

Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



MICROCHIP

dsPIC30F6010A/6015
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, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC³² logo, rPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


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

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

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

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

© 2011, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

ISBN: 978-1-60932-953-2

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.

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

High-Performance, 16-bit Digital Signal Controllers

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the “*dsPIC30F Family Reference Manual*” (DS70046). For more information on the device instruction set and programming, refer to the “*16-bit MCU and DSC Programmer’s Reference Manual*” (DS70157).

High-Performance Modified RISC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set architecture with flexible Addressing modes
- 83 base instructions
- 24-bit wide instructions, 16-bit wide data path
- 144 Kbytes on-chip Flash program space (Instruction words)
- 8 Kbytes of on-chip data RAM
- 4 Kbytes of nonvolatile data EEPROM
- Up to 30 MIPS operation:
 - DC to 40 MHz external clock input
 - 4 MHz-10 MHz oscillator input with PLL active (4x, 8x, 16x)
 - 7.37 MHz internal RC with PLL active (4x, 8x, 16x)
- 44 interrupt sources:
 - Five external interrupt sources
 - Eight user selectable priority levels for each interrupt source
 - Four processor trap sources
- 16 x 16-bit working register array

DSP Engine Features:

- Dual data fetch
- Accumulator write-back for DSP operations
- Modulo and Bit-Reversed Addressing modes
- Two, 40-bit wide accumulators with optional saturation logic
- 17-bit x 17-bit single-cycle hardware fractional/integer multiplier
- All DSP instructions single cycle
- ± 16 -bit single-cycle shift

Peripheral Features:

- High-current sink/source I/O pins: 25 mA/25 mA
- Timer module with programmable prescaler:
 - Five 16-bit timers/counters; optionally pair 16-bit timers into 32-bit timer modules
- 16-bit Capture input functions
- 16-bit Compare/PWM output functions
- 3-wire SPI modules (supports 4 Frame modes)
- I²C™ module supports Multi-Master/Slave mode and 7-bit/10-bit addressing
- Two UART modules with FIFO Buffers
- Two CAN modules, 2.0B compliant (dsPIC306010A)
- One CAN module, 2.0B compliant (dsPIC306015)

Motor Control PWM Module Features:

- Eight PWM output channels:
 - Complementary or Independent Output modes
 - Edge and Center-Aligned modes
- Four duty cycle generators
- Dedicated time base
- Programmable output polarity
- Dead-Time control for Complementary mode
- Manual output control
- Trigger for A/D conversions

Quadrature Encoder Interface Module Features:

- 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 Features:

- 10-bit Analog-to-Digital Converter (ADC) with four S/H Inputs:
 - 1 Msps conversion rate
 - 16 input channels
 - Conversion available during Sleep and Idle
- Programmable Brown-out Reset

dsPIC30F6010A/6015

Special Microcontroller Features:

- Enhanced Flash program memory:
 - 10,000 erase/write cycle (min.) for industrial temperature range, 100K (typical)
- Data EEPROM memory:
 - 100,000 erase/write cycle (min.) for industrial temperature range, 1M (typical)
- Self-reprogrammable under software control
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Flexible Watchdog Timer (WDT) with on-chip, low-power RC oscillator for reliable operation
- Fail-Safe Clock Monitor operation detects clock failure and switches to on-chip, low-power RC oscillator
- Programmable code protection
- In-Circuit Serial Programming™ (ICSP™)
- Selectable Power Management modes
 - Sleep, Idle and Alternate Clock modes

CMOS Technology:

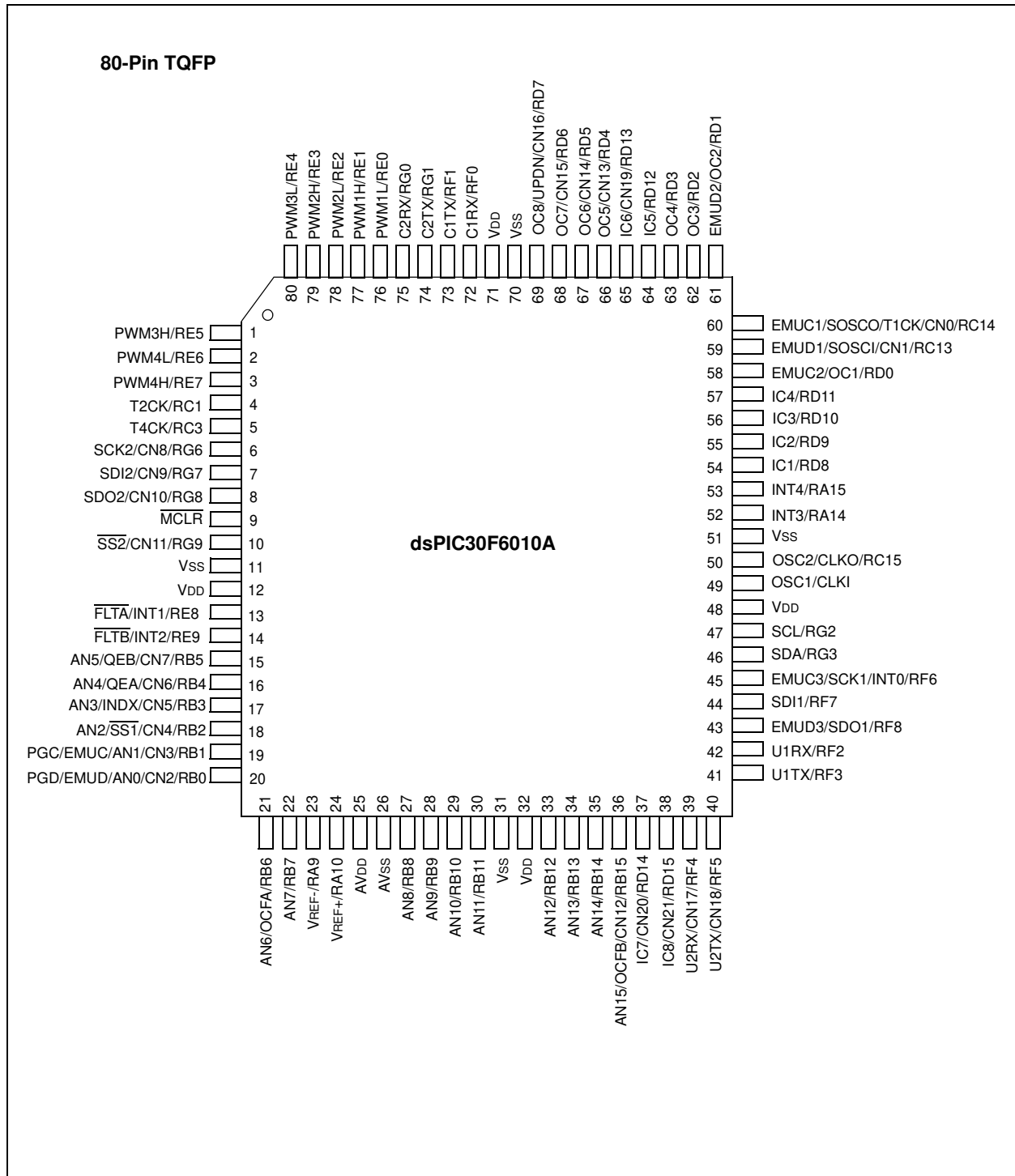
- Low-power, high-speed Flash technology
- Wide operating voltage range (2.5V to 5.5V)
- Industrial and Extended temperature ranges
- Low-power consumption

dsPIC30F Motor Control and Power Conversion Family

Device	Pins	Program Mem. Bytes/Instructions	SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp/Std PWM	Motor Control PWM	A/D 10-bit 1 Msps	Quad Enc	UART	SPI	I ² C™	CAN
dsPIC30F6010A	80	144K/48K	8192	4096	5	8	8	8 ch	16 ch	Yes	2	2	1	2
dsPIC30F6015	64	144K/48K	8192	4096	5	8	8	8 ch	16 ch	Yes	2	2	1	1

dsPIC30F6010A/6015

Pin Diagram



dsPIC30F6010A/6015

Pin Diagram

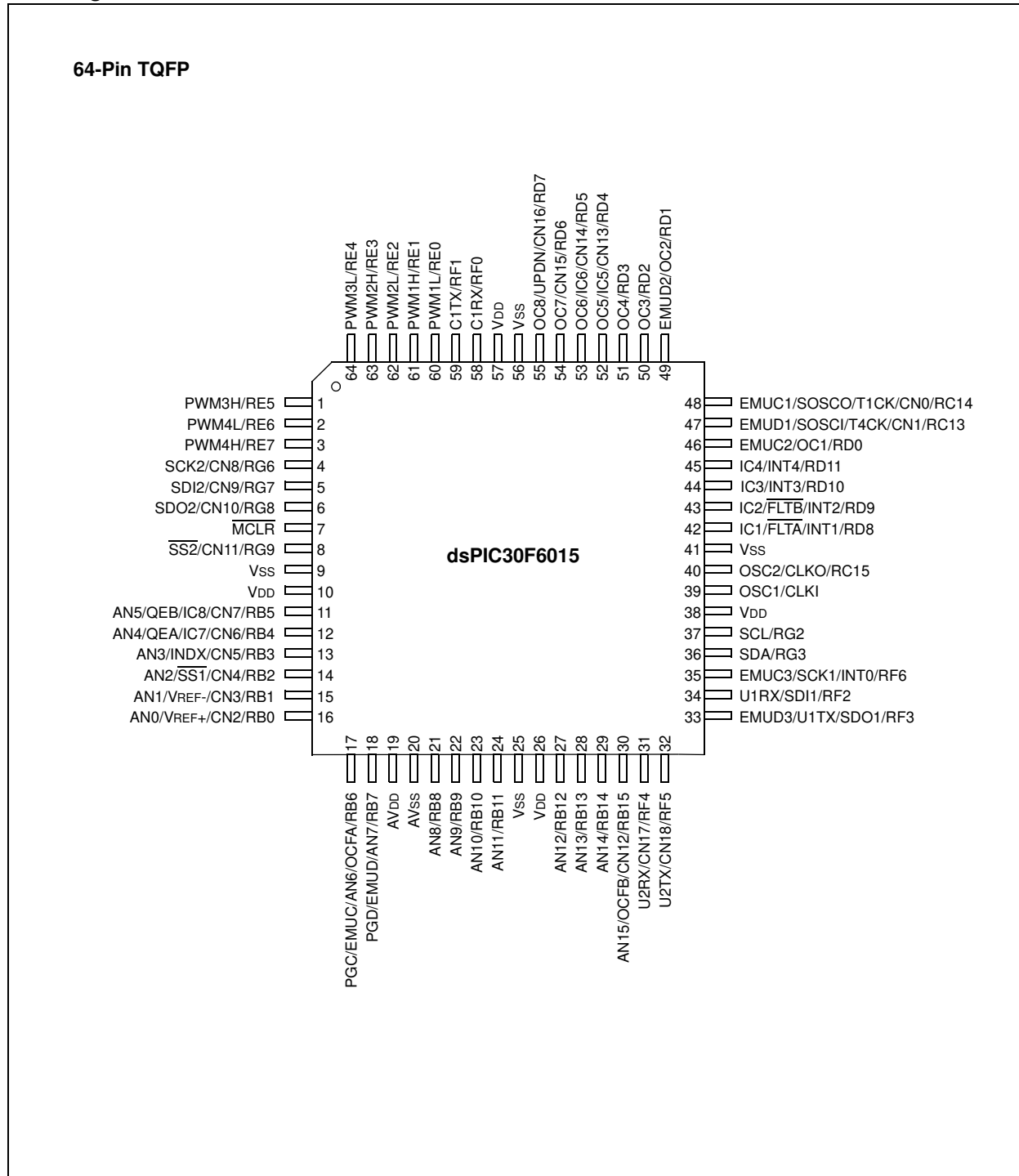


Table of Contents

1.0	Device Overview	9
2.0	CPU Architecture Overview.....	15
3.0	Memory Organization	23
4.0	Address Generator Units.....	35
5.0	Interrupts	41
6.0	Flash Program Memory.....	49
7.0	Data EEPROM Memory	55
8.0	I/O Ports	59
9.0	Timer1 Module	65
10.0	Timer2/3 Module	69
11.0	Timer4/5 Module	77
12.0	Input Capture Module	81
13.0	Output Compare Module.....	85
14.0	Quadrature Encoder Interface (QEI) Module	91
15.0	Motor Control PWM Module.....	97
16.0	SPI Module.....	107
17.0	I2C™ Module	111
18.0	Universal Asynchronous Receiver Transmitter (UART) Module	119
19.0	CAN Module.....	127
20.0	10-bit High-Speed Analog-to-Digital Converter (ADC) Module	139
21.0	System Integration	151
22.0	Instruction Set Summary	165
23.0	Development Support.....	173
24.0	Electrical Characteristics	177
25.0	Packaging Information.....	217
	Appendix A: Revision History.....	225
	Index	227
	The Microchip Web Site.....	233
	Customer Change Notification Service	233
	Customer Support	233
	Reader Response	234
	Product Identification System	235

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.

dsPIC30F6010A/6015

NOTES:

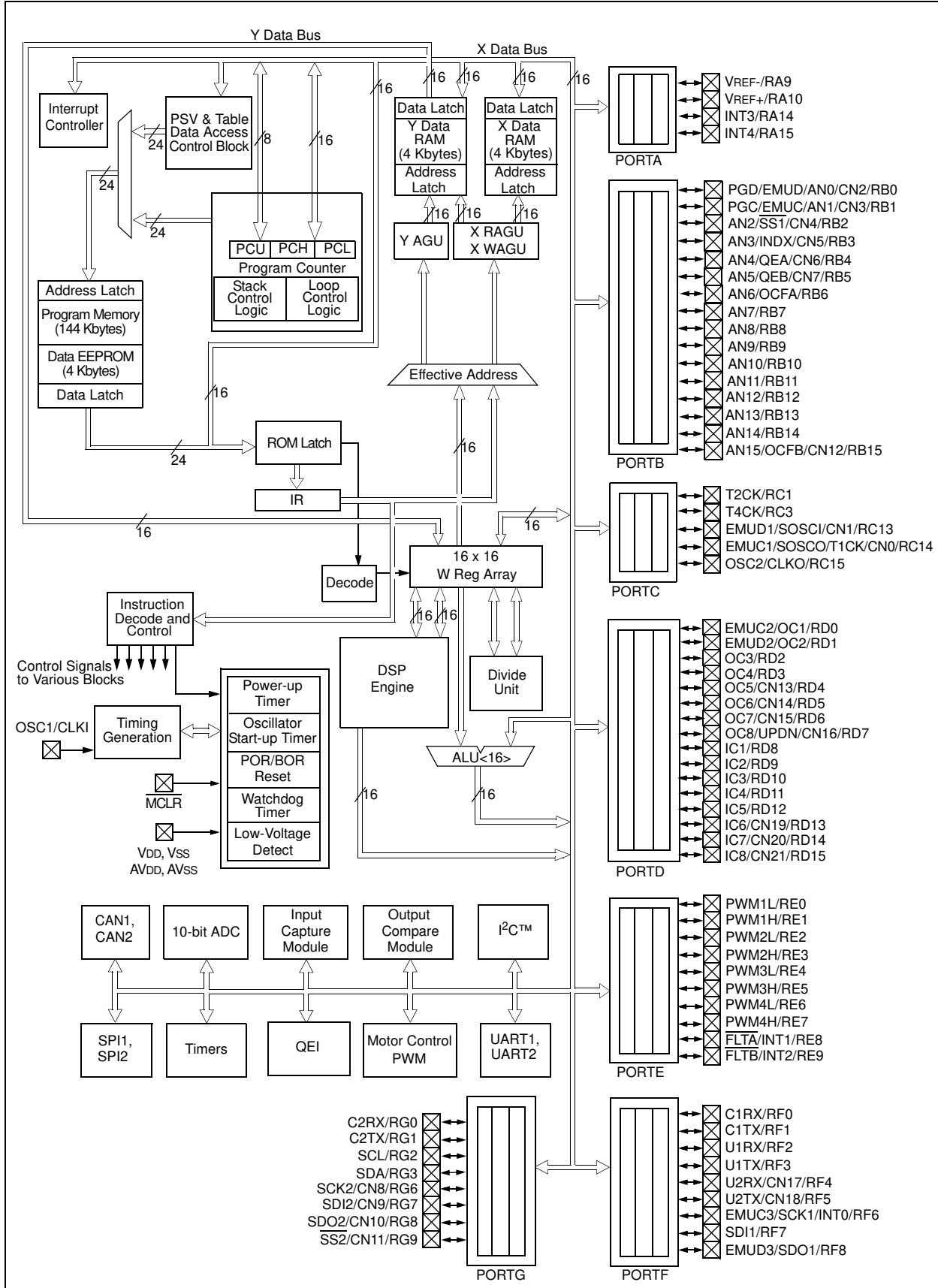
1.0 DEVICE OVERVIEW

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the “*dsPIC30F Family Reference Manual*” (DS70046). For more information on the device instruction set and programming, refer to the “*16-bit MCU and DSC Programmer’s Reference Manual*” (DS70157).

This document contains device-specific information for the dsPIC30F6010A and dsPIC30F6015 devices. The dsPIC30F devices contain extensive Digital Signal Processor (DSP) functionality within a high-performance 16-bit microcontroller (MCU) architecture. [Figure 1-1](#) shows a device block diagram for the dsPIC30F6010A device. [Figure 1-2](#) shows a device block diagram for the dsPIC30F6015 device.

dsPIC30F6010A/6015

FIGURE 1-1: dsPIC30F6010A BLOCK DIAGRAM



dsPIC30F6010A/6015

Table 1-1 provides a brief description of the device I/O pinout and the functions that are multiplexed to a port pin. Multiple functions may exist on one port pin. When multiplexing occurs, the peripheral module's functional requirements may force an override of the data direction of the port pin.

TABLE 1-1: dsPIC30F6010A/6015 I/O PIN DESCRIPTIONS

Pin Name	Pin Type	Buffer Type	Description
AN0-AN15	I	Analog	Analog input channels. AN0 and AN1 are also used for device programming data and clock inputs, respectively.
AVDD	P	P	Positive supply for analog module. This pin must be connected at all times.
AVSS	P	P	Ground reference for analog module. This pin must be connected at all times.
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	CAN1 bus receive pin.
C1TX	O	—	CAN1 bus transmit pin.
C2RX	I	ST	CAN2 bus receive pin.
C2TX	O	—	CAN2 bus transmit pin.
EMUD	I/O	ST	ICD Primary Communication Channel data input/output pin.
EMUC	I/O	ST	ICD Primary Communication Channel clock input/output pin.
EMUD1	I/O	ST	ICD Secondary Communication Channel data input/output pin.
EMUC1	I/O	ST	ICD Secondary Communication Channel clock input/output pin.
EMUD2	I/O	ST	ICD Tertiary Communication Channel data input/output pin.
EMUC2	I/O	ST	ICD Tertiary Communication Channel clock input/output pin.
EMUD3	I/O	ST	ICD Quaternary Communication Channel data input/output pin.
EMUC3	I/O	ST	ICD Quaternary Communication Channel clock input/output pin.
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.
QEB	I	ST	Quadrature Encoder Phase B input in QEI mode.
UPDN	O	—	Auxiliary Timer External Clock/Gate input in Timer mode. 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.

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

dsPIC30F6010A/6015

TABLE 1-1: dsPIC30F6010A/6015 I/O PIN DESCRIPTIONS (CONTINUED)

Pin Name	Pin Type	Buffer Type	Description
FLTA	I	ST	PWM Fault A input.
FLTB	I	ST	PWM Fault B input.
PWM1L	O	—	PWM1 Low output.
PWM1H	O	—	PWM1 High output.
PWM2L	O	—	PWM2 Low output.
PWM2H	O	—	PWM2 High output.
PWM3L	O	—	PWM3 Low output.
PWM3H	O	—	PWM3 High output.
PWM4L	O	—	PWM4 Low output.
PWM4H	O	—	PWM4 High output.
MCLR	I/P	ST	Master Clear (Reset) input or programming voltage 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.
PGD	I/O	ST	In-Circuit Serial Programming™ data input/output pin.
PGC	I	ST	In-Circuit Serial Programming clock input pin.
RA9-RA10	I/O	ST	PORTA is a bidirectional I/O port.
RA14-RA15	I/O	ST	
RB0-RB15	I/O	ST	PORTB is a bidirectional I/O port.
RC1	I/O	ST	PORTC is a bidirectional I/O port.
RC3	I/O	ST	
RC13-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.
RG0-RG3	I/O	ST	PORTG is a bidirectional I/O port.
RG6-RG9	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.
SS1	I	ST	SPI1 Slave Synchronization.
SCK2	I/O	ST	Synchronous serial clock input/output for SPI2.
SDI2	I	ST	SPI2 Data In.
SDO2	O	—	SPI2 Data Out.
SS2	I	ST	SPI2 Slave Synchronization.
SCL	I/O	ST	Synchronous serial clock input/output for I ² C™.
SDA	I/O	ST	Synchronous serial data input/output for I ² C.
SOSCO	O	—	32 kHz low-power oscillator crystal output.
SOSCI	I	ST/CMOS	32 kHz low-power oscillator crystal input. ST buffer when configured in RC mode; CMOS otherwise.
T1CK	I	ST	Timer1 external clock input.
T2CK	I	ST	Timer2 external clock input.
T4CK	I	ST	Timer4 external clock input.

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

dsPIC30F6010A/6015

TABLE 1-1: dsPIC30F6010A/6015 I/O PIN DESCRIPTIONS (CONTINUED)

Pin Name	Pin Type	Buffer Type	Description
U1RX	I	ST	UART1 Receive.
U1TX	O	—	UART1 Transmit.
U1ARX	I	ST	UART1 Alternate Receive.
U1ATX	O	—	UART1 Alternate Transmit.
U2RX	I	ST	UART2 Receive.
U2TX	O	—	UART2 Transmit.
VDD	P	—	Positive supply for logic and I/O pins.
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 Analog = Analog input
ST = Schmitt Trigger input with CMOS levels O = Output
I = Input P = Power

2.0 CPU ARCHITECTURE OVERVIEW

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the peripherals, register descriptions and general device functionality, refer to the “*dsPIC30F Family Reference Manual*” (DS70046). For more information on the device instruction set and programming, refer to the “*16-bit MCU and DSC Programmer’s Reference Manual*” (DS70157).

This chapter summarizes the CPU and peripheral functions of the dsPIC30F6010A/6015.

2.1 Core Overview

The core has a 24-bit instruction word. The Program Counter (PC) is 23 bits wide with the Least Significant bit (LSb) always clear (see [Section 3.1 “Program Address Space”](#)), and the Most Significant bit (MSb) is ignored during normal program execution, except for certain specialized instructions. Thus, the PC can address up to 4M instruction words of user program space. An instruction prefetch mechanism is used to help maintain throughput. Program loop constructs, free from loop count management overhead, are supported using the `DO` and `REPEAT` instructions, both of which are interruptible at any point.

The working register array consists of 16x16-bit registers, each of which can act as data, address or offset registers. One working register (W15) operates as a Software Stack Pointer for interrupts and calls.

The data space is 64 Kbytes (32K words) and is split into two blocks, referred to as X and Y data memory. Each block has its own independent Address Generation Unit (AGU). Most instructions operate solely through the X memory AGU, which provides the appearance of a single unified data space. The Multiply-Accumulate (MAC) class of dual source DSP instructions operate through both the X and Y AGUs, splitting the data address space into two parts (see [Section 3.2 “Data Address Space”](#)). The X and Y data space boundary is device-specific and cannot be altered by the user. Each data word consists of 2 bytes, and most instructions can address data either as words or bytes.

There are two methods of accessing data stored in program memory:

- The upper 32 Kbytes of data space memory can be mapped into the lower half (user space) of program space at any 16K program word boundary, defined by the 8-bit Program Space Visibility Page (PSVPAG) register. This lets any instruction access program space as if it were data space, with a limitation that the access requires an additional cycle. Moreover, only the lower 16 bits of each instruction word can be accessed using this method.

- Linear indirect access of 32K word pages within program space is also possible using any working register, via table read and write instructions. Table read and write instructions can be used to access all 24 bits of an instruction word.

Overhead-free circular buffers (Modulo Addressing) are supported in both X and Y address spaces. This is primarily intended to remove the loop overhead for DSP algorithms.

The X AGU also supports Bit-Reversed Addressing on destination Effective Addresses, to greatly simplify input or output data reordering for radix-2 FFT algorithms. Refer to [Section 4.0 “Address Generator Units”](#) for details on Modulo and Bit-Reversed Addressing.

The core supports Inherent (no operand), Relative, Literal, Memory Direct, Register Direct, Register Indirect, Register Offset and Literal Offset Addressing modes. Instructions are associated with predefined addressing modes, depending upon their functional requirements.

For most instructions, the core is capable of executing a data (or program data) memory read, a working register (data) read, a data memory write and a program (instruction) memory read per instruction cycle. As a result, 3-operand instructions are supported, allowing $C = A + B$ operations to be executed in a single cycle.

A DSP engine has been included to significantly enhance the core arithmetic capability and throughput. It 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. Data in the accumulator or any working register can be shifted up to 16 bits right or 16 bits 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 class of instructions can concurrently fetch two data operands from memory, while multiplying two W registers. To enable this concurrent fetching of data operands, the data space has been split for these instructions and linear for all others. This has been achieved in a transparent and flexible manner, by dedicating certain working registers to each address space for the MAC class of instructions.

The core does not support a multi-stage instruction pipeline. However, a single stage instruction prefetch mechanism is used, which accesses and partially decodes instructions a cycle ahead of execution, in order to maximize available execution time. Most instructions execute in a single cycle, with certain exceptions.

The core features a vectored exception processing structure for traps and interrupts, with 62 independent vectors. The exceptions consist of up to 8 traps (of which 4 are reserved) and 54 interrupts. Each interrupt is prioritized based on a user-assigned priority between 1 and 7 (1 being the lowest priority and 7 being the highest) in conjunction with a predetermined ‘natural order’. Traps have fixed priorities, ranging from 8 to 15.

dsPIC30F6010A/6015

2.2 Programmer's Model

The programmer's model is shown in Figure 2-1 and consists of 16x16-bit working registers (W0 through W15), 2x40-bit accumulators (AccA and AccB), STATUS register (SR), Data Table Page register (TBLPAG), Program Space Visibility Page register (PSVPAG), DO and REPEAT registers (DOSTART, DOEND, DCOUNT and RCOUNT), and Program Counter (PC). The working registers can act as data, address or offset registers. All registers are memory mapped. W0 acts as the W register for file register addressing.

Some of these registers have a shadow register associated with each of them, as shown in Figure 2-1. The shadow register is used as a temporary holding register and can transfer its contents to or from its host register upon the occurrence of an event. None of the shadow registers are accessible directly. The following rules apply for transfer of registers into and out of shadows.

- PUSH.S and POP.S
W0, W1, W2, W3, SR (DC, N, OV, Z and C bits only) are transferred.
- DO instruction
DOSTART, DOEND, DCOUNT shadows are pushed on loop start and popped on loop end.

When a byte operation is performed on a working register, only the Least Significant Byte of the target register is affected. However, a benefit of memory mapped working registers is that both the Least and Most Significant Bytes can be manipulated through byte-wide data memory space accesses.

2.2.1 SOFTWARE STACK POINTER/ FRAME POINTER

The dsPIC DSC devices contain a software stack. W15 is the dedicated Software Stack Pointer (SP), and will be automatically modified by exception processing and subroutine calls and returns. However, W15 can be referenced by any instruction in the same manner as all other W registers. This simplifies the reading, writing and manipulation of the Stack Pointer (e.g., creating stack frames).

Note: In order to protect against misaligned stack accesses, W15<0> is always clear.

W15 is initialized to 0x0800 during a Reset. The user may reprogram the SP during initialization to any location within data space.

W14 has been dedicated as a Stack Frame Pointer as defined by the LNK and ULNK instructions. However, W14 can be referenced by any instruction in the same manner as all other W registers.

2.2.2 STATUS REGISTER

The dsPIC DSC core has a 16-bit STATUS register (SR), the LSB of which is referred to as the SR Low Byte (SRL) and the MSB as the SR High Byte (SRH). See Figure 2-1 for SR layout.

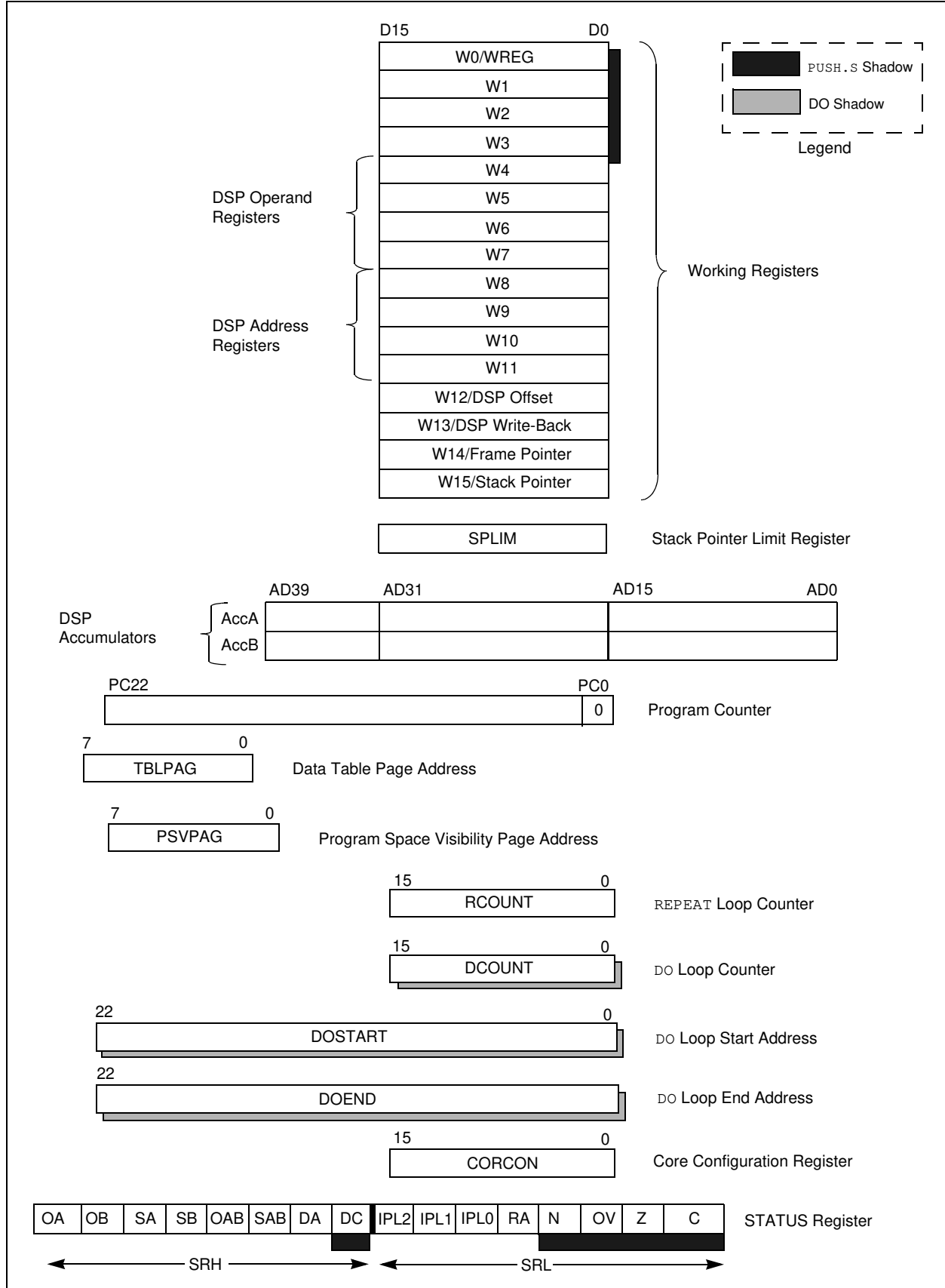
SRL contains all the MCU ALU operation status flags (including the Z bit), as well as the CPU Interrupt Priority Level Status bits, IPL<2:0>, and the Repeat Active Status bit, RA. During exception processing, SRL is concatenated with the MSB of the PC to form a complete word value which is then stacked.

The upper byte of the SR register contains the DSP adder/subtractor Status bits, the DO Loop Active bit (DA) and the Digit Carry (DC) Status bit.

2.2.3 PROGRAM COUNTER

The Program Counter is 23 bits wide. Bit 0 is always clear. Therefore, the PC can address up to 4M instruction words.

FIGURE 2-1: dsPIC30F6010A/6015 PROGRAMMER'S MODEL



dsPIC30F6010A/6015

2.3 Divide Support

The dsPIC DSC devices feature a 16/16-bit signed fractional divide operation, as well as 32/16-bit and 16/16-bit signed and unsigned integer divide operations, in the form of single instruction iterative divides. The following instructions and data sizes are supported:

- `DIVF` – 16/16 signed fractional divide
- `DIV.sd` – 32/16 signed divide
- `DIV.ud` – 32/16 unsigned divide
- `DIV.s` – 16/16 signed divide
- `DIV.u` – 16/16 unsigned divide

The divide instructions must be executed within a REPEAT loop. Any other form of execution (e.g., a series of discrete divide instructions) will not function correctly because the instruction flow depends on RCOUNT. The divide instruction does not automatically set up the RCOUNT value, and it must, therefore, be explicitly and correctly specified in the REPEAT instruction, as shown in [Table 2-2](#) (REPEAT will execute the target instruction {operand value + 1} times). The REPEAT loop count must be set up for 18 iterations of the `DIV/DIVF` instruction. Thus, a complete divide operation requires 19 cycles.

Note: The divide flow is interruptible. However, the user needs to save the context as appropriate.

2.4 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 dsPIC30F devices have a single instruction flow which can execute either DSP or MCU instructions. Many of the hardware resources are shared between the DSP and MCU instructions. For example, the instruction set has both DSP and MCU multiply instructions which use the same hardware multiplier.

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 Configuration 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).

Note: For CORCON layout, see [Table 3-3](#).

A block diagram of the DSP engine is shown in [Figure 2-2](#).

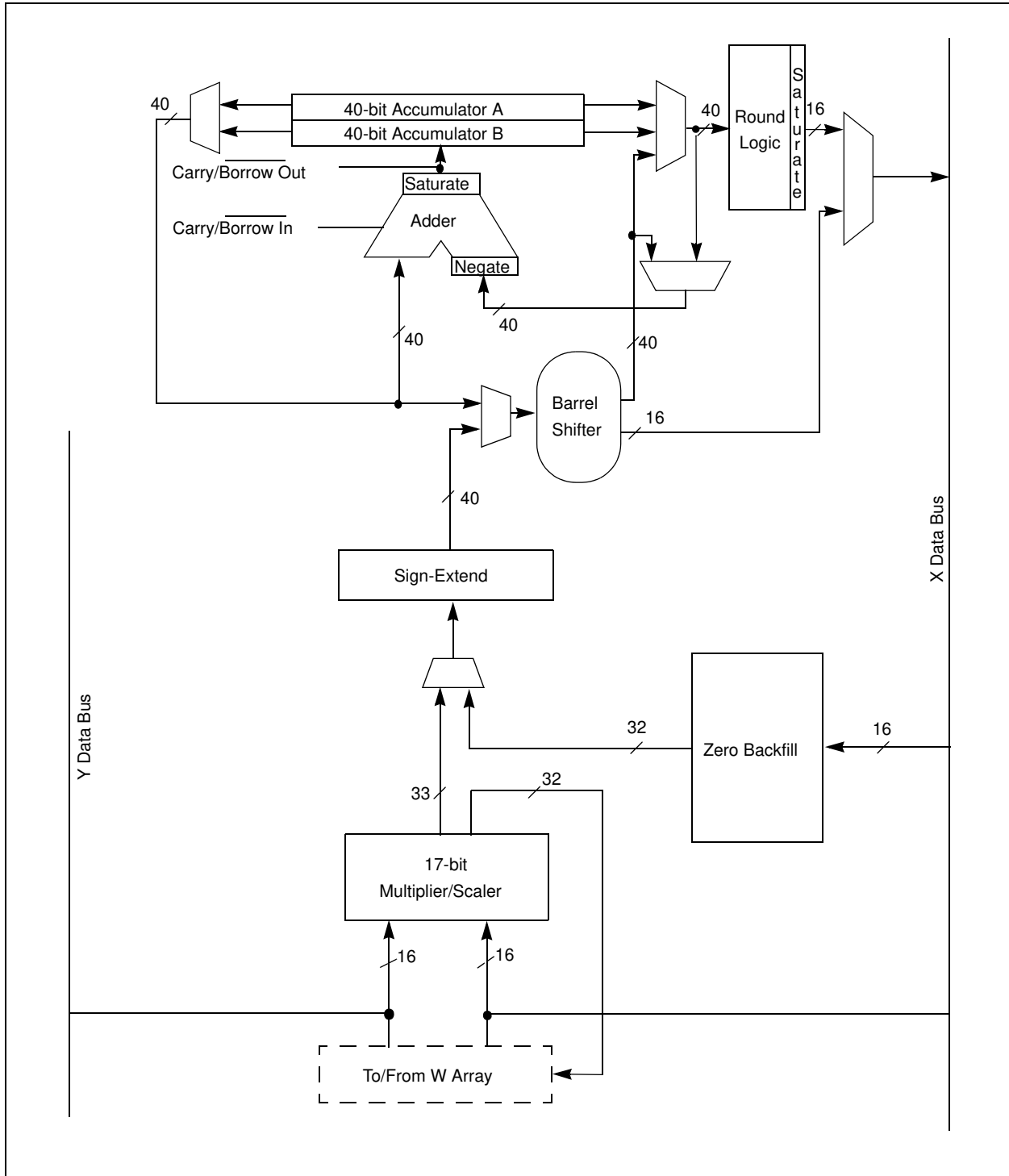
TABLE 2-1: DSP INSTRUCTION SUMMARY

Instruction	Algebraic Operation
CLR	$A = 0$
ED	$A = (x - y)^2$
EDAC	$A = A + (x - y)^2$
MAC	$A = A + (x * y)$
MOVSAC	No change in A
MPY	$A = x * y$
MPY.N	$A = -x * y$
MSC	$A = A - x * y$

TABLE 2-2: DIVIDE INSTRUCTIONS

Instruction	Function
<code>DIVF</code>	Signed fractional divide: $Wm/Wn \rightarrow W0$; Rem $\rightarrow W1$
<code>DIV.sd</code>	Signed divide: $(Wm+1:Wm)/Wn \rightarrow W0$; Rem $\rightarrow W1$
<code>DIV.s</code>	Signed divide: $Wm/Wn \rightarrow W0$; Rem $\rightarrow W1$
<code>DIV.ud</code>	Unsigned divide: $(Wm+1:Wm)/Wn \rightarrow W0$; Rem $\rightarrow W1$
<code>DIV.u</code>	Unsigned divide: $Wm/Wn \rightarrow W0$; Rem $\rightarrow W1$

FIGURE 2-2: DSP ENGINE BLOCK DIAGRAM



dsPIC30F6010A/6015

2.4.1 MULTIPLIER

The 17x17-bit multiplier is capable of signed or unsigned operations 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 17x17-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 2,147,483,645 (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 $(1-2^{-N})$. 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, a 16x16 multiply operation generates a 1.31 product, which 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 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.

2.4.2 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 via the barrel shifter, prior to accumulation.

2.4.2.1 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), whereas 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 which controls accumulator data saturation, if selected. It uses the result of the adder, the Overflow Status bits described above, and the SATA/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/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 bit (OVATE, OVBTE) in the INTCON1 register (refer to [Section 5.0 "Interrupts"](#)) is 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/subtractor, 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 (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. 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 (0x00800000) 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 remain 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.

2.4.2.2 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).

2.4.2.3 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 takes bit 15 of the accumulator, zero-extends it 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 will tend to be biased slightly positive.

Convergent (or unbiased) rounding operates in the same manner as conventional rounding, except when ACCxL equals 0x8000. If this is the case, the LSb (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 will remove 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 2.4.2.4 "Data Space Write Saturation"](#)). Note that 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.

2.4.2.4 Data Space Write Saturation

In addition to adder/subtractor saturation, writes to data space may 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 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 MSb 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.

2.4.3 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 will shift the operand right. A negative value will shift the operand left. A value of '0' will 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 bit positions 0 to 15 for left shifts.

3.0 MEMORY ORGANIZATION

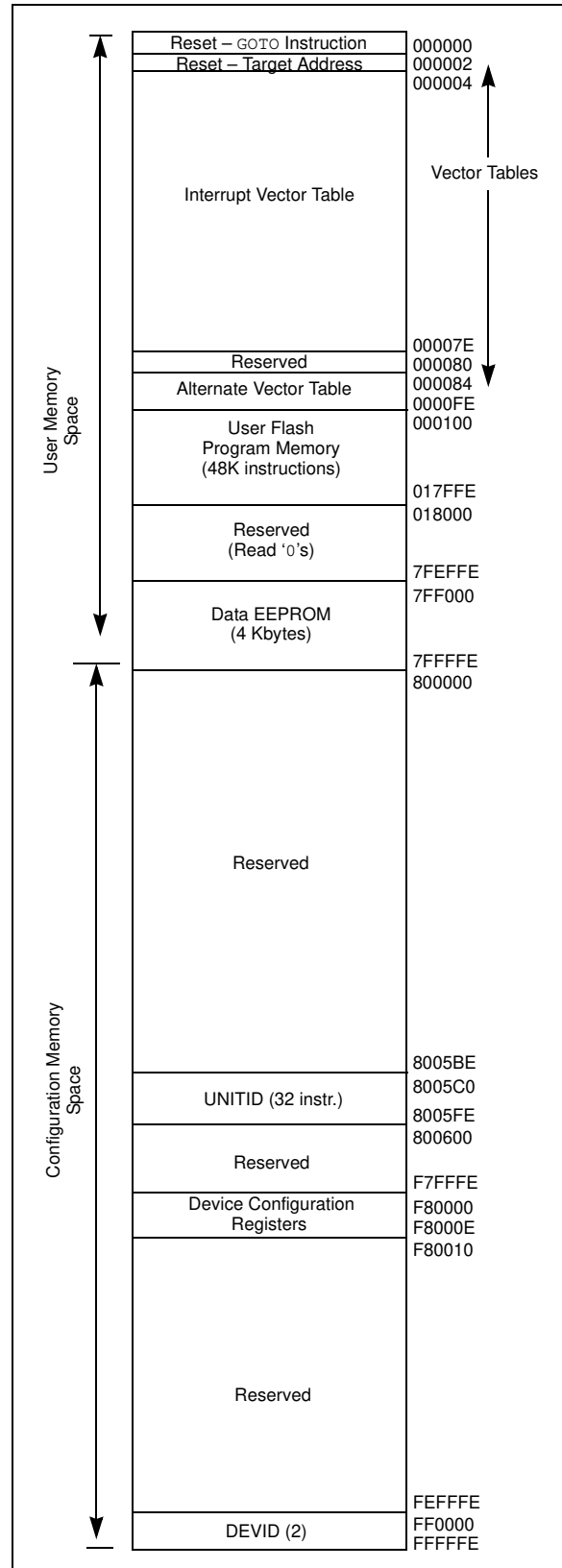
Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the “dsPIC30F Family Reference Manual” (DS70046). For more information on the device instruction set and programming, refer to the “16-bit MCU and DSC Programmer’s Reference Manual” (DS70157).

3.1 Program Address Space

The program address space is 4M instruction words. It is addressable by the 23-bit PC, table instruction Effective Address (EA), or data space EA, when program space is mapped into data space, as defined by Table 3-1. Note that the program space address is incremented by two between successive program words, in order to provide compatibility with data space addressing.

User program space access is restricted to the lower 4M instruction word address range (0x000000 to 0x7FFFFFFE), for all accesses other than TBLRD/TBLWT, which use TBLPAG<7> to determine user or configuration space access. In Table 3-1, read/write instructions, bit 23 allows access to the device ID, the user ID and the Configuration bits. Otherwise, bit 23 is always clear.

FIGURE 3-1: PROGRAM SPACE MEMORY MAP FOR dsPIC30F6010A/6015

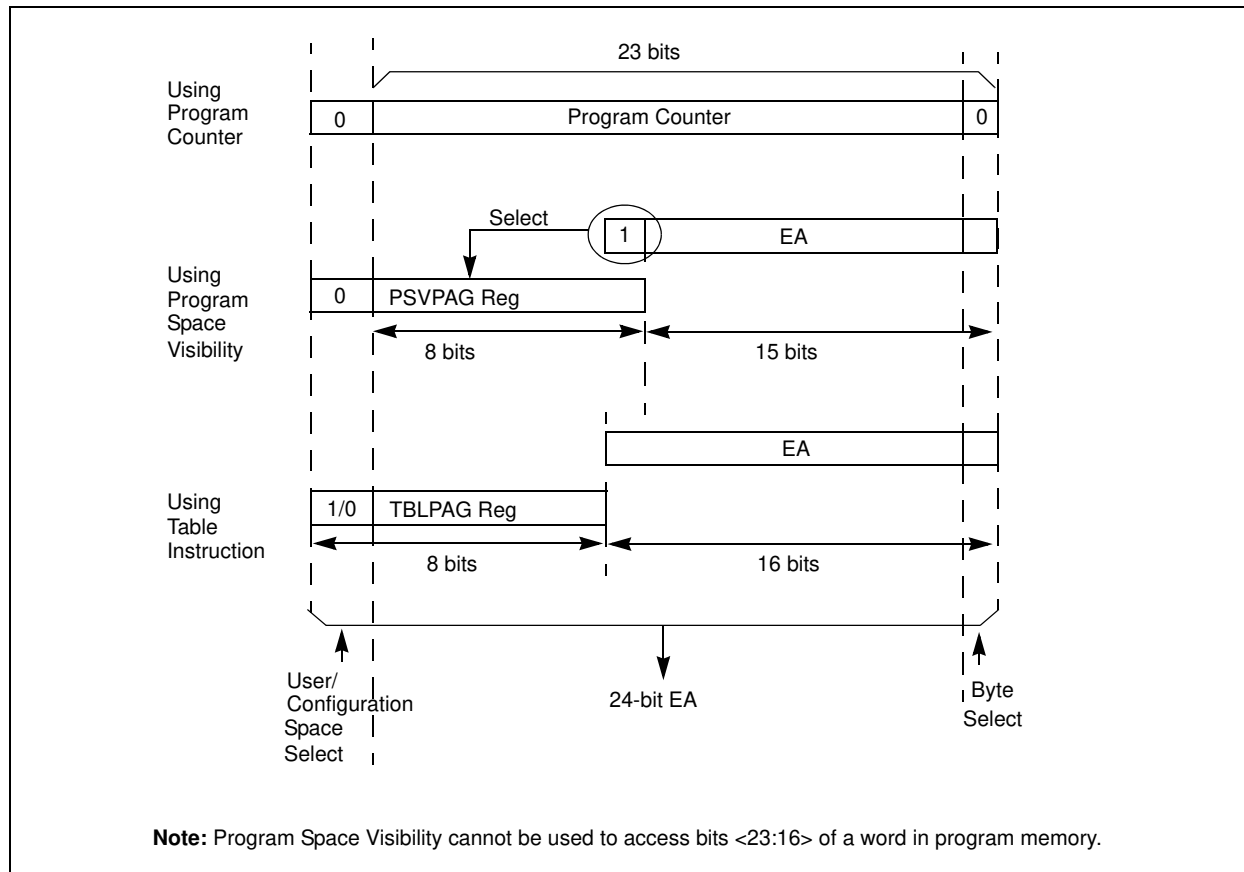


dsPIC30F6010A/6015

TABLE 3-1: PROGRAM SPACE ADDRESS CONSTRUCTION

Access Type	Access Space	Program Space Address				
		<23>	<22:16>	<15>	<14:1>	<0>
Instruction Access	User	0	PC<22:1>			0
TBLRD/TBLWT	User (TBLPAG<7> = 0)	TBLPAG<7:0>		Data EA <15:0>		
TBLRD/TBLWT	Configuration (TBLPAG<7> = 1)	TBLPAG<7:0>		Data EA <15:0>		
Program Space Visibility	User	0	PSVPAG<7:0>	Data EA <14:0>		

FIGURE 3-2: DATA ACCESS FROM PROGRAM SPACE ADDRESS GENERATION



3.1.1 DATA ACCESS FROM PROGRAM MEMORY USING TABLE INSTRUCTIONS

This architecture fetches 24-bit wide program memory. Consequently, instructions are always aligned. However, as the architecture is modified Harvard, data can also be present in program space.

There are two methods by which program space can be accessed; via special table instructions, or through the remapping of a 16K word program space page into the upper half of data space (see [Section 3.1.2 “Data Access from Program Memory Using Program Space Visibility”](#)). The **TBLRDL** and **TBLWTL** instructions offer a direct method of reading or writing the least significant word of any address within program space, without going through data space. The **TBLRDH** and **TBLWTH** instructions are the only method whereby the upper 8 bits of a program space word can be accessed 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 lsw, and **TBLRDH** and **TBLWTH** access the space which contains the MSB.

[Figure 3-2](#) shows how the EA is created for table operations and data space accesses (PSV = 1). Here, P<23:0> refers to a program space word, whereas D<15:0> refers to a data space word.

A set of table instructions are provided to move byte or word-sized data to and from program space.

1. **TBLRDL**: Table Read Low
Word: Read the least significant word of the program address;
P<15:0> maps to D<15:0>.
Byte: Read one of the Least Significant Bytes of the program address;
P<7:0> maps to the destination byte when byte select = 0;
P<15:8> maps to the destination byte when byte select = 1.
2. **TBLWTL**: Table Write Low (refer to [Section 6.0 “Flash Program Memory”](#) for details on Flash Programming).
3. **TBLRDH**: Table Read High
Word: Read the most significant word of the program address;
P<23:16> maps to D<7:0>; D<15:8> always be = 0.
Byte: Read one of the Most Significant Bytes of the program address;
P<23:16> maps to the destination byte when byte select = 0;
The destination byte will always be = 0 when byte select = 1.
4. **TBLWTH**: Table Write High (refer to [Section 6.0 “Flash Program Memory”](#) for details on Flash Programming).

FIGURE 3-3: PROGRAM DATA TABLE ACCESS (LEAST SIGNIFICANT WORD)

