row Write Time, Page Erase Time, and Word Write Cycle Time parameters (see Table 31-12).

EQUATION 5-1: PROGRAMMING TIME

$$T = \frac{11064 \text{ Cycles}}{7.37 \text{ MHz} \times (\text{FRC Accuracy})\% \times (\text{FRC Tuning})\%}$$

For example, if the device is operating at +125°C, the FRC accuracy will be ±5%. If the `TUN<5:0>` bits (see Register 9-4) are set to `'b1111111`, the Minimum Row Write Time is:

$$T_{RW} = \frac{11064 \text{ Cycles}}{7.37 \text{ MHz} \times (1 + 0.05) \times (1 - 0.00375)} = 1.435 \text{ ms}$$

and, the Maximum Row Write Time is:

$$T_{RW} = \frac{11064 \text{ Cycles}}{7.37 \text{ MHz} \times (1 - 0.05) \times (1 - 0.00375)} = 1.586 \text{ ms}$$

Setting the `WR` bit (`NVMCON<15>`) starts the operation, and the `WR` bit is automatically cleared when the operation is finished.

5.4 Control Registers

Two SFRs are 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 the start of the programming cycle.

`NVMKEY` (Register 5-2) is a write-only register that is used for write protection. To start a programming or erase sequence, the user application must consecutively write `0x55` and `0xAA` to the `NVMKEY` register. Refer to **Section 5.3 “Programming Operations”** for further details.

REGISTER 5-1: NVMCON: FLASH MEMORY CONTROL REGISTER

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

| | | | | | | | |
|-------|----------------------|-----|-----|---------------------------|----------------------|----------------------|----------------------|
| U-0 | R/W-0 ⁽¹⁾ | U-0 | U-0 | R/W-0 ⁽¹⁾ | R/W-0 ⁽¹⁾ | R/W-0 ⁽¹⁾ | R/W-0 ⁽¹⁾ |
| — | ERASE | — | — | NVMOP<3:0> ⁽²⁾ | | | |
| bit 7 | | | | | | | bit 0 |

| | |
|-------------------|------------------------------------|
| Legend: | SO = Satable only 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 **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 NVMOP<3:0> on the next WR command
 - 0 = Perform the program operation specified by NVMOP<3:0> on the next WR command
- bit 5-4 **Unimplemented:** Read as '0'
- bit 3-0 **NVMOP<3:0>:** NVM Operation Select bits⁽²⁾
 - If ERASE = 1:
 - 1111 = Memory bulk erase operation
 - 1110 = Reserved
 - 1101 = Erase General Segment
 - 1100 = Erase Secure Segment
 - 1011 = Reserved
 - 0011 = No operation
 - 0010 = Memory page erase operation
 - 0001 = No operation
 - 0000 = Erase a single Configuration register byte
 - If ERASE = 0:
 - 1111 = No operation
 - 1110 = Reserved
 - 1101 = No operation
 - 1100 = No operation
 - 1011 = Reserved
 - 0011 = Memory word program operation
 - 0010 = No operation
 - 0001 = Memory row program operation
 - 0000 = Program a single Configuration register byte

Note 1: These bits can only be reset on POR.

2: All other combinations of NVMOP<3:0> are unimplemented.

REGISTER 5-2: NVMKEY: NONVOLATILE MEMORY KEY REGISTER

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

| | | | | | | | |
|-------------|-----|-----|-----|-----|-----|-----|-------|
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| NVMKEY<7:0> | | | | | | | |
| 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 15-8 **Unimplemented:** Read as '0'
bit 7-0 **NVMKEY<7:0>:** Key Register (write-only) bits

5.4.1 PROGRAMMING ALGORITHM FOR FLASH PROGRAM MEMORY

Programmers can program one row of program Flash memory at a time. To do this, it is necessary to erase the 8-row erase page that contains 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 page to be erased into the TBLPAG and W registers.
 - c) Write 0x55 to NVMKEY.
 - d) Write 0xAA 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 5-2).
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 0x55 to NVMKEY.
 - c) Write 0xAA 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 application 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 PAGE

```

; Set up NVMCON for block erase operation
MOV    #0x4042, W0          ;
MOV    W0, NVMCON          ; Initialize NVMCON
; Init pointer to row to be ERASED
MOV    #tblpage(PROG_ADDR), W0 ;
MOV    W0, TBLPAG          ; Initialize PM Page Boundary SFR
MOV    #tbloffset(PROG_ADDR), W0 ; Initialize in-page EA[15:0] pointer
TBLWTL W0, [W0]           ; Set base address of erase block
DISI   #5                  ; Block all interrupts with priority <7
                                     ; for next 5 instructions

MOV    #0x55, W0
MOV    W0, NVMKEY          ; Write the 55 key
MOV    #0xAA, W1          ;
MOV    W1, NVMKEY          ; Write the AA key
BSET   NVMCON, #WR        ; Start the erase sequence
NOP    ; Insert two NOPs after the erase
NOP    ; command is asserted
    
```

EXAMPLE 5-2: 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
NOP                          ; Insert two NOPs after the
NOP                          ; erase command is asserted
```

6.0 RESETS

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 8. “Reset” (DS70192), which is available from the Microchip website (www.microchip.com).

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
- BOR: Brown-out Reset
- $\overline{\text{MCLR}}$: Master Clear Pin Reset
- SWR: `RESET` Instruction
- WDTO: Watchdog Timer Reset
- CM: Configuration Mismatch Reset
- TRAPR: Trap Conflict Reset
- IOPUWR: Illegal Condition Device Reset
 - Illegal Opcode Reset
 - Uninitialized W Register Reset
 - Security 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. On system Reset, some of the registers associated with the CPU and peripherals are forced to a known Reset state and some are unaffected.

Note: Refer to the specific peripheral section or Section 3.0 “CPU” of this manual for register Reset states.

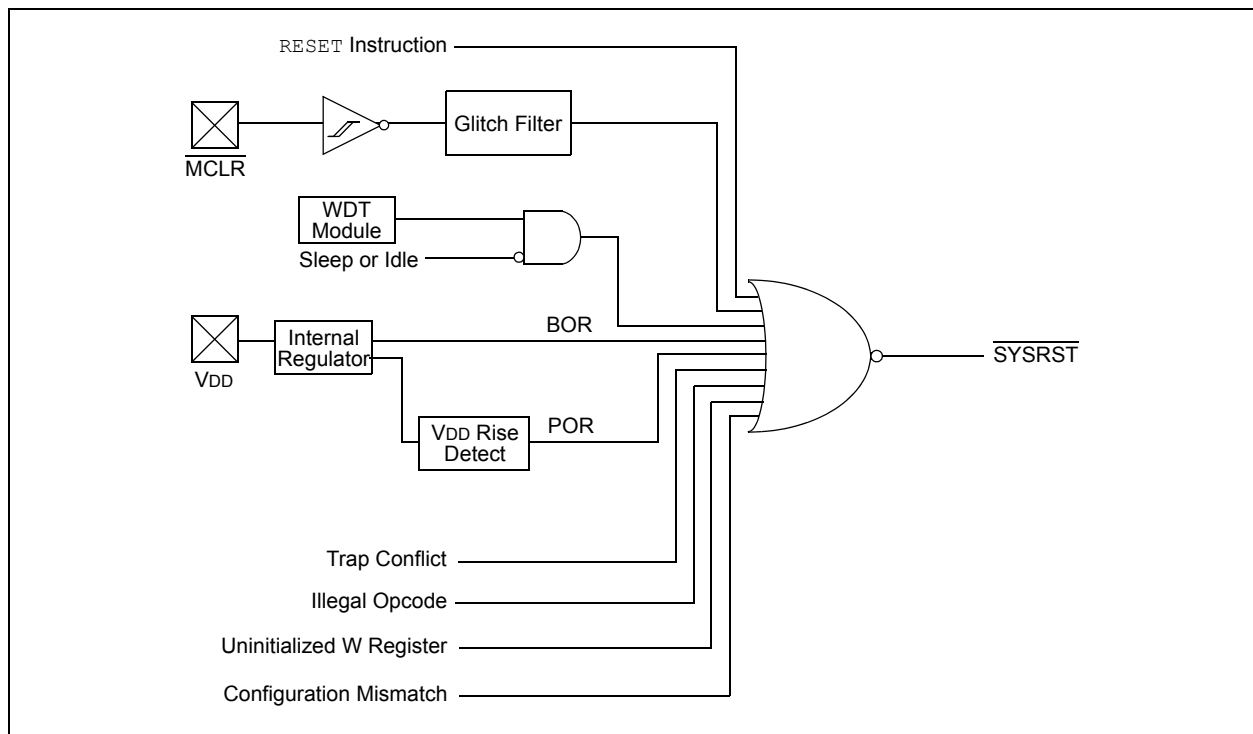
All types of device Reset sets a corresponding status bit in the RCON register to indicate the type of Reset (see Register 6-1).

A POR clears all the bits, except for the POR bit (`RCON<0>`), that are set. The user application can 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 does 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 is meaningful.

FIGURE 6-1: RESET SYSTEM BLOCK DIAGRAM



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

| | | | | | | | |
|--------|--------|-----|-----|-----|-----|-------|-------|
| R/W-0 | R/W-0 | U-0 | U-0 | U-0 | 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 ⁽²⁾ | 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 Mismatch Flag bit
 1 = A configuration mismatch Reset has occurred
 0 = A configuration mismatch Reset has NOT occurred
- bit 8 **VREGS:** Voltage Regulator Standby During Sleep bit
 1 = Voltage regulator is active during Sleep
 0 = Voltage regulator goes into Standby mode during Sleep
- bit 7 **EXTR:** External Reset ($\overline{\text{MCLR}}$) Pin bit
 1 = A Master Clear (pin) Reset has occurred
 0 = A Master Clear (pin) Reset has not occurred
- bit 6 **SWR:** Software Reset (Instruction) Flag bit
 1 = A `RESET` instruction has been executed
 0 = A `RESET` instruction has not been executed
- bit 5 **SWDTEN:** Software Enable/Disable of WDT bit⁽²⁾
 1 = WDT is enabled
 0 = WDT is disabled
- bit 4 **WDTO:** Watchdog Timer Time-out Flag bit
 1 = WDT time-out has occurred
 0 = WDT time-out has not occurred
- bit 3 **SLEEP:** Wake-up 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 was in Idle mode
 0 = Device was not in Idle mode

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

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

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

- bit 1 **BOR:** Brown-out Reset Flag bit
 1 = A Brown-out Reset has occurred
 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 can be set or cleared in software. Setting one of these bits in software does not cause a device Reset.
- 2:** If the FWDTEN Configuration bit is '1' (unprogrammed), the WDT is always enabled, regardless of the SWDTEN bit setting.

6.1 System Reset

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 family of devices have two types of Reset:

- Cold Reset
- Warm Reset

A cold Reset is the result of a Power-on Reset (POR) or a Brown-out Reset (BOR). On a cold Reset, the FNOSC configuration bits in the FOSC device configuration register selects the device clock source.

A warm Reset is the result of all other reset sources, including the `RESET` instruction. On warm Reset, the device will continue to operate from the current clock source as indicated by the Current Oscillator Selection (COSC<2:0>) bits in the Oscillator Control (OSCCON<14:12>) register.

The device is kept in a Reset state until the system power supplies have stabilized at appropriate levels and the oscillator clock is ready. The sequence in which this occurs is detailed below and is shown in Figure 6-2.

1. **POR Reset:** A POR circuit holds the device in Reset when the power supply is turned on. The POR circuit is active until VDD crosses the VPOR threshold and the delay TPOR has elapsed.
2. **BOR Reset:** The on-chip voltage regulator has a BOR circuit that keeps the device in Reset until VDD crosses the VBOR threshold and the delay TBOR has elapsed. The delay TBOR ensures that the voltage regulator output becomes stable.
3. **PWRT Timer:** The programmable power-up timer continues to hold the processor in Reset for a specific period of time (TPWRT) after a BOR. The delay TPWRT ensures that the system power supplies have stabilized at the appropriate level for full-speed operation. After the delay TPWRT has elapsed, the SYSRST becomes inactive, which in turn enables the selected oscillator to start generating clock cycles.
4. **Oscillator Delay:** The total delay for the clock to be ready for various clock source selections is given in Table 6-1. Refer to **Section 9.0 “Oscillator Configuration”** for more information.
5. When the oscillator clock is ready, the processor begins execution from location 0x000000. The user application programs a `GOTO` instruction at the reset address, which redirects program execution to the appropriate start-up routine.
6. The Fail-Safe Clock Monitor (FSCM), if enabled, begins to monitor the system clock when the system clock is ready and the delay TFSCM elapsed.

TABLE 6-1: OSCILLATOR DELAY

| Oscillator Mode | Oscillator Startup Delay | Oscillator Startup Timer | PLL Lock Time | Total Delay |
|------------------------|--------------------------|--------------------------|---------------|----------------------|
| FRC, FRCDIV16, FRCDIVN | TOSCD | — | — | TOSCD |
| FRCPLL | TOSCD | — | TLOCK | TOSCD + TLOCK |
| XT | TOSCD | TOST | — | TOSCD + TOST |
| HS | TOSCD | TOST | — | TOSCD + TOST |
| EC | — | — | — | — |
| XTPLL | TOSCD | TOST | TLOCK | TOSCD + TOST + TLOCK |
| HSPLL | TOSCD | TOST | TLOCK | TOSCD + TOST + TLOCK |
| ECPLL | — | — | TLOCK | TLOCK |
| SOSC | TOSCD | TOST | — | TOSCD + TOST |
| LPRC | TOSCD | — | — | TOSCD |

- Note 1:** TOSCD = Oscillator Start-up Delay (1.1 μs max for FRC, 70 μs max for LPRC). Crystal Oscillator start-up times vary with crystal characteristics, load capacitance, etc.
- 2:** TOST = Oscillator Start-up Timer Delay (1024 oscillator clock period). For example, TOST = 102.4 μs for a 10 MHz crystal and TOST = 32 ms for a 32 kHz crystal.
- 3:** TLOCK = PLL lock time (1.5 ms nominal), if PLL is enabled.

FIGURE 6-2: SYSTEM RESET TIMING

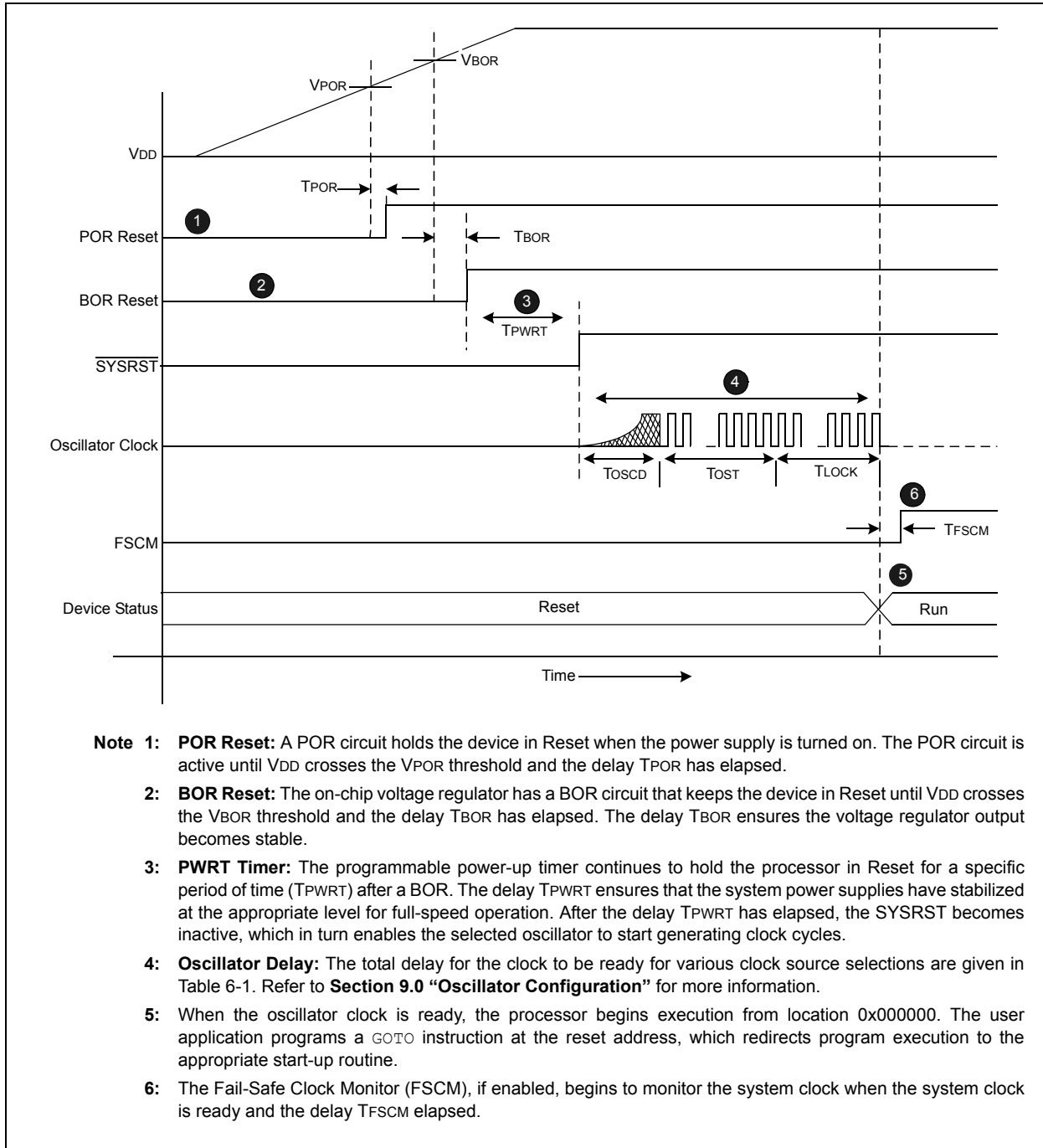


TABLE 6-2: OSCILLATOR DELAY

| Symbol | Parameter | Value |
|--------|----------------------------------|---------------------|
| VPOR | POR threshold | 1.8V nominal |
| TPOR | POR extension time | 30 μ s maximum |
| VBOR | BOR threshold | 2.5V nominal |
| TBOR | BOR extension time | 100 μ s maximum |
| TPWRT | Programmable power-up time delay | 0-128 ms nominal |
| TFSCM | Fail-Safe Clock Monitor Delay | 900 μ s maximum |

Note: When the device exits the Reset condition (begins normal operation), the device operating parameters (voltage, frequency, temperature, etc.) must be within their operating ranges, otherwise the device may not function correctly. The user application must ensure that the delay between the time power is first applied, and the time SYSRST becomes inactive, is long enough to get all operating parameters within specification.

6.2 Power-on Reset (POR)

A Power-on Reset (POR) circuit ensures the device is reset from power-on. The POR circuit is active until VDD crosses the VPOR threshold and the delay TPOR has elapsed. The delay TPOR ensures the internal device bias circuits become stable.

The device supply voltage characteristics must meet the specified starting voltage and rise rate requirements to generate the POR. Refer to **Section 31.0 “Electrical Characteristics”** for details.

The POR status (POR) bit in the Reset Control (RCON<0>) register is set to indicate the Power-on Reset.

6.2.1 Brown-out Reset (BOR) and Power-up timer (PWRT)

The on-chip regulator has a Brown-out Reset (BOR) circuit that resets the device when the VDD is too low ($V_{DD} < V_{BOR}$) for proper device operation. The BOR circuit keeps the device in Reset until VDD crosses VBOR threshold and the delay TBOR has elapsed. The delay TBOR ensures the voltage regulator output becomes stable.

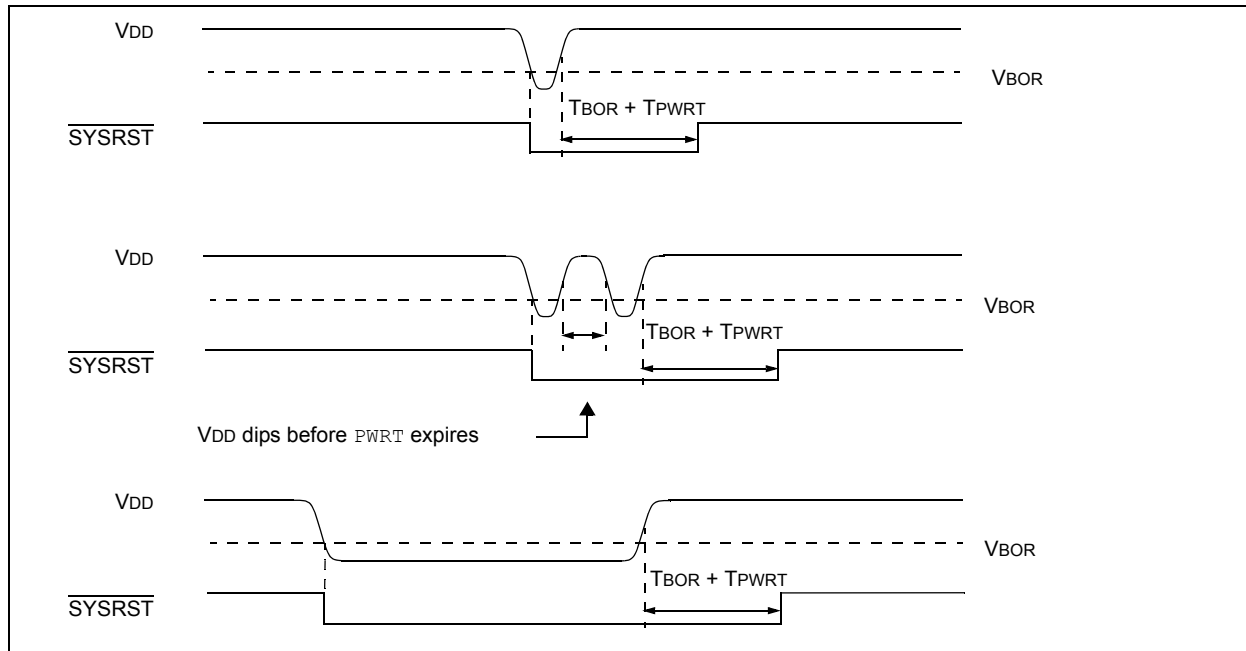
The BOR status (BOR) bit in the Reset Control (RCON<1>) register is set to indicate the Brown-out Reset.

The device will not run at full speed after a BOR as the VDD should rise to acceptable levels for full-speed operation. The PWRT provides power-up time delay (TPWRT) to ensure that the system power supplies have stabilized at the appropriate levels for full-speed operation before the SYSRST is released.

The power-up timer delay (TPWRT) is programmed by the Power-on Reset Timer Value Select (FPWRT<2:0>) bits in the POR Configuration (FPOR<2:0>) register, which provides eight settings (from 0 ms to 128 ms). Refer to **Section 28.0 “Special Features”** for further details.

Figure 6-3 shows the typical brown-out scenarios. The reset delay (TBOR + TPWRT) is initiated each time VDD rises above the VBOR trip point

FIGURE 6-3: BROWN-OUT SITUATIONS



6.3 External Reset (EXTR)

The external Reset is generated by driving the $\overline{\text{MCLR}}$ pin low. The $\overline{\text{MCLR}}$ pin is a Schmitt trigger input with an additional glitch filter. Reset pulses that are longer than the minimum pulse width will generate a Reset. Refer to **Section 31.0 “Electrical Characteristics”** for minimum pulse width specifications. The External Reset ($\overline{\text{MCLR}}$) Pin (EXTR) bit in the Reset Control (RCON) register is set to indicate the $\overline{\text{MCLR}}$ Reset.

6.3.0.1 EXTERNAL SUPERVISORY CIRCUIT

Many systems have external supervisory circuits that generate reset signals to Reset multiple devices in the system. This external Reset signal can be directly connected to the $\overline{\text{MCLR}}$ pin to Reset the device when the rest of system is Reset.

6.3.0.2 INTERNAL SUPERVISORY CIRCUIT

When using the internal power supervisory circuit to Reset the device, the external reset pin ($\overline{\text{MCLR}}$) should be tied directly or resistively to VDD. In this case, the $\overline{\text{MCLR}}$ pin will not be used to generate a Reset. The external reset pin ($\overline{\text{MCLR}}$) does not have an internal pull-up and must not be left unconnected.

6.4 Software RESET Instruction (SWR)

Whenever the `RESET` instruction is executed, the device will assert $\overline{\text{SYSRST}}$, placing the device in a special Reset state. This Reset state will not reinitialize the clock. The clock source in effect prior to the `RESET` instruction will remain. $\overline{\text{SYSRST}}$ is released at the next instruction cycle, and the reset vector fetch will commence.

The Software Reset (Instruction) Flag (SWR) bit in the Reset Control (RCON<6>) register is set to indicate the software Reset.

6.5 Watchdog Time-out Reset (WDTO)

Whenever a Watchdog time-out occurs, the device will asynchronously assert $\overline{\text{SYSRST}}$. The clock source will remain unchanged. A WDT time-out during Sleep or Idle mode will wake-up the processor, but will not reset the processor.

The Watchdog Timer Time-out Flag (WDTO) bit in the Reset Control (RCON<4>) register is set to indicate the Watchdog Reset. Refer to **Section 28.4 “Watchdog Timer (WDT)”** for more information on Watchdog Reset.

6.6 Trap Conflict Reset

If a lower-priority hard trap occurs while a higher-priority trap is being processed, a hard trap conflict Reset occurs. The hard traps include exceptions of priority level 13 through level 15, inclusive. The address error (level 13) and oscillator error (level 14) traps fall into this category.

The Trap Reset Flag (TRAPR) bit in the Reset Control (RCON<15>) register is set to indicate the Trap Conflict Reset. Refer to **Section 7.0 “Interrupt Controller”** for more information on trap conflict Resets.

6.7 Configuration Mismatch Reset

To maintain the integrity of the peripheral pin select control registers, they are constantly monitored with shadow registers in hardware. If an unexpected change in any of the registers occur (such as cell disturbances caused by ESD or other external events), a configuration mismatch Reset occurs.

The Configuration Mismatch Flag (CM) bit in the Reset Control (RCON<9>) register is set to indicate the configuration mismatch Reset. Refer to **Section 11.0 “I/O Ports”** for more information on the configuration mismatch Reset.

Note: The configuration mismatch feature and associated reset flag is not available on all devices.

6.8 Illegal Condition Device Reset

An illegal condition device Reset occurs due to the following sources:

- Illegal Opcode Reset
- Uninitialized W Register Reset
- Security Reset

The Illegal Opcode or Uninitialized W Access Reset Flag (IOPUWR) bit in the Reset Control (RCON<14>) register is set to indicate the illegal condition device Reset.

6.8.0.1 ILLEGAL OPCODE RESET

A device Reset is generated if the device attempts to execute an illegal opcode value that is fetched from program memory.

The illegal opcode Reset function can prevent the device from executing program memory sections that are used to store constant data. To take advantage of the illegal opcode Reset, use only the lower 16 bits of

each program memory section to store the data values. The upper 8 bits should be programmed with 3Fh, which is an illegal opcode value.

6.8.0.2 UNINITIALIZED W REGISTER RESET

Any attempts to use the uninitialized W register as an address pointer will Reset the device. The W register array (with the exception of W15) is cleared during all resets and is considered uninitialized until written to.

6.8.0.3 SECURITY RESET

If a Program Flow Change (PFC) or Vector Flow Change (VFC) targets a restricted location in a protected segment (Boot and Secure Segment), that operation will cause a security Reset.

The PFC occurs when the Program Counter is reloaded as a result of a Call, Jump, Computed Jump, Return, Return from Subroutine, or other form of branch instruction.

The VFC occurs when the Program Counter is reloaded with an Interrupt or Trap vector.

Refer to **Section 28.8 “Code Protection and CodeGuard Security”** for more information on Security Reset.

6.9 Using the RCON Status Bits

The user application can read the Reset Control (RCON) register after any device Reset to determine the cause of the reset.

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.

Table 6-3 provides a summary of the reset flag bit operation.

TABLE 6-3: RESET FLAG BIT OPERATION

| Flag Bit | Set by: | Cleared by: |
|------------------|---|--|
| TRAPR (RCON<15>) | Trap conflict event | POR,BOR |
| IOPWR (RCON<14>) | Illegal opcode or uninitialized W register access or Security Reset | POR,BOR |
| CM (RCON<9>) | Configuration Mismatch | POR,BOR |
| EXTR (RCON<7>) | MCLR Reset | POR |
| SWR (RCON<6>) | RESET instruction | POR,BOR |
| WDTO (RCON<4>) | WDT time-out | PWRSV instruction, CLRWDT instruction, POR,BOR |
| SLEEP (RCON<3>) | PWRSV #SLEEP instruction | POR,BOR |
| IDLE (RCON<2>) | PWRSV #IDLE instruction | POR,BOR |
| BOR (RCON<1>) | POR, BOR | |
| POR (RCON<0>) | POR | |

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

7.0 INTERRUPT CONTROLLER

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 32. “Interrupts (Part III)” (DS70214), which is available from the Microchip website (www.microchip.com).

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 interrupt controller reduces the numerous peripheral interrupt request signals to a single interrupt request signal to the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 CPU.

The interrupt controller has the following features:

- Up to eight processor exceptions and software traps
- Eight 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), shown in Figure 7-1, resides in program memory, starting at location 000004h. The IVT contains 126 vectors consisting of eight nonmaskable 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 priority is linked to their position in the vector table. Lower addresses generally have a higher natural priority. For example, the interrupt associated with vector 0 takes priority over interrupts at any other vector address.

dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices implement up to 53 unique interrupts and five nonmaskable traps. These are summarized in Table 7-1.

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 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 debugging 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 dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 device clears its registers in response to a Reset, which forces the PC to zero. The digital signal controller then begins program execution at location 0x000000. A GOTO instruction at the Reset address can redirect program execution to the appropriate start-up routine.

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

FIGURE 7-1: dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, AND dsPIC33FJ128MCX02/X04 INTERRUPT VECTOR TABLE

| | | | |
|--|-----------------------------|----------|--|
| | Reset – GOTO Instruction | 0x000000 | |
| | Reset – GOTO Address | 0x000002 | |
| | Reserved | 0x000004 | |
| | Oscillator Fail Trap Vector | | |
| | Address Error Trap Vector | | |
| | Stack Error Trap Vector | | |
| | Math Error Trap Vector | | |
| | DMA Error Trap Vector | | |
| | Reserved | | |
| | Reserved | | |
| | Interrupt Vector 0 | 0x000014 | Interrupt Vector Table (IVT) ⁽¹⁾ |
| | Interrupt Vector 1 | | |
| | ~ | | |
| | ~ | | |
| | ~ | | |
| | Interrupt Vector 52 | 0x00007C | |
| | Interrupt Vector 53 | 0x00007E | |
| | Interrupt Vector 54 | 0x000080 | |
| | ~ | | |
| | ~ | | |
| | Interrupt Vector 116 | 0x0000FC | Alternate Interrupt Vector Table (AIVT) ⁽¹⁾ |
| | Interrupt Vector 117 | 0x0000FE | |
| | Reserved | 0x000100 | |
| | Reserved | 0x000102 | |
| | Reserved | | |
| | Oscillator Fail Trap Vector | | |
| | Address Error Trap Vector | | |
| | Stack Error Trap Vector | | |
| | Math Error Trap Vector | | |
| | DMA Error Trap Vector | | |
| | Reserved | | |
| | Reserved | | |
| | Interrupt Vector 0 | 0x000114 | |
| | Interrupt Vector 1 | | |
| | ~ | | |
| | ~ | | |
| | ~ | | |
| | Interrupt Vector 52 | 0x00017C | |
| | Interrupt Vector 53 | 0x00017E | |
| | Interrupt Vector 54 | 0x000180 | |
| | ~ | | |
| | ~ | | |
| | ~ | | |
| | Interrupt Vector 116 | | |
| | Interrupt Vector 117 | 0x0001FE | |
| | Start of Code | 0x000200 | |

Decreasing Natural Order Priority

Note 1: See Table 7-1 for the list of implemented interrupt vectors.

TABLE 7-1: INTERRUPT VECTORS

| Vector Number | IVT Address | AIVT Address | Interrupt Source |
|---------------|-------------|--------------|-------------------------------|
| 0 | 0x000004 | 0x000104 | Reserved |
| 1 | 0x000006 | 0x000106 | Oscillator Failure |
| 2 | 0x000008 | 0x000108 | Address Error |
| 3 | 0x00000A | 0x00010A | Stack Error |
| 4 | 0x00000C | 0x00010C | Math Error |
| 5 | 0x00000E | 0x00010E | DMA Error |
| 6 | 0x000010 | 0x000110 | Reserved |
| 7 | 0x000012 | 0x000112 | Reserved |
| 8 | 0x000014 | 0x000114 | INT0 – External Interrupt 0 |
| 9 | 0x000016 | 0x000116 | IC1 – Input Compare 1 |
| 10 | 0x000018 | 0x000118 | OC1 – Output Compare 1 |
| 11 | 0x00001A | 0x00011A | T1 – Timer1 |
| 12 | 0x00001C | 0x00011C | DMA0 – DMA Channel 0 |
| 13 | 0x00001E | 0x00011E | IC2 – Input Capture 2 |
| 14 | 0x000020 | 0x000120 | OC2 – Output Compare 2 |
| 15 | 0x000022 | 0x000122 | T2 – Timer2 |
| 16 | 0x000024 | 0x000124 | T3 – Timer3 |
| 17 | 0x000026 | 0x000126 | SPI1E – SPI1 Error |
| 18 | 0x000028 | 0x000128 | SPI1 – SPI1 Transfer Done |
| 19 | 0x00002A | 0x00012A | U1RX – UART1 Receiver |
| 20 | 0x00002C | 0x00012C | U1TX – UART1 Transmitter |
| 21 | 0x00002E | 0x00012E | ADC1 – ADC 1 |
| 22 | 0x000030 | 0x000130 | DMA1 – DMA Channel 1 |
| 23 | 0x000032 | 0x000132 | Reserved |
| 24 | 0x000034 | 0x000134 | SI2C1 – I2C1 Slave Events |
| 25 | 0x000036 | 0x000136 | MI2C1 – I2C1 Master Events |
| 26 | 0x000038 | 0x000138 | CM – Comparator Interrupt |
| 27 | 0x00003A | 0x00013A | Change Notification Interrupt |
| 28 | 0x00003C | 0x00013C | INT1 – External Interrupt 1 |
| 29 | 0x00003E | 0x00013E | Reserved |
| 30 | 0x000040 | 0x000140 | IC7 – Input Capture 7 |
| 31 | 0x000042 | 0x000142 | IC8 – Input Capture 8 |
| 32 | 0x000044 | 0x000144 | DMA2 – DMA Channel 2 |
| 33 | 0x000046 | 0x000146 | OC3 – Output Compare 3 |
| 34 | 0x000048 | 0x000148 | OC4 – Output Compare 4 |
| 35 | 0x00004A | 0x00014A | T4 – Timer4 |
| 36 | 0x00004C | 0x00014C | T5 – Timer5 |
| 37 | 0x00004E | 0x00014E | INT2 – External Interrupt 2 |
| 38 | 0x000050 | 0x000150 | U2RX – UART2 Receiver |
| 39 | 0x000052 | 0x000152 | U2TX – UART2 Transmitter |
| 40 | 0x000054 | 0x000154 | SPI2E – SPI2 Error |
| 41 | 0x000056 | 0x000156 | SPI2 – SPI2 Transfer Done |
| 42 | 0x000058 | 0x000158 | C1RX – ECAN1 RX Data Ready |
| 43 | 0x00005A | 0x00015A | C1 – ECAN1 Event |
| 44 | 0x00005C | 0x00015C | DMA3 – DMA Channel 3 |
| 45 | 0x00005E | 0x00015E | Reserved |
| 46 | 0x000060 | 0x000160 | Reserved |

TABLE 7-1: INTERRUPT VECTORS (CONTINUED)

| Vector Number | IVT Address | AIVT Address | Interrupt Source |
|---------------|-------------------|-------------------|---------------------------------|
| 47 | 0x000062 | 0x000162 | Reserved |
| 48 | 0x000064 | 0x000164 | Reserved |
| 49 | 0x000066 | 0x000166 | Reserved |
| 50 | 0x000068 | 0x000168 | Reserved |
| 51 | 0x00006A | 0x00016A | Reserved |
| 52 | 0x00006C | 0x00016C | Reserved |
| 53 | 0x00006E | 0x00016E | PMP – Parallel Master Port |
| 54 | 0x000070 | 0x000170 | DMA – DMA Channel 4 |
| 55 | 0x000072 | 0x000172 | Reserved |
| 56 | 0x000074 | 0x000174 | Reserved |
| 57 | 0x000076 | 0x000176 | Reserved |
| 58 | 0x000078 | 0x000178 | Reserved |
| 59 | 0x00007A | 0x00017A | Reserved |
| 60 | 0x00007C | 0x00017C | Reserved |
| 61 | 0x00007E | 0x00017E | Reserved |
| 62 | 0x000080 | 0x000180 | Reserved |
| 63 | 0x000082 | 0x000182 | Reserved |
| 64 | 0x000084 | 0x000184 | Reserved |
| 65 | 0x000086 | 0x000186 | PWM1 – PWM1 Period Match |
| 66 | 0x000088 | 0x000188 | QE11 – Position Counter Compare |
| 67 | 0x00008A | 0x00018A | Reserved |
| 68 | 0x00008C | 0x00018C | Reserved |
| 69 | 0x00008E | 0x00018E | DMA5 – DMA Channel 5 |
| 70 | 0x000090 | 0x000190 | RTCC – Real Time Clock |
| 71 | 0x000092 | 0x000192 | FLTA1 – PWM1 Fault A |
| 72 | 0x000094 | 0x000194 | Reserved |
| 73 | 0x000096 | 0x000196 | U1E – UART1 Error |
| 74 | 0x000098 | 0x000198 | U2E – UART2 Error |
| 75 | 0x00009A | 0x00019A | CRC – CRC Generator Interrupt |
| 76 | 0x00009C | 0x00019C | DMA6 – DMA Channel 6 |
| 77 | 0x00009E | 0x00019E | DMA7 – DMA Channel 7 |
| 78 | 0x0000A0 | 0x0001A0 | C1TX – ECAN1 TX Data Request |
| 79 | 0x0000A2 | 0x0001A2 | Reserved |
| 80 | 0x0000A4 | 0x0001A4 | Reserved |
| 81 | 0x0000A6 | 0x0001A6 | PWM2 – PWM2 Period Match |
| 82 | 0x0000A8 | 0x0001A8 | FLTA2 – PWM2 Fault A |
| 83 | 0x0000AA | 0x0001AA | QEI2 – Position Counter Compare |
| 84 | 0x0000AC | 0x0001AC | Reserved |
| 85 | 0x0000AE | 0x0001AE | Reserved |
| 86 | 0x0000B0 | 0x0001B0 | DAC1R – DAC1 Right Data Request |
| 87 | 0x0000B2 | 0x0001B2 | DAC1L – DAC1 Left Data Request |
| 88-126 | 0x0000B4-0x0000FE | 0x0001B4-0x0001FE | Reserved |

7.3 Interrupt Control and Status Registers

dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices implement a total of 30 registers for the interrupt controller:

- INTCON1
- INTCON2
- IFSx
- IECx
- IPCx
- INTTREG

7.3.1 INTCON1 AND INTCON2

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 (AIVT).

7.3.2 IFSx

The IFS 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.

7.3.3 IECx

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

7.3.4 IPCx

The IPC 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.

7.3.5 INTTREG

The INTTREG register contains the associated interrupt vector number and the new CPU interrupt priority level, which are latched into vector number (VECNUM<6:0>) and Interrupt level (ILR<3:0>) bit fields in the INTTREG register. The new interrupt priority level is the priority of the pending interrupt.

The interrupt sources are assigned to the IFSx, IECx and IPCx registers in the same sequence that they are listed in Table 7-1. For example, the INT0 (External Interrupt 0) is shown as having vector number 8 and a natural order priority of 0. Thus, the INTOIF bit is found in IFS0<0>, the INTOIE bit in IEC0<0>, and the INTOIP bits in the first position of IPC0 (IPC0<2:0>).

7.3.6 STATUS/CONTROL REGISTERS

Although they are not specifically part of the interrupt control hardware, two of the CPU Control registers contain bits that control interrupt functionality.

- The CPU STATUS register, SR, contains the IPL<2:0> bits (SR<7:5>). These bits indicate the current CPU interrupt priority level. The user software can change the current CPU priority level by writing to the IPL bits.
- The CORCON register contains the IPL3 bit which, together with IPL<2:0>, 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 7-32 in the following pages.

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

| | | | | | | | |
|----------------------|----------------------|----------------------|-------|-------|-------|-------|-------|
| R-0 | R-0 | R/C-0 | R/C-0 | R-0 | R/C-0 | R-0 | R/W-0 |
| OA | OB | SA | SB | OAB | SAB | DA | DC |
| bit 15 | | | | | | bit 8 | |
| R/W-0 ⁽³⁾ | R/W-0 ⁽³⁾ | R/W-0 ⁽³⁾ | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| IPL2 ⁽²⁾ | IPL1 ⁽²⁾ | IPL0 ⁽²⁾ | RA | N | OV | Z | C |
| bit 7 | | | | | | bit 0 | |

Legend:

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

bit 7-5 **IPL<2:0>**: CPU Interrupt Priority Level Status bits⁽²⁾

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: For complete register details, see **Register 3-1: "SR: CPU STATUS Register"**.

2: The IPL<2:0> bits are concatenated with the IPL<3> bit (CORCON<3>) to form the CPU Interrupt Priority Level. The value in parentheses indicates the IPL if IPL<3> = 1. User interrupts are disabled when IPL<3> = 1.

3: The IPL<2:0> Status bits are read-only when NSTDIS (INTCON1<15>) = 1.

REGISTER 7-2: CORCON: CORE CONTROL REGISTER⁽¹⁾

| | | | | | | | |
|--------|-------|-------|--------|---------------------|---------|-------|-------|
| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R-0 | R-0 | R-0 |
| — | — | — | US | EDT | DL<2:0> | | |
| bit 15 | | | | | | bit 8 | |
| R/W-0 | R/W-0 | R/W-1 | R/W-0 | R/C-0 | R/W-0 | R/W-0 | R/W-0 |
| SATA | SATB | SATDW | ACCSAT | IPL3 ⁽²⁾ | PSV | RND | IF |
| bit 7 | | | | | | bit 0 | |

Legend:

| | | | | |
|----------------------|----------------------|------------------------------------|-------------------|------------------|
| R = Readable bit | C = Clear only 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 3 **IPL3**: CPU Interrupt Priority Level Status bit⁽²⁾

1 = CPU interrupt priority level is greater than 7
 0 = CPU interrupt priority level is 7 or less

Note 1: For complete register details, see **Register 3-2: "CORCON: CORE Control Register"**.

2: The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU Interrupt Priority Level.

REGISTER 7-3: INTCON1: INTERRUPT CONTROL REGISTER 1

| | | | | | | | |
|--------|--------|--------|---------|---------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| NSTDIS | OVAERR | OVBERR | COVAERR | COVBERR | OVATE | OVBTE | COVTE |
| 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 |
| SFTACERR | DIV0ERR | DMACERR | 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 **OVAERR:** Accumulator A Overflow Trap Flag bit
 1 = Trap was caused by overflow of Accumulator A
 0 = Trap was not caused by overflow of Accumulator A
- bit 13 **OVBERR:** Accumulator B Overflow Trap Flag bit
 1 = Trap was caused by overflow of Accumulator B
 0 = Trap was not caused by overflow of Accumulator B
- bit 12 **COVAERR:** Accumulator A Catastrophic Overflow Trap Flag bit
 1 = Trap was caused by catastrophic overflow of Accumulator A
 0 = Trap was not caused by catastrophic overflow of Accumulator A
- bit 11 **COVBERR:** Accumulator B Catastrophic Overflow Trap Flag bit
 1 = Trap was caused by catastrophic overflow of Accumulator B
 0 = Trap was not caused by catastrophic overflow of Accumulator B
- bit 10 **OVATE:** Accumulator A Overflow Trap Enable bit
 1 = Trap overflow of Accumulator A
 0 = Trap disabled
- bit 9 **OVBTE:** Accumulator B Overflow Trap Enable bit
 1 = Trap overflow of Accumulator B
 0 = Trap disabled
- bit 8 **COVTE:** Catastrophic Overflow Trap Enable bit
 1 = Trap on catastrophic overflow of Accumulator A or B enabled
 0 = Trap disabled
- bit 7 **SFTACERR:** Shift Accumulator Error Status bit
 1 = Math error trap was caused by an invalid accumulator shift
 0 = Math error trap was not caused by an invalid accumulator shift
- bit 6 **DIV0ERR:** Arithmetic Error Status bit
 1 = Math error trap was caused by a divide by zero
 0 = Math error trap was not caused by a divide by zero
- bit 5 **DMACERR:** DMA Controller Error Status bit
 1 = DMA controller error trap has occurred
 0 = DMA controller error trap has not occurred
- bit 4 **MATHERR:** Arithmetic Error Status bit
 1 = Math error trap has occurred
 0 = Math error trap has not occurred

REGISTER 7-3: INTCON1: INTERRUPT CONTROL REGISTER 1 (CONTINUED)

- bit 3 **ADDRERR:** Address Error Trap Status bit
 1 = Address error trap has occurred
 0 = Address error trap has not occurred
- bit 2 **STKERR:** Stack Error Trap Status bit
 1 = Stack error trap has occurred
 0 = Stack error trap has not occurred
- bit 1 **OSCFAIL:** Oscillator Failure Trap Status bit
 1 = Oscillator failure trap has occurred
 0 = Oscillator failure trap has not occurred
- bit 0 **Unimplemented:** Read as '0'

REGISTER 7-4: INTCON2: INTERRUPT CONTROL REGISTER 2

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

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

Legend:

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

- bit 15 **ALTIVT:** Enable Alternate Interrupt Vector Table bit
 1 = Use alternate vector table
 0 = Use standard (default) vector table
- bit 14 **DISI:** DISI Instruction Status bit
 1 = DISI instruction is active
 0 = DISI instruction is not active
- bit 13-3 **Unimplemented:** Read as '0'
- bit 2 **INT2EP:** External Interrupt 2 Edge Detect Polarity Select bit
 1 = Interrupt on negative edge
 0 = Interrupt on positive edge
- bit 1 **INT1EP:** External Interrupt 1 Edge Detect Polarity Select bit
 1 = Interrupt on negative edge
 0 = Interrupt on positive edge
- bit 0 **INT0EP:** External Interrupt 0 Edge Detect Polarity Select bit
 1 = Interrupt on negative edge
 0 = Interrupt on positive edge

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

| | | | | | | | |
|--------|--------|-------|--------|--------|--------|---------|-------|
| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | DMA1IF | AD1IF | U1TXIF | U1RXIF | SPI1IF | SPI1EIF | T3IF |
| 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 |
| T2IF | OC2IF | IC2IF | DMA0IF | 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 **Unimplemented:** Read as '0'
- bit 14 **DMA1IF:** DMA Channel 1 Data Transfer Complete Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 13 **AD1IF:** ADC1 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 **SPI1EIF:** SPI1 Error 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 **DMA0IF:** DMA Channel 0 Data Transfer Complete Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 3 **T1IF:** Timer1 Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred

REGISTER 7-5: IFS0: INTERRUPT FLAG STATUS REGISTER 0 (CONTINUED)

- bit 2 **OC1IF:** Output Compare Channel 1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred
- bit 1 **IC1IF:** Input Capture Channel 1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred
- bit 0 **INT0IF:** External Interrupt 0 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

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

| | | | | | | | |
|--------|--------|--------|-------|-------|-------|-------|--------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| U2TXIF | U2RXIF | INT2IF | T5IF | T4IF | OC4IF | OC3IF | DMA2IF |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------|-------|-----|--------|-------|-------|---------|---------|
| R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| IC8IF | IC7IF | — | 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 **DMA2IF:** DMA Channel 2 Data Transfer Complete Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 7 **IC8IF:** Input Capture Channel 8 Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 6 **IC7IF:** Input Capture Channel 7 Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 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

REGISTER 7-6: IFS1: INTERRUPT FLAG STATUS REGISTER 1 (CONTINUED)

- bit 2 **CMIF:** Comparator Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 1 **M12C1IF:** I2C1 Master Events Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 0 **S12C1IF:** I2C1 Slave Events Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred

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

| | | | | | | | |
|--------|--------|-------|-----|-----|-----|-----|-------|
| U-0 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| — | DMA4IF | PMPIF | — | — | — | — | — |
| 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 |
| — | — | — | DMA3IF | C1IF ⁽¹⁾ | C1RXIF ⁽¹⁾ | SPI2IF | SPI2EIF |
| bit 7 | | | | | | | bit 0 |

| | | | |
|-------------------|------------------|------------------------------------|--------------------|
| Legend: | | | |
| R = Readable bit | W = Writable 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 **DMA4IF:** DMA Channel 4 Data Transfer Complete Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 13 **PMPIF:** Parallel Master Port Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 12-5 **Unimplemented:** Read as '0'
- bit 4 **DMA3IF:** DMA Channel 3 Data Transfer Complete Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 3 **C1IF:** ECAN1 Event Interrupt Flag Status bit⁽¹⁾
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 2 **C1RXIF:** ECAN1 Receive Data Ready Interrupt Flag Status bit⁽¹⁾
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 1 **SPI2IF:** SPI2 Event Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 0 **SPI2EIF:** SPI2 Error Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred

Note 1: Interrupts disabled on devices without ECAN™ modules

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

| | | | | | | | |
|---------|-------|--------|-----|-----|--------|--------|-------|
| R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | U-0 |
| FLTA1IF | RTCIF | DMA5IF | — | — | QE11IF | PWM1IF | — |
| 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 **FLTA1IF:** PWM1 Fault A Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 14 **RTCIF:** Real-Time Clock/Calendar Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 13 **DMA5IF:** DMA Channel 5 Data Transfer Complete Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 12-11 **Unimplemented:** Read as '0'
- bit 10 **QE11IF:** QE11 Event Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 9 **PWM1IF:** PWM1 Error Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 8-0 **Unimplemented:** Read as '0'

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

| | | | | | | | |
|------------------------|------------------------|--------|--------|--------|---------|--------|-------|
| R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | U-0 |
| DAC1LIF ⁽²⁾ | DAC1RIF ⁽²⁾ | — | — | QEI2IF | FLTA2IF | PWM2IF | — |
| bit 15 | | | | | | | bit 8 |
| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 |
| — | C1TXIF ⁽¹⁾ | DMA7IF | DMA6IF | CRCIF | U2EIF | U1EIF | — |
| bit 7 | | | | | | | bit 0 |

Legend:

R = Readable bit W = Writable 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 **DAC1LIF:** DAC Left Channel Interrupt Flag Status bit⁽²⁾
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 14 **DAC1RIF:** DAC Right Channel Interrupt Flag Status bit⁽²⁾
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 13-12 **Unimplemented:** Read as '0'
- bit 11 **QEI2IF:** QEI2 Event Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 10 **FLTA2IF:** PWM2 Fault A Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 9 **PWM2IF:** PWM2 Error Interrupt Enable bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 8-7 **Unimplemented:** Read as '0'
- bit 6 **C1TXIF:** ECAN1 Transmit Data Request Interrupt Flag Status bit⁽¹⁾
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 5 **DMA7IF:** DMA Channel 7 Data Transfer Complete Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 4 **DMA6IF:** DMA Channel 6 Data Transfer Complete Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 3 **CRCIF:** CRC Generator Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 2 **U2EIF:** UART2 Error Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 1 **U1EIF:** UART1 Error Interrupt Flag Status bit
 1 = Interrupt request has occurred
 0 = Interrupt request has not occurred
- bit 0 **Unimplemented:** Read as '0'

Note 1: Interrupts disabled on devices without ECAN™ modules.
2: Interrupts disabled on devices without DAC modules.

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

| | | | | | | | |
|--------|--------|-------|--------|--------|--------|---------|-------|
| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | DMA1IE | AD1IE | U1TXIE | U1RXIE | SPI1IE | SPI1EIE | T3IE |
| 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 |
| T2IE | OC2IE | IC2IE | DMA0IE | T1IE | OC1IE | IC1IE | INT0IE |
| bit 7 | | | | | | | bit 0 |

Legend:

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

- bit 15 **Unimplemented:** Read as '0'
- bit 14 **DMA1IE:** DMA Channel 1 Data Transfer Complete Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled
- bit 13 **AD1IE:** ADC1 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 Event Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled
- bit 9 **SPI1EIE:** SPI1 Error 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 **DMA0IE:** DMA Channel 0 Data Transfer Complete Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled
- bit 3 **T1IE:** Timer1 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

REGISTER 7-10: IEC0: INTERRUPT ENABLE CONTROL REGISTER 0 (CONTINUED)

- 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 Flag Status bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled

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

| | | | | | | | |
|--------|--------|--------|-------|-------|-------|-------|--------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| U2TXIE | U2RXIE | INT2IE | T5IE | T4IE | OC4IE | OC3IE | DMA2IE |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------|-------|-----|--------|-------|-------|---------|---------|
| R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| IC8IE | IC7IE | — | INT1IE | CNIE | CMIE | MI2C1IE | SI2C1IE |
| bit 7 | | | | | | | bit 0 |

Legend:

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

- bit 15 **U2TXIE:** UART2 Transmitter Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 14 **U2RXIE:** UART2 Receiver Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 13 **INT2IE:** External Interrupt 2 Enable bit
 1 = 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 **DMA2IE:** DMA Channel 2 Data Transfer Complete Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 7 **IC8IE:** Input Capture Channel 8 Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 6 **IC7IE:** Input Capture Channel 7 Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 5 **Unimplemented:** Read as '0'
- bit 4 **INT1IE:** External Interrupt 1 Enable bit
 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

REGISTER 7-11: IEC1: INTERRUPT ENABLE CONTROL REGISTER 1 (CONTINUED)

- bit 2 **CMIE:** Comparator Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 1 **M12C1IE:** I2C1 Master Events Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 0 **S12C1IE:** I2C1 Slave Events Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled

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

| | | | | | | | |
|--------|--------|-------|-----|-----|-----|-------|-----|
| U-0 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| — | DMA4IE | PMPIE | — | — | — | — | — |
| 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 |
| — | — | — | DMA3IE | C1IE ⁽¹⁾ | C1RXIE ⁽¹⁾ | SPI2IE | SPI2EIE |
| bit 7 | | | | | | bit 0 | |

Legend:

R = Readable bit W = Writable 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 **DMA4IE:** DMA Channel 4 Data Transfer Complete Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 13 **PMPIE:** Parallel Master Port Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 12-5 **Unimplemented:** Read as '0'
- bit 4 **DMA3IE:** DMA Channel 3 Data Transfer Complete Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request has enabled
- bit 3 **C1IE:** ECAN1 Event Interrupt Enable bit⁽¹⁾
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 2 **C1RXIE:** ECAN1 Receive Data Ready Interrupt Enable bit⁽¹⁾
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 1 **SPI2IE:** SPI2 Event Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 0 **SPI2EIE:** SPI2 Error Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled

Note 1: Interrupts are disabled on devices without ECAN™ modules.

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

| | | | | | | | |
|---------|-------|--------|-----|-----|--------|--------|-------|
| R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | U-0 |
| FLTA1IE | RTCIE | DMA5IE | — | — | QE11IE | PWM1IE | — |
| 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 **FLTA1IE:** PWM1 Fault A Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 14 **RTCIE:** Real-Time Clock/Calendar Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 13 **DMA5IE:** DMA Channel 5 Data Transfer Complete Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 12-11 **Unimplemented:** Read as '0'
- bit 10 **QE11IE:** QE11 Event Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 9 **PWM1IE:** PWM1 Error Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 8-0 **Unimplemented:** Read as '0'

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

| | | | | | | | |
|------------------------|------------------------|--------|--------|--------|---------|--------|-------|
| R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | U-0 |
| DAC1LIE ⁽²⁾ | DAC1RIE ⁽²⁾ | — | — | QEI2IE | FLTA2IE | PWM2IE | — |
| bit 15 | | | | | | | bit 8 |
| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 |
| — | C1TXIE ⁽¹⁾ | DMA7IE | DMA6IE | CRCIE | U2EIE | U1EIE | — |
| bit 7 | | | | | | | bit 0 |

Legend:

R = Readable bit W = Writable 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 **DAC1LIE:** DAC Left Channel Interrupt Enable bit⁽²⁾
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 14 **DAC1RIE:** DAC Right Channel Interrupt Enable bit⁽²⁾
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 13-12 **Unimplemented:** Read as '0'
- bit 11 **QEI2IE:** QEI2 Event Interrupt Flag Status bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 10 **FLTA2IE:** PWM2 Fault A Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 9 **PWM2IE:** PWM2 Error Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 8-7 **Unimplemented:** Read as '0'
- bit 6 **C1TXIE:** ECAN1 Transmit Data Request Interrupt Enable bit⁽¹⁾
 1 = Interrupt request occurred
 0 = Interrupt request not occurred
- bit 5 **DMA7IE:** DMA Channel 7 Data Transfer Complete Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 4 **DMA6IE:** DMA Channel 6 Data Transfer Complete Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 3 **CRCIE:** CRC Generator Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 2 **U2EIE:** UART2 Error Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 1 **U1EIE:** UART1 Error Interrupt Enable bit
 1 = Interrupt request enabled
 0 = Interrupt request not enabled
- bit 0 **Unimplemented:** Read as '0'

Note 1: Interrupts are disabled on devices without ECAN™ modules.
2: Interrupts are disabled on devices without DAC modules.

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 |
| — | T1IP<2:0> | | | — | OC1IP<2:0> | | |
| 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 |
| — | IC1IP<2:0> | | | — | INT0IP<2:0> | | |
| 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 **Unimplemented:** Read as '0'
- bit 14-12 **T1IP<2:0>:** Timer1 Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 11 **Unimplemented:** Read as '0'
- bit 10-8 **OC1IP<2:0>:** Output Compare Channel 1 Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 7 **Unimplemented:** Read as '0'
- bit 6-4 **IC1IP<2:0>:** Input Capture Channel 1 Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 3 **Unimplemented:** Read as '0'
- bit 2-0 **INT0IP<2:0>:** External Interrupt 0 Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled

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

| | | | | | | | |
|--------|-----------|-------|-------|-------|------------|-------|-------|
| U-0 | R/W-1 | R/W-0 | R/W-0 | U-0 | R/W-1 | R/W-0 | R/W-0 |
| — | T2IP<2:0> | | | — | OC2IP<2:0> | | |
| 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 |
| — | IC2IP<2:0> | | | — | DMA0IP<2:0> | | |
| 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 **Unimplemented:** Read as '0'
- bit 14-12 **T2IP<2:0>:** Timer2 Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 11 **Unimplemented:** Read as '0'
- bit 10-8 **OC2IP<2:0>:** Output Compare Channel 2 Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 7 **Unimplemented:** Read as '0'
- bit 6-4 **IC2IP<2:0>:** Input Capture Channel 2 Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 3 **Unimplemented:** Read as '0'
- bit 2-0 **DMA0IP<2:0>:** DMA Channel 0 Data Transfer Complete Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled

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 |
| — | U1RXIP<2:0> | | | — | SPI1IP<2:0> | | |
| 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 |
| — | SPI1EIP<2:0> | | | — | T3IP<2:0> | | |
| 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 **Unimplemented:** Read as '0'
- bit 14-12 **U1RXIP<2:0>:** UART1 Receiver Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 11 **Unimplemented:** Read as '0'
- bit 10-8 **SPI1IP<2:0>:** SPI1 Event Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 7 **Unimplemented:** Read as '0'
- bit 6-4 **SPI1EIP<2:0>:** SPI1 Error Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 3 **Unimplemented:** Read as '0'
- bit 2-0 **T3IP<2:0>:** Timer3 Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled

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

| | | | | | | | |
|--------|-----|-----|-----|-----|-------------|-------|-------|
| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-1 | R/W-0 | R/W-0 |
| — | — | — | — | — | DMA1IP<2:0> | | |
| 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 |
| — | AD1IP<2:0> | | — | U1TXIP<2:0> | | | |
| 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 **DMA1IP<2:0>:** DMA Channel 1 Data Transfer Complete 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 **AD1IP<2:0>:** ADC1 Conversion Complete Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 3 **Unimplemented:** Read as '0'
- bit 2-0 **U1TXIP<2:0>:** UART1 Transmitter Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled

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

| | | | | | | | |
|--------|-----------|-------|-------|-------|-----------|-------|-------|
| U-0 | R/W-1 | R/W-0 | R/W-0 | U-0 | R/W-1 | R/W-0 | R/W-0 |
| — | CNIP<2:0> | | | — | CMIP<2:0> | | |
| 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 |
| — | MI2C1IP<2:0> | | | — | SI2C1IP<2:0> | | |
| 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 **Unimplemented:** Read as '0'

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

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

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

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

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

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

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

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

| | | | | | | | |
|--------|------------|-------|-------|-------|------------|-------|-------|
| U-0 | R/W-1 | R/W-0 | R/W-0 | U-0 | R/W-1 | R/W-0 | R/W-0 |
| — | IC8IP<2:0> | | | — | IC7IP<2:0> | | |
| bit 15 | | | | bit 8 | | | |

| | | | | | | | |
|-------|-----|-----|-----|-------|-------------|-------|-------|
| U-0 | U-1 | U-0 | U-0 | U-0 | R/W-1 | R/W-0 | R/W-0 |
| — | — | — | — | — | INT1IP<2:0> | | |
| 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 **Unimplemented:** Read as '0'
- bit 14-12 **IC8IP<2:0>:** Input Capture Channel 8 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 **IC7IP<2:0>:** Input Capture Channel 7 Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 7-3 **Unimplemented:** Read as '0'
- bit 2-0 **INT1IP<2:0>:** External Interrupt 1 Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled

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

| | | | | | | | |
|--------|-----------|-------|-------|-------|------------|-------|-------|
| U-0 | R/W-1 | R/W-0 | R/W-0 | U-0 | R/W-1 | R/W-0 | R/W-0 |
| — | T4IP<2:0> | | | — | OC4IP<2:0> | | |
| 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 |
| — | OC3IP<2:0> | | | — | DMA2IP<2:0> | | |
| 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 **Unimplemented:** Read as '0'
- bit 14-12 **T4IP<2:0>:** Timer4 Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 11 **Unimplemented:** Read as '0'
- bit 10-8 **OC4IP<2:0>:** Output Compare Channel 4 Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 7 **Unimplemented:** Read as '0'
- bit 6-4 **OC3IP<2:0>:** Output Compare Channel 3 Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 3 **Unimplemented:** Read as '0'
- bit 2-0 **DMA2IP<2:0>:** DMA Channel 2 Data Transfer Complete Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled

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 |
| — | U2TXIP<2:0> | | | — | U2RXIP<2:0> | | |
| 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 |
| — | INT2IP<2:0> | | | — | T5IP<2:0> | | |
| 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 **Unimplemented:** Read as '0'
- bit 14-12 **U2TXIP<2:0>:** UART2 Transmitter Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 11 **Unimplemented:** Read as '0'
- bit 10-8 **U2RXIP<2:0>:** UART2 Receiver Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 7 **Unimplemented:** Read as '0'
- bit 6-4 **INT2IP<2:0>:** External Interrupt 2 Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 3 **Unimplemented:** Read as '0'
- bit 2-0 **T5IP<2:0>:** Timer5 Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled

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

| | | | | | | | |
|--------|--------------------------|-------|-------|-------|----------------------------|-------|-------|
| U-0 | R/W-1 | R/W-0 | R/W-0 | U-0 | R/W-1 | R/W-0 | R/W-0 |
| — | C1IP<2:0> ⁽¹⁾ | | | — | C1RXIP<2:0> ⁽¹⁾ | | |
| 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 |
| — | SPI2IP<2:0> | | | — | SPI2EIP<2:0> | | |
| 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 **Unimplemented:** Read as '0'
- bit 14-12 **C1IP<2:0>:** ECAN1 Event Interrupt Priority bits⁽¹⁾
 - 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 **C1RXIP<2:0>:** ECAN1 Receive Data Ready Interrupt Priority bits⁽¹⁾
 - 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 **SPI2IP<2:0>:** SPI2 Event Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 3 **Unimplemented:** Read as '0'
- bit 2-0 **SPI2EIP<2:0>:** SPI2 Error Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled

Note 1: Interrupts disabled on devices without ECAN™ modules

REGISTER 7-24: IPC9: INTERRUPT PRIORITY CONTROL 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 | U-0 | U-0 | R/W-1 | R/W-0 | R/W-0 |
| — | — | — | — | — | DMA3IP<2:0> | | |
| 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 15-3 **Unimplemented:** Read as '0'

bit 2-0 **DMA3IP<2:0>:** DMA Channel 3 Data Transfer Complete Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)

-
-
-

001 = Interrupt is priority 1

000 = Interrupt source is disabled

REGISTER 7-25: IPC11: INTERRUPT PRIORITY CONTROL REGISTER 11

| | | | | | | | |
|--------|-------------|-------|-------|-------|-------------|-------|-------|
| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-1 | R/W-0 | R/W-0 |
| — | — | — | — | — | DMA4IP<2:0> | | |
| bit 15 | | | | | bit 8 | | |
| U-0 | R/W-1 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
| — | PMPPIP<2:0> | | | — | — | — | — |
| 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 **DMA4IP<2:0>:** DMA Channel 4 Data Transfer Complete 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 **PMPPIP<2:0>:** Parallel Master Port Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 3-0 **Unimplemented:** Read as '0'

REGISTER 7-26: IPC14: INTERRUPT PRIORITY CONTROL REGISTER 14

| | | | | | | | |
|--------|-----|-----|-----|-----|-------------|-------|-------|
| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-1 | R/W-0 | R/W-0 |
| — | — | — | — | — | QE11IP<2:0> | | |
| bit 15 | | | | | bit 8 | | |

| | | | | | | | |
|-------|-------------|-------|-------|-------|-----|-----|-----|
| U-0 | R/W-1 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
| — | PWM1IP<2:0> | | | — | — | — | — |
| 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 **QE11IP<2:0>:** QE11 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 **PWM1IP<2:0>:** PWM1 Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 3-0 **Unimplemented:** Read as '0'

REGISTER 7-27: IPC15: INTERRUPT PRIORITY CONTROL REGISTER 15

| | | | | | | | |
|--------|--------------|-------|-------|-------|------------|-------|-------|
| U-0 | R/W-1 | R/W-0 | R/W-0 | U-0 | R/W-1 | R/W-0 | R/W-0 |
| — | FLTA1IP<2:0> | | | — | RTCIP<2:0> | | |
| bit 15 | | | | bit 8 | | | |

| | | | | | | | |
|-------|-------------|-------|-------|-------|-----|-----|-----|
| U-0 | R/W-1 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
| — | DMA5IP<2:0> | | | — | — | — | — |
| 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 **Unimplemented:** Read as '0'
- bit 14-12 **FLTA1IP<2:0>:** PWM Fault A 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 **RTCIP<2:0>:** Real-Time Clock/Calendar Interrupt Flag Status 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 **DMA5IP<2:0>:** DMA Channel 5 Data Transfer Complete Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 3-0 **Unimplemented:** Read as '0'

REGISTER 7-28: 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 |
| — | CRCIP<2:0> | | | — | U2EIP<2:0> | | |
| bit 15 | | | | bit 8 | | | |

| | | | | | | | |
|-------|------------|-------|-------|-------|-----|-----|-----|
| U-0 | R/W-1 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
| — | U1EIP<2:0> | | | — | — | — | — |
| 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 **Unimplemented:** Read as '0'
- bit 14-12 **CRCIP<2:0>:** CRC Generator Error Interrupt Flag 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 **U2EIP<2:0>:** UART2 Error Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 7 **Unimplemented:** Read as '0'
- bit 6-4 **U1EIP<2:0>:** UART1 Error Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled
- bit 3-0 **Unimplemented:** Read as '0'

REGISTER 7-29: IPC17: INTERRUPT PRIORITY CONTROL REGISTER 17

| | | | | | | | |
|--------|-----|-----|-----|-----|----------------------------|-------|-------|
| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-1 | R/W-0 | R/W-0 |
| — | — | — | — | — | C1TXIP<2:0> ⁽¹⁾ | | |
| 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 |
| — | DMA7IP<2:0> | | | — | DMA6IP<2:0> | | |
| 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 **C1TXIP<2:0>:** ECAN1 Transmit Data Request Interrupt Priority bits⁽¹⁾
 - 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 **DMA7IP<2:0>:** DMA Channel 7 Data Transfer 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 **DMA6IP<2:0>:** DMA Channel 6 Data Transfer Complete Interrupt Priority bits
 - 111 = Interrupt is priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is priority 1
 - 000 = Interrupt source is disabled

Note 1: Interrupts are disabled on devices without ECAN™ modules.

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

| | | | | | | | |
|--------|-------------|-------|-------|-------|--------------|-------|-------|
| U-0 | R/W-1 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
| — | QEI2IP<2:0> | | | — | FLTA2IP<2:0> | | |
| bit 15 | | | | bit 8 | | | |

| | | | | | | | |
|-------|-------------|-------|-------|-------|-----|-----|-----|
| U-0 | R/W-1 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
| — | PWM2IP<2:0> | | | — | — | — | — |
| 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 **Unimplemented:** Read as '0'
- bit 14-12 **QEI2IP<2:0>:** QEI2 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 **FLTA2IP<2:0>:** PWM2 Fault A 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 **PWM2IP<2:0>:** PWM2 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 7-31: IPC19: INTERRUPT PRIORITY CONTROL REGISTER 19

| | | | | | | | |
|--------|-----------------------------|-------|-------|-------|-----------------------------|-------|-------|
| U-0 | R/W-1 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
| — | DAC1LIP<2:0> ⁽¹⁾ | | | — | DAC1RIP<2:0> ⁽¹⁾ | | |
| 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 **Unimplemented:** Read as '0'
- bit 14-12 **DAC1LIP<2:0>:** DAC Left Channel Interrupt Flag Status bit⁽¹⁾
 - 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 **DAC1RIP<2:0>:** DAC Right Channel Interrupt Flag Status bit⁽¹⁾
 - 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'

Note 1: Interrupts are disabled on devices without DAC modules.

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

| | | | | | | | |
|--------|-----|-----|-----|----------|-----|-----|-----|
| U-0 | U-0 | U-0 | U-0 | R-0 | R-0 | R-0 | R-0 |
| — | — | — | — | ILR<3:0> | | | |
| bit 15 | | | | bit 8 | | | |

| | | | | | | | |
|-------|-------------|-----|-----|-----|-----|-----|-------|
| U-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| — | VECNUM<6:0> | | | | | | |
| 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 15-12 **Unimplemented:** Read as '0'

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

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

bit 6-0 **VECNUM<6:0>:** Vector Number of Pending Interrupt bits
 0111111 = Interrupt Vector pending is number 135
 •
 •
 •
 0000001 = Interrupt Vector pending is number 9
 0000000 = Interrupt Vector pending is number 8

7.4 Interrupt Setup Procedures

7.4.1 INITIALIZATION

To configure an interrupt source at initialization:

1. Set the NSTDIS 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 depends 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 can 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 used to declare an ISR and initialize the IVT with the correct vector address depends on the programming language (C or assembler) and the language development tool suite used to develop the application.

In general, the user application must clear the interrupt flag in the appropriate IFSx register for the source of interrupt that the ISR handles. Otherwise, the program re-enters the ISR 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 this 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 0Eh with SRL.

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

Note: Only user interrupts with a priority level of 7 or lower can be disabled. Trap sources (level 8-level 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.

8.0 DIRECT MEMORY ACCESS (DMA)

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 38. “Direct Memory Access (DMA) (Part III)” (DS70215), which is available from the Microchip website (www.microchip.com).

Direct Memory Access (DMA) is a very efficient mechanism of copying data between peripheral SFRs (e.g., UART Receive register, Input Capture 1 buffer), and buffers or variables stored in RAM, with minimal CPU intervention. The DMA controller can automatically copy entire blocks of data without requiring the user software to read or write the peripheral Special Function Registers (SFRs) every time a peripheral interrupt occurs. The DMA controller uses a dedicated bus for data transfers and therefore, does not steal cycles from the code execution flow of the CPU. To exploit the DMA capability, the corresponding user buffers or variables must be located in DMA RAM.

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 peripherals that can utilize DMA are listed in Table 8-1.

TABLE 8-1: DMA CHANNEL TO PERIPHERAL ASSOCIATIONS

| Peripheral to DMA Association | DMAxREQ Register IRQSEL<6:0> Bits | DMAxPAD Register Values to Read From Peripheral | DMAxPAD Register Values to Write to Peripheral |
|---------------------------------------|-----------------------------------|---|--|
| INT0 – External Interrupt 0 | 0000000 | — | — |
| IC1 – Input Capture 1 | 0000001 | 0x0140 (IC1BUF) | — |
| OC1 – Output Compare 1 Data | 0000010 | — | 0x0182 (OC1R) |
| OC1 – Output Compare 1 Secondary Data | 0000010 | — | 0x0180 (OC1RS) |
| IC2 – Input Capture 2 | 0000101 | 0x0144 (IC2BUF) | — |
| OC2 – Output Compare 2 Data | 0000110 | — | 0x0188 (OC2R) |
| OC2 – Output Compare 2 Secondary Data | 0000110 | — | 0x0186 (OC2RS) |
| TMR2 – Timer2 | 0000111 | — | — |
| TMR3 – Timer3 | 0001000 | — | — |
| SPI1 – Transfer Done | 0001010 | 0x0248 (SPI1BUF) | 0x0248 (SPI1BUF) |
| UART1RX – UART1 Receiver | 0001011 | 0x0226 (U1RXREG) | — |
| UART1TX – UART1 Transmitter | 0001100 | — | 0x0224 (U1TXREG) |
| ADC1 – ADC1 convert done | 0001101 | 0x0300 (ADC1BUF0) | — |
| UART2RX – UART2 Receiver | 0011110 | 0x0236 (U2RXREG) | — |
| UART2TX – UART2 Transmitter | 0011111 | — | 0x0234 (U2TXREG) |
| SPI2 – Transfer Done | 0100001 | 0x0268 (SPI2BUF) | 0x0268 (SPI2BUF) |
| ECAN1 – RX Data Ready | 0100010 | 0x0440 (C1RXD) | — |
| PMP - Master Data Transfer | 0101101 | 0x0608 (PMDIN1) | 0x0608 (PMDIN1) |
| ECAN1 – TX Data Request | 1000110 | — | 0x0442 (C1TXD) |
| DAC1 - Right Data Output | 1001110 | — | 0x3F6 (DAC1RDAT) |
| DAC2 - Left Data Output | 1001111 | — | 0x03F8 (DAC1LDAT) |

The DMA controller features eight identical data transfer channels.

Each channel has its own set of control and status registers. Each DMA channel can be configured to copy data either from buffers stored in dual port DMA RAM to peripheral SFRs, or from peripheral SFRs to buffers in DMA RAM.

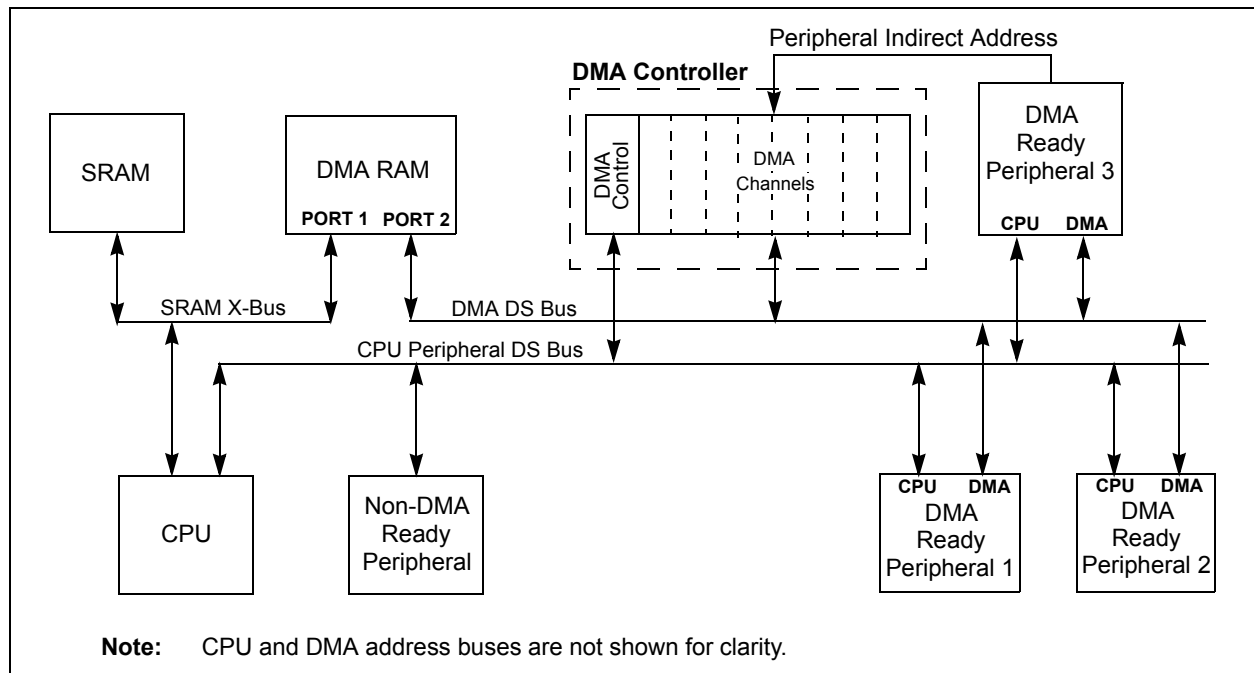
The DMA controller supports the following features:

- Eight DMA channels
- Register Indirect With Post-increment Addressing mode
- Register Indirect Without Post-increment Addressing mode
- Peripheral Indirect Addressing mode (peripheral generates destination address)
- CPU interrupt after half or full block transfer complete

- Byte or word transfers
- Fixed priority channel arbitration
- Manual (software) or Automatic (peripheral DMA requests) transfer initiation
- One-Shot or Auto-Repeat block transfer modes
- Ping-Pong mode (automatic switch between two DPSRAM start addresses after each block transfer complete)
- DMA request for each channel can be selected from any supported interrupt source
- Debug support features

For each DMA channel, a DMA interrupt request is generated when a block transfer is complete. Alternatively, an interrupt can be generated when half of the block has been filled.

FIGURE 8-1: TOP LEVEL SYSTEM ARCHITECTURE USING A DEDICATED TRANSACTION BUS



8.1 DMAC Registers

Each DMAC Channel x ($x = 0, 1, 2, 3, 4, 5, 6$ or 7) contains the following registers:

- A 16-bit DMA Channel Control register (DMAxCON)
- A 16-bit DMA Channel IRQ Select register (DMAxREQ)
- A 16-bit DMA RAM Primary Start Address register (DMAxSTA)
- A 16-bit DMA RAM Secondary Start Address register (DMAxSTB)
- A 16-bit DMA Peripheral Address register (DMAxPAD)
- A 10-bit DMA Transfer Count register (DMAxCNT)

An additional pair of status registers, DMACS0 and DMACS1, are common to all DMAC channels. DMACS0 contains the DMA RAM and SFR write collision flags, XWCOL x and PWCOL x , respectively. DMACS1 indicates DMA channel and Ping-Pong mode status.

The DMAxCON, DMAxREQ, DMAxPAD and DMAxCNT are all conventional read/write registers. Reads of DMAxSTA or DMAxSTB reads the contents of the DMA RAM Address register. Writes to DMAxSTA or DMAxSTB write to the registers. This allows the user to determine the DMA buffer pointer value (address) at any time.

The interrupt flags (DMAxIF) are located in an IFS x register in the interrupt controller. The corresponding interrupt enable control bits (DMAxIE) are located in an IEC x register in the interrupt controller, and the corresponding interrupt priority control bits (DMAxIP) are located in an IPC x register in the interrupt controller.

REGISTER 8-1: DMAxCON: DMA CHANNEL x CONTROL REGISTER

| | | | | | | | |
|--------|-------|------------|-------|-------|-----|-----------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 |
| CHEN | SIZE | DIR | HALF | NULLW | — | — | — |
| bit 15 | | | | | | | bit 8 |
| U-0 | U-0 | R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 |
| — | — | AMODE<1:0> | | — | — | MODE<1:0> | |
| 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 **CHEN:** Channel Enable bit
1 = Channel enabled
0 = Channel disabled
- bit 14 **SIZE:** Data Transfer Size bit
1 = Byte
0 = Word
- bit 13 **DIR:** Transfer Direction bit (source/destination bus select)
1 = Read from DMA RAM address, write to peripheral address
0 = Read from peripheral address, write to DMA RAM address
- bit 12 **HALF:** Early Block Transfer Complete Interrupt Select bit
1 = Initiate block transfer complete interrupt when half of the data has been moved
0 = Initiate block transfer complete interrupt when all of the data has been moved
- bit 11 **NULLW:** Null Data Peripheral Write Mode Select bit
1 = Null data write to peripheral in addition to DMA RAM write (DIR bit must also be clear)
0 = Normal operation
- bit 10-6 **Unimplemented:** Read as '0'
- bit 5-4 **AMODE<1:0>:** DMA Channel Operating Mode Select bits
11 = Reserved (acts as Peripheral Indirect Addressing mode)
10 = Peripheral Indirect Addressing mode
01 = Register Indirect without Post-Increment mode
00 = Register Indirect with Post-Increment mode
- bit 3-2 **Unimplemented:** Read as '0'
- bit 1-0 **MODE<1:0>:** DMA Channel Operating Mode Select bits
11 = One-Shot, Ping-Pong modes enabled (one block transfer from/to each DMA RAM buffer)
10 = Continuous, Ping-Pong modes enabled
01 = One-Shot, Ping-Pong modes disabled
00 = Continuous, Ping-Pong modes disabled

REGISTER 8-2: DMAxREQ: DMA CHANNEL x IRQ SELECT REGISTER

| | | | | | | | |
|----------------------|-----|-----|-----|-----|-----|-----|-------|
| R/W-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| FORCE ⁽¹⁾ | — | — | — | — | — | — | — |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------|----------------------------|-------|-------|-----|-----|-------|-------|
| U-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 |
| — | IRQSEL<6:0> ⁽²⁾ | | | | | | — |
| bit 7 | | | | | | | bit 0 |

Legend:

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

- bit 15 **FORCE:** Force DMA Transfer bit⁽¹⁾
 1 = Force a single DMA transfer (Manual mode)
 0 = Automatic DMA transfer initiation by DMA request
- bit 14-7 **Unimplemented:** Read as '0'
- bit 6-0 **IRQSEL<6:0>:** DMA Peripheral IRQ Number Select bits⁽²⁾
 0000000-1111111 = DMAIRQ0-DMAIRQ127 selected to be Channel DMAREQ

- Note 1:** The FORCE bit cannot be cleared by the user. The FORCE bit is cleared by hardware when the forced DMA transfer is complete.
- 2:** Refer to Table 7-1 for a complete listing of IRQ numbers for all interrupt sources.

REGISTER 8-3: DMAxSTA: DMA CHANNEL x RAM START ADDRESS REGISTER A⁽¹⁾

| | | | | | | | |
|-----------|-------|-------|-------|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| STA<15:8> | | | | | | | |
| 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 |
| STA<7:0> | | | | | | | |
| 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 15-0 **STA<15:0>**: Primary DMA RAM Start Address bits (source or destination)

Note 1: A read of this address register returns the current contents of the DMA RAM Address register, not the contents written to STA<15:0>. If the channel is enabled (i.e., active), writes to this register may result in unpredictable behavior of the DMA channel and should be avoided.

REGISTER 8-4: DMAxSTB: DMA CHANNEL x RAM START ADDRESS REGISTER B⁽¹⁾

| | | | | | | | |
|-----------|-------|-------|-------|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| STB<15:8> | | | | | | | |
| 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 |
| STB<7:0> | | | | | | | |
| 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 15-0 **STB<15:0>**: Secondary DMA RAM Start Address bits (source or destination)

Note 1: A read of this address register returns the current contents of the DMA RAM Address register, not the contents written to STB<15:0>. If the channel is enabled (i.e., active), writes to this register may result in unpredictable behavior of the DMA channel and should be avoided.

REGISTER 8-5: DMAxPAD: DMA CHANNEL x PERIPHERAL ADDRESS REGISTER⁽¹⁾

| | | | | | | | |
|-----------|-------|-------|-------|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| PAD<15:8> | | | | | | | |
| 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 |
| PAD<7:0> | | | | | | | |
| 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 15-0 **PAD<15:0>**: Peripheral Address Register bits

Note 1: If the channel is enabled (i.e., active), writes to this register may result in unpredictable behavior of the DMA channel and should be avoided.

REGISTER 8-6: DMAxCNT: DMA CHANNEL x TRANSFER COUNT REGISTER⁽¹⁾

| | | | | | | | |
|--------|-----|-----|-----|-----|-----|-------------------------|-------|
| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 |
| — | — | — | — | — | — | CNT<9:8> ⁽²⁾ | |
| 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 |
| CNT<7:0> ⁽²⁾ | | | | | | | |
| bit 7 | | | | | | | bit 0 |

Legend:

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

bit 15-10 **Unimplemented:** Read as '0'

bit 9-0 **CNT<9:0>**: DMA Transfer Count Register bits⁽²⁾

Note 1: If the channel is enabled (i.e., active), writes to this register may result in unpredictable behavior of the DMA channel and should be avoided.

2: Number of DMA transfers = CNT<9:0> + 1.

REGISTER 8-7: DMACS0: DMA CONTROLLER STATUS REGISTER 0

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|
| R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 |
| PWCOL7 | PWCOL6 | PWCOL5 | PWCOL4 | PWCOL3 | PWCOL2 | PWCOL1 | PWCOL0 |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|
| R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 |
| XWCOL7 | XWCOL6 | XWCOL5 | XWCOL4 | XWCOL3 | XWCOL2 | XWCOL1 | XWCOL0 |
| bit 7 | | | | | | | bit 0 |

Legend:

| | | |
|-------------------|------------------|------------------------------------|
| R = Readable bit | W = Writable 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 **PWCOL7:** Channel 7 Peripheral Write Collision Flag bit
1 = Write collision detected
0 = No write collision detected
- bit 14 **PWCOL6:** Channel 6 Peripheral Write Collision Flag bit
1 = Write collision detected
0 = No write collision detected
- bit 13 **PWCOL5:** Channel 5 Peripheral Write Collision Flag bit
1 = Write collision detected
0 = No write collision detected
- bit 12 **PWCOL4:** Channel 4 Peripheral Write Collision Flag bit
1 = Write collision detected
0 = No write collision detected
- bit 11 **PWCOL3:** Channel 3 Peripheral Write Collision Flag bit
1 = Write collision detected
0 = No write collision detected
- bit 10 **PWCOL2:** Channel 2 Peripheral Write Collision Flag bit
1 = Write collision detected
0 = No write collision detected
- bit 9 **PWCOL1:** Channel 1 Peripheral Write Collision Flag bit
1 = Write collision detected
0 = No write collision detected
- bit 8 **PWCOL0:** Channel 0 Peripheral Write Collision Flag bit
1 = Write collision detected
0 = No write collision detected
- bit 7 **XWCOL7:** Channel 7 DMA RAM Write Collision Flag bit
1 = Write collision detected
0 = No write collision detected
- bit 6 **XWCOL6:** Channel 6 DMA RAM Write Collision Flag bit
1 = Write collision detected
0 = No write collision detected
- bit 5 **XWCOL5:** Channel 5 DMA RAM Write Collision Flag bit
1 = Write collision detected
0 = No write collision detected
- bit 4 **XWCOL4:** Channel 4 DMA RAM Write Collision Flag bit
1 = Write collision detected
0 = No write collision detected

REGISTER 8-7: DMACS0: DMA CONTROLLER STATUS REGISTER 0 (CONTINUED)

- bit 3 **XWCOL3:** Channel 3 DMA RAM Write Collision Flag bit
1 = Write collision detected
0 = No write collision detected
- bit 2 **XWCOL2:** Channel 2 DMA RAM Write Collision Flag bit
1 = Write collision detected
0 = No write collision detected
- bit 1 **XWCOL1:** Channel 1 DMA RAM Write Collision Flag bit
1 = Write collision detected
0 = No write collision detected
- bit 0 **XWCOL0:** Channel 0 DMA RAM Write Collision Flag bit
1 = Write collision detected
0 = No write collision detected

REGISTER 8-8: DMACS1: DMA CONTROLLER STATUS REGISTER 1

| | | | | | | | |
|--------|-----|-----|-----|------------|-----|-----|-----|
| U-0 | U-0 | U-0 | U-0 | R-1 | R-1 | R-1 | R-1 |
| — | — | — | — | LSTCH<3:0> | | | |
| bit 15 | | | | bit 8 | | | |

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| PPST7 | PPST6 | PPST5 | PPST4 | PPST3 | PPST2 | PPST1 | PPST0 |
| bit 7 | | | | | | | bit 0 |

Legend:

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

- bit 15-12 **Unimplemented:** Read as '0'
- bit 11-8 **LSTCH<3:0>:** Last DMA Channel Active bits
 - 1111 = No DMA transfer has occurred since system Reset
 - 1110-1000 = Reserved
 - 0111 = Last data transfer was by DMA Channel 7
 - 0110 = Last data transfer was by DMA Channel 6
 - 0101 = Last data transfer was by DMA Channel 5
 - 0100 = Last data transfer was by DMA Channel 4
 - 0011 = Last data transfer was by DMA Channel 3
 - 0010 = Last data transfer was by DMA Channel 2
 - 0001 = Last data transfer was by DMA Channel 1
 - 0000 = Last data transfer was by DMA Channel 0
- bit 7 **PPST7:** Channel 7 Ping-Pong Mode Status Flag bit
 - 1 = DMA7STB register selected
 - 0 = DMA7STA register selected
- bit 6 **PPST6:** Channel 6 Ping-Pong Mode Status Flag bit
 - 1 = DMA6STB register selected
 - 0 = DMA6STA register selected
- bit 5 **PPST5:** Channel 5 Ping-Pong Mode Status Flag bit
 - 1 = DMA5STB register selected
 - 0 = DMA5STA register selected
- bit 4 **PPST4:** Channel 4 Ping-Pong Mode Status Flag bit
 - 1 = DMA4STB register selected
 - 0 = DMA4STA register selected
- bit 3 **PPST3:** Channel 3 Ping-Pong Mode Status Flag bit
 - 1 = DMA3STB register selected
 - 0 = DMA3STA register selected
- bit 2 **PPST2:** Channel 2 Ping-Pong Mode Status Flag bit
 - 1 = DMA2STB register selected
 - 0 = DMA2STA register selected
- bit 1 **PPST1:** Channel 1 Ping-Pong Mode Status Flag bit
 - 1 = DMA1STB register selected
 - 0 = DMA1STA register selected
- bit 0 **PPST0:** Channel 0 Ping-Pong Mode Status Flag bit
 - 1 = DMA0STB register selected
 - 0 = DMA0STA register selected

REGISTER 8-9: DSADR: MOST RECENT DMA RAM ADDRESS

| | | | | | | | |
|-------------|-----|-----|-----|-------|-----|-----|-----|
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| DSADR<15:8> | | | | | | | |
| bit 15 | | | | bit 8 | | | |

| | | | | | | | |
|------------|-----|-----|-----|-------|-----|-----|-----|
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| DSADR<7:0> | | | | | | | |
| 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 15-0 **DSADR<15:0>**: Most Recent DMA RAM Address Accessed by DMA Controller bits

NOTES:

9.0 OSCILLATOR CONFIGURATION

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 39. “Oscillator (Part III)” (DS70216), which is available from the Microchip website (www.microchip.com).

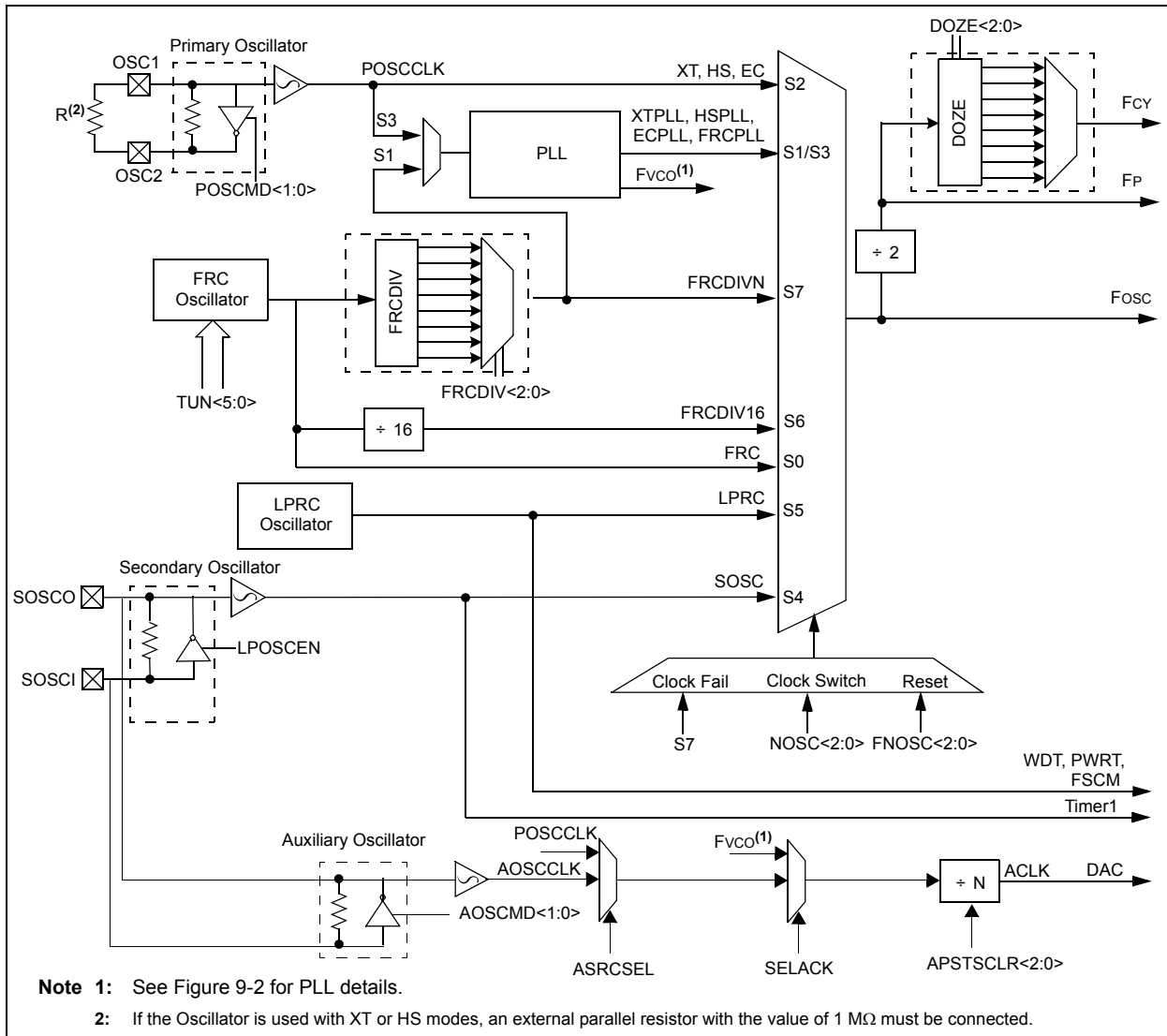
The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 oscillator system provides:

- External and internal oscillator options as clock sources

- An on-chip Phase-Locked Loop (PLL) to scale the internal operating frequency to the required system clock frequency
- An internal FRC oscillator that can also be used with the PLL, thereby allowing full-speed operation without any external clock generation hardware
- Clock switching between various clock sources
- Programmable clock postscaler for system power savings
- A Fail-Safe Clock Monitor (FSCM) that detects clock failure and takes fail-safe measures
- A Clock Control register (OSCCON)
- Nonvolatile Configuration bits for main oscillator selection.
- An auxiliary crystal oscillator for audio DAC

A simplified diagram of the oscillator system is shown in Figure 9-1.

FIGURE 9-1: dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, AND dsPIC33FJ128MCX02/X04 OSCILLATOR SYSTEM DIAGRAM



9.1 CPU Clocking System

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices provide seven system clock options:

- Fast RC (FRC) Oscillator
- FRC Oscillator with Phase Locked Loop (PLL)
- Primary (XT, HS or EC) Oscillator
- Primary Oscillator with PLL
- Secondary (LP) Oscillator
- Low-Power RC (LPRC) Oscillator
- FRC Oscillator with postscaler

9.1.1 SYSTEM CLOCK SOURCES

The Fast RC (FRC) internal oscillator runs at a nominal frequency of 7.37 MHz. User software can tune the FRC frequency. User software can optionally specify a factor (ranging from 1:2 to 1:256) by which the FRC clock frequency is divided. This factor is selected using the `FRCDIV<2:0>` (`CLKDIV<10:8>`) bits.

The primary oscillator can use one of the following as its clock source:

- Crystal (XT): Crystals and ceramic resonators in the range of 3 MHz to 10 MHz. The crystal is connected to the OSC1 and OSC2 pins.
- High-Speed Crystal (HS): Crystals in the range of 10 MHz to 40 MHz. The crystal is connected to the OSC1 and OSC2 pins.
- External Clock (EC): External clock signal is directly applied to the OSC1 pin.

The secondary (LP) oscillator is designed for low power and uses a 32.768 kHz crystal or ceramic resonator. The LP oscillator uses the SOSC1 and SOSCO pins.

The Low-Power RC (LPRC) internal oscillator runs at a nominal frequency of 32.768 kHz. It is also used as a reference clock by the Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM).

The clock signals generated by the FRC and primary oscillators can be optionally applied to an on-chip PLL to provide a wide range of output frequencies for device operation. PLL configuration is described in **Section 9.1.4 “PLL Configuration”**.

The FRC frequency depends on the FRC accuracy (see Table 31-19) and the value of the FRC Oscillator Tuning register (see Register 9-4).

9.1.2 SYSTEM CLOCK SELECTION

The oscillator source 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 28.1 “Configuration Bits”** for further details.) The Initial Oscillator Selection Configuration bits, `FNOSC<2:0>` (`FOSCSEL<2:0>`), and the Primary Oscillator Mode Select Configuration bits, `POSCMD<1:0>` (`FOSC<1:0>`), select the oscillator source that is used at a Power-on Reset. The FRC primary oscillator is the default (unprogrammed) selection.

The Configuration bits allow users to choose among 12 different clock modes, shown in Table 9-1.

The output of the oscillator (or the output of the PLL if a PLL mode has been selected) `FOSC` is divided by 2 to generate the device instruction clock (`FCY`) and the peripheral clock time base (`FP`). `FCY` defines the operating speed of the device, and speeds up to 40 MHz are supported by the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 architecture.

Instruction execution speed or device operating frequency, `FCY`, is given by:

EQUATION 9-1: DEVICE OPERATING FREQUENCY

| |
|----------------|
| $FCY = FOSC/2$ |
|----------------|

9.1.3 AUXILIARY OSCILLATOR

The Auxiliary Oscillator (AOSC) can be used for peripheral that needs to operate at a frequency unrelated to the system clock such as DAC.

The Auxiliary Oscillator can use one of the following as its clock source:

Crystal (XT): Crystal and ceramic resonators in the range of 3 Mhz to 10 Mhz. The crystal is connected to the SOCI and SOSCO pins.

High-Speed Crystal (HS): Crystals in the range of 10 to 40 Hz. The crystal is connected to the SOSC1 and SOSCO pins.

External Clock (EC): External clock signal up to 64 Mhz. The external clock signal is directly applied to SOSC1 pin.

9.1.4 PLL CONFIGURATION

The primary oscillator and internal FRC oscillator can optionally use an on-chip PLL to obtain higher speeds of operation. The PLL provides significant flexibility in selecting the device operating speed. A block diagram of the PLL is shown in Figure 9-2.

The output of the primary oscillator or FRC, denoted as 'FIN', is divided down by a prescale factor (N1) of 2, 3, ... or 33 before being provided to the PLL's Voltage Controlled Oscillator (VCO). The input to the VCO must be selected in the range of 0.8 MHz to 8 MHz. The prescale factor 'N1' is selected using the PLLPRE<4:0> bits (CLKDIV<4:0>).

The PLL Feedback Divisor, selected using the PLLDIV<8:0> bits (PLLFBD<8:0>), provides a factor 'M,' by which the input to the VCO is multiplied. This factor must be selected such that the resulting VCO output frequency is in the range of 100 MHz to 200 MHz.

The VCO output is further divided by a postscale factor 'N2.' This factor is selected using the PLLPOST<1:0> bits (CLKDIV<7:6>). 'N2' can be either 2, 4 or 8, and must be selected such that the PLL output frequency (Fosc) is in the range of 12.5 MHz to 80 MHz, which generates device operating speeds of 6.25-40 MIPS.

For a primary oscillator or FRC oscillator, output 'FIN', the PLL output 'Fosc' is given by:

EQUATION 9-2: Fosc CALCULATION

$$F_{OSC} = F_{IN} \cdot \left(\frac{M}{N1 \cdot N2} \right)$$

For example, suppose a 10 MHz crystal is being used with the selected oscillator mode of XT with PLL.

- If PLLPRE<4:0> = 0, then N1 = 2. This yields a VCO input of 10/2 = 5 MHz, which is within the acceptable range of 0.8 MHz - 8 MHz.
- If PLLDIV<8:0> = 0x1E, then M = 32. This yields a VCO output of 5 x 32 = 160 MHz, which is within the 100 MHz - 200 MHz ranged needed.
- If PLLPOST<1:0> = 0, then N2 = 2. This provides a Fosc of 160/2 = 80 MHz. The resultant device operating speed is 80/2 = 40 MIPS.

EQUATION 9-3: XT WITH PLL MODE EXAMPLE

$$F_{CY} = \frac{F_{OSC}}{2} = \frac{1}{2} \left(\frac{10000000 \cdot 32}{2 \cdot 2} \right) = 40 \text{ MIPS}$$

FIGURE 9-2: dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, AND dsPIC33FJ128MCX02/X04 PLL BLOCK DIAGRAM

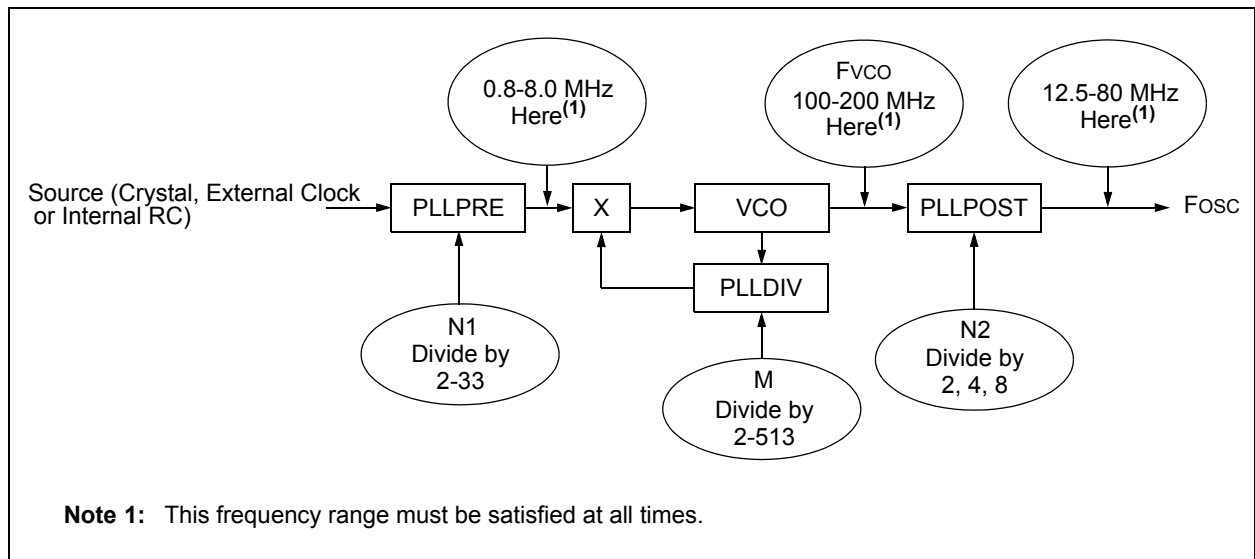


TABLE 9-1: CONFIGURATION BIT VALUES FOR CLOCK SELECTION

| Oscillator Mode | Oscillator Source | POSCMD<1:0> | FNOSC<2:0> | Note |
|---|-------------------|-------------|------------|-------------|
| Fast RC Oscillator with Divide-by-N (FRCDIVN) | Internal | xx | 111 | 1, 2 |
| Fast RC Oscillator with Divide-by-16 (FRCDIV16) | Internal | xx | 110 | 1 |
| Low-Power RC Oscillator (LPRC) | Internal | xx | 101 | 1 |
| Secondary (Timer1) Oscillator (SOSC) | Secondary | xx | 100 | 1 |
| Primary Oscillator (HS) with PLL (HSPLL) | Primary | 10 | 011 | |
| Primary Oscillator (XT) with PLL (XTPLL) | Primary | 01 | 011 | |
| Primary Oscillator (EC) with PLL (ECPLL) | Primary | 00 | 011 | 1 |
| Primary Oscillator (HS) | Primary | 10 | 010 | |
| Primary Oscillator (XT) | Primary | 01 | 010 | |
| Primary Oscillator (EC) | Primary | 00 | 010 | 1 |
| Fast RC Oscillator with PLL (FRCPLL) | Internal | xx | 001 | 1 |
| Fast RC Oscillator (FRC) | Internal | xx | 000 | 1 |

Note 1: OSC2 pin function is determined by the OSCIOFNC Configuration bit.

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

REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER⁽¹⁾

| | | | | | | | |
|--------|-----------|-----|-----|-------|--------------------------|-------|-------|
| U-0 | R-0 | R-0 | R-0 | U-0 | R/W-y | R/W-y | R/W-y |
| — | COSC<2:0> | | | — | NOSC<2:0> ⁽²⁾ | | |
| bit 15 | | | | bit 8 | | | |

| | | | | | | | |
|---------|--------|------|-----|-------|-----|---------|-------|
| R/W-0 | R/W-0 | R-0 | U-0 | R/C-0 | U-0 | R/W-0 | R/W-0 |
| CLKLOCK | IOLOCK | LOCK | — | CF | — | LPOSCEN | OSWEN |
| bit 7 | | | | | | bit 0 | |

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

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

bit 14-12 **COSC<2:0>:** Current Oscillator Selection bits (read-only)

- 000 = Fast RC oscillator (FRC)
- 001 = Fast RC oscillator (FRC) with PLL
- 010 = Primary oscillator (XT, HS, EC)
- 011 = Primary oscillator (XT, HS, EC) with PLL
- 100 = Secondary oscillator (SOSC)
- 101 = Low-Power RC oscillator (LPRC)
- 110 = Fast RC oscillator (FRC) with Divide-by-16
- 111 = Fast RC oscillator (FRC) with Divide-by-n

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

bit 10-8 **NOSC<2:0>:** New Oscillator Selection bits⁽²⁾

- 000 = Fast RC oscillator (FRC)
- 001 = Fast RC oscillator (FRC) with PLL
- 010 = Primary oscillator (XT, HS, EC)
- 011 = Primary oscillator (XT, HS, EC) with PLL
- 100 = Secondary oscillator (SOSC)
- 101 = Low-Power RC oscillator (LPRC)
- 110 = Fast RC oscillator (FRC) with Divide-by-16
- 111 = Fast RC oscillator (FRC) with Divide-by-n

bit 7 **CLKLOCK:** Clock Lock Enable bit

If clock switching is enabled and FSCM is disabled, (FOSC<FCKSM> = 0b01)

- 1 = Clock switching is disabled, system clock source is locked
- 0 = Clock switching is enabled, system clock source can be modified by clock switching

bit 6 **IOLOCK:** Peripheral Pin Select Lock bit

- 1 = Peripheral pin select is locked, write to peripheral pin select registers not allowed
- 0 = Peripheral pin select is not locked, write to peripheral pin select registers allowed

bit 5 **LOCK:** PLL Lock Status bit (read-only)

- 1 = Indicates that PLL is in lock, or PLL start-up timer is satisfied
- 0 = Indicates that PLL is out of lock, start-up timer is in progress or PLL is disabled

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

bit 3 **CF:** Clock Fail Detect bit (read/clear by application)

- 1 = FSCM has detected clock failure
- 0 = FSCM has not detected clock failure

Note 1: Writes to this register require an unlock sequence. Refer to **Section 39. "Oscillator (Part III)"** (DS70216) in the *"dsPIC33F Family Reference Manual"* (available from the Microchip website) for details.

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.

REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER⁽¹⁾ (CONTINUED)

- bit 2 **Unimplemented:** Read as '0'
- bit 1 **LPOSCEN:** Secondary (LP) Oscillator Enable bit
 1 = Enable secondary oscillator
 0 = Disable secondary oscillator
- bit 0 **OSWEN:** Oscillator Switch Enable bit
 1 = Request oscillator switch to selection specified by NOSC<2:0> bits
 0 = Oscillator switch is complete

- Note 1:** Writes to this register require an unlock sequence. Refer to **Section 39. "Oscillator (Part III)"** (DS70216) in the *"dsPIC33F Family Reference Manual"* (available from the Microchip website) for details.
- 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.

REGISTER 9-2: CLKDIV: CLOCK DIVISOR REGISTER

| | | | | | | | |
|--------|-----------|-------|-------|----------------------|-------------|-------|-------|
| R/W-0 | R/W-0 | R/W-1 | R/W-1 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| ROI | DOZE<2:0> | | | DOZEN ⁽¹⁾ | FRCDIV<2:0> | | |
| bit 15 | | | | bit 8 | | | |

| | | | | | | | |
|--------------|-------|-----|-------------|-------|-------|-------|-------|
| R/W-0 | R/W-1 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| PLLPOST<1:0> | | — | PLLPRE<4:0> | | | | |
| bit 7 | | | | | | | bit 0 |

| | | | |
|-------------------|--|------------------------------------|--------------------|
| Legend: | y = Value set from Configuration bits on POR | | |
| R = Readable bit | W = Writable 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 clears the DOZEN bit and the processor clock/peripheral clock ratio is set to 1:1
 0 = Interrupts have no effect on the DOZEN bit
- bit 14-12 **DOZE<2:0>:** Processor Clock Reduction Select bits
 000 = Fcy/1
 001 = Fcy/2
 010 = Fcy/4
 011 = Fcy/8 (default)
 100 = Fcy/16
 101 = Fcy/32
 110 = Fcy/64
 111 = Fcy/128
- bit 11 **DOZEN:** DOZE Mode Enable bit⁽¹⁾
 1 = DOZE<2:0> field specifies the ratio between the peripheral clocks and the processor clocks
 0 = Processor clock/peripheral clock ratio forced to 1:1
- bit 10-8 **FRCDIV<2:0>:** Internal Fast RC Oscillator Postscaler bits
 000 = FRC divide by 1 (default)
 001 = FRC divide by 2
 010 = FRC divide by 4
 011 = FRC divide by 8
 100 = FRC divide by 16
 101 = FRC divide by 32
 110 = FRC divide by 64
 111 = FRC divide by 256
- bit 7-6 **PLLPOST<1:0>:** PLL VCO Output Divider Select bits (also denoted as 'N2', PLL postscaler)
 00 = Output/2
 01 = Output/4 (default)
 10 = Reserved
 11 = Output/8
- bit 5 **Unimplemented:** Read as '0'
- bit 4-0 **PLLPRE<4:0>:** PLL Phase Detector Input Divider bits (also denoted as 'N1', PLL prescaler)
 00000 = Input/2 (default)
 00001 = Input/3
 •
 •
 •
 11111 = Input/33

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

REGISTER 9-3: PLLFBD: PLL FEEDBACK DIVISOR REGISTER

| | | | | | | | |
|--------|-----|-----|-----|-----|-----|-----|----------------------|
| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 ⁽¹⁾ |
| — | — | — | — | — | — | — | PLLDIV<8> |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------------|-------|-------|-------|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-1 | R/W-1 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| PLLDIV<7:0> | | | | | | | |
| 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 15-9 **Unimplemented:** Read as '0'

bit 8-0 **PLLDIV<8:0>:** PLL Feedback Divisor bits (also denoted as 'M', PLL multiplier)

- 000000000 = 2
- 000000001 = 3
- 000000010 = 4
-
-
-
- 000110000 = 50 (default)
-
-
-
- 111111111 = 513

REGISTER 9-4: OSCTUN: FRC OSCILLATOR TUNING 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 |
| — | — | TUN<5:0> ⁽¹⁾ | | | | | |
| bit 7 | | | | | | bit 0 | |

Legend:

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

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

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

011111 = Center frequency +11.625% (8.23 MHz)

011110 = Center frequency +11.25% (8.20 MHz)

•

•

•

000001 = Center frequency +0.375% (7.40 MHz)

000000 = Center frequency (7.37 MHz nominal)

111111 = Center frequency -0.375% (7.345 MHz)

•

•

•

100001 = Center frequency -11.625% (6.52 MHz)

100000 = Center frequency -12% (6.49 MHz)

Note 1: OSCTUN functionality has been provided to help customers compensate for temperature effects on the FRC frequency over a wide range of temperatures. The tuning step size is an approximation and is neither characterized nor tested.

REGISTER 9-5: ACLKCON: AUXILIARY CLOCK DIVISOR CONTROL REGISTER

| | | | | | | | | |
|--------|-----|---------|-------------|---------------|-------|-------|-------|-------|
| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | |
| — | — | SELACLK | AOSCMD<1:0> | APSTSCLR<2:0> | | | | |
| bit 15 | | | | | | | | bit 8 |

| | | | | | | | | |
|---------|-----|-----|-----|-----|-----|-----|-----|-------|
| R/W-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | |
| ASRCSEL | — | — | — | — | — | — | — | |
| bit 7 | | | | | | | | bit 0 |

| | | | |
|-------------------|------------------|------------------------------------|--------------------|
| Legend: | | | |
| R = Readable bit | W = Writable 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 **SELACLK:** Select Auxiliary Clock Source for Auxiliary Clock Divider
 - 1 = Auxiliary Oscillators provides the source clock for Auxiliary Clock Divider
 - 0 = PLL output (Fvco) provides the source clock for the Auxiliary Clock Divider
- bit 12-11 **AOSCMD<1:0>:** Auxiliary Oscillator Mode
 - 11 = EC External Clock Mode Select
 - 10 = XT Oscillator Mode Select
 - 01 = HS Oscillator Mode Select
 - 00 = Auxiliary Oscillator Disabled (default)
- bit 10-8 **APSTSCLR<2:0>:** Auxiliary Clock Output Divider
 - 111 = divided by 1
 - 110 = divided by 2
 - 101 = divided by 4
 - 100 = divided by 8
 - 011 = divided by 16
 - 010 = divided by 32
 - 001 = divided by 64
 - 000 = divided by 256 (default)
- bit 7 **ASRCSEL:** Select Reference Clock Source for Auxiliary Clock
 - 1 = Primary Oscillator is the Clock Source
 - 0 = Auxiliary Oscillator is the Clock Source
- bit 6-0 **Unimplemented:** Read as '0'

9.2 Clock Switching Operation

Applications are free to switch among any of the four clock sources (Primary, LP, FRC and LPRC) under software control at any time. To limit the possible side effects of this flexibility, dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices have a safeguard lock built into the switch process.

Note: Primary Oscillator mode has three different submodes (XT, HS and EC), which are determined by the POSCMD<1:0> Configuration bits. While an application can switch to and from Primary Oscillator mode in software, it cannot switch among the different primary submodes without reprogramming the device.

9.2.1 ENABLING CLOCK SWITCHING

To enable clock switching, the FCKSM1 Configuration bit in the Configuration register must be programmed to '0'. (Refer to **Section 28.1 "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 NOSC control bits (OSCCON<10:8>) do not control the clock selection when clock switching is disabled. However, the COSC bits (OSCCON<14:12>) reflect the clock source selected by the FNOSC Configuration bits.

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

9.2.2 OSCILLATOR SWITCHING SEQUENCE

Performing a clock switch requires this basic sequence:

1. If desired, read the COSC 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 NOSC control 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 (OSCCON<0>) 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 COSC status bits with the new value of the NOSC control bits. If they are the same, 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 the CF (OSCCON<3>) status 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 waits until the Oscillator Start-up Timer (OST) expires. If the new source is using the PLL, 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 NOSC bit values are transferred to the COSC status bits.
6. The old clock source is turned off at this time, with the exception of LPRC (if WDT or FSCM are enabled) or LP (if LPOSCEN remains set).

Note 1: The processor continues 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.

3: Refer to **Section 39. "Oscillator (Part III)"** (DS70216) in the *"dsPIC33F Family Reference Manual"* for details.

9.3 Fail-Safe Clock Monitor (FSCM)

The Fail-Safe Clock Monitor (FSCM) allows the device to continue to operate even in the event of an oscillator failure. The FSCM function is enabled by programming. If the FSCM function is enabled, the LPRC internal oscillator runs at all times (except during Sleep mode) and is not subject to control by the Watchdog Timer.

In the event of an oscillator failure, the FSCM generates a clock failure trap event and switches the system clock over to the FRC oscillator. Then the application program can either attempt to restart the oscillator or execute a controlled shutdown. The trap can be treated as a warm Reset by simply loading the Reset address into the oscillator fail trap vector.

If the PLL multiplier is used to scale the system clock, the internal FRC is also multiplied by the same factor on clock failure. Essentially, the device switches to FRC with PLL on a clock failure.

NOTES:

10.0 POWER-SAVING FEATURES

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 9. “Watchdog Timer and Power Savings Modes” (DS70196), which is available from the Microchip website (www.microchip.com).

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices provide 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.

dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices can manage power consumption in four 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.

10.1 Clock Frequency and Clock Switching

dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices allow 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 (OSCCON<10:8>). The process of changing a system clock during operation, as well as limitations to the process, are discussed in more detail in Section 9.0 “Oscillator Configuration”.

EXAMPLE 10-1: PWRSAV INSTRUCTION SYNTAX

```
PWRSAV #SLEEP_MODE ; Put the device into SLEEP mode
PWRSAV #IDLE_MODE ; Put the device into IDLE mode
```

10.2 Instruction-Based Power-Saving Modes

dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 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 assembler syntax of the PWRSAV instruction is shown in Example 10-1.

Note: SLEEP_MODE and IDLE_MODE are constants defined in the assembler include file for the selected device.

Sleep and Idle modes can be exited as a result of an enabled interrupt, WDT time-out or a device Reset. When the device exits these modes, it is said to wake up.

10.2.1 SLEEP MODE

The following occur in Sleep mode:

- The system clock source is shut down. If an on-chip oscillator is used, it is turned off.
- The device current consumption is reduced to a minimum, provided that no I/O pin is sourcing current.
- The Fail-Safe Clock Monitor does not operate, since the system clock source is disabled.
- The LPRC clock continues 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 can continue to operate. 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 is disabled.

The device wakes up from Sleep mode on any of the these events:

- Any interrupt source that is individually enabled
- Any form of device Reset
- A WDT time-out

On wake-up from Sleep mode, the processor restarts with the same clock source that was active when Sleep mode was entered.

10.2.2 IDLE MODE

The following occur in Idle mode:

- The CPU stops 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 10.4 “Peripheral Module Disable”**).
- If the WDT or FSCM is enabled, the LPRC also remains active.

The device wakes 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 mode, the clock is reapplied to the CPU and instruction execution will begin (2 to 4 cycles later), starting with the instruction following the `PWRSVAV` instruction, or the first instruction in the ISR.

10.2.3 INTERRUPTS COINCIDENT WITH POWER SAVE INSTRUCTIONS

Any interrupt that coincides with the execution of a `PWRSVAV` instruction is held off until entry into Sleep or Idle mode has completed. The device then wakes up from Sleep or Idle mode.

10.3 Doze Mode

The preferred strategies for reducing power consumption are changing clock speed and invoking one of the power-saving modes. In some circumstances, this cannot be 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 can introduce communication errors, while using a power-saving mode can stop communications completely.

Doze mode is a simple and effective alternative method to reduce power consumption while the device is still executing code. In this mode, the system clock continues to operate from the same source and at the same speed. Peripheral modules continue to be clocked at the same speed, while the CPU clock speed is reduced. Synchronization between the two clock domains is maintained, allowing the peripherals to access the SFRs while the CPU executes code at a slower rate.

Doze mode is enabled by setting the DOZEN bit (`CLKDIV<11>`). The ratio between peripheral and core clock speed is determined by the DOZE<2:0> bits (`CLKDIV<14:12>`). There are eight possible configurations, from 1:1 to 1:128, with 1:1 being the default setting.

Programs can 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. An automatic return to full-speed CPU operation on interrupts can be enabled by setting the ROI bit (`CLKDIV<15>`). By default, interrupt events have no effect on Doze mode operation.

For example, suppose the device is operating at 20 MIPS and the ECAN module has been configured for 500 kbps based on this device operating speed. If the device is placed in Doze mode with a clock frequency ratio of 1:4, the ECAN module continues to communicate at the required bit rate of 500 kbps, but the CPU now starts executing instructions at a frequency of 5 MIPS.

10.4 Peripheral Module Disable

The Peripheral Module Disable (PMD) registers provide a method to disable a peripheral module by stopping all clock sources supplied to that module. When a peripheral is disabled using the appropriate PMD control bit, the peripheral is in a minimum power consumption state. The control and status registers associated with the peripheral are also disabled, so writes to those registers do not have effect and read values are invalid.

A peripheral module is enabled only if both the associated bit in the PMD register is cleared and the peripheral is supported by the specific dsPIC® DSC variant. If the peripheral is present in the device, it is enabled in the PMD register by default.

Note: If a PMD bit is set, the corresponding module is disabled after a delay of one instruction cycle. Similarly, if a PMD bit is cleared, the corresponding module is enabled after a delay of one instruction cycle (assuming the module control registers are already configured to enable module operation).

REGISTER 10-1: PMD1: PERIPHERAL MODULE DISABLE 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 |
| T5MD | T4MD | T3MD | T2MD | T1MD | QE11MD | PWM1MD | — |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|--------|-------|-------|--------|--------|-----|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 |
| I2C1MD | U2MD | U1MD | SPI2MD | SPI1MD | — | C1MD | AD1MD |
| bit 7 | | | | | | | bit 0 |

Legend:

| | | |
|-------------------|------------------|--|
| R = Readable bit | W = Writable 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 **T5MD:** Timer5 Module Disable bit
1 = Timer5 module is disabled
0 = Timer5 module is enabled
- bit 14 **T4MD:** Timer4 Module Disable bit
1 = Timer4 module is disabled
0 = Timer4 module is enabled
- bit 13 **T3MD:** Timer3 Module Disable bit
1 = Timer3 module is disabled
0 = Timer3 module is enabled
- bit 12 **T2MD:** Timer2 Module Disable bit
1 = Timer2 module is disabled
0 = Timer2 module is enabled
- bit 11 **T1MD:** Timer1 Module Disable bit
1 = Timer1 module is disabled
0 = Timer1 module is enabled
- bit 10 **QE11MD:** QE11 Module Disable bit
1 = QE11 module is disabled
0 = QE11 module is enabled
- bit 9 **PWM1MD:** PWM1 Module Disable bit
1 = PWM1 module is disabled
0 = PWM1 module is enabled
- bit 8 **Unimplemented:** Read as '0'
- bit 7 **I2C1MD:** I²C1 Module Disable bit
1 = I²C1 module is disabled
0 = I²C1 module is enabled
- bit 6 **U2MD:** UART2 Module Disable bit
1 = UART2 module is disabled
0 = UART2 module is enabled
- bit 5 **U1MD:** UART1 Module Disable bit
1 = UART1 module is disabled
0 = UART1 module is enabled
- bit 4 **SPI2MD:** SPI2 Module Disable bit
1 = SPI2 module is disabled
0 = SPI2 module is enabled
- bit 3 **SPI1MD:** SPI1 Module Disable bit
1 = SPI1 module is disabled
0 = SPI1 module is enabled
- bit 2 **Unimplemented:** Read as '0'

REGISTER 10-1: PMD1: PERIPHERAL MODULE DISABLE CONTROL REGISTER 1 (CONTINUED)

bit 1 **C1MD:** ECAN1 Module Disable bit
 1 = ECAN1 module is disabled
 0 = ECAN1 module is enabled

bit 0 **AD1MD:** ADC1 Module Disable bit
 1 = ADC1 module is disabled
 0 = ADC1 module is enabled

REGISTER 10-2: PMD2: PERIPHERAL MODULE DISABLE CONTROL REGISTER 2

| | | | | | | | |
|--------|-------|-----|-----|-----|-----|-------|-------|
| R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 |
| IC8MD | IC7MD | — | — | — | — | IC2MD | IC1MD |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------|-----|-----|-----|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | — | OC4MD | OC3MD | OC2MD | OC1MD |
| bit 7 | | | | | | | bit 0 |

Legend:

R = Readable bit W = Writable 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 **IC8MD:** Input Capture 8 Module Disable bit
 1 = Input Capture 8 module is disabled
 0 = Input Capture 8 module is enabled
- bit 14 **IC7MD:** Input Capture 2 Module Disable bit
 1 = Input Capture 7 module is disabled
 0 = Input Capture 7 module is enabled
- bit 13-10 **Unimplemented:** Read as '0'
- bit 9 **IC2MD:** Input Capture 2 Module Disable bit
 1 = Input Capture 2 module is disabled
 0 = Input Capture 2 module is enabled
- bit 8 **IC1MD:** Input Capture 1 Module Disable bit
 1 = Input Capture 1 module is disabled
 0 = Input Capture 1 module is enabled
- bit 7-4 **Unimplemented:** Read as '0'
- bit 3 **OC4MD:** Output Compare 4 Module Disable bit
 1 = Output Compare 4 module is disabled
 0 = Output Compare 4 module is enabled
- bit 2 **OC3MD:** Output Compare 3 Module Disable bit
 1 = Output Compare 3 module is disabled
 0 = Output Compare 3 module is enabled
- bit 1 **OC2MD:** Output Compare 2 Module Disable bit
 1 = Output Compare 2 module is disabled
 0 = Output Compare 2 module is enabled
- bit 0 **OC1MD:** Output Compare 1 Module Disable bit
 1 = Output Compare 1 module is disabled
 0 = Output Compare 1 module is enabled

REGISTER 10-3: PMD3: PERIPHERAL MODULE DISABLE CONTROL REGISTER 3

| | | | | | | | |
|--------|-----|-----|-----|-----|-------|--------|-------|
| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | — | — | CMPMD | RTCCMD | PMPMD |
| bit 15 | | | | | | bit 8 | |

| | | | | | | | |
|-------|--------|--------|--------|-----|-----|-------|-----|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
| CRCMD | DAC1MD | QEI2MD | PWM2MD | — | — | — | — |
| bit 7 | | | | | | bit 0 | |

Legend:

R = Readable bit W = Writable 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 **CMPMD:** Comparator Module Disable bit
 1 = Comparator module is disabled
 0 = Comparator module is enabled
- bit 9 **RTCCMD:** RTCC Module Disable bit
 1 = RTCC module is disabled
 0 = RTCC module is enabled
- bit 8 **PMPMD:** PMP Module Disable bit
 1 = PMP module is disabled
 0 = PMP module is enabled
- bit 7 **CRCMD:** CRC Module Disable bit
 1 = CRC module is disabled
 0 = CRC module is enabled
- bit 6 **DAC1MD:** DAC1 Module Disable bit
 1 = DAC1 module is disabled
 0 = DAC1 module is enabled
- bit 5 **QEI2MD:** QEI2 Module Disable bit
 1 = QEI2 module is disabled
 0 = QEI2 module is enabled
- bit 4 **PWM2MD:** PWM2 Module Disable bit
 1 = PWM2 module is disabled
 0 = PWM2 module is enabled
- bit 3-0 **Unimplemented:** Read as '0'

11.0 I/O PORTS

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 10. “I/O Ports” (DS70193), which is available from the Microchip website (www.microchip.com).

All of the device pins (except VDD, VSS, MCLR and OSC1/CLKI) are shared among the peripherals and the parallel I/O ports. All I/O input ports feature Schmitt Trigger inputs for improved noise immunity.

11.1 Parallel I/O (PIO) Ports

Generally a parallel I/O port that shares a pin with a peripheral is 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 11-1 shows how ports are shared with other peripherals and the associated I/O pin to which they are connected.

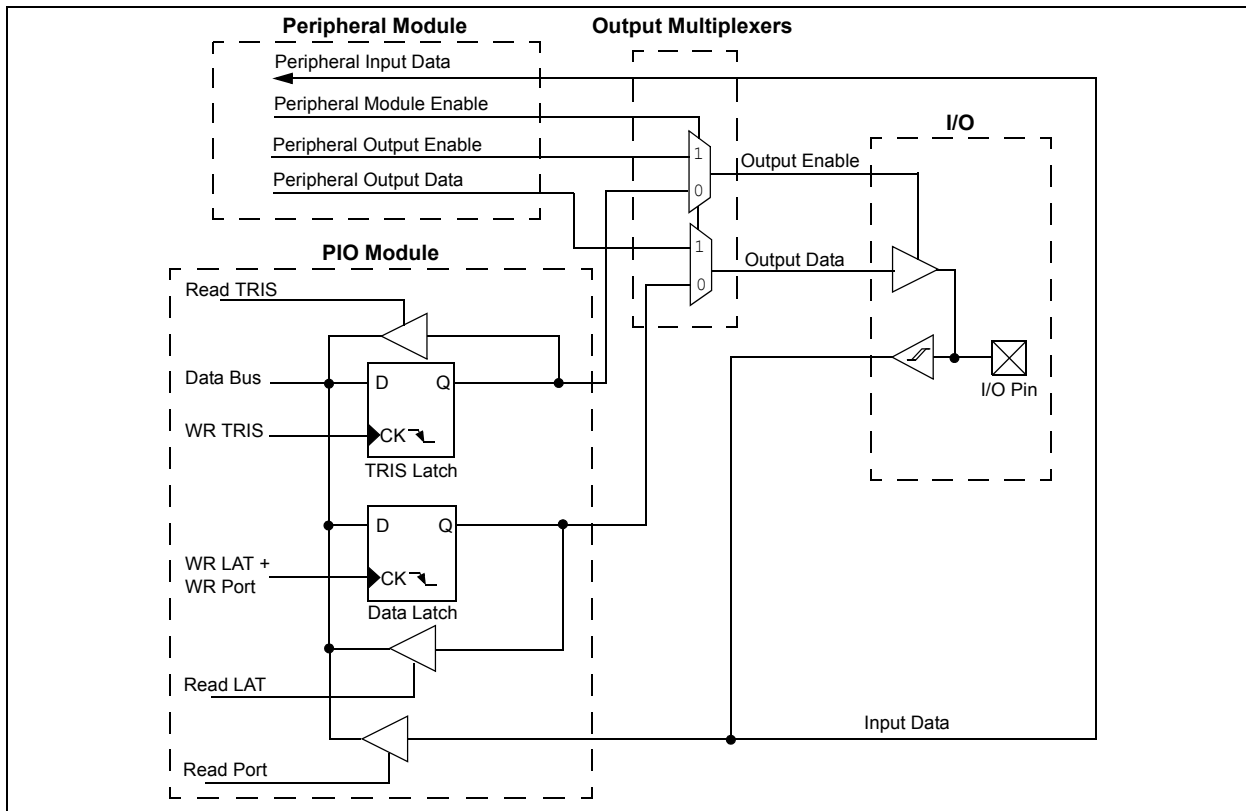
When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled. The I/O pin can be read, but the output driver for the parallel port bit is disabled. If a peripheral is enabled, but the peripheral is not actively driving a pin, that pin can 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 latch (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 is disabled. This means the corresponding LATx and TRISx registers and the port pin are 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 11-1: BLOCK DIAGRAM OF A TYPICAL SHARED PORT STRUCTURE



11.1.1 OPEN-DRAIN CONFIGURATION

In addition to the PORT, LAT and TRIS registers for data control, some port pins 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 V_{DD} (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 V_{IH} specification.

See “Pin Diagrams” for the available pins and their functionality.

11.2 Configuring Analog Port Pins

The AD1PCFGL and TRIS registers control the operation of the analog-to-digital (A/D) port pins. The port pins that are to function as analog inputs must have their corresponding TRIS bit set (input). If the TRIS bit is cleared (output), the digital output level (V_{OH} or V_{OL}) is converted.

The AD1PCFGL register has a default value of 0x0000; therefore, all pins that share ANx functions are analog (not digital) by default.

When the PORT register is read, all pins configured as analog input channels are read as cleared (a low level).

Pins configured as digital inputs do not convert an analog input. Analog levels on any pin defined as a digital input (including the ANx pins) can cause the input buffer to consume current that exceeds the device specifications.

11.2.1 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 an NOP, as shown in Example 11-1.

EXAMPLE 11-1: 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
```

11.3 Input Change Notification

The input change notification function of the I/O ports allows the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices to generate interrupt requests to the processor in response to a change-of-state on selected input pins. This feature can detect input change-of-states even in Sleep mode, when the clocks are disabled. Depending on the device pin count, up to 21 external signals (CNx pin) can be selected (enabled) for generating an interrupt request on a change-of-state.

Four control registers are 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 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.

Note: Pull-ups on change notification pins should always be disabled when the port pin is configured as a digital output.

11.4 Peripheral Pin Select

Peripheral pin select configuration enables peripheral set selection and placement on a wide range of I/O pins. By increasing the pinout options available on a particular device, programmers can better tailor the microcontroller to their entire application, rather than trimming the application to fit the device.

The peripheral pin select configuration feature operates over a fixed subset of digital I/O pins. Programmers can independently map the input and/or output of most 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.

11.4.1 AVAILABLE PINS

The peripheral pin select feature is used with a range of up to 26 pins. The number of available pins depends on the particular device and its pin count. Pins that support the peripheral pin select feature include the designation “RPn” in their full pin designation, where “RP” designates a remappable peripheral and “n” is the remappable pin number.

11.4.2 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 whether an input or output is being mapped.

11.4.2.1 Input Mapping

The inputs of the peripheral pin select options are mapped on the basis of the peripheral. A control register associated with a peripheral dictates the pin it is mapped to. The RPINRx registers are used to configure peripheral input mapping (see Register 11-1 through Register 11-20). Each register contains sets of 5-bit fields, with each set associated with one of the remappable 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 bit field corresponds to the maximum number of peripheral pin selections supported by the device.

Figure 11-2 Illustrates remappable pin selection for U1RX input.

Note: For input mapping only, the Peripheral Pin Select (PPS) functionality does not have priority over the TRISx settings. Therefore, when configuring the RPx pin for input, the corresponding bit in the TRISx register must also be configured for input (i.e., set to ‘1’).

FIGURE 11-2: REMAPPABLE MUX INPUT FOR U1RX

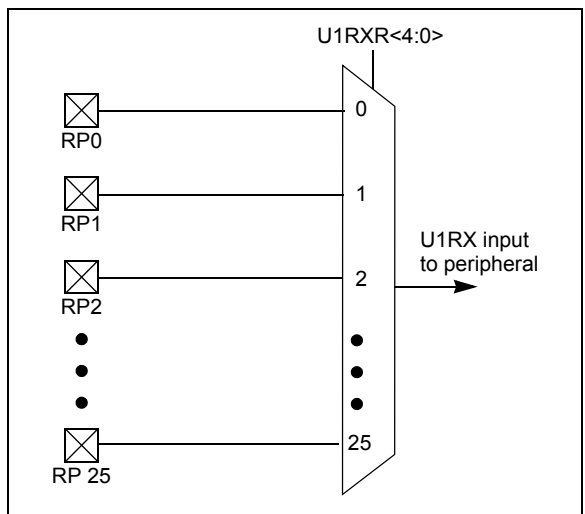


TABLE 11-1: SELECTABLE INPUT SOURCES (MAPS INPUT TO FUNCTION)⁽¹⁾

| Input Name | Function Name | Register | Configuration Bits |
|-------------------------|---------------------------|----------|--------------------|
| External Interrupt 1 | INT1 | RPINR0 | INT1R<4:0> |
| External Interrupt 2 | INT2 | RPINR1 | INT2R<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 7 | IC7 | RPINR10 | IC7R<4:0> |
| Input Capture 8 | IC8 | RPINR10 | IC8R<4:0> |
| Output Compare Fault A | OCFA | RPINR11 | OCFAR<4:0> |
| PWM1 Fault | $\overline{\text{FLTA1}}$ | RPINR12 | FLTA1R<4:0> |
| PWM2 Fault | $\overline{\text{FLTA2}}$ | RPINR13 | FLTA2R<4:0> |
| QE11 Phase A | QEA1 | RPINR14 | QEAIR<4:0> |
| QE11 Phase B | QEB1 | RPINR14 | QEBIR<4:0> |
| QE11 Index | INDX1 | RPINR15 | INDXIR<4:0> |
| QE12 Phase A | QEA2 | RPINR16 | QEA2R<4:0> |
| QE12Phase B | QEB2 | RPINR16 | QEB2R<4:0> |
| QE12 Index | INDX2 | RPINR17 | INDX2R<4:0> |
| UART1 Receive | $\overline{\text{U1RX}}$ | RPINR18 | U1RXR<4:0> |
| UART1 Clear To Send | $\overline{\text{U1CTS}}$ | RPINR18 | U1CTSR<4:0> |
| UART2 Receive | $\overline{\text{U2RX}}$ | RPINR19 | U2RXR<4:0> |
| UART2 Clear To Send | $\overline{\text{U2CTS}}$ | RPINR19 | U2CTSR<4:0> |
| SPI1 Data Input | SDI1 | RPINR20 | SDI1R<4:0> |
| SPI1 Clock Input | SCK1 | RPINR20 | SCK1R<4:0> |
| SPI1 Slave Select Input | $\overline{\text{SS1}}$ | RPINR21 | SS1R<4:0> |
| SPI2 Data Input | SDI2 | RPINR22 | SDI2R<4:0> |
| SPI2 Clock Input | SCK2 | RPINR22 | SCK2R<4:0> |
| SPI2 Slave Select Input | $\overline{\text{SS2}}$ | RPINR23 | SS2R<4:0> |
| ECAN1 Receive | CIRX | RPINR26 | CIRXR<4:0> |

Note 1: Unless otherwise noted, all inputs use Schmitt input buffers.

11.4.2.2 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 sets of 5-bit fields, with each set associated with one RPn pin (see Register 11-21 through Register 11-33). The value of the bit field corresponds to one of the peripherals, and that peripheral's output is mapped to the pin (see Table 11-2 and Figure 11-3).

The list of peripherals for output mapping also includes a null value of '00000' because of the mapping technique. This permits any given pin to remain unconnected from the output of any of the pin selectable peripherals.

FIGURE 11-3: MULTIPLEXING OF REMAPPABLE OUTPUT FOR RPn

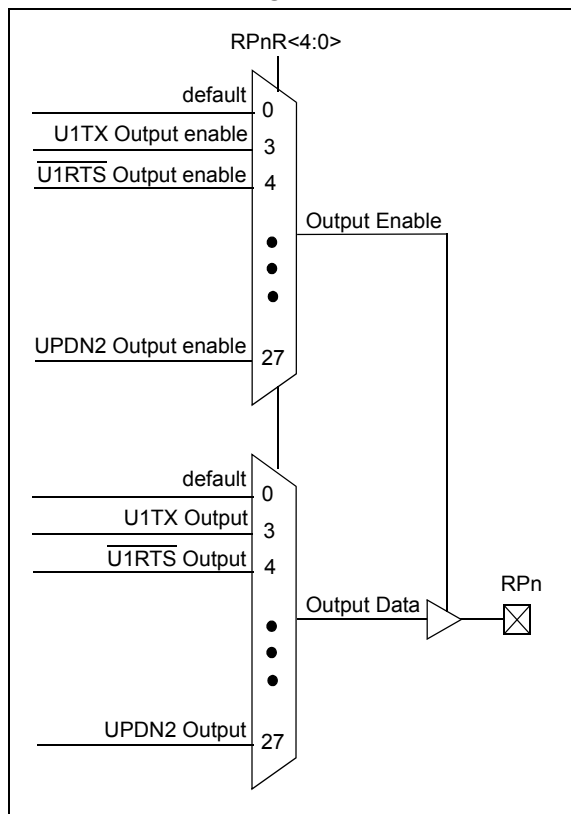


TABLE 11-2: OUTPUT SELECTION FOR REMAPPABLE PIN (RPn)

| Function | RPnR<4:0> | Output Name |
|--------------------|-----------|---|
| NULL | 00000 | RPn tied to default port pin |
| C1OUT | 00001 | RPn tied to Comparator1 Output |
| C2OUT | 00010 | RPn tied to Comparator2 Output |
| U1TX | 00011 | RPn tied to UART1 Transmit |
| $\overline{U1RTS}$ | 00100 | RPn tied to UART1 Ready To Send |
| U2TX | 00101 | RPn tied to UART2 Transmit |
| $\overline{U2RTS}$ | 00110 | RPn tied to UART2 Ready To Send |
| SDO1 | 00111 | RPn tied to SPI1 Data Output |
| SCK1 | 01000 | RPn tied to SPI1 Clock Output |
| $\overline{SS1}$ | 01001 | RPn tied to SPI1 Slave Select Output |
| SDO2 | 01010 | RPn tied to SPI2 Data Output |
| SCK2 | 01011 | RPn tied to SPI2 Clock Output |
| $\overline{SS2}$ | 01100 | RPn tied to SPI2 Slave Select Output |
| C1TX | 10000 | RPn tied to ECAN1 Transmit |
| OC1 | 10010 | RPn tied to Output Compare 1 |
| OC2 | 10011 | RPn tied to Output Compare 2 |
| OC3 | 10100 | RPn tied to Output Compare 3 |
| OC4 | 10101 | RPn tied to Output Compare 4 |
| UPDN1 | 11010 | RPn tied to QE1 direction (UPDN) status |
| UPDN2 | 11011 | RPn tied to QE2 direction (UPDN) status |

11.4.3 CONTROLLING CONFIGURATION CHANGES

Because peripheral remapping can be changed during run time, some restrictions on peripheral remapping are needed to prevent accidental configuration changes. dsPIC33F devices include three features to prevent alterations to the peripheral map:

- Control register lock sequence
- Continuous state monitoring
- Configuration bit pin select lock

11.4.3.1 Control Register Lock

Under normal operation, writes to the RPINRx and RPORx registers are not allowed. Attempted writes appear to execute normally, but the contents of the registers 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 0x46 to OSCCON<7:0>.
2. Write 0x57 to OSCCON<7:0>.
3. Clear (or set) IOLOCK as a single operation.

Note: MPLAB® C30 provides built-in C language functions for unlocking the OSCCON register:

```
__builtin_write_OSCCONL(value)  
__builtin_write_OSCCONH(value)
```

See MPLAB IDE Help for more information.

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.

11.4.3.2 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 is triggered.

11.4.3.3 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 (FOSC<IOL1WAY>) configuration bit blocks the IOLOCK bit from being cleared after it has been set once. If IOLOCK remains set, the register unlock procedure does 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 user applications unlimited access (with the proper use of the unlock sequence) to the peripheral pin select registers.

11.5 Peripheral Pin Select Registers

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 family of devices implement 33 registers for remappable peripheral configuration:

- 20 Input Remappable Peripheral Registers:
 - RPINR0-RPINR1, RPINR3-RPINR4, RPINR7, RPINR10-RPINR21, PRINR23, and PRINR26
- 13 Output Remappable Peripheral Registers:
 - RPOR0-RPOR12

Note: Input and Output Register values can only be changed if the IOLOCK bit (OSCCON<6>) is set to '0'. See **Section 11.4.3.1 “Control Register Lock”** for a specific command sequence.

REGISTER 11-1: 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 | |
| — | — | — | INT1R<4:0> | | | | | |
| bit 15 | | | | | | | | bit 8 |

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

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

- bit 15-13 **Unimplemented:** Read as '0'
- bit 12-8 **INT1R<4:0>:** Assign External Interrupt 1 (INTR1) to the corresponding RPn pin
 - 11111 = Input tied to Vss
 - 11001 = Input tied to RP25
 -
 -
 -
 - 00001 = Input tied to RP1
 - 00000 = Input tied to RP0
- bit 7-0 **Unimplemented:** Read as '0'

REGISTER 11-2: 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 |
| — | — | — | INTR2R<4:0> | | | | |
| 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 15-5 **Unimplemented:** Read as '0'

bit 4-0 **INTR2R<4:0>:** Assign External Interrupt 2 (INTR2) to the corresponding RPn pin

11111 = Input tied to Vss

11001 = Input tied to RP25

•

•

•

00001 = Input tied to RP1

00000 = Input tied to RP0

REGISTER 11-3: 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 |
| — | — | — | T3CKR<4:0> | | | | |
| 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 |
| — | — | — | T2CKR<4:0> | | | | |
| 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 **T3CKR<4:0>:** Assign Timer3 External Clock (T3CK) to the corresponding RPN pin

11111 = Input tied to Vss
11001 = Input tied to RP25

-
-
-

00001 = Input tied to RP1
00000 = Input tied to RP0

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

bit 4-0 **T2CKR<4:0>:** Assign Timer2 External Clock (T2CK) to the corresponding RPN pin

11111 = Input tied to Vss
11001 = Input tied to RP25

-
-
-

00001 = Input tied to RP1
00000 = Input tied to RP0

REGISTER 11-4: 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 |
| — | — | — | T5CKR<4:0> | | | | |
| 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 |
| — | — | — | T4CKR<4:0> | | | | |
| 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 **T5CKR<4:0>:** Assign Timer5 External Clock (T5CK) to the corresponding RPn pin

11111 = Input tied to Vss
 11001 = Input tied to RP25

-
-
-

00001 = Input tied to RP1
 00000 = Input tied to RP0

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

bit 4-0 **T4CKR<4:0>:** Assign Timer4 External Clock (T4CK) to the corresponding RPn pin

11111 = Input tied to Vss
 11001 = Input tied to RP25

-
-
-

00001 = Input tied to RP1
 00000 = Input tied to RP0

REGISTER 11-5: 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 |
| — | — | — | IC2R<4:0> | | | | |
| 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 |
| — | — | — | IC1R<4:0> | | | | |
| 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 **IC2R<4:0>:** Assign Input Capture 2 (IC2) to the corresponding RPn pin
 - 11111 = Input tied to Vss
 - 11001 = Input tied to RP25
 -
 -
 -
 - 00001 = Input tied to RP1
 - 00000 = Input tied to RP0
- bit 7-5 **Unimplemented:** Read as '0'
- bit 4-0 **IC1R<4:0>:** Assign Input Capture 1 (IC1) to the corresponding RPn pin
 - 11111 = Input tied to Vss
 - 11001 = Input tied to RP25.
 -
 -
 -
 - 00001 = Input tied to RP1
 - 00000 = Input tied to RP0

REGISTER 11-6: RPINR10: PERIPHERAL PIN SELECT INPUT REGISTERS 10

| | | | | | | | |
|--------|-----|-----|-----------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
| — | — | — | IC8R<4:0> | | | | |
| 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 |
| — | — | — | IC7R<4:0> | | | | |
| 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 **IC8R<4:0>:** Assign Input Capture 8 (IC8) to the corresponding pin RPn pin

11111 = Input tied to Vss
11001 = Input tied to RP25

-
-
-

00001 = Input tied to RP1
00000 = Input tied to RP0

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

bit 4-0 **IC7R<4:0>:** Assign Input Capture 7 (IC7) to the corresponding pin RPn pin

11111 = Input tied to Vss
11001 = Input tied to RP25

-
-
-

00001 = Input tied to RP1
00000 = Input tied to RP0

REGISTER 11-7: RPINR11: PERIPHERAL PIN SELECT INPUT REGISTER 11

| | | | | | | | |
|--------|-----|-----|-----|-----|-----|-----|-------|
| 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 | |
| — | — | — | OCFAR<4:0> | | | | | |
| 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 15-5 **Unimplemented:** Read as '0'
 bit 4-0 **OCFAR<4:0>:** Assign Output Compare A (OCFA) to the corresponding RPn pin
 11111 = Input tied to Vss
 11001 = Input tied to RP25
 •
 •
 •
 00001 = Input tied to RP1
 00000 = Input tied to RP0

REGISTER 11-8: RPINR12: PERIPHERAL PIN SELECT INPUT REGISTER 12

| | | | | | | | |
|--------|-----|-----|-----|-----|-----|-----|-------|
| 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 | |
| — | — | — | FLTA1R<4:0> | | | | | |
| 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 15-5 **Unimplemented:** Read as '0'
 bit 4-0 **FLTA1R<4:0>:** Assign PWM1 Fault ($\overline{\text{FLTA1}}$) to the corresponding RPn pin
 11111 = Input tied to Vss
 11001 = Input tied to RP25
 •
 •
 •
 00001 = Input tied to RP1
 00000 = Input tied to RP0

REGISTER 11-9: RPINR13: PERIPHERAL PIN SELECT INPUT REGISTER 13

| | | | | | | | |
|--------|-----|-----|-----|-----|-----|-----|-------|
| 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 |
| — | — | — | FLTA2R<4:0> | | | | |
| 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 15-5 **Unimplemented:** Read as '0'

bit 4-0 **FLTA2R<4:0>:** Assign PWM2 Fault ($\overline{\text{FLTA2}}$) to the corresponding RPn pin

11111 = Input tied to Vss

11001 = Input tied to RP25

-
-
-

00001 = Input tied to RP1

00000 = Input tied to RP0

REGISTER 11-10: RPINR14: PERIPHERAL PIN SELECT INPUT REGISTERS 14

| | | | | | | | |
|--------|-----|-----|------------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
| — | — | — | QEB1R<4:0> | | | | |
| 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 |
| — | — | — | QEA1R<4:0> | | | | |
| 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 **QEB1R<4:0>:** Assign B (QEB1) to the corresponding pin
 - 11111 = Input tied to Vss
 - 11001 = Input tied to RP25
 -
 -
 -
 - 00001 = Input tied to RP1
 - 00000 = Input tied to RP0
- bit 7-5 **Unimplemented:** Read as '0'
- bit 4-0 **QEA1R<4:0>:** Assign A(QEA1) to the corresponding pin
 - 11111 = Input tied to Vss
 - 11001 = Input tied to RP25
 -
 -
 -
 - 00001 = Input tied to RP1
 - 00000 = Input tied to RP0

REGISTER 11-11: RPNR15: PERIPHERAL PIN SELECT INPUT REGISTER 15

| | | | | | | | |
|--------|-----|-----|-----|-----|-----|-----|-------|
| 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 |
| — | — | — | INDX1R<4:0> | | | | |
| 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 15-5 **Unimplemented:** Read as '0'
- bit 4-0 **INDX1R<4:0>:** Assign QE11 INDEX (INDX1) to the corresponding RPN pin
 - 11111 = Input tied to Vss
 - 11001 = Input tied to RP25
 -
 -
 -
 - 00001 = Input tied to RP1
 - 00000 = Input tied to RP0

REGISTER 11-12: RPINR16: PERIPHERAL PIN SELECT INPUT REGISTERS 16

| | | | | | | | |
|--------|-----|-----|------------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
| — | — | — | QEB2R<4:0> | | | | |
| 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 |
| — | — | — | QEA2R<4:0> | | | | |
| 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 **QEB2R<4:0>:** Assign B (QEB2) to the corresponding pin

- 11111 = Input tied to Vss
- 11001 = Input tied to RP25
-
-
-
- 00001 = Input tied to RP1
- 00000 = Input tied to RP0

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

bit 4-0 **QEA2R<4:0>:** Assign A(QEA2) to the corresponding pin

- 11111 = Input tied to Vss
- 11001 = Input tied to RP25
-
-
-
- 00001 = Input tied to RP1
- 00000 = Input tied to RP0

REGISTER 11-13: RPINR17: PERIPHERAL PIN SELECT INPUT REGISTER 17

| | | | | | | | |
|--------|-----|-----|-----|-----|-----|-----|-------|
| 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 |
| — | — | — | INDX2R<4:0> | | | | |
| 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 15-5 **Unimplemented:** Read as '0'
- bit 4-0 **INDX2R<4:0>:** Assign QEI2 INDEX (INDX2) to the corresponding RPn pin
 - 11111 = Input tied to Vss
 - 11001 = Input tied to RP25
 -
 -
 -
 - 00001 = Input tied to RP1
 - 00000 = Input tied to RP0

REGISTER 11-14: 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 |
| — | — | — | U1CTSR<4:0> | | | | |
| 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 |
| — | — | — | U1RXR<4:0> | | | | |
| 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 **U1CTSR<4:0>:** Assign UART1 Clear to Send (U1CTS) to the corresponding RPn pin

11111 = Input tied to Vss
11001 = Input tied to RP25

•
•
•

00001 = Input tied to RP1
00000 = Input tied to RP0

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

bit 4-0 **U1RXR<4:0>:** Assign UART1 Receive (U1RX) to the corresponding RPn pin

11111 = Input tied to Vss
11001 = Input tied to RP25

•
•
•

00001 = Input tied to RP1
00000 = Input tied to RP0

REGISTER 11-15: 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 |
| — | — | — | U2CTSR<4:0> | | | | |
| 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 |
| — | — | — | U2RXR<4:0> | | | | |
| 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 **U2CTSR<4:0>:** Assign UART2 Clear to Send ($\overline{U2CTS}$) to the corresponding RPn pin

11111 = Input tied to Vss
 11001 = Input tied to RP25

-
-
-

00001 = Input tied to RP1
 00000 = Input tied to RP0

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

bit 4-0 **U2RXR<4:0>:** Assign UART2 Receive (U2RX) to the corresponding RPn pin

11111 = Input tied to Vss
 11001 = Input tied to RP25

-
-
-

00001 = Input tied to RP1
 00000 = Input tied to RP0

REGISTER 11-16: 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 |
| — | — | — | SCK1R<4:0> | | | | |
| 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 |
| — | — | — | SDI1R<4:0> | | | | |
| 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 **SCK1R<4:0>:** Assign SPI1 Clock Input (SCK1) to the corresponding RPn pin

11111 = Input tied to Vss
11001 = Input tied to RP25

•
•
•

00001 = Input tied to RP1
00000 = Input tied to RP0

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

bit 4-0 **SDI1R<4:0>:** Assign SPI1 Data Input (SDI1) to the corresponding RPn pin

11111 = Input tied to Vss
11001 = Input tied to RP25

•
•
•

00001 = Input tied to RP1
00000 = Input tied to RP0

REGISTER 11-17: 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 |
| — | — | — | SS1R<4:0> | | | | |
| 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 15-5 **Unimplemented:** Read as '0'

bit 4-0 **SS1R<4:0>:** Assign SPI1 Slave Select Input (SS1) to the corresponding RPn pin

11111 = Input tied to Vss

11001 = Input tied to RP25

-
-
-

00001 = Input tied to RP1

00000 = Input tied to RP0

REGISTER 11-18: 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 |
| — | — | — | SCK2R<4:0> | | | | |
| 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 |
| — | — | — | SDI2R<4:0> | | | | |
| 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 **SCK2R<4:0>:** Assign SPI2 Clock Input (SCK2) to the corresponding RPn pin
 - 11111 = Input tied to Vss
 - 11001 = Input tied to RP25
 -
 -
 -
 - 00001 = Input tied to RP1
 - 00000 = Input tied to RP0
- bit 7-5 **Unimplemented:** Read as '0'
- bit 4-0 **SDI2R<4:0>:** Assign SPI2 Data Input (SDI2) to the corresponding RPn pin
 - 11111 = Input tied to Vss
 - 11001 = Input tied to RP25
 -
 -
 -
 - 00001 = Input tied to RP1
 - 00000 = Input tied to RP0

REGISTER 11-19: 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 |
| — | — | — | SS2R<4:0> | | | | |
| 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 15-5 **Unimplemented:** Read as '0'
 bit 4-0 **SS2R<4:0>:** Assign SPI2 Slave Select Input (SS2) to the corresponding RPn pin
 11111 = Input tied to Vss
 11001 = Input tied to RP25
 •
 •
 •
 00001 = Input tied to RP1
 00000 = Input tied to RP0

REGISTER 11-20: RPINR26: PERIPHERAL PIN SELECT INPUT REGISTER 26⁽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 |
| — | — | — | C1RXR<4:0> | | | | |
| 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 15-5 **Unimplemented:** Read as '0'
 bit 4-0 **C1RXR<4:0>:** Assign ECAN1Receive (C1RX) to the corresponding RPn pin
 11111 = Input tied to Vss
 11001 = Input tied to RP25
 •
 •
 •
 00001 = Input tied to RP1
 00000 = Input tied to RP0

Note 1: This register is disabled on devices without ECAN™.

REGISTER 11-21: RPOR0: PERIPHERAL PIN SELECT OUTPUT REGISTERS 0

| | | | | | | | |
|--------|-----|-----|-----------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | RP1R<4:0> | | | | |
| 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 |
| — | — | — | RP0R<4:0> | | | | |
| 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 **RP1R<4:0>:** Peripheral Output Function is Assigned to RP1 Output Pin bits (see Table 11-2 for peripheral function numbers)
- bit 7-5 **Unimplemented:** Read as '0'
- bit 4-0 **RP0R<4:0>:** Peripheral Output Function is Assigned to RP0 Output Pin bits (see Table 11-2 for peripheral function numbers)

REGISTER 11-22: RPOR1: PERIPHERAL PIN SELECT OUTPUT REGISTERS 1

| | | | | | | | |
|--------|-----|-----|-----------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | RP3R<4:0> | | | | |
| 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 |
| — | — | — | RP2R<4:0> | | | | |
| 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 **RP3R<4:0>:** Peripheral Output Function is Assigned to RP3 Output Pin bits (see Table 11-2 for peripheral function numbers)
- bit 7-5 **Unimplemented:** Read as '0'
- bit 4-0 **RP2R<4:0>:** Peripheral Output Function is Assigned to RP2 Output Pin bits (see Table 11-2 for peripheral function numbers)

REGISTER 11-23: RPOR2: PERIPHERAL PIN SELECT OUTPUT REGISTERS 2

| | | | | | | | |
|--------|-----|-----|-----------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | RP5R<4:0> | | | | |
| 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 |
| — | — | — | RP4R<4:0> | | | | |
| 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 **RP5R<4:0>:** Peripheral Output Function is Assigned to RP5 Output Pin bits (see Table 11-2 for peripheral function numbers)
- bit 7-5 **Unimplemented:** Read as '0'
- bit 4-0 **RP4R<4:0>:** Peripheral Output Function is Assigned to RP4 Output Pin bits (see Table 11-2 for peripheral function numbers)

REGISTER 11-24: RPOR3: PERIPHERAL PIN SELECT OUTPUT REGISTERS 3

| | | | | | | | |
|--------|-----|-----|-----------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | RP7R<4:0> | | | | |
| 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 |
| — | — | — | RP6R<4:0> | | | | |
| 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 **RP7R<4:0>:** Peripheral Output Function is Assigned to RP7 Output Pin bits (see Table 11-2 for peripheral function numbers)
- bit 7-5 **Unimplemented:** Read as '0'
- bit 4-0 **RP6R<4:0>:** Peripheral Output Function is Assigned to RP6 Output Pin bits (see Table 11-2 for peripheral function numbers)

REGISTER 11-25: RPOR4: PERIPHERAL PIN SELECT OUTPUT REGISTERS 0

| | | | | | | | |
|--------|-----|-----|-----------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | RP9R<4:0> | | | | |
| 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 |
| — | — | — | RP8R<4:0> | | | | |
| 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 **RP9R<4:0>:** Peripheral Output Function is Assigned to RP9 Output Pin bits (see Table 11-2 for peripheral function numbers)
- bit 7-5 **Unimplemented:** Read as '0'
- bit 4-0 **RP8R<4:0>:** Peripheral Output Function is Assigned to RP8 Output Pin bits (see Table 11-2 for peripheral function numbers)

REGISTER 11-26: RPOR5: PERIPHERAL PIN SELECT OUTPUT REGISTERS 5

| | | | | | | | |
|--------|-----|-----|------------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | RP11R<4:0> | | | | |
| 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 |
| — | — | — | RP10R<4:0> | | | | |
| 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 **RP11R<4:0>:** Peripheral Output Function is Assigned to RP11 Output Pin bits (see Table 11-2 for peripheral function numbers)
- bit 7-5 **Unimplemented:** Read as '0'
- bit 4-0 **RP10R<4:0>:** Peripheral Output Function is Assigned to RP10 Output Pin bits (see Table 11-2 for peripheral function numbers)

REGISTER 11-27: RPOR6: PERIPHERAL PIN SELECT OUTPUT REGISTERS 6

| | | | | | | | |
|--------|-----|-----|------------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | RP13R<4:0> | | | | |
| 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 |
| — | — | — | RP12R<4:0> | | | | |
| 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 **RP13R<4:0>:** Peripheral Output Function is Assigned to RP13 Output Pin bits (see Table 11-2 for peripheral function numbers)
- bit 7-5 **Unimplemented:** Read as '0'
- bit 4-0 **RP12R<4:0>:** Peripheral Output Function is Assigned to RP12 Output Pin bits (see Table 11-2 for peripheral function numbers)

REGISTER 11-28: RPOR7: PERIPHERAL PIN SELECT OUTPUT REGISTERS 7

| | | | | | | | |
|--------|-----|-----|------------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | RP15R<4:0> | | | | |
| 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 |
| — | — | — | RP14R<4:0> | | | | |
| 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 **RP15R<4:0>:** Peripheral Output Function is Assigned to RP15 Output Pin bits (see Table 11-2 for peripheral function numbers)
- bit 7-5 **Unimplemented:** Read as '0'
- bit 4-0 **RP14R<4:0>:** Peripheral Output Function is Assigned to RP14 Output Pin bits (see Table 11-2 for peripheral function numbers)

REGISTER 11-29: RPOR8: PERIPHERAL PIN SELECT OUTPUT REGISTERS 8⁽¹⁾

| | | | | | | | |
|--------|-----|-----|------------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | RP17R<4:0> | | | | |
| 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 |
| — | — | — | RP16R<4:0> | | | | |
| 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 **RP17R<4:0>:** Peripheral Output Function is Assigned to RP17 Output Pin bits (see Table 11-2 for peripheral function numbers)
- bit 7-5 **Unimplemented:** Read as '0'
- bit 4-0 **RP16R<4:0>:** Peripheral Output Function is Assigned to RP16 Output Pin bits (see Table 11-2 for peripheral function numbers)

Note 1: This register is implemented in 44-pin devices only.

REGISTER 11-30: RPOR9: PERIPHERAL PIN SELECT OUTPUT REGISTERS 9⁽¹⁾

| | | | | | | | |
|--------|-----|-----|------------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | RP19R<4:0> | | | | |
| 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 |
| — | — | — | RP18R<4:0> | | | | |
| 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 **RP19R<4:0>:** Peripheral Output Function is Assigned to RP19 Output Pin bits (see Table 11-2 for peripheral function numbers)
- bit 7-5 **Unimplemented:** Read as '0'
- bit 4-0 **RP18R<4:0>:** Peripheral Output Function is Assigned to RP18 Output Pin bits (see Table 11-2 for peripheral function numbers)

Note 1: This register is implemented in 44-pin devices only.

REGISTER 11-31: RPOR10: PERIPHERAL PIN SELECT OUTPUT REGISTERS 10⁽¹⁾

| | | | | | | | |
|--------|-----|-----|------------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | RP21R<4:0> | | | | |
| 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 |
| — | — | — | RP20R<4:0> | | | | |
| 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 **RP21R<4:0>:** Peripheral Output Function is Assigned to RP21 Output Pin bits (see Table 11-2 for peripheral function numbers)
- bit 7-5 **Unimplemented:** Read as '0'
- bit 4-0 **RP20R<4:0>:** Peripheral Output Function is Assigned to RP20 Output Pin bits (see Table 11-2 for peripheral function numbers)

Note 1: This register is implemented in 44-pin devices only.

REGISTER 11-32: RPOR11: PERIPHERAL PIN SELECT OUTPUT REGISTERS 11⁽¹⁾

| | | | | | | | |
|--------|-----|-----|------------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | RP23R<4:0> | | | | |
| 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 |
| — | — | — | RP22R<4:0> | | | | |
| 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 **RP23R<4:0>:** Peripheral Output Function is Assigned to RP23 Output Pin bits (see Table 11-2 for peripheral function numbers)
- bit 7-5 **Unimplemented:** Read as '0'
- bit 4-0 **RP22R<4:0>:** Peripheral Output Function is Assigned to RP22 Output Pin bits (see Table 11-2 for peripheral function numbers)

Note 1: This register is implemented in 44-pin devices only.

REGISTER 11-33: RPOR12: PERIPHERAL PIN SELECT OUTPUT REGISTERS 12⁽¹⁾

| | | | | | | | |
|--------|-----|-----|------------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | RP25R<4:0> | | | | |
| 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 |
| — | — | — | RP24R<4:0> | | | | |
| 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 **RP25R<4:0>:** Peripheral Output Function is Assigned to RP25 Output Pin bits (see Table 11-2 for peripheral function numbers)
- bit 7-5 **Unimplemented:** Read as '0'
- bit 4-0 **RP24R<4:0>:** Peripheral Output Function is Assigned to RP24 Output Pin bits (see Table 11-2 for peripheral function numbers)

Note 1: This register is implemented in 44-pin devices only.

NOTES:

12.0 TIMER1

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 11. “Timers” (DS70205), which is available from the Microchip website (www.microchip.com).

The Timer1 module is a 16-bit timer, which can serve as the time counter for the real-time clock, or operate as a free-running interval timer/counter.

The Timer1 module has the following unique features over other timers:

- Can be operated from the low power 32 kHz crystal oscillator available on the device
- Can be operated in Asynchronous Counter mode from an external clock source.
- The external clock input (T1CK) can optionally be synchronized to the internal device clock and the clock synchronization is performed after the prescaler.

The unique features of Timer1 allow it to be used for Real Time Clock (RTC) applications. A block diagram of Timer1 is shown in Figure 12-1.

The Timer1 module can operate in one of the following modes:

- Timer mode
- Gated Timer mode
- Synchronous Counter mode
- Asynchronous Counter mode

In Timer and Gated Timer modes, the input clock is derived from the internal instruction cycle clock (FCY). In Synchronous and Asynchronous Counter modes, the input clock is derived from the external clock input at the T1CK pin.

The Timer modes are determined by the following bits:

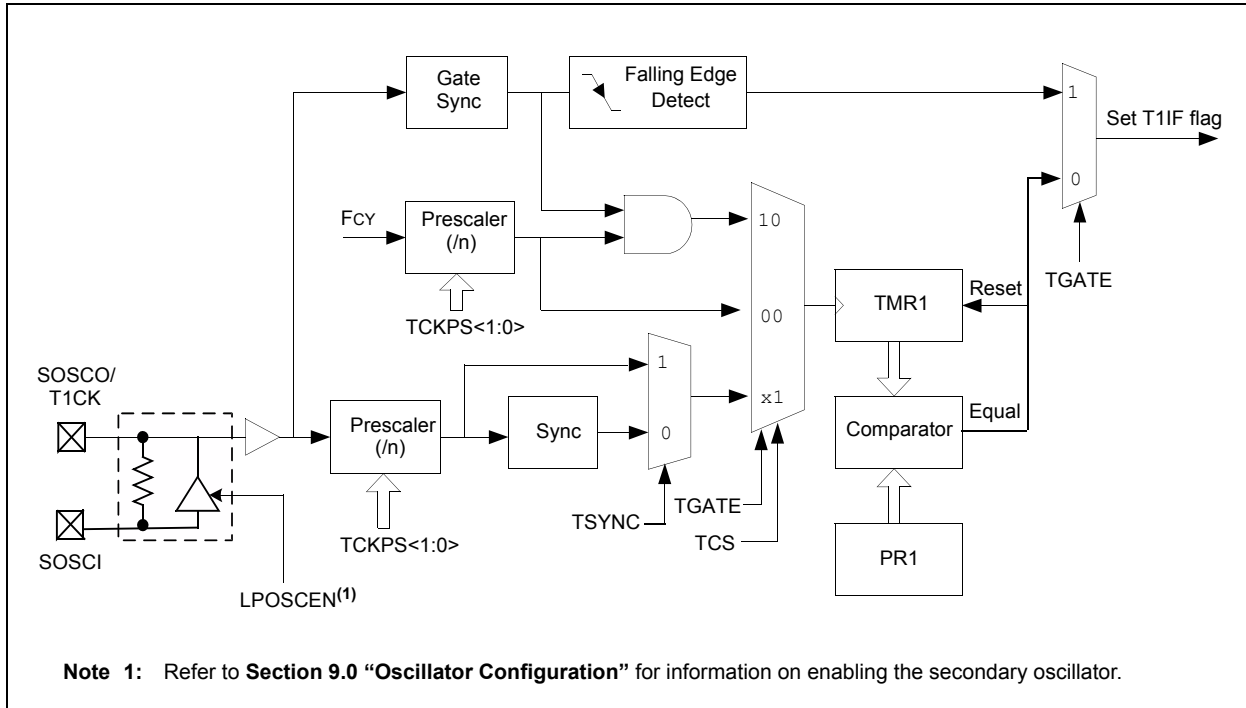
- Timer Clock Source Control bit (TCS): T1CON<1>
- Timer Synchronization Control bit (TSYNC): T1CON<2>
- Timer Gate Control bit (TGATE): T1CON<6>

Timer control bit setting for different operating modes are given in the Table 12-1.

TABLE 12-1: TIMER MODE SETTINGS

| Mode | TCS | TGATE | TSYNC |
|----------------------|-----|-------|-------|
| Timer | 0 | 0 | x |
| Gated timer | 0 | 1 | x |
| Synchronous counter | 1 | x | 1 |
| Asynchronous counter | 1 | x | 0 |

FIGURE 12-1: 16-BIT TIMER1 MODULE BLOCK DIAGRAM



REGISTER 12-1: 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 | TCKPS<1:0> | | — | 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 T1CS = 1:
 This bit is ignored.
 When T1CS = 0:
 1 = Gated time accumulation enabled
 0 = Gated time accumulation disabled
- bit 5-4 **TCKPS<1:0>** Timer1 Input Clock Prescale Select bits
 11 = 1:256
 10 = 1:64
 01 = 1:8
 00 = 1:1
- bit 3 **Unimplemented:** Read as '0'
- bit 2 **TSYNC:** Timer1 External Clock Input Synchronization Select bit
 When TCS = 1:
 1 = Synchronize external clock input
 0 = Do not synchronize external clock input
 When TCS = 0:
 This bit is ignored.
- bit 1 **TCS:** Timer1 Clock Source Select bit
 1 = External clock from pin T1CK (on the rising edge)
 0 = Internal clock (Fcy)
- bit 0 **Unimplemented:** Read as '0'

The Timer2/3 and Timer4/5 modules can operate in one of the following modes:

- Timer mode
- Gated Timer mode
- Synchronous Counter mode

In Timer and Gated Timer modes, the input clock is derived from the internal instruction cycle clock (FCY). In Synchronous Counter mode, the input clock is derived from the external clock input at TxCK pin.

The timer modes are determined by the following bits:

- TCS (TxCON<1>): Timer Clock Source Control bit
- TGATE (TxCON<6>): Timer Gate Control bit

Timer control bit settings for different operating modes are given in the Table 13-1.

TABLE 13-1: TIMER MODE SETTINGS

| Mode | TCS | TGATE |
|---------------------|-----|-------|
| Timer | 0 | 0 |
| Gated timer | 0 | 1 |
| Synchronous counter | 1 | x |

13.1 16-bit Operation

To configure any of the timers for individual 16-bit operation:

1. Clear the T32 bit corresponding to that timer.
2. Select the timer prescaler ratio using the TCKPS<1:0> bits.
3. Set the Clock and Gating modes using the TCS and TGATE bits.
4. Load the timer period value into the PRx register.
5. If interrupts are required, set the interrupt enable bit, TxIE. Use the priority bits, TxIP<2:0>, to set the interrupt priority.
6. Set the TON bit.

Note: Only Timer2 and Timer3 can trigger a DMA data transfer.

13.2 32-bit Operation

A 32-bit timer module can be formed by combining a Type B and a Type C 16-bit timer module. For 32-bit timer operation, the T32 control bit in the Type B Timer Control (TxCON<3>) register must be set. The Type C timer holds the most significant word (msw) and the Type B timer holds the least significant word (lsw) for 32-bit operation.

When configured for 32-bit operation, only the Type B Timer Control (TxCON) register bits are required for setup and control. Type C timer control register bits are ignored (except TSIDL bit).

For interrupt control, the combined 32-bit timer uses the interrupt enable, interrupt flag and interrupt priority control bits of the Type C timer. The interrupt control and status bits for the Type B timer are ignored during 32-bit timer operation.

The Type B and Type C timers that can be combined to form a 32-bit timer are listed in Table 13-2.

TABLE 13-2: 32-BIT TIMER

| TYPE B Timer (lsw) | TYPE C Timer (msw) |
|--------------------|--------------------|
| Timer2 | Timer3 |
| Timer4 | Timer5 |

A block diagram representation of the 32-bit timer module is shown in Figure 13-3. The 32-timer module can operate in one of the following modes:

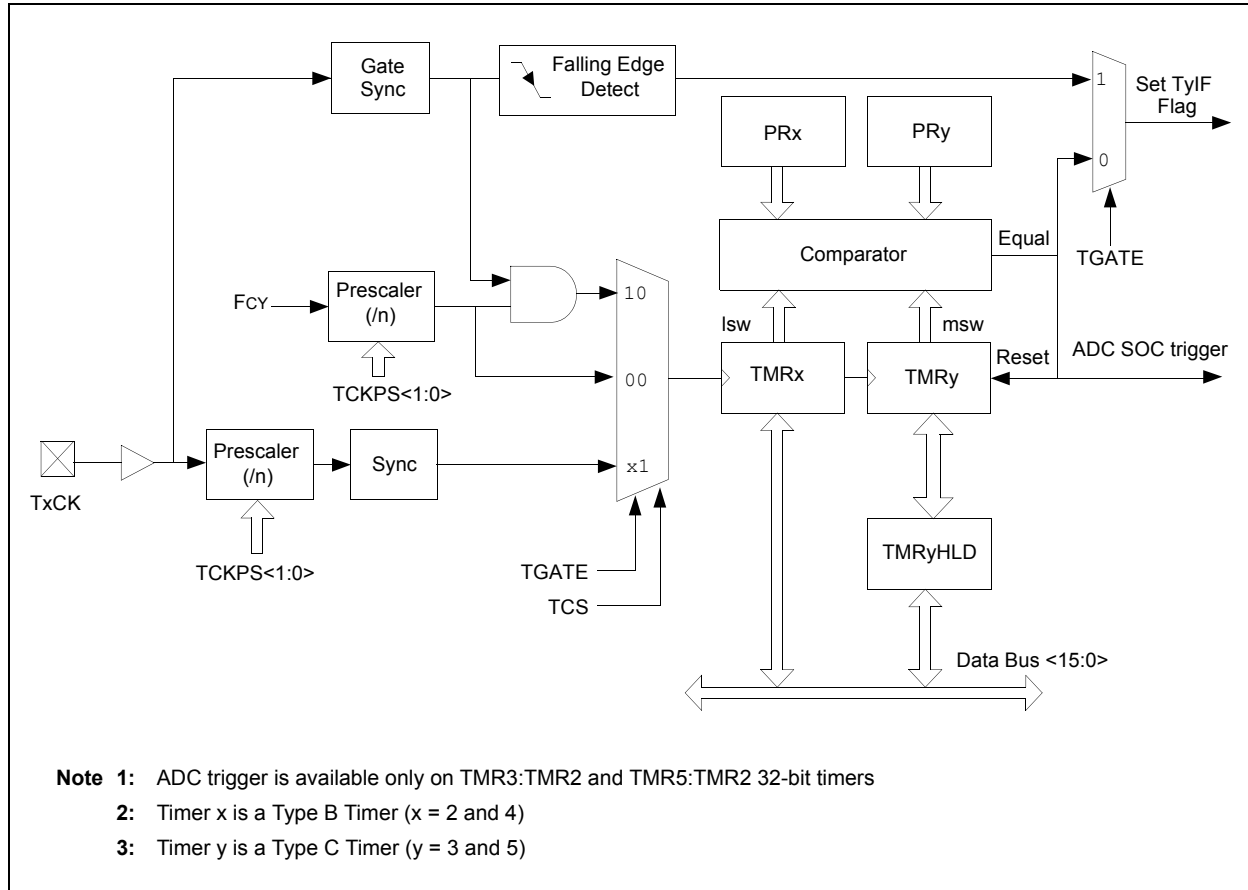
- Timer mode
- Gated Timer mode
- Synchronous Counter mode

To configure the features of Timer2/3 or Timer4/5 for 32-bit operation:

1. Set the T32 control bit.
2. Select the prescaler ratio for Timer2 or Timer4 using the TCKPS<1:0> bits.
3. Set the Clock and Gating modes using the corresponding TCS and TGATE bits.
4. Load the timer period value. PR3 or PR5 contains 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 bits, T3IE or T5IE. Use the priority bits, T3IP<2:0> or T5IP<2:0> to set the interrupt priority. While Timer2 or Timer4 controls the timer, the interrupt appears as a Timer3 or Timer5 interrupt.
6. Set the corresponding TON bit.

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

FIGURE 13-3: 32-BIT TIMER BLOCK DIAGRAM



REGISTER 13-1: TxCON: TIMER CONTROL REGISTER (x = 2 or 4)

| | | | | | | | |
|--------|-------|------------|-------|-------|-----|-------|-------|
| 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 | TCKPS<1:0> | | T32 | — | 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:** Timerx On bit
When T32 = 1 (in 32-bit Timer mode):
 1 = Starts 32-bit TMRx:TMRy timer pair
 0 = Stops 32-bit TMRx:TMRy timer pair
When T32 = 0 (in 16-bit Timer mode):
 1 = Starts 16-bit timer
 0 = Stops 16-bit timer
- bit 14 **Unimplemented:** Read as '0'
- bit 13 **TSIDL:** Stop in Idle Mode bit
 1 = Discontinue timer operation when device enters Idle mode
 0 = Continue timer 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 **TCKPS<1:0>:** Timerx Input Clock Prescale Select bits
 11 = 1:256 prescale value
 10 = 1:64 prescale value
 01 = 1:8 prescale value
 00 = 1:1 prescale value
- bit 3 **T32:** 32-bit Timerx Mode Select bit
 1 = TMRx and TMRy form a 32-bit timer
 0 = TMRx and TMRy form separate 16-bit timer
- bit 2 **Unimplemented:** Read as '0'
- bit 1 **TCS:** Timerx Clock Source Select bit
 1 = External clock from TxCK pin
 0 = Internal clock (FOSC/2)
- bit 0 **Unimplemented:** Read as '0'

REGISTER 13-2: TyCON: TIMER CONTROL REGISTER (y = 3 or 5)

| | | | | | | | |
|--------------------|-----|----------------------|-----|-----|-----|-------|-----|
| R/W-0 | U-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| TON ⁽²⁾ | — | TSIDL ⁽¹⁾ | — | — | — | — | — |
| bit 15 | | | | | | bit 8 | |

| | | | | | | | |
|-------|----------------------|---------------------------|-------|-----|-----|--------------------|-----|
| U-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | U-0 |
| — | TGATE ⁽²⁾ | TCKPS<1:0> ⁽²⁾ | | — | — | TCS ⁽²⁾ | — |
| bit 7 | | | | | | bit 0 | |

Legend:

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

- bit 15 **TON:** Timery On bit⁽²⁾
 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⁾
 1 = Discontinue timer operation when device enters Idle mode
 0 = Continue timer operation in Idle mode
- bit 12-7 **Unimplemented:** Read as '0'
- bit 6 **TGATE:** Timerx Gated Time Accumulation Enable bit⁽²⁾
 When TCS = 1:
 This bit is ignored.
 When TCS = 0:
 1 = Gated time accumulation enabled
 0 = Gated time accumulation disabled
- bit 5-4 **TCKPS<1:0>:** Timerx Input Clock Prescale Select bits⁽²⁾
 11 = 1:256 prescale value
 10 = 1:64 prescale value
 01 = 1:8 prescale value
 00 = 1:1 prescale value
- bit 3-2 **Unimplemented:** Read as '0'
- bit 1 **TCS:** Timerx Clock Source Select bit⁽²⁾
 1 = External clock from TxCK pin
 0 = Internal clock (Fosc/2)
- bit 0 **Unimplemented:** Read as '0'

- Note 1:** When 32-bit timer operation is enabled (T32 = 1) in the Timer Control (TxCON<3>) register, the TSIDL bit must be cleared to operate the 32-bit timer in Idle mode.
- 2:** When the 32-bit timer operation is enabled (T32 = 1) in the Timer Control (TxCON<3>) register, these bits have no effect.

NOTES:

14.0 INPUT CAPTURE

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 12. “Input Capture” (DS70198), which is available from the Microchip website (www.microchip.com).

The Input Capture module is useful in applications that requires frequency (period) and pulse measurement. The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices support up to four input capture channels.

The input capture module captures the 16-bit value of the selected Time Base register when an event occurs at the ICx pin. The events that cause a capture event are listed below in three categories:

1. Simple Capture Event modes:
 - Capture timer value on every falling edge of input at ICx pin
 - Capture timer value on every rising edge of input at ICx pin

2. Capture timer value on every edge (rising and falling)
3. Prescaler Capture Event modes:
 - Capture timer value on every 4th rising edge of input at ICx pin
 - Capture timer value on every 16th rising edge of input at ICx pin

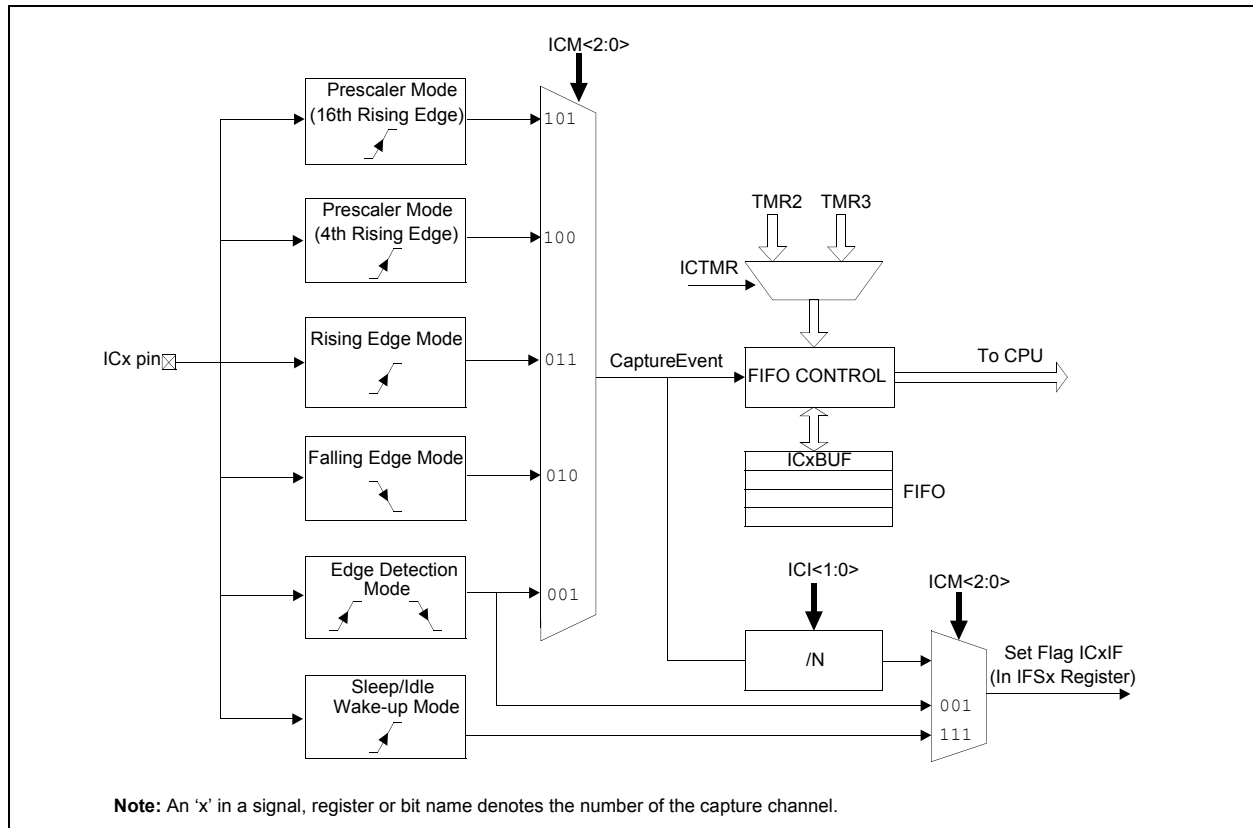
Each input capture channel can select one of two 16-bit timers (Timer2 or Timer3) for the time base. The selected timer can use either an internal or external clock.

Other operational features include:

- Device wake-up from capture pin during CPU Sleep and Idle modes
- Interrupt on input capture event
- 4-word FIFO buffer for capture values
 - Interrupt optionally generated after 1, 2, 3 or 4 buffer locations are filled
- Use of input capture to provide additional sources of external interrupts

Note: Only IC1 and IC2 can trigger a DMA data transfer. If DMA data transfers are required, the FIFO buffer size must be set to '1' (IC1<1:0> = 00)

FIGURE 14-1: INPUT CAPTURE BLOCK DIAGRAM



14.1 Input Capture Registers

REGISTER 14-1: ICxCON: INPUT CAPTURE x CONTROL REGISTER (x = 1, 2, 7 or 8)

| | | | | | | | |
|--------|-----|--------|-----|-----|-----|-----|-------|
| 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 | ICI<1:0> | | ICOV | ICBNE | ICM<2:0> | | |
| 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 15-14 **Unimplemented:** Read as '0'
- bit 13 **ICSIDL:** Input Capture Module Stop in Idle Control bit
 1 = Input capture module halts in CPU Idle mode
 0 = Input capture module continues to operate in CPU Idle mode
- bit 12-8 **Unimplemented:** Read as '0'
- bit 7 **ICTMR:** Input Capture Timer Select bits
 1 = TMR2 contents are captured on capture event
 0 = TMR3 contents are captured on capture event
- bit 6-5 **ICI<1:0>:** Select Number of Captures per Interrupt bits
 11 = Interrupt on every fourth capture event
 10 = Interrupt on every third capture event
 01 = Interrupt on every second capture event
 00 = Interrupt on every capture event
- bit 4 **ICOV:** Input Capture Overflow Status Flag bit (read-only)
 1 = Input capture overflow occurred
 0 = No input capture overflow occurred
- bit 3 **ICBNE:** Input Capture Buffer Empty Status bit (read-only)
 1 = Input capture buffer is not empty, at least one more capture value can be read
 0 = Input capture buffer is empty
- bit 2-0 **ICM<2:0>:** Input Capture Mode Select bits
 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

15.0 OUTPUT COMPARE

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 13. “Output Compare” (DS70209), which is available from the Microchip website (www.microchip.com).

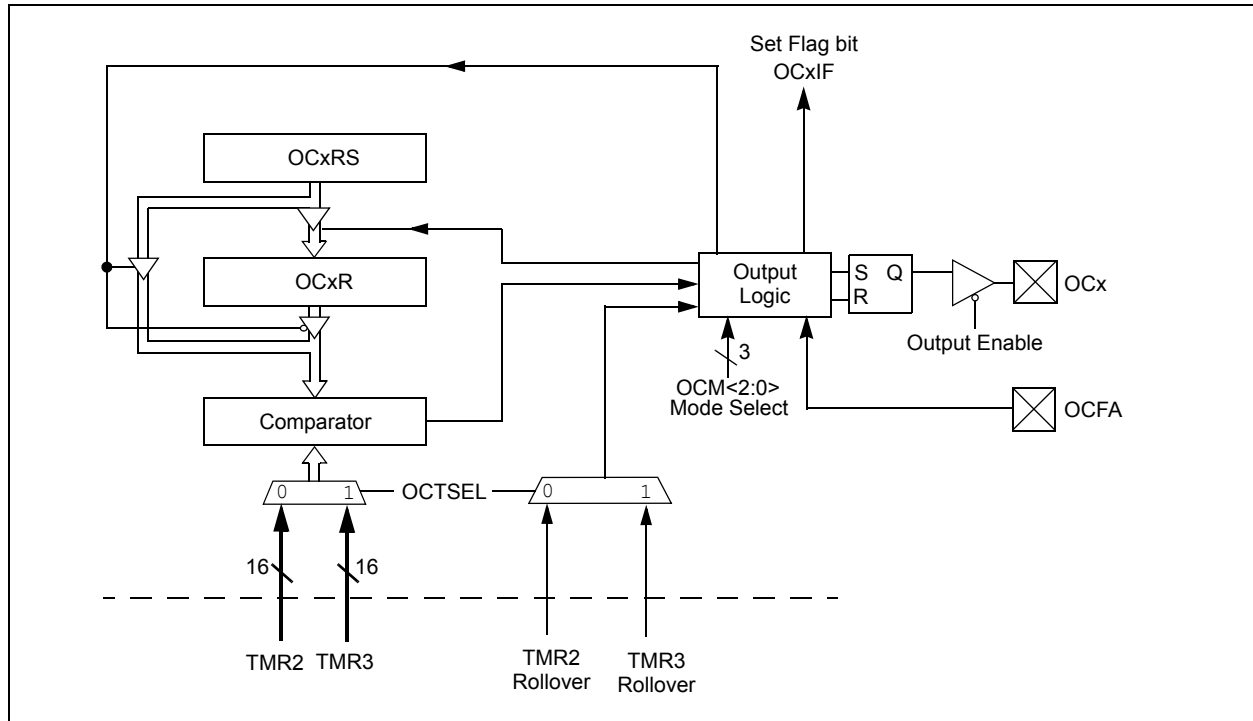
The Output Compare module can select either Timer2 or Timer3 for its time base. The module compares the value of the timer with the value of one or two compare registers depending on the operating mode selected.

The state of the output pin changes when the timer value matches the compare register value. The Output Compare module generates either a single output pulse or a sequence of output pulses, by changing the state of the output pin on the compare match events. The Output Compare module can also generate interrupts on compare match events.

The Output Compare module has multiple operating modes:

- Active Low One-Shot mode
- Active High One-Shot mode
- Toggle mode
- Delayed One-Shot mode
- Continuous Pulse mode
- PWM mode without fault protection
- PWM mode with fault protection

FIGURE 15-1: OUTPUT COMPARE MODULE BLOCK DIAGRAM



15.1 Output Compare Modes

Configure the Output Compare modes by setting the appropriate Output Compare Mode (OCM<2:0>) bits in the Output Compare Control (OCxCON<2:0>) register. Table 15-1 lists the different bit settings for the Output Compare modes. Figure 15-2 illustrates the output compare operation for various modes. The user application must disable the associated timer when writing to the output compare control registers to avoid malfunctions.

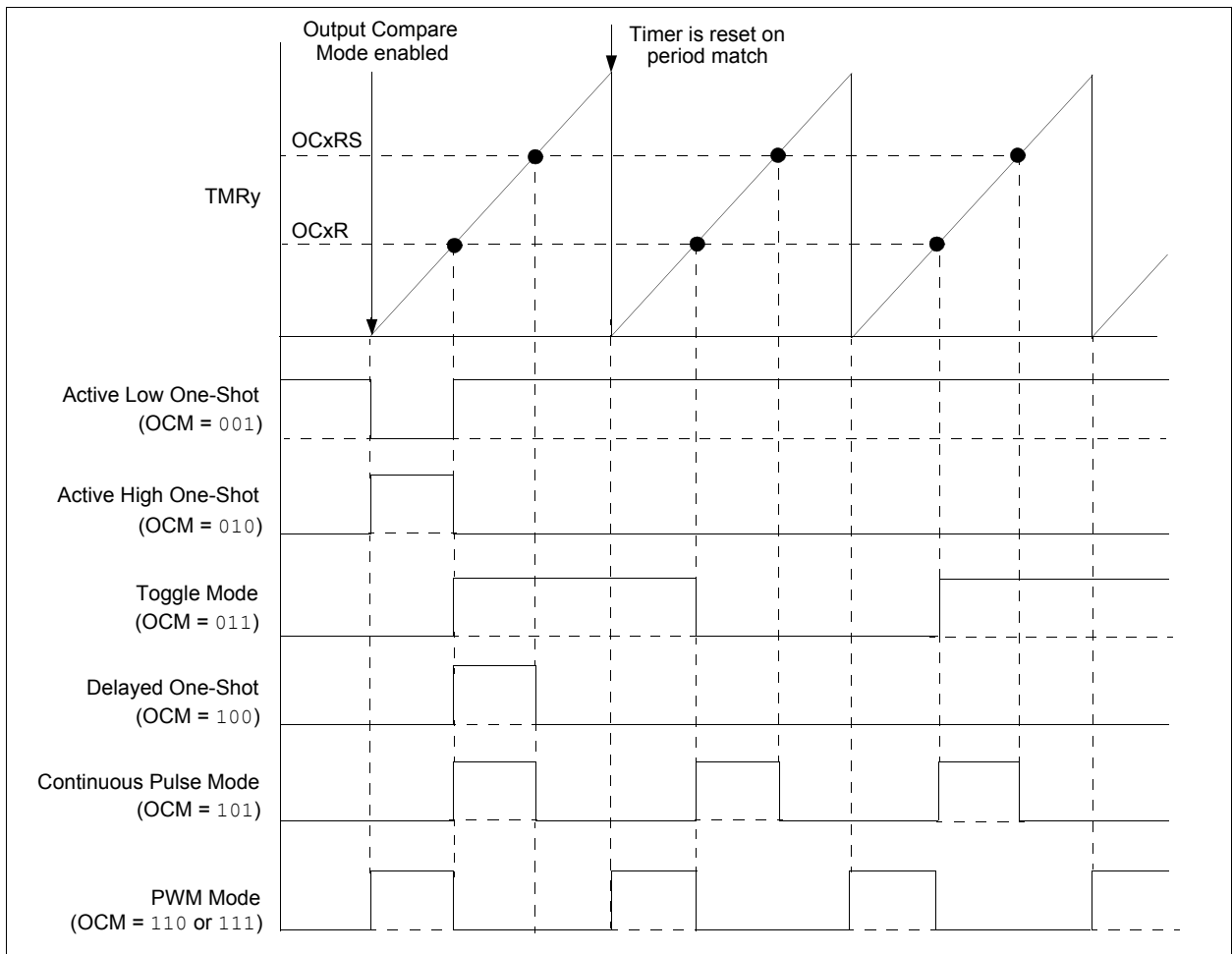
Note 1: Only OC1 and OC2 can trigger a DMA data transfer.

2: See **Section 13. “Output Compare”** in the “*dsPIC33F Family Reference Manual*” (DS70209) for OCxR and OCxRS register restrictions.

TABLE 15-1: OUTPUT COMPARE MODES

| OCM<2:0> | Mode | OCx Pin Initial State | OCx Interrupt Generation |
|----------|-----------------------------------|--|----------------------------------|
| 000 | Module Disabled | Controlled by GPIO register | — |
| 001 | Active Low One-Shot | 0 | OCx Rising edge |
| 010 | Active High One-Shot | 1 | OCx Falling edge |
| 011 | Toggle Mode | Current output is maintained | OCx Rising and Falling edge |
| 100 | Delayed One-Shot | 0 | OCx Falling edge |
| 101 | Continuous Pulse mode | 0 | OCx Falling edge |
| 110 | PWM mode without fault protection | 0, if OCxR is zero 1, if OCxR is non-zero | No interrupt |
| 111 | PWM mode with fault protection | 0, if OCxR is zero 1, if OCxR is non-zero | OCFA Falling edge for OC1 to OC4 |

FIGURE 15-2: OUTPUT COMPARE OPERATION



REGISTER 15-1: OCxCON: OUTPUT COMPARE x CONTROL REGISTER (x = 1, 2, 3 or 4)

| | | | | | | | |
|--------|-----|--------|-----|-----|-----|-------|-----|
| 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 | OCM<2:0> | | |
| bit 7 | | | | | | bit 0 | |

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

- bit 15-14 **Unimplemented:** Read as '0'
- bit 13 **OCSIDL:** Stop Output Compare in Idle Mode Control bit
 - 1 = Output Compare x halts in CPU Idle mode
 - 0 = Output Compare x continues to operate in CPU Idle mode
- bit 12-5 **Unimplemented:** Read as '0'
- bit 4 **OCFLT:** PWM Fault Condition Status bit
 - 1 = PWM Fault condition has occurred (cleared in hardware only)
 - 0 = No PWM Fault condition has occurred
 - (This bit is only used when OCM<2:0> = 111.)
- bit 3 **OCTSEL:** Output Compare Timer Select bit
 - 1 = Timer3 is the clock source for Compare x
 - 0 = Timer2 is the clock source for Compare x
- bit 2-0 **OCM<2:0>:** Output Compare Mode Select bits
 - 111 = PWM mode on OCx, Fault pin enabled
 - 110 = PWM mode on OCx, Fault pin disabled
 - 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

NOTES:

16.0 MOTOR CONTROL PWM MODULE

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “*dsPIC33F Family Reference Manual*”, **Section 14. “Motor Control PWM”** (DS70187), which is available from the Microchip website (www.microchip.com).

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 device supports up to two dedicated Pulse Width Modulation (PWM) modules. The PWM1 module is a 6-channel PWM generator, and the PWM2 module is a 2-channel PWM generator.

The PWM module has the following features:

- Up to 16-bit resolution
- On-the-fly PWM frequency changes
- Edge and Center-Aligned Output modes
- Single Pulse Generation mode
- Interrupt support for asymmetrical updates in Center-Aligned mode
- Output override control for Electrically Commutative Motor (ECM) operation or Brushless DC (BLDC)
- Special Event Comparator for scheduling other peripheral events
- Fault pins to optionally drive each of the PWM output pins to a defined state
- Duty cycle updates configurable to be immediate or synchronized to the PWM time base

16.1 PWM1: 6-Channel PWM Module

This module simplifies the task of generating multiple synchronized PWM outputs. The following power and motion control applications are supported by the PWM module:

- 3-Phase AC Induction Motor
- Switched Reluctance (SR) Motor
- Brushless DC (BLDC) Motor
- Uninterruptible Power Supply (UPS)

This module contains three duty cycle generators, numbered 1 through 3. The module has six PWM output pins, numbered PWM1H1/PWM1L1 through PWM1H3/PWM1L3. The six I/O pins are grouped into high/low numbered pairs, denoted by the suffix H or L, respectively. For complementary loads, the low PWM pins are always the complement of the corresponding high I/O pin.

16.2 PWM2: 2-Channel PWM Module

This module provides an additional pair of complimentary PWM outputs that can be used for:

- Independent PFC correction in a motor system
- Induction cooking

This module contains a duty cycle generator that provides two PWM outputs, numbered PWM2H1/PWM2L1.

FIGURE 16-1: 6-CHANNEL PWM MODULE BLOCK DIAGRAM (PWM1)

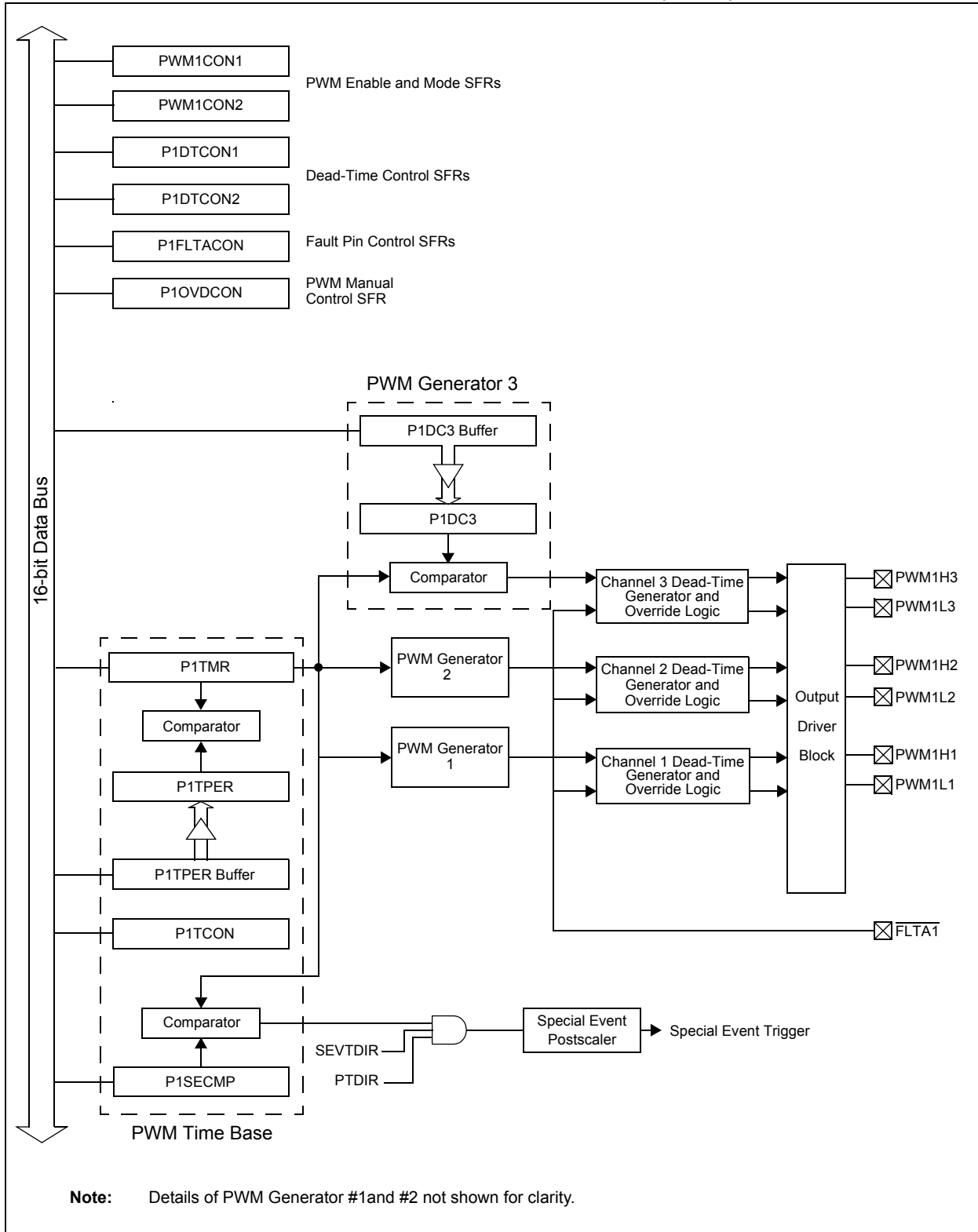
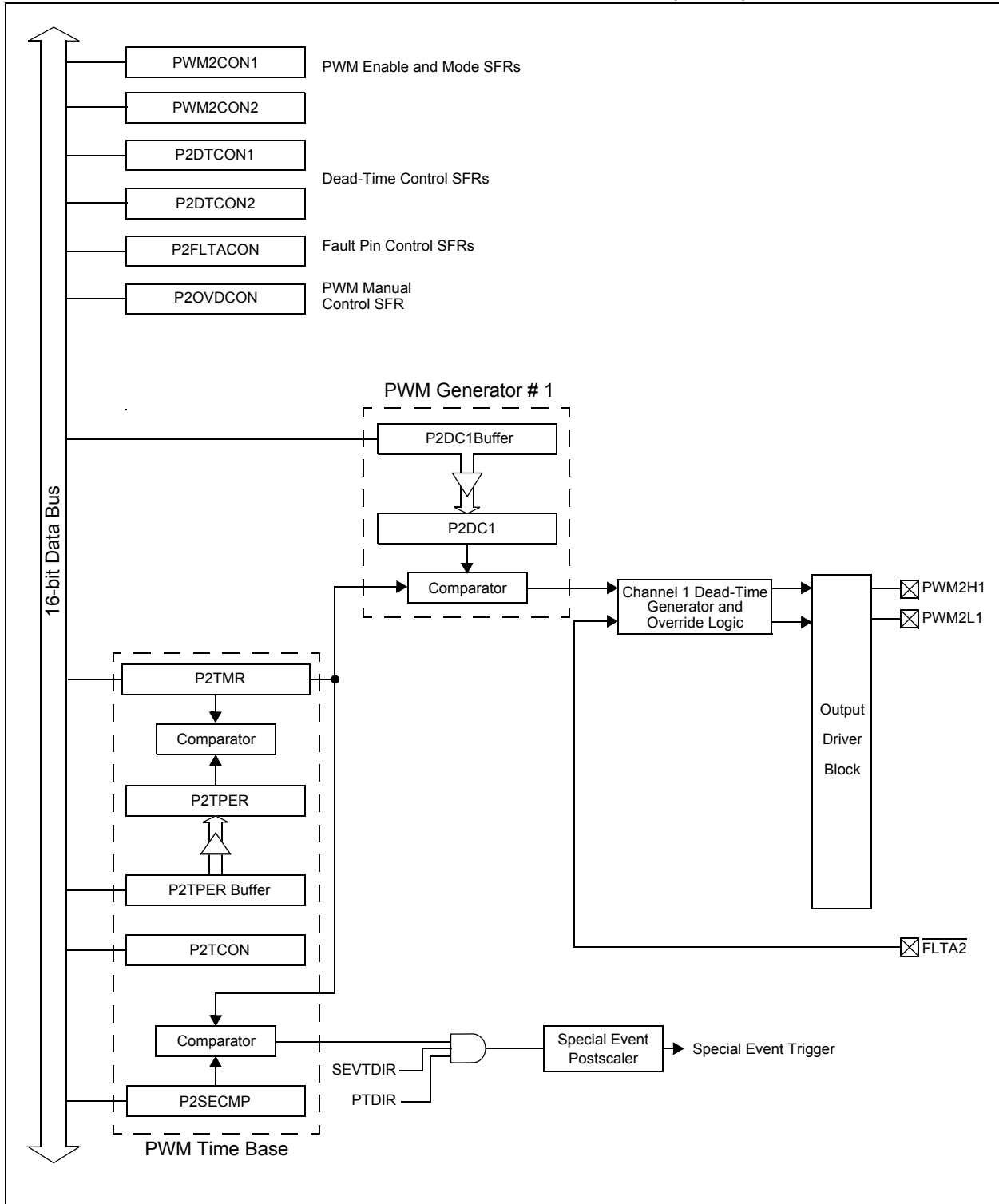


FIGURE 16-2: 2-CHANNEL PWM MODULE BLOCK DIAGRAM (PWM2)



REGISTER 16-1: P_xTCON: PWM TIME BASE CONTROL REGISTER

| | | | | | | | |
|--------|-----|--------|-----|-----|-----|-------|-----|
| R/W-0 | U-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| PTEN | — | PTSIDL | — | — | — | — | — |
| 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 |
| PTOPS<3:0> | | | PTCKPS<1:0> | | PTMOD<1:0> | | |
| 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 **PTEN:** PWM Time Base Timer Enable bit
 1 = PWM time base is on
 0 = PWM time base is off
- bit 14 **Unimplemented:** Read as '0'
- bit 13 **PTSIDL:** PWM Time Base Stop in Idle Mode bit
 1 = PWM time base halts in CPU Idle mode
 0 = PWM time base runs in CPU Idle mode
- bit 12-8 **Unimplemented:** Read as '0'
- bit 7-4 **PTOPS<3:0>:** PWM Time Base Output Postscale Select bits
 1111 = 1:16 postscale
 •
 •
 •
 0001 = 1:2 postscale
 0000 = 1:1 postscale
- bit 3-2 **PTCKPS<1:0>:** PWM Time Base Input Clock Prescale Select bits
 11 = PWM time base input clock period is 64 T_{CY} (1:64 prescale)
 10 = PWM time base input clock period is 16 T_{CY} (1:16 prescale)
 01 = PWM time base input clock period is 4 T_{CY} (1:4 prescale)
 00 = PWM time base input clock period is T_{CY} (1:1 prescale)
- bit 1-0 **PTMOD<1:0>:** PWM Time Base Mode Select bits
 11 = PWM time base operates in a Continuous Up/Down Count mode with interrupts for double PWM updates
 10 = PWM time base operates in a Continuous Up/Down Count mode
 01 = PWM time base operates in Single Pulse mode
 00 = PWM time base operates in a Free-Running mode

REGISTER 16-2: P_xTMR: PWM TIMER COUNT VALUE REGISTER

| | | | | | | | | |
|-----------|------------|-------|-------|-------|-------|-------|-------|-------|
| R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | |
| PTDIR | PTMR<14:8> | | | | | | | |
| 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 | |
| PTMR<7:0> | | | | | | | | |
| 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 **PTDIR:** PWM Time Base Count Direction Status bit (read-only)
 1 = PWM time base is counting down
 0 = PWM time base is counting up

bit 14-0 **PTMR <14:0>:** PWM Time Base Register Count Value bits

REGISTER 16-3: P_xTPER: PWM TIME BASE PERIOD REGISTER

| | | | | | | | | |
|------------|-------------|-------|-------|-------|-------|-------|-------|-------|
| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | |
| — | PTPER<14:8> | | | | | | | |
| 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 | |
| PTPER<7:0> | | | | | | | | |
| 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 **Unimplemented:** Read as '0'

bit 14-0 **PTPER<14:0>:** PWM Time Base Period Value bits

REGISTER 16-4: PxSECMP: SPECIAL EVENT COMPARE 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 |
| SEVTDIR ⁽¹⁾ | SEVTCMP<14:8> ⁽²⁾ | | | | | | |
| 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 |
| SEVTCMP<7:0> ⁽²⁾ | | | | | | | |
| bit 7 | bit 0 | | | | | | |

Legend:

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

bit 15 **SEVTDIR:** Special Event Trigger Time Base Direction bit⁽¹⁾
 1 = A Special Event Trigger occurs when the PWM time base is counting downward
 0 = A Special Event Trigger occurs when the PWM time base is counting upward

bit 14-0 **SEVTCMP<14:0>:** Special Event Compare Value bits⁽²⁾

- Note 1:** SEVTDIR is compared with PTDIR (PxTMR<15>) to generate the Special Event Trigger.
2: PxSECMP<14:0> is compared with PxTMR<14:0> to generate the Special Event Trigger.

REGISTER 16-5: PWMxCON1: PWM CONTROL REGISTER 1 ⁽²⁾

| | | | | | | | |
|--------|-----|-----|-----|-----|-------|-------|-------|
| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | — | — | PMOD3 | PMOD2 | PMOD1 |
| bit 15 | | | | | bit 8 | | |

| | | | | | | | |
|-------|----------------------|----------------------|----------------------|-----|----------------------|----------------------|----------------------|
| U-0 | R/W-1 | R/W-1 | R/W-1 | U-0 | R/W-1 | R/W-1 | R/W-1 |
| — | PEN3H ⁽¹⁾ | PEN2H ⁽¹⁾ | PEN1H ⁽¹⁾ | — | PEN3L ⁽¹⁾ | PEN2L ⁽¹⁾ | PEN1L ⁽¹⁾ |
| bit 7 | | | | | bit 0 | | |

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

- bit 15-11 **Unimplemented:** Read as '0'
- bit 10-8 **PMOD4:PMOD1:** PWM I/O Pair Mode bits
 1 = PWM I/O pin pair is in the Independent PWM Output mode
 0 = PWM I/O pin pair is in the Complementary Output mode
- bit 7 **Unimplemented:** Read as '0'
- bit 6-4 **PEN3H:PEN1H:** PWMxH I/O Enable bits⁽¹⁾
 1 = PWMxH pin is enabled for PWM output
 0 = PWMxH pin disabled, I/O pin becomes general purpose I/O
- bit 3 **Unimplemented:** Read as '0'
- bit 2-0 **PEN3L:PEN1L:** PWMxL I/O Enable bits⁽¹⁾
 1 = PWMxL pin is enabled for PWM output
 0 = PWMxL pin disabled, I/O pin becomes general purpose I/O

Note 1: Reset condition of the PENxH and PENxL bits depends on the value of the PWMPIN Configuration bit in the FPOR Configuration register.

2: PWM2 supports only one PWM I/O pin pair.

REGISTER 16-6: PWMxCON2: PWM CONTROL REGISTER 2

| | | | | | | | |
|--------|-----|-----|-----|-------------|-------|-------|-------|
| U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | — | SEVOPS<3:0> | | | |
| bit 15 | | | | bit 8 | | | |

| | | | | | | | |
|-------|-----|-----|-----|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | — | — | IUE | OSYNC | UDIS |
| bit 7 | | | | bit 0 | | | |

Legend:

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

bit 15-12 **Unimplemented:** Read as '0'

bit 11-8 **SEVOPS<3:0>:** PWM Special Event Trigger Output Postscale Select bits

1111 = 1:16 postscale

•

•

•

0001 = 1:2 postscale

0000 = 1:1 postscale

bit 7-3 **Unimplemented:** Read as '0'

bit 2 **IUE:** Immediate Update Enable bit

1 = Updates to the active PxDC registers are immediate

0 = Updates to the active PxDC registers are synchronized to the PWM time base

bit 1 **OSYNC:** Output Override Synchronization bit

1 = Output overrides via the PxOVDCON register are synchronized to the PWM time base

0 = Output overrides via the PxOVDCON register occur on next Tcy boundary

bit 0 **UDIS:** PWM Update Disable bit

1 = Updates from Duty Cycle and Period Buffer registers are disabled

0 = Updates from Duty Cycle and Period Buffer registers are enabled

REGISTER 16-7: PxDTCON1: DEAD-TIME CONTROL REGISTER 1

| | | | | | | | |
|------------|-------|----------|-------|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| DTBPS<1:0> | | DTB<5:0> | | | | | |
| 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 |
| DTAPS<1:0> | | DTA<5:0> | | | | | |
| 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 15-14 **DTBPS<1:0>**: Dead-Time Unit B Prescale Select bits
 - 11 = Clock period for Dead-Time Unit B is 8 TCY
 - 10 = Clock period for Dead-Time Unit B is 4 TCY
 - 01 = Clock period for Dead-Time Unit B is 2 TCY
 - 00 = Clock period for Dead-Time Unit B is TCY
- bit 13-8 **DTB<5:0>**: Unsigned 6-bit Dead-Time Value for Dead-Time Unit B bits
- bit 7-6 **DTAPS<1:0>**: Dead-Time Unit A Prescale Select bits
 - 11 = Clock period for Dead-Time Unit A is 8 TCY
 - 10 = Clock period for Dead-Time Unit A is 4 TCY
 - 01 = Clock period for Dead-Time Unit A is 2 TCY
 - 00 = Clock period for Dead-Time Unit A is TCY
- bit 5-0 **DTA<5:0>**: Unsigned 6-bit Dead-Time Value for Dead-Time Unit A bits

REGISTER 16-8: PxDTCON2: DEAD-TIME CONTROL REGISTER 2 ⁽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 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | DTS3A | DTS3I | DTS2A | DTS2I | DTS1A | DTS1I |
| bit 7 | | | | | | bit 0 | |

Legend:

R = Readable bit W = Writable 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 **DTS3A:** Dead-Time Select for PWMxH3 Signal Going Active bit
 1 = Dead time provided from Unit B
 0 = Dead time provided from Unit A
- bit 4 **DTS3I:** Dead-Time Select for PWMxL3 Signal Going Inactive bit
 1 = Dead time provided from Unit B
 0 = Dead time provided from Unit A
- bit 3 **DTS2A:** Dead-Time Select for PWMxH2 Signal Going Active bit
 1 = Dead time provided from Unit B
 0 = Dead time provided from Unit A
- bit 2 **DTS2I:** Dead-Time Select for PWMxL2 Signal Going Inactive bit
 1 = Dead time provided from Unit B
 0 = Dead time provided from Unit A
- bit 1 **DTS1A:** Dead-Time Select for PWMxH1 Signal Going Active bit
 1 = Dead time provided from Unit B
 0 = Dead time provided from Unit A
- bit 0 **DTS1I:** Dead-Time Select for PWMxL1 Signal Going Inactive bit
 1 = Dead time provided from Unit B
 0 = Dead time provided from Unit A

Note 1: PWM2 supports only one PWM I/O pin pair.

REGISTER 16-9: PxFLTAcon: FAULT A CONTROL REGISTER⁽¹⁾

| | | | | | | | |
|--------|-----|--------|--------|--------|--------|--------|--------|
| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | FAOV3H | FAOV3L | FAOV2H | FAOV2L | FAOV1H | FAOV1L |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------|-----|-----|-----|-----|-------|-------|-------|
| R/W-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
| FLTAM | — | — | — | — | FAEN3 | FAEN2 | FAEN1 |
| bit 7 | | | | | | | bit 0 |

| | | | |
|-------------------|------------------|------------------------------------|--------------------|
| Legend: | | | |
| R = Readable bit | W = Writable 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 13-8 **FAOVxH<3:1>:FAOVxL<3:1>:** Fault Input A PWM Override Value bits
 - 1 = The PWM output pin is driven active on an external Fault input event
 - 0 = The PWM output pin is driven inactive on an external Fault input event
- bit 7 **FLTAM:** Fault A Mode bit
 - 1 = The Fault A input pin functions in the Cycle-by-Cycle mode
 - 0 = The Fault A input pin latches all control pins to the programmed states in PxFLTAcon<13:8>
- bit 6-3 **Unimplemented:** Read as '0'
- bit 2 **FAEN3:** Fault Input A Enable bit
 - 1 = PWMxH3/PWMxL3 pin pair is controlled by Fault Input A
 - 0 = PWMxH3/PWMxL3 pin pair is not controlled by Fault Input A
- bit 1 **FAEN2:** Fault Input A Enable bit
 - 1 = PWMxH2/PWMxL2 pin pair is controlled by Fault Input A
 - 0 = PWMxH2/PWMxL2 pin pair is not controlled by Fault Input A
- bit 0 **FAEN1:** Fault Input A Enable bit
 - 1 = PWMxH1/PWMxL1 pin pair is controlled by Fault Input A
 - 0 = PWMxH1/PWMxL1 pin pair is not controlled by Fault Input A

Note 1: PWM2 supports only one PWM I/O pin pair.

REGISTER 16-10: P_xOVDCON: OVERRIDE CONTROL REGISTER⁽¹⁾

| | | | | | | | |
|--------|-----|--------|--------|--------|--------|--------|--------|
| U-0 | U-0 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
| — | — | POVD3H | POVD3L | POVD2H | POVD2L | POVD1H | POVD1L |
| 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 |
| — | — | POUT3H | POUT3L | POUT2H | POUT2L | POUT1H | POUT1L |
| bit 7 | | | | | | bit 0 | |

Legend:

| | | |
|-------------------|------------------|--|
| R = Readable bit | W = Writable 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 13-8 **POVDxH<3:1>:POVDxL<3:1>:** PWM Output Override bits
 - 1 = Output on PWMx I/O pin is controlled by the PWM generator
 - 0 = Output on PWMx I/O pin is controlled by the value in the corresponding POUTxH:POUTxL bit
- bit 7-6 **Unimplemented:** Read as '0'
- bit 5-0 **POUTxH<3:1>:POUTxL<3:1>:** PWM Manual Output bits
 - 1 = PWMx I/O pin is driven active when the corresponding POVDxH:POVDxL bit is cleared
 - 0 = PWMx I/O pin is driven inactive when the corresponding POVDxH:POVDxL bit is cleared

Note 1: PWM2 supports only one PWM I/O pin pair.

REGISTER 16-11: PxDC1: PWM DUTY CYCLE REGISTER 1

| | | | | | | | |
|------------|-------|-------|-------|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| PDC1<15:8> | | | | | | | |
| 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 |
| PDC1<7:0> | | | | | | | |
| 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 15-0 **PDC1<15:0>**: PWM Duty Cycle 1 Value bits

REGISTER 16-12: P1DC2: PWM DUTY CYCLE REGISTER 2

| | | | | | | | |
|------------|-------|-------|-------|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| PDC2<15:8> | | | | | | | |
| 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 |
| PDC2<7:0> | | | | | | | |
| 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 15-0 **PDC2<15:0>**: PWM Duty Cycle 2 Value bits

REGISTER 16-13: P1DC3: PWM DUTY CYCLE REGISTER 3

| | | | | | | | |
|------------|-------|-------|-------|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| PDC3<15:8> | | | | | | | |
| 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 |
| PDC3<7:0> | | | | | | | |
| 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 15-0 **PDC3<15:0>**: PWM Duty Cycle 3 Value bits

NOTES:

17.0 QUADRATURE ENCODER INTERFACE (QEI) MODULE

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 15. “Quadrature Encoder Interface (QEI)” (DS70208), which is available from the Microchip website (www.microchip.com).

This chapter describes the Quadrature Encoder Interface (QEI) module and associated operational modes. The QEI module provides the interface to incremental encoders for obtaining mechanical position data.

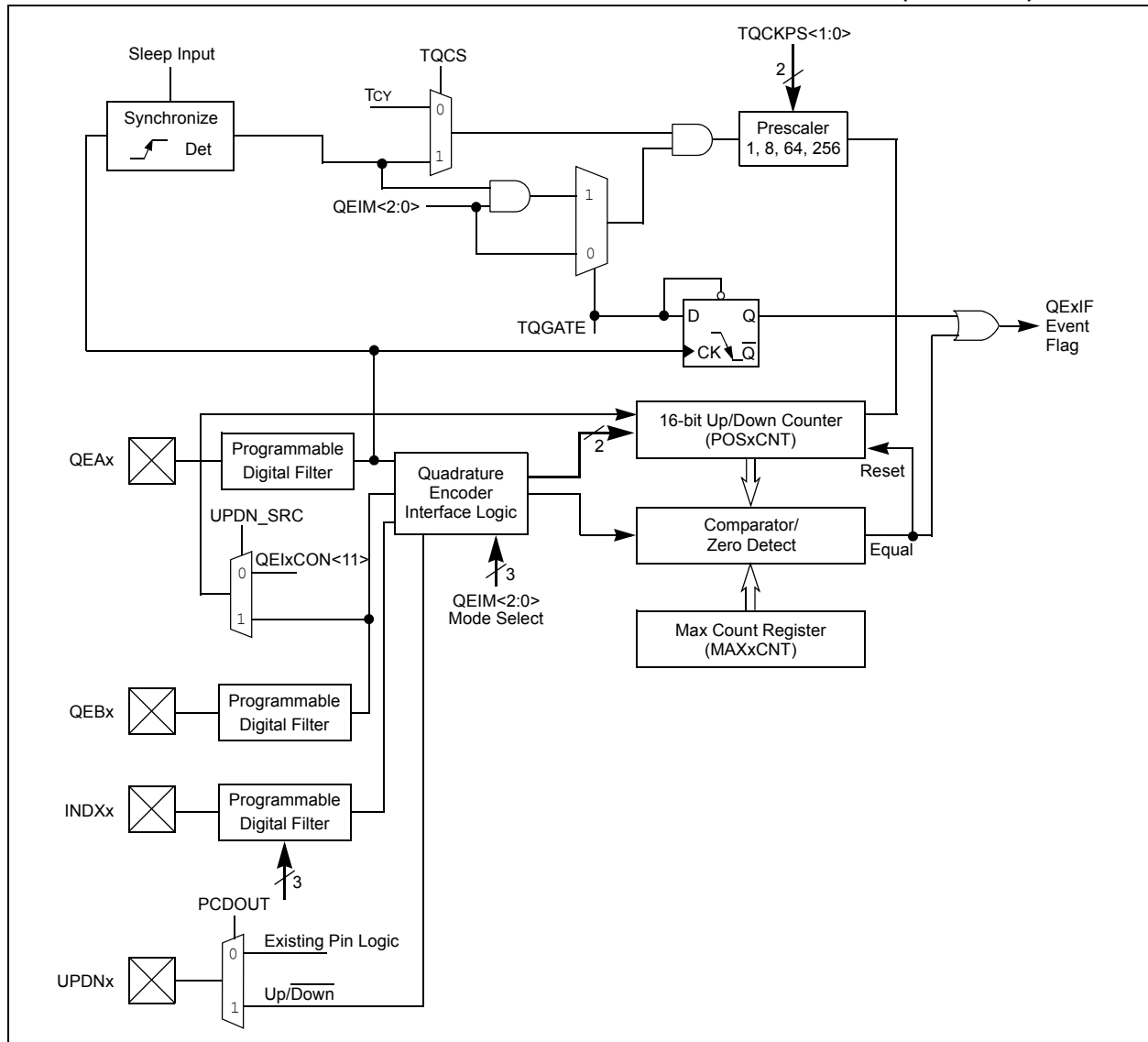
The operational features of the QEI include:

- Three input channels for two phase signals and index pulse
- 16-bit up/down position counter
- Count direction status
- Position Measurement (x2 and x4) mode
- Programmable digital noise filters on inputs
- Alternate 16-bit Timer/Counter mode
- Quadrature Encoder Interface interrupts

These operating modes are determined by setting the appropriate bits, QEIM<2:0> in (QEIxCON<10:8>). Figure 17-1 depicts the Quadrature Encoder Interface block diagram.

Note: An ‘x’ used in the names of pins, control/status bits and registers denotes a particular Quadrature Encoder Interface (QEI) module number (x = 1 or 2).

FIGURE 17-1: QUADRATURE ENCODER INTERFACE BLOCK DIAGRAM (x = 1 OR 2)



REGISTER 17-1: QEIxCON: QEIx CONTROL REGISTER (x = 1 or 2)

| | | | | | | | |
|--------|-----|---------|-------|-------|-----------|-------|-------|
| R/W-0 | U-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| CNTERR | — | QEISIDL | INDEX | UPDN | QEIM<2:0> | | |
| 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 |
| SWPAB | PCDOUT | TQGATE | TQCKPS<1:0> | | POSRES | TQCS | UPDN_SRC |
| bit 7 | | | | | | bit 0 | |

Legend:

| | | |
|-------------------|------------------|------------------------------------|
| R = Readable bit | W = Writable 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 **CNTERR:** Count Error Status Flag bit⁽¹⁾
 1 = Position count error has occurred
 0 = No position count error has occurred
- bit 14 **Unimplemented:** Read as '0'
- bit 13 **QEISIDL:** Stop in Idle Mode bit
 1 = Discontinue module operation when device enters Idle mode
 0 = Continue module operation in Idle mode
- bit 12 **INDEX:** Index Pin State Status bit (Read-Only)
 1 = Index pin is High
 0 = Index pin is Low
- bit 11 **UPDN:** Position Counter Direction Status bit⁽²⁾
 1 = Position Counter Direction is positive (+)
 0 = Position Counter Direction is negative (-)
- bit 10-8 **QEIM<2:0>:** Quadrature Encoder Interface Mode Select bits
 111 = Quadrature Encoder Interface enabled (x4 mode) with position counter reset by match (MAXxCNT)
 110 = Quadrature Encoder Interface enabled (x4 mode) with Index Pulse reset of position counter
 101 = Quadrature Encoder Interface enabled (x2 mode) with position counter reset by match (MAXxCNT)
 100 = Quadrature Encoder Interface enabled (x2 mode) with Index Pulse reset of position counter
 011 = Unused (Module disabled)
 010 = Unused (Module disabled)
 001 = Starts 16-bit Timer
 000 = Quadrature Encoder Interface/Timer off
- bit 7 **SWPAB:** Phase A and Phase B Input Swap Select bit
 1 = Phase A and Phase B inputs swapped
 0 = Phase A and Phase B inputs not swapped
- bit 6 **PCDOUT:** Position Counter Direction State Output Enable bit
 1 = Position Counter Direction Status Output Enable (QEI logic controls state of I/O pin)
 0 = Position Counter Direction Status Output Disabled (Normal I/O pin operation)

- Note 1:** CNTERR flag only applies when QEIM<2:0> = '110' or '100'.
- 2:** Read-only bit when QEIM<2:0> = '1xx'. Read/write bit when QEIM<2:0> = '001'.
- 3:** Prescaler utilized for 16-bit Timer mode only.
- 4:** This bit applies only when QEIM<2:0> = 100 or 110.
- 5:** When configured for QEI mode, this control bit is a 'don't care'.

REGISTER 17-1: QEIxCON: QEIx CONTROL REGISTER (x = 1 or 2) (CONTINUED)

| | |
|---------|--|
| bit 5 | TQGATE: Timer Gated Time Accumulation Enable bit 1 = Timer gated time accumulation enabled 0 = Timer gated time accumulation disabled |
| bit 4-3 | TQCKPS<1:0>: Timer Input Clock Prescale Select bits ⁽³⁾ 11 = 1:256 prescale value 10 = 1:64 prescale value 01 = 1:8 prescale value 00 = 1:1 prescale value |
| bit 2 | POSRES: Position Counter Reset Enable bit ⁽⁴⁾ 1 = Index Pulse resets Position Counter 0 = Index Pulse does not reset Position Counter |
| bit 1 | TQCS: Timer Clock Source Select bit 1 = External clock from pin QEAx (on the rising edge) 0 = Internal clock (Tcy) |
| bit 0 | UPDN_SRC: Position Counter Direction Selection Control bit ⁽⁵⁾ 1 = QEBx pin state defines position counter direction 0 = Control/Status bit, UPDN (QEIxCON<11>), defines timer counter (POSxCNT) direction |

Note 1: CNTERR flag only applies when QEIM<2:0> = '110' or '100'.

2: Read-only bit when QEIM<2:0> = '1xx'. Read/write bit when QEIM<2:0> = '001'.

3: Prescaler utilized for 16-bit Timer mode only.

4: This bit applies only when QEIM<2:0> = 100 or 110.

5: When configured for QEI mode, this control bit is a 'don't care'.

REGISTER 17-2: DFLT_xCON: DIGITAL FILTER CONTROL REGISTER

| | | | | | | | |
|--------|-----|-----|-----|-----|----------|-------|-------|
| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | — | — | IMV<2:0> | | CEID |
| bit 15 | | | | | | | bit 8 |

| | | | | | |
|-------|-----------|-----|-----|-----|-------|
| R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
| QEOUT | QECK<2:0> | — | — | — | — |
| 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 10-9 **IMV<1:0>:** Index Match Value bits – These bits allow the user application to specify the state of the QEAx and QEBx input pins during an Index pulse when the POSxCNT register is to be reset.
In x4 Quadrature Count Mode:
 IMV1 = Required State of Phase B input signal for match on index pulse
 IMV0 = Required State of Phase A input signal for match on index pulse
In x4 Quadrature Count Mode:
 IMV1 = Selects Phase input signal for Index state match (0 = Phase A, 1 = Phase B)
 IMV0 = Required state of the selected Phase input signal for match on index pulse
- bit 8 **CEID:** Count Error Interrupt Disable bit
 1 = Interrupts due to count errors are disabled
 0 = Interrupts due to count errors are enabled
- bit 7 **QEOUT:** QEAx/QEBx/INDXx Pin Digital Filter Output Enable bit
 1 = Digital filter outputs enabled
 0 = Digital filter outputs disabled (normal pin operation)
- bit 6-4 **QECK<2:0>:** QEAx/QEBx/INDXx Digital Filter Clock Divide Select Bits
 111 = 1:256 Clock Divide
 110 = 1:128 Clock Divide
 101 = 1:64 Clock Divide
 100 = 1:32 Clock Divide
 011 = 1:16 Clock Divide
 010 = 1:4 Clock Divide
 001 = 1:2 Clock Divide
 000 = 1:1 Clock Divide
- bit 3-0 **Unimplemented:** Read as '0'

18.0 SERIAL PERIPHERAL INTERFACE (SPI)

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 18. “Serial Peripheral Interface (SPI)” (DS70206), which is available from the Microchip website (www.microchip.com).

The Serial Peripheral Interface (SPI) module is a synchronous serial interface useful for communicating with other peripheral or microcontroller devices. These

peripheral devices can be serial EEPROMs, shift registers, display drivers, analog-to-digital converters, etc. The SPI module is compatible with SPI and SIOP from Motorola®.

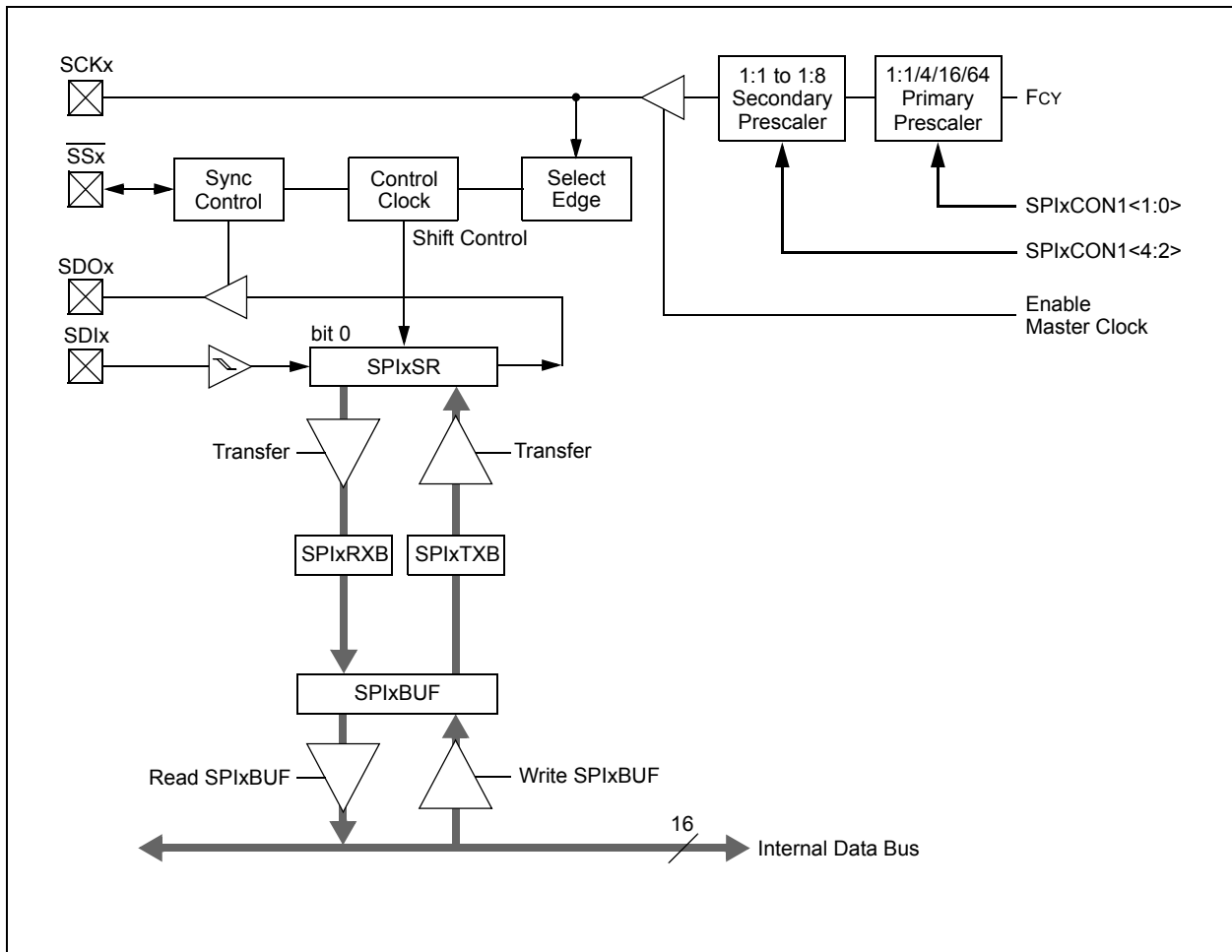
Each SPI module consists of a 16-bit shift register, SPIxSR (where x = 1 or 2), used for shifting data in and out, and a buffer register, SPIxBUF. A control register, SPIxCON, configures the module. Additionally, a status register, SPIxSTAT, indicates status conditions.

The serial interface consists of 4 pins:

- SDIx (serial data input)
- SDOx (serial data output)
- SCKx (shift clock input or output)
- SSx (active low slave select).

In Master mode operation, SCK is a clock output. In Slave mode, it is a clock input.

FIGURE 18-1: SPI MODULE BLOCK DIAGRAM



REGISTER 18-1: SPIxSTAT: SPIx STATUS AND CONTROL REGISTER

| | | | | | | | |
|--------|-----|---------|-----|-----|-----|-------|-----|
| R/W-0 | U-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| SPIEN | — | SPISIDL | — | — | — | — | — |
| bit 15 | | | | | | bit 8 | |

| | | | | | | | |
|-------|--------|-----|-----|-----|-----|--------|--------|
| U-0 | R/C-0 | U-0 | U-0 | U-0 | U-0 | R-0 | R-0 |
| — | SPIROV | — | — | — | — | SPITBF | SPIRBF |
| bit 7 | | | | | | bit 0 | |

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

- bit 15 **SPIEN:** SPIx Enable bit
 1 = Enables module and configures SCKx, SDOx, SDIx and \overline{SSx} as serial port pins
 0 = Disables module
- bit 14 **Unimplemented:** Read as '0'
- bit 13 **SPISIDL:** Stop in Idle Mode bit
 1 = Discontinue module operation when device enters Idle mode
 0 = Continue module operation in Idle mode
- bit 12-7 **Unimplemented:** Read as '0'
- 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 5-2 **Unimplemented:** Read as '0'
- bit 1 **SPITBF:** SPIx Transmit Buffer Full Status bit
 1 = Transmit not yet started, SPIxTXB is full
 0 = Transmit started, SPIxTXB is empty
 Automatically set in hardware when CPU writes SPIxBUF location, loading SPIxTXB
 Automatically cleared in hardware when SPIx module transfers data from SPIxTXB to SPIxSR
- bit 0 **SPIRBF:** SPIx Receive Buffer Full Status bit
 1 = Receive complete, SPIxRXB is full
 0 = Receive is not complete, SPIxRXB is empty
 Automatically set in hardware when SPIx transfers data from SPIxSR to SPIxRXB
 Automatically cleared in hardware when core reads SPIxBUF location, reading SPIxRXB

REGISTER 18-2: 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 | DISSDO | MODE16 | SMP | CKE ⁽¹⁾ |
| bit 15 | | | | | | | bit 8 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| SSEN ⁽³⁾ | CKP | MSTEN | SPRE<2:0> ⁽²⁾ | | | PPRE<1:0> ⁽²⁾ | |
| bit 7 | | | | | | | bit 0 |

Legend:

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

- bit 15-13 **Unimplemented:** Read as '0'
- bit 12 **DISSCK:** Disable SCKx pin bit (SPI Master modes only)
 1 = Internal SPI clock is disabled, pin functions as I/O
 0 = Internal SPI clock is enabled
- bit 11 **DISSDO:** Disable SDOx pin bit
 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⁽¹⁾
 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)⁽³⁾
 1 = SSx pin used for Slave mode
 0 = 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: The CKE bit is not used in the Framed SPI modes. Program this bit to '0' for the Framed SPI modes (FRMEN = 1).

2: Do not set both Primary and Secondary prescalers to a value of 1:1.

3: This bit must be cleared when FRMEN = 1.

REGISTER 18-2: SPIxCON1: SPIx CONTROL REGISTER 1 (CONTINUED)

bit 4-2 **SPRE<2:0>**: Secondary Prescale bits (Master mode)⁽²⁾

111 = Secondary prescale 1:1

110 = Secondary prescale 2:1

.

.

.

000 = Secondary prescale 8:1

bit 1-0 **PPRE<1:0>**: Primary Prescale bits (Master mode)⁽²⁾

11 = Primary prescale 1:1

10 = Primary prescale 4:1

01 = Primary prescale 16:1

00 = Primary prescale 64:1

Note 1: The CKE bit is not used in the Framed SPI modes. Program this bit to '0' for the Framed SPI modes (FRMEN = 1).

2: Do not set both Primary and Secondary prescalers to a value of 1:1.

3: This bit must be cleared when FRMEN = 1.

REGISTER 18-3: 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 | FRMPOL | — | — | — | — | — |
| bit 15 | | | | | | | bit 8 |
| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | U-0 |
| — | — | — | — | — | — | FRMDLY | — |
| bit 7 | | | | | | | bit 0 |

Legend:

| | | |
|-------------------|------------------|------------------------------------|
| R = Readable bit | W = Writable 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 (\overline{SSx} pin used as frame sync pulse input/output)
 0 = Framed SPIx support disabled
- bit 14 **SPIFSD:** Frame Sync Pulse Direction Control bit
 1 = Frame sync pulse input (slave)
 0 = Frame sync pulse output (master)
- bit 13 **FRMPOL:** Frame Sync Pulse Polarity bit
 1 = Frame sync pulse is active-high
 0 = Frame sync pulse is active-low
- bit 12-2 **Unimplemented:** Read as '0'
- bit 1 **FRMDLY:** 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 **Unimplemented:** This bit must not be set to '1' by the user application.

NOTES:

19.0 INTER-INTEGRATED CIRCUIT (I²C™)

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “*dsPIC33F Family Reference Manual*”, **Section 19. “Inter-Integrated Circuit (I²C™)”** (DS70195), which is available from the Microchip website (www.microchip.com).

The Inter-Integrated Circuit (I²C) module provides complete hardware support for both Slave and Multi-Master modes of the I²C serial communication standard, with a 16-bit interface.

The I²C module has a 2-pin interface:

- The SCLx pin is clock.
- The SDAx pin is data.

The I²C module offers the following key features:

- I²C interface supporting both Master and Slave modes of operation.
- I²C Slave mode supports 7 and 10-bit address.
- I²C Master mode supports 7 and 10-bit address.
- I²C port allows bidirectional transfers between master and slaves.
- Serial clock synchronization for I²C port can be used as a handshake mechanism to suspend and resume serial transfer (SCLREL control).
- I²C supports multi-master operation, detects bus collision and arbitrates accordingly.

19.1 Operating Modes

The hardware fully implements all the master and slave functions of the I²C Standard and Fast mode specifications, as well as 7 and 10-bit addressing.

The I²C module can operate either as a slave or a master on an I²C bus.

The following types of I²C operation are supported:

- I²C slave operation with 7-bit address
- I²C slave operation with 10-bit address
- I²C master operation with 7- or 10-bit address

For details about the communication sequence in each of these modes, refer to the “*dsPIC33F Family Reference Manual*”. Please see the Microchip website (www.microchip.com) for the latest dsPIC33F Family Reference Manual chapters.

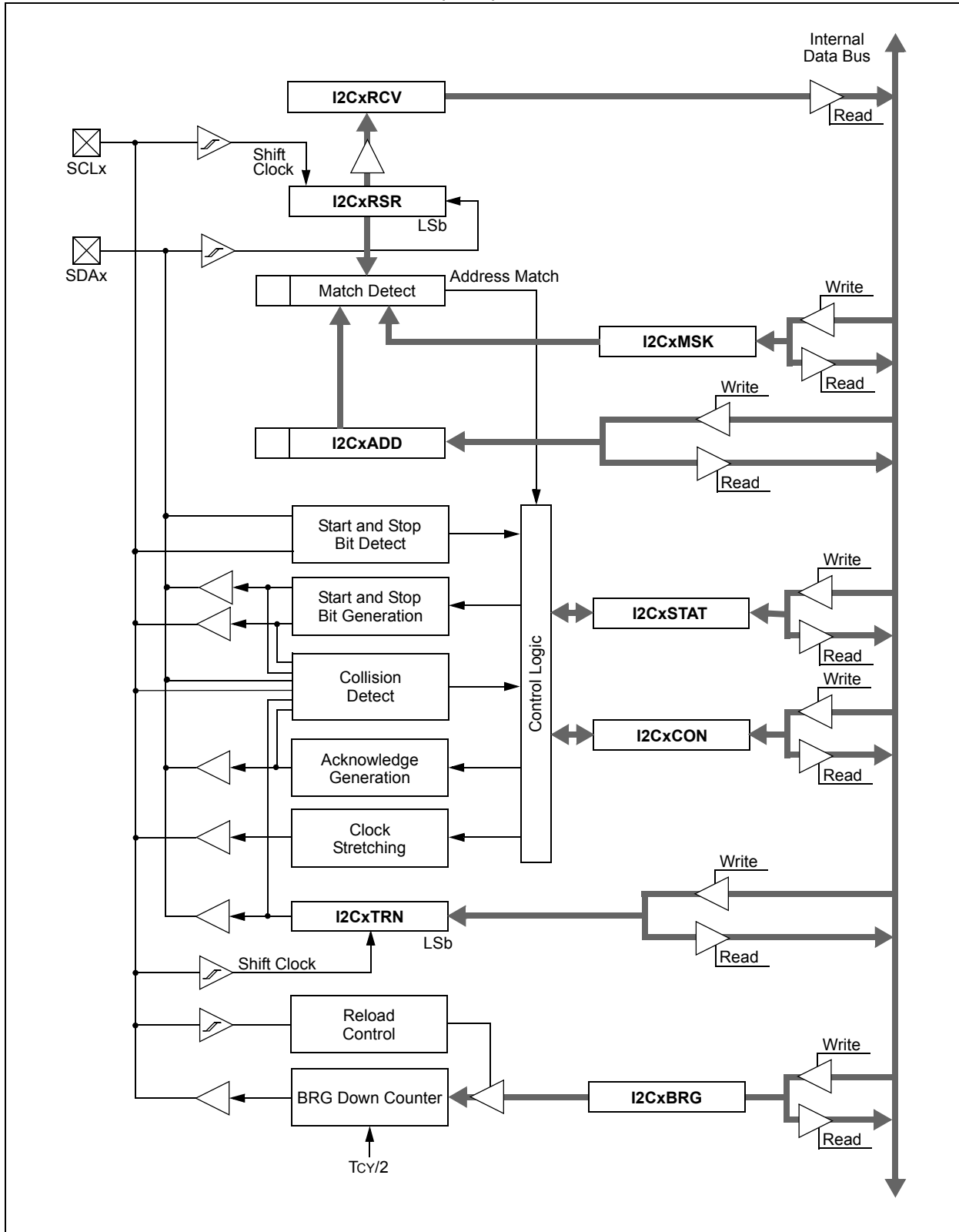
19.2 I²C Registers

I2CxCON and I2CxSTAT are control and status registers, respectively. The I2CxCON register is readable and writable. The lower six bits of I2CxSTAT are read-only. The remaining bits of the I2CxSTAT are read/write:

- I2CxRSR is the shift register used for shifting data internal to the module and the user application has no access to it.
- I2CxRCV is the receive buffer and the register to which data bytes are written, or from which data bytes are read.
- I2CxTRN is the transmit register to which bytes are written during a transmit operation.
- The I2CxADD register holds the slave address.
- A status bit, ADD10, indicates 10-bit Address mode.
- The I2CxBRG acts as the Baud Rate Generator (BRG) reload value.

In receive operations, I2CxRSR and I2CxRCV together form a double-buffered receiver. When I2CxRSR receives a complete byte, it is transferred to I2CxRCV, and an interrupt pulse is generated.

FIGURE 19-1: I²C™ BLOCK DIAGRAM (x = 1)



REGISTER 19-1: I2CxCON: I2Cx CONTROL REGISTER

| | | | | | | | |
|--------|-----|---------|----------|--------|-------|--------|-------|
| R/W-0 | U-0 | R/W-0 | R/W-1 HC | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| I2CEN | — | I2CSIDL | SCLREL | IPMIEN | A10M | DISSLW | SMEN |
| bit 15 | | | | | | bit 8 | |

| | | | | | | | |
|-------|-------|-------|----------|----------|----------|----------|----------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 HC | R/W-0 HC | R/W-0 HC | R/W-0 HC | R/W-0 HC |
| GCEN | STREN | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN |
| bit 7 | | | | | | bit 0 | |

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

- bit 15 **I2CEN:** I2Cx Enable bit
 1 = Enables the I2Cx module and configures the SDAx and SCLx pins as serial port pins
 0 = Disables the I2Cx module. All I²C™ pins are controlled by port functions
- bit 14 **Unimplemented:** Read as '0'
- bit 13 **I2CSIDL:** Stop in Idle Mode bit
 1 = Discontinue module operation when device enters an Idle mode
 0 = Continue module operation in Idle mode
- bit 12 **SCLREL:** SCLx Release Control bit (when operating as I²C slave)
 1 = Release SCLx clock
 0 = Hold SCLx clock low (clock stretch)
If STREN = 1:
 Bit is R/W (i.e., software can 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 can 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 mode is enabled; all addresses Acknowledged
 0 = IPMI mode disabled
- bit 10 **A10M:** 10-bit Slave Address 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 = Enable I/O pin thresholds compliant with SMBus specification
 0 = Disable SMBus input thresholds
- bit 7 **GCEN:** General Call Enable bit (when operating as I²C slave)
 1 = Enable 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 I²C slave)
 Used in conjunction with SCLREL bit.
 1 = Enable software or receive clock stretching
 0 = Disable software or receive clock stretching

REGISTER 19-1: I2CxCON: I2Cx CONTROL REGISTER (CONTINUED)

- bit 5 **ACKDT:** Acknowledge Data bit (when operating as I²C master, applicable during master receive)
Value that is transmitted when the software initiates an Acknowledge sequence.
1 = Send NACK during Acknowledge
0 = Send ACK during Acknowledge
- bit 4 **ACKEN:** Acknowledge Sequence Enable bit
(when operating as I²C master, applicable during master receive)
1 = Initiate Acknowledge sequence on SDAx and SCLx pins and transmit 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 I²C master)
1 = Enables Receive mode for I²C. Hardware clear at end of eighth bit of master receive data byte.
0 = Receive sequence not in progress
- bit 2 **PEN:** Stop Condition Enable bit (when operating as I²C master)
1 = Initiate 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 Enable bit (when operating as I²C master)
1 = Initiate 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 Enable bit (when operating as I²C master)
1 = Initiate Start condition on SDAx and SCLx pins. Hardware clear at end of master Start sequence.
0 = Start condition not in progress

REGISTER 19-2: 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_A | P | S | R_W | RBF | TBF |
| bit 7 | | | | | | bit 0 | |

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

- bit 15 **ACKSTAT:** Acknowledge Status bit
(when operating as I²C™ master, applicable to master transmit operation)
1 = NACK received from slave
0 = ACK received from slave
Hardware set or clear at end of slave Acknowledge.
- bit 14 **TRSTAT:** Transmit Status bit (when operating as I²C 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 I²C 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_A:** Data/Address bit (when operating as I²C 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 reception of slave byte.
- 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.

REGISTER 19-2: I2CxSTAT: I2Cx STATUS REGISTER (CONTINUED)

- 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 I²C 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 I²C 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.

REGISTER 19-3: 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 **AMSKx:** Mask for Address bit x Select bit

 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

NOTES:

20.0 UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 17. “UART” (DS70188), which is available from the Microchip website (www.microchip.com).

The Universal Asynchronous Receiver Transmitter (UART) module is one of the serial I/O modules available in the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 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:

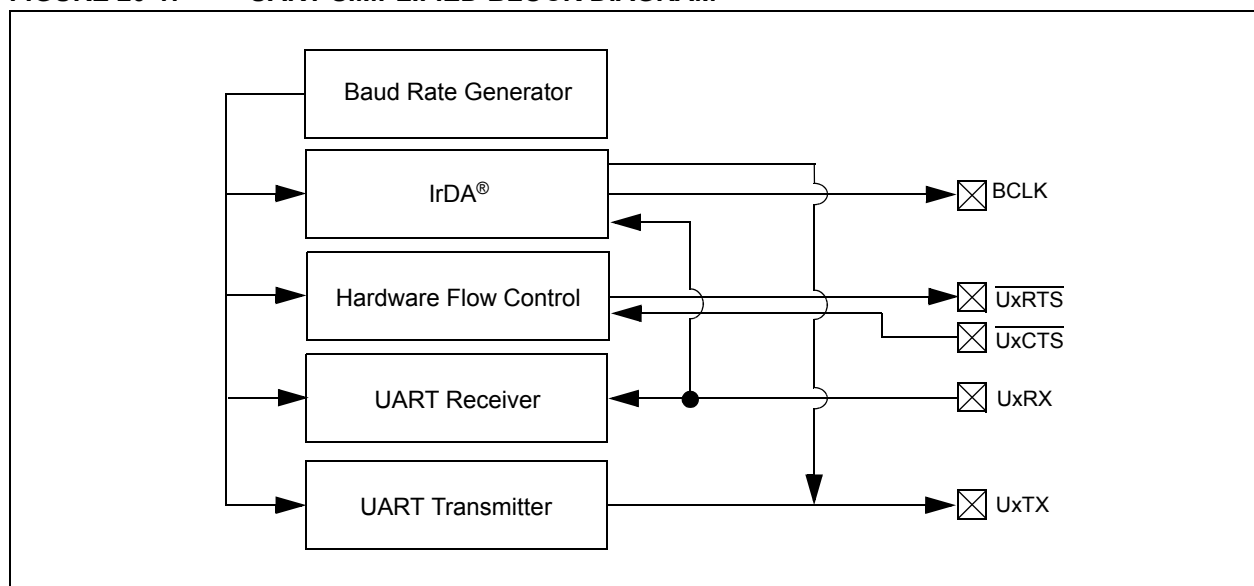
- Full-Duplex, 8- or 9-bit Data Transmission through the UxTX and UxRX pins
- Even, Odd or No Parity Options (for 8-bit data)

- One or two stop bits
- Hardware flow control option with $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ pins
- Fully integrated Baud Rate Generator with 16-bit prescaler
- Baud rates ranging from 1 Mbps to 15 bps at 16x mode at 40 MIPS
- Baud rates ranging from 4 Mbps to 61 bps at 4x mode at 40 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
- A separate interrupt for all UART error conditions
- Loopback mode for diagnostic support
- Support for sync and break characters
- Support for automatic baud rate detection
- IrDA® encoder and decoder logic
- 16x baud clock output for IrDA® support

A simplified block diagram of the UART module is shown in Figure 20-1. The UART module consists of these key hardware elements:

- Baud Rate Generator
- Asynchronous Transmitter
- Asynchronous Receiver

FIGURE 20-1: UART SIMPLIFIED BLOCK DIAGRAM



REGISTER 20-1: UxMODE: UARTx MODE REGISTER

| | | | | | | | |
|-----------------------|-----|-------|---------------------|-------|-----|----------|-------|
| R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 |
| UARTEN ⁽¹⁾ | — | USIDL | IREN ⁽²⁾ | RTSMD | — | UEN<1:0> | |
| bit 15 | | | | | | bit 8 | |

| | | | | | | | |
|----------|--------|----------|--------|-------|------------|-------|-------|
| R/W-0 HC | R/W-0 | R/W-0 HC | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| WAKE | LPBACK | ABAUD | URXINV | BRGH | PDSEL<1:0> | | STSEL |
| bit 7 | | | | | | bit 0 | |

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

- bit 15 **UARTEN:** UARTx Enable bit
 - 1 = 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 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[®] Encoder and Decoder Enable bit⁽²⁾
 - 1 = IrDA[®] encoder and decoder enabled
 - 0 = IrDA[®] encoder and decoder disabled
- bit 11 **RTSMD:** Mode Selection for UxRTS Pin bit
 - 1 = UxRTS pin in Simplex mode
 - 0 = UxRTS pin in Flow Control mode
- bit 10 **Unimplemented:** Read as '0'
- bit 9-8 **UEN<1:0>:** UARTx Enable bits
 - 11 = UxTX, UxRX and BCLK pins are enabled and used; UxCTS pin controlled by port latches
 - 10 = UxTX, UxRX, UxCTS and UxRTS pins are enabled and used
 - 01 = UxTX, UxRX and UxRTS pins are enabled and used; UxCTS pin controlled by port latches
 - 00 = UxTX and UxRX pins are enabled and used; UxCTS and UxRTS/BCLK pins controlled by port latches
- bit 7 **WAKE:** Wake-up on Start bit Detect During Sleep Mode Enable bit
 - 1 = UARTx continues to sample the 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) before other data; cleared in hardware upon completion
 - 0 = Baud rate measurement disabled or completed

Note 1: Refer to **Section 17. “UART”** (DS70188) in the “*dsPIC33F Family Reference Manual*” for information on enabling the UART module for receive or transmit operation.

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

REGISTER 20-1: UxMODE: UARTx MODE REGISTER (CONTINUED)

- bit 4 **URXINV:** Receive Polarity Inversion bit
1 = UxRX Idle state is '0'
0 = UxRX Idle state is '1'
- bit 3 **BRGH:** High Baud Rate Enable bit
1 = BRG generates 4 clocks per bit period (4x baud clock, High-Speed mode)
0 = BRG generates 16 clocks per bit period (16x baud clock, Standard mode)
- bit 2-1 **PDSEL<1:0>:** Parity and Data Selection bits
11 = 9-bit data, no parity
10 = 8-bit data, odd parity
01 = 8-bit data, even parity
00 = 8-bit data, no parity
- bit 0 **STSEL:** Stop Bit Selection bit
1 = Two Stop bits
0 = One Stop bit

Note 1: Refer to **Section 17. "UART"** (DS70188) in the "*dsPIC33F Family Reference Manual*" for information on enabling the UART module for receive or transmit operation.

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

REGISTER 20-2: 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 ⁽¹⁾ | 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 |
| URXISEL<1:0> | | ADDEN | RIDL | PERR | FERR | OERR | URXDA |
| bit 7 | | | | | | bit 0 | |

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

- bit 15,13 **UTXISEL<1:0>**: Transmission Interrupt Mode Selection bits
 - 11 = Reserved; do not use
 - 10 = Interrupt when a character is transferred to the Transmit Shift Register, 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**: Transmit Polarity Inversion bit
 - If IREN = 0:
 - 1 = UxTX Idle state is '0'
 - 0 = UxTX Idle state is '1'
 - If IREN = 1:
 - 1 = IrDA[®] encoded UxTX Idle state is '1'
 - 0 = IrDA[®] encoded UxTX Idle state is '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⁽¹⁾
 - 1 = Transmit enabled, UxTX pin controlled by UARTx
 - 0 = Transmit disabled, any pending transmission is aborted and buffer is reset. UxTX pin controlled by port.
- bit 9 **UTXBF**: Transmit Buffer Full Status bit (read-only)
 - 1 = Transmit buffer is full
 - 0 = Transmit buffer is not full, at least one more character can be written
- bit 8 **TRMT**: Transmit Shift Register Empty bit (read-only)
 - 1 = Transmit Shift Register is empty and transmit buffer is empty (the last transmission has completed)
 - 0 = Transmit Shift Register is not empty, a transmission is in progress or queued
- bit 7-6 **URXISEL<1:0>**: Receive Interrupt Mode Selection bits
 - 11 = Interrupt is set on UxRSR transfer making the receive buffer full (i.e., has 4 data characters)
 - 10 = Interrupt is set on UxRSR 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 UxRSR to the receive buffer. Receive buffer has one or more characters.

Note 1: Refer to **Section 17. “UART”** (DS70188) in the “dsPIC33F Family Reference Manual” for information on enabling the UART module for transmit operation.

REGISTER 20-2: 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 (read/clear only)
1 = Receive buffer has overflowed
0 = Receive buffer has not overflowed. Clearing a previously set OERR bit (1 → 0 transition) resets the receiver buffer and the UxRSR 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: Refer to **Section 17. “UART”** (DS70188) in the “*dsPIC33F Family Reference Manual*” for information on enabling the UART module for transmit operation.

NOTES:

21.0 ENHANCED CAN (ECAN™) MODULE

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “*dsPIC33F Family Reference Manual*”, **Section 21. “Enhanced Controller Area Network (ECAN™)”** (DS70185), which is available from the Microchip website (www.microchip.com).

21.1 Overview

The Enhanced Controller Area Network (ECAN) module is a serial interface, useful for communicating with other CAN modules or microcontroller devices. This interface/protocol was designed to allow communications within noisy environments. The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices contain up to two ECAN modules.

The ECAN module is a communication controller implementing the CAN 2.0 A/B protocol, as defined in the BOSCH CAN specification. The module supports CAN 1.2, CAN 2.0A, CAN 2.0B Passive and CAN 2.0B Active versions of the protocol. The module implementation is a full CAN system. The CAN specification is not covered within this data sheet. The reader can refer to the BOSCH CAN specification for further details.

The module features are as follows:

- Implementation of the CAN protocol, CAN 1.2, CAN 2.0A and CAN 2.0B
- Standard and extended data frames
- 0-8 bytes data length
- Programmable bit rate up to 1 Mbit/sec
- Automatic response to remote transmission requests
- Up to eight transmit buffers with application specified prioritization and abort capability (each buffer can contain up to 8 bytes of data)
- Up to 32 receive buffers (each buffer can contain up to 8 bytes of data)
- Up to 16 full (standard/extended identifier) acceptance filters
- Three full acceptance filter masks
- DeviceNet™ addressing support
- Programmable wake-up functionality with integrated low-pass filter
- Programmable Loopback mode supports self-test operation
- Signaling via interrupt capabilities for all CAN receiver and transmitter error states
- Programmable clock source

- Programmable link to input capture module (IC2 for CAN1) for time-stamping and network synchronization
- Low-power Sleep and Idle mode

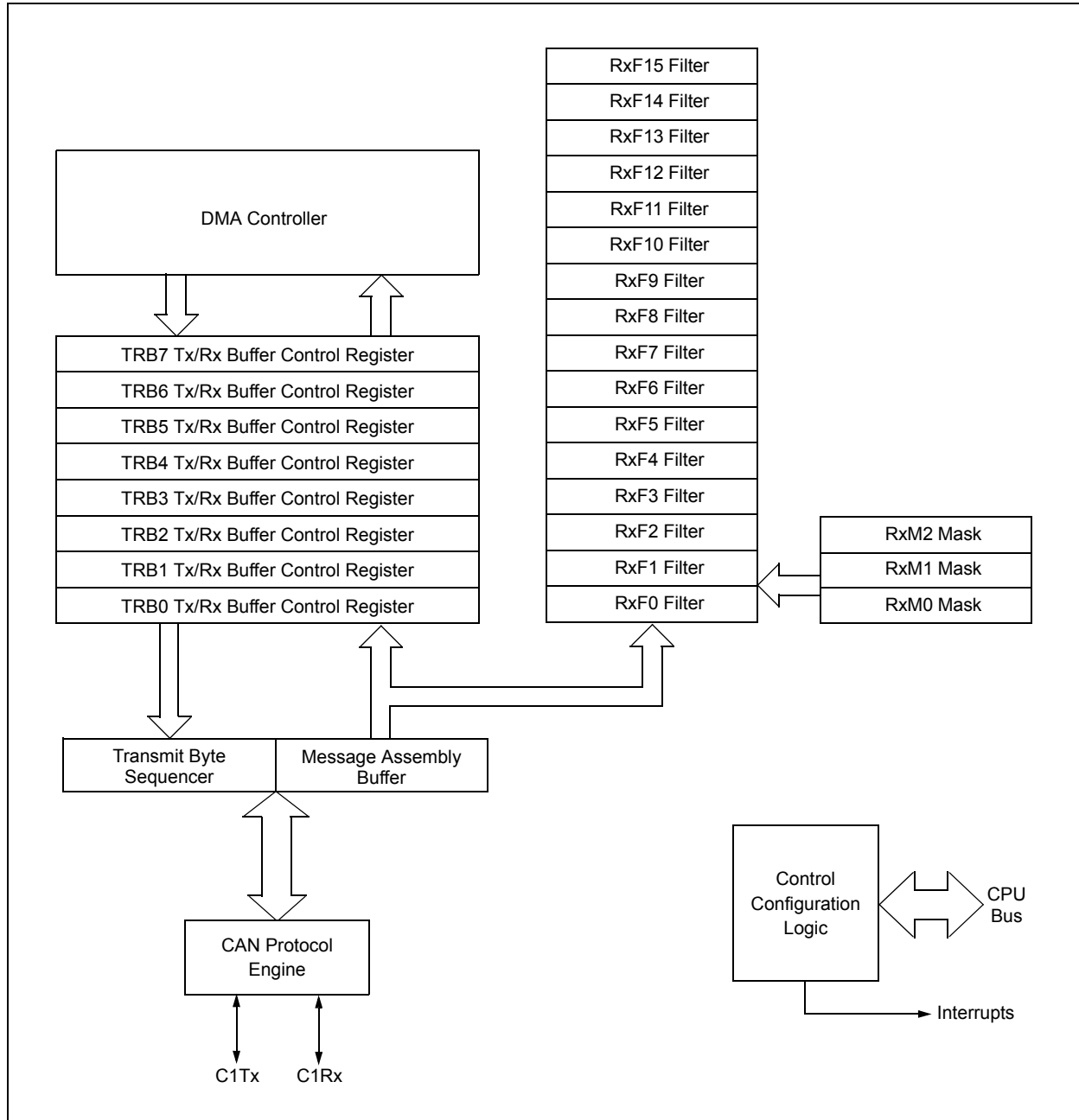
The CAN bus module consists of a protocol engine and message buffering/control. The CAN protocol engine handles all functions for receiving and transmitting messages on the CAN bus. Messages are transmitted by first loading the appropriate data registers. Status and errors can be checked by reading the appropriate registers. Any message detected on the CAN bus is checked for errors and then matched against filters to see if it should be received and stored in one of the receive registers.

21.2 Frame Types

The ECAN module transmits various types of frames which include data messages, or remote transmission requests initiated by the user, as other frames that are automatically generated for control purposes. The following frame types are supported:

- **Standard Data Frame:**
A standard data frame is generated by a node when the node wishes to transmit data. It includes an 11-bit Standard Identifier (SID), but not an 18-bit Extended Identifier (EID).
- **Extended Data Frame:**
An extended data frame is similar to a standard data frame, but includes an extended identifier as well.
- **Remote Frame:**
It is possible for a destination node to request the data from the source. For this purpose, the destination node sends a remote frame with an identifier that matches the identifier of the required data frame. The appropriate data source node sends a data frame as a response to this remote request.
- **Error Frame:**
An error frame is generated by any node that detects a bus error. An error frame consists of two fields: an error flag field and an error delimiter field.
- **Overload Frame:**
An overload frame can be generated by a node as a result of two conditions. First, the node detects a dominant bit during interframe space which is an illegal condition. Second, due to internal conditions, the node is not yet able to start reception of the next message. A node can generate a maximum of 2 sequential overload frames to delay the start of the next message.
- **Interframe Space:**
Interframe space separates a preceding frame (of whatever type) from a following data or remote frame.

FIGURE 21-1: ECAN™ MODULE BLOCK DIAGRAM



21.3 Modes of Operation

The ECAN module can operate in one of several operation modes selected by the user. These modes include:

- Initialization mode
- Disable mode
- Normal Operation mode
- Listen Only mode
- Listen All Messages mode
- Loopback mode

Modes are requested by setting the REQOP<2:0> bits (CiCTRL1<10:8>). Entry into a mode is Acknowledged by monitoring the OPMODE<2:0> bits (CiCTRL1<7:5>). The module does not change the mode and the OPMODE bits until a change in mode is acceptable, generally during bus Idle time, which is defined as at least 11 consecutive recessive bits.

21.3.1 INITIALIZATION MODE

In the Initialization mode, the module does not transmit or receive. The error counters are cleared and the interrupt flags remain unchanged. The user application has access to Configuration registers that are access restricted in other modes. The module protects the user from accidentally violating the CAN protocol through programming errors. All registers which control the configuration of the module can not be modified while the module is on-line. The ECAN module is not allowed to enter the Configuration mode while a transmission is taking place. The Configuration mode serves as a lock to protect the following registers:

- All Module Control registers
- Baud Rate and Interrupt Configuration registers
- Bus Timing registers
- Identifier Acceptance Filter registers
- Identifier Acceptance Mask registers

21.3.2 DISABLE MODE

In Disable mode, the module does not transmit or receive. The module has the ability to set the WAKIF bit due to bus activity, however, any pending interrupts remains and the error counters retains their value.

If the REQOP<2:0> bits (CiCTRL1<10:8>) = 001, the module enters the Module Disable mode. If the module is active, the module waits for 11 recessive bits on the CAN bus, detect that condition as an Idle bus, then accept the module disable command. When the OPMODE<2:0> bits (CiCTRL1<7:5>) = 001, that indicates whether the module successfully went into Module Disable mode. The I/O pins reverts to normal I/O function when the module is in the Module Disable mode.

The module can be programmed to apply a low-pass filter function to the CiRX input line while the module or the CPU is in Sleep mode. The WAKFIL bit (CiCFG2<14>) enables or disables the filter.

Note: Typically, if the ECAN module is allowed to transmit in a particular mode of operation and a transmission is requested immediately after the ECAN module has been placed in that mode of operation, the module waits for 11 consecutive recessive bits on the bus before starting transmission. If the user switches to Disable mode within this 11-bit period, then this transmission is aborted and the corresponding TXABT bit is set and TXREQ bit is cleared.

21.3.3 NORMAL OPERATION MODE

Normal Operation mode is selected when REQOP<2:0> = 000. In this mode, the module is activated and the I/O pins assumes the CAN bus functions. The module transmits and receive CAN bus messages via the CiTX and CiRX pins.

21.3.4 LISTEN ONLY MODE

If the Listen Only mode is activated, the module on the CAN bus is passive. The transmitter buffers revert to the port I/O function. The receive pins remain inputs. For the receiver, no error flags or Acknowledge signals are sent. The error counters are deactivated in this state. The Listen Only mode can be used for detecting the baud rate on the CAN bus. To use this, it is necessary that there are at least two further nodes that communicate with each other.

21.3.5 LISTEN ALL MESSAGES MODE

The module can be set to ignore all errors and receive any message. The Listen All Messages mode is activated by setting REQOP<2:0> = '111'. In this mode, the data which is in the message assembly buffer, until the time an error occurred, is copied in the receive buffer and can be read via the CPU interface.

21.3.6 LOOPBACK MODE

If the Loopback mode is activated, the module connects the internal transmit signal to the internal receive signal at the module boundary. The transmit and receive pins revert to their port I/O function.

REGISTER 21-1: CiCTRL1: ECAN™ CONTROL REGISTER 1

| | | | | | | | |
|--------|-----|-------|-------|-------|------------|-------|-------|
| U-0 | U-0 | R/W-0 | R/W-0 | r-0 | R/W-1 | R/W-0 | R/W-0 |
| — | — | CSIDL | ABAT | — | REQOP<2:0> | | |
| bit 15 | | | | bit 8 | | | |

| | | | | | | | |
|-------------|-----|-----|--------|-------|-----|-------|-------|
| R-1 | R-0 | R-0 | U-0 | R/W-0 | U-0 | U-0 | R/W-0 |
| OPMODE<2:0> | | — | CANCAP | — | — | WIN | |
| bit 7 | | | | | | bit 0 | |

| | | |
|-------------------|--|------------------------------------|
| Legend: | C = Writable bit, but only '0' can be written to clear the bit | r = Bit is Reserved |
| R = Readable bit | W = Writable 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 **CSIDL:** Stop in Idle Mode bit
 - 1 = Discontinue module operation when device enters Idle mode
 - 0 = Continue module operation in Idle mode
- bit 12 **ABAT:** Abort All Pending Transmissions bit
 - 1 = Signal all transmit buffers to abort transmission.
 - 0 = Module will clear this bit when all transmissions are aborted
- bit 11 **Reserved:** Do not use
 - 1 = CAN FCAN clock is FCY
 - 0 = CAN FCAN clock is FOSC
- bit 10-8 **REQOP<2:0>:** Request Operation Mode bits
 - 000 = Set Normal Operation mode
 - 001 = Set Disable mode
 - 010 = Set Loopback mode
 - 011 = Set Listen Only Mode
 - 100 = Set Configuration mode
 - 101 = Reserved
 - 110 = Reserved
 - 111 = Set Listen All Messages mode
- bit 7-5 **OPMODE<2:0>:** Operation Mode bits
 - 000 = Module is in Normal Operation mode
 - 001 = Module is in Disable mode
 - 010 = Module is in Loopback mode
 - 011 = Module is in Listen Only mode
 - 100 = Module is in Configuration mode
 - 101 = Reserved
 - 110 = Reserved
 - 111 = Module is in Listen All Messages mode
- bit 4 **Unimplemented:** Read as '0'
- bit 3 **CANCAP:** CAN Message Receive Timer Capture Event Enable bit
 - 1 = Enable input capture based on CAN message receive
 - 0 = Disable CAN capture
- bit 2-1 **Unimplemented:** Read as '0'
- bit 0 **WIN:** SFR Map Window Select bit
 - 1 = Use filter window
 - 0 = Use buffer window

REGISTER 21-2: CICTRL2: ECAN™ CONTROL REGISTER 2

| | | | | | | | |
|--------|-----|-----|-----|-----|-----|-------|-----|
| 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-0 | R-0 | R-0 | R-0 | R-0 |
| — | — | — | DNCNT<4:0> | | | | |
| bit 7 | | | | | | bit 0 | |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the bit |
| R = Readable bit | W = Writable bit |
| -n = Value at POR | '1' = Bit is set |
| | '0' = Bit is cleared |
| | x = Bit is unknown |

- bit 15-5 **Unimplemented:** Read as '0'
- bit 4-0 **DNCNT<4:0>:** DeviceNet™ Filter Bit Number bits
 - 10010-11111 = Invalid selection
 - 10001 = Compare up to data byte 3, bit 6 with EID<17>
 -
 -
 -
 - 00001 = Compare up to data byte 1, bit 7 with EID<0>
 - 00000 = Do not compare data bytes

REGISTER 21-3: CIVEC: ECAN™ INTERRUPT CODE REGISTER

| | | | | | | | | |
|--------|-----|-----|-------------|-----|-----|-----|-----|-------|
| U-0 | U-0 | U-0 | R-0 | R-0 | R-0 | R-0 | R-0 | |
| — | — | — | FILHIT<4:0> | | | | | |
| bit 15 | | | | | | | | bit 8 |

| | | | | | | | |
|-------|------------|-----|-----|-----|-----|-----|-------|
| U-0 | R-1 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| — | ICODE<6:0> | | | | | | |
| bit 7 | | | | | | | bit 0 |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the bit |
| R = Readable bit | W = Writable bit U = Unimplemented bit, read as '0' |
| -n = Value at POR | '1' = Bit is set '0' = Bit is cleared x = Bit is unknown |

- bit 15-13 **Unimplemented:** Read as '0'
- bit 12-8 **FILHIT<4:0>:** Filter Hit Number bits
 - 10000-11111 = Reserved
 - 01111 = Filter 15
 -
 -
 -
 - 00001 = Filter 1
 - 00000 = Filter 0
- bit 7 **Unimplemented:** Read as '0'
- bit 6-0 **ICODE<6:0>:** Interrupt Flag Code bits
 - 1000101-1111111 = Reserved
 - 1000100 = FIFO almost full interrupt
 - 1000011 = Receiver overflow interrupt
 - 1000010 = Wake-up interrupt
 - 1000001 = Error interrupt
 - 1000000 = No interrupt
 -
 -
 -
 - 0010000-0111111 = Reserved
 - 0001111 = RB15 buffer interrupt
 -
 -
 -
 - 0001001 = RB9 buffer interrupt
 - 0001000 = RB8 buffer interrupt
 - 0000111 = TRB7 buffer interrupt
 - 0000110 = TRB6 buffer interrupt
 - 0000101 = TRB5 buffer interrupt
 - 0000100 = TRB4 buffer interrupt
 - 0000011 = TRB3 buffer interrupt
 - 0000010 = TRB2 buffer interrupt
 - 0000001 = TRB1 buffer interrupt
 - 0000000 = TRB0 Buffer interrupt

REGISTER 21-4: CifCTRL: ECAN™ FIFO CONTROL REGISTER

| | | | | | | | | |
|------------|-------|-------|-----|-----|-----|-----|-----|-------|
| R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 | U-0 | |
| DMABS<2:0> | | | — | — | — | — | — | |
| 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 | |
| — | — | — | FSA<4:0> | | | | | |
| bit 7 | | | | | | | | bit 0 |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the 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 15-13 **DMABS<2:0>**: DMA Buffer Size bits

- 111 = Reserved
- 110 = 32 buffers in DMA RAM
- 101 = 24 buffers in DMA RAM
- 100 = 16 buffers in DMA RAM
- 011 = 12 buffers in DMA RAM
- 010 = 8 buffers in DMA RAM
- 001 = 6 buffers in DMA RAM
- 000 = 4 buffers in DMA RAM

bit 12-5 **Unimplemented**: Read as '0'

bit 4-0 **FSA<4:0>**: FIFO Area Starts with Buffer bits

- 11111 = Read buffer RB31
- 11110 = Read buffer RB30
-
-
-
- 00001 = Tx/Rx buffer TRB1
- 00000 = Tx/Rx buffer TRB0

REGISTER 21-5: CiFIFO: ECAN™ FIFO STATUS REGISTER

| | | | | | | | | |
|--------|-----|----------|-----|-----|-----|-----|-------|--|
| U-0 | U-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | |
| — | — | FBP<5:0> | | | | | | |
| bit 15 | | | | | | | bit 8 | |

| | | | | | | | | |
|-------|-----|-----------|-----|-----|-----|-----|-------|--|
| U-0 | U-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | |
| — | — | FNRB<5:0> | | | | | | |
| bit 7 | | | | | | | bit 0 | |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the bit |
| R = Readable bit | W = Writable bit U = Unimplemented bit, read as '0' |
| -n = Value at POR | '1' = Bit is set '0' = Bit is cleared x = Bit is unknown |

- bit 15-14 **Unimplemented:** Read as '0'
- bit 13-8 **FBP<5:0>:** FIFO Buffer Pointer bits
 - 011111 = RB31 buffer
 - 011110 = RB30 buffer
 -
 -
 -
 - 000001 = TRB1 buffer
 - 000000 = TRB0 buffer
- bit 7-6 **Unimplemented:** Read as '0'
- bit 5-0 **FNRB<5:0>:** FIFO Next Read Buffer Pointer bits
 - 011111 = RB31 buffer
 - 011110 = RB30 buffer
 -
 -
 -
 - 000001 = TRB1 buffer
 - 000000 = TRB0 buffer

REGISTER 21-6: CIINTF: ECAN™ INTERRUPT FLAG REGISTER

| | | | | | | | |
|--------|-----|------|------|------|-------|-------|-------|
| U-0 | U-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| — | — | TXBO | TXBP | RXBP | TXWAR | RXWAR | EWARN |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------|-------|-------|-----|--------|--------|-------|-------|
| R/C-0 | R/C-0 | R/C-0 | U-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 |
| IVRIF | WAKIF | ERRIF | — | FIFOIF | RBOVIF | RBIF | TBIF |
| bit 7 | | | | | | | bit 0 |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the 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 **TXBO:** Transmitter in Error State Bus Off bit
1 = Transmitter is in Bus Off state
0 = Transmitter is not in Bus Off state
- bit 12 **TXBP:** Transmitter in Error State Bus Passive bit
1 = Transmitter is in Bus Passive state
0 = Transmitter is not in Bus Passive state
- bit 11 **RXBP:** Receiver in Error State Bus Passive bit
1 = Receiver is in Bus Passive state
0 = Receiver is not in Bus Passive state
- bit 10 **TXWAR:** Transmitter in Error State Warning bit
1 = Transmitter is in Error Warning state
0 = Transmitter is not in Error Warning state
- bit 9 **RXWAR:** Receiver in Error State Warning bit
1 = Receiver is in Error Warning state
0 = Receiver is not in Error Warning state
- bit 8 **EWARN:** Transmitter or Receiver in Error State Warning bit
1 = Transmitter or Receiver is in Error State Warning state
0 = Transmitter or Receiver is not in Error State Warning state
- bit 7 **IVRIF:** Invalid Message Received Interrupt Flag bit
1 = Interrupt Request has occurred
0 = Interrupt Request has not occurred
- bit 6 **WAKIF:** Bus Wake-up Activity Interrupt Flag bit
1 = Interrupt Request has occurred
0 = Interrupt Request has not occurred
- bit 5 **ERRIF:** Error Interrupt Flag bit (multiple sources in CiINTF<13:8> register)
1 = Interrupt Request has occurred
0 = Interrupt Request has not occurred
- bit 4 **Unimplemented:** Read as '0'
- bit 3 **FIFOIF:** FIFO Almost Full Interrupt Flag bit
1 = Interrupt Request has occurred
0 = Interrupt Request has not occurred
- bit 2 **RBOVIF:** RX Buffer Overflow Interrupt Flag bit
1 = Interrupt Request has occurred
0 = Interrupt Request has not occurred
- bit 1 **RBIF:** RX Buffer Interrupt Flag bit
1 = Interrupt Request has occurred
0 = Interrupt Request has not occurred
- bit 0 **TBIF:** TX Buffer Interrupt Flag bit
1 = Interrupt Request has occurred
0 = Interrupt Request has not occurred

REGISTER 21-7: CIINTE: ECAN™ INTERRUPT ENABLE 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 |
| IVRIE | WAKIE | ERRIE | — | FIFOIE | RBOVIE | RBIE | TBIE |
| bit 7 | | | | | | | bit 0 |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the bit |
| R = Readable bit | W = Writable bit U = Unimplemented bit, read as '0' |
| -n = Value at POR | '1' = Bit is set '0' = Bit is cleared x = Bit is unknown |

- bit 15-8 **Unimplemented:** Read as '0'
- bit 7 **IVRIE:** Invalid Message Received Interrupt Enable bit
1 = Interrupt Request Enabled
0 = Interrupt Request not enabled
- bit 6 **WAKIE:** Bus Wake-up Activity Interrupt Flag bit
1 = Interrupt Request Enabled
0 = Interrupt Request not enabled
- bit 5 **ERRIE:** Error Interrupt Enable bit
1 = Interrupt Request Enabled
0 = Interrupt Request not enabled
- bit 4 **Unimplemented:** Read as '0'
- bit 3 **FIFOIE:** FIFO Almost Full Interrupt Enable bit
1 = Interrupt Request Enabled
0 = Interrupt Request not enabled
- bit 2 **RBOVIE:** RX Buffer Overflow Interrupt Enable bit
1 = Interrupt Request Enabled
0 = Interrupt Request not enabled
- bit 1 **RBIE:** RX Buffer Interrupt Enable bit
1 = Interrupt Request Enabled
0 = Interrupt Request not enabled
- bit 0 **TBIE:** TX Buffer Interrupt Enable bit
1 = Interrupt Request Enabled
0 = Interrupt Request not enabled

REGISTER 21-8: CIEC: ECAN™ TRANSMIT/RECEIVE ERROR COUNT REGISTER

| | | | | | | | |
|--------------|-----|-----|-----|-----|-----|-----|-------|
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| TERRCNT<7:0> | | | | | | | |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|--------------|-----|-----|-----|-----|-----|-----|-------|
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| RERRCNT<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the bit |
| R = Readable bit | W = Writable bit U = Unimplemented bit, read as '0' |
| -n = Value at POR | '1' = Bit is set '0' = Bit is cleared x = Bit is unknown |

bit 15-8 **TERRCNT<7:0>**: Transmit Error Count bits

bit 7-0 **RERRCNT<7:0>**: Receive Error Count bits

REGISTER 21-9: CICFG1: ECAN™ BAUD RATE CONFIGURATION REGISTER 1

| | | | | | | | |
|--------|-----|-----|-----|-----|-----|-----|-------|
| 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 |
| SJW<1:0> | | | BRP<5:0> | | | | |
| bit 7 | | | | | | | bit 0 |

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

bit 15-8 **Unimplemented:** Read as '0'

bit 7-6 **SJW<1:0>**: Synchronization Jump Width bits

- 11 = Length is 4 x T_Q
- 10 = Length is 3 x T_Q
- 01 = Length is 2 x T_Q
- 00 = Length is 1 x T_Q

bit 5-0 **BRP<5:0>**: Baud Rate Prescaler bits

- 11 1111 = T_Q = 2 x 64 x 1/FCAN
-
-
-
- 00 0010 = T_Q = 2 x 3 x 1/FCAN
- 00 0001 = T_Q = 2 x 2 x 1/FCAN
- 00 0000 = T_Q = 2 x 1 x 1/FCAN

REGISTER 21-10: C1CFG2: ECAN™ BAUD RATE CONFIGURATION REGISTER 2

| | | | | | | | |
|--------|--------|-----|-----|-----|-------------|-------|-------|
| U-0 | R/W-x | U-0 | U-0 | U-0 | R/W-x | R/W-x | R/W-x |
| — | WAKFIL | — | — | — | SEG2PH<2:0> | | |
| 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 |
| SEG2PHTS | SAM | SEG1PH<2:0> | | | PRSEG<2:0> | | |
| 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 **Unimplemented:** Read as '0'
- bit 14 **WAKFIL:** Select CAN bus Line Filter for Wake-up bit
 - 1 = Use CAN bus line filter for wake-up
 - 0 = CAN bus line filter is not used for wake-up
- bit 13-11 **Unimplemented:** Read as '0'
- bit 10-8 **SEG2PH<2:0>:** Phase Segment 2 bits
 - 111 = Length is 8 x Tq
 -
 -
 -
 - 000 = Length is 1 x Tq
- bit 7 **SEG2PHTS:** Phase Segment 2 Time Select bit
 - 1 = Freely programmable
 - 0 = Maximum of SEG1PH bits or Information Processing Time (IPT), whichever is greater
- bit 6 **SAM:** Sample of the CAN bus Line bit
 - 1 = Bus line is sampled three times at the sample point
 - 0 = Bus line is sampled once at the sample point
- bit 5-3 **SEG1PH<2:0>:** Phase Segment 1 bits
 - 111 = Length is 8 x Tq
 -
 -
 -
 - 000 = Length is 1 x Tq
- bit 2-0 **PRSEG<2:0>:** Propagation Time Segment bits
 - 111 = Length is 8 x Tq
 -
 -
 -
 - 000 = Length is 1 x Tq

REGISTER 21-11: CiFEN1: ECAN™ ACCEPTANCE FILTER ENABLE REGISTER

| | | | | | | | |
|---------|---------|---------|---------|---------|---------|--------|--------|
| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
| FLTEN15 | FLTEN14 | FLTEN13 | FLTEN12 | FLTEN11 | FLTEN10 | FLTEN9 | FLTEN8 |
| bit 15 | | | | | | bit 8 | |

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|
| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
| FLTEN7 | FLTEN6 | FLTEN5 | FLTEN4 | FLTEN3 | FLTEN2 | FLTEN1 | FLTEN0 |
| bit 7 | | | | | | bit 0 | |

Legend: C = Writable bit, but only '0' can be written to clear the 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 15-0 **FLTENn:** Enable Filter n to Accept Messages bits
 1 = Enable Filter n
 0 = Disable Filter n

REGISTER 21-12: CiBUFNT1: ECAN™ FILTER 0-3 BUFFER POINTER 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 |
| F3BP<3:0> | | | | F2BP<3:0> | | | |
| 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 |
| F1BP<3:0> | | | | F0BP<3:0> | | | |
| bit 7 | | | | | | bit 0 | |

Legend: C = Writable bit, but only '0' can be written to clear the 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 15-12 **F3BP<3:0>:** RX Buffer mask for Filter 3
 1111 = Filter hits received in RX FIFO buffer
 1110 = Filter hits received in RX Buffer 14
 •
 •
 •
 0001 = Filter hits received in RX Buffer 1
 0000 = Filter hits received in RX Buffer 0

bit 11-8 **F2BP<3:0>:** RX Buffer mask for Filter 2 (same values as bit 15-12)
 bit 7-4 **F1BP<3:0>:** RX Buffer mask for Filter 1 (same values as bit 15-12)
 bit 3-0 **F0BP<3:0>:** RX Buffer mask for Filter 0 (same values as bit 15-12)

REGISTER 21-13: CIBUFPNT2: ECAN™ FILTER 4-7 BUFFER POINTER 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 |
| F7BP<3:0> | | | | F6BP<3:0> | | | |
| 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 |
| F5BP<3:0> | | | | F4BP<3:0> | | | |
| bit 7 | | | | bit 0 | | | |

| | | | |
|-------------------|--|------------------------------------|--------------------|
| Legend: | C = Writable bit, but only '0' can be written to clear the 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 15-12 **F7BP<3:0>**: RX Buffer mask for Filter 7
 - 1111 = Filter hits received in RX FIFO buffer
 - 1110 = Filter hits received in RX Buffer 14
 -
 -
 -
 - 0001 = Filter hits received in RX Buffer 1
 - 0000 = Filter hits received in RX Buffer 0
- bit 11-8 **F6BP<3:0>**: RX Buffer mask for Filter 6 (same values as bit 15-12)
- bit 7-4 **F5BP<3:0>**: RX Buffer mask for Filter 5 (same values as bit 15-12)
- bit 3-0 **F4BP<3:0>**: RX Buffer mask for Filter 4 (same values as bit 15-12)

REGISTER 21-14: CIBUFPNT3: ECAN™ FILTER 8-11 BUFFER POINTER 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 |
| F11BP<3:0> | | | | F10BP<3:0> | | | |
| 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 |
| F9BP<3:0> | | | | F8BP<3:0> | | | |
| bit 7 | | | | bit 0 | | | |

| | | | |
|-------------------|--|------------------------------------|--------------------|
| Legend: | C = Writable bit, but only '0' can be written to clear the 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 15-12 **F11BP<3:0>**: RX Buffer mask for Filter 11
 - 1111 = Filter hits received in RX FIFO buffer
 - 1110 = Filter hits received in RX Buffer 14
 -
 -
 -
 - 0001 = Filter hits received in RX Buffer 1
 - 0000 = Filter hits received in RX Buffer 0
- bit 11-8 **F10BP<3:0>**: RX Buffer mask for Filter 10 (same values as bit 15-12)
- bit 7-4 **F9BP<3:0>**: RX Buffer mask for Filter 9 (same values as bit 15-12)
- bit 3-0 **F8BP<3:0>**: RX Buffer mask for Filter 8 (same values as bit 15-12)

REGISTER 21-15: CiBUFNT4: ECAN™ FILTER 12-15 BUFFER POINTER 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 |
| F15BP<3:0> | | | | F14BP<3:0> | | | |
| 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 |
| F13BP<3:0> | | | | F12BP<3:0> | | | |
| bit 7 | | | | bit 0 | | | |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the 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 15-12 **F15BP<3:0>**: RX Buffer mask for Filter 15
 - 1111 = Filter hits received in RX FIFO buffer
 - 1110 = Filter hits received in RX Buffer 14
 -
 -
 -
 - 0001 = Filter hits received in RX Buffer 1
 - 0000 = Filter hits received in RX Buffer 0
- bit 11-8 **F14BP<3:0>**: RX Buffer mask for Filter 14 (same values as bit 15-12)
- bit 7-4 **F13BP<3:0>**: RX Buffer mask for Filter 13 (same values as bit 15-12)
- bit 3-0 **F12BP<3:0>**: RX Buffer mask for Filter 12 (same values as bit 15-12)

**REGISTER 21-16: CIRXFnSID: ECAN™ ACCEPTANCE FILTER STANDARD IDENTIFIER REGISTER
n (n = 0-15)**

| | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|
| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
| SID10 | SID9 | SID8 | SID7 | SID6 | SID5 | SID4 | SID3 |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------|-------|-------|-----|-------|-----|-------|-------|
| R/W-x | R/W-x | R/W-x | U-0 | R/W-x | U-0 | R/W-x | R/W-x |
| SID2 | SID1 | SID0 | — | EXIDE | — | EID17 | EID16 |
| bit 7 | | | | | | | bit 0 |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the 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 15-5 **SID<10:0>**: Standard Identifier bits
 1 = Message address bit SIDx must be '1' to match filter
 0 = Message address bit SIDx must be '0' to match filter

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

bit 3 **EXIDE**: Extended Identifier Enable bit
 If MIDE = 1 then:
 1 = Match only messages with extended identifier addresses
 0 = Match only messages with standard identifier addresses
 If MIDE = 0 then:
 Ignore EXIDE bit.

bit 2 **Unimplemented**: Read as '0'

bit 1-0 **EID<17:16>**: Extended Identifier bits
 1 = Message address bit EIDx must be '1' to match filter
 0 = Message address bit EIDx must be '0' to match filter

**REGISTER 21-17: CiRXFnEID: ECAN™ ACCEPTANCE FILTER EXTENDED IDENTIFIER REGISTER
n (n = 0-15)**

| | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|
| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
| EID15 | EID14 | EID13 | EID12 | EID11 | EID10 | EID9 | EID8 |
| 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 |
| EID7 | EID6 | EID5 | EID4 | EID3 | EID2 | EID1 | EID0 |
| bit 7 | | | | | | | bit 0 |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the 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 15-0 **EID<15:0>**: Extended Identifier bits
 1 = Message address bit EIDx must be '1' to match filter
 0 = Message address bit EIDx must be '0' to match filter

REGISTER 21-18: CiFMSKSEL1: ECAN™ FILTER 7-0 MASK SELECTION 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 |
| F7MSK<1:0> | | F6MSK<1:0> | | F5MSK<1:0> | | F4MSK<1:0> | |
| 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 |
| F3MSK<1:0> | | F2MSK<1:0> | | F1MSK<1:0> | | F0MSK<1:0> | |
| bit 7 | | | | | | | bit 0 |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the bit |
| R = Readable bit | W = Writable bit U = Unimplemented bit, read as '0' |
| -n = Value at POR | '1' = Bit is set '0' = Bit is cleared x = Bit is unknown |

bit 15-14 **F7MSK<1:0>**: Mask Source for Filter 7 bit
 11 = No mask
 10 = Acceptance Mask 2 registers contain mask
 01 = Acceptance Mask 1 registers contain mask
 00 = Acceptance Mask 0 registers contain mask

bit 13-12 **F6MSK<1:0>**: Mask Source for Filter 6 bit (same values as bit 15-14)

bit 11-10 **F5MSK<1:0>**: Mask Source for Filter 5 bit (same values as bit 15-14)

bit 9-8 **F4MSK<1:0>**: Mask Source for Filter 4 bit (same values as bit 15-14)

bit 7-6 **F3MSK<1:0>**: Mask Source for Filter 3 bit (same values as bit 15-14)

bit 5-4 **F2MSK<1:0>**: Mask Source for Filter 2 bit (same values as bit 15-14)

bit 3-2 **F1MSK<1:0>**: Mask Source for Filter 1 bit (same values as bit 15-14)

bit 1-0 **F0MSK<1:0>**: Mask Source for Filter 0 bit (same values as bit 15-14)

REGISTER 21-19: C1FMSKSEL2: ECAN™ FILTER 15-8 MASK SELECTION 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 |
| F15MSK<1:0> | | F14MSK<1:0> | | F13MSK<1:0> | | F12MSK<1:0> | |
| 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 |
| F11MSK<1:0> | | F10MSK<1:0> | | F9MSK<1:0> | | F8MSK<1:0> | |
| bit 7 | | | | | | | bit 0 |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the bit |
| R = Readable bit | W = Writable bit U = Unimplemented bit, read as '0' |
| -n = Value at POR | '1' = Bit is set '0' = Bit is cleared x = Bit is unknown |

- bit 15-14 **F15MSK<1:0>**: Mask Source for Filter 15 bit
 11 = No mask
 10 = Acceptance Mask 2 registers contain mask
 01 = Acceptance Mask 1 registers contain mask
 00 = Acceptance Mask 0 registers contain mask
- bit 13-12 **F14MSK<1:0>**: Mask Source for Filter 14 bit (same values as bit 15-14)
- bit 11-10 **F13MSK<1:0>**: Mask Source for Filter 13 bit (same values as bit 15-14)
- bit 9-8 **F12MSK<1:0>**: Mask Source for Filter 12 bit (same values as bit 15-14)
- bit 7-6 **F11MSK<1:0>**: Mask Source for Filter 11 bit (same values as bit 15-14)
- bit 5-4 **F10MSK<1:0>**: Mask Source for Filter 10 bit (same values as bit 15-14)
- bit 3-2 **F9MSK<1:0>**: Mask Source for Filter 9 bit (same values as bit 15-14)
- bit 1-0 **F8MSK<1:0>**: Mask Source for Filter 8 bit (same values as bit 15-14)

REGISTER 21-20: CIRXMnSID: ECAN™ ACCEPTANCE FILTER MASK STANDARD IDENTIFIER REGISTER n (n = 0-2)

| | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|
| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
| SID10 | SID9 | SID8 | SID7 | SID6 | SID5 | SID4 | SID3 |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------|-------|-------|-----|-------|-----|-------|-------|
| R/W-x | R/W-x | R/W-x | U-0 | R/W-x | U-0 | R/W-x | R/W-x |
| SID2 | SID1 | SID0 | — | MIDE | — | EID17 | EID16 |
| bit 7 | | | | | | | bit 0 |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the 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 15-5 **SID<10:0>**: Standard Identifier bits
 - 1 = Include bit SIDx in filter comparison
 - 0 = Bit SIDx is don't care in filter comparison
- bit 4 **Unimplemented**: Read as '0'
- bit 3 **MIDE**: Identifier Receive Mode bit
 - 1 = Match only message types (standard or extended address) that correspond to EXIDE bit in filter
 - 0 = Match either standard or extended address message if filters match
(i.e., if (Filter SID) = (Message SID) or if (Filter SID/EID) = (Message SID/EID))
- bit 2 **Unimplemented**: Read as '0'
- bit 1-0 **EID<17:16>**: Extended Identifier bits
 - 1 = Include bit EIDx in filter comparison
 - 0 = Bit EIDx is don't care in filter comparison

REGISTER 21-21: CIRXMnEID: ECAN™ ACCEPTANCE FILTER MASK EXTENDED IDENTIFIER REGISTER n (n = 0-2)

| | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|
| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
| EID15 | EID14 | EID13 | EID12 | EID11 | EID10 | EID9 | EID8 |
| 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 |
| EID7 | EID6 | EID5 | EID4 | EID3 | EID2 | EID1 | EID0 |
| bit 7 | | | | | | | bit 0 |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the 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 15-0 **EID<15:0>**: Extended Identifier bits
 - 1 = Include bit EIDx in filter comparison
 - 0 = Bit EIDx is don't care in filter comparison

REGISTER 21-22: CiRXFUL1: ECAN™ RECEIVE BUFFER FULL REGISTER 1

| | | | | | | | |
|---------|---------|---------|---------|---------|---------|--------|--------|
| R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 |
| RXFUL15 | RXFUL14 | RXFUL13 | RXFUL12 | RXFUL11 | RXFUL10 | RXFUL9 | RXFUL8 |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|
| R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 |
| RXFUL7 | RXFUL6 | RXFUL5 | RXFUL4 | RXFUL3 | RXFUL2 | RXFUL1 | RXFUL0 |
| bit 7 | | | | | | | bit 0 |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the 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 15-0 **RXFUL<15:0>**: Receive Buffer n Full bits
 1 = Buffer is full (set by module)
 0 = Buffer is empty

REGISTER 21-23: CiRXFUL2: ECAN™ RECEIVE BUFFER FULL REGISTER 2

| | | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|---------|
| R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 |
| RXFUL31 | RXFUL30 | RXFUL29 | RXFUL28 | RXFUL27 | RXFUL26 | RXFUL25 | RXFUL24 |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|---------|
| R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 |
| RXFUL23 | RXFUL22 | RXFUL21 | RXFUL20 | RXFUL19 | RXFUL18 | RXFUL17 | RXFUL16 |
| bit 7 | | | | | | | bit 0 |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the 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 15-0 **RXFUL<31:16>**: Receive Buffer n Full bits
 1 = Buffer is full (set by module)
 0 = Buffer is empty

REGISTER 21-24: C_iRXOVF1: ECAN™ RECEIVE BUFFER OVERFLOW REGISTER 1

| | | | | | | | |
|---------|---------|---------|---------|---------|---------|--------|--------|
| R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 |
| RXOVF15 | RXOVF14 | RXOVF13 | RXOVF12 | RXOVF11 | RXOVF10 | RXOVF9 | RXOVF8 |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|
| R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 |
| RXOVF7 | RXOVF6 | RXOVF5 | RXOVF4 | RXOVF3 | RXOVF2 | RXOVF1 | RXOVF0 |
| bit 7 | | | | | | | bit 0 |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the 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 15-0 **RXOVF<15:0>**: Receive Buffer n Overflow bits
 1 = Module attempted to write to a full buffer (set by module)
 0 = No overflow condition

REGISTER 21-25: C_iRXOVF2: ECAN™ RECEIVE BUFFER OVERFLOW REGISTER 2

| | | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|---------|
| R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 |
| RXOVF31 | RXOVF30 | RXOVF29 | RXOVF28 | RXOVF27 | RXOVF26 | RXOVF25 | RXOVF24 |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|---------|
| R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 | R/C-0 |
| RXOVF23 | RXOVF22 | RXOVF21 | RXOVF20 | RXOVF19 | RXOVF18 | RXOVF17 | RXOVF16 |
| bit 7 | | | | | | | bit 0 |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the 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 15-0 **RXOVF<31:16>**: Receive Buffer n Overflow bits
 1 = Module attempted to write to a full buffer (set by module)
 0 = No overflow condition

REGISTER 21-26: CiTRmnCON: ECAN™ Tx/Rx BUFFER m CONTROL REGISTER
(m = 0,2,4,6; n = 1,3,5,7)

| | | | | | | | |
|--------|--------|---------|--------|--------|--------|-------------|-------|
| R/W-0 | R-0 | R-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| TXENn | TXABTn | TXLARBn | TXERRn | TXREQn | RTRENn | TXnPRI<1:0> | |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------|-----------------------|------------------------|-----------------------|--------|--------|-------------|-------|
| R/W-0 | R-0 | R-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| TXENm | TXABTm ⁽¹⁾ | TXLARBm ⁽¹⁾ | TXERRm ⁽¹⁾ | TXREQm | RTRENm | TXmPRI<1:0> | |
| bit 7 | | | | | | | bit 0 |

| | |
|-------------------|--|
| Legend: | C = Writable bit, but only '0' can be written to clear the bit |
| R = Readable bit | W = Writable bit U = Unimplemented bit, read as '0' |
| -n = Value at POR | '1' = Bit is set '0' = Bit is cleared x = Bit is unknown |

- bit 15-8 See Definition for Bits 7-0, Controls Buffer n
- bit 7 **TXENm:** TX/RX Buffer Selection bit
 1 = Buffer TRBn is a transmit buffer
 0 = Buffer TRBn is a receive buffer
- bit 6 **TXABTm:** Message Aborted bit⁽¹⁾
 1 = Message was aborted
 0 = Message completed transmission successfully
- bit 5 **TXLARBm:** Message Lost Arbitration bit⁽¹⁾
 1 = Message lost arbitration while being sent
 0 = Message did not lose arbitration while being sent
- bit 4 **TXERRm:** Error Detected During Transmission bit⁽¹⁾
 1 = A bus error occurred while the message was being sent
 0 = A bus error did not occur while the message was being sent
- bit 3 **TXREQm:** Message Send Request bit
 1 = Requests that a message be sent. The bit automatically clears when the message is successfully sent.
 0 = Clearing the bit to '0' while set requests a message abort.
- bit 2 **RTRENm:** Auto-Remote Transmit Enable bit
 1 = When a remote transmit is received, TXREQ will be set
 0 = When a remote transmit is received, TXREQ will be unaffected
- bit 1-0 **TXmPRI<1:0>:** Message Transmission Priority bits
 11 = Highest message priority
 10 = High intermediate message priority
 01 = Low intermediate message priority
 00 = Lowest message priority

Note 1: This bit is cleared when TXREQ is set.

Note: The buffers, SID, EID, DLC, Data Field and Receive Status registers are located in DMA RAM.

21.4 ECAN Message Buffers

ECAN Message Buffers are part of DMA RAM Memory. They are not ECAN special function registers. The user application must directly write into the DMA RAM area that is configured for ECAN Message Buffers. The location and size of the buffer area is defined by the user application.

BUFFER 21-1: ECAN™ MESSAGE BUFFER WORD 0

| | | | | | | | |
|--------|-----|-----|-------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
| — | — | — | SID10 | SID9 | SID8 | SID7 | SID6 |
| 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 |
| SID5 | SID4 | SID3 | SID2 | SID1 | SID0 | SRR | IDE |
| bit 7 | | | | | | | bit 0 |

| | | | |
|-------------------|------------------|------------------------------------|--------------------|
| Legend: | | | |
| R = Readable bit | W = Writable 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 12-2 **SID<10:0>:** Standard Identifier bits
- bit 1 **SRR:** Substitute Remote Request bit
 - 1 = Message will request remote transmission
 - 0 = Normal message
- bit 0 **IDE:** Extended Identifier bit
 - 1 = Message will transmit extended identifier
 - 0 = Message will transmit standard identifier

BUFFER 21-2: ECAN™ MESSAGE BUFFER WORD 1

| | | | | | | | |
|--------|-----|-----|-----|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | U-0 | R/W-x | R/W-x | R/W-x | R/W-x |
| — | — | — | — | EID17 | EID16 | EID15 | EID14 |
| 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 |
| EID13 | EID12 | EID11 | EID10 | EID9 | EID8 | EID7 | EID6 |
| bit 7 | | | | | | | bit 0 |

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

- bit 15-12 **Unimplemented:** Read as '0'
- bit 11-0 **EID<17:6>:** Extended Identifier bits

BUFFER 21-3: ECAN™ MESSAGE BUFFER WORD 2

| | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|
| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
| EID5 | EID4 | EID3 | EID2 | EID1 | EID0 | RTR | RB1 |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------|-----|-----|-------|-------|-------|-------|-------|
| U-x | U-x | U-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
| — | — | — | RB0 | DLC3 | DLC2 | DLC1 | DLC0 |
| bit 7 | | | | | | | bit 0 |

Legend:

R = Readable bit W = Writable 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 **EID<5:0>**: Extended Identifier bits
- bit 9 **RTR**: Remote Transmission Request bit
 1 = Message will request remote transmission
 0 = Normal message
- bit 8 **RB1**: Reserved Bit 1
 User must set this bit to '0' per CAN protocol.
- bit 7-5 **Unimplemented**: Read as '0'
- bit 4 **RB0**: Reserved Bit 0
 User must set this bit to '0' per CAN protocol.
- bit 3-0 **DLC<3:0>**: Data Length Code bits

BUFFER 21-4: ECAN™ MESSAGE BUFFER WORD 3

| | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|
| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
| Byte 1 | | | | | | | |
| 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 |
| Byte 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 15-8 **Byte 1<15:8>**: ECAN™ Message byte 0
- bit 7-0 **Byte 0<7:0>**: ECAN Message byte 1

BUFFER 21-5: ECAN™ MESSAGE BUFFER WORD 4

| | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|
| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
| Byte 3 | | | | | | | |
| 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 |
| Byte 2 | | | | | | | |
| bit 7 | | | | bit 0 | | | |

Legend:

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

bit 15-8 **Byte 3<15:8>**: ECAN™ Message byte 3

bit 7-0 **Byte 2<7:0>**: ECAN Message byte 2

BUFFER 21-6: ECAN™ MESSAGE BUFFER WORD 5

| | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|
| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
| Byte 5 | | | | | | | |
| 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 |
| Byte 4 | | | | | | | |
| bit 7 | | | | bit 0 | | | |

Legend:

R = Readable bit W = Writable 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 **Byte 5<15:8>**: ECAN™ Message byte 5

bit 7-0 **Byte 4<7:0>**: ECAN Message byte 4

BUFFER 21-7: ECAN™ MESSAGE BUFFER WORD 6

| | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|
| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
| Byte 7 | | | | | | | |
| 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 |
| Byte 6 | | | | | | | |
| bit 7 | | | | | | | bit 0 |

Legend:

R = Readable bit W = Writable 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 **Byte 7<15:8>**: ECAN™ Message byte 7

bit 7-0 **Byte 6<7:0>**: ECAN Message byte 6

BUFFER 21-8: ECAN™ MESSAGE BUFFER WORD 7

| | | | | | | | |
|--------|-----|-----|----------------------------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
| — | — | — | FILHIT<4:0> ⁽¹⁾ | | | | |
| 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 **FILHIT<4:0>**: Filter Hit Code bits⁽¹⁾
 Encodes number of filter that resulted in writing this buffer.

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

Note 1: Only written by module for receive buffers, unused for transmit buffers.

22.0 10-BIT/12-BIT ANALOG-TO-DIGITAL CONVERTER (ADC1)

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 16. “Analog-to-Digital Converter (ADC)” (DS70183), which is available from the Microchip website (www.microchip.com).

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices have up to 9 ADC input channels.

The AD12B bit (AD1CON1<10>) allows each of the ADC modules to be configured by the user as either a 10-bit, 4-sample/hold ADC (default configuration) or a 12-bit, 1-sample/hold ADC.

Note: The ADC module needs to be disabled before modifying the AD12B bit.

22.1 Key Features

The 10-bit ADC configuration has the following key features:

- Successive Approximation (SAR) conversion
- Conversion speeds of up to 1.1 Msps
- Up to nine analog input pins
- External voltage reference input pins
- Simultaneous sampling of up to four analog input pins
- Automatic Channel Scan mode
- Selectable conversion trigger source
- Selectable Buffer Fill modes
- Four result alignment options (signed/unsigned, fractional/integer)
- Operation during CPU Sleep and Idle modes

The 12-bit ADC configuration supports all the above features, except:

- In the 12-bit configuration, conversion speeds of up to 500 ksps are supported
- There is only one sample/hold amplifier in the 12-bit configuration, so simultaneous sampling of multiple channels is not supported.

Depending on the particular device pinout, the ADC can have up to nine analog input pins, designated AN0 through AN8. In addition, there are two analog input pins for external voltage reference connections. These voltage reference inputs can be shared with other analog input pins. The actual number of analog input pins and external voltage reference input configuration depends on the specific device.

Block diagrams of the ADC module are shown in Figure 22-1 and Figure 22-2.

22.2 ADC Initialization

The following configuration steps should be performed.

1. Configure the ADC module:
 - a) Select port pins as analog inputs (AD1PCFGH<15:0> or AD1PCFGL<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) Determine how many S/H channels is used (AD1CON2<9:8> and AD1PCFGH<15:0> or AD1PCFGL<15:0>)
 - e) Select the appropriate sample/conversion sequence (AD1CON1<7:5> and AD1CON3<12:8>)
 - f) Select how conversion results are presented in the buffer (AD1CON1<9:8>)
 - g) Turn on ADC module (AD1CON1<15>)
2. Configure ADC interrupt (if required):
 - a) Clear the AD1IF bit
 - b) Select ADC interrupt priority

22.3 ADC and DMA

If more than one conversion result needs to be buffered before triggering an interrupt, DMA data transfers can be used. ADC1 can trigger a DMA data transfer. If ADC1 is selected as the DMA IRQ source, a DMA transfer occurs when the AD1IF bit gets set as a result of an ADC1 sample conversion sequence.

The SMPI<3:0> bits (AD1CON2<5:2>) are used to select how often the DMA RAM buffer pointer is incremented.

The ADDMABM bit (AD1CON1<12>) determines how the conversion results are filled in the DMA RAM buffer area being used for ADC. If this bit is set, DMA buffers are written in the order of conversion. The module provides an address to the DMA channel that is the same as the address used for the non-DMA stand-alone buffer. If the ADDMABM bit is cleared, then DMA buffers are written in Scatter/Gather mode. The module provides a scatter/gather address to the DMA channel, based on the index of the analog input and the size of the DMA buffer.

FIGURE 22-1: ADC MODULE BLOCK DIAGRAM FOR dsPIC33FJ32MC304, dsPIC33FJ64MC204/804, AND dsPIC33FJ128MC204/804 DEVICES

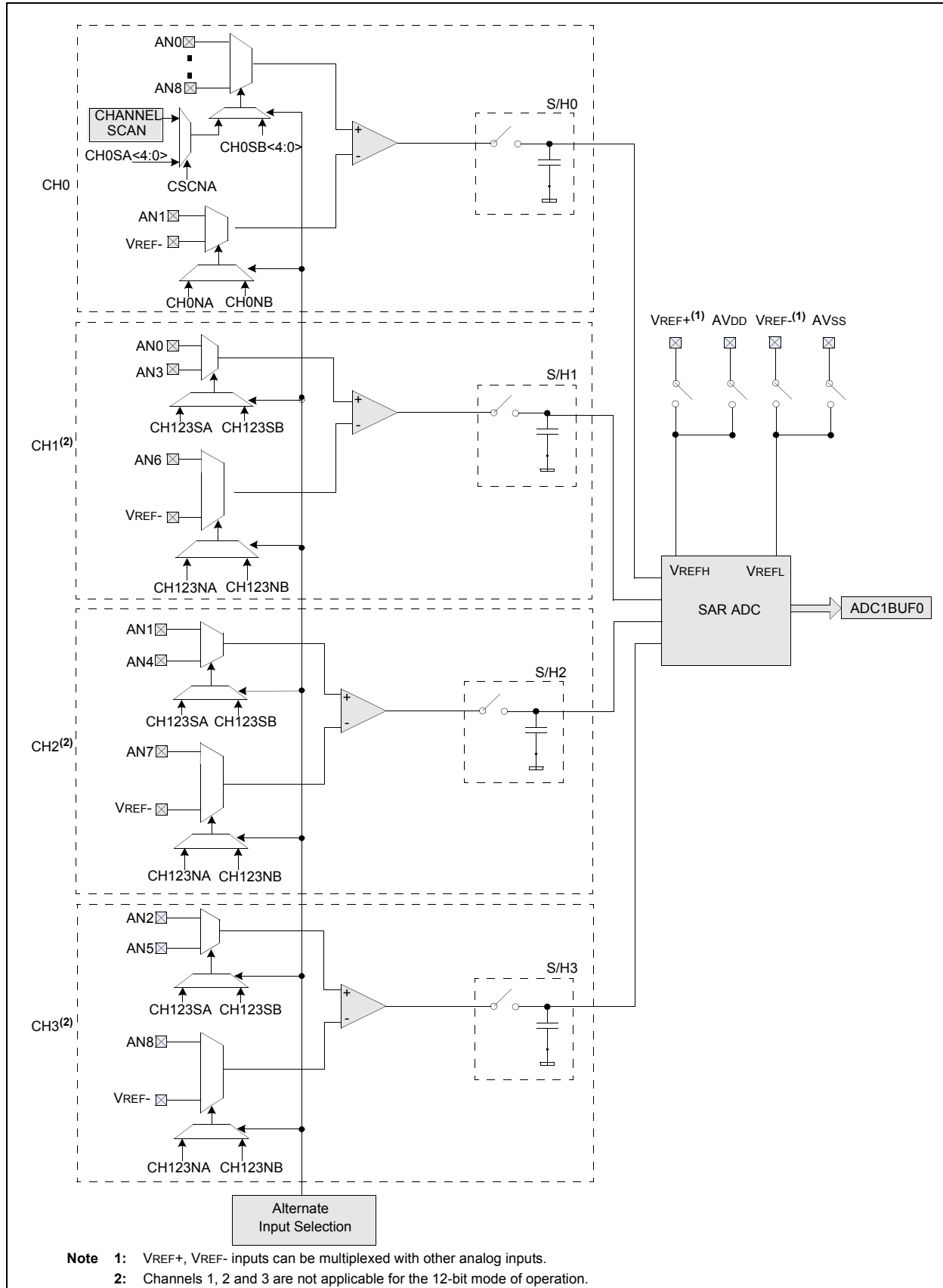


FIGURE 22-2: ADC1 MODULE BLOCK DIAGRAM FOR dsPIC33FJ32MC302, dsPIC33FJ64MC202/802, AND dsPIC33FJ128MC202/802 DEVICES

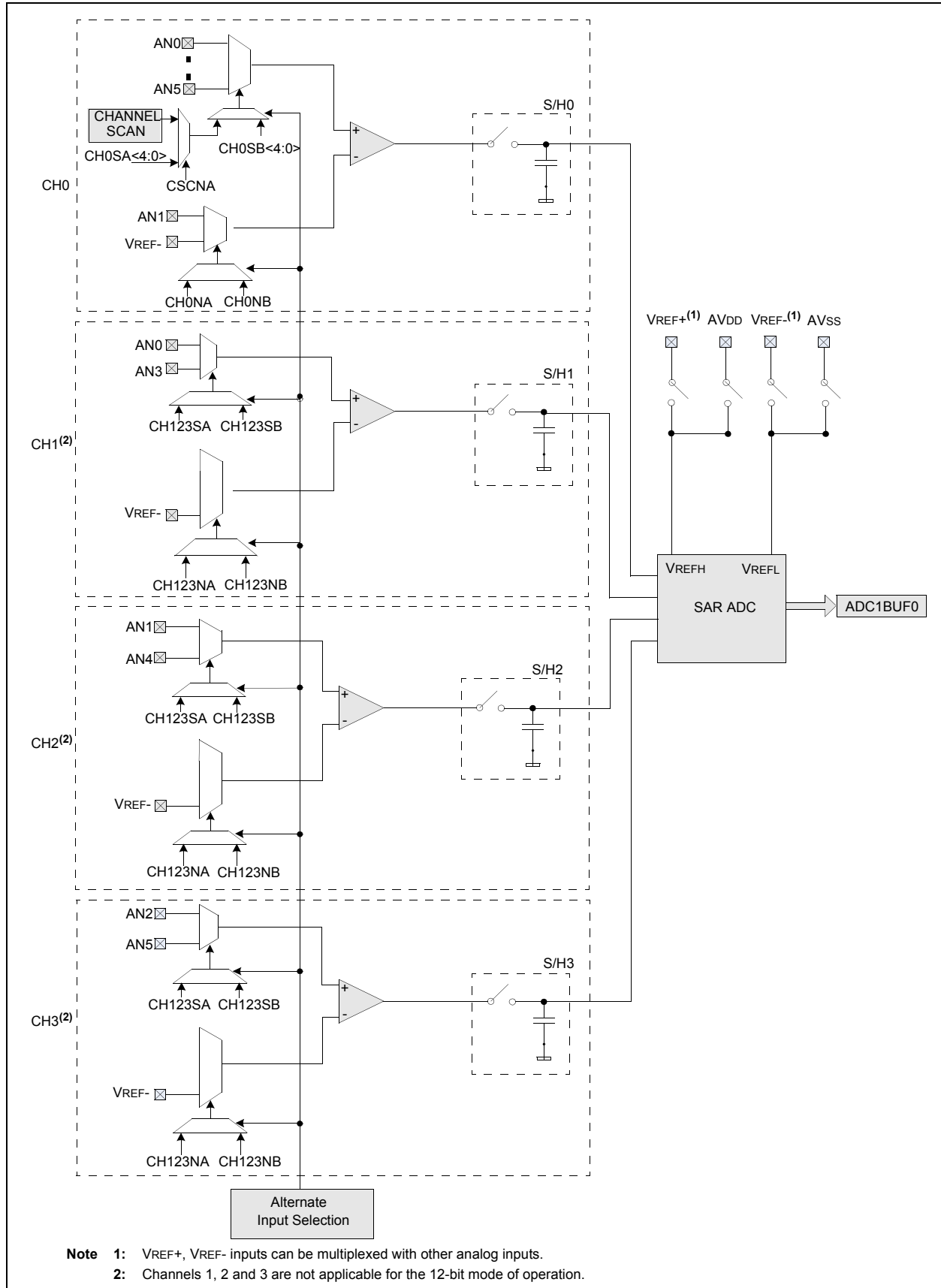
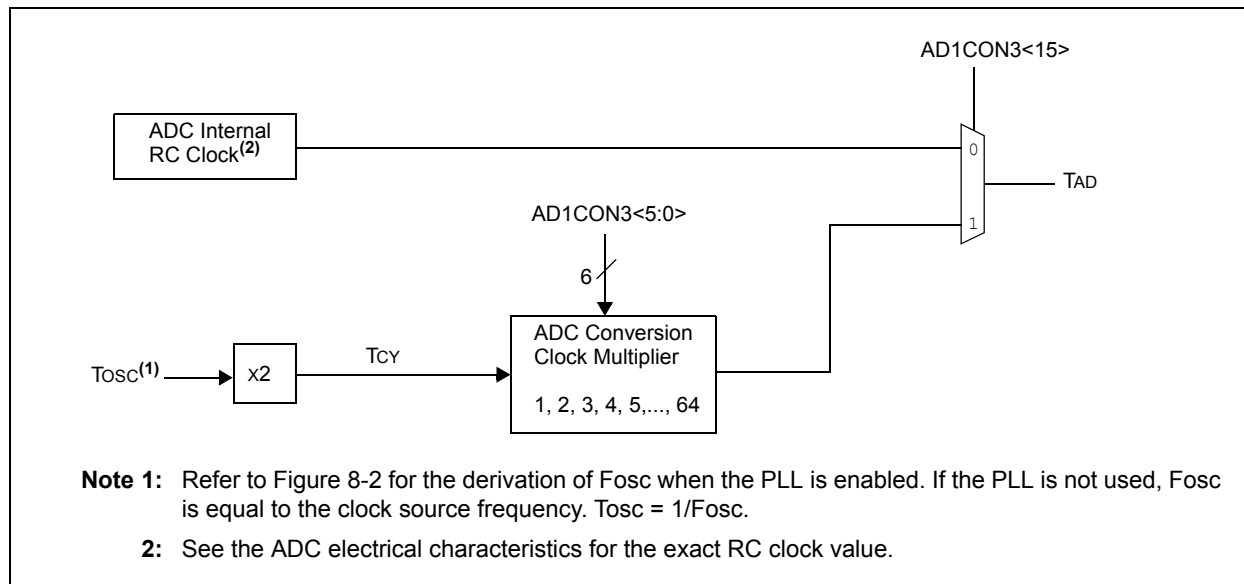


FIGURE 22-3: ADC CONVERSION CLOCK PERIOD BLOCK DIAGRAM



REGISTER 22-1: AD1CON1: ADC1 CONTROL REGISTER 1

| | | | | | | | |
|--------|-----|--------|---------|-----|-------|-----------|-------|
| R/W-0 | U-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
| ADON | — | ADSIDL | ADDMABM | — | AD12B | FORM<1:0> | |
| bit 15 | | | | | | bit 8 | |

| | | | | | | | |
|-----------|-------|-------|-----|--------|-------|----------------|-----------------|
| R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 HC,HS | R/C-0 HC, HS |
| SSRC<2:0> | | | — | SIMSAM | ASAM | SAMP | DONE |
| bit 7 | | | | | | bit 0 | |

| | | |
|-------------------|--------------------------|------------------------------------|
| Legend: | HC = Cleared by hardware | HS = Set by hardware |
| R = Readable bit | W = Writable 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:** ADC Operating Mode bit
 1 = ADC module is operating
 0 = ADC 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 **ADDMABM:** DMA Buffer Build Mode bit
 1 = DMA buffers are written in the order of conversion. The module provides an address to the DMA channel that is the same as the address used for the non-DMA stand-alone buffer.
 0 = DMA buffers are written in Scatter/Gather mode. The module provides a scatter/gather address to the DMA channel, based on the index of the analog input and the size of the DMA buffer.
- bit 11 **Unimplemented:** Read as '0'
- bit 10 **AD12B:** 10-bit or 12-bit Operation Mode bit
 1 = 12-bit, 1-channel ADC operation
 0 = 10-bit, 4-channel ADC operation
- bit 9-8 **FORM<1:0>:** Data Output Format bits
For 10-bit operation:
 11 = Signed fractional (DOUT = sddd dddd dd00 0000, where s = .NOT.d<9>)
 10 = Fractional (DOUT = dddd dddd dd00 0000)
 01 = Signed integer (DOUT = ssss sssd dddd dddd, where s = .NOT.d<9>)
 00 = Integer (DOUT = 0000 00dd dddd dddd)
For 12-bit operation:
 11 = Signed fractional (DOUT = sddd dddd dddd 0000, where s = .NOT.d<11>)
 10 = Fractional (DOUT = dddd dddd dddd 0000)
 01 = Signed Integer (DOUT = ssss sddd dddd dddd, where s = .NOT.d<11>)
 00 = Integer (DOUT = 0000 dddd dddd dddd)
- bit 7-5 **SSRC<2:0>:** Sample Clock Source Select bits
 111 = Internal counter ends sampling and starts conversion (auto-convert)
 110 = Reserved
 101 = Motor Control PWM2 interval ends sampling and starts conversion
 100 = GP timer (Timer5 for ADC1) compare ends sampling and starts conversion
 011 = Motor Control PWM1 interval ends sampling and starts conversion
 010 = GP timer (Timer3 for ADC1) compare ends sampling and starts conversion
 001 = Active transition on INT0 pin ends sampling and starts conversion
 000 = Clearing sample bit ends sampling and starts conversion
- bit 4 **Unimplemented:** Read as '0'

REGISTER 22-1: AD1CON1: ADC1 CONTROL REGISTER 1 (CONTINUED)

- bit 3 **SIMSAM:** Simultaneous Sample Select bit (only applicable when CHPS<1:0> = 01 or 1x)
When AD12B = 1, SIMSAM is: U-0, Unimplemented, Read as '0'
1 = Samples CH0, CH1, CH2, CH3 simultaneously (when CHPS<1:0> = 1x); or
 Samples CH0 and CH1 simultaneously (when CHPS<1:0> = 01)
0 = Samples multiple channels individually in sequence
- bit 2 **ASAM:** ADC Sample Auto-Start bit
1 = Sampling begins immediately after last conversion. SAMP bit is auto-set.
0 = Sampling begins when SAMP bit is set
- bit 1 **SAMP:** ADC Sample Enable bit
1 = ADC sample/hold amplifiers are sampling
0 = ADC sample/hold amplifiers are holding
If ASAM = 0, software can write '1' to begin sampling. Automatically set by hardware if ASAM = 1.
If SSRC = 000, software can write '0' to end sampling and start conversion. If SSRC ≠ 000,
automatically cleared by hardware to end sampling and start conversion.
- bit 0 **DONE:** ADC Conversion Status bit
1 = ADC conversion cycle is completed.
0 = ADC conversion not started or in progress
Automatically set by hardware when ADC conversion is complete. Software can write '0' to clear
DONE status (software not allowed to write '1'). Clearing this bit does NOT affect any operation in
progress. Automatically cleared by hardware at start of a new conversion.

REGISTER 22-2: AD1CON2: ADC1 CONTROL REGISTER 2

| | | | | | | | | |
|-----------|-------|-------|-----|-----|-------|-----------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | |
| VCFG<2:0> | | | — | — | CSCNA | CHPS<1:0> | | |
| 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 | — | SMPI<3:0> | | | | 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 **VCFG<2:0>**: Converter Voltage Reference Configuration bits

| | ADREF+ | ADREF- |
|-----|----------------|----------------|
| 000 | AVDD | AVSS |
| 001 | External VREF+ | AVSS |
| 010 | AVDD | External VREF- |
| 011 | External VREF+ | External VREF- |
| 1xx | AVDD | Avss |

bit 12-11 **Unimplemented**: Read as '0'

bit 10 **CSCNA**: Scan Input Selections for CH0+ during Sample A bit

- 1 = Scan inputs
- 0 = Do not scan inputs

bit 9-8 **CHPS<1:0>**: Selects Channels Utilized bits

When AD12B = 1, CHPS<1:0> is: U-0, Unimplemented, Read as '0'

- 1x =Converts CH0, CH1, CH2 and CH3
- 01 =Converts CH0 and CH1
- 00 =Converts CH0

bit 7 **BUFS**: Buffer Fill Status bit (only valid when BUFM = 1)

- 1 = ADC is currently filling buffer 0x8-0xF, user should access data in 0x0-0x7
- 0 = ADC is currently filling buffer 0x0-0x7, user should access data in 0x8-0xF

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

bit 5-2 **SMPI<3:0>**: Selects Increment Rate for DMA Addresses bits or number of sample/conversion operations per interrupt.

- 1111 =Increments the DMA address or generates interrupt after completion of every 16th sample/conversion operation
- 1110 =Increments the DMA address or generates interrupt after completion of every 15th sample/conversion operation

-
-
-

- 0001 =Increments the DMA address after completion of every 2nd sample/conversion operation
- 0000 =Increments the DMA address after completion of every sample/conversion operation

bit 1 **BUFM**: Buffer Fill Mode Select bit

- 1 = Starts buffer filling at address 0x0 on first interrupt and 0x8 on next interrupt
- 0 = Always starts filling buffer at address 0x0

bit 0 **ALTS**: Alternate Input Sample Mode Select bit

- 1 = Uses channel input selects for Sample A on first sample and Sample B on next sample
- 0 = Always uses channel input selects for Sample A

REGISTER 22-3: AD1CON3: ADC1 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 | — | — | SAMC<4:0> ⁽¹⁾ | | | | |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|--------------------------|-------|-------|-------|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| ADCS<7:0> ⁽²⁾ | | | | | | | |
| bit 7 | | | | | | | bit 0 |

Legend:

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

bit 15 **ADRC:** ADC Conversion Clock Source bit
 1 = ADC internal RC clock
 0 = Clock derived from system clock

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

bit 12-8 **SAMC<4:0>:** Auto Sample Time bits⁽¹⁾
 11111 = 31 TAD
 •
 •
 •
 00001 = 1 TAD
 00000 = 0 TAD

bit 7-0 **ADCS<7:0>:** ADC Conversion Clock Select bits⁽²⁾
 11111111 = Reserved
 •
 •
 •
 •
 01000000 = Reserved
 00111111 = $T_{CY} \cdot (ADCS<7:0> + 1) = 64 \cdot T_{CY} = TAD$
 •
 •
 •
 00000010 = $T_{CY} \cdot (ADCS<7:0> + 1) = 3 \cdot T_{CY} = TAD$
 00000001 = $T_{CY} \cdot (ADCS<7:0> + 1) = 2 \cdot T_{CY} = TAD$
 00000000 = $T_{CY} \cdot (ADCS<7:0> + 1) = 1 \cdot T_{CY} = TAD$

Note 1: This bit only used if AD1CON1<SSRC> = 1.
2: This bit is not used if AD1CON3<ADRC> = 1.

REGISTER 22-4: AD1CON4: ADC1 CONTROL REGISTER 4

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

| | | | | | | | |
|-------|-----|-----|-----|-----|------------|-------|-------|
| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | — | — | DMABL<2:0> | | |
| 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 15-3 **Unimplemented:** Read as '0'

bit 2-0 **DMABL<2:0>:** Selects Number of DMA Buffer Locations per Analog Input bits

- 111 = Allocates 128 words of buffer to each analog input
- 110 = Allocates 64 words of buffer to each analog input
- 101 = Allocates 32 words of buffer to each analog input
- 100 = Allocates 16 words of buffer to each analog input
- 011 = Allocates 8 words of buffer to each analog input
- 010 = Allocates 4 words of buffer to each analog input
- 001 = Allocates 2 words of buffer to each analog input
- 000 = Allocates 1 word of buffer to each analog input

REGISTER 22-5: AD1CHS123: ADC1 INPUT CHANNEL 1, 2, 3 SELECT REGISTER

| | | | | | | | |
|--------|-----|-----|-----|-----|--------------|-------|---------|
| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | — | — | CH123NB<1:0> | | CH123SB |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------|-----|-----|-----|-----|--------------|-------|---------|
| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
| — | — | — | — | — | CH123NA<1:0> | | CH123SA |
| bit 7 | | | | | | | bit 0 |

Legend:

R = Readable bit W = Writable 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 10-9 **CH123NB<1:0>:** Channel 1, 2, 3 Negative Input Select for Sample B bits
dsPIC33FJ32MC302, dsPIC33FJ64MC202/802, and dsPIC33FJ128MC202/802 devices only:
If AD12B = 1:
 11 = Reserved
 10 = Reserved
 01 = Reserved
 00 = Reserved

If AD12B = 0:
 11 = Reserved
 10 = Reserved
 01 = CH1, CH2, CH3 negative input is VREF-
 00 = CH1, CH2, CH3 negative input is VREF-

dsPIC33FJ32MC304, dsPIC33FJ64MC204/804, and dsPIC33FJ128MC204/804 devices only:
If AD12B = 1:
 11 = Reserved
 10 = Reserved
 01 = Reserved
 00 = Reserved

If AD12B = 0:
 11 = Reserved
 10 = CH1 negative input is AN6, CH2 negative input is AN7, CH3 negative input is AN8
 01 = CH1, CH2, CH3 negative input is VREF-
 00 = CH1, CH2, CH3 negative input is VREF-

bit 8 **CH123SB:** Channel 1, 2, 3 Positive Input Select for Sample B bit
If AD12B = 1:
 1 = Reserved
 0 = Reserved

If AD12B = 0:
 1 = CH1 positive input is AN3, CH2 positive input is AN4, CH3 positive input is AN5
 0 = CH1 positive input is AN0, CH2 positive input is AN1, CH3 positive input is AN2

bit 7-3 **Unimplemented:** Read as '0'

REGISTER 22-5: AD1CHS123: ADC1 INPUT CHANNEL 1, 2, 3 SELECT REGISTER (CONTINUED)

bit 2-1 **CH123NA<1:0>**: Channel 1, 2, 3 Negative Input Select for Sample A bits
dsPIC33FJ32MC302, dsPIC33FJ64MC202/802, and dsPIC33FJ128MC202/802 devices only:
If AD12B = 1:
11 = Reserved
10 = Reserved
01 = Reserved
00 = Reserved

If AD12B = 0:
11 = Reserved
10 = Reserved
01 = CH1, CH2, CH3 negative input is VREF-
00 = CH1, CH2, CH3 negative input is VREF-

dsPIC33FJ32MC304, dsPIC33FJ64MC204/804, and dsPIC33FJ128MC204/804 devices only:
If AD12B = 1:
11 = Reserved
10 = Reserved
01 = Reserved
00 = Reserved

If AD12B = 0:
11 = Reserved
10 = CH1 negative input is AN6, CH2 negative input is AN7, CH3 negative input is AN8
01 = CH1, CH2, CH3 negative input is VREF-
00 = CH1, CH2, CH3 negative input is VREF-

bit 0 **CH123SA**: Channel 1, 2, 3 Positive Input Select for Sample A bit
If AD12B = 1:
1 = Reserved
0 = Reserved

If AD12B = 0:
1 = CH1 positive input is AN3, CH2 positive input is AN4, CH3 positive input is AN5
0 = CH1 positive input is AN0, CH2 positive input is AN1, CH3 positive input is AN2

REGISTER 22-6: AD1CHS0: ADC1 INPUT CHANNEL 0 SELECT REGISTER

| | | | | | | | |
|--------|-----|-----|------------|-------|-------|-------|-------|
| R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| CH0NB | — | — | CH0SB<4:0> | | | | |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------|-----|-----|------------|-------|-------|-------|-------|
| R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| CH0NA | — | — | CH0SA<4:0> | | | | |
| 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 **CH0NB:** Channel 0 Negative Input Select for Sample B bit
 1 = Channel 0 negative input is AN1
 0 = Channel 0 negative input is VREF-
- bit 14-13 **Unimplemented:** Read as '0'
- bit 12-8 **CH0SB<4:0>:** Channel 0 Positive Input Select for Sample B bits
 dsPIC33FJ32MC304, dsPIC33FJ64MC204/804, and dsPIC33FJ128MC204/804 devices only:
 01000 = Channel 0 positive input is AN8
 •
 •
 •
 00010 = Channel 0 positive input is AN2
 00001 = Channel 0 positive input is AN1
 00000 = Channel 0 positive input is AN0
 dsPIC33FJ32MC302, dsPIC33FJ64MC202/802, and dsPIC33FJ128MC202/802 devices only:
 00101 = Channel 0 positive input is AN5
 •
 •
 •
 00010 = Channel 0 positive input is AN2
 00001 = Channel 0 positive input is AN1
 00000 = Channel 0 positive input is AN0.
- bit 7 **CH0NA:** Channel 0 Negative Input Select for Sample A bit
 1 = Channel 0 negative input is AN1
 0 = Channel 0 negative input is VREF-
- bit 6-5 **Unimplemented:** Read as '0'

REGISTER 22-6: AD1CHS0: ADC1 INPUT CHANNEL 0 SELECT REGISTER (CONTINUED)

bit 4-0

CH0SA<4:0>: Channel 0 Positive Input Select for Sample A bits

dsPIC33FJ32MC304, dsPIC33FJ64MC204/804, and dsPIC33FJ128MC204/804 devices only:

01000 = Channel 0 positive input is AN8

•

•

•

00010 = Channel 0 positive input is AN2

00001 = Channel 0 positive input is AN1

00000 = Channel 0 positive input is AN0

dsPIC33FJ32MC302, dsPIC33FJ64MC202/802, and dsPIC33FJ128MC202/802 devices only:

00101 = Channel 0 positive input is AN5

•

•

•

00010 = Channel 0 positive input is AN2

00001 = Channel 0 positive input is AN1

00000 = Channel 0 positive input is AN0

REGISTER 22-7: AD1CSSL: ADC1 INPUT SCAN SELECT REGISTER LOW^(1,2)

| | | | | | | | |
|--------|-----|-----|-----|-----|-----|-----|-------|
| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 |
| — | — | — | — | — | — | — | CSS8 |
| 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 |
| CSS7 | CSS6 | CSS5 | CSS4 | CSS3 | CSS2 | CSS1 | CSS0 |
| bit 7 | | | | | | | bit 0 |

Legend:

R = Readable bit W = Writable 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 8-0 **CSS<8:0>:** ADC Input Scan Selection bits
 1 = Select ANx for input scan
 0 = Skip ANx for input scan

- Note 1:** On devices without nine analog inputs, all AD1CSSL bits can be selected by user application. However, inputs selected for scan without a corresponding input on device converts VREFL.
2: CSSx = ANx, where x = 0 through 8.

REGISTER 22-8: AD1PCFGL: ADC1 PORT CONFIGURATION REGISTER LOW^(1,2, 3)

| | | | | | | | |
|--------|-----|-----|-----|-----|-----|-----|-------|
| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 |
| — | — | — | — | — | — | — | PCFG8 |
| 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 | PCFG6 | PCFG5 | PCFG4 | PCFG3 | PCFG2 | PCFG1 | PCFG0 |
| bit 7 | | | | | | | bit 0 |

Legend:

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

bit 15-9 **Unimplemented:** Read as '0'
 bit 8-0 **PCFG<8:0>:** ADC Port Configuration Control bits
 1 = Port pin in Digital mode, port read input enabled, ADC input multiplexor connected to AVSS
 0 = Port pin in Analog mode, port read input disabled, ADC samples pin voltage

- Note 1:** On devices without nine analog inputs, all PCFG bits are R/W by user. However, PCFG bits are ignored on ports without a corresponding input on device.
2: PCFGx = ANx, where x = 0 through 8.
3: PCFGx bits have no effect if ADC module is disabled by setting ADxMD bit in the PMDx register. In this case, all port pins are multiplexed with ANx will be in Digital mode.

23.0 AUDIO DIGITAL-TO-ANALOG CONVERTER (DAC)

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the *dsPIC33F Family Reference Manual, Section 33. “Audio Digital-to-Analog Converter (DAC)”* (DS70211), which is available from the Microchip website (www.microchip.com).

The Audio Digital-to-Analog Converter (DAC) module is a 16-bit Delta-Sigma signal converter designed for audio applications. It has two output channels, left and right to support stereo applications. Each DAC output channel provides three voltage outputs, positive DAC output, negative DAC output, and the midpoint voltage output for the dsPIC33FJ64MC804 and dsPIC33FJ128MC804 devices. The dsPIC33FJ64MC802 and dsPIC33FJ128MC802 devices provide positive DAC output and negative DAC output voltages. The positive and negative DAC outputs are differential about a midpoint voltage of approximately 0.884 volts to drive the speakers with a Bridge-Tied Load (BTL) configuration.

23.1 KEY FEATURES

- 16-bit resolution (14-bit accuracy)
- Second-Order Digital Delta-Sigma Modulator
- 256 X Over-Sampling Ratio
- 128-Tap FIR Current-Steering Analog Reconstruction Filter
- 100 KSPS Maximum Sampling Rate
- User controllable Sample Clock
- Input Frequency 45 kHz max
- Differential Analog Outputs
- Signal-To-Noise: 90 dB
- 4-deep input Buffer
- 16-bit Processor I/O, and DMA interfaces

23.2 DAC Module Operation

The functional block diagram of the Audio DAC module is shown in Figure 23-1. The Audio DAC module provides a 4-deep data input FIFO buffer for each output channel. If the DMA module and/or the processor cannot provide output data in a timely manner, and the FIFO becomes empty, the DAC accepts data from the DAC Default Data register (DACDFLT). This safety feature is useful for industrial control applications where the DAC output controls an important processor or machinery. The DACDFLT

register should be initialized with a “safe” output value. Often the safe output value is either the midpoint value (0x8000) or a zero value (0x0000).

The digital interpolator up-samples the input signals, where the over-sampling ratio is 256x which creates data points between the user supplied data points. The interpolator also includes processing by digital filters to provide “noise shaping” to move the converter noise above 20 kHz (upper limit of the pass band). The output of the interpolator drives the Sigma-Delta modulator. The serial data bit stream from the Sigma-Delta modulator is processed by the reconstruction filter. The differential outputs of the reconstruction filter are amplified by Op Amps to provide the required 2 volts peak-to-peak voltage swing into a 1 kOhm load.

23.3 DAC Output Format

The DAC output data stream can be in a two’s complement signed number format or as an unsigned number format.

The Audio DAC module features the ability to accept the 16-bit input data in a two’s complement signed number format or as an unsigned number format. The data formatting is controlled by the Data Format Control (FORM<8>) bit in the DAC1CON register. The supported formats are:

- 1 = Signed (two’s complement)
- 0 = Unsigned

If the FORM bit is configured for “Unsigned data” then the user input data yields the following behavior:

- 0xFFFF = most positive output voltage
- 0x8000 = mid point output voltage
- 0x7FFF = a value just below the midpoint
- 0x0000 = minimum output voltage

If the FORM bit is configured for “signed data” then the user input data yields the following behavior:

- 0x7FFF = most positive output voltage
- 0x0000 = mid point output voltage
- 0xFFFF = value just below the midpoint
- 0x8000 = minimum output voltage

The Audio DAC provides an analog output proportional to the digital input value. The maximum 100,000 samples per second (100ksp/s) update rate provides good quality audio reproduction.

23.4 DAC CLOCK

The DAC clock signal clocks the internal logic of the Audio DAC module. The data sample rate of the Audio DAC is an integer division of the rate of the DAC clock. The DAC clock is generated via a clock divider circuit that accepts an auxiliary clock from the auxiliary oscillator. The divisor ratio is programmed by clock divider

bits (DACFDIV<6:0>) in the DAC Control register (DAC1CON). The resulting DAC clock must not exceed 25.6 MHz. If lower sample rates are to be used, then the DAC filter clock frequency may be reduced to reduce power consumption. The DAC clock frequency is 256 times the sampling frequency.

FIGURE 23-1: BLOCK DIAGRAM OF AUDIO DIGITAL-TO-ANALOG (DAC) CONVERTER

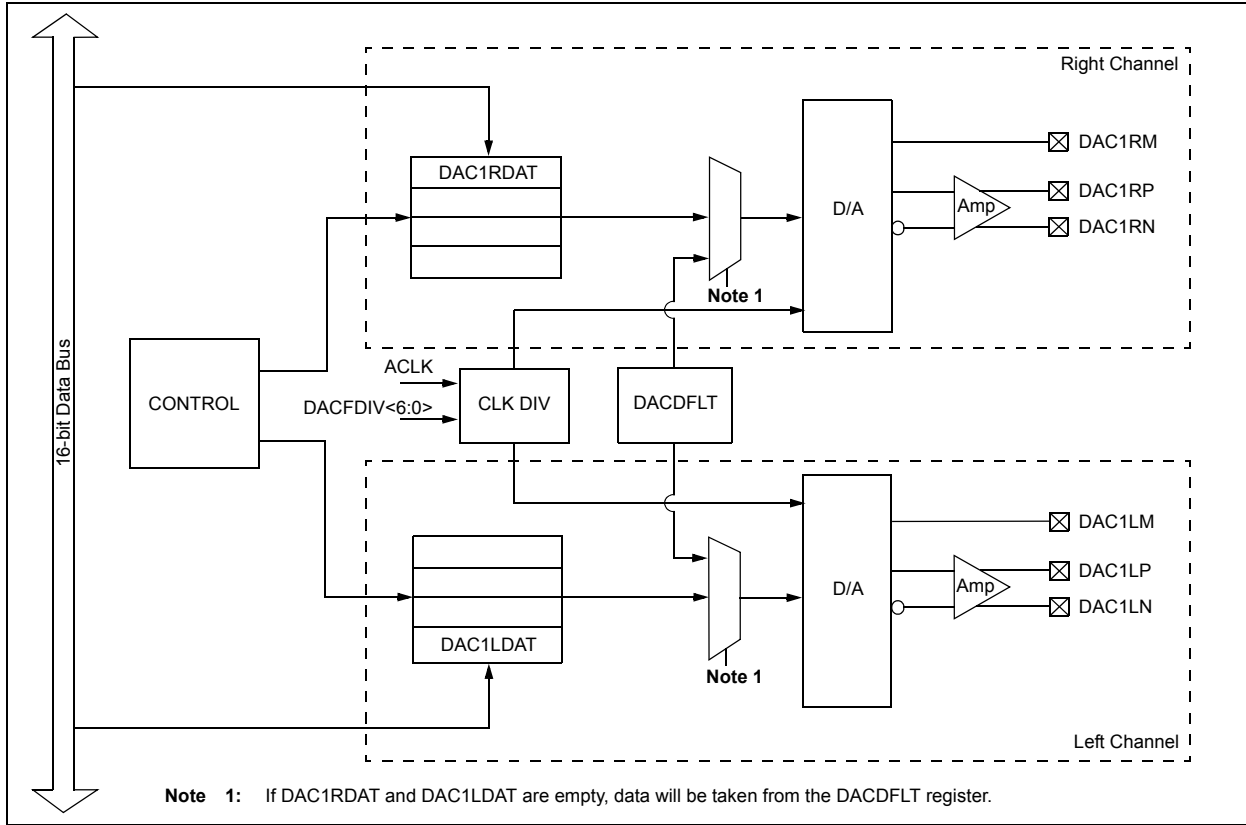
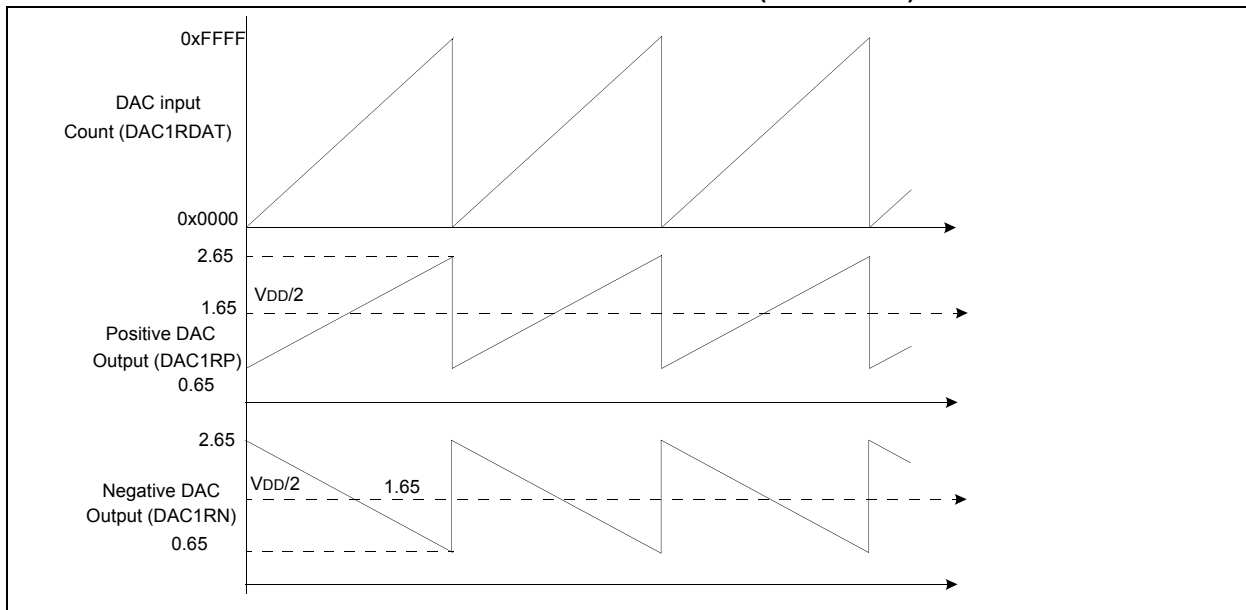


FIGURE 23-2: AUDIO DAC OUTPUT FOR RAMP INPUT (UNSIGNED)



REGISTER 23-1: DAC1CON: DAC CONTROL REGISTER

| | | | | | | | |
|--------|--------------|---------|-------|-------|-------|-------|-------|
| R/W-0 | U-0 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | R/W-0 |
| DACEN | — | DACSIDL | AMPON | — | — | — | FORM |
| bit 15 | | | | | | | bit 8 |
| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-1 | R/W-0 | R/W-1 |
| — | DACFDIV<6:0> | | | | | | |
| 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 **DACEN:** DAC1 Enable bit
1 = Enables module
0 = Disables module
- bit 14 **Unimplemented:** Read as '0'
- bit 13 **DACSIDL:** Stop in Ideal Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode.
- bit 12 **AMPON:** Enable Analog Output Amplifier in Sleep Mode/Stop-in Idle Mode
1 = Analog Output Amplifier is enabled during Sleep Mode/Stop-in Idle mode
0 = Analog Output Amplifier is disabled during Sleep Mode/Stop-in Idle mode
- bit 11-9 **Unimplemented:** Read as '0'
- bit 8 **FORM:** Data Format Select bit
1 = Signed integer
0 = Unsigned integer
- bit 7 **Unimplemented:** Read as '0'
- bit 6-0 **DACFDIV<6:0>:** DAC Clock Divider.
1111111 = Divide input clock by 128
•
•
•
0000101 = Divide input clock by 6 (default)
•
•
•
0000010 = Divide input clock by 3
0000001 = Divide input clock by 2
0000000 = Divide input clock by 1 (no divide)

REGISTER 23-2: DAC1STAT: DAC STATUS REGISTER

| | | | | | | | |
|--------|-----|--------|-----|-----|--------|-------|-------|
| R/W-0 | U-0 | R/W-0 | U-0 | U-0 | R/W-0 | R-0 | R-0 |
| LOEN | — | LMVOEN | — | — | LITYPE | LFULL | LEMPY |
| bit 15 | | | | | | bit 8 | |

| | | | | | | | |
|-------|-----|--------|-----|-----|--------|-------|-------|
| R/W-0 | U-0 | R/W-0 | U-0 | U-0 | R/W-0 | R-0 | R-0 |
| ROEN | — | RMVOEN | — | — | RITYPE | RFULL | REMPY |
| bit 7 | | | | | | bit 0 | |

Legend:

| | | |
|-------------------|------------------|--|
| R = Readable bit | W = Writable 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 **LOEN:** Left Channel DAC output enable
1 = Positive and negative DAC outputs are enabled.
0 = DAC outputs are disabled.
- bit 14 **Unimplemented:** Read as '0'
- bit 13 **LMVOEN:** Left Channel Midpoint DAC output voltage enable
1 = Midpoint DAC output is enabled.
0 = Midpoint output is disabled.
- bit 12-11 **Unimplemented:** Read as '0'
- bit 10 **LITYPE:** Left Channel Type of Interrupt
1 = Interrupt if FIFO is EMPTY.
0 = Interrupt if FIFO is NOT FULL.
- bit 9 **LFULL:** Status, Left Channel Data input FIFO is FULL
1 = FIFO is Full.
0 = FIFO is not full.
- bit 8 **LEMPY:** Status, Left Channel Data input FIFO is EMPTY
1 = FIFO is Empty.
0 = FIFO is not Empty.
- bit 7 **ROEN:** Right Channel DAC output enable
1 = Positive and negative DAC outputs are enabled.
0 = DAC outputs are disabled.
- bit 6 **Unimplemented:** Read as '0'
- bit 5 **RMVOEN:** Right Channel Midpoint DAC output voltage enable
1 = Midpoint DAC output is enabled.
0 = Midpoint output is disabled.
- bit 4-3 **Unimplemented:** Read as '0'
- bit 2 **RITYPE:** Right Channel Type of Interrupt
1 = Interrupt if FIFO is EMPTY.
0 = Interrupt if FIFO is NOT FULL.
- bit 1 **RFULL:** Status, Right Channel Data input FIFO is FULL
1 = FIFO is Full.
0 = FIFO is not full.
- bit 0 **REMPY:** Status, Right Channel Data input FIFO is EMPTY
1 = FIFO is Empty.
0 = FIFO is not Empty.

REGISTER 23-3: DAC1DFLT: DAC DEFAULT DATA 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 |
| DAC1DFLT<15:8> | | | | | | | |
| 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 |
| DAC1DFLT<7:0> | | | | | | | |
| 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 15-0 **DACDFLT:** DAC Default Value

REGISTER 23-4: DAC1LDAT: DAC LEFT DATA 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 |
| DAC1LDAT<15:8> | | | | | | | |
| 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 |
| DAC1LDAT<7:0> | | | | | | | |
| 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 15-0 **DACLDAT:** Left Channel Data Port.

REGISTER 23-5: DAC1RDAT: DAC RIGHT DATA 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 |
| DAC1RDAT<15:8> | | | | | | | |
| 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 |
| DAC1RDAT<7:0> | | | | | | | |
| 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 15-0 **DACRDAT:** Right Channel Data Port.

NOTES:

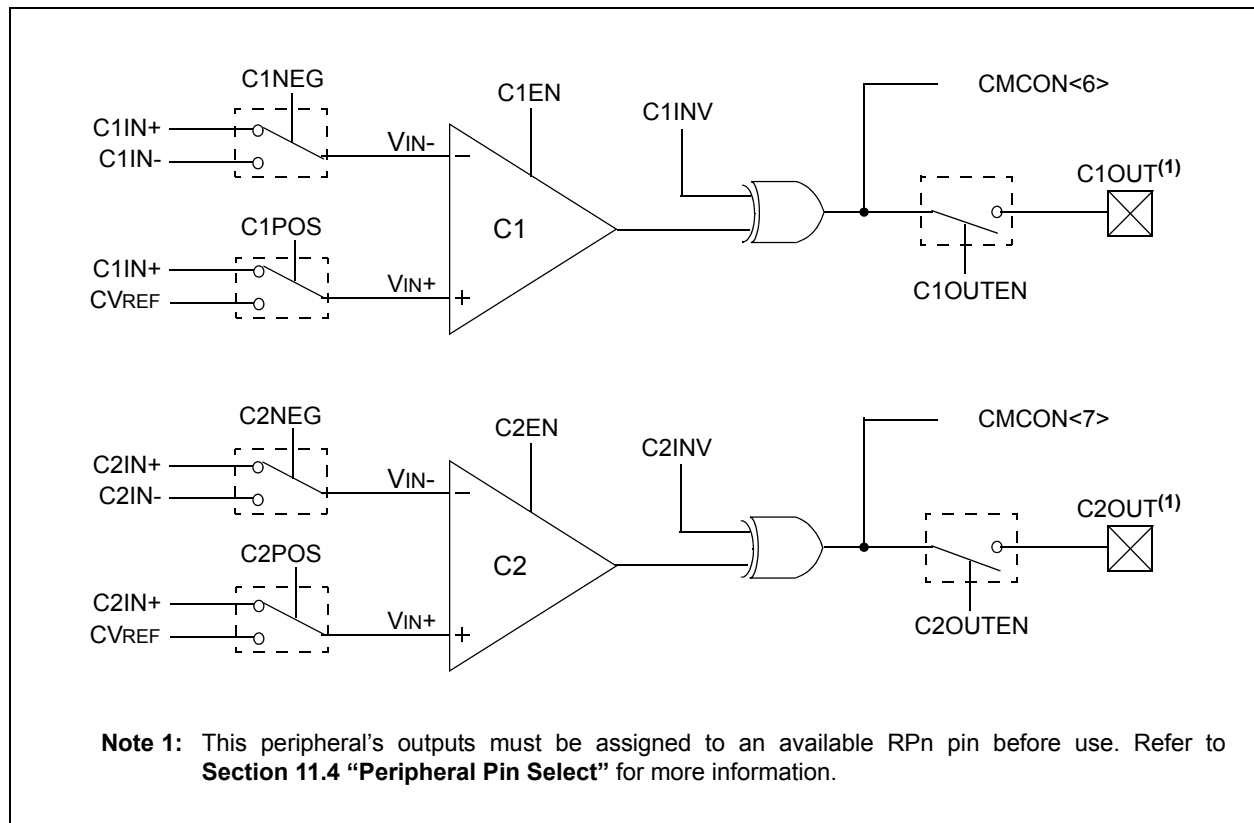
24.0 COMPARATOR MODULE

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 34. “Comparator”, which is available from the Microchip website (www.microchip.com).

The Comparator module provides a set of dual input comparators. The inputs to the comparator can be configured to use any one of the four pin inputs (C1IN+, C1IN-, C2IN+ and C2IN-) as well as the Comparator Voltage Reference Input (CVREF).

Note: This peripheral contains output functions that may need to be configured by the peripheral pin select feature. For more information, see Section 11.4 “Peripheral Pin Select”

FIGURE 24-1: COMPARATOR I/O OPERATING MODES



REGISTER 24-1: CMCON: COMPARATOR 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 |
| CMIDL | — | C2EVT | C1EVT | C2EN | C1EN | C2OUTEN ⁽¹⁾ | C1OUTEN ⁽²⁾ |
| 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
 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⁽¹⁾
 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⁽²⁾
 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-

- Note 1:** If C2OUTEN = 1, the C2OUT peripheral output must be configured to an available RPx pin. See Section 11.4 “Peripheral Pin Select” for more information.
- 2:** If C1OUTEN = 1, the C1OUT peripheral output must be configured to an available RPx pin. See Section 11.4 “Peripheral Pin Select” for more information.

REGISTER 24-1: CMCON: COMPARATOR CONTROL REGISTER (CONTINUED)

| | |
|-------|---|
| bit 6 | C1OUT: Comparator 1 Output bit <u>When C1INV = 0:</u> 1 = C1 VIN+ > C1 VIN- 0 = C1 VIN+ < C1 VIN- <u>When C1INV = 1:</u> 0 = C1 VIN+ > C1 VIN- 1 = C1 VIN+ < C1 VIN- |
| bit 5 | C2INV: Comparator 2 Output Inversion bit 1 = C2 output inverted 0 = C2 output not inverted |
| bit 4 | C1INV: Comparator 1 Output Inversion bit 1 = C1 output inverted 0 = C1 output not inverted |
| bit 3 | C2NEG: Comparator 2 Negative Input Configure bit 1 = Input is connected to VIN+ 0 = Input is connected to VIN- See Figure 24-1 for the comparator modes. |
| bit 2 | C2POS: Comparator 2 Positive Input Configure bit 1 = Input is connected to VIN+ 0 = Input is connected to CVREF See Figure 24-1 for the comparator modes. |
| bit 1 | C1NEG: Comparator 1 Negative Input Configure bit 1 = Input is connected to VIN+ 0 = Input is connected to VIN- See Figure 24-1 for the comparator modes. |
| bit 0 | C1POS: Comparator 1 Positive Input Configure bit 1 = Input is connected to VIN+ 0 = Input is connected to CVREF See Figure 24-1 for the comparator modes. |

Note 1: If C2OUTEN = 1, the C2OUT peripheral output must be configured to an available RPx pin. See **Section 11.4 “Peripheral Pin Select”** for more information.

2: If C1OUTEN = 1, the C1OUT peripheral output must be configured to an available RPx pin. See **Section 11.4 “Peripheral Pin Select”** for more information.

24.1 Comparator Voltage Reference

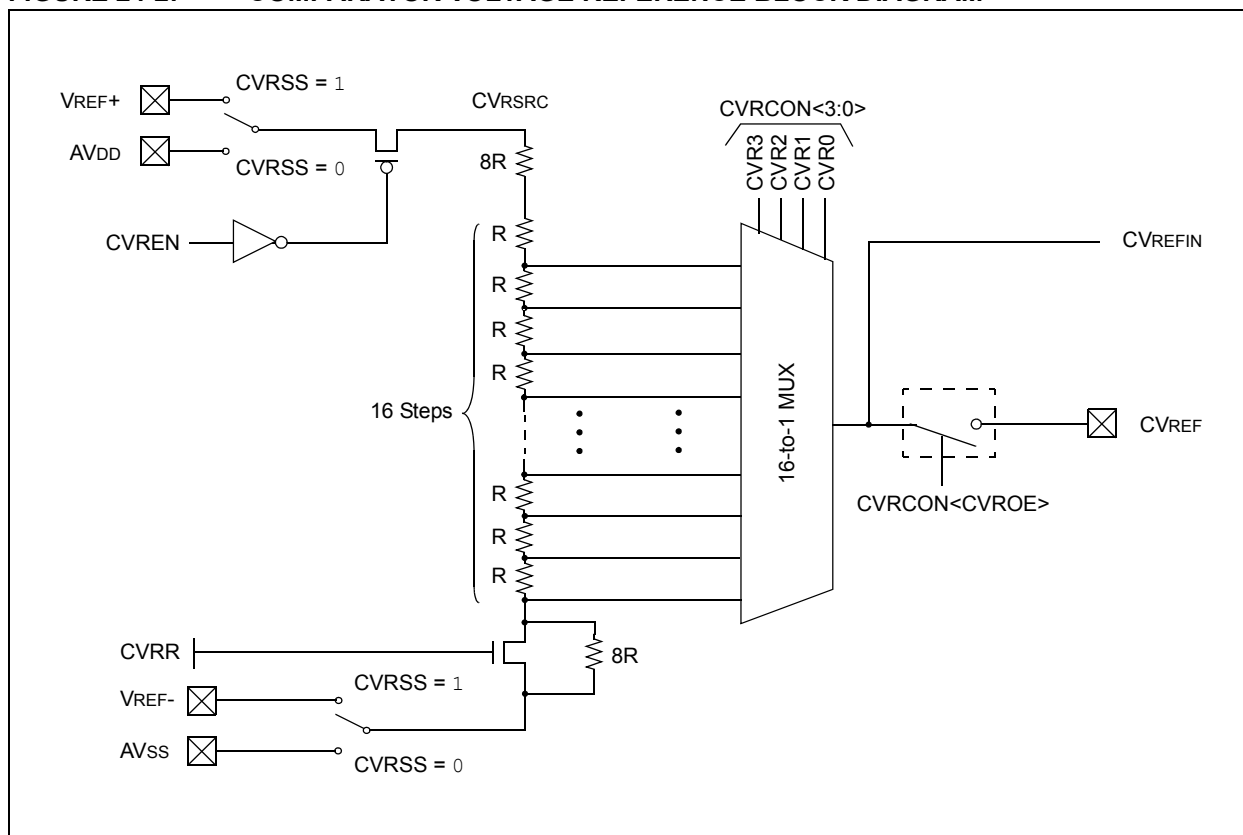
24.1.1 CONFIGURING THE COMPARATOR VOLTAGE REFERENCE

The voltage reference module is controlled through the CVRCON register (Register 24-2). 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 24-2: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM



REGISTER 24-2: 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 | CVR<3:0> | | | |
| 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 15-8 **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 **CVR<3:0>:** Comparator VREF Value Selection $0 \leq \text{CVR<3:0>} \leq 15$ bits
 When CVRR = 1:

$$\text{CVREF} = (\text{CVR<3:0>} / 24) \cdot (\text{CVRSRC})$$
 When CVRR = 0:

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

NOTES:

25.0 REAL-TIME CLOCK AND CALENDAR (RTCC)

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 37. “Real-Time Clock and Calendar (RTCC)”, which is available from the Microchip website (www.microchip.com).

This chapter discusses the Real-Time Clock and Calendar (RTCC) module, available on dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices, and its operation. Listed below are some of the key features of this module:

- Time: hours, minutes, and seconds
- 24-hour format (military time)
- Calendar: weekday, date, month, and year

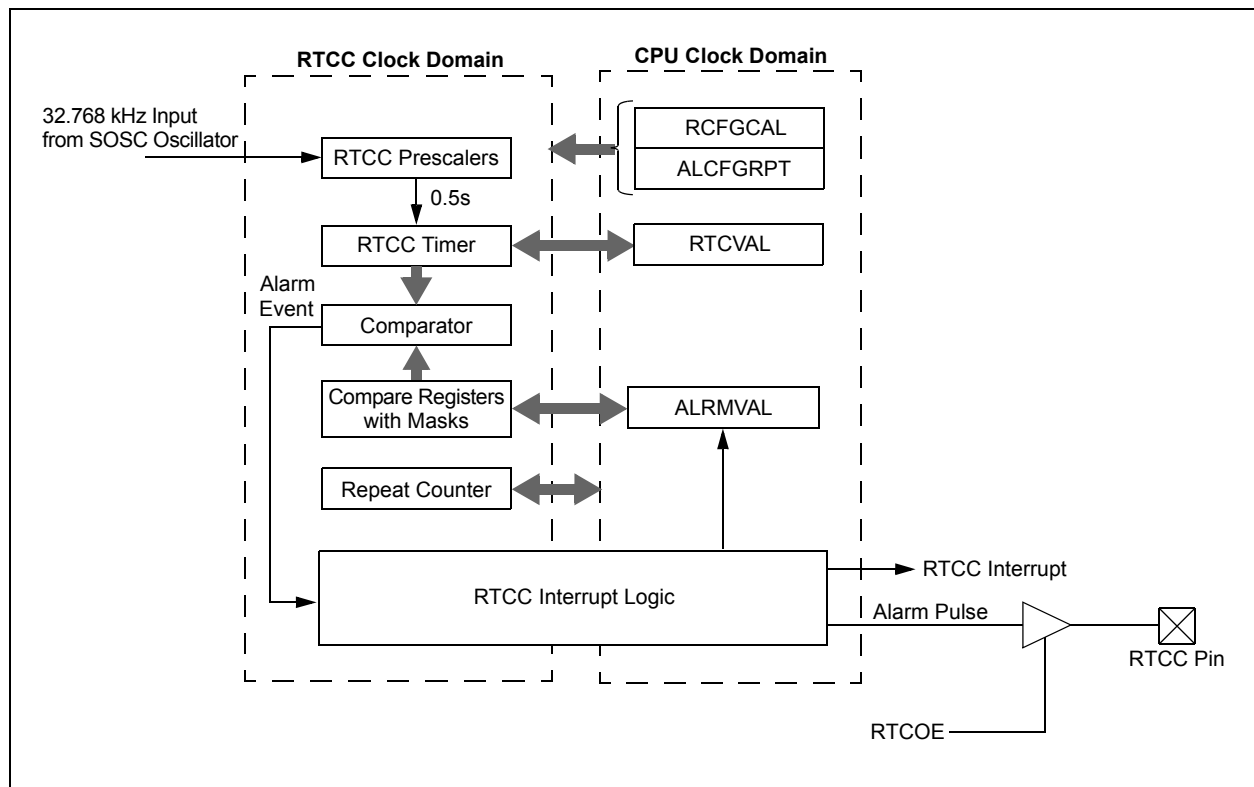
- Alarm configurable
- Year range: 2000 to 2099
- Leap year correction
- BCD format for compact firmware
- Optimized for low-power operation
- User calibration with auto-adjust
- Calibration range: ± 2.64 seconds error per month
- Requirements: External 32.768 kHz clock crystal
- Alarm pulse or seconds clock output on RTCC pin

The RTCC module is intended for applications where accurate time must be maintained for extended periods of time with minimum to no intervention from the CPU. The RTCC module is optimized for low-power usage to provide extended battery lifetime while keeping track of time.

The RTCC module is a 100-year clock and calendar with automatic leap year detection. The range of the clock is from 00:00:00 (midnight) on January 1, 2000 to 23:59:59 on December 31, 2099.

The hours are available in 24-hour (military time) format. The clock provides a granularity of one second with half-second visibility to the user.

FIGURE 25-1: RTCC BLOCK DIAGRAM



25.1 RTCC Module Registers

The RTCC module registers are organized into three categories:

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

25.1.1 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 25-1).

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 25-1: RTCVAL REGISTER MAPPING

| RTCPTR <1:0> | RTCC Value Register Window | |
|-----------------|----------------------------|-------------|
| | RTCVAL<15:8> | RTCVAL<7:0> |
| 00 | MINUTES | SECONDS |
| 01 | WEEKDAY | HOURS |
| 10 | MONTH | DAY |
| 11 | — | YEAR |

The Alarm Value register window (ALRMVALH and ALRMVALL) uses the ALRMPTR bits (ALCFGRPT<9:8>) to select the desired Alarm register pair (see Table 25-2).

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.

TABLE 25-2: ALRMVAL REGISTER MAPPING

| ALRMPTR <1:0> | Alarm Value Register Window | |
|------------------|-----------------------------|--------------|
| | ALRMVAL<15:8> | ALRMVAL<7:0> |
| 00 | ALRMMIN | ALRMSEC |
| 01 | ALRMWD | ALRMHR |
| 10 | ALRMMNTH | ALRMDAY |
| 11 | — | — |

Considering that the 16-bit core does not distinguish between 8-bit and 16-bit read operations, the user must be aware that when reading either the ALRMVALH or ALRMVALL bytes 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.

25.1.2 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 25-1).

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 25-1.

EXAMPLE 25-1: SETTING THE RTCWREN BIT

```

MOV    #NVMKEY, W1      ;move the address of NVMKEY into W1
MOV    #0x55, W2
MOV    #0xAA, W3
MOV    W2, [W1]         ;start 55/AA sequence
MOV    W3, [W1]
BSET   RCFGCAL, #13    ;set the RTCWREN bit
    
```

REGISTER 25-1: RCFGAL: RTCC CALIBRATION AND CONFIGURATION REGISTER⁽¹⁾

| | | | | | | | |
|----------------------|-----|---------|---------|------------------------|-------|-------------|-------|
| R/W-0 | U-0 | R/W-0 | R-0 | R-0 | R/W-0 | R/W-0 | R/W-0 |
| RTCEN ⁽²⁾ | — | RTCWREN | RTCSYNC | HALFSEC ⁽³⁾ | RTCOE | RTCPTR<1:0> | |
| 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 |
| CAL<7:0> | | | | | | | |
| 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 **RTCEN:** RTCC Enable bit⁽²⁾
 1 = RTCC module is enabled
 0 = RTCC module is disabled
- bit 14 **Unimplemented:** Read as '0'
- bit 13 **RTCWREN:** RTCC Value Registers Write Enable bit
 1 = RTCVALH and RTCVALL registers can be written to by the user
 0 = RTCVALH and RTCVALL registers are locked out from being written to by the user
- bit 12 **RTCSYNC:** RTCC Value Registers Read Synchronization bit
 1 = RTCVALH, RTCVALL and ALCFGRPT registers can change while reading due to a rollover ripple resulting in an invalid data read. If the register is read twice and results in the same data, the data can be assumed to be valid.
 0 = RTCVALH, RTCVALL or ALCFGRPT registers can be read without concern over a rollover ripple
- bit 11 **HALFSEC:** Half-Second Status bit⁽³⁾
 1 = Second half period of a second
 0 = First half period of a second
- bit 10 **RTCOE:** RTCC Output Enable bit
 1 = RTCC output enabled
 0 = RTCC output disabled
- bit 9-8 **RTCPTR<1:0>:** RTCC Value Register Window Pointer bits
 Points to the corresponding RTCC Value registers when reading 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.
- 2:** A write to the RTCEN bit is only allowed when RTCWREN = 1.
- 3:** This bit is read-only. It is cleared to '0' on a write to the lower half of the MINSEC register.

REGISTER 25-1: RCFGAL: RTCC CALIBRATION AND CONFIGURATION REGISTER⁽¹⁾ (CONTINUED)

bit 7-0 **CAL<7:0>**: RTC Drift Calibration bits

01111111 =Maximum positive adjustment; adds 508 RTC clock pulses every one minute

•

•

•

00000001 =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.
- 2:** A write to the RTCEN bit is only allowed when RTCWREN = 1.
- 3:** This bit is read-only. It is cleared to '0' on a write to the lower half of the MINSEC register.

REGISTER 25-2: 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 ⁽¹⁾ | PMP TTL |
| bit 7 | | | | | | bit 0 | |

| | | | |
|-------------------|------------------|------------------------------------|--------------------|
| Legend: | | | |
| R = Readable bit | W = Writable 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⁽¹⁾
 - 1 = RTCC seconds clock is selected for the RTCC pin
 - 0 = RTCC alarm pulse is selected for the RTCC pin
- bit 0 **PMP TTL:** 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.

REGISTER 25-3: 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 | AMASK<3:0> | | | | ALRMPTR<1:0> | |
| 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 |
| ARPT<7:0> | | | | | | | |
| 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 **ALRMEN:** Alarm Enable bit
 1 = Alarm is enabled (cleared automatically after an alarm event whenever ARPT<7:0> = 00h and CHIME = 0)
 0 = Alarm is disabled
- bit 14 **CHIME:** Chime Enable bit
 1 = Chime is enabled; ARPT<7:0> bits are allowed to roll over from 00h to FFh
 0 = Chime is disabled; ARPT<7:0> bits stop once they reach 00h
- bit 13-10 **AMASK<3:0>:** Alarm Mask Configuration bits
 0000 = Every half second
 0001 = Every second
 0010 = Every 10 seconds
 0011 = Every minute
 0100 = Every 10 minutes
 0101 = Every hour
 0110 = Once a day
 0111 = Once a week
 1000 = Once a month
 1001 = Once a year (except when configured for February 29th, once every 4 years)
 101x = Reserved – do not use
 11xx = Reserved – do not use
- bit 9-8 **ALRMPTR<1:0>:** Alarm Value Register Window Pointer bits
 Points to the corresponding Alarm Value registers when reading ALRMVALH and ALRMVALL registers; the ALRMPTR<1:0> value decrements on every read or write of ALRMVALH until it reaches '00'.
ALRMVAL<15:8>:
 00 = ALRMMIN
 01 = ALRMWD
 10 = ALRMMNTH
 11 = Unimplemented
ALRMVAL<7:0>:
 00 = ALRMSEC
 01 = ALRMHR
 10 = ALRMDAY
 11 = Unimplemented
- bit 7-0 **ARPT<7:0>:** Alarm Repeat Counter Value bits
 11111111 = Alarm will repeat 255 more times
 •
 •
 •
 00000000 = Alarm will not repeat
 The counter decrements on any alarm event. The counter is prevented from rolling over from 00h to FFh unless CHIME = 1.

REGISTER 25-4: RTCVAL (WHEN RTCPTR<1:0> = 11): YEAR VALUE REGISTER⁽¹⁾

| | | | | | | | |
|--------|-----|-----|-----|-----|-----|-----|-------|
| 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 |
| YRTEN<3:0> | | | | YRONE<3:0> | | | |
| 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 15-8 **Unimplemented:** Read as '0'
- bit 7-4 **YRTEN<3:0>:** Binary Coded Decimal Value of Year's Tens Digit; contains a value from 0 to 9
- bit 3-0 **YRONE<3:0>:** 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 25-5: RTCVAL (WHEN RTCPTR<1:0> = 10): MONTH AND DAY VALUE REGISTER⁽¹⁾

| | | | | | | | |
|--------|-----|-----|---------|-------------|-----|-----|-------|
| U-0 | U-0 | U-0 | R-x | R-x | R-x | R-x | R-x |
| — | — | — | MHTTEN0 | MTHONE<3:0> | | | |
| 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 |
| — | — | DAYTEN<1:0> | | DAYONE<3:0> | | | |
| 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 **MHTTEN0:** Binary Coded Decimal Value of Month's Tens Digit; contains a value of 0 or 1
- bit 11-8 **MTHONE<3:0>:** 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 **DAYTEN<1:0>:** Binary Coded Decimal Value of Day's Tens Digit; contains a value from 0 to 3
- bit 3-0 **DAYONE<3:0>:** 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 25-6: RTCVAL (WHEN RTCPTR<1:0> = 01): WKDYHR: WEEKDAY AND HOURS VALUE REGISTER⁽¹⁾

| | | | | | | | |
|--------|-----|-----|-----|-----|-----------|-------|-------|
| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-x | R/W-x | R/W-x |
| — | — | — | — | — | WDAY<2:0> | | |
| 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 |
| — | — | HRTEN<1:0> | | HRONE<3:0> | | | |
| 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 **WDAY<2:0>:** Binary Coded Decimal Value of Weekday Digit; contains a value from 0 to 6
- bit 7-6 **Unimplemented:** Read as '0'
- bit 5-4 **HRTEN<1:0>:** Binary Coded Decimal Value of Hour's Tens Digit; contains a value from 0 to 2
- bit 3-0 **HRONE<3:0>:** 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 25-7: RTCVAL (WHEN RTCPTR<1:0> = 00): 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 |
| — | MINTEN<2:0> | | | MINONE<3:0> | | | |
| 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 |
| — | SECTEN<2:0> | | | SECONE<3:0> | | | |
| 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 **Unimplemented:** Read as '0'
- bit 14-12 **MINTEN<2:0>:** Binary Coded Decimal Value of Minute's Tens Digit; contains a value from 0 to 5
- bit 11-8 **MINONE<3:0>:** 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 **SECTEN<2:0>:** Binary Coded Decimal Value of Second's Tens Digit; contains a value from 0 to 5
- bit 3-0 **SECONE<3:0>:** Binary Coded Decimal Value of Second's Ones Digit; contains a value from 0 to 9

REGISTER 25-8: ALRMVAL (WHEN ALRMPTR<1:0> = 10): ALARM MONTH AND DAY VALUE REGISTER⁽¹⁾

| | | | | | | | | |
|--------|-----|-----|---------|-------------|-------|-------|-------|-------|
| U-0 | U-0 | U-0 | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | |
| — | — | — | MTHTEN0 | MTHONE<3:0> | | | | |
| 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 |
| — | — | DAYTEN<1:0> | | DAYONE<3:0> | | | |
| 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 **MTHTEN0:** Binary Coded Decimal Value of Month's Tens Digit; contains a value of 0 or 1
- bit 11-8 **MTHONE<3:0>:** 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 **DAYTEN<1:0>:** Binary Coded Decimal Value of Day's Tens Digit; contains a value from 0 to 3
- bit 3-0 **DAYONE<3:0>:** 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 25-9: ALRMVAL (WHEN ALRMPTR<1:0> = 01): ALARM WEEKDAY AND HOURS VALUE REGISTER⁽¹⁾

| | | | | | | | |
|--------|-----|-----|-----|-----|-------|-------|-------|
| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-x | R/W-x | R/W-x |
| — | — | — | — | — | WDAY2 | WDAY1 | WDAY0 |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------|-----|------------|-------|------------|-------|-------|-------|
| U-0 | U-0 | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
| — | — | HRTEN<1:0> | | HRONE<3:0> | | | |
| 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 **WDAY<2:0>:** Binary Coded Decimal Value of Weekday Digit; contains a value from 0 to 6
- bit 7-6 **Unimplemented:** Read as '0'
- bit 5-4 **HRTEN<1:0>:** Binary Coded Decimal Value of Hour's Tens Digit; contains a value from 0 to 2
- bit 3-0 **HRONE<3:0>:** 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 25-10: ALRMVAL (WHEN ALRMPTR<1:0> = 00): 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 |
| — | MINTEN<2:0> | | | MINONE<3:0> | | | |
| 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 |
| — | SECTEN<2:0> | | | SECONE<3:0> | | | |
| 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 **Unimplemented:** Read as '0'
- bit 14-12 **MINTEN<2:0>:** Binary Coded Decimal Value of Minute's Tens Digit; contains a value from 0 to 5
- bit 11-8 **MINONE<3:0>:** 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 **SECTEN<2:0>:** Binary Coded Decimal Value of Second's Tens Digit; contains a value from 0 to 5
- bit 3-0 **SECONE<3:0>:** Binary Coded Decimal Value of Second's Ones Digit; contains a value from 0 to 9

26.0 PROGRAMMABLE CYCLIC REDUNDANCY CHECK (CRC) GENERATOR

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 36. “Programmable Cyclic Redundancy Check (CRC)”, which is available from the Microchip website (www.microchip.com).

The programmable CRC generator offers the following features:

- User-programmable polynomial CRC equation
- Interrupt output
- Data FIFO

26.1 Overview

The module implements a software configurable CRC generator. The terms of the polynomial and its length can be programmed using the CRCXOR (X<15:1>) bits and the CRCCON (PLEN<3:0>) bits, respectively.

EQUATION 26-1: CRC EQUATION

$$x^{16} + x^{12} + x^5 + 1$$

To program this polynomial into the CRC generator, the CRC register bits should be set as shown in Table 26-1.

TABLE 26-1: EXAMPLE CRC SETUP

| Bit Name | Bit Value |
|-----------|-----------------|
| PLEN<3:0> | 1111 |
| X<15:1> | 000100000010000 |

For the value of X<15:1>, the 12th bit and the 5th bit are set to ‘1’, as required by the CRC equation. The 0th bit required by the CRC equation is always XORed. For a 16-bit polynomial, the 16th bit is also always assumed to be XORed; therefore, the X<15:1> bits do not have the 0th bit or the 16th bit.

The topology of a standard CRC generator is shown in Figure 26-2.

FIGURE 26-1: CRC SHIFTER DETAILS

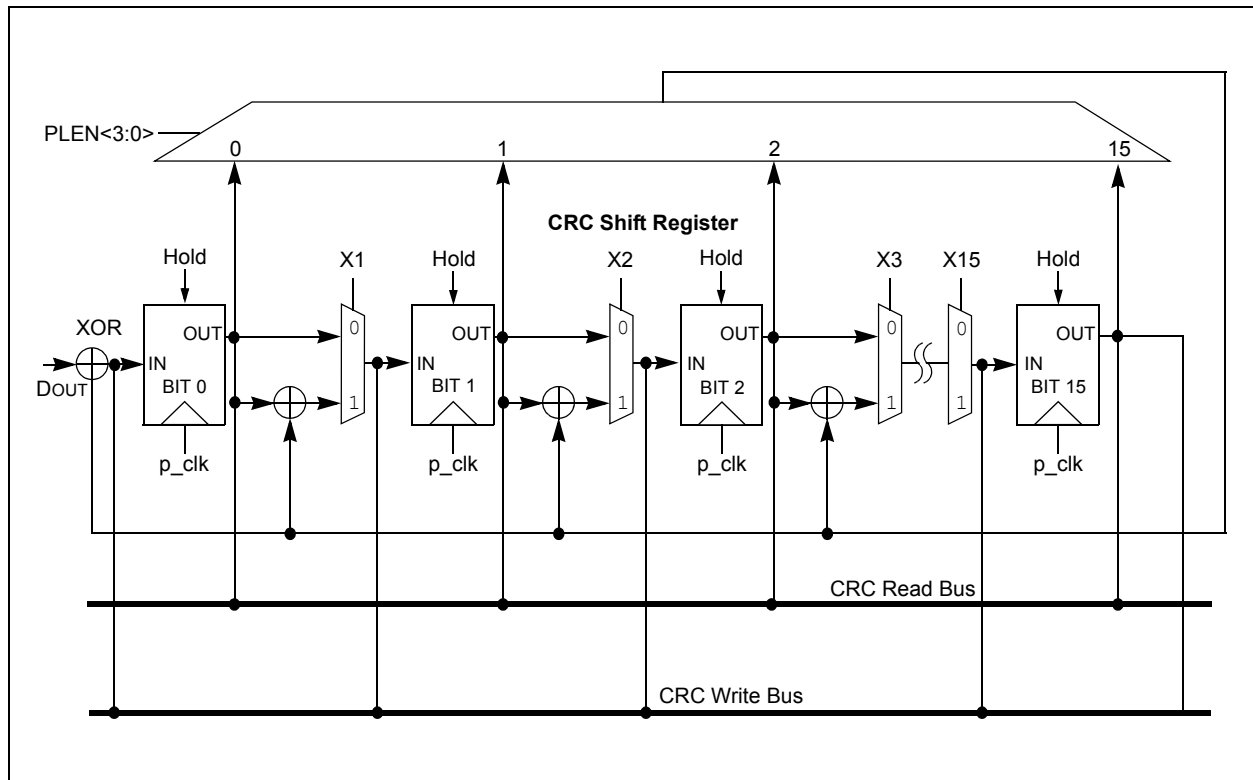
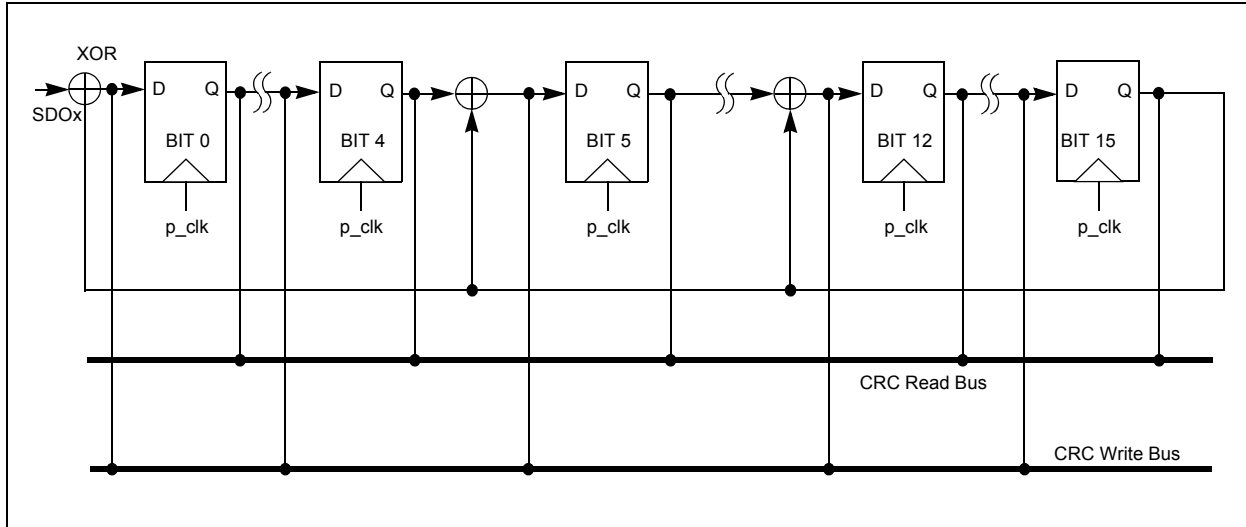


FIGURE 26-2: CRC GENERATOR RECONFIGURED FOR $x^{16} + x^{12} + x^5 + 1$



26.2 User Interface

26.2.1 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 VWORD ($VWORD < 4:0 >$) 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 26.2.2 "Interrupt Operation").

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

26.2.2 INTERRUPT OPERATION

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

26.3 Operation in Power Save Modes

26.3.1 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.

26.3.2 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.

26.4 Registers

The CRC module provides the following registers:

- CRC Control Register
- CRC XOR Polynomial Register

REGISTER 26-1: CRCCON: CRC CONTROL REGISTER

| | | | | | | | |
|--------|-----|-------|------------|-----|-----|-----|-------|
| U-0 | U-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| — | — | CSIDL | VWORD<4:0> | | | | |
| 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 | PLEN<3:0> | | | |
| 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 15-14 **Unimplemented:** Read as '0'

bit 13 **CSIDL:** CRC Stop in Idle Mode bit

- 1 = Discontinue module operation when device enters Idle mode
- 0 = Continue module operation in Idle mode

bit 12-8 **VWORD<4:0>:** Pointer Value bits

Indicates the number of valid words in the FIFO. Has a maximum value of 8 when PLEN<3:0> > 7, or 16 when PLEN<3:0> ≤ 7.

bit 7 **CRCFUL:** FIFO Full bit

- 1 = FIFO is full
- 0 = FIFO is not full

bit 6 **CRCMPT:** FIFO Empty Bit

- 1 = FIFO is empty
- 0 = FIFO is not empty

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

bit 4 **CRCGO:** Start CRC bit

- 1 = Start CRC serial shifter
- 0 = Turn off the CRC serial shifter after the FIFO is empty

bit 3-0 **PLEN<3:0>:** Polynomial Length bits

Denotes the length of the polynomial to be generated minus 1.

REGISTER 26-2: 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 |
| X<15:8> | | | | | | | |
| 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 |
| X<7:1> | | | | | | | — |
| bit 7 | | | | | | | bit 0 |

Legend:

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

bit 15-1 **X<15:1>**: XOR of Polynomial Term X^n Enable bits

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

27.0 PARALLEL MASTER PORT (PMP)

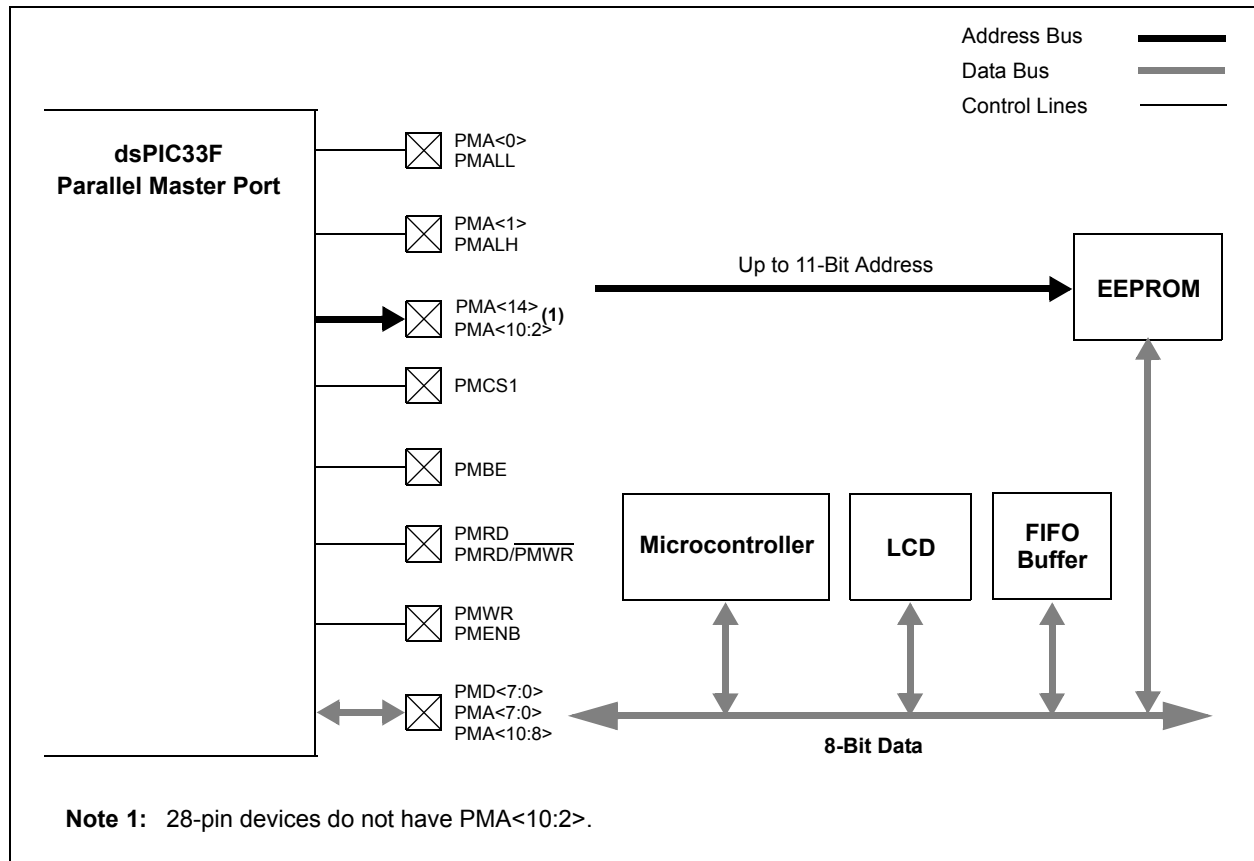
Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33F Family Reference Manual”, Section 35. “Parallel Master Port (PMP)”, which is available from the Microchip website (www.microchip.com).

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.

Key features of the PMP module include:

- Fully multiplexed address/data mode
 - 16-bits of address
- De multiplexed or partially multiplexed address/data mode
 - Up to 11 address lines with single chip select
 - Up to 12 address lines without chip select
- 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 27-1: PMP MODULE OVERVIEW



REGISTER 27-1: PMCON: PARALLEL PORT CONTROL REGISTER

| | | | | | | | |
|--------|-----|-------|---------|---------|--------|--------|--------|
| R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| PMPEN | — | PSIDL | ADRMUX1 | ADRMUX0 | PTBEEN | PTWREN | PTRDEN |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------|-------|----------------------|-----|----------------------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 ⁽¹⁾ | U-0 | R/W-0 ⁽¹⁾ | 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 **ADRMUX1:ADRMUX0:** Address/Data Multiplexing Selection bits⁽¹⁾
 11 =Reserved
 10 =All 16 bits of address are multiplexed on PMD<7:0> pins
 01 =Lower 8 bits of address are multiplexed on PMD<7:0> pins, upper 3 bits are multiplexed on PMA<10:8>
 00 =Address and data appear on separate pins
- bit 10 **PTBEEN:** Byte Enable Port Enable bit (16-bit Master mode)
 1 = PMBE port 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 select
 0x = PMCS1 functions as address bit 14
- bit 5 **ALP:** Address Latch Polarity bit⁽²⁾
 1 = Active-high ($\overline{\text{PMALL}}$ and $\overline{\text{PMALH}}$)
 0 = Active-low (PMALL and PMALH)
- bit 4 **Unimplemented:** Read as '0'
- bit 3 **CS1P:** Chip Select 1 Polarity bit⁽²⁾
 1 = Active-high (PMCS1/PMCS1)
 0 = Active-low (PMCS1/PMCS1)

Note 1: 28-pin devices do not have PMA<10:2>.

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

REGISTER 27-1: 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: 28-pin devices do not have PMA<10:2>.

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

Register 27-2: 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 | IRQM<1:0> | | INCM<1:0> | | MODE16 | MODE<1:0> | |
| 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 |
| WAITB<1:0> ⁽¹⁾ | | WAITM<3:0> | | | WAITE<1:0> ⁽¹⁾ | | |
| bit 7 | | | | | | bit 0 | |

Legend:

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

- bit 15 **BUSY:** Busy bit (Master mode only)
 - 1 = Port is busy (not useful when the processor stall is active)
 - 0 = Port is not busy
- bit 14-13 **IRQM<1:0>:** Interrupt Request Mode bits
 - 11 = Interrupt generated when Read Buffer 3 is read or Write Buffer 3 is written (Buffered PSP mode) or on a read or write operation when PMA<1:0> = 11 (Addressable PSP mode only)
 - 10 = No interrupt generated, processor stall activated
 - 01 = Interrupt generated at the end of the read/write cycle
 - 00 = No interrupt generated
- bit 12-11 **INCM<1:0>:** Increment Mode bits
 - 11 = PSP read and write buffers auto-increment (Legacy PSP mode only)
 - 10 = Decrement ADDR<10:0> by 1 every read/write cycle
 - 01 = Increment ADDR<10:0> by 1 every read/write cycle
 - 00 = No increment or decrement of address
- bit 10 **MODE16:** 8/16-bit Mode bit
 - 1 = 16-bit mode: data register is 16 bits, a read or write to the data register invokes two 8-bit transfers
 - 0 = 8-bit mode: data register is 8 bits, a read or write to the data register invokes one 8-bit transfer
- bit 9-8 **MODE<1:0>:** Parallel Port Mode Select bits
 - 11 = Master mode 1 (PMCS1, PMRD/PMWR, PMENB, PMBE, PMA<x:0> and PMD<7:0>)
 - 10 = Master mode 2 (PMCS1, PMRD, PMWR, PMBE, PMA<x:0> and PMD<7:0>)
 - 01 = Enhanced PSP, control signals (PMRD, PMWR, PMCS1, PMD<7:0> and PMA<1:0>)
 - 00 = Legacy Parallel Slave Port, control signals (PMRD, PMWR, PMCS1 and PMD<7:0>)
- bit 7-6 **WAITB<1:0>:** Data Setup to Read/Write Wait State Configuration bits⁽¹⁾
 - 11 = Data wait of 4 T_{cy}; multiplexed address phase of 4 T_{cy}
 - 10 = Data wait of 3 T_{cy}; multiplexed address phase of 3 T_{cy}
 - 01 = Data wait of 2 T_{cy}; multiplexed address phase of 2 T_{cy}
 - 00 = Data wait of 1 T_{cy}; multiplexed address phase of 1 T_{cy}
- bit 5-2 **WAITM<3:0>:** Read to Byte Enable Strobe Wait State Configuration bits
 - 1111 = Wait of additional 15 T_{cy}
 -
 -
 -
 - 0001 = Wait of additional 1 T_{cy}
 - 0000 = No additional wait cycles (operation forced into one T_{cy})
- bit 1-0 **WAITE<1:0>:** Data Hold After Strobe Wait State Configuration bits⁽¹⁾
 - 11 = Wait of 4 T_{cy}
 - 10 = Wait of 3 T_{cy}
 - 01 = Wait of 2 T_{cy}
 - 00 = Wait of 1 T_{cy}

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

REGISTER 27-3: PMADDR: PARALLEL PORT ADDRESS 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 |
| ADDR15 | CS1 | ADDR<13:8> | | | | | |
| 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> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

Legend:

R = Readable bit W = Writable 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 **ADDR15:** Parallel Port Destination Address bits
 bit 14 **CS1:** Chip Select 1 bit
 1 = Chip select 1 is active
 0 = Chip select 1 is inactive
 bit 13-0 **ADDR13:ADDR0:** Parallel Port Destination Address bits

REGISTER 27-4: 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 | — | — | — | PTEN<10:8> ⁽¹⁾ | | |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|--------------------------|-------|-------|-------|-------|-------|-----------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| PTEN<7:2> ⁽¹⁾ | | | | | | PTEN<1:0> | |
| 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 **Unimplemented:** Read as '0'
 bit 14 **PTEN14:** PMCS1 Strobe Enable bit
 1 = PMA14 functions as either PMA<14> bit or PMCS1
 0 = PMA14 pin functions as port I/O
 bit 13-11 **Unimplemented:** Read as '0'
 bit 10-2 **PTEN<10:2>:** PMP Address Port Enable bits⁽¹⁾
 1 = PMA<10:2> function as PMP address lines
 0 = PMA<10:2> function as port I/O
 bit 1-0 **PTEN<1:0>:** PMALH/PMALL Strobe Enable bits
 1 = PMA1 and PMA0 function as either PMA<1:0> or PMALH and PMALL
 0 = PMA1 and PMA0 pads functions as port I/O

Note 1: Devices with 28 pins do not have PMA<10:2>.

REGISTER 27-5: PMSTAT: PARALLEL PORT STATUS REGISTER

| | | | | | | | |
|--------|-----------|-----|-----|------|------|------|-------|
| R-0 | R/W-0, HS | U-0 | U-0 | R-0 | R-0 | R-0 | R-0 |
| IBF | IBOV | — | — | IB3F | IB2F | IB1F | IB0F |
| bit 15 | | | | | | | bit 8 |

| | | | | | | | |
|-------|-----------|-----|-----|------|------|------|-------|
| R-1 | R/W-0, HS | U-0 | U-0 | R-1 | R-1 | R-1 | R-1 |
| OBE | OBUF | — | — | OB3E | OB2E | OB1E | OB0E |
| bit 7 | | | | | | | bit 0 |

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

- bit 15 **IBF:** Input Buffer Full Status bit
 - 1 = All writable input buffer registers are full
 - 0 = Some or all of the writable input buffer registers are empty

- bit 14 **IBOV:** Input Buffer Overflow Status bit
 - 1 = A write attempt to a full input byte register occurred (must be cleared in software)
 - 0 = No overflow occurred

- bit 13-12 **Unimplemented:** Read as '0'

- bit 11-8 **IB3F:IB0F** Input Buffer x Status Full bits
 - 1 = Input buffer contains data that has not been read (reading buffer will clear this bit)
 - 0 = Input buffer does not contain any unread data

- bit 7 **OBE:** Output Buffer Empty Status bit
 - 1 = All readable output buffer registers are empty
 - 0 = Some or all of the readable output buffer registers are full

- bit 6 **OBUF:** Output Buffer Underflow Status bits
 - 1 = A read occurred from an empty output byte register (must be cleared in software)
 - 0 = No underflow occurred

- bit 5-4 **Unimplemented:** Read as '0'

- bit 3-0 **OB3E:OB0E** Output Buffer x Status Empty bit
 - 1 = Output buffer is empty (writing data to the buffer will clear this bit)
 - 0 = Output buffer contains data that has not been transmitted

REGISTER 27-6: 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 ⁽¹⁾ | PMP TTL |
| bit 7 | | | | | | bit 0 | |

Legend:

R = Readable bit W = Writable 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⁽¹⁾

- 1 = RTCC seconds clock is selected for the RTCC pin
- 0 = RTCC alarm pulse is selected for the RTCC pin

bit 0 **PMP TTL:** 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.

NOTES:

28.0 SPECIAL FEATURES

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the related section in the “dsPIC33F Family Reference Manual”, which is available from the Microchip website (www.microchip.com).

dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 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 and CodeGuard™ Security
- JTAG Boundary Scan Interface
- In-Circuit Serial Programming™ (ICSP™)
- In-Circuit emulation

28.1 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 0xF80000.

The individual Configuration bit descriptions for the FBS, FGS, FOSCSEL, FOSC, FWDT, and FPOR Configuration registers are shown in Table 28-2.

Note that address 0xF80000 is beyond the user program memory space. It belongs to the configuration memory space (0x800000-0xFFFFF), which can only be accessed using table reads and table writes.

The upper byte of all device Configuration registers 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.

To prevent inadvertent configuration changes during code execution, all programmable Configuration bits are write-once. After a bit is initially programmed during a power cycle, it cannot be written to again. Changing a device configuration requires that power to the device be cycled.

The Device Configuration register map is shown in Table 28-1.

TABLE 28-1: DEVICE CONFIGURATION REGISTER MAP

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|----------|--------------------|-------------------------|--------|---------|--------|--------------|------------|-------------|-------|
| 0xF80000 | FBS | RBS<1:0> | | — | — | BSS<2:0> | | BWRP | |
| 0xF80002 | FSS ⁽¹⁾ | RSS<1:0> | | — | — | SSS<2:0> | | SWRP | |
| 0xF80004 | FGS | — | — | — | — | — | GSS<1:0> | | GWRP |
| 0xF80006 | FOSCSEL | IESO | — | — | — | | FNOSC<2:0> | | |
| 0xF80008 | FOSC | FCKSM<1:0> | | IOL1WAY | — | — | OSCI0FNC | POSCMD<1:0> | |
| 0xF8000A | FWDT | FWDTEN | WINDIS | — | WDTPRE | WDTPOST<3:0> | | | |
| 0xF8000C | FPOR | PWMPIN | HPOL | LPOL | ALTI2C | — | FPWRT<2:0> | | |
| 0xF8000E | FICD | Reserved ⁽²⁾ | | JTAGEN | — | — | — | ICS<1:0> | |
| 0xF80010 | FUID0 | User Unit ID Byte 0 | | | | | | | |
| 0xF80012 | FUID1 | User Unit ID Byte 1 | | | | | | | |
| 0xF80014 | FUID2 | User Unit ID Byte 2 | | | | | | | |
| 0xF80016 | FUID3 | User Unit ID Byte 3 | | | | | | | |

Note 1: This Configuration register is not available on dsPIC33FJ32MC302/304 devices.

2: When read, these bits will appear as ‘1’. When you write to these bits, set these bits to ‘1’.

TABLE 28-2: dsPIC33F CONFIGURATION BITS DESCRIPTION

| Bit Field | Register | Description |
|-------------------------|--------------------|--|
| BWRP | FBS | Boot Segment Program Flash Write Protection 1 = Boot segment can be written 0 = Boot segment is write-protected |
| BSS<2:0> | FBS | Boot Segment Program Flash Code Protection Size x11 = No Boot program Flash segment Boot space is 1K Instruction Words (except interrupt vectors) 110 = Standard security; boot program Flash segment ends at 0x0007FE 010 = High security; boot program Flash segment ends at 0x0007FE Boot space is 4K Instruction Words (except interrupt vectors) 101 = Standard security; boot program Flash segment, ends at 0x001FFE 001 = High security; boot program Flash segment ends at 0x001FFE Boot space is 8K Instruction Words (except interrupt vectors) 100 = Standard security; boot program Flash segment ends at 0x003FFE 000 = High security; boot program Flash segment ends at 0x003FFE |
| RBS<1:0> ⁽¹⁾ | FBS | Boot Segment RAM Code Protection Size 11 = No Boot RAM defined 10 = Boot RAM is 128 bytes 01 = Boot RAM is 256 bytes 00 = Boot RAM is 1024 bytes |
| SWRP ⁽¹⁾ | FSS ⁽¹⁾ | Secure Segment Program Flash Write-Protect bit 1 = Secure Segment can bet written 0 = Secure Segment is write-protected |
| SSS<2:0> | FSS | Secure Segment Program Flash Code Protection Size (Secure segment is not implemented on 32K devices) x11 = No Secure program flash segment Secure space is 4K IW less BS 110 = Standard security; secure program flash segment starts at End of BS, ends at 0x001FFE 010 = High security; secure program flash segment starts at End of BS, ends at 0x001FFE Secure space is 8K IW less BS 101 = Standard security; secure program flash segment starts at End of BS, ends at 0x003FFE 001 = High security; secure program flash segment starts at End of BS, ends at 0x003FFE Secure space is 16K IW less BS 100 = Standard security; secure program flash segment starts at End of BS, ends at 007FFEH 000 = High security; secure program flash segment starts at End of BS, ends at 0x007FFE |
| RSS<1:0> ⁽¹⁾ | FSS ⁽¹⁾ | Secure Segment RAM Code Protection 11 = No Secure RAM defined 10 = Secure RAM is 256 Bytes less BS RAM 01 = Secure RAM is 2048 Bytes less BS RAM 00 = Secure RAM is 4096 Bytes less BS RAM |

Note 1: This Configuration register is not available on dsPIC33FJ32MC302/304 devices.

TABLE 28-2: dsPIC33F CONFIGURATION BITS DESCRIPTION (CONTINUED)

| Bit Field | Register | Description |
|-------------|----------|---|
| GSS<1:0> | FGS | General Segment Code-Protect bit 11 = User program memory is not code-protected 10 = Standard security 0x = High security |
| GWRP | FGS | General Segment Write-Protect bit 1 = User program memory is not write-protected 0 = User program memory is write-protected |
| IESO | FOSCSSEL | Two-speed Oscillator Start-up Enable bit 1 = Start-up device with FRC, then automatically switch to the user-selected oscillator source when ready 0 = Start-up device with user-selected oscillator source |
| FNOSC<2:0> | FOSCSSEL | Initial Oscillator Source Selection bits 111 = Internal Fast RC (FRC) oscillator with postscaler 110 = Internal Fast RC (FRC) oscillator with divide-by-16 101 = LPRC oscillator 100 = Secondary (LP) oscillator 011 = Primary (XT, HS, EC) oscillator with PLL 010 = Primary (XT, HS, EC) oscillator 001 = Internal Fast RC (FRC) oscillator with PLL 000 = FRC oscillator |
| FCKSM<1:0> | FOSC | Clock Switching Mode bits 1x = Clock switching is disabled, Fail-Safe Clock Monitor is disabled 01 = Clock switching is enabled, Fail-Safe Clock Monitor is disabled 00 = Clock switching is enabled, Fail-Safe Clock Monitor is enabled |
| IOL1WAY | FOSC | Peripheral pin select configuration 1 = Allow only one reconfiguration 0 = Allow multiple reconfigurations |
| OSCIOFNC | FOSC | OSC2 Pin Function bit (except in XT and HS modes) 1 = OSC2 is clock output 0 = OSC2 is general purpose digital I/O pin |
| POSCMD<1:0> | FOSC | Primary Oscillator Mode Select bits 11 = Primary oscillator disabled 10 = HS Crystal Oscillator mode 01 = XT Crystal Oscillator mode 00 = EC (External Clock) mode |
| FWDTEN | FWDT | Watchdog Timer Enable bit 1 = Watchdog Timer always enabled (LPRC oscillator cannot be disabled. Clearing the SWDTEN bit in the RCON register has no effect.) 0 = Watchdog Timer enabled/disabled by user software (LPRC can be disabled by clearing the SWDTEN bit in the RCON register) |
| WINDIS | FWDT | Watchdog Timer Window Enable bit 1 = Watchdog Timer in Non-Window mode 0 = Watchdog Timer in Window mode |
| WDTPRE | FWDT | Watchdog Timer Prescaler bit 1 = 1:128 0 = 1:32 |

Note 1: This Configuration register is not available on dsPIC33FJ32MC302/304 devices.

TABLE 28-2: dsPIC33F CONFIGURATION BITS DESCRIPTION (CONTINUED)

| Bit Field | Register | Description |
|--------------|----------|---|
| WDTPOST<3:0> | FWDT | Watchdog Timer Postscaler bits 1111 = 1:32,768 1110 = 1:16,384 • • • 0001 = 1:2 0000 = 1:1 |
| PWMPIN | FPOR | Motor Control PWM Module Pin Mode bit 1 = PWM module pins controlled by PORT register at device Reset (tri-stated) 0 = PWM module pins controlled by PWM module at device Reset (configured as output pins) |
| HPOL | FPOR | Motor Control PWM High Side Polarity bit 1 = PWM module high side output pins have active-high output polarity 0 = PWM module high side output pins have active-low output polarity |
| LPOL | FPOR | Motor Control PWM Low Side Polarity bit 1 = PWM module low side output pins have active-high output polarity 0 = PWM module low side output pins have active-low output polarity |
| FPWRT<2:0> | FPOR | Power-on Reset Timer Value Select bits 111 = PWRT = 128 ms 110 = PWRT = 64 ms 101 = PWRT = 32 ms 100 = PWRT = 16 ms 011 = PWRT = 8 ms 010 = PWRT = 4 ms 001 = PWRT = 2 ms 000 = PWRT = Disabled |
| ALTI2C | FPOR | Alternate I ² C™ pins 1 = I ² C mapped to SDA1/SCL1 pins 0 = I ² C mapped to ASDA1/ASCL1 pins |
| JTAGEN | FICD | JTAG Enable bit 1 = JTAG enabled 0 = JTAG disabled |
| ICS<1:0> | FICD | ICD Communication Channel Select bits 11 = Communicate on PGEC1 and PGED1 10 = Communicate on PGEC2 and PGED2 01 = Communicate on PGEC3 and PGED3 00 = Reserved, do not use |

Note 1: This Configuration register is not available on dsPIC33FJ32MC302/304 devices.

28.2 On-Chip Voltage Regulator

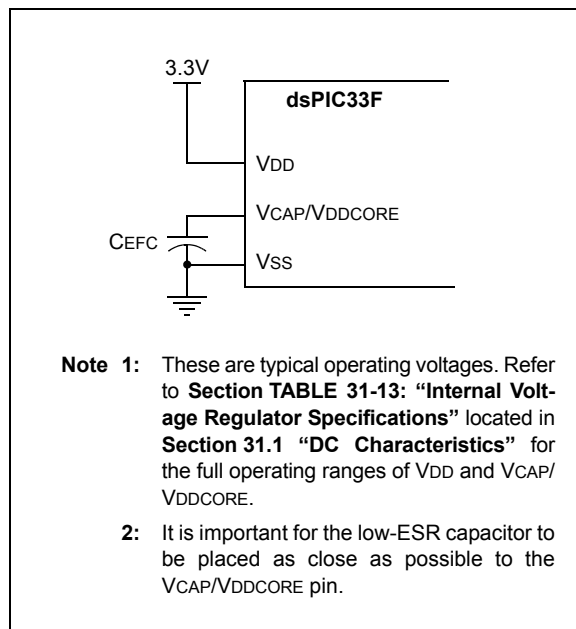
All of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices power their core digital logic at a nominal 2.5V. This can create a conflict for designs that are required to operate at a higher typical voltage, such as 3.3V. To simplify system design, all devices in the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 family incorporate an on-chip regulator that allows the device to run its core logic from VDD.

The regulator provides power to the core from the other VDD pins. When the regulator is enabled, a low-ESR (less than 5 Ohms) capacitor (such as tantalum or ceramic) must be connected to the VCAP/VDDCORE pin (Figure 28-1). This helps to maintain the stability of the regulator. The recommended value for the filter capacitor is provided in Table 31-13 located in Section 31.0 “Electrical Characteristics”.

Note: It is important for the low-ESR capacitor to be placed as close as possible to the VCAP/VDDCORE pin.

On a POR, it takes approximately 20 μ s for the on-chip voltage regulator to generate an output voltage. During this time, designated as TSTARTUP, code execution is disabled. TSTARTUP is applied every time the device resumes operation after a power-down.

FIGURE 28-1: CONNECTIONS FOR THE ON-CHIP VOLTAGE REGULATOR⁽¹⁾



28.3 BOR: Brown-Out Reset

The Brown-out Reset (BOR) module is based on an internal voltage reference circuit that monitors the regulated supply voltage VCAP/VDDCORE. The main purpose of the BOR module is to generate a device Reset when a brown-out condition occurs. Brown-out conditions are generally caused by glitches on the AC mains (for example, missing portions of the AC cycle waveform due to bad power transmission lines, or voltage sags due to excessive current draw when a large inductive load is turned on).

A BOR generates a Reset pulse, which resets the device. The BOR selects the clock source, based on the device Configuration bit values (FNOSC<2:0> and POSCMD<1:0>).

If an oscillator mode is selected, the BOR activates the Oscillator Start-up Timer (OST). The system clock is held until OST expires. If the PLL is used, the clock is held until the LOCK bit (OSCCON<5>) is ‘1’.

Concurrently, the PWRT time-out (TPWRT) is applied before the internal Reset is released. If TPWRT = 0 and a crystal oscillator is being used, then a nominal delay of TFSCM = 100 is applied. The total delay in this case is TFSCM.

The BOR Status bit (RCON<1>) is set to indicate that a BOR has occurred. The BOR circuit, if enabled, continues to operate while in Sleep or Idle modes and resets the device should VDD fall below the BOR threshold voltage.

28.4 Watchdog Timer (WDT)

For dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices, the WDT is driven by the LPRC oscillator. When the WDT is enabled, the clock source is also enabled.

28.4.1 PRESCALER/POSTSCALER

The nominal WDT clock source from LPRC is 32 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 WDTPRE Configuration bit. With a 32 kHz input, the prescaler yields a nominal WDT time-out period (T_{WDT}) 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 WDTPOST<3:0> Configuration bits (FWDT<3:0>), which allow the selection 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

Note: The CLRWDT and PWRSAV instructions clear the prescaler and postscaler counts when executed.

28.4.2 SLEEP AND IDLE MODES

If the WDT is enabled, it continues to run during Sleep or Idle modes. When the WDT time-out occurs, the device wakes the device and code execution continues from where the PWRSAV instruction was executed. The corresponding SLEEP or IDLE bits (RCON<3,2>) needs to be cleared in software after the device wakes up.

28.4.3 ENABLING WDT

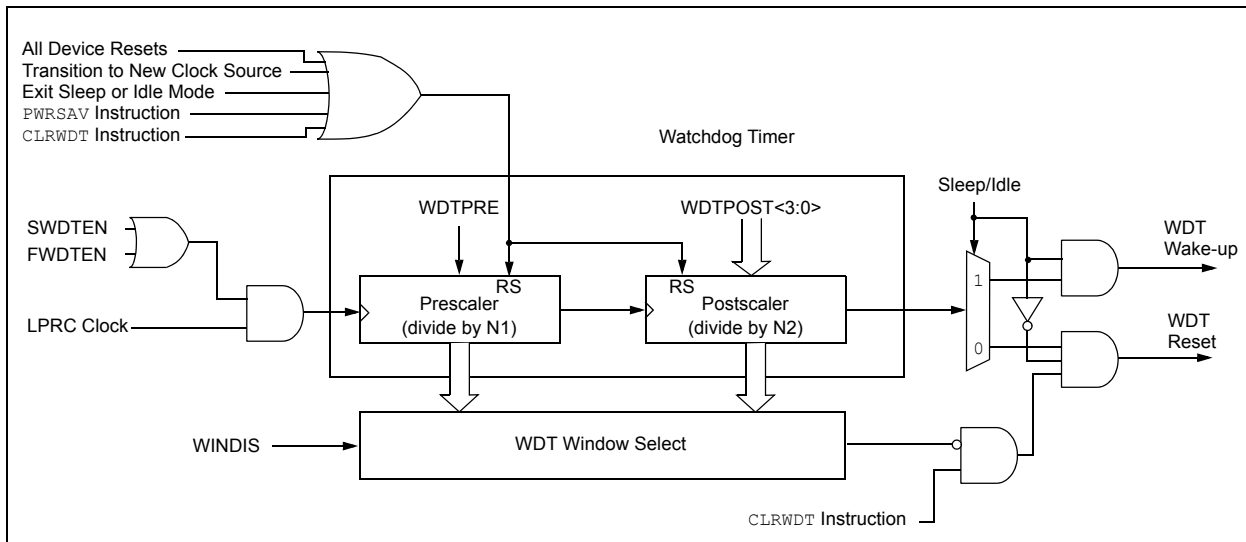
The WDT is enabled or disabled by the FWDTEN Configuration bit in the FWDT Configuration register. When the FWDTEN Configuration bit is set, the WDT is always enabled.

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 application to enable the WDT for critical code segments and disable the WDT during non-critical segments for maximum power savings.

Note: If the WINDIS bit (FWDT<6>) is cleared, the CLRWDT instruction should be executed by the application software only during the last 1/4 of the WDT period. This CLRWDT window can be determined by using a timer. If a CLRWDT instruction is executed before this window, a WDT Reset occurs.

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.

FIGURE 28-2: WDT BLOCK DIAGRAM



28.5 JTAG Interface

dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices implement a JTAG interface, which supports boundary scan device testing, as well as in-circuit programming. Detailed information on this interface is provided in future revisions of the document.

Note: Refer to **Section 24. “Programming and Diagnostics”** (DS70207) of the *dsPIC33F Family Reference Manual* for further information on usage, configuration and operation of the JTAG interface.

28.6 In-Circuit Serial Programming

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices can be serially programmed while in the end application circuit. This is done with two lines for clock and data and three other lines for power, ground and the programming sequence. Serial programming allows customers to manufacture boards with unprogrammed devices and then program the digital signal controller just before shipping the product. Serial programming also allows the most recent firmware or a custom firmware to be programmed. Refer to the “*dsPIC33F/PIC24H Flash Programming Specification*” (DS70152) for details about In-Circuit Serial Programming (ICSP).

Any of the three pairs of programming clock/data pins can be used:

- PGEC1 and PGED1
- PGEC2 and PGED2
- PGEC3 and PGED3

28.7 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 PGECx (Emulation/Debug Clock) and PGEDx (Emulation/Debug Data) pin functions.

Any of the three pairs of debugging clock/data pins can be used:

- PGEC1 and PGED1
- PGEC2 and PGED2
- PGEC3 and PGED3

To use the in-circuit debugger function of the device, the design must implement ICSP connections to MCLR, VDD, VSS, PGC, PGD and the PGECx/PGEDx 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.

28.8 Code Protection and CodeGuard Security

The dsPIC33FJ64MCX02/X04 and dsPIC33FJ128MCX02/X04 devices offer advanced implementation of CodeGuard Security that supports BS, SS and GS while, the dsPIC33FJ32MC302/304 devices offer the intermediate level of CodeGuard Security that supports only BS and GS. CodeGuard Security enables multiple parties to securely share resources (memory, interrupts and peripherals) on a single chip. This feature helps protect individual Intellectual Property in collaborative system designs.

When coupled with software encryption libraries, CodeGuard Security can be used to securely update Flash even when multiple IPs reside on the single chip. The code protection features vary depending on the actual dsPIC33F implemented. The following sections provide an overview of these features.

Secure segment and RAM protection is implemented on the dsPIC33FJ64MCX02/X04 and dsPIC33FJ128MCX02/X04 devices. The dsPIC33FJ32MC302/304 devices do not support secure segment and RAM protection.

Note: Refer to **Section 23. “CodeGuard™ Security”** (DS70199) of the *dsPIC33F Family Reference Manual* for further information on usage, configuration and operation of CodeGuard Security.

TABLE 28-3: CODE FLASH SECURITY SEGMENT SIZES FOR 32K BYTE DEVICES

| CONFIG BITS | BSS<2:0>=x11 0K | BSS<2:0>=x10 1K | BSS<2:0>=x01 4K | BSS<2:0>=x00 8K | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|--|--|-----------------|--|--|---------|---|-------------|--|-------------|--|---------------|--|--|---------|---|-------------|--|--------------|--|--------------|--|--|---------|---|-------------|--|--------------|--|--------------|--|--|---------|
| SSS<2:0> = x11 0K | <table border="1"> <tr> <td>VS = 256 IW</td> <td>000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh</td> </tr> <tr> <td>GS = 11008 IW</td> <td>000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh</td> </tr> <tr> <td style="background-color: #cccccc;"></td> <td>0157FEh</td> </tr> </table> | VS = 256 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | GS = 11008 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | | 0157FEh | <table border="1"> <tr> <td>VS = 256 IW</td> <td>000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh</td> </tr> <tr> <td>BS = 768 IW</td> <td>000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh</td> </tr> <tr> <td>GS = 10240 IW</td> <td>000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh</td> </tr> <tr> <td style="background-color: #cccccc;"></td> <td>0157FEh</td> </tr> </table> | VS = 256 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | BS = 768 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | GS = 10240 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | | 0157FEh | <table border="1"> <tr> <td>VS = 256 IW</td> <td>000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh</td> </tr> <tr> <td>BS = 3840 IW</td> <td>000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh</td> </tr> <tr> <td>GS = 7168 IW</td> <td>000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh</td> </tr> <tr> <td style="background-color: #cccccc;"></td> <td>0157FEh</td> </tr> </table> | VS = 256 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | BS = 3840 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | GS = 7168 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | | 0157FEh | <table border="1"> <tr> <td>VS = 256 IW</td> <td>000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh</td> </tr> <tr> <td>BS = 7936 IW</td> <td>000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh</td> </tr> <tr> <td>GS = 3072 IW</td> <td>000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh</td> </tr> <tr> <td style="background-color: #cccccc;"></td> <td>0157FEh</td> </tr> </table> | VS = 256 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | BS = 7936 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | GS = 3072 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | | 0157FEh |
| | VS = 256 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | GS = 11008 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0157FEh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VS = 256 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BS = 768 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| GS = 10240 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 0157FEh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VS = 256 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BS = 3840 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| GS = 7168 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 0157FEh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VS = 256 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BS = 7936 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| GS = 3072 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 0157FEh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

TABLE 28-4: CODE FLASH SECURITY SEGMENT SIZES FOR 64K BYTE DEVICES

| CONFIG BITS | BSS<2:0>=x11 0K | BSS<2:0>=x10 1K | BSS<2:0>=x01 4K | BSS<2:0>=x00 8K |
|---------------------------|--|---|--|---|
| SSS<2:0> = x11 0K | VS = 256 IW 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh GS = 21760 IW | VS = 256 IW 000000h 0001FEh BS = 768 IW 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh GS = 20992 IW | VS = 256 IW 000000h 0001FEh BS = 3840 IW 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh GS = 17920 IW | VS = 256 IW 000000h 0001FEh BS = 7936 IW 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh GS = 13824 IW |
| | 0157FEh | 0157FEh | 0157FEh | 0157FEh |
| SSS<2:0> = x10 4K | VS = 256 IW 000000h 0001FEh 000200h 0007FEh SS = 3840 IW 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh GS = 17920 IW | VS = 256 IW 000000h 0001FEh BS = 768 IW 000200h 0007FEh SS = 3072 IW 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh GS = 17920 IW | VS = 256 IW 000000h 0001FEh BS = 3840 IW 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh GS = 17920 IW | VS = 256 IW 000000h 0001FEh BS = 7936 IW 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh GS = 13824 IW |
| | 0157FEh | 0157FEh | 0157FEh | 0157FEh |
| SSS<2:0> = x01 8K | VS = 256 IW 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh SS = 7936 IW GS = 13824 IW | VS = 256 IW 000000h 0001FEh BS = 768 IW 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh SS = 7168 IW GS = 13824 IW | VS = 256 IW 000000h 0001FEh BS = 3840 IW 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh SS = 4096 IW GS = 13824 IW | VS = 256 IW 000000h 0001FEh BS = 7936 IW 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh GS = 13824 IW |
| | 0157FEh | 0157FEh | 0157FEh | 0157FEh |
| SSS<2:0> = x00 16K | VS = 256 IW 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh SS = 16128 IW GS = 5632 IW | VS = 256 IW 000000h 0001FEh BS = 768 IW 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh SS = 15360 IW GS = 5632 IW | VS = 256 IW 000000h 0001FEh BS = 3840 IW 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh SS = 12288 IW GS = 5632 IW | VS = 256 IW 000000h 0001FEh BS = 7936 IW 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh SS = 8192 IW GS = 5632 IW |
| | 0157FEh | 0157FEh | 0157FEh | 0157FEh |

TABLE 28-5: CODE FLASH SECURITY SEGMENT SIZES FOR 128K BYTE DEVICES

| CONFIG BITS | BSS<2:0>=x11 0K | | BSS<2:0>=x10 1K | | BSS<2:0>=x01 4K | | BSS<2:0>=x00 8K | |
|---------------------------|-----------------|---|---|---|--|---|---|---|
| SSS<2:0> = x11 0K | VS = 256 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h | VS = 256 IW BS = 768 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h | VS = 256 IW BS = 3840 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h | VS = 256 IW BS = 7936 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h |
| | GS = 43776 IW | 0157FEh | GS = 43008 IW | 0157FEh | GS = 39936 IW | 0157FEh | GS = 35840 IW | 0157FEh |
| SSS<2:0> = x10 4K | VS = 256 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh | VS = 256 IW BS = 768 IW SS = 3072 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh | VS = 256 IW BS = 3840 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh | VS = 256 IW BS = 7936 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh |
| | GS = 39936 IW | 0157FEh | GS = 39936 IW | 0157FEh | GS = 39936 IW | 0157FEh | GS = 35840 IW | 0157FEh |
| SSS<2:0> = x01 8K | VS = 256 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h | VS = 256 IW BS = 768 IW SS = 7168 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h | VS = 256 IW BS = 3840 IW SS = 4096 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h | VS = 256 IW BS = 7936 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h |
| | GS = 35840 IW | 0157FEh | GS = 35840 IW | 0157FEh | GS = 35840 IW | 0157FEh | GS = 35840 IW | 0157FEh |
| SSS<2:0> = x00 16K | VS = 256 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h | VS = 256 IW BS = 768 IW SS = 15360 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h | VS = 256 IW BS = 3840 IW SS = 12288 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h | VS = 256 IW BS = 7936 IW SS = 8192 IW | 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h |
| | GS = 27648 IW | 0157FEh | GS = 27648 IW | 0157FEh | GS = 27648 IW | 0157FEh | GS = 27648 IW | 0157FEh |

29.0 INSTRUCTION SET SUMMARY

Note: This data sheet summarizes the features of the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the related section in the “dsPIC33F Family Reference Manual”, which is available from the Microchip website (www.microchip.com).

The dsPIC33F instruction set is identical to that of the dsPIC30F.

Most instructions are a single program memory word (24 bits). 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 five basic categories:

- Word or byte-oriented operations
- Bit-oriented operations
- Literal operations
- DSP operations
- Control operations

Table 29-1 shows the general symbols used in describing the instructions.

The dsPIC33F instruction set summary in Table 29-2 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 be either the file register ‘f’ or the W0 register, which is denoted as ‘WREG’

Most bit-oriented instructions (including simple rotate/shift instructions) have two operands:

- The W register (with or without an address modifier) or file register (specified by the value of ‘Ws’ or ‘f’)
- The bit in the W register or file register (specified by a literal value or indirectly by the contents of register ‘Wb’)

The literal instructions that involve data movement can use some of the following operands:

- A literal value to be loaded into a W register or file register (specified by ‘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 MAC class of DSP instructions can use some of the following operands:

- The accumulator (A or B) to be used (required operand)
- The W registers to be used as the two operands
- The X and Y address space prefetch operations
- The X and Y address space prefetch destinations
- The accumulator write back destination

The other DSP instructions do not involve any multiplication and can include:

- The accumulator to be used (required)
- The source or destination operand (designated as Wso or Wdo, respectively) with or without an address modifier
- The amount of shift specified by a W register ‘Wn’ or a literal value

The control instructions can use some of the following operands:

- A program memory address
- The mode of the table read and table write instructions

dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, AND dsPIC33FJ128MCX02/X04

Most instructions are a single word. Certain double-word instructions are designed to provide all the required information 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 executes as a NOP.

The double-word instructions execute in two instruction cycles.

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.

Note: For more details on the instruction set, refer to the "dsPIC30/33F Programmer's Reference Manual" (DS70157).

TABLE 29-1: 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) |
| Acc | One of two accumulators {A, B} |
| AWB | Accumulator write back destination address register $\in \{W13, [W13]+ = 2\}$ |
| 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 $\in \{0x0000...0x1FFF\}$ |
| 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, can be blank |
| OA, OB, SA, SB | DSP Status bits: ACCA Overflow, ACCB Overflow, ACCA Saturate, ACCB Saturate |
| 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) |
| Wm*Wm | Multiplicand and Multiplier working register pair for Square instructions $\in \{W4 * W4, W5 * W5, W6 * W6, W7 * W7\}$ |

TABLE 29-1: SYMBOLS USED IN OPCODE DESCRIPTIONS (CONTINUED)

| Field | Description |
|-------|---|
| Wm*Wn | Multiplicand and Multiplier working register pair for DSP instructions $\in \{W4 * W5, W4 * W6, W4 * W7, W5 * W6, W5 * W7, W6 * W7\}$ |
| 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 $\in \{Wns, [Wns], [Wns++] , [Wns--], [++Wns], [--Wns], [Wns+Wb] \}$ |
| Wx | X data space prefetch address register for DSP instructions $\in \{[W8] + = 6, [W8] + = 4, [W8] + = 2, [W8], [W8] - = 6, [W8] - = 4, [W8] - = 2, [W9] + = 6, [W9] + = 4, [W9] + = 2, [W9], [W9] - = 6, [W9] - = 4, [W9] - = 2, [W9 + W12], \text{none}\}$ |
| Wxd | X data space prefetch destination register for DSP instructions $\in \{W4...W7\}$ |
| Wy | Y data space prefetch address register for DSP instructions $\in \{[W10] + = 6, [W10] + = 4, [W10] + = 2, [W10], [W10] - = 6, [W10] - = 4, [W10] - = 2, [W11] + = 6, [W11] + = 4, [W11] + = 2, [W11], [W11] - = 6, [W11] - = 4, [W11] - = 2, [W11 + W12], \text{none}\}$ |
| Wyd | Y data space prefetch destination register for DSP instructions $\in \{W4...W7\}$ |

TABLE 29-2: INSTRUCTION SET OVERVIEW

| Base Instr # | Assembly Mnemonic | Assembly Syntax | Description | # of Words | # of Cycles | Status Flags Affected |
|--------------|-------------------|-----------------------|---|------------|-------------|-----------------------|
| 1 | ADD | ADD Acc | Add Accumulators | 1 | 1 | OA,OB,SA,SB |
| | | ADD f | $f = f + \text{WREG}$ | 1 | 1 | C,DC,N,OV,Z |
| | | ADD f, WREG | $\text{WREG} = f + \text{WREG}$ | 1 | 1 | C,DC,N,OV,Z |
| | | ADD #lit10, Wn | $\text{Wd} = \text{lit10} + \text{Wd}$ | 1 | 1 | C,DC,N,OV,Z |
| | | ADD Wb, Ws, Wd | $\text{Wd} = \text{Wb} + \text{Ws}$ | 1 | 1 | C,DC,N,OV,Z |
| | | ADD Wb, #lit5, Wd | $\text{Wd} = \text{Wb} + \text{lit5}$ | 1 | 1 | C,DC,N,OV,Z |
| 2 | ADDC | ADDC f | $f = f + \text{WREG} + (C)$ | 1 | 1 | C,DC,N,OV,Z |
| | | ADDC f, WREG | $\text{WREG} = f + \text{WREG} + (C)$ | 1 | 1 | C,DC,N,OV,Z |
| | | ADDC #lit10, Wn | $\text{Wd} = \text{lit10} + \text{Wd} + (C)$ | 1 | 1 | C,DC,N,OV,Z |
| | | ADDC Wb, Ws, Wd | $\text{Wd} = \text{Wb} + \text{Ws} + (C)$ | 1 | 1 | C,DC,N,OV,Z |
| | | ADDC Wb, #lit5, Wd | $\text{Wd} = \text{Wb} + \text{lit5} + (C)$ | 1 | 1 | C,DC,N,OV,Z |
| | | ADDC Wso, #Slit4, Acc | 16-bit Signed Add to Accumulator | 1 | 1 | OA,OB,SA,SB |
| 3 | AND | AND f | $f = f .\text{AND. WREG}$ | 1 | 1 | N,Z |
| | | AND f, WREG | $\text{WREG} = f .\text{AND. WREG}$ | 1 | 1 | N,Z |
| | | AND #lit10, Wn | $\text{Wd} = \text{lit10} .\text{AND. Wd}$ | 1 | 1 | N,Z |
| | | AND Wb, Ws, Wd | $\text{Wd} = \text{Wb} .\text{AND. Ws}$ | 1 | 1 | N,Z |
| | | AND Wb, #lit5, Wd | $\text{Wd} = \text{Wb} .\text{AND. lit5}$ | 1 | 1 | N,Z |
| 4 | ASR | ASR f | $f = \text{Arithmetic Right Shift } f$ | 1 | 1 | C,N,OV,Z |
| | | ASR f, WREG | $\text{WREG} = \text{Arithmetic Right Shift } f$ | 1 | 1 | C,N,OV,Z |
| | | ASR Ws, Wd | $\text{Wd} = \text{Arithmetic Right Shift } \text{Ws}$ | 1 | 1 | C,N,OV,Z |
| | | ASR Wb, Wns, Wnd | $\text{Wnd} = \text{Arithmetic Right Shift } \text{Wb by } \text{Wns}$ | 1 | 1 | N,Z |
| | | ASR Wb, #lit5, Wnd | $\text{Wnd} = \text{Arithmetic Right Shift } \text{Wb by } \text{lit5}$ | 1 | 1 | N,Z |
| 5 | BCLR | BCLR f, #bit4 | Bit Clear f | 1 | 1 | None |
| | | BCLR Ws, #bit4 | Bit Clear Ws | 1 | 1 | None |
| 6 | 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 OA, Expr | Branch if Accumulator A overflow | 1 | 1 (2) | None |
| | | BRA OB, Expr | Branch if Accumulator B overflow | 1 | 1 (2) | None |
| | | BRA OV, Expr | Branch if Overflow | 1 | 1 (2) | None |
| | | BRA SA, Expr | Branch if Accumulator A saturated | 1 | 1 (2) | None |
| | | BRA SB, Expr | Branch if Accumulator B saturated | 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 | | |
| 7 | BSET | BSET f, #bit4 | Bit Set f | 1 | 1 | None |
| | | BSET Ws, #bit4 | Bit Set Ws | 1 | 1 | None |
| 8 | 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 |
| 9 | BTG | BTG f, #bit4 | Bit Toggle f | 1 | 1 | None |
| | | BTG Ws, #bit4 | Bit Toggle Ws | 1 | 1 | None |

TABLE 29-2: INSTRUCTION SET OVERVIEW (CONTINUED)

| Base Instr # | Assembly Mnemonic | Assembly Syntax | Description | # of Words | # of Cycles | Status Flags Affected |
|--------------|-------------------|--------------------------------|---|------------|---------------|-----------------------|
| 10 | BTSC | BTSC f, #bit4 | Bit Test f, Skip if Clear | 1 | 1 (2 or 3) | None |
| | | BTSC Ws, #bit4 | Bit Test Ws, Skip if Clear | 1 | 1 (2 or 3) | None |
| 11 | BTSS | BTSS f, #bit4 | Bit Test f, Skip if Set | 1 | 1 (2 or 3) | None |
| | | BTSS Ws, #bit4 | Bit Test Ws, Skip if Set | 1 | 1 (2 or 3) | None |
| 12 | 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 |
| 13 | 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 |
| 14 | CALL | CALL lit23 | Call subroutine | 2 | 2 | None |
| | | CALL Wn | Call indirect subroutine | 1 | 2 | None |
| 15 | CLR | CLR f | f = 0x0000 | 1 | 1 | None |
| | | CLR WREG | WREG = 0x0000 | 1 | 1 | None |
| | | CLR Ws | Ws = 0x0000 | 1 | 1 | None |
| | | CLR Acc, Wx, Wxd, Wy, Wyd, AWB | Clear Accumulator | 1 | 1 | OA,OB,SA,SB |
| 16 | CLRWDT | CLRWDT | Clear Watchdog Timer | 1 | 1 | WDTO, Sleep |
| 17 | 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 |
| 18 | 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 |
| 19 | 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 |
| 20 | 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 |
| 21 | CPSEQ | CPSEQ Wb, Wn | Compare Wb with Wn, skip if = | 1 | 1 (2 or 3) | None |
| 22 | CPSGT | CPSGT Wb, Wn | Compare Wb with Wn, skip if > | 1 | 1 (2 or 3) | None |
| 23 | CPSLT | CPSLT Wb, Wn | Compare Wb with Wn, skip if < | 1 | 1 (2 or 3) | None |
| 24 | CPSNE | CPSNE Wb, Wn | Compare Wb with Wn, skip if ≠ | 1 | 1 (2 or 3) | None |
| 25 | DAW | DAW Wn | Wn = decimal adjust Wn | 1 | 1 | C |
| 26 | 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 |
| 27 | 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 |
| 28 | DISI | DISI #lit14 | Disable Interrupts for k instruction cycles | 1 | 1 | None |

TABLE 29-2: INSTRUCTION SET OVERVIEW (CONTINUED)

| Base Instr # | Assembly Mnemonic | Assembly Syntax | Description | # of Words | # of Cycles | Status Flags Affected |
|---------------|--|---------------------------------------|--|------------|-------------|-----------------------|
| 29 | DIV | DIV.S 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.U 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 |
| 30 | DIVF | DIVF Wm, Wn | Signed 16/16-bit Fractional Divide | 1 | 18 | N,Z,C,OV |
| 31 | DO | DO #lit14, Expr | Do code to PC + Expr, lit14 + 1 times | 2 | 2 | None |
| | | DO Wn, Expr | Do code to PC + Expr, (Wn) + 1 times | 2 | 2 | None |
| 32 | ED | ED Wm*Wm, Acc, Wx, Wy, Wxd | Euclidean Distance (no accumulate) | 1 | 1 | OA,OB,OAB,SA,SB,SAB |
| 33 | EDAC | EDAC Wm*Wm, Acc, Wx, Wy, Wxd | Euclidean Distance | 1 | 1 | OA,OB,OAB,SA,SB,SAB |
| 34 | EXCH | EXCH Wns, Wnd | Swap Wns with Wnd | 1 | 1 | None |
| 35 | FBCL | FBCL Ws, Wnd | Find Bit Change from Left (MSb) Side | 1 | 1 | C |
| 36 | FF1L | FF1L Ws, Wnd | Find First One from Left (MSb) Side | 1 | 1 | C |
| 37 | FF1R | FF1R Ws, Wnd | Find First One from Right (LSb) Side | 1 | 1 | C |
| 38 | GOTO | GOTO Expr | Go to address | 2 | 2 | None |
| | | GOTO Wn | Go to indirect | 1 | 2 | None |
| 39 | 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 |
| 40 | 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 |
| 41 | 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 |
| 42 | LAC | LAC Wso, #Slit4, Acc | Load Accumulator | 1 | 1 | OA,OB,OAB,SA,SB,SAB |
| 43 | LNK | LNK #lit14 | Link Frame Pointer | 1 | 1 | None |
| 44 | 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 |
| 45 | MAC | MAC Wm*Wn, Acc, Wx, Wxd, Wy, Wyd, AWB | Multiply and Accumulate | 1 | 1 | OA,OB,OAB,SA,SB,SAB |
| | | MAC Wm*Wm, Acc, Wx, Wxd, Wy, Wyd | Square and Accumulate | 1 | 1 | OA,OB,OAB,SA,SB,SAB |
| 46 | MOV | MOV f, Wn | Move f to Wn | 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 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 | | |
| 47 | MOVSAC | MOVSAC Acc, Wx, Wxd, Wy, Wyd, AWB | Prefetch and store accumulator | 1 | 1 | None |

TABLE 29-2: INSTRUCTION SET OVERVIEW (CONTINUED)

| Base Instr # | Assembly Mnemonic | Assembly Syntax | Description | # of Words | # of Cycles | Status Flags Affected |
|--------------|-------------------|---------------------------------------|--|------------|-------------|-----------------------|
| 48 | MPY | MPY Wm*Wn, Acc, Wx, Wxd, Wy, Wyd | Multiply Wm by Wn to Accumulator | 1 | 1 | OA,OB,OAB,SA,SB,SAB |
| | | MPY Wm*Wm, Acc, Wx, Wxd, Wy, Wyd | Square Wm to Accumulator | 1 | 1 | OA,OB,OAB,SA,SB,SAB |
| 49 | MPY.N | MPY.N Wm*Wn, Acc, Wx, Wxd, Wy, Wyd | -(Multiply Wm by Wn) to Accumulator | 1 | 1 | None |
| 50 | MSC | MSC Wm*Wm, Acc, Wx, Wxd, Wy, Wyd, AWB | Multiply and Subtract from Accumulator | 1 | 1 | OA,OB,OAB,SA,SB,SAB |
| 51 | MUL | MUL.SS Wb, Ws, Wnd | {Wnd + 1, Wnd} = signed(Wb) * signed(Ws) | 1 | 1 | None |
| | | MUL.SU Wb, Ws, Wnd | {Wnd + 1, Wnd} = signed(Wb) * unsigned(Ws) | 1 | 1 | None |
| | | MUL.US Wb, Ws, Wnd | {Wnd + 1, Wnd} = unsigned(Wb) * signed(Ws) | 1 | 1 | None |
| | | MUL.UU Wb, Ws, Wnd | {Wnd + 1, Wnd} = unsigned(Wb) * unsigned(Ws) | 1 | 1 | None |
| | | MUL.SU Wb, #lit5, Wnd | {Wnd + 1, Wnd} = signed(Wb) * unsigned(lit5) | 1 | 1 | None |
| | | MUL.UU Wb, #lit5, Wnd | {Wnd + 1, Wnd} = unsigned(Wb) * unsigned(lit5) | 1 | 1 | None |
| 52 | NEG | NEG Acc | Negate Accumulator | 1 | 1 | OA,OB,OAB,SA,SB,SAB |
| | | 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 | $Wd = \bar{Ws} + 1$ | 1 | 1 | C,DC,N,OV,Z |
| 53 | NOP | NOP | No Operation | 1 | 1 | None |
| | | NOPR | No Operation | 1 | 1 | None |
| 54 | 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 |
| 55 | 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 |
| 56 | PWRSVAV | PWRSVAV #lit1 | Go into Sleep or Idle mode | 1 | 1 | WDTO,Sleep |
| 57 | RCALL | RCALL Expr | Relative Call | 1 | 2 | None |
| | | RCALL Wn | Computed Call | 1 | 2 | None |
| 58 | REPEAT | REPEAT #lit14 | Repeat Next Instruction lit14 + 1 times | 1 | 1 | None |
| | | REPEAT Wn | Repeat Next Instruction (Wn) + 1 times | 1 | 1 | None |
| 59 | RESET | RESET | Software device Reset | 1 | 1 | None |
| 60 | RETFIE | RETFIE | Return from interrupt | 1 | 3 (2) | None |
| 61 | RETLW | RETLW #lit10, Wn | Return with literal in Wn | 1 | 3 (2) | None |
| 62 | RETURN | RETURN | Return from Subroutine | 1 | 3 (2) | None |
| 63 | 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 |
| 64 | 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 |
| 65 | 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 |
| 66 | 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 |

TABLE 29-2: INSTRUCTION SET OVERVIEW (CONTINUED)

| Base Instr # | Assembly Mnemonic | Assembly Syntax | Description | # of Words | # of Cycles | Status Flags Affected |
|--------------|-------------------|-------------------------------|---|------------|-------------|-----------------------|
| 67 | SAC | SAC <i>Acc, #Slit4, Wdo</i> | Store Accumulator | 1 | 1 | None |
| | | SAC.R <i>Acc, #Slit4, Wdo</i> | Store Rounded Accumulator | 1 | 1 | None |
| 68 | SE | SE <i>Ws, Wnd</i> | Wnd = sign-extended Ws | 1 | 1 | C,N,Z |
| 69 | SETM | SETM <i>f</i> | $f = 0xFFFF$ | 1 | 1 | None |
| | | SETM <i>WREG</i> | WREG = 0xFFFF | 1 | 1 | None |
| | | SETM <i>Ws</i> | Ws = 0xFFFF | 1 | 1 | None |
| 70 | SFTAC | SFTAC <i>Acc, Wn</i> | Arithmetic Shift Accumulator by (Wn) | 1 | 1 | OA,OB,OAB,SA,SB,SAB |
| | | SFTAC <i>Acc, #Slit6</i> | Arithmetic Shift Accumulator by Slit6 | 1 | 1 | OA,OB,OAB,SA,SB,SAB |
| 71 | SL | SL <i>f</i> | $f = \text{Left Shift } f$ | 1 | 1 | C,N,OV,Z |
| | | SL <i>f, WREG</i> | WREG = Left Shift <i>f</i> | 1 | 1 | C,N,OV,Z |
| | | SL <i>Ws, Wd</i> | Wd = Left Shift Ws | 1 | 1 | C,N,OV,Z |
| | | SL <i>Wb, Wns, Wnd</i> | Wnd = Left Shift Wb by Wns | 1 | 1 | N,Z |
| | | SL <i>Wb, #lit5, Wnd</i> | Wnd = Left Shift Wb by lit5 | 1 | 1 | N,Z |
| 72 | SUB | SUB <i>Acc</i> | Subtract Accumulators | 1 | 1 | OA,OB,OAB,SA,SB,SAB |
| | | SUB <i>f</i> | $f = f - WREG$ | 1 | 1 | C,DC,N,OV,Z |
| | | SUB <i>f, WREG</i> | WREG = $f - WREG$ | 1 | 1 | C,DC,N,OV,Z |
| | | SUB <i>#lit10, Wn</i> | $Wn = Wn - \text{lit10}$ | 1 | 1 | C,DC,N,OV,Z |
| | | SUB <i>Wb, Ws, Wd</i> | $Wd = Wb - Ws$ | 1 | 1 | C,DC,N,OV,Z |
| | | SUB <i>Wb, #lit5, Wd</i> | $Wd = Wb - \text{lit5}$ | 1 | 1 | C,DC,N,OV,Z |
| 73 | SUBB | SUBB <i>f</i> | $f = f - WREG - (\overline{C})$ | 1 | 1 | C,DC,N,OV,Z |
| | | SUBB <i>f, WREG</i> | WREG = $f - WREG - (\overline{C})$ | 1 | 1 | C,DC,N,OV,Z |
| | | SUBB <i>#lit10, Wn</i> | $Wn = Wn - \text{lit10} - (\overline{C})$ | 1 | 1 | C,DC,N,OV,Z |
| | | SUBB <i>Wb, Ws, Wd</i> | $Wd = Wb - Ws - (\overline{C})$ | 1 | 1 | C,DC,N,OV,Z |
| | | SUBB <i>Wb, #lit5, Wd</i> | $Wd = Wb - \text{lit5} - (\overline{C})$ | 1 | 1 | C,DC,N,OV,Z |
| 74 | SUBR | SUBR <i>f</i> | $f = WREG - f$ | 1 | 1 | C,DC,N,OV,Z |
| | | SUBR <i>f, WREG</i> | WREG = $WREG - f$ | 1 | 1 | C,DC,N,OV,Z |
| | | SUBR <i>Wb, Ws, Wd</i> | $Wd = Ws - Wb$ | 1 | 1 | C,DC,N,OV,Z |
| | | SUBR <i>Wb, #lit5, Wd</i> | $Wd = \text{lit5} - Wb$ | 1 | 1 | C,DC,N,OV,Z |
| 75 | SUBBR | SUBBR <i>f</i> | $f = WREG - f - (\overline{C})$ | 1 | 1 | C,DC,N,OV,Z |
| | | SUBBR <i>f, WREG</i> | WREG = $WREG - f - (\overline{C})$ | 1 | 1 | C,DC,N,OV,Z |
| | | SUBBR <i>Wb, Ws, Wd</i> | $Wd = Ws - Wb - (\overline{C})$ | 1 | 1 | C,DC,N,OV,Z |
| | | SUBBR <i>Wb, #lit5, Wd</i> | $Wd = \text{lit5} - Wb - (\overline{C})$ | 1 | 1 | C,DC,N,OV,Z |
| 76 | SWAP | SWAP.b <i>Wn</i> | Wn = nibble swap Wn | 1 | 1 | None |
| | | SWAP <i>Wn</i> | Wn = byte swap Wn | 1 | 1 | None |
| 77 | TBLRDH | TBLRDH <i>Ws, Wd</i> | Read Prog<23:16> to Wd<7:0> | 1 | 2 | None |
| 78 | TBLRDL | TBLRDL <i>Ws, Wd</i> | Read Prog<15:0> to Wd | 1 | 2 | None |
| 79 | TBLWTH | TBLWTH <i>Ws, Wd</i> | Write Ws<7:0> to Prog<23:16> | 1 | 2 | None |
| 80 | TBLWTL | TBLWTL <i>Ws, Wd</i> | Write Ws to Prog<15:0> | 1 | 2 | None |
| 81 | ULNK | ULNK | Unlink Frame Pointer | 1 | 1 | None |
| 82 | 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> | $Wd = \text{lit10} .XOR. Wd$ | 1 | 1 | N,Z |
| | | XOR <i>Wb, Ws, Wd</i> | $Wd = Wb .XOR. Ws$ | 1 | 1 | N,Z |
| | | XOR <i>Wb, #lit5, Wd</i> | $Wd = Wb .XOR. \text{lit5}$ | 1 | 1 | N,Z |
| 83 | ZE | ZE <i>Ws, Wnd</i> | Wnd = Zero-extend Ws | 1 | 1 | C,Z,N |

30.0 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

30.1 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.

30.2 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

30.3 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.

30.4 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

30.5 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

30.6 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.

30.7 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.

30.8 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.

30.9 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.

30.10 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.

30.11 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.

30.12 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.

30.13 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) for the complete list of demonstration, development and evaluation kits.

31.0 ELECTRICAL CHARACTERISTICS

This section provides an overview of dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 electrical characteristics. Additional information will be provided in future revisions of this document as it becomes available.

Absolute maximum ratings for the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 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⁽¹⁾

| | |
|---|-----------------------|
| Ambient temperature under bias | -40°C to +125°C |
| Storage temperature | -65°C to +150°C |
| Voltage on VDD with respect to VSS | -0.3V to +4.0V |
| Voltage on any combined analog and digital pin and $\overline{\text{MCLR}}$, with respect to VSS | -0.3V to (VDD + 0.3V) |
| Voltage on any digital-only pin with respect to VSS | -0.3V to +5.6V |
| Voltage on VCAP/VDDCORE with respect to VSS | 2.25V to 2.75V |
| Maximum current out of VSS pin | 300 mA |
| Maximum current into VDD pin ⁽²⁾ | 250 mA |
| Maximum output current sunk by any I/O pin ⁽³⁾ | 4 mA |
| Maximum output current sourced by any I/O pin ⁽³⁾ | 4 mA |
| Maximum current sunk by all ports | 200 mA |
| Maximum current sourced by all ports ⁽²⁾ | 200 mA |

Note 1: 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.

2: Maximum allowable current is a function of device maximum power dissipation (see Table 31-2).

3: Exceptions are CLKOUT, which is able to sink/source 25 mA, and the VREF+, VREF-, SCLx, SDAx, PGECx and PGEDx pins, which are able to sink/source 12 mA.

31.1 DC Characteristics

TABLE 31-1: OPERATING MIPS VS. VOLTAGE

| Characteristic | VDD Range (in Volts) | Temp Range (in °C) | Max MIPS |
|----------------|-------------------------|-----------------------|---|
| | | | dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 |
| | 3.0-3.6V | -40°C to +85°C | 40 |
| | 3.0-3.6V | -40°C to +125°C | 40 |

TABLE 31-2: THERMAL OPERATING CONDITIONS

| Rating | Symbol | Min | Typ | Max | Unit |
|--|--------|---------------------------|-----|------|------|
| Industrial Temperature Devices | | | | | |
| Operating Junction Temperature Range | TJ | -40 | — | +125 | °C |
| Operating Ambient Temperature Range | TA | -40 | — | +85 | °C |
| Extended Temperature Devices | | | | | |
| Operating Junction Temperature Range | TJ | -40 | — | +140 | °C |
| Operating Ambient Temperature Range | TA | -40 | — | +125 | °C |
| Power Dissipation: Internal chip power dissipation: $P_{INT} = V_{DD} \times (I_{DD} - \sum I_{OH})$ I/O Pin Power Dissipation: $I/O = \sum (\{V_{DD} - V_{OH}\} \times I_{OH}) + \sum (V_{OL} \times I_{OL})$ | PD | PINT + PI/O | | | W |
| Maximum Allowed Power Dissipation | PDMAX | $(T_J - T_A)/\theta_{JA}$ | | | W |

TABLE 31-3: THERMAL PACKAGING CHARACTERISTICS

| Characteristic | Symbol | Typ | Max | Unit | Notes |
|--|---------------|-----|-----|------|-------|
| Package Thermal Resistance, 44-pin QFN | θ_{JA} | 30 | — | °C/W | 1 |
| Package Thermal Resistance, 44-pin TFQP | θ_{JA} | 40 | — | °C/W | 1 |
| Package Thermal Resistance, 28-pin SPDIP | θ_{JA} | 45 | — | °C/W | 1 |
| Package Thermal Resistance, 28-pin SOIC | θ_{JA} | 50 | — | °C/W | 1 |
| Package Thermal Resistance, 28-pin QFN-S | θ_{JA} | 30 | — | °C/W | 1 |

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

TABLE 31-4: DC TEMPERATURE AND VOLTAGE SPECIFICATIONS

| DC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | |
|--------------------------|-----------------------|---|---|--------------------|------|-------|---|
| Param No. | Symbol | Characteristic | Min | Typ ⁽¹⁾ | Max | Units | Conditions |
| Operating Voltage | | | | | | | |
| DC10 | Supply Voltage | | | | | | |
| | VDD | | 3.0 | — | 3.6 | V | Industrial and Extended |
| DC12 | VDR | RAM Data Retention Voltage⁽²⁾ | 1.8 | — | — | V | |
| DC16 | VPOR | VDD Start Voltage⁽⁴⁾ to ensure internal Power-on Reset signal | — | — | VSS | V | |
| DC17 | SVDD | VDD Rise Rate to ensure internal Power-on Reset signal | 0.03 | — | — | V/ms | 0-3.0V in 0.1s |
| DC18 | VCORE | VDD Core⁽³⁾ Internal regulator voltage | 2.25 | — | 2.75 | V | Voltage is dependent on load, temperature and VDD |

- Note 1:** Data in “Typ” column is at 3.3V, 25°C unless otherwise stated.
Note 2: This is the limit to which VDD may be lowered without losing RAM data.
Note 3: These parameters are characterized but not tested in manufacturing.
Note 4: VDD voltage must remain at VSS for a minimum of 200 μs to ensure POR.

TABLE 31-5: DC CHARACTERISTICS: OPERATING CURRENT (IDD)

| DC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | |
|--|------------------------|-----|---|------------|-----------------|
| Parameter No. | Typical ⁽¹⁾ | Max | Units | Conditions | |
| Operating Current (IDD)⁽²⁾ | | | | | |
| DC20d | 19 | 30 | mA | -40°C | 3.3V 10 MIPS |
| DC20a | 19 | 30 | mA | +25°C | |
| DC20b | 19 | 30 | mA | +85°C | |
| DC20c | 19 | 35 | mA | +125°C | |
| DC21d | 29 | 40 | mA | -40°C | 3.3V 16 MIPS |
| DC21a | 29 | 40 | mA | +25°C | |
| DC21b | 28 | 45 | mA | +85°C | |
| DC21c | 28 | 45 | mA | +125°C | |
| DC22d | 33 | 50 | mA | -40°C | 3.3V 20 MIPS |
| DC22a | 33 | 50 | mA | +25°C | |
| DC22b | 33 | 55 | mA | +85°C | |
| DC22c | 33 | 55 | mA | +125°C | |
| DC23d | 47 | 70 | mA | -40°C | 3.3V 30 MIPS |
| DC23a | 48 | 70 | mA | +25°C | |
| DC23b | 48 | 70 | mA | +85°C | |
| DC23c | 48 | 70 | mA | +125°C | |
| DC24d | 60 | 90 | mA | -40°C | 3.3V 40 MIPS |
| DC24a | 60 | 90 | mA | +25°C | |
| DC24b | 60 | 90 | mA | +85°C | |
| DC24c | 60 | 90 | mA | +125°C | |

Note 1: Data in "Typical" column is at 3.3V, 25°C unless otherwise stated.

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: OSC1 driven with external square wave from rail to rail. All I/O pins are configured as inputs and pulled to Vss. MCLR = VDD, WDT and FSCM are disabled. CPU, SRAM, program memory and data memory are operational. No peripheral modules are operating; however, every peripheral is being clocked (PMD bits are all zeroed).

TABLE 31-6: DC CHARACTERISTICS: IDLE CURRENT (IDLE)

| DC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | |
|--|------------------------|-----|---|------------|-----------------|
| Parameter No. | Typical ⁽¹⁾ | Max | Units | Conditions | |
| Idle Current (IDLE): Core OFF Clock ON Base Current⁽²⁾ | | | | | |
| DC40d | 4 | 25 | mA | -40°C | 3.3V 10 MIPS |
| DC40a | 4 | 25 | mA | +25°C | |
| DC40b | 4 | 25 | mA | +85°C | |
| DC40c | 4 | 25 | mA | +125°C | |
| DC41d | 6 | 25 | mA | -40°C | 3.3V 16 MIPS |
| DC41a | 6 | 25 | mA | +25°C | |
| DC41b | 6 | 25 | mA | +85°C | |
| DC41c | 6 | 25 | mA | +125°C | |
| DC42d | 9 | 25 | mA | -40°C | 3.3V 20 MIPS |
| DC42a | 9 | 25 | mA | +25°C | |
| DC42b | 9 | 25 | mA | +85°C | |
| DC42c | 9 | 25 | mA | +125°C | |
| DC43d | 16 | 25 | mA | -40°C | 3.3V 30 MIPS |
| DC43a | 16 | 25 | mA | +25°C | |
| DC43b | 16 | 25 | mA | +85°C | |
| DC43c | 16 | 25 | mA | +125°C | |
| DC44d | 18 | 25 | mA | -40°C | 3.3V 40 MIPS |
| DC44a | 18 | 25 | mA | +25°C | |
| DC44b | 19 | 25 | mA | +85°C | |
| DC44c | 19 | 25 | mA | +125°C | |

Note 1: Data in “Typical” column is at 3.3V, 25°C unless otherwise stated.

2: Base IDLE current is measured with core off, clock on and all modules turned off. Peripheral Module Disable SFR registers are zeroed. All I/O pins are configured as inputs and pulled to Vss.

TABLE 31-7: DC CHARACTERISTICS: POWER-DOWN CURRENT (IPD)

| DC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | |
|---|------------------------|------|---|------------|---|
| Parameter No. | Typical ⁽¹⁾ | Max | Units | Conditions | |
| Power-Down Current (IPD)⁽²⁾ | | | | | |
| DC60d | 24 | 500 | μA | -40°C | 3.3V Base Power-Down Current ^(3,4) |
| DC60a | 28 | 500 | μA | +25°C | |
| DC60b | 124 | 750 | μA | +85°C | |
| DC60c | 350 | 1000 | μA | +125°C | |
| DC61d | 8 | 13 | μA | -40°C | 3.3V Watchdog Timer Current: ΔI _{WDT} ⁽³⁾ |
| DC61a | 10 | 15 | μA | +25°C | |
| DC61b | 12 | 20 | μA | +85°C | |
| DC61c | 13 | 25 | μA | +125°C | |

Note 1: Data in the Typical column is at 3.3V, 25°C unless otherwise stated.

Note 2: Base IPD is measured with all peripherals and clocks shut down. All I/Os are configured as inputs and pulled to V_{SS}. WDT, etc., are all switched off and VREGS (RCON<8>) = 1.

Note 3: The Δ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

Note 4: These currents are measured on the device containing the most memory in this family.

TABLE 31-8: DC CHARACTERISTICS: DOZE CURRENT (IDOZE)

| DC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | |
|--------------------|------------------------|-----|---|-------|---------------------|
| Parameter No. | Typical ⁽¹⁾ | Max | Doze Ratio | Units | Conditions |
| DC73a | 42 | 50 | 1:2 | mA | -40°C 3.3V 40 MIPS |
| DC73f | 23 | 30 | 1:64 | mA | |
| DC73g | 23 | 30 | 1:128 | mA | |
| DC70a | 42 | 50 | 1:2 | mA | +25°C 3.3V 40 MIPS |
| DC70f | 26 | 30 | 1:64 | mA | |
| DC70g | 25 | 30 | 1:128 | mA | |
| DC71a | 41 | 50 | 1:2 | mA | +85°C 3.3V 40 MIPS |
| DC71f | 25 | 30 | 1:64 | mA | |
| DC71g | 24 | 30 | 1:128 | mA | |
| DC72a | 42 | 50 | 1:2 | mA | +125°C 3.3V 40 MIPS |
| DC72f | 26 | 30 | 1:64 | mA | |
| DC72g | 25 | 30 | 1:128 | mA | |

Note 1: Data in the Typical column is at 3.3V, 25°C unless otherwise stated.

TABLE 31-9: DC CHARACTERISTICS: I/O PIN INPUT SPECIFICATIONS

| DC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|--------------------|-------------------|--|---|--------------------|----------------------|-------|---|
| Param No. | Symbol | Characteristic | Min | Typ ⁽¹⁾ | Max | Units | Conditions |
| | V _{IL} | Input Low Voltage | | | | | |
| DI10 | | I/O pins | V _{SS} | — | 0.2 V _{DD} | V | PMP _{TTL} = 1 |
| DI11 | | PMP pins | V _{SS} | — | 0.15 V _{DD} | V | |
| DI15 | | <u>MCLR</u> | V _{SS} | — | 0.2 V _{DD} | V | |
| DI16 | | I/O Pins with OSC1 or SOSC1 | V _{SS} | — | 0.2 V _{DD} | V | |
| DI18 | | I/O Pins with SDAx, SCLx | V _{SS} | — | 0.3 V _{DD} | V | |
| DI19 | | I/O Pins with SDAx, SCLx | V _{SS} | — | 0.2 V _{DD} | V | SMbus enabled |
| | V _{IH} | Input High Voltage | | | | | |
| DI20 | | I/O Pins Not 5V Tolerant ⁽⁴⁾ | 0.8 V _{DD} | — | V _{DD} | V | SMbus disabled |
| | | I/O Pins 5V Tolerant ⁽⁴⁾ | 0.8 V _{DD} | — | 5.5 | V | |
| DI21 | | I/O Pins Not 5V Tolerant with PMP ⁽⁴⁾ | 0.24 V _{DD} + 0.8 | — | V _{DD} | V | |
| | | I/O Pins 5V Tolerant with PMP ⁽⁴⁾ | 0.24 V _{DD} + 0.8 | — | 5.5 | V | |
| DI26 | | I/O Pins with OSC1 or SOSC1 | 0.7 V _{DD} | — | V _{DD} | V | |
| DI28 | | I/O Pins with SDAx, SCLx | 0.7 V _{DD} | — | 5.5 | V | |
| DI29 | | I/O Pins with SDAx, SCLx | 0.8 V _{DD} | — | 5.5 | V | SMbus enabled |
| | IC _{NPU} | CNx Pull-up Current | | | | | |
| DI30 | | | 50 | 250 | 400 | μA | V _{DD} = 3.3V, V _{PIN} = V _{SS} |
| | I _{IL} | Input Leakage Current^(2,3) | | | | | |
| DI50 | | I/O pins 5V Tolerant ⁽⁴⁾ | — | — | ±2 | μA | V _{SS} ≤ V _{PIN} ≤ V _{DD} , Pin at high-impedance, 40°C ≤ TA ≤ +85°C |
| DI51 | | I/O Pins Not 5V Tolerant ⁽⁴⁾ | — | — | ±1 | μA | V _{SS} ≤ V _{PIN} ≤ V _{DD} , Pin at high-impedance, 40°C ≤ TA ≤ +85°C |
| DI51a | | I/O Pins Not 5V Tolerant ⁽⁴⁾ | — | — | ±2 | μA | Shared with external reference pins, 40°C ≤ TA ≤ +85°C |
| DI51b | | I/O Pins Not 5V Tolerant ⁽⁴⁾ | — | — | ±3.5 | μA | V _{SS} ≤ V _{PIN} ≤ V _{DD} , Pin at high-impedance, -40°C ≤ TA ≤ +125°C |
| DI51c | | I/O Pins Not 5V Tolerant ⁽⁴⁾ | — | — | ±8 | μA | Analog pins shared with external reference pins, -40°C ≤ TA ≤ +125°C |
| DI55 | | <u>MCLR</u> | — | — | ±2 | μA | V _{SS} ≤ V _{PIN} ≤ V _{DD} |
| DI56 | | OSC1 | — | — | ±2 | μA | V _{SS} ≤ V _{PIN} ≤ V _{DD} , XT and HS modes |

Note 1: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated.

2: The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current can be measured at different input voltages.

3: Negative current is defined as current sourced by the pin.

4: See “Pin Diagrams” for the 5V tolerant I/O pins.

TABLE 31-10: DC CHARACTERISTICS: I/O PIN OUTPUT SPECIFICATIONS

| DC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|--------------------|--------|----------------------------|---|-----|-----|-------|---|
| Param No. | Symbol | Characteristic | Min | Typ | Max | Units | Conditions |
| DO10 DO16 | VOL | Output Low Voltage | | | | | |
| | | I/O ports | — | — | 0.4 | V | IO _L = 2 mA, V _{DD} = 3.3V |
| | | OSC2/CLKO | — | — | 0.4 | V | IO _L = 2 mA, V _{DD} = 3.3V |
| DO20 DO26 | VOH | Output High Voltage | | | | | |
| | | I/O ports | 2.40 | — | — | V | IO _H = -2.3 mA, V _{DD} = 3.3V |
| | | OSC2/CLKO | 2.41 | — | — | V | IO _H = -1.3 mA, V _{DD} = 3.3V |

TABLE 31-11: ELECTRICAL CHARACTERISTICS: BOR

| DC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|--------------------|--------|---|---|-----|------|-------|------------|
| Param No. | Symbol | Characteristic | Min ⁽¹⁾ | Typ | Max | Units | Conditions |
| BO10 | VBOR | BOR Event on V _{DD} transition high-to-low BOR event is tied to V _{DD} core voltage decrease | 2.40 | — | 2.55 | V | |

Note 1: Parameters are for design guidance only and are not tested in manufacturing.

TABLE 31-12: DC CHARACTERISTICS: PROGRAM MEMORY

| DC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|-----------------------------|--------|-----------------------------------|---|--------------------|------|-------|--|
| Param No. | Symbol | Characteristic | Min | Typ ⁽¹⁾ | Max | Units | Conditions |
| Program Flash Memory | | | | | | | |
| D130 | EP | Cell Endurance | 10,000 | — | — | E/W | -40°C to +125°C |
| D131 | VPR | VDD for Read | V _{MIN} | — | 3.6 | V | V _{MIN} = Minimum operating voltage |
| D132B | VPEW | VDD for Self-Timed Write | V _{MIN} | — | 3.6 | V | V _{MIN} = Minimum operating voltage |
| D134 | TRETD | Characteristic Retention | 20 | — | — | Year | Provided no other specifications are violated, -40°C to +125°C |
| D135 | IDDP | Supply Current during Programming | — | 10 | — | mA | |
| D136a | TRW | Row Write Time | 1.32 | — | 1.74 | ms | TRW = 11064 FRC cycles, TA = +85°C, See Note 2 |
| D136b | TRW | Row Write Time | 1.28 | — | 1.79 | ms | TRW = 11064 FRC cycles, TA = +125°C, See Note 2 |
| D137a | TPE | Page Erase Time | 20.1 | — | 26.5 | ms | TPE = 168517 FRC cycles, TA = +85°C, See Note 2 |
| D137b | TPE | Page Erase Time | 19.5 | — | 27.3 | ms | TPE = 168517 FRC cycles, TA = +125°C, See Note 2 |
| D138a | TWW | Word Write Cycle Time | 42.3 | — | 55.9 | μs | TWW = 355 FRC cycles, TA = +85°C, See Note 2 |
| D138b | TWW | Word Write Cycle Time | 41.1 | — | 57.6 | μs | TWW = 355 FRC cycles, TA = +125°C, See Note 2 |

Note 1: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated.

- 2:** Other conditions: FRC = 7.37 MHz, TUN<5:0> = b'0111111 (for Min), TUN<5:0> = b'100000 (for Max). This parameter depends on the FRC accuracy (see Table 31-19) and the value of the FRC Oscillator Tuning register (see Register 9-4). For complete details on calculating the Minimum and Maximum time see **Section 5.3 “Programming Operations”**.

TABLE 31-13: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

| Standard Operating Conditions (unless otherwise stated): Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | | | | |
|---|--------|---------------------------------|-----|-----|-----|-------|--|
| Param No. | Symbol | Characteristics | Min | Typ | Max | Units | Comments |
| | CEFC | External Filter Capacitor Value | 4.7 | 10 | — | μF | Capacitor must be low series resistance (< 5 ohms) |

31.2 AC Characteristics and Timing Parameters

This section defines dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 AC characteristics and timing parameters.

TABLE 31-14: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

| | |
|---------------------------|---|
| AC CHARACTERISTICS | <p>Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)</p> <p>Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended</p> <p>Operating voltage V_{DD} range as described in Section 31.0 “Electrical Characteristics”.</p> |
|---------------------------|---|

FIGURE 31-1: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS

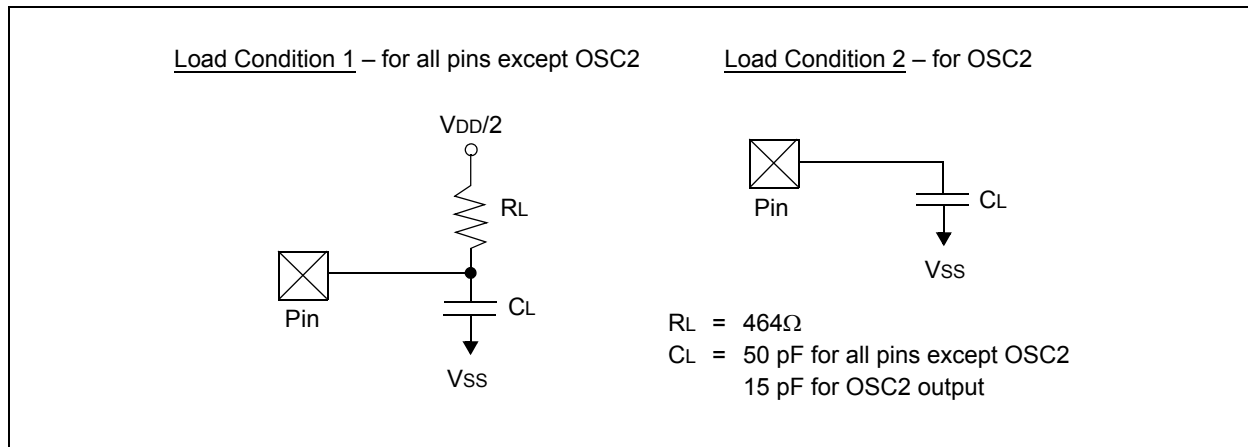


TABLE 31-15: CAPACITIVE LOADING REQUIREMENTS ON OUTPUT PINS

| Param No. | Symbol | Characteristic | Min | Typ | Max | Units | Conditions |
|-----------|--------|-----------------------|-----|-----|-----|-------|--|
| DO50 | Cosco | OSC2/SOSCO pin | — | — | 15 | pF | In XT and HS modes when external clock is used to drive OSC1 |
| DO56 | Cio | All I/O pins and OSC2 | — | — | 50 | pF | EC mode |
| DO58 | CB | SCLx, SDAx | — | — | 400 | pF | In I ² C™ mode |

FIGURE 31-2: EXTERNAL CLOCK TIMING

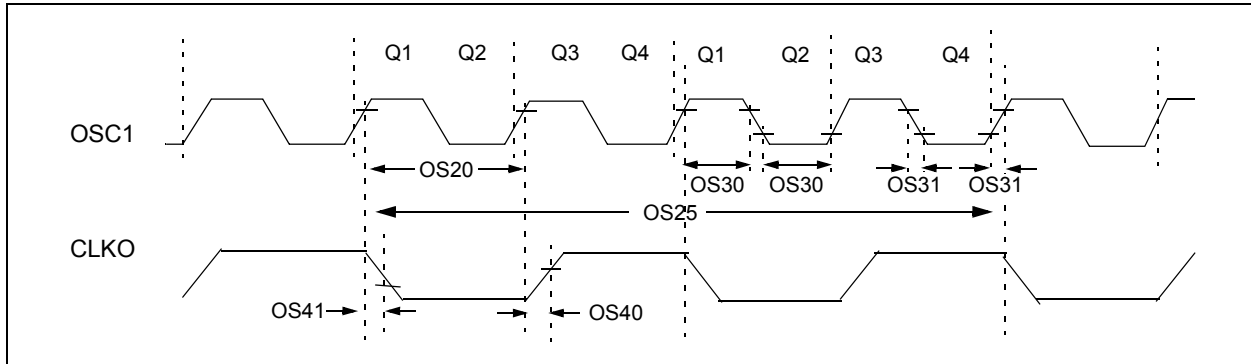


TABLE 31-16: EXTERNAL CLOCK TIMING REQUIREMENTS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | |
|--------------------|---------------|--|---|--------------------|------------------------|-------------------|--|
| Param No. | Symb | Characteristic | Min | Typ ⁽¹⁾ | Max | Units | Conditions |
| OS10 | FIN | External CLKI Frequency (External clocks allowed only in EC and ECPLL modes) | DC | — | 40 | MHz | EC |
| | | Oscillator Crystal Frequency | 3.5 10 | — — — | 10 40 33 | MHz MHz kHz | XT HS SOSC |
| OS20 | Tosc | $T_{osc} = 1/F_{osc}$ | 12.5 | — | DC | ns | |
| OS25 | Tcy | Instruction Cycle Time ⁽²⁾ | 25 | — | DC | ns | |
| OS30 | TosL, TosH | External Clock in (OSC1) High or Low Time | $0.375 \times T_{osc}$ | — | $0.625 \times T_{osc}$ | ns | EC |
| OS31 | TosR, TosF | External Clock in (OSC1) Rise or Fall Time | — | — | 20 | ns | EC |
| OS40 | TckR | CLKO Rise Time ⁽³⁾ | — | 5.2 | — | ns | |
| OS41 | TckF | CLKO Fall Time ⁽³⁾ | — | 5.2 | — | ns | |
| OS42 | GM | External Oscillator Transconductance ⁽⁴⁾ | 14 | 16 | 18 | mA/V | $V_{DD} = 3.3V$ $T_A = +25^{\circ}\text{C}$ |

Note 1: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated.

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 OSC1/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 OSC2 pin.

4: Data for this parameter is Preliminary. This parameter is characterized, but not tested in manufacturing.

TABLE 31-17: PLL CLOCK TIMING SPECIFICATIONS (V_{DD} = 3.0V TO 3.6V)

| AC CHARACTERISTICS | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | | |
|--------------------|--------|--|-----|--------------------|-----|-------|-----------------------------|
| Param No. | Symbol | Characteristic | Min | Typ ⁽¹⁾ | Max | Units | Conditions |
| OS50 | FPLLI | PLL Voltage Controlled Oscillator (VCO) Input Frequency Range | 0.8 | — | 8 | MHz | ECPLL, XTPLL modes |
| OS51 | FSYS | On-Chip VCO System Frequency | 100 | — | 200 | MHz | |
| OS52 | TLOCK | PLL Start-up Time (Lock Time) | 0.9 | 1.5 | 3.1 | mS | |
| OS53 | DCLK | CLKO Stability (Jitter) | -3 | 0.5 | 3 | % | Measured over 100 ms period |

Note 1: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

TABLE 31-18: AC CHARACTERISTICS: INTERNAL RC ACCURACY

| AC CHARACTERISTICS | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
|---|----------------|--|-----|-----|-------|---------------------|----------------------------|
| Param No. | Characteristic | Min | Typ | Max | Units | Conditions | |
| Internal FRC Accuracy @ FRC Frequency = 7.37 MHz^(1,2) | | | | | | | |
| F20 | FRC | -2 | — | +2 | % | -40°C ≤ TA ≤ +85°C | V _{DD} = 3.0-3.6V |
| | FRC | -5 | — | +5 | % | -40°C ≤ TA ≤ +125°C | V _{DD} = 3.0-3.6V |

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

2: FRC is set to initial frequency of 7.37 MHz (±2%) at 25°C.

TABLE 31-19: INTERNAL RC ACCURACY

| AC CHARACTERISTICS | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | | |
|--|----------------|--|-----|-----|-------|---------------------|----------------------------|
| Param No. | Characteristic | Min | Typ | Max | Units | Conditions | |
| LPRC @ 32.768 kHz⁽¹⁾ | | | | | | | |
| F21 | LPRC | -20 | ±6 | +20 | % | -40°C ≤ TA ≤ +85°C | V _{DD} = 3.0-3.6V |
| | LPRC | -70 | — | +70 | % | -40°C ≤ TA ≤ +125°C | V _{DD} = 3.0-3.6V |

Note 1: Change of LPRC frequency as V_{DD} changes.

FIGURE 31-3: I/O TIMING CHARACTERISTICS

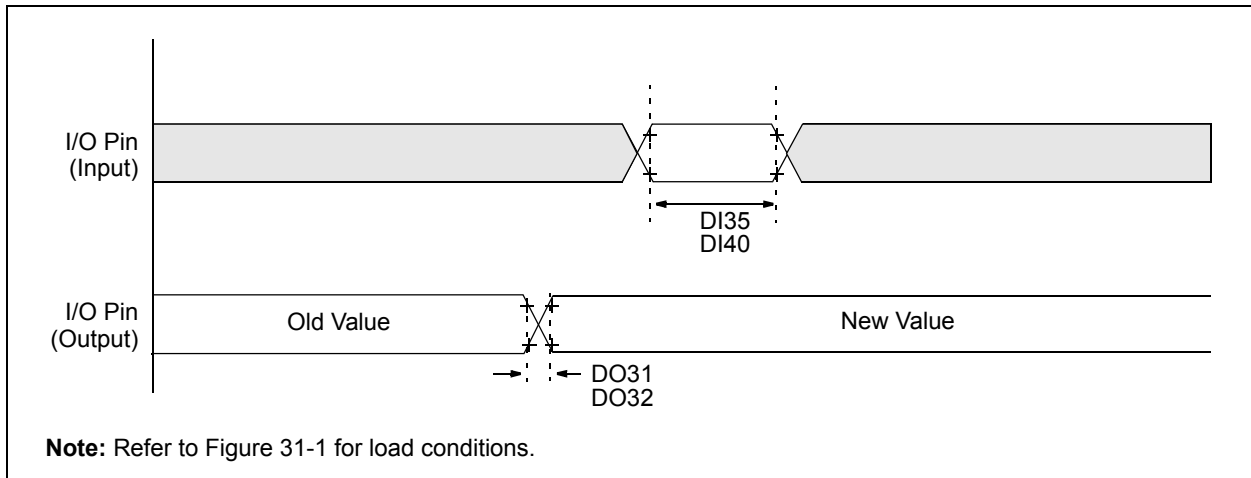


TABLE 31-20: I/O TIMING REQUIREMENTS

| AC CHARACTERISTICS | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | | |
|--------------------|--------|---|-----|--------------------|-----|-------|------------|
| Param No. | Symbol | Characteristic | Min | Typ ⁽¹⁾ | Max | Units | Conditions |
| DO31 | TioR | Port Output Rise Time | — | 10 | 25 | ns | — |
| DO32 | TioF | Port Output Fall Time | — | 10 | 25 | ns | — |
| DI35 | TINP | INTx Pin High or Low Time (output) | 20 | — | — | ns | — |
| DI40 | TRBP | CNx High or Low Time (input) | 2 | — | — | Tcy | — |

Note 1: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated.

FIGURE 31-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING CHARACTERISTICS

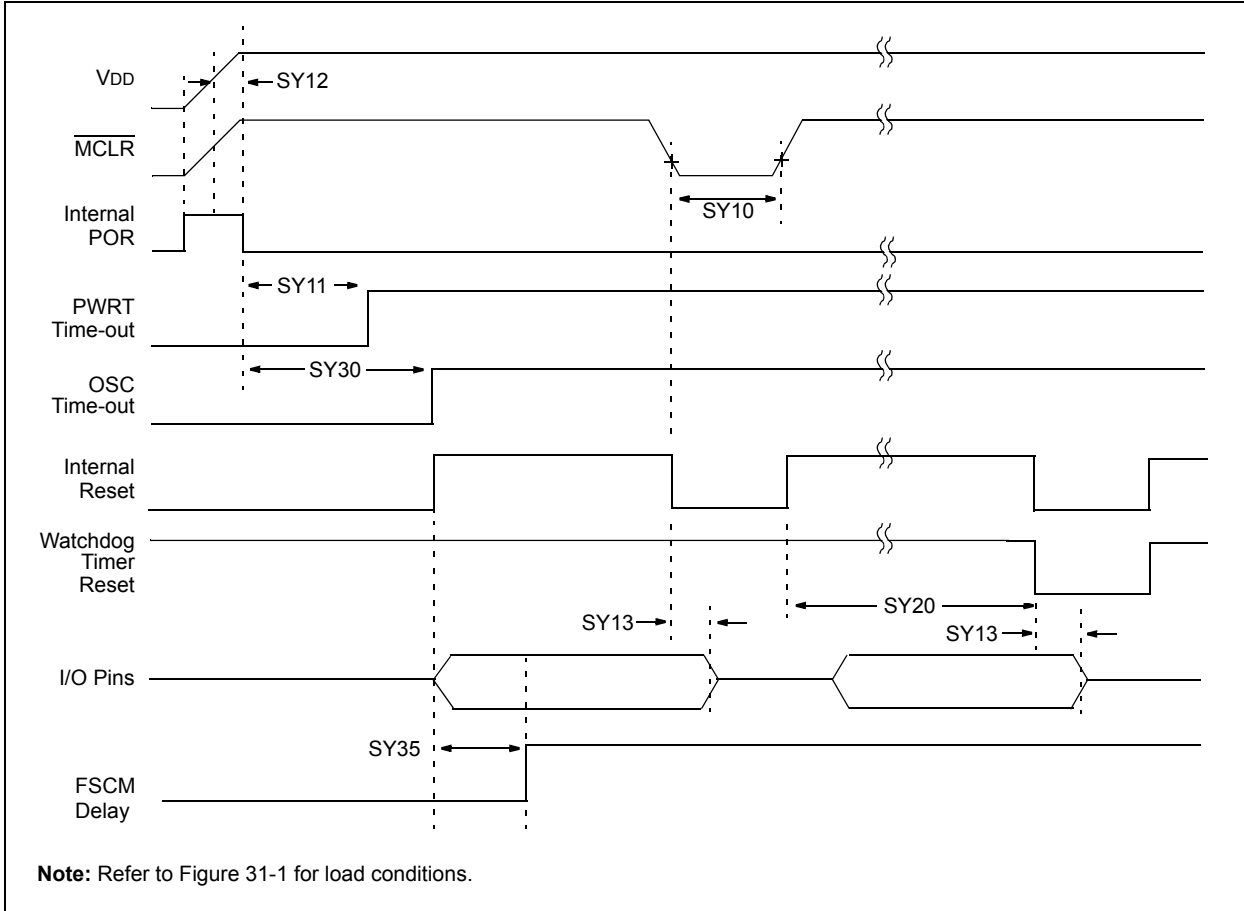


TABLE 31-21: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER TIMING REQUIREMENTS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|--------------------|--------|--|---|--------------------------------------|-----|-------|---|
| Param No. | Symbol | Characteristic ⁽¹⁾ | Min | Typ ⁽²⁾ | Max | Units | Conditions |
| SY10 | TMCL | MCLR Pulse Width (low) | 2 | — | — | μs | -40°C to +85°C |
| SY11 | TPWRT | Power-up Timer Period | — | 2 4 8 16 32 64 128 | — | ms | -40°C to +85°C User programmable |
| SY12 | TPOR | Power-on Reset Delay | 3 | 10 | 30 | μs | -40°C to +85°C |
| SY13 | TIOZ | I/O High-Impedance from MCLR Low or Watchdog Timer Reset | 0.68 | 0.72 | 1.2 | μs | |
| SY20 | TWDT1 | Watchdog Timer Time-out Period | — | — | — | — | See Section 28.4 “Watchdog Timer (WDT)” and LPRC specification F21 (Table 31-19) |
| SY30 | TOST | Oscillator Start-up Time | — | 1024 TOSC | — | — | TOSC = OSC1 period |
| SY35 | TFSCM | Fail-Safe Clock Monitor Delay | — | 500 | 900 | μs | -40°C to +85°C |

- 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.

FIGURE 31-5: TIMER1, 2 AND 3 EXTERNAL CLOCK TIMING CHARACTERISTICS

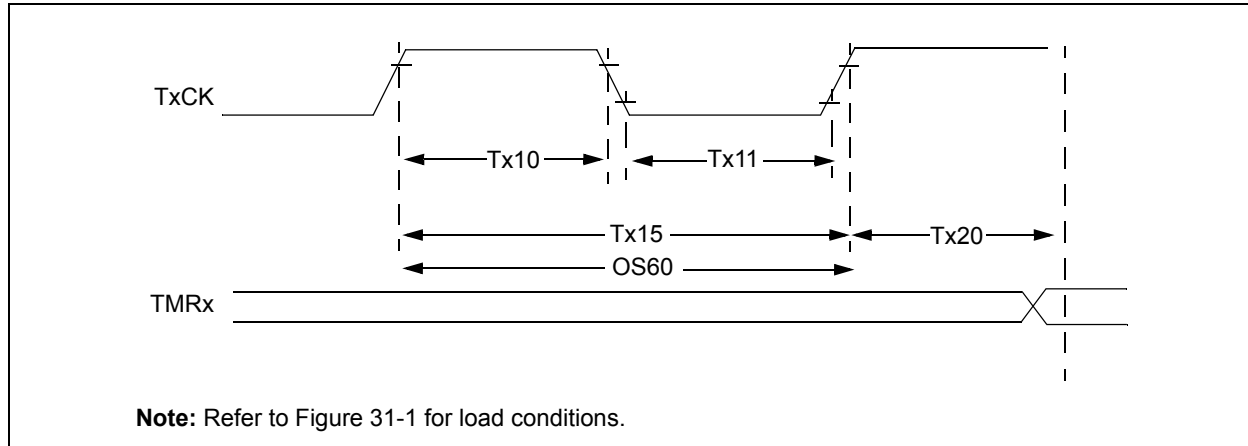


TABLE 31-22: TIMER1 EXTERNAL CLOCK TIMING REQUIREMENTS⁽¹⁾

| AC CHARACTERISTICS | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | | | |
|--------------------|-----------------------|---|-----------------------------|---|-----|---------------------|-------|------------------------------------|
| Param No. | Symbol | Characteristic | | Min | Typ | Max | Units | Conditions |
| TA10 | T _{TxH} | TxCK High Time | Synchronous, no prescaler | 0.5 T _{CY} + 20 | — | — | ns | Must also meet parameter TA15 |
| | | | Synchronous, with prescaler | 10 | — | — | ns | |
| | | | Asynchronous | 10 | — | — | ns | |
| TA11 | T _{TxL} | TxCK Low Time | Synchronous, no prescaler | 0.5 T _{CY} + 20 | — | — | ns | Must also meet parameter TA15 |
| | | | Synchronous, with prescaler | 10 | — | — | ns | |
| | | | Asynchronous | 10 | — | — | ns | |
| TA15 | T _{TxP} | TxCK Input Period | Synchronous, no prescaler | T _{CY} + 40 | — | — | ns | N = prescale value (1, 8, 64, 256) |
| | | | Synchronous, with prescaler | Greater of: 20 ns or (T _{CY} + 40)/N | — | — | — | |
| | | | Asynchronous | 20 | — | — | ns | |
| OS60 | F _{t1} | SOSCI/T1CK Oscillator Input frequency Range (oscillator enabled by setting bit TCS (T1CON<1>)) | | DC | — | 50 | kHz | |
| TA20 | T _{CKEXTMRL} | Delay from External TxCK Clock Edge to Timer Increment | | 0.5 T _{CY} | | 1.5 T _{CY} | — | |

Note 1: Timer1 is a Type A.

TABLE 31-23: TIMER2 EXTERNAL CLOCK TIMING REQUIREMENTS

| AC CHARACTERISTICS | | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | |
|--------------------|------------|--|-----------------------------|---|-----|--------------|-------|------------------------------------|
| Param No. | Symbol | Characteristic | | Min | Typ | Max | Units | Conditions |
| TB10 | TtxH | TxCK High Time | Synchronous, no prescaler | $0.5 T_{CY} + 20$ | — | — | ns | Must also meet parameter TB15 |
| | | | Synchronous, with prescaler | 10 | — | — | ns | |
| TB11 | TtxL | TxCK Low Time | Synchronous, no prescaler | $0.5 T_{CY} + 20$ | — | — | ns | Must also meet parameter TB15 |
| | | | Synchronous, with prescaler | 10 | — | — | ns | |
| TB15 | TtxP | TxCK Input Period | Synchronous, no prescaler | $T_{CY} + 40$ | — | — | ns | N = prescale value (1, 8, 64, 256) |
| | | | Synchronous, with prescaler | Greater of: 20 ns or $(T_{CY} + 40)/N$ | — | — | | |
| TB20 | TCKEXT-MRL | Delay from External TxCK Clock Edge to Timer Increment | | $0.5 T_{CY}$ | — | $1.5 T_{CY}$ | — | |

TABLE 31-24: TIMER3 EXTERNAL CLOCK TIMING REQUIREMENTS

| AC CHARACTERISTICS | | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | |
|--------------------|-----------|--|-----------------------------|---|-----|--------------|-------|------------------------------------|
| Param No. | Symbol | Characteristic | | Min | Typ | Max | Units | Conditions |
| TC10 | TtxH | TxCK High Time | Synchronous | $0.5 T_{CY} + 20$ | — | — | ns | Must also meet parameter TC15 |
| TC11 | TtxL | TxCK Low Time | Synchronous | $0.5 T_{CY} + 20$ | — | — | ns | Must also meet parameter TC15 |
| TC15 | TtxP | TxCK Input Period | Synchronous, no prescaler | $T_{CY} + 40$ | — | — | ns | N = prescale value (1, 8, 64, 256) |
| | | | Synchronous, with prescaler | Greater of: 20 ns or $(T_{CY} + 40)/N$ | — | — | | |
| TC20 | TCKEXTMRL | Delay from External TxCK Clock Edge to Timer Increment | | $0.5 T_{CY}$ | — | $1.5 T_{CY}$ | — | |

FIGURE 31-6: TIMERQ (QEI MODULE) EXTERNAL CLOCK TIMING CHARACTERISTICS

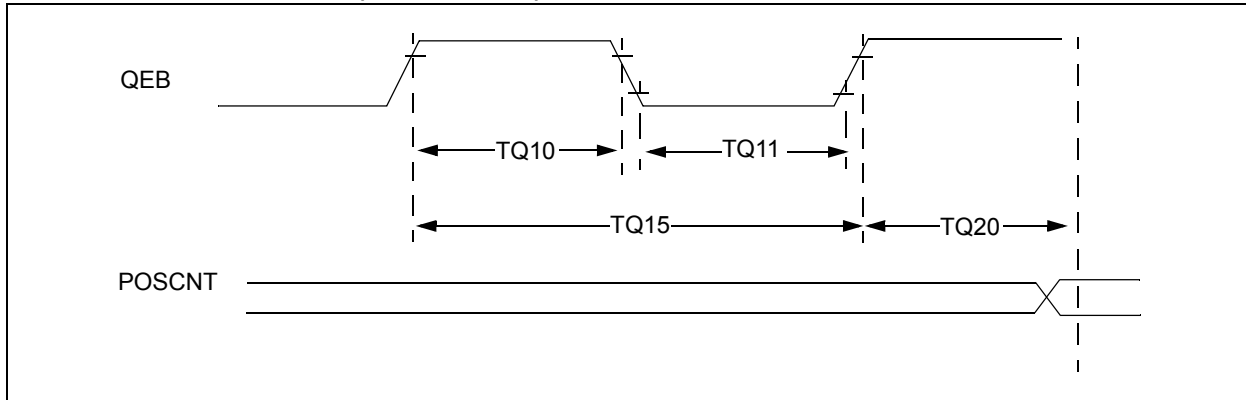


TABLE 31-25: QEI MODULE EXTERNAL CLOCK TIMING REQUIREMENTS

| AC CHARACTERISTICS | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | | | |
|--------------------|-----------|---|-----------------------------|-------------------|-----|--------------|-------|-------------------------------|
| Param No. | Symbol | Characteristic ⁽¹⁾ | | Min | Typ | Max | Units | Conditions |
| TQ10 | TtQH | TQCK High Time | Synchronous, with prescaler | $T_{CY} + 20$ | | — | ns | Must also meet parameter TQ15 |
| TQ11 | TtQL | TQCK Low Time | Synchronous, with prescaler | $T_{CY} + 20$ | | — | ns | Must also meet parameter TQ15 |
| TQ15 | TtQP | TQCP Input Period | Synchronous, with prescaler | $2 * T_{CY} + 40$ | | — | ns | — |
| TQ20 | TCKEXTMRL | Delay from External TxCK Clock Edge to Timer Increment | | $0.5 T_{CY}$ | | $1.5 T_{CY}$ | — | — |

Note 1: These parameters are characterized but not tested in manufacturing.

FIGURE 31-7: INPUT CAPTURE (CAPx) TIMING CHARACTERISTICS

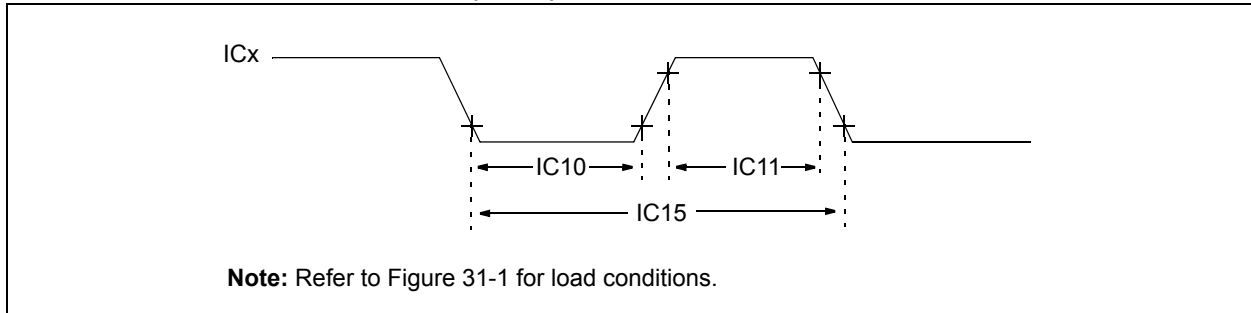


TABLE 31-26: INPUT CAPTURE TIMING REQUIREMENTS

| AC CHARACTERISTICS | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | | |
|--------------------|--------|---|----------------|-------------------|-----|-------|-------------------------------|
| Param No. | Symbol | Characteristic ⁽¹⁾ | | Min | Max | Units | Conditions |
| IC10 | TccL | ICx Input Low Time | No Prescaler | $0.5 T_{CY} + 20$ | — | ns | |
| | | | With Prescaler | 10 | — | ns | |
| IC11 | TccH | ICx Input High Time | No Prescaler | $0.5 T_{CY} + 20$ | — | ns | |
| | | | With Prescaler | 10 | — | ns | |
| IC15 | TccP | ICx Input Period | | $(T_{CY} + 40)/N$ | — | ns | N = prescale value (1, 4, 16) |

Note 1: These parameters are characterized but not tested in manufacturing.

FIGURE 31-8: OUTPUT COMPARE MODULE (OCx) TIMING CHARACTERISTICS

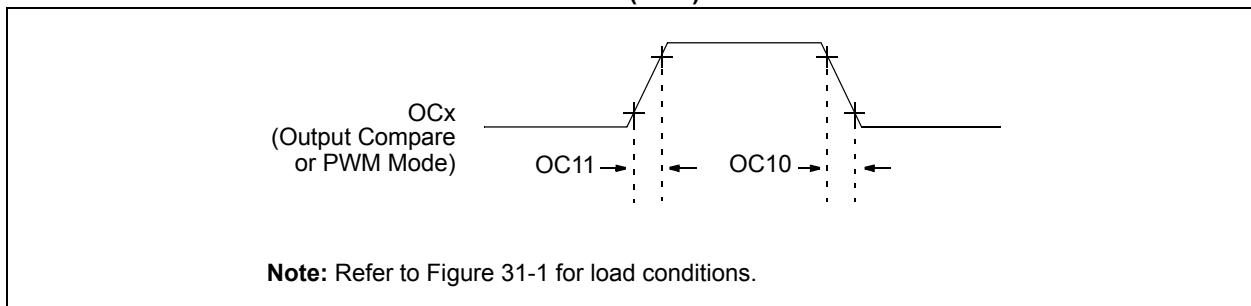


TABLE 31-27: OUTPUT COMPARE MODULE TIMING REQUIREMENTS

| AC CHARACTERISTICS | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | | |
|--------------------|--------|---|-----|-----|-----|-------|--------------------|
| Param No. | Symbol | Characteristic ⁽¹⁾ | Min | Typ | Max | Units | Conditions |
| OC10 | TccF | OCx Output Fall Time | — | — | — | ns | See parameter D032 |
| OC11 | TccR | OCx Output Rise Time | — | — | — | ns | See parameter D031 |

Note 1: These parameters are characterized but not tested in manufacturing.

FIGURE 31-9: OC/PWM MODULE TIMING CHARACTERISTICS

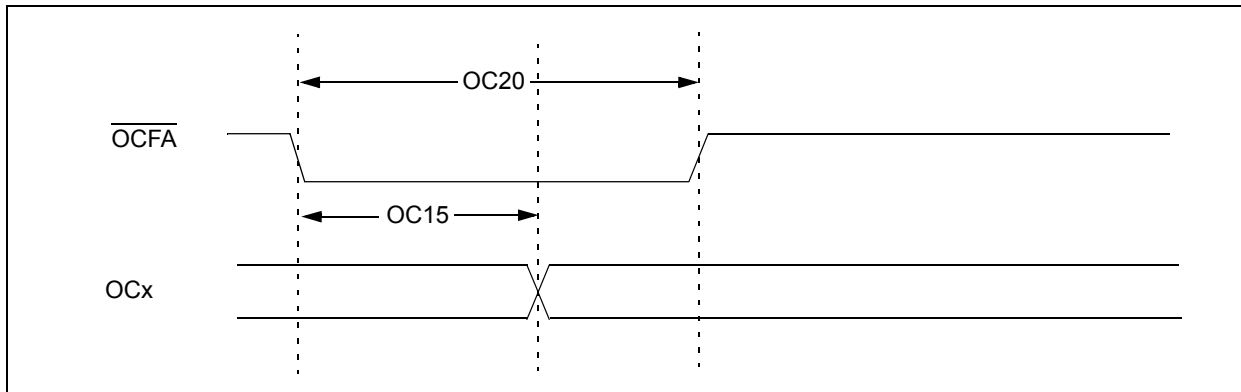


TABLE 31-28: SIMPLE OC/PWM MODE TIMING REQUIREMENTS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | |
|--------------------|------------------|-------------------------------|---|-----|-----|-------|------------|
| Param No. | Symbol | Characteristic ⁽¹⁾ | Min | Typ | Max | Units | Conditions |
| OC15 | T _{FD} | Fault Input to PWM I/O Change | — | — | 50 | ns | — |
| OC20 | T _{FLT} | Fault Input Pulse Width | 50 | — | — | ns | — |

Note 1: These parameters are characterized but not tested in manufacturing.

FIGURE 31-10: MOTOR CONTROL PWM MODULE FAULT TIMING CHARACTERISTICS

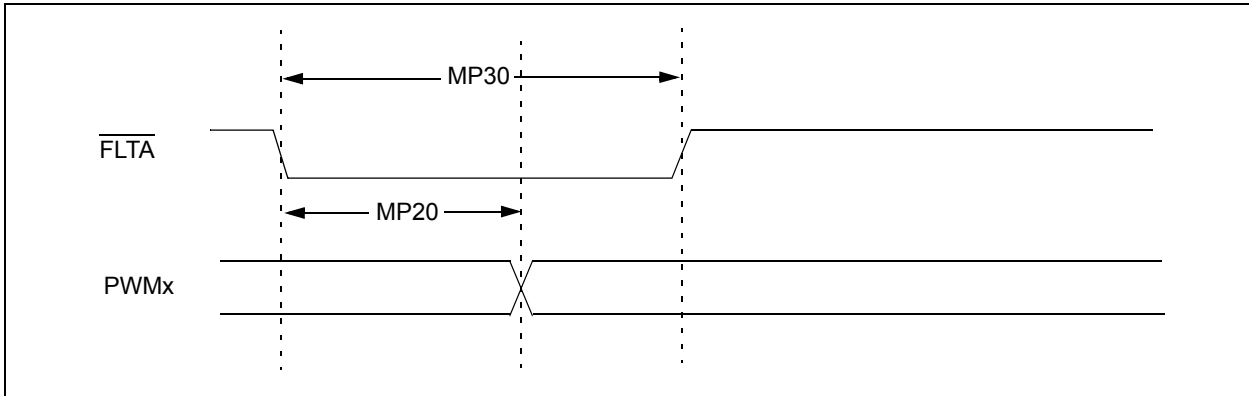


FIGURE 31-11: MOTOR CONTROL PWM MODULE TIMING CHARACTERISTICS

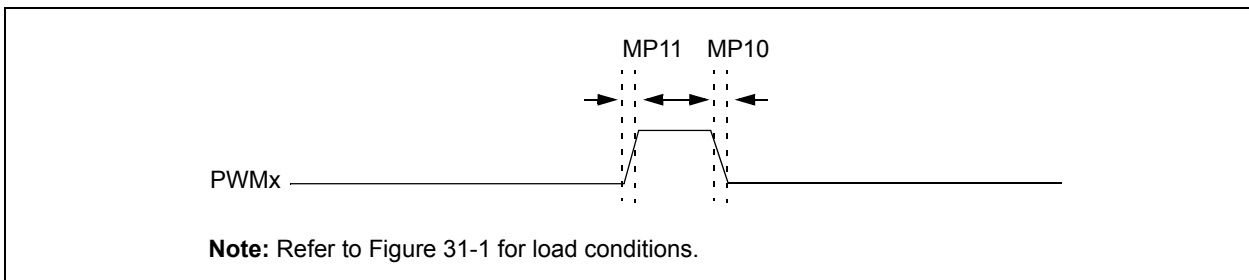


TABLE 31-29: MOTOR CONTROL PWM MODULE TIMING REQUIREMENTS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|--------------------|-------------------|---------------------------------|---|-----|-----|-------|--------------------|
| Param No. | Symbol | Characteristic ⁽¹⁾ | Min | Typ | Max | Units | Conditions |
| MP10 | T _{FPWM} | PWM Output Fall Time | — | — | — | ns | See parameter D032 |
| MP11 | T _{RPWM} | PWM Output Rise Time | — | — | — | ns | See parameter D031 |
| MP20 | T _{FD} | Fault Input ↓ to PWM I/O Change | — | — | 50 | ns | — |
| MP30 | T _{FH} | Minimum Pulse Width | 50 | — | — | ns | — |

Note 1: These parameters are characterized but not tested in manufacturing.

FIGURE 31-12: QEA/QEB INPUT CHARACTERISTICS

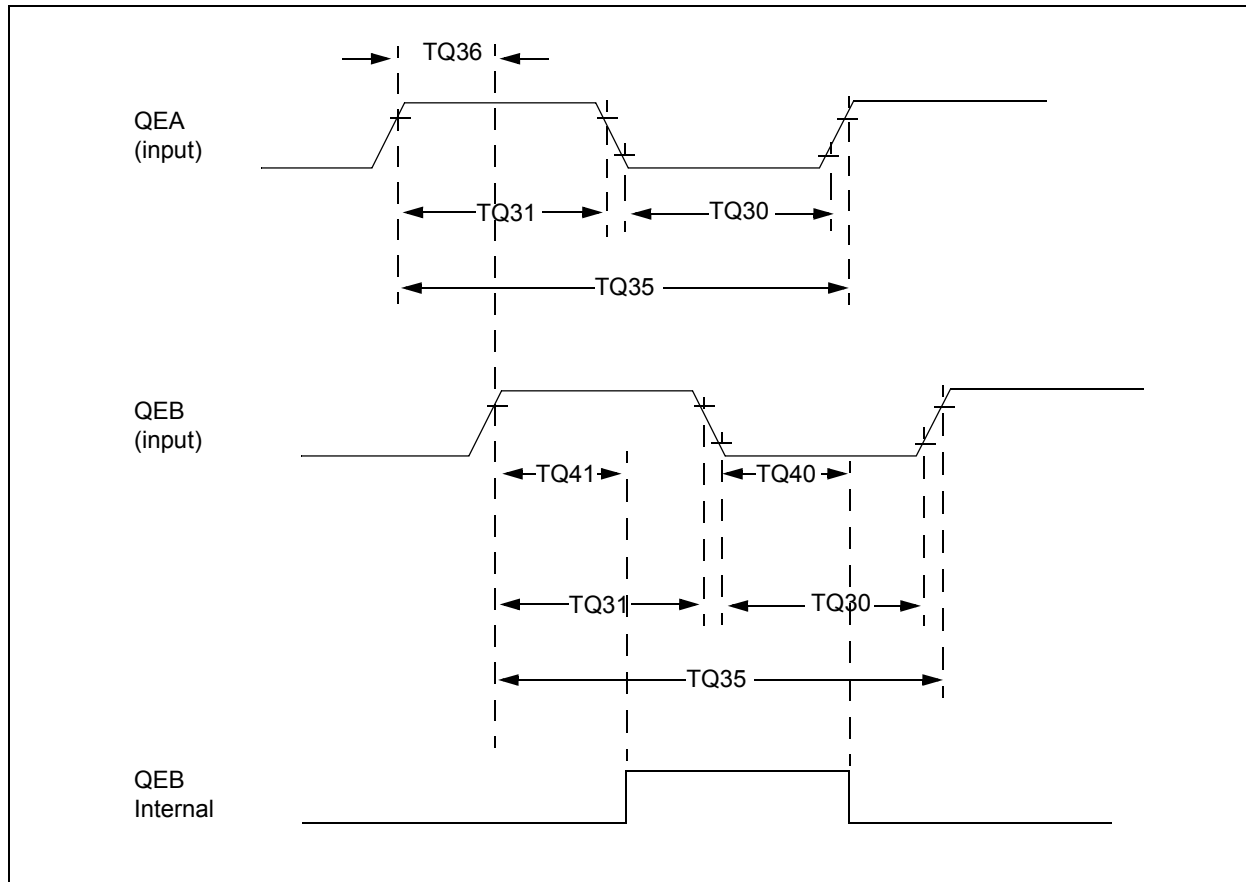


TABLE 31-30: QUADRATURE DECODER TIMING REQUIREMENTS

| AC CHARACTERISTICS | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | |
|--------------------|--------|---|--------------------|-----|-------|--|
| Param No. | Symbol | Characteristic ⁽¹⁾ | Typ ⁽²⁾ | Max | Units | Conditions |
| TQ30 | TQuL | Quadrature Input Low Time | 6 TcY | — | ns | — |
| TQ31 | TQuH | Quadrature Input High Time | 6 TcY | — | ns | — |
| TQ35 | TQuIN | Quadrature Input Period | 12 TcY | — | ns | — |
| TQ36 | TQuP | Quadrature Phase Period | 3 TcY | — | ns | — |
| TQ40 | TQuFL | Filter Time to Recognize Low, with Digital Filter | 3 * N * TcY | — | ns | N = 1, 2, 4, 16, 32, 64, 128 and 256 (Note 3) |
| TQ41 | TQuFH | Filter Time to Recognize High, with Digital Filter | 3 * N * TcY | — | ns | N = 1, 2, 4, 16, 32, 64, 128 and 256 (Note 3) |

Note 1: These parameters are characterized but not tested in manufacturing.

2: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

3: N = Index Channel Digital Filter Clock Divide Select bits. Refer to **Section 15. “Quadrature Encoder Interface (QEI)”** in the “dsPIC33F Family Reference Manual”. Please see the Microchip web site for the latest dsPIC33F Family Reference Manual sections.

FIGURE 31-13: QEI MODULE INDEX PULSE TIMING CHARACTERISTICS

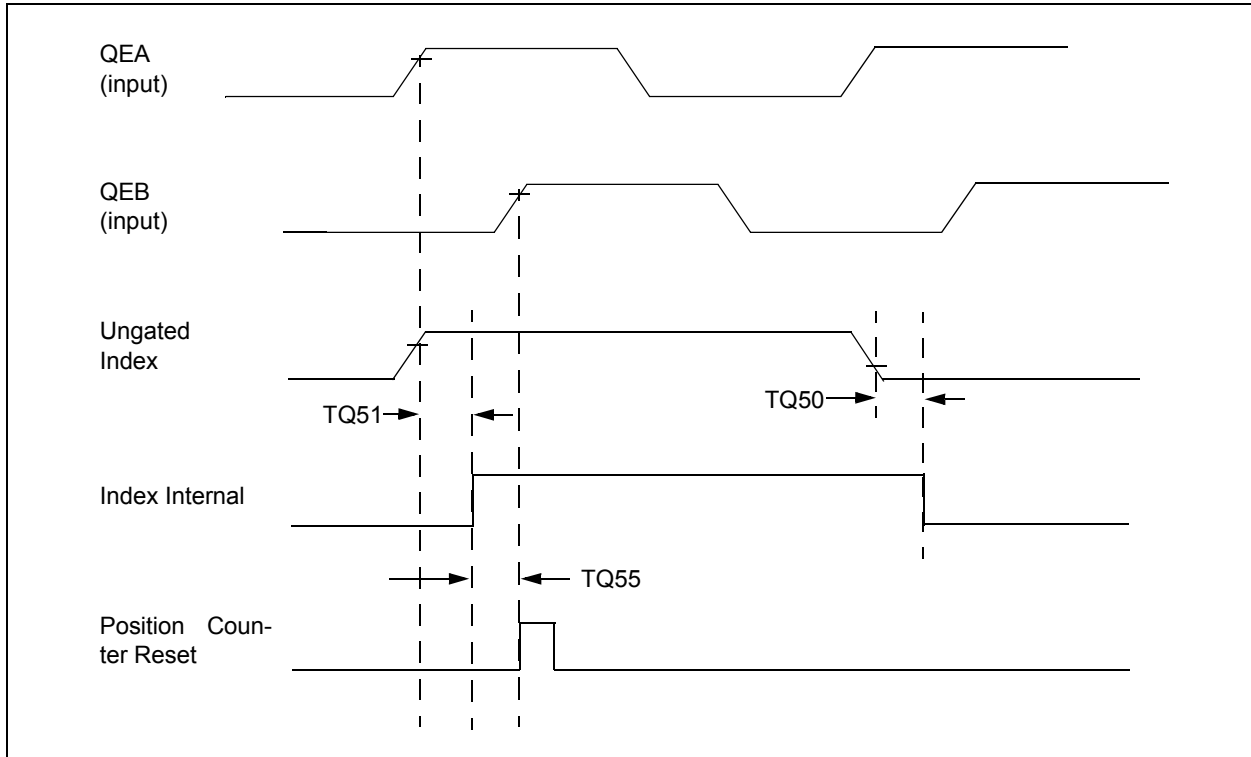


TABLE 31-31: QEI INDEX PULSE TIMING REQUIREMENTS

| AC CHARACTERISTICS | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | |
|--------------------|--------|---|------------------|-----|-------|--|
| Param No. | Symbol | Characteristic ⁽¹⁾ | Min | Max | Units | Conditions |
| TQ50 | TqIL | Filter Time to Recognize Low, with Digital Filter | $3 * N * T_{CY}$ | — | ns | $N = 1, 2, 4, 16, 32, 64, 128$ and 256 (Note 2) |
| TQ51 | TqiH | Filter Time to Recognize High, with Digital Filter | $3 * N * T_{CY}$ | — | ns | $N = 1, 2, 4, 16, 32, 64, 128$ and 256 (Note 2) |
| TQ55 | Tqidxr | Index Pulse Recognized to Position Counter Reset (ungated index) | $3 T_{CY}$ | — | ns | — |

Note 1: These parameters are characterized but not tested in manufacturing.

Note 2: Alignment of index pulses to QEA and QEB is shown for position counter Reset timing only. Shown for forward direction only (QEA leads QEB). Same timing applies for reverse direction (QEA lags QEB) but index pulse recognition occurs on falling edge.

FIGURE 31-14: SPIx MODULE MASTER MODE (CKE = 0) TIMING CHARACTERISTICS

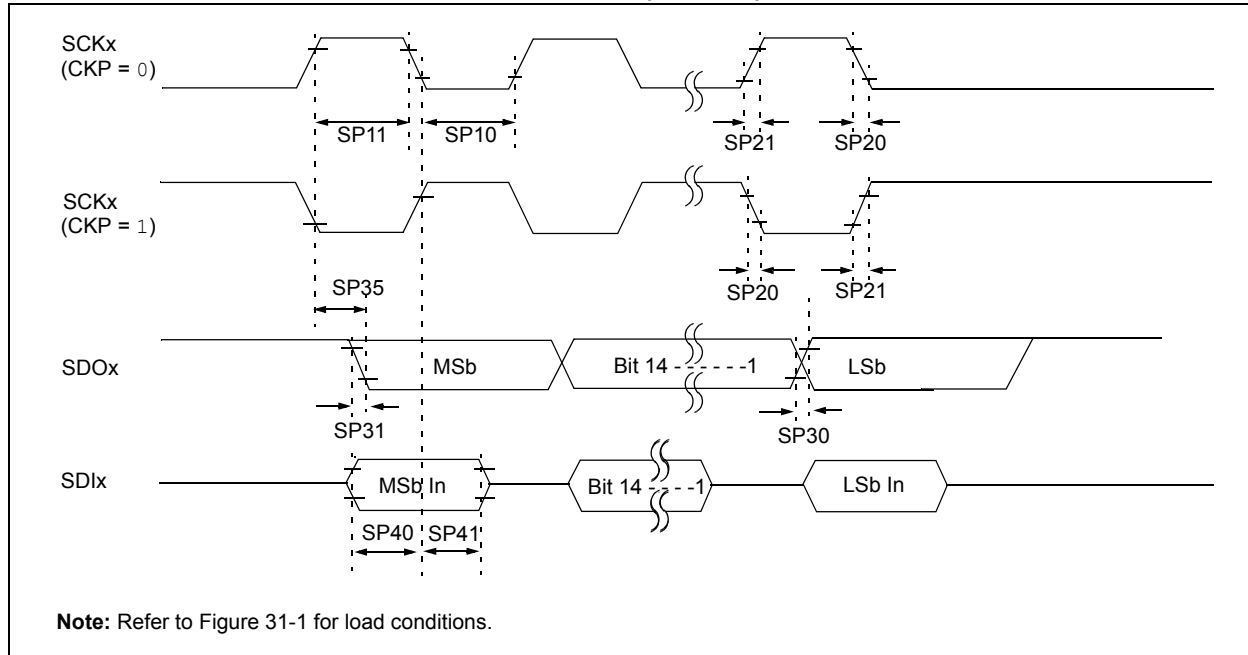


TABLE 31-32: SPIx MASTER MODE (CKE = 0) TIMING REQUIREMENTS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|--------------------|-----------------------|--|---|--------------------|-----|-------|--------------------------------------|
| Param No. | Symbol | Characteristic ⁽¹⁾ | Min | Typ ⁽²⁾ | Max | Units | Conditions |
| SP10 | TscL | SCKx Output Low Time | Tcy/2 | — | — | ns | See Note 3 |
| SP11 | TscH | SCKx Output High Time | Tcy/2 | — | — | ns | See Note 3 |
| SP20 | TscF | SCKx Output Fall Time | — | — | — | ns | See parameter D032 and Note 4 |
| SP21 | TscR | SCKx Output Rise Time | — | — | — | ns | See parameter D031 and Note 4 |
| SP30 | TdoF | SDOx Data Output Fall Time | — | — | — | ns | See parameter D032 and Note 4 |
| SP31 | TdoR | SDOx Data Output Rise Time | — | — | — | ns | See parameter D031 and Note 4 |
| SP35 | Tsch2doV, TscL2doV | SDOx Data Output Valid after SCKx Edge | — | 6 | 20 | ns | — |
| SP40 | TdiV2sch, TdiV2scL | Setup Time of SDIx Data Input to SCKx Edge | 23 | — | — | ns | — |
| SP41 | Tsch2diL, TscL2diL | Hold Time of SDIx Data Input to SCKx Edge | 30 | — | — | ns | — |

- 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.
- Note 3:** The minimum clock period for SCKx is 100 ns. Therefore, the clock generated in Master mode must not violate this specification.
- Note 4:** Assumes 50 pF load on all SPIx pins.

FIGURE 31-15: SPIx MODULE MASTER MODE (CKE = 1) TIMING CHARACTERISTICS

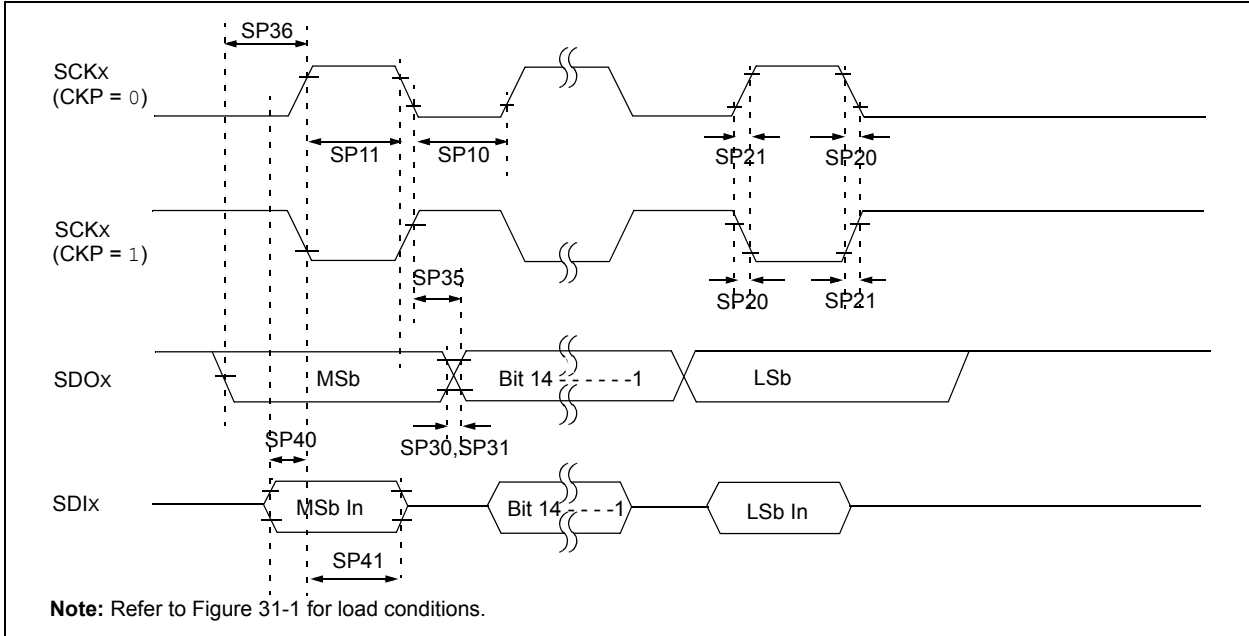


TABLE 31-33: SPIx MODULE MASTER MODE (CKE = 1) TIMING REQUIREMENTS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|--------------------|-----------------------|--|---|--------------------|-----|-------|--------------------------------------|
| Param No. | Symbol | Characteristic ⁽¹⁾ | Min | Typ ⁽²⁾ | Max | Units | Conditions |
| SP10 | TscL | SCKx Output Low Time | Tcy/2 | — | — | ns | See Note 3 |
| SP11 | Tsch | SCKx Output High Time | Tcy/2 | — | — | ns | See Note 3 |
| SP20 | TscF | SCKx Output Fall Time | — | — | — | ns | See parameter D032 and Note 4 |
| SP21 | TscR | SCKx Output Rise Time | — | — | — | ns | See parameter D031 and Note 4 |
| SP30 | TdoF | SDOx Data Output Fall Time | — | — | — | ns | See parameter D032 and Note 4 |
| SP31 | TdoR | SDOx Data Output Rise Time | — | — | — | ns | See parameter D031 and Note 4 |
| SP35 | Tsch2doV, TscL2doV | SDOx Data Output Valid after SCKx Edge | — | 6 | 20 | ns | — |
| SP36 | TdoV2sc, TdoV2scL | SDOx Data Output Setup to First SCKx Edge | 30 | — | — | ns | — |
| SP40 | TdiV2scH, TdiV2scL | Setup Time of SDIx Data Input to SCKx Edge | 23 | — | — | ns | — |
| SP41 | Tsch2diL, TscL2diL | Hold Time of SDIx Data Input to SCKx Edge | 30 | — | — | ns | — |

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.

Note 3: The minimum clock period for SCKx is 100 ns. The clock generated in Master mode must not violate this specification.

Note 4: Assumes 50 pF load on all SPIx pins.

FIGURE 31-16: SPIx MODULE SLAVE MODE (CKE = 0) TIMING CHARACTERISTICS

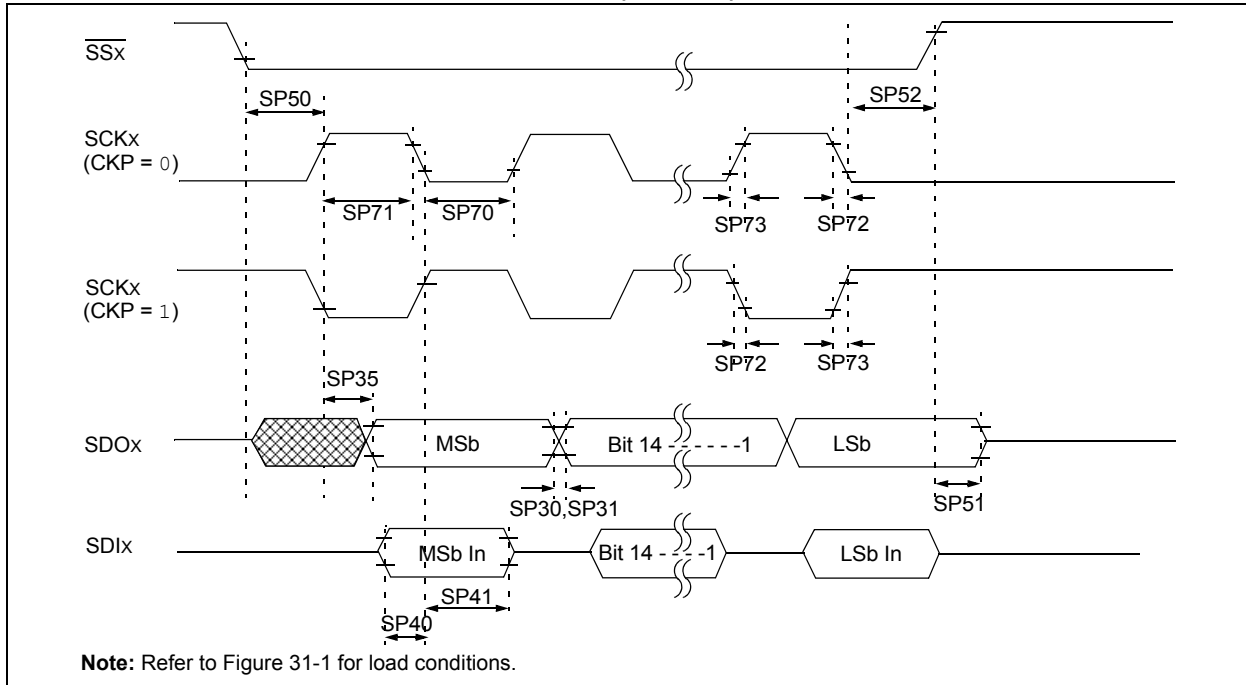


TABLE 31-34: SPIx MODULE SLAVE MODE (CKE = 0) TIMING REQUIREMENTS

| AC CHARACTERISTICS | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | | |
|--------------------|-----------------------|--|-----|--------------------|-----|-------|--------------------------------------|
| Param No. | Symbol | Characteristic ⁽¹⁾ | Min | Typ ⁽²⁾ | Max | Units | Conditions |
| SP70 | TscL | SCKx Input Low Time | 30 | — | — | ns | — |
| SP71 | Tsch | SCKx Input High Time | 30 | — | — | ns | — |
| SP72 | TscF | SCKx Input Fall Time | — | 10 | 25 | ns | See Note 3 |
| SP73 | TscR | SCKx Input Rise Time | — | 10 | 25 | ns | See Note 3 |
| SP30 | TdoF | SDOx Data Output Fall Time | — | — | — | ns | See parameter D032 and Note 3 |
| SP31 | TdoR | SDOx Data Output Rise Time | — | — | — | ns | See parameter D031 and Note 3 |
| SP35 | Tsch2doV, TscL2doV | SDOx Data Output Valid after SCKx Edge | — | — | 30 | ns | — |
| SP40 | TdiV2scH, TdiV2scL | Setup Time of SDIx Data Input to SCKx Edge | 20 | — | — | ns | — |
| SP41 | Tsch2diL, TscL2diL | Hold Time of SDIx Data Input to SCKx Edge | 20 | — | — | ns | — |
| SP50 | TssL2scH, TssL2scL | SSx ↓ to SCKx ↑ or SCKx Input | 120 | — | — | ns | — |
| SP51 | TssH2doZ | SSx ↑ to SDOx Output High-Impedance | 10 | — | 50 | ns | See Note 3 |

- 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.
Note 3: Assumes 50 pF load on all SPIx pins.

TABLE 31-34: SPIx MODULE SLAVE MODE (CKE = 0) TIMING REQUIREMENTS (CONTINUED)

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | |
|--------------------|----------------------|-------------------------------|---|--------------------|-----|-------|------------|
| Param No. | Symbol | Characteristic ⁽¹⁾ | Min | Typ ⁽²⁾ | Max | Units | Conditions |
| SP52 | Tsch2ssH TscL2ssH | SSx after SCKx Edge | 1.5 Tcy +40 | — | — | ns | — |

- 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.
Note 3: Assumes 50 pF load on all SPIx pins.

FIGURE 31-17: SPIx MODULE SLAVE MODE (CKE = 1) TIMING CHARACTERISTICS

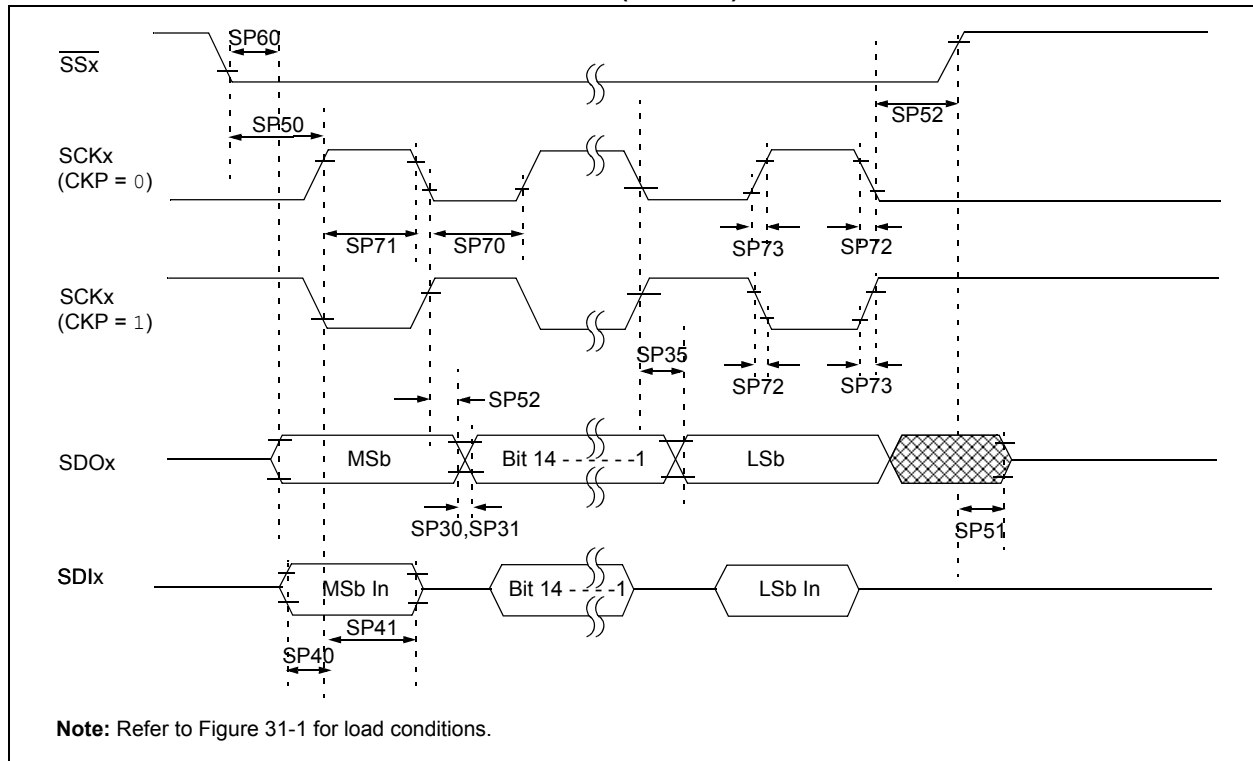


TABLE 31-35: SPIx MODULE SLAVE MODE (CKE = 1) TIMING REQUIREMENTS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | |
|--------------------|-----------------------|---|---|--------------------|-----|-------|--------------------------------------|
| Param No. | Symbol | Characteristic ⁽¹⁾ | Min | Typ ⁽²⁾ | Max | Units | Conditions |
| SP70 | TscL | SCKx Input Low Time | 30 | — | — | ns | — |
| SP71 | TscH | SCKx Input High Time | 30 | — | — | ns | — |
| SP72 | TscF | SCKx Input Fall Time | — | 10 | 25 | ns | See Note 3 |
| SP73 | TscR | SCKx Input Rise Time | — | 10 | 25 | ns | See Note 3 |
| SP30 | TdoF | SDOx Data Output Fall Time | — | — | — | ns | See parameter D032 and Note 3 |
| SP31 | TdoR | SDOx Data Output Rise Time | — | — | — | ns | See parameter D031 and Note 3 |
| SP35 | Tsch2doV, TscL2doV | SDOx Data Output Valid after SCKx Edge | — | — | 30 | ns | — |
| SP40 | TdiV2sch, TdiV2scL | Setup Time of SDIx Data Input to SCKx Edge | 20 | — | — | ns | — |
| SP41 | Tsch2diL, TscL2diL | Hold Time of SDIx Data Input to SCKx Edge | 20 | — | — | ns | — |
| SP50 | TssL2sch, TssL2scL | $\overline{\text{SSx}}$ ↓ to SCKx ↓ or SCKx ↑ Input | 120 | — | — | ns | — |
| SP51 | TssH2doZ | $\overline{\text{SSx}}$ ↑ to SDOx Output High-Impedance | 10 | — | 50 | ns | See Note 4 |
| SP52 | Tsch2ssH TscL2ssH | $\overline{\text{SSx}}$ ↑ after SCKx Edge | 1.5 Tcy + 40 | — | — | ns | — |
| SP60 | TssL2doV | SDOx Data Output Valid after $\overline{\text{SSx}}$ Edge | — | — | 50 | ns | — |

Note 1: These parameters are characterized but not tested in manufacturing.

2: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated.

3: The minimum clock period for SCKx is 100 ns. The clock generated in Master mode must not violate this specification.

4: Assumes 50 pF load on all SPIx pins.

FIGURE 31-18: I2Cx BUS START/STOP BITS TIMING CHARACTERISTICS (MASTER MODE)

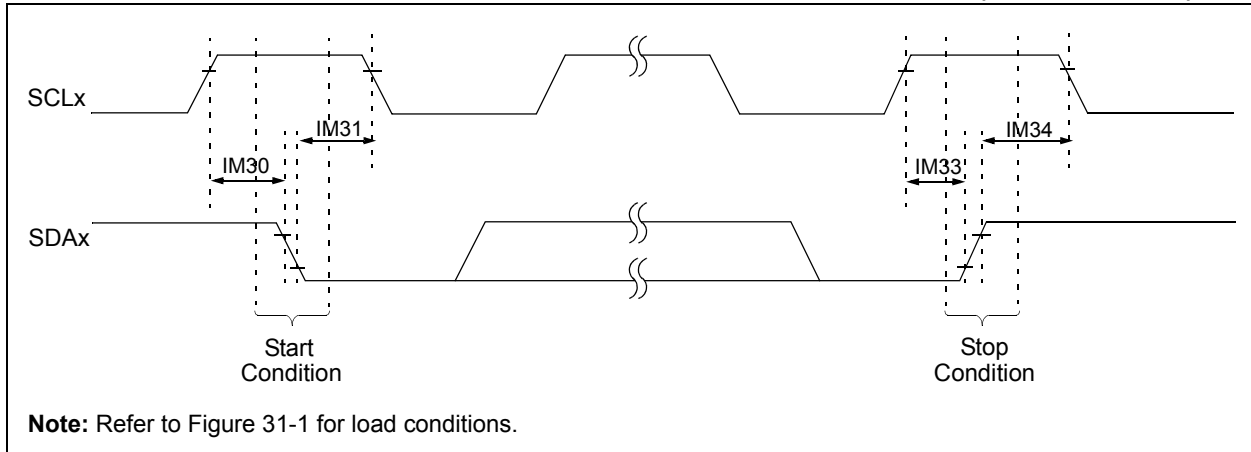


FIGURE 31-19: I2Cx BUS DATA TIMING CHARACTERISTICS (MASTER MODE)

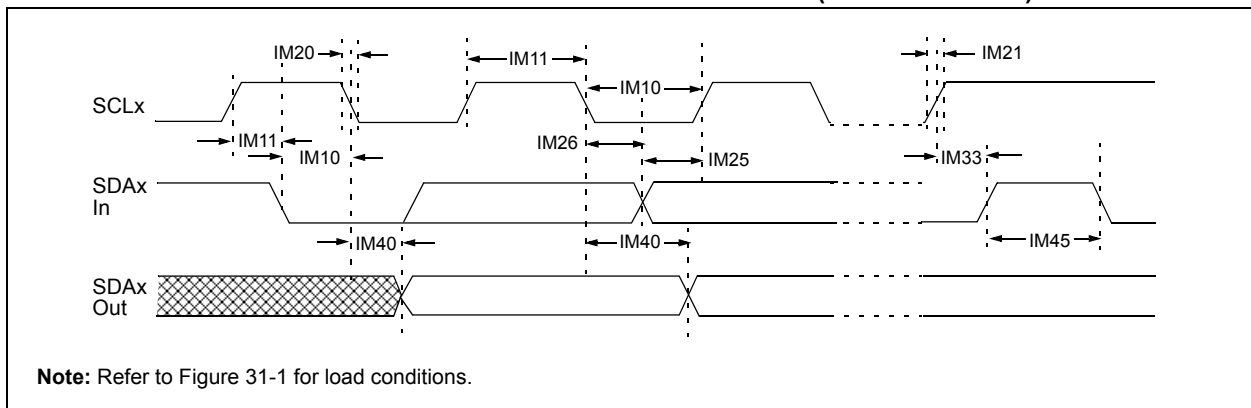


TABLE 31-36: I2Cx BUS DATA TIMING REQUIREMENTS (MASTER MODE)

| AC CHARACTERISTICS | | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | |
|--------------------|---------|----------------------------|---------------------------|---|------|---------------|---|
| Param No. | Symbol | Characteristic | | Min ⁽¹⁾ | Max | Units | Conditions |
| IM10 | TLO:SCL | Clock Low Time | 100 kHz mode | $T_{CY}/2$ (BRG + 1) | — | μs | — |
| | | | 400 kHz mode | $T_{CY}/2$ (BRG + 1) | — | μs | — |
| | | | 1 MHz mode ⁽²⁾ | $T_{CY}/2$ (BRG + 1) | — | μs | — |
| IM11 | THI:SCL | Clock High Time | 100 kHz mode | $T_{CY}/2$ (BRG + 1) | — | μs | — |
| | | | 400 kHz mode | $T_{CY}/2$ (BRG + 1) | — | μs | — |
| | | | 1 MHz mode ⁽²⁾ | $T_{CY}/2$ (BRG + 1) | — | μs | — |
| IM20 | TF:SCL | SDAx and SCLx Fall Time | 100 kHz mode | — | 300 | ns | Cb is specified to be from 10 to 400 pF |
| | | | 400 kHz mode | $20 + 0.1 C_B$ | 300 | ns | |
| | | | 1 MHz mode ⁽²⁾ | — | 100 | ns | |
| IM21 | TR:SCL | SDAx and SCLx Rise Time | 100 kHz mode | — | 1000 | ns | Cb is specified to be from 10 to 400 pF |
| | | | 400 kHz mode | $20 + 0.1 C_B$ | 300 | ns | |
| | | | 1 MHz mode ⁽²⁾ | — | 300 | ns | |
| IM25 | TSU:DAT | Data Input Setup Time | 100 kHz mode | 250 | — | ns | — |
| | | | 400 kHz mode | 100 | — | ns | |
| | | | 1 MHz mode ⁽²⁾ | 40 | — | ns | |
| IM26 | THD:DAT | Data Input Hold Time | 100 kHz mode | 0 | — | μs | — |
| | | | 400 kHz mode | 0 | 0.9 | μs | |
| | | | 1 MHz mode ⁽²⁾ | 0.2 | — | μs | |
| IM30 | TSU:STA | Start Condition Setup Time | 100 kHz mode | $T_{CY}/2$ (BRG + 1) | — | μs | Only relevant for Repeated Start condition |
| | | | 400 kHz mode | $T_{CY}/2$ (BRG + 1) | — | μs | |
| | | | 1 MHz mode ⁽²⁾ | $T_{CY}/2$ (BRG + 1) | — | μs | |
| IM31 | THD:STA | Start Condition Hold Time | 100 kHz mode | $T_{CY}/2$ (BRG + 1) | — | μs | After this period the first clock pulse is generated |
| | | | 400 kHz mode | $T_{CY}/2$ (BRG + 1) | — | μs | |
| | | | 1 MHz mode ⁽²⁾ | $T_{CY}/2$ (BRG + 1) | — | μs | |
| IM33 | TSU:STO | Stop Condition Setup Time | 100 kHz mode | $T_{CY}/2$ (BRG + 1) | — | μs | — |
| | | | 400 kHz mode | $T_{CY}/2$ (BRG + 1) | — | μs | |
| | | | 1 MHz mode ⁽²⁾ | $T_{CY}/2$ (BRG + 1) | — | μs | |
| IM34 | THD:STO | Stop Condition Hold Time | 100 kHz mode | $T_{CY}/2$ (BRG + 1) | — | ns | — |
| | | | 400 kHz mode | $T_{CY}/2$ (BRG + 1) | — | ns | |
| | | | 1 MHz mode ⁽²⁾ | $T_{CY}/2$ (BRG + 1) | — | ns | |
| IM40 | TAA:SCL | Output Valid From Clock | 100 kHz mode | — | 3500 | ns | — |
| | | | 400 kHz mode | — | 1000 | ns | — |
| | | | 1 MHz mode ⁽²⁾ | — | 400 | ns | — |
| IM45 | TBF:SDA | Bus Free Time | 100 kHz mode | 4.7 | — | μs | Time the bus must be free before a new transmission can start |
| | | | 400 kHz mode | 1.3 | — | μs | |
| | | | 1 MHz mode ⁽²⁾ | 0.5 | — | μs | |
| IM50 | CB | Bus Capacitive Loading | — | 400 | pF | — | |

Note 1: BRG is the value of the I²C Baud Rate Generator. Refer to **Section 19. “Inter-Integrated Circuit (I²C™)”** in the “dsPIC33F Family Reference Manual”. Please see the Microchip web site for the latest dsPIC33F Family Reference Manual sections.

2: Maximum pin capacitance = 10 pF for all I2Cx pins (for 1 MHz mode only).

3: Typical value for this parameter is 130 ns.

TABLE 31-36: I2Cx BUS DATA TIMING REQUIREMENTS (MASTER MODE)

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | |
|--------------------|--------|---------------------|---|-----|-------|------------|
| Param No. | Symbol | Characteristic | Min ⁽¹⁾ | Max | Units | Conditions |
| IM51 | TPGD | Pluse Gobbler Delay | 65 | 390 | ns | See Note 3 |

Note 1: BRG is the value of the I²C Baud Rate Generator. Refer to Section 19. “Inter-Integrated Circuit (I²C™)” in the “dsPIC33F Family Reference Manual”. Please see the Microchip web site for the latest dsPIC33F Family Reference Manual sections.

2: Maximum pin capacitance = 10 pF for all I2Cx pins (for 1 MHz mode only).

3: Typical value for this parameter is 130 ns.

FIGURE 31-20: I2Cx BUS START/STOP BITS TIMING CHARACTERISTICS (SLAVE MODE)

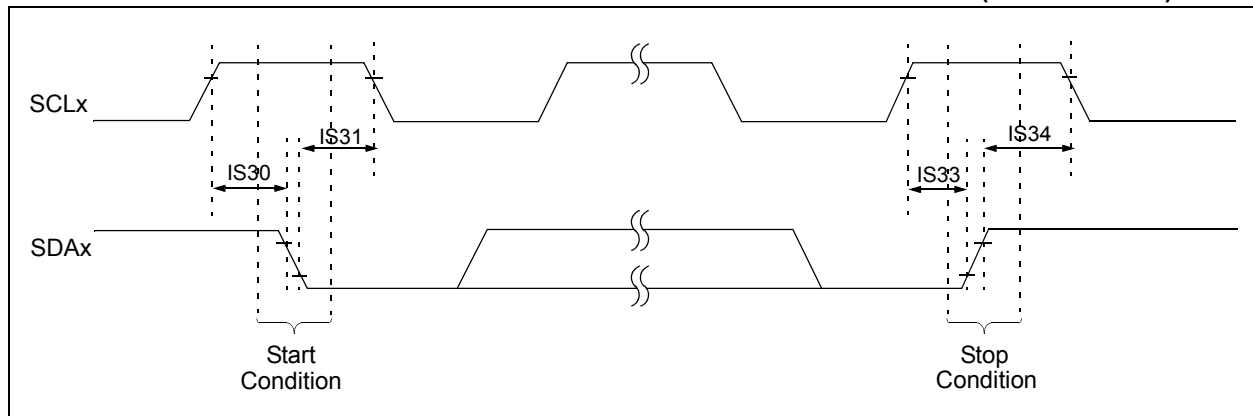


FIGURE 31-21: I2Cx BUS DATA TIMING CHARACTERISTICS (SLAVE MODE)

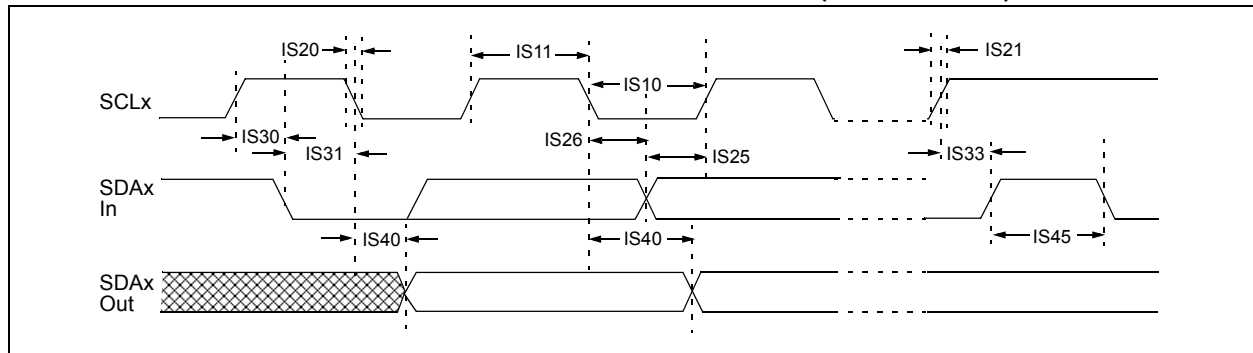


TABLE 31-37: I2Cx BUS DATA TIMING REQUIREMENTS (SLAVE MODE)

| AC CHARACTERISTICS | | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | |
|--------------------|---------|----------------------------|---------------------------|---|------|-------|---|
| Param. | Symbol | Characteristic | | Min | Max | Units | Conditions |
| IS10 | TLO:SCL | Clock Low Time | 100 kHz mode | 4.7 | — | μs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 1.3 | — | μs | Device must operate at a minimum of 10 MHz |
| | | | 1 MHz mode ⁽¹⁾ | 0.5 | — | μs | — |
| IS11 | THI:SCL | Clock High Time | 100 kHz mode | 4.0 | — | μs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 0.6 | — | μs | Device must operate at a minimum of 10 MHz |
| | | | 1 MHz mode ⁽¹⁾ | 0.5 | — | μs | — |
| IS20 | TF:SCL | SDAx and SCLx Fall Time | 100 kHz mode | — | 300 | ns | CB is specified to be from 10 to 400 pF |
| | | | 400 kHz mode | 20 + 0.1 CB | 300 | ns | |
| | | | 1 MHz mode ⁽¹⁾ | — | 100 | ns | |
| IS21 | TR:SCL | SDAx and SCLx Rise Time | 100 kHz mode | — | 1000 | ns | CB is specified to be from 10 to 400 pF |
| | | | 400 kHz mode | 20 + 0.1 CB | 300 | ns | |
| | | | 1 MHz mode ⁽¹⁾ | — | 300 | ns | |
| IS25 | TSU:DAT | Data Input Setup Time | 100 kHz mode | 250 | — | ns | — |
| | | | 400 kHz mode | 100 | — | ns | |
| | | | 1 MHz mode ⁽¹⁾ | 100 | — | ns | |
| IS26 | THD:DAT | Data Input Hold Time | 100 kHz mode | 0 | — | μs | — |
| | | | 400 kHz mode | 0 | 0.9 | μs | |
| | | | 1 MHz mode ⁽¹⁾ | 0 | 0.3 | μs | |
| IS30 | TSU:STA | Start Condition Setup Time | 100 kHz mode | 4.7 | — | μs | Only relevant for Repeated Start condition |
| | | | 400 kHz mode | 0.6 | — | μs | |
| | | | 1 MHz mode ⁽¹⁾ | 0.25 | — | μs | |
| IS31 | THD:STA | Start Condition Hold Time | 100 kHz mode | 4.0 | — | μs | After this period, the first clock pulse is generated |
| | | | 400 kHz mode | 0.6 | — | μs | |
| | | | 1 MHz mode ⁽¹⁾ | 0.25 | — | μs | |
| IS33 | TSU:STO | Stop Condition Setup Time | 100 kHz mode | 4.7 | — | μs | — |
| | | | 400 kHz mode | 0.6 | — | μs | |
| | | | 1 MHz mode ⁽¹⁾ | 0.6 | — | μs | |
| IS34 | THD:STO | Stop Condition Hold Time | 100 kHz mode | 4000 | — | ns | — |
| | | | 400 kHz mode | 600 | — | ns | |
| | | | 1 MHz mode ⁽¹⁾ | 250 | — | ns | |
| IS40 | TAA:SCL | Output Valid From Clock | 100 kHz mode | 0 | 3500 | ns | — |
| | | | 400 kHz mode | 0 | 1000 | ns | |
| | | | 1 MHz mode ⁽¹⁾ | 0 | 350 | ns | |
| IS45 | TBF:SDA | Bus Free Time | 100 kHz mode | 4.7 | — | μs | Time the bus must be free before a new transmission can start |
| | | | 400 kHz mode | 1.3 | — | μs | |
| | | | 1 MHz mode ⁽¹⁾ | 0.5 | — | μs | |
| IS50 | CB | Bus Capacitive Loading | | — | 400 | pF | — |

Note 1: Maximum pin capacitance = 10 pF for all I2Cx pins (for 1 MHz mode only).

FIGURE 31-22: ECAN MODULE I/O TIMING CHARACTERISTICS

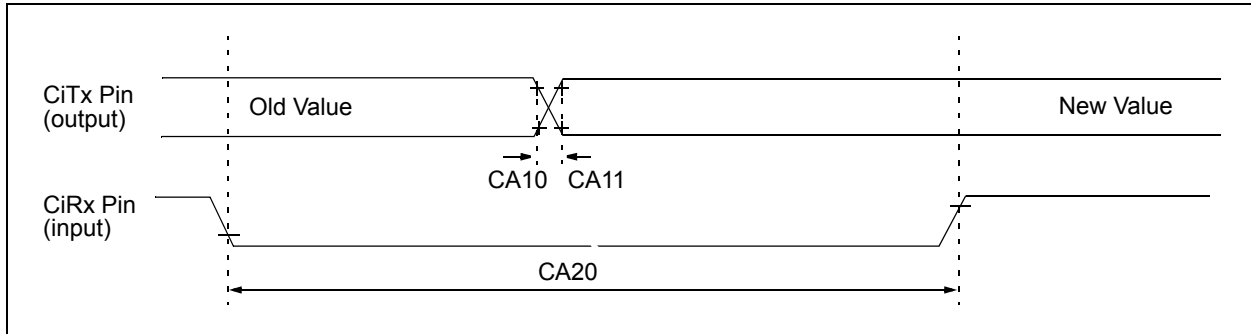


TABLE 31-38: ECAN MODULE I/O TIMING REQUIREMENTS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ | | | | |
|--------------------|--------|---|---|--------------------|-----|-------|--------------------|
| Param No. | Symbol | Characteristic ⁽¹⁾ | Min | Typ ⁽²⁾ | Max | Units | Conditions |
| CA10 | TioF | Port Output Fall Time | — | — | — | ns | See parameter D032 |
| CA11 | TioR | Port Output Rise Time | — | — | — | ns | See parameter D031 |
| CA20 | Tcwf | Pulse Width to Trigger CAN Wake-up Filter | 120 | | | ns | — |

Note 1: These parameters are characterized but not tested in manufacturing.

2: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

TABLE 31-39: ADC MODULE SPECIFICATIONS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | |
|-------------------------|--------|--|---|-----|----------------------------|---------------|---|
| Param No. | Symbol | Characteristic | Min. | Typ | Max. | Units | Conditions |
| Device Supply | | | | | | | |
| AD01 | AVDD | Module VDD Supply | Greater of VDD – 0.3 or 3.0 | — | Lesser of VDD + 0.3 or 3.6 | V | — |
| AD02 | AVSS | Module VSS Supply | VSS – 0.3 | — | VSS + 0.3 | V | — |
| Reference Inputs | | | | | | | |
| AD05 | VREFH | Reference Voltage High | AVSS + 2.7 | — | AVDD | V | See Note 1 |
| AD05a | | | 3.0 | — | 3.6 | V | VREFH = AVDD VREFL = AVSS = 0 |
| AD06 | VREFL | Reference Voltage Low | AVSS | — | AVDD – 2.7 | V | See Note 1 |
| AD06a | | | 0 | — | 0 | V | VREFH = AVDD VREFL = AVSS = 0 |
| AD07 | VREF | Absolute Reference Voltage | 2.7 | — | 3.6 | V | VREF = VREFH - VREFL |
| AD08 | IREF | Current Drain | — | — | 10 | μA | ADC off |
| AD09 | IAD | Operating Current | — | 7.0 | 9.0 | mA | ADC operating in 10-bit mode, see Note 1 |
| | | | — | 2.7 | 3.2 | mA | ADC operating in 12-bit mode, see Note 1 |
| Analog Input | | | | | | | |
| AD12 | VINH | Input Voltage Range VINH | VINL | — | VREFH | V | This voltage reflects Sample and Hold Channels 0, 1, 2, and 3 (CH0-CH3), positive input |
| AD13 | VINL | Input Voltage Range VINL | VREFL | — | AVSS + 1V | V | This voltage reflects Sample and Hold Channels 0, 1, 2, and 3 (CH0-CH3), negative input |
| AD17 | RIN | Recommended Impedance of Analog Voltage Source | — | — | 200 | Ω | 10-bit ADC |
| | | | — | — | 200 | Ω | 12-bit ADC |

Note 1: These parameters are not characterized or tested in manufacturing.

TABLE 31-40: ADC MODULE SPECIFICATIONS (12-BIT MODE)

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | |
|--|--------|--------------------------------|---|------|------|-------|---|
| Param No. | Symbol | Characteristic | Min. | Typ | Max. | Units | Conditions |
| ADC Accuracy (12-bit Mode) – Measurements with external VREF+/VREF- | | | | | | | |
| AD20a | Nr | Resolution | 12 data bits | | | bits | |
| AD21a | INL | Integral Nonlinearity | -2 | — | +2 | LSb | V _{INL} = AV _{SS} = V _{REFL} = 0V, AV _{DD} = V _{REFH} = 3.6V |
| AD22a | DNL | Differential Nonlinearity | >-1 | — | <1 | LSb | V _{INL} = AV _{SS} = V _{REFL} = 0V, AV _{DD} = V _{REFH} = 3.6V |
| AD23a | GERR | Gain Error | 1.25 | 1.5 | 3 | LSb | V _{INL} = AV _{SS} = V _{REFL} = 0V, AV _{DD} = V _{REFH} = 3.6V |
| AD24a | EOFF | Offset Error | 1.25 | 1.52 | 2 | LSb | V _{INL} = AV _{SS} = V _{REFL} = 0V, AV _{DD} = V _{REFH} = 3.6V |
| AD25a | — | Monotonicity | — | — | — | — | Guaranteed |
| ADC Accuracy (12-bit Mode) – Measurements with internal VREF+/VREF- | | | | | | | |
| AD20a | Nr | Resolution | 12 data bits | | | bits | |
| AD21a | INL | Integral Nonlinearity | -2 | — | +2 | LSb | V _{INL} = AV _{SS} = 0V, AV _{DD} = 3.6V |
| AD22a | DNL | Differential Nonlinearity | >-1 | — | <1 | LSb | V _{INL} = AV _{SS} = 0V, AV _{DD} = 3.6V |
| AD23a | GERR | Gain Error | 2 | 3 | 7 | LSb | V _{INL} = AV _{SS} = 0V, AV _{DD} = 3.6V |
| AD24a | EOFF | Offset Error | 2 | 3 | 5 | LSb | V _{INL} = AV _{SS} = 0V, AV _{DD} = 3.6V |
| AD25a | — | Monotonicity | — | — | — | — | Guaranteed |
| Dynamic Performance (12-bit Mode) | | | | | | | |
| AD30a | THD | Total Harmonic Distortion | — | — | -75 | dB | — |
| AD31a | SINAD | Signal to Noise and Distortion | 68.5 | 69.5 | — | dB | — |
| AD32a | SFDR | Spurious Free Dynamic Range | 80 | — | — | dB | — |
| AD33a | FNYQ | Input Signal Bandwidth | — | — | 250 | kHz | — |
| AD34a | ENOB | Effective Number of Bits | 11.09 | 11.3 | — | bits | — |

TABLE 31-41: ADC MODULE SPECIFICATIONS (10-BIT MODE)

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | |
|--|------------------|--------------------------------|---|------|------|-------|---|
| Param No. | Symbol | Characteristic | Min. | Typ | Max. | Units | Conditions |
| ADC Accuracy (10-bit Mode) – Measurements with external VREF+/VREF- | | | | | | | |
| AD20b | Nr | Resolution | 10 data bits | | | bits | |
| AD21b | INL | Integral Nonlinearity | -1.5 | — | +1.5 | LSb | V _{INL} = AV _{SS} = V _{REFL} = 0V, AV _{DD} = V _{REFH} = 3.6V |
| AD22b | DNL | Differential Nonlinearity | >-1 | — | <1 | LSb | V _{INL} = AV _{SS} = V _{REFL} = 0V, AV _{DD} = V _{REFH} = 3.6V |
| AD23b | GERR | Gain Error | 1 | 3 | 6 | LSb | V _{INL} = AV _{SS} = V _{REFL} = 0V, AV _{DD} = V _{REFH} = 3.6V |
| AD24b | E _{OFF} | Offset Error | 1 | 2 | 5 | LSb | V _{INL} = AV _{SS} = V _{REFL} = 0V, AV _{DD} = V _{REFH} = 3.6V |
| AD25b | — | Monotonicity | — | — | — | — | Guaranteed |
| ADC Accuracy (10-bit Mode) – Measurements with internal VREF+/VREF- | | | | | | | |
| AD20b | Nr | Resolution | 10 data bits | | | bits | |
| AD21b | INL | Integral Nonlinearity | -1 | — | +1 | LSb | V _{INL} = AV _{SS} = 0V, AV _{DD} = 3.6V |
| AD22b | DNL | Differential Nonlinearity | >-1 | — | <1 | LSb | V _{INL} = AV _{SS} = 0V, AV _{DD} = 3.6V |
| AD23b | GERR | Gain Error | 1 | 5 | 6 | LSb | V _{INL} = AV _{SS} = 0V, AV _{DD} = 3.6V |
| AD24b | E _{OFF} | Offset Error | 1 | 2 | 3 | LSb | V _{INL} = AV _{SS} = 0V, AV _{DD} = 3.6V |
| AD25b | — | Monotonicity | — | — | — | — | Guaranteed |
| Dynamic Performance (10-bit Mode) | | | | | | | |
| AD30b | THD | Total Harmonic Distortion | — | — | -64 | dB | — |
| AD31b | SINAD | Signal to Noise and Distortion | 57 | 58.5 | — | dB | — |
| AD32b | SFDR | Spurious Free Dynamic Range | 72 | — | — | dB | — |
| AD33b | F _{NYQ} | Input Signal Bandwidth | — | — | 550 | kHz | — |
| AD34b | ENOB | Effective Number of Bits | 9.16 | 9.4 | — | bits | — |

FIGURE 31-23: ADC CONVERSION (12-BIT MODE) TIMING CHARACTERISTICS
(ASAM = 0, SSRC<2:0> = 000)

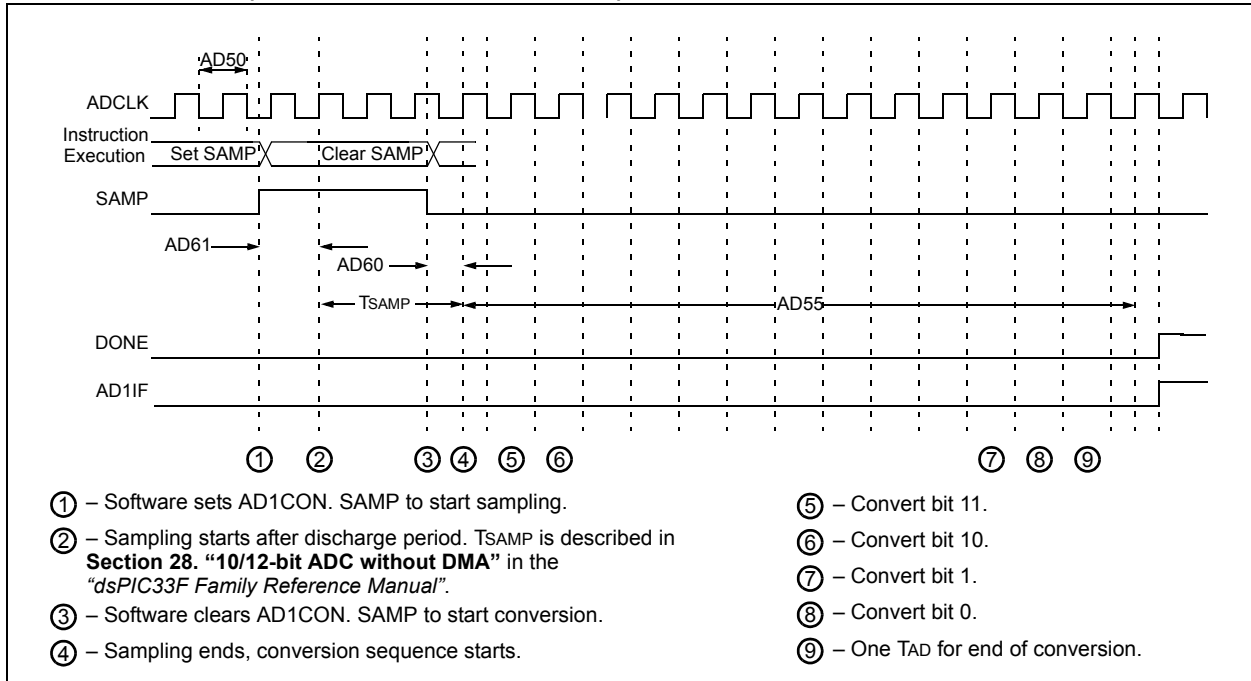


TABLE 31-42: ADC CONVERSION (12-BIT MODE) TIMING REQUIREMENTS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | |
|--------------------------|--------|--|---|---------|-------|---------------|-----------------------------------|
| Param No. | Symbol | Characteristic | Min. | Typ | Max. | Units | Conditions |
| Clock Parameters | | | | | | | |
| AD50 | TAD | ADC Clock Period | 117.6 | — | — | ns | |
| AD51 | tRC | ADC Internal RC Oscillator Period | — | 250 | — | ns | |
| Conversion Rate | | | | | | | |
| AD55 | tCONV | Conversion Time | — | 14 TAD | | ns | |
| AD56 | FCNV | Throughput Rate | — | — | 500 | Ksps | |
| AD57 | TSAMP | Sample Time | 3 TAD | — | — | — | |
| Timing Parameters | | | | | | | |
| AD60 | tPCS | Conversion Start from Sample Trigger ⁽²⁾ | 2 TAD | — | 3 TAD | — | Auto convert trigger not selected |
| AD61 | tPSS | Sample Start from Setting Sample (SAMP) bit ⁽²⁾ | 2 TAD | — | 3 TAD | — | — |
| AD62 | tCSS | Conversion Completion to Sample Start (ASAM = 1) ⁽²⁾ | — | 0.5 TAD | — | — | — |
| AD63 | tDPU | Time to Stabilize Analog Stage from ADC Off to ADC On ⁽²⁾ | — | — | 20 | μs | — |

- Note 1:** Because the sample caps will eventually lose charge, clock rates below 10 kHz may affect linearity performance, especially at elevated temperatures.
- 2:** These parameters are characterized but not tested in manufacturing.
- 3:** The tDPU is the time required for the ADC module to stabilize at the appropriate level when the module is turned on (AD1CON1<ADON>='1'). During this time, the ADC result is indeterminate.

FIGURE 31-24: ADC CONVERSION (10-BIT MODE) TIMING CHARACTERISTICS (CHPS<1:0> = 01, SIMSAM = 0, ASAM = 0, SSRC<2:0> = 000)

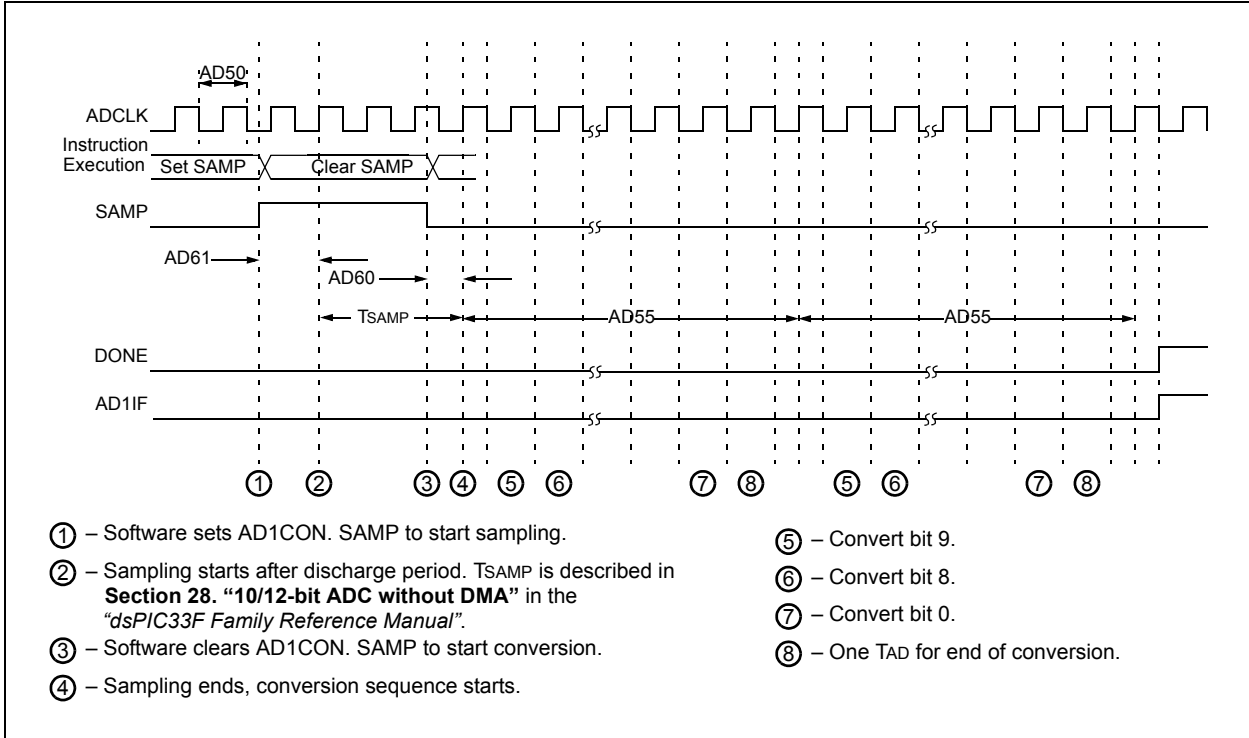


FIGURE 31-25: ADC CONVERSION (10-BIT MODE) TIMING CHARACTERISTICS (CHPS<1:0> = 01, SIMSAM = 0, ASAM = 1, SSRC<2:0> = 111, SAMC<4:0> = 00001)

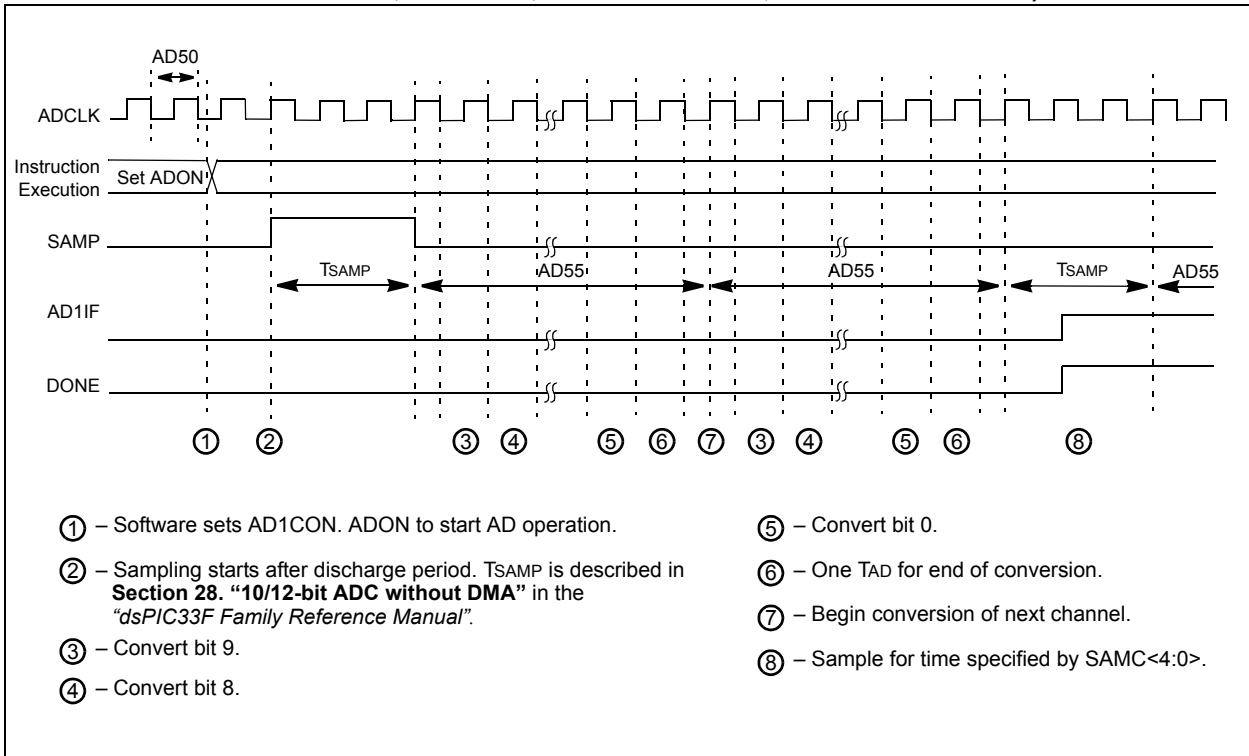


TABLE 31-43: ADC CONVERSION (10-BIT MODE) TIMING REQUIREMENTS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|--------------------------|--------|--|---|--------------------|-------|-------|-----------------------------------|
| Param No. | Symbol | Characteristic | Min. | Typ ⁽¹⁾ | Max. | Units | Conditions |
| Clock Parameters | | | | | | | |
| AD50 | TAD | ADC Clock Period | 76 | — | — | ns | |
| AD51 | tRC | ADC Internal RC Oscillator Period | — | 250 | — | ns | |
| Conversion Rate | | | | | | | |
| AD55 | tCONV | Conversion Time | — | 12 TAD | — | — | |
| AD56 | FCNV | Throughput Rate | — | — | 1.1 | Msp/s | |
| AD57 | TSAMP | Sample Time | 2 TAD | — | — | — | |
| Timing Parameters | | | | | | | |
| AD60 | tPCS | Conversion Start from Sample Trigger ⁽¹⁾ | 2 TAD | — | 3 TAD | — | Auto-Convert Trigger not selected |
| AD61 | tPSS | Sample Start from Setting Sample (SAMP) bit ⁽¹⁾ | 2 TAD | — | 3 TAD | — | — |
| AD62 | tCSS | Conversion Completion to Sample Start (ASAM = 1) ⁽¹⁾ | — | 0.5 TAD | — | — | — |
| AD63 | tDPU | Time to Stabilize Analog Stage from ADC Off to ADC On ⁽¹⁾ | — | — | 20 | μs | — |

- Note 1:** These parameters are characterized but not tested in manufacturing.
- 2:** Because the sample caps will eventually lose charge, clock rates below 10 kHz may affect linearity performance, especially at elevated temperatures.
- 3:** The tDPU is the time required for the ADC module to stabilize at the appropriate level when the module is turned on (AD1CON1<ADON>='1'). During this time, the ADC result is indeterminate.

TABLE 31-44: AUDIO DAC MODULE SPECIFICATIONS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|-------------------------|--------|--------------------------------------|---|---------|------|-------|--|
| Param No. | Symbol | Characteristic | Min. | Typ | Max. | Units | Conditions |
| Clock Parameters | | | | | | | |
| | VRES | Resolution | — | 16-bits | — | — | |
| | | Gain Error | — | -2% | — | — | |
| | | Offset Error | — | — | ±30 | MV | Dependent on voltage reference stability |
| | | Differential Non-Linearity (DNL) | — | ±0.1 | — | LSB | Relative to 14-bit accuracy |
| | | Integral Non-Linearity (INL) | — | ±0.2 | — | LSB | Relative to 14-bit accuracy |
| | VOD+ | Positive Output Differential Voltage | — | 1.15 | — | V | See Note 1 . |
| | VOD- | Negative Output Differential Voltage | — | 1.15 | — | V | See Note 1 . |

Note 1: Measured across DAC positive and negative outputs with no load and FORM bit (DACxCON<8>) = 0.

TABLE 31-45: AUDIO DAC MODULE SPECIFICATIONS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|-------------------------|--------|--|---|-----|------|-------|------------------------------|
| Param No. | Symbol | Characteristic | Min. | Typ | Max. | Units | Conditions |
| Clock Parameters | | | | | | | |
| | | Clock frequency | — | — | 25.6 | MHz | Clock |
| | | Sample Rate | 0 | — | 100 | kHz | |
| | | Input data frequency | 0 | — | 45 | kHz | Sampling frequency = 100 kHz |
| | | Initialization period | 1024 | — | — | Clks | Time before first sample |
| | | Signal to Noise Ratio | — | 62 | — | dB | Sampling frequency = 96 kHz |
| | | Signal to Noise Ratio and Distortion Ratio | — | 62 | — | dB | Sampling frequency = 96 kHz |

TABLE 31-46: COMPARATOR TIMING SPECIFICATIONS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|--------------------|--------|---|---|-----|------|-------|------------|
| Param No. | Symbol | Characteristic | Min. | Typ | Max. | Units | Conditions |
| 300 | TRESP | Response Time ^(1,2) | — | 150 | 400 | ns | |
| 301 | TMC2OV | Comparator Mode Change to Output Valid ⁽¹⁾ | — | — | 10 | μs | |

Note 1: Parameters are characterized but not tested.

Note 2: Response time measured with one comparator input at (VDD - 1.5)/2, while the other input transitions from VSS to VDD.

TABLE 31-47: COMPARATOR MODULE SPECIFICATIONS

| DC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|--------------------|-------------------|--|---|-----|-----------|-------|------------|
| Param No. | Symbol | Characteristic | Min. | Typ | Max. | Units | Conditions |
| D300 | VI _{OFF} | Input Offset Voltage ⁽¹⁾ | — | ±10 | — | mV | |
| D301 | VI _{CM} | Input Common Mode Voltage ⁽¹⁾ | 0 | — | AVDD-1.5V | V | |
| D302 | CMRR | Common Mode Rejection Ratio ⁽¹⁾ | -54 | — | — | dB | |

Note 1: Parameters are characterized but not tested.

TABLE 31-48: COMPARATOR REFERENCE VOLTAGE SETTling TIME SPECIFICATIONS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|--------------------|--------|------------------------------|---|-----|------|-------|------------|
| Param No. | Symbol | Characteristic | Min. | Typ | Max. | Units | Conditions |
| VR310 | TSET | Settling Time ⁽¹⁾ | — | — | 10 | μs | |

Note 1: Setting time measured while CVRR = 1 and CVR3:CVR0 bits transition from '0000' to '1111'.

TABLE 31-49: COMPARATOR REFERENCE VOLTAGE SPECIFICATIONS

| DC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|--------------------|--------|-------------------------|---|-----|-----------|-------|------------|
| Param No. | Symbol | Characteristic | Min. | Typ | Max. | Units | Conditions |
| VRD310 | CVRES | Resolution | CVRSRC/24 | — | CVRSRC/32 | LSb | |
| VRD311 | CVRAA | Absolute Accuracy | — | — | 0.5 | LSb | |
| VRD312 | CVRUR | Unit Resistor Value (R) | — | 2k | — | Ω | |

FIGURE 31-26: PARALLEL SLAVE PORT TIMING DIAGRAM

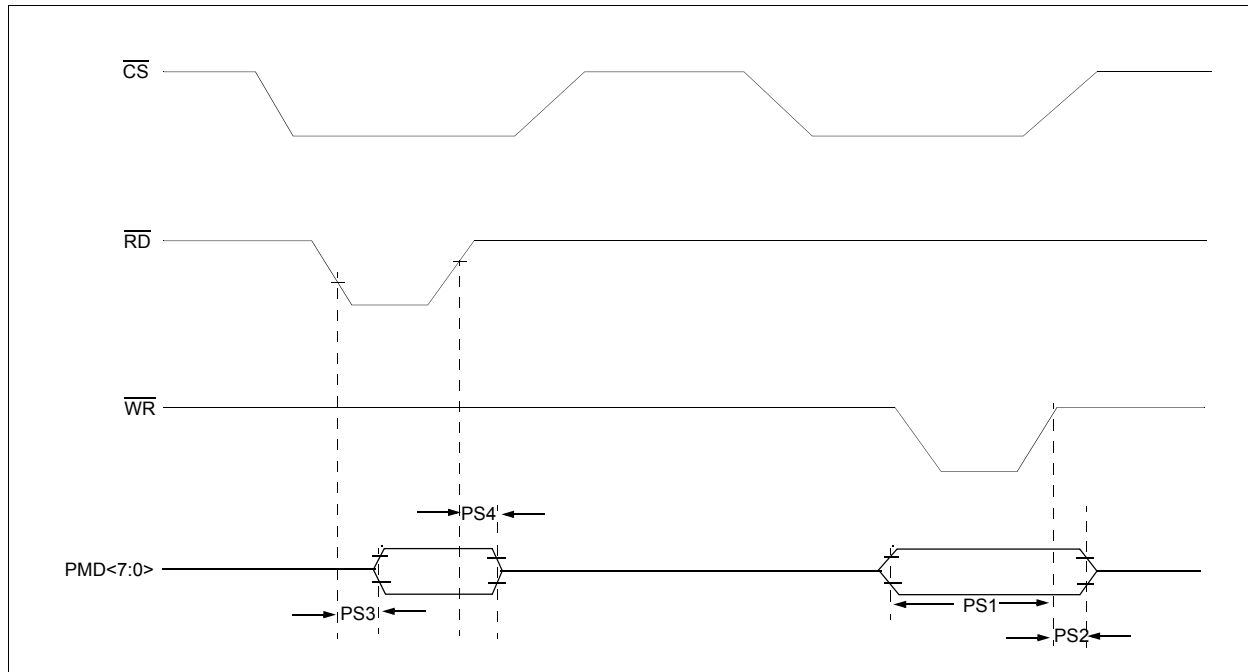


TABLE 31-50: SETTING TIME SPECIFICATIONS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | |
|--------------------|----------|---|---|-----|------|-------|------------|
| Param No. | Symbol | Characteristic | Min. | Typ | Max. | Units | Conditions |
| PS1 | TdtV2wrH | Data in Valid before $\overline{\text{WR}}$ or $\overline{\text{CS}}$ Inactive (setup time) | 20 | — | — | ns | |
| PS2 | TwrH2dtI | $\overline{\text{WR}}$ or $\overline{\text{CS}}$ Inactive to Data-In Invalid (hold time) | 20 | — | — | ns | |
| PS3 | TrdL2dtV | $\overline{\text{RD}}$ and $\overline{\text{CS}}$ to Active Data-Out Valid | — | — | 80 | ns | |
| PS4 | TrdH2dtI | $\overline{\text{RD}}$ Active or $\overline{\text{CS}}$ Inactive to Data-Out Invalid | 10 | — | 30 | ns | |

FIGURE 31-27: PARALLEL MASTER PORT READ TIMING DIAGRAM

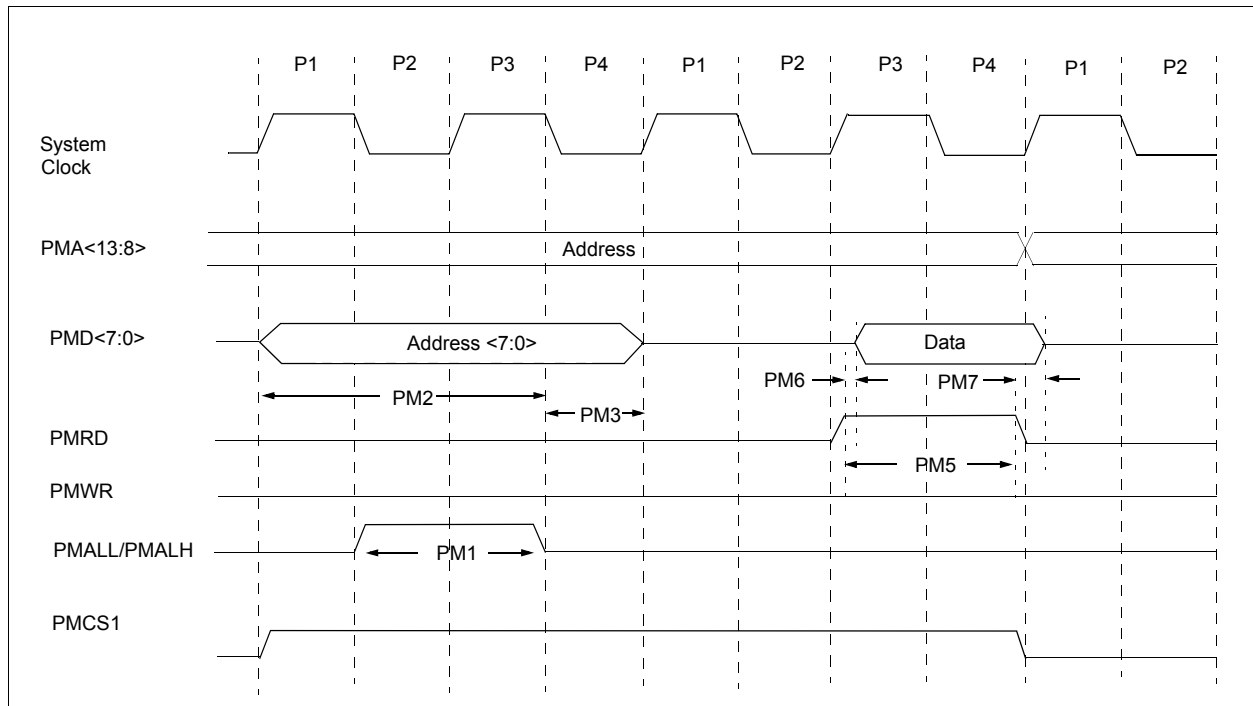


TABLE 31-51: PARALLEL MASTER PORT READ TIMING REQUIREMENTS

| AC CHARACTERISTICS | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|--------------------|--|---|----------|------|-------|------------|
| Param No. | Characteristic | Min. | Typ | Max. | Units | Conditions |
| PM1 | PMALL/PMALH Pulse Width | — | 0.5 Tcy | — | ns | |
| PM2 | Address Out Valid to PMALL/PMALH Invalid (address setup time) | — | 0.75 Tcy | — | ns | |
| PM3 | PMALL/PMALH Invalid to Address Out Invalid (address hold time) | — | 0.25 Tcy | — | ns | |
| PM5 | PMRD Pulse Width | — | 0.5 Tcy | — | ns | |
| PM6 | PMRD or PMENB Active to Data In Valid (data setup time) | — | — | — | ns | |
| PM7 | PMRD or PMENB Inactive to Data In Invalid (data hold time) | — | — | — | ns | |

FIGURE 31-28: PARALLEL MASTER PORT WRITE TIMING DIAGRAM

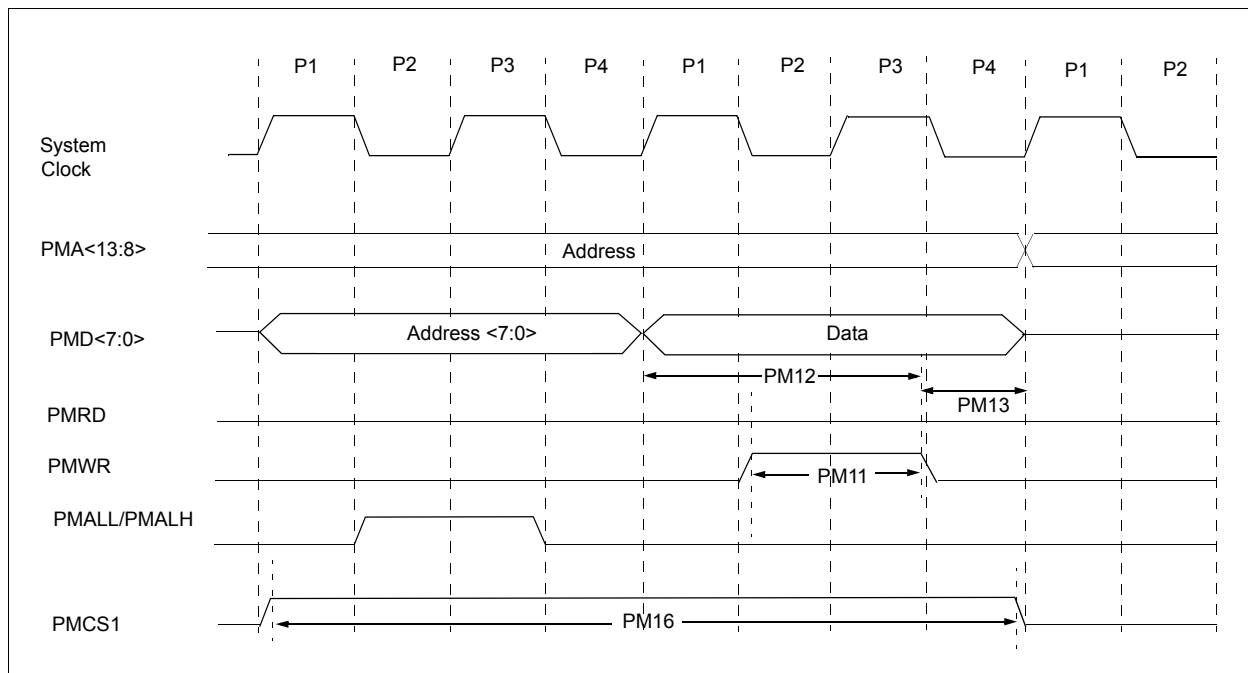


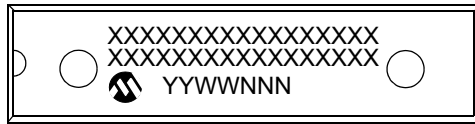
TABLE 31-52: PARALLEL MASTER PORT WRITE TIMING REQUIREMENTS

| AC CHARACTERISTICS | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended | | | | |
|--------------------|---|---|---------------------|------|-------|------------|
| Param No. | Characteristic | Min. | Typ | Max. | Units | Conditions |
| PM11 | PMWR Pulse Width | — | 0.5 T _{CY} | — | ns | |
| PM12 | Data Out Valid before PMWR or PMENB goes Inactive (data setup time) | — | — | — | ns | |
| PM13 | PMWR or PMEMB Invalid to Data Out Invalid (data hold time) | — | — | — | ns | |
| PM16 | PMCSx Pulse Width | T _{CY} - 5 | — | — | ns | |

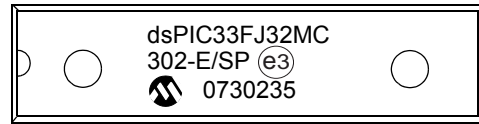
NOTES:

32.0 PACKAGING INFORMATION

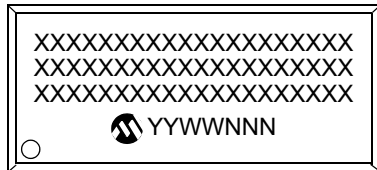
28-Lead SPDIP



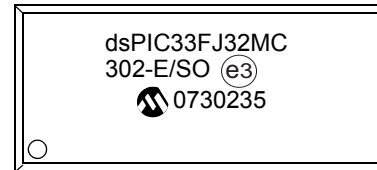
Example



28-Lead SOIC (.300")



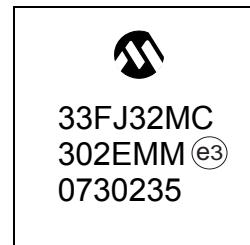
Example



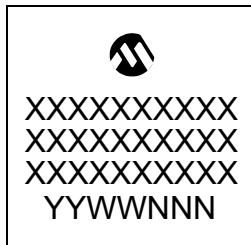
28-Lead QFN-S



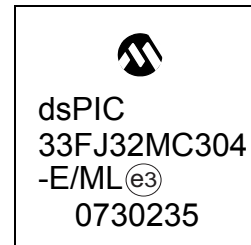
Example



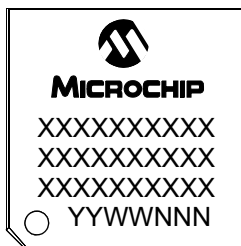
44-Lead QFN



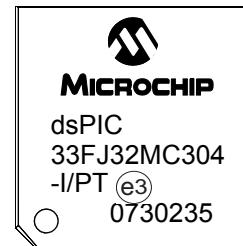
Example



44-Lead TQFP



Example

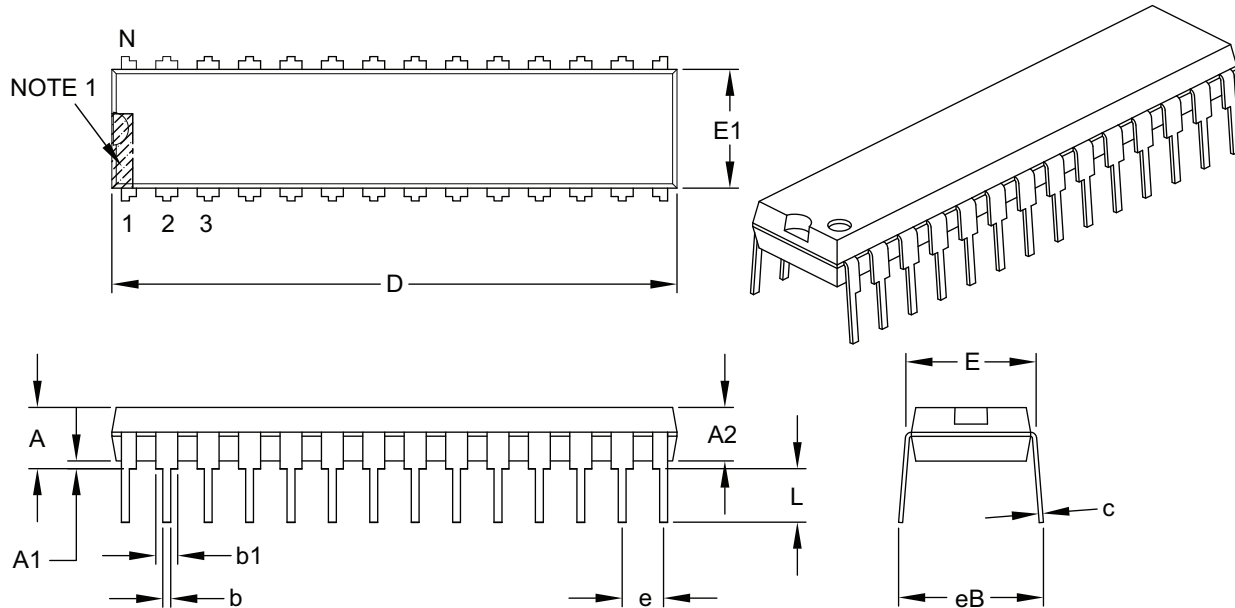


| | | |
|----------------|--|--|
| Legend: | XX...X | Customer-specific information |
| | Y | Year code (last digit of calendar year) |
| | YY | Year code (last 2 digits of calendar year) |
| | WW | Week code (week of January 1 is week '01') |
| | NNN | Alphanumeric traceability code |
| | (e3) | Pb-free JEDEC designator for Matte Tin (Sn) |
| | * | This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package. |
| Note: | If the full Microchip part number cannot be marked on one line, it is carried over to the next line, thus limiting the number of available characters for customer-specific information. | |

32.1 Package Details

28-Lead Skinny Plastic Dual In-Line (SP) – 300 mil Body [SPDIP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



| | | Units | INCHES | | |
|----------------------------|----|-------|----------|-------|-------|
| Dimension Limits | | | MIN | NOM | MAX |
| Number of Pins | N | | 28 | | |
| Pitch | e | | .100 BSC | | |
| Top to Seating Plane | A | | – | – | .200 |
| Molded Package Thickness | A2 | | .120 | .135 | .150 |
| Base to Seating Plane | A1 | | .015 | – | – |
| Shoulder to Shoulder Width | E | | .290 | .310 | .335 |
| Molded Package Width | E1 | | .240 | .285 | .295 |
| Overall Length | D | | 1.345 | 1.365 | 1.400 |
| Tip to Seating Plane | L | | .110 | .130 | .150 |
| Lead Thickness | c | | .008 | .010 | .015 |
| Upper Lead Width | b1 | | .040 | .050 | .070 |
| Lower Lead Width | b | | .014 | .018 | .022 |
| Overall Row Spacing § | eB | | – | – | .430 |

Notes:

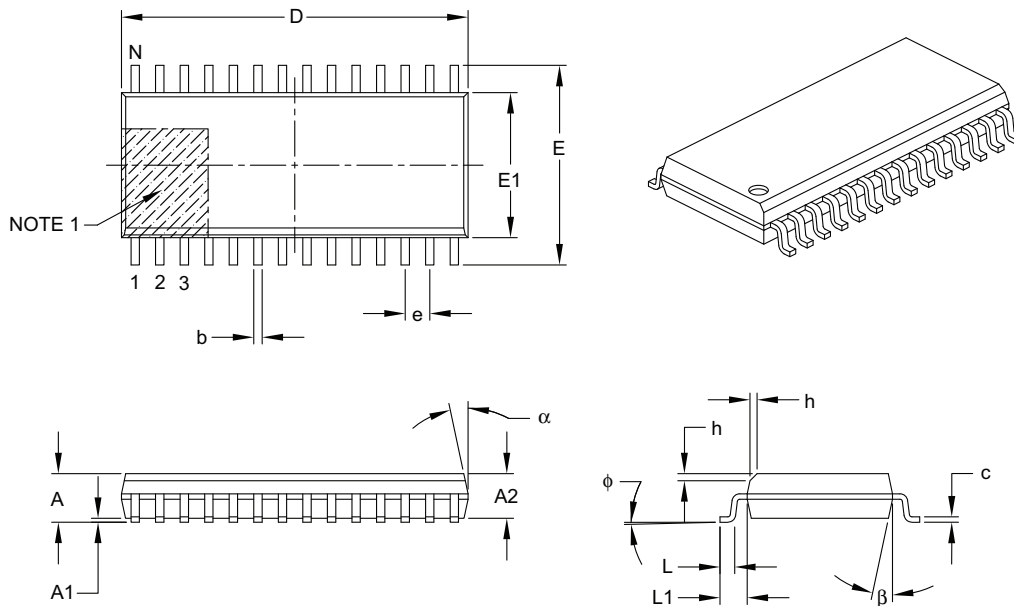
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-070B

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 | φ | 0° | – | 8° |
| Lead Thickness | c | 0.18 | – | 0.33 |
| Lead Width | b | 0.31 | – | 0.51 |
| Mold Draft Angle Top | α | 5° | – | 15° |
| Mold Draft Angle Bottom | β | 5° | – | 15° |

Notes:

- 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.

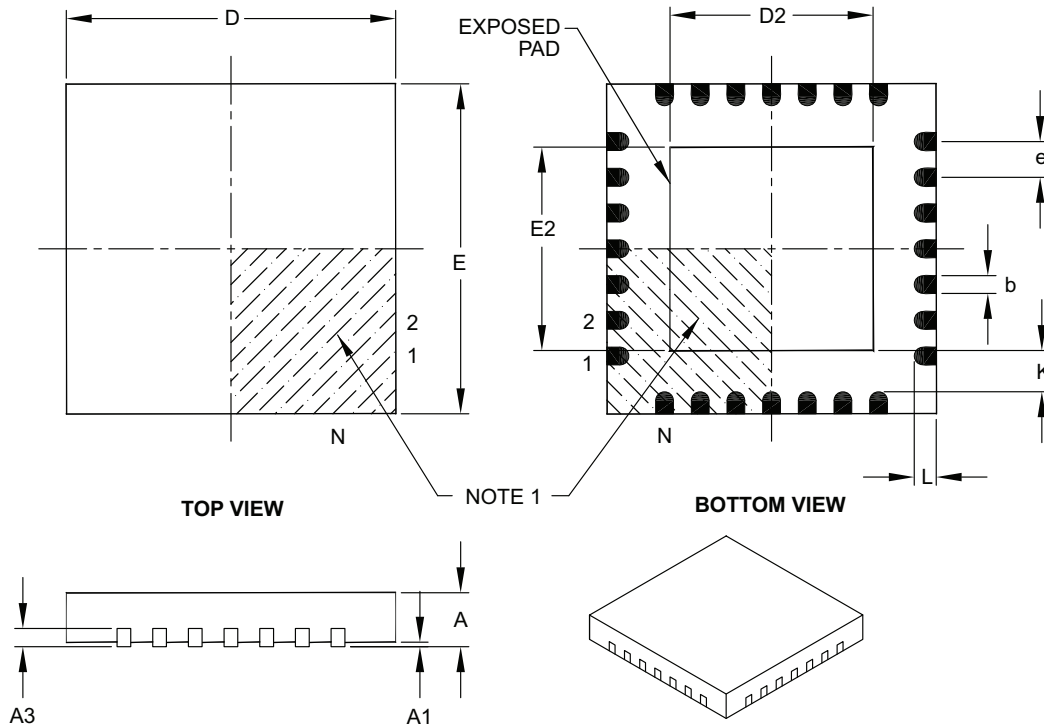
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

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

Microchip Technology Drawing C04-052E

28-Lead Plastic Quad Flat, No Lead Package (MM) – 6x6x0.9 mm Body [QFN-S]
with 0.40 mm Contact Length

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 | 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 | 4.70 |
| Overall Length | D | 6.00 BSC | | |
| Exposed Pad Length | D2 | 3.65 | 3.70 | 4.70 |
| Contact Width | b | 0.23 | 0.38 | 0.43 |
| 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.

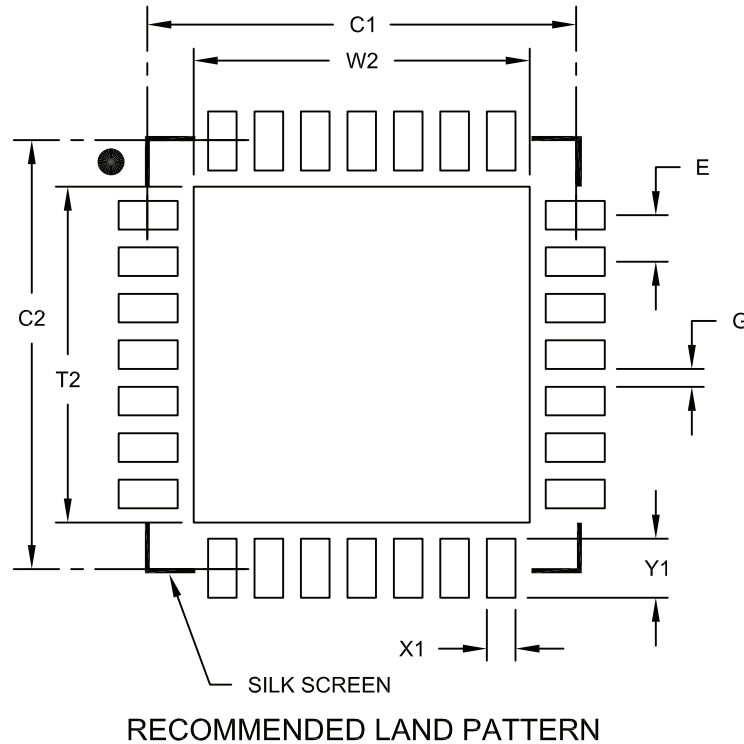
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

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

Microchip Technology Drawing C04-124B

**28-Lead Plastic Quad Flat, No Lead Package (MM) – 6x6x0.9 mm Body [QFN-S]
with 0.40 mm Contact Length**

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 |
| Contact Pitch | E | 0.65 BSC | | |
| Optional Center Pad Width | W2 | | | 4.70 |
| Optional Center Pad Length | T2 | | | 4.70 |
| Contact Pad Spacing | C1 | | 6.00 | |
| Contact Pad Spacing | C2 | | 6.00 | |
| Contact Pad Width (X28) | X1 | | | 0.40 |
| Contact Pad Length (X28) | Y1 | | | 0.85 |
| Distance Between Pads | G | 0.25 | | |

Notes:

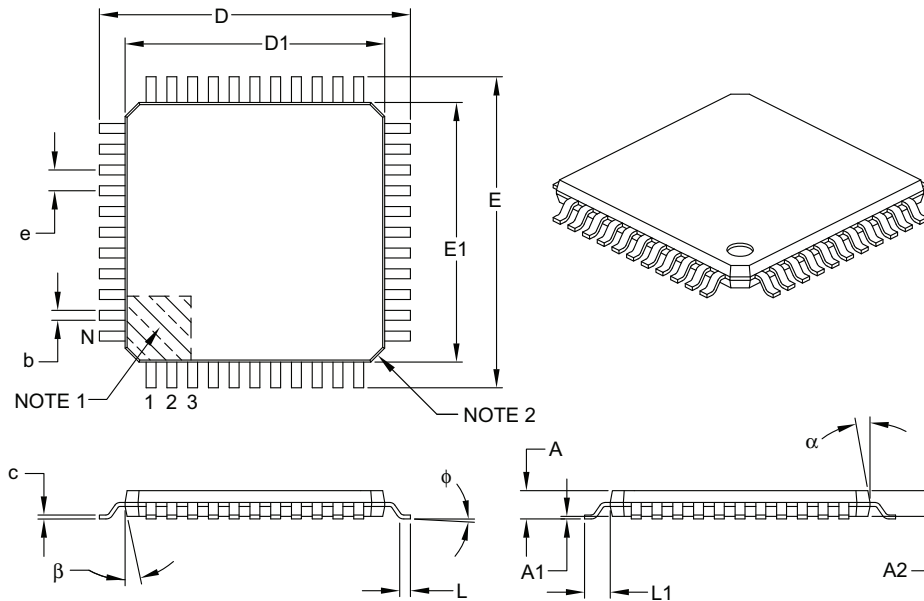
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2124A

44-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



| | | Units | MILLIMETERS | | |
|--------------------------|----------|-------|-------------|------|------|
| Dimension Limits | | | MIN | NOM | MAX |
| Number of Leads | N | | 44 | | |
| Lead Pitch | e | | 0.80 BSC | | |
| Overall Height | A | | – | – | 1.20 |
| Molded Package Thickness | A2 | | 0.95 | 1.00 | 1.05 |
| Standoff | A1 | | 0.05 | – | 0.15 |
| Foot Length | L | | 0.45 | 0.60 | 0.75 |
| Footprint | L1 | | 1.00 REF | | |
| Foot Angle | ϕ | | 0° | 3.5° | 7° |
| Overall Width | E | | 12.00 BSC | | |
| Overall Length | D | | 12.00 BSC | | |
| Molded Package Width | E1 | | 10.00 BSC | | |
| Molded Package Length | D1 | | 10.00 BSC | | |
| Lead Thickness | c | | 0.09 | – | 0.20 |
| Lead Width | b | | 0.30 | 0.37 | 0.45 |
| Mold Draft Angle Top | α | | 11° | 12° | 13° |
| Mold Draft Angle Bottom | β | | 11° | 12° | 13° |

Notes:

- 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.

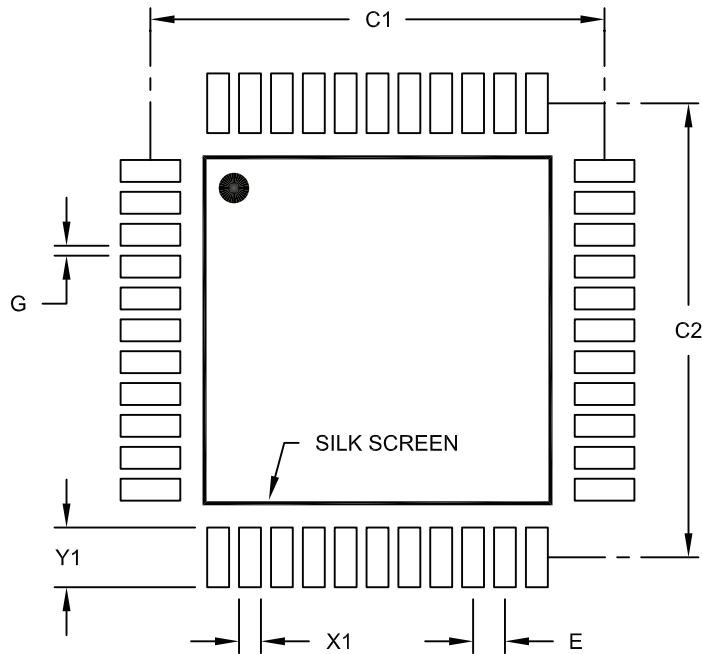
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

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

Microchip Technology Drawing C04-076B

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:

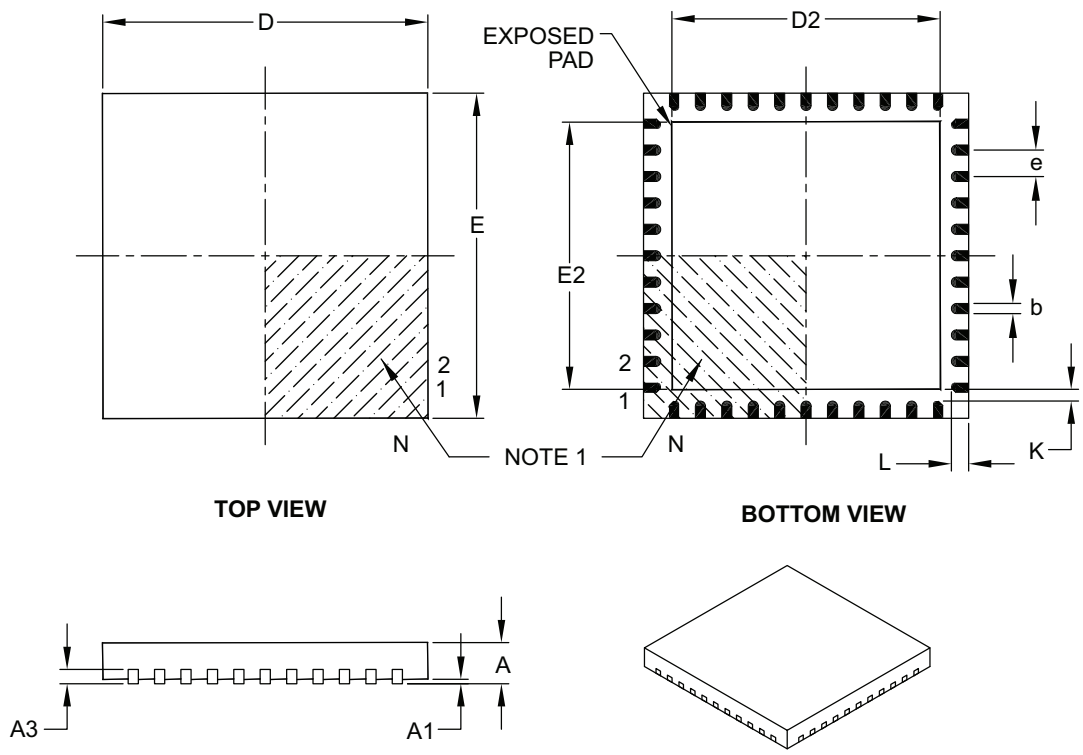
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2076A

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>



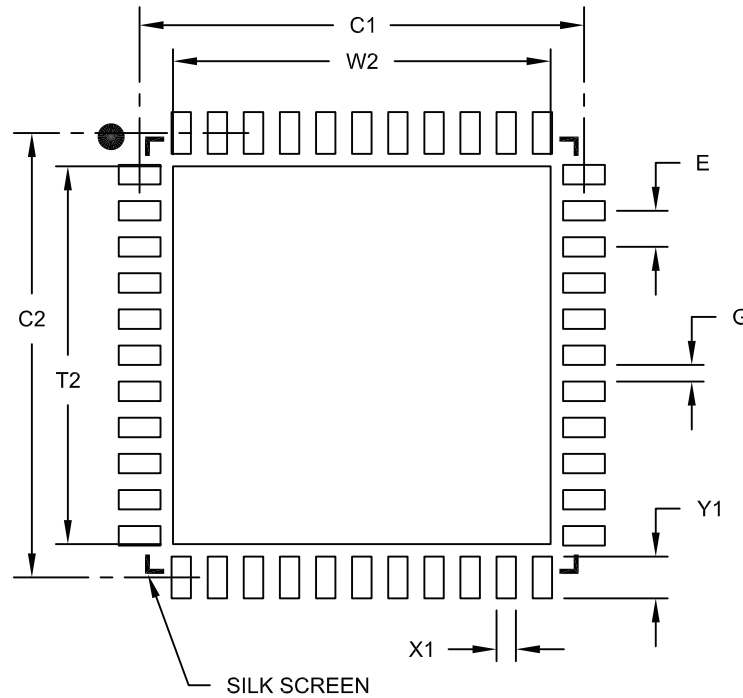
| Dimension Limits | Units | MILLIMETERS | | |
|------------------------|-------|-------------|------|------|
| | | MIN | NOM | MAX |
| Number of Pins | N | 44 | | |
| Pitch | e | 0.65 BSC | | |
| Overall Height | A | 0.80 | 0.90 | 1.00 |
| Standoff | A1 | 0.00 | 0.02 | 0.05 |
| Contact Thickness | A3 | 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

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

NOTES:

APPENDIX A: REVISION HISTORY

Revision A (August 2007)

Initial release of this document.

Revision B (March 2008)

This revision includes minor typographical and formatting changes throughout the data sheet text. In addition, redundant information was removed that is now available in the respective chapters of the *dsPIC33F Family Reference Manual*, which can be obtained from the Microchip website (www.microchip.com).

The major changes are referenced by their respective section in the following table.

TABLE A-1: MAJOR SECTION UPDATES

| Section Name | Update Description |
|---|--|
| "High-Performance, 16-bit Digital Signal Controllers" | Note 1 added to all pin diagrams (see "Pin Diagrams") Add External Interrupts column and Note 4 to the "dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 Controller Families" table |
| Section 1.0 "Device Overview" | Updated parameters PMA0, PMA1, and PMD0 through PMPD7 (Table 1-1) |
| Section 3.0 "Memory Organization" | Updated FAEN bits in Table 4-8 |
| Section 6.0 "Interrupt Controller" | IFS0-IFS04 changed to IFSx (see Section 6.3.2 "IFSx") IEC0-IEC4 changed to IECx (see Section 6.3.3 "IECx") IPC0-IPC19 changed to IPCx (see Section 6.3.4 "IPCx") |
| Section 7.0 "Direct Memory Access (DMA)" | Updated parameter PMP (see Table 8-1) |
| Section 8.0 "Oscillator Configuration" | Updated the third clock source item (External Clock) in Section 8.1.1 "System Clock Sources" Updated TUN<5:0> (OSCTUN<5:0>) bit description (see Register 8-4) |
| Section 21.0 "10-bit/12-bit Analog-to-Digital Converter (ADC1)" | Added Note 2 to Figure 21-3 |
| Section 27.0 "Special Features" | Added Note 2 to Figure 27-1 Added parameter FICD in Table 27-1 Added parameters BKBUG, COE, JTAGEN, and ICS in Table 27-2 Added Note after second paragraph in Section 27.2 "On-Chip Voltage Regulator" |

TABLE A-1: MAJOR SECTION UPDATES (CONTINUED)

| Section Name | Update Description |
|--|---|
| Section 30.0 “Electrical Characteristics” | Updated Max MIPS for temperature range of -40°C to +125°C in Table 30-1 Updated typical values in Thermal Packaging Characteristics in Table 30-3 Added parameters DI11 and DI12 to Table 30-9 Updated minimum values for parameters D136 (TRW) and D137 (TPE) and removed typical values in Table 30-12 Added Extended temperature range to Table 30-13 Updated Note 2 in Table 30-38 Updated parameter AD63 and added Note 3 to Table 30-42 and Table 30-43 |

Revision C (May 2009)

This revision includes minor typographical and formatting changes throughout the data sheet text.

Global changes include:

- Changed all instances of OSCI to OSC1 and OSCO to OSC2
- Changed all instances of VDDCORE and VDDCORE/VCAP to VCAP/VDDCORE

The other changes are referenced by their respective section in the following table.

TABLE A-2: MAJOR SECTION UPDATES

| Section Name | Update Description |
|--|--|
| “High-Performance, 16-bit Digital Signal Controllers” | Updated all pin diagrams to denote the pin voltage tolerance (see “Pin Diagrams”). Added Note 2 to the 28-Pin QFN-S and 44-Pin QFN pin diagrams, which references pin connections to Vss. |
| Section 1.0 “Device Overview” | Updated AVDD in the PINOUT I/O Descriptions (see Table 1-1). |
| Section 2.0 “Guidelines for Getting Started with 16-bit Digital Signal Controllers” | Added new section to the data sheet that provides guidelines on getting started with 16-bit Digital Signal Controllers. |
| Section 3.0 “CPU” | Updated CPU Core Block Diagram with a connection from the DSP Engine to the Y Data Bus (see Figure 3-1). Vertically extended the X and Y Data Bus lines in the DSP Engine Block Diagram (see Figure 3-3). |
| Section 4.0 “Memory Organization” | Updated Reset value for CORCON in the CPU Core Register Map (see Table 4-1). Removed the FLTA1IE bit (IEC3) from the Interrupt Controller Register Map (see Table 4-4). Updated bit locations for RPINR25 in the Peripheral Pin Select Input Register Map (see Table 4-24). Updated the Reset value for CLKDIV in the System Control Register Map (see Table 4-36). |
| Section 5.0 “Flash Program Memory” | Updated Section 5.3 “Programming Operations” with programming time formula. |
| Section 9.0 “Oscillator Configuration” | Updated the Oscillator System Diagram and added Note 2 (see Figure 9-1). Updated default bit values for DOZE<2:0> and FRCDIV<2:0> in the Clock Divisor (CLKDIV) Register (see Register 9-2). Added a paragraph regarding FRC accuracy at the end of Section 9.1.1 “System Clock Sources” . Added Note 3 to Section 9.2.2 “Oscillator Switching Sequence” . Added Note 1 to the FRC Oscillator Tuning (OSCTUN) Register (see Register 9-4). |
| Section 10.0 “Power-Saving Features” | Added the following registers: <ul style="list-style-type: none"> • PMD1: Peripheral Module Disable Control Register 1 (Register 10-1) • PMD2: Peripheral Module Disable Control Register 2 (Register 10-2) • PMD3: Peripheral Module Disable Control Register 3 (Register 10-3) |

TABLE A-2: MAJOR SECTION UPDATES (CONTINUED)

| Section Name | Update Description |
|--|---|
| Section 11.0 “I/O Ports” | <p>Removed Table 11-1 and added reference to pin diagrams for I/O pin availability and functionality.</p> <p>Added paragraph on ADPCFG register default values to Section 11.2 “Configuring Analog Port Pins”.</p> <p>Added Note box regarding PPS functionality with input mapping to Section 11.4.2.1 “Input Mapping”.</p> |
| Section 18.0 “Serial Peripheral Interface (SPI)” | <p>Added Note 2 and 3 to the SPIxCON1 register (see Register 18-2).</p> |
| Section 20.0 “Universal Asynchronous Receiver Transmitter (UART)” | <p>Updated the Notes in the UxMode register (see Register 20-1).</p> <p>Updated the UTXINV bit settings in the UxSTA register and added Note 1 (see Register 20-2).</p> |
| Section 21.0 “Enhanced CAN (ECAN™) Module” | <p>Changed bit 11 in the ECAN Control Register 1 (CiCTRL1) to Reserved (see Register 21-1).</p> |
| Section 22.0 “10-bit/12-bit Analog-to-Digital Converter (ADC1)” | <p>Replaced the ADC1 Module Block Diagrams with new diagrams (see Figure 22-1 and Figure 22-2).</p> <p>Updated bit values for ADCS<7:0> and added Notes 1 and 2 to the ADC1 Control Register 3 (AD1CON3) (see Register 22-3).</p> <p>Added Note 2 to the ADC1 Input Scan Select Register Low (AD1CSSL) (see Register 22-7).</p> <p>Added Note 2 to the ADC1 Port Configuration Register Low (AD1PCFGL) (see Register 22-8).</p> |
| Section 23.0 “Audio Digital-to-Analog Converter (DAC)” | <p>Updated the midpoint voltage in the last sentence of the first paragraph.</p> <p>Updated the voltage swing values in the last sentence of the last paragraph in Section 23.3 “DAC Output Format”.</p> |
| Section 24.0 “Comparator Module” | <p>Updated the Comparator Voltage Reference Block Diagram (see Figure 24-2).</p> |
| Section 25.0 “Real-Time Clock and Calendar (RTCC)” | <p>Updated the minimum positive adjust value for CAL<7:0> in the RTCC Calibration and Configuration (RCFGCAL) Register (see Register 25-1).</p> |
| Section 28.0 “Special Features” | <p>Added Note 1 to the Device Configuration Register Map (see Table 28-1).</p> <p>Updated Note 1 in the dsPIC33F Configuration Bits Description (see Table 28-2).</p> |

TABLE A-2: MAJOR SECTION UPDATES (CONTINUED)

| Section Name | Update Description |
|---|--|
| <p>Section 31.0 “Electrical Characteristics”</p> | <p>Updated Typical values for Thermal Packaging Characteristics (see Table 31-3).</p> <p>Updated Min and Max values for parameter DC12 (RAM Data Retention Voltage) and added Note 4 (see Table 31-4).</p> <p>Updated Power-Down Current Max values for parameters DC60b and DC60c (see Table 31-7).</p> <p>Updated Characteristics for I/O Pin Input Specifications (see Table 31-9).</p> <p>Updated Program Memory values for parameters 136, 137, and 138 (renamed to 136a, 137a, and 138a), added parameters 136b, 137b, and 138b, and added Note 2 (see Table 31-12).</p> <p>Added parameter OS42 (GM) to the External Clock Timing Requirements (see Table 31-16).</p> <p>Updated Watchdog Timer Time-out Period parameter SY20 (see Table 31-21).</p> <p>Removed VOMIN, renamed VOMAX to VO, and updated the Min and Max values in the Audio DAC Module Specifications (see Table 31-44).</p> |

NOTES:

INDEX

A

A/D Converter 267
 DMA 267
 Initialization 267
 Key Features 267
 AC Characteristics 346
 Internal RC Accuracy 348
 Load Conditions 346
 ADC Module
 ADC11 Register Map 49, 50
 Alternate Vector Table (AIVT) 85
 Arithmetic Logic Unit (ALU) 30
 Assembler
 MPASM Assembler 334

B

Barrel Shifter 34
 Bit-Reversed Addressing 65
 Example 65
 Implementation 65
 Sequence Table (16-Entry) 66
 Block Diagrams
 16-bit Timer1 Module 189
 A/D Module 268, 269
 Connections for On-Chip Voltage Regulator 319
 Device Clock 139, 141
 DSP Engine 31
 dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04,
 and dsPIC33FJ128MCX02/X04 14
 dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04,
 and dsPIC33FJ128MCX02/X04 CPU Core 24
 ECAN Module 242
 Input Capture 197
 Output Compare 199
 PLL 141
 PWM Module 204, 205
 Quadrature Encoder Interface 217
 Reset System 77
 Shared Port Structure 157
 SPI 221
 Timer2 (16-bit) 191
 Timer2/3 (32-bit) 193
 UART 235
 Watchdog Timer (WDT) 320

C

C Compilers
 MPLAB C18 334
 MPLAB C30 334
 Clock Switching 149
 Enabling 149
 Sequence 149
 Code Examples
 Erasing a Program Memory Page 75
 Initiating a Programming Sequence 76
 Loading Write Buffers 76
 Port Write/Read 158
 PWRSAV Instruction Syntax 151
 Code Protection 315, 321
 Comparator Module 281
 Configuration Bits 315
 Configuration Register Map 315
 Configuring Analog Port Pins 158
 CPU
 Control Register 26

CPU Clocking System 140
 PLL Configuration 141
 Selection 140
 Sources 140
 Customer Change Notification Service 402
 Customer Notification Service 402
 Customer Support 402

D

Data Accumulators and Adder/Subtractor 32
 Data Space Write Saturation 34
 Overflow and Saturation 32
 Round Logic 33
 Write Back 33
 Data Address Space 37
 Alignment 37
 Memory Map for dsPIC33FJ128MC202/204
 and dsPIC33FJ64MC202/204 Devices with
 8 KB RAM 39
 Memory Map for dsPIC33FJ128MC802/804
 and dsPIC33FJ64MC802/804 Devices with
 16 KB RAM 40
 Memory Map for dsPIC33FJ32MC302/304
 Devices with 4 KB RAM 38
 Near Data Space 37
 Software Stack 62
 Width 37
 DC Characteristics 338
 I/O Pin Input Specifications 343
 I/O Pin Output Specifications 344
 Idle Current (ID0ZE) 342
 Idle Current (IDLE) 341
 Operating Current (IDD) 340
 Power-Down Current (IPD) 342
 Program Memory 345
 Temperature and Voltage Specifications 339
 Development Support 333
 DMA Module
 DMA Register Map 51
 DMAC Registers 129
 DMAxCNT 129
 DMAxCON 129
 DMAxPAD 129
 DMAxREQ 129
 DMAxSTA 129
 DMAxSTB 129
 Doze Mode 152
 DSP Engine 30
 Multiplier 32

E

ECAN Module
 CiBUFNT1 register 253
 CiBUFNT2 register 254
 CiBUFNT3 register 254
 CiBUFNT4 register 255
 CiCFG1 register 251
 CiCFG2 register 252
 CiCTRL1 register 244
 CiCTRL2 register 245
 CiEC register 251
 CiFCTRL register 247
 CiFEN1 register 253
 CiFIFO register 248
 CiFMSKSEL1 register 257
 CiFMSKSEL2 register 258

| | |
|---|-----|
| CiINTE register | 250 |
| CiINTF register..... | 249 |
| CiRXFnEID register | 257 |
| CiRXFnSID register | 256 |
| CiRXFUL1 register..... | 260 |
| CiRXFUL2 register..... | 260 |
| CiRXMnEID register..... | 259 |
| CiRXMnSID register..... | 259 |
| CiRXOVF1 register..... | 261 |
| CiRXOVF2 register..... | 261 |
| CiTRmnCON register..... | 262 |
| CiVEC register | 246 |
| ECAN1 Register Map (C1CTRL1.WIN = 0 or 1)..... | 53 |
| ECAN1 Register Map (C1CTRL1.WIN = 0)..... | 53 |
| ECAN1 Register Map (C1CTRL1.WIN = 1)..... | 54 |
| Frame Types..... | 241 |
| Modes of Operation | 243 |
| Overview | 241 |
| ECAN Registers | |
| Acceptance Filter Enable Register (CiFEN1)..... | 253 |
| Acceptance Filter Extended Identifier Register n (CiRXFnEID)..... | 257 |
| Acceptance Filter Mask Extended Identifier Register n (CiRXMnEID)..... | 259 |
| Acceptance Filter Mask Standard Identifier Register n (CiRXMnSID)..... | 259 |
| Acceptance Filter Standard Identifier Register n (CiRXFnSID)..... | 256 |
| Baud Rate Configuration Register 1 (CiCFG1)..... | 251 |
| Baud Rate Configuration Register 2 (CiCFG2)..... | 252 |
| Control Register 1 (CiCTRL1)..... | 244 |
| Control Register 2 (CiCTRL2)..... | 245 |
| FIFO Control Register (CiFCTRL) | 247 |
| FIFO Status Register (CiFIFO) | 248 |
| Filter 0-3 Buffer Pointer Register (CiBUPPNT1)..... | 253 |
| Filter 12-15 Buffer Pointer Register (CiBUPPNT4)..... | 255 |
| Filter 15-8 Mask Selection Register (CiFMSKSEL2)..... | 258 |
| Filter 4-7 Buffer Pointer Register (CiBUPPNT2)..... | 254 |
| Filter 7-0 Mask Selection Register (CiFMSKSEL1)..... | 257 |
| Filter 8-11 Buffer Pointer Register (CiBUPPNT3)..... | 254 |
| Interrupt Code Register (CiVEC) | 246 |
| Interrupt Enable Register (CiINTE)..... | 250 |
| Interrupt Flag Register (CiINTF)..... | 249 |
| Receive Buffer Full Register 1 (CiRXFUL1)..... | 260 |
| Receive Buffer Full Register 2 (CiRXFUL2)..... | 260 |
| Receive Buffer Overflow Register 2 (CiRXOVF2)..... | 261 |
| Receive Overflow Register (CiRXOVF1) | 261 |
| ECAN Transmit/Receive Error Count Register (CiEC)..... | 251 |
| ECAN TX/RX Buffer m Control Register (CiTRmnCON) .. | 262 |
| Electrical Characteristics..... | 337 |
| AC..... | 346 |
| Enhanced CAN Module..... | 241 |
| Equations | |
| Device Operating Frequency | 140 |
| Errata | 11 |
| F | |
| Flash Program Memory..... | 71 |
| Control Registers..... | 72 |
| Operations | 72 |
| Programming Algorithm | 75 |
| RTSP Operation..... | 72 |
| Table Instructions..... | 71 |
| Flexible Configuration | 315 |

| | |
|--|----------|
| I | |
| I/O Ports..... | 157 |
| Parallel I/O (PIO) | 157 |
| Write/Read Timing..... | 158 |
| i ² C | |
| Operating Modes | 227 |
| Registers | 227 |
| In-Circuit Debugger..... | 321 |
| In-Circuit Emulation | 315 |
| In-Circuit Serial Programming (ICSP)..... | 315, 321 |
| Input Capture..... | 197 |
| Registers | 198 |
| Input Change Notification | 158 |
| Instruction Addressing Modes | 62 |
| File Register Instructions | 62 |
| Fundamental Modes Supported | 63 |
| MAC Instructions | 63 |
| MCU Instructions | 62 |
| Move and Accumulator Instructions..... | 63 |
| Other Instructions | 63 |
| Instruction Set | |
| Overview | 328 |
| Summary | 325 |
| Instruction-Based Power-Saving Modes..... | 151 |
| Idle..... | 152 |
| Sleep | 151 |
| Internal RC Oscillator | |
| Use with WDT..... | 320 |
| Internet Address | 402 |
| Interrupt Control and Status Registers | 89 |
| IECx..... | 89 |
| IFSx | 89 |
| INTCON1..... | 89 |
| INTCON2..... | 89 |
| IPCx..... | 89 |
| Interrupt Setup Procedures..... | 126 |
| Initialization | 126 |
| Interrupt Disable | 126 |
| Interrupt Service Routine..... | 126 |
| Trap Service Routine..... | 126 |
| Interrupt Vector Table (IVT)..... | 85 |
| Interrupts Coincident with Power Save Instructions | 152 |
| J | |
| JTAG Boundary Scan Interface..... | 315 |
| JTAG Interface..... | 321 |
| M | |
| Memory Organization | 35 |
| Microchip Internet Web Site..... | 402 |
| Modes of Operation | |
| Disable..... | 243 |
| Initialization | 243 |
| Listen All Messages..... | 243 |
| Listen Only..... | 243 |
| Loopback | 243 |
| Normal Operation | 243 |
| Modulo Addressing..... | 64 |
| Applicability..... | 65 |
| Operation Example..... | 64 |
| Start and End Address | 64 |
| W Address Register Selection | 64 |
| Motor Control PWM | 203 |

| | |
|--|-----|
| Motor Control PWM Module | |
| 2-Output Register Map..... | 47 |
| 6-Output Register Map..... | 46 |
| MPLAB ASM30 Assembler, Linker, Librarian | 334 |
| MPLAB ICD 2 In-Circuit Debugger | 335 |
| MPLAB ICE 2000 High-Performance Universal | |
| In-Circuit Emulator | 335 |
| MPLAB Integrated Development Environment Software .. | 333 |
| MPLAB PM3 Device Programmer | 335 |
| MPLAB REAL ICE In-Circuit Emulator System..... | 335 |
| MPLINK Object Linker/MPLIB Object Librarian | 334 |

N

| | |
|-------------------|----|
| NVM Module | |
| Register Map..... | 61 |

O

| | |
|--------------------------------|-----|
| Open-Drain Configuration | 158 |
| Output Compare | 199 |

P

| | |
|--|--------|
| Packaging | 382 |
| Details | 383 |
| Marking | 382 |
| Peripheral Module Disable (PMD) | 152 |
| PICSTART Plus Development Programmer | 336 |
| Pinout I/O Descriptions (table) | 15 |
| PMD Module | |
| Register Map..... | 61 |
| PORTA | |
| Register Map..... | 59, 60 |
| PORTB | |
| Register Map..... | 60 |
| Power-on Reset (POR) | 82 |
| Power-Saving Features | 151 |
| Clock Frequency and Switching..... | 151 |
| Program Address Space..... | 35 |
| Construction..... | 67 |
| Data Access from Program Memory | |
| Using Program Space Visibility..... | 70 |
| Data Access from Program Memory | |
| Using Table Instructions | 69 |
| Data Access from, Address Generation..... | 68 |
| Memory Map | 35 |
| Table Read Instructions | |
| TBLRDH | 69 |
| TBLRDL | 69 |
| Visibility Operation | 70 |
| Program Memory | |
| Interrupt Vector | 36 |
| Organization..... | 36 |
| Reset Vector | 36 |

Q

| | |
|---|-----|
| Quadrature Encoder Interface (QEI)..... | 217 |
| Quadrature Encoder Interface (QEI) Module | |
| Register Map..... | 47 |

R

| | |
|---|-----|
| Reader Response | 403 |
| Register Map | |
| CRC | 59 |
| Dual Comparator..... | 59 |
| Parallel Master/Slave Port | 58 |
| Real-Time Clock and Calendar..... | 59 |
| Registers | |
| AD1CHS0 (ADC1 Input Channel 0 Select)..... | 278 |
| AD1CHS123 (ADC1 Input Channel 1, 2, 3 Select) ... | 276 |

| | |
|---|---------------|
| AD1CON1 (ADC1 Control 1)..... | 271 |
| AD1CON2 (ADC1 Control 2)..... | 273 |
| AD1CON3 (ADC1 Control 3)..... | 274 |
| AD1CON4 (ADC1 Control 4)..... | 275 |
| AD1CSSL (ADC1 Input Scan Select Low) | 280 |
| AD1PCFGL (ADC1 Port Configuration Low)..... | 280 |
| CiBUFNT1 (ECAN Filter 0-3 Buffer Pointer)..... | 253 |
| CiBUFNT2 (ECAN Filter 4-7 Buffer Pointer)..... | 254 |
| CiBUFNT3 (ECAN Filter 8-11 Buffer Pointer)..... | 254 |
| CiBUFNT4 (ECAN Filter 12-15 Buffer Pointer)..... | 255 |
| CiCFG1 (ECAN Baud Rate Configuration 1)..... | 251 |
| CiCFG2 (ECAN Baud Rate Configuration 2)..... | 252 |
| CiCTRL1 (ECAN Control 1)..... | 244 |
| CiCTRL2 (ECAN Control 2)..... | 245 |
| CiEC (ECAN Transmit/Receive Error Count)..... | 251 |
| CiFCTRL (ECAN FIFO Control) | 247 |
| CiFEN1 (ECAN Acceptance Filter Enable)..... | 253 |
| CiFIFO (ECAN FIFO Status) | 248 |
| CiFMSKSEL1 (ECAN Filter 7-0 Mask Selection) ... | 257, 258 |
| CiINTE (ECAN Interrupt Enable)..... | 250 |
| CiINTF (ECAN Interrupt Flag) | 249 |
| CiRXFnEID (ECAN Acceptance Filter n | |
| Extended Identifier) | 257 |
| CiRXFnSID (ECAN Acceptance Filter n | |
| Standard Identifier)..... | 256 |
| CiRXFUL1 (ECAN Receive Buffer Full 1)..... | 260 |
| CiRXFUL2 (ECAN Receive Buffer Full 2)..... | 260 |
| CiRXMnEID (ECAN Acceptance Filter Mask n | |
| Extended Identifier) | 259 |
| CiRXMnSID (ECAN Acceptance Filter Mask n | |
| Standard Identifier)..... | 259 |
| CiRXOVF1 (ECAN Receive Buffer Overflow 1)..... | 261 |
| CiRXOVF2 (ECAN Receive Buffer Overflow 2)..... | 261 |
| CiTRBnSID (ECAN Buffer n Standard Identifier).... | 263, 264, 266 |
| CiTRmnCON (ECAN TX/RX Buffer m Control)..... | 262 |
| CiVEC (ECAN Interrupt Code) | 246 |
| CLKDIV (Clock Divisor) | 145 |
| CORCON (Core Control)..... | 28, 90 |
| DFLTCON (QEI Control) | 220 |
| DMACS0 (DMA Controller Status 0) | 134 |
| DMACS1 (DMA Controller Status 1) | 136 |
| DMAxCNT (DMA Channel x Transfer Count)..... | 133 |
| DMAxCON (DMA Channel x Control)..... | 130 |
| DMAxPAD (DMA Channel x Peripheral Address) ... | 133 |
| DMAxREQ (DMA Channel x IRQ Select) | 131 |
| DMAxSTA (DMA Channel x RAM Start Address A) . | 132 |
| DMAxSTB (DMA Channel x RAM Start Address B) . | 132 |
| DSADR (Most Recent DMA RAM Address) | 137 |
| I2CxCON (I2Cx Control)..... | 229 |
| I2CxMSK (I2Cx Slave Mode Address Mask)..... | 233 |
| I2CxSTAT (I2Cx Status)..... | 231 |
| IFS0 (Interrupt Flag Status 0)..... | 94, 101 |
| IFS1 (Interrupt Flag Status 1)..... | 96, 103 |
| IFS2 (Interrupt Flag Status 2)..... | 98, 105 |
| IFS3 (Interrupt Flag Status 3)..... | 99, 106 |
| IFS4 (Interrupt Flag Status 4)..... | 100, 107 |
| INTCON1 (Interrupt Control 1) | 91 |
| INTCON2 (Interrupt Control 2) | 93 |
| INTTREG Interrupt Control and Status Register | 125 |
| IPC0 (Interrupt Priority Control 0)..... | 108 |
| IPC1 (Interrupt Priority Control 1)..... | 109 |
| IPC11 (Interrupt Priority Control 11)..... | 118 |
| IPC14 (Interrupt Priority Control 14)..... | 119 |
| IPC15 (Interrupt Priority Control 15)..... | 120 |
| IPC16 (Interrupt Priority Control 16)..... | 121 |
| IPC17 (Interrupt Priority Control 17)..... | 122 |

| | |
|--|--------------------|
| IPC18 (Interrupt Priority Control 18) | 123, 124 |
| IPC2 (Interrupt Priority Control 2) | 110 |
| IPC3 (Interrupt Priority Control 3) | 111 |
| IPC4 (Interrupt Priority Control 4) | 112 |
| IPC5 (Interrupt Priority Control 5) | 113 |
| IPC6 (Interrupt Priority Control 6) | 114 |
| IPC7 (Interrupt Priority Control 7) | 115 |
| IPC8 (Interrupt Priority Control 8) | 116 |
| IPC9 (Interrupt Priority Control 9) | 117 |
| NVMCON (Flash Memory Control) | 73 |
| NVMKEY (Nonvolatile Memory Key) | 74 |
| OCxCON (Output Compare x Control) | 201 |
| OSCCON (Oscillator Control) | 143 |
| OSCTUN (FRC Oscillator Tuning) | 147 |
| P1DC3 (PWM Duty Cycle 3) | 215 |
| PLLFB (PLL Feedback Divisor) | 146 |
| PMD1 (Peripheral Module Disable Control Register 1) | 153 |
| PMD2 (Peripheral Module Disable Control Register 2) | 155 |
| PMD3 (Peripheral Module Disable Control Register 3) | 156 |
| PWMxCON1 (PWM Control 1) | 209 |
| PWMxCON2 (PWM Control 2) | 210 |
| PxDC1 (PWM Duty Cycle 1) | 215 |
| PxDC2 (PWM Duty Cycle 2) | 215 |
| PxDTC1 (Dead-Time Control 1) | 211 |
| PxDTC2 (Dead-Time Control 2) | 212 |
| PxFLTA (Fault A Control) | 213 |
| PxOVDCON (Override Control) | 214 |
| PxSECMP (Special Event Compare) | 208 |
| PxTCON (PWM Time Base Control) | 206, 283, 284, 285 |
| PxTMR (PWM Timer Count Value) | 207 |
| PxTPER (PWM Time Base Period) | 207 |
| QEICON (QEI Control) | 218 |
| RCON (Reset Control) | 78 |
| SPIxCON1 (SPIx Control 1) | 223 |
| SPIxCON2 (SPIx Control 2) | 225 |
| SPIxSTAT (SPIx Status and Control) | 222 |
| SR (CPU Status) | 26, 90 |
| T1CON (Timer1 Control) | 190 |
| TCxCON (Input Capture x Control) | 198 |
| TxCON (Type B Time Base Control) | 194 |
| TyCON (Type C Time Base Control) | 195 |
| UxMODE (UARTx Mode) | 236 |
| UxSTA (UARTx Status and Control) | 238 |
| Reset | |
| Illegal Opcode | 77, 84 |
| Trap Conflict | 83, 84 |
| Uninitialized W Register | 77, 84 |
| Reset Sequence | 85 |
| Resets | 77 |
| S | |
| Serial Peripheral Interface (SPI) | 221 |
| Software Reset Instruction (SWR) | 83 |
| Software Simulator (MPLAB SIM) | 334 |
| Software Stack Pointer, Frame Pointer | |
| CALLL Stack Frame | 62 |
| Special Features of the CPU | 315 |
| SPI Module | |
| SPI1 Register Map | 49 |
| Symbols Used in Opcode Descriptions | 326 |
| System Control | |
| Register Map | 60, 61 |

T

| | |
|---|-----|
| Temperature and Voltage Specifications | |
| AC | 346 |
| Timer1 | 189 |
| Timer2/3 | 191 |
| Timing Characteristics | |
| CLKO and I/O | 349 |
| Timing Diagrams | |
| 10-bit ADC Conversion (CHPS = 01, SIMSAM = 0, ASAM = 0, SSR = 000) | 375 |
| 10-bit ADC Conversion (CHPS = 01, SIMSAM = 0, ASAM = 1, SSR = 111, SAMC = 00001) | 375 |
| 10-bit ADC Conversion (CHPS = 01, SIMSAM = 0, ASAM = 1, SSR = 111, SAMC = 00001) | 375 |
| 12-bit ADC Conversion (ASAM = 0, SSR = 000) | 373 |
| Brown-out Situations | 83 |
| ECAN I/O | 369 |
| External Clock | 347 |
| I2Cx Bus Data (Master Mode) | 365 |
| I2Cx Bus Data (Slave Mode) | 367 |
| I2Cx Bus Start/Stop Bits (Master Mode) | 365 |
| I2Cx Bus Start/Stop Bits (Slave Mode) | 367 |
| Input Capture (CAPx) | 355 |
| Motor Control PWM | 357 |
| Motor Control PWM Fault | 357 |
| OC/PWM | 356 |
| Output Compare (OCx) | 355 |
| QEA/QEB Input | 358 |
| QEI Module Index Pulse | 359 |
| Reset, Watchdog Timer, Oscillator Start-up Timer and Power-up Timer | 350 |
| SPIx Master Mode (CKE = 0) | 360 |
| SPIx Master Mode (CKE = 1) | 361 |
| SPIx Slave Mode (CKE = 0) | 362 |
| SPIx Slave Mode (CKE = 1) | 363 |
| Timer1, 2, 3 External Clock | 352 |
| TimerQ (QEI Module) External Clock | 354 |
| Timing Requirements | |
| CLKO and I/O | 349 |
| External Clock | 347 |
| Input Capture | 355 |
| Timing Specifications | |
| 10-bit ADC Conversion Requirements | 376 |
| 12-bit ADC Conversion Requirements | 374 |
| CAN I/O Requirements | 369 |
| I2Cx Bus Data Requirements (Master Mode) | 366 |
| I2Cx Bus Data Requirements (Slave Mode) | 368 |
| Motor Control PWM Requirements | 357 |
| Output Compare Requirements | 355 |
| PLL Clock | 348 |
| QEI External Clock Requirements | 354 |
| QEI Index Pulse Requirements | 359 |
| Quadrature Decoder Requirements | 358 |
| Reset, Watchdog Timer, Oscillator Start-up Timer, Power-up Timer and Brown-out Reset Requirements | 351 |
| Simple OC/PWM Mode Requirements | 356 |
| SPIx Master Mode (CKE = 0) Requirements | 360 |
| SPIx Master Mode (CKE = 1) Requirements | 361 |
| SPIx Slave Mode (CKE = 0) Requirements | 362 |
| SPIx Slave Mode (CKE = 1) Requirements | 364 |
| Timer1 External Clock Requirements | 352 |
| Timer2 External Clock Requirements | 353 |
| Timer3 External Clock Requirements | 353 |

U

UART Module
 UART1 Register Map 48
Universal Asynchronous Receiver Transmitter (UART).... 235
Using the RCON Status Bits 84

V

Voltage Regulator (On-Chip) 319

W

Watchdog Time-out Reset (WDTR) 83
Watchdog Timer (WDT) 315, 320
 Programming Considerations 320
WWW Address 402
WWW, On-Line Support 11

NOTES:

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website 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 receives 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 website at www.microchip.com, click on Customer Change Notification and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

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 website at: <http://support.microchip.com>

READER RESPONSE

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

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

To: Technical Publications Manager Total Pages Sent _____
RE: Reader Response
From: Name _____
Company _____
Address _____
City / State / ZIP / Country _____
Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and Literature Number: DS70291C
dsPIC33FJ128MCX02/X04

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|---------------|---|---|----------------------------------|----------------------|----|---|----------------------------|----------------|-----|---|----------------------|--|-----|---|----------------------|--|-----|---|----------------------|------------|----|---|--------|--|----|---|--------|--------------------|---|---|----------------------------|--|---|---|---------------------------|----------|----|---|--|--|----|---|--|--|----|---|---|--|----|---|---|--|----|---|---|
| <p style="text-align: center;">dsPIC 33 FJ 32 MC3 02 T E / SP - XXX</p> <p>Microchip Trademark _____</p> <p>Architecture _____</p> <p>Flash Memory Family _____</p> <p>Program Memory Size (KB) _____</p> <p>Product Group _____</p> <p>Pin Count _____</p> <p>Tape and Reel Flag (if applicable) _____</p> <p>Temperature Range _____</p> <p>Package _____</p> <p>Pattern _____</p> | <p>Examples:</p> <p>a) dsPIC33FJ32MC302-E/SP: Motor Control dsPIC33, 32 KB program memory, 28-pin, Extended temperature, SPDIP package.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Architecture:</td> <td style="width: 10%;">33</td> <td style="width: 10%;">=</td> <td>16-bit Digital Signal Controller</td> </tr> <tr> <td>Flash Memory Family:</td> <td>FJ</td> <td>=</td> <td>Flash program memory, 3.3V</td> </tr> <tr> <td>Product Group:</td> <td>MC2</td> <td>=</td> <td>Motor Control family</td> </tr> <tr> <td></td> <td>MC3</td> <td>=</td> <td>Motor Control family</td> </tr> <tr> <td></td> <td>MC8</td> <td>=</td> <td>Motor Control family</td> </tr> <tr> <td>Pin Count:</td> <td>02</td> <td>=</td> <td>28-pin</td> </tr> <tr> <td></td> <td>04</td> <td>=</td> <td>44-pin</td> </tr> <tr> <td>Temperature Range:</td> <td>I</td> <td>=</td> <td>-40°C to+85°C (Industrial)</td> </tr> <tr> <td></td> <td>E</td> <td>=</td> <td>-40°C to+125°C (Extended)</td> </tr> <tr> <td>Package:</td> <td>SP</td> <td>=</td> <td>Skinny Plastic Dual In-Line - 300 mil body (SPDIP)</td> </tr> <tr> <td></td> <td>SO</td> <td>=</td> <td>Plastic Small Outline - Wide - 300 mil body (SOIC)</td> </tr> <tr> <td></td> <td>ML</td> <td>=</td> <td>Plastic Quad, No Lead Package - 8x8 mm body (QFN)</td> </tr> <tr> <td></td> <td>MM</td> <td>=</td> <td>Plastic Quad, No Lead Package - 6x6x0.9 mm body (QFN-S)</td> </tr> <tr> <td></td> <td>PT</td> <td>=</td> <td>Plastic Thin Quad Flatpack - 10x10x1 mm body (TQFP)</td> </tr> </table> | | Architecture: | 33 | = | 16-bit Digital Signal Controller | Flash Memory Family: | FJ | = | Flash program memory, 3.3V | Product Group: | MC2 | = | Motor Control family | | MC3 | = | Motor Control family | | MC8 | = | Motor Control family | Pin Count: | 02 | = | 28-pin | | 04 | = | 44-pin | Temperature Range: | I | = | -40°C to+85°C (Industrial) | | E | = | -40°C to+125°C (Extended) | Package: | SP | = | Skinny Plastic Dual In-Line - 300 mil body (SPDIP) | | SO | = | Plastic Small Outline - Wide - 300 mil body (SOIC) | | ML | = | Plastic Quad, No Lead Package - 8x8 mm body (QFN) | | MM | = | Plastic Quad, No Lead Package - 6x6x0.9 mm body (QFN-S) | | PT | = | Plastic Thin Quad Flatpack - 10x10x1 mm body (TQFP) |
| Architecture: | 33 | = | 16-bit Digital Signal Controller | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Flash Memory Family: | FJ | = | Flash program memory, 3.3V | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Product Group: | MC2 | = | Motor Control family | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | MC3 | = | Motor Control family | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | MC8 | = | Motor Control family | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Pin Count: | 02 | = | 28-pin | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 04 | = | 44-pin | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Temperature Range: | I | = | -40°C to+85°C (Industrial) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | E | = | -40°C to+125°C (Extended) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Package: | SP | = | Skinny Plastic Dual In-Line - 300 mil body (SPDIP) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | SO | = | Plastic Small Outline - Wide - 300 mil body (SOIC) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | ML | = | Plastic Quad, No Lead Package - 8x8 mm body (QFN) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | MM | = | Plastic Quad, No Lead Package - 6x6x0.9 mm body (QFN-S) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | PT | = | Plastic Thin Quad Flatpack - 10x10x1 mm body (TQFP) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |



Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
<http://support.microchip.com>
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - 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 - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4080

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: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

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

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

EUROPE

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

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

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

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

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

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

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

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

02/04/09