Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

PIC18F1XK50/PIC18LF1XK50

Flash Memory Programming Specification

1.0 DEVICE OVERVIEW

This document includes the programming specifications for the following devices:

- PIC18F13K50
- PIC18F14K50
- PIC18LF13K50
- PIC18LF14K50

2.0 PROGRAMMING OVERVIEW

The PIC18F1XK50/PIC18LF1XK50 devices can be programmed using either the high-voltage In-Circuit Serial Programming™ (ICSP™) method or the low-voltage ICSP method. Both methods can be done with the device in the user's system. The low-voltage ICSP method is slightly different than the high-voltage method and these differences are noted where applicable. The PIC18F1XK50 devices operate from 1.8 to 5.5 volts, and the PIC18LF1XK50 devices operate from 1.8 to 3.6 volts. All other aspects of the PIC18F1XK50 with regards to the PIC18LF1XK50 devices are identical.

2.1 Hardware Requirements

In High-Voltage ICSP mode, the PIC18F1XK50/PIC18LF1XK50 devices require two programmable power supplies: one for VDD and one for $\overline{\text{MCLR}}$/VPP/RA3. Both supplies should have a minimum resolution of 0.25V. Refer to **Section 8.1 "AC/DC Characteristics Timing Requirements for Program/Verify Test Mode"** for additional hardware parameters.

Note:	The VIH voltage levels on port pins RA0/D+/PGD and RA1/D-/PGC must be limited to 3.3V maximum, due to USB circuitry. The device must not be attached to a USB host and the USB module must be disabled. Refer to Figure 2-1, Figure 2-2 and Figure 2-3.

FIGURE 2-1: IN-CIRCUIT SERIAL PROGRAMMING™ (ICSP™) PIC18F1XK50 RECOMMENDED CIRCUIT

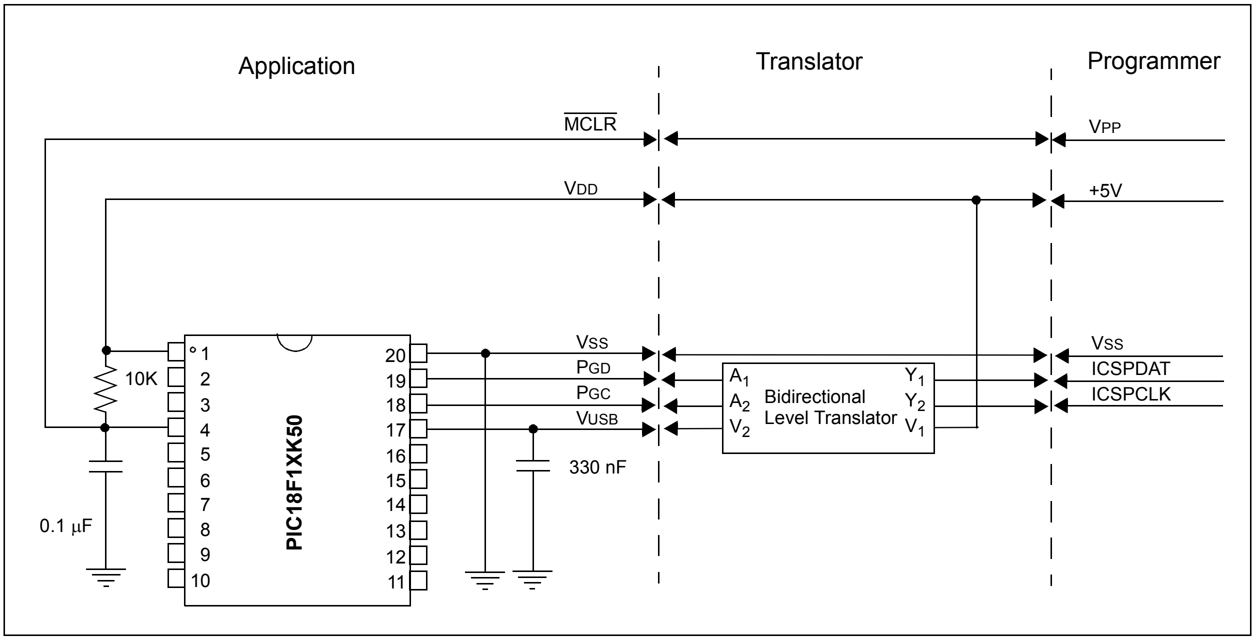


FIGURE 2-2: IN-CIRCUIT SERIAL PROGRAMMING™ (ICSP™) PIC18LF1XK50 RECOMMENDED CIRCUIT

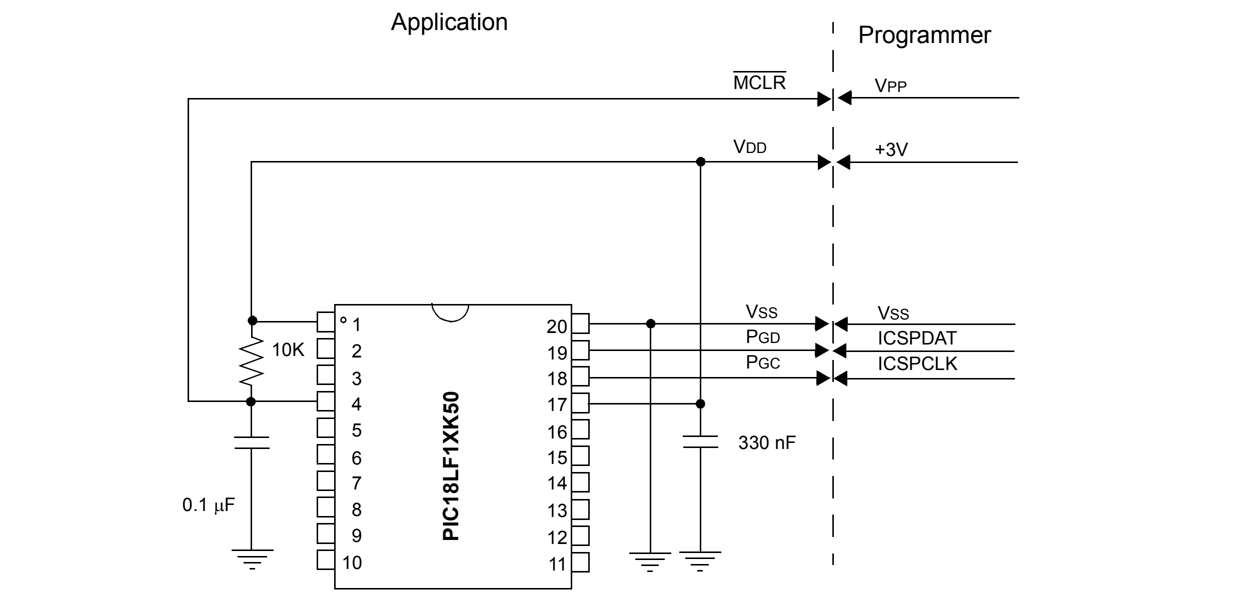
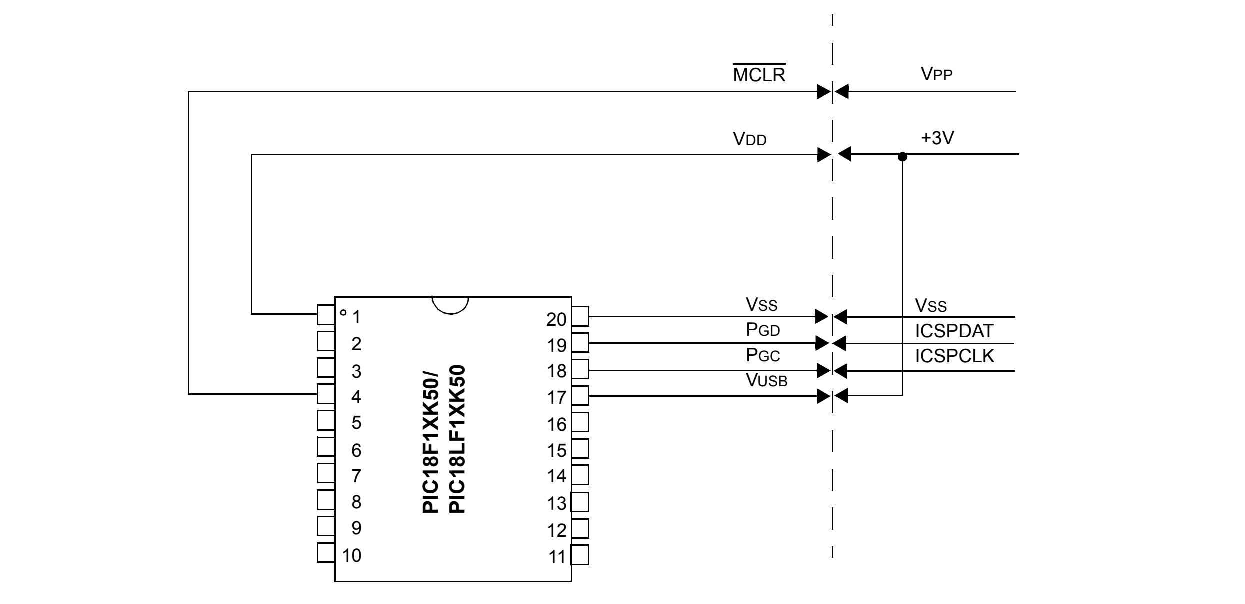


FIGURE 2-3: OUT OF CIRCUIT PROGRAMMING



2.1.1 LOW-VOLTAGE ICSP PROGRAMMING

In Low-Voltage ICSP mode, the PIC18F1XK50/PIC18LF1XK50 devices can be programmed using a single V_{DD} source in the operating range. The $\overline{\text{MCLR}}$/V_{PP}/RA3 does not have to be brought to a different voltage, but can instead be left at the normal operating voltage. Refer to **Section 8.1 "AC/DC Characteristics Timing Requirements for Program/Verify Test Mode"** for additional hardware parameters.

2.1.1.1 Single-Supply ICSP Programming

The LVP bit in Configuration register, CONFIG4L, enables single-supply (low-voltage) ICSP programming. The LVP bit defaults to a '1' (enabled) from the factory.

If Single-Supply Programming mode is not used, the LVP bit can be programmed to a '0' and RC3/PGM becomes a digital I/O pin. However, the LVP bit may only be programmed by entering the High-Voltage ICSP mode, where $\overline{\text{MCLR}}$/V_{PP}/RA3 is raised to V_{IHH}. Once the LVP bit is programmed to a '0', only the High-Voltage ICSP mode is available and only the High-Voltage ICSP mode can be used to program the device.

Note 1: The High-Voltage ICSP mode is always available, regardless of the state of the LVP bit, by applying V_{IHH} to the $\overline{\text{MCLR}}$/V_{PP}/RA3 pin.

2: While in Low-Voltage ICSP mode, the RC3 pin can no longer be used as a general purpose I/O.

2.2 Pin Diagrams

The pin diagrams for the PIC18F1XK50/PIC18LF1XK50 family are shown in Figure 2-4 and Figure 2-5.

TABLE 2-1: PIN DESCRIPTIONS (DURING PROGRAMMING): PIC18F1XK50/PIC18LF1XK50

Pin Name	During Programming		
	Pin Name	**Pin Type**	**Pin Description**
$\overline{\text{MCLR}}$/VPP/RA3	VPP	P	Programming Enable
VDD(2)	VDD	P	Power Supply
VUSB(3)	VUSB	P	Internal USB 3.3V Voltage Regulator
VSS(2)	VSS	P	Ground
RC3	PGM	I	Low-Voltage ICSP™ input when LVP Configuration bit equals ‘1’(1)
RA1	PGC	I	Serial Clock
RA0	PGD	I/O	Serial Data

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

Note 1: See Figure 6-1 for more information.

2: All power supply (VDD) and ground (VSS) pins must be connected.

3: Valid only for PIC18LF1XK50. This pin should be connected to VDD during programming.

FIGURE 2-4: 20-PIN PDIP, SSOP AND SOIC PIN DIAGRAM FOR PIC18F1XK50 AND PIC18LF1XK50

PDIP, SSOP, SOIC (300 MIL)

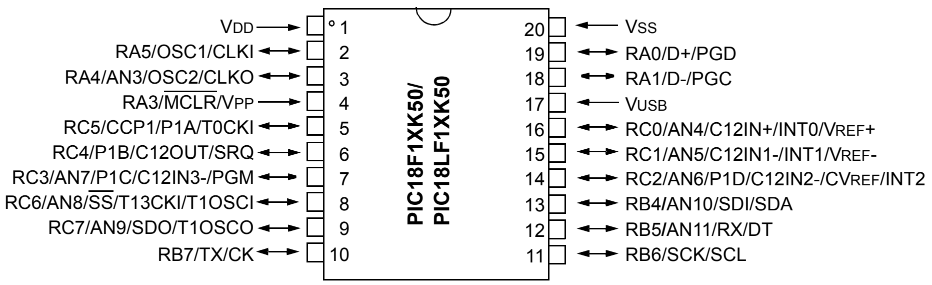
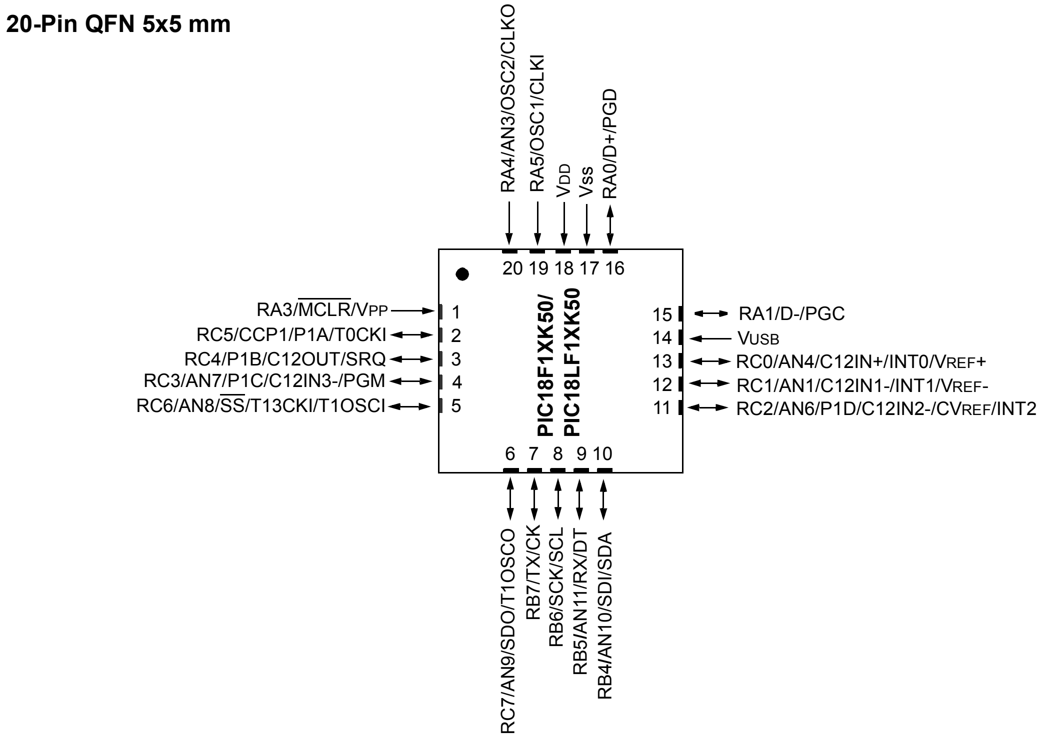


FIGURE 2-5: 20-PIN QFN PIN DIAGRAMS FOR PIC18F1XK50 AND PIC18LF1XK50



3.0 MEMORY MAPS

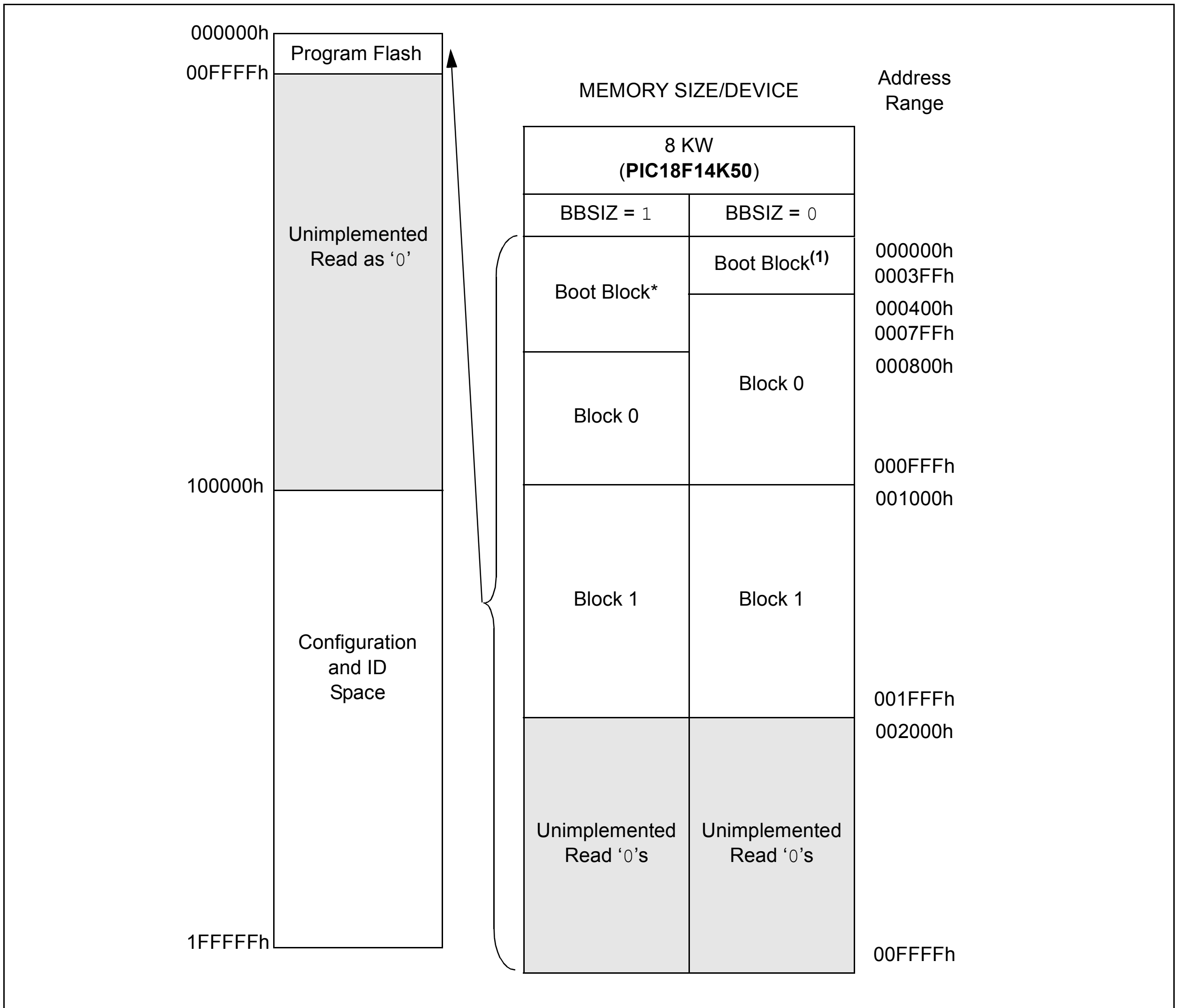
For the PIC18F14K50/PIC18LF14K50 device, the program Flash space extends from 0000h to 03FFFh (16 Kbytes) in two 8-Kbyte blocks. For the PIC18F13K50/PIC18LF13K50 device, the program Flash space extends from 0000h to 01FFFh (8 Kbytes) in two 4-Kbyte blocks.

For the PIC18F14K50/PIC18LF14K50 addresses 0000h through 0FFFh, however, define a “Boot Block” region that is treated separately from Block 0. For the PIC18F13K50/PIC18LF13K50 addresses 0000h through 07FFh, define the “Boot Block” region. All of these blocks define code protection boundaries within the program Flash space. The size of the Boot Block in the PIC18F14K50/PIC18LF14K50 devices can be configured as 2K, or 4 Kbyte (see Figure 3-1). The size of the Boot Block in the PIC18F13K50/PIC18LF13K50 devices can be configured as 1K, or 2 Kbytes, as illustrated in Figure 3-1. This is done through the BBSIZ bit in the Configuration register, CONFIG4L. It is important to note that increasing the size of the Boot Block decreases the size of the Block 0.

TABLE 3-1: IMPLEMENTATION OF PROGRAM FLASH

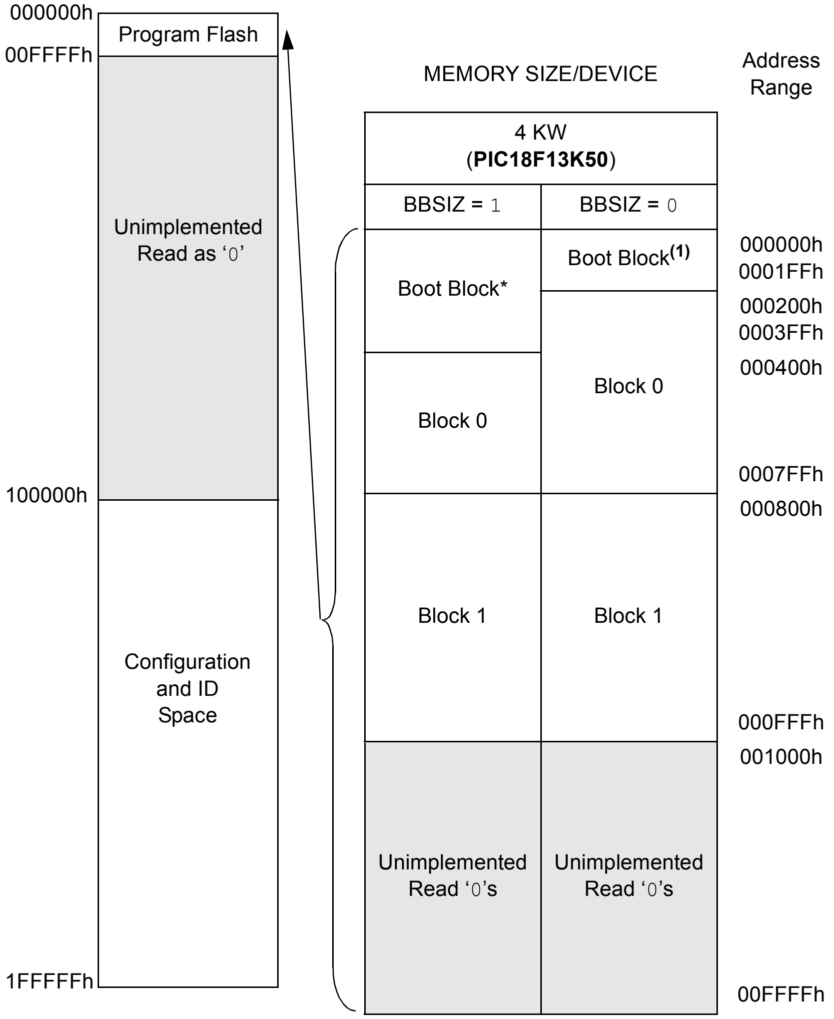
Device	Program Flash Size (Words)
PIC18F13K50/ PIC18LF13K50	000000h-000FFFh (4K)
PIC18F14K50/ PIC18LF14K50	000000h-001FFFh (8K)

FIGURE 3-1: MEMORY MAP AND THE PROGRAM FLASH SPACE FOR PIC18F14K50/ PIC18LF14K50 DEVICES



Note 1: Boot Block size is determined by the BBSIZ bit in the CONFIG4L register.

FIGURE 3-2: MEMORY MAP AND THE PROGRAM FLASH SPACE FOR PIC18F13K50/ PIC18LF13K50 DEVICES



Note 1: Boot Block size is determined by the BBSIZ bit in the CONFIG4L register.

In addition to the program Flash space, there are three blocks in the Configuration and ID space that are accessible to the user through table reads and table writes. Their locations in the memory map are shown in Figure 3-3.

Users may store identification information (ID) in eight ID registers. These ID registers are mapped in addresses 200000h through 200007h. The ID locations read out normally, even after code protection is applied.

Locations 300000h through 30000Dh are reserved for the Configuration bits. These bits select various device options and are described in **Section 6.0 "Configuration Word"**. These Configuration bits read out normally, even after code protection.

Locations 3FFFFEh and 3FFFFFh are reserved for the device ID bits. These bits may be used by the programmer to identify what device type is being programmed, and are described in **Section 6.0 "Configuration Word"**. These device ID bits read out normally, even after code protection.

3.1 Memory Address Pointer

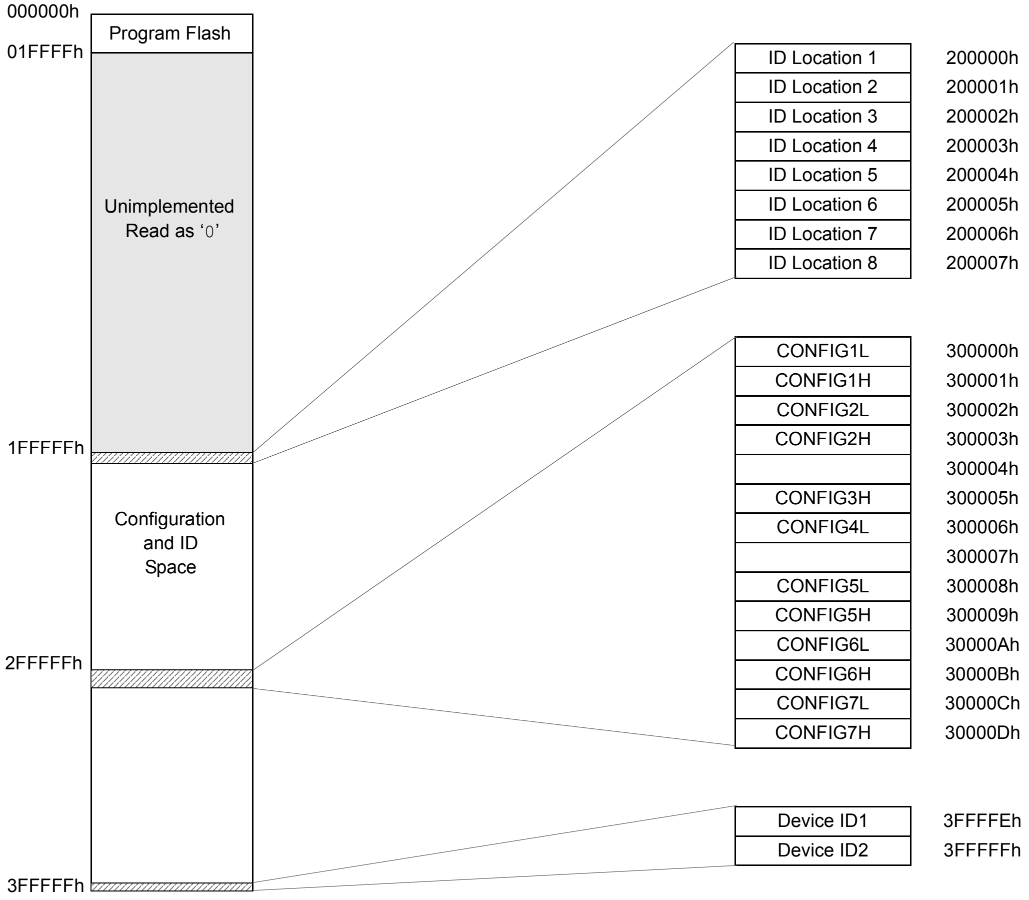
Memory in the address space, 0000000h to 3FFFFFh, is addressed via the Table Pointer register, which is comprised of three Pointer registers:

- TBLPTRU, at RAM address 0FF8h
- TBLPTRH, at RAM address 0FF7h
- TBLPTRL, at RAM address 0FF6h

TBLPTRU	TBLPTRH	TBLPTRL
Addr[21:16]	Addr[15:8]	Addr[7:0]

The 4-bit command, '0000' (core instruction), is used to load the Table Pointer prior to using any read or write operations.

FIGURE 3-3: CONFIGURATION AND ID LOCATIONS FOR PIC18F1XK50/PIC18LF1XK50 DEVICES

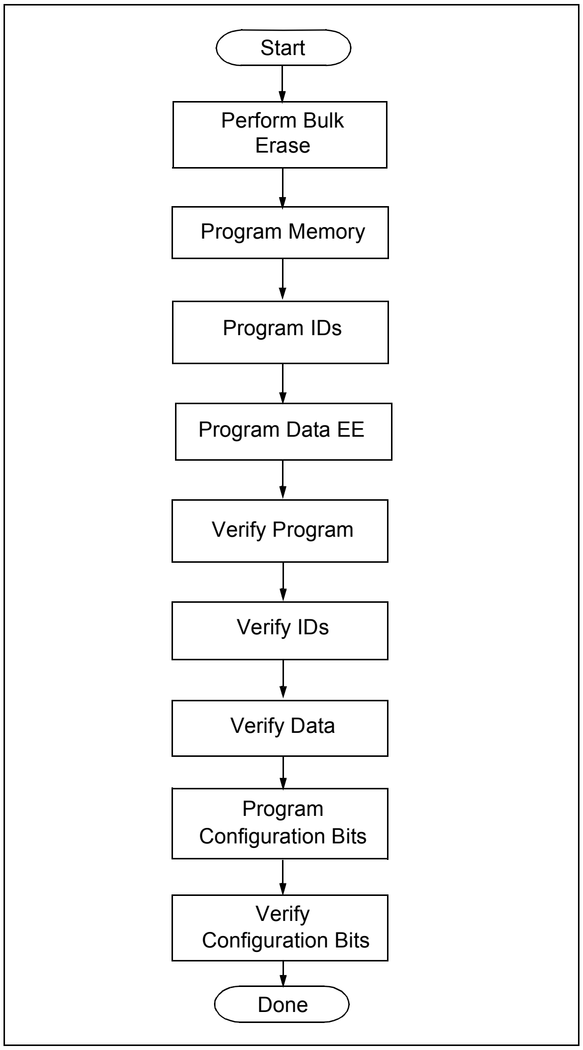


Note: Sizes of memory areas are not to scale.

3.2 High-Level Overview of the Programming Process

Figure 3-4 shows the high-level overview of the programming process. First, a Bulk Erase is performed. Next, the program Flash, ID locations and data EEPROM are programmed. These memories are then verified to ensure that programming was successful. If no errors are detected, the Configuration bits are then programmed and verified.

FIGURE 3-4: HIGH-LEVEL PROGRAMMING FLOW

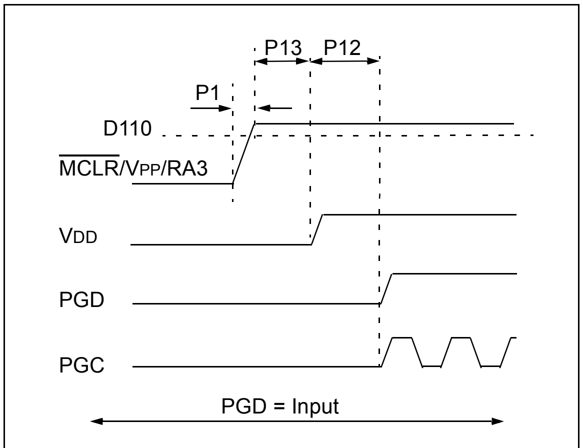


3.3 Entering and Exiting High-Voltage ICSP Program/Verify Mode

As shown in Figure 3-6, the High-Voltage ICSP Program/Verify mode is entered by holding PGC and PGD low and then raising $\overline{\text{MCLR}}$/VPP/RA3 to VIHH (high voltage). Once in this mode, the program Flash, data EEPROM, ID locations and Configuration bits can be accessed and programmed in serial fashion. Figure 3-7 shows the exit sequence.

The sequence that enters the device into the Program/Verify mode places all unused I/Os in the high-impedance state.

FIGURE 3-5: VPP-FIRST PROGRAM/VERIFY MODE ENTRY



Note: This method of entry is valid, regardless of Configuration Word selected.

FIGURE 3-6: VDD-FIRST PROGRAM/VERIFY MODE ENTRY

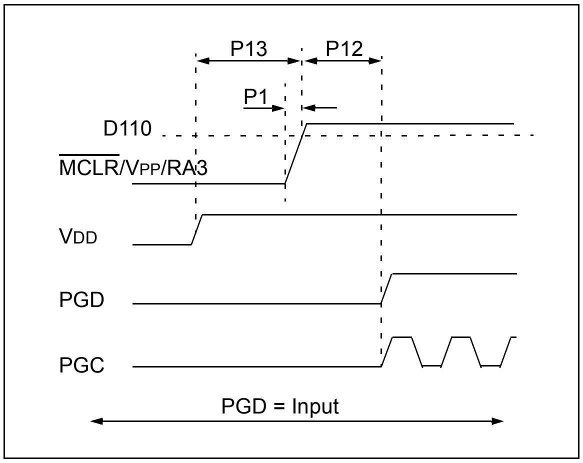
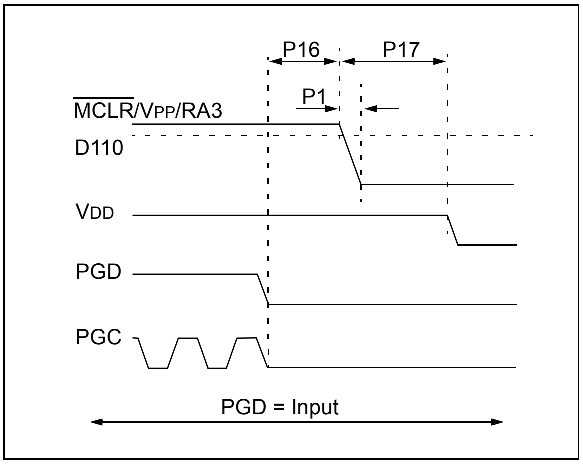


FIGURE 3-7: EXITING HIGH-VOLTAGE PROGRAM/VERIFY MODE



3.4 Entering and Exiting Low-Voltage ICSP Program/Verify Mode

When the LVP Configuration bit is '1' (see **Section 2.1.1.1 "Single-Supply ICSP Programming"**), the Low-Voltage ICSP mode is enabled. As shown in Figure 3-8, Low-Voltage ICSP Program/Verify mode is entered by holding PGC and PGD low, placing a logic high on PGM and then raising MCLR/VPP/RA3 to VIH. In this mode, the RC3/PGM pin is dedicated to the programming function and ceases to be a general purpose I/O pin. Figure 3-9 shows the exit sequence.

The sequence that enters the device into the Program/ Verify mode places all unused I/Os in the high-impedance state.

FIGURE 3-8: ENTERING LOW-VOLTAGE PROGRAM/VERIFY MODE

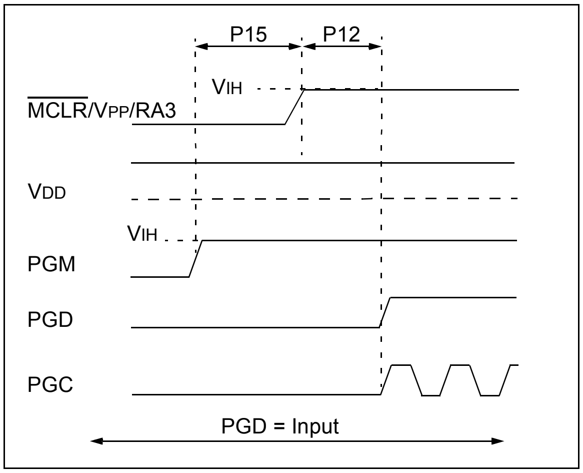
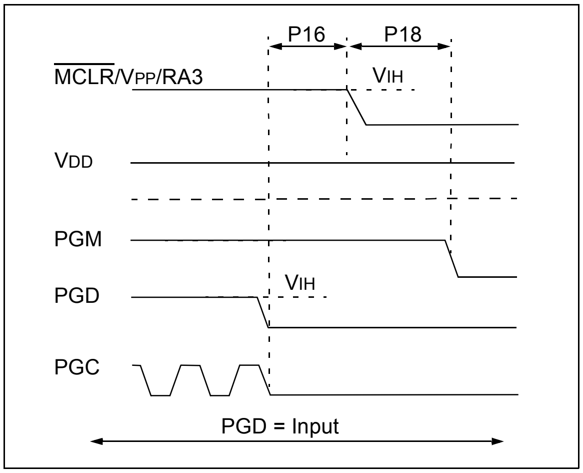


FIGURE 3-9: EXITING LOW-VOLTAGE PROGRAM/VERIFY MODE



3.5 Serial Program/Verify Operation

The PGC pin is used as a clock input pin and the PGD pin is used for entering command bits and data input/output during serial operation. Commands and data are transmitted on the rising edge of PGC, latched on the falling edge of PGC, and are Least Significant bit (LSb) first.

3.5.1 4-BIT COMMANDS

All instructions are 20 bits, consisting of a leading 4-bit command, followed by a 16-bit operand, which depends on the type of command being executed. To input a command, PGC is cycled four times. The commands needed for programming and verification are shown in Table 3-2.

Depending on the 4-bit command, the 16-bit operand represents 16 bits of input data or 8 bits of input data, and 8 bits of output data.

Throughout this specification, commands and data are presented as illustrated in Table 3-3. The 4-bit command, Most Signification bit (MSb), is shown first. The command operand, or “Data Payload”, is shown <MSB><LSB>. Figure 3-10 demonstrates how to serially present a 20-bit command/operand to the device.

3.5.2 CORE INSTRUCTION

The core instruction passes a 16-bit instruction to the CPU core for execution. This is needed to set up registers as appropriate for use with other commands.

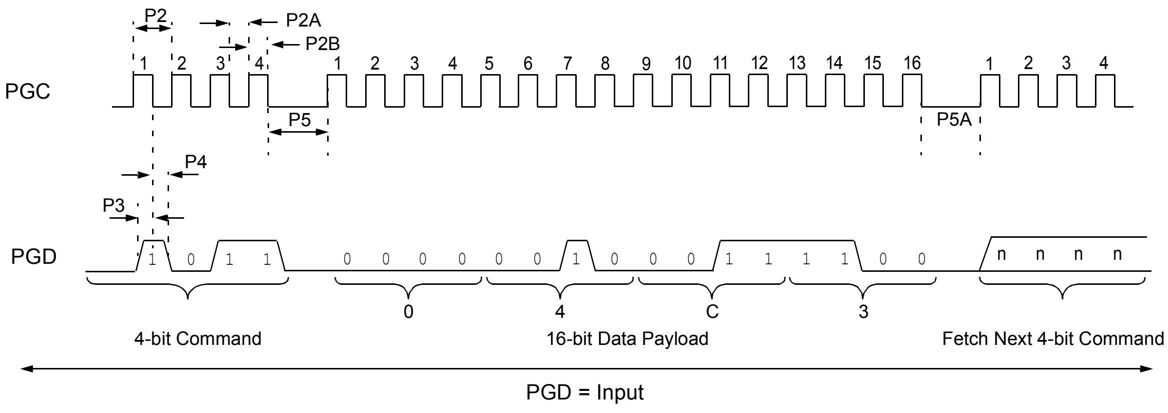
TABLE 3-2: COMMANDS FOR PROGRAMMING

Description	4-Bit Command
Core Instruction (Shift in16-bit instruction)	0000
Shift out TABLAT register	0010
Table Read	1000
Table Read, post-increment	1001
Table Read, post-decrement	1010
Table Read, pre-increment	1011
Table Write	1100
Table Write, post-increment by 2	1101
Table Write, start programming, post-increment by 2	1110
Table Write, start programming	1111

TABLE 3-3: SAMPLE COMMAND SEQUENCE

4-Bit Command	Data Payload	Core Instruction
1101	3C 40	Table Write, post-increment by 2

FIGURE 3-10: TABLE WRITE, POST-INCREMENT TIMING DIAGRAM (1101)



4.0 DEVICE PROGRAMMING

Programming includes the ability to erase or write the various memory regions within the device.

In all cases, except high-voltage ICSP Bulk Erase, the EECON1 register must be configured in order to operate on a particular memory region.

When using the EECON1 register to act on program Flash, the EEPGD bit must be set (EECON1<7> = 1) and the CFGS bit must be cleared (EECON1<6> = 0). The WREN bit must be set (EECON1<2> = 1) to enable writes of any sort (e.g., erases) and this must be done prior to initiating a write sequence. The FREE bit must be set (EECON1<4> = 1) in order to erase the program space being pointed to by the Table Pointer. The erase or write sequence is initiated by setting the WR bit (EECON1<1> = 1). It is strongly recommended that the WREN bit only be set immediately, prior to a program or erase.

4.1 ICSP Erase

4.1.1 HIGH-VOLTAGE ICSP BULK ERASE

Erasing program Flash or data EEPROM is accomplished by configuring two Bulk Erase Control registers, located at 3C0004h and 3C0005h. Program Flash may be erased portions at a time, or the user may erase the entire device in one action. Bulk Erase operations will also clear any code-protect settings associated with the memory block erased. Erase options are detailed in Table 4-1. If data EEPROM is code-protected (CPD = 0), the user must request an erase of data EEPROM (e.g., 0084h as shown in Table 4-1).

TABLE 4-1: BULK ERASE OPTIONS

Description	Data (3C0005h:3C0004h)
Chip Erase	0F8Fh
Erase User IDs	0088h
Erase Data EEPROM	0084h
Erase Boot Block	0081h
Erase Config Bits	0082h
Erase Program Flash Block 0	0180h
Erase Program Flash Block 1	0280h
Erase Program Flash Block 2	0480h
Erase Program Flash Block 3	0880h

The actual Bulk Erase function is a self-timed operation. Once the erase has started (falling edge of the 4th PGC after the NOP command), serial execution will cease until the erase completes (parameter P11). During this time, PGC may continue to toggle but PGD must be held low.

The code sequence to erase the entire device is shown in Table 4-2 and the flowchart is shown in Figure 4-1.

Note:	A Bulk Erase is the only way to reprogram code-protect bits from an “on” state to an “off” state.

TABLE 4-2: BULK ERASE COMMAND SEQUENCE

4-Bit Command	Data Payload	Core Instruction
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 05	MOVLW 05h
0000	6E F6	MOVWF TBLPTRL
1100	0F 0F	Write 0Fh to 3C0005h
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 04	MOVLW 04h
0000	6E F6	MOVWF TBLPTRL
1100	8F 8F	Write 8F8Fh TO 3C0004h to erase entire device.
0000	00 00	NOP
0000	00 00	Hold PGD low until erase completes.

FIGURE 4-1: BULK ERASE FLOW

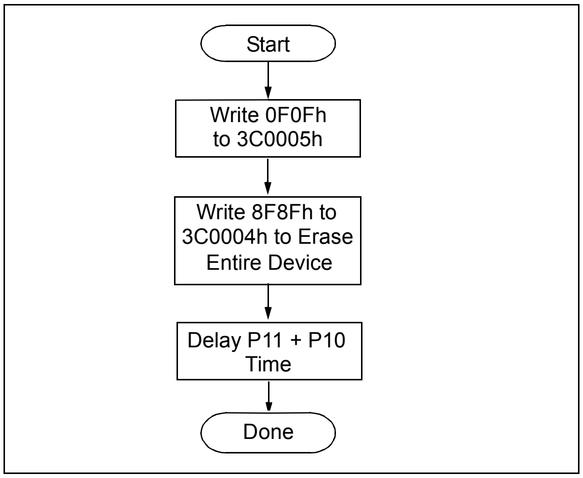
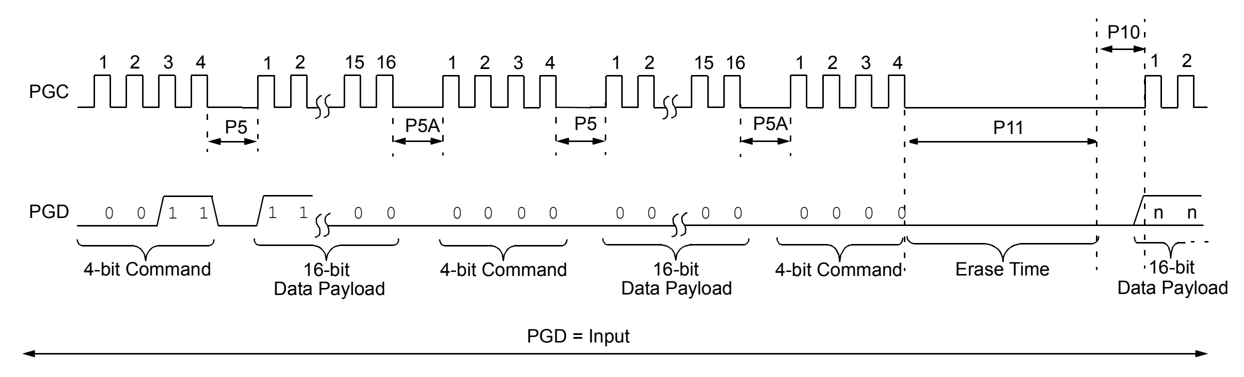


FIGURE 4-2: BULK ERASE TIMING DIAGRAM



4.1.2 LOW-VOLTAGE ICSP BULK ERASE

When using low-voltage ICSP, the part must be supplied by the voltage specified in parameter D111 if a Bulk Erase is to be executed. All other Bulk Erase details, as described above, apply.

If it is determined that a program memory erase must be performed at a supply voltage below the Bulk Erase limit, refer to the erase methodology described in **Section 4.1.3 "ICSP Row Erase"** and **Section 4.2.1 "Modifying Program Flash"**.

If it is determined that a data EEPROM erase must be performed at a supply voltage below the Bulk Erase limit, follow the methodology described in **Section 4.3 "Data EEPROM Programming"** and write '1's to the array.

4.1.3 ICSP ROW ERASE

Regardless of whether high or low-voltage ICSP is used, it is possible to erase one row (64 bytes of data), provided the block is not code or write-protected. Rows are located at static boundaries, beginning at program memory address 000000h, extending to the internal program memory limit (see **Section 3.0 "Memory Maps"**).

The Row Erase duration is self-timed. After the WR bit in EECON1 is set, two NOPs are issued. Erase starts upon the 4th PGC of the second NOP. It ends when the WR bit is cleared by hardware.

The code sequence to Row Erase a PIC18F1XK50/PIC18LF1XK50 device is shown in Table 4-3. The flowchart shown in Figure 4-3 depicts the logic necessary to completely erase the PIC18F1XK50/PIC18LF1XK50 devices. The timing diagram for Row Erase is identical to the data EEPROM write timing, shown in Figure 4-7.

Note 1: The TBLPTR register can point at any byte within the row intended for erase.

2: ICSP row erase of the User ID locations is also possible using the technique described in **Section 4.1.3 "ICSP Row Erase"**. The address argument used should be 0x200000. A row erase of the User ID locations is required when VDD is below the Bulk Erase threshold.

TABLE 4-3: ERASE PROGRAM FLASH CODE SEQUENCE

4-bit Command	Data Payload	Core Instruction
Step 1: Direct access to program Flash and enable writes.		

`0000`
`0000`
`0000`
	8E A6
	9C A6
	84 A6
		`BSF    EECON1, EEPGD`
		`BCF    EECON1, CFGS`
		`BSF    EECON1, WREN`

Step 2: Point to first row in program Flash.		

`0000`
`0000`
`0000`
	6A F8
	6A F7
	6A F6
		`CLRF   TBLPTRU`
		`CLRF   TBLPTRH`
		`CLRF   TBLPTRL`

Step 3: Enable erase and erase single row.		

`0000`
`0000`
`0000`
`0000`
	88 A6
	82 A6
	00 00
	00 00
		`BSF    EECON1, FREE`
		`BSF    EECON1, WR`
		`NOP`
		`NOP    Erase starts on the 4th clock of this instruction`

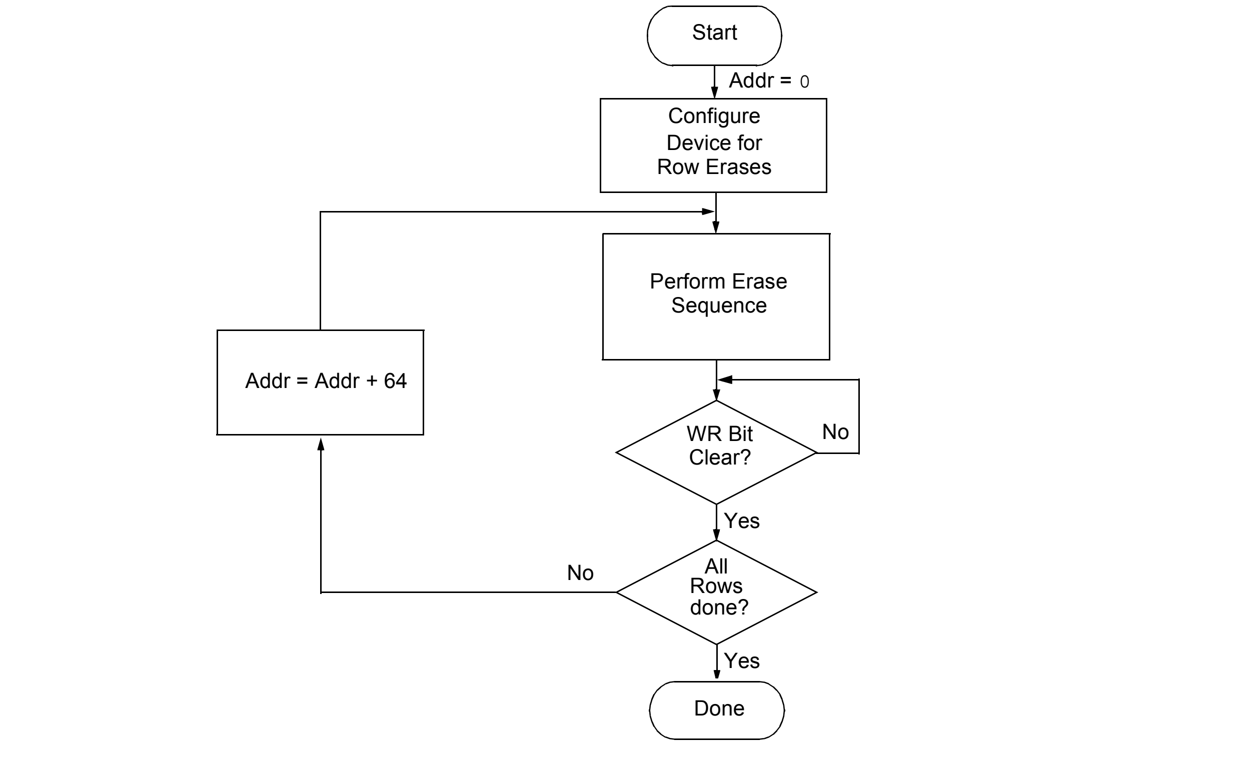
Step 4: Poll WR bit. Repeat until bit is clear.		

`0000`
`0000`
`0000`
`0010`
	50 A6
	6E F5
	00 00
	<MSB><LSB>
		`MOVF EECON1, W, 0`
		`MOVWF TABLAT`
		`NOP`
		`Shift out data`[1]

Step 5: Hold PGC low for time P10.		
Step 6: Repeat step 3 with Address Pointer incremented by 64 until all rows are erased.		
Step 7: Disable writes.		
`0000`	94 A6	`BCF EECON1, WREN`

Note 1: See Figure 5-4 for details on shift out data timing.

FIGURE 4-3: SINGLE ROW ERASE PROGRAM FLASH FLOW



4.2 Program Flash Programming

Programming program Flash is accomplished by first loading data into the write buffer and then initiating a programming sequence. The write and erase buffer sizes shown in Table 4-4 can be mapped to any location of the same size, beginning at 000000h. The actual memory write sequence takes the contents of this buffer and programs the proper amount of program Flash that contains the Table Pointer.

The programming duration is externally timed and is controlled by PGC. After a Start Programming command is issued (4-bit command, '`1111`'), a `NOP` is issued, where the 4th PGC is held high for the duration of the programming time, P9.

After PGC is brought low, the programming sequence is terminated. PGC must be held low for the time specified by parameter P10 to allow high-voltage discharge of the memory array.

The code sequence to program a PIC18F1XK50/PIC18LF1XK50 device is shown in Table 4-5. The flowchart shown in Figure 4-4 depicts the logic necessary to completely write a PIC18F1XK50/PIC18LF1XK50 device. The timing diagram that details the Start Programming command and parameters P9 and P10 is shown in Figure 4-5.

Note:	The TBLPTR register must point to the same region when initiating the programming sequence as it did when the write buffers were loaded.

TABLE 4-4: WRITE AND ERASE BUFFER SIZES

Devices	Write Buffer Size (bytes)	Erase Size (bytes)
PIC18F14K50	16	64
PIC18F13K50	8	64

TABLE 4-5: WRITE PROGRAM FLASH CODE SEQUENCE

4-bit Command	Data Payload	Core Instruction
Step 1: Direct access to program Flash.		

`0000`
`0000`
`0000`
	8E A6
	9C A6
	84 A6
		`BSF   EECON1, EEPGD`
		`BCF   EECON1, CFGS`
		`BSF   EECON1, WREN`

Step 2: Point to row to write.		

`0000`
`0000`
`0000`
`0000`
`0000`
`0000`
	0E <Addr[21:16]>
	6E F8
	0E <Addr[15:8]>
	6E F7
	0E <Addr[7:0]>
	6E F6
		`MOVLW <Addr[21:16]>`
		`MOVWF TBLPTRU`
		`MOVLW <Addr[15:8]>`
		`MOVWF TBLPTRH`
		`MOVLW <Addr[7:0]>`
		`MOVWF TBLPTRL`

Step 3: Load write buffer. Repeat for all but the last two bytes.		
`1101`	<MSB><LSB>	`Write 2 bytes and post-increment address by 2.`
Step 4: Load write buffer for last two bytes and start programming.		

`1111`
`0000`
	<MSB><LSB>
	00 00
		`Write 2 bytes and start programming.`
		`NOP - hold PGC high for time P9 and low for time P10.`

To continue writing data, repeat steps 2 through 4, where the Address Pointer is incremented by 2 at each iteration of the loop.		

FIGURE 4-4: PROGRAM FLASH FLOW

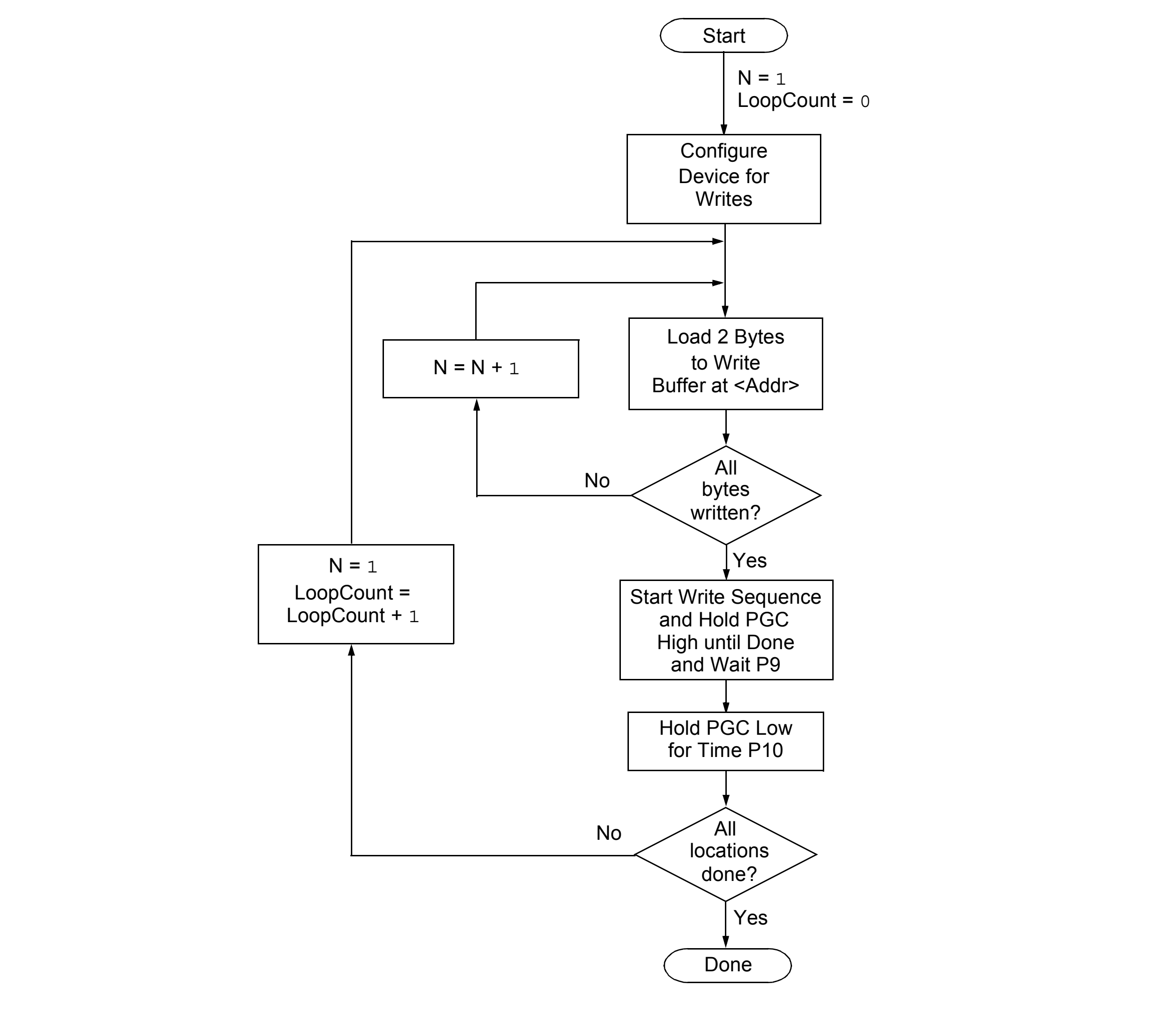
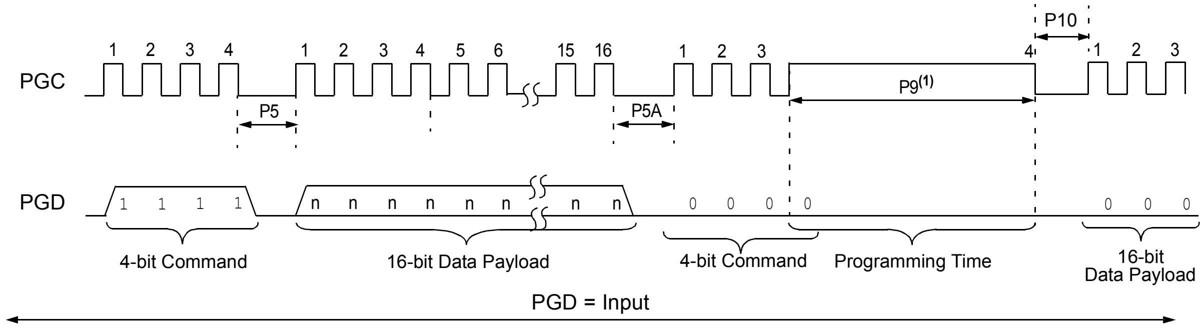

Start
N = 1
LoopCount = 0
Configure Device for Writes
N = N + 1
Load 2 Bytes to Write Buffer at <Addr>
No
All bytes written?
Yes
N = 1
LoopCount = LoopCount + 1
Start Write Sequence and Hold PGC High until Done and Wait P9
Hold PGC Low for Time P10
No
All locations done?
Yes
Done

FIGURE 4-5: TABLE WRITE AND START PROGRAMMING INSTRUCTION TIMING DIAGRAM (1111)



Note 1: Use P9A for Configuration Word programming.

4.2.1 MODIFYING PROGRAM FLASH

The previous programming example assumed that the device has been Bulk Erased prior to programming (see **Section 4.1.1 “High-Voltage ICSP Bulk Erase”**). It may be the case, however, that the user wishes to modify only a section of an already programmed device.

The appropriate number of bytes required for the erase buffer must be read out of program Flash (as described in **Section 5.2 “Verify Program Flash and ID Locations”**) and buffered. Modifications can be made on this buffer. Then, the block of program Flash that was read out must be erased and rewritten with the modified data.

The WREN bit must be set if the WR bit in EECON1 is used to initiate a write sequence.

TABLE 4-6: MODIFYING PROGRAM FLASH

4-bit Command	Data Payload	Core Instruction
Step 1: Direct access to program Flash.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
Step 2: Read program Flash into buffer (**Section 5.1 “Read Program Flash, ID Locations and Configuration Bits”**).		
Step 3: Set the Table Pointer for the block to be erased.		
0000	0E <Addr[21:16]>	MOVLW <Addr[21:16]>
0000	6E F8	MOVWF TBLPTRU
0000	0E <Addr[8:15]>	MOVLW <Addr[8:15]>
0000	6E F7	MOVWF TBLPTRH
0000	0E <Addr[7:0]>	MOVLW <Addr[7:0]>
0000	6E F6	MOVWF TBLPTRL
Step 4: Enable memory writes and setup an erase.		
0000	84 A6	BSF EECON1, WREN
0000	88 A6	BSF EECON1, FREE
Step 5: Initiate erase.		
0000	88 A6	BSF EECON1, FREE
0000	82 A6	BSF EECON1, WR
0000	00 00	NOP
0000	00 00	NOP Erase starts on the 4th clock of this instruction
Step 6: Poll WR bit. Repeat until bit is clear.		
0000	50 A6	MOVF EECON1, W, 0
0000	6E F5	MOVWF TABLAT
0000	00 00	NOP
0000	<MSB><LSB>	Shift out data(1)
Step 7: Load write buffer. The correct bytes will be selected based on the Table Pointer.		
0000	0E <Addr[21:16]>	MOVLW <Addr[21:16]>
0000	6E F8	MOVWF TBLPTRU
0000	0E <Addr[8:15]>	MOVLW <Addr[8:15]>
0000	6E F7	MOVWF TBLPTRH
0000	0E <Addr[7:0]>	MOVLW <Addr[7:0]>
0000	6E F6	MOVWF TBLPTRL
1101	<MSB><LSB>	Write 2 bytes and post-increment address by 2.
•	•	
•	•	Repeat as many times as necessary to fill the write buffer
•	•	Write 2 bytes and start programming.
1111	<MSB><LSB>	NOP - hold PGC high for time P9 and low for time P10.
0000	00 00	
To continue modifying data, repeat Steps 2 through 6, where the Address Pointer is incremented by the appropriate number of bytes (see Table 4-4) at each iteration of the loop. The write cycle must be repeated enough times to completely rewrite the contents of the erase buffer.		
Step 8: Disable writes.		
0000	94 A6	BCF EECON1, WREN

4.3 Data EEPROM Programming

Data EEPROM is accessed one byte at a time via an Address Pointer (register pair EEADRH:EEADR) and a data latch (EEDATA). Data EEPROM is written by loading EEADRH:EEADR with the desired memory location, EEDATA, with the data to be written and initiating a memory write by appropriately configuring the EECON1 register. A byte write automatically erases the location and writes the new data (erase-before-write).

When using the EECON1 register to perform a data EEPROM write, both the EEPGD and CFGS bits must be cleared (EECON1<7:6> = `00`). The WREN bit must be set (EECON1<2> = `1`) to enable writes of any sort and this must be done prior to initiating a write sequence. The write sequence is initiated by setting the WR bit (EECON1<1> = `1`).

The write begins on the falling edge of the 24th PGC after the WR bit is set. It ends when the WR bit is cleared by hardware.

After the programming sequence terminates, PGC must be held low for the time specified by parameter P10 to allow high-voltage discharge of the memory array.

FIGURE 4-6: PROGRAM DATA FLOW

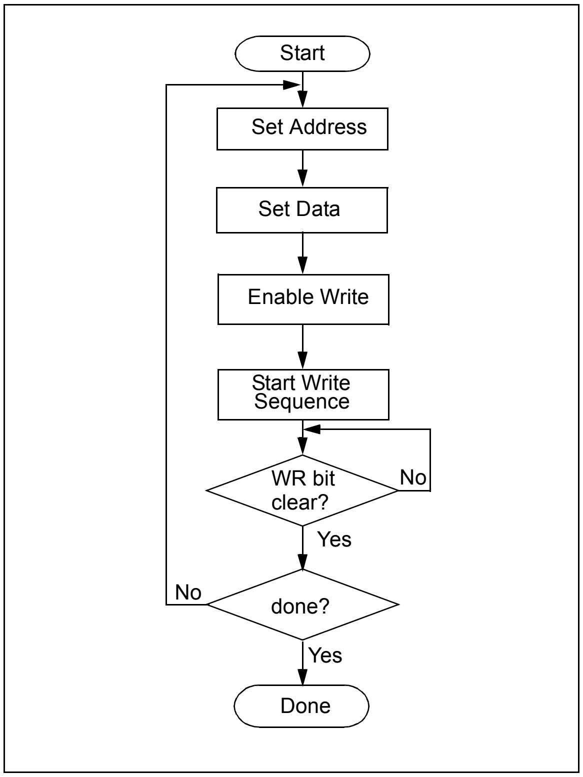


FIGURE 4-7: DATA EEPROM WRITE TIMING DIAGRAM

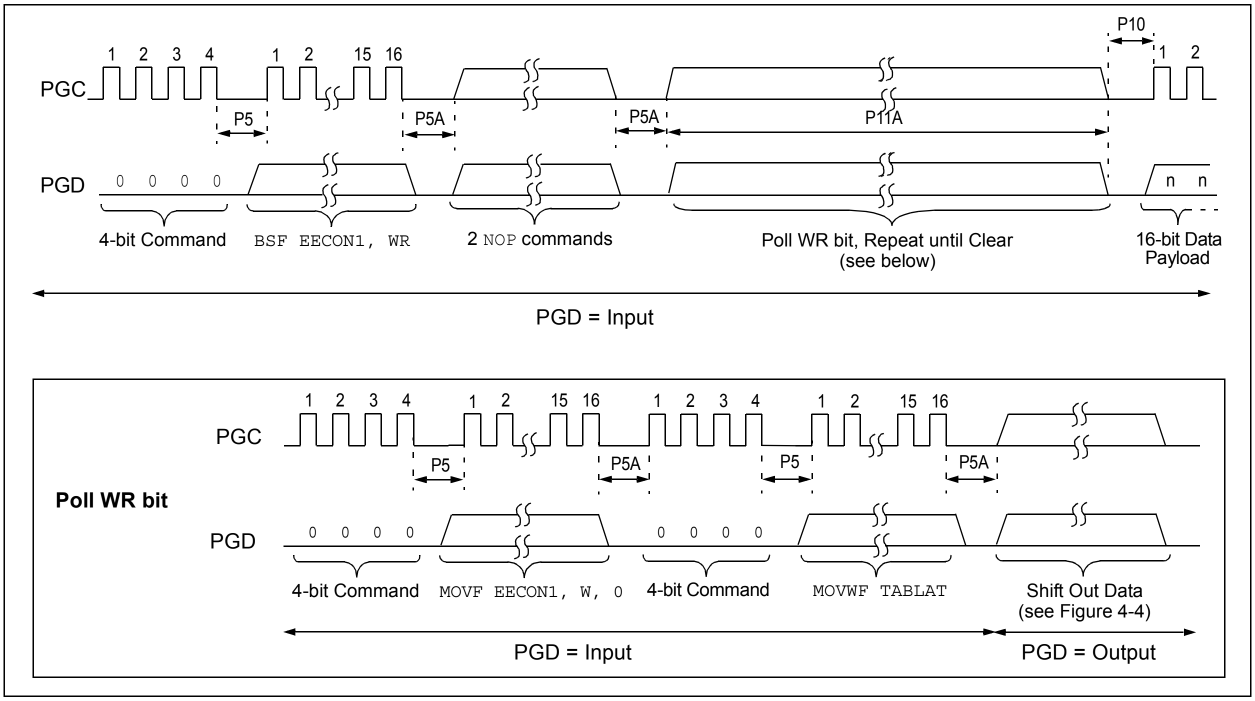


TABLE 4-7: PROGRAMMING DATA MEMORY

4-bit Command	Data Payload	Core Instruction
Step 1: Direct access to data EEPROM.		

0000
0000
	9E A6
	9C A6
		`BCF EECON1, EEPGD`
		`BCF EECON1, CFGS`

Step 2: Set the data EEPROM Address Pointer.		

0000
0000
0000
0000
	0E <Addr>
	6E A9
	OE <AddrH>
	6E AA
		`MOVLW <Addr>`
		`MOVWF EEADR`
		`MOVLW <AddrH>`
		`MOVWF EEADRH`

Step 3: Load the data to be written.		

0000
0000
	0E <Data>
	6E A8
		`MOVLW <Data>`
		`MOVWF EEDATA`

Step 4: Enable memory writes.		
0000	84 A6	`BSF EECON1, WREN`
Step 5: Initiate write.		

0000
0000
0000
	82 A6
	00 00
	00 00
		`BSF EECON1, WR`
		`NOP`
		`NOP ;write starts on 4th clock of this instruction`

Step 6: Poll WR bit, repeat until the bit is clear.		

0000
0000
0000
0010
	50 A6
	6E F5
	00 00
	<MSB><LSB>
		`MOVF EECON1, W, 0`
		`MOVWF TABLAT`
		`NOP`
		`Shift out data`(1)

Step 7: Hold PGC low for time P10.		
Step 8: Disable writes.		
0000	94 A6	`BCF EECON1, WREN`
Repeat steps 2 through 8 to write more data.		

Note 1: See Figure 5-4 for details on shift out data timing.

4.4 ID Location Programming

The ID locations are programmed much like the program Flash. The ID registers are mapped in addresses 200000h through 200007h. These locations read out normally even after code protection.

Note:	The user only needs to fill the first 8 bytes of the write buffer, in order to write the ID locations.

Table 4-8 demonstrates the code sequence, required to write the ID locations.

In order to modify the ID locations, refer to the methodology described in **Section 4.2.1 "Modifying Program Flash"**. As with program Flash, the ID locations must be erased before being modified.

TABLE 4-8: WRITE ID SEQUENCE

4-bit Command	Data Payload	Core Instruction
Step 1: Direct access to program Flash.		

`0000`
`0000`
`0000`
	8E A6
	9C A6
	84 A6
		`BSF EECON1, EEPGD`
		`BCF EECON1, CFGS`
		`BSF EECON1, WREN`

Step 2: Set Table Pointer to ID. Load write buffer with 8 bytes and write.		

`0000`
`0000`
`0000`
`0000`
`0000`
`0000`
`1101`
`1101`
`1101`
`1111`
`0000`
	0E 20
	6E F8
	0E 00
	6E F7
	0E 00
	6E F6
	<MSB><LSB>
	<MSB><LSB>
	<MSB><LSB>
	<MSB><LSB>
	00 00
		`MOVLW 20h`
		`MOVWF TBLPTRU`
		`MOVLW 00h`
		`MOVWF TBLPTRH`
		`MOVLW 00h`
		`MOVWF TBLPTRL`
		`Write 2 bytes and post-increment address by 2.`
		`Write 2 bytes and post-increment address by 2.`
		`Write 2 bytes and post-increment address by 2.`
		`Write 2 bytes and start programming.`
		`NOP - hold PGC high for time P9 and low for time P10.`

4.5 Boot Block Programming

The code sequence detailed in Table 4-5 should be used, except that the address used in "Step 2" will be in the range of 000000h to 0007FFh.

4.6 Configuration Bits Programming

Unlike program Flash, the Configuration bits are programmed a byte at a time. The Table Write, Begin Programming 4-bit command ('`1111`') is used, but only 8 bits of the following 16-bit payload will be written. The LSB of the payload will be written to even addresses and the MSB will be written to odd addresses. The code sequence to program two consecutive configuration locations is shown in Table 4-9. See Figure 4-5 for the timing diagram.

Note:	The address must be explicitly written for each byte programmed. The addresses cannot be incremented in this mode.

TABLE 4-9: SET ADDRESS POINTER TO CONFIGURATION LOCATION

4-bit Command	Data Payload	Core Instruction
Step 1: Direct access to config memory.		

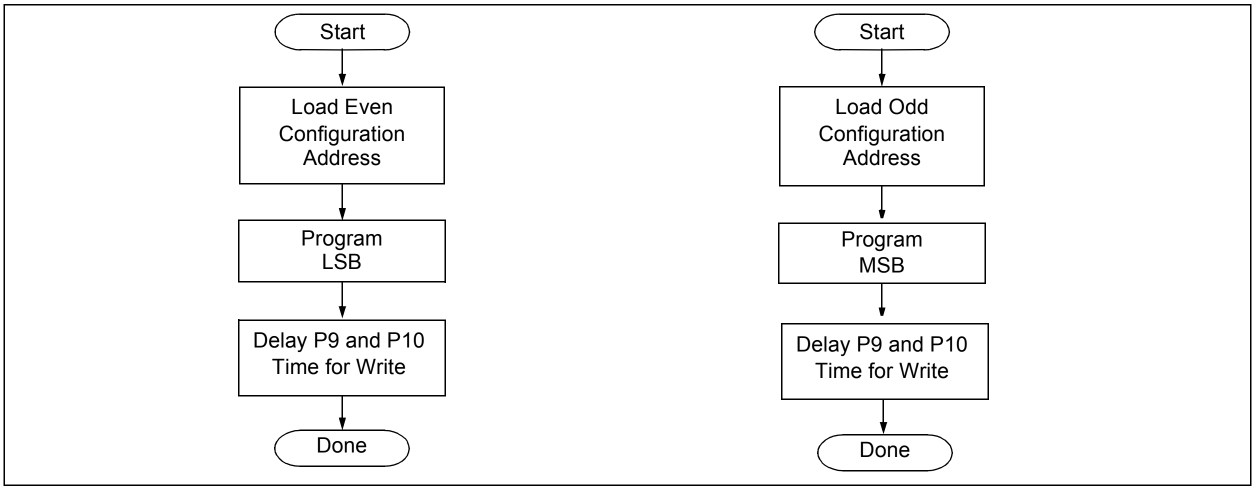
`0000`
`0000`
`0000`
	8E A6
	8C A6
	84 A6
		`BSF EECON1, EEPGD`
		`BSF EECON1, CFGS`
		`BSF EECON1, WREN`

Step 2[1]: Set Table Pointer for config byte to be written. Write even/odd addresses.		

`0000`
`0000`
`0000`
`0000`
`0000`
`0000`
`1111`
`0000`
`0000`
`0000`
`1111`
`0000`
	0E 30
	6E F8
	0E 00
	6E F7
	0E 00
	6E F6
	<MSB ignored><LSB>
	00 00
	0E 01
	6E F6
	<MSB><LSB ignored>
	00 00
		`MOVLW 30h`
		`MOVWF TBLPTRU`
		`MOVLW 00h`
		`MOVWF TBLPRTH`
		`MOVLW 00h`
		`MOVWF TBLPTRL`
		`Load 2 bytes and start programming.`
		`NOP - hold PGC high for time P9 and low for time P10.`
		`MOVLW 01h`
		`MOVWF TBLPTRL`
		`Load 2 bytes and start programming.`
		`NOP - hold PGC high for time P9A and low for time P10.`

Note 1: Enabling the write protection of Configuration bits (WRTC = 0 in CONFIG6H) will prevent further writing of Configuration bits. Always write all the Configuration bits before enabling the write protection for Configuration bits.

FIGURE 4-8: CONFIGURATION PROGRAMMING FLOW


Start
Load Even Configuration Address
Program LSB
Delay P9 and P10 Time for Write
Done
Start
Load Odd Configuration Address
Program MSB
Delay P9 and P10 Time for Write
Done

5.0 READING THE DEVICE

5.1 Read Program Flash, ID Locations and Configuration Bits

Program Flash is accessed one byte at a time via the 4-bit command, ‘1001’ (table read, post-increment). The contents of memory pointed to by the Table Pointer (TBLPTRU:TBLPTRH:TBLPTRL) are serially output on PGD.

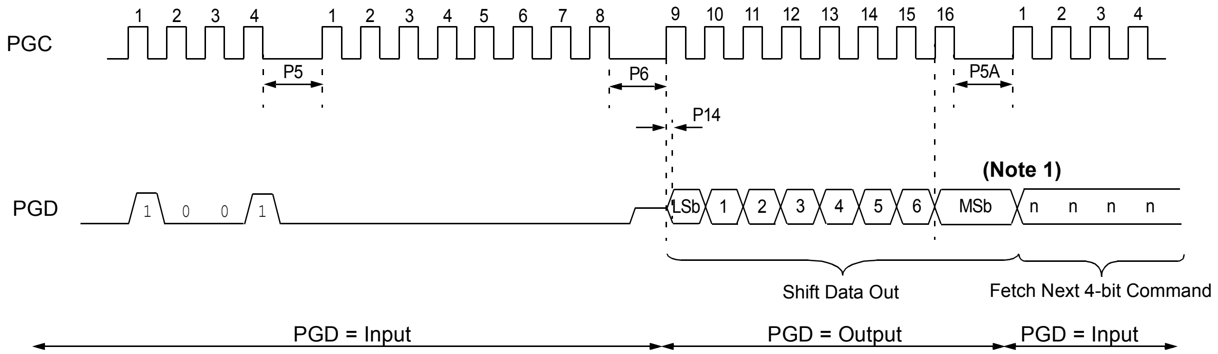
The 4-bit command is shifted in LSb first. The read is executed during the next 8 clocks, then shifted out on PGD during the last 8 clocks, LSb to MSb. A delay of P6 must be introduced after the falling edge of the 8th PGC of the operand to allow PGD to transition from an input to an output. During this time, PGC must be held low, as illustrated in Figure 5-1. This operation also increments the Table Pointer by one, pointing to the next byte in program Flash for the next read.

This technique will work to read any memory in the 000000h to 3FFFFFh address space, so it also applies to the reading of the ID and Configuration registers.

TABLE 5-1: READ PROGRAM FLASH SEQUENCE

4-bit Command	Data Payload	Core Instruction
Step 1: Set Table Pointer		
0000 0000 0000 0000 0000 0000	0E <Addr[21:16]> 6E F8 0E <Addr[15:8]> 6E F7 0E <Addr[7:0]> 6E F6	MOVLW Addr[21:16] MOVWF TBLPTRU MOVLW <Addr[15:8]> MOVWF TBLPTRH MOVLW <Addr[7:0]> MOVWF TBLPTRL
Step 2: Read memory and then shift out on PGD, LSb to MSb		
1001	00 00	TBLRD *+

FIGURE 5-1: TABLE READ POST-INCREMENT INSTRUCTION TIMING DIAGRAM (1001)



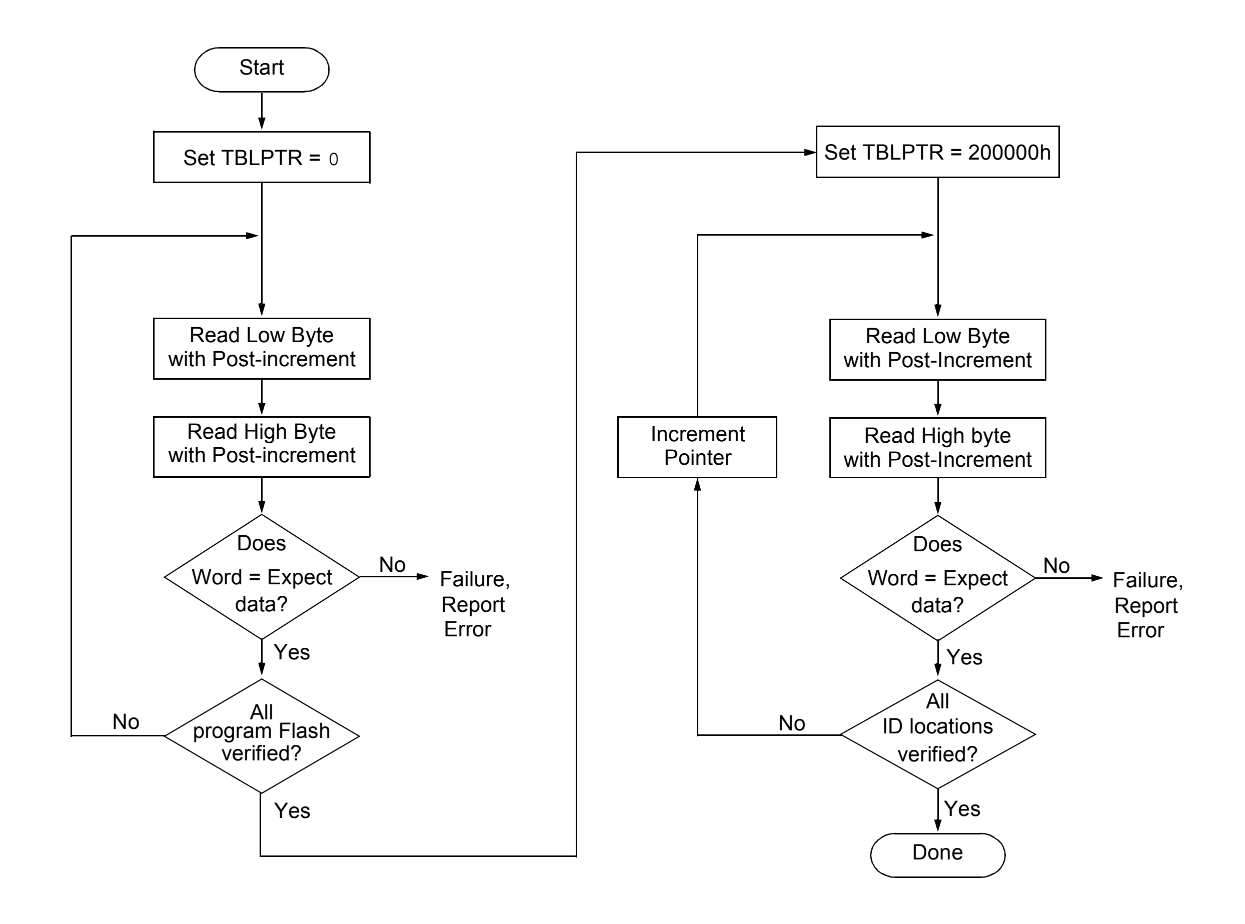
Note 1: Magnification of the high-impedance delay between PGC and PGD is shown in Figure 5-5.

5.2 Verify Program Flash and ID Locations

The verify step involves reading back the program Flash space and comparing it against the copy held in the programmer's buffer. Memory reads occur a single byte at a time, so two bytes must be read to compare against the word in the programmer's buffer. Refer to **Section 5.1 "Read Program Flash, ID Locations and Configuration Bits"** for implementation details of reading program Flash.

The Table Pointer must be manually set to 200000h (base address of the ID locations) once the program Flash has been verified. The post-increment feature of the Table Read 4-bit command can not be used to increment the Table Pointer beyond the program Flash space. In a 64-Kbyte device, for example, a post-increment read of address FFFFh will wrap the Table Pointer back to 000000h, rather than point to unimplemented address, 010000h.

FIGURE 5-2: VERIFY PROGRAM FLASH FLOW


Start
Set TBLPTR = 0
Read Low Byte with Post-increment
Read High Byte with Post-increment
Does Word = Expect data?
No
Failure, Report Error
Yes
All program Flash verified?
No
Yes
Set TBLPTR = 200000h
Read Low Byte with Post-Increment
Read High byte with Post-Increment
Increment Pointer
Does Word = Expect data?
No
Failure, Report Error
Yes
All ID locations verified?
No
Yes
Done

5.3 Verify Configuration Bits

A Configuration address may be read and output on PGD via the 4-bit command, ‘`1001`’. Configuration data is read and written in a byte-wise fashion, so it is not necessary to merge two bytes into a word prior to a compare. The result may then be immediately compared to the appropriate Configuration data in the programmer’s memory for verification. Refer to **Section 5.1 “Read Program Flash, ID Locations and Configuration Bits”** for implementation details of reading Configuration data.

5.4 Read Data EEPROM Memory

Data EEPROM is accessed one byte at a time via an Address Pointer (register pair EEADRH:EEADR) and a data latch (EEDATA). Data EEPROM is read by loading EEADRH: EEADR with the desired memory location and initiating a memory read by appropriately configuring the EECON1 register. The data will be loaded into EEDATA, where it may be serially output on PGD via the 4-bit command, ‘`0010`’ (Shift Out Data Holding register). A delay of P6 must be introduced after the falling edge of the 8th PGC of the operand to allow PGD to transition from an input to an output. During this time, PGC must be held low, as shown in Figure 5-4.

The command sequence to read a single byte of data is shown in Table 5-2.

FIGURE 5-3: READ DATA EEPROM FLOW

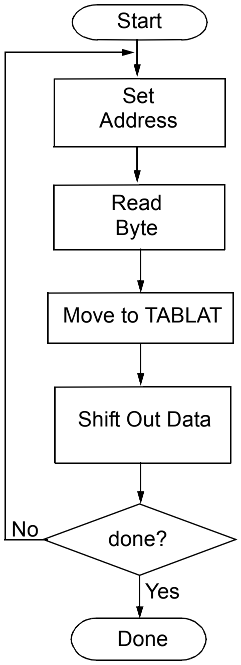


TABLE 5-2: READ DATA EEPROM MEMORY

4-bit Command	Data Payload	Core Instruction
Step 1: Direct access to data EEPROM.		

`0000`
`0000`
	9E A6
	9C A6
		`BCF EECON1, EEPGD`
		`BCF EECON1, CFGS`

Step 2: Set the data EEPROM Address Pointer.		

`0000`
`0000`
`0000`
`0000`
	0E <Addr>
	6E A9
	OE <AddrH>
	6E AA
		`MOVLW <Addr>`
		`MOVWF EEADR`
		`MOVLW <AddrH>`
		`MOVWF EEADRH`

Step 3: Initiate a memory read.		
`0000`	80 A6	`BSF EECON1, RD`
Step 4: Load data into the Serial Data Holding register.		

`0000`
`0000`
`0000`
`0010`
	50 A8
	6E F5
	00 00
	<MSB><LSB>
		`MOVF EEDATA, W, 0`
		`MOVWF TABLAT`
		`NOP`
		`Shift Out Data`(1)

Note 1: The <LSB> is undefined. The <MSB> is the data.