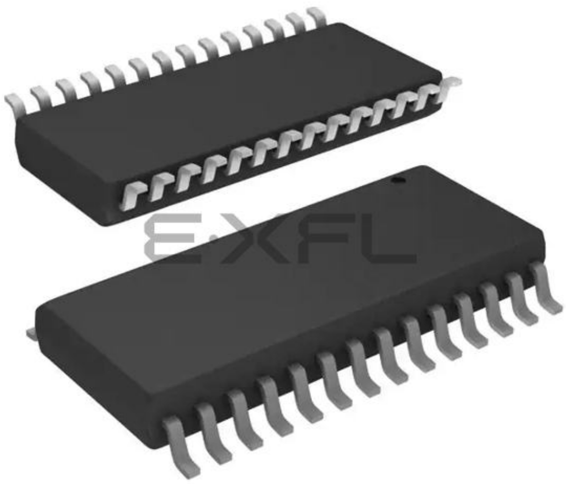

Welcome to E-XFL.COM

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	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	20
Program Memory Size	12KB (4K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 10x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic30f2012t-20i-so

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)

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

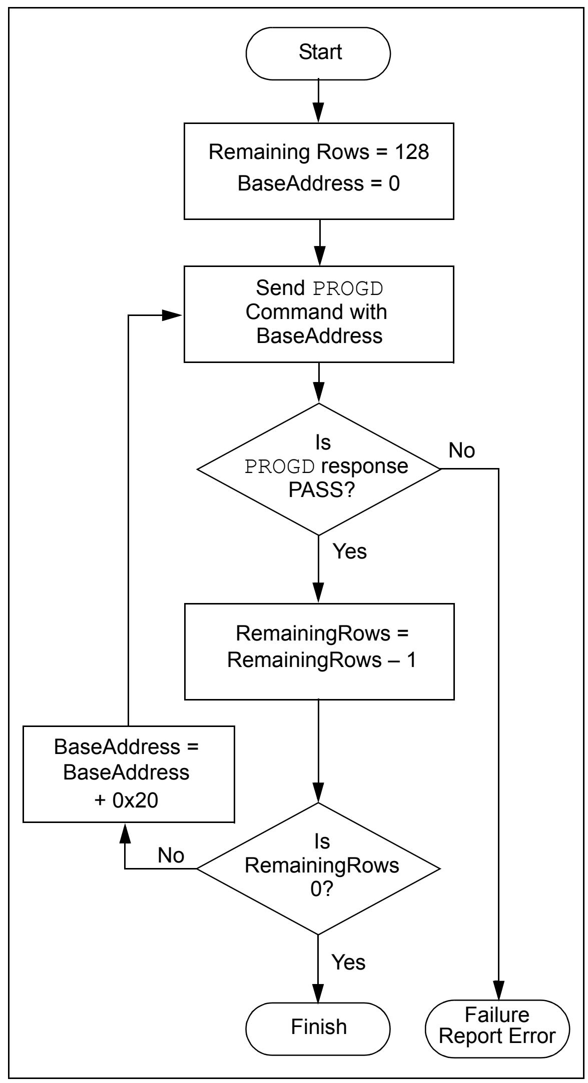

TABLE 5-5: FOSC CONFIGURATION BITS DESCRIPTION FOR dsPIC30F4011/4012 AND dsPIC30F5011/5013

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)

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 5-7: CONFIGURATION BITS DESCRIPTION (CONTINUED)

Bit Field	Register	Description
EBS	FBS	**Boot Segment Data EEPROM Code Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)** 1 = No Data EEPROM is reserved for Boot Segment 0 = 128 bytes of Data EEPROM are reserved for Boot Segment in dsPIC30F5011/5013, and 256 bytes in dsPIC30F6010A/6011A/6012A/6013A/6014A/6015
BSS<2:0>	FBS	**Boot Segment Program Memory Code Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)** 111 = No Boot Segment 110 = Standard security; Small-sized Boot Program Flash [Boot Segment starts after BS and ends at 0x0003FF] 101 = Standard security; Medium-sized Boot Program Flash [Boot Segment starts after BS and ends at 0x000FFF] 100 = Standard security; Large-sized Boot Program Flash [Boot Segment starts after BS and ends at 0x001FF] 011 = No Boot Segment 010 = High security; Small-sized Boot Program Flash [Boot Segment starts after BS and ends at 0x0003FF] 001 = High security; Medium-sized Boot Program Flash [Boot Segment starts after BS and ends at 0x000FFF] 000 = High security; Large-sized Boot Program Flash [Boot Segment starts after BS and ends at 0x001FFF]
BWRP	FBS	**Boot Segment Program Memory Write Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)** 1 = Boot Segment program memory is not write-protected 0 = Boot Segment program memory is write-protected
RSS<1:0>	FSS	**Secure Segment Data RAM Code Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)** 11 = No Data RAM is reserved for Secure Segment 10 = Small-sized Secure RAM [(256 – N) bytes of RAM are reserved for Secure Segment] 01 = Medium-sized Secure RAM [(768 – N) bytes of RAM are reserved for Secure Segment in dsPIC30F5011/5013, and (2048 – N) bytes in dsPIC30F6010A/6011A/6012A/6013A/6014A/6015] 00 = Large-sized Secure RAM [(1024 – N) bytes of RAM are reserved for Secure Segment in dsPIC30F5011/5013, and (4096 – N) bytes in dsPIC30F6010A/6011A/6012A/6013A/6014A/6015] where N = Number of bytes of RAM reserved for Boot Sector.
ESS<1:0>	FSS	**Secure Segment Data EEPROM Code Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)** 11 = No Data EEPROM is reserved for Secure Segment 10 = Small-sized Secure Data EEPROM [(128 – N) bytes of Data EEPROM are reserved for Secure Segment in dsPIC30F5011/5013, and (256 – N) bytes in dsPIC30F6010A/6011A/6012A/6013A/6014A/6015] 01 = Medium-sized Secure Data EEPROM [(256 – N) bytes of Data EEPROM are reserved for Secure Segment in dsPIC30F5011/5013, and (512 – N) bytes in dsPIC30F6010A/6011A/6012A/6013A/6014A/6015] 00 = Large-sized Secure Data EEPROM [(512 – N) bytes of Data EEPROM are reserved for Secure Segment in dsPIC30F5011/5013, (1024 – N) bytes in dsPIC30F6011A/6013A, and (2048 – N) bytes in dsPIC30F6010A/6012A/6014A/6015] where N = Number of bytes of Data EEPROM reserved for Boot Sector.

TABLE 5-7: CONFIGURATION BITS DESCRIPTION (CONTINUED)

Bit Field	Register	Description
SSS<2:0>	FSS	**Secure Segment Program Memory Code Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)** 111 = No Secure Segment 110 = Standard security; Small-sized Secure Program Flash [Secure Segment starts after BS and ends at 0x001FFF] 101 = Standard security; Medium-sized Secure Program Flash [Secure Segment starts after BS and ends at 0x003FFF] 100 = Standard security; Large-sized Secure Program Flash [Secure Segment starts after BS and ends at 0x007FFF] 011 = No Secure Segment 010 = High security; Small-sized Secure Program Flash [Secure Segment starts after BS and ends at 0x001FFF] 001 = High security; Medium-sized Secure Program Flash [Secure Segment starts after BS and ends at 0x003FFF] 000 = High security; Large-sized Secure Program Flash [Secure Segment starts after BS and ends at 0x007FFF]
SWRP	FSS	**Secure Segment Program Memory Write Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)** 1 = Secure Segment program memory is not write-protected 0 = Secure program memory is write-protected
GSS<1:0>	FGS	**General Segment Program Memory Code Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)** 11 = Code protection is disabled 10 = Standard security code protection is enabled 0x = High security code protection is enabled
GCP	FGS	**General Segment Program Memory Code Protection (present in all devices except dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)** 1 = General Segment program memory is not code-protected 0 = General Segment program memory is code-protected
GWRP	FGS	**General Segment Program Memory Write Protection** 1 = General Segment program memory is not write-protected 0 = General Segment program memory is write-protected
BKBUG	FICD	**Debugger/Emulator Enable** 1 = Device will reset into Operational mode 0 = Device will reset into Debug/Emulation mode
COE	FICD	**Debugger/Emulator Enable** 1 = Device will reset into Operational mode 0 = Device will reset into Clip-on Emulation mode
ICS<1:0>	FICD	**ICD Communication Channel Select** 11 = Communicate on PGC/EMUC and PGD/EMUD 10 = Communicate on EMUC1 and EMUD1 01 = Communicate on EMUC2 and EMUD2 00 = Communicate on EMUC3 and EMUD3
RESERVED	FBS, FSS, FGS	**Reserved (read as '1', write as '1')**
—	All	**Unimplemented (read as '0', write as '0')**

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(2)	19	5 ms	Program one row of data EEPROM at the specified address, then verify.
0x5	PROGP(1)	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(2)	3	5 ms/row	Erase rows of data EEPROM from specified address.
0x9	ERASEP(1)	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.
2: One row of data EEPROM consists of (16) 16-bit words. Refer to Table 5-3 for device-specific information.

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.

TABLE 10-3: DEVICE ID BITS DESCRIPTION

Bit Field	Register	Description
DEVID<15:0>	DEVID	Encodes the device ID.
PROC<3:0>	DEVREV	Encodes the process of the device (always read as 0x001).
REV<5:0>	DEVREV	Encodes the major revision number of the device. `000000` = A `000001` = B `000010` = C
DOT<5:0>	DEVREV	Encodes the minor revision number of the device. `000000` = 0 `000001` = 1 `000010` = 2 `000011` = 3
Examples: Rev A.1 = `0000 0000 0000 0001` Rev A.2 = `0000 0000 0000 0010` Rev B.0 = `0000 0000 0100 0000` This formula applies to all dsPIC30F devices, with the exception of the following: • dsPIC30F6010 • dsPIC30F6011 • dsPIC30F6012 • dsPIC30F6013 • dsPIC30F6014 Refer to Table 10-1 for the actual revision IDs.		

11.2.2 REGOUT SERIAL INSTRUCTION EXECUTION

The REGOUT control code allows for data to be extracted from the device in ICSP mode. It is used to clock the contents of the VISI register out of the device over the PGD pin. Once the REGOUT control code is received, eight clock cycles are required to process the command. During this time, the CPU is held idle. After these eight cycles, an additional 16 cycles are required to clock the data out (see Figure 11-3).

The REGOUT instruction is unique because the PGD pin is an input when the control code is transmitted to the device. However, once the control code is processed, the PGD pin becomes an output as the VISI register is shifted out. After the contents of the VISI are shifted out, PGD becomes an input again as the state machine holds the CPU idle until the next 4-bit control code is shifted in.

Note:	Once the contents of VISI are shifted out, the dsPIC® DSC device maintains PGD as an output until the first rising edge of the next clock is received.

FIGURE 11-1: PROGRAM ENTRY AFTER RESET

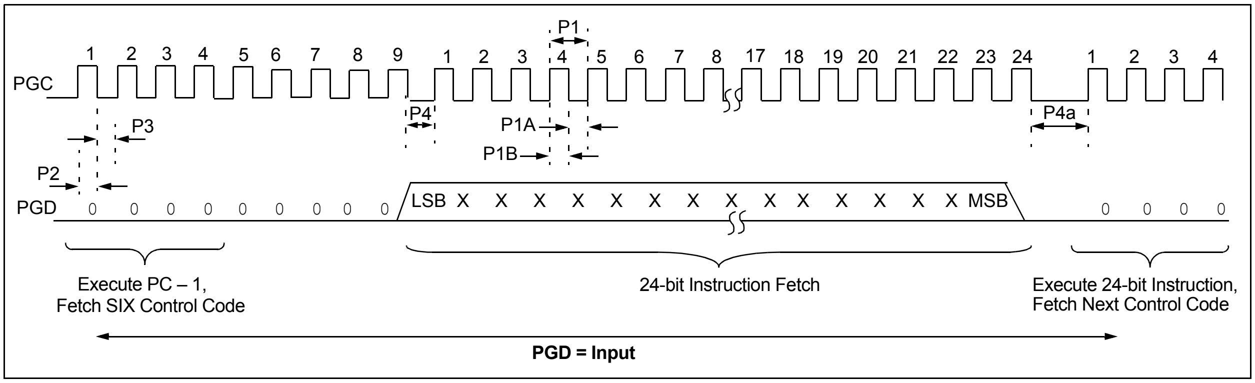

FIGURE 11-2: SIX SERIAL EXECUTION

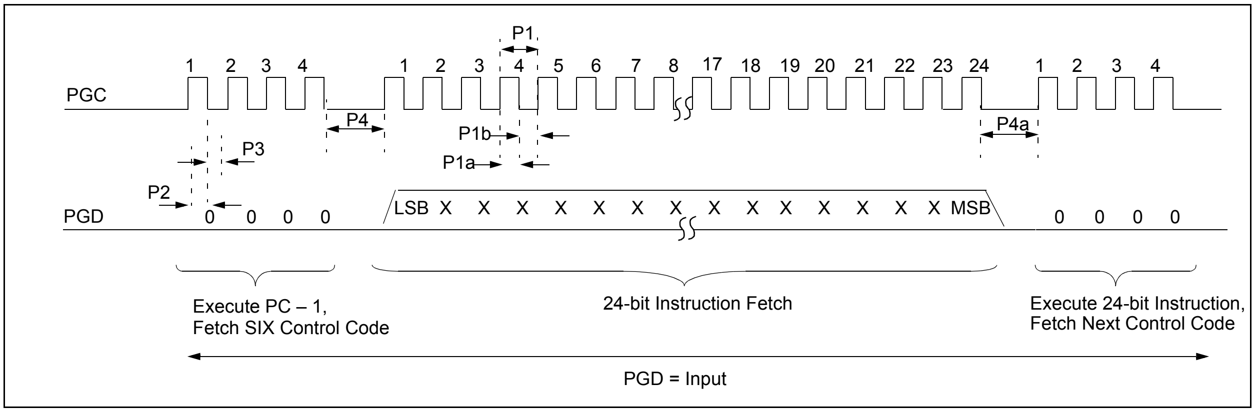

FIGURE 11-3: REGOUT SERIAL EXECUTION

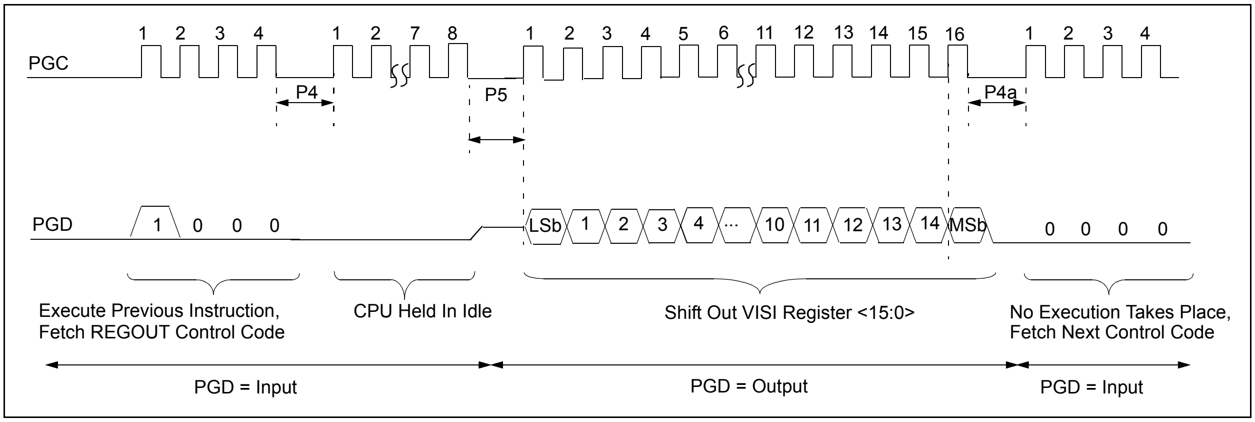

TABLE 11-5: SERIAL INSTRUCTION EXECUTION FOR ERASING PROGRAM MEMORY (EITHER IN LOW-VOLTAGE OR NORMAL-VOLTAGE SYSTEMS) (CONTINUED)

Command (Binary)	Data (Hexadecimal)	Description
Step 6: 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`
Step 7: Reset device internal PC.		
0000	040100	`GOTO 0x100`
0000	000000	`NOP`
Step 8: Repeat Steps 3-7 until all rows of code memory are erased.		
Step 9: 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`
Step 10: Set NVMCON to erase 1 row of executive memory.		
0000	24071A	`MOV #0x4071, W10`
0000	883B0A	`MOV W10, NVMCON`
Step 11: 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`
Step 12: 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 13: Update the row address stored in NVMADR.		
0000	430307	`ADD W6, W7, W6`
0000	883B16	`MOV W6, NVMADR`
Step 14: Reset device internal PC.		
0000	040100	`GOTO 0x100`
0000	000000	`NOP`
Step 15: Repeat Steps 10-14 until all 24 rows of executive memory are erased.		
Step 16: 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`
Step 17: Set NVMCON to erase 1 row of data memory.		
0000	24075A	`MOV #0x4075, W10`
0000	883B0A	`MOV W10, NVMCON`

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
0000
0000
	BB1B96
	000000
	000000
		`TBLWTL W6, [W7++]`
		`NOP`
		`NOP`

Step 7: Unlock the NVMCON for programming.		

0000
0000
0000
0000
	200558
	883B38
	200AA9
	883B39
		`MOV #0x55, W8`
		`MOV W8, NVMKEY`
		`MOV #0xAA, W9`
		`MOV W9, NVMKEY`

Step 8: Initiate the write cycle.		

0000
0000
0000
—
0000
0000
0000
0000
0000
	A8E761
	000000
	000000
	—
	000000
	000000
	A9E761
	000000
	000000
		`BSET NVMCON, #WR`
		`NOP`
		`NOP`
		`Externally time 'P12a' ms` (see **Section 13.0 "AC/DC Characteristics and Timing Requirements"**)
		`NOP`
		`NOP`
		`BCLR NVMCON, #WR`
		`NOP`
		`NOP`

Step 9: Reset device internal PC.		
0000 0000	040100 000000	`GOTO 0x100` `NOP`
Step 10: Repeat steps 3-9 until all 7 Configuration registers are cleared.		

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

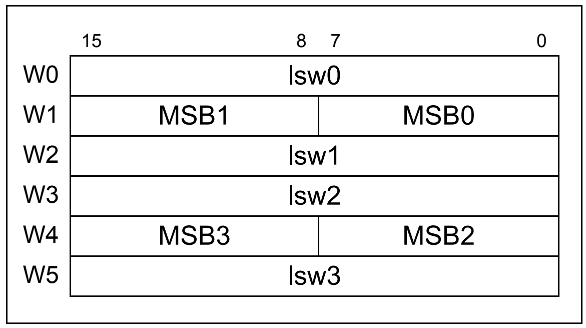

TABLE 11-8: SERIAL INSTRUCTION EXECUTION FOR WRITING CODE MEMORY

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 program 32 instruction words.		
0000 0000	24001A 883B0A	`MOV #0x4001, W10` `MOV W10, NVMCON`
Step 3: Initialize the write pointer (W7) for `TBLWT` instruction.		

0000
0000
0000
	200xx0
	880190
	2xxxx7
		`MOV #<DestinationAddress23:16>, W0`
		`MOV W0, TBLPAG`
		`MOV #<DestinationAddress15:0>, W7`

Step 4: Initialize the read pointer (W6) and load W0:W5 with the next 4 instruction words to program.		

0000
0000
0000
0000
0000
0000
	2xxxx0
	2xxxx1
	2xxxx2
	2xxxx3
	2xxxx4
	2xxxx5
		`MOV #<LSW0>, W0`
		`MOV #<MSB1:MSB0>, W1`
		`MOV #<LSW1>, W2`
		`MOV #<LSW2>, W3`
		`MOV #<MSB3:MSB2>, W4`
		`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.		

11.12 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.		

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.		

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))

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

NOTES: