

dsPIC33EPXXXGS70X/80X Family Flash Programming Specification

1.0 DEVICE OVERVIEW

This document defines the programming specification for the dsPIC33EPXXXGS70X/80X 16-bit Digital Signal Controller (DSC) family. This programming specification is required only for those developing programming support for the following devices:

- | | |
|---------------------|---------------------|
| • dsPIC33EP64GS708 | • dsPIC33EP128GS705 |
| • dsPIC33EP64GS804 | • dsPIC33EP128GS706 |
| • dsPIC33EP64GS805 | • dsPIC33EP128GS708 |
| • dsPIC33EP64GS806 | • dsPIC33EP128GS804 |
| • dsPIC33EP64GS808 | • dsPIC33EP128GS805 |
| • dsPIC33EP128GS702 | • dsPIC33EP128GS806 |
| • dsPIC33EP128GS704 | • dsPIC33EP128GS808 |

Customers using only one of these devices should use development tools that already provide support for device programming.

Topics covered include:

- [Section 1.0 “Device Overview”](#)
- [Section 2.0 “Programming Overview”](#)
- [Section 3.0 “Device Programming – ICSP”](#)
- [Section 4.0 “Device Programming – Enhanced ICSP”](#)
- [Section 5.0 “Programming the Programming Executive to Memory”](#)
- [Section 6.0 “The Programming Executive”](#)
- [Section 7.0 “Dual Partition Flash Programming Considerations”](#)
- [Section 8.0 “Device ID/Unique ID”](#)
- [Section 9.0 “Checksum Computation”](#)
- [Section 10.0 “AC/DC Characteristics and Timing Requirements”](#)

2.0 PROGRAMMING OVERVIEW

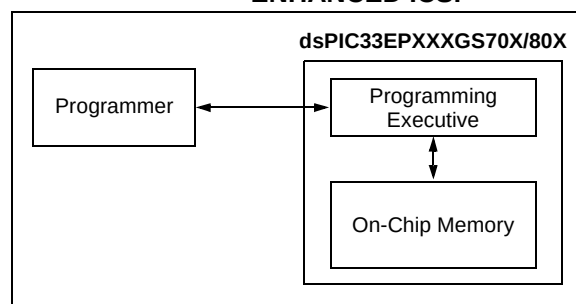
The following are the two methods of programming that are discussed in this programming specification:

- In-Circuit Serial Programming™ (ICSP™)
- Enhanced In-Circuit Serial Programming

The ICSP programming method is the most direct method to program the device; however, it is also the slower of the two methods. It provides native, low-level programming capability to erase, program and verify the device.

The Enhanced ICSP protocol uses a faster method that takes advantage of the Programming Executive (PE), as illustrated in [Figure 2-1](#). The PE provides all the necessary functionality to erase, program and verify the chip through a small command set. The command set allows the programmer to program a dsPIC33EPXXXGS70X/80X device without dealing with the low-level programming protocols.

FIGURE 2-1: PROGRAMMING SYSTEM OVERVIEW FOR ENHANCED ICSP™



This programming specification is divided into two major sections that describe the programming methods independently. [Section 3.0 “Device Programming – ICSP”](#) describes the ICSP method. [Section 4.0 “Device Programming – Enhanced ICSP”](#) describes the Enhanced ICSP method.

dsPIC33EPXXXGS70X/80X FAMILY

2.1 Required Connections

These devices require specific connections for programming to take place. These connections include power, VCAP, MCLR and one programming pair (PGDx/PGCx). [Table 2-1](#) describes these connections (refer to the specific device data sheet for pin descriptions and power connection requirements).

2.2 Power Requirements

All dsPIC33EPXXXGS70X/80X devices power their core digital logic at a nominal 1.8V. To simplify system design, all devices in the dsPIC33EPXXXGS70X/80X family incorporate an on-chip regulator that allows the device to run its core logic from VDD.

The regulator provides power to the core from the other VDD pins. A low-ESR capacitor (such as ceramic or tantalum) must be connected to the VCAP pin (see [Table 2-1](#) and [Figure 2-2](#)); this helps to maintain the stability of the regulator. The specifications for core voltage and capacitance are listed in [Section 10.0 “AC/DC Characteristics and Timing Requirements”](#).

FIGURE 2-2: CONNECTIONS FOR THE ON-CHIP REGULATOR

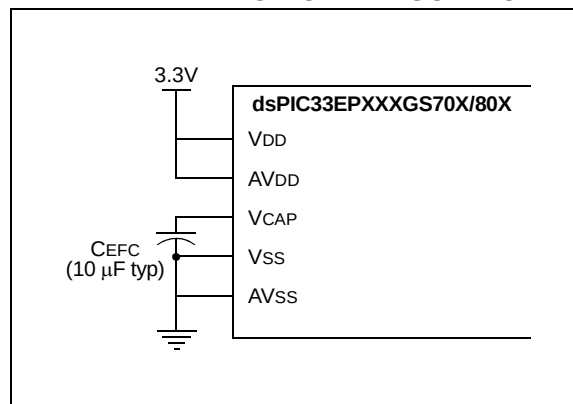


TABLE 2-1: PINS USED DURING PROGRAMMING

Pin Name	Pin Type	Pin Description
MCLR	I	Programming Enable
VDD and AVDD ⁽¹⁾	P	Power Supply ⁽¹⁾
VSS and AVSS ⁽¹⁾	P	Ground ⁽¹⁾
VCAP	P	Internal Voltage Regulator Filter Capacitor
PGCx	I	Programming Pin Pair: Serial Clock
PGDx	I/O	Programming Pin Pair: Serial Data

Legend: I = Input O = Output P = Power

Note 1: All power supply and ground pins must be connected, including AVDD and AVSS.

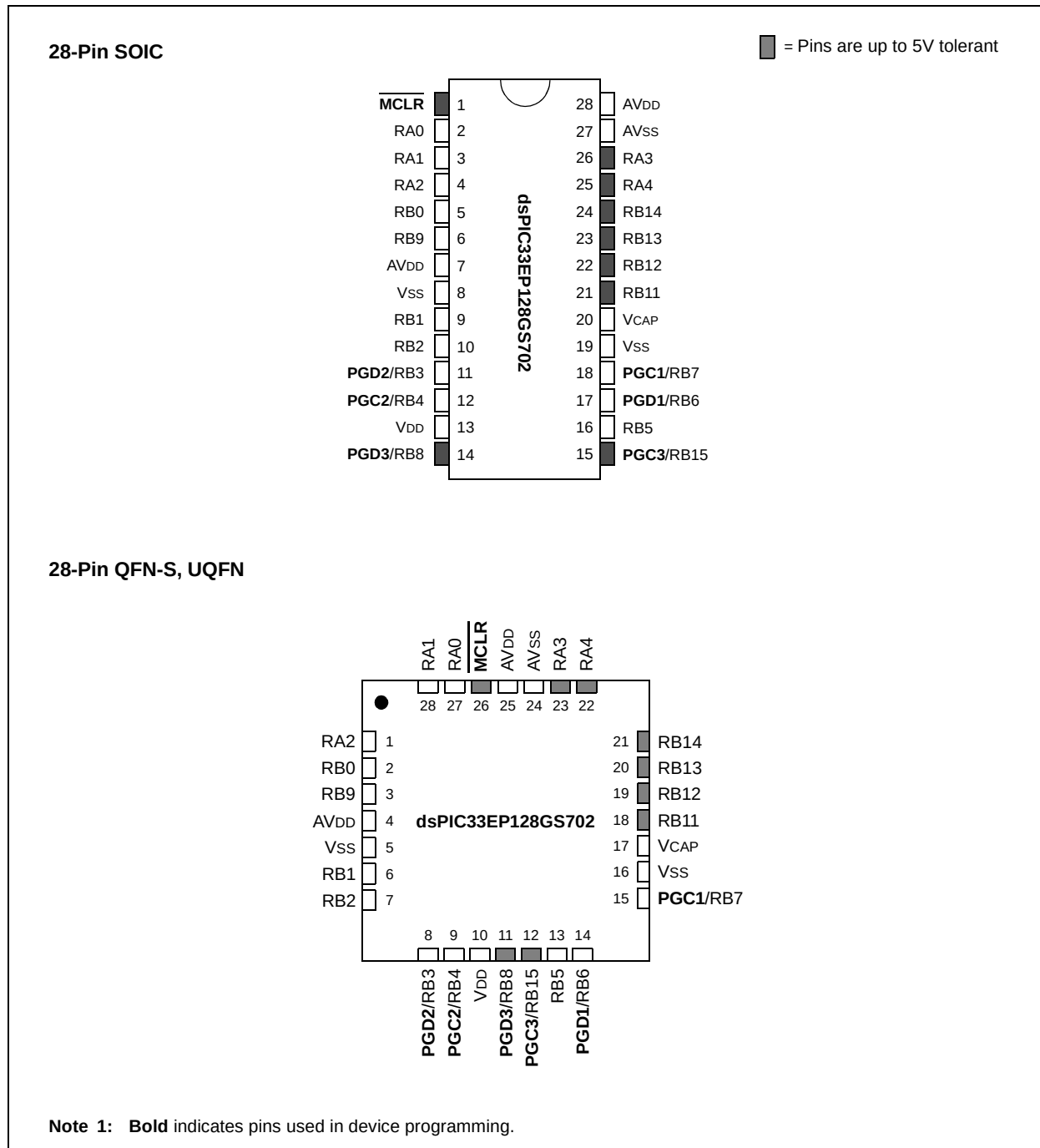
2.3 Pin Diagrams

Figure 2-3 through Figure 2-7 provide the pin diagrams for the dsPIC33EPXXXGS70X/80X family. The pins that are required for programming are listed in Table 2-1 and are indicated in bold text in the figures. Refer to the specific device data sheet for complete pin descriptions.

2.3.1 PGCx AND PGDx PIN PAIRS

All devices in the dsPIC33EPXXXGS70X/80X family have three separate pairs of programming pins, labeled as PGC1/PGD1, PGC2/PGD2 and PGC3/PGD3. Any one of these pin pairs may be used for device programming by either ICSP or Enhanced ICSP. Unlike voltage supply and ground pins, it is not necessary to connect all three pin pairs to program the device. However, the programming method must use both pins of the same pair.

FIGURE 2-3: PIN DIAGRAMS



dsPIC33EPXXXGS70X/80X FAMILY

FIGURE 2-4: PIN DIAGRAMS (CONTINUED)

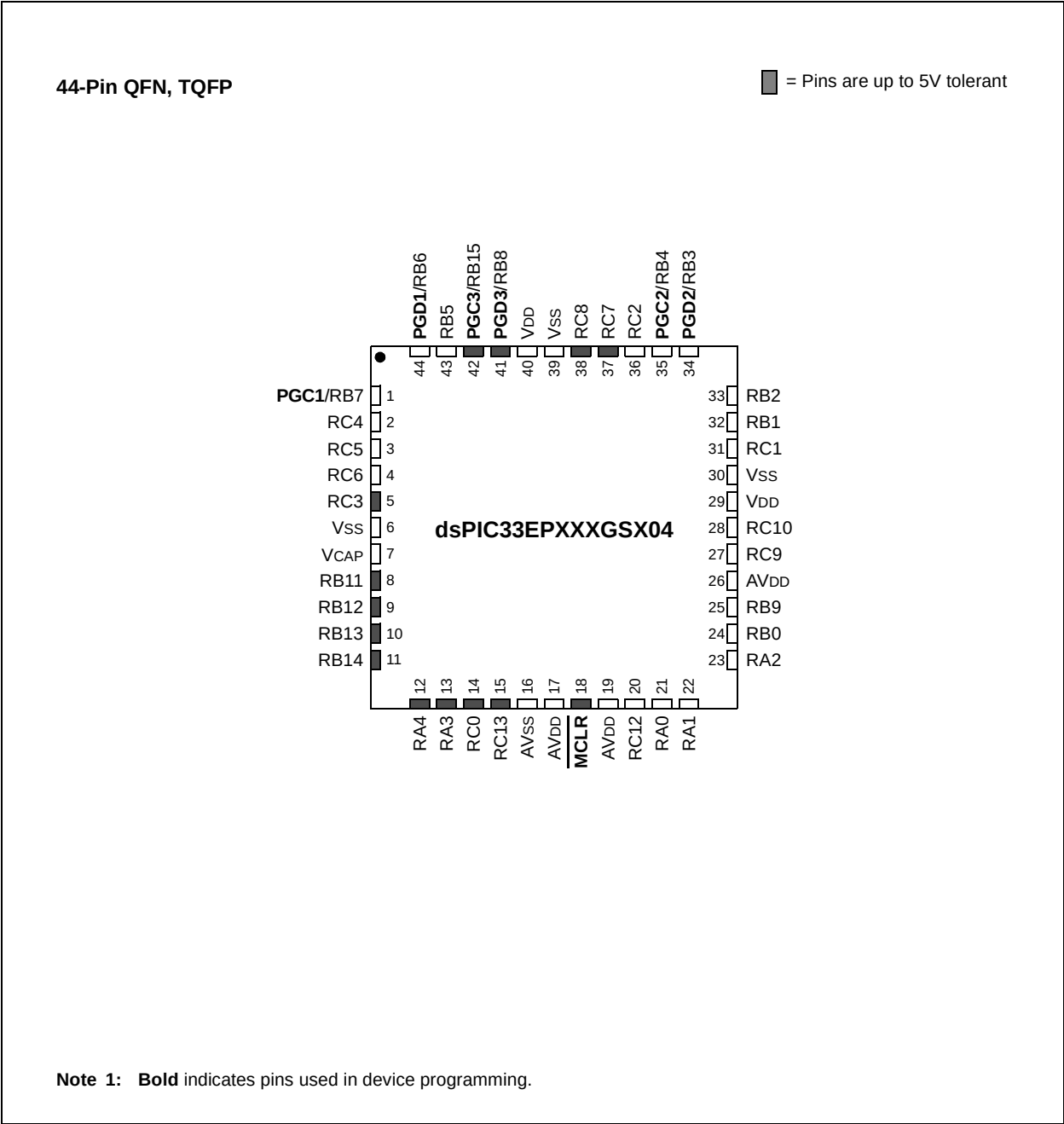
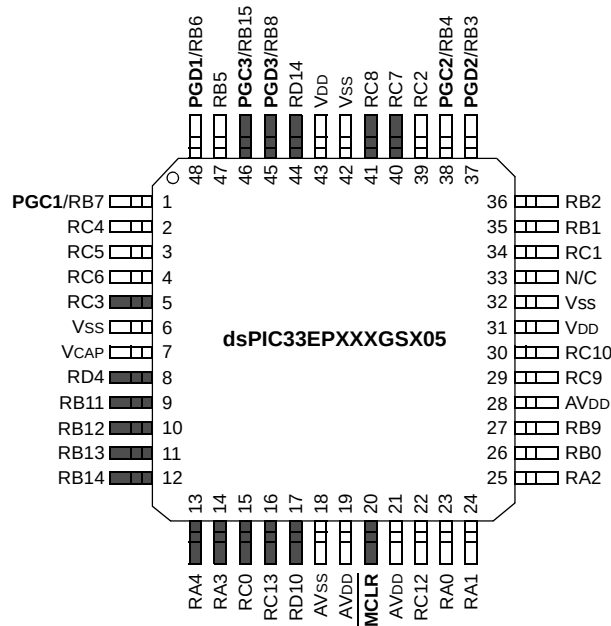


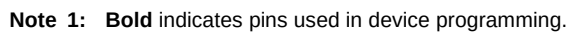
FIGURE 2-5: PIN DIAGRAMS (CONTINUED)

48-Pin TQFP

 = Pins are up to 5V tolerant

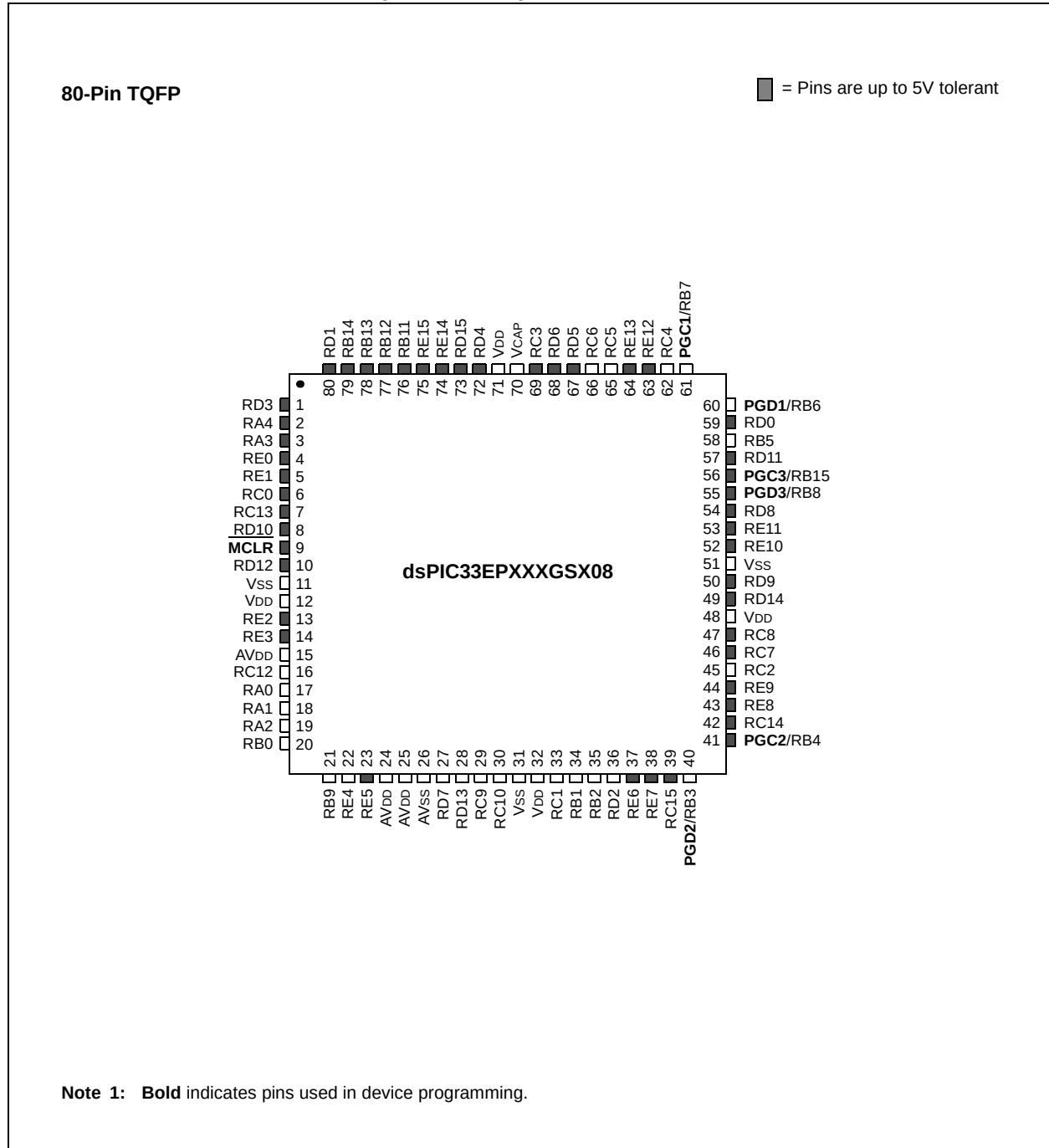


Note 1: **Bold** indicates pins used in device programming.



dsPIC33EPXXXGS70X/80X FAMILY

FIGURE 2-7: PIN DIAGRAMS (CONTINUED)



dsPIC33EPXXXGS70X/80X FAMILY

2.4 Program Memory Write/Erase Requirements

The program Flash memory has a specific write/erase requirement that must be adhered to for proper device operation. The rule is that any given word in memory must not be written without first erasing the page in which it is located. Thus, the easiest way to conform to this rule is to write all the data in a programming block within one write cycle. The programming methods specified in this document comply with this requirement.

Note: A program memory word can be programmed twice before an erase, but only if (a) the same data is used in both program operations or (b) bits containing '1' are set to '0', but no '0' is set to '1'.

2.5 Memory Map

The program memory map extends from 000000h to FFFFFFFh. Code storage is located at the base of the memory map. The last locations of implemented program memory are reserved for the device Configuration bits.

Table 2-2 lists the code memory size, the size of the erase blocks and the number of erase blocks present in each device variant.

Locations, 800200h through 800BFEh, are reserved for executive code memory. This region stores the Programming Executive (PE) and the debugging executive, which is used for device programming. This region of memory cannot be used to store user code. See [Section 6.0 “The Programming Executive”](#) for more information.

Locations, FF0000h and FF0002h, are reserved for the Device ID Word registers. These bits can be used by the programmer to identify which device type is being programmed. They are described in [Section 8.0 “Device ID/Unique ID”](#). The Device ID registers read out normally, even after code protection is applied.

The locations, 800F80h to 800FFEh, are a One-Time-Programmable (OTP) memory area. The user OTP Words can be used for storing product information, such as serial numbers, system manufacturing dates, manufacturing lot numbers and other application-specific information. They are described in [Section 2.6.3 “User OTP \(One-Time-Programmable\) Memory”](#).

Figure 2-8 through Figure 2-11 show a generic memory map for the devices listed in Table 2-2. See the “Memory Organization” chapter in the specific device data sheet for exact memory addresses.

TABLE 2-2: SINGLE PARTITION FLASH CODE MEMORY SIZE

Device Family	User Memory Address Limit (Instruction Words)	Erase Blocks/ No. of Pages ⁽¹⁾	Executive Memory Address Limit (Instruction Words)
dsPIC33EP64GS708	00AF7Eh (22,464)	44	0x800200-0x800BFE
dsPIC33EP64GS804			
dsPIC33EP64GS805			
dsPIC33EP64GS806			
dsPIC33EP64GS808			
dsPIC33EP128GS702	01577Eh (43,968)	86	
dsPIC33EP128GS704			
dsPIC33EP128GS705			
dsPIC33EP128GS706			
dsPIC33EP128GS708			
dsPIC33EP128GS804			
dsPIC33EP128GS805			
dsPIC33EP128GS806			
dsPIC33EP128GS808			

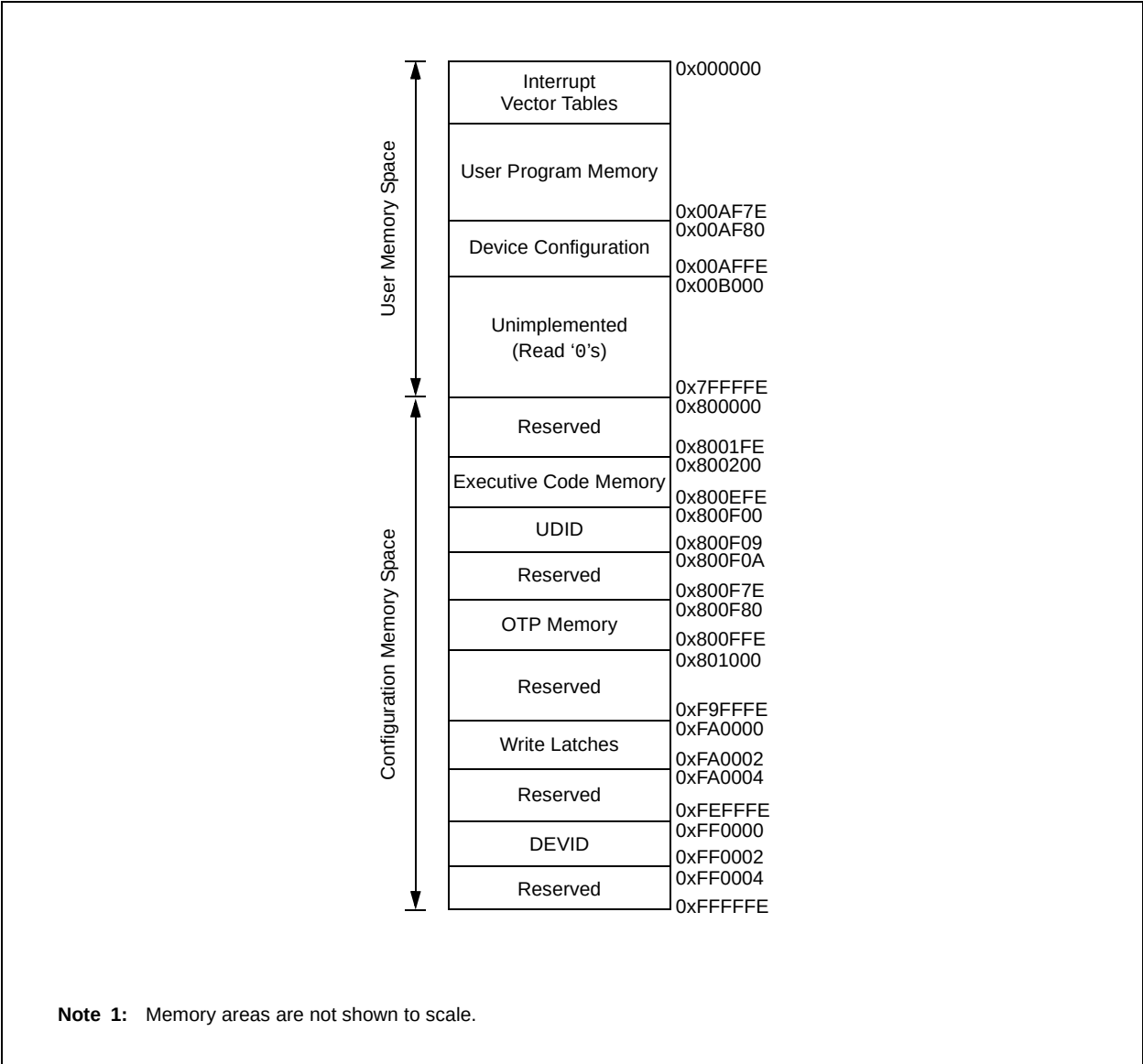
Note 1: One erase block is equal to 512 24-bit instruction words.

TABLE 2-3: CONFIGURATION REGISTER ADDRESSES

Device	FSEC	FBSLIM	FSIGN	FOSCSSEL	FOSC	FWDT	FPOR	FICD	FDEVOPT	FALTREG	FBTSEQ	FBOOT
dsPIC33EP64GS708	00AF80	00AF90	00AF94	00AF98	00AF9C	00AFA0	00AFA4	00AFA8	00AFAC	00AFB0	00AFFC	801000
dsPIC33EP64GS804												
dsPIC33EP64GS805												
dsPIC33EP64GS806												
dsPIC33EP64GS808												
dsPIC33EP128GS702	015780	015790	015794	015798	01579C	0157A0	0157A4	0157A8	0157AC	0157B0	0157FC	801000
dsPIC33EP128GS704												
dsPIC33EP128GS705												
dsPIC33EP128GS706												
dsPIC33EP128GS708												
dsPIC33EP128GS804												
dsPIC33EP128GS805												
dsPIC33EP128GS806												
dsPIC33EP128GS808												

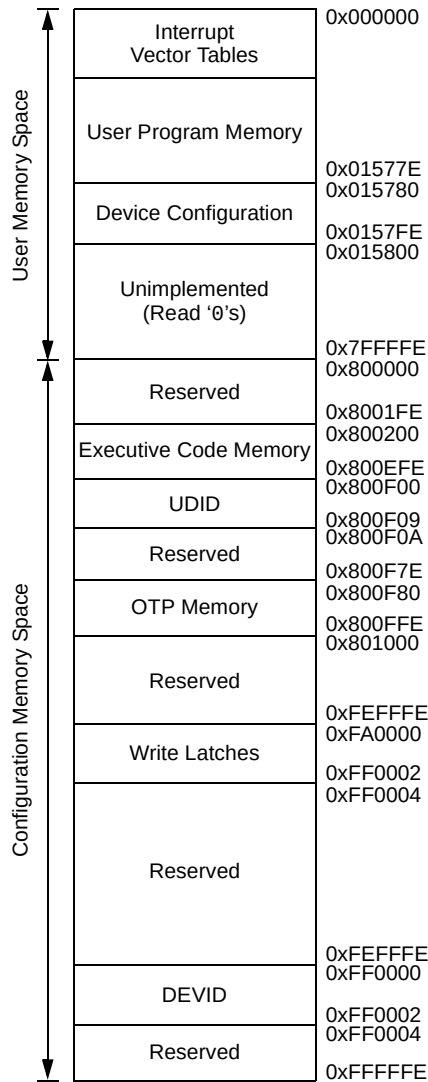
dsPIC33EPXXXGS70X/80X FAMILY

FIGURE 2-8: PROGRAM MEMORY MAP FOR dsPIC33EP64GS70X/80X DEVICES⁽¹⁾



dsPIC33EPXXXGS70X/80X FAMILY

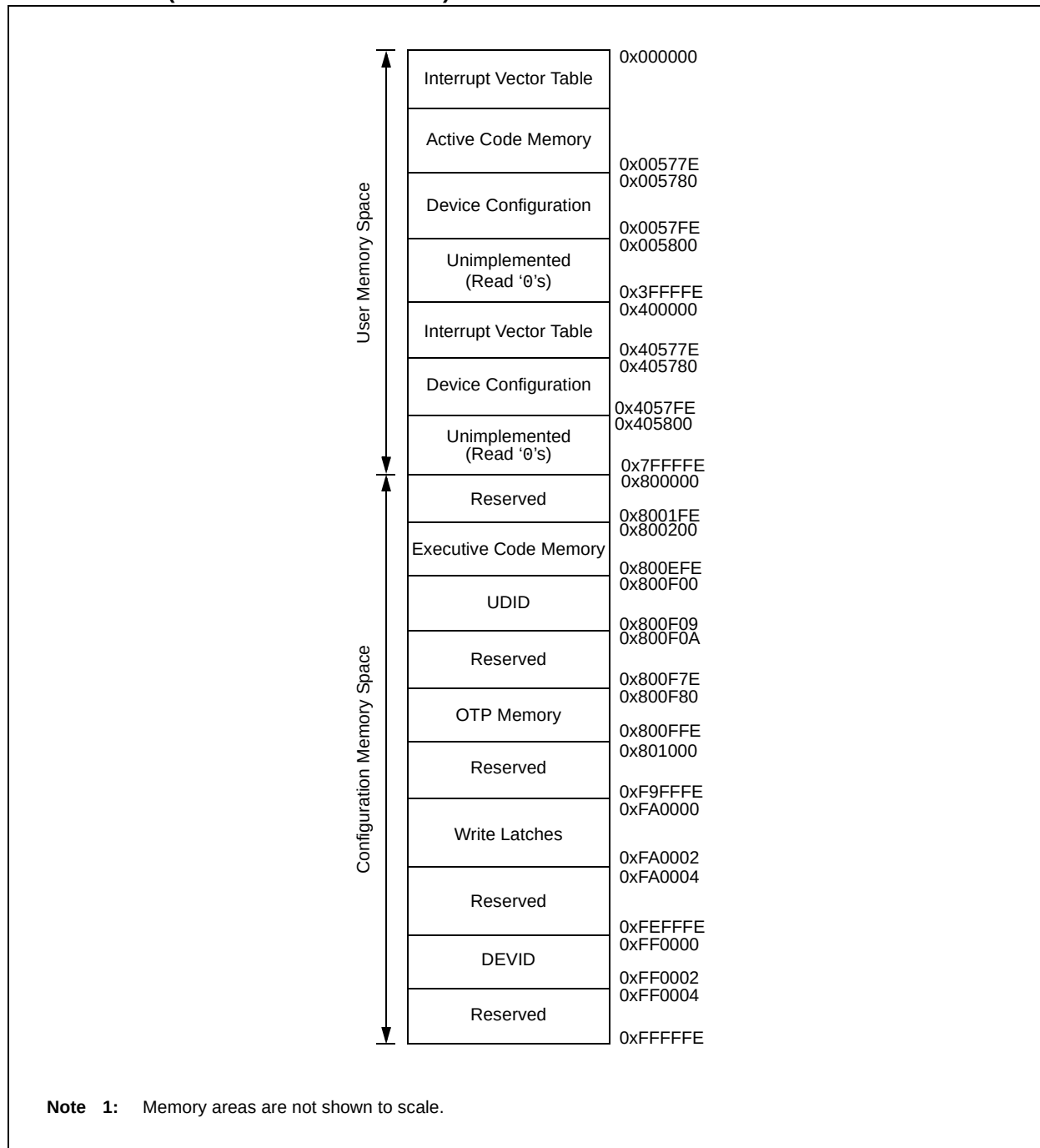
FIGURE 2-9: PROGRAM MEMORY MAP FOR dsPIC33EP128GS70X/80X DEVICES⁽¹⁾



Note 1: Memory areas are not shown to scale.

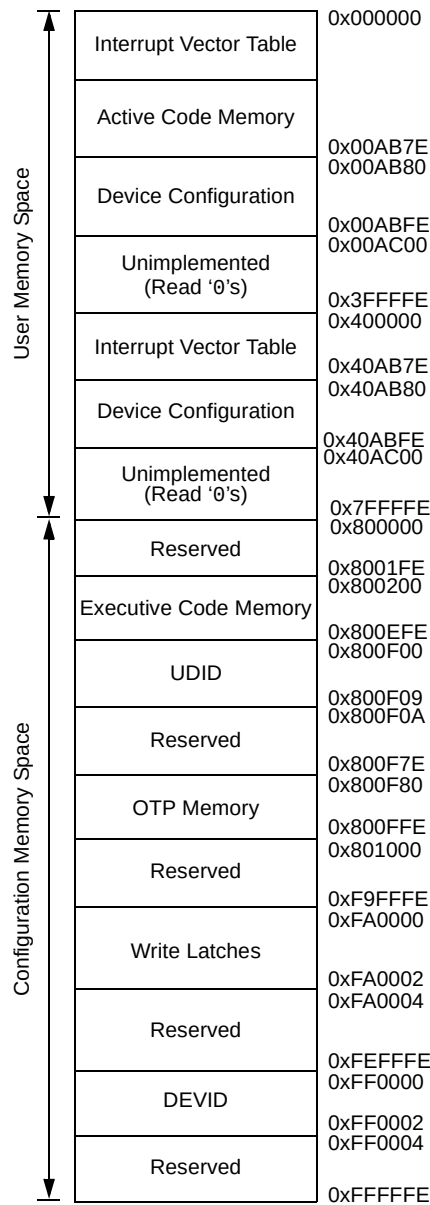
dsPIC33EPXXXGS70X/80X FAMILY

**FIGURE 2-10: PROGRAM MEMORY MAP FOR DSPIC33EP64GS70X/80X DEVICES
(DUAL PARTITION FLASH)⁽¹⁾**



dsPIC33EPXXXGS70X/80X FAMILY

FIGURE 2-11: PROGRAM MEMORY MAP FOR dsPIC33EP128GS70X/80X DEVICES (DUAL PARTITION FLASH)⁽¹⁾



Note 1: Memory areas are not shown to scale.

2.6 Configuration Bits

2.6.1 OVERVIEW

The Configuration bits are stored in the last page location of implemented program memory. 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 the Watchdog Timer. The code-protect bits prevent program memory from being read and written.

Table 2-3 and Table 2-4 list the Configuration register address range for each device.

Table 2-4 is an example of a Configuration register map. Refer to the “**Special Features**” chapter in the specific device data sheet for the full Configuration Word register descriptions for your device.

2.6.2 CODE-PROTECT CONFIGURATION BITS

The devices implement intermediate security features defined by the FSEC register. The Boot Segment (BS) is the highest privileged segment and the General Segment (GS) is the lowest privileged segment. The total user code memory can be split into BS or GS. The size of the segments is determined by the BSLIM<12:0> bits. The relative location of the segments within user space does not change, such that BS (if present) occupies the memory area just after VS (Interrupt Vector Table, IVT), and the GS occupies the space just after BS (or if the Alternate Interrupt Vector Table (AIVT) is enabled, just after AIVT VS). The Configuration Segment (CS) is a small segment (less than a page, typically just one row) within user Flash address space that contains all user configuration data that is loaded by the NVM controller during the Reset sequence.

2.6.3 USER OTP (ONE-TIME-PROGRAMMABLE) MEMORY

The dsPIC33EPXXXGS70X/80X family devices provide 64 words (32 address locations of 24-bit wide memory) of One-Time-Programmable (OTP) memory, located at addresses, 800F80h through 800FFEh. This memory can be used for persistent storage of application-specific information that will not be erased by reprogramming the device. This includes many types of information, such as:

- Application checksums
- Code revision information
- Product information
- Serial numbers
- System manufacturing dates
- Manufacturing lot numbers

Customer OTP memory may be programmed in any mode, including user RTSP mode, but it cannot be erased. Data is not cleared by a Chip Erase.

TABLE 2-4: CONFIGURATION REGISTER MAP⁽²⁾

File Name	Address	Device Memory Size (Kbytes)	Bits 23-16	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
FSEC	00AF80	64	—	AIVTDIS	—	—	—	CSS<2:0>			CWRP	GSS<1:0>		GWRP	—	BSEN	BSS<1:0>		BWRP
	015780	128																	
FBSLIM	00AF90	64	—	—	—	—	BSLIM<12:0>												
	015790	128																	
FSIGN	00AF94	64	—	Reserved ⁽⁴⁾	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	015794	128																	
FOSCSSEL	00AF98	64	—	—	—	—	—	—	—	—	IESO	—	—	—	—	—	FNOSC<2:0>		
	015798	128																	
FOSC	00AF9C	64	—	—	—	—	—	—	—	PLLKEN	FCKSM<1:0>		IOL1WAY	—	—	OSCIOFNC	POSCMD<1:0>		
	01579C	128																	
FWDI	00AFA0	64	—	—	—	—	—	—	—	WDTWIN<1:0>		WINDIS	WDTEN<1:0>		WDTPRE	WDTPOST<3:0>			
	0157A0	128																	
FPOR	00AFA4	64	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Reserved ⁽¹⁾
	0157A4	128																	
FICD	00AFA8	64	—	BTSWP	—	—	—	—	—	—	Reserved ⁽¹⁾	—	JTAGEN	—	—	—	—	ICS<1:0>	
	0157A8	128																	
FDEVOPT	00AFAC	64	—	—	—	—	—	—	—	—	—	—	DBCC	—	ALTI2C<2:1>		Reserved ⁽¹⁾	—	PWMLOCK
	0157AC	128																	
FALTREG	00AFB0	64	—	—	CTXT4<2:0>			Reserved ⁽¹⁾	CTXT3<2:0>			Reserved ⁽¹⁾	CTXT2<2:0>			—	CTXT1<2:0>		
	0157B0	128																	
FBTSEQ	00AFFC	64	IBSEQ<11:0>					BSEQ<11:0>											
	0157FC	128																	
FBOOT ⁽³⁾	801000	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BTMODE<1:0>	

Legend: — = unimplemented, read as '1'.

Note 1: These bits are reserved and must be programmed as '1'.

2: When operating in Dual Partition Flash mode, each Flash partition will have dedicated Configuration registers. On a device Reset, the configuration values of the Active Partition are read at start-up, but during a soft swap condition, the configuration settings of the newly Active Partition are ignored.

3: FBOOT resides in configuration memory space.

4: This bit is reserved and must be programmed as '0'.

dsPIC33EPXXXGS70X/80X FAMILY

TABLE 2-5: dsPIC33EPXXXGS70X/80X CONFIGURATION BITS DESCRIPTION

Bit Field	Description
BSS<1:0>	Boot Segment Code-Protect Level bits 11 = No code protection (other than BWRP code protection) 10 = Standard security 0x = High security
BSEN	Boot Segment Control bit 1 = Boot Segment is not enabled 0 = Boot Segment size is determined by the BSLIM<12:0> bits
BWRP	Boot Segment Write-Protect bit 1 = Boot Segment can be written 0 = Boot Segment is write-protected
BSLIM<12:0>	Boot Segment Flash Page Address Limit bits Contains the page address of the first active General Segment page. The value to be programmed is the inverted page address, such that programming additional '0's can only increase the Boot Segment size (i.e., 0x1FFD = 2 Pages or 1024 IW).
GSS<1:0>	General Segment Code-Protect Level bits 11 = User program memory is not code-protected 10 = Standard security 0x = High security
GWRP	General Segment Write-Protect bit 1 = User program memory is not write-protected 0 = User program memory is write-protected
CWRP	Configuration Segment Write-Protect bit 1 = Configuration Segment is not write-protected 0 = Configuration Segment is write-protected
CSS<2:0>	Configuration Segment Code-Protect Level bits 111 = Configuration data is not code-protected 110 = Standard security 10x = Enhanced security 0xx = High security
BTSWP	BOOTSWP Instruction Enable/Disable bit 1 = BOOTSWP instruction is disabled 0 = BOOTSWP instruction is enabled
BSEQ<11:0>	Boot Sequence Number bits (Dual Partition Flash modes only) Relative value defining which partition will be active after device Reset; the partition containing a lower boot number will be active.
IBSEQ<11:0>	Inverse Boot Sequence Number bits (Dual Partition Flash modes only) The one's complement of BSEQ<11:0>; must be calculated by the user and written for device programming. If BSEQx and IBSEQx are not complements of each other, the Boot Sequence Number is considered to be invalid.
AIVTDIS	Alternate Interrupt Vector Table bit ⁽¹⁾ 1 = Alternate Interrupt Vector Table is disabled 0 = Alternate Interrupt Vector Table is enabled if INTCON2<8> = 1
IESO	Two-Speed Oscillator Start-up Enable bit 1 = Starts up device with FRC, then automatically switches to the user-selected oscillator source when ready 0 = Starts up device with the user-selected oscillator source
PWMLOCK	PWM Lock Enable bit 1 = Certain PWM registers may only be written after a key sequence 0 = PWM registers may be written without a key sequence

Note 1: The Boot Segment must be present to use the Alternate Interrupt Vector Table.

dsPIC33EPXXXGS70X/80X FAMILY

TABLE 2-5: dsPIC33EPXXXGS70X/80X CONFIGURATION BITS DESCRIPTION (CONTINUED)

Bit Field	Description
FNOSC<2:0>	Initial Oscillator Source Selection bits 111 = Fast RC (FRC) Oscillator with Divide-by-N (FRCDIVN) 110 = Fast RC (FRC) Oscillator with Divide-by-16 101 = Low-Power RC Oscillator (LPRC) 100 = Reserved; do not use 011 = Primary (XT, HS, EC) Oscillator with PLL module 010 = Primary (XT, HS, EC) Oscillator 001 = Fast RC Oscillator with Divide-by-N with PLL module (FRCPLL) 000 = Fast RC (FRC) Oscillator
FCKSM<1:0>	Clock Switching Mode bits 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
IOL1WAY	Peripheral Pin Select Configuration bit 1 = Allows only one reconfiguration 0 = Allows multiple reconfigurations
OSCIOFNC	OSC2 Pin Function bit (except in XT and HS modes) 1 = OSC2 is the clock output 0 = OSC2 is the general purpose digital I/O pin
POSCMD<1:0>	Primary Oscillator Mode Select bits 11 = Primary Oscillator is disabled 10 = HS Crystal Oscillator mode 01 = XT Crystal Oscillator mode 00 = External Clock (EC) mode
WDTEN<1:0>	Watchdog Timer Enable bits 11 = Watchdog Timer is always enabled (LPRC oscillator cannot be disabled; clearing the SWDTEN bit in the RCON register will have no effect) 10 = Watchdog Timer is enabled/disabled by user software (LPRC can be disabled by clearing the SWDTEN bit in the RCON register) 01 = Watchdog Timer is enabled only while device is active and is disabled while in Sleep mode; software control is disabled in this mode 00 = Watchdog Timer and SWDTEN bit are disabled
WINDIS	Watchdog Timer Window Enable bit 1 = Watchdog Timer is in Non-Window mode 0 = Watchdog Timer is in Window mode
PLLKEN	PLL Lock Enable bit 1 = PLL lock is enabled 0 = PLL lock is disabled
WDTPRE	Watchdog Timer Prescaler bit 1 = 1:128 0 = 1:32
WDTPOST<3:0>	Watchdog Timer Postscaler bits 1111 = 1:32,768 1110 = 1:16,384 • • • 0001 = 1:2 0000 = 1:1

Note 1: The Boot Segment must be present to use the Alternate Interrupt Vector Table.

dsPIC33EPXXXGS70X/80X FAMILY

TABLE 2-5: dsPIC33EPXXXGS70X/80X CONFIGURATION BITS DESCRIPTION (CONTINUED)

Bit Field	Description
WDTWIN<1:0>	Watchdog Timer Window Select bits 11 = WDT window is 25% of the WDT period 10 = WDT window is 37.5% of the WDT period 01 = WDT window is 50% of the WDT period 00 = WDT window is 75% of the WDT period
ALTI2C1	Alternate I ² C pins for I2C1 bit 1 = I2C1 is mapped to the SDA1/SCL1 pins 0 = I2C1 is mapped to the ASDA1/ASCL1 pins
ALTI2C2	Alternate I ² C pins for I2C2 bit 1 = I2C2 is mapped to the SDA2/SCL2 pins 0 = I2C2 is mapped to the ASDA2/ASCL2 pins
JTAGEN	JTAG Enable bit 1 = JTAG is enabled 0 = JTAG is disabled
ICS<1:0>	ICD Communication Channel Select bits 11 = Communicates on PGC1 and PGD1 10 = Communicates on PGC2 and PGD2 01 = Communicates on PGC3 and PGD3 00 = Reserved, do not use
DBCC	DAC Output Cross-Connection bit 1 = DAC outputs are not cross-connected 0 = Interconnects DACOUT1 and DACOUT2
CTXT1<2:0>	Specifies Interrupt Priority Level (IPL) Associated to Alternate Working Register 1 bits 111 = Reserved 110 = Assigned to IPL of 7 101 = Assigned to IPL of 6 100 = Assigned to IPL of 5 011 = Assigned to IPL of 4 010 = Assigned to IPL of 3 001 = Assigned to IPL of 2 000 = Assigned to IPL of 1
CTXT2<2:0>	Specifies Interrupt Priority Level (IPL) Associated to Alternate Working Register 2 bits 111 = Reserved 110 = Assigned to IPL of 7 101 = Assigned to IPL of 6 100 = Assigned to IPL of 5 011 = Assigned to IPL of 4 010 = Assigned to IPL of 3 001 = Assigned to IPL of 2 000 = Assigned to IPL of 1
CTXT3<2:0>	Specifies Interrupt Priority Level (IPL) Associated to Alternate Working Register 3 bits 111 = Reserved 110 = Assigned to IPL of 7 101 = Assigned to IPL of 6 100 = Assigned to IPL of 5 011 = Assigned to IPL of 4 010 = Assigned to IPL of 3 001 = Assigned to IPL of 2 000 = Assigned to IPL of 1

Note 1: The Boot Segment must be present to use the Alternate Interrupt Vector Table.

dsPIC33EPXXXGS70X/80X FAMILY

TABLE 2-5: dsPIC33EPXXXGS70X/80X CONFIGURATION BITS DESCRIPTION (CONTINUED)

Bit Field	Description
CTXT4<2:0>	Specifies Interrupt Priority Level (IPL) Associated to Alternate Working Register 4 bits 111 = Reserved 110 = Assigned to IPL of 7 101 = Assigned to IPL of 6 100 = Assigned to IPL of 5 011 = Assigned to IPL of 4 010 = Assigned to IPL of 3 001 = Assigned to IPL of 2 000 = Assigned to IPL of 1
BTMODE<1:0>	dsPIC33EPXXXGS70X/80X Boot Configuration bits 11 = Device is operating in Single Partition Flash mode 10 = Device is operating in Dual Partition Flash mode 01 = Device is operating in Protected Dual Partition Flash mode 00 = Reserved; Do not use

Note 1: The Boot Segment must be present to use the Alternate Interrupt Vector Table.

3.0 DEVICE PROGRAMMING – ICSP

ICSP™ mode is a special programming protocol that allows you to read and write to device memory. The ICSP mode is the most direct method used to program the device, which is accomplished by applying control codes and instructions serially to the device, using the PGCx and PGDx pins. ICSP mode also has the ability to read executive memory to determine if the Programming Executive (PE) is present and to write the PE to executive memory if Enhanced ICSP mode will be used.

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

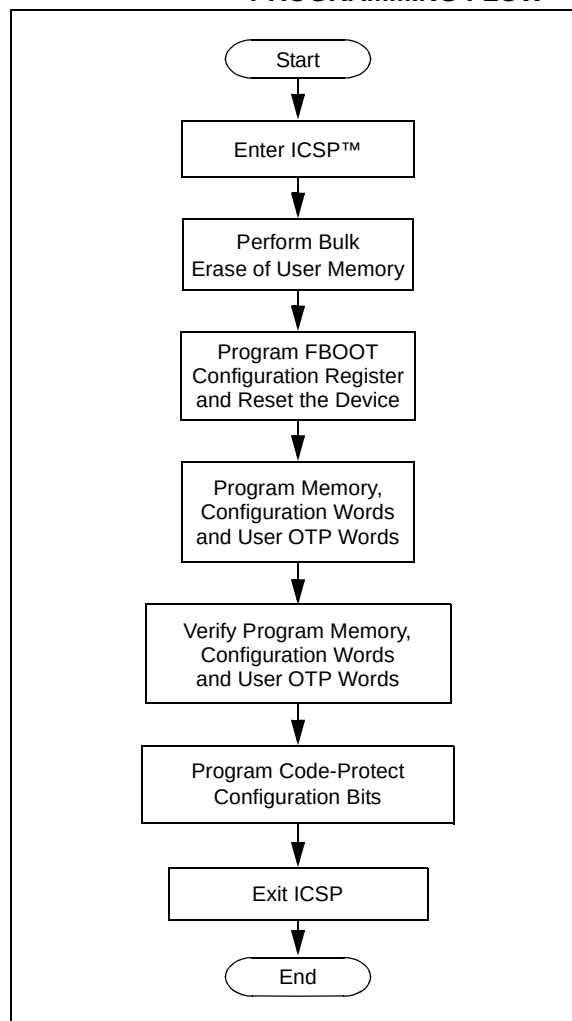
Note 1: During ICSP operation, the operating frequency of PGCx must not exceed 5 MHz.

2: ICSP mode is slower than Enhanced ICSP mode for programming.

3.1 Overview of the Programming Process

Figure 3-1 illustrates the high-level overview of the programming process. After entering ICSP mode, the first action is to Bulk Erase user program memory. Next, the code memory is programmed, followed by the device Configuration bits. Code memory (including the Configuration bits) is then verified to ensure that programming was successful. Then, the code-protect Configuration bits are programmed, if required.

FIGURE 3-1: HIGH-LEVEL ICSP™ PROGRAMMING FLOW



3.2 Entering ICSP Mode

As illustrated in Figure 3-2, entering ICSP Program/Verify mode requires four steps:

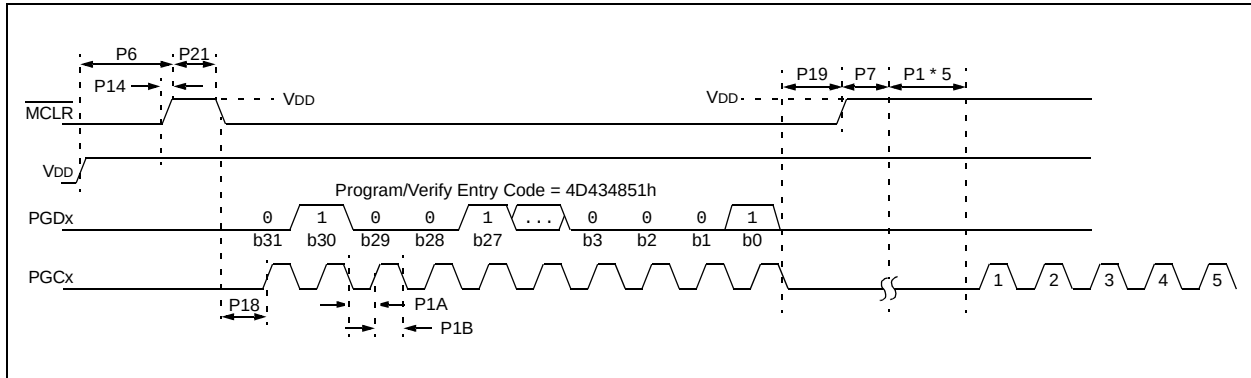
1. $\overline{\text{MCLR}}$ is briefly driven high and then low (P21).
2. A 32-bit key sequence is clocked into PGDx. An interval of at least P18 must elapse before presenting the key sequence on PGDx.
3. $\overline{\text{MCLR}}$ is held low during a specified period, P19, and then driven high.
4. After a $P7 + 5 * P1$ delay, five clock pulses must be generated on the PGCx pin.

Note: If a capacitor is present on the $\overline{\text{MCLR}}$ pin, the high time for entering ICSP mode can vary.

The key sequence is a specific 32-bit pattern, '0100 1101 0100 0011 0100 1000 0101 0001' (more easily remembered as 4D434851h in hexadecimal). The device will enter ICSP mode only if the sequence is valid. The Most Significant bit (MSb) of the most significant nibble must be shifted in first.

On successful ICSP mode entry, the program memory can be accessed and programmed in serial fashion.

FIGURE 3-2: ENTERING ICSP™ MODE



3.3 ICSP Operation

After entering into ICSP mode, the CPU is Idle. Execution of the CPU is governed by an internal state machine. A 4-bit control code is clocked in using PGCx and PGDx, and this control code is used to command the CPU (see [Table 3-1](#)).

The SIX control code is used to send instructions to the CPU for execution and the REGOUT control code is used to read data out of the device through the VISI register.

TABLE 3-1: CPU CONTROL CODES IN ICSP™ MODE

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

3.3.1 SIX SERIAL INSTRUCTION EXECUTION

The SIX control code allows execution of the family assembly instructions. When the SIX code is received, the CPU is suspended for 24 clock cycles, as the instruction is then clocked into the internal buffer. Once the instruction is shifted in, the state machine allows it to be executed over the next four PGCx clock cycles. While the received instruction is executed, the state machine simultaneously shifts in the next 4-bit command (see [Figure 3-3](#)).

Note: Data bits on PGDx are latched on the rising edge of the PGCx clock pulses.

3.3.1.1 Differences Between SIX Instruction Execution and Normal Instruction Execution

There are some differences between executing instructions using the SIX ICSP command and normal device instruction execution. As a result, the code examples in this specification might not match those required to perform the same operations during normal device operation.

The differences are:

- Two-word instructions require two SIX operations to clock in all of the necessary data. Examples of two-word instructions are GOTO and CALL.

- Two-cycle instructions require two SIX operations to complete. The first SIX operation shifts in the instruction and begins to execute it. A second SIX operation, which should shift in a NOP to avoid losing data, allows the required CPU clocks to finish executing the instruction.

Examples of two-cycle instructions are Table Read (TBLRD) and Table Write (TBLWT) instructions.

- The CPU does not automatically stall to account for pipeline changes. A CPU Stall occurs when an instruction modifies a register, which is used by the instruction immediately following the CPU Stall for Indirect Addressing. During normal operation, the CPU forces a NOP while the new data is read. To account for this, while using ICSP, any indirect references to a recently modified register should be proceeded with a NOP.

For example, MOV #0x0, W0, followed by, MOV [W0], W1, must have a NOP inserted in between.

If a two-cycle instruction modifies a register, which is used indirectly, it requires two following NOPs. One NOP executes the second half of the instruction and the other NOP stalls the CPU to correct the pipeline.

For example, TBLWTL [W0++], [W1], should be followed by two NOPs.

- The device Program Counter (PC) continues to automatically increment during the ICSP instruction execution, even though the Flash memory is not being used. As a result, it is possible for the PC to be incremented so that it points to invalid memory locations.

Examples of invalid memory spaces are unimplemented Flash addresses or the vector space (location: 0x0 to 0x1FF).

If the PC ever points to these locations, it causes the device to reset, possibly interrupting the ICSP operation. To prevent this, instructions should be periodically executed to reset the PC to a safe space. The optimal method of achieving this is to perform a "GOTO 0x200" instruction.

3.3.2 REGOUT SERIAL INSTRUCTION EXECUTION

The REGOUT control code allows the 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 and over the PGDx pin. After the REGOUT control code is received, the CPU is held Idle for 8 cycles. After this, an additional 16 cycles are required to clock the data out (see Figure 3-4).

The REGOUT code is unique as the PGDx pin is an input when the control code is transmitted to the device. However, after the control code is processed, the PGDx pin becomes an output as the VISI register is shifted out.

- Note 1:** After the contents of VISI are shifted out, the devices maintain PGDx as an output until the first rising edge of the next clock is received.
- 2:** Data changes on the falling edge and latches on the rising edge of PGCx. For all data transmissions, the Least Significant bit (LSb) is transmitted first.

FIGURE 3-3: SIX SERIAL EXECUTION

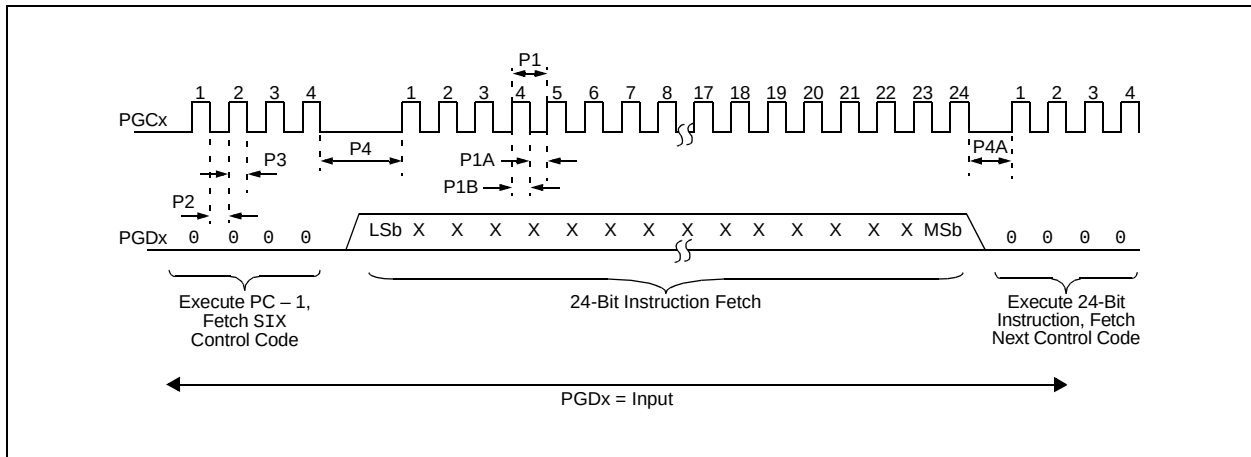
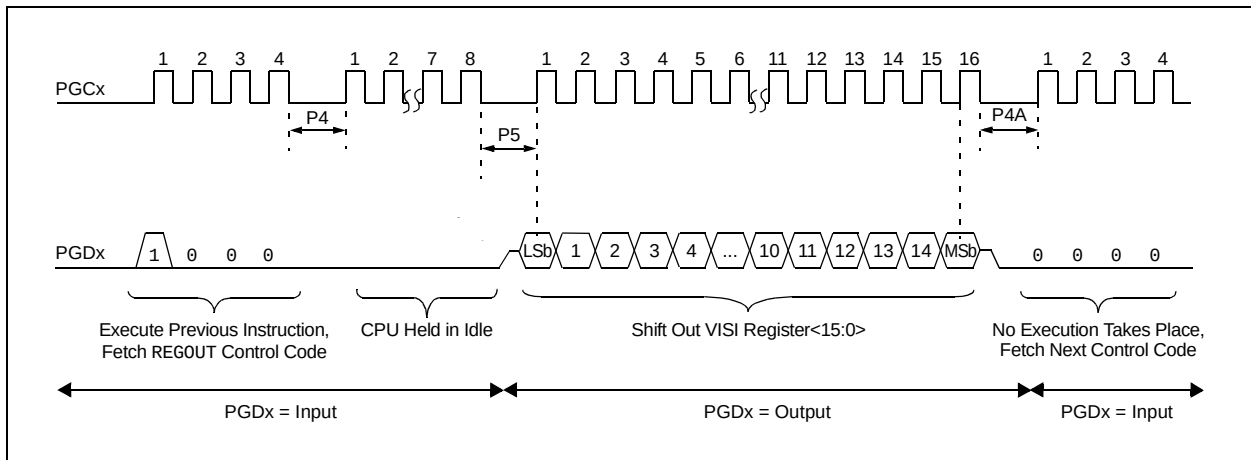


FIGURE 3-4: REGOUT SERIAL EXECUTION



3.4 Flash Memory Programming in ICSP Mode

3.4.1 PROGRAMMING OPERATIONS

Flash memory write/erase operations are controlled by the NVMCON register. Programming is performed by setting NVMCON to select the type of erase operation (Table 3-2) or write operation (Table 3-3), and initiating the programming by setting the WREN (Write Enable) bit (NVMCON<14>) and WR (Write Control) bit (NVMCON<15>).

In ICSP mode, all programming operations are self-timed. There is an internal delay between the user setting the WR bit and the automatic clearing of the WR bit when the programming operation is complete. Refer to [Section 10.0 “AC/DC Characteristics and Timing Requirements”](#) for detailed information about the delays associated with various programming operations.

TABLE 3-2: NVMCON ERASE OPERATIONS

NVMCON Value	Erase Operation
400Eh	Bulk Erase of user memory only (does not erase Device ID, executive memory and user OTP Words).
4003h	Erase a page of program or executive memory.
4004h	Inactive Partition Memory Erase operation.

TABLE 3-3: NVMCON WRITE OPERATIONS

NVMCON Value	Write Operation
4001h	Double-Word Program operation.
4008h	The next WR command will perform a Boot mode program (writing to FBOOT) and then program the Dual Partition Flash Signature (FSIGN) bit. Device must be reset before the newly programmed mode can take effect. This operation code should only be used on 64K memory parts.

3.4.2 STARTING AND STOPPING A PROGRAMMING CYCLE

For protection against accidental operations, the erase/write initiate sequence must be written to the NVMKEY register to allow any erase or program operation to proceed. The two instructions following the start of the programming sequence should be NOPs. To start an erase or write sequence, the following steps must be completed:

1. Write 55h to the NVMKEY register.
2. Write AAh to the NVMKEY register.
3. Set the WR bit in the NVMCON register.
4. Execute three NOP instructions.

All erase and write cycles are self-timed. The WR bit should be polled to determine if the erase or write cycle has been completed.

dsPIC33EPXXXGS70X/80X FAMILY

REGISTER 3-1: NVMCON: NONVOLATILE MEMORY (NVM) CONTROL REGISTER

R/SO-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0	R/C-0	R-0	R/W-0	R/C-0
WR	WREN	WRERR	NVMSIDL ⁽²⁾	SFTSWP ⁽⁶⁾	P2ACTIV ⁽⁶⁾	RPDF ⁽⁸⁾	URERR ⁽⁸⁾
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾
—	—	—	—	NVMOP3 ^(3,4)	NVMOP2 ^(3,4)	NVMOP1 ^(3,4)	NVMOP0 ^(3,4)
bit 7							bit 0

Legend:	C = Clearable bit	SO = Settable Only bit
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared
		x = Bit is unknown

- bit 15 **WR:** Write Control bit⁽¹⁾
 1 = Initiates a Flash memory program or erase operation; the operation is self-timed and the bit is cleared by hardware once operation is complete
 0 = Program or erase operation is complete and inactive
- bit 14 **WREN:** Write Enable bit⁽¹⁾
 1 = Enables Flash program/erase operations
 0 = Inhibits Flash program/erase operations
- bit 13 **WRERR:** Write Sequence Error Flag bit⁽¹⁾
 1 = An improper program/erase sequence attempt or termination has occurred (bit is set automatically on any set attempt of the WR bit)
 0 = The program/erase operation completed normally
- bit 12 **NVMSIDL:** NVM Stop in Idle Control bit⁽²⁾
 1 = Flash voltage regulator goes into Standby mode during Idle mode
 0 = Flash voltage regulator is active during Idle mode
- bit 11 **SFTSWP:** Panel Soft Swap Status bit⁽⁶⁾
 1 = Panels have been successfully swapped using the BOOTSWP instruction (soft swap)
 0 = Awaiting successful panel swap using the BOOTSWP instruction or a device Reset will determine the active panel based on the FBTSEQ Configuration register
- bit 10 **P2ACTIV:** Panel 2 Active Status bit⁽⁶⁾
 1 = Panel 2 Flash is mapped into the active region
 0 = Panel 1 Flash is mapped into the active region

Note 1: These bits can only be reset on a POR.

- 2:** If this bit is set, there will be minimal power savings (IDLE), and upon exiting Idle mode, there is a delay (TVREG) before Flash memory becomes operational.
- 3:** All other combinations of NVMOP<3:0> are unimplemented.
- 4:** Execution of the PWRSAV instruction is ignored while any of the NVM operations are in progress.
- 5:** Two adjacent words on a 4-word boundary are programmed during execution of this operation.
- 6:** Only available on dsPIC33EPXXXGS70X/80X devices operating in Dual Partition Flash mode. For all other devices, these bits are reserved.
- 7:** The specific Boot mode depends on bits<1:0> of the programmed data:
 11 = Single Partition Flash mode
 10 = Dual Partition Flash mode
 01 = Protected Dual Partition Flash mode
 00 = Reserved
- 8:** Not used in ICSP™ mode.

dsPIC33EPXXXGS70X/80X FAMILY

REGISTER 3-1: NVMCON: NONVOLATILE MEMORY (NVM) CONTROL REGISTER (CONTINUED)

bit 9	RPDF: Row Programming Data Format Control bit ⁽⁸⁾ 1 = Row data to be stored in RAM is in compressed format 0 = Row data to be stored in RAM is in uncompressed format
bit 8	URERR: Row Programming Data Underrun Error Flag bit ⁽⁸⁾ 1 = Row Program operation has been terminated due to data underrun error 0 = No data underrun has occurred
bit 7-4	Unimplemented: Read as '0'
bit 3-0	NVMOP<3:0>: NVM Operation Select bits ^(1,3,4) 1111 = Reserved 1110 = User memory Bulk Erase operation 1010 = Reserved 1001 = Reserved 1000 = Boot memory Double-Word Program operation in a Dual Partition Flash mode ⁽⁷⁾ 0101 = Reserved 0100 = Inactive Partition Memory Erase operation 0011 = Memory Page Erase operation 0010 = Memory Row Program operation ⁽⁸⁾ 0001 = Memory Double-Word Program operation ⁽⁵⁾ 0000 = Reserved

- Note 1:** These bits can only be reset on a POR.
- 2:** If this bit is set, there will be minimal power savings (IDLE), and upon exiting Idle mode, there is a delay (TVREG) before Flash memory becomes operational.
- 3:** All other combinations of NVMOP<3:0> are unimplemented.
- 4:** Execution of the PWRSAV instruction is ignored while any of the NVM operations are in progress.
- 5:** Two adjacent words on a 4-word boundary are programmed during execution of this operation.
- 6:** Only available on dsPIC33EPXXXGS70X/80X devices operating in Dual Partition Flash mode. For all other devices, these bits are reserved.
- 7:** The specific Boot mode depends on bits<1:0> of the programmed data:
11 = Single Partition Flash mode
10 = Dual Partition Flash mode
01 = Protected Dual Partition Flash mode
00 = Reserved
- 8:** Not used in ICSP™ mode.

3.5 Erasing Program Memory

The procedure for erasing the entire code memory using Bulk Erase is shown in [Figure 3-5](#).

[Figure 3-6](#) shows the procedure for erasing a page of code memory.

[Table 3-4](#) and [Table 3-5](#) illustrate the ICSP programming process for Bulk Erase and Page Erase, respectively.

- Note 1:** Program memory must be erased before writing any data to program memory.
- 2:** For Page Erase operations, the NVMCON value should be modified suitably according to [Table 3-2](#). The NVMADR/U registers should be pointing to any of the locations of the page to be erased.

FIGURE 3-5: BULK ERASE FLOW

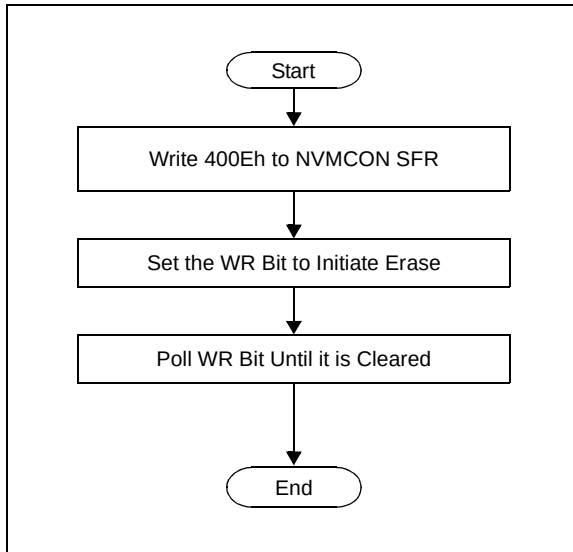
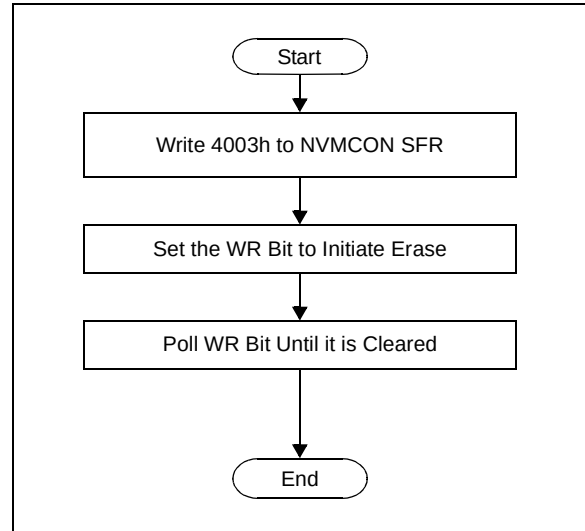


FIGURE 3-6: PAGE ERASE FLOW



dsPIC33EPXXXGS70X/80X FAMILY

TABLE 3-4: SERIAL INSTRUCTION EXECUTION FOR BULK ERASE OF CODE MEMORY

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 2: Set the NVMCON register to erase all user program memory.		
0000	2400EA	MOV #0x400E, W10
0000	88394A	MOV W10, NVMCON
0000	000000	NOP
0000	000000	NOP
Step 3: Initiate the erase cycle.		
0000	200551	MOV #0x55, W1
0000	883971	MOV W1, NVMKEY
0000	200AA1	MOV #0xAA, W1
0000	883971	MOV W1, NVMKEY
0000	A8E729	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 4: Generate clock pulses for the user memory Bulk Erase operation to complete until the WR bit is clear.		
0000	000000	NOP
0000	803940	MOV NVMCON, W0
0000	000000	NOP
0000	887C40	MOV W0, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
—	—	Repeat until the WR bit is clear.

dsPIC33EPXXXGS70X/80X FAMILY

TABLE 3-5: SERIAL INSTRUCTION EXECUTION FOR ERASING A PAGE OF CODE MEMORY

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 2: Set the NVMADRU/NVMADR register pair to point to the correct page to be erased.		
0000	2xxxx3	MOV #DestinationAddress<15:0>, W3
0000	2xxxx4	MOV #DestinationAddress<23:16>, W4
0000	883953	MOV W3, NVMADR
0000	883964	MOV W4, NVMADRU
Step 3: Set the NVMCON register to erase the first page of executive memory.		
0000	24003A	MOV #0x4003, W10
0000	88394A	MOV W10, NVMCON
0000	000000	NOP
0000	000000	NOP
Step 4: Initiate the erase cycle.		
0000	200551	MOV #0x55, W1
0000	883971	MOV W1, NVMKEY
0000	200AA1	MOV #0xAA, W1
0000	883971	MOV W1, NVMKEY
0000	A8E729	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 5: Generate clock pulses for Page Erase operation to complete until the WR bit is clear.		
0000	000000	NOP
0000	803940	MOV NVMCON, W0
0000	000000	NOP
0000	887C40	MOV W0, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
—	—	Repeat until the WR bit is clear.

dsPIC33EPXXXGS70X/80X FAMILY

3.6 Programming the FBOOT Configuration Register

Before code memory, Configuration registers and user OTP are programmed, the FBOOT Configuration register (located at address, 0x801000) must be programmed in order to configure the device in one of the Dual Partition Flash modes. The BTMODE<1:0> bits

cannot be written as '00' (reserved) or as '11' (Single Partition Flash). Single Partition Flash mode must be set by erasing the FBOOT register. See [Table 3-6](#) for details on how to write to the FBOOT Configuration register.

TABLE 3-6: SERIAL INSTRUCTION EXECUTION FOR WRITING THE FBOOT CONFIGURATION REGISTER

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 2: Initialize the TBLPAG register for writing to the latches.		
0000	200FAC	MOV #0xFA, W12
0000	8802AC	MOV W12, TBLPAG
Step 3: Load W0:W1 with the next two Configuration Words to program.		
0000	2xxxx0	MOV #<Config lower word data>, W0
0000	2xxxx1	MOV #<Config upper word data>, W1
Step 4: Set the Write Pointer (W3) and load the write latches.		
0000	EB0030	CLR W6
0000	000000	NOP
0000	BB0B00	TBLWTL W0, [W6]
0000	000000	NOP
0000	000000	NOP
0000	BB9B01	TBLWTH W1, [W6++]
0000	000000	NOP
0000	000000	NOP

dsPIC33EPXXXGS70X/80X FAMILY

TABLE 3-6: SERIAL INSTRUCTION EXECUTION FOR WRITING THE FBOOT CONFIGURATION REGISTER (CONTINUED)

Command (Binary)	Data (Hex)	Description
Step 5: Set the NVMCON register to program FBOOT.		
—	—	; Mask off FBOOT<1:0> = 00 and FBOOT<1:0> = 11 values ; which are a reserved value or the erased default Single ; Partition value, neither of which should be programmed. ; This code clears WREN (NVMCON<14>) so the write will not take ; place.
0000	A31000	BTST.C W0, #1
0000	B08000	ADDC #0, W0
0000	DD004E	SL W0, #14, W0
0000	700068	IOR W0, #0x08, W0
0000	883940	MOV W0, NVMCON
0000	000000	NOP
0000	000000	NOP
Step 6: Initiate the write cycle.		
0000	200551	MOV #0x55, W1
0000	883971	MOV W1, NVMKEY
0000	200AA1	MOV #0xAA, W1
0000	883971	MOV W1, NVMKEY
0000	A8E729	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 7: Wait for program operation to complete and make sure the WR bit is clear.		
—	—	Externally time 'P13' ms (see Section 10.0 “AC/DC Characteristics and Timing Requirements”) to allow sufficient time for the program operation to complete.
0000	000000	NOP
0000	803940	MOV NVMCON, W0
0000	000000	NOP
0000	887C40	MOV W0, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
—	—	Repeat until the WR bit is clear.
Step 8: Reset the device.		

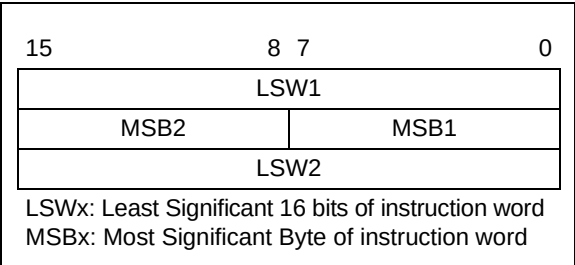
3.7 Writing Code Memory

Figure 3-8 provides a high-level description of the method for writing to code memory.

Two-word writes program the code memory with two instruction words at a time. Two words are loaded into the write latches, located at address, FA0000h, and the destination address must be loaded to the NVMADRU/ NVMADR register pair. Next, the WR bit is set to initiate the write sequence. Then, the WR bit must be checked for the sequence to be complete. This process continues for all the data to be programmed. Table 3-7 shows the ICSP programming details.

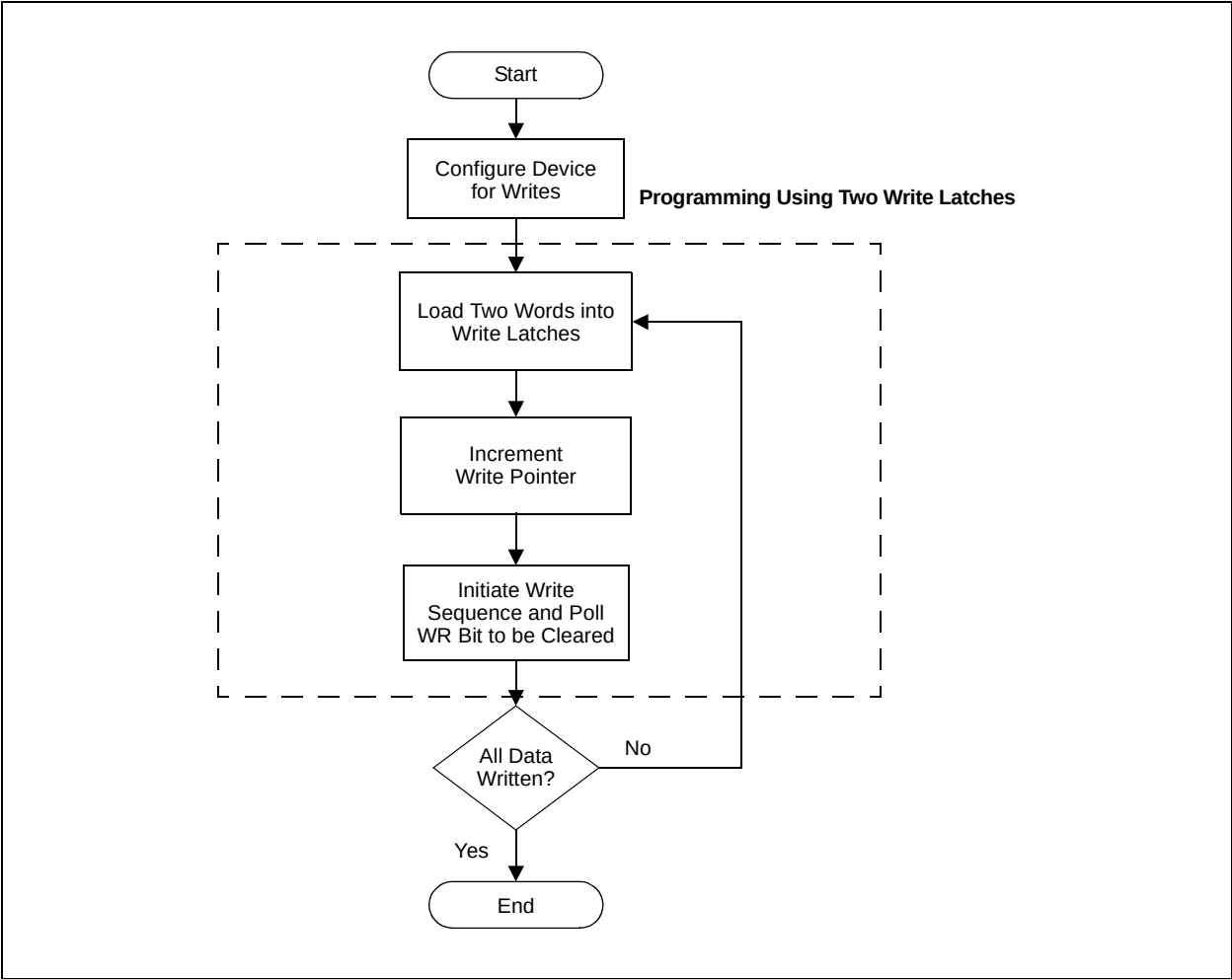
The data loaded into the programming latches must be in the packed format, as shown in Figure 3-7.

FIGURE 3-7: PACKED INSTRUCTION WORD FORMAT



- Note 1:** When the number of instruction words transferred is odd, MSB2 is zero and LSW2 cannot be transmitted.
- 2:** Before reprogramming either word in a double-word pair, the programmer must erase the Flash memory page in which the word is located.

FIGURE 3-8: PROGRAM CODE MEMORY FLOW



dsPIC33EPXXXGS70X/80X FAMILY

**TABLE 3-7: SERIAL INSTRUCTION EXECUTION FOR PROGRAMMING CODE MEMORY:
TWO-WORD LATCH WRITES**

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 2: Initialize the TBLPAG register for writing to the latches.		
0000	200FAC	MOV #0xFA, W12
0000	8802AC	MOV W12, TBLPAG
Step 3: Load W0:W2 with the next two packed instruction words to program.		
0000	2xxxx0	MOV #<LSW0>, W0
0000	2xxxx1	MOV #<MSB1:MSB0>, W1
0000	2xxxx2	MOV #<LSW1>, W2
Step 4: Set the Read Pointer (W6) and Write Pointer (W7), and load the (next set of) write latches.		
0000	EB0300	CLR W6
0000	000000	NOP
0000	EB0380	CLR W7
0000	000000	NOP
0000	BB0BB6	TBLWTL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0000	BBDBB6	TBLWTH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BEBBB6	TBLWTH.B [W6++], [++W7]
0000	000000	NOP
0000	000000	NOP
0000	BB0B96	TBLWTL.W [W6], [W7]
0000	000000	NOP
0000	000000	NOP
Step 5: Set the NVMADRU/NVMADR register pair to point to the correct address.		
0000	2xxxx3	MOV #DestinationAddress<15:0>, W3
0000	2xxxx4	MOV #DestinationAddress<23:16>, W4
0000	883953	MOV W3, NVMADR
0000	883964	MOV W4, NVMADRU
Step 6: Set the NVMCON register to program two instruction words.		
0000	24001A	MOV #0x4001, W10
0000	000000	NOP
0000	88394A	MOV W10, NVMCON
0000	000000	NOP
0000	000000	NOP

dsPIC33EPXXXGS70X/80X FAMILY

**TABLE 3-7: SERIAL INSTRUCTION EXECUTION FOR PROGRAMMING CODE MEMORY:
TWO-WORD LATCH WRITES (CONTINUED)**

Command (Binary)	Data (Hex)	Description
Step 7: Initiate the write cycle.		
0000	200551	MOV #0x55, W1
0000	883971	MOV W1, NVMKEY
0000	200AA1	MOV #0xAA, W1
0000	883971	MOV W1, NVMKEY
0000	A8E729	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 8: Generate clock pulses for program operation to complete until the WR bit is clear.		
0000	000000	NOP
0000	803940	MOV NVMCON, W0
0000	000000	NOP
0000	887C40	MOV W0, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
—	—	Repeat until the WR bit is clear.
Step 9: Repeat Steps 3-8 until all code memory is programmed.		

3.8 Writing Configuration Bits

The procedure for writing Configuration bits is similar to the procedure for writing code memory, except that only two 24-bit words can be programmed at a time.

To change the values of the Configuration bits once they have been programmed, the device must be erased, as described in [Section 3.5 “Erasing Program Memory”](#), and reprogrammed to the desired value. Code protection can be enabled by programming ‘0’ in the code protection Configuration bits.

[Table 3-6](#) shows the ICSP programming details for writing the Configuration bits.

In order to verify the data by reading the Configuration bits after performing the write, the code protection bits should initially be programmed to a ‘1’ to ensure that the verification can be performed properly. After verification is finished, the code protection bits can be programmed to a ‘0’ by using a word write to the appropriate Configuration register.

Note: Since each Configuration register location is followed by an unused memory location, 0xFFFFF can be written to the second word in every Configuration register double-word pair.

TABLE 3-8: SERIAL INSTRUCTION EXECUTION FOR WRITING CONFIGURATION WORDS

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 2: Initialize the TBLPAG register for writing to the latches.		
0000	200FAC	MOV #0xFA, W12
0000	8802AC	MOV W12, TBLPAG
Step 3: Load W0:W1 with the next two Configuration Words to program.		
0000	2xxxx0	MOV #<Config1 lower word data>, W0
0000	2xxxx1	MOV #<Config1 upper word data>, W1
0000	2xxxx2	MOV #<Config2 lower word data>, W2
0000	2xxxx3	MOV #<Config2 upper word data>, W3
Step 4: Set the Write Pointer (W3) and load the write latches.		
0000	EB0300	CLR W6
0000	000000	NOP
0000	BB0B00	TBLWTL W0, [W6]
0000	000000	NOP
0000	000000	NOP
0000	BB9B01	TBLWTH W1, [W6++]
0000	000000	NOP
0000	000000	NOP
0000	BB0B02	TBLWTL W2, [W6]
0000	000000	NOP
0000	000000	NOP
0000	BB9B03	TBLWTH W3, [W6++]
0000	000000	NOP
0000	000000	NOP
Step 5: Set the NVMADRU/NVMADR register pair to point to the correct Configuration Word address.		
0000	2xxxx4	MOV #DestinationAddress<15:0>, W4
0000	2xxxx5	MOV #DestinationAddress<23:16>, W5
0000	883954	MOV W4, NVMADR
0000	883965	MOV W5, NVMADRU

dsPIC33EPXXXGS70X/80X FAMILY

TABLE 3-8: SERIAL INSTRUCTION EXECUTION FOR WRITING CONFIGURATION WORDS (CONTINUED)

Command (Binary)	Data (Hex)	Description
Step 6: Set the NVMCON register to program two instruction words.		
0000	24001A	MOV #0x4001, W10
0000	000000	NOP
0000	88394A	MOV W10, NVMCON
0000	000000	NOP
0000	000000	NOP
Step 7: Initiate the write cycle.		
0000	200551	MOV #0x55, W1
0000	883971	MOV W1, NVMKEY
0000	200AA1	MOV #0xAA, W1
0000	883971	MOV W1, NVMKEY
0000	A8E729	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 8: Generate clock pulses for program operation to complete until the WR bit is clear.		
0000	000000	NOP
0000	803940	MOV NVMCON, W0
0000	000000	NOP
0000	887C40	MOV W0, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
—	—	Repeat until the WR bit is clear.
Step 9: Repeat Steps 3-8 until all Configuration registers are programmed.		

3.9 Writing OTP Words

The procedure for writing to the OTP Words is similar to the procedure for writing code, except that the OTP Words can only be written once. It is not possible to program a '0' to a '1', but the OTP Words may be programmed from a '1' to a '0'. Refer to [Figure 2-8](#), through [Figure 2-11](#) for the locations of the user OTP Words.

OTP memory must be written, one double word at a time. Before writing to any double word in OTP memory, all of its memory locations must be read first. The OTP memory may be programmed only if the entire user OTP memory is in an erased state (i.e., only if every memory location in OTP memory has a value of 0xFFFFF).

3.10 Reading OTP Words

The procedure for reading OTP Words is similar to the procedure for reading code memory. Since there are multiple OTP Words, they are read one at a time.

3.11 Reading Code Memory

Reading from code memory is performed by executing a series of TBLRD instructions and clocking out the data using the REGOUT command.

[Table 3-9](#) shows the ICSP programming details for reading code memory.

To minimize reading time, the same packed data format that the PE uses is utilized. See [Section 6.2 “Programming Executive Commands”](#) for more details on the packed data format.

TABLE 3-9: SERIAL INSTRUCTION EXECUTION FOR READING CODE MEMORY

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 2: Initialize the TBLPAG register and the Read Pointer (W6) for the TBLRD instruction.		
0000	200xx0	MOV #<SourceAddress23:16>, W0
0000	8802A0	MOV W0, TBLPAG
0000	2xxxx6	MOV #<SourceAddress15:0>, W6

Command (Binary)	Data (Hex)	Description
---------------------	---------------	-------------

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	BA1B96	TBLRDL	[W6], [W7++]
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	BADBB6	TBLRDH.B	[W6++], [W7++]
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	BADBD6	TBLRDH.B	[++W6], [W7++]
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	BA1BB6	TBLRDL	[W6++], [W7++]
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	BA1B96	TBLRDL	[W6], [W7++]
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	BADBB6	TBLRDH.B	[W6++], [W7++]
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	BADBD6	TBLRDH.B	[++W6], [W7++]
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	BA0BB6	TBLRDL	[W6++], [W7]
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	
0000	000000	NOP	

dsPIC33EPXXXGS70X/80X FAMILY

TABLE 3-9: SERIAL INSTRUCTION EXECUTION FOR READING CODE MEMORY (CONTINUED)

Command (Binary)	Data (Hex)	Description
Step 4: Output W0:W5 using the VISI register and REGOUT command.		
0000	887C40	MOV W0, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	887C41	MOV W1, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	887C42	MOV W2, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	887C43	MOV W3, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	887C44	MOV W4, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	887C45	MOV W5, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
Step 5: Reset the device's internal PC.		
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 6: Repeat Steps 3-5 until all desired code memory is read.		

dsPIC33EPXXXGS70X/80X FAMILY

3.12 Reading Configuration Registers

The procedure for reading Configuration bits is similar to the procedure for reading code memory. Since there are multiple Configuration Words, they are read one at a time.

Table 3-10 shows the ICSP programming details for reading the Configuration bits.

TABLE 3-10: SERIAL INSTRUCTION EXECUTION FOR READING CONFIGURATION WORDS

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 2: Initialize the TBLPAG register, the Write Pointer (W7) and the Read Pointer (W6) for the TBLRD instruction.		
0000	200xx0	MOV #<Address23:16>, w0 ⁽¹⁾
0000	20F887	MOV #<VISI>, W7
0000	8802A0	MOV W0, TBLPAG
0000	2xxxx6	MOV #<Address15:0>, w6 ⁽¹⁾
Step 3: Store the Configuration register and send the contents of the VISI register.		
0000	000000	NOP
0000	BA8B96	TBLRDH [w6], [w7]
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	BA0B96	TBLRDL [w6], [w7]
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
Step 4: Repeat Steps 1-3 until all Configuration registers are read.		

Note 1: The FBOOT register must be read before reading any other Configuration Word. Based on the device's current Partition Flash mode, the Configuration Word addresses must be determined by the programmer.

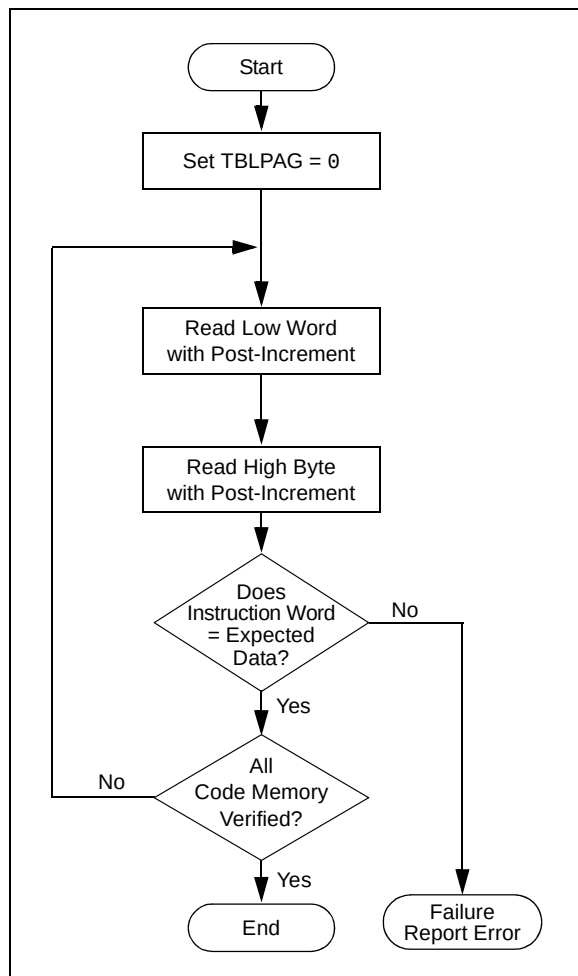
3.13 Verify Code Memory and Configuration Words

The verify step involves reading back the code memory space and comparing it against the copy held in the programmer's buffer. The Configuration Words are verified with the rest of the code.

The verify process is illustrated in Figure 3-9. The lower word of the instruction is read, and then the lower byte of the upper word is read and compared against the instruction stored in the programmer's buffer. Refer to Section 3.11 "Reading Code Memory" for implementation details of reading code memory.

Note: Because the Configuration Words include the device code protection bit, code memory should be verified immediately after writing if the code protection is to be enabled. This is because the device will not be readable or verifiable if a device Reset occurs after the code-protect bit has been cleared.

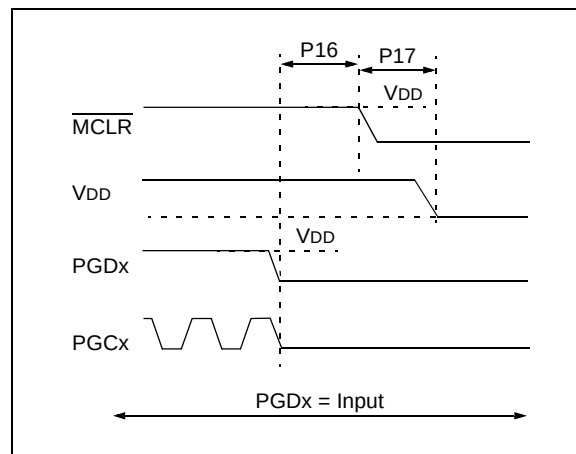
FIGURE 3-9: VERIFY CODE MEMORY FLOW



3.14 Exiting ICSP Mode

Exiting Program/Verify mode is done by removing VDD from MCLR, as illustrated in Figure 3-10. The only requirement for exit is that an interval, P16, should elapse between the last clock, and program signals on PGDx and PGDx, before removing VDD.

FIGURE 3-10: EXITING ICSP™ MODE



dsPIC33EPXXXGS70X/80X FAMILY

4.0 DEVICE PROGRAMMING – ENHANCED ICSP

This section discusses programming the device through Enhanced ICSP and the Programming Executive (PE). The PE resides in executive memory (separate from code memory) and is executed when Enhanced ICSP Programming mode is entered. The PE provides the mechanism for the programmer (host device) to program and verify the dsPIC33EPXXXGS70X/80X devices, using a simple command set and communication protocol. There are several basic functions provided by the PE:

- Read Memory
- Erase Memory
- Program Memory
- Blank Check

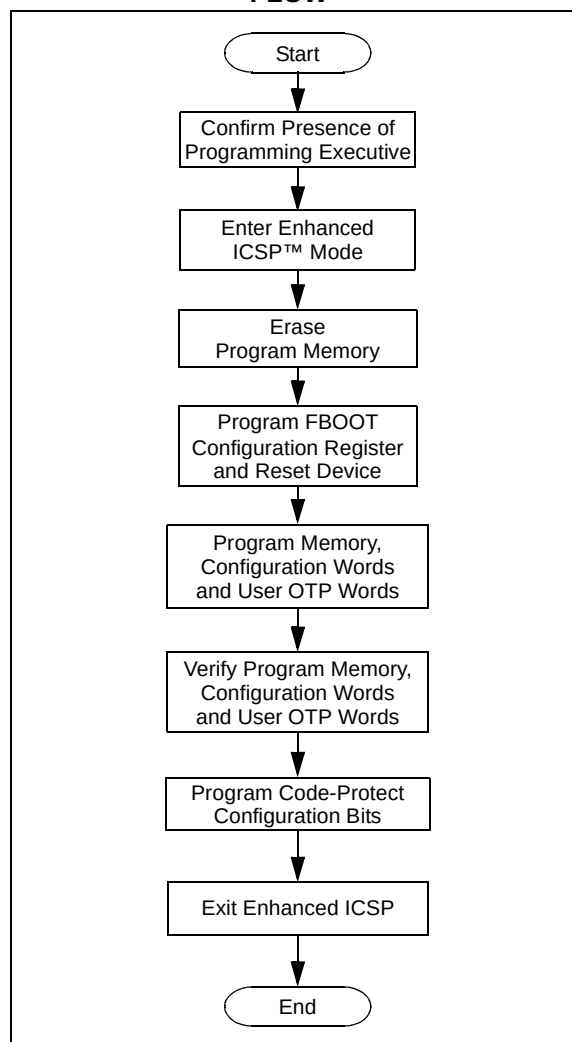
The PE performs the low-level tasks required for erasing, programming and verifying a device. This allows the programmer to program the device by issuing the appropriate commands and data. A detailed description for each command is provided in [Section 6.2 “Programming Executive Commands”](#).

Note: The PE uses the device's data RAM for variable storage and program execution. After running the PE, no assumptions should be made about the contents of data RAM.

4.1 Overview of the Programming Process

[Figure 4-1](#) shows the high-level overview of the programming process. First, it must be determined if the PE is present in executive memory, and then, Enhanced ICSP mode is entered. The program memory is then erased, and the program memory and Configuration Words are programmed and verified. Last, the code-protect Configuration bits are programmed (if required) and Enhanced ICSP mode is exited.

FIGURE 4-1: HIGH-LEVEL ENHANCED ICSP™ PROGRAMMING FLOW



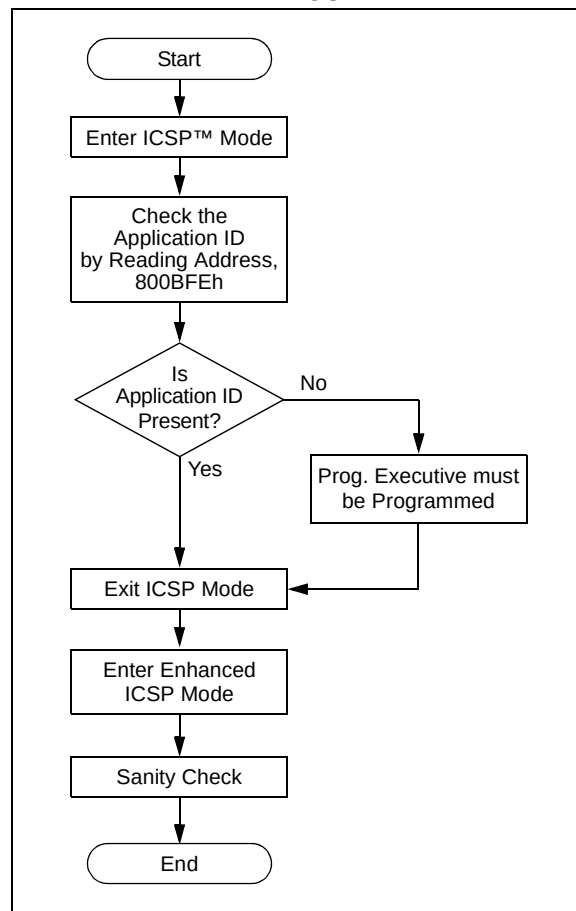
4.2 Confirming the Presence of the Programming Executive

Before programming, the programmer must confirm that the PE is stored in executive memory. The procedure for this task is illustrated in [Figure 4-2](#).

First, ICSP mode is entered. Then, the unique Application ID Word, stored in executive memory, is read. If the PE is resident, the correct Application ID Word, 0xDF, is read and programming can resume as normal. However, if the Application ID Word is not present, the PE must be programmed to executive code memory using the method described in [Section 5.0 “Programming the Programming Executive to Memory”](#).

[Section 3.0 “Device Programming – ICSP”](#) describes the ICSP programming method. [Section 4.3 “Reading the Application ID Word”](#) describes the procedure for reading the Application ID Word in ICSP mode.

FIGURE 4-2: CONFIRMING PRESENCE OF PROGRAMMING EXECUTIVE



dsPIC33EPXXXGS70X/80X FAMILY

4.3 Reading the Application ID Word

The Application ID Word is stored at address, 800BFEh, 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. Then, the REGOUT control code must 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 4-1](#).

After the programmer has clocked out the Application ID Word, it must be inspected. If the Application ID has the value, 0xDF, the PE is resident in memory and the device can be programmed using the mechanism described in [Section 4.0 “Device Programming – Enhanced ICSP”](#). However, if the Application ID has any other value, the PE is not resident in memory; it must be loaded to memory before the device can be programmed. The procedure for loading the PE to memory is described in [Section 5.0 “Programming the Programming Executive to Memory”](#).

TABLE 4-1: SERIAL INSTRUCTION EXECUTION FOR READING THE APPLICATION ID WORD

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 2: Initialize the TBLPAG register and the Read Pointer (W0) for the TBLRD instruction.		
0000	200800	MOV #0x80, W0
0000	8802A0	MOV W0, TBLPAG
0000	20BFE0	MOV #0xBFE, W0
0000	20F881	MOV #VISI, W1
0000	000000	NOP
0000	BA0890	TBLRDL [W0], [W1]
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 3: Output the VISI register using the REGOUT command.		
0001	<VISI>	Clock out the contents of the VISI register.

4.4 Entering Enhanced ICSP Mode

As illustrated in Figure 4-3, entering Enhanced ICSP Program/Verify mode requires three steps:

1. The $\overline{\text{MCLR}}$ pin is briefly driven high and then low.
2. A 32-bit key sequence is clocked into PGDx.
3. $\overline{\text{MCLR}}$ is then driven high within a specified period of time and held.

The key sequence is a specific 32-bit pattern, '0100 1101 0100 0011 0100 1000 0101 0000' (more easily remembered as 4D434850h in hexadecimal format). The device will enter Program/Verify mode only if the key sequence is valid. The Most Significant bit (MSb) of the most significant nibble must be shifted in first.

Once the key sequence is complete, VDD must be applied to MCLR and held at that level for as long as Program/Verify mode is to be maintained. An interval time of at least P19, P7 and P1 * 5, must elapse before presenting data on PGDx. Signals appearing on PGDx before P7 has elapsed will not be interpreted as valid.

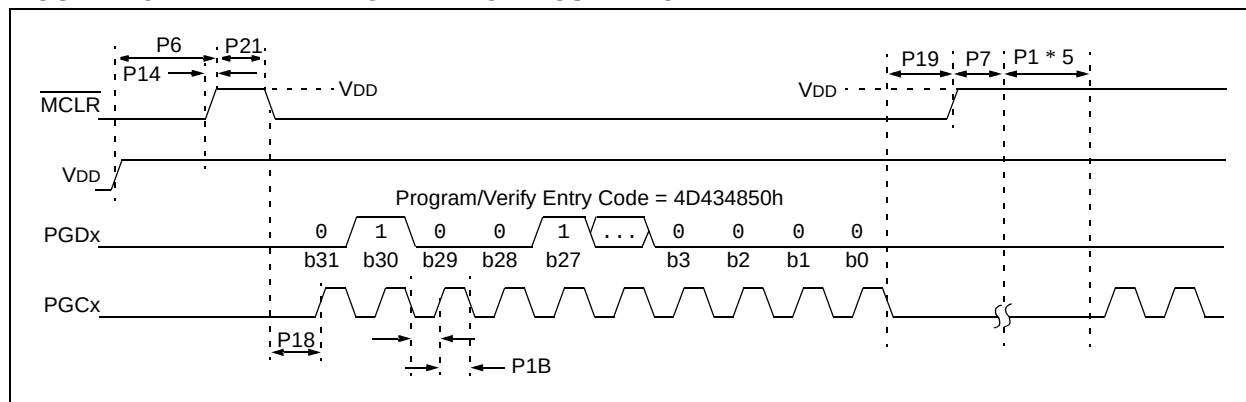
4.5 Blank Check

The term, "Blank Check", implies verifying that the device has been successfully erased and has no programmed memory locations. A blank or erased memory location is always read as '1'.

The Device ID registers (FF0000h:FF0002h) can be ignored by the Blank Check, since this region stores device information that cannot be erased. Additionally, all unimplemented memory space and Calibration registers should be ignored from the Blank Check.

The QBLANK command is used for the Blank Check. It determines if the code memory is erased by testing these memory regions. A 'BLANK' or 'NOT BLANK' response is returned. If it is determined that the device is not blank, it must be erased before attempting to program the chip.

FIGURE 4-3: ENTERING ENHANCED ICSP™ MODE



4.6 Code Memory Programming

4.6.1 PROGRAMMING METHODOLOGY

There are two commands that can be used for programming code memory when utilizing the PE. The PROG2W command programs and verifies two 24-bit instruction words into the program memory, starting at the address specified. The second and faster command, PROGP, allows up to 128 24-bit instruction words to be programmed and verified into program memory, starting at the address specified. See [Section 6.0 “The Programming Executive”](#) for a full description for each of these commands.

Figure 4-4 and Figure 4-5 show the programming methodology for the PROG2W and PROGP commands. In both instances, 22464 instruction words of the dsPIC33EPXXXGS70X/80X device are programmed.

Note: When programming a bootloader, it is advisable to program the bootloader code into the second page of the code memory (e.g., at 400h). A bootloader, located in one of the beginning addresses (e.g., at 200h) while attempting to erase the first page, might erase itself.

FIGURE 4-4: FLOWCHART FOR DOUBLE-WORD PROGRAMMING

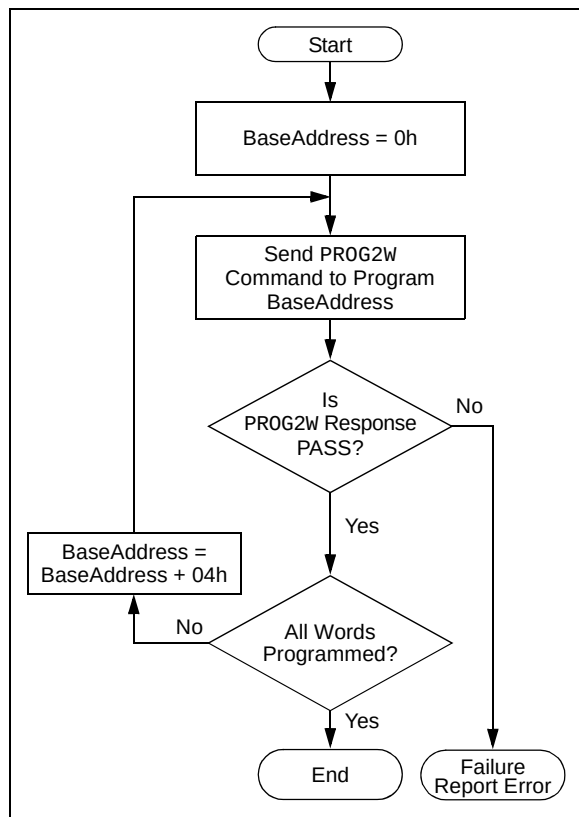
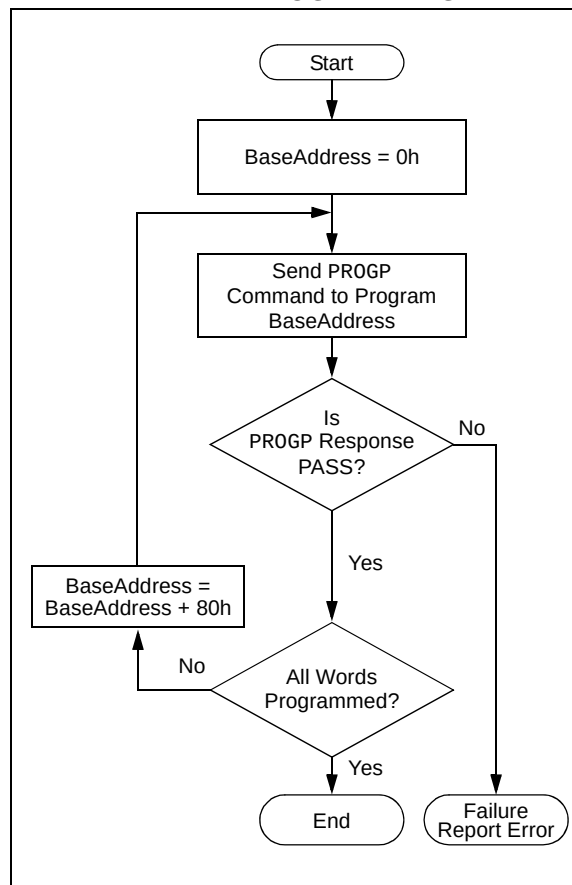


FIGURE 4-5: FLOWCHART FOR MULTIPLE WORD PROGRAMMING

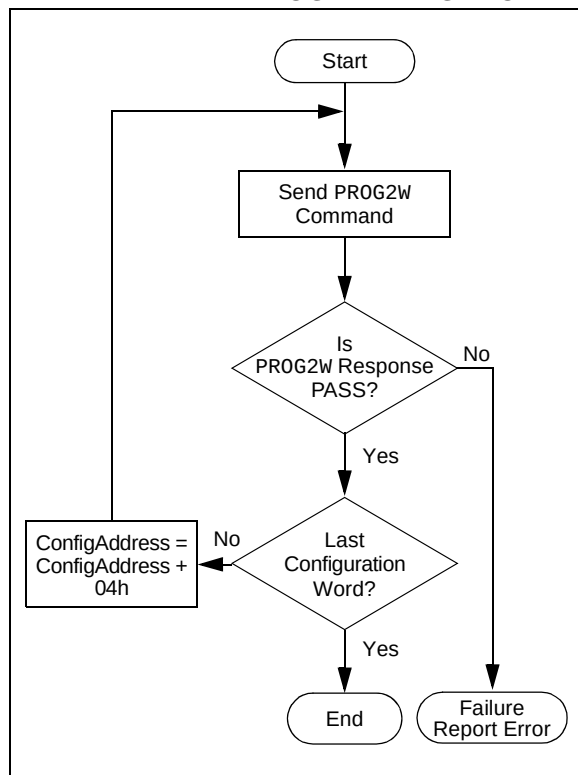


4.7 Configuration Bit Programming

Configuration bits are programmed one at a time using the PROG2W command. This command specifies the configuration data and address. When Configuration bits are programmed, any unimplemented bits must be programmed with a '1'.

Multiple PROG2W commands are required to program all Configuration bits. A flowchart for Configuration bit programming is shown in [Figure 4-6](#).

FIGURE 4-6: CONFIGURATION BIT PROGRAMMING FLOW



4.8 Programming Verification

After 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 and Configuration Words.

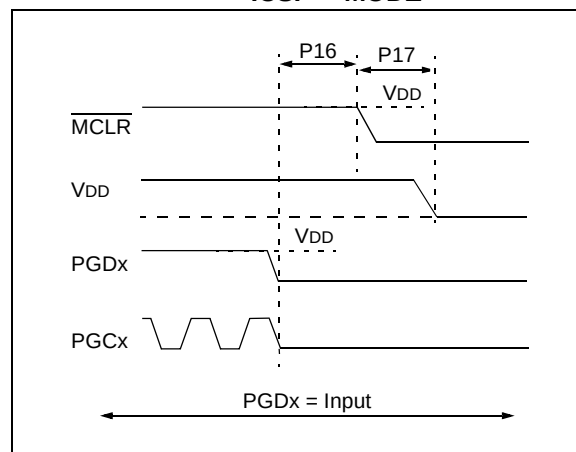
Alternatively, you can have the programmer perform the verification after the entire device is programmed using a checksum computation.

See [Section 9.0 "Checksum Computation"](#) for more information on calculating the checksum.

4.9 Exiting Enhanced ICSP Mode

Exiting Program/Verify mode is done by removing VDD from MCLR, as illustrated in [Figure 4-7](#). The only requirement for exit is that an interval, P16, should elapse between the last clock, and program signals on PGCx and PGDx before removing VDD.

FIGURE 4-7: EXITING ENHANCED ICSP™ MODE



5.0 PROGRAMMING THE PROGRAMMING EXECUTIVE TO MEMORY

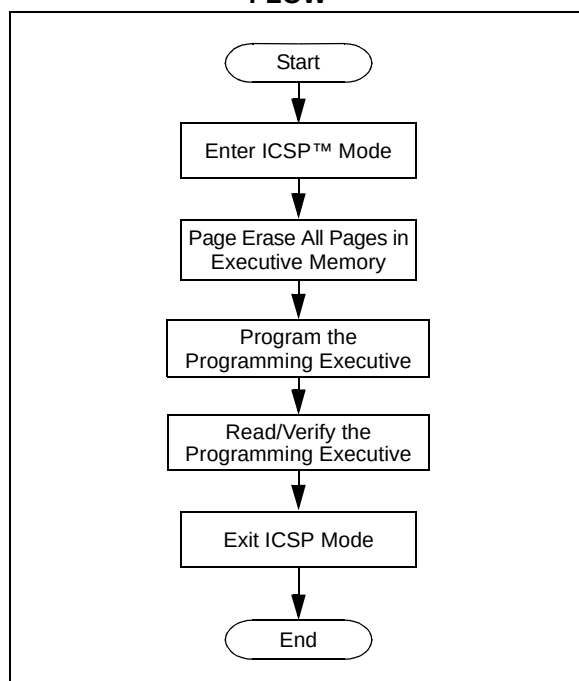
Note: The Programming Executive (PE) can be obtained from each device page on the Microchip web site: www.microchip.com.

5.1 Overview

If it is determined that the PE is not present in executive memory (as described in [Section 4.2 “Confirming the Presence of the Programming Executive”](#)), the PE must be programmed to executive memory.

[Figure 5-1](#) shows the high-level process of programming the PE into executive memory. First, ICSP mode must be entered, and executive memory and user memory are erased. Then, the PE is programmed and verified. Finally, ICSP mode is exited.

FIGURE 5-1: HIGH-LEVEL PROGRAMMING EXECUTIVE PROGRAM FLOW



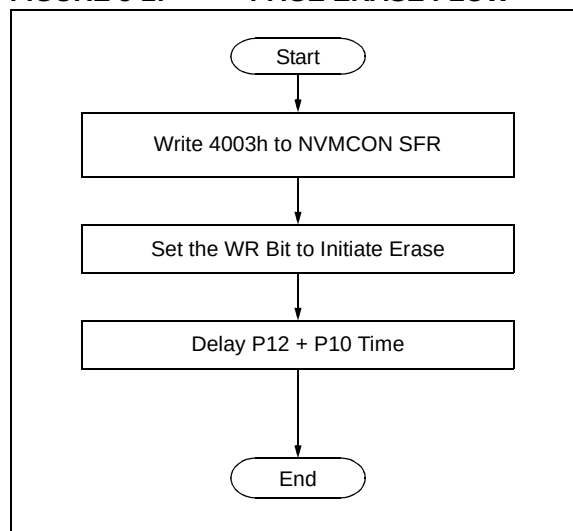
5.2 Erasing Executive Memory

The procedure for erasing each page of executive memory is similar to that of erasing program memory and is shown in [Figure 5-2](#). It consists of setting NVMCON to 4003h and then executing the programming cycle. Note that program memory is also erased with this operation.

[Table 5-1](#) illustrates the ICSP programming process for Bulk Erasing memory.

Note: The PE must always be erased before it is programmed, as described in [Figure 5-1](#).

FIGURE 5-2: PAGE ERASE FLOW



dsPIC33EPXXXGS70X/80X FAMILY

TABLE 5-1: SERIAL INSTRUCTION EXECUTION FOR ERASING ALL PAGES OF EXECUTIVE MEMORY

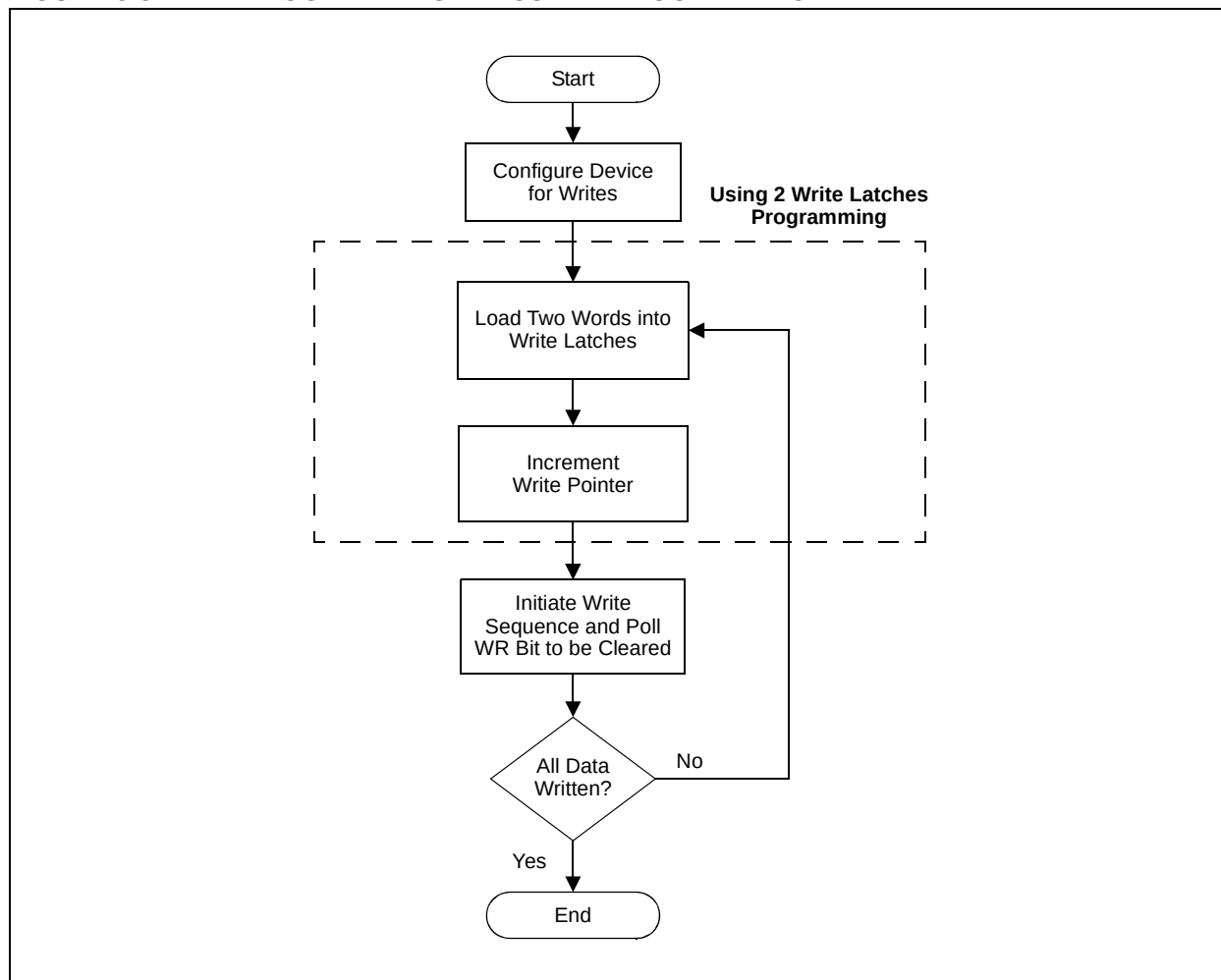
Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 2: Set the NVMADRU/NVMADR register pair to point to the correct page of executive memory to be erased.		
0000	2xxxx3	MOV #DestinationAddress<15:0>, W3
0000	2xxxx4	MOV #DestinationAddress<23:16>, W4
0000	883953	MOV W3, NVMADR
0000	883964	MOV W4, NVMADRU
Step 3: Set the NVMCON register to erase the first page of executive memory.		
0000	24003A	MOV #0x4003, W10
0000	88394A	MOV W10, NVMCON
0000	000000	NOP
0000	000000	NOP
Step 4: Initiate the erase cycle.		
0000	200551	MOV #0x55, W1
0000	883971	MOV W1, NVMKEY
0000	200AA1	MOV #0xAA, W1
0000	883971	MOV W1, NVMKEY
0000	A8E729	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 5: Generate clock pulses for the Page Erase operation to complete until the WR bit is clear.		
0000	000000	NOP
0000	803940	MOV NVMCON, W0
0000	000000	NOP
0000	887C40	MOV W0, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
—	—	Repeat until the WR bit is clear.
Step 6: Repeat Steps 2-5 for all pages of executive memory.		

5.3 Program the Programming Executive

Storing the PE to executive memory is similar to normal programming of code memory. The executive memory must first be erased and then programmed using two-word writes (two instruction words). The control flow for this method is summarized in [Figure 5-3](#).

[Table 5-2](#) illustrates the ICSP programming processes for PE memory. To minimize programming time, the same packed data format that the PE uses is utilized. See [Section 6.2 “Programming Executive Commands”](#) for more details on the packed data format.

FIGURE 5-3: PROGRAMMING EXECUTIVE PROGRAM FLOW



**TABLE 5-2: PROGRAMMING THE PROGRAMMING EXECUTIVE
(TWO-WORD LATCH WRITES)**

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 2: Initialize the TBLPAG register for writing to the latches.		
0000	200FAC	MOV #0xFA, W12
0000	8802AC	MOV W12, TBLPAG
Step 3: Load W0:W2 with the next two packed instruction words to program.		
0000	2xxxx0	MOV #<LSW0>, W0
0000	2xxxx1	MOV #<MSB1:MSB0>, W1
0000	2xxxx2	MOV #<LSW1>, W2
Step 4: Set the Read Pointer (W6) and the Write Pointer (W7), and load the write latches.		
0000	EB0300	CLR W6
0000	000000	NOP
0000	EB0380	CLR W7
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	BB0B96	TBLWTL.W [W6], [W7]
0000	000000	NOP
0000	000000	NOP
Step 5: Set the NVMADRU/NVMADR register pair to point to the correct row.		
0000	2xxxx3	MOV #DestinationAddress<15:0>, W3
0000	2xxxx4	MOV #DestinationAddress<23:16>, W4
0000	883953	MOV W3, NVMADR
0000	883964	MOV W4, NVMADRU
Step 6: Set the NVMCON register to program two instruction words.		
0000	24001A	MOV #0x4001, W10
0000	000000	NOP
0000	88394A	MOV W10, NVMCON
0000	000000	NOP
0000	000000	NOP

dsPIC33EPXXXGS70X/80X FAMILY

**TABLE 5-2: PROGRAMMING THE PROGRAMMING EXECUTIVE
(TWO-WORD LATCH WRITES) (CONTINUED)**

Command (Binary)	Data (Hex)	Description
Step 7: Initiate the write cycle.		
0000	200551	MOV #0x55, W1
0000	883971	MOV W1, NVMKEY
0000	200AA1	MOV #0xAA, W1
0000	883971	MOV W1, NVMKEY
0000	A8E729	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 8: Generate clock pulses for program operation to complete until the WR bit is clear.		
0000	000000	NOP
0000	803940	MOV NVMCON, W0
0000	000000	NOP
0000	887C40	MOV W0, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
—	—	Repeat until the WR bit is clear.
Step 9: Repeat Steps 3-8 until all code memory is programmed.		

5.4 Reading Executive Memory

Reading from executive memory is performed by executing a series of TBLRD instructions and clocking out the data using the REGOUT command.

Table 5-3 shows the ICSP programming details for reading executive memory.

To minimize reading time, the same packed data format that the PE uses is utilized. See [Section 6.2 “Programming Executive Commands”](#) for more details on the packed data format.

TABLE 5-3: SERIAL INSTRUCTION EXECUTION FOR READING CODE MEMORY

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 2: Initialize the TBLPAG register and the Read Pointer (W6) for the TBLRD instruction.		
0000	200xx0	MOV #<SourceAddress23:16>, W0
0000	8802A0	MOV W0, TBLPAG
0000	2xxxx6	MOV #<SourceAddress15:0>, W6

dsPIC33EPXXXGS70X/80X FAMILY

TABLE 5-3: SERIAL INSTRUCTION EXECUTION FOR READING CODE MEMORY (CONTINUED)

Command (Binary)	Data (Hex)	Description
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	BA1B96	TBLRDL [W6], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	BADBB6	TBLRDH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	BADBD6	TBLRDH.B [++W6], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	BA1BB6	TBLRDL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	BA1B96	TBLRDL [W6], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	BADBB6	TBLRDH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	BADBD6	TBLRDH.B [++W6], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	BA0BB6	TBLRDL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP

dsPIC33EPXXXGS70X/80X FAMILY

TABLE 5-3: SERIAL INSTRUCTION EXECUTION FOR READING CODE MEMORY (CONTINUED)

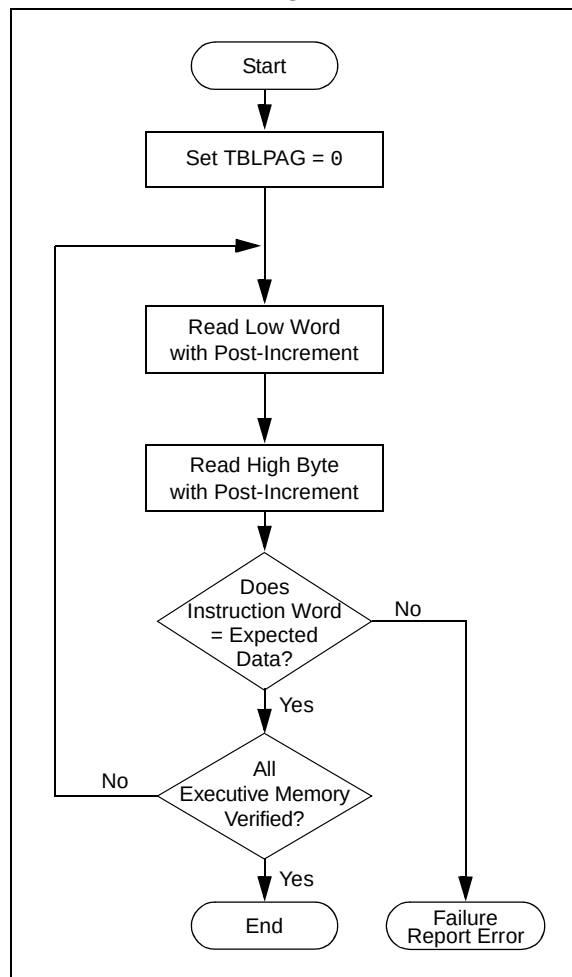
Command (Binary)	Data (Hex)	Description
Step 4: Output W0:W5 using the VISI register and REGOUT command.		
0000	887C40	MOV W0, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	887C41	MOV W1, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	887C42	MOV W2, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	887C43	MOV W3, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	887C44	MOV W4, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
0000	887C45	MOV W5, VISI
0000	000000	NOP
0001	<VISI>	Clock out the contents of the VISI register.
0000	000000	NOP
Step 5: Reset the device's internal PC.		
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	000000	NOP
0000	000000	NOP
Step 6: Repeat Steps 3-5 until all desired code memory is read.		

5.5 Verify Programming Executive

The verify step involves reading back the executive memory space and comparing it against the copy held in the programmer's buffer.

The verify process is illustrated in [Figure 5-4](#). The lower word of the instruction is read, and then the lower byte of the upper word is read and compared against the instruction stored in the programmer's buffer. Refer to [Section 5.4 "Reading Executive Memory"](#) for implementation details of reading executive memory.

FIGURE 5-4: VERIFY PROGRAMMING EXECUTIVE MEMORY FLOW



6.0 THE PROGRAMMING EXECUTIVE

Note: The Programming Executive (PE) can be obtained from each device page on the Microchip web site: www.microchip.com.

6.1 Programming Executive Communication

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

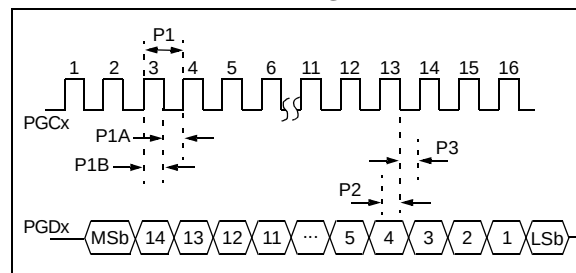
All communication is initiated by the programmer in the form of a command. Only one command at a time can be sent to the PE. In turn, the PE only sends one response to the programmer after receiving and processing a command. The PE command set is described in [Section 6.2 “Programming Executive Commands”](#). The response set is described in [Section 6.3 “Programming Executive Responses”](#).

6.1.1 COMMUNICATION INTERFACE AND PROTOCOL

The ICSP/Enhanced ICSP interface is a 2-wire SPI, implemented using the PGCx and PGDx pins. The PGCx pin is used as a clock input pin and the clock source must be provided by the programmer. The PGDx pin is used for sending command data to, and receiving response data from, the PE.

Note: For Enhanced ICSP, all serial data is transmitted on the falling edge of PGCx and latched on the rising edge of PGCx. All data transmissions are sent MSb first, using 16-bit mode (see [Figure 6-1](#)).

FIGURE 6-1: PROGRAMMING EXECUTIVE SERIAL TIMING



Since a 2-wire SPI is used, and data transmissions are bidirectional, a simple protocol is used to control the direction of PGDx. When the programmer completes a command transmission, it releases the PGDx line and allows the PE to drive this line high. The PE keeps the PGDx line high to indicate that it is processing the command.

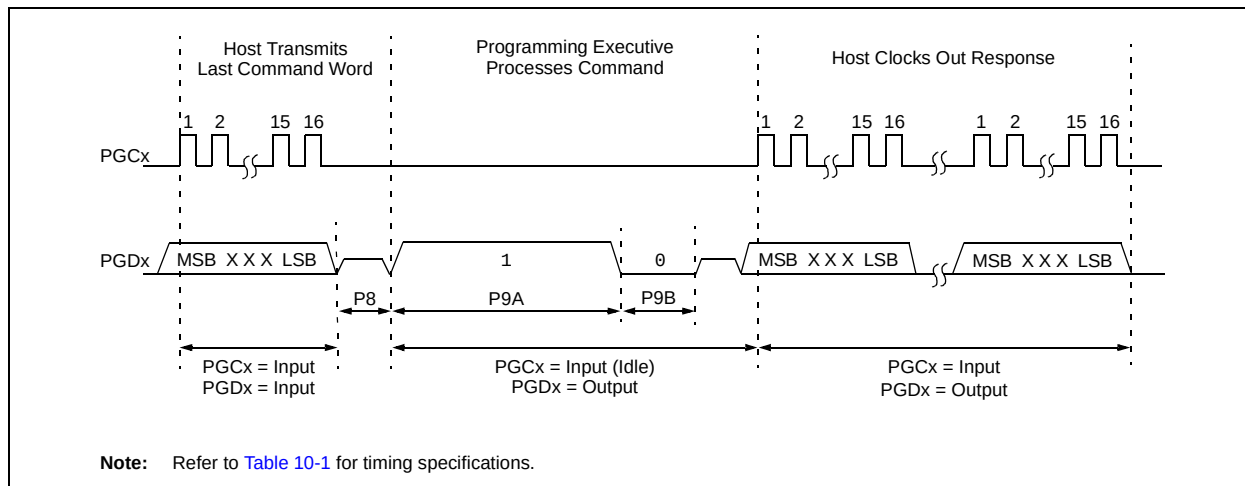
After the PE has processed the command, it brings PGDx low (P9B) to indicate to the programmer that the response is available to be clocked out. The programmer can begin to clock out the response after a maximum wait (P9B) and it must provide the necessary amount of clock pulses to receive the entire response from the PE.

After the entire response is clocked out, the programmer should terminate the clock on PGCx until it is time to send another command to the PE. This protocol is illustrated in [Figure 6-2](#).

6.1.2 SPI RATE

In Enhanced ICSP mode, the dsPIC33EPXXXGS70X/80X devices operate from the internal Fast RC (FRC) oscillator, which has a nominal frequency of 7.3728 MHz. This oscillator frequency yields an effective system clock frequency of 3.6864 MHz. To ensure that the programmer does not clock too fast, it is recommended that a 1.8432 MHz clock be provided by the programmer.

FIGURE 6-2: PROGRAMMING EXECUTIVE — PROGRAMMER COMMUNICATION PROTOCOL



dsPIC33EPXXXGS70X/80X FAMILY

6.1.3 TIME-OUTS

The PE uses no Watchdog Timer or time-out for transmitting responses to the programmer. If the programmer does not follow the flow control mechanism using PGCx as described in [Section 6.1.1 “Communication Interface and Protocol”](#), it is possible that the PE will behave unexpectedly while trying to send a response to the programmer. Since the PE 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 6-1](#). If the command time-out expires, the programmer should reset the PE and start programming the device again.

TABLE 6-1: PROGRAMMING EXECUTIVE COMMAND SET

Opcode	Mnemonic	Length (bytes)	Time-out	Description
0x0	SCHECK	1	1 ms	Sanity check.
0x1	Reserved	N/A	N/A	This command is reserved; it will return a NACK.
0x2	READP	4	1 ms/row	Read 'N' 24-bit instruction words of primary Flash memory, starting from the specified address.
0x3	PROG2W	8	5 ms	Program a double instruction word of code memory at the specified address and verify.
0x4	Reserved	N/A	N/A	This command is reserved; it will return a NACK.
0x5	PROGP	195	125 ms	Program 128 words of program memory at the specified starting address, then verify.
0x6	Reserved	N/A	N/A	This command is reserved; it will return a NACK.
0x7	ERASEB	1	125 ms	Bulk Erase user memory.
0x8	Reserved	N/A	N/A	This command is reserved; it will return a NACK.
0x9	ERASEP	3	25 ms	Command to erase a page.
0xA	Reserved	N/A	N/A	This command is reserved; it will return a NACK.
0xB	QVER	1	1 ms	Query the PE software version.
0xC	CRCP	5	1.5s	Perform a CRC-16 on the specified range of memory.
0xD	Reserved	N/A	N/A	This command is reserved; it will return a NACK.
0xE	QBLANK	5	1s	Query to check whether the code memory is blank.

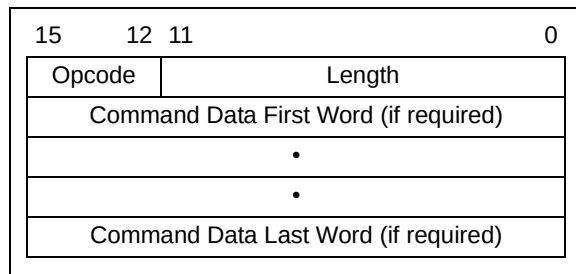
6.2 Programming Executive Commands

The PE command set is shown in [Table 6-1](#). This table contains the opcode, mnemonic, length, time-out and description for each command. Functional details on each command are provided in the command descriptions (see [Section 6.2.4 “Command Descriptions”](#)).

6.2.1 COMMAND FORMAT

All PE commands have a general format, consisting of a 16-bit header and any required data for the command (see [Figure 6-3](#)). The 16-bit header consists of a 4-bit opcode field, which is used to identify the command, followed by a 12-bit command length field.

FIGURE 6-3: COMMAND FORMAT



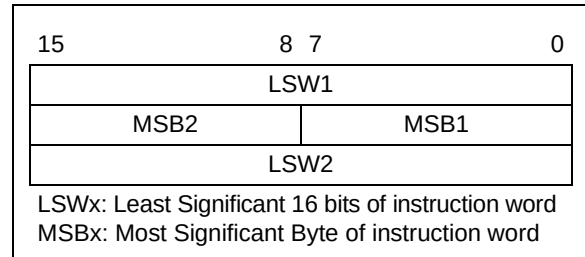
The command opcode must match one of those in the command set. Any command that is received, which does not match the list in [Table 6-1](#), will return a “NACK” response (see [Section 6.3.1.1 “Opcode Field”](#)).

The command length is represented in 16-bit words since the SPI operates in 16-bit mode. The PE uses the command length field to determine the number of words to read from the SPI port. If the value of this field is incorrect, the command will not be properly received by the PE.

6.2.2 PACKED DATA FORMAT

When 24-bit instruction words are transferred across the 16-bit SPI interface, they are packed to conserve space using the format illustrated in [Figure 6-4](#). This format minimizes traffic over the SPI and provides the PE with data that is properly aligned for performing Table Write operations.

FIGURE 6-4: PACKED INSTRUCTION WORD FORMAT



Note: When the number of instruction words transferred is odd, MSB2 is zero and LSW2 cannot be transmitted.

6.2.3 PROGRAMMING EXECUTIVE ERROR HANDLING

The PE will “NACK” all unsupported commands. Additionally, due to the memory constraints of the PE, no checking is performed on the data contained in the programmer command. It is the responsibility of the programmer to command the PE with valid command arguments or the programming operation may fail. Additional information on error handling is provided in [Section 6.3.1.3 “QE_Code Field”](#).

dsPIC33EPXXXGS70X/80X FAMILY

6.2.4 COMMAND DESCRIPTIONS

All commands supported by the PE are described in [Section 6.2.4.1 “SCHECK Command”](#) through [Section 6.2.4.10 “QBLANK Command”](#).

6.2.4.1 SCHECK Command

15	12	11	0
Opcode	Length		

[Table 6-2](#) shows the description for the SCHECK command.

TABLE 6-2: COMMAND DESCRIPTION

Field	Description
Opcode	0x0
Length	0x1

The SCHECK command instructs the PE to do nothing but generate a response. This command is used as a “Sanity Check” to verify that the PE is operational.

Expected Response (2 words):

0x10000
0x0002

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

6.2.4.2 READC Command

15	12	11	8	7	0
Opcode		Length			
N			Addr_MSB		
Addr_LS					

[Table 6-3](#) shows the description for the READC command.

TABLE 6-3: COMMAND DESCRIPTION

Field	Description
Opcode	0x1
Length	0x3
N	Number of 8-bit Configuration registers or Device ID registers to read (maximum of 256)
Addr_MSB	MSB of 24-bit source address
Addr_LS	Least Significant 16 bits of 24-bit source address

The READC command instructs the PE to read N Configuration registers or Device ID registers, starting from the 24-bit address specified by Addr_MSB and Addr_LS. This command can only be used to read 8-bit or 16-bit data.

When this command is used to read Configuration registers, the upper byte in every data word returned by the PE is 0x00 and the lower byte contains the Configuration register value.

Expected Response ($4 + 3 * (N - 1)/2$ words for N odd):

0x1100
 $2 + N$
Configuration register or Device ID Register 1
...
Configuration register or Device ID Register N

Note: Reading unimplemented memory will cause the PE to reset. To prevent this from occurring, ensure that only memory locations present on a particular device are accessed.

6.2.4.3 READP Command

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

Table 6-4 shows the description for the READP command.

TABLE 6-4: COMMAND DESCRIPTION

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

The READP command instructs the PE 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 the response to this command uses the packed data format described in [Section 6.2.2 “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 PE to reset. To prevent this from occurring, ensure that only memory locations present on a particular device are accessed.

6.2.4.4 PROG2W Command

15	12	11	8	7	0
Opcode		Length			
Reserved			Addr_MSB		
Addr_LS					
DataL_LS					
DataH_MSB			DataL_MSB		
DataH_LS					

Table 6-5 shows the description for the PROG2W command.

TABLE 6-5: COMMAND DESCRIPTION

Field	Description
Opcode	0x3
Length	0x6
DataL_MSB	MSB of 24-bit data for low instruction word
DataH_MSB	MSB of 24-bit data for high instruction word
Addr_MSB	MSB of 24-bit destination address
Addr_LS	Least Significant 16 bits of 24-bit destination address
DataL_LS	Least Significant 16 bits of 24-bit data for low instruction word
DataH_LS	Least Significant 16 bits of 24-bit data for high instruction word

The PROG2W command instructs the PE to program two instruction words of code memory (6 bytes) to the specified memory address.

After the words have been programmed to code memory, the PE verifies the programmed data against the data in the command.

Expected Response (2 words):

0x1300
 0x0002

dsPIC33EPXXXGS70X/80X FAMILY

6.2.4.5 PROGP Command

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

Table 6-6 shows the description for the PROGP command.

TABLE 6-6: COMMAND DESCRIPTION

Field	Description
Opcode	0x5
Length	0xC3
Reserved	0x0
Addr_MSB	MSB of 24-bit destination address
Addr_LS	Least Significant 16 bits of 24-bit destination address
D_1	16-bit Data Word 1
D_2	16-bit Data Word 2
...	16-bit Data Word 3 through 95
D_96	16-bit Data Word 96

The PROGP command instructs the PE to program one row of code memory (128 instruction words) to the specified memory address. Programming begins with the row address specified in the command. The destination address should be a multiple of 0x80.

The data to program the memory, located in command words, D_1 through D_96, must be arranged using the packed instruction word format illustrated in Figure 6-4.

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

Expected Response (2 words):

0x1500
0x0002

Note: Refer to Table 2-2 for code memory size information.

6.2.4.6 ERASEB Command

15	12	11	8	7	0
Opcode		Length			

Table 6-7 shows the description for the ERASEB command.

TABLE 6-7: COMMAND DESCRIPTION

Field	Description
Opcode	0x7
Length	0x1

The ERASEB command instructs the PE to perform a Bulk Erase of the user Flash memory.

Expected Response (2 words):

0x1700
0x0002

6.2.4.7 ERASEP Command

15	12	11	8	7	0
Opcode		Length			
NUM_PAGES			Addr_MSB		
Addr_LS					

Table 6-6 shows the description for the ERASEP command.

TABLE 6-8: COMMAND DESCRIPTION

Field	Description
Opcode	0x9
Length	0x3
NUM_PAGES	Up to 255
Addr_MSB	Most Significant Byte of the 24-bit address
Addr_LS	Least Significant 16 bits of the 24-bit address

The ERASEP command instructs the PE to Page Erase [NUM_PAGES] of code memory. The code memory must be erased at an “even” 512 instruction words address boundary.

Expected Response (2 words):

0x1900
0x0002

6.2.4.8 QVER Command

15	12	11	0
Opcode	Length		

Table 6-9 shows the description for the QVER command.

TABLE 6-9: COMMAND DESCRIPTION

Field	Description
Opcode	0xB
Length	0x1

The QVER command queries the version of the PE 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 means Version 2.3 of PE software).

Expected Response (2 words):

0x1BMN (where “MN” stands for version M.N)
0x0002

6.2.4.9 CRCP Command

15	12	11	8	7	0
Opcode		Length			
Reserved			Addr_MSB		
Addr_LSW					
Reserved			Size_MSB		
Size_LSW					

Table 6-11 shows the description for the CRCP command.

TABLE 6-10: COMMAND DESCRIPTION

Field	Description
Opcode	Ch
Length	5h
Addr_MSB	Most Significant Byte of 24-bit address
Addr_LSW	Least Significant 16 bits of 24-bit address
Size	Number of 24-bit locations (address range divided by 2)

The CRCP command performs a CRC-16 on the range of memory specified. This command can substitute for a full chip verify. Data is shifted in a packed method, as demonstrated in Figure 6-4, bitwise, Least Significant Byte (LSB) first.

Example:

CRC-CCITT-16 with test data of “123456789” becomes 29B1h

Expected Response (3 words):

QE_Code: 0x1C00
Length: 0x0003
CRC Value: 0xFFFF

6.2.4.10 QBLANK Command

15	12	11	0
Opcode		Length	
Reserved		Size_MSB	
Size_LSW			
Reserved		Addr_MSB	
Addr_LSW			

Table 6-11 shows the description for the QBLANK command.

TABLE 6-11: COMMAND DESCRIPTION

Field	Description
Opcode	0xE
Length	0x5
Size	Length of program memory to check (in 24-bit words) + Addr_MS
Addr_MSB	Most Significant Byte of the 24-bit address
Addr_LSW	Least Significant 16 bits of the 24-bit address

The QBLANK command queries the PE to determine if the contents of code memory are blank (contains all ‘1’s). The size of code memory to check must be specified in the command.

The Blank Check for code memory begins at [Addr] and advances toward larger addresses for the specified number of instruction words.

QBLANK returns a QE_Code of F0h if the specified code memory is blank; otherwise, QBLANK returns a QE_Code of 0Fh.

Expected Response (2 words for blank device):

0x1EF0
0x0002

Expected Response (2 words for non-blank device):

0x1E0F
0x0002

Note: The QBLANK command does not check the system operation Configuration bits, since these bits are not set to ‘1’ when a Chip Erase is performed.

6.3 Programming Executive Responses

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

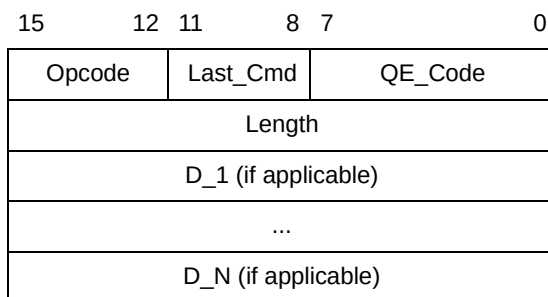
The PE response set is shown in [Table 6-12](#). This table contains the opcode, mnemonic and description for each response. The response format is described in [Section 6.3.1 “Response Format”](#).

TABLE 6-12: PROGRAMMING EXECUTIVE RESPONSE OPCODES

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

6.3.1 RESPONSE FORMAT

All PE responses have a general format, consisting of a two-word header and any required data for the command.



[Table 6-13](#) shows the description of the response format.

TABLE 6-13: RESPONSE FORMAT DESCRIPTION

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)

6.3.1.1 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 6-12](#)). If the command was processed successfully, the response opcode is PASS. If there was 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 PE is not identified, the PE returns a NACK response.

6.3.1.2 Last_Cmd Field

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

6.3.1.3 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 PE 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 6-14](#).

TABLE 6-14: QE_Code FOR QUERIES

Query	QE_Code
QBLANK	0x0F = Code memory is NOT blank 0xF0 = Code memory is blank
QVER	0xMN, where PE Software Version = M.N (i.e., 0x32 means Software Version 3.2)

When the PE processes any command other than a query, the QE_Code represents an error code. Supported error codes are shown in [Table 6-15](#). If a command is successfully processed, the returned QE_Code is set to 0x0, which indicates that there is no error in the command processing. If the verify of the programming for the PROGW command fails, the QE_Code is set to 0x1. For all other PE errors, the QE_Code is 0x02.

TABLE 6-15: QE_Code FOR NON-QUERY COMMANDS

QE_Code	Description
0x0	No error
0x1	Verify failed
0x2	Other error

6.3.1.4 Response Length

The response length indicates the length of the PE's response in 16-bit words. This field includes the 2 words of the response header.

With the exception of the response for the read commands, the length of each response is only 2 words.

The response to the READP commands uses the packed instruction word format described in [Section 6.2.2 "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.

7.0 DUAL PARTITION FLASH PROGRAMMING CONSIDERATIONS

The dsPIC33EPXXXGS70X/80X family of devices supports a Single Partition Flash mode and two Dual Partition Flash modes. The Dual Partition Flash modes allow the device to be programmed with two separate applications to facilitate bootloading, or to allow an application to be programmed at run time without stalling the CPU.

The device's Boot mode is determined by the BTMODE<1:0> bits in the FBOOT Configuration register (see [Table 7-1](#)). The device will automatically check FBOOT on Reset and determine the appropriate Boot mode.

TABLE 7-1: BOOT MODE SELECT

FBOOT<1:0>	Boot Mode
00	Reserved
01	Protected Dual Partition Flash mode
10	Dual Partition Flash mode
11	Single Partition Flash mode (default)

Protected Dual Partition Flash mode prevents run-time programming and erase functions for Partition 1; ICSP modes are not affected.

7.1 Dual Partition Flash Program Memory Organization

In the Dual Partition Flash modes, the device's memory is divided evenly into two physical sections, known as Partition 1 and Partition 2. Each of these partitions contains its own program memory and Configuration Words. During the program execution, the code on only one of these partitions can be executed and that will be the Active Partition. The other partition, or the Inactive Partition, cannot be used for execution but can be programmed.

The Active Partition is always mapped to program address, 000000h, while the Inactive Partition will always be mapped to program address, 400000h. Even when the code partitions are switched between Active and Inactive by the user, the address of the Active Partition will still be 000000h and the address of the Inactive Partition will be 400000h.

The Boot Sequence Configuration Word (FBTSEQ) determines whether Partition 1 or Partition 2 will be active after Reset. If the device is operating in Dual Partition Flash mode, the partition with the lower Boot Sequence Number will operate as the Active Partition (FBTSEQ is unused in Single Partition mode). The partitions can be switched between Active and Inactive by reprogramming their Boot Sequence Numbers, but the Active Partition will not change until a device Reset is performed. If both the Boot Sequence Numbers are

the same, or if both are corrupted, the part will use Partition 1 as the Active Partition. If only one Boot Sequence Number is corrupted, the device will use the partition without a corrupted Boot Sequence Number as the Active Partition.

The user can also change which partition is active at run time using the BOOTSWP instruction. The BOOTSWP instruction must be enabled before it can be used (located in the FICD Configuration Word). Issuing a BOOTSWP instruction does not affect which partition will be the Active Partition after a Reset. [Figure 7-1](#) shows how Partitions 1 and 2 are swapped between the Active and Inactive Partitions during FBTSEQ reprogramming, and BOOTSWP execution, respectively.

The P2ACTIV bit (NVMCON<10>) can be used to determine which physical partition is the Active Partition. If P2ACTIV = 1, Partition 2 is active; if P2ACTIV = 0, Partition 1 is active.

7.2 Erase Operations with Dual Partition Flash

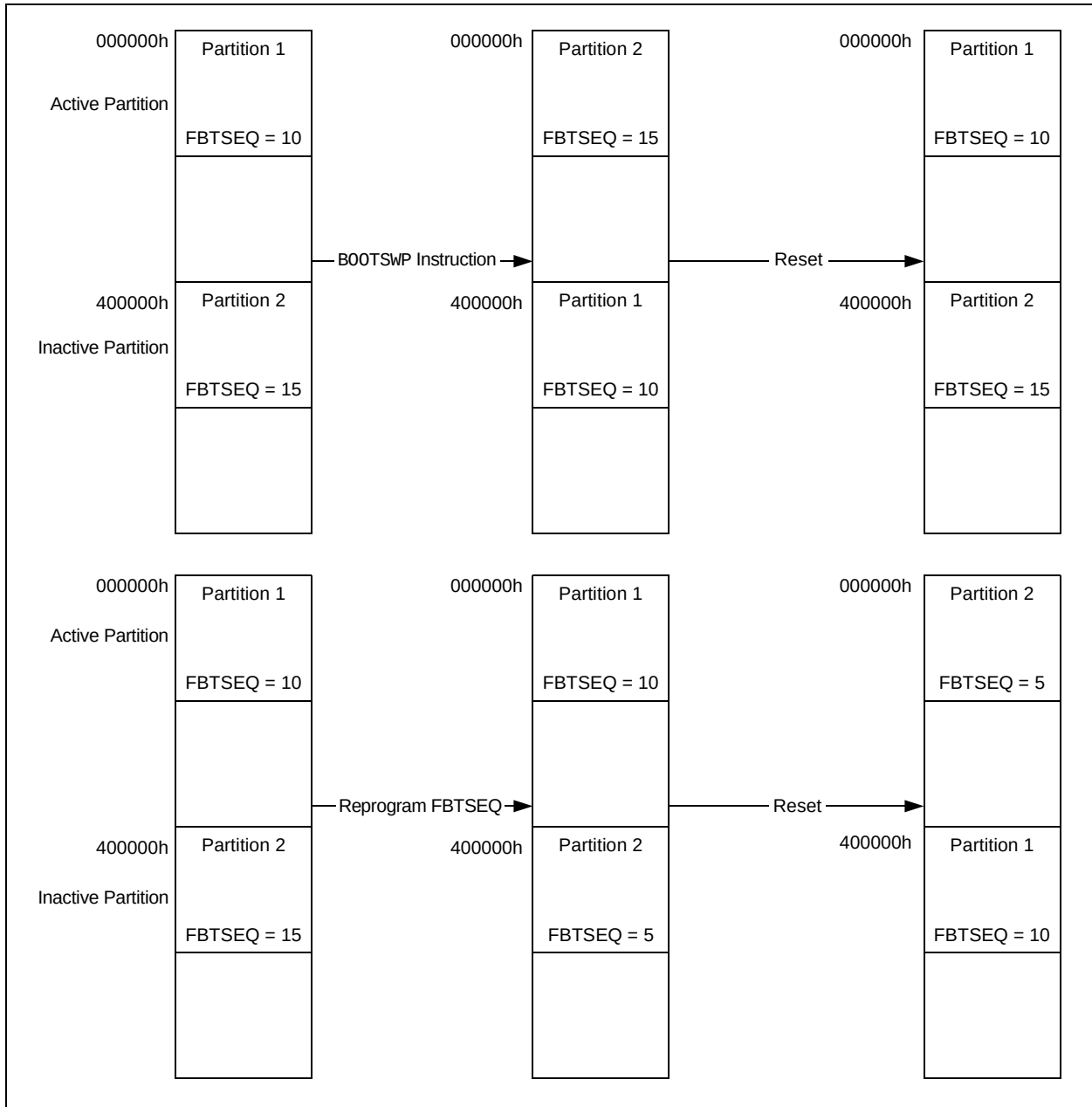
dsPIC33EPXXXGS70X/80X family devices support the following three erase operations: Bulk Erase, Inactive Partition Erase and Page Erase.

A Bulk Erase operation erases all user memory, including the Flash Configuration Words and the FBOOT Configuration register. This restores the Boot mode of the device to its default, Single Partition mode after Reset.

An Inactive Partition Erase operation can be executed at run time from the Active Partition. It will erase all user memory and Flash Configuration Words in the Inactive Partition. The Inactive Partition Erase command is functional only when the device is in one of the Dual Partition Flash modes.

The Flash Configuration Words reside in the last locations of each partition. They may be erased using a Bulk Erase, an Inactive Partition Erase or a Page Erase that targets the last page of the partition.

FIGURE 7-1: RELATIONSHIP BETWEEN PARTITIONS 1 AND 2 AND ACTIVE/INACTIVE PARTITIONS



7.3 Dual Partition Configuration Words

In Dual Partition modes, each partition has its own set of Flash Configuration Words. The full set of Configuration registers in the Active Partition is used to determine the device's configuration. The Configuration Words in the Inactive Partition are not used until FBTSEQ is programmed for it to be the Active Partition and a Reset occurs. The BOOTSWP instruction does not change the device's effective configuration based on the newly Active Partition's Configuration Words. However, some of the Configuration registers in the Inactive Partition (FSEC, FBSLIM and FSIGN) may be used to determine how the Active Partition is enabled or allowed to access the Inactive Partition.

Bulk Erase of user memory will automatically program the reserved bit in Partition 1's FSIGN register (FSIGN<15>). Therefore, it is necessary to manually program the reserved bit in Partition 2's FSIGN register if the user reprograms FBOOT to use a Dual Partition Flash mode.

7.4 Programming in Dual Partition Flash Mode

The following are the steps required to program a dsPIC33EPXXXGS70X/80X device with Dual Partition Flash:

1. Perform a Bulk Erase of user program memory.
2. Re-enter ICSP mode so that the program memory address map starts at a known state.
3. Program the FBOOT register to one of the Dual Partition Flash modes.
4. Execute a Page Erase at address, 0x005400.

Note: Do not reset the device now, as resetting at this stage would require repeating execution of the programming sequence from Step 1.

5. Program address, 0x00AB94 = 0xFF7FFF (leaving the adjacent word as 0xFFFFF).
6. Issue a device Reset. This will split the memory into two partitions.
7. Program the first application, including its Boot Sequence Number and Configuration Words into the Active Partition at address, 000000h.
8. Program the second application, its Boot Sequence Number and Configuration Words into the Inactive Partition at address, 400000h.

dsPIC33EPXXXGS70X/80X FAMILY

8.0 DEVICE ID/UNIQUE ID

The Device ID region of memory can be used to determine variant and manufacturing information about the chip. This region of memory is read-only and can be read when code protection is enabled.

Table 8-1 lists the identification information for each device. Table 8-2 shows the Device ID registers.

TABLE 8-1: DEVICE IDs

Device	DEVID	DEVREV	Silicon Revision
dsPIC33EP64GS708	0x6C03	0x6C00	A0
dsPIC33EP64GS804	0x6C40		
dsPIC33EP64GS805	0x6C60		
dsPIC33EP64GS806	0x6C42		
dsPIC33EP64GS808	0x6C43		
dsPIC33EP128GS702	0x6C11		
dsPIC33EP128GS704	0x6C10		
dsPIC33EP128GS705	0x6C30		
dsPIC33EP128GS706	0x6C12		
dsPIC33EP128GS708	0x6C13		
dsPIC33EP128GS804	0x6C50		
dsPIC33EP128GS805	0x6C70		
dsPIC33EP128GS806	0x6C52		
dsPIC33EP128GS808	0x6C53		

TABLE 8-2: DEVICE ID REGISTERS

Address	Name	Bit															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FF0000h	DEVID	DEVID Value															
FF0002h	DEVREV	DEVREV Value															

REGISTER 8-1: JTAG ID REGISTER

31	28	27	12	11	0
DEVREV<3:0>		DEVID<15:0>		Manufacturer ID (053h)	
4 bits		16 bits		12 bits	

dsPIC33EPXXXGS70X/80X FAMILY

8.1 Unique Device ID (UDID)

All dsPIC33EPXXXGS70X/80X family devices are individually encoded during final manufacturing with a Unique Device Identifier or UDID. The UDID cannot be erased by a Bulk Erase command or any other user-accessible means. This feature allows for manufacturing traceability of Microchip Technology devices in applications where this is a requirement. It may also be used by the application manufacturer for any number of things that may require unique identification, such as:

- Tracking the device
- Unique serial number
- Unique security key

The UDID comprises five 24-bit program words. When taken together, these fields form a unique 120-bit identifier. The UDID is stored in five read-only locations in the device configuration space. [Table 8-3](#) lists the addresses of the Identifier Words and shows their contents.

TABLE 8-3: UDID ADDRESSES

UDID	Address	Content
UDID1	800F00	UDID Word 1
UDID2	800F02	UDID Word 2
UDID3	800F04	UDID Word 3
UDID4	800F06	UDID Word 4
UDID5	800F08	UDID Word 5

9.0 CHECKSUM COMPUTATION

Checksums for devices are 16 bits in size. The checksum is calculated by summing the following:

- Contents of code memory locations
- Contents of Configuration Words (except the FBTSEQ Configuration register)

If a Configuration bit is defaulted to a '0' (for example, JTAGEN), the erased state is still a '1'. The bit is set to '0' by the device programmer and is masked when computing the checksum. Therefore, the term, 'Blank Checksum', when used by MPLAB® X IDE or other programmers, must take masked Configuration bits into account.

All memory locations, including Configuration Words, are summed by adding all three bytes of each memory address. For FSIGN and FICD, a read mask is used to ignore the bit which is reserved.

The following masks are used when calculating the checksum for the configuration memory portion:

- FSIGN: FF7FFFh, which masks off bit 15
- FICD: FFFFDFh, which masks off the JTAGEN bit (FICD<5>)

The checksum computation spans the entire regions of program memory and configuration memory. The Configuration Words may occupy a small area of the total configuration memory, but the entire address space is used in the checksum computation.

TABLE 9-1: CHECKSUM COMPUTATION EXAMPLE (SINGLE PARTITION FLASH)

Device	Read Code Protection	Checksum Computation	Blank Value	Value with 0xAAAAAA at 0x0 and Last Code Address
dsPIC33EP64GS70X	Disabled	PROG[0:00AF7Eh] + CFGB[00AF80h:00AFFEh]	0xF463	0xF265
	Enabled	0x00	0x0000	0x0000
dsPIC33EP128GS70X	Disabled	PROG[0:01577Eh] + CFGB[015780h:0157FEh]	0xF863	0xF665
	Enabled	0x00	0x0000	0x0000

Legend: PROG[a:b] = Program memory byte sum of locations, a to b inclusive (all 3 bytes of code memory)

CFGB[c:d] = Configuration memory byte sum of locations, c to d inclusive, with FSIGN and FICD masked, and FBTSEQ omitted

TABLE 9-2: CHECKSUM COMPUTATION EXAMPLE (DUAL PARTITION FLASH)

Device	Read Code Protection	Checksum Computation	Blank Value	Value with 0xAAAAAA at 0x0 and Last Code Address
dsPIC33EP64GS70X	Disabled	PROG[0:00577Eh] + CFGB[005780h:0057FEh] + PROG[40000h:40577Eh] + CFGB[40AB80h:40ABFEh]	0xF0C6	0xECCA
	Enabled	0x00	0x0000	0x0000
dsPIC33EP128GS70X	Disabled	PROG[0:00AB7Eh] + CFGB[00AB80h:00ABFEh] + PROG[40000h:40AB7Eh] + CFGB[40AB80h:40ABFEh]	0xF4C6	0xF0CA
	Enabled	0x00	0x0000	0x0000

Legend: PROG[a:b] = Program memory byte sum of locations, a to b inclusive (all 3 bytes of code memory)

CFGB[c:d] = Configuration memory byte sum of locations, c to d inclusive, with FSIGN and FICD masked, and FBTSEQ omitted

dsPIC33EPXXXGS70X/80X FAMILY

10.0 AC/DC CHARACTERISTICS AND TIMING REQUIREMENTS

Table 10-1 lists the AC/DC characteristics and timing requirements.

TABLE 10-1: AC/DC CHARACTERISTICS AND TIMING REQUIREMENTS

Standard Operating Conditions						
Operating Temperature: -40°C to +85°C. Programming at +25°C is recommended.						
Param. No.	Symbol	Characteristic	Min.	Max.	Units	Conditions
D111	VDD	Supply Voltage During Programming	3.0	3.6	V	See Note 1
P1	TPGC	Serial Clock (PGCx) Period (ICSP™)	200	—	ns	
P1	TPGC	Serial Clock (PGCx) Period (Enhanced ICSP)	500	—	ns	
P1A	TPGCL	Serial Clock (PGCx) Low Time (ICSP)	80	—	ns	
P1A	TPGCL	Serial Clock (PGCx) Low Time (Enhanced ICSP)	200	—	ns	
P1B	TPGCH	Serial Clock (PGCx) High Time (ICSP)	80	—	ns	
P1B	TPGCH	Serial Clock (PGCx) High Time (Enhanced ICSP)	200	—	ns	
P2	TSET1	Input Data Setup Time to Serial Clock ↓	15	—	ns	
P3	THLD1	Input Data Hold Time from PGCx ↓	15	—	ns	
P4	TDLY1	Delay Between 4-Bit Command and Command Operand	40	—	ns	
P4A	TDLY1A	Delay Between Command Operand and Next 4-Bit Command	40	—	ns	
P5	TDLY2	Delay Between Last PGCx ↓ of Command to First PGCx ↑ of Read of Data Word	20	—	ns	
P6	TSET2	VDD ↑ Setup Time to MCLR ↑	100	—	ns	
P7	THLD2	Input Data Hold Time from MCLR ↑	50	—	ms	
P8	TDLY3	Delay Between Last PGCx ↓ of Command Byte to PGDx ↑ by PE	12	—	μs	
P9A	TDLY4	PE Command Processing Time	10	—	μs	
P9B	TDLY5	Delay Between PGDx ↓ by PE to PGDx Released by PE	15	23	μs	
P10	TDLY6	PGCx Low Time After Programming	400	—	ns	
P11	TDLY7	Bulk Erase Time	16	24	ms	
P12	TDLY8	Page Erase Time	—	—	ms	See Note 2
P13A	TDLY9A	Row Programming Time	—	—	μs	See Note 2 and Note 3
P13B	TDLY9B	Double-Word Programming Time	—	—	μs	See Note 2 and Note 3
P14	TR	MCLR Rise Time to Enter ICSP mode	—	1.0	μs	
P15	TVALID	Data Out Valid from PGCx ↑	10	—	ns	
P16	TDLY10	Delay Between Last PGCx ↓ and MCLR ↓	0	—	s	

Note 1: VDD must also be supplied to the AVDD pins during programming. AVDD and AVSS should always be within ±0.3V of VDD and VSS, respectively.

2: Time depends on the FRC accuracy and the value of the FRC Oscillator Tuning register. Refer to the “**Electrical Characteristics**” chapter in the specific device data sheet.

3: This time applies to Program Memory Words, Configuration Words and user OTP Words.

TABLE 10-1: AC/DC CHARACTERISTICS AND TIMING REQUIREMENTS (CONTINUED)

Standard Operating Conditions Operating Temperature: -40°C to +85°C. Programming at +25°C is recommended.						
Param. No.	Symbol	Characteristic	Min.	Max.	Units	Conditions
P17	THLD3	MCLR ↓ to VDD ↓	100	—	ns	
P18	TKEY1	Delay from First MCLR ↓ to First PGCx ↑ for Key Sequence on PGDx	1	—	ms	
P19	TKEY2	Delay from Last PGCx ↓ for Key Sequence on PGDx to Second MCLR ↑	25	—	ns	
P21	TMCLRH	MCLR High Time	—	500	μs	

- Note 1:** VDD must also be supplied to the AVDD pins during programming. AVDD and AVSS should always be within ±0.3V of VDD and VSS, respectively.
- 2:** Time depends on the FRC accuracy and the value of the FRC Oscillator Tuning register. Refer to the “**Electrical Characteristics**” chapter in the specific device data sheet.
- 3:** This time applies to Program Memory Words, Configuration Words and user OTP Words.

dsPIC33EPXXXGS70X/80X FAMILY

APPENDIX A: REVISION HISTORY

Revision A (April 2016)

Original version of the programming specification created for the dsPIC33EPXXXGS70X/80X device family.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949 ==

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, dsPIC, FlashFlex, flexPWR, Helder, JukeBlox, KeeLoq, KeeLoq logo, Klear, LANCheck, LINK MD, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC32 logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, ETHERSYNCH, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and QUIET-WIRE are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KlearNet, KlearNet logo, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, RightTouch logo, REAL ICE, Ripple Blocker, Serial Quad I/O, SQL, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2016, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-0522-1

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
<http://www.microchip.com/support>
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820