PIC24FXXKMXXX/KLXXX Flash Programming Specifications

1.0 DEVICE OVERVIEW

This document defines the programming specifications for the PIC24FXXKMXXX/KLXXX family of 16-bit microcontroller devices. This is required only for developing programming support for the PIC24FXXKMXXX/KLXXX family. Users of any one of these devices should use the development tools that are already supporting the device programming.

The programming specifications are specific to the following devices:

- PIC24F04KL100
- PIC24F04KL101
- PIC24F08KL200
- PIC24F08KL201
- PIC24F08KL301
- PIC24FV08KM101(1)
- PIC24FV08KM204(1)
- PIC24FV16KM104(1)
- PIC24FV08KM102(1)
- PIC24F08KL302
- PIC24F08KL401
- PIC24F08KL402
- PIC24F16KL401
- PIC24F16KL402
- PIC24FV16KM202(1)
- PIC24FV08KM202(1)
- PIC24FV16KM102(1)
- PIC24FV16KM204(1)

Note 1: Includes corresponding PIC24FXXKMXXX devices.

2.0 PROGRAMMING OVERVIEW OF THE PIC24FXXKMXXX/KLXXX FAMILY

PIC24FXXKMXXX/KLXXX family devices are programmed exclusively using In-Circuit Serial Programming™ (ICSP™).

The ICSP programming method is the most direct method for programming the device. It provides a native, low-level programming capability to erase, program and verify the device. **Section 3.0 "Device Programming – ICSP"** describes the ICSP method.

2.1 Power Requirements

Devices in the PIC24FXXKLXXX and PIC24FXXKMXXX families are 3.3V supply designs. The devices can operate from 1.8V to 3.6V.

Devices in the PIC24FVXXKMXXX families are 5.0V supply designs. The devices can operate from 2.0V to 5.5V; an internal regulator operates the core logic at 3.25V.

Table 2.1 provides the pins that are required for programming, which are indicated in Figure 2-2. Refer to the specific device data sheet for complete pin descriptions. Note that all power supply and ground pins must be connected appropriately for programming.

TABLE 2-1: PIN DESCRIPTIONS (DURING PROGRAMMING)

Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
$\overline{MCLR}$/VPP	$\overline{MCLR}$/VPP	P	Programming Enable
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground
CEFC	VCAP	P	Stabilizing Capacitor for Voltage Regulator(1)
PGECx	PGEC	I	Programming Pin Pair: Serial Clock
PGEDx	PGED	I/O	Programming Pin Pair: Serial Data

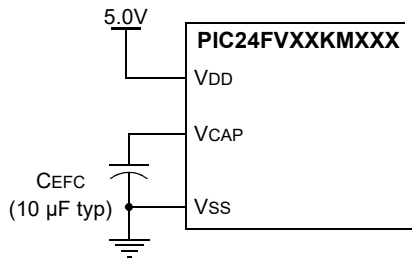
Legend: I = Input, O = Output, P = Power

Note 1: PIC24FVXXKMXXX devices only.

2.1.1 ON-CHIP VOLTAGE REGULATOR CONNECTIONS

For PIC24FVXXKMXXX devices, an on-chip regulator provides power to the core from the other VDD pins. A low-ESR capacitor (such as high-quality ceramic or tantalum) must be connected to the VDDCORE pin (Figure 2-1). This helps to maintain the stability of the regulator. The specifications for core voltage and capacitance are listed in **Section 5.0 "AC/DC Characteristics and Timing Requirements"**. PIC24FXXKMXXX and PIC24FXXKLXXX devices do not use an on-chip regulator.

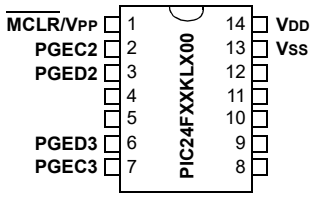
FIGURE 2-1: CONNECTIONS FOR THE ON-CHIP REGULATOR



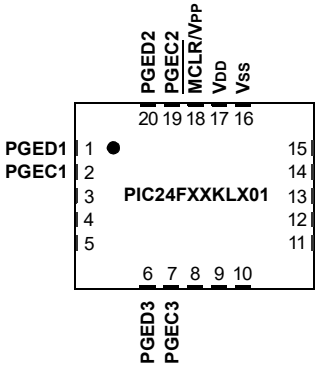
Note 1: These are typical operating voltages. Refer to **Section 5.0 "AC/DC Characteristics and Timing Requirements"** for the full operating ranges of VDD and VDDCORE.

FIGURE 2-2: PIC24FXXKLXXX PIN DIAGRAMS

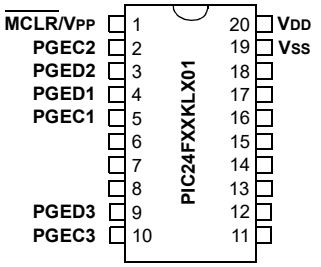
14-Pin PDIP



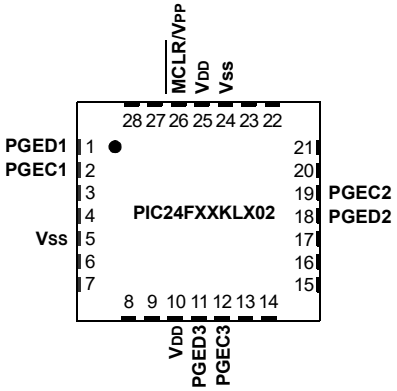
20-Pin QFN



20-Pin PDIP



28-Pin QFN



28-Pin PDIP

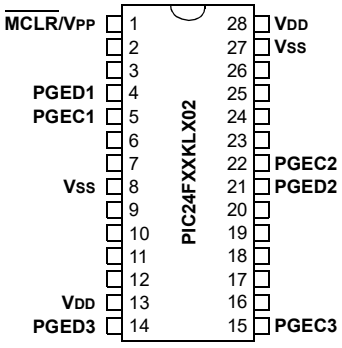
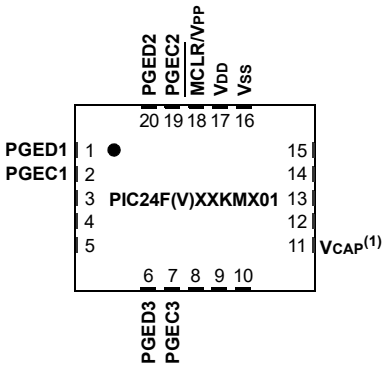
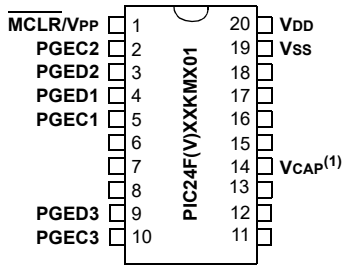


FIGURE 2-3: PIC24F(V)XXKMXXX PIN DIAGRAMS

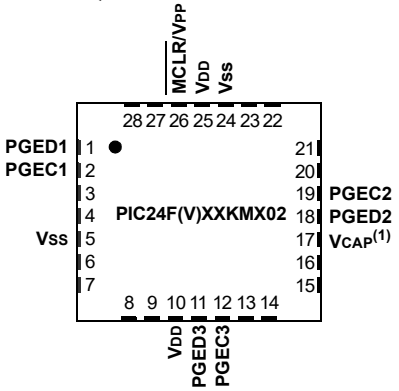
20-Pin QFN



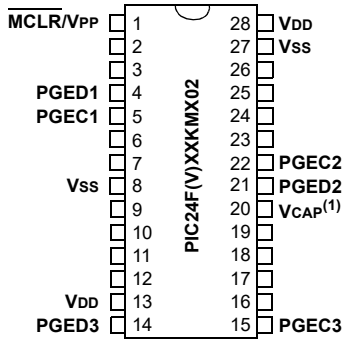
20-Pin PDIP



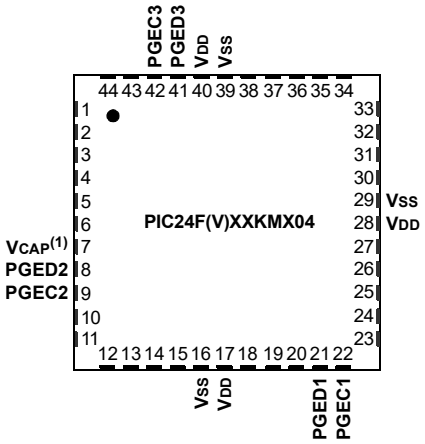
28-Pin QFN



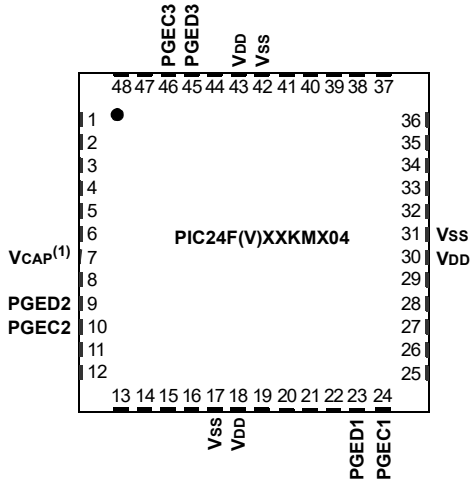
28-Pin PDIP



44-Pin QFN/TQFP



48-Pin QFN



Note 1: PIC24FVXXKMXXX devices only. PIC24FXXKMXXX devices do not use VCAP.

2.2 Memory Map

The program memory map for the PIC24FXXKMXXX/KLXXX devices extends from 000000h to FFFFFEh. Code storage is located at the base of the memory map and supports up to 5.5K instruction words (about 16 Kbytes).

Additionally, PIC24FXXKMXXX/KLXXX family devices have an on-chip data EEPROM. This data EEPROM is mapped to the program memory area from location, 7FFE00h to 7FFFFEh.

Table 2-2 provides the program memory and data EEPROM size, and the number of memory rows present in each device variant.

The erase operation can be done on one word, half of a row or one row at a time. The program operation can be done only one word at a time.

TABLE 2-2: MEMORY SIZES FOR PIC24FXXKMXXX/KLXXX DEVICES

Device	Program Memory Upper Address (Instruction Words)	Flash Rows	Data EEPROM Size (Words)	Data EEPROM Rows
PIC24F16KL4XX	2BFEh (5.5K)	176	256	32
PIC24F08KL4XX	15FEh (2.75K)	88	256	32
PIC24F08KL3XX	15FEh (2.75K)	88	128	16
PIC24F08KL2XX	15FEh (2.75K)	88	—	—
PIC24F04KL1XX	0AFEh (1.375K)	44	—	—
PIC24FV16KM204	2BFEh (5.5K)	176	256	32
PIC24FV16KM202	2BFEh (5.5K)	176	256	32
PIC24FV08KM204	15FEh (2.75K)	88	256	32
PIC24FV08KM202	15FEh (2.75K)	88	256	32
PIC24FV16KM104	2BFEh (5.5K)	176	256	32
PIC24FV16KM102	2BFEh (5.5K)	176	256	32
PIC24FV08KM102	15FEh (2.75K)	88	256	32
PIC24FV08KM101	15FEh (2.75K)	88	256	32
PIC24F16KM204	2BFEh (5.5K)	176	256	32
PIC24F16KM202	2BFEh (5.5K)	176	256	32
PIC24F08KM204	15FEh (2.75K)	88	256	32
PIC24F08KM202	15FEh (2.75K)	88	256	32
PIC24F16KM104	2BFEh (5.5K)	176	256	32
PIC24F16KM102	2BFEh (5.5K)	176	256	32
PIC24F08KM102	15FEh (2.75K)	88	256	32
PIC24F08KM101	15FEh (2.75K)	88	256	32

Locations, 800000h through 8007FEh, are reserved for executive code memory. This region stores the debugging executive and the Diagnostic Words. The debug executive is used for in-circuit debugging. This region of memory cannot be used to store user code.

The device Configuration registers are implemented from location, F80000h to F80010h, and can be erased or programmed, one register at a time. Table 2-3 provides the implemented Configuration registers and their locations.

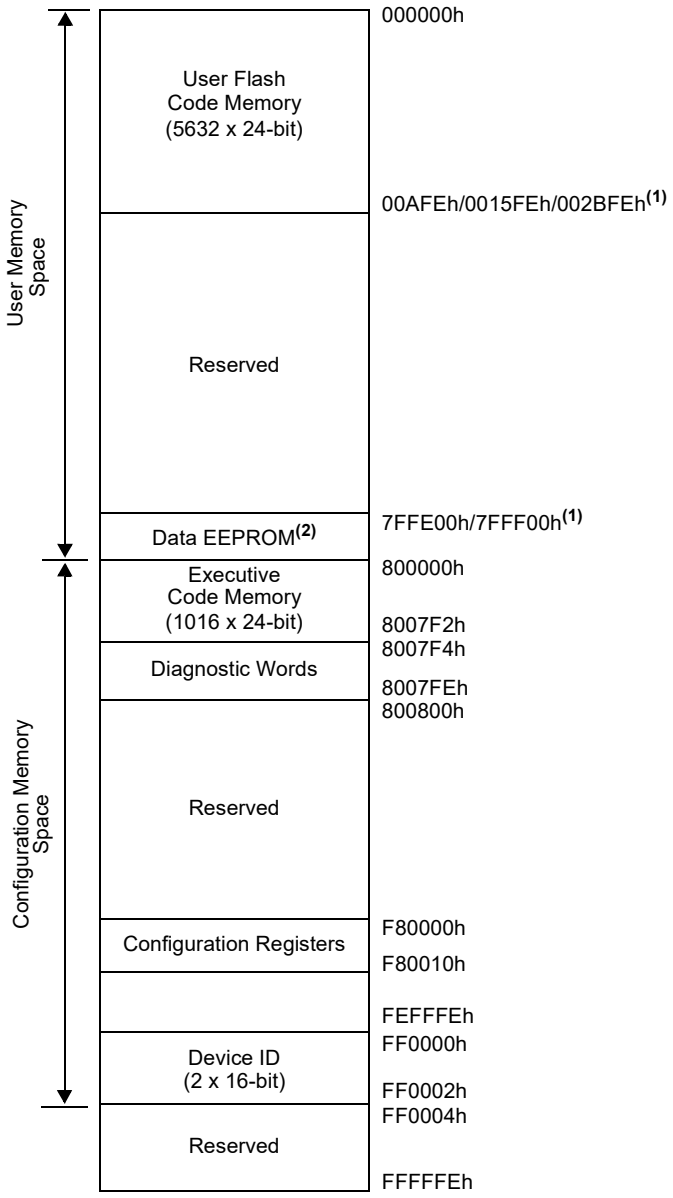
Locations, FF0000h and FF0002h, are reserved for the Device ID registers. These bits can be used by the programmer to identify the device type that is being programmed. For more information, see **Section 4.0 "Device ID"**. The Device ID registers read out normally, even after code protection is applied.

Figure 2-4 illustrates the memory map for the PIC24FXXKMXXX/KLXXX family variants.

TABLE 2-3: CONFIGURATION REGISTER LOCATIONS

Configuration Register	Address
FBS	F80000
FGS	F80004
FOSCSEL	F80006
FOSC	F80008
FWDT	F8000A
FPOR	F8000C
FICD	F8000E

FIGURE 2-4: PROGRAM MEMORY MAP



Note 1: The upper address boundaries for user program memory are device-specific. See Table 2-2 for details.
2: PIC24F08KL2XX and PIC24F04KL1XX devices do not have a data EEPROM section.

3.0 DEVICE PROGRAMMING – ICSP

The ICSP method is a special programming protocol that allows reading and writing to the PIC24FXXKMXXX/KLXXX device family memory. ICSP is the most direct method used to program a device. This is accomplished by applying control codes and instructions, serially to the device, using the PGECx and PGEDx pins.

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

Note:	During ICSP operation, the operating frequency of PGECx should not exceed 8 MHz.

3.1 Overview of the Programming Process

Figure 3-1 illustrates the high-level overview of the programming process.

After entering the ICSP mode, perform the following:

1. Bulk Erase the device.
2. Program and verify the code memory.
3. Program and verify the data EEPROM memory.
4. Program and verify the device configuration.
5. Program the code-protect Configuration bits, if required.

3.2 ICSP Operation

Upon entry into ICSP mode, the CPU is Idle. An internal state machine governs the execution of the CPU. A 4-bit control code is clocked in, using PGECx and PGEDx, 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 via the VISI register.

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

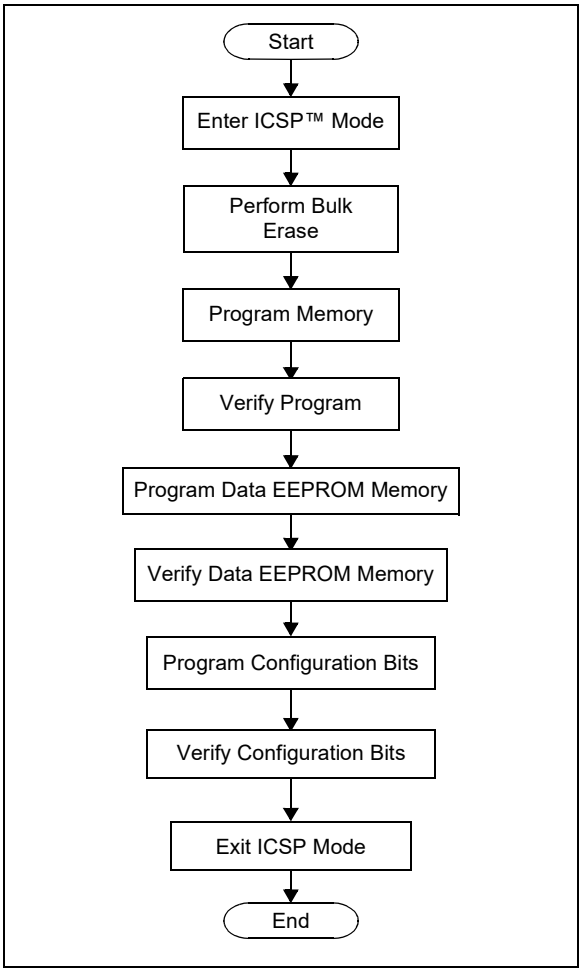


TABLE 3-1: CPU CONTROL CODES IN ICSP™ MODE

4-Bit Control Code	Mnemonic	Description
0000b	SIX	Shift in 24-bit instruction and execute.
0001b	REGOUT	Shift out the VISI (0784h) register.
0010b-1111b	N/A	This is reserved.

3.2.1 SIX SERIAL INSTRUCTION EXECUTION

The SIX control code allows execution of the PIC24FXXKMXXX/KLXXX 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 PGECx 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 PGECx clocks are needed on start-up, thereby resulting in a 9-bit SIX command instead of the normal 4-bit SIX command.

After the forced SIX is clocked in, the ICSP operation returns to 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 codes in this specification begin with a NOP to ensure that no data are lost.

3.2.1.1 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 CPU clocks required 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 are 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.

3.2.2 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 PGEDx pin. After the REGOUT control code is received, the CPU is held Idle for eight cycles. After this, an additional 16 cycles are required to clock the data out (see Figure 3-3).

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

Note 1:	After the contents of VISI are shifted out, the PIC24FXXKMXXX/KLXXX devices maintain PGEDx as an output until the first rising edge of the next clock is received.
2:	Data change on the falling edge and latch on the rising edge of PGECx. For all data transmissions, the Least Significant bit (LSb) is transmitted first.

FIGURE 3-2: SIX SERIAL EXECUTION

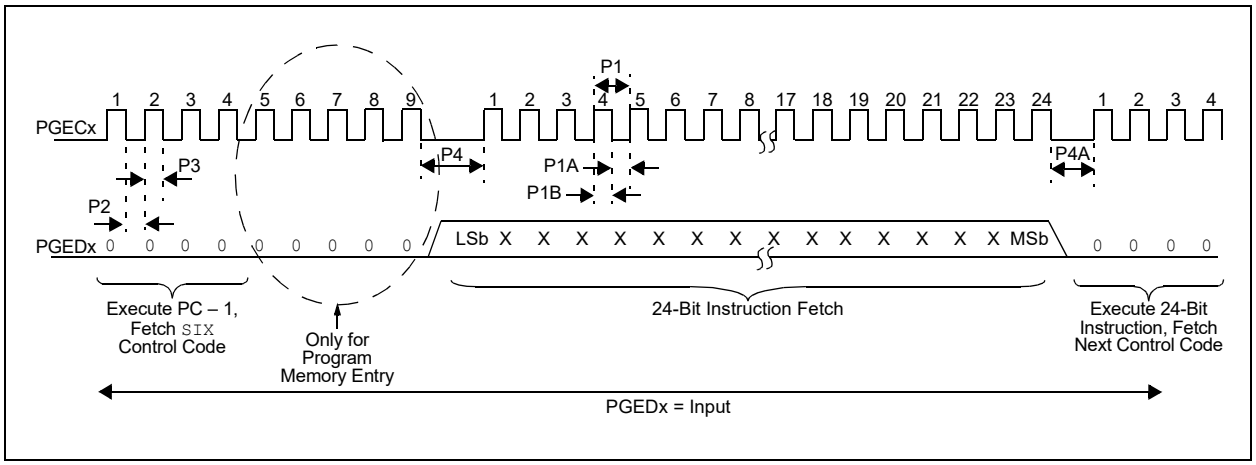
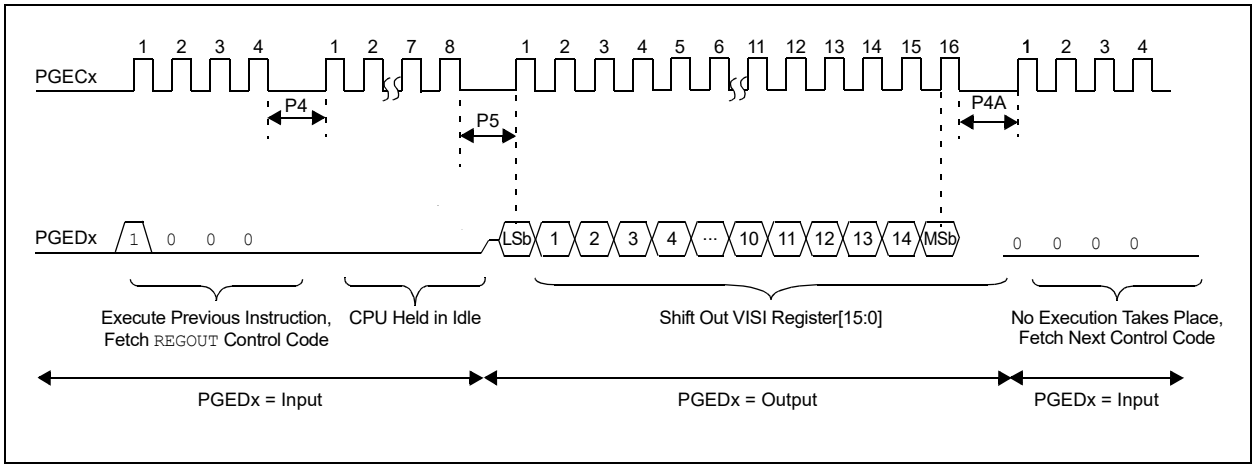


FIGURE 3-3: REGOUT SERIAL EXECUTION



3.3 Entering ICSP Mode

3.3.1 LOW-VOLTAGE ICSP ENTRY

As illustrated in Figure 3-4, the following processes are involved in entering ICSP Program/Verify mode using MCLR:

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

The programming voltage, VIH, is applied to MCLR; this is VDD in the case of PIC24FXXKMXXX/KLXXX 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 PGEDx.

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 the Program/Verify mode is to be maintained. An interval of at least P19 and P7 must elapse before presenting data on PGEDx. Signals appearing on PGECx, before P7 has elapsed, would not be interpreted as valid.

3.3.2 HIGH-VOLTAGE ICSP ENTRY

Entering the ICSP Program/Verify mode, using the VPP pin, is the same as entering the mode using MCLR. The only difference is the programming voltage applied to VPP is VIHH, and before presenting the key sequence on PGEDx, an interval of at least P18 should elapse (see Figure 3-5).

Once the key sequence is complete, an interval of at least P7 should elapse and the voltage should remain at VIHH. The voltage, VIHH, must be held at that level for as long as the Program/Verify mode is to be maintained. An interval of at least P7 must elapse before presenting the data on PGEDx.

Signals appearing on PGEDx before P7 has elapsed will not be interpreted as valid.

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

3.3.3 CODE-PROTECT ICSP ENTRY

When code protection is employed on the PIC24FXXKMXXX/KLXXX devices (BWRP, GSS0 or GWRP = 0), then the voltage on VDD must be above VBULK in order to erase, and then program, the device. Care must be taken in the design and layout of a board so that any parts connected to VDD can withstand what may be an increase in voltage if the device is running below VBULK.

FIGURE 3-4: ENTERING ICSP™ MODE USING LOW-VOLTAGE ENTRY

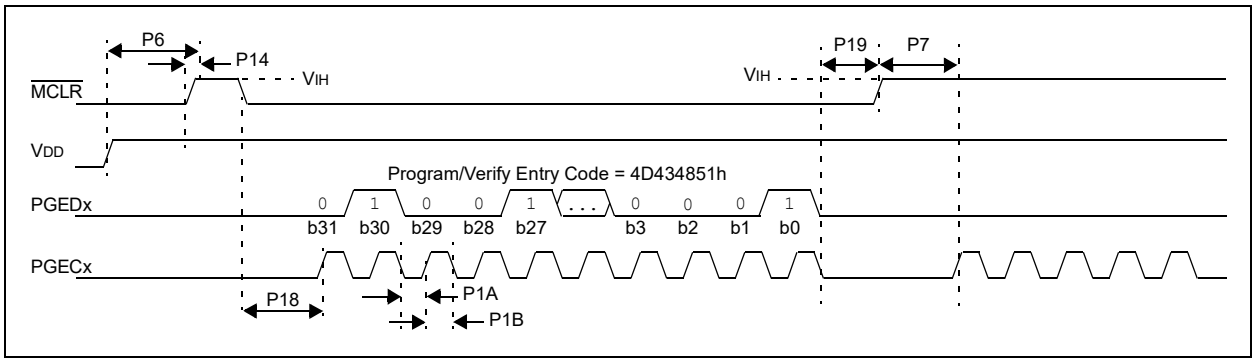
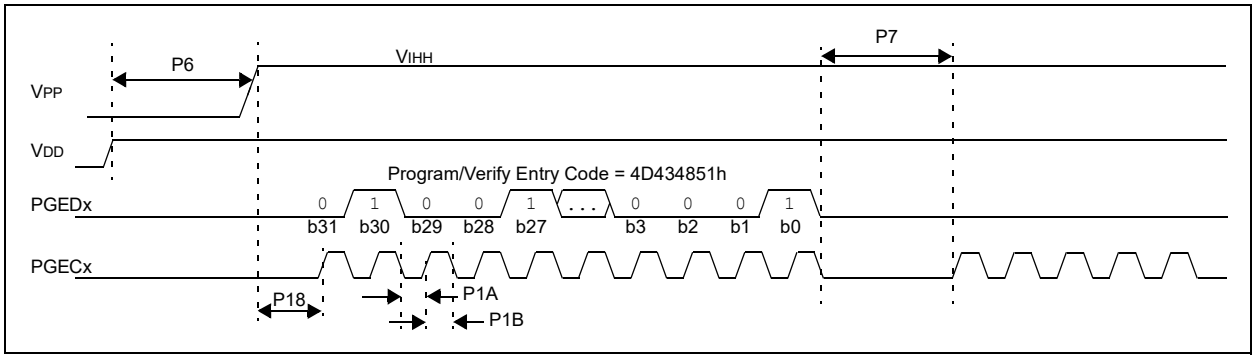


FIGURE 3-5: ENTERING ICSP™ MODE USING HIGH-VOLTAGE ENTRY



3.4 Flash Memory Programming in ICSP Mode

3.4.1 PROGRAMMING OPERATIONS

The NVMCON register controls the Flash memory write and erase operations. To program the device, set the NVMCON register to select the type of erase operation (see Table 3-2) or write operation (see Table 3-3). Set the WR control bit (NVMCON[15]) to initiate the program.

In ICSP mode, all programming operations are self-timed. There is an internal delay between setting and automatic clearing of the WR control bit when the programming operation is complete. Refer to **Section 5.0 "AC/DC Characteristics and Timing Requirements"** for information on the delays associated with various programming operations.

3.4.2 STARTING AND STOPPING A PROGRAMMING CYCLE

The WR bit (NVMCON[15]) is used to start an erase or write cycle. Initiate the programming cycle by setting the WR bit.

All erase and write cycles are self-timed. The WR bit should be polled to determine if the erase or write cycle is complete. Start a programming cycle as follows:

```
BSET    NVMCON, #WR
```

TABLE 3-2: NVMCON VALUES FOR ERASE OPERATIONS

NVMCON Value	Erase Operation
4064h	Erase the code memory and Configuration registers (does not erase executive code and Device ID registers).
404Ch	Erase the General Segment and Configuration bits associated with it.
4068h	Erase the Boot Segment and Configuration bits associated with it.
405Ah(1)	Erase four rows of code memory.
4059h(1)	Erase two rows of code memory.
4058h(1)	Erase a row of code memory.
4050h	Erase the entire data EEPROM memory and Configuration bits associated with it.
405Ah(1)	Erase eight words of data EEPROM memory.
4059h(1)	Erase four words of data EEPROM memory.
4058h(1)	Erase one word of data EEPROM memory.
4054h	Erase all of the Configuration registers (except the code-protect Configuration bits).
4058h(1)	Erase all of the Configuration registers except for FBS and FGS.

Note 1: The destination address decides the region (code memory, data EEPROM memory or Configuration register) of the erased rows/words.

TABLE 3-3: NVMCON VALUES FOR WRITE OPERATIONS

NVMCON Value	Write Operation8
4004h(1)	Write one Configuration register.
4004h(1)	Program one row (32 instruction words) of code memory or executive memory.
4004h(1)	Program one word of data EEPROM memory.

Note 1: The destination address decides the region (code memory, data EEPROM memory or Configuration register) of the erased rows/words.

3.5 Erasing Program Memory

To erase the program memory (all of the code memory, data memory and Configuration bits, including the code-protect bits), set the NVMCON register to 4064h and then execute the programming cycle.

Figure 3-6 illustrates the ICSP programming process for Bulk Erase. This process includes the ICSP command code, which must be transmitted (for each instruction), LSb first, using the PGECx and PGEDx pins (see Figure 3-2).

Table 3-4 provides the steps for executing the serial instruction for the Bulk Erase mode.

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

FIGURE 3-6: BULK ERASE FLOW

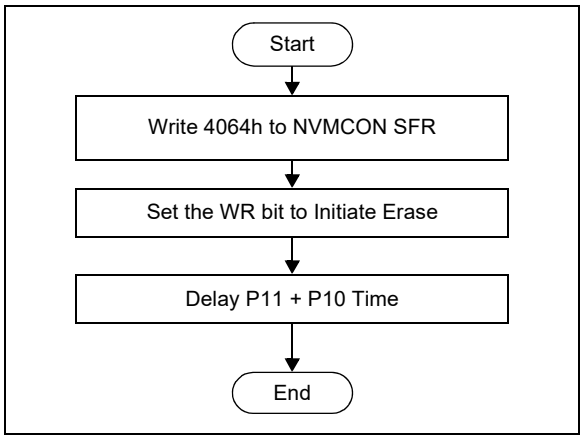


TABLE 3-4: SERIAL INSTRUCTION EXECUTION FOR CHIP ERASE

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
Step 2: Set the NVMCON register to erase the entire program memory.		
0000	24064A	MOV #0x4064, W10
0000	883B0A	MOV W10, NVMCON
Step 3: Set the TBLPAG and perform a dummy Table Write to select the erased memory.		
0000	200000	MOV #[PAGEVAL], W0
0000	880190	MOV W0, TBLPAG
0000	200000	MOV #0x0000, W0
0000	BB0800	TBLWTL W0, [W0]
0000	000000	NOP
0000	000000	NOP
Step 4: Initiate the erase cycle.		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
Step 5: Repeat this step to poll the WR bit (bit 15 of NVMCON) until it is cleared by the hardware.		
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
0000	803B02	MOV NVMCON, W2
0000	883C22	MOV W2, VISI
0000	000000	NOP
0001	[VISI]	Clock out the contents of the VISI register.
0000	000000	NOP

3.6 Writing Code Memory

The procedure for writing code memory is the same as writing the Configuration registers. The difference is that the 32 instruction words are programmed, one at a time. To facilitate this operation, Working registers, W0:W5, are used as temporary holding registers for the data to be programmed. Figure 3-8 illustrates the code memory writing flow.

Table 3-5 provides the ICSP programming details, including the serial pattern with the ICSP command code, which must be transmitted, LSb first, using the PGECx and PGEDx pins (see Figure 3-2).

In Step 1 of Table 3-5, the Reset vector is exited, in Step 2, the NVMCON register is initialized for programming a full row of code memory and in Step 3, the 24-bit starting destination address for programming is loaded into the TBLPAG and W7 registers. The upper byte of the starting destination address is stored in TBLPAG and the lower 16 bits of the destination address are stored in W7.

To minimize the programming time, a packed instruction format is used (see Figure 3-7).

In Step 4 of Table 3-5, four packed instruction words are stored in Working registers, W0:W5, using the `MOV` instruction. The Read Pointer, W6, is initialized. Figure 3-7 illustrates the contents of W0:W5, holding the packed instruction word data. In Step 5, eight `TBLWT` instructions are used to copy the data from W0:W5 to the write latches of the code memory. Since code memory is programmed in 32 instruction words at a time, Steps 3 to 5 are repeated eight times to load all the write latches (see Step 6).

After the write latches are loaded, initiate programming by writing to the NVMCON register in Steps 7 and 8. In Step 9, the internal PC is reset to 200h. This is a precautionary measure to prevent the PC from incrementing to unimplemented memory when large devices are being programmed. Finally, in Step 10, repeat Steps 3 through 9 until all of the code memory is programmed.

FIGURE 3-7: PACKED INSTRUCTION WORDS IN W0:W5

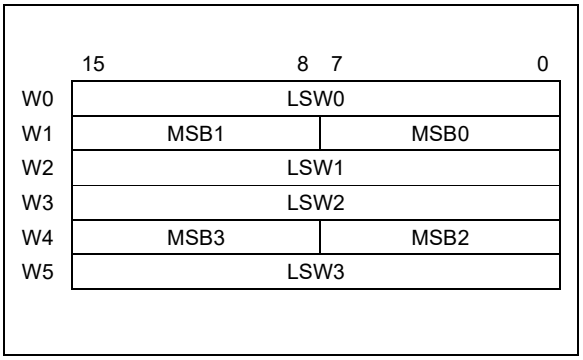


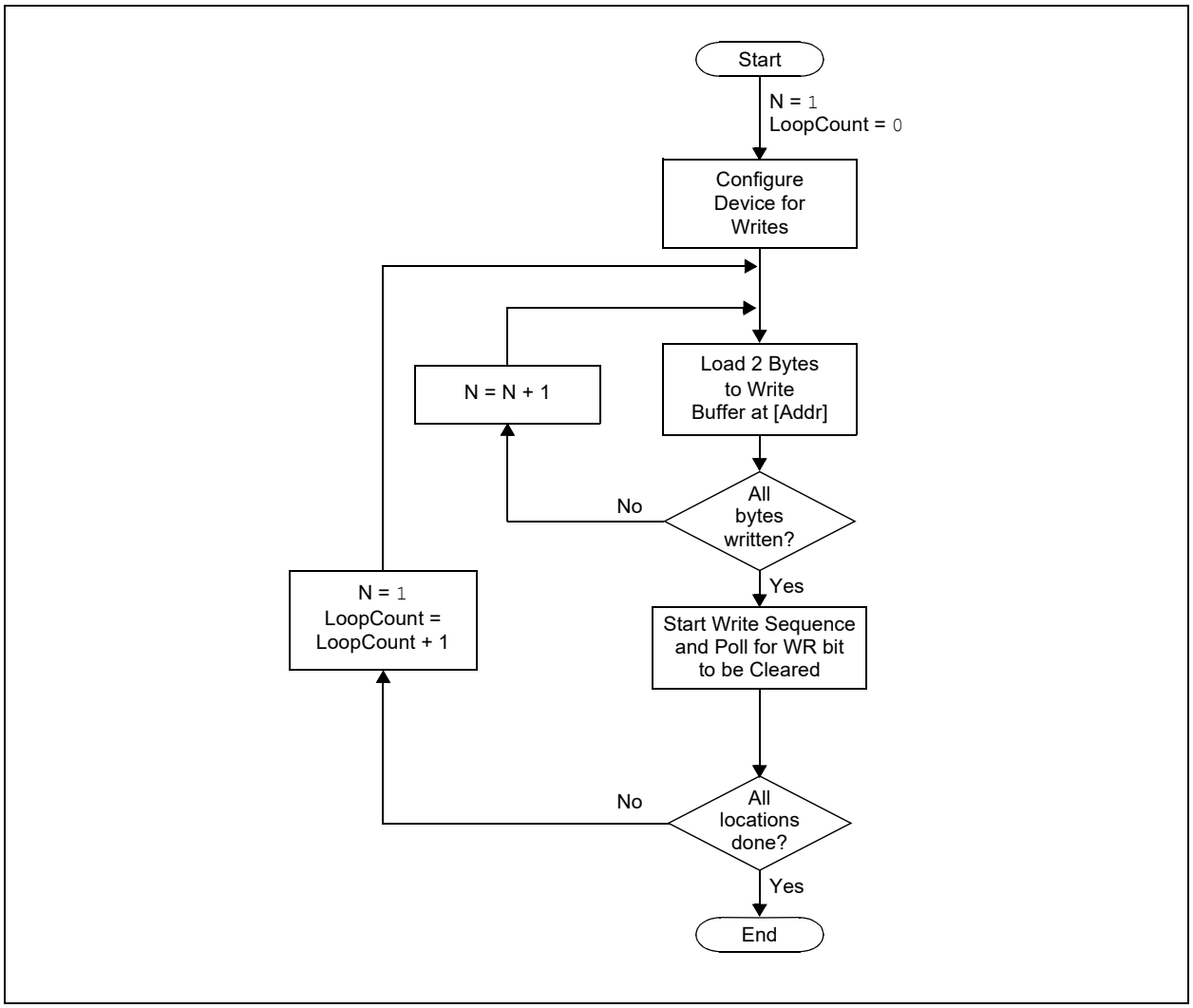
TABLE 3-5: SERIAL INSTRUCTION EXECUTION FOR WRITING CODE MEMORY

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
Step 2: Set the NVMCON to program 32 instruction words.		
0000	24004A	MOV #0x4004, W10
0000	883B0A	MOV W10, NVMCON
Step 3: Initialize the Write Pointer (W7) for the `TBLWT` instruction.		
0000	200xx0	MOV #[DestinationAddress23:16], W0
0000	880190	MOV W0, TBLPAG
0000	2xxxx7	MOV #[DestinationAddress15:0], W7
Step 4: Load W0:W5 with the next four instruction words to program.		
0000	2xxxx0	MOV #[LSW0], W0
0000	2xxxx1	MOV #[MSB1:MSB0], W1
0000	2xxxx2	MOV #[LSW1], W2
0000	2xxxx3	MOV #[LSW2], W3
0000	2xxxx4	MOV #[MSB3:MSB2], W4
0000	2xxxx5	MOV #[LSW3], W5

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 3 though 5, eight times, to load the write latches for 32 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 the device's internal PC.		
0000	040200	GOTO 0x200
0000	000000	NOP
Step 10: Repeat Steps 3 through 9 until the entire code memory is programmed.		

FIGURE 3-8: PROGRAM CODE MEMORY FLOW



3.7 Writing Data EEPROM

Figure 3-9 illustrates the flow of programming the data EEPROM memory. The procedure is the same as writing code memory. The only difference is that just one word is programmed in each operation. When writing data EEPROM, one word is programmed during each operation. Working register, W0, is used as a temporary holding register for the data to be programmed.

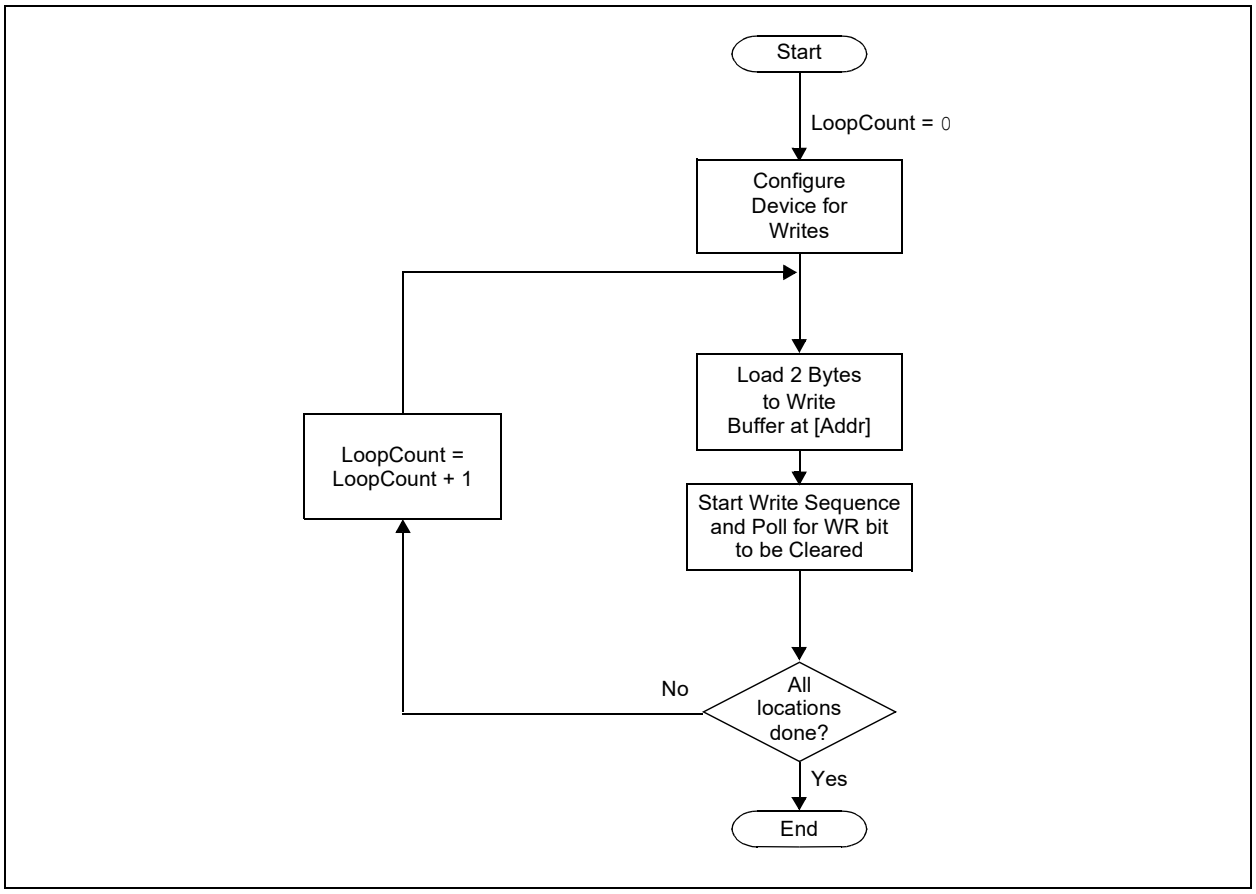
Table 3-6 provides the ICSP programming details for writing data EEPROM.

Note:	When writing to EEPROM, always set the TBLPAG register to 7Fh. This is the upper byte address of all locations of data EEPROM.

TABLE 3-6: INSTRUCTION EXECUTION FOR WRITING DATA EEPROM

Command (Binary)	Data (Hex)	Description
Step 1: Exit the Reset vector.		
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
Step 2: Set the NVMCON register to program one data word.		
0000	24004A	MOV #0x4004, W10
0000	883B0A	MOV W10, NVMCON
Step 3: Initialize the Write Pointer (W7) for the TBLWT instruction.		
0000	2007F0	MOV #0x7F, W0
0000	880190	MOV W0, TBLPAG
0000	2FExx7	MOV #[DestinationAddress15:0], W7
Step 4: Load W0 with the data word to program and load the write latch.		
0000	2xxxx0	MOV #[Data_Word_Value], W0
0000	BB1B80	TBLWTL W0, [W7++]
0000	000000	NOP
0000	000000	NOP
Step 5: Initiate the write cycle.		
0000	A8E761	BSET NVMCON, #WR
0000	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
0000	000000	NOP
00001	[VISI]	Clock out contents of the VISI register.
0000	000000	NOP
Step 7: Reset the device's internal PC.		
0000	040200	GOTO 0x200
0000	000000	NOP
Step 8: Repeat Steps 4 through 7 until the entire data EEPROM memory is programmed.		

FIGURE 3-9: PROGRAM DATA EEPROM MEMORY FLOW



3.8 Writing Configuration Registers

The procedure for writing the Configuration registers is the same as for writing code memory. The only difference is that only one word is programmed in each operation. When writing Configuration registers, one word is programmed during each operation. Only Working register, W0, is used as a temporary holding register for the data to be programmed.

Table 3-7 provides the default values of the Configuration registers.

Note: When writing to the Configuration registers, always set the TBLPAG register to F8h. This is the upper byte address of all locations of the Configuration registers.

Table 3-7 provides the ICSP programming details for programming the Configuration registers, including the serial pattern with the ICSP command code. This code must be transmitted, LSb first, using the PGECx and PGEDx pins (see Figure 3-2).

In Step 1 of Table 3-8, the Reset vector is exited. In Step 2 and 4, the 24-bit starting destination address for programming is loaded into the TBLPAG register and the W7 register. In Step 3, the NVMCON register is initialized for programming the Configuration register. In Step 5, the Configuration register data are loaded to W6. In Step 6, `TBLWT` instructions are used to copy the data from W6 to the write latch of the Configuration register.

After the write latch is loaded, the programming is initiated by setting the write bit in the NVMCON register in Steps 7 and 8. In Step 9, the internal PC is reset to 200h. This is a precautionary measure to prevent the PC from incrementing to unimplemented memory when large devices are being programmed. Finally, in Step 10, repeat Steps 5 through 9 to write other Configuration registers. While programming other Configuration registers, load W6 with their respective values and W7 with their respective addresses.

TABLE 3-7: DEFAULT VALUES FOR CONFIGURATION REGISTER SERIAL INSTRUCTION

Configuration Registers	Value
FBS	0Fh
FGS	03h
FOSCSEL	E1h
FOSC	3Bh
FWDT	DFh
FPOR(1)	FBh
FICD	E3h

Note 1: The I2C1SEL bit (FPOR[4]) is only implemented on PIC24FXXKLX02/KMX02/KMX04 devices and in all other devices, it should be programmed as '1'.

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

Command (Binary)	Data (Hex)	Command (Binary)
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	200007	MOV #0x0000, W7
Step 3: Set the NVMCON register to program the Configuration registers.		
0000	24004A	MOV #0x4004, W10
0000	883B0A	MOV W10, NVMCON
Step 4: Initialize the TBLPAG register.		
0000	200F80	MOV #0xF8, W6
0000	880190	MOV W0, TBLPAG
Step 5: Load the Configuration register data to W6.		
0000	2xxxx6	MOV #[FBS_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 the device's internal PC.		
0000	040200	GOTO 0x200
0000	000000	NOP
Step 10: Repeat Steps 5 through 9 to write other Configuration registers. Load W6 with their respective values and W7 with their respective addresses.		

3.8.1 CODE-PROTECT CONFIGURATION BITS

The FBS and FGS Configuration registers are special Configuration registers which control the code protection for the Boot Segment and General Segment, respectively. For each segment, two forms of code protection are provided. One form prevents code memory from being written (write protection), while the other prevents code memory from being read (read protection).

The BWRP and GWRP bits control write protection, and the BSS and GSS0 bits control read protection.

When write protection is enabled, any programming operation to code memory will fail. When read protection is enabled, any read from code memory will cause a 00h to be read, regardless of the actual contents of the code memory.

It is imperative that all code protection bits should be '1' while the device is being programmed and verified. Only after the device is programmed and verified, should any of the above bits be programmed to '0'.

Note 1: All bits in the FBS and FGS Configuration registers can only be programmed to a value of '0'. Bulk Erasing the chip is the only way to reprogram the code-protect bits from on ('0') to off ('1').

2: Code-protect bits (BSS, BWRP, GSS, GWRP) are in a group subject to write restrictions. If any bit is cleared, the rest cannot be cleared on a subsequent operation. All bits must be cleared using one operation.

TABLE 3-9: PIC24FXXKMXXX/KLXXX FAMILY CONFIGURATION BITS

Bit Field	Register	Description
BOREN[1:0]	FPOR[1:0]	Brown-out Reset Enable bits 11 = Brown-out Reset is enabled in hardware; SBOREN bit is disabled 10 = Brown-out Reset is enabled only while device is active and disabled in Sleep; SBOREN bit is disabled 01 = Brown-out Reset is controlled with the SBOREN bit setting 00 = Brown-out Reset is disabled in hardware; SBOREN bit is disabled
BORV[1:0]	FPOR[6:5]	Brown-out Reset Voltage bits 11 = Brown-out Reset is set to low trip point(1) 10 = Brown-out Reset is set to middle trip point(1) 01 = Brown-out Reset is set to high trip point(1) 00 = Downside protection on POR is enabled (low-power BOR is selected)
BSS[2:0]	FBS[3:1]	Boot Segment Program Flash Code Protection bits 111 = No Boot Segment; all program memory space is General Segment 110 = Standard security; Boot Segment starts at 0200h, ends at 0AFEh 101 = Standard security; boot program Flash segment starts at 0200h, ends at 15FEh(2) 100 = Reserved 011 = Reserved 010 = High-security Boot Segment starts at 0200h, ends at 0AFEh 001 = High-security, Boot Segment starts at 0200h, ends at 15FEh(2) 000 = Reserved
BWRP	FBS[0]	Boot Segment Program Flash Write Protection bit 1 = Boot Segment may be written 0 = Boot Segment is write-protected
$\overline{\text{DEBUG}}$	FICD[7]	Background Debugger Enable bit 1 = Background debugger is disabled 0 = Background debugger functions are enabled

Note 1: For the particular value of a trip point, refer to the specific device data sheet.

2: This selection is available only on PIC24F16KLXXX/KMXXX devices.

3: This applies only to 28-pin devices.

4: The MCLRE Configuration bit can only be changed when using the VPP-Based Test mode entry. This prevents a user from accidentally locking out the device from a low-voltage test entry.

TABLE 3-9: PIC24FXXKMXXX/KLXXX FAMILY CONFIGURATION BITS (CONTINUED)

Bit Field	Register	Description
FCKSM[1:0]	FOSC[7:6]	Clock Switching and Monitor Selection Configuration 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
FNOSC[2:0]	FOSCSEL[2:0]	Oscillator Selection bits 111 = Fast RC Oscillator with Divide-by-N (FRCDIV) 110 = 500 kHz Low-Power FRC Oscillator with Divide-by-N (LPFRCDIV) 101 = Low-Power RC Oscillator (LPRC) 100 = Secondary Oscillator (SOSC) 011 = Primary Oscillator with PLL module (HS+PLL, EC+PLL) 010 = Primary Oscillator (XT, HS, EC) 001 = Fast RC Oscillator with Divide-by-N with PLL module (FRCDIV+PLL) 000 = Fast RC Oscillator (FRC)
FWDTEN[1:0]	FWDT[7,5]	Watchdog Timer Enable bits 11 = WDT is enabled in hardware 10 = WDT is controlled with the SWDTEN bit 01 = WDT is enabled only while the device is active and is disabled in Sleep; SWDTEN bit is disabled 00 = WDT is disabled in hardware; SWDTEN bit is disabled
GSS0	FGS[1]	General Segment Code Flash Code Protection bit 1 = No protection 0 = Standard security is enabled
GWRP	FGS[0]	General Segment Code Flash Write Protection bit 1 = General Segment may be written 0 = General Segment is write-protected
ICS[1:0]	FICD[1:0]	ICD Pin Placement Select bits 11 = ICD EMUC/EMUD pins are shared with PGEC1/PGED1 10 = ICD EMUC/EMUD pins are shared with PGEC2/PGED2 01 = ICD EMUC/EMUD pins are shared with PGEC3/PGED3 00 = Reserved; do not use
IESO	FOSCSEL[7]	Internal External Switchover bit 1 = Internal External Switchover mode is enabled (Two-Speed Start-up is enabled) 0 = Internal External Switchover mode is disabled (Two-Speed Start-up is disabled)
I2C1SEL(3)	FPOR[4]	Alternate I2C1 Pin Mapping bit(3) 1 = Default location for SCL1/SDA1 pins 0 = Alternate location for SCL1/SDA1 pins
MCLRE(4)	FPOR[7]	$\overline{\text{MCLR}}$ Pin Enable bit(4) 1 = $\overline{\text{MCLR}}$ pin is enabled; RA5 input pin is disabled 0 = RA5 input pin is enabled; $\overline{\text{MCLR}}$ is disabled
OSCIOFNC	FOSC[2]	CLKO Enable Configuration bit 1 = CLKO output signal is active on the OSCO pin; the primary oscillator must be disabled or configured for the External Clock (EC) mode for the CLKO to be active (POSCMD[1:0] = 11 or 00) 0 = CLKO output is disabled

Note 1: For the particular value of a trip point, refer to the specific device data sheet.

2: This selection is available only on PIC24F16KLXXX/KMXXX devices.

3: This applies only to 28-pin devices.

4: The MCLRE Configuration bit can only be changed when using the VPP-Based Test mode entry. This prevents a user from accidentally locking out the device from a low-voltage test entry.

TABLE 3-9: PIC24FXXKMXXX/KLXXX FAMILY CONFIGURATION BITS (CONTINUED)

Bit Field	Register	Description
LPRCSEL	FOSCSEL[6]	Internal LPRC Oscillator Power Select bit 1 = High-Power/High-Accuracy mode 0 = Low-Power/Low-Accuracy mode
SOSCSRC	FOSCSEL[5]	Secondary Oscillator Clock Source Power Selection Configuration bit 1 = SOSC analog crystal function is available on the SOSCI/SOSCO pins 0 = SOSC crystal is disabled; digital SCLKI function is selected on the SOSCO pin
POSCMD[1:0]	FOSC[1:0]	Primary Oscillator Configuration bits 11 = Primary Oscillator mode is disabled 10 = HS Oscillator mode is selected (4 MHz-25 MHz) 01 = XT Oscillator mode is selected (100 kHz-4 MHz) 00 = External Clock mode is selected
POSCFREQ[1:0]	FOSC[4:3]	Primary Oscillator Frequency Range Configuration bits 11 = Primary oscillator/external clock input frequency is greater than 8 MHz 10 = Primary oscillator/external clock input frequency is between 100 kHz and 8 MHz 01 = Primary oscillator/external clock input frequency is less than 100 kHz 00 = Reserved; do not use
PWRTEN	FPOR[3]	Power-up Timer Enable bit 1 = PWRT is enabled 0 = PWRT is disabled
SOSCSEL	FOSC[5]	Secondary Oscillator Select bit 1 = Secondary oscillator is configured for high-power operation 0 = Secondary oscillator is configured for low-power operation
FWPSA	FWDT[4]	WDT Prescaler bit 1 = WDT prescaler ratio of 1:128 0 = WDT prescaler ratio of 1:32
WDTPS[3:0]	FWDT[3:0]	Watchdog Timer Postscale Select bits 1111 = 1:32,768 1110 = 1:16,384 • • • 0001 = 1:2 0000 = 1:1
WINDIS	FWDT[6]	Windowed Watchdog Timer Disable bit 1 = Standard WDT is selected; windowed WDT is disabled 0 = Windowed WDT is enabled

Note 1: For the particular value of a trip point, refer to the specific device data sheet.

2: This selection is available only on PIC24F16KLXXX/KMXXX devices.

3: This applies only to 28-pin devices.

4: The MCLRE Configuration bit can only be changed when using the VPP-Based Test mode entry. This prevents a user from accidentally locking out the device from a low-voltage test entry.

3.9 Reading Code Memory

To read the code memory, execute a series of TBLRD instructions and clock out the data using the REGOUT command.

Table 3-10 provides 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 the 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, which was used for writing, is also used for reading (see Figure 3-7). 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 required amount of code memory is read.

TABLE 3-10: SERIAL INSTRUCTION EXECUTION FOR READING CODE 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 the 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	BA1B96	TBLRDL [W6], [W7]
0000	000000	NOP
0000	000000	NOP
0001	[VISI]	Clock out contents of VISI register.
0000	000000	NOP
0000	BADBB6	TBLRDH [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's internal PC.		
0000	040200	GOTO 0x200
0000	000000	NOP
Step 6: Repeat Steps 4 and 5 until the required code memory is read.		

3.10 Reading Data EEPROM Memory

The procedure for reading data EEPROM memory is the same as reading the code memory. The only difference is that the 16-bit data words are read instead of the 24-bit words.

Table 3-11 provides the ICSP programming details for reading data memory.

Note:	When reading from EEPROM, always set the TBLPAG register to 7Fh. This is the upper byte address of all locations of data EEPROM.

TABLE 3-11: SERIAL INSTRUCTION EXECUTION FOR READING DATA EEPROM 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 the TBLRD instruction.		
0000	2007F0	MOV #0x7F, W0
0000	880190	MOV W0, TBLPAG
0000	2FExx6h	MOV #[SourceAddress15:0], W6;(FExx)
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 location of data EEPROM memory through the VISI register using the REGOUT command.		
0000	BA1B96	TBLRDL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0001	[VISI]	Clock out contents of VISI register.
0000	000000	NOP
Step 5: Repeat Step 4 until the required data EEPROM memory is read.		
Step 6: Reset the device's internal PC.		
0000	040200	GOTO 0x200
0000	000000	NOP

3.11 Reading Configuration Memory

The procedure for reading a Configuration register is the same as reading the code memory. The only difference is that the 16-bit data words are read (with the upper byte read being all '0's) instead of the 24-bit words. There are eight Configuration registers and they are read, one register at a time.

Table 3-12 provides the ICSP programming details for reading all of the Configuration registers.

Note:	When reading from the Configuration registers, always set the TBLPAG register to F8h. This is the upper byte address of all locations of the Configuration registers. The Read Pointer, W6, should be initialized to 00h.

TABLE 3-12: SERIAL INSTRUCTION EXECUTION FOR READING ALL THE 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 TBLPAG, the Read Pointer (W6) and the Write Pointer (W7) for the TBLRD instruction.		
0000	200F80	MOV #0xF8, W0
0000	880190	MOV W0, TBLPAG
0000	200007	MOV #0x0000,W6
0000	207847	MOV #VISI, W7
0000	000000	NOP
Step 3: Read the Configuration register and write it to the VISI register (located at 784h) and clock out the VISI register using the REGOUT command.		
0000	BA0BB6	TBLRDL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0001	[VISI]	Clock out contents of VISI register.
Step 4: Repeat Step 3 to read other Configuration registers. Load W6 with their respective addresses.		
Step 5: Reset the device's internal PC.		
0000	040200	GOTO 0x200
0000	000000	NOP

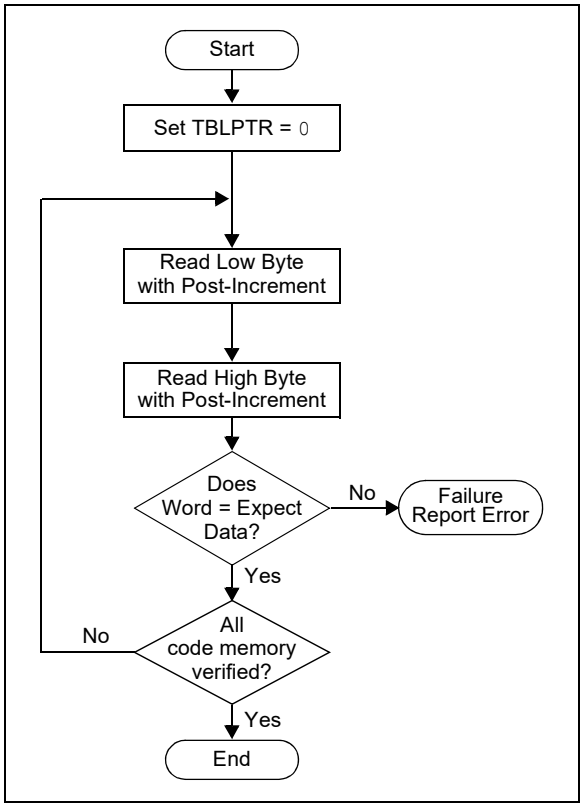
3.12 Verifying Code Memory, Data EEPROM Memory and Configuration Registers

To verify the code memory, read the code memory space and compare it with the copy held in the programmer's buffer. To verify the data EEPROM and Configuration registers, follow the similar procedure.

Figure 3-10 illustrates the verify process flowchart. Memory reads occur 1 byte at a time; hence, 2 bytes must be read to compare with the word in the programmer's buffer. Refer to **Section 3.9 "Reading Code Memory"** for implementation details of reading code memory. Additionally, the data EEPROM and Configuration registers can also be verified.

Note:	Code memory should be verified immediately after writing if code protection is enabled. Since Configuration registers include the device code protection bit, the device will not be readable or verifiable if a device Reset occurs after the code-protect bits are set (Value = 0).

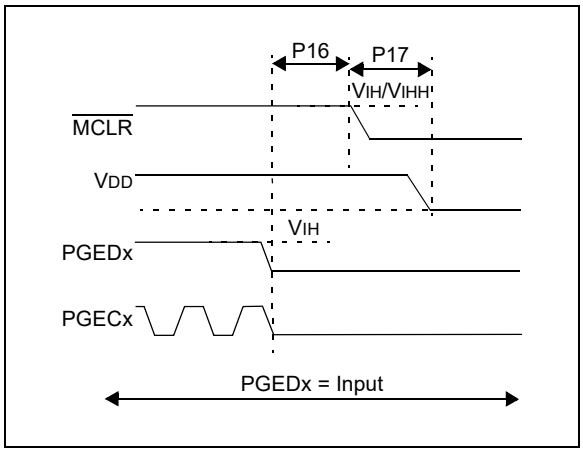
FIGURE 3-10: VERIFY CODE MEMORY FLOW



3.13 Exiting ICSP Mode

Exit the Program/Verify mode by removing VIH from MCLR/VPP, as illustrated in Figure 3-11. The only requirement to exit is that an interval of P16 should elapse between the last clock, and program signals on PGECx and PGEDx, before removing VIH.

FIGURE 3-11: EXITING ICSP™ MODE



4.0 DEVICE ID

The Device ID region of memory can be used to determine the mask, variant and manufacturing information about the device. The Device ID region is 2 x 16 bits and it can be read using the READC command. This region of memory is read-only and can also be read when code protection is enabled.

Table 4-1 provides the Device ID for each device and Table 4-2 provides the Device ID registers. Table 4-3 describes the bit field of each register.

TABLE 4-1: DEVICE IDs

Device ID	DEVID
PIC24F16KL402	4B14h
PIC24F16KL401	4B1Eh
PIC24F08KL402	4B04h
PIC24F08KL401	4B0Eh
PIC24F08KL302	4B00h
PIC24F08KL301	4B0Ah
PIC24F08KL201	4B06h
PIC24F08KL200	4B05h
PIC24F04KL101	4B02h
PIC24F04KL100	4B01h
PIC24FV16KM204	551F
PIC24FV08KM204	5517
PIC24FV16KM104	550F
PIC24FV16KM202	551B
PIC24FV08KM202	5513
PIC24FV16KM102	550B
PIC24FV08KM102	5503
PIC24FV08KM101	5501
PIC24F16KM204	551E
PIC24F08KM204	5516
PIC24F16KM104	550E
PIC24F16KM202	551A
PIC24F08KM202	5512
PIC24F16KM102	550A
PIC24F08KM102	5502
PIC24F08KM101	5500

TABLE 4-2: PIC24FXXKMXXX/KLXXX DEVICE ID REGISTERS

Address	Name	Bit															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FF0000h	DEVID	FAMID[7:0]								DEV[7:0]							
FF0002h	DEVREV	—												REV[3:0]			

TABLE 4-3: DEVICE ID BITS DESCRIPTION

Bit Field	Register	Description
FAMID[7:0]	DEVID	Encodes the family ID of the device.
DEV[7:0]	DEVID	Encodes the individual ID of the device.
REV[3:0]	DEVREV	Encodes the revision number of the device.

4.1 Checksums

4.1.1 CHECKSUM COMPUTATION

Checksums for the PIC24FXXKMXXX/KLXXX family are 16 bits. The checksum is calculated by summing the following:

- Contents of the code memory locations
- Contents of the Configuration registers

Table 4-4 describes how to calculate the checksum for each device.

All memory locations are summed, one byte at a time, using only their native data size. More specifically, Configuration registers are summed by adding the lower two bytes of these locations (the upper byte is ignored), while the code memory is summed by adding all three bytes of the code memory.

TABLE 4-4: CHECKSUM COMPUTATION

Device	Read Code Protection	Checksum Computation	Erased Checksum Value	Chip Checksum with 0xAAAAAA at 0x00 Location and at Last Location
PIC24F16KLXXX	Disabled	CFGB + SUM (0:002BFEh)	0xC18B	0xBF8D
	Enabled	0	0x0000	0x0000
PIC24F08KLXXX	Disabled	CFGB + SUM (0:0015FE)	0xE28B	0xE08D
	Enabled	0	0x0000	0x0000
PIC24F04KLXXX	Disabled	CFGB + SUM (0:000AFE)	0x730B	0x710D
	Enabled	0	0x0000	0x0000
PIC24FV16KMXXX(1)	Disabled	CFGB + SUM (0:002BFE)	0xC279	0xC07B
	Enabled	0	0x0000	0x0000
PIC24FV08KMXXX(1)	Disabled	CFGB + SUM (0:0015FE)	0xE379	0xE17B
	Enabled	0	0x0000	0x0000

Legend: Item Description
SUM[a:b] = Byte sum of locations, a to b inclusive (all 3 bytes of code memory)
CFGB = Configuration Block (masked):

For PIC24FXXKLXXX devices: The byte sum of (FBS & 0x000F + FGS & 0x0003 + FOSCSEL & 0x00E7 + FOSC & 0x00FF + FWDT & 0x00FF + FPOR & 0x00FB + FICD & 0x0083).

For PIC24FVXXKMXXX devices: The byte sum of (FBS & +0x0F + FGS & 0x03 + FOSCSEL & 0xE7 + FOSC & 0xFF + FWDT & 0xFF + FPOR & 0xFF + FICD & 0x83)

Note 1: Includes PIC24FXXKMXXX devices.

5.0 AC/DC CHARACTERISTICS AND TIMING REQUIREMENTS

TABLE 5-1: STANDARD OPERATING CONDITIONS

Standard Operating Conditions
Operating Temperature: 0°C to +70°C and programming: +25°C is recommended.

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
D110	VDD	Supply Voltage During Programming	2.0	5.5	V	Normal programming, PIC24FVXXKMXXX
D111	VDD	Supply Voltage During Programming	1.8	3.60	V	Normal programming, PIC24FXXKLXXX/KMXXX
D112	IPP	Programming Current on $\overline{MCLR}$	—	50	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	
D042	VIHH	Programing Voltage on VPP	7.75	9	V	
D080	VOL	Output Low Voltage	—	0.4	V	IOL = 8.5 mA @ 3.6V
D090	VOH	Output High Voltage	1.4	—	V	IOH = -3.0 mA @ 3.6V
D100	VBULK	Bulk Erase Voltage	2.5	—	V	
D115	RGOUT	Regulator Output Voltage	3.1	3.6	V	
D120	CEFC	External Filter Capacitor Value	4.7	—	µF	Series resistance < 3 Ohm recommended; < 5 Ohm required.
D012	CIO	Capacitive Loading on I/O Pin (PGEDx)	—	50	pF	To meet AC specifications
P1	TPGC	Serial Clock (PGECx) Period	125	—	ns	ICSP™ mode
P1A	TPGCL	Serial Clock (PGECx) Low Time	50	—	ns	ICSP mode
P1B	TPGCH	Serial Clock (PGECx) High Time	50	—	ns	ICSP mode
P2	TSET1	Input Data Setup Time to Serial Clock ↑	15	—	ns	
P3	THLD1	Input Data Hold Time from PGECx ↑	15	—	ns	
P4	TDLY1	Delay Between 4-Bit Command and Command Operand	40	—	ns	
P4A	TDLY1A	Delay Between 4-Bit Command Operand and the Next 4-Bit Command	40	—	ns	
P5	TDLY2	Delay Between the Last PGECx ↓ of Command Byte and the First PGECx ↑ of Read of Data Word	20	—	ns	
P6	TSET2	VDD ↑ Setup Time to $\overline{MCLR}$ ↑	100	—	ns	
P7	THLD2	Input Data Hold Time from $\overline{MCLR}$ ↑ VPP ↓ (from VIHH to VIH)	25	—	ms	
P10	TDLY6	PGECx Low Time After Programming	400	—	ns	
P11	TDLY7	Chip Erase Time	2.5	—	ms	
P12	TDLY10	Page (four rows) Erase Time	2.5	—	ms	
P13	TDLY9	Row Programming Time	1.25	—	ms	
P14	TR	$\overline{MCLR}$ Rise Time to Enter ICSP™ mode	—	1.0	µs	
P15	TVALID	Data Out Valid from PGECx ↑	10	—	ns	
P16	TDLY10	Delay Between Last PGECx ↓ and $\overline{MCLR}$ ↓	0	—	s	
P17	THLD3	$\overline{MCLR}$ ↓ to VDD ↓	—	100	ns	
P18	TKEY1	Delay Between First $\overline{MCLR}$ ↓ and First PGECx ↑ for Key Sequence on PGEDx	1	—	ms	
P19	TKEY2	Delay Between Last PGECx ↓ for Key Sequence on PGEDx and Second $\overline{MCLR}$ ↑	1	—	ms	

APPENDIX A: REVISION HISTORY

Rev A Document (09/2011)

This is the initial release of this document.

Rev B Document 11/2011

Corrects configuration default values in Table 3-7 and corrects checksum values in Table 4-4.

Rev C Document 4/2012

Added PIC24FXXKMXXX device information.

Rev D Document 5/2013

Corrects checksum values for PIC24FV16KMXXX and PIC24FV08KMXXX in Table 4-4. Edits to the minimum value for Parameter D042 in Table 5-1.

Rev E Document 11/2019

Added a note for clarification to **Section 3.8.1 "Code-Protect Configuration Bits"**. Minor text/ formatting updates applied throughout.

NOTES:

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.

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Trademarks

ISBN: 978-1-5224-5325-3

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

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
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

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

China - Beijing
Tel: 86-10-8569-7000

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

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

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

China - Guangzhou
Tel: 86-20-8755-8029

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

China - Hong Kong SAR
Tel: 852-2943-5100

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

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

China - Shanghai
Tel: 86-21-3326-8000

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

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

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

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

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

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

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

India - Pune
Tel: 91-20-4121-0141

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

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

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

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

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

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

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

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

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

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

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

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

Finland - Espoo
Tel: 358-9-4520-820

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

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

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

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

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

Italy - Padova
Tel: 39-049-7625286

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

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

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

Sweden - Gothenberg
Tel: 46-31-704-60-40

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

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