PIC24FXXKA2XX

PIC24FXXKA2XX Flash Programming Specifications

1.0 DEVICE OVERVIEW

This document defines the programming specifications for the PIC24FXXKA2XX family of 16-bit microcontroller devices. This is required only for developing programming support for the PIC24FXXKA2XX 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:

- PIC24F04KA200
- PIC24F04KA201

2.0 PROGRAMMING OVERVIEW OF THE PIC24FXXKA2XX FAMILY

PIC24FXXKA2XX family devices are programmed exclusively using In-Circuit Serial Programming™ (ICSP™). This method provides native, low-level programming capability to erase, program and verify the device. **Section 3.0 "ICSP Programming"** describes the ICSP method.

Note:	Unlike other PIC24F devices, PIC24FXXKA2XX devices do not support Enhanced ICSP programming. These devices also do not support in-circuit debugging over the ICSP interface.

2.1 Power Requirements

All devices in the PIC24FXXKA2XX family are 3.3V supply designs. The core, the peripherals and the I/O pins operate at 3.3V. The device can operate from 1.8V to 3.6V.

Table 2-1 provides the pins that are required for programming, which are indicated in Figure 2-1. Refer to the device data sheet for complete pin descriptions.

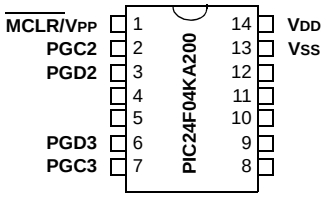
TABLE 2-1: PIN DESCRIPTIONS (DURING PROGRAMMING)

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
PGCx	PGC	I	Programming Pin Pair: Serial Clock
PGDx	PGD	I/O	Programming Pin Pair: Serial Data

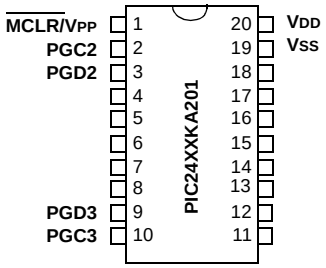
Legend: I = Input, O = Output, P = Power

FIGURE 2-1: PIC24FXXKA2XX PIN DIAGRAMS

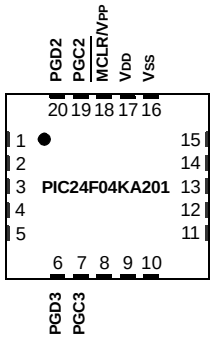
14-Pin SPDIP, SOIC



20-Pin SPDIP, SOIC



20-Pin QFN



2.2 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 about 1.38 K words of instructions (about 4 Kbytes). Figure 2-2 depicts the memory map.

Table 2-2 provides the program memory size and number of program 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.

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. See **Section 4.0 "Device ID"** for more information. The Device ID registers read out normally even after code protection is applied.

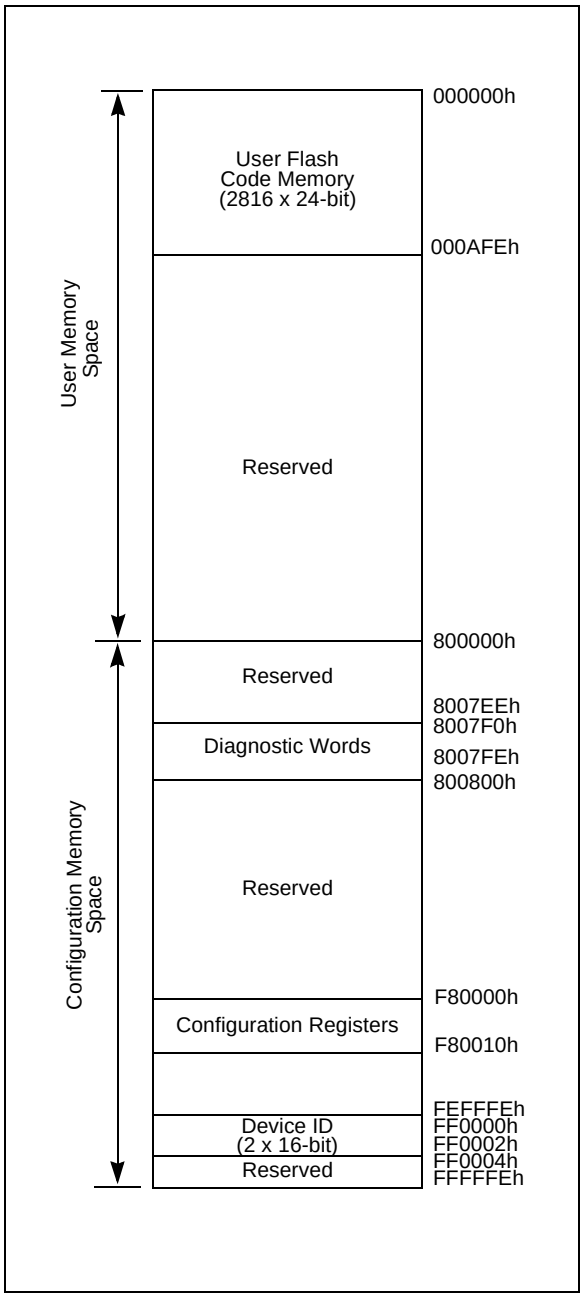
TABLE 2-2: PROGRAM MEMORY SIZES

Device	Program Memory Upper Address (Instruction Words)	Rows
PIC24F04KA200	AFEh (1.375K)	44
PIC24F04KA201		

TABLE 2-3: CONFIGURATION REGISTER LOCATIONS

Configuration Register	Address
FGS	F80004
FOSCSEL	F80006
FOSC	F80008
FWDT	F8000A
FPOR	F8000C
FICD	F8000E
FDS	F80010

FIGURE 2-2: PROGRAM MEMORY MAP



3.0 ICSP PROGRAMMING

The ICSP method is a special programming protocol that allows reading and writing to the PIC24FXXKA2XX device family memory. This is accomplished by applying control codes and instructions, serially to the device, using PGCx and PGDx pins.

In ICSP mode, the system clock is taken from the PGCx 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. PGDx is used to shift data in, and PGCx is used as both the serial shift clock and the CPU execution clock.

Note:	During ICSP operation, the operating frequency of PGCx 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 device configuration.
4. 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 PGCx and PGDx, and this control code is used to command the CPU (see Table 3-1).

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

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

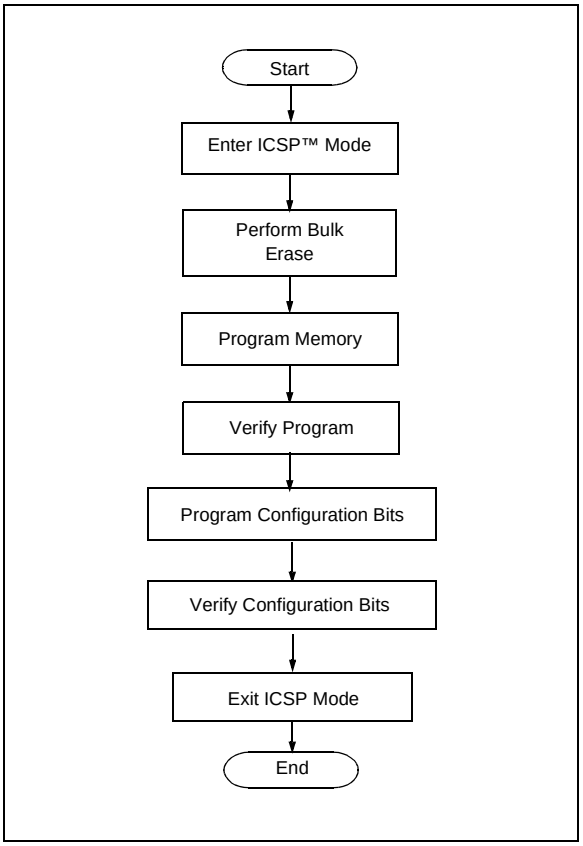


TABLE 3-1: CPU CONTROL CODES IN ICSP™ MODE

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

3.2.1 SIX SERIAL INSTRUCTION EXECUTION

The SIX control code allows execution of PIC24FXXKA2XX 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; 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 resumes 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 is 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 2 SIX operations to clock in all the necessary data.

  Examples of two-word instructions are: GOTO and CALL.

- Two-cycle instructions require 2 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 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, which is used by the instruction immediately following the CPU stall for Indirect Addressing. During normal operation, the CPU forces a NOP while the new data is read. To account for this, while using ICSP, any indirect references to a recently modified register should be proceeded with a NOP.

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

  If a two-cycle instruction modifies a register, which is used indirectly, it requires two following NOPs; one to execute the second half of the instruction and the other to stall the CPU to correct the pipeline.

  For example, TBLWTL [W0++],[W1] should be followed by 2 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".

3.2.2 REGOUT SERIAL INSTRUCTION EXECUTION

The REGOUT control code allows for the data to be extracted from the device in the ICSP mode. It is used to clock the contents of the VISI register out of the device over the PGDx pin. After the REGOUT control code is received, the CPU is held Idle for 8 cycles. After this, an additional 16 cycles are required to clock the data out (see Figure 3-3).

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

Note 1:	After the contents of VISI are shifted out, the PIC24FXXKA2XX devices maintain PGDx as an output until the first rising edge of the next clock is received.
2:	Data changes on the falling edge and latches on the rising edge of PGCx. For all data transmissions, the Least Significant bit (LSb) is transmitted first.

FIGURE 3-2: SIX SERIAL EXECUTION

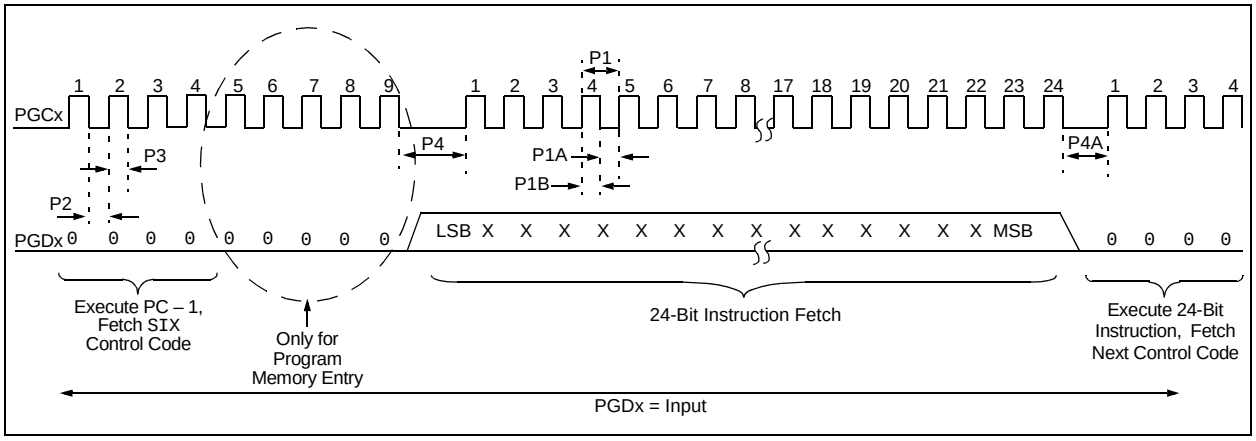
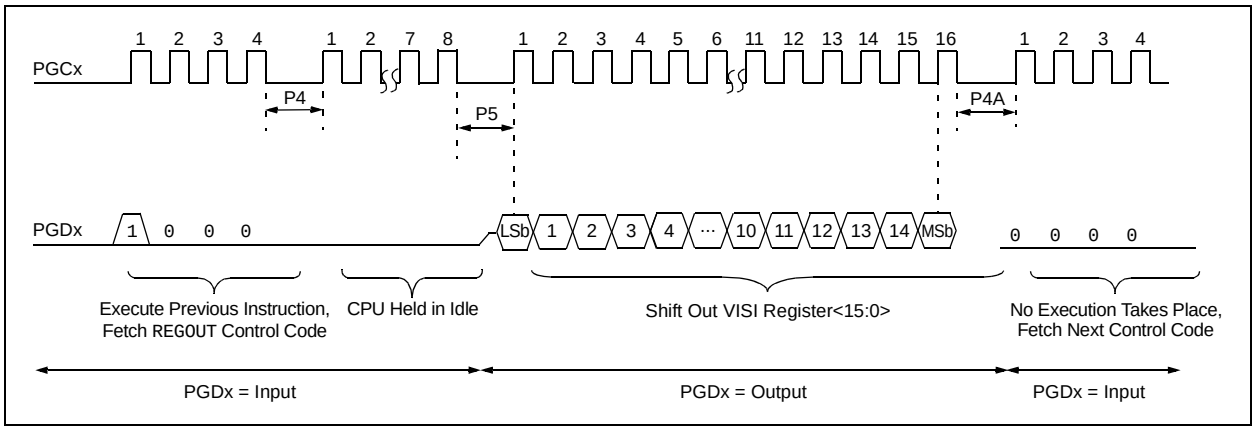


FIGURE 3-3: REG0UT SERIAL EXECUTION



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 PGDx.
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 PIC24FXXKA2XX 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 the Program/Verify mode is to be maintained. An interval of at least P19 and P7 must elapse before presenting data on PGDx. Signals appearing on PGCx 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 PGDx, 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 PGDx.

Signals appearing on PGDx 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.

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

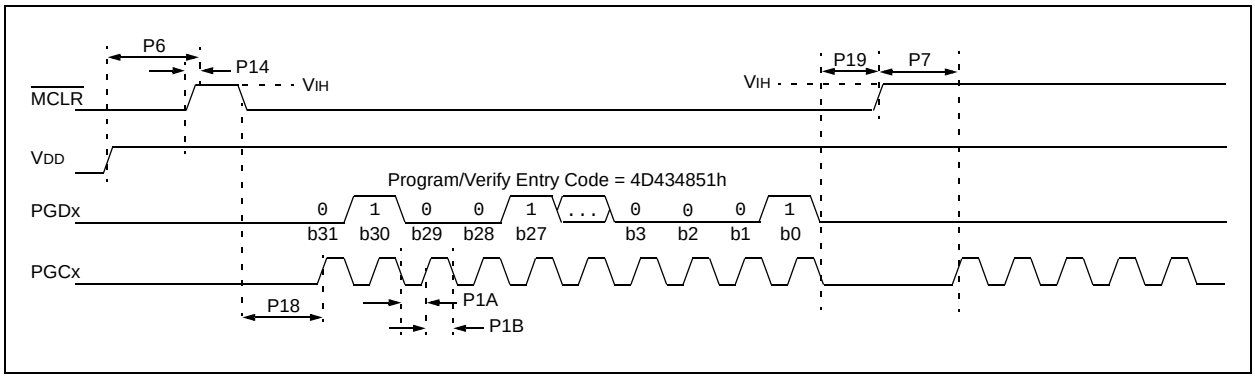
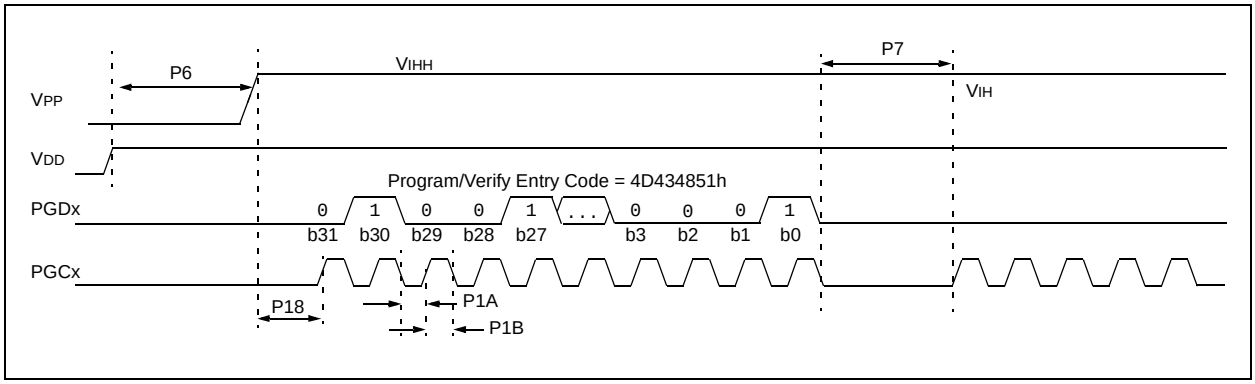


FIGURE 3-5: ENTERING ICSP™ MODE USING HIGH-VOLTAGE ENTRY



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 completed. 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 programming 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.
4054h	Erase all the Configuration registers (except the code-protect fuses).
4058h(1)	Erase Configuration registers except FBS and FGS.

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

TABLE 3-3: NVMCON VALUES FOR WRITE OPERATIONS

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

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

3.5 Erasing Program Memory

To erase the program memory (all of code memory, data memory and Configuration bits, including the code-protect bits), set the NVMCON 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 PGCx and PGDx pins (see Figure 3-2).

Table 3-4 provides the steps for executing 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

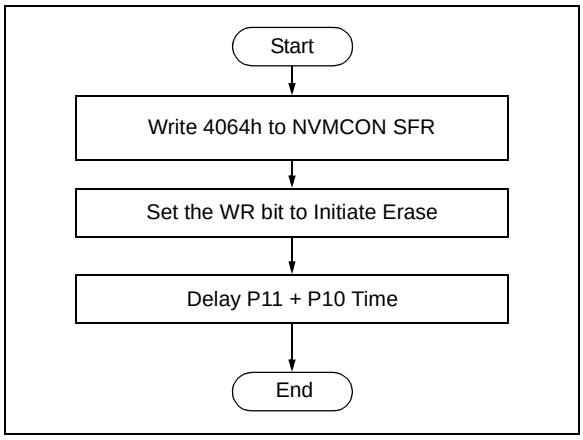


TABLE 3-4: SERIAL INSTRUCTION EXECUTION FOR CHIP ERASE

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 to erase the entire program memory.		
0000	24064A	MOV #0x4064, W10
0000	883B0A	MOV W10, NVMCON
Step 3: Set the TBLPAG and perform 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 PGCx and PGDx 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. In Step 3, the 24-bit starting destination address for programming is loaded into the TBLPAG register and W7 register. 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 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

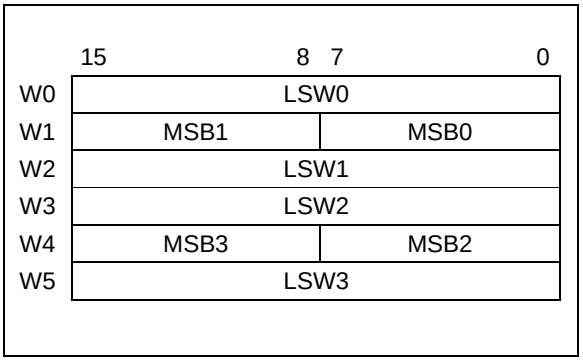


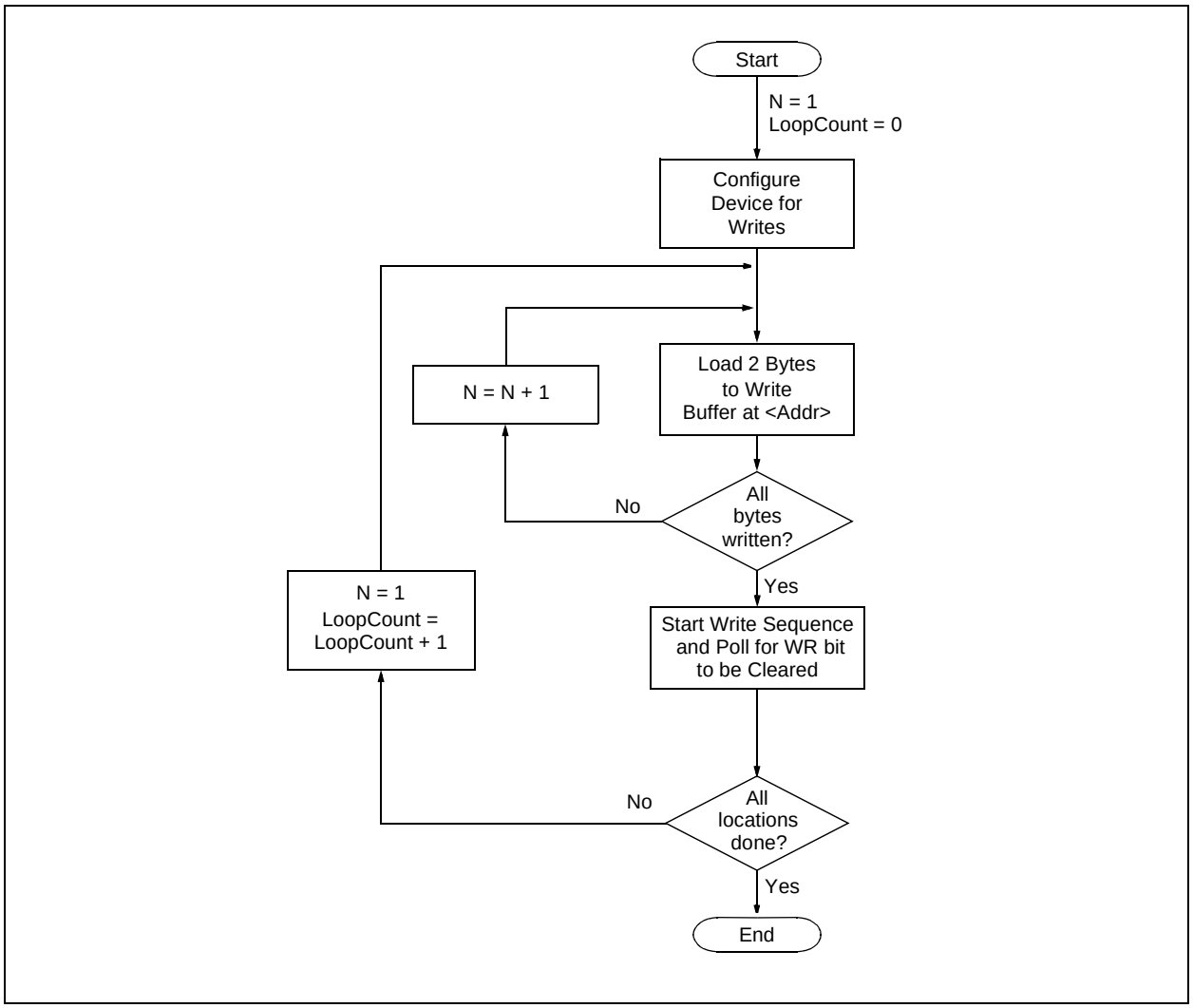
TABLE 3-5: SERIAL INSTRUCTION EXECUTION FOR WRITING CODE MEMORY

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 TBLWT instruction.		
0000	200xx0	MOV #<DestinationAddress23:16>, W0
0000	880190	MOV W0, TBLPAG
0000	2xxxx7	MOV #<DestinationAddress15:0>, W7

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

Command (Binary)	Data (Hex)	Description
Step 4: Load W0:W5 with the next 4 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
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 device 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 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 3-6 provides the default values of the Configuration registers.

Note:	The TBLPAG register is hard-coded to 0xF8 (the upper byte address of all locations of the Configuration registers).

Table 3-6 provides the ICSP programming details for programming the Configuration registers, including the serial pattern with the ICSP command code, which must be transmitted, LSB first, using the PGCx and PGDx pins (see Figure 3-2). In Step 1 of Table 3-7, the Reset vector is exited. In Step 2, the NVMCON register is initialized for programming code memory. In Step 3, the 24-bit starting destination address for programming is loaded into the TBLPAG register and W7 register.

Note:	The TBLPAG register must be loaded with F8h.

TABLE 3-6: DEFAULT VALUES FOR CONFIGURATION REGISTER SERIAL INSTRUCTION

Configuration Registers	Value
FGS	03h
FOSCSEL	87h
FOSC	FFh
FWDT	DFh
FPOR	FBh
FICD(1)	C3h
FDS	FFh

Note 1: The Configuration register, FICD, is a reserved location and should be programmed with the default value given above.

TABLE 3-7: 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 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 device internal PC.		
0000	040200	GOTO 0x200
0000	000000	NOP
Step 10: Repeat Steps 5 through 9 to write other fuses, Load W6 with their respective values and W7 with their respective addresses.		

TABLE 3-8: PIC24FXXKA2XX FAMILY CONFIGURATION BITS DESCRIPTION

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 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 = V_{BOR} set to 1.8V min 10 = V_{BOR} set to 2.0V min 01 = V_{BOR} set to 2.7V min 00 = Downside protection on POR enabled – “Zero-power” selected
DSWDTEN	FDS<7>	Deep Sleep Watchdog Timer Enable bit 1 = DSWDT is enabled 0 = DSWDT is disabled
DSWDTPS<3:0>	FDS<3:0>	Deep Sleep Watchdog Timer Postscale Select bits The DSWDT prescaler is 32; this creates an approximate base time unit of 1 ms. 1111 = 1:2,147,483,648 (25.7 days) 1110 = 1:536,870,912 (6.4 days) 1101 = 1:134,217,728 (38.5 hours) 1100 = 1:33,554,432 (9.6 hours) 1011 = 1:8,388,608 (2.4 hours) 1010 = 1:2,097,152 (36 minutes) 1001 = 1:524,288 (9 minutes) 1000 = 1:131,072 (135 seconds) 0111 = 1:32,768 (34 seconds) 0110 = 1:8,192 (8.5 seconds) 0101 = 1:2,048 (2.1 seconds) 0100 = 1:512 (528 ms) 0011 = 1:128 (132 ms) 0010 = 1:32 (33 ms) 0001 = 1:8 (8.3 ms) 0000 = 1:2 (2.1 ms)
DSZPBOR	FDS<6>	Deep Sleep Zero-Power BOR Enable bit 1 = Zero-Power BOR is enabled in Deep Sleep 0 = Zero-Power BOR is disabled in Deep Sleep (does not affect operation in non Deep Sleep modes)
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 000 = Fast RC Oscillator (FRC) 001 = Fast RC Oscillator with divide-by-N with PLL module (FRCDIV+PLL) 010 = Primary Oscillator (XT, HS, EC) 011 = Primary Oscillator with PLL module (HS+PLL, EC+PLL) 100 = Secondary Oscillator (SOSC) 101 = Low-Power RC Oscillator (LPRC) 110 = Reserved; do not use 111 = Fast RC Oscillator with divide-by-N (FRCDIV)

Note 1: The MCLRE fuse can only be changed when using the V_{PP}-Based Test mode entry. This prevents a user from accidentally locking out the device from low-voltage test entry.

TABLE 3-8: PIC24FXXKA2XX FAMILY CONFIGURATION BITS DESCRIPTION (CONTINUED)

Bit Field	Register	Description
FWPSA	FWDT<4>	WDT Prescaler 1 = WDT prescaler ratio of 1:128 0 = WDT prescaler ratio of 1:32
FWDTEN	FWDT<7>	Watchdog Timer Enable bit 1 = WDT is enabled 0 = WDT is disabled (control is placed on the SWDTEN bit)
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 bit 11 = Reserved; do not use 10 = PGEC2/PGED2 are used for ICSP programming 01 = PGEC3/PGED3 are used for ICSP programming 00 = Reserved; do not use
IESO	FOSCSEL<7>	Internal External Switchover bit 1 = Internal External Switchover mode is enabled (Two-Speed Start-up enabled) 0 = Internal External Switchover mode is disabled (Two-Speed Start-up disabled)
MCLRE(1)	FPOR<7>	$\overline{\text{MCLR}}$ Pin Enable bit(1) 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; primary oscillator must be disabled or configured for the External Clock mode (EC) for the CLKO to be active (POSCMD<1:0> = 11 or 00) 0 = CLKO output is disabled
POSCMD<1:0>	FOSC<1:0>	Primary Oscillator Configuration bits 11 = Primary oscillator disabled 10 = HS Oscillator mode selected (4 MHz-25 MHz) 01 = XT Oscillator mode selected (100 kHz-4 MHz) 00 = External Clock mode 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 0 = PWRT is disabled 1 = PWRT is enabled
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

Note 1: The MCLRE fuse can only be changed when using the VPP-Based Test mode entry. This prevents a user from accidentally locking out the device from low-voltage test entry.

TABLE 3-8: PIC24FXXKA2XX FAMILY CONFIGURATION BITS DESCRIPTION (CONTINUED)

Bit Field	Register	Description
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 selected; windowed WDT is disabled 0 = Windowed WDT is enabled

Note 1: The MCLRE fuse can only be changed when using the VPP-Based Test mode entry. This prevents a user from accidentally locking out the device from low-voltage test entry.

3.8 Reading Code Memory

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

Table 3-9 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 3-9: 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	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 device internal PC.		
0000	040200	GOTO 0x200
0000	000000	NOP
Step 6: Repeat Steps 4 and 5 until the required code memory is read.		

3.9 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 3-10 provides the ICSP programming details for reading all of the Configuration registers.

Note:	The TBLPAG register should be hard-coded to 0xF8 (the upper byte address of the Configuration register) and the Read Pointer, W6, is initialized to 0x00h.

TABLE 3-10: SERIAL INSTRUCTION EXECUTION FOR READING ALL THE CONFIGURATION REGISTERS

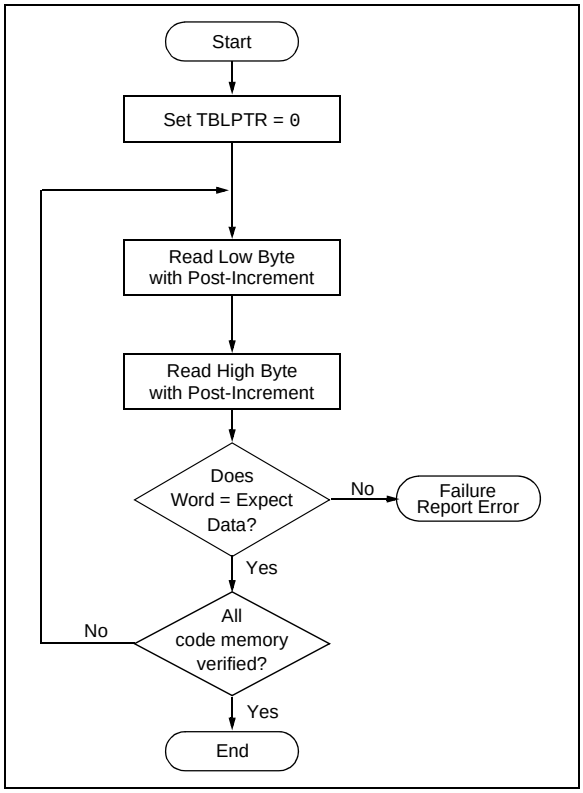
Command (Binary)	Data (Hex)	Description
Step 1: Exit Reset vector.		
0000	000000	NOP
0000	040200	GOTO 0x200
0000	000000	NOP
Step 2: Initialize TBLPAG, the Read Pointer (W6) and the Write Pointer (W7) for TBLRD instruction.		
0000	200F80	MOV #0xF8, W0
0000	880190	MOV W0, TBLPAG
0000	200007	MOV #0x0000,W6
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 fuses. Load W6 with their respective address.		
Step 5: Reset device internal PC.		
0000	040200	GOTO 0x200
0000	000000	NOP

3.10 Verifying Code 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. Figure 3-9 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 3.8 "Reading Code Memory"** for implementation details of reading code memory. On the same lines, the data EEPROM and Configuration registers can 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).

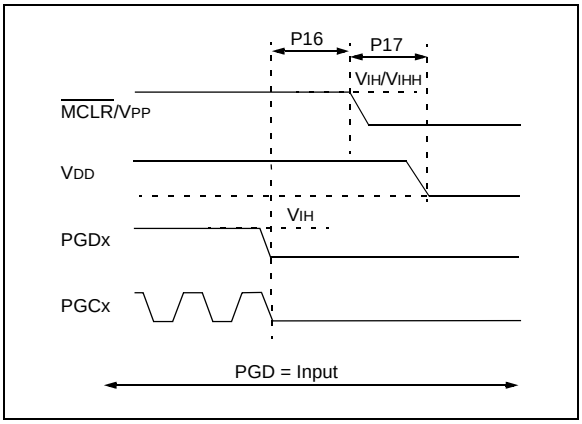
FIGURE 3-9: VERIFY CODE MEMORY FLOW



3.11 Exiting ICSP Mode

Exit the Program/Verify mode by removing VIH from MCLR/VPP as illustrated in Figure 3-10. The only requirement to exit is that an interval of P16 should elapse between the last clock and the program signals on PGCx and PGDx before removing VIH.

FIGURE 3-10: 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; 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
PIC24F04KA200	0D02h
PIC24F04KA201	0D00h

4.1 Checksums

4.1.1 CHECKSUM COMPUTATION

Checksums for the PIC24FXXKA2XX 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 4-2: PIC24FXXKA2XX 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.

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
PIC24F04KAXXX	Disabled	CFGB + SUM (0:000AFE)	0x74B4	0x72B6
	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),
Byte sum of (FGS & 0x0003 + FOSCSEL & 0x0087 + FOSC & 0x00DF + FWDT & 0x00DF + FPOR & 0x00FB + FICD & 0x00C3 + FDS & 0x00FF)

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**
D111	VDD	Supply Voltage During Programming	VDDCORE	3.60	V	Normal programming
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	VDD +1.5	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
D012	CIO	Capacitive Loading on I/O Pin (PGDx)	—	50	pF	To meet AC specifications
P1	TPGC	Serial Clock (PGCx) Period	125	—	ns	—
P1A	TPGCL	Serial Clock (PGCx) Low Time	50	—	ns	—
P1B	TPGCH	Serial Clock (PGCx) High Time	50	—	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 the Next 4-Bit Command	40	—	ns	—
P5	TDLY2	Delay Between Last PGCx ↓ of Command Byte and 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 $\overline{MCLR}$ ↑ VPP↓ (from VIHH to VIH)	25	—	ms	—
P10	TDLY6	PGCx Low Time After Programming	400	—	ns	—
P11	TDLY7	Chip Erase Time	5	—	ms	—
P12	TDLY10	Page (4 rows) Erase Time	5	—	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 Between First $\overline{MCLR}$ ↓ and First PGCx ↑ for Key Sequence on PGDx	1	—	ms	—
P19	TKEY2	Delay Between Last PGCx ↓ for Key Sequence on PGDx and Second $\overline{MCLR}$ ↑	1	—	ms	—

APPENDIX A: REVISION HISTORY

Rev A Document (9/2010)

Original version of this document; takes all information specific to PIC24XXKA20X family devices, originally found in DS39919, and created a new specification. No technical information regarding the devices or their programming has changed.

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.

Trademarks

ISBN: 978-1-60932-604-3

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

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://support.microchip.com
Web Address:
www.microchip.com

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

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

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

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

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

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

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

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

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

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

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

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

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

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

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

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

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

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

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

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

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

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

ASIA/PACIFIC

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

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

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

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

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

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

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

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

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

Taiwan - Hsin Chu
Tel: 886-3-6578-300
Fax: 886-3-6578-370

Taiwan - Kaohsiung
Tel: 886-7-213-7830
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

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

EUROPE

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

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

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

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

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

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

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

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

08/04/10