PIC18FXX39

Programming for PIC18FXX39 Flash MCUs

1.0 DEVICE OVERVIEW

This document includes the programming specifications for the following devices:

- PIC18F2439
- PIC18F2539
- PIC18F4439
- PIC18F4539

2.0 PROGRAMMING OVERVIEW OF THE PIC18FXX39

The PIC18FXX39 can be programmed using the high voltage In-Circuit Serial Programming™ (ICSP™) method, or the low voltage ICSP method; both while in the users' system. The low voltage ICSP method is slightly different than the high voltage method, and these differences are noted where applicable. This programming specification applies to PIC18FXX39 devices in all package types.

2.1 Hardware Requirements

In High Voltage ICSP mode, the PIC18FXX39 requires two programmable power supplies: one for VDD and one for MCLR/VPP. Both supplies should have a minimum resolution of 0.25V. Refer to Section 6.0 for additional hardware parameters.

2.1.1 LOW VOLTAGE ICSP PROGRAMMING

In Low Voltage ICSP mode, the PIC18FXX39 can be programmed using a VDD source in the operating range. This only means that MCLR/VPP does not have to be brought to a different voltage, but can instead be left at the normal operating voltage. Refer to Section 6.0 for additional hardware parameters.

2.2 Pin Diagrams

The pin diagrams for the PIC18FXX39 family are shown in Figure 2-1. The pin descriptions of these diagrams do not represent the complete functionality of the device types. One should refer to the appropriate device data sheet for complete pin descriptions.

TABLE 2-1: PIN DESCRIPTIONS (DURING PROGRAMMING): PIC18FXX39

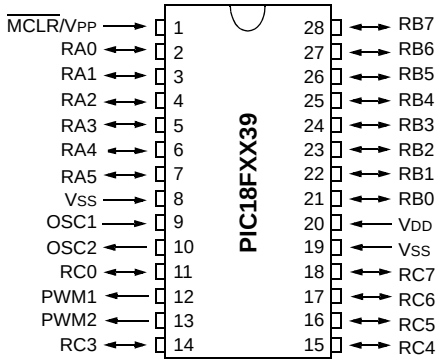
Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
MCLR/VPP	VPP	P	Programming Enable
VDD	VDD	P	Power Supply
Vss	VSS	P	Ground
RB5	PGM	I	Low Voltage ICSP™ Input when LVP Configuration bit equals '1'(1)
RB6	SCLK	I	Serial Clock
RB7	SDATA	I/O	Serial Data

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

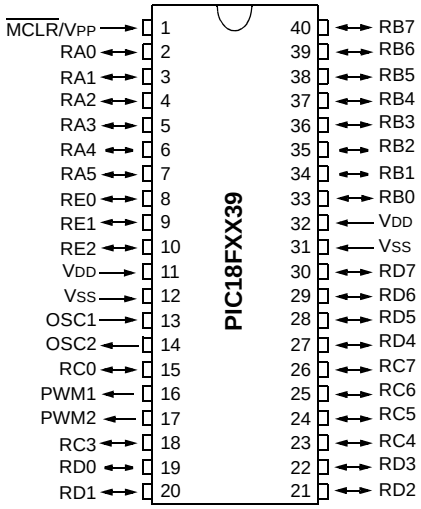
Note 1: See Section 5.3 for more detail.

FIGURE 2-1: PIC18FXX39 FAMILY PIN DIAGRAMS

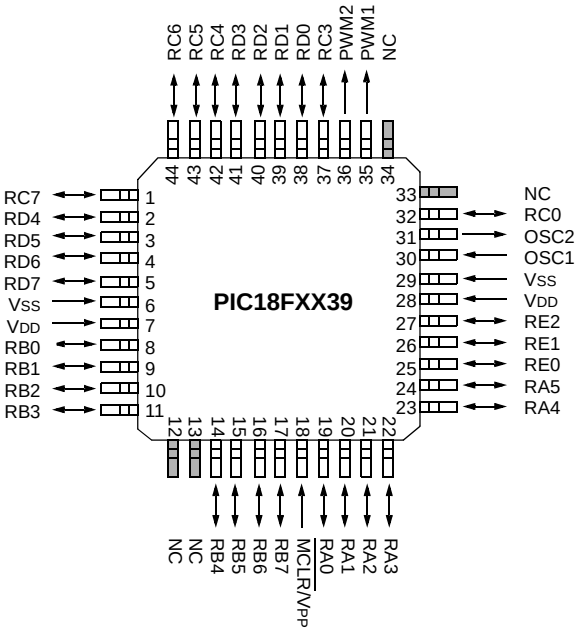
28-Pin DIP, SOIC



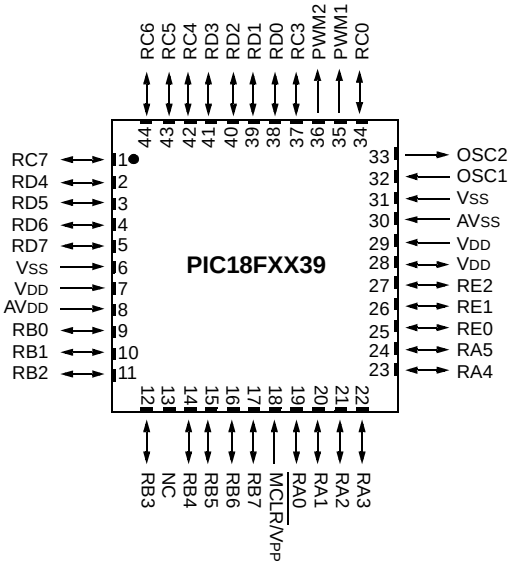
40-Pin DIP



44-Pin TQFP



44-Pin QFN



Note: Not all multiplexed pin definitions are shown. Refer to the appropriate data sheet for complete pin descriptions.

2.3 Memory Map

The code memory space extends from 0000h to 5FFFh (24 Kbytes) in three 8-Kbyte panels in PIC18FX539 parts. In PIC18FX439 parts, program space is from 0000h to 2FFFh (12 Kbytes). Addresses 0000h through 01FFh, however, define a “Boot Block” region that is treated separately from Panel 1. All code memory is on-chip.

A user 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 300001h through 30000Dh are reserved for the configuration bits. These bits may be set to select various device options, and are described in Section 5.0. These configuration bits read out normally, even after code protected.

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 5.0. These configuration bits read out normally, even after code protection.

TABLE 2-2: IMPLEMENTATION OF CODE MEMORY

Device	Code Memory Size
PIC18F2439	0000h - 2FFFh (12 Kbytes)
PIC18F2539	0000h - 5FFFh (24 Kbytes)
PIC18F4439	0000h - 2FFFh (12 Kbytes)
PIC18F4539	0000h - 5FFFh (24 Kbytes)

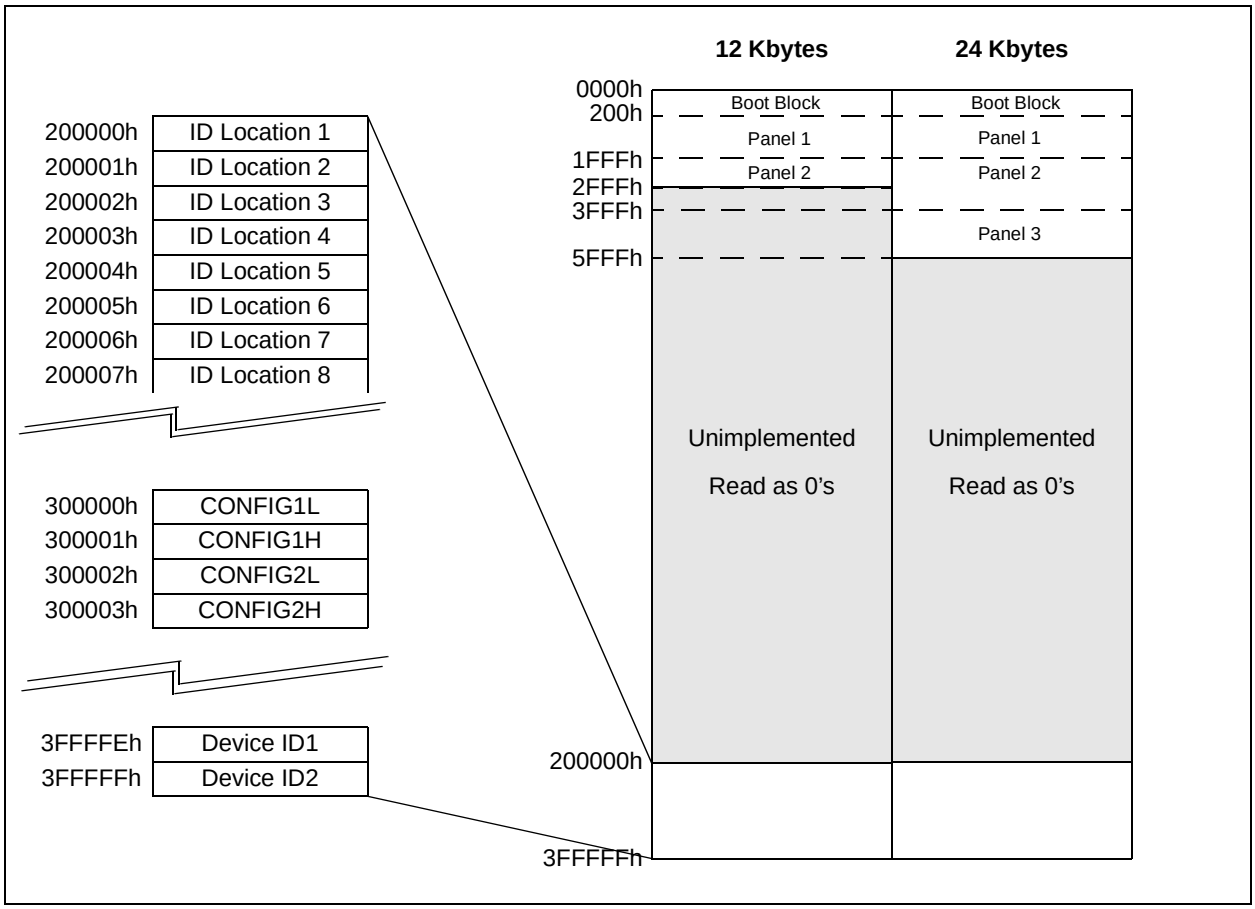
2.3.1 MEMORY ADDRESS POINTER

Memory in the address space 000000h to 3FFFFFh is addressed via the Table Pointer, which is comprised of three pointer registers:

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

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

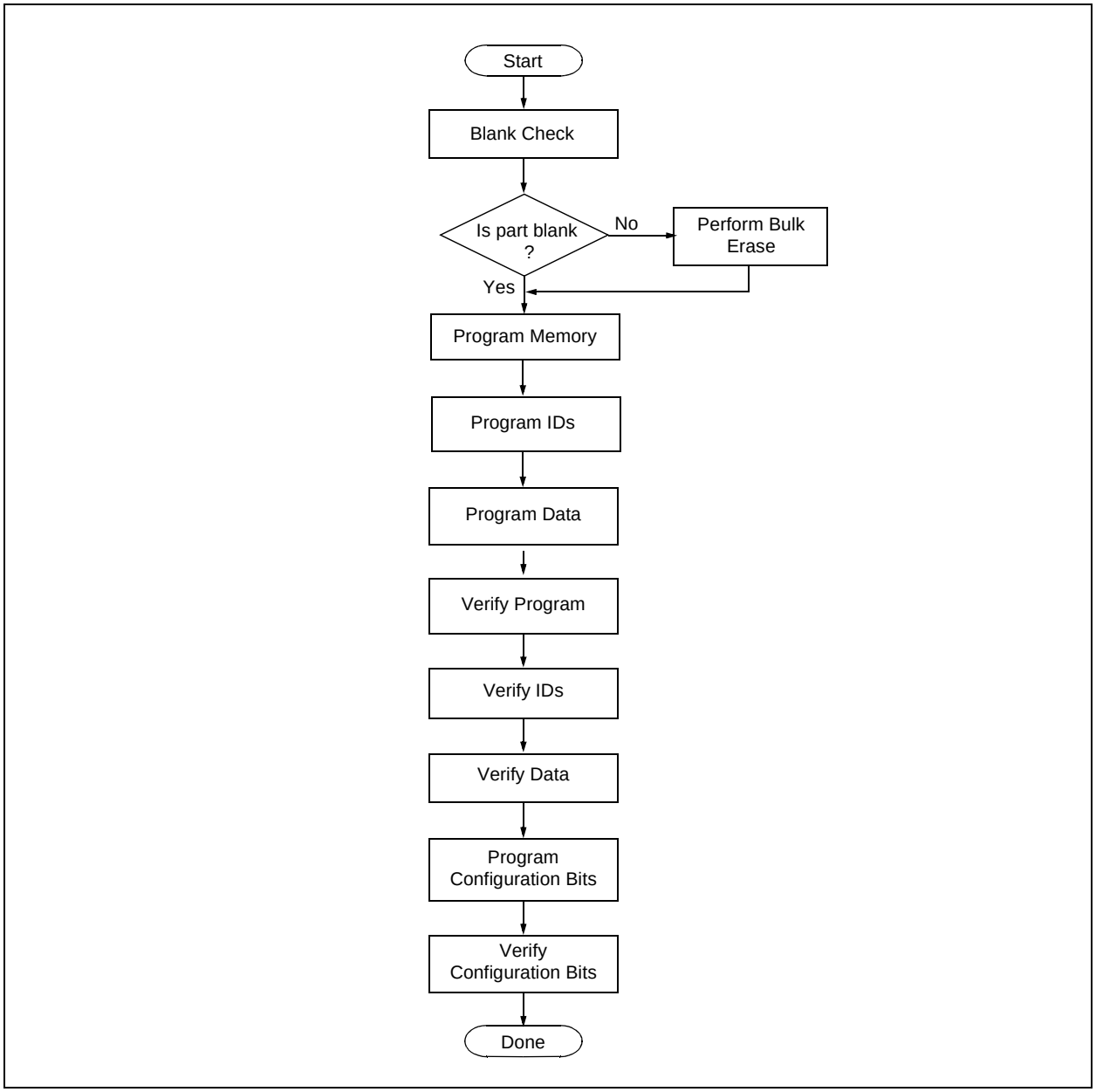
FIGURE 2-2: MEMORY MAP FOR PIC18FXX39



2.4 High Level Overview of the Programming Process

Figure 2-3 shows the high level overview of the programming process. The device is first checked to see if it is blank; if it is not, a bulk erase is performed. Next, the code memory, 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 2-3: HIGH LEVEL PROGRAMMING FLOW

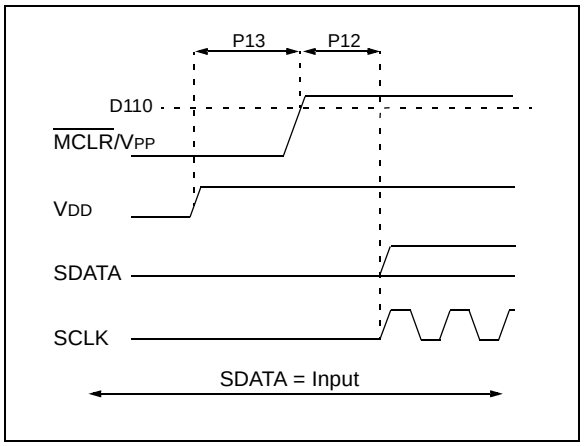


2.5 Entering High Voltage ICSP Program/Verify Mode

The High Voltage ICSP Program/Verify mode is entered by holding SCLK and SDATA low, and then raising MCLR/VPP to VIHH (high voltage). Once in this mode, the code memory, data EEPROM, ID locations, and configuration bits can be accessed and programmed in serial fashion.

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

FIGURE 2-4: ENTERING HIGH VOLTAGE PROGRAM/ VERIFY MODE

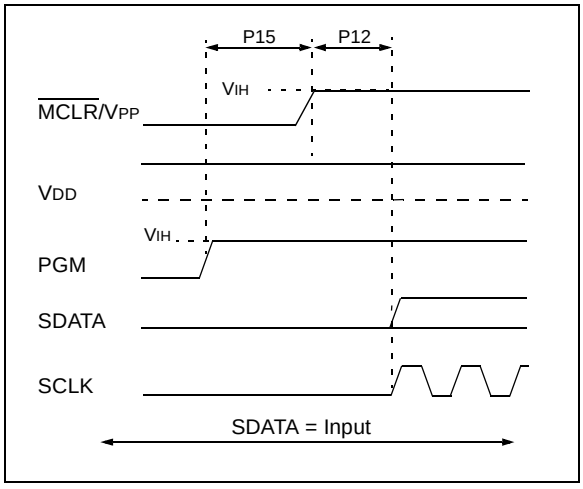


2.5.1 ENTERING LOW VOLTAGE ICSP PROGRAM/VERIFY MODE

When the LVP configuration bit is '1' (see Section 5.3), the Low Voltage ICSP mode is enabled. Low Voltage ICSP Program/Verify mode is entered by holding SCLK and SDATA low, placing a logic high on PGM, and then raising MCLR/VPP to VIH. In this mode, the RB5/PGM pin is dedicated to the programming function and ceases to be a general purpose I/O pin.

The sequence that enters the device into the Programming/Verify mode places all unused I/O's in the high impedance state.

FIGURE 2-5: ENTERING LOW VOLTAGE PROGRAM/ VERIFY MODE



2.6 Serial Program/Verify Operation

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

2.6.1 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, SCLK is cycled four times. The commands needed for programming and verification are shown in Table 2-3.

TABLE 2-3: 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, post-decrement by 2	1110
Table Write, start programming	1111

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 2-4. The 4-bit command is shown MSb first. The command operand, or “Data Payload”, is shown <MSB><LSB>. Figure 2-6 demonstrates how to serially present a 20-bit command/operand to the device.

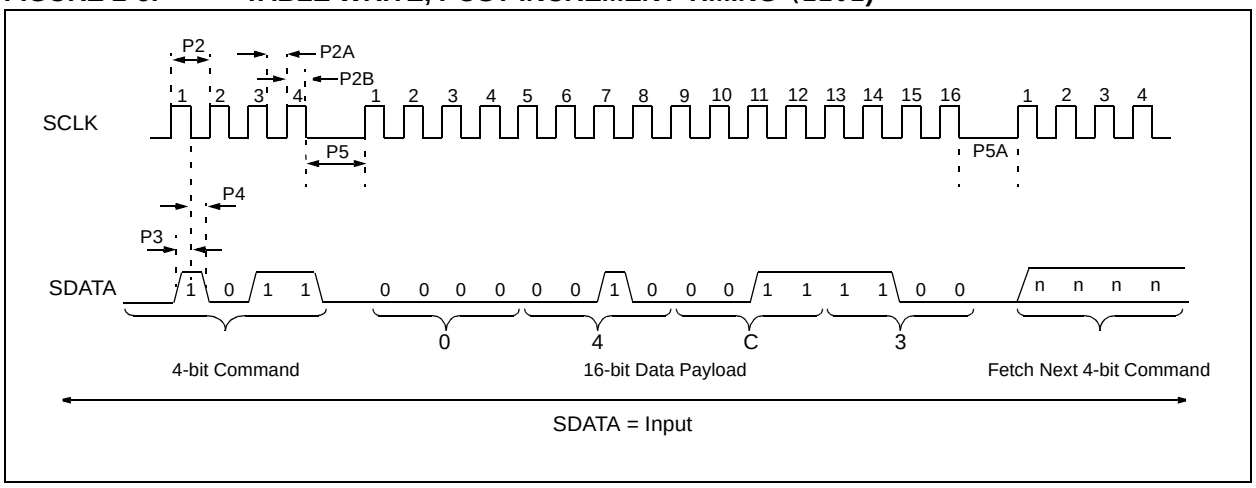
TABLE 2-4: SAMPLE COMMAND SEQUENCE

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

2.6.2 CORE INSTRUCTION

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

FIGURE 2-6: TABLE WRITE, POST-INCREMENT TIMING (1101)



3.0 DEVICE PROGRAMMING

3.1 Blank Check

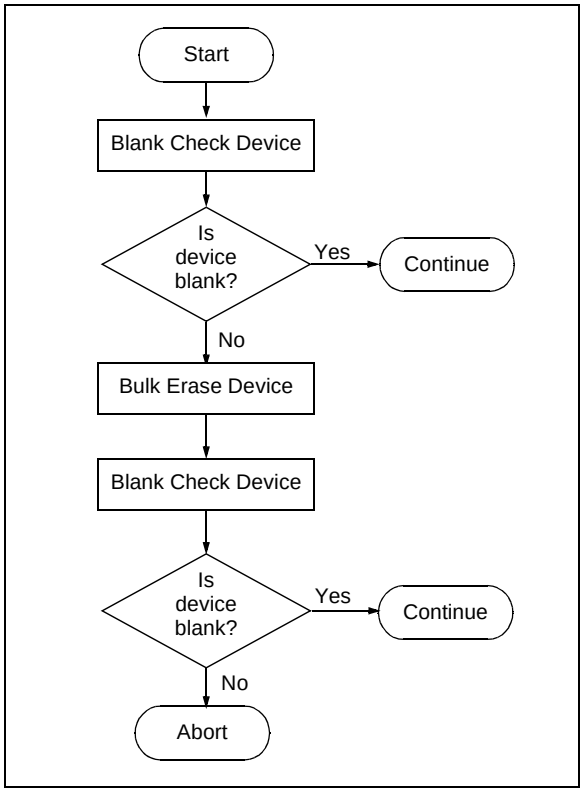
The term, “Blank Check”, means to verify that the device has no programmed memory cells. All memories must be verified: code memory, data EEPROM, ID locations, and configuration bits. The Device ID registers (3FFFFEh:3FFFFFh) should be ignored.

A “blank” or “erased” memory cell will read as a ‘1’. So, “Blank Checking” a device merely means to verify that all bytes read as FFh, except the configuration bits. Unused (reserved) configuration bits will read ‘0’ (programmed). Refer to Table 5-2 for blank configuration expect data for the various PIC18FXX39 devices.

If it is determined that the device is not blank, then the device should be Bulk Erased (see Section 3.2) before any attempt to program is made.

Given that “Blank Checking” is merely code and data EEPROM verification with FFh expect data, refer to Section 4.1 and Section 4.3 for implementation details.

FIGURE 3-1: BLANK CHECK FLOW



3.2 High Voltage ICSP Bulk Erase

Erasing code or data EEPROM is accomplished by writing an “erase option” to address 3C0004h. Code memory 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 3-1.

TABLE 3-1: BULK ERASE OPTIONS

Description	Data
Erase Data EEPROM	81h
Erase Boot Block	83h
Erase Panel 1	88h
Erase Panel 2 (PIC18FX539 only)	89h
Erase Panel 3 (PIC18FX539 only)	8Ah

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

The code sequence to erase the entire device is shown in Table 3-2 and Table 3-3. The corresponding flowcharts are shown in Figure 3-2 and Figure 3-3.

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

TABLE 3-2: CHIP ERASE COMMAND SEQUENCE FOR PIC18FX439

4-bit Command	Data Payload	Core Instruction
Step 1: Load 3C0004h to Address Pointer.		
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
Step 2: Erase boot block.		
0000	00 83	Write 83h to 3C0004h to erase the boot block
0000	00 00	NOP
0000	00 00	Hold SDATA low until erase complete
Step 3: Erase Panel 1.		
0000	00 88	Write 88h to 3C0004h to erase the panel 1
0000	00 00	NOP
0000	00 00	Hold SDATA low until erase complete
Step 4: Configure device for single panel write.		
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 06	MOVLW 06h
0000	6E F6	MOVWF TBLPTRL
1100	00 00	Write 00h to 3C0006h to enable single panel writes
Step 5: Direct access to code memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
Step 6: Set the Table Pointer to point to the first 64-byte block of Panel 2.		
0000	0E 20	MOVLW 20h
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 00	MOVLW 00h
0000	6E F6	MOVWF TBLPTRL
Step 7: Enable memory writes and setup an erase.		
0000	84 A6	BSF EECON1, WREN
0000	88 A6	BSF EECON1, FREE
Step 8: Perform Data EEPROM unlock sequence.		
0000	0E 55	MOVLW 55h
0000	6E A7	MOVWF EECON2
0000	0E AA	MOVLW AAh
0000	6E A7	MOVWF EECON2
Step 9: Initiate erase.		
0000	82 A6	BSF EECON1, WR
0000	00 00	NOP
Step 10: Wait for P11+P10 and then disable writes.		
0000	94 A6	BCF EECON1, WREN
Step 11: Increment Table Pointer by 64 and repeat from step 7 to step 10, 40 times.		

FIGURE 3-2: CHIP ERASE FLOW FOR PIC18FX439

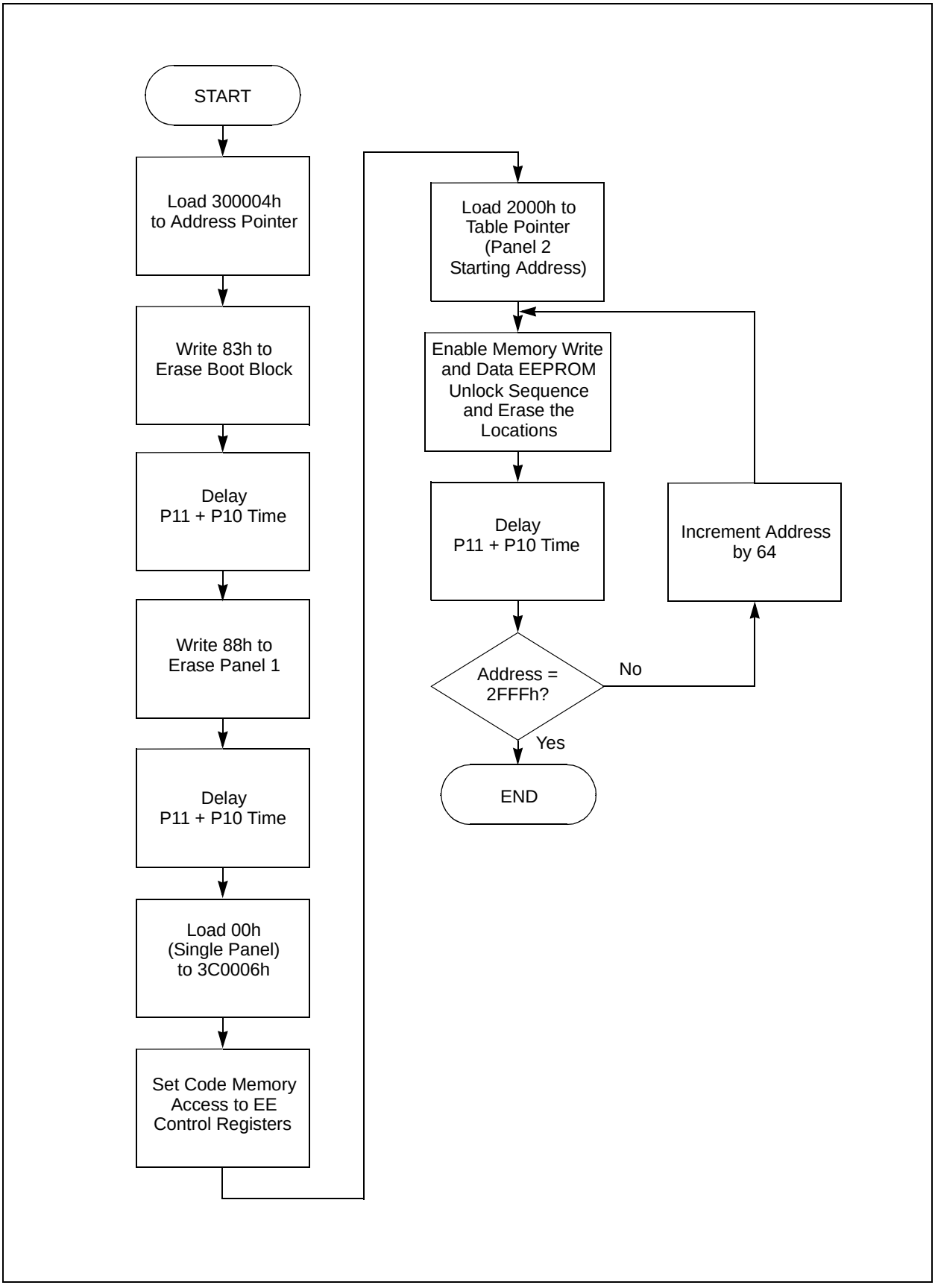


TABLE 3-3: CHIP ERASE COMMAND SEQUENCE FOR PIC18FX539

4-bit Command	Data Payload	Core Instruction
Step 1: Load 3C0004h to Address Pointer.		
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
Step 2: Erase boot block.		
0000	00 83	Write 83h to 3C0004h to erase the boot block
0000	00 00	NOP
0000	00 00	Hold SDATA low until erase complete (P11+P10)
Step 3: Erase Panel 1.		
0000	00 88	Write 88h to 3C0004h to erase the panel 1
0000	00 00	NOP
0000	00 00	Hold SDATA low until erase complete (P11+P10)
Step 4: Erase Panel 2.		
0000	00 89	Write 89h to 3C0004h to erase the panel 2
0000	00 00	NOP
0000	00 00	Hold SDATA low until erase complete (P11+P10)
Step 5: Erase Panel 3.		
0000	00 8A	Write 8Ah to 3C0004h to erase the panel 3
0000	00 00	NOP
0000	00 00	Hold SDATA low until erase complete (P11+P10)

FIGURE 3-3: CHIP ERASE FLOW FOR PIC18FX539

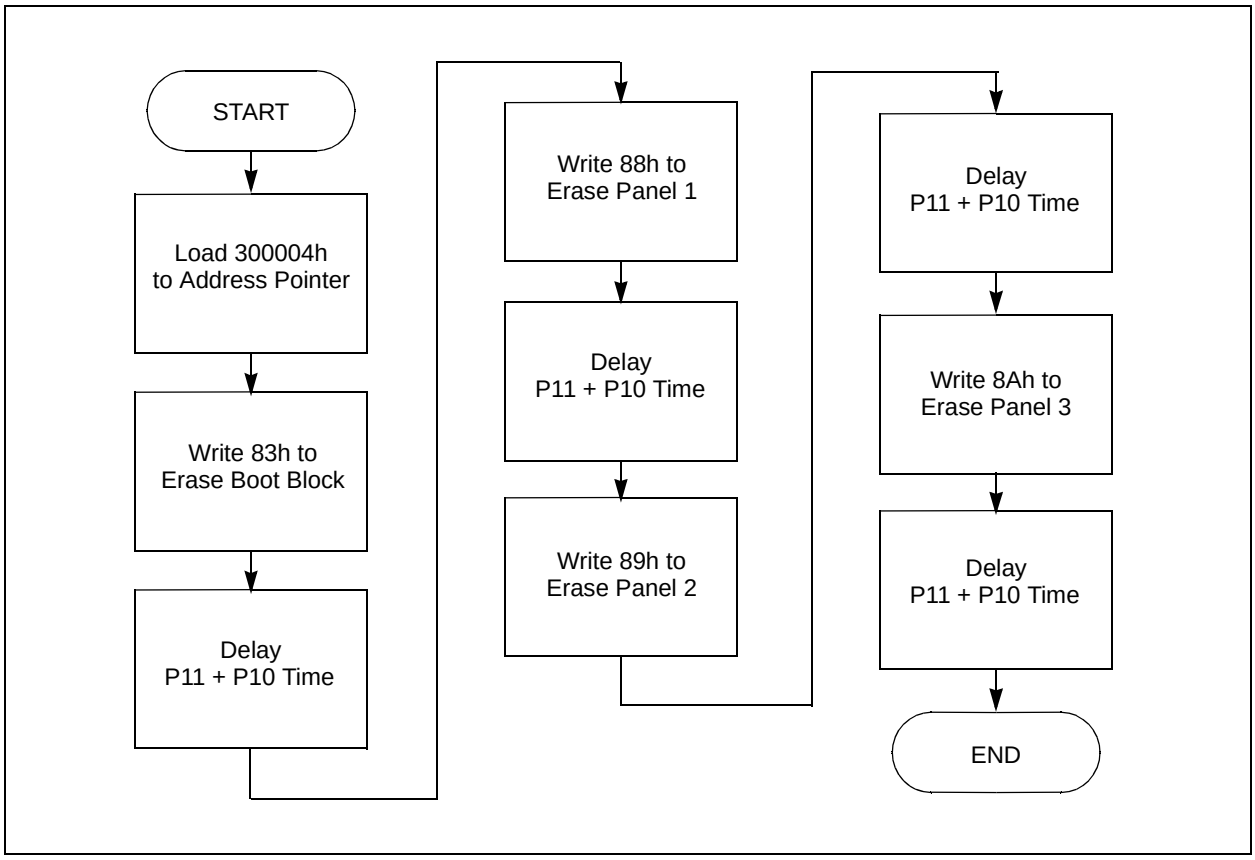
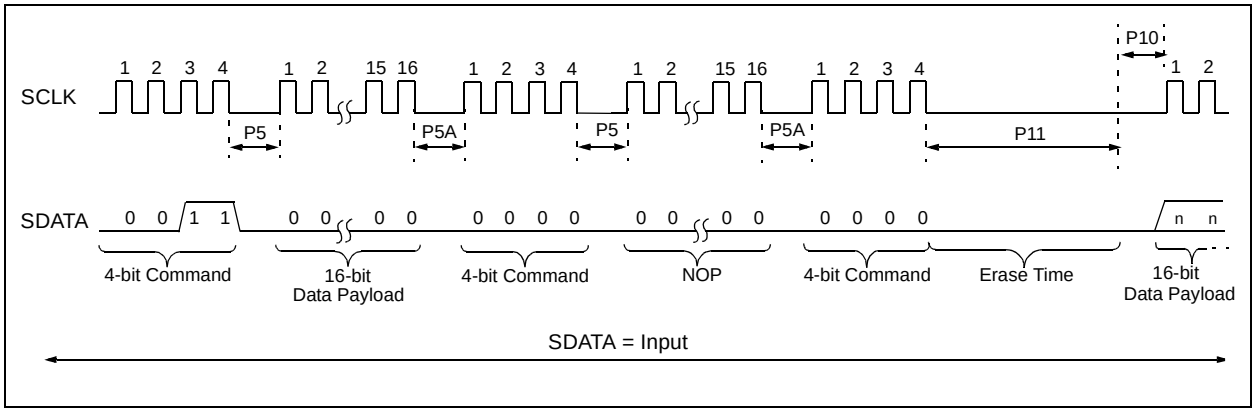


FIGURE 3-4: BULK ERASE TIMING



3.2.1 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 3.3.1.

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 3.4 and write zeroes to the array.

3.3 Code Memory Programming

Programming code memory is accomplished by first loading data into the appropriate write buffers and then initiating a programming sequence. Each panel in the code memory space (see Figure 2-2) has an 8-byte deep write buffer that must be loaded prior to initiating a write sequence. The actual memory write sequence takes the contents of these buffers and programs the associated EEPROM code memory.

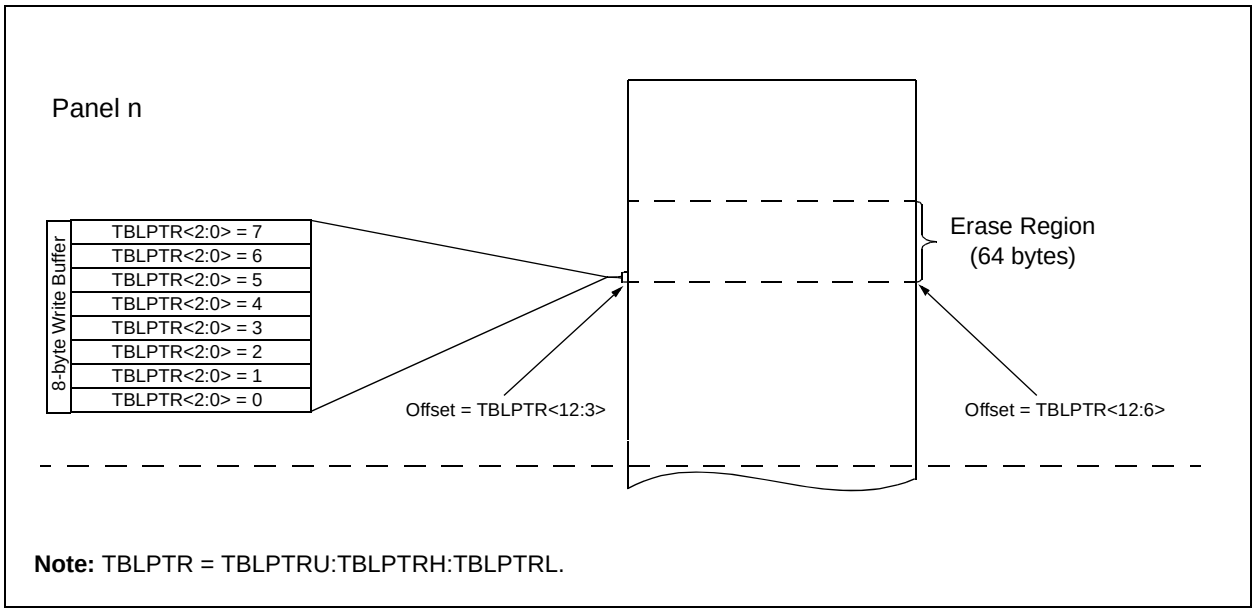
The programming duration is externally timed and is controlled by SCLK. After a “Start Programming” command is issued (4-bit command, ‘1111’), a NOP is issued where the 4th SCLK is held high for the duration of the programming time, P9.

After SCLK is brought low, the programming sequence is terminated. SCLK 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 PIC18FXX39 device is shown in Table 3-4. The flowchart shown in Figure 3-7 depicts the logic necessary to completely write a PIC18FXX39 device.

Note:	The TBLPTR register must contain the same offset value when initiating the programming sequence as it did when the write buffers were loaded.

FIGURE 3-5: ERASE AND WRITE BOUNDARIES

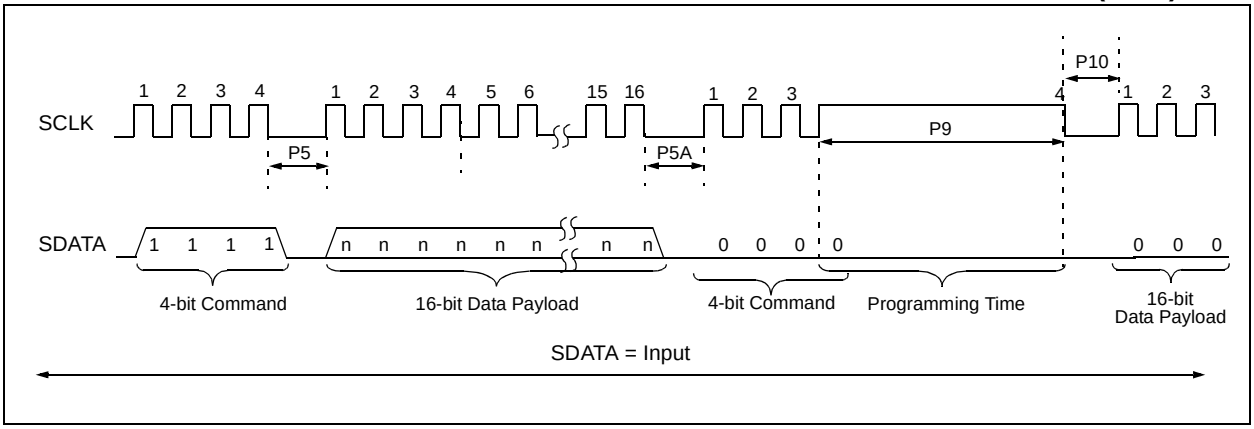


Note: TBLPTR = TBLPTRU:TBLPTRH:TBLPTRL.

TABLE 3-4: WRITE CODE MEMORY CODE SEQUENCE

4-bit Command	Data Payload	Core Instruction
Step 1: Configure device for multi-panel writes.		
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 06	MOVLW 06h
0000	6E F6	MOVWF TBLPTRL
1100	00 40	Write 00h to 3C0006h to enable single panel writes.
Step 2: Direct access to code memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
Step 3: Load write buffer.		
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	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1101	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1101	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1111	<LSB><MSB>	Write 2 bytes and start programming
0000	00 00	NOP – Hold SCLK high for Time P9
Step 4: Delay of P10.		
To continue writing data, repeat steps 2 through 4, where the Address Pointer is incremented by 8 at each iteration of the loop.		

FIGURE 3-6: TABLE WRITE AND START PROGRAMMING INSTRUCTION TIMING (1111)



3.3.1 MODIFYING CODE MEMORY

All of the programming examples up to this point have assumed that the device is blank prior to programming. In fact, if the device is not blank, the direction has been to completely erase the device via a Bulk Erase operation (see Section 3.2).

It may be the case, however, that the user wishes to modify only a section of an already programmed device. In such a situation, erasing the entire device is not a realistic option.

The minimum amount of data that can be written to the device is 8 bytes. This is accomplished by placing the device in Single Panel Write mode, loading the 8-byte write buffer for the panel, and then initiating a write sequence, as shown in Table 3-4. In this case, however, it is assumed that the address space to be written already has data in it (i.e., it is not blank).

The minimum amount of code memory that may be erased at a given time is 64 bytes. Again, the device must be placed in Single Panel Write mode. The EECON1 register must then be used to erase the 64-byte target space prior to writing the data.

When using the EECON1 register to act on code memory, 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 sequence is initiated by the setting the WR bit (EECON1<1> = 1). It is strongly recommended that the WREN bit be set only when absolutely necessary.

To help prevent inadvertent writes when using the EECON1 register, EECON2 is used to “enable” the WR bit. This register must be sequentially loaded with 55h and then, AAh, immediately prior to asserting the WR bit in order for the write to occur.

The erase will begin on the falling edge of the 4th SCLK after the WR bit is set.

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

FIGURE 3-7: PROGRAM CODE MEMORY FLOW

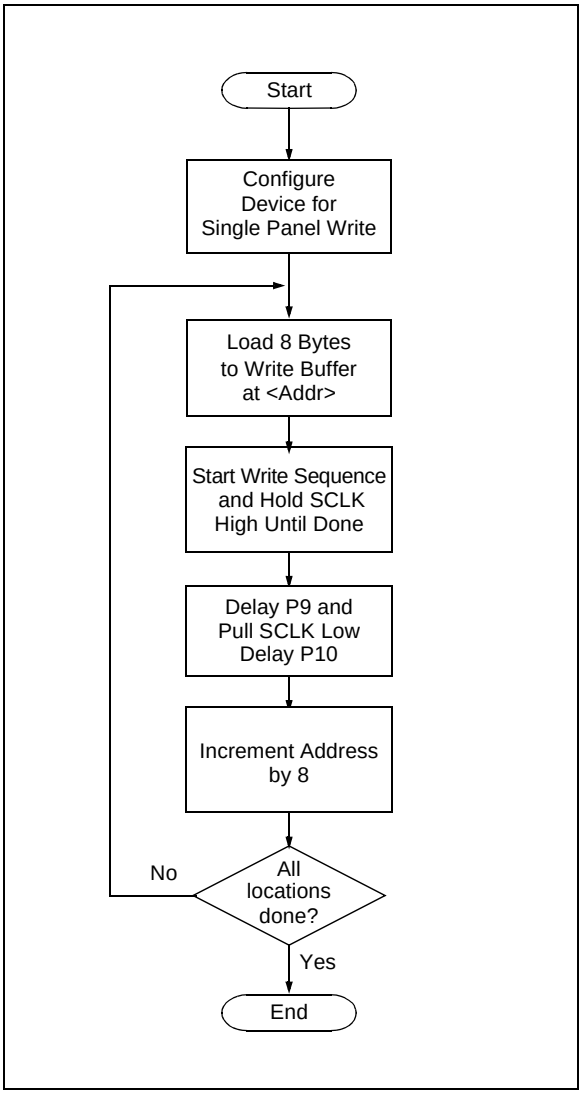


TABLE 3-5: MODIFYING CODE MEMORY

4-bit Command	Data Payload	Core Instruction
Step 1: Configure device for single panel writes.		
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 06	MOVLW 06h
0000	6E F6	MOVWF TBLPTRL
1100	00 00	Write 00h to 3C0006h to enable single panel writes.
Step 2: Direct access to code memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
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: Perform Data EEPROM unlock sequence.		
0000	0E 55	MOVLW 0X55
0000	6E A7	MOVWF EECON2
0000	0E AA	MOVLW 0XAA
0000	6E A7	MOVWF EECON2
Step 6: Initiate erase.		
0000	82 A6	BSF EECON1, WR
0000	00 00	NOP
Step 7: Wait for P11+P10 and then disable writes.		
0000	94 A6	BCF EECON1, WREN
Step 8: Load write buffer for panel. Correct panel will be selected based on the Table Pointer.		
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	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1101	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1101	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1111	<LSB><MSB>	Write 2 bytes and start programming
0000	00 00	NOP - hold SCLK high for time P9
To continue writing data, repeat step 8, where the Address Pointer is incremented by 8 at each iteration of the loop.		

3.4 Data EEPROM Programming

Data EEPROM is accessed one byte at a time via an Address Pointer, EEADR, and a data latch, EEDATA. Data EEPROM is written by loading EEADR with the desired memory location, EEDATA with the data to be written, and initiating a memory write by appropriately configuring the EECON1 and EECON2 registers. 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, the EEPGD bit must be cleared (EECON1<7> = 0) 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, and this must be done prior to initiating a write sequence. The write sequence is initiated by the setting the WR bit (EECON1<1> = 1). It is strongly recommended that the WREN bit be set only when absolutely necessary.

To help prevent inadvertent writes when using the EECON1 register, EECON2 is used to “enable” the WR bit. This register must be sequentially loaded with 55h and then, AAh, immediately prior to asserting the WR bit in order for the write to occur.

The write will begin on the falling edge of the 4th SCLK after the WR bit is set.

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

FIGURE 3-8: PROGRAM DATA FLOW

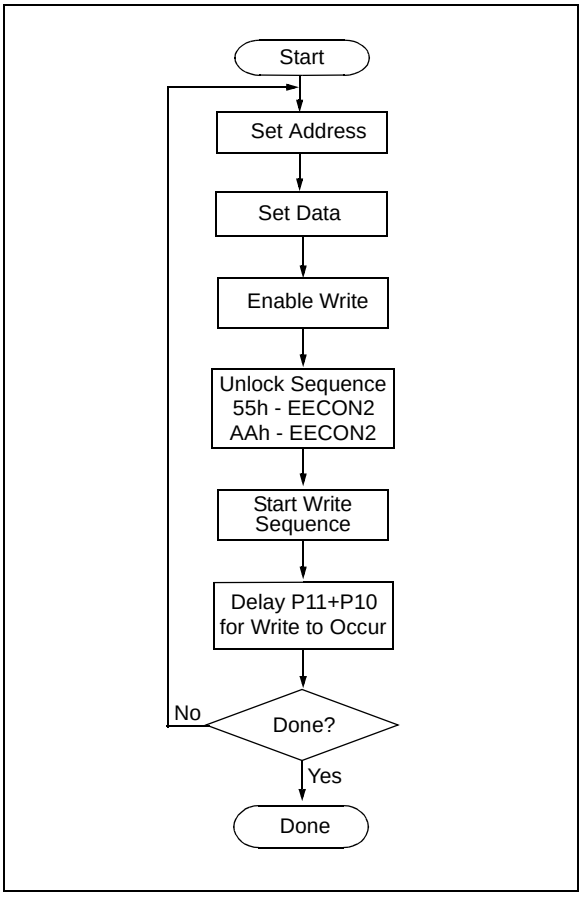


FIGURE 3-9: DATA EEPROM WRITE TIMING

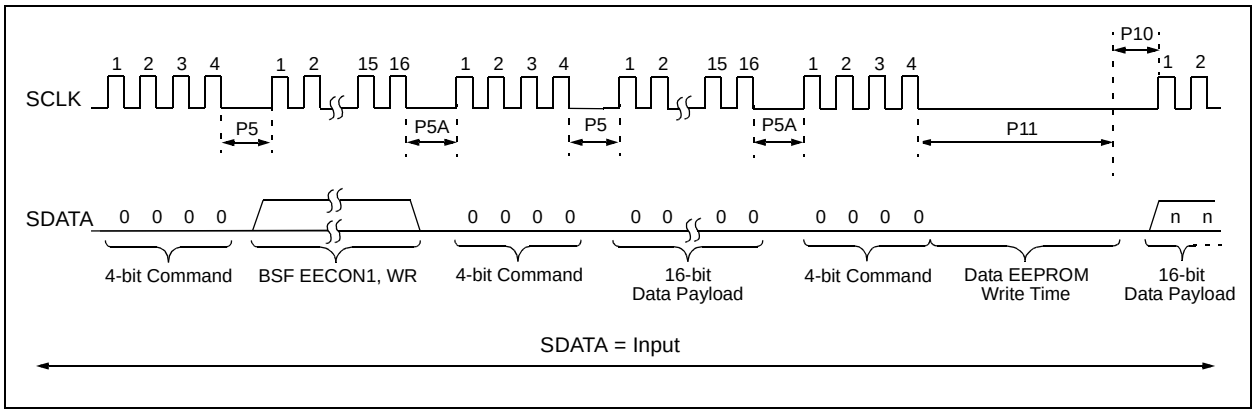


TABLE 3-6: PROGRAMMING DATA MEMORY

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	0E <Addr> 6E A9	MOVLW <Addr> MOVWF EEADR
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: Perform Data EEPROM unlock sequence.		
0000 0000 0000 0000	0E 55 6E A7 0E AA 6E A7	MOVLW 0X55 MOVWF EECON2 MOVLW 0XAA MOVWF EECON2
Step 6: Initiate write.		
0000 0000 0000	82 A6 00 00 00 00	BSF EECON1, WR NOP Hold SDATA low until write completes
Step 7: Wait for P11 and then disable writes.		
0000	94 A6	BCF EECON1, WREN
Repeat steps 2 through 7 to write more data.		

3.5 ID Location Programming

The ID locations are programmed much like the code memory, except that multi-panel writes must be disabled. The single panel that will be written will automatically be enabled, based on the value of the Table Pointer. The ID registers are mapped in addresses 200000h through 200007h. These locations read out normally, even after code protection.

Note:	For single panel programming, the user must still fill the 8-byte data buffer for the panel.

Table 3-7 demonstrates the code sequence required to write the ID locations.

TABLE 3-7: WRITE ID SEQUENCE

4-bit Command	Data Payload	Core Instruction
Step 1: Configure device for single panel writes.		
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 06	MOVLW 06h
0000	6E F6	MOVWF TBLPTRL
1100	00 00	Write 00h to 3C0006h to enable single panel writes.
Step 2: Direct access to code memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
Step 3: Load write buffer. Panel will be automatically determined by address.		
0000	0E 20	MOVLW 20h
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 00	MOVLW 00h
0000	6E F6	MOVWF TBLPTRL
1101	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1101	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1101	<LSB><MSB>	Write 2 bytes and post-increment address by 2
1111	<LSB><MSB>	Write 2 bytes and start programming
0000	00 00	NOP - hold SCLK high for time P9

3.6 Boot Block Programming

The Boot Block segment is programmed in exactly the same manner as the ID locations (see Section 3.5). Multi-panel writes must be disabled so that only addresses in the range 0000h to 01FFh will be written.

The code sequence detailed in Table 3-7 should be used, except that the address data used in “Step 3” will be in the range 000000h to 0001FFh.

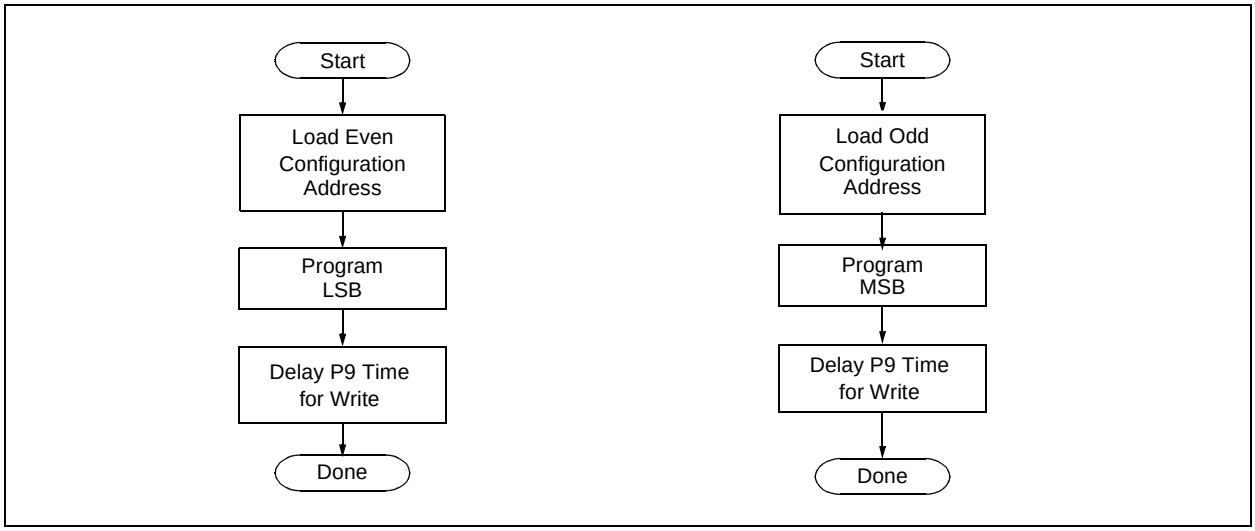
3.7 Configuration Bits Programming

Unlike code memory, 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 Figure 3-8.

TABLE 3-8: SET ADDRESS POINTER TO CONFIGURATION LOCATION

4-bit Command	Data Payload	Core Instruction
Step 1: Configure device for single panel writes.		
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 06	MOVLW 06h
0000	6E F6	MOVWF TBLPTRL
1100	00 00	Write 00h to 3C0006h to enable single panel writes.
Step 2: Direct access to configuration memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	8C A6	BSF EECON1, CFGS
Step 3: Set Table Pointer for configuration word to be written. Write even/odd addresses.		
0000	0E 30	MOVLW 30h
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPRTH
0000	0E 00	MOVLW 00h
0000	6E F6	MOVWF TBLPTRL
1111	<LSB><MSB ignored>	Load 2 bytes and start programming
0000	00 00	NOP - hold SCLK high for time P9
0000	2A F6	INCF TBLPTRL
1111	<LSB ignored><MSB>	Load 2 bytes and start programming
0000	00 00	NOP - hold SCLK high for time P9

FIGURE 3-10: CONFIGURATION PROGRAMMING FLOW



4.0 READING THE DEVICE

4.1 Read Data EEPROM Memory

Data EEPROM is accessed one byte at a time via an Address Pointer, EEADR, and a data latch, EEDATA. Data EEPROM is read by loading 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 SDATA 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 SCLK of the operand to allow SDATA to transition from an input to an output. During this time, SCLK must be held low (see Figure 4-2).

The command sequence to read a single byte of data is shown in Table 4-1.

FIGURE 4-1: READ DATA EEPROM FLOW

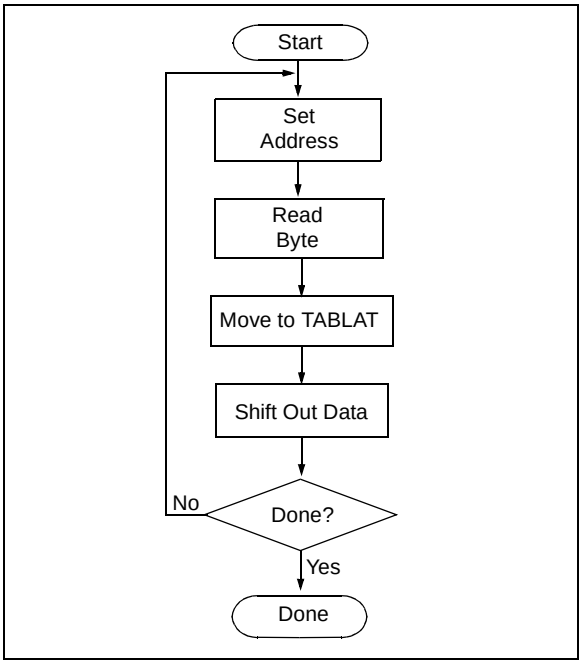
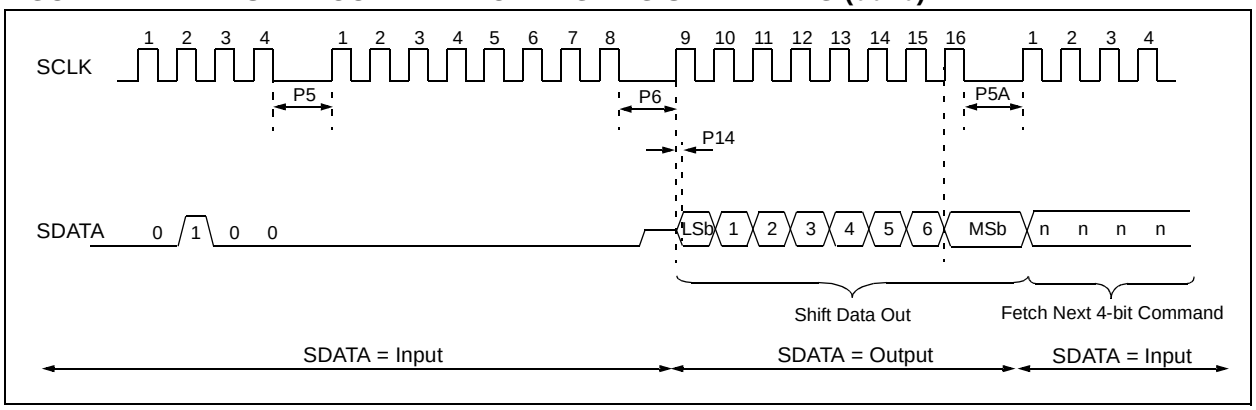


TABLE 4-1: READ DATA EEPROM MEMORY

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	0E <Addr> 6E A9	MOVLW <Addr> MOVWF EEADR
Step 3: Initiate a memory read.		
0000	80 A6	BSF EECON1, RD
Step 4: Load data into the serial data holding register.		
0000 0000 0010	50 A8 6E F5 <LSB><MSB>	MOVF EEDATA, W, 0 MOVWF TABLAT Shift Out Data(1)

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

FIGURE 4-2: SHIFT OUT DATA HOLDING REGISTER TIMING (0010)



4.2 Read Code Memory, ID Locations, and Configuration Bits

Code memory 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 loaded into the Table Latch and then serially output on SDATA.

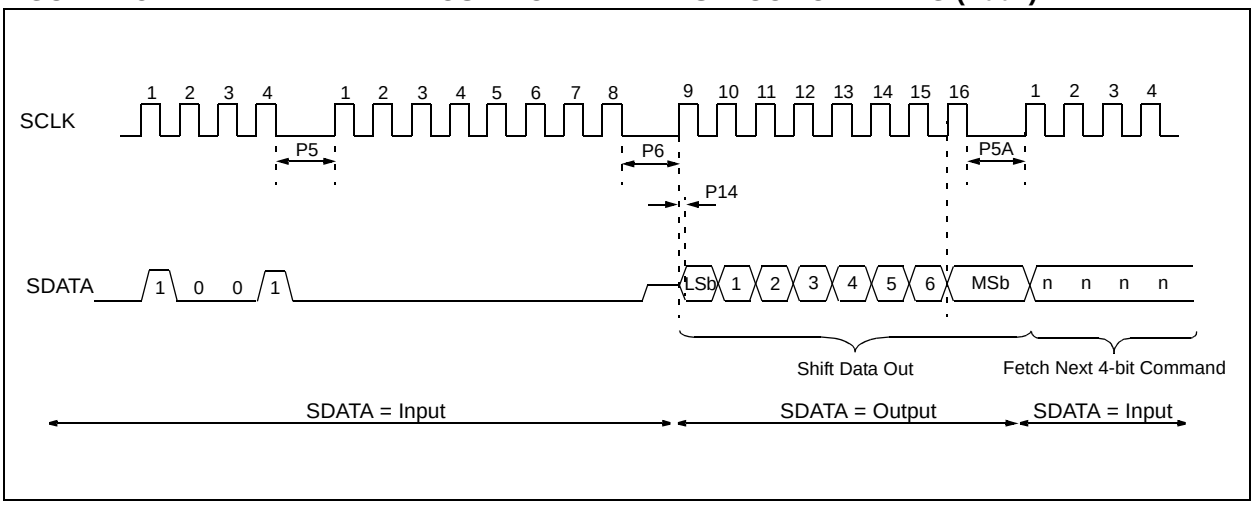
The 4-bit command is shifted in LSb first. The Table Read is executed during the next 8 clocks, then shifted out on SDATA during the last 8 clocks, LSb to MSb. A delay of P6 must be introduced after the falling edge of the 8th SCLK of the operand to allow SDATA to transition from an input to an output. During this time, SCLK must be held low (see Table 4-2). This operation also increments the Table Pointer by one, pointing to the next byte in code memory 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 4-2: READ CODE MEMORY SEQUENCE

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

FIGURE 4-3: TABLE READ POST-INCREMENT INSTRUCTION TIMING (1001)

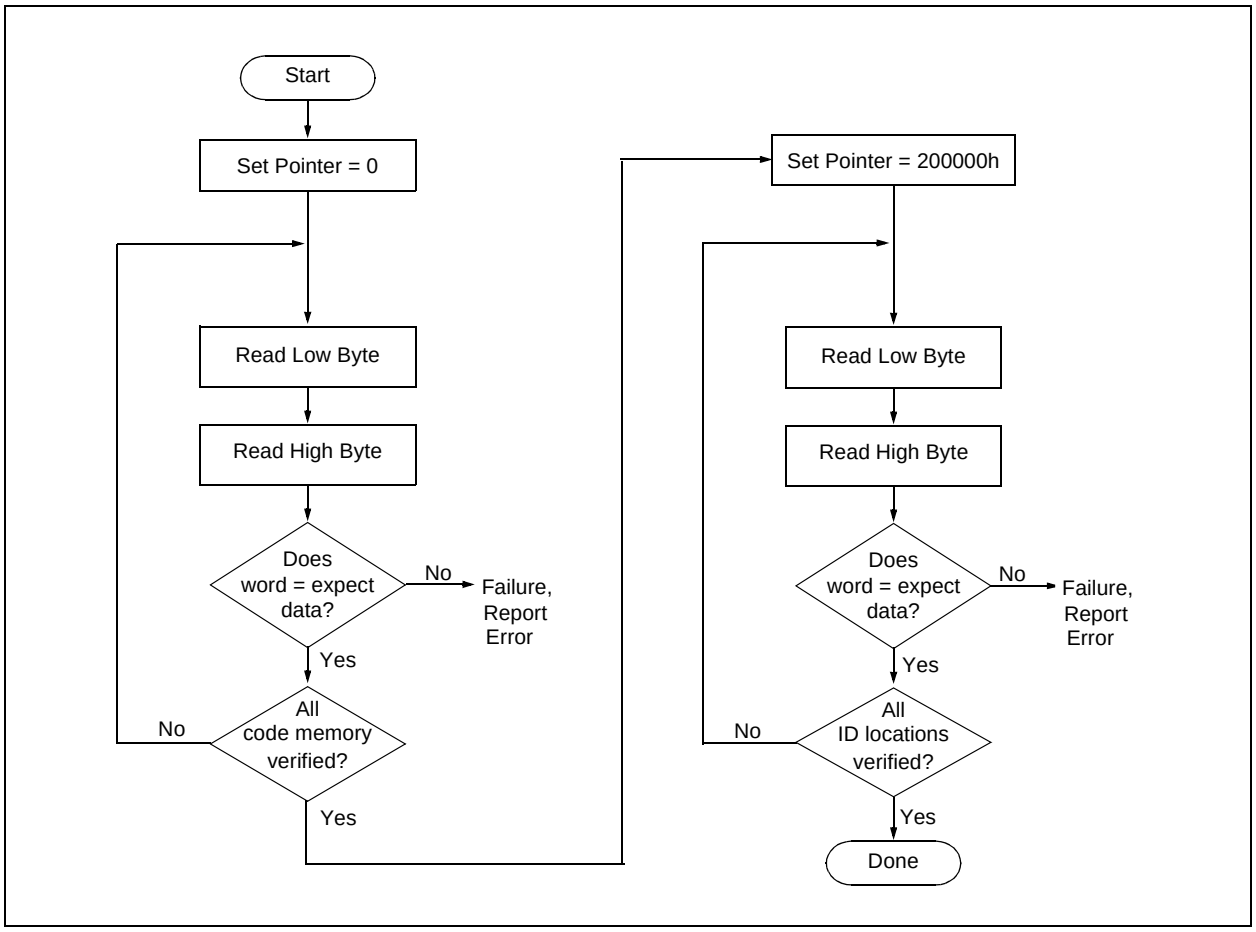


4.3 Verify Code Memory and ID Locations

The verify step involves reading back the code memory space and comparing 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 4.2 for implementation details of reading code memory.

The Table Pointer must be manually set to 200000h (base address of the ID locations) once the code memory has been verified. The post-increment feature of the Table Read 4-bit command may not be used to increment the Table Pointer beyond 1FFFFFh.

FIGURE 4-4: VERIFY CODE MEMORY FLOW



4.4 Verify Configuration Bits

A configuration address may be read and output on SDATA 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 4.2 for implementation details of reading configuration data.

4.5 Verify Data EEPROM

A data EEPROM address may be read via a sequence of core instructions (4-bit command, '0000') and then output on SDATA via the 4-bit command, '0010' (shift out data holding register). The result may then be immediately compared to the appropriate data in the programmer's memory for verification. Refer to Section 4.1 for implementation details of reading data EEPROM.

5.0 CONFIGURATION WORD

The PIC18FXX39 has several configuration words. These bits can be set or cleared to select various device configurations. All other memory areas should be programmed and verified prior to setting configuration words. These bits may be read out normally, even after read or code protected.

5.1 ID Locations

A user may store identification information (ID) in eight ID locations mapped in 200000h:200007h. It is recommended that the Most Significant nibble of each ID be 0Fh. In doing so, if the user code inadvertently tries to execute from the ID space, the ID data will execute as a NOP.

5.2 Device ID Word

The device ID word for the PIC18FXX39 is located at 3FFFFEh:3FFFFFh. These bits may be used by the programmer to identify what device type is being programmed and read out normally, even after code or read protected.

5.3 Low Voltage Programming (LVP) Bit

The LVP bit in configuration register, CONFIG4L, enables low voltage ICSP programming. The LVP bit defaults to a '1' from the factory.

If Low Voltage Programming mode is not used, the LVP bit can be programmed to a '0' and RB5/PGM becomes a digital I/O pin. However, the LVP bit may only be programmed by entering the High Voltage ICSP mode, where $\overline{MCLR}/V_{PP}$ 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 normal ICSP mode is always available, regardless of the state of the LVP bit, by applying V_{IHH} to the $\overline{MCLR}/V_{PP}$ pin.

2: While in Low Voltage ICSP mode, the RB5 pin can no longer be used as a general purpose I/O.

TABLE 5-1: DEVICE ID VALUE

Device	Device ID Value	
	DEVID2	DEVID1
PIC18F2439	04h	100x xxxx
PIC18F2539	04h	000x xxxx
PIC18F4439	04h	101x xxxx
PIC18F4539	04h	001x xxxx

TABLE 5-2: PIC18FXX39 CONFIGURATION BITS AND DEVICE IDs

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Erased or "Blank" Value
300000h	CONFIG1L	—	—	—	—	—	—	—	—	0000 0000
300001h	CONFIG1H	—	—	—(1)	—	—	FOSC2	FOSC1	FOSC0	0010 0111
300002h	CONFIG2L	—	—	—	—	BORV1	BORV2	BOREN	$\overline{\text{PWRTEN}}$	0000 1111
300003h	CONFIG2H	—	—	—	—	WDTPS2	WDTPS1	WDTPS0	WDTEN	0000 1111
300004h	CONFIG3L	—	—	—	—	—	—	—	—	0000 0000
300005h	CONFIG3H	—	—	—	—	—	—	—	—(1)	0000 0001
300006h	CONFIG4L	$\overline{\text{BKBUG}}$	—	—	—	—	LVP	—	STVREN	1000 0101
300007h	CONFIG4H	—	—	—	—	—	—	—	—	0000 0000
300008h	CONFIG5L	—	—	—	—	—(1)	CP2	CP1	CP0	0000 1111
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—	1100 0000
30000Ah	CONFIG6L	—	—	—	—	—(1)	WRT2	WRT1	WRT0	0000 1111
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	—	—	—	—	—	1110 0000
30000Ch	CONFIG7L	—	—	—	—	—(1)	EBTR2	EBTR1	EBTR0	0000 1111
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—	0100 0000
3FFFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	Table 5-1
3FFFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	Table 5-1

Note 1: Unimplemented, but reserved; maintain this bit set.
2: Shaded cells are unimplemented, read as '0'.

TABLE 5-3: PIC18FXX39 BIT DESCRIPTION

Bit Name	Configuration Words	Description
FOSC2:FOSC0	CONFIG1H	Oscillator Selection bits 111 = Reserved 110 = HS oscillator w/ PLL enabled 101 = EC oscillator w/ OSC2 configured as RA6 100 = EC oscillator w/ OSC2 configured as “divide by 4 clock output” 011 = Reserved 010 = HS oscillator 001 = Reserved 000 = Reserved
BORV1:BORV0	CONFIG2L	Brown-out Reset Voltage bits 11 = VBOR set to 2.0V 10 = VBOR set to 2.7V 01 = VBOR set to 4.2V 00 = VBOR set to 4.5V
BOREN	CONFIG2L	Brown-out Reset Enable bit 1 = Brown-out Reset enabled 0 = Brown-out Reset disabled
$\overline{\text{PWRTEN}}$	CONFIG2L	Power-up Timer Enable bit 1 = PWRT disabled 0 = PWRT enabled
WDTPS2:WDTPS0	CONFIG2H	Watchdog Timer Postscaler Select bits 111 = 1:128 110 = 1:64 101 = 1:32 100 = 1:16 011 = 1:8 010 = 1:4 001 = 1:2 000 = 1:1
WDTEN	CONFIG2H	Watchdog Timer Enable bit 1 = WDT enabled 0 = WDT disabled (control is placed on SWDTEN bit)
$\overline{\text{BKBUG}}$	CONFIG4L	Background Debugger Enable bit 1 = Background debugger disabled 0 = Background debugger enabled
LVP	CONFIG4L	Low Voltage Programming Enable bit 1 = Low voltage programming enabled 0 = Low voltage programming disabled
STVREN	CONFIG4L	Stack Overflow/Underflow Reset Enable bit 1 = Stack overflow/underflow will cause RESET 0 = Stack overflow/underflow will not cause RESET
CP0	CONFIG5L	Code Protection bits (code memory area 0200h - 1FFFh) 1 = Code memory not code protected 0 = Code memory code protected
CP1	CONFIG5L	Code Protection bits (code memory area 2000h - 3FFFh) 1 = Code memory not code protected 0 = Code memory code protected
CP2	CONFIG5L	Code Protection bits (code memory area 4000h - 5FFFh) 1 = Code memory not code protected 0 = Code memory code protected

TABLE 5-3: PIC18FXX39 BIT DESCRIPTION (CONTINUED)

Bit Name	Configuration Words	Description
CPD	CONFIG5H	Code Protection bits (data EEPROM) 1 = Data EEPROM not code protected 0 = Data EEPROM code protected
CPB	CONFIG5H	Code Protection bits (boot block, memory area 0000h - 01FFh) 1 = Boot block not code protected 0 = Boot block code protected
WRT0	CONFIG6L	Table Write Protection bit (code memory area 0200h - 1FFFh) 1 = Code memory not write protected 0 = Code memory write protected
WRT1	CONFIG6L	Table Write Protection bit (code memory area 2000h - 3FFFh) 1 = Code memory not write protected 0 = Code memory write protected
WRT2	CONFIG6L	Table Write Protection bit (code memory area 4000h - 5FFFh) 1 = Code memory not write protected 0 = Code memory write protected
WRTD	CONFIG6H	Table Write Protection bit (data EEPROM) 1 = Data EEPROM not write protected 0 = Data EEPROM write protected
WRTB	CONFIG6H	Table Write Protection bit (boot block, memory area 0000h - 01FFh) 1 = Boot block not write protected 0 = Boot block write protected
WRTC	CONFIG6H	Table Write Protection bit (Configuration registers) 1 = Configuration registers not write protected 0 = Configuration registers write protected
EBTR0	CONFIG7L	Table Read Protection bit (code memory area 0200h - 01FFFh) 1 = Code memory not protected from table reads executed in other blocks 0 = Code memory protected from table reads executed in other blocks
EBTR1	CONFIG7L	Table Read Protection bit (code memory area 2000h - 3FFFh) 1 = Code memory not protected from table reads executed in other blocks 0 = Code memory protected from table reads executed in other blocks
EBTR2	CONFIG7L	Table Read Protection bit (code memory area 4000h - 5FFFh) 1 = Code memory not protected from table reads executed in other blocks 0 = Code memory protected from table reads executed in other blocks
EBTRB	CONFIG7H	Table Read Protection bit (boot block, memory area 0000h - 01FFh) 1 = Boot block not protected from table reads executed in other blocks 0 = Boot block protected from table reads executed in other blocks
DEV10:DEV3	DEVID2	Device ID bits are used with the DEV2:DEV0 bits in the DEVID1 register to identify part number.
DEV2:DEV0	DEVID1	Device ID bits are used with the DEV10:DEV3 bits in the DEVID2 register to identify part number.
REV4:REV0	DEVID1	These bits are used to indicate the revision of the device.

5.4 Embedding Configuration Word Information in the HEX File

To allow portability of code, a PIC18FXX39 programmer is required to read the configuration word locations from the HEX file. If configuration word information is not present in the HEX file, then a simple warning message should be issued. Similarly, while saving a HEX file, all configuration word information must be included. An option to not include the configuration word information may be provided. When embedding configuration word information in the HEX file, it should start at address 300000h.

Microchip Technology Inc. feels strongly that this feature is important for the benefit of the end customer.

5.5 Checksum Computation

The checksum is calculated by summing the following:

- The contents of all code memory locations
- The configuration word, appropriately masked
- ID locations

The Least Significant 16 bits of this sum are the checksum.

Table 5-4 describes how to calculate the checksum for each device.

Note 1: The checksum calculation differs depending on the code protect setting. Since the code memory locations read out differently depending on the code protect setting, the table describes how to manipulate the actual code memory values to simulate the values that would be read from a protected device. When calculating a checksum by reading a device, the entire code memory can simply be read and summed. The configuration word and ID locations can always be read.

TABLE 5-4: CHECKSUM COMPUTATION

Device	Code Protect or Mode	Checksum Calculation Method	Blank Checksum	Seed at 0	Seed at 0 and Max
PIC18F2439	None	SUM(0000:01FF)+SUM(0200:1FFF)+SUM(2000:3FFF)+ SUM(4000:5FFF)+(CONFIG0 & 0000)+(CONFIG1 & 0027)+ (CONFIG2 & 000F)+(CONFIG3 & 000F)+(CONFIG4 & 0000)+ (CONFIG5 & 0000)+(CONFIG6 & 0085)+(CONFIG7 & 0000)+ (CONFIG8 & 0007)+(CONFIG9 & 00C0)+(CONFIG10 & 0007)+ (CONFIG11 & 00E0)+(CONFIG12 & 0007)+(CONFIG13 & 0040)	A2BF	A26A	A215
	Boot Block	SUM(0200:1FFF)+SUM(2000:3FFF)+SUM(4000:5FFF)+ (CONFIG0 & 0000)+(CONFIG1 & 0027)+(CONFIG2 & 000F)+ (CONFIG3 & 000F)+(CONFIG4 & 0000)+(CONFIG5 & 0000)+ (CONFIG6 & 0085)+(CONFIG7 & 0000)+(CONFIG8 & 0007)+ (CONFIG9 & 00C0)+(CONFIG10 & 0007)+(CONFIG11 & 00E0)+ (CONFIG12 & 0007)+(CONFIG13 & 0040)+SUM(IDs)	A4A5	A49B	A43C
	Boot/Panel 1/ Panel 2	SUM(4000:5FFF)+(CONFIG0 & 0000)+(CONFIG1 & 0027)+ (CONFIG2 & 000F)+(CONFIG3 & 000F)+(CONFIG4 & 0000)+ (CONFIG5 & 0000)+(CONFIG6 & 0085)+(CONFIG7 & 0000)+ (CONFIG8 & 0007)+(CONFIG9 & 00C0)+(CONFIG10 & 0007)+ (CONFIG11 & 00E0)+(CONFIG12 & 0007)+(CONFIG13 & 0040)+ SUM(IDs)	E2A2	E298	E239
	All	(CONFIG0 & 0000)+(CONFIG1 & 0027)+(CONFIG2 & 000F)+ (CONFIG3 & 000F)+(CONFIG4 & 0000)+(CONFIG5 & 0000)+ (CONFIG6 & 0085)+(CONFIG7 & 0000)+(CONFIG8 & 0007)+ (CONFIG9 & 00C0)+(CONFIG10 & 0007)+(CONFIG11 & 00E0)+ (CONFIG12 & 0007)+(CONFIG13 & 0040)+SUM(IDs)	029E	0294	028A
PIC18F2539	None	SUM(0000:01FF)+SUM(0200:1FFF)+SUM(2000:3FFF)+ SUM(4000:5FFF)+(CONFIG0 & 0000)+(CONFIG1 & 0027)+ (CONFIG2 & 000F)+(CONFIG3 & 000F)+(CONFIG4 & 0000)+ (CONFIG5 & 0000)+(CONFIG6 & 0085)+(CONFIG7 & 0000)+ (CONFIG8 & 0007)+(CONFIG9 & 00C0)+(CONFIG10 & 0007)+ (CONFIG11 & 00E0)+(CONFIG12 & 0007)+(CONFIG13 & 0040)	A2BF	A26A	A215
	Boot Block	SUM(0200:1FFF)+SUM(2000:3FFF)+SUM(4000:5FFF)+ (CONFIG0 & 0000)+(CONFIG1 & 0027)+(CONFIG2 & 000F)+ (CONFIG3 & 000F)+(CONFIG4 & 0000)+(CONFIG5 & 0000)+ (CONFIG6 & 0085)+(CONFIG7 & 0000)+(CONFIG8 & 0007)+ (CONFIG9 & 00C0)+(CONFIG10 & 0007)+(CONFIG11 & 00E0)+ (CONFIG12 & 0007)+(CONFIG13 & 0040)+SUM(IDs)	A4A5	A49B	A43C
	Boot/Panel 1/ Panel 2	SUM(4000:5FFF)+(CONFIG0 & 0000)+(CONFIG1 & 0027)+ (CONFIG2 & 000F)+(CONFIG3 & 000F)+(CONFIG4 & 0000)+ (CONFIG5 & 0000)+(CONFIG6 & 0085)+(CONFIG7 & 0000)+ (CONFIG8 & 0007)+(CONFIG9 & 00C0)+(CONFIG10 & 0007)+ (CONFIG11 & 00E0)+(CONFIG12 & 0007)+(CONFIG13 & 0040)+ SUM(IDs)	E2A2	E298	E239
	All	(CONFIG0 & 0000)+(CONFIG1 & 0027)+(CONFIG2 & 000F)+ (CONFIG3 & 000F)+(CONFIG4 & 0000)+(CONFIG5 & 0000)+ (CONFIG6 & 0085)+(CONFIG7 & 0000)+(CONFIG8 & 0007)+ (CONFIG9 & 00C0)+(CONFIG10 & 0007)+(CONFIG11 & 00E0)+ (CONFIG12 & 0007)+(CONFIG13 & 0040)+SUM(IDs)	029E	0294	028A

Legend:

Item		Description
CFGW	=	Configuration Word
SUM[a:b]	=	Sum of locations a to b inclusive
SUM_ID	=	Byte-wise sum of lower four bits of all customer ID locations
+	=	Addition
&	=	Bit-wise AND

TABLE 5-4: CHECKSUM COMPUTATION (CONTINUED)

Device	Code Protect or Mode	Checksum Calculation Method	Blank Checksum	Seed at 0	Seed at 0 and Max
PIC18F4439	None	SUM(0000:01FF)+SUM(0200:1FFF)+SUM(2000:3FFF)+ SUM(4000:5FFF)+(CONFIG0 & 0000)+(CONFIG1 & 0027)+ (CONFIG2 & 000F)+(CONFIG3 & 000F)+(CONFIG4 & 0000)+ (CONFIG5 & 0000)+(CONFIG6 & 0085)+(CONFIG7 & 0000)+ (CONFIG8 & 0007)+(CONFIG9 & 00C0)+(CONFIG10 & 0007)+ (CONFIG11 & 00E0)+(CONFIG12 & 0007)+(CONFIG13 & 0040)	A2BF	A26A	A215
	Boot Block	SUM(0200:1FFF)+SUM(2000:3FFF)+SUM(4000:5FFF)+ (CONFIG0 & 0000)+(CONFIG1 & 0027)+(CONFIG2 & 000F)+ (CONFIG3 & 000F)+(CONFIG4 & 0000)+(CONFIG5 & 0000)+ (CONFIG6 & 0085)+(CONFIG7 & 0000)+(CONFIG8 & 0007)+ (CONFIG9 & 00C0)+(CONFIG10 & 0007)+(CONFIG11 & 00E0)+ (CONFIG12 & 0007)+(CONFIG13 & 0040)+SUM(IDs)	A4A5	A49B	A43C
	Boot/Panel 1/ Panel 2	SUM(4000:5FFF)+(CONFIG0 & 0000)+(CONFIG1 & 0027)+ (CONFIG2 & 000F)+(CONFIG3 & 000F)+(CONFIG4 & 0000)+ (CONFIG5 & 0000)+(CONFIG6 & 0085)+(CONFIG7 & 0000)+ (CONFIG8 & 0007)+(CONFIG9 & 00C0)+(CONFIG10 & 0007)+ (CONFIG11 & 00E0)+(CONFIG12 & 0007)+(CONFIG13 & 0040)+ SUM(IDs)	E2A2	E298	E239
	All	(CONFIG0 & 0000)+(CONFIG1 & 0027)+(CONFIG2 & 000F)+ (CONFIG3 & 000F)+(CONFIG4 & 0000)+(CONFIG5 & 0000)+ (CONFIG6 & 0085)+(CONFIG7 & 0000)+(CONFIG8 & 0007)+ (CONFIG9 & 00C0)+(CONFIG10 & 0007)+(CONFIG11 & 00E0)+ (CONFIG12 & 0007)+(CONFIG13 & 0040)+SUM(IDs)	029E	0294	028A
PIC18F4539	None	SUM(0000:01FF)+SUM(0200:1FFF)+SUM(2000:3FFF)+ SUM(4000:5FFF)+(CONFIG0 & 0000)+(CONFIG1 & 0027)+ (CONFIG2 & 000F)+(CONFIG3 & 000F)+(CONFIG4 & 0000)+ (CONFIG5 & 0000)+(CONFIG6 & 0085)+(CONFIG7 & 0000)+ (CONFIG8 & 0007)+(CONFIG9 & 00C0)+(CONFIG10 & 0007)+ (CONFIG11 & 00E0)+(CONFIG12 & 0007)+(CONFIG13 & 0040)	A2BF	A26A	A215
	Boot Block	SUM(0200:1FFF)+SUM(2000:3FFF)+SUM(4000:5FFF)+ (CONFIG0 & 0000)+(CONFIG1 & 0027)+(CONFIG2 & 000F)+ (CONFIG3 & 000F)+(CONFIG4 & 0000)+(CONFIG5 & 0000)+ (CONFIG6 & 0085)+(CONFIG7 & 0000)+(CONFIG8 & 0007)+ (CONFIG9 & 00C0)+(CONFIG10 & 0007)+(CONFIG11 & 00E0)+ (CONFIG12 & 0007)+(CONFIG13 & 0040)+SUM(IDs)	A4A5	A49B	A43C
	Boot/Panel 1/ Panel 2	SUM(4000:5FFF)+(CONFIG0 & 0000)+(CONFIG1 & 0027)+ (CONFIG2 & 000F)+(CONFIG3 & 000F)+(CONFIG4 & 0000)+ (CONFIG5 & 0000)+(CONFIG6 & 0085)+(CONFIG7 & 0000)+ (CONFIG8 & 0007)+(CONFIG9 & 00C0)+(CONFIG10 & 0007)+ (CONFIG11 & 00E0)+(CONFIG12 & 0007)+(CONFIG13 & 0040)+ SUM(IDs)	E2A2	E298	E239
	All	(CONFIG0 & 0000)+(CONFIG1 & 0027)+(CONFIG2 & 000F)+ (CONFIG3 & 000F)+(CONFIG4 & 0000)+(CONFIG5 & 0000)+ (CONFIG6 & 0085)+(CONFIG7 & 0000)+(CONFIG8 & 0007)+ (CONFIG9 & 00C0)+(CONFIG10 & 0007)+(CONFIG11 & 00E0)+ (CONFIG12 & 0007)+(CONFIG13 & 0040)+SUM(IDs)	029E	0294	028A

Legend:

Item		Description
CFGW	=	Configuration Word
SUM[a:b]	=	Sum of locations a to b inclusive
SUM_ID	=	Byte-wise sum of lower four bits of all customer ID locations
+	=	Addition
&	=	Bit-wise AND

5.6 Embedding Data EEPROM Information In the HEX File

To allow portability of code, a PIC18FXX39 programmer is required to read the data EEPROM information from the HEX file. If data EEPROM information is not present, a simple warning message should be issued. Similarly, when saving a HEX file, all data EEPROM information must be included. An option to not include the data EEPROM information may be provided. When embedding data EEPROM information in the HEX file, it should start at address F00000h.

Microchip Technology Inc. believes that this feature is important for the benefit of the end customer.

6.0 AC/DC CHARACTERISTICS TIMING REQUIREMENTS FOR PROGRAM/VERIFY TEST MODE

Standard Operating Conditions
Operating Temperature: 25°C is recommended

Param No.	Sym	Characteristic	Min	Max	Units	Conditions
D110	VIHH	High Voltage Programming Voltage on MCLR/VPP	9.00	13.25	V	
D110A	VIHL	Low Voltage Programming Voltage on MCLR/VPP	2.00	5.50	V	
D111	VDD	Supply Voltage during Programming	2.00	5.50	V	Normal programming
			4.50	5.50	V	Bulk erase operations
D112	IPP	Programming Current on MCLR/VPP	—	300	µA	
D113	IDDP	Supply Current during Programming	—	1	mA	
D031	VIL	Input Low Voltage	VSS	0.2 VSS	V	
D041	VIH	Input High Voltage	0.8 VDD	VDD	V	
D080	VOL	Output Low Voltage	—	0.6	V	IOL = 8.5 mA
D090	VOH	Output High Voltage	VDD – 0.7	—	V	IOH = -3.0 mA
D012	CIO	Capacitive Loading on I/O pin (SDATA)	—	50	pF	To meet AC specifications
P2	Tsclk	Serial Clock (SCLK) Period	100	—	ns	VDD = 5.0V
			1	—	µs	VDD = 2.0V
P2A	TsclkL	Serial Clock (SCLK) Low Time	40	—	ns	VDD = 5.0V
			400	—	ns	VDD = 2.0V
P2B	TsclkH	Serial Clock (SCLK) High Time	40	—	ns	VDD = 5.0V
			400	—	ns	VDD = 2.0V
P3	Tset1	Input Data Setup Time to Serial Clock ↓	15	—	ns	
P4	Thld1	Input Data Hold Time from SCLK ↓	15	—	ns	
P5	Tdly1	Delay between 4-bit Command and Command Operand	20	—	ns	
P5A	Tdly1a	Delay between 4-bit Command Operand and next 4-bit Command	20	—	ns	
P6	Tdly2	Delay between Last SCLK ↓ of Command Byte to First SCLK ↑ of Read of Data Word	20	—	ns	
P9	Tdly5	SCLK High Time (minimum programming time)	1	—	ms	
P10	Tdly6	SCLK Low Time after Programming (high voltage discharge time)	5	—	µs	
P11	Tdly7	Delay to allow Self-Timed Data Write or Bulk Erase to occur	10	—	ms	
P12	Thld2	Input Data Hold Time from MCLR/VPP ↑	2	—	µs	
P13	Tset2	VDD ↑ Setup Time to MCLR/VPP ↑	100	—	ns	
P14	Tvalid	Data Out Valid from SCLK ↑	10	—	ns	
P15	Tset3	PGM ↑ Setup Time to MCLR/VPP ↑	2	—	µs	

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

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: 886-7-536-4818
Fax: 886-7-536-4803

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

01/05/10