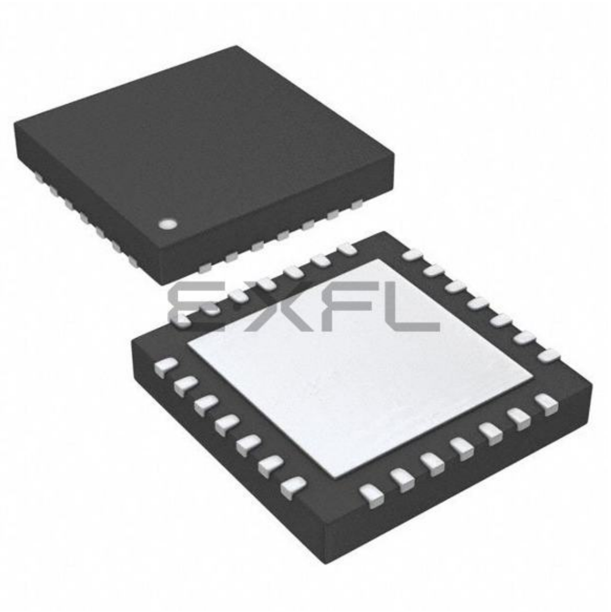
Microchip Technology - DSPIC30F2011T-20I/ML Datasheet



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	12
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 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic30f2011t-20i-ml

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
$\overline{\text{MCLR}}$/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)

5.0 DEVICE PROGRAMMING

5.1 Overview of the Programming Process

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

TABLE 5-1: COMMAND SET SUMMARY

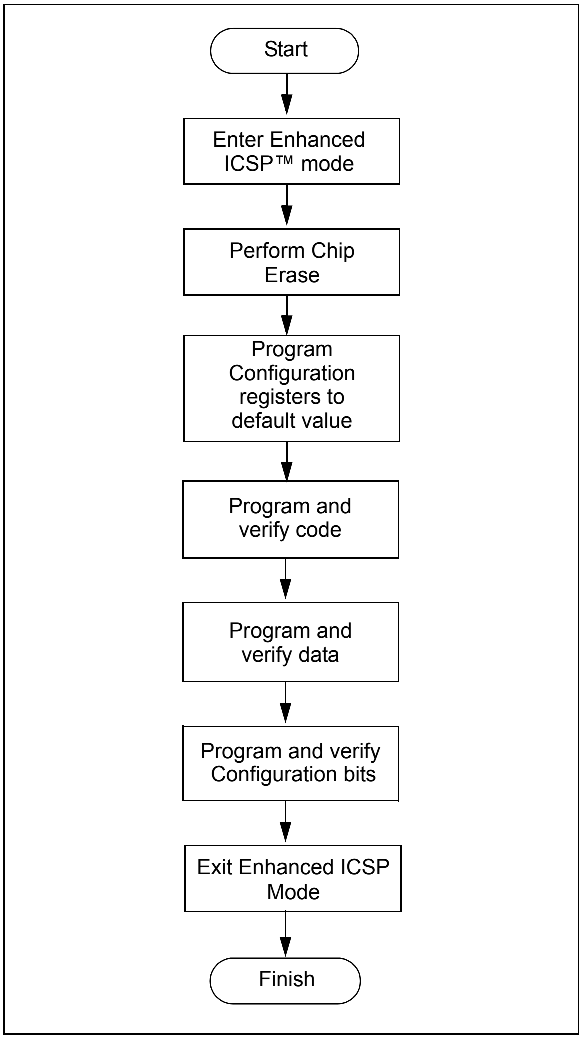
Command	Description
`SCHECK`	Sanity check
`READD`	Read data EEPROM, Configuration registers and device ID
`READP`	Read code memory
`PROGD`	Program one row of data EEPROM and verify
`PROGP`	Program one row of code memory and verify
`PROGC`	Program Configuration bits and verify
`ERASEB`	Bulk Erase, or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

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

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

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

FIGURE 5-1: PROGRAMMING FLOW



5.5.3 PROGRAMMING VERIFICATION

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

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

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

5.6 Data EEPROM Programming

5.6.1 OVERVIEW

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

TABLE 5-3: DATA EEPROM SIZE

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

5.6.2 PROGRAMMING METHODOLOGY

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

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

FIGURE 5-4: FLOWCHART FOR PROGRAMMING dsPIC30F6014A DATA EEPROM

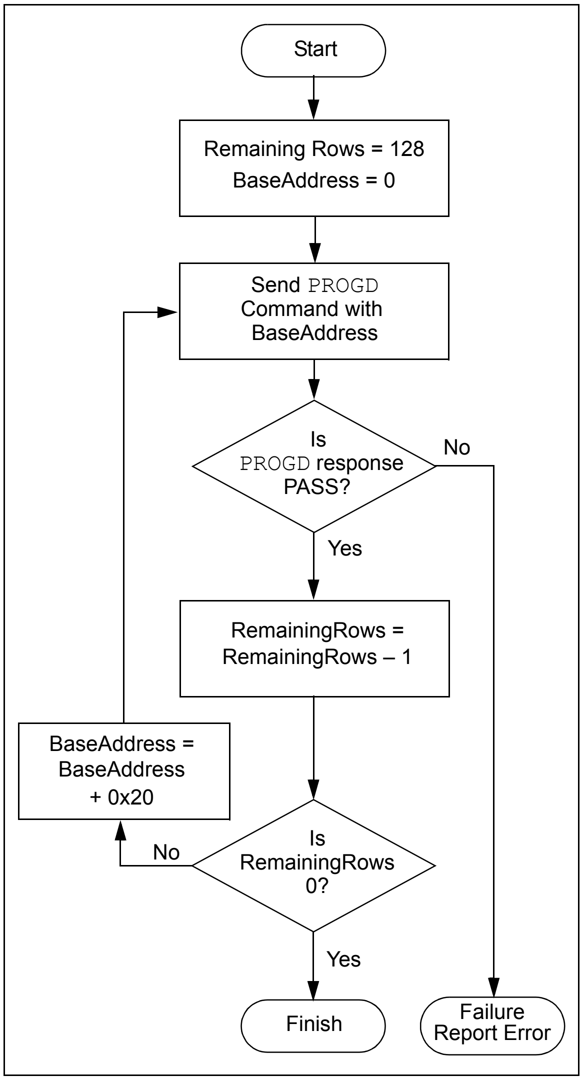


TABLE 5-5: FOSC CONFIGURATION BITS DESCRIPTION FOR dsPIC30F4011/4012 AND dsPIC30F5011/5013

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 5-7: CONFIGURATION BITS DESCRIPTION

Bit Field	Register	Description
FWPSA<1:0>	FWDT	**Watchdog Timer Prescaler A** `11` = 1:512 `10` = 1:64 `01` = 1:8 `00` = 1:1
FWPSB<3:0>	FWDT	**Watchdog Timer Prescaler B** `1111` = 1:16 `1110` = 1:15 • • • `0001` = 1:2 `0000` = 1:1
FWDTEN	FWDT	**Watchdog Enable** `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	**Master Clear Enable** `1` = Master Clear pin ($\overline{\text{MCLR}}$) is enabled `0` = $\overline{\text{MCLR}}$ pin is disabled
PWMPIN	FBORPOR	**Motor Control PWM Module Pin Mode** `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	**Motor Control PWM Module High-Side Polarity** `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	**Motor Control PWM Module Low-Side Polarity** `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	**PBOR Enable** `1` = PBOR enabled `0` = PBOR disabled
BORV<1:0>	FBORPOR	**Brown-out Voltage Select** `11` = 2.0V (not a valid operating selection) `10` = 2.7V `01` = 4.2V `00` = 4.5V
FPWRT<1:0>	FBORPOR	**Power-on Reset Timer Value Select** `11` = PWRT = 64 ms `10` = PWRT = 16 ms `01` = PWRT = 4 ms `00` = Power-up Timer disabled
RBS<1:0>	FBS	**Boot Segment Data RAM Code Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)** `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 5-7: 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 5-8: dsPIC30F CONFIGURATION REGISTERS (FOR dsPIC30F2010, dsPIC30F4011/4012 AND dsPIC30F6010/ 6011/6012/6013/ 6014)

Address	Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xF80000	FOSC	FCKSM<1:0>		—	—	—	—	FOS<1:0>		—	—	—	—	FPR<3:0>			
0xF80002	FWDT	FWDTEN	—	—	—	—	—	—	—	—	—	FWPSA<1:0>		FWPSB<3:0>			
0xF80004	FBORPOR	MCLREN	—	—	—	—	PWMPIN$^{(1)}$	HPOL$^{(1)}$	LPOL$^{(1)}$	BOREN	—	BORV<1:0>		—	—	FPWRT<1:0>	
0xF80006	FBS	—	—	Reserved$^{(2)}$		—	—	—	Reserved$^{(2)}$	—	—	—	—	Reserved$^{(2)}$			
0xF80008	FSS	—	—	Reserved$^{(2)}$		—	—	Reserved$^{(2)}$		—	—	—	—	Reserved$^{(2)}$			
0xF8000A	FGS	—	—	—	—	—	—	—	—	—	—	—	—	—	Reserved$^{(2)}$	GCP	GWRP
0xF8000C	FICD	BKBUG	COE	—	—	—	—	—	—	—	—	—	—	—	—	ICS<1:0>	

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

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

Address	Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xF80000	FOSC	FCKSM<1:0>		—	—	—	—	FOS<1:0>		—	—	—	—	FPR<3:0>			
0xF80002	FWDT	FWDTEN	—	—	—	—	—	—	—	—	—	FWPSA<1:0>		FWPSB<3:0>			
0xF80004	FBORPOR	MCLREN	—	—	—	—	Reserved$^{(1)}$			BOREN	—	BORV<1:0>		—	—	FPWRT<1:0>	
0xF80006	FBS	—	—	RBS<1:0>		—	—	—	EBS	—	—	—	—	BSS<2:0>			BWRP
0xF80008	FSS	—	—	RSS<1:0>		—	—	ESS<1:0>		—	—	—	—	SSS<2:0>			SWRP
0xF8000A	FGS	—	—	—	—	—	—	—	—	—	—	—	—	—	GSS<1:0>		GWRP
0xF8000C	FICD	BKBUG	COE	—	—	—	—	—	—	—	—	—	—	—	—	ICS<1:0>	

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

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.

8.5.3 READP COMMAND

<table>
<tr><th>15 12</th><th>11 8</th><th>7 0</th></tr>
<tr><td>Opcode</td><td colspan="2">Length</td></tr>
<tr><td colspan="3">N</td></tr>
<tr><td colspan="2">Reserved</td><td>Addr_MSB</td></tr>
<tr><td colspan="3">Addr_LS</td></tr>
</table>

Field	Description
Opcode	0x2
Length	0x4
N	Number of 24-bit instructions to read (max of 32768)
Reserved	0x0
Addr_MSB	MSB of 24-bit source address
Addr_LS	LS 16 bits of 24-bit source address

The `READP` command instructs the programming executive to read N 24-bit words of code memory starting from the 24-bit address specified by Addr_MSB and Addr_LS. This command can only be used to read 24-bit data. All data returned in response to this command uses the packed data format described in **Section 8.3 "Packed Data Format"**.

Expected Response (2 + 3 * N/2 words for N even):
0x1200
2 + 3 * N/2
Least significant program memory word 1
...
Least significant data word N

Expected Response (4 + 3 * (N – 1)/2 words for N odd):
0x1200
4 + 3 * (N – 1)/2
Least significant program memory word 1
...
MSB of program memory word N (zero padded)

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

8.5.4 PROGD COMMAND

<table>
<tr><th>15 12</th><th>11 8</th><th>7 0</th></tr>
<tr><td>Opcode</td><td colspan="2">Length</td></tr>
<tr><td colspan="2">Reserved</td><td>Addr_MSB</td></tr>
<tr><td colspan="3">Addr_LS</td></tr>
<tr><td colspan="3">D_1</td></tr>
<tr><td colspan="3">D_2</td></tr>
<tr><td colspan="3">...</td></tr>
<tr><td colspan="3">D_16</td></tr>
</table>

Field	Description
Opcode	0x4
Length	0x13
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 words 3 through 15
D_16	16-bit data word 16

The `PROGD` command instructs the programming executive to program one row of data EEPROM. The data to be programmed is specified by the 16 data words (D_1, D_2,..., D_16) and is programmed to the destination address specified by Addr_MSB and Addr_LSB. The destination address should be a multiple of 0x20.

Once the row of data EEPROM has been programmed, the programming executive verifies the programmed data against the data in the command.

Expected Response (2 words):
0x1400
0x0002

Note: Refer to Table 5-3 for data EEPROM size information.

8.5.11 `QVER` COMMAND

15 12	11 0
Opcode	Length

Field	Description
Opcode	0xB
Length	0x1

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

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

9.0 PROGRAMMING EXECUTIVE RESPONSES

9.1 Overview

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

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

TABLE 9-1: PROGRAMMING EXECUTIVE RESPONSE SET

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

9.2 Response Format

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

EXAMPLE 9-1: FORMAT

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

TABLE 9-2: FIELDS AND DESCRIPTIONS

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

9.2.1 Opcode FIELD

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

9.2.2 Last_Cmd FIELD

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

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

Command (Binary)	Data (Hexadecimal)	Description
Step 18: Unlock the NVMCON to erase 1 row of data memory.		

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

Step 19: Initiate the erase cycle.		

`0000`
`0000`
`0000`
—
`0000`
`0000`
`0000`
`0000`
`0000`
	`A8E761`
	`000000`
	`000000`
	—
	`000000`
	`000000`
	`A9E761`
	`000000`
	`000000`
		`BSET NVMCON, #WR`
		`NOP`
		`NOP`
		`Externally time 'P13a' ms` (see **Section 13.0 "AC/DC Characteristics and Timing Requirements"**)
		`NOP`
		`NOP`
		`BCLR NVMCON, #WR`
		`NOP`
		`NOP`

Step 20: Update the row address stored in NVMADR.		
`0000` `0000`	`430307` `883B16`	`ADD W6, W7, W6` `MOV W6, NVMADR`
Step 21: Reset device internal PC.		
`0000` `0000`	`040100` `000000`	`GOTO 0x100` `NOP`
Step 22: Repeat Steps 17-21 until all rows of data memory are erased.		

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	BBDBB6	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	BBDBB6	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.		

11.13 Reading the Application ID Word

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

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

11.14 Exiting ICSP Mode

After confirming that the programming executive is resident in memory, or loading the programming executive, ICSP mode is exited by removing power to the device or bringing $\overline{\text{MCLR}}$ 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`
`0000`
`0000`
	`040100`
	`040100`
	`000000`
		`GOTO 0x100`
		`GOTO 0x100`
		`NOP`

Step 2: Initialize TBLPAG and the read pointer (W0) for `TBLRD` instruction.		

`0000`
`0000`
`0000`
`0000`
`0000`
`0000`
`0000`
`0000`
	`200800`
	`880190`
	`205BE0`
	`207841`
	`000000`
	`BA0890`
	`000000`
	`000000`
		`MOV #0x80, W0`
		`MOV W0, TBLPAG`
		`MOV #0x5BE, W0`
		`MOV VISI, W1`
		`NOP`
		`TBLRDL [W0], [W1]`
		`NOP`
		`NOP`

Step 3: Output the VISI register using the REGOUT command.		
`0001` `0000`	`<VISI>` `000000`	`Clock out contents of the VISI register` `NOP`

12.0 PROGRAMMING THE PROGRAMMING EXECUTIVE TO MEMORY

12.1 Overview

If it is determined that the programming executive does not reside in executive memory (as described in **Section 4.0 "Confirming the Contents of Executive Memory"**), it must be programmed into executive memory using ICSP and the techniques described in **Section 11.0 "ICSP™ Mode"**.

Storing the programming executive to executive memory is similar to normal programming of code memory. The executive memory must first be erased, and then the programming executive must be programmed 32 words at a time. This control flow is summarized in Table 12-1.

TABLE 12-1: PROGRAMMING THE PROGRAMMING EXECUTIVE

Command (Binary)	Data (Hexadecimal)	Description
Step 1: Exit the Reset vector and erase executive memory.		

0000
0000
0000
	040100
	040100
	000000
		`GOTO 0x100`
		`GOTO 0x100`
		`NOP`

Step 2: Initialize the NVMCON to erase executive memory.		
0000 0000	24072A 883B0A	`MOV #0x4072, W10` `MOV W10, NVMCON`
Step 3: 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 4: Initiate the erase cycle.		

0000
0000
0000
—
0000
0000
0000
0000
0000
	A8E761
	000000
	000000
	—
	000000
	000000
	A9E761
	000000
	000000
		`BSET NVMCON, #15`
		`NOP`
		`NOP`
		`Externally time 'P13a' ms` (see **Section 13.0 "AC/DC Characteristics and Timing Requirements"**)
		`NOP`
		`NOP`
		`BCLR NVMCON, #15`
		`NOP`
		`NOP`

Step 5: Initialize the TBLPAG and the write pointer (W7).		

0000
0000
0000
0000
0000
	200800
	880190
	EB0380
	000000
	000000
		`MOV #0x80, W0`
		`MOV W0, TBLPAG`
		`CLR W7`
		`NOP`
		`NOP`

Step 6: Initialize the NVMCON to program 32 instruction words.		
0000 0000	24001A 883B0A	`MOV #0x4001, W10` `MOV W10, NVMCON`
Step 7: Load W0:W5 with the next 4 words of packed programming executive code and initialize W6 for programming. Programming starts from the base of executive memory (0x800000) using W6 as a read pointer and W7 as a write pointer.		

0000
0000
0000
0000
0000
0000
	2<LSW0>0
	2<MSB1:MSB0>1
	2<LSW1>2
	2<LSW2>3
	2<MSB3:MSB2>4
	2<LSW3>5
		`MOV #<LSW0>, W0`
		`MOV #<MSB1:MSB0>, W1`
		`MOV #<LSW1>, W2`
		`MOV #<LSW2>, W3`
		`MOV #<MSB3:MSB2>, W4`
		`MOV #<LSW3>, W5`

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	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	BBDBB6	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 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.		

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.		

13.0 AC/DC CHARACTERISTICS AND TIMING REQUIREMENTS

TABLE 13-1: AC/DC CHARACTERISTICS

AC/DC CHARACTERISTICS			**Standard Operating Conditions** **(unless otherwise stated)** Operating Temperature: 25° C is recommended			
Param. No.	**Sym**	**Characteristic**	**Min**	**Max**	**Units**	**Conditions**
D110	VIHH	High Programming Voltage on $\overline{\text{MCLR}}$/VPP	9.00	13.25	V	—
D112	IPP	Programming Current on $\overline{\text{MCLR}}$/VPP	—	300	µA	—
D113	IDDP	Supply Current during programming	—	30	mA	Row Erase Program memory
			—	30	mA	Row Erase Data EEPROM
			—	30	mA	Bulk Erase
D001	VDD	Supply voltage	2.5	5.5	V	—
D002	VDDBULK	Supply voltage for Bulk Erase programming	4.5	5.5	V	—
D031	VIL	Input Low Voltage	VSS	0.2 VSS	V	—
D041	VIH	Input High Voltage	0.8 VDD	VDD	V	—
D080	VOL	Output Low Voltage	—	0.6	V	IOL = 8.5 mA
D090	VOH	Output High Voltage	VDD - 0.7	—	V	IOH = -3.0 mA
D012	CIO	Capacitive Loading on I/O Pin (PGD)	—	50	pF	To meet AC specifications
P1	TSCLK	Serial Clock (PGC) period	50	—	ns	ICSP™ mode
			1	—	µs	Enhanced ICSP mode
P1a	TSCLKL	Serial Clock (PGC) low time	20	—	ns	ICSP mode
			400	—	ns	Enhanced ICSP mode
P1b	TSCLKH	Serial Clock (PGC) high time	20	—	ns	ICSP mode
			400	—	ns	Enhanced ICSP mode
P2	TSET1	Input Data Setup Timer to PGC ↓	15	—	ns	—
P3	THLD1	Input Data Hold Time from PGC ↓	15	—	ns	—
P4	TDLY1	Delay between 4-bit command and command operand	20	—	ns	—
P4a	TDLY1a	Delay between 4-bit command operand and next 4-bit command	20	—	ns	—
P5	TDLY2	Delay between last PGC ↓of command to first PGC ↑ of VISI output	20	—	ns	—
P6	TSET2	VDD ↑ setup time to $\overline{\text{MCLR}}$/VPP	100	—	ns	—
P7	THLD2	Input data hold time from $\overline{\text{MCLR}}$/VPP ↑	2	—	µs	ICSP mode
			5	—	ms	Enhanced ICSP mode
P8	TDLY3	Delay between last PGC ↓of command word to PGD driven ↑ by programming executive	20	—	µs	—
P9a	TDLY4	Programming Executive Command processing time	10	—	µs	—

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