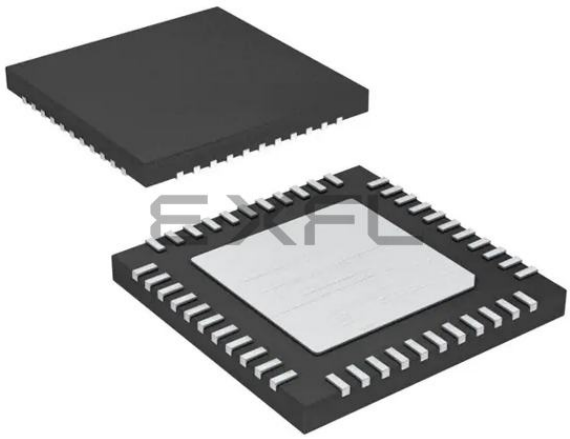
Microchip Technology - DSPIC30F4012T-20E/ML Datasheet



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	CANbus, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, Motor Control PWM, QEI, POR, PWM, WDT
Number of I/O	20
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 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic30f4012t-20e-ml

5.5.3 PROGRAMMING VERIFICATION

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

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

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

5.6 Data EEPROM Programming

5.6.1 OVERVIEW

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

TABLE 5-3: DATA EEPROM SIZE

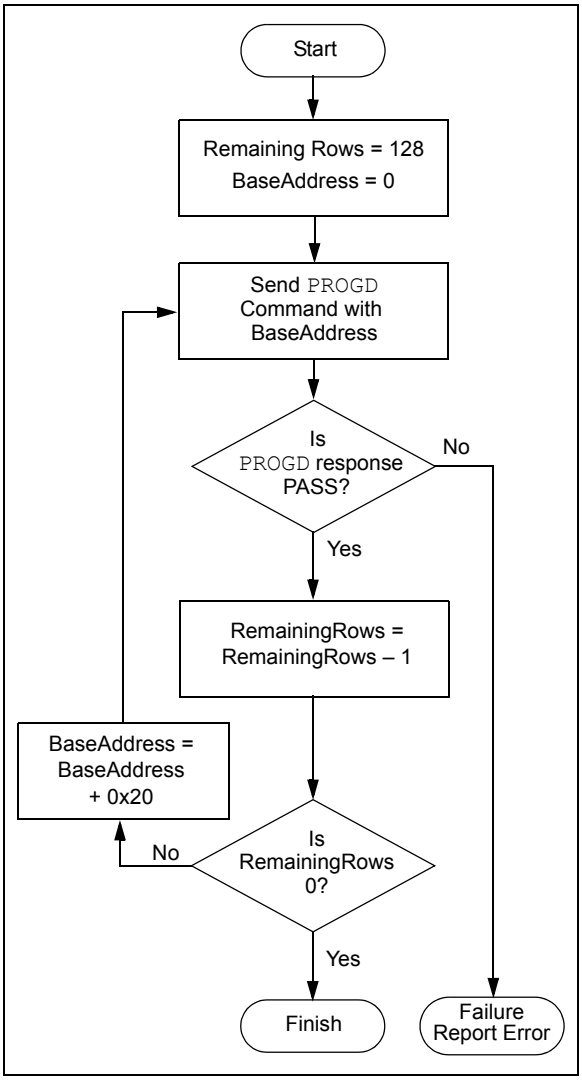
Device	Data EEPROM Size (Words)	Number of Rows
dsPIC30F2010	512	32
dsPIC30F2011	0	0
dsPIC30F2012	0	0
dsPIC30F3010	512	32
dsPIC30F3011	512	32
dsPIC30F3012	512	32
dsPIC30F3013	512	32
dsPIC30F3014	512	32
dsPIC30F4011	512	32
dsPIC30F4012	512	32
dsPIC30F4013	512	32
dsPIC30F5011	512	32
dsPIC30F5013	512	32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

5.6.2 PROGRAMMING METHODOLOGY

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

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

FIGURE 5-4: FLOWCHART FOR PROGRAMMING dsPIC30F6014A DATA EEPROM



5.6.3 PROGRAMMING VERIFICATION

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

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

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

5.7 Configuration Bits Programming

5.7.1 OVERVIEW

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

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

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

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

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

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

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

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

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

TABLE 5-5: 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** $\overline{\text{MCLR}}$ 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]

clocked out. The programmer can begin to clock out the response 20 µsec after PGD is brought low, and it must provide the necessary amount of clock pulses to receive the entire response from the programming executive.

Once the entire response is clocked out, the programmer should terminate the clock on PGC until it is time to send another command to the programming executive. This protocol is illustrated in Figure 7-2.

7.3 SPI Rate

In Enhanced ICSP mode, the dsPIC30F operates from the fast internal RC oscillator, which has a nominal frequency of 7.37 MHz. This oscillator frequency yields an effective system clock frequency of 1.84 MHz. Since the SPI module operates in Slave mode, the programmer must limit the SPI clock rate to a frequency no greater than 1 MHz.

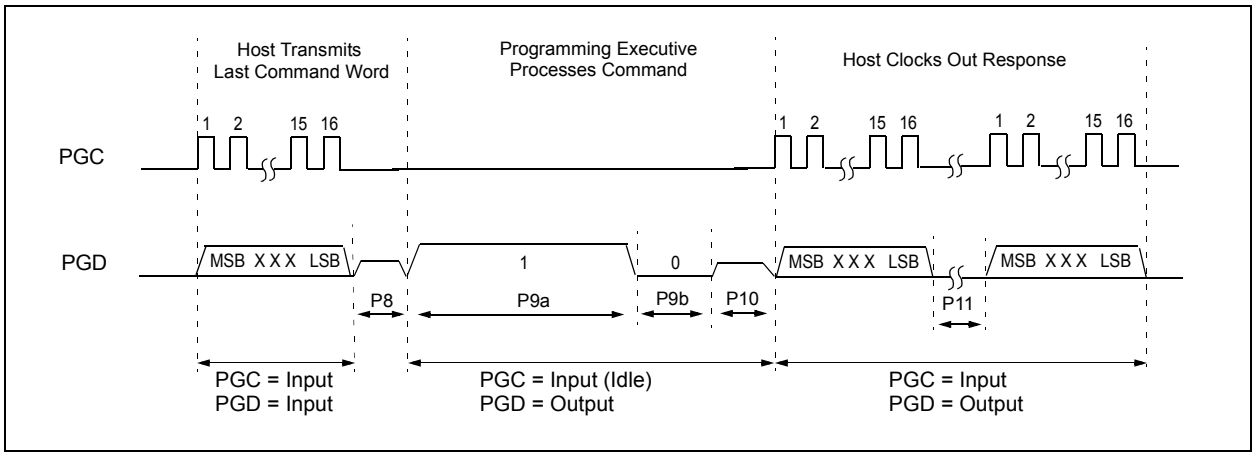
Note:	If the programmer provides the SPI with a clock faster than 1 MHz, the behavior of the programming executive will be unpredictable.

7.4 Time Outs

The programming executive uses no Watchdog Timer or time out for transmitting responses to the programmer. If the programmer does not follow the flow control mechanism using PGC, as described in **Section 7.2 "Communication Interface and Protocol"**, it is possible that the programming executive will behave unexpectedly while trying to send a response to the programmer. Since the programming executive has no time out, it is imperative that the programmer correctly follow the described communication protocol.

As a safety measure, the programmer should use the command time outs identified in Table 8-1. If the command time out expires, the programmer should reset the programming executive and start programming the device again.

FIGURE 7-2: PROGRAMMING EXECUTIVE – PROGRAMMER COMMUNICATION PROTOCOL



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.

10.0 DEVICE ID

The device ID region is 2 x 16 bits and can be read using the `READD` command. This region of memory is read-only and can also be read when code protection is enabled.

Table 10-1 shows the device ID for each device, Table 10-2 shows the device ID registers and Table 10-3 describes the bit field of each register.

TABLE 10-1: DEVICE IDS

Device	DEVID	Silicon Revision							
		A0	A1	A2	A3	A4	B0	B1	B2
dsPIC30F2010	0x0040	0x1000	0x1001	0x1002	0x1003	0x1004	—	—	—
dsPIC30F2011	0x0240	—	0x1001	—	—	—	—	—	—
dsPIC30F2012	0x0241	—	0x1001	—	—	—	—	—	—
dsPIC30F3010	0x01C0	0x1000	0x1001	0x1002	—	—	—	—	—
dsPIC30F3011	0x01C1	0x1000	0x1001	0x1002	—	—	—	—	—
dsPIC30F3012	0x00C1	—	—	—	—	—	0x1040	0x1041	—
dsPIC30F3013	0x00C3	—	—	—	—	—	0x1040	0x1041	—
dsPIC30F3014	0x0160	—	0x1001	0x1002	—	—	—	—	—
dsPIC30F4011	0x0101	—	0x1001	0x1002	0x1003	0x1003	—	—	—
dsPIC30F4012	0x0100	—	0x1001	0x1002	0x1003	0x1003	—	—	—
dsPIC30F4013	0x0141	—	0x1001	0x1002	—	—	—	—	—
dsPIC30F5011	0x0080	—	0x1001	0x1002	0x1003	0x1003	—	—	—
dsPIC30F5013	0x0081	—	0x1001	0x1002	0x1003	0x1003	—	—	—
dsPIC30F5015	0x0200	0x1000	—	—	—	—	—	—	—
dsPIC30F5016	0x0201	0x1000	—	—	—	—	—	—	—
dsPIC30F6010	0x0188	—	—	—	—	—	—	0x1040	0x1042
dsPIC30F6010A	0x0281	—	—	0x1002	0x1003	0x1004	—	—	—
dsPIC30F6011	0x0192	—	—	—	0x1003	—	—	0x1040	0x1042
dsPIC30F6011A	0x02C0	—	—	0x1002	—	—	0x1040	0x1041	—
dsPIC30F6012	0x0193	—	—	—	0x1003	—	—	0x1040	0x1042
dsPIC30F6012A	0x02C2	—	—	0x1002	—	—	0x1040	0x1041	—
dsPIC30F6013	0x0197	—	—	—	0x1003	—	—	0x1040	0x1042
dsPIC30F6013A	0x02C1	—	—	0x1002	—	—	0x1040	0x1041	—
dsPIC30F6014	0x0198	—	—	—	0x1003	—	—	0x1040	0x1042
dsPIC30F6014A	0x02C3	—	—	0x1002	—	—	0x1040	0x1041	—
dsPIC30F6015	0x0280	—	—	0x1002	0x1003	0x1004	—	—	—

TABLE 10-2: dsPIC30F DEVICE ID REGISTERS

Address	Name	Bit															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFF0000	DEVID	DEVID<15:0>															
0xFF0002	DEVREV	PROC<3:0>				REV<5:0>						DOT<5:0>					

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

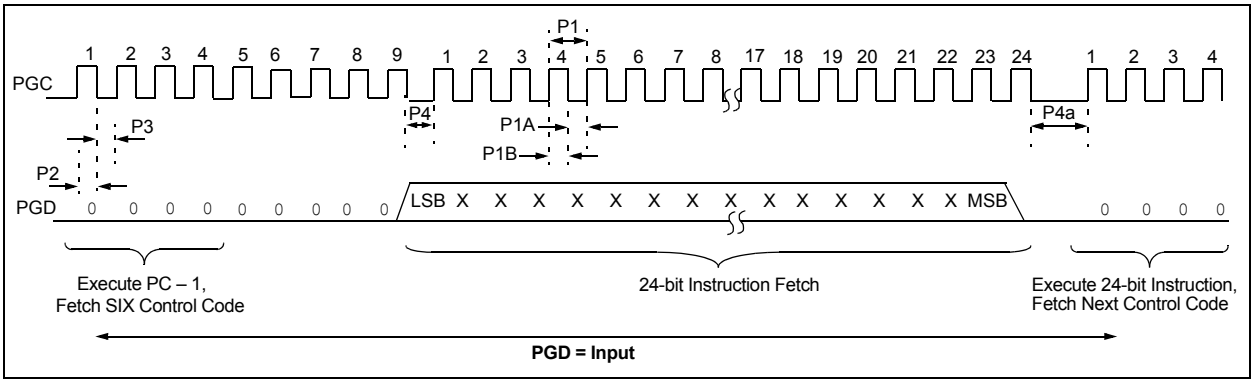


FIGURE 11-2: SIX SERIAL EXECUTION

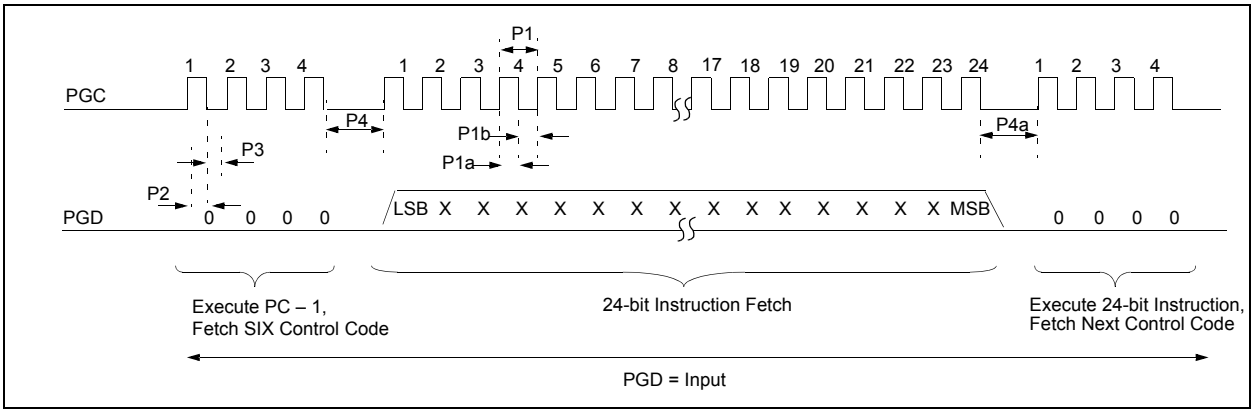


FIGURE 11-3: REGOUT SERIAL EXECUTION

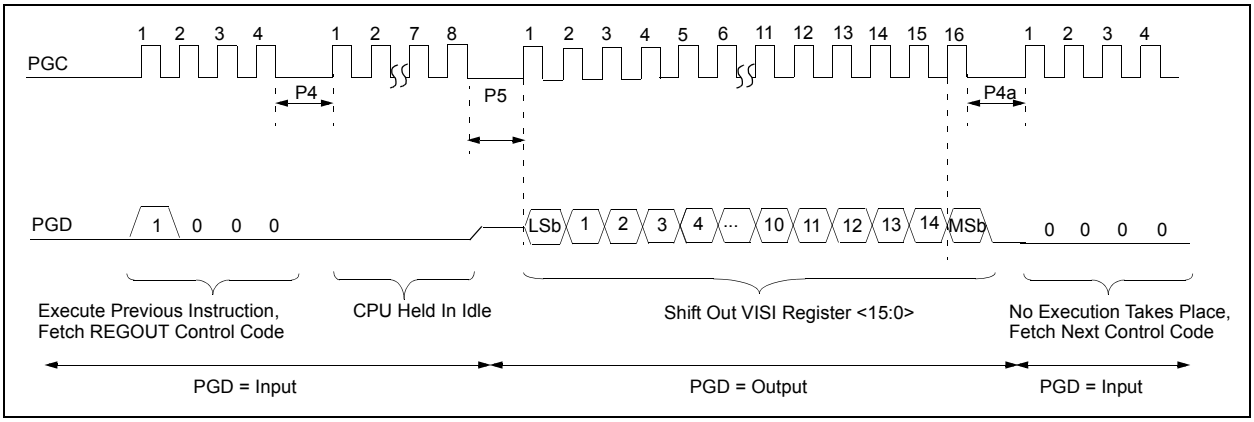


Table 11-4 shows the ICSP programming process for bulk-erasing program memory. This process includes the ICSP command code, which must be transmitted (for each instruction) to the Least Significant bit first using the PGC and PGD pins (see Figure 11-2).

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 0000 0000	040100 040100 000000	GOTO 0x100 GOTO 0x100 NOP
Step 2: Set NVMCON to program the FBS Configuration register.(1)		
0000 0000	24008A 883B0A	MOV #0x4008, W10 MOV W10, NVMCON
Step 3: Initialize the TBLPAG and write pointer (W7) for TBLWT instruction for Configuration register.(1)		

0000
0000
0000
	200F80
	880190
	200067
		MOV #0xF8, W0
		MOV W0, TBLPAG
		MOV #0x6, W7

Step 4: Load the Configuration Register data to W6.(1)		
0000 0000	EB0300 000000	CLR W6 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
0000
0000
0000
	200558
	200AA9
	883B38
	883B39
		MOV #0x55, W8
		MOV #0xAA, W9
		MOV W8, NVMKEY
		MOV W9, NVMKEY

Step 7: Initiate the programming cycle.(1)		
0000 0000 0000 – 0000 0000 0000 0000 0000	A8E761 000000 000000 – 000000 000000 A9E761 000000 000000	BSET NVMCON, #WR NOP NOP Externally time 2 ms NOP NOP BCLR NVMCON, #WR NOP 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 0000	2407FA 883B0A	MOV #0x407F, W10 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.

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

Command (Binary)	Data (Hexadecimal)	Description
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
Step 11: 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

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.

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

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

11.9 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	040100	GOTO 0x100
0000	040100	GOTO 0x100
0000	000000	NOP
Step 2: Set the NVMCON to write 16 data words.		
0000	24005A	MOV #0x4005, W10
0000	883B0A	MOV W10, NVMCON
Step 3: Initialize the write pointer (W7) for TBLWT instruction.		
0000	2007F0	MOV #0x7F, W0
0000	880190	MOV W0, TBLPAG
0000	2xxxx7	MOV #<DestinationAddress15:0>, W7
Step 4: Load W0:W3 with the next 4 data words to program.		
0000	2xxxx0	MOV #<WORD0>, W0
0000	2xxxx1	MOV #<WORD1>, W1
0000	2xxxx2	MOV #<WORD2>, W2
0000	2xxxx3	MOV #<WORD3>, W3
Step 5: Set the read pointer (W6) and load the (next set of) write latches.		
0000	EB0300	CLR W6
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [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 four times to load the write latches for 16 data words.		

11.11 Reading Configuration Memory

The procedure for reading configuration memory is similar to the procedure for reading code memory, except that 16-bit data words are read instead of 24-bit words. Since there are seven Configuration registers, they are read one register at a time.

Table 11-11 shows the ICSP programming details for reading all of the configuration memory. Note that the TBLPAG register is hard-coded to 0xF8 (the upper byte address of configuration memory), and the read pointer W6 is initialized to 0x0000.

TABLE 11-11: SERIAL INSTRUCTION EXECUTION FOR READING ALL CONFIGURATION 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) and the write pointer (W7) for TBLRD instruction.		
0000	200F80	MOV #0xF8, W0
0000	880190	MOV W0, TBLPAG
0000	EB0300	CLR W6
0000	EB0380	CLR W7
0000	000000	NOP
Step 3: Read the Configuration register and write it to the VISI register (located at 0x784).		
0000	BA0BB6	TBLRDL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0000	883C20	MOV W0, VISI
0000	000000	NOP
Step 4: Output the VISI register using the REGOUT command.		
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 six times to read all of configuration memory.		

11.13 Reading the Application ID Word

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

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

11.14 Exiting ICSP Mode

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

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

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

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	040100	GOTO 0x100
0000	040100	GOTO 0x100
0000	000000	NOP
Step 2: Initialize the NVMCON to erase executive memory.		
0000	24072A	MOV #0x4072, W10
0000	883B0A	MOV W10, NVMCON
Step 3: 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 4: Initiate the erase cycle.		
0000	A8E761	BSET NVMCON, #15
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, #15
0000	000000	NOP
0000	000000	NOP
Step 5: Initialize the TBLPAG and the write pointer (W7).		
0000	200800	MOV #0x80, W0
0000	880190	MOV W0, TBLPAG
0000	EB0380	CLR W7
0000	000000	NOP
0000	000000	NOP
Step 6: Initialize the NVMCON to program 32 instruction words.		
0000	24001A	MOV #0x4001, W10
0000	883B0A	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	2<LSW0>0	MOV #<LSW0>, W0
0000	2<MSB1:MSB0>1	MOV #<MSB1:MSB0>, W1
0000	2<LSW1>2	MOV #<LSW1>, W2
0000	2<LSW2>3	MOV #<LSW2>, W3
0000	2<MSB3:MSB2>4	MOV #<MSB3:MSB2>, W4
0000	2<LSW3>5	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.		

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

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 $\overline{MCLR}$/VPP	9.00	13.25	V	—
D112	IPP	Programming Current on $\overline{MCLR}$/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 $\overline{MCLR}$/VPP	100	—	ns	—
P7	THLD2	Input data hold time from $\overline{MCLR}$/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	—

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