



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

High-Performance,
16-bit Digital Signal Controllers

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

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

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

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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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

Trademarks

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


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

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, PICkit, PICDEM, PICDEM.net, PICTail, PIC³² logo, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rLAB, Select Mode, Total Endurance, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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

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

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

 Printed on recycled paper.

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

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



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

High-Performance, 16-bit Digital Signal Controllers

Operating Range:

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

High-Performance DSC CPU:

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

Direct Memory Access (DMA):

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

Timers/Capture/Compare/PWM:

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

Interrupt Controller:

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

Digital I/O:

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

On-Chip Flash and SRAM:

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

System Management:

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

Power Management:

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

Analog-to-Digital Converters (ADCs):

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

Audio Digital-to-Analog Converter (DAC):

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

Comparator Module:

- Two analog comparators with programmable input/output configuration

CMOS Flash Technology:

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

Motor Control Peripherals:

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

Communication Modules:

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

Packaging:

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

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

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

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

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

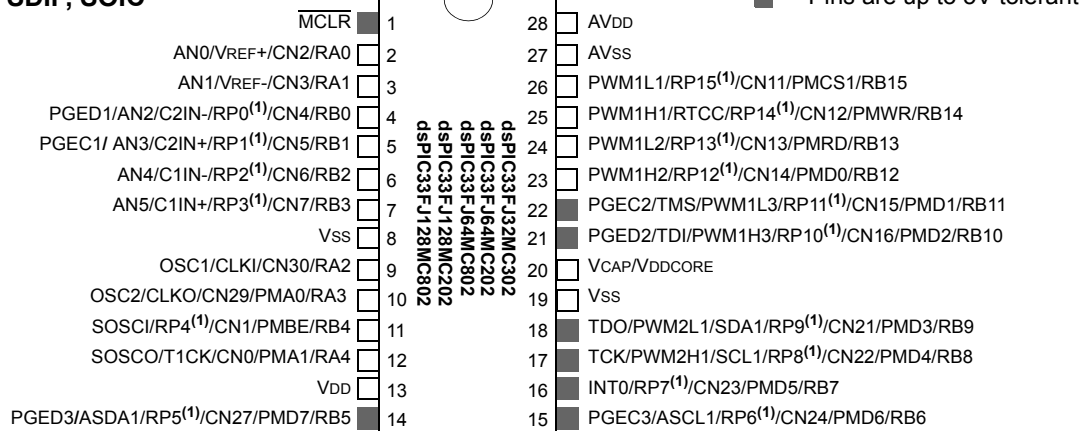
dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 Controller Families

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

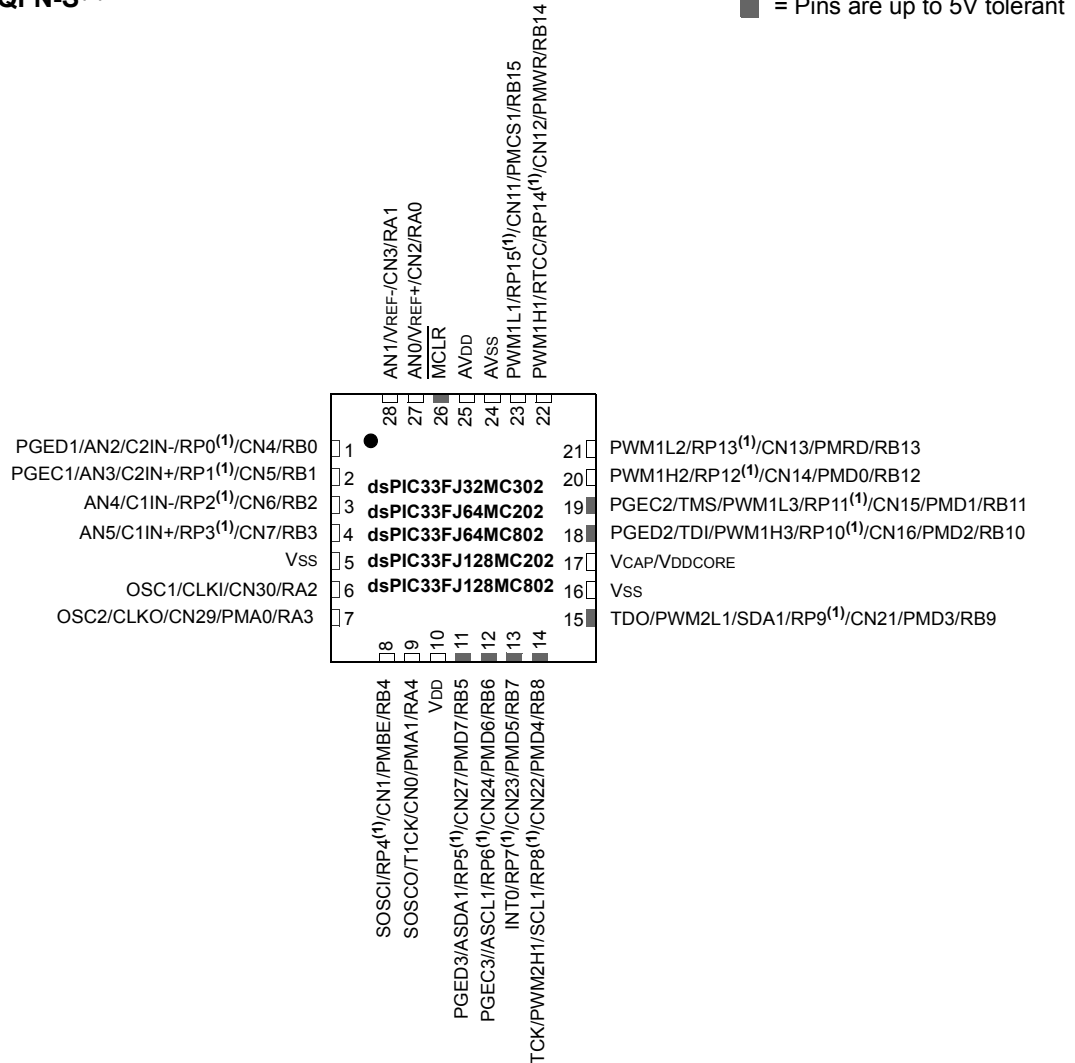
- Note**
- 1: RAM size is inclusive of 2 Kbytes of DMA RAM for all devices except dsPIC33FJ32MC302/304, which include 1 Kbyte of DMA RAM.
 - 2: Only four out of five timers are remappable.
 - 3: Only PWM fault pins are remappable.
 - 4: Only two out of three interrupts are remappable.

Pin Diagrams

28-Pin SDIP, SOIC

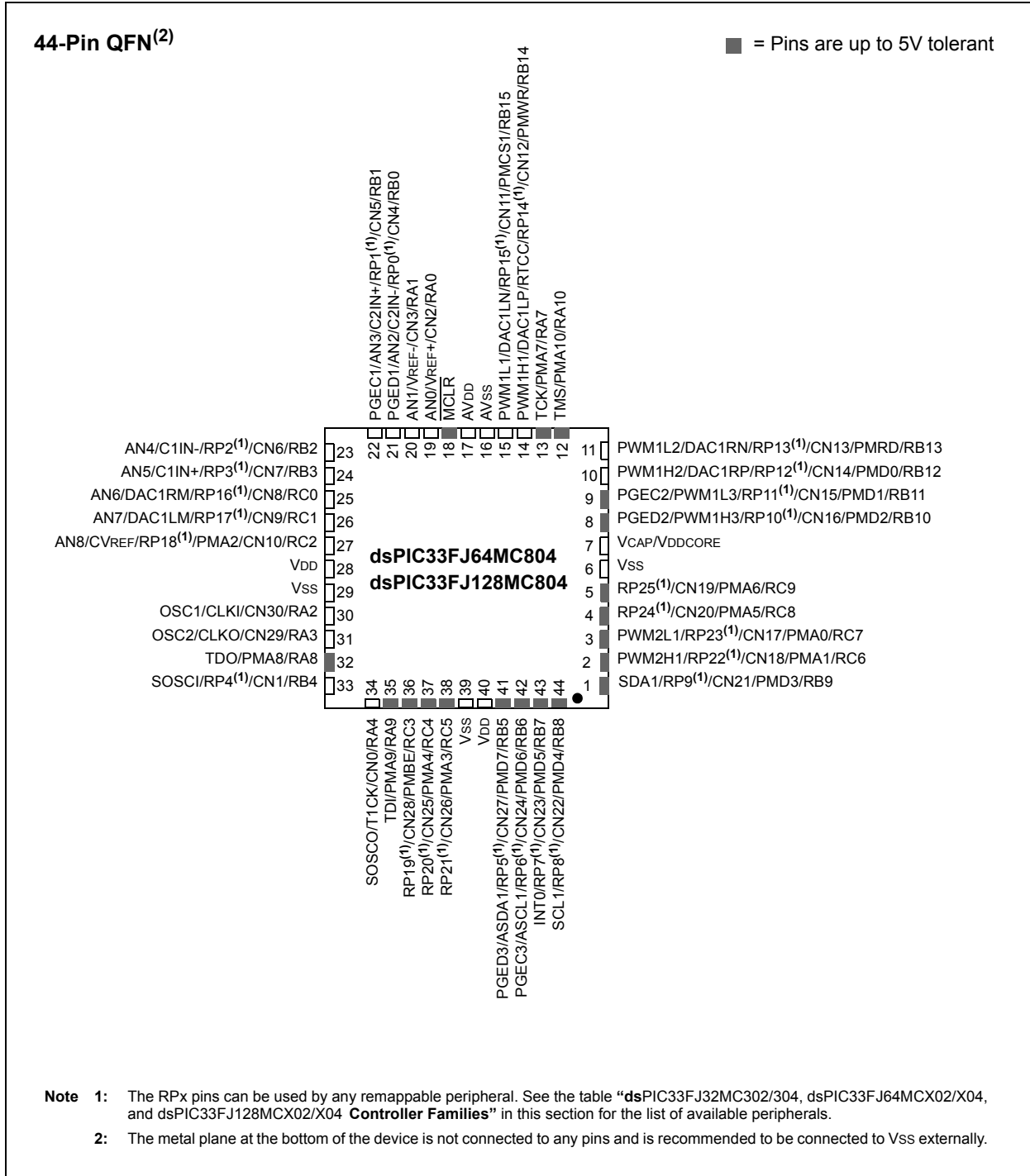


28-Pin QFN-S⁽²⁾

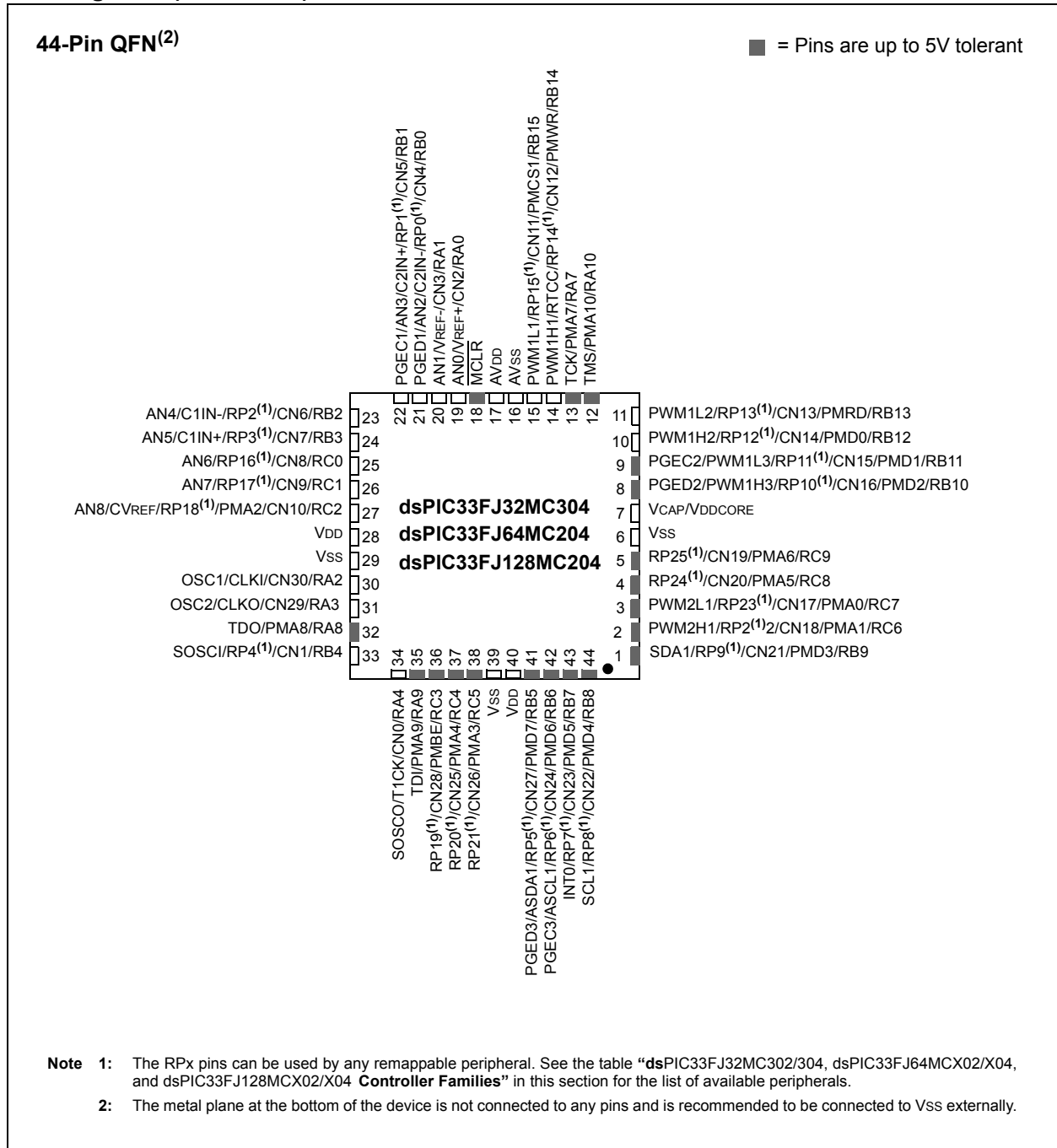


- Note 1:** The RPx pins can be used by any remappable peripheral. See the table “dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 Controller Families” in this section for the list of available peripherals.
- Note 2:** The metal plane at the bottom of the device is not connected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

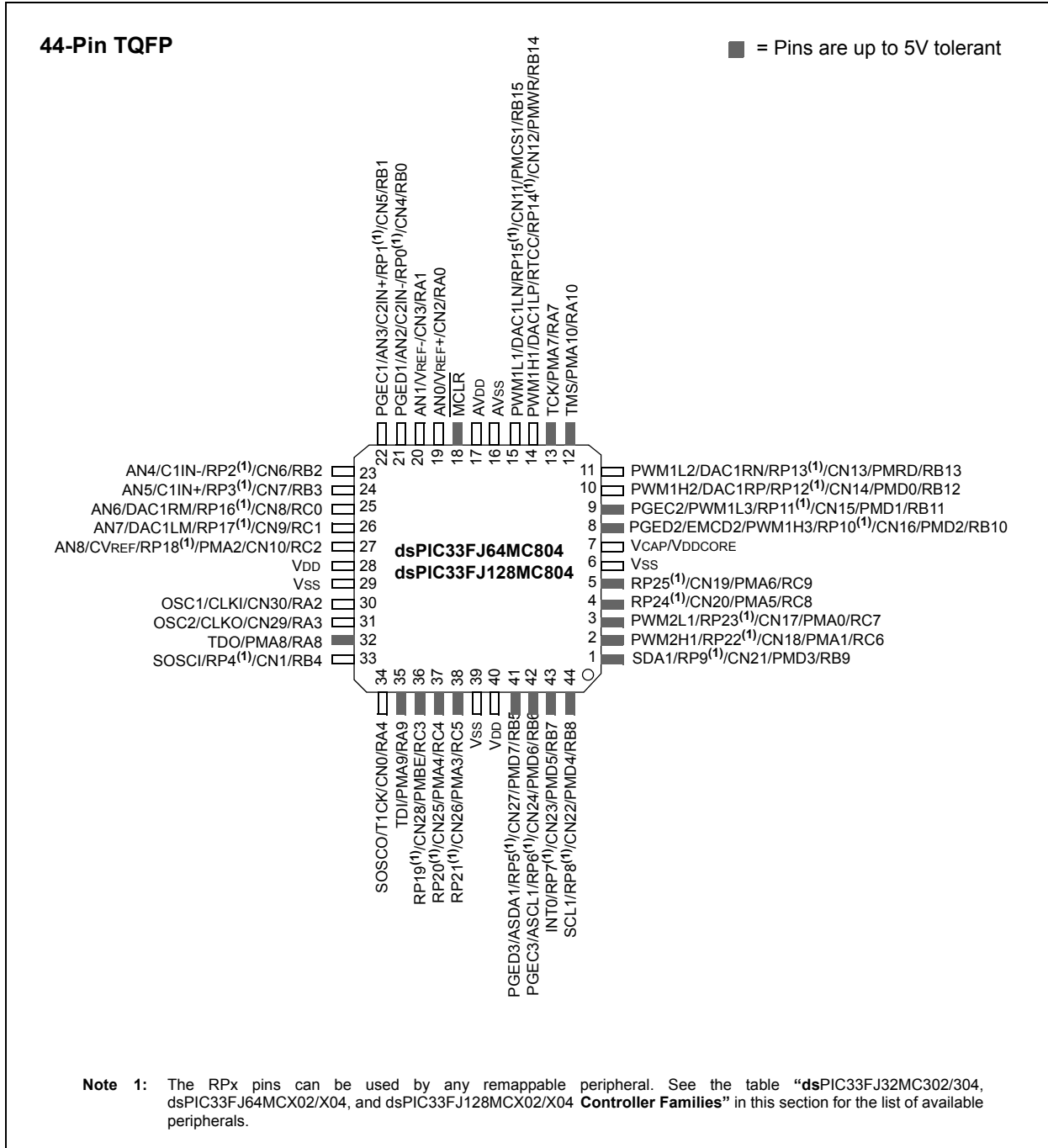
Pin Diagrams (Continued)



Pin Diagrams (Continued)



Pin Diagrams (Continued)



Pin Diagrams (Continued)

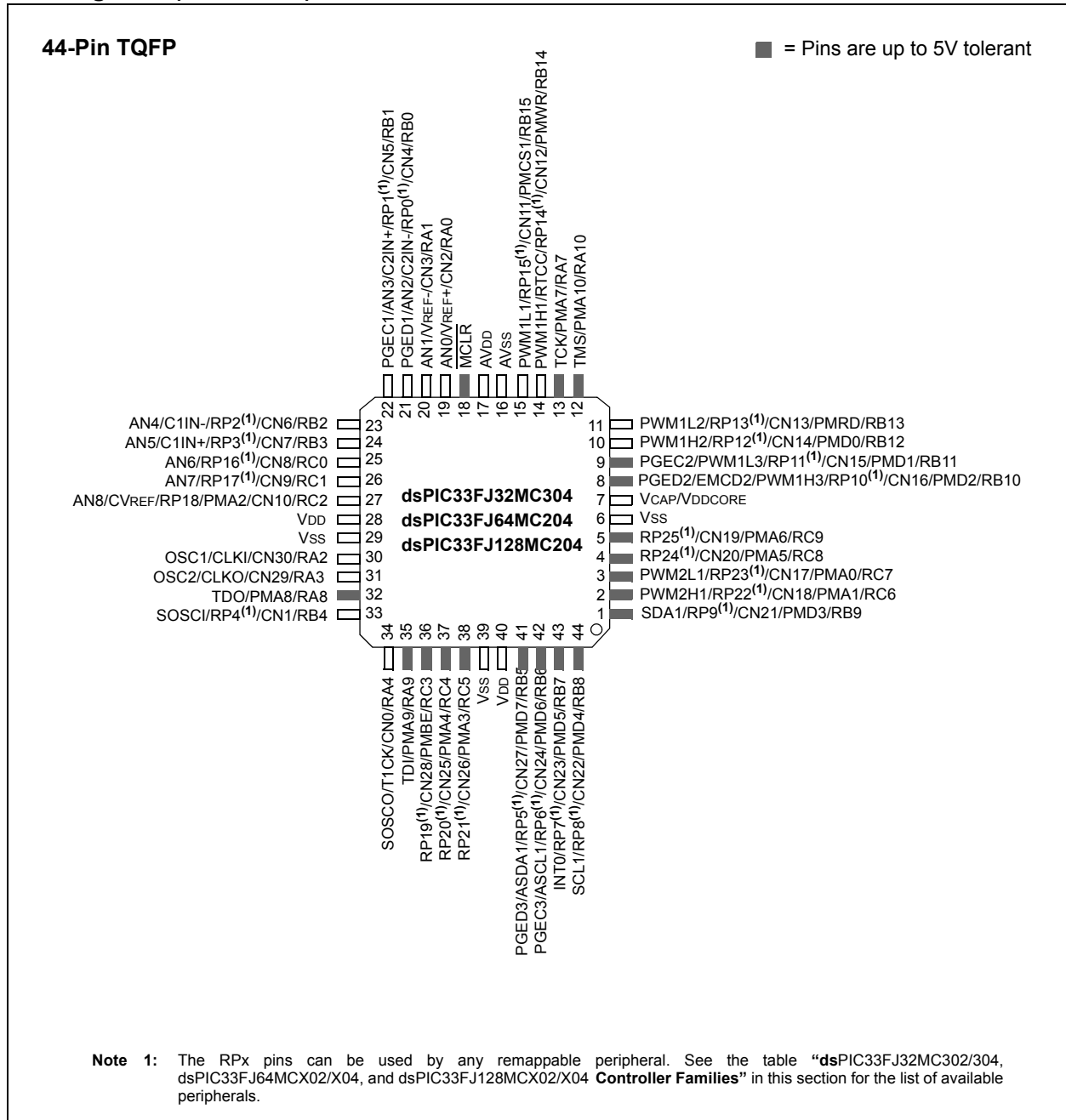


Table of Contents

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

TO OUR VALUED CUSTOMERS

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

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

Most Current Data Sheet

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

<http://www.microchip.com>

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

Errata

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

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

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

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

Customer Notification System

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

NOTES:

1.0 DEVICE OVERVIEW

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

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

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

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

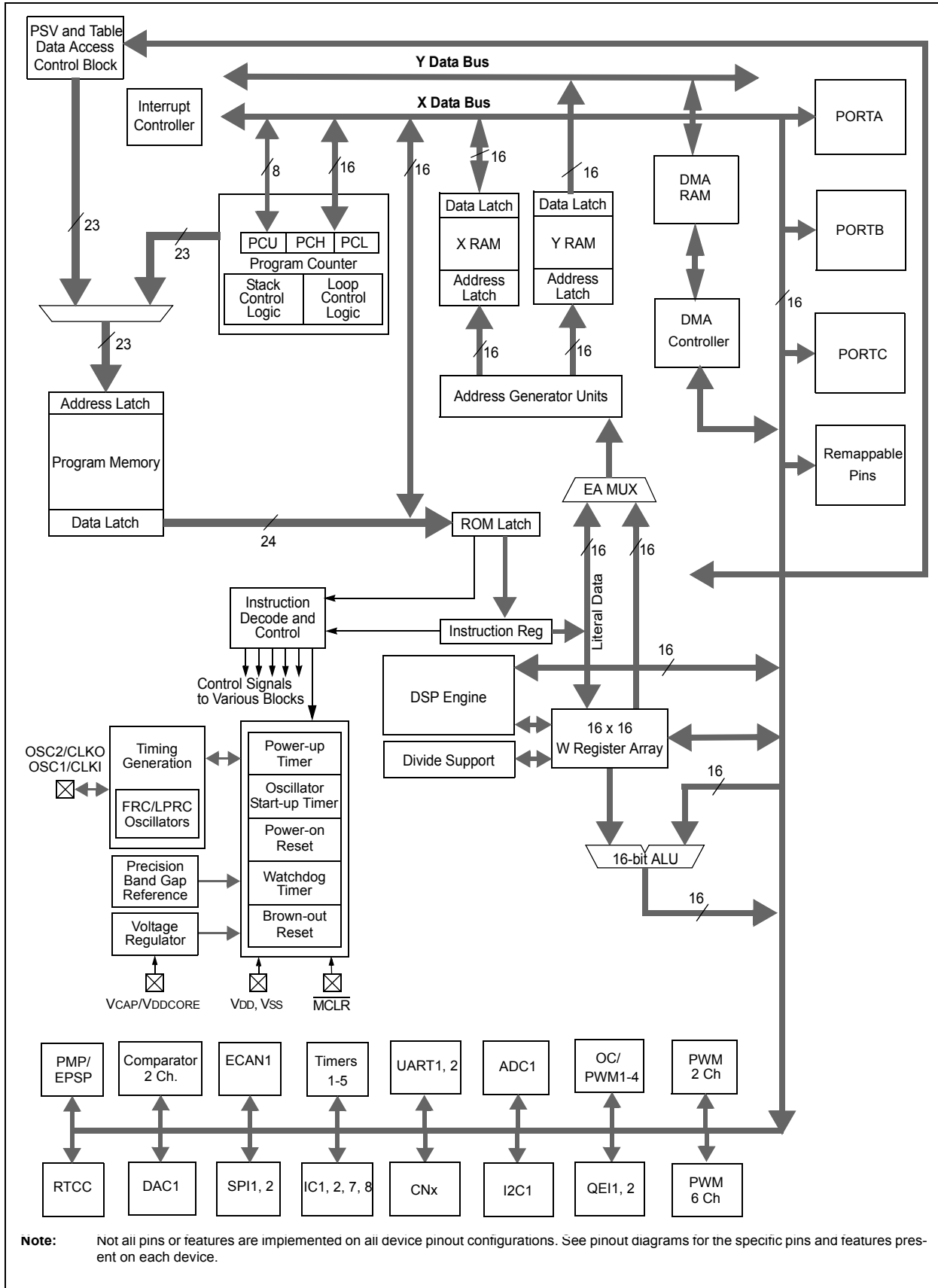


TABLE 1-1: PINOUT I/O DESCRIPTIONS

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

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

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

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

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

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

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

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

NOTES:

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

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

2.1 Basic Connection Requirements

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

- All VDD and VSS pins
(see **Section 2.2 “Decoupling Capacitors”**)
- All AVDD and AVSS pins (regardless if ADC module is not used)
(see **Section 2.2 “Decoupling Capacitors”**)
- VCAP/VDDCORE
(see **Section 2.3 “Capacitor on Internal Voltage Regulator (Vcap/Vddcore)”**)
- MCLR pin
(see **Section 2.4 “Master Clear (MCLR) Pin”**)
- PGECx/PGEDx pins used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes
(see **Section 2.5 “ICSP Pins”**)
- OSC1 and OSC2 pins when external oscillator source is used
(see **Section 2.6 “External Oscillator Pins”**)

Additionally, the following pins may be required:

- VREF+/VREF- pins used when external voltage reference for ADC module is implemented

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

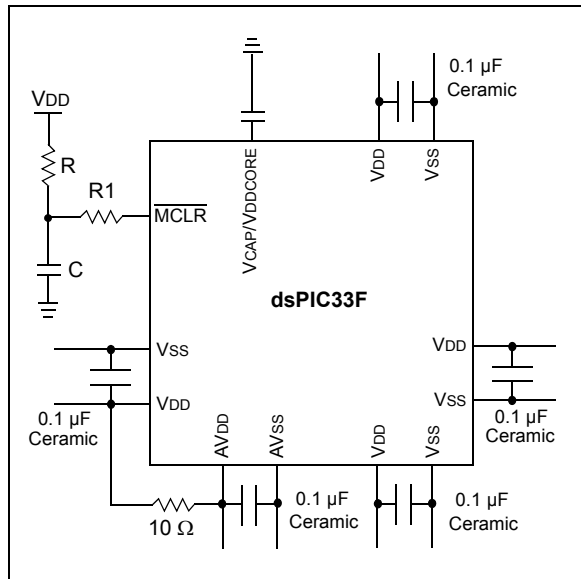
2.2 Decoupling Capacitors

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

Consider the following criteria when using decoupling capacitors:

- **Value and type of capacitor:** Recommendation of 0.1 μF (100 nF), 10-20V. This capacitor should be a low-ESR and have resonance frequency in the range of 20 MHz and higher. It is recommended that ceramic capacitors be used.
- **Placement on the printed circuit board:** The decoupling capacitors should be placed as close to the pins as possible. It is recommended to place the capacitors on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer on the PCB using a via; however, ensure that the trace length from the pin to the capacitor is within one-quarter inch (6 mm) in length.
- **Handling high frequency noise:** If the board is experiencing high frequency noise, upward of tens of MHz, add a second ceramic-type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 μF to 0.001 μF . Place this second capacitor next to the primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible. For example, 0.1 μF in parallel with 0.001 μF .
- **Maximizing performance:** On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum thereby reducing PCB track inductance.

FIGURE 2-1: RECOMMENDED MINIMUM CONNECTION



2.2.1 TANK CAPACITORS

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

2.3 Capacitor on Internal Voltage Regulator (VCAP/VDDCORE)

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

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

2.4 Master Clear (MCLR) Pin

The MCLR pin provides for two specific device functions:

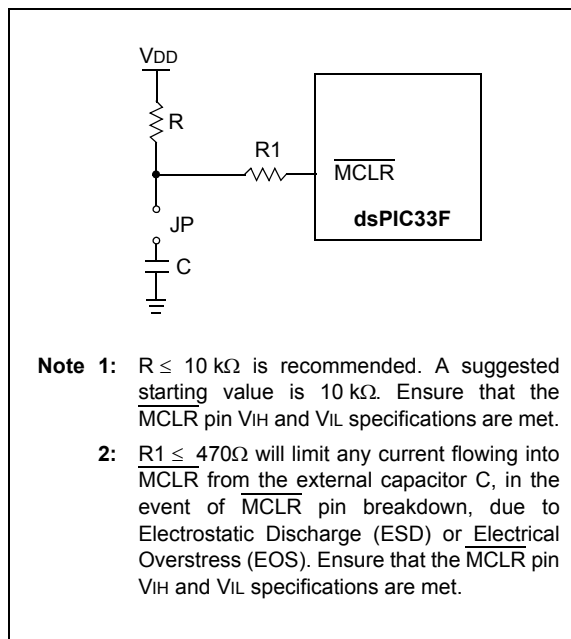
- Device Reset
- Device programming and debugging.

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

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

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

FIGURE 2-2: EXAMPLE OF MCLR PIN CONNECTIONS



2.5 ICSP Pins

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

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

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

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

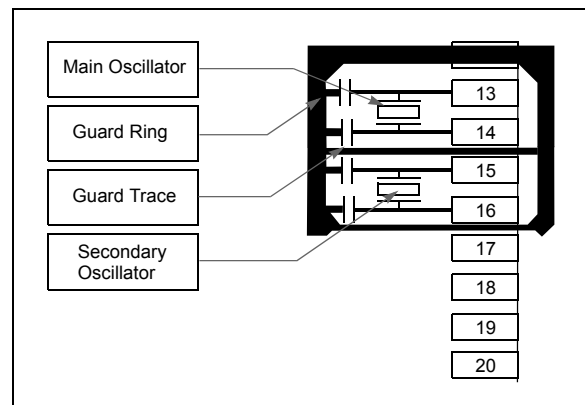
- “MPLAB® ICD 2 In-Circuit Debugger User’s Guide” DS51331
- “Using MPLAB® ICD 2” (poster) DS51265
- “MPLAB® ICD 2 Design Advisory” DS51566
- “Using MPLAB® ICD 3” (poster) DS51765
- “MPLAB® ICD 3 Design Advisory” DS51764
- “MPLAB® REAL ICE™ In-Circuit Debugger User’s Guide” DS51616
- “Using MPLAB® REAL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

2.6 External Oscillator Pins

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

The oscillator circuit should be placed on the same side of the board as the device. Also, place the oscillator circuit close to the respective oscillator pins, not exceeding one-half inch (12 mm) distance between them. The load capacitors should be placed next to the oscillator itself, on the same side of the board. Use a grounded copper pour around the oscillator circuit to isolate them from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed. A suggested layout is shown in Figure 2-3.

FIGURE 2-3: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT



2.7 Oscillator Value Conditions on Device Start-up

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

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

2.8 Configuration of Analog and Digital Pins During ICSP Operations

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

The bits in this register that correspond to the A/D pins that are initialized by MPLAB ICD 2, ICD 3, or REAL ICE, must not be cleared by the user application firmware; otherwise, communication errors will result between the debugger and the device.

If your application needs to use certain A/D pins as analog input pins during the debug session, the user application must clear the corresponding bits in the AD1PCFGL register during initialization of the ADC module.

When MPLAB ICD 2, ICD 3, or REAL ICE is used as a programmer, the user application firmware must correctly configure the AD1PCFGL register. Automatic initialization of this register is only done during debugger operation. Failure to correctly configure the register(s) will result in all A/D pins being recognized as analog input pins, resulting in the port value being read as a logic '0', which may affect user application functionality.

2.9 Unused I/Os

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

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

3.0 CPU

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

3.1 Overview

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 CPU module has a 16-bit (data) modified Harvard architecture with an enhanced instruction set, including significant support for DSP. The CPU has a 24-bit instruction word with a variable length opcode field. The Program Counter (PC) is 23 bits wide and addresses up to 4M x 24 bits of user program memory space. The actual amount of program memory implemented varies by device. A single-cycle instruction prefetch mechanism is used to help maintain throughput and provides predictable execution. All instructions execute in a single cycle, with the exception of instructions that change the program flow, the double-word move (MOV.D) instruction and the table instructions. Overhead-free program loop constructs are supported using the DO and REPEAT instructions, both of which are interruptible at any time.

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

There are two classes of instruction in the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 devices: MCU and DSP. These two instruction classes are seamlessly integrated into a single CPU. The instruction set includes many addressing modes and is designed for optimum C compiler efficiency. For most instructions, the dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 is capable of executing a data (or program data) memory read, a working register (data) read, a data memory write and a program (instruction) memory read per instruction cycle. As a result, three parameter instructions can be supported, allowing $A + B = C$ operations to be executed in a single cycle.

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

3.2 Data Addressing Overview

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

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

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

3.3 DSP Engine Overview

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

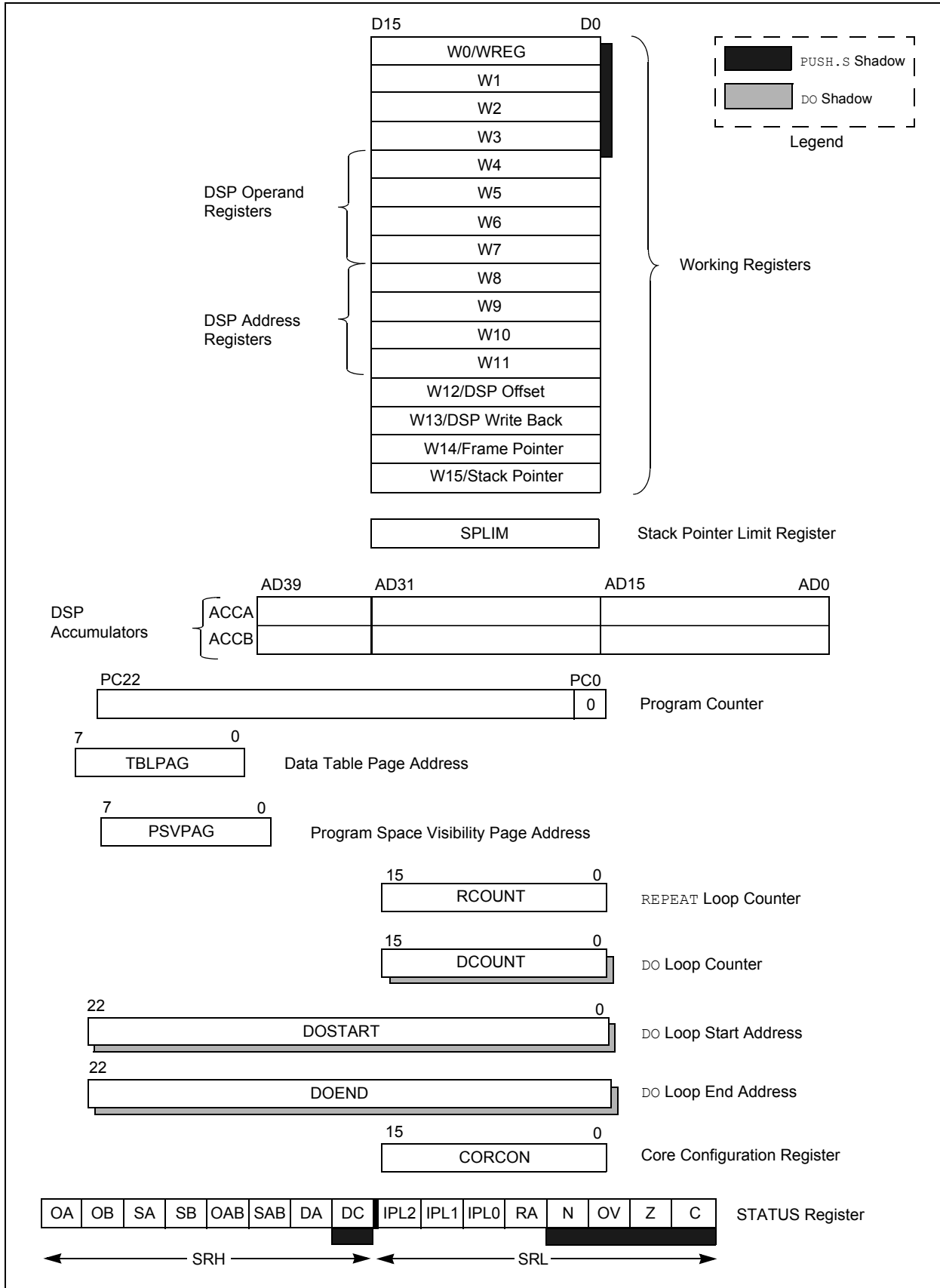
3.4 Special MCU Features

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

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

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

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



3.5 CPU Control Registers

REGISTER 3-1: SR: CPU STATUS REGISTER

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

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

Legend:

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

- bit 15 **OA:** Accumulator A Overflow Status bit
1 = Accumulator A overflowed
0 = Accumulator A has not overflowed
- bit 14 **OB:** Accumulator B Overflow Status bit
1 = Accumulator B overflowed
0 = Accumulator B has not overflowed
- bit 13 **SA:** Accumulator A Saturation 'Sticky' Status bit⁽¹⁾
1 = Accumulator A is saturated or has been saturated at some time
0 = Accumulator A is not saturated
- bit 12 **SB:** Accumulator B Saturation 'Sticky' Status bit⁽¹⁾
1 = Accumulator B is saturated or has been saturated at some time
0 = Accumulator B is not saturated
- bit 11 **OAB:** OA || OB Combined Accumulator Overflow Status bit
1 = Accumulators A or B have overflowed
0 = Neither Accumulators A or B have overflowed
- bit 10 **SAB:** SA || SB Combined Accumulator (Sticky) Status bit⁽⁴⁾
1 = Accumulators A or B are saturated or have been saturated at some time in the past
0 = Neither Accumulator A or B are saturated
- bit 9 **DA:** DO Loop Active bit
1 = DO loop in progress
0 = DO loop not in progress
- bit 8 **DC:** MCU ALU Half Carry/Borrow bit
1 = A carry-out from the 4th low-order bit (for byte-sized data) or 8th low-order bit (for word-sized data) of the result occurred
0 = No carry-out from the 4th low-order bit (for byte-sized data) or 8th low-order bit (for word-sized data) of the result occurred

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

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

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

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

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

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

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

- 2:** The IPL<2:0> bits are concatenated with the IPL<3> bit (CORCON<3>) to form the CPU Interrupt Priority Level. The value in parentheses indicates the IPL if IPL<3> = 1. User interrupts are disabled when IPL<3> = 1.
- 3:** The IPL<2:0> Status bits are read only when NSTDIS = 1 (INTCON1<15>).
- 4:** This bit can be read or cleared (not set). Clearing this bit clears SA and SB.

REGISTER 3-2: CORCON: CORE CONTROL REGISTER

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

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

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

- bit 15-13 **Unimplemented:** Read as '0'
- bit 12 **US:** DSP Multiply Unsigned/Signed Control bit
 1 = DSP engine multiplies are unsigned
 0 = DSP engine multiplies are signed
- bit 11 **EDT:** Early DO Loop Termination Control bit⁽¹⁾
 1 = Terminate executing DO loop at end of current loop iteration
 0 = No effect
- bit 10-8 **DL<2:0>:** DO Loop Nesting Level Status bits
 111 = 7 DO loops active
 •
 •
 •
 001 = 1 DO loop active
 000 = 0 DO loops active
- bit 7 **SATA:** ACCA Saturation Enable bit
 1 = Accumulator A saturation enabled
 0 = Accumulator A saturation disabled
- bit 6 **SATB:** ACCB Saturation Enable bit
 1 = Accumulator B saturation enabled
 0 = Accumulator B saturation disabled
- bit 5 **SATDW:** Data Space Write from DSP Engine Saturation Enable bit
 1 = Data space write saturation enabled
 0 = Data space write saturation disabled
- bit 4 **ACCSAT:** Accumulator Saturation Mode Select bit
 1 = 9.31 saturation (super saturation)
 0 = 1.31 saturation (normal saturation)
- bit 3 **IPL3:** CPU Interrupt Priority Level Status bit 3⁽²⁾
 1 = CPU interrupt priority level is greater than 7
 0 = CPU interrupt priority level is 7 or less
- bit 2 **PSV:** Program Space Visibility in Data Space Enable bit
 1 = Program space visible in data space
 0 = Program space not visible in data space

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

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

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

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

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

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

3.6 Arithmetic Logic Unit (ALU)

The dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 ALU is 16 bits wide and is capable of addition, subtraction, bit shifts and logic operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. Depending on the operation, the ALU can affect the values of the Carry (C), Zero (Z), Negative (N), Overflow (OV) and Digit Carry (DC) Status bits in the SR register. The C and DC Status bits operate as Borrow and Digit Borrow bits, respectively, for subtraction operations.

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

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

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

3.6.1 MULTIPLIER

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

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

3.6.2 DIVIDER

The divide block supports 32-bit/16-bit and 16-bit/16-bit signed and unsigned integer divide operations with the following data sizes:

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

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

3.7 DSP Engine

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

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

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

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

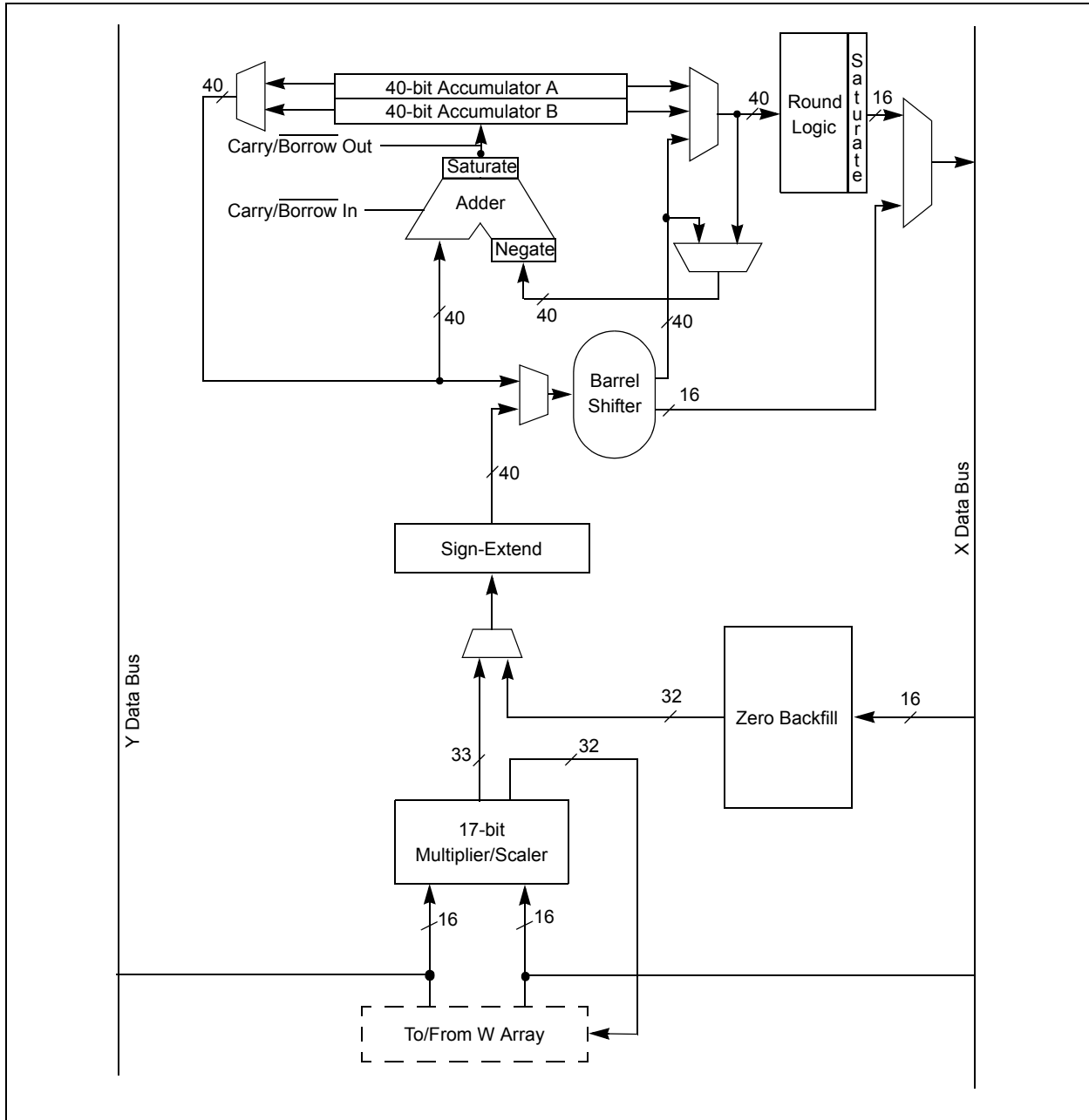
- Fractional or integer DSP multiply (IF)
- Signed or unsigned DSP multiply (US)
- Conventional or convergent rounding (RND)
- Automatic saturation on/off for ACCA (SATA)
- Automatic saturation on/off for ACCB (SATB)
- Automatic saturation on/off for writes to data memory (SATDW)
- Accumulator Saturation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

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

TABLE 3-1: DSP INSTRUCTIONS SUMMARY

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

FIGURE 3-3: DSP ENGINE BLOCK DIAGRAM



3.7.1 MULTIPLIER

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

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

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

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

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

3.7.2 DATA ACCUMULATORS AND ADDER/SUBTRACTOR

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

3.7.2.1 Adder/Subtractor, Overflow and Saturation

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

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

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

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

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

Six STATUS register bits support saturation and overflow:

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

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

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

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

The device supports three Saturation and Overflow modes:

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

3.7.3 ACCUMULATOR 'WRITE BACK'

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

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

3.7.3.1 Round Logic

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

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

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

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

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

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

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

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

3.7.3.2 Data Space Write Saturation

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

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

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

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

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

3.7.4 BARREL SHIFTER

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

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

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

4.0 MEMORY ORGANIZATION

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

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

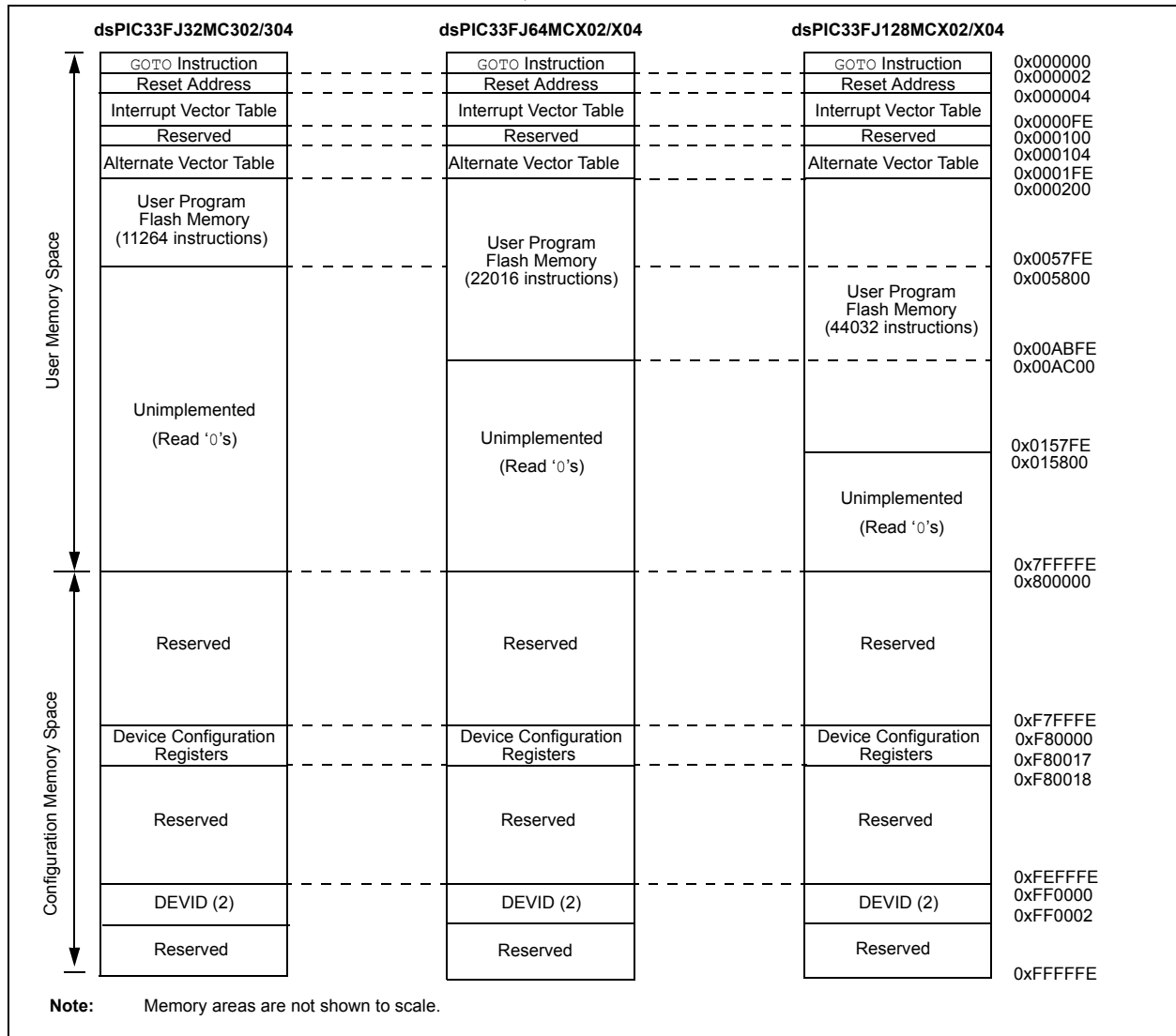
4.1 Program Address Space

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

User application access to the program memory space is restricted to the lower half of the address range (0x000000 to 0x7FFFFFFF). The exception is the use of TBLRD/TBLWT operations, which use TBLPAG<7> to permit access to the Configuration bits and Device ID sections of the configuration memory space.

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

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



4.1.1 PROGRAM MEMORY ORGANIZATION

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

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

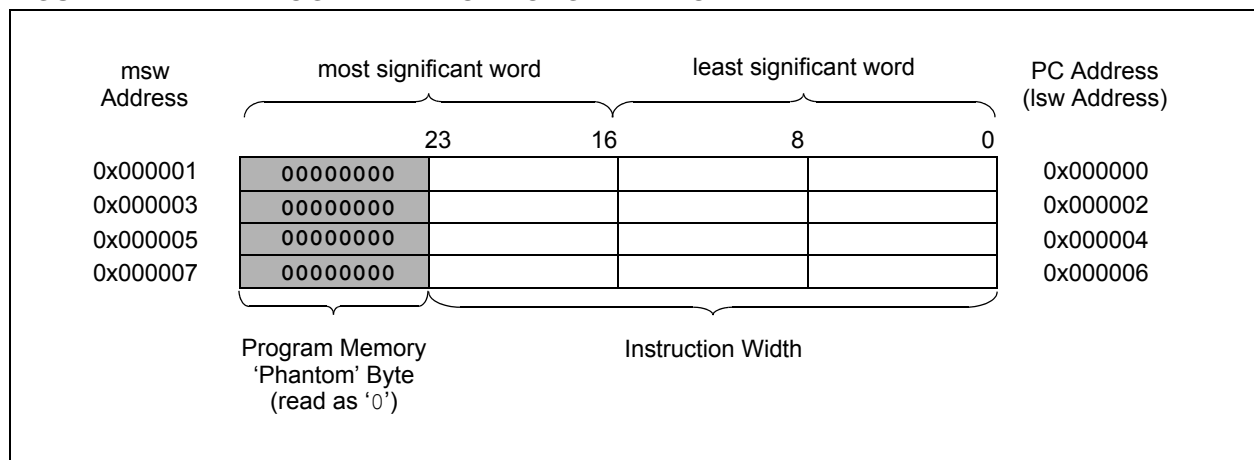
addresses between 0x00000 and 0x000200 for hard-coded program execution vectors. A hardware Reset vector is provided to redirect code execution from the default value of the PC on device Reset to the actual start of code. A GOTO instruction is programmed by the user application at 0x000000, with the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

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

4.1.2 INTERRUPT AND TRAP VECTORS

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

FIGURE 4-2: PROGRAM MEMORY ORGANIZATION



4.2 Data Address Space

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

All Effective Addresses (EAs) in the data memory space are 16 bits wide and point to bytes within the data space. This arrangement gives a data space address range of 64 Kbytes or 32K words. The lower half of the data memory space (that is, when $EA<15> = 0$) is used for implemented memory addresses, while the upper half ($EA<15> = 1$) is reserved for the Program Space Visibility area (see **Section 4.6.3 “Reading Data From Program Memory Using Program Space Visibility”**).

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

4.2.1 DATA SPACE WIDTH

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

4.2.2 DATA MEMORY ORGANIZATION AND ALIGNMENT

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

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

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

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

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

4.2.3 SFR SPACE

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

SFRs are distributed among the modules that they control, and are generally grouped together by module. Much of the SFR space contains unused addresses; these are read as '0'.

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

4.2.4 NEAR DATA SPACE

The 8 Kbyte area between 0x0000 and 0x1FFF is referred to as the near data space. Locations in this space are directly addressable via a 13-bit absolute address field within all memory direct instructions. Additionally, the whole data space is addressable using MOV instructions, which support Memory Direct Addressing mode with a 16-bit address field, or by using Indirect Addressing mode using a working register as an address pointer.

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

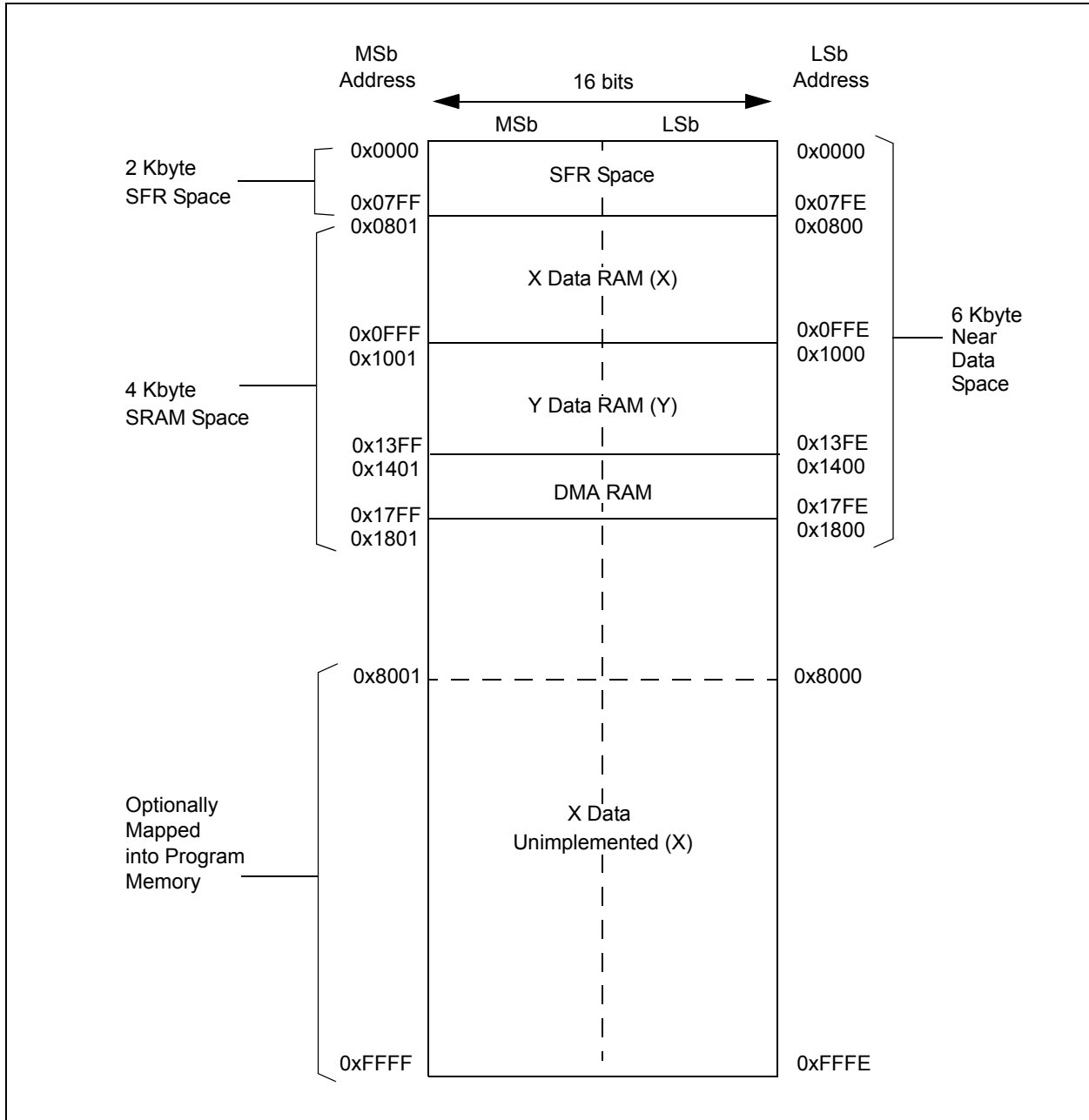


FIGURE 4-4: DATA MEMORY MAP FOR dsPIC33FJ128MC202/204 AND dsPIC33FJ64MC202/204 DEVICES WITH 8 KB RAM

