

# dsPIC30F6010 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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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.

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

#### Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Migratable Memory, 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, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Linear Active Thermistor, Mindi, MiWi, MPASM, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

 $\ensuremath{\mathsf{SQTP}}$  is a service mark of Microchip Technology Incorporated in the U.S.A.

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

© 2006, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona, Gresham, Oregon and Mountain View, California. The Company's quality system processes and procedures are for its PIC<sup>®</sup> 8-bit MCUs, KEELOQ<sup>®</sup> code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.



### dsPIC30F6010 Enhanced Flash 16-Bit Digital Signal Controller

**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 "*dsPIC30F/ 33F 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)
- 44 interrupt sources:
  - 5 external interrupt sources
  - 8 user-selectable priority levels for each interrupt source
  - 4 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)
- I<sup>2</sup>C<sup>™</sup> module supports Multi-Master/Slave mode and 7-bit/10-bit addressing
- 2 UART modules with FIFO Buffers
- 2 CAN modules, 2.0B compliant

#### Motor Control PWM Module Features:

- 8 PWM output channels
  - Complementary or Independent Output modes
  - Edge and Center-Aligned modes
- 4 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 4 S/H Inputs:
  - 1 Msps conversion rate
  - 16 input channels
- Conversion available during Sleep and Idle
- Programmable Low-Voltage Detection (PLVD)
- Programmable Brown-out Reset

#### **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<sup>™</sup> (ICSP<sup>™</sup>)
- 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	ADC 10-bit 1 Msps	Quad Enc	UART	SPI	I <sup>2</sup> C™	CAN
dsPIC30F2010	28	12K/4K	512	1024	3	4	2	6 ch	6 ch	Yes	1	1	1	-
dsPIC30F3010	28	24K/8K	1024	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	-
dsPIC30F4012	28	48K/16K	2048	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	1
dsPIC30F3011	40/44	24K/8K	1024	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	-
dsPIC30F4011	40/44	48K/16K	2048	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	1
dsPIC30F5015	64	66K/22K	2048	1024	5	4	4	8 ch	16 ch	Yes	1	2	1	1
dsPIC30F6010	80	144K/48K	8192	4096	5	8	8	8 ch	16 ch	Yes	2	2	1	2

\* This table provides a summary of the dsPIC30F6010 peripheral features. Other available devices in the dsPIC30F Motor Control and Power Conversion Family are shown for feature comparison.

#### Pin Diagram



#### **Table of Contents**

2.0   CPU Architecture Overview.   11     3.0   Memory Organization   19     3.0   Memory Organization   31     5.0   Interrupts   37     6.0   Flash Program Memory   43     7.0   Data EEPROM Memory   49     8.0   //O Ports   53     9.0   Timer1 Module   57     10.0   Timer4/5 Module   61     11.0   Timer4/5 Module   67     12.0   Input Capture Module   71     13.0   Output Compare Module   75     14.0   Quadrature Encoder Interface (QEI) Module   75     15.0   SPI Module   85     16.0   SPI Module   95     17.0   I2C Module   99     18.0   Universal Asynchronous Receiver Transmitter (UART) Module   99     18.0   Universal Asynchronous Receiver Transmitter (UART) Module   115     20.1   Obeit High-Speed Analog-to-Digital Converter (ADC) Module   127     21.0   System Integration   139     22.0   Development Support   153     23	1.0	Device Overview	5
3.0   Memory Organization   19     4.0   Address Generator Units   31     5.0   Interrupts   37     6.0   Flash Program Memory   43     7.0   Flash Program Memory   43     7.0   Data EEPROM Memory   43     8.0   I/O Ports   53     9.0   Timer1 Module   57     11.0   Timer1/S Module   61     11.10   Timer4/5 Module   61     11.0   Output Compare Module   71     13.0   Output Compare Module   75     14.0   Quadrature Encoder Interface (QEI) Module   85     15.0   Motor Control PWM Module   85     15.0   SPI Module   99     13.0   Universal Asynchronous Receiver Transmitter (UART) Module   107     14.0   CAN Module   115     20.0   Iovit High-Speed Analog-to-Digital Converter (ADC) Module   127     21.0	2.0	CPU Architecture Overview	11
4.0   Address Generator Units.   31     5.0   Interrupts   37     6.0   Flash Program Memory.   43     7.0   Data EEPROM Memory   49     8.0   I/O Ports   53     9.0   Timer1 Module   57     10.0   Timer2/3 Module   61     11.0   Timer4/5 Module   67     12.0   Input Capture Module   71     13.0   Output Compare Module   71     13.0   Output Compare Module   75     14.0   Quadrature Encoder Interface (QEI) Module   79     15.0   Motor Control PWM Module   85     16.0   SPI Module   95     17.0   Ize Kodule   95     17.0   Ize Module   95     18.0   Universal Asynchronous Receiver Transmitter (UART) Module   107     19.0   CAN Module   115     20.0   Development Support   132     21.0   System Integration   133     22.0   Development Support   157     24.0   Electrical Characteristics   167 <td>3.0</td> <td>Memory Organization</td> <td> 19</td>	3.0	Memory Organization	19
5.0   Interrupts   37     6.0   Flash Program Memory   43     7.0   Data EEPROM Memory   49     8.0   I/O Ports   53     9.0   Timer1 Module   57     11.0   Timer2/3 Module   61     11.0   Timer4/5 Module   61     11.0   Timer4/5 Module   61     12.0   Input Capture Module   71     13.0   Output Compare Module   75     14.0   Quadrature Encoder Interface (QEI) Module   79     15.0   Motor Control PWM Module   85     16.0   SPI Module   95     17.0   IzC Module   99     18.0   Universal Asynchronous Receiver Transmitter (UART) Module   107     19.0   CAN Module   115     20.0   Development Support.   139     21.0   System Integration   139     22.0   Development Support.   153     23.0   Instruction Set Summary   167     24.0   Electrical Characteristics   167     25.0   Packaging Information   207	4.0	Address Generator Units	31
6.0   Flash Program Memory.   43     7.0   Data EEPROM Memory   49     8.0   I/O Ports   53     9.0   Timer1 Module   57     10.0   Timer2/3 Module   61     11.0   Timer4/5 Module   67     12.0   Input Capture Module   67     13.0   Output Compare Module   75     14.0   Quadrature Encoder Interface (QEI) Module   79     15.0   Motor Control PWM Module   85     16.0   SPI Module   95     17.0   Izc Module   95     17.0   Iversal Asynchronous Receiver Transmitter (UART) Module   107     19.0   CAN Module   107     10.0   Tobit High-Speed Analog-to-Digital Converter (ADC) Module   127     21.0   Development Support.   153     23.0   Instruction Set Summary   157     24.0   Electrical Characteristics   167     25.0   Packaging Information.   207     74.0   Electrical Characteristics   167     25.0   Packaging Information.   207 <td< td=""><td>5.0</td><td>Interrupts</td><td> 37</td></td<>	5.0	Interrupts	37
7.0   Data EEPROM Memory   49     8.0   I/O Ports   53     9.0   Timer1 Module   57     10.0   Timer2/3 Module   67     11.0   Timer4/5 Module   67     12.0   Input Capture Module   71     13.0   Output Compare Module   75     14.0   Quadrature Encoder Interface (QEI) Module   75     15.0   Motor Control PWM Module   85     16.0   SPI Module   95     17.0   I2C Module   99     18.0   Universal Asynchronous Receiver Transmitter (UART) Module   107     19.0   CAN Module   115     20.0   10-bit High-Speed Analog-to-Digital Converter (ADC) Module   127     21.0   System Integration   139     22.0   Development Support.   153     23.0   Instruction Set Summary   157     24.0   Electrical Characteristics   167     24.0   Electrical Characteristics   167     24.0   Electrical Characteristics   167     24.0   Electrical Characteristics   167	6.0	Flash Program Memory	43
8.0   I/O Ports   53     9.0   Timer1 Module   57     10.0   Timer2/3 Module   61     11.0   Timer4/5 Module   67     12.0   Input Capture Module   75     13.0   Output Compare Module   75     14.0   Quadrature Encoder Interface (QEI) Module   79     15.0   Motor Control PWM Module   85     16.0   SPI Module   95     17.0   I2C Module   97     18.0   Universal Asynchronous Receiver Transmitter (UART) Module   107     19.0   CAN Module   107     19.0   CAN Module   115     20.0   Ivelopment Support.   153     23.0   Instruction Set Summary   157     24.0   Electrical Characteristics   167     25.0   Packaging Information   207     7	7.0	Data EEPROM Memory	49
9.0   Timer1 Module   57     10.0   Timer2/3 Module   61     11.0   Timer4/5 Module   67     12.0   Input Capture Module   67     13.0   Output Compare Module   75     14.0   Quadrature Encoder Interface (QEI) Module   79     15.0   Motor Control PWM Module   85     16.0   SPI Module   95     17.0   I2C Module   95     18.0   Universal Asynchronous Receiver Transmitter (UART) Module   107     19.0   CAN Module   107     19.0   CAN Module   127     20.1   Development Support.   139     21.0   System Integration   139     22.0   Development Support.   157     23.0   Instruction Set Summary.   157     24.0   Electrical Characteristics   167     25.0   Packaging Information.   207	8.0	I/O Ports	53
10.0   Timer2/3 Module   61     11.0   Timer4/5 Module   67     12.0   Input Capture Module   71     13.0   Output Compare Module   75     14.0   Quadrature Encoder Interface (QEI) Module   79     15.0   Motor Control PWM Module   85     16.0   SPI Module   95     17.0   I2C Module   99     18.0   Universal Asynchronous Receiver Transmitter (UART) Module   107     19.0   CAN Module   115     20.0   10-bit High-Speed Analog-to-Digital Converter (ADC) Module   127     21.0   System Integration   139     22.0   Development Support.   153     23.0   Instruction Set Summary   157     24.0   Electrical Characteristics   167     25.0   Packaging Information   207     The Microchip Web Site   217   217     Customer Change Notification Service   217     Reader Response   218     Product Identification System   219	9.0	Timer1 Module	57
11.0   Timer4/5 Module   67     12.0   Input Capture Module   71     13.0   Output Compare Module   75     14.0   Quadrature Encoder Interface (QEI) Module   79     15.0   Motor Control PWM Module   85     16.0   SPI Module   95     17.0   I2C Module   99     18.0   Universal Asynchronous Receiver Transmitter (UART) Module   107     19.0   CAN Module   115     20.0   I0-bit High-Speed Analog-to-Digital Converter (ADC) Module   127     21.0   System Integration   133     22.0   Development Support   153     23.0   Instruction Set Summary   157     24.0   Electrical Characteristics   167     25.0   Packaging Information   207     The Microchip Web Site   217   217     Customer Change Notification Service   217     Reader Response   218     Product Identification System   219	10.0	Timer2/3 Module	61
12.0Input Capture Module7113.0Output Compare Module7514.0Quadrature Encoder Interface (QEI) Module7915.0Motor Control PWM Module8516.0SPI Module9517.0I2C Module9918.0Universal Asynchronous Receiver Transmitter (UART) Module10719.0CAN Module10720.010-bit High-Speed Analog-to-Digital Converter (ADC) Module12721.0System Integration13922.0Development Support.15323.0Instruction Set Summary15724.0Electrical Characteristics16725.0Packaging Information207The Microchip Web Site217Customer Change Notification Service217Customer Support218Product Identification System218Product Identification System219	11.0	Timer4/5 Module	67
13.0Output Compare Module7514.0Quadrature Encoder Interface (QEI) Module7915.0Motor Control PWM Module8516.0SPI Module9517.0I2C Module9918.0Universal Asynchronous Receiver Transmitter (UART) Module10719.0CAN Module11520.010-bit High-Speed Analog-to-Digital Converter (ADC) Module12721.0System Integration13922.0Development Support.15323.0Instruction Set Summary15724.0Electrical Characteristics16725.0Packaging Information207The Microchip Web Site217Customer Change Notification Service217Reader Response218Product Identification System219	12.0	Input Capture Module	71
14.0Quadrature Encoder Interface (QEI) Module7915.0Motor Control PWM Module8516.0SPI Module9517.0I2C Module9918.0Universal Asynchronous Receiver Transmitter (UART) Module10719.0CAN Module10719.0CAN Module11520.010-bit High-Speed Analog-to-Digital Converter (ADC) Module12721.0System Integration13922.0Development Support15323.0Instruction Set Summary15724.0Electrical Characteristics16725.0Packaging Information207The Microchip Web Site217Customer Change Notification Service217Customer Support217Reader Response218Product Identification System219	13.0	Output Compare Module	75
15.0   Motor Control PWM Module   85     16.0   SPI Module   95     17.0   I2C Module   99     18.0   Universal Asynchronous Receiver Transmitter (UART) Module   107     19.0   CAN Module   107     19.0   CAN Module   115     20.0   10-bit High-Speed Analog-to-Digital Converter (ADC) Module   127     21.0   System Integration   139     22.0   Development Support   153     23.0   Instruction Set Summary   157     24.0   Electrical Characteristics   167     25.0   Packaging Information   207     The Microchip Web Site   217   217     Customer Change Notification Service   217     Customer Support   217   218     Product Identification System   218	14.0	Quadrature Encoder Interface (QEI) Module	79
16.0   SPI Module	15.0	Motor Control PWM Module	85
17.0I2C Module9918.0Universal Asynchronous Receiver Transmitter (UART) Module10719.0CAN Module11520.010-bit High-Speed Analog-to-Digital Converter (ADC) Module12721.0System Integration13922.0Development Support15323.0Instruction Set Summary15724.0Electrical Characteristics16725.0Packaging Information207The Microchip Web Site217Customer Change Notification Service217Customer Support217Reader Response218Product Identification System219	16.0	SPI Module	95
18.0Universal Asynchronous Receiver Transmitter (UART) Module10719.0CAN Module11520.010-bit High-Speed Analog-to-Digital Converter (ADC) Module12721.0System Integration13922.0Development Support15323.0Instruction Set Summary15724.0Electrical Characteristics16725.0Packaging Information207The Microchip Web Site217Customer Change Notification Service217Customer Support218Product Identification System219	17.0	I2C Module	99
19.0CAN Module11520.010-bit High-Speed Analog-to-Digital Converter (ADC) Module12721.0System Integration13922.0Development Support15323.0Instruction Set Summary15724.0Electrical Characteristics16725.0Packaging Information207The Microchip Web Site217Customer Change Notification Service217Customer Support217Reader Response218Product Identification System219	18.0	Universal Asynchronous Receiver Transmitter (UART) Module	107
20.010-bit High-Speed Analog-to-Digital Converter (ADC) Module12721.0System Integration13922.0Development Support15323.0Instruction Set Summary15724.0Electrical Characteristics16725.0Packaging Information207The Microchip Web Site217Customer Change Notification Service217Customer Support217Reader Response218Product Identification System219	19.0	CAN Module	115
21.0System Integration13922.0Development Support15323.0Instruction Set Summary15724.0Electrical Characteristics16725.0Packaging Information207The Microchip Web Site217Customer Change Notification Service217Customer Support217Reader Response218Product Identification System219	20.0	10-bit High-Speed Analog-to-Digital Converter (ADC) Module	127
22.0 Development Support.15323.0 Instruction Set Summary15724.0 Electrical Characteristics16725.0 Packaging Information207The Microchip Web Site217Customer Change Notification Service217Customer Support217Reader Response218Product Identification System219	21.0	System Integration	139
23.0   Instruction Set Summary   157     24.0   Electrical Characteristics   167     25.0   Packaging Information   207     The Microchip Web Site   217     Customer Change Notification Service   217     Customer Support   217     Reader Response   218     Product Identification System   219	22.0	Development Support	153
24.0   Electrical Characteristics   167     25.0   Packaging Information   207     The Microchip Web Site   217     Customer Change Notification Service   217     Customer Support   217     Reader Response   218     Product Identification System   219	23.0	Instruction Set Summary	157
25.0   Packaging Information   207     The Microchip Web Site   217     Customer Change Notification Service   217     Customer Support   217     Reader Response   218     Product Identification System   219	24.0	Electrical Characteristics	167
The Microchip Web Site   217     Customer Change Notification Service   217     Customer Support   217     Reader Response   218     Product Identification System   219	25.0	Packaging Information	207
Customer Change Notification Service   217     Customer Support   217     Reader Response   218     Product Identification System   219	The N	Aicrochip Web Site	217
Customer Support   217     Reader Response   218     Product Identification System   219	Custo	omer Change Notification Service	217
Reader Response   218     Product Identification System   219	Custo	omer Support	217
Product Identification System	Read	er Response	218
	Produ	uct Identification System	219

### 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@mail.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)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, 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/cn to receive the most current information on all of our products.

#### 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 "*dsPIC30F/ 33F Programmer's Reference Manual*" (DS70157). This document contains device specific information for the dsPIC30F6010 device. 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 dsPIC30F6010 device.





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.

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	Р	Р	ositive supply for analog module.				
AVss	Р	Р	Ground reference for analog module.				
CLKI	I	ST/CMOS	External clock source input. Always associated with OSC1 pin function.				
CLKO	0	—	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-CN21	Ι	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	0	_	CAN1 bus transmit pin.				
C2RX		SI	CAN2 bus receive pin.				
	0		CANZ bus transmit pin.				
EMUD	I/O	SI	ICD Primary Communication Channel data input/output pin.				
	1/0	SI ST	ICD Primary Communication Channel clock input/output pin.				
EMUC1	1/0	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.				
055		OT	Auxiliary Timer External Clock/Gate input in Timer mode.				
QEB	I	SI	Quadrature Encoder Phase A input in QEI mode.				
UPDN	0	CMOS	Position Up/Down Counter Direction State.				
INT0	I	ST	External interrupt 0.				
INT1	I	ST	External interrupt 1.				
INT2	I	ST	External interrupt 2.				
INT3	I	ST	External interrupt 3.				
IN I 4		SI	External Interrupt 4.				
LVDIN	I	Analog	Low-Voltage Detect Reference Voltage input pin.				
Legend: CM ST	OS = = =	CMOS co Schmitt T	ompatible input or outputAnalog= Analog inputrigger input with CMOS levelsO= OutputP= Power				

#### TABLE 1-1: DSPIC30F6010 I/O PIN DESCRIPTIONS

### TABLE 1-1: DSPIC30F6010 I/O PIN DESCRIPTIONS (CONTINUED)

Pin Name	Pin Type	Buffer Type	Description					
FLTA	Ι	ST	PWM Fault A input.					
FLTB	I	ST	PWM Fault B input.					
PWM1L	0		PWM 1 Low output.					
PWM1H	0	—	PWM 1 High output.					
PWM2L	0		PWM 2 Low output.					
PWM2H	0	—	PWM 2 High output.					
	0		PWM 3 Low output.					
	0		PWM 3 High output					
PWM4H	0		PWM 4 High output					
MCLR	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active					
MOEN	1/1	01	low Reset to the device.					
OCFA	I	ST	Compare Fault A input (for Compare channels 1, 2, 3 and 4).					
OCFB		ST	Compare Fault B input (for Compare channels 5, 6, 7 and 8).					
0C1-0C8	0	—	Compare outputs 1 through 8.					
OSC1	l	ST/CMOS	Oscillator crystal input. ST buffer when configured in RC mode; CMOS					
OSC2	I/O	—	otherwise.					
			mode. Optionally functions as CLKO in RC and EC modes.					
PGD	I/O	ST	In-Circuit Serial Programming data input/output pin.					
PGC	Ι	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	1/0	SI						
RC13-RC15	1/0	SI	POPTD is a hidiractional I/O port					
	1/0	OT ST	POPTE is a bidirectional I/O port					
RE0-RE9	1/0	0T	PORTE is a bidirectional I/O port.					
RFU-RF8	1/0	51						
RG0-RG3	1/0	SI	PORIG is a bidirectional I/O port.					
RG6-RG9	1/0	51 0T	Sum alternative against all all in mut/autorit fam CDL #4					
SUKI	1/0	SI ST	Synchronous serial clock input/output for SPI #1.					
SDO1	0		SPI #1 Data III. SPI #1 Data Out					
SS1	I	ST	SPI #1 Slave Synchronization.					
SCK2	I/O	ST	Synchronous serial clock input/output for SPI #2.					
SDI2	I	ST	SPI #2 Data In.					
SDO2	0	—	SPI #2 Data Out.					
SS2	Ι	ST	SPI #2 Slave Synchronization.					
SCL	I/O	ST	Synchronous serial clock input/output for I <sup>2</sup> C™.					
SDA	I/O	ST	Synchronous serial data input/output for I <sup>2</sup> C.					
SOSCO	0	_	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	<u> </u>	ST	Timer4 external clock input.					
Legend: CM	OS =	CMOS co	ompatible input or output Analog = Analog input					
ST	=	Schmitt 7	rigger input with CMOS levels O = Output					
I	=	Input	P = Power					

Pin Name	Pin Type	Buffer Type	Description				
U1RX	1	ST	UART1 Receive.				
U1TX	0		UART1 Transmit.				
U2RX	I	ST	UART2 Receive.				
U2TX	0	—	UART2 Transmit.				
Vdd	Р	—	Positive supply for logic and I/O pins.				
Vss	Р	—	Ground reference for logic and I/O pins.				
VREF+	Ι	Analog	Analog Voltage Reference (High) input.				
VREF-	I	Analog	Analog Voltage Reference (Low) input.				
Legend: CN	/IOS =	CMOS co	ompatible input or output Analog = Analog input				
- S1	- =	Schmitt 7	Frigger input with CMOS levels O = Output				
1	=	Input	P = Power				

### TABLE 1-1: DSPIC30F6010 I/O PIN DESCRIPTIONS (CONTINUED)

NOTES:

### 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 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 "*dsPIC30F/ 33F Programmer's Reference Manual*" (DS70157).

This document provides a summary of the dsPIC30F6010 CPU and peripheral function. For a complete description of this functionality, please refer to the "*dsPIC30F Family Reference Manual*" (DS70046).

#### 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.

#### 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 (LSB) 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<sup>®</sup> 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 Most Significant Byte (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/Subtracter 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.



#### 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:

- 1. DIVF 16/16 signed fractional divide
- 2. DIV.sd 32/16 signed divide
- 3. DIV.ud 32/16 unsigned divide
- 4. DIV. sw 16/16 signed divide
- 5. DIV.uw 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 2-1 (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.

Instruction	Function
DIVF	Signed fractional divide: Wm/Wn $\rightarrow$ W0; Rem $\rightarrow$ W1
DIV.sd	Signed divide: (Wm + 1:Wm)/Wn $\rightarrow$ W0; Rem $\rightarrow$ W1
DIV.ud	Unsigned divide: (Wm + 1:Wm)/Wn $\rightarrow$ W0; Rem $\rightarrow$ W1
DIV.sw	Signed divide: Wm/Wn $\rightarrow$ W0; Rem $\rightarrow$ W1
DIV.uw	Unsigned divide: Wm/Wn $\rightarrow$ W0; Rem $\rightarrow$ W1

#### TABLE 2-1: DIVIDE INSTRUCTIONS

#### 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/sub-tracter (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 2-2: 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



#### 2.4.1 MULTIPLIER

The 17x17-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 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  $(1-2^{1-N})$ . For a 16-bit fraction, the Q15 data range is -1.0 (0x8000) to 0.999969482 (0x7FFF), including 0 and has a precision of 3.01518x10<sup>-5</sup>. In Fractional mode, a 16x16 multiply operation generates a 1.31 product, which has a precision of 4.65661x10<sup>-10</sup>.

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/SUBTRACTER

The data accumulator consists of a 40-bit adder/subtracter with automatic sign extension logic. It can select one of two accumulators (A or B) as its preaccumulation 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/Subtracter, Overflow and Saturation

The adder/subtracter is a 40-bit adder with an optional zero input into one side and either true or complement data into the other input. In the case of addition, the carry/borrow input is active high and the other input is true data (not complemented), whereas in the case of subtraction, the carry/borrow input is active low and the other input is complemented. The adder/subtracter generates overflow status bits SA/SB and OA/OB, which are latched and reflected in the status register.

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

The adder has an additional saturation block which controls accumulator data saturation, if selected. It uses the result of the adder, the overflow status bits described above, and the 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/subtracter. When set, they indicate that the most recent operation has overflowed into the accumulator guard bits (bits 32 through 39). The OA and OB bits can also optionally generate an arithmetic warning trap when 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/subtracter, but can only be cleared by the user. When set, they indicate that the accumulator has overflowed its maximum range (bit 31 for 32-bit saturation, or bit 39 for 40-bit saturation) and will be saturated (if saturation is enabled). When saturation is not enabled, SA and SB default to bit 39 overflow and thus indicate that a catastrophic overflow has occurred. If the COVTE bit in the INTCON1 register is set, SA and SB bits will generate an arithmetic warning trap when saturation is disabled.

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

The device supports three Saturation and Overflow modes.

- 1. Bit 39 Overflow and Saturation:
  - When bit 39 overflow and saturation occurs, the saturation logic loads the maximally positive 9.31 (0x7FFFFFFFFF) or maximally negative 9.31 value (0x800000000) 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).
- Bit 31 Overflow and Saturation: When bit 31 overflow and saturation occurs, the saturation logic then loads the maximally positive 1.31 value (0x007FFFFFF) or maximally negative 1.31 value (0x0080000000) into the target accumulator. The SA or SB bit is set and remains set until cleared by the user. When this Saturation mode is in effect, the guard bits are not used (so the OA, OB or OAB bits are never set).
- 3. Bit 39 Catastrophic Overflow The bit 39 overflow status bit from the adder is used to set the SA or SB bit, which 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.
- [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/subtracter 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 "*dsPIC30F/33F 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 0x7FFFFE), 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 dsPIC30F6010



#### TABLE 3-1: PROGRAM SPACE ADDRESS CONSTRUCTION

	Access	Program Space Address						
Access Type	Space	<23>	<22:16>	<15>	<14:1>	<0>		
Instruction Access	struction Access User		PC<22:1>					
TBLRD/TBLWT	User (TBLPAG<7> = 0)	TBL	PAG<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> Da		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 least significant data word, and TBLRDH and TBLWTH access the space which contains the Most Significant data Byte.

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.

- 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 LSBs 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.
- TBLWTL: Table Write Low (refer to Section 6.0 "Flash Program Memory" for details on Flash Programming).
- 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 MSBs 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.

 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)



#### 3.1.2 DATA ACCESS FROM PROGRAM MEMORY USING PROGRAM SPACE VISIBILITY

The upper 32 Kbytes of data space may optionally be mapped into any 16K word program space page. This provides transparent access of stored constant data from X data space, without the need to use special instructions (i.e., TBLRDL/H, TBLWTL/H instructions).

Program space access through the data space occurs if the MSb of the data space EA is set and program space visibility is enabled, by setting the PSV bit in the Core Control register (CORCON). The functions of CORCON are discussed in **Section 2.4** "**DSP Engine**", DSP Engine.

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

Note that the upper half of addressable data space is always part of the X data space. Therefore, when a DSP operation uses program space mapping to access this memory region, Y data space should typically contain state (variable) data for DSP operations, whereas X data space should typically contain coefficient (constant) data.

Although each data space address, 0x8000 and higher, maps directly into a corresponding program memory address (see Figure 3-5), only the lower 16-bits of the 24-bit program word are used to contain the data. The upper 8 bits should be programmed to force an illegal instruction to maintain machine robustness. Refer to the "dsPIC30F/33F Programmer's Reference

Manual' (DS70157) for details on instruction encoding.

Note that by incrementing the PC by 2 for each program memory word, the Least Significant 15 bits of data space addresses directly map to the Least Significant 15 bits in the corresponding program space addresses. The remaining bits are provided by the Program Space Visibility Page register, PSVPAG<7:0>, as shown in Figure 3-5.

Note:	PSV access is temporarily disabled during
	Table Reads/Writes.

For instructions that use PSV which are executed outside a REPEAT loop:

- The following instructions will require one instruction cycle in addition to the specified execution time:
  - MAC class of instructions with data operand prefetch
  - MOV instructions
  - MOV.D instructions
- All other instructions will require two instruction cycles in addition to the specified execution time of the instruction.

For instructions that use PSV which are executed inside a  $\ensuremath{\mathtt{REPEAT}}$  loop:

- The following instances will require two instruction cycles in addition to the specified execution time of the instruction:
  - Execution in the first iteration
  - Execution in the last iteration
  - Execution prior to exiting the loop due to an interrupt
  - Execution upon re-entering the loop after an interrupt is serviced
- Any other iteration of the REPEAT loop will allow the instruction, accessing data using PSV, to execute in a single cycle.



FIGURE 3-5: DATA SPACE WINDOW INTO PROGRAM SPACE OPERATION

**Note:** PSVPAG is an 8-bit register, containing bits <22:15> of the program space address (i.e., it defines the page in program space to which the upper half of data space is being mapped).

#### 3.2 Data Address Space

The core has two data spaces. The data spaces can be considered either separate (for some DSP instructions), or as one unified linear address range (for MCU instructions). The data spaces are accessed using two Address Generation Units (AGUs) and separate data paths.

#### 3.2.1 DATA SPACE MEMORY MAP

The data space memory is split into two blocks, X and Y data space. A key element of this architecture is that Y space is a subset of X space, and is fully contained within X space. In order to provide an apparent linear addressing space, X and Y spaces have contiguous addresses.

When executing any instruction other than one of the MAC class of instructions, the X block consists of the 64 Kbyte data address space (including all Y addresses). When executing one of the MAC class of instructions, the X block consists of the 64 Kbyte data address space excluding the Y address block (for data reads only). In other words, all other instructions regard the entire data memory as one composite address space. The MAC class instructions extract the Y address space from data space and address it using EAs sourced from W10 and W11. The remaining X data space is addressed using W8 and W9. Both address spaces are concurrently accessed only with the MAC class instructions.

A data space memory map is shown in Figure 3-6.

Figure 3-7 shows a graphical summary of how X and Y data spaces are accessed for MCU and DSP instructions.





#### 3.2.2 DATA SPACES

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

The X data space also supports Modulo Addressing for all instructions, subject to addressing mode restrictions. Bit-Reversed Addressing is only supported for writes to X data space.

The Y data space is used in concert with the X data space by the MAC class of instructions (CLR, ED, EDAC, MAC, MOVSAC, MPY, MPY.N and MSC) to provide two concurrent data read paths. No writes occur across the Y bus. This class of instructions dedicates two W register pointers, W10 and W11, to always address Y data space, independent of X data space, whereas W8 and W9 always address X data space. Note that during accumulator write back, the data address space is considered a combination of X and Y data spaces, so the write occurs across the X bus. Consequently, the write can be to any address in the entire data space.

The Y data space can only be used for the data prefetch operation associated with the MAC class of instructions. It also supports Modulo Addressing for automated circular buffers. Of course, all other instructions can access the Y data address space through the X data path, as part of the composite linear space.

The boundary between the X and Y data spaces is defined as shown in Figure 3-6 and is not user programmable. Should an EA point to data outside its own assigned address space, or to a location outside physical memory, an all-zero word/byte will be returned. For example, although Y address space is visible by all non-MAC instructions using any addressing mode, an attempt by a MAC instruction to fetch data from that space, using W8 or W9 (X space pointers), will return 0x0000.

### TABLE 3-2:EFFECT OF INVALIDMEMORY ACCESSES

Attempted Operation	Data Returned
EA = an unimplemented address	0x0000
W8 or W9 used to access Y data space in a MAC instruction	0x0000
W10 or W11 used to access X data space in a MAC instruction	0x0000

All effective addresses are 16 bits wide and point to bytes within the data space. Therefore, the data space address range is 64 Kbytes or 32K words.

#### 3.2.3 DATA SPACE WIDTH

The core data width is 16 bits. All internal registers are organized as 16-bit wide words. Data space memory is organized in byte addressable, 16-bit wide blocks.

#### 3.2.4 DATA ALIGNMENT

To help maintain backward compatibility with PIC® MCU devices and improve data space memory usage efficiency, the dsPIC30F instruction set supports both word and byte operations. Data is aligned in data memory and registers as words, but all data space EAs resolve to bytes. Data byte reads will read the complete word, which contains the byte, using the LSb of any EA to determine which byte to select. The selected byte is placed onto the LSB of the X data path (no byte accesses are possible from the Y data path as the MAC class of instruction can only fetch words). That is, data memory and registers are organized as two parallel byte wide entities with shared (word) address decode, but separate write lines. Data byte writes only write to the corresponding side of the array or register which matches the byte address.

As a consequence of this byte accessibility, all effective address calculations (including those generated by the DSP operations, which are restricted to word sized data) are internally scaled to step through word aligned memory. For example, the core would recognize that Post-Modified Register Indirect Addressing mode, [Ws++], will result in a value of Ws + 1 for byte operations and Ws + 2 for word operations.

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

#### FIGURE 3-8: DATA ALIGNMENT

	15 <b>MSB</b>	8 7 <b>LSB</b>	)
0001	Byte 1	Byte 0	0000
0003	Byte 3	Byte 2	0002
0005	Byte 5	Byte 4	0004

All byte loads into any W register are loaded into the LSB. The MSB is not modified.

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

Although most instructions are capable of operating on word or byte data sizes, it should be noted that some instructions, including the DSP instructions, operate only on words.

#### 3.2.5 NEAR DATA SPACE

An 8 Kbyte 'near' data space is reserved in X address memory space between 0x0000 and 0x1FFF, which is directly addressable via a 13-bit absolute address field within all memory direct instructions. The remaining X address space and all of the Y address space is addressable indirectly. Additionally, the whole of X data space is addressable using MOV instructions, which support memory direct addressing with a 16-bit address field.

#### 3.2.6 SOFTWARE STACK

The dsPIC DSC device contains a software stack. W15 is used as the Stack Pointer.

The Stack Pointer always points to the first available free word and grows from lower addresses towards higher addresses. It pre-decrements for stack pops and post-increments for stack pushes, as shown in Figure 3-9. Note that for a PC push during any CALL instruction, the MSB of the PC is zero-extended before the push, ensuring that the MSB is always clear.

Note: A PC push during exception processing will concatenate the SRL register to the MSB of the PC prior to the push. There is a Stack Pointer Limit register (SPLIM) associated with the Stack Pointer. SPLIM is uninitialized at Reset. As is the case for the Stack Pointer, SPLIM<0> is forced to '0', because all stack operations must be word aligned. Whenever an effective address (EA) is generated using W15 as a source or destination pointer, the address thus generated is compared with the value in SPLIM. If the contents of the Stack Pointer (W15) and the SPLIM register are equal and a push operation is performed, a stack error trap will not occur. The stack error trap will occur on a subsequent push operation. Thus, for example, if it is desirable to cause a stack error trap when the stack grows beyond address 0x2000 in RAM, initialize the SPLIM with the value, 0x1FFE.

Similarly, a stack pointer underflow (stack error) trap is generated when the Stack Pointer address is found to be less than 0x0800, thus preventing the stack from interfering with the Special Function Register (SFR) space.

A write to the SPLIM register should not be immediately followed by an indirect read operation using W15.

#### FIGURE 3-9: CALL STACK FRAME



#### TABLE 3-3: CORE REGISTER MAP

SFR Name	Address (Home)	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
W0	0000	W0 / WREG														
W1	0002		W1													
W2	0004		W2													
W3	0006		W3													
W4	0008															
W5	000A		W5													
W6	000C		W6													
W7	000E								W7							
W8	0010								W8							
W9	0012								W9							
W10	0014								W10							
W11	0016								W11							
W12	0018								W12							
W13	001A								W13							
W14	001C								W14							
W15	001E								W15							
SPLIM	0020								SPLIN	1						
ACCAL	0022		ACCAL													
ACCAH	0024		ACCAH													
ACCAU	0026		Sign-Extension (ACCA<39>) ACCAU													
ACCBL	0028								ACCB	L						
ACCBH	002A		ACCBH													
ACCBU	002C		Sign-Extension (ACCB<39>) ACCBU													
PCL	002E		PCL													
PCH	0030		_	_		—	—			_	РСН					
TBLPAG	0032		—	_		—	—	_	_				TBLP	PAG		
PSVPAG	0034	—	— — — — — — — — PSVPAG													
RCOUNT	0036								RCOUN	NT						
DCOUNT	0038								DCOUN	T						
DOSTARTL	003A	DOSTARTL														
DOSTARTH	003C	— — — — — — — — — DOSTARTH														
DOENDL	003E	DOENDL														
DOENDH	0040	—	_	—	_	—	_	—	—	_			C	OENDH		
SR	0042	OA	OB	SA	SB	OAB	SAB	DA	DC	IPL2	IPL1	IPL0	RA	N	OV	Z

**Legend:** u = uninitialized bit

**Note:** Refer to *"dsPIC30F Family Reference Manual"* (DS70046) for descriptions of register bit fields.

#### TABLE 3-3: CORE REGISTER MAP (CONTINUED)

							/									
SFR Name	Address (Home)	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
CORCON	0044	_			US	EDT	DL2	DL1	DL0	SATA	SATB	SATDW	ACCSAT	IPL3	PSV	RND
MODCON	0046	XMODEN	YMODEN		— BWM<3:0> YWM<3:0>				XWM<3:0>							
XMODSRT	0048		XS<15:1>													
XMODEND	004A		XE<15:1>													
YMODSRT	004C		YS<15:1>													
YMODEND	004E		YE<15:1>													
XBREV	0050	BREN XB<14:0>														
DISICNT	0052	— — DISICNT<13:0>														
Lonondi		المعالمة المنا	1													

**Legend:** u = uninitialized bit

Note: Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

NOTES:

#### 4.0 ADDRESS GENERATOR UNITS

**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 "*dsPIC30F/ 33F Programmer's Reference Manual*" (DS70157).

The dsPIC DSC core contains two independent address generator units: the X AGU and Y AGU. The Y AGU supports word sized data reads for the DSP MAC class of instructions only. The dsPIC DSC AGUs support three types of data addressing:

- Linear Addressing
- Modulo (Circular) Addressing
- Bit-Reversed Addressing

Linear and Modulo Data Addressing modes can be applied to data space or program space. Bit-Reversed Addressing is only applicable to data space addresses.

#### 4.1 Instruction Addressing Modes

The addressing modes in Table 4-1 form the basis of the addressing modes optimized to support the specific features of individual instructions. The addressing modes provided in the MAC class of instructions are somewhat different from those in the other instruction types.

#### 4.1.1 FILE REGISTER INSTRUCTIONS

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

#### 4.1.2 MCU INSTRUCTIONS

The three-operand MCU instructions are of the form:

#### Operand 3 = Operand 1 < function> Operand 2

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

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

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

#### TABLE 4-1: FUNDAMENTAL ADDRESSING MODES SUPPORTED

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

### 4.1.3 MOVE AND ACCUMULATOR INSTRUCTIONS

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

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

In summary, the following addressing modes are supported by Move and Accumulator instructions:

- Register Direct
- Register Indirect
- Register Indirect Post-modified
- Register Indirect Pre-modified
- Register Indirect with Register Offset (Indexed)
- Register Indirect with Literal Offset
- 8-bit Literal
- 16-bit Literal
- Note: Not all instructions support all the addressing modes given above. Individual instructions may support different subsets of these addressing modes.

#### 4.1.4 MAC INSTRUCTIONS

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

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

Note: Register Indirect with Register Offset Addressing is only available for W9 (in X space) and W11 (in Y space). In summary, the following addressing modes are supported by the  ${\tt MAC}$  class of instructions:

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

#### 4.1.5 OTHER INSTRUCTIONS

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

#### 4.2 Modulo Addressing

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

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

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

The only exception to the usage restrictions is for buffers which have a power-of-2 length. As these buffers satisfy the start and end address criteria, they may operate in a Bidirectional mode, (i.e., address boundary checks will be performed on both the lower and upper address boundaries).

#### 4.2.1 START AND END ADDRESS

The Modulo Addressing scheme requires that a starting and an end address be specified and loaded into the 16-bit modulo buffer address registers: XMODSRT, XMODEND, YMODSRT and YMODEND (see Table 3-3)..

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

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

#### 4.2.2 W ADDRESS REGISTER SELECTION

The Modulo and Bit-Reversed Addressing Control register MODCON<15:0> contains enable flags as well as a W register field to specify the W address registers. The XWM and YWM fields select which registers will operate with Modulo Addressing. If XWM = 15, X RAGU and X WAGU Modulo Addressing are disabled. Similarly, if YWM = 15, Y AGU Modulo Addressing is disabled.

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

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



#### 4.2.3 MODULO ADDRESSING APPLICABILITY

Modulo Addressing can be applied to the effective address (EA) calculation associated with any W register. It is important to realize that the address boundaries check for addresses less than or greater than the upper (for incrementing buffers) and lower (for decrementing buffers) boundary addresses (not just equal to). Address changes may, therefore, jump beyond boundaries and still be adjusted correctly.

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

#### 4.3 Bit-Reversed Addressing

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

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

### 4.3.1 BIT-REVERSED ADDRESSING IMPLEMENTATION

Bit-Reversed Addressing is enabled when:

- BWM (W register selection) in the MODCON register is any value other than 15 (the stack can not be accessed using Bit-Reversed Addressing) and
- 2. the BREN bit is set in the XBREV register and
- 3. the addressing mode used is Register Indirect with Pre-Increment or Post-Increment.

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

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

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

When enabled, Bit-Reversed Addressing will only be executed for register indirect with pre-increment or post-increment addressing and word sized data writes. It will not function for any other addressing mode or for byte-sized data, and normal addresses will be generated instead. When Bit-Reversed Addressing is active, the W address pointer will always be added to the address modifier (XB) and the offset associated with the Register Indirect Addressing mode will be ignored. In addition, as word sized data is a requirement, the LSb of the EA is ignored (and always clear).

Note:	Modulo Addressing and Bit-Reversed
	Addressing should not be enabled
	together. In the event that the user
	attempts to do this, bit reversed address-
	ing will assume priority when active for the
	X WAGU, and X WAGU Modulo Address-
	ing will be disabled. However, Modulo
	Addressing will continue to function in the
	X RAGU.

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



#### FIGURE 4-2: BIT-REVERSED ADDRESS EXAMPLE
							,						
		Norma	al Addre	SS		Bit-Reversed Address							
A3	A2	A1	A0	Decimal	A3	A2	A1	A0	Decimal				
0	0	0	0	0	0	0	0	0	0				
0	0	0	1	1	1	0	0	0	8				
0	0	1	0	2	0	1	0	0	4				
0	0	1	1	3	1	1	0	0	12				
0	1	0	0	4	0	0	1	0	2				
0	1	0	1	5	1	0	1	0	10				
0	1	1	0	6	0	1	1	0	6				
0	1	1	1	7	1	1	1	0	14				
1	0	0	0	8	0	0	0	1	1				
1	0	0	1	9	1	0	0	1	9				
1	0	1	0	10	0	1	0	1	5				
1	0	1	1	11	1	1	0	1	13				
1	1	0	0	12	0	0	1	1	3				
1	1	0	1	13	1	0	1	1	11				
1	1	1	0	14	0	1	1	1	7				
1	1	1	1	15	1	1	1	1	15				

### TABLE 4-2: BIT-REVERSED ADDRESS SEQUENCE (16-ENTRY)

#### TABLE 4-3: BIT-REVERSED ADDRESS MODIFIER VALUES FOR XBREV REGISTER

Buffer Size (Words)	XB<14:0> Bit-Reversed Address Modifier Value
4096	0x0800
2048	0x0400
1024	0x0200
512	0x0100
256	0x0080
128	0x0040
64	0x0020
32	0x0010
16	0x0008
8	0x0004
4	0x0002
2	0x0001

NOTES:

## 5.0 INTERRUPTS

**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 "*dsPIC30F/ 33F Programmer's Reference Manual*" (DS70157).

The dsPIC30F6010 has 44 interrupt sources and 4 processor exceptions (traps), which must be arbitrated based on a priority scheme.

The CPU is responsible for reading the Interrupt Vector Table (IVT) and transferring the address contained in the interrupt vector to the program counter. The interrupt vector is transferred from the program data bus into the program counter, via a 24-bit wide multiplexer on the input of the program counter.

The Interrupt Vector Table (IVT) and Alternate Interrupt Vector Table (AIVT) are placed near the beginning of program memory (0x000004). The IVT and AIVT are shown in Figure 5-1.

The interrupt controller is responsible for preprocessing the interrupts and processor exceptions, prior to their being presented to the processor core. The peripheral interrupts and traps are enabled, prioritized and controlled using centralized special function registers:

- IFS0<15:0>, IFS1<15:0>, IFS2<15:0> All interrupt request flags are maintained in these three registers. The flags are set by their respective peripherals or external signals, and they are cleared via software.
- IEC0<15:0>, IEC1<15:0>, IEC2<15:0> All Interrupt Enable Control bits are maintained in these three registers. These control bits are used to individually enable interrupts from the peripherals or external signals.
- IPC0<15:0>... IPC11<7:0> The user assignable priority level associated with each of these 44 interrupts is held centrally in these twelve registers.
- IPL<3:0> The current CPU priority level is explicitly stored in the IPL bits. IPL<3> is present in the CORCON register, whereas IPL<2:0> are present in the status register (SR) in the processor core.
- INTCON1<15:0>, INTCON2<15:0> Global interrupt control functions are derived from these two registers. INTCON1 contains the control and status flags for the processor exceptions. The INTCON2 register controls the external interrupt request signal behavior and the use of the alternate vector table.

Note: Interrupt Flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding Enable bit. User software should ensure the appropriate Interrupt Flag bits are clear prior to enabling an interrupt.

All interrupt sources can be user assigned to one of seven priority levels, 1 through 7, via the IPCx registers. Each interrupt source is associated with an interrupt vector, as shown in Table 5-1. Levels 7 and 1 represent the highest and lowest maskable priorities, respectively.

Note:	Assigning a priority level of 0 to an inter-
	rupt source is equivalent to disabling that
	interrupt.

If the NSTDIS bit (INTCON1<15>) is set, nesting of interrupts is prevented. Thus, if an interrupt is currently being serviced, processing of a new interrupt is prevented, even if the new interrupt is of higher priority than the one currently being serviced.

Note: The IPL bits become read-only whenever the NSTDIS bit has been set to '1'.

Certain interrupts have specialized control bits for features like edge or level triggered interrupts, interrupt-on-change, etc. Control of these features remains within the peripheral module which generates the interrupt.

The DISI instruction can be used to disable the processing of interrupts of priorities 6 and lower for a certain number of instructions, during which the DISI bit (INTCON2<14>) remains set.

When an interrupt is serviced, the PC is loaded with the address stored in the vector location in Program Memory that corresponds to the interrupt. There are 63 different vectors within the IVT (refer to Figure 5-2). These vectors are contained in locations 0x000004 through 0x0000FE of program memory (refer to Figure 5-2). These locations contain 24-bit addresses, and in order to preserve robustness, an address error trap will take place should the PC attempt to fetch any of these words during normal execution. This prevents execution of random data as a result of accidentally decrementing a PC into vector space, accidentally mapping a data space address into vector space, or the PC rolling over to 0x000000 after reaching the end of implemented program memory space. Execution of a GOTO instruction to this vector space will also generate an address error trap.

## 5.1 Interrupt Priority

The user assignable Interrupt Priority (IP<2:0>) bits for each individual interrupt source are located in the Least Significant 3 bits of each nibble, within the IPCx register(s). Bit 3 of each nibble is not used and is read as a '0'. These bits define the priority level assigned to a particular interrupt by the user.

Note:	The user selectable priority levels start at
	0, as the lowest priority, and level 7, as the
	highest priority.

Since more than one interrupt request source may be assigned to a specific user specified priority level, a means is provided to assign priority within a given level. This method is called "Natural Order Priority".

Natural Order Priority is determined by the position of an interrupt in the vector table, and only affects interrupt operation when multiple interrupts with the same user-assigned priority become pending at the same time.

Table 5-1 lists the interrupt numbers and interrupt sources for the dsPIC DSC devices and their associated vector numbers.

- **Note 1:** The natural order priority scheme has 0 as the highest priority and 53 as the lowest priority.
  - **2:** The natural order priority number is the same as the INT number.

The ability for the user to assign every interrupt to one of seven priority levels implies that the user can assign a very high overall priority level to an interrupt with a low natural order priority. For example, the PLVD (Low-Voltage Detect) can be given a priority of 7. The INTO (external interrupt 0) may be assigned to priority level 1, thus giving it a very low effective priority.

#### TABLE 5-1: INTERRUPT VECTOR TABLE

INT Number	Vector Number	Interrupt Source
Highest N	atural Orde	er Priority
0	8	INT0 – External Interrupt 0
1	9	IC1 – Input Capture 1
2	10	OC1 – Output Compare 1
3	11	T1 – Timer 1
4	12	IC2 – Input Capture 2
5	13	OC2 – Output Compare 2
6	14	T2 – Timer 2
7	15	T3 – Timer 3
8	16	SPI1
9	17	U1RX – UART1 Receiver
10	18	U1TX – UART1 Transmitter
11	19	ADC – ADC Convert Done
12	20	NVM – NVM Write Complete
13	21	SI2C – I <sup>2</sup> C <sup>™</sup> Slave Interrupt
14	22	$MI2C - I^2C$ Master Interrupt
15	23	Input Change Interrupt
16	24	INT1 – External Interrupt 1
17	25	IC7 - Input Capture 7
18	26	IC8 - Input Capture 8
10	20	OC3 - Output Compare 3
20	28	OC4 - Output Compare 4
20	20	T4 – Timer 4
21	20	$T_4 = T_{\text{inter}} + T_5 = T_{\text{inter}} + T_5$
22	31	INT2 External Interrupt 2
23	32	
24	32	LI2TX LIAPT2 Transmittor
25	34	
20	35	C1 Combined IPO for CAN1
21	30	
20	30	IC3 – Input Capture 3
29	37	
30	38	ICS – Input Capture 5
31	39	
32	40	OC5 – Output Compare 5
33	41	OC6 – Output Compare 6
34	42	OC7 – Output Compare 7
35	43	UC8 – Output Compare 8
36	44	IN13 – External Interrupt 3
37	45	IN14 – External Interrupt 4
38	46	C2 – Combined IRQ for CAN2
39	47	PWM – PWM Period Match
40	48	QEI – QEI Interrupt
41	49	Reserved
42	50	LVD – Low-Voltage Detect
43	51	FLTA – PWM Fault A
44	52	FLTB – PWM Fault B
45-53	53-61	Reserved
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45-53 Lowest No	34       35       36       37       38       39       40       41       42       43       44       45       46       47       48       49       50       51       52       53-61	SPI2 SPI2 C1 – Combined IRQ for CAN1 IC3 – Input Capture 3 IC4 – Input Capture 4 IC5 – Input Capture 5 IC6 – Input Capture 6 OC5 – Output Compare 5 OC6 – Output Compare 6 OC7 – Output Compare 7 OC8 – Output Compare 8 INT3 – External Interrupt 3 INT4 – External Interrupt 3 INT4 – External Interrupt 4 C2 – Combined IRQ for CAN2 PWM – PWM Period Match QEI – QEI Interrupt Reserved LVD – Low-Voltage Detect FLTA – PWM Fault A FLTB – PWM Fault B Reserved

## 5.2 Reset Sequence

A Reset is not a true exception, because the interrupt controller is not involved in the Reset process. The processor initializes its registers in response to a Reset, which forces the PC to zero. The processor then begins program execution at location 0x000000. A GOTO instruction is stored in the first program memory location, immediately followed by the address target for the GOTO instruction. The processor executes the GOTO to the specified address and then begins operation at the specified target (start) address.

#### 5.2.1 RESET SOURCES

There are 5 sources of error which will cause a device Reset:

- Watchdog Time-out: The watchdog has timed out, indicating that the processor is no longer executing the correct flow of code.
- Uninitialized W Register Trap: An attempt to use an uninitialized W register as an address pointer will cause a Reset.
- Illegal Instruction Trap: Attempted execution of any unused opcodes will result in an illegal instruction trap. Note that a fetch of an illegal instruction does not result in an illegal instruction trap if that instruction is flushed prior to execution due to a flow change.
- Brown-out Reset (BOR): A momentary dip in the power supply to the device has been detected, which may result in malfunction.
- Trap Lockout: Occurrence of multiple Trap conditions simultaneously will cause a Reset.

## 5.3 Traps

Traps can be considered as non-maskable interrupts indicating a software or hardware error, which adhere to a predefined priority as shown in Figure 5-1. They are intended to provide the user a means to correct erroneous operation during debug and when operating within the application.

Note: If the user does not intend to take corrective action in the event of a trap error condition, these vectors must be loaded with the address of a default handler that simply contains the RESET instruction. If, on the other hand, one of the vectors containing an invalid address is called, an address error trap is generated.

Note that many of these Trap conditions can only be detected when they occur. Consequently, the questionable instruction is allowed to complete prior to trap exception processing. If the user chooses to recover from the error, the result of the erroneous action that caused the trap may have to be corrected.

There are 8 fixed priority levels for traps: Level 8 through Level 15, which implies that the IPL3 is always set during processing of a trap.

If the user is not currently executing a trap, and he sets the IPL<3:0> bits to a value of '0111' (Level 7), then all interrupts are disabled, but traps can still be processed.

#### 5.3.1 TRAP SOURCES

The following traps are provided with increasing priority. However, since all traps can be nested, priority has little effect.

#### Math Error Trap:

The math error trap executes under the following four circumstances:

- 1. Should an attempt be made to divide by zero, the divide operation will be aborted on a cycle boundary and the trap taken.
- 2. If enabled, a math error trap will be taken when an arithmetic operation on either accumulator A or B causes an overflow from bit 31 and the Accumulator Guard bits are not utilized.
- 3. If enabled, a math error trap will be taken when an arithmetic operation on either accumulator A or B causes a catastrophic overflow from bit 39 and all saturation is disabled.
- 4. If the shift amount specified in a shift instruction is greater than the maximum allowed shift amount, a trap will occur.

#### Address Error Trap:

This trap is initiated when any of the following circumstances occurs:

- 1. A misaligned data word access is attempted.
- 2. A data fetch from our unimplemented data memory location is attempted.
- 3. A data access of an unimplemented program memory location is attempted.
- 4. An instruction fetch from vector space is attempted.
  - Note: In the MAC class of instructions, wherein the data space is split into X and Y data space, unimplemented X space includes all of Y space, and unimplemented Y space includes all of X space.
- 5. Execution of a "BRA #literal" instruction or a "GOTO #literal" instruction, where literal is an unimplemented program memory address.
- Executing instructions after modifying the PC to point to unimplemented program memory addresses. The PC may be modified by loading a value into the stack and executing a RETURN instruction.

#### Stack Error Trap:

This trap is initiated under the following conditions:

- 1. The Stack Pointer is loaded with a value which is greater than the (user programmable) limit value written into the SPLIM register (stack overflow).
- 2. The Stack Pointer is loaded with a value which is less than 0x0800 (simple stack underflow).

#### Oscillator Fail Trap:

This trap is initiated if the external oscillator fails and operation becomes reliant on an internal RC backup.

### 5.3.2 HARD AND SOFT TRAPS

It is possible that multiple traps can become active within the same cycle (e.g., a misaligned word stack write to an overflowed address). In such a case, the fixed priority shown in Figure 5-2 is implemented, which may require the user to check if other traps are pending, in order to completely correct the fault.

'Soft' traps include exceptions of priority level 8 through level 11, inclusive. The arithmetic error trap (level 11) falls into this category of traps.

'Hard' traps include exceptions of priority level 12 through level 15, inclusive. The address error (level 12), stack error (level 13) and oscillator error (level 14) traps fall into this category.

Each hard trap that occurs must be acknowledged before code execution of any type may continue. If a lower priority hard trap occurs while a higher priority trap is pending, acknowledged, or is being processed, a hard trap conflict will occur.

The device is automatically Reset in a hard trap conflict condition. The TRAPR status bit (RCON<15>) is set when the Reset occurs, so that the condition may be detected in software.

#### FIGURE 5-1: TRAP VECTORS



### 5.4 Interrupt Sequence

All interrupt event flags are sampled in the beginning of each instruction cycle by the IFSx registers. A pending interrupt request (IRQ) is indicated by the flag bit being equal to a '1' in an IFSx register. The IRQ will cause an interrupt to occur if the corresponding bit in the interrupt enable (IECx) register is set. For the remainder of the instruction cycle, the priorities of all pending interrupt requests are evaluated.

If there is a pending IRQ with a priority level greater than the current processor priority level in the IPL bits, the processor will be interrupted.

The processor then stacks the current program counter and the low byte of the processor status register (SRL), as shown in Figure 5-2. The low byte of the status register contains the processor priority level at the time, prior to the beginning of the interrupt cycle. The processor then loads the priority level for this interrupt into the status register. This action will disable all lower priority interrupts until the completion of the Interrupt Service Routine.

FIGURE 5-2: INTERRUPT STACK FRAME



- Note 1: The user can always lower the priority level by writing a new value into SR. The Interrupt Service Routine must clear the interrupt flag bits in the IFSx register before lowering the processor interrupt priority, in order to avoid recursive interrupts.
  - **2:** The IPL3 bit (CORCON<3>) is always clear when interrupts are being processed. It is set only during execution of traps.

The RETFIE (Return from Interrupt) instruction will unstack the program counter and status registers to return the processor to its state prior to the interrupt sequence.

## 5.5 Alternate Vector Table

In Program Memory, the Interrupt Vector Table (IVT) is followed by the Alternate Interrupt Vector Table (AIVT), as shown in Figure 5-1. Access to the Alternate Vector Table is provided by the ALTIVT bit in the INTCON2 register. If the ALTIVT bit is set, all interrupt and exception processes will use the alternate vectors instead of the default vectors. The alternate vectors are organized in the same manner as the default vectors. The AIVT supports emulation and debugging efforts by providing a means to switch between an application and a support environment, without requiring the interrupt vectors to be reprogrammed. This feature also enables switching between applications for evaluation of different software algorithms at run time.

If the AIVT is not required, the program memory allocated to the AIVT may be used for other purposes. AIVT is not a protected section and may be freely programmed by the user.

## 5.6 Fast Context Saving

A context saving option is available using shadow registers. Shadow registers are provided for the DC, N, OV, Z and C bits in SR, and the registers W0 through W3. The shadows are only one level deep. The shadow registers are accessible using the PUSH.S and POP.S instructions only.

When the processor vectors to an interrupt, the PUSH.S instruction can be used to store the current value of the aforementioned registers into their respective shadow registers.

If an ISR of a certain priority uses the PUSH.S and POP.S instructions for fast context saving, then a higher priority ISR should not include the same instructions. Users must save the key registers in software during a lower priority interrupt, if the higher priority ISR uses fast context saving.

## 5.7 External Interrupt Requests

The interrupt controller supports five external interrupt request signals, INT0-INT4. These inputs are edge sensitive; they require a low-to-high or a high-to-low transition to generate an interrupt request. The INTCON2 register has five bits, INT0EP-INT4EP, that select the polarity of the edge detection circuitry.

## 5.8 Wake-up from Sleep and Idle

The interrupt controller may be used to wake up the processor from either Sleep or Idle modes, if Sleep or Idle mode is active when the interrupt is generated.

If an enabled interrupt request of sufficient priority is received by the interrupt controller, then the standard interrupt request is presented to the processor. At the same time, the processor will wake-up from Sleep or Idle and begin execution of the Interrupt Service Routine (ISR) needed to process the interrupt request.

### TABLE 5-2: INTERRUPT CONTROLLER REGISTER MAP

SFR Name	ADR	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	
INTCON1	0080	NSTDIS	_	—	_	—	OVATE	OVBTE	COVTE	_	-	_	MATHERR	ADDRERR	STKERR	OSCFAIL	
INTCON2	0082	ALTIVT	_	_	_	_	_	_	_	_		_	INT4EP	INT3EP	INT2EP	INT1EP	
IFS0	0084	CNIF	MI2CIF	SI2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF	T3IF	T2IF	OC2IF	IC2IF	T1IF	OC1IF	IC1IF	
IFS1	0086	IC6IF	IC5IF	IC4IF	IC3IF	C1IF	SPI2IF	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	IC8IF	IC7IF	
IFS2	0088	_	—	_	FLTBIF	FLTAIF	LVDIF	_	QEIIF	PWMIF	C2IF	INT4IF	INT3IF	OC8IF	OC7IF	OC6IF	
IEC0	008C	CNIE	MI2CIE	SI2CIE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE	T3IE	T2IE	OC2IE	IC2IE	T1IE	OC1IE	IC1IE	
IEC1	008E	IC6IE	IC5IE	IC4IE	IC3IE	C1IE	SPI2IE	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	IC8IE	IC7IE	
IEC2	0090	_	_	_	FLTBIE	FLTAIE	LVDIE	_	QEIIE	PWMIE	C2IE	INT4IE	INT3IE	OC8IE	OC7IE	OC6IE	
IPC0	0094	_	-	T1IP<2:0>	>	_	(	OC1IP<2:0>		—	IC1IP<2:0>		2:0>	—	INT0IP<2:0>		>
IPC1	0096	—	-	F31P<2:0:	>	_		T2IP<2:0>				OC2IP<	2:0>	_		IC2IP<2:0>	
IPC2	0098	—		ADIP<2:0:	>	_	U	U1TXIP<2:0>		—		U1RXIP	<2:0>	_	5	SPI1IP<2:0	>
IPC3	009A	—	(	CNIP<2:0:	>	—	N	MI2CIP<2:0>		—		SI2CIP<	:2:0>	—	NVMIP<2:0>		>
IPC4	009C	—	C	C3IP<2:0	)>	—		IC8IP<2:0:	>	—	IC7IP<2:0>		2:0>	—	INT1IP<2:0>		>
IPC5	009E	—	11	NT2IP<2:0	)>	—		T5IP<2:0>	•	—	T4IP<2:0>		2:0>	—	(	OC4IP<2:0>	>
IPC6	00A0	—		C1IP<2:0>	>	—	S	SPI2IP<2:0	)>	—	U2TXIP<2:0>		<2:0>	—	U	2RXIP<2:0	>
IPC7	00A2	_		C6IP<2:0:	>	_		IC5IP<2:0;	>	_		IC4IP<	2:0>	_		IC3IP<2:0>	
IPC8	00A4	—	C	C8IP<2:0	)>	—	0	DC7IP<2:0	>	—		OC6IP<	2:0>	—	(	DC5IP<2:0>	>
IPC9	00A6	—	P	WMIP<2:0	)>	—		C2IP<2:0>	>	—		INT41IP	<2:0>	—		NT3IP<2:0;	>
IPC10	00A8	_	F	LTAIP<2:0	)>	_	L	VDIP<2:0	>	_	—	—	—	_		QEIIP<2:0>	,
IPC11	00AA	_	_	_	_	_	—	_	_	—	_	_	_	—	F	LTBIP<2:0	>

**Legend:** u = uninitialized bit

Note: Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

## 6.0 FLASH PROGRAM MEMORY

**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 *"dsPIC30F/ 33F Programmer's Reference Manual"* (DS70157).

The dsPIC30F family of devices contains internal program Flash memory for executing user code. There are two methods by which the user can program this memory:

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

# 6.1 In-Circuit Serial Programming (ICSP)

dsPIC30F devices can be serially programmed while in the end application circuit. This is simply done with two lines for Programming Clock and Programming Data (which are named PGC and PGD respectively), and three other lines for Power (VDD), Ground (Vss) and Master Clear (MCLR). This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

## 6.2 Run-Time Self-Programming (RTSP)

RTSP is accomplished using TBLRD (table read) and TBLWT (table write) instructions.

With RTSP, the user may erase program memory, 32 instructions (96 bytes) at a time and can write program memory data, 32 instructions (96 bytes) at a time.

#### 6.3 Table Instruction Operation Summary

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

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

A 24-bit program memory address is formed using bits<7:0> of the TBLPAG register and the effective address (EA) from a W register specified in the table instruction, as shown in Figure 6-1.





## 6.4 RTSP Operation

The dsPIC30F Flash program memory is organized into rows and panels. Each row consists of 32 instructions, or 96 bytes. Each panel consists of 128 rows, or  $4K \times 24$  instructions. RTSP allows the user to erase one row (32 instructions) at a time and to program 32 instructions at one time.

Each panel of program memory contains write latches that hold 32 instructions of programming data. Prior to the actual programming operation, the write data must be loaded into the panel write latches. The data to be programmed into the panel is loaded in sequential order into the write latches; instruction 0, instruction 1, etc. The addresses loaded must always be from an even group of 32 boundary.

The basic sequence for RTSP programming is to set up a table pointer, then do a series of TBLWT instructions to load the write latches. Programming is performed by setting the special bits in the NVMCON register. 32 TBLWTL and 32 TBLWTH instructions are required to load the 32 instructions.

All of the table write operations are single word writes (2 instruction cycles), because only the table latches are written.

After the latches are written, a programming operation needs to be initiated to program the data.

The Flash Program Memory is readable, writable and erasable during normal operation over the entire VDD range.

## 6.5 RTSP Control Registers

The four SFRs used to read and write the program Flash memory are:

- NVMCON
- NVMADR
- NVMADRU
- NVMKEY

#### 6.5.1 NVMCON REGISTER

The NVMCON register controls which blocks are to be erased, which memory type is to be programmed, and start of the programming cycle.

#### 6.5.2 NVMADR REGISTER

The NVMADR register is used to hold the lower two bytes of the effective address. The NVMADR register captures the EA<15:0> of the last table instruction that has been executed and selects the row to write.

#### 6.5.3 NVMADRU REGISTER

The NVMADRU register is used to hold the upper byte of the effective address. The NVMADRU register captures the EA<23:16> of the last table instruction that has been executed.

#### 6.5.4 NVMKEY REGISTER

NVMKEY is a write-only register that is used for write protection. To start a programming or an erase sequence, the user must consecutively write 0x55 and 0xAA to the NVMKEY register. Refer to **Section 6.6 "Programming Operations"** for further details.

Note: The user can also directly write to the NVMADR and NVMADRU registers to specify a program memory address for erasing or programming.

#### 6.6 **Programming Operations**

A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. A programming operation is nominally 2 msec in duration and the processor stalls (waits) until the operation is finished. Setting the WR bit (NVMCON<15>) starts the operation, and the WR bit is automatically cleared when the operation is finished.

#### 6.6.1 PROGRAMMING ALGORITHM FOR PROGRAM FLASH

The user can erase or program one row of program Flash memory at a time. The general process is:

- 1. Read one row of program Flash (32 instruction words) and store into data RAM as a data "image".
- 2. Update the data image with the desired new data.
- 3. Erase program Flash row.
  - a) Setup NVMCON register for multi-word, program Flash, erase, and set WREN bit.
  - b) Write address of row to be erased into NVMADRU/NVMDR.
  - c) Write '55' to NVMKEY.
  - d) Write 'AA' to NVMKEY.
  - e) Set the WR bit. This will begin erase cycle.
  - f) CPU will stall for the duration of the erase cycle.
  - g) The WR bit is cleared when erase cycle ends.

- 4. Write 32 instruction words of data from data RAM "image" into the program Flash write latches.
- 5. Program 32 instruction words into program Flash.
  - Setup NVMCON register for multi-word, program Flash, program, and set WREN bit.
  - b) Write '55' to NVMKEY.
  - c) Write 'AA' to NVMKEY.
  - d) Set the WR bit. This will begin program cycle.
  - e) CPU will stall for duration of the program cycle.
  - f) The WR bit is cleared by the hardware when program cycle ends.
- 6. Repeat steps 1 through 5 as needed to program desired amount of program Flash memory.

## 6.6.2 ERASING A ROW OF PROGRAM MEMORY

Example 6-1 shows a code sequence that can be used to erase a row (32 instructions) of program memory.

#### EXAMPLE 6-1: ERASING A ROW OF PROGRAM MEMORY

	; Setu	NVMCON	for erase operation, multi wor	d	write
	; prog	ram memor	ry selected, and writes enabled	l	
		MOV	#0x4041,W0	;	
		MOV	W0,NVMCON	;	Init NVMCON SFR
	; Init	pointer	to row to be ERASED		
		MOV	<pre>#tblpage(PROG_ADDR),W0</pre>	;	
		MOV	W0,NVMADRU	;	Initialize PM Page Boundary SFR
		MOV	<pre>#tbloffset(PROG_ADDR),W0</pre>	;	Intialize in-page EA[15:0] pointer
		MOV	W0, NVMADR	;	Intialize NVMADR SFR
		DISI	#5	;	Block all interrupts with priority <7
				;	for next 5 instructions
		MOV	#0x55,W0		
		MOV	W0,NVMKEY	;	Write the 0x55 key
		MOV	#OXAA,W1	;	
		MOV	W1 NVMKEY	;	Write the OxAA key
		BSET	NVMCON, #WR	;	Start the erase sequence
I		NOP		;	Insert two NOPs after the erase
		NOP		;	command is asserted

#### 6.6.3 LOADING WRITE LATCHES

Example 6-2 shows a sequence of instructions that can be used to load the 96 bytes of write latches. 32 TBLWTL and 32 TBLWTH instructions are needed to load the write latches selected by the table pointer.

#### EXAMPLE 6-2: LOADING WRITE LATCHES

```
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
       MOV
              #0x0000,W0
       MOV
              W0 TBLPAG
                                                 ; Initialize PM Page Boundary SFR
       MOV
              #0x6000,W0
                                                 ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
              #LOW WORD 0,W2
       MOV
                                                 ;
       MOV
              #HIGH BYTE 0,W3
                                                 ;
       TBLWTL W2 [W0]
                                                ; Write PM low word into program latch
       TBLWTH W3 [W0++]
                                                 ; Write PM high byte into program latch
; 1st_program word
       MOV
              #LOW WORD 1,W2
                                                 ;
       MOV
              #HIGH BYTE 1,W3
                                                 ;
       TBLWTL W2 [W0]
                                                 ; Write PM low word into program latch
       TBLWTH W3 [W0++]
                                                 ; Write PM high byte into program latch
 2nd_program_word
;
            #LOW_WORD 2,W2
       MOV
                                                ;
       MOV
            #HIGH_BYTE_2,W3
                                                ;
       TBLWTL W2<sub>,</sub> [W0]
                                                ; Write PM low word into program latch
       TBLWTH W3 [W0++]
                                                ; Write PM high byte into program latch
; 31st_program_word
       MOV
              #LOW WORD 31,W2
                                                 ;
              #HIGH BYTE 31,W3
       MOV
                                                 ;
       TBLWTL W2 [W0]
                                                 ; Write PM low word into program latch
       TBLWTH W3 [W0++]
                                                 ; Write PM high byte into program latch
```

Note: In Example 6-2, the contents of the upper byte of W3 has no effect.

## 6.6.4 INITIATING THE PROGRAMMING SEQUENCE

For protection, the write initiate sequence for NVMKEY must be used to allow any erase or program operation to proceed. After the programming command has been executed, the user must wait for the programming time until programming is complete. The two instructions following the start of the programming sequence should be NOPS.

#### EXAMPLE 6-3: INITIATING A PROGRAMMING SEQUENCE

DISI	#5	;	Block all interrupts with priority <7
		;	for next 5 instructions
MOV	#0x55,W0		
MOV	WO,NVMKEY	;	Write the 0x55 key
MOV	#0xAA,W1	;	
MOV	W1,NVMKEY	;	Write the OxAA key
BSET	NVMCON, #WR	;	Start the erase sequence
NOP		;	Insert two NOPs after the erase
NOP		;	command is asserted

## TABLE 6-1: NVM REGISTER MAP

Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0760	WR	WREN	WRERR	_	_	_		TWRI	_			PF	ROGOP<	6:0>		
0762 NVMADR<15:0>																
0764	_	_	_	_	_	_	_	_				NVMAD	R<23:16	>		
0766	_	KEY<7:0>														
	Addr. 0760 0762 0764 0766	Addr.         Bit 15           0760         WR           0762         -           0764            0766	Addr.         Bit 15         Bit 14           0760         WR         WREN           0762	Addr.         Bit 15         Bit 14         Bit 13           0760         WR         WREN         WRERR           0762	Addr.         Bit 15         Bit 14         Bit 13         Bit 12           0760         WR         WREN         WRERR         —           0762         —         —         —         —           0764         —         —         —         —           0766         —         —         —         —         —	Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 11           0760         WR         WREN         WRERR         —         —           0762         —         —         —         —         —           0764         —         —         —         —         —           0766         —         —         —         —         —	Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 10           0760         WR         WREN         WRERR         —         —         —           0762         —         —         —         —         —         —           0764         —         —         —         —         —         —           0766         —         —         —         —         —         —	Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 11         Bit 0         Bit 9           0760         WR         WREN         WRERR         —         … <td< td=""><td>Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 11         Bit 0         Bit 3         Bit 3           0760         WR         WREN         WRERR         -         -         -         TWRI           0762         -         -         -         -         -         -         TWRI           0764         -         -         -         -         -         -         -         -           0766         -         -         -         -         -         -         -         -</td><td>Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 11         Bit 10         Bit 9         Bit 8         Bit 7           0760         WR         WREN         WRERR         -         -         TWRI         -           0762        </td><td>Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 11         Bit 0         Bit 9         Bit 8         Bit 7         Bit 6           0760         WR         WREN         WRERR         -         -         TWRI         -            0762        </td><td>Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 10         Bit 9         Bit 8         Bit 7         Bit 6         Bit 5           0760         WR         WREN         WRERR         -         -         -         TWRI         -<!--</td--><td>Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 11         Bit 0         Bit 9         Bit 8         Bit 7         Bit 6         Bit 5         Bit 4           0760         WR         WREN         WRERR         -         -         -         TWRI         -         -         F           0760         WR         WREN         -         -         -         -         TWRI         -         -         F           0762         -         -         -         -         -         -         -         -         F         -         F         -         F         -         F</td><td>Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 10         Bit 9         Bit 8         Bit 7         Bit 6         Bit 5         Bit 4         Bit 3           0760         WR         WREN         WRERR         -         -         -         TWRI         -         E         E         F         Bit 4         Bit 3         Bit 3         Bit 4         Bit 3         Bit 3         Bit 4         Bit 3         Bit 4         Bit 3         Bit 3         Bit 4         Bit 3         Bit 3         Bit 3         Bit 4         Bit 3         Bit 3         Bit 3         Bit 4         Bit 3         Bit 4         Bit 3         Bit 4         Bit 3         Bit 3         Bit 3         Bit 4         Bit 3         Bit 3         Bit 3         Bit 3         Bit 4         Bit 3         Bit 3</td></td></td<> <td>Addr.Bit 15Bit 14Bit 13Bit 12Bit 10Bit 9Bit 8Bit 7Bit 6Bit 5Bit 4Bit 3Bit 20760WRWRENWRERRTWRI<td>Addr.         Bit 15         Bit 14         Bit 12         Bit 12         Bit 10         Bit 9         Bit 8         Bit 7         Bit 6         Bit 5         Bit 4         Bit 3         Bit 2         Bit 1           0760         WR         WREN         WRERR         -         -         -         TWRI         -         E         E         E         Bit 3         Bit 2         Bit 1         Bit 1         Bit 3         Bit 3         Bit 4         Bit 3         Bit 2         Bit 1           0760         WR         WREN         WRERR         -         -         -         TWRI         -         E</td></td>	Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 11         Bit 0         Bit 3         Bit 3           0760         WR         WREN         WRERR         -         -         -         TWRI           0762         -         -         -         -         -         -         TWRI           0764         -         -         -         -         -         -         -         -           0766         -         -         -         -         -         -         -         -	Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 11         Bit 10         Bit 9         Bit 8         Bit 7           0760         WR         WREN         WRERR         -         -         TWRI         -           0762	Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 11         Bit 0         Bit 9         Bit 8         Bit 7         Bit 6           0760         WR         WREN         WRERR         -         -         TWRI         -            0762	Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 10         Bit 9         Bit 8         Bit 7         Bit 6         Bit 5           0760         WR         WREN         WRERR         -         -         -         TWRI         - </td <td>Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 11         Bit 0         Bit 9         Bit 8         Bit 7         Bit 6         Bit 5         Bit 4           0760         WR         WREN         WRERR         -         -         -         TWRI         -         -         F           0760         WR         WREN         -         -         -         -         TWRI         -         -         F           0762         -         -         -         -         -         -         -         -         F         -         F         -         F         -         F</td> <td>Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 10         Bit 9         Bit 8         Bit 7         Bit 6         Bit 5         Bit 4         Bit 3           0760         WR         WREN         WRERR         -         -         -         TWRI         -         E         E         F         Bit 4         Bit 3         Bit 3         Bit 4         Bit 3         Bit 3         Bit 4         Bit 3         Bit 4         Bit 3         Bit 3         Bit 4         Bit 3         Bit 3         Bit 3         Bit 4         Bit 3         Bit 3         Bit 3         Bit 4         Bit 3         Bit 4         Bit 3         Bit 4         Bit 3         Bit 3         Bit 3         Bit 4         Bit 3         Bit 3         Bit 3         Bit 3         Bit 4         Bit 3         Bit 3</td>	Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 11         Bit 0         Bit 9         Bit 8         Bit 7         Bit 6         Bit 5         Bit 4           0760         WR         WREN         WRERR         -         -         -         TWRI         -         -         F           0760         WR         WREN         -         -         -         -         TWRI         -         -         F           0762         -         -         -         -         -         -         -         -         F         -         F         -         F         -         F	Addr.         Bit 15         Bit 14         Bit 13         Bit 12         Bit 10         Bit 9         Bit 8         Bit 7         Bit 6         Bit 5         Bit 4         Bit 3           0760         WR         WREN         WRERR         -         -         -         TWRI         -         E         E         F         Bit 4         Bit 3         Bit 3         Bit 4         Bit 3         Bit 3         Bit 4         Bit 3         Bit 4         Bit 3         Bit 3         Bit 4         Bit 3         Bit 3         Bit 3         Bit 4         Bit 3         Bit 3         Bit 3         Bit 4         Bit 3         Bit 4         Bit 3         Bit 4         Bit 3         Bit 3         Bit 3         Bit 4         Bit 3         Bit 3         Bit 3         Bit 3         Bit 4         Bit 3         Bit 3	Addr.Bit 15Bit 14Bit 13Bit 12Bit 10Bit 9Bit 8Bit 7Bit 6Bit 5Bit 4Bit 3Bit 20760WRWRENWRERRTWRI <td>Addr.         Bit 15         Bit 14         Bit 12         Bit 12         Bit 10         Bit 9         Bit 8         Bit 7         Bit 6         Bit 5         Bit 4         Bit 3         Bit 2         Bit 1           0760         WR         WREN         WRERR         -         -         -         TWRI         -         E         E         E         Bit 3         Bit 2         Bit 1         Bit 1         Bit 3         Bit 3         Bit 4         Bit 3         Bit 2         Bit 1           0760         WR         WREN         WRERR         -         -         -         TWRI         -         E</td>	Addr.         Bit 15         Bit 14         Bit 12         Bit 12         Bit 10         Bit 9         Bit 8         Bit 7         Bit 6         Bit 5         Bit 4         Bit 3         Bit 2         Bit 1           0760         WR         WREN         WRERR         -         -         -         TWRI         -         E         E         E         Bit 3         Bit 2         Bit 1         Bit 1         Bit 3         Bit 3         Bit 4         Bit 3         Bit 2         Bit 1           0760         WR         WREN         WRERR         -         -         -         TWRI         -         E

**Legend:** u = uninitialized bit

Note: Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

NOTES:

## 7.0 DATA EEPROM MEMORY

**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 "*dsPIC30F/ 33F Programmer's Reference Manual*" (DS70157).

The Data EEPROM Memory is readable and writable during normal operation over the entire VDD range. The data EEPROM memory is directly mapped in the program memory address space.

The four SFRs used to read and write the program Flash memory are used to access data EEPROM memory, as well. As described in Section 4.0, these registers are:

- NVMCON
- NVMADR
- NVMADRU
- NVMKEY

The EEPROM data memory allows read and write of single words and 16-word blocks. When interfacing to data memory, NVMADR, in conjunction with the NVMADRU register, is used to address the EEPROM location being accessed. TBLRDL and TBLWTL instructions are used to read and write data EEPROM. The dsPIC30F6010 device has 8 Kbytes (4K words) of data EEPROM, with an address range from 0x7FF000 to 0x7FFFFE.

A word write operation should be preceded by an erase of the corresponding memory location(s). The write typically requires 2 ms to complete, but the write time will vary with voltage and temperature. A program or erase operation on the data EEPROM does not stop the instruction flow. The user is responsible for waiting for the appropriate duration of time before initiating another data EEPROM write/erase operation. Attempting to read the data EEPROM while a programming or erase operation is in progress results in unspecified data.

Control bit WR initiates write operations, similar to program Flash writes. This bit cannot be cleared, only set, in software. This bit is cleared in hardware at the completion of the write operation. The inability to clear the WR bit in software prevents the accidental or premature termination of a write operation.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a  $\overline{\text{MCLR}}$  Reset, or a WDT Time-out Reset, during normal operation. In these situations, following Reset, the user can check the WRERR bit and rewrite the location. The address register NVMADR remains unchanged.

Note: Interrupt flag bit NVMIF in the IFS0 register is set when write is complete. It must be cleared in software.

## 7.1 Reading the Data EEPROM

A TBLRD instruction reads a word at the current program word address. This example uses W0 as a pointer to data EEPROM. The result is placed in register W4, as shown in Example 7-1.

#### EXAMPLE 7-1: DATA EEPROM READ

MOV	#LOW_ADDR_WORD,W0	;	Init	Point	cer
MOV	#HIGH_ADDR_WORD,W1				
MOV	W1, TBLPAG				
TBLRDL	[ WO ], W4	;	read	data	EEPROM

#### 7.2 Erasing Data EEPROM

7.2.1 ERASING A BLOCK OF DATA EEPROM

In order to erase a block of data EEPROM, the NVMADRU and NVMADR registers must initially point to the block of memory to be erased. Configure NVMCON for erasing a block of data EEPROM, and set the ERASE and WREN bits in NVMCON register. Setting the WR bit initiates the erase, as shown in Example 7-2.

```
EXAMPLE 7-2: DATA EEPROM BLOCK ERASE
```

```
; Select data EEPROM block, ERASE, WREN bits
   MOV
           #4045,W0
   MOV
           W0 NVMCON
                                         ; Initialize NVMCON SFR
; Start erase cycle by setting WR after writing key sequence
   DISI
          #5
                                         ; Block all interrupts with priority <7
                                         ; for next 5 instructions
   MOV
           #0x55,W0
                                         ;
                                         ; Write the 0x55 key
   MOV
           W0 NVMKEY
   MOV
           #0xAA,W1
                                         ;
          W1 NVMKEY
   MOV
                                         ; Write the OxAA key
   BSET
          NVMCON, #WR
                                         ; Initiate erase sequence
   NOP
   NOP
; Erase cycle will complete in 2mS. CPU is not stalled for the Data Erase Cycle
; User can poll WR bit, use NVMIF or Timer IRQ to determine erasure complete
```

#### 7.2.2 ERASING A WORD OF DATA EEPROM

The NVMADRU and NVMADR registers must point to the block. Select erase a block of data Flash, and set the ERASE and WREN bits in NVMCON register. Setting the WR bit initiates the erase, as shown in Example 7-3.

#### EXAMPLE 7-3: DATA EEPROM WORD ERASE

```
; Select data EEPROM word, ERASE, WREN bits
   MOV
           #4044,W0
   MOV
           W0 NVMCON
; Start erase cycle by setting WR after writing key sequence
   DIST
                                         ; Block all interrupts with priority <7
           #5
                                         ; for next 5 instructions
   MOV
           #0x55,W0
                                 ;
                                 ; Write the 0x55 key
   MOV
           W0 NVMKEY
   MOV
           #0xAA,W1
                                 ;
                                 ; Write the 0xAA key
   MOV
           W1 NVMKEY
   BSET
          NVMCON, #WR
                                 ; Initiate erase sequence
   NOP
   NOP
; Erase cycle will complete in 2mS. CPU is not stalled for the Data Erase Cycle
; User can poll WR bit, use NVMIF or Timer IRQ to determine erasure complete
```

#### 7.3 Writing to the Data EEPROM

To write an EEPROM data location, the following sequence must be followed:

- 1. Erase data EEPROM word.
  - a) Select word, data EEPROM, erase and set WREN bit in NVMCON register.
  - b) Write address of word to be erased into NVMADRU/NVMADR.
  - c) Enable NVM interrupt (optional).
  - d) Write '55' to NVMKEY.
  - e) Write 'AA' to NVMKEY.
  - f) Set the WR bit. This will begin erase cycle.
  - g) Either poll NVMIF bit or wait for NVMIF interrupt.
  - h) The WR bit is cleared when the erase cycle ends.
- 2. Write data word into data EEPROM write latches.
- 3. Program 1 data word into data EEPROM.
  - a) Select word, data EEPROM, program, and set WREN bit in NVMCON register.
  - b) Enable NVM write done interrupt (optional).
  - c) Write '55' to NVMKEY.
  - d) Write 'AA' to NVMKEY.
  - e) Set The WR bit. This will begin program cycle.
  - f) Either poll NVMIF bit or wait for NVM interrupt.
  - g) The WR bit is cleared when the write cycle ends.

The write will not initiate if the above sequence is not exactly followed (write 0x55 to NVMKEY, write 0xAA to NVMCON, then set WR bit) for each word. It is strongly recommended that interrupts be disabled during this code segment.

Additionally, the WREN bit in NVMCON must be set to enable writes. This mechanism prevents accidental writes to data EEPROM, due to unexpected code execution. The WREN bit should be kept clear at all times, except when updating the EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect the current write cycle. The WR bit will be inhibited from being set unless the WREN bit is set. The WREN bit must be set on a previous instruction. Both WR and WREN cannot be set with the same instruction.

At the completion of the write cycle, the WR bit is cleared in hardware and the Non-Volatile Memory Write Complete Interrupt Flag bit (NVMIF) is set. The user may either enable this interrupt, or poll this bit. NVMIF must be cleared by software.

## 7.3.1 WRITING A WORD OF DATA EEPROM

Once the user has erased the word to be programmed, then a table write instruction is used to write one write latch, as shown in Example 7-4.

#### EXAMPLE 7-4: DATA EEPROM WORD WRITE

;	Point to data me	mory	
	MOV	#LOW_ADDR_WORD,W0	; Init pointer
	MOV	#HIGH_ADDR_WORD,W1	
	MOV	W1,TBLPAG	
	MOV	#LOW(WORD),W2	; Get data
	TBLWTL	W2 [ W0]	; Write data
;	The NVMADR captu	res last table access address	
;	Select data EEPR	OM for 1 word op	
	MOV	#0x4004,W0	
	MOV	W0,NVMCON	
;	Operate key to a	llow write operation	
	DISI #5	; Block al	l interrupts with priority <7
		; for next	5 instructions
	MOV	#0x55,W0	
	MOV	WONVMKEY	; Write the 0x55 key
	MOV	#0xAA,W1	
	MOV	W1 NVMKEY	; Write the 0xAA key
	BSET	NVMCON, #WR	; Initiate program sequence
	NOP		
	NOP		
;	Write cycle will	complete in 2mS. CPU is not stall	led for the Data Write Cycle
;	User can poll WR	. bit, use NVMIF or Timer IRQ to de	etermine write complete
	-	~	-

#### 7.3.2 WRITING A BLOCK OF DATA EEPROM

To write a block of data EEPROM, write to all sixteen latches first, then set the NVMCON register and program the block.

MOV	#LOW_ADDR_WORD,W0	;	Init pointer
MOV	#HIGH_ADDR_WORD,W1		
MOV	W1_TBLPAG		
MOV	#data1,W2	;	Get 1st data
TBLWTL	W2 [W0] ++	;	write data
MOV	#data2,W2	;	Get 2nd data
TBLWTL	W2 [W0] ++	;	write data
MOV	, #data3,W2	;	Get 3rd data
TBLWTL	W2,[W0]++	;	write data
MOV	#data4,W2	;	Get 4th data
TBLWTL	W2,[W0]++	;	write data
MOV	#data5,W2	;	Get 5th data
TBLWTL	W2,[W0]++	;	write data
MOV	#data6,W2	;	Get 6th data
TBLWTL	W2,[W0]++	;	write data
MOV	#data7,W2	;	Get 7th data
TBLWTL	W2,[W0]++	;	write data
MOV	#data8,W2	;	Get 8th data
TBLWTL	W2,[W0]++	;	write data
MOV	#data9,W2	;	Get 9th data
TBLWTL	W2,[W0]++	;	write data
MOV	#data10,W2	;	Get 10th data
TBLWTL	W2,[W0]++	;	write data
MOV	#data11,W2	;	Get 11th data
TBLWTL	W2,[W0]++	;	write data
MOV	#data12,W2	;	Get 12th data
TBLWTL	W2,[W0]++	;	write data
MOV	#data13,W2	;	Get 13th data
TBLWTL	W2,[W0]++	;	write data
MOV	#data14,W2	;	Get 14th data
TBLWTL	W2,[W0]++	;	write data
MOV	#data15,W2	;	Get 15th data
TBLWTL	W2,[W0]++	;	write data
MOV	#data16,W2	;	Get 16th data
TBLWTL	W2,[W0]++	;	write data. The NVMADR captures last table access address.
MOV	#0x400A,W0	;	Select data EEPROM for multi word op
MOV	W0,NVMCON	;	Operate Key to allow program operation
DISI	#5	;	Block all interrupts with priority <7
		;	for next 5 instructions
MOV	#0x55,W0		
MOV	W0,NVMKEY	;	Write the 0x55 key
MOV	#0xAA,W1		
MOV	W1,NVMKEY	;	Write the 0xAA key
BSET	NVMCON, #WR	;	Start write cycle
NOP			
NOP			

## EXAMPLE 7-5: DATA EEPROM BLOCK WRITE

## 7.4 Write Verify

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

## 7.5 Protection Against Spurious Write

There are conditions when the device may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built-in. On power-up, the WREN bit is cleared; also, the Power-up Timer prevents EEPROM write.

The write initiate sequence and the WREN bit together, help prevent an accidental write during brown-out, power glitch or software malfunction.

## 8.0 I/O PORTS

**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).

All of the device pins (except VDD, VSS, MCLR and OSC1/CLKI) are shared between the peripherals and the parallel I/O ports.

All I/O input ports feature Schmitt Trigger inputs for improved noise immunity.

## 8.1 Parallel I/O (PIO) Ports

When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled. The I/O pin may be read, but the output driver for the Parallel Port bit will be disabled. If a peripheral is enabled, but the peripheral is not actively driving a pin, that pin may be driven by a port.

All port pins have three registers directly associated with the operation of the port pin. The data direction register (TRISx) determines whether the pin is an input or an output. If the Data Direction bit is a '1', then the pin is an input. All port pins are defined as inputs after a Reset. Reads from the latch (LATx), read the latch. Writes to the latch, write the latch (LATx). Reads from the port (PORTx), read the port pins, and writes to the port pins, write the latch (LATx).

Any bit and its associated data and control registers that are not valid for a particular device will be disabled. That means the corresponding LATx and TRISx registers and the port pin will read as zeros.

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

The format of the registers for PORTA are shown in Table 8-1.

The TRISA (Data Direction Control) register controls the direction of the RA<7:0> pins, as well as the INTx pins and the VREF pins. The LATA register supplies data to the outputs, and is readable/writable. Reading the PORTA register yields the state of the input pins, while writing the PORTA register modifies the contents of the LATA register.

A parallel I/O (PIO) port that shares a pin with a peripheral is, in general, subservient to the peripheral. The peripheral's output buffer data and control signals are provided to a pair of multiplexers. The multiplexers select whether the peripheral or the associated port has ownership of the output data and control signals of the I/O pad cell. Figure 8-2 shows how ports are shared with other peripherals, and the associated I/O cell (pad) to which they are connected. Table 8-1 shows the formats of the registers for the shared ports, PORTB through PORTG.







### FIGURE 8-2: BLOCK DIAGRAM OF A SHARED PORT STRUCTURE

## 8.2 Configuring Analog Port Pins

The use of the ADPCFG and TRIS registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bit set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

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

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

#### 8.2.1 I/O PORT WRITE/READ TIMING

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

#### EXAMPLE 8-1: PORT WRITE/READ EXAMPLE

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

## TABLE 8-1: dsPIC30F6010 PORT REGISTER MAP

		1	1	r	1					1			1	r			
SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit
TRISA	02C0	TRISA15	TRISA14	_	_	_	TRISA10	TRISA9	_	—	_	_		—	_	_	_
PORTA	02C2	RA15	RA14	—	_	_	RA10	RA9	—	—	—	—	_	—	—	—	_
LATA	02C4	LATA15	LATA14	_	_	_	LATA10	LATA9	—	—	_	_	_	—	_	_	l
TRISB	02C6	TRISB15	TRISB14	TRISB13	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRIS
PORTB	02C8	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RE
LATB	02CA	LATB15	LATB14	LATB13	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LAT
TRISC	02CC	TRISC15	TRISC14	TRISC13	_	-	_	_	_	—	_	_	_	TRISC3	_	TRISC1	
PORTC	02CE	RC15	RC14	RC13	_	_	_	_	—	—	_	_	_	RC3	_	RC1	_
LATC	02D0	LATC15	LATC14	LATC13	_	-	_	_	_	—	_	_	_	LATC3	_	LATC1	
TRISD	02D2	TRISD15	TRISD14	TRISD13	TRISD12	TRISD11	TRISD10	TRISD9	TRISD8	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRIS
PORTD	02D4	RD15	RD14	RD13	RD12	RD11	RD10	RD9	RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1	R
LATD	02D6	LATD15	LATD14	LATD13	LATD12	LATD11	LATD10	LATD9	LATD8	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LAT
TRISE	02D8	—	—	_	_	_	_	TRISE9	TRISE8	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRIS
PORTE	02DA	_	_	_	_	_	_	RE9	RE8	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE
LATE	02DC	_	_	_	_	_	_	LATE9	LATE8	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LAT
TRISF	02DE	_	_	_	_	_	_	_	TRISF8	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRIS
PORTF	02E0	_	_	_	_	_	_	_	RF8	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF
LATF	02E2	_	_	_	_	_	_	_	LATF8	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LAT
TRISG	02E4	—	—	—	—	—	—	TRISG9	TRISG8	TRISG7	TRISG6	—	—	TRISG3	TRISG2	TRISG1	TRIS
PORTG	02E6	_	—	—	_		—	RG9	RG8	RG7	RG6			RG3	RG2	RG1	RC
LATG	02E8	_	_	_	_	_	_	LATG9	LATG8	LATG7	LATG6	_	_	LATG3	LATG2	LATG1	LAT

Legend: u = uninitialized bit

Note: Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

## 8.3 Input Change Notification Module

The Input Change Notification module provides the dsPIC30F devices the ability to generate interrupt requests to the processor in response to a change-of-state on selected input pins. This module is capable of detecting input change-of-states even in Sleep mode, when the clocks are disabled. There are 22 external signals (CN0 through CN21) that may be selected (enabled) for generating an interrupt request on a change-of-state.

Please refer to the Pin Diagram for CN pin locations.

#### TABLE 8-2: INPUT CHANGE NOTIFICATION REGISTER MAP (BITS 15-8)

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Reset State
CNEN1	00C0	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE	0000 0000 0000 0000
CNEN2	00C2	_	-	_	_	—				0000 0000 0000 0000
CNPU1	00C4	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	0000 0000 0000 0000
CNPU2	00C6	-		_	_	_				0000 0000 0000 0000

**Legend:** u = uninitialized bit

#### TABLE 8-3:INPUT CHANGE NOTIFICATION REGISTER MAP (BITS 7-0)

SFR Name	Addr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
CNEN1	00C0	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000 0000 0000 0000
CNEN2	00C2	_	_	CN21IE	CN20IE	CN19IE	CN18IE	CN17IE	CN16IE	0000 0000 0000 0000
CNPU1	00C4	CN7PUE	CN6PUE	CN5PUE	CN4PUE	<b>CN3PUE</b>	CN2PUE	CN1PUE	CN0PUE	0000 0000 0000 0000
CNPU2	00C6	_	_	CN21PUE	CN20PUE	CN19PUE	CN18PUE	CN17PUE	CN16PUE	0000 0000 0000 0000

**Legend:** u = uninitialized bit

## 9.0 TIMER1 MODULE

**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).

This section describes the 16-bit General Purpose (GP) Timer1 module and associated operational modes.

Note: Timer1 is a 'Type A' timer. Please refer to the specifications for a Type A timer in Section 24.0 "Electrical Characteristics" of this document.

The following sections provide a detailed description, including setup and control registers along with associated block diagrams for the operational modes of the timers.

The Timer1 module is a 16-bit timer which can serve as the time counter for the real-time clock, or operate as a free running interval timer/counter. The 16-bit timer has the following modes:

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

Further, the following operational characteristics are supported:

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

These operating modes are determined by setting the appropriate bit(s) in the 16-bit SFR, T1CON. Figure 9-1 presents a block diagram of the 16-bit timer module.

**16-bit Timer Mode:** In the 16-bit Timer mode, the timer increments on every instruction cycle up to a match value, preloaded into the period register PR1, then resets to '0' and continues to count.

When the CPU goes into the Idle mode, the timer will stop incrementing, unless the TSIDL (T1CON<13>) bit = 0. If TSIDL = 1, the timer module logic will resume the incrementing sequence upon termination of the CPU Idle mode.

**16-bit Synchronous Counter Mode:** In the 16-bit Synchronous Counter mode, the timer increments on the rising edge of the applied external clock signal, which is synchronized with the internal phase clocks. The timer counts up to a match value preloaded in PR1, then resets to 0 and continues.

When the CPU goes into the Idle mode, the timer will stop incrementing, unless the respective TSIDL bit = 0. If TSIDL = 1, the timer module logic will resume the incrementing sequence upon termination of the CPU Idle mode.

**16-bit Asynchronous Counter Mode:** In the 16-bit Asynchronous Counter mode, the timer increments on every rising edge of the applied external clock signal. The timer counts up to a match value preloaded in PR1, then resets to 0 and continues.

When the timer is configured for the Asynchronous mode of operation and the CPU goes into the Idle mode, the timer will stop incrementing if TSIDL = 1.





## 9.1 Timer Gate Operation

The 16-bit timer can be placed in the Gated Time Accumulation mode. This mode allows the internal TCY to increment the respective timer when the gate input signal (T1CK pin) is asserted high. Control bit TGATE (T1CON<6>) must be set to enable this mode. The timer must be enabled (TON = 1) and the timer clock source set to internal (TCS = 0).

When the CPU goes into the Idle mode, the timer will stop incrementing, unless TSIDL = 0. If TSIDL = 1, the timer will resume the incrementing sequence upon termination of the CPU Idle mode.

## 9.2 Timer Prescaler

The input clock (Fosc/4 or external clock) to the 16-bit Timer, has a prescale option of 1:1, 1:8, 1:64, and 1:256 selected by control bits TCKPS<1:0> (T1CON<5:4>). The prescaler counter is cleared when any of the following occurs:

- a write to the TMR1 register
- clearing of the TON bit (T1CON<15>)
- device Reset such as POR and BOR

However, if the timer is disabled (TON = 0), then the timer prescaler cannot be reset since the prescaler clock is halted.

TMR1 is not cleared when T1CON is written. It is cleared by writing to the TMR1 register.

## 9.3 Timer Operation During Sleep Mode

During CPU Sleep mode, the timer will operate if:

- The timer module is enabled (TON = 1) and
- The timer clock source is selected as external (TCS = 1) and
- The TSYNC bit (T1CON<2>) is asserted to a logic '0', which defines the external clock source as asynchronous

When all three conditions are true, the timer will continue to count up to the period register and be reset to 0x0000.

When a match between the timer and the period register occurs, an interrupt can be generated, if the respective Timer Interrupt Enable bit is asserted.

## 9.4 Timer Interrupt

The 16-bit timer has the ability to generate an interrupt on period match. When the timer count matches the period register, the T1IF bit is asserted and an interrupt will be generated, if enabled. The T1IF bit must be cleared in software. The timer interrupt flag T1IF is located in the IFS0 control register in the Interrupt Controller.

When the Gated Time Accumulation mode is enabled, an interrupt will also be generated on the falling edge of the gate signal (at the end of the accumulation cycle).

Enabling an interrupt is accomplished via the respective Timer Interrupt Enable bit, T1IE. The Timer Interrupt Enable bit is located in the IEC0 control register in the Interrupt Controller.

### 9.5 Real-Time Clock

Timer1, when operating in Real-Time Clock (RTC) mode, provides time-of-day and event time stamping capabilities. Key operational features of the RTC are:

- Operation from 32 kHz LP oscillator
- 8-bit prescaler
- Low power
- Real-Time Clock Interrupts

These operating modes are determined by setting the appropriate bit(s) in the T1CON Control register

#### FIGURE 9-2: RECOMMENDED COMPONENTS FOR TIMER1 LP OSCILLATOR RTC



#### 9.5.1 RTC OSCILLATOR OPERATION

When the TON = 1, TCS = 1 and TGATE = 0, the timer increments on the rising edge of the 32 kHz LP oscillator output signal, up to the value specified in the period register, and is then reset to '0'.

The TSYNC bit must be asserted to a logic '0' (Asynchronous mode) for correct operation.

Enabling LPOSCEN (OSCCON<1>) will disable the normal Timer and Counter modes and enable a timer carry-out wake-up event.

When the CPU enters Sleep mode, the RTC will continue to operate, provided the 32 kHz external crystal oscillator is active and the control bits have not been changed. The TSIDL bit should be cleared to '0' in order for RTC to continue operation in Idle mode.

#### 9.5.2 RTC INTERRUPTS

When an interrupt event occurs, the respective interrupt flag, T1IF, is asserted and an interrupt will be generated, if enabled. The T1IF bit must be cleared in software. The respective Timer interrupt flag, T1IF, is located in the IFS0 status register in the Interrupt Controller.

Enabling an interrupt is accomplished via the respective Timer Interrupt Enable bit, T1IE. The Timer Interrupt Enable bit is located in the IEC0 control register in the Interrupt Controller.

## TABLE 9-1: TIMER1 REGISTER MAP

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	B
TMR1	0100		Timer 1 Register														
PR1	0102		Period Register 1														
T1CON	0104	TON	_	TSIDL	_	_	_	_	_	_	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	
Logond		hinitializa	d hit						-						-		

**Legend:** u = uninitialized bit

Note: Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

## 10.0 TIMER2/3 MODULE

**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).

This section describes the 32-bit General Purpose Timer module (Timer2/3) and associated operational modes. Figure 10-1 depicts the simplified block diagram of the 32-bit Timer2/3 module. Figure 10-2 and Figure 10-3 show Timer2/3 configured as two independent 16-bit timers; Timer2 and Timer3, respectively.

Note:	Timer2 is a 'Type B' timer and Timer3 is a 'Type C' timer. Please refer to the appro-
	priate timer type in Section 24.0 "Electri- cal Characteristics" of this document.

The Timer2/3 module is a 32-bit timer, which can be configured as two 16-bit timers, with selectable operating modes. These timers are utilized by other peripheral modules such as:

- Input Capture
- Output Compare/Simple PWM

The following sections provide a detailed description, including setup and control registers, along with associated block diagrams for the operational modes of the timers.

The 32-bit timer has the following modes:

- Two independent 16-bit timers (Timer2 and Timer3) with all 16-bit operating modes (except Asynchronous Counter mode)
- Single 32-bit Timer operation
- Single 32-bit Synchronous Counter

Further, the following operational characteristics are supported:

- ADC Event Trigger
- Timer Gate Operation
- Selectable Prescaler Settings
- · Timer Operation during Idle and Sleep modes
- Interrupt on a 32-bit Period Register Match

These operating modes are determined by setting the appropriate bit(s) in the 16-bit T2CON and T3CON SFRs.

For 32-bit timer/counter operation, Timer2 is the least significant word and Timer3 is the most significant word of the 32-bit timer.

Note: For 32-bit timer operation, T3CON control bits are ignored. Only T2CON control bits are used for setup and control. Timer 2 clock and gate inputs are utilized for the 32-bit timer module, but an interrupt is generated with the Timer3 interrupt flag (T3IF) and the interrupt is enabled with the Timer3 Interrupt Enable bit (T3IE).

**16-bit Mode:** In the 16-bit mode, Timer2 and Timer3 can be configured as two independent 16-bit timers. Each timer can be set up in either 16-bit Timer mode or 16-bit Synchronous Counter mode. See **Section 9.0 "Timer1 Module**, for details on these two operating modes.

The only functional difference between Timer2 and Timer3 is that Timer2 provides synchronization of the clock prescaler output. This is useful for high-frequency external clock inputs.

**32-bit Timer Mode:** In the 32-bit Timer mode, the timer increments on every instruction cycle up to a match value, preloaded into the combined 32-bit period register PR3/PR2, then resets to 0 and continues to count.

For synchronous 32-bit reads of the Timer2/Timer3 pair, reading the least significant word (TMR2 register) will cause the most significant word to be read and latched into a 16-bit holding register, termed TMR3HLD.

For synchronous 32-bit writes, the holding register (TMR3HLD) must first be written to. When followed by a write to the TMR2 register, the contents of TMR3HLD will be transferred and latched into the MSB of the 32-bit timer (TMR3).

**32-bit Synchronous Counter Mode:** In the 32-bit Synchronous Counter mode, the timer increments on the rising edge of the applied external clock signal, which is synchronized with the internal phase clocks. The timer counts up to a match value preloaded in the combined 32-bit period register PR3/PR2, then resets to '0' and continues.

When the timer is configured for the Synchronous Counter mode of operation and the CPU goes into the Idle mode, the timer will stop incrementing, unless the TSIDL (T2CON<13>) bit = 0. If TSIDL = 1, the timer module logic will resume the incrementing sequence upon termination of the CPU Idle mode.

#### FIGURE 10-1: 32-BIT TIMER2/3 BLOCK DIAGRAM







### FIGURE 10-3: 16-BIT TIMER3 BLOCK DIAGRAM (TYPE C TIMER)



## 10.1 Timer Gate Operation

The 32-bit timer can be placed in the Gated Time Accumulation mode. This mode allows the internal TCY to increment the respective timer when the gate input signal (T2CK pin) is asserted high. Control bit TGATE (T2CON<6>) must be set to enable this mode. When in this mode, Timer2 is the originating clock source. The TGATE setting is ignored for Timer3. The timer must be enabled (TON = 1) and the timer clock source set to internal (TCS = 0).

The falling edge of the external signal terminates the count operation, but does not reset the timer. The user must reset the timer in order to start counting from zero.

## 10.2 ADC Event Trigger

When a match occurs between the 32-bit timer (TMR3/ TMR2) and the 32-bit combined period register (PR3/ PR2), a special ADC trigger event signal is generated by Timer3.

### 10.3 Timer Prescaler

The input clock (FOSC/4 or external clock) to the timer has a prescale option of 1:1, 1:8, 1:64, and 1:256 selected by control bits TCKPS<1:0> (T2CON<5:4> and T3CON<5:4>). For the 32-bit timer operation, the originating clock source is Timer2. The prescaler operation for Timer3 is not applicable in this mode. The prescaler counter is cleared when any of the following occurs:

- a write to the TMR2/TMR3 register
- clearing either of the TON (T2CON<15> or T3CON<15>) bits to '0'
- device Reset such as POR and BOR

However, if the timer is disabled (TON = 0), then the Timer 2 prescaler cannot be reset, since the prescaler clock is halted.

TMR2/TMR3 is not cleared when T2CON/T3CON is written.

## 10.4 Timer Operation During Sleep Mode

During CPU Sleep mode, the timer will not operate, because the internal clocks are disabled.

## 10.5 Timer Interrupt

The 32-bit timer module can generate an interrupt on period match, or on the falling edge of the external gate signal. When the 32-bit timer count matches the respective 32-bit period register, or the falling edge of the external "gate" signal is detected, the T3IF bit (IFS0<7>) is asserted and an interrupt will be generated if enabled. In this mode, the T3IF interrupt flag is used as the source of the interrupt. The T3IF bit must be cleared in software.

Enabling an interrupt is accomplished via the respective Timer Interrupt Enable bit, T3IE (IEC0<7>).

## TABLE 10-1: TIMER2/3 REGISTER MAP

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	В
TMR2	0106								Ti	mer2 Regist	er						
TMR3HLD	0108						Timer:	3 Holding	g Registe	er (For 32-bi	t timer ope	rations only	y)				
TMR3	010A		Timer3 Register														
PR2	010C	Period Register 2															
PR3	010E		_			_			Pe	riod Registe	r 3			_		_	
T2CON	0110	TON	_	TSIDL	_		_	-	_	_	TGATE	TCKPS1	TCKPS0	T32	_	TCS	
T3CON	0112	TON	_	TSIDL	_		_	_	—	_	TGATE	TCKPS1	TCKPS0		_	TCS	
Larrande		- ا م الا ما ا	بلاما امم														

**Legend:** u = uninitialized bit

Note: Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

NOTES:

## 11.0 TIMER4/5 MODULE

**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).

This section describes the second 32-bit General Purpose Timer module (Timer4/5) and associated operational modes. Figure 11-1 depicts the simplified block diagram of the 32-bit Timer4/5 Module. Figure 11-2 and Figure 11-3 show Timer4/5 configured as two independent 16-bit timers, Timer4 and Timer5, respectively.

Note: Timer4 is a 'Type B' timer and Timer5 is a 'Type C' timer. Please refer to the appropriate timer type in Section 24.0 "Electrical Characteristics" of this document. The Timer4/5 module is similar in operation to the Timer 2/3 module. However, there are some differences, which are listed below:

- The Timer4/5 module does not support the ADC Event Trigger feature
- Timer4/5 can not be utilized by other peripheral modules such as Input Capture and Output Compare

The operating modes of the Timer4/5 module are determined by setting the appropriate bit(s) in the 16-bit T4CON and T5CON SFRs.

For 32-bit timer/counter operation, Timer4 is the least significant word and Timer5 is the most significant word of the 32-bit timer.

Note: For 32-bit timer operation, T5CON control bits are ignored. Only T4CON control bits are used for setup and control. Timer4 clock and gate inputs are utilized for the 32-bit timer module, but an interrupt is generated with the Timer5 interrupt flag (T5IF) and the interrupt is enabled with the Timer5 Interrupt Enable bit (T5IE).



FIGURE 11-1: 32-BIT TIMER4/5 BLOCK DIAGRAM

#### FIGURE 11-2: 16-BIT TIMER4 BLOCK DIAGRAM (TYPE B TIMER)



## FIGURE 11-3: 16-BIT TIMER5 BLOCK DIAGRAM (TYPE C TIMER)



## TABLE 11-1: TIMER4/5 REGISTER MAP

0114								БКО		BIto	BIt 5	BIT 4	BIT 3	Bit 2	Bit 1	i E
	Timer 4 Register															
0116						Time	er 5 Holo	ling Reg	ister (For 3	2-bit oper	ations only)					
0118		Timer 5 Register														
011A	Period Register 4															
011C		_						Pe	riod Regist	er 5	-		_		_	
011E	TON	_	TSIDL	—	_	_	_	_		TGATE	TCKPS1	TCKPS0	T45	_	TCS	
0120	TON	_	TSIDL	_	_	_	_	_	_	TGATE	TCKPS1	TCKPS0	_	_	TCS	
01 01 01 01 01	16 18 1A 1C 1E 20	116           118           1A           1C           1E           120	116 118 14 16 112 112 10 115 100 100 100 100 100 100	116       118       1A       1C       1E     TON       20     TON	116       118       1A       1C       1E     TON       20     TON	116       118       1A       1C       1E     TON       20     TON         TSIDL	Time       Time       Time       Time       TIME       TON     —       TSIDL     —       10	Timer 5 Hold       Timer 5 Hold       118       1A       1A       1C       1E     TON — TSIDL — — — — —       20     TON — TSIDL — — — — —     —	Timer 5 Holding Reg       Pe       Timer 5 Holding Reg       Timer 5 Holding Reg       Timer 5 Holding Reg       T	Timer 5 Holding Register (For 3       Timer 5 Regist       Period Register       Period Register       TON — TSIDL — — — — — — —       20 TON — TSIDL — — — — — — — — —	Timer 5 Holding Register (For 32-bit operative operations)       Timer 5 Holding Register (For 32-bit operative operations)       Timer 5 Register       Period Register (For 32-bit operative oper	Timer 5 Holding Register (For 32-bit operations only)         Timer 5 Holding Register (For 32-bit operations only)         Timer 5 Register         Period Register (For 32-bit operations only)         Timer 5 Register         Timer 5 Register         Period Register 5         TON — TSIDL — — — — — — — TGATE TCKPS1         20       TON — TSIDL — — — — — — — — TGATE TCKPS1	Timer 5 Holding Register (For 32-bit operations only)         Timer 5 Holding Register (For 32-bit operations only)         Timer 5 Register         Timer 5 Register         Period Register (For 32-bit operations only)         Timer 5 Register         Timer 5 Register         Period Register         TON — TSIDL — — — — — — TGATE TCKPS1 TCKPS0         TON — TSIDL — — — — — — — TGATE TCKPS1 TCKPS0	Timer 5 Holding Register (For 32-bit operations only)         Timer 5 Holding Register (For 32-bit operations only)         Timer 5 Register         Timer 5 Register         Period Register (For 32-bit operations only)         Timer 5 Register         Timer 5 Register         Period Register /         TON — TSIDL — — — — — — TGATE TCKPS1 TCKPS0 T45         TON — TSIDL — — — — — — — TGATE TCKPS1 TCKPS0 —	Timer 5 Holding Register (For 32-bit operations only)         Timer 5 Holding Register (For 32-bit operations only)         Timer 5 Register         Timer 5 Register         Period Register 4         TON — TSIDL — — — — — — TGATE TCKPS1 TCKPS0 T45 —         20       TON — TSIDL — — — — — — — TGATE TCKPS1 TCKPS0 — — —	Timer 5 Holding Register (For 32-bit operations only)         Timer 5 Holding Register (For 32-bit operations only)         Timer 5 Register         Timer 5 Register         Period Register 4         TON - TSIDL TGATE       TCKPS1       TCKPS0       T45       - TCS         20       TON - TSIDL TGATE       TCKPS1       TCKPS0       TCS

**Legend:** u = uninitialized bit

Note: Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

NOTES:
## 12.0 INPUT CAPTURE MODULE

**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).

This section describes the Input Capture module and associated operational modes. The features provided by this module are useful in applications requiring Frequency (Period) and Pulse measurement. Figure 12-1 depicts a block diagram of the Input Capture module. Input capture is useful for such modes as:

- Frequency/Period/Pulse Measurements
- · Additional sources of External Interrupts

The key operational features of the Input Capture module are:

- Simple Capture Event mode
- Timer2 and Timer3 mode selection
- Interrupt on input capture event

These operating modes are determined by setting the appropriate bits in the ICxCON register (where x = 1,2,...,N). The dsPIC30F6010 device has 8 capture channels.

#### 12.1 Simple Capture Event Mode

The simple capture events in the dsPIC30F product family are:

- Capture every falling edge
- Capture every rising edge
- · Capture every 4th rising edge
- · Capture every 16th rising edge
- · Capture every rising and falling edge

These simple Input Capture modes are configured by setting the appropriate bits ICM<2:0> (ICxCON<2:0>).

#### 12.1.1 CAPTURE PRESCALER

There are four input capture prescaler settings, specified by bits ICM<2:0> (ICxCON<2:0>). Whenever the capture channel is turned off, the prescaler counter will be cleared. In addition, any Reset will clear the prescaler counter.





### 12.1.2 CAPTURE BUFFER OPERATION

Each capture channel has an associated FIFO buffer, which is four 16-bit words deep. There are two status flags, which provide status on the FIFO buffer:

- ICBFNE Input Capture Buffer Not Empty
- ICOV Input Capture Overflow

The ICBFNE will be set on the first input capture event and remain set until all capture events have been read from the FIFO. As each word is read from the FIFO, the remaining words are advanced by one position within the buffer.

In the event that the FIFO is full with four capture events and a fifth capture event occurs prior to a read of the FIFO, an overflow condition will occur and the ICOV bit will be set to a logic '1'. The fifth capture event is lost and is not stored in the FIFO. No additional events will be captured until all four events have been read from the buffer.

If a FIFO read is performed after the last read and no new capture event has been received, the read will yield indeterminate results.

## 12.1.3 TIMER2 AND TIMER3 SELECTION MODE

Each capture channel can select between one of two timers for the time base, Timer2 or Timer3.

Selection of the timer resource is accomplished through SFR bit ICTMR (ICxCON<7>). Timer3 is the default timer resource available for the input capture module.

#### 12.1.4 HALL SENSOR MODE

When the input capture module is set for capture on every edge, rising and falling, ICM<2:0> = 001, the following operations are performed by the input capture logic:

- The input capture interrupt flag is set on every edge, rising and falling.
- The interrupt on Capture mode setting bits, ICI<1:0>, is ignored, since every capture generates an interrupt.
- A capture overflow condition is not generated in this mode.

### 12.2 Input Capture Operation During Sleep and Idle Modes

An input capture event will generate a device wake-up or interrupt, if enabled, if the device is in CPU Idle or Sleep mode.

Independent of the timer being enabled, the input capture module will wake-up from the CPU Sleep or Idle mode when a capture event occurs, if ICM<2:0> = 111 and the Interrupt Enable bit is asserted. The same wake-up can generate an interrupt, if the conditions for processing the interrupt have been satisfied. The wake-up feature is useful as a method of adding extra external pin interrupts.

#### 12.2.1 INPUT CAPTURE IN CPU SLEEP MODE

CPU Sleep mode allows input capture module operation with reduced functionality. In the CPU Sleep mode, the ICI<1:0> bits are not applicable, and the input capture module can only function as an external interrupt source.

The capture module must be configured for interrupt only on the rising edge (ICM<2:0> = 111), in order for the input capture module to be used while the device is in Sleep mode. The prescale settings of 4:1 or 16:1 are not applicable in this mode.

## 12.2.2 INPUT CAPTURE IN CPU IDLE MODE

CPU Idle mode allows input capture module operation with full functionality. In the CPU Idle mode, the Interrupt mode selected by the ICI<1:0> bits are applicable, as well as the 4:1 and 16:1 capture prescale settings, which are defined by control bits ICM<2:0>. This mode requires the selected timer to be enabled. Moreover, the ICSIDL bit must be asserted to a logic '0'.

If the input capture module is defined as ICM<2:0> = 111 in CPU Idle mode, the input capture pin will serve only as an external interrupt pin.

## 12.3 Input Capture Interrupts

The input capture channels have the ability to generate an interrupt, based upon the selected number of capture events. The selection number is set by control bits ICI<1:0> (ICxCON<6:5>).

Each channel provides an interrupt flag (ICxIF) bit. The respective capture channel interrupt flag is located in the corresponding IFSx Status register.

Enabling an interrupt is accomplished via the respective capture channel interrupt enable (ICxIE) bit. The Capture Interrupt Enable bit is located in the corresponding IEC Control register.

## TABLE 12-1: INPUT CAPTURE REGISTER MAP

	-				i					i		1			
SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6 Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
IC1BUF	0140					_		Input	1 Captur	e Register	ſ				
IC1CON	0142	_	_	ICSIDL	_	_	_	_	_	ICTMR	ICI<1:0>	ICOV	ICBNE	10	CM<2:0>
IC2BUF	0144							Input	t 2 Captur	e Register	r			_	
IC2CON	0146	_	—	ICSIDL	_	_	_	_	_	ICTMR	ICI<1:0>	ICOV	ICBNE	10	CM<2:0>
<b>IC3BUF</b>	0148		-	-		-	-	Input	t 3 Captur	e Register	r	-	-		
IC3CON	014A	_	_	ICSIDL	_	_	_	_	_	ICTMR	ICI<1:0>	ICOV	ICBNE	10	CM<2:0>
IC4BUF	014C							Input	t 4 Captur	e Register	r				
IC4CON	014E	_	—	ICSIDL	_	_	_	_	_	ICTMR	ICI<1:0>	ICOV	ICBNE	10	CM<2:0>
IC5BUF	0150		-					Input	t 5 Captur	e Register	r				
IC5CON	0152	_	—	ICSIDL	_	_	_	_	_	ICTMR	ICI<1:0>	ICOV	ICBNE	10	CM<2:0>
IC6BUF	0154		-					Input	t 6 Captur	e Register	r				
IC6CON	0156	_	_	ICSIDL	_	_	_	_	_	ICTMR	ICI<1:0>	ICOV	ICBNE	10	CM<2:0>
IC7BUF	0158							Input	7 Captur	e Register	r				
IC7CON	015A	—	_	ICSIDL	_	_	_	—	_	ICTMR	ICI<1:0>	ICOV	ICBNE	10	CM<2:0>
IC8BUF	015C							Input	t 8 Captur	e Register	r				
IC8CON	015E	_	_	ICSIDL	_		_	_	_	ICTMR	ICI<1:0>	ICOV	ICBNE	10	CM<2:0>
Lonondi		initialize	dhit												

**Legend:** u = uninitialized bit

Note: Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

# dsPIC30F6010

NOTES:

## 13.0 OUTPUT COMPARE MODULE

**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).

This section describes the Output Compare module and associated operational modes. The features provided by this module are useful in applications requiring operational modes such as:

- Generation of Variable Width Output Pulses
- Power Factor Correction

Figure 13-1 depicts a block diagram of the Output Compare module.

The key operational features of the Output Compare module include:

- Timer2 and Timer3 Selection mode
- Simple Output Compare Match mode
- Dual Output Compare Match mode
- Simple PWM mode
- · Output Compare during Sleep and Idle modes
- Interrupt on Output Compare/PWM Event

These operating modes are determined by setting the appropriate bits in the 16-bit OCxCON SFR (where x = 1,2,3,...,N). The dsPIC30F6010 device has 8 compare channels.

OCxRS and OCxR in the figure represent the Dual Compare registers. In the Dual Compare mode, the OCxR register is used for the first compare and OCxRS is used for the second compare.



#### FIGURE 13-1: OUTPUT COMPARE MODE BLOCK DIAGRAM

## 13.1 Timer2 and Timer3 Selection Mode

Each output compare channel can select between one of two 16-bit timers; Timer2 or Timer3.

The selection of the timers is controlled by the OCTSEL bit (OCxCON<3>). Timer2 is the default timer resource for the Output Compare module.

### 13.2 Simple Output Compare Match Mode

When control bits OCM<2:0> (OCxCON<2:0>) = 001, 010 or 011, the selected output compare channel is configured for one of three simple output compare match modes:

- Compare forces I/O pin low
- Compare forces I/O pin high
- Compare toggles I/O pin

The OCxR register is used in these modes. The OCxR register is loaded with a value and is compared to the selected incrementing timer count. When a compare occurs, one of these compare match modes occurs. If the counter resets to zero before reaching the value in OCxR, the state of the OCx pin remains unchanged.

## 13.3 Dual Output Compare Match Mode

When control bits OCM<2:0> (OCxCON<2:0>) = 100or 101, the selected output compare channel is configured for one of two dual output compare modes, which are:

- Single Output Pulse mode
- Continuous Output Pulse mode

#### 13.3.1 SINGLE PULSE MODE

For the user to configure the module for the generation of a single output pulse, the following steps are required (assuming timer is off):

- Determine instruction cycle time Tcy.
- Calculate desired pulse width value based on TCY.
- Calculate time to start pulse from timer start value of 0x0000.
- Write pulse width start and stop times into OCxR and OCxRS compare registers (x denotes channel 1, 2, ...,N).
- Set timer period register to value equal to, or greater than, value in OCxRS compare register.
- Set OCM<2:0> = 100.
- Enable timer, TON (TxCON<15>) = 1.

To initiate another single pulse, issue another write to set OCM<2:0> = 100.

#### 13.3.2 CONTINUOUS PULSE MODE

For the user to configure the module for the generation of a continuous stream of output pulses, the following steps are required:

- Determine instruction cycle time Tcy.
- Calculate desired pulse value based on Tcy.
- Calculate timer to start pulse width from timer start value of 0x0000.
- Write pulse width start and stop times into OCxR and OCxRS (x denotes channel 1, 2, ...,N) compare registers, respectively.
- Set timer period register to value equal to, or greater than, value in OCxRS compare register.
- Set OCM<2:0> = 101.
- Enable timer, TON (TxCON<15>) = 1.

## 13.4 Simple PWM Mode

When control bits OCM<2:0> (OCxCON<2:0>) = 110 or 111, the selected output compare channel is configured for the PWM mode of operation. When configured for the PWM mode of operation, OCxR is the Main latch (read only) and OCxRS is the Secondary latch. This enables glitchless PWM transitions.

The user must perform the following steps in order to configure the output compare module for PWM operation:

- 1. Set the PWM period by writing to the appropriate period register.
- 2. Set the PWM duty cycle by writing to the OCxRS register.
- 3. Configure the output compare module for PWM operation.
- 4. Set the TMRx prescale value and enable the Timer, TON (TxCON<15>) = 1.

#### 13.4.1 INPUT PIN FAULT PROTECTION FOR PWM

When control bits OCM<2:0> (OCxCON<2:0>) = 111, the selected output compare channel is again configured for the PWM mode of operation, with the additional feature of input fault protection. While in this mode, if a logic '0' is detected on the OCFA/B pin, the respective PWM output pin is placed in the high impedance input state. The OCFLT bit (OCxCON<4>) indicates whether a Fault condition has occurred. This state will be maintained until both of the following events have occurred:

- The external Fault condition has been removed.
- The PWM mode has been re-enabled by writing to the appropriate control bits.

#### 13.4.2 PWM PERIOD

The PWM period is specified by writing to the PRx register. The PWM period can be calculated using Equation 13-1.

#### EQUATION 13-1: PWM PERIOD

 $PWM period = [(PRx) + 1] \cdot 4 \cdot TOSC \cdot (TMRx prescale value)$ 

PWM frequency is defined as 1/[PWM period].

When the selected TMRx is equal to its respective period register, PRx, the following four events occur on the next increment cycle:

- TMRx is cleared.
- The OCx pin is set.
  - Exception 1: If PWM duty cycle is 0x0000, the OCx pin will remain low.
  - Exception 2: If duty cycle is greater than PRx, the pin will remain high.
- The PWM duty cycle is latched from OCxRS into OCxR.
- The corresponding timer interrupt flag is set.

See Figure 13-1 for key PWM period comparisons. Timer3 is referred to in the figure for clarity.



## 13.5 Output Compare Operation During CPU Sleep Mode

When the CPU enters the Sleep mode, all internal clocks are stopped. Therefore, when the CPU enters the Sleep state, the output compare channel will drive the pin to the active state that was observed prior to entering the CPU Sleep state.

For example, if the pin was high when the CPU entered the Sleep state, the pin will remain high. Likewise, if the pin was low when the CPU entered the Sleep state, the pin will remain low. In either case, the output compare module will resume operation when the device wakes up.

## 13.6 Output Compare Operation During CPU Idle Mode

When the CPU enters the Idle mode, the output compare module can operate with full functionality.

The output compare channel will operate during the CPU Idle mode if the OCSIDL bit (OCxCON<13>) is at logic 0 and the selected time base (Timer2 or Timer3) is enabled and the TSIDL bit of the selected timer is set to logic 0.

## 13.7 Output Compare Interrupts

The output compare channels have the ability to generate an interrupt on a compare match, for whichever match mode has been selected.

For all modes except the PWM mode, when a compare event occurs, the respective interrupt flag (OCxIF) is asserted and an interrupt will be generated, if enabled. The OCxIF bit is located in the corresponding IFS Status register, and must be cleared in software. The interrupt is enabled via the respective compare interrupt enable (OCxIE) bit, located in the corresponding IEC Control register.

For the PWM mode, when an event occurs, the respective timer interrupt flag (T2IF or T3IF) is asserted and an interrupt will be generated, if enabled. The IF bit is located in the IFS0 Status register, and must be cleared in software. The interrupt is enabled via the respective Timer Interrupt Enable bit (T2IE or T3IE), located in the IEC0 Control register. The output compare interrupt flag is never set during the PWM mode of operation.

## TABLE 13-1: OUTPUT COMPARE REGISTER MAP

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bi
OC1RS	0180							Outp	ut Compar	e 1 Secon	idary Regi	ster					
OC1R	0182		_	-	-	-	-	0	utput Com	pare 1 Ma	in Registe	r	-	-	-		
OC1CON	0184	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL		OCM<2:0:	>
OC2RS	0186							Outp	ut Compar	e 2 Secon	idary Regi	ster					
OC2R	0188							0	utput Com	pare 2 Ma	in Registe	r					
OC2CON	018A	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSE		OCM<2:0:	>
OC3RS	018C							Outp	ut Compar	e 3 Secon	idary Regi	ster					
OC3R	018E			-	-			0	utput Com	pare 3 Ma	in Registe	r	-		-		
OC3CON	0190	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL		OCM<2:0:	>
OC4RS	0192							Outp	ut Compar	e 4 Secon	idary Regi	ster					
OC4R	0194							0	utput Com	pare 4 Ma	in Registe	r					
OC4CON	0196	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL		OCM<2:0:	>
OC5RS	0198							Outp	ut Compar	e 5 Secon	idary Regi	ster					
OC5R	019A							0	utput Com	pare 5 Ma	in Registe	r					
OC5CON	019C	—	—	OCSIDL	_	—	—	_	—	_	_	_	OCFLT	OCTSEL		OCM<2:0:	>
OC6RS	019E							Outp	ut Compar	e 6 Secon	idary Regi	ster					
OC6R	01A0		_	-	-	_		0	utput Com	pare 6 Ma	in Registe	r	_	_	-		
OC6CON	01A2	_	_	OCSIDL	_	—	_		_		_	_	OCFLT	OCTSEL		OCM<2:0:	>
OC7RS	01A4							Outp	ut Compar	e 7 Secon	idary Regi	ster					
OC7R	01A6		-	-		_	-	0	utput Com	pare 7 Ma	in Registe	r	-				
OC7CON	01A8	_	—	OCSIDL	—	_	—	I	—		—	—	OCFLT	OCTSEL		OCM<2:0;	>
OC8RS	01AA		-					Outp	ut Compar	e 8 Secon	idary Regi	ster					
OC8R	01AC							0	utput Com	pare 8 Ma	in Registe	r					
OC8CON	01AE	_	_	OCSIDL	_	_	_		_	_	_	_	OCFLT	OCTSEL		OCM<2:0:	>
Leavende		in it ali-			-												

**Legend:** u = uninitialized bit

Note: Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

## 14.0 QUADRATURE ENCODER INTERFACE (QEI) MODULE

**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).

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

- Three input channels for two phase signals and index pulse
- 16-bit up/down position counter
- Count direction status
- Position Measurement (x2 and x4) mode
- Programmable digital noise filters on inputs
- Alternate 16-bit Timer/Counter mode
- Quadrature Encoder Interface interrupts

These operating modes are determined by setting the appropriate bits QEIM<2:0> (QEICON<10:8>). Figure 14-1 depicts the Quadrature Encoder Interface block diagram.





## 14.1 Quadrature Encoder Interface Logic

A typical incremental (a.k.a. optical) encoder has three outputs: Phase A, Phase B, and an index pulse. These signals are useful and often required in position and speed control of ACIM and SR motors.

The two channels, Phase A (QEA) and Phase B (QEB), have a unique relationship. If Phase A leads Phase B, then the direction (of the motor) is deemed positive or forward. If Phase A lags Phase B, then the direction (of the motor) is deemed negative or reverse.

A third channel, termed index pulse, occurs once per revolution and is used as a reference to establish an absolute position. The index pulse coincides with Phase A and Phase B, both low.

### 14.2 16-bit Up/Down Position Counter Mode

The 16-bit Up/Down Counter counts up or down on every count pulse, which is generated by the difference of the Phase A and Phase B input signals. The counter acts as an integrator, whose count value is proportional to position. The direction of the count is determined by the UPDN signal, which is generated by the Quadrature Encoder Interface Logic.

#### 14.2.1 POSITION COUNTER ERROR CHECKING

Position count error checking in the QEI is provided for and indicated by the CNTERR bit (QEICON<15>). The error checking only applies when the position counter is configured for Reset on the Index Pulse modes (QEIM < 2:0 > = '110' or '100'). In these modes, the contents of the POSCNT register is compared with the values (0xFFFF or MAXCNT+1, depending on direction). If these values are detected, an error condition is generated by setting the CNTERR bit and a QEI count error interrupt is generated. The QEI count error interrupt can be disabled by setting the CEID bit (DFLTCON<8>). The position counter continues to count encoder edges after an error has been detected. The POSCNT register continues to count up/down until a natural rollover/underflow. No interrupt is generated for the natural rollover/underflow event. The CNTERR bit is a Read/Write bit and reset in software by the user.

## 14.2.2 POSITION COUNTER RESET

The Position Counter Reset Enable bit, POSRES (QEICON<2>) controls whether the position counter is reset when the index pulse is detected. This bit is only applicable when QEIM<2:0> = 100' or 110'.

If the POSRES bit is set to '1', then the position counter is reset when the index pulse is detected. If the POSRES bit is set to '0', then the position counter is not reset when the index pulse is detected. The position counter will continue counting up or down, and will be reset on the rollover or underflow condition.

The interrupt is still generated on the detection of the index pulse and not on the position counter overflow/ underflow.

### 14.2.3 COUNT DIRECTION STATUS

As mentioned in the previous section, the QEI logic generates an UPDN signal, based upon the relationship between Phase A and Phase B. In addition to the output pin, the state of this internal UPDN signal is supplied to a SFR bit UPDN (QEICON<11>) as a read only bit. To place the state of this signal on an I/O pin, the SFR bit PCDOUT (QEICON<6>) must be '1'.

## 14.3 Position Measurement Mode

There are two measurement modes which are supported and are termed x2 and x4. These modes are selected by the QEIM<2:0> mode select bits located in SFR QEICON<10:8>.

When control bits QEIM<2:0> = 100 or 101, the x2 Measurement mode is selected and the QEI logic only looks at the Phase A input for the position counter increment rate. Every rising and falling edge of the Phase A signal causes the position counter to be incremented or decremented. The Phase B signal is still utilized for the determination of the counter direction, just as in the x4 mode.

Within the x2 Measurement mode, there are two variations of how the position counter is Reset:

- 1. Position counter reset by detection of index pulse, QEIM<2:0> = 100.
- Position counter reset by match with MAXCNT, QEIM<2:0> = 101.

When control bits QEIM<2:0> = 110 or 111, the x4 Measurement mode is selected and the QEI logic looks at both edges of the Phase A and Phase B input signals. Every edge of both signals causes the position counter to increment or decrement.

Within the x4 Measurement mode, there are two variations of how the position counter is reset:

- Position counter reset by detection of index pulse, QEIM<2:0> = 110.
- Position counter reset by match with MAXCNT, QEIM<2:0> = 111.

The x4 Measurement mode provides for finer resolution data (more position counts) for determining motor position.

#### 14.4 Programmable Digital Noise Filters

The digital noise filter section is responsible for rejecting noise on the incoming capture or quadrature signals. Schmitt Trigger inputs and a three-clock cycle delay filter combine to reject low level noise and large, short duration noise spikes that typically occur in noise prone applications, such as a motor system.

The filter ensures that the filtered output signal is not permitted to change until a stable value has been registered for three consecutive clock cycles.

For the QEA, QEB and INDX pins, the clock divide frequency for the digital filter is programmed by bits QECK<2:0> (DFLTCON<6:4>) and are derived from the base instruction cycle Tcy.

To enable the filter output for channels QEA, QEB and INDX, the QEOUT bit must be '1'. The filter network for all channels is disabled on POR and BOR.

## 14.5 Alternate 16-bit Timer/Counter

When the QEI module is not configured for the QEI mode QEIM<2:0> = 001, the module can be configured as a simple 16-bit timer/counter. The setup and control of the auxiliary timer is accomplished through the QEICON SFR register. This timer functions identically to Timer1. The QEA pin is used as the timer clock input.

When configured as a timer, the POSCNT register serves as the Timer Count Register and the MAXCNT register serves as the Period Register. When a timer/ period register match occur, the QEI interrupt flag will be asserted.

The only exception between the general purpose timers and this timer is the added feature of external Up/ Down input select. When the UPDN pin is asserted high, the timer will increment up. When the UPDN pin is asserted low, the timer will be decremented.

Note:	Changing the operational mode (i.e., from
	QEI to Timer or vice versa), will not affect
	the Timer/Position Count Register
	contents.

The UPDN Control/Status bit (QEICON<11>) can be used to select the count direction state of the Timer register. When UPDN = 1, the timer will count up. When UPDN = 0, the timer will count down.

In addition, control bit UPDN\_SRC (QEICON<0>) determines whether the timer count direction state is based on the logic state, written into the UPDN Control/Status bit (QEICON<11>), or the QEB pin state. When UPDN\_SRC = 1, the timer count direction is controlled from the QEB pin. Likewise, when UPDN\_SRC = 0, the timer count direction is controlled by the UPDN bit.

Note: This Timer does not support the External Asynchronous Counter mode of operation. If using an external clock source, the clock will automatically be synchronized to the internal instruction cycle.

### 14.6 QEI Module Operation During CPU Sleep Mode

#### 14.6.1 QEI OPERATION DURING CPU SLEEP MODE

The QEI module will be halted during the CPU Sleep mode.

#### 14.6.2 TIMER OPERATION DURING CPU SLEEP MODE

During CPU Sleep mode, the timer will not operate, because the internal clocks are disabled.

### 14.7 QEI Module Operation During CPU Idle Mode

Since the QEI module can function as a quadrature encoder interface, or as a 16-bit timer, the following section describes operation of the module in both modes.

#### 14.7.1 QEI OPERATION DURING CPU IDLE MODE

When the CPU is placed in the Idle mode, the QEI module will operate if the QEISIDL bit (QEICON<13>) = 0. This bit defaults to a logic '0' upon executing POR and BOR. For halting the QEI module during the CPU Idle mode, QEISIDL should be set to '1'.

#### 14.7.2 TIMER OPERATION DURING CPU IDLE MODE

When the CPU is placed in the Idle mode and the QEI module is configured in the 16-bit Timer mode, the 16-bit timer will operate if the QEISIDL bit (QEICON<13>) = 0. This bit defaults to a logic '0' upon executing POR and BOR. For halting the timer module during the CPU Idle mode, QEISIDL should be set to '1'.

If the QEISIDL bit is cleared, the timer will function normally, as if the CPU Idle mode had not been entered.

### 14.8 Quadrature Encoder Interface Interrupts

The quadrature encoder interface has the ability to generate an interrupt on occurrence of the following events:

- Interrupt on 16-bit up/down position counter rollover/underflow
- Detection of qualified index pulse, or if CNTERR bit is set
- Timer period match event (overflow/underflow)
- · Gate accumulation event

The QEI Interrupt Flag bit, QEIIF, is asserted upon occurrence of any of the above events. The QEIIF bit must be cleared in software. QEIIF is located in the IFS2 Status register.

Enabling an interrupt is accomplished via the respective Enable bit, QEIIE. The QEIIE bit is located in the IEC2 Control register.

## TABLE 14-1: QEI REGISTER MAP

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit
QEICON	0122	CNTERR	_	QEISIDL	INDX	UPDN	QEIM2	QEIM1	QEIM0	SWPAB	PCDOUT	TQGATE	TQCKPS1	TQCKPS0	POSRES	TQCS	UPDN
DFLTCON	0124	_	_	_	_	_	IMV1	IMV0	CEID	QEOUT	QECK2	QECK1	QECK0	_	_		-
POSCNT	0126								Po	osition Co	unter<15:0>	•					
MAXCNT	0128		Maximun Count<15:0>														

**Legend:** u = uninitialized bit

**Note:** Refer to "*dsPIC30F Family Reference Manual*" (DS70046) for descriptions of register bit fields.

# dsPIC30F6010

NOTES:

## 15.0 MOTOR CONTROL PWM MODULE

**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).

This module simplifies the task of generating multiple, synchronized Pulse Width Modulated (PWM) outputs. In particular, the following power and motion control applications are supported by the PWM module:

- Three Phase AC Induction Motor
- Switched Reluctance (SR) Motor
- Brushless DC (BLDC) Motor
- Uninterruptible Power Supply (UPS)

The PWM module has the following features:

- 8 PWM I/O pins with 4 duty cycle generators
- Up to 16-bit resolution
- 'On-the-Fly' PWM frequency changes
- Edge and Center-Aligned Output modes
- Single Pulse Generation mode
- Interrupt support for asymmetrical updates in Center-Aligned mode
- Output override control for Electrically Commutative Motor (ECM) operation
- 'Special Event' comparator for scheduling other peripheral events
- Fault pins to optionally drive each of the PWM output pins to a defined state

This module contains 4 duty cycle generators, numbered 1 through 4. The module has 8 PWM output pins, numbered PWM1H/PWM1L through PWM4H/PWM4L. The eight I/O pins are grouped into high/low numbered pairs, denoted by the suffix H or L, respectively. For complementary loads, the low PWM pins are always the complement of the corresponding high I/O pin.

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

# dsPIC30F6010



### 15.1 PWM Time Base

The PWM time base is provided by a 15-bit timer with a prescaler and postscaler. The time base is accessible via the PTMR SFR. PTMR<15> is a Read Only Status bit, PTDIR, that indicates the present count direction of the PWM time base. If PTDIR is cleared, PTMR is counting upwards. If PTDIR is set, PTMR is counting downwards. The PWM time base is configured via the PTCON SFR. The time base is enabled/disabled by setting/clearing the PTEN bit in the PTCON SFR. PTMR is not cleared when the PTEN bit is cleared in software.

The PTPER SFR sets the counting period for PTMR. The user must write a 15-bit value to PTPER<14:0>. When the value in PTMR<14:0> matches the value in PTPER<14:0>, the time base will either Reset to '0', or reverse the count direction on the next occurring clock cycle. The action taken depends on the operating mode of the time base.

**Note:** If the period register is set to 0x0000, the timer will stop counting, and the interrupt and the special event trigger will not be generated, even if the special event value is also 0x0000. The module will not update the period register, if it is already at 0x0000; therefore, the user must disable the module in order to update the period register.

The PWM time base can be configured for four different modes of operation:

- Free Running mode
- Single Shot mode
- Continuous Up/Down Count mode
- Continuous Up/Down Count mode with interrupts for double updates

These four modes are selected by the PTMOD<1:0> bits in the PTCON SFR. The Up/Down Counting modes support center-aligned PWM generation. The Single Shot mode allows the PWM module to support pulse control of certain Electronically Commutative Motors (ECMs).

The interrupt signals generated by the PWM time base depend on the mode selection bits (PTMOD<1:0>) and the postscaler bits (PTOPS<3:0>) in the PTCON SFR.

#### 15.1.1 FREE RUNNING MODE

In the Free Running mode, the PWM time base counts upwards until the value in the Time Base Period register (PTPER) is matched. The PTMR register is reset on the following input clock edge and the time base will continue to count upwards as long as the PTEN bit remains set.

When the PWM time base is in the Free Running mode (PTMOD<1:0> = 00), an interrupt event is generated each time a match with the PTPER register occurs and the PTMR register is reset to zero. The postscaler selection bits may be used in this mode of the timer to reduce the frequency of the interrupt events.

#### 15.1.2 SINGLE SHOT MODE

In the Single Shot Counting mode, the PWM time base begins counting upwards when the PTEN bit is set. When the value in the PTMR register matches the PTPER register, the PTMR register will be reset on the following input clock edge and the PTEN bit will be cleared by the hardware to halt the time base.

When the PWM time base is in the Single Shot mode (PTMOD<1:0> = 01), an interrupt event is generated when a match with the PTPER register occurs, the PTMR register is reset to zero on the following input clock edge, and the PTEN bit is cleared. The postscaler selection bits have no effect in this mode of the timer.

#### 15.1.3 CONTINUOUS UP/DOWN COUNTING MODES

In the Continuous Up/Down Counting modes, the PWM time base counts upwards until the value in the PTPER register is matched. The timer will begin counting downwards on the following input clock edge. The PTDIR bit in the PTMR SFR is read only and indicates the counting direction The PTDIR bit is set when the timer counts downwards.

In the Up/Down Counting mode (PTMOD<1:0> = 10), an interrupt event is generated each time the value of the PTMR register becomes zero and the PWM time base begins to count upwards. The postscaler selection bits may be used in this mode of the timer to reduce the frequency of the interrupt events.

### 15.1.4 DOUBLE UPDATE MODE

In the Double Update mode (PTMOD<1:0> = 11), an interrupt event is generated each time the PTMR register is equal to zero, as well as each time a period match occurs. The postscaler selection bits have no effect in this mode of the timer.

The Double Update mode provides two additional functions to the user. First, the control loop bandwidth is doubled because the PWM duty cycles can be updated, twice per period. Second, asymmetrical center-aligned PWM waveforms can be generated, which are useful for minimizing output waveform distortion in certain motor control applications.

Note:	Programming a value of 0x0001 in the
	period register could generate a continu-
	ous interrupt pulse, and hence, must be
	avoided.

#### 15.1.5 PWM TIME BASE PRESCALER

The input clock to PTMR (Fosc/4), has prescaler options of 1:1, 1:4, 1:16, or 1:64, selected by control bits PTCKPS<1:0> in the PTCON SFR. The prescaler counter is cleared when any of the following occurs:

- a write to the PTMR register
- a write to the PTCON register
- · any device Reset

The PTMR register is not cleared when PTCON is written.

#### 15.1.6 PWM TIME BASE POSTSCALER

The match output of PTMR can optionally be postscaled through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling).

The postscaler counter is cleared when any of the following occurs:

- a write to the PTMR register
- a write to the PTCON register
- any device Reset

The PTMR register is not cleared when PTCON is written.

#### 15.2 PWM Period

PTPER is a 15-bit register and is used to set the counting period for the PWM time base. PTPER is a doublebuffered register. The PTPER buffer contents are loaded into the PTPER register at the following instants:

- Free Running and Single Shot modes: When the PTMR register is reset to zero after a match with the PTPER register.
- <u>Up/Down Counting modes</u>: When the PTMR register is zero.

The value held in the PTPER buffer is automatically loaded into the PTPER register when the PWM time base is disabled (PTEN = 0).

The PWM period can be determined using Equation 15-1:

#### EQUATION 15-1: PWM PERIOD

 $TPWM = \frac{Tcy \bullet (PTPER + 1)}{(PTMR Prescale Value)}$ 

If the PWM time base is configured for one of the Up/ Down Count modes, the PWM period will be twice the value provided by Equation 15-2.

## EQUATION 15-2: PWM PERIOD (UP/DOWN MODE)

 $TPWM = \frac{2 \cdot Tcy \cdot (PTPER + 0.75)}{(PTMR Prescale Value)}$ 

The maximum resolution (in bits) for a given device oscillator and PWM frequency can be determined using Equation 15-3:

#### EQUATION 15-3: PWM RESOLUTION

Resolution =  $\frac{\log (2 \cdot \text{Tpwm} / \text{Tcy})}{\log (2)}$ 

## 15.3 Edge-Aligned PWM

Edge-aligned PWM signals are produced by the module when the PWM time base is in the Free Running or Single Shot mode. For edge-aligned PWM outputs, the output has a period specified by the value in PTPER and a duty cycle specified by the appropriate duty cycle register (see Figure 15-2). The PWM output is driven active at the beginning of the period (PTMR = 0) and is driven inactive when the value in the duty cycle register matches PTMR.

If the value in a particular duty cycle register is zero, then the output on the corresponding PWM pin will be inactive for the entire PWM period. In addition, the output on the PWM pin will be active for the entire PWM period if the value in the duty cycle register is greater than the value held in the PTPER register.



#### 15.4 Center-Aligned PWM

Center-aligned PWM signals are produced by the module when the PWM time base is configured in an Up/Down Counting mode (see Figure 15-3).

The PWM compare output is driven to the active state when the value of the duty cycle register matches the value of PTMR and the PWM time base is counting downwards (PTDIR = 1). The PWM compare output is driven to the inactive state when the PWM time base is counting upwards (PTDIR = 0) and the value in the PTMR register matches the duty cycle value.

If the value in a particular duty cycle register is zero, then the output on the corresponding PWM pin will be inactive for the entire PWM period. In addition, the output on the PWM pin will be active for the entire PWM period if the value in the duty cycle register is equal to the value held in the PTPER register.



#### **FIGURE 15-3: CENTER-ALIGNED PWM**

#### 15.5 **PWM Duty Cycle Comparison** Units

There are four 16-bit special function registers (PDC1, PDC2, PDC3 and PDC4) used to specify duty cycle values for the PWM module.

The value in each duty cycle register determines the amount of time that the PWM output is in the active state. The duty cycle registers are 16 bits wide. The LSb of a duty cycle register determines whether the PWM edge occurs in the beginning. Thus, the PWM resolution is effectively doubled.

#### 15.5.1 DUTY CYCLE REGISTER BUFFERS

The four PWM duty cycle registers are double-buffered to allow glitchless updates of the PWM outputs. For each duty cycle, there is a duty cycle register that is accessible by the user and a second duty cycle register that holds the actual compare value used in the present PWM period.

For edge-aligned PWM output, a new duty cycle value will be updated whenever a match with the PTPER register occurs and PTMR is reset. The contents of the duty cycle buffers are automatically loaded into the duty cycle registers when the PWM time base is disabled (PTEN = 0) and the UDIS bit is cleared in PWMCON2.

When the PWM time base is in the Up/Down Counting mode, new duty cycle values are updated when the value of the PTMR register is zero and the PWM time base begins to count upwards. The contents of the duty cycle buffers are automatically loaded into the duty cycle registers when the PWM time base is disabled (PTEN = 0).

When the PWM time base is in the Up/Down Counting mode with double updates, new duty cycle values are updated when the value of the PTMR register is zero, and when the value of the PTMR register matches the value in the PTPER register. The contents of the duty cycle buffers are automatically loaded into the duty cycle registers when the PWM time base is disabled (PTEN = 0).

### 15.6 Complementary PWM Operation

In the Complementary mode of operation, each pair of PWM outputs is obtained by a complementary PWM signal. A dead time may be optionally inserted during device switching, when both outputs are inactive for a short period (Refer to **Section 15.7** "**Dead-Time Generators**").

In Complementary mode, the duty cycle comparison units are assigned to the PWM outputs as follows:

- PDC1 register controls PWM1H/PWM1L outputs
- PDC2 register controls PWM2H/PWM2L outputs
- PDC3 register controls PWM3H/PWM3L outputs
- PDC4 register controls PWM4H/PWM4L outputs

The Complementary mode is selected for each PWM I/O pin pair by clearing the appropriate PMODx bit in the PWMCON1 SFR. The PWM I/O pins are set to Complementary mode by default upon a device Reset.

#### 15.7 Dead-Time Generators

Dead-time generation may be provided when any of the PWM I/O pin pairs are operating in the Complementary Output mode. The PWM outputs use Push-Pull drive circuits. Due to the inability of the power output devices to switch instantaneously, some amount of time must be provided between the turn off event of one PWM output in a complementary pair and the turn on event of the other transistor.

The PWM module allows two different dead times to be programmed. These two dead times may be used in one of two methods described below to increase user flexibility:

- The PWM output signals can be optimized for different turn off times in the high side and low side transistors in a complementary pair of transistors. The first dead time is inserted between the turn off event of the lower transistor of the complementary pair and the turn on event of the upper transistor. The second dead time is inserted between the turn off event of the upper transistor and the turn on event of the upper transistor and the turn on event of the lower transistor.
- The two dead times can be assigned to individual PWM I/O pin pairs. This operating mode allows the PWM module to drive different transistor/load combinations with each complementary PWM I/O pin pair.

#### 15.7.1 DEAD-TIME GENERATORS

Each complementary output pair for the PWM module has a 6-bit down counter that is used to produce the dead-time insertion. As shown in Figure 15-4, each dead-time unit has a rising and falling edge detector connected to the duty cycle comparison output.

#### 15.7.2 DEAD-TIME ASSIGNMENT

The DTCON2 SFR contains control bits that allow the dead times to be assigned to each of the complementary outputs. Table 15-1 summarizes the function of each dead-time selection control bit.

TABLE 15-1: DEA	D-TIME SEL	ECTION BITS
-----------------	------------	-------------

Bit	Function
DTS1A	Selects PWM1L/PWM1H active edge dead time.
DTS1I	Selects PWM1L/PWM1H inactive edge dead time.
DTS2A	Selects PWM2L/PWM2H active edge dead time.
DTS2I	Selects PWM2L/PWM2H inactive edge dead time.
DTS3A	Selects PWM3L/PWM3H active edge dead time.
DTS3I	Selects PWM3L/PWM3H inactive edge dead time.
DTS4A	Selects PWM4L/PWM4H active edge dead time.
DTS4I	Selects PWM4L/PWM4H inactive edge dead time.

### 15.7.3 DEAD-TIME RANGES

The amount of dead time provided by each dead-time unit is selected by specifying the input clock prescaler value and a 6-bit unsigned value. The amount of dead time provided by each unit may be set independently.

Four input clock prescaler selections have been provided to allow a suitable range of dead times, based on the device operating frequency. The clock prescaler option may be selected independently for each of the two dead-time values. The dead-time clock prescaler values are selected using the DTAPS<1:0> and DTBPS<1:0> control bits in the DTCON1 SFR. One of four clock prescaler options (TcY, 2TcY, 4TcY or 8TcY) may be selected for each of the dead-time values.

After the prescaler values are selected, the dead time for each unit is adjusted by loading two 6-bit unsigned values into the DTCON1 SFR.

The dead-time unit prescalers are cleared on the following events:

- On a load of the down timer due to a duty cycle comparison edge event.
- On a write to the DTCON1 or DTCON2 registers.
- On any device Reset.

Note: The user should not modify the DTCON1 or DTCON2 values while the PWM module is operating (PTEN = 1). Unexpected results may occur.

FIGURE 15-4: DEAD-TIME TIMING DIAGRAM



## 15.8 Independent PWM Output

An independent PWM Output mode is required for driving certain types of loads. A particular PWM output pair is in the Independent Output mode when the corresponding PMOD bit in the PWMCON1 register is set. No dead-time control is implemented between adjacent PWM I/O pins when the module is operating in the Independent mode and both I/O pins are allowed to be active simultaneously.

In the Independent mode, each duty cycle generator is connected to both of the PWM I/O pins in an output pair. By using the associated duty cycle register and the appropriate bits in the OVDCON register, the user may select the following signal output options for each PWM I/O pin operating in the Independent mode:

- I/O pin outputs PWM signal
- I/O pin inactive
- I/O pin active

## 15.9 Single Pulse PWM Operation

The PWM module produces single pulse outputs when the PTCON control bits PTMOD<1:0> = 10. Only edgealigned outputs may be produced in the Single Pulse mode. In Single Pulse mode, the PWM I/O pin(s) are driven to the active state when the PTEN bit is set. When a match with a duty cycle register occurs, the PWM I/O pin is driven to the inactive state. When a match with the PTPER register occurs, the PTMR register is cleared, all active PWM I/O pins are driven to the inactive state, the PTEN bit is cleared, and an interrupt is generated.

## 15.10 PWM Output Override

The PWM output override bits allow the user to manually drive the PWM I/O pins to specified logic states, independent of the duty cycle comparison units.

All control bits associated with the PWM output override function are contained in the OVDCON register. The upper half of the OVDCON register contains eight bits, POVDxH<4:1> and POVDxL<4:1>, that determine which PWM I/O pins will be overridden. The lower half of the OVDCON register contains eight bits, POUTxH<4:1> and POUTxL<4:1>, that determine the state of the PWM I/O pins when a particular output is overridden via the POVD bits.

#### 15.10.1 COMPLEMENTARY OUTPUT MODE

When a PWMxL pin is driven active via the OVDCON register, the output signal is forced to be the complement of the corresponding PWMxH pin in the pair. Dead-time insertion is still performed when PWM channels are overridden manually.

## 15.10.2 OVERRIDE SYNCHRONIZATION

If the OSYNC bit in the PWMCON2 register is set, all output overrides performed via the OVDCON register are synchronized to the PWM time base. Synchronous output overrides occur at the following times:

- Edge-Aligned mode, when PTMR is zero.
- Center-Aligned modes, when PTMR is zero and when the value of PTMR matches PTPER.

## 15.11 PWM Output and Polarity Control

There are three device Configuration bits associated with the PWM module that provide PWM output pin control:

- HPOL Configuration bit
- LPOL Configuration bit
- PWMPIN Configuration bit

These three bits in the FPORBOR configuration register (see Section 21) work in conjunction with the four PWM Enable bits (PWMEN<4:1>) located in the PWMCON1 SFR. The Configuration bits and PWM Enable bits ensure that the PWM pins are in the correct states after a device Reset occurs. The PWMPIN configuration fuse allows the PWM module outputs to be optionally enabled on a device Reset. If PWMPIN = 0, the PWM outputs will be driven to their inactive states at Reset. If PWMPIN = 1 (default), the PWM outputs will be tri-stated. The HPOL bit specifies the polarity for the PWMxH outputs, whereas the LPOL bit specifies the polarity for the PWMxL outputs.

#### 15.11.1 OUTPUT PIN CONTROL

The PEN<4:1>H and PEN<4:1>L control bits in the PWMCON1 SFR enable each high PWM output pin and each low PWM output pin, respectively. If a particular PWM output pin not enabled, it is treated as a general purpose I/O pin.

#### 15.12 PWM Fault Pins

There are two Fault pins (FLTA and FLTB) associated with the PWM module. When asserted, these pins can optionally drive each of the PWM I/O pins to a defined state.

#### 15.12.1 FAULT PIN ENABLE BITS

The FLTACON and FLTBCON SFRs each have 4 control bits that determine whether a particular pair of PWM I/O pins is to be controlled by the Fault input pin. To enable a specific PWM I/O pin pair for Fault overrides, the corresponding bit should be set in the FLTA-CON or FLTBCON register.

If all enable bits are cleared in the FLTACON or FLTBCON registers, then the corresponding Fault input pin has no effect on the PWM module and the pin may be used as a general purpose interrupt or I/O pin.

Note: The Fault pin logic can operate independent of the PWM logic. If all the enable bits in the FLTACON/FLTBCON register are cleared, then the Fault pin(s) could be used as general purpose interrupt pin(s). Each Fault pin has an interrupt vector, Interrupt Flag bit and Interrupt Priority bits associated with it.

#### 15.12.2 FAULT STATES

The FLTACON and FLTBCON special function registers have 8 bits each that determine the state of each PWM I/O pin when it is overridden by a Fault input. When these bits are cleared, the PWM I/O pin is driven to the inactive state. If the bit is set, the PWM I/O pin will be driven to the active state. The active and inactive states are referenced to the polarity defined for each PWM I/O pin (HPOL and LPOL polarity control bits).

A special case exists when a PWM module I/O pair is in the Complementary mode and both pins are programmed to be active on a Fault condition. The PWMxH pin always has priority in the Complementary mode, so that both I/O pins cannot be driven active simultaneously.

#### 15.12.3 FAULT PIN PRIORITY

If both Fault input pins have been assigned to control a particular PWM I/O pin, the Fault state programmed for the Fault A input pin will take priority over the Fault B input pin.

#### 15.12.4 FAULT INPUT MODES

Each of the Fault input pins has two modes of operation:

- Latched Mode: When the Fault pin is driven low, the PWM outputs will go to the states defined in the FLTACON/FLTBCON register. The PWM outputs will remain in this state until the Fault pin is driven high and the corresponding interrupt flag has been cleared in software. When both of these actions have occurred, the PWM outputs will return to normal operation at the beginning of the next PWM cycle or half-cycle boundary. If the interrupt flag is cleared before the Fault condition ends, the PWM module will wait until the Fault pin is no longer asserted, to restore the outputs.
- Cycle-by-Cycle Mode: When the Fault input pin is driven low, the PWM outputs remain in the defined Fault states for as long as the Fault pin is held low. After the Fault pin is driven high, the PWM outputs return to normal operation at the beginning of the following PWM cycle or half-cycle boundary.

The operating mode for each Fault input pin is selected using the FLTAM and FLTBM control bits in the FLTACON and FLTBCON Special Function Registers.

Each of the Fault pins can be controlled manually in software.

### 15.13 PWM Update Lockout

For a complex PWM application, the user may need to write up to four duty cycle registers and the time base period register, PTPER, at a given time. In some applications, it is important that all buffer registers be written before the new duty cycle and period values are loaded for use by the module.

The PWM update lockout feature is enabled by setting the UDIS control bit in the PWMCON2 SFR. The UDIS bit affects all duty cycle buffer registers and the PWM time base period buffer, PTPER. No duty cycle changes or period value changes will have effect while UDIS = 1.

### 15.14 PWM Special Event Trigger

The PWM module has a special event trigger that allows A/D conversions to be synchronized to the PWM time base. The A/D sampling and conversion time may be programmed to occur at any point within the PWM period. The special event trigger allows the user to minimize the delay between the time when A/D conversion results are acquired and the time when the duty cycle value is updated.

The PWM special event trigger has an SFR named SEVTCMP, and five control bits to control its operation. The PTMR value for which a special event trigger should occur is loaded into the SEVTCMP register. When the PWM time base is in an Up/Down Counting mode, an additional control bit is required to specify the counting phase for the special event trigger. The count phase is selected using the SEVTDIR control bit in the SEVTCMP SFR. If the SEVTDIR bit is cleared, the special event trigger will occur on the upward counting cycle of the PWM time base. If the SEVTDIR bit is set, the special event trigger will occur on the downward count cycle of the PWM time base. The SEVTDIR control bit has no effect unless the PWM time base is configured for an Up/Down Counting mode.

#### 15.14.1 SPECIAL EVENT TRIGGER POSTSCALER

The PWM special event trigger has a postscaler that allows a 1:1 to 1:16 postscale ratio. The postscaler is configured by writing the SEVOPS<3:0> control bits in the PWMCON2 SFR.

The special event output postscaler is cleared on the following events:

- Any write to the SEVTCMP register
- Any device Reset

#### 15.15 PWM Operation During CPU Sleep Mode

The Fault A and Fault B input pins have the ability to wake the CPU from Sleep mode. The PWM module generates an interrupt if either of the Fault pins is driven low while in Sleep.

### 15.16 PWM Operation During CPU Idle Mode

The PTCON SFR contains a PTSIDL control bit. This bit determines if the PWM module will continue to operate or stop when the device enters Idle mode. If PTSIDL = 0, the module will continue to operate. If PTSIDL = 1, the module will stop operation as long as the CPU remains in Idle mode.

## TABLE 15-2: 8-OUTPUT PWM REGISTER MAP

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	
PTCON	01C0	PTEN	_	PTSIDL	—	_	_	_	—		PTOP	S<3:0>		PTCKP	'S<1:0>	PTMO	).
PTMR	01C2	PTDIR							PWM Ti	imer Count	Value						
PTPER	01C4	_						P	WM Time	Base Perio	d Register						
SEVTCMP	01C6	SEVTDIR		_	_		_	PWI	M Special E	Event Com	pare Regis	ter	_	_	_	_	
PWMCON1	01C8	_	_	_	_	PTMOD4	PTMOD3	PTMOD2	PTMOD1	PEN4H	PEN3H	PEN2H	PEN1H	PEN4L	PEN3L	PEN2L	Ī
PWMCON2	01CA	_	_	_	_		SEVOF	PS<3:0>		_	_	_	_	_	_	OSYNC	ſ
DTCON1	01CC	DTBPS	S<1:0>	0> Dead-Time B Value DTAPS<1:0> Dead-Time A Valu								e A Value					
DTCON2	01CE	_	_	_	_	—	_	_	_	DTS4A	DTS4I	DTS3A	DTS3I	DTS2A	DTS2I	DTS1A	Ē
FLTACON	01D0	FAOV4H	FAOV4L	FAOV3H	FAOV3L	FAOV2H	FAOV2L	FAOV1H	FAOV1L	FLTAM	_	_	_	FAEN4	FAEN3	FAEN2	Ī
FLTBCON	01D2	FBOV4H	FBOV4L	FBOV3H	FBOV3L	FBOV2H	FBOV2L	FBOV1H	FBOV1L	FLTBM	_	_	_	FBEN4	FBEN3	FBEN2	ſ
OVDCON	01D4	POVD4H	POVD4L	POVD3H	POVD3L	POVD2H	POVD2L	POVD1H	POVD1L	POUT4H	POUT4L	POUT3H	POUT3L	POUT2H	POUT2L	POUT1H	
PDC1	01D6					-		PWN	I Duty Cycl	e #1 Regis	ter	-					
PDC2	01D8							PWM	1 Duty Cycl	e #2 Regis	ter						
PDC3	01DA							PWM	1 Duty Cycl	e #3 Regis	ter						
PDC4	01DC							PWM	1 Duty Cycl	e #4 Regis	ter						

**Legend:** u = uninitialized bit

Note: Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

## 16.0 SPI MODULE

**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).

The Serial Peripheral Interface (SPI) module is a synchronous serial interface. It is useful for communicating with other peripheral devices such as EEPROMs, shift registers, display drivers and A/D converters, or other microcontrollers. It is compatible with Motorola's SPI and SIOP interfaces.

### 16.1 Operating Function Description

Each SPI module consists of a 16-bit shift register, SPIxSR (where x = 1 or 2), used for shifting data in and out, and a buffer register, SPIxBUF. A control register, SPIxCON, configures the module. Additionally, a status register, SPIxSTAT, indicates various status conditions.

The serial interface consists of 4 pins: SDIx (serial data input), SDOx (serial data output), SCKx (shift clock input or output), and SSx (active low slave select).

In Master mode operation, SCK is a clock output, but in Slave mode, it is a clock input.

A series of eight (8) or sixteen (16) clock pulses shifts out bits from the SPIxSR to SDOx pin and simultaneously shifts in data from SDIx pin. An interrupt is generated when the transfer is complete and the corresponding Interrupt Flag bit (SPI1IF or SPI2IF) is set. This interrupt can be disabled through an Interrupt Enable bit (SPI1IE or SPI2IE).

The receive operation is double-buffered. When a complete byte is received, it is transferred from SPIxSR to SPIxBUF.

If the receive buffer is full when new data is being transferred from SPIxSR to SPIxBUF, the module will set the SPIROV bit, indicating an overflow condition. The transfer of the data from SPIxSR to SPIxBUF will not be completed and the new data will be lost. The module will not respond to SCL transitions while SPIROV is 1, effectively disabling the module until SPIxBUF is read by user software. Transmit writes are also double-buffered. The user writes to SPIxBUF. When the master or slave transfer is completed, the contents of the shift register (SPIxSR) is moved to the receive buffer. If any transmit data has been written to the buffer register, the contents of the transmit buffer are moved to SPIxSR. The received data is thus placed in SPIxBUF and the transmit data in SPIxSR is ready for the next transfer.

Note: Both the transmit buffer (SPIxTXB) and the receive buffer (SPIxRXB) are mapped to the same register address, SPIxBUF.

In Master mode, the clock is generated by prescaling the system clock. Data is transmitted as soon as a value is written to SPIxBUF. The interrupt is generated at the middle of the transfer of the last bit.

In Slave mode, data is transmitted and received as external clock pulses appear on SCK. Again, the interrupt is generated when the last bit is latched. If  $\overline{SSx}$  control is enabled, then transmission and reception are enabled only when  $\overline{SSx} = \text{low}$ . The SDOx output will be disabled in  $\overline{SSx}$  mode with  $\overline{SSx}$  high.

The clock provided to the module is (FOSC/4). This clock is then prescaled by the primary (PPRE<1:0>) and the secondary (SPRE<2:0>) prescale factors. The CKE bit determines whether transmit occurs on transition from active clock state to Idle clock state, or vice versa. The CKP bit selects the Idle state (high or low) for the clock.

#### 16.1.1 WORD AND BYTE COMMUNICATION

A control bit, MODE16 (SPIxCON<10>), allows the module to communicate in either 16-bit or 8-bit mode. 16-bit operation is identical to 8-bit operation, except that the number of bits transmitted is 16 instead of 8.

The user software must disable the module prior to changing the MODE16 bit. The SPI module is reset when the MODE16 bit is changed by the user.

A basic difference between 8-bit and 16-bit operation is that the data is transmitted out of bit 7 of the SPIxSR for 8-bit operation, and data is transmitted out of bit 15 of the SPIxSR for 16-bit operation. In both modes, data is shifted into bit 0 of the SPIxSR.

#### 16.1.2 SDOx DISABLE

A control bit, DISSDO, is provided to the SPIxCON register to allow the SDOx output to be disabled. This will allow the SPI module to be connected in an input only configuration. SDO can also be used for general purpose I/O.

# dsPIC30F6010

### FIGURE 16-1: SPI BLOCK DIAGRAM



### FIGURE 16-2: SPI MASTER/SLAVE CONNECTION



## 16.2 Framed SPI Support

The module supports a basic framed SPI protocol in Master or Slave mode. The control bit FRMEN enables framed SPI support and causes the SSx pin to perform the frame synchronization pulse (FSYNC) function. The control bit SPIFSD determines whether the SSx pin is an input or an output (i.e., whether the module receives or generates the frame synchronization pulse). The frame pulse is an active high pulse for a single SPI clock cycle. When frame synchronization is enabled, the data transmission starts only on the subsequent transmit edge of the SPI clock.

#### 16.3 Slave Select Synchronization

The SSx pin allows a Synchronous Slave mode. The SPI must be configured in SPI Slave mode, with SSx pin control enabled (SSEN = 1). When the SSx pin is low, transmission and reception are enabled, and the SDOx pin is driven. When SSx pin goes high, the SDOx pin is no longer driven. Also, the SPI module is resynchronized, and all counters/control circuitry are reset. Therefore, when the SSx pin is asserted low again, transmission/reception will begin at the MSb, even if SSx had been deasserted in the middle of a transmit/receive.

#### 16.4 SPI Operation During CPU Sleep Mode

During Sleep mode, the SPI module is shut-down. If the CPU enters Sleep mode while an SPI transaction is in progress, then the transmission and reception is aborted.

The transmitter and receiver will stop in Sleep mode. However, register contents are not affected by entering or exiting Sleep mode.

#### 16.5 SPI Operation During CPU Idle Mode

When the device enters Idle mode, all clock sources remain functional. The SPISIDL bit (SPIxSTAT<13>) selects if the SPI module will stop or continue on Idle. If SPISIDL = 0, the module will continue to operate when the CPU enters Idle mode. If SPISIDL = 1, the module will stop when the CPU enters Idle mode.

#### TABLE 16-1: SPI1 REGISTER MAP

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	
SPI1STAT	0220	SPIEN	—	SPISIDL		—	—		_	_	SPIROV	_	_	_	_	SPITBF	S
SPI1CON	0222	_	FRMEN	SPIFSD	_	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	Ρ
SPI1BUF	0224							Tra	ansmit an	d Receive	Buffer						
																	_

**Legend:** u = uninitialized bit

Note: Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

#### TABLE 16-2: SPI2 REGISTER MAP

																	-
SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	B
SPI2STAT	0226	SPIEN	_	SPISIDL	—	—	—	_	_	—	SPIROV	_		—	_	SPITBF	SP
SPI2CON	0228	_	FRMEN	SPIFSD	_	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PF
SPI2BUF	022A							Tra	ansmit and	d Receive	Buffer						

**Legend:** u = uninitialized bit

Note: Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

## 17.0 I<sup>2</sup>C MODULE

**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).

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

This module offers the following key features:

- I<sup>2</sup>C interface supporting both Master and Slave operation.
- I<sup>2</sup>C Slave mode supports 7 and 10-bit address.
- I<sup>2</sup>C Master mode supports 7 and 10-bit address.
- I<sup>2</sup>C port allows bidirectional transfers between master and slaves.

- Serial clock synchronization for I<sup>2</sup>C port can be used as a handshake mechanism to suspend and resume serial transfer (SCLREL control).
- I<sup>2</sup>C supports Multi-Master operation; detects bus collision and will arbitrate accordingly.

## 17.1 Operating Function Description

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

Thus, the I<sup>2</sup>C module can operate either as a slave or a master on an I<sup>2</sup>C bus.

## 17.1.1 VARIOUS I<sup>2</sup>C MODES

The following types of I<sup>2</sup>C operation are supported:

- I<sup>2</sup>C Slave operation with 7-bit address
- I<sup>2</sup>C Slave operation with 10-bit address
- I<sup>2</sup>C Master operation with 7 or 10-bit address

See the I<sup>2</sup>C programmer's model in Figure 17-1.

#### I2CRCV (8 bits) bit 7 bit 0 I2CTRN (8 bits) bit 7 bit 0 I2CBRG (9 bits) bit 8 bit 0 I2CCON (16 bits) bit 15 bit 0 I2CSTAT (16 bits) bit 15 bit 0 I2CADD (10 bits) bit 9 bit 0

## 17.1.2 PIN CONFIGURATION IN I<sup>2</sup>C MODE

 ${\rm I}^2{\rm C}$  has a 2-pin interface; pin SCL is clock and pin SDA is data.

## 17.1.3 I<sup>2</sup>C REGISTERS

I2CCON and I2CSTAT are control and status registers, respectively. The I2CCON register is readable and writable. The lower 6 bits of I2CSTAT are read only. The remaining bits of the I2CSTAT are read/write.

I2CRSR is the shift register used for shifting data, whereas I2CRCV is the buffer register to which data bytes are written, or from which data bytes are read. I2CRCV is the receive buffer, as shown in Figure 16-1. I2CTRN is the transmit register to which bytes are written during a transmit operation, as shown in Figure 16-2. The I2CADD register holds the slave address. A status bit, ADD10, indicates 10-bit Address mode. The I2CBRG acts as the Baud Rate Generator reload value.

In receive operations, I2CRSR and I2CRCV together form a double-buffered receiver. When I2CRSR receives a complete byte, it is transferred to I2CRCV and an interrupt pulse is generated. During transmission, the I2CTRN is not double-buffered.

**Note:** Following a Restart condition in 10-bit mode, the user only needs to match the first 7-bit address.

## FIGURE 17-1: PROGRAMMER'S MODEL

# dsPIC30F6010





## 17.2 I<sup>2</sup>C Module Addresses

The I2CADD register contains the Slave mode addresses. The register is a 10-bit register. If the A10M bit (I2CCON<10>) is '0', the address is interpreted by the module as a 7-bit address. When an address is received, it is compared to the 7 LSbs of the I2CADD register.

If the A10M bit is 1, the address is assumed to be a 10bit address. When an address is received, it will be compared with the binary value '1 1 1 1 0 A9 A8' (where A9, A8 are two Most Significant bits of I2CADD). If that value matches, the next address will be compared with the Least Significant 8 bits of I2CADD, as specified in the 10-bit addressing protocol.

Table 17-1 lists the Slave addresses supported by dsPIC30F devices.

#### TABLE 17-1: 7-BIT I<sup>2</sup>C<sup>™</sup> SLAVE ADDRESSES SUPPORTED BY dsPIC30F

0x00	General call address or Start byte
0x01-0x03	Reserved
0x04-0x07	Hs mode Master codes
0x08-0x77	Valid 7-bit addresses
0x78-0x7b	Valid 10-bit addresses (lower 7 bits)
0x7c-0x7f	Reserved

## 17.3 I<sup>2</sup>C 7-bit Slave Mode Operation

Once enabled (I2CEN = 1), the slave module waits for a Start bit to occur (i.e., the I<sup>2</sup>C module is 'Idle'). Following the detection of a Start bit, 8 bits are shifted into I2CRSR and the address is compared against I2CADD. In 7-bit mode (A10M = 0), bits I2CADD<6:0> are compared against I2CRSR<7:1> and I2CRSR<0> is the R\_W bit. All incoming bits are sampled on the rising edge of SCL.

If an address match occurs, an acknowledgement is sent, and the slave event interrupt flag (SI2CIF) is set on the falling edge of the ninth ( $\overline{ACK}$ ) bit. The address match does not affect the contents of the I2CRCV buffer or the RBF bit.

## 17.3.1 SLAVE TRANSMISSION

If the R\_W bit received is a '1', the serial port goes into Transmit mode. It sends an ACK on the ninth bit and then holds SCL to '0' until the CPU responds by writing to I2CTRN. SCL is released by setting the SCLREL bit, and 8 bits of data are shifted out. Data bits are shifted out on the falling edge of SCL, such that SDA is valid during SCL high (see timing diagram). The interrupt pulse is sent on the falling edge of the ninth clock pulse, regardless of the status of the ACK received from the master.

## 17.3.2 SLAVE RECEPTION

If the R\_W bit received is a '0' during an address match, Receive mode is initiated. Incoming bits are sampled on the rising edge of SCL. After 8 bits are received, if I2CRCV is not full or I2COV is not set, I2CRSR is transferred to I2CRCV. ACK is sent on the ninth clock.

If the RBF flag is set, indicating that I2CRCV is still holding data from a previous operation (RBF = 1), the ACK is not sent; however, the interrupt pulse is generated. In the case of an overflow, the contents of the I2CRSR are not loaded into the I2CRCV.

Note:	The I2CRCV is loaded if the I2COV bit = 1
	and the RBF flag = 0. In this case, a read
	of the I2CRCV was performed, but the
	user did not clear the state of the I2COV
	bit before the next receive occurred. The
	acknowledgement is not sent $(\overline{ACK} = 1)$
	and the I2CRCV is updated.

## 17.4 I<sup>2</sup>C 10-bit Slave Mode Operation

In 10-bit mode, the basic receive and transmit operations are the same as in the 7-bit mode. However, the criteria for address match is more complex.

The  $I^2C$  specification dictates that a slave must be addressed for a write operation, with two address bytes following a Start bit.

The A10M bit is a control bit that signifies that the address in I2CADD is a 10-bit address rather than a 7-bit address. The address detection protocol for the first byte of a message address is identical for 7-bit and 10-bit messages, but the bits being compared are different.

I2CADD holds the entire 10-bit address. Upon receiving an address following a Start bit, I2CRSR <7:3> is compared against a literal '11110' (the default 10-bit address) and I2CRSR<2:1> are compared against I2CADD<9:8>. If a match occurs, and if  $R_W = 0$ , the interrupt pulse is sent. The ADD10 bit is cleared to indicate a partial address match. If a match fails or  $R_W =$ 1, the ADD10 bit is cleared and the module returns to the Idle state.

The low byte of the address is then received and compared with I2CADD<7:0>. If an address match occurs, the interrupt pulse is generated and the ADD10 bit is set, indicating a complete 10-bit address match. If an address match did not occur, the ADD10 bit is cleared and the module returns to the Idle state.

#### 17.4.1 10-BIT MODE SLAVE TRANSMISSION

Once a slave is addressed in this fashion, with the full 10-bit address (we refer to this state as "PRIOR\_ADDR\_MATCH"), the master can begin sending data bytes for a slave reception operation.

### 17.4.2 10-BIT MODE SLAVE RECEPTION

Once addressed, the master can generate a Repeated Start, reset the high byte of the address and set the R\_W bit without generating a Stop bit, thus initiating a slave transmit operation.

## 17.5 Automatic Clock Stretch

In the slave modes, the module can synchronize buffer reads and write to the master device by clock stretching.

#### 17.5.1 TRANSMIT CLOCK STRETCHING

Both 10-bit and 7-bit transmit modes implement clock stretching by asserting the SCLREL bit after the falling edge of the ninth clock if the TBF bit is cleared, indicating the buffer is empty.

In slave transmit modes, clock stretching is always performed, irrespective of the STREN bit.

Clock synchronization takes place following the ninth clock of the transmit sequence. If the device samples an ACK on the falling edge of the ninth clock, and if the TBF bit is still clear, then the SCLREL bit is automatically cleared. The SCLREL being cleared to '0' will assert the SCL line low. The user's ISR must set the SCLREL bit before transmission is allowed to continue. By holding the SCL line low, the user has time to service the ISR and load the contents of the I2CTRN before the master device can initiate another transmit sequence.

- Note 1: If the user loads the contents of I2CTRN, setting the TBF bit before the falling edge of the ninth clock, the SCLREL bit is not cleared, and clock stretching does not occur.
  - 2: The SCLREL bit can be set in software, regardless of the state of the TBF bit.

#### 17.5.2 RECEIVE CLOCK STRETCHING

The STREN bit in the I2CCON register can be used to enable clock stretching in Slave Receive mode. When the STREN bit is set, the SCL pin is held low at the end of each data receive sequence.

#### 17.5.3 CLOCK STRETCHING DURING 7-BIT ADDRESSING (STREN = 1)

When the STREN bit is set in Slave Receive mode, the SCL line is held low when the buffer register is full. The method for stretching the SCL output is the same for both 7 and 10-bit addressing modes.

Clock stretching takes place following the ninth clock of the receive sequence. On the falling edge of the ninth clock at the end of the ACK sequence, if the RBF bit is set, the SCLREL bit is automatically cleared, forcing the SCL output to be held low. The user's ISR must set the SCLREL bit before reception is allowed to continue. By holding the SCL line low, the user has time to service the ISR and read the contents of the I2CRCV before the master device can initiate another receive sequence. This prevents buffer overruns from occurring.

- Note 1: If the user reads the contents of the I2CRCV, clearing the RBF bit before the falling edge of the ninth clock, the SCLREL bit is not cleared and clock stretching does not occur.
  - 2: The SCLREL bit can be set in software, regardless of the state of the RBF bit. The user should be careful to clear the RBF bit in the ISR before the next receive sequence in order to prevent an overflow condition.

#### 17.5.4 CLOCK STRETCHING DURING 10-BIT ADDRESSING (STREN = 1)

Clock stretching takes place automatically during the addressing sequence. Because this module has a register for the entire address, it is not necessary for the protocol to wait for the address to be updated.

After the address phase is complete, clock stretching occurs on each data receive or transmit sequence, as described earlier.

## 17.6 Software Controlled Clock Stretching (STREN = 1)

When the STREN bit is '1', the SCLREL bit can be cleared by software. The logic synchronizes writes to the SCLREL bit with the SCL clock. Clearing the SCLREL bit does not assert the SCL output until the module detects a falling edge on the SCL output and SCL is sampled low. If the SCLREL bit is cleared by the user while the SCL line has been sampled low, the SCL output is asserted (held low). The SCL output remains low until the SCLREL bit is set, and all other devices on the  $I^2$ C bus have deasserted SCL. This ensures that a write to the SCLREL bit does not violate the minimum high time requirement for SCL.

If the STREN bit is '0', a software write to the SCLREL bit is disregarded and has no effect on the SCLREL bit.

## 17.7 Interrupts

The I<sup>2</sup>C module generates two interrupt flags, MI2CIF (I<sup>2</sup>C Master Interrupt Flag) and SI2CIF (I<sup>2</sup>C Slave Interrupt Flag). The MI2CIF interrupt flag is activated on completion of a master message event. The SI2CIF interrupt flag is activated on detection of a message directed to the slave.

### 17.8 Slope Control

The I<sup>2</sup>C standard requires slope control on the SDA and SCL signals for Fast Mode (400 kHz). The control bit, DISSLW, enables the user to disable slew rate control, if desired. It is necessary to disable the slew rate control for 1 MHz mode.

#### 17.9 IPMI Support

The control bit IPMIEN enables the module to support Intelligent Peripheral Management Interface (IPMI). When this bit is set, the module accepts and acts upon all addresses.

#### 17.10 General Call Address Support

The general call address can address all devices. When this address is used, all devices should, in theory, respond with an acknowledgement.

The general call address is one of eight addresses reserved for specific purposes by the  $I^2C$  protocol. It consists of all 0's with  $R_W = 0$ .

The general call address is recognized when the General Call Enable (GCEN) bit is set (I2CCON<7> = 1). Following a Start bit detection, 8 bits are shifted into I2CRSR and the address is compared with I2CADD, and is also compared with the general call address which is fixed in hardware.

If a general call address match occurs, the I2CRSR is transferred to the I2CRCV after the eighth clock, the RBF flag is set, and on the falling edge of the ninth bit (ACK bit), the master event interrupt flag (MI2CIF) is set.

When the interrupt is serviced, the source for the interrupt can be checked by reading the contents of the I2CRCV to determine if the address was device specific, or a general call address.

## 17.11 I<sup>2</sup>C Master Support

As a Master device, six operations are supported.

- Assert a Start condition on SDA and SCL.
- Assert a Restart condition on SDA and SCL.
- Write to the I2CTRN register initiating transmission of data/address.
- Generate a Stop condition on SDA and SCL.
- Configure the I<sup>2</sup>C port to receive data.
- Generate an ACK condition at the end of a received byte of data.

## 17.12 I<sup>2</sup>C Master Operation

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the  $I^2C$  bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the data direction bit. In this case, the data direction bit (R\_W) is logic 0. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an ACK bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the data direction bit. In this case, the data direction bit (R\_W) is logic 1. Thus, the first byte transmitted is a 7-bit slave address, followed by a '1' to indicate receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an ACK bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

#### 17.12.1 I<sup>2</sup>C MASTER TRANSMISSION

Transmission of a data byte, a 7-bit address, or the second half of a 10-bit address is accomplished by simply writing a value to I2CTRN register. The user should only write to I2CTRN when the module is in a WAIT state. This action will set the buffer full flag (TBF) and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted. The Transmit Status Flag, TRSTAT (I2CSTAT<14>), indicates that a master transmit is in progress.

#### 17.12.2 I<sup>2</sup>C MASTER RECEPTION

Master mode reception is enabled by programming the receive enable (RCEN) bit (I2CCON<3>). The  $I^2C$  module must be Idle before the RCEN bit is set, otherwise the RCEN bit will be disregarded. The Baud Rate Generator (BRG) begins counting, and on each rollover, the state of the SCL pin toggles, and data is shifted in to the I2CRSR on the rising edge of each clock.

#### 17.12.3 BAUD RATE GENERATOR

In I<sup>2</sup>C Master mode, the reload value for the BRG is located in the I2CBRG register. When the BRG is loaded with this value, the BRG counts down to 0 and stops until another reload has taken place. If clock arbitration is taking place, for instance, the BRG is reloaded when the SCL pin is sampled high.

As per the  $I^2C$  standard, FSCK may be 100 kHz or 400 kHz. However, the user can specify any baud rate up to 1 MHz. I2CBRG values of 0 or 1 are illegal.

## EQUATION 17-1: SERIAL CLOCK RATE

## $I2CBRG = \left(\frac{FCY}{FSCL} - \frac{FCY}{1, 111, 111}\right) - 1$

## 17.12.4 CLOCK ARBITRATION

Clock arbitration occurs when the master deasserts the SCL pin (SCL allowed to float high) during any receive, transmit, or Restart/Stop condition. When the SCL pin is allowed to float high, the Baud Rate Generator is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the Baud Rate Generator is reloaded with the contents of I2CBRG and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device.

#### 17.12.5 MULTI-MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master operation support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high while another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a 1 and the data sampled on the SDA pin = 0, then a bus collision has taken place. The master sets the MI2CIF pulse and resets the master portion of the  $I^2C$  port to its Idle state.

If a transmit was in progress when the bus collision occurred, the transmission is halted, the TBF flag is cleared, the SDA and SCL lines are deasserted, and a value can now be written to I2CTRN. When the user services the  $I^2C$  master event Interrupt Service Routine, if the  $I^2C$  bus is free (i.e., the P bit is set) the user can resume communication by asserting a Start condition.

If a Start, Restart, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted, and the respective control bits in the I2CCON register are cleared to 0. When the user services the bus collision Interrupt Service Routine, and if the I<sup>2</sup>C bus is free, the user can resume communication by asserting a Start condition. The Master continues to monitor the SDA and SCL pins, and if a Stop condition occurs, the MI2CIF bit is set.

A write to the I2CTRN starts the transmission of data at the first data bit, regardless of where the transmitter left off when bus collision occurred.

In a Multi-Master environment, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the  $I^2C$ bus can be taken when the P bit is set in the I2CSTAT register, or the bus is Idle and the S and P bits are cleared.

## 17.13 I<sup>2</sup>C Module Operation During CPU Sleep and Idle Modes

#### 17.13.1 I<sup>2</sup>C OPERATION DURING CPU SLEEP MODE

When the device enters Sleep mode, all clock sources to the module are shutdown and stay at logic '0'. If Sleep occurs in the middle of a transmission, and the state machine is partially into a transmission as the clocks stop, the transmission is aborted. Similarly, if Sleep occurs in the middle of a reception, the reception is aborted.

## 17.13.2 I<sup>2</sup>C OPERATION DURING CPU IDLE MODE

For the I<sup>2</sup>C, the I2CSIDL bit selects if the module stops or continues on Idle. If I2CSIDL = 0, the module continues operation on assertion of the Idle mode. If I2CSIDL = 1, the module stops on Idle.

## TABLE 17-2: I<sup>2</sup>C REGISTER MAP

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	
I2CRCV	0200	—		—	—	—		—	_	Receive Register							
I2CTRN	0202	_		—	—	_			_	Transmit Register							
I2CBRG	0204	_	—	_	—	—	—	_		Baud Rate Generator							
I2CCON	0206	I2CEN	_	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	
I2CSTAT	0208	ACKSTAT	TRSTAT	—	_	_	BCL	GCSTAT	ADD10	IWCOL	I2COV	D_A	Р	S	R_W	RBF	
I2CADD	020A			—	—	_			Address Register								

**Legend:** u = uninitialized bit

**Note:** Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

# dsPIC30F6010

NOTES:
## 18.0 UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART) MODULE

**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).

This section describes the Universal Asynchronous Receiver/Transmitter Communications module.

### 18.1 UART Module Overview

The key features of the UART module are:

- Full-duplex, 8 or 9-bit data communication
- Even, Odd or No Parity options (for 8-bit data)
- · One or two Stop bits
- Fully integrated Baud Rate Generator with 16-bit prescaler
- Baud rates range from 38 bps to 1.875 Mbps at a 30 MHz instruction rate
- 4-word deep transmit data buffer
- 4-word deep receive data buffer
- Parity, Framing and Buffer Overrun error detection
- Support for Interrupt only on Address Detect (9th bit = 1)
- Separate Transmit and Receive Interrupts
- Loopback mode for diagnostic support



#### FIGURE 18-1: UART TRANSMITTER BLOCK DIAGRAM





## 18.2 Enabling and Setting Up UART

#### 18.2.1 ENABLING THE UART

The UART module is enabled by setting the UARTEN bit in the UXMODE register (where x = 1 or 2). Once enabled, the UxTX and UxRX pins are configured as an output and an input respectively, overriding the TRIS and LATCH Register bit settings for the corresponding I/O port pins. The UxTX pin is at logic '1' when no transmission is taking place.

#### 18.2.2 DISABLING THE UART

The UART module is disabled by clearing the UARTEN bit in the UxMODE register. This is the default state after any Reset. If the UART is disabled, all I/O pins operate as port pins under the control of the latch and TRIS bits of the corresponding port pins.

Disabling the UART module resets the buffers to empty states. Any data characters in the buffers are lost, and the baud rate counter is reset.

All error and status flags associated with the UART module are reset when the module is disabled. The URXDA, OERR, FERR, PERR, UTXEN, UTXBRK and UTXBF bits are cleared, whereas RIDLE and TRMT are set. Other control bits, including ADDEN, URXISEL<1:0>, UTXISEL, as well as the UXMODE and UxBRG registers, are not affected.

Clearing the UARTEN bit while the UART is active will abort all pending transmissions and receptions and reset the module as defined above. Re-enabling the UART will restart the UART in the same configuration.

## 18.2.3 SETTING UP DATA, PARITY AND STOP BIT SELECTIONS

Control bits PDSEL<1:0> in the UxMODE register are used to select the data length and parity used in the transmission. The data length may either be 8 bits with even, odd or no parity, or 9 bits with no parity.

The STSEL bit determines whether one or two Stop bits will be used during data transmission.

The default (Power-on) setting of the UART is 8 bits, no parity, 1 Stop bit (typically represented as 8, N, 1).

## 18.3 Transmitting Data

#### 18.3.1 TRANSMITTING IN 8-BIT DATA MODE

The following steps must be performed in order to transmit 8-bit data:

- 1. Set up the UART:
  - First, the data length, parity and number of Stop bits must be selected. Then, the Transmit and Receive Interrupt enable and priority bits are setup in the UxMODE and UxSTA registers. Also, the appropriate baud rate value must be written to the UxBRG register.
- 2. Enable the UART by setting the UARTEN bit (UxMODE<15>).
- 3. Set the UTXEN bit (UxSTA<10>), thereby enabling a transmission.

Note: The UTXEN bit must be set after the UARTEN bit is set to enable UART transmissions.

- 4. Write the byte to be transmitted to the lower byte of UxTXREG. The value will be transferred to the Transmit Shift register (UxTSR) immediately and the serial bit stream will start shifting out during the next rising edge of the baud clock. Alternatively, the data byte may be written while UTXEN = 0, following which, the user may set UTXEN. This will cause the serial bit stream to begin immediately because the baud clock will start from a cleared state.
- 5. A Transmit interrupt will be generated depending on the value of the interrupt control bit UTXISEL (UxSTA<15>).

#### 18.3.2 TRANSMITTING IN 9-BIT DATA MODE

The sequence of steps involved in the transmission of 9-bit data is similar to 8-bit transmission, except that a 16-bit data word (of which the upper 7 bits are always clear) must be written to the UxTXREG register.

## 18.3.3 TRANSMIT BUFFER (UXTXB)

The transmit buffer is 9 bits wide and 4 characters deep. Including the Transmit Shift Register (UxTSR), the user effectively has a 5-deep FIFO (First In First Out) buffer. The UTXBF Status bit (UxSTA<9>) indicates whether the transmit buffer is full.

If a user attempts to write to a full buffer, the new data will not be accepted into the FIFO, and no data shift will occur within the buffer. This enables recovery from a buffer overrun condition.

The FIFO is reset during any device Reset, but is not affected when the device enters or wakes up from a Power Saving mode.

#### 18.3.4 TRANSMIT INTERRUPT

The transmit interrupt flag (U1TXIF or U2TXIF) is located in the corresponding interrupt flag register.

The transmitter generates an edge to set the UxTXIF bit. The condition for generating the interrupt depends on UTXISEL control bit:

- a) If UTXISEL = 0, an interrupt is generated when a word is transferred from the Transmit buffer to the Transmit Shift register (UxTSR). This implies that the transmit buffer has at least one empty word.
- b) If UTXISEL = 1, an interrupt is generated when a word is transferred from the Transmit buffer to the Transmit Shift register (UxTSR) and the Transmit buffer is empty.

Switching between the two interrupt modes during operation is possible and sometimes offers more flexibility.

#### 18.3.5 TRANSMIT BREAK

Setting the UTXBRK bit (UxSTA<11>) will cause the UxTX line to be driven to logic '0'. The UTXBRK bit overrides all transmission activity. Therefore, the user should generally wait for the transmitter to be Idle before setting UTXBRK.

To send a break character, the UTXBRK bit must be set by software and must remain set for a minimum of 13 baud clock cycles. The UTXBRK bit is then cleared by software to generate Stop bits. The user must wait for a duration of at least one or two baud clock cycles in order to ensure a valid Stop bit(s) before reloading the UxTXB or starting other transmitter activity. Transmission of a break character does not generate a transmit interrupt.

## 18.4 Receiving Data

#### 18.4.1 RECEIVING IN 8-BIT OR 9-BIT DATA MODE

The following steps must be performed while receiving 8-bit or 9-bit data:

- 1. Set up the UART (see Section 18.3.1 "Transmitting in 8-bit data mode").
- 2. Enable the UART (see Section 18.3.1 "Transmitting in 8-bit data mode").
- A receive interrupt will be generated when one or more data words have been received, depending on the receive interrupt settings specified by the URXISEL bits (UxSTA<7:6>).
- 4. Read the OERR bit to determine if an overrun error has occurred. The OERR bit must be reset in software.
- Read the received data from UxRXREG. The act of reading UxRXREG will move the next word to the top of the receive FIFO, and the PERR and FERR values will be updated.

#### 18.4.2 RECEIVE BUFFER (UXRXB)

The receive buffer is 4 words deep. Including the Receive Shift register (UxRSR), the user effectively has a 5-word deep FIFO buffer.

URXDA (UxSTA<0>) = 1 indicates that the receive buffer has data available. URXDA = 0 implies that the buffer is empty. If a user attempts to read an empty buffer, the old values in the buffer will be read and no data shift will occur within the FIFO.

The FIFO is reset during any device Reset. It is not affected when the device enters or wakes up from a Power Saving mode.

#### 18.4.3 RECEIVE INTERRUPT

The receive interrupt flag (U1RXIF or U2RXIF) can be read from the corresponding interrupt flag register. The interrupt flag is set by an edge generated by the receiver. The condition for setting the receive interrupt flag depends on the settings specified by the URXISEL<1:0> (UxSTA<7:6>) control bits.

- a) If URXISEL<1:0> = 00 or 01, an interrupt is generated every time a data word is transferred from the Receive Shift Register (UxRSR) to the Receive Buffer. There may be one or more characters in the receive buffer.
- b) If URXISEL<1:0> = 10, an interrupt is generated when a word is transferred from the Receive Shift Register (UxRSR) to the Receive Buffer, which, as a result of the transfer, contains 3 characters.
- c) If URXISEL<1:0> = 11, an interrupt is set when a word is transferred from the Receive Shift Register (UxRSR) to the Receive Buffer, which, as a result of the transfer, contains 4 characters (i.e., becomes full).

Switching between the Interrupt modes during operation is possible, though generally not advisable during normal operation.

## 18.5 Reception Error Handling

#### 18.5.1 RECEIVE BUFFER OVERRUN ERROR (OERR BIT)

The OERR bit (UxSTA<1>) is set if all of the following conditions occur:

- a) The receive buffer is full.
- b) The receive shift register is full, but unable to transfer the character to the receive buffer.
- c) The Stop bit of the character in the UxRSR is detected, indicating that the UxRSR needs to transfer the character to the buffer.

Once OERR is set, no further data is shifted in UxRSR (until the OERR bit is cleared in software or a Reset occurs). The data held in UxRSR and UxRXREG remains valid.

#### 18.5.2 FRAMING ERROR (FERR)

The FERR bit (UxSTA<2>) is set if a '0' is detected instead of a Stop bit. If two Stop bits are selected, both Stop bits must be '1', otherwise FERR will be set. The read only FERR bit is buffered along with the received data. It is cleared on any Reset.

#### 18.5.3 PARITY ERROR (PERR)

The PERR bit (UxSTA<3>) is set if the parity of the received word is incorrect. This error bit is applicable only if a Parity mode (odd or even) is selected. The read only PERR bit is buffered along with the received data bytes. It is cleared on any Reset.

#### 18.5.4 IDLE STATUS

When the receiver is active (i.e., between the initial detection of the Start bit and the completion of the Stop bit), the RIDLE bit (UxSTA<4>) is '0'. Between the completion of the Stop bit and detection of the next Start bit, the RIDLE bit is '1', indicating that the UART is Idle.

#### 18.5.5 RECEIVE BREAK

The receiver will count and expect a certain number of bit times based on the values programmed in the PDSEL (UxMODE<2:1>) and STSEL (UxMODE<0>) bits.

If the break is longer than 13 bit times, the reception is considered complete after the number of bit times specified by PDSEL and STSEL. The URXDA bit is set, FERR is set, zeros are loaded into the receive FIFO, interrupts are generated, if appropriate and the RIDLE bit is set.

When the module receives a long break signal and the receiver has detected the Start bit, the data bits and the invalid Stop bit (which sets the FERR), the receiver must wait for a valid Stop bit before looking for the next Start bit. It cannot assume that the break condition on the line is the next Start bit.

Break is regarded as a character containing all 0's, with the FERR bit set. The break character is loaded into the buffer. No further reception can occur until a Stop bit is received. Note that RIDLE goes high when the Stop bit has not been received yet.

#### 18.6 Address Detect Mode

Setting the ADDEN bit (UxSTA<5>) enables this special mode, in which a 9th bit (URX8) value of '1' identifies the received word as an address rather than data. This mode is only applicable for 9-bit data communication. The URXISEL control bit does not have any impact on interrupt generation in this mode, since an interrupt (if enabled) will be generated every time the received word has the 9th bit set.

#### 18.7 Loopback Mode

Setting the LPBACK bit enables this special mode in which the UxTX pin is internally connected to the UxRX pin. When configured for the Loopback mode, the UxRX pin is disconnected from the internal UART receive logic. However, the UxTX pin still functions as in a normal operation.

To select this mode:

- a) Configure UART for desired mode of operation.
- b) Set LPBACK = 1 to enable Loopback mode.
- c) Enable transmission as defined in **Section 18.3** "**Transmitting Data**".

#### 18.8 Baud Rate Generator

The UART has a 16-bit Baud Rate Generator to allow maximum flexibility in baud rate generation. The Baud Rate Generator register (UxBRG) is readable and writable. The baud rate is computed as follows:

- BRG = 16-bit value held in UxBRG register (0 through 65535)
- FCY = Instruction Clock Rate (1/TCY)

The Baud Rate is given by Equation 18-1.

#### EQUATION 18-1: BAUD RATE

Baud Rate = FCY/(16\*(BRG+1))

Therefore, maximum baud rate possible is

FCY/16 (if BRG = 0),

and the minimum baud rate possible is

Fcy/(16\* 65536).

With a full 16-bit Baud Rate Generator, at 30 MIPS operation, the minimum baud rate achievable is 28.5 bps.

#### 18.9 Auto Baud Support

To allow the system to determine baud rates of received characters, the input can be optionally linked to a selected capture input. To enable this mode, the user must program the input capture module to detect the falling and rising edges of the Start bit.

## 18.10 UART Operation During CPU Sleep and Idle Modes

#### 18.10.1 UART OPERATION DURING CPU SLEEP MODE

When the device enters Sleep mode, all clock sources to the module are shutdown and stay at logic '0'. If entry into Sleep mode occurs while a transmission is in progress, then the transmission is aborted. The UxTX pin is driven to logic '1'. Similarly, if entry into Sleep mode occurs while a reception is in progress, then the reception is aborted. The UxSTA, UxMODE, transmit and receive registers and buffers, and the UxBRG register are not affected by Sleep mode.

If the WAKE bit (UxMODE<7>) is set before the device enters Sleep mode, then a falling edge on the UxRX pin will generate a receive interrupt. The Receive Interrupt Select Mode bit (URXISEL) has no effect for this function. If the receive interrupt is enabled, then this will wake-up the device from Sleep. The UARTEN bit must be set in order to generate a wake-up interrupt.

## 18.10.2 UART OPERATION DURING CPU IDLE MODE

For the UART, the USIDL bit selects if the module will stop operation when the device enters Idle mode, or whether the module will continue on Idle. If USIDL = 0, the module will continue operation during Idle mode. If USIDL = 1, the module will stop on Idle.

### TABLE 18-1: UART1 REGISTER MAP

							_										_					
SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1						
U1MODE	020C	UARTEN	_	USIDL	_	_	_	_	_	WAKE	LPBACK	ABAUD	_	—	PDSEL1	PDSEL0	;					
U1STA	020E	UTXISEL	_	—	_	UTXBRK	UTXEN	UTXBF	TRMT	T URXISEL1 URXISEL0 ADDEN RIDLE PERR FERR OERR												
U1TXREG	0210	—	_	—	-	_		—	UTX8	Transmit Register												
U1RXREG	0212	—	_	_	_	—	_	_	URX8	Receive Register												
U1BRG	0214							Ba	ud Rate Ge	Rate Generator Prescaler												

**Legend:** u = uninitialized bit

Note: Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

#### TABLE 18-2: UART2 REGISTER MAP

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1				
U2MODE	0216	UARTEN	_	USIDL	_	_	_	_		WAKE	LPBACK	ABAUD		_	PDSEL1	PDSEL0	Ś			
U2STA	0218	UTXISEL		_		UTXBRK	UTXEN	UTXBF	TRMT	T URXISEL1 URXISEL0 ADDEN RIDLE PERR FERR OERR										
U2TXREG	021A	_	-	—	-	-	_	—	UTX8	8 Transmit Register										
U2RXREG	021C	_	Ι	_			_	_	URX8	8 Receive Register										
U2BRG	021E	Baud Rate Generator Prescaler																		

**Legend:** u = uninitialized bit

Note: Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

NOTES:

## 19.0 CAN MODULE

**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).

### 19.1 Overview

The Controller Area Network (CAN) module is a serial interface, useful for communicating with other CAN modules or microcontroller devices. This interface/ protocol was designed to allow communications within noisy environments. The dsPIC30F6010 has 2 CAN modules.

The CAN module is a communication controller implementing the CAN 2.0 A/B protocol, as defined in the BOSCH specification. The module will support CAN 1.2, CAN 2.0A, CAN2.0B Passive and CAN 2.0B Active versions of the protocol. The module implementation is a full CAN system. The CAN specification is not covered within this data sheet. The reader may refer to the BOSCH CAN specification for further details.

The module features are as follows:

- Implementation of the CAN protocol CAN 1.2, CAN 2.0A and CAN 2.0B
- Standard and extended data frames
- 0-8 bytes data length
- Programmable bit rate up to 1 Mbit/sec
- · Support for remote frames
- Double-buffered receiver with two prioritized received message storage buffers (each buffer may contain up to 8 bytes of data)
- 6 full (standard/extended identifier) acceptance filters, 2 associated with the high priority receive buffer, and 4 associated with the low priority receive buffer
- 2 full acceptance filter masks, one each associated with the high and low priority receive buffers
- Three transmit buffers with application specified prioritization and abort capability (each buffer may contain up to 8 bytes of data)
- Programmable wake-up functionality with integrated low pass filter
- Programmable Loopback mode supports self-test operation
- Signaling via interrupt capabilities for all CAN receiver and transmitter error states
- Programmable clock source
- Programmable link to timer module for time-stamping and network synchronization
- · Low power Sleep and Idle mode

The CAN bus module consists of a protocol engine, and message buffering/control. The CAN protocol engine handles all functions for receiving and transmitting messages on the CAN bus. Messages are transmitted by first loading the appropriate data registers. Status and errors can be checked by reading the appropriate registers. Any message detected on the CAN bus is checked for errors and then matched against filters to see if it should be received and stored in one of the receive registers.

## 19.2 Frame Types

The CAN module transmits various types of frames, which include data messages or remote transmission Requests initiated by the user as other frames that are automatically generated for control purposes. The following frame types are supported:

Standard Data Frame

A Standard Data Frame is generated by a node when the node wishes to transmit data. It includes a 11-bit Standard Identifier (SID) but not an 18-bit Extended Identifier (EID).

• Extended Data Frame

An Extended Data Frame is similar to a Standard Data Frame, but includes an Extended Identifier as well.

Remote Frame

It is possible for a destination node to request the data from the source. For this purpose, the destination node sends a Remote Frame with an identifier that matches the identifier of the required Data Frame. The appropriate data source node will then send a Data Frame as a response to this Remote request.

Error Frame

An Error Frame is generated by any node that detects a bus error. An error frame consists of 2 fields: an Error Flag field and an Error Delimiter field.

Overload Frame

An Overload Frame can be generated by a node as a result of 2 conditions. First, the node detects a dominant bit during Interframe Space which is an illegal condition. Second, due to internal conditions, the node is not yet able to start reception of the next message. A node may generate a maximum of 2 sequential Overload Frames to delay the start of the next message.

Interframe Space

Interframe Space separates a proceeding frame (of whatever type) from a following Data or Remote Frame.



## 19.3 Modes of Operation

The CAN Module can operate in one of several operation modes selected by the user. These modes include:

- Initialization Mode
- Disable Mode
- Normal Operation Mode
- Listen Only Mode
- Loop Back Mode
- Error Recognition Mode

Modes are requested by setting the REQOP<2:0> bits (CiCTRL<10:8>). Entry into a mode is acknowledged by monitoring the OPMODE<2:0> bits (CiCTRL<7:5>). The module will not change the mode and the OPMODE bits until a change in mode is acceptable, generally during bus idle time which is defined as at least 11 consecutive recessive bits.

#### 19.3.1 INITIALIZATION MODE

In the Initialization mode, the module will not transmit or receive. The error counters are cleared and the interrupt flags remain unchanged. The programmer will have access to configuration registers that are access restricted in other modes. The module will protect the user from accidentally violating the CAN protocol through programming errors. All registers which control the configuration of the module can not be modified while the module is on-line. The CAN module will not be allowed to enter the Configuration mode while a transmission is taking place. The Configuration mode serves as a lock to protect the following registers.

- All Module Control Registers
- Baud Rate and interrupt Configuration Registers
- Bus Timing Registers
- Identifier Acceptance Filter Registers
- Identifier Acceptance Mask Registers

#### 19.3.2 DISABLE MODE

In Disable mode, the module will not transmit or receive. The module has the ability to set the WAKIF bit due to bus activity, however any pending interrupts will remain and the error counters will retain their value.

If the REQOP<2:0> bits (CiCTRL<10:8>) = 001, the module will enter the Module Disable mode. If the module is active, the module will wait for 11 recessive bits on the CAN bus, detect that condition as an idle bus, then accept the module disable command. When the OPMODE<2:0> bits (CiCTRL<7:5>) = 001, that indicates whether the module successfully went into Module Disable mode. The I/O pins will revert to normal I/O function when the module is in the Module Disable mode.

The module can be programmed to apply a low-pass filter function to the CiRX input line while the module or the CPU is in Sleep mode. The WAKFIL bit (CiCFG2<14>) enables or disables the filter.

Note: Typically, if the CAN module is allowed to transmit in a particular mode of operation and a transmission is requested immediately after the CAN module has been placed in that mode of operation, the module waits for 11 consecutive recessive bits on the bus before starting transmission. If the user switches to Disable mode within this 11-bit period, then this transmission is aborted and the corresponding TXABT bit is set and TXREQ bit is cleared.

### 19.3.3 NORMAL OPERATION MODE

Normal operating mode is selected when REQOP<2:0> = 000. In this mode, the module is activated, the I/O pins will assume the CAN bus functions. The module will transmit and receive CAN bus messages via the CxTX and CxRX pins.

#### 19.3.4 LISTEN ONLY MODE

If the Listen Only mode is activated, the module on the CAN bus is passive. The transmitter buffers revert to the Port I/O function. The receive pins remain inputs. For the receiver, no error flags or acknowledge signals are sent. The error counters are deactivated in this state. The Listen Only mode can be used for detecting the baud rate on the CAN bus. To use this, it is necessary that there are at least two further nodes that communicate with each other.

#### 19.3.5 ERROR RECOGNITION MODE

The module can be set to ignore all errors and receive any message. The Error Recognition mode is activated by setting the RXM<1:0> bits (CiRXnCON<6:5>) registers to '11'. In this mode the data which is in the message assembly buffer until the time an error occurred, is copied in the receive buffer and can be read via the CPU interface.

#### 19.3.6 LOOP BACK MODE

If the Loopback mode is activated, the module will connect the internal transmit signal to the internal receive signal at the module boundary. The transmit and receive pins revert to their Port I/O function.

## 19.4 Message Reception

#### 19.4.1 RECEIVE BUFFERS

The CAN bus module has 3 receive buffers. However, one of the receive buffers is always committed to monitoring the bus for incoming messages. This buffer is called the message assembly buffer (MAB). So there are 2 receive buffers visible, RXB0 and RXB1, that can essentially instantaneously receive a complete message from the protocol engine.

All messages are assembled by the MAB, and are transferred to the RXBn buffers only if the acceptance filter criterion are met. When a message is received, the RXnIF flag (CiINTF<0> or CiINRF<1>) will be set. This bit can only be set by the module when a message is received. The bit is cleared by the CPU when it has completed processing the message in the buffer. If the RXnIE bit (CiINTE<0> or CiINTE<1>) is set, an

interrupt will be generated when a message is received.

RXF0 and RXF1 filters with RXM0 mask are associated with RXB0. The filters RXF2, RXF3, RXF4, and RXF5 and the mask RXM1 are associated with RXB1.

#### 19.4.2 MESSAGE ACCEPTANCE FILTERS

The message acceptance filters and masks are used to determine if a message in the message assembly buffer should be loaded into either of the receive buffers. Once a valid message has been received into the Message Assembly Buffer (MAB), the identifier fields of the message are compared to the filter values. If there is a match, that message will be loaded into the appropriate receive buffer.

The acceptance filter looks at incoming messages for the RXIDE bit (CiRXnSID<0>) to determine how to compare the identifiers. If the RXIDE bit is clear, the message is a standard frame, and only filters with the EXIDE bit (CiRXFnSID<0>) clear are compared. If the RXIDE bit is set, the message is an extended frame, and only filters with the EXIDE bit set are compared. Configuring the RXM<1:0> bits to 01 or 10 can override the EXIDE bit.

#### 19.4.3 MESSAGE ACCEPTANCE FILTER MASKS

The mask bits essentially determine which bits to apply the filter to. If any mask bit is set to a zero, then that bit will automatically be accepted regardless of the filter bit. There are 2 programmable acceptance filter masks associated with the receive buffers, one for each buffer.

#### 19.4.4 RECEIVE OVERRUN

An overrun condition occurs when the Message Assembly Buffer (MAB) has assembled a valid received message, the message is accepted through the acceptance filters, and when the receive buffer associated with the filter has not been designated as clear of the previous message.

The overrun error flag, RXnOVR (CiINTF<15> or CiINTF<14>) and the ERRIF bit (CiINTF<5>) will be set and the message in the MAB will be discarded.

If the DBEN bit is clear, RXB1 and RXB0 operate independently. When this is the case, a message intended for RXB0 will not be diverted into RXB1 if RXB0 contains an unread message and the RX00VR bit will be set.

If the DBEN bit is set, the overrun for RXB0 is handled differently. If a valid message is received for RXB0 and RXFUL = 1 indicates that RXB0 is full, and RXFUL = 0 indicates that RXB1 is empty, the message for RXB0 will be loaded into RXB1. An overrun error will not be generated for RXB0. If a valid message is received for RXB0 and RXFUL = 1, and RXFUL = 1 indicating that both RXB0 and RXB1 are full, the message will be lost and an overrun will be indicated for RXB1.

#### 19.4.5 RECEIVE ERRORS

The CAN module will detect the following receive errors:

- Cyclic Redundancy Check (CRC) Error
- Bit Stuffing Error
- Invalid message receive error

These receive errors do not generate an interrupt. However, the receive error counter is incremented by one in case one of these errors occur. The RXWAR bit (CiINTF<9>) indicates that the Receive Error Counter has reached the CPU warning limit of 96 and an interrupt is generated.

#### 19.4.6 RECEIVE INTERRUPTS

Receive interrupts can be divided into 3 major groups, each including various conditions that generate interrupts:

Receive Interrupt

A message has been successfully received and loaded into one of the receive buffers. This interrupt is activated immediately after receiving the End-of-Frame (EOF) field. Reading the RXnIF flag will indicate which receive buffer caused the interrupt.

• Wake-up interrupt

The CAN module has woken up from Disable mode or the device has woken up from Sleep mode.

• Receive Error Interrupts

A receive error interrupt will be indicated by the ERRIF bit. This bit shows that an error condition occurred. The source of the error can be determined by checking the bits in the CAN Interrupt Status Register CiINTF.

- Invalid message received
- If any type of error occurred during reception of the last message, an error will be indicated by the IVRIF bit.
- Receiver overrun
- The RXnOVR bit indicates that an overrun condition occurred.
- Receiver warning
- The RXWAR bit indicates that the Receive Error Counter (RERRCNT<7:0>) has reached the Warning limit of 96.
- Receiver error passive
- The RXEP bit indicates that the Receive Error Counter has exceeded the Error Passive limit of 127 and the module has gone into Error Passive state.

## 19.5 Message Transmission

#### 19.5.1 TRANSMIT BUFFERS

The CAN module has three transmit buffers. Each of the three buffers occupies 14 bytes of data. Eight of the bytes are the maximum 8 bytes of the transmitted message. Five bytes hold the standard and extended identifiers and other message arbitration information.

#### 19.5.2 TRANSMIT MESSAGE PRIORITY

Transmit priority is a prioritization within each node of the pending transmittable messages. There are 4 levels of transmit priority. If TXPRI<1:0> (CiTXnCON<1:0>, where n = 0, 1 or 2 represents a particular transmit buffer) for a particular message buffer is set to '11', that buffer has the highest priority. If TXPRI<1:0> for a particular message buffer is set to '10' or '01', that buffer has an intermediate priority. If TXPRI<1:0> for a particular message buffer is '00', that buffer has the lowest priority.

#### 19.5.3 TRANSMISSION SEQUENCE

To initiate transmission of the message, the TXREQ bit (CiTXnCON<3>) must be set. The CAN bus module resolves any timing conflicts between setting of the TXREQ bit and the Start of Frame (SOF), ensuring that if the priority was changed, it is resolved correctly before the SOF occurs. When TXREQ is set, the TXABT (CiTXnCON<6>), TXLARB (CiTXnCON<5>) and TXERR (CiTXnCON<4>) flag bits are automatically cleared. Setting TXREQ bit simply flags a message buffer as enqueued for transmission. When the module detects an available bus, it begins transmitting the message which has been determined to have the highest priority.

If the transmission completes successfully on the first attempt, the TXREQ bit is cleared automatically and an interrupt is generated if TXIE was set.

If the message transmission fails, one of the error condition flags will be set and the TXREQ bit will remain set indicating that the message is still pending for transmission. If the message encountered an error condition during the transmission attempt, the TXERR bit will be set and the error condition may cause an interrupt. If the message loses arbitration during the transmission attempt, the TXLARB bit is set. No interrupt is generated to signal the loss of arbitration.

#### 19.5.4 ABORTING MESSAGE TRANSMISSION

The system can also abort a message by clearing the TXREQ bit associated with each message buffer. Setting the ABAT bit (CiCTRL<12>) will request an abort of all pending messages. If the message has not yet started transmission, or if the message started but is interrupted by loss of arbitration or an error, the abort will be processed. The abort is indicated when the module sets the TXABT bit, and the TXnIF flag is not automatically set.

#### 19.5.5 TRANSMISSION ERRORS

The CAN module will detect the following transmission errors:

- Acknowledge Error
- Form Error
- Bit Error

These transmission errors will not necessarily generate an interrupt but are indicated by the transmission error counter. However, each of these errors will cause the transmission error counter to be incremented by one. Once the value of the error counter exceeds the value of 96, the ERRIF (CiINTF<5>) and the TXWAR bit (CiINTF<10>) are set. Once the value of the error counter exceeds the value of 96, an interrupt is generated and the TXWAR bit in the error flag register is set.

#### 19.5.6 TRANSMIT INTERRUPTS

Transmit interrupts can be divided into 2 major groups, each including various conditions that generate interrupts:

Transmit Interrupt

At least one of the three transmit buffers is empty (not scheduled) and can be loaded to schedule a message for transmission. Reading the TXnIF flags will indicate which transmit buffer is available and caused the interrupt.

• Transmit Error Interrupts

A transmission error interrupt will be indicated by the ERRIF flag. This flag shows that an error condition occurred. The source of the error can be determined by checking the error flags in the CAN Interrupt Status register, CiINTF. The flags in this register are related to receive and transmit errors.

- Transmitter Warning Interrupt
- The TXWAR bit indicates that the Transmit Error Counter has reached the CPU warning limit of 96.
- Transmitter Error Passive
- The TXEP bit (CiINTF<12>) indicates that the Transmit Error Counter has exceeded the Error Passive limit of 127 and the module has gone to Error Passive state.
- Bus Off
- The TXBO bit (CiINTF<13>) indicates that the Transmit Error Counter has exceeded 255 and the module has gone to Bus Off state.

## 19.6 Baud Rate Setting

All nodes on any particular CAN bus must have the same nominal bit rate. In order to set the baud rate, the following parameters have to be initialized:

- Synchronization Jump Width
- · Baud rate prescaler
- Phase segments
- Length determination of Phase2 Seg
- Sample Point
- · Propagation segment bits

#### 19.6.1 BIT TIMING

All controllers on the CAN bus must have the same baud rate and bit length. However, different controllers are not required to have the same master oscillator clock. At different clock frequencies of the individual controllers, the baud rate has to be adjusted by adjusting the number of time quanta in each segment.

The Nominal Bit Time can be thought of as being divided into separate non-overlapping time segments. These segments are shown in Figure 19-2.

- Synchronization segment (Sync Seg)
- Propagation time segment (Prop Seg)
- Phase segment 1 (Phase1 Seg)
- Phase segment 2 (Phase2 Seg)

The time segments and also the nominal bit time are made up of integer units of time called time quanta or Tq. By definition, the Nominal Bit Time has a minimum of 8 Tq and a maximum of 25 Tq. Also, by definition, the minimum nominal bit time is 1  $\mu$ sec, corresponding to a maximum bit rate of 1 MHz.



#### FIGURE 19-2: CAN BIT TIMING

#### 19.6.2 PRESCALER SETTING

There is a programmable prescaler, with integral values ranging from 1 to 64, in addition to a fixed divideby-2 for clock generation. The Time Quantum (TQ) is a fixed unit of time derived from the oscillator period, and is given by Equation 19-1, where FCAN is FCY (if the CANCKS bit is set or 4 FCY (if CANCKS is cleared).

Note:	FCAN	must	not	exceed	30	MHz.	lf
	CANC	KS = 0	, the	n FCY mu	ist no	ot exce	ed
	7.5 MH	Ηz.					

#### EQUATION 19-1: TIME QUANTUM FOR CLOCK GENERATION

TQ = 2 (BRP < 5:0 > +1)/FCAN

#### 19.6.3 PROPAGATION SEGMENT

This part of the bit time is used to compensate physical delay times within the network. These delay times consist of the signal propagation time on the bus line and the internal delay time of the nodes. The Propagation Segment can be programmed from  $1 \text{ T}_Q$  to  $8 \text{ T}_Q$  by setting the PRSEG<2:0> bits (CiCFG2<2:0>).

#### 19.6.4 PHASE SEGMENTS

The phase segments are used to optimally locate the sampling of the received bit within the transmitted bit time. The sampling point is between Phase1 Seg and Phase2 Seg. These segments are lengthened or shortened by re-synchronization. The end of the Phase1 Seg determines the sampling point within a bit period. The segment is programmable from 1 Tq to 8 Tq. Phase2 Seg provides delay to the next transmitted data transition. The segment is programmable from 1 Tq to 8 Tq. Phase2 Seg or the Information Processing Time (2 Tq). The Phase1 Seg is initialized by setting bits SEG1PH<2:0> (CiCFG2<5:3>), and Phase2 Seg is initialized by setting SEG2PH<2:0> (CiCFG2<10:8>).

The following requirement must be fulfilled while setting the lengths of the Phase Segments:

 Propagation Segment + Phase1 Seg > = Phase2 Seg

#### 19.6.5 SAMPLE POINT

The Sample Point is the point of time at which the bus level is read and interpreted as the value of that respective bit. The location is at the end of Phase1 Seg. If the bit timing is slow and contains many TQ, it is possible to specify multiple sampling of the bus line at the sample point. The level determined by the CAN bus then corresponds to the result from the majority decision of three values. The majority samples are taken at the sample point and twice before with a distance of TQ/2. The CAN module allows the user to chose between sampling three times at the same point or once at the same point, by setting or clearing the SAM bit (CiCFG2<6>).

Typically, the sampling of the bit should take place at about 60-70% through the bit time, depending on the system parameters.

#### 19.6.6 SYNCHRONIZATION

To compensate for phase shifts between the oscillator frequencies of the different bus stations, each CAN controller must be able to synchronize to the relevant signal edge of the incoming signal. When an edge in the transmitted data is detected, the logic will compare the location of the edge to the expected time (Synchronous Segment). The circuit will then adjust the values of Phase1 Seg and Phase2 Seg. There are 2 mechanisms used to synchronize.

#### 19.6.6.1 Hard Synchronization

Hard Synchronization is only done whenever there is a 'recessive' to 'dominant' edge during Bus Idle, indicating the start of a message. After hard synchronization, the bit time counters are restarted with the Synchronous Segment. Hard synchronization forces the edge which has caused the hard synchronization to lie within the synchronization segment of the restarted bit time. If a hard synchronization is done, there will not be a re-synchronization within that bit time.

#### 19.6.6.2 Resynchronization

As a result of resynchronization, Phase1 Seg may be lengthened or Phase2 Seg may be shortened. The amount of lengthening or shortening of the phase buffer segment has an upper bound known as the Synchronization Jump Width, and is specified by the SJW<1:0> bits (CiCFG1<7:6>). The value of the synchronization jump width will be added to Phase1 Seg or subtracted from Phase2 Seg. The re-synchronization jump width is programmable between 1 To and 4 To.

The following requirement must be fulfilled while setting the SJW<1:0> bits:

• Phase2 Seg > Synchronization Jump Width

## TABLE 19-1: CAN1 REGISTER MAP

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	ſ
C1RXF0SID	0300	—	_	—				Receive A	cceptance	e Filter 0 Stand	dard Ident	ifier <10:0>	•			—	
C1RXF0EIDH	0302	_			_				Rece	ive Acceptanc	e Filter 0	Extended I	dentifier <	17:6>			
C1RXF0EIDL	0304	Rece	vive Accepta	nce Filter 0 E	xtended Id	entifier <5:0	)>	—	—	—	—	—	_	—	—	—	
C1RXF1SID	0308	—	—	_				Receive A	cceptance	e Filter 1 Stand	dard Ident	ifier <10:0>				—	
C1RXF1EIDH	030A	—		_	_				Rece	ive Acceptanc	e Filter 1	Extended I	dentifier <	17:6>			
C1RXF1EIDL	030C	Rece	ive Accepta	nce Filter 1 E	xtended Id	entifier <5:0	)>	—	—	—	—	—	—	—	—	—	
C1RXF2SID	0310	—	—	—				Receive A	cceptance	e Filter 2 Stand	dard Ident	ifier <10:0>				—	
C1RXF2EIDH	0312	—	—	—	—				Rece	ive Acceptanc	e Filter 2	Extended I	dentifier <	17:6>			
C1RXF2EIDL	0314	Rece	ive Accepta	nce Filter 2 E	xtended Id	entifier <5:0	)>	—	_	_	_	_	_	_	—	_	L
C1RXF3SID	0318	—	—	—		_		Receive A	cceptance	e Filter 3 Stand	dard Ident	ifier <10:0>	•			-	
C1RXF3EIDH	031A	—	-	—	—				Rece	ive Acceptanc	e Filter 3	Extended I	dentifier <	17:6>			
C1RXF3EIDL	031C	Rece	ive Accepta	nce Filter 3 E	xtended Id	entifier <5:0	)>	—	—	—	—	—	—	—	—		L
C1RXF4SID	0320	—	—	_		-		Receive A	cceptance	e Filter 4 Stand	dard Ident	ifier <10:0>				—	
C1RXF4EIDH	0322	—		-	_				Rece	ive Acceptanc	e Filter 4	Extended I	dentifier <	17:6>			
C1RXF4EIDL	0324	Rece	eive Accepta	nce Filter 4 E	xtended Id	entifier <5:0	)>	—	—	_	_	_	—	_	—	_	L
C1RXF5SID	0328	—	—	—	Receive Acceptance Filter 5 Standard Identifier <10:0>										-		
C1RXF5EIDH	032A	_			—			-	Rece	ive Acceptanc	e Filter 5	Extended I	dentifier <	17:6>			
C1RXF5EIDL	032C	Rece	vive Accepta	nce Filter 5 E	xtended Id	entifier <5:0	)>	—	—	—	—	—	—	—	—	—	Ĺ
C1RXM0SID	0330	—	—	—				Receive A	cceptance	e Mask 0 Stan	dard Iden	tifier <10:0>	>			-	
C1RXM0EIDH	0332	—	-	—	—			_	Rece	ive Acceptanc	e Mask 0	Extended I	dentifier <	17:6>			
C1RXM0EIDL	0334	Rece	ive Accepta	nce Mask 0 E	xtended Id	entifier <5:0	)>	—	—	—	—	—	—	—	—	_	L
C1RXM1SID	0338	—	—	_				Receive A	cceptance	e Mask 1 Stan	dard Iden	tifier <10:0>	>			—	
C1RXM1EIDH	033A	—		_	_				Rece	ive Acceptanc	e Mask 1	Extended I	dentifier <	17:6>			
C1RXM1EIDL	033C	Rece	ive Acceptar	nce Mask 1 E	Extended Id	entifier <5:0	)>	—	—	—	—	—	—	—	—	—	
C1TX2SID	0340	Tran	smit Buffer 2	2 Standard Id	entifier <10	:6>	—			TI	ransmit B	uffer 2 Stan	dard Ident	ifier <5:0>		SRR	L
C1TX2EID	0342	Transmit B	uffer 2 Exten	ided Identifie	r <17:14>	—	—	—	—		Tra	ansmit Buffe	er 2 Extend	ded Identifier	<13:6>		
C1TX2DLC	0344		Transmit Bu	Iffer 2 Extend	led Identifie	r <5:0>		TXRTR	TXRB1	TXRB0		DL	C<3:0>		—		
C1TX2B1	0346			Trans	smit Buffer 2	2 Byte 1						Trar	nsmit Buffe	er 2 Byte 0			
C1TX2B2	0348			Trans	smit Buffer 2	2 Byte 3						Trar	nsmit Buffe	er 2 Byte 2			
C1TX2B3	034A			Trans	smit Buffer 2	2 Byte 5						Trar	nsmit Buffe	er 2 Byte 4			
C1TX2B4	034C			Trans	smit Buffer 2	2 Byte 7						Trar	nsmit Buffe	er 2 Byte 6			
C1TX2CON	034E	_	_	_	—	_	_	—		_	TXABT	TXLARB	TXERR	TXREQ	—	TXPR	ti T
C1TX1SID	0350	Tran	smit Buffer 1	Standard Id	entifier <10	:6>	-	-	—	Т	ransmit B	uffer 1 Stan	dard Ident	ifier <5:0>		SRR	L
C1TX1EID	0352	Transmit B	uffer 1 Exten	ided Identifie	r <17:14>	—	-	—	—		Tra	ansmit Buffe	er 1 Extend	ded Identifier	<13:6>		F
C1TX1DLC	0354		Transmit Bu	uffer 1 Extend	led Identifie	r <5:0>		TXRTR	TXRB1	TXRB0		DL	C<3:0>		—	—	

**Legend:** u = uninitialized bit

**Note:** Refer to "*dsPIC30F Family Reference Manual*" (DS70046) for descriptions of register bit fields.

#### TABLE 19-1: CAN1 REGISTER MAP (CONTINUED)

IADEE I	<i>J</i> -1.	UANTI			(0011		7	-		-	_		_	_									
SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 3         Bit 2         Bit 1           3vte 0         3vte 0         3vte 0								
C1TX1B1	0356			Trans	smit Buffer	1 Byte 1						Trar	nsmit Buffe	er 1 Byte 0			-						
C1TX1B2	0358			Trans	smit Buffer	1 Byte 3						Trar	nsmit Buffe	er 1 Byte 2									
C1TX1B3	035A			Trans	smit Buffer	1 Byte 5						Trar	nsmit Buffe	er 1 Byte 4									
C1TX1B4	035C			Trans	smit Buffer	1 Byte 7						Trar	nsmit Buffe	er 1 Byte 6									
C1TX1CON	035E	—	—	—	—	-	—	—	—	—	TXABT	TXLARB	TXERR	TXREQ	—	TXPF	ł١						
C1TX0SID	0360	Tran	smit Buffer (	) Standard Id	entifier <10	:6>	—	_	_	Т	ransmit Bu	uffer 0 Stan	dard Ident	ifier <5:0>		SRR							
C1TX0EID	0362	Transmit B	uffer 0 Exter	nded Identifie	r <17:14>	—	—	_	—		Tra	insmit Buffe	er 0 Extend	ded Identifier	<13:6>								
C1TX0DLC	0364		Transmit Bu	uffer 0 Extend	led Identifie	er <5:0>		TXRTR	TXRB1	TXRB0		DL	C<3:0>		—	—							
C1TX0B1	0366			Trans	smit Buffer	0 Byte 1						Trar	nsmit Buffe	er 0 Byte 0									
C1TX0B2	0368			Trans	smit Buffer	0 Byte 3						Trar	nsmit Buffe	er 0 Byte 2									
C1TX0B3	036A			Trans	smit Buffer	0 Byte 5						Trar	nsmit Buffe	er 0 Byte 4									
C1TX0B4	036C			Trans	smit Buffer	0 Byte 7					-	Trar	nsmit Buffe	er 0 Byte 6									
C1TX0CON	036E	—	—	—	_	—	—	_	—	—	TXABT	TXLARB	TXERR	TXREQ	—	TXPF	ł١						
C1RX1SID	0370	—	—	-	Receive Buffer 1 Standard Identifier <10:0>									SRR									
C1RX1EID	0372	—	—	—	—			-	-	Receive Buffe	e Buffer 1 Extended Identifier <17:6>												
C1RX1DLC	0374		Receive Bu	uffer 1 Extend	r 1 Extended Identifier <5:0> RXRTR RXRB1 — — RXRB0 DLC<3:0									3:0>									
C1RX1B1	0376			Rece	eive Buffer	1 Byte 1		-	-		-	Rec	eive Buffe	er 1 Byte 0									
C1RX1B2	0378			Rece	eive Buffer	1 Byte 3						Rec	eive Buffe	er 1 Byte 2									
C1RX1B3	037A			Rece	eive Buffer	1 Byte 5						Rec	eive Buffe	er 1 Byte 4									
C1RX1B4	037C			Rece	eive Buffer	1 Byte 7						Rec	eive Buffe	er 1 Byte 6									
C1RX1CON	037E	—	—	—		—	—	_	_	RXFUL	_	_	_	RXRTRRO	F	-ILHIT<2	:0						
C1RX0SID	0380	—	—	—				Rece	eive Buffe	r 0 Standard Io	dentifier <	10:0>				SRR							
C1RX0EID	0382	—	—	—	—			_	_	Receive Buffe	er 0 Exten	ded Identifi	er <17:6>	_									
C1RX0DLC	0384		Receive Bu	uffer 0 Extend	ed Identifie	er <5:0>		RXRTR	RXRB1	—	—	—	RXRB0		DLC<3	3:0>							
C1RX0B1	0386			Rece	eive Buffer (	0 Byte 1						Rec	eive Buffe	er 0 Byte 0									
C1RX0B2	0388			Rece	eive Buffer (	0 Byte 3						Rec	eive Buffe	er 0 Byte 2									
C1RX0B3	038A			Rece	eive Buffer (	0 Byte 5						Rec	eive Buffe	er 0 Byte 4									
C1RX0B4	038C			Rece	eive Buffer (	0 Byte 7						Rec	eive Buffe	er 0 Byte 6									
C1RX0CON	038E	_	—	—		—	—	_	_	RXFUL	_	_	_	- RXRTRRO DBEN JTO									
C1CTRL	0390	CANCAP	—	CSIDLE	ABAT	CANCKS	R	REQOP<2:0> OPMODE<2:0> — ICODE<2:0>							>								
C1CFG1	0392	—	—																				
C1CFG2	0394	-	WAKFIL	-         -         SEG2PH<2:0>         SEG2PHTS         SAM         SEG1PH<2:0>								P	RSEG<2	2:(									
C1INTF	0396	RX00VR	RX10VR	ТХВО	TXEP	RXEP	TXWAR	RXWAR	EWARN	IVRIF	WAKIF	ERRIF	TX2IF	TX2IF TX1IF TX0IF RX1									
C1INTE	0398	_	—	—	—	_	_	_	_	IVRIE	WAKIE	ERRIE	TX2IE	TX1IE	TX0IE	RX1E	Γ						
C1EC	039A			Transmi	t Error Cou	nt Register						Receiv	e Error Co	ount Register									

**Legend:** u = uninitialized bit

Note: Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

## TABLE 19-2: CAN2 REGISTER MAP

-		-														-	
SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	
C2RXF0SID	03C0	—	—	—				Receive A	cceptance	Filter 0 Stand	lard Identifi	er <10:0>				-	
C2RXF0EIDH	03C2	_	_	_	_				Receiv	ve Acceptance	Filter 0 Ex	tended Ide	ntifier <17	7:6>			
C2RXF0EIDL	03C4	Rece	eive Acceptan	ce Filter 0 E	3         Bit 12         Bit 11         Bit 0         Bit 8         Bit 7         Bit 6         Bit 5         Bit 4         Bit 3         Bit 2         Bit 3												
C2RXF1SID	03C8	—	_	-	Bit 12         Bit 11         Bit 10         Bit 9         Bit 8         Bit 7         Bit 6         Bit 5         Bit 4         Bit 3         Bit 2         Bit 4												
C2RXF1EIDH	03CA	_	_	_	—				Receiv	ve Acceptance	Filter 1 Ex	tended Ide	ntifier <17	7:6>			
C2RXF1EIDL	03CC	Rece	eive Acceptan	ce Filter 1 E	xtended Id	entifier <5:	0>	_	—	_	_	_	—	_	_	—	
C2RXF2SID	03D0	—	_	-				Receive A	cceptance	Filter 2 Stand	lard Identifi	er <10:0>		-		—	
C2RXF2EIDH	03D2	_	_	—	_				Receiv	ve Acceptance	Filter 2 Ex	tended Ide	ntifier <17	/:6>		-	
C2RXF2EIDL	03D4	Rece	eive Acceptan	ce Filter 2 E	xtended Id	entifier <5:	0>	—	—	_	_	—	—	_	-	_	
C2RXF3SID	03D8	—	_	-				Receive A	cceptance	Filter 3 Stand	lard Identifi	er <10:0>	-			—	
C2RXF3EIDH	03DA	_	_	_	—				Receiv	ve Acceptance	Filter 3 Ex	tended Ide	ntifier <17	7:6>			
C2RXF3EIDL	03DC	Rece	eive Acceptan	ce Filter 3 E	xtended Id	entifier <5:	0>	_	—	_	_	_	—	_	_	_	
C2RXF4SID	03E0	—	—	-				Receive A	cceptance	Filter 4 Stand	lard Identifi	er <10:0>	-			—	
C2RXF4EIDH	03E2	_	_	_	_				Receiv	ve Acceptance	Filter 4 Ex	tended Ide	ntifier <17	7:6>			
C2RXF4EIDL	03E4	Rece	vive Acceptan	ce Filter 4 E	xtended Id	entifier <5:	0>	_	—	_	_	_	—	_	-	_	
C2RXF5SID	03E8	—	_	—	- Receive Acceptance Filter 5 Standard Identifier <10:0>											—	
C2RXF5EIDH	03EA	_	_	_	Receive Acceptance Filter 5 Extended Identifier <17:6>												
C2RXF5EIDL	03EC	Rece	vive Acceptan	ce Filter 5 E	—         —         Receive Acceptance Filter 5 Extended Identifier <17:6>           Filter 5 Extended Identifier <5:0>         —         .         .         .         .         . <td< td=""><td>_</td></td<>											_	
C2RXM0SID	03F0	_	_	—				Receive A	cceptance	Mask 0 Stand	lard Identif	ier <10:0>				—	
C2RXM0EIDH	03F2	_	_	_	_				Receiv	ve Acceptance	Mask 0 E	xtended Ide	ntifier <17	7:6>			
C2RXM0EIDL	03F4	Rece	ive Acceptance	ce Mask 0 E	xtended Ic	lentifier <5:	:0>	—	-	—	_	—	—	-	_	_	
C2RXM1SID	03F8	—	—	-				Receive A	cceptance	Mask 1 Stand	lard Identif	ier <10:0>				—	
C2RXM1EIDH	03FA		_	—	_				Receiv	e Acceptance	Mask 1 E	xtended Ide	ntifier <17	7:6>			
C2RXM1EIDL	03FC	Rece	ive Acceptance	ce Mask 1 E	xtended Ic	lentifier <5:	:0>	—	—	—	—	—	—	-	-	—	
C2TX2SID	0400	Trans	smit Buffer 2 S	Standard Ide	ntifier <10	:6>	—	—	—	Т	ransmit Bu	ffer 2 Stand	lard Identi	ifier <5:0>		SRR	
C2TX2EID	0402	Transmit Bu	uffer 2 Extend	ed Identifier	<17:14>	—	—	—	—		Trai	nsmit Buffei	2 Extend	led Identifier	<13:6>		
C2TX2DLC	0404		Transmit Buff	er 2 Extend	ed Identifie	er <5:0>		TXRTR	TXRB1	TXRB0		DLC	<3:0>		—	-	
C2TX2B1	0406			Tran	smit Buffe	r 2 Byte 1						Trans	smit Buffe	er 2 Byte 0			
C2TX2B2	0408			Tran	smit Buffe	r 2 Byte 3						Trans	smit Buffe	r 2 Byte 2			
C2TX2B3	040A			Tran	smit Buffe	r 2 Byte 5						Trans	smit Buffe	r 2 Byte 4			
C2TX2B4	040C			Tran	smit Buffe	r 2 Byte 7		1				Trans	smit Buffe	r 2 Byte 6			
C2TX2CON	040E	—	—	—	—	—		—		_	TXABT	TXLARB	TXERR	TXREQ	-	TXF	
C2TX1SID	0410	Trans	smit Buffer 1 S	Standard Ide	ntifier <10	:6>				Т	ransmit Bu	ffer 1 Stand	lard Identi	ifier <5:0>		SRR	
C2TX1EID	0412	Transmit Bu	uffer 1 Extend	ed Identifier	<17:14>	—	-	—			Trai	nsmit Buffer	1 Extend	led Identifier	<13:6>		
C2TX1DLC	0414		Transmit Buff	er 1 Extend	ed Identifie	er <5:0>		TXRTR	TXRB1	TXRB0		DLC	<3:0>		—	-	
C2TX1B1	0416			Tran	smit Buffe	r 1 Byte 1						Trans	smit Buffe	er 1 Byte 0			
C2TX1B2	0418			Tran	smit Buffe	r 1 Byte 3						Trans	smit Buffe	er 1 Byte 2			
C2TX1B3	041A			Tran	smit Buffe	r 1 Byte 5						Trans	smit Buffe	r 1 Byte 4			

### TABLE 19-2: CAN2 REGISTER MAP (CONTINUED)

		•/ •/ •			(0011		- /			-			Bit 5 Bit 4 Bit 3 Bit 2 Bit							
SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1				
C2TX1B4	041C			Tran	smit Buffe	r 1 Byte 7						Trans	smit Buffe	r 1 Byte 6						
C2TX1CON	041E	_	_	_	_	—	—	—	-	_	TXABT	TXLARB	TXERR	TXREQ	_	TXP				
C2TX0SID	0420	Trans	smit Buffer 0 S	standard Ide	ntifier <10	:6>	—	—	-	Т	ransmit Bu	ffer 0 Stand	ard Ident	ifier <5:0>		SRR				
C2TX0EID	0422	Transmit Bu	uffer 0 Extende	ed Identifier	<17:14>	—	—	—	-		Trar	nsmit Buffer	0 Extend	led Identifier «	<13:6>					
C2TX0DLC	0424		Transmit Buff	er 0 Extend	ed Identifi	er <5:0>		TXRTR	TXRB1	TXRB0		DLC	<3:0>		_	_				
C2TX0B1	0426			Tran	smit Buffe	r 0 Byte 1			-			Trans	smit Buffe	er 0 Byte 0						
C2TX0B2	0428			Tran	smit Buffe	r 0 Byte 3						Trans	smit Buffe	er 0 Byte 2						
C2TX0B3	042A			Tran	smit Buffe	r 0 Byte 5						Trans	smit Buffe	er 0 Byte 4						
C2TX0B4	042C			Tran	smit Buffe	r 0 Byte 7						Trans	smit Buffe	er 0 Byte 6						
C2TX0CON	042E	—	—	—	—	—	—	—	—	—	TXABT	TXLARB	TXERR	TXREQ	—	TXF				
C2RX1SID	0430	_	—	_				Rec	eive Buffer	1 Standard Id	entifier <10	):0>				SRR				
C2RX1EID	0432	_	—	_	_				I	Receive Buffer	1 Extende	ed Identifier	<17:6>							
C2RX1DLC	0434		Receive Buff	er 1 Extende	ed Identifie	er <5:0>		RXRTR	RXRB1	—	-	—	RXRB0	KRB0 DLC<3:0						
C2RX1B1	0436			Rece	eive Buffer	1 Byte 1				Receive Buffer 1 Byte 0										
C2RX1B2	0438			Rece	eive Buffer	1 Byte 3				Receive Buffer 1 Byte 2										
C2RX1B3	043A			Rece	eive Buffer	1 Byte 5						Rece	eive Buffe	r 1 Byte 4						
C2RX1B4	043C		_	Rece	eive Buffer	1 Byte 7	_					Rece	eive Buffe	r 1 Byte 6						
C2RX1CON	043E	_	—	_	_	—	_	_	—	RXFUL	—	_	—	RXRTRRO	F	FILHIT<				
C2RX0SID	0440	_	—	—				Rec	eive Buffer	0 Standard Id	entifier <10	0:0>	-			SRR				
C2RX0EID	0442	_	—	_	_					Receive Buffer	0 Extende	d Identifier	<17:6>							
C2RX0DLC	0444		Receive Buff	er 0 Extende	ed Identifie	er <5:0>		RXRTR	RXRB1	—	_	—	RXRB0		DLC<	3:0>				
C2RX0B1	0446			Rece	eive Buffer	r 0 Byte 1			-		-	Rece	eive Buffe	r 0 Byte 0						
C2RX0B2	0448			Rece	eive Buffei	r 0 Byte 3						Rece	ive Buffe	r 0 Byte 2						
C2RX0B3	044A			Rece	eive Buffei	0 Byte 5						Rece	eive Buffe	r 0 Byte 4						
C2RX0B4	044C		-	Rece	eive Buffer	0 Byte 7						Rece	eive Buffe	r 0 Byte 6						
C2RX0CON	044E	—	—	—	—	_	_	_	—	RXFUL	_	_	—	RXRTRRO	DBEN	JTOFF				
C2CTRL	0450	CANCAP	—	CSIDLE	ABAT	CANCKS	F	REQOP<2:0	>	OPI	MODE<2:0	>	—	ICO	DE<2:0>	>				
C2CFG1	0452	_	—	_	—	_	_	_	—	SJW<1	:0>			BRP<5:0>						
C2CFG2	0454		WAKFIL	—	_	—	S	EG2PH<2:(	)>	SEG2PHTS	SAM	S	EG1PH<	2:0>	P	RSEG<				
C2INTF	0456	RX00VR	RX10VR	ТХВО	TXEP	RXEP	TXWAR	RXWAR	EWARN	IVRIF	WAKIF	ERRIF	TX2IF	TX1IF	TX0IF	RX1IF				
C2INTE	0458	—	—	—	_	—	_	—	_	IVRIE	WAKIE	ERRIE	TX2IE	TX1IE	TX0IE	RX1E				
C2EC	045A			Transm	it Error Co	unt Reaiste	er					Receive	Error Co	ount Register						

NOTES:

## 20.0 10-BIT HIGH-SPEED ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

**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).

The10-bit high-speed Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit digital number. This module is based on a Successive Approximation Register (SAR) architecture, and provides a maximum sampling rate of 1 Msps. The ADC module has 16 analog inputs which are multiplexed into four sample and hold amplifiers. The output of the sample and hold is the input into the converter, which generates the result. The analog reference voltages are software selectable to either the device supply voltage (AVDD/AVSS) or the voltage level on the (VREF+/VREF-) pin. The ADC has a unique feature of being able to operate while the device is in Sleep mode.

The ADC module has six 16-bit registers:

- ADC Control Register1 (ADCON1)
- ADC Control Register2 (ADCON2)
- ADC Control Register3 (ADCON3)
- ADC Input Select Register (ADCHS)
- ADC Port Configuration Register (ADPCFG)
- ADC Input Scan Selection Register (ADCSSL)

The ADCON1, ADCON2 and ADCON3 registers control the operation of the A/D module. The ADCHS register selects the input channels to be converted. The ADPCFG register configures the port pins as analog inputs or as digital I/O. The ADCSSL register selects inputs for scanning.

Note:	The	SSRC<2:0>,	ASAM,	SIMSAM,
	SMPI	<3:0>, BUFM a	nd ALTS b	oits, as well
	as th	e ADCON3 and	d ADCSSI	_ registers,
	must	not be written	to while A	ADON = 1.
	This <b>v</b>	would lead to in	determina	te results.

The block diagram of the ADC module is shown in Figure 20-1.



## 20.1 ADC Result Buffer

The module contains a 16-word dual port read-only buffer, called ADCBUF0...ADCBUFF, to buffer the A/D results. The RAM is 10 bits wide, but is read into different format 16-bit words. The contents of the sixteen ADC conversion result buffer registers, ADCBUF0 through ADCBUFF, cannot be written by user software.

### 20.2 Conversion Operation

After the ADC module has been configured, the sample acquisition is started by setting the SAMP bit. Various sources, such as a programmable bit, timer time-outs and external events, will terminate acquisition and start a conversion. When the A/D conversion is complete, the result is loaded into ADCBUF0...ADCBUFF, and the ADC interrupt flag ADIF and the DONE bit are set after the number of samples specified by the SMPI bit.

The following steps should be followed for doing an ADC conversion:

- 1. Configure the ADC module:
  - Configure analog pins, voltage reference and digital I/O
  - Select ADC input channels
  - Select ADC conversion clock
  - Select ADC conversion trigger
  - Turn on ADC module
- 2. Configure ADC interrupt (if required):
  - Clear ADIF bit
- Select A/D interrupt priority
- 3. Start sampling.
- 4. Wait the required acquisition time.
- 5. Trigger acquisition end, start conversion
- 6. Wait for ADC conversion to complete, by either:
- Waiting for the ADC interrupt
- Waiting for the DONE bit to get set
- 7. Read A/D result buffer, clear ADIF if required.

## 20.3 Selecting the Conversion Sequence

Several groups of control bits select the sequence in which the ADC connects inputs to the sample/hold channels, converts channels, writes the buffer memory, and generates interrupts. The sequence is controlled by the sampling clocks.

The SIMSAM bit controls the acquire/convert sequence for multiple channels. If the SIMSAM bit is '0', the two or four selected channels are acquired and converted sequentially, with two or four sample clocks. If the SIMSAM bit is '1', two or four selected channels are acquired simultaneously, with one sample clock. The channels are then converted sequentially. Obviously, if there is only 1 channel selected, the SIMSAM bit is not applicable.

The CHPS bits selects how many channels are sampled. This can vary from 1, 2 or 4 channels. If CHPS selects 1 channel, the CH0 channel will be sampled at the sample clock and converted. The result is stored in the buffer. If CHPS selects 2 channels, the CH0 and CH1 channels will be sampled and converted. If CHPS selects 4 channels, the CH0, CH1, CH2 and CH3 channels will be sampled and converted.

The SMPI bits select the number of acquisition/conversion sequences that would be performed before an interrupt occurs. This can vary from 1 sample per interrupt to 16 samples per interrupt.

The user cannot program a combination of CHPS and SMPI bits that specifies more than 16 conversions per interrupt, or 8 conversions per interrupt, depending on the BUFM bit. The BUFM bit, when set, will split the 16-word results buffer (ADCBUF0...ADCBUFF) into two 8-word groups. Writing to the 8-word buffers will be alternated on each interrupt event. Use of the BUFM bit will depend on how much time is available for moving data out of the buffers after the interrupt, as determined by the application.

If the processor can quickly unload a full buffer within the time it takes to acquire and convert one channel, the BUFM bit can be '0' and up to 16 conversions may be done per interrupt. The processor will have one sample and conversion time to move the sixteen conversions.

If the processor cannot unload the buffer within the acquisition and conversion time, the BUFM bit should be '1'. For example, if SMPI<3:0> (ADCON2<5:2>) = 0111, then eight conversions will be loaded into 1/2 of the buffer, following which an interrupt occurs. The next eight conversions will be loaded into the other 1/2 of the buffer. The processor will have the entire time between interrupts to move the eight conversions.

The ALTS bit can be used to alternate the inputs selected during the sampling sequence. The input multiplexer has two sets of sample inputs: MUX A and MUX B. If the ALTS bit is '0', only the MUX A inputs are selected for sampling. If the ALTS bit is '1' and SMPI<3:0> = 0000, on the first sample/convert sequence, the MUX A inputs are selected, and on the next acquire/convert sequence, the MUX B inputs are selected.

The CSCNA bit (ADCON2<10>) will allow the CH0 channel inputs to be alternately scanned across a selected number of analog inputs for the MUX A group. The inputs are selected by the ADCSSL register. If a particular bit in the ADCSSL register is '1', the corresponding input is selected. The inputs are always scanned from lower to higher numbered inputs, starting after each interrupt. If the number of inputs selected is greater than the number of samples taken per interrupt, the higher numbered inputs are unused.

#### 20.4 Programming the Start of Conversion Trigger

The conversion trigger will terminate acquisition and start the requested conversions.

The SSRC<2:0> bits select the source of the conversion trigger.

The SSRC bits provide for up to 5 alternate sources of conversion trigger.

When SSRC<2:0> = 000, the conversion trigger is under software control. Clearing the SAMP bit will cause the conversion trigger.

When SSRC<2:0> = 111 (Auto-Start mode), the conversion trigger is under ADC clock control. The SAMC bits select the number of ADC clocks between the start of acquisition and the start of conversion. This provides the fastest conversion rates on multiple channels. SAMC must always be at least 1 clock cycle.

Other trigger sources can come from timer modules, Motor Control PWM module, or external interrupts.

Note: To operate the ADC at the maximum specified conversion speed, the Auto Convert Trigger option should be selected (SSRC = 111) and the Auto Sample Time bits should be set to 1 TAD (SAMC = 00001). This configuration will give a total conversion period (sample + convert) of 13 TAD.

> The use of any other conversion trigger will result in additional TAD cycles to synchronize the external event to the ADC.

## 20.5 Aborting a Conversion

Clearing the ADON bit during a conversion will abort the current conversion and stop the sampling sequencing. The ADCBUF will not be updated with the partially completed A/D conversion sample. That is, the ADCBUF will continue to contain the value of the last completed conversion (or the last value written to the ADCBUF register).

If the clearing of the ADON bit coincides with an auto start, the clearing has a higher priority.

After the A/D conversion is aborted, a 2 TAD wait is required before the next sampling may be started by setting the SAMP bit.

If sequential sampling is specified, the A/D will continue at the next sample pulse which corresponds with the next channel converted. If simultaneous sampling is specified, the ADC will continue with the next multi-channel group conversion sequence.

### 20.6 Selecting the A/D Conversion Clock

The A/D conversion requires 12 TAD. The source of the A/D conversion clock is software selected using a six bit counter. There are 64 possible options for TAD.

#### EQUATION 20-1: A/D CONVERSION CLOCK

TAD = TCY \* 
$$(0.5 * (ADCS < 5:0 > + 1))$$
  
ADCS < 5:0 > = 2  $\frac{TAD}{TCY} - 1$ 

The internal RC oscillator is selected by setting the ADRC bit.

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 83.33 nsec (for VDD = 5V). Refer to the **Section 24.0 "Electrical Characteristics"** for minimum TAD under other operating conditions.

Example 20-1 shows a sample calculation for the ADCS<5:0> bits, assuming a device operating speed of 30 MIPS.

## EXAMPLE 20-1: A/D CONVERSION CLOCK CALCULATION

TAD = 154 nsec  
TCY = 33 nsec (30 MIPS)  
ADCS<5:0> = 2 
$$\frac{TAD}{TCY} - 1$$
  
= 2  $\cdot \frac{154 \text{ nsec}}{33 \text{ nsec}} - 1$   
= 8.33  
Therefore,  
Set ADCS<5:0> = 9  
Actual TAD =  $\frac{TCY}{2}$  (ADCS<5:0> + 1)  
=  $\frac{33 \text{ nsec}}{2}$  (9 + 1)  
= 165 nsec

## 20.7 A/D Conversion Speeds

The dsPIC30F 10-bit ADC specifications permit a maximum 1 Msps sampling rate. Table 20-1 summarizes the conversion speeds for the dsPIC30F 10-bit ADC and the required operating conditions.

|--|

			dsPIC30F	10-bit ADC Con	version Rates	
ADC Speed	TAD Minimum	Sampling Time Min	Rs Max	Vdd	Temperature	A/D Channels Configuration
Up to 1 Msps <sup>(1)</sup>	83.33 ns	12 Tad	500Ω	4.5V to 5.5V	-40°C to +85°C	ANX CHI, CH2 or CH3 SH CH0 SH CH0 SH
Up to 750 ksps <sup>(1)</sup>	95.24 ns	2 Tad	500Ω	4.5V to 5.5V	-40°C to +85°C	ANX CHX ADC
Up to 600 ksps <sup>(1)</sup>	138.89 ns	12 Tad	500Ω	3.0V to 5.5V	-40°C to +125°C	ANX CHI, CH2 or CH3 SH GH0 SH CH0 SH
Up to 500 ksps	153.85 ns	1 Tad	5.0 kΩ	4.5V to 5.5V	-40°C to +125°C	ANX ANX O'VREF-
Up to 300 ksps	256.41 ns	1 Tad	5.0 kΩ	3.0V to 5.5V	-40°C to +125°C	ANX ADC ANX O'VREF-

Note 1: External VREF- and VREF+ pins must be used for correct operation. See Figure 20-2 for recommended circuit.

The configuration guidelines give the required setup values for the conversion speeds above 500 ksps, since they require external VREF pins usage and there are some differences in the configuration procedure. Configuration details that are not critical to the conversion speed have been omitted.

The following figure depicts the recommended circuit for the conversion rates above 500 ksps.





#### 20.7.1 1 Msps CONFIGURATION GUIDELINE

The configuration for 1 Msps operation is dependent on whether a single input pin is to be sampled or whether multiple pins will be sampled.

#### 20.7.1.1 Single Analog Input

For conversions at 1 Msps for a single analog input, at least two sample and hold channels must be enabled. The analog input multiplexer must be configured so that the same input pin is connected to both sample and hold channels. The ADC converts the value held on one S/H channel, while the second S/H channel acquires a new input sample.

#### 20.7.1.2 Multiple Analog Inputs

The ADC can also be used to sample multiple analog inputs using multiple sample and hold channels. In this case, the total 1 Msps conversion rate is divided among the different input signals. For example, four inputs can be sampled at a rate of 250 ksps for each signal or two inputs could be sampled at a rate of 500 ksps for each signal. Sequential sampling must be used in this configuration to allow adequate sampling time on each input.

#### 20.7.1.3 1 Msps Configuration Items

The following configuration items are required to achieve a 1 Msps conversion rate.

- Comply with conditions provided in Table 20-2
- Connect external VREF+ and VREF- pins following the recommended circuit shown in Table 20-2
- Set SSRC<2:0> = 111 in the ADCON1 register to enable the auto-convert option
- Enable automatic sampling by setting the ASAM control bit in the ADCON1 register
- Enable sequential sampling by clearing the SIMSAM bit in the ADCON1 register
- Enable at least two sample and hold channels by writing the CHPS<1:0> control bits in the

ADCON2 register

- Write the SMPI<3:0> control bits in the ADCON2 register for the desired number of conversions between interrupts. At a minimum, set SMPI<3:0> = 0001 since at least two sample and hold channels should be enabled
- Configure the A/D clock period to be:

= 83.33 ns

by writing to the ADCS<5:0> control bits in the ADCON3 register

- Configure the sampling time to be 2 TAD by writing: SAMC<4:0> = 00010
- Select at least two channels per analog input pin by writing to the ADCHS register

#### 20.7.2 750 ksps CONFIGURATION GUIDELINE

The following configuration items are required to achieve a 750 ksps conversion rate. This configuration assumes that a single analog input is to be sampled.

- Comply with conditions provided in Table 20-2
- Connect external VREF+ and VREF- pins following the recommended circuit shown in Figure 20-2
- Set SSRC<2:0> = 111 in the ADCON1 register to enable the auto-convert option
- Enable automatic sampling by setting the ASAM control bit in the ADCON1 register
- Enable one sample and hold channel by setting CHPS<1:0> = 00 in the ADCON2 register
- Write the SMPI<3:0> control bits in the ADCON2 register for the desired number of conversions between interrupts
- Configure the A/D clock period to be:

$$(12+2) \ge 750,000 = 95.24 \text{ ns}$$

by writing to the ADCS<5:0> control bits in the ADCON3 register

• Configure the sampling time to be 2 TAD by writing: SAMC<4:0> = 00010

#### 20.7.3 600 ksps CONFIGURATION GUIDELINE

The configuration for 600 ksps operation is dependent on whether a single input pin is to be sampled or whether multiple pins will be sampled.

#### 20.7.3.1 Single Analog Input

When performing conversions at 600 ksps for a single analog input, at least two sample and hold channels must be enabled. The analog input multiplexer must be configured so that the same input pin is connected to both sample and hold channels. The A/D converts the value held on one S/H channel, while the second S/H channel acquires a new input sample.

#### 20.7.3.2 Multiple Analog Input

The ADC can also be used to sample multiple analog inputs using multiple sample and hold channels. In this case, the total 600 ksps conversion rate is divided among the different input signals. For example, four inputs can be sampled at a rate of 150 ksps for each signal or two inputs can be sampled at a rate of 300 ksps for each signal. Sequential sampling must be used in this configuration to allow adequate sampling time on each input.

#### 20.7.3.3 600 ksps Configuration Items

The following configuration items are required to achieve a 600 ksps conversion rate.

- Comply with conditions provided in Table 20-2
- Connect external VREF+ and VREF- pins following the recommended circuit shown in Figure 20-2
- Set SSRC<2:0> = 111 in the ADCON1 register to enable the auto-convert option
- Enable automatic sampling by setting the ASAM control bit in the ADCON1 register
- Enable sequential sampling by clearing the SIMSAM bit in the ADCON1 register
- Enable at least two sample and hold channels by writing the CHPS<1:0> control bits in the ADCON2 register
- Write the SMPI<3:0> control bits in the ADCON2 register for the desired number of conversions between interrupts. At a minimum, set SMPI<3:0> = 0001 since at least two sample and hold channels should be enabled
- Configure the A/D clock period to be:

= 138.89 ns

by writing to the ADCS<5:0> control bits in the ADCON3 register

• Configure the sampling time to be 2 TAD by writing: SAMC<4:0> = 00010

Select at least two channels per analog input pin by writing to the ADCHS register

## 20.8 ADC Acquisition Requirements

The analog input model of the 10-bit ADC is shown in Figure 20-3. The total sampling time for the ADC is a function of the internal amplifier settling time, device VDD and the holding capacitor charge time.

For the ADC to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the voltage level on the analog input pin. The source impedance (Rs), the interconnect impedance (RIC), and the internal sampling switch (Rss) impedance combine to directly affect the time required to charge the capacitor CHOLD. The combined impedance of the analog sources must therefore be small enough to fully charge the holding capacitor within the chosen sample time. To minimize the effects of pin leakage currents on the accuracy of the A/D converter, the maximum recommended source impedance, Rs, is 5 k $\Omega$ . After the analog input channel is selected (changed), this sampling function must be completed prior to starting the conversion. The internal holding capacitor will be in a discharged state prior to each sample operation.

The user must allow at least 1 TAD period of sampling time, TSAMP, between conversions to allow each sample to be acquired. This sample time may be controlled manually in software by setting/clearing the SAMP bit, or it may be automatically controlled by the ADC. In an automatic configuration, the user must allow enough time between conversion triggers so that the minimum sample time can be satisfied. Refer to the Electrical Specifications for TAD and sample time requirements.

#### FIGURE 20-3: ADC ANALOG INPUT MODEL



#### 20.9 Module Power-Down Modes

The module has 3 internal power modes. When the ADON bit is '1', the module is in Active mode; it is fully powered and functional. When ADON is '0', the module is in Off mode. The digital and analog portions of the circuit are disabled for maximum current savings. In order to return to the Active mode from Off mode, the user must wait for the ADC circuitry to stabilize.

#### 20.10 ADC Operation During CPU Sleep and Idle Modes

#### 20.10.1 ADC OPERATION DURING CPU SLEEP MODE

When the device enters Sleep mode, all clock sources to the module are shutdown and stay at logic '0'.

If Sleep occurs in the middle of a conversion, the conversion is aborted. The converter will not continue with a partially completed conversion on exit from Sleep mode.

Register contents are not affected by the device entering or leaving Sleep mode.

The ADC module can operate during Sleep mode if the ADC clock source is set to RC (ADRC = 1). When the RC clock source is selected, the ADC module waits one instruction cycle before starting the conversion. This allows the SLEEP instruction to be executed, which eliminates all digital switching noise from the conversion. When the conversion is complete, the DONE bit will be set and the result loaded into the ADCBUF register.

If the ADC interrupt is enabled, the device will wake-up from Sleep. If the ADC interrupt is not enabled, the ADC module will then be turned off, although the ADON bit will remain set.

## 20.10.2 A/D OPERATION DURING CPU IDLE MODE

The ADSIDL bit selects if the module will stop on Idle or continue on Idle. If ADSIDL = 0, the module will continue operation on assertion of Idle mode. If ADSIDL = 1, the module will stop on Idle.

#### 20.11 Effects of a Reset

A device Reset forces all registers to their Reset state. This forces the ADC module to be turned off, and any conversion and acquisition sequence is aborted. The values that are in the ADCBUF registers are not modified. The ADC result register will contain unknown data after a Power-on Reset.

#### 20.12 Output Formats

The ADC result is 10 bits wide. The data buffer RAM is also 10 bits wide. The 10-bit data can be read in one of four different formats. The FORM<1:0> bits select the format. Each of the output formats translates to a 16-bit result on the data bus.

Write data will always be in right justified (integer) format.

RAM Contents:						d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
								<u> </u>							
Read to Bus:															
Signed Fractional (1.15)	d09 d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0	0	0
Fractional (1.15)	d09 d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0	0	0
		I													
Signed Integer	d09 d09	d09	d09	d09	d09	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
								1							
Integer	0 0	0	0	0	0	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
	<u>II</u>	1						1							1]

#### FIGURE 20-4: ADC OUTPUT DATA FORMATS

## 20.13 Configuring Analog Port Pins

The use of the ADPCFG and TRIS registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bit set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CH0SA<3:0>/CH0SB<3:0> bits and the TRIS bits.

When reading the PORT register, all pins configured as analog input channels will read as cleared.

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

## 20.14 Connection Considerations

The analog inputs have diodes to VDD and VSS as ESD protection. This requires that the analog input be between VDD and VSS. If the input voltage exceeds this range by greater than 0.3V (either direction), one of the diodes becomes forward biased and it may damage the device if the input current specification is exceeded.

An external RC filter is sometimes added for antialiasing of the input signal. The R component should be selected to ensure that the sampling time requirements are satisfied. Any external components connected (via high impedance) to an analog input pin (capacitor, zener diode, etc.) should have very little leakage current at the pin.

## TABLE 20-2: ADC REGISTER MAP

										_							
SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	
ADCBUF0	0280	_	—	_	_	_	_					ADC Data	Buffer 0				
ADCBUF1	0282	_	_	_	_	_	_					ADC Data	Buffer 1				
ADCBUF2	0284	_	_	_	_	_	_					ADC Data	Buffer 2				
ADCBUF3	0286	_	_	_	_	_	_					ADC Data	Buffer 3				
ADCBUF4	0288	_	_	_	_	_	_					ADC Data	Buffer 4				
ADCBUF5	028A	—	_	_	_	_	_					ADC Data	Buffer 5				
ADCBUF6	028C	_	_	_	_	_	_					ADC Data	Buffer 6				
ADCBUF7	028E	_	_	_	_	_	_					ADC Data	Buffer 7				
ADCBUF8	0290	_	_	_	_	_	_	ADC Data Buffer 8									
ADCBUF9	0292	_	_	_	_	_	_	ADC Data Buffer 9									
ADCBUFA	0294	_	_	_	_	_	_					ADC Data	Buffer 10				
ADCBUFB	0296	_	_	_	—	_	_					ADC Data	Buffer 11				
ADCBUFC	0298	_	_	_	_	_	_					ADC Data	Buffer 12				
ADCBUFD	029A	_	_	_	—	_	—					ADC Data	Buffer 13				
ADCBUFE	029C	_	_	_	—	_	_					ADC Data	Buffer 14				
ADCBUFF	029E	_	_	_	_	_	_			-		ADC Data	Buffer 15				
ADCON1	02A0	ADON	_	ADSIDL	—	_	_	FORM	<b>l</b> <1:0>		SSRC<2	0>	_	SIMSAM	ASAM	SAMP	
ADCON2	02A2		VCFG<2:0	>	_	_	CSCNA	CHPS	<1:0>	BUFS	_		SMPI<	<3:0>		BUFM	
ADCON3	02A4	_	_	_		S	AMC<4:0>			ADRC	_			ADCS<	5:0>		
ADCHS	02A6	CH1231	NB<1:0>	CH123SB	CH0NB		CH0SB	<3:0>		CH123	NA<1:0>	CH123SA	CHONA		CHOSA	<3:0>	
ADPCFG	02A8	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	F
ADCSSL	02AA	CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8	CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	(

**Legend:** u = uninitialized bit

Note: Refer to "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

NOTES:

## 21.0 SYSTEM INTEGRATION

**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 "*dsPIC30F/ 33F Programmer's Reference Manual*" (DS70157).

There are several features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection:

- Oscillator Selection
- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Programmable Brown-out Reset (BOR)
- Watchdog Timer (WDT)
- Power Saving modes (Sleep and Idle)
- Code Protection
- Unit ID Locations
- In-Circuit Serial Programming (ICSP)

dsPIC30F devices have a Watchdog Timer, which is permanently enabled via the Configuration bits or can be software controlled. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Startup Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable. The other is the Powerup Timer (PWRT), which provides a delay on power-up only, designed to keep the part in Reset while the power supply stabilizes. With these two timers on-chip, most applications need no external Reset circuitry.

Sleep mode is designed to offer a very low current Power-down mode. The user can wake-up from Sleep through external Reset, Watchdog Timer Wake-up or through an interrupt. Several oscillator options are also made available to allow the part to fit a wide variety of applications. In the Idle mode, the clock sources are still active, but the CPU is shut-off. The RC oscillator option saves system cost, while the LP crystal option saves power.

### 21.1 Oscillator System Overview

The dsPIC30F oscillator system has the following modules and features:

- Various external and internal oscillator options as clock sources
- An on-chip PLL to boost internal operating frequency
- A clock switching mechanism between various clock sources
- Programmable clock postscaler for system power savings
- A Fail-Safe Clock Monitor (FSCM) that detects clock failure and takes fail-safe measures
- Clock Control Register OSCCON
- · Configuration bits for main oscillator selection

Table 21-1 provides a summary of the dsPIC30F oscillator operating modes. A simplified diagram of the oscillator system is shown in Figure 21-1.

Configuration bits determine the clock source upon Power-on Reset (POR) and Brown-out Reset (BOR). Thereafter, the clock source can be changed between permissible clock sources. The OSCCON register controls the clock switching and reflects system clock related status bits.

### TABLE 21-1: OSCILLATOR OPERATING MODES

Oscillator Mode	Description				
XTL	200 kHz-4 MHz crystal on OSC1:OSC2.				
ХТ	4 MHz-10 MHz crystal on OSC1:OSC2.				
XT w/ PLL 4x	4 MHz-10 MHz crystal on OSC1:OSC2. 4x PLL enabled.				
XT w/ PLL 8x	4 MHz-10 MHz crystal on OSC1:OSC2. 8x PLL enabled.				
XT w/ PLL 16x	4 MHz-10 MHz crystal on OSC1:OSC2. 16x PLL enabled <sup>(1)</sup> .				
LP	32 kHz crystal on SOSCO:SOSCI <sup>(2)</sup> .				
HS	10 MHz-25 MHz crystal.				
EC	External clock input (0-40 MHz).				
ECIO	External clock input (0-40 MHz). OSC2 pin is I/O.				
EC w/ PLL 4x	External clock input (0-40 MHz). OSC2 pin is I/O. 4x PLL enabled <sup>(1)</sup> .				
EC w/ PLL 8x	External clock input (0-40 MHz). OSC2 pin is I/O. 8x PLL enabled <sup>(1)</sup> .				
EC w/ PLL 16x	External clock input (0-40 MHz). OSC2 pin is I/O. 16x PLL enabled <sup>(1)</sup> .				
ERC	External RC oscillator. OSC2 pin is Fosc/4 output <sup>(3)</sup> .				
ERCIO	External RC oscillator. OSC2 pin is I/O <sup>(3)</sup> .				
FRC	7.37 MHz internal RC Oscillator.				
LPRC	512 kHz internal RC Oscillator.				

**Note 1:** dsPIC30F maximum operating frequency of 120 MHz must be met.

2: LP oscillator can be conveniently shared as system clock, as well as real-time clock for Timer1.

3: Requires external R and C. Frequency operation up to 4 MHz.



## 21.2 Oscillator Configurations

#### 21.2.1 INITIAL CLOCK SOURCE SELECTION

While coming out of Power-on Reset or Brown-out Reset, the device selects its clock source based on:

- a) FOS<1:0> Configuration bits that select one of four oscillator groups.
- b) AND FPR<3:0> Configuration bits that select one of 13 oscillator choices within the primary group.

The selection is as shown in Table 21-2.

Oscillator Mode	Oscillator Source	FOS1	FOS0	FPR3	FPR2	FPR1	FPR0	OSC2 Function
EC	Primary	1	1	1	0	1	1	CLKO
ECIO	Primary	1	1	1	1	0	0	I/O
EC w/ PLL 4x	Primary	1	1	1	1	0	1	I/O
EC w/ PLL 8x	Primary	1	1	1	1	1	0	I/O
EC w/ PLL 16x	Primary	1	1	1	1	1	1	I/O
ERC	Primary	1	1	1	0	0	1	CLKO
ERCIO	Primary	1	1	1	0	0	0	I/O
ХТ	Primary	1	1	0	1	0	0	OSC2
XT w/ PLL 4x	Primary	1	1	0	1	0	1	OSC2
XT w/ PLL 8x	Primary	1	1	0	1	1	0	OSC2
XT w/ PLL 16x	Primary	1	1	0	1	1	1	OSC2
XTL	Primary	1	1	0	0	0	Х	OSC2
HS	Primary	1	1	0	0	1	Х	OSC2
LP	Secondary	0	0		—		—	(Notes 1, 2)
FRC	Internal FRC	0	1	—	—	—	—	(Notes 1, 2)
LPRC	Internal LPRC	1	0	—	—	—	—	(Notes 1, 2)

#### TABLE 21-2: CONFIGURATION BIT VALUES FOR CLOCK SELECTION

**Note** 1: OSC2 pin function is determined by the Primary Oscillator mode selection (FPR<3:0>).

2: Note that OSC1 pin cannot be used as an I/O pin, even if the secondary oscillator or an internal clock source is selected at all times.

#### 21.2.2 OSCILLATOR START-UP TIMER (OST)

In order to ensure that a crystal oscillator (or ceramic resonator) has started and stabilized, an oscillator start-up timer is included. It is a simple 10-bit counter that counts 1024 Tosc cycles before releasing the oscillator clock to the rest of the system. The time-out period is designated as TOST. The TOST time is involved every time the oscillator has to restart (i.e., on POR, BOR and wake-up from Sleep). The oscillator start-up timer is applied to the LP Oscillator, XT, XTL, and HS modes (upon wake-up from Sleep, POR and BOR) for the primary oscillator.

## 21.2.3 LP OSCILLATOR CONTROL

Enabling the LP oscillator is controlled with two elements:

- 1. The current oscillator group bits COSC<1:0>.
- 2. The LPOSCEN bit (OSCON register).

The LP oscillator is ON (even during Sleep mode) if LPOSCEN = 1. The LP oscillator is the device clock if:

- COSC<1:0> = 00 (LP selected as main oscillator) and
- LPOSCEN = 1

Keeping the LP oscillator ON at all times allows for a fast switch to the 32 kHz system clock for lower power operation. Returning to the faster main oscillator will still require a start-up time.
#### 21.2.4 PHASE LOCKED LOOP (PLL)

The PLL multiplies the clock which is generated by the primary oscillator. The PLL is selectable to have either gains of x4, x8, and x16. Input and output frequency ranges are summarized in Table 21-3.

Fin	PLL Multiplier	Fout
4 MHz-10 MHz	x4	16 MHz-40 MHz
4 MHz-10 MHz	x8	32 MHz-80 MHz
4 MHz-7.5 MHz	x16	64 MHz-120 MHz

TABLE 21-3: PLL FREQUENCY RANGE

The PLL features a lock output, which is asserted when the PLL enters a phase locked state. Should the loop fall out of lock (e.g., due to noise), the lock signal will be rescinded. The state of this signal is reflected in the read only LOCK bit in the OSCCON register.

#### 21.2.5 FAST RC OSCILLATOR (FRC)

The FRC oscillator is a fast (7.37 MHz +/- 2% nominal) internal RC oscillator. This oscillator is intended to provide reasonable device operating speeds without the use of an external crystal, ceramic resonator, or RC network.

The dsPIC30F operates from the FRC oscillator whenever the Current Oscillator Selection control bits in the OSCCON register (OSCCON<13:12>) are set to '01'.

There are four tuning bits (TUN<3:0>) for the FRC oscillator in the OSCCON register. These tuning bits allow the FRC oscillator frequency to be adjusted as close to 7.37 MHz as possible, depending on the device operating conditions. The FRC oscillator frequency has been calibrated during factory testing. Table 21-4 describes the adjustment range of the TUN<3:0> bits.

TUN<3:0> Bits	FRC Frequency
0111	+ 10.5%
0110	+ 9.0%
0101	+ 7.5%
0100	+ 6.0%
0011	+ 4.5%
0010	+ 3.0%
0001	+ 1.5%
0000	Center Frequency (oscillator is running at calibrated frequency)
1111	- 1.5%
1110	- 3.0%
1101	- 4.5%
1100	- 6.0%
1011	- 7.5%
1010	- 9.0%

TABI F	21-4.	FRC	TUNING
IADLL	<u> </u>	1110	

TUN<3:0> Bits	FRC Frequency
1001	- 10.5%
1000	- 12.0%

#### 21.2.6 LOW POWER RC OSCILLATOR (LPRC)

The LPRC oscillator is a component of the Watchdog Timer (WDT) and oscillates at a nominal frequency of 512 kHz. The LPRC oscillator is the clock source for the Power-up Timer (PWRT) circuit, WDT and clock monitor circuits. It may also be used to provide a lowfrequency clock source option for applications where power consumption is critical, and timing accuracy is not required.

The LPRC oscillator is always enabled at a Power-on Reset, because it is the clock source for the PWRT. After the PWRT expires, the LPRC oscillator will remain ON if one of the following is TRUE:

- The Fail-Safe Clock Monitor is enabled
- The WDT is enabled
- The LPRC oscillator is selected as the system clock via the COSC<1:0> control bits in the OSCCON register

If one of the above conditions is not true, the LPRC will shut-off after the PWRT expires.

Note 1:	OSC2 pin function is determined by the Primary Oscillator mode selection (FPR<3:0>).
2:	Note that OSC1 pin cannot be used as an I/O pin, even if the secondary oscillator or an internal clock source is selected at all
	times.

#### 21.2.7 FAIL-SAFE CLOCK MONITOR

The Fail-Safe Clock Monitor (FSCM) allows the device to continue to operate even in the event of an oscillator failure. The FSCM function is enabled by appropriately programming the FCKSM Configuration bits (Clock Switch and Monitor Selection bits) in the Fosc device configuration register. If the FSCM function is enabled, the LPRC Internal oscillator will run at all times (except during Sleep mode) and will not be subject to control by the SWDTEN bit.

In the event of an oscillator failure, the FSCM will generate a clock failure trap event and will switch the system clock over to the FRC oscillator. The user will then have the option to either attempt to restart the oscillator or execute a controlled shutdown. The user may decide to treat the trap as a warm Reset by simply loading the Reset address into the oscillator fail trap vector. In this event, the CF (Clock Fail) status bit (OSCCON<3>) is also set whenever a clock failure is recognized.

In the event of a clock failure, the WDT is unaffected and continues to run on the LPRC clock.

# dsPIC30F6010

If the oscillator has a very slow start-up time coming out of POR, BOR or Sleep, it is possible that the PWRT timer will expire before the oscillator has started. In such cases, the FSCM will be activated and the FSCM will initiate a clock failure trap, and the COSC<1:0> bits are loaded with FRC oscillator selection. This will effectively shut-off the original oscillator that was trying to start.

The user may detect this situation and restart the oscillator in the clock fail trap ISR.

Upon a clock failure detection, the FSCM module will initiate a clock switch to the FRC Oscillator as follows:

- 1. The COSC bits (OSCCON<13:12>) are loaded with the FRC Oscillator selection value.
- 2. CF bit is set (OSCCON<3>).
- 3. OSWEN control bit (OSCCON<0>) is cleared.

For the purpose of clock switching, the clock sources are sectioned into four groups:

- 1. Primary
- 2. Secondary
- 3. Internal FRC
- 4. Internal LPRC

The user can switch between these functional groups, but cannot switch between options within a group. If the primary group is selected, then the choice within the group is always determined by the FPR<3:0> Configuration bits.

The OSCCON register holds the Control and Status bits related to clock switching.

- COSC<1:0>: Read only status bits always reflect the current oscillator group in effect.
- NOSC<1:0>: Control bits which are written to indicate the new oscillator group of choice.
  - On POR and BOR, COSC<1:0> and NOSC<1:0> are both loaded with the Configuration bit values FOS<1:0>.
- LOCK: The LOCK status bit indicates a PLL lock.
- CF: Read only status bit indicating if a clock fail detect has occurred.
- OSWEN: Control bit changes from a '0' to a '1' when a clock transition sequence is initiated. Clearing the OSWEN control bit will abort a clock transition in progress (used for hang-up situations).

If Configuration bits FCKSM<1:0> = 1x, then the clock switching and fail-safe clock monitor functions are disabled. This is the default Configuration bit setting.

If clock switching is disabled, then the FOS<1:0> and FPR<3:0> bits directly control the oscillator selection and the COSC<1:0> bits do not control the clock selection. However, these bits will reflect the clock source selection.

Note: The application should not attempt to switch to a clock of frequency lower than 100 KHz when the fail-safe clock monitor is enabled. If such clock switching is performed, the device may generate an oscillator fail trap and switch to the Fast RC oscillator.

#### 21.2.8 PROTECTION AGAINST ACCIDENTAL WRITES TO OSCCON

A write to the OSCCON register is intentionally made difficult because it controls clock switching and clock scaling.

To write to the OSCCON low byte, the following code sequence must be executed without any other instructions in between:

- Byte Write "0x46" to OSCCON low
- Byte Write "0x57" to OSCCON low

*Byte Write is allowed for one instruction cycle.* Write the desired value or use bit manipulation instruction.

To write to the OSCCON high byte, the following instructions must be executed without any other instructions in between:

- Byte Write "0x78" to OSCCON high
- Byte Write "0x9A" to OSCCON high

Byte Write is allowed for one instruction cycle. Write the desired value or use bit manipulation instruction.

#### 21.3 Reset

The PIC18F1220/1320 differentiates between various kinds of Reset:

- a) Power-on Reset (POR)
- b) MCLR Reset during normal operation
- c) MCLR Reset during Sleep
- d) Watchdog Timer (WDT) Reset (during normal operation)
- e) Programmable Brown-out Reset (BOR)
- f) RESET Instruction
- g) Reset cause by trap lockup (TRAPR)
- Reset caused by illegal opcode, or by using an uninitialized W register as an address pointer (IOPUWR)

Different registers are affected in different ways by various Reset conditions. Most registers are not affected by a WDT wake-up, since this is viewed as the resumption of normal operation. Status bits from the RCON register are set or cleared differently in different Reset situations, as indicated in Table 21-5. These bits are used in software to determine the nature of the Reset.

A block diagram of the on-chip Reset circuit is shown in Figure 21-2.

A MCLR noise filter is provided in the MCLR Reset path. The filter detects and ignores small pulses.

Internally generated Resets do not drive MCLR pin low.



#### 21.3.1 POR: POWER-ON RESET

A power-on event will generate an internal POR pulse when a VDD rise is detected. The Reset pulse will occur at the POR circuit threshold voltage (VPOR), which is nominally 1.85V. The device supply voltage characteristics must meet specified starting voltage and rise rate requirements. The POR pulse will reset a POR timer and place the device in the Reset state. The POR also selects the device clock source identified by the oscillator configuration fuses. The POR circuit inserts a small delay, TPOR, which is nominally 10  $\mu$ s and ensures that the device bias circuits are stable. Furthermore, a user selected power-up time-out (TPWRT) is applied. The TPWRT parameter is based on device Configuration bits and can be 0 ms (no delay), 4 ms, 16 ms or 64 ms. The total delay is at device power-up TPOR + TPWRT. When these delays have expired, SYSRST will be negated on the next leading edge of the Q1 clock, and the PC will jump to the Reset vector.

The timing for the SYSRST signal is shown in Figure 21-3 through Figure 21-5.

# dsPIC30F6010



## FIGURE 21-4: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 1



#### FIGURE 21-5: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 2



# 21.3.1.1 POR with Long Crystal Start-up Time (with FSCM Enabled)

The oscillator start-up circuitry is not linked to the POR circuitry. Some crystal circuits (especially low-frequency crystals) will have a relatively long start-up time. Therefore, one or more of the following conditions is possible after the POR timer and the PWRT have expired:

- The oscillator circuit has not begun to oscillate.
- The oscillator start-up timer has NOT expired (if a crystal oscillator is used).
- The PLL has not achieved a LOCK (if PLL is used).

If the FSCM is enabled and one of the above conditions is true, then a clock failure trap will occur. The device will automatically switch to the FRC oscillator and the user can switch to the desired crystal oscillator in the trap ISR.

#### 21.3.1.2 Operating without FSCM and PWRT

If the FSCM is disabled and the Power-up Timer (PWRT) is also disabled, then the device will exit rapidly from Reset on power-up. If the clock source is FRC, LPRC, EXTRC or EC, it will be active immediately.

If the FSCM is disabled and the system clock has not started, the device will be in a frozen state at the Reset vector until the system clock starts. From the user's perspective, the device will appear to be in Reset until a system clock is available.

#### 21.3.2 BOR: PROGRAMMABLE BROWN-OUT RESET

The BOR (Brown-out Reset) module is based on an internal voltage reference circuit. The main purpose of the BOR module is to generate a device Reset when a brown-out condition occurs. Brown-out conditions are generally caused by glitches on the AC mains (i.e., missing portions of the AC cycle waveform due to bad power transmission lines or voltage sags due to excessive current draw when a large inductive load is turned on).

The BOR module allows selection of one of the following voltage trip points:

- 2.6V-2.71V
- 4.1V-4.4V
- 4.58V-4.73V

**Note:** The BOR voltage trip points indicated here are nominal values provided for design guidance only.

A BOR will generate a Reset pulse which will reset the device. The BOR will select the clock source, based on the device Configuration bit values (FOS<1:0> and FPR<3:0>). Furthermore, if an Oscillator mode is selected, the BOR will activate the Oscillator Start-up Timer (OST). The system clock is held until OST expires. If the PLL is used, then the clock will be held until the LOCK bit (OSCCON<5>) is '1'.

Concurrently, the POR time-out (TPOR) and the PWRT time-out (TPWRT) will be applied before the internal Reset is released. If TPWRT = 0 and a crystal oscillator is being used, then a nominal delay of TFSCM = 100  $\mu$ s is applied. The total delay in this case is (TPOR + TFSCM).

The BOR status bit (RCON<1>) will be set to indicate that a BOR has occurred. The BOR circuit, if enabled, will continue to operate while in Sleep or Idle modes and will reset the device should VDD fall below the BOR threshold voltage.

FIGURE 21-6:

#### EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



- Note 1: External Power-on Reset circuit is required only if the VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.
  - 2: R should be suitably chosen so as to make sure that the voltage drop across R does not violate the device's electrical specification.
  - R1 should be suitably chosen so as to limit any current flowing into MCLR from external capacitor C, in the event of MCLR/VPP pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

Note: Dedicated supervisory devices, such as the MCP1XX and MCP8XX, may also be used as an external Power-on Reset circuit. Table 21-5 shows the Reset conditions for the RCON Register. Since the control bits within the RCON register are R/W, the information in the table implies that all the bits are negated prior to the action specified in the condition column.

Condition	Program Counter	TRAPR	IOPUWR	EXTR	SWR	WDTO	IDLE	SLEEP	POR	BOR
Power-on Reset	0x000000	0	0	0	0	0	0	0	1	1
Brown-out Reset	0x000000	0	0	0	0	0	0	0	0	1
MCLR Reset during normal operation	0x000000	0	0	1	0	0	0	0	0	0
Software Reset during normal operation	0x000000	0	0	0	1	0	0	0	0	0
MCLR Reset during Sleep	0x000000	0	0	1	0	0	0	1	0	0
MCLR Reset during Idle	0x000000	0	0	1	0	0	1	0	0	0
WDT Time-out Reset	0x000000	0	0	0	0	1	0	0	0	0
WDT Wake-up	PC + 2	0	0	0	0	1	0	1	0	0
Interrupt Wake-up from Sleep	PC + 2 <sup>(1)</sup>	0	0	0	0	0	0	1	0	0
Clock Failure Trap	0x000004	0	0	0	0	0	0	0	0	0
Trap Reset	0x000000	1	0	0	0	0	0	0	0	0
Illegal Operation Trap	0x000000	0	1	0	0	0	0	0	0	0

## TABLE 21-5: INITIALIZATION CONDITION FOR RCON REGISTER CASE 1

**Note 1:** When the wake-up is due to an enabled interrupt, the PC is loaded with the corresponding interrupt vector.

Table 21-6 shows a second example of the bit conditions for the RCON Register. In this case, it is not assumed the user has set/cleared specific bits prior to action specified in the condition column.

#### TABLE 21-6: INITIALIZATION CONDITION FOR RCON REGISTER CASE 2

Condition	Program Counter	TRAPR	IOPUWR	EXTR	SWR	WDTO	IDLE	SLEEP	POR	BOR
Power-on Reset	0x000000	0	0	0	0	0	0	0	1	1
Brown-out Reset	0x000000	u	u	u	u	u	u	u	0	1
MCLR Reset during normal operation	0x000000	u	u	1	0	0	0	0	u	u
Software Reset during normal operation	0x000000	u	u	0	1	0	0	0	u	u
MCLR Reset during Sleep	0x000000	u	u	1	u	0	0	1	u	u
MCLR Reset during Idle	0x000000	u	u	1	u	0	1	0	u	u
WDT Time-out Reset	0x000000	u	u	0	0	1	0	0	u	u
WDT Wake-up	PC + 2	u	u	u	u	1	u	1	u	u
Interrupt Wake-up from Sleep	PC + 2 <sup>(1)</sup>	u	u	u	u	u	u	1	u	u
Clock Failure Trap	0x000004	u	u	u	u	u	u	u	u	u
Trap Reset	0x000000	1	u	u	u	u	u	u	u	u
Illegal Operation Reset	0x000000	u	1	u	u	u	u	u	u	u

Legend: u = unchanged

Note 1: When the wake-up is due to an enabled interrupt, the PC is loaded with the corresponding interrupt vector.

#### 21.4 Watchdog Timer (WDT)

#### 21.4.1 WATCHDOG TIMER OPERATION

The primary function of the Watchdog Timer (WDT) is to reset the processor in the event of a software malfunction. The WDT is a free running timer, which runs off an on-chip RC oscillator, requiring no external component. Therefore, the WDT timer will continue to operate even if the main processor clock (e.g., the crystal oscillator) fails.

# 21.4.2 ENABLING AND DISABLING THE WDT

The Watchdog Timer can be "enabled" or "disabled" only through a Configuration bit (FWDTEN) in the configuration register FWDT.

Setting FWDTEN = 1 enables the Watchdog Timer. The enabling is done when programming the device. By default, after chip-erase, FWDTEN bit = 1. Any device programmer capable of programming dsPIC30F devices allows programming of this and other Configuration bits.

If enabled, the WDT will increment until it overflows or "times out". A WDT time-out will force a device Reset (except during Sleep). To prevent a WDT time-out, the user must clear the Watchdog Timer using a CLRWDT instruction.

If a WDT times out during Sleep, the device will wakeup. The WDTO bit in the RCON register will be cleared to indicate a wake-up resulting from a WDT time-out.

Setting FWDTEN = 0 allows user software to enable/ disable the Watchdog Timer via the SWDTEN (RCON<5>) control bit.

#### 21.5 Low-Voltage Detect

The Low-Voltage Detect (LVD) module is used to detect when the VDD of the device drops below a threshold value VLVD, which is determined by the LVDL<3:0> bits (RCON<11:8>) and is thus user-programmable. The internal voltage reference circuitry requires a nominal amount of time to stabilize, and the BGST bit (RCON<13>) indicates when the voltage reference has stabilized.

In some devices, the LVD threshold voltage may be applied externally on the LVDIN pin.

The LVD module is enabled by setting the LVDEN bit (RCON<12>).

#### 21.6 Power Saving Modes

There are two power saving states that can be entered through the execution of a special instruction, PWRSAV.

These are: Sleep and Idle.

The format of the PWRSAV instruction is as follows:

PWRSAV <parameter>, where 'parameter' defines Idle or Sleep mode.

#### 21.6.1 SLEEP MODE

In Sleep mode, the clock to the CPU and peripherals is shutdown. If an on-chip oscillator is being used, it is shutdown.

The fail-safe clock monitor is not functional during Sleep, since there is no clock to monitor. However, LPRC clock remains active if WDT is operational during Sleep.

The Brown-out protection circuit and the Low-Voltage Detect circuit, if enabled, will remain functional during Sleep.

The processor wakes up from Sleep if at least one of the following conditions has occurred:

- any interrupt that is individually enabled and meets the required priority level
- any Reset (POR, BOR and MCLR)
- WDT time-out

On waking up from Sleep mode, the processor will restart the same clock that was active prior to entry into Sleep mode. When clock switching is enabled, bits COSC<1:0> will determine the oscillator source that will be used on wake-up. If clock switch is disabled, then there is only one system clock.

**Note:** If a POR or BOR occurred, the selection of the oscillator is based on the FOS<1:0> and FPR<3:0> Configuration bits.

If the clock source is an oscillator, the clock to the device will be held off until OST times out (indicating a stable oscillator). If PLL is used, the system clock is held off until LOCK = 1 (indicating that the PLL is stable). In either case, TPOR, TLOCK and TPWRT delays are applied.

If EC, FRC, LPRC or EXTRC oscillators are used, then a delay of TPOR (~10  $\mu$ s) is applied. This is the smallest delay possible on wake-up from Sleep.

Moreover, if LP oscillator was active during Sleep, and LP is the oscillator used on wake-up, then the start-up delay will be equal to TPOR. PWRT delay and OST timer delay are not applied. In order to have the smallest possible start-up delay when waking up from Sleep, one of these faster wake-up options should be selected before entering Sleep. Any interrupt that is individually enabled (using the corresponding IE bit) and meets the prevailing priority level will be able to wake-up the processor. The processor will process the interrupt and branch to the ISR. The Sleep status bit in RCON register is set upon wake-up.

Note: In spite of various delays applied (TPOR, TLOCK and TPWRT), the crystal oscillator (and PLL) may not be active at the end of the time-out (e.g., for low-frequency crystals. In such cases), if FSCM is enabled, then the device will detect this as a clock failure and process the clock failure trap, the FRC oscillator will be enabled, and the user will have to re-enable the crystal oscillator. If FSCM is not enabled, then the device will simply suspend execution of code until the clock is stable, and will remain in Sleep until the oscillator clock has started.

All Resets will wake-up the processor from Sleep mode. Any Reset, other than POR, will set the Sleep status bit. In a POR, the Sleep bit is cleared.

If Watchdog Timer is enabled, then the processor will wake-up from Sleep mode upon WDT time-out. The Sleep and WDTO status bits are both set.

#### 21.6.2 IDLE MODE

In Idle mode, the clock to the CPU is shutdown while peripherals keep running. Unlike Sleep mode, the clock source remains active.

Several peripherals have a control bit in each module, that allows them to operate during Idle.

LPRC fail-safe clock remains active if clock failure detect is enabled.

The processor wakes up from Idle if at least one of the following conditions is true:

- on any interrupt that is individually enabled (IE bit is '1') and meets the required priority level
- on any Reset (POR, BOR, MCLR)
- on WDT time-out

Upon wake-up from Idle mode, the clock is re-applied to the CPU and instruction execution begins immediately, starting with the instruction following the PWRSAV instruction.

Any interrupt that is individually enabled (using IE bit) and meets the prevailing priority level will be able to wake-up the processor. The processor will process the interrupt and branch to the ISR. The Idle status bit in RCON register is set upon wake-up.

Any Reset, other than POR, will set the Idle status bit. On a POR, the Idle bit is cleared.

If Watchdog Timer is enabled, then the processor will wake-up from Idle mode upon WDT time-out. The Idle and WDTO status bits are both set.

Unlike wake-up from Sleep, there are no time delays involved in wake-up from Idle.

## 21.7 Device Configuration Registers

The Configuration bits in each device configuration register specify some of the device modes and are programmed by a device programmer, or by using the In-Circuit Serial Programming (ICSP) feature of the device. Each device configuration register is a 24-bit register, but only the lower 16 bits of each register are used to hold configuration data. There are four device configuration registers available to the user:

- 1. Fosc (0xF80000): Oscillator Configuration Register
- 2. FWDT (0xF80002): Watchdog Timer Configuration Register
- 3. FBORPOR (0xF80004): BOR and POR Configuration Register
- 4. FGS (0xF8000A): General Code Segment Configuration Register

The placement of the Configuration bits is automatically handled when you select the device in your device programmer. The desired state of the Configuration bits may be specified in the source code (dependent on the language tool used), or through the programming interface. After the device has been programmed, the application software may read the Configuration bit values through the table read instructions. For additional information, please refer to the programming specifications of the device.

**Note:** If the code protection configuration fuse bits (FGS<GCP> and FGS<GWRP>) have been programmed, an erase of the entire code-protected device is only possible at voltages  $VDD \ge 4.5V$ .

## 21.8 In-Circuit Debugger

When MPLAB ICD2 is selected as a Debugger, the In-Circuit Debugging functionality is enabled. This function allows simple debugging functions when used with MPLAB IDE. When the device has this feature enabled, some of the resources are not available for general use. These resources include the first 80 bytes of Data RAM and two I/O pins.

One of four pairs of Debug I/O pins may be selected by the user using configuration options in MPLAB IDE. These pin pairs are named EMUD/EMUC, EMUD1/ EMUC1, EMUD2/EMUC2 and EMUD3/EMUC3. In each case, the selected EMUD pin is the Emulation/ Debug Data line, and the EMUC pin is the Emulation/ Debug Clock line. These pins will interface to the MPLAB ICD 2 module available from Microchip. The selected pair of Debug I/O pins is used by MPLAB ICD 2 to send commands and receive responses, as well as to send and receive data. To use the In-Circuit Debugger function of the device, the design must implement ICSP connections to MCLR, VDD, VSS, PGC, PGD and the selected EMUDx/EMUCx pin pair.

This gives rise to two possibilities:

- If EMUD/EMUC is selected as the Debug I/O pin pair, then only a 5-pin interface is required, as the EMUD and EMUC pin functions are multiplexed with the PGD and PGC pin functions in all dsPIC30F devices.
- If EMUD1/EMUC1, EMUD2/EMUC2 or EMUD3/ EMUC3 is selected as the Debug I/O pin pair, then a 7-pin interface is required, as the EMUDx/EMUCx pin functions (x = 1, 2 or 3) are not multiplexed with the PGD and PGC pin functions.

#### TABLE 21-7: SYSTEM INTEGRATION REGISTER MAP

SFR Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RCON	0740	TRAPR	IOPUWR	BGST	LVDEN		LVDI	_<3:0>		EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR
OSCCON	0742	_		COSC	C<1:0>	_		NOSC	<1:0>	POST	<1:0>	LOCK		CF		LPOSCEN	OSWEI
PMD1	0770	T5MD	T4MD	T3MD	T2MD	T1MD	QEIMD	PWMMD	_	I2CMD	U2MD	U1MD	SPI2MD	SPI1MD	C2MD	C1MD	ADCM
PMD2	0772	IC8MD	IC7MD	IC6MD	IC5MD	IC4MD	IC3MD	IC2MD	IC1MD	OC8MD	OC7MD	OC6MD	OC5MD	OC4MD	OC3MD	OC2MD	OC1M
l egend.	11 - 11	initialized	l hit														

uninitialized bit end:

# TABLE 21-8: DEVICE CONFIGURATION REGISTER MAP

File Name	Addr.	Bits 23-16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bi
FOSC	F80000	_	FCKSM<	<1:0>	_	_	_	_	FOS<	:1:0>	_	-	_	_	
FWDT	F80002	_	FWDTEN	_	_	_	—	_	—	_	_	-	FWPS	A<1:0>	
FBORPOR	F80004	_	MCLREN	_	_	_	_	PWMPIN	HPOL	LPOL	BOREN	_	BORV	/<1:0>	-
FGS	F8000A	-		_		_	_		_	_		_	—	_	-

Note: Refer to dsPIC30F Family Reference Manual (DS70046) for descriptions of register bit fields.

# 22.0 DEVELOPMENT SUPPORT

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

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

### 22.1 MPLAB Integrated Development Environment Software

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

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

The MPLAB IDE allows you to:

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

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

## 22.2 MPASM Assembler

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

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

The MPASM Assembler features include:

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

#### 22.3 MPLAB C18 and MPLAB C30 C Compilers

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

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

#### 22.4 MPLINK Object Linker/ MPLIB Object Librarian

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

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

The object linker/library features include:

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

# 22.5 MPLAB ASM30 Assembler, Linker and Librarian

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

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

## 22.6 MPLAB SIM Software Simulator

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

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

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

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

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

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

## 22.8 MPLAB REAL ICE In-Circuit Emulator System

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

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

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

# 22.9 MPLAB ICD 2 In-Circuit Debugger

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

## 22.10 MPLAB PM3 Device Programmer

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

#### 22.11 PICSTART Plus Development Programmer

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

## 22.12 PICkit 2 Development Programmer

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

## 22.13 Demonstration, Development and Evaluation Boards

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

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

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

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

Check the Microchip web page (www.microchip.com) and the latest *"Product Selector Guide"* (DS00148) for the complete list of demonstration, development and evaluation kits.

# 23.0 INSTRUCTION SET SUMMARY

**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 "*dsPIC30F/ 33F Programmer's Reference Manual*" (DS70030).

The dsPIC30F instruction set adds many enhancements to the previous  $PIC^{(R)}$  MCU instruction sets, while maintaining an easy migration from PIC MCU instruction sets.

Most instructions are a single program memory word (24 bits). Only three instructions require two program memory locations.

Each single-word instruction is a 24-bit word divided into an 8-bit opcode which specifies the instruction type, and one or more operands which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into five basic categories:

- Word or byte-oriented operations
- Bit-oriented operations
- · Literal operations
- DSP operations
- · Control operations

Table 23-1 shows the general symbols used in describing the instructions.

The dsPIC30F instruction set summary in Table 23-2 lists all the instructions along with the status flags affected by each instruction.

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

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

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

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

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

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

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

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

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

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

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

- The accumulator (A or B) to be used (required operand)
- The W registers to be used as the two operands
- The X and Y address space prefetch operations
- The X and Y address space prefetch destinations
- · The accumulator write-back destination

The other DSP instructions do not involve any multiplication, and may include:

- The accumulator to be used (required)
- The source or destination operand (designated as Wso or Wdo, respectively) with or without an address modifier
- The amount of shift, specified by a W register 'Wn' or a literal value

The control instructions may use some of the following operands:

- A program memory address
- The mode of the Table Read and Table Write instructions

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

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

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

Field	Description
#text	Means literal defined by "text"
(text)	Means "content of text"
[text]	Means "the location addressed by text"
{ }	Optional field or operation
<n:m></n:m>	Register bit field
.b	Byte mode selection
.d	Double-word mode selection
.S	Shadow register select
. W	Word mode selection (default)
Acc	One of two accumulators {A, B}
AWB	Accumulator write-back destination address register ∈ {W13, [W13]+=2}
bit4	4-bit bit selection field (used in word addressed instructions) $\in \{015\}$
C, DC, N, OV, Z	MCU status bits: Carry, Digit Carry, Negative, Overflow, Zero
Expr	Absolute address, label or expression (resolved by the linker)
f	File register address ∈ {0x00000x1FFF}
lit1	1-bit unsigned literal $\in \{0,1\}$
lit4	4-bit unsigned literal $\in \{015\}$
lit5	5-bit unsigned literal ∈ {031}
lit8	8-bit unsigned literal ∈ {0255}
lit10	10-bit unsigned literal $\in$ {0255} for Byte mode, {0:1023} for Word mode
lit14	14-bit unsigned literal $\in \{016384\}$
lit16	16-bit unsigned literal $\in \{065535\}$
lit23	23-bit unsigned literal $\in$ {08388608}; LSB must be 0
None	Field does not require an entry, may be blank
OA, OB, SA, SB	DSP status bits: ACCA Overflow, ACCB Overflow, ACCA Saturate, ACCB Saturate
PC	Program Counter
Slit10	10-bit signed literal ∈ {-512511}
Slit16	16-bit signed literal ∈ {-3276832767}
Slit6	6-bit signed literal ∈ {-1616}

TABLE 23-1:SYMBOLS USED IN OPCODE DESCRIPTIONS

Field	Description
Wb	Base W register ∈ {W0W15}
Wd	Destination W register ∈ { Wd, [Wd], [Wd++], [Wd], [++Wd], [Wd] }
Wdo	Destination W register ∈ { Wnd, [Wnd], [Wnd++], [Wnd], [++Wnd], [Wnd], [Wnd+Wb] }
Wm,Wn	Dividend, Divisor working register pair (direct addressing)
Wm*Wm	Multiplicand and Multiplier working register pair for Square instructions $\in$ {W4*W4,W5*W5,W6*W6,W7*W7}
Wm*Wn	Multiplicand and Multiplier working register pair for DSP instructions ∈ {W4*W5,W4*W6,W4*W7,W5*W6,W5*W7,W6*W7}
Wn	One of 16 working registers ∈ {W0W15}
Wnd	One of 16 destination working registers $\in$ {W0W15}
Wns	One of 16 source working registers ∈ {W0W15}
WREG	W0 (working register used in file register instructions)
Ws	Source W register ∈ { Ws, [Ws], [Ws++], [Ws], [++Ws], [Ws] }
Wso	Source W register ∈ { Wns, [Wns], [Wns++], [Wns], [++Wns], [Wns], [Wns+Wb] }
Wx	X data space prefetch address register for DSP instructions ∈ {[W8]+=6, [W8]+=4, [W8]+=2, [W8], [W8]-=6, [W8]-=4, [W8]-=2, [W9]+=6, [W9]+=4, [W9]+=2, [W9], [W9]-=6, [W9]-=4, [W9]-=2, [W9+W12],none}
Wxd	X data space prefetch destination register for DSP instructions $\in$ {W4W7}
Wy	Y data space prefetch address register for DSP instructions ∈ {[W10]+=6, [W10]+=4, [W10]+=2, [W10], [W10]-=6, [W10]-=4, [W10]-=2, [W11]+=6, [W11]+=4, [W11]+=2, [W11], [W11]-=6, [W11]-=4, [W11]-=2, [W11+W12], none}
Wyd	Y data space prefetch destination register for DSP instructions $\in$ {W4W7}

# TABLE 23-1: SYMBOLS USED IN OPCODE DESCRIPTIONS (CONTINUED)

## TABLE 23-2: INSTRUCTION SET OVERVIEW

Base Instr #	Assembly Mnemonic		Assembly Syntax	Description	# of words	# of cycle s	Status Flags Affected
1	ADD	ADD	Acc	Add Accumulators	1	1	OA,OB,SA,SB
		ADD	f	f = f + WREG	1	1	C,DC,N,OV,Z
		ADD	f,WREG	WREG = f + WREG	1	1	C,DC,N,OV,Z
		ADD	#lit10,Wn	Wd = lit10 + Wd	1	1	C,DC,N,OV,Z
		ADD	Wb,Ws,Wd	Wd = Wb + Ws	1	1	C,DC,N,OV,Z
		ADD	Wb,#lit5,Wd	Wd = Wb + lit5	1	1	C,DC,N,OV,Z
		ADD	Wso,#Slit4,Acc	16-bit Signed Add to Accumulator	1	1	OA,OB,SA,SB
2	ADDC	ADDC	f	f = f + WREG + (C)	1	1	C,DC,N,OV,Z
		ADDC	f,WREG	WREG = f + WREG + (C)	1	1	C,DC,N,OV,Z
		ADDC	#lit10,Wn	Wd = lit10 + Wd + (C)	1	1	C,DC,N,OV,Z
		ADDC	Wb,Ws,Wd	Wd = Wb + Ws + (C)	1	1	C,DC,N,OV,Z
		ADDC	Wb,#lit5,Wd	Wd = Wb + lit5 + (C)	1	1	C,DC,N,OV,Z
3	AND	AND	f	f = f .AND. WREG	1	1	N,Z
		AND	f,WREG	WREG = f .AND. WREG	1	1	N,Z
		AND	#lit10.Wn	Wd = lit10 .AND. Wd	1	1	N,Z
		AND	Wb.Ws.Wd	Wd = Wb .AND. Ws	1	1	N,Z
		AND	Wb.#lit5.Wd	Wd = Wb .AND. lit5	1	1	N,Z
4	ASR	ASR	f	f = Arithmetic Right Shift f	1	1	C.N.OV.Z
		ASR	f WREG	WREG = Arithmetic Right Shift f	1	1	C.N.OV.Z
		ASR	We Wd	Wd = Arithmetic Right Shift Ws	1	1	C.N.OV.Z
		ASR	Wh Whe Whd	Wnd = Arithmetic Right Shift Wb by Wns	1	1	N.Z
		ASR	Wb #litE Wod	Wnd = Arithmetic Right Shift Wb by lit5	1	1	N Z
5	BCLR	BCLR	f #bit4	Bit Clear f	1	1	None
Ũ	DODIC	BCLR	I, HDICH	Bit Clear Ws	1	1	None
6	BRA	BRA	C Evpr	Branch if Carry	1	1 (2)	None
		BRA	CF Evor	Branch if greater than or equal	1	1 (2)	None
		BRA	CELL EVDY	Branch if unsigned greater than or equal	1	1 (2)	None
		BRA	CT Ever	Branch if greater than	1	1 (2)	None
		BRA	GI, EXPI	Branch if unsigned greater than	1	1 (2)	None
		BRA	GIO, EXPI	Branch if less than or equal	1	1 (2)	None
		BRA		Branch if unsigned less than or equal	1	1 (2)	None
		BRA		Branch if less than	1	1 (2)	None
		BRA		Branch if unsigned less than	1	1 (2)	None
		BPA	LIU, Expr	Branch if Negative	1	1 (2)	None
		DDA	N, Expr	Branch if Not Carry	1	1 (2)	None
		BPA	NC, Expr	Branch if Not Negative	1	1 (2)	None
		DDA	NN, Expr	Branch if Not Overflow	1	1 (2)	None
		DDA	NOV, Expr	Branch if Not Zoro	1	1 (2)	None
		DRA	NZ, Expr	Branch if accumulator A overflow	1	1 (2)	None
		BRA	OA, Expr	Propoh if accumulator P overflow	1	1 (2)	None
		BRA	OB,Expr	Dranch if Quarflaw	1	1 (2)	None
		BRA	OV,Expr	Branch II Overliow	1	1 (2)	None
		BRA	SA,Expr	Branch if accumulator A saturated	1	1 (2)	None
		BRA	SB,Expr	Branch if accumulator B saturated	1	1 (2)	None
		BRA	Expr		1	2	None
		BRA	Z,Expr	Branch if Zero	1	1 (2)	None
L		BRA	Wn	Computed Branch	1	2	None
7	BSET	BSET	f,#bit4	Bit Set f	1	1	None
L		BSET	Ws,#bit4	Bit Set Ws	1	1	None
8	BSW	BSW.C	Ws,Wb	Write C bit to Ws <wb></wb>	1	1	None
		BSW.Z	Ws,Wb	Write Z bit to Ws <wb></wb>	1	1	None

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of words	# of cycle s	Status Flags Affected
9	BTG	BTG	f,#bit4	Bit Toggle f	1	1	None
		BTG	Ws,#bit4	Bit Toggle Ws	1	1	None
10	BTSC	BTSC	f,#bit4	Bit Test f, Skip if Clear	1	1 (2 or 3)	None
		BTSC	Ws,#bit4	Bit Test Ws, Skip if Clear	1	1 (2 or 3)	None
11	BTSS	BTSS	f,#bit4	Bit Test f, Skip if Set	1	1 (2 or 3)	None
		BTSS	Ws,#bit4	Bit Test Ws, Skip if Set	1	1 (2 or 3)	None
12	BTST	BTST	f,#bit4	Bit Test f	1	1	Z
		BTST.C	Ws,#bit4	Bit Test Ws to C	1	1	С
		BTST.Z	Ws,#bit4	Bit Test Ws to Z	1	1	Z
		BTST.C	Ws,Wb	Bit Test Ws <wb> to C</wb>	1	1	С
		BTST.Z	Ws,Wb	Bit Test Ws <wb> to Z</wb>	1	1	Z
13	BTSTS	BTSTS	f,#bit4	Bit Test then Set f	1	1	Z
		BTSTS.C	Ws,#bit4	Bit Test Ws to C, then Set	1	1	С
		BTSTS.Z	Ws,#bit4	Bit Test Ws to Z, then Set	1	1	Z
14	CALL	CALL	lit23	Call subroutine	2	2	None
		CALL	Wn	Call indirect subroutine	1	2	None
15	CLR	CLR	f	f = 0x0000	1	1	None
		CLR	WREG	WREG = 0x0000	1	1	None
		CLR	Ws	Ws = 0x0000	1	1	None
		CLR	Acc,Wx,Wxd,Wy,Wyd,AWB	Clear Accumulator	1	1	OA,OB,SA,SB
16	CLRWDT	CLRWDT		Clear Watchdog Timer	1	1	WDTO,Sleep
17	COM	COM	f	f = f	1	1	N,Z
		COM	f,WREG	WREG = f	1	1	N,Z
		COM	Ws,Wd	$Wd = \overline{Ws}$	1	1	N,Z
18	CP	CP	f	Compare f with WREG	1	1	C,DC,N,OV,Z
		CP	Wb,#lit5	Compare Wb with lit5	1	1	C,DC,N,OV,Z
		CP	Wb,Ws	Compare Wb with Ws (Wb - Ws)	1	1	C,DC,N,OV,Z
19	CP0	CP0	f	Compare f with 0x0000	1	1	C,DC,N,OV,Z
		CP0	Ws	Compare Ws with 0x0000	1	1	C,DC,N,OV,Z
20	CPB	CPB	f	Compare f with WREG, with Borrow	1	1	C,DC,N,OV,Z
		CPB	Wb,#lit5	Compare Wb with lit5, with Borrow	1	1	C,DC,N,OV,Z
		CPB	Wb,Ws	Compare W <u>b</u> with Ws, with Borrow (Wb - Ws - C)	1	1	C,DC,N,OV,Z
21	CPSEQ	CPSEQ	Wb, Wn	Compare Wb with Wn, skip if =	1	1 (2 or 3)	None
22	CPSGT	CPSGT	Wb, Wn	Compare Wb with Wn, skip if >	1	1 (2 or 3)	None
23	CPSLT	CPSLT	Wb, Wn	Compare Wb with Wn, skip if <	1	1 (2 or 3)	None
24	CPSNE	CPSNE	Wb, Wn	Compare Wb with Wn, skip if ≠	1	1 (2 or 3)	None
25	DAW	DAW	Wn	Wn = decimal adjust Wn	1	1	С
26	DEC	DEC	f	f = f -1	1	1	C,DC,N,OV,Z
		DEC	f,WREG	WREG = f -1	1	1	C,DC,N,OV,Z
		DEC	Ws,Wd	Wd = Ws - 1	1	1	C,DC,N,OV,Z

TABLE 23-2:	INSTRUCTION SET	<b>OVERVIEW</b>
		01010101

## TABLE 23-2: INSTRUCTION SET OVERVIEW

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of words	# of cycle s	Status Flags Affected
27	DEC2	DEC2 f		f = f -2	1	1	C,DC,N,OV,Z
		DEC2	f,WREG	WREG = f -2	1	1	C,DC,N,OV,Z
		DEC2	Ws,Wd	Wd = Ws - 2	1	1	C,DC,N,OV,Z
28	DISI	DISI	#lit14	Disable Interrupts for k instruction cycles	1	1	None
29	DIV	DIV.S	Wm,Wn	Signed 16/16-bit Integer Divide	1	18	N,Z,C, OV
		DIV.SD	Wm,Wn	Signed 32/16-bit Integer Divide	1	18	N,Z,C, OV
		DIV.U	Wm,Wn	Unsigned 16/16-bit Integer Divide	1	18	N,Z,C, OV
		DIV.UD	Wm,Wn	Unsigned 32/16-bit Integer Divide	1	18	N,Z,C, OV
30	DIVF	DIVF	Wm,Wn	Signed 16/16-bit Fractional Divide	1	18	N,Z,C, OV
31	DO	DO	#lit14,Expr	Do code to PC+Expr, lit14+1 times	2	2	None
		DO	Wn,Expr	Do code to PC+Expr, (Wn)+1 times	2	2	None
32	ED	ED	D Wm*Wm, Acc, Wx, Wy, Wxd Euclidean Distance (no accumulate)		1	1	OA,OB,OAB, SA,SB,SAB
33	EDAC	EDAC	DAC Wm*Wm, Acc, Wx, Wy, Wxd Euclidean Distance		1	1	OA,OB,OAB, SA,SB,SAB
34	EXCH	EXCH	Wns,Wnd	Swap Wns with Wnd	1	1	None
35	FBCL	FBCL	Ws,Wnd	Find Bit Change from Left (MSb) Side	1	1	С
36	FF1L	FF1L	Ws,Wnd	Find First One from Left (MSb) Side	1	1	С
37	FF1R	FF1R	FF1R Ws, Wnd Find First One from Right (LSt		1	1	С
38	GOTO	GOTO	Expr	Go to address	2	2	None
		GOTO	Wn	Go to indirect	1	2	None
39	INC	INC	f	f = f + 1	1	1	C,DC,N,OV,Z
		INC	f,WREG	WREG = f + 1	1	1	C,DC,N,OV,Z
		INC	Ws,Wd	Wd = Ws + 1	1	1	C,DC,N,OV,Z
40	INC2	INC2	f	f = f + 2	1	1	C,DC,N,OV,Z
		INC2	f,WREG	WREG = f + 2	1	1	C,DC,N,OV,Z
		INC2	Ws,Wd	Wd = Ws + 2	1	1	C,DC,N,OV,Z
41	IOR	IOR	f	f = f .IOR. WREG	1	1	N,Z
		IOR	f,WREG	WREG = f .IOR. WREG	1	1	N,Z
		IOR	#lit10,Wn	Wd = lit10 .IOR. Wd	1	1	N,Z
		IOR	Wb,Ws,Wd	Wd = Wb .IOR. Ws	1	1	N,Z
		IOR	Wb,#lit5,Wd	Wd = Wb .IOR. lit5	1	1	N,Z
42	LAC	LAC	Wso,#Slit4,Acc	Load Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
43	LNK	LNK	#lit14	Link frame pointer	1	1	None
44	LSR	LSR	f	f = Logical Right Shift f	1	1	C,N,OV,Z
	LSR f, WREG W		f,WREG	WREG = Logical Right Shift f	1	1	C,N,OV,Z
		LSR	Ws,Wd	Wd = Logical Right Shift Ws	1	1	C,N,OV,Z
		LSR	Wb,Wns,Wnd	Wnd = Logical Right Shift Wb by Wns	1	1	N,Z
		LSR	Wb,#lit5,Wnd	Wnd = Logical Right Shift Wb by lit5	1	1	N,Z
45	MAC	MAC	Wm*Wn,Acc,Wx,Wxd,Wy,Wyd, AWB	Multiply and Accumulate	1	1	OA,OB,OAB, SA,SB,SAB
	MAC Wm*Wm, Acc, Wx, Wxd, Wy, Wyd Square and Accumulate		Square and Accumulate	1	1	OA,OB,OAB, SA,SB,SAB	

TABLE 23-2:	INSTRUCTION SET	OVERVIEW
		•••••••

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of words	# of cycle s	Status Flags Affected
46	MOV	MOV	f,Wn	Move f to Wn	1	1	None
		MOV f		Move f to f	1	1	N,Z
		MOV	f,WREG	Move f to WREG	1	1	N,Z
		MOV	#lit16,Wn	Move 16-bit literal to Wn	1	1	None
		MOV.b	#lit8,Wn	Move 8-bit literal to Wn	1	1	None
		MOV	Wn,f	Move Wn to f	1	1	None
		MOV	Wso,Wdo	Move Ws to Wd	1	1	None
		MOV	WREG, f	Move WREG to f	1	1	N,Z
		MOV.D	Wns,Wd	Move Double from W(ns):W(ns+1) to Wd	1	2	None
		MOV.D	Ws,Wnd	Move Double from Ws to W(nd+1):W(nd)	1	2	None
47	MOVSAC	MOVSAC	Acc,Wx,Wxd,Wy,Wyd,AWB	Prefetch and store accumulator	1	1	None
48	MPY	MPY Wm*Wn,Ac	c,Wx,Wxd,Wy,Wyd	Multiply Wm by Wn to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
		MPY Wm*Wm,Ac	c,Wx,Wxd,Wy,Wyd	Square Wm to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
49	MPY.N	MPY.N	Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	-(Multiply Wm by Wn) to Accumulator	1	1	None
50	MSC	MSC	Wm*Wm,Acc,Wx,Wxd,Wy,Wyd, AWB	Multiply and Subtract from Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
51	MUL	MUL.SS	Wb,Ws,Wnd	{Wnd+1, Wnd} = signed(Wb) * signed(Ws)	1	1	None
		MUL.SU	Wb,Ws,Wnd	{Wnd+1, Wnd} = signed(Wb) * unsigned(Ws)	1	1	None
		MUL.US	Wb,Ws,Wnd	{Wnd+1, Wnd} = unsigned(Wb) * signed(Ws)	1	1	None
		MUL.UU	Wb,Ws,Wnd	{Wnd+1, Wnd} = unsigned(Wb) * unsigned(Ws)	1	1	None
	MUL.SU Wb, #lit5, Wnd {Wnd+1, Wnd} =		{Wnd+1, Wnd} = signed(Wb) * unsigned(lit5)	1	1	None	
		MUL.UU	Wb,#lit5,Wnd	{Wnd+1, Wnd} = unsigned(Wb) * unsigned(lit5)	1	1	None
		MUL	f	W3:W2 = f * WREG	1	1	None
52	NEG	NEG	Acc	Negate Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
		NEG	f	$f = \overline{f} + 1$	1	1	C,DC,N,OV,Z
		NEG	f,WREG	WREG = $\overline{f}$ + 1	1	1	C,DC,N,OV,Z
		NEG	Ws,Wd	$Wd = \overline{Ws} + 1$	1	1	C,DC,N,OV,Z
53	NOP	NOP		No Operation	1	1	None
		NOPR		No Operation	1	1	None
54	POP	POP	f	Pop f from top-of-stack (TOS)	1	1	None
		POP	Wdo	Pop from top-of-stack (TOS) to Wdo	1	1	None
		POP.D	Wnd	Pop from top-of-stack (TOS) to W(nd):W(nd+1)	1	2	None
		POP.S		Pop Shadow Registers	1	1	All
55	PUSH	PUSH	f	Push f to top-of-stack (TOS)	1	1	None
		PUSH	Wso	Push Wso to top-of-stack (TOS)	1	1	None
		PUSH.D	Wns	Push W(ns):W(ns+1) to top-of-stack (TOS)	1	2	None
		PUSH.S		Push Shadow Registers	1	1	None
56	PWRSAV	PWRSAV #lit1		Go into Sleep or Idle mode	1	1	WDTO,Sleep
57	RCALL	RCALL	Expr	Relative Call	1	2	None
		RCALL	Wn	Computed Call	1	2	None
58	REPEAT	REPEAT	#lit14	Repeat Next Instruction lit14+1 times	1	1	None
		REPEAT	Wn	Repeat Next Instruction (Wn)+1 times	1	1	None
59	RESET	RESET		Software device Reset	1	1	None
60	RETFIE	RETFIE		Return from interrupt	1	3 (2)	None
61	RETLW	RETLW	#lit10,Wn	Return with literal in Wn	1	3 (2)	None
62	RETURN	RETURN		Return from Subroutine	1	3 (2)	None

## TABLE 23-2: INSTRUCTION SET OVERVIEW

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of words	# of cycle s	Status Flags Affected
63	RLC	RLC	f	f = Rotate Left through Carry f	1	1	C,N,Z
		RLC	f,WREG	WREG = Rotate Left through Carry f	1	1	C,N,Z
		RLC	Ws,Wd	Wd = Rotate Left through Carry Ws	1	1	C,N,Z
64	RLNC	RLNC	f	f = Rotate Left (No Carry) f	1	1	N,Z
		RLNC	f,WREG	WREG = Rotate Left (No Carry) f	1	1	N,Z
		RLNC	Ws,Wd	Wd = Rotate Left (No Carry) Ws	1	1	N,Z
65	RRC	RRC	f	f = Rotate Right through Carry f	1	1	C,N,Z
		RRC	f,WREG	WREG = Rotate Right through Carry f	1	1	C,N,Z
		RRC	Ws,Wd	Wd = Rotate Right through Carry Ws	1	1	C,N,Z
66	RRNC	RRNC	f	f = Rotate Right (No Carry) f	1	1	N,Z
		RRNC	f,WREG	WREG = Rotate Right (No Carry) f	1	1	N,Z
		RRNC	Ws,Wd	Wd = Rotate Right (No Carry) Ws	1	1	N,Z
67	SAC	SAC	Acc,#Slit4,Wdo	Store Accumulator	1	1	None
		SAC.R	Acc,#Slit4,Wdo	Store Rounded Accumulator	1	1	None
68	SE	SE	Ws,Wnd	Wnd = sign extended Ws	1	1	C,N,Z
69	SETM	SETM	f	f = 0xFFFF	1	1	None
		SETM	WREG	WREG = 0xFFFF	1	1	None
		SETM	Ws	Ws = 0xFFFF	1	1	None
70	SFTAC	SFTAC	Acc,Wn	Arithmetic Shift Accumulator by (Wn)	1	1	OA,OB,OAB, SA,SB,SAB
		SFTAC	Acc,#Slit6	Arithmetic Shift Accumulator by Slit6	umulator by Slit6 1 1		OA,OB,OAB, SA,SB,SAB
71	SL	SL	f	f = Left Shift f	1	1	C,N,OV,Z
		SL	f,WREG	WREG = Left Shift f	1	1	C,N,OV,Z
		SL	Ws,Wd	Wd = Left Shift Ws	1	1	C,N,OV,Z
	SL Wb, Wns, Wnd		Wb,Wns,Wnd	Wnd = Left Shift Wb by Wns	1	1	N,Z
		SL	Wb,#lit5,Wnd	Wnd = Left Shift Wb by lit5	1	1	N,Z
72	SUB	JB SUB Acc		Subtract Accumulators	1	1	OA,OB,OAB, SA,SB,SAB
	SUB $f = f - WREG$		f = f - WREG	1	1	C,DC,N,OV,Z	
		SUB	f,WREG	WREG = f - WREG	1	1	C,DC,N,OV,Z
		SUB	#lit10,Wn	Wn = Wn - lit10		1	C,DC,N,OV,Z
		SUB	Wb,Ws,Wd	Wd = Wb - Ws	1	1	C,DC,N,OV,Z
		SUB	Wb,#lit5,Wd	Wd = Wb - lit5	1	1	C,DC,N,OV,Z
73	SUBB	SUBB	f	$f = f - WREG - (\overline{C})$	1	1	C,DC,N,OV,Z
		SUBB	f,WREG	WREG = $f - WREG - (\overline{C})$	1	1	C,DC,N,OV,Z
		SUBB	#lit10,Wn	$Wn = Wn - lit10 - (\overline{C})$	1	1	C,DC,N,OV,Z
		SUBB	Wb,Ws,Wd	$Wd = Wb - Ws - (\overline{C})$	1	1	C,DC,N,OV,Z
		SUBB	Wb,#lit5,Wd	$Wd = Wb - lit5 - (\overline{C})$	1	1	C,DC,N,OV,Z
74	SUBR	SUBR	f	f = WREG - f	1	1	C,DC,N,OV,Z
		SUBR	f,WREG	WREG = WREG - f	1	1	C,DC,N,OV,Z
		SUBR	Wb,Ws,Wd	Wd = Ws - Wb	1	1	C,DC,N,OV,Z
		SUBR	Wb,#lit5,Wd	Wd = lit5 - Wb	1	1	C,DC,N,OV,Z
75	SUBBR	SUBBR	f	$f = WREG - f - (\overline{C})$	1	1	C,DC,N,OV,Z
		SUBBR	f,WREG	WREG = WREG -f - $(\overline{C})$	1	1	C,DC,N,OV,Z
		SUBBR	Wb,Ws,Wd	$Wd = Ws - Wb - (\overline{C})$	1	1	C,DC,N,OV,Z
		SUBBR	Wb,#lit5,Wd	$Wd = lit5 - Wb - (\overline{C})$	1	1	C,DC,N,OV,Z
76	SWAP	SWAP.b	Wn	Wn = nibble swap Wn	1	1	None
L		SWAP	Wn	Wn = byte swap Wn	1	1	None
77	TBLRDH	TBLRDH	Ws,Wd	Read Prog<23:16> to Wd<7:0>	1	2	None
78	TBLRDL	TBLRDL	Ws,Wd	Read Prog<15:0> to Wd	1	2	None
79	TBLWTH	TBLWTH Ws, Wd Write Ws<7:0> to Prog<23:16>		1	2	None	

TABLE 23-2:	INSTRUCTION SET	OVERVIEW
		•••••••

Base Instr #	Assembly Mnemonic		Assembly Syntax	Description	# of words	# of cycle s	Status Flags Affected
80	TBLWTL	TBLWTL	Ws,Wd	Write Ws to Prog<15:0>	1	2	None
81	ULNK	ULNK		Unlink frame pointer	1	1	None
82	XOR	XOR	f	f = f .XOR. WREG	1	1	N,Z
		XOR	f,WREG	WREG = f .XOR. WREG	1	1	N,Z
		XOR	#lit10,Wn	Wd = lit10 .XOR. Wd	1	1	N,Z
		XOR	Wb,Ws,Wd	Wd = Wb .XOR. Ws	1	1	N,Z
		XOR	Wb,#lit5,Wd	Wd = Wb .XOR. lit5	1	1	N,Z
83	ZE	ZE	Ws,Wnd	Wnd = Zero-Extend Ws	1	1	C,Z,N

# dsPIC30F6010

NOTES:

# 24.0 ELECTRICAL CHARACTERISTICS

This section provides an overview of dsPIC30F electrical characteristics. Additional information will be provided in future revisions of this document as it becomes available.

For detailed information about the dsPIC30F architecture and core, refer to "dsPIC30F Family Reference Manual" (DS70046).

Absolute maximum ratings for the dsPIC30F family are listed below. Exposure to these maximum rating conditions for extended periods may affect device reliability. Functional operation of the device at these or any other conditions above the parameters indicated in the operation listings of this specification is not implied.

# Absolute Maximum Ratings<sup>(†)</sup>

Ambient temperature under bias	40°C to +125°C
Storage temperature	65°C to +150°C
Voltage on any pin with respect to Vss (except VDD and MCLR) (Note 1)	
Voltage on VDD with respect to VSS	0.3V to +5.5V
Voltage on MCLR with respect to Vss	
Maximum current out of Vss pin	
Maximum current into VDD pin (Note 2)	
Input clamp current, Iık (VI < 0 or VI > VDD)	±20 mA
Output clamp current, IOK (VO < 0 or VO > VDD)	±20 mA
Maximum output current sunk by any I/O pin	
Maximum output current sourced by any I/O pin	
Maximum current sunk by all ports	
Maximum current sourced by all ports (Note 2)	200 mA

- Note 1: Voltage spikes below Vss at the MCLR/VPP pin, inducing currents greater than 80 mA, may cause latchup. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the MCLR/VPP pin, rather than pulling this pin directly to Vss.
  - 2: Maximum allowable current is a function of device maximum power dissipation. See Table 24-4.

<sup>†</sup>NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

## 24.1 DC Characteristics

**Note:** All peripheral electrical characteristics are specified. For exact peripherals available on specific devices, please refer the Family Cross Reference Table.

VDD Range	Temp Range	Max MIPS				
(in Volts)	(in °C)	dsPIC30F6010-30I	dsPIC30F6010-20I	dsPIC30F6010-20E		
4.75-5.5V	-40°C to +85°C	30	20	—		
4.75-5.5V	-40°C to +125°C	—	—	20		
3.0-3.6V	-40°C to +85°C	15	10	—		
3.0-3.6V	-40°C to +125°C	—	—	10		
2.5-3.0V	-40°C to +85°C	7.5	7.5	_		

#### TABLE 24-1: OPERATING MIPS VS. VOLTAGE

#### TABLE 24-2: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min	Тур	Мах	Unit
dsPIC30F6010-30I					
Operating Junction Temperature Range	TJ	-40		+125	°C
Operating Ambient Temperature Range	T <sub>A</sub>	-40		+85	°C
dsPIC30F6010-20I					
Operating Junction Temperature Range	TJ	-40		+150	°C
Operating Ambient Temperature Range	T <sub>A</sub>	-40		+85	°C
dsPIC30F6010-20E					
Operating Junction Temperature Range	TJ	-40		+150	°C
Operating Ambient Temperature Range	T <sub>A</sub>	-40		+125	°C
Power Dissipation: Internal chip power dissipation: $P_{INT} = V_{DD} \times (I_{DD} - \sum I_{OH})$ I/O Pin power dissipation: $_{I/O} = \sum (\{V_{DD} - V_{OH}\} \times I_{OH}) + \sum (V_{OL} \times I_{OL})$	PD	PINT + PI/O		W	
Maximum Allowed Power Dissipation	Pdmax	(	TJ - TA)/θJ	A	W

#### TABLE 24-3: THERMAL PACKAGING CHARACTERISTICS

Characteristic	Symbol	Тур	Max	Unit	Notes
Package Thermal Resistance, 80-pin TQFP (14x14x1mm)	θja	50		°C/W	1
Package Thermal Resistance, 64-pin TQFP (14x14x1mm)	θја	50		°C/W	1

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

#### TABLE 24-4: DC TEMPERATURE AND VOLTAGE SPECIFICATIONS

DC CHA	$\begin{tabular}{lllllllllllllllllllllllllllllllllll$					<b>s: 2.5V to 5.5V</b> ≤ TA ≤ +85°C for Industrial ≤ TA ≤ +125°C for Extended	
Param No.	Symbol	Characteristic	Min Typ <sup>(1)</sup> Max U			Units	Conditions
Operatir	ng Voltage	(2)					
DC10	Vdd	Supply Voltage	2.5	—	5.5	V	Industrial temperature
DC11	Vdd	Supply Voltage	2.5	_	5.5	V	Extended temperature
DC12	Vdr	RAM Data Retention Voltage <sup>(3)</sup>		1.5	—	V	
DC16	VPOR	VDD Start Voltage to ensure internal Power-on Reset signal		Vss	_	V	
DC17	SVDD	<b>VDD Rise Rate</b> to ensure internal Power-on Reset signal	0.05			V/ms	0-5V in 0.1 sec 0-3V in 60 ms

**Note 1:** Data in "Typ" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: These parameters are characterized but not tested in manufacturing.

**3:** This is the limit to which VDD can be lowered without losing RAM data.

			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated)					
DC CHARACT	ERISTICS		Operating te	emperature	$-40^{\circ}C \le TA \le +$	85°C for Industrial		
Parameter No.	Typical <sup>(1)</sup>	Мах	Units	Conditions				
Operating Cur	rent (IDD) <sup>(2)</sup>							
DC31a	7.1	11	mA	25°C				
DC31b	6.8	11	mA	85°C	3.3V			
DC31c	6.6	11	mA	125°C		0.128 MIPS		
DC31e	14	20	mA	25°C		LPRC (512 kHz)		
DC31f	14	20	mA	85°C	5V			
DC31g	13	20	mA	125°C				
DC30a	14	21	mA	25°C				
DC30b	14	21	mA	85°C	3.3V			
DC30c	14	21	mA	125°C		(1.8 MIPS)		
DC30e	28	44	mA	25°C		FRC (7.37 MHz)		
DC30f	27	44	mA	85°C	5V			
DC30g	27	44	mA	125°C				
DC23a	30	47	mA	25°C				
DC23b	30	47	mA	85°C	3.3V			
DC23c	31	47	mA	125°C				
DC23e	37	60	mA	25°C		4 MIFS		
DC23f	40	60	mA	85°C	5V			
DC23g	40	60	mA	125°C				
DC24a	49	74	mA	25°C				
DC24b	49	74	mA	85°C	3.3V			
DC24c	49	74	mA	125°C				
DC24e	82	120	mA	25°C		10 MIPS		
DC24f	81	120	mA	85°C	5V			
DC24g	81	120	mA	125°C				
DC27a	88	120	mA	25°C	2.2)/			
DC27b	88	120	mA	85°C	3.3V			
DC27d	138	190	mA	25°C		20 MIPS		
DC27e	142	190	mA	85°C	5V			
DC27f	137	190	mA	125°C				
DC29a	203	255	mA	25°C	5\/	20 MIDS		
DC29b	200	255	mA	85°C	50			

#### TABLE 24-5: DC CHARACTERISTICS: OPERATING CURRENT (IDD)

**Note 1:** Data in "Typical" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature also have an impact on the current consumption. The test conditions for all IDD measurements are as follows: OSC1 driven with external square wave from rail to rail. All I/O pins are configured as Inputs and pulled to VDD. MCLR = VDD, WDT, FSCM, LVD and BOR are disabled. CPU, SRAM, Program Memory and Data Memory are operational. No peripheral modules are operating.

# dsPIC30F6010

#### TABLE 24-6: DC CHARACTERISTICS: IDLE CURRENT (IIDLE)

			Standard Operating Conditions: 2.5V to 5.5V						
DC CHARACT	ERISTICS		(unless oth	Unitess otherwise stated) Operating temperature $-40^{\circ}C < TA < +85^{\circ}C$ for industrial					
			Operating te	$-40^{\circ}\text{C} \le T\text{A} \le +125^{\circ}\text{C}$ for Extended					
Parameter No.	Typical <sup>(1,2)</sup>	Мах	Units	Conditions					
Operating Cur	rent (IDD)								
DC51a	6.7	10	mA	25°C					
DC51b	6.3	10	mA	85°C	3.3V				
DC51c	6.1	10	mA	125°C	1	0.128 MIPS			
DC51e	13	18	mA	25°C		LPRC (512 kHz)			
DC51f	13	18	mA	85°C	5V				
DC51g	13	18	mA	125°C	1				
DC50a	11	15	mA	25°C					
DC50b	10	15	mA	85°C	3.3V				
DC50c	10	15	mA	125°C	1	(1.8 MIPS)			
DC50e	23	35	mA	25°C		FRC (7.37 MHz)			
DC50f	21	35	mA	85°C	5V				
DC50g	21	35	mA	125°C	1				
DC43a	17	26	mA	25°C					
DC43b	16	26	mA	85°C	3.3V				
DC43c	16	26	mA	125°C	1				
DC43e	31	44	mA	25°C		4 MIFS			
DC43f	28	44	mA	85°C	5V				
DC43g	28	44	mA	125°C	1				
DC44a	31	45	mA	25°C					
DC44b	31	45	mA	85°C	3.3V				
DC44c	31	45	mA	125°C	1				
DC44e	53	69	mA	25°C		TO MIPS			
DC44f	52	69	mA	85°C	5V				
DC44g	52	69	mA	125°C	1				
DC47a	54	70	mA	25°C	2.21/				
DC47b	54	70	mA	85°C	3.3V				
DC47d	89	110	mA	25°C	1	20 MIPS			
DC47e	94	110	mA	85°C	5V				
DC47f	89	110	mA	125°C	1				
DC49a	125	145	mA	25°C	5\/	20 MIDS			
DC49b	124	145	mA	85°C	57	30 101153			

**Note 1:** Data in "Typical" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Base IIDLE current is measured with Core off, Clock on and all modules turned off.

DC CHARACT	ERISTICS		Standard C (unless oth Operating to	Operating Con Derwise state emperature	nditions: 2.5 d) -40°C ≤ TA ≤ -40°C ≤ TA ≤	W <b>to 5.5V</b> ≤ +85°C for Industrial ≤ +125°C for Extended			
Parameter No.	Typical <sup>(1)</sup>	Мах	Units		Conditions				
Power Down	Current (IPD)								
DC60a	0.5	—	μΑ	25°C					
DC60b	2.8	60	μΑ	85°C	3.3V				
DC60c	24	120	μΑ	125°C		Rang Rower Down Current <sup>(2)</sup>			
DC60e	1	_	μΑ	25°C		Base Power Down Current			
DC60f	4.4	110	μΑ	85°C	5V				
DC60g	36	180	μΑ	125°C					
DC61a	10	16	μΑ	25°C					
DC61b	10	16	μΑ	85°C	3.3V				
DC61c	9	16	μΑ	125°C		Watehdag Timor Current: Alwot(3)			
DC61e	19	30	μΑ	25°C					
DC61f	18	30	μΑ	85°C	5V				
DC61g	17	30	μΑ	125°C					
DC62a	4	10	μΑ	25°C					
DC62b	5	10	μΑ	85°C	3.3V				
DC62c	4	10	μΑ	125°C		Timor $1 \frac{1}{2} \frac{1}{2} \frac{1}{3}$			
DC62e	4	15	μΑ	25°C					
DC62f	6	15	μΑ	85°C	5V				
DC62g	5	15	μΑ	125°C					
DC63a	33	55	μΑ	25°C					
DC63b	34	55	μΑ	85°C	3.3V				
DC63c	36	55	μΑ	125°C		ROD On: Alpon <sup>(3)</sup>			
DC63e	38	65	μΑ	25°C		BOR OII. AIBOR			
DC63f	40	65	μΑ	85°C	5V				
DC63g	39	65	μΑ	125°C					
DC66a	20	40	μΑ	25°C					
DC66b	22	40	μΑ	85°C	3.3V				
DC66c	22	40	μΑ	125°C		Low Voltago Dotact: Ally (3)			
DC66e	24	50	μΑ	25°C					
DC66f	25	50	μΑ	85°C	5V				
DC66g	24	50	μΑ	125°C					

**Note 1:** Data in the Typical column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

**2:** Base IPD is measured with all peripherals and clocks shut down. All I/Os are configured as inputs and pulled high. LVD, BOR, WDT, etc. are all switched off.

3: The  $\Delta$  current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

# dsPIC30F6010

TABLE 24-8:	DC CHARACTERISTICS: I/O PIN INPUT SPECIFICATIONS

DC CHARACTERISTICS			Standard (unless Operatin	d Opera otherwi g tempe	ting Con se stated erature -	ditions: ) 40°C ≤	<b>2.5V to 5.5V</b> TA $\leq$ +85°C for Industrial		
Param No.	Symbol	Characteristic	Min	$\begin{tabular}{ c c c c c } -40^{\circ}C \leq TA \leq +125^{\circ}C \mbox{ for Extended} \\ \hline Min & Typ^{(1)} & Max & Units & Conditions \\ \end{tabular}$					
	VIL	Input Low-Voltage <sup>(2)</sup>							
DI10		I/O pins: with Schmitt Trigger buffer	Vss	_	0.2 Vdd	V			
DI15		MCLR	Vss		0.2 Vdd	V			
DI16		OSC1 (in XT, HS and LP modes)	Vss	_	0.2 Vdd	V			
DI17		OSC1 (in RC mode) <sup>(3)</sup>	Vss		0.3 Vdd	V			
DI18		SDA, SCL	Vss	_	0.3 Vdd	V	SMbus disabled		
DI19		SDA, SCL	Vss		0.2 Vdd	V	SMbus enabled		
	Viн	Input High Voltage <sup>(2)</sup>							
DI20		I/O pins: with Schmitt Trigger buffer	0.8 Vdd	_	Vdd	V			
DI25		MCLR	0.8 Vdd		Vdd	V			
DI26		OSC1 (in XT, HS and LP modes)	0.7 Vdd		Vdd	V			
DI27		OSC1 (in RC mode) <sup>(3)</sup>	0.9 Vdd		Vdd	V			
DI28		SDA, SCL	0.7 Vdd		Vdd	V	SMbus disabled		
DI29		SDA, SCL	0.8 Vdd		Vdd	V	SMbus enabled		
	ICNPU	CNxx Pull-up Current <sup>(2)</sup>							
DI30			50	250	400	μA	VDD = 5V, VPIN = VSS		
	lı∟	Input Leakage Current <sup>(2)(4)(5)</sup>							
DI50		I/O ports	—	0.01	±1	μA	$Vss \le VPIN \le VDD,$ Pin at high-impedance		
DI51		Analog Input Pins	_	0.50	-	μA	$Vss \le VPIN \le VDD,$ Pin at high-impedance		
DI55		MCLR	—	0.05	±5	μA	$VSS \leq VPIN \leq VDD$		
DI56		OSC1	—	0.05	±5	μA	Vss $\leq$ VPIN $\leq$ VDD, XT, HS and LP Osc mode		

Note 1: Data in "Typ" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: These parameters are characterized but not tested in manufacturing.

**3:** In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the dsPIC30F device be driven with an external clock while in RC mode.

4: The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

5: Negative current is defined as current sourced by the pin.

DC CHARACTERISTICS			Standard ( (unless of Operating	<b>Operatii</b> herwise tempera	ng Cond stated ature - -	ditions: ) 40°C ≤ <sup>-</sup> 40°C ≤ <sup>-</sup>	<b>2.5V to 5.5V</b> TA $\leq$ +85°C for Industrial TA $\leq$ +125°C for Extended		
Param No.	Symbol	Characteristic	Min	Min Typ <sup>(1)</sup> Max Units Conditions					
-	Vol	Output Low-Voltage <sup>(2)</sup>							
DO10		I/O ports	—	—	0.6	V	IOL = 8.5 mA, VDD = 5V		
			—	—	TBD	V	IOL = 2.0 mA, VDD = 3V		
DO16		OSC2/CLKO	—	—	0.6	V	IOL = 1.6 mA, VDD = 5V		
		(RC or EC Osc mode)	—	—	TBD	V	IOL = 2.0 mA, VDD = 3V		
	Vон	Output High Voltage <sup>(2)</sup>							
DO20		I/O ports	Vdd - 0.7	—	—	V	IOH = -3.0 mA, VDD = 5V		
			TBD	—	—	V	IOH = -2.0 mA, VDD = 3V		
DO26		OSC2/CLKO	Vdd - 0.7	—	—	V	IOH = -1.3 mA, VDD = 5V		
		(RC or EC Osc mode)	TBD	—	_	V	IOH = -2.0 mA, VDD = 3V		
		Capacitive Loading Specs on Output Pins <sup>(2)</sup>							
DO50	Cosc2	OSC2/SOSC2 pin	_	_	15	pF	In XTL, XT, HS and LP modes when external clock is used to drive OSC1.		
DO56	Сю	All I/O pins and OSC2	—	—	50	pF	RC or EC Osc mode		
DO58	Св	SCL, SDA	—	—	400	pF	In l <sup>2</sup> C™ mode		

#### TABLE 24-9: DC CHARACTERISTICS: I/O PIN OUTPUT SPECIFICATIONS

**Note 1:** Data in "Typ" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: These parameters are characterized but not tested in manufacturing.

#### FIGURE 24-1: LOW-VOLTAGE DETECT CHARACTERISTICS



#### TABLE 24-10: ELECTRICAL CHARACTERISTICS: LVDL

DC CHARACTERISTICS		$\begin{tabular}{lllllllllllllllllllllllllllllllllll$							
Param No.	Symbol	Characteristic <sup>(</sup>	1)	Min	Тур	Max	Units	Conditions	
LV10	Vplvd	LVDL Voltage on VDD transition high to low	LVDL = 0000 <sup>(2)</sup>		_		V		
			LVDL = 0001 <sup>(2)</sup>	_			V		
			LVDL = 0010 <sup>(2)</sup>	_			V		
			LVDL = 0011 <sup>(2)</sup>	—	_	—	V		
			LVDL = 0100	2.50		2.65	V		
			LVDL = 0101	2.70	_	2.86	V		
			LVDL = 0110	2.80	_	2.97	V		
			LVDL = 0111	3.00	_	3.18	V		
			LVDL = 1000	3.30	_	3.50	V		
			LVDL = 1001	3.50	_	3.71	V		
			LVDL = 1010	3.60		3.82	V		
			LVDL = 1011	3.80	_	4.03	V		
			LVDL = 1100	4.00	_	4.24	V		
			LVDL = 1101	4.20		4.45	V		
			LVDL = 1110	4.50	_	4.77	V		
LV15	VLVDIN	External LVD input pin threshold voltage	LVDL = 1111		—	_	V		

**Note 1:** These parameters are characterized but not tested in manufacturing.

2: These values not in usable operating range.

## FIGURE 24-2: BROWN-OUT RESET CHARACTERISTICS



#### TABLE 24-11: ELECTRICAL CHARACTERISTICS: BOR

DC CHAR	ACTERIST	ICS	Standard Opera (unless otherw Operating temp	ating Co ise state erature	nditions ed) -40°C ≤ -40°C ≤	s: <b>2.5V t</b> e ≤ Ta ≤ +8 ≤ Ta ≤ +1	<b>5°C</b> for 25°C for	Industrial r Extended
Param No.	Symbol	Character	istic	Min	Typ <sup>(1)</sup>	Max	Units	Conditions
BO10	VBOR	BOR Voltage <sup>(2)</sup> on VDD transition high to	BORV = 11 <sup>(3)</sup>		_	_	V	Not in operating range
		low	BORV = 10	2.6		2.71	V	
			BORV = 01	4.1		4.4	V	
			BORV = 00	4.58		4.73	V	
BO15	VBHYS			_	5		mV	

**Note 1:** Data in "Typ" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: These parameters are characterized but not tested in manufacturing.

**3:** '11' values not in usable operating range.

#### TABLE 24-12: DC CHARACTERISTICS: PROGRAM AND EEPROM

			Standard Operating Conditions: 2.5V to 5.5V							
DC CHA	RACTER	STICS	(unless Operati	<b>s otherw</b> ina temp	erature	<b>d)</b> -40°C <	0°C ≤ Ta ≤ +85°C for Industrial			
			5 T T T			$-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended				
Param No.	Symbol	Characteristic	Min	Typ <sup>(1)</sup>	Max	Units	Conditions			
		Data EEPROM Memory <sup>(2)</sup>								
D120	ED	Byte Endurance	100K	1M	_	E/W	$-40^{\circ}C \le TA \le +85^{\circ}C$			
D121	Vdrw	VDD for Read/Write	Vmin	_	5.5	V	Using EECON to read/write VMIN = Minimum operating voltage			
D122	TDEW	Erase/Write Cycle Time	—	2	—	ms				
D123	TRETD	Characteristic Retention	40	100	—	Year	Provided no other specifications are violated			
D124	IDEW	IDD During Programming	—	10	30	mA	Row Erase			
		Program FLASH Memory <sup>(2)</sup>								
D130	Eр	Cell Endurance	10K	100K	_	E/W	$-40^{\circ}C \le TA \le +85^{\circ}C$			
D131	Vpr	VDD for Read	VMIN	—	5.5	V	VMIN = Minimum operating voltage			
D132	VEB	VDD for Bulk Erase	4.5	—	5.5	V				
D133	VPEW	VDD for Erase/Write	3.0	—	5.5	V				
D134	TPEW	Erase/Write Cycle Time	—	2	_	ms				
D135	TRETD	Characteristic Retention	40	100	—	Year	Provided no other specifications are violated			
D136	Тев	ICSP Block Erase Time	—	4	—	ms				
D137	IPEW	IDD During Programming	—	10	30	mA	Row Erase			
D138	IEB	IDD During Programming	—	10	30	mA	Bulk Erase			

**Note 1:** Data in "Typ" column is at 5V, 25°C unless otherwise stated.

**2:** These parameters are characterized but not tested in manufacturing.

# 24.2 AC Characteristics and Timing Parameters

The information contained in this section defines dsPIC30F AC characteristics and timing parameters.

#### TABLE 24-13: TEMPERATURE AND VOLTAGE SPECIFICATIONS - AC

	Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated)
AC CHARACTERISTICS	Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial
	-40°C $\leq$ TA $\leq$ +125°C for Extended
	Operating voltage VDD range as described in DC Spec Section 24.1.

#### FIGURE 24-3: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS



#### FIGURE 24-4: EXTERNAL CLOCK TIMING



AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended				
Param No.	Symb ol	Characteristic	Min	Typ <sup>(1)</sup>	Max	Units	Conditions
OS10	Fosc	External CLKI Frequency <sup>(2)</sup> (External clocks allowed only in EC mode)	DC 4 4 4	 	40 10 10 7.5	MHz MHz MHz MHz	EC EC with 4x PLL EC with 8x PLL EC with 16x PLL
		Oscillator Frequency-	DC 0.4 4 4 4 4 10 31 —	   7.37 512	4 10 10 7.5 25 33 —	MHZ MHZ MHZ MHZ MHZ MHZ KHZ KHZ	XTL XT XT with 4x PLL XT with 8x PLL XT with 16x PLL HS LP FRC internal LPRC internal
OS20	Tosc	Tosc = 1/Fosc	—	—	—		See parameter OS10 for Fosc value
OS25	TCY	Instruction Cycle Time <sup>(2)(3)</sup>	33	—	DC	ns	See Table
OS30	TosL, TosH	External Clock <sup>(2)</sup> in (OSC1) High or Low Time	.45 x Tosc	—	—	ns	EC
OS31	TosR, TosF	External Clock <sup>(2)</sup> in (OSC1) Rise or Fall Time	—	—	20	ns	EC
OS40	TckR	CLKO Rise Time <sup>(2)(4)</sup>	—	6	10	ns	
OS41	TckF	CLKO Fall Time <sup>(2)(4)</sup>	_	6	10	ns	

#### TABLE 24-14: EXTERNAL CLOCK TIMING REQUIREMENTS

**Note 1:** Data in "Typ" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: These parameters are characterized but not tested in manufacturing.

- 3: Instruction cycle period (TCY) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "Min." values with an external clock applied to the OSC1/CLKI pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.
- 4: Measurements are taken in EC or ERC modes. The CLKO signal is measured on the OSC2 pin. CLKO is low for the Q1-Q2 period (1/2 TCY) and high for the Q3-Q4 period (1/2 TCY).

#### TABLE 24-15: PLL CLOCK TIMING SPECIFICATIONS (VDD = 2.5 TO 5.5 V)

АС СНА	RACTERI	STICS	Standard Operating Conditions: 2.5V to 5.5V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended					
Param No. Symbol Characterist			ic <sup>(1)</sup>	Min	Тур <sup>(2)</sup>	Max	Units	Conditions
OS50	Fplli	PLL Input Frequency	/ Range <sup>(2)</sup>	4 4 4 4 4 4		10 10 7.5 <sup>(3)</sup> 10 10 7.5 <sup>(3)</sup>	MHz MHz MHz MHz MHz MHz	EC with 4x PLL EC with 8x PLL EC with 16x PLL XT with 4x PLL XT with 8x PLL XT with 16x PLL
OS51	Fsys	On-Chip PLL Output	(2)	16		120	MHz	EC, XT modes with PLL
OS52	TLOC	PLL Start-up Time (L	ock Time)	_	20	50	μs	

**Note 1:** These parameters are characterized but not tested in manufacturing.

2: Data in "Typ" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

3: Limited by device operating frequency range.

#### TABLE 24-16: PLL JITTER

AC CHARACTERISTICS		$\begin{array}{ll} \mbox{Standard Operating Conditions: 2.5V to 5.5V} \\ \mbox{(unless otherwise stated)} \\ \mbox{Operating temperature} & -40^\circ C \leq TA \leq +85^\circ C \mbox{ for Industrial} \\ -40^\circ C \leq TA \leq +125^\circ C \mbox{ for Extended} \end{array}$							
Param No.	Characteristic	Min	Typ <sup>(1)</sup>	Max	Units	Conditions			
OS61	x4 PLL	—	0.251	0.413	%	$-40^{\circ}C \le TA \le +85^{\circ}C$	VDD = 3.0 to 3.6V		
		—	0.251	0.413	%	$-40^{\circ}C \le TA \le +125^{\circ}C$	VDD = 3.0 to 3.6V		
		—	0.256	0.47	%	$-40^{\circ}C \le TA \le +85^{\circ}C$	VDD = 4.5 to 5.5V		
		—	0.256	0.47	%	$-40^{\circ}C \leq TA \leq +125^{\circ}C$	VDD = 4.5 to 5.5V		
	x8 PLL		0.355	0.584	%	$-40^{\circ}C \le TA \le +85^{\circ}C$	VDD = 3.0 to 3.6V		
		_	0.355	0.584	%	$-40^{\circ}C \leq TA \leq +125^{\circ}C$	VDD = 3.0 to 3.6V		
		_	0.362	0.664	%	$\textbf{-40^{\circ}C} \leq \textbf{TA} \leq \textbf{+85^{\circ}C}$	VDD = 4.5 to 5.5V		
		—	0.362	0.664	%	$-40^{\circ}C \leq TA \leq +125^{\circ}C$	VDD = 4.5 to 5.5V		
	x16 PLL	_	0.67	0.92	%	$-40^{\circ}C \le TA \le +85^{\circ}C$	VDD = 3.0 to 3.6V		
			0.632	0.956	%	$-40^{\circ}C \le TA \le +85^{\circ}C$	VDD = 4.5 to 5.5V		
		_	0.632	0.956	%	$-40^{\circ}C \le TA \le +125^{\circ}C$	VDD = 4.5 to 5.5V		

Note	1:	These parameters	are characterized	l but not tested ir	n manufacturing.	
	-		-			
-----------------------------	------------------------------	---	--------------------------------	---------------------------------	---------------------------------	----------------------------------
Clock Oscillator Mode	Fosc (MHz) <sup>(1)</sup>	Τ <b>ϲ</b> Υ (μ <b>sec)<sup>(2)</sup></b>	MIPS <sup>(3)</sup> w/o PLL	MIPS <sup>(3)</sup> w PLL x4	MIPS <sup>(3)</sup> w PLL x8	MIPS <sup>(3)</sup> w PLL x16
EC	0.200	20.0	0.05		_	—
	4	1.0	1.0	4.0	8.0	16.0
	10	0.4	2.5	10.0	20.0	—
	25	0.16	6.25	—	—	—
ХТ	4	1.0	1.0	4.0	8.0	16.0
	10	0.4	2.5	10.0	20.0	_

TABLE 24-17: INTERNAL CLOCK TIMING EXAMPLES

**Note 1:** Assumption: Oscillator Postscaler is divide by 1.

**2:** Instruction Execution Cycle Time: TCY = 1/MIPS.

**3:** Instruction Execution Frequency: MIPS = (Fosc \* PLLx)/4 [since there are 4 Q clocks per instruction cycle].

#### TABLE 24-18: AC CHARACTERISTICS: INTERNAL RC ACCURACY

AC CHA	RACTERISTICS	Standard Operating Conditions: 2.5V to 5.5V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended											
Param No.	Characteristic	Min	Тур	Max	Units	Conditions							
	Internal FRC Jitter @ FRC Freq. = 7.37 MHz <sup>(1)</sup>												
OS62	FRC	_	<u>+</u> 0.04	<u>+</u> 0.16	%	$\text{-40°C} \leq \text{TA} \leq \text{+85°C}$	VDD = 3.0-3.6V						
			<u>+</u> 0.07	<u>+</u> 0.23	%	$-40^{\circ}C \leq TA \leq +125^{\circ}C$	VDD = 4.5-5.5V						
	Internal FRC Accuracy @	FRC Fr	eq. = 7.3	7 MHz <sup>(1)</sup>									
OS63	FRC			<u>+</u> 1.50	%	$-40^{\circ}C \leq TA \leq +125^{\circ}C$	VDD = 3.0-5.5V						
	Internal FRC Drift @ FRC	C Freq. =	7.37 MH	z <sup>(1)</sup>									
OS64		-0.7		0.5	%	$\textbf{-40^{\circ}C} \leq \text{TA} \leq \textbf{+85^{\circ}C}$	VDD = 3.0-3.6V						
		-0.7		0.7	%	$-40^{\circ}C \le TA \le +125^{\circ}C \qquad VDD = 3.0^{\circ}$							
		-0.7		0.5	%	% $-40^{\circ}C \le TA \le +85^{\circ}C$ VDD = 4.5-5.							
		-0.7		0.7	%	$-40^{\circ}C \le TA \le +125^{\circ}C$	VDD = 4.5 - 5.5V						

**Note 1:** Frequency calibrated at 7.372 MHz ±2%, 25°C and 5V. TUN <3:0> bits can be used to compensate for temperature drift.

2: Overall FRC variation can be calculated by adding the absolute values of jitter, accuracy and drift percentages.

#### TABLE 24-19: INTERNAL RC ACCURACY

AC CHA	RACTERISTICS	Standard Operating Conditions: 2.5V to 5.5V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended								
Param No.	Characteristic	Min	Тур	Max	Units	Conditions				
LPRC @ Freq. = 512 kHz <sup>(1)</sup>										
OS65		-35 — +35 % —								

**Note 1:** Change of LPRC frequency as VDD changes.

#### FIGURE 24-5: **CLKO AND I/O TIMING CHARACTERISTICS**



TABLE 24-20:	CLKO AND I/O TIMING	REQUIREMENTS
		Standard Operating Conditions: 2.5V to 5.5V

AC CHARACTERISTICS			(unless otherwise stated) Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended					
Param No.	Symbol	Characteristi	Min	Typ <sup>(4)</sup>	Max	Units	Conditions	
DO31	TIOR	Port output rise time		_	7	20	ns	_
DO32	TIOF	Port output fall time		_	7	20	ns	
DI35	TINP	INTx pin high or low time (output)		20	_	_	ns	
DI40	TRBP	CNx high or low time	2 TCY	_	_		_	

Note 1: These parameters are asynchronous events not related to any internal clock edges

2: Measurements are taken in RC mode and EC mode where CLKO output is 4 x Tosc.

3: These parameters are characterized but not tested in manufacturing.

4: Data in "Typ" column is at 5V, 25°C unless otherwise stated.





### TABLE 24-21: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET TIMING REQUIREMENTS

				Standard Operating Conditions: 2.5V to 5.5V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended						
Param No.	Symbol	Characteristic <sup>(1)</sup>	Min Typ <sup>(2)</sup> Max Units Condition							
SY10	TmcL	MCLR Pulse Width (low)	2	—		μs	-40°C to +85°C			
SY11	TPWRT	RT Power-up Timer Period		4 16 64	6 22 90	ms	-40°C to +85°C User programmable			
SY12	TPOR	Power On Reset Delay	3	10	30	μs	-40°C to +85°C			
SY13	Tioz	I/O High-impedance from MCLR Low or Watchdog Timer Reset	_	0.8	1.0	μs				
SY20	TWDT1	Watchdog Timer Time-out Period (No Prescaler)	1.4	2.1	2.8	ms	VDD = 5V, -40°C to +85°C			
	Twdt2		1.4	2.1	2.8	ms	VDD = $3V$ , $-40^{\circ}C$ to $+85^{\circ}C$			
SY25	TBOR	Brown-out Reset Pulse Width <sup>(3)</sup>	100	_	_	μs	$VDD \leq VBOR$ (D034)			
SY30	Tost	Oscillation Start-up Timer Period	—	1024 Tosc	_		Tosc = OSC1 period			
SY35	TFSCM	Fail-Safe Clock Monitor Delay	—	500	900	μs	-40°C to +85°C			

**Note 1:** These parameters are characterized but not tested in manufacturing.

2: Data in "Typ" column is at 5V, 25°C unless otherwise stated.

**3:** Refer to Figure 24-2 and Table 24-11 for BOR.

#### FIGURE 24-7: BAND GAP START-UP TIME CHARACTERISTICS



#### TABLE 24-22: BAND GAP START-UP TIME REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended					
Param No.	Symbol	Characteristic <sup>(1)</sup>	Min Typ <sup>(2)</sup> Max Units Conditions					
SY40	Tbgap	Band Gap Start-up Time		40	65	μs	Defined as the time between the instant that the band gap is enabled and the moment that the band gap reference voltage is stable. RCON<13>Status bit	

Note 1: These parameters are characterized but not tested in manufacturing.

2: Data in "Typ" column is at 5V, 25°C unless otherwise stated.

#### FIGURE 24-8: TIMER 1, 2, 3, 4 AND 5 EXTERNAL CLOCK TIMING CHARACTERISTICS



AC CHARACTERISTICS				Standard Operating Conditions: 2.5V to 5.5V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended							
Param No.	Symbol	Charact	eristic		Min	Тур	Max	Units	Conditions		
TA10	ТтхН	TxCK High Time	Synchron no presc	nous, aler	0.5 TCY + 20		—	ns	Must also meet parameter TA15		
			Synchro with pres	nous, scaler	10		—	ns			
			Asynchro	onous	10	_	—	ns			
TA11	TTXL	TxCK Low Time	Synchronous, no prescaler		Synchronous, no prescaler		0.5 Tcy + 20		—	ns	Must also meet parameter TA15
			Synchron with pres	nous, scaler	10		_	ns			
			Asynchro	onous	10	_	—	ns			
TA15	ΤτχΡ	TxCK Input Period	Synchron no presc	nous, aler	Tcy + 10	_	—	ns			
			Synchron with pres	nous, scaler	Greater of: 20 ns or (Tcy + 40)/N		_	_	N = prescale value (1, 8, 64, 256)		
			Asynchro	onous	20	_	—	ns			
OS60	Ft1	SOSC1/T1CK oscil frequency range (o by setting bit TCS (	lator input scillator enabled T1CON, bit 1))		DC		50	kHz			
TA20	TCKEXTMRL	Delay from Externa	al TxCK Cl	lock	0.5 TCY		1.5 Toy	-			

#### TABLE 24-23: TIMER1 EXTERNAL CLOCK TIMING REQUIREMENTS

Note: Timer1 is a Type A.

#### TABLE 24-24: TIMER2 AND TIMER4 EXTERNAL CLOCK TIMING REQUIREMENTS

Standard Operating Conditions: 2.5V to 5.5V

AC CHARACTERISTICS			Standa (unles Opera	ard Operating s otherwise st ting temperatur	Condition tated) re -40° -40°	$C \le TA \le C$ $C \le TA \le C$	′ <b>to 5.5V</b> +85°C fo +125°C f	r Industrial or Extended	
Param No.	Symbol	Characte	teristic		Min	Тур	Max	Units	Conditions
TB10	TtxH	TxCK High Time	Synchro no preso	onous, caler	0.5 Tcy + 20			ns	Must also meet parameter TB15
			Synchronous, with prescaler		10		_	ns	
TB11	TtxL	TxCK Low Time	Synchro no preso	nous, caler	0.5 TCY + 20	_		ns	Must also meet parameter TB15
			Synchro with pre	onous, scaler	10		—	ns	
TB15	TtxP	TxCK Input Period	Synchronous, no prescaler		Tcy + 10		_	ns	N = prescale value
			Synchro with pre	onous, scaler	Greater of: 20 ns or (TCY + 40)/N				(1, 8, 64, 256)
TB20	TCKEXT- MRL	Delay from Externa Edge to Timer Incre	al TxCK C ement	Clock	0.5 TCY	_	1.5 TCY	_	

#### TABLE 24-25: TIMER3 AND TIMER5 EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS				(unless Operat	s otherwise sta ing temperature	a <b>ted)</b> e -40°( -40°(	$C \le TA \le C \le TA \le C$	+85°C fo +125°C	or Industrial for Extended
Param No.	Symbol	Characte	eristic		Min	Тур	Max	Units	Conditions
TC10	TtxH	TxCK High Time	Synchronous		0.5 Tcy + 20	_		ns	Must also meet parameter TC15
TC11	TtxL	TxCK Low Time	Synchron	ous	0.5 TCY + 20	Ι		ns	Must also meet parameter TC15
TC15	TtxP	TxCK Input Period	Synchron no presca	ous, aler	Tcy + 10		_	ns	N = prescale value
			Synchronous, with prescaler		Greater of: 20 ns or (TCY + 40)/N				(1, 8, 64, 256)
TC20	TCKEXT- MRL	Delay from Externa Edge to Timer Incre	I TxCK Clo ement	ock	0.5 TCY	-	1.5 Тсү	—	





#### TABLE 24-26: QEI MODULE EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS			<b>Standa</b> (unles Opera	ard Operating s otherwise s ting temperatu	Conditi tated) re -40 -40	ons: 2.5 °C ≤ Ta ≤ °C ≤ Ta ≤	<b>/ to 5.5∖</b> +85°C f +125°C	/ or Industrial for Extended	
Param No.	Symbol	Characteristic <sup>(1)</sup>			Min	Тур	Max	Units	Conditions
TQ10	TtQH	TQCK High Time	Synchronous, with prescaler		Tcy + 20			ns	Must also meet parameter TQ15
TQ11	TtQL	TQCK Low Time	Synchro with pre	onous, escaler	Tcy + 20			ns	Must also meet parameter TQ15
TQ15	TtQP	TQCP Input Period	Synchronous, with prescaler		2 * Tcy + 40			ns	_
TQ20	TCKEXTMRL	Delay from External Edge to Timer Incre	I TxCK Clock		0.5 TCY		1.5 TCY	ns	

Note 1: These parameters are characterized but not tested in manufacturing.

#### FIGURE 24-10: INPUT CAPTURE (CAPx) TIMING CHARACTERISTICS



#### TABLE 24-27: INPUT CAPTURE TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended						
Param No.         Symbol         Characteristic <sup>(1)</sup>			ristic <sup>(1)</sup>	Min	Max	Units	Conditions		
IC10	TccL	ICx Input Low Time	No Prescaler	0.5 Tcy + 20		ns			
			With Prescaler	10		ns			
IC11	TccH	ICx Input High Time	No Prescaler	0.5 Tcy + 20		ns			
			With Prescaler	10		ns			
IC15	TccP	ICx Input Period		(2 TCY + 40)/N		ns	N = prescale value (1, 4, 16)		

Note 1: These parameters are characterized but not tested in manufacturing.

#### FIGURE 24-11: OUTPUT COMPARE MODULE (OCx) TIMING CHARACTERISTICS



#### TABLE 24-28: OUTPUT COMPARE MODULE TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended					
Param No.	Symbol	Characteristic <sup>(1)</sup>	Min	Тур <sup>(2)</sup>	Max	Units	Conditions	
OC10	TccF	OCx Output Fall Time	_	_		ns	See parameter D032	
OC11	TccR	OCx Output Rise Time	—	—	—	ns	See parameter D031	

Note 1: These parameters are characterized but not tested in manufacturing.

#### FIGURE 24-12: OC/PWM MODULE TIMING CHARACTERISTICS



#### TABLE 24-29: SIMPLE OC/PWM MODE TIMING REQUIREMENTS

AC CHARACTERISTICS				Standard Operating Conditions: 2.5V to 5.5V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended				
Param No.	Symbol	Characteristic <sup>(1)</sup>	Min	Typ <sup>(2)</sup>	Max	Units	Co	onditions
OC15	Tfd	Fault Input to PWM I/O Change	_	_	50	ns	_	OC15
OC20	TFLT	Fault Input Pulse Width	50			ns	_	OC20

Note 1: These parameters are characterized but not tested in manufacturing.

#### FIGURE 24-13: MOTOR CONTROL PWM MODULE FAULT TIMING CHARACTERISTICS



#### FIGURE 24-14: MOTOR CONTROL PWM MODULE TIMING CHARACTERISTICS



#### TABLE 24-30: MOTOR CONTROL PWM MODULE TIMING REQUIREMENTS

АС СНА	<b>Standa</b> (unless Operation	Standard Operating Conditions: 2.5V to 5.5V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended								
Param No.	Symbol	Characteristic <sup>(1)</sup>	Min Typ <sup>(2)</sup> Max Units Conditions							
MP10	TFPWM	PWM Output Fall Time	_	—	—	ns	See parameter D032			
MP11	TRPWM	PWM Output Rise Time	—	—	—	ns	See parameter D031			
MP20	TFD	Fault Input ↓ to PWM I/O Change	—	—	— 50 ns _					
MP30	Tfh	Minimum Pulse Width	50	—	_	ns	—			

**Note 1:** These parameters are characterized but not tested in manufacturing.





#### TABLE 24-31: QUADRATURE DECODER TIMING REQUIREMENTS

AC CHARACTERISTICS			$\begin{array}{l} \mbox{Standard Operating Conditions: } 2.5V \mbox{ to } 5.5V \\ \mbox{(unless otherwise stated)} \\ \mbox{Operating temperature} & -40^{\circ}\mbox{C} \leq \mbox{Ta} \leq +85^{\circ}\mbox{C for Industrial} \\ & -40^{\circ}\mbox{C} \leq \mbox{Ta} \leq +125^{\circ}\mbox{C for Extended} \end{array}$					
Param No.	Symbol	Characteristic <sup>(1)</sup>		Тур <sup>(2)</sup>	Мах	Units	Conditions	
TQ30	TQUL	Quadrature Input Low Time		6 Тсү	—	ns	—	
TQ31	ΤουΗ	Quadrature Input High Time		6 Tcy	—	ns	—	
TQ35	ΤαυΙΝ	Quadrature Input Period		12 TCY	—	ns	_	
TQ36	ΤουΡ	Quadrature Phase Period		3 TCY	—	ns	—	
TQ40	TQUFL	Filter Time to Recognize Low with Digital Filter	Ι,	3 * N * Tcy	—	ns	N = 1, 2, 4, 16, 32, 64, 128 and 256 <b>(Note 2)</b>	
TQ41	TQUFH	Filter Time to Recognize Hig with Digital Filter	h,	3 * N * Tcy		ns	N = 1, 2, 4, 16, 32, 64, 128 and 256 <b>(Note 2)</b>	

Note 1: These parameters are characterized but not tested in manufacturing.

2: N = Index Channel Digital Filter Clock Divide Select Bits. Refer to the "Quadrature Encoder Interface (QEI)" section in the "dsPIC30F Family Reference Manual" (DS70046).





#### TABLE 24-32: QEI INDEX PULSE TIMING REQUIREMENTS

AC CHARACTERISTICS			$\begin{array}{llllllllllllllllllllllllllllllllllll$					
Param No.	Symbol	Characteristic	; <sup>(1)</sup>	Min	Max	Units	Conditions	
TQ50	TqIL	Filter Time to Recognize with Digital Filter	Low,	3 * N * Tcy	_	ns	N = 1, 2, 4, 16, 32, 64, 128 and 256 <b>(Note 2)</b>	
TQ51	TqiH	Filter Time to Recognize with Digital Filter	High,	3 * N * Tcy	—	ns	N = 1, 2, 4, 16, 32, 64, 128 and 256 <b>(Note 2)</b>	
TQ55	Tqidxr	Index Pulse Recognized Counter Reset (Ungated	to Position Index)	3 TCY	_	ns	_	

**Note 1:** These parameters are characterized but not tested in manufacturing.

2: Alignment of Index Pulses to QEA and QEB is shown for Position Counter reset timing only. Shown for forward direction only (QEA leads QEB). Same timing applies for reverse direction (QEA lags QEB) but Index Pulse recognition occurs on falling edge.



#### FIGURE 24-17: SPI MODULE MASTER MODE (CKE = 0) TIMING CHARACTERISTICS

#### TABLE 24-33: SPI MASTER MODE (CKE = 0) TIMING REQUIREMENTS

AC CHARACTERISTICS			$\begin{tabular}{lllllllllllllllllllllllllllllllllll$						
Param No.	Symbol	Characteristic <sup>(1)</sup>	Min Typ <sup>(2)</sup> Max Units Con						
SP10	TscL	SCKx Output Low Time <sup>(3)</sup>	Tcy / 2		_	ns	_		
SP11	TscH	SCKx Output High Time <sup>(3)</sup>	Tcy / 2	—	_	ns	_		
SP20	TscF	SCKx Output Fall Time <sup>(4</sup>	—		—	ns	See parameter D032		
SP21	TscR	SCKx Output Rise Time <sup>(4)</sup>	—		—	ns	See parameter D031		
SP30	TdoF	SDOx Data Output Fall Time <sup>(4)</sup>	—		_	ns	See parameter D032		
SP31	TdoR	SDOx Data Output Rise Time <sup>(4)</sup>	—		—	ns	See parameter D031		
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—		30	ns	_		
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	20	_		ns			
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	20		_	ns	_		

Note 1: These parameters are characterized but not tested in manufacturing.

2: Data in "Typ" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

**3:** The minimum clock period for SCK is 100 ns. Therefore, the clock generated in Master mode must not violate this specification.



TABLE 24-34:	SPI MODULE MASTER M	<b>ODE (CKE =</b> 1)	) TIMING REQUIREMENTS
--------------	---------------------	----------------------	-----------------------

AC CHA	AC CHARACTERISTICS			$\begin{array}{l} \mbox{Standard Operating Conditions: 2.5V to 5.5V} \\ \mbox{(unless otherwise stated)} \\ \mbox{Operating temperature} & -40^\circ C \leq TA \leq +85^\circ C \mbox{ for Industrial} \\ & -40^\circ C \leq TA \leq +125^\circ C \mbox{ for Extended} \end{array}$						
Param No.	Symbol	Characteristic <sup>(1)</sup>	Min	Тур <sup>(2)</sup>	Мах	Units	Conditions			
SP10	TscL	SCKx output low time <sup>(3)</sup>	Tcy / 2	—		ns				
SP11	TscH	SCKx output high time <sup>(3)</sup>	Tcy / 2	_	_	ns	—			
SP20	TscF	SCKx output fall time <sup>(4)</sup>	—	—	_	ns	See parameter D032			
SP21	TscR	SCKx output rise time <sup>(4)</sup>	—	—	_	ns	See parameter D031			
SP30	TdoF	SDOx data output fall time <sup>(4)</sup>	—	_	_	ns	See parameter D032			
SP31	TdoR	SDOx data output rise time <sup>(4)</sup>	—	—	_	ns	See parameter D031			
SP35	TscH2doV, TscL2doV	SDOx data output valid after SCKx edge	—	—	30	ns	—			
SP36	TdoV2sc, TdoV2scL	SDOx data output setup to first SCKx edge	30	—		ns	—			
SP40	TdiV2scH, TdiV2scL	Setup time of SDIx data input to SCKx edge	20	_		ns				
SP41	TscH2diL, TscL2diL	Hold time of SDIx data input to SCKx edge	20	_		ns				

**Note 1:** These parameters are characterized but not tested in manufacturing.

2: Data in "Typ" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

**3:** The minimum clock period for SCK is 100 ns. Therefore, the clock generated in Master mode must not violate this specification.





#### TABLE 24-35: SPI MODULE SLAVE MODE (CKE = 0) TIMING REQUIREMENTS

AC CHAI	AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended					
Param No.	Symbol	Characteristic <sup>(1)</sup>	Min	Тур <sup>(2)</sup>	Мах	Units	Conditions		
SP70	TscL	SCKx Input Low Time	30	—	—	ns	—		
SP71	TscH	SCKx Input High Time	30	—	_	ns	—		
SP72	TscF	SCKx Input Fall Time <sup>(3)</sup>		10	25	ns	—		
SP73	TscR	SCKx Input Rise Time <sup>(3)</sup>		10	25	ns	—		
SP30	TdoF	SDOx Data Output Fall Time <sup>(3)</sup>	_	—	—	ns	See parameter D032		
SP31	TdoR	SDOx Data Output Rise Time <sup>(3)</sup>	_	—	—	ns	See parameter D031		
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	_	—	30	ns	—		
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	20	—		ns	_		
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	20	—		ns	_		
SP50	TssL2scH, TssL2scL	SSx↓ to SCKx↑ or SCKx↓ Input	120	-	—	ns	—		
SP51	TssH2doZ	SSx↑ to SDOx Output High-Impedance <sup>(3)</sup>	10	—	50	ns	—		
SP52	TscH2ssH TscL2ssH	SSx after SCK Edge	1.5 Tcy +40	—	—	ns	—		

Note 1: These parameters are characterized but not tested in manufacturing.

2: Data in "Typ" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.



#### FIGURE 24-20: SPI MODULE SLAVE MODE (CKE = 1) TIMING CHARACTERISTICS

АС СНА	RACTERIS	TICS	Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated)						
			Operating ter	for Industrial C for Extended					
Param No.	Symbol	Characteristic <sup>(1)</sup>	Min	Тур <sup>(2)</sup>	Мах	Units	Conditions		
SP70	TscL	SCKx Input Low Time	30			ns			
SP71	TscH	SCKx Input High Time	30	—	-	ns	—		
SP72	TscF	SCKx Input Fall Time <sup>(3)</sup>	—	10	25	ns	—		
SP73	TscR	SCKx Input Rise Time <sup>(3)</sup>	_	10	25	ns	—		
SP30	TdoF	SDOx Data Output Fall Time <sup>(3)</sup>	—			ns	See parameter D032		
SP31	TdoR	SDOx Data Output Rise Time <sup>(3)</sup>	_			ns	See parameter D031		
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	_	—	30	ns	—		
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	20	—	_	ns	—		
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	20	—	_	ns	—		
SP50	TssL2scH, TssL2scL	SSx↓ to SCKx↓ or SCKx↑ input	120	—		ns	—		
SP51	TssH2doZ	SS↑ to SDOx Output High-Impedance <sup>(4)</sup>	10	_	50	ns	—		
SP52	TscH2ssH TscL2ssH	SSx↑ after SCKx Edge	1.5 TCY + 40	—	_	ns	_		
SP60	TssL2doV	SDOx Data Output Valid after SSx Edge	_	_	50	ns			

#### TABLE 24-36: SPI MODULE SLAVE MODE (CKE = 1) TIMING REQUIREMENTS

**Note 1:** These parameters are characterized but not tested in manufacturing.

2: Data in "Typ" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

**3:** The minimum clock period for SCK is 100 ns. Therefore, the clock generated in Master mode must not violate this specification.

#### FIGURE 24-21: I<sup>2</sup>C<sup>™</sup> BUS START/STOP BITS TIMING CHARACTERISTICS (MASTER MODE)







### TABLE 24-37: I<sup>2</sup>C<sup>™</sup> BUS DATA TIMING REQUIREMENTS (MASTER MODE)

АС СНА	RACTER	ISTICS		Standard Operating Conditions: 2.5V to 5.5V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended					
Param No.	Symbol	Charac	teristic	Min <sup>(1)</sup>	Max	Units	Conditions		
IM10	TLO:SCL	Clock Low Time	100 kHz mode	Tcy/2(BRG + 1)	_	μs	—		
			400 kHz mode	Tcy/2(BRG + 1)		μs			
			1 MHz mode <sup>(2)</sup>	Tcy/2(BRG + 1)		μs			
IM11	THI:SCL	Clock High Time	100 kHz mode	Tcy/2(BRG + 1)		μs	—		
			400 kHz mode	Tcy/2(BRG + 1)		μs	—		
			1 MHz mode <sup>(2)</sup>	Tcy/2(BRG + 1)	_	μs	_		
IM20	TF:SCL	SDA and SCL	100 kHz mode		300 ns		CB is specified to be		
		Fall Time	400 kHz mode	20 + 0.1 Св	300	ns	from 10 to 400 pF		
			1 MHz mode <sup>(2)</sup>	—	100	ns			
IM21	TR:SCL	SDA and SCL	100 kHz mode	—	1000	ns	CB is specified to be		
		Rise Time	400 kHz mode	20 + 0.1 Св	300	ns	from 10 to 400 pF		
			1 MHz mode <sup>(2)</sup>		300	ns			
IM25	TSU:DAT	Data Input	100 kHz mode	250		ns	—		
		Setup Time	400 kHz mode	100	_	ns			
			1 MHz mode <sup>(2)</sup>	TBD	_	ns			
IM26	THD:DAT	Data Input	100 kHz mode	0		ns	—		
		Hold Time	400 kHz mode	0	0.9	μs			
			1 MHz mode <sup>(2)</sup>	TBD	_	ns			
IM30	TSU:STA	Start Condition	100 kHz mode	Tcy/2(BRG + 1)		μs	Only relevant for		
		Setup Time	400 kHz mode	Tcy/2(BRG + 1)		μs	repeated Start		
			1 MHz mode <sup>(2)</sup>	Tcy/2(BRG + 1)		μs	condition		
IM31	THD:STA	Start Condition	100 kHz mode	Tcy/2(BRG + 1)		μs	After this period the		
		Hold Time	400 kHz mode	Tcy/2(BRG + 1)		μs	first clock pulse is		
			1 MHz mode <sup>(2)</sup>	Tcy/2(BRG + 1)		μs	generated		
IM33	Tsu:sto	Stop Condition	100 kHz mode	Tcy/2(BRG + 1)	_	μs	—		
		Setup Time	400 kHz mode	Tcy/2(BRG + 1)	_	μs			
			1 MHz mode <sup>(2)</sup>	Tcy/2(BRG + 1)	_	μs			
IM34	THD:STO	Stop Condition	100 kHz mode	Tcy/2(BRG + 1)	_	ns	—		
		Hold Time	400 kHz mode	Tcy/2(BRG + 1)	—	ns			
			1 MHz mode <sup>(2)</sup>	Tcy/2(BRG + 1)	—	ns			
IM40	TAA:SCL	Output Valid	100 kHz mode	—	3500	ns	—		
		From Clock	400 kHz mode	—	1000	ns	—		
			1 MHz mode <sup>(2)</sup>	—	_	ns	—		
IM45	TBF:SDA	Bus Free Time	100 kHz mode	4.7		μs	Time the bus must be		
			400 kHz mode	1.3		μs	free before a new		
			1 MHz mode <sup>(2)</sup>	TBD	TBD — µs		transmission can start		
IM50	Св	Bus Capacitive L	oading		400	pF			

Note 1: BRG is the value of the I<sup>2</sup>C Baud Rate Generator. Refer to the "Inter-Integrated Circuit™ (I<sup>2</sup>C)" section in the "dsPIC30F Family Reference Manual" (DS70046).

**<sup>2:</sup>** Maximum pin capacitance = 10 pF for all  $I^2C^{TM}$  pins (for 1 MHz mode only).

#### FIGURE 24-23: I<sup>2</sup>C<sup>™</sup> BUS START/STOP BITS TIMING CHARACTERISTICS (SLAVE MODE)



#### FIGURE 24-24: I<sup>2</sup>C<sup>™</sup> BUS DATA TIMING CHARACTERISTICS (SLAVE MODE)



### TABLE 24-38: I<sup>2</sup>C<sup>™</sup> BUS DATA TIMING REQUIREMENTS (SLAVE MODE)

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended						
Param No.	Symbol	Charac	teristic	Min	Max	Units	Conditions		
IS10	TLO:SCL	Clock Low Time	100 kHz mode	4.7		μs	Device must operate at a minimum of 1.5 MHz		
			400 kHz mode	1.3	_	μs	Device must operate at a minimum of 10 MHz.		
			1 MHz mode <sup>(1)</sup>	0.5	_	μs	—		
IS11	THI:SCL	Clock High Time	100 kHz mode	4.0	_	μs	Device must operate at a minimum of 1.5 MHz		
			400 kHz mode	0.6		μs	Device must operate at a minimum of 10 MHz		
			1 MHz mode <sup>(1)</sup>	0.5		μs	—		
IS20	TF:SCL	SDA and SCL	100 kHz mode		300	ns	CB is specified to be from		
		Fall Time	400 kHz mode	20 + 0.1 Св	300	ns	10 to 400 pF		
			1 MHz mode <sup>(1)</sup>		100	ns			
IS21	TR:SCL	SDA and SCL Rise Time	100 kHz mode	—	1000	ns	CB is specified to be from		
			400 kHz mode	20 + 0.1 Св	300	ns	10 to 400 pF		
			1 MHz mode <sup>(1)</sup>	_	300	ns			

**Note 1:** Maximum pin capacitance = 10 pF for all  $I^2C^{TM}$  pins (for 1 MHz mode only).

### TABLE 24-38: I<sup>2</sup>C<sup>™</sup> BUS DATA TIMING REQUIREMENTS (SLAVE MODE) (CONTINUED)

AC CHARACTERISTICS		Standard Operating Conditions: 2.5V to 5.5V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended						
Param No.	Symbol	Charac	teristic	Min	Max	Units	Conditions	
IS25	TSU:DAT	Data Input	100 kHz mode	250	_	ns	—	
		Setup Time	400 kHz mode	100	—	ns		
			1 MHz mode <sup>(1)</sup>	100	_	ns		
IS26	THD:DAT	Data Input	100 kHz mode	0	—	ns	—	
	Hold Time	400 kHz mode	0	0.9	μs			
			1 MHz mode <sup>(1)</sup>	0	0.3	μs		
IS30	TSU:STA	Start Condition	100 kHz mode	4.7	_	μs	Only relevant for repeated	
		Setup Time	400 kHz mode	0.6	—	μs	Start condition	
			1 MHz mode <sup>(1)</sup>	0.25	—	μs		
IS31	THD:STA	Start Condition	100 kHz mode	4.0	—	μs	After this period the first	
		Hold lime	400 kHz mode	0.6	—	μs	clock pulse is generated	
			1 MHz mode <sup>(1)</sup>	0.25	—	μs		
IS33	TSU:STO	Stop Condition	100 kHz mode	4.7	—	μs	—	
		Setup Time	400 kHz mode	0.6		μs		
			1 MHz mode <sup>(1)</sup>	0.6		μs		
IS34	THD:STO	Stop Condition	100 kHz mode	4000		ns	_	
		Hold Time	400 kHz mode	600	—	ns		
			1 MHz mode <sup>(1)</sup>	250		ns		
IS40	TAA:SCL	Output Valid	100 kHz mode	0	3500	ns	—	
		From Clock	400 kHz mode	0	1000	ns		
			1 MHz mode <sup>(1)</sup>	0	350	ns		
IS45	TBF:SDA	Bus Free Time	100 kHz mode	4.7		μs	Time the bus must be free	
			400 kHz mode	1.3		μs	before a new transmission	
			1 MHz mode <sup>(1)</sup>	0.5		μs	Call Sidil	
IS50	Св	Bus Capacitive Loading		—	400	pF	_	

**Note 1:** Maximum pin capacitance = 10 pF for all  $I^2C^{TM}$  pins (for 1 MHz mode only).

#### FIGURE 24-25: CAN MODULE I/O TIMING CHARACTERISTICS



#### TABLE 24-39: CAN MODULE I/O TIMING REQUIREMENTS

AC CHARA	CTERISTICS	Standard Op (unless othe Operating ter	erating C rwise sta mperature	Conditions Ited) e -40°C : -40°C :	<b>s: 2.5V to</b> ≤ TA ≤ +85 ≤ TA ≤ +12	5.5V °C for Ind 5°C for E	ustrial ktended	
Param No.	Symbol	Characteri	Min	Тур <sup>(2)</sup>	Max	Units	Conditions	
CA10	TioF	Port Output Fall Ti	me	—	10	25	ns	_
CA11	TioR	Port Output Rise Time		—	10	25	ns	
CA20	Tcwf	Pulse Width to Trig CAN Wakeup Filte	gger r	500			ns	

**Note 1:** These parameters are characterized but not tested in manufacturing.

AC CHARACTERISTICS			Standard Operating Conditions: 2.7V to 5.5V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended						
Param No.	Symbol	Characteristic	Min.	Тур	Max.	Units	Conditions		
			Device Su	ipply					
AD01	AVdd	Module VDD Supply	Greater of VDD - 0.3 or 2.7		Lesser of VDD + 0.3 or 5.5	V			
AD02	AVss	Module Vss Supply	Vss - 0.3		Vss + 0.3	V	_		
			Reference	Inputs					
AD05	Vrefh	Reference Voltage High	AVss+2.7		AVdd	V	—		
AD06	Vrefl	Reference Voltage Low	AVss		AVDD - 2.7	V	—		
AD07	Vref	Absolute Reference Voltage	AVss - 0.3		AVDD + 0.3	V	—		
AD08	IREF	Current Drain	—	200 .001	300 3	μΑ μΑ	A/D operating A/D off		
			Analog Ir	nput					
AD10	VINH-VINL	Full-Scale Input Span	Vrefl		Vrefh	V	—		
AD12	_	Leakage Current	_	±0.001	±0.244	μA	$      VINL = AVSS = VREFL = 0V, \\       AVDD = VREFH = 5V \\       Source Impedance = 5 k \Omega $		
AD13	_	Leakage Current	_	±0.001	±0.244	μA			
AD17	Rin	Recommended Impedance Of Analog Voltage Source	—		—	Ω	See Table 20-1		
			DC Accu	racy					
AD20	Nr	Resolution	1	0 data b	its	bits	—		
AD21	INL	Integral Nonlinearity <sup>(3)</sup>	_	±1	±1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V		
AD21A	INL	Integral Nonlinearity <sup>(3)</sup>	_	±1	±1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V		
AD22	DNL	Differential Nonlinearity <sup>(3)</sup>	—	±1	±1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V		
AD22A	DNL	Differential Nonlinearity <sup>(3)</sup>	—	±1	±1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V		
AD23	Gerr	Gain Error <sup>(3)</sup>	<u>+</u> 1	±5	±6	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V		
AD23A	Gerr	Gain Error <sup>(3)</sup>	<u>+</u> 1	±5	±6	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V		

#### TABLE 24-40: 10-BIT HIGH-SPEED ADC MODULE SPECIFICATIONS

Note 1: These parameters are characterized but not tested in manufacturing..

2: The A/D conversion result never decreases with an increase in the input voltage, and has no missing codes.

3: Measurements taken with external VREF+ and VREF- used as the ADC voltage reference.

### TABLE 24-40: 10-BIT HIGH-SPEED ADC MODULE SPECIFICATIONS (CONTINUED)

AC CHARACTERISTICS			$\begin{array}{l} \mbox{Standard Operating Conditions: 2.7V to 5.5V} \\ \mbox{(unless otherwise stated)} \\ \mbox{Operating temperature} & -40^{\circ}C \leq TA \leq +85^{\circ}C \mbox{ for Industrial} \\ & -40^{\circ}C \leq TA \leq +125^{\circ}C \mbox{ for Extended} \end{array}$						
Param No.	Symbol	Characteristic	Min.	Тур	Max.	Units	Conditions		
	DC Accuracy (Continued)								
AD24	EOFF	Offset Error	±1	±2	±3	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V		
AD24A	EOFF	Offset Error	±1	±2	±3	LSb	Vinl = AVss = Vrefl = 0V, AVdd = Vrefh = 3V		
AD25	_	Monotonicity <sup>(2)</sup>	_	_		-	Guaranteed		
		Dy	namic Perf	ormance	)				
AD30	THD	Total Harmonic Distortion	_	-64	-67	dB	_		
AD31	SINAD	Signal to Noise and Distortion	—	57	58	dB			
AD32	SFDR	Spurious Free Dynamic Range	—	67	71	dB	_		
AD33	FNYQ	Input Signal Bandwidth	_	—	500	kHz	—		
AD34	ENOB	Effective Number of Bits	9.29	9.41	_	bits			

Note 1: These parameters are characterized but not tested in manufacturing..

2: The A/D conversion result never decreases with an increase in the input voltage, and has no missing codes.

**3:** Measurements taken with external VREF+ and VREF- used as the ADC voltage reference.



FIGURE 24-26: 10-BIT HIGH-SPEED A/D CONVERSION TIMING CHARACTERISTICS (CHPS = 01, SIMSAM = 0, ASAM = 0, SSRC = 000)



#### FIGURE 24-27: 10-BIT HIGH-SPEED ADC CONVERSION TIMING CHARACTERISTICS (CHPS = 01, SIMSAM = 0, ASAM = 1, SSRC = 111, SAMC = 00001)

#### TABLE 24-41: 10-BIT HIGH-SPEED ADC CONVERSION TIMING REQUIREMENTS

AC CH	ARACTE	RISTICS	$\begin{array}{l} \mbox{Standard Operating Conditions: 2.7V to 5.5V} \\ \mbox{(unless otherwise stated)} \\ \mbox{Operating temperature} & -40^{\circ}C \leq TA \leq +85^{\circ}C \mbox{ for Industrial} \\ & -40^{\circ}C \leq TA \leq +125^{\circ}C \mbox{ for Extended} \end{array}$						
Param No.	Symbol	Characteristic	Min.	Typ <sup>(1)</sup>	Max.	Units	Conditions		
		Cloc	k Parame	ters					
AD50	TAD	A/D Clock Period	—	83.3	_	ns	See Table 20-2 <sup>(2)</sup>		
AD51	tRC	A/D Internal RC Oscillator Period	700	900	1100	ns	—		
	Conversion Rate								
AD55	tCONV	Conversion Time	—	12 Tad			—		
AD56	FCNV	Throughput Rate	—	1.0	-	Msps	See Table 20-2 <sup>(2)</sup>		
AD57	TSAMP	Sample Time	—	1 Tad	—	—	See Table 20-2 <sup>(2)</sup>		
		Timin	g Parame	eters					
AD60	tPCS	Conversion Start from Sample Trigger	—	1.0 Tad		ns	_		
AD61	tPSS	Sample Start from Setting Sample (SAMP) Bit	0.5 Tad	—	1.5 Tad	ns	_		
AD62	tCSS	Conversion Completion to Sample Start (ASAM = 1)	—	0.5 Tad	_	ns	—		
AD63	tdpu	Time to Stabilize Analog Stage from A/D Off to A/D On	—	20	—	μs	_		

**Note 1:** These parameters are characterized but not tested in manufacturing.

2: Characterized by design but not tested.

NOTES:

### 25.0 PACKAGING INFORMATION

### 25.1 Package Marking Information

80-Lead TQFP (14x14x1mm)



Example



Legend	: XXX Y YY WW NNN @3 *	Customer-specific information Year code (last digit of calendar year) Year code (last 2 digits of calendar year) Week code (week of January 1 is week '01') Alphanumeric traceability code Pb-free JEDEC designator for Matte Tin (Sn) This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.
Note:	In the even be carried characters	nt the full Microchip part number cannot be marked on one line, it will d over to the next line, thus limiting the number of available s for customer-specific information.

#### 80-Lead Plastic Thin Quad Flatpack (PF) 14x14x1 mm Body, 1.0/0.10 mm Lead Form (TQFP)

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units	INCHES MILLIMETERS*					
Dimension	MIN	NOM	MAX	MIN	NOM	MAX	
Number of Pins	n		80			80	
Pitch	р		.026			0.65	
Pins per Side	n1		20			20	
Overall Height	А			.047			1.20
Molded Package Thickness	A2	.037	.039	.041	0.95	1.00	1.05
Standoff	A1	.002		.006	0.05		0.15
Foot Length	L	.018	.024	.030	0.45	0.60	0.75
Footprint	F	.039 REF.					
Foot Angle	¢	0°	3.5°	7°	0°	3.5°	7°
Overall Width	Е		.630 BSC			16.00 BSC	
Overall Length	D		.630 BSC		16.00 BSC		
Molded Package Width	E1		.551 BSC		14.00 BSC		
Molded Package Length	D1	.551 BSC		14.00 BSC			
Lead Thickness	С	.004		.008	0.09		0.20
Lead Width	В	.011	.013	.015	0.27	0.32	0.37
Mold Draft Angle Top	α	11°	12°	13°	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°	11°	12°	13°

\* Controlling Parameter

Notes:

Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side. BSC: Basic Dimension. Theoretically exact value shown without tolerances.

See ASME Y14.5M

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

See ASME Y14.5M

JEDEC Equivalent: MS-026

Drawing No. C04-116

Revised 7-20-06

### APPENDIX A: REVISION HISTORY

#### Revision E (November 2006)

Previous versions of this data sheet contained Advance or Preliminary Information. They were distributed with incomplete characterization data.

Revision E of this document reflects the following updates:

- Supported I<sup>2</sup>C Slave Addresses (see Table 17-1)
- ADC Conversion Clock selection to allow 1 Msps operation (see Section 20.0 "10-bit High-Speed Analog-to-Digital Converter (ADC) Module")
- Base Instruction CP1 removed from instruction set (see Table 23-2)
- Revised electrical characteristics:
  - Operating Current (IDD) (see Table 24-5)
  - Idle Current (IIDLE) (see Table 24-6)
  - Power-down Current (IPD) (see Table 24-7)
  - I/O Pin Input specifications (see Table 24-8)
  - Brown-out Reset (BOR) (see Table 24-11)
  - Watchdog Timer (see Table 24-21)

NOTES:

### INDEX

L	7
•	•

A/D	
1 Msps Configuration Guideline	
600 ksps Configuration Guideline	133
750 ksps Configuration Guideline	133
Conversion Rate Parameters	131
Conversion Speeds	131
Selecting the Conversion Clock	130
Voltage Reference Schematic	
AC Characteristics	
Load Conditions	176
AC Temperature and Voltage Specifications	
Address Generator Units	
Alternate 16-bit Timer/Counter	
Alternate Vector Table	41
Assembler	
MPASM Assembler	154
Automatic Clock Stretch	102
During 10-bit Addressing (STREN = 1)	
During 7-bit Addressing (STREN = 1)	102
Receive Mode	
Transmit Mode	102

### В

Bandgap Start-up Time
Requirements
Timing Characteristics
Barrel Shifter
Bit-Reversed Addressing
Example
Implementation
Modifier Values (table)
Sequence Table (16-Entry)
Block Diagrams
10-bit High Speed A/D Functional
16-bit Timer1 Module58
16-bit Timer468
16-bit Timer568
32-bit Timer4/567
CAN Buffers and Protocol Engine116
Dedicated Port Structure53
DSP Engine15
dsPIC30F60106
External Power-on Reset Circuit147
I <sup>2</sup> C100
Input Capture Mode71
Oscillator System 141
Output Compare Mode75
PWM Module86
Quadrature Encoder Interface79
Reset System145
Shared Port Structure54
SPI
SPI Master/Slave Connection96
UART Receiver 108
UART Transmitter 107
BOR Characteristics175
BOR. See Brown-out Reset
Brown-out Reset
Characteristics 174
Timing Requirements182
Brown-out Reset (BOR)

### С

C Compilers	
MPLAB C18 15	4
MPLAB C30 15	4
CAN Module 11	5
CAN1 Register Map 12	2
CAN2 Register Map 12	4
I/O Timing Characteristics 20	0
Overview11	5
Center Aligned PWM 8	9
CLKOUT and I/O Timing	
Characteristics 18	0
Requirements 18	0
Code Examples	
Data EEPROM Block Erase 5	0
Data EEPROM Block Write 5	2
Data EEPROM Read 4	9
Data EEPROM Word Erase 5	0
Data EEPROM Word Write 5	1
Erasing a Row of Program Memory 4	5
Initiating a Programming Sequence4	6
Loading Write Latches 4	6
Code Protection	9
Complementary PWM Operation	0
Configuring Analog Port Pins5	4
Control Registers 4	4
NVMADR 4	4
NVMADRU 4	4
NVMCON 4	4
NVMKEY 4	4
Core Overview 1	1
Core Register Map 2	7
CPU Architecture Overview 1	1
Customer Change Notification Service	6
Customer Notification Service	6
Customer Support	6

### D

Data Access from Program Memory Using	
Program Space Visibility	. 22
Data Accumulators and Adder/Subtractor	. 16
Data Space Write Saturation	. 18
Overflow and Saturation	. 16
Round Logic	. 17
Write Back	. 17
Data Address Space	. 23
Alignment	. 26
Alignment (Figure)	. 26
Effect of Invalid Memory Accesses	. 26
MCU and DSP (MAC Class) Instructions Example	. 25
Memory Map 23	, 24
Near Data Space	. 27
Software Stack	. 27
Spaces	. 26
	. 26

Data EEPROM Memory	49
Erasing	50
Erasing, Block	
Erasing, Word	
Protection Against Spurious Write	52
Reading	49
Write Verify	
Writing	51
Writing. Block	
Writing. Word	
DC Characteristics	
BOR	
Brown-out Reset	
I/O Pin Input Specifications	
I/O Pin Output Specifications	
Idle Current (IDLE)	170
Low-Voltage Detect	173
L VDI	174
Operating Current (IDD)	169
Power-Down Current (IPD)	171
Program and FEPROM	175
Temperature and Voltage Specifications	167
Dead-Time Generators	90
Assignment	90
Ranges	90
Selection Bits	
Development Support	
Device Configuration	
Register Man	152
Device Configuration Registers	150
	150
FCS	
FOSC	
FUSC	150
Divide Support	
Multiplior	14
Multiplier	
Dual Output Compare Match Mode	
Single Dulee Mede	
Single Pulse Mode	
E	
Edge Aligned DWM	00

Edge Aligned PWM	38
Electrical Characteristics	67
AC	76
DC16	67
Equations	
A/D Conversion Clock	30
Baud Rate111, 12	21
PWM Period	38
PWM Resolution	38
Serial Clock Rate10	)4
Errata	. 4
Exception Processing	
Interrupt Priority	38
Exception Sequence	
Trap Sources	39
External Clock Timing Characteristics	
Type A, B and C Timer18	33
External Clock Timing Requirements17	77
Type A Timer	33
Type B Timer	34
Type C Timer	34
External Interrupt Requests	41

#### F

Fast Context Saving
Flash Program Memory
In-Circuit Serial Programming (ICSP)
Run Time Self-Programming (RTSP)
Table Instruction Operation Summary
I
I/O Pin Specifications
Input 173
Output 173
I/O Ports
Parallel I/O (PIO)
I <sup>2</sup> C 10-bit Slave Mode Operation 101
Reception
I <sup>2</sup> C 7-bit Slave Mode Operation
Reception
I ransmission
Raud Pate Concreter 400
Dauu Rate Generator
Multi-Master Communication Rue Collision
and Rus Arbitration
Recention 103
Transmission 103
1 <sup>2</sup> C Module
Addresses 101
Bus Data Timing Characteristics
Master Mode
Slave Mode198
Bus Data Timing Requirements
Master Mode 197
Slave Mode 198
Bus Start/Stop Bits Timing Characteristics
Master Mode 196
Slave Mode 198
General Call Address Support 103
Interrupts 102
IPMI Support 103
Master Operation 103
Master Support 103
Operating Function Description
Operation During CPU Sleep and Idle Modes 104
Pin Configuration
Programmer's Model
Register Map
Slope Control 103
Software Controlled Clock Stratching
(STRENI – 1) 102
Various Modes 99
Idle Current (IDLE)
In-Circuit Serial Programming (ICSP)
Independent PWM Output
Initialization Condition for RCON Register Case 1
Initialization Condition for RCON Register Case 2
Input Capture (CAPX) Timing Characteristics
Input Capture Interrupts72
Register Map 73
Input Capture Module71
In CPU Sleep Mode
Simple Capture Event Mode71
Input Capture Liming Requirements

Input Change Notification Module	56
Register Map (bits 15-8)	56
Register Map (bits 7-0)	56
Input Characteristics	
QEA/QEB	189
Instruction Addressing Modes	31
File Register Instructions	31
Fundamental Modes Supported	31
MAC Instructions	32
MCU Instructions	31
Move and Accumulator Instructions	32
Other Instructions	32
Instruction Set Summary	157
Internet Address	216
Interrupt Controller	
Register Map	42
Interrupt Priority	
Traps	39
Interrupt Sequence	41
Interrupt Stack Frame	41
Interrupts	37

### L

6
3
4
4

#### Μ

Memory Organization	
Microchip Internet Web Site	
Modulo Addressing	
Applicability	
Operation Example	33
Start and End Address	33
W Address Register Selection	
Motor Control PWM Module	
Fault Timing Characteristics	
Timing Characteristics	
Timing Requirements	188
MPLAB ASM30 Assembler, Linker, Librarian	154
MPLAB ICD 2 In-Circuit Debugger	155
MPLAB ICE 2000 High-Performance Universal	
In-Circuit Emulator	155
MPLAB Integrated Development Environment	
Software	153
MPLAB PM3 Device Programmer	155
MPLAB REAL ICE In-Circuit Emulator System	155
MPLINK Object Linker/MPLIB Object Librarian	154

### 0

OC/PWM Module Timing Characteristics	
Operating Current (IDD)	169
Operating Frequency vs Voltage	
dsPIC30FXXXX-20 (Extended)	167
Oscillator Configurations	142
Fail-Safe Clock Monitor	143
Fast RC (FRC)	143
Initial Clock Source Selection	142
Low Power RC (LPRC)	143
LP Oscillator Control	142
Phase Locked Loop (PLL)	
Start-up Timer (OST)	
Oscillator Operating Modes Table	140
Oscillator Selection	

Oscillator Start-up Timer	
Timing Characteristics	181
Timing Requirements	182
Output Compare Interrupts	77
Output Compare Mode	
Register Map	78
Output Compare Module	75
Timing Characteristics	186
Timing Requirements	186
Output Compare Operation During CPU Idle Mode	77
Output Compare Sleep Mode Operation	77
Ρ	
Packaging Information	207
Marking	207

Packaging Information	207
Marking	207
PICSTART Plus Development Programmer	156
PLL Clock Timing Specifications	178
POR. See Power-on Reset	
Port Write/Read Example	54
Position Measurement Mode	80
Power Saving Modes	149
Idle	150
Sleep	149
Power Saving Modes (Sleep and Idle)	139
Power-Down Current (IPD)	171
Power-on Reset (POR)	139
Oscillator Start-up Timer (OST)	130
Bower up Timer (DM/PT)	120
Power-up Timer (FWKT)	159
Timin a Characteristica	404
Timing Characteristics	181
Liming Requirements	182
Program Address Space	
Construction	
Data Access From Program Memory Using	
Table Instructions	21
Data Access from, Address Generation	20
Memory Map	19
Table Instructions	
TBLRDH	21
TBLRDL	21
TBLWTH	21
TBLWTL	
Program and EEPROM Characteristics	175
Program Counter	12
Program Data Table Access	22
Program Space Visibility	
Window into Program Space Operation	
Programmable	139
Programmable Digital Noise Filters	
Programmer's Model	
Diagram	
Programming Operations	
Algorithm for Program Flash	
Erasing a Pow of Program Momony	
Initiating the Brogramming Sequence	
Initiating the Programming Sequence	
Loading write Latches	
Protection Against Accidental Writes to OSCCON	144
PVVM Duty Cycle Comparison Units	
Duty Cycle Register Buffers	
PWM Fault Pins	
Enable Bits	92
Fault States	92
Modes	92
Cycle-by-Cycle	92
Latched	92
Priority	92

PWM Operation During CPU Idle Mode9	3
PWM Operation During CPU Sleep Mode9	3
PWM Output and Polarity Control9	2
Output Pin Control9	2
PWM Output Override9	1
Complementary Output Mode9	1
Synchronization9	1
PWM Period8	8
PWM Special Event Trigger9	3
Postscaler9	3
PWM Time Base8	7
Continuous Up/Down Counting Modes	7
Double Update Mode8	8
Free Running Mode8	7
Postscaler8	8
Prescaler8	8
Single Shot Mode8	7
PWM Update Lockout9	3

### Q

QEA/QEB Input Characteristics	189
QEI Module	
External Clock Timing Requirements	185
Index Pulse Timing Characteristics	190
Index Pulse Timing Requirements	190
Operation During CPU Idle Mode	81
Operation During CPU Sleep Mode	81
Register Map	83
Timer Operation During CPU Idle Mode	82
Timer Operation During CPU Sleep Mode	81
Quadrature Decoder Timing Requirements	189
Quadrature Encoder Interface (QEI) Module	79
Quadrature Encoder Interface Interrupts	82
Quadrature Encoder Interface Logic	80

### R

Reader Response	
Reset	. 139, 145
Reset Sequence	
Reset Sources	
Reset Timing Characteristics	
Reset Timing Requirements	
Resets	
BOR, Programmable	
POR	145
POR with Long Crystal Start-up Time	
POR, Operating without FSCM and PWRT	147

#### S

Simple Capture Event Mode	
Capture Buffer Operation	72
Capture Prescaler	71
Hall Sensor Mode	72
Input Capture in CPU Idle Mode	72
Timer2 and Timer3 Selection Mode	72
Simple OC/PWM Mode Timing Requirements	187
Simple Output Compare Match Mode	76
Simple PWM Mode	76
Input Pin Fault Protection	76
Period	77
Single Pulse PWM Operation	91
Software Simulator (MPLAB SIM)	154
Software Stack Pointer, Frame Pointer	12
CALL Stack Frame	27

SPI Mode	
Slave Select Synchronization	
SPI1 Register Map	
SPI2 Register Map	
SPI Module	
Framed SPI Support	
Operating Function Description	
SDOx Disable	
Timing Characteristics	
Master Mode (CKE = 0)	191
Master Mode (CKE = 1)	192
Slave Mode (CKE = 1)	193, 194
Timing Requirements	
Master Mode (CKE = 0)	
Master Mode (CKE = 1)	192
Slave Mode (CKE = 0)	193
Slave Mode (CKE = 1)	195
Word and Byte Communication	
SPI Operation During CPU Idle Mode	
SPI Operation During CPU Sleep Mode	
Status Register	
System Integration	139
Overview	139
Register Map	152

### т

Temperature and Voltage Specifications	
AC1	76
DC	67
Timer1 Module	57
16-bit Asynchronous Counter Mode	57
16-bit Synchronous Counter Mode	57
16-bit Timer Mode	57
Gate Operation	58
Interrupt	59
Operation During Sleep Mode	58
Prescaler	58
Real-Time Clock	59
RTC Interrupts	59
RTC Oscillator Operation	59
Register Map	60
Timer2 and Timer3 Selection Mode	76
Timer2/3 Module	61
32-bit Synchronous Counter Mode	61
32-bit Timer Mode	61
ADC Event Trigger	64
Gate Operation	64
Interrupt	64
Operation During Sleep Mode	64
Register Map	65
Timer Prescaler	64
Timer4/5 Module	67
Register Map	69
TimerQ (QEI Module) External Clock Timing Characteristics 185	s.
# dsPIC30F6010

Timing Characteristics A/D Conversion
10-Bit High-speed (CHPS = 01,
SIMSAM = 0, ASAM = 0, SSRC = 000)203
10-Bit High-speed (CHPS = 01, SIMSAM = 0,
ASAM = 1, SSRC = 111,
SAMC = 00001)
Bandgap Start-up Time
CAN Module I/O
CLKOUT and I/O
External Clock
I-C Bus Data
Master Mode
Slave Mode
T C Bus Statt/Stop Bits Master Mode 106
Master Mode 108
Slave Mode
Motor Coptrol DW/M Modulo
Motor Control PWM Module Falult
Oc/F WM Module
Output Compare Module
Power up Timer 191
OEL Module Index Pulse 100
Reset 181
SPI Module
Master Mode (CKE – 0) 191
Master Mode ( $CKE = 0$ )
Slave Mode (CKE = $1$ ) 193
Slave Mode (CKE = 0)
TimerO (OEI Module) External Clock 185
Type A B and C Timer External Clock 183
Watchdog Timer 181
Timing Diagrams
Center Aligned PW/M 89
Dead-Time 91
Edge Aligned PWM 89
PWM Output 77
Time-out Sequence on Power-up (MCLR Not
Tied to VDD). Case 1
Time-out Sequence on Power-up (MCLR Not
Tied to VDD), Case 2
Time-out Sequence on Power-up (MCLR
Tied to VDD)
Timing Diagrams.See Timing Characteristics
Timing Requirements
A/D Conversion
10-Bit High-speed
Bandgap Start-up Time182
Brown-out Reset
CLKOUT and I/O 180
External Clock177
I <sup>2</sup> C Bus Data (Master Mode)197
I <sup>2</sup> C Bus Data (Slave Mode)198
Input Capture186
Motor Control PWM Module188
Oscillator Start-up Timer 182
Output Compare Module186
Power-up Timer
QEI Module
External Clock185
Index Pulse
Quadrature Decoder
Keset

Simple OC/PWM Mode 18	37
SPI Module	
Master Mode (CKE = 0) 19	<b>3</b> 1
Master Mode (CKE = 1) 19	<del>)</del> 2
Slave Mode (CKE = 0) 19	ЭЗ
Slave Mode (CKE = 1)	<del>)</del> 5
Type A Timer External Clock18	33
Type B Timer External Clock	34
Type C Timer External Clock	34
Watchdog Timer 18	32
Timing Specifications	
PLL Clock 17	78
Trap Vectors	10
0	
UART	
Address Detect Mode 11	1
Auto Baud Support 11	11
Baud Rate Generator 11	1
Enabling and Setting Up UART 10	)9
Disabling10	)9
Enabling10	)9
Setting Up Data, Parity and Stop	
Bit Selections10	)9
Loopback Mode 11	11
Module Overview 10	)7
Operation During CPU Sleep and Idle Modes 11	12
Receiving Data 11	10
In 8-bit or 9-bit Data Mode11	10
Interrupt 11	0
Receive Buffer (UxRCB)11	10
Reception Error Handling 11	10
Framing Error (FERR) 11	11
Idle Status 11	11
Parity Error (PERR) 11	11
Receive Break 11	11
Receive Buffer Overrun Error (OERR Bit) 11	10
Transmitting Data 10	)9
In 8-bit Data Mode 10	)9
In 9-bit Data Mode10	)9
Interrupt 11	10
Transmit Buffer (UxTXB) 10	)9
UART1 Register Map 11	13
UART2 Register Map 11	13
Unit ID Locations	39
Universal Asynchronous Receiver Transmitter	
Module (UART) 10	)7
14/	

#### W

Wake-up from Sleep	139
Wake-up from Sleep and Idle	41
Watchdog Timer	
Timing Characteristics	
Timing Requirements	
Watchdog Timer (WDT)	139, 149
Enabling and Disabling	149
Operation	149
WWW Address	
WWW, On-Line Support	4

## dsPIC30F6010

NOTES:

## THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- Product Support Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- General Technical Support Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- Business of Microchip Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

# CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com, click on Customer Change Notification and follow the registration instructions.

### **CUSTOMER SUPPORT**

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://support.microchip.com

## dsPIC30F6010

### **READER RESPONSE**

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

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

To:	Technical Publications Manager	Total Pages Sent			
RE:	Reader Response				
From	: Name				
	Company				
	Address				
	City / State / ZIP / Country				
	Telephone: ()	FAX: ()			
Appli	cation (optional):				
Woul	d you like a reply?YN				
Devid	ce: dsPIC30F6010	Literature Number: DS70119E			
Ques	stions:				
1. V	1. What are the best features of this document?				
_					
_					
2. ⊦	2. How does this document meet your hardware and software development needs?				
-					
-					
3. Do you find the organization of this document easy to follow? If not, why?					
_					
	What additions to the document do y	you think would enhance the structure and subject?			
ч. <b>v</b>	what additions to the about tent do y				
-					
5. V	5. What deletions from the document could be made without affecting the overall usefulness?				
_					
_					
6. Is	6. Is there any incorrect or misleading information (what and where)?				
_					
_					
7. ⊦	low would you improve this docume	ent?			
_					
-					

## **PRODUCT IDENTIFICATION SYSTEM**

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.





## WORLDWIDE SALES AND SERVICE

#### AMERICAS

Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://support.microchip.com Web Address: www.microchip.com

Atlanta Alpharetta, GA Tel: 770-640-0034 Fax: 770-640-0307

Boston Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088

**Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075

**Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924

Detroit Farmington Hills, MI Tel: 248-538-2250 Fax: 248-538-2260

Kokomo Kokomo, IN Tel: 765-864-8360 Fax: 765-864-8387

Los Angeles Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608

Santa Clara Santa Clara, CA Tel: 408-961-6444 Fax: 408-961-6445

Toronto Mississauga, Ontario, Canada Tel: 905-673-0699 Fax: 905-673-6509

#### ASIA/PACIFIC

Asia Pacific Office Suites 3707-14, 37th Floor Tower 6, The Gateway Habour City, Kowloon Hong Kong Tel: 852-2401-1200 Fax: 852-2401-3431

Australia - Sydney Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

**China - Beijing** Tel: 86-10-8528-2100 Fax: 86-10-8528-2104

**China - Chengdu** Tel: 86-28-8665-5511 Fax: 86-28-8665-7889

**China - Fuzhou** Tel: 86-591-8750-3506 Fax: 86-591-8750-3521

**China - Hong Kong SAR** Tel: 852-2401-1200 Fax: 852-2401-3431

**China - Qingdao** Tel: 86-532-8502-7355 Fax: 86-532-8502-7205

**China - Shanghai** Tel: 86-21-5407-5533 Fax: 86-21-5407-5066

China - Shenyang Tel: 86-24-2334-2829 Fax: 86-24-2334-2393

**China - Shenzhen** Tel: 86-755-8203-2660 Fax: 86-755-8203-1760

**China - Shunde** Tel: 86-757-2839-5507 Fax: 86-757-2839-5571

**China - Wuhan** Tel: 86-27-5980-5300 Fax: 86-27-5980-5118

**China - Xian** Tel: 86-29-8833-7250 Fax: 86-29-8833-7256

### ASIA/PACIFIC

India - Bangalore Tel: 91-80-4182-8400 Fax: 91-80-4182-8422

India - New Delhi Tel: 91-11-4160-8631 Fax: 91-11-4160-8632

India - Pune Tel: 91-20-2566-1512 Fax: 91-20-2566-1513

**Japan - Yokohama** Tel: 81-45-471- 6166 Fax: 81-45-471-6122

**Korea - Gumi** Tel: 82-54-473-4301 Fax: 82-54-473-4302

Korea - Seoul Tel: 82-2-554-7200 Fax: 82-2-558-5932 or 82-2-558-5934

**Malaysia - Penang** Tel: 60-4-646-8870 Fax: 60-4-646-5086

Philippines - Manila Tel: 63-2-634-9065

Fax: 63-2-634-9069 Singapore Tel: 65-6334-8870 Fax: 65-6334-8850

**Taiwan - Hsin Chu** Tel: 886-3-572-9526 Fax: 886-3-572-6459

**Taiwan - Kaohsiung** Tel: 886-7-536-4818 Fax: 886-7-536-4803

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

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

#### EUROPE

Austria - Wels Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 Denmark - Copenhagen Tel: 45-4450-2828 Fax: 45-4485-2829

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

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

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

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

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

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

10/19/06

## **X-ON Electronics**

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Digital Signal Processors & Controllers - DSP, DSC category:

Click to view products by Microchip manufacturer:

Other Similar products are found below :

PIC33FJ32GP104-I/PT ADSP-2189NBCAZ-320 ADSP-BF703BBCZ-3 ADSP-2185MKCAZ-300 ADSP-BF701KBCZ-1 646890G ADSP-BF701KBCZ-2 ADSP-BF703BBCZ-4 0W888-002-XTP ADBF704WCCPZ311 AD21573WCBCZ402 ADSC572WCBCZ302 ADSC572WCBCZ402 ADSC572WCBCZ4202 ADSC572WCBCZ4200 ADSC572WCBCZ400 ADSC573WCBCZ300 ADSC573WCBCZ500 ADSC571WCSWZ300 ADSC571WCSWZ500 ADSC571WCSWZ400 ADSP-21567KBCZ6 ADSP-21566KBCZ4 ADSC570WCSWZ502 AD21488WBCPZ302 AD21488WBCPZ4202 ADSC570WCSWZ402 ADSC570WCSWZ4202 AD21488WBCPZ402 AD21488WBCPZ202 DSPIC33EP64MC202-E/SP ADSP-21583KBCZ-4A ADSP-BF701BBCZ-2 ADSP-BF705BBCZ-4 ADSP-SC582BBCZ-4A ADSP-SC584BBCZ-3A TMS320VC5506ZHH DSPIC33FJ256GP510A-I/PT DSPIC33FJ16GS502-50I/SP DSPIC33FJ06GS102-I/SP DSPIC33EP32GS202-I/MX DSPIC33EP32GS202-I/M6 DSPIC33EP16GS506-I/PT DSPIC33EP16GS202-I/M6 DSPIC30F2011-30I/ML DSPIC33FJ32GP202-E/MM DSPIC33FJ256GP510A-I/PF DSPIC33EP16GS504-I/PT DSPIC30F2012-30I/ML