PIC18FXX80/XX85

FLASH Microcontroller Programming Specification

1.0 DEVICE OVERVIEW

This document includes the programming specifications for the following devices:

- PIC18F6585
- PIC18F8585
- PIC18F6680
- PIC18F8680

2.0 PROGRAMMING OVERVIEW

PIC18FXX80/XX85 devices can be programmed using either the high voltage In-Circuit Serial Programming™ (ICSP™) method, or the low voltage ICSP method. Both of these programming 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. This programming specification applies to PIC18FXX80/XX85 devices in all package types.

2.1 Hardware Requirements

2.1.1 HIGH VOLTAGE ICSP PROGRAMMING

In High Voltage ICSP mode, these devices require 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.

2.1.2 LOW VOLTAGE ICSP PROGRAMMING

In Low Voltage ICSP mode, these devices 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 PIC18FXX80/XX85 family are shown in Figure 2-1, Figure 2-2 and Figure 2-3. The pin descriptions of these diagrams do not represent the complete functionality of the device types. Users should refer to the appropriate device data sheet for complete pin descriptions.

TABLE 2-1: PIN DESCRIPTIONS (DURING PROGRAMMING): PIC18FXX80/XX85

Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
MCLR/VPP/RA5	VPP	P	Programming Enable
VDD(2)	VDD	P	Power Supply
VSS(2)	VSS	P	Ground
AVDD(2)	AVDD	P	Analog Power Supply
AVSS(2)	AVSS	P	Analog 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

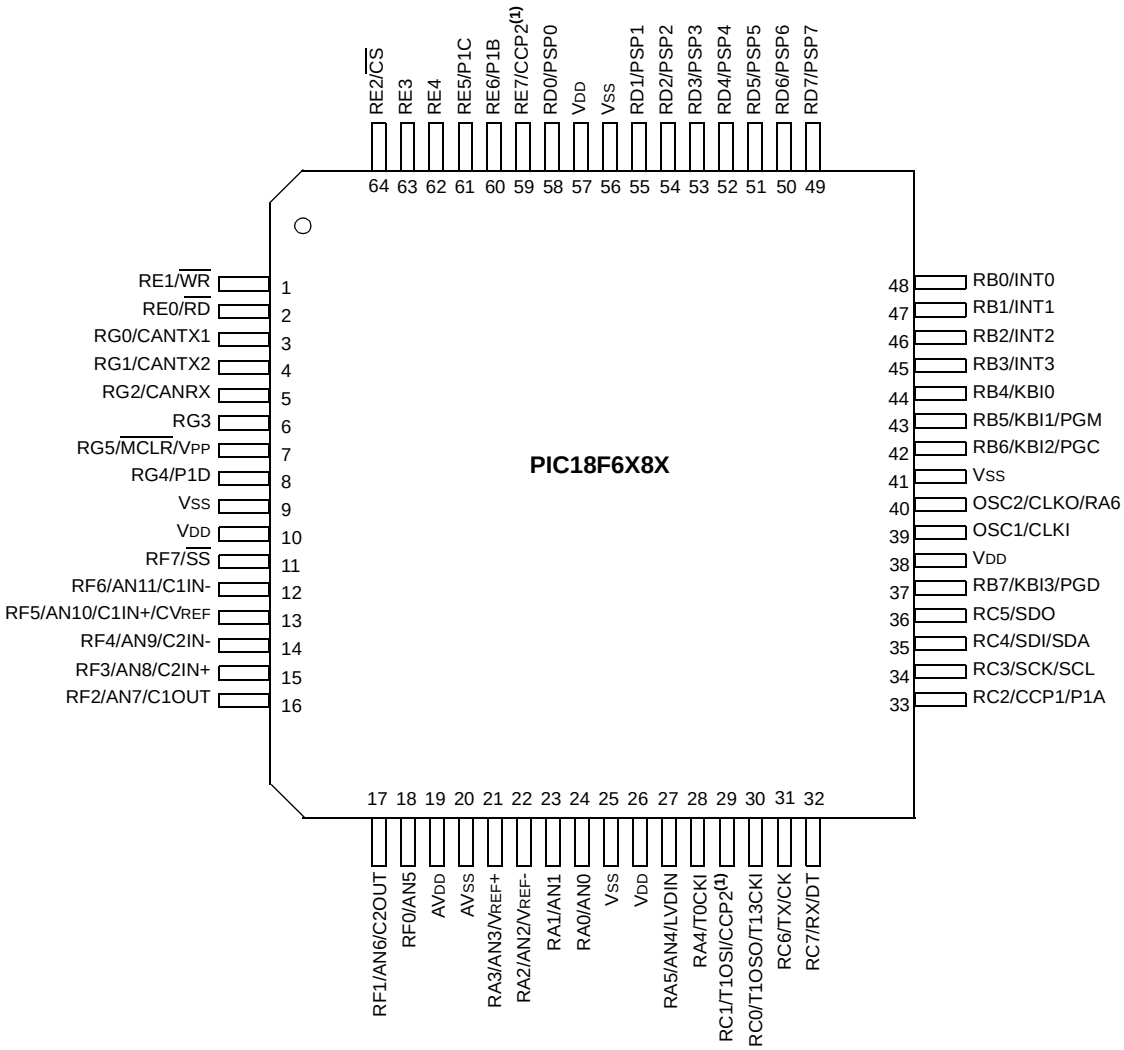
Legend: I = Input, O = Output, P = Power

Note 1: See Section 5.3 for more detail.

2: All power supplies and ground must be connected.

FIGURE 2-1: 64-PIN TQFP PACKAGE DIAGRAM FOR PIC18F6X8X

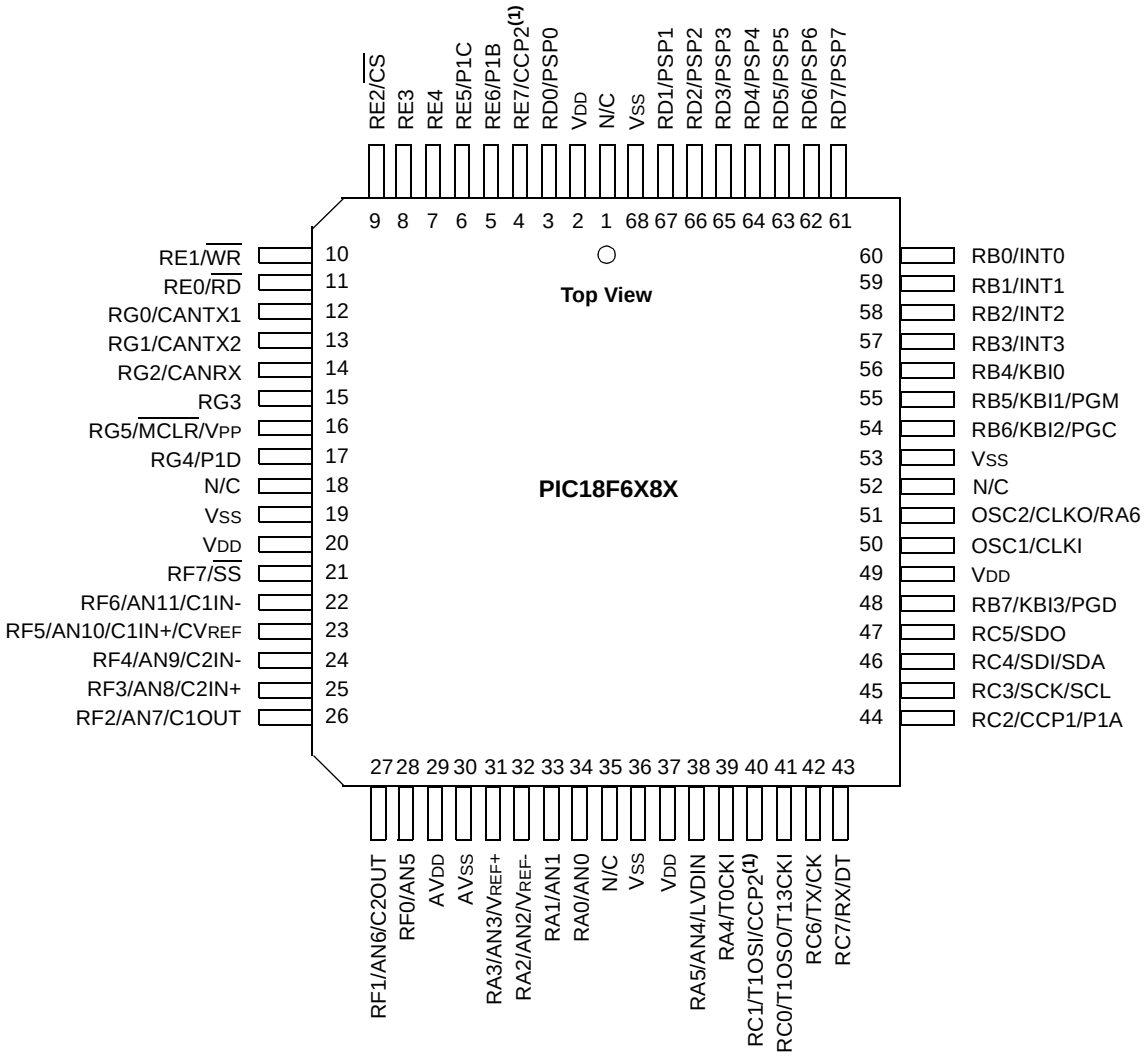
TQFP



Note 1: CCP2 pin placement depends on Processor mode settings.

FIGURE 2-2: 68-PIN PLCC PACKAGE DIAGRAM FOR PIC18F6X8X

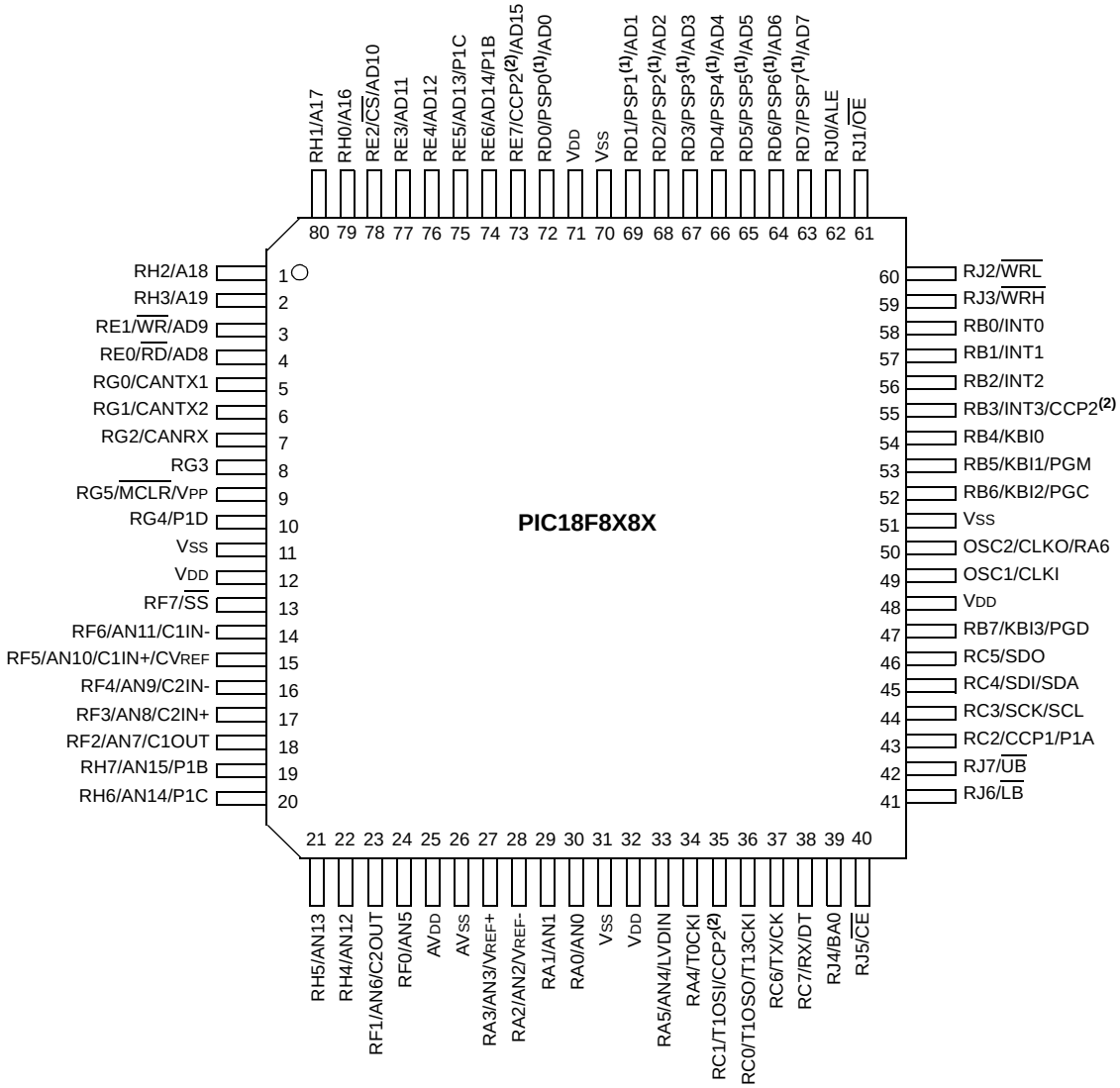
PLCC



Note 1: CCP2 pin placement depends on Processor mode settings.

FIGURE 2-3: 80-PIN TQFP PACKAGE DIAGRAM FOR PIC18F8X8X

TQFP



Note 1: PSP is available only in Microcontroller mode.

2: CCP2 pin placement depends on Processor mode settings.

2.3 Memory Map

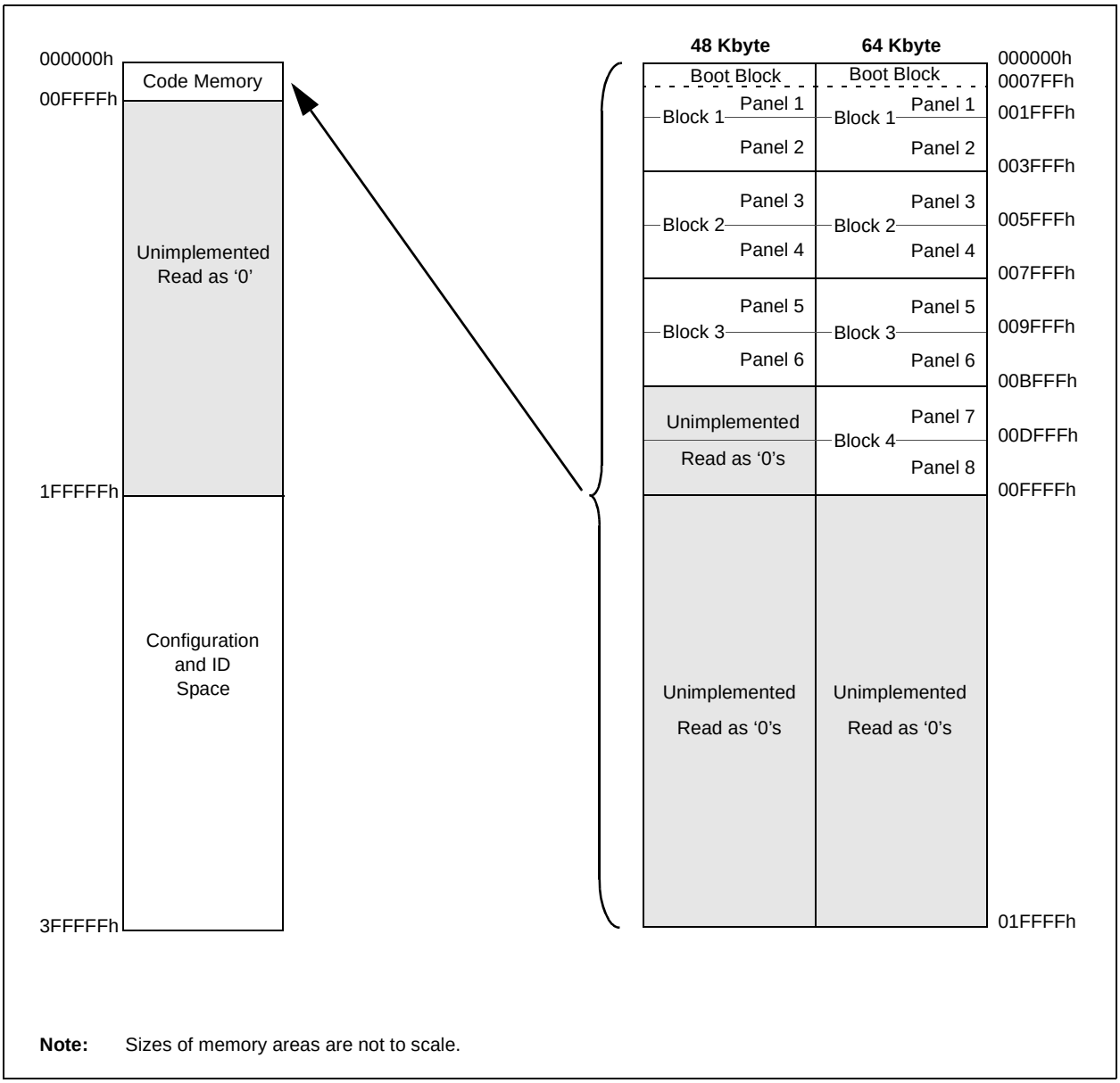
The code memory space extends from 0000h to 0FFFFh (64 Kbytes) in four 16-Kbyte blocks. However, addresses 0000h through 07FFh define a “Boot Block” region that is treated separately from Block 1. All of these blocks define code protection boundaries within the code memory space.

In contrast, code memory panels are defined in 8-Kbyte boundaries. Panels are discussed in greater detail in Section 3.2.

TABLE 2-2: IMPLEMENTATION OF CODE MEMORY

Device	Code Memory Size (Bytes)
PIC18F6585	0000h - 00BFFFh (48K)
PIC18F8585	
PIC18F6680	0000h -00FFFFh (64K)
PIC18F8680	

FIGURE 2-4: MEMORY MAP AND THE CODE MEMORY SPACE FOR PIC18FXX80/XX85 DEVICES



Note: Sizes of memory areas are not to scale.

In addition to the code memory 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 2-5.

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 5.0. 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 5.0. These device ID bits read out normally even after code protection.

2.3.1 MEMORY ADDRESS POINTER

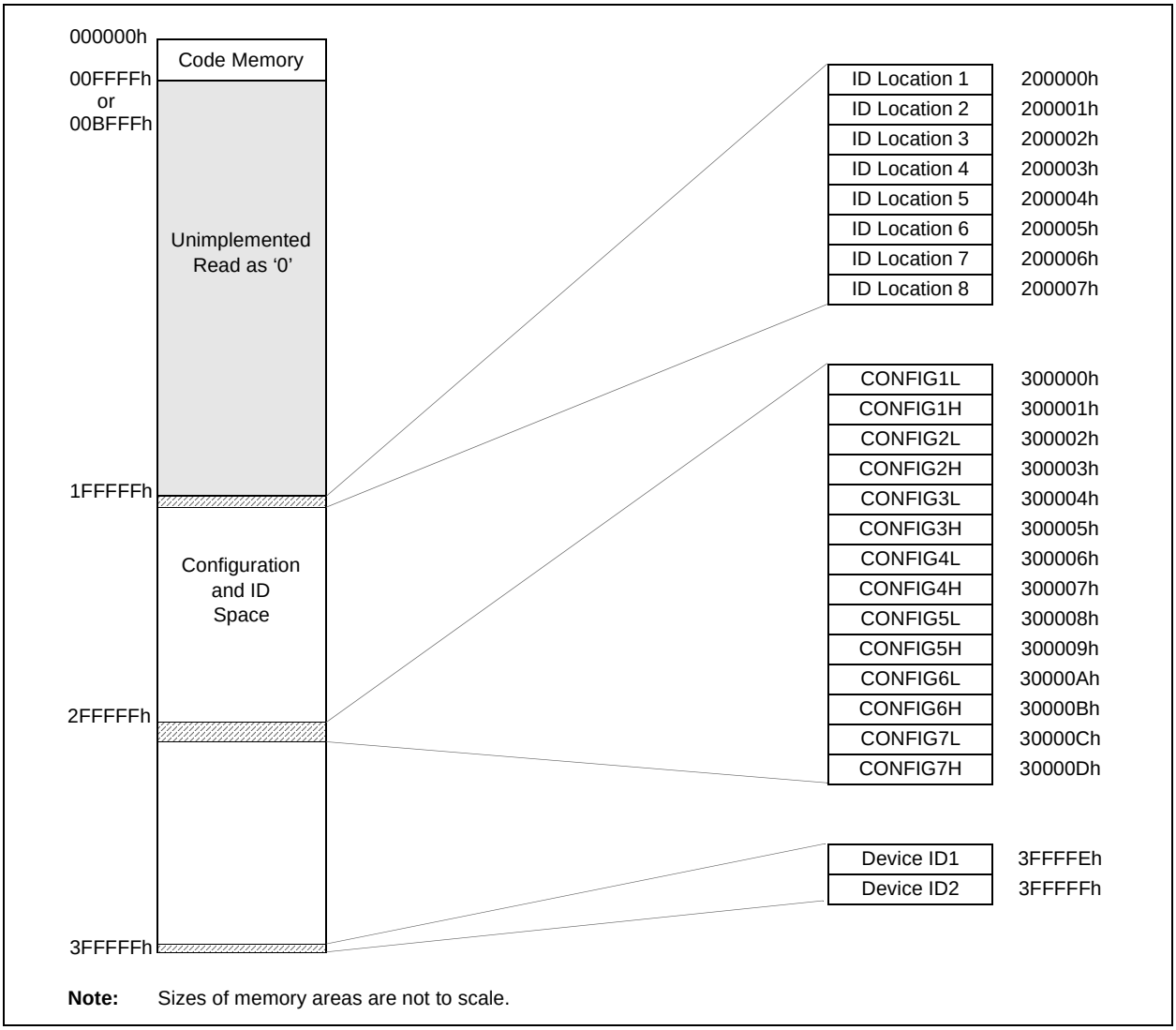
Memory in the address space 000000h to 3FFFFFh is addressed via the table pointer which is comprised of three pointer registers:

- TBLPTRU, at 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 many read or write operations.

FIGURE 2-5: CONFIGURATION AND ID LOCATIONS FOR PIC18FXX80/XX85 DEVICES



Note: Sizes of memory areas are not to scale.

2.4 High Level Overview of the Programming Process

Figure 2-7 shows the high level overview of the programming process. First, 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.

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 $\overline{MCLR}/V_{PP}$ to V_{IHH} (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 Program/Verify mode places all unused I/Os in the high impedance state.

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 $\overline{MCLR}/V_{PP}$ to V_{IH}. 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 Program/Verify mode places all unused I/Os in the high impedance state.

FIGURE 2-7: HIGH LEVEL PROGRAMMING FLOW

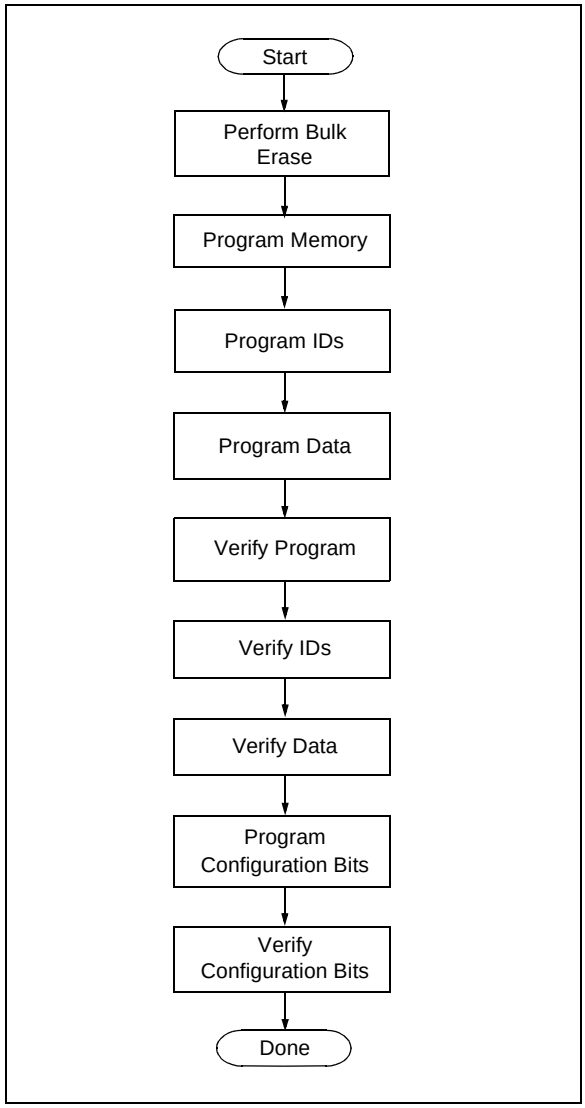


FIGURE 2-6: ENTERING HIGH VOLTAGE PROGRAM/VERIFY MODE

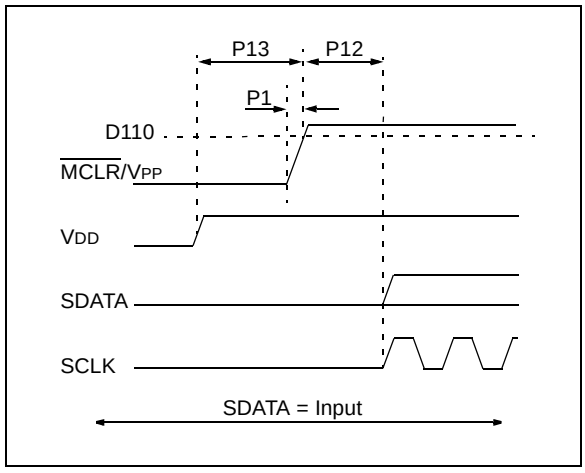
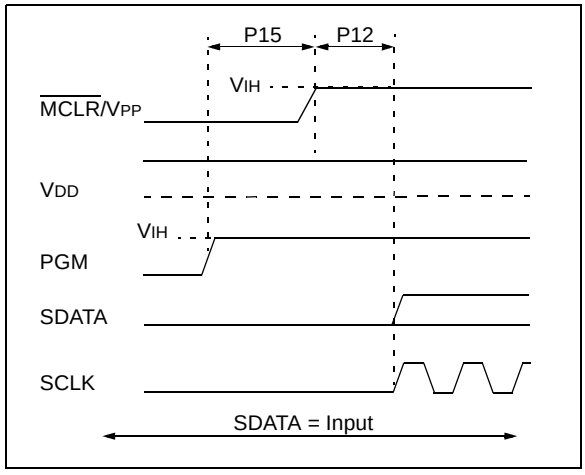


FIGURE 2-8: ENTERING LOW VOLTAGE PROGRAM/ VERIFY MODE



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.

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 Most Significant bit (MSb) first. The command operand, or "Data Payload", is shown <MSB><LSB>. Figure 2-9 demonstrates how to serially present a 20-bit command/operand to the device.

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.

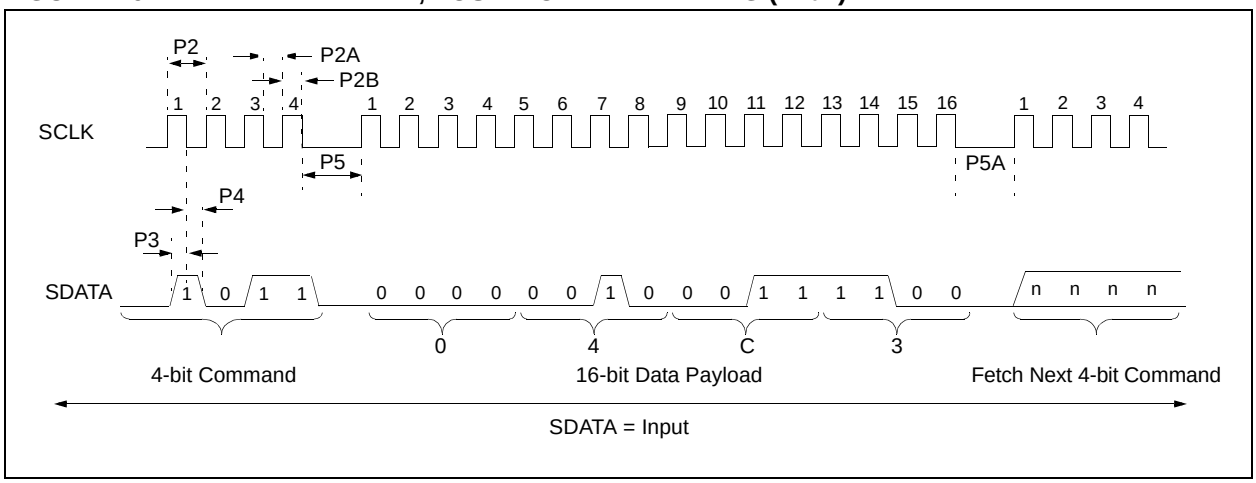
TABLE 2-3: COMMANDS FOR PROGRAMMING

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

TABLE 2-4: SAMPLE COMMAND SEQUENCE

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

FIGURE 2-9: TABLE WRITE, POST-INCREMENT TIMING (1101)



3.0 DEVICE PROGRAMMING

3.1 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
Chip Erase	80h
Erase Data EEPROM	81h
Erase Boot Block	83h
Erase Block 1	88h
Erase Block 2	89h
Erase Block 3	8Ah
Erase Block 4	8Bh

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 the flow chart is shown in Figure 3-1.

Note: A bulk erase is the only way to reprogram code protect bits from an on-state to an off-state.

Non-code protect bits are not returned to default settings by a bulk erase. These bits should be programmed to ones, as outlined in Section 3.6, “Configuration Bits Programming”.

TABLE 3-2: BULK ERASE COMMAND SEQUENCE

4-Bit Command	Data Payload	Core Instruction
0000	8E A6	BSF EECON1, EEPGD
0000	8C A6	BSF EECON1, CFGS
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	00 80	Write 80h TO 3C0004h to erase entire device.
0000	00 00	NOP
0000	00 00	Hold SDATA low until erase completes.

FIGURE 3-1: BULK ERASE FLOW

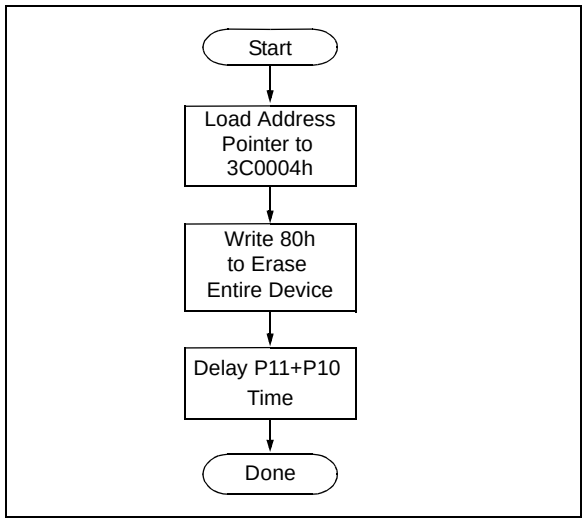
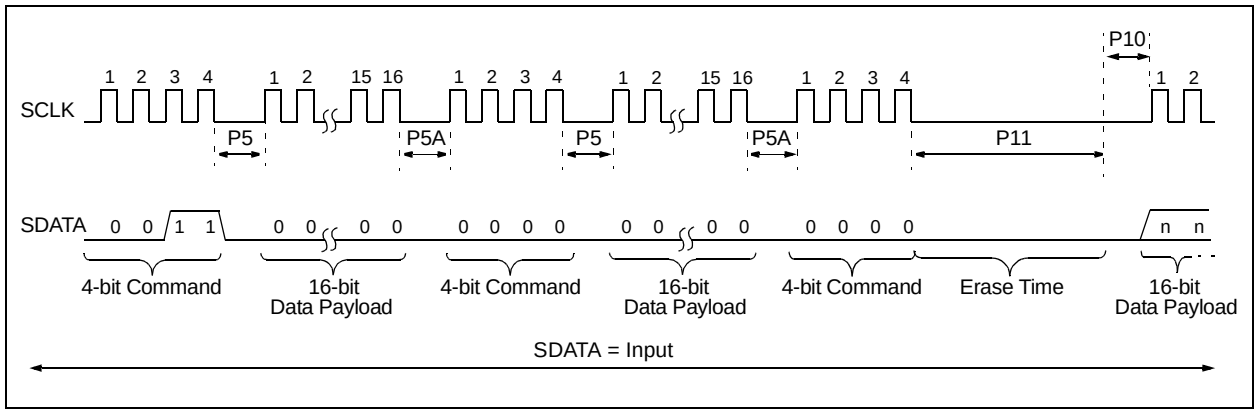


FIGURE 3-2: BULK ERASE TIMING



3.1.1 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 Sections 3.1.2 and 3.2.2.

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 3.3 and write ones to the array.

3.1.2 ICSP MULTI-PANEL SINGLE ROW ERASE

Irrespective of whether high or low voltage ICSP is used, it is possible to erase single row (64 bytes of data) in all panels at once. For example, in the case of a 64-Kbyte device (8 panels), 512 bytes through 64 bytes in each panel can be erased simultaneously during each erase sequence. In this case, the offset of the erase within each panel is the same (see Figure 3-5).

Multi-panel single row erase is enabled by appropriately configuring the Programming Control register located at 3C0006h.

The multi-panel single row erase duration is externally timed and is controlled by SCLK. After a “Start Programming” command is issued (4-bit, ‘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.

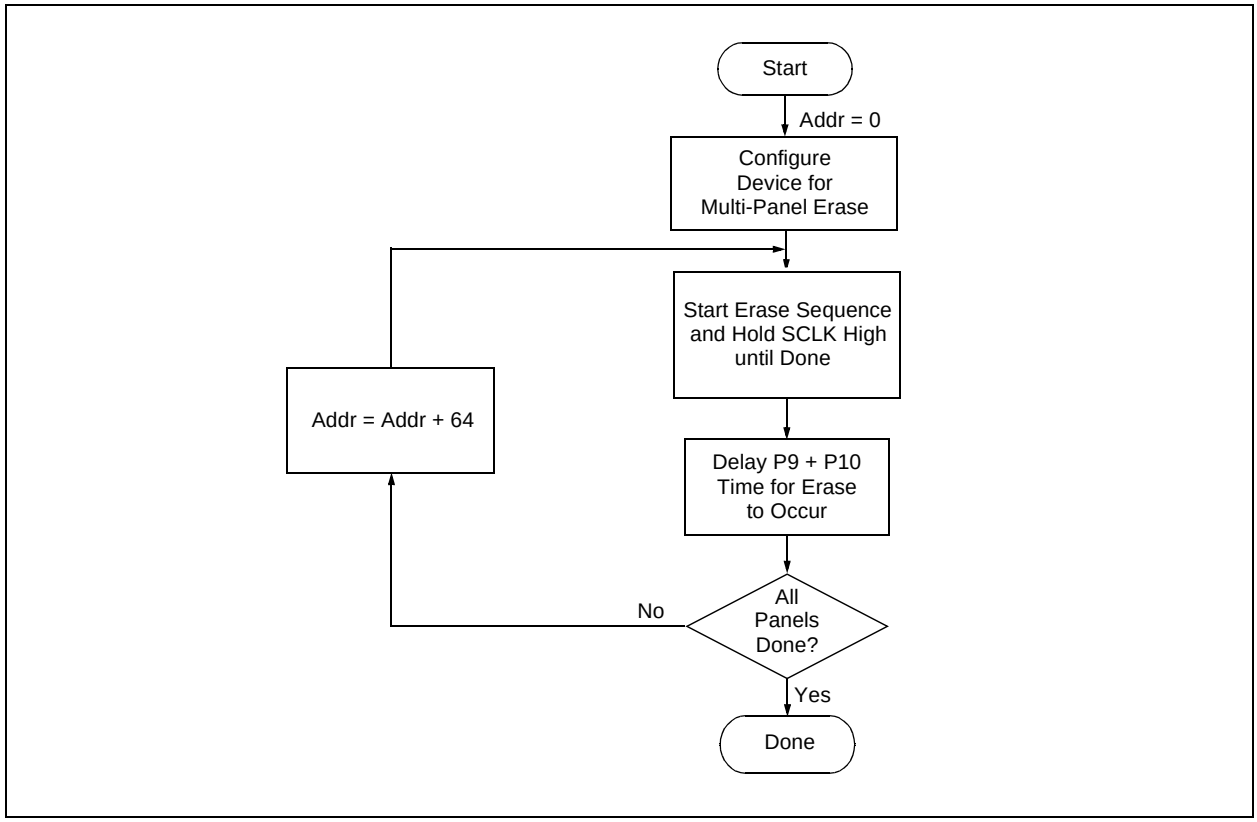
The code sequence to program a PIC18FXX80/XX85 device is shown in Table 3-3. The flow chart shown in Figure 3-3 depicts the logic necessary to completely erase a PIC18FXX80/XX85 device. The timing diagram that details the “Start Programming” command and parameters P9 and P10 is shown in Figure 3-6.

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

TABLE 3-3: ERASE CODE MEMORY CODE SEQUENCE

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to config memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	8C A6	BSF EECON1, CFGS
0000	86 A6	BSF EECON1, WREN
Step 2: 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 40h to 3C0006h to enable multi-panel erase.
Step 3: Direct access to code memory and enable erase.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
0000	88 A6	BSF EECON1, FREE
0000	6A F8	CLRF TBLPTRU
0000	6A F7	CLRF TBLPTRH
0000	6A F6	CLRF TBLPTRL
Step 4: Erase single row of all panels at an offset.		
1111	<DummyLSB> <DummyMSB>	Write 2 dummy bytes and start programming.
0000	00 00	NOP - hold SCLK high for time P9.
Step 5: Repeat step 4, with address pointer incremented by 64 until all panels are erased.		

FIGURE 3-3: MULTI-PANEL SINGLE ROW ERASE CODE MEMORY FLOW



3.2 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 2-4) 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.

Typically, all of the program buffers are written in parallel (Multi-Panel Write mode). For example, in the case of a 64-Kbyte device (8 panels with an 8-byte buffer per panel), 64 bytes will be simultaneously programmed during each programming sequence. In this case, the offset of the write within each panel is the same (see Figure 3-4). Multi-Panel Write mode is enabled by appropriately configuring the Programming Control register located at 3C0006h.

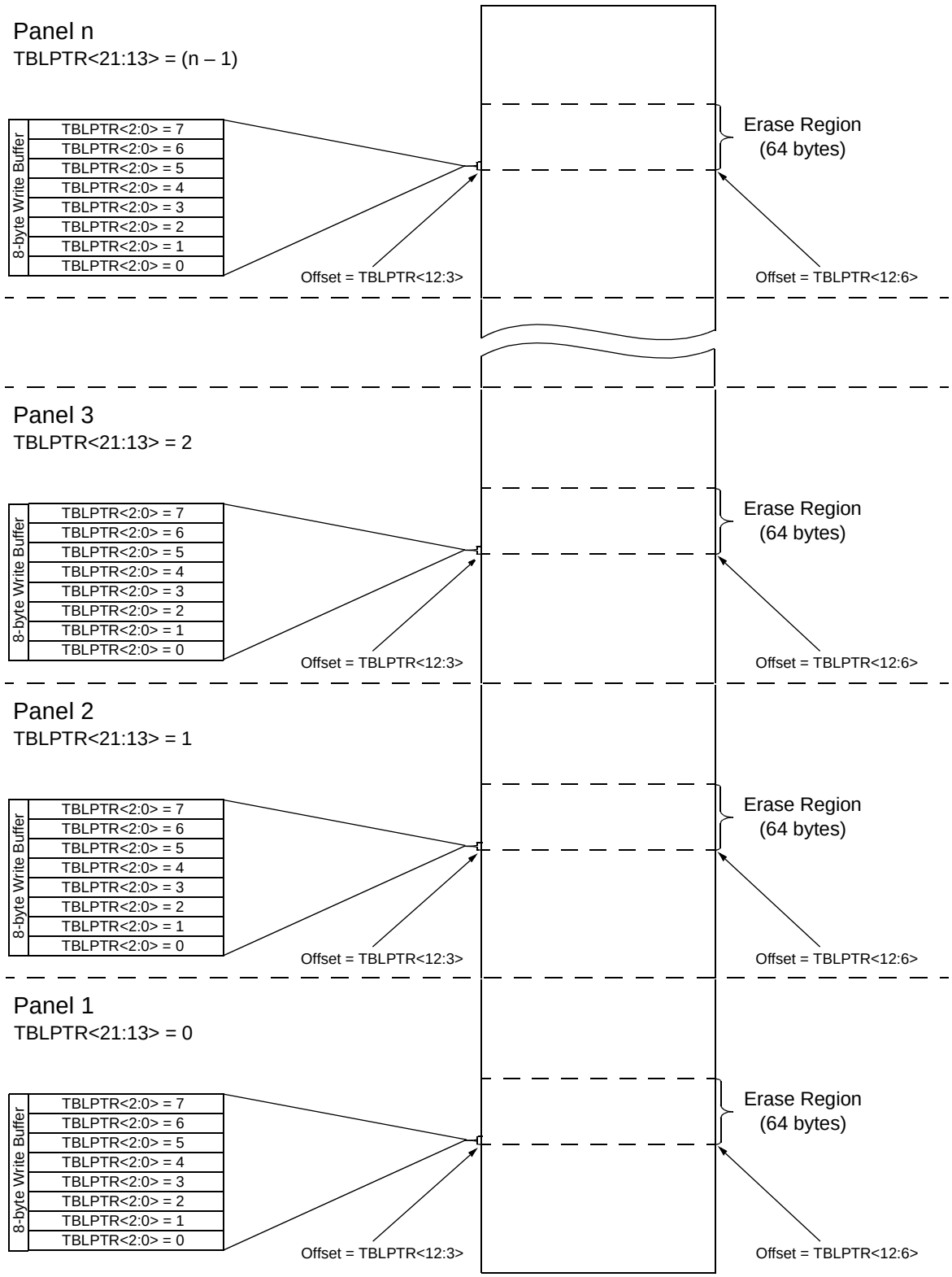
The programming duration is externally timed and is controlled by SCLK. After a “Start Programming” command is issued (4-bit command, ‘1111’), a NOP is issued, where the 4th SCLK is held high for the duration of the programming time, P9.

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 PIC18FXX80/XX85 device is shown in Table 3-4. The flow chart shown in Figure 3-5 depicts the logic necessary to completely write a PIC18FXX80/XX85 device. The timing diagram that details the “Start Programming” command and parameters P9 and P10 is shown in Figure 3-6.

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 3-4: 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: Direct access to config memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	8C A6	BSF EECON1, CFGS
0000	86 A6	BSF EECON1, WREN
Step 2: 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 40h to 3C0006h to enable multi-panel writes.
Step 3: Direct access to code memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
Step 4: Load write buffer for Panel 1.		
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
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
1100	<LSB><MSB>	Write 2 bytes
Step 5: Repeat for Panel 2.		
Step 6: Repeat for all but the last panel (N – 1).		
Step 7: Load write buffer for last panel.		
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
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 steps 2 through 5, where the address pointer is incremented by 8 in each panel at each iteration of the loop.		

FIGURE 3-5: PROGRAM CODE MEMORY FLOW

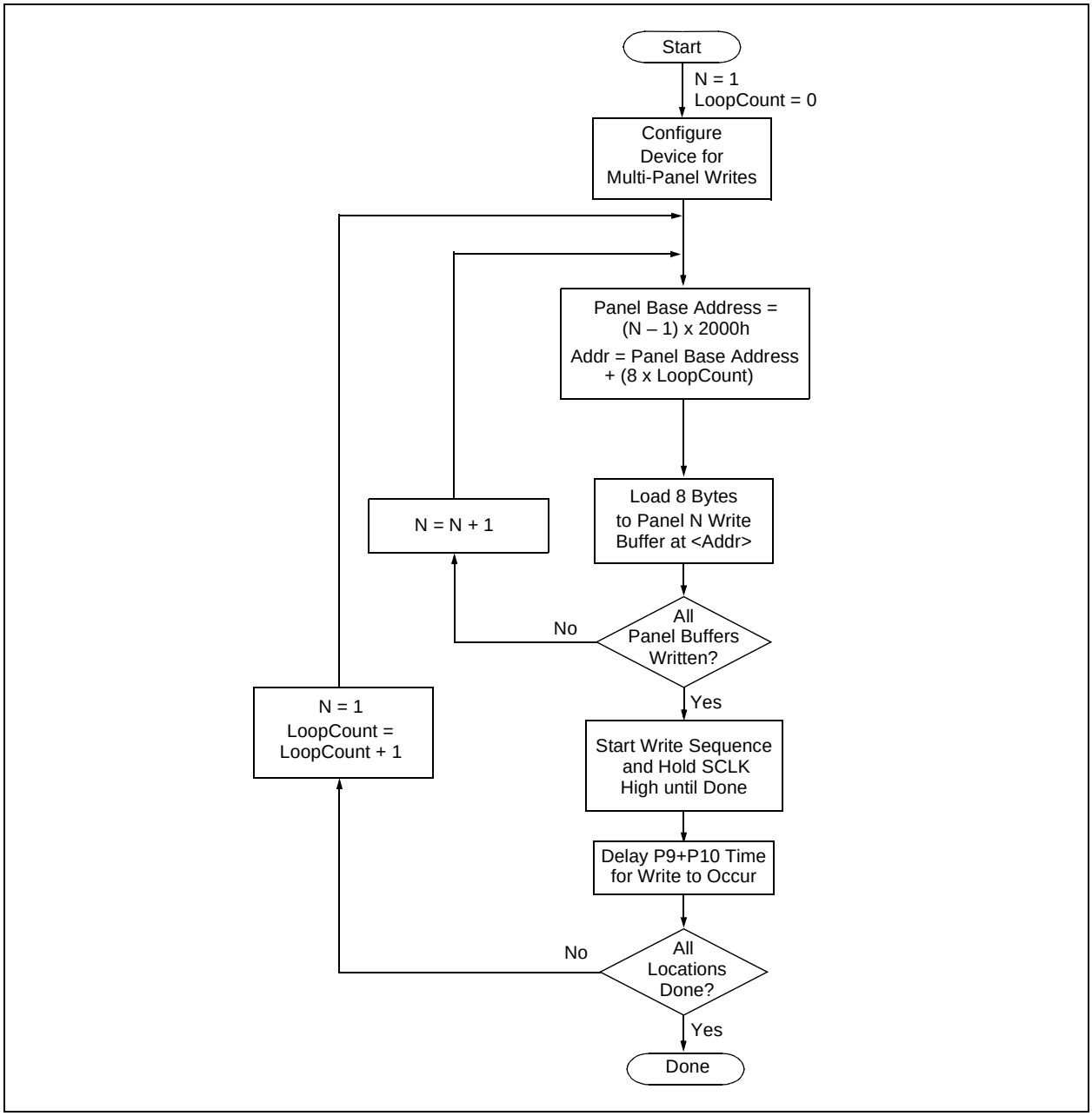
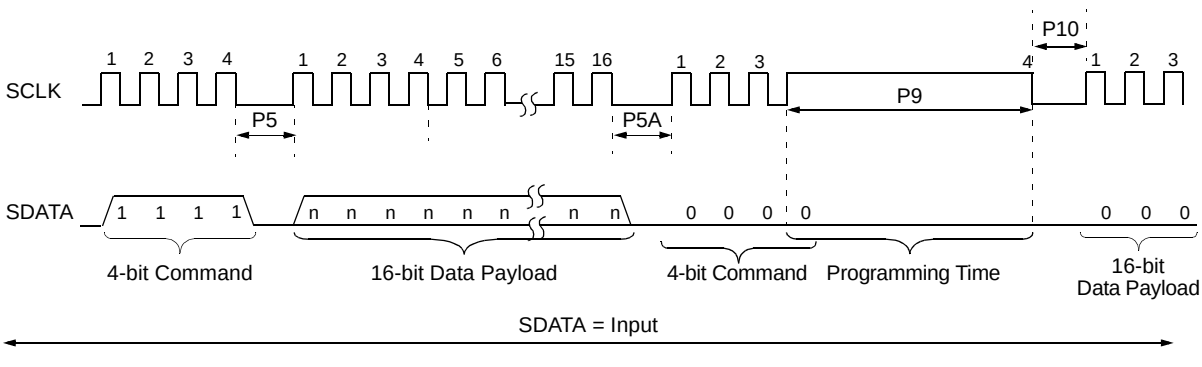


FIGURE 3-6: TABLE WRITE AND START PROGRAMMING INSTRUCTION TIMING (1111)



3.2.1 SINGLE PANEL PROGRAMMING

The programming example presented in Section 3.2 utilizes multi-panel programming. This technique greatly decreases the total amount of time necessary to completely program a device and is the recommended method of completely programming a device.

There may be situations, however, where it is advantageous to limit writes to a single panel. In such cases, the user only needs to disable the multi-panel write feature of the device by appropriately configuring the Programming Control register located at 3C0006h.

The single panel that will be written will automatically be enabled based on the value of the table pointer.

Note:	Even though multi-panel writes are disabled, the user must still fill the 8-byte write buffer for the given panel.

3.2.2 MODIFYING CODE MEMORY

All of the programming examples up to this point have assumed that the device has been bulk erased prior to programming (see Section 3.1). However, it may be the case that the user wishes to modify only a section of an already programmed device.

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 (see Section 3.2.1), loading the 8-byte write buffer for the panel, and then initiating a write sequence. In this case, 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 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.

TABLE 3-5: MODIFYING CODE MEMORY

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to config memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	8C A6	BSF EECON1, CFGS
Step 2: 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 3: Direct access to code memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
Step 4: 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 5: Enable memory writes and setup an erase.		
0000	84 A6	BSF EECON1, WREN
0000	88 A6	BSF EECON1, FREE
Step 6: Perform required sequence.		
0000	0E 55	MOVLW 55h
0000	6E A7	MOVWF EECON2
0000	0E AA	MOVLW 0AAh
0000	6E A7	MOVWF EECON2
Step 7: Initiate erase.		
0000	82 A6	BSF EECON1, WR
0000	00 00	NOP
Step 8: Wait for P11+P10 and then disable writes.		
0000	94 A6	BCF EECON1, WREN
Step 9: Load write buffer for panel. The 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.		

3.3 Data EEPROM Programming

Data EEPROM is accessed one byte at a time via an address pointer (register pair EEADR:EEADRH) and a data latch (EEDATA). Data EEPROM is written by loading EEADR:EEADRH 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, 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). 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 begins on the falling edge of the 4th SCLK after the WR bit is set. It ends when the WR bit is cleared by hardware.

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 3-7: PROGRAM DATA FLOW

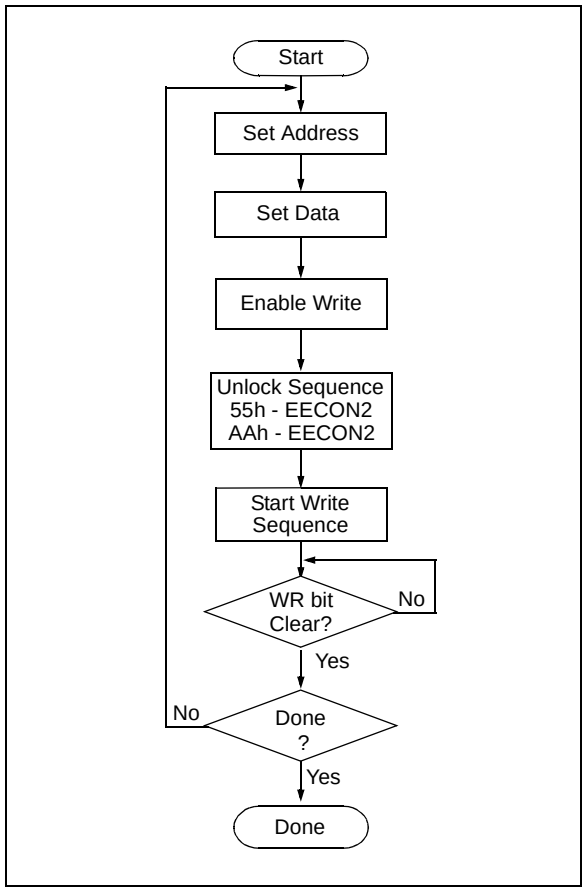


FIGURE 3-8: DATA EEPROM WRITE TIMING

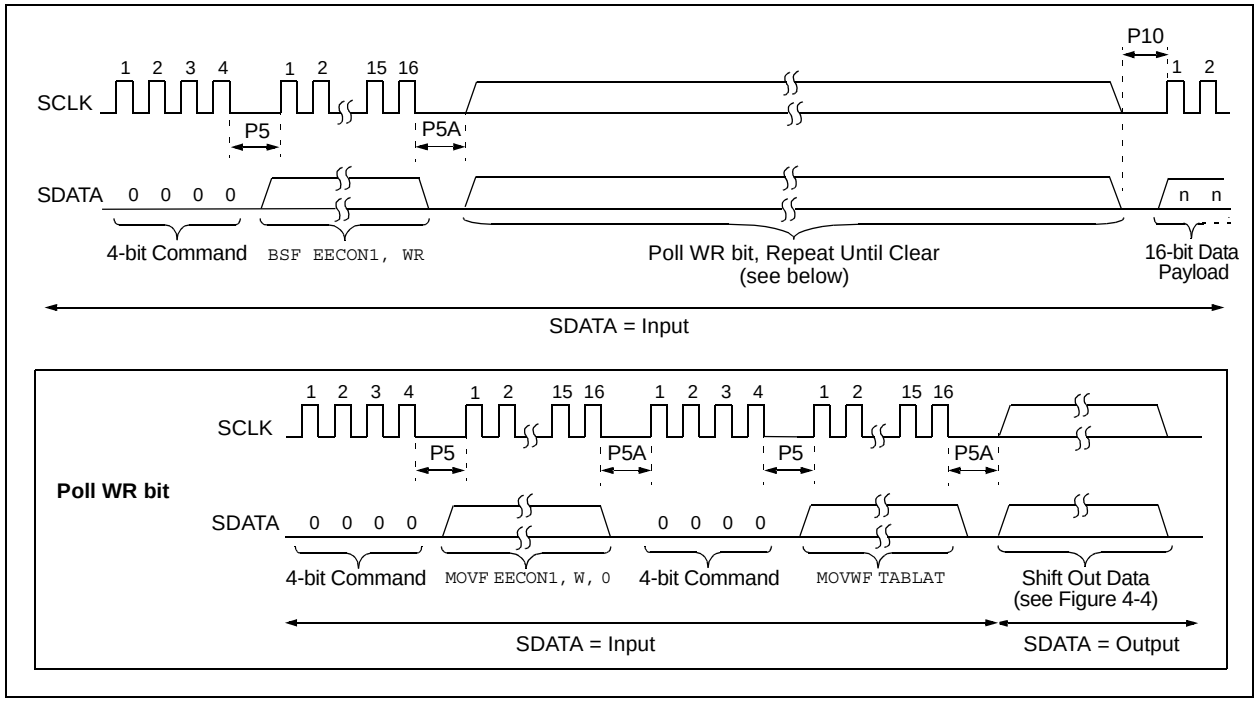


TABLE 3-6: PROGRAMMING DATA MEMORY

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to data EEPROM.		
0000	9E A6	BCF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
Step 2: Set the data EEPROM address pointer.		
0000	0E <Addr>	MOVLW <Addr>
0000	6E A9	MOVWF EEADR
0000	OE <AddrH>	MOVLW <AddrH>
0000	6E AA	MOVWF EEADRH
Step 3: Load the data to be written.		
0000	0E <Data>	MOVLW <Data>
0000	6E A8	MOVWF EEDATA
Step 4: Enable memory writes.		
0000	84 A6	BSF EECON1, WREN
Step 5: Perform required sequence.		
0000	0E 55	MOVLW 55h
0000	6E A7	MOVWF EECON2
0000	0E AA	MOVLW 0AAh
0000	6E A7	MOVWF EECON2
Step 6: Initiate write.		
0000	82 A6	BSF EECON1, WR
Step 7: Poll WR bit, repeat until the bit is clear.		
0000	50 A6	MOVF EECON1, W, 0
0000	6E F5	MOVWF TABLAT
0010	<LSB><MSB>	Shift out data(1)
Step 8: Disable writes.		
0000	94 A6	BCF EECON1, WREN
Repeat steps 2 through 8 to write more data.		

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

3.4 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:	Even though multi-panel writes are disabled, 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: Direct access to config memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	8C A6	BSF EECON1, CFGS
Step 2: 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 3: Direct access to code memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
Step 4: 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

In order to modify the ID locations, refer to the methodology described in Section 3.2.2, “Modifying Code Memory”. As with code memory, the ID locations must be erased before modified.

3.5 Boot Block Programming

The boot block segment is programmed in exactly the same manner as the ID locations (see Section 3.4). Multi-panel writes must be disabled so that only addresses in the range 0000h to 07FFh will be written.

The code sequence detailed in Table 3-7 should be used except that the address data used in “Step 2” will be in the range 000000h to 0007FFh.

3.6 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 Table 3-8.

Note: Execute four NOPs between every configuration byte programming.

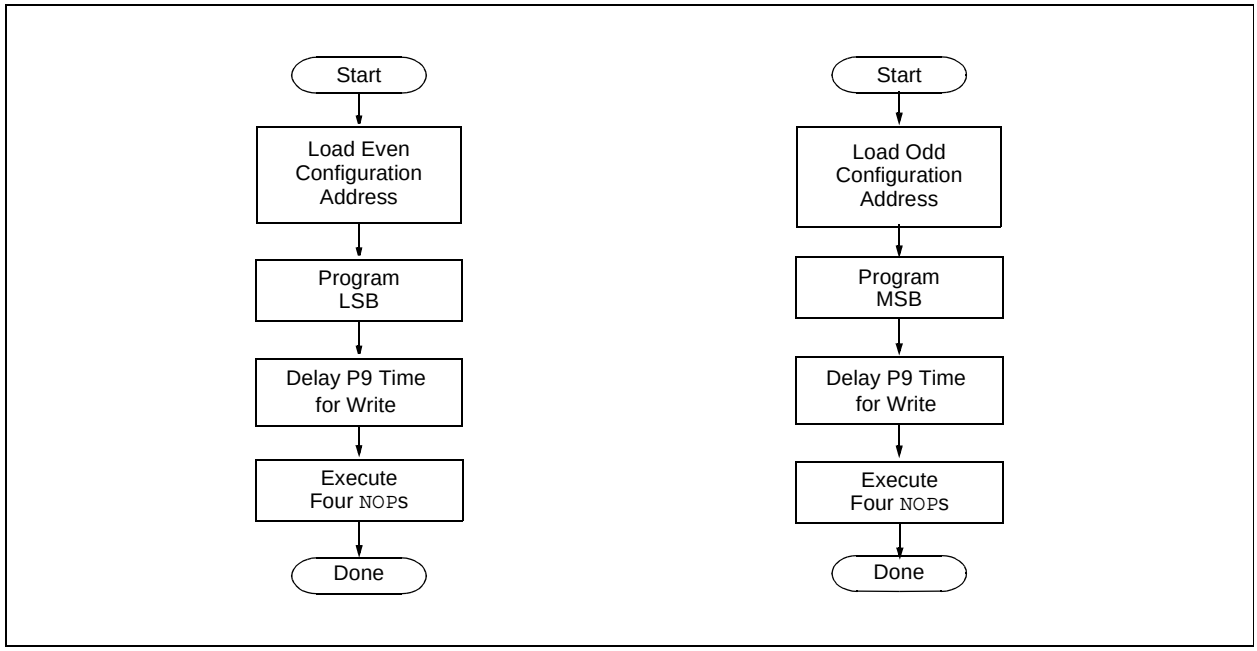
TABLE 3-8: SET ADDRESS POINTER TO CONFIGURATION LOCATION

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to config memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	8C A6	BSF EECON1, CFGS
Step 2: Position the program counter(1).		
0000	EF 00	GOTO 100000h
0000	F8 00	
Step 3(2): Set table pointer for config byte 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
Step 4: Execute four NOPs.		
0000	00 00	
0000	00 00	
0000	00 00	
0000	00 00	

Note 1: If the code protection bits are programmed while the program counter resides in the same block, then the interaction of code protection logic may prevent further table write. To avoid this situation, move the program counter outside the code protection area (e.g., GOTO 100000h).

2: 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 3-9: CONFIGURATION PROGRAMMING FLOW



4.0 READING THE DEVICE

4.1 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.

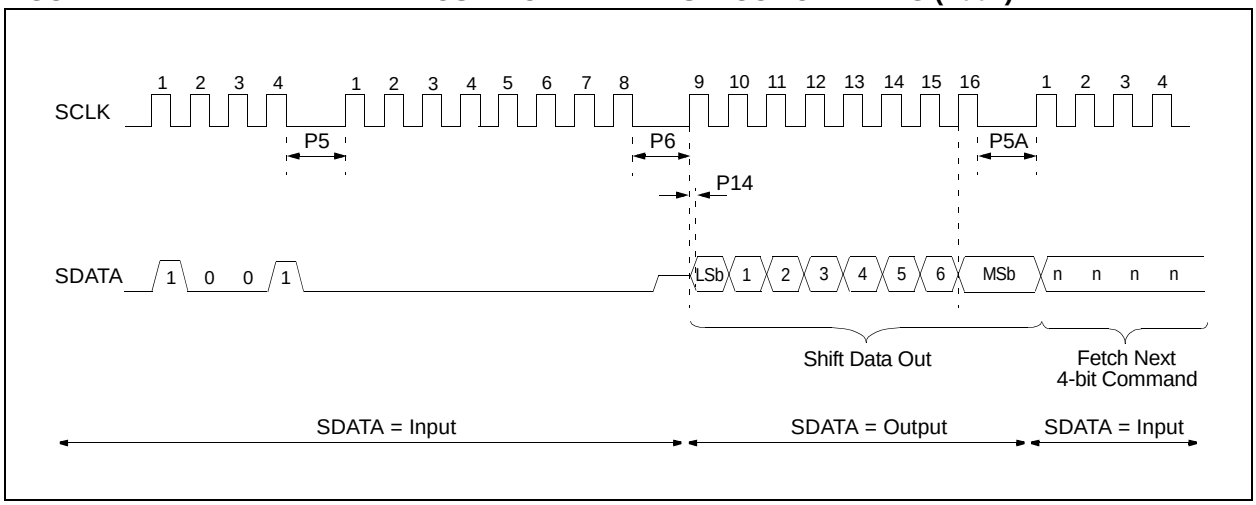
The 4-bit command is shifted in LSb first. The read is executed during the next 8 clocks, then shifted out on SDATA during the last 8 clocks, LSb to MSb. A delay of P6 must be introduced after the falling edge of the 8th SCLK of the operand to allow SDATA to transition from an input to an output. During this time, SCLK must be held low (see Figure 4-1). 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 4-1: 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 4-1: TABLE READ POST-INCREMENT INSTRUCTION TIMING (1001)

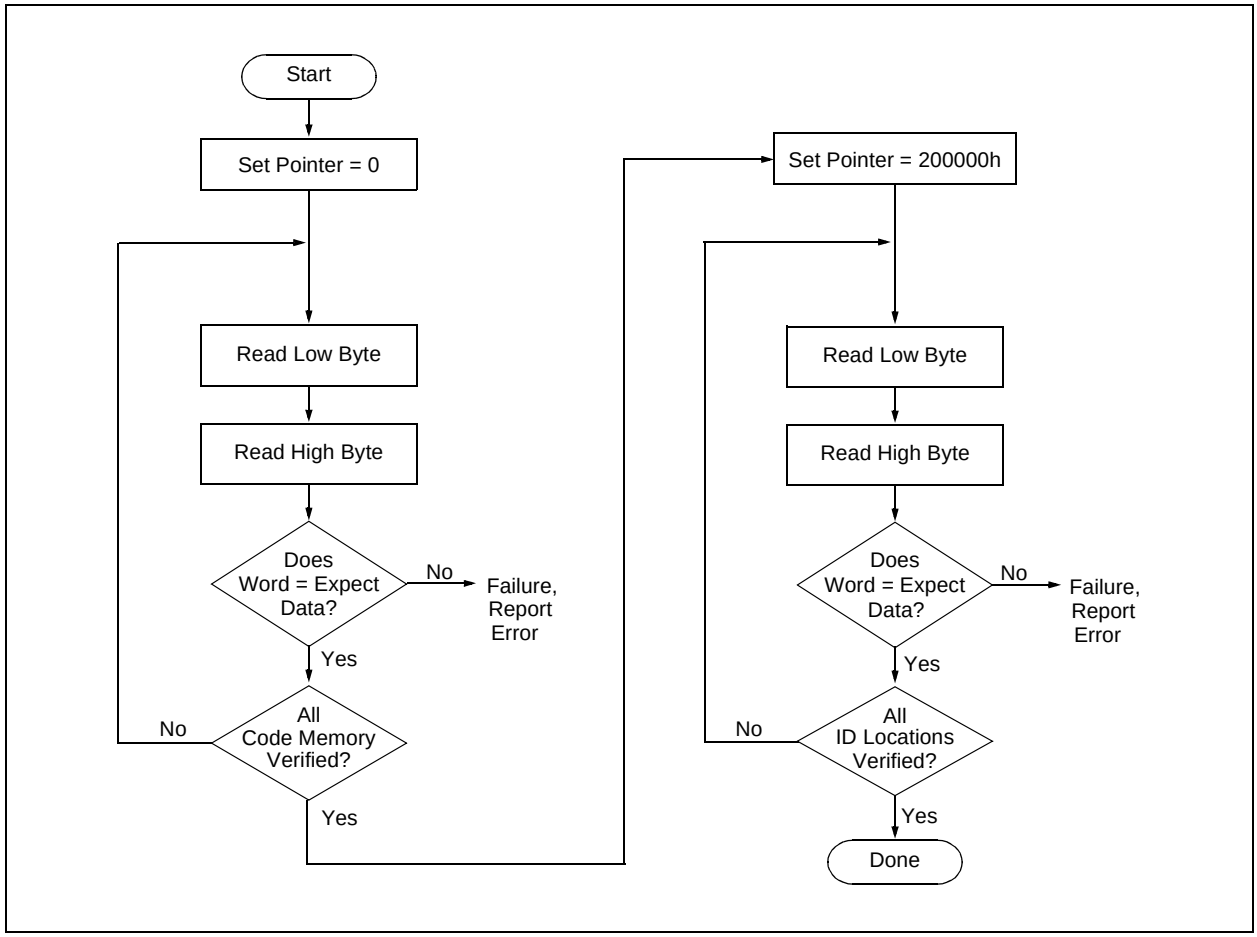


4.2 Verify Code Memory and ID Locations

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 4.1 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 the code memory space. In a 48-Kbyte device, for example, a post-increment read of address 0BFFFh will wrap the table pointer back to 0000h, rather than point to unimplemented address 0C000h.

FIGURE 4-2: VERIFY CODE MEMORY FLOW



4.3 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 4.1 for implementation details of reading configuration data.

4.4 Read Data EEPROM Memory

Data EEPROM is accessed one byte at a time via an address pointer (register pair EEADR:EEADRH) and a data latch (EEDATA). Data EEPROM is read by loading EEADR:EEADRH 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 4-4).

The command sequence to read a single byte of data is shown in Table 4-2.

FIGURE 4-3: READ DATA EEPROM FLOW

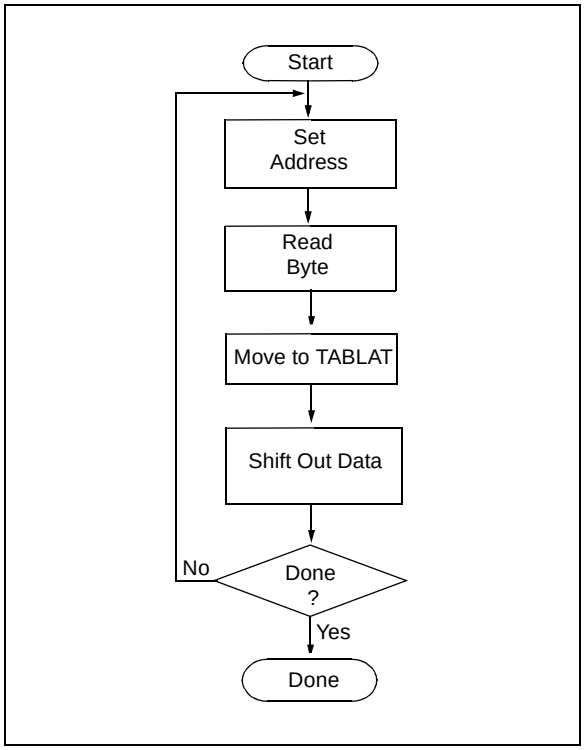
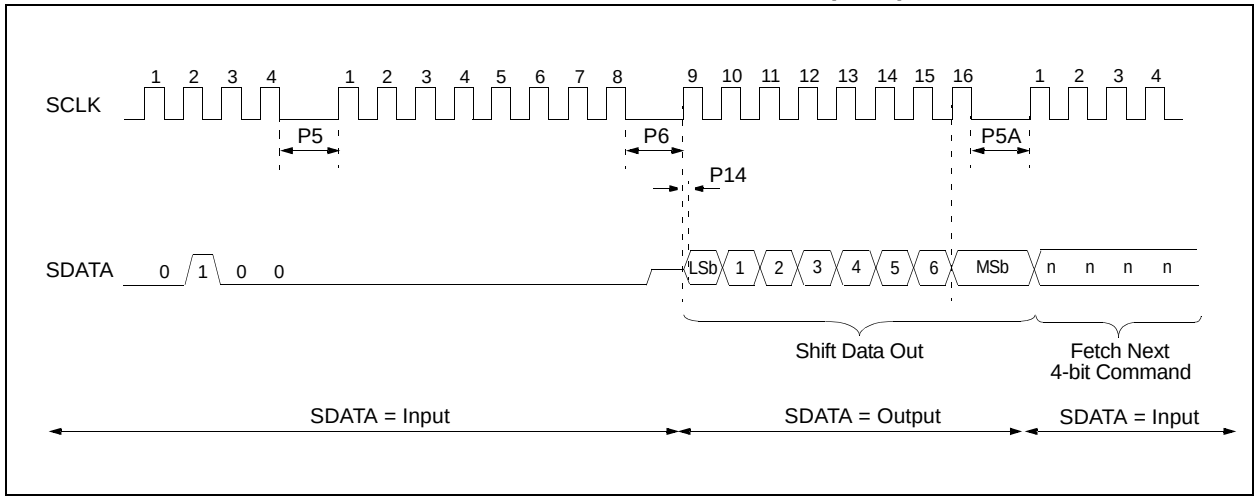


TABLE 4-2: READ DATA EEPROM MEMORY

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



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 4.4 for implementation details of reading data EEPROM.

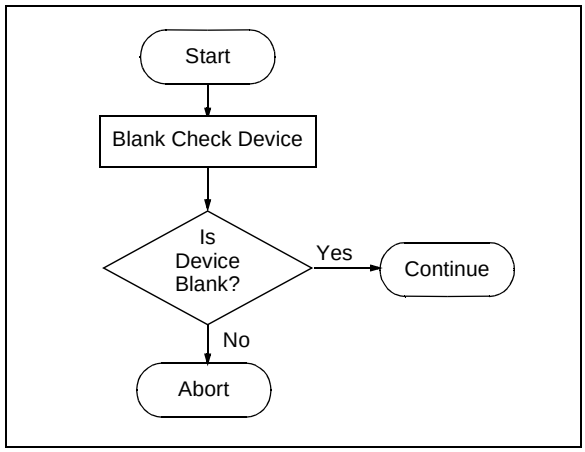
4.6 Blank Check

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 PIC18FXX80/XX85 devices.

Given that "Blank Checking" is merely code and data EEPROM verification with FFh expect data, refer to Section 4.2 and Section 4.4 for implementation details.

FIGURE 4-5: BLANK CHECK FLOW



5.0 CONFIGURATION WORD

The PIC18FXX80/XX85 devices have 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 PIC18FXX80/XX85 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 VALUES

Device	Device ID Value	
	DEVID2	DEVID1
PIC18F6585	0Ah	011x xxxx
PIC18F6680	0Ah	001x xxxx
PIC18F8585	0Ah	010x xxxx
PIC18F8680	0Ah	000x xxxx

TABLE 5-2: PIC18FXX80/XX85 CONFIGURATION BITS AND DEVICE IDS

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
300001h	CONFIG1H	—	—	$\overline{\text{OSCSEN}}$	—	FOSC3	FOSC2	FOSC1	FOSC0	--1- 1111
300002h	CONFIG2L	—	—	—	—	BORV1	BORV0	BODEN	$\overline{\text{PWRTEN}}$	---- 1111
300003h	CONFIG2H	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN	---1 1111
300004h(1)	CONFIG3L	WAIT	—	—	—	—	—	PM1	PM0	1--- --11
300005h	CONFIG3H	MCLRE	—	—	—	—	—	r	r	1--- --00
300006h	CONFIG4L	$\overline{\text{DEBUG}}$	—	—	—	—	LVP	—	STVREN	1--- -1-1
300008h	CONFIG5L	—	—	—	—	CP3(2)	CP2	CP1	CP0	---- 1111
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—	11-- ----
30000Ah	CONFIG6L	—	—	—	—	WRT3(2)	WRT2	WRT1	WRT0	---- 1111
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	—	—	—	—	—	111- ----
30000Ch	CONFIG7L	—	—	—	—	EBTR3(2)	EBTR2	EBTR1	EBTR0	---- 1111
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—	-1-- ----
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

Legend: x = unknown, u = unchanged, – = unimplemented, q = value depends on condition, r = reserved.
Shaded cells are unimplemented, read as '0'.

Note 1: Unimplemented in PIC18F6X8X devices; maintain this bit set.

2: Unimplemented in PIC18FX585 devices; maintain this bit set.

TABLE 5-3: PIC18FXX80/XX85 CONFIGURATION BIT DESCRIPTIONS

Bit Name	Configuration Bytes	Description
$\overline{\text{OSCEN}}$	CONFIG1H	Low Power System Clock Option (Timer1) Enable bit 1 = Disabled 0 = Timer1 oscillator system clock option enabled
FOSC3:FOSC0	CONFIG1H	Oscillator Selection bits 1111 = RC oscillator w/ OSC2 configured as RA6 1110 = HS oscillator w/ software controlled PLL 1101 = EC oscillator with OSC2 configured as RA6 w/ SW controlled PLL 1100 = EC oscillator with OSC2 configured as RA6 w/ PLL enabled 1011 = Reserved; do not use 1010 = Reserved; do not use 1001 = Reserved; do not use 1000 = Reserved; do not use 0111 = RC oscillator w/ OSC2 configured as RA6 0110 = HS oscillator w/ PLL enabled 0101 = EC oscillator w/ OSC2 configured as RA6 0100 = EC oscillator w/ OSC2 configured as “divide by 4 clock output” 0011 = RC oscillator 0010 = HS oscillator 0001 = XT oscillator 0000 = LP oscillator
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
WDTPS3:WDTPS0	CONFIG2H	Watchdog Timer Postscaler Select bits 1111 = 1:32768 1110 = 1:16384 1101 = 1:8192 1100 = 1:4096 1011 = 1:2048 1010 = 1:1024 1001 = 1:512 1000 = 1:256 0111 = 1:128 0110 = 1:64 0101 = 1:32 0100 = 1:16 0011 = 1:8 0010 = 1:4 0001 = 1:2 0000 = 1:1
WDTEN	CONFIG2H	Watchdog Timer Enable bit 1 = WDT enabled 0 = WDT disabled (control is placed on SWDTEN bit)

Note 1: Unimplemented in PIC18F6X8X (64-pin) devices; maintain this bit set.

2: Unimplemented in PIC18FX585 devices; maintain this bit set.

TABLE 5-3: PIC18FXX80/XX85 CONFIGURATION BIT DESCRIPTIONS (CONTINUED)

Bit Name	Configuration Bytes	Description
WAIT(1)	CONFIG3L	External Bus Data Wait Enable bit 1 = Wait selections unavailable 0 = Wait selections determined by WAIT1:WAIT0 bits of MEMCOM register
PM1:PM0(1)	CONFIG3L	Processor Mode Select bits 11 = Microcontroller mode 10 = Microprocessor mode 01 = Microprocessor with Boot Block mode 00 = Extended Microcontroller mode
MCLRE	CONFIG3H	$\overline{\text{MCLR}}$ Enable bit 1 = $\overline{\text{MCLR}}$ pin enabled, RG5 disabled 0 = $\overline{\text{MCLR}}$ pin disabled, RG5 enabled
$\overline{\text{DEBUG}}$	CONFIG4L	Background Debugger Enable bit 1 = Background debugger disabled (RB6, RB7 are I/O pins) 0 = Background debugger enabled (RB6, RB7 are ISCP pins)
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 0800h - 3FFFh) 1 = Code memory not code protected 0 = Code memory code protected
CP1	CONFIG5L	Code Protection bits (code memory area 4000h - 7FFFh) 1 = Code memory not code protected 0 = Code memory code protected
CP2	CONFIG5L	Code Protection bits (code memory area 8000h - 0BFFFh) 1 = Code memory not code protected 0 = Code memory code protected
CP3(2)	CONFIG5L	Code Protection bits (code memory area 0C000h - 0FFFFh) 1 = Code memory not code protected 0 = Code memory code protected
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 - 07FFh) 1 = Boot block not code protected 0 = Boot block code protected
WRT0	CONFIG6L	Table Write Protection bit (code memory area 0800h - 3FFFh) 1 = Code memory not write protected 0 = Code memory write protected
WRT1	CONFIG6L	Table Write Protection bit (code memory area 4000h - 7FFFh) 1 = Code memory not write protected 0 = Code memory write protected
WRT2	CONFIG6L	Table Write Protection bit (code memory area 8000h - 0BFFFh) 1 = Code memory not write protected 0 = Code memory write protected
WRT3(2)	CONFIG6L	Table Write Protection bit (code memory area 0C000h - 0FFFFh) 1 = Code memory not write protected 0 = Code memory write protected

Note 1: Unimplemented in PIC18F6X8X (64-pin) devices; maintain this bit set.

2: Unimplemented in PIC18FX585 devices; maintain this bit set.

TABLE 5-3: PIC18FXX80/XX85 CONFIGURATION BIT DESCRIPTIONS (CONTINUED)

Bit Name	Configuration Bytes	Description
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 - 07FFh) 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 0800h - 3FFFh) 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 4000h - 7FFFh) 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 8000h - 0BFFFh) 1 = Code memory not protected from table reads executed in other blocks 0 = Code memory protected from table reads executed in other blocks
EBTR3(2)	CONFIG7L	Table Read Protection bit (code memory area 0C000h - 0FFFFh) 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 - 07FFh) 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 These bits are used with the DEV2:DEV0 bits in the DEVID1 register to identify part number.
DEV2:DEV0	DEVID1	Device ID bits These 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.

Note 1: Unimplemented in PIC18F6X8X (64-pin) devices; maintain this bit set.

2: Unimplemented in PIC18FX585 devices; maintain this bit set.

5.4 Embedding Configuration Word Information in the HEX File

To allow portability of code, a PIC18FXX80/XX85 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 (pages 32 through 35) describes how to calculate the checksum for each device.

Note:	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	Checksum	Blank Value	0AAh at 0 and Max Address
PIC18F6585	None	SUM(0000:07FFh) + SUM(0800:3FFFh) + SUM(4000:7FFFh) + SUM(8000:0BFFFh) + (CFGW1L & 00h) + (CFGW1H & 2Fh) + (CFGW2L & 0Fh) + (CFGW2H & 1Fh) + (CFGW3L & 00h) + (CFGW3H & 80h) + (CFGW4L & 85h) + (CFGW4H & 00h) + (CFGW5L & 07h) + (CFGW5H & 0C0h) + (CFGW6L & 07h) + (CFGW6H & 0E0h) + (CFGW7L & 07h) + (CFGW7H & 40h)	0C357h	0C2ADh
	Boot Block	SUM(0800:3FFFh) + SUM(4000:7FFFh) + SUM(8000:0BFFFh) + (CFGW1L & 00h) + (CFGW1H & 2Fh) + (CFGW2L & 0Fh) + (CFGW2H & 1Fh) + (CFGW3L & 00h) + (CFGW3H & 80h) + (CFGW4L & 85h) + (CFGW4H & 00h) + (CFGW5L & 07h) + (CFGW5H & 80h) + (CFGW6L & 07h) + (CFGW6H & 0E0h) + (CFGW7L & 07h) + (CFGW7H & 40h) + SUM(IDs)	0BB32h	0BAE7h
	Boot, Block 0, Block 1	SUM(8000:0BFFFh) + (CFGW1L & 00h) + (CFGW1H & 2Fh) + (CFGW2L & 0Fh) + (CFGW2H & 1Fh) + (CFGW3L & 00h) + (CFGW3H & 80h) + (CFGW4L & 85h) + (CFGW4H & 00h) + (CFGW5L & 04h) + (CFGW5H & 80h) + (CFGW6L & 07h) + (CFGW6H & 0E0h) + (CFGW7L & 07h) + (CFGW7H & 40h) + SUM(IDs)	432Fh	42E4h
	All	(CFGW1L & 00h) + (CFGW1H & 2Fh) + (CFGW2L & 0Fh) + (CFGW2H & 1Fh) + (CFGW3L & 00h) + (CFGW3H & 80h) + (CFGW4L & 85h) + (CFGW4H & 00h) + (CFGW5L & 00h) + (CFGW5H & 00h) + (CFGW6L & 07h) + (CFGW6H & 0E0h) + (CFGW7L & 07h) + (CFGW7H & 40h) + SUM(IDs)	02A6h	02B5h

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	Checksum	Blank Value	0AAh at 0 and Max Address
PIC18F6680	None	SUM(0000:07FFh) + SUM(0800:3FFFh) + SUM(4000:7FFFh) + SUM(8000:0BFFFh) + SUM(0C000:0FFFFh) + (CFGW1L & 00h) + (CFGW1H & 2Fh) + (CFGW2L & 0Fh) + (CFGW2H & 1Fh) + (CFGW3L & 00h) + (CFGW3H & 80h) + (CFGW4L & 85h) + (CFGW4H & 00h) + (CFGW5L & 0Fh) + (CFGW5H & 0C0h) + (CFGW6L & 0Fh) + (CFGW6H & 0E0h) + (CFGW7L & 0Fh) + (CFGW7H & 40h)	036Fh	02C5h
	Boot Block	SUM(0800:3FFFh) + SUM(4000:7FFFh) + SUM(8000:0BFFFh) + SUM(0C000:0FFFFh) + (CFGW1L & 00h) + (CFGW1H & 2Fh) + (CFGW2L & 0Fh) + (CFGW2H & 1Fh) + (CFGW3L & 00h) + (CFGW3H & 80h) + (CFGW4L & 85h) + (CFGW4H & 00h) + (CFGW5L & 0Fh) + (CFGW5H & 80h) + (CFGW6L & 0Fh) + (CFGW6H & 0E0h) + (CFGW7L & 0Fh) + (CFGW7H & 40h) + SUM(IDs)	0FB47h	0FAEDh
	Boot/ Block1/ Block2	SUM(8000:0BFFFh) + SUM(0C000:0FFFFh) + (CFGW1L & 00h) + (CFGW1H & 2Fh) + (CFGW2L & 0Fh) + (CFGW2H & 1Fh) + (CFGW3L & 00h) + (CFGW3H & 80h) + (CFGW4L & 85h) + (CFGW4H & 00h) + (CFGW5L & 0Ch) + (CFGW5H & 80h) + (CFGW6L & 0Fh) + (CFGW6H & 0E0h) + (CFGW7L & 0Fh) + (CFGW7H & 40h) + SUM(IDs)	8344h	82EAh
	All	(CFGW1L & 00h) + (CFGW1H & 2Fh) + (CFGW2L & 0Fh) + (CFGW2H & 1Fh) + (CFGW3L & 00h) + (CFGW3H & 80h) + (CFGW4L & 85h) + (CFGW4H & 00h) + (CFGW5L & 00h) + (CFGW5H & 00h) + (CFGW6L & 0Fh) + (CFGW6H & 0E0h) + (CFGW7L & 0Fh) + (CFGW7H & 40h) + SUM(IDs)	02B8h	02B3h

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	Checksum	Blank Value	0AAh at 0 and Max Address
PIC18F8585	None	SUM(0000:07FFh) + SUM(0800:3FFFh) + SUM(4000:7FFFh) + SUM(8000:0BFFFh) + (CFGW1L & 00h) + (CFGW1H & 2Fh) + (CFGW2L & 0Fh) + (CFGW2H & 1Fh) + (CFGW3L & 83h) + (CFGW3H & 80h) + (CFGW4L & 85h) + (CFGW4H & 00h) + (CFGW5L & 07h) + (CFGW5H & 0C0h) + (CFGW6L & 07h) + (CFGW6H & 0E0h) + (CFGW7L & 07h) + (CFGW7H & 40h)	0C3DAh	0C330h
	Boot Block	SUM(0800:3FFFh) + SUM(4000:7FFFh) + SUM(8000:0BFFFh) + (CFGW1L & 00h) + (CFGW1H & 2Fh) + (CFGW2L & 0Fh) + (CFGW2H & 1Fh) + (CFGW3L & 83h) + (CFGW3H & 80h) + (CFGW4L & 85h) + (CFGW4H & 00h) + (CFGW5L & 07h) + (CFGW5H & 80h) + (CFGW6L & 07h) + (CFGW6H & 0E0h) + (CFGW7L & 07h) + (CFGW7H & 40h) + SUM(IDs)	0BBC0h	0BB57h
	Boot/ Block1/ Block2	SUM(8000:0BFFFh) + (CFGW1L & 00h) + (CFGW1H & 2Fh) + (CFGW2L & 0Fh) + (CFGW2H & 1Fh) + (CFGW3L & 83h) + (CFGW3H & 80h) + (CFGW4L & 85h) + (CFGW4H & 00h) + (CFGW5L & 04h) + (CFGW5H & 80h) + (CFGW6L & 07h) + (CFGW6H & 0E0h) + (CFGW7L & 07h) + (CFGW7H & 40h) + SUM(IDs)	43BDh	4354h
	All	(CFGW1L & 00h) + (CFGW1H & 2Fh) + (CFGW2L & 0Fh) + (CFGW2H & 1Fh) + (CFGW3L & 83h) + (CFGW3H & 80h) + (CFGW4L & 85h) + (CFGW4H & 00h) + (CFGW5L & 00h) + (CFGW5H & 00h) + (CFGW6L & 07h) + (CFGW6H & 0E0h) + (CFGW7L & 07h) + (CFGW7H & 40h) + SUM(IDs)	0339h	0325h

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	Checksum	Blank Value	0AAh at 0 and Max Address
PIC18F8680	None	SUM(0000:07FFh) + SUM(0800:3FFFh) + SUM(4000:7FFFh) + SUM(8000:0BFFFh) + SUM(0C000:0FFFFh) + (CFGW1L & 00h) + (CFGW1H & 2Fh) + (CFGW2L & 0Fh) + (CFGW2H & 1Fh) + (CFGW3L & 83h) + (CFGW3H & 80h) + (CFGW4L & 85h) + (CFGW4H & 00h) + (CFGW5L & 0Fh) + (CFGW5H & 0C0h) + (CFGW6L & 0Fh) + (CFGW6H & 0E0h) + (CFGW7L & 0Fh) + (CFGW7H & 40h)	03F2h	0348h
	Boot Block	SUM(0800:3FFFh) + SUM(4000:7FFFh) + SUM(8000:0BFFFh) + SUM(0C000:0FFFFh) + (CFGW1L & 00h) + (CFGW1H & 2Fh) + (CFGW2L & 0Fh) + (CFGW2H & 1Fh) + (CFGW3L & 83h) + (CFGW3H & 80h) + (CFGW4L & 85h) + (CFGW4H & 00h) + (CFGW5L & 0Fh) + (CFGW5H & 80h) + (CFGW6L & 0Fh) + (CFGW6H & 0E0h) + (CFGW7L & 0Fh) + (CFGW7H & 40h) + SUM(IDs)	0FBC6h	0FB6Ch
	Boot/ Block1/ Block2	SUM(8000:0BFFFh) + SUM(0C000:0FFFFh) + (CFGW1L & 00h) + (CFGW1H & 2Fh) + (CFGW2L & 0Fh) + (CFGW2H & 1Fh) + (CFGW3L & 83h) + (CFGW3H & 80h) + (CFGW4L & 85h) + (CFGW4H & 00h) + (CFGW5L & 0Ch) + (CFGW5H & 80h) + (CFGW6L & 0Fh) + (CFGW6H & 0E0h) + (CFGW7L & 0Fh) + (CFGW7H & 40h) + SUM(IDs)	83C3h	8369h
	All	(CFGW1L & 00h) + (CFGW1H & 2Fh) + (CFGW2L & 0Fh) + (CFGW2H & 1Fh) + (CFGW3L & 83h) + (CFGW3H & 80h) + (CFGW4L & 85h) + (CFGW4H & 00h) + (CFGW5L & 00h) + (CFGW5H & 00h) + (CFGW6L & 0Fh) + (CFGW6H & 0E0h) + (CFGW7L & 0Fh) + (CFGW7H & 40h) + SUM(IDs)	0337h	0332h

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 PIC18FXX80/XX85 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 $\overline{MCLR}$/VPP	9.00	13.25	V	
D110A	VIHL	Low Voltage Programming Voltage on $\overline{MCLR}$/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 $\overline{MCLR}$/VPP	—	300	µA	
D113	IDDP	Supply Current During Programming	—	10	mA	
D031	VIL	Input Low Voltage	VSS	0.2 VDD	V	
D041	VIH	Input High Voltage	0.8 VDD	VDD	V	
D080	VOL	Output Low Voltage	—	0.6	V	IOL = 8.5 mA @ 4.5V
D090	VOH	Output High Voltage	VDD – 0.7	—	V	IOH = -3.0 mA @ 4.5V
D012	CIO	Capacitive Loading on I/O pin (SDATA)	—	50	pF	To meet AC specifications
P1	Tr	$\overline{MCLR}$/VPP Rise Time to enter Program/Verify mode	—	1.0	µs	**(Note 1)**
P2	Tsclk	Serial Clock (SCLK) Period	100	—	ns	
P2A	TsclkL	Serial Clock (SCLK) Low Time	40	—	ns	
P2B	TsclkH	Serial Clock (SCLK) High Time	40	—	ns	
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	40	—	ns	
P5A	Tdly1a	Delay between 4-bit Command Operand and next 4-bit Command	40	—	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	5	—	ms	
P11A	Tdrwt	Data Write Polling Time	4	—	ms	
P12	Thld2	Input Data Hold Time from $\overline{MCLR}$/VPP ↑	2	—	µs	
P13	Tset2	VDD ↑ Setup Time to $\overline{MCLR}$/VPP ↑	100	—	ns	
P14	Tvalid	Data Out Valid from SCLK ↑	10	—	ns	
P15	Tset3	PGM ↑ Setup Time to $\overline{MCLR}$/VPP ↑	2	—	µs	

Note 1: Do not allow excess time when transitioning $\overline{MCLR}$ between VIL and VIHH; this can cause spurious program executions to occur. The maximum transition time is:

1 TCY + TPWRT (if enabled) + 1024 TOSC (for LP, HS, HS/PLL and XT modes only)
+ 2 ms (for HS/PLL mode only) + 1.5 µs (for EC mode only)

where TCY is the Instruction Cycle Time, TPWRT is the Power-up Timer Period, and TOSC is the Oscillator Period. For specific values, refer to the Electrical Characteristics section of the device data sheet for the particular device.

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

Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 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: 480-792-7627
Web Address: http://www.microchip.com

Atlanta
3780 Mansell Road, Suite 130
Alpharetta, GA 30022
Tel: 770-640-0034 Fax: 770-640-0307

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848 Fax: 978-692-3821

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423 Fax: 972-818-2924

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

Kokomo
2767 S. Albright Road
Kokomo, IN 46902
Tel: 765-864-8360 Fax: 765-864-8387

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

Phoenix
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966 Fax: 480-792-4338

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699 Fax: 905-673-6509

ASIA/PACIFIC

Australia
Microchip Technology Australia Pty Ltd
Marketing Support Division
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

China - Beijing
Microchip Technology Consulting (Shanghai) Co., Ltd., Beijing Liaison Office
Unit 915
Bei Hai Wan Tai Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

China - Chengdu
Microchip Technology Consulting (Shanghai) Co., Ltd., Chengdu Liaison Office
Rm. 2401-2402, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-86766200 Fax: 86-28-86766599

China - Fuzhou
Microchip Technology Consulting (Shanghai) Co., Ltd., Fuzhou Liaison Office
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506 Fax: 86-591-7503521

China - Hong Kong SAR
Microchip Technology Hongkong Ltd.
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

China - Shanghai
Microchip Technology Consulting (Shanghai) Co., Ltd.
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

China - Shenzhen
Microchip Technology Consulting (Shanghai) Co., Ltd., Shenzhen Liaison Office
Rm. 1812, 18/F, Building A, United Plaza
No. 5022 Binhe Road, Futian District
Shenzhen 518033, China
Tel: 86-755-82901380 Fax: 86-755-8295-1393

China - Qingdao
Rm. B505A, Fullhope Plaza,
No. 12 Hong Kong Central Rd.
Qingdao 266071, China
Tel: 86-532-5027355 Fax: 86-532-5027205

India
Microchip Technology Inc.
India Liaison Office
Marketing Support Division
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

Japan
Microchip Technology Japan K.K.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5932 or 82-2-558-5934

Singapore
Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

Taiwan
Microchip Technology (Barbados) Inc., Taiwan Branch
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Austria
Microchip Technology Austria GmbH
Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark
Microchip Technology Nordic ApS
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45-4420-9895 Fax: 45-4420-9910

France
Microchip Technology SARL
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany
Microchip Technology GmbH
Steinheilstrasse 10
D-85737 Ismaning, Germany
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy
Microchip Technology SRL
Via Quasimodo, 12
20025 Legnano (MI)
Milan, Italy
Tel: 39-0331-742611 Fax: 39-0331-466781

Netherlands
Microchip Technology Netherlands
P. A. De Biesbosch 14
NL-5152 SC Drunen, Netherlands
Tel: 31-416-690399 Fax: 31-416-690340

United Kingdom
Microchip Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44-118-921-5869 Fax: 44-118-921-5820

07/10/03