Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

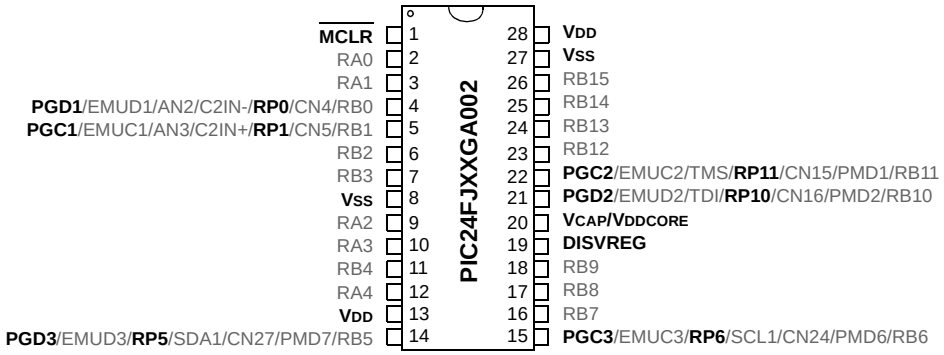
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

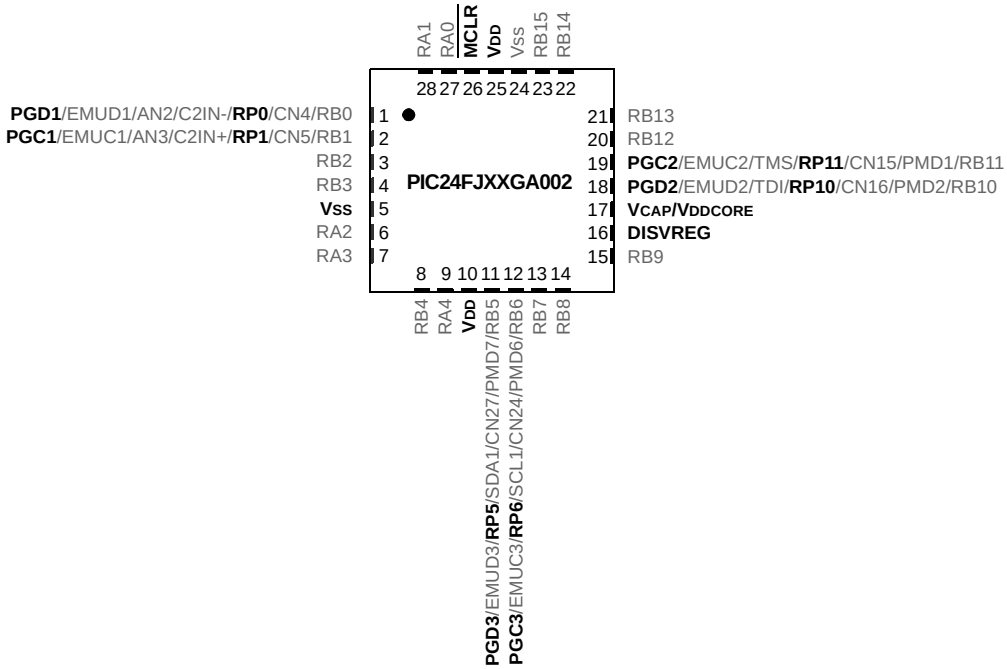
Details	
Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	16MHz
Connectivity	I²C, PMP, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	69
Program Memory Size	128KB (43K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj128ga008-i-pt

Pin Diagrams

28-Pin PDIP, SSOP, SOIC



28-Pin QFN(1)

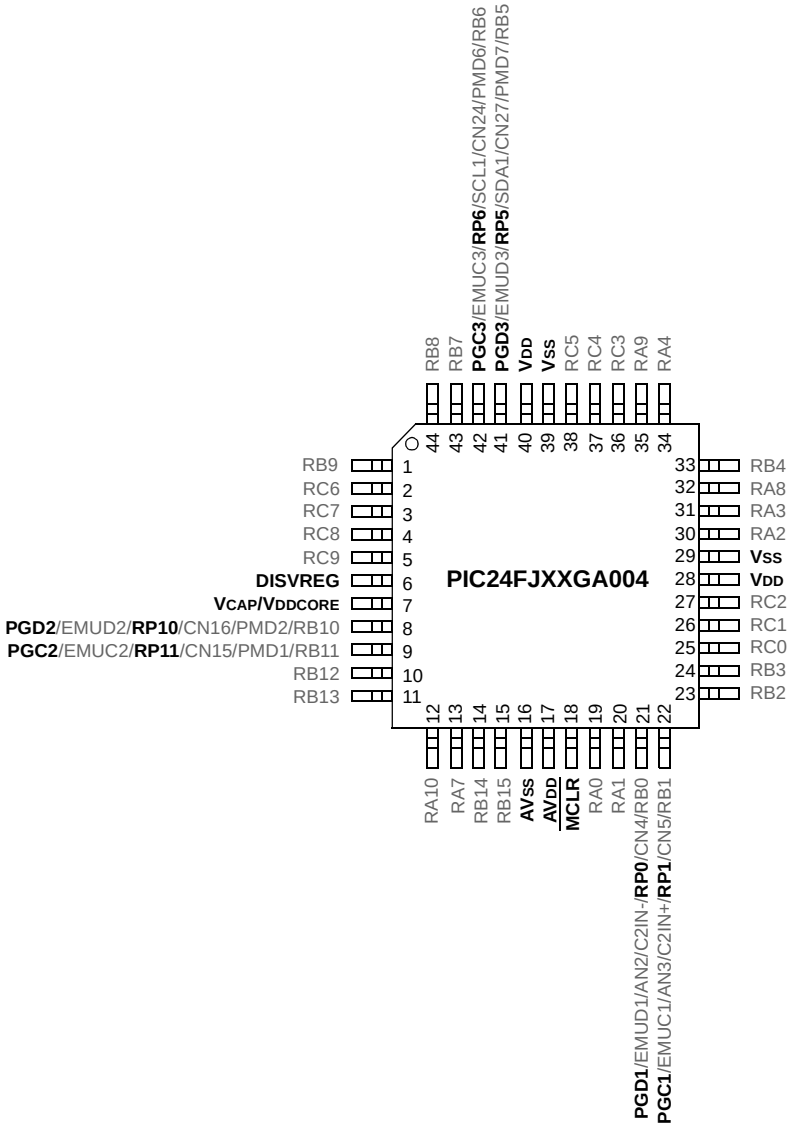


Legend: **RPx** represents remappable peripheral pins.

Note 1: The bottom pad of QFN packages should be connected to VSS.

Pin Diagrams (Continued)

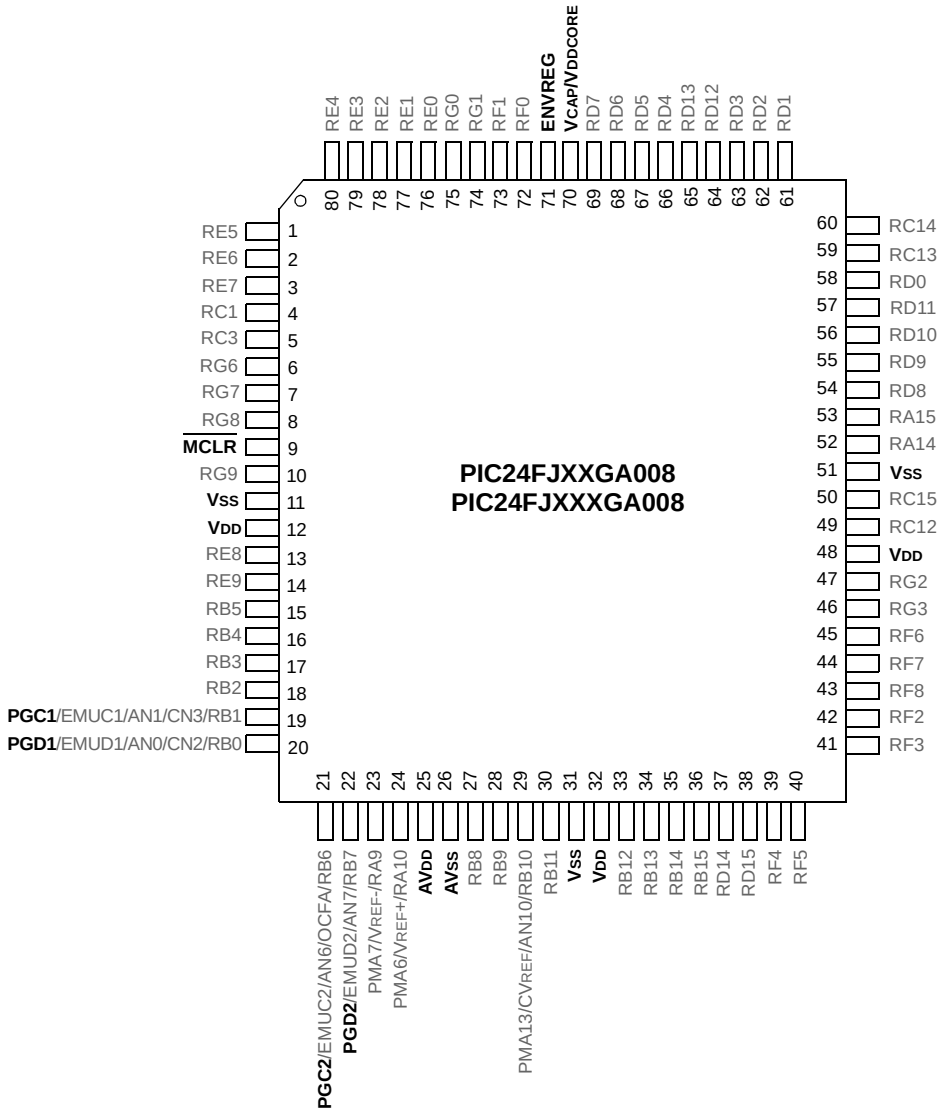
44-Pin TQFP



Legend: **RPx** represents remappable peripheral pins.

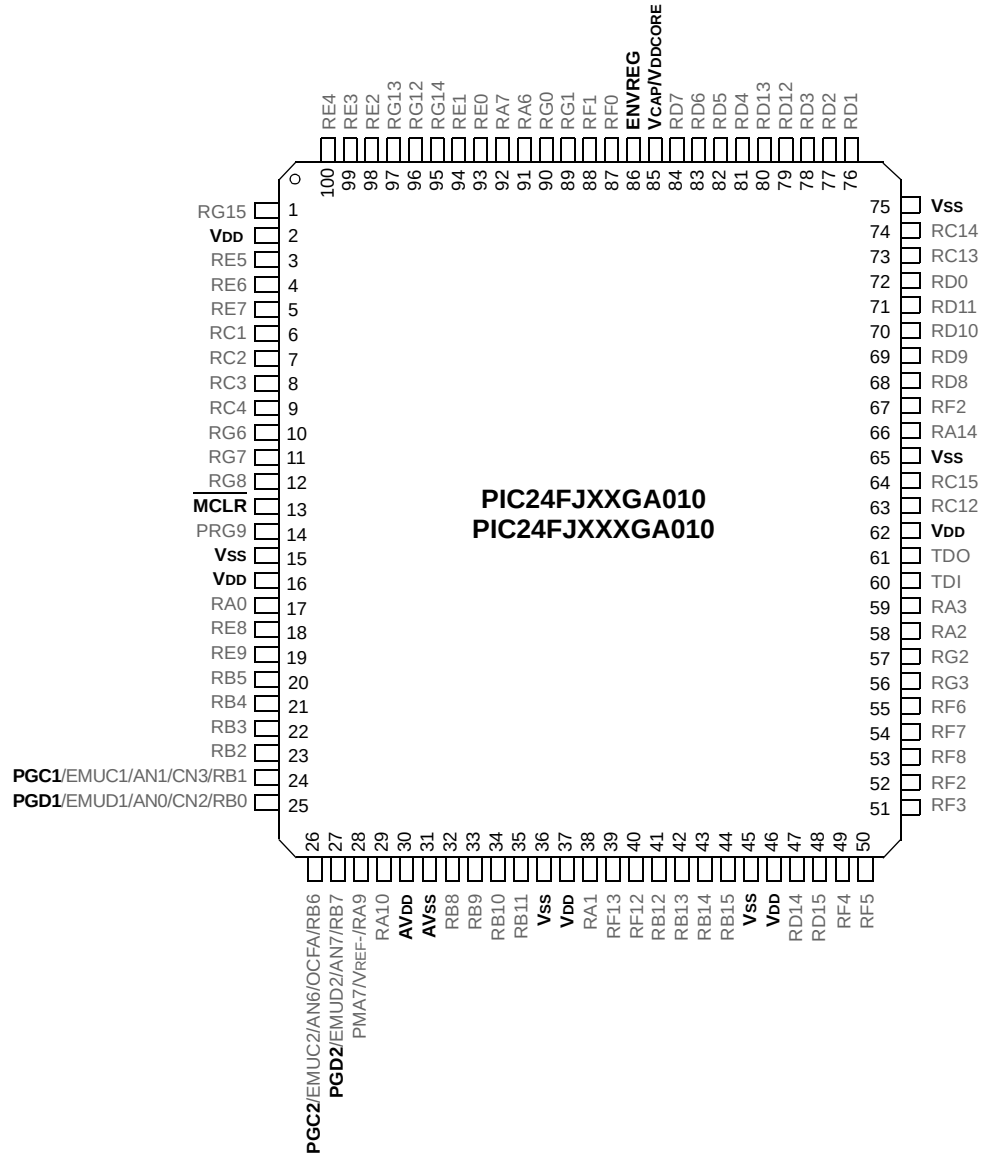
Pin Diagrams (Continued)

80-Pin TQFP



Pin Diagrams (Continued)

100-Pin TQFP



2.4 Memory Map

The program memory map extends from 000000h to FFFFFEh. Code storage is located at the base of the memory map and supports up to 44K instruction words (about 128 Kbytes). Table 2-3 shows the program memory size and number of erase and program blocks present in each device variant. Each erase block, or page, contains 512 instructions, and each program block, or row, contains 64 instructions.

Locations 800000h through 8007FEh are reserved for executive code memory. This region stores the programming executive and the debugging executive. The programming executive is used for device programming and the debugging executive is used for in-circuit debugging. This region of memory can not be used to store user code.

The last two implemented program memory locations are reserved for the device Configuration registers.

TABLE 2-2: FLASH CONFIGURATION WORD LOCATIONS FOR PIC24FJXXXGA0XX DEVICES

Device	Configuration Word Addresses	
	1	2
PIC24FJ16GA	002BFEh	002BFCh
PIC24FJ32GA	0057FEh	0057FCh
PIC24FJ48GA	0083FEh	0083FCh
PIC24FJ64GA	00ABFEh	00ABFCh
PIC24FJ96GA	00FFFEh	00FFFCh
PIC24FJ128GAGA	0157FEh	0157FCh

Locations, FF0000h and FF0002h, are reserved for the Device ID registers. These bits can be used by the programmer to identify what device type is being programmed. They are described in **Section 6.1 "Device ID"**. The Device ID registers read out normally, even after code protection is applied.

Figure 2-4 shows the memory map for the PIC24FJXXXGA0XX family variants.

TABLE 2-3: CODE MEMORY SIZE

Device	User Memory Address Limit (Instruction Words)	Write Blocks	Erase Blocks
PIC24FJ16GA	002BFEh (5.5K)	88	11
PIC24FJ32GA	0057FEh (11K)	176	22
PIC24FJ48GA	0083FEh (16.5K)	264	33
PIC24FJ64GA	00ABFEh (22K)	344	43
PIC24FJ96GA	00FFFEh (32K)	512	64
PIC24FJ128GA	0157FEh (44K)	688	86

3.2.1 SIX SERIAL INSTRUCTION EXECUTION

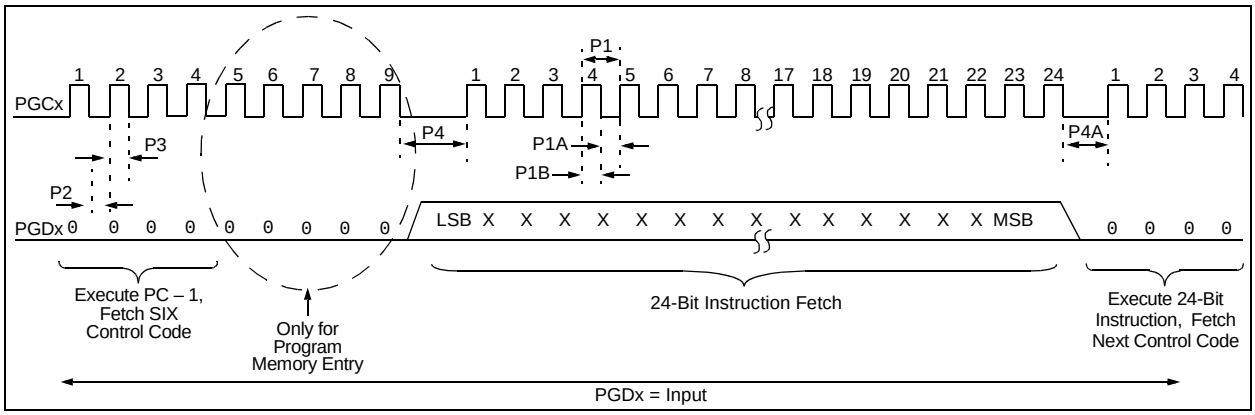
The SIX control code allows execution of the PIC24FJXXXGA0XX family assembly instructions. When the SIX code is received, the CPU is suspended for 24 clock cycles, as the instruction is then clocked into the internal buffer. Once the instruction is shifted in, the state machine allows it to be executed over the next four PGC clock cycles. While the received instruction is executed, the state machine simultaneously shifts in the next 4-bit command (see Figure 3-2).

Coming out of Reset, the first 4-bit control code is always forced to SIX and a forced NOP instruction is executed by the CPU. Five additional PGCx clocks are needed on start-up, resulting in a 9-bit SIX command instead of the normal 4-bit SIX command.

After the forced SIX is clocked in, ICSP operation resumes as normal. That is, the next 24 clock cycles load the first instruction word to the CPU.

Note:	To account for this forced NOP, all example code in this specification begin with a NOP to ensure that no data is lost.

FIGURE 3-2: SIX SERIAL EXECUTION



3.2.1.1 Differences Between Execution of SIX and Normal Instructions

There are some differences between executing instructions normally and using the SIX ICSP command. As a result, the code examples in this specification may not match those for performing the same functions during normal device operation.

The important differences are:

- Two-word instructions require two SIX operations to clock in all the necessary data.

  Examples of two-word instructions are GOTO and CALL.

- Two-cycle instructions require two SIX operations.

  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 – provides the CPU clocks required to finish executing the instruction.

  Examples of two-cycle instructions are table read and table write instructions.

- The CPU does not automatically stall to account for pipeline changes.

  A CPU stall occurs when an instruction modifies a register that is used for Indirect Addressing by the following instruction.

  During normal operation, the CPU automatically will force a NOP while the new data is read. When using ICSP, there is no automatic stall, so any indirect references to a recently modified register should be preceded by a NOP.

  For example, the instructions, `mov #0x0,W0` and `mov [W0],W1`, must have a NOP inserted between them.

  If a two-cycle instruction modifies a register that is used indirectly, it will require two following NOPs: one to execute the second half of the instruction and a second to stall the CPU to correct the pipeline.

  Instructions such as `tblwtl [W0++],[W1]` should be followed by two NOPs.

- The device Program Counter (PC) continues to automatically increment during ICSP instruction execution, even though the Flash memory is not being used.

  As a result, the PC may be incremented to point to invalid memory locations. Invalid memory spaces include unimplemented Flash addresses and the vector space (locations 0x0 to 0x1FF).

  If the PC points to these locations, the device will 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 to accomplish this is to perform a `GOTO 0x200`.

3.3 Entering ICSP Mode

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

1. MCLR is briefly driven high, then low.
2. A 32-bit key sequence is clocked into PGDx.
3. MCLR is then driven high within a specified period of time and held.

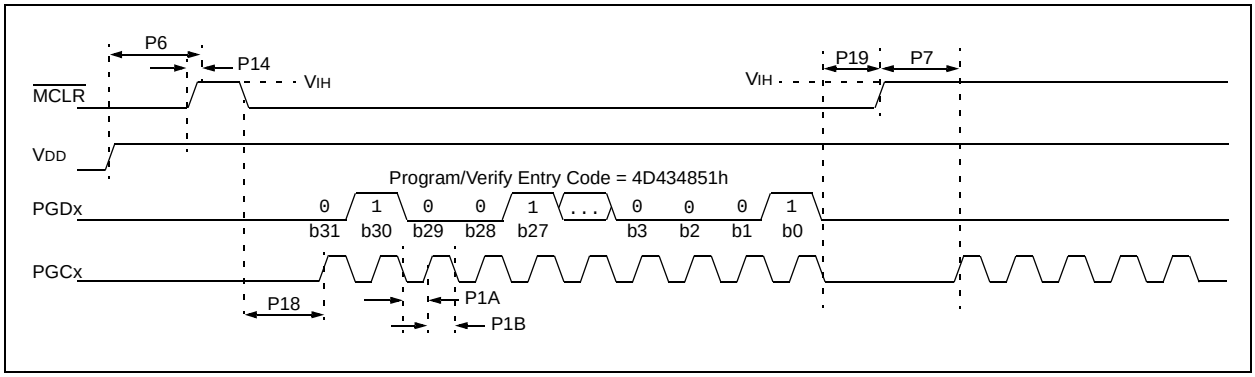
The programming voltage applied to MCLR is VIH, which is essentially VDD in the case of PIC24FJXXXGA0XX devices. There is no minimum time requirement for holding at VIH. After VIH is removed, an interval of at least P18 must elapse before presenting the key sequence on PGDx.

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 Program/Verify mode only if the sequence is valid. The Most Significant bit (MSb) of the most significant nibble must be shifted in first.

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

On successful entry, the program memory can be accessed and programmed in serial fashion. While in ICSP mode, all unused I/Os are placed in the high-impedance state.

FIGURE 3-4: ENTERING ICSP™ MODE



3.4 Flash Memory Programming in ICSP Mode

3.4.1 PROGRAMMING OPERATIONS

Flash memory write and 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 WR control bit (NVMCON<15>).

In ICSP mode, all programming operations are self-timed. There is an internal delay between the user setting the WR control bit and the automatic clearing of the WR control bit when the programming operation is complete. Please refer to **Section 7.0 "AC/DC Characteristics and Timing Requirements"** for information about the delays associated with various programming operations.

TABLE 3-2: NVMCON ERASE OPERATIONS

NVMCON Value	Erase Operation
404Fh	Erase all code memory, executive memory and Configuration registers (does not erase Unit ID or Device ID registers).
4042h	Erase a page of code memory or executive memory.

TABLE 3-3: NVMCON WRITE OPERATIONS

NVMCON Value	Write Operation
4003h	Write a Configuration Word register.
4001h	Program 1 row (64 instruction words) of code memory or executive memory.

3.4.2 STARTING AND STOPPING A PROGRAMMING CYCLE

The WR bit (NVMCON<15>) is used to start an erase or write cycle. Setting the WR bit initiates the programming cycle.

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. Starting a programming cycle is performed as follows:

```
BSET    NVMCON, #WR
```

3.5 Erasing Program Memory

The procedure for erasing program memory (all of code memory, data memory, executive memory and code-protect bits) consists of setting NVMCON to 404Fh and executing the programming cycle.

A Chip Erase can erase all of user memory or all of both the user and configuration memory. A table write instruction should be executed prior to performing the Chip Erase to select which sections are erased.

When this table write instruction is executed:

- If the TBLPAG register points to user space (is less than 0x80), the Chip Erase will erase only user memory.
- If TBLPAG points to configuration space (is greater than or equal to 0x80), the Chip Erase will erase both user and configuration memory.

  If configuration memory is erased, the internal oscillator Calibration Word, located at 0x807FE, will be erased. This location should be stored prior to performing a whole Chip Erase and restored afterward to prevent internal oscillators from becoming uncalibrated.

Figure 3-5 shows the ICSP programming process for performing a Chip Erase. This process includes the ICSP command code, which must be transmitted (for each instruction), Least Significant bit first, using the PGCx and PGDx pins (see Figure 3-2).

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

FIGURE 3-5: CHIP ERASE FLOW

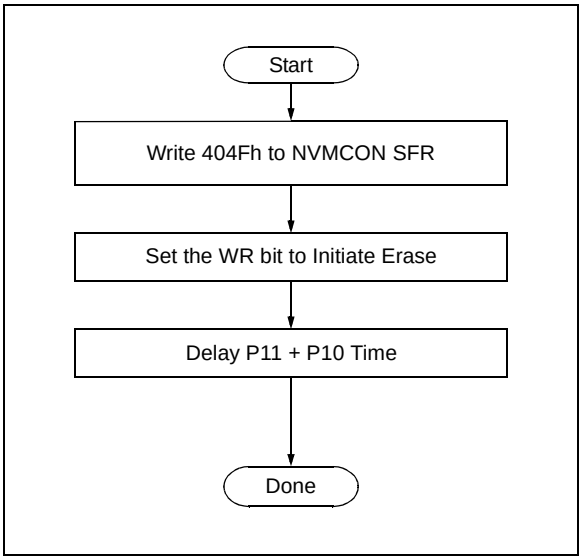


TABLE 3-5: SERIAL INSTRUCTION EXECUTION FOR WRITING CODE MEMORY (CONTINUED)

Command (Binary)	Data (Hex)	Description
Step 5: Set the Read Pointer (W6) and load the (next set of) write latches.		
0000	EB0300	CLR W6
0000	000000	NOP
0000	BB0BB6	TBLWTL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0000	BBDBB6	TBLWTH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BBEBB6	TBLWTH.B [W6++], [++W7]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BB0BB6	TBLWTL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0000	BBDBB6	TBLWTH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BBEBB6	TBLWTH.B [W6++], [++W7]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
Step 6: Repeat Steps 4 and 5, sixteen times, to load the write latches for 64 instructions.		
Step 7: Initiate the write cycle.		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
Step 8: Repeat this step to poll the WR bit (bit 15 of NVMCON) until it is cleared by the hardware.		
0000	040200	GOTO 0x200
0000	000000	NOP
0000	803B02	MOV NVMCON, W2
0000	883C22	MOV W2, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of the VISI register.
0000	000000	NOP
Step 9: Reset device internal PC.		
0000	040200	GOTO 0x200
0000	000000	NOP
Step 10: Repeat Steps 3-9 until all code memory is programmed.		

FIGURE 3-7: PROGRAM CODE MEMORY FLOW

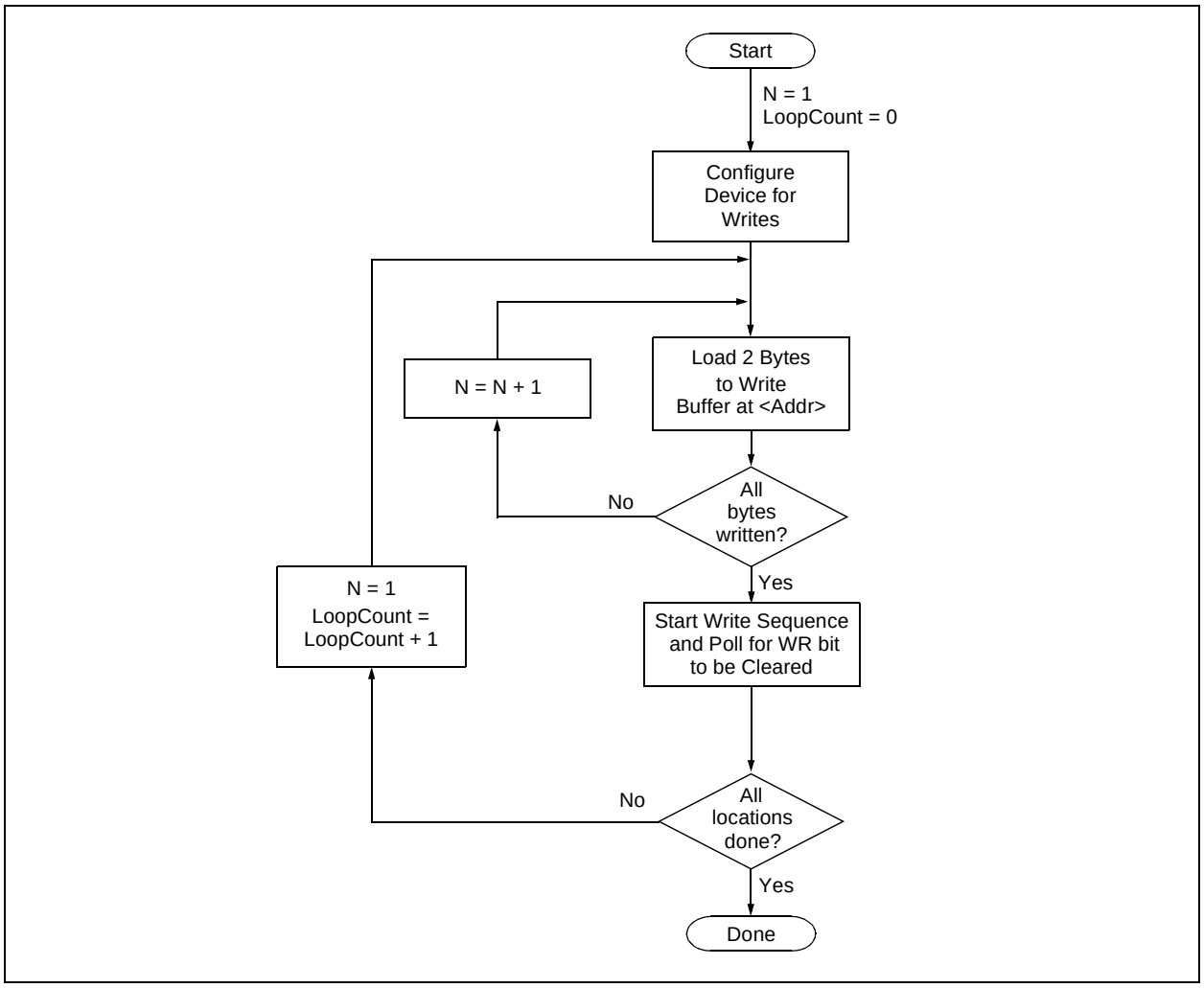


TABLE 3-7: SERIAL INSTRUCTION EXECUTION FOR WRITING CONFIGURATION REGISTERS

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
Step 2: Initialize the Write Pointer (W7) for the TBLWT instruction.		
0000	2xxxx7	MOV <CW2Address15:0>, W7
Step 3: Set the NVMCON register to program CW2.		
0000	24003A	MOV #0x4003, W10
0000	883B0A	MOV W10, NVMCON
Step 4: Initialize the TBLPAG register.		
0000	200xx0	MOV <CW2Address23:16>, W0
0000	880190	MOV W0, TBLPAG
Step 5: Load the Configuration register data to W6.		
0000	2xxxx6	MOV #<CW2_VALUE>, W6
Step 6: Write the Configuration register data to the write latch and increment the Write Pointer.		
0000	000000	NOP
0000	BB1B86	TBLWTL W6, [W7++]
0000	000000	NOP
0000	000000	NOP
Step 7: Initiate the write cycle.		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
Step 8: Repeat this step to poll the WR bit (bit 15 of NVMCON) until it is cleared by the hardware.		
0000	040200	GOTO 0x200
0000	000000	NOP
0000	803B02	MOV NVMCON, W2
0000	883C22	MOV W2, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of the VISI register.
0000	000000	NOP
Step 9: Reset device internal PC.		
0000	040200	GOTO 0x200
0000	000000	NOP
Step 10: Repeat Steps 5-9 to write CW1.		

3.8 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 3-8 shows the ICSP programming details for reading code memory. In Step 1, the Reset vector is exited. In Step 2, the 24-bit starting source address for reading is loaded into the TBLPAG register and W6 register. The upper byte of the starting source address is stored in TBLPAG and the lower 16 bits of the source address are stored in W6.

To minimize the reading time, the packed instruction word format that was utilized for writing is also used for reading (see Figure 3-6). In Step 3, the Write Pointer, W7, is initialized. In Step 4, two instruction words are read from code memory and clocked out of the device, through the VISI register, using the REGOUT command. Step 4 is repeated until the desired amount of code memory is read.

TABLE 3-8: SERIAL INSTRUCTION EXECUTION FOR READING CODE MEMORY

Command (Binary)	Data (Hex)	Description
Step 1: Exit Reset vector.		
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
Step 2: Initialize TBLPAG and the Read Pointer (W6) for TBLRD instruction.		
0000	200xx0	MOV #<SourceAddress23:16>, W0
0000	880190	MOV W0, TBLPAG
0000	2xxxx6	MOV #<SourceAddress15:0>, W6
Step 3: Initialize the Write Pointer (W7) to point to the VISI register.		
0000	207847	MOV #VISI, W7
0000	000000	NOP
Step 4: Read and clock out the contents of the next two locations of code memory, through the VISI register, using the REGOUT command.		
0000	BA0B96	TBLRDL [W6], [W7]
0000	000000	NOP
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	BADBB6	TBLRDH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BAD3D6	TBLRDH.B [++W6], [W7--]
0000	000000	NOP
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	BA0BB6	TBLRDL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
Step 5: Reset device internal PC.		
0000	040200	GOTO 0x200
0000	000000	NOP
Step 6: Repeat Steps 4 and 5 until all desired code memory is read.		

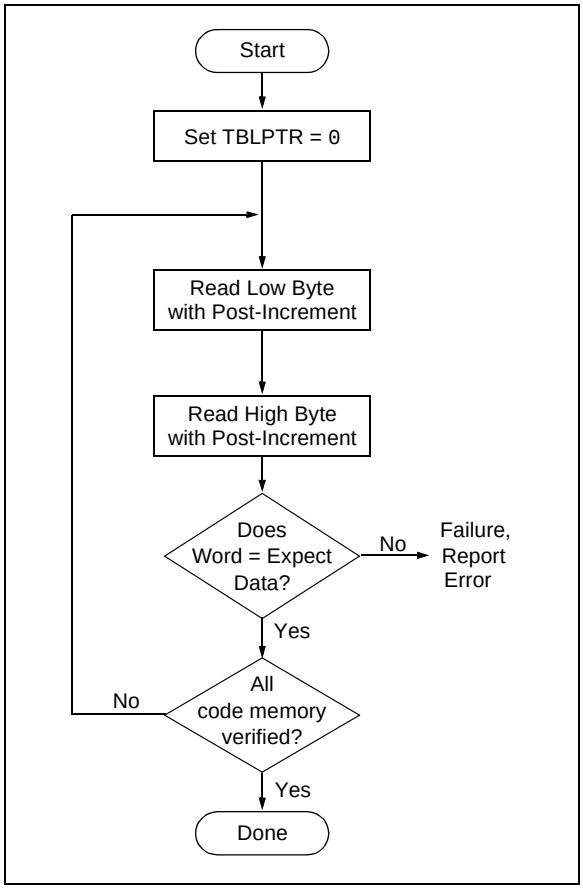
3.10 Verify Code Memory and Configuration Word

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

The verify process is shown in the flowchart in Figure 3-8. Memory reads occur a single byte at a time, so two bytes must be read to compare against the word in the programmer's buffer. Refer to **Section 3.8 "Reading Code Memory"** for implementation details of reading code memory.

Note:	Because the Configuration registers include the device code protection bit, code memory should be verified immediately after writing if code protection is enabled. This is because the device will not be readable or verifiable if a device Reset occurs after the code-protect bit in CW1 has been cleared.

FIGURE 3-8: VERIFY CODE MEMORY FLOW



3.11 Reading the Application ID Word

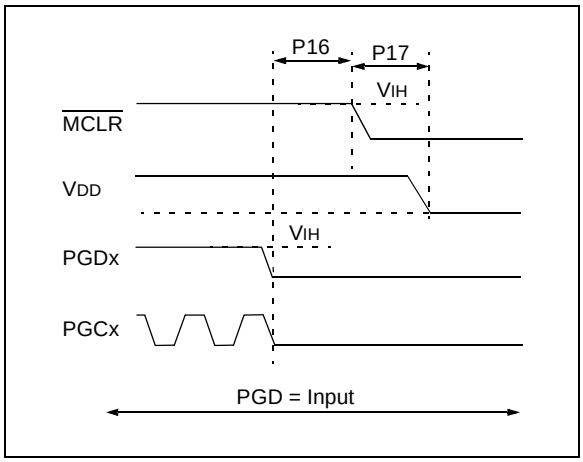
The Application ID Word is stored at address 8005BEh in executive code memory. To read this memory location, you must use the SIX control code to move this program memory location to the VISI register. Then, the REGOUT control code must be used to clock the contents of the VISI register out of the device. The corresponding control and instruction codes that must be serially transmitted to the device to perform this operation are shown in Table 3-10.

After the programmer has clocked out the Application ID Word, it must be inspected. If the Application ID has the value, BBh, the programming executive 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 programming executive is not resident in memory; it must be loaded to memory before the device can be programmed. The procedure for loading the programming executive to memory is described in **Section 5.4 "Programming the Programming Executive to Memory"**.

3.12 Exiting ICSP Mode

Exiting Program/Verify mode is done by removing V_{IH} from $\overline{MCLR}$, as shown in Figure 3-9. 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 V_{IH}.

FIGURE 3-9: EXITING ICSP™ MODE



4.0 DEVICE PROGRAMMING – ENHANCED ICSP

This section discusses programming the device through Enhanced ICSP and the programming executive. The programming executive resides in executive memory (separate from code memory) and is executed when Enhanced ICSP Programming mode is entered. The programming executive provides the mechanism for the programmer (host device) to program and verify the PIC24FJXXXGA0XX devices using a simple command set and communication protocol. There are several basic functions provided by the programming executive:

- Read Memory
- Erase Memory
- Program Memory
- Blank Check
- Read Executive Firmware Revision

The programming executive 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. Table 4-1 summarizes the commands. A detailed description for each command is provided in **Section 5.2 "Programming Executive Commands"**.

TABLE 4-1: COMMAND SET SUMMARY

Command	Description
SCHECK	Sanity Check
READC	Read Device ID Registers
READP	Read Code Memory
PROGP	Program One Row of Code Memory and Verify
PROGW	Program One Word of Code Memory and Verify
QBLANK	Query if the Code Memory is Blank
QVER	Query the Software Version

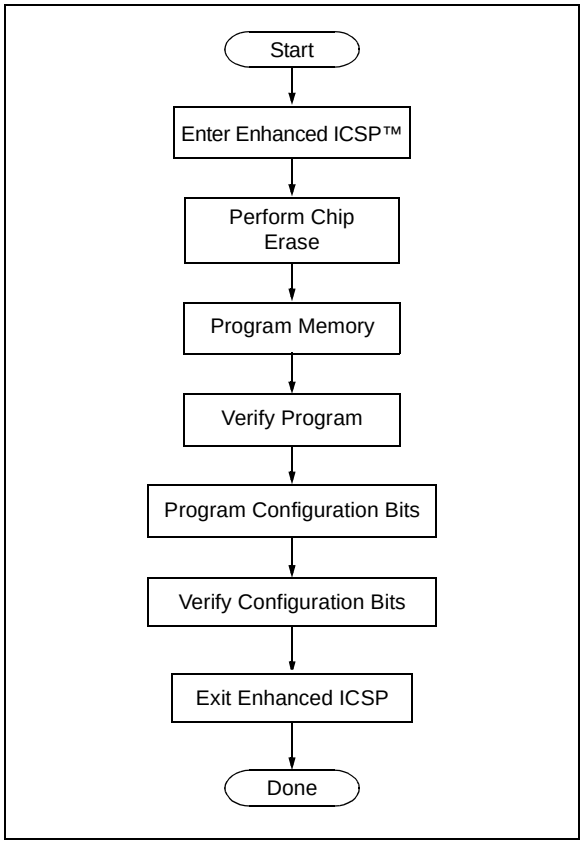
The programming executive uses the device's data RAM for variable storage and program execution. After the programming executive has run, no assumptions should be made about the contents of data RAM.

4.1 Overview of the Programming Process

Figure 4-1 shows the high-level overview of the programming process. After entering Enhanced ICSP mode, the programming executive is verified. Next, the device is erased. Then, the code memory is programmed, followed by the configuration locations. Code memory (including the Configuration registers) is then verified to ensure that programming was successful.

After the programming executive has been verified in memory (or loaded if not present), the PIC24FJXXXGA0XX family can be programmed using the command set shown in Table 4-1.

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



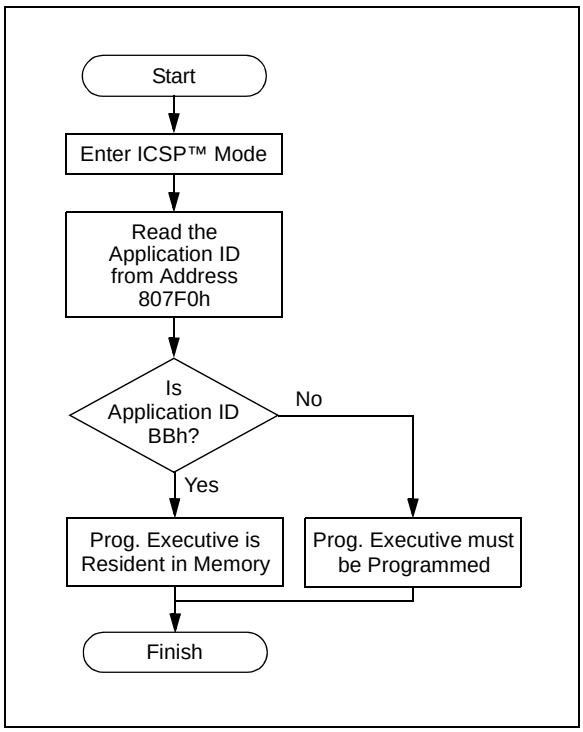
4.2 Confirming the Presence of the Programming Executive

Before programming can begin, the programmer must confirm that the programming executive is stored in executive memory. The procedure for this task is shown in Figure 4-2.

First, In-Circuit Serial Programming mode (ICSP) is entered. Then, the unique Application ID Word stored in executive memory is read. If the programming executive is resident, the Application ID Word is BBh, which means programming can resume as normal. However, if the Application ID Word is not BBh, the programming executive must be programmed to executive code memory using the method described in **Section 5.4 "Programming the Programming Executive to Memory"**.

Section 3.0 "Device Programming – ICSP" describes the ICSP programming method. **Section 3.11 "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



4.3 Entering Enhanced ICSP Mode

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

1. The $\overline{\text{MCLR}}$ pin is briefly driven high, 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 programming voltage applied to $\overline{\text{MCLR}}$ is V_{IH}, which is essentially V_{DD} in the case of PIC24FJXXXGA0XX devices. There is no minimum time requirement for holding at V_{IH}. After V_{IH} is removed, an interval of at least P18 must elapse before presenting the key sequence on PGDx.

The key sequence is a specific 32-bit pattern: '0100 1101 0100 0011 0100 1000 0101 0000' (more easily remembered as 4D434850h in hexa-decimal 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, V_{IH} must be applied to $\overline{\text{MCLR}}$ and held at that level for as long as Program/Verify mode is to be maintained. An interval of at least time P19 and P7 must elapse before presenting data on PGDx. Signals appearing on PGDx before P7 has elapsed will not be interpreted as valid.

On successful entry, the program memory can be accessed and programmed in serial fashion. While in the Program/Verify mode, all unused I/Os are placed in the high-impedance state.

FIGURE 4-3: ENTERING ENHANCED ICSP™ MODE

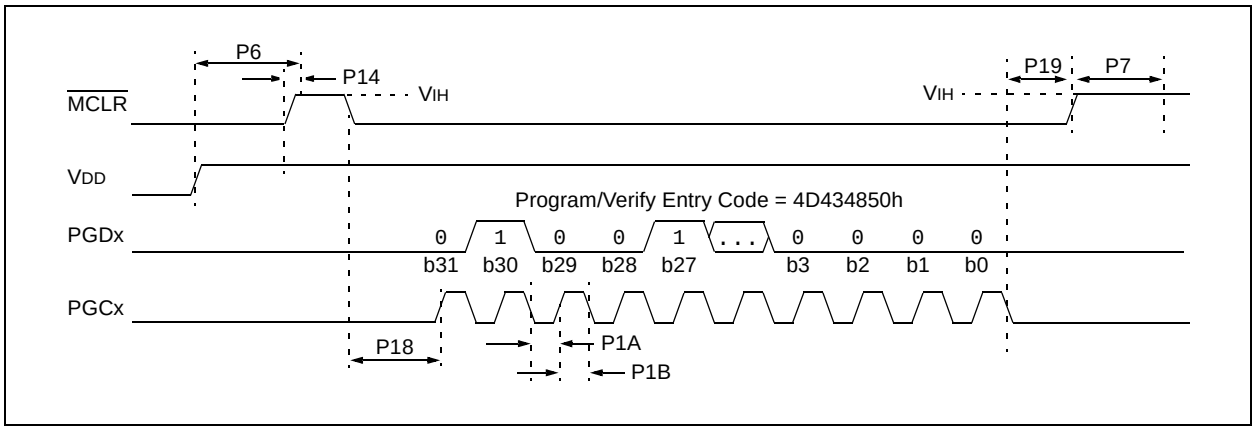


TABLE 5-1: PROGRAMMING EXECUTIVE COMMAND SET

Opcode	Mnemonic	Length (16-bit words)	Time-out	Description
0h	SCHECK	1	1 ms	Sanity check.
1h	READC	3	1 ms	Read an 8-bit word from the specified Device ID register.
2h	READP	4	1 ms/row	Read N 24-bit instruction words of code memory starting from the specified address.
3h	RESERVED	N/A	N/A	This command is reserved. It will return a NACK.
4h	PROGC	4	5 ms	Write an 8-bit word to the specified Device ID registers.
5h	PROGP	99	5 ms	Program one row of code memory at the specified address, then verify.(1)
7h	RESERVED	N/A	N/A	This command is reserved. It will return a NACK.
8h	RESERVED	N/A	N/A	This command is reserved. It will return a NACK.
9h	RESERVED	N/A	N/A	This command is reserved. It will return a NACK.
Ah	QBLANK	3	TBD	Query if the code memory is blank.
Bh	QVER	1	1 ms	Query the programming executive software version.
Dh	PROGW	4	5 ms	Program one instruction word of code memory at the specified address, then verify.

Legend: TBD = To Be Determined

Note 1: One row of code memory consists of (64) 24-bit words. Refer to Table 2-3 for device-specific information.

5.2.4 COMMAND DESCRIPTIONS

All commands supported by the programming executive are described in **Section 5.2.5 “SCHECK Command”** through **Section 5.2.12 “QVER Command”**.

5.2.5 SCHECK COMMAND

15 12	11 0
Opcode	Length

Field	Description
Opcode	0h
Length	1h

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

Expected Response (2 words):

1000h

0002h

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

5.2.6 READC COMMAND

15 12	11 8	7 0
Opcode	Length	
N		Addr_MSB
Addr_LS		

Field	Description
Opcode	1h
Length	3h
N	Number of 8-bit Device ID registers to read (max. 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 programming executive to read N 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 Device ID registers, the upper byte in every data word returned by the programming executive is 00h and the lower byte contains the Device ID register value.

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

1100h

2 + N

Device ID Register 1

...

Device ID Register N

Note:	Reading unimplemented memory will cause the programming executive to reset. Please ensure that only memory locations present on a particular device are accessed.

5.2.7 READP COMMAND

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

Field	Description
Opcode	2h
Length	4h
N	Number of 24-bit instructions to read (max. of 32768)
Reserved	0h
Addr_MSB	MSB of 24-bit source address
Addr_LS	Least Significant 16 bits of 24-bit source address

The READP command instructs the programming executive to read N 24-bit words of code memory, including Configuration Words, starting from the 24-bit address specified by Addr_MSB and Addr_LS. This command can only be used to read 24-bit data. All data returned in response to this command uses the packed data format described in **Section 5.2.2 “Packed Data Format”**.

Expected Response (2 + 3 * N/2 words for N even):

1200h

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):

1200h

4 + 3 * (N – 1)/2

Least significant program memory word 1

...

MSB of program memory word N (zero padded)

Note:	Reading unimplemented memory will cause the programming executive to reset. Please ensure that only memory locations present on a particular device are accessed.

5.4 Programming the Programming Executive to Memory

5.4.1 OVERVIEW

If it is determined that the programming executive is not present in executive memory (as described in **Section 4.2 "Confirming the Presence of the Programming Executive"**), it must be programmed into executive memory using ICSP, as described in **Section 3.0 "Device Programming – ICSP"**.

Storing the programming executive to executive memory is similar to normal programming of code memory. Namely, the executive memory must be erased, and then the programming executive must be programmed 64 words at a time. Erasing the last page of executive memory will cause the FRC oscillator calibration settings and device diagnostic data in the Diagnostic and Calibration Words, at addresses 8007F0h to 8007FEh, to be erased. In order to retain this calibration, these memory locations should be read and stored prior to erasing executive memory. They should then be reprogrammed in the last words of program memory. This control flow is summarized in Table 5-5.

TABLE 5-5: PROGRAMMING THE PROGRAMMING EXECUTIVE

Command (Binary)	Data (Hex)	Description
Step 1: Exit Reset vector and erase executive memory.		
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
Step 2: Initialize pointers to read Diagnostic and Calibration Words for storage in W6-W13.		
0000	200800	MOV #0x80, W0
0000	880190	MOV W0, TBLPAG
0000	207F00	MOV #0x07F0, W1
0000	2000C2	MOV #0xC, W2
0000	000000	NOP
Step 3: Repeat this step 8 times to read Diagnostic and Calibration Words, storing them in W registers, W6-W13.		
0000	BA1931	TBLRDL [W1++].[W2++]
0000	000000	NOP
0000	000000	NOP
Step 4: Initialize the NVMCON to erase executive memory.		
0000	240420	MOV #0x4042, W0
0000	883B00	MOV W0, NVMCON
Step 5: Initialize Erase Pointers to first page of executive and then initiate the erase cycle.		
0000	200800	MOV #0x80, W0
0000	880190	MOV W0, TBLPAG
0000	200001	MOV #0x0, W1
0000	000000	NOP
0000	BB0881	TBLWTL W1, [W1]
0000	000000	NOP
0000	000000	NOP
0000	A8E761	BSET NVMCON, #15
00000	000000	NOP
0000	000000	NOP
Step 6: Repeat this step to poll the WR bit (bit 15 of NVMCON) until it is cleared by the hardware.		
0000	040200	GOTO 0x200
0000	000000	NOP
0000	803B02	MOV NVMCON, W2
0000	883C22	MOV W2, VISI
0001	000000	NOP
	<VISI>	Clock out contents of the VISI register.
0000	000000	NOP

5.4.2 PROGRAMMING VERIFICATION

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

Reading the contents of executive memory can be performed using the same technique described in **Section 3.8 "Reading Code Memory"**. A procedure for reading executive memory is shown in Table 5-6. Note that in Step 2, the TBLPAG register is set to 80h, such that executive memory may be read. The last eight words of executive memory should be verified with stored values of the Diagnostic and Calibration Words to ensure accuracy.

TABLE 5-6: READING EXECUTIVE MEMORY

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
Step 2: Initialize TBLPAG and the Read Pointer (W6) for TBLRD instruction.		
0000	200800	MOV #0x80, W0
0000	880190	MOV W0, TBLPAG
0000	EB0300	CLR W6
Step 3: Initialize the Write Pointer (W7) to point to the VISI register.		
0000	207847	MOV #VISI, W7
0000	000000	NOP
Step 4: Read and clock out the contents of the next two locations of executive memory through the VISI register using the REGOUT command.		
0000	BA0B96	TBLRDL [W6], [W7]
0000	000000	NOP
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	BADBB6	TBLRDH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BAD3D6	TBLRDH.B [++W6], [W7--]
0000	000000	NOP
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	BA0BB6	TBLRDL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
Step 5: Reset the device internal PC.		
0000	040200	GOTO 0x200
0000	000000	NOP
Step 6: Repeat Steps 4 and 5 until all desired code memory is read.		

7.0 AC/DC CHARACTERISTICS AND TIMING REQUIREMENTS

Standard Operating Conditions
Operating Temperature: 0°C to +70°C. Programming at +25°C is recommended.

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
D111	VDD	Supply Voltage During Programming	VDDCORE + 0.1	3.60	V	Normal programming(1,2)
D112	IPP	Programming Current on $\overline{MCLR}$	—	5	μA	
D113	IDDP	Supply Current During Programming	—	2	mA	
D031	VIL	Input Low Voltage	VSS	0.2 VDD	V	
D041	VIH	Input High Voltage	0.8 VDD	VDD	V	
D080	VOL	Output Low Voltage	—	0.4	V	IOL = 8.5 mA @ 3.6V
D090	VOH	Output High Voltage	3.0	—	V	IOH = -3.0 mA @ 3.6V
D012	CIO	Capacitive Loading on I/O pin (PGDx)	—	50	pF	To meet AC specifications
D013	CF	Filter Capacitor Value on VCAP	4.7	10	μF	Required for controller core
P1	TPGC	Serial Clock (PGCx) Period	100	—	ns	
P1A	TPGCL	Serial Clock (PGCx) Low Time	40	—	ns	
P1B	TPGCH	Serial Clock (PGCx) High Time	40	—	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 4-Bit Command Operand and Next 4-Bit Command	40	—	ns	
P5	TDLY2	Delay Between Last PGCx ↓ of Command Byte to First PGCx ↑ of Read of Data Word	20	—	ns	
P6	TSET2	VDD ↑ Setup Time to $\overline{MCLR}$ ↑	100	—	ns	
P7	THLD2	Input Data Hold Time from $\overline{MCLR}$ ↑	25	—	ms	
P8	TDLY3	Delay Between Last PGCx ↓ of Command Byte to PGDx ↑ by Programming Executive	12	—	μs	
P9	TDLY4	Programming Executive Command Processing Time	40	—	μs	
P10	TDLY6	PGCx Low Time After Programming	400	—	ns	
P11	TDLY7	Chip Erase Time	400	—	ms	
P12	TDLY8	Page Erase Time	40	—	ms	
P13	TDLY9	Row Programming Time	2	—	ms	
P14	TR	$\overline{MCLR}$ 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 $\overline{MCLR}$ ↓	0	—	s	
P17	THLD3	$\overline{MCLR}$ ↓ to VDD ↓	100	—	ns	
P18	TKEY1	Delay from First $\overline{MCLR}$ ↓ to First PGCx ↑ for Key Sequence on PGDx	40	—	ns	
P19	TKEY2	Delay from Last PGCx ↓ for Key Sequence on PGDx to Second $\overline{MCLR}$ ↑	1	—	ms	
P20	TDLY11	Delay Between PGDx ↓ by Programming Executive to PGDx Driven by Host	23	—	μs	
P21	TDLY12	Delay Between Programming Executive Command Response Words	8	—	ns	

Note 1: VDDCORE must be supplied to the VDDCORE/VCAP pin if the on-chip voltage regulator is disabled. See **Section 2.1 "Power Requirements"** for more information. (Minimum VDDCORE allowing Flash programming is 2.25V.)

2: 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.