



Welcome to [E-XFL.COM](https://www.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, I ² C, SPI, UART/USART
Peripherals	AC'97, Brown-out Detect/Reset, I ² S, POR, PWM, WDT
Number of I/O	30
Program Memory Size	48KB (16K 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 13x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic30f4013t-20i-pt

dsPIC30F Flash Programming Specification

5.5 Code Memory Programming

5.5.1 OVERVIEW

The Flash code memory array consists of 512 rows of thirty-two, 24-bit instructions. Each panel stores 16K instruction words, and each dsPIC30F device has either 1, 2 or 3 memory panels (see [Table 5-2](#)).

TABLE 5-2: DEVICE CODE MEMORY SIZE

Device	Code Size (24-bit Words)	Number of Rows	Number of Panels
dsPIC30F2010	4K	128	1
dsPIC30F2011	4K	128	1
dsPIC30F2012	4K	128	1
dsPIC30F3010	8K	256	1
dsPIC30F3011	8K	256	1
dsPIC30F3012	8K	256	1
dsPIC30F3013	8K	256	1
dsPIC30F3014	8K	256	1
dsPIC30F4011	16K	512	1
dsPIC30F4012	16K	512	1
dsPIC30F4013	16K	512	1
dsPIC30F5011	22K	704	2
dsPIC30F5013	22K	704	2
dsPIC30F5015	22K	704	2
dsPIC30F5016	22K	704	2
dsPIC30F6010	48K	1536	3
dsPIC30F6010A	48K	1536	3
dsPIC30F6011	44K	1408	3
dsPIC30F6011A	44K	1408	3
dsPIC30F6012	48K	1536	3
dsPIC30F6012A	48K	1536	3
dsPIC30F6013	44K	1408	3
dsPIC30F6013A	44K	1408	3
dsPIC30F6014	48K	1536	3
dsPIC30F6014A	48K	1536	3
dsPIC30F6015	48K	1536	3

5.5.2 PROGRAMMING METHODOLOGY

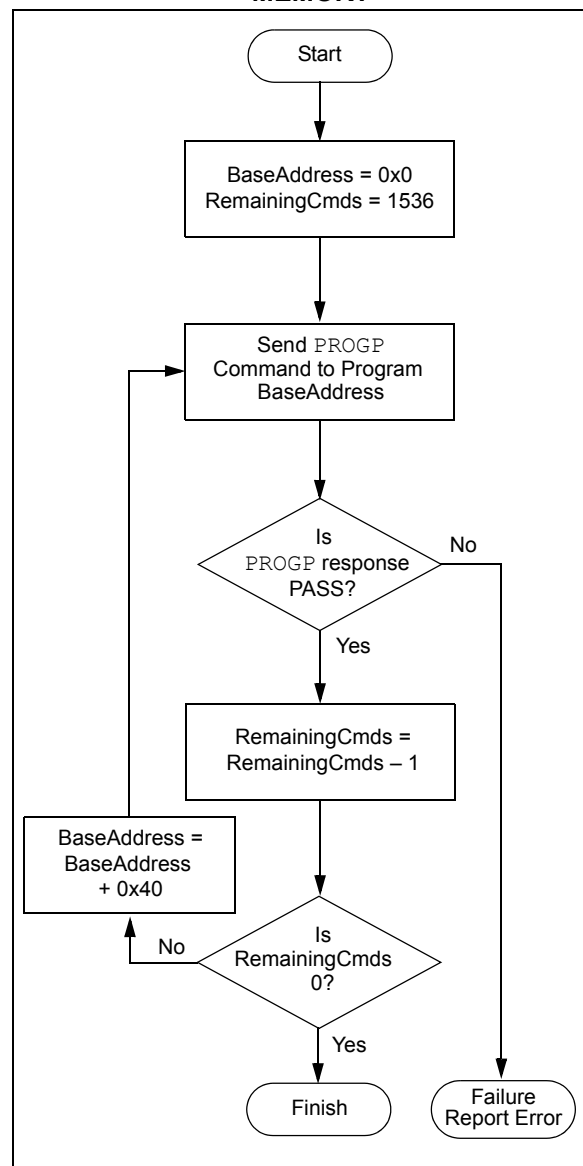
Code memory is programmed with the `PROGP` command. `PROGP` programs one row of code memory to the memory address specified in the command. The number of `PROGP` commands required to program a device depends on the number of rows that must be programmed in the device.

A flowchart for programming of code memory is illustrated in [Figure 5-3](#). In this example, all 48K instruction words of a dsPIC30F6014A device are programmed. First, the number of commands to send (called 'RemainingCmds' in the flowchart) is set to 1536 and the destination address (called 'BaseAddress') is set to '0'.

Next, one row in the device is programmed with a `PROGP` command. Each `PROGP` command contains data for one row of code memory of the dsPIC30F6014A. After the first command is processed successfully, 'RemainingCmds' is decremented by 1 and compared to 0. Since there are more `PROGP` commands to send, 'BaseAddress' is incremented by 0x40 to point to the next row of memory.

On the second `PROGP` command, the second row of each memory panel is programmed. This process is repeated until the entire device is programmed. No special handling must be performed when a panel boundary is crossed.

FIGURE 5-3: FLOWCHART FOR PROGRAMMING dsPIC30F6014A CODE MEMORY



dsPIC30F Flash Programming Specification

TABLE 5-5: FOSC CONFIGURATION BITS DESCRIPTION FOR dsPIC30F4011/4012 AND dsPIC30F5011/5013

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 = FRC w/PLL 8x – Internal fast RC oscillator with 8x PLL. OSC2 pin is I/O 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) 0011 = FRC w/PLL 16x – Internal fast RC oscillator with 16x PLL. OSC2 pin is I/O 0010 = HS – HS Crystal Oscillator mode (10 MHz-25 MHz crystal) 0001 = FRC w/PLL 4x – Internal fast RC oscillator with 4x PLL. OSC2 pin is I/O 0000 = XTL – XTL Crystal Oscillator mode (200 kHz-4 MHz crystal)

dsPIC30F Flash Programming Specification

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)

dsPIC30F Flash Programming Specification

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 (CONTINUED)

Bit Field	Register	Description
FPR<4:0>	FOSC	Alternate Oscillator Mode (when FOS<2:0> = 011b) 1xxxx = Reserved (do not use) 0111x = Reserved (do not use) 01101 = Reserved (do not use) 01100 = ECIO – External clock. OSC2 pin is I/O 01011 = EC – External clock. OSC2 pin is system clock output (Fosc/4) 01010 = Reserved (do not use) 01001 = ERC – External RC oscillator. OSC2 pin is system clock output (Fosc/4) 01000 = ERCIO – External RC oscillator. OSC2 pin is I/O 00111 = Reserved (do not use) 00110 = Reserved (do not use) 00101 = Reserved (do not use) 00100 = XT – XT crystal oscillator (4 MHz-10 MHz crystal) 00010 = HS – HS crystal oscillator (10 MHz-25 MHz crystal) 00001 = Reserved (do not use) 00000 = XTL – XTL crystal oscillator (200 kHz-4 MHz crystal)

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

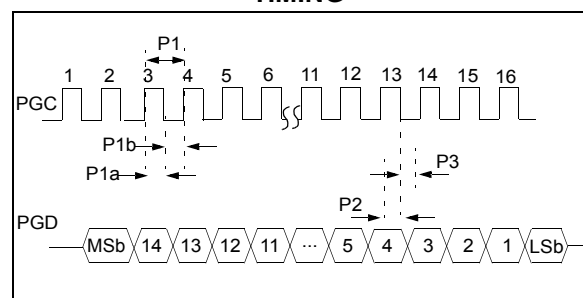
The programmer and programming executive have a master-slave relationship, where the programmer is the master programming device and the programming executive is the slave.

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 μ sec to indicate to the programmer that the response is available to be

dsPIC30F Flash Programming Specification

8.5 Command Descriptions

All commands that are supported by the programming executive are described in [Section 8.5.1 “SCHECK Command”](#) through [Section 8.5.11 “QVER Command”](#).

8.5.1 SCHECK COMMAND

15	12	11	0
Opcode	Length		

Field	Description
Opcode	0x0
Length	0x1

The `SCHECK` command instructs the programming executive to do nothing, but generate a response. This command is used as a “sanity check” to verify that the programming executive is operational.

Expected Response (2 words):

0x1000
0x0002

Note: This instruction is not required for programming, but is provided for development purposes only.

8.5.2 READD COMMAND

15	12	11	8	7	0
Opcode		Length			
Reserved0		N			
Reserved1			Addr_MSB		
Addr_LS					

Field	Description
Opcode	0x1
Length	0x4
Reserved0	0x0
N	Number of 16-bit words to read (max of 2048)
Reserved1	0x0
Addr_MSB	MSB of 24-bit source address
Addr_LS	LS 16 bits of 24-bit source address

The `READD` command instructs the programming executive to read N 16-bit words of memory starting from the 24-bit address specified by `Addr_MSB` and `Addr_LS`. This command can only be used to read 16-bit data. It can be used to read data EEPROM, Configuration registers and the device ID.

Expected Response (2+N words):

0x1100
N + 2
Data word 1
...
Data word N

Note: Reading unimplemented memory will cause the programming executive to reset.

dsPIC30F Flash Programming Specification

8.5.5 PROGP COMMAND

15	12	11	8	7	0
Opcode		Length			
Reserved			Addr_MSB		
Addr_LS					
D_1					
D_2					
...					
D_N					

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

15	12	11	8	7	0
Opcode		Length			
Reserved			Addr_MSB		
Addr_LS					
Data					

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

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 $(3 \cdot (N + 1)/2 + 2)$ words. When reading an even number of program memory words (N even), the response to the `READP` command is $(3 \cdot N/2 + 2)$ words.

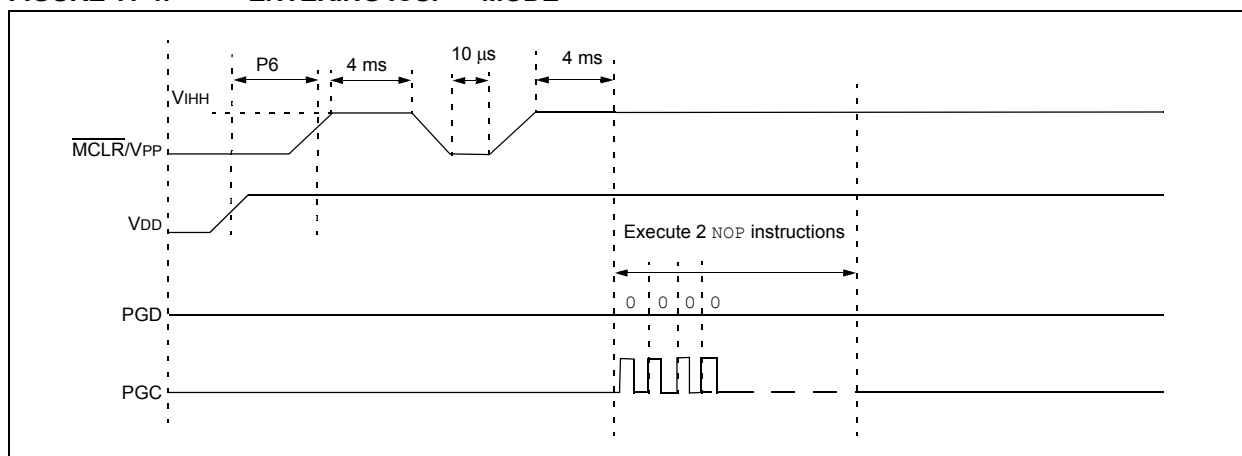
dsPIC30F Flash Programming Specification

11.3 Entering ICSP Mode

The ICSP mode is entered by holding PGC and PGD low, raising MCLR/VPP to VIH (high voltage), and then performing additional steps as illustrated in [Figure 11-4](#).

- Note 1:** The sequence that places the device into ICSP mode places all unused I/O pins to the high-impedance state.
- 2:** Once ICSP mode is entered, the PC is set to 0x0 (the Reset vector).
- 3:** Before leaving the Reset vector, execute two `GOTO` instructions, followed by a single `NOP` instruction must be executed.

FIGURE 11-4: ENTERING ICSP™ MODE



dsPIC30F Flash Programming Specification

Table 11-4 shows the ICSP programming process for bulk-erasing program memory. This process includes the ICSP command code, which must be transmitted (for each instruction) to the Least Significant bit first using the PGC and PGD pins (see Figure 11-2).

If an individual Segment Erase operation is required, the NVMCON value must be replaced by the value for the corresponding Segment Erase operation.

Note: Program memory must be erased before writing any data to program memory.

TABLE 11-4: SERIAL INSTRUCTION EXECUTION FOR BULK ERASING PROGRAM MEMORY (ONLY IN NORMAL-VOLTAGE SYSTEMS)

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 NVMCON to program the FBS Configuration register.⁽¹⁾		
0000	24008A	MOV #0x4008, W10
0000	883B0A	MOV W10, NVMCON
Step 3: Initialize the TBLPAG and write pointer (W7) for TBLWT instruction for Configuration register.⁽¹⁾		
0000	200F80	MOV #0xF8, W0
0000	880190	MOV W0, TBLPAG
0000	200067	MOV #0x6, W7
Step 4: Load the Configuration Register data to W6.⁽¹⁾		
0000	EB0300	CLR W6
0000	000000	NOP
Step 5: Load the Configuration Register write latch. Advance W7 to point to next Configuration register.⁽¹⁾		
0000	BB1B86	TBLWTL W6, [W7++]
Step 6: Unlock the NVMCON for programming the Configuration register.⁽¹⁾		
0000	200558	MOV #0x55, W8
0000	200AA9	MOV #0xAA, W9
0000	883B38	MOV W8, NVMKEY
0000	883B39	MOV W9, NVMKEY
Step 7: Initiate the programming cycle.⁽¹⁾		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 2 ms
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
Step 8: Repeat steps 5-7 one time to program 0x0000 to RESERVED2 Configuration register.⁽¹⁾		
Step 9: Set the NVMCON to erase all Program Memory.		
00000	2407FA	MOV #0x407F, W10
0000	883B0A	MOV W10, NVMCON
Step 10: Unlock the NVMCON for programming.		

Note 1: Steps 2-8 are only required for the dsPIC30F5011/5013 devices. These steps may be skipped for all other devices in the dsPIC30F family.

dsPIC30F Flash Programming Specification

11.7 Writing Configuration Memory

The FOSC, FWDT, FBORPOR and FICD registers are not erasable. It is recommended that all Configuration registers be set to a default value after erasing program memory. The FWDT, FBORPOR and FICD registers can be set to a default all '1's value by programming 0xFFFF to each register. Since these registers contain unimplemented bits that read as '0' the default values shown in [Table 11-6](#) will be read instead of 0xFFFF. The recommended default FOSC value is 0xC100, which selects the FRC clock oscillator setting.

The FGS, FBS and FSS Configuration registers are special since they enable code protection for the device. For security purposes, once any bit in these registers is programmed to '0' (to enable some code protection feature), it can only be set back to '1' by performing a Bulk Erase or Segment Erase as described in [Section 11.5 "Erasing Program Memory in Normal-Voltage Systems"](#). Programming these bits from a '0' to '1' is not possible, but they may be programmed from a '1' to a '0' to enable code protection.

[Table 11-7](#) shows the ICSP programming details for clearing the Configuration registers. In Step 1, the Reset vector is exited. In Step 2, the write pointer (W7) is loaded with 0x0000, which is the original destination address (in TBLPAG 0xF8 of program memory). In Step 3, the NVMCON is set to program one Configura-

tion register. In Step 4, the TBLPAG register is initialized, to 0xF8, for writing to the Configuration registers. In Step 5, the value to write to the each Configuration register (0xFFFF) is loaded to W6. In Step 6, the Configuration register data is written to the write latch using the TBLWTL instruction. In Steps 7 and 8, the NVMCON is unlocked for programming and the programming cycle is initiated, as described in [Section 11.4 "Flash Memory Programming in ICSP Mode"](#). In Step 9, the internal PC is set to 0x100 as a safety measure to prevent the PC from incrementing into unimplemented memory. Lastly, Steps 3-9 are repeated six times until all seven Configuration registers are cleared.

TABLE 11-6: DEFAULT CONFIGURATION REGISTER VALUES

Address	Register	Default Value
0xF80000	FOSC	0xC100
0xF80002	FWDT	0x803F
0xF80004	FBORPOR	0x87B3
0xF80006	FBS	0x310F
0xF80008	FSS	0x330F
0xF8000A	FGS	0x0007
0xF8000C	FICD	0xC003

TABLE 11-7: SERIAL INSTRUCTION EXECUTION FOR WRITING CONFIGURATION REGISTERS

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 the write pointer (W7) for the TBLWT instruction.		
0000	200007	MOV #0x0000, W7
Step 3: Set the NVMCON to program 1 Configuration register.		
0000	24008A	MOV #0x4008, W10
0000	883B0A	MOV W10, NVMCON
Step 4: Initialize the TBLPAG register.		
0000	200F80	MOV #0xF8, W0
0000	880190	MOV W0, TBLPAG
Step 5: Load the Configuration register data to W6.		
0000	2xxxx0	MOV #<CONFIG_VALUE>, W0
0000	000000	NOP

dsPIC30F Flash Programming Specification

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

Command (Binary)	Data (Hexadecimal)	Description
Step 6: Write the Configuration register data to the write latch and increment the write pointer.		
0000	BB1B96	TBLWTL W6, [W7++]
0000	000000	NOP
0000	000000	NOP
Step 7: Unlock the NVMCON for programming.		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
Step 8: Initiate the write cycle.		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P12a' 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 9: Reset device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
Step 10: Repeat steps 3-9 until all 7 Configuration registers are cleared.		

dsPIC30F Flash Programming Specification

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

Command (Binary)	Data (Hexadecimal)	Description
Step 5: Set the read pointer (W6) and load the (next set of) write latches.		
0000	EB0300	CLR W6
0000	000000	NOP
0000	BB0BB6	TBLWTL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0000	BBDDB6	TBLWTH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BBEBB6	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	BBDDB6	TBLWTH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BBEBB6	TBLWTH.B [W6++], [++W7]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
Step 6: Repeat steps 4-5 eight times to load the write latches for 32 instructions.		
Step 7: Unlock the NVMCON for writing.		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
Step 8: Initiate the write cycle.		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P12a' 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 9: Reset device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
Step 10: Repeat steps 2-9 until all code memory is programmed.		

dsPIC30F Flash Programming Specification

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

Command (Binary)	Data (Hexadecimal)	Description
Step 7: Unlock the NVMCON for writing.		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
Step 8: Initiate the write cycle.		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P12a' 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 9: Reset device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
Step 10: Repeat steps 2-9 until all data memory is programmed.		

dsPIC30F Flash Programming Specification

11.12 Reading Data Memory

The procedure for reading data memory is similar to that of reading code memory, except that 16-bit data words are read instead of 24-bit words. Since less data is read in each operation, only working registers W0:W3 are used as temporary holding registers for the data to be read.

Table 11-12 shows the ICSP programming details for reading data memory. Note that the TBLPAG register is hard-coded to 0x7F (the upper byte address of all locations of data memory).

TABLE 11-12: SERIAL INSTRUCTION EXECUTION FOR READING DATA 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: Initialize TBLPAG and the read pointer (W6) for TBLRD instruction.		
0000	2007F0	MOV #0x7F, W0
0000	880190	MOV W0, TBLPAG
0000	2xxxx6	MOV #<SourceAddress15:0>, W6
Step 3: Initialize the write pointer (W7) and store the next four locations of code memory to W0:W5.		
0000	EB0380	CLR W7
0000	000000	NOP
0000	BA1BB6	TBLRDL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BA1BB6	TBLRDL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BA1BB6	TBLRDL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BA1BB6	TBLRDL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
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
Step 5: Reset device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
Step 6: Repeat steps 3-5 until all desired data memory is read.		

dsPIC30F Flash Programming Specification

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 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

dsPIC30F Flash Programming Specification

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

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

dsPIC30F Flash Programming Specification

12.2 Programming Verification

After the programming executive has been programmed to executive memory using ICSP, it must be verified. Verification is performed by reading out the contents of executive memory and comparing it with the image of the programming executive stored in the programmer.

Reading the contents of executive memory can be performed using the same technique described in [Section 11.10 “Reading Code Memory”](#). A procedure for reading executive memory is shown in [Table 12-2](#). Note that in Step 2, the TBLPAG register is set to 0x80 such that executive memory may be read.

TABLE 12-2: READING EXECUTIVE 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: Initialize TBLPAG and the read pointer (W6) for TBLRD instruction.		
0000	200800	MOV #0x80, W0
0000	880190	MOV W0, TBLPAG
0000	EB0300	CLR W6
Step 3: Initialize the write pointer (W7), and store the next four locations of executive memory to W0:W5.		
0000	EB0380	CLR W7
0000	000000	NOP
0000	BA1B96	TBLRDL [W6], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BADBB6	TBLRDH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BADBD6	TBLRDH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BA1BB6	TBLRDL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BA1B96	TBLRDL [W6], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BADBB6	TBLRDH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BADBD6	TBLRDH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BA1BB6	TBLRDL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP

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))

dsPIC30F Flash Programming Specification

NOTES: