

Welcome to [E-XFL.COM](http://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	I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	12
Program Memory Size	24KB (8K x 24)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/dspic30f3012t-20e-so">https://www.e-xfl.com/product-detail/microchip-technology/dspic30f3012t-20e-so</a>

# dsPIC30F Flash Programming Specification

## 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

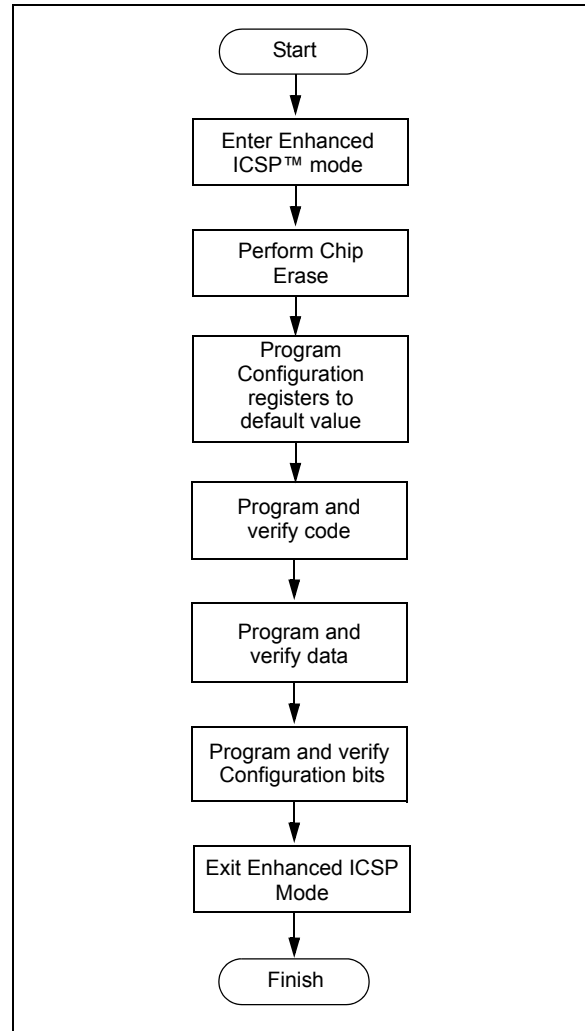
Command	Description
SCHECK	Sanity check
READD	Read data EEPROM, Configuration registers and device ID
READP	Read code memory
PROGD	Program one row of data EEPROM and verify
PROGP	Program one row of code memory and verify
PROGC	Program Configuration bits and verify
ERASEB	Bulk Erase, or erase by segment
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

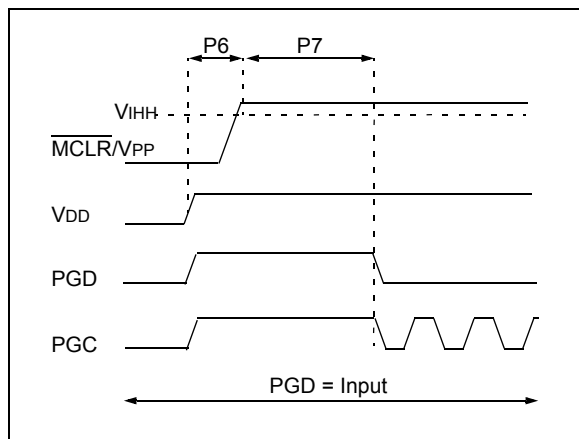


# dsPIC30F Flash Programming Specification

## 5.2 Entering Enhanced ICSP Mode

The Enhanced ICSP mode is entered by holding PGC and PGD high, and then raising MCLR/VPP to VIH (high voltage), as illustrated in Figure 5-2. In this mode, the code memory, data EEPROM and Configuration bits can be efficiently programmed using the programming executive commands that are serially transferred using PGC and PGD.

**FIGURE 5-2: ENTERING ENHANCED ICSP™ MODE**



**Note 1:** The sequence that places the device into Enhanced ICSP mode places all unused I/Os in the high-impedance state.

**2:** Before entering Enhanced ICSP mode, clock switching must be disabled using ICSP, by programming the FCKSM<1:0> bits in the FOSC Configuration register to '11' or '10'.

**3:** When in Enhanced ICSP mode, the SPI output pin (SDO1) will toggle while the device is being programmed.

## 5.3 Chip Erase

Before a chip can be programmed, it must be erased. The Bulk Erase command (ERASEB) is used to perform this task. Executing this command with the MS command field set to 0x3 erases all code memory, data EEPROM and code-protect Configuration bits. The Chip Erase process sets all bits in these three memory regions to '1'.

Since non-code-protect Configuration bits cannot be erased, they must be manually set to '1' using multiple PROGC commands. One PROGC command must be sent for each Configuration register (see Section 5.7 "Configuration Bits Programming").

If Advanced Security features are enabled, then individual Segment Erase operations would need to be performed, depending on which segment needs to be programmed at a given stage of system programming. The user should have the flexibility to select specific segments for programming.

**Note:** The Device ID registers cannot be erased. These registers remain intact after a Chip Erase is performed.

## 5.4 Blank Check

The term "Blank Check" means to verify that the device has been successfully erased and has no programmed memory cells. A blank or erased memory cell reads as '1'. The following memories must be blank checked:

- All implemented code memory
- All implemented data EEPROM
- All Configuration bits (for their default value)

The Device ID registers (0xFF0000:0xFF0002) can be ignored by the Blank Check since this region stores device information that cannot be erased. Additionally, all unimplemented memory space should be ignored from the Blank Check.

The QBLANK command is used for the Blank Check. It determines if the code memory and data EEPROM are erased by testing these memory regions. A 'BLANK' or 'NOT BLANK' response is returned. The READD command is used to read the Configuration registers. If it is determined that the device is not blank, it must be erased (see Section 5.3 "Chip Erase") before attempting to program the chip.

# dsPIC30F Flash Programming Specification

## 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**

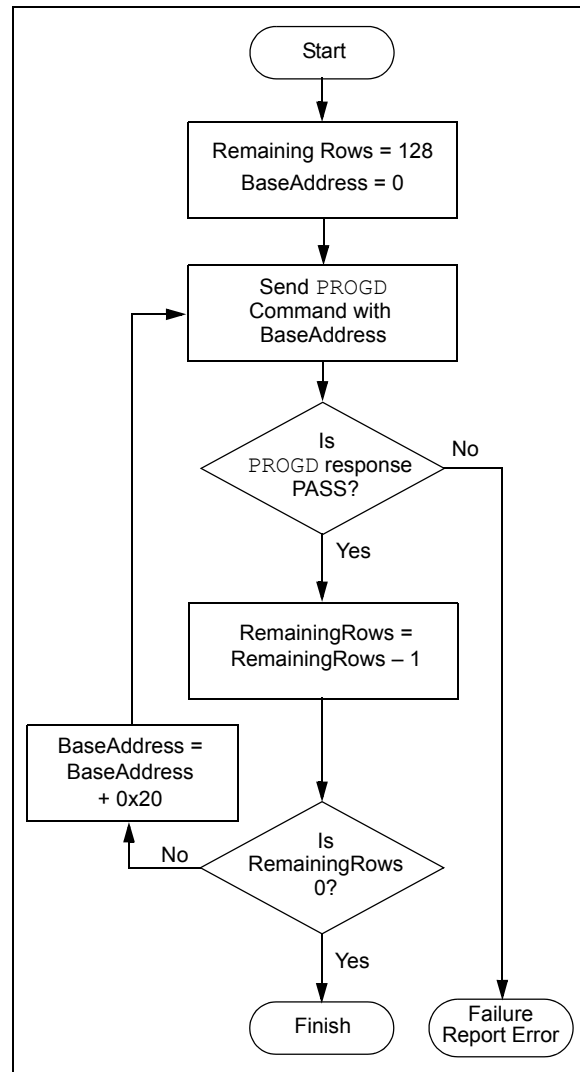
Device	Data EEPROM Size (Words)	Number of Rows
dsPIC30F2010	512	32
dsPIC30F2011	0	0
dsPIC30F2012	0	0
dsPIC30F3010	512	32
dsPIC30F3011	512	32
dsPIC30F3012	512	32
dsPIC30F3013	512	32
dsPIC30F3014	512	32
dsPIC30F4011	512	32
dsPIC30F4012	512	32
dsPIC30F4013	512	32
dsPIC30F5011	512	32
dsPIC30F5013	512	32
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**



# dsPIC30F Flash Programming Specification

**TABLE 5-7: CONFIGURATION BITS DESCRIPTION**

Bit Field	Register	Description
FWPSA<1:0>	FWDT	<b>Watchdog Timer Prescaler A</b> 11 = 1:512 10 = 1:64 01 = 1:8 00 = 1:1
FWPSB<3:0>	FWDT	<b>Watchdog Timer Prescaler B</b> 1111 = 1:16 1110 = 1:15 . . . 0001 = 1:2 0000 = 1:1
FWDTEN	FWDT	<b>Watchdog Enable</b> 1 = Watchdog enabled (LPRC oscillator cannot be disabled. Clearing the SWDTEN bit in the RCON register will have no effect) 0 = Watchdog disabled (LPRC oscillator can be disabled by clearing the SWDTEN bit in the RCON register)
MCLREN	FBORPOR	<b>Master Clear Enable</b> 1 = Master Clear pin (MCLR) is enabled 0 = MCLR pin is disabled
PWMPIN	FBORPOR	<b>Motor Control PWM Module Pin Mode</b> 1 = PWM module pins controlled by PORT register at device Reset (tri-stated) 0 = PWM module pins controlled by PWM module at device Reset (configured as output pins)
HPOL	FBORPOR	<b>Motor Control PWM Module High-Side Polarity</b> 1 = PWM module high-side output pins have active-high output polarity 0 = PWM module high-side output pins have active-low output polarity
LPOL	FBORPOR	<b>Motor Control PWM Module Low-Side Polarity</b> 1 = PWM module low-side output pins have active-high output polarity 0 = PWM module low-side output pins have active-low output polarity
BOREN	FBORPOR	<b>PBOR Enable</b> 1 = PBOR enabled 0 = PBOR disabled
BORV<1:0>	FBORPOR	<b>Brown-out Voltage Select</b> 11 = 2.0V (not a valid operating selection) 10 = 2.7V 01 = 4.2V 00 = 4.5V
FPWRT<1:0>	FBORPOR	<b>Power-on Reset Timer Value Select</b> 11 = PWRT = 64 ms 10 = PWRT = 16 ms 01 = PWRT = 4 ms 00 = Power-up Timer disabled
RBS<1:0>	FBS	<b>Boot Segment Data RAM Code Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)</b> 11 = No Data RAM is reserved for Boot Segment 10 = Small-sized Boot RAM [128 bytes of RAM are reserved for Boot Segment] 01 = Medium-sized Boot RAM [256 bytes of RAM are reserved for Boot Segment] 00 = Large-sized Boot RAM [512 bytes of RAM are reserved for Boot Segment in dsPIC30F5011/5013, and 1024 bytes in dsPIC30F6010A/6011A/6012A/6013A/6014A/6015]

**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 <sup>(1)</sup>	HPOL <sup>(1)</sup>	LPOL <sup>(1)</sup>	BOREN	—	BORV<1:0>		—	—	FPWRT<1:0>	
0xF80006	FBS	—	—	Reserved <sup>(2)</sup>		—	—	—	Reserved <sup>(2)</sup>	—	—	—	—	Reserved <sup>(2)</sup>			
0xF80008	FSS	—	—	Reserved <sup>(2)</sup>		—	—	Reserved <sup>(2)</sup>		—	—	—	—	Reserved <sup>(2)</sup>			
0xF8000A	FGS	—	—	—	—	—	—	—	—	—	—	—	—	—	Reserved <sup>(2)</sup>	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').  
**Note 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 <sup>(1)</sup>			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'.

**TABLE 5-10: dsPIC30F CONFIGURATION REGISTERS (FOR dsPIC30F2011/2012, dsPIC30F3010/3011/3012/3013/3014, dsPIC30F4013 AND dsPIC30F5015/5016)**

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<2:0>			—	—	—	FPR<4:0>				
0xF80002	FWDT	FWDTEN	—	—	—	—	—	—	—	—	—	FWPSA<1:0>		FWPSB<3:0>			
0xF80004	FBORPOR	MCLREN	—	—	—	—	PWMPIN <sup>(1)</sup>	HPOL <sup>(1)</sup>	LPOL <sup>(1)</sup>	BOREN	—	BORV<1:0>		—	—	FPWRT<1:0>	
0xF80006	FBS	—	—	Reserved <sup>(2)</sup>		—	—	—	Reserved <sup>(2)</sup>	—	—	—	—	Reserved <sup>(2)</sup>			
0xF80008	FSS	—	—	Reserved <sup>(2)</sup>		—	—	—	Reserved <sup>(2)</sup>	—	—	—	—	Reserved <sup>(2)</sup>			
0xF8000A	FGS	—	—	—	—	—	—	—	—	—	—	—	—	—	Reserved <sup>(3)</sup>	GCP	GWRP
0xF8000C	FICD	BKBUG	COE	—	—	—	—	—	—	—	—	—	—	—	—	ICS<1:0>	

- Note** 1: On the 2011, 2012, 3012, 3013, 3014 and 4013, 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'.  
3: The FGS<2> bit is a read-only copy of the GCP bit (FGS<1>).

**TABLE 5-11: dsPIC30F CONFIGURATION REGISTERS (FOR dsPIC30F6010A/6011A/6012A/6013A/6014A AND dsPIC30F6015)**

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<2:0>			—	—	—	FPR<4:0>				
0xF80002	FWDT	FWDTEN	—	—	—	—	—	—	—	—	—	FWPSA<1:0>		FWPSB<3:0>			
0xF80004	FBORPOR	MCLREN	—	—	—	—	PWMPIN <sup>(1)</sup>	HPOL <sup>(1)</sup>	LPOL <sup>(1)</sup>	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: On the 6011A, 6012A, 6013A and 6014A, these bits are reserved (read as '1' and must be programmed as '1').

# dsPIC30F Flash Programming Specification

## 5.7.2 PROGRAMMING METHODOLOGY

System operation Configuration bits are inherently different than all other memory cells. Unlike code memory, data EEPROM and code-protect Configuration bits, the system operation bits cannot be erased. If the chip is erased with the `ERASEB` command, the system-operation bits retain their previous value. Consequently, you should make no assumption about the value of the system operation bits. They should always be programmed to their desired setting.

Configuration bits are programmed as a single word at a time using the `PROGC` command. The `PROGC` command specifies the configuration data and Configuration register address. When Configuration bits are programmed, any unimplemented bits must be programmed with a '0', and any reserved bits must be programmed with a '1'.

Four `PROGC` commands are required to program all the Configuration bits. [Figure 5-5](#) illustrates the flowchart of Configuration bit programming.

**Note:** If the General Code Segment Code Protect (GCP) bit is programmed to '0', code memory is code-protected and cannot be read. Code memory must be verified before enabling read protection. See [Section 5.7.4 "Code-Protect Configuration Bits"](#) for more information about code-protect Configuration bits.

## 5.7.3 PROGRAMMING VERIFICATION

Once the Configuration bits are programmed, the contents of memory should be verified to ensure that the programming was successful. Verification requires the Configuration bits to be read back and compared against the copy held in the programmer's buffer. The `READD` command reads back the programmed Configuration bits and verifies whether the programming was successful.

Any unimplemented Configuration bits are read-only and read as '0'.

## 5.7.4 CODE-PROTECT CONFIGURATION BITS

The FBS, FSS and FGS Configuration registers are special Configuration registers that control the size and level of code protection for the Boot Segment, Secure Segment and General Segment, respectively. For each segment, two main forms of code protection are provided. One form prevents code memory from being written (write protection), while the other prevents code memory from being read (read protection).

The BWRP, SWRP and GWRP bits control write protection; and BSS<2:0>, SSS<2:0> and GSS<1:0> bits control read protection. The Chip Erase `ERASEB` command sets all the code protection bits to '1', which allows the device to be programmed.

When write protection is enabled, any programming operation to code memory will fail. When read protection is enabled, any read from code memory will cause a '0x0' to be read, regardless of the actual contents of code memory. Since the programming executive always verifies what it programs, attempting to program code memory with read protection enabled will also result in failure.

It is imperative that all code protection bits are '1' while the device is being programmed and verified. Only after the device is programmed and verified should any of the above bits be programmed to '0' (see [Section 5.7 "Configuration Bits Programming"](#)).

In addition to code memory protection, parts of data EEPROM and/or data RAM can be configured to be accessible only by code resident in the Boot Segment and/or Secure Segment. The sizes of these "reserved" sections are user-configurable, using the EBS, RBS<1:0>, ESS<1:0> and RSS<1:0> bits.

**Note 1:** All bits in the FBS, FSS and FGS Configuration registers can only be programmed to a value of '0'. `ERASEB` is the only way to reprogram code-protect bits from ON ('0') to OFF ('1').

**2:** If any of the code-protect bits in FBS, FSS, or FGS are clear, the entire device must be erased before it can be reprogrammed.



# dsPIC30F Flash Programming Specification

## 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

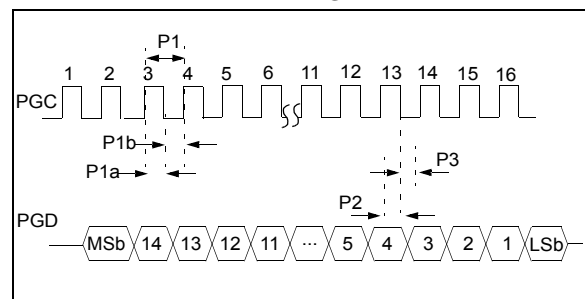
The programmer and programming executive have a master-slave relationship, where the programmer is the master programming device and the programming executive is the slave.

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 PGD. 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  $\mu$ sec to indicate to the programmer that the response is available to be

# dsPIC30F Flash Programming Specification

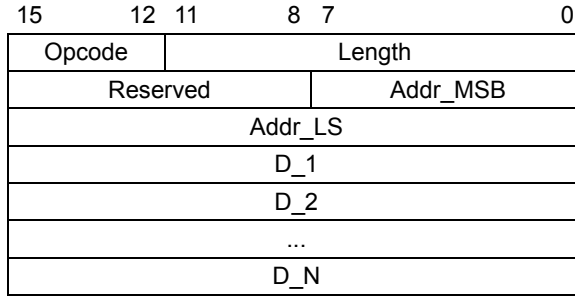
**TABLE 8-1: PROGRAMMING EXECUTIVE COMMAND SET**

Opcode	Mnemonic	Length (16-bit words)	Time Out	Description
0x0	SCHECK	1	1 ms	Sanity check.
0x1	READD	4	1 ms/row	Read N 16-bit words of data EEPROM, Configuration registers or device ID starting from specified address.
0x2	READP	4	1 ms/row	Read N 24-bit instruction words of code memory starting from specified address.
0x3	Reserved	N/A	N/A	This command is reserved. It will return a NACK.
0x4	PROGD <sup>(2)</sup>	19	5 ms	Program one row of data EEPROM at the specified address, then verify.
0x5	PROGP <sup>(1)</sup>	51	5 ms	Program one row of code memory at the specified address, then verify.
0x6	PROGC	4	5 ms	Write byte or 16-bit word to specified Configuration register.
0x7	ERASEB	2	5 ms	Bulk Erase (entire code memory or data EEPROM), or erase by segment.
0x8	ERASED <sup>(2)</sup>	3	5 ms/row	Erase rows of data EEPROM from specified address.
0x9	ERASEP <sup>(1)</sup>	3	5 ms/row	Erase rows of code memory from specified address.
0xA	QBLANK	3	300 ms	Query if the code memory and data EEPROM are blank.
0xB	QVER	1	1 ms	Query the programming executive software version.

- Note 1:** One row of code memory consists of (32) 24-bit words. Refer to [Table 5-2](#) for device-specific information.  
**Note 2:** One row of data EEPROM consists of (16) 16-bit words. Refer to [Table 5-3](#) for device-specific information.

# dsPIC30F Flash Programming Specification

## 8.5.5 PROGP COMMAND



Field	Description
Opcode	0x5
Length	0x33
Reserved	0x0
Addr_MSB	MSB of 24-bit destination address
Addr_LS	LS 16 bits of 24-bit destination address
D_1	16-bit data word 1
D_2	16-bit data word 2
...	16-bit data word 3 through 47
D_48	16-bit data word 48

The `PROGP` command instructs the programming executive to program one row of code memory (32 instruction words) to the specified memory address. Programming begins with the row address specified in the command. The destination address should be a multiple of 0x40.

The data to program to memory, located in command words D\_1 through D\_48, must be arranged using the packed instruction word format shown in [Figure 8-2](#).

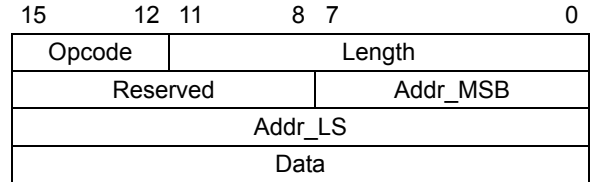
After all data has been programmed to code memory, the programming executive verifies the programmed data against the data in the command.

### Expected Response (2 words):

0x1500  
0x0002

**Note:** Refer to [Table 5-2](#) for code memory size information.

## 8.5.6 PROGC COMMAND



Field	Description
Opcode	0x6
Length	0x4
Reserved	0x0
Addr_MSB	MSB of 24-bit destination address
Addr_LS	LS 16 bits of 24-bit destination address
Data	Data to program

The `PROGC` command programs data to the specified Configuration register and verifies the programming. Configuration registers are 16 bits wide, and this command allows one Configuration register to be programmed.

### Expected Response (2 words):

0x1600  
0x0002

**Note:** This command can only be used for programming Configuration registers.

# dsPIC30F Flash Programming Specification

## 8.5.7 ERASEB COMMAND

15 12 11 2 0

Opcode	Length
Reserved	
MS	

Field	Description
Opcode	0x7
Length	0x2
Reserved	0x0
MS	Select memory to erase: 0x0 = All Code in General Segment 0x1 = All Data EEPROM in General Segment 0x2 = All Code and Data EEPROM in General Segment, interrupt vectors and FGS Configuration register 0x3 = Full Chip Erase 0x4 = All Code and Data EEPROM in Boot, Secure and General Segments, and FBS, FSS and FGS Configuration registers 0x5 = All Code and Data EEPROM in Secure and General Segments, and FSS and FGS Configuration registers 0x6 = All Data EEPROM in Boot Segment 0x7 = All Data EEPROM in Secure Segment

The **ERASEB** command performs a Bulk Erase. The MS field selects the memory to be bulk erased, with options for erasing Code and/or Data EEPROM in individual memory segments.

When Full Chip Erase is selected, the following memory regions are erased:

- All code memory (even if code-protected)
- All data EEPROM
- All code-protect Configuration registers

Only the executive code memory, Unit ID, device ID and Configuration registers that are not code-protected remain intact after a Chip Erase.

### Expected Response (2 words):

0x1700  
0x0002

**Note:** A Full Chip Erase cannot be performed in low-voltage programming systems (VDD less than 4.5 volts). **ERASED** and **ERASEP** must be used to erase code memory, executive memory and data memory. Alternatively, individual Segment Erase operations may be performed.

## 8.5.8 ERASED COMMAND

15 12 11 8 7 0

Opcode	Length
Num_Rows	Addr_MSB
Addr_LS	

Field	Description
Opcode	0x8
Length	0x3
Num_Rows	Number of rows to erase (max of 128)
Addr_MSB	MSB of 24-bit base address
Addr_LS	LS 16 bits of 24-bit base address

The **ERASED** command erases the specified number of rows of data EEPROM from the specified base address. The specified base address must be a multiple of 0x20. Since the data EEPROM is mapped to program space, a 24-bit base address must be specified.

After the erase is performed, all targeted bytes of data EEPROM will contain 0xFF.

### Expected Response (2 words):

0x1800  
0x0002

**Note:** The **ERASED** command cannot be used to erase the Configuration registers or device ID. Code-protect Configuration registers can only be erased with the **ERASEB** command, while the device ID is read-only.

# dsPIC30F Flash Programming Specification

## 11.0 ICSP™ MODE

### 11.1 ICSP Mode

ICSP mode is a special programming protocol that allows you to read and write to the dsPIC30F programming executive. The ICSP mode is the second (and slower) method used to program the device. This mode also has the ability to read the contents of executive memory to determine whether the programming executive is present. This capability is accomplished by applying control codes and instructions serially to the device using pins PGC and PGD.

In ICSP mode, the system clock is taken from the PGC pin, regardless of the device's oscillator Configuration bits. All instructions are first shifted serially into an internal buffer, then loaded into the Instruction register and executed. No program fetching occurs from internal memory. Instructions are fed in 24 bits at a time. PGD is used to shift data in and PGC is used as both the serial shift clock and the CPU execution clock.

Data is transmitted on the rising edge and latched on the falling edge of PGC. For all data transmissions, the Least Significant bit (LSb) is transmitted first.

**Note 1:** During ICSP operation, the operating frequency of PGC must not exceed 5 MHz.

**2:** Because ICSP is slower, it is recommended that only Enhanced ICSP (E-ICSP) mode be used for device programming, as described in [Section 5.1 “Overview of the Programming Process”](#).

### 11.2 ICSP Operation

Upon entry into ICSP mode, the CPU is idle. Execution of the CPU is governed by an internal state machine. A 4-bit control code is clocked in using PGC and PGD, and this control code is used to command the CPU (see [Table 11-1](#)).

The SIX control code is used to send instructions to the CPU for execution, while the REGOUT control code is used to read data out of the device via the VISI register. The operation details of ICSP mode are provided in [Section 11.2.1 “SIX Serial Instruction Execution”](#) and [Section 11.2.2 “REGOUT Serial Instruction Execution”](#).

TABLE 11-1: CPU CONTROL CODES IN ICSP™ MODE

4-bit Control Code	Mnemonic	Description
0000b	SIX	Shift in 24-bit instruction and execute.
0001b	REGOUT	Shift out the VISI register.
0010b-1111b	N/A	Reserved.

#### 11.2.1 SIX SERIAL INSTRUCTION EXECUTION

The SIX control code allows execution of dsPIC30F assembly instructions. When the SIX code is received, the CPU is suspended for 24 clock cycles as the instruction is then clocked into the internal buffer. Once the instruction is shifted in, the state machine allows it to be executed over the next four clock cycles. While the received instruction is executed, the state machine simultaneously shifts in the next 4-bit command (see [Figure 11-2](#)).

**Note 1:** Coming out of the ICSP entry sequence, the first 4-bit control code is always forced to SIX and a forced NOP instruction is executed by the CPU. Five additional PGC clocks are needed on start-up, thereby resulting in a 9-bit SIX command instead of the normal 4-bit SIX command. After the forced SIX is clocked in, ICSP operation resumes as normal (the next 24 clock cycles load the first instruction word to the CPU). See [Figure 11-1](#) for details.

**2:** TBLRDH, TBLRDL, TBLWTH and TBLWTL instructions must be followed by a NOP instruction.

# dsPIC30F Flash Programming Specification

## 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”](#).

# dsPIC30F Flash Programming Specification

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

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 6:</b> Update the row address stored in NVMADRU:NVMADR. When W6 rolls over to 0x0, NVMADRU must be incremented.		
0000	430307	ADD W6, W7, W6
0000	AF0042	BTSC SR, #C
0000	EC2764	INC NVMADRU
0000	883B16	MOV W6, NVMADR
<b>Step 7:</b> Reset device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 8:</b> Repeat Steps 3-7 until all rows of code memory are erased.		
<b>Step 9:</b> Initialize NVMADR and NVMADRU to erase executive memory and initialize W7 for row address updates.		
0000	EB0300	CLR W6
0000	883B16	MOV W6, NVMADR
0000	200807	MOV #0x80, W7
0000	883B27	MOV W7, NVMADRU
0000	200407	MOV #0x40, W7
<b>Step 10:</b> Set NVMCON to erase 1 row of executive memory.		
0000	24071A	MOV #0x4071, W10
0000	883B0A	MOV W10, NVMCON
<b>Step 11:</b> Unlock the NVMCON to erase 1 row of executive memory.		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
<b>Step 12:</b> Initiate the erase cycle.		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P13a' ms (see <a href="#">Section 13.0 "AC/DC Characteristics and Timing Requirements"</a> )
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
<b>Step 13:</b> Update the row address stored in NVMADR.		
0000	430307	ADD W6, W7, W6
0000	883B16	MOV W6, NVMADR
<b>Step 14:</b> Reset device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 15:</b> Repeat Steps 10-14 until all 24 rows of executive memory are erased.		
<b>Step 16:</b> Initialize NVMADR and NVMADRU to erase data memory and initialize W7 for row address updates.		
0000	2XXXX6	MOV #<lower 16-bits of starting Data EEPROM address>, W6
0000	883B16	MOV W6, NVMADR
0000	2007F6	MOV #0x7F, W6
0000	883B16	MOV W6, NVMADRU
0000	200207	MOV #0x20, W7
<b>Step 17:</b> Set NVMCON to erase 1 row of data memory.		
0000	24075A	MOV #0x4075, W10
0000	883B0A	MOV W10, NVMCON

# dsPIC30F Flash Programming Specification

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

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



# dsPIC30F Flash Programming Specification

**TABLE 11-7: SERIAL INSTRUCTION EXECUTION FOR WRITING CONFIGURATION REGISTERS (CONTINUED)**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 6: Write the Configuration register data to the write latch and increment the write pointer.</b>		
0000	BB1B96	TBLWTL W6, [W7++]
0000	000000	NOP
0000	000000	NOP
<b>Step 7: Unlock the NVMCON for programming.</b>		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
<b>Step 8: Initiate the write cycle.</b>		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
–	–	Externally time 'P12a' ms (see <a href="#">Section 13.0 “AC/DC Characteristics and Timing Requirements”</a> )
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
<b>Step 9: Reset device internal PC.</b>		
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 10: Repeat steps 3-9 until all 7 Configuration registers are cleared.</b>		

# dsPIC30F Flash Programming Specification

**TABLE 11-9: SERIAL INSTRUCTION EXECUTION FOR WRITING DATA EEPROM (CONTINUED)**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 7: Unlock the NVMCON for writing.</b>		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
<b>Step 8: Initiate the write cycle.</b>		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P12a' ms (see <a href="#">Section 13.0 “AC/DC Characteristics and Timing Requirements”</a> )
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
<b>Step 9: Reset device internal PC.</b>		
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 10: Repeat steps 2-9 until all data memory is programmed.</b>		

# dsPIC30F Flash Programming Specification

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

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 4: Output W0:W5 using the VISI register and REGOUT command.</b>		
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
<b>Step 5: Reset the device internal PC.</b>		
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 6: Repeat steps 3-5 until all desired code memory is read.</b>		

# dsPIC30F Flash Programming Specification

**TABLE 12-1: PROGRAMMING THE PROGRAMMING EXECUTIVE (CONTINUED)**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 8: Set the read pointer (W6) and load the (next four write) latches.</b>		
0000	EB0300	CLR W6
0000	000000	NOP
0000	BB0BB6	TBLWTL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0000	BBDBB6	TBLWTH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BEBBB6	TBLWTH.B [W6++], [++W7]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BB0BB6	TBLWTL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0000	BBDBB6	TBLWTH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BEBBB6	TBLWTH.B [W6++], [++W7]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
<b>Step 9: Repeat Steps 7-8 eight times to load the write latches for the 32 instructions.</b>		
<b>Step 10: Unlock the NVMCON for programming.</b>		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
<b>Step 11: Initiate the programming cycle.</b>		
0000	A8E761	BSET NVMCON, #15
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P12a' ms (see <a href="#">Section 13.0 “AC/DC Characteristics and Timing Requirements”</a> )
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #15
0000	000000	NOP
0000	000000	NOP
<b>Step 12: Reset the device internal PC.</b>		
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 13: Repeat Steps 7-12 until all 23 rows of executive memory are programmed.</b>		

# dsPIC30F Flash Programming Specification

**TABLE A-1: CHECKSUM COMPUTATION (CONTINUED)**

Device	Read Code Protection	Checksum Computation	Erased Value	Value with 0xAAAAAA at 0x0 and Last Code Address
dsPIC30F5016	Disabled	CFGB+SUM(0:00AFFF)	0xFC06	0xFA08
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6010	Disabled	CFGB+SUM(0:017FFF)	0xC406	0xC208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6010A	Disabled	CFGB+SUM(0:017FFF)	0xC406	0xC208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6011	Disabled	CFGB+SUM(0:015FFF)	0xF406	0xF208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6011A	Disabled	CFGB+SUM(0:015FFF)	0xF406	0xF208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6012	Disabled	CFGB+SUM(0:017FFF)	0xC406	0xC208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6012A	Disabled	CFGB+SUM(0:017FFF)	0xC406	0xC208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6013	Disabled	CFGB+SUM(0:015FFF)	0xF406	0xF208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6013A	Disabled	CFGB+SUM(0:015FFF)	0xF406	0xF208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6014	Disabled	CFGB+SUM(0:017FFF)	0xC406	0xC208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6014A	Disabled	CFGB+SUM(0:017FFF)	0xC406	0xC208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6015	Disabled	CFGB+SUM(0:017FFF)	0xC406	0xC208
	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))