dsPIC33FJXXXMCX06/X08/X10 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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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, rfPIC, 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, nanoWatt XLP, PICkit, PICDEM, PICDEM.net, PICtail, PIC^{32} logo, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Total Endurance, TSHARC, 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.

Downloaded from Arrow.com.

dsPIC33FJXXXMCX06/X08/X10

High-Performance, 16-Bit Digital Signal Controllers

Operating Range:

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

Interrupt Controller:

- 5-cycle latency
- Up to 67 available interrupt sources
- Up to five external interrupts
- Seven programmable priority levels
- Five processor exceptions

Digital I/O:

- Up to 85 programmable digital I/O pins
- Wake-up/Interrupt-on-Change on up to 24 pins
- Output pins can drive from 3.0V to 3.6V
- 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 256 Kbytes
- Data SRAM, up to 30 Kbytes (includes 2 Kbytes of DMA RAM)

System Management:

- Flexible clock options:
  - External, crystal, resonator, internal RC
  - Fully integrated 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

Timers/Capture/Compare/PWM:

- Timer/Counters, up to nine 16-bit timers:
  - Can pair up to make four 32-bit timers
  - 1 timer runs as Real-Time Clock with external 32.768 kHz oscillator
  - Programmable prescaler
- Input Capture (up to eight channels):
  - Capture on up, down or both edges
  - 16-bit capture input functions
  - 4-deep FIFO on each capture
- Output Compare (up to eight channels):
  - Single or Dual 16-Bit Compare mode
  - 16-bit Glitchless PWM mode

Communication Modules:

- 3-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
- I^2C™ (up to two modules):
  - 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 2 modules):
  - 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

Motor Control Peripherals:

- Motor Control PWM (up to eight channels):
  - Four duty cycle generators
  - Independent or Complementary mode
  - Programmable dead time and output polarity
  - Edge or center-aligned
  - Manual output override control
  - Up to two Fault inputs
  - 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
- 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

Analog-to-Digital Converters (ADCs):

- Up to two ADC modules in a device
- 10-bit, 1.1 Msps or 12-bit, 500 ksps conversion:
  - Two, four or eight simultaneous samples
  - Up to 32 input channels with auto-scanning
  - Conversion start can be manual or synchronized with one of four trigger sources
  - Conversion possible in Sleep mode
  - ±1 LSb max integral nonlinearity
  - ±1 LSb max differential nonlinearity

CMOS Flash Technology:

- Low-power, high-speed Flash technology
- Fully static design
- 3.3V (±10%) operating voltage
- Industrial temperature
- Low-power consumption

Packaging:

- 100-pin TQFP (14x14x1 mm and 12x12x1 mm)
- 80-pin TQFP (12x12x1 mm)
- 64-pin TQFP (10x10x1 mm)

Note: See the device variant tables for exact peripheral features per device.

dsPIC33F PRODUCT FAMILIES

The dsPIC33FJXXXMCX06/X08/X10 family of devices supports a variety of motor control applications, such as brushless DC motors, single and 3-phase induction motors and switched reluctance motors. The dsPIC33F Motor Control products are also well-suited for Uninterrupted Power Supply (UPS), inverters, switched mode power supplies, power factor correction and also for controlling the power management module in servers, telecommunication equipment and other industrial equipment.

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

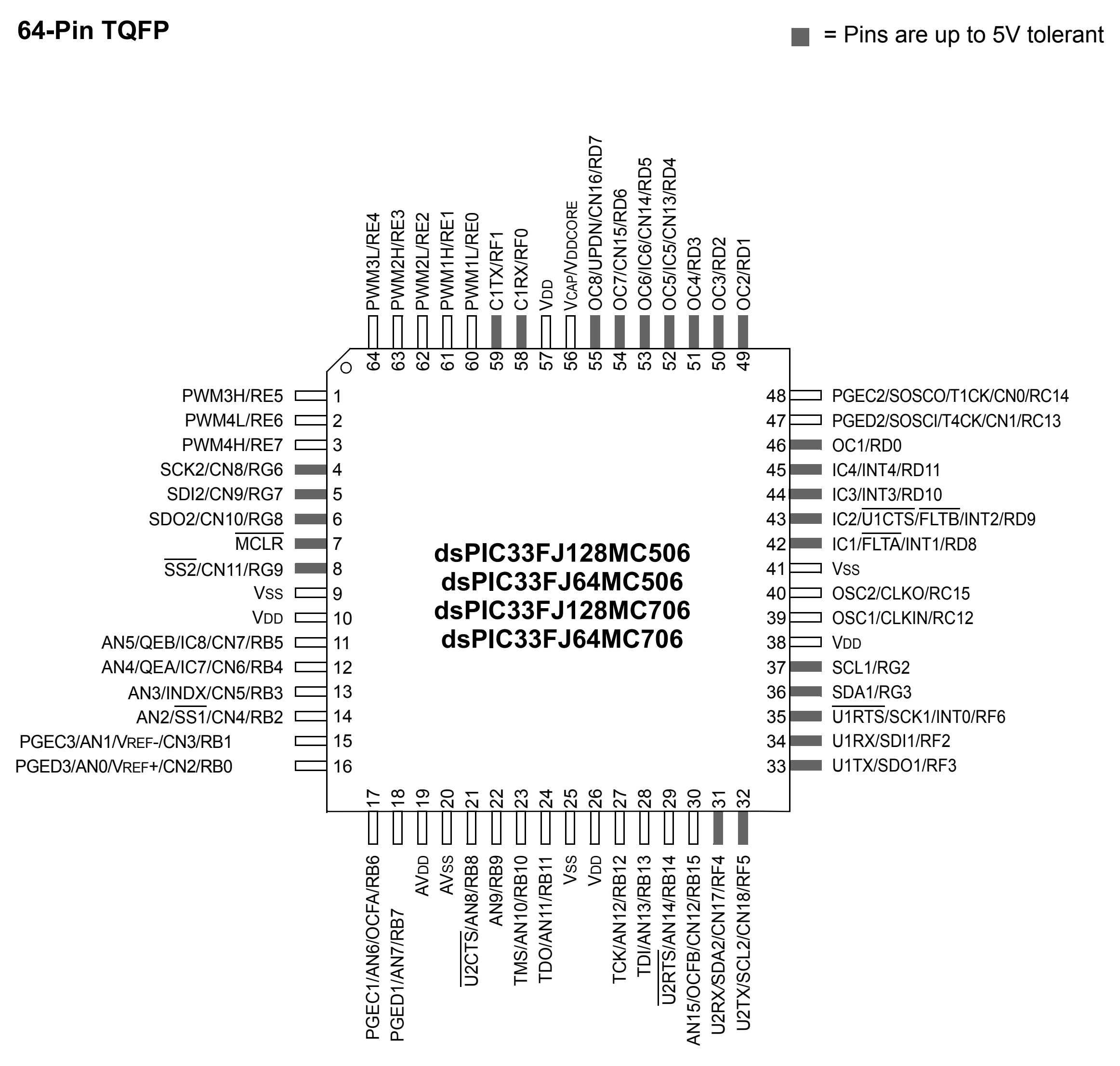
dsPIC33FJXXXMCX06/X08/X10 Controller Families

Device	Pins	Program Flash Memory (Kbyte)	RAM (Kbyte)(1)	Timer 16-bit	Input Capture	Output Compare Std. PWM	Motor Control PWM	Quadrature Encoder Interface	Codec Interface	ADC	UART	SPI	I^{2}C™	Enhanced CAN™	I/O Pins (Max)(2)	Packages
dsPIC33FJ64MC506	64	64	8	9	8	8	8 ch	1	0	1 ADC, 16 ch	2	2	2	1	53	PT
dsPIC33FJ64MC508	80	64	8	9	8	8	8 ch	1	0	1 ADC, 18 ch	2	2	2	1	69	PT
dsPIC33FJ64MC510	100	64	8	9	8	8	8 ch	1	0	1 ADC, 24 ch	2	2	2	1	85	PF, PT
dsPIC33FJ64MC706	64	64	16	9	8	8	8 ch	1	0	2 ADC, 16 ch	2	2	2	1	53	PT
dsPIC33FJ64MC710	100	64	16	9	8	8	8 ch	1	0	2 ADC, 24 ch	2	2	2	2	85	PF, PT
dsPIC33FJ128MC506	64	128	8	9	8	8	8 ch	1	0	1 ADC, 16 ch	2	2	2	1	53	PT
dsPIC33FJ128MC510	100	128	8	9	8	8	8 ch	1	0	1 ADC, 24 ch	2	2	2	1	85	PF, PT
dsPIC33FJ128MC706	64	128	16	9	8	8	8 ch	1	0	2 ADC, 16 ch	2	2	2	1	53	PT
dsPIC33FJ128MC708	80	128	16	9	8	8	8 ch	1	0	2 ADC, 18 ch	2	2	2	2	69	PT
dsPIC33FJ128MC710	100	128	16	9	8	8	8 ch	1	0	2 ADC, 24 ch	2	2	2	2	85	PF, PT
dsPIC33FJ256MC510	100	256	16	9	8	8	8 ch	1	0	1 ADC, 24 ch	2	2	2	1	85	PF, PT
dsPIC33FJ256MC710	100	256	30	9	8	8	8 ch	1	0	2 ADC, 24 ch	2	2	2	2	85	PF, PT

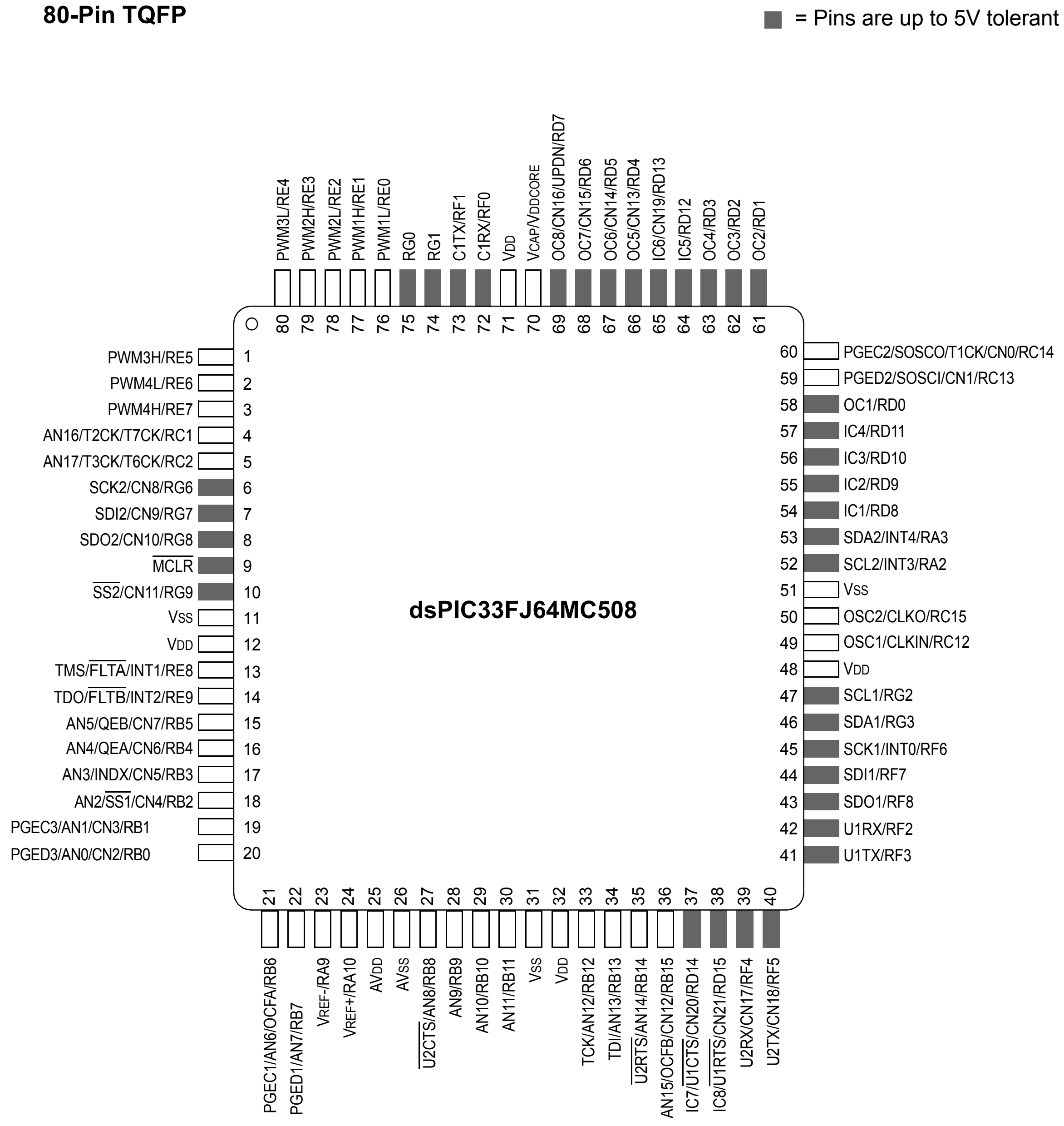
Note 1: RAM size is inclusive of 2 Kbytes DMA RAM.

2: Maximum I/O pin count includes pins shared by the peripheral functions.

Pin Diagrams



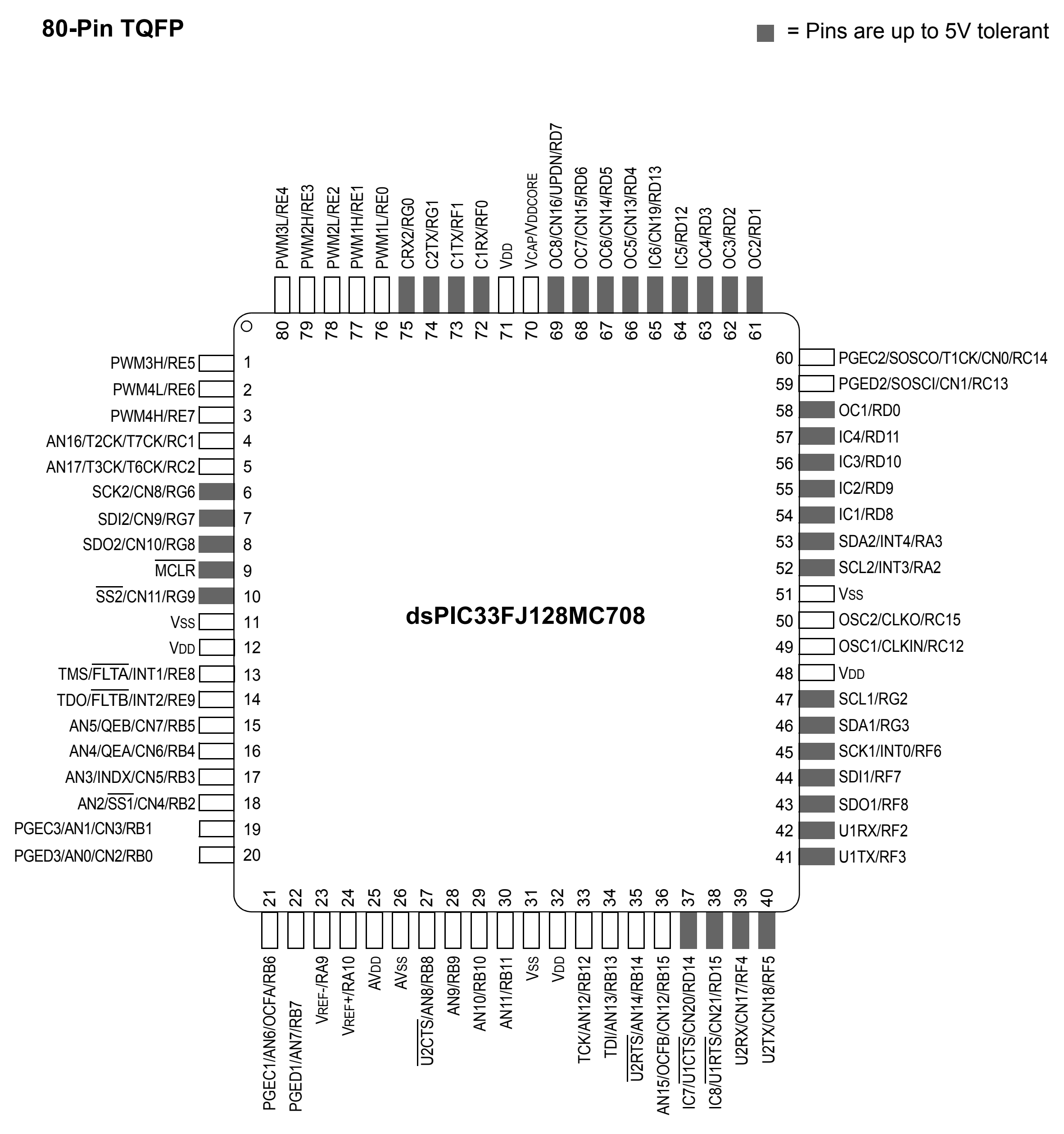
Pin Diagrams (Continued)



80-Pin TQFP

■ = Pins are up to 5V tolerant

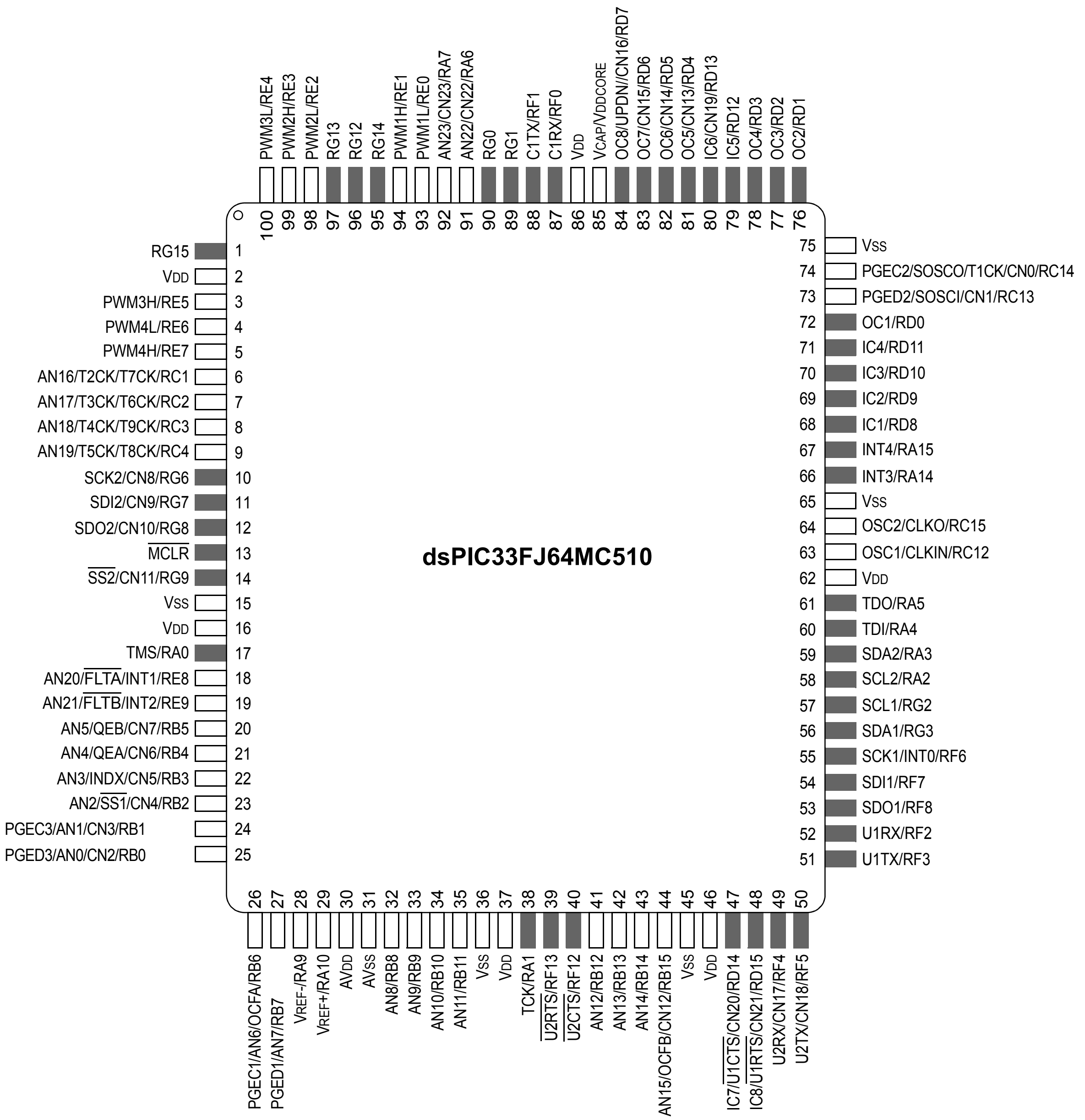
Pin Diagrams (Continued)



Pin Diagrams (Continued)

100-Pin TQFP

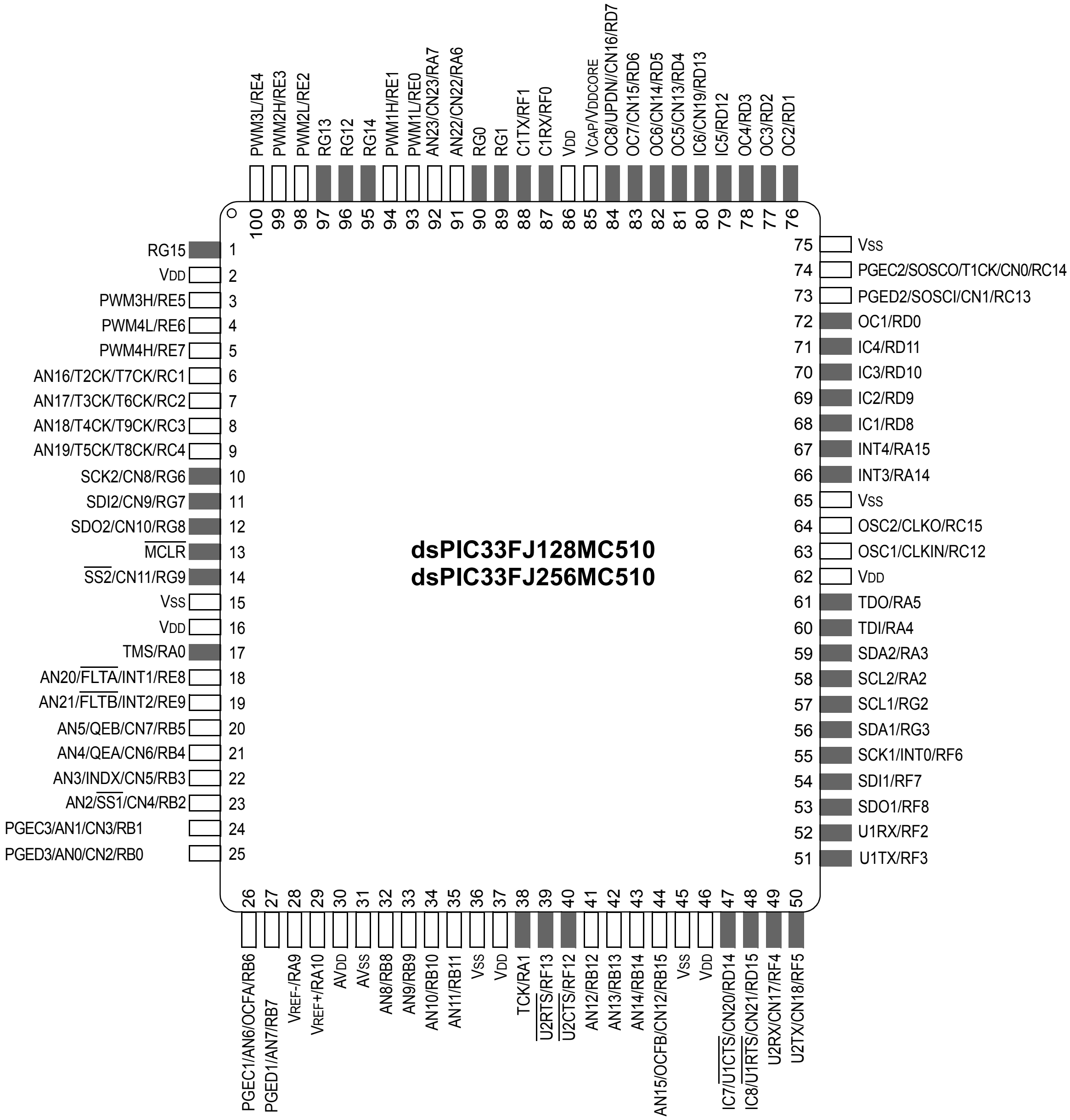
■ = Pins are up to 5V tolerant



Pin Diagrams (Continued)

100-Pin TQFP

▨ = Pins are up to 5V tolerant



Pin Diagrams (Continued)

100-Pin TQFP

▇ = Pins are up to 5V tolerant

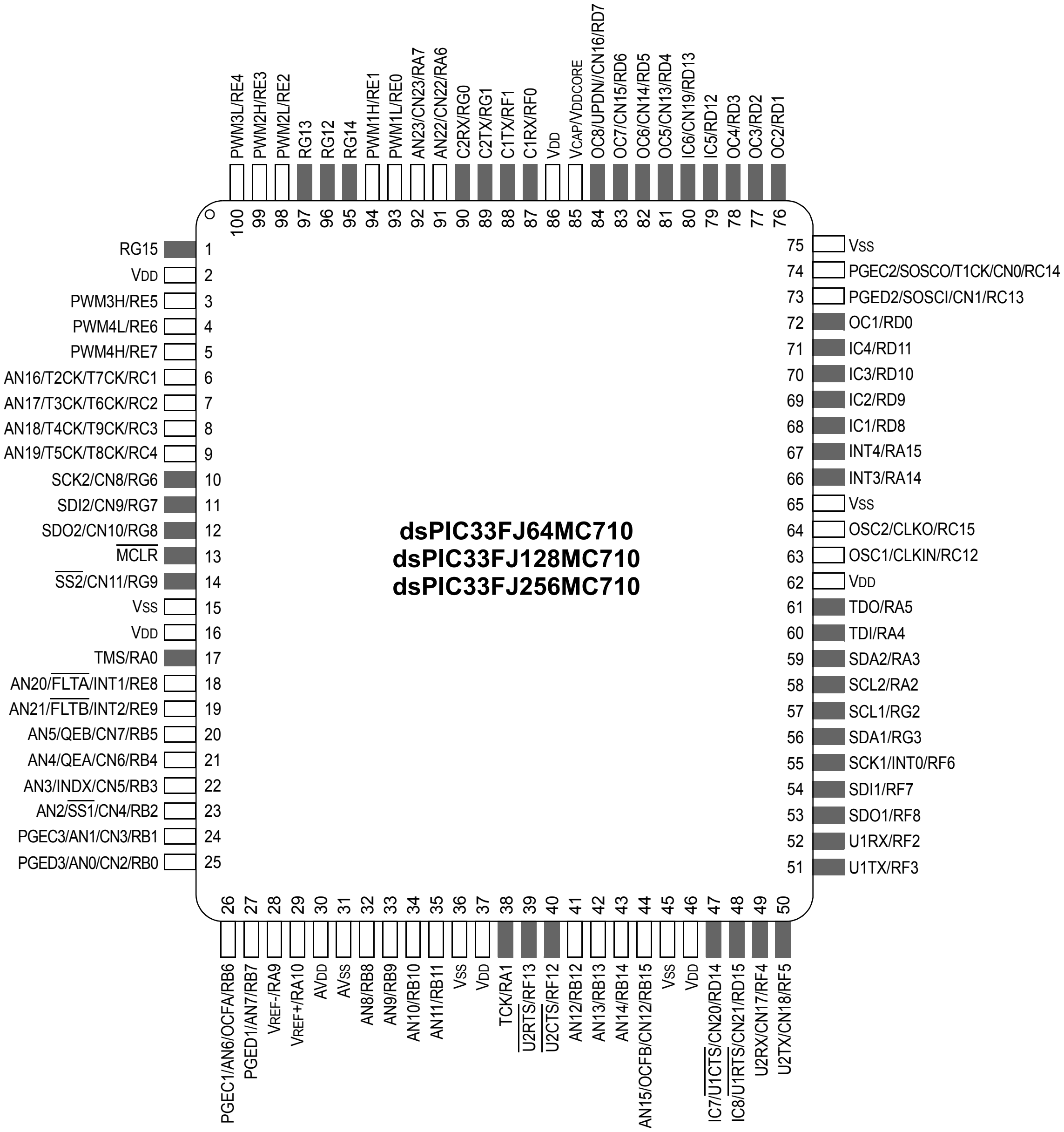


Table of Contents

TO OUR VALUED CUSTOMERS

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

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

Most Current Data Sheet

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

http://www.microchip.com

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

Errata

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

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

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

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

Customer Notification System

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

NOTES:

1.0 DEVICE OVERVIEW

Note: This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the latest family reference sections of the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

This document contains device specific information for the following devices:

- dsPIC33FJ64MC506
- dsPIC33FJ64MC508
- dsPIC33FJ64MC510
- dsPIC33FJ64MC706
- dsPIC33FJ64MC710
- dsPIC33FJ128MC506
- dsPIC33FJ128MC510
- dsPIC33FJ128MC706
- dsPIC33FJ128MC708
- dsPIC33FJ128MC710
- dsPIC33FJ256MC510
- dsPIC33FJ256MC710

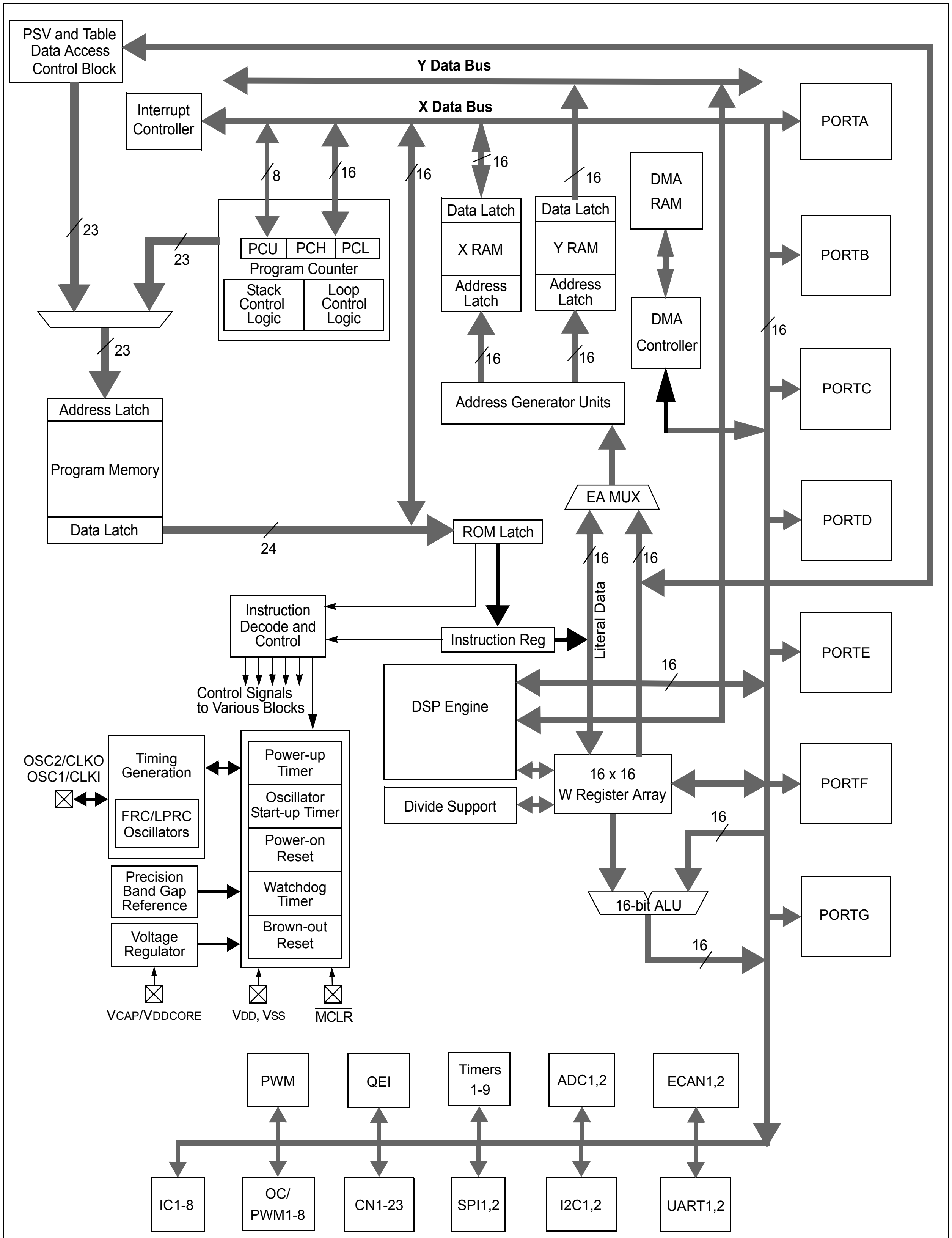
The dsPIC33FJXXXMCX06/X08/X10 includes devices with a wide range of pin counts (64, 80 and 100), different program memory sizes (64 Kbytes, 128 Kbytes and 256 Kbytes) and different RAM sizes (8 Kbytes, 16 Kbytes and 30 Kbytes).

These features make this family suitable for a wide variety of high-performance digital signal control applications. The devices are pin compatible with the PIC24H family of devices, and also share a very high degree of compatibility with the dsPIC30F family devices. This allows easy migration between device families as may be necessitated by the specific functionality, computational resource and system cost requirements of the application.

The dsPIC33FJXXXMCX06/X08/X10 family of devices employ a powerful 16-bit architecture that seamlessly integrates the control features of a Microcontroller (MCU) with the computational capabilities of a Digital Signal Processor (DSP). The resulting functionality is ideal for applications that rely on high-speed, repetitive computations, as well as control.

The DSP engine, dual 40-bit accumulators, hardware support for division operations, barrel shifter, 17 x 17 multiplier, a large array of 16-bit working registers and a wide variety of data addressing modes, together, provide the dsPIC33FJXXXMCX06/X08/X10 Central Processing Unit (CPU) with extensive mathematical processing capability. Flexible and deterministic interrupt handling, coupled with a powerful array of peripherals, renders the dsPIC33FJXXXMCX06/X08/X10 devices suitable for control applications. Further, Direct Memory Access (DMA) enables overhead-free transfer of data between several peripherals and a dedicated DMA RAM. Reliable, field programmable Flash program memory ensures scalability of applications that use dsPIC33FJXXXMCX06/X08/X10 devices.

FIGURE 1-1: dsPIC33FJXXXMCX06/X08/X10 GENERAL BLOCK DIAGRAM



Note: Not all pins or features are implemented on all device pinout configurations. See pinout diagrams for the specific pins and features present on each device.

TABLE 1-1: PINOUT I/O DESCRIPTIONS

Pin Name	Pin Type	Buffer Type	Description
AN0-AN31	I	Analog	Analog input channels.
AVDD	P	P	Positive supply for analog modules. This pin must be connected at all times.
AVSS	P	P	Ground reference for analog modules.
CLKI	I	ST/CMOS	External clock source input. Always associated with OSC1 pin function.
CLKO	O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes. Always associated with OSC2 pin function.
CN0-CN23	I	ST	Input change notification inputs. Can be software programmed for internal weak pull-ups on all inputs.
C1RX	I	ST	ECAN1 bus receive pin.
C1TX	O	—	ECAN1 bus transmit pin.
C2RX	I	ST	ECAN2 bus receive pin.
C2TX	O	—	ECAN2 bus transmit pin.
PGED1	I/O	ST	Data I/O pin for programming/debugging communication channel 1.
PGEC1	I	ST	Clock input pin for programming/debugging communication channel 1.
PGED2	I/O	ST	Data I/O pin for programming/debugging communication channel 2.
PGEC2	I	ST	Clock input pin for programming/debugging communication channel 2.
PGED3	I/O	ST	Data I/O pin for programming/debugging communication channel 3.
PGEC3	I	ST	Clock input pin for programming/debugging communication channel 3.
IC1-IC8	I	ST	Capture inputs 1 through 8.
INDX	I	ST	Quadrature Encoder Index Pulse input.
QEA	I	ST	Quadrature Encoder Phase A input in QEI mode. Auxiliary Timer External Clock/Gate input in Timer mode.
QEB	I	ST	Quadrature Encoder Phase A input in QEI mode. Auxiliary Timer External Clock/Gate input in Timer mode.
UPDN	O	CMOS	Position Up/Down Counter Direction State.
INT0	I	ST	External interrupt 0.
INT1	I	ST	External interrupt 1.
INT2	I	ST	External interrupt 2.
INT3	I	ST	External interrupt 3.
INT4	I	ST	External interrupt 4.
$\overline{\text{FLTA}}$	I	ST	PWM Fault A input.
$\overline{\text{FLTB}}$	I	ST	PWM Fault B input.
PWM1L	O	—	PWM 1 low output.
PWM1H	O	—	PWM 1 high output.
PWM2L	O	—	PWM 2 low output.
PWM2H	O	—	PWM 2 high output.
PWM3L	O	—	PWM 3 low output.
PWM3H	O	—	PWM 3 high output.
PWM4L	O	—	PWM 4 low output.
PWM4H	O	—	PWM 4 high output.
$\overline{\text{MCLR}}$	I/P	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
OCFA	I	ST	Compare Fault A input (for Compare Channels 1, 2, 3 and 4).
OCFB	I	ST	Compare Fault B input (for Compare Channels 5, 6, 7 and 8).
OC1-OC8	O	—	Compare outputs 1 through 8.
OSC1	I	ST/CMOS	Oscillator crystal input. ST buffer when configured in RC mode; CMOS otherwise.
OSC2	I/O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes.

Legend: CMOS = CMOS compatible input or output   Analog = Analog input   P = Power
ST = Schmitt Trigger input with CMOS levels   O = Output   I = Input

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

Pin Name	Pin Type	Buffer Type	Description
RA0-RA7	I/O	ST	PORTA is a bidirectional I/O port.
RA9-RA10	I/O	ST	
RA12-RA15	I/O	ST	
RB0-RB15	I/O	ST	PORTB is a bidirectional I/O port.
RC1-RC4	I/O	ST	PORTC is a bidirectional I/O port.
RC12-RC15	I/O	ST	
RD0-RD15	I/O	ST	PORTD is a bidirectional I/O port.
RE0-RE9	I/O	ST	PORTE is a bidirectional I/O port.
RF0-RF8	I/O	ST	PORTF is a bidirectional I/O port.
RF12-RF13			
RG0-RG3	I/O	ST	PORTG is a bidirectional I/O port.
RG6-RG9	I/O	ST	
RG12-RG15	I/O	ST	
SCK1	I/O	ST	Synchronous serial clock input/output for SPI1.
SDI1	I	ST	SPI1 data in.
SDO1	O	—	SPI1 data out.
$\overline{\text{SS1}}$	I/O	ST	SPI1 slave synchronization or frame pulse I/O.
SCK2	I/O	ST	Synchronous serial clock input/output for SPI2.
SDI2	I	ST	SPI2 data in.
SDO2	O	—	SPI2 data out.
$\overline{\text{SS2}}$	I/O	ST	SPI2 slave synchronization or frame pulse I/O.
SCL1	I/O	ST	Synchronous serial clock input/output for I2C1.
SDA1	I/O	ST	Synchronous serial data input/output for I2C1.
SCL2	I/O	ST	Synchronous serial clock input/output for I2C2.
SDA2	I/O	ST	Synchronous serial data input/output for I2C2.
SOSCI	I	ST/CMOS	32.768 kHz low-power oscillator crystal input; CMOS otherwise.
SOSCO	O	—	32.768 kHz low-power oscillator crystal output.
TMS	I	ST	JTAG Test mode select pin.
TCK	I	ST	JTAG test clock input pin.
TDI	I	ST	JTAG test data input pin.
TDO	O	—	JTAG test data output pin.
T1CK	I	ST	Timer1 external clock input.
T2CK	I	ST	Timer2 external clock input.
T3CK	I	ST	Timer3 external clock input.
T4CK	I	ST	Timer4 external clock input.
T5CK	I	ST	Timer5 external clock input.
T6CK	I	ST	Timer6 external clock input.
T7CK	I	ST	Timer7 external clock input.
T8CK	I	ST	Timer8 external clock input.
T9CK	I	ST	Timer9 external clock input.
$\overline{\text{U1CTS}}$	I	ST	UART1 clear to send.
$\overline{\text{U1RTS}}$	O	—	UART1 ready to send.
U1RX	I	ST	UART1 receive.
U1TX	O	—	UART1 transmit.
$\overline{\text{U2CTS}}$	I	ST	UART2 clear to send.
$\overline{\text{U2RTS}}$	O	—	UART2 ready to send.
U2RX	I	ST	UART2 receive.
U2TX	O	—	UART2 transmit.
VDD	P	—	Positive supply for peripheral logic and I/O pins.
VCAP/VDDCORE	P	—	CPU logic filter capacitor connection.

Legend: CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels
Analog = Analog input
O = Output
P = Power
I = Input

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

Pin Name	Pin Type	Buffer Type	Description
VSS	P	—	Ground reference for logic and I/O pins.
VREF+	I	Analog	Analog voltage reference (high) input.
VREF-	I	Analog	Analog voltage reference (low) input.

Legend:
CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels
Analog = Analog input
O = Output
P = Power
I = Input

NOTES:

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

Note: This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 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 dsPIC33FJXXXMCX06/X08/X10 family of 16-bit Digital Signal Controllers (DSCs) 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)”**)
- $\overline{\text{MCLR}}$ pin
  (see **Section 2.4 “Master Clear ($\overline{\text{MCLR}}$) 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.

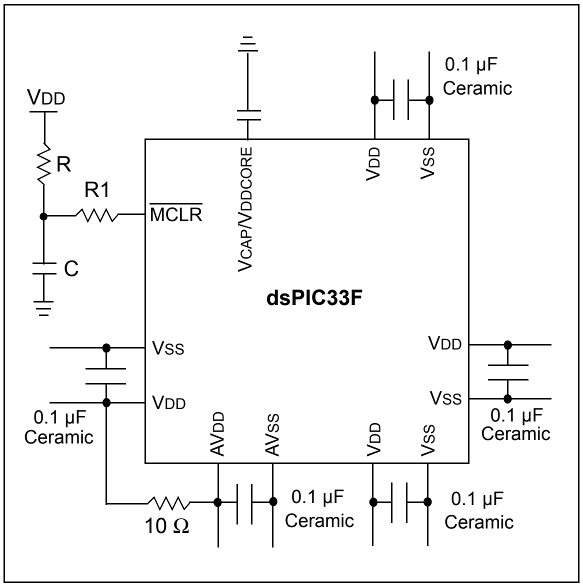
2.2 Decoupling Capacitors

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 26.0 “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 23.2 “On-Chip Voltage Regulator”** for details.

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

The $\overline{\text{MCLR}}$ pin provides for two specific device functions:

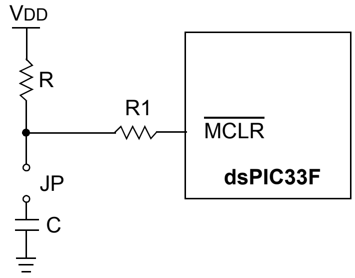
- Device Reset
- Device programming and debugging

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 $\overline{\text{MCLR}}$ 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 $\overline{\text{MCLR}}$ pin during programming and debugging operations.

Place the components shown in Figure 2-2 within one-quarter inch (6 mm) from the $\overline{\text{MCLR}}$ pin.

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



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

2: R1 ≤ 470Ω will limit any current flowing into $\overline{\text{MCLR}}$ from the external capacitor C, in the event of $\overline{\text{MCLR}}$ pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS). Ensure that the $\overline{\text{MCLR}}$ pin VIH and VIL specifications are met.

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.

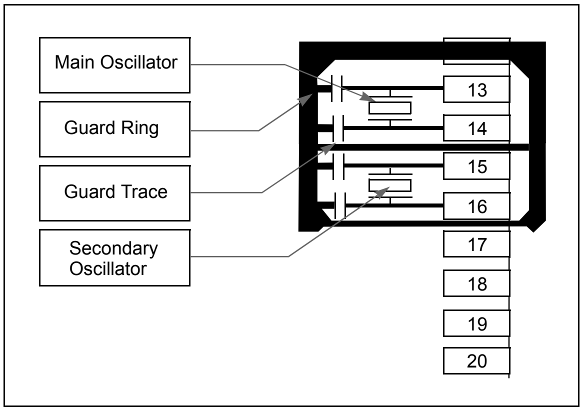
- *“MPLAB® ICD 2 In-Circuit Debugger User’s Guide”* DS51331
- *“Using MPLAB® ICD 2”* (poster) DS51265
- *“MPLAB® ICD 2 Design Advisory”* DS51566
- *“Using MPLAB® ICD 3 In-Circuit Debugger”* (poster) DS51765
- *“MPLAB® ICD 3 Design Advisory”* DS51764
- *“MPLAB® REAL ICE™ In-Circuit Emulator User’s Guide”* 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 MHz < F_{IN} < 8 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 the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 2. "CPU"** (DS70204) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

The dsPIC33FJXXXMCX06/X08/X10 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 point.

The dsPIC33FJXXXMCX06/X08/X10 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.

The dsPIC33FJXXXMCX06/X08/X10 instruction set has two classes of instructions: 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 dsPIC33FJXXXMCX06/X08/X10 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 dsPIC33FJXXXMCX06/X08/X10 is shown in Figure 3-2.

3.1 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.

The data space also includes 2 Kbytes of DMA RAM, which is primarily used for DMA data transfers but may be used as general purpose RAM.

3.2 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 memory 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.

3.3 Special MCU Features

The dsPIC33FJXXXMCX06/X08/X10 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) x (-1.0).

The dsPIC33FJXXXMCX06/X08/X10 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 a 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: dsPIC33FJXXXMCX06/X08/X10 CPU CORE BLOCK DIAGRAM

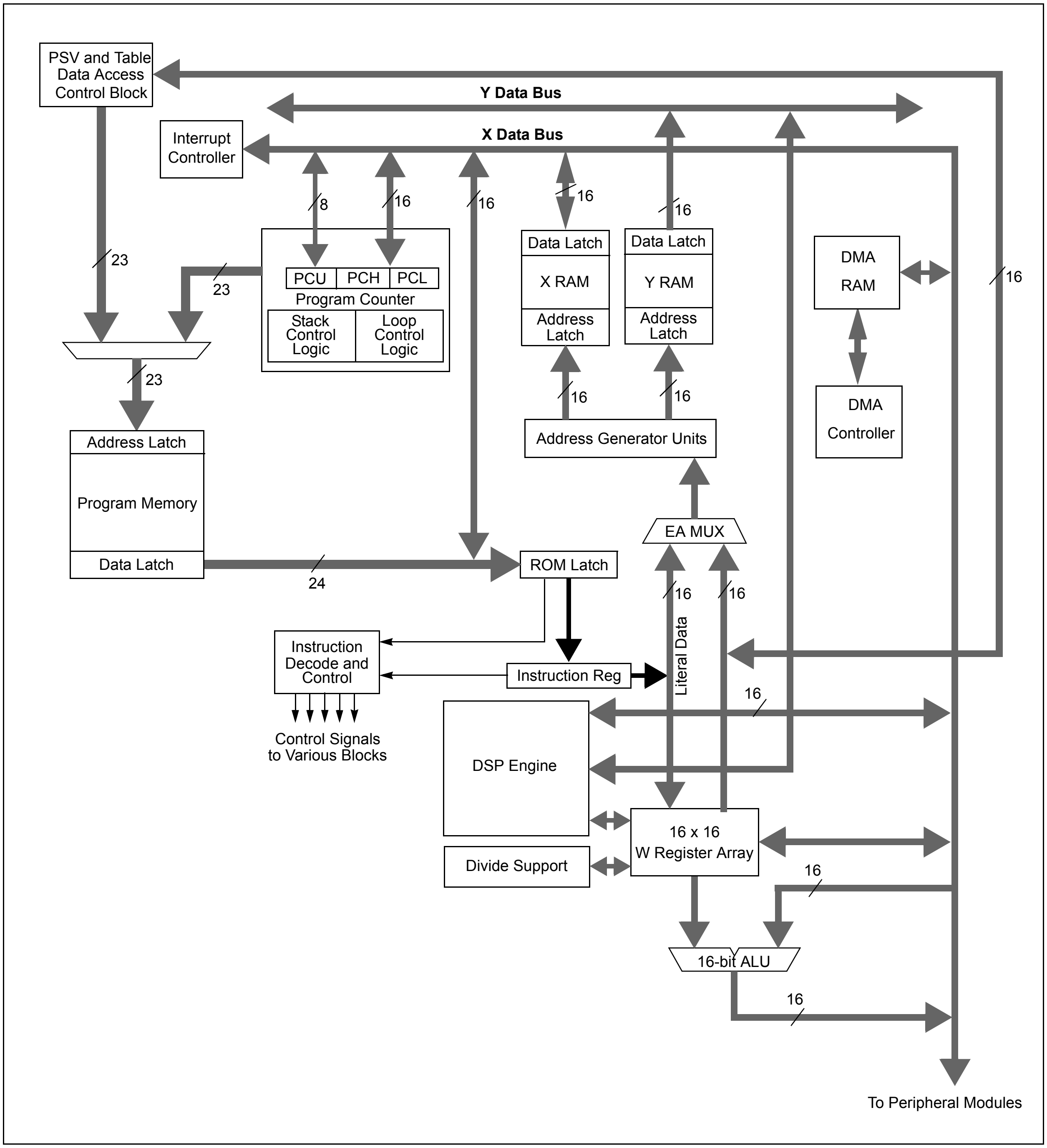
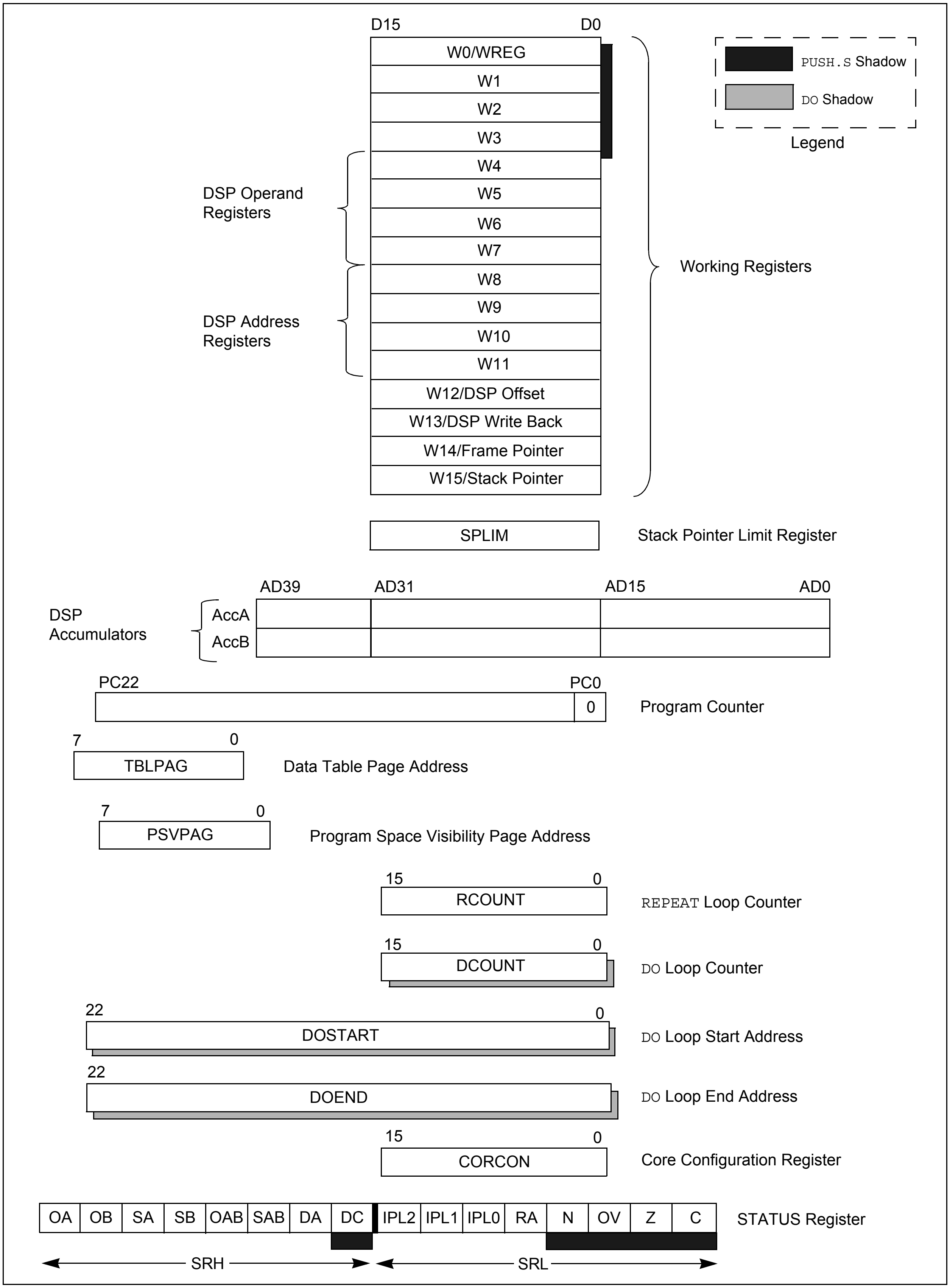


FIGURE 3-2: dsPIC33FJXXXMCX06/X08/X10 PROGRAMMER'S MODEL



3.4 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(2)	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

`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

Note: This bit may be read or cleared (not set). Clearing this bit will clear SA and SB.

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/$\overline{\text{Borrow}}$ 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 may 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>).

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 (2’s complement). It indicates an overflow of the 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 which affects the Z bit has set it at some time in the past
`0` = The most recent operation which affects the Z bit has cleared it (i.e., a non-zero result)

bit 0 **C:** MCU ALU Carry/$\overline{\text{Borrow}}$ 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 may 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>).

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
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 will 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.

3.5 Arithmetic Logic Unit (ALU)

The dsPIC33FJXXXMCX06/X08/X10 ALU is 16 bits wide and is capable of addition, subtraction, bit shifts and logic operations. Unless otherwise mentioned, arithmetic operations are 2's complement in nature. Depending on the operation, the ALU may affect the values of the Carry (C), Zero (Z), Negative (N), Overflow (OV) and Digit Carry (DC) Status bits in the SR register. The C and DC Status bits operate as Borrow and Digit Borrow bits, respectively, for subtraction operations.

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

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

The dsPIC33FJXXXMCX06/X08/X10 CPU incorporates hardware support for both multiplication and division. This includes a dedicated hardware multiplier and support hardware for 16-bit-divisor division.

3.5.1 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:

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

3.5.2 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.

3.6 DSP Engine

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

The dsPIC33FJXXXMCX06/X08/X10 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 may be used concurrently by the same instruction (e.g., ED, EDAC).

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

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

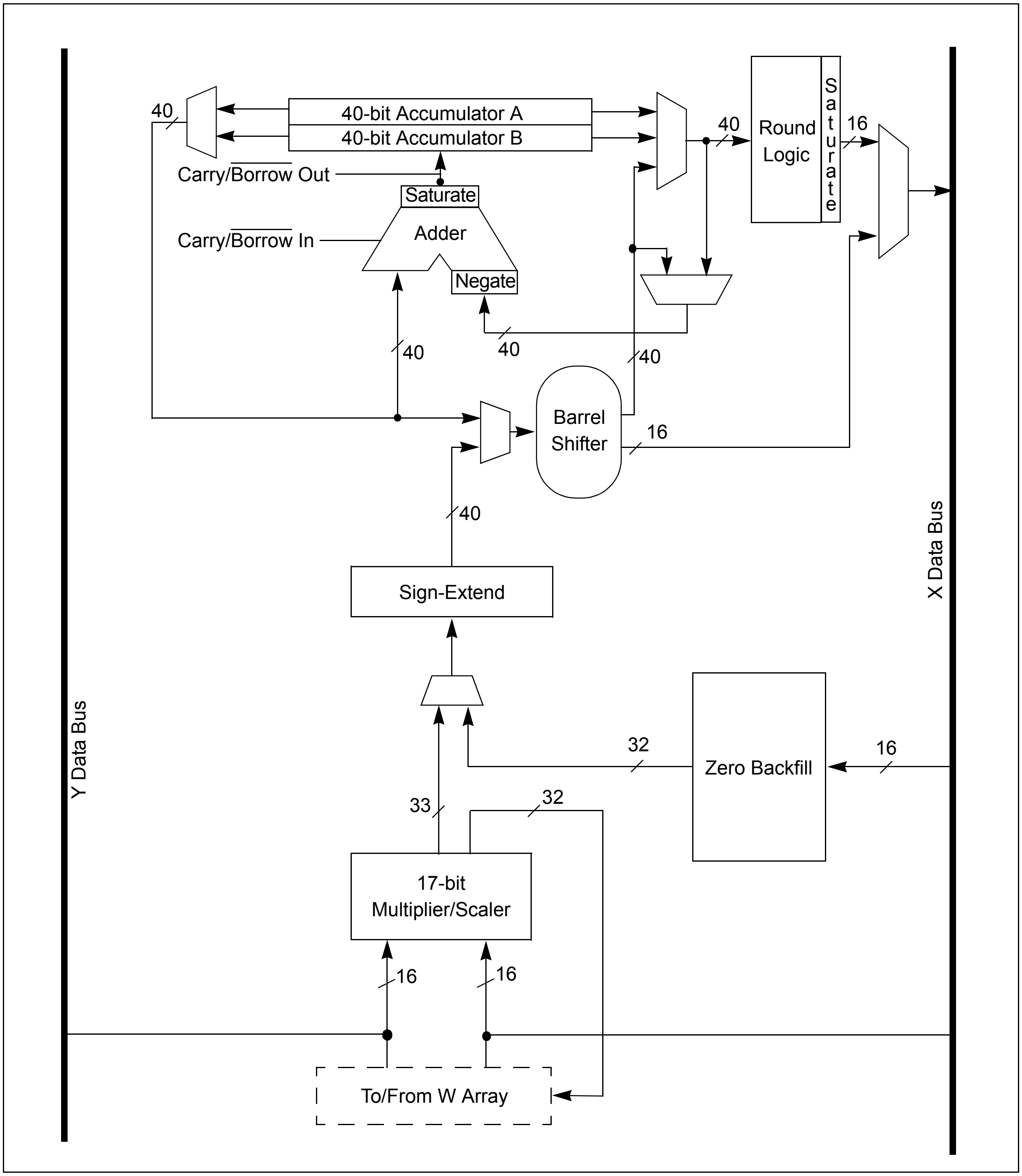
1. Fractional or integer DSP multiply (IF)
2. Signed or unsigned DSP multiply (US)
3. Conventional or convergent rounding (RND)
4. Automatic saturation on/off for AccA (SATA)
5. Automatic saturation on/off for AccB (SATB)
6. Automatic saturation on/off for writes to data memory (SATDW)
7. Accumulator Saturation mode selection (ACCSAT)

Table 3-1 provides a summary of DSP instructions. 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 * y)	Yes
`MAC`	A = A + x2	No
`MOVSAC`	No change in A	Yes
`MPY`	A = x * y	No
`MPY`	A = x 2	No
`MPY.N`	A = – x * y	No
`MSC`	A = A – x * y	Yes

FIGURE 3-3: DSP ENGINE BLOCK DIAGRAM



3.6.1 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 which is sign-extended to 40 bits. Integer data is inherently represented as a signed two's complement value, where the MSb is defined as a sign bit. Generally speaking, 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 $3.01518x10^{-5}$. In Fractional mode, the 16 x 16 multiply operation generates a 1.31 product which has a precision of $4.65661 \text{ x } 10^{-10}$.

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

The `MUL` instruction may be directed to use byte or word sized operands. Byte operands will direct a 16-bit result, and word operands will direct a 32-bit result to the specified register(s) in the W array.

3.6.2 DATA ACCUMULATORS AND ADDER/SUBTRACTER

The data accumulator consists of a 40-bit adder/subtracter 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 via the barrel shifter prior to accumulation.

3.6.2.1 Adder/Subtracter, Overflow and Saturation

The adder/subtracter 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), whereas in the case of subtraction, the Carry/Borrow input is active-low and the other input is complemented. The adder/subtracter 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 which controls accumulator data saturation, if selected. It uses the result of the adder, the Overflow Status bits described above 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 have been provided to support saturation and overflow; they are:

1. OA:
   AccA overflowed into guard bits
2. OB:
   AccB overflowed into guard bits
3. SA:
   AccA saturated (bit 31 overflow and saturation)
   or
   AccA overflowed into guard bits and saturated (bit 39 overflow and saturation)
4. SB:
   AccB saturated (bit 31 overflow and saturation)
   or
   AccB overflowed into guard bits and saturated (bit 39 overflow and saturation)
5. OAB:
   Logical OR of OA and OB
6. SAB:
   Logical OR of SA and SB

The OA and OB bits are modified each time data passes through the adder/subtracter. 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 they and the corresponding Overflow Trap Flag Enable bits (OVATE, OVBTE) in the INTCON1 register (refer to **Section 7.0 "Interrupt Controller"**) are set. This allows the user to take immediate action, for example, to correct system gain.

The SA and SB bits are modified each time data passes through the adder/subtracter, but can only be cleared by the user. 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 will be 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, SA and SB bits will 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). This allows programmers to 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 would be useful for complex number arithmetic, which typically uses both the accumulators.

The device supports three Saturation and Overflow modes:

1. Bit 39 Overflow and Saturation:
   When bit 39 overflow and saturation occurs, the saturation logic loads the maximally positive 9.31 (0x7FFFFFFFFF) or maximally negative 9.31 value (0x8000000000) into the target accumulator. The SA or SB bit is set and remains set until cleared by the user. This is referred to as 'super saturation' and provides protection against erroneous data or unexpected algorithm problems (e.g., gain calculations).
2. 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 (0x0080000000) into the target accumulator. The SA or SB bit is set and remains set until cleared by the user. When this Saturation mode is in effect, the guard bits are not used (so the OA, OB or OAB bits are never set).
3. 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. 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.

3.6.2.2 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:

1. W13, Register Direct:
   The rounded contents of the non-target accumulator are written into W13 as a 1.15 fraction.
2. [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).

3.6.2.3 Round Logic

The round logic is a combinational block which 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 which 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 3.6.2.4 "Data Space Write Saturation"**). For the `MAC` class of instructions, the accumulator write-back operation will function in the same manner, addressing combined MCU (X and Y) data space though the X bus. For this class of instructions, the data is always subject to rounding.

3.6.2.4 Data Space Write Saturation

In addition to adder/subtracter 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.

3.6.3 BARREL SHIFTER

The barrel shifter is capable of performing 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 to 31 for right shifts and between bit positions 0 to 16 for left shifts.

NOTES:

4.0 MEMORY ORGANIZATION

Note: This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 3. "Data Memory"** (DS70202) and **Section 4. "Program Memory"** (DS70203) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

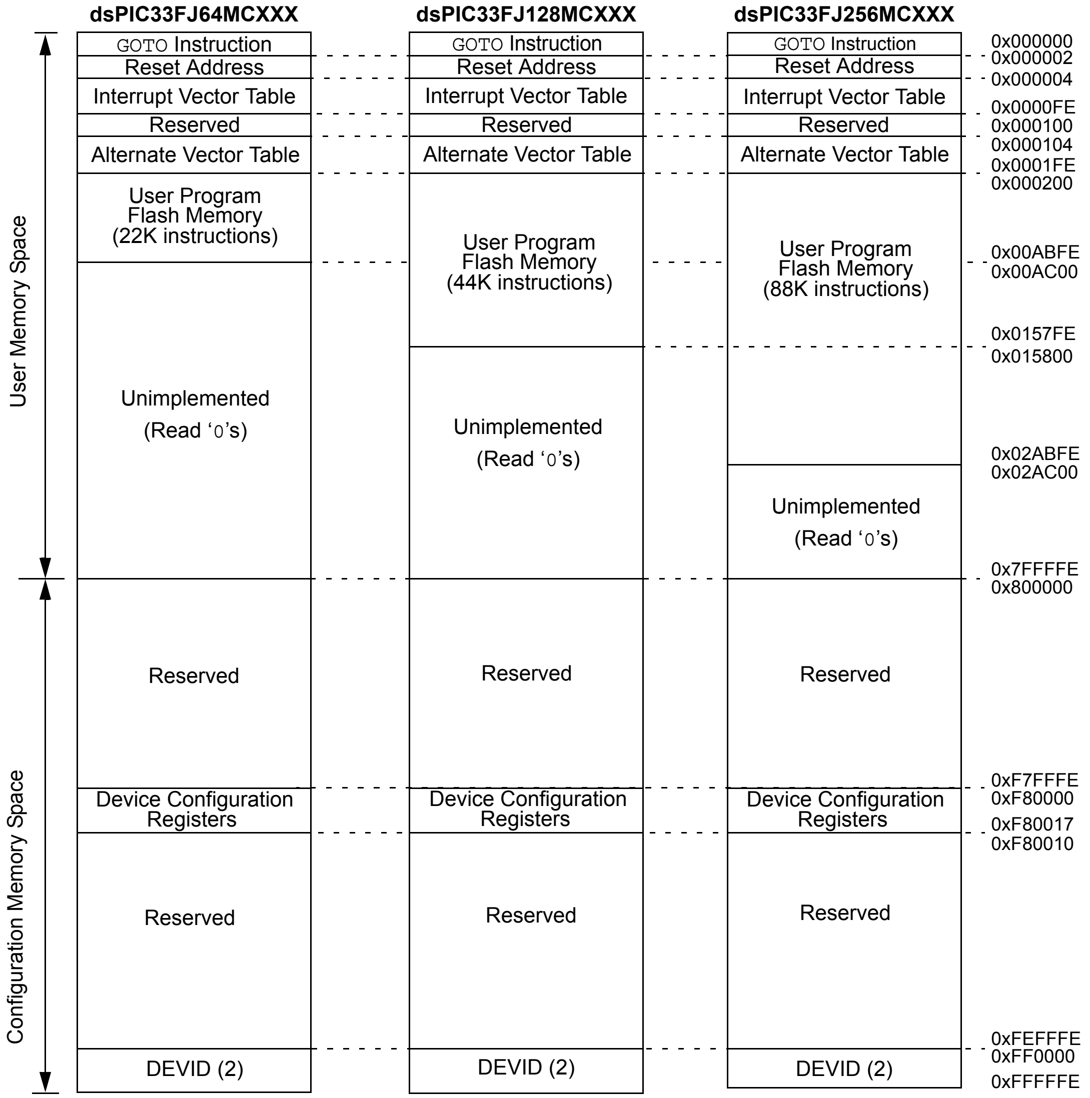
The dsPIC33FJXXXMCX06/X08/X10 architecture features separate program and data memory spaces and buses. This architecture also allows the direct access of program memory from the data space during code execution.

4.1 Program Address Space

The program address memory space of the dsPIC33FJXXXMCX06/X08/X10 devices is 4M instructions. The space is addressable by a 24-bit value derived from either the 23-bit Program Counter (PC) during program execution, or from table operation or data space remapping as described in **Section 4.6 "Interfacing Program and Data Memory Spaces"**.

User access to the program memory space is restricted to the lower half of the address range (0x000000 to 0x7FFFFF). The exception is the use of `TBLRD`/`TBLWT` operations, which use TBLPAG<7> to permit access to the Configuration bits and Device ID sections of the configuration memory space. Memory usage for the dsPIC33FJXXXMCX06/X08/X10 family of devices is shown in Figure 4-1.

FIGURE 4-1: PROGRAM MEMORY MAP FOR dsPIC33FJXXXMCX06/X08/X10 DEVICES



Note: Memory areas are not shown to scale.

4.1.1 PROGRAM MEMORY ORGANIZATION

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

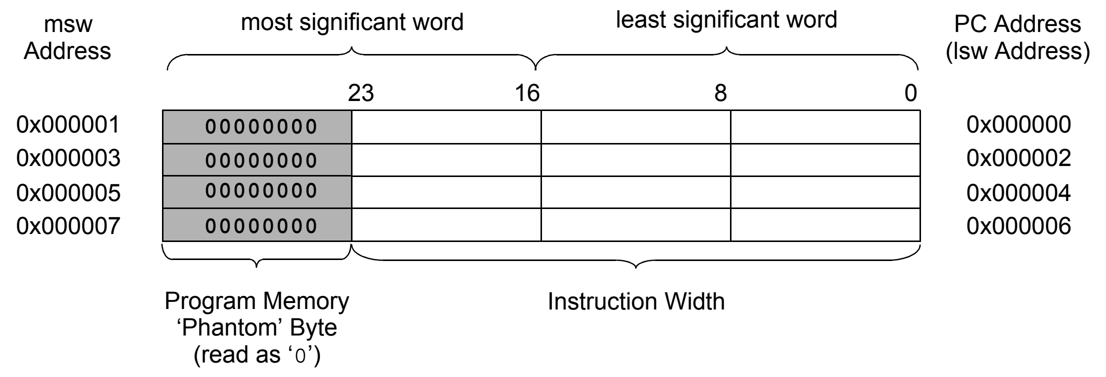
Program memory addresses are always word-aligned on the lower word, and addresses are incremented or decremented by two during code execution. This arrangement also provides compatibility with data memory space addressing and makes it possible to access data in the program memory space.

4.1.2 INTERRUPT AND TRAP VECTORS

All dsPIC33FJXXXMCX06/X08/X10 devices reserve the 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 at 0x000000, with the actual address for the start of code at 0x000002.

dsPIC33FJXXXMCX06/X08/X10 devices also have two interrupt vector tables located from 0x000004 to 0x0000FF and 0x000100 to 0x0001FF. These vector tables allow each of the many 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"**.

FIGURE 4-2: PROGRAM MEMORY ORGANIZATION



4.2 Data Address Space

The dsPIC33FJXXXMCX06/X08/X10 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. Data memory maps of devices with different RAM sizes are shown in Figure 4-3 through Figure 4-5.

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”**).

dsPIC33FJXXXMCX06/X08/X10 devices implement a total of up to 30 Kbytes of data memory. Should an EA point to a location outside of this area, an all-zero word or byte will be returned.

4.2.1 DATA SPACE WIDTH

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

4.2.2 DATA MEMORY ORGANIZATION AND ALIGNMENT

To maintain backward compatibility with PIC® microcontrollers and improve data space memory usage efficiency, the dsPIC33FJXXXMCX06/X08/X10 instruction set supports both word and byte operations. As a consequence of byte accessibility, all effective address calculations are internally scaled to step through word-aligned memory. For example, the core recognizes that Post-Modified Register Indirect Addressing mode [Ws++] will result in a value of Ws + 1 for byte operations and Ws + 2 for word operations.

Data byte reads will read the complete word 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 which matches the byte address.

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

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

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

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 dsPIC33FJXXXMCX06/X08/X10 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. Please refer to the corresponding device tables and pinout 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 dsPIC33FJXXXMCX06/X08/X10 DEVICES WITH 8 KBS RAM

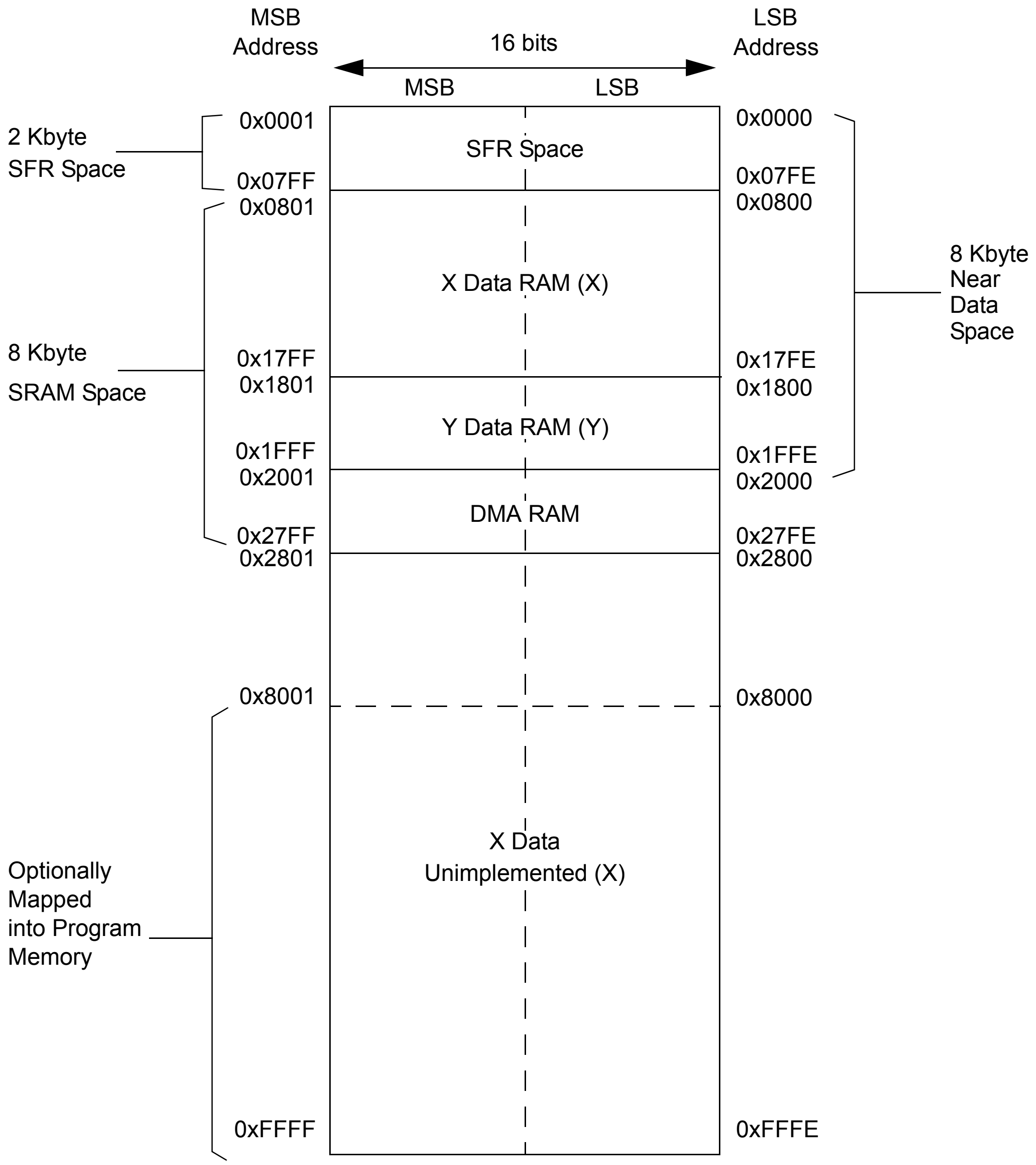


FIGURE 4-4: DATA MEMORY MAP FOR dsPIC33FJXXXMCX06/X08/X10 DEVICES WITH 16 KB RAM

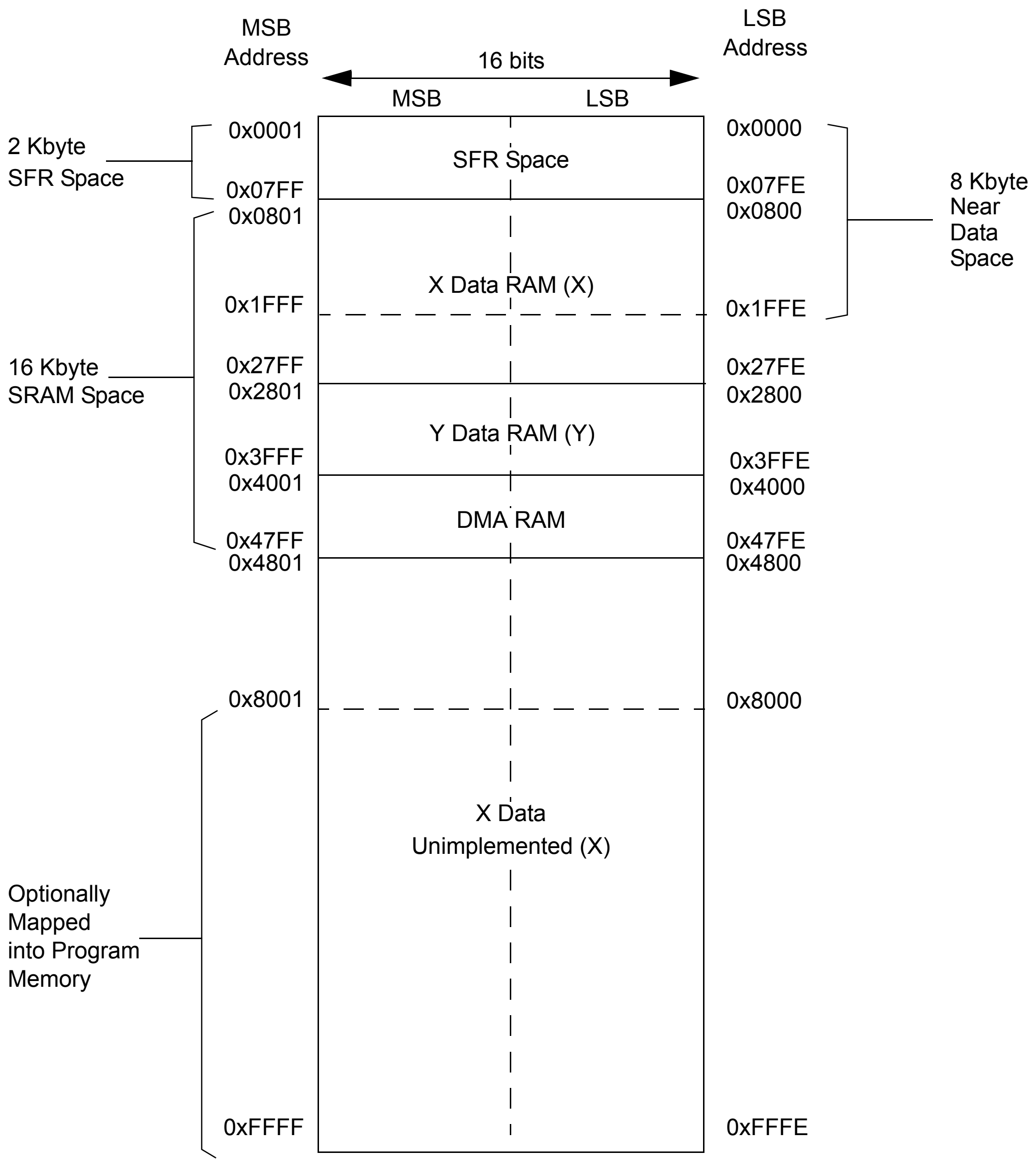
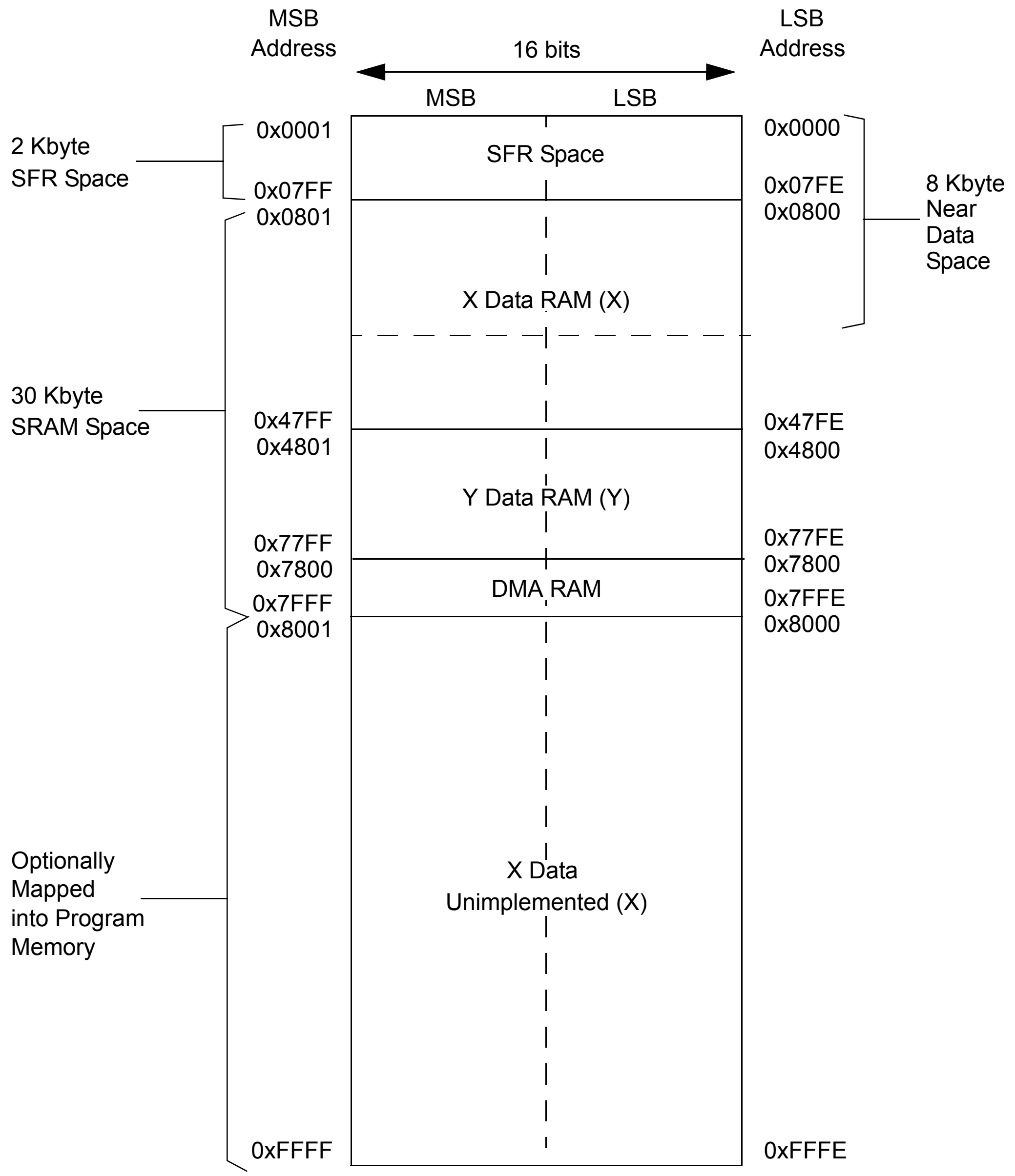


FIGURE 4-5: DATA MEMORY MAP FOR dsPIC33FJXXXMCX06/X08/X10 DEVICES WITH 30 KB RAM



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. There are separate read and write data buses for X data space. 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`, `MOVSAC`, `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 dsPIC33FJXXXMCX06/X08/X10 device contains 2 Kbytes of dual ported DMA RAM located at the end of Y data space. Memory locations is part of Y data RAM and is 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.

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	Accumulator A Low Word Register																0000
ACCAH	0024	Accumulator A High Word Register																0000
ACCAU	0026	Accumulator A Upper Word Register																0000
ACCBL	0028	Accumulator B Low Word Register																0000
ACCBH	002A	Accumulator B High Word Register																0000
ACCBU	002C	Accumulator B Upper Word Register																0000
PCL	002E	Program Counter Low Word Register																0000
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
XMODSRT	0048	XS<15:1>															0	xxxx
XMODEND	004A	XE<15:1>															1	xxxx

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
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
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 4-2: CHANGE NOTIFICATION REGISTER MAP FOR dsPIC33FJXXXMCX10 DEVICES

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	0062	—	—	—	—	—	—	—	—	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	—	—	—	—	—	—	—	—	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 4-3: CHANGE NOTIFICATION REGISTER MAP FOR dsPIC33FJXXXMCX08 DEVICES

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	0062	—	—	—	—	—	—	—	—	—	—	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	—	—	—	—	—	—	—	—	—	—	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: CHANGE NOTIFICATION REGISTER MAP FOR dsPIC33FJXXXMCX06 DEVICES

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	0062	—	—	—	—	—	—	—	—	—	—	CN21IE	CN20IE	—	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	—	—	—	—	—	—	—	—	—	—	CN21PUE	CN20PUE	—	CN18PUE	CN17PUE	CN16PUE	0000

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

TABLE 4-5: 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	COVBERR	OVATE	OVBTE	COVTE	SFTACERR	DIV0ERR	DMACERR	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	INT4EP	INT3EP	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	AD2IF	INT1IF	CNIF	—	MI2C1IF	SI2C1IF	0000
IFS2	0088	T6IF	DMA4IF	—	OC8IF	OC7IF	OC6IF	OC5IF	IC6IF	IC5IF	IC4IF	IC3IF	DMA3IF	C1IF	C1RXIF	SPI2IF	SPI2EIF	0000
IFS3	008A	FLTAIF	—	DMA5IF	—	—	QEIIF	PWMIF	C2IF	C2RXIF	INT4IF	INT3IF	T9IF	T8IF	MI2C2IF	SI2C2IF	T7IF	0000
IFS4	008C	—	—	—	—	—	—	—	—	C2TXIF	C1TXIF	DMA7IF	DMA6IF	—	U2EIF	U1EIF	FLTBIF	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	AD2IE	INT1IE	CNIE	—	MI2C1IE	SI2C1IE	0000
IEC2	0098	T6IE	DMA4IE	—	OC8IE	OC7IE	OC6IE	OC5IE	IC6IE	IC5IE	IC4IE	IC3IE	DMA3IE	C1IE	C1RXIE	SPI2IE	SPI2EIE	0000
IEC3	009A	FLTAIE	—	DMA5IE	—	—	QEIIE	PWMIE	C2IE	C2RXIE	INT4IE	INT3IE	T9IE	T8IE	MI2C2IE	SI2C2IE	T7IE	0000
IEC4	009C	—	—	—	—	—	—	—	—	C2TXIE	C1TXIE	DMA7IE	DMA6IE	—	U2EIE	U1EIE	FLTBIE	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>			—	—	—	—	—	MI2C1IP<2:0>			—	SI2C1IP<2:0>			4044
IPC5	00AE	—	IC8IP<2:0>			—	IC7IP<2:0>			—	AD2IP<2:0>			—	INT1IP<2:0>			4444
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>			—	C1RXIP<2:0>			—	SPI2IP<2:0>			—	SPI2EIP<2:0>			4444
IPC9	00B6	—	IC5IP<2:0>			—	IC4IP<2:0>			—	IC3IP<2:0>			—	DMA3IP<2:0>			4444
IPC10	00B8	—	OC7IP<2:0>			—	OC6IP<2:0>			—	OC5IP<2:0>			—	IC6IP<2:0>			4444
IPC11	00BA	—	T6IP<2:0>			—	DMA4IP<2:0>			—	—	—	—	—	OC8IP<2:0>			4404
IPC12	00BC	—	T8IP<2:0>			—	MI2C2IP<2:0>			—	SI2C2IP<2:0>			—	T7IP<2:0>			4444
IPC13	00BE	—	C2RXIP<2:0>			—	INT4IP<2:0>			—	INT3IP<2:0>			—	T9IP<2:0>			4444
IPC14	00C0	—	—	—	—	—	QEIIP<2:0>			—	PWMIP<2:0>			—	C2IP<2:0>			0444
IPC15	00C2	—	FLTAIP<2:0>			—	—	—	—	—	DMA5IP<2:0>			—	—	—	—	4040
IPC16	00C4	—	—	—	—	—	U2EIP<2:0>			—	U1EIP<2:0>			—	FLTBIP<2:0>			0444
IPC17	00C6	—	C2TXIP<2:0>			—	C1TXIP<2:0>			—	DMA7IP<2:0>			—	DMA6IP<2:0>			4444
INTTREG	00E0	—	—	—	—	ILR<3:0>				—	VECNUM<6:0>							0000

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

TABLE 4-6: 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 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
TMR6	0122	Timer6 Register																xxxx
TMR7HLD	0124	Timer7 Holding Register (for 32-bit operations only)																xxxx
TMR7	0126	Timer7 Register																xxxx
PR6	0128	Period Register 6																FFFF
PR7	012A	Period Register 7																FFFF
T6CON	012C	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS<1:0>		T32	—	TCS	—	0000
T7CON	012E	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS<1:0>		—	—	TCS	—	0000
TMR8	0130	Timer8 Register																xxxx
TMR9HLD	0132	Timer9 Holding Register (for 32-bit operations only)																xxxx
TMR9	0134	Timer9 Register																xxxx
PR8	0136	Period Register 8																FFFF
PR9	0138	Period Register 9																FFFF
T8CON	013A	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS<1:0>		T32	—	TCS	—	0000
T9CON	013C	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 4-7: 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
IC3BUF	0148	Input 3 Capture Register																xxxx
IC3CON	014A	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>			0000
IC4BUF	014C	Input 4 Capture Register																xxxx
IC4CON	014E	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>			0000
IC5BUF	0150	Input 5 Capture Register																xxxx
IC5CON	0152	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>			0000
IC6BUF	0154	Input 6 Capture Register																xxxx
IC6CON	0156	—	—	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 4-8: 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
OC5RS	0198	Output Compare 5 Secondary Register																xxxx
OC5R	019A	Output Compare 5 Register																xxxx
OC5CON	019C	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000
OC6RS	019E	Output Compare 6 Secondary Register																xxxx
OC6R	01A0	Output Compare 6 Register																xxxx
OC6CON	01A2	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000
OC7RS	01A4	Output Compare 7 Secondary Register																xxxx
OC7R	01A6	Output Compare 7 Register																xxxx
OC7CON	01A8	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000
OC8RS	01AA	Output Compare 8 Secondary Register																xxxx
OC8R	01AC	Output Compare 8 Register																xxxx
OC8CON	01AE	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000

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

TABLE 4-9: 8-OUTPUT PWM 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
P1TCON	01C0	PTEN	—	PTSIDL	—	—	—	—	—	PTOPS<3:0>				PTCKPS<1:0>		PTMOD<1:0>		0000 0000 0000 0000
P1TMR	01C2	PTDIR	PWM Timer Count Value Register															0000 0000 0000 0000
P1TPER	01C4	—	PWM Time Base Period Register															0000 0000 0000 0000
P1SECMP	01C6	SEVTDIR	PWM Special Event Compare Register															0000 0000 0000 0000
PWM1CON1	01C8	—	—	—	—	PMOD4	PMOD3	PMOD2	PMOD1	PEN4H	PEN3H	PEN2H	PEN1H	PEN4L	PEN3L	PEN2L	PEN1L	0000 0000 1111 1111
PWM1CON2	01CA	—	—	—	—	SEVOPS<3:0>				—	—	—	—	—	IUE	OSYNC	UDIS	0000 0000 0000 0000
P1DTCON1	01CC	DTBPS<1:0>		DTB<5:0>						DTAPS<1:0>		DTA<5:0>						0000 0000 0000 0000
P1DTCON2	01CE	—	—	—	—	—	—	—	—	DTS4A	DTS4I	DTS3A	DTS3I	DTS2A	DTS2I	DTS1A	DTS1I	0000 0000 0000 0000
P1FLTACON	01D0	FAOV4H	FAOV4L	FAOV3H	FAOV3L	FAOV2H	FAOV2L	FAOV1H	FAOV1L	FLTAM	—	—	—	FAEN4	FAEN3	FAEN2	FAEN1	0000 0000 0000 0000
P1FLTBCON	01D2	FBOV4H	FBOV4L	FBOV3H	FBOV3L	FBOV2H	FBOV2L	FBOV1H	FBOV1L	FLTBM	—	—	—	FBEN4	FBEN3	FBEN2	FBEN1	0000 0000 0000 0000
P1OVDCON	01D4	POVD4H	POVD4L	POVD3H	POVD3L	POVD2H	POVD2L	POVD1H	POVD1L	POUT4H	POUT4L	POUT3H	POUT3L	POUT2H	POUT2L	POUT1H	POUT1L	1111 1111 0000 0000
P1DC1	01D6	PWM Duty Cycle #1 Register																0000 0000 0000 0000
P1DC2	01D8	PWM Duty Cycle #2 Register																0000 0000 0000 0000
P1DC3	01DA	PWM Duty Cycle #3 Register																0000 0000 0000 0000
P1DC4	01DC	PWM Duty Cycle #4 Register																0000 0000 0000 0000

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

TABLE 4-10: QEI 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
QEI1CON	01E0	CNTERR	—	QEISIDL	INDX	UPDN	QEIM<2:0>			SWPAB	PCDOUT	TQGATE	TQCKPS<1:0>		POSRES	TQCS	UPDN_SRC	0000 0000 0000 0000
DFLT1CON	01E2	—	—	—	—	—	IMV<1:0>		CEID	QEOUT	QECK<2:0>			—	—	—	—	0000 0000 0000 0000
POS1CNT	01E4	Position Counter<15:0>																0000 0000 0000 0000
MAX1CNT	01E6	Maximum Count<15:0>																1111 1111 1111 1111

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

TABLE 4-11: 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 4-12: I2C2 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
I2C2RCV	0210	—	—	—	—	—	—	—	—	Receive Register								0000
I2C2TRN	0212	—	—	—	—	—	—	—	—	Transmit Register								00FF
I2C2BRG	0214	—	—	—	—	—	—	—	Baud Rate Generator Register									0000
I2C2CON	0216	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C2STAT	0218	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D_A	P	S	R_W	RBF	TBF	0000
I2C2ADD	021A	—	—	—	—	—	—	Address Register										0000
I2C2MSK	021C	—	—	—	—	—	—	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	—	—	—	—	—	—	—	UART Transmit Register									xxxx
U1RXREG	0226	—	—	—	—	—	—	—	UART Receive 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	—	—	—	—	—	—	—	UART Transmit Register									xxxx
U2RXREG	0236	—	—	—	—	—	—	—	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

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
AD1PCFGH[1]	032A	PCFG31	PCFG30	PCFG29	PCFG28	PCFG27	PCFG26	PCFG25	PCFG24	PCFG23	PCFG22	PCFG21	PCFG20	PCFG19	PCFG18	PCFG17	PCFG16	0000
AD1PCFGL	032C	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
AD1CSSH[1]	032E	CSS31	CSS30	CSS29	CSS28	CSS27	CSS26	CSS25	CSS24	CSS23	CSS22	CSS21	CSS20	CSS19	CSS18	CSS17	CSS16	0000
AD1CSSL	0330	CSS15	CSS14	CSS13	CSS12	CSS11	CSS10	CSS9	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.

Note 1: Not all ANx inputs are available on all devices. Refer to the device pin diagrams for available ANx inputs.

TABLE 4-18: ADC2 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
ADC2BUF0	0340	ADC Data Buffer 0																xxxx
AD2CON1	0360	ADON	—	ADSIDL	ADDMABM	—	AD12B	FORM<1:0>		SSRC<2:0>			—	SIMSAM	ASAM	SAMP	DONE	0000
AD2CON2	0362	VCFG<2:0>			—	—	CSCNA	CHPS<1:0>		BUFS	—	SMPI<3:0>				BUFM	ALTS	0000
AD2CON3	0364	ADRC	—	—	SAMC<4:0>					ADCS<7:0>								0000
AD2CHS123	0366	—	—	—	—	—	CH123NB<1:0>		CH123SB	—	—	—	—	—	CH123NA<1:0>		CH123SA	0000
AD2CHS0	0368	CH0NB	—	—	—	CH0SB<3:0>				CH0NA	—	—	—	CH0SA<3:0>				0000
Reserved	036A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
AD2PCFGL	036C	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
Reserved	036E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
AD2CSSL	0370	CSS15	CSS14	CSS13	CSS12	CSS11	CSS10	CSS9	CSS8	CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0	0000
AD2CON4	0372	—	—	—	—	—	—	—	—	—	—	—	—	—	DMABL<2:0>			0000

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

TABLE 4-19: 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 4-19: 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 4-20: ECAN1 REGISTER MAP WHEN C1CTRL1.WIN = 0 OR 1

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 4-21: ECAN1 REGISTER MAP WHEN C1CTRL1.WIN = 0

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	TXABAT0	TXLARB0	TXERR0	TXREQ0	RTREN0	TX0PRI<1:0>		0000
C1TR23CON	0432	TXEN3	TXABT3	TXLARB3	TXERR3	TXREQ3	RTREN3	TX3PRI<1:0>		TXEN2	TXABAT2	TXLARB2	TXERR2	TXREQ2	RTREN2	TX2PRI<1:0>		0000
C1TR45CON	0434	TXEN5	TXABT5	TXLARB5	TXERR5	TXREQ5	RTREN5	TX5PRI<1:0>		TXEN4	TXABAT4	TXLARB4	TXERR4	TXREQ4	RTREN4	TX4PRI<1:0>		0000
C1TR67CON	0436	TXEN7	TXABT7	TXLARB7	TXERR7	TXREQ7	RTREN7	TX7PRI<1:0>		TXEN6	TXABAT6	TXLARB6	TXERR6	TXREQ6	RTREN6	TX6PRI<1:0>		xxxx
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 4-22: ECAN1 REGISTER MAP WHEN C1CTRL1.WIN = 1

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																
C1BUFPNT1	0420	F3BP<3:0>				F2BP<3:0>				F1BP<3:0>				F0BP<3:0>				0000
C1BUFPNT2	0422	F7BP<3:0>				F6BP<3:0>				F5BP<3:0>				F4BP<3:0>				0000
C1BUFPNT3	0424	F11BP<3:0>				F10BP<3:0>				F9BP<3:0>				F8BP<3:0>				0000
C1BUFPNT4	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

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

TABLE 4-22: ECAN1 REGISTER MAP WHEN C1CTRL1.WIN = 1(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
C1RXF11SID	046C	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
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 4-23: ECAN2 REGISTER MAP WHEN C2CTRL1.WIN = 0 OR 1 FOR dsPIC33FJXXXMC708/710 DEVICES

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
C2CTRL1	0500	—	—	CSIDL	ABAT	—	REQOP<2:0>			OPMODE<2:0>			—	CANCAP	—	—	WIN	0480
C2CTRL2	0502	—	—	—	—	—	—	—	—	—	—	—	DNCNT<4:0>					0000
C2VEC	0504	—	—	—	FILHIT<4:0>					—	ICODE<6:0>							0000
C2FCTRL	0506	DMABS<2:0>			—	—	—	—	—	—	—	—	FSA<4:0>					0000
C2FIFO	0508	—	—	FBP<5:0>						—	—	FNRB<5:0>						0000
C2INTF	050A	—	—	TXBO	TXBP	RXBP	TXWAR	RXWAR	EWARN	IVRIF	WAKIF	ERRIF	—	FIFOIF	RBOVIF	RBIF	TBIF	0000
C2INTE	050C	—	—	—	—	—	—	—	—	IVRIE	WAKIE	ERRIE	—	FIFOIE	RBOVIE	RBIE	TBIE	0000
C2EC	050E	TERRCNT<7:0>								RERRCNT<7:0>								0000
C2CFG1	0510	—	—	—	—	—	—	—	—	SJW<1:0>		BRP<5:0>						0000
C2CFG2	0512	—	WAKFIL	—	—	—	SEG2PH<2:0>			SEG2PHTS	SAM	SEG1PH<2:0>			PRSEG<2:0>			0000
C2FEN1	0514	FLTEN15	FLTEN14	FLTEN13	FLTEN12	FLTEN11	FLTEN10	FLTEN9	FLTEN8	FLTEN7	FLTEN6	FLTEN5	FLTEN4	FLTEN3	FLTEN2	FLTEN1	FLTEN0	FFFF
C2FMSKSEL1	0518	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
C2FMSKSEL2	051A	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 4-24: ECAN2 REGISTER MAP WHEN C2CTRL1.WIN = 0 FOR dsPIC33FJXXXMC708/710 DEVICES

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
	0500-051E	**See definition when WIN = x**																
C2RXFUL1	0520	RXFUL15	RXFUL14	RXFUL13	RXFUL12	RXFUL11	RXFUL10	RXFUL9	RXFUL8	RXFUL7	RXFUL6	RXFUL5	RXFUL4	RXFUL3	RXFUL2	RXFUL1	RXFUL0	0000
C2RXFUL2	0522	RXFUL31	RXFUL30	RXFUL29	RXFUL28	RXFUL27	RXFUL26	RXFUL25	RXFUL24	RXFUL23	RXFUL22	RXFUL21	RXFUL20	RXFUL19	RXFUL18	RXFUL17	RXFUL16	0000
C2RXOVF1	0528	RXOVF15	RXOVF14	RXOVF13	RXOVF12	RXOVF11	RXOVF10	RXOVF09	RXOVF08	RXOVF7	RXOVF6	RXOVF5	RXOVF4	RXOVF3	RXOVF2	RXOVF1	RXOVF0	0000
C2RXOVF2	052A	RXOVF31	RXOVF30	RXOVF29	RXOVF28	RXOVF27	RXOVF26	RXOVF25	RXOVF24	RXOVF23	RXOVF22	RXOVF21	RXOVF20	RXOVF19	RXOVF18	RXOVF17	RXOVF16	0000
C2TR01CON	0530	TXEN1	TX ABAT1	TX LARB1	TX ERR1	TX REQ1	RTREN1	TX1PRI<1:0>		TXEN0	TX ABAT0	TX LARB0	TX ERR0	TX REQ0	RTREN0	TX0PRI<1:0>		0000
C2TR23CON	0532	TXEN3	TX ABAT3	TX LARB3	TX ERR3	TX REQ3	RTREN3	TX3PRI<1:0>		TXEN2	TX ABAT2	TX LARB2	TX ERR2	TX REQ2	RTREN2	TX2PRI<1:0>		0000
C2TR45CON	0534	TXEN5	TX ABAT5	TX LARB5	TX ERR5	TX REQ5	RTREN5	TX5PRI<1:0>		TXEN4	TX ABAT4	TX LARB4	TX ERR4	TX REQ4	RTREN4	TX4PRI<1:0>		0000
C2TR67CON	0536	TXEN7	TX ABAT7	TX LARB7	TX ERR7	TX REQ7	RTREN7	TX7PRI<1:0>		TXEN6	TX ABAT6	TX LARB6	TX ERR6	TX REQ6	RTREN6	TX6PRI<1:0>		xxxx
C2RXD	0540	Recieved Data Word																xxxx
C2TXD	0542	Transmit Data Word																xxxx

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

TABLE 4-25: ECAN2 REGISTER MAP WHEN C2CTRL1.WIN = 1 FOR dsPIC33FJXXXMC708/710 DEVICES

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
	0500-051E	See definition when WIN = x																
C2BUFPNT1	0520	F3BP<3:0>				F2BP<3:0>				F1BP<3:0>				F0BP<3:0>				0000
C2BUFPNT2	0522	F7BP<3:0>				F6BP<3:0>				F5BP<3:0>				F4BP<3:0>				0000
C2BUFPNT3	0524	F11BP<3:0>				F10BP<3:0>				F9BP<3:0>				F8BP<3:0>				0000
C2BUFPNT4	0526	F15BP<3:0>				F14BP<3:0>				F13BP<3:0>				F12BP<3:0>				0000
C2RXM0SID	0530	SID<10:3>								SID<2:0>			—	MIDE	—	EID<17:16>		xxxx
C2RXM0EID	0532	EID<15:8>								EID<7:0>								xxxx
C2RXM1SID	0534	SID<10:3>								SID<2:0>			—	MIDE	—	EID<17:16>		xxxx
C2RXM1EID	0536	EID<15:8>								EID<7:0>								xxxx
C2RXM2SID	0538	SID<10:3>								SID<2:0>			—	MIDE	—	EID<17:16>		xxxx
C2RXM2EID	053A	EID<15:8>								EID<7:0>								xxxx
C2RXF0SID	0540	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
C2RXF0EID	0542	EID<15:8>								EID<7:0>								xxxx
C2RXF1SID	0544	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
C2RXF1EID	0546	EID<15:8>								EID<7:0>								xxxx
C2RXF2SID	0548	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
C2RXF2EID	054A	EID<15:8>								EID<7:0>								xxxx
C2RXF3SID	054C	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
C2RXF3EID	054E	EID<15:8>								EID<7:0>								xxxx
C2RXF4SID	0550	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
C2RXF4EID	0552	EID<15:8>								EID<7:0>								xxxx
C2RXF5SID	0554	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
C2RXF5EID	0556	EID<15:8>								EID<7:0>								xxxx
C2RXF6SID	0558	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
C2RXF6EID	055A	EID<15:8>								EID<7:0>								xxxx
C2RXF7SID	055C	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
C2RXF7EID	055E	EID<15:8>								EID<7:0>								xxxx
C2RXF8SID	0560	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
C2RXF8EID	0562	EID<15:8>								EID<7:0>								xxxx
C2RXF9SID	0564	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
C2RXF9EID	0566	EID<15:8>								EID<7:0>								xxxx
C2RXF10SID	0568	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
C2RXF10EID	056A	EID<15:8>								EID<7:0>								xxxx

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

TABLE 4-25: ECAN2 REGISTER MAP WHEN C2CTRL1.WIN = 1 FOR dsPIC33FJXXXMC708/710 DEVICES (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
C2RXF11SID	056C	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
C2RXF11EID	056E	EID<15:8>								EID<7:0>								xxxx
C2RXF12SID	0570	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
C2RXF12EID	0572	EID<15:8>								EID<7:0>								xxxx
C2RXF13SID	0574	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
C2RXF13EID	0576	EID<15:8>								EID<7:0>								xxxx
C2RXF14SID	0578	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
C2RXF14EID	057A	EID<15:8>								EID<7:0>								xxxx
C2RXF15SID	057C	SID<10:3>								SID<2:0>			—	EXIDE	—	EID<17:16>		xxxx
C2RXF15EID	057E	EID<15:8>								EID<7:0>								xxxx

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

TABLE 4-26: PORTA REGISTER MAP[(1)]

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	TRISA15	TRISA14	—	—	—	TRISA10	TRISA9	—	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	C6FF
PORTA	02C2	RA15	RA14	—	—	—	RA10	RA9	—	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx
LATA	02C4	LATA15	LATA14	—	—	—	LATA10	LATA9	—	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx
ODCA	06C0	ODCA15	ODCA14	—	—	—	—	—	—	—	—	ODCA5	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0	0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal for PinHigh devices.
Note 1: The actual set of I/O port pins varies from one device to another. Please refer to the corresponding pinout diagrams.

TABLE 4-27: PORTB REGISTER MAP[(1)]

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	02C6	TRISB15	TRISB14	TRISB13	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	FFFF
PORTB	02C8	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx
LATB	02CA	LATB15	LATB14	LATB13	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal for PinHigh devices.
Note 1: The actual set of I/O port pins varies from one device to another. Please refer to the corresponding pinout diagrams.

TABLE 4-28: PORTC REGISTER MAP[(1)]

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	02CC	TRISC15	TRISC14	TRISC13	TRISC12	—	—	—	—	—	—	—	TRISC4	TRISC3	TRISC2	TRISC1	—	F01E
PORTC	02CE	RC15	RC14	RC13	RC12	—	—	—	—	—	—	—	RC4	RC3	RC2	RC1	—	xxxx
LATC	02D0	LATC15	LATC14	LATC13	LATC12	—	—	—	—	—	—	—	LATC4	LATC3	LATC2	LATC1	—	xxxx

Legend: x = unknown value on Reset, — = unimplemented, read as ‘0’. Reset values are shown in hexadecimal for PinHigh devices.
Note 1: The actual set of I/O port pins varies from one device to another. Please refer to the corresponding pinout diagrams.

TABLE 4-29: PORTD REGISTER MAP[(1)]

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
TRISD	02D2	TRISD15	TRISD14	TRISD13	TRISD12	TRISD11	TRISD10	TRISD9	TRISD8	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	FFFF
PORTD	02D4	RD15	RD14	RD13	RD12	RD11	RD10	RD9	RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx
LATD	02D6	LATD15	LATD14	LATD13	LATD12	LATD11	LATD10	LATD9	LATD8	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	xxxx
ODCD	06D2	ODCD15	ODCD14	ODCD13	ODCD12	ODCD11	ODCD10	ODCD9	ODCD8	ODCD7	ODCD6	ODCD5	ODCD4	ODCD3	ODCD2	ODCD1	ODCD0	0000

Legend: x = unknown value on Reset, — = unimplemented, read as ‘0’. Reset values are shown in hexadecimal for PinHigh devices.
Note 1: The actual set of I/O port pins varies from one device to another. Please refer to the corresponding pinout diagrams.

TABLE 4-30: PORTE REGISTER MAP[(1)]

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
TRISE	02D8	—	—	—	—	—	—	TRISE9	TRISE8	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	01FF
PORTE	02DA	—	—	—	—	—	—	RE9	RE8	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	xxxx
LATE	02DC	—	—	—	—	—	—	LATE9	LATE8	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	xxxx

Legend: x = unknown value on Reset, — = unimplemented, read as ‘0’. Reset values are shown in hexadecimal for PinHigh devices.
Note 1: The actual set of I/O port pins varies from one device to another. Please refer to the corresponding pinout diagrams.

TABLE 4-31: PORTF REGISTER MAP[(1)]

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
TRISF	02DE	—	—	TRISF13	TRISF12	—	—	—	TRISF8	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	31FF
PORTF	02E0	—	—	RF13	RF12	—	—	—	RF8	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0	xxxx
LATF	02E2	—	—	LATF13	LATF12	—	—	—	LATF8	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0	xxxx
ODCF	06DE	—	—	ODCF13	ODCF12	—	—	—	ODCF8	ODCF7	ODCF6	ODCF5	ODCF4	ODCF3	ODCF2	ODCF1	ODCF0	0000

Legend: x = unknown value on Reset, — = unimplemented, read as ‘0’. Reset values are shown in hexadecimal for PinHigh devices.
Note 1: The actual set of I/O port pins varies from one device to another. Please refer to the corresponding pinout diagrams.

TABLE 4-32: PORTG REGISTER MAP[1]

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
TRISG	02E4	TRISG15	TRISG14	TRISG13	TRISG12	—	—	TRISG9	TRISG8	TRISG7	TRISG6	—	—	TRISG3	TRISG2	TRISG1	TRISG0	F3CF
PORTG	02E6	RG15	RG14	RG13	RG12	—	—	RG9	RG8	RG7	RG6	—	—	RG3	RG2	RG1	RG0	xxxx
LATG	02E8	LATG15	LATG14	LATG13	LATG12	—	—	LATG9	LATG8	LATG7	LATG6	—	—	LATG3	LATG2	LATG1	LATG0	xxxx
ODCG	06E4	ODCG15	ODCG14	ODCG13	ODCG12	—	—	ODCG9	ODCG8	ODCG7	ODCG6	—	—	ODCG3	ODCG2	ODCG1	ODCG0	0000

Legend: x = unknown value on Reset, — = unimplemented, read as ‘0’. Reset values are shown in hexadecimal for PinHigh devices.

Note 1: The actual set of I/O port pins varies from one device to another. Please refer to the corresponding pinout diagrams.

TABLE 4-33: 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	—	—	—	—	—	VREGS	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	xxxx[1]
OSCCON	0742	—	COSC<2:0>			—	NOSC<2:0>			CLKLOCK	—	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

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.

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

TABLE 4-34: 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[1]
NVMKEY	0766	—	—	—	—	—	—	—	—	NVMKEY<7:0>								0000

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

Note 1: Reset value shown is for POR only. Value on other Reset states is dependent on the state of memory write or erase operations at the time of Reset.

TABLE 4-35: 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	QEIMD	PWMMD	—	I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	C2MD	C1MD	AD1MD	0000
PMD2	0772	IC8MD	IC7MD	IC6MD	IC5MD	IC4MD	IC3MD	IC2MD	IC1MD	OC8MD	OC7MD	OC6MD	OC5MD	OC4MD	OC3MD	OC2MD	OC1MD	0000
PMD3	0774	T9MD	T8MD	T7MD	T6MD	—	—	—	—	—	—	—	—	—	—	I2C2MD	AD2MD	0000

Legend: x = unknown value on Reset, — = unimplemented, read as ‘0’. Reset values are shown in hexadecimal for PinHigh devices.

4.2.7 SOFTWARE STACK

In addition to its use as a working register, the W15 register in the dsPIC33FJXXXMCX06/X08/X10 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.

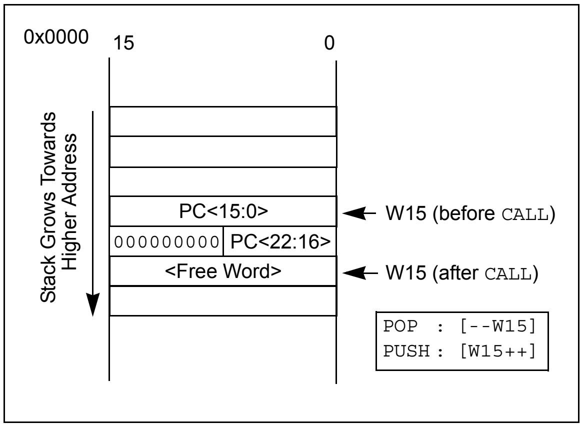
Note: A PC push during exception processing concatenates the SRL register to the MSb of the PC prior to the push.

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 will not occur. The stack error trap will occur on a subsequent push operation. Thus, for example, if it is desirable to cause a stack error trap when the stack grows beyond address 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 dsPIC33FJXXXMCX06/X08/X10 devices supports Data RAM protection features which 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 in Table 4-36 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 are somewhat different 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 3-operand MCU instructions are of the following form:

Operand 3 = Operand 1 <function> Operand 2

where Operand 1 is always a working register (i.e., 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 may support different subsets of these addressing modes.

TABLE 4-36: 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 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	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 between 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, utilize a simplified set of addressing modes to allow the user to effectively manipulate the data pointers through register indirect tables.

The 2-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 will always be 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 only available 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 various addressing modes outlined above, 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 `NOP`, 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 only be configured to operate in 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 which have a power-of-2 length. As these buffers satisfy the start and end address criteria, they may operate in a bidirectional mode (i.e., address boundary checks will be 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).

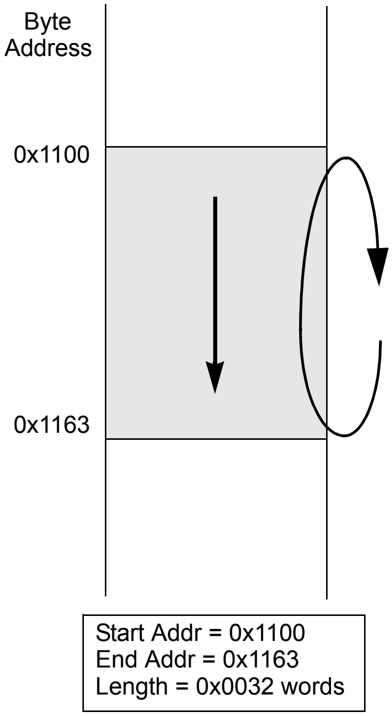
4.4.2 W ADDRESS REGISTER SELECTION

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 which registers will operate with Modulo Addressing. If XWM = `15`, X RAGU and X WAGU Modulo Addressing is disabled. Similarly, 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



```
MOV     #0x1100, W0
MOV     W0, XMODSRT     ;set modulo start address
MOV     #0x1163, W0
MOV     W0, MODEND      ;set modulo end address
MOV     #0x8001, W0
MOV     W0, MODCON      ;enable W1, X AGU for modulo

MOV     #0x0000, W0     ;W0 holds buffer fill value

MOV     #0x1110, W1     ;point W1 to buffer

DO      AGAIN, #0x31    ;fill the 50 buffer locations
MOV     W0, [W1++]      ;fill the next location
AGAIN:  INC  W0, W0     ;increment the fill value
```

4.4.3 MODULO ADDRESSING APPLICABILITY

Modulo Addressing can be applied to the Effective Address (EA) calculation associated with any W register. 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 may, 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 (e.g., [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 may 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 when the following conditions exist:

1. The BWM bits (W register selection) in the MODCON register are any value other than ‘`15`’ (the stack cannot be accessed using Bit-Reversed Addressing).
2. The BREN bit is set in the XBREV register.
3. 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 only executed for Register Indirect with Pre-Increment or Post-Increment Addressing and word sized data writes. It will not function for any other addressing mode or for byte sized data; 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. In the event that the user attempts to do so, Bit-Reversed Addressing will assume priority for the X WAGU, and X WAGU Modulo Addressing will be disabled. However, Modulo Addressing will continue to function in the X RAGU.

If Bit-Reversed Addressing has already been enabled by setting the BREN (XBREV<15>) bit, then 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

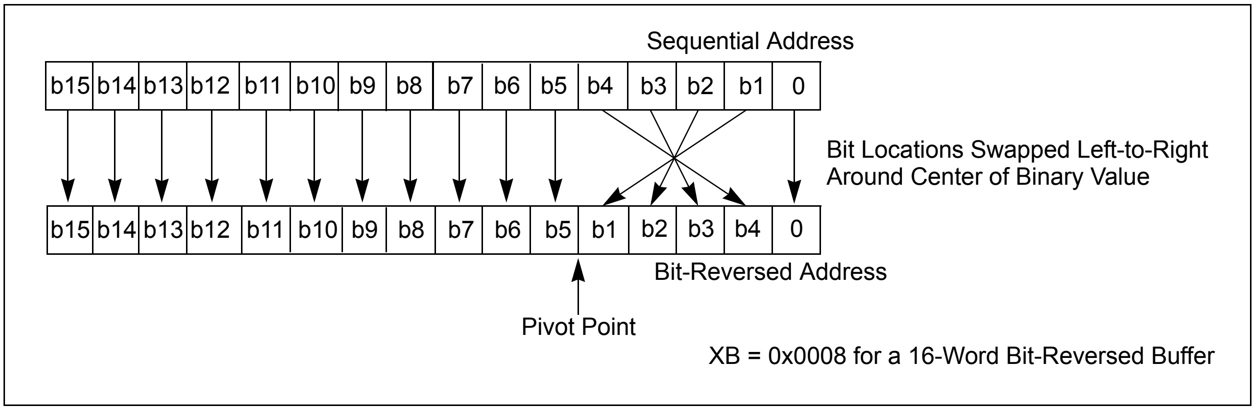


TABLE 4-37: BIT-REVERSED ADDRESS SEQUENCE (16-ENTRY)

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 dsPIC33FJXXXMCX06/X08/X10 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 dsPIC33FJXXXMCX06/X08/X10 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 from time to time. It also allows access to all bytes of the program word. The remapping method allows an application to access a large block of data on a read-only basis, which is ideal for look ups from a large table of static data. It can only access the least significant word of the program word.

4.6.1 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.

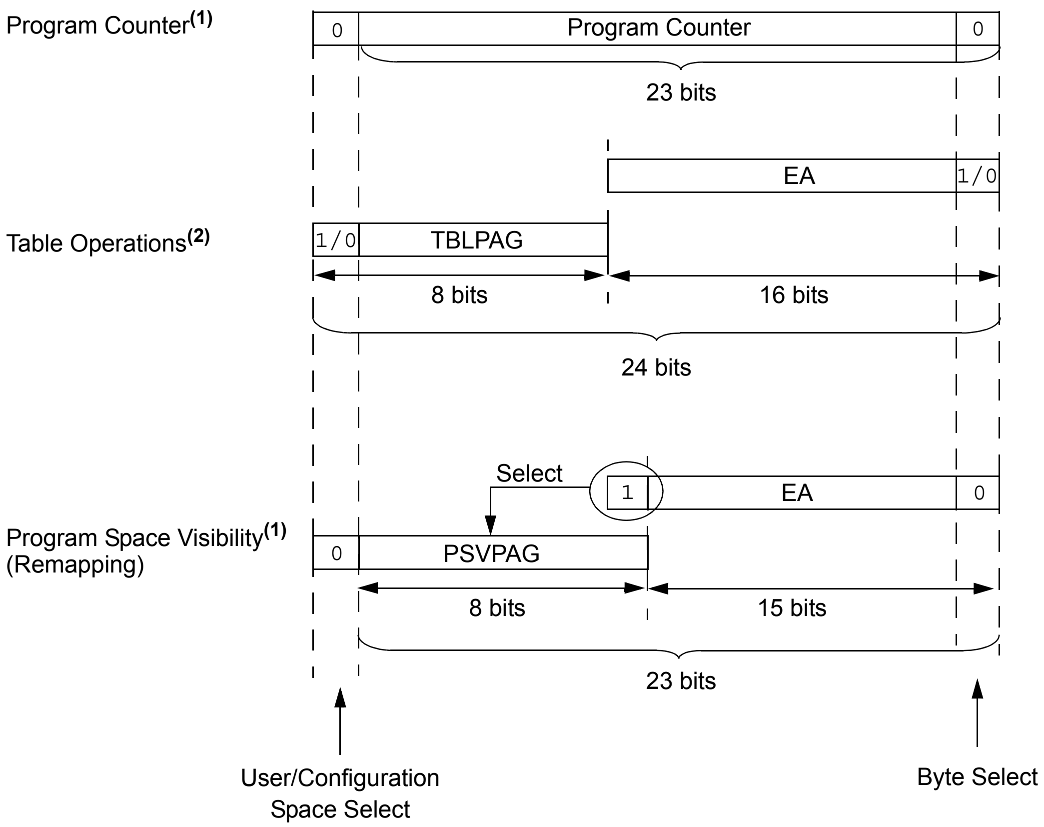
Table 4-38 and Figure 4-9 show how the program EA is created for table operations and remapping accesses from the data EA. Here, P<23:0> refers to a program space word, whereas D<15:0> refers to a data space word.

TABLE 4-38: 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
		0xxx 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



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

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

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

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

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

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

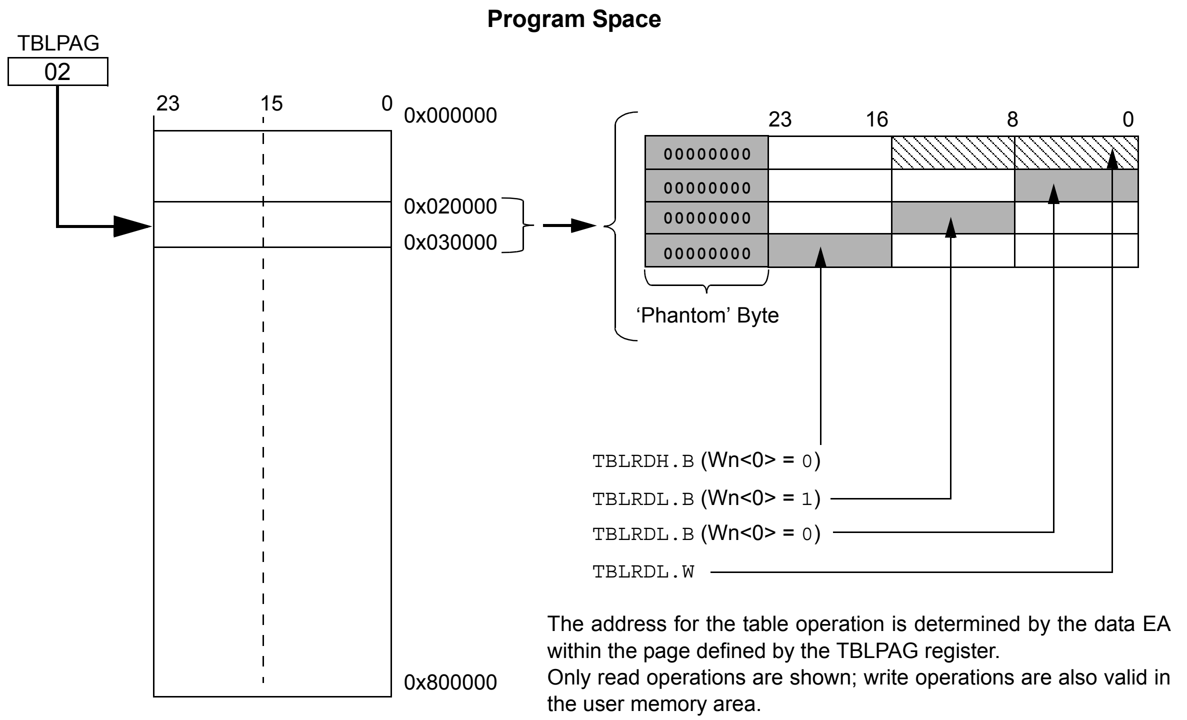
2. `TBLRDH` (Table Read High): In Word mode, it maps the entire upper word of a program address (P<23:16>) to a data address. Note that D<15:8>, the ‘phantom’ byte, will always be ‘`0`’.

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

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

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

FIGURE 4-10: ACCESSING PROGRAM MEMORY WITH TABLE INSTRUCTIONS



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

4.6.3 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 of stored constant data from the data space without the need to use special instructions (i.e., `TBLRDL/H`).

Program space access through the data space occurs if the Most Significant bit of the data space EA is '`1`' and program space visibility is enabled by setting the PSV bit in the 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. Note that by incrementing the PC by 2 for each program memory word, the lower 15 bits of data space addresses directly map to the lower 15 bits in the corresponding program space addresses.

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

Although each data space address, 8000h and higher, maps directly into a corresponding program memory address (see Figure 4-11), 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.

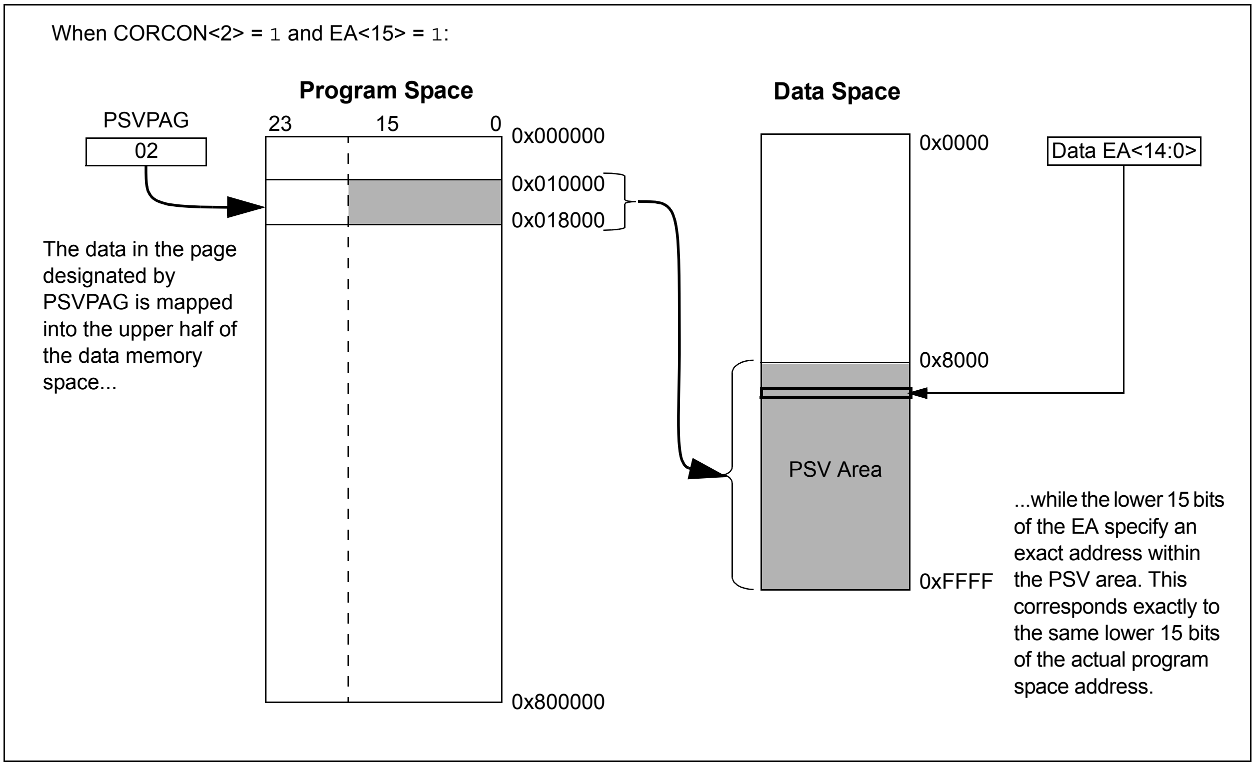
For operations that use PSV and are executed outside a `REPEAT` loop, the `MOV` and `MOV.D` instructions require one instruction cycle in addition to the specified execution time. All other instructions require two instruction cycles in addition to the specified execution time.

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

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

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

FIGURE 4-11: PROGRAM SPACE VISIBILITY OPERATION



NOTES:

5.0 FLASH PROGRAM MEMORY

> **Note:** This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 5. "Flash Programming"** (DS70191) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

The dsPIC33FJXXXMCX06/X08/X10 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:

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

ICSP allows a dsPIC33FJXXXMCX06/X08/X10 device to be serially programmed while in the end application circuit. This is simply 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 can write program memory data by blocks (or 'rows') of 64 instructions (192 bytes) at a time or by single program memory word; and the user can erase program memory in blocks or 'pages' of 512 instructions (1536 bytes) at a time.

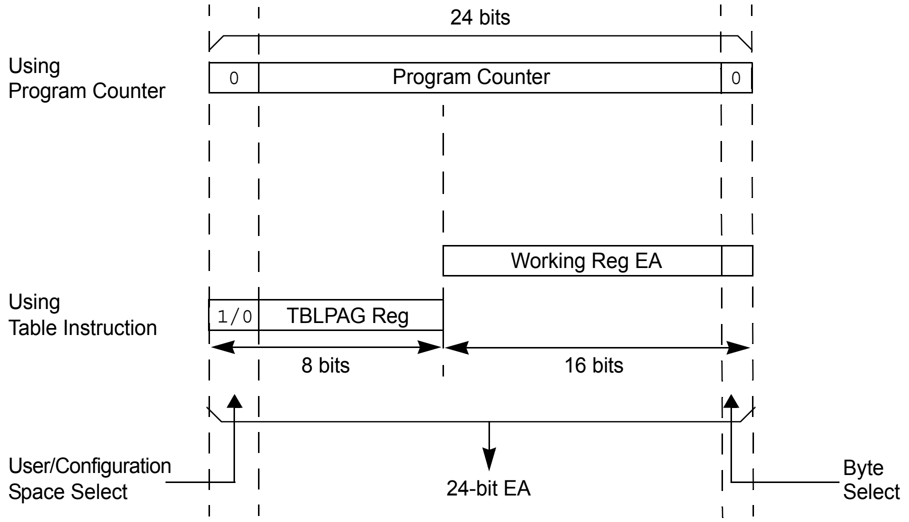
5.1 Table Instructions and Flash Programming

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 `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 dsPIC33FJXXXMCX06/X08/X10 Flash program memory array is organized into rows of 64 instructions or 192 bytes. RTSP allows the user to erase a page of memory at a time, which consists of eight rows (512 instructions), and to program one row or one word at a time. Table 26-12 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 in sequential order. 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 26-19) 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 26-12).

EQUATION 5-1: PROGRAMMING TIME

$$\frac{T}{7.37\ MHz \times (FRC\ Accuracy)\% \times (FRC\ Tuning)\%}$$

For example, if the device is operating at +85°C, the FRC accuracy will be ±2%. If the TUN<5:0> bits (see Register 9-4) are set to `'b111111`, the Minimum Row Write Time is:

$$T_{RW} = \frac{11064\ Cycles}{7.37\ MHz \times (1 + 0.02) \times (1 - 0.00375)} = 1.48ms$$

and, the Maximum Row Write Time is:

$$T_{RW} = \frac{11064\ Cycles}{7.37\ MHz \times (1 - 0.02) \times (1 - 0.00375)} = 1.54ms$$

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

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

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

NVMKEY is a write-only register that is used for write protection. To start a programming or erase sequence, the user 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 = Settable-only bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **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.

5.4.1 PROGRAMMING ALGORITHM FOR FLASH PROGRAM MEMORY

The user 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 as follows:

1. Read eight rows of program memory (512 instructions) and store it 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 must wait for the programming time until programming is complete. The two instructions following the start of the programming sequence should be `NOP`s, 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                               ; Insert two NOPs after the
       NOP                               ; erase command is asserted
```

NOTES:

6.0 RESET

Note: This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 8. "Reset"** (DS70192) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (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
- WDT: Watchdog Timer Reset
- TRAPR: Trap Conflict Reset
- IOPUWR: Illegal Opcode and Uninitialized W Register Reset

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

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

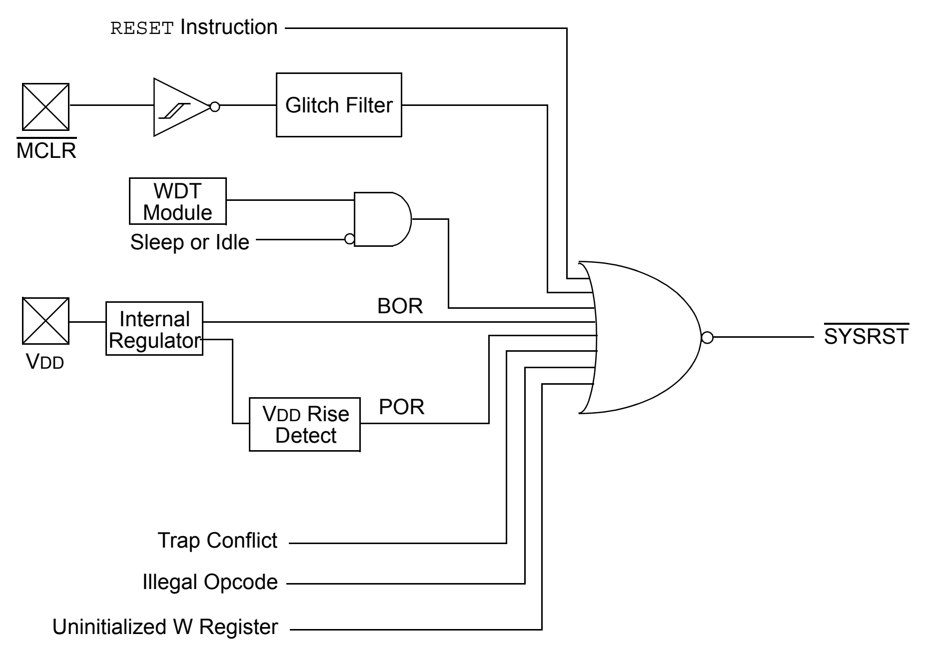
Note: Refer to the specific peripheral or CPU section of this manual for register Reset states.

All types of device Reset will set a corresponding status bit in the RCON register to indicate the type of Reset (see Register 6-1). A POR will clear all bits except for the POR bit (RCON<0>), which is set. The user 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 will be meaningful.

FIGURE 6-1: RESET SYSTEM BLOCK DIAGRAM



REGISTER 6-1: RCON: RESET CONTROL REGISTER[1]

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0
TRAPR	IOPUWR	—	—	—	—	—	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 13-9 **Unimplemented:** Read as ‘0’

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 may be set or cleared in software. Setting one of these bits in software does not cause a device Reset.

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

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

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

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

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

TABLE 6-1: RESET FLAG BIT OPERATION

Flag Bit	Setting Event	Clearing Event
TRAPR (RCON<15>)	Trap conflict event	POR, BOR
IOPUWR (RCON<14>)	Illegal opcode or uninitialized W register access	POR, BOR
EXTR (RCON<7>)	MCLR Reset	POR
SWR (RCON<6>)	`RESET` instruction	POR, BOR
WDTO (RCON<4>)	WDT time-out	`PWRSAV` instruction, POR, BOR
SLEEP (RCON<3>)	`PWRSAV #SLEEP` instruction	POR, BOR
IDLE (RCON<2>)	`PWRSAV #IDLE` instruction	POR, BOR
BOR (RCON<1>)	BOR, POR	—
POR (RCON<0>)	POR	—

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

6.1 Clock Source Selection at Reset

If clock switching is enabled, the system clock source at device Reset is chosen, as shown in Table 6-2. If clock switching is disabled, the system clock source is always selected according to the oscillator Configuration bits. Refer to **Section 9.0 "Oscillator Configuration"** for further details.

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

Reset Type	Clock Source Determinant
POR	Oscillator Configuration bits (FNOSC<2:0>)
BOR	
MCLR	COSC Control bits (OSCCON<14:12>)
WDTR	
SWR	

6.2 Device Reset Times

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

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

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

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

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

Note 1: TPOR = Power-on Reset delay (10 µs nominal).

2: TSTARTUP = Conditional POR delay of 20 µs nominal (if on-chip regulator is enabled) or 64 ms nominal Power-up Timer delay (if regulator is disabled). TSTARTUP is also applied to all returns from powered-down states, including waking from Sleep mode, if the regulator is enabled.

3: TRST = Internal state Reset time (20 µs nominal).

4: TOST = Oscillator Start-up Timer. A 10-bit counter counts 1024 oscillator periods before releasing the oscillator clock to the system.

5: TLOCK = PLL lock time (20 µs nominal).

6: TFSCM = Fail-Safe Clock Monitor delay (100 µs nominal).

6.2.1 POR AND LONG OSCILLATOR START-UP TIMES

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

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

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

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

If the FSCM is enabled, it begins to monitor the system clock source when $\overline{\text{SYSRST}}$ is released. If a valid clock source is not available at this time, the device automatically switches to the FRC oscillator and the user can switch to the desired crystal oscillator in the Trap Service Routine.

6.2.2.1 FSCM Delay for Crystal and PLL Clock Sources

When the system clock source is provided by a crystal oscillator and/or the PLL, a small delay, TFSCM, is automatically inserted after the POR and PWRT delay times. The FSCM does not begin to monitor the system clock source until this delay expires. The FSCM delay time is nominally 500 µs and provides additional time for the oscillator and/or PLL to stabilize. In most cases, the FSCM delay prevents an oscillator failure trap at a device Reset when the PWRT is disabled.

6.3 Special Function Register Reset States

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

The Reset value for each SFR does not depend on the type of Reset, with the exception of two registers. The Reset value for the Reset Control register, RCON, depends on the type of device Reset. The Reset value for the Oscillator Control register, OSCCON, depends on the type of Reset and the programmed values of the oscillator Configuration bits in the FOSC Configuration register.

7.0 INTERRUPT CONTROLLER

> **Note:** This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 6. "Interrupts"** (DS70184) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

The interrupt controller for the dsPIC33FJXXXMCX06/X08/X10 family of devices reduces the numerous peripheral interrupt request signals to a single interrupt request signal to the dsPIC33FJXXXMCX06/X08/X10 CPU. It has the following features:

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

7.1 Interrupt Vector Table

The Interrupt Vector Table (IVT) is shown in Figure 7-1. The IVT resides in program memory, starting at location 000004h. The IVT contains 126 vectors consisting of 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. All other things being equal, lower addresses have a higher natural priority. For example, the interrupt associated with vector 0 will take priority over interrupts at any other vector address.

The dsPIC33FJXXXMCX06/X08/X10 family of devices implement up to 67 unique interrupts and five nonmaskable traps. These are summarized in Table 7-1 and Table 7-2.

7.1.1 ALTERNATE INTERRUPT VECTOR TABLE

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

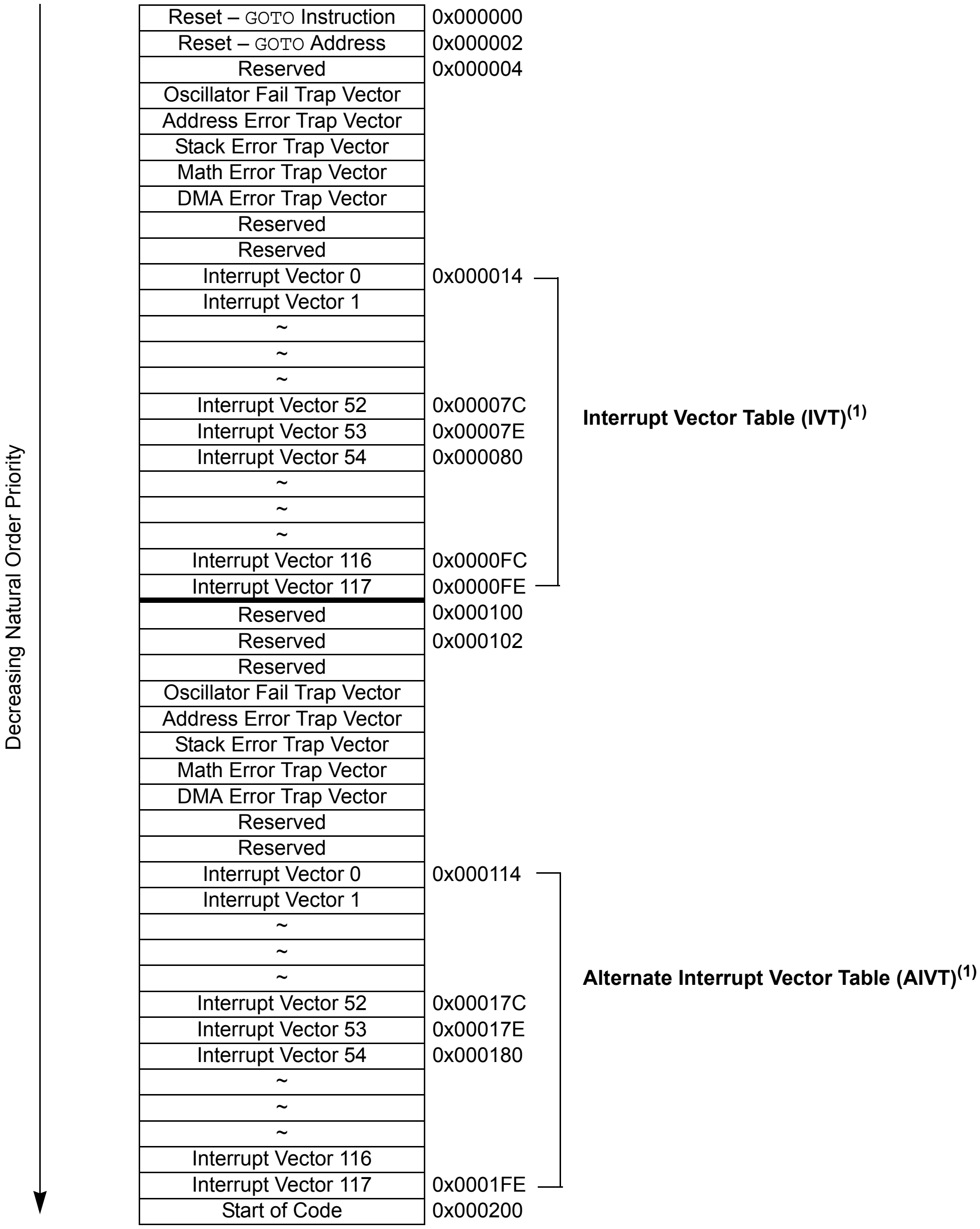
The AIVT supports debugging by providing a means to switch between an application and a support environment without requiring the interrupt vectors to be reprogrammed. This feature also enables switching between applications for evaluation of different software algorithms at run time. If the AIVT is 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 dsPIC33FJXXXMCX06/X08/X10 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. The user programs a `GOTO` instruction at the Reset address, which redirects program execution to the appropriate start-up routine.

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

FIGURE 7-1: dsPIC33FJXXXMCX06/X08/X10 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 1	
~	
~	
~	
Interrupt Vector 52	0x00007C
Interrupt Vector 53	0x00007E
Interrupt Vector 54	0x000080
~	
~	
~	
Interrupt Vector 116	0x0000FC
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

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

TABLE 7-1: INTERRUPT VECTORS

Vector Number	Interrupt Request (IRQ) Number	IVT Address	AIVT Address	Interrupt Source
8	0	0x000014	0x000114	INT0 – External Interrupt 0
9	1	0x000016	0x000116	IC1 – Input Compare 1
10	2	0x000018	0x000118	OC1 – Output Compare 1
11	3	0x00001A	0x00011A	T1 – Timer1
12	4	0x00001C	0x00011C	DMA0 – DMA Channel 0
13	5	0x00001E	0x00011E	IC2 – Input Capture 2
14	6	0x000020	0x000120	OC2 – Output Compare 2
15	7	0x000022	0x000122	T2 – Timer2
16	8	0x000024	0x000124	T3 – Timer3
17	9	0x000026	0x000126	SPI1E – SPI1 Error
18	10	0x000028	0x000128	SPI1 – SPI1 Transfer Done
19	11	0x00002A	0x00012A	U1RX – UART1 Receiver
20	12	0x00002C	0x00012C	U1TX – UART1 Transmitter
21	13	0x00002E	0x00012E	ADC1 – ADC 1
22	14	0x000030	0x000130	DMA1 – DMA Channel 1
23	15	0x000032	0x000132	Reserved
24	16	0x000034	0x000134	SI2C1 – I2C1 Slave Events
25	17	0x000036	0x000136	MI2C1 – I2C1 Master Events
26	18	0x000038	0x000138	Reserved
27	19	0x00003A	0x00013A	Change Notification Interrupt
28	20	0x00003C	0x00013C	INT1 – External Interrupt 1
29	21	0x00003E	0x00013E	ADC2 – ADC 2
30	22	0x000040	0x000140	IC7 – Input Capture 7
31	23	0x000042	0x000142	IC8 – Input Capture 8
32	24	0x000044	0x000144	DMA2 – DMA Channel 2
33	25	0x000046	0x000146	OC3 – Output Compare 3
34	26	0x000048	0x000148	OC4 – Output Compare 4
35	27	0x00004A	0x00014A	T4 – Timer4
36	28	0x00004C	0x00014C	T5 – Timer5
37	29	0x00004E	0x00014E	INT2 – External Interrupt 2
38	30	0x000050	0x000150	U2RX – UART2 Receiver
39	31	0x000052	0x000152	U2TX – UART2 Transmitter
40	32	0x000054	0x000154	SPI2E – SPI2 Error
41	33	0x000056	0x000156	SPI1 – SPI1 Transfer Done
42	34	0x000058	0x000158	C1RX – ECAN1 Receive Data Ready
43	35	0x00005A	0x00015A	C1 – ECAN1 Event
44	36	0x00005C	0x00015C	DMA3 – DMA Channel 3
45	37	0x00005E	0x00015E	IC3 – Input Capture 3
46	38	0x000060	0x000160	IC4 – Input Capture 4
47	39	0x000062	0x000162	IC5 – Input Capture 5
48	40	0x000064	0x000164	IC6 – Input Capture 6
49	41	0x000066	0x000166	OC5 – Output Compare 5
50	42	0x000068	0x000168	OC6 – Output Compare 6
51	43	0x00006A	0x00016A	OC7 – Output Compare 7
52	44	0x00006C	0x00016C	OC8 – Output Compare 8
53	45	0x00006E	0x00016E	Reserved

TABLE 7-1: INTERRUPT VECTORS (CONTINUED)

Vector Number	Interrupt Request (IRQ) Number	IVT Address	AIVT Address	Interrupt Source
54	46	0x000070	0x000170	DMA4 – DMA Channel 4
55	47	0x000072	0x000172	T6 – Timer6
56	48	0x000074	0x000174	T7 – Timer7
57	49	0x000076	0x000176	SI2C2 – I2C2 Slave Events
58	50	0x000078	0x000178	MI2C2 – I2C2 Master Events
59	51	0x00007A	0x00017A	T8 – Timer8
60	52	0x00007C	0x00017C	T9 – Timer9
61	53	0x00007E	0x00017E	INT3 – External Interrupt 3
62	54	0x000080	0x000180	INT4 – External Interrupt 4
63	55	0x000082	0x000182	C2RX – ECAN2 Receive Data Ready
64	56	0x000084	0x000184	C2 – ECAN2 Event
65	57	0x000086	0x000186	PWM – PWM Period Match
66	58	0x000088	0x000188	QEI – Position Counter Compare
69	61	0x00008E	0x00018E	DMA5 – DMA Channel 5
70	62	0x000090	0x000190	Reserved
71	63	0x000092	0x000192	$\overline{\text{FLTA}}$ – MCPWM Fault A
72	64	0x000094	0x000194	$\overline{\text{FLTB}}$ – MCPWM Fault B
73	65	0x000096	0x000196	U1E – UART1 Error
74	66	0x000098	0x000198	U2E – UART2 Error
75	67	0x00009A	0x00019A	Reserved
76	68	0x00009C	0x00019C	DMA6 – DMA Channel 6
77	69	0x00009E	0x00019E	DMA7 – DMA Channel 7
78	70	0x0000A0	0x0001A0	C1TX – ECAN1 Transmit Data Request
79	71	0x0000A2	0x0001A2	C2TX – ECAN2 Transmit Data Request
80-125	72-117	0x0000A4- 0x0000FE	0x0001A4- 0x0001FE	Reserved

TABLE 7-2: TRAP VECTORS

Vector Number	IVT Address	AIVT Address	Trap 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 Trap
6	0x000010	0x000110	Reserved
7	0x000012	0x000112	Reserved

7.3 Interrupt Control and Status Registers

dsPIC33FJXXXMCX06/X08/X10 devices implement a total of 30 registers for the interrupt controller:

- INTCON1
- INTCON2
- IFS0 through IFS4
- IEC0 through IEC4
- IPC0 through IPC17
- INTTREG

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

The 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.

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.

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.

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 INT0IF bit is found in IFS0<0>, the INT0IE bit in IEC0<0> and the INT0IP bits in the first position of IPC0 (IPC0<2:0>).

Although they are not specifically part of the interrupt control hardware, two of the CPU Control registers contain bits that control interrupt functionality. The CPU STATUS register, SR, contains the IPL<2:0> bits (SR<7:5>). These bits indicate the current CPU interrupt priority level. The user 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:	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 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

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	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT4EP	INT3EP	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 13-5	**Unimplemented:** Read as ‘`0`’
bit 4	**INT4EP:** External Interrupt 4 Edge Detect Polarity Select bit `1` = Interrupt on negative edge `0` = Interrupt on positive edge
bit 3	**INT3EP:** External Interrupt 3 Edge Detect Polarity Select bit `1` = Interrupt on negative edge `0` = Interrupt on positive edge
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	DMA01IF	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 Fault Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

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

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

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

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

bit 4 **DMA01IF:** 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	DMA21IF
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
IC8IF	IC7IF	AD2IF	INT1IF	CNIF	—	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	**DMA21IF:** 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	**AD2IF:** ADC2 Conversion Complete Interrupt Flag Status bit `1` = Interrupt request has occurred `0` = Interrupt request has not occurred
bit 4	**INT1IF:** External Interrupt 1 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 3	**CNIF:** Input Change Notification Interrupt Flag Status bit `1` = Interrupt request has occurred `0` = Interrupt request has not occurred
bit 2	**Unimplemented:** Read as ‘`0`’
bit 1	**MI2C1IF:** I2C1 Master Events Interrupt Flag Status bit `1` = Interrupt request has occurred `0` = Interrupt request has not occurred
bit 0	**SI2C1IF:** 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

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T6IF	DMA4IF	—	OC8IF	OC7IF	OC6IF	OC5IF	IC6IF
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
IC5IF	IC4IF	IC3IF	DMA3IF	C1IF	C1RXIF	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 **T6IF:** Timer6 Interrupt Flag Status bit
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

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 **Unimplemented:** Read as ‘`0`’

bit 12 **OC8IF:** Output Compare Channel 8 Interrupt Flag Status bit
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

bit 11 **OC7IF:** Output Compare Channel 7 Interrupt Flag Status bit
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

bit 10 **OC6IF:** Output Compare Channel 6 Interrupt Flag Status bit
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

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

bit 8 **IC6IF:** Input Capture Channel 6 Interrupt Flag Status bit
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

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

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

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

bit 4 **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` = Interrupt request has occurred
`0` = Interrupt request has not occurred

REGISTER 7-7: IFS2: INTERRUPT FLAG STATUS REGISTER 2 (CONTINUED)

bit 2	**C1RXIF:** ECAN1 Receive Data Ready Interrupt Flag Status bit `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

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

R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
FLTAIF	—	DMA5IF	—	—	QEIIF	PWMIF	C2IF
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
C2RXIF	INT4IF	INT3IF	T9IF	T8IF	MI2C2IF	SI2C2IF	T7IF
bit 7							bit 0

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

bit 15 **FLTAIF:** PWM Fault A Interrupt Flag Status bit
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

bit 14 **Unimplemented:** Read as ‘`0`’

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 **QEIIF:** QEI Event Interrupt Flag Status bit
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

bit 9 **PWMIF:** PWM Error Interrupt Flag Status bit
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

bit 8 **C2IF:** ECAN2 Event Interrupt Flag Status bit
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

bit 7 **C2RXIF:** ECAN2 Receive Data Ready Interrupt Flag Status bit
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

bit 6 **INT4IF:** External Interrupt 4 Flag Status bit
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

bit 5 **INT3IF:** External Interrupt 3 Flag Status bit
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

bit 4 **T9IF:** Timer9 Interrupt Flag Status bit
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

bit 3 **T8IF:** Timer8 Interrupt Flag Status bit
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

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

REGISTER 7-8: IFS3: INTERRUPT FLAG STATUS REGISTER 3 (CONTINUED)

bit 1	**SI2C2IF:** I2C2 Slave Events Interrupt Flag Status bit `1` = Interrupt request has occurred `0` = Interrupt request has not occurred
bit 0	**T7IF:** Timer7 Interrupt Flag Status bit `1` = Interrupt request has occurred `0` = Interrupt request has not occurred

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

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	U-0	R/W-0	R/W-0	R/W-0
C2TXIF	C1TXIF	DMA7IF	DMA6IF	—	U2EIF	U1EIF	FLTBIF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	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	**C2TXIF:** ECAN2 Transmit Data Request Interrupt Flag Status bit 1 = Interrupt request has occurred 0 = Interrupt request has not occurred
bit 6	**C1TXIF:** ECAN1 Transmit Data Request Interrupt Flag Status bit 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	**Unimplemented:** Read as ‘0’
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	**FLTBIF:** PWM Fault B Interrupt Flag Status bit 1 = Interrupt request has occurred 0 = Interrupt request has not occurred

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 Enable bit `1` = Interrupt request enabled `0` = Interrupt request not enabled

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	DMA2IE
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
IC8IE	IC7IE	AD2IE	INT1IE	CNIE	—	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 **AD2IE:** ADC2 Conversion Complete Interrupt Enable bit
`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 4 **INT1IE:** External Interrupt 1 Enable bit
`1` = Interrupt request enabled
`0` = Interrupt request not enabled

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

bit 3	**CNIE:** Input Change Notification Interrupt Enable bit `1` = Interrupt request enabled `0` = Interrupt request not enabled
bit 2	**Unimplemented:** Read as ‘`0`’
bit 1	**MI2C1IE:** I2C1 Master Events Interrupt Enable bit `1` = Interrupt request enabled `0` = Interrupt request not enabled
bit 0	**SI2C1IE:** I2C1 Slave Events Interrupt Enable bit `1` = Interrupt request enabled `0` = Interrupt request not enabled

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

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T6IE	DMA4IE	—	OC8IE	OC7IE	OC6IE	OC5IE	IC6IE
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
IC5IE	IC4IE	IC3IE	DMA3IE	C1IE	C1RXIE	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 **T6IE:** Timer6 Interrupt Enable bit

`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 14 **DMA4IE:** DMA Channel 4 Data Transfer Complete Interrupt Enable bit

`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 13 **Unimplemented:** Read as ‘`0`’

bit 12 **OC8IE:** Output Compare Channel 8 Interrupt Enable bit

`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 11 **OC7IE:** Output Compare Channel 7 Interrupt Enable bit

`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 10 **OC6IE:** Output Compare Channel 6 Interrupt Enable bit

`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 9 **OC5IE:** Output Compare Channel 5 Interrupt Enable bit

`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 8 **IC6IE:** Input Capture Channel 6 Interrupt Enable bit

`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 7 **IC5IE:** Input Capture Channel 5 Interrupt Enable bit

`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 6 **IC4IE:** Input Capture Channel 4 Interrupt Enable bit

`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 5 **IC3IE:** Input Capture Channel 3 Interrupt Enable bit

`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 4 **DMA3IE:** DMA Channel 3 Data Transfer Complete Interrupt Enable bit

`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 3 **C1IE:** ECAN1 Event Interrupt Enable bit

`1` = Interrupt request enabled
`0` = Interrupt request not enabled

REGISTER 7-12: IEC2: INTERRUPT ENABLE CONTROL REGISTER 2 (CONTINUED)

bit 2 **C1RXIE:** ECAN1 Receive Data Ready Interrupt Enable bit
`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

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

R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
FLTAIE	—	DMA5IE	—	—	QEIIE	PWMIE	C2IE
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
C2RXIE	INT4IE	INT3IE	T9IE	T8IE	MI2C2IE	SI2C2IE	T7IE
bit 7							bit 0

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

bit 15 **FLTAIE:** PWM Fault A Interrupt Enable bit
`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 14 **Unimplemented:** Read as ‘`0`’

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 **QEIIE:** QEI Event Interrupt Enable bit
`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 9 **PWMIE:** PWM Error Interrupt Enable bit
`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 8 **C2IE:** ECAN2 Event Interrupt Enable bit
`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 7 **C2RXIE:** ECAN2 Receive Data Ready Interrupt Enable bit
`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 6 **INT4IE:** External Interrupt 4 Enable bit
`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 5 **INT3IE:** External Interrupt 3 Enable bit
`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 4 **T9IE:** Timer9 Interrupt Enable bit
`1` = Interrupt request enabled
`0` = Interrupt request not enabled

bit 3 **T8IE:** Timer8 Interrupt Enable bit
`1` = Interrupt request enabled
`0` = Interrupt request not enabled

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

REGISTER 7-13: IEC3: INTERRUPT ENABLE CONTROL REGISTER 3 (CONTINUED)

bit 1	**SI2C2IE:** I2C2 Slave Events Interrupt Enable bit `1` = Interrupt request enabled `0` = Interrupt request not enabled
bit 0	**T7IE:** Timer7 Interrupt Enable bit `1` = Interrupt request enabled `0` = Interrupt request not enabled

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

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	U-0	R/W-0	R/W-0	R/W-0
C2TXIE	C1TXIE	DMA7IE	DMA6IE	—	U2EIE	U1EIE	FLTBIE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	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	**C2TXIE:** ECAN2 Transmit Data Request Interrupt Enable bit `1` = Interrupt request enabled `0` = Interrupt request not enabled
bit 6	**C1TXIE:** ECAN1 Transmit Data Request Interrupt Enable bit `1` = Interrupt request enabled `0` = Interrupt request not enabled
bit 5	**DMA7IE:** DMA Channel 7 Data Transfer Complete Enable Status bit `1` = Interrupt request enabled `0` = Interrupt request not enabled
bit 4	**DMA6IE:** DMA Channel 6 Data Transfer Complete Enable Status bit `1` = Interrupt request enabled `0` = Interrupt request not enabled
bit 3	**Unimplemented:** Read as ‘`0`’
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	**FLTBIE:** PWM Fault B Interrupt Enable bit `1` = Interrupt request enabled `0` = Interrupt request not enabled

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	U-0	U-0	U-0
—	CNIP<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 11-7 **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	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	AD2IP<2:0>			—	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 **Unimplemented:** Read as ‘`0`’

bit 6-4 **AD2IP<2:0>:** ADC2 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 **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>			—	C1RXIP<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
—	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

`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

`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

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

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC5IP<2:0>			—	IC4IP<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
—	IC3IP<2:0>			—	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	**Unimplemented:** Read as ‘`0`’
bit 14-12	**IC5IP<2:0>:** Input Capture Channel 5 Interrupt Priority bits `111` = Interrupt is priority 7 (highest priority interrupt) • • • `001` = Interrupt is priority 1 `000` = Interrupt source is disabled
bit 11	**Unimplemented:** Read as ‘`0`’
bit 10-8	**IC4IP<2:0>:** Input Capture Channel 4 Interrupt Priority bits `111` = Interrupt is priority 7 (highest priority interrupt) • • • `001` = Interrupt is priority 1 `000` = Interrupt source is disabled
bit 7	**Unimplemented:** Read as ‘`0`’
bit 6-4	**IC3IP<2:0>:** Input Capture Channel 3 Interrupt Priority bits `111` = Interrupt is priority 7 (highest priority interrupt) • • • `001` = Interrupt is priority 1 `000` = Interrupt source is disabled
bit 3	**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: IPC10: INTERRUPT PRIORITY CONTROL REGISTER 10

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	OC7IP<2:0>			—	OC6IP<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
—	OC5IP<2:0>			—	IC6IP<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 **OC7IP<2:0>:** Output Compare Channel 7 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 **OC6IP<2:0>:** Output Compare Channel 6 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 **OC5IP<2:0>:** Output Compare Channel 5 Interrupt Priority bits

`111` = Interrupt is priority 7 (highest priority interrupt)
•
•
•
`001` = Interrupt is priority 1
`000` = Interrupt source is disabled

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

bit 2-0 **IC6IP<2:0>:** Input Capture Channel 6 Interrupt Priority bits

`111` = Interrupt is priority 7 (highest priority interrupt)
•
•
•
`001` = Interrupt is priority 1
`000` = Interrupt source is disabled

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

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T6IP<2:0>			—	DMA4IP<2:0>		
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	OC8IP<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 **T6IP<2:0>:** Timer6 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 **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 7-3 **Unimplemented:** Read as ‘`0`’

bit 2-0 **OC8IP<2:0>:** Output Compare Channel 8 Interrupt Priority bits

`111` = Interrupt is priority 7 (highest priority interrupt)
•
•
•
`001` = Interrupt is priority 1
`000` = Interrupt source is disabled

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

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T8IP<2:0>			—	MI2C2IP<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
—	SI2C2IP<2:0>			—	T7IP<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	**T8IP<2:0>:** Timer8 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	**MI2C2IP<2:0>:** I2C2 Master Events Interrupt Priority bits `111` = Interrupt is priority 7 (highest priority interrupt) • • • `001` = Interrupt is priority 1 `000` = Interrupt source is disabled
bit 7	**Unimplemented:** Read as ‘0’
bit 6-4	**SI2C2IP<2:0>:** I2C2 Slave 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	**T7IP<2:0>:** Timer7 Interrupt Priority bits `111` = Interrupt is priority 7 (highest priority interrupt) • • • `001` = Interrupt is priority 1 `000` = Interrupt source is disabled

REGISTER 7-28: IPC13: INTERRUPT PRIORITY CONTROL REGISTER 13

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	C2RXIP<2:0>			—	INT4IP<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
—	INT3IP<2:0>			—	T9IP<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 **C2RXIP<2:0>:** ECAN2 Receive Data Ready 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 **INT4IP<2:0>:** External Interrupt 4 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 **INT3IP<2:0>:** External Interrupt 3 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 **T9IP<2:0>:** Timer9 Interrupt Priority bits

`111` = Interrupt is priority 7 (highest priority interrupt)
•
•
•
`001` = Interrupt is priority 1
`000` = Interrupt source is disabled

REGISTER 7-29: 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
—	—	—	—	—	QEIIP<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
—	PWMIP<2:0>			—	C2IP<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 **QEIIP<2:0>:** QEI 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 **PWMIP<2:0>:** PWM 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 **C2IP<2:0>:** ECAN2 Event Interrupt Priority bits

`111` = Interrupt is priority 7 (highest priority interrupt)
•
•
•
`001` = Interrupt is priority 1
`000` = Interrupt source is disabled

REGISTER 7-30: IPC15: INTERRUPT PRIORITY CONTROL REGISTER 15

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	FLTAIP<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 **FLTAIP<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 11-7 **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 7-31: IPC16: INTERRUPT PRIORITY CONTROL REGISTER 16

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	U2EIP<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
—	U1EIP<2:0>			—	FLTBIP<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 **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 **Unimplemented:** Read as ‘`0`’

bit 2-0 **FLTBIP<2:0>:** PWM Fault B Interrupt Priority bits

`111` = Interrupt is priority 7 (highest priority interrupt)
•
•
•
`001` = Interrupt is priority 1
`000` = Interrupt source is disabled

REGISTER 7-32: IPC17: INTERRUPT PRIORITY CONTROL REGISTER 17

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	C2TXIP<2:0>			—	C1TXIP<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
—	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 **Unimplemented:** Read as ‘`0`’

bit 14-12 **C2TXIP<2:0>:** ECAN2 Transmit Data Request 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 **C1TXIP<2:0>:** ECAN1 Transmit Data Request 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 **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

REGISTER 7-33: 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, do the following:

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 will depend on the specific application and type of interrupt source. If multiple priority levels are not desired, the IPCx register control bits for all enabled interrupt sources may be programmed to the same non-zero value.

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

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

7.4.2 INTERRUPT SERVICE ROUTINE

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

7.4.3 TRAP SERVICE ROUTINE

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

7.4.4 INTERRUPT DISABLE

All user interrupts can be disabled using the following procedure:

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

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

Note that only user interrupts with a priority level of 7 or less can be disabled. Trap sources (level 8-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.

NOTES:

8.0 DIRECT MEMORY ACCESS (DMA)

Note: This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 22. "Direct Memory Access (DMA)"** (DS70182) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

Direct Memory Access (DMA) is a very efficient mechanism of copying data between peripheral SFRs (e.g., the UART Receive register and 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 dsPIC33FJXXXMCX06/X08/X10 peripherals that can utilize DMA are listed in Table 8-1 along with their associated Interrupt Request (IRQ) numbers.

TABLE 8-1: PERIPHERALS WITH DMA SUPPORT

Peripheral	IRQ Number
INT0	0
Input Capture 1	1
Input Capture 2	5
Output Compare 1	2
Output Compare 2	6
Timer2	7
Timer3	8
SPI1	10
SPI2	33
UART1 Reception	11
UART1 Transmission	12
UART2 Reception	30
UART2 Transmission	31
ADC1	13
ADC2	21
ECAN1 Reception	34
ECAN1 Transmission	70
ECAN2 Reception	55
ECAN2 Transmission	71

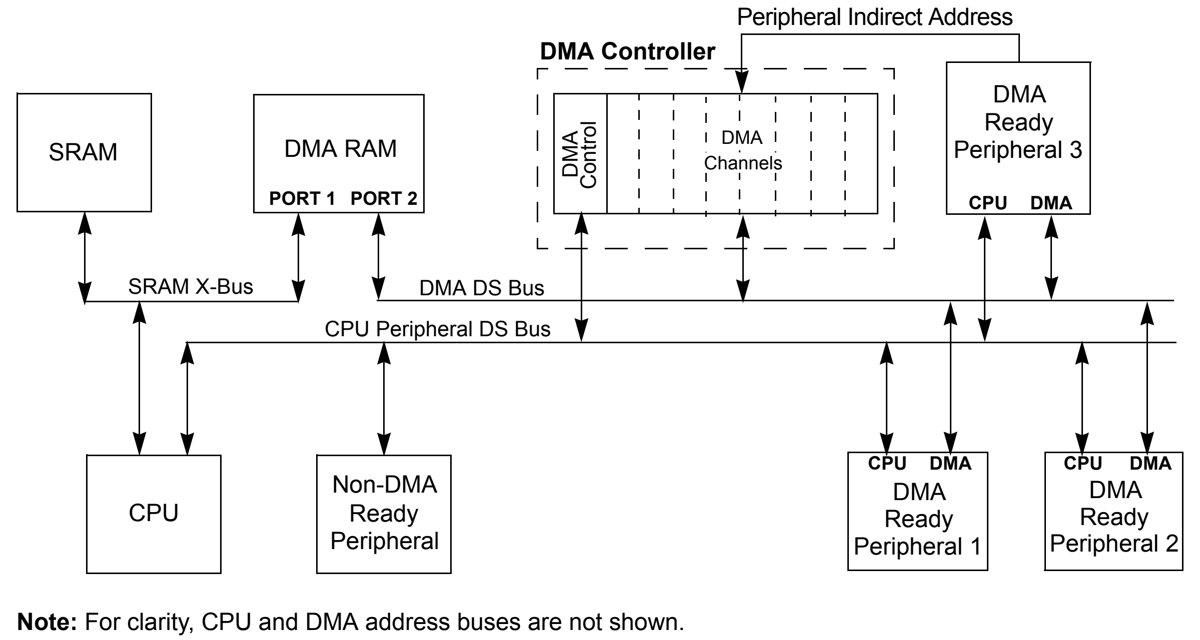
The DMA controller features eight identical data transfer channels. Each channel has its own set of control and status registers. Each DMA channel can be configured to copy data either from buffers stored in dual port DMA RAM to peripheral SFRs or from peripheral SFRs to buffers in DMA RAM.

The DMA controller supports the following features:

- Word or byte sized data transfers.
- Transfers from peripheral to DMA RAM or DMA RAM to peripheral.
- Indirect Addressing of DMA RAM locations with or without automatic post-increment.
- Peripheral Indirect Addressing – In some peripherals, the DMA RAM read/write addresses may be partially derived from the peripheral.
- One-Shot Block Transfers – Terminating DMA transfer after one block transfer.
- Continuous Block Transfers – Reloading DMA RAM buffer start address after every block transfer is complete.
- Ping-Pong Mode – Switching between two DMA RAM start addresses between successive block transfers, thereby filling two buffers alternately.
- Automatic or manual initiation of block transfers.
- Each channel can select from 20 possible sources of data sources or destinations.

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



Note: For clarity, CPU and DMA address buses are not shown.

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 Offset register (DMAxSTA)
- A 16-bit DMA RAM Secondary Start Address Offset 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.

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 `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	R/W-0	R/W-0	R/W-0	R/W-0
—	IRQSEL6[2]	IRQSEL5[2]	IRQSEL4[2]	IRQSEL3[2]	IRQSEL2[2]	IRQSEL1[2]	IRQSEL0[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: See Table 8-1 for a complete listing of IRQ numbers for all interrupt sources.

REGISTER 8-3: DMAxSTA: DMA CHANNEL x RAM START ADDRESS OFFSET REGISTER A

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)

REGISTER 8-4: DMAxSTB: DMA CHANNEL x RAM START ADDRESS OFFSET REGISTER B

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)

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

bit 15 **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

9.0 OSCILLATOR CONFIGURATION

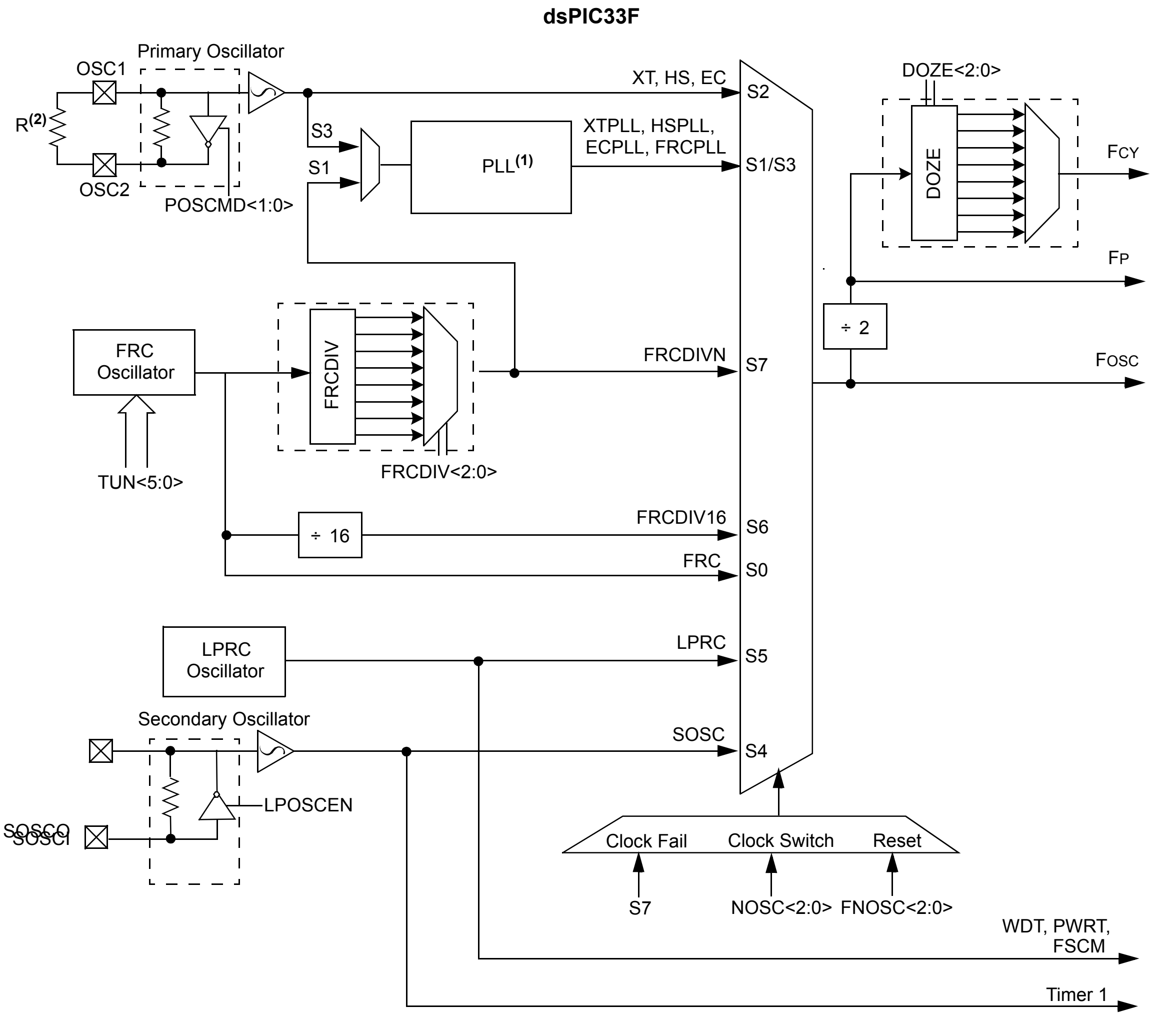
Note:	This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 7. "Oscillator"** (DS70186) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

The dsPIC33FJXXXMCX06/X08/X10 oscillator system provides the following:

- Various external and internal oscillator options as clock sources
- An on-chip PLL to scale the internal operating frequency to the required system clock frequency
- The internal FRC oscillator 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

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

FIGURE 9-1: dsPIC33FJXXXMCX06/X08/X10 OSCILLATOR SYSTEM DIAGRAM



Note 1: See Figure 9-2 for PLL details.

2: If the Oscillator is used with XT or HS modes, an extended parallel resistor with the value of 1 MΩ must be connected.

9.1 CPU Clocking System

There are seven system clock options provided by the dsPIC33FJXXXMCX06/X08/X10:

- FRC Oscillator
- FRC Oscillator with PLL
- Primary (XT, HS or EC) Oscillator
- Primary Oscillator with PLL
- Secondary (LP) Oscillator
- LPRC Oscillator
- FRC Oscillator with postscaler

9.1.1 SYSTEM CLOCK SOURCES

The FRC (Fast RC) internal oscillator runs at a nominal frequency of 7.37 MHz. The 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:

1. XT (Crystal): Crystals and ceramic resonators in the range of 3 MHz to 10 MHz. The crystal is connected to the OSC1 and OSC2 pins.
2. HS (High-Speed Crystal): Crystals in the range of 10 MHz to 40 MHz. The crystal is connected to the OSC1 and OSC2 pins.
3. EC (External Clock): 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 SOSCI and SOSCO pins.

The LPRC (Low-Power RC) 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 Phase Locked Loop (PLL) to provide a wide range of output frequencies for device operation. PLL configuration is described in **Section 9.1.3 "PLL Configuration"**.

The FRC frequency depends on the FRC accuracy (see Table 26-19) and the value of the FRC Oscillator Tuning register (see Register 9-4).

9.1.2 SYSTEM CLOCK SELECTION

The oscillator source that is used at a device Power-on Reset event is selected using Configuration bit settings. The oscillator Configuration bit settings are located in the Configuration registers in the program memory. (Refer to **Section 23.1 "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 between twelve 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 dsPIC33FJXXXMCX06/X08/X10 architecture.

Instruction execution speed or device operating frequency, FCY, is given by the following equation:

EQUATION 9-1: DEVICE OPERATING FREQUENCY

$$F_{CY} = \frac{F_{OSC}}{2}$$

9.1.3 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 a significant amount of 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 to be in the range of 0.8 MHz to 8 MHz. Since the minimum prescale factor is 2, this implies that FIN must be chosen to be in the range of 1.6 MHz to 16 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 the following equation:

EQUATION 9-2: FOSC CALCULATION

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

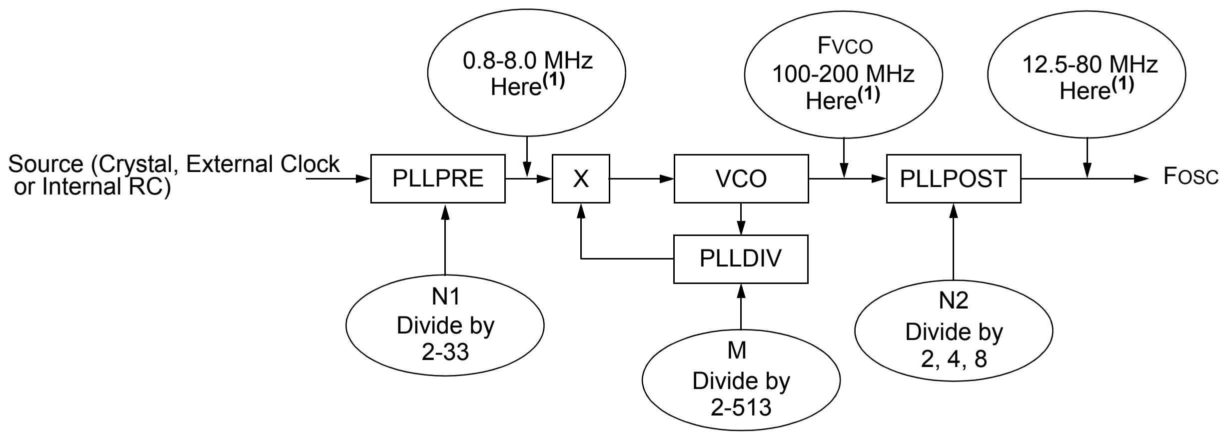
For example, suppose a 10 MHz crystal is being used with “XT with PLL” as the selected oscillator mode. If PLLPRE<4:0> = 0, then N1 = 2. This yields a VCO input of 10/2 = 5 MHz, which is within the acceptable range of 0.8-8 MHz. If PLLDIV<8:0> = 0x1E, then M = 32. This yields a VCO output of 5 * 32 = 160 MHz, which is within the 100-200 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: dsPIC33FJXXXMCX06/X08/X10 PLL BLOCK DIAGRAM



Note 1: This frequency range must be satisfied at all times.

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	U-0	R-0	U-0	R/C-0	U-0	R/W-0	R/W-0
CLKLOCK	—	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

`1` = If (FCKSM0 = `1`), then clock and PLL configurations are locked
If (FCKSM0 = `0`), then clock and PLL configurations may be modified
`0` = Clock and PLL selections are not locked; configurations may be modified

bit 6 **Unimplemented:** Read as ‘`0`’

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

bit 2 **Unimplemented:** Read as ‘`0`’

Note 1: Writes to this register require an unlock sequence. Refer to **Section 7. “Oscillator”** (DS70186) 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 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 7. “Oscillator”** (DS70186) 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 will clear 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.

9.2 Clock Switching Operation

Applications are free to switch between any of the four clock sources (Primary, LP, FRC and LPRC) under software control at any time. To limit the possible side effects that could result from this flexibility, dsPIC33FJXXXMCX06/X08/X10 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 between 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 23.1 "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

At a minimum, performing a clock switch requires the following 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 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, then the clock switch is a redundant operation. In this case, the OSWEN bit is cleared automatically and the clock switch is aborted.
2. If a valid clock switch has been initiated, the LOCK (OSCCON<5>) and 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 **7. "Oscillator"** (DS70186) 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 dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 9. "Watchdog Timer and Power-Saving Modes"** (DS70196) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

The dsPIC33FJXXXMCX06/X08/X10 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. dsPIC33FJXXXMCX06/X08/X10 devices can manage power consumption in four different ways:

- Clock frequency
- Instruction-based Sleep and Idle modes
- Software-controlled Doze mode
- Selective peripheral control in software

Combinations of these methods can be used to selectively tailor an application's power consumption while still maintaining critical application features, such as timing-sensitive communications.

10.1 Clock Frequency and Clock Switching

dsPIC33FJXXXMCX06/X08/X10 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"**.

10.2 Instruction-Based Power-Saving Modes

dsPIC33FJXXXMCX06/X08/X10 devices have two special power-saving modes that are entered through the execution of a special `PWRSAV` instruction. Sleep mode stops clock operation and halts all code execution. Idle mode halts the CPU and code execution, but allows peripheral modules to continue operation. The assembly syntax of the `PWRSAV` instruction is shown in Example 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

Sleep mode has the following features:

- 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 during Sleep mode 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 may continue to operate in Sleep mode. This includes items such as the input change notification on the I/O ports and peripherals that use an external clock input. Any peripheral that requires the system clock source for its operation is disabled in Sleep mode.

The device will wake-up from Sleep mode on any of the following events:

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

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

EXAMPLE 10-1: `PWRSAV` INSTRUCTION SYNTAX

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

10.2.2 IDLE MODE

Idle mode has the following features:

- 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 will wake from Idle mode on any of the following events:

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

On wake-up from Idle, the clock is reapplied to the CPU and instruction execution begins immediately, starting with the instruction following the `PWRSAV` instruction or the first instruction in the ISR.

10.2.3 INTERRUPTS COINCIDENT WITH POWER SAVE INSTRUCTIONS

Any interrupt that coincides with the execution of a `PWRSAV` 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

Generally, changing clock speed and invoking one of the power-saving modes are the preferred strategies for reducing power consumption. There may be circumstances, however, where this is not practical. For example, it may be necessary for an application to maintain uninterrupted synchronous communication, even while it is doing nothing else. Reducing system clock speed may introduce communication errors, while using a power-saving mode may stop communications completely.

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

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

It is also possible to use Doze mode to selectively reduce power consumption in event-driven applications. This allows clock-sensitive functions, such as synchronous communications, to continue without interruption while the CPU idles, waiting for something to invoke an interrupt routine. Enabling the automatic return to full-speed CPU operation on interrupts is enabled by setting the ROI bit (CLKDIV<15>). By default, interrupt events have no effect on Doze mode operation.

For example, suppose the device is operating at 20 MIPS and the CAN module has been configured for 500 kbps based on this device operating speed. If the device is now placed in Doze mode with a clock frequency ratio of 1:4, the CAN 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 via 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 will have no effect and read values will be invalid.

A peripheral module is only enabled 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 1 instruction cycle. Similarly, if a PMD bit is cleared, the corresponding module is enabled after a delay of 1 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	QEI1MD	PWMMD	—
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
I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	C2MD	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 **QEI1MD:** QEI1 Module Disable bit
`1` = QEI1 module is disabled
`0` = QEI1 module is enabled

bit 9 **PWMMD:** PWM Module Disable bit
`1` = PWM module is disabled
`0` = PWM module is enabled

bit 8 **Unimplemented:** Read as ‘`0`’

bit 7 **I2C1MD:** I^{2}C1 Module Disable bit
`1` = I^{2}C1 module is disabled
`0` = I^{2}C1 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

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

bit 2	**C2MD:** ECAN2 Module Disable bit `1` = ECAN2 module is disabled `0` = ECAN2 module is enabled
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	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IC8MD	IC7MD	IC6MD	IC5MD	IC4MD	IC3MD	IC2MD	IC1MD
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
OC8MD	OC7MD	OC6MD	OC5MD	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 7 Module Disable bit `1` = Input Capture 7 module is disabled `0` = Input Capture 7 module is enabled
bit 13	**IC6MD:** Input Capture 6 Module Disable bit `1` = Input Capture 6 module is disabled `0` = Input Capture 6 module is enabled
bit 12	**IC5MD:** Input Capture 5 Module Disable bit `1` = Input Capture 5 module is disabled `0` = Input Capture 5 module is enabled
bit 11	**IC4MD:** Input Capture 4 Module Disable bit `1` = Input Capture 4 module is disabled `0` = Input Capture 4 module is enabled
bit 10	**IC3MD:** Input Capture 3 Module Disable bit `1` = Input Capture 3 module is disabled `0` = Input Capture 3 module is enabled
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	**OC8MD:** Output Compare 8 Module Disable bit `1` = Output Compare 8 module is disabled `0` = Output Compare 8 module is enabled
bit 6	**OC7MD:** Output Compare 4 Module Disable bit `1` = Output Compare 7 module is disabled `0` = Output Compare 7 module is enabled
bit 5	**OC6MD:** Output Compare 6 Module Disable bit `1` = Output Compare 6 module is disabled `0` = Output Compare 6 module is enabled
bit 4	**OC5MD:** Output Compare 5 Module Disable bit `1` = Output Compare 5 module is disabled `0` = Output Compare 5 module is enabled

REGISTER 10-2: PMD2: PERIPHERAL MODULE DISABLE CONTROL REGISTER 2 (CONTINUED)

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

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
T9MD	T8MD	T7MD	T6MD	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	I2C2MD	AD2MD
bit 7							bit 0

Legend:

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

bit 15	**T9MD:** Timer9 Module Disable bit `1` = Timer9 module is disabled `0` = Timer9 module is enabled
bit 14	**T8MD:** Timer8 Module Disable bit `1` = Timer8 module is disabled `0` = Timer8 module is enabled
bit 13	**T7MD:** Timer7 Module Disable bit `1` = Timer7 module is disabled `0` = Timer7 module is enabled
bit 12	**T6MD:** Timer6 Module Disable bit `1` = Timer6 module is disabled `0` = Timer6 module is enabled
bit 11-2	**Unimplemented:** Read as ‘`0`’
bit 1	**I2C2MD:** I2C2 Module Disable bit `1` = I2C2 module is disabled `0` = I2C2 module is enabled
bit 0	**AD2MD:** AD2 Module Disable bit `1` = AD2 module is disabled `0` = AD2 module is enabled

NOTES:

11.0 I/O PORTS

Note:	This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 10. "I/O Ports"** (DS70193) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

All of the device pins (except VDD, VSS, MCLR and OSC1/CLKIN) are shared between 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

A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral. The peripheral's output buffer data and control signals are provided to a pair of multiplexers. The multiplexers select whether the peripheral or the associated port has ownership of the output data and control signals of the I/O pin. The logic also prevents "loop through," in which a port's digital output can drive the input of a peripheral that shares the same pin. Figure 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 actively driving an associated pin, the use of the pin as a general purpose output pin is disabled. The I/O pin may be read, but the output driver for the parallel port bit will be disabled. If a peripheral is enabled but the peripheral is not actively driving a pin, that pin may be driven by a port.

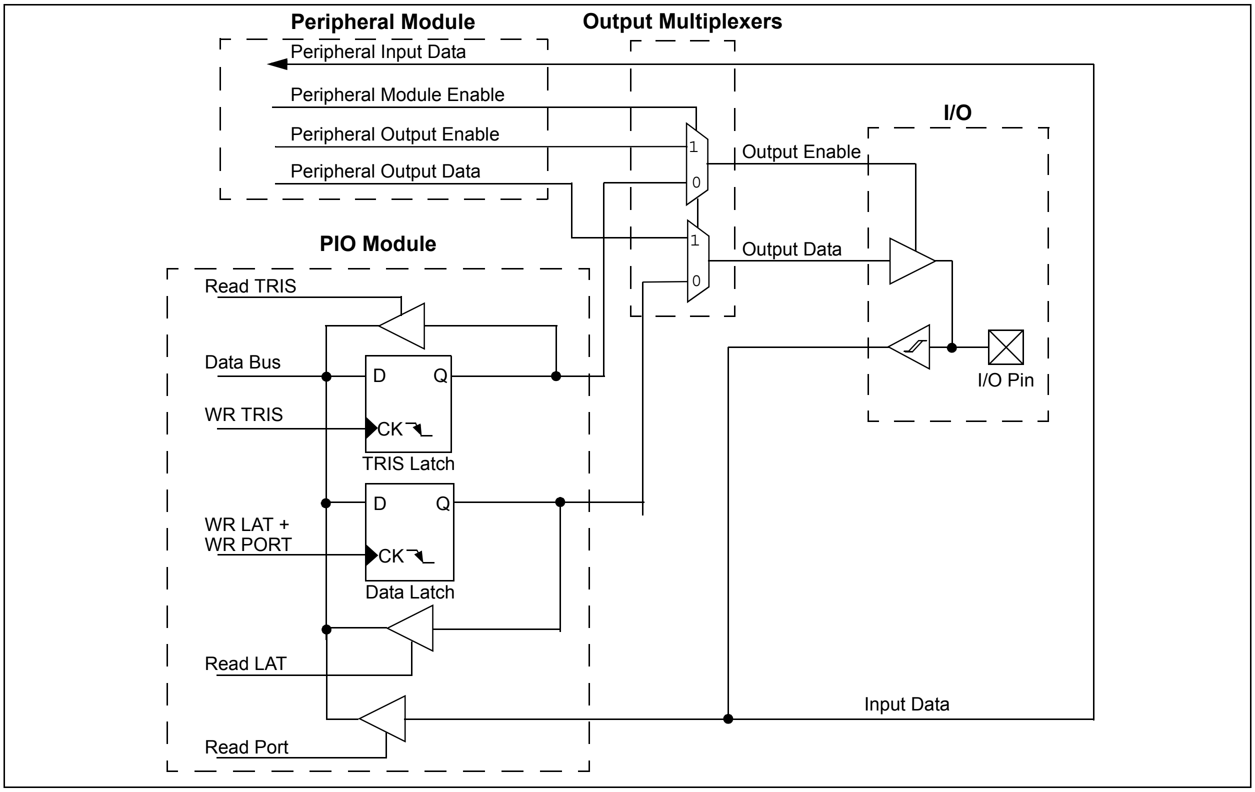
All port pins have three registers directly associated with their operation as digital I/O. The data direction register (TRISx) determines whether the pin is an input or an output. If the data direction bit is a `‘1’`, then the pin is an input. All port pins are defined as inputs after a Reset. Reads from the 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 will be disabled. That means the corresponding LATx and TRISx registers and the port pins will read as zeros.

When a pin is shared with another peripheral or function that is defined as an input only, it is nevertheless regarded as a dedicated port because there is no other competing source of outputs. An example is the INT4 pin.

Note:	The voltage on a digital input pin can be between -0.3V to 5.6V.

FIGURE 11-1: BLOCK DIAGRAM OF A TYPICAL SHARED PORT STRUCTURE



11.2 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 VDD (e.g., 5V) on any desired digital-only pins by using external pull-up resistors. The maximum open-drain voltage allowed is the same as the maximum VIH specification.

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

11.3 Configuring Analog Port Pins

The ADxPCFGH, ADxPCFGL and TRIS registers control the operation of the ADC port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bit set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) is converted.

Clearing any bit in the ADxPCFGH or ADxPCFGL register configures the corresponding bit to be an analog pin. This is also the Reset state of any I/O pin that has an analog (ANx) function associated with it.

Note: In devices with two ADC modules, if the corresponding PCFG bit in either AD1PCFGH(L) and AD2PCFGH(L) is cleared, the pin is configured as an analog input.

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

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

Note: The voltage on an analog input pin can be between -0.3V to (VDD + 0.3 V).

11.4 I/O Port Write/Read Timing

One instruction cycle is required between a port direction change or port write operation and a read operation of the same port. Typically, this instruction would be a `NOP`.

11.5 Input Change Notification

The input change notification function of the I/O ports allows the dsPIC33FJXXXMCX06/X08/X10 devices to generate interrupt requests to the processor in response to a change-of-state on selected input pins. This feature is capable of detecting input change-of-states even in Sleep mode, when the clocks are disabled. Depending on the device pin count, there are up to 24 external signals (CN0 through CN23) that can be selected (enabled) for generating an interrupt request on a change-of-state.

There are four control registers associated with the CN module. The CNEN1 and CNEN2 registers contain the CN interrupt enable (CNxIE) control bits for each of the CN input pins. Setting any of these bits enables a CN interrupt for the corresponding pins.

Each CN pin also has a weak pull-up connected to it. The pull-ups act as a current source that is connected to the pin and eliminate the need for external resistors when push button or keypad devices are connected. The pull-ups are enabled separately using the CNPU1 and CNPU2 registers, which contain the weak pull-up enable (CNxPUE) 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 whenever the port pin is configured as a digital output.

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
```

12.0 TIMER1

Note:	This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 11. "Timers"** (DS70205) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (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. Timer1 can operate in three modes:

- 16-bit Timer
- 16-bit Synchronous Counter
- 16-bit Asynchronous Counter

Timer1 also supports the following features:

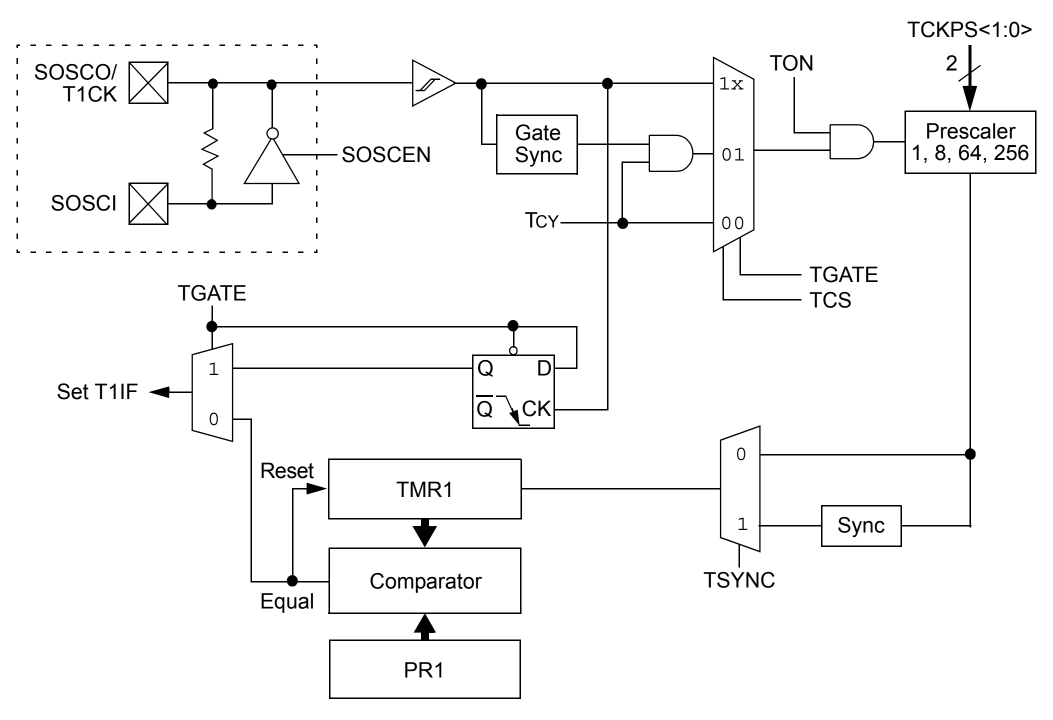
- Timer gate operation
- Selectable prescaler settings
- Timer operation during CPU Idle and Sleep modes
- Interrupt on 16-bit Period register match or falling edge of external gate signal

Figure 12-1 presents a block diagram of the 16-bit timer module.

To configure Timer1 for operation, do the following:

1. Set the TON bit (= `1`) in the T1CON register.
2. Select the timer prescaler ratio using the TCKPS<1:0> bits in the T1CON register.
3. Set the Clock and Gating modes using the TCS and TGATE bits in the T1CON register.
4. Set or clear the TSYNC bit in T1CON to select synchronous or asynchronous operation.
5. Load the timer period value into the PR1 register.
6. If interrupts are required, set the interrupt enable bit, T1IE. Use the priority bits, T1IP<2:0>, to set the interrupt priority.

FIGURE 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`’

13.0 TIMER2/3, TIMER4/5, TIMER6/7 AND TIMER8/9

Note:	This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 11. "Timers"** (DS70205) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

The Timer2/3, Timer4/5, Timer6/7 and Timer8/9 modules are 32-bit timers that can also be configured as four independent 16-bit timers with selectable operating modes.

As a 32-bit timer, Timer2/3, Timer4/5, Timer6/7 and Timer8/9 operate in three modes:

- Two Independent 16-bit Timers (e.g., Timer2 and Timer3) with all 16-bit operating modes (except Asynchronous Counter mode)
- Single 32-bit Timer
- Single 32-bit Synchronous Counter

They also support the following features:

- Timer Gate Operation
- Selectable Prescaler Settings
- Timer Operation during Idle and Sleep modes
- Interrupt on a 32-bit Period Register Match
- Time Base for Input Capture and Output Compare Modules (Timer2 and Timer3 only)
- ADC1 Event Trigger (Timer2/3 only)
- ADC2 Event Trigger (Timer4/5 only)

Individually, all eight of the 16-bit timers can function as synchronous timers or counters. They also offer the features listed above, except for the event trigger; this is implemented only with Timer2/3. The operating modes and enabled features are determined by setting the appropriate bit(s) in the T2CON, T3CON, T4CON, T5CON, T6CON, T7CON, T8CON and T9CON registers. T2CON, T4CON, T6CON and T8CON are shown in generic form in Register 13-1. T3CON, T5CON, T7CON and T9CON are shown in Register 13-2.

For 32-bit timer/counter operation, Timer2, Timer4, Timer6 or Timer8 is the least significant word; Timer3, Timer5, Timer7 or Timer9 is the most significant word of the 32-bit timers.

Note:	For 32-bit operation, T3CON, T5CON, T7CON and T9CON control bits are ignored. Only T2CON, T4CON, T6CON and T8CON control bits are used for setup and control. Timer2, Timer4, Timer6 and Timer8 clock and gate inputs are utilized for the 32-bit timer modules, but an interrupt is generated with the Timer3, Timer5, Ttimer7 and Timer9 interrupt flags.

To configure Timer2/3, Timer4/5, Timer6/7 or Timer8/9 for 32-bit operation, do the following:

1. Set the corresponding T32 control bit.
2. Select the prescaler ratio for Timer2, Timer4, Timer6 or Timer8 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, PR5, PR7 or PR9 contains the most significant word of the value, while PR2, PR4, PR6 or PR8 contains the least significant word.
5. If interrupts are required, set the interrupt enable bit, T3IE, T5IE, T7IE or T9IE. Use the priority bits, T3IP<2:0>, T5IP<2:0>, T7IP<2:0> or T9IP<2:0>, to set the interrupt priority. While Timer2, Timer4, Timer6 or Timer8 control the timer, the interrupt appears as a Timer3, Timer5, Timer7 or Timer9 interrupt.
6. Set the corresponding TON bit.

The timer value at any point is stored in the register pair, TMR3:TMR2, TMR5:TMR4, TMR7:TMR6 or TMR9:TMR8. TMR3, TMR5, TMR7 or TMR9 always contain the most significant word of the count, while TMR2, TMR4, TMR6 or TMR8 contain the least significant word.

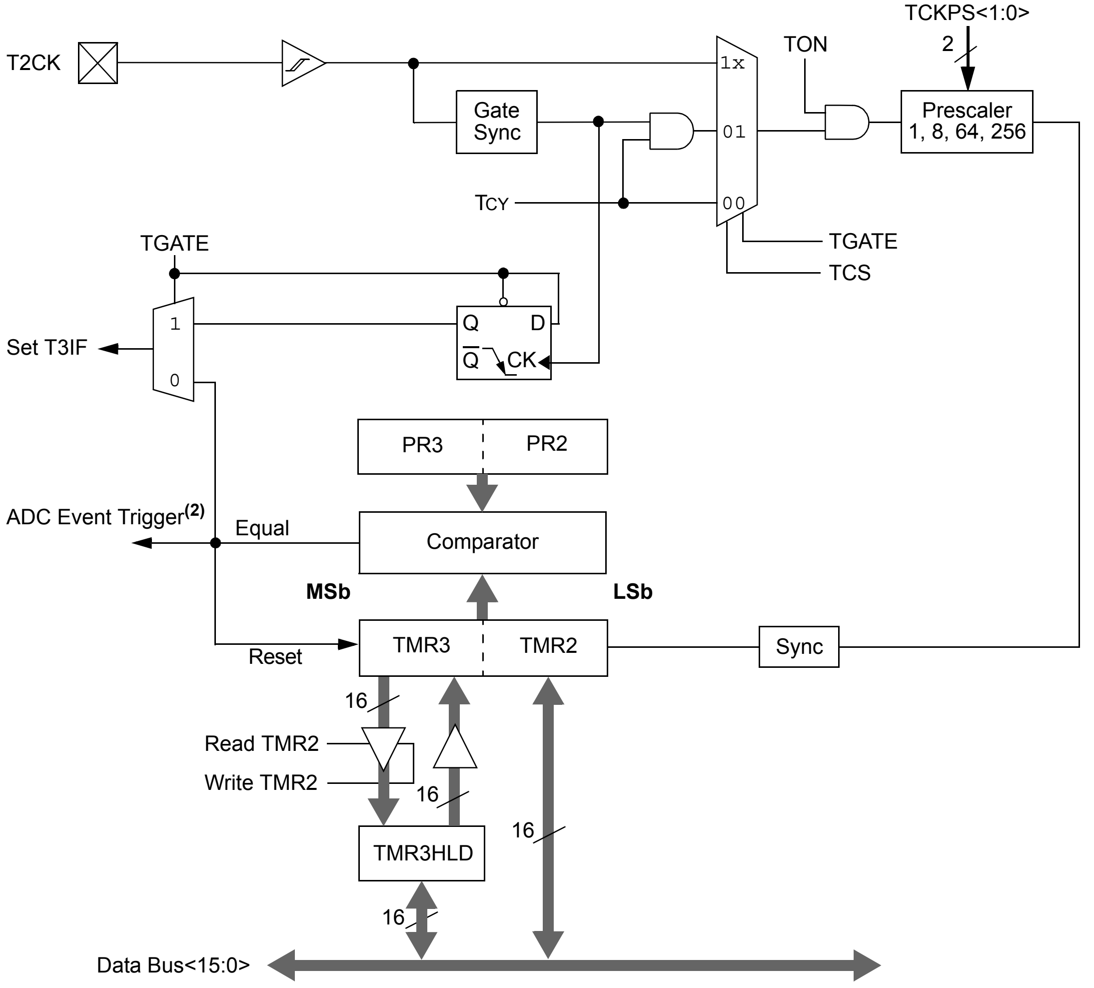
To configure any of the timers for individual 16-bit operation, do the following:

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.

A block diagram for a 32-bit timer pair (Timer4/5) example is shown in Figure 13-1, and a timer (Timer4) operating in 16-bit mode example is shown in Figure 13-2.

Note:	Only Timer2 and Timer3 can trigger a DMA data transfer.

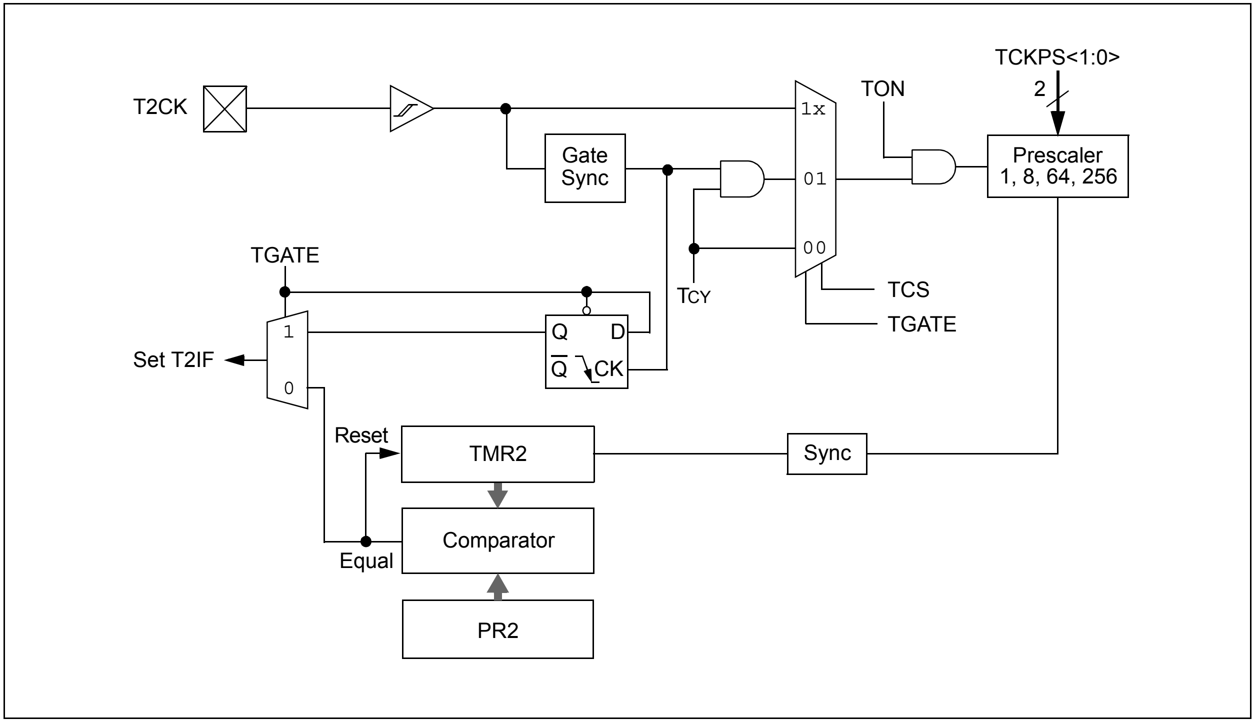
FIGURE 13-1: TIMER2/3 (32-BIT) BLOCK DIAGRAM[1]



Note 1: The 32-bit timer control bit, T32, must be set for 32-bit timer/counter operation. All control bits are respective to the T2CON register.

2: The ADC event trigger is available only on Timer2/3.

FIGURE 13-2: TIMER2 (16-BIT) BLOCK DIAGRAM



REGISTER 13-1: TxCON (T2CON, T4CON, T6CON OR T8CON) CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
—	TGATE	TCKPS<1:0>		T32	—	TCS(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 **TON:** Timerx On bit

When T32 = 1:
1 = Starts 32-bit Timerx/y
0 = Stops 32-bit Timerx/y

When T32 = 0:
1 = Starts 16-bit Timerx
0 = Stops 16-bit Timerx

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **TSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

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

bit 6 **TGATE:** Timerx Gated Time Accumulation Enable bit

When TCS = 1:
This bit is ignored

When TCS = 0:
1 = Gated time accumulation enabled
0 = Gated time accumulation disabled

bit 5-4 **TCKPS<1:0>:** Timerx Input Clock Prescale Select bits
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3 **T32:** 32-bit Timer Mode Select bit
1 = Timerx and Timery form a single 32-bit timer
0 = Timerx and Timery act as two 16-bit timers

bit 2 **Unimplemented:** Read as ‘0’

bit 1 **TCS:** Timerx Clock Source Select bit(1)
1 = External clock from pin TxCK (on the rising edge)
0 = Internal clock (FCY)

bit 0 **Unimplemented:** Read as ‘0’

Note 1: The TxCK pin is not available on all timers. Refer to the **“Pin Diagrams”** section for the available pins.

REGISTER 13-2: TyCON (T3CON, T5CON, T7CON OR T9CON) CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON(1)	—	TSIDL(2)	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
—	TGATE(1)	TCKPS<1:0>(1)		—	—	TCS(1,3)	—
bit 7							bit 0

Legend:

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

bit 15 **TON:** Timery On bit(1)
1 = Starts 16-bit Timery
0 = Stops 16-bit Timery

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **TSIDL:** Stop in Idle Mode bit(2)
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

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

bit 6 **TGATE:** Timery Gated Time Accumulation Enable bit(1)
When TCS = 1:
This bit is ignored
When TCS = 0:
1 = Gated time accumulation enabled
0 = Gated time accumulation disabled

bit 5-4 **TCKPS<1:0>:** Timer3 Input Clock Prescale Select bits(1)
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1 **TCS:** Timery Clock Source Select bit(1,3)
1 = External clock from pin TyCK (on the rising edge)
0 = Internal clock (FCY)

bit 0 **Unimplemented:** Read as ‘0’

Note 1: When 32-bit operation is enabled (T2CON<3> = 1), these bits have no effect on Timery operation; all timer functions are set through T2CON.

2: When 32-bit timer operation is enabled (T32 = 1) in the Timer Control register (TxCON<3>), the TSIDL bit must be cleared to operate the 32-bit timer in Idle mode.

3: The TyCK pin is not available on all timers. Refer to the **“Pin Diagrams”** section for the available pins.

NOTES:

14.0 INPUT CAPTURE

> **Note:** This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 12. "Input Capture"** (DS70198) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

The input capture module is useful in applications requiring frequency (period) and pulse measurement. The dsPIC33FJXXXMCX06/X08/X10 devices support up to eight 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) of input at ICx pin
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

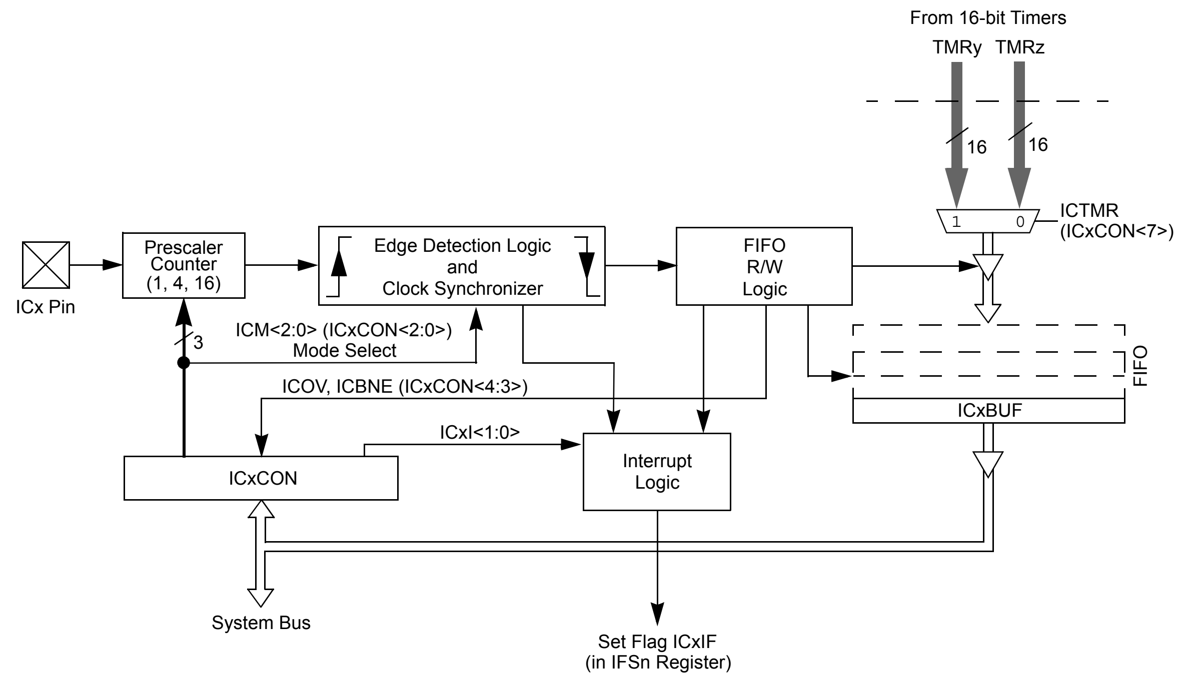
Each input capture channel can select between one of two 16-bit timers (Timer2 or Timer3) for the time base. The selected timer can use either an internal or external clock.

Other operational features include the following:

- 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
- Input capture can also be used 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`' (ICI<1:0> = `00`).

FIGURE 14-1: INPUT CAPTURE BLOCK DIAGRAM



Note: An 'x' in a signal, register or bit name denotes the number of the capture channel.

14.1 Input Capture Registers

REGISTER 14-1: ICxCON: INPUT CAPTURE x CONTROL REGISTER

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

R/W-0	R/W-0	R/W-0	R-0, HC	R-0, HC	R/W-0	R/W-0	R/W-0
ICTMR(1)	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 will halt in CPU Idle mode
`0` = Input capture module will continue to operate in CPU Idle mode

bit 12-8 **Unimplemented:** Read as ‘`0`’

bit 7 **ICTMR:** Input Capture Timer Select bits(1)
`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

Note 1: Timer selections may vary. Refer to the device data sheet for details.

15.0 OUTPUT COMPARE

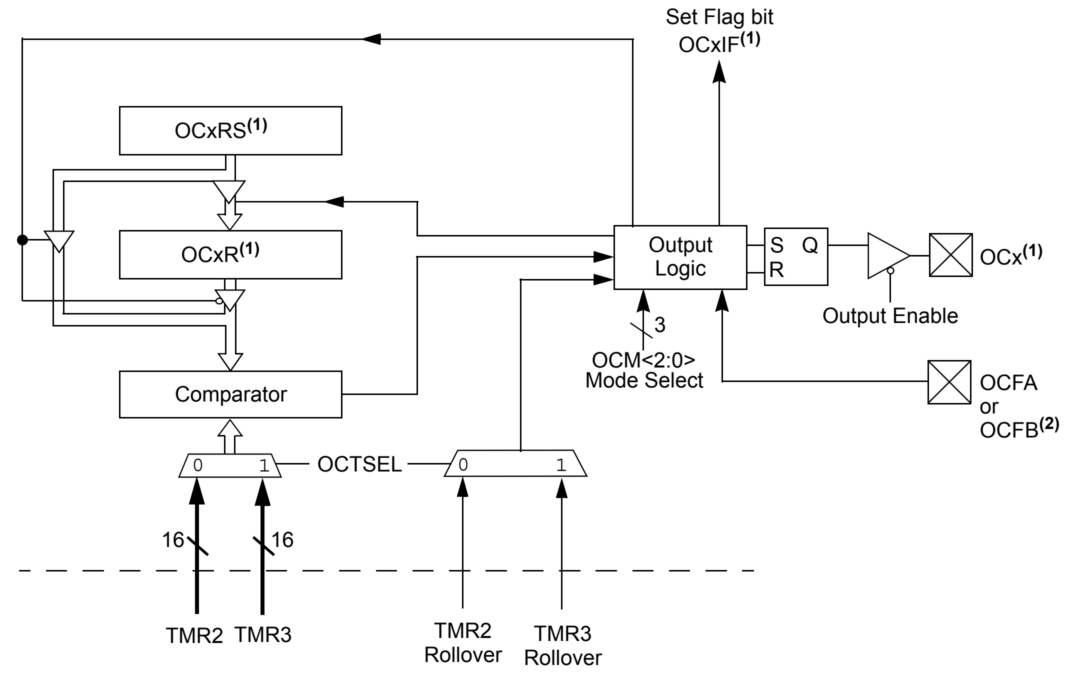
Note: This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the *"dsPIC33F Family Reference Manual"*, **Section 13. "Output Compare"** (DS70209), which is available on the Microchip web site (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



Note 1: An 'x' in a signal, register or bit name denotes the number of the output compare channels.

2: The OCFA pin controls OC1 through OC4. The OCFB pin controls OC5 through OC8.

15.1 Output Compare Modes

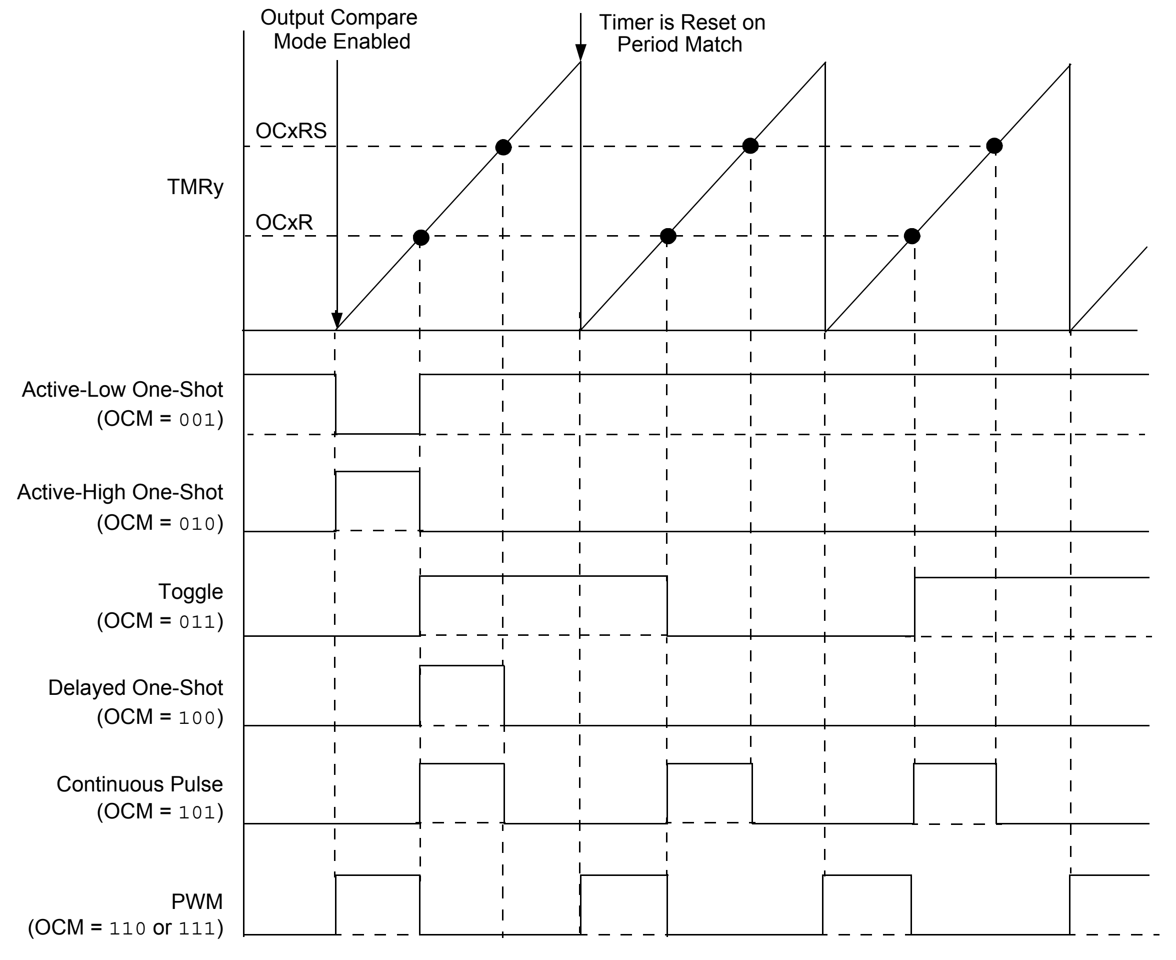
Configure the Output Compare modes by setting the appropriate Output Compare Mode (OCM<2:0>) bits in the Output Compare Control (OCxCON<2:0>) register. Table 15-1 lists the different bit settings for the Output Compare modes. Figure 15-2 illustrates the output compare operation for various modes. The user application must disable the associated timer when writing to the Output Compare Control registers to avoid malfunctions.

Note: 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	Current output is maintained	OCx rising and falling edge
`100`	Delayed One-Shot	`0`	OCx falling edge
`101`	Continuous Pulse	`0`	OCx falling edge
`110`	PWM without Fault Protection	‘`0`’, if OCxR is zero ‘`1`’, if OCxR is non-zero	No interrupt
`111`	PWM 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)

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

bit 15-14	**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 dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 14. "Motor Control PWM"** (DS70187) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

This module simplifies the task of generating multiple, synchronized Pulse-Width Modulated (PWM) outputs. In particular, 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)

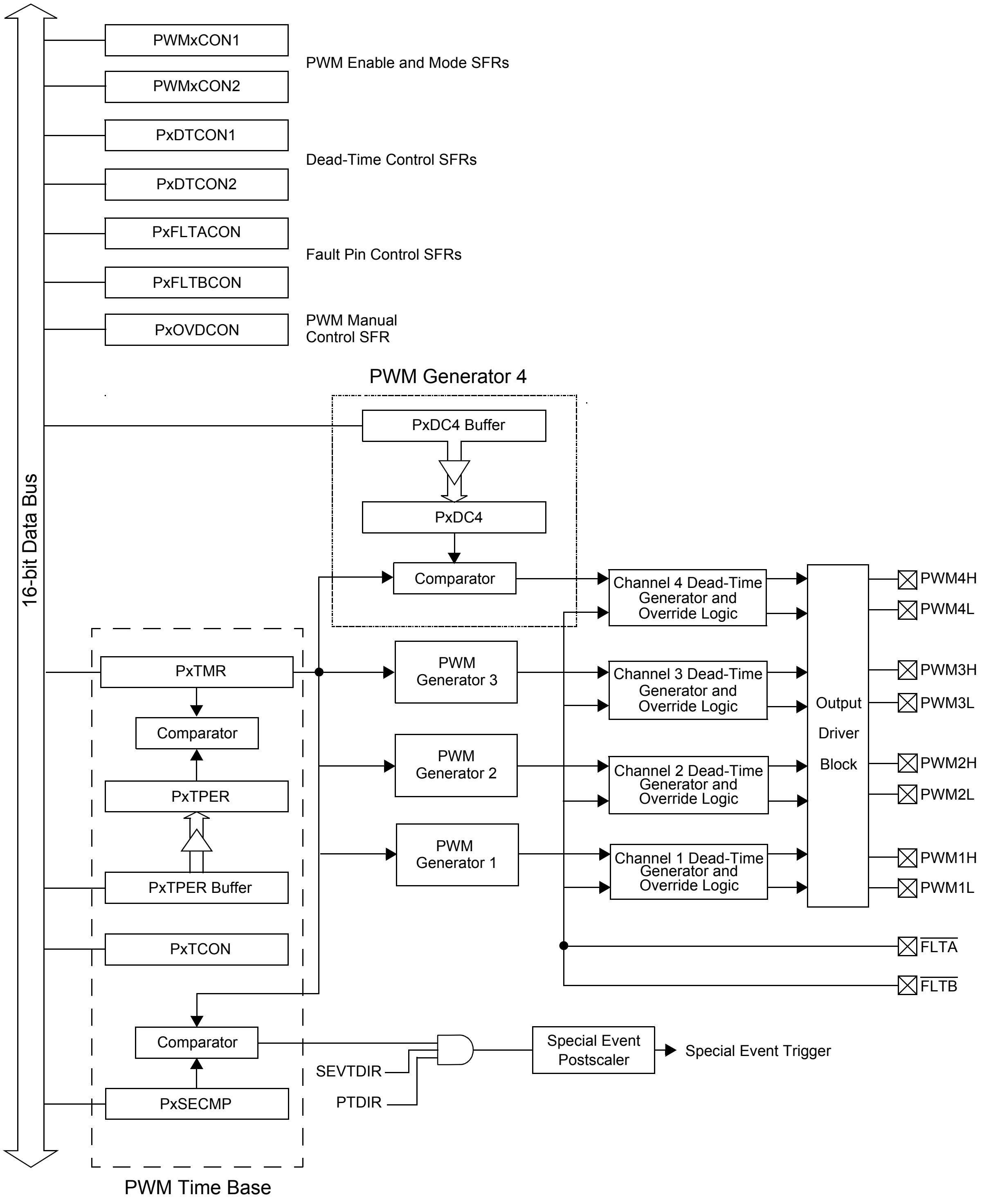
The PWM module has the following features:

- Eight PWM I/O pins with four duty cycle generators
- 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
- '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 are configurable to be immediate or synchronized to the PWM time base

This module contains four duty cycle generators, numbered 1 through 4. The module has eight PWM output pins, numbered PWM1H/PWM1L through PWM4H/PWM4L. The eight 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.

The PWM module allows several modes of operation which are beneficial for specific power control applications.

FIGURE 16-1: PWM MODULE BLOCK DIAGRAM



Note: For clarity, details of PWM Generator 1, 2 and 3 are not shown.

REGISTER 16-1: PxTCON: 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 TCY (1:64 prescale)
`10` = PWM time base input clock period is 16 TCY (1:16 prescale)
`01` = PWM time base input clock period is 4 TCY (1:4 prescale)
`00` = PWM time base input clock period is TCY (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: PxTMR: 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: PxTPER: 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 will occur when the PWM time base is counting downwards
`0` = A Special Event Trigger will occur when the PWM time base is counting upwards

bit 14-0 **SEVTCMP<14:0>:** Special Event Compare Value bits[2]

Note 1: SEVTDIR is compared with PTDIR (PTMR<15>) to generate the Special Event Trigger.

2: SEVTCMP<14:0> is compared with PTMR<14:0> to generate the Special Event Trigger.

REGISTER 16-5: PWMxCON1: PWM CONTROL REGISTER 1

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	PMOD4	PMOD3	PMOD2	PMOD1
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
PEN4H[1]	PEN3H[1]	PEN2H[1]	PEN1H[1]	PEN4L[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 15-12	**Unimplemented:** Read as ‘`0`’
bit 11-8	**PMOD<4:1>:** 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 7-4	**PEN4H:PEN1H:** PWMxH I/O Enable bits[1] `1` = PWMxH pin is enabled for PWM output `0` = PWMxH pin is disabled; I/O pin becomes general purpose I/O
bit 3-0	**PEN4L:PEN1L:** PWMxL I/O Enable bits[1] `1` = PWMxL pin is enabled for PWM output `0` = PWMxL pin is 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.

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 PDC registers are immediate `0` = Updates to the active PDC registers are synchronized to the PWM time base
bit 1	**OSYNC:** Output Override Synchronization bit `1` = Output overrides via the OVDCON register are synchronized to the PWM time base `0` = Output overrides via the OVDCON 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

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
DTS4A	DTS4I	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 15-8	**Unimplemented:** Read as ‘`0`’
bit 7	**DTS4A:** Dead-Time Select for PWM4 Signal Going Active bit `1` = Dead time provided from Unit B `0` = Dead time provided from Unit A
bit 6	**DTS4I:** Dead-Time Select for PWM4 Signal Going Inactive bit `1` = Dead time provided from Unit B `0` = Dead time provided from Unit A
bit 5	**DTS3A:** Dead-Time Select for PWM3 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 PWM3 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 PWM2 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 PWM2 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 PWM1 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 PWM1 Signal Going Inactive bit `1` = Dead time provided from Unit B `0` = Dead time provided from Unit A

REGISTER 16-9: PxFLTACON: FAULT A CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FAOV4H	FAOV4L	FAOV3H	FAOV3L	FAOV2H	FAOV2L	FAOV1H	FAOV1L
bit 15							bit 8

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
FLTAM	—	—	—	FAEN4	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 15-8 **FAOVxH<4:1>:FAOVxL<4:1>:** 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 states programmed in FLTACON<15:8>

bit 6-4 **Unimplemented:** Read as ‘`0`’

bit 3 **FAEN4:** Fault Input A Enable bit

`1` = PWM4H/PWM4L pin pair is controlled by Fault Input A
`0` = PWM4H/PWM4L pin pair is not controlled by Fault Input A

bit 2 **FAEN3:** Fault Input A Enable bit

`1` = PWM3H/PWM3L pin pair is controlled by Fault Input A
`0` = PWM3H/PWM3L pin pair is not controlled by Fault Input A

bit 1 **FAEN2:** Fault Input A Enable bit

`1` = PWM2H/PWM2L pin pair is controlled by Fault Input A
`0` = PWM2H/PWM2L pin pair is not controlled by Fault Input A

bit 0 **FAEN1:** Fault Input A Enable bit

`1` = PWM1H/PWM1L pin pair is controlled by Fault Input A
`0` = PWM1H/PWM1L pin pair is not controlled by Fault Input A

REGISTER 16-10: PxFLTBCON: FAULT B CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FBOV4H	FBOV4L	FBOV3H	FBOV3L	FBOV2H	FBOV2L	FBOV1H	FBOV1L
bit 15							bit 8

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
FLTBM	—	—	—	FBEN4[1]	FBEN3[1]	FBEN2[1]	FBEN1[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 15-8 **FBOVxH<4:1>:FBOVxL<4:1>:** Fault Input B 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 **FLTBM:** Fault B Mode bit
`1` = The Fault B input pin functions in the Cycle-by-Cycle mode
`0` = The Fault B input pin latches all control pins to the states programmed in FLTBCON<15:8>

bit 6-4 **Unimplemented:** Read as ‘`0`’

bit 3 **FBEN4:** Fault Input B Enable bit[1]
`1` = PWM4H/PWM4L pin pair is controlled by Fault Input B
`0` = PWM4H/PWM4L pin pair is not controlled by Fault Input B

bit 2 **FBEN3:** Fault Input B Enable bit[1]
`1` = PWM3H/PWM3L pin pair is controlled by Fault Input B
`0` = PWM3H/PWM3L pin pair is not controlled by Fault Input B

bit 1 **FBEN2:** Fault Input B Enable bit[1]
`1` = PWM2H/PWM2L pin pair is controlled by Fault Input B
`0` = PWM2H/PWM2L pin pair is not controlled by Fault Input B

bit 0 **FBEN1:** Fault Input B Enable bit[1]
`1` = PWM1H/PWM1L pin pair is controlled by Fault Input B
`0` = PWM1H/PWM1L pin pair is not controlled by Fault Input B

Note 1: Fault A pin has priority over Fault B pin, if enabled.

REGISTER 16-11: PxOVDCON: OVERRIDE CONTROL REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
POVD4H	POVD4L	POVD3H	POVD3L	POVD2H	POVD2L	POVD1H	POVD1L
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
POUT4H	POUT4L	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 15-8 **POVDxH<4:1>:POVDxL<4:1>:** 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 7-0 **POUTxH<4:1>:POUTxL<4:1>:** 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

REGISTER 16-12: 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 16-13: PxDC2: 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 16-14: PxDC3: 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

REGISTER 16-15: PxDC4: PWM DUTY CYCLE REGISTER 4

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PDC4<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
PDC4<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	**PDC4<15:0>:** PWM Duty Cycle #4 Value bits

17.0 QUADRATURE ENCODER INTERFACE (QEI) MODULE

> **Note:** This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 15. "Quadrature Encoder Interface (QEI)"** (DS70208) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

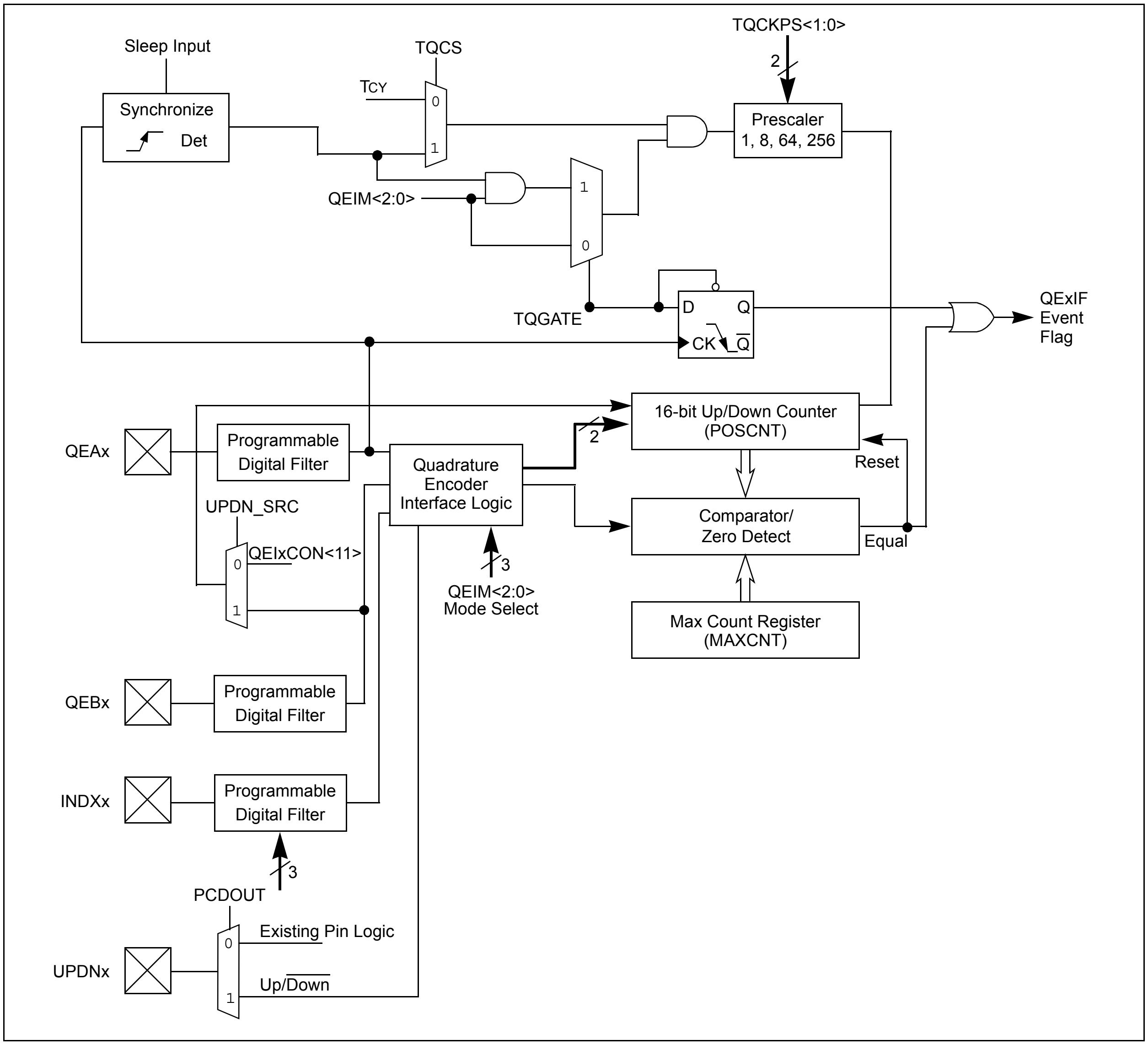
This section describes the Quadrature Encoder Interface (QEI) module and associated operational modes. The QEI module provides the interface to incremental encoders for obtaining mechanical position data.

The operational features of the QEI include the following:

- Three input channels for two phase signals and an 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

The QEI module's operating mode is determined by setting the appropriate bits, QEIM<2:0> (QEICON<10:8>). Figure 17-1 depicts the Quadrature Encoder Interface block diagram.

FIGURE 17-1: QUADRATURE ENCODER INTERFACE BLOCK DIAGRAM



REGISTER 17-1: QEIxCON: QEI CONTROL REGISTER

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 = Position count error has occurred 0 = No position count error has occurred (CNTERR flag only applies when QEIM<2:0> = ‘`110`’ or ‘`100`’)
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 `1` = Position Counter direction is positive (+) `0` = Position Counter direction is negative (-) (Read-only bit when QEIM<2:0> = ‘`1XX`’) (Read/Write bit when QEIM<2:0> = ‘`001`’)
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 (MAXCNT) `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 (MAXCNT) `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)
bit 5	**TQGATE:** Timer Gated Time Accumulation Enable bit `1` = Timer gated time accumulation enabled `0` = Timer gated time accumulation disabled

REGISTER 17-1: QEIxCON: QEI CONTROL REGISTER (CONTINUED)

bit 4-3	**TQCKPS<1:0>:** Timer 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 (Prescaler utilized for 16-bit timer mode only)
bit 2	**POSRES:** Position Counter Reset Enable bit 1 = Index Pulse resets Position Counter 0 = Index Pulse does not reset Position Counter (Bit only applies when QEIM<2:0> = 100 or 110)
bit 1	**TQCS:** Timer Clock Source Select bit 1 = External clock from pin QEA (on the rising edge) 0 = Internal clock TCY)
bit 0	**UPDN_SRC:** Position Counter Direction Selection Control bit 1 = QEB pin state defines Position Counter direction 0 = Control/status bit UPDN (QEICON<11>) defines Position Counter (POSCNT) direction **Note:** When configured for QEI mode, control bit is a ‘don’t care’.

REGISTER 17-2: DFLTxCON: DIGITAL FILTER CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	IMV<1:0>		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 to specify the state of the QEAx and QEBx input pins during an index pulse when the POSxCNT register is to be reset

In 4X 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 2X 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 dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 18. "Serial Peripheral Interface (SPI)"** (DS70206) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (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 may be serial EEPROMs, shift registers, display drivers, ADC, etc. The SPI module is compatible with SPI and SIOP from Motorola®.

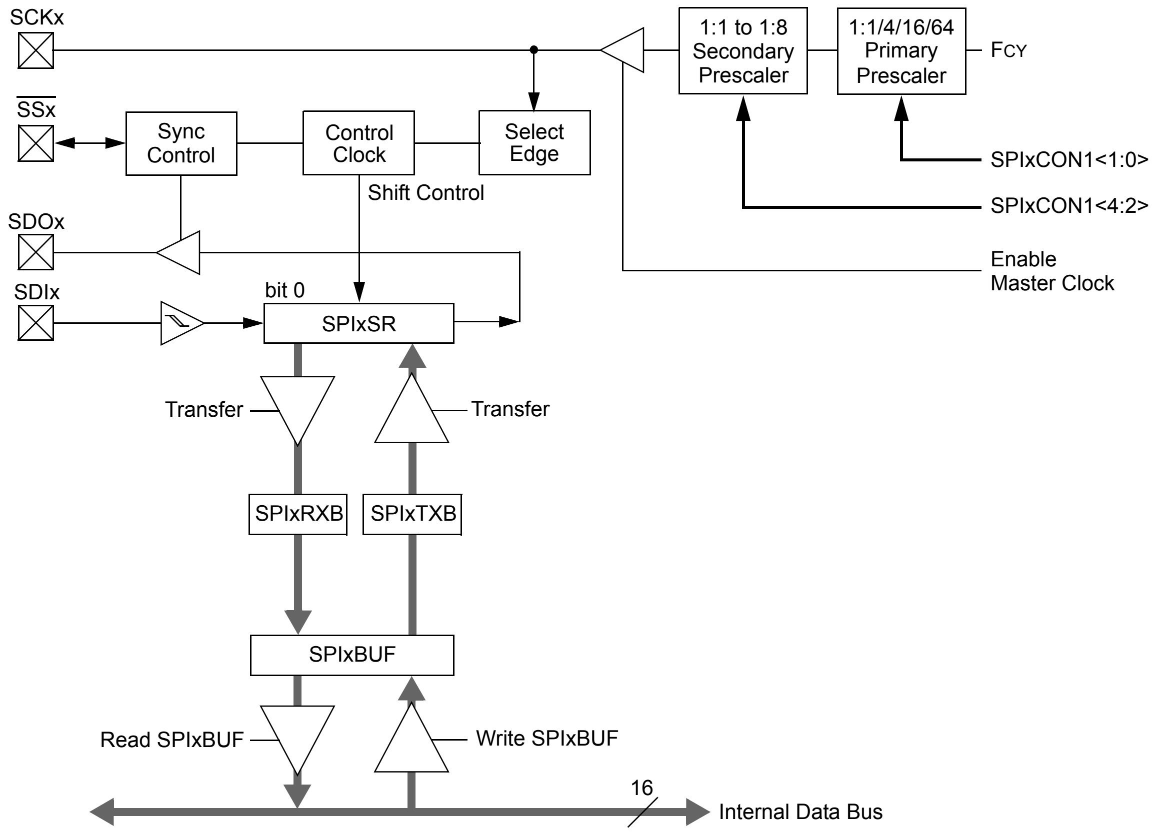
Note: In this section, the SPI modules are referred to together as SPIx, or separately as SPI1 and SPI2. Special Function Registers will follow a similar notation. For example, SPIxCON refers to the control register for the SPI1 or SPI2 module.

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 various status conditions.

The serial interface consists of 4 pins: SDIx (serial data input), SDOx (serial data output), SCKx (shift clock input or output) and $\overline{\text{SSx}}$ (active-low slave select).

In Master mode operation, SCK is a clock output, but 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 $\overline{\text{SSx}}$ 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` = $\overline{\text{SSx}}$ pin used for Slave mode
`0` = $\overline{\text{SSx}}$ 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. The user should program this bit to ‘`0`’ for the Framed SPI modes (FRMEN = `1`).

2: Do not set both the 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. The user should program this bit to ‘`0`’ for the Framed SPI modes (FRMEN = `1`).

2: Do not set both the 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 ($\overline{\text{SSx}}$ 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™ (I^2C™)

Note:	This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 19. “Inter-Integrated Circuit™ (I^2C™)”** (DS70195) in the *“dsPIC33F Family Reference Manual”*, which is available from the Microchip web site (www.microchip.com).

The Inter-Integrated Circuit (I^2C) module, with its 16-bit interface, provides complete hardware support for both Slave and Multi-Master modes of the I^2C serial communication standard.

The dsPIC33FJXXXMCX06/X08/X10 devices have up to two I^2C interface modules, denoted as I2C1 and I2C2. Each I^2C module has a 2-pin interface: the SCLx pin is clock and the SDAx pin is data.

Each I^2C module ‘x’ (x = 1 or 2) offers the following key features:

- I^2C interface supports both master and slave operation.
- I^2C Slave mode supports 7- and 10-bit addresses.
- I^2C Master mode supports 7- and 10-bit addresses.
- I^2C Port allows bidirectional transfers between master and slaves.
- Serial clock synchronization for the I^2C port can be used as a handshake mechanism to suspend and resume serial transfer (SCLREL control).
- I^2C supports multi-master operation; it detects bus collision and will arbitrate accordingly.

19.1 Operating Modes

The hardware fully implements all the master and slave functions of the I^2C Standard and Fast mode specifications, as well as 7 and 10-bit addressing.

The I^2C module can operate either as a slave or a master on an I^2C bus.

The following types of I^2C operation are supported:

- I^2C slave operation with 7-bit address
- I^2C slave operation with 10-bit address
- I^2C master operation with 7 or 10-bit address

For details about the communication sequence in each of these modes, please refer to the “*dsPIC30F Family Reference Manual”*.

19.2 I^2C Registers

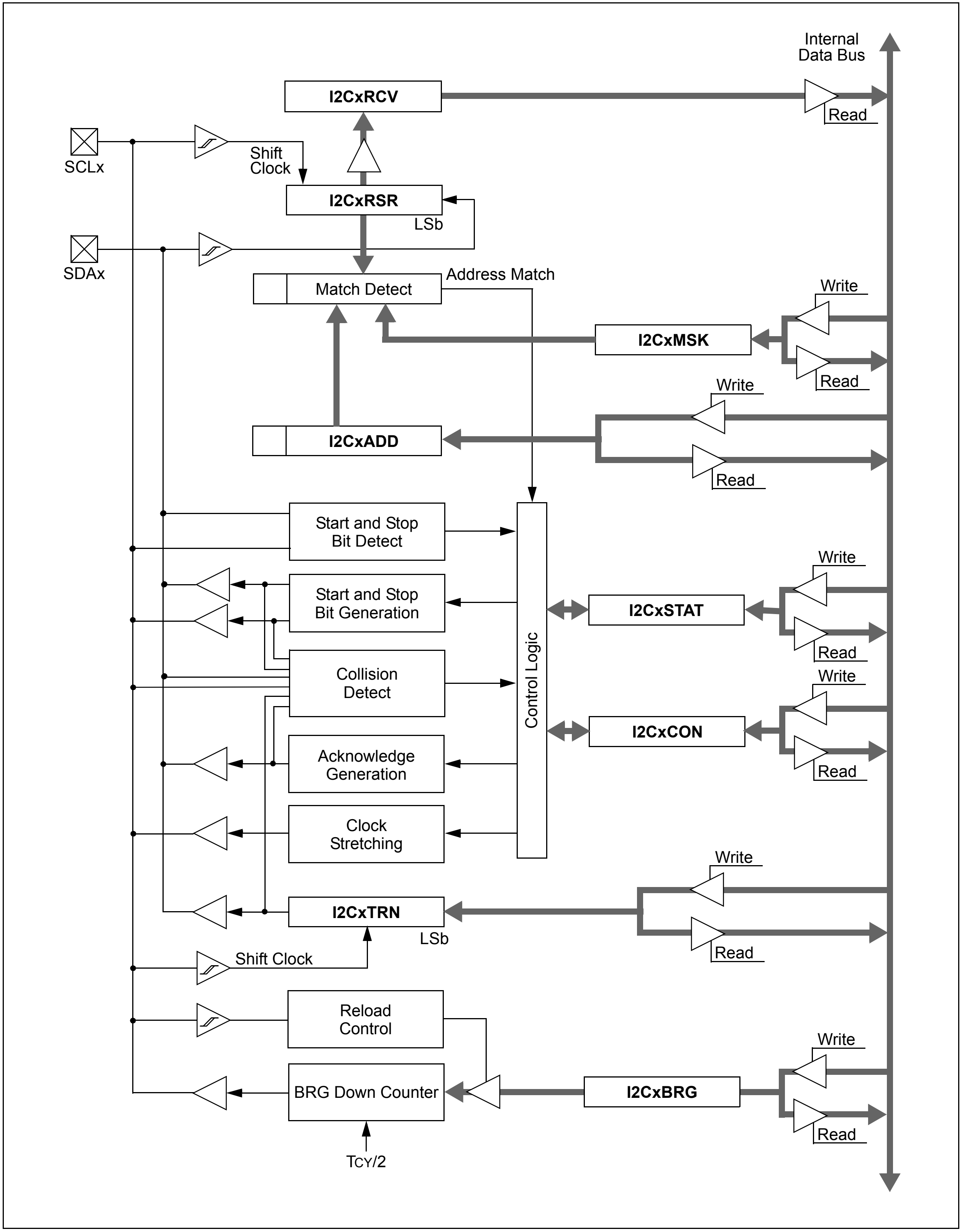
I2CxCON and I2CxSTAT are control and status registers, respectively. The I2CxCON register is readable and writable. The lower six bits of I2CxSTAT are read-only. The remaining bits of the I2CSTAT are read/write.

I2CxRSR is the shift register used for shifting data, whereas I2CxRCV is the buffer register to which data bytes are written, or from which data bytes are read. I2CxRCV is the receive buffer. 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 OR 2)



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	HS = Set in hardware	HC = Cleared in hardware
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **I2CEN:** I2Cx Enable bit

`1` = Enables the I2Cx module and configures the SDAx and SCLx pins as serial port pins
`0` = Disables the I2Cx module. All I^2C 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 I^2C slave)

`1` = Release SCLx clock
`0` = Hold SCLx clock low (clock stretch)

If STREN = `1`:
Bit is R/W (i.e., software may write ‘`0`’ to initiate stretch and write ‘`1`’ to release clock). Hardware clear at beginning of slave transmission. Hardware clear at end of slave reception.

If STREN = `0`:
Bit is R/S (i.e., software may only write ‘`1`’ to release clock). Hardware clear at beginning of slave transmission.

bit 11 **IPMIEN:** Intelligent Peripheral Management Interface (IPMI) Enable bit

`1` = IPMI 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 I^2C 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 I^2C 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 I^{2}C master, applicable during master receive) Value that will be 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 I^{2}C 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 I^{2}C master) `1` = Enables Receive mode for I^{2}C. 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 I^{2}C 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 I^{2}C 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 I^{2}C 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 I^2C 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 I^2C master, applicable to master transmit operation)
`1` = Master transmit is in progress (8 bits + ACK)
`0` = Master transmit is not in progress
Hardware set at beginning of master transmission. Hardware clear at end of slave Acknowledge.

bit 13-11 **Unimplemented:** Read as ‘`0`’

bit 10 **BCL:** Master Bus Collision Detect bit
`1` = A bus collision has been detected during a master operation
`0` = No collision
Hardware set at detection of bus collision.

bit 9 **GCSTAT:** General Call Status bit
`1` = General call address was received
`0` = General call address was not received
Hardware set when address matches general call address. Hardware clear at Stop detection.

bit 8 **ADD10:** 10-Bit Address Status bit
`1` = 10-bit address was matched
`0` = 10-bit address was not matched
Hardware set at match of 2nd byte of matched 10-bit address. Hardware clear at Stop detection.

bit 7 **IWCOL:** Write Collision Detect bit
`1` = An attempt to write the I2CxTRN register failed because the I^2C 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 I^2C 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 I^2C slave) 1 = Read – indicates data transfer is output from slave 0 = Write – indicates data transfer is input to slave Hardware set or clear after reception of I^2C 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 dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 17. "UART"** (DS70188) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

The Universal Asynchronous Receiver Transmitter (UART) module is one of the serial I/O modules available in the dsPIC33FJXXXMCX06/X08/X10 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 $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ pins and also includes an IrDA® encoder and decoder.

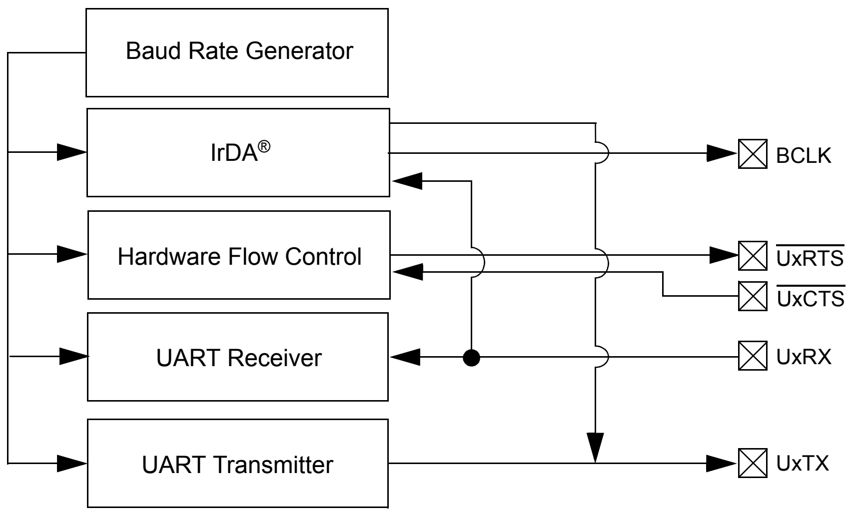
The primary features of the UART module are:

- Full-Duplex, 8-bit 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
- Supports Automatic Baud Rate Detection
- IrDA® Encoder and Decoder Logic
- 16x Baud Clock Output for IrDA® Support

A simplified block diagram of the UART is shown in Figure 20-1. The UART module consists of the key important hardware elements:

- Baud Rate Generator
- Asynchronous Transmitter
- Asynchronous Receiver

FIGURE 20-1: UART SIMPLIFIED BLOCK DIAGRAM



Note 1: Both UART1 and UART2 can trigger a DMA data transfer. If U1TX, U1RX, U2TX or U2RX is selected as a DMA IRQ source, a DMA transfer occurs when the U1TXIF, U1RXIF, U2TXIF or U2RXIF bit gets set as a result of a UART1 or UART2 transmission or reception.

2: If DMA transfers are required, the UART TX/RX FIFO buffer must be set to a size of 1 byte/word (i.e., UTXISEL<1:0> = `00` and URXISEL<1:0> = `00`).

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)

`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 $\overline{\text{UxRTS}}$ Pin bit

`1` = $\overline{\text{UxRTS}}$ pin in Simplex mode
`0` = $\overline{\text{UxRTS}}$ pin in Flow Control mode

bit 10 **Unimplemented:** Read as ‘`0`’

bit 9-8 **UEN<1:0>:** UARTx Enable bits

`11` = UxTX, UxRX and BCLK pins are enabled and used; $\overline{\text{UxCTS}}$ pin controlled by port latches
`10` = UxTX, UxRX, $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ pins are enabled and used
`01` = UxTX, UxRX and $\overline{\text{UxRTS}}$ pins are enabled and used; $\overline{\text{UxCTS}}$ pin controlled by port latches
`00` = UxTX and UxRX pins are enabled and used; and $\overline{\text{UxRTS}}$/BCLK pins controlled by port latches

bit 7 **WAKE:** Wake-up on Start bit Detect During Sleep Mode Enable bit

`1` = UARTx will continue to sample the 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 (0x55) 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	RIDLE	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) will reset 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 dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 21. "Enhanced Controller Area Network (ECAN™)"** (DS70185) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (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 dsPIC33FJXXXMCX06/X08/X10 devices contain up to two ECAN modules.

The CAN module is a communication controller implementing the CAN 2.0 A/B protocol, as defined in the BOSCH specification. The module will support 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 may 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 may contain up to 8 bytes of data)
- Up to 32 receive buffers (each buffer may 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 both CAN1 and CAN2) 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 CAN 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:

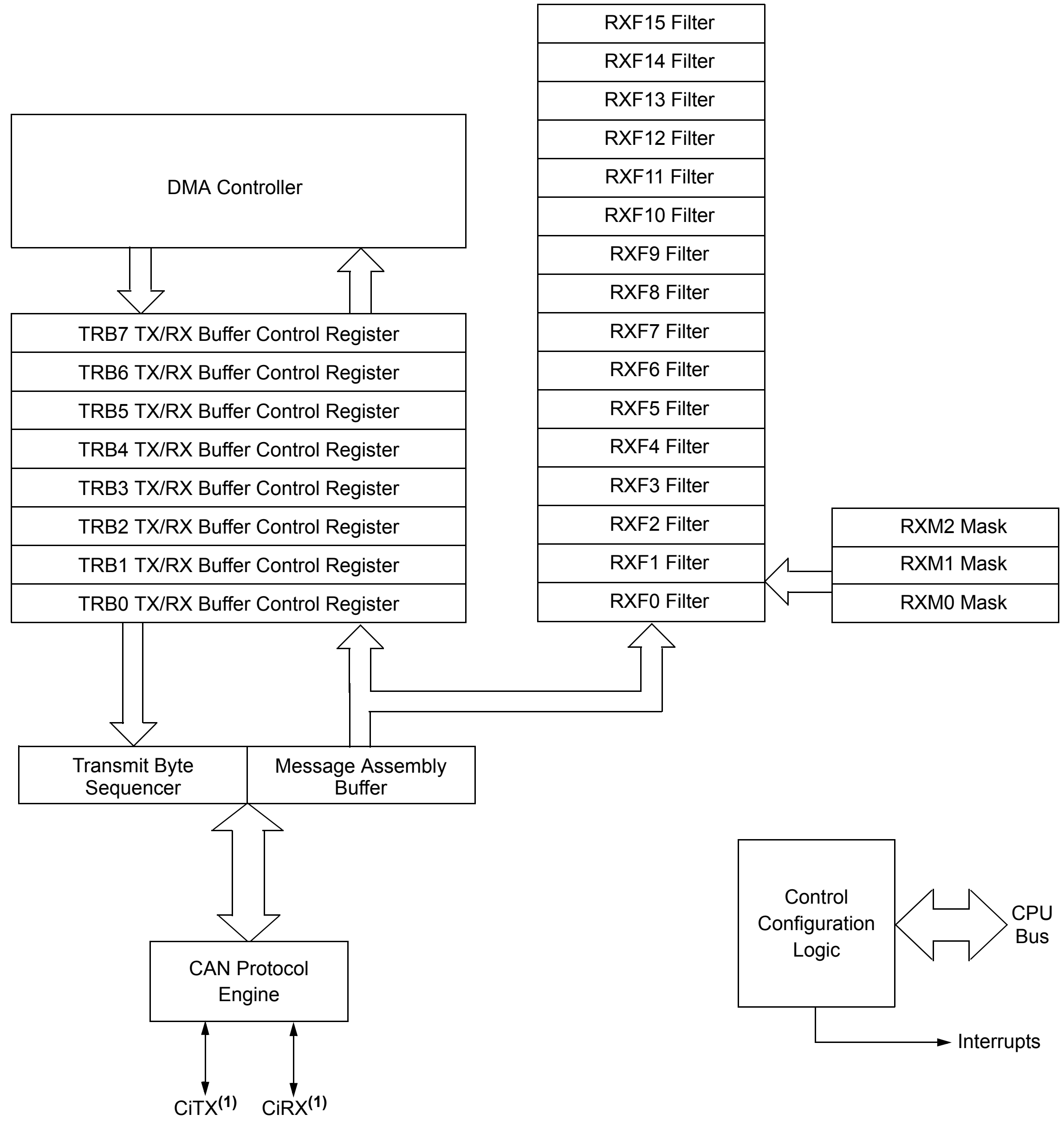
  It is possible for a destination node to request the data from the source. For this purpose, the destination node sends a remote frame with an identifier that matches the identifier of the required data frame. The appropriate data source node will then send a data frame as a response to this remote request.
- 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 may generate a maximum of 2 sequential overload frames to delay the start of the next message.
- Interframe Space:

  Interframe space separates a proceeding frame (of whatever type) from a following data or remote frame.

FIGURE 21-1: ECAN™ MODULE BLOCK DIAGRAM



Note 1: i = 1 or 2 refers to a particular ECAN™ module (ECAN1 or ECAN2).

21.3 Modes of Operation

The CAN 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 will 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 will not transmit or receive. The error counters are cleared and the interrupt flags remain unchanged. The programmer will have access to Configuration registers that are access restricted in other modes. The module will protect 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 CAN module will not be 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 will not transmit or receive. The module has the ability to set the WAKIF bit due to bus activity, however, any pending interrupts will remain and the error counters will retain their value.

If the REQOP<2:0> bits (CiCTRL1<10:8>) = `001`, the module will enter the Module Disable mode. If the module is active, the module will wait 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 will revert 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 CAN module is allowed to transmit in a particular mode of operation and a transmission is requested immediately after the CAN 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 will assume the CAN bus functions. The module will transmit 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 will connect 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:	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
Signal all transmit buffers to abort transmission. Module will clear this bit when all transmissions are aborted

bit 11 **Reserved:** Do no use

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 – do not use
`110` = Reserved – do not use
`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:			
R = Readable bit	W = Writable bit	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	**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:

R = Readable bit
W = Writable bit
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:			
R = Readable bit	W = Writable bit	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 **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` = RB31 buffer
`11110` = RB30 buffer
. . . .
`00001` = TRB1 buffer
`00000` = TRB0 buffer

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

bit 15-14	**Unimplemented:** Read as ‘0’
bit 13	**TXBO:** Transmitter in Error State Bus Off bit
bit 12	**TXBP:** Transmitter in Error State Bus Passive bit
bit 11	**RXBP:** Receiver in Error State Bus Passive bit
bit 10	**TXWAR:** Transmitter in Error State Warning bit
bit 9	**RXWAR:** Receiver in Error State Warning bit
bit 8	**EWARN:** Transmitter or Receiver in Error State Warning bit
bit 7	**IVRIF:** Invalid Message Received Interrupt Flag bit
bit 6	**WAKIF:** Bus Wake-up Activity Interrupt Flag bit
bit 5	**ERRIF:** Error Interrupt Flag bit (multiple sources in CiINTF<13:8> register)
bit 4	**Unimplemented:** Read as ‘0’
bit 3	**FIFOIF:** FIFO Almost Full Interrupt Flag bit
bit 2	**RBOVIF:** RX Buffer Overflow Interrupt Flag bit
bit 1	**RBIF:** RX Buffer Interrupt Flag bit
bit 0	**TBIF:** TX Buffer Interrupt Flag bit

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	U-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:

R = Readable bit

W = Writable bit

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
bit 6	**WAKIE:** Bus Wake-up Activity Interrupt Flag bit
bit 5	**ERRIE:** Error Interrupt Enable bit
bit 4	**Unimplemented:** Read as ‘0’
bit 3	**FIFOIE:** FIFO Almost Full Interrupt Enable bit
bit 2	**RBOVIE:** RX Buffer Overflow Interrupt Enable bit
bit 1	**RBIE:** RX Buffer Interrupt Enable bit
bit 0	**TBIE:** TX Buffer Interrupt Enable bit

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:			
R = Readable bit	W = Writable bit	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:

R = Readable bit W = Writable bit 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 7-6 **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/F_{CAN}
•
•
•
`00 0010` = T_Q = 2 x 3 x 1/F_{CAN}
`00 0001` = T_Q = 2 x 2 x 1/F_{CAN}
`00 0000` = T_Q = 2 x 1 x 1/F_{CAN}

REGISTER 21-10: CiCFG2: 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 Buffer 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 Buffer 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:			
R = Readable bit	W = Writable bit	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: CiBUFPNT1: 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:			
R = Readable bit	W = Writable bit	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 Written when Filter 3 Hits bits
bit 11-8	**F2BP<3:0>:** RX Buffer Written when Filter 2 Hits bits
bit 7-4	**F1BP<3:0>:** RX Buffer Written when Filter 1 Hits bits
bit 3-0	**F0BP<3:0>:** RX Buffer Written when Filter 0 Hits bits `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

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:

R = Readable bit	W = Writable bit	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 Written when Filter 7 Hits bits
bit 11-8	**F6BP<3:0>:** RX Buffer Written when Filter 6 Hits bits
bit 7-4	**F5BP<3:0>:** RX Buffer Written when Filter 5 Hits bits
bit 3-0	**F4BP<3:0>:** RX Buffer Written when Filter 4 Hits bits

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:

R = Readable bit	W = Writable bit	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 Written when Filter 11 Hits bits
bit 11-8	**F10BP<3:0>:** RX Buffer Written when Filter 10 Hits bits
bit 7-4	**F9BP<3:0>:** RX Buffer Written when Filter 9 Hits bits
bit 3-0	**F8BP<3:0>:** RX Buffer Written when Filter 8 Hits bits

REGISTER 21-15: CiBUFPNT4: 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:			
R = Readable bit	W = Writable bit	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 Written when Filter 15 Hits bits
bit 11-8	**F14BP<3:0>:** RX Buffer Written when Filter 14 Hits bits
bit 7-4	**F13BP<3:0>:** RX Buffer Written when Filter 13 Hits bits
bit 3-0	**F12BP<3:0>:** RX Buffer Written when Filter 12 Hits bits

REGISTER 21-16: CiRXFnSID: ECAN™ ACCEPTANCE FILTER n STANDARD IDENTIFIER (n = 0, 1, ...,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:
R = Readable bit W = Writable bit 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 n EXTENDED IDENTIFIER (n = 0, 1, ..., 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:
R = Readable bit W = Writable bit 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:

R = Readable bit	W = Writable bit	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
bit 13-12	**F6MSK<1:0>:** Mask Source for Filter 6 bit
bit 11-10	**F5MSK<1:0>:** Mask Source for Filter 5 bit
bit 9-8	**F4MSK<1:0>:** Mask Source for Filter 4 bit
bit 7-6	**F3MSK<1:0>:** Mask Source for Filter 3 bit
bit 5-4	**F2MSK<1:0>:** Mask Source for Filter 2 bit
bit 3-2	**F1MSK<1:0>:** Mask Source for Filter 1 bit
bit 1-0	**F0MSK<1:0>:** Mask Source for Filter 0 bit `11` = Reserved `10` = Acceptance Mask 2 registers contain mask `01` = Acceptance Mask 1 registers contain mask `00` = Acceptance Mask 0 registers contain mask

REGISTER 21-19: CiFMSKSEL2: 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:

R = Readable bit W = Writable bit 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` = Reserved `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 n STANDARD IDENTIFIER

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:			
R = Readable bit	W = Writable bit	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` = 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 n EXTENDED IDENTIFIER

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

bit 15-0	**RXFUL<15:0>:** Receive Buffer n Full bits `1` = Buffer is full (set by module) `0` = Buffer is empty (clear by application software)

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

bit 15-0	**RXFUL<31:16>:** Receive Buffer n Full bits `1` = Buffer is full (set by module) `0` = Buffer is empty (clear by application software)

REGISTER 21-24: CiRXOVF1: 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 = Clear only bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0	**RXOVF<15:0>:** Receive Buffer n Overflow bits `1` = Module pointed a write to a full buffer (set by module) `0` = Overflow is cleared (clear by application software)

REGISTER 21-25: CiRXOVF2: 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 = Clear only bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0	**RXOVF<31:16>:** Receive Buffer n Overflow bits `1` = Module pointed a write to a full buffer (set by module) `0` = Overflow is cleared (clear by application software)

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:			
R = Readable bit	W = Writable bit	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
Setting this bit to ‘`1`’ requests sending a message. The bit will automatically clear when the message is successfully sent. Clearing the bit to ‘`0`’ while set will request 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.

REGISTER 21-27: CiTRBnSID: ECAN™ BUFFER n STANDARD IDENTIFIER (n = 0, 1, ..., 31)

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

REGISTER 21-28: CiTRBnEID: ECAN™ BUFFER n EXTENDED IDENTIFIER (n = 0, 1, ..., 31)

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

REGISTER 21-29: CiTRBnDLC: ECAN™ BUFFER n DATA LENGTH CONTROL (n = 0, 1, ..., 31)

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-0	U-0	U-0	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

REGISTER 21-30: CiTRBnDm: ECAN™ BUFFER n DATA FIELD BYTE m (n = 0, 1, ..., 31; m = 0, 1, ..., 7)[(1)]

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
TRBnDm7	TRBnDm6	TRBnDm5	TRBnDm4	TRBnDm3	TRBnDm2	TRBnDm1	TRBnDm0
bit 7							bit 0

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

bit 7-0	**TRBnDm<7:0>:** Data Field Buffer ‘n’ Byte ‘m’ bits

Note 1: The Most Significant Byte contains byte (m + 1) of the buffer.

REGISTER 21-31: CiTRBnSTAT: ECAN™ RECEIVE BUFFER n STATUS (n = 0, 1, ..., 31)

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	FILHIT4	FILHIT3	FILHIT2	FILHIT1	FILHIT0
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 (only written by module for receive buffers, unused for transmit buffers) Encodes number of filter that resulted in writing this buffer.
bit 7-0	**Unimplemented:** Read as ‘0’

22.0 10-BIT/12-BIT ANALOG-TO-DIGITAL CONVERTER (ADC)

Note:	This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 16. "Analog-to-Digital Converter (ADC)"** (DS70183) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

The dsPIC33FJXXXMCX06/X08/X10 devices have up to 32 ADC input channels. These devices also have up to 2 ADC modules (ADCx, where 'x' = 1 or 2), each with its own set of Special Function Registers.

The AD12B bit (ADxCON1<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 32 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 1 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 32 analog input pins, designated AN0 through AN31. In addition, there are two analog input pins for external voltage reference connections. These voltage reference inputs may be shared with other analog input pins. The actual number of analog input pins and external voltage reference input configuration will depend on the specific device. Refer to the device data sheet for further details.

A block diagram of the ADC is shown in Figure 22-1.

22.2 ADC Initialization

The following configuration steps should be performed.

1. Configure the ADC module:
    a) Select port pins as analog inputs (ADxPCFGH<15:0> or ADxPCFGL<15:0>)
    b) Select voltage reference source to match expected range on analog inputs (ADxCON2<15:13>)
    c) Select the analog conversion clock to match desired data rate with processor clock (ADxCON3<7:0>)
    d) Determine how many S/H channels will be used (ADxCON2<9:8> and ADxPCFGH<15:0> or ADxPCFGL<15:0>)
    e) Select the appropriate sample/conversion sequence (ADxCON1<7:5> and ADxCON3<12:8>)
    f) Select how conversion results are presented in the buffer (ADxCON1<9:8>)
    g) Turn on ADC module (ADxCON1<15>)
2. Configure ADC interrupt (if required):
    a) Clear the ADxIF bit
    b) Select ADC interrupt priority

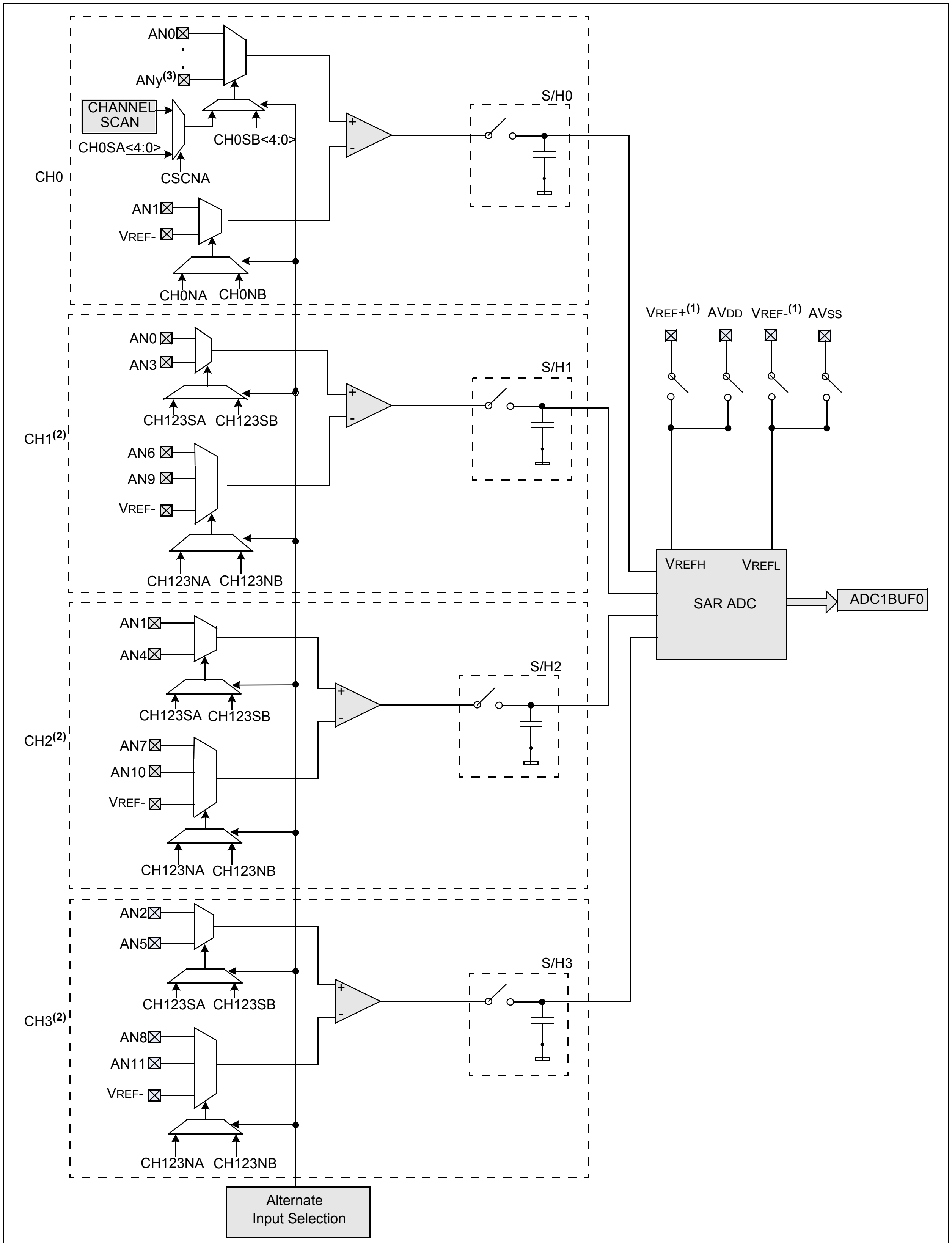
22.3 ADC and DMA

If more than one conversion result needs to be buffered before triggering an interrupt, DMA data transfers can be used. Both ADC1 and ADC2 can trigger a DMA data transfer. If ADC1 or ADC2 is selected as the DMA IRQ source, a DMA transfer occurs when the AD1IF or AD2IF bit gets set as a result of an ADC1 or ADC2 sample conversion sequence.

The SMPI<3:0> bits (ADxCON2<5:2>) are used to select how often the DMA RAM buffer pointer is incremented.

The ADDMABM bit (ADxCON1<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 will provide 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 will provide 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: ADC1 MODULE BLOCK DIAGRAM

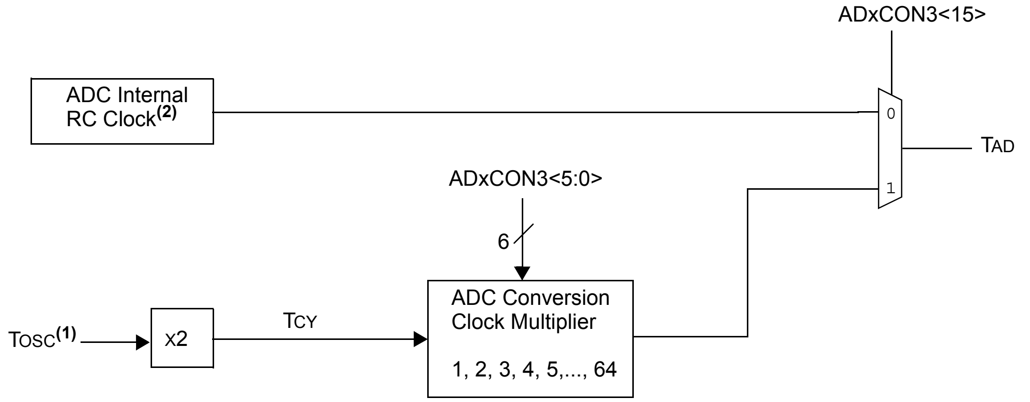


Note 1: VREF+, VREF- inputs can be multiplexed with other analog inputs.

2: Channels 1, 2 and 3 are not applicable for the 12-bit mode of operation.

3: For 64-pin devices, y = 15; for 80-pin devices, y = 17; for 100-pin devices, y = 23; for ADC2, y = 15.

FIGURE 22-2: ADC CONVERSION CLOCK PERIOD BLOCK DIAGRAM



Note 1: Refer to Figure 9-2 for the derivation of FOSC when the PLL is enabled. If the PLL is not used, FOSC is equal to the clock source frequency. TOSC = 1/FOSC.

2: See the ADC electrical specifications for the exact RC clock value.

REGISTER 22-1: ADxCON1: ADCx CONTROL REGISTER 1 (where x = 1 or 2)

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 will provide 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 will provide 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` = Reserved
`100` = Reserved
`011` = MPWM interval ends sampling and starts conversion
`010` = GP timer (Timer3 for ADC1, Timer5 for ADC2) 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: ADxCON1: ADCx CONTROL REGISTER 1 (where x = 1 or 2) (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 may write '1' to begin sampling. Automatically set by hardware if ASAM = 1. If SSRC = 000, software may 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 may write '0' to clear DONE status (software not allowed to write '1'). Clearing this bit will NOT affect any operation in progress. Automatically cleared by hardware at start of a new conversion.

REGISTER 22-2: ADxCON2: ADCx CONTROL REGISTER 2 (where x = 1 or 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

	VREF**+**	VREF**-**
`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 second half of buffer, user should access data in the first half
`0` = ADC is currently filling first half of buffer, user should access data in the second half

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 or generates interrupt after completion of every 2nd sample/conversion operation
`0000` = Increments the DMA address or generates interrupt after completion of every sample/conversion operation

bit 1 **BUFM:** Buffer Fill Mode Select bit

`1` = Starts filling first half of buffer on first interrupt and the second half of buffer on next interrupt
`0` = Always starts filling buffer from the beginning

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: ADxCON3: ADCx 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 = TCY · (ADCS<7:0> + 1) = 64 · TCY = TAD
•
•
•
00000010 = TCY · (ADCS<7:0> + 1) = 3 · TCY = TAD
00000001 = TCY · (ADCS<7:0> + 1) = 2 · TCY = TAD
00000000 = TCY · (ADCS<7:0> + 1) = 1 · TCY = TAD

Note 1: This bit only used if ADxCON1<SSRC> = 1.
2: This bit is not used if ADxCON3<ADRC> = 1.

REGISTER 22-4: ADxCON4: ADCx 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: ADxCHS123: ADCx 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 **When AD12B = `1`, CHxNB is: U-0, Unimplemented, Read as ‘`0`’** `11` = CH1 negative input is AN9, CH2 negative input is AN10, CH3 negative input is AN11 `10` = CH1 negative input is AN6, CH2 negative input is AN7, CH3 negative input is AN8 `0x` = CH1, CH2, CH3 negative input is VREF-
bit 8	**CH123SB:** Channel 1, 2, 3 Positive Input Select for Sample B bit **When AD12B = `1`, CHxSB is: U-0, Unimplemented, Read as ‘`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`’
bit 2-1	**CH123NA<1:0>:** Channel 1, 2, 3 Negative Input Select for Sample A bits **When AD12B = `1`, CHxNA is: U-0, Unimplemented, Read as ‘`0`’** `11` = CH1 negative input is AN9, CH2 negative input is AN10, CH3 negative input is AN11 `10` = CH1 negative input is AN6, CH2 negative input is AN7, CH3 negative input is AN8 `0x` = CH1, CH2, CH3 negative input is VREF-
bit 0	**CH123SA:** Channel 1, 2, 3 Positive Input Select for Sample A bit **When AD12B = `1`, CHxSA is: U-0, Unimplemented, Read as ‘`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: ADxCHS0: ADCx 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 Same definition as bit 7.
bit 14-13	**Unimplemented:** Read as ‘0’
bit 12-8	**CH0SB<4:0>:** Channel 0 Positive Input Select for Sample B bits Same definition as bit<4:0>.
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’
bit 4-0	**CH0SA<4:0>:** Channel 0 Positive Input Select for Sample A bits 11111 = Channel 0 positive input is AN31 11110 = Channel 0 positive input is AN30 • • • 00010 = Channel 0 positive input is AN2 00001 = Channel 0 positive input is AN1 00000 = Channel 0 positive input is AN0

Note: ADC2 can only select AN0-AN15 as positive inputs.

REGISTER 22-7: ADxCSSH: ADCx INPUT SCAN SELECT REGISTER HIGH(1,2)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSS31	CSS30	CSS29	CSS28	CSS27	CSS26	CSS25	CSS24
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
CSS23	CSS22	CSS21	CSS20	CSS19	CSS18	CSS17	CSS16
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	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 **CSS<31:16>:** ADC Input Scan Selection bits
`1` = Select ANx for input scan
`0` = Skip ANx for input scan

Note 1: On devices without 32 analog inputs, all ADxCSSH bits may be selected by user. However, inputs selected for scan without a corresponding input on device will convert VREFL.

2: CSSx = ANx, where x = 16 through 31.

REGISTER 22-8: ADxCSSL: ADCx INPUT SCAN SELECT REGISTER LOW(1,2)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSS15	CSS14	CSS13	CSS12	CSS11	CSS10	CSS9	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 15-0 **CSS<15:0>:** ADC Input Scan Selection bits
`1` = Select ANx for input scan
`0` = Skip ANx for input scan

Note 1: On devices without 16 analog inputs, all ADxCSSL bits may be selected by user. However, inputs selected for scan without a corresponding input on device will convert VREFL.

2: CSSx = ANx, where x = 0 through 15.

REGISTER 22-9: AD1PCFGH: ADC1 PORT CONFIGURATION REGISTER HIGH(1,2,3)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG31	PCFG30	PCFG29	PCFG28	PCFG27	PCFG26	PCFG25	PCFG24
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
PCFG23	PCFG22	PCFG21	PCFG20	PCFG19	PCFG18	PCFG17	PCFG16
bit 7							bit 0

Legend:

R = Readable bit
W = Writable bit
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 **PCFG<31:16>:** ADC Port Configuration Control bits

`1` = Port pin in Digital mode, port read input enabled, ADC input multiplexer connected to AVSS
`0` = Port pin in Analog mode, port read input disabled, ADC samples pin voltage

Note 1: On devices without 32 analog inputs, all PCFG bits are R/W by user. However, PCFG bits are ignored on ports without a corresponding input on device.

2: ADC2 only supports analog inputs AN0-AN15; therefore, no ADC2 port Configuration register exists.

3: PCFGx = ANx, where x = 16 through 31.

REGISTER 22-10: ADxPCFGL: ADCx PORT CONFIGURATION REGISTER LOW(1,2,3)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	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 15-0 **PCFG<15:0>:** ADC Port Configuration Control bits

`1` = Port pin in Digital mode, port read input enabled, ADC input multiplexer connected to AVSS
`0` = Port pin in Analog mode, port read input disabled, ADC samples pin voltage

Note 1: On devices without 16 analog inputs, all PCFG bits are R/W by user. However, PCFG bits are ignored on ports without a corresponding input on device.

2: On devices with two analog-to-digital modules, both AD1PCFGL and AD2PCFGL will affect the configuration of port pins multiplexed with AN0-AN15.

3: PCFGx = ANx, where x = 0 through 15.

23.0 SPECIAL FEATURES

> **Note:** This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 23. "CodeGuard™ Security"** (DS70199), **Section 24. "Programming and Diagnostics"** (DS70207), and **Section 25. "Device Configuration"** (DS70194) in the *"dsPIC33F Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

dsPIC33FJXXXMCX06/X08/X10 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

23.1 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 device Configuration register map is shown in Table 23-1.

The individual Configuration bit descriptions for the FBS, FSS, FGS, FOSCSEL, FOSC, FWDT, FPOR and FICD Configuration registers are shown in Table 23-2.

Note that address 0xF80000 is beyond the user program memory space. In fact, it belongs to the configuration memory space (0x800000-0xFFFFFF) 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.

TABLE 23-1: 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	RSS<1:0>		—	—	SSS<2:0>			SWRP
0xF80004	FGS	—	—	—	—	—	GSS1	GSS0	GWRP
0xF80006	FOSCSEL	IESO	Reserved(2)	—	—	—	FNOSC<2:0>		
0xF80008	FOSC	FCKSM<1:0>		—	—	—	OSCIOFNC	POSCMD<1:0>	
0xF8000A	FWDT	FWDTEN	WINDIS	—	WDTPRE	WDTPOST<3:0>			
0xF8000C	FPOR	PWMPIN	HPOL	LPOL	—	—	FPWRT<2:0>		
0xF8000E	FICD	Reserved(1)		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: When read, these bits will appear as '`1`'. When you write to these bits, set these bits to '`1`'.

2: When read, this bit returns the current programmed value.

TABLE 23-2: dsPIC33FJXXXMCX06/X08/X10 CONFIGURATION BITS DESCRIPTION

Bit Field	Register	Description
BWRP	FBS	Boot Segment Program Flash Write Protection `1` = Boot segment may 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 IW less VS `110` = Standard security; boot program Flash segment starts at End of VS, ends at 0007FEh `010` = High security; boot program Flash segment starts at End of VS, ends at 0007FEh Boot space is 4K IW less VS `101` = Standard security; boot program Flash segment starts at End of VS, ends at 001FFEh `001` = High security; boot program Flash segment starts at End of VS, ends at 001FFEh Boot space is 8K IW less VS `100` = Standard security; boot program Flash segment starts at End of VS, ends at 003FFEh `000` = High security; boot program Flash segment starts at End of VS, ends at 003FFEh
RBS<1:0>	FBS	Boot Segment RAM Code Protection `11` = No Boot RAM defined `10` = Boot RAM is 128 Bytes `01` = Boot RAM is 256 Bytes `00` = Boot RAM is 1024 Bytes
SWRP	FSS	Secure Segment Program Flash Write Protection `1` = Secure segment may be written `0` = Secure segment is write-protected

TABLE 23-2: dsPIC33FJXXXMCX06/X08/X10 CONFIGURATION BITS DESCRIPTION (CONTINUED)

Bit Field	Register	Description
SSS<2:0>	FSS	Secure Segment Program Flash Code Protection Size **(FOR 128K and 256K DEVICES)** `X11` = No Secure program Flash segment Secure space is 8K IW less BS `110` = Standard security; secure program Flash segment starts at End of BS, ends at 0x003FFE `010` = High security; secure program Flash segment starts at End of BS, ends at 0x003FFE Secure space is 16K IW less BS `101` = Standard security; secure program Flash segment starts at End of BS, ends at 0x007FFE `001` = High security; secure program Flash segment starts at End of BS, ends at 0x007FFE Secure space is 32K IW less BS `100` = Standard security; secure program Flash segment starts at End of BS, ends at 0x00FFFE `000` = High security; secure program Flash segment starts at End of BS, ends at 0x00FFFE **(FOR 64K 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>	FSS	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
GSS<1:0>	FGS	General Segment Code-Protect bit `11` = User program memory is not code-protected `10` = Standard security; general program Flash segment starts at End of SS, ends at EOM `0x` = High security; general program Flash segment starts at End of SS, ends at EOM

TABLE 23-2: dsPIC33FJXXXMCX06/X08/X10 CONFIGURATION BITS DESCRIPTION (CONTINUED)

Bit Field	Register	Description
GWRP	FGS	General Segment Write-Protect bit `1` = User program memory is not write-protected `0` = User program memory is write-protected
IESO	FOSCSEL	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>	FOSCSEL	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
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 will have 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
WDTPOST	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)

TABLE 23-2: dsPIC33FJXXXMCX06/X08/X10 CONFIGURATION BITS DESCRIPTION (CONTINUED)

Bit Field	Register	Description
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
JTAGEN	FICD	JTAG Enable bits `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

23.2 On-Chip Voltage Regulator

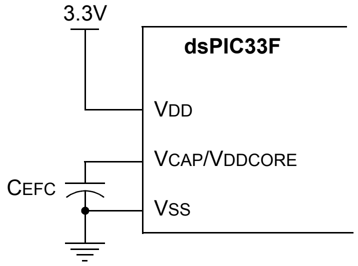
All of the dsPIC33FJXXXMCX06/X08/X10 devices power their core digital logic at a nominal 2.5V. This may create an issue for designs that are required to operate at a higher typical voltage, such as 3.3V. To simplify system design, all devices in the dsPIC33FJXXXMCX06/X08/X10 family incorporate an on-chip regulator that allows the device to run its core logic from VDD.

The regulator provides power to the core from the other VDD pins. The regulator requires that a low-ESR (less than 5 ohms) capacitor (such as tantalum or ceramic) be connected to the VCAP/VDDCORE pin (Figure 23-1). This helps to maintain the stability of the regulator. The recommended value for the filter capacitor is provided in Table 26-13 of **Section 26.1 "DC 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 µs 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 any power-down.

FIGURE 23-1: CONNECTIONS FOR THE ON-CHIP VOLTAGE REGULATOR(1)



Note 1: These are typical operating voltages. Refer to **TABLE 26-13: "Internal Voltage Regulator Specifications"** located in **Section 26.1 "DC Characteristics"** for the full operating ranges of VDD and VCAP/VDDCORE.

2: It is important for the low-ESR capacitor to be placed as close as possible to the VCAP/VDDCORE pin.

23.3 BOR: Brown-Out Reset

The BOR (Brown-out Reset) 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 (i.e., 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 will generate a Reset pulse which will reset the device. The BOR will select the clock source, based on the device Configuration bit values (FNOSC<2:0> and POSCMD<1:0>). Furthermore, if an oscillator mode is selected, the BOR will activate the Oscillator Start-up Timer (OST). The system clock is held until OST expires. If the PLL is used, then the clock will be held until the LOCK bit (OSCCON<5>) is '1'.

Concurrently, the PWRT time-out (TPWRT) will be 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>) will be set to indicate that a BOR has occurred. The BOR circuit continues to operate while in Sleep or Idle modes and will reset the device should VDD fall below the BOR threshold voltage.

23.4 Watchdog Timer (WDT)

For dsPIC33FJXXXMCX06/X08/X10 devices, the WDT is driven by the LPRC oscillator. When the WDT is enabled, the clock source is also enabled.

The nominal WDT clock source from LPRC is 32 kHz. This feeds a prescaler than 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 (TWDT) of 1 ms in 5-bit mode, or 4 ms in 7-bit mode.

A variable postscaler divides down the WDT prescaler output and allows for a wide range of time-out periods. The postscaler is controlled by the WDTPOST<3:0> Configuration bits (FWDT<3:0>) which allow the selection of a total of 16 settings, from 1:1 to 1:32,768. Using the prescaler and postscaler, time-out periods ranging from 1 ms to 131 seconds can be achieved.

The WDT, prescaler and postscaler are reset:

- On any device Reset
- On the completion of a clock switch, whether invoked by software (i.e., setting the OSWEN bit after changing the NOSC bits) or by hardware (i.e., Fail-Safe Clock Monitor)
- When a `PWRSAV` instruction is executed (i.e., Sleep or Idle mode is entered)
- When the device exits Sleep or Idle mode to resume normal operation
- By a `CLRWDT` instruction during normal execution

If the WDT is enabled, it will continue to run during Sleep or Idle modes. When the WDT time-out occurs, the device will wake the device and code execution will continue from where the `PWRSAV` instruction was executed. The corresponding SLEEP or IDLE bits (RCON<3,2>) will need to be cleared in software after the device wakes up.

The WDT flag bit, WDTO (RCON<4>), is not automatically cleared following a WDT time-out. To detect subsequent WDT events, the flag must be cleared in software.

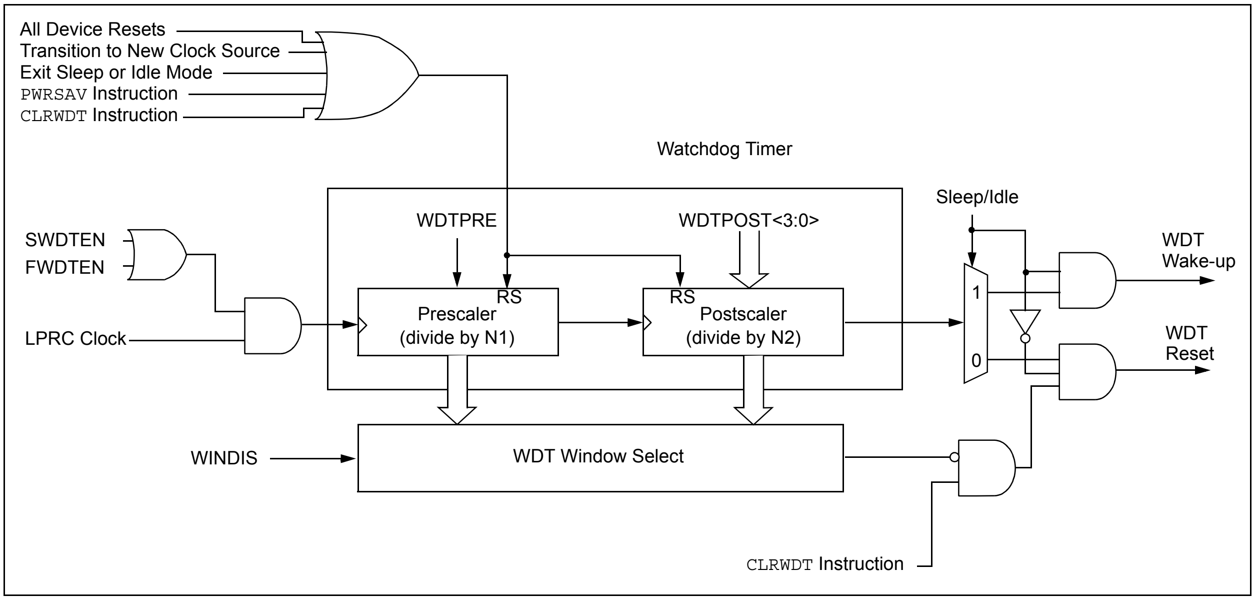
> **Note:** The `CLRWDT` and `PWRSAV` instructions clear the prescaler and postscaler counts when executed.

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 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.

FIGURE 23-2: WDT BLOCK DIAGRAM



23.5 JTAG Interface

dsPIC33FJXXXMCX06/X08/X10 devices implement a JTAG interface, which supports boundary scan device testing, as well as in-circuit programming. Detailed information on the interface will be provided in future revisions of the document.

23.6 Code Protection and CodeGuard™ Security

The dsPIC33FJXXXMCX06/X08/X10 devices offer the advanced implementation of CodeGuard™ Security. 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 IP are resident on the single chip. The code protection features vary depending on the actual device implemented. The following sections provide an overview of these features.

The code protection features are controlled by the Configuration registers: FBS, FSS and FGS.

Note: Refer to **Section 23. "CodeGuard™ Security"** (DS70199) in the *"dsPIC33F Family Reference Manual"* for further information on usage, configuration and operation of CodeGuard Security.

23.7 In-Circuit Serial Programming

dsPIC33FJXXXMCX06/X08/X10 family digital signal controllers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data and three other lines for power, ground and the programming sequence. 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. Please refer to the "*dsPIC33F/PIC24H Flash Programming Specification"* (DS70152) document for details about ICSP.

Any one out of three pairs of programming clock/data pins may be used:

- PGEC1 and PGED1
- PGEC2 and PGED2
- PGEC3 and PGED3

23.8 In-Circuit Debugger

When MPLAB® ICD 2 is selected as a debugger, the in-circuit debugging functionality is enabled. This function allows simple debugging functions when used with MPLAB IDE. Debugging functionality is controlled through the PGECx (Emulation/Debug Clock) and PGEDx (Emulation/Debug Data) pin functions.

Any one out of three pairs of debugging clock/data pins may 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 $\overline{\text{MCLR}}$, VDD, VSS 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.

24.0 INSTRUCTION SET SUMMARY

Note: This data sheet summarizes the features of the dsPIC33FJXXXMCX06/X08/X10 family of devices. However, 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 web site (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 24-1 shows the general symbols used in describing the instructions.

The dsPIC33F instruction set summary in Table 24-2 lists all the instructions, along with the status flags affected by each instruction.

Most word or byte-oriented W register instructions (including barrel shift instructions) have three operands:

- The first source operand which is typically a register 'Wb' without any address modifier
- The second source operand which is typically a register 'Ws' with or without an address modifier
- The destination of the result which is typically a register 'Wd' with or without an address modifier

However, word or byte-oriented file register instructions have two operands:

- The file register specified by the value 'f'
- The destination, which could either be the file register 'f' or the W0 register, which is denoted as 'WREG'

Most bit-oriented instructions (including simple rotate/shift instructions) have two operands:

- The W register (with or without an address modifier) or file register (specified by the value of 'Ws' or 'f')
- The bit in the W register or file register (specified by a literal value or indirectly by the contents of register 'Wb')

The literal instructions that involve data movement may use some of the following operands:

- A literal value to be loaded into a W register or file register (specified by the value of 'k')
- The W register or file register where the literal value is to be loaded (specified by 'Wb' or 'f')

However, literal instructions that involve arithmetic or logical operations use some of the following operands:

- The first source operand which is a register 'Wb' without any address modifier
- The second source operand which is a literal value
- The destination of the result (only if not the same as the first source operand) which is typically a register 'Wd' with or without an address modifier

The `MAC` class of DSP instructions may 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 may 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 may use some of the following operands:

- A program memory address
- The mode of the table read and table write instructions

All instructions are a single word, except for certain double-word instructions, which were made double-word instructions so that all the required information is available in these 48 bits. In the second word, the 8 MSbs are '`0`'s. If this second word is executed as an instruction (by itself), it will execute as a `NOP`.

Most single-word instructions are executed in a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles with the additional instruction cycle(s) executed as a `NOP`. Notable exceptions are the `BRA` (unconditional/computed branch), indirect `CALL/GOTO`, all table reads and writes and `RETURN/RETFIE` instructions, which are single-word instructions but take two or three cycles. Certain instructions that involve skipping over the subsequent instruction require either two or three cycles if the skip is performed, depending on whether the instruction being skipped is a single-word or two-word instruction. Moreover, double-word moves require two cycles. The double-word instructions execute in two instruction cycles.

Note: For more details on the instruction set, refer to the *"dsPIC30F/33F Programmer's Reference Manual"* (DS70157).

TABLE 24-1: 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 ∈ {W13, [W13]+ = 2}
bit4	4-bit bit selection field (used in word addressed instructions) ∈ {0...15}
C, DC, N, OV, Z	MCU Status bits: Carry, Digit Carry, Negative, Overflow, Sticky Zero
Expr	Absolute address, label or expression (resolved by the linker)
f	File register address ∈ {0x0000...0x1FFF}
lit1	1-bit unsigned literal ∈ {0,1}
lit4	4-bit unsigned literal ∈ {0...15}
lit5	5-bit unsigned literal ∈ {0...31}
lit8	8-bit unsigned literal ∈ {0...255}
lit10	10-bit unsigned literal ∈ {0...255} for Byte mode, {0:1023} for Word mode
lit14	14-bit unsigned literal ∈ {0...16384}
lit16	16-bit unsigned literal ∈ {0...65535}
lit23	23-bit unsigned literal ∈ {0...8388608}; LSb must be '`0`'
None	Field does not require an entry, may 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 ∈ {-512...511}
Slit16	16-bit signed literal ∈ {-32768...32767}
Slit6	6-bit signed literal ∈ {-16...16}
Wb	Base W register ∈ {W0..W15}
Wd	Destination W register ∈ { Wd, [Wd], [Wd++], [Wd--], [++Wd], [--Wd] }
Wdo	Destination W register ∈ { Wnd, [Wnd], [Wnd++], [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb] }
Wm,Wn	Dividend, Divisor working register pair (direct addressing)
Wm*Wm	Multiplicand and Multiplier working register pair for Square instructions ∈ {W4 * W4,W5 * W5,W6 * W6,W7 * W7}

TABLE 24-1: SYMBOLS USED IN OPCODE DESCRIPTIONS (CONTINUED)

Field	Description
Wm*Wn	Multiplicand and Multiplier working register pair for DSP instructions ∈ {W4 * W5,W4 * W6,W4 * W7,W5 * W6,W5 * W7,W6 * W7}
Wn	One of 16 working registers ∈ {W0..W15}
Wnd	One of 16 destination working registers ∈ {W0...W15}
Wns	One of 16 source working registers ∈ {W0...W15}
WREG	W0 (working register used in file register instructions)
Ws	Source W register ∈ { Ws, [Ws], [Ws++], [Ws--], [++Ws], [--Ws] }
Wso	Source W register ∈ { Wns, [Wns], [Wns++], [Wns--], [++Wns], [--Wns], [Wns+Wb] }
Wx	X data space prefetch address register for DSP instructions ∈ {[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], none}
Wxd	X data space prefetch destination register for DSP instructions ∈ {W4...W7}
Wy	Y data space prefetch address register for DSP instructions ∈ {[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], none}
Wyd	Y data space prefetch destination register for DSP instructions ∈ {W4...W7}

TABLE 24-2: 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 + WREG	1	1	C,DC,N,OV,Z
		`ADD f,WREG`	WREG = f + WREG	1	1	C,DC,N,OV,Z
		`ADD #lit10,Wn`	Wd = lit10 + Wd	1	1	C,DC,N,OV,Z
		`ADD Wb,Ws,Wd`	Wd = Wb + Ws	1	1	C,DC,N,OV,Z
		`ADD Wb,#lit5,Wd`	Wd = Wb + lit5	1	1	C,DC,N,OV,Z
		`ADD Wso,#Slit4,Acc`	16-bit Signed Add to Accumulator	1	1	OA,OB,SA,SB
2	`ADDC`	`ADDC f`	f = f + WREG + (C)	1	1	C,DC,N,OV,Z
		`ADDC f,WREG`	WREG = f + WREG + (C)	1	1	C,DC,N,OV,Z
		`ADDC #lit10,Wn`	Wd = lit10 + Wd + (C)	1	1	C,DC,N,OV,Z
		`ADDC Wb,Ws,Wd`	Wd = Wb + Ws + (C)	1	1	C,DC,N,OV,Z
		`ADDC Wb,#lit5,Wd`	Wd = Wb + lit5 + (C)	1	1	C,DC,N,OV,Z
3	`AND`	`AND f`	f = f .AND. WREG	1	1	N,Z
		`AND f,WREG`	WREG = f .AND. WREG	1	1	N,Z
		`AND #lit10,Wn`	Wd = lit10 .AND. Wd	1	1	N,Z
		`AND Wb,Ws,Wd`	Wd = Wb .AND. Ws	1	1	N,Z
		`AND Wb,#lit5,Wd`	Wd = Wb .AND. lit5	1	1	N,Z
4	`ASR`	`ASR f`	f = Arithmetic Right Shift f	1	1	C,N,OV,Z
		`ASR f,WREG`	WREG = Arithmetic Right Shift f	1	1	C,N,OV,Z
		`ASR Ws,Wd`	Wd = Arithmetic Right Shift Ws	1	1	C,N,OV,Z
		`ASR Wb,Wns,Wnd`	Wnd = Arithmetic Right Shift Wb by Wns	1	1	N,Z
		`ASR Wb,#lit5,Wnd`	Wnd = Arithmetic Right Shift Wb by lit5	1	1	N,Z
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 24-2: 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 = $\overline{f}$	1	1	N,Z
		COM f,WREG	WREG = $\overline{f}$	1	1	N,Z
		COM Ws,Wd	Wd = $\overline{Ws}$	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 – $\overline{C}$)	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 24-2: 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 24-2: 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
		MUL f	W3:W2 = f * WREG	1	1	None
52	NEG	NEG Acc	Negate Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
		NEG f	f = $\overline{f}$ + 1	1	1	C,DC,N,OV,Z
		NEG f,WREG	WREG = $\overline{f}$ + 1	1	1	C,DC,N,OV,Z
		NEG Ws,Wd	Wd = $\overline{Ws}$ + 1	1	1	C,DC,N,OV,Z
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	PWRSAV	PWRSAV #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

TABLE 24-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
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
67	SAC	SAC Acc,#Slit4,Wdo	Store Accumulator	1	1	None
		SAC.R Acc,#Slit4,Wdo	Store Rounded Accumulator	1	1	None
68	SE	SE Ws,Wnd	Wnd = sign-extended Ws	1	1	C,N,Z
69	SETM	SETM f	f = 0xFFFF	1	1	None
		SETM WREG	WREG = 0xFFFF	1	1	None
		SETM Ws	Ws = 0xFFFF	1	1	None
70	SFTAC	SFTAC Acc,Wn	Arithmetic Shift Accumulator by (Wn)	1	1	OA,OB,OAB, SA,SB,SAB
		SFTAC Acc,#Slit6	Arithmetic Shift Accumulator by Slit6	1	1	OA,OB,OAB, SA,SB,SAB
71	SL	SL f	f = Left Shift f	1	1	C,N,OV,Z
		SL f,WREG	WREG = Left Shift f	1	1	C,N,OV,Z
		SL Ws,Wd	Wd = Left Shift Ws	1	1	C,N,OV,Z
		SL Wb,Wns,Wnd	Wnd = Left Shift Wb by Wns	1	1	N,Z
		SL Wb,#lit5,Wnd	Wnd = Left Shift Wb by lit5	1	1	N,Z
72	SUB	SUB Acc	Subtract Accumulators	1	1	OA,OB,OAB, SA,SB,SAB
		SUB f	f = f – WREG	1	1	C,DC,N,OV,Z
		SUB f,WREG	WREG = f – WREG	1	1	C,DC,N,OV,Z
		SUB #lit10,Wn	Wn = Wn – lit10	1	1	C,DC,N,OV,Z
		SUB Wb,Ws,Wd	Wd = Wb – Ws	1	1	C,DC,N,OV,Z
		SUB Wb,#lit5,Wd	Wd = Wb – lit5	1	1	C,DC,N,OV,Z
73	SUBB	SUBB f	f = f – WREG – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB f,WREG	WREG = f – WREG – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB #lit10,Wn	Wn = Wn – lit10 – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB Wb,Ws,Wd	Wd = Wb – Ws – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB Wb,#lit5,Wd	Wd = Wb – lit5 – ($\overline{C}$)	1	1	C,DC,N,OV,Z
74	SUBR	SUBR f	f = WREG – f	1	1	C,DC,N,OV,Z
		SUBR f,WREG	WREG = WREG – f	1	1	C,DC,N,OV,Z
		SUBR Wb,Ws,Wd	Wd = Ws – Wb	1	1	C,DC,N,OV,Z
		SUBR Wb,#lit5,Wd	Wd = lit5 – Wb	1	1	C,DC,N,OV,Z
75	SUBBR	SUBBR f	f = WREG – f – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR f,WREG	WREG = WREG – f – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR Wb,Ws,Wd	Wd = Ws – Wb – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR Wb,#lit5,Wd	Wd = lit5 – Wb – ($\overline{C}$)	1	1	C,DC,N,OV,Z
76	SWAP	SWAP.b Wn	Wn = nibble swap Wn	1	1	None
		SWAP Wn	Wn = byte swap Wn	1	1	None
77	TBLRDH	TBLRDH Ws,Wd	Read Prog<23:16> to Wd<7:0>	1	2	None
78	TBLRDL	TBLRDL Ws,Wd	Read Prog<15:0> to Wd	1	2	None
79	TBLWTH	TBLWTH Ws,Wd	Write Ws<7:0> to Prog<23:16>	1	2	None
80	TBLWTL	TBLWTL Ws,Wd	Write Ws to Prog<15:0>	1	2	None
81	ULNK	ULNK	Unlink Frame Pointer	1	1	None
82	XOR	XOR f	f = f .XOR. WREG	1	1	N,Z
		XOR f,WREG	WREG = f .XOR. WREG	1	1	N,Z
		XOR #lit10,Wn	Wd = lit10 .XOR. Wd	1	1	N,Z
		XOR Wb,Ws,Wd	Wd = Wb .XOR. Ws	1	1	N,Z
		XOR Wb,#lit5,Wd	Wd = Wb .XOR. lit5	1	1	N,Z
83	ZE	ZE Ws,Wnd	Wnd = Zero-extend Ws	1	1	C,Z,N

25.0 DEVELOPMENT SUPPORT

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

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

25.1 MPLAB Integrated Development Environment Software

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

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

The MPLAB IDE allows you to:

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

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

25.2 MPASM Assembler

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

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

The MPASM Assembler features include:

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

25.3 MPLAB C18 and MPLAB C30 C Compilers

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

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

25.4 MPLINK Object Linker/ MPLIB Object Librarian

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

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

The object linker/library features include:

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

25.5 MPLAB ASM30 Assembler, Linker and Librarian

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

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

25.6 MPLAB SIM Software Simulator

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

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

25.7 MPLAB ICE 2000 High-Performance In-Circuit Emulator

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

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

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

25.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC® Flash MCUs and dsPIC® Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

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

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

25.9 MPLAB ICD 2 In-Circuit Debugger

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

25.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} 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.

25.11 PICSTART Plus Development Programmer

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

25.12 PICkit 2 Development Programmer

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

25.13 Demonstration, Development and Evaluation Boards

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

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

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

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

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

26.0 ELECTRICAL CHARACTERISTICS

This section provides an overview of dsPIC33FJXXXMCX06/X08/X10 electrical characteristics. Additional information will be provided in future revisions of this document as it becomes available.

Absolute maximum ratings for the dsPIC33FJXXXMCX06/X08/X10 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 +85°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 26-2).

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.

26.1 DC Characteristics

TABLE 26-1: OPERATING MIPS VS. VOLTAGE

Characteristic	V_{DD} Range (in Volts)	Temp Range (in °C)	Max MIPS
			dsPIC33FJXXXMCX06/X08/X10
DC5	3.0-3.6V	-40°C to +85°C	40

TABLE 26-2: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min	Typ	Max	Unit
dsPIC33FJXXXMCX06/X08/X10					
Operating Junction Temperature Range	T_J	-40	—	+125	°C
Operating Ambient Temperature Range	T_A	-40	—	+85	°C
Power Dissipation: Internal chip power dissipation: $P_{INT} = V_{DD} \times (I_{DD} - \Sigma\ I_{OH})$ I/O Pin Power Dissipation: $I/O = \Sigma\ (\{V_{DD} - V_{OH}\} \times I_{OH}) + \Sigma\ (V_{OL} \times I_{OL})$	P_D	$P_{INT} + P_{I/O}$			W
Maximum Allowed Power Dissipation	P_{DMAX}	$(T_J - T_A)/\theta_{JA}$			W

TABLE 26-3: THERMAL PACKAGING CHARACTERISTICS

Characteristic	Symbol	Typ	Max	Unit	Notes
Package Thermal Resistance, 100-pin TQFP (14x14x1 mm)	θ_{JA}	40	—	°C/W	**1**
Package Thermal Resistance, 100-pin TQFP (12x12x1 mm)	θ_{JA}	40	—	°C/W	**1**
Package Thermal Resistance, 80-pin TQFP (12x12x1 mm)	θ_{JA}	40	—	°C/W	**1**
Package Thermal Resistance, 64-pin TQFP (10x10x1 mm)	θ_{JA}	40	—	°C/W	**1**

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

TABLE 26-4: DC TEMPERATURE AND VOLTAGE SPECIFICATIONS

DC CHARACTERISTICS			**Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
Operating Voltage							
DC10	**Supply Voltage**						
	VDD		3.0	—	3.6	V	—
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.

2: This is the limit to which VDD can be lowered without losing RAM data.

3: These parameters are characterized but not tested in manufacturing.

4: VDD voltage must remain at VSS for a minimum of 200 μs to ensure POR.

TABLE 26-5: DC CHARACTERISTICS: OPERATING CURRENT (IDD)

DC CHARACTERISTICS			**Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial			
Parameter No.	**Typical[(1)]**	**Max**	**Units**	**Conditions**		
Operating Current (IDD)(2)						
DC20d	27	30	mA	-40°C	3.3V	10 MIPS
DC20a	27	30	mA	+25°C		
DC20b	27	30	mA	+85°C		
DC21d	36	40	mA	-40°C	3.3V	16 MIPS
DC21a	37	40	mA	+25°C		
DC21b	38	45	mA	+85°C		
DC22d	43	50	mA	-40°C	3.3V	20 MIPS
DC22a	46	50	mA	+25°C		
DC22b	46	55	mA	+85°C		
DC23d	65	70	mA	-40°C	3.3V	30 MIPS
DC23a	65	70	mA	+25°C		
DC23b	65	70	mA	+85°C		
DC24d	84	90	mA	-40°C	3.3V	40 MIPS
DC24a	84	90	mA	+25°C		
DC24b	84	90	mA	+85°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. $\overline{\text{MCLR}}$ = 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 26-6: DC CHARACTERISTICS: IDLE CURRENT (I_{IDLE})

DC CHARACTERISTICS			**Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial			
Parameter No.	**Typical(1)**	**Max**	**Units**	**Conditions**		
Idle Current (I_{IDLE}): Core OFF Clock ON Base Current(2)						
DC40d	3	25	mA	-40°C	3.3V	10 MIPS
DC40a	3	25	mA	+25°C		
DC40b	3	25	mA	+85°C		
DC41d	4	25	mA	-40°C	3.3V	16 MIPS
DC41a	5	25	mA	+25°C		
DC41b	6	25	mA	+85°C		
DC42d	8	25	mA	-40°C	3.3V	20 MIPS
DC42a	9	25	mA	+25°C		
DC42b	10	25	mA	+85°C		
DC43a	15	25	mA	+25°C	3.3V	30 MIPS
DC43d	15	25	mA	-40°C		
DC43b	15	25	mA	+85°C		
DC44d	16	25	mA	-40°C	3.3V	40 MIPS
DC44a	16	25	mA	+25°C		
DC44b	16	25	mA	+85°C		

Note 1: Data in “Typical” column is at 3.3V, 25°C unless otherwise stated.

2: Base I_{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 V_{SS}.

TABLE 26-7: 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			
Parameter No.	**Typical(1)**	**Max**	**Units**	**Conditions**		
Power-Down Current (IPD)(2)						
DC60d	55	500	μA	-40°C	3.3V	Base Power-Down Current(3,4)
DC60a	211	500	μA	+25°C		
DC60b	244	500	μA	+85°C		
DC61d	8	13	μA	-40°C	3.3V	Watchdog Timer Current: ΔIWDT(3)
DC61a	10	15	μA	+25°C		
DC61b	12	20	μA	+85°C		

Note 1: Data in the Typical column is at 3.3V, 25°C unless otherwise stated.

2: Base IPD is measured with all peripherals and clocks shut down. All I/Os are configured as inputs and pulled to VSS. WDT, etc., are all switched off and VREGS (RCON<8>) = `1`.

3: The Δ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

4: These currents are measured on the device containing the most memory in this family.

TABLE 26-8: 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				
Parameter No.	**Typical(1)**	**Max**	**Doze Ratio**	**Units**	**Conditions**		
DC73a	11	35	1:2	mA	-40°C	3.3V	40 MIPS
DC73f	11	30	1:64	mA			
DC73g	11	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			

Note 1: Data in the Typical column is at 3.3V, 25°C unless otherwise stated.

TABLE 26-9: 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				
Param No.	Symbol	Characteristic	Min	Typ(1)	Max	Units	Conditions
	VIL	**Input Low Voltage**					
DI10		I/O pins	VSS	—	0.2 VDD	V	
DI15		$\overline{\text{MCLR}}$	VSS	—	0.2 VDD	V	
DI16		I/O Pins with OSC1 or SOSCI	VSS	—	0.2 VDD	V	
DI18		I/O Pins with I^{2}C	VSS	—	0.3 VDD	V	SMbus disabled
DI19		I/O Pins with I^{2}C	VSS	—	0.2 VDD	V	SMbus enabled
	VIH	**Input High Voltage**					
DI20		I/O Pins Not 5V Tolerant**(4)**	0.8 VDD	—	VDD	V	
		I/O Pins 5V Tolerant**(4)**	0.8 VDD	—	5.5	V	
		I/O Pins Not 5V Tolerant**(4)**	2	—	VDD	V	VDD = 3.3V
		I/O Pins 5V Tolerant**(4)**	2	—	5.5	V	VDD = 3.3V
DI26		I/O Pins with OSC1 or SOSCI	0.7 VDD	—	VDD	V	
DI28		I/O Pins with I^{2}C	0.7 VDD	—	5.5	V	SMbus disabled
DI29		I/O Pins with I^{2}C	0.8 VDD	—	5.5	V	SMbus enabled
	ICNPU	**CNx Pull-up Current**					
DI30			50	250	400	μA	VDD = 3.3V, VPIN = VSS
	IIL	**Input Leakage Current(2,3)**					
DI50		I/O Pins	—	—	±2	μA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance
DI51		I/O Pins Not 5V Tolerant**(4)**	—	—	±2	μA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance
DI51a		I/O Pins Not 5V Tolerant**(4)**	—	—	±2	μA	Shared with external reference pins
DI51b		I/O Pins Not 5V Tolerant**(4)**	—	—	±3.5	μA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance
DI51c		I/O Pins Not 5V Tolerant**(4)**	—	—	±8	μA	Analog pins shared with external reference pins
DI55		$\overline{\text{MCLR}}$	—	—	±2	μA	VSS ≤ VPIN ≤ VDD
DI56		OSC1	—	—	±2	μA	VSS ≤ VPIN ≤ VDD, 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 $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as current sourced by the pin.

4: See **"Pin Diagrams"** for a list of 5V tolerant pins.

TABLE 26-10: 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				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
	VOL	**Output Low Voltage**					
DO10		I/O ports	—	—	0.4	V	IOL = 2mA, VDD = 3.3V
DO16		OSC2/CLKO	—	—	0.4	V	IOL = 2mA, VDD = 3.3V
	VOH	**Output High Voltage**					
DO20		I/O ports	2.40	—	—	V	IOH = -2.3 mA, VDD = 3.3V
DO26		OSC2/CLKO	2.41	—	—	V	IOH = -1.3 mA, VDD = 3.3V

TABLE 26-11: 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				
Param No.	**Symbol**	**Characteristic**	**Min**(1)	**Typ**	**Max**(1)	**Units**	**Conditions**
BO10	VBOR	BOR Event on VDD transition high-to-low BOR event is tied to VDD core voltage decrease	2.40	—	2.55	V	—

Note 1: Parameters are for design guidance only and are not tested in manufacturing.

TABLE 26-12: 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				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
		Program Flash Memory					
D130a	EP	Cell Endurance	100	1000	—	E/W	See **Note 2**
D131	VPR	VDD for Read	VMIN	—	3.6	V	VMIN = Minimum operating voltage
D132B	VPEW	VDD for Self-Timed Write	VMIN	—	3.6	V	VMIN = Minimum operating voltage
D134	TRETD	Characteristic Retention	20	—	—	Year	Provided no other specifications are violated
D135	IDDP	Supply Current during Programming	—	10	—	mA	
D136a	TRW	Row Write Time	1.32	—	1.74	ms	TRW = 11064 FRC cycles, See **Note 2**
D137a	TPE	Page Erase Time	20.1	—	26.5	ms	TPE = 168517 FRC cycles, See **Note 2**
D138a	TWW	Word Write Cycle Time	42.3	—	55.9	µs	TWW = 355 FRC cycles, 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'011111` (for Min), TUN<5:0> = `b'100000` (for Max). This parameter depends on the FRC accuracy (see Table 26-19) 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 26-13: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial							
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)

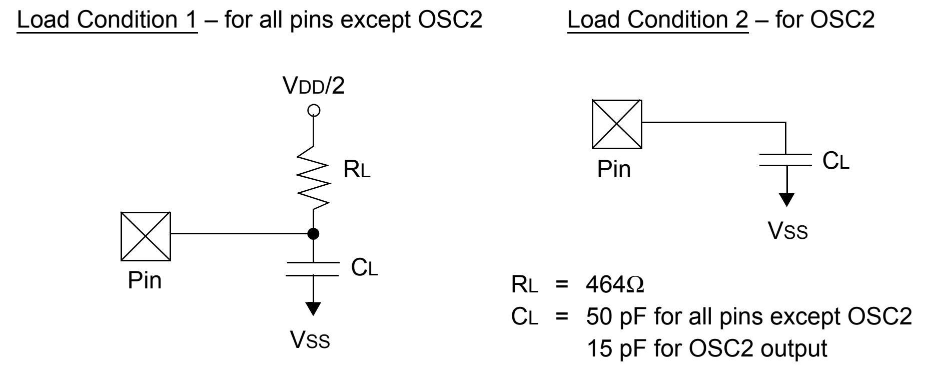
26.2 AC Characteristics and Timing Parameters

The information contained in this section defines dsPIC33FJXXXMCX06/X08/X10 AC characteristics and timing parameters.

TABLE 26-14: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

AC CHARACTERISTICS	**Standard Operating Conditions: 3.0V to 3.6V** **(unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial Operating voltage VDD range as described in **Section 26.0 “Electrical Characteristics”**.

FIGURE 26-1: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS



RL = 464Ω

CL = 50 pF for all pins except OSC2
15 pF for OSC2 output

TABLE 26-15: CAPACITIVE LOADING REQUIREMENTS ON OUTPUT PINS

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
DO50	COSC2	OSC2/SOSC2 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 26-2: EXTERNAL CLOCK TIMING

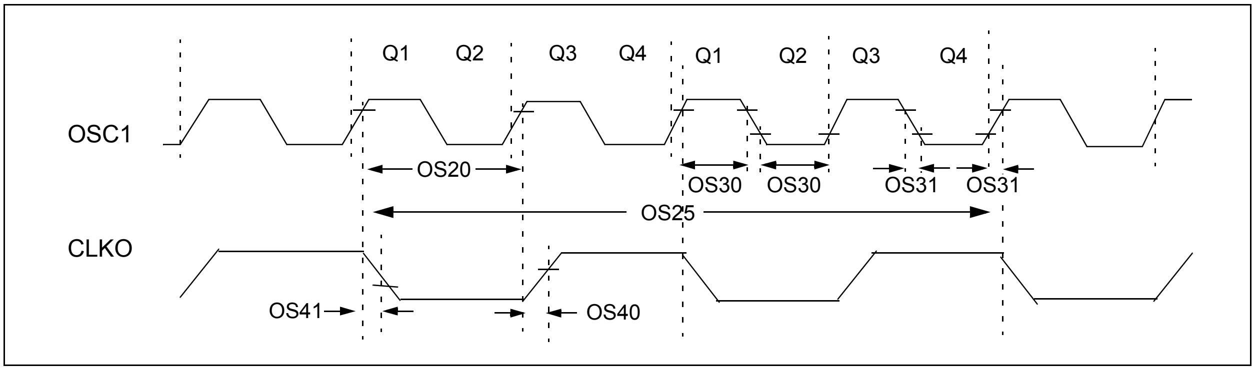


TABLE 26-16: EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS			**Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Symbol**	**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	TOSC = 1/FOSC	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 x TOSC	—	0.625 x TOSC	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	VDD = 3.3V TA = +25°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 26-17: PLL CLOCK TIMING SPECIFICATIONS (VDD = 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					
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ$^{(1)}$**	**Max**	**Units**	**Conditions**
OS50	FPLLI	PLL Voltage Controlled Oscillator (VCO) Input Frequency Range$^{(2)}$	0.8	—	8.0	MHz	ECPLL, HSPLL, 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	0.5	3.0	%	Measured over 100 ms period

Note 1: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated.

TABLE 26-18: AC CHARACTERISTICS: INTERNAL FRC 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 VDD = 3.0-3.6V

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

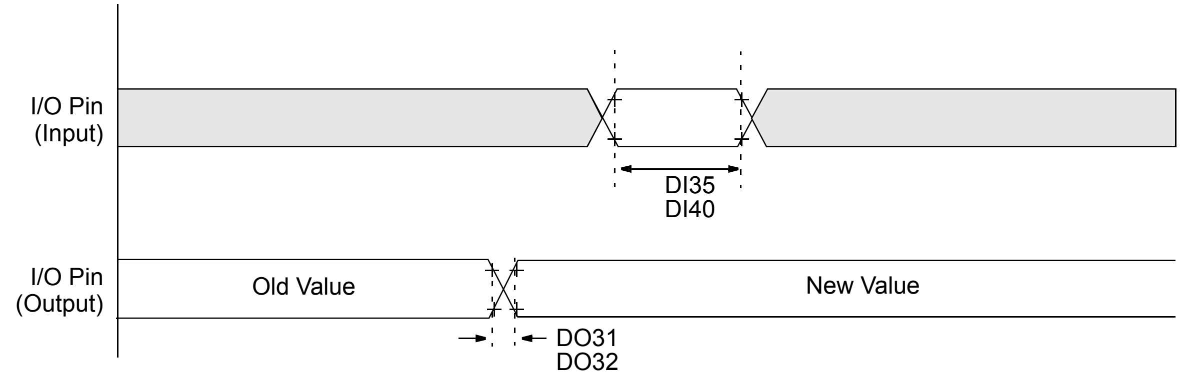
2: FRC set to initial frequency of 7.37 MHz (+1-2%) at 25° C FRC.

TABLE 26-19: INTERNAL LPRC 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**
	LPRC @ 32.768 kHz$^{(1)}$					
F21		-20	±6	+20	%	-40°C ≤ TA ≤ +85°C VDD = 3.0-3.6V

Note 1: Change of LPRC frequency as VDD changes.

FIGURE 26-3: CLKO AND I/O TIMING CHARACTERISTICS



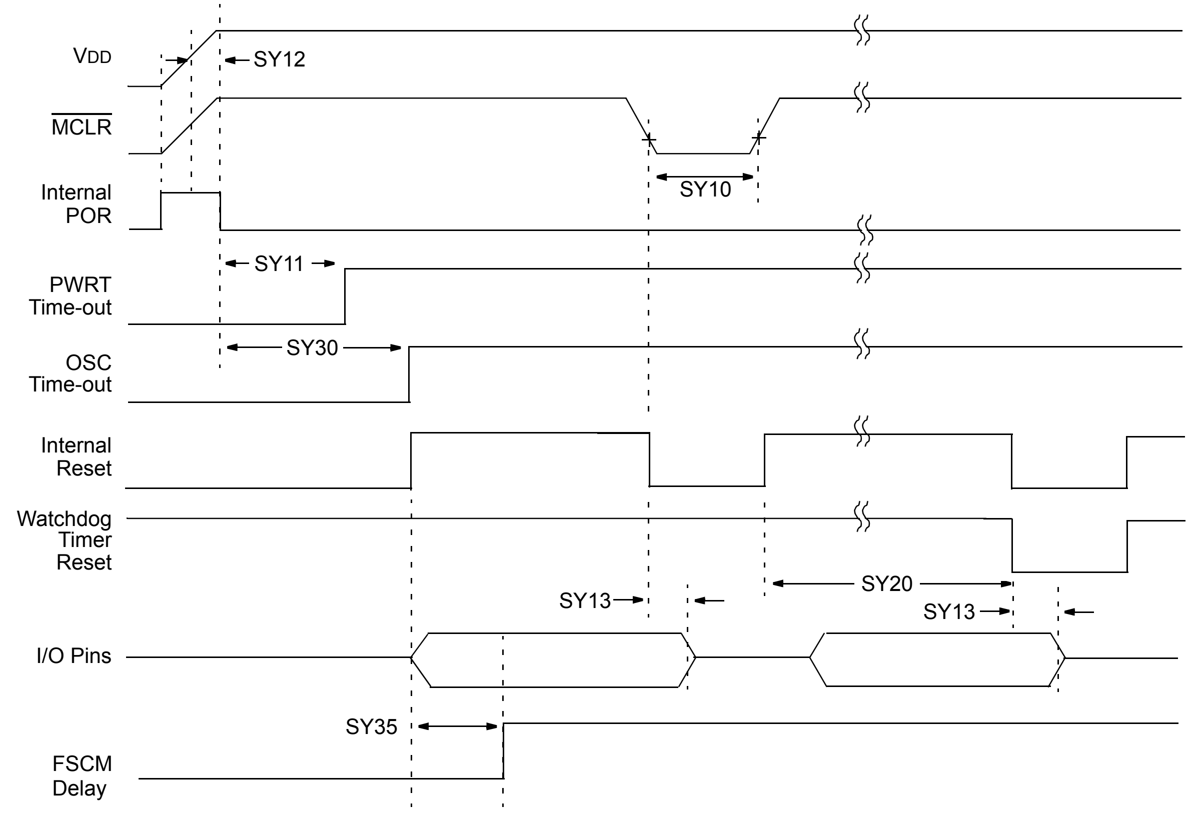
Note: Refer to Figure 26-1 for load conditions.

TABLE 26-20: I/O TIMING REQUIREMENTS

AC CHARACTERISTICS			**Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
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 26-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING CHARACTERISTICS



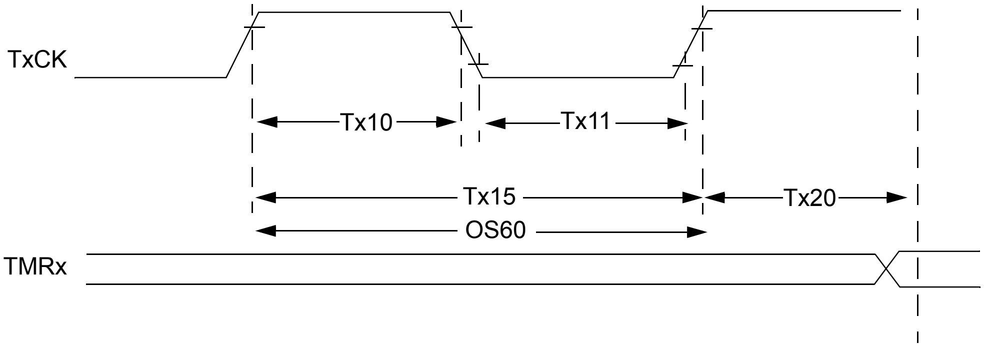
Note: Refer to Figure 26-1 for load conditions.

TABLE 26-21: 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				
Param No.	**Symbol**	**Characteristic[1]**	**Min**	**Typ[2]**	**Max**	**Units**	**Conditions**
SY10	TMCL	$\overline{\text{MCLR}}$ 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 $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset	0.68	0.72	1.2	μs	—
SY20	TWDT1	Watchdog Timer Time-out Period	—	—	—	—	See **Section 23.4 "Watchdog Timer (WDT)"** and LPRC specification F21 (Table 26-19)
SY30	TOST	Oscillator Start-up Timer Period	—	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.
2: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated.

FIGURE 26-5: TIMER1, 2, 3, 4, 5, 6, 7, 8 AND 9 EXTERNAL CLOCK TIMING CHARACTERISTICS



Note: Refer to Figure 26-1 for load conditions.

TABLE 26-22: 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				
Param No.	**Symbol**	**Characteristic**		**Min**	**Typ**	**Max**	**Units**	**Conditions**
TA10	TTXH	TxCK High Time	Synchronous, no prescaler	0.5 TCY + 20	—	—	ns	Must also meet parameter TA15
			Synchronous, with prescaler	10	—	—	ns	
			Asynchronous	10	—	—	ns	
TA11	TTXL	TxCK Low Time	Synchronous, no prescaler	0.5 TCY + 20	—	—	ns	Must also meet parameter TA15
			Synchronous, with prescaler	10	—	—	ns	
			Asynchronous	10	—	—	ns	
TA15	TTXP	TxCK Input Period	Synchronous, no prescaler	TCY + 40	—	—	ns	—
			Synchronous, with prescaler	Greater of: 20 ns or (TCY + 40)/N	—	—	—	N = prescale value (1, 8, 64, 256)
			Asynchronous	20	—	—	ns	—
OS60	Ft1	SOSC1/T1CK Oscillator Input frequency Range (oscillator enabled by setting bit TCS (T1CON<1>))		DC	—	50	kHz	—
TA20	TCKEXTMRL	Delay from External TxCK Clock Edge to Timer Increment		0.5 TCY	—	1.5 TCY	—	—

Note 1: Timer1 is a Type A.

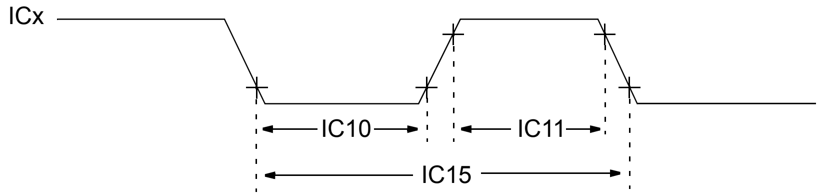
TABLE 26-23: TIMER2, TIMER4, TIMER6 AND TIMER8 EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS				**Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Symbol**	**Characteristic**		**Min**	**Typ**	**Max**	**Units**	**Conditions**
TB10	TtxH	TxCK High Time	Synchronous, no prescaler	0.5 TCY + 20	—	—	ns	Must also meet parameter TB15
			Synchronous, with prescaler	10	—	—	ns	
TB11	TtxL	TxCK Low Time	Synchronous, no prescaler	0.5 TCY + 20	—	—	ns	Must also meet parameter TB15
			Synchronous, with prescaler	10	—	—	ns	
TB15	TtxP	TxCK Input Period	Synchronous, no prescaler	TCY + 40	—	—	ns	N = prescale value (1, 8, 64, 256)
			Synchronous, with prescaler	Greater of: 20 ns or (TCY + 40)/N				
TB20	TCKEXTMRL	Delay from External TxCK Clock Edge to Timer Increment		0.5 TCY	—	1.5 TCY	—	—

TABLE 26-24: TIMER3, TIMER5, TIMER7 AND TIMER9 EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS				**Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Symbol**	**Characteristic**		**Min**	**Typ**	**Max**	**Units**	**Conditions**
TC10	TtxH	TxCK High Time	Synchronous	0.5 TCY + 20	—	—	ns	Must also meet parameter TC15
TC11	TtxL	TxCK Low Time	Synchronous	0.5 TCY + 20	—	—	ns	Must also meet parameter TC15
TC15	TtxP	TxCK Input Period	Synchronous, no prescaler	TCY + 40	—	—	ns	N = prescale value (1, 8, 64, 256)
			Synchronous, with prescaler	Greater of: 20 ns or (TCY + 40)/N				
TC20	TCKEXTMRL	Delay from External TxCK Clock Edge to Timer Increment		0.5 TCY	—	1.5 TCY	—	—

FIGURE 26-6: INPUT CAPTURE (CAPx) TIMING CHARACTERISTICS



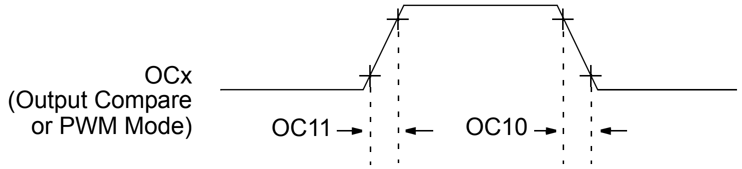
Note: Refer to Figure 26-1 for load conditions.

TABLE 26-25: INPUT CAPTURE TIMING REQUIREMENTS

AC CHARACTERISTICS				**Standard Operating Conditions: 3.0V to 3.6V** **(unless otherwise stated)** Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial			
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 26-7: OUTPUT COMPARE MODULE (OCx) TIMING CHARACTERISTICS



Note: Refer to Figure 26-1 for load conditions.

TABLE 26-26: OUTPUT COMPARE MODULE TIMING REQUIREMENTS

AC CHARACTERISTICS			**Standard Operating Conditions: 3.0V to 3.6V** **(unless otherwise stated)** Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial				
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 26-8: OC/PWM MODULE TIMING CHARACTERISTICS

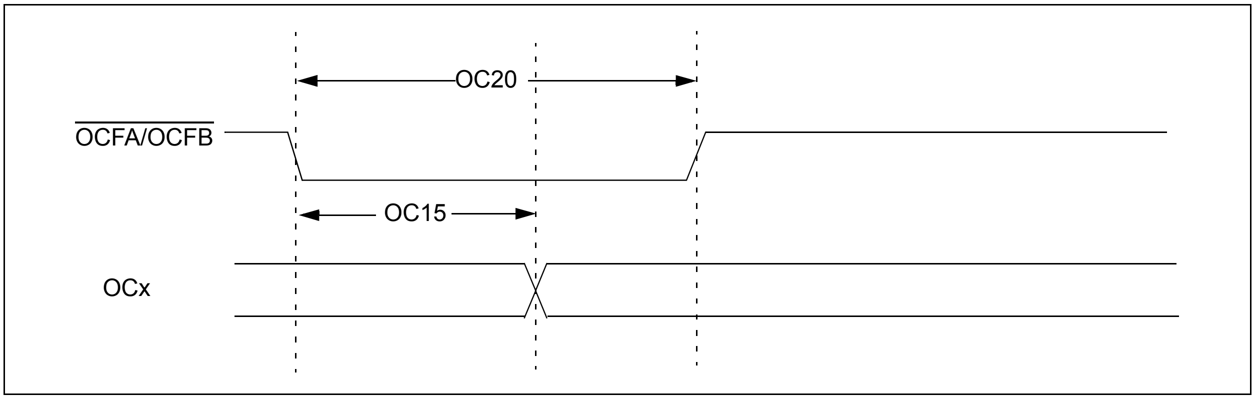


TABLE 26-27: SIMPLE OC/PWM MODE TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Symbol**	**Characteristic**[1]	**Min**	**Typ**	**Max**	**Units**	**Conditions**
OC15	TFD	Fault Input to PWM I/O Change	—	—	50	ns	—
OC20	TFLT	Fault Input Pulse-Width	50	—	—	ns	—

Note 1: These parameters are characterized but not tested in manufacturing.

FIGURE 26-9: MOTOR CONTROL PWM MODULE FAULT TIMING CHARACTERISTICS

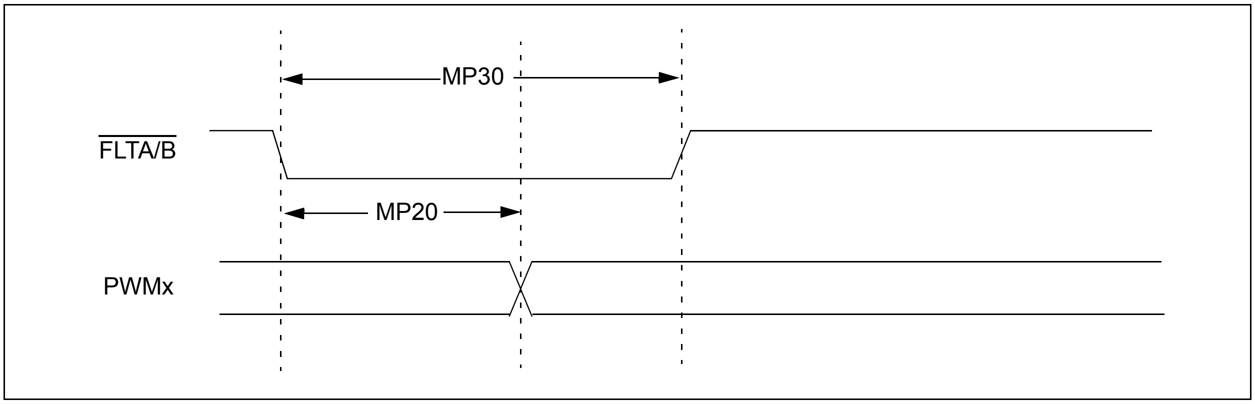
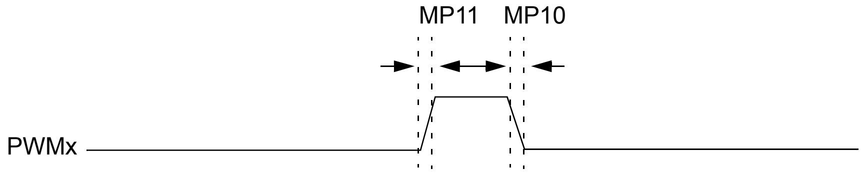


FIGURE 26-10: MOTOR CONTROL PWM MODULE TIMING CHARACTERISTICS



Note: Refer to Figure 26-1 for load conditions.

TABLE 26-28: 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				
Param No.	**Symbol**	**Characteristic[1]**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
MP10	TFPWM	PWM Output Fall Time	—	—	—	ns	See parameter D032
MP11	TRPWM	PWM Output Rise Time	—	—	—	ns	See parameter D031
MP20	TFD	Fault Input ↓ to PWM I/O Change	—	—	50	ns	—
MP30	TFH	Minimum Pulse-Width	50	—	—	ns	—

Note 1: These parameters are characterized but not tested in manufacturing.

FIGURE 26-11: QEA/QEB INPUT CHARACTERISTICS

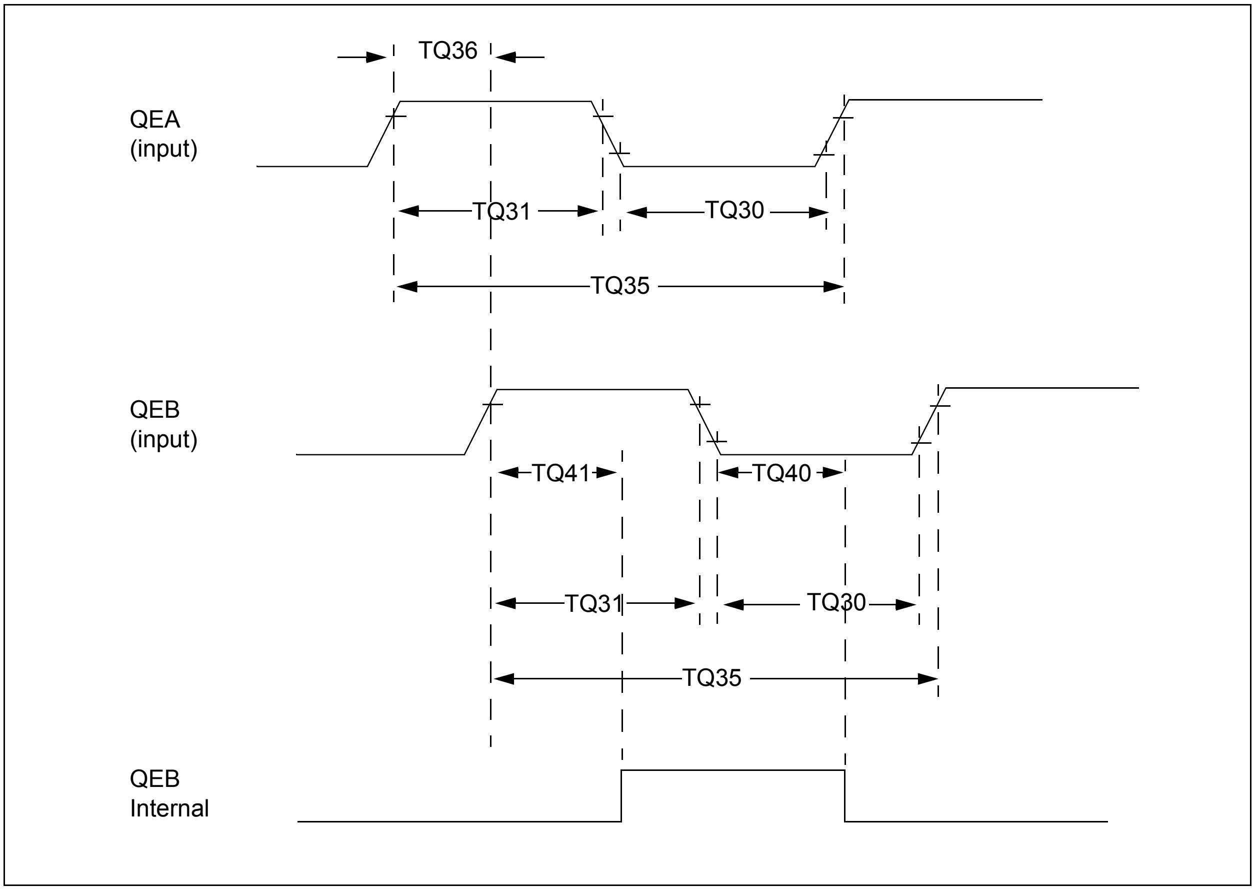


TABLE 26-29: QUADRATURE DECODER TIMING REQUIREMENTS

AC CHARACTERISTICS			**Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial			
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.

3: N = Index Channel Digital Filter Clock Divide Select bits. Refer to **Section 15. “Quadrature Encoder Interface (QEI)”** (DS70208) in the “*dsPIC33F Family Reference Manual*”.

FIGURE 26-12: QEI MODULE INDEX PULSE TIMING CHARACTERISTICS

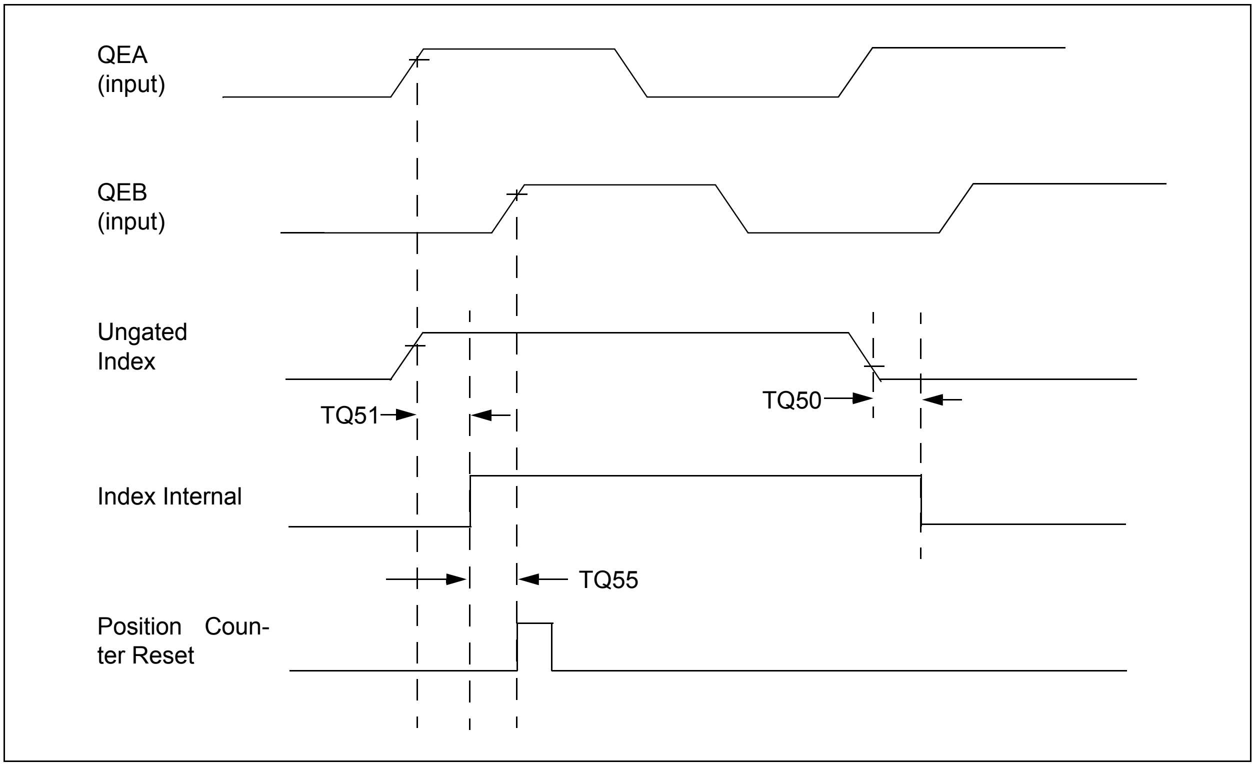


TABLE 26-30: QEI INDEX PULSE TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial			
Param No.	**Symbol**	**Characteristic**[1]	**Min**	**Max**	**Units**	**Conditions**
TQ50	TqIL	Filter Time to Recognize Low, with Digital Filter	3 * N * TCY	—	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 * TCY	—	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 TCY	—	ns	—

Note 1: These parameters are characterized but not tested in manufacturing.

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 26-13: TIMERQ (QEI MODULE) EXTERNAL CLOCK TIMING CHARACTERISTICS

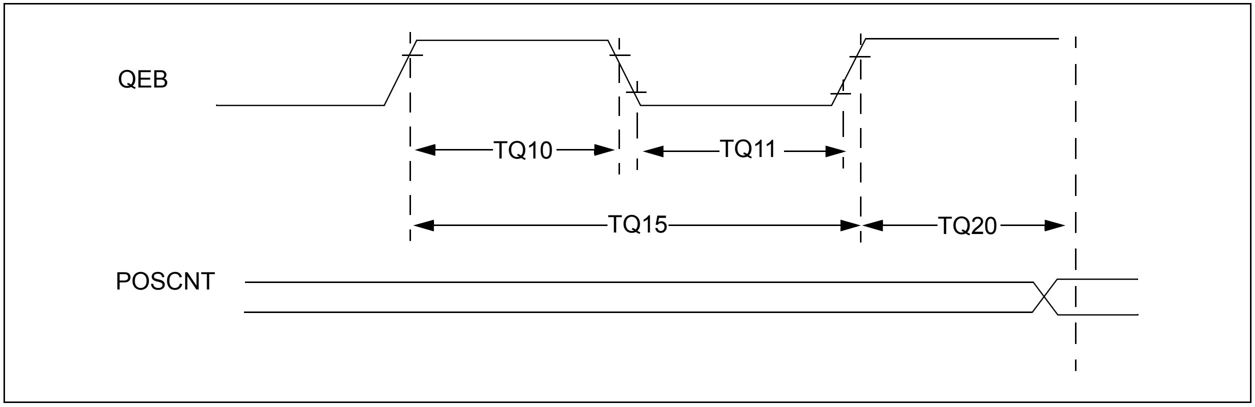
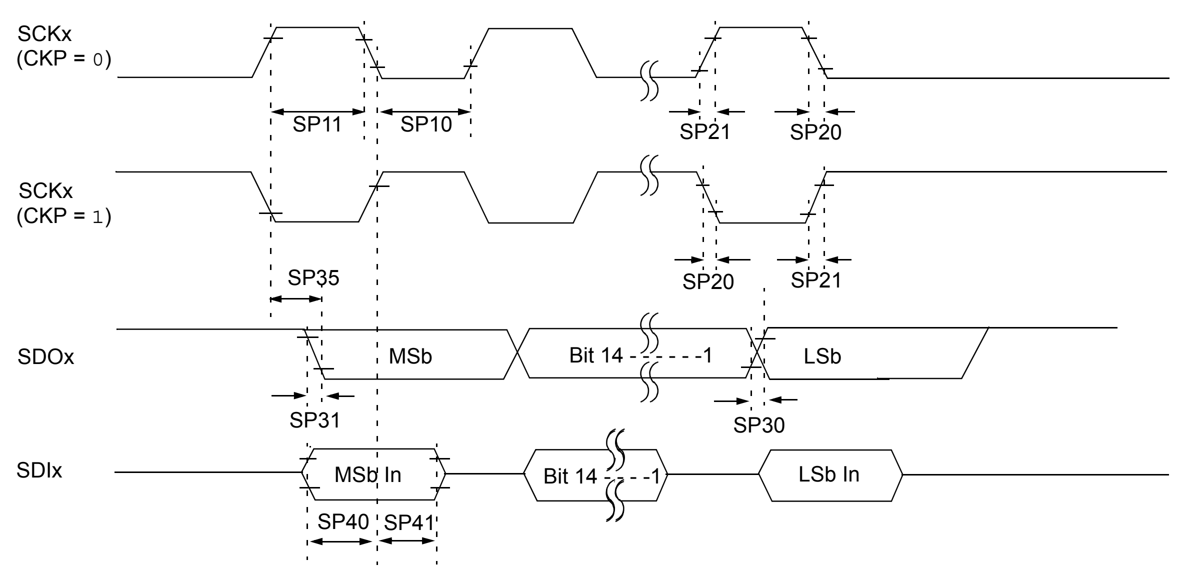


TABLE 26-31: QEI MODULE EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS				**Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Symbol**	**Characteristic[1]**		**Min**	**Typ**	**Max**	**Units**	**Conditions**
TQ10	TtQH	TQCK High Time	Synchronous, with prescaler	TCY + 20	—	—	ns	Must also meet parameter TQ15
TQ11	TtQL	TQCK Low Time	Synchronous, with prescaler	TCY + 20	—	—	ns	Must also meet parameter TQ15
TQ15	TtQP	TQCP Input Period	Synchronous, with prescaler	2 * TCY + 40	—	—	ns	—
TQ20	TCKEXTMRL	Delay from External TxCK Clock Edge to Timer Increment		0.5 TCY	—	1.5 TCY	—	—

Note 1: These parameters are characterized but not tested in manufacturing.

FIGURE 26-14: SPIx MODULE MASTER MODE (CKE = 0) TIMING CHARACTERISTICS



Note: Refer to Figure 26-1 for load conditions.

TABLE 26-32: 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				
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	—

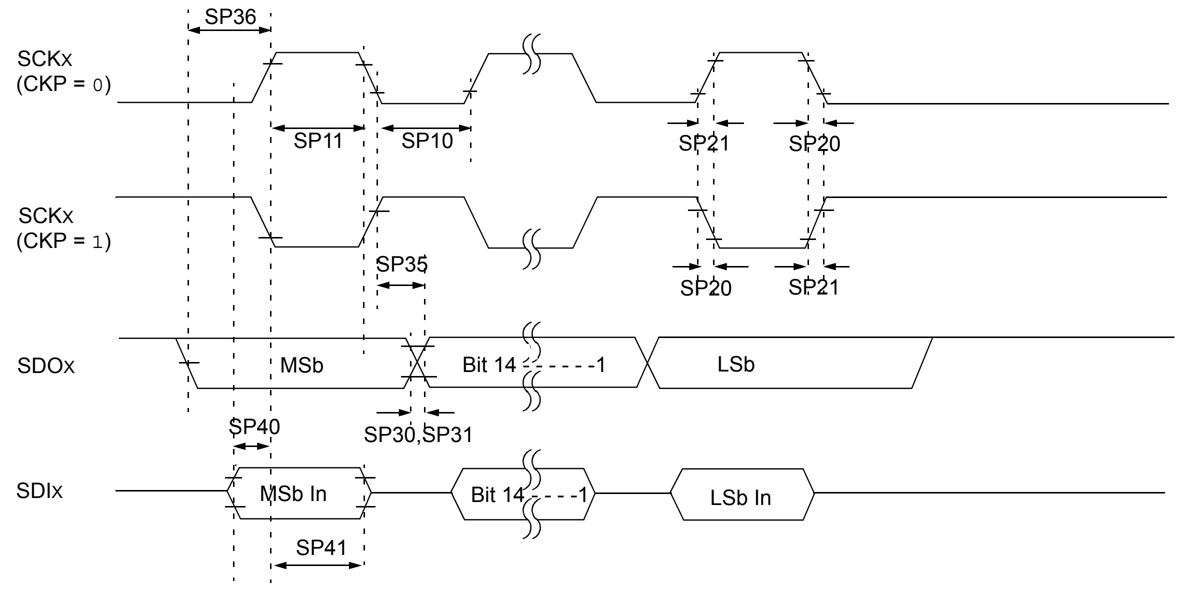
Note 1: These parameters are characterized but not tested in manufacturing.

2: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated.

3: The minimum clock period for SCKx is 100 ns. Therefore, the clock generated in Master mode must not violate this specification.

4: Assumes 50 pF load on all SPIx pins.

FIGURE 26-15: SPIx MODULE MASTER MODE (CKE = 1) TIMING CHARACTERISTICS



Note: Refer to Figure 26-1 for load conditions.

TABLE 26-33: 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				
Param No.	**Symbol**	**Characteristic[1]**	**Min**	**Typ[2]**	**Max**	**Units**	**Conditions**
SP10	TscL	SCKx Output Low Time[3]	TCY/2	—	—	ns	—
SP11	TscH	SCKx Output High Time[3]	TCY/2	—	—	ns	—
SP20	TscF	SCKx Output Fall Time[4]	—	—	—	ns	See parameter D032
SP21	TscR	SCKx Output Rise Time[4]	—	—	—	ns	See parameter D031
SP30	TdoF	SDOx Data Output Fall Time[4]	—	—	—	ns	See parameter D032
SP31	TdoR	SDOx Data Output Rise Time[4]	—	—	—	ns	See parameter D031
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	6	20	ns	—
SP36	TdoV2sc, TdoV2scL	SDOx Data Output Setup to First SCKx Edge	20	—	—	ns	—
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	30	—	—	ns	—
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	20	—	—	ns	—

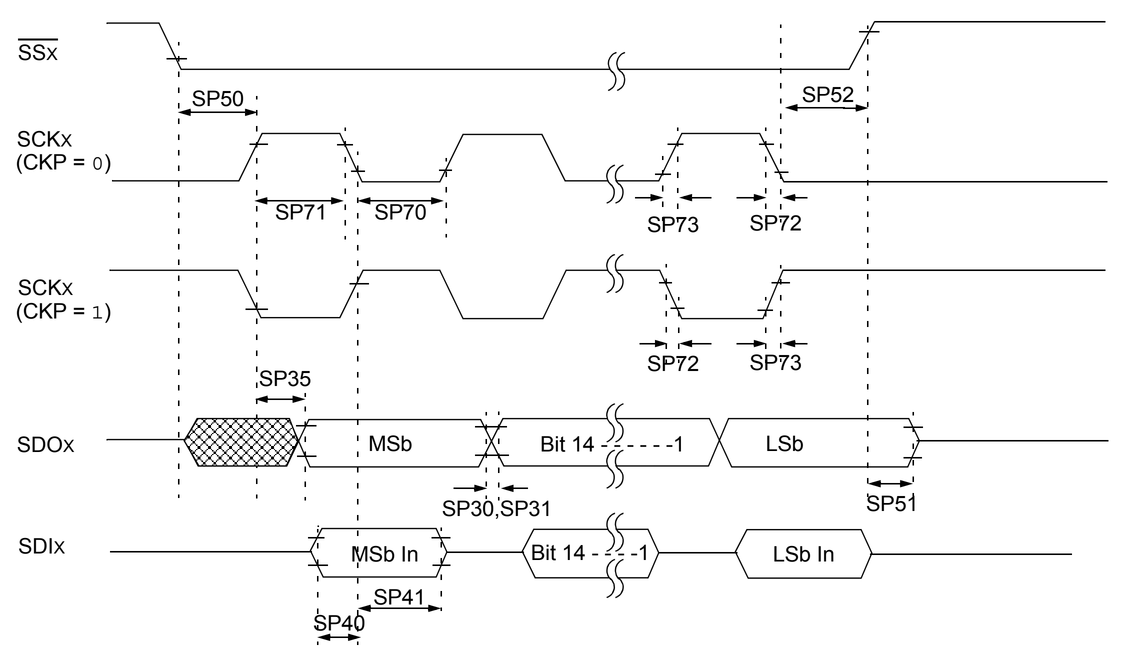
Note 1: These parameters are characterized but not tested in manufacturing.

2: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated.

3: The minimum clock period for SCKx is 100 ns. Therefore, the clock generated in Master mode must not violate this specification.

4: Assumes 50 pF load on all SPIx pins.

FIGURE 26-16: SPIx MODULE SLAVE MODE (CKE = 0) TIMING CHARACTERISTICS



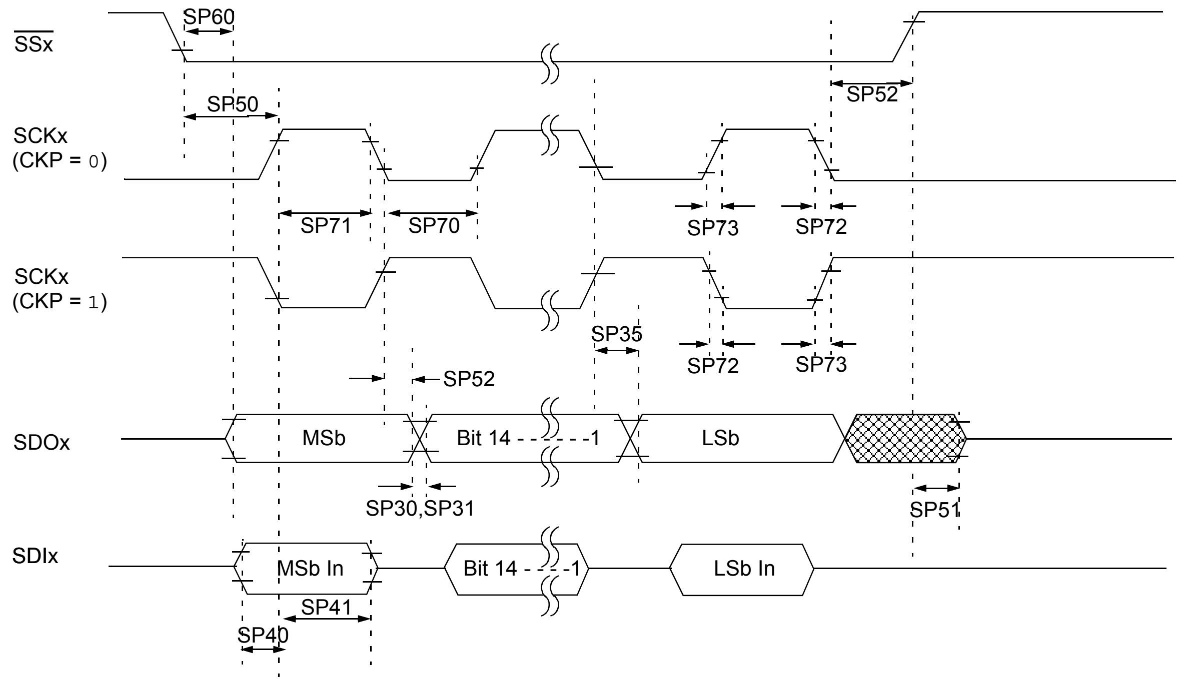
Note: Refer to Figure 26-1 for load conditions.

TABLE 26-34: 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				
Param No.	**Symbol**	**Characteristic[1]**	**Min**	**Typ**	**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[3]	—	10	25	ns	—
SP73	TscR	SCKx Input Rise Time[3]	—	10	25	ns	—
SP30	TdoF	SDOx Data Output Fall Time[3]	—	—	—	ns	See parameter D032
SP31	TdoR	SDOx Data Output Rise Time[3]	—	—	—	ns	See parameter D031
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[3]	10	—	50	ns	—
SP52	TscH2ssH TscL2ssH	$\overline{\text{SSx}}$ after SCKx Edge	1.5 TCY + 40	—	—	ns	—

Note 1: These parameters are characterized but not tested in manufacturing.

FIGURE 26-17: SPIx MODULE SLAVE MODE (CKE = 1) TIMING CHARACTERISTICS



Note: Refer to Figure 26-1 for load conditions.

TABLE 26-35: SPIx MODULE SLAVE MODE (CKE = 1) TIMING REQUIREMENTS

AC CHARACTERISTICS			**Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial				
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[3]	—	10	25	ns	—
SP73	TscR	SCKx Input Rise Time[3]	—	10	25	ns	—
SP30	TdoF	SDOx Data Output Fall Time[3]	—	—	—	ns	See parameter D032
SP31	TdoR	SDOx Data Output Rise Time[3]	—	—	—	ns	See parameter D031
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[4]	10	—	50	ns	—
SP52	TscH2ssH TscL2ssH	$\overline{\text{SSx}}$ ↑ after SCKx Edge	1.5 T_{CY} + 40	—	—	ns	—
SP60	TssL2doV	SDOx Data Output Valid after $\overline{\text{SSx}}$ Edge	—	—	50	ns	—

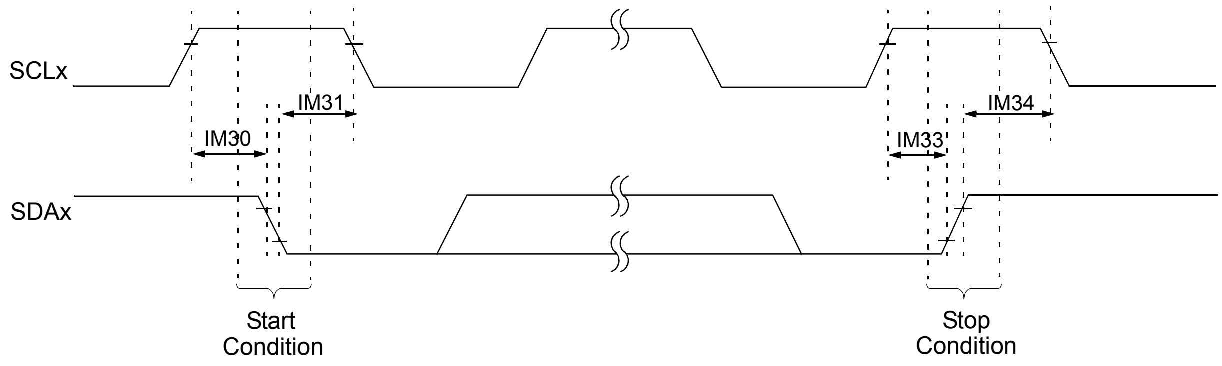
Note 1: These parameters are characterized but not tested in manufacturing.

2: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated.

3: The minimum clock period for SCKx is 100 ns. Therefore, the clock generated in Master mode must not violate this specification.

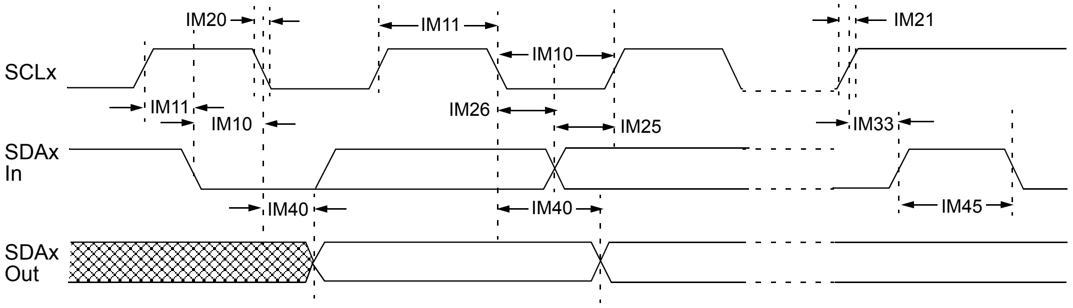
4: Assumes 50 pF load on all SPIx pins.

FIGURE 26-18: I2Cx BUS START/STOP BITS TIMING CHARACTERISTICS (MASTER MODE)



Note: Refer to Figure 26-1 for load conditions.

FIGURE 26-19: I2Cx BUS DATA TIMING CHARACTERISTICS (MASTER MODE)



Note: Refer to Figure 26-1 for load conditions.

TABLE 26-36: I2Cx BUS DATA TIMING REQUIREMENTS (MASTER MODE)

AC CHARACTERISTICS				Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial			
Param No.	**Symbol**	**Characteristic**		**Min(1)**	**Max**	**Units**	**Conditions**
IM10	TLO:SCL	Clock Low Time	100 kHz mode	TCY/2 (BRG + 1)	—	µs	—
			400 kHz mode	TCY/2 (BRG + 1)	—	µs	—
			1 MHz mode(2)	TCY/2 (BRG + 1)	—	µs	—
IM11	THI:SCL	Clock High Time	100 kHz mode	TCY/2 (BRG + 1)	—	µs	—
			400 kHz mode	TCY/2 (BRG + 1)	—	µs	—
			1 MHz mode(2)	TCY/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 CB	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 CB	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	TCY/2 (BRG + 1)	—	µs	Only relevant for Repeated Start condition
			400 kHz mode	TCY/2 (BRG + 1)	—	µs	
			1 MHz mode(2)	TCY/2 (BRG + 1)	—	µs	
IM31	THD:STA	Start Condition Hold Time	100 kHz mode	TCY/2 (BRG + 1)	—	µs	After this period the first clock pulse is generated
			400 kHz mode	TCY/2 (BRG + 1)	—	µs	
			1 MHz mode(2)	TCY/2 (BRG + 1)	—	µs	
IM33	TSU:STO	Stop Condition Setup Time	100 kHz mode	TCY/2 (BRG + 1)	—	µs	—
			400 kHz mode	TCY/2 (BRG + 1)	—	µs	
			1 MHz mode(2)	TCY/2 (BRG + 1)	—	µs	
IM34	THD:STO	Stop Condition Hold Time	100 kHz mode	TCY/2 (BRG + 1)	—	ns	—
			400 kHz mode	TCY/2 (BRG + 1)	—	ns	
			1 MHz mode(2)	TCY/2 (BRG + 1)	—	ns	
IM40	TAA:SCL	Output Valid From Clock	100 kHz mode	—	3500	µs	—
			400 kHz mode	—	1000	µs	—
			1 MHz mode(2)	—	400	µs	—
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 I^{2}C Baud Rate Generator. Refer to **Section 19. "Inter-Integrated Circuit™ (I^{2}C™)"** (DS70195) in the "*dsPIC33F Family Reference Manual".*

2: Maximum pin capacitance = 10 pF for all I2Cx pins (for 1 MHz mode only).

FIGURE 26-20: I2Cx BUS START/STOP BITS TIMING CHARACTERISTICS (SLAVE MODE)

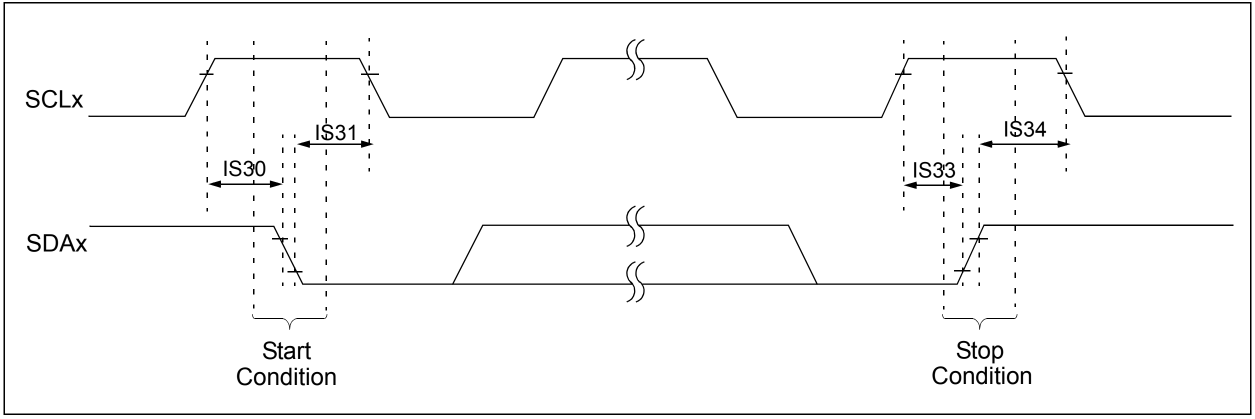


FIGURE 26-21: I2Cx BUS DATA TIMING CHARACTERISTICS (SLAVE MODE)

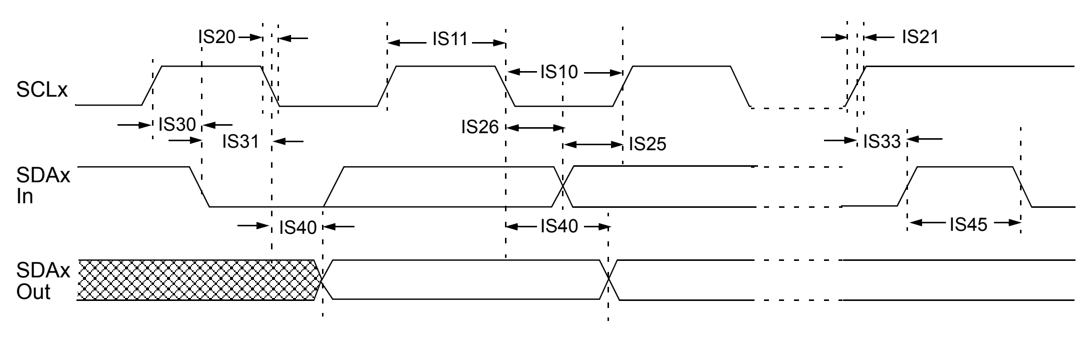


TABLE 26-37: I2Cx BUS DATA TIMING REQUIREMENTS (SLAVE MODE)

AC CHARACTERISTICS				**Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial			
Param No.	**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 26-22: CAN MODULE I/O TIMING CHARACTERISTICS

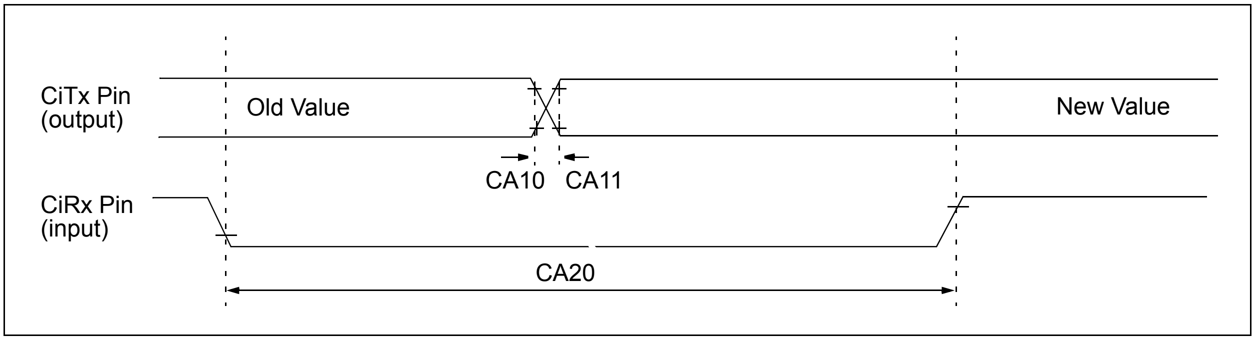


TABLE 26-38: ECAN™ MODULE I/O TIMING REQUIREMENTS

AC CHARACTERISTICS			**Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Symbol**	**Characteristic[1]**	**Min**	**Typ**	**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.

TABLE 26-39: ADC MODULE SPECIFICATIONS

AC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
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	— —	250 —	550 10	µA µA	ADC operating, see **Note 1** ADC off, see **Note 1**
AD08a	IAD	Operating Current	— —	7.0 2.7	9.0 3.2	mA mA	10-bit ADC mode, See **Note 2** 12-bit ADC mode, See **Note 2**
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	— —	— —	200 200	Ω Ω	10-bit ADC 12-bit ADC

Note 1: These parameters are not characterized or tested in manufacturing.

2: These parameters are characterized; but not tested in manufacturing

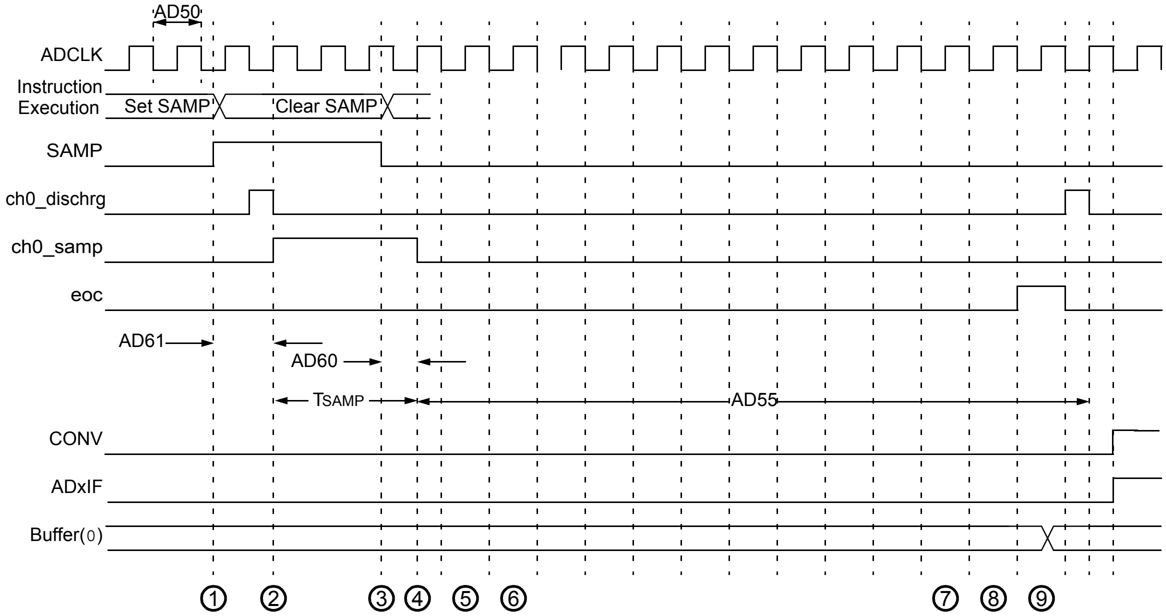
TABLE 26-40: ADC MODULE SPECIFICATIONS (12-BIT MODE)

AC CHARACTERISTICS			**Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
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	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V
AD22a	DNL	Differential Nonlinearity	>-1	—	<1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V
AD23a	GERR	Gain Error	1.25	1.5	3	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V
AD24a	EOFF	Offset Error	1.25	1.52	2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V
AD25a	—	Monotonicity	—	—	—	—	Guaranteed
ADC Accuracy (12-bit Mode) – Measurements with internal VREF+/VREF-							
AD20b	Nr	Resolution	12 data bits			bits	
AD21b	INL	Integral Nonlinearity	-2	—	+2	LSb	VINL = AVSS = 0V, AVDD = 3.6V
AD22b	DNL	Differential Nonlinearity	>-1	—	<1	LSb	VINL = AVSS = 0V, AVDD = 3.6V
AD23b	GERR	Gain Error	2	3	7	LSb	VINL = AVSS = 0V, AVDD = 3.6V
AD24b	EOFF	Offset Error	2	3	5	LSb	VINL = AVSS = 0V, AVDD = 3.6V
AD25b	—	Monotonicity	—	—	—	—	Guaranteed
Dynamic Performance (12-bit Mode)							
AD30a	THD	Total Harmonic Distortion	-77	-69	-61	dB	—
AD31a	SINAD	Signal to Noise and Distortion	59	63	64	dB	—
AD32a	SFDR	Spurious Free Dynamic Range	63	72	74	dB	—
AD33a	FNYQ	Input Signal Bandwidth	—	—	250	kHz	—
AD34a	ENOB	Effective Number of Bits	10.95	11.1	—	bits	—

TABLE 26-41: ADC MODULE SPECIFICATIONS (10-BIT MODE)

AC CHARACTERISTICS			**Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
ADC Accuracy (10-bit Mode) – Measurements with external VREF+/VREF-							
AD20c	Nr	Resolution	10 data bits			bits	
AD21c	INL	Integral Nonlinearity	-1.5	—	+1.5	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V
AD22c	DNL	Differential Nonlinearity	>-1	—	<1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V
AD23c	GERR	Gain Error	1	3	6	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V
AD24c	EOFF	Offset Error	1	2	5	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V
AD25c	—	Monotonicity	—	—	—	—	Guaranteed
ADC Accuracy (10-bit Mode) – Measurements with internal VREF+/VREF-							
AD20d	Nr	Resolution	10 data bits			bits	
AD21d	INL	Integral Nonlinearity	-1	—	+1	LSb	VINL = AVSS = 0V, AVDD = 3.6V
AD22d	DNL	Differential Nonlinearity	>-1	—	<1	LSb	VINL = AVSS = 0V, AVDD = 3.6V
AD23d	GERR	Gain Error	1	5	6	LSb	VINL = AVSS = 0V, AVDD = 3.6V
AD24d	EOFF	Offset Error	1	2	3	LSb	VINL = AVSS = 0V, AVDD = 3.6V
AD25d	—	Monotonicity	—	—	—	—	Guaranteed
Dynamic Performance (10-bit Mode)							
AD30b	THD	Total Harmonic Distortion	—	-64	-67	dB	—
AD31b	SINAD	Signal to Noise and Distortion	—	57	58	dB	—
AD32b	SFDR	Spurious Free Dynamic Range	—	60	62	dB	—
AD33b	FNYQ	Input Signal Bandwidth	—	—	550	kHz	—
AD34b	ENOB	Effective Number of Bits	9.1	9.7	9.8	bits	—

FIGURE 26-23: ADC CONVERSION (12-BIT MODE) TIMING CHARACTERISTICS (ASAM = `0`, SSRC<2:0> = `000`)



① – Software sets ADxCON. SAMP to start sampling.

② – Sampling starts after discharge period. TSAMP is described in **Section 16. “Analog-to-Digital Converter (ADC)”** (DS70183) in the *“dsPIC33F Family Reference Manual”*.

③ – Software clears ADxCON. SAMP to start conversion.

④ – Sampling ends, conversion sequence starts.

⑤ – Convert bit 11.

⑥ – Convert bit 10.

⑦ – Convert bit 1.

⑧ – Convert bit 0.

⑨ – One TAD for end of conversion.

TABLE 26-42: ADC CONVERSION (12-BIT MODE) TIMING REQUIREMENTS

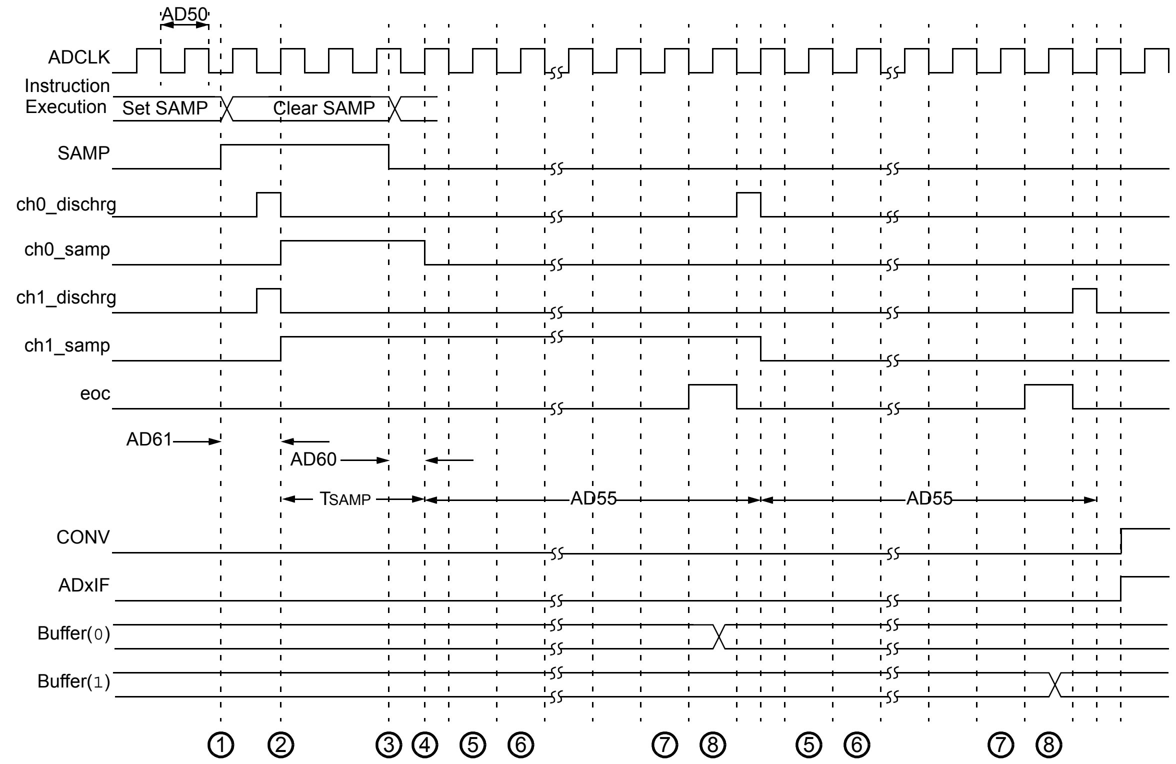
AC CHARACTERISTICS			**Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial				
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ(1)**	**Max.**	**Units**	**Conditions**
		Clock Parameters					
AD50a	T_{AD}	ADC Clock Period	117.6	—	—	ns	—
AD51a	t_{RC}	ADC Internal RC Oscillator Period	—	250	—	ns	—
		Conversion Rate					
AD55a	t_{CONV}	Conversion Time	—	14 T_{AD}		—	—
AD56a	F_{CNV}	Throughput Rate	—	—	500	ksps	—
AD57a	T_{SAMP}	Sample Time	3.0 T_{AD}	—	—	—	—
		Timing Parameters					
AD60a	t_{PCS}	Conversion Start from Sample Trigger(2)	2.0 T_{AD}	—	3.0 T_{AD}	—	—
AD61a	t_{PSS}	Sample Start from Setting Sample (SAMP) bit(2)	2.0 T_{AD}	—	3.0 T_{AD}	—	—
AD62a	t_{CSS}	Conversion Completion to Sample Start (ASAM = `1`)(2)	—	0.5 T_{AD}	—	—	—
AD63a	t_{DPU}	Time to Stabilize Analog Stage from ADC Off to ADC On(2,3)	—	—	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 can affect linearity performance, especially at elevated temperatures.

3: t_{DPU} is the time required for the ADC module to stabilize when it is turned on (AD1CON1<ADON> = `1`). During this time, the ADC result is indeterminate.

FIGURE 26-24: ADC CONVERSION (10-BIT MODE) TIMING CHARACTERISTICS (CHPS<1:0> = `01`, SIMSAM = `0`, ASAM = `0`, SSRC<2:0> = `000`)



① – Software sets ADxCON. SAMP to start sampling.

② – Sampling starts after discharge period. TSAMP is described in **Section 16. “Analog-to-Digital Converter (ADC)”** (DS70183) in the *“dsPIC33F Family Reference Manual”*.

③ – Software clears ADxCON. SAMP to start conversion.

④ – Sampling ends, conversion sequence starts.

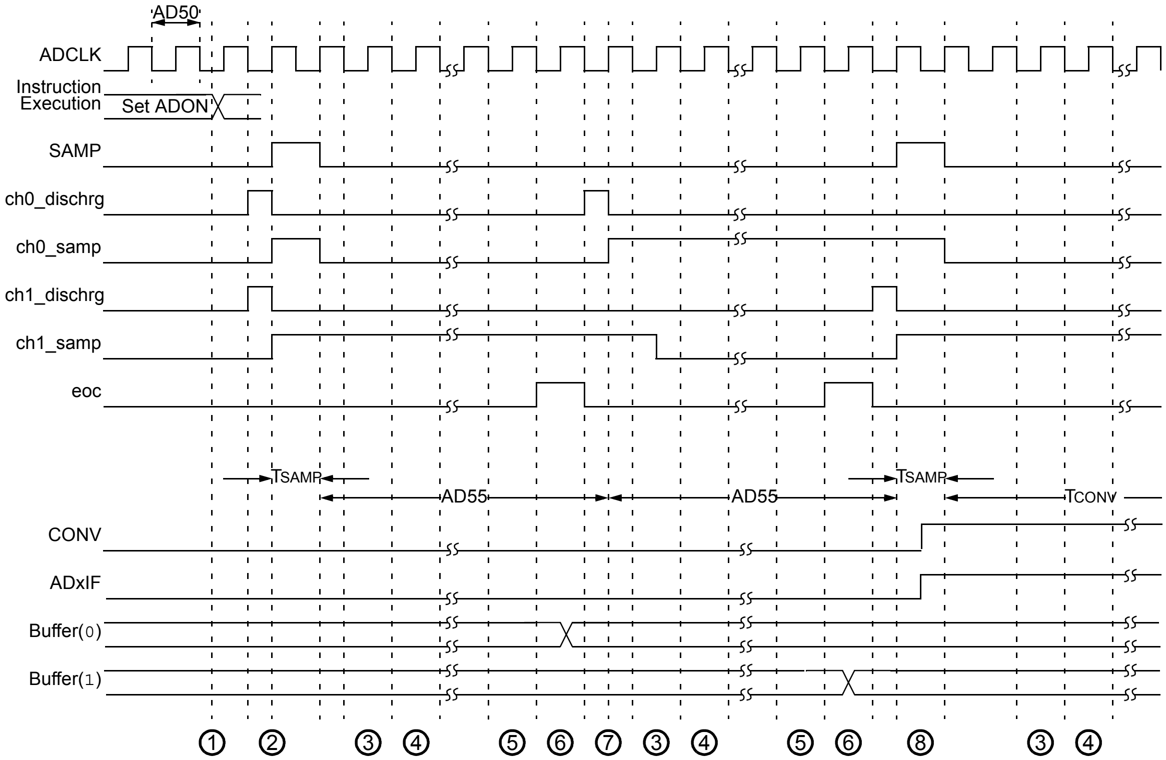
⑤ – Convert bit 9.

⑥ – Convert bit 8.

⑦ – Convert bit 0.

⑧ – One TAD for end of conversion.

FIGURE 26-25: ADC CONVERSION (10-BIT MODE) TIMING CHARACTERISTICS (CHPS<1:0> = `01`, SIMSAM = `0`, ASAM = `1`, SSRC<2:0> = `111`, SAMC<4:0> = `00001`)



① – Software sets ADxCON. ADON to start AD operation.

② – Sampling starts after discharge period. TSAMP is described in **Section 16. “Analog-to-Digital Converter (ADC)”** (DS70183) in the *“dsPIC33F Family Reference Manual”*.

③ – Convert bit 9.

④ – Convert bit 8.

⑤ – Convert bit 0.

⑥ – One TAD for end of conversion.

⑦ – Begin conversion of next channel.

⑧ – Sample for time specified by SAMC<4:0>.

TABLE 26-43: 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				
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ[1]**	**Max.**	**Units**	**Conditions**
		Clock Parameters					
AD50b	TAD	ADC Clock Period	76	—	—	ns	—
AD51b	tRC	ADC Internal RC Oscillator Period	—	250	—	ns	—
		Conversion Rate					
AD55b	tCONV	Conversion Time	—	12 TAD	—	—	—
AD56b	FCNV	Throughput Rate	—	—	1.1	Msps	—
AD57b	TSAMP	Sample Time	2 TAD	—	—	—	—
		Timing Parameters					
AD60b	tPCS	Conversion Start from Sample Trigger[2]	2.0 TAD	—	3.0 TAD	—	Auto-Convert Trigger (SSRC<2:0> = `111`) not selected
AD61b	tPSS	Sample Start from Setting Sample (SAMP) bit[2]	2.0 TAD	—	3.0 TAD	—	—
AD62b	tCSS	Conversion Completion to Sample Start (ASAM = `1`)[2]	—	0.5 TAD	—	—	—
AD63b	tDPU	Time to Stabilize Analog Stage from ADC Off to ADC On[2,3]	—	—	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 can affect linearity performance, especially at elevated temperatures.

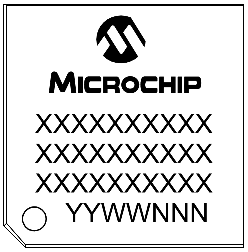
3: tDPU is the time required for the ADC module to stabilize when it is turned on (AD1CON1<ADON> = `1`). During this time, the ADC result is indeterminate.

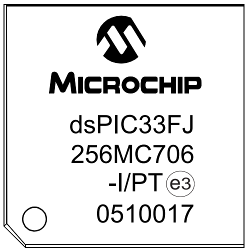
NOTES:

27.0 PACKAGING INFORMATION

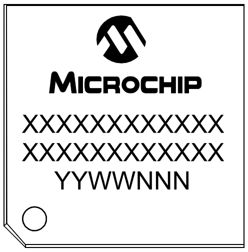
27.1 Package Marking Information

64-Lead TQFP (10x10x1 mm)

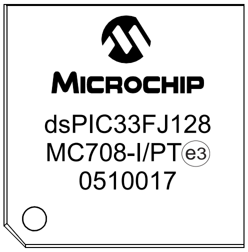




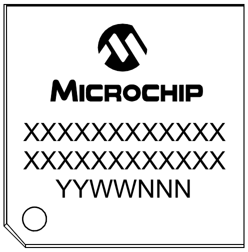
80-Lead TQFP (12x12x1 mm)



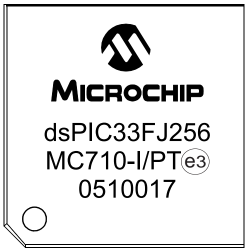
Example



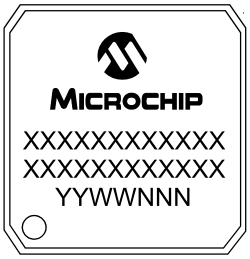
100-Lead TQFP (12x12x1 mm)



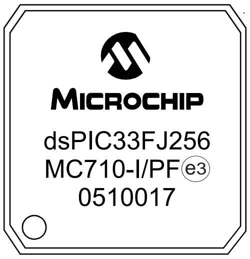
Example



100-Lead TQFP (14x14x1mm)



100-Lead TQFP (14x14x1mm)



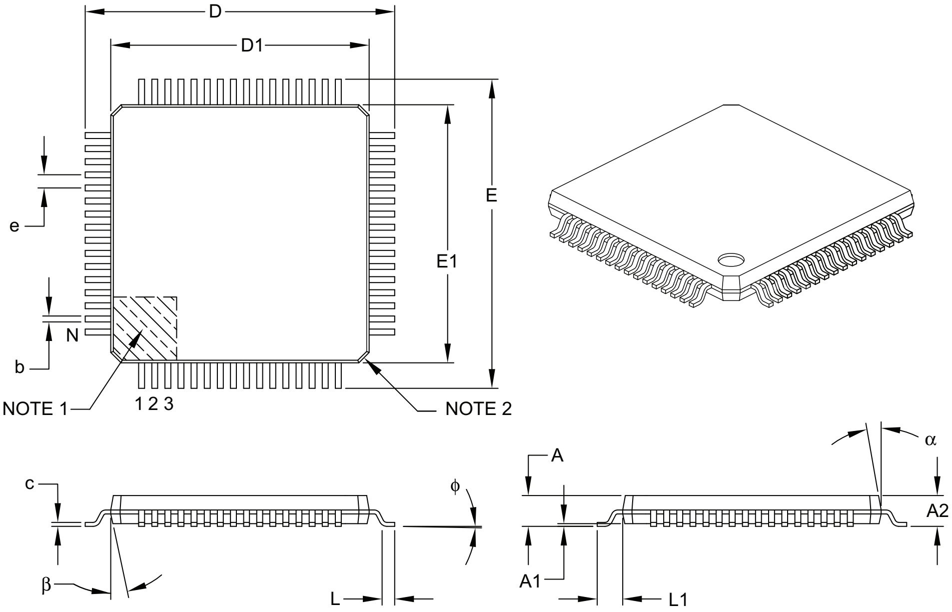
Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	ⓔ3	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (ⓔ3) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

27.2 Package Details

64-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm Footprint [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		64	
Lead Pitch	e		0.50 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	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

Notes:

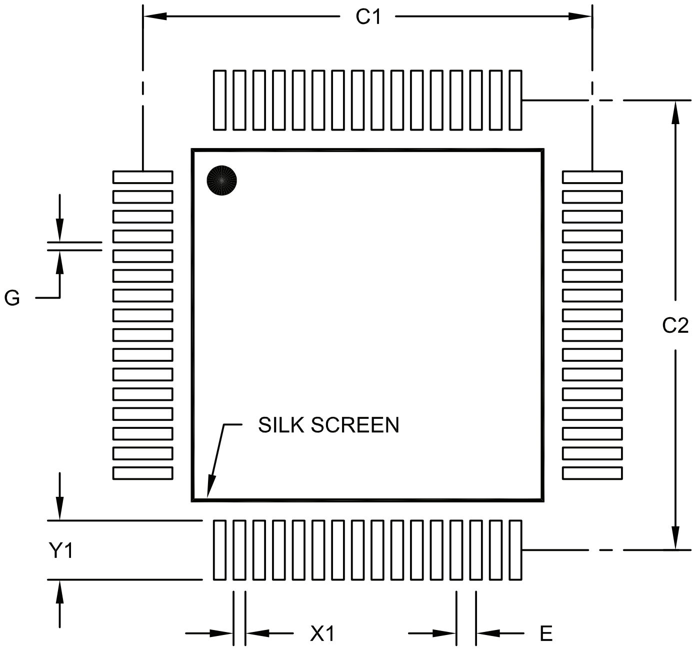
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-085B

64-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X64)	X1			0.30
Contact Pad Length (X64)	Y1			1.50
Distance Between Pads	G	0.20		

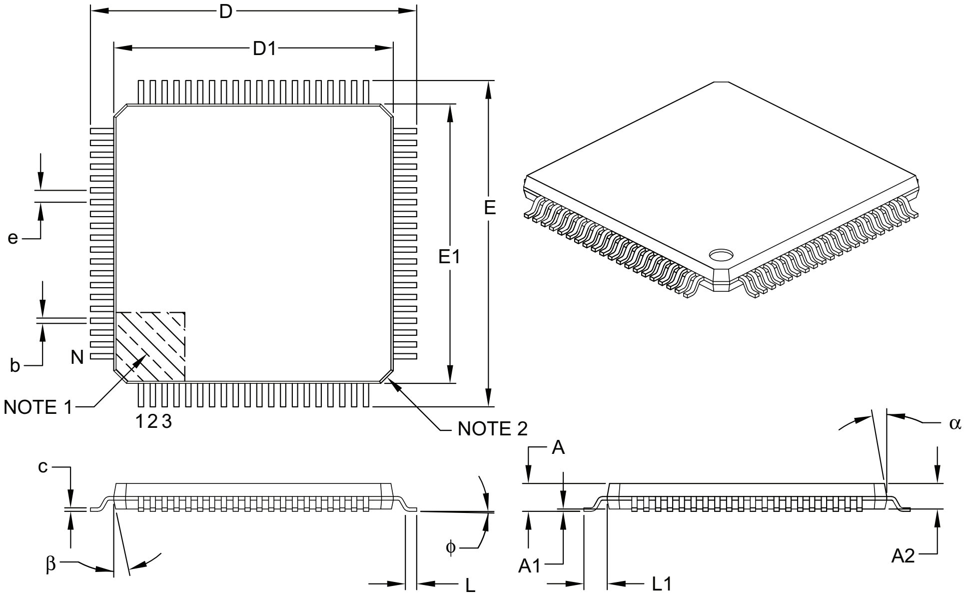
Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2085A

80-Lead Plastic Thin Quad Flatpack (PT) – 12x12x1 mm Body, 2.00 mm Footprint [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		80	
Lead Pitch	e		0.50 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		14.00 BSC	
Overall Length	D		14.00 BSC	
Molded Package Width	E1		12.00 BSC	
Molded Package Length	D1		12.00 BSC	
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

Notes:

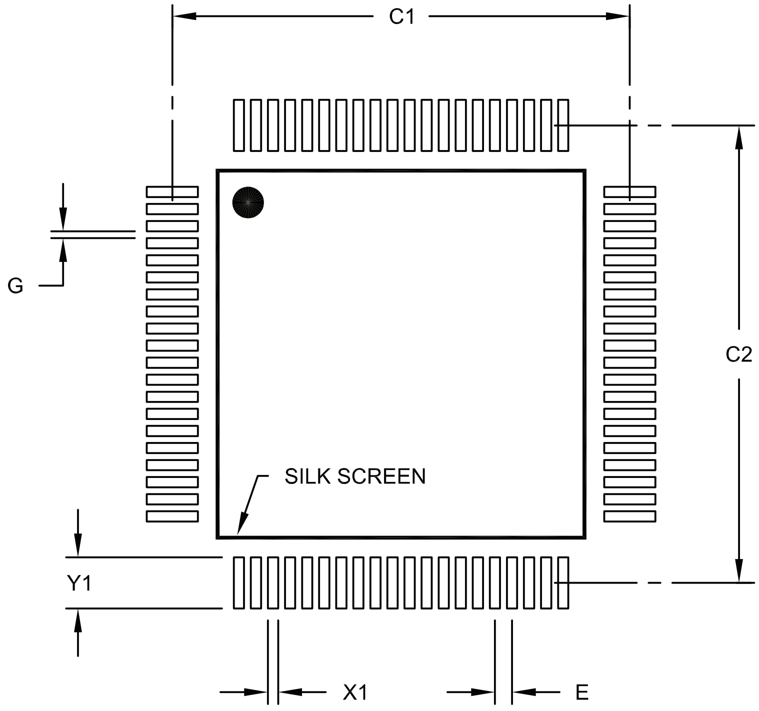
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-092B

80-Lead Plastic Thin Quad Flatpack (PT) – 12x12x1 mm Body, 2.00 mm [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Contact Pad Spacing	C1		13.40	
Contact Pad Spacing	C2		13.40	
Contact Pad Width (X80)	X1			0.30
Contact Pad Length (X80)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

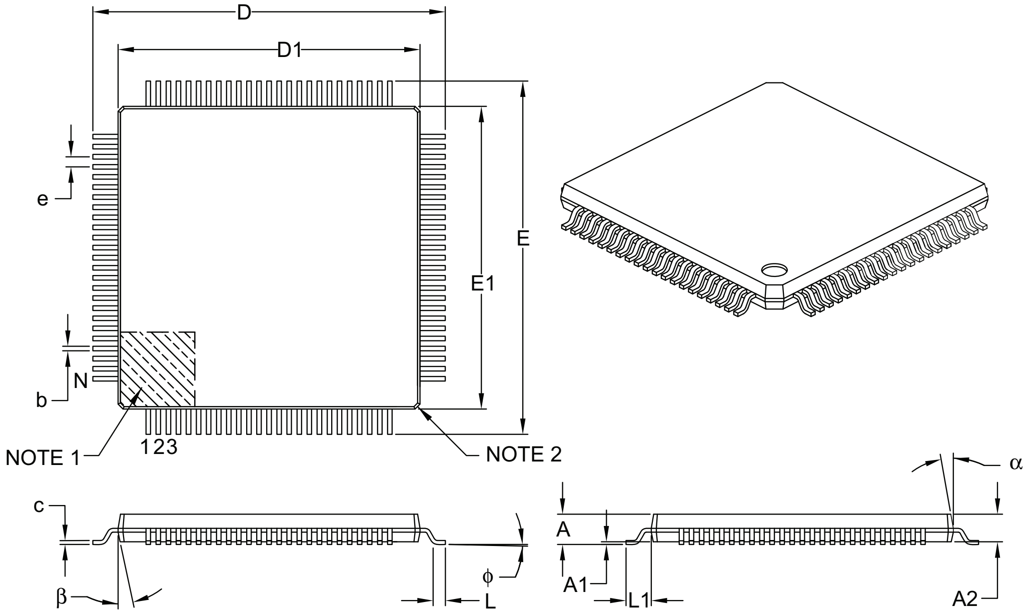
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2092A

100-Lead Plastic Thin Quad Flatpack (PT) – 12x12x1 mm Body, 2.00 mm Footprint [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	100		
Lead Pitch	e	0.40 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	14.00 BSC		
Overall Length	D	14.00 BSC		
Molded Package Width	E1	12.00 BSC		
Molded Package Length	D1	12.00 BSC		
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.13	0.18	0.23
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

Notes:

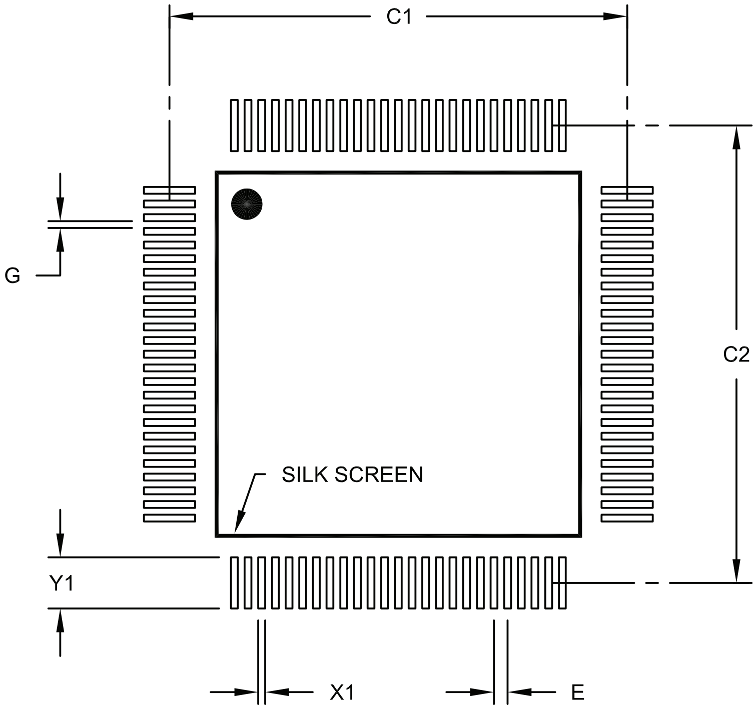
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-100B

100-Lead Plastic Thin Quad Flatpack (PT) – 12x12x1 mm Body, 2.00 mm [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.40 BSC		
Contact Pad Spacing	C1		13.40	
Contact Pad Spacing	C2		13.40	
Contact Pad Width (X100)	X1			0.20
Contact Pad Length (X100)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

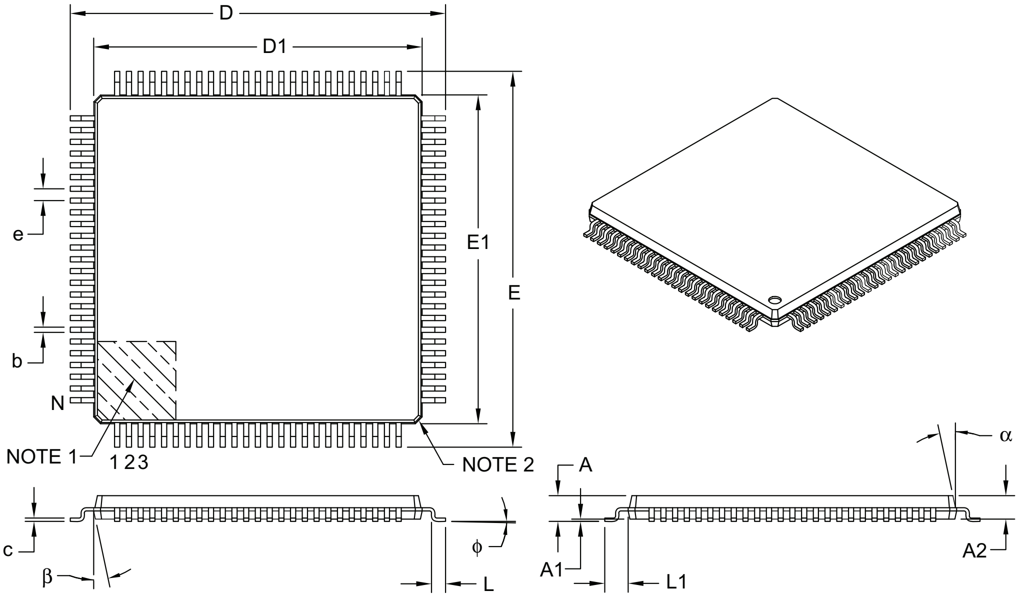
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2100A

100-Lead Plastic Thin Quad Flatpack (PF) – 14x14x1 mm Body, 2.00 mm Footprint [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	100		
Lead Pitch	e	0.50 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	16.00 BSC		
Overall Length	D	16.00 BSC		
Molded Package Width	E1	14.00 BSC		
Molded Package Length	D1	14.00 BSC		
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

Notes:

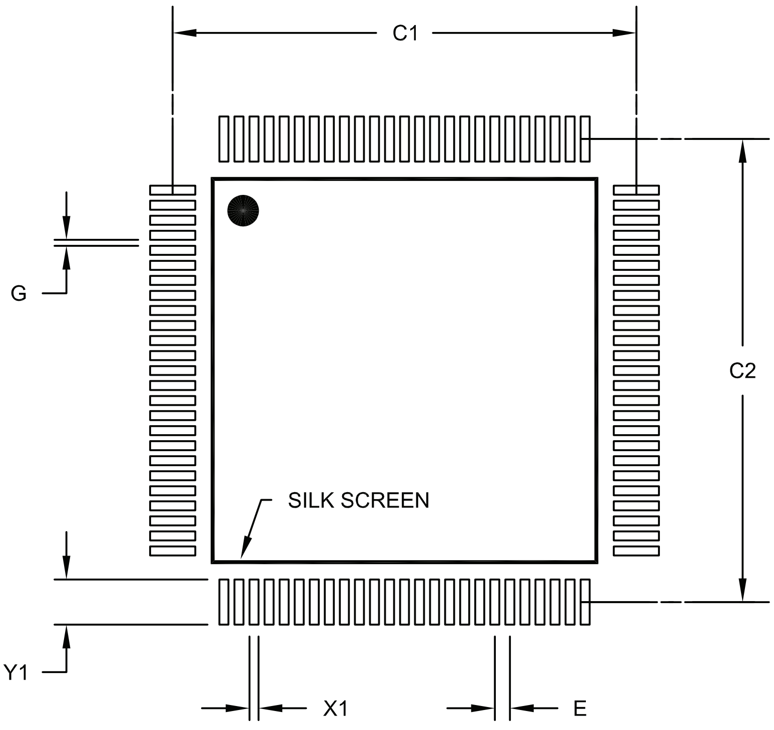
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-110B

100-Lead Plastic Thin Quad Flatpack (PF) – 14x14x1 mm Body, 2.00 mm [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

		Units	MILLIMETERS		
		Dimension Limits	MIN	NOM	MAX
Contact Pitch	E			0.50 BSC	
Contact Pad Spacing	C1			15.40	
Contact Pad Spacing	C2			15.40	
Contact Pad Width (X100)	X1				0.30
Contact Pad Length (X100)	Y1				1.50
Distance Between Pads	G		0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2110A

NOTES:

APPENDIX A: REVISION HISTORY

Revision A (June 2007)

Initial release of this document.

Revision B (March 2008)

This revision includes minor typographical and formatting changes throughout the data sheet text.

The major changes are referenced by their respective section in the following table.

TABLE A-1: MAJOR SECTION UPDATES

Section Name	Update Description
Section 3.0 "Memory Organization"	Updated Change Notification Register Map table title to reflect application with dsPIC33FJXXXMCX10 devices (Table 3-2). Added Change Notification Register Map tables (Table 3-3 and Table 3-4) for dsPIC33FJXXXMCX08 and dsPIC33FJXXXMCX06 devices, respectively. Updated SFR names in 8-Output PWM Register Map (Table 3-9). Updated SFR names in QEI Register Map (Table 3-10). Updated the bit range for AD1CON3 (ADCS<7:0>) in the ADC1 Register Map and added Note 1 (Table 3-17). Updated the bit range for AD2CON3 (ADCS<7:0>) in the ADC2 Register Map (Table 3-18). Updated the Reset value for C1FEN1 (`FFFF`) in the ECAN1 Register Map When C1CTRL1.WIN = `0` or `1` (Table 3-20). Updated the Reset value for C2FEN1 (`FFFF`) in the ECAN2 Register Map When C2CTRL1.WIN = `0` or `1` and updated the title to reflect application for dsPIC33FJXXXMC708/710 devices (Table 3-23). Updated the title for the ECAN2 Register Map When C2CTRL1.WIN = `0` to reflect application toward dsPIC33FJXXXMC708/710 devices (Table 3-24). Updated the title for the ECAN2 Register Map When C2CTRL1.WIN = `1` to reflect application with dsPIC33FJXXXMC708/710 devices (Table 3-25). Updated Reset value for TRISA (`C6FF`) and changed the bit 12 and bit 13 values for all File Names to unimplemented in the PORTA Register Map (Table 3-26). Added PMD Register Map (Table 4-35).
Section 5.0 "Reset"	Added POR and BOR references in Reset Flag Bit Operation (Table 5-1).
Section 7.0 "Direct Memory Access (DMA)"	Updated the table cross-reference in Note 2 in the DMAxREQ register (Register 7-2).

TABLE A-1: MAJOR SECTION UPDATES (CONTINUED)

Section Name	Update Description
Section 8.0 “Oscillator Configuration”	Updated the third clock source item (External Clock) in **Section 8.1.1 “System Clock Sources”.** Added the center frequency in the OSCTUN register for the FRC Tuning bits (TUN<5:0>) value `011111` and updated the center frequency for bits value `011110` (Register 8-4).
Section 15.0 “Motor Control PWM Module”	Removed sections 15.1 through 15.16 (redundant information, which is now available in the related section in the *“dsPIC33F Family Reference Manual”*). Updated SFR names in the PWM Module Block Diagram (Figure 15-1). Updated all register names (Register 16-1 through Register 15-15).
Section 16.0 “Quadrature Encoder Interface (QEI) Module”	Removed sections 16.1 through 16.9 (redundant information, which is now available in the related section in the *“dsPIC33F Family Reference Manual”*). Updated names in Quadrature Encoder Interface Block Diagram (Figure 16-1). Updated register names (Register 16-1 and Register 16-2).
Section 17.0 “Serial Peripheral Interface (SPI)”	Removed redundant information, which is now available in the related section in the *“dsPIC33F Family Reference Manual”*.
Section 18.0 “Inter-Integrated Circuit™ (I^2C™)”	Removed sections 18.3 through 18.14, while retaining the I^2C Block Diagram (Figure 18-1) (redundant information, which is now available in the related section in the *“dsPIC33F Family Reference Manual”*).
Section 19.0 “Universal Asynchronous Receiver Transmitter (UART)”	Removed sections 19.1 through 19.7 (redundant information, which is now available in the related section in the *“dsPIC33F Family Reference Manual”*).
Section 20.0 “Enhanced CAN (ECAN™) Module”	Removed sections 20.4 through 20.6 (redundant information, which is now available in the related section in the *“dsPIC33F Family Reference Manual”*). Updated Baud Rate Prescaler (BRP<5:0>) bit values in the CiCFG1 register (Register 20-9). Changed default bit value from ‘`0`’ to ‘`1`’ for bits 6 through 15 (FLTEN6-FLTEN15) in the CiFEN1 register (Register 20-11).
Section 21.0 “10-Bit/12-Bit Analog-to-Digital Converter (ADC)”	Removed Equation 21-1 (ADC Conversion Clock Period) and Figure 21-3 (ADC Transfer Function (10-Bit Example) in **Section 21.0 “10-bit/12-bit Analog-to-Digital Converter (ADC)”** Updated AN14 and AN15 ADC values in the ADC2 Module Block Diagram (Figure 21-2). Added Note 2 to ADC Conversion Clock Period Block Diagram (Figure 21-3). Added Note to ADxCHS0 register (Register 21-6). Updated ADC Conversion Clock Select bits in the ADxCON3 register from ADCS<**5**:0> to ADCS<**7**:0>. Any references to these bits have also been updated throughout this data sheet (Register 21-3).

TABLE A-1: MAJOR SECTION UPDATES (CONTINUED)

Section Name	Update Description
Section 22.0 “Special Features”	Added a Note after the second paragraph in **Section 22.2 “On-Chip Voltage Regulator”**. Updated address 0xF8000E in the Device Configuration Register Map (Table 22-1). Added FICD register content (BKBUG, COE, JTAGEN and ICS<1:0>) to the dsPIC33F Configuration Bits Description and removed the last two rows (Table 22-2).

Revision C (March 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 V_{DDCORE} and V_{DDCORE}/V_{CAP} to V_{CAP}/V_{DDCORE}

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 V_{SS}.
Section 1.0 "Device Overview"	Updated AV_{DD} 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 Microcontrollers.
Section 4.0 "Memory Organization"	Add Accumulator A and B SFRs (ACCAL, ACCAH, ACCAU, ACCBL, ACCBH and ACCBU) and updated the Reset value for CORCON in the CPU Core Register Map (see Table 4-1). Updated Reset values for IPC3, IPC4, IPC11 and IPC13-IPC15 in the Interrupt Controller Register Map (see Table 4-5). Updated the Reset value for CLKDIV in the System Control Register Map (see Table 4-33).
Section 5.0 "Flash Program Memory"	Updated **Section 5.3 "Programming Operations"** with programming time formula.
Section 9.0 "Oscillator Configuration"	Added Note 2 to the Oscillator System Diagram (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 1 to the FRC Oscillator Tuning (OSCTUN) Register (see Register 9-4).
Section 10.0 "Power-Saving Features"	Added the following registers: • 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)
Section 11.0 "I/O Ports"	Added reference to pin diagrams for I/O pin availability and functionality (see **Section 11.2 "Open-Drain Configuration"**).
Section 18.0 "Serial Peripheral Interface (SPI)"	Added Note 2 to the SPIxCON1 register (see Register 18-2).
Section 20.0 "Universal Asynchronous Receiver Transmitter (UART)"	Updated the UTXINV bit settings in the UxSTA register (see Register 20-2).

TABLE A-2: MAJOR SECTION UPDATES (CONTINUED)

Section Name	Update Description
Section 21.0 "Enhanced CAN (ECAN™) Module"	Changed bit 11 in the ECAN Control Register 1 (CiCTRL1) to Reserved (see Register 21-1). Added the ECAN Filter 15-8 Mask Selection (CiFMSKSEL2) register (see Register 21-19).
Section 22.0 "10-Bit/12-Bit Analog-to-Digital Converter (ADC)"	Replaced the ADC Module Block Diagram (see Figure 22-1) and removed Figure 21-2.
Section 23.0 "Special Features"	Added Note 2 to the Device Configuration Register Map (see Table 23-1).
Section 26.0 "Electrical Characteristics"	Updated Typical values for Thermal Packaging Characteristics (see Table 26-3). Updated Min and Max values for parameter DC12 (RAM Data Retention Voltage) and added Note 4 (see Table 26-4). Updated Power-Down Current Max values for parameters DC60b and DC60c (see Table 26-7). Updated Characteristics for I/O Pin Input Specifications (see Table 26-9). 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 26-12). Added parameter OS42 (GM) to the External Clock Timing Requirements (see Table 26-16). Updated Watchdog Timer Time-out Period parameter SY20 (see Table 26-21).

NOTES:

INDEX

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com, click on Customer Change Notification and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://support.microchip.com

READER RESPONSE

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

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

To: Technical Publications Manager

Total Pages Sent ________

RE: Reader Response

From: Name ____

Company ____

Address ____

City / State / ZIP / Country ____

Telephone: (_______) _________ - _________

FAX: (______) _________ - _________

Application (optional):

Would you like a reply?____Y ____N

Device: dsPIC33FJXXXMCX06/X08/X10 Literature Number: DS70287C

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.

dsPIC 33 FJ 256 MC7 10 T I / PT - XXX

Microchip Trademark
Architecture
Flash Memory Family
Program Memory Size (KB)
Product Group
Pin Count
Tape and Reel Flag (if applicable)
Temperature Range
Package
Pattern

Architecture:	33	=	16-bit Digital Signal Controller
Flash Memory Family:	FJ	=	Flash program memory, 3.3V
Product Group:	MC5 MC7	= =	Motor Control family Motor Control family
Pin Count:	06 08 10	= = =	64-pin 80-pin 100-pin
Temperature Range:	I	=	-40°C to +85°C (Industrial)
Package:	PT PF	= =	10x10 or 12x12 mm TQFP (Thin Quad Flatpack) 14x14 mm TQFP (Thin Quad Flatpack)
Pattern	Three-digit QTP, SQTP, Code or Special Requirements (blank otherwise)		

Examples:

a) dsPIC33FJ64MC706I/PT:
Motor Control dsPIC33, 64 KB program memory, 64-pin, Industrial temp., TQFP package.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

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