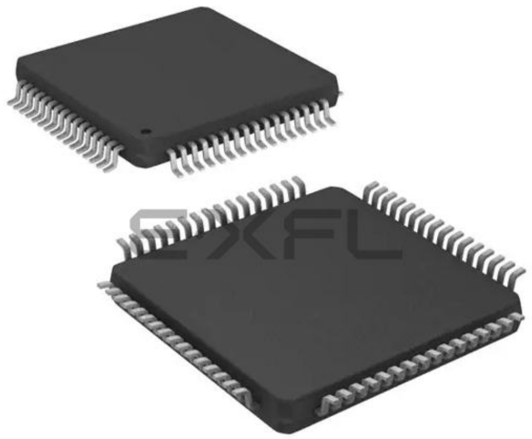


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	dsPIC
Core Size	16-Bit
Speed	20 MIPS
Connectivity	CANbus, I²C, SPI, UART/USART
Peripherals	AC'97, Brown-out Detect/Reset, I²S, LVD, POR, PWM, WDT
Number of I/O	52
Program Memory Size	66KB (22K x 24)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic30f5011t-20i-ptg

2.2 Pins Used During Programming

The pins identified in Table 2-1 are used for device programming. Refer to the appropriate device data sheet for complete pin descriptions.

TABLE 2-1: dsPIC30F PIN DESCRIPTIONS DURING PROGRAMMING

Pin Name	Pin Type	Pin Description
$\overline{MCLR}$/VPP	P	Programming Enable
VDD	P	Power Supply
VSS	P	Ground
PGC	I	Serial Clock
PGD	I/O	Serial Data

Legend: I = Input, O = Output, P = Power

2.3 Program Memory Map

The program memory space extends from 0x0 to 0xFFFFFE. Code storage is located at the base of the memory map and supports up to 144 Kbytes (48K instruction words). Code is stored in three, 48 Kbyte memory panels that reside on-chip. Table 2-2 shows the location and program memory size of each device.

Locations 0x800000 through 0x8005BE are reserved for executive code memory. This region stores either the programming executive or debugging executive. The programming executive is used for device programming, while the debug executive is used for in-circuit debugging. This region of memory cannot be used to store user code.

Locations 0xF80000 through 0xF8000E are reserved for the Configuration registers. The bits in these registers may be set to select various device options, and are described in **Section 5.7 "Configuration Bits Programming"**.

Locations 0xFF0000 and 0xFF0002 are reserved for the Device ID registers. These bits can be used by the programmer to identify what device type is being programmed and are described in **Section 10.0 "Device ID"**. The device ID reads out normally, even after code protection is applied.

Figure 2-2 illustrates the memory map for the dsPIC30F devices.

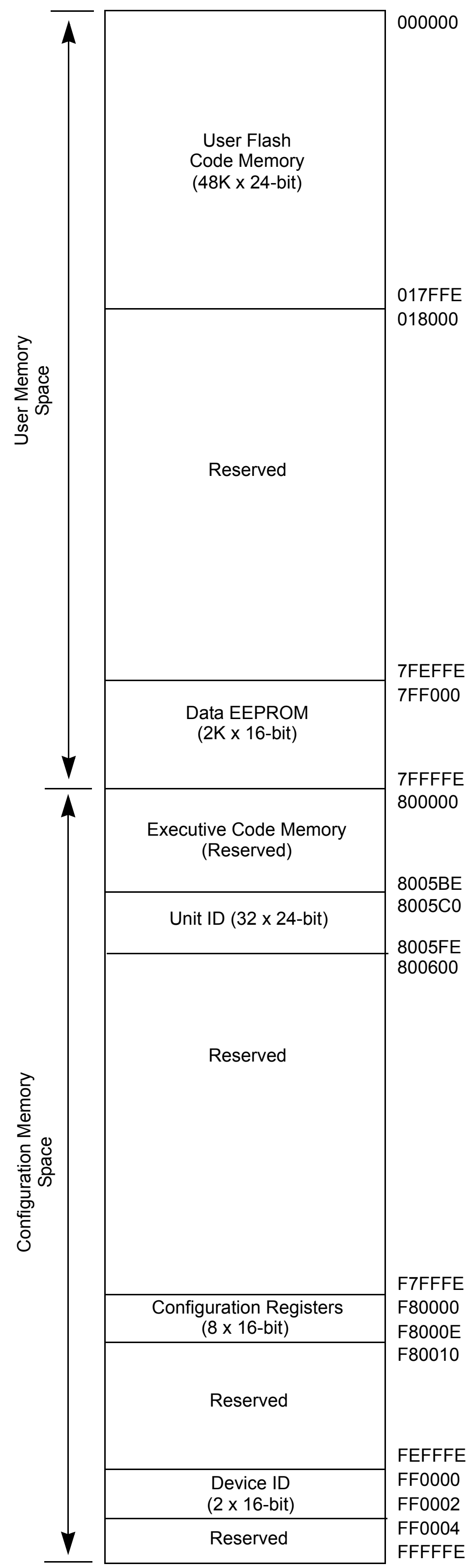
2.4 Data EEPROM Memory

The Data EEPROM array supports up to 4 Kbytes of data and is located in one memory panel. It is mapped in program memory space, residing at the end of User Memory Space (see Figure 2-2). Table 2-2 shows the location and size of data EEPROM in each device.

TABLE 2-2: CODE MEMORY AND DATA EEPROM MAP AND SIZE

Device	Code Memory map (Size in Instruction Words)	Data EEPROM Memory Map (Size in Bytes)
dsPIC30F2010	0x000000-0x001FFE (4K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F2011	0x000000-0x001FFE (4K)	None (0K)
dsPIC30F2012	0x000000-0x001FFE (4K)	None (0K)
dsPIC30F3010	0x000000-0x003FFE (8K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F3011	0x000000-0x003FFE (8K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F3012	0x000000-0x003FFE (8K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F3013	0x000000-0x003FFE (8K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F3014	0x000000-0x003FFE (8K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F4011	0x000000-0x007FFE (16K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F4012	0x000000-0x007FFE (16K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F4013	0x000000-0x007FFE (16K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F5011	0x000000-0x00AFFE (22K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F5013	0x000000-0x00AFFE (22K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F5015	0x000000-0x00AFFE (22K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F5016	0x000000-0x00AFFE (22K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F6010	0x000000-0x017FFE (48K)	0x7FF000-0x7FFFFE (4K)
dsPIC30F6010A	0x000000-0x017FFE (48K)	0x7FF000-0x7FFFFF (4K)
dsPIC30F6011	0x000000-0x015FFE (44K)	0x7FF800-0x7FFFFE (2K)
dsPIC30F6011A	0x000000-0x015FFE (44K)	0x7FF800-0x7FFFFE (2K)
dsPIC30F6012	0x000000-0x017FFE (48K)	0x7FF000-0x7FFFFE (4K)
dsPIC30F6012A	0x000000-0x017FFE (48K)	0x7FF000-0x7FFFFE (4K)
dsPIC30F6013	0x000000-0x015FFE (44K)	0x7FF800-0x7FFFFE (2K)
dsPIC30F6013A	0x000000-0x015FFE (44K)	0x7FF800-0x7FFFFE (2K)
dsPIC30F6014	0x000000-0x017FFE (48K)	0x7FF000-0x7FFFFE (4K)
dsPIC30F6014A	0x000000-0x017FFE (48K)	0x7FF000-0x7FFFFE (4K)
dsPIC30F6015	0x000000-0x017FFE (48K)	0x7FF000-0x7FFFFE (4K)

FIGURE 2-2: PROGRAM MEMORY MAP



Note: The address boundaries for user Flash code memory and data EEPROM are device-dependent.

5.0 DEVICE PROGRAMMING

5.1 Overview of the Programming Process

Once the programming executive has been verified in memory (or loaded if not present), the dsPIC30F can be programmed using the command set shown in Table 5-1. A detailed description for each command is provided in **Section 8.0 "Programming Executive Commands"**.

TABLE 5-1: COMMAND SET SUMMARY

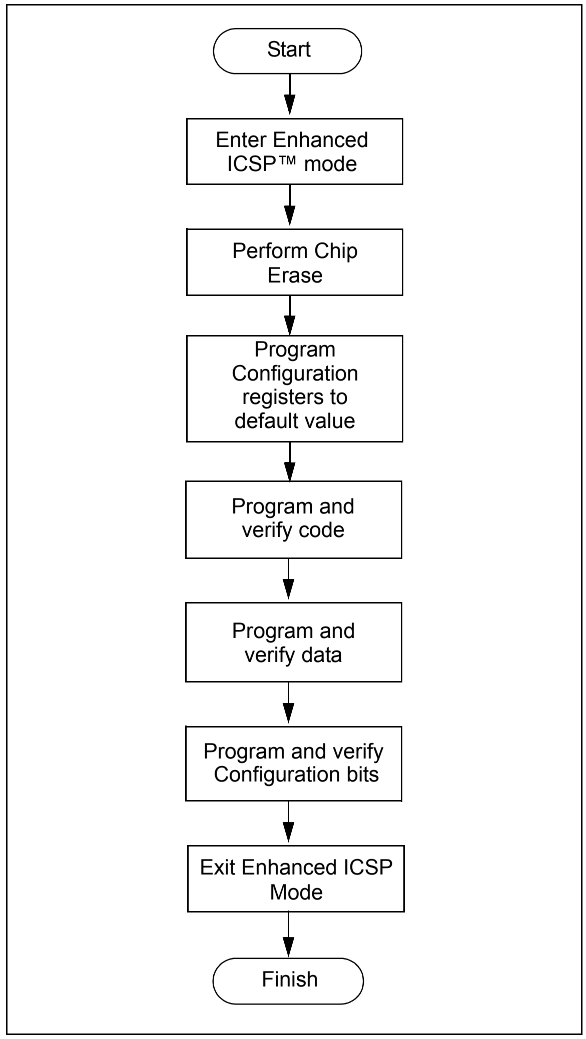
Command	Description
`SCHECK`	Sanity check
`READD`	Read data EEPROM, Configuration registers and device ID
`READP`	Read code memory
`PROGD`	Program one row of data EEPROM and verify
`PROGP`	Program one row of code memory and verify
`PROGC`	Program Configuration bits and verify
`ERASEB`	Bulk Erase, or erase by segment
`ERASED`	Erase data EEPROM
`ERASEP`	Erase code memory
`QBLANK`	Query if the code memory and data EEPROM are blank
`QVER`	Query the software version

A high-level overview of the programming process is illustrated in Figure 5-1. The process begins by entering Enhanced ICSP mode. The chip is then bulk erased, which clears all memory to '`1`' and allows the device to be programmed. The Chip Erase is verified before programming begins. Next, the code memory, data Flash and Configuration bits are programmed. As these memories are programmed, they are each verified to ensure that programming was successful. If no errors are detected, the programming is complete and Enhanced ICSP mode is exited. If any of the verifications fail, the procedure should be repeated, starting from the Chip Erase.

If Advanced Security features are enabled, then individual Segment Erase operations need to be performed, based on user selections (i.e., based on the specific needs of the user application). The specific operations that are used typically depend on the order in which various segments need to be programmed for a given application or system.

Section 5.2 "Entering Enhanced ICSP Mode" through **Section 5.8 "Exiting Enhanced ICSP Mode"** describe the programming process in detail.

FIGURE 5-1: PROGRAMMING FLOW



5.5.3 PROGRAMMING VERIFICATION

Once code memory is programmed, the contents of memory can be verified to ensure that programming was successful. Verification requires code memory to be read back and compared against the copy held in the programmer's buffer.

The READP command can be used to read back all the programmed code memory.

Alternatively, you can have the programmer perform the verification once the entire device is programmed using a checksum computation, as described in **Section 6.8 "Checksum Computation"**.

5.6 Data EEPROM Programming

5.6.1 OVERVIEW

The panel architecture for the data EEPROM memory array consists of 128 rows of sixteen 16-bit data words. Each panel stores 2K words. All devices have either one or no memory panels. Devices with data EEPROM provide either 512 words, 1024 words or 2048 words of memory on the one panel (see Table 5-3).

TABLE 5-3: DATA EEPROM SIZE

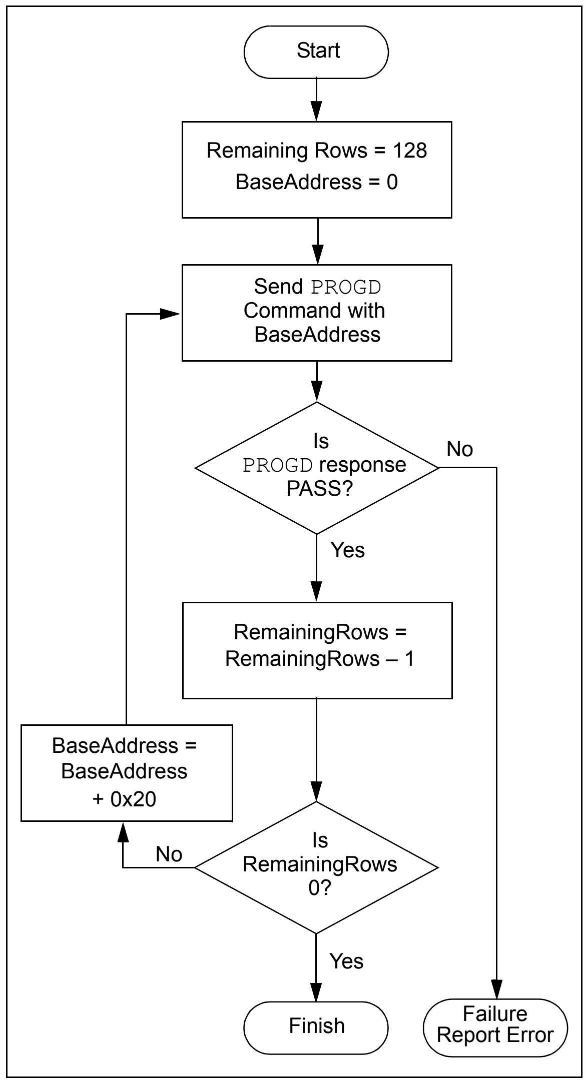
Device	Data EEPROM Size (Words)	Number of Rows
dsPIC30F2010	512	32
dsPIC30F2011	0	0
dsPIC30F2012	0	0
dsPIC30F3010	512	32
dsPIC30F3011	512	32
dsPIC30F3012	512	32
dsPIC30F3013	512	32
dsPIC30F3014	512	32
dsPIC30F4011	512	32
dsPIC30F4012	512	32
dsPIC30F4013	512	32
dsPIC30F5011	512	32
dsPIC30F5013	512	32
dsPIC30F5015	512	32
dsPIC30F5016	512	32
dsPIC30F6010	2048	128
dsPIC30F6010A	2048	128
dsPIC30F6011	1024	64
dsPIC30F6011A	1024	64
dsPIC30F6012	2048	128
dsPIC30F6012A	2048	128
dsPIC30F6013	1024	64
dsPIC30F6013A	1024	64
dsPIC30F6014	2048	128
dsPIC30F6014A	2048	128
dsPIC30F6015	2048	128

5.6.2 PROGRAMMING METHODOLOGY

The programming executive uses the PROGD command to program the data EEPROM. Figure 5-4 illustrates the flowchart of the process. Firstly, the number of rows to program (RemainingRows) is based on the device size, and the destination address (DestAddress) is set to '0'. In this example, 128 rows (2048 words) of data EEPROM will be programmed.

The first PROGD command programs the first row of data EEPROM. Once the command completes successfully, 'RemainingRows' is decremented by 1 and compared with 0. Since there are 127 more rows to program, 'BaseAddress' is incremented by 0x20 to point to the next row of data EEPROM. This process is then repeated until all 128 rows of data EEPROM are programmed.

FIGURE 5-4: FLOWCHART FOR PROGRAMMING dsPIC30F6014A DATA EEPROM



5.6.3 PROGRAMMING VERIFICATION

Once the data EEPROM is programmed, the contents of memory can be verified to ensure that the programming was successful. Verification requires the data EEPROM to be read back and compared against the copy held in the programmer's buffer. The `READD` command reads back the programmed data EEPROM.

Alternatively, the programmer can perform the verification once the entire device is programmed using a checksum computation, as described in **Section 6.8 "Checksum Computation"**.

> **Note:** `TBLRDL` instructions executed within a `REPEAT` loop must not be used to read from Data EEPROM. Instead, it is recommended to use PSV access.

5.7 Configuration Bits Programming

5.7.1 OVERVIEW

The dsPIC30F has Configuration bits stored in seven 16-bit registers. These bits can be set or cleared to select various device configurations. There are two types of Configuration bits: system-operation bits and code-protect bits. The system-operation bits determine the power-on settings for system-level components such as the oscillator and Watchdog Timer. The code-protect bits prevent program memory from being read and written.

The FOSC Configuration register has three different register descriptions, based on the device. The FOSC Configuration register description for the dsPIC30F2010 and dsPIC30F6010/6011/6012/6013/6014 devices are shown in Table 5-4.

> **Note:** If user software performs an erase operation on the configuration fuse, it must be followed by a write operation to this fuse with the desired value, even if the desired value is the same as the state of the erased fuse.

The FOSC Configuration register description for the dsPIC30F4011/4012 and dsPIC30F5011/5013 devices is shown in Table 5-5.

The FOSC Configuration register description for all remaining devices (dsPIC30F2011/2012, dsPIC30F3010/3011/3012/3013, dsPIC30F3014/4013, dsPIC30F5015 and dsPIC30F6011A/6012A/6013A/ 6014A) is shown in Table 5-6. Always use the correct register descriptions for your target processor.

The FWDT, FBORPOR, FBS, FSS, FGS and FICD Configuration registers are not device-dependent. The register descriptions for these Configuration registers are shown in Table 5-7.

The Device Configuration register maps are shown in Table 5-8 through Table 5-11.

TABLE 5-4: FOSC CONFIGURATION BITS DESCRIPTION FOR dsPIC30F2010 AND dsPIC30F6010/6011/6012/6013/6014

Bit Field	Register	Description
FCKSM<1:0>	FOSC	**Clock Switching Mode** `1x` = Clock switching is disabled, Fail-Safe Clock Monitor is disabled `01` = Clock switching is enabled, Fail-Safe Clock Monitor is disabled `00` = Clock switching is enabled, Fail-Safe Clock Monitor is enabled
FOS<1:0>	FOSC	**Oscillator Source Selection on POR** `11` = Primary Oscillator `10` = Internal Low-Power RC Oscillator `01` = Internal Fast RC Oscillator `00` = Low-Power 32 kHz Oscillator (Timer1 Oscillator)
FPR<3:0>	FOSC	**Primary Oscillator Mode** `1111` = ECIO w/PLL 16X – External Clock mode with 16X PLL. OSC2 pin is I/O `1110` = ECIO w/PLL 8X – External Clock mode with 8X PLL. OSC2 pin is I/O `1101` = ECIO w/PLL 4X – External Clock mode with 4X PLL. OSC2 pin is I/O `1100` = ECIO – External Clock mode. OSC2 pin is I/O `1011` = EC – External Clock mode. OSC2 pin is system clock output (FOSC/4) `1010` = Reserved (do not use) `1001` = ERC – External RC Oscillator mode. OSC2 pin is system clock output (FOSC/4) `1000` = ERCIO – External RC Oscillator mode. OSC2 pin is I/O `0111` = XT w/PLL 16X – XT Crystal Oscillator mode with 16X PLL `0110` = XT w/PLL 8X – XT Crystal Oscillator mode with 8X PLL `0101` = XT w/PLL 4X – XT Crystal Oscillator mode with 4X PLL `0100` = XT – XT Crystal Oscillator mode (4 MHz-10 MHz crystal) `001x` = HS – HS Crystal Oscillator mode (10 MHz-25 MHz crystal) `000x` = XTL – XTL Crystal Oscillator mode (200 kHz-4 MHz crystal)

TABLE 5-6: FOSC CONFIGURATION BITS DESCRIPTION FOR dsPIC30F2011/2012, dsPIC30F3010/3011/3012/3013/3014, dsPIC30F4013, dsPIC30F5015/5016, dsPIC30F6010A/6011A/6012A/6013A/6014A AND dsPIC30F6015

Bit Field	Register	Description
FCKSM<1:0>	FOSC	**Clock Switching Mode** 1x = Clock switching is disabled, Fail-Safe Clock Monitor is disabled 01 = Clock switching is enabled, Fail-Safe Clock Monitor is disabled 00 = Clock switching is enabled, Fail-Safe Clock Monitor is enabled
FOS<2:0>	FOSC	**Oscillator Source Selection on POR** 111 = Primary Oscillator 110 = Reserved 101 = Reserved 100 = Reserved 011 = Reserved 010 = Internal Low-Power RC Oscillator 001 = Internal Fast RC Oscillator (no PLL) 000 = Low-Power 32 kHz Oscillator (Timer1 Oscillator)
FPR<4:0>	FOSC	**Primary Oscillator Mode (when FOS<2:0> = 111b)** 11xxx = Reserved (do not use) 10111 = HS/3 w/PLL 16X – HS/3 crystal oscillator with 16X PLL (10 MHz-25 MHz crystal) 10110 = HS/3 w/PLL 8X – HS/3 crystal oscillator with 8X PLL (10 MHz-25 MHz crystal) 10101 = HS/3 w/PLL 4X – HS/3 crystal oscillator with 4X PLL (10 MHz-25 MHz crystal) 10100 = Reserved (do not use) 10011 = HS/2 w/PLL 16X – HS/2 crystal oscillator with 16X PLL (10 MHz-25 MHz crystal) 10010 = HS/2 w/PLL 8X – HS/2 crystal oscillator with 8X PLL (10 MHz-25 MHz crystal 10001 = HS/2 w/PLL 4X – HS/2 crystal oscillator with 4X PLL (10 MHz-25 MHz crystal) 10000 = Reserved (do not use) 01111 = ECIO w/PLL 16x – External clock with 16x PLL. OSC2 pin is I/O 01110 = ECIO w/PLL 8x – External clock with 8x PLL. OSC2 pin is I/O 01101 = ECIO w/PLL 4x – External clock with 4x PLL. OSC2 pin is I/O 01100 = Reserved (do not use) 01011 = Reserved (do not use) 01010 = FRC w/PLL 8x – Internal fast RC oscillator with 8x PLL. OSC2 pin is I/O 01001 = Reserved (do not use) 01000 = Reserved (do not use) 00111 = XT w/PLL 16X – XT crystal oscillator with 16X PLL 00110 = XT w/PLL 8X – XT crystal oscillator with 8X PLL 00101 = XT w/PLL 4X – XT crystal oscillator with 4X PLL 00100 = Reserved (do not use) 00011 = FRC w/PLL 16x – Internal fast RC oscillator with 8x PLL. OSC2 pin is I/O 00010 = Reserved (do not use) 00001 = FRC w/PLL 4x – Internal fast RC oscillator with 4x PLL. OSC2 pin is I/O 00000 = Reserved (do not use)

TABLE 5-8: dsPIC30F CONFIGURATION REGISTERS (FOR dsPIC30F2010, dsPIC30F4011/4012 AND dsPIC30F6010/ 6011/6012/6013/ 6014)

Address	Name	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
0xF80000	FOSC	FCKSM<1:0>		—	—	—	—	FOS<1:0>		—	—	—	—	FPR<3:0>			
0xF80002	FWDT	FWDTEN	—	—	—	—	—	—	—	—	—	FWPSA<1:0>		FWPSB<3:0>			
0xF80004	FBORPOR	MCLREN	—	—	—	—	PWMPIN(1)	HPOL(1)	LPOL(1)	BOREN	—	BORV<1:0>		—	—	FPWRT<1:0>	
0xF80006	FBS	—	—	Reserved(2)		—	—	—	Reserved(2)	—	—	—	—	Reserved(2)			
0xF80008	FSS	—	—	Reserved(2)		—	—	Reserved(2)		—	—	—	—	Reserved(2)			
0xF8000A	FGS	—	—	—	—	—	—	—	—	—	—	—	—	—	Reserved(2)	GCP	GWRP
0xF8000C	FICD	BKBUG	COE	—	—	—	—	—	—	—	—	—	—	—	—	ICS<1:0>	

Note 1: On the 6011, 6012, 6013 and 6014, these bits are reserved (read as '1' and must be programmed as '1').
2: Reserved bits read as '1' and must be programmed as '1'.

TABLE 5-9: dsPIC30F CONFIGURATION REGISTERS (FOR dsPIC30F5011/5013)

Address	Name	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
0xF80000	FOSC	FCKSM<1:0>		—	—	—	—	FOS<1:0>		—	—	—	—	FPR<3:0>			
0xF80002	FWDT	FWDTEN	—	—	—	—	—	—	—	—	—	FWPSA<1:0>		FWPSB<3:0>			
0xF80004	FBORPOR	MCLREN	—	—	—	—	Reserved(1)			BOREN	—	BORV<1:0>		—	—	FPWRT<1:0>	
0xF80006	FBS	—	—	RBS<1:0>		—	—	—	EBS	—	—	—	—	BSS<2:0>			BWRP
0xF80008	FSS	—	—	RSS<1:0>		—	—	ESS<1:0>		—	—	—	—	SSS<2:0>			SWRP
0xF8000A	FGS	—	—	—	—	—	—	—	—	—	—	—	—	—	GSS<1:0>		GWRP
0xF8000C	FICD	BKBUG	COE	—	—	—	—	—	—	—	—	—	—	—	—	ICS<1:0>	

Note 1: Reserved bits read as '1' and must be programmed as '1'.

6.6 Configuration Information in the Hexadecimal File

To allow portability of code, the programmer must read the Configuration register locations from the hexadecimal file. If configuration information is not present in the hexadecimal file, a simple warning message should be issued by the programmer. Similarly, while saving a hexadecimal file, all configuration information must be included. An option to not include the configuration information can be provided.

Microchip Technology Inc. feels strongly that this feature is important for the benefit of the end customer.

6.7 Unit ID

The dsPIC30F devices contain 32 instructions of Unit ID. These are located at addresses 0x8005C0 through 0x8005FF. The Unit ID can be used for storing product information such as serial numbers, system manufacturing dates, manufacturing lot numbers and other such application-specific information.

A Bulk Erase does not erase the Unit ID locations. Instead, erase all executive memory using steps 1-4 as shown in Table 12-1, and program the Unit ID along with the programming executive. Alternately, use a Row Erase to erase the row containing the Unit ID locations.

6.8 Checksum Computation

Checksums for the dsPIC30F are 16 bits in size. The checksum is to total sum of the following:

- Contents of code memory locations
- Contents of Configuration registers

Table A-1 describes how to calculate the checksum for each device. All memory locations are summed one byte at a time, using only their native data size. More specifically, Configuration and device ID registers are summed by adding the lower two bytes of these locations (the upper byte is ignored), while code memory is summed by adding all three bytes of code memory.

Note:	The checksum calculation differs depending on the code-protect setting. Table A-1 describes how to compute the checksum for an unprotected device and a read-protected device. Regardless of the code-protect setting, the Configuration registers can always be read.

7.0 PROGRAMMER – PROGRAMMING EXECUTIVE COMMUNICATION

7.1 Communication Overview

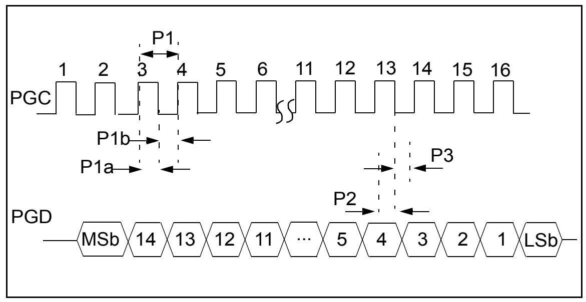
The programmer and programming executive have a master-slave relationship, where the programmer is the master programming device and the programming executive is the slave.

All communication is initiated by the programmer in the form of a command. Only one command at a time can be sent to the programming executive. In turn, the programming executive only sends one response to the programmer after receiving and processing a command. The programming executive command set is described in **Section 8.0 "Programming Executive Commands"**. The response set is described in **Section 9.0 "Programming Executive Responses"**.

7.2 Communication Interface and Protocol

The Enhanced ICSP interface is a 2-wire SPI interface implemented using the PGC and PGD pins. The PGC pin is used as a clock input pin, and the clock source must be provided by the programmer. The PGD pin is used for sending command data to, and receiving response data from, the programming executive. All serial data is transmitted on the falling edge of PGC and latched on the rising edge of PGC. All data transmissions are sent Most Significant bit (MSb) first, using 16-bit mode (see Figure 7-1).

FIGURE 7-1: PROGRAMMING EXECUTIVE SERIAL TIMING



Since a 2-wire SPI interface is used, and data transmissions are bidirectional, a simple protocol is used to control the direction of PGD. When the programmer completes a command transmission, it releases the PGD line and allows the programming executive to drive this line high. The programming executive keeps the PGD line high to indicate that it is processing the command.

After the programming executive has processed the command, it brings PGD low for 15 µsec to indicate to the programmer that the response is available to be

8.5.11 QVER COMMAND

15 12	11 0
Opcode	Length

Field	Description
Opcode	0xB
Length	0x1

The QVER command queries the version of the programming executive software stored in test memory. The "version.revision" information is returned in the response's QE_Code using a single byte with the following format: main version in upper nibble and revision in the lower nibble (i.e., 0x23 is version 2.3 of programming executive software).

Expected Response (2 words):
0x1BMN (where "MN" stands for version M.N)
0x0002

9.0 PROGRAMMING EXECUTIVE RESPONSES

9.1 Overview

The programming executive sends a response to the programmer for each command that it receives. The response indicates if the command was processed correctly, and includes any required response or error data.

The programming executive response set is shown in Table 9-1. This table contains the opcode, mnemonic and description for each response. The response format is described in **Section 9.2 "Response Format"**.

TABLE 9-1: PROGRAMMING EXECUTIVE RESPONSE SET

Opcode	Mnemonic	Description
0x1	PASS	Command successfully processed.
0x2	FAIL	Command unsuccessfully processed.
0x3	NACK	Command not known.

9.2 Response Format

As shown in Example 9-1, all programming executive responses have a general format consisting of a two word header and any required data for the command. Table 9-2 lists the fields and their descriptions.

EXAMPLE 9-1: FORMAT

15 12	11 8	7 0
Opcode	Last_Cmd	QE_Code
Length		
D_1 (if applicable)		
...		
D_N (if applicable)		

TABLE 9-2: FIELDS AND DESCRIPTIONS

Field	Description
Opcode	Response opcode.
Last_Cmd	Programmer command that generated the response.
QE_Code	Query code or Error code.
Length	Response length in 16-bit words (includes 2 header words.)
D_1	First 16-bit data word (if applicable).
D_N	Last 16-bit data word (if applicable).

9.2.1 Opcode FIELD

The Opcode is a 4-bit field in the first word of the response. The Opcode indicates how the command was processed (see Table 9-1). If the command is processed successfully, the response opcode is PASS. If there is an error in processing the command, the response opcode is FAIL, and the QE_Code indicates the reason for the failure. If the command sent to the programming executive is not identified, the programming executive returns a NACK response.

9.2.2 Last_Cmd FIELD

The Last_Cmd is a 4-bit field in the first word of the response and indicates the command that the programming executive processed. Since the programming executive can only process one command at a time, this field is technically not required. However, it can be used to verify whether the programming executive correctly received the command that the programmer transmitted.

9.2.3 QE_Code FIELD

The QE_Code is a byte in the first word of the response. This byte is used to return data for query commands, and error codes for all other commands.

When the programming executive processes one of the two query commands (QBLANK or QVER), the returned opcode is always PASS and the QE_Code holds the query response data. The format of the QE_Code for both queries is shown in Table 9-3.

TABLE 9-3: QE_Code FOR QUERIES

Query	QE_Code
QBLANK	0x0F = Code memory and data EEPROM are NOT blank 0xF0 = Code memory and data EEPROM are blank
QVER	0xMN, where programming executive software version = M.N (i.e., 0x32 means software version 3.2)

When the programming executive processes any command other than a Query, the QE_Code represents an error code. Supported error codes are shown in Table 9-4. If a command is successfully processed, the returned QE_Code is set to 0x0, which indicates that there was no error in the command processing. If the verify of the programming for the PROGD, PROGP or PROGC command fails, the QE_Code is set to 0x1. For all other programming executive errors, the QE_Code is 0x2.

TABLE 9-4: QE_Code FOR NON-QUERY COMMANDS

QE_Code	Description
0x0	No error
0x1	Verify failed
0x2	Other error

9.2.4 RESPONSE LENGTH

The response length indicates the length of the programming executive's response in 16-bit words. This field includes the 2 words of the response header.

With the exception of the response for the READD and READP commands, the length of each response is only 2 words.

The response to the READD command is N + 2 words, where N is the number of words specified in the READD command.

The response to the READP command uses the packed instruction word format described in **Section 8.3 "Packed Data Format"**. When reading an odd number of program memory words (N odd), the response to the READP command is (3 • (N + 1)/2 + 2) words. When reading an even number of program memory words (N even), the response to the READP command is (3 • N/2 + 2) words.

11.4 Flash Memory Programming in ICSP Mode

Programming in ICSP mode is described in **Section 11.4.1 "Programming Operations"** through **Section 11.4.3 "Starting and Stopping a Programming Cycle"**. Step-by-step procedures are described in **Section 11.5 "Erasing Program Memory in Normal-Voltage Systems"** through **Section 11.13 "Reading the Application ID Word"**. All programming operations must use serial execution, as described in **Section 11.2 "ICSP Operation"**.

11.4.1 PROGRAMMING OPERATIONS

Flash memory write and erase operations are controlled by the NVMCON register. Programming is performed by setting NVMCON to select the type of erase operation (Table 11-2) or write operation (Table 11-3), writing a key sequence to enable the programming and initiating the programming by setting the WR control bit, NVMCON<15>.

In ICSP mode, all programming operations are externally timed. An external 2 ms delay must be used between setting the WR control bit and clearing the WR control bit to complete the programming operation.

TABLE 11-2: NVMCON ERASE OPERATIONS

NVMCON Value	Erase Operation
0x407F	Erase all code memory, data memory (does not erase UNIT ID).
0x4075	Erase 1 row (16 words) of data EEPROM.
0x4074	Erase 1 word of data EEPROM.
0x4072	Erase all executive memory.
0x4071	Erase 1 row (32 instruction words) from 1 panel of code memory.
0x406E	Erase Boot Secure and General Segments, then erase FBS, FSS and FGS configuration registers.
0x4066	Erase all Data EEPROM allocated to Boot Segment.
0x405E	Erase Secure and General Segments, then erase FSS and FGS configuration registers.
0x4056	Erase all Data EEPROM allocated to Secure Segment.
0x404E	Erase General Segment, then erase FGS configuration register.
0x4046	Erase all Data EEPROM allocated to General Segment.

TABLE 11-3: NVMCON WRITE OPERATIONS

NVMCON Value	Write Operation
0x4008	Write 1 word to configuration memory.
0x4005	Write 1 row (16 words) to data memory.
0x4004	Write 1 word to data memory.
0x4001	Write 1 row (32 instruction words) into 1 panel of program memory.

11.4.2 UNLOCKING NVMCON FOR PROGRAMMING

Writes to the WR bit (NVMCON<15>) are locked to prevent accidental programming from taking place. Writing a key sequence to the NVMKEY register unlocks the WR bit and allows it to be written to. The unlock sequence is performed as follows:

```
MOV     #0x55, W8
MOV     W8, NVMKEY
MOV     #0xAA, W9
MOV     W9, NVMKEY
```

Note: Any working register, or working register pair, can be used to write the unlock sequence.

11.4.3 STARTING AND STOPPING A PROGRAMMING CYCLE

Once the unlock key sequence has been written to the NVMKEY register, the WR bit (NVMCON<15>) is used to start and stop an erase or write cycle. Setting the WR bit initiates the programming cycle. Clearing the WR bit terminates the programming cycle.

All erase and write cycles must be externally timed. An external delay must be used between setting and clearing the WR bit. Starting and stopping a programming cycle is performed as follows:

```
BSET    NVMCON, #WR
<Wait 2 ms>
BCLR    NVMCON, #WR
```

11.5 Erasing Program Memory in Normal-Voltage Systems

The procedure for erasing program memory (all code memory, data memory, executive memory and code-protect bits) consists of setting NVMCON to 0x407F, unlocking NVMCON for erasing and then executing the programming cycle. This method of bulk erasing program memory only works for systems where VDD is between 4.5 volts and 5.5 volts. The method for erasing program memory for systems with a lower VDD (3.0 volts-4.5 volts) is described in **Section 6.1 "Erasing Memory"**.

TABLE 11-5: SERIAL INSTRUCTION EXECUTION FOR ERASING PROGRAM MEMORY (EITHER IN LOW-VOLTAGE OR NORMAL-VOLTAGE SYSTEMS) (CONTINUED)

Command (Binary)	Data (Hexadecimal)	Description
Step 18: Unlock the NVMCON to erase 1 row of data memory.		
0000	200558	`MOV #0x55, W8`
0000	883B38	`MOV W8, NVMKEY`
0000	200AA9	`MOV #0xAA, W9`
0000	883B39	`MOV W9, NVMKEY`
Step 19: Initiate the erase cycle.		
0000	A8E761	`BSET NVMCON, #WR`
0000	000000	`NOP`
0000	000000	`NOP`
—	—	`Externally time 'P13a' ms` (see **Section 13.0 "AC/DC Characteristics and Timing Requirements"**)
0000	000000	`NOP`
0000	000000	`NOP`
0000	A9E761	`BCLR NVMCON, #WR`
0000	000000	`NOP`
0000	000000	`NOP`
Step 20: Update the row address stored in NVMADR.		
0000	430307	`ADD W6, W7, W6`
0000	883B16	`MOV W6, NVMADR`
Step 21: Reset device internal PC.		
0000	040100	`GOTO 0x100`
0000	000000	`NOP`
Step 22: Repeat Steps 17-21 until all rows of data memory are erased.		

11.8 Writing Code Memory

The procedure for writing code memory is similar to the procedure for clearing the Configuration registers, except that 32 instruction words are programmed at a time. To facilitate this operation, working registers W0:W5 are used as temporary holding registers for the data to be programmed.

Table 11-8 shows the ICSP programming details, including the serial pattern with the ICSP command code, which must be transmitted Least Significant bit first using the PGC and PGD pins (see Figure 11-2). In Step 1, the Reset vector is exited. In Step 2, the NVMCON register is initialized for single-panel programming of code memory. In Step 3, the 24-bit starting destination address for programming is loaded into the TBLPAG register and W7 register. The upper byte of the starting destination address is stored to TBLPAG, while the lower 16 bits of the destination address are stored to W7.

To minimize the programming time, the same packed instruction format that the programming executive uses is utilized (Figure 8-2). In Step 4, four packed instruction words are stored to working registers W0:W5 using the MOV instruction and the read pointer W6 is initialized. The contents of W0:W5 holding the packed instruction word data is shown in Figure 11-4.

In Step 5, eight TBLWT instructions are used to copy the data from W0:W5 to the write latches of code memory. Since code memory is programmed 32 instruction words at a time, Steps 4 and 5 are repeated eight times to load all the write latches (Step 6).

After the write latches are loaded, programming is initiated by writing to the NVMKEY and NVMCON registers in Steps 7 and 8. In Step 9, the internal PC is reset to 0x100. This is a precautionary measure to prevent the PC from incrementing into unimplemented memory when large devices are being programmed. Lastly, in Step 10, Steps 2-9 are repeated until all of code memory is programmed.

FIGURE 11-5: PACKED INSTRUCTION WORDS IN W0:W5

	15 ... 8	7 ... 0
W0	lsw0	
W1	MSB1	MSB0
W2	lsw1	
W3	lsw2	
W4	MSB3	MSB2
W5	lsw3	

TABLE 11-8: SERIAL INSTRUCTION EXECUTION FOR WRITING CODE MEMORY

Command (Binary)	Data (Hexadecimal)	Description
Step 1: Exit the Reset vector.		
0000	040100	GOTO 0x100
0000	040100	GOTO 0x100
0000	000000	NOP
Step 2: Set the NVMCON to program 32 instruction words.		
0000	24001A	MOV #0x4001, W10
0000	883B0A	MOV W10, NVMCON
Step 3: Initialize the write pointer (W7) for TBLWT instruction.		
0000	200xx0	MOV #<DestinationAddress23:16>, W0
0000	880190	MOV W0, TBLPAG
0000	2xxxx7	MOV #<DestinationAddress15:0>, W7
Step 4: Initialize the read pointer (W6) and load W0:W5 with the next 4 instruction words to program.		
0000	2xxxx0	MOV #<LSW0>, W0
0000	2xxxx1	MOV #<MSB1:MSB0>, W1
0000	2xxxx2	MOV #<LSW1>, W2
0000	2xxxx3	MOV #<LSW2>, W3
0000	2xxxx4	MOV #<MSB3:MSB2>, W4
0000	2xxxx5	MOV #<LSW3>, W5

11.9 Writing Data EEPROM

The procedure for writing data EEPROM is very similar to the procedure for writing code memory, except that fewer words are programmed in each operation. When writing data EEPROM, one row of data EEPROM is programmed at a time. Each row consists of sixteen 16-bit data words. Since fewer words are programmed during each operation, only working registers W0:W3 are used as temporary holding registers for the data to be programmed.

Table 11-9 shows the ICSP programming details for writing data EEPROM. Note that a different NVMCON value is required to write to data EEPROM, and that the TBLPAG register is hard-coded to 0x7F (the upper byte address of all locations of data EEPROM).

TABLE 11-9: SERIAL INSTRUCTION EXECUTION FOR WRITING DATA EEPROM

Command (Binary)	Data (Hexadecimal)	Description
Step 1: Exit the Reset vector.		
0000 0000 0000	040100 040100 000000	GOTO 0x100 GOTO 0x100 NOP
Step 2: Set the NVMCON to write 16 data words.		
0000 0000	24005A 883B0A	MOV #0x4005, W10 MOV W10, NVMCON
Step 3: Initialize the write pointer (W7) for TBLWT instruction.		

0000
0000
0000
	2007F0
	880190
	2xxxx7
		MOV #0x7F, W0
		MOV W0, TBLPAG
		MOV #<DestinationAddress15:0>, W7

Step 4: Load W0:W3 with the next 4 data words to program.		

0000
0000
0000
0000
	2xxxx0
	2xxxx1
	2xxxx2
	2xxxx3
		MOV #<WORD0>, W0
		MOV #<WORD1>, W1
		MOV #<WORD2>, W2
		MOV #<WORD3>, W3

Step 5: Set the read pointer (W6) and load the (next set of) write latches.		
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	EB0300 000000 BB1BB6 000000 000000 BB1BB6 000000 000000 BB1BB6 000000 000000 BB1BB6 000000 000000	CLR W6 NOP TBLWTL [W6++], [W7++] NOP NOP TBLWTL [W6++], [W7++] NOP NOP TBLWTL [W6++], [W7++] NOP NOP TBLWTL [W6++], [W7++] NOP NOP
Step 6: Repeat steps 4-5 four times to load the write latches for 16 data words.		

TABLE 11-10: SERIAL INSTRUCTION EXECUTION FOR READING CODE MEMORY (CONTINUED)

Command (Binary)	Data (Hexadecimal)	Description
Step 4: Output W0:W5 using the VISI register and REGOUT command.		
0000	883C20	MOV W0, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C21	MOV W1, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C22	MOV W2, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C23	MOV W3, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C24	MOV W4, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C25	MOV W5, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
Step 5: Reset the device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
Step 6: Repeat steps 3-5 until all desired code memory is read.		

11.13 Reading the Application ID Word

The application ID word is stored at address 0x8005BE in executive code memory. To read this memory location, you must use the SIX control code to move this program memory location to the VISI register. The REGOUT control code must then be used to clock the contents of the VISI register out of the device. The corresponding control and instruction codes that must be serially transmitted to the device to perform this operation are shown in Table 11-13.

Once the programmer has clocked-out the application ID word, it must be inspected. If the application ID has the value 0xBB, the programming executive is resident in memory and the device can be programmed using the mechanism described in **Section 5.0 "Device Programming"**. However, if the application ID has any other value, the programming executive is not resident in memory. It must be loaded to memory before the device can be programmed. The procedure for loading the programming executive to the memory is described in **Section 12.0 "Programming the Programming Executive to Memory"**.

11.14 Exiting ICSP Mode

After confirming that the programming executive is resident in memory, or loading the programming executive, ICSP mode is exited by removing power to the device or bringing $\overline{MCLR}$ to V_{IL}. Programming can then take place by following the procedure outlined in **Section 5.0 "Device Programming"**.

TABLE 11-13: SERIAL INSTRUCTION EXECUTION FOR READING THE APPLICATION ID WORD

Command (Binary)	Data (Hexadecimal)	Description
Step 1: Exit the Reset vector.		
0000	040100	`GOTO 0x100`
0000	040100	`GOTO 0x100`
0000	000000	`NOP`
Step 2: Initialize TBLPAG and the read pointer (W0) for `TBLRD` instruction.		
0000	200800	`MOV #0x80, W0`
0000	880190	`MOV W0, TBLPAG`
0000	205BE0	`MOV #0x5BE, W0`
0000	207841	`MOV VISI, W1`
0000	000000	`NOP`
0000	BA0890	`TBLRDL [W0], [W1]`
0000	000000	`NOP`
0000	000000	`NOP`
Step 3: Output the VISI register using the REGOUT command.		
0001	<VISI>	Clock out contents of the VISI register
0000	000000	`NOP`

TABLE 12-2: READING EXECUTIVE MEMORY (CONTINUED)

Command (Binary)	Data (Hexadecimal)	Description
Step 4: Output W0:W5 using the VISI register and REGOUT command.		
0000	883C20	MOV W0, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
0000	883C21	MOV W1, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
0000	883C22	MOV W2, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
0000	883C23	MOV W3, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
0000	883C24	MOV W4, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
0000	883C25	MOV W5, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
Step 5: Reset the device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
Step 6: Repeat Steps 3-5 until all 736 instruction words of executive memory are read.		

APPENDIX A: DEVICE-SPECIFIC INFORMATION

A.1 Checksum Computation

The checksum computation is described in **Section 6.8 "Checksum Computation"**. Table A-1 shows how this 16-bit computation can be made for each dsPIC30F device. Computations for read code protection are shown both enabled and disabled. The checksum values assume that the Configuration registers are also erased. However, when code protection is enabled, the value of the FGS register is assumed to be 0x5.

A.2 dsPIC30F5011 and dsPIC30F5013

A.2.1 ICSP PROGRAMMING

The dsPIC30F5011 and dsPIC30F5013 processors require that the FBS and FSS registers be programmed with 0x0000 before the device is chip erased. The steps to perform this action are shown in Table 11-4.

A.2.2 ENHANCED ICSP PROGRAMMING

The dsPIC30F5011 and dsPIC30F5013 processors require that the FBS and FSS registers be programmed with 0x0000 using the `PROGC` command before the `ERASEB` command is used to erase the chip.

TABLE A-1: CHECKSUM COMPUTATION

Device	Read Code Protection	Checksum Computation	Erased Value	Value with 0xAAAAAA at 0x0 and Last Code Address
dsPIC30F2010	Disabled	CFGB+SUM(0:001FFF)	0xD406	0xD208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F2011	Disabled	CFGB+SUM(0:001FFF)	0xD406	0xD208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F2012	Disabled	CFGB+SUM(0:001FFF)	0xD406	0xD208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F3010	Disabled	CFGB+SUM(0:003FFF)	0xA406	0xA208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F3011	Disabled	CFGB+SUM(0:003FFF)	0xA406	0xA208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F3012	Disabled	CFGB+SUM(0:003FFF)	0xA406	0xA208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F3013	Disabled	CFGB+SUM(0:003FFF)	0xA406	0xA208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F3014	Disabled	CFGB+SUM(0:003FFF)	0xA406	0xA208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F4011	Disabled	CFGB+SUM(0:007FFF)	0x4406	0x4208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F4012	Disabled	CFGB+SUM(0:007FFF)	0x4406	0x4208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F4013	Disabled	CFGB+SUM(0:007FFF)	0x4406	0x4208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F5011	Disabled	CFGB+SUM(0:00AFFF)	0xFC06	0xFA08
	Enabled	CFGB	0x0404	0x0404
dsPIC30F5013	Disabled	CFGB+SUM(0:00AFFF)	0xFC06	0xFA08
	Enabled	CFGB	0x0404	0x0404
dsPIC30F5015	Disabled	CFGB+SUM(0:00AFFF)	0xFC06	0xFA08
	Enabled	CFGB	0x0404	0x0404

Item Description:

SUM(a:b) = Byte sum of locations a to b inclusive (all 3 bytes of code memory)

CFGB = **Configuration Block (masked)** = Byte sum of ((FOSC&0xC10F) + (FWDT&0x803F) + (FBORPOR&0x87B3) + (FBS&0x310F) + (FSS&0x330F) + (FGS&0x0007) + (FICD&0xC003))

APPENDIX C: REVISION HISTORY

Note:	Revision histories were not recorded for revisions A through H. The previous revision (J), was published in August 2007.

Revision K (November 2010)

This version of the document includes the following updates:

- Added Note three to **Section 5.2 "Entering Enhanced ICSP Mode"**
- Updated the first paragraph of **Section 10.0 "Device ID"**
- Updated Table 10-1: Device IDs
- Removed the VARIANT bit and updated the bit definition for the DEVID register in Table 10-2: dsPIC30F Device ID Registers
- Removed the VARIANT bit and updated the bit field definition and description for the DEVID register in Table 10-3: Device ID Bits Description
- Updated Note 3 in **Section 11.3 "Entering ICSP Mode"**
- Updated Step 11 in Table 11-4: Serial Instruction Execution for BUlk Erasing Program Memory (Only in Normal-voltage Systems)
- Updated Steps 5, 12 and 19 in Table 11-5: Serial Instruction Execution for Erasing Program Memory (Either in Low-voltage or Normal-voltage Systems)
- Updated Steps 5, 6 and 8 in Table 11-7: Serial Instruction Execution for Writing Configuration Registers
- Updated Steps 6 and 8 in Table 11-8: Serial Instruction Execution for Writing Code Memory
- Updated Steps 6 and 8 in Table 11-9: Serial Instruction Execution for Writing Data EEPROM
- Updated Entering ICSP™ Mode (see Figure 11-4)
- Updated Steps 4 and 11 in Table 12-1: Programming the Programming Executive
- Renamed parameters: P12 to P12a and P13 to P13a, and added parameters P12b and P13b in Table 13-1: AC/DC Characteristics

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

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

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

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

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

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

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

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

Taiwan - Hsin Chu
Tel: 886-3-6578-300
Fax: 886-3-6578-370

Taiwan - Kaohsiung
Tel: 886-7-213-7830
Fax: 886-7-330-9305

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

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

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

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

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

08/04/10