dsPIC30F2011, dsPIC30F2012, dsPIC30F3012, dsPIC30F3013 Data Sheet

High-Performance

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.

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, MXDEV, MXLAB, PICMASTER, 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, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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

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

© 2005, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

MICROCHIP

dsPIC30F2011/2012/3012/3013

dsPIC30F2011/2012/3012/3013 High-Performance Digital Signal Controllers

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the *dsPIC30F Family Reference Manual* (DS70046). For more information on the device instruction set and programming, refer to the *dsPIC30F Programmer's Reference Manual* (DS70030).

High-Performance Modified RISC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set architecture
- Flexible addressing modes
- 84 base instructions
- 24-bit wide instructions, 16-bit wide data path
- Up to 24 Kbytes on-chip Flash program space
- Up to 2 Kbytes of on-chip data RAM
- Up to 1 Kbytes of non-volatile data EEPROM
- 16 x 16-bit working register array
- Up to 30 MIPs operation:
  - DC to 40 MHz external clock input
  - 4 MHz-10 MHz oscillator input with PLL active (4x, 8x, 16x)
- Up to 21 interrupt sources:
  - 8 user selectable priority levels
  - 3 external interrupt sources
  - 4 processor trap sources

DSP Features:

- Dual data fetch
- Modulo and Bit-reversed modes
- Two 40-bit wide accumulators with optional saturation logic
- 17-bit x 17-bit single cycle hardware fractional/integer multiplier
- All DSP instructions are single cycle
  - Multiply-Accumulate (MAC) operation
- Single cycle ±16 shift

Peripheral Features:

- High current sink/source I/O pins: 25 mA/25 mA
- Three 16-bit timers/counters; optionally pair up 16-bit timers into 32-bit timer modules
- 16-bit Capture input functions

- 16-bit Compare/PWM output functions:
- 3-wire SPI™ modules (supports four Frame modes)
- I²C™ module supports Multi-Master/Slave mode and 7-bit/10-bit addressing
- Up to two addressable UART modules with FIFO buffers

Analog Features:

- 12-bit Analog-to-Digital Converter (A/D) with:
  - 100 Ksps conversion rate
  - Up to 10 input channels
  - Conversion available during Sleep and Idle
- Programmable Low Voltage Detection (PLVD)
- Programmable Brown-out Detection and Reset generation

Special Microcontroller Features:

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

CMOS Technology:

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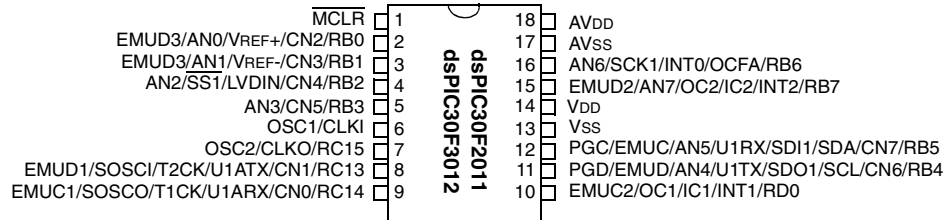- Low power consumption

dsPIC30F2011/2012/3012/3013
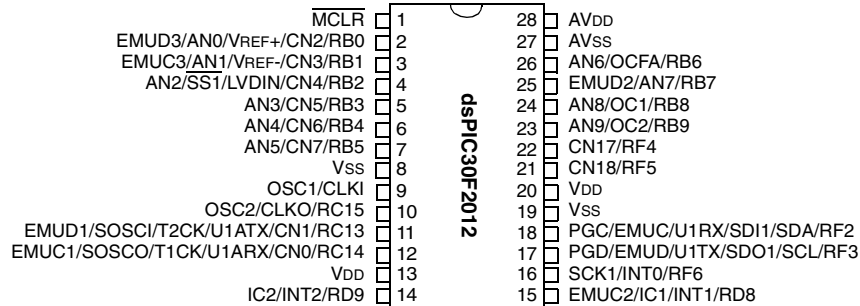
dsPIC30F2011/2012/3012/3013 Sensor Family

Device	Pins	Program Memory		SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp/Std PWM	A/D 12-bit 100 Ksps	UART	SPI™	I²C™
		Bytes	Instructions									
dsPIC30F2011	18	12K	4K	1024	–	3	2	2	8 ch	1	1	1
dsPIC30F3012	18	24K	8K	2048	1024	3	2	2	8 ch	1	1	1
dsPIC30F2012	28	12K	4K	1024	–	3	2	2	10 ch	1	1	1
dsPIC30F3013	28	24K	8K	2048	1024	3	2	2	10 ch	2	1	1

Pin Diagrams

18-Pin PDIP and SOIC

dsPIC30F2011
dsPIC30F3012

$\overline{MCLR}$	1	18 AV$_{DD}$
EMUD3/AN0/V$_{REF}$+/CN2/RB0	2	17 AV$_{SS}$
EMUD3/$\overline{AN1}$/V$_{REF}$-/CN3/RB1	3	16 AN6/SCK1/INT0/OCFA/RB6
AN2/$\overline{SS1}$/LVDIN/CN4/RB2	4	15 EMUD2/AN7/OC2/IC2/INT2/RB7
AN3/CN5/RB3	5	14 V$_{DD}$
OSC1/CLKI	6	13 V$_{SS}$
OSC2/CLKO/RC15	7	12 PGC/EMUC/AN5/U1RX/SDI1/SDA/CN7/RB5
EMUD1/SOSCI/T2CK/U1ATX/CN1/RC13	8	11 PGD/EMUD/AN4/U1TX/SDO1/SCL/CN6/RB4
EMUC1/SOSCO/T1CK/U1ARX/CN0/RC14	9	10 EMUC2/OC1/IC1/INT1/RD0

28-Pin PDIP and SOIC

dsPIC30F2012

$\overline{MCLR}$	1	28 AV$_{DD}$
EMUD3/AN0/V$_{REF}$+/CN2/RB0	2	27 AV$_{SS}$
EMUC3/$\overline{AN1}$/V$_{REF}$-/CN3/RB1	3	26 AN6/OCFA/RB6
AN2/$\overline{SS1}$/LVDIN/CN4/RB2	4	25 EMUD2/AN7/RB7
AN3/CN5/RB3	5	24 AN8/OC1/RB8
AN4/CN6/RB4	6	23 AN9/OC2/RB9
AN5/CN7/RB5	7	22 CN17/RF4
V$_{SS}$	8	21 CN18/RF5
OSC1/CLKI	9	20 V$_{DD}$
OSC2/CLKO/RC15	10	19 V$_{SS}$
EMUD1/SOSCI/T2CK/U1ATX/CN1/RC13	11	18 PGC/EMUC/U1RX/SDI1/SDA/RF2
EMUC1/SOSCO/T1CK/U1ARX/CN0/RC14	12	17 PGD/EMUD/U1TX/SDO1/SCL/RF3
V$_{DD}$	13	16 SCK1/INT0/RF6
IC2/INT2/RD9	14	15 EMUC2/IC1/INT1/RD8

28-Pin SPDIP and SOIC

dsPIC30F3013

$\overline{MCLR}$	1	28 AV$_{DD}$
EMUD3/AN0/V$_{REF}$+/CN2/RB0	2	27 AV$_{SS}$
EMUC3/$\overline{AN1}$/V$_{REF}$-/CN3/RB1	3	26 AN6/OCFA/RB6
AN2/$\overline{SS1}$/LVDIN/CN4/RB2	4	25 EMUD2/AN7/RB7
AN3/CN5/RB3	5	24 AN8/OC1/RB8
AN4/CN6/RB4	6	23 AN9/OC2/RB9
AN5/CN7/RB5	7	22 U2RX/CN17/RF4
V$_{SS}$	8	21 U2TX/CN18/RF5
OSC1/CLKI	9	20 V$_{DD}$
OSC2/CLKO/RC15	10	19 V$_{SS}$
EMUD1/SOSCI/T2CK/U1ATX/CN1/RC13	11	18 PGC/EMUC/U1RX/SDI1/SDA/RF2
EMUC1/SOSCO/T1CK/U1ARX/CN0/RC14	12	17 PGD/EMUD/U1TX/SDO1/SCL/RF3
V$_{DD}$	13	16 SCK1/INT0/RF6
IC2/INT2/RD9	14	15 EMUC2/IC1/INT1/RD8

Note: For descriptions of individual pins, see Section 1.0.

Pin Diagrams

28-Pin QFN

Top pins (left to right, 28→22):
- EMUC3/AN1/VREF-/CN3/RB1
- EMUD3/AN0/VREF+/CN2/RB0
- MCLR
- AVDD
- AVSS
- AN6/SCK1/INT0/OCFA/RB6
- EMUD2/AN7/OC2/IC2/INT2/RB7

Left pins (1→7):
- 1 — AN2/SS1/LVDIN/CN4/RB2
- 2 — AN3/CN5/RB3
- 3 — NC
- 4 — NC
- 5 — VSS
- 6 — OSC1/CLKI
- 7 — OSC2/CLKO/RC15

dsPIC30F2011

Right pins (21→15):
- 21 — NC
- 20 — NC
- 19 — NC
- 18 — NC
- 17 — VDD
- 16 — VSS
- 15 — PGC/EMUC/AN5/U1RX/SDI1/SDA/CN7/RB5

Bottom pins (8→14):
- 8 — EMUD1/SOSCI/T2CK/U1ATX/CN1/RC13
- 9 — EMUC1/SOSCO/T1CK/U1ARX/CN0/RC14
- 10 — VDD
- 11 — NC
- 12 — EMUC2/OC1/IC1/INT1/RD0
- 13 — NC
- 14 — PGD/EMUD/AN4/U1TX/SDO1/SCL/CN6/RB4

Note: For descriptions of individual pins, see Section 1.0.

dsPIC30F2011/2012/3012/3013

Pin Diagrams

28-Pin QFN

Note: For descriptions of individual pins, see Section 1.0.

Pin Diagram

44-Pin QFN

dsPIC30F3012

Top side pins (left to right, 44–34):
PGD/EMUD/AN4/U1TX/SDO1/SCL/CN6/RB4
NC
EMUC2/OC1/IC1/INT1/RD0
NC
NC
NC
NC
NC
V$_{DD}$
EMUC1/SOSCO/T1CK/U1ARX/CN0/RC14
EMUD1/SOSCI/T2CK/U1ATX/CN1/RC13

Left side pins (1–11):
Pin	Signal
1	PGC/EMUC/AN5/U1RX/SDI1/SDA/CN7/RB5
2	V$_{SS}$
3	NC
4	V$_{DD}$
5	NC
6	NC
7	NC
8	NC
9	NC
10	NC
11	NC

Right side pins (33–23):
Pin	Signal
33	OSC2/CLKO/RC15
32	OSC1/CLKI
31	V$_{SS}$
30	V$_{SS}$
29	NC
28	NC
27	NC
26	NC
25	AN3/CN5/RB3
24	NC
23	AN2/$\overline{SS1}$/LVDIN/CN4/RB2

Bottom side pins (left to right, 12–22):
EMUD2/AN7/OC2/IC2/INT2/RB7
NC
AN6/SCK1/INT0/OCFA/RB6
NC
AV$_{SS}$
AV$_{DD}$
$\overline{MCLR}$
EMUD3/AN0/V$_{REF+}$/CN2/RB0
EMUC3/AN1/V$_{REF-}$/CN3/RB1
NC
NC

Note: For descriptions of individual pins, see Section 1.0.

dsPIC30F2011/2012/3012/3013

Pin Diagrams

44-Pin QFN

Top side pins (34–44), left to right:
EMUD1/SOSCI/T2CK/U1ATX/CN1/RC13 (34), EMUC1/SOSCO/T1CK/U1ARX/CN0/RC14 (35), VDD (36), IC2/INT2/RD9 (37), NC (38), NC (39), NC (40), EMUC2/IC1/INT1/RD8 (42), SCK1/INT0/RF6 (43), PGD/EMUD/U1TX/SDO1/SCL/RF3 (44)

Left side pins (1–11):
- PGC/EMUC/U1RX/SDI1/SDA/RF2 — 1
- Vss — 2
- NC — 3
- VDD — 4
- NC — 5
- NC — 6
- U2TX/CN18/RF5 — 7
- NC — 8
- U2RX/CN17/RF4 — 9
- AN9/OC2/RB9 — 10
- AN8/OC1/RB8 — 11

Center: **dsPIC30F3013**

Right side pins (23–33):
- 33 — OSC2/CLKO/RC15
- 32 — OSC1/CLKI
- 31 — Vss
- 30 — Vss
- 29 — NC
- 28 — NC
- 27 — AN5/CN7/RB5
- 26 — AN4/CN6/RB4
- 25 — AN3/CN5/RB3
- 24 — NC
- 23 — AN2/SS1/LVDIN/CN4/RB2

Bottom side pins (12–22), left to right:
EMUD2/AN7/RB7 (12), NC (13), AN6/OCFA/RB6 (14), NC (15), AVss (16), AVDD (17), MCLR (18), EMUD3/AN0/VREF+/CN2/RB0 (19), EMUC3/AN1/VREF-/CN3/RB1 (20), NC (21), NC (22)

Note: For descriptions of individual pins, see Section 1.0.

Preliminary

Table of Contents

TO OUR VALUED CUSTOMERS

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

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

Most Current Data Sheet

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

> http://www.microchip.com

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

Errata

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

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

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

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

Customer Notification System

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

NOTES:

Preliminary

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 Programmer's Reference Manual* (DS70030).

This data sheet contains information specific to the dsPIC30F2011, dsPIC30F2012, dsPIC30F3012 and dsPIC30f3013 Digital Signal Controllers. These devices contain extensive Digital Signal Processor (DSP) functionality within a high-performance 16-bit microcontroller (MCU) architecture.

The following block diagrams depict the architecture for these devices:

- Figure 1-1 illustrates the dsPIC30F2011
- Figure 1-2 illustrates the dsPIC30F2012
- Figure 1-3 illustrates the dsPIC30F3012
- Figure 1-4 illustrates the dsPIC30F3013

Following the block diagrams, Table 1-1 relates the I/O functions to pinout information.
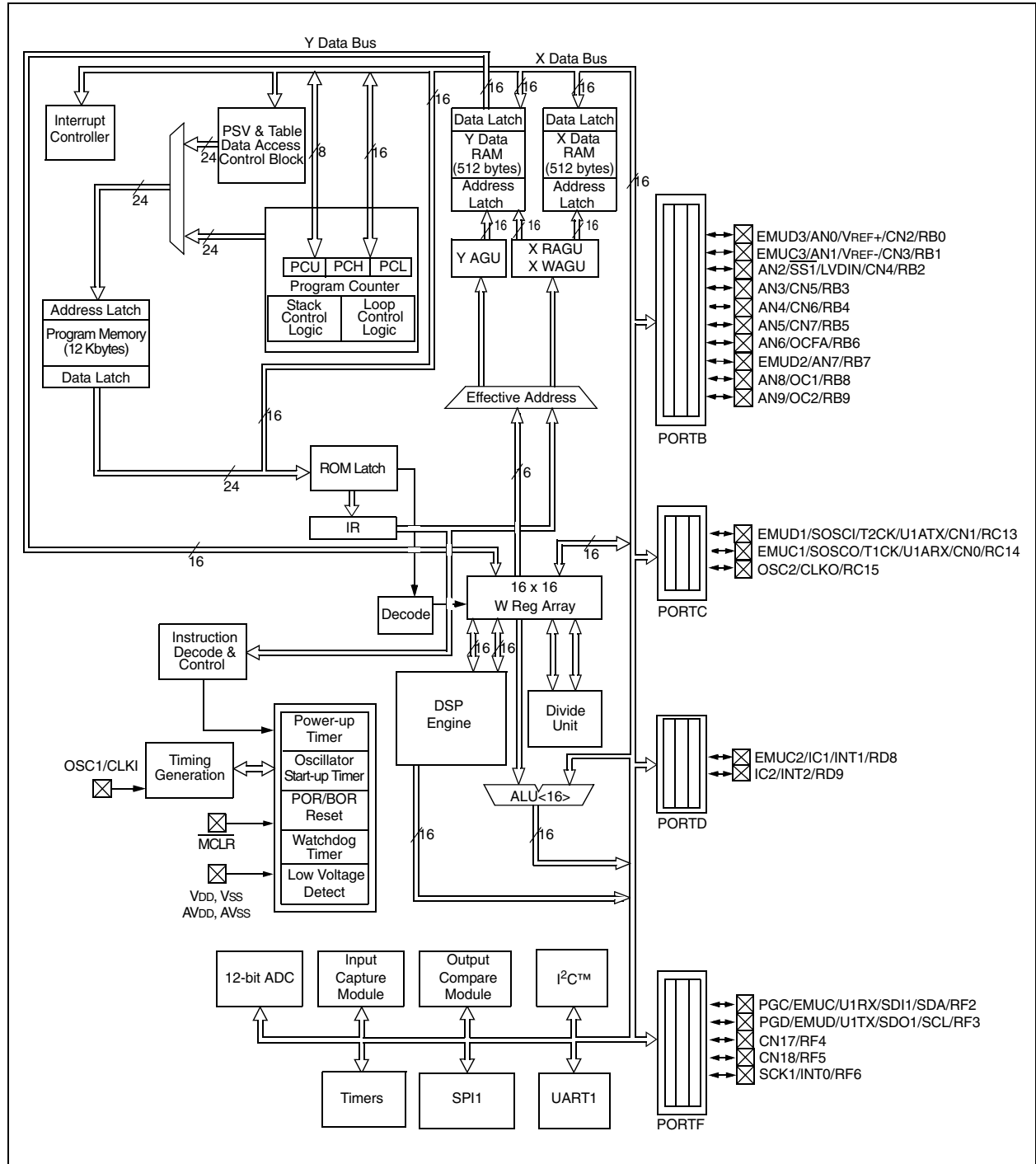
FIGURE 1-1: **dsPIC30F2011 BLOCK DIAGRAM**
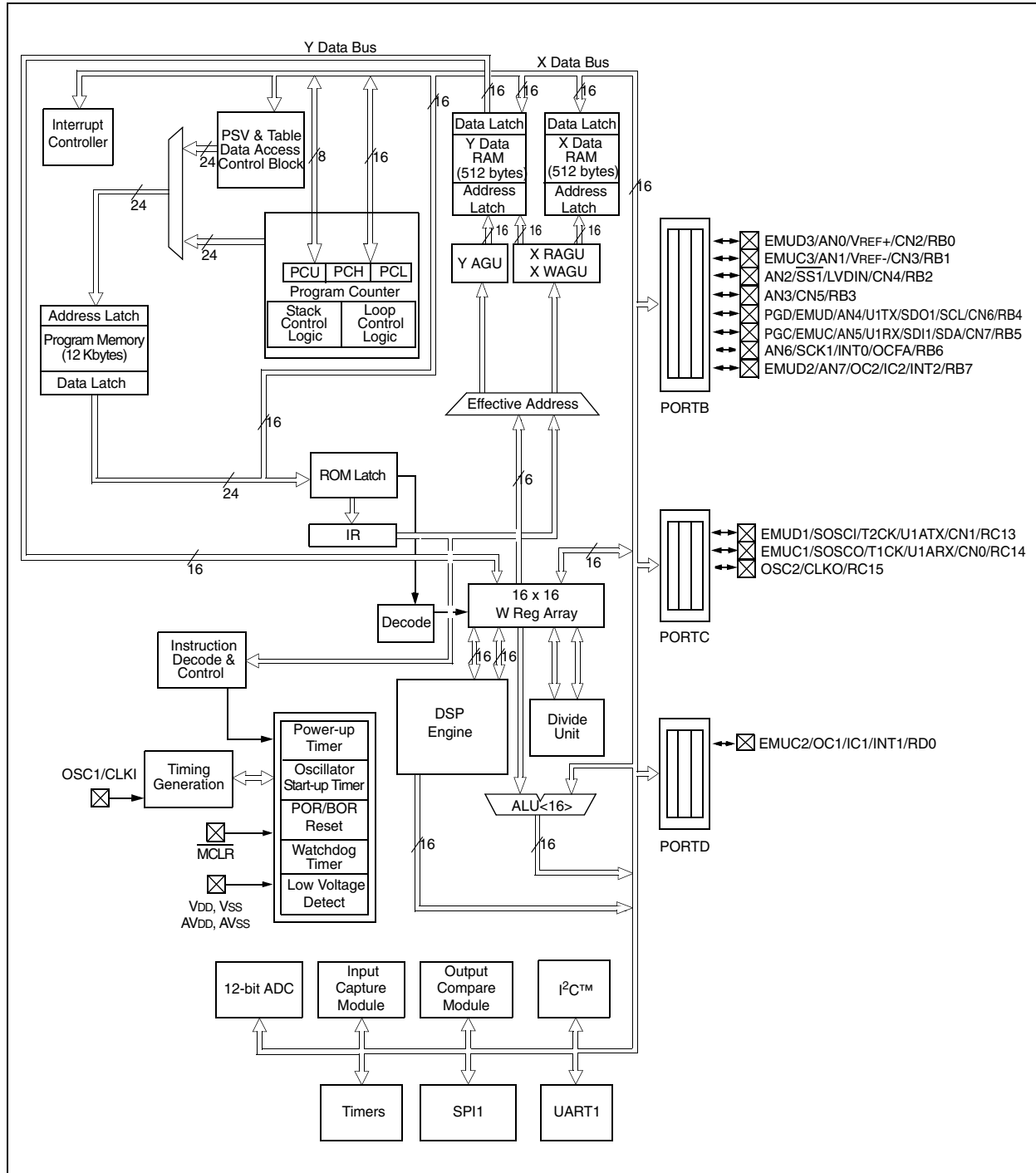
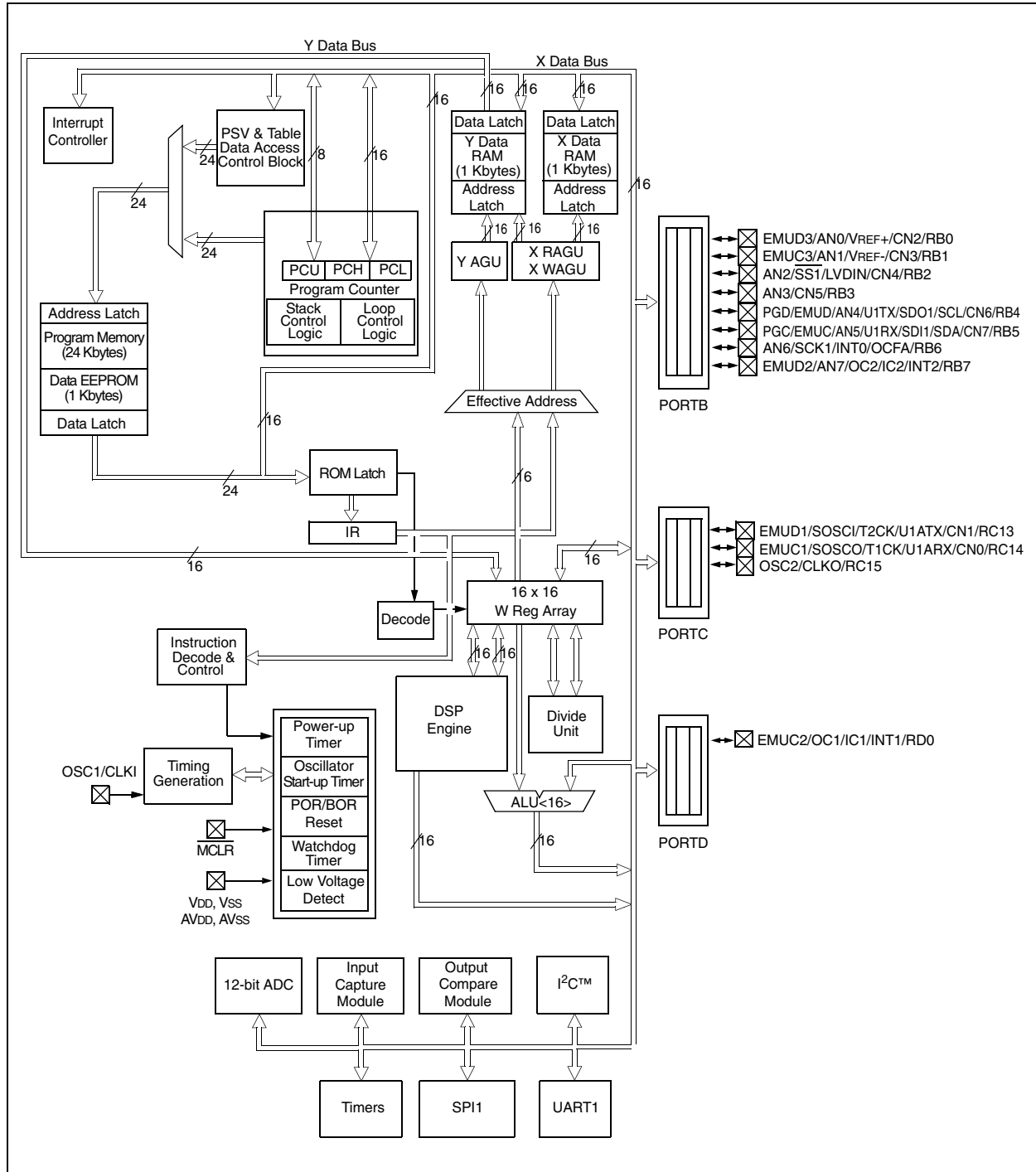**FIGURE 1-2:** **dsPIC30F2012 BLOCK DIAGRAM**

FIGURE 1-3: **dsPIC30F3012 BLOCK DIAGRAM**

Preliminary   © 2005 Microchip Technology Inc.

FIGURE 1-4: **dsPIC30F3013 BLOCK DIAGRAM**
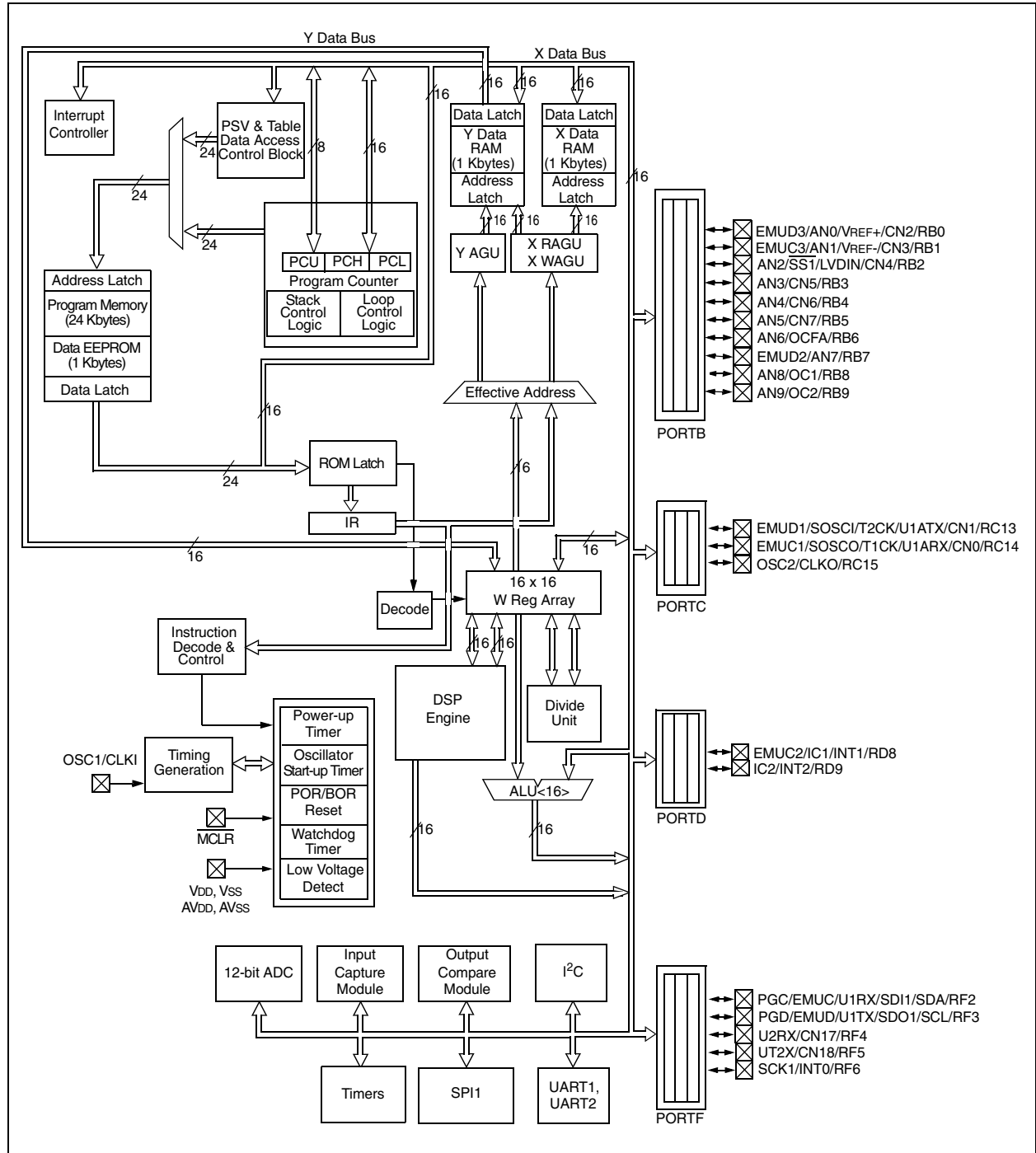
Table 1-1 provides a brief description of device I/O pinouts and the functions that may be multiplexed to a port pin. Multiple functions may exist on one port pin. When multiplexing occurs, the peripheral module's functional requirements may force an override of the data direction of the port pin.

TABLE 1-1: PINOUT I/O DESCRIPTIONS

Pin Name	Pin Type	Buffer Type	Description
AN0-AN9	I	Analog	Analog input channels.
AV$_{DD}$	P	P	Positive supply for analog module.
AV$_{SS}$	P	P	Ground reference for analog module.
CLKI	I	ST/CMOS	External clock source input. Always associated with OSC1 pin function.
CLKO	O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes. Always associated with OSC2 pin function.
CN0-CN7	I	ST	Input change notification inputs. Can be software programmed for internal weak pull-ups on all inputs.
EMUD	I/O	ST	ICD Primary Communication Channel data input/output pin.
EMUC	I/O	ST	ICD Primary Communication Channel clock input/output pin.
EMUD1	I/O	ST	ICD Secondary Communication Channel data input/output pin.
EMUC1	I/O	ST	ICD Secondary Communication Channel clock input/output pin.
EMUD2	I/O	ST	ICD Tertiary Communication Channel data input/output pin.
EMUC2	I/O	ST	ICD Tertiary Communication Channel clock input/output pin.
EMUD3	I/O	ST	ICD Quaternary Communication Channel data input/output pin.
EMUC3	I/O	ST	ICD Quaternary Communication Channel clock input/output pin.
IC1-IC2	I	ST	Capture inputs 1 through 2.
INT0	I	ST	External interrupt 0.
INT1	I	ST	External interrupt 1.
INT2	I	ST	External interrupt 2.
LVDIN	I	Analog	Low Voltage Detect Reference Voltage input pin.
MCLR	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low Reset to the device.
OC1-OC2	O	—	Compare outputs 1 through 2.
OCFA	I	ST	Compare Fault A input.
OSC1	I	ST/CMOS	Oscillator crystal input. ST buffer when configured in RC mode; CMOS otherwise.
OSC2	I/O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes.
PGD	I/O	ST	In-Circuit Serial Programming data input/output pin.
PGC	I	ST	In-Circuit Serial Programming clock input pin.
RB0-RB9	I/O	ST	PORTB is a bidirectional I/O port.
RC13-RC15	I/O	ST	PORTC is a bidirectional I/O port.
RD0, RD8-RD9	I/O	ST	PORTD is a bidirectional I/O port.
RF2-RF5	I/O	ST	PORTF is a bidirectional I/O port.
SCK1	I/O	ST	Synchronous serial clock input/output for SPI1.
SDI1	I	ST	SPI1 Data In.
SDO1	O	—	SPI1 Data Out.
SS1	I	ST	SPI1 Slave Synchronization.

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

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

Pin Name	Pin Type	Buffer Type	Description
SCL SDA	I/O I/O	ST ST	Synchronous serial clock input/output for I^{2}C. Synchronous serial data input/output for I^{2}C.
SOSCO SOSCI	O I	— ST/CMOS	32 kHz low power oscillator crystal output. 32 kHz low power oscillator crystal input. ST buffer when configured in RC mode; CMOS otherwise.
T1CK T2CK	I I	ST ST	Timer1 external clock input. Timer2 external clock input.
U1RX U1TX U1ARX U1ATX U2RX U2TX	I O I O I O	ST — ST — ST —	UART1 Receive. UART1 Transmit. UART1 Alternate Receive. UART1 Alternate Transmit. UART2 Receive. UART2 Transmit.
VDD	P	—	Positive supply for logic and I/O pins.
VSS	P	—	Ground reference for logic and I/O pins.
VREF+	I	Analog	Analog Voltage Reference (High) input.
VREF-	I	Analog	Analog Voltage Reference (Low) input.

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

NOTES:

Preliminary

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 Programmer's Reference Manual* (DS70030).

This section is an overview of the CPU architecture of the dsPIC30F. The core has a 24-bit instruction word. The Program Counter (PC) is 23-bits wide with the Least Significant (LS) bit always clear (see Section 3.1). The Most Significant (MS) bit 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 pre-fetch mechanism helps 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.

2.1 Core Overview

The working register array consists of 16 x 16-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). 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.

Two ways to access data in program memory are:

- 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. Thus any instruction can access program space as if it were data space, with a limitation that the access requires an additional cycle. 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 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 15 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 pre-fetch 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 16 x 16-bit working registers (W0 through W15), 2 x 40-bit accumulators (AccA and AccB), STATUS register (SR), Data Table Page register (TBLPAG), Program Space Visibility Page register (PSVPAG), DO and REPEAT registers (DOSTART, DOEND, DCOUNT and RCOUNT) and Program Counter (PC). The working registers can act as data, address or offset registers. All registers are memory mapped. W0 acts as the W register for file register addressing.

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

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

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

2.2.1 SOFTWARE STACK POINTER/ FRAME POINTER

The dsPIC® 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 core has a 16-bit STATUS register (SR), the LS Byte of which is referred to as the SR Low byte (SRL) and the MS Byte 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 MS Byte of the PC to form a complete word value which is then stacked.
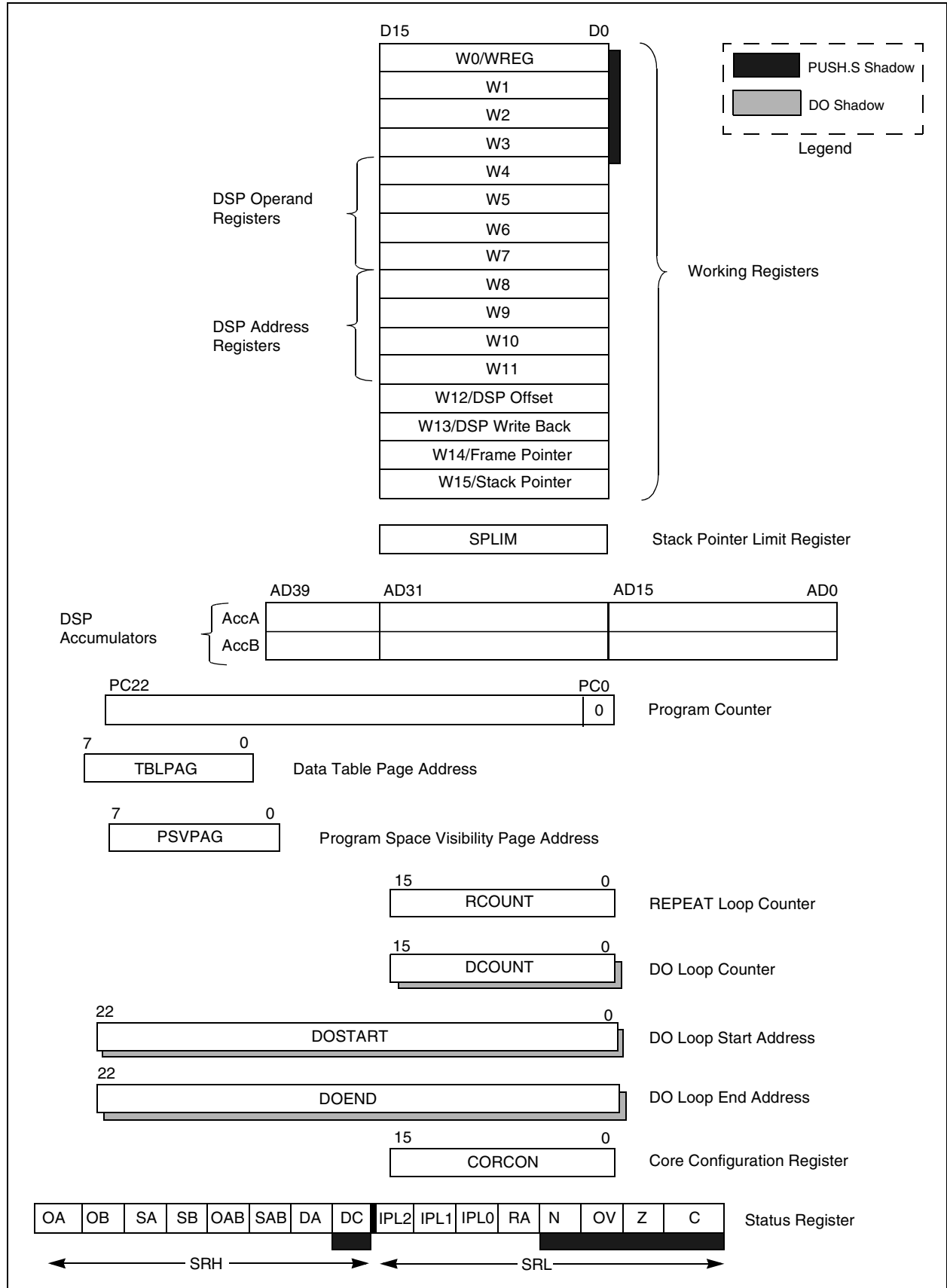
The upper byte of the STATUS register contains the DSP Adder/Subtracter status bits, the DO Loop Active bit (DA) and the Digit Carry (DC) status bit.

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.

FIGURE 2-1:   **PROGRAMMER'S MODEL**

dsPIC30F2011/2012/3012/3013

2.3   Divide Support

The dsPIC 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 16/16 divides are similar to the 32/16 (same number of iterations), but the dividend is either zero-extended or sign-extended during the first iteration.

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 setup 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.

TABLE 2-1:   DIVIDE INSTRUCTIONS

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.sw or DIV.s`	Signed divide: Wm/Wn $\rightarrow$ W0; Rem $\rightarrow$ W1
`DIV.ud`	Unsigned divide: (Wm+1:Wm)/Wn $\rightarrow$ W0; Rem $\rightarrow$ W1
`DIV.uw or DIV.u`	Unsigned divide: Wm/Wn $\rightarrow$ W0; Rem $\rightarrow$ W1

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/subtracter (with two target accumulators, round and saturation logic).

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 dsPIC30F is a single-cycle instruction flow architecture, threfore, concurrent operation of the DSP engine with MCU instruction flow is not possible. However, some MCU ALU and DSP engine resources may be used concurrently by the same instruction (e.g., ED, EDAC).

The DSP engine 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).
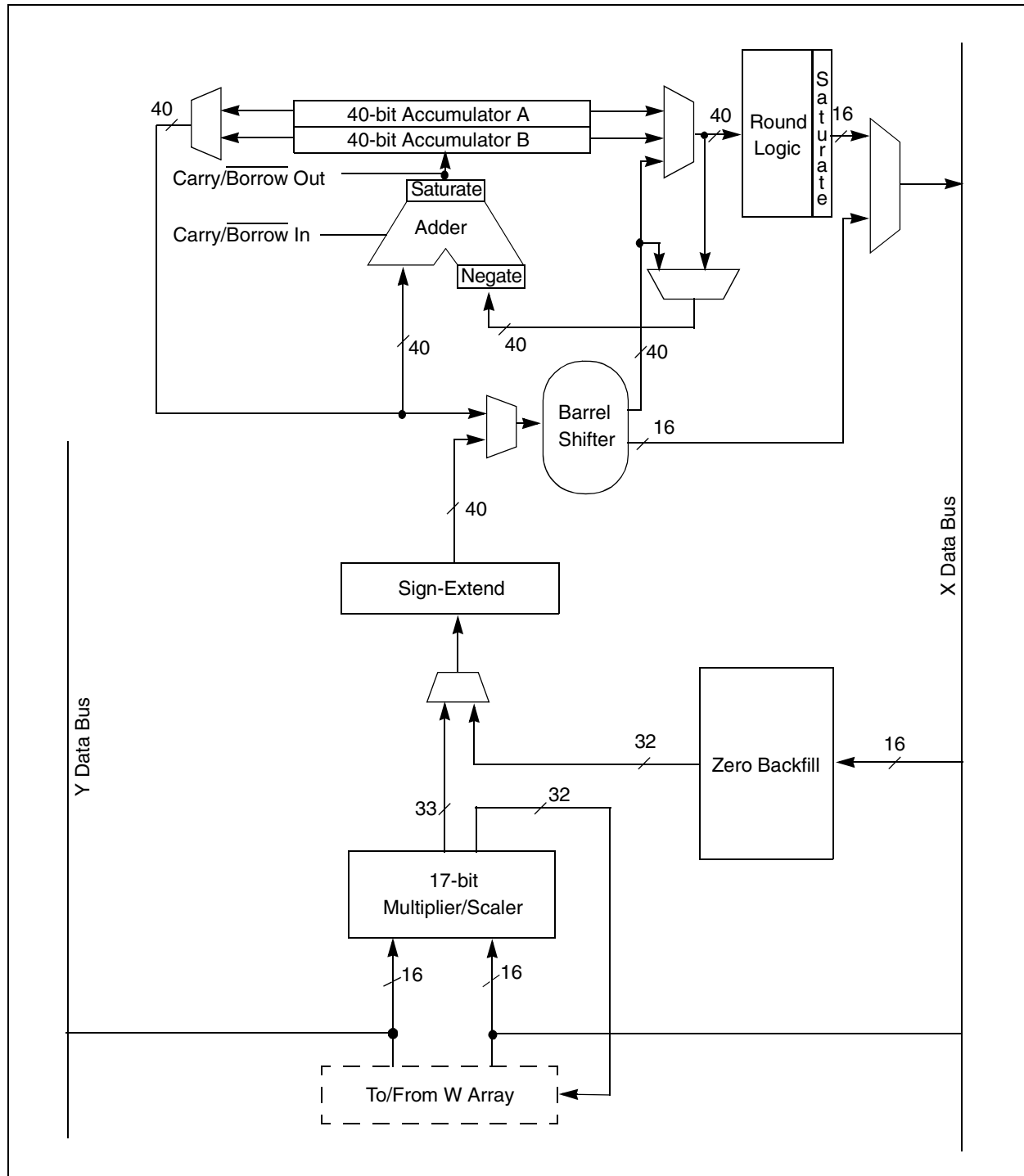
> **Note:**   For CORCON layout, see Table 3-3.

A block diagram of the DSP engine is shown in Figure 2-2.

TABLE 2-2:   DSP INSTRUCTION SUMMARY

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

FIGURE 2-2: **DSP ENGINE BLOCK DIAGRAM**

2.4.1 MULTIPLIER

The 17 x 17-bit multiplier is capable of signed or unsigned operation and can multiplex its output using a scaler to support either 1.31 fractional (Q31) or 32-bit integer results. Unsigned operands are zero-extended into the 17th bit of the multiplier input value. Signed operands are sign-extended into the 17th bit of the multiplier input value. The output of the 17 x 17-bit multiplier/scaler is a 33-bit value which is sign-extended to 40 bits. Integer data is inherently represented as a signed two's complement value, where the MSB is defined as a sign bit. Generally speaking, the range of an N-bit two's complement integer is -2^{N-1} to $2^{N-1} - 1$. For a 16-bit integer, the data range is -32768 (0x8000) to 32767 (0x7FFF) including '0'. For a 32-bit integer, the data range is -2,147,483,648 (0x8000 0000) to 2,147,483,645 (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 3.01518×10^{-5}. In Fractional mode, the 16x16 multiply operation generates a 1.31 product which has a precision of 4.65661×10^{-10}.

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

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

2.4.2 DATA ACCUMULATORS AND ADDER/SUBTRACTER

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

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 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 (OVATEN, OVBTEN) in the INTCON1 register (refer to Section 8.0) 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 (0x8000000000) into the target accumulator. The SA or SB bit is set and remains set until cleared by the user. This is referred to as 'super saturation' and provides protection against erroneous data, or unexpected algorithm problems (e.g., gain calculations).

2. Bit 31 Overflow and Saturation:
   When bit 31 overflow and saturation occurs, the saturation logic then loads the maximally positive 1.31 value (0x007FFFFFFF), or maximally negative 1.31 value (0x0080000000) into the target accumulator. The SA or SB bit is set and remains set until cleared by the user. When this Saturation mode is in effect, the guard bits are not used (so the OA, OB or OAB bits are never set).

3. Bit 39 Catastrophic Overflow:
   The bit 39 overflow status bit from the adder is used to set the SA or SB bit which remain set until cleared by the user. No saturation operation is performed and the accumulator is allowed to overflow (destroying its sign). If the COVTE bit in the INTCON1 register is set, a catastrophic overflow can initiate a trap exception.

2.4.2.2 Accumulator 'Write Back'

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

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

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

2.4.2.3 Round Logic

The round logic is a combinational block which performs a conventional (biased) or convergent (unbiased) round function during an accumulator write (store). The Round mode is determined by the state of the RND bit in the CORCON register. It generates a 16-bit, 1.15 data value which is passed to the data space write saturation logic. If rounding is not indicated by the instruction, a truncated 1.15 data value is stored and the LS 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 LS bit (bit 16 of the accumulator) of ACCxH is examined. If it is '1', ACCxH is incremented. If it is '0', ACCxH is not modified. Assuming that bit 16 is effectively random in nature, this scheme 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). 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 MS bit of the source (bit 39) is used to determine the sign of the operand being tested.

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

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 16 for left shifts.

NOTES:

Preliminary   © 2005 Microchip Technology Inc.

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 Programmer's Reference Manual* (DS70030).

3.1   Program Address Space

The program address space is 4M instruction words. The program space memory map for the dsPI30F2011/2012 is shown in Figure 3-1. The program space memory map for the dsPI30F3012/3013 is shown in Figure 3-2.

Program memory is addressable by a 24-bit value from either 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, Program Space Address Construction, bit 23 allows access to the Device ID, the User ID and the configuration bits. Otherwise, bit 23 is always clear.

dsPIC30F2011/2012/3012/3013

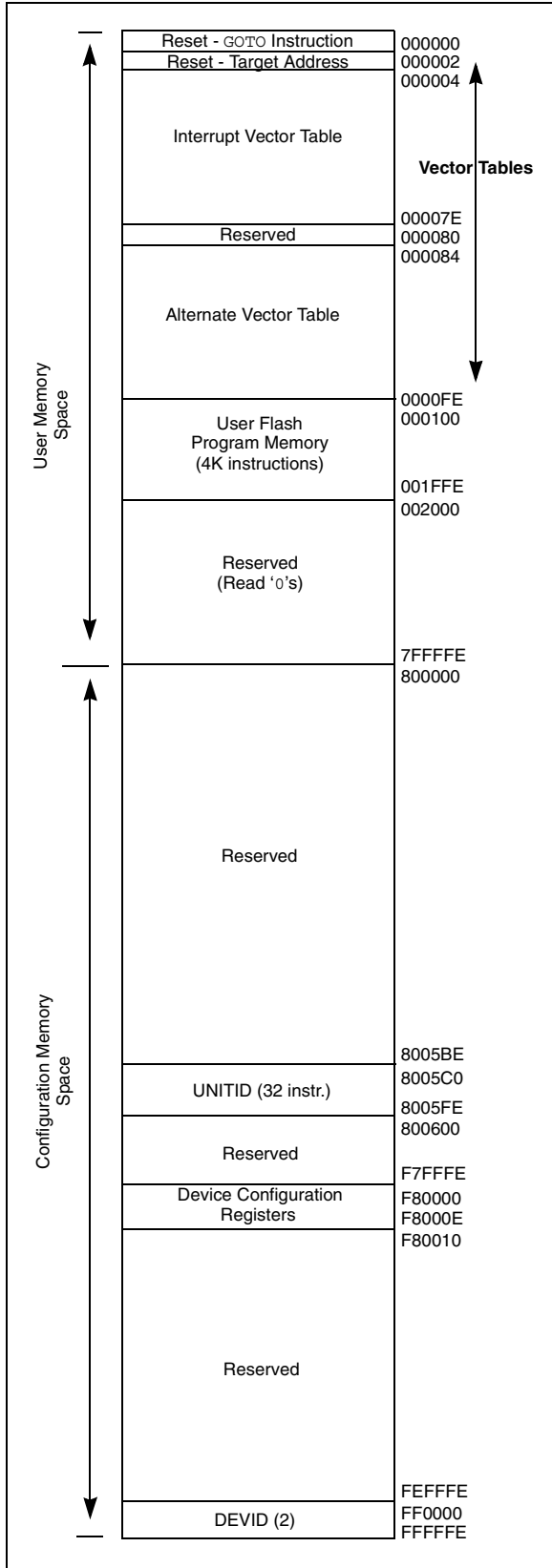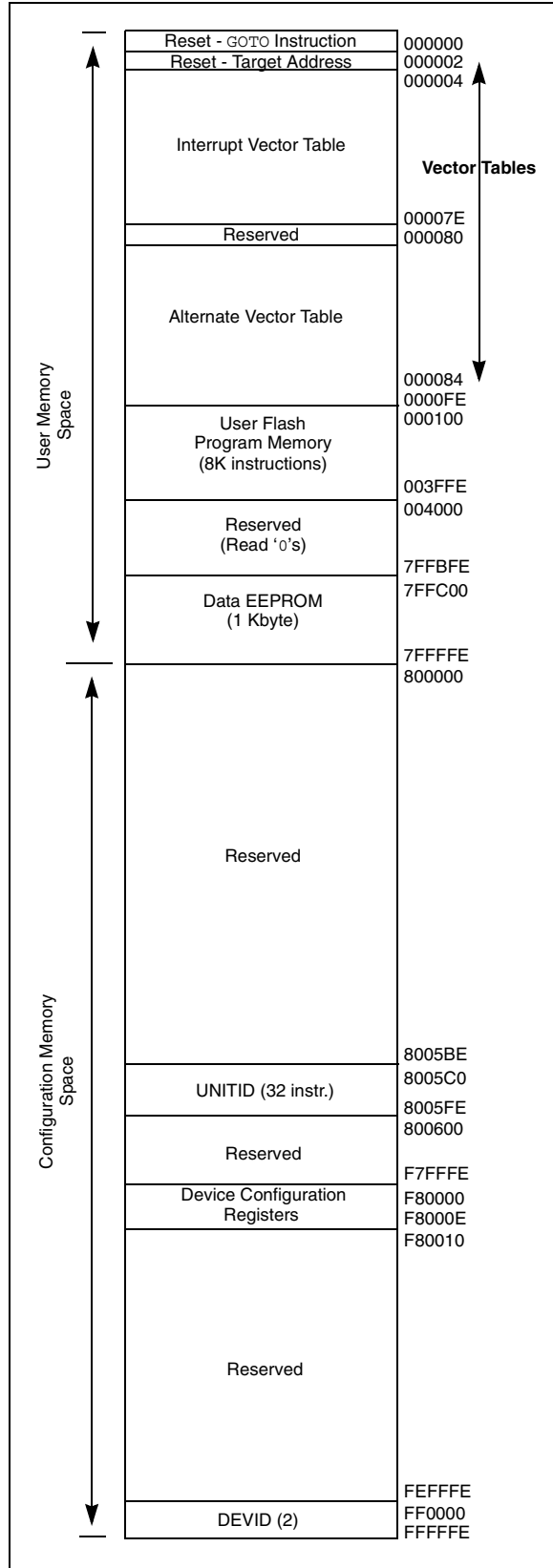**FIGURE 3-1:** **dsPIC30F2011/2012 PROGRAM SPACE MEMORY MAP**

	Address
Reset - GOTO Instruction	000000
Reset - Target Address	000002
	000004
Interrupt Vector Table	
Vector Tables	
	00007E
Reserved	000080
	000084
Alternate Vector Table	
	0000FE
User Flash Program Memory (4K instructions)	000100
	001FFE
	002000
Reserved (Read '0's)	
	7FFFFE
	800000
Reserved	
	8005BE
	8005C0
UNITID (32 instr.)	
	8005FE
	800600
Reserved	
	F7FFFE
Device Configuration Registers	F80000
	F8000E
	F80010
Reserved	
	FEFFFE
DEVID (2)	FF0000
	FFFFFE

User Memory Space

Configuration Memory Space

FIGURE 3-2: **dsPIC30F3012/3013 PROGRAM SPACE MEMORY MAP**

	Address
Reset - GOTO Instruction	000000
Reset - Target Address	000002
	000004
Interrupt Vector Table	
Vector Tables	
	00007E
Reserved	000080
Alternate Vector Table	
	000084
	0000FE
User Flash Program Memory (8K instructions)	000100
	003FFE
	004000
Reserved (Read '0's)	
	7FFBFE
Data EEPROM (1 Kbyte)	7FFC00
	7FFFFE
	800000
Reserved	
	8005BE
	8005C0
UNITID (32 instr.)	
	8005FE
	800600
Reserved	
	F7FFFE
Device Configuration Registers	F80000
	F8000E
	F80010
Reserved	
	FEFFFE
DEVID (2)	FF0000
	FFFFFE

User Memory Space

Configuration Memory Space

Preliminary

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

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

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

Note: Program space visibility cannot be used to access bits <23:16> of a word in program memory.
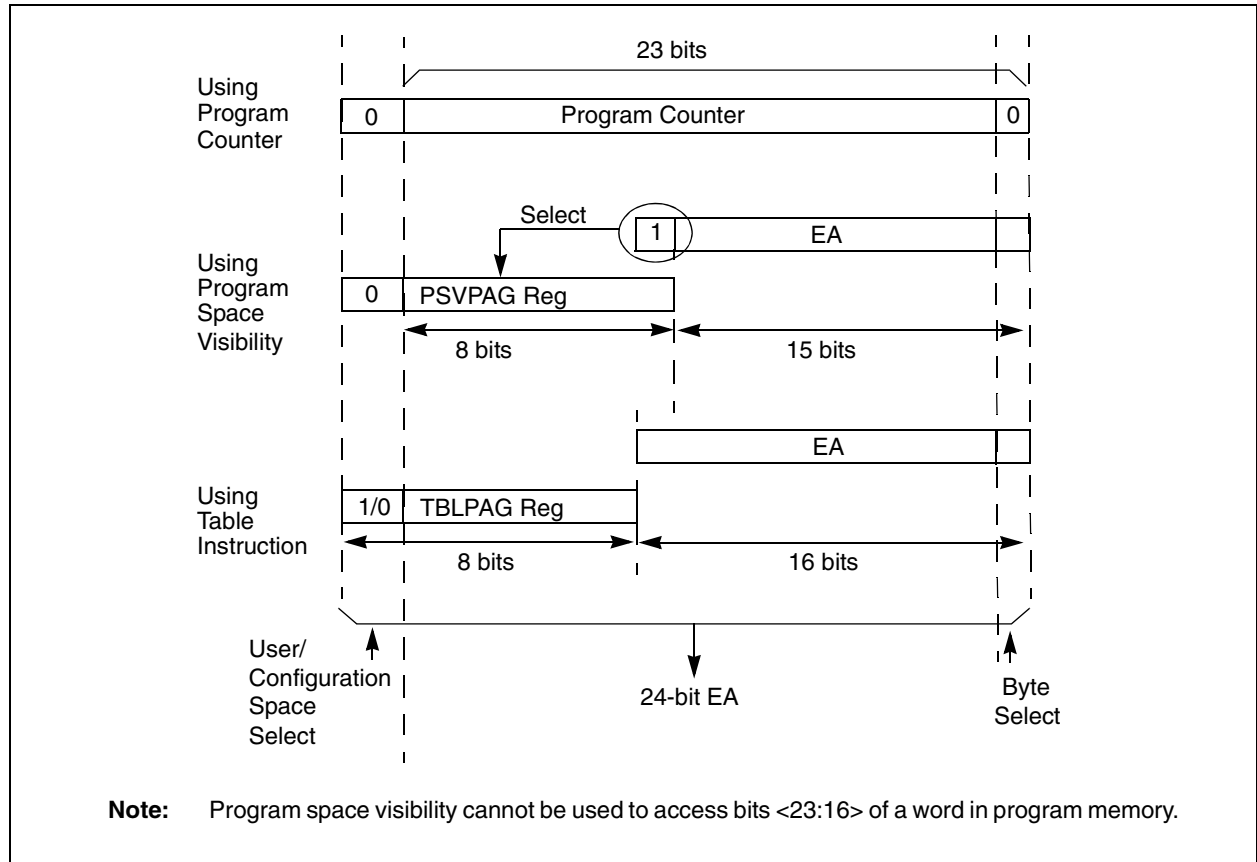
dsPIC30F2011/2012/3012/3013

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). The TBLRDL and TBLWTL instructions offer a direct method of reading or writing the LS 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 LS Data Word, and TBLRDH and TBLWTH access the space which contains the MS Data Byte.

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

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

1. TBLRDL: Table Read Low
   Word: Read the LS Word of the program address; P<15:0> maps to D<15:0>.
   Byte: Read one of the LS Bytes of the program address;
   P<7:0> maps to the destination byte when byte select = 0;
   P<15:8> maps to the destination byte when byte select = 1.
2. TBLWTL: Table Write Low (refer to Section 5.0 for details on Flash Programming)
3. TBLRDH: Table Read High
   Word: Read the MS Word of the program address; P<23:16> maps to D<7:0>; D<15:8> will always be = 0.
   Byte: Read one of the MS Bytes of the program address;
   P<23:16> maps to the destination byte when byte select = 0;
   The destination byte will always be = 0 when byte select = 1.
4. TBLWTH: Table Write High (refer to Section 5.0 for details on Flash Programming)
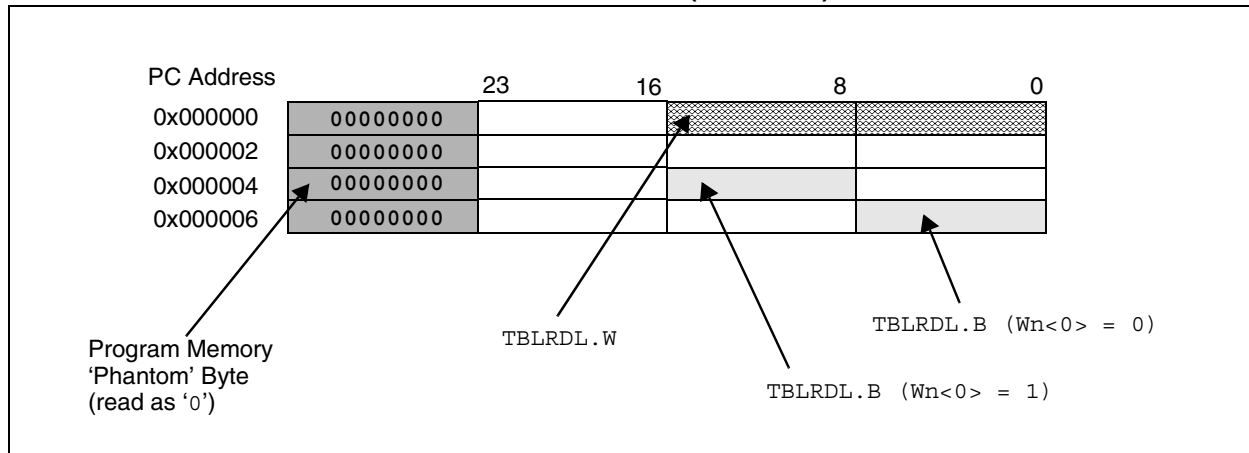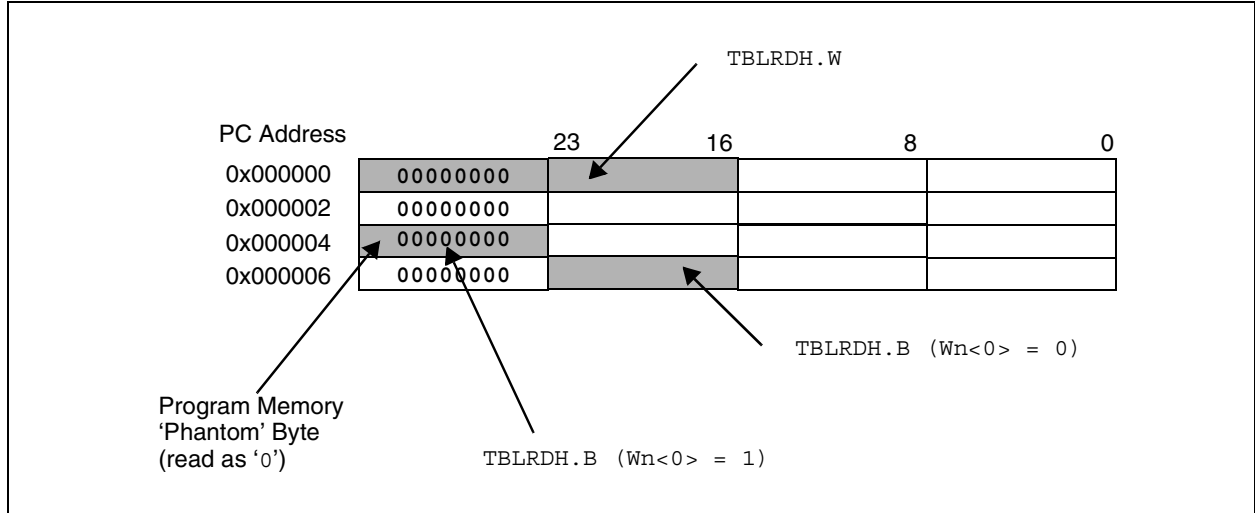
FIGURE 3-4: PROGRAM DATA TABLE ACCESS (LS WORD)

FIGURE 3-5: **PROGRAM DATA TABLE ACCESS (MS BYTE)**

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 MS bit 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.

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 3-6), 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 Programmer's Reference Manual (DS70030) for details on instruction encoding.

Note that by incrementing the PC by 2 for each program memory word, the LS 15 bits of data space addresses directly map to the LS 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 3-6.

> **Note:** PSV access is temporarily disabled during table reads/writes.

For instructions that use PSV which are executed outside a REPEAT loop:
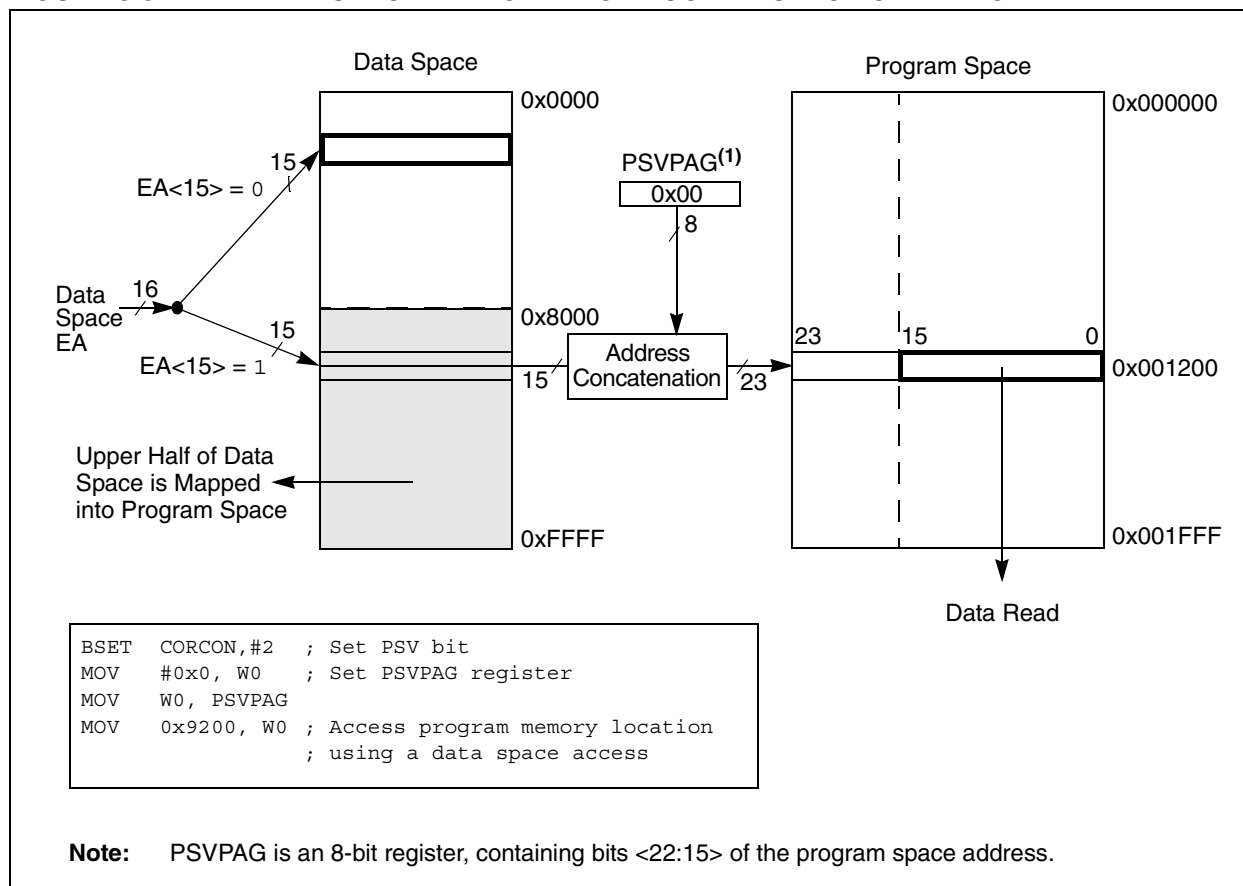
- The following instructions will require one instruction cycle in addition to the specified execution time:
  - MAC class of instructions with data operand pre-fetch
  - MOV instructions
  - MOV.D instructions
- All other instructions will require two instruction cycles in addition to the specified execution time of the instruction.

For instructions that use PSV which are executed inside a 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 3-6: DATA SPACE WINDOW INTO PROGRAM SPACE OPERATION

Data Space

Program Space

0x0000

PSVPAG[1]

0x00

8

EA<15> = 0

15

Data
Space
EA

16

0x000000

15

EA<15> = 1

0x8000

Address
Concatenation

15

23

23

15

0

0x001200

Upper Half of Data
Space is Mapped
into Program Space

0xFFFF

0x001FFF

Data Read

```
BSET   CORCON,#2  ; Set PSV bit
MOV    #0x0, W0   ; Set PSVPAG register
MOV    W0, PSVPAG
MOV    0x9200, W0 ; Access program memory location
                  ; using a data space access
```

Note: PSVPAG is an 8-bit register, containing bits <22:15> of the program space address.

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.
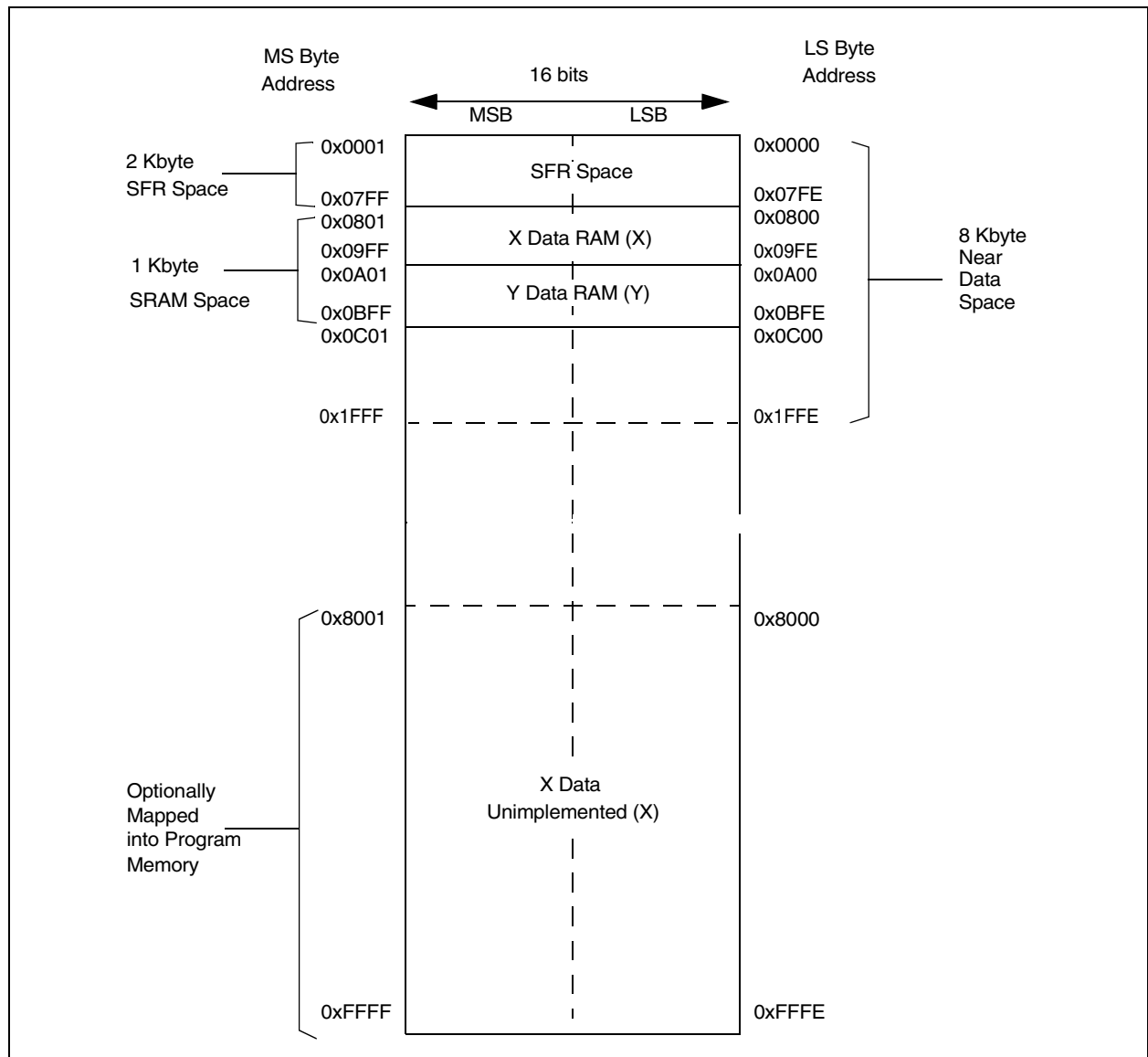
3.2.1 DATA SPACE MEMORY MAP

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.

The data space memory map for the dsPIC30F2011 and dsPIC30F2012 is shown in Figure 3-7. The data space memory map for the dsPIC30F3012 and dsPIC30F3013 is shown in Figure 3-8.

FIGURE 3-7: **dsPIC30F2011/2012 DATA SPACE MEMORY MAP**

FIGURE 3-8: **dsPIC30F3012/3013 DATA SPACE MEMORY MAP**

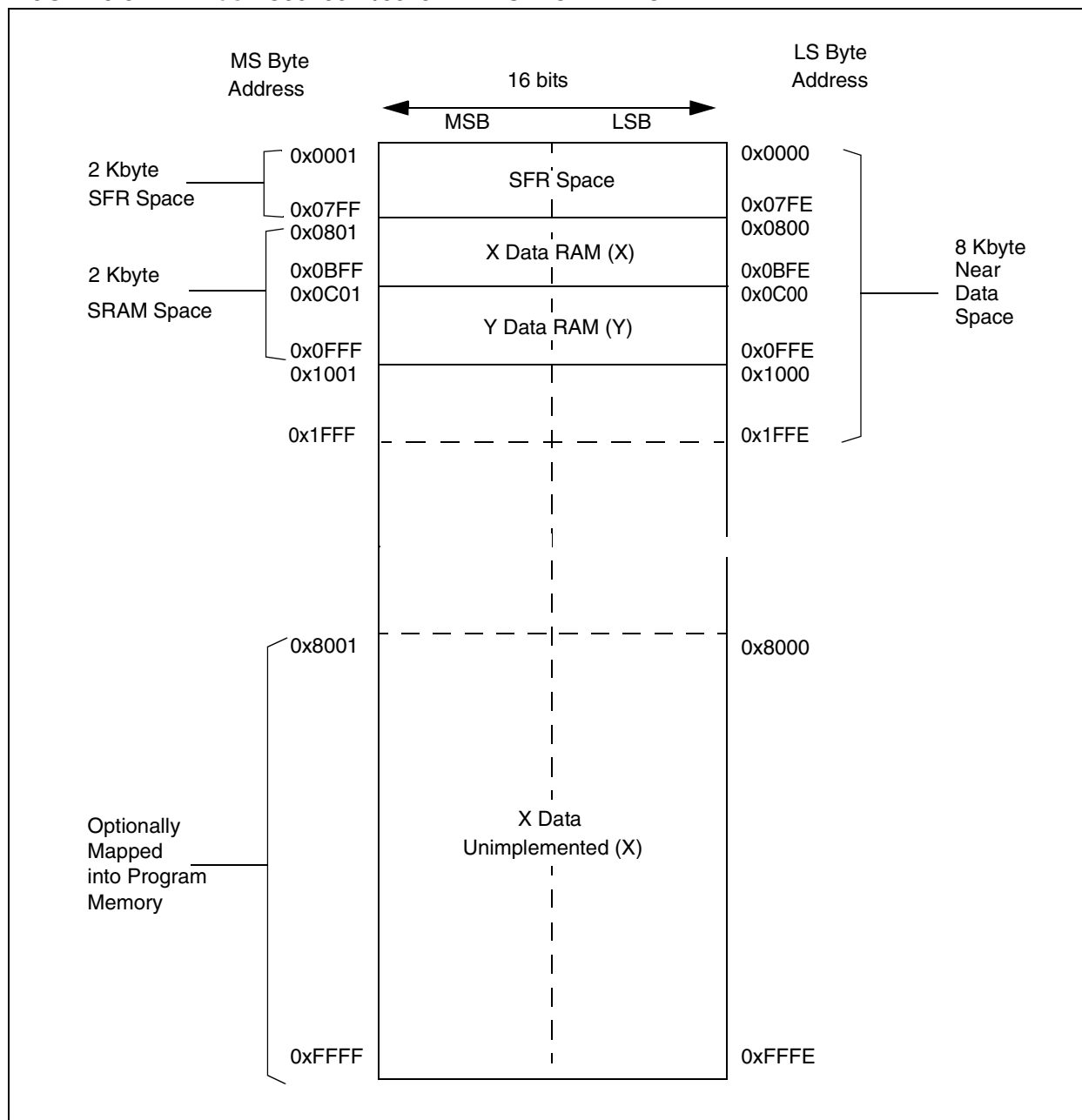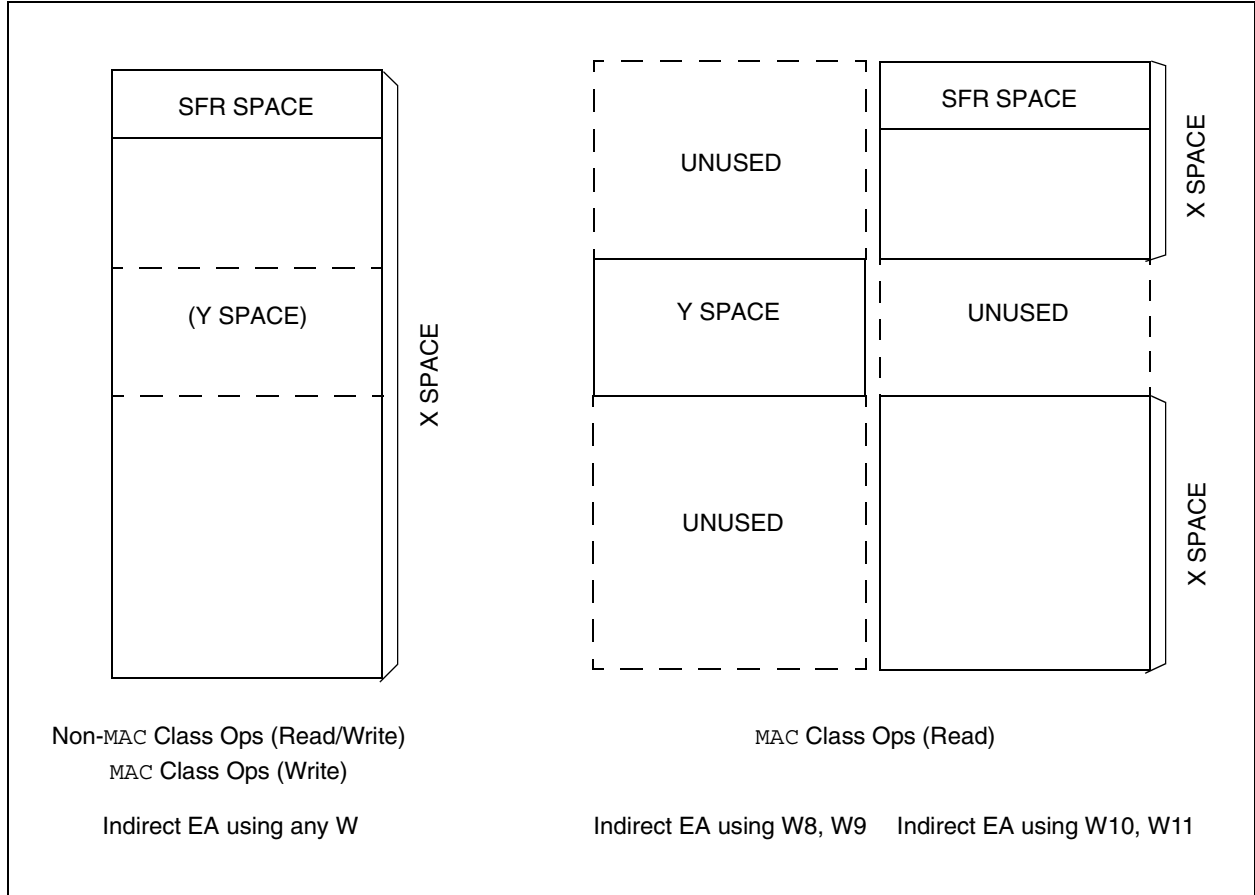**FIGURE 3-9:** **DATA SPACE FOR MCU AND DSP (MAC CLASS) INSTRUCTIONS EXAMPLE**

Non-MAC Class Ops (Read/Write)
MAC Class Ops (Write)

Indirect EA using any W

MAC Class Ops (Read)

Indirect EA using W8, W9   Indirect EA using W10, W11

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 3-8 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 INVALID MEMORY 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 PICmicro® 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 LS bit of any EA to determine which byte to select. The selected byte is placed onto the LS Byte 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. Misaligned word data fetches are not supported so care must be taken when mixing byte and word operations, or translating from 8-bit MCU code. 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 3-10: DATA ALIGNMENT

	MS Byte	LS Byte	
	15 8	7 0	
0001	Byte1	Byte 0	0000
0003	Byte3	Byte 2	0002
0005	Byte5	Byte 4	0004
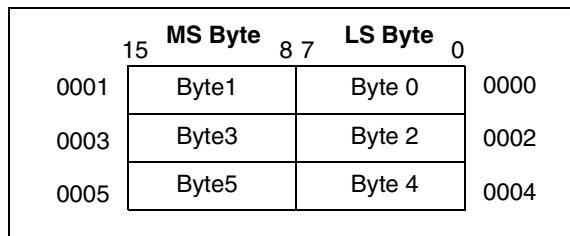
All byte loads into any W register are loaded into the LS Byte. The MSB is not modified.

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 devices contain 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 3-11. 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.

FIGURE 3-11: CALL **STACK FRAME**

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.
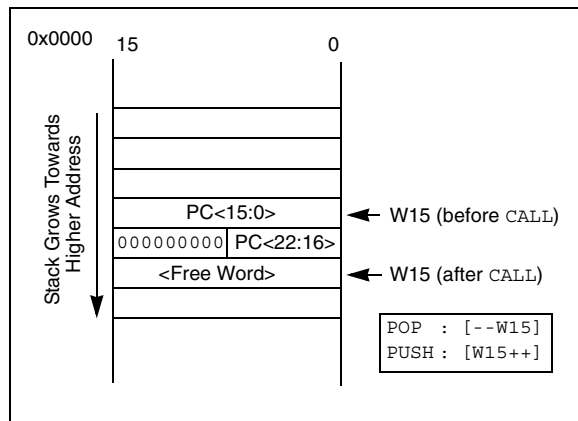
TABLE 3-3: CORE REGISTER MAP

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	Bit 0	Reset State
W0	0000	W0/WREG																0000 0000 0000 0000
W1	0002	W1																0000 0000 0000 0000
W2	0004	W2																0000 0000 0000 0000
W3	0006	W3																0000 0000 0000 0000
W4	0008	W4																0000 0000 0000 0000
W5	000A	W5																0000 0000 0000 0000
W6	000C	W6																0000 0000 0000 0000
W7	000E	W7																0000 0000 0000 0000
W8	0010	W8																0000 0000 0000 0000
W9	0012	W9																0000 0000 0000 0000
W10	0014	W10																0000 0000 0000 0000
W11	0016	W11																0000 0000 0000 0000
W12	0018	W12																0000 0000 0000 0000
W13	001A	W13																0000 0000 0000 0000
W14	001C	W14																0000 1000 0000 0000
W15	001E	W15																0000 0000 0000 0000
SPLIM	0020	SPLIM																0000 0000 0000 0000
ACCAL	0022	ACCAL																0000 0000 0000 0000
ACCAH	0024	ACCAH																0000 0000 0000 0000
ACCAU	0026	Sign-Extension (ACCA<39>)												ACCAU				0000 0000 0000 0000
ACCBL	0028	ACCBL																0000 0000 0000 0000
ACCBH	002A	ACCBH																0000 0000 0000 0000
ACCBU	002C	Sign-Extension (ACCB<39>)												ACCBU				0000 0000 0000 0000
PCL	002E	PCL																0000 0000 0000 0000
PCH	0030	—	—	—	—	—	—	—	—	PCH								0000 0000 0000 0000
TBLPAG	0032	—	—	—	—	—	—	—	—	TBLPAG								0000 0000 0000 0000
PSVPAG	0034	—	—	—	—	—	—	—	—	PSVPAG								0000 0000 0000 0000
RCOUNT	0036	RCOUNT																uuuu uuuu uuuu uuuu
DCOUNT	0038	DCOUNT																uuuu uuuu uuuu uuuu
DOSTARTL	003A	DOSTARTL															0	uuuu uuuu uuuu uuu0
DOSTARTH	003C	—	—	—	—	—	—	—	—	DOSTARTH								0000 0000 0uuu uuuu
DOENDL	003E	DOENDL															0	uuuu uuuu uuuu uuu0
DOENDH	0040	—	—	—	—	—	—	—	—	DOENDH								0000 0000 0uuu uuuu
SR	0042	OA	OB	SA	SB	OAB	SAB	DA	DC	IPL2	IPL1	IPL0	RA	N	OV	Z	C	0000 0000 0000 0000

Legend: u = uninitialized bit

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	Bit 0	Reset State
CORCON	0044	—	—	—	US	EDT	DL2	DL1	DL0	SATA	SATB	SATDW	ACCSAT	IPL3	PSV	RND	IF	0000 0000 0010 0000
MODCON	0046	XMODEN	YMODEN	—	—	BWM<3:0>				YWM<3:0>				XWM<3:0>				0000 0000 0000 0000
XMODSRT	0048	XS<15:1>															0	uuuu uuuu uuuu uuu0
XMODEND	004A	XE<15:1>															1	uuuu uuuu uuuu uuu1
YMODSRT	004C	YS<15:1>															0	uuuu uuuu uuuu uuu0
YMODEND	004E	YE<15:1>															1	uuuu uuuu uuuu uuu1
XBREV	0050	BREN	XB<14:0>															uuuu uuuu uuuu uuuu
DISICNT	0052	—	—	DISICNT<13:0>														0000 0000 0000 0000

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 Programmer's Reference Manual* (DS70030).

The dsPIC 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 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 2 source operand pre-fetch 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 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 ending 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 calculations assume word sized data (LS bit 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).
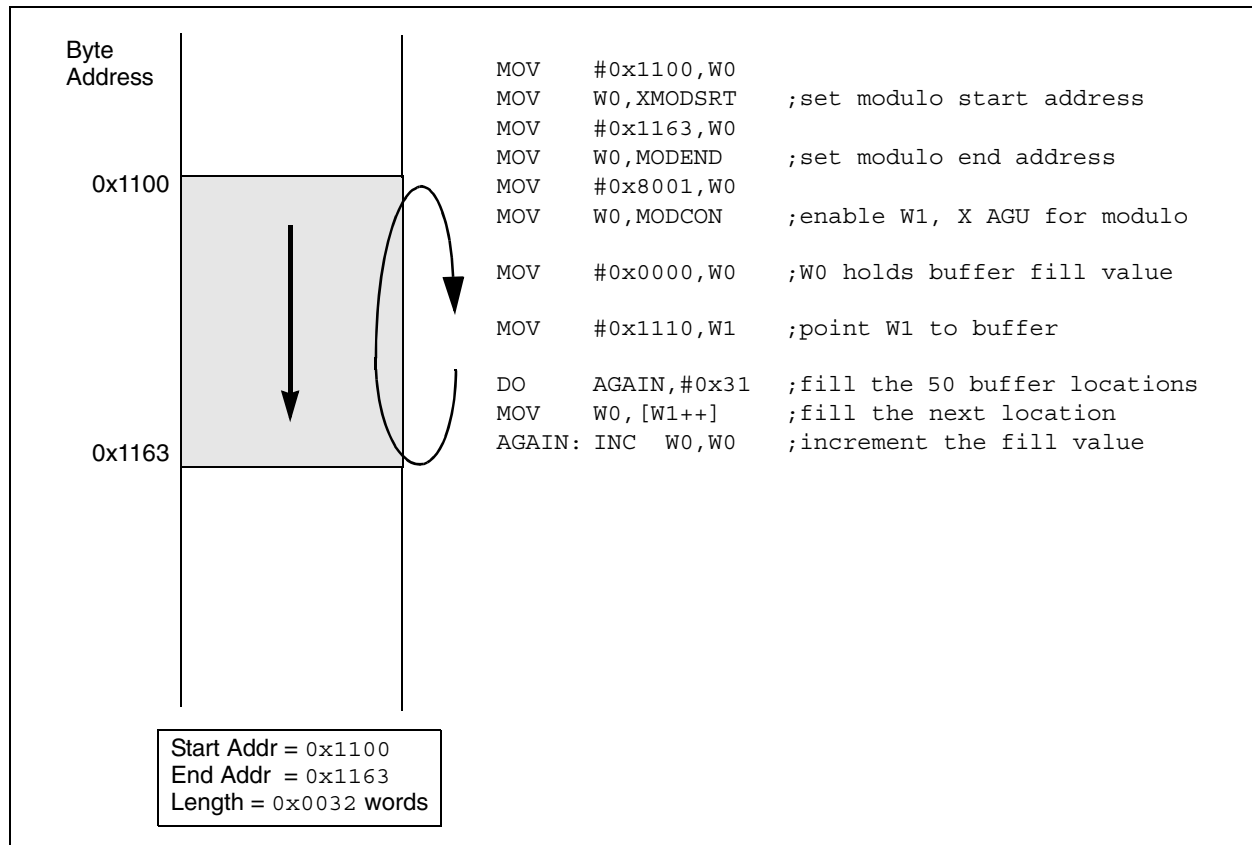
4.2.2 W ADDRESS REGISTER SELECTION

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 is disabled. Similarly, if YWM = 15, Y AGU modulo addressing is disabled.

The X Address Space Pointer W register (XWM), to which modulo addressing is to be applied, is stored in MODCON<3:0> (see Table 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>.

FIGURE 4-1: **MODULO ADDRESSING OPERATION EXAMPLE**


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

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

                              MOV    #0x1110,W1    ;point W1 to buffer

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

Byte Address

0x1100

0x1163

Start Addr = 0x1100
End Addr = 0x1163
Length = 0x0032 words

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 remain 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:

1. BWM (W register selection) in the MODCON register is any value other than '15' (the stack cannot 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 (LS bit 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 LS bit 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 addressing will assume priority when active for the X WAGU, and X WAGU modulo addressing will be disabled. However, modulo addressing will continue to function in the X RAGU.

If bit-reversed addressing has already been enabled by setting the BREN (XBREV<15>) bit, then a write to the XBREV register should not be immediately followed by an indirect read operation using the W register that has been designated as the bit-reversed pointer.
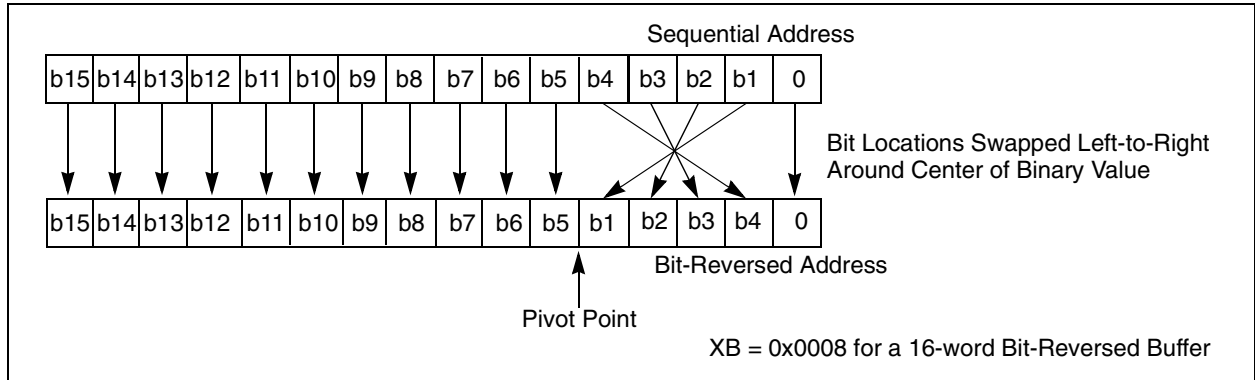
FIGURE 4-2: **BIT-REVERSED ADDRESS EXAMPLE**

Sequential Address

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | 0 |

Bit Locations Swapped Left-to-Right
Around Center of Binary Value

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b1 | b2 | b3 | b4 | 0 |

Bit-Reversed Address

Pivot Point

XB = 0x0008 for a 16-word Bit-Reversed Buffer

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

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

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

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

NOTES:

5.0 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 Programmer's Reference Manual* (DS70030).

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

5.1 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.

5.2 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.

5.3 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 5-1.

FIGURE 5-1: ADDRESSING FOR TABLE AND NVM REGISTERS
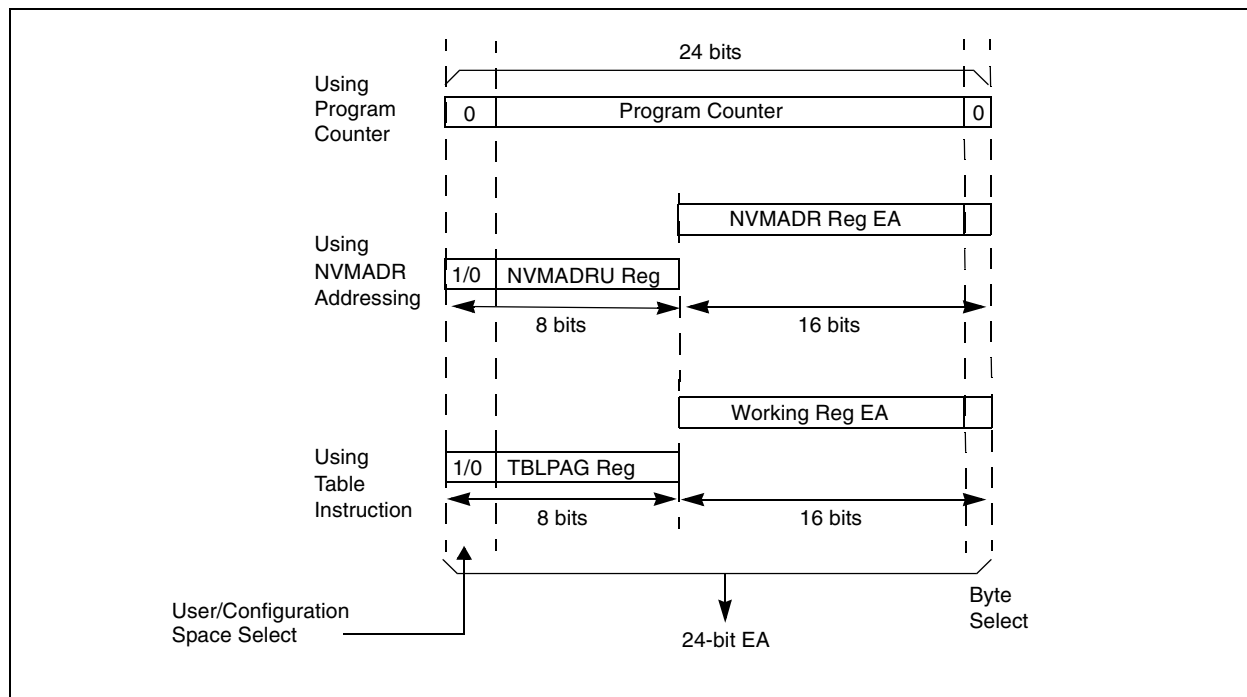
5.4   RTSP Operation

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 x 24 instructions. RTSP allows the user to erase one row (32 instructions) at a time and to program four instructions at one time. RTSP may be used to program multiple program memory panels, but the table pointer must be changed at each panel boundary.

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 instruction words loaded must always be from a 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 four TBLWTH instructions are required to load the 32 instructions. If multiple panel programming is required, the table pointer needs to be changed and the next set of multiple write latches written.

All of the table write operations are single word writes (2 instruction cycles), because only the table latches are written. A programming cycle is required for programming each row.

The Flash Program Memory is readable, writable and erasable during normal operation over the entire V$_{DD}$ range.

5.5   Control Registers

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

- NVMCON
- NVMADR
- NVMADRU
- NVMKEY

5.5.1   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.

5.5.2   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.

5.5.3   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.

5.5.4   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 5.6 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.

5.6   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.

5.6.1   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.
    a) 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.

5.6.2   ERASING A ROW OF PROGRAM MEMORY

Example 5-1 shows a code sequence that can be used to erase a row (32 instructions) of program memory.

EXAMPLE 5-1:   ERASING A ROW OF PROGRAM MEMORY

```
; Setup NVMCON for erase operation, multi word write
; program memory selected, and writes enabled
        MOV     #0x4041,W0              ;
        MOV     W0,NVMCON               ; Init NVMCON SFR
; Init pointer to row to be ERASED
        MOV     #tblpage(PROG_ADDR),W0  ;
        MOV     W0,NVMADRU              ; Initialize PM Page Boundary SFR
        MOV     #tbloffset(PROG_ADDR),W0 ; Intialize in-page EA[15:0] pointer
        MOV     W0, NVMADR              ; Initialize 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     #0xAA,W1                ;
        MOV     W1,NVMKEY               ; Write the 0xAA key
        BSET    NVMCON,#WR              ; Start the erase sequence
        NOP                             ; Insert two NOPs after the erase
        NOP                             ; command is asserted
```

5.6.3   LOADING WRITE LATCHES

Example 5-2 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.

5.6.4   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 `NOP`s.

EXAMPLE 5-2:   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
      MOV     #LOW_WORD_0,W2                 ;
      MOV     #HIGH_BYTE_0,W3                ;
      TBLWTL  W2,[W0]                        ; Write PM low word into program latch
      TBLWTH  W3,[W0++]                      ; Write PM high byte into program latch
; 1st_program_word
      MOV     #LOW_WORD_1,W2                 ;
      MOV     #HIGH_BYTE_1,W3                ;
      TBLWTL  W2,[W0]                        ; Write PM low word into program latch
      TBLWTH  W3,[W0++]                      ; Write PM high byte into program latch
;  2nd_program_word
      MOV     #LOW_WORD_2,W2                 ;
      MOV     #HIGH_BYTE_2,W3                ;
      TBLWTL  W2, [W0]                       ; Write PM low word into program latch
      TBLWTH  W3, [W0++]                     ; Write PM high byte into program latch
      •
      •
      •
; 31st_program_word
      MOV     #LOW_WORD_31,W2                ;
      MOV     #HIGH_BYTE_31,W3               ;
      TBLWTL  W2, [W0]                       ; Write PM low word into program latch
      TBLWTH  W3, [W0++]                     ; Write PM high byte into program latch
```

Note: In Example 5-2, the contents of the upper byte of W3 has no effect.

EXAMPLE 5-3:   INITIATING A PROGRAMMING SEQUENCE

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

TABLE 5-1: NVM REGISTER MAP

File Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All RESETS
NVMCON	0760	WR	WREN	WRERR	—	—	—	—	TWRI	—				PROGOP<6:0>				0000 0000 0000 0000
NVMADR	0762										NVMADR<15:0>							uuuu uuuu uuuu uuuu
NVMADRU	0764	—	—	—	—	—	—	—	—				NVMADR<23:16>					0000 0000 uuuu uuuu
NVMKEY	0766	—	—	—	—	—	—	—	—				KEY<7:0>					0000 0000 0000 0000

Legend: u = uninitialized bit

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

NOTES:

6.0   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 Programmer's Reference Manual* (DS70030).

The Data EEPROM Memory is readable and writable during normal operation over the entire V_{DD} 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 6.5, 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 are used to address the EEPROM location being accessed. `TBLRDL` and `TBLWTL` instructions are used to read and write data EEPROM. The dsPIC30F devices have up to 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. They are 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 $\overline{MCLR}$ 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.

6.1   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 6-1.

EXAMPLE 6-1:   DATA EEPROM READ

```
MOV     #LOW_ADDR_WORD,W0  ; Init Pointer
MOV     #HIGH_ADDR_WORD,W1
MOV     W1,TBLPAG
TBLRDL  [ W0 ], W4         ; read data EEPROM
```

6.2 Erasing Data EEPROM

6.2.1 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 the NVMCON register. Setting the WR bit initiates the erase as shown in Example 6-2.

EXAMPLE 6-2:   DATA EEPROM BLOCK ERASE

```
; Select data EEPROM block, ERASE, WREN bits
   MOV    #0x4045,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              ;
   MOV    W0,NVMKEY             ; Write the 0x55 key
   MOV    #0xAA,W1              ;
   MOV    W1,NVMKEY             ; Write the 0xAA 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
```

6.2.2 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 the NVMCON register. Setting the WR bit initiates the erase as shown in Example 6-3.

EXAMPLE 6-3:   DATA EEPROM WORD ERASE

```
; Select data EEPROM word, ERASE, WREN bits
   MOV    #0x4044,W0
   MOV    W0,NVMCON

; 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              ;
   MOV    W0,NVMKEY             ; Write the 0x55 key
   MOV    #0xAA,W1              ;
   MOV    W1,NVMKEY             ; Write the 0xAA 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
```

6.3   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 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.

6.3.1   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 6-4.

6.3.2   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.

EXAMPLE 6-4:   DATA EEPROM WORD WRITE

```
; Point to data memory
    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 captures last table access address
; Select data EEPROM for 1 word op
    MOV        #0x4004,W0
    MOV        W0,NVMCON

; Operate key to allow write 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                      ; Initiate program sequence
    NOP
    NOP
; Write cycle will complete in 2mS. CPU is not stalled for the Data Write Cycle
; User can poll WR bit, use NVMIF or Timer IRQ to determine write complete
```

dsPIC30F2011/2012/3012/3013

EXAMPLE 6-5: **DATA EEPROM BLOCK WRITE**

```
        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
```

6.4   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.

6.5   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.

7.0 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.

7.1 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.

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 7-1 shows how ports are shared with other peripherals and the associated I/O cell (pad) to which they are connected.

The format of the registers for the shared ports, (PORTB, PORTC, PORTD and PORTF) are shown in Table 7-1 through Table 7-6.

> **Note:** The actual bits in use vary between devices.

FIGURE 7-1: BLOCK DIAGRAM OF A SHARED PORT STRUCTURE
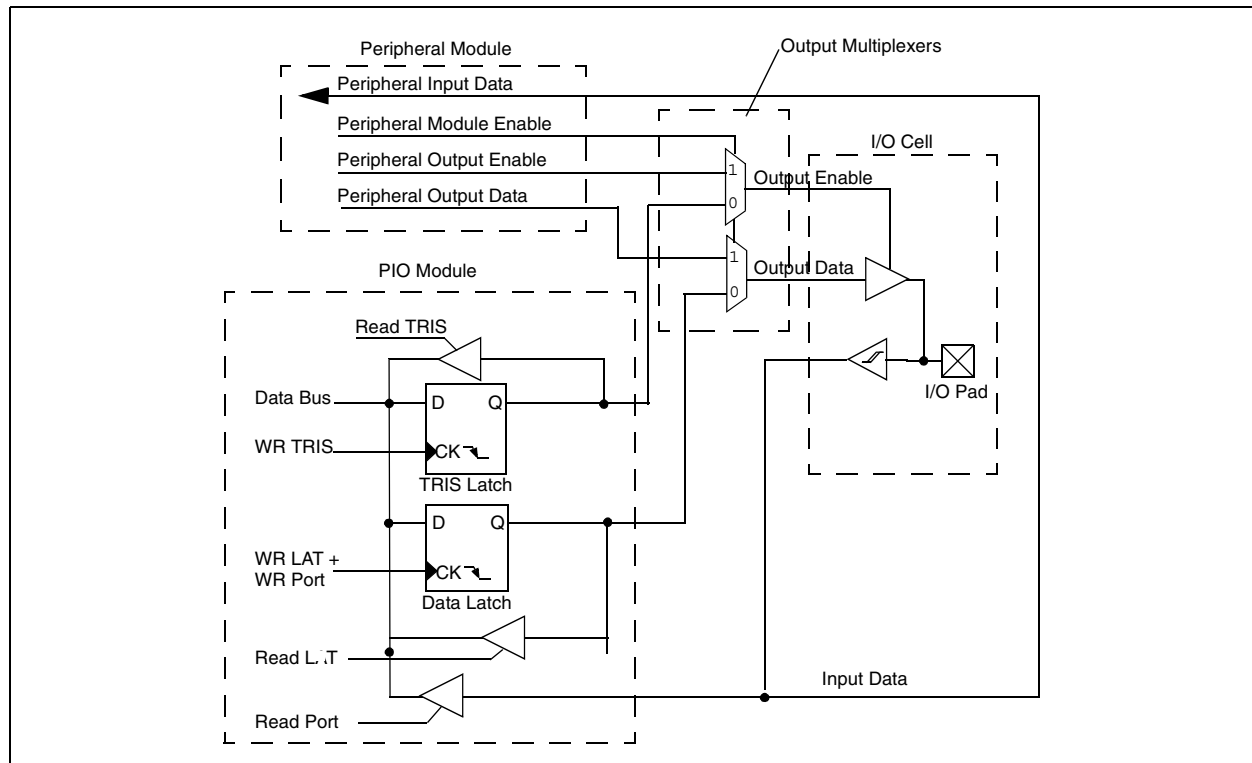
7.2 Configuring Analog Port Pins

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 (V_{OH} or V_{OL}) will be converted.

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

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

7.2.1 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 7-1: PORT WRITE/READ EXAMPLE

```
MOV  #0xF0, W0; Configure PORTB<7:4>
             ; as inputs
MOV  W0, TRISB; and PORTB<3:0> as outputs
NOP          ; additional instruction cycle
btss PORTB, #7; bit test RB7 and skip if set
```

TABLE 7-1: **PORTB REGISTER MAP FOR dsPIC30F2011/3012**

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
TRISB	02C6	—	—	—	—	—	—	—	—	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	0000 0000 1111 1111
PORTB	02C8	—	—	—	—	—	—	—	—	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	0000 0000 0000 0000
LATB	02CB	—	—	—	—	—	—	—	—	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	0000 0000 0000 0000

TABLE 7-2: **PORTB REGISTER MAP FOR dsPIC30F2012/3013**

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
TRISB	02C6	—	—	—	—	—	—	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	0000 0011 1111 1111
PORTB	02C8	—	—	—	—	—	—	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	0000 0000 0000 0000
LATB	02CB	—	—	—	—	—	—	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	0000 0000 0000 0000

TABLE 7-3: **PORTC REGISTER MAP FOR dsPIC30F2011/2012/3012/3013**

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
TRISC	02CC	TRISC15	TRISC14	TRISC13	—	—	—	—	—	—	—	—	—	—	—	—	—	1110 0000 0000 0000
PORTC	02CE	RC15	RC14	RC13	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
LATC	02D0	LATC15	LATC14	LATC13	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000

TABLE 7-4: **PORTD REGISTER MAP FOR dsPIC30F2011/3012**

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
TRISD	02D2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRISD0	0000 0000 0000 0001
PORTD	02D4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RD0	0000 0000 0000 0000
LATD	02D6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LATD0	0000 0000 0000 0000

TABLE 7-5: PORTD REGISTER MAP FOR dsPIC30F2012/3013

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
TRISD	02D2	—	—	—	—	—	—	TRISD9	TRISD8	—	—	—	—	—	—	—	—	0000 0011 0000 0000
PORTD	02D4	—	—	—	—	—	—	RD9	RD8	—	—	—	—	—	—	—	—	0000 0000 0000 0000
LATD	02D6	—	—	—	—	—	—	LATD9	LATD8	—	—	—	—	—	—	—	—	0000 0000 0000 0000

TABLE 7-6: PORTF REGISTER MAP FOR dsPIC30F2012/3013

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
TRISF	02DE	—	—	—	—	—	—	—	—	—	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	—	—	0000 0000 0111 1100
PORTF	02E0	—	—	—	—	—	—	—	—	—	RF6	RF5	RF4	RF3	RF2	—	—	0000 0000 0000 0000
LATF	02E2	—	—	—	—	—	—	—	—	—	LATF6	LATF5	LATF4	LATF3	LATF2	—	—	0000 0000 0000 0000

Note: The dsPIC30F2011/3012 do not have TRISF, PORTF or LATF.

7.3 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 up to 10 external signals (CN0 through CN7, CN17 and CN18) that may be selected (enabled) for generating an interrupt request on a change of state.

TABLE 7-7: INPUT CHANGE NOTIFICATION REGISTER MAP FOR dsPIC30F2011/3012 (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	—	—	—	—	—	—	—	—	0000 0000 0000 0000
CNPU1	00C4	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000 0000 0000 0000
CNPU2	00C6	—	—	—	—	—	—	—	—	0000 0000 0000 0000

TABLE 7-8: INPUT CHANGE NOTIFICATION REGISTER MAP FOR dsPIC30F2012/3013 (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	—	—	—	—	—	CN18IE	CN17IE	—	0000 0000 0000 0000
CNPU1	00C4	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000 0000 0000 0000
CNPU2	00C6	—	—	—	—	—	CN18PUE	CN17PUE	—	0000 0000 0000 0000

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

NOTES:

8.0   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 Programmer's Reference Manual* (DS70030).

The dsPIC30F Sensor Family has up to 21 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 8-1.

The interrupt controller is responsible for pre-processing the interrupts and processor exceptions before they are 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>... IPC10<7:0>
  The user assignable priority level associated with each of these 41 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 7 priority levels, 1 through 7, via the IPCx registers. Each interrupt source is associated with an interrupt vector, as shown in Table 8-1. Levels 7 and 1 represent the highest and lowest maskable priorities, respectively.

> **Note:** Assigning a priority level of '0' to an interrupt 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 Table 8-1). These vectors are contained in locations 0x000004 through 0x0000FE of program memory (refer to Table 8-1). 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.

dsPIC30F2011/2012/3012/3013

8.1 Interrupt Priority

The user assignable interrupt priority (IP<2:0>) bits for each individual interrupt source are located in the LS 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.

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 8-1 lists the interrupt numbers and interrupt sources for the dsPIC30F2011/2012/3012/3013 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 INT0 (External Interrupt 0) may be assigned to priority level 1, thus giving it a very low effective priority.

TABLE 8-1: INTERRUPT VECTOR TABLE

INT Number	Vector Number	Interrupt Source
Highest Natural Order 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^2C Slave Interrupt
14	22	MI2C - I^2C Master Interrupt
15	23	Input Change Interrupt
16	24	INT1 - External Interrupt 1
17-22	25-30	Reserved
23	31	INT2 - External Interrupt 2
24	32	*U2RX - UART2 Receiver
25	33	*U2TX - UART2 Transmitter
26-41	34-49	Reserved
42	50	LVD - Low Voltage Detect
43-53	51-61	Reserved
Lowest Natural Order Priority		

* Only the dsPIC30F3013 has UART2 and the U2RX, U2TX interrupts. These locations are reserved for the dsPIC30F2011/2012/3012.

Preliminary

8.2   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.

8.2.1   RESET SOURCES

In addition to external Reset and Power-on Reset (POR), there are 6 sources of error conditions which 'trap' to the Reset vector.

- 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.

8.3   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 8-1. 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.

8.3.1   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 three 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.
6. 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.

8.3.2 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 8-2 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 8-1: TRAP VECTORS

8.4 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 8-2. 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 8-2: INTERRUPT STACK FRAME
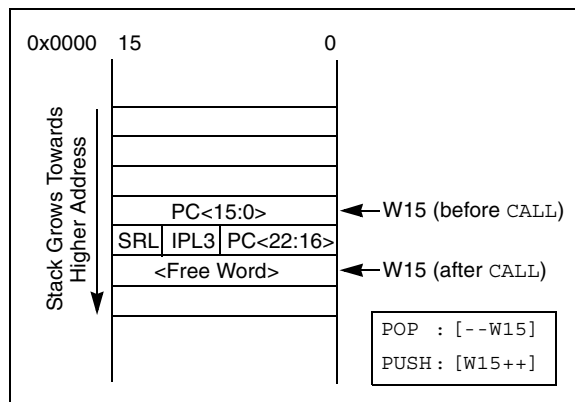
> **Note 1:** The user can always lower the priority level by writing a new value into SR. The Interrupt Service Routine must clear the interrupt flag bits in the IFSx register before lowering the processor interrupt priority, in order to avoid recursive interrupts.
>
> **2:** The IPL3 bit (CORCON<3>) 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.

8.5 Alternate Vector Table

In program memory, the Interrupt Vector Table (IVT) is followed by the Alternate Interrupt Vector Table (AIVT), as shown in Figure 8-1. 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.

8.6   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.

8.7   External Interrupt Requests

The interrupt controller supports three external interrupt request signals, INT0-INT2. 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 three bits, INT0EP-INT2EP, that select the polarity of the edge detection circuitry.

8.8   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 8-2:   dsPIC30F2011/2012/3012 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	Bit 0	Reset State
INTCON1	0080	NSTDIS	—	—	—	—	OVATE	OVBTE	COVTE	—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000 0000 0000 0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000 0000 0000 0000
IFS0	0084	CNIF	MI2CIF	SI2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF	T3IF	T2IF	OC2IF	IC2IF	T1IF	OC1IF	IC1IF	INT0IF	0000 0000 0000 0000
IFS1	0086	—	—	—	—	—	—	—	—	INT2IF	—	—	—	—	—	—	INT1IF	0000 0000 0000 0000
IFS2	0088	—	—	—	—	—	LVDIF	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IEC0	008C	CNIE	MI2CIE	SI2CIE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE	T3IE	T2IE	OC2IE	IC2IE	T1IE	OC1IE	IC1IE	INT0IE	0000 0000 0000 0000
IEC1	008E	—	—	—	—	—	—	—	—	INT2IE	—	—	—	—	—	—	INT1IE	0000 0000 0000 0000
IEC2	0090	—	—	—	—	—	LVDIE	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC0	0094	—	T1IP<2:0>			—	OC1IP<2:0>			—	IC1IP<2:0>			—	INT0IP<2:0>			—	0100 0100 0100 0100
IPC1	0096	—	T3IP<2:0>			—	T2IP<2:0>			—	OC2IP<2:0>			—	IC2IP<2:0>			—	0100 0100 0100 0100
IPC2	0098	—	ADIP<2:0>			—	U1TXIP<2:0>			—	U1RXIP<2:0>			—	SPI1IP<2:0>			—	0100 0100 0100 0100
IPC3	009A	—	CNIP<2:0>			—	MI2CIP<2:0>			—	SI2CIP<2:0>			—	NVMIP<2:0>			—	0100 0100 0100 0100
IPC4	009C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC5	009E	—	INT2IP<2:0>			—	—	—	—	—	—	—	—	—	INT1IP<2:0>			—	0100 0000 0000 0100
IPC6	00A0	—	—	—	—	—	—	—	—	—	1	0	0	—	1	0	0	0000 0000 0100 0100
IPC7	00A2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC8	00A4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC9	00A6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC10	00A8	—	—	—	—	—	—	LVDIP<2:0>			—	—	—	—	—	—	—	0000 0100 0000 0000

Legend: u = uninitialized bit

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

TABLE 8-3: dsPIC30F3013 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	Bit 0	Reset State
INTCON1	0080	NSTDIS	—	—	—	—	OVATE	OVBTE	COVTE	—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000 0000 0000 0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000 0000 0000 0000
IFS0	0084	CNIF	MI2CIF	SI2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF	T3IF	T2IF	OC2IF	IC2IF	T1IF	OC1IF	IC1IF	INT0IF	0000 0000 0000 0000
IFS1	0086	—	—	—	—	—	—	U2TXIF	U2RXIF	INT2IF	—	—	—	—	—	—	INT1IF	0000 0000 0000 0000
IFS2	0088	—	—	—	—	—	LVDIF	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IEC0	008C	CNIE	MI2CIE	SI2CIE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE	T3IE	T2IE	OC2IE	IC2IE	T1IE	OC1IE	IC1IE	INT0IE	0000 0000 0000 0000
IEC1	008E	—	—	—	—	—	—	U2TXIE	U2RXIE	INT2IE	—	—	—	—	—	—	INT1IE	0000 0000 0000 0000
IEC2	0090	—	—	—	—	—	LVDIE	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC0	0094	—	—	T1IP<2:0>			—	OC1IP<2:0>			—	IC1IP<2:0>			—	INT0IP<2:0>			0100 0100 0100 0100
IPC1	0096	—	—	T3IP<2:0>			—	T2IP<2:0>			—	OC2IP<2:0>			—	IC2IP<2:0>			0100 0100 0100 0100
IPC2	0098	—	—	ADIP<2:0>			—	U1TXIP<2:0>			—	U1RXIP<2:0>			—	SPI1IP<2:0>			0100 0100 0100 0100
IPC3	009A	—	—	CNIP<2:0>			—	MI2CIP<2:0>			—	SI2CIP<2:0>			—	NVMIP<2:0>			0100 0100 0100 0100
IPC4	009C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INT1IP<2:0>			0000 0000 0000 0100
IPC5	009E	—	—	INT2IP<2:0>			—	—	—	—	—	—	—	—	—	—	—	0100 0000 0000 0000
IPC6	00A0	—	—	—	—	—	—	U2TXIP<2:0>			—	U2RXIP<2:0>			—	—	—	—	0000 0000 0100 0100
IPC7	00A2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC8	00A4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC9	00A6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC10	00A8	—	—	—	—	—	—	LVDIP<2:0>			—	—	—	—	—	—	—	0000 0100 0000 0000

Legend: u = uninitialized bit

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

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. Figure 9-1 depicts the simplified block diagram of the 16-bit Timer1 module. The following sections provide detailed descriptions 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 that serves as the time counter for the real-time clock or operates 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

These 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 stops incrementing unless the TSIDL (T1CON<13>) bit = 0. If TSIDL = 1, the timer module logic resumes the incrementing sequence on termination of 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.
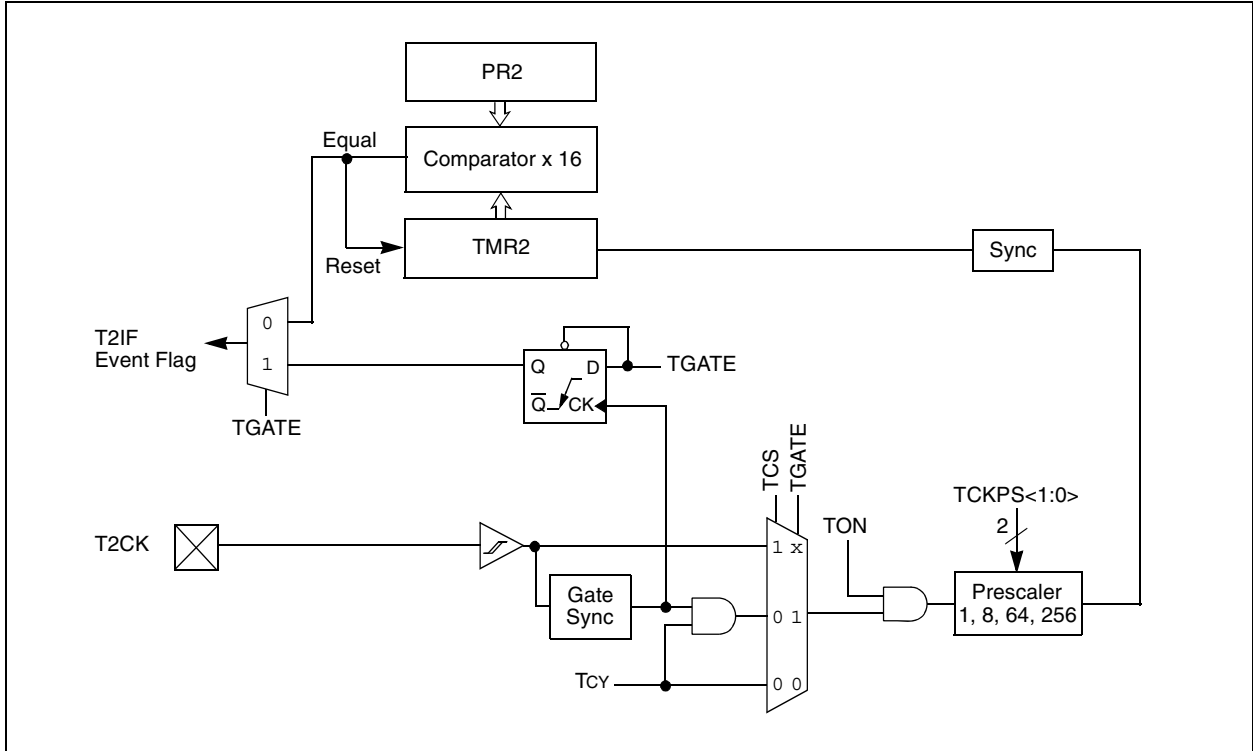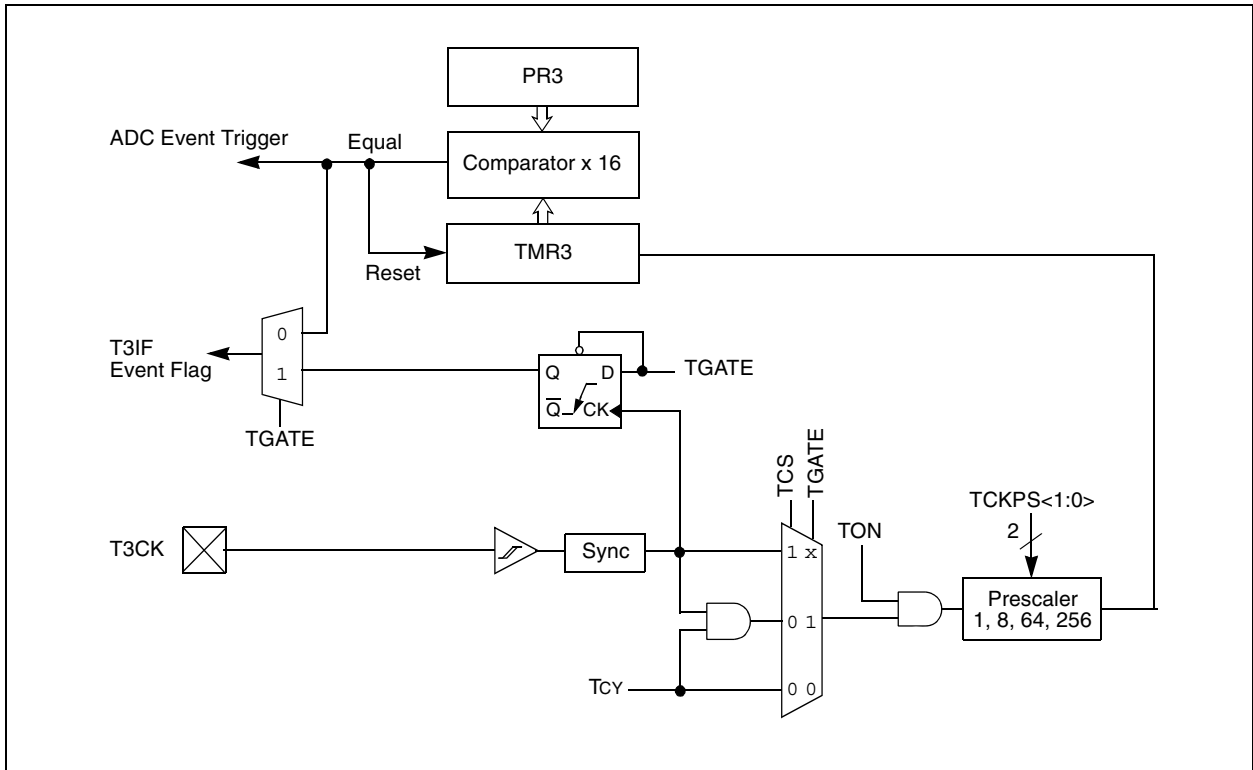
FIGURE 9-1:   **16-BIT TIMER1 MODULE BLOCK DIAGRAM**
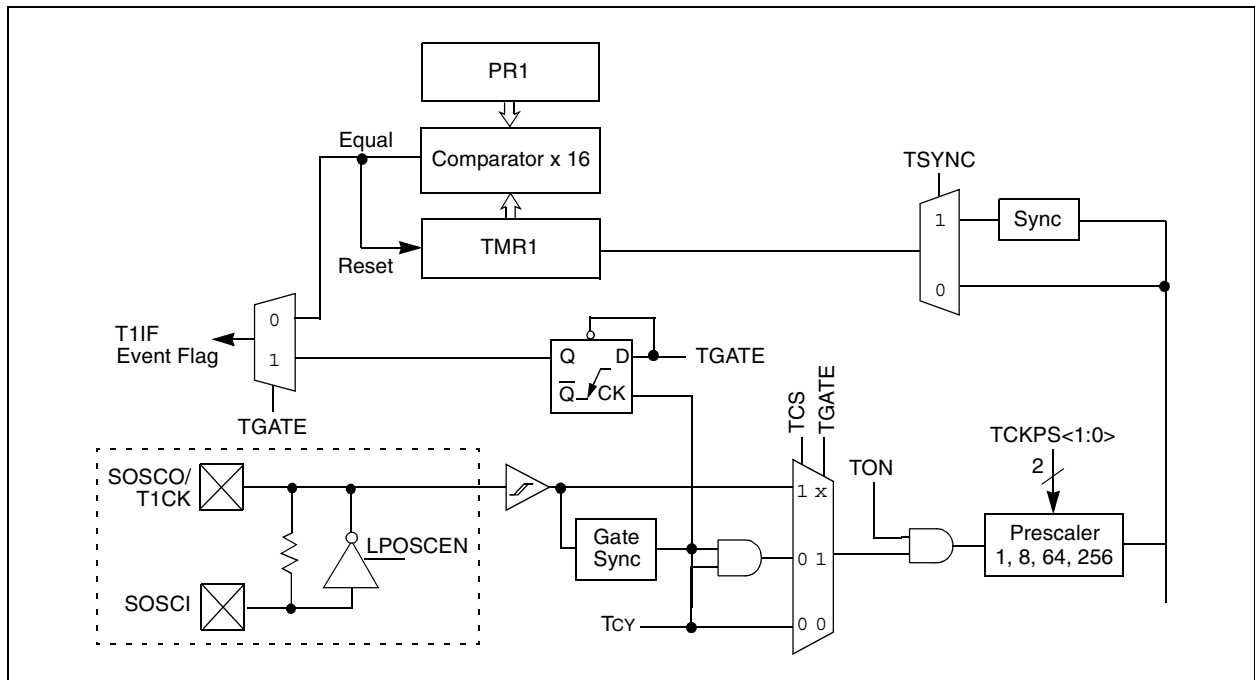
dsPIC30F2011/2012/3012/3013

9.1 Timer Gate Operation

The 16-bit timer can be placed in the Gated Time Accumulation mode. This mode allows the internal T$_{CY}$ 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 (F$_{OSC}$/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
• a write to the T1CON register
• 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

C1 = C2 = 18 pF; R = 100K
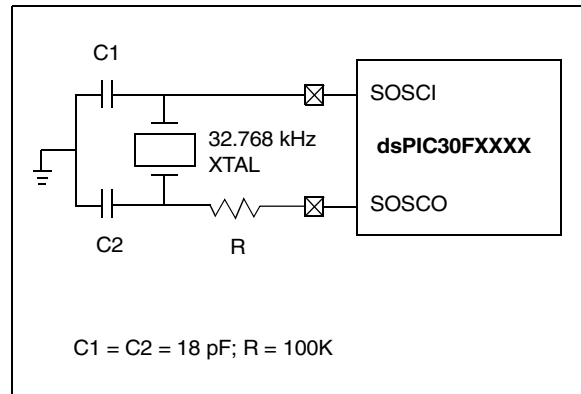
9.5.1 RTC OSCILLATOR OPERATION

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	Bit 0	Reset State
TMR1	0100	Timer1 Register																uuuu uuuu uuuu uuuu
PR1	0102	Period Register 1																1111 1111 1111 1111
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000 0000 0000 0000

Legend: u = uninitialized bit

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

Preliminary

© 2005 Microchip Technology Inc.

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 (GP) 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.

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 LS Word and Timer3 is the MS 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. Timer2 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 Timer 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 LS Word (TMR2 register) will cause the MS 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**

Note: Timer configuration bit T32 (T2CON<3>) must be set to '1' for a 32-bit timer/counter operation. All control bits are respective to the T2CON register.

FIGURE 10-2: **16-BIT TIMER2 BLOCK DIAGRAM**

FIGURE 10-3: **16-BIT TIMER3 BLOCK DIAGRAM**

10.1 Timer Gate Operation

The 32-bit timer can be placed in the Gated Time Accumulation mode. This mode allows the internal T_{CY} 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), or between the 16-bit timer TMR3 and the 16-bit period register PR3, a special ADC trigger event signal is generated by Timer3.

10.3 Timer Prescaler

The input clock ($F_{OSC}/4$ 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
• a write to the T2CON/T3CON register
• 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	Bit 0	Reset State
TMR2	0106	Timer2 Register																uuuu uuuu uuuu uuuu
TMR3HLD	0108	Timer3 Holding Register (for 32-bit timer operations only)																uuuu uuuu uuuu uuuu
TMR3	010A	Timer3 Register																uuuu uuuu uuuu uuuu
PR2	010C	Period Register 2																1111 1111 1111 1111
PR3	010E	Period Register 3																1111 1111 1111 1111
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000 0000 0000 0000
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000 0000 0000 0000

Legend: u = uninitialized bit

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

NOTES:

Preliminary

11.0   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 11-1 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

Important operational features of the input capture module are its

- 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 IC1CON and IC2CON registers. The dsPIC30F2011/2012/3012/3013 devices have two capture channels.

11.1   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>).

11.1.1   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.

FIGURE 11-1:   INPUT CAPTURE MODE BLOCK DIAGRAM

Note: Where 'x' is shown, reference is made to the registers or bits associated to the respective input capture channel (1 or 2).
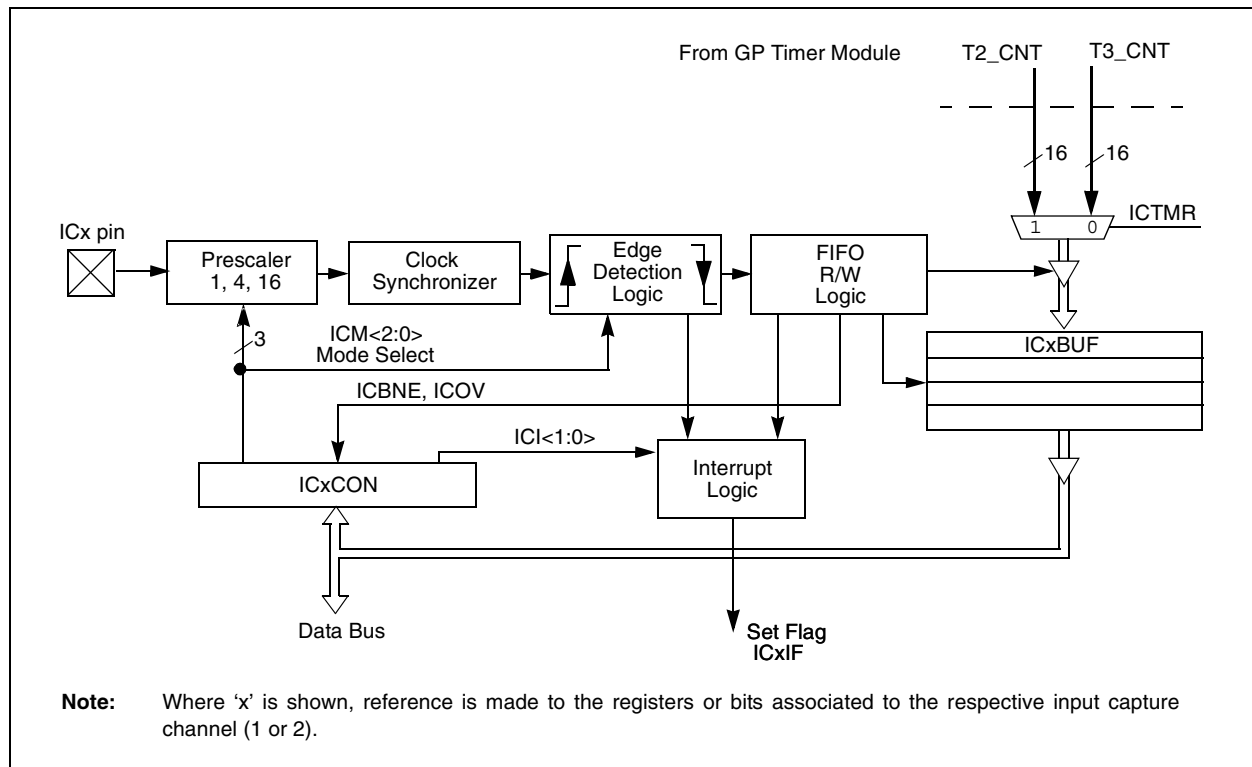
11.1.2 CAPTURE BUFFER OPERATION

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.

11.1.3 TIMER2 AND TIMER3 SELECTION MODE

The input capture module consists of up to 8 input capture channels. Each 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.

11.1.4 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.

11.2 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.

11.2.1 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 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.

11.2.2 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 is 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.

11.3 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 11-1:   INPUT CAPTURE REGISTER MAP

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
IC1BUF	0140	Input 1 Capture Register																uuuu uuuu uuuu uuuu
IC1CON	0142	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000
IC2BUF	0144	Input 2 Capture Register																uuuu uuuu uuuu uuuu
IC2CON	0146	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000

Legend: u = uninitialized bit

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

NOTES:

12.0   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 12-1 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 OC1CON and OC2CON registers. The dsPIC30F2011/2012/3012/3013 devices have 2 compare channels.

OCxRS and OCxR in Figure 12-1 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 12-1:   OUTPUT COMPARE MODE BLOCK DIAGRAM

Note: Where 'x' is shown, reference is made to the registers associated with the respective output compare channel (1 or 2).
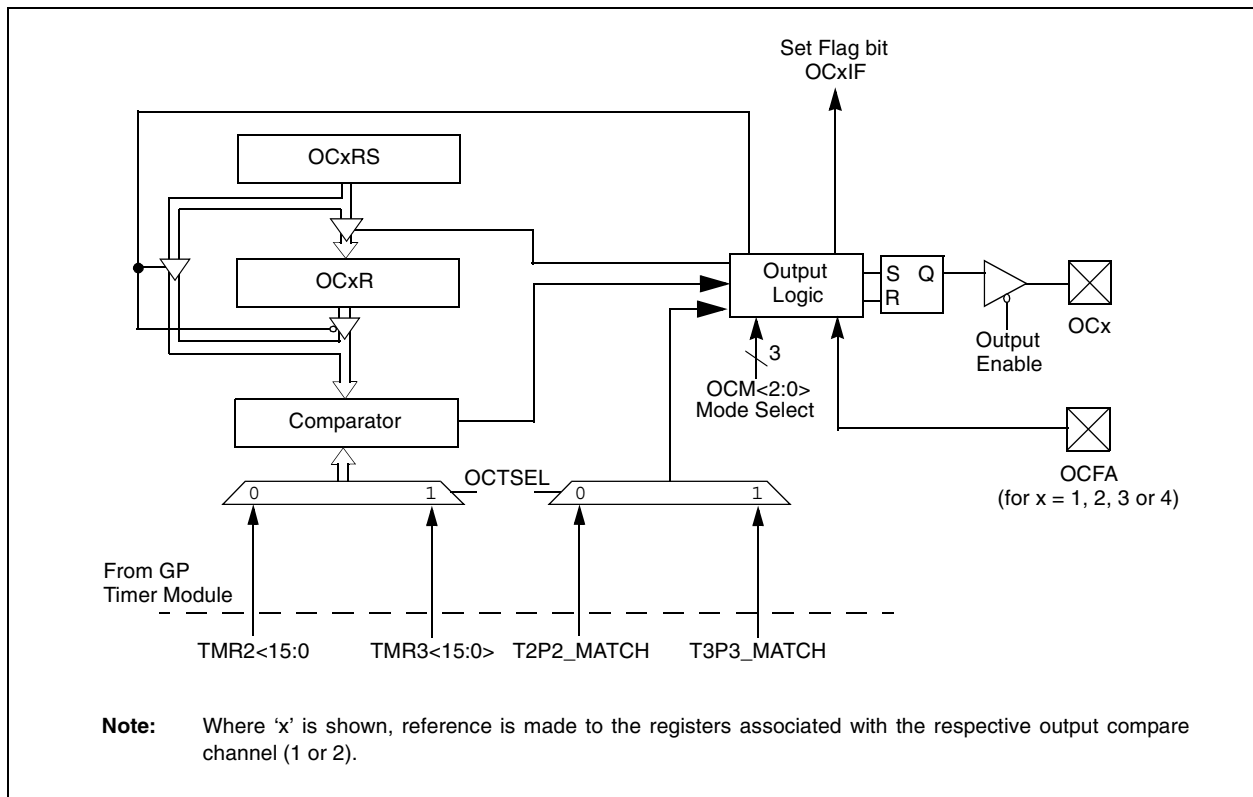
dsPIC30F2011/2012/3012/3013

12.1 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.

12.2 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.

12.3 Dual Output Compare Match Mode

When control bits OCM<2:0> (OCxCON<2:0>) = `100` or `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

12.3.1 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 T_{CY}.
- Calculate desired pulse width value based on T_{CY}.
- 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`.

12.3.2 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 T_{CY}.
- Calculate desired pulse value based on T_{CY}.
- 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`.

12.4 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`.

12.4.1 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.

Preliminary

12.4.2 PWM PERIOD

The PWM period is specified by writing to the PRx register. The PWM period can be calculated using Equation 12-1.

EQUATION 12-1:

$$\text{PWM period} = [(\text{PRx}) + 1] \cdot 4 \cdot \text{T}_{\text{OSC}} \cdot (\text{TMRx prescale value})$$

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 12-2 for key PWM period comparisons. Timer3 is referred to in Figure 12-2 for clarity.

FIGURE 12-2: PWM OUTPUT TIMING
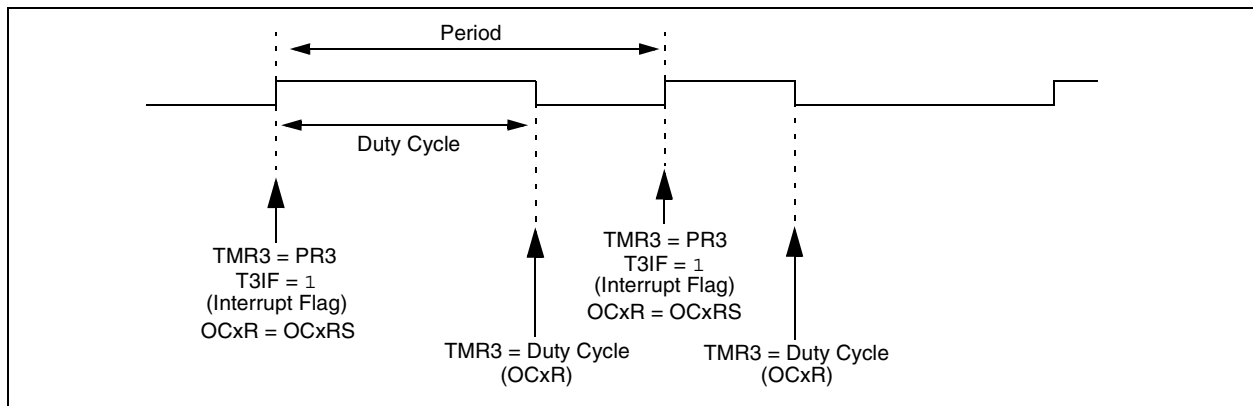
12.5 Output Compare Operation During CPU Sleep Mode

When the CPU enters 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.

12.6 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'.

12.7 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 12-1: 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	Bit 0	Reset State
OC1RS	0180						Output Compare 1 Secondary Register											0000 0000 0000 0000
OC1R	0182						Output Compare 1 Main Register											0000 0000 0000 0000
OC1CON	0184	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL		OCM<2:0>		0000 0000 0000 0000
OC2RS	0186						Output Compare 2 Secondary Register											0000 0000 0000 0000
OC2R	0188						Output Compare 2 Main Register											0000 0000 0000 0000
OC2CON	018A	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSE		OCM<2:0>		0000 0000 0000 0000

Legend: u = uninitialized bit

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

13.0   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. The dsPIC30F2011/2012/3012/3013 devices feature one SPI module, SPI1.

13.1   Operating Function Description

Figure 13-1 is a simplified block diagram of the SPI module, which consists of a 16-bit shift register, SPI1SR , used for shifting data in and out, and a buffer register, SPI1BUF. Control register SPI1CON (not shown) configures the module. Additionally, status register SPI1STAT (not shown) indicates various status conditions.

> **Note:**   See *dsPIC30F Family Reference Manual (DS70046)* for detailed information on the control and status registers.

Four I/O pins comprise the serial interface:

- SDI1 (serial data input)
- SDO1 (serial data output)
- SCK1 (shift clock input or output)

- $\overline{SS1}$ (active low slave select).

In Master mode operation, SCK1 is a clock output. In Slave mode, it is a clock input.

A series of eight (8) or sixteen (16) clock pulses shift out bits from the SPI1SR to SDO1 pin and simultaneously shift in data from SDI1 pin. An interrupt is generated when the transfer is complete and the interrupt flag bit (SPI1IF) is set. This interrupt can be disabled through the interrupt enable bit SPI1IE.

The receive operation is double-buffered. When a complete byte is received, it is transferred from SPI1SR to SPI1BUF.

If the receive buffer is full when new data is being transferred from SPI1SR to SPI1BUF, the module will set the SPIROV bit indicating an overflow condition. The transfer of the data from SPI1SR to SPI1BUF 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 SPI1BUF is read by user software.

Transmit writes are also double-buffered. The user writes to SPI1BUF. When the master or slave transfer is completed, the contents of the shift register (SPI1SR) are 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 SPI1SR. The received data is thus placed in SPI1BUF and the transmit data in SPI1SR is ready for the next transfer.

> **Note:**   Both the transmit buffer (SPI1TXB) and the receive buffer (SPI1RXB) are mapped to the same register address, SPI1BUF.

FIGURE 13-1:   SPI BLOCK DIAGRAM
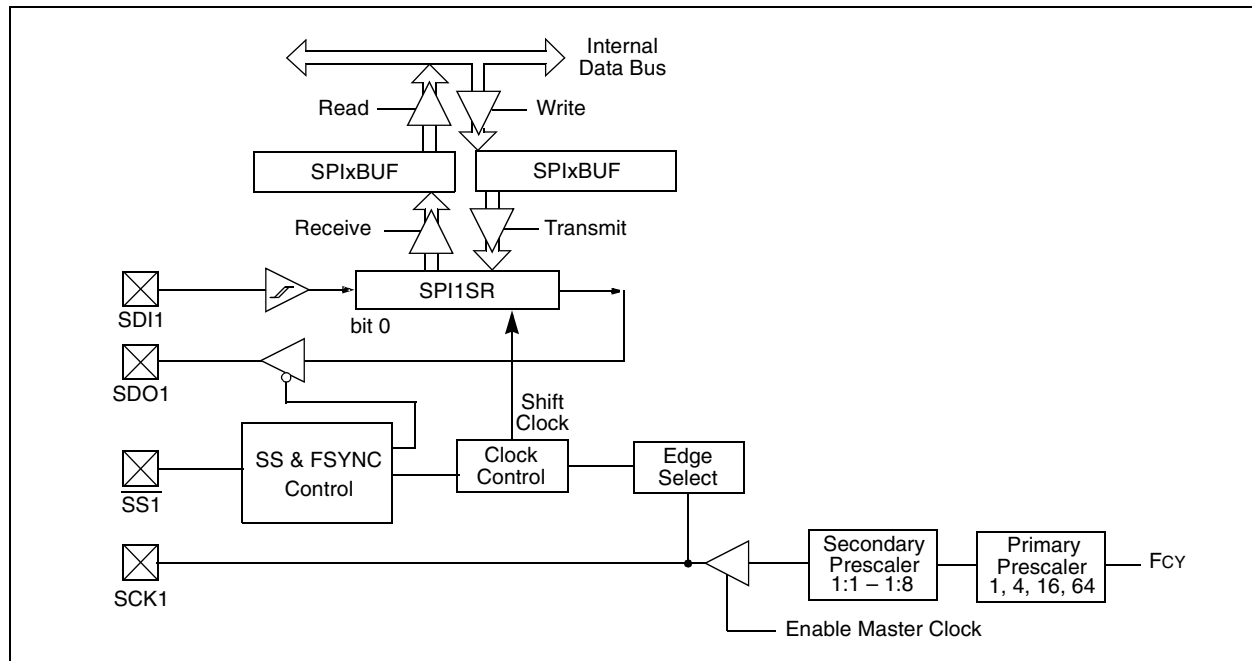
Figure 13-2 depicts the a master/slave connection between two processors. In Master mode, the clock is generated by prescaling the system clock. Data is transmitted as soon as a value is written to SPI1BUF. The interrupt is generated at the middle of the transfer of the last bit.

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 $\overline{SS1}$ control is enabled, then transmission and reception are enabled only when $\overline{SS1}$ = low. The SDO1 output will be disabled in $\overline{SS1}$ mode with $\overline{SS1}$ high.

The clock provided to the module is (F_{OSC}/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.

13.1.1   WORD AND BYTE COMMUNICATION

A control bit, MODE16 (SPI1CON<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 SPI1SR for 8-bit operation, and data is transmitted out of bit15 of the SPI1SR for 16-bit operation. In both modes, data is shifted into bit 0 of the SPI1SR.

13.1.2   SDO1 DISABLE

A control bit, DISSDO, is provided to the SPI1CON register to allow the SDO1 output to be disabled. This will allow the SPI module to be connected in an input only configuration. SDO1 can also be used for general purpose I/O.

13.2   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 $\overline{SS1}$ pin to perform the frame synchronization pulse (FSYNC) function. The control bit SPIFSD determines whether the $\overline{SS1}$ 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.

FIGURE 13-2:   **SPI MASTER/SLAVE CONNECTION**
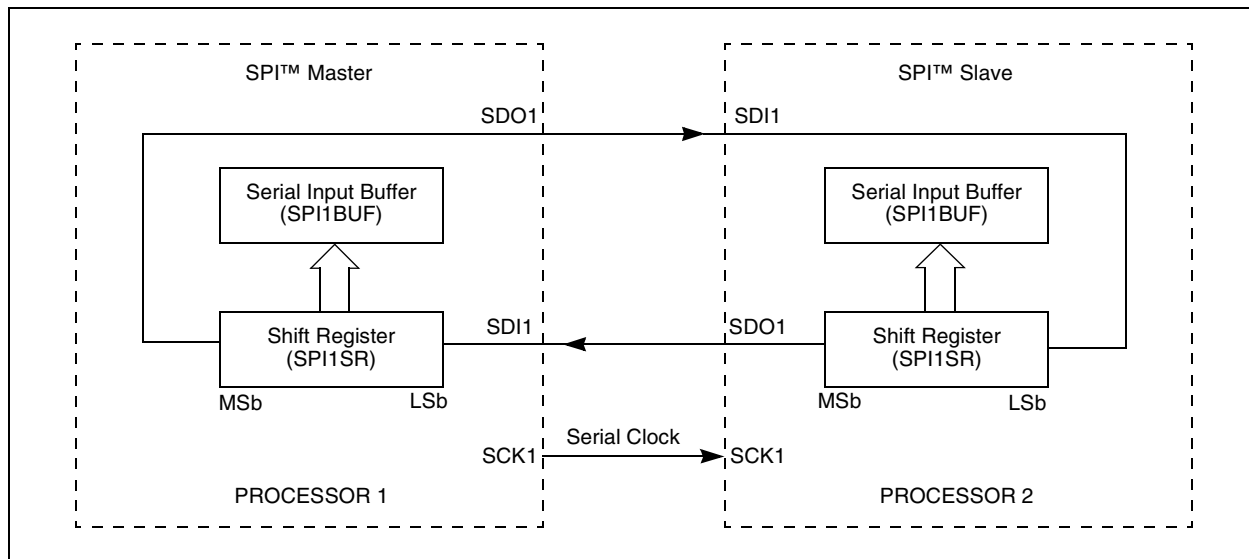
13.3   Slave Select Synchronization

The $\overline{SS1}$ pin allows a Synchronous Slave mode. The SPI must be configured in SPI Slave mode with $\overline{SS1}$ pin control enabled (SSEN = 1). When the $\overline{SS1}$ pin is low, transmission and reception are enabled and the SDOx pin is driven. When $\overline{SS1}$ pin goes high, the SDOx pin is no longer driven. Also, the SPI module is re-synchronized, and all counters/control circuitry are reset. Therefore, when the $\overline{SS1}$ pin is asserted low again, transmission/reception will begin at the MS bit even if $\overline{SS1}$ had been de-asserted in the middle of a transmit/receive.

13.4   SPI Operation During CPU Sleep Mode

During Sleep mode, the SPI module is shutdown. 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.

13.5   SPI Operation During CPU Idle Mode

When the device enters Idle mode, all clock sources remain functional. The SPISIDL bit (SPI1STAT<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 13-1:   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	Bit 0	Reset State
SPI1STAT	0220	SPIEN	—	SPISIDL	—	—	—	—	—	—	SPIROV	—	—	—	—	—	SPITBF	SPIRBF	0000 0000 0000 0000
SPI1CON	0222	—	FRMEN	SPIFSD	—	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000 0000 0000 0000
SPI1BUF	0224	Transmit and Receive Buffer																0000 0000 0000 0000

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

14.0   I²C 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 (I²C™) module provides complete hardware support for both Slave and Multi-Master modes of the I²C serial communication standard, with a 16-bit interface.

This module offers the following key features:

- I²C interface supporting both master and slave operation.
- I²C Slave mode supports 7 and 10-bit address.
- I²C Master mode supports 7 and 10-bit address.
- I²C port allows bidirectional transfers between master and slaves.
- Serial clock synchronization for I²C port can be used as a handshake mechanism to suspend and resume serial transfer (SCLREL control).
- I²C supports multi-master operation; detects bus collision and will arbitrate accordingly.

14.1   Operating Function Description

The hardware fully implements all the master and slave functions of the I²C Standard and Fast mode specifications, as well as 7 and 10-bit addressing.

Thus, the I²C module can operate either as a slave or a master on an I²C bus.

14.1.1   VARIOUS I²C MODES

The following types of I²C operation are supported:

- I²C slave operation with 7-bit address
- I²C slave operation with 10-bit address
- I²C master operation with 7 or 10-bit address

See the I²C programmer's model (Figure 14-1).

14.1.2   PIN CONFIGURATION IN I²C MODE

I²C has a 2-pin interface: the SCL pin is clock and the SDA pin is data.

14.1.3   I²C 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 14-1. I2CTRN is the transmit register to which bytes are written during a transmit operation, as shown in Figure 14-2.

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 14-1:   PROGRAMMER'S MODEL
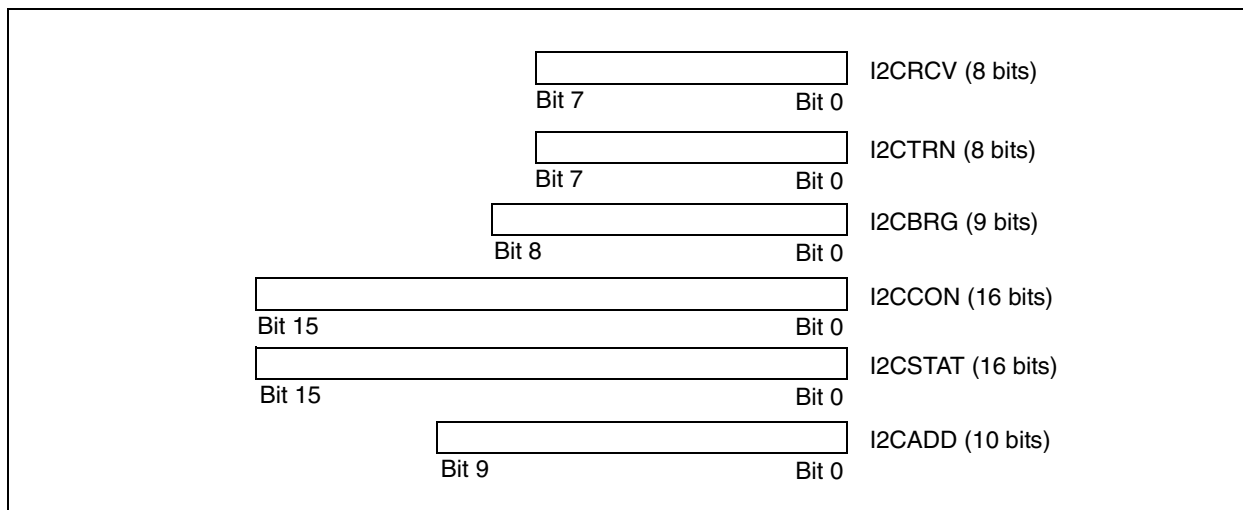
FIGURE 14-2: **I²C BLOCK DIAGRAM**
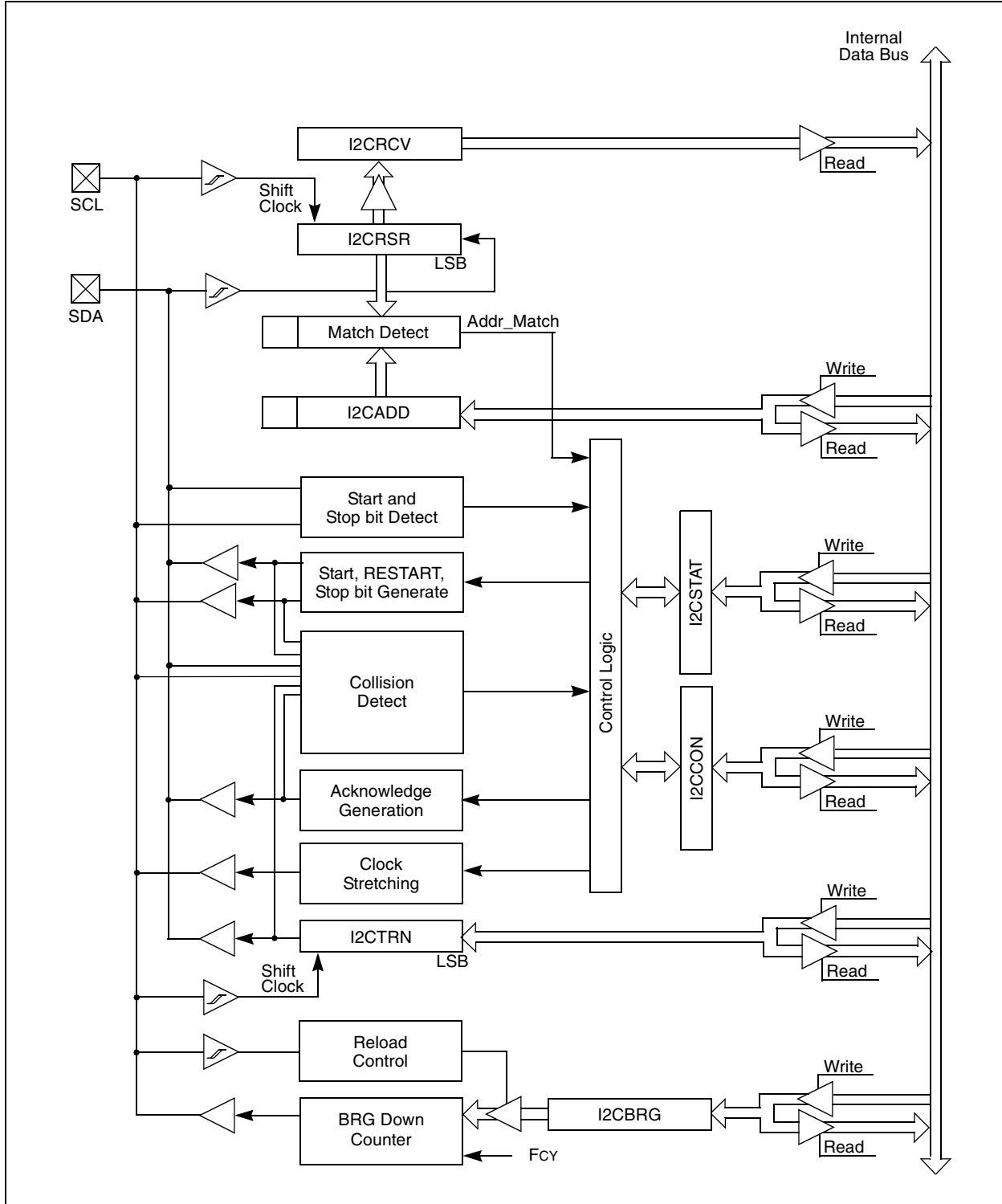
14.2 I²C Module Addresses

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 LS bits of the I2CADD register.

If the A10M bit is '1', the address is assumed to be a 10-bit address. When an address is received, it will be compared with the binary value '11110 A9 A8' (where A9 and 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.

The 7-bit I²C Slave Addresses supported by the dsPIC30Fare shown in Table 14-1.

TABLE 14-1: 7-BIT I²C SLAVE ADDRESSES

0x00	General call address or start byte
0x01-0x03	Reserved
0x04-0x77	Valid 7-bit addresses
0x78-0x7b	Valid 10-bit addresses (lower 7 bits)
0x7c-0x7f	Reserved

14.3 I²C 7-bit Slave Mode Operation

Once enabled (I2CEN = 1), the slave module will wait for a Start bit to occur (i.e., the I²C 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 will be 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.

14.3.1 SLAVE TRANSMISSION

If the R_W bit received is a '1', then the serial port will go into Transmit mode. It will send $\overline{ACK}$ on the ninth bit and then hold 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. The interrupt pulse is sent on the falling edge of the ninth clock pulse, regardless of the status of the $\overline{ACK}$ received from the master.

14.3.2 SLAVE RECEPTION

If the R_W bit received is a '0' during an address match, then 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. $\overline{ACK}$ 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), then $\overline{ACK}$ 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 will be 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.

14.4 I²C 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 I²C 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 will be 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.

14.4.1 10-BIT MODE SLAVE TRANSMISSION

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

14.4.2 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.

14.5 Automatic Clock Stretch

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

14.5.1 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 $\overline{\text{ACK}}$ 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 will not be cleared and clock stretching will not occur.
2:	The SCLREL bit can be set in software, regardless of the state of the TBF bit.

14.5.2 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 will be held low at the end of each data receive sequence.

14.5.3 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 I^{2}CRCV before the master device can initiate another receive sequence. This will prevent 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 will not be cleared and clock stretching will 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.

14.5.4 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 will occur on each data receive or transmit sequence as was described earlier.

14.6 Software Controlled Clock Stretching (STREN = 1)

When the STREN bit is '1', the SCLREL bit may be cleared by software to allow software to control the clock stretching. The logic will synchronize writes to the SCLREL bit with the SCL clock. Clearing the SCLREL bit will 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 will be asserted (held low). The SCL output will remain low until the SCLREL bit is set, and all other devices on the I^{2}C bus have de-asserted SCL. This ensures that a write to the SCLREL bit will not violate the minimum high time requirement for SCL.

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

14.7   Interrupts

The I^{2}C module generates two interrupt flags, MI2CIF (I^{2}C Master Interrupt Flag) and SI2CIF (I^{2}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.

14.8   Slope Control

The I^{2}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.

14.9   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.

14.10   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^{2}C 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<15> = 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 ($\overline{ACK}$ 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.

14.11   I^{2}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^{2}C port to receive data.
- Generate an $\overline{ACK}$ condition at the end of a received byte of data.

14.12   I^{2}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^{2}C 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 $\overline{ACK}$ 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 $\overline{ACK}$ bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

14.12.1   I^{2}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.

14.12.2   I²C MASTER RECEPTION

Master mode reception is enabled by programming the Receive Enable bit, RCEN (I2CCON<11>). The I²C module must be Idle before the RCEN bit is set, otherwise the RCEN bit will be disregarded. The baud rate generator begins counting and on each rollover, the state of the SCL pin $\overline{ACK}$ and data are shifted into the I2CRSR on the rising edge of each clock.

14.12.3   BAUD RATE GENERATOR

In I²C 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 I²C 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 14-1:   SERIAL CLOCK RATE

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

14.12.4   CLOCK ARBITRATION

Clock arbitration occurs when the master de-asserts 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 (BRG) 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.

14.12.5   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 will set the MI2CIF pulse and reset the master portion of the I²C 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 de-asserted and a value can now be written to I2CTRN. When the user services the I²C master event Interrupt Service Routine, if the I²C 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 de-asserted, 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 I²C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins, and if a Stop condition occurs, the MI2CIF bit will be set.

A write to the I2CTRN will start 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 I²C 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.

14.13   I²C Module Operation During CPU Sleep and Idle Modes

14.13.1   I²C 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, then the transmission is aborted. Similarly, if Sleep occurs in the middle of a reception, then the reception is aborted.

14.13.2   I²C OPERATION DURING CPU IDLE MODE

For the I²C, the I2CSIDL bit selects if the module will stop on Idle or continue on Idle. If I2CSIDL = 0, the module will continue operation on assertion of the Idle mode. If I2CSIDL = 1, the module will stop on Idle.

TABLE 14-2: **I²C REGISTER MAP**

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
I2CRCV	0200	—	—	—	—	—	—	—	—	Receive Register								0000 0000 0000 0000
I2CTRN	0202	—	—	—	—	—	—	—	—	Transmit Register								0000 0000 1111 1111
I2CBRG	0204	—	—	—	—	—	—	—	Baud Rate Generator									0000 0000 0000 0000
I2CCON	0206	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0001 0000 0000 0000
I2CSTAT	0208	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D_A	P	S	R_W	RBF	TBF	0000 0000 0000 0000
I2CADD	020A	—	—	—	—	—	—	Address Register										0000 0000 0000 0000

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

NOTES:

15.0 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. The dsPIC30F2011/2012/3012 processors have one UART module (UART1). The dsPIC30F3013 processor has two UART modules (UART1 and UART2).

15.1 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
- Alternate receive and transmit pins for UART1

FIGURE 15-1: UART TRANSMITTER BLOCK DIAGRAM
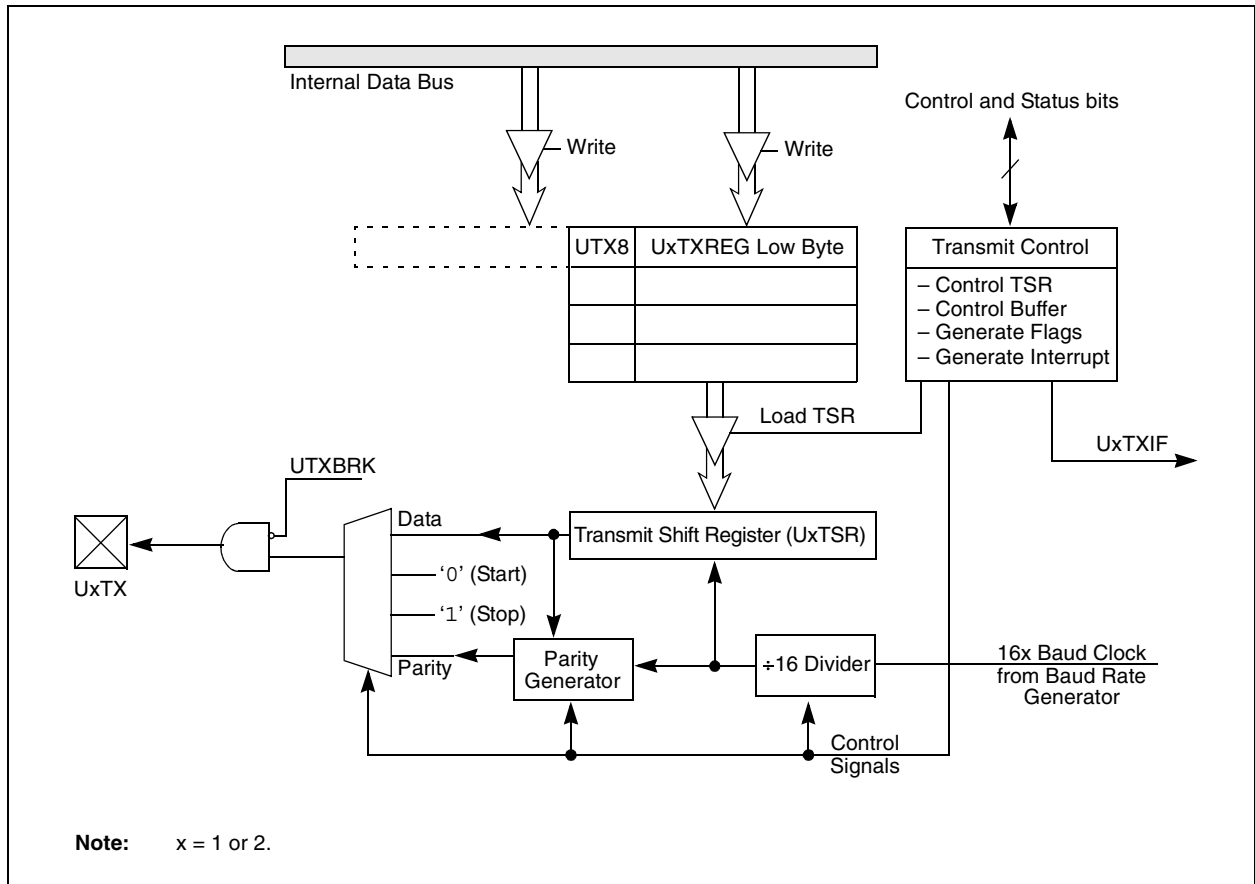
FIGURE 15-2: **UART RECEIVER BLOCK DIAGRAM**
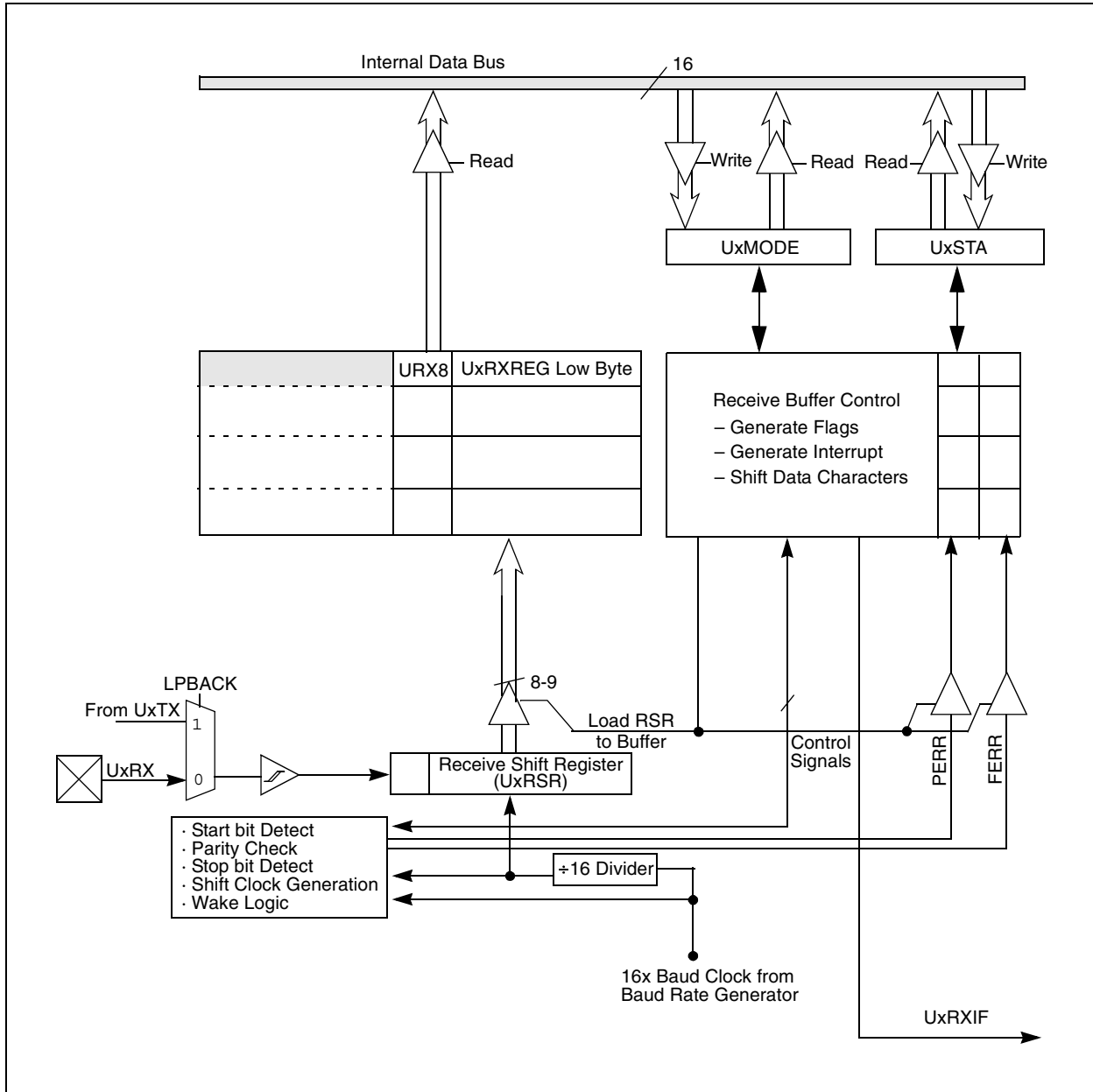
15.2   Enabling and Setting Up UART

15.2.1   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.

15.2.2   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.

15.2.3   ALTERNATE I/O

The alternate I/O function is enabled by setting the ALTIO bit (UxMODE<10>). If ALTIO = 1, the UxATX and UxARX pins (alternate transmit and alternate receive pins, respectively) are used by the UART module instead of the UxTX and UxRX pins. If ALTIO = 0, the UxTX and UxRX pins are used by the UART module.

15.2.4   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 and 1 Stop bit (typically represented as 8, N, 1).

15.3   Transmitting Data

15.3.1   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.
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>).

15.3.2   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.

15.3.3   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.

15.3.4   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 the 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.

15.3.5   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.

15.4   Receiving Data

15.4.1   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 15.3.1).
2.   Enable the UART (see Section 15.3.1).
3.   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.
5.   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.

15.4.2   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.

15.4.3   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.

15.5   Reception Error Handling

15.5.1   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.

15.5.2 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.

15.5.3 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.

15.5.4 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.

15.5.5 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 yet been received.

15.6 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.

15.7 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 15.3.

15.8 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)

F_{CY} = Instruction Clock Rate (1/T_{CY})

The Baud Rate is given by Equation 15-1.

EQUATION 15-1: BAUD RATE

$$\text{Baud Rate} = F_{CY} / (16*(BRG+1))$$

Therefore, the maximum baud rate possible is

F_{CY} /16   (if BRG = 0),

and the minimum baud rate possible is

F_{CY} / (16* 65536).

With a full 16-bit baud rate generator at 30 MIPs operation, the minimum baud rate achievable is 28.5 bps.

15.9   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 (IC1 for UART1 and IC2 for UART2). To enable this mode, you must program the input capture module to detect the falling and rising edges of the Start bit.

15.10   UART Operation During CPU Sleep and Idle Modes

15.10.1   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.

15.10.2   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 15-1: UART1 REGISTER MAP FOR dsPIC30F2011/2012/3012/3013

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
U1MODE	020C	UARTEN	—	USIDL	—	—	ALTIO	—	—	WAKE	LPBACK	ABAUD	—	—	PDSEL1	PDSEL0	STSEL	0000 0000 0000 0000
U1STA	020E	UTXISEL	—	—	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0000 0001 0001 0000
U1TXREG	0210	—	—	—	—	—	—	—	UTX8	Transmit Register								0000 000u uuuu uuuu
U1RXREG	0212	—	—	—	—	—	—	—	URX8	Receive Register								0000 0000 0000 0000
U1BRG	0214	Baud Rate Generator Prescaler																0000 0000 0000 0000

Legend: u = uninitialized bit

TABLE 15-2: UART2 REGISTER MAP FOR DSPIC30F3013

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
U2MODE	0216	UARTEN	—	USIDL	—	—	—	—	—	WAKE	LPBACK	ABAUD	—	—	PDSEL1	PDSEL0	STSEL	0000 0000 0000 0000
U2STA	0218	UTXISEL	—	—	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0000 0001 0001 0000
U2TXREG	021A	—	—	—	—	—	—	—	UTX8	Transmit Register								0000 000u uuuu uuuu
U2RXREG	021C	—	—	—	—	—	—	—	URX8	Receive Register								0000 0000 0000 0000
U2BRG	021E	Baud Rate Generator Prescaler																0000 0000 0000 0000

Legend: u = uninitialized bit

Note 1: UART2 is not available on the dsPIC30F2011/2012/3012

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

NOTES:

16.0 12-BIT ANALOG-TO-DIGITAL CONVERTER (A/D) 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 12-bit Analog-to-Digital converter (A/D) allows conversion of an analog input signal to a 12-bit digital number. This module is based on a Successive Approximation Register (SAR) architecture and provides a maximum sampling rate of 100 ksps. The A/D module has up to 10 analog inputs which are multiplexed into a sample and hold amplifier. The output of the sample and hold is the input into the converter which generates the result. The analog reference voltage is software selectable to either the device supply voltage (AVDD/AVSS) or the voltage level on the (VREF+/VREF-) pin. The A/D converter has a unique feature of being able to operate while the device is in Sleep mode with RC oscillator selection.

The A/D module has six 16-bit registers:

- A/D Control Register 1 (ADCON1)
- A/D Control Register 2 (ADCON2)
- A/D Control Register 3 (ADCON3)
- A/D Input Select Register (ADCHS)
- A/D Port Configuration Register (ADPCFG)
- A/D 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, SMPI<3:0>, BUFM and ALTS bits, as well as the ADCON3 and ADCSSL registers, must not be written to while ADON = 1. This would lead to indeterminate results.

The block diagram of the 12-bit A/D module is shown in Figure 16-1.

FIGURE 16-1: 12-BIT A/D FUNCTIONAL BLOCK DIAGRAM
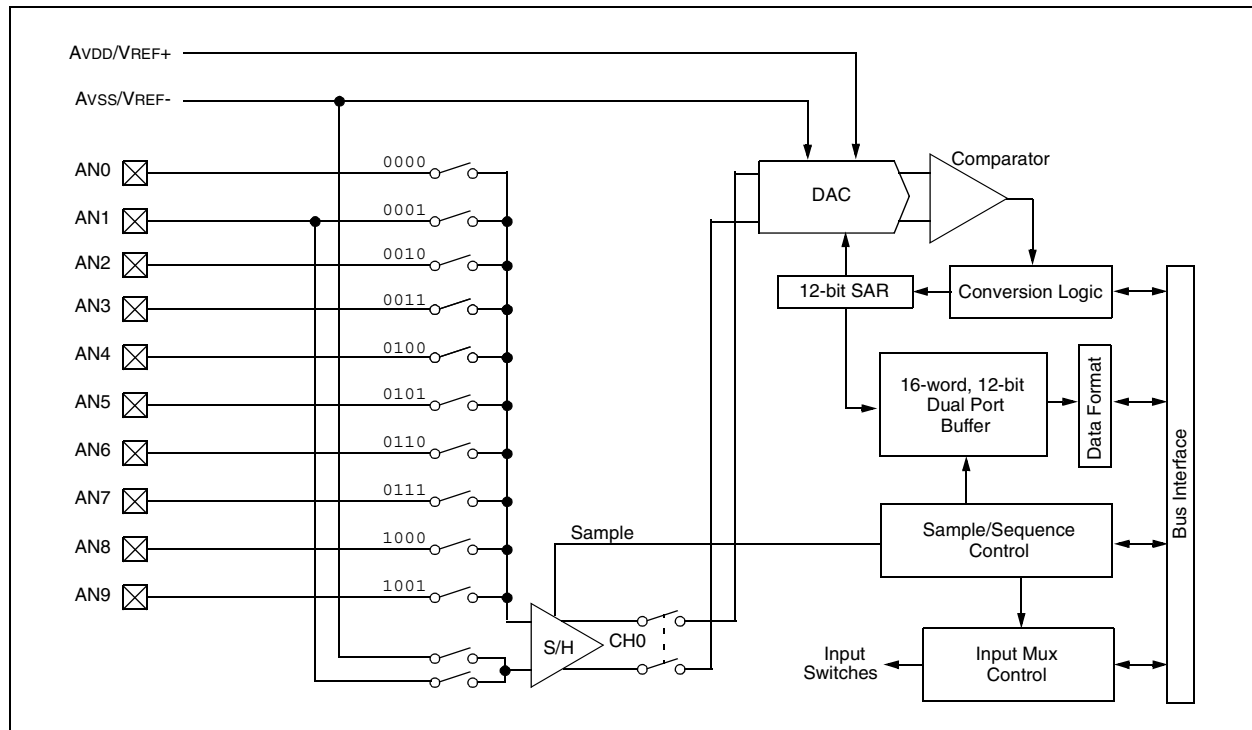
16.1   A/D Result Buffer

The module contains a 16-word dual port read only buffer, called ADCBUF0...ADCBUFF, to buffer the A/D results. The RAM is 12 bits wide but the data obtained is represented in one of four different 16-bit data formats. The contents of the sixteen A/D Conversion Result Buffer registers, ADCBUF0 through ADCBUFF, cannot be written by user software.

16.2   Conversion Operation

After the A/D 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 DONE bit and the A/D interrupt flag ADIF are set after the number of samples specified by the SMPI bit. The ADC module can be configured for different interrupt rates as described in Section 16.3.

The following steps should be followed for doing an A/D conversion:

1. Configure the A/D module:
    • Configure analog pins, voltage reference and digital I/O
    •Select A/D input channels
    •Select A/D conversion clock
    •Select A/D conversion trigger
    • Turn on A/D module
2. Configure A/D 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 A/D conversion to complete, by either:
    • Waiting for the A/D interrupt, or
    • Waiting for the DONE bit to get set.
7. Read A/D result buffer, clear ADIF if required.

16.3   Selecting the Conversion Sequence

Several groups of control bits select the sequence in which the A/D connects inputs to the sample/hold channel, converts a channel, writes the buffer memory and generates interrupts.

The sequence is controlled by the sampling clocks.

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 BUFM bit will split the 16-word results buffer 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 the moving of the buffers after the interrupt.

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 (corresponding to the 16 input channels) may be done per interrupt. The processor will have one acquisition 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 multiplexer input 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.

16.4 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 4 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 A/D clock control. The SAMC bits select the number of A/D 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 or external interrupts.

16.5 Aborting a Conversion

Clearing the ADON bit during a conversion will abort the current conversion and stop the sampling sequencing until the next sampling trigger. 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 and a new conversion will not start.

After the A/D conversion is aborted, a 2 T_{AD} wait is required before the next sampling may be started by setting the SAMP bit.

16.6 Selecting the A/D Conversion Clock

The A/D conversion requires 14 T_{AD}. The source of the A/D conversion clock is software selected, using a six-bit counter. There are 64 possible options for T_{AD}.

EQUATION 16-1: A/D CONVERSION CLOCK

$$T_{AD} = T_{CY} * (0.5*(ADCS<5:0> + 1))$$

The internal RC oscillator is selected by setting the ADRC bit.

For correct A/D conversions, the A/D conversion clock (T_{AD}) must be selected to ensure a minimum T_{AD} time of 667 nsec (for V_{DD} = 5V). Refer to the Electrical Specifications section for minimum T_{AD} under other operating conditions.

Example 16-1 shows a sample calculation for the ADCS<5:0> bits, assuming a device operating speed of 30 MIPS.

EXAMPLE 16-1: A/D CONVERSION CLOCK AND SAMPLING RATE CALCULATION

$$\text{Minimum } T_{AD} = 667 \text{ nsec}$$
$$T_{CY} = 33.33 \text{ nsec (30 MIPS)}$$

$$ADCS<5:0> = 2 \frac{T_{AD}}{T_{CY}} - 1$$
$$= 2 \bullet \frac{667 \text{ nsec}}{33.33 \text{ nsec}} - 1$$
$$= 39$$
Therefore,
Set ADCS<5:0> = 39

$$\text{Actual } T_{AD} = \frac{T_{CY}}{2} (ADCS<5:0> + 1)$$
$$= \frac{33.33 \text{ nsec}}{2} (39 + 1)$$
$$= 667 \text{ nsec}$$

If SSRC<2:0> = '111' and SAMC<4:0> = '00001'

Since,
Sampling Time = Acquisition Time + Conversion Time
$$= 1 \text{ } T_{AD} + 14 \text{ } T_{AD}$$
$$= 15 \text{ x } 667 \text{ nsec}$$

Therefore,
$$\text{Sampling Rate} = \frac{1}{(15 \text{ x } 667 \text{ nsec})}$$
$$= \sim100 \text{ kHz}$$

16.7 A/D Acquisition Requirements

The analog input model of the 12-bit A/D converter is shown in Figure 16-2. The total sampling time for the A/D is a function of the internal amplifier settling time and the holding capacitor charge time.

For the A/D converter to meet its specified accuracy, the charge holding capacitor (C_{HOLD}) must be allowed to fully charge to the voltage level on the analog input pin. The source impedance (R_S), the interconnect impedance (R_{IC}), and the internal sampling switch (R_{SS}) impedance combine to directly affect the time required to charge the capacitor C_{HOLD}. 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, R_S, is 2.5 kΩ. 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.

FIGURE 16-2: **12-BIT A/D CONVERTER ANALOG INPUT MODEL**

Legend: CPIN = input capacitance
VT = threshold voltage
I leakage = leakage current at the pin due to various junctions
RIC = interconnect resistance
RSS = sampling switch resistance
CHOLD = sample/hold capacitance (from DAC)

Note: CPIN value depends on device package and is not tested. Effect of CPIN negligible if Rs ≤ 2.5 kΩ.
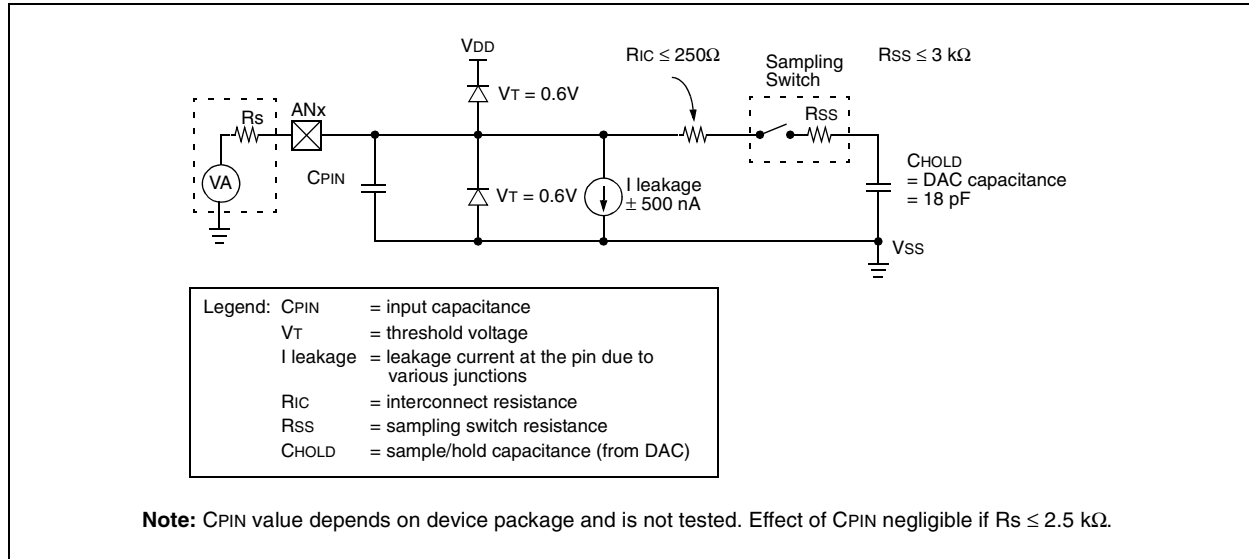
16.8 Module Power-down Modes

The module has 2 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.

16.9 A/D Operation During CPU Sleep and Idle Modes

16.9.1 A/D 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 A/D module can operate during Sleep mode if the A/D clock source is set to RC (ADRC = 1). When the RC clock source is selected, the A/D 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 CONV bit will be cleared and the result loaded into the ADCBUF register.

If the A/D interrupt is enabled, the device will wake-up from Sleep. If the A/D interrupt is not enabled, the A/D module will then be turned off, although the ADON bit will remain set.

16.9.2 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.

16.10 Effects of a Reset

A device Reset forces all registers to their Reset state. This forces the A/D module to be turned off, and any conversion and sampling sequence is aborted. The values that are in the ADCBUF registers are not modified. The A/D Result register will contain unknown data after a Power-on Reset.

16.11 Output Formats

The A/D result is 12 bits wide. The data buffer RAM is also 12 bits wide. The 12-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.

FIGURE 16-3: **A/D OUTPUT DATA FORMATS**

RAM Contents:				d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00

Read to Bus:

Signed Fractional	$\overline{d11}$	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0

Fractional	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0

Signed Integer	d11	d11	d11	d11	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00

Integer	0	0	0	0	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00

16.12 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 (V_{OH} or V_{OL}) 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.

16.13 Connection Considerations

The analog inputs have diodes to V_{DD} and V_{SS} as ESD protection. This requires that the analog input be between V_{DD} and V_{SS}. 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 anti-aliasing 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 16-1: A/D CONVERTER REGISTER MAP FOR DSPIC30F2011/3012

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
ADCBUF0	0280	—	—	—	—	ADC Data Buffer 0												0000 uuuu uuuu uuuu
ADCBUF1	0282	—	—	—	—	ADC Data Buffer 1												0000 uuuu uuuu uuuu
ADCBUF2	0284	—	—	—	—	ADC Data Buffer 2												0000 uuuu uuuu uuuu
ADCBUF3	0286	—	—	—	—	ADC Data Buffer 3												0000 uuuu uuuu uuuu
ADCBUF4	0288	—	—	—	—	ADC Data Buffer 4												0000 uuuu uuuu uuuu
ADCBUF5	028A	—	—	—	—	ADC Data Buffer 5												0000 uuuu uuuu uuuu
ADCBUF6	028C	—	—	—	—	ADC Data Buffer 6												0000 uuuu uuuu uuuu
ADCBUF7	028E	—	—	—	—	ADC Data Buffer 7												0000 uuuu uuuu uuuu
ADCBUF8	0290	—	—	—	—	ADC Data Buffer 8												0000 uuuu uuuu uuuu
ADCBUF9	0292	—	—	—	—	ADC Data Buffer 9												0000 uuuu uuuu uuuu
ADCBUFA	0294	—	—	—	—	ADC Data Buffer 10												0000 uuuu uuuu uuuu
ADCBUFB	0296	—	—	—	—	ADC Data Buffer 11												0000 uuuu uuuu uuuu
ADCBUFC	0298	—	—	—	—	ADC Data Buffer 12												0000 uuuu uuuu uuuu
ADCBUFD	029A	—	—	—	—	ADC Data Buffer 13												0000 uuuu uuuu uuuu
ADCBUFE	029C	—	—	—	—	ADC Data Buffer 14												0000 uuuu uuuu uuuu
ADCBUFF	029E	—	—	—	—	ADC Data Buffer 15												0000 uuuu uuuu uuuu
ADCON1	02A0	ADON	—	ADSIDL	—	—	—	FORM<1:0>		SSRC<2:0>			—	—	ASAM	SAMP	DONE	0000 0000 0000 0000
ADCON2	02A2	VCFG<2:0>			—	—	CSCNA	—	—	BUFS	—	SMPI<3:0>				BUFM	ALTS	0000 0000 0000 0000
ADCON3	02A4	—	—	—	SAMC<4:0>					ADRC	—	ADCS<5:0>						0000 0000 0000 0000
ADCHS	02A6	—	—	—	CH0NB	CH0SB<3:0>				—	—	—	CH0NA	CH0SA<3:0>				0000 0000 0000 0000
ADPCFG	02A8	—	—	—	—	—	—	—	—	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000 0000 0000 0000
ADCSSL	02AA	—	—	—	—	—	—	—	—	CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000 0000 0000 0000

Legend: u = uninitialized bit

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

Preliminary

TABLE 16-2: A/D CONVERTER REGISTER MAP FOR DSPIC30F2012/3013

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
ADCBUF0	0280	—	—	—	—					ADC Data Buffer 0								0000 uuuu uuuu uuuu
ADCBUF1	0282	—	—	—	—					ADC Data Buffer 1								0000 uuuu uuuu uuuu
ADCBUF2	0284	—	—	—	—					ADC Data Buffer 2								0000 uuuu uuuu uuuu
ADCBUF3	0286	—	—	—	—					ADC Data Buffer 3								0000 uuuu uuuu uuuu
ADCBUF4	0288	—	—	—	—					ADC Data Buffer 4								0000 uuuu uuuu uuuu
ADCBUF5	028A	—	—	—	—					ADC Data Buffer 5								0000 uuuu uuuu uuuu
ADCBUF6	028C	—	—	—	—					ADC Data Buffer 6								0000 uuuu uuuu uuuu
ADCBUF7	028E	—	—	—	—					ADC Data Buffer 7								0000 uuuu uuuu uuuu
ADCBUF8	0290	—	—	—	—					ADC Data Buffer 8								0000 uuuu uuuu uuuu
ADCBUF9	0292	—	—	—	—					ADC Data Buffer 9								0000 uuuu uuuu uuuu
ADCBUFA	0294	—	—	—	—					ADC Data Buffer 10								0000 uuuu uuuu uuuu
ADCBUFB	0296	—	—	—	—					ADC Data Buffer 11								0000 uuuu uuuu uuuu
ADCBUFC	0298	—	—	—	—					ADC Data Buffer 12								0000 uuuu uuuu uuuu
ADCBUFD	029A	—	—	—	—					ADC Data Buffer 13								0000 uuuu uuuu uuuu
ADCBUFE	029C	—	—	—	—					ADC Data Buffer 14								0000 uuuu uuuu uuuu
ADCBUFF	029E	—	—	—	—					ADC Data Buffer 15								0000 uuuu uuuu uuuu
ADCON1	02A0	ADON	—	ADSIDL	—	—	—	FORM<1:0>		SSRC<2:0>			—	—	ASAM	SAMP	DONE	0000 0000 0000 0000
ADCON2	02A2	VCFG<2:0>			—	—	CSCNA	—	—	BUFS	—	SMPI<3:0>				BUFM	ALTS	0000 0000 0000 0000
ADCON3	02A4	—	—	—	—	SAMC<4:0>				ADRC	—	ADCS<5:0>						0000 0000 0000 0000
ADCHS	02A6	—	—	—	CH0NB	CH0SB<3:0>				—	—		CH0NA	CH0SA<3:0>				0000 0000 0000 0000
ADPCFG	02A8	—	—	—	—	—	—	PCFG9	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000 0000 0000 0000
ADCSSL	02AA	—	—	—	—	—	—	CSSL9	CSSL8	CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000 0000 0000 0000

Legend: u = uninitialized bit

Note: Refer to *dSPIC30F Family Reference Manual* (DS70046) for descriptions of register bit fields.

NOTES:

Preliminary © 2005 Microchip Technology Inc.

17.0 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 Programmer's Reference Manual* (DS70030).

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 Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable. The other is the Power-up 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.

17.1 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

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 17-1 provides a summary of the dsPIC30F Oscillator Operating modes. A simplified diagram of the oscillator system is shown in Figure 17-1.

TABLE 17-1: OSCILLATOR OPERATING MODES

Oscillator Mode	Description
XTL	200 kHz-4 MHz crystal on OSC1:OSC2.
XT	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-7.5 MHz crystal on OSC1:OSC2, 16x PLL enabled[1].
LP	32 kHz crystal on SOSCO:SOSCI[2].
HS	10 MHz-25 MHz crystal.
HS/2 w/PLL 4x	10 MHz -20 MHz crystal, divide by 2, 4x PLL enabled.
HS/2 w/ PLL 8x	10 MHz-20 MHz crystal, divide by 2, 8x PLL enabled.
HS/2 w/ PLL 16x	10 MHz-15 MHz crystal, divide by 2, 16x PLL enabled[1].
HS/3 w/PLL 4x	12 MHz-25 MHz crystal, divide by 3, 4x PLL enabled.
HS/3 w/ PLL 8x	12 MHz-25 MHz crystal, divide by 3, 8x PLL enabled.
HS/3 w/ PLL 16x	12 MHz-22.5 MHz crystal, divide by 3, 16x PLL enabled[1].
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 (4-10 MHz), OSC2 pin is I/O, 4x PLL enabled.
EC w/ PLL 8x	External clock input (4-10 MHz), OSC2 pin is I/O, 8x PLL enabled.
EC w/ PLL 16x	External clock input (4-7.5 MHz), OSC2 pin is I/O, 16x PLL enabled[1].
ERC	External RC oscillator, OSC2 pin is F$_{OSC}$/4 output[3].
ERCIO	External RC oscillator, OSC2 pin is I/O[3].
FRC	7.37 MHz internal RC oscillator.
FRC w/ PLL 4x	7.37 MHz Internal RC oscillator, 4x PLL enabled.
FRC w/ PLL 8x	7.37 MHz Internal RC oscillator, 8x PLL enabled.
FRC w/ PLL 16x	7.37 MHz Internal RC oscillator, 16x PLL enabled.
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.

FIGURE 17-1:   OSCILLATOR SYSTEM BLOCK DIAGRAM
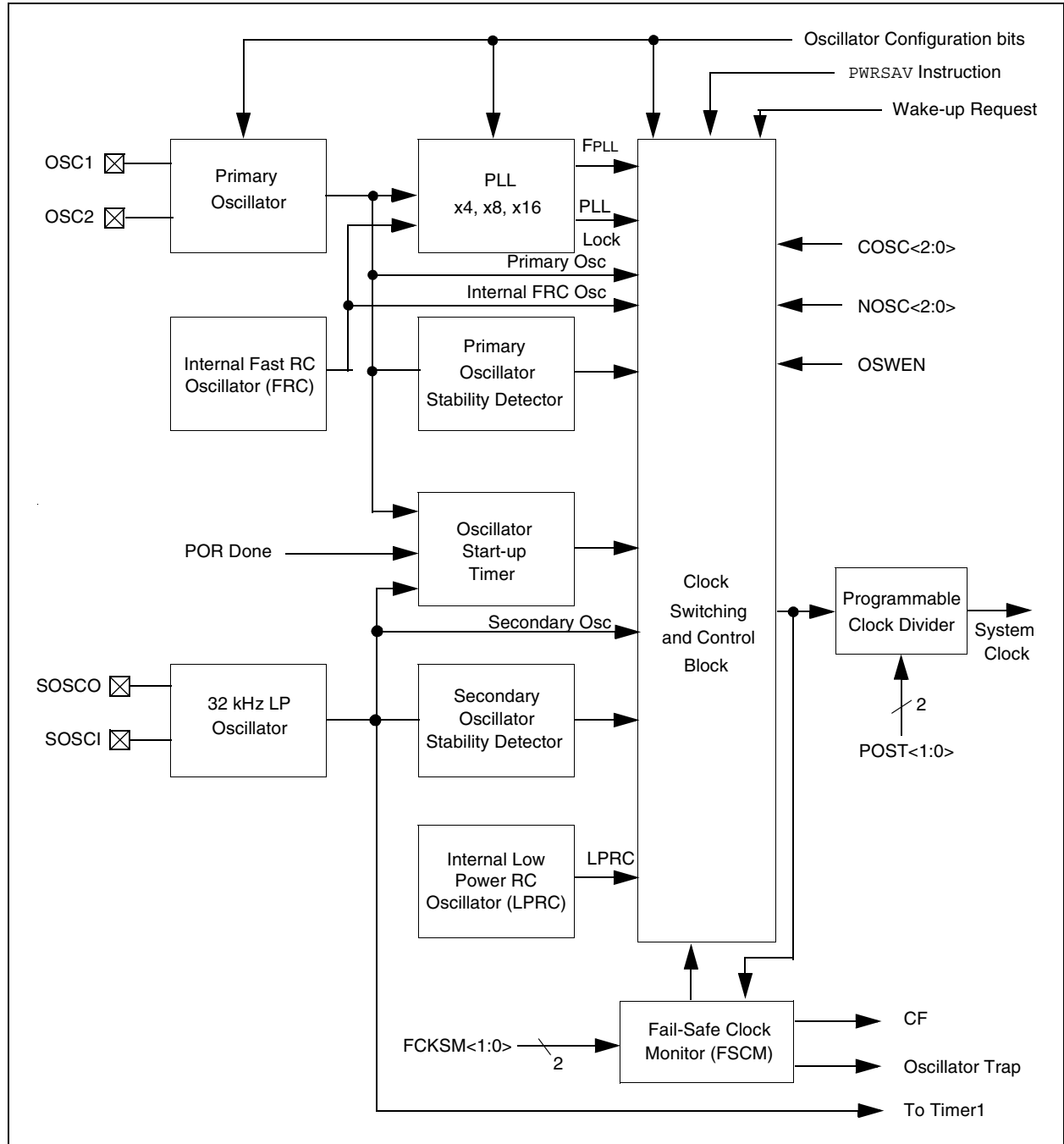
17.2   Oscillator Configurations

17.2.1   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<2:0> configuration bits that select one of four oscillator groups,

b) and FPR<4:0> configuration bits that select one of 15 oscillator choices within the primary group.

The selection is as shown in Table 17-2.

17.2.2   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 T_{OSC} cycles before releasing the oscillator clock to the rest of the system. The time-out period is designated as T_{OST}.

The T_{OST} 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.

TABLE 17-2:   CONFIGURATION BIT VALUES FOR CLOCK SELECTION

Oscillator Mode	Oscillator Source	FOS<2:0>			FPR<4:0>					OSC2 Function
ECIO w/ PLL 4x	PLL	1	1	1	0	1	1	0	1	I/O
ECIO w/ PLL 8x	PLL	1	1	1	0	1	1	1	0	I/O
ECIO w/ PLL 16x	PLL	1	1	1	0	1	1	1	1	I/O
FRC w/ PLL 4X	PLL	1	1	1	0	0	0	0	1	I/O
FRC w/ PLL 8x	PLL	1	1	1	0	1	0	1	0	I/O
FRC w/ PLL 16x	PLL	1	1	1	0	0	0	1	1	I/O
XT w/ PLL 4x	PLL	1	1	1	0	0	1	0	1	OSC2
XT w/ PLL 8x	PLL	1	1	1	0	0	1	1	0	OSC2
XT w/ PLL 16x	PLL	1	1	1	0	0	1	1	1	OSC2
HS2 w/ PLL 4x	PLL	1	1	1	1	0	0	0	1	OSC2
HS2 w/ PLL 8x	PLL	1	1	1	1	0	0	1	0	OSC2
HS2 w/ PLL 16x	PLL	1	1	1	1	0	0	1	1	OSC2
HS3 w/ PLL 4x	PLL	1	1	1	1	0	1	0	1	OSC2
HS3 w/ PLL 8x	PLL	1	1	1	1	0	1	1	0	OSC2
HS3 w/ PLL 16x	PLL	1	1	1	1	0	1	1	1	OSC2
ECIO	External	0	1	1	0	1	1	0	0	I/O
XT	External	0	1	1	0	0	1	0	0	OSC2
HS	External	0	1	1	0	0	0	1	0	OSC2
EC	External	0	1	1	0	1	0	1	1	CLKOUT
ERC	External	0	1	1	0	1	0	0	1	CLKOUT
ERCIO	External	0	1	1	0	1	0	0	0	I/O
XTL	External	0	1	1	0	0	0	0	0	OSC2
LP	Secondary	0	0	0	X	X	X	X	X	**(Note 1, 2)**
FRC	Internal FRC	0	0	1	X	X	X	X	X	**(Note 1, 2)**
LPRC	Internal LPRC	0	1	0	X	X	X	X	X	**(Note 1, 2)**

Note 1:   OSC2 pin function is determined by the Primary Oscillator mode selection (FPR<4:0>).

2:   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.

17.2.3 LP OSCILLATOR CONTROL

Enabling the LP oscillator is controlled with two elements:

1. The current oscillator group bits COSC<2:0>.
2. The LPOSCEN bit (OSCCON register).

The LP oscillator is on (even during Sleep mode) if LPOSCEN = 1. The LP oscillator is the device clock if:

- COSC<2:0> = 000 (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

17.2.4 PHASE LOCKED LOOP (PLL)

The PLL multiplies the clock which is generated by the primary oscillator or Fast RC oscillator. The PLL is selectable to have either gains of x4, x8, and x16. Input and output frequency ranges are summarized in Table 17-3.

TABLE 17-3: PLL FREQUENCY RANGE

F$_{IN}$	PLL Multiplier	F$_{OUT}$
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

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.

17.2.5 FAST RC OSCILLATOR (FRC)

The FRC oscillator is a fast (7.37 MHz 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 FRC oscillator can be used with the PLL to obtain higher clock frequencies.

The dsPIC30F operates from the FRC oscillator whenever the current oscillator selection control bits in the OSCCON register (OSCCON<14:12>) are set to '001'.

The four bit field specified by TUN<3:0> (OSCTUN <3:0>) allows the user to tune the internal fast RC oscillator (nominal 7.37 MHz). The user can tune the FRC oscillator within a range of +10.5% (840 kHz) and -12% (960 kHz) in steps of 1.50% around the factory-calibrated setting, see Table 17-4.

If OSCCON<14:12> are set to '111' and FPR<4:0> are set to '00001', '01010' or '00011', then a PLL multiplier of 4, 8 or 16 (respectively) is applied.

Note: When a 16x PLL is used, the FRC frequency must not be tuned to a frequency greater than 7.5 MHz.

TABLE 17-4: FRC TUNING

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%
1001	- 10.5%
1000	- 12.0%

17.2.6 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 low frequency 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<2:0> 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<4:0>).

2: 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.

17.2.7 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.

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<2:0> 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<14:12>) 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 (with or without PLL)
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<4:0> configuration bits.

The OSCCON register holds the control and status bits related to clock switching.

- COSC<2:0>: Read only status bits always reflect the current oscillator group in effect.
- NOSC<2:0>: Control bits which are written to indicate the new oscillator group of choice.
  - On POR and BOR, COSC<2:0> and NOSC<2:0> are both loaded with the configuration bit values FOS<2:0>.
- 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 monitoring functions are disabled. This is the default configuration bit setting.

If clock switching is disabled, then the FOS<2:0> and FPR<4:0> bits directly control the oscillator selection and the COSC<2:0> 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.

17.2.8 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.

17.3 Reset

The dsPIC30F2011/2012/3012/3013 differentiates between various kinds of Reset:

a) Power-on Reset (POR)
b) $\overline{MCLR}$ Reset during normal operation
c) $\overline{MCLR}$ Reset during Sleep
d) Watchdog Timer (WDT) Reset (during normal operation)
e) Programmable Brown-out Reset (BOR)
f) RESET Instruction
g) Reset caused by trap lockup (TRAPR)
h) 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 17-5. 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 17-2.

A $\overline{MCLR}$ noise filter is provided in the $\overline{MCLR}$ Reset path. The filter detects and ignores small pulses.

Internally generated Resets do not drive $\overline{MCLR}$ pin low.

FIGURE 17-2: RESET SYSTEM BLOCK DIAGRAM
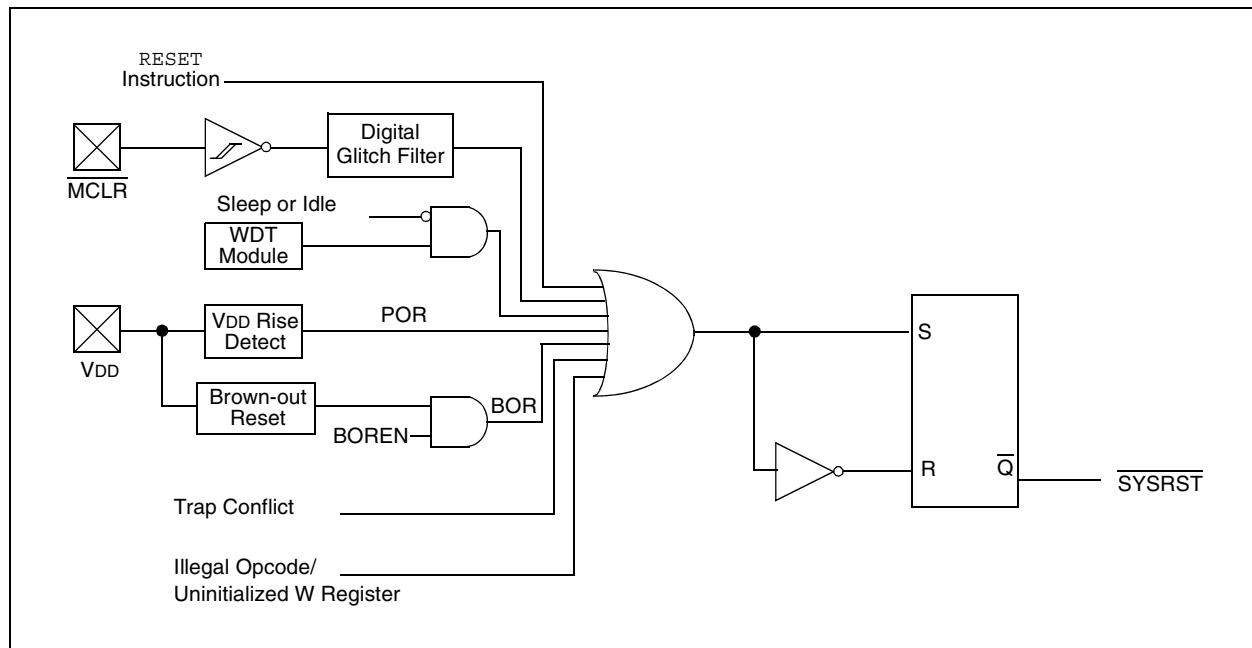
17.3.1 POR: POWER-ON RESET

A power-on event will generate an internal POR pulse when a V_{DD} rise is detected. The Reset pulse will occur at the POR circuit threshold voltage (V_{POR}) 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, T_{POR}, which is nominally 10 μs and ensures that the device bias circuits are stable. Furthermore, a user selected power-up time-out (T_{PWRT}) is applied. The T_{PWRT} 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, T_{POR} + T_{PWRT}. When these delays have expired, $\overline{SYSRST}$ 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 $\overline{SYSRST}$ signal is shown in Figure 17-3 through Figure 17-5.

FIGURE 17-3: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO V_DD)
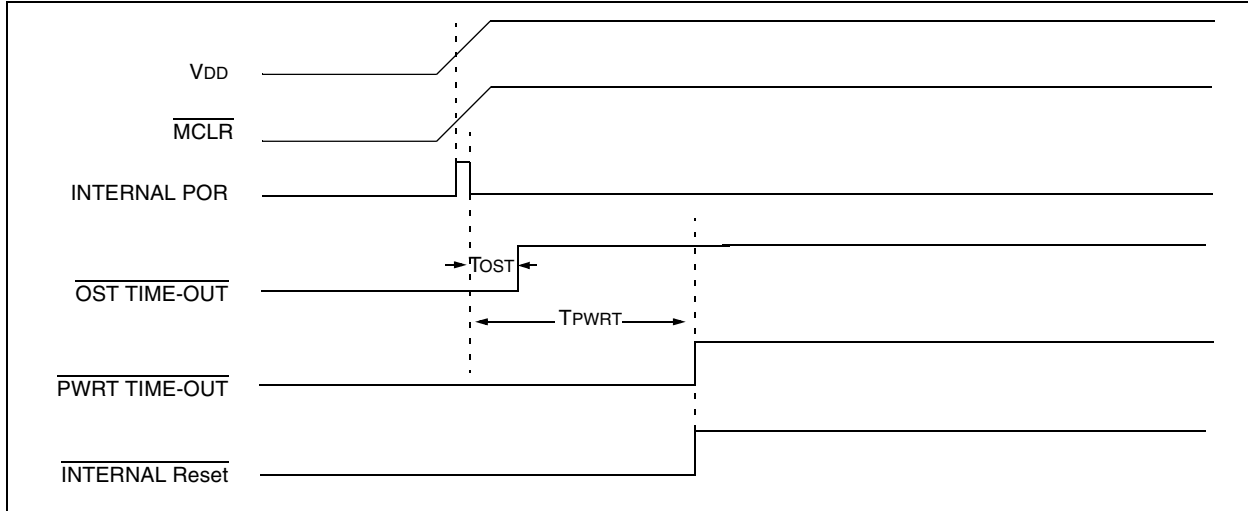
FIGURE 17-4: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO V_DD): CASE 1
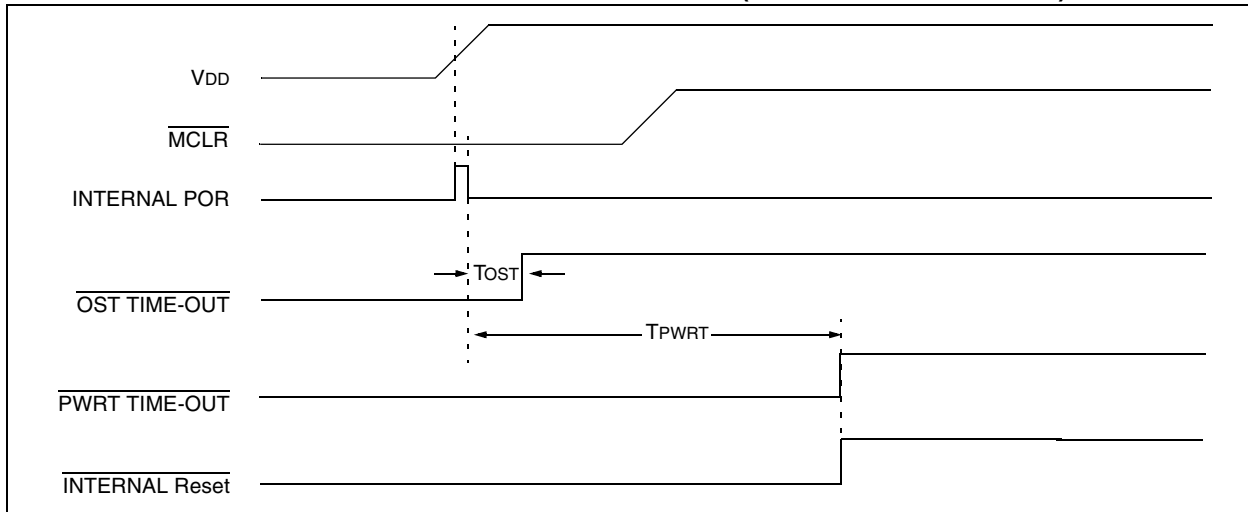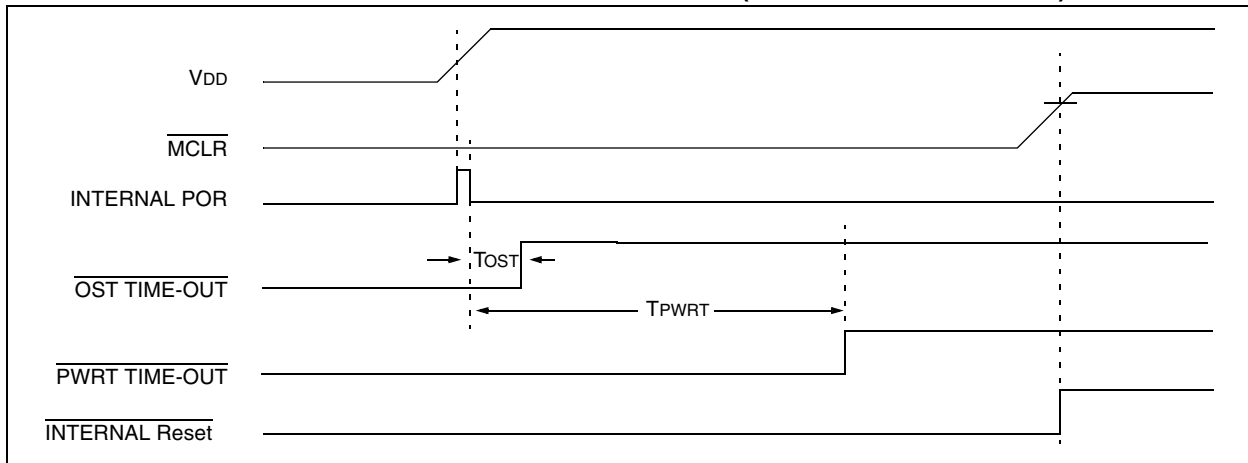

FIGURE 17-5: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO V_DD): CASE 2

Preliminary

17.3.1.1 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.

17.3.1.2 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.

17.3.2 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:
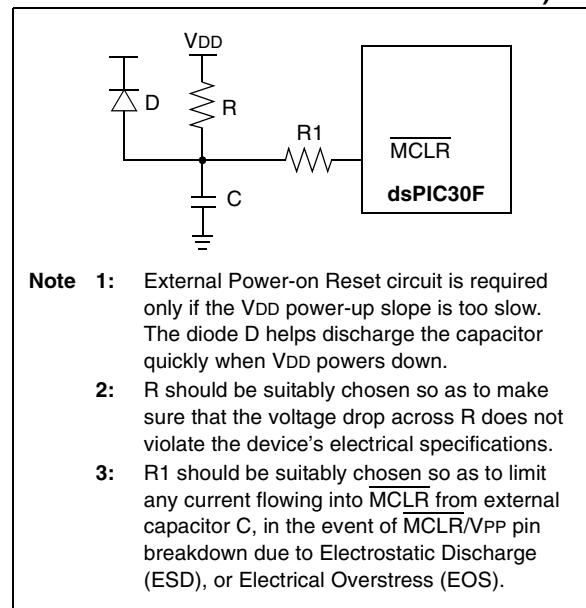
• 2.0V
• 2.7V
• 4.2V
• 4.5V

Note: The BOR voltage trip points indicated here are nominal values provided for design guidance only. Refer to the Electrical Specifications in the specific device data sheet for BOR voltage limit specifications.

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 (T_{POR}) and the PWRT time-out (T_{PWRT}) will be applied before the internal Reset is released. If $T_{PWRT} = 0$ and a crystal oscillator is being used, then a nominal delay of $T_{FSCM} = 100$ μs is applied. The total delay in this case is ($T_{POR} + T_{FSCM}$).

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 17-6: 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 specifications.

3: 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 17-5 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.

TABLE 17-5: INITIALIZATION CONDITION FOR RCON REGISTER: CASE 1

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[1]	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

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0'

Note 1: When the wake-up is due to an enabled interrupt, the PC is loaded with the corresponding interrupt vector.

Table 17-6 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 17-6: 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[1]	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, x = unknown, - = unimplemented bit, read as '0'

Note 1: When the wake-up is due to an enabled interrupt, the PC is loaded with the corresponding interrupt vector.

17.4   Watchdog Timer (WDT)

17.4.1   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.

17.4.2   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 wake-up. 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.

17.5   Low Voltage Detect

The Low Voltage Detect (LVD) module is used to detect when the V_{DD} of the device drops below a threshold value, V_{LVD}, 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>).

17.6   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.

17.6.1   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 $\overline{MCLR}$)
- 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<2:0> and FPR<4:0> 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, T_{POR}, T_{LOCK} and T_{PWRT} delays are applied.

If EC, FRC, LPRC or EXTRC oscillators are used, then a delay of T_{POR} (~ 10 μs) 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 T_{POR}. 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.

dsPIC30F2011/2012/3012/3013

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 the RCON register is set upon wake-up.

Note:	In spite of various delays applied (T$_{POR}$, T$_{LOCK}$ and T$_{PWRT}$), 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 the 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.

17.6.2 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 has occurred:

- any interrupt that is individually enabled (IE bit is '1') and meets the required priority level
- any Reset (POR, BOR, $\overline{MCLR}$)
- 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 the 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.

17.7 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 V$_{DD}$ ≥ 4.5V.

17.8 Peripheral Module Disable (PMD) Registers

The Peripheral Module Disable (PMD) registers provide a method to disable a peripheral module by stopping all clock sources supplied to that module. When a peripheral is disabled via the appropriate PMD control bit, the peripheral is in a minimum power consumption state. The control and status registers associated with the peripheral will also be disabled so writes to those registers will have no effect and read values will be invalid.

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

Note:	If a PMD bit is set, the corresponding module is disabled after a delay of 1 instruction cycle. Similarly, if a PMD bit is cleared, the corresponding module is enabled after a delay of 1 instruction cycle (assuming the module control registers are already configured to enable module operation).

Note:	In the dsPIC30F2011, dsPIC30F3012 and dsPIC30F2012 devices, the U2MD bit is readable and writable and will be read as '1' when set.

17.9 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 $\overline{MCLR}$, VDD, VSS, PGC, PGD, and the selected EMUDx/EMUCx pin pair.

This gives rise to two possibilities:

1. 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.

2. 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.

dsPIC30F2011/2012/3012/3013

TABLE 17-7: 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	Reset State
RCON	0740	TRAPR	IOPUWR	BGST	LVDEN	LVDL<3:0>				EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	(Note 1)
OSCCON	0742	—	COSC<2:0>			—	NOSC<2:0>			POST<1:0>		LOCK	—	CF	—	LPOSCEN	OSWEN	(Note 2)
OSCTUN	0744	—	—	—	—	—	—	—	—	—	—	—	—	TUN3	TUN2	TUN1	TUN0	(Note 2)
PMD1	0770	—	—	T3MD	T2MD	T1MD	—	—	—	I2CMD	U2MD(3)	U1MD	—	SPI1MD	—	—	ADCMD	0000 0000 0000 0000
PMD2	0772	—	—	—	—	—	—	IC2MD	IC1MD	—	—	—	—	—	—	OC2MD	OC1MD	0000 0000 0000 0000

Note 1: Reset state depends on type of Reset.
2: Reset state depends on configuration bits.
3: Only available on dsPIC30F3013.

TABLE 17-8: 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	Bit 3	Bit 2	Bit 1	Bit 0
FOSC	F80000	—	FCKSM<1:0>		—	—	—	FOS<2:0>			—	—	—	FPR<4:0>				
FWDT	F80002	—	FWDTEN	—	—	—	—	—	—	—	—	—	FWPSA<1:0>		FWPSB<3:0>			
FBORPOR	F80004	—	MCLREN	—	—	—	—	Reserved(1)	Reserved(1)	Reserved(1)	BOREN	BORV<1:0>		—	—	—	Reserved(1)	FPWRT<1:0>	
FGS	F8000A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Reserved(1)	GCP	GWRP

Note 1: Always reads as '1'.

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

Preliminary

18.0 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 Programmer's Reference Manual* (DS70030).

The dsPIC30F instruction set adds many enhancements to the previous PICmicro® instruction sets, while maintaining an easy migration from PICmicro 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 18-1 shows the general symbols used in describing the instructions.

The dsPIC30F instruction set summary in Table 18-2 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 pre-fetch operations
- The X and Y address space pre-fetch destinations
- The accumulator write back destination

The other DSP instructions do not involve any multiplication, and may include:

- The accumulator to be used (required)
- The source or destination operand (designated as Wso or Wdo, respectively) with or without an address modifier
- The amount of shift specified by a W register 'Wn' or a literal value

The control instructions may use some of the following operands:

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

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

Most single word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles with the additional instruction cycle(s) executed as a NOP. Notable exceptions are the BRA (unconditional/computed branch), indirect CALL/GOTO, all table reads and writes, and RETURN/RETFIE instructions,

which are single word instructions but take two or three cycles. Certain instructions that involve skipping over the subsequent instruction require either two or three cycles if the skip is performed, depending on whether the instruction being skipped is a single word or two-word instruction. Moreover, double-word moves require two cycles. The double-word instructions execute in two instruction cycles.

> **Note:** For more details on the instruction set, refer to the Programmer's Reference Manual.

TABLE 18-1: SYMBOLS USED IN OPCODE DESCRIPTIONS

Field	Description
#text	Means literal defined by "text"
(text)	Means "content of text"
[text]	Means "the location addressed by text"
{ }	Optional field or operation
<n:m>	Register bit field
.b	Byte mode selection
.d	Double-Word mode selection
.S	Shadow register select
.w	Word mode selection (default)
Acc	One of two accumulators {A, B}
AWB	Accumulator write back destination address register $\in$ {W13, [W13]+=2}
bit4	4-bit bit selection field (used in word addressed instructions) $\in$ {0...15}
C, DC, N, OV, Z	MCU status bits: Carry, Digit Carry, Negative, Overflow, Sticky Zero
Expr	Absolute address, label or expression (resolved by the linker)
f	File register address $\in$ {0x0000...0x1FFF}
lit1	1-bit unsigned literal $\in$ {0,1}
lit4	4-bit unsigned literal $\in$ {0...15}
lit5	5-bit unsigned literal $\in$ {0...31}
lit8	8-bit unsigned literal $\in$ {0...255}
lit10	10-bit unsigned literal $\in$ {0...255} for Byte mode, {0:1023} for Word mode
lit14	14-bit unsigned literal $\in$ {0...16384}
lit16	16-bit unsigned literal $\in$ {0...65535}
lit23	23-bit unsigned literal $\in$ {0...8388608}; LSB must be 0
None	Field does not require an entry, 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 $\in$ {-512...511}
Slit16	16-bit signed literal $\in$ {-32768...32767}
Slit6	6-bit signed literal $\in$ {-16...16}

TABLE 18-1: **SYMBOLS USED IN OPCODE DESCRIPTIONS (CONTINUED)**

Field	Description
Wb	Base W register ∈ {W0..W15}
Wd	Destination W register ∈ { Wd, [Wd], [Wd++], [Wd--], [++Wd], [--Wd] }
Wdo	Destination W register ∈ { Wnd, [Wnd], [Wnd++], [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb] }
Wm,Wn	Dividend, Divisor working register pair (direct addressing)
Wm*Wm	Multiplicand and Multiplier working register pair for Square instructions ∈ {W4*W4,W5*W5,W6*W6,W7*W7}
Wm*Wn	Multiplicand and Multiplier working register pair for DSP instructions ∈ {W4*W5,W4*W6,W4*W7,W5*W6,W5*W7,W6*W7}
Wn	One of 16 working registers ∈ {W0..W15}
Wnd	One of 16 destination working registers ∈ {W0..W15}
Wns	One of 16 source working registers ∈ {W0..W15}
WREG	W0 (working register used in file register instructions)
Ws	Source W register ∈ { Ws, [Ws], [Ws++], [Ws--], [++Ws], [--Ws] }
Wso	Source W register ∈ { Wns, [Wns], [Wns++], [Wns--], [++Wns], [--Wns], [Wns+Wb] }

Wx	X data space pre-fetch 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 pre-fetch destination register for DSP instructions ∈ {W4..W7}

Wy	Y data space pre-fetch 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 pre-fetch destination register for DSP instructions ∈ {W4..W7}

TABLE 18-2: INSTRUCTION SET OVERVIEW

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
1	ADD	ADD	Acc	Add Accumulators	1	1	OA,OB,SA,SB
		ADD	f	f = f + WREG	1	1	C,DC,N,OV,Z
		ADD	f,WREG	WREG = f + WREG	1	1	C,DC,N,OV,Z
		ADD	#lit10,Wn	Wd = lit10 + Wd	1	1	C,DC,N,OV,Z
		ADD	Wb,Ws,Wd	Wd = Wb + Ws	1	1	C,DC,N,OV,Z
		ADD	Wb,#lit5,Wd	Wd = Wb + lit5	1	1	C,DC,N,OV,Z
		ADD	Wso,#Slit4,Acc	16-bit Signed Add to Accumulator	1	1	OA,OB,SA,SB
2	ADDC	ADDC	f	f = f + WREG + (C)	1	1	C,DC,N,OV,Z
		ADDC	f,WREG	WREG = f + WREG + (C)	1	1	C,DC,N,OV,Z
		ADDC	#lit10,Wn	Wd = lit10 + Wd + (C)	1	1	C,DC,N,OV,Z
		ADDC	Wb,Ws,Wd	Wd = Wb + Ws + (C)	1	1	C,DC,N,OV,Z
		ADDC	Wb,#lit5,Wd	Wd = Wb + lit5 + (C)	1	1	C,DC,N,OV,Z
3	AND	AND	f	f = f .AND. WREG	1	1	N,Z
		AND	f,WREG	WREG = f .AND. WREG	1	1	N,Z
		AND	#lit10,Wn	Wd = lit10 .AND. Wd	1	1	N,Z
		AND	Wb,Ws,Wd	Wd = Wb .AND. Ws	1	1	N,Z
		AND	Wb,#lit5,Wd	Wd = Wb .AND. lit5	1	1	N,Z
4	ASR	ASR	f	f = Arithmetic Right Shift f	1	1	C,N,OV,Z
		ASR	f,WREG	WREG = Arithmetic Right Shift f	1	1	C,N,OV,Z
		ASR	Ws,Wd	Wd = Arithmetic Right Shift Ws	1	1	C,N,OV,Z
		ASR	Wb,Wns,Wnd	Wnd = Arithmetic Right Shift Wb by Wns	1	1	N,Z
		ASR	Wb,#lit5,Wnd	Wnd = Arithmetic Right Shift Wb by lit5	1	1	N,Z
5	BCLR	BCLR	f,#bit4	Bit Clear f	1	1	None
		BCLR	Ws,#bit4	Bit Clear Ws	1	1	None
6	BRA	BRA	C,Expr	Branch if Carry	1	1 (2)	None
		BRA	GE,Expr	Branch if greater than or equal	1	1 (2)	None
		BRA	GEU,Expr	Branch if unsigned greater than or equal	1	1 (2)	None
		BRA	GT,Expr	Branch if greater than	1	1 (2)	None
		BRA	GTU,Expr	Branch if unsigned greater than	1	1 (2)	None
		BRA	LE,Expr	Branch if less than or equal	1	1 (2)	None
		BRA	LEU,Expr	Branch if unsigned less than or equal	1	1 (2)	None
		BRA	LT,Expr	Branch if less than	1	1 (2)	None
		BRA	LTU,Expr	Branch if unsigned less than	1	1 (2)	None
		BRA	N,Expr	Branch if Negative	1	1 (2)	None
		BRA	NC,Expr	Branch if Not Carry	1	1 (2)	None
		BRA	NN,Expr	Branch if Not Negative	1	1 (2)	None
		BRA	NOV,Expr	Branch if Not Overflow	1	1 (2)	None
		BRA	NZ,Expr	Branch if Not Zero	1	1 (2)	None
		BRA	OA,Expr	Branch if Accumulator A overflow	1	1 (2)	None
		BRA	OB,Expr	Branch if Accumulator B overflow	1	1 (2)	None
		BRA	OV,Expr	Branch if Overflow	1	1 (2)	None
		BRA	SA,Expr	Branch if Accumulator A saturated	1	1 (2)	None
		BRA	SB,Expr	Branch if Accumulator B saturated	1	1 (2)	None
		BRA	Expr	Branch Unconditionally	1	2	None
		BRA	Z,Expr	Branch if Zero	1	1 (2)	None
		BRA	Wn	Computed Branch	1	2	None
7	BSET	BSET	f,#bit4	Bit Set f	1	1	None
		BSET	Ws,#bit4	Bit Set Ws	1	1	None
8	BSW	BSW.C	Ws,Wb	Write C bit to Ws<Wb>	1	1	None
		BSW.Z	Ws,Wb	Write Z bit to Ws<Wb>	1	1	None

TABLE 18-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	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	C
		BTST.Z	Ws,#bit4	Bit Test Ws to Z	1	1	Z
		BTST.C	Ws,Wb	Bit Test Ws<Wb> to C	1	1	C
		BTST.Z	Ws,Wb	Bit Test Ws<Wb> to Z	1	1	Z
13	BTSTS	BTSTS	f,#bit4	Bit Test then Set f	1	1	Z
		BTSTS.C	Ws,#bit4	Bit Test Ws to C, then Set	1	1	C
		BTSTS.Z	Ws,#bit4	Bit Test Ws to Z, then Set	1	1	Z
14	CALL	CALL	lit23	Call subroutine	2	2	None
		CALL	Wn	Call indirect subroutine	1	2	None
15	CLR	CLR	f	f = 0x0000	1	1	None
		CLR	WREG	WREG = 0x0000	1	1	None
		CLR	Ws	Ws = 0x0000	1	1	None
		CLR	Acc,Wx,Wxd,Wy,Wyd,AWB	Clear Accumulator	1	1	OA,OB,SA,SB
16	CLRWDT	CLRWDT		Clear Watchdog Timer	1	1	WDTO,Sleep
17	COM	COM	f	f = $\bar{f}$	1	1	N,Z
		COM	f,WREG	WREG = $\bar{f}$	1	1	N,Z
		COM	Ws,Wd	Wd = $\overline{Ws}$	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	CP1	CP1	f	Compare f with 0xFFFF	1	1	C,DC,N,OV,Z
		CP1	Ws	Compare Ws with 0xFFFF	1	1	C,DC,N,OV,Z
21	CPB	CPB	f	Compare f with WREG, with Borrow	1	1	C,DC,N,OV,Z
		CPB	Wb,#lit5	Compare Wb with lit5, with Borrow	1	1	C,DC,N,OV,Z
		CPB	Wb,Ws	Compare Wb with Ws, with Borrow (Wb - Ws - $\overline{C}$)	1	1	C,DC,N,OV,Z
22	CPSEQ	CPSEQ	Wb, Wn	Compare Wb with Wn, skip if =	1	1 (2 or 3)	None
23	CPSGT	CPSGT	Wb, Wn	Compare Wb with Wn, skip if >	1	1 (2 or 3)	None
24	CPSLT	CPSLT	Wb, Wn	Compare Wb with Wn, skip if <	1	1 (2 or 3)	None
25	CPSNE	CPSNE	Wb, Wn	Compare Wb with Wn, skip if ≠	1	1 (2 or 3)	None
26	DAW	DAW	Wn	Wn = decimal adjust Wn	1	1	C
27	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
28	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

TABLE 18-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
29	DISI	DISI	#lit14	Disable Interrupts for k instruction cycles	1	1	None
30	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
31	DIVF	DIVF	Wm,Wn	Signed 16/16-bit Fractional Divide	1	18	N,Z,C,OV
32	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
33	ED	ED	Wm*Wm,Acc,Wx,Wy,Wxd	Euclidean Distance (no accumulate)	1	1	OA,OB,OAB, SA,SB,SAB
34	EDAC	EDAC	Wm*Wm,Acc,Wx,Wy,Wxd	Euclidean Distance	1	1	OA,OB,OAB, SA,SB,SAB
35	EXCH	EXCH	Wns,Wnd	Swap Wns with Wnd	1	1	None
36	FBCL	FBCL	Ws,Wnd	Find Bit Change from Left (MSb) Side	1	1	C
37	FF1L	FF1L	Ws,Wnd	Find First One from Left (MSb) Side	1	1	C
38	FF1R	FF1R	Ws,Wnd	Find First One from Right (LSb) Side	1	1	C
39	GOTO	GOTO	Expr	Go to address	2	2	None
		GOTO	Wn	Go to indirect	1	2	None
40	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
41	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
42	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
43	LAC	LAC	Wso,#Slit4,Acc	Load Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
44	LNK	LNK	#lit14	Link frame pointer	1	1	None
45	LSR	LSR	f	f = Logical Right Shift f	1	1	C,N,OV,Z
		LSR	f,WREG	WREG = Logical Right Shift f	1	1	C,N,OV,Z
		LSR	Ws,Wd	Wd = Logical Right Shift Ws	1	1	C,N,OV,Z
		LSR	Wb,Wns,Wnd	Wnd = Logical Right Shift Wb by Wns	1	1	N,Z
		LSR	Wb,#lit5,Wnd	Wnd = Logical Right Shift Wb by lit5	1	1	N,Z
46	MAC	MAC	Wm*Wn,Acc,Wx,Wxd,Wy,Wyd, AWB	Multiply and Accumulate	1	1	OA,OB,OAB, SA,SB,SAB
		MAC	Wm*Wm,Acc,Wx,Wxd,Wy,Wyd	Square and Accumulate	1	1	OA,OB,OAB, SA,SB,SAB
47	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
48	MOVSAC	MOVSAC	Acc,Wx,Wxd,Wy,Wyd,AWB	Pre-fetch and store accumulator	1	1	None

TABLE 18-2:   INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
49	MPY	MPY	Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	Multiply Wm by Wn to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
		MPY	Wm*Wm,Acc,Wx,Wxd,Wy,Wyd	Square Wm to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
50	MPY.N	MPY.N	Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	-(Multiply Wm by Wn) to Accumulator	1	1	None
51	MSC	MSC	Wm*Wm,Acc,Wx,Wxd,Wy,Wyd, AWB	Multiply and Subtract from Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
52	MUL	MUL.SS	Wb,Ws,Wnd	{Wnd+1, Wnd} = signed(Wb) * signed(Ws)	1	1	None
		MUL.SU	Wb,Ws,Wnd	{Wnd+1, Wnd} = signed(Wb) * unsigned(Ws)	1	1	None
		MUL.US	Wb,Ws,Wnd	{Wnd+1, Wnd} = unsigned(Wb) * signed(Ws)	1	1	None
		MUL.UU	Wb,Ws,Wnd	{Wnd+1, Wnd} = unsigned(Wb) * unsigned(Ws)	1	1	None
		MUL.SU	Wb,#lit5,Wnd	{Wnd+1, Wnd} = signed(Wb) * unsigned(lit5)	1	1	None
		MUL.UU	Wb,#lit5,Wnd	{Wnd+1, Wnd} = unsigned(Wb) * unsigned(lit5)	1	1	None
		MUL	f	W3:W2 = f * WREG	1	1	None
53	NEG	NEG	Acc	Negate Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
		NEG	f	$f = \bar{f} + 1$	1	1	C,DC,N,OV,Z
		NEG	f,WREG	$WREG = \bar{f} + 1$	1	1	C,DC,N,OV,Z
		NEG	Ws,Wd	$Wd = \overline{Ws} + 1$	1	1	C,DC,N,OV,Z
54	NOP	NOP		No Operation	1	1	None
		NOPR		No Operation	1	1	None
55	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
56	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
57	PWRSAV	PWRSAV	#lit1	Go into Sleep or Idle mode	1	1	WDTO,Sleep
58	RCALL	RCALL	Expr	Relative Call	1	2	None
		RCALL	Wn	Computed Call	1	2	None
59	REPEAT	REPEAT	#lit14	Repeat Next Instruction lit14+1 times	1	1	None
		REPEAT	Wn	Repeat Next Instruction (Wn)+1 times	1	1	None
60	RESET	RESET		Software device Reset	1	1	None
61	RETFIE	RETFIE		Return from interrupt	1	3 (2)	None
62	RETLW	RETLW	#lit10,Wn	Return with literal in Wn	1	3 (2)	None
63	RETURN	RETURN		Return from Subroutine	1	3 (2)	None
64	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
65	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
66	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
67	RRNC	RRNC	f	f = Rotate Right (No Carry) f	1	1	N,Z
		RRNC	f,WREG	WREG = Rotate Right (No Carry) f	1	1	N,Z
		RRNC	Ws,Wd	Wd = Rotate Right (No Carry) Ws	1	1	N,Z

TABLE 18-2:   INSTRUCTION SET OVERVIEW (CONTINUED)

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

19.0 DEVELOPMENT SUPPORT

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

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C17 and MPLAB C18 C Compilers
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB C30 C Compiler
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
  - MPLAB dsPIC30 Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB ICE 4000 In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PRO MATE® II Universal Device Programmer
  - PICSTART® Plus Development Programmer
- Low-Cost Demonstration Boards
  - PICDEM™ 1 Demonstration Board
  - PICDEM.net™ Demonstration Board
  - PICDEM 2 Plus Demonstration Board
  - PICDEM 3 Demonstration Board
  - PICDEM 4 Demonstration Board
  - PICDEM 17 Demonstration Board
  - PICDEM 18R Demonstration Board
  - PICDEM LIN Demonstration Board
  - PICDEM USB Demonstration Board
- Evaluation Kits
  - KEELOQ®
  - PICDEM MSC
  - microID®
  - CAN
  - PowerSmart®
  - Analog

19.1 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® based application that contains:

- An interface to 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
- Mouse over variable inspection
- Extensive on-line help

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PICmicro emulator and simulator tools (automatically updates all project information)
- Debug using:
  - source files (assembly or C)
  - absolute listing file (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 increasing flexibility and power.

19.2 MPASM Assembler

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

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

The MPASM assembler features include:

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

19.3 MPLAB C17 and MPLAB C18 C Compilers

The MPLAB C17 and MPLAB C18 Code Development Systems are complete ANSI C compilers for Microchip's PIC17CXXX and PIC18CXXX family of microcontrollers. 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.

19.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. 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

19.5 MPLAB C30 C Compiler

The MPLAB C30 C compiler is a full-featured, ANSI compliant, optimizing compiler that translates standard ANSI C programs into dsPIC30F assembly language source. The compiler also supports many command line options and language extensions to take full advantage of the dsPIC30F device hardware capabilities and afford fine control of the compiler code generator.

MPLAB C30 is distributed with a complete ANSI C standard library. All library functions have been validated and conform to the ANSI C library standard. The library includes functions for string manipulation, dynamic memory allocation, data conversion, timekeeping and math functions (trigonometric, exponential and hyperbolic). The compiler provides symbolic information for high-level source debugging with the MPLAB IDE.

19.6 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 compiler uses the assembler to produce it's 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

19.7 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC hosted environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any pin. The execution can be performed in Single-Step, Execute Until Break or Trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and MPLAB C18 C Compilers, as well as the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent, economical software development tool.

19.8 MPLAB SIM30 Software Simulator

The MPLAB SIM30 software simulator allows code development in a PC hosted environment by simulating the dsPIC30F series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any of the pins.

The MPLAB SIM30 simulator fully supports symbolic debugging using the MPLAB C30 C Compiler and MPLAB ASM30 assembler. The simulator runs in either a Command Line mode for automated tasks, or from MPLAB IDE. This high-speed simulator is designed to debug, analyze and optimize time intensive DSP routines.

19.9 MPLAB ICE 2000 High-Performance Universal In-Circuit Emulator

The MPLAB ICE 2000 universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PICmicro 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 universal architecture of the MPLAB ICE in-circuit emulator allows expansion to support new PICmicro microcontrollers.

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

19.10 MPLAB ICE 4000 High-Performance Universal In-Circuit Emulator

The MPLAB ICE 4000 universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for high-end PICmicro microcontrollers. Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICD 4000 is a premium emulator system, providing the features of MPLAB ICE 2000, but with increased emulation memory and high-speed performance for dsPIC30F and PIC18XXXX devices. Its advanced emulator features include complex triggering and timing, up to 2 Mb of emulation memory and the ability to view variables in real-time.

The MPLAB ICE 4000 in-circuit emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft Windows 32-bit operating system were chosen to best make these features available in a simple, unified application.

19.11 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 PICmicro MCUs and can be used to develop for these and other PICmicro microcontrollers. The MPLAB ICD 2 utilizes the in-circuit debugging capability built into the Flash devices. This feature, along with Microchip's In-Circuit Serial Programming™ (ICSP™) protocol, offers cost effective in-circuit Flash debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by setting breakpoints, single-stepping and watching variables, 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 PICmicro devices.

19.12 PRO MATE II Universal Device Programmer

The PRO MATE II is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features an LCD display for instructions and error messages and a modular detachable socket assembly to support various package types. In Stand-Alone mode, the PRO MATE II device programmer can read, verify and program PICmicro devices without a PC connection. It can also set code protection in this mode.

19.13 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 PICmicro devices 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.

19.14 PICDEM 1 PICmicro Demonstration Board

The PICDEM 1 demonstration board demonstrates the capabilities of the PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The sample microcontrollers provided with the PICDEM 1 demonstration board can be programmed with a PRO MATE II device programmer or a PICSTART Plus development programmer. The PICDEM 1 demonstration board can be connected to the MPLAB ICE in-circuit emulator for testing. A prototype area extends the circuitry for additional application components. Features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs.

19.15 PICDEM.net Internet/Ethernet Demonstration Board

The PICDEM.net demonstration board is an Internet/Ethernet demonstration board using the PIC18F452 microcontroller and TCP/IP firmware. The board supports any 40-pin DIP device that conforms to the standard pinout used by the PIC16F877 or PIC18C452. This kit features a user friendly TCP/IP stack, web server with HTML, a 24L256 Serial EEPROM for Xmodem download to web pages into Serial EEPROM, ICSP/MPLAB ICD 2 interface connector, an Ethernet interface, RS-232 interface and a 16 x 2 LCD display. Also included is the book and CD-ROM *"TCP/IP Lean, Web Servers for Embedded Systems,"* by Jeremy Bentham

19.16 PICDEM 2 Plus Demonstration Board

The PICDEM 2 Plus demonstration board supports many 18, 28 and 40-pin microcontrollers, including PIC16F87X and PIC18FXX2 devices. All the necessary hardware and software is included to run the demonstration programs. The sample microcontrollers provided with the PICDEM 2 demonstration board can be programmed with a PRO MATE II device programmer, PICSTART Plus development programmer, or MPLAB ICD 2 with a Universal Programmer Adapter. The MPLAB ICD 2 and MPLAB ICE in-circuit emulators may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area extends the circuitry for additional application components. Some of the features include an RS-232 interface, a 2 x 16 LCD display, a piezo speaker, an on-board temperature sensor, four LEDs and sample PIC18F452 and PIC16F877 Flash microcontrollers.

19.17 PICDEM 3 PIC16C92X Demonstration Board

The PICDEM 3 demonstration board supports the PIC16C923 and PIC16C924 in the PLCC package. All the necessary hardware and software is included to run the demonstration programs.

19.18 PICDEM 4 8/14/18-Pin Demonstration Board

The PICDEM 4 can be used to demonstrate the capabilities of the 8, 14 and 18-pin PIC16XXXX and PIC18XXXX MCUs, including the PIC16F818/819, PIC16F87/88, PIC16F62XA and the PIC18F1320 family of microcontrollers. PICDEM 4 is intended to showcase the many features of these low pin count parts, including LIN and Motor Control using ECCP. Special provisions are made for low-power operation with the supercapacitor circuit and jumpers allow on-board hardware to be disabled to eliminate current draw in this mode. Included on the demo board are provisions for Crystal, RC or Canned Oscillator modes, a five volt regulator for use with a nine volt wall adapter or battery, DB-9 RS-232 interface, ICD connector for programming via ICSP and development with MPLAB ICD 2, 2 x 16 liquid crystal display, PCB footprints for H-Bridge motor driver, LIN transceiver and EEPROM. Also included are: header for expansion, eight LEDs, four potentiometers, three push buttons and a prototyping area. Included with the kit is a PIC16F627A and a PIC18F1320. Tutorial firmware is included along with the User's Guide.

19.19 PICDEM 17 Demonstration Board

The PICDEM 17 demonstration board is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756A, PIC17C762 and PIC17C766. A programmed sample is included. The PRO MATE II device programmer, or the PICSTART Plus development programmer, can be used to reprogram the device for user tailored application development. The PICDEM 17 demonstration board supports program download and execution from external on-board Flash memory. A generous prototype area is available for user hardware expansion.

19.20 PICDEM 18R PIC18C601/801 Demonstration Board

The PICDEM 18R demonstration board serves to assist development of the PIC18C601/801 family of Microchip microcontrollers. It provides hardware implementation of both 8-bit Multiplexed/Demultiplexed and 16-bit Memory modes. The board includes 2 Mb external Flash memory and 128 Kb SRAM memory, as well as serial EEPROM, allowing access to the wide range of memory types supported by the PIC18C601/801.

19.21 PICDEM LIN PIC16C43X Demonstration Board

The powerful LIN hardware and software kit includes a series of boards and three PICmicro microcontrollers. The small footprint PIC16C432 and PIC16C433 are used as slaves in the LIN communication and feature on-board LIN transceivers. A PIC16F874 Flash microcontroller serves as the master. All three micro-controllers are programmed with firmware to provide LIN bus communication.

19.22 PICkit™ 1 Flash Starter Kit

A complete "development system in a box", the PICkit Flash Starter Kit includes a convenient multi-section board for programming, evaluation and development of 8/14-pin Flash PIC® microcontrollers. Powered via USB, the board operates under a simple Windows GUI. The PICkit 1 Starter Kit includes the User's Guide (on CD ROM), PICkit 1 tutorial software and code for various applications. Also included are MPLAB® IDE (Integrated Development Environment) software, software and hardware "Tips 'n Tricks for 8-pin Flash PIC® Microcontrollers" Handbook and a USB interface cable. Supports all current 8/14-pin Flash PIC microcontrollers, as well as many future planned devices.

19.23 PICDEM USB PIC16C7X5 Demonstration Board

The PICDEM USB Demonstration Board shows off the capabilities of the PIC16C745 and PIC16C765 USB microcontrollers. This board provides the basis for future USB products.

19.24 Evaluation and Programming Tools

In addition to the PICDEM series of circuits, Microchip has a line of evaluation kits and demonstration software for these products.

- KEELOQ evaluation and programming tools for Microchip's HCS Secure Data Products
- CAN developers kit for automotive network applications
- Analog design boards and filter design software
- PowerSmart battery charging evaluation/calibration kits
- IrDA® development kit
- microID development and rfLab™ development software
- SEEVAL® designer kit for memory evaluation and endurance calculations
- PICDEM MSC demo boards for Switching mode power supply, high-power IR driver, delta sigma ADC and flow rate sensor

Check the Microchip web page and the latest Product Selector Guide for the complete list of demonstration and evaluation kits.

NOTES:

20.0 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[(†)]

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 $\overline{MCLR}$) **(Note 1)**................................... -0.3V to (VDD + 0.3V)
Voltage on VDD with respect to Vss .... -0.3V to +5.5V
Voltage on $\overline{MCLR}$ with respect to Vss... 0V to +13.25V
Maximum current out of Vss pin ....300 mA
Maximum current into VDD pin **(Note 2)**...250 mA
Input clamp current, IIK (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...25 mA
Maximum output current sourced by any I/O pin ....25 mA
Maximum current sunk by all ports ....200 mA
Maximum current sourced by all ports **(Note 2)**..200 mA

Note 1: Voltage spikes below Vss at the $\overline{MCLR}$/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 $\overline{MCLR}$/VPP pin, rather than pulling this pin directly to Vss.

2: Maximum allowable current is a function of device maximum power dissipation. See Table 20-2 for PDMAX.

[†]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.

Note: All peripheral electrical characteristics are specified. For exact peripherals available on specific devices, please refer the the Family Cross Reference Table.

20.1 DC Characteristics

TABLE 20-1: OPERATING MIPS VS. VOLTAGE

VDD Range	Temp Range	Max MIPS		
		dsPIC30FXXX-30I	dsPIC30FXXX-20I	dsPIC30FXXX-20E
4.5-5.5V	-40°C to 85°C	30	20	—
4.5-5.5V	-40°C to 125°C	—	—	20
3.0-3.6V	-40°C to 85°C	20	15	—
3.0-3.6V	-40°C to 125°C	—	—	15
2.5-3.0V	-40°C to 85°C	10	7.5	—

TABLE 20-2: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min	Typ	Max	Unit
dsPIC30F201x-30I dsPIC30F301x-30I					
Operating Junction Temperature Range	T_J	-40		+125	°C
Operating Ambient Temperature Range	T_A	-40		+85	°C
dsPIC30F201x-20I dsPIC30F301x-20I					
Operating Junction Temperature Range	T_J	-40		+150	°C
Operating Ambient Temperature Range	T_A	-40		+85	°C
dsPIC30F201x-20E dsPIC30F301x-20E					
Operating Junction Temperature Range	T_J	-40		+150	°C
Operating Ambient Temperature Range	T_A	-40		+125	°C

Power Dissipation:
Internal chip power dissipation:
    $P_{INT} = V_{DD} \times (I_{DD} - \sum I_{OH})$
I/O Pin power dissipation:
    $P_{I/O} = \sum (\{ V_{DD} - V_{OH} \} \times I_{OH}) + \sum (V_{OL} \times I_{OL})$
	P_D	$P_{INT} + P_{I/O}$			W

Maximum Allowed Power Dissipation	P_{DMAX}	$(T_J - T_A) / \theta_{JA}$			W

TABLE 20-3: THERMAL PACKAGING CHARACTERISTICS

Characteristic	Symbol	Typ	Max	Unit	Notes
Package Thermal Resistance, 18-pin PDIP (P)	θ_{JA}	44	—	°C/W	1
Package Thermal Resistance, 18-pin SOIC (SO)	θ_{JA}	57	—	°C/W	1
Package Thermal Resistance, 28-pin SPDIP (SP)	θ_{JA}	42	—	°C/W	1
Package Thermal Resistance, 28-pin (SOIC)	θ_{JA}	49	—	°C/W	1
Package Thermal Resistance, 44-pin QFN	θ_{JA}	28	—	°C/W	1

Note 1: Junction to ambient thermal resistance, Theta-ja (θ_{JA}) numbers are achieved by package simulations.

TABLE 20-4: DC TEMPERATURE AND VOLTAGE SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial -40°C ≤ T_A ≤ +125°C for Extended				
Param No.	Symbol	Characteristic	Min	Typ[1]	Max	Units	Conditions
Operating Voltage[2]							
DC10	V_{DD}	**Supply Voltage**	2.5	—	5.5	V	Industrial temperature
DC11	V_{DD}	**Supply Voltage**	3.0	—	5.5	V	Extended temperature
DC12	V_{DR}	**RAM Data Retention Voltage**[3]	—	1.5	—	V	
DC16	V_{POR}	**V_{DD} Start Voltage** (to ensure internal Power-on Reset signal)	—	V_{SS}	—	V	
DC17	S_{VDD}	**V_{DD} Rise Rate** (to ensure internal Power-on Reset signal)	0.05			V/ms	0-5V in 0.1 sec 0-3V in 60 ms

Note 1: "Typ" column data 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 V_{DD} can be lowered without losing RAM data.

TABLE 20-5:   DC CHARACTERISTICS: OPERATING CURRENT (IDD)

DC CHARACTERISTICS				Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended		
Parameter No.	Typical[1]	Max	Units	Conditions		
Operating Current (IDD)[2]						
DC20	—	—	mA	-40°C	3.3V	1 MIPS EC mode
DC20a	TBD	—	mA	25°C	3.3V	1 MIPS EC mode
DC20b	—	—	mA	85°C	3.3V	1 MIPS EC mode
DC20c	—	—	mA	125°C	3.3V	1 MIPS EC mode
DC20d	—	—	mA	-40°C	5V	1 MIPS EC mode
DC20e	TBD	—	mA	25°C	5V	1 MIPS EC mode
DC20f	—	—	mA	85°C	5V	1 MIPS EC mode
DC20g	—	—	mA	125°C	5V	1 MIPS EC mode
DC23	—	—	mA	-40°C	3.3V	4 MIPS EC mode, 4X PLL
DC23a	TBD	—	mA	25°C	3.3V	4 MIPS EC mode, 4X PLL
DC23b	—	—	mA	85°C	3.3V	4 MIPS EC mode, 4X PLL
DC23c	—	—	mA	125°C	3.3V	4 MIPS EC mode, 4X PLL
DC23d	—	—	mA	-40°C	5V	4 MIPS EC mode, 4X PLL
DC23e	TBD	—	mA	25°C	5V	4 MIPS EC mode, 4X PLL
DC23f	—	—	mA	85°C	5V	4 MIPS EC mode, 4X PLL
DC23g	—	—	mA	125°C	5V	4 MIPS EC mode, 4X PLL
DC24	—	—	mA	-40°C	3.3V	10 MIPS EC mode, 4X PLL
DC24a	TBD	—	mA	25°C	3.3V	10 MIPS EC mode, 4X PLL
DC24b	—	—	mA	85°C	3.3V	10 MIPS EC mode, 4X PLL
DC24c	—	—	mA	125°C	3.3V	10 MIPS EC mode, 4X PLL
DC24d	—	—	mA	-40°C	5V	10 MIPS EC mode, 4X PLL
DC24e	TBD	—	mA	25°C	5V	10 MIPS EC mode, 4X PLL
DC24f	—	—	mA	85°C	5V	10 MIPS EC mode, 4X PLL
DC24g	—	—	mA	125°C	5V	10 MIPS EC mode, 4X PLL
DC25	—	—	mA	-40°C	3V	8 MIPS EC mode, 8X PLL
DC25a	TBD	—	mA	25°C	3V	8 MIPS EC mode, 8X PLL
DC25b	—	—	mA	85°C	3V	8 MIPS EC mode, 8X PLL
DC25c	—	—	mA	125°C	3V	8 MIPS EC mode, 8X PLL
DC25d	—	—	mA	-40°C	5V	8 MIPS EC mode, 8X PLL
DC25e	TBD	—	mA	25°C	5V	8 MIPS EC mode, 8X PLL
DC25f	—	—	mA	85°C	5V	8 MIPS EC mode, 8X PLL
DC25g	—	—	mA	125°C	5V	8 MIPS EC mode, 8X PLL

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. $\overline{MCLR}$ = VDD, WDT, FSCM, LVD and BOR are disabled. CPU, SRAM, Program Memory and Data Memory are operational. No peripheral modules are operating.

TABLE 20-5: DC CHARACTERISTICS: OPERATING CURRENT (I$_{DD}$) (CONTINUED)

DC CHARACTERISTICS				Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ T$_A$ ≤ +85°C for Industrial -40°C ≤ T$_A$ ≤ +125°C for Extended		
Parameter No.	Typical[1]	Max	Units	Conditions		
Operating Current (I$_{DD}$)[2]						
DC27	—	—	mA	-40°C	3.3V	20 MIPS EC mode, 8X PLL
DC27a	TBD	—	mA	25°C		
DC27b	—	—	mA	85°C		
DC27c	—	—	mA	-40°C	5V	
DC27d	TBD	—	mA	25°C		
DC27e	—	—	mA	85°C		
DC27f	—	—	mA	125°C		
DC28	—	—	mA	-40°C	3.3V	16 MIPS EC mode, 16X PLL
DC28a	TBD	—	mA	25°C		
DC28b	—	—	mA	85°C		
DC28c	—	—	mA	-40°C	5V	
DC28d	TBD	—	mA	25°C		
DC28e	—	—	mA	85°C		
DC28f	—	—	mA	125°C		
DC29	—	—	mA	-40°C	5V	30 MIPS EC mode, 16X PLL
DC29a	TBD	—	mA	25°C		
DC29b	—	—	mA	85°C		
DC29c	—	—	mA	125°C		
DC30	—	—	mA	-40°C	3.3V	FRC (~ 2 MIPS)
DC30a	TBD	—	mA	25°C		
DC30b	—	—	mA	85°C		
DC30c	—	—	mA	125°C		
DC30d	—	—	mA	-40°C	5V	
DC30e	TBD	—	mA	25°C		
DC30f	—	—	mA	85°C		
DC30g	—	—	mA	125°C		
DC31	—	—	mA	-40°C	3.3V	LPRC (~ 512 kHz)
DC31a	TBD	—	mA	25°C		
DC31b	—	—	mA	85°C		
DC31c	—	—	mA	125°C		
DC31d	—	—	mA	-40°C	5V	
DC31e	TBD	—	mA	25°C		
DC31f	—	—	mA	85°C		
DC31g	—	—	mA	125°C		

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 I$_{DD}$ 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 V$_{DD}$. $\overline{MCLR}$ = V$_{DD}$, WDT, FSCM, LVD and BOR are disabled. CPU, SRAM, Program Memory and Data Memory are operational. No peripheral modules are operating.

TABLE 20-6: DC CHARACTERISTICS: IDLE CURRENT (I$_{IDLE}$)

DC CHARACTERISTICS				Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ T$_A$ ≤ +85°C for Industrial -40°C ≤ T$_A$ ≤ +125°C for Extended		
Parameter No.	Typical[1]	Max	Units	Conditions		
Idle Current (I$_{IDLE}$): Core OFF Clock ON Base Current[2]						
DC40	—	—	mA	-40°C	3.3V	1 MIPS EC mode
DC40a	TBD	—	mA	25°C		
DC40b	—	—	mA	85°C		
DC40c	—	—	mA	125°C		
DC40d	—	—	mA	-40°C	5V	
DC40e	TBD	—	mA	25°C		
DC40f	—	—	mA	85°C		
DC40g	—	—	mA	125°C		
DC43	—	—	mA	-40°C	3.3V	4 MIPS EC mode, 4X PLL
DC43a	TBD	—	mA	25°C		
DC43b	—	—	mA	85°C		
DC43c	—	—	mA	125°C		
DC43d	—	—	mA	-40°C	5V	
DC43e	TBD	—	mA	25°C		
DC43f	—	—	mA	85°C		
DC43g	—	—	mA	125°C		
DC44	—	—	mA	-40°C	3.3V	10 MIPS EC mode, 4X PLL
DC44a	TBD	—	mA	25°C		
DC44b	—	—	mA	85°C		
DC44c	—	—	mA	125°C		
DC44d	—	—	mA	-40°C	5V	
DC44e	TBD	—	mA	25°C		
DC44f	—	—	mA	85°C		
DC44g	—	—	mA	125°C		
DC45	—	—	mA	-40°C	3.3V	8 MIPS EC mode, 8X PLL
DC45a	TBD	—	mA	25°C		
DC45b	—	—	mA	85°C		
DC45c	—	—	mA	125°C		
DC45d	—	—	mA	-40°C	5V	
DC45e	TBD	—	mA	25°C		
DC45f	—	—	mA	85°C		
DC45g	—	—	mA	125°C		

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 I$_{IDLE}$ current is measured with Core off, Clock on and all modules turned off.

TABLE 20-6:   DC CHARACTERISTICS: IDLE CURRENT (IIDLE) (CONTINUED)

DC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended		
Parameter No.	Typical[1]	Max	Units		Conditions
Idle Current (IIDLE): Core OFF Clock ON Base Current[2]					
DC47	—	—	mA	-40°C	
DC47a	TBD	—	mA	25°C	3.3V
DC47b	—	—	mA	85°C	
DC47c	—	—	mA	-40°C	20 MIPS EC mode, 8X PLL
DC47d	TBD	—	mA	25°C	5V
DC47e	—	—	mA	85°C	
DC47f	—	—	mA	125°C	
DC48	—	—	mA	-40°C	
DC48a	TBD	—	mA	25°C	3.3V
DC48b	—	—	mA	85°C	
DC48c	—	—	mA	-40°C	16 MIPS EC mode, 16X PLL
DC48d	TBD	—	mA	25°C	5V
DC48e	—	—	mA	85°C	
DC48f	—	—	mA	125°C	
DC49	—	—	mA	-40°C	
DC49a	TBD	—	mA	25°C	5V
DC49b	—	—	mA	85°C	30 MIPS EC mode, 16X PLL
DC49c	—	—	mA	125°C	
DC50	—	—	mA	-40°C	
DC50a	TBD	—	mA	25°C	3.3V
DC50b	—	—	mA	85°C	
DC50c	—	—	mA	125°C	
DC50d	—	—	mA	-40°C	FRC (~ 2 MIPS)
DC50e	TBD	—	mA	25°C	5V
DC50f	—	—	mA	85°C	
DC50g	—	—	mA	125°C	
DC51	—	—	mA	-40°C	
DC51a	TBD	—	mA	25°C	3.3V
DC51b	—	—	mA	85°C	
DC51c	—	—	mA	125°C	
DC51d	—	—	mA	-40°C	LPRC (~ 512 kHz)
DC51e	TBD	—	mA	25°C	5V
DC51f	—	—	mA	85°C	
DC51g	—	—	mA	125°C	

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.

TABLE 20-7:   DC CHARACTERISTICS: POWER-DOWN CURRENT (IPD)

DC CHARACTERISTICS				Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended	
Parameter No.	Typical[1]	Max	Units	Conditions	
Power Down Current (IPD)[2]					
DC60	—	—	μA	-40°C	Base Power Down Current[3]
DC60a	TBD	—	μA	25°C	
DC60b	—	—	μA	85°C	3.3V
DC60c	—	—	μA	125°C	
DC60d	—	—	μA	-40°C	
DC60e	TBD	—	μA	25°C	
DC60f	—	—	μA	85°C	5V
DC60g	—	—	μA	125°C	
DC61	—	—	μA	-40°C	Watchdog Timer Current: ΔIWDT[3]
DC61a	TBD	—	μA	25°C	
DC61b	—	—	μA	85°C	3.3V
DC61c	—	—	μA	125°C	
DC61d	—	—	μA	-40°C	
DC61e	TBD	—	μA	25°C	
DC61f	—	—	μA	85°C	5V
DC61g	—	—	μA	125°C	
DC62	—	—	μA	-40°C	Timer 1 w/32 kHz Crystal: ΔITI32[3]
DC62a	TBD	—	μA	25°C	
DC62b	—	—	μA	85°C	3.3V
DC62c	—	—	μA	125°C	
DC62d	—	—	μA	-40°C	
DC62e	TBD	—	μA	25°C	
DC62f	—	—	μA	85°C	5V
DC62g	—	—	μA	125°C	
DC63	—	—	μA	-40°C	BOR On: ΔIBOR[3]
DC63a	TBD	—	μA	25°C	
DC63b	—	—	μA	85°C	3.3V
DC63c	—	—	μA	125°C	
DC63d	—	—	μA	-40°C	
DC63e	TBD	—	μA	25°C	
DC63f	—	—	μA	85°C	5V
DC63g	—	—	μA	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 Δ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

TABLE 20-7:   DC CHARACTERISTICS: POWER-DOWN CURRENT (IPD) (CONTINUED)

DC CHARACTERISTICS				Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended		
Parameter No.	Typical[1]	Max	Units	Conditions		
Power Down Current (IPD)[2]						
DC66	—	—	μA	-40°C	3.3V	Low Voltage Detect: ΔILVD[3]
DC66a	TBD	—	μA	25°C		
DC66b	—	—	μA	85°C		
DC66c	—	—	μA	125°C		
DC66d	—	—	μA	-40°C	5V	
DC66e	TBD	—	μA	25°C		
DC66f	—	—	μA	85°C		
DC66g	—	—	μA	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 Δ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

TABLE 20-8:   DC CHARACTERISTICS: I/O PIN INPUT SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic	Min	Typ[1]	Max	Units	Conditions
	V_{IL}	**Input Low Voltage**[2]					
DI10		I/O pins: with Schmitt Trigger buffer	V_{SS}	—	0.2 V_{DD}	V	
DI15		$\overline{MCLR}$	V_{SS}	—	0.2 V_{DD}	V	
DI16		OSC1 (in XT, HS and LP modes)	V_{SS}	—	0.2 V_{DD}	V	
DI17		OSC1 (in RC mode)[3]	V_{SS}	—	0.3 V_{DD}	V	
DI18		SDA, SCL	TBD	—	TBD	V	SM bus disabled
DI19		SDA, SCL	TBD	—	TBD	V	SM bus enabled
	V_{IH}	**Input High Voltage**[2]					
DI20		I/O pins: with Schmitt Trigger buffer	0.8 V_{DD}	—	V_{DD}	V	
DI25		$\overline{MCLR}$	0.8 V_{DD}	—	V_{DD}	V	
DI26		OSC1 (in XT, HS and LP modes)	0.7 V_{DD}	—	V_{DD}	V	
DI27		OSC1 (in RC mode)[3]	0.9 V_{DD}	—	V_{DD}	V	
DI28		SDA, SCL	TBD	—	TBD	V	SM bus disabled
DI29		SDA, SCL	TBD	—	TBD	V	SM bus enabled
	I_{CNPU}	**CNxx Pull-up Current**[2]					
DI30			50	250	400	µA	V_{DD} = 5V, V_{PIN} = V_{SS}
DI31			TBD	TBD	TBD	µA	V_{DD} = 3V, V_{PIN} = V_{SS}
	I_{IL}	**Input Leakage Current**[2][4][5]					
DI50		I/O ports	—	0.01	±1	µA	V_{SS} ≤ V_{PIN} ≤ V_{DD}, Pin at hi-impedance
DI51		Analog input pins	—	0.50	—	µA	V_{SS} ≤ V_{PIN} ≤ V_{DD}, Pin at hi-impedance
DI55		$\overline{MCLR}$	—	0.05	±5	µA	V_{SS} ≤ V_{PIN} ≤ V_{DD}
DI56		OSC1	—	0.05	±5	µA	V_{SS} ≤ V_{PIN} ≤ V_{DD}, 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 $\overline{MCLR}$ 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.

TABLE 20-9: DC CHARACTERISTICS: I/O PIN OUTPUT SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic	Min	Typ[1]	Max	Units	Conditions
	VOL	**Output Low Voltage[2]**					
DO10		I/O ports	—	—	0.6	V	I_{OL} = 8.5 mA, V_{DD} = 5V
			—	—	TBD	V	I_{OL} = 2.0 mA, V_{DD} = 3V
DO16		OSC2/CLKOUT	—	—	0.6	V	I_{OL} = 1.6 mA, V_{DD} = 5V
		(RC or EC Osc mode)	—	—	TBD	V	I_{OL} = 2.0 mA, V_{DD} = 3V
	VOH	**Output High Voltage[2]**					
DO20		I/O ports	VDD – 0.7	—	—	V	I_{OH} = -3.0 mA, V_{DD} = 5V
			TBD	—	—	V	I_{OH} = -2.0 mA, V_{DD} = 3V
DO26		OSC2/CLKOUT	VDD – 0.7	—	—	V	I_{OH} = -1.3 mA, V_{DD} = 5V
		(RC or EC Osc mode)	TBD	—	—	V	I_{OH} = -2.0 mA, V_{DD} = 3V
		Capacitive Loading Specs on Output Pins[2]					
DO50	COSC2	OSC2/SOSC2 pin	—	—	15	pF	In XTL, XT, HS and LP modes when external clock is used to drive OSC1.
DO56	CIO	All I/O pins and OSC2	—	—	50	pF	RC or EC Osc mode
DO58	CB	SCL, SDA	—	—	400	pF	In I²C 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.

FIGURE 20-1: LOW-VOLTAGE DETECT CHARACTERISTICS
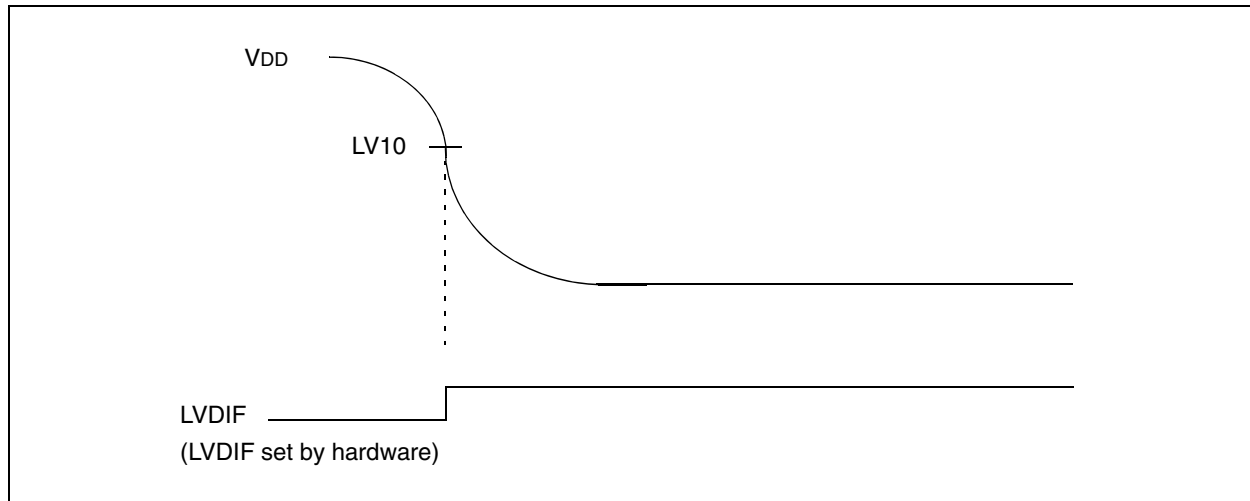
TABLE 20-10: ELECTRICAL CHARACTERISTICS: LVDL

DC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Param No.	Symbol	Characteristic[1]		Min	Typ	Max	Units	Conditions
LV10	VPLVD	LVDL Voltage on VDD transition high to low	LVDL = 0000[2]	—	—	—	V	
			LVDL = 0001[2]	—	—	—	V	
			LVDL = 0010[2]	—	—	—	V	
			LVDL = 0011[2]	—	—	—	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 20-2: BROWN-OUT RESET CHARACTERISTICS
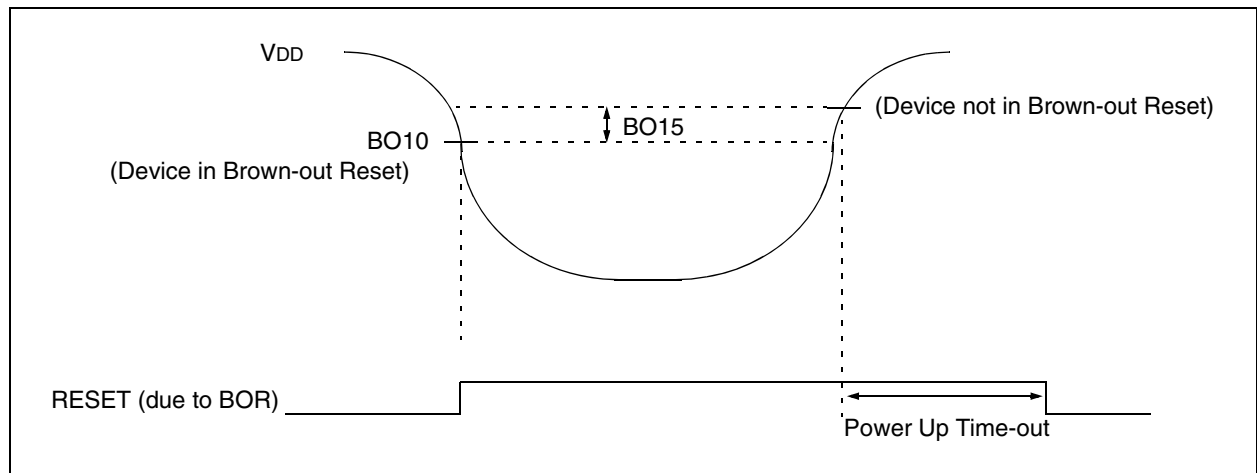
TABLE 20-11: ELECTRICAL CHARACTERISTICS: BOR

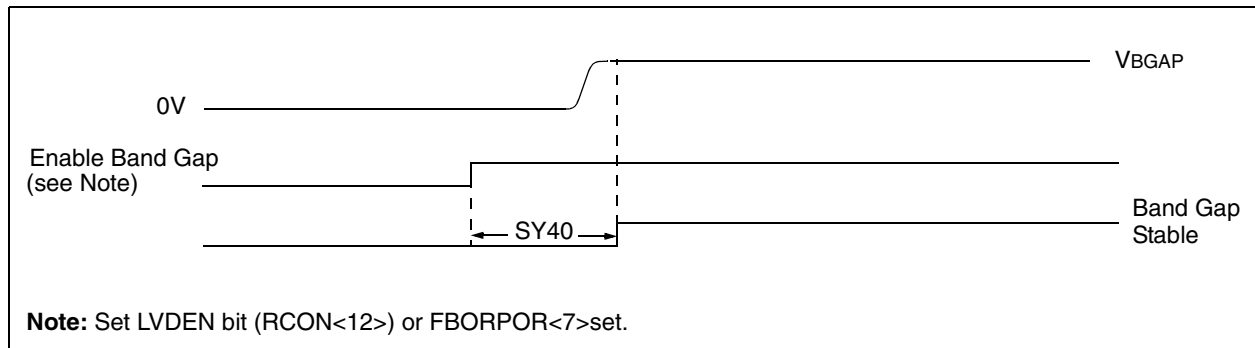| DC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Param No.	Symbol	Characteristic		Min	Typ[1]	Max	Units	Conditions
BO10	VBOR	BOR Voltage[2] on VDD transition high to low	BORV = 00[3]	—	—	—	V	Not in operating range
			BORV = 01	2.7	—	2.86	V	
			BORV = 10	4.2	—	4.46	V	
			BORV = 11	4.5	—	4.78	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: `00` values not in usable operating range.

TABLE 20-12: DC CHARACTERISTICS: PROGRAM AND EEPROM

DC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic	Min	Typ[1]	Max	Units	Conditions
		Data EEPROM Memory[2]					
D120	ED	Byte Endurance	100K	1M	—	E/W	-40°C ≤ TA ≤ +85°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[2]					
D130	EP	Cell Endurance	10K	100K	—	E/W	-40°C ≤ TA ≤ +85°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	TEB	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.

20.2   AC Characteristics and Timing Parameters

The information contained in this section defines dsPIC30F AC characteristics and timing parameters.

TABLE 20-13:   TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

AC CHARACTERISTICS	Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended Operating voltage VDD range as described in DC Spec Section 20.0.

FIGURE 20-3:   LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS
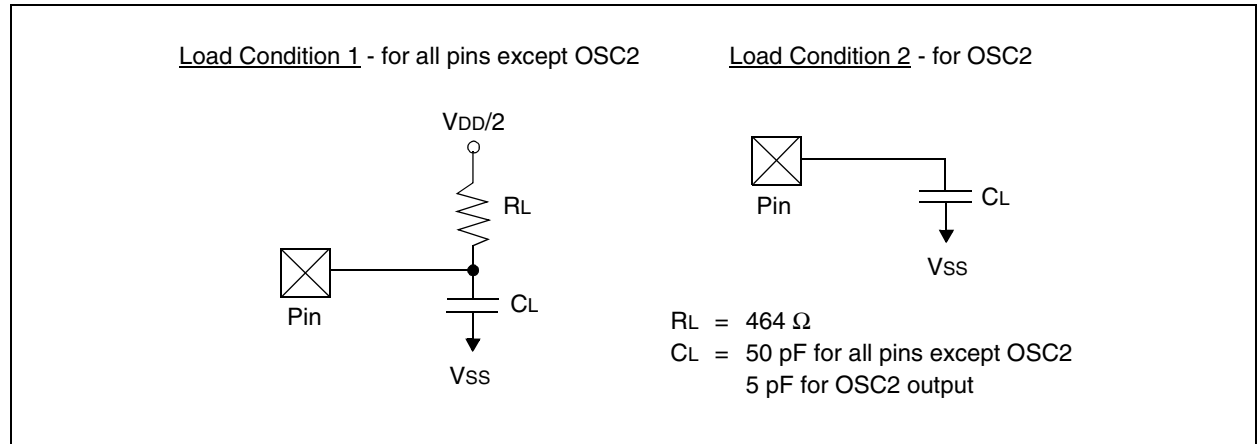
FIGURE 20-4:   EXTERNAL CLOCK TIMING
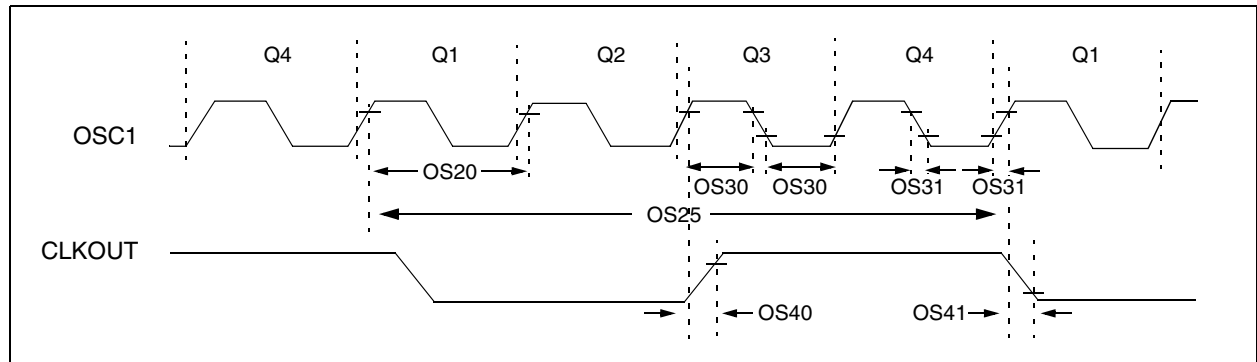
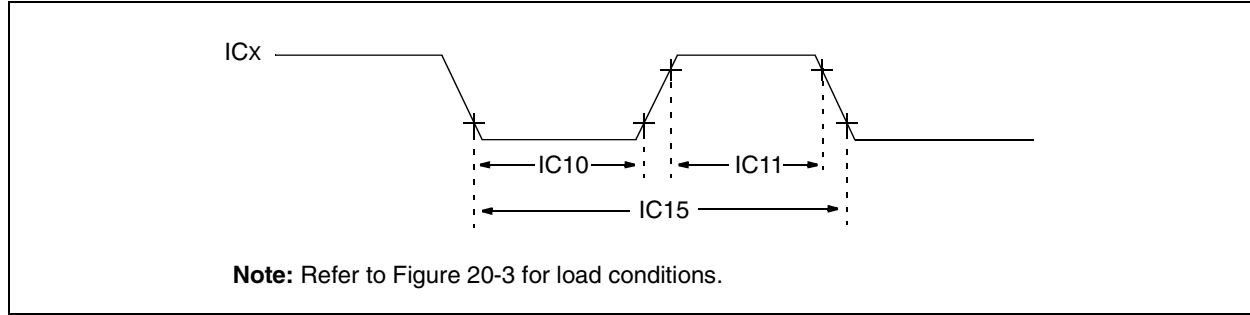## TABLE 20-14: EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic	Min	Typ[1]	Max	Units	Conditions
OS10	FOSC	External CLKIN Frequency[2] (External clocks allowed only in EC mode)	DC	—	40	MHz	EC
			4	—	10	MHz	EC with 4x PLL
			4	—	10	MHz	EC with 8x PLL
			4	—	7.5	MHz	EC with 16x PLL
		Oscillator Frequency[2]	DC	—	4	MHz	RC
			0.4	—	4	MHz	XTL
			4	—	10	MHz	XT
			4	—	10	MHz	XT with 4x PLL
			4	—	10	MHz	XT with 8x PLL
			4	—	7.5	MHz	XT with 16x PLL
			10	—	25	MHz	HS
			10	—	20	MHz	HS/2 with 4x PLL
			10	—	20	MHz	HS/2 with 8x PLL
			10	—	15	MHz	HS/2 with 16x PLL
			12	—	25	MHz	HS/3 with 4x PLL
			12	—	25	MHz	HS/3 with 8x PLL
			12	—	22.5	MHz	HS/3 with 16x PLL
			31	—	33	kHz	LP
			—	7.37	—	MHz	FRC internal
			—	7.37	—	MHz	FRC internal w/4x PLL
			—	7.37	—	MHz	FRC internal w/8x PLL
			—	7.37	—	MHz	FRC internal w/16x PLL
			—	512	—	kHz	LPRC internal
OS20	TOSC	TOSC = 1/FOSC	—	—	—	—	See parameter OS10 for FOSC value
OS25	TCY	Instruction Cycle Time[2][3]	33	—	DC	ns	See Table 20-16
OS30	TosL, TosH	External Clock[2] in (OSC1) High or Low Time	.45 x TOSC	—	—	ns	EC
OS31	TosR, TosF	External Clock[2] in (OSC1) Rise or Fall Time	—	—	20	ns	EC
OS40	TckR	CLKOUT Rise Time[2][4]	—	6	10	ns	
OS41	TckF	CLKOUT Fall Time[2][4]	—	6	10	ns	

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 CLKOUT signal is measured on the OSC2 pin. CLKOUT is low for the Q1-Q2 period (1/2 TCY) and high for the Q3-Q4 period (1/2 TCY).

TABLE 20-15: PLL CLOCK TIMING SPECIFICATIONS (V$_{DD}$ = 2.5 TO 5.5 V)

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ T$_A$ ≤ +85°C for Industrial -40°C ≤ T$_A$ ≤ +125°C for Extended					
Param No.	Symbol	Characteristic[1]	Min	Typ[2]	Max	Units	Conditions	
OS50	F$_{PLLI}$	PLL Input Frequency Range[2]	4	—	10	MHz	EC, XT modes with PLL	
OS51	F$_{SYS}$	On-chip PLL Output[2]	16	—	120	MHz	EC, XT modes with PLL	
OS52	T$_{LOC}$	PLL Start-up Time (Lock Time)	—	20	50	μs		
OS53	D$_{CLK}$	CLKOUT Stability (Jitter)	TBD	1	TBD	%	Measured over 100 ms period	

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.

TABLE 20-16: INTERNAL CLOCK TIMING EXAMPLES

Clock Oscillator Mode	F$_{OSC}$ (MHz)[1]	T$_{CY}$ (μsec)[2]	MIPS[3] w/o PLL	MIPS[3] w PLL x4	MIPS[3] w PLL x8	MIPS[3] 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	—	—	—
XT	4	1.0	1.0	4.0	8.0	16.0
	10	0.4	2.5	10.0	20.0	—

Note 1: Assumption: Oscillator Postscaler is divide by 1.

2: Instruction Execution Cycle Time: T$_{CY}$ = 1 / MIPS.

3: Instruction Execution Frequency: MIPS = (F$_{OSC}$ * PLLx) / 4 [since there are 4 Q clocks per instruction cycle].

TABLE 20-17: AC CHARACTERISTICS: INTERNAL RC ACCURACY

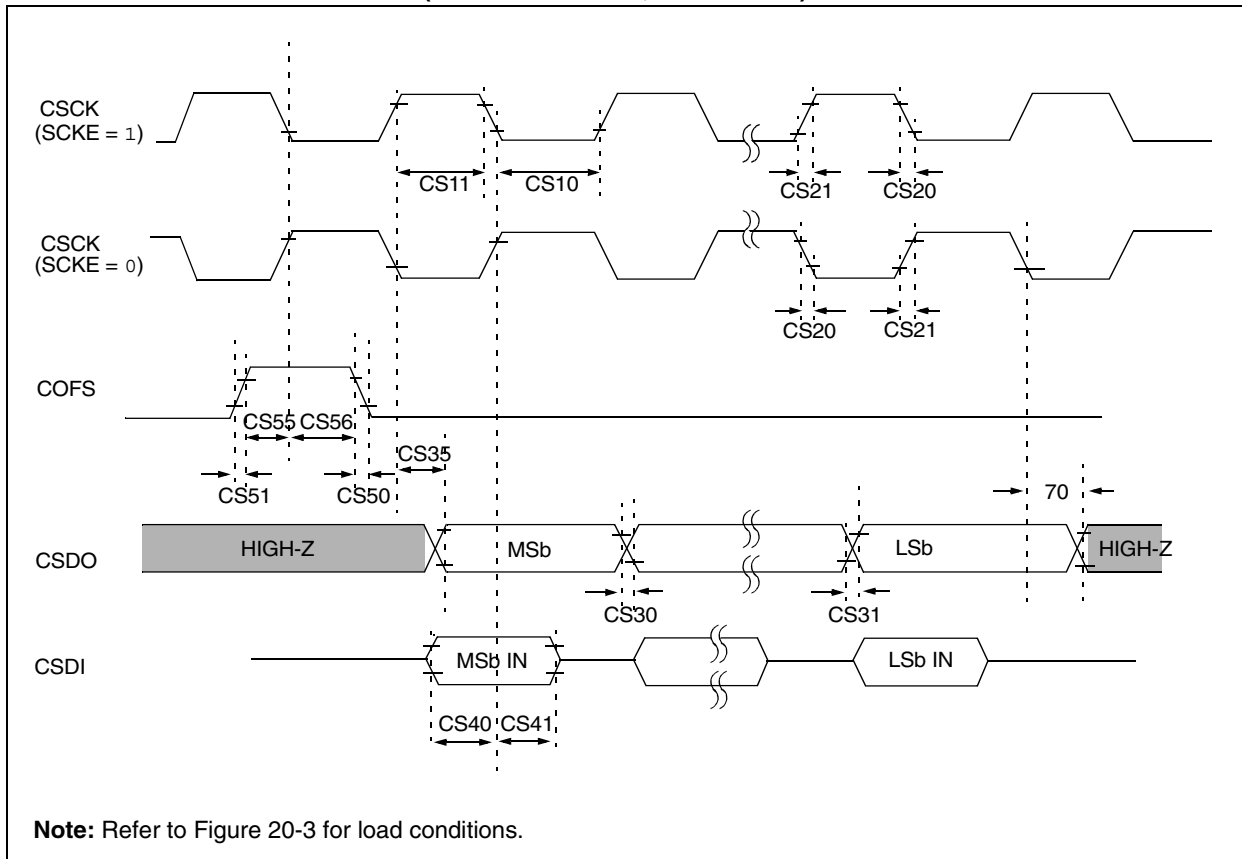| AC CHARACTERISTICS						Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for Extended	
Param No.	Characteristic	Min	Typ	Max	Units	Conditions	
Internal FRC Accuracy @ FRC Freq = 7.37 MHz[1]							
	FRC		TBD		%	+25°C	VDD = 3.0-3.6V
			TBD		%	+25°C	VDD = 4.5-5.5V
			TBD		%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
			TBD		%	-40°C ≤ TA ≤ +85°C	VDD = 4.5-5.5V
			TBD		%	-40°C ≤ TA ≤ +125°C	VDD = 4.5-5.5V
	FRC with x4 PLL		TBD		%	+25°C	VDD = 3.0-3.6V
			TBD		%	+25°C	VDD = 4.5-5.5V
			TBD		%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
			TBD		%	-40°C ≤ TA ≤ +85°C	VDD = 4.5-5.5V
			TBD		%	-40°C ≤ TA ≤ +125°C	VDD = 4.5-5.5V
	FRC with x8 PLL		TBD		%	+25°C	VDD = 3.0-3.6V
			TBD		%	+25°C	VDD = 4.5-5.5V
			TBD		%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
			TBD		%	-40°C ≤ TA ≤ +85°C	VDD = 4.5-5.5V
			TBD		%	-40°C ≤ TA ≤ +125°C	VDD = 4.5-5.5V
	FRC with x16 PLL		TBD		%	+25°C	VDD = 3.0-3.6V
			TBD		%	+25°C	VDD = 4.5-5.5V
			TBD		%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
			TBD		%	-40°C ≤ TA ≤ +85°C	VDD = 4.5-5.5V
			TBD		%	-40°C ≤ TA ≤ +125°C	VDD = 4.5-5.5V

Note 1: Frequency calibrated at 25°C and 5V. TUN bits can be used to compensate for temperature drift.

TABLE 20-18: AC CHARACTERISTICS: INTERNAL RC JITTER

AC CHARACTERISTICS		Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for Extended					
Param No.	Characteristic	Min	Typ	Max	Units	Conditions	
	Internal FRC Jitter @ FRC Freq = 7.37 MHz[1]						
	FRC		TBD		%	+25°C	VDD = 3.0-3.6V
			TBD		%	+25°C	VDD = 4.5-5.5V
			TBD		%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
			TBD		%	-40°C ≤ TA ≤ +85°C	VDD = 4.5-5.5V
			TBD		%	-40°C ≤ TA ≤ +125°C	VDD = 4.5-5.5V
	FRC with x4 PLL		TBD		%	+25°C	VDD = 3.0-3.6V
			TBD		%	+25°C	VDD = 4.5-5.5V
			TBD		%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
			TBD		%	-40°C ≤ TA ≤ +85°C	VDD = 4.5-5.5V
			TBD		%	-40°C ≤ TA ≤ +125°C	VDD = 4.5-5.5V
	FRC with x8 PLL		TBD		%	+25°C	VDD = 3.0-3.6V
			TBD		%	+25°C	VDD = 4.5-5.5V
			TBD		%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
			TBD		%	-40°C ≤ TA ≤ +85°C	VDD = 4.5-5.5V
			TBD		%	-40°C ≤ TA ≤ +125°C	VDD = 4.5-5.5V
	FRC with x16 PLL		TBD		%	+25°C	VDD = 3.0-3.6V
			TBD		%	+25°C	VDD = 4.5-5.5V
			TBD		%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
			TBD		%	-40°C ≤ TA ≤ +85°C	VDD = 4.5-5.5V
			TBD		%	-40°C ≤ TA ≤ +125°C	VDD = 4.5-5.5V

Note 1: Frequency calibrated at 25°C and 5V. TUN bits can be used to compensate for temperature drift.

TABLE 20-19: INTERNAL RC ACCURACY

AC CHARACTERISTICS		Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Param No.	Characteristic	Min	Typ	Max	Units	Conditions	
	LPRC @ Freq = 512 kHz						
F20		TBD	—	TBD	%	-40°C to +85°C	VDD = 3V
F21		TBD	—	TBD	%	-40°C to +85°C	VDD = 5V

Note 1: Frequency at 25°C and 5V.
    2: Change of LPRC frequency as VDD changes.

FIGURE 20-5: CLKOUT AND I/O TIMING CHARACTERISTICS
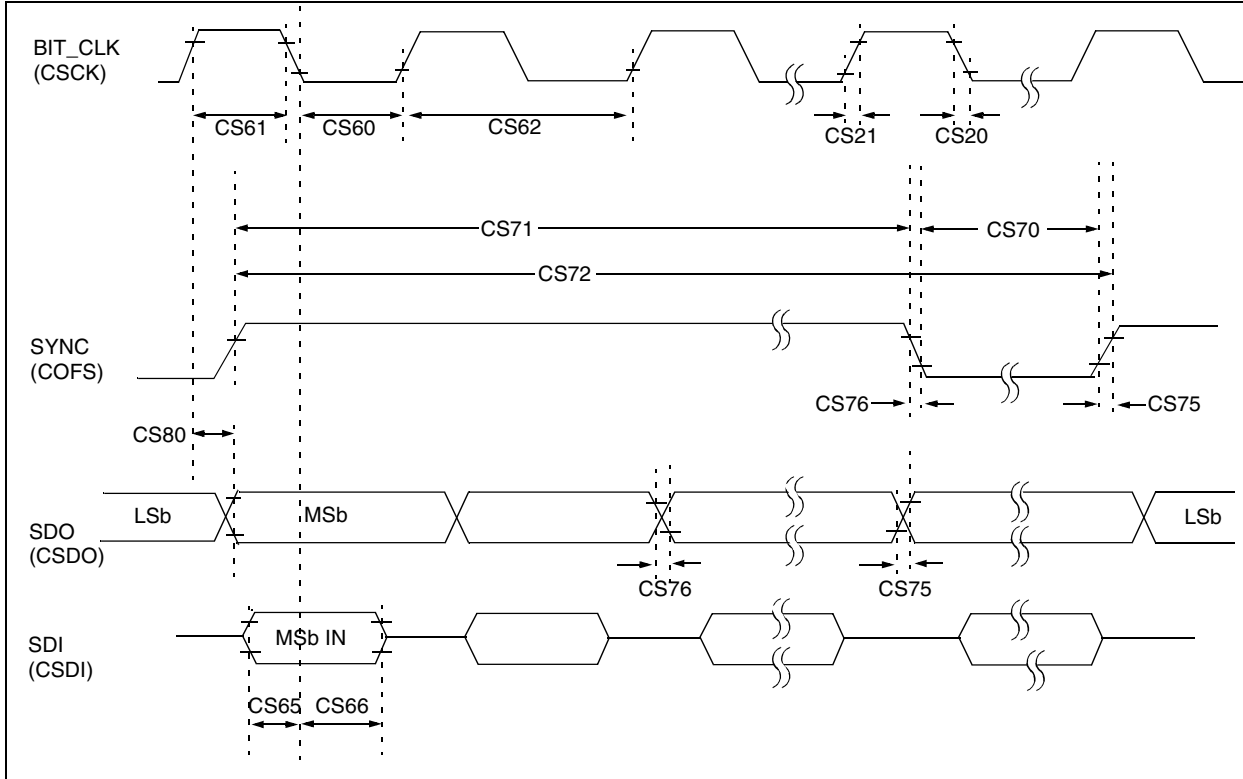

Note: Refer to Figure 20-3 for load conditions.
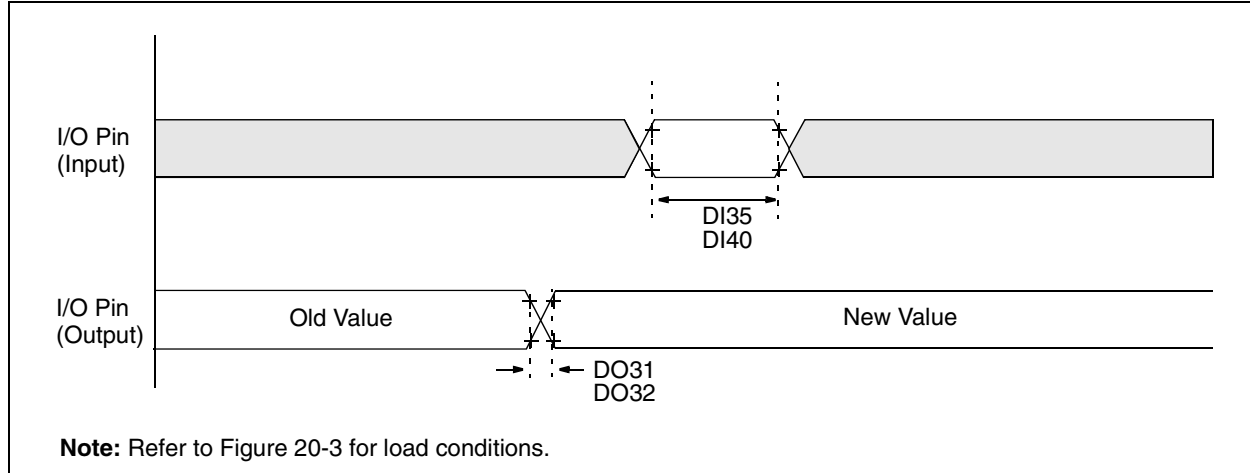
TABLE 20-20: CLKOUT AND I/O TIMING REQUIREMENTS

AC CHARACTERISTICS			**Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Param No.	**Symbol**	**Characteristic**[1][2][3]	**Min**	**Typ**[4]	**Max**	**Units**	**Conditions**	
DO31	TIOR	Port output rise time	—	10	25	ns	—	
DO32	TIOF	Port output fall time	—	10	25	ns	—	
DI35	TINP	INTx pin high or low time (output)	20	—	—	ns	—	
DI40	TRBP	CNx high or low time (input)	2 TCY	—	—	ns	—	

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 CLKOUT 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.

FIGURE 20-6: **RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING CHARACTERISTICS**

Note: Refer to Figure 20-3 for load conditions.
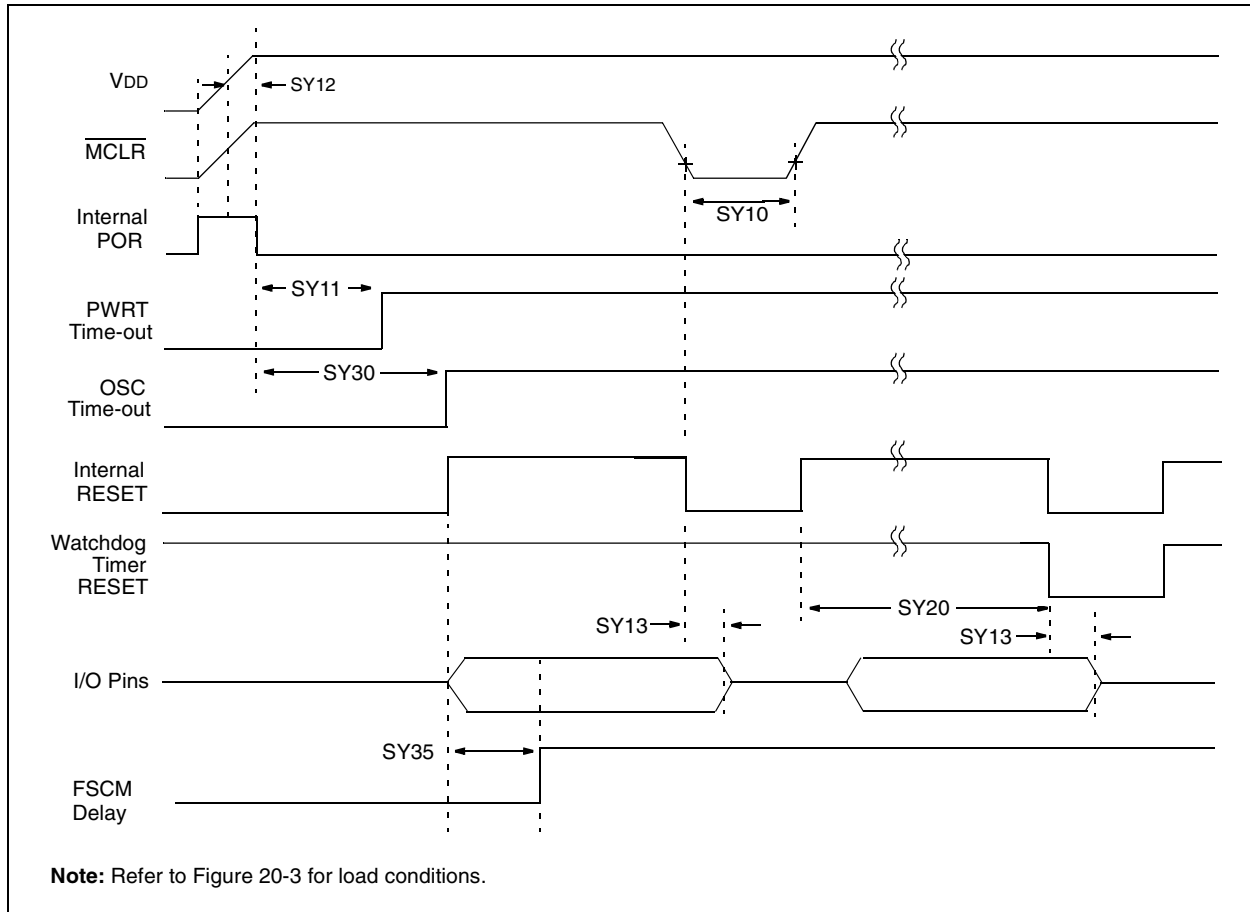
TABLE 20-21: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET TIMING REQUIREMENTS

AC CHARACTERISTICS							Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended
Param No.	Symbol	Characteristic[1]	Min	Typ[2]	Max	Units	Conditions
SY10	TmcL	$\overline{\text{MCLR}}$ Pulse Width (low)	2	—	—	µs	-40°C to +85°C
SY11	TPWRT	Power-up Timer Period	TBD TBD TBD TBD	0 4 16 64	TBD TBD TBD TBD	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 Hi-impedance from $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset	—	—	100	ns	
SY20	TWDT1	Watchdog Timer Time-out Period (No Prescaler)	1.8	2.0	2.2	ms	VDD = 5V, -40°C to +85°C
	TWDT2		1.9	2.1	2.3	ms	VDD = 3V, -40°C to +85°C
SY25	TBOR	Brown-out Reset Pulse Width[3]	100	—	—	µs	VDD ≤ VBOR (D034)
SY30	TOST	Oscillation Start-up Timer Period	—	1024 TOSC	—	—	TOSC = OSC1 period
SY35	TFSCM	Fail-Safe Clock Monitor Delay	—	100	—	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 20-2 and Table 20-11 for BOR.

FIGURE 20-7:   BAND GAP START-UP TIME CHARACTERISTICS

Note: Set LVDEN bit (RCON<12>) or FBORPOR<7>set.

Preliminary

TABLE 20-22: BAND GAP START-UP TIME REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic[1]	Min	Typ[2]	Max	Units	Conditions
SY40	TBGAP	Band Gap Start-up Time	—	20	50	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 20-8: TYPE A, B AND C TIMER EXTERNAL CLOCK TIMING CHARACTERISTICS

Note: Refer to Figure 20-3 for load conditions.
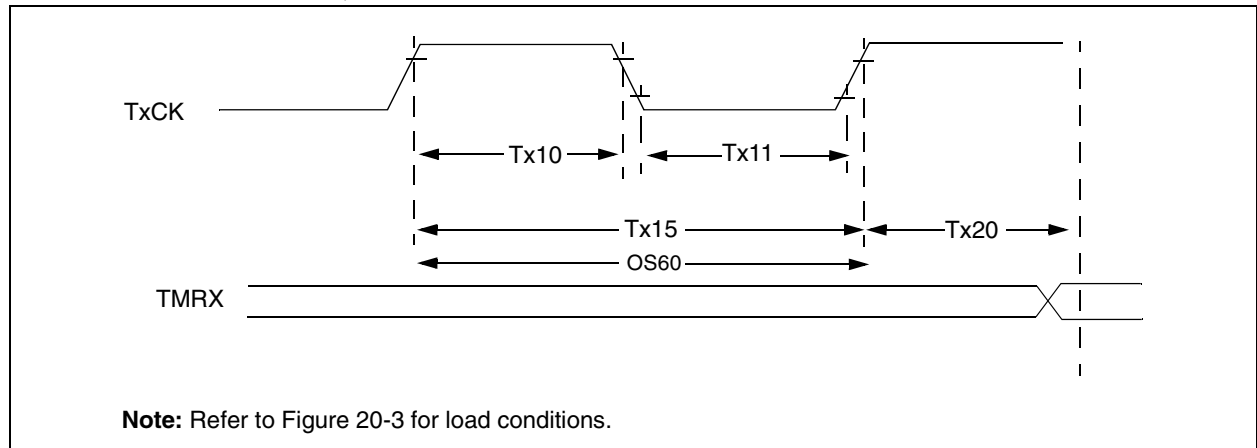
TABLE 20-23: TYPE A TIMER (TIMER1) EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS				Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic		Min	Typ	Max	Units	Conditions
TA10	TTxH	TxCK High Time	Synchronous, no prescaler	0.5 TCY + 20	—	—	ns	Must also meet parameter TA15
			Synchronous, with prescaler	10	—	—	ns	
			Asynchronous	10	—	—	ns	
TA11	TTxL	TxCK Low Time	Synchronous, no prescaler	0.5 TCY + 20	—	—	ns	Must also meet parameter TA15
			Synchronous, with prescaler	10	—	—	ns	
			Asynchronous	10	—	—	ns	
TA15	TTxP	TxCK Input Period	Synchronous, no prescaler	TCY + 10	—	—	ns	
			Synchronous, with prescaler	Greater of: 20 ns or (TCY + 40)/N	—	—	—	N = prescale value (1, 8, 64, 256)
			Asynchronous	20	—	—	ns	
OS60	Ft1	SOSC1/T1CK oscillator input frequency range (oscillator enabled by setting bit TCS (T1CON, bit 1))		DC	—	50	kHz	
TA20	TCKEXTMRL	Delay from External TxCK Clock Edge to Timer Increment		2 TOSC		6 TOSC	—	

Note: Timer1 is a Type A.

TABLE 20-24: TYPE B TIMER (TIMER2 AND TIMER4) EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS				Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic		Min	Typ	Max	Units	Conditions
TB10	TtxH	TxCK High Time	Synchronous, no prescaler	0.5 TCY + 20	—	—	ns	Must also meet parameter TB15
			Synchronous, with prescaler	10	—	—	ns	
TB11	TtxL	TxCK Low Time	Synchronous, no prescaler	0.5 TCY + 20	—	—	ns	Must also meet parameter TB15
			Synchronous, with prescaler	10	—	—	ns	
TB15	TtxP	TxCK Input Period	Synchronous, no prescaler	TCY + 10	—	—	ns	N = prescale value (1, 8, 64, 256)
			Synchronous, with prescaler	Greater of: 20 ns or (TCY + 40)/N				
TB20	TCKEXTMRL	Delay from External TxCK Clock Edge to Timer Increment		2 TOSC	—	6 TOSC	—	

Note: Timer2 and Timer4 are Type B.

TABLE 20-25: TYPE C TIMER (TIMER3 AND TIMER5) EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS				Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic		Min	Typ	Max	Units	Conditions
TC10	TtxH	TxCK High Time	Synchronous	0.5 TCY + 20	—	—	ns	Must also meet parameter TC15
TC11	TtxL	TxCK Low Time	Synchronous	0.5 TCY + 20	—	—	ns	Must also meet parameter TC15
TC15	TtxP	TxCK Input Period	Synchronous, no prescaler	TCY + 10	—	—	ns	N = prescale value (1, 8, 64, 256)
			Synchronous, with prescaler	Greater of: 20 ns or (TCY + 40)/N				
TC20	TCKEXTMRL	Delay from External TxCK Clock Edge to Timer Increment		2 TOSC	—	6 TOSC	—	

Note: Timer3 and Timer5 are Type C.

FIGURE 20-9: **INPUT CAPTURE (CAPx) TIMING CHARACTERISTICS**

Note: Refer to Figure 20-3 for load conditions.

TABLE 20-26: INPUT CAPTURE TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic[1]		Min	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 20-10: **OUTPUT COMPARE MODULE (OCx) TIMING CHARACTERISTICS**

Note: Refer to Figure 20-3 for load conditions.
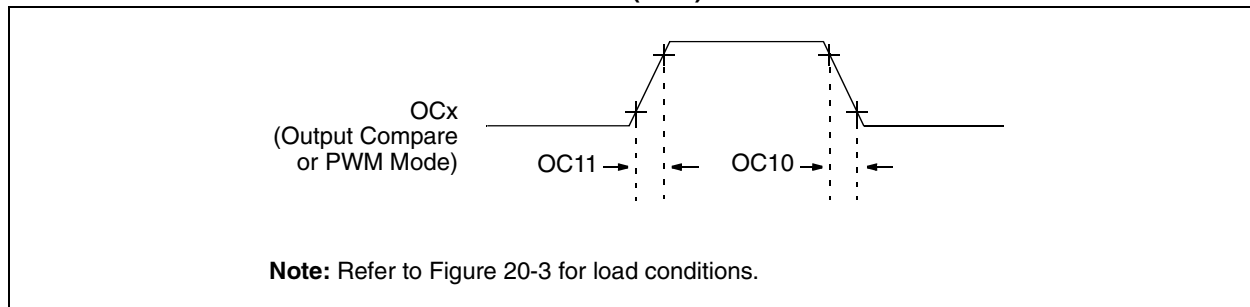
TABLE 20-27: OUTPUT COMPARE MODULE TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic[1]	Min	Typ[2]	Max	Units	Conditions
OC10	TccF	OCx Output Fall Time	—	10	25	ns	—
OC11	TccR	OCx Output Rise Time	—	10	25	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 20-11: OC/PWM MODULE TIMING CHARACTERISTICS
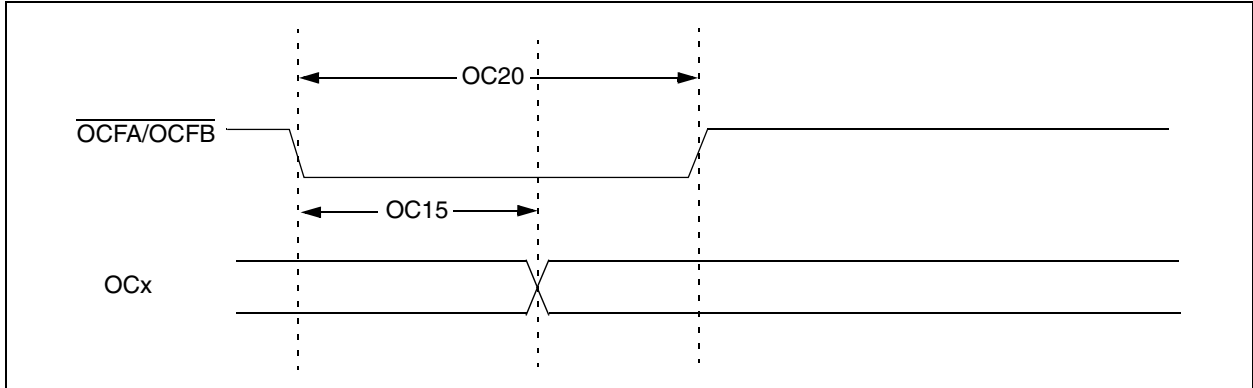
TABLE 20-28: SIMPLE OC/PWM MODE TIMING REQUIREMENTS

AC CHARACTERISTICS							Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended	
Param No.	Symbol	Characteristic[1]	Min	Typ[2]	Max	Units	Conditions	
OC15	TFD	Fault Input to PWM I/O Change	—	—	25	ns	VDD = 3V	-40°C to +85°C
					TBD	ns	VDD = 5V	
OC20	TFLT	Fault Input Pulse Width	—	—	50	ns	VDD = 3V	-40°C to +85°C
					TBD	ns	VDD = 5V	

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 20-12: DCI MODULE (MULTICHANNEL, I²S MODES) TIMING CHARACTERISTICS

Note: Refer to Figure 20-3 for load conditions.

TABLE 20-29: DCI MODULE (MULTICHANNEL, I²S MODES) TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic[1]	Min	Typ[2]	Max	Units	Conditions
CS10	TcSCKL	CSCK Input Low Time (CSCK pin is an input)	TCY / 2 + 20	—	—	ns	—
		CSCK Output Low Time[3] (CSCK pin is an output)	30	—	—	ns	—
CS11	TcSCKH	CSCK Input High Time (CSCK pin is an input)	TCY / 2 + 20	—	—	ns	—
		CSCK Output High Time[3] (CSCK pin is an output)	30	—	—	ns	—
CS20	TcSCKF	CSCK Output Fall Time[4] (CSCK pin is an output)	—	10	25	ns	—
CS21	TcSCKR	CSCK Output Rise Time[4] (CSCK pin is an output)	—	10	25	ns	—
CS30	TcSDOF	CSDO Data Output Fall Time[4]	—	10	25	ns	—
CS31	TcSDOR	CSDO Data Output Rise Time[4]	—	10	25	ns	—
CS35	TDV	Clock edge to CSDO data valid	—	—	10	ns	—
CS36	TDIV	Clock edge to CSDO tri-stated	10	—	20	ns	—
CS40	TCSDI	Setup time of CSDI data input to CSCK edge (CSCK pin is input or output)	20	—	—	ns	—
CS41	THCSDI	Hold time of CSDI data input to CSCK edge (CSCK pin is input or output)	20	—	—	ns	—
CS50	TcOFSF	COFS Fall Time (COFS pin is output)	—	10	25	ns	**Note 1**
CS51	TcOFSR	COFS Rise Time (COFS pin is output)	—	10	25	ns	**Note 1**
CS55	TscOFS	Setup time of COFS data input to CSCK edge (COFS pin is input)	20	—	—	ns	—
CS56	THCOFS	Hold time of COFS data input to CSCK edge (COFS pin is input)	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 CSCK is 100 ns. Therefore, the clock generated in Master mode must not violate this specification.
4: Assumes 50 pF load on all DCI pins.

FIGURE 20-13: DCI MODULE (AC-LINK MODE) TIMING CHARACTERISTICS

TABLE 20-30: DCI MODULE (AC-LINK MODE) TIMING REQUIREMENTS

AC CHARACTERISTICS			**Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic[1][2]**	**Min**	**Typ[3]**	**Max**	**Units**	**Conditions**
CS60	TBCLKL	BIT_CLK Low Time	36	40.7	45	ns	—
CS61	TBCLKH	BIT_CLK High Time	36	40.7	45	ns	—
CS62	TBCLK	BIT_CLK Period	—	81.4	—	ns	Bit clock is input
CS65	TSACL	Input Setup Time to Falling Edge of BIT_CLK	—	—	10	ns	—
CS66	THACL	Input Hold Time from Falling Edge of BIT_CLK	—	—	10	ns	—
CS70	TSYNCLO	SYNC Data Output Low Time	—	19.5	—	μs	**Note 1**
CS71	TSYNCHI	SYNC Data Output High Time	—	1.3	—	μs	**Note 1**
CS72	TSYNC	SYNC Data Output Period	—	20.8	—	μs	**Note 1**
CS75	TRACL	Rise Time, SYNC, SDATA_OUT	—	10	25	ns	CLOAD = 50 pF, VDD = 5V
CS76	TFACL	Fall Time, SYNC, SDATA_OUT	—	10	25	ns	CLOAD = 50 pF, VDD = 5V
CS77	TRACL	Rise Time, SYNC, SDATA_OUT	—	TBD	TBD	ns	CLOAD = 50 pF, VDD = 3V
CS78	TFACL	Fall Time, SYNC, SDATA_OUT	—	TBD	TBD	ns	CLOAD = 50 pF, VDD = 3V
CS80	TOVDACL	Output valid delay from rising edge of BIT_CLK	—	—	15	ns	—

Note 1: These parameters are characterized but not tested in manufacturing.

2: These values assume BIT_CLK frequency is 12.288 MHz.

3: Data in "Typ" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

FIGURE 20-14: SPI MODULE MASTER MODE (CKE = 0) TIMING CHARACTERISTICS

Note: Refer to Figure 20-3 for load conditions.
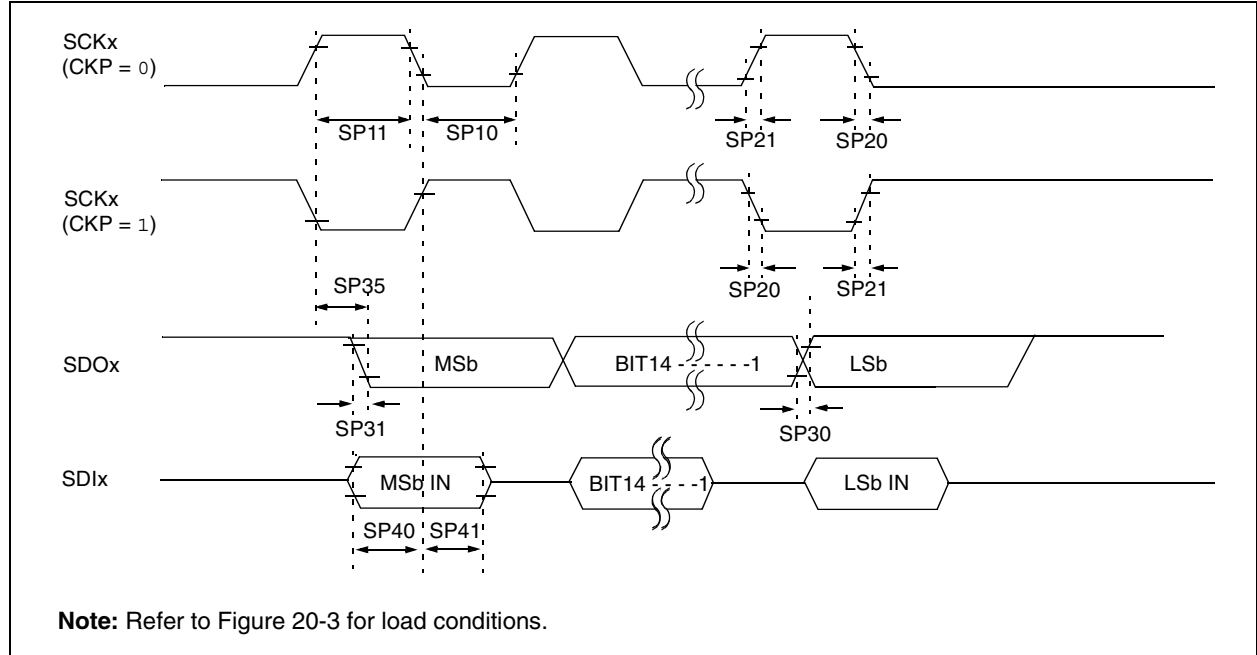
TABLE 20-31: SPI MASTER MODE (CKE = 0) TIMING REQUIREMENTS
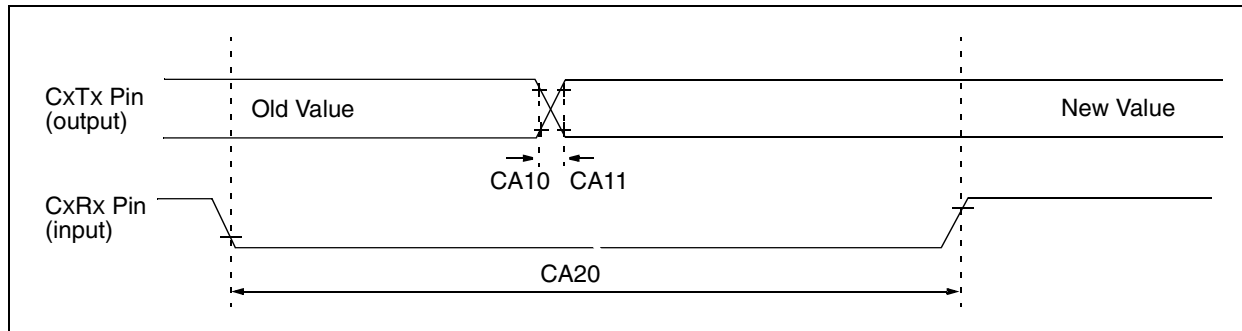
AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic[1]	Min	Typ[2]	Max	Units	Conditions
SP10	TscL	SCKx Output Low Time[3]	T_{CY} / 2	—	—	ns	—
SP11	TscH	SCKx Output High Time[3]	T_{CY} / 2	—	—	ns	—
SP20	TscF	SCKx Output Fall Time[4]	—	10	25	ns	—
SP21	TscR	SCKx Output Rise Time[4]	—	10	25	ns	—
SP30	TdoF	SDOx Data Output Fall Time[4]	—	10	25	ns	—
SP31	TdoR	SDOx Data Output Rise Time[4]	—	10	25	ns	—
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.

4: Assumes 50 pF load on all SPI pins.

FIGURE 20-15: **SPI MODULE MASTER MODE (CKE =1) TIMING CHARACTERISTICS**

Note: Refer to Figure 20-3 for load conditions.
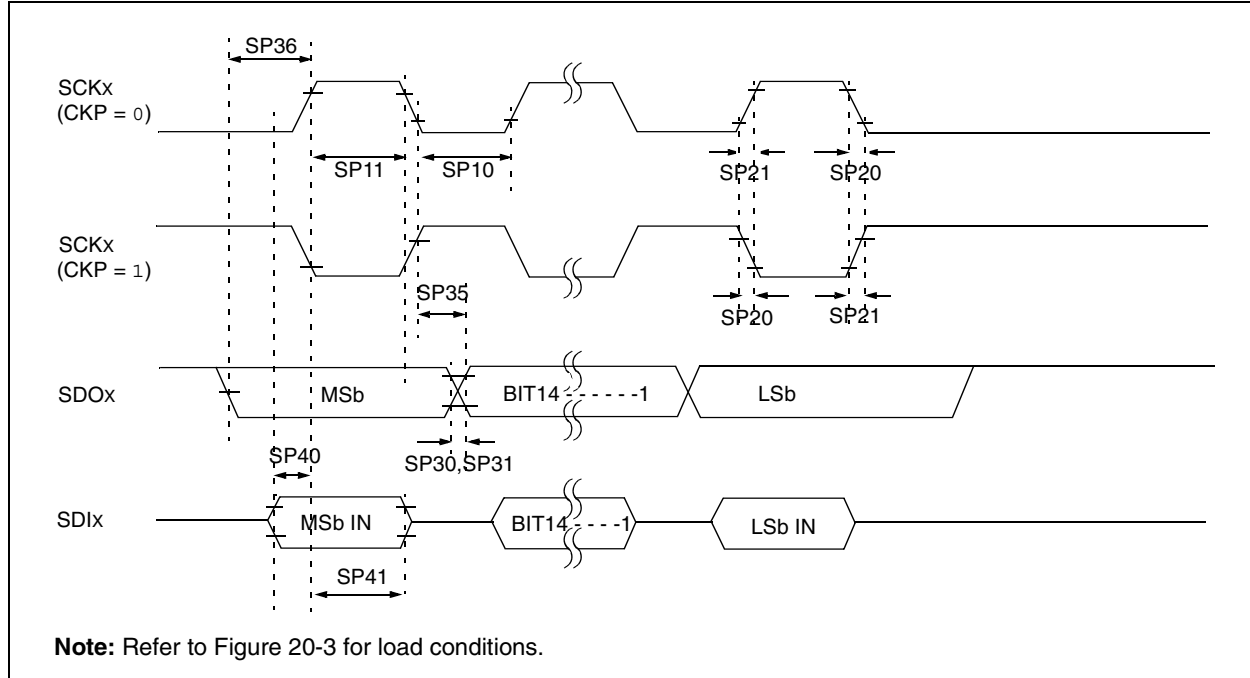
TABLE 20-32: SPI MODULE MASTER MODE (CKE = 1) TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic[1]	Min	Typ[2]	Max	Units	Conditions
SP10	TscL	SCKx output low time[3]	TCY / 2	—	—	ns	—
SP11	TscH	SCKx output high time[3]	TCY / 2	—	—	ns	—
SP20	TscF	SCKx output fall time[4]	—	10	25	ns	—
SP21	TscR	SCKx output rise time[4]	—	10	25	ns	—
SP30	TdoF	SDOx data output fall time[4]	—	10	25	ns	—
SP31	TdoR	SDOx data output rise time[4]	—	10	25	ns	—
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.

4: Assumes 50 pF load on all SPI pins.

FIGURE 20-16: **SPI MODULE SLAVE MODE (CKE = 0) TIMING CHARACTERISTICS**

Note: Refer to Figure 20-3 for load conditions.
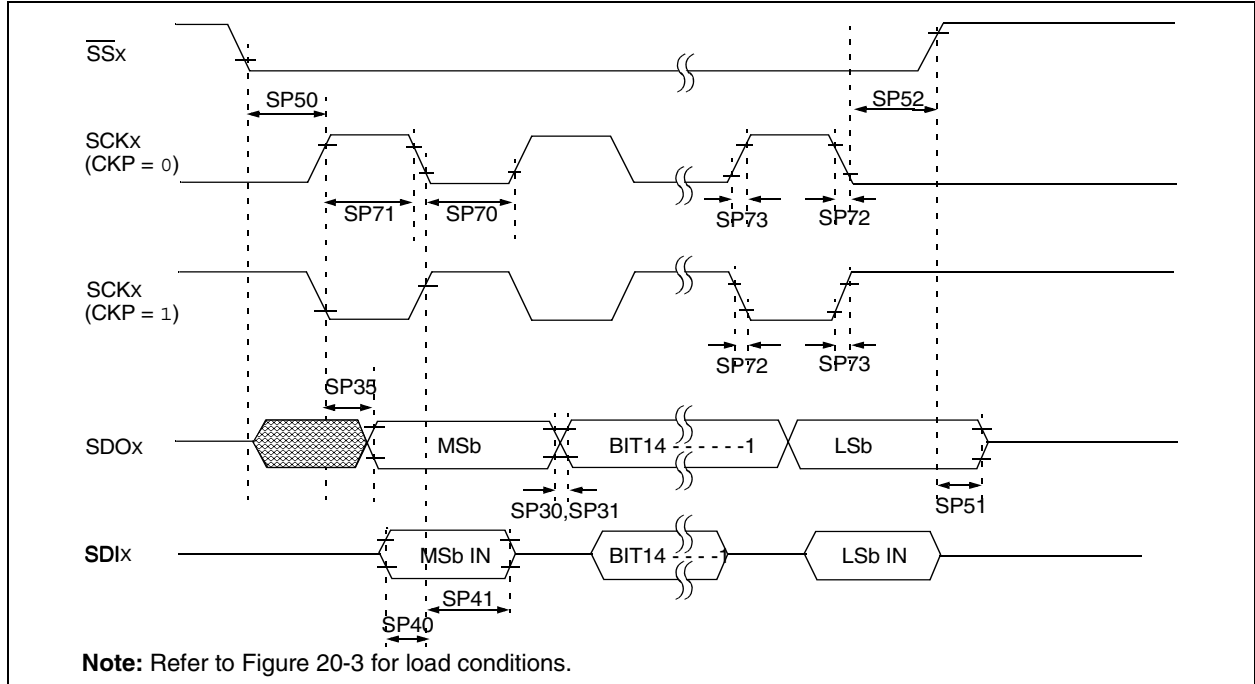
TABLE 20-33: SPI MODULE SLAVE MODE (CKE = 0) TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic[1]	Min	Typ[2]	Max	Units	Conditions
SP70	TscL	SCKx Input Low Time	30	—	—	ns	—
SP71	TscH	SCKx Input High Time	30	—	—	ns	—
SP72	TscF	SCKx Input Fall Time[3]	—	10	25	ns	—
SP73	TscR	SCKx Input Rise Time[3]	—	10	25	ns	—
SP30	TdoF	SDOx Data Output Fall Time[3]	—	10	25	ns	—
SP31	TdoR	SDOx Data Output Rise Time[3]	—	10	25	ns	—
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	—	30	ns	—
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	20	—	—	ns	—
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	20	—	—	ns	—
SP50	TssL2scH, TssL2scL	$\overline{SS}$x↓ to SCKx↑ or SCKx↓ Input	120	—	—	ns	—
SP51	TssH2doZ	$\overline{SS}$x↑ to SDOx Output Hi-Impedance[3]	10	—	50	ns	—
SP52	TscH2ssH TscL2ssH	$\overline{SS}$x 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.

3: Assumes 50 pF load on all SPI pins.

FIGURE 20-17: SPI MODULE SLAVE MODE (CKE = 1) TIMING CHARACTERISTICS

Note: Refer to Figure 20-3 for load conditions.
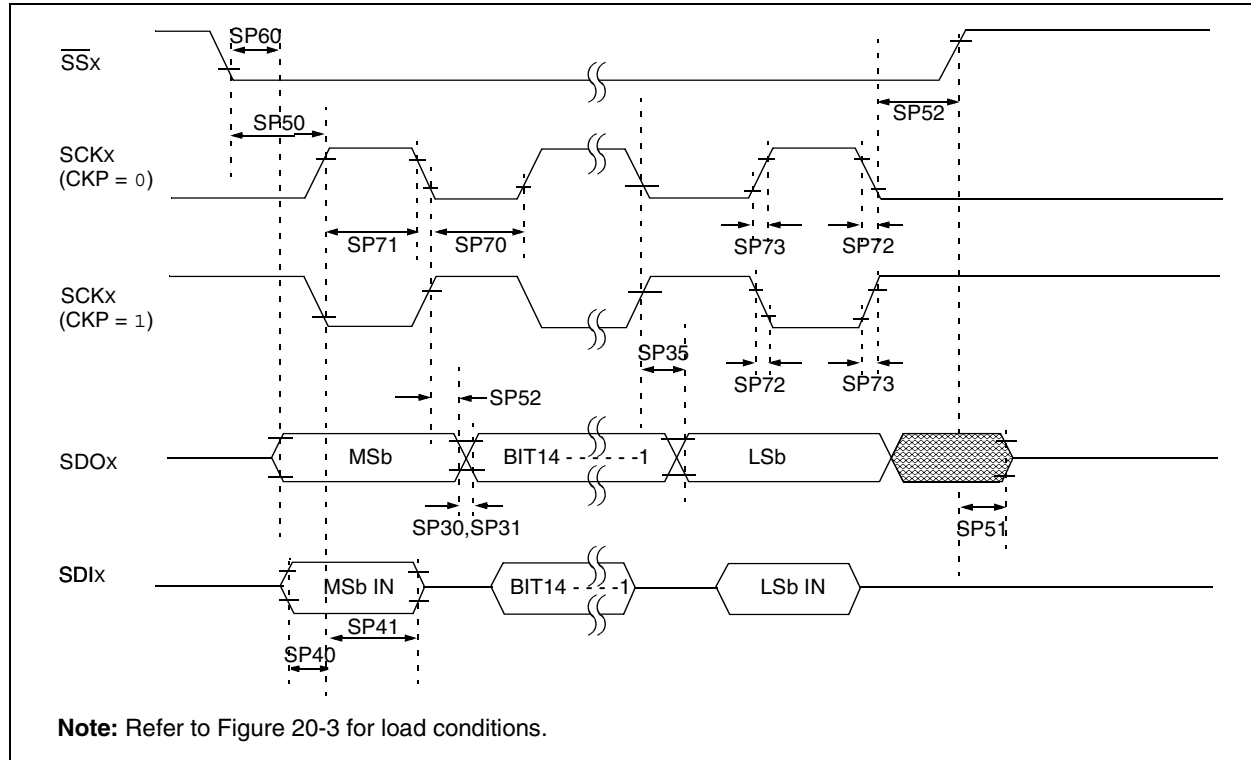
TABLE 20-34: SPI MODULE SLAVE MODE (CKE = 1) TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic[1]	Min	Typ[2]	Max	Units	Conditions
SP70	TscL	SCKx Input Low Time	30	—	—	ns	—
SP71	TscH	SCKx Input High Time	30	—	—	ns	—
SP72	TscF	SCKx Input Fall Time[3]	—	10	25	ns	—
SP73	TscR	SCKx Input Rise Time[3]	—	10	25	ns	—
SP30	TdoF	SDOx Data Output Fall Time[3]	—	10	25	ns	—
SP31	TdoR	SDOx Data Output Rise Time[3]	—	10	25	ns	—
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	—	30	ns	—
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	20	—	—	ns	—
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	20	—	—	ns	—
SP50	TssL2scH, TssL2scL	$\overline{SS}$x↓ to SCKx↓ or SCKx↑ input	120	—	—	ns	—
SP51	TssH2doZ	$\overline{SS}$↑ to SDOx Output Hi-Impedance[4]	10	—	50	ns	—
SP52	TscH2ssH TscL2ssH	$\overline{SS}$x↑ after SCKx Edge	1.5 TCY + 40	—	—	ns	—
SP60	TssL2doV	SDOx Data Output Valid after SCKx Edge	—	—	50	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.

4: Assumes 50 pF load on all SPI pins.

FIGURE 20-18: I²C BUS START/STOP BITS TIMING CHARACTERISTICS (MASTER MODE)

Note: Refer to Figure 20-3 for load conditions.
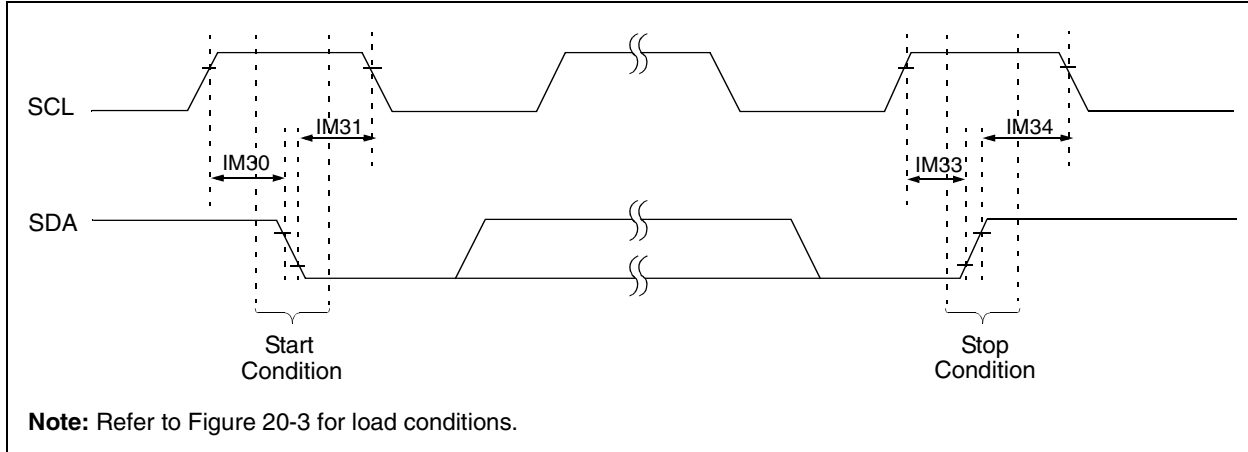
FIGURE 20-19: I²C BUS DATA TIMING CHARACTERISTICS (MASTER MODE)

Note: Refer to Figure 20-3 for load conditions.
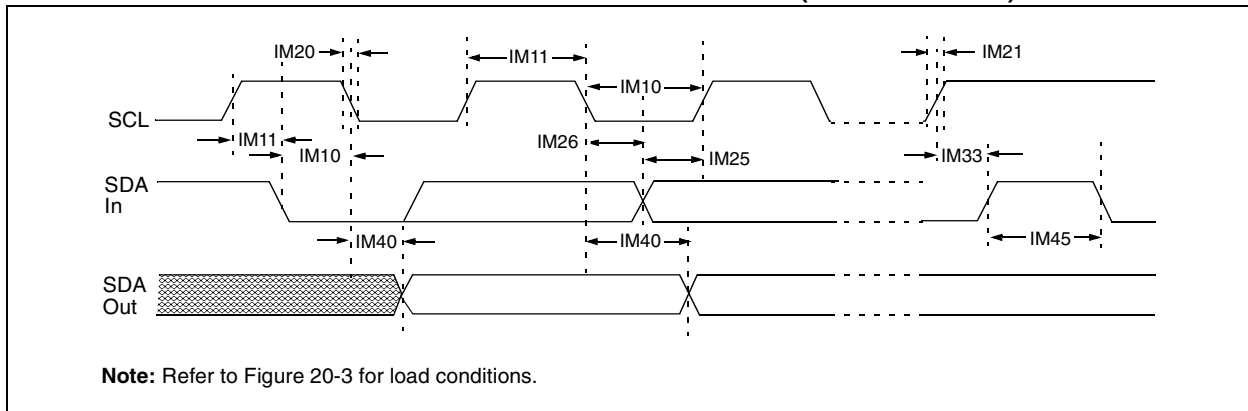
TABLE 20-35: I²C BUS DATA TIMING REQUIREMENTS (MASTER MODE)

AC CHARACTERISTICS				Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Param No.	Symbol	Characteristic		Min[1]	Max	Units	Conditions
IM10	TLO:SCL	Clock Low Time	100 kHz mode	TCY / 2 (BRG + 1)	—	µs	—
			400 kHz mode	TCY / 2 (BRG + 1)	—	µs	—
			1 MHz mode[2]	TCY / 2 (BRG + 1)	—	µs	—
IM11	THI:SCL	Clock High Time	100 kHz mode	TCY / 2 (BRG + 1)	—	µs	—
			400 kHz mode	TCY / 2 (BRG + 1)	—	µs	—
			1 MHz mode[2]	TCY / 2 (BRG + 1)	—	µs	—
IM20	TF:SCL	SDA and SCL Fall Time	100 kHz mode	—	300	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode[2]	—	100	ns	
IM21	TR:SCL	SDA and SCL Rise Time	100 kHz mode	—	1000	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode[2]	—	300	ns	
IM25	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	—
			400 kHz mode	100	—	ns	
			1 MHz mode[2]	TBD	—	ns	
IM26	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	—
			400 kHz mode	0	0.9	µs	
			1 MHz mode[2]	TBD	—	ns	
IM30	TSU:STA	Start Condition Setup Time	100 kHz mode	TCY / 2 (BRG + 1)	—	µs	Only relevant for repeated Start condition
			400 kHz mode	TCY / 2 (BRG + 1)	—	µs	
			1 MHz mode[2]	TCY / 2 (BRG + 1)	—	µs	
IM31	THD:STA	Start Condition Hold Time	100 kHz mode	TCY / 2 (BRG + 1)	—	µs	After this period the first clock pulse is generated
			400 kHz mode	TCY / 2 (BRG + 1)	—	µs	
			1 MHz mode[2]	TCY / 2 (BRG + 1)	—	µs	
IM33	TSU:STO	Stop Condition Setup Time	100 kHz mode	TCY / 2 (BRG + 1)	—	µs	—
			400 kHz mode	TCY / 2 (BRG + 1)	—	µs	
			1 MHz mode[2]	TCY / 2 (BRG + 1)	—	µs	
IM34	THD:STO	Stop Condition Hold Time	100 kHz mode	TCY / 2 (BRG + 1)	—	ns	—
			400 kHz mode	TCY / 2 (BRG + 1)	—	ns	
			1 MHz mode[2]	TCY / 2 (BRG + 1)	—	ns	
IM40	TAA:SCL	Output Valid From Clock	100 kHz mode	—	3500	ns	—
			400 kHz mode	—	1000	ns	—
			1 MHz mode[2]	—	—	ns	—
IM45	TBF:SDA	Bus Free Time	100 kHz mode	4.7	—	µs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	µs	
			1 MHz mode[2]	TBD	—	µs	
IM50	CB	Bus Capacitive Loading		—	400	pF	

Note 1: BRG is the value of the I²C Baud Rate Generator. Refer to **Section 21 "Inter-Integrated Circuit™ (I²C)"** in the *dsPIC30F Family Reference Manual.*

   2: Maximum pin capacitance = 10 pF for all I²C pins (for 1 MHz mode only).

FIGURE 20-20: I²C BUS START/STOP BITS TIMING CHARACTERISTICS (SLAVE MODE)
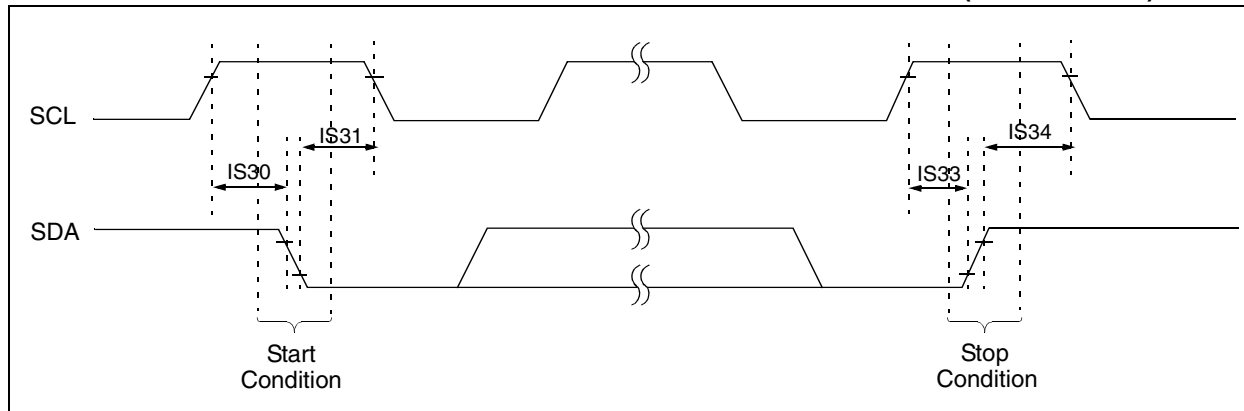
FIGURE 20-21: I²C BUS DATA TIMING CHARACTERISTICS (SLAVE MODE)
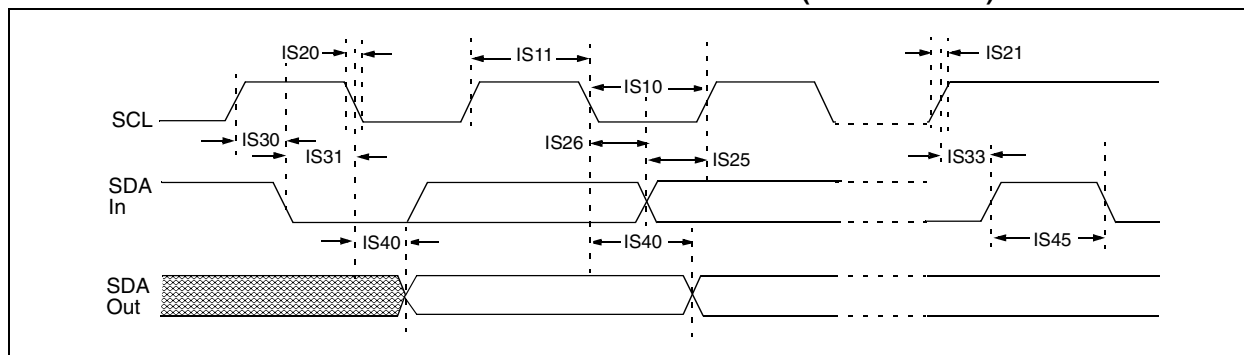
TABLE 20-36: I²C BUS DATA TIMING REQUIREMENTS (SLAVE MODE)

AC CHARACTERISTICS			**Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**		**Min**	**Max**	**Units**	**Conditions**
IS10	TLO:SCL	Clock Low Time	100 kHz mode	4.7	—	µs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	µs	Device must operate at a minimum of 10 MHz.
			1 MHz mode[1]	0.5	—	µs	—
IS11	THI:SCL	Clock High Time	100 kHz mode	4.0	—	µs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	µs	Device must operate at a minimum of 10 MHz
			1 MHz mode[1]	0.5	—	µs	—
IS20	TF:SCL	SDA and SCL Fall Time	100 kHz mode	—	300	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode[1]	—	100	ns	
IS21	TR:SCL	SDA and SCL Rise Time	100 kHz mode	—	1000	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode[1]	—	300	ns	
IS25	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	—
			400 kHz mode	100	—	ns	
			1 MHz mode[1]	100	—	ns	
IS26	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	—
			400 kHz mode	0	0.9	µs	
			1 MHz mode[1]	0	0.3	µs	
IS30	TSU:STA	Start Condition Setup Time	100 kHz mode	4.7	—	µs	Only relevant for repeated Start condition
			400 kHz mode	0.6	—	µs	
			1 MHz mode[1]	0.25	—	µs	
IS31	THD:STA	Start Condition Hold Time	100 kHz mode	4.0	—	µs	After this period the first clock pulse is generated
			400 kHz mode	0.6	—	µs	
			1 MHz mode[1]	0.25	—	µs	
IS33	TSU:STO	Stop Condition Setup Time	100 kHz mode	4.7	—	µs	—
			400 kHz mode	0.6	—	µs	
			1 MHz mode[1]	0.6	—	µs	
IS34	THD:STO	Stop Condition Hold Time	100 kHz mode	4000	—	ns	—
			400 kHz mode	600	—	ns	
			1 MHz mode[1]	250		ns	
IS40	TAA:SCL	Output Valid From Clock	100 kHz mode	0	3500	ns	—
			400 kHz mode	0	1000	ns	
			1 MHz mode[1]	0	350	ns	
IS45	TBF:SDA	Bus Free Time	100 kHz mode	4.7	—	µs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	µs	
			1 MHz mode[1]	0.5	—	µs	
IS50	CB	Bus Capacitive Loading		—	400	pF	—

Note 1: Maximum pin capacitance = 10 pF for all I²C pins (for 1 MHz mode only).

FIGURE 20-22: **CAN MODULE I/O TIMING CHARACTERISTICS**

TABLE 20-37: **CAN MODULE I/O TIMING REQUIREMENTS**

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Param No.	Symbol	Characteristic[1]	Min	Typ[2]	Max	Units	Conditions
CA10	TioF	Port Output Fall Time	—	10	25	ns	—
CA11	TioR	Port Output Rise Time	—	10	25	ns	—
CA20	Tcwf	Pulse Width to Trigger CAN Wakeup Filter	500			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.

TABLE 20-38: 12-BIT A/D MODULE SPECIFICATIONS

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic	Min.	Typ	Max.	Units	Conditions
Device Supply							
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	µA µA	A/D operating A/D off
Analog Input							
AD10	VINH-VINL	Full-Scale Input Span	VREFL		VREFH	V	See Note
AD11	VIN	Absolute Input Voltage	AVSS - 0.3		AVDD + 0.3	V	—
AD12	—	Leakage Current	—	±0.001	±0.610	µA	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V Source Impedance = 2.5 kΩ
AD13	—	Leakage Current	—	±0.001	±0.610	µA	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V Source Impedance = 2.5 kΩ
AD15	RSS	Switch Resistance	—	3.2K	—	Ω	—
AD16	CSAMPLE	Sample Capacitor	—	18		pF	—
AD17	RIN	Recommended Impedance of Analog Voltage Source	—	—	2.5K	Ω	—
DC Accuracy							
AD20	Nr	Resolution	12 data bits			bits	
AD21	INL	Integral Nonlinearity	—	±0.75	TBD	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V
AD21A	INL	Integral Nonlinearity	—	±0.75	TBD	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD22	DNL	Differential Nonlinearity	—	±0.5	< ±1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V
AD22A	DNL	Differential Nonlinearity	—	±0.5	< ±1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD23	GERR	Gain Error	—	±1.25	TBD	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V
AD23A	GERR	Gain Error	—	±1.25	TBD	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V

Note 1: The A/D conversion result never decreases with an increase in the input voltage, and has no missing codes.

TABLE 20-38: 12-BIT A/D MODULE SPECIFICATIONS (CONTINUED)

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic	Min.	Typ	Max.	Units	Conditions
AD24	E_{OFF}	Offset Error	—	±1.25	TBD	LSb	$V_{INL} = AV_{SS} = V_{REFL} = 0V, AV_{DD} = V_{REFH} = 5V$
AD24A	E_{OFF}	Offset Error	—	±1.25	TBD	LSb	$V_{INL} = AV_{SS} = V_{REFL} = 0V, AV_{DD} = V_{REFH} = 3V$
AD25	—	Monotonicity[1]	—	—	—	—	Guaranteed
AD26	CMRR	Common-Mode Rejection	—	TBD	—	dB	—
AD27	PSRR	Power Supply Rejection Ratio	—	TBD	—	dB	—
AD28	CTLK	Channel to Channel Crosstalk	—	TBD	—	dB	—
Dynamic Performance							
AD30	THD	Total Harmonic Distortion	—	—	—	dB	—
AD31	SINAD	Signal to Noise and Distortion	—	TBD	—	dB	—
AD32	SFDR	Spurious Free Dynamic Range	—	TBD	—	dB	—
AD33	F_{NYQ}	Input Signal Bandwidth	—	—	50	kHz	—
AD34	ENOB	Effective Number of Bits	—	TBD	TBD	bits	—

Note 1: The A/D conversion result never decreases with an increase in the input voltage, and has no missing codes.

FIGURE 20-23: **12-BIT A/D CONVERSION TIMING CHARACTERISTICS (ASAM = 0, SSRC = 000)**

① - Software sets ADCON. SAMP to start sampling.

② - Sampling starts after discharge period.
TSAMP is described in the *dsPIC30F Family Reference Manual*, Section 18.

③ - Software clears ADCON. SAMP to start conversion.

④ - Sampling ends, conversion sequence starts.

⑤ - Convert bit 11.

⑥ - Convert bit 10.

⑦ - Convert bit 1.

⑧ - Convert bit 0.

⑨ - One TAD for end of conversion.
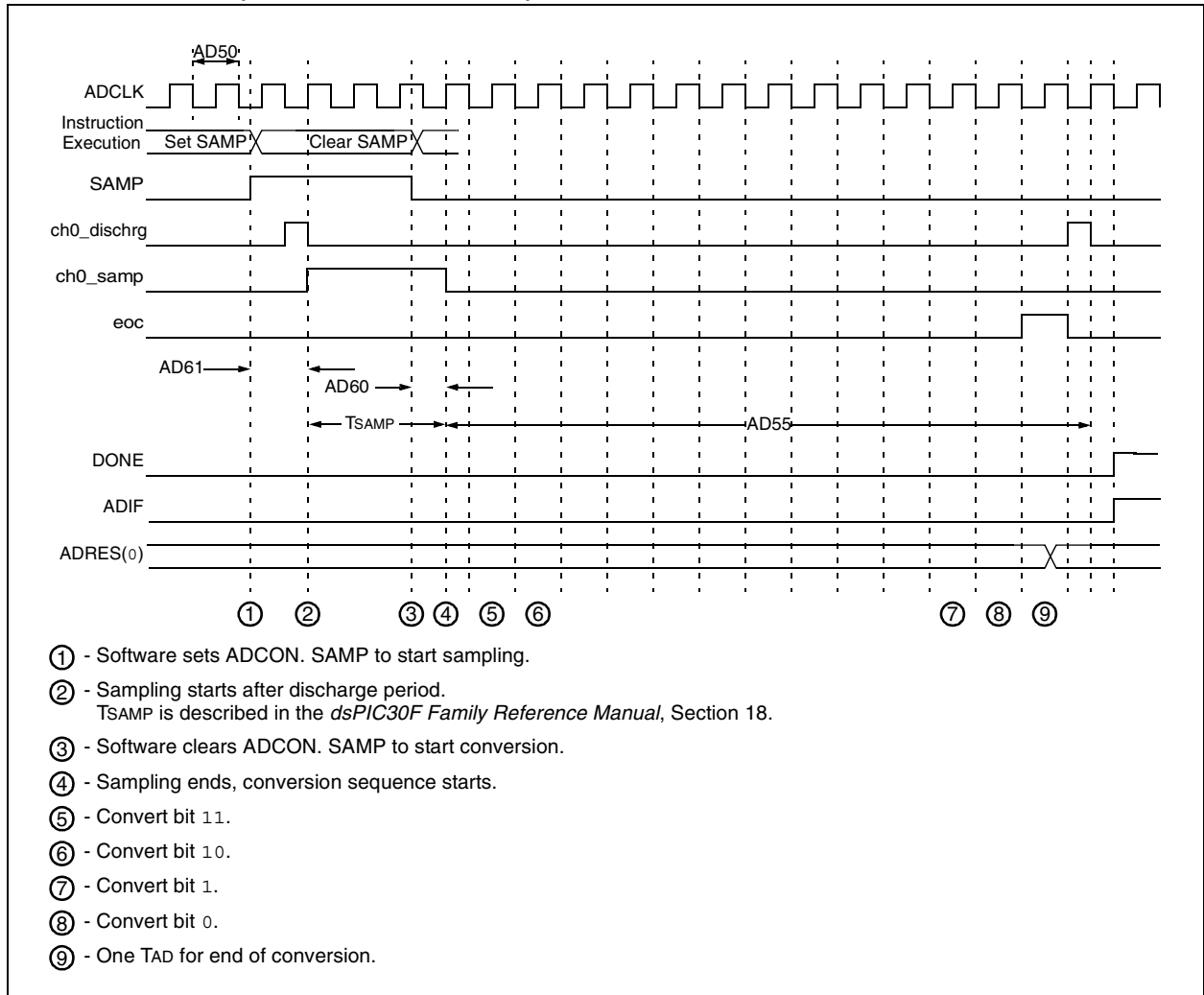
TABLE 20-39: 12-BIT A/D CONVERSION TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.7V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic	Min.	Typ	Max.	Units	Conditions
Clock Parameters							
AD50	TAD	A/D Clock Period	—	667	—	ns	VDD = 3-5.5V **(Note 1)**
AD51	tRC	A/D Internal RC Oscillator Period	1.2	1.5	1.8	μs	—
Conversion Rate							
AD55	tCONV	Conversion Time	—	14 TAD	—	ns	—
AD56	FCNV	Throughput Rate	—	—	100	ksps	VDD = VREF = 5V
AD57	TSAMP	Sampling Time	—	1 TAD	—	ns	VDD = 3-5.5V source resistance RS = 0-2.5 kΩ
Timing Parameters							
AD60	tPCS	Conversion Start from Sample Trigger	—	—	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)	—	—	TBD	ns	—
AD63	tDPU	Time to Stabilize Analog Stage from A/D Off to A/D On	—	—	TBD	μs	—

Note 1: Because the sample caps will eventually lose charge, clock rates below 10 kHz can affect linearity performance, especially at elevated temperatures.

   2: These parameters are characterized but not tested in manufacturing.

21.0   PACKAGING INFORMATION

21.1   Package Marking Information

18-Lead PDIP

```
XXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX
     YYWWNNN
```

Example

```
dsPIC30F3012
30I/P(e3)
     0510017
```

18-Lead SOIC

```
XXXXXXXXXXX
XXXXXXXXXXX
XXXXXXXXXXX
     YYWWNNN
```

Example

```
dsPIC30F2011
30I/SO(e3)

     0510017
```

28-Lead SPDIP

```
    XXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXX
     YYWWNNN
```

Example

```
    dsPIC30F2012
    30I/SP(e3)
     0510017
```

Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

* Standard PICmicro device marking consists of Microchip part number, year code, week code and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

21.2   Package Marking Information (Continued)

28-Lead SOIC (.300")

```
XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX
        YYWWNNN
```

Example

```
    dsPIC30F3013
    30I/SO e3

        0510017
```

28-Lead QFN

```
    XXXXXXX
    XXXXXXX

    YYWWNNN
```

Example

```
    30F2011
    30I/MM e3

    0510017
```

44-Lead QFN

```
    XXXXXXXXXX
    XXXXXXXXXX
    XXXXXXXXXX
    YYWWNNN
```

Example

```
    dsPIC
    30F3013
    30I/ML e3
    0510017
```
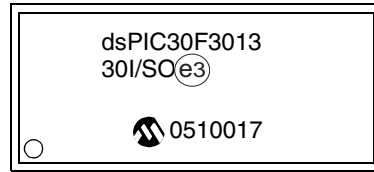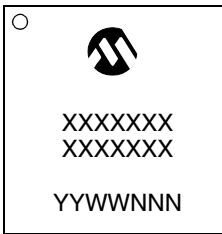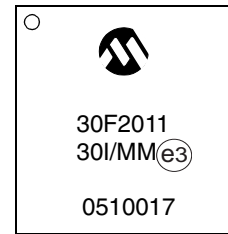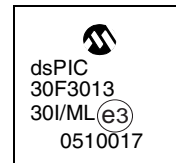
18-Lead Plastic Dual In-line (P) – 300 mil Body (PDIP)
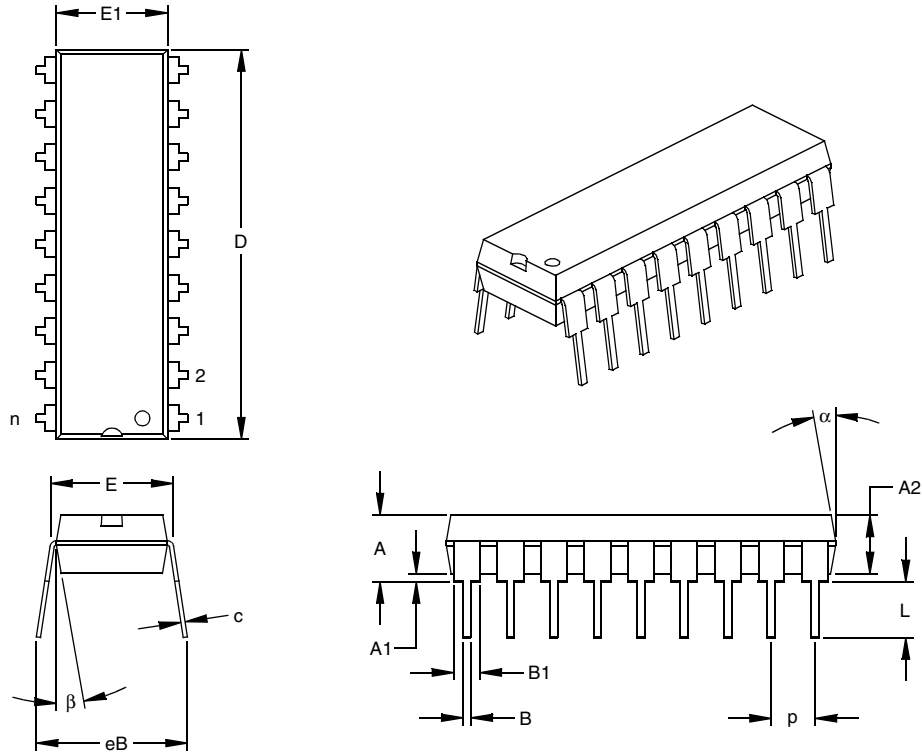
	Units	INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.890	.898	.905	22.61	22.80	22.99
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing §	eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed
.010" (0.254mm) per side.
JEDEC Equivalent: MS-001
Drawing No. C04-007

18-Lead Plastic Small Outline (SO) – Wide, 300 mil Body (SOIC)
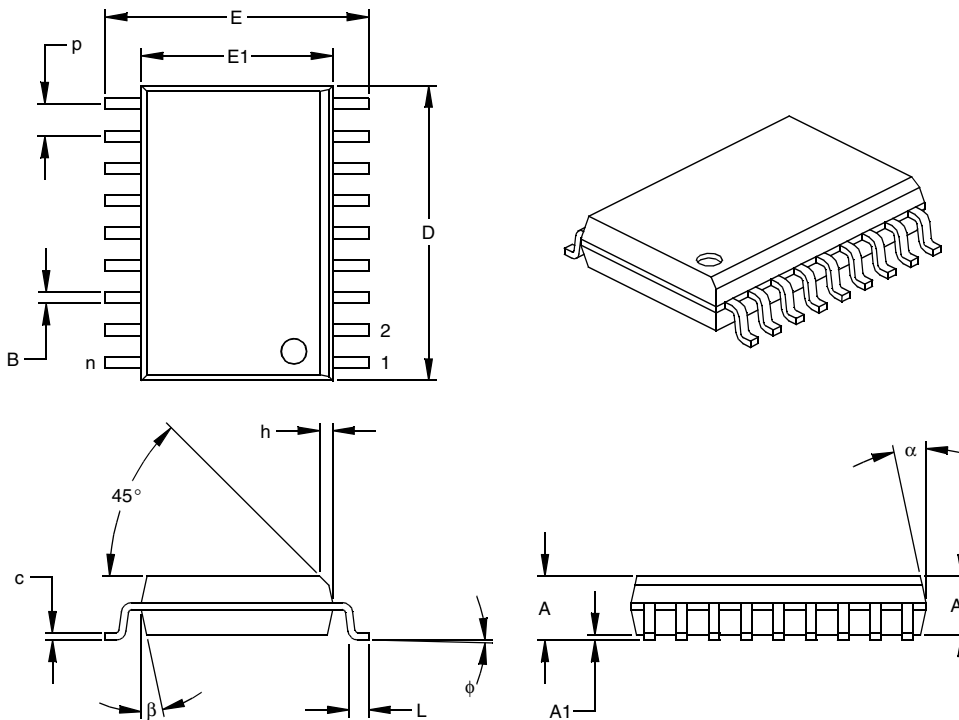
	Units	INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.050			1.27	
Overall Height	A	.093	.099	.104	2.36	2.50	2.64
Molded Package Thickness	A2	.088	.091	.094	2.24	2.31	2.39
Standoff §	A1	.004	.008	.012	0.10	0.20	0.30
Overall Width	E	.394	.407	.420	10.01	10.34	10.67
Molded Package Width	E1	.291	.295	.299	7.39	7.49	7.59
Overall Length	D	.446	.454	.462	11.33	11.53	11.73
Chamfer Distance	h	.010	.020	.029	0.25	0.50	0.74
Foot Length	L	.016	.033	.050	0.41	0.84	1.27
Foot Angle	φ	0	4	8	0	4	8
Lead Thickness	c	.009	.011	.012	0.23	0.27	0.30
Lead Width	B	.014	.017	.020	0.36	0.42	0.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed
.010" (0.254mm) per side.
JEDEC Equivalent: MS-013
Drawing No. C04-051

Preliminary

28-Lead Skinny Plastic Dual In-line (SP) – 300 mil Body (PDIP)
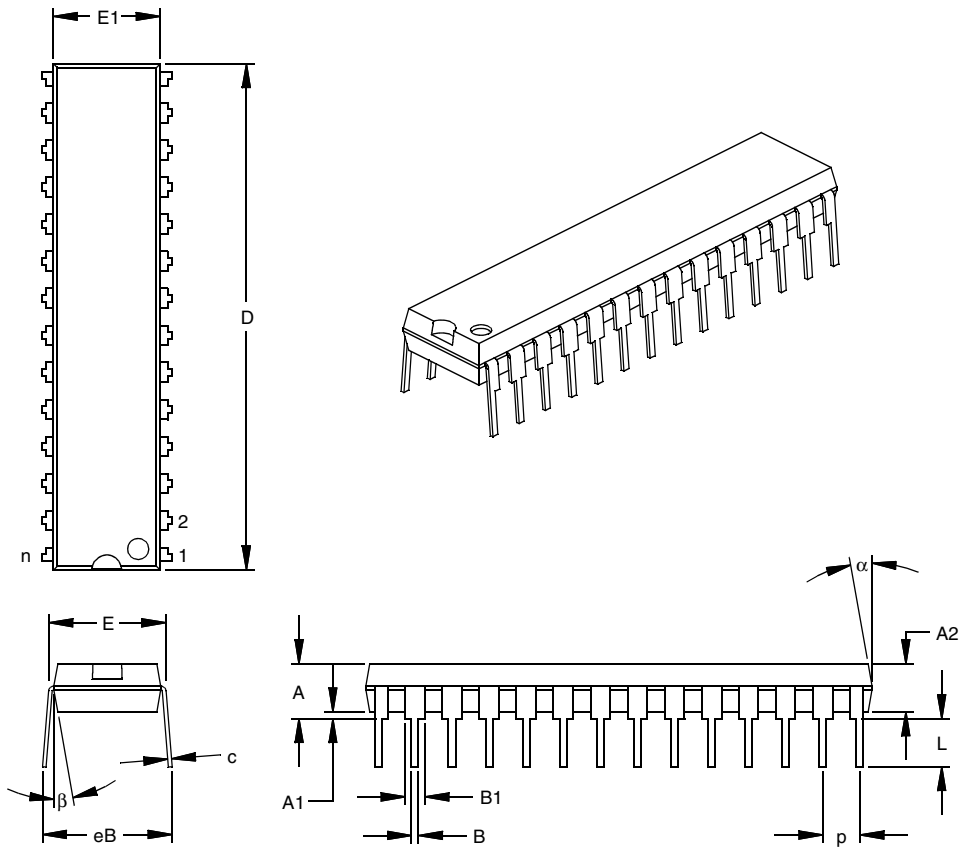
	Units	INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.150	.160	3.56	3.81	4.06
Molded Package Thickness	A2	.125	.130	.135	3.18	3.30	3.43
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.310	.325	7.62	7.87	8.26
Molded Package Width	E1	.275	.285	.295	6.99	7.24	7.49
Overall Length	D	1.345	1.365	1.385	34.16	34.67	35.18
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.040	.053	.065	1.02	1.33	1.65
Lower Lead Width	B	.016	.019	.022	0.41	0.48	0.56
Overall Row Spacing §	eB	.320	.350	.430	8.13	8.89	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter
§ Significant Characteristic
Notes:
Dimension D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed
.010" (0.254mm) per side.
JEDEC Equivalent: MO-095
Drawing No. C04-070

28-Lead Plastic Small Outline - Wide, 300 mil Body (SOIC)
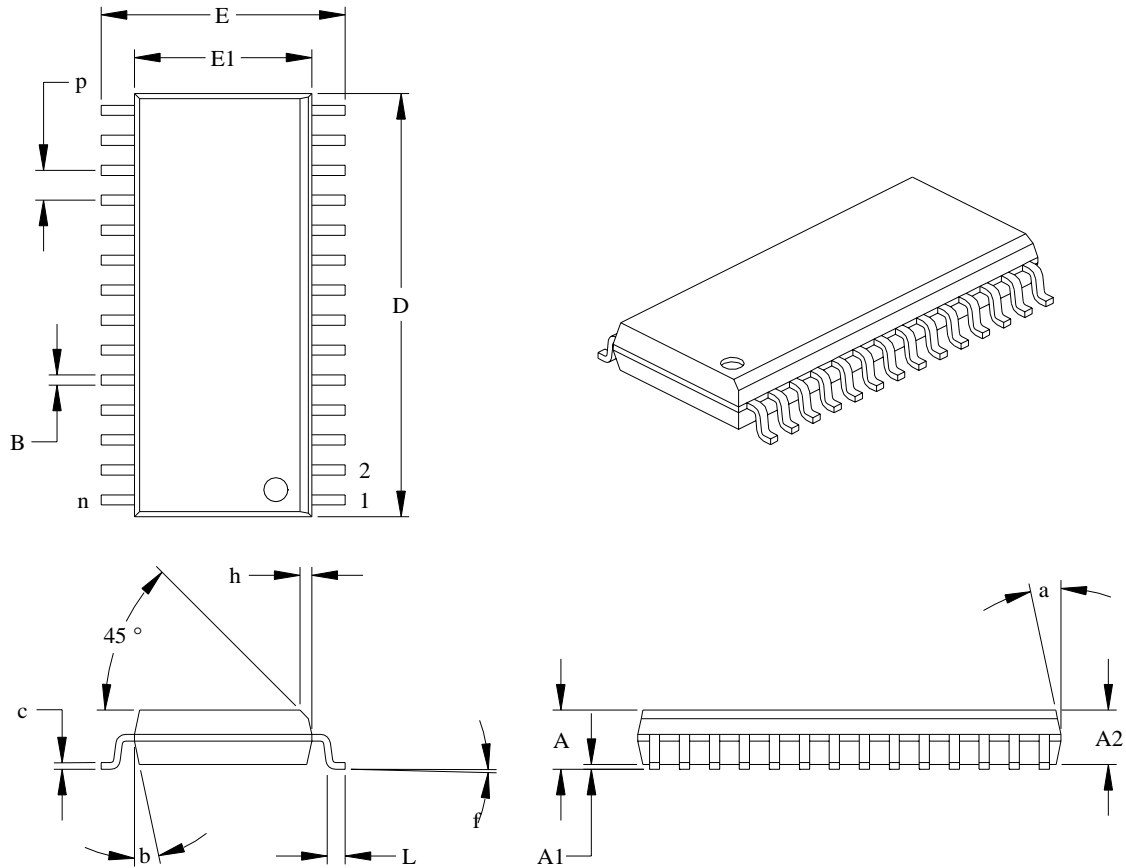
	Units		INCHES			MILLIMETERS	
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28	
Pitch	p		.050			1.27	
Overall Height	A	.093	.099	.104	2.36	2.50	2.64
Molded Package Thickness	A2	.088	.091	.094	2.24	2.31	2.39
Standoff	A1	.004	.008	.012	0.10	0.20	0.30
Overall Width	E	.394	.407	.420	10.01	10.34	10.67
Molded Package Width	E1	.288	.295	.299	7.32	7.49	7.59
Overall Length	D	.695	.704	.712	17.65	17.87	18.08
Chamfer Distance	h	.010	.020	.029	0.25	0.50	0.74
Foot Length	L	.016	.033	.050	0.41	0.84	1.27
Foot Angle Top	f	0	4	8	0	4	8
Lead Thickness	c	.009	.011	.013	0.23	0.28	0.33
Lead Width	B	.014	.017	.020	0.36	0.42	0.51
Mold Draft Angle Top	a	0	12	15	0	12	15
Mold Draft Angle Bottom	b	0	12	15	0	12	15

Drawing No. C04-052

Preliminary

28-Lead Plastic Quad Flat No Lead Package (ML) 6x6x0.9 mm Body (QFN-S) – With 0.40 mm Contact Length (Saw Singulated)
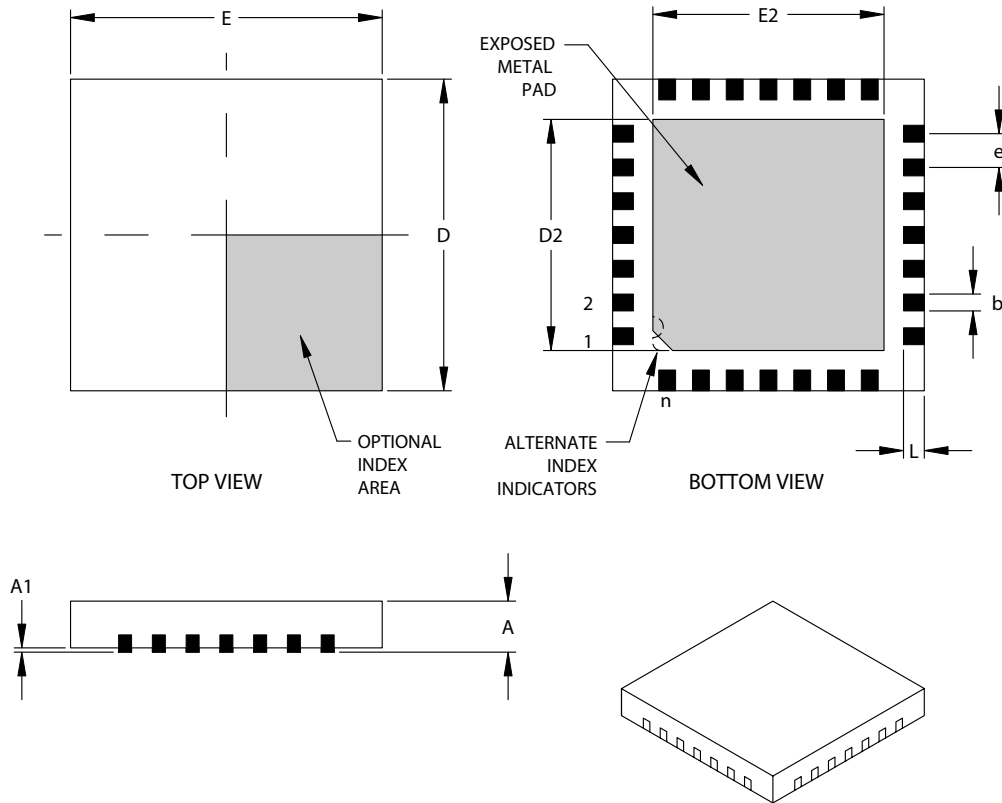
TOP VIEW

EXPOSED METAL PAD

OPTIONAL INDEX AREA

ALTERNATE INDEX INDICATORS

BOTTOM VIEW

	Units	INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28	
Pitch	e		.026 BSC			0.65 BSC	
Overall Height	A	.031	.035	.039	0.80	0.90	1.00
Standoff	A1	.000	.001	.002	0.00	0.02	0.05
Overall Width	E	.232	.236	.240	5.90	6.00	6.10
Exposed Pad Width	E2	.169	.175	.177	4.30	4.45	4.50
Overall Length	D	.232	.236	.240	5.90	6.00	6.10
Exposed Pad Length	D2	.169	.175	.177	4.30	4.45	4.50
Lead Width	b	.013	.015	.017	0.33	0.38	0.43
Lead Length	L	.012	.016	.020	0.30	0.40	0.50

*Controlling Parameter

Notes:

JEDEC equivalent: MO-220

Drawing No. C04-124

Revised 05/24/04

dsPIC30F2011/2012/3012/3013

44-Lead Plastic Quad Flat No Lead Package (ML) 8x8 mm Body (QFN)

TOP VIEW

BOTTOM VIEW

DETAIL: CONTACT VARIANTS (Note 3)
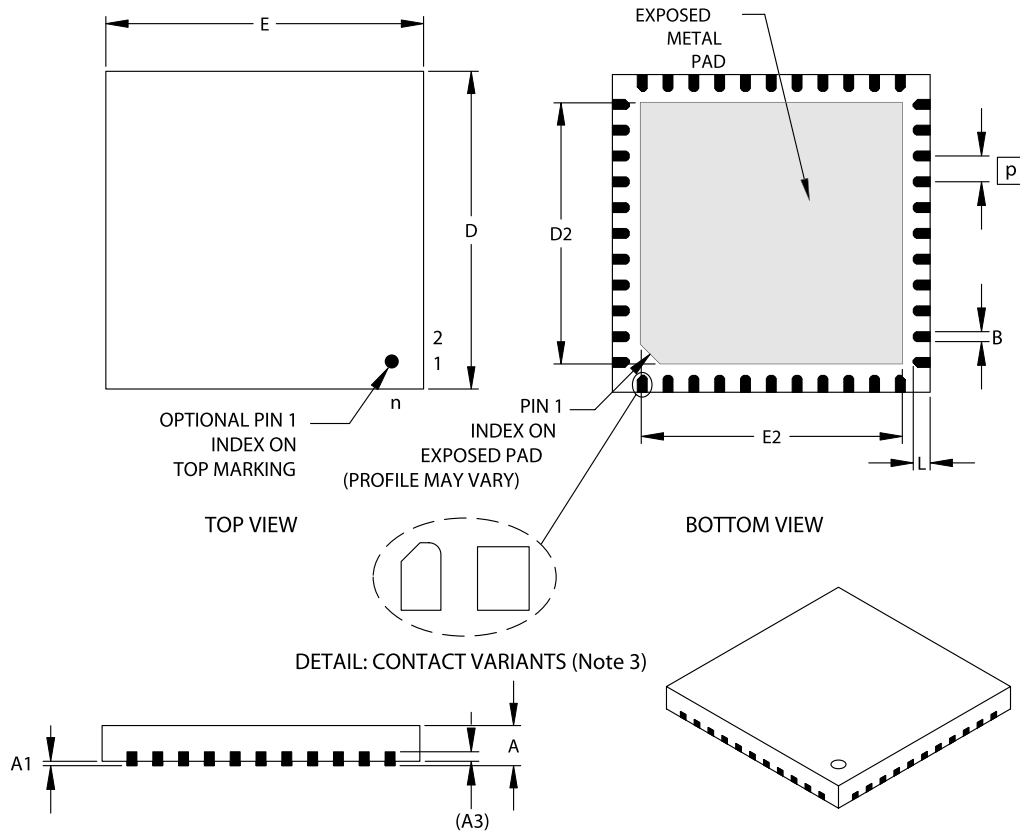
	Units	INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Contacts	n		44			44	
Pitch	p		.026 BSC [1]			0.65 BSC [1]	
Overall Height	A	.031	.035	.039	0.80	0.90	1.00
Standoff	A1	.000	.001	.002	0	0.02	0.05
Base Thickness	(A3)		.010 REF [2]			0.25 REF [2]	
Overall Width	E	.309	.315	.321	7.85	8.00	8.15
Exposed Pad Width	E2	.246	.268	.274	6.25	6.80	6.95
Overall Length	D	.309	.315	.321	7.85	8.00	8.15
Exposed Pad Length	D2	.246	.268	.274	6.25	6.80	6.95
Contact Width	B	.008	.013	.013	0.20	0.33	0.35
Contact Length	L	.014	.016	.019	0.35	0.40	0.48

*Controlling Parameter

Notes:
1. BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   See ASME Y14.5M
2. REF: Reference Dimension, usually without tolerance, for information purposes only.
   See ASME Y14.5M
3. Contact profiles may vary.

JEDEC equivalent: M0-220
Drawing No. C04-103

INDEX

dsPIC30F2011/2012/3012/3013

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

In addition, there is a Development Systems Information Line which lists the latest versions of Microchip's development systems software products. This line also provides information on how customers can receive currently available upgrade kits.
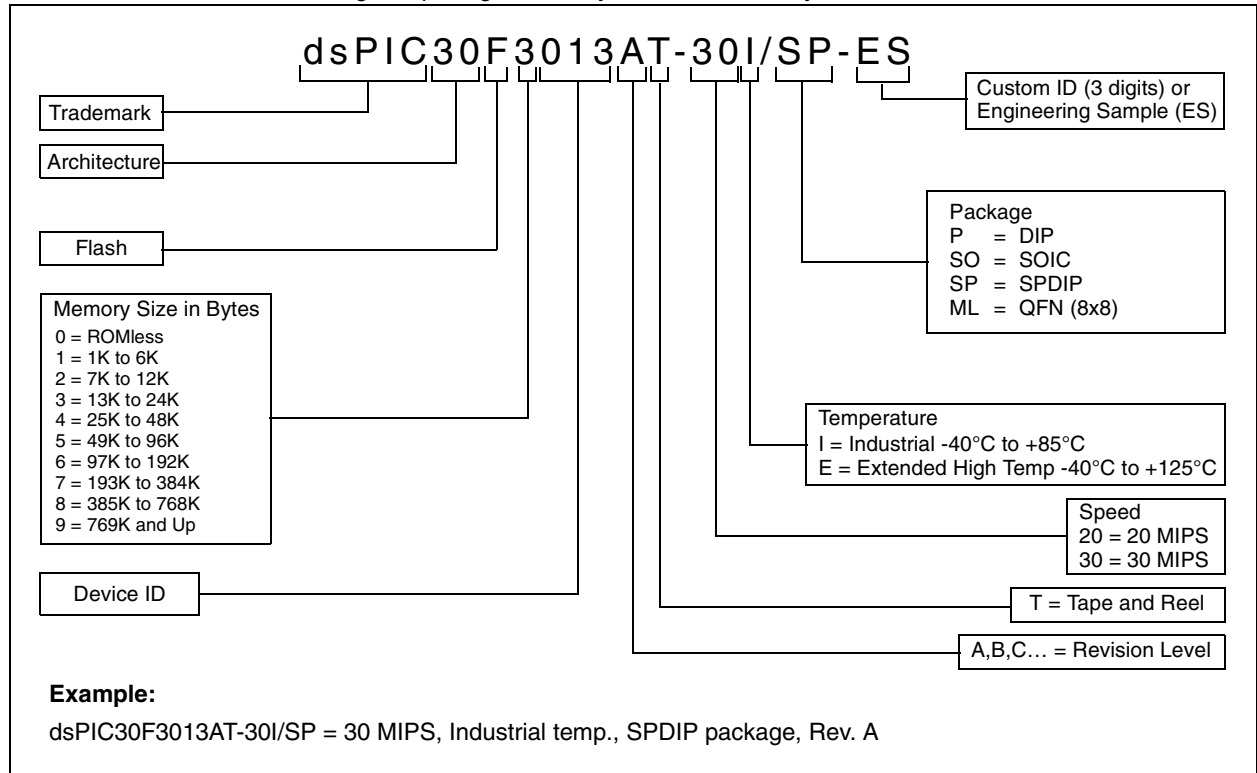
The Development Systems Information Line numbers are:

1-800-755-2345 – United States and most of Canada

1-480-792-7302 – Other International Locations

dsPIC30F2011/2012/3012/3013

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager   Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____   FAX: (_____) _____ - _____

Application (optional):

Would you like a reply?____Y ____N

Device: dsPIC30F2011/2012/3012/3013   Literature Number: DS70139C

Questions:

1. What are the best features of this document?

_____

_____

2. How does this document meet your hardware and software development needs?

_____

_____

3. Do you find the organization of this document easy to follow? If not, why?

_____

_____

4. What additions to the document do you think would enhance the structure and subject?

_____

_____

5. What deletions from the document could be made without affecting the overall usefulness?

_____

_____

6. Is there any incorrect or misleading information (what and where)?

_____

_____

7. How would you improve this document?

_____

_____

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

dsPIC30F3013AT-30I/SP-ES

Trademark

Architecture

Flash

Memory Size in Bytes
0 = ROMless
1 = 1K to 6K
2 = 7K to 12K
3 = 13K to 24K
4 = 25K to 48K
5 = 49K to 96K
6 = 97K to 192K
7 = 193K to 384K
8 = 385K to 768K
9 = 769K and Up

Device ID

Custom ID (3 digits) or
Engineering Sample (ES)

Package
P   = DIP
SO = SOIC
SP = SPDIP
ML = QFN (8x8)

Temperature
I = Industrial -40°C to +85°C
E = Extended High Temp -40°C to +125°C

Speed
20 = 20 MIPS
30 = 30 MIPS

T = Tape and Reel

A,B,C… = Revision Level

Example:

dsPIC30F3013AT-30I/SP = 30 MIPS, Industrial temp., SPDIP package, Rev. A

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
Westford, MA
Tel: 978-692-3848
Fax: 978-692-3821

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

San Jose
Mountain View, CA
Tel: 650-215-1444
Fax: 650-961-0286

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

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: 86-28-8676-6200
Fax: 86-28-8676-6599

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 - 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 - Qingdao
Tel: 86-532-502-7355
Fax: 86-532-502-7205

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-2229-0061
Fax: 91-80-2229-0062

India - New Delhi
Tel: 91-11-5160-8631
Fax: 91-11-5160-8632

Japan - Kanagawa
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Taiwan - Hsinchu
Tel: 886-3-572-9526
Fax: 886-3-572-6459

EUROPE

Austria - Weis
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark - Ballerup
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Massy
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Ismaning
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

England - Berkshire
Tel: 44-118-921-5869
Fax: 44-118-921-5820

10/20/04

Preliminary

© 2005 Microchip Technology Inc.