PIC18FXXK80 FAMILY

Flash Microcontroller Programming Specification

1.0 DEVICE OVERVIEW

This document includes the programming specifications for the following devices:

- PIC18F25K80
- PIC18LF25K80
- PIC18F45K80
- PIC18LF45K80
- PIC18F65K80
- PIC18LF65K80
- PIC18F26K80
- PIC18LF26K80
- PIC18F46K80
- PIC18LF46K80
- PIC18F66K80
- PIC18LF66K80

2.0 PROGRAMMING OVERVIEW

The PIC18FXXK80 family of devices can be programmed using the In-Circuit Serial Programming™ (ICSP™) method. This programming specification applies to the PIC18FXXK80 family of devices in all package types.

2.1 Hardware Requirements

When programming with the ICSP, the PIC18FXXK80 family requires two programmable power supplies; one for VDD and one for MCLR/VPP/RE3. Both supplies should have a minimum resolution of 0.25V. Refer to **Section 6.0 "AC/DC Characteristics Timing Requirements for Program/Verify Test Mode"** for additional hardware parameters.

2.1.1 LOW-VOLTAGE ICSP™ PROGRAMMING

In Low-Voltage ICSP mode, the PIC18FXXK80 family can be programmed using a VDD source in the operating range. The MCLR/VPP/RE3 pin 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 "AC/DC Characteristics Timing Requirements for Program/Verify Test Mode"** for additional hardware parameters.

2.2 Pin Diagrams

The pin diagrams for the PIC18FXXK80 family are shown in Figure 2-1 and Figure 2-2.

TABLE 2-1: PIN DESCRIPTIONS (DURING PROGRAMMING): PIC18FXXK80 FAMILY

Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
MCLR/VPP/RE3	VPP	P	Programming Enable
VDD(1)	VDD	P	Power Supply
VSS(1)	VSS	P	Ground
AVDD	AVDD	P	Analog Power Supply
AVSS	AVSS	P	Analog Ground
RB6	PGC	I	Serial Clock
RB7	PGD	I/O	Serial Data
VDDCORE/ VCAP	VDDCORE	P	Regulated Power Supply for Microcontroller Core
	VCAP	I	Filter Capacitor for On-Chip Voltage Regulator

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

Note 1: All power supply (VDD) and ground (VSS) pins must be connected.

FIGURE 2-1: PIC18FXXK80 FAMILY PIN DIAGRAMS

28-Pin QFN

The following devices are included in 28-pin QFN parts:

- PIC18F25K80
- PIC18F26K80
- PIC18LF25K80
- PIC18LF26K80

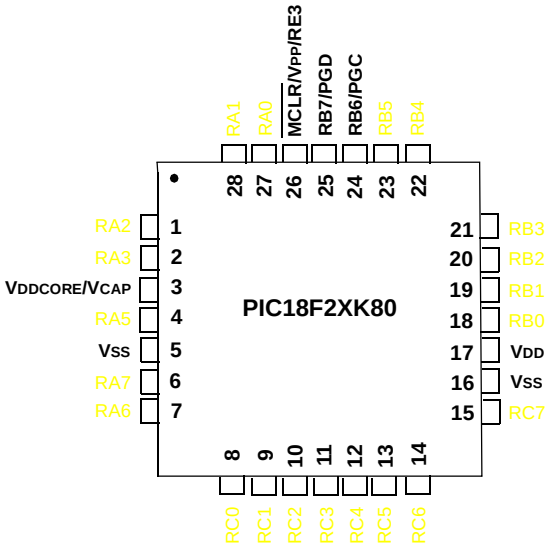


FIGURE 2-2: PIC18F8XKXX FAMILY PIN DIAGRAMS

28-PIN PDIP/SOIC/SSOP

The following devices are included in 28-pin PDIP/SOIC/SSOP parts:

- PIC18F25K80
- PIC18F26K80
- PIC18LF25K80
- PIC18LF26K80

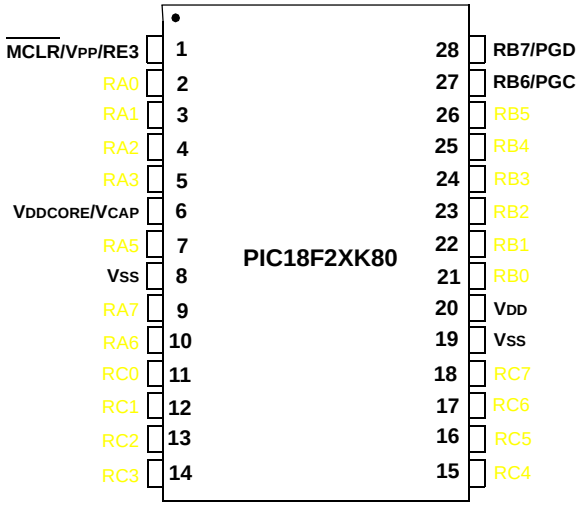


FIGURE 2-3: PIC18F8XKXX FAMILY PIN DIAGRAMS

40-PIN PDIP

The following devices are included in 40-pin PDIP parts:

- PIC18F45K80
- PIC18F46K80
- PIC18LF45K80
- PIC18LF46K80

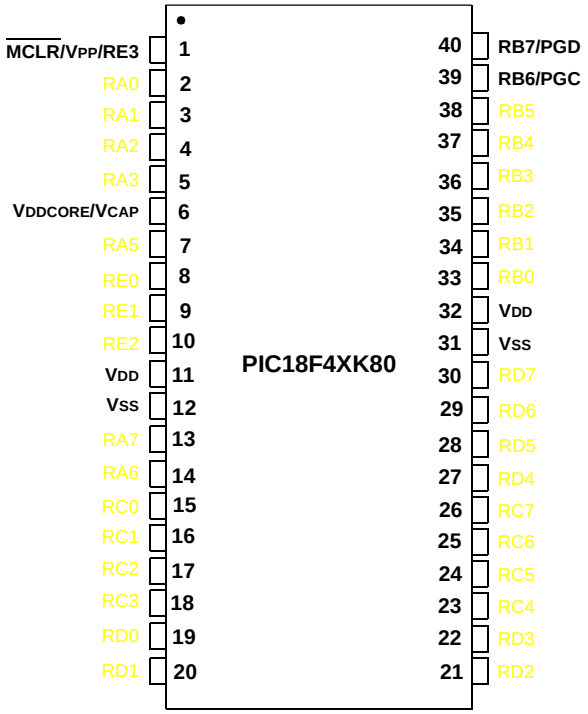


FIGURE 2-4: PIC18F8XKXX FAMILY PIN DIAGRAMS

44-PIN TQFP/QFN

The following devices are included in 44-pin TQFP/QFN parts:

- PIC18F45K80
- PIC18F46K80
- PIC18LF45K80
- PIC18LF46K80

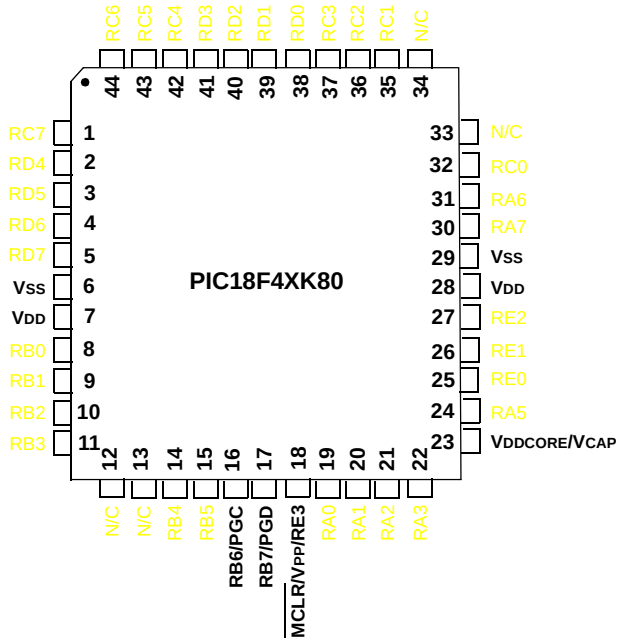
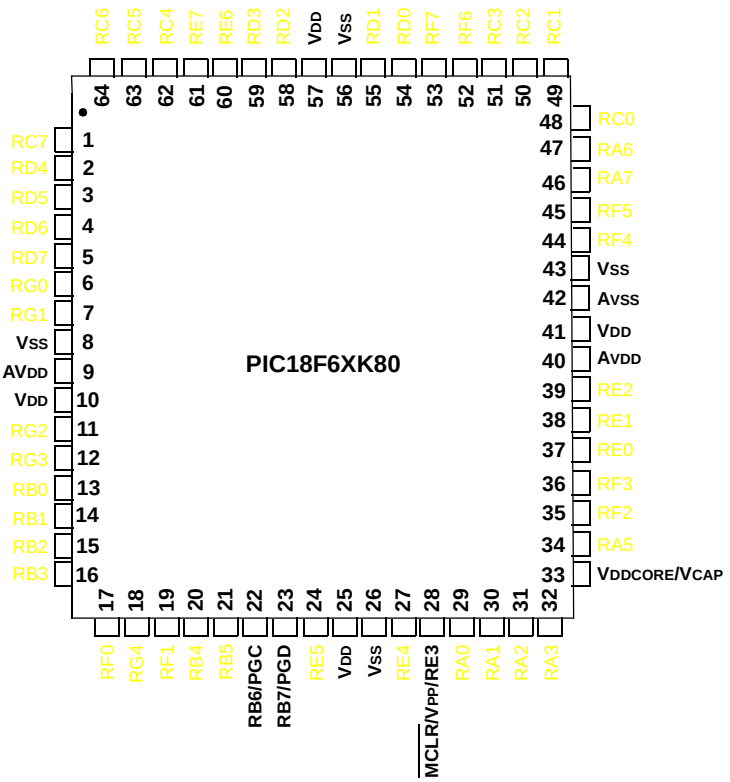


FIGURE 2-5: PIC18F8XKXX FAMILY PIN DIAGRAMS

64-PIN TQFP/QFN

The following devices are included in 64-pin TQFP/QFN parts:

- PIC18F65K80
- PIC18F66K80
- PIC18LF65K80
- PIC18LF66K80



2.3 On-Chip Voltage Regulator

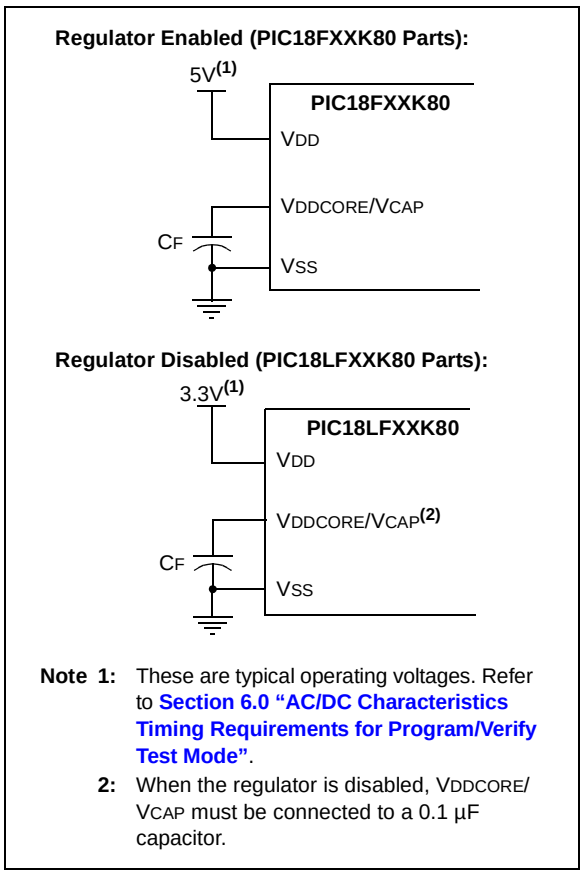
The PIC18FXXK80 device family is available with or without an internal core voltage regulator.

On the devices with a voltage regulator ("PIC18**F**" in the part number), the regulator is always enabled. The regulator input is taken from the microcontroller V_{DD} pins. The output of the regulator is supplied internally to the V_{DDCORE}/V_{CAP} pin. This pin simultaneously serves as both the regulator output and the microcontroller core power input pin. For these devices, a low-ESR (< 5Ω) capacitor is required on the V_{CAP}/V_{DDCORE} pin to stabilize the voltage regulator output voltage. The V_{CAP}/V_{DDCORE} pin must not be connected to V_{DD} and must use a capacitor that is typically 10 µF connected to ground.

On the devices that do not have a voltage regulator ("PIC18**LF**" in the part number), power to the CPU core must be externally supplied through the microcontroller V_{DD} pins. V_{DDCORE}/V_{CAP} is internally connected to V_{DD}. A 0.1 µF capacitor should be connected to the V_{DDCORE}/V_{CAP} pin. Examples are shown in Figure 2-6.

The specifications for core voltage and capacitance are listed in **Section 6.0 "AC/DC Characteristics Timing Requirements for Program/Verify Test Mode"**.

FIGURE 2-6: CONNECTIONS FOR THE ON-CHIP REGULATOR



Note 1: These are typical operating voltages. Refer to **Section 6.0 "AC/DC Characteristics Timing Requirements for Program/Verify Test Mode"**.

2: When the regulator is disabled, V_{DDCORE}/V_{CAP} must be connected to a 0.1 µF capacitor.

2.4 Memory Maps

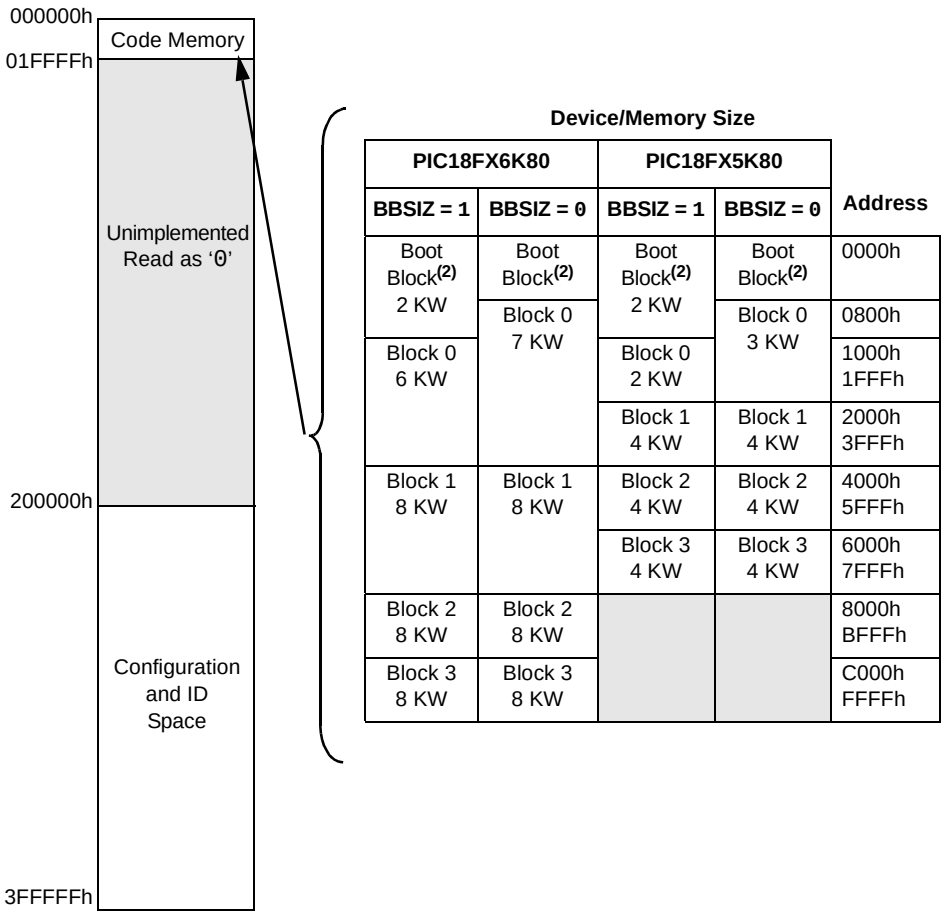
For PIC18FX6K80 devices, the code memory space extends from 000000h to 00FFFFh (64 Kbytes) in four 16-Kbyte blocks. For PIC18FX5K80 devices, the code memory space extends from 000000h to 007FFFh (32 Kbytes) in four 8-Kbyte blocks. Addresses, 0000h through 07FFh or 0FFFh, however, define a “Boot Block” region that is treated separately from Block 0. All of these blocks define code protection boundaries within the code memory space.

The size of the Boot Block in PIC18FXXK80 devices can be configured as 1 or 2K words (see Table 5-3). This is done through the BBSIZ bit in the Configuration register, CONFIG4L (see Table 5-1). It is important to note that increasing the size of the Boot Block decreases the size of Block 0.

TABLE 2-2: IMPLEMENTATION OF CODE MEMORY

Device	Code Memory Size (Bytes)
PIC18F65K80	000000h-007FFFh (32K)
PIC18F45K80	
PIC18F25K80	
PIC18LF65K80	
PIC18LF45K80	
PIC18LF25K80	
PIC18F66K80	000000h-00FFFFh (64K)
PIC18F46K80	
PIC18F26K80	
PIC18LF66K80	
PIC18LF46K80	
PIC18LF26K80	

FIGURE 2-7: MEMORY MAP AND THE CODE MEMORY SPACE FOR PIC18FXXK80 DEVICES(1)



Device/Memory Size

PIC18FX6K80		PIC18FX5K80		Address
BBSIZ = 1	BBSIZ = 0	BBSIZ = 1	BBSIZ = 0	
Boot Block(2) 2 KW	Boot Block(2)	Boot Block(2) 2 KW	Boot Block(2)	0000h
	Block 0 7 KW		Block 0 3 KW	0800h
Block 0 6 KW		Block 0 2 KW		1000h 1FFFh
		Block 1 4 KW	Block 1 4 KW	2000h 3FFFh
Block 1 8 KW	Block 1 8 KW	Block 2 4 KW	Block 2 4 KW	4000h 5FFFh
		Block 3 4 KW	Block 3 4 KW	6000h 7FFFh
Block 2 8 KW	Block 2 8 KW			8000h BFFFh
Block 3 8 KW	Block 3 8 KW			C000h FFFFh

Note 1: Sizes of memory areas are not to scale.

2: Boot block size is determined by the BBSIZ bit (CONFIG4L<4>).

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 2-8.

Users may store Identification (ID) information 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 "Configuration Word"**. These Configuration bits read out normally, even after code protection.

Locations, 3FFFFEh and 3FFFFFh, are reserved for the Device ID bits. These bits may be used by the programmer to identify what device type is being programmed and are described in **Section 5.0 "Configuration Word"**. These Device ID bits read out normally, even after code protection.

2.4.1 MEMORY ADDRESS POINTER

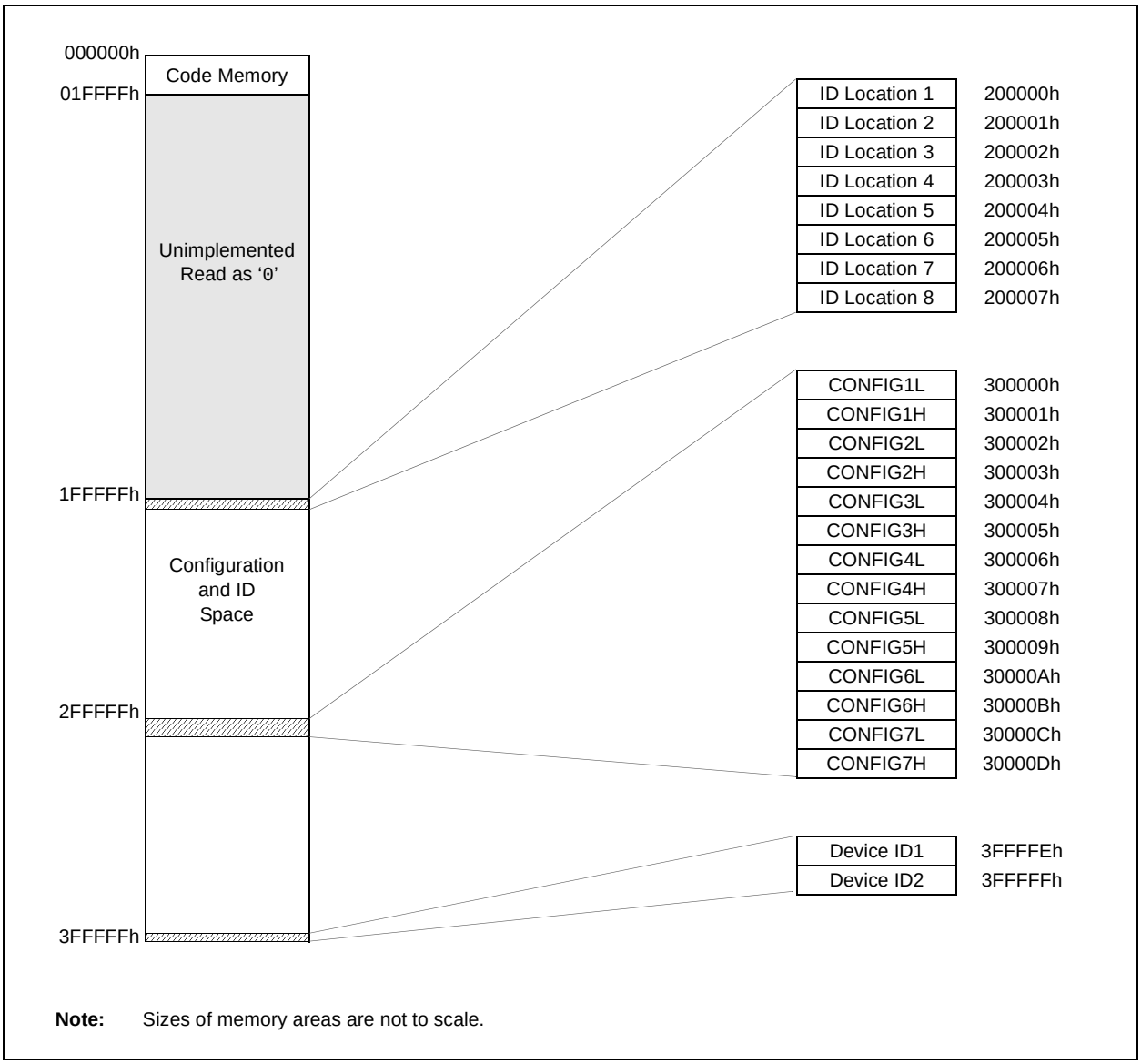
Memory in the address space, 0000000h to 3FFFFFh, is addressed via the Table Pointer register, which is comprised of three Pointer registers:

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

TBLPTRU	TBLPTRH	TBLPTRL
Addr<21:16>	Addr<15:8>	Addr<7:0>

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

FIGURE 2-8: CONFIGURATION AND ID LOCATIONS FOR PIC18FXXK80 FAMILY DEVICES

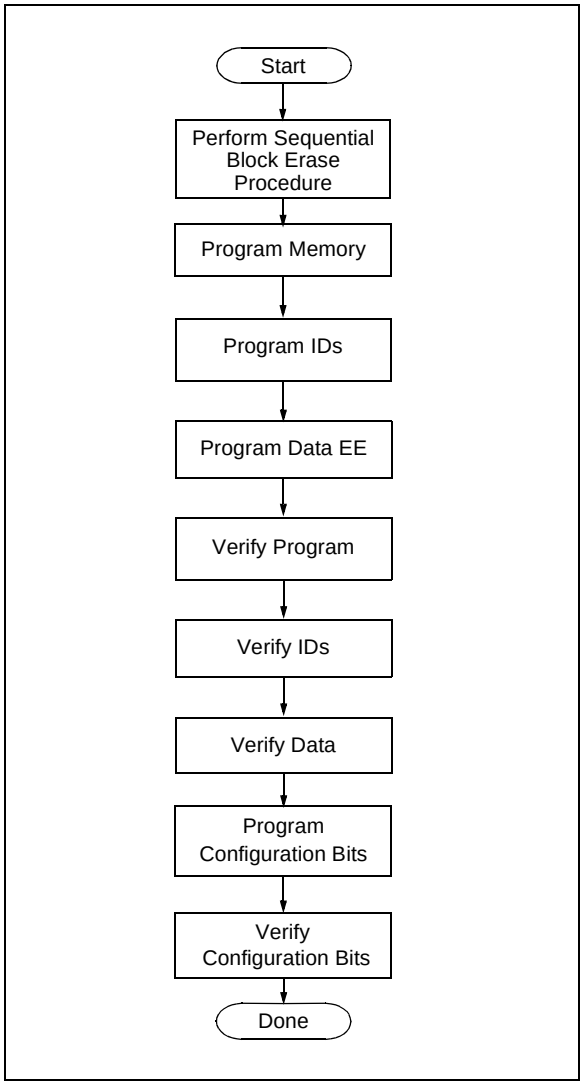


Note: Sizes of memory areas are not to scale.

2.5 High-Level Overview of the Programming Process

Figure 2-9 shows the high-level overview of the programming process. First, a Block Erase is performed for each block. 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 2-9: HIGH-LEVEL PROGRAMMING FLOW



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

As shown in Figure 2-11, entering High-Voltage ICSP Program/Verify mode requires two steps. First, voltage is applied to the MCLR pin. Second, a 32-bit key sequence is presented on PGD.

The programming voltage applied to MCLR is VIHH. VIHH must be applied to MCLR during the transfer of the key sequence. After VIHH is applied to MCLR, an interval of at least P12 must elapse before presenting the key sequence on PGD.

The key sequence is a specific 32-bit pattern,'0100 1101 0100 0011 0100 1000 0101 0000' (more easily remembered as 4D434850h in hexadecimal). The device will enter Program/Verify mode only if the sequence is valid. The Most Significant bit of the most significant nibble must be shifted in first.

Once the key sequence is complete, Program/Verify mode is entered, and the program memory can be accessed and programmed in serial fashion. While in the Program/Verify mode, all unused I/Os are placed in the high-impedance state.

Exiting Program/Verify mode is done by removing VIHH from MCLR, as shown in Figure 2-13. The only requirement for exit is that an interval, P16, should elapse between the last clock and the program signals on PGC and PGD before removing VIHH.

FIGURE 2-10: ENTERING LOW-VOLTAGE PROGRAM/VERIFY MODE

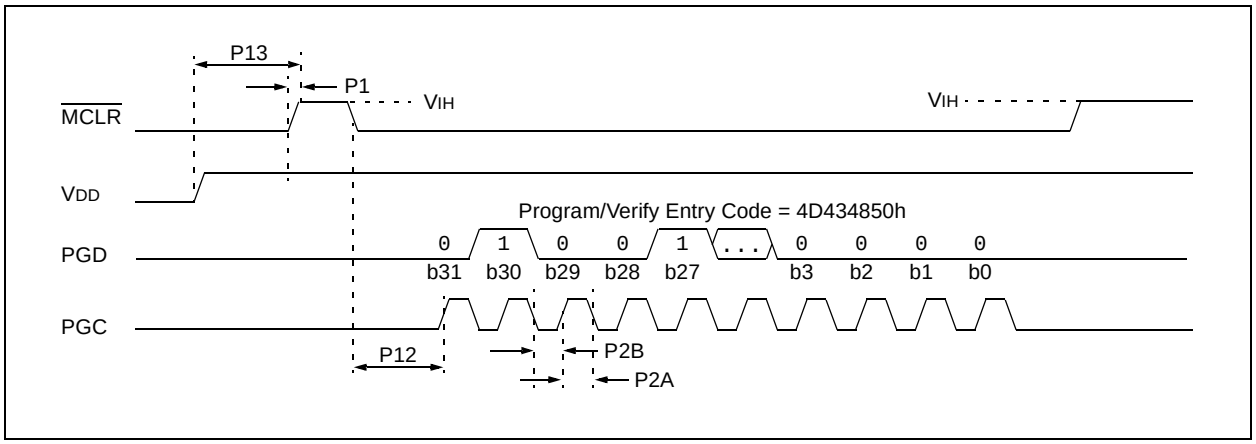


FIGURE 2-11: ENTERING HIGH-VOLTAGE PROGRAM/VERIFY MODE

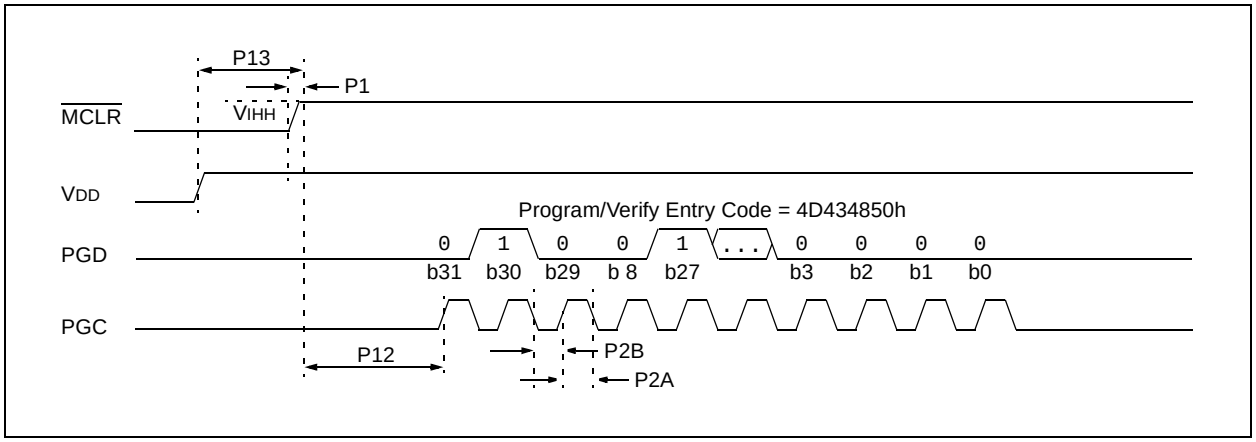


FIGURE 2-12: EXITING LOW-VOLTAGE PROGRAM/VERIFY MODE

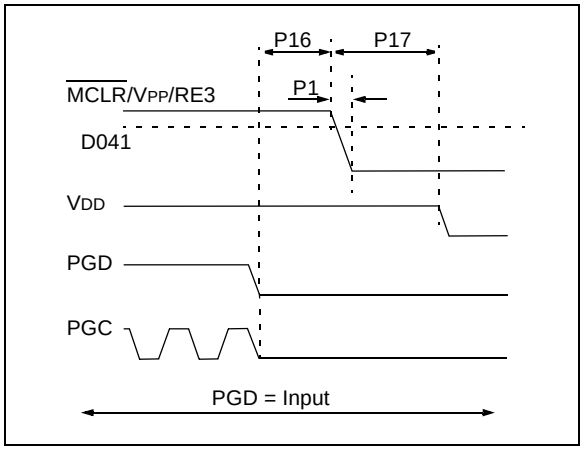
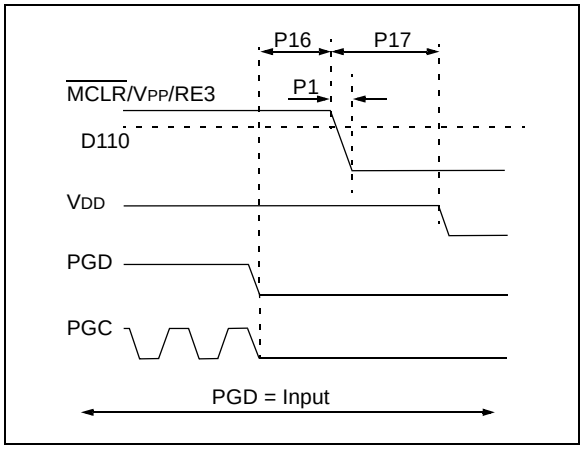


FIGURE 2-13: EXITING HIGH-VOLTAGE PROGRAM/VERIFY MODE



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

As shown in Figure 2-10, entering Low-Voltage ICSP Program/Verify mode requires three steps:

1. The $\overline{MCLR}$ pin is grounded.
2. A 32-bit key sequence is presented on PGD.
3. The $\overline{MCLR}$ pin is brought to VDD

The $\overline{MCLR}$ pin must be grounded during the transfer of the key sequence. After $\overline{MCLR}$ is grounded, an interval of at least P12 must elapse before presenting the key sequence on PGD. The key sequence is a specific 32-bit pattern,'0100 1101 0100 0011 0100 1000 0101 0000' (more easily remembered as 4D434850h in hexadecimal). The device will enter Program/Verify mode only if the sequence is valid. The Most Significant bit of the most significant nibble must be shifted in first.

Once the key sequence is complete, VIH, or usually VDD, must be applied to $\overline{MCLR}$ and held at that level for as long as Program/Verify mode is to be maintained. There is no minimum time requirement before presenting data on PGD. On successful entry, the program memory can be accessed and programmed in serial fashion. While in the Program/Verify mode, all unused I/Os are placed in the high-impedance state.

Exiting Program/Verify mode is done by grounding the $\overline{MCLR}$ again, as shown in Figure 2-12. The only requirement for exit is that an interval, P16, should elapse between the last clock, and the program signals on PGC and PGD before grounding $\overline{MCLR}$.

2.8 Serial Program/Verify Operation

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

2.8.1 4-BIT COMMANDS

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

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 and data are shown Most Significant bit (MSb) first. The command operand, or "Data Payload", is shown as <LSB><MSB>. Figure 2-14 demonstrates how to serially present a 20-bit command/operand to the device.

2.8.2 CORE INSTRUCTION

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

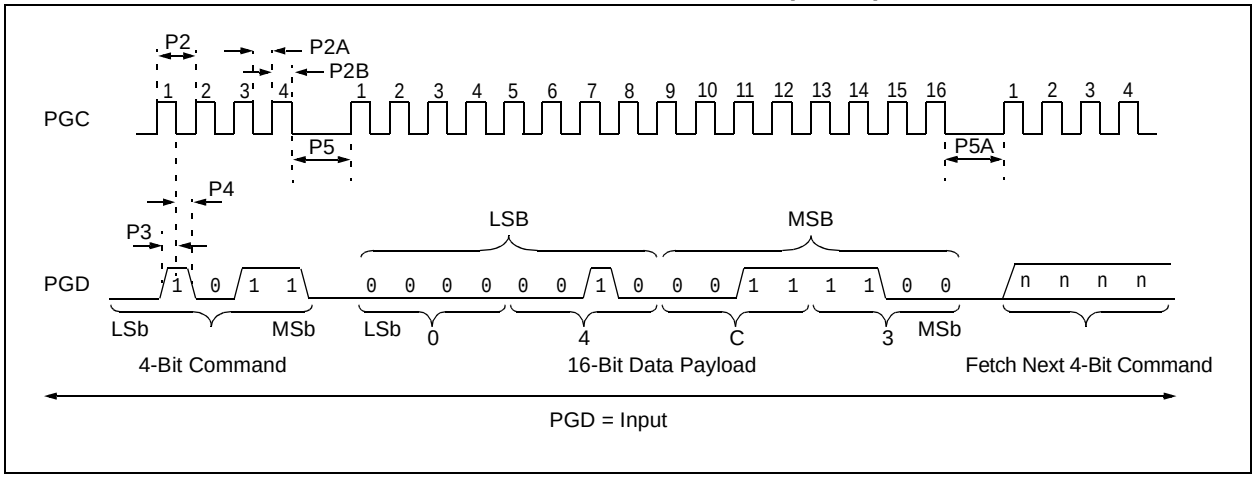
TABLE 2-3: COMMANDS FOR PROGRAMMING

Description	4-Bit Command
Core Instruction (shift in 16-bit instruction)	0000
Shift out TABLAT Register	0010
Table Read	1000
Table Read, Post-Increment	1001
Table Read, Post-Decrement	1010
Table Read, Pre-Increment	1011
Table Write	1100
Table Write, Post-Increment by 2	1101
Table Write, Start Programming, Post-Increment by 2	1110
Table Write, Start Programming	1111

TABLE 2-4: SAMPLE COMMAND SEQUENCE

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

FIGURE 2-14: TABLE WRITE, POST-INCREMENT TIMING ('1101')



3.0 DEVICE PROGRAMMING

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

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

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 or write sequence is initiated by setting the WR bit (EECON1<1> = 1). It is strongly recommended that the WREN bit only be set immediately prior to a program or erase.

REGISTER 3-1: EECON1 REGISTER

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	—	FREE	WRERR(1)	WREN	WR	RD
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit		
S = Bit can be set by software, but not cleared		U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7 **EEPGD:** Flash Program or Data EEPROM Memory Select bit
- 1 = Access Flash program memory
- 0 = Access data EEPROM memory

bit 6 **CFGS:** Flash Program/Data EEPROM or Configuration Select bit
- 1 = Access Configuration registers
- 0 = Access Flash program or data EEPROM memory

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **FREE:** Flash Row Erase Enable bit
- 1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation)
- 0 = Perform write-only

bit 3 **WRERR:** Flash Program/Data EEPROM Error Flag bit(1)
- 1 = A write operation is prematurely terminated (any Reset during self-timed programming in normal operation or an improper write attempt)
- 0 = The write operation completed

bit 2 **WREN:** Flash Program/Data EEPROM Write Enable bit
- 1 = Allows write cycles to Flash program/data EEPROM
- 0 = Inhibits write cycles to Flash program/data EEPROM

bit 1 **WR:** Write Control bit
- 1 = Initiates a data EEPROM erase/write cycle or a program memory erase/write cycle (The operation is self-timed and the bit is cleared by hardware once the write is complete. The WR bit can only be set (not cleared) in software.)
- 0 = Write cycle to the EEPROM is complete

bit 0 **RD:** Read Control bit
- 1 = Initiates an EEPROM read (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software. The RD bit cannot be set when EEPGD = 1 or CFGS = 1.)
- 0 = Does not initiate an EEPROM read

Note 1: When a WRERR occurs, the EEPGD and CFGS bits are not cleared. This allows tracing of the error condition.

3.1 ICSP Erase

3.1.1 ICSP BLOCK ERASE

Erasing code or data EEPROM is accomplished by configuring three Block Erase Control registers, located at 3C0004h through 3C0006h. Code memory can only be erased, portions at a time. In order to erase the entire device, every block must be erased sequentially. Block Erase operations will also clear any code-protect settings associated with the memory block being erased. Erase options are detailed in Table 3-1. Data EEPROM is erased at the same time as all Block Erase commands. In order to erase data EEPROM by itself, the first code sequence in Table 3-1 must be used. If the entire device is being erased, this code is not necessary.

TABLE 3-1: BLOCK ERASE OPERATIONS

Description	Data (3C0006h:3C0004h)
Erase Data EEPROM	800004h
Erase Boot Block	800005h
Erase Config Bits	800002h
Erase Code EEPROM Block 0	800104h
Erase Code EEPROM Block 1	800204h
Erase Code EEPROM Block 2	800404h
Erase Code EEPROM Block 3	800804h

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

The code sequence to erase the entire device is shown in Table 3-2 through Table 3-7 and the flowchart is shown in Figure 3-1. The code sequence to just erase data EEPROM is shown in Table 3-8.

Note:	A Block Erase is the only way to reprogram code-protect bits from an ON state to an OFF state.

TABLE 3-2: ERASE BLOCK 0

4-Bit Command	Data Payload	Core Instruction
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 04	MOVLW 04h
0000	6E F6	MOVWF TBLPTRL
1100	04 04	Write 04h to 3C0004h
0000	0E 05	MOVLW 05h
0000	6E F6	MOVWF TBLPTRL
1100	01 01	Write 01h to 3C0005h
0000	0E 06	MOVLW 06h
0000	6E F6	MOVWF TBLPTRL
1100	80 80	Write 80h to 3C0006h to erase block 0
0000	00 00	NOP
0000	00 00	Hold PGD low until erase completes

TABLE 3-3: ERASE BLOCK 1

4-Bit Command	Data Payload	Core Instruction
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 04	MOVLW 04h
0000	6E F6	MOVWF TBLPTRL
1100	04 04	Write 04h to 3C0004h
0000	0E 05	MOVLW 05h
0000	6E F6	MOVWF TBLPTRL
1100	02 02	Write 02h to 3C0005h
0000	0E 06	MOVLW 06h
0000	6E F6	MOVWF TBLPTRL
1100	80 80	Write 80h to 3c0006h to erase block 1
0000		NOP
0000	00 00	Hold PGD low until
	00 00	erase completes

TABLE 3-4: ERASE BLOCK 2

4-Bit Command	Data Payload	Core Instruction
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 04	MOVLW 04h
0000	6E F6	MOVWF TBLPTRL
1100	04 04	Write 04h to 3C0004h
0000	0E 05	MOVLW 05h
0000	6E F6	MOVWF TBLPTRL
1100	04 04	Write 04h to 3C0005h
0000	0E 06	MOVLW 06h
0000	6E F6	MOVWF TBLPTRL
1100	80 80	Write 80h to 3C0006h to erase block 2
		NOP
0000	00 00	Hold PGD low until
0000	00 00	erase completes

TABLE 3-5: ERASE BLOCK 3

4-Bit Command	Data Payload	Core Instruction
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 04	MOVLW 04h
0000	6E F6	MOVWF TBLPTRL
1100	04 04	Write 04h to 3C0004h
0000	0E 05	MOVLW 05h
0000	6E F6	MOVWF TBLPTRL
1100	08 08	Write 08h to 3C0005h
0000	0E 06	MOVLW 06h
0000	6E F6	MOVWF TBLPTRL
1100	80 80	Write 80h to 3C0006h to erase block 3
0000	00 00	NOP
0000	00 00	Hold PGD low until Erase completes

TABLE 3-6: ERASE BOOT BLOCK

4-Bit Command	Data Payload	Core Instruction
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 04	MOVLW 04h
0000	6E F6	MOVWF TBLPTRL
1100	05 05	Write 05h to 3C0004h
0000	0E 05	MOVLW 05h
0000	6E F6	MOVWF TBLPTRL
1100	00 00	Write 00h to 3C0005h
0000	0E 06	MOVLW 06h
0000	6E F6	MOVWF TBLPTRL
1100	80 80	Write 80h to 3C0006h to erase boot block
0000	00 00	NOP
0000	00 00	Hold PGD low until Erase completes

TABLE 3-7: ERASE CONFIGURATION FUSES

4-Bit Command	Data Payload	Core Instruction
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 04	MOVLW 04h
0000	6E F6	MOVWF TBLPTRL
1100	02 02	Write 02h to 3C0004h
0000	0E 05	MOVLW 05h
0000	6E F6	MOVWF TBLPTRL
1100	00 00	Write 00h to 3C0005h
0000	0E 06	MOVLW 06h
0000	6E F6	MOVWF TBLPTRL
1100	80 80	Write 80h to 3C0006h to erase configuration fuses NOP
0000	00 00	Hold PGD low until
0000	00 00	Erase completes

TABLE 3-8: ERASE DATA EEPROM

4-Bit Command	Data Payload	Core Instruction
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 04	MOVLW 04h
0000	6E F6	MOVWF TBLPTRL
1100	04 04	Write 04h to 3C0004h
0000	0E 05	MOVLW 05h
0000	6E F6	MOVWF TBLPTRL
1100	00 00	Write 00h to 3C0005h
0000	0E 06	MOVLW 06h
0000	6E F6	MOVWF TBLPTRL
1100	80 80	Write 80h to 3C0006h to erase Data EEPROM NOP
0000	00 00	Hold PGD low until
0000	00 00	Erase completes

FIGURE 3-1: BLOCK ERASE FLOW

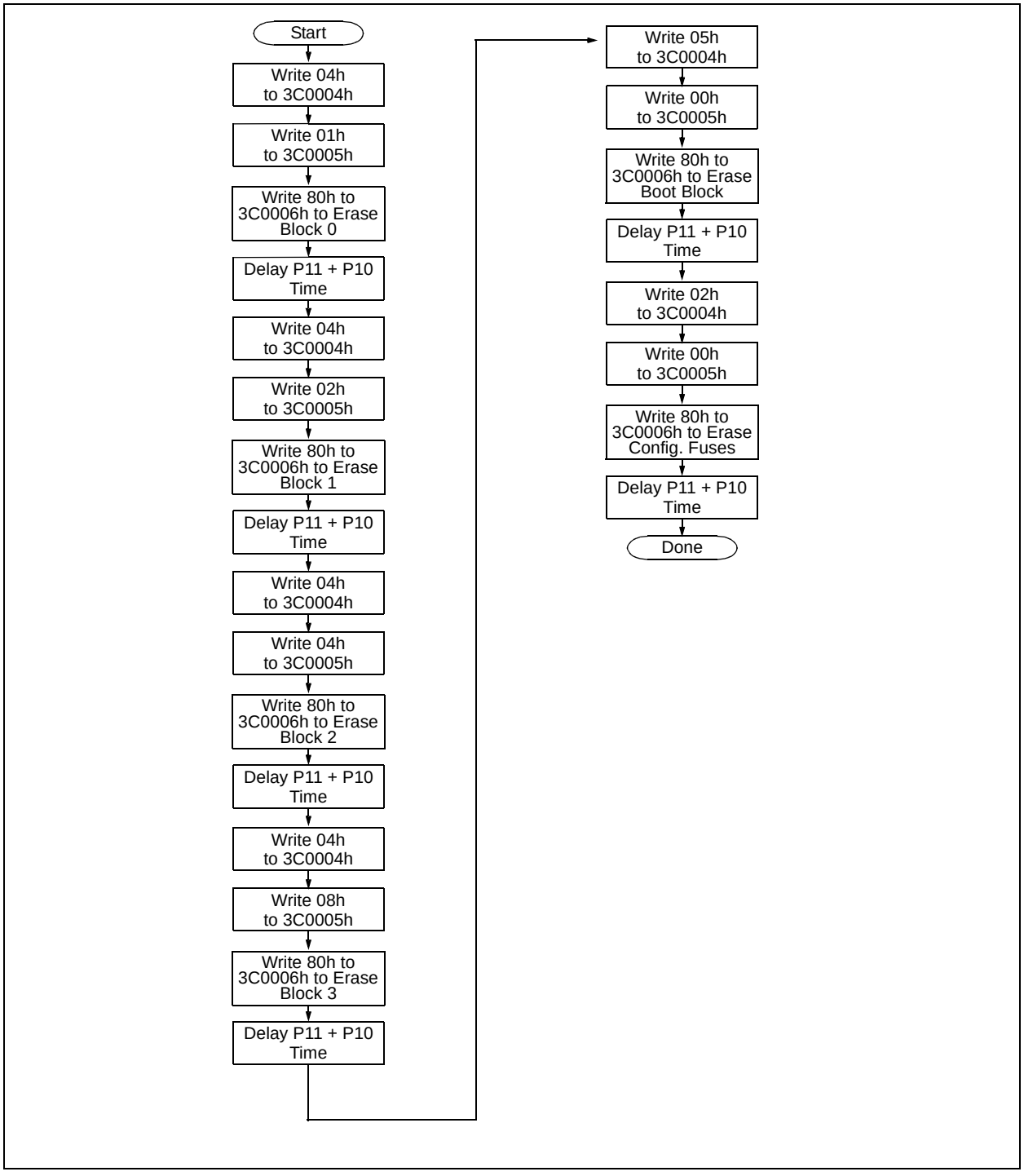
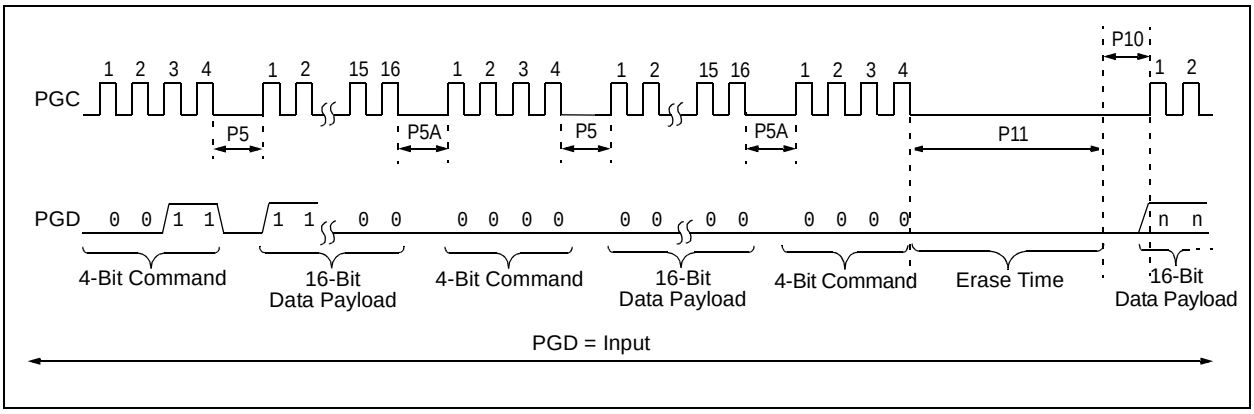


FIGURE 3-2: BLOCK ERASE TIMING



3.1.2 ICSP ROW ERASE

It is possible to erase one row (64 bytes of data) provided the block is not code or write-protected. Rows are located at static boundaries beginning at program memory address, 000000h, extending to the internal program memory limit (see **Section 2.4 "Memory Maps"**).

The Row Erase duration is externally timed and is controlled by PGC. After the WR bit in EECON1 is set, a NOP is issued, where the 4th PGC is held high for the duration of the programming time, P9.

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

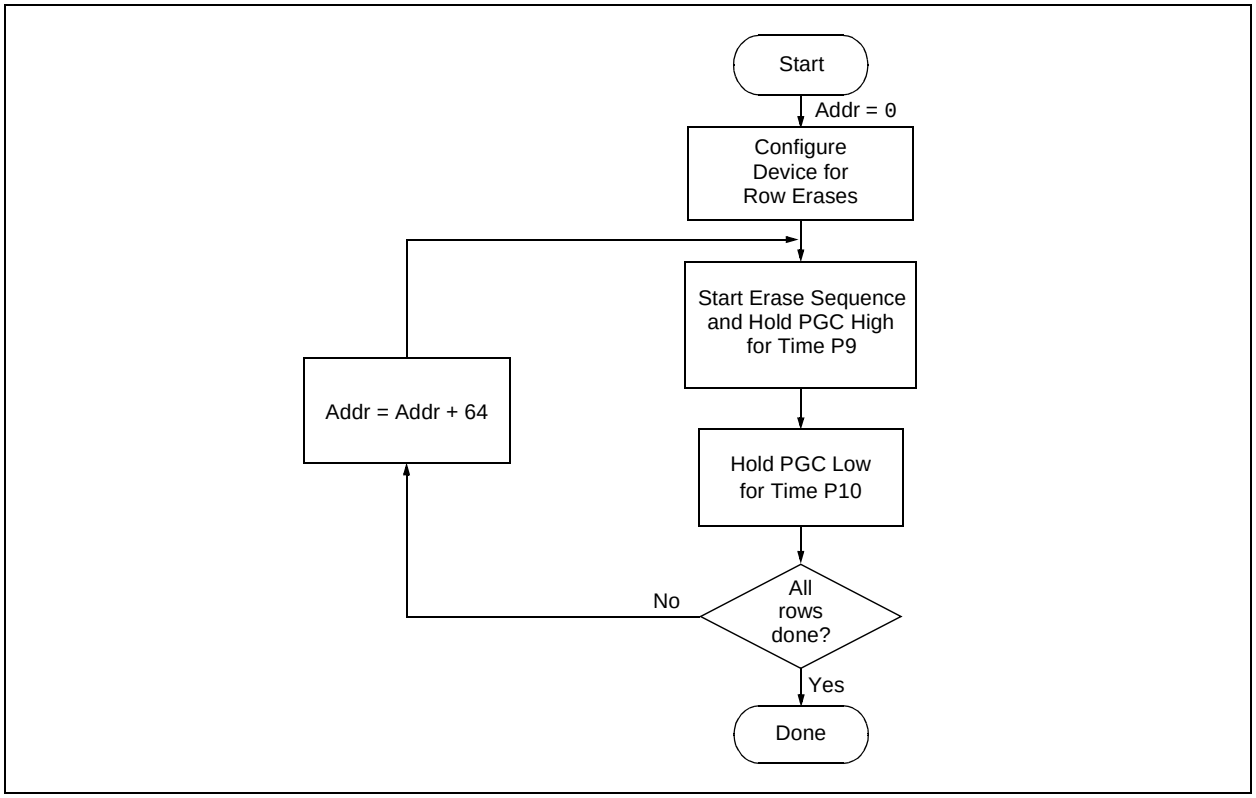
The code sequence to Row Erase a PIC18FXXK80 family device is shown in Table 3-9. The flowchart shown in Figure 3-3 depicts the logic necessary to completely erase a PIC18FXXK80 family device. The timing diagram that details the Start Programming command and Parameters P9 and P10 is shown in Figure 3-4.

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

TABLE 3-9: SINGLE ROW ERASE CODE MEMORY CODE SEQUENCE

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to code memory and enable writes.		
0000	8E 7F	BSF EECON1, EEPGD
0000	9C 7F	BCF EECON1, CFGS
0000	84 7F	BSF EECON1, WREN
Step 2: Point to first row in code memory.		
0000	6A F8	CLRF TBLPTRU
0000	6A F7	CLRF TBLPTRH
0000	6A F6	CLRF TBLPTRL
Step 3: Enable erase and erase single row.		
0000	88 7F	BSF EECON1, FREE
0000	82 7F	BSF EECON1, WR
0000	00 00	NOP – hold PGC high for time P9 and low for time P10.
Step 4: Repeat Step 3 with Address Pointer incremented by 64 until all rows are erased.		

FIGURE 3-3: SINGLE ROW ERASE CODE MEMORY FLOW



3.2 Code Memory Programming

Programming code memory is accomplished by first loading data into the write buffer and then initiating a programming sequence. The write and erase buffer sizes, shown in Table 3-10, can be mapped to any location of the same size beginning at 000000h. The actual memory write sequence takes the contents of this buffer and programs the proper amount of code memory that contains the Table Pointer.

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

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

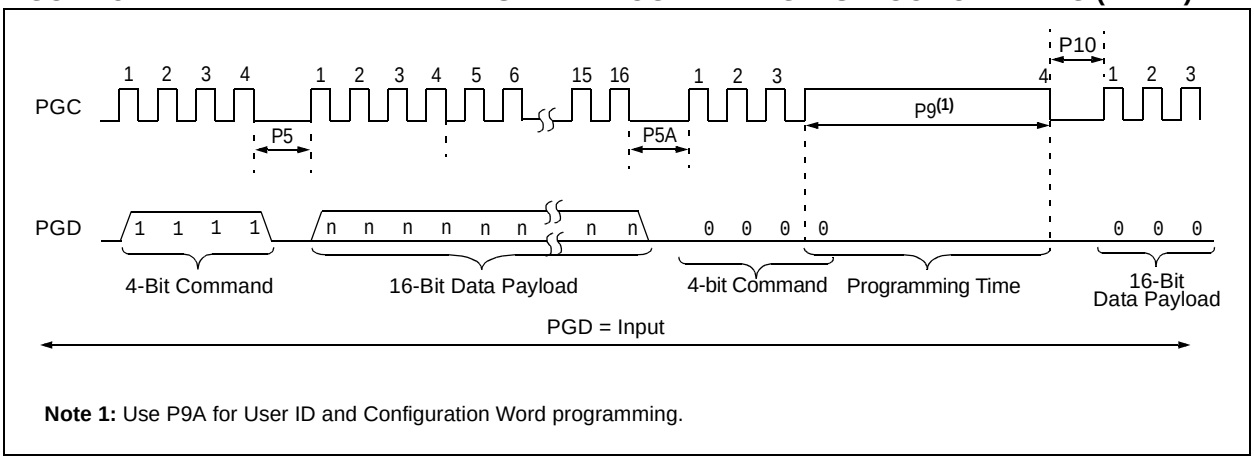
The code sequence to program a PIC18FXXK80 family device is shown in Table 3-11. The flowchart, shown in Figure 3-6, depicts the logic necessary to completely write a PIC18FXXK80 family device. The timing diagram that details the Start Programming command, and Parameters P9 and P10 is shown in Figure 3-4.

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

TABLE 3-10: WRITE AND ERASE BUFFER SIZES

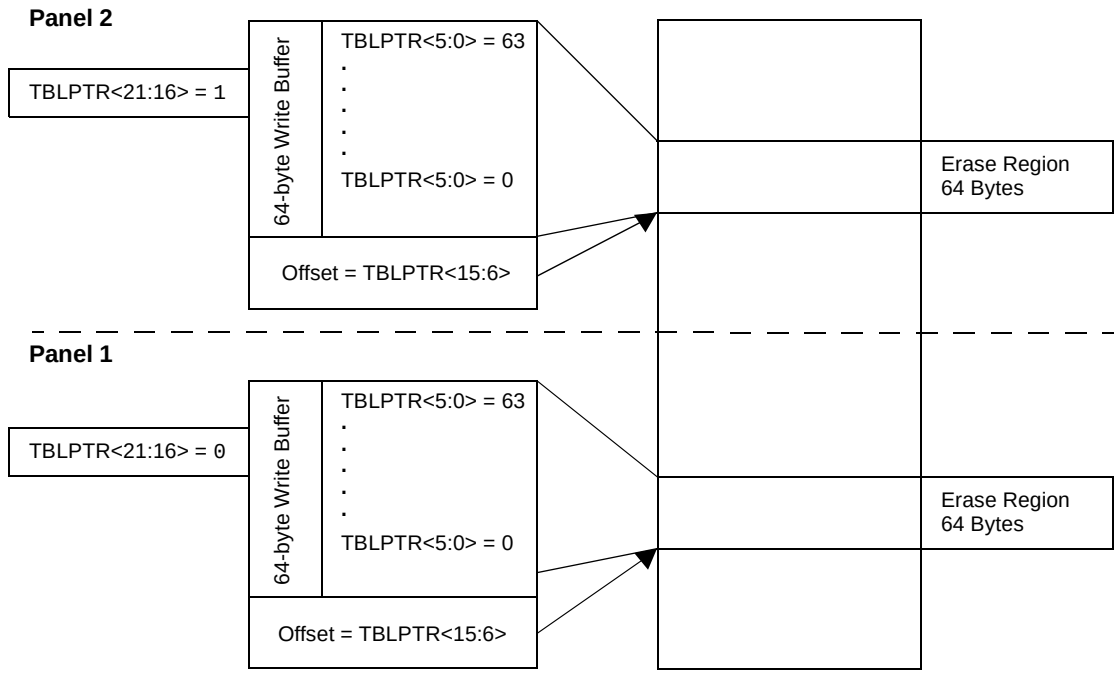
All Devices	Write Buffer Size in Bytes	Erase Buffer Size in Bytes
PIC18FXXK80	64	64

FIGURE 3-4: TABLE WRITE AND START PROGRAMMING INSTRUCTION TIMING ('1111')



Note 1: Use P9A for User ID and Configuration Word programming.

FIGURE 3-5: ERASE AND WRITE BOUNDARIES



Note: TBLPTR = TBLPTRU:TBLPTRH:TBLPTRL.

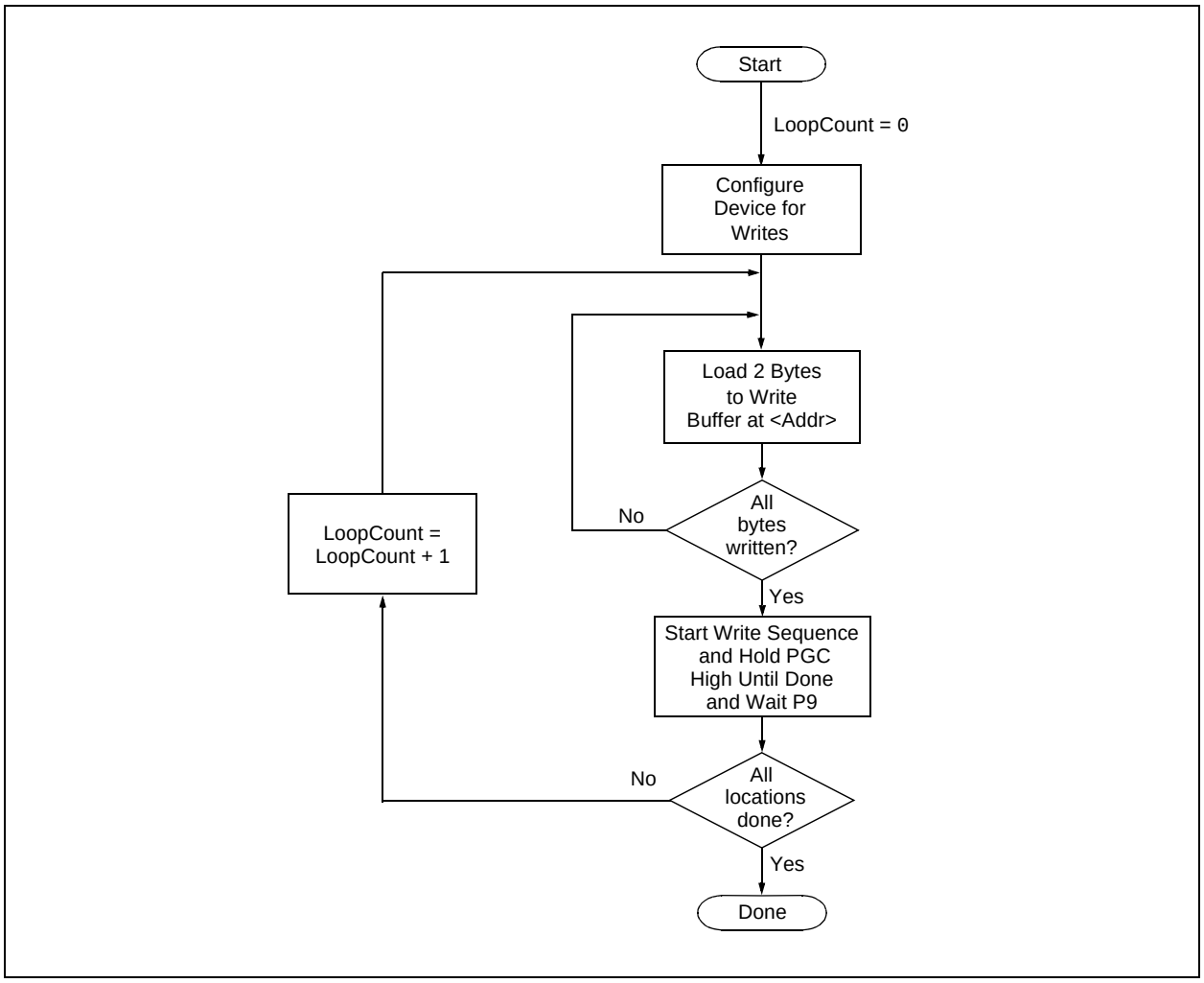
3.2.1 PROGRAMMING

A maximum of 64 bytes can be programmed into the block referenced by TBLPTR<21:6>. The panel that will be written will automatically be enabled based on the value of the Table Pointer.

TABLE 3-11: WRITE CODE MEMORY CODE SEQUENCE FOR PROGRAMMING

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to code memory and enable writes.		
0000	8E 7F	BSF EECON1, EEPGD
0000	9C 7F	BCF EECON1, CFGS
0000	84 7F	BSF EECON1, WREN
Step 2: Point to row to be written.		
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 3: Load write buffer for panel. Repeat for all but the last two bytes. Any unused locations should be filled with FFFFh.		
1101	<MSB><LSB>	Write 2 bytes and post-increment address by 2.
.	.	.
.	.	Repeat 31 times.
Step 4: Load write buffer for last two bytes.		
.	.	.
1111	<MSB><LSB>	Write 2 bytes and start programming
0000	00 00	NOP - hold SCLK high for time P9, low for time P10
To continue writing data, repeat Steps 3 and 4, where the Address Pointer is incremented by 64 at each iteration of the loop.		

FIGURE 3-6: PROGRAM CODE MEMORY FLOW



3.2.2 MODIFYING CODE MEMORY

The previous programming example assumed that the device had been erased entirely prior to programming (see **Section 3.1.1 "ICSP Block Erase"**). It may be the case, however, that the user wishes to modify only a section of an already programmed device.

The appropriate number of bytes required for the erase buffer must be read out of code memory (as described in **Section 4.2 "Verify Code Memory and ID Locations"**) and buffered. Modifications can be made on this buffer. Then, the block of code memory that was read out must be erased and rewritten with the modified data (see **Section 3.2.1 "Programming"**).

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

TABLE 3-12: MODIFYING CODE MEMORY

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to code memory.		
Step 2: Read and modify code memory (see **Section 4.1 "Read Code Memory, ID Locations and Configuration Bits"**).		
0000	8E 7F	BSF EECON1, EEPGD
0000	9C 7F	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 set up an erase.		
0000	84 7F	BSF EECON1, WREN
0000	88 7F	BSF EECON1, FREE
Step 5: Initiate erase.		
0000	82 7F	BSF EECON1, WR
0000	00 00	NOP - hold PGC high for time P9 and low for time P10.
Step 6: Direct access to configuration memory.		
0000	8E 7F	BSF EECON1, EEPGD
0000	8C 7F	BSF EECON1, CFGS
0000	84 7F	BSF EECON1, WREN
Step 7: Direct access to code memory and enable writes.		
0000	8E 7F	BSF EECON1, EEPGD
0000	9C 7F	BCF EECON1, CFGS
Step 8: Load write buffer. The correct bytes will be selected based on the Table Pointer.		
0000	0E <Addr[21:16]>	MOVLW <Addr[21:16]>
0000	6E F8	MOVWF TBLPTRU
0000	0E <Addr[8:15]>	MOVLW <Addr[8:15]>
0000	6E F7	MOVWF TBLPTRH
0000	0E <Addr[7:0]>	MOVLW <Addr[7:0]>
0000	6E F6	MOVWF TBLPTRL
1101	<MSB><LSB>	Write 2 bytes and post-increment address by 2.
.	.	
.	.	Repeat 31 times
.	.	
1111	<MSB><LSB>	Write 2 bytes and start programming.
0000	00 00	NOP - hold PGC high for time P9 and low for time P10.
To continue modifying data, repeat Steps 2 through 8, where the Address Pointer is incremented by 64 bytes at each iteration of the loop.		
Step 9: Disable writes.		
0000	94 7F	BCF EECON1, WREN

3.3 Data EEPROM Programming

Data EEPROM is accessed, one byte at a time, via an Address Pointer (register pair, EEADRH:EEADR) and a Data Latch (EEDATA). Data EEPROM is written by loading EEADRH:EEADR with the desired memory location, EEDATA with the data to be written and initiating a memory write by appropriately configuring the EECON1 register (Register 3-1). A byte write automatically erases the location and writes the new data (erase-before-write).

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

The write begins on the falling edge of the 4th PGC after the WR bit is set. It ends when the WR bit is cleared by hardware.

After the programming sequence terminates, PGC 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

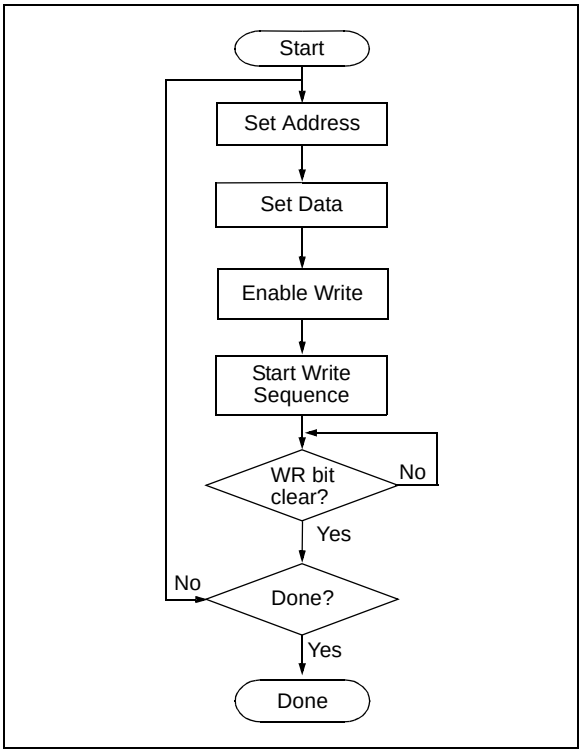


FIGURE 3-8: DATA EEPROM WRITE TIMING

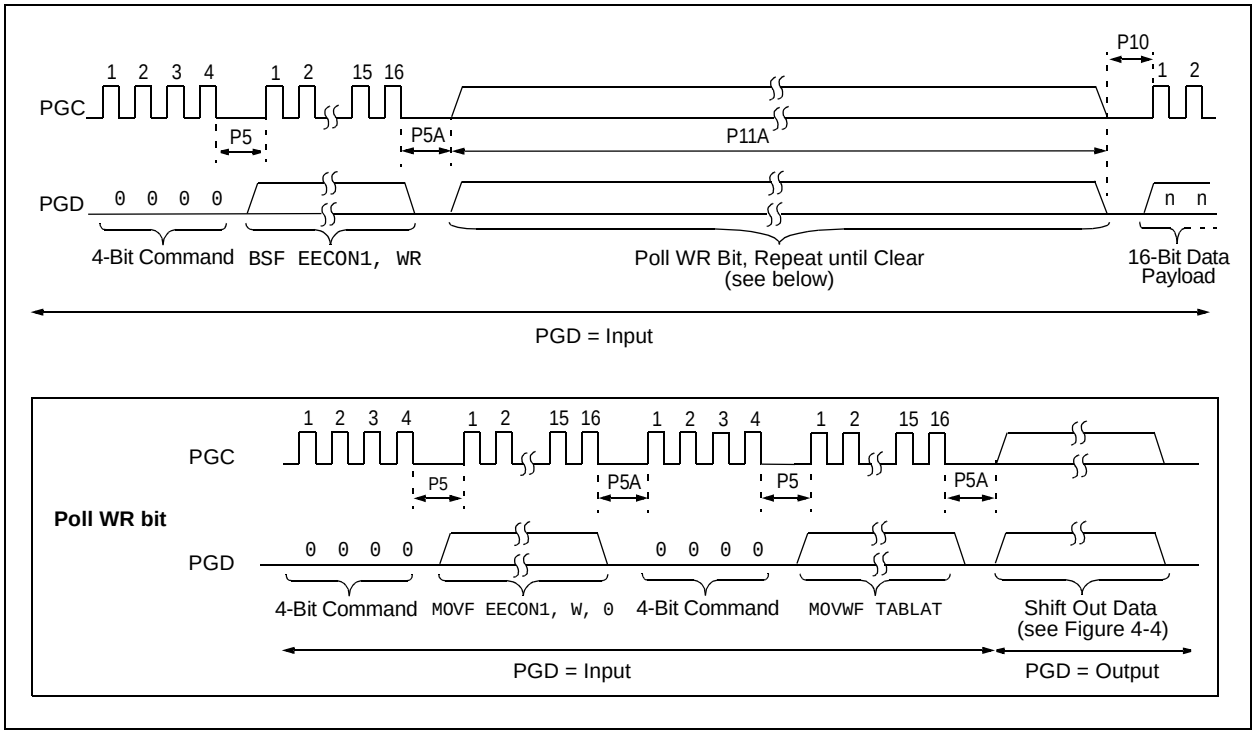


TABLE 3-13: PROGRAMMING DATA MEMORY

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to data EEPROM.		
0000	9E 7F	BCF EECON1, EEPGD
0000	9C 7F	BCF EECON1, CFGS
Step 2: Set the data EEPROM Address Pointer.		
0000	0E <Addr>	MOVLW <Addr>
0000	6E 74	MOVWF EEADR
0000	OE <AddrH>	MOVLW <AddrH>
0000	6E 75	MOVWF EEADRH
Step 3: Load the data to be written.		
0000	0E <Data>	MOVLW <Data>
0000	6E 73	MOVWF EEDATA
Step 4: Enable memory writes.		
0000	84 7F	BSF EECON1, WREN
Step 5: Initiate write.		
0000	82 7F	BSF EECON1, WR
Step 6: Poll WR bit, repeat until the bit is clear.		
0000	50 7F	MOVF EECON1, W, 0
0000	6E F5	MOVWF TABLAT
0000	00 00	NOP
0010	<MSB><LSB>	Shift out data(1)
Step 7: Hold PGC low for time, P10.		
Step 8: Disable writes.		
0000	94 7F	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. The ID registers are mapped in addresses, 200000h through 200007h. These locations read out normally even after code protection.

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

Table 3-14 demonstrates the code sequence required to write the ID locations.

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 being modified.

TABLE 3-14: WRITE ID SEQUENCE

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to code memory and enable writes.		
0000	8E 7F	BSF EECON1, EEPGD
0000	9C 7F	BCF EECON1, CFGS
Step 2: Load write buffer with 8 bytes and write.		
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	<MSB><LSB>	Write 2 bytes and post-increment address by 2.
1101	<MSB><LSB>	Write 2 bytes and post-increment address by 2.
1101	<MSB><LSB>	Write 2 bytes and post-increment address by 2.
1111	<MSB><LSB>	Write 2 bytes and start programming.
0000	00 00	NOP - hold PGC high for time P9 and low for time P10.

3.5 Boot Block Programming

The code sequence detailed in Table 3-11 should be used, except that the address used in “Step 2” will be in the range of 000000h to 0007FFh, or 000000h to 000FFFh, as defined by the BBSIZ bit in the CONFIG4L register (see Table 5-1).

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 3-15.

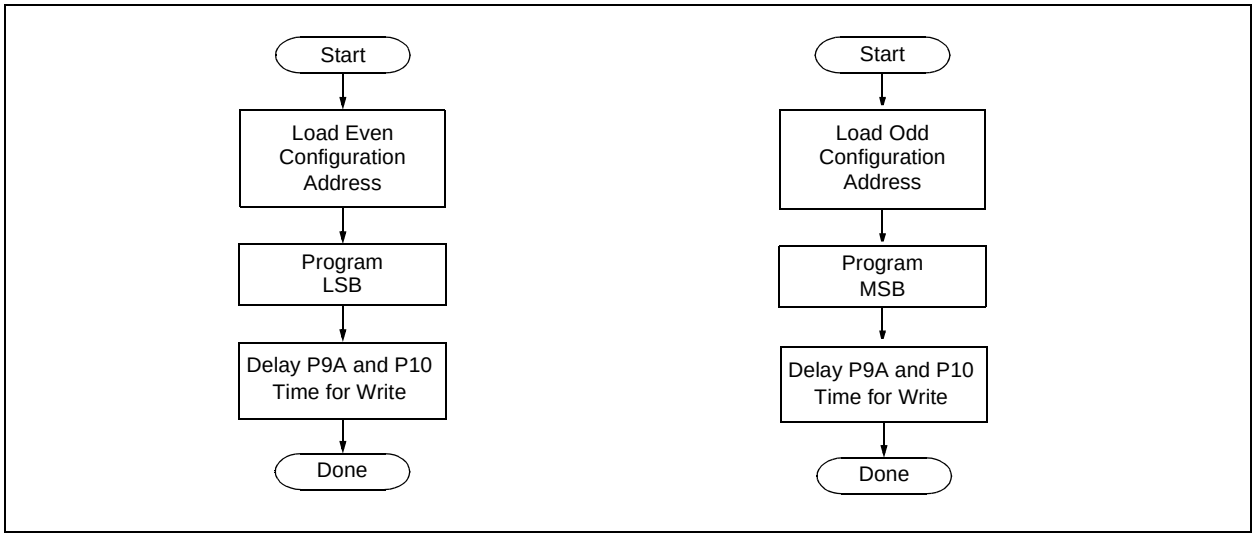
Note:	The address must be explicitly written for each byte programmed. The addresses can not be incremented in this mode.

TABLE 3-15: SET ADDRESS POINTER TO CONFIGURATION LOCATION

4-Bit Command	Data Payload	Core Instruction
Step 1: Enable writes and direct access to configuration memory.		
0000	8E 7F	BSF EECON1, EEPGD
0000	8C 7F	BSF EECON1, CFGS
Step 2: Set Table Pointer for configuration byte to be written; write even/odd addresses.(1)		
0000	0E 30	MOVLW 30h
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
1111	<MSB ignored><LSB>	Load 2 bytes and start programming.
0000	00 00	NOP - hold PGC high for time P9 and low for time P10.
0000	0E 01	MOVLW 01h
0000	6E F6	MOVWF TBLPTRL
1111	<MSB><LSB ignored>	Load 2 bytes and start programming.
0000	00 00	NOP - hold PGC high for time P9A and low for time P10.

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

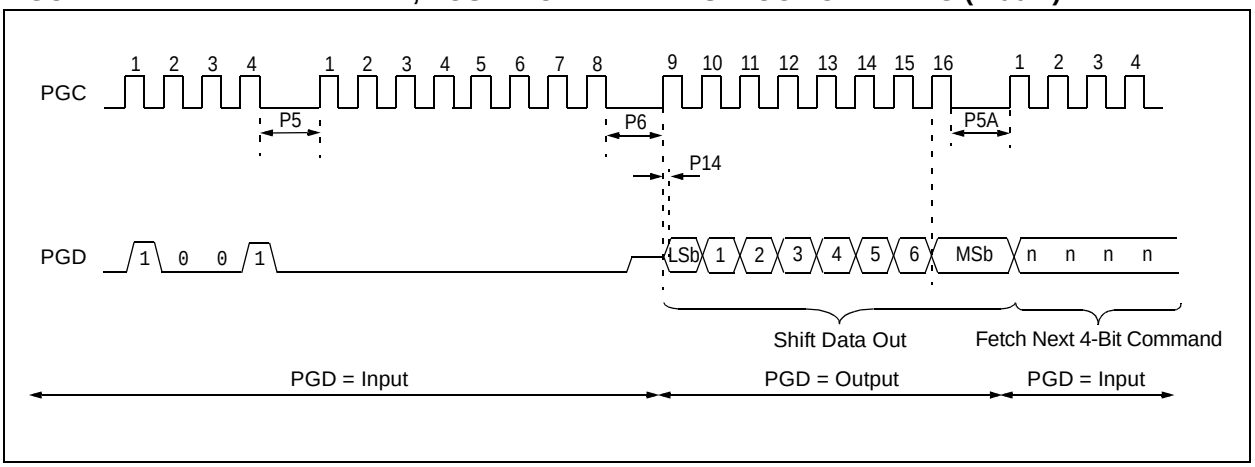
The 4-bit command is shifted in, LSb first. The read is executed during the next 8 clocks, then shifted out on PGD during the last 8 clocks, LSb to MSb. A delay of P6 must be introduced after the falling edge of the 8th PGC of the operand to allow PGD to transition from an input to an output. During this time, PGC must be held low (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 reading 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 and then shift out on PGD, LSb to MSb.		
1001	00 00	TBLRD *+

FIGURE 4-1: TABLE READ, POST-INCREMENT INSTRUCTION TIMING ('1001')

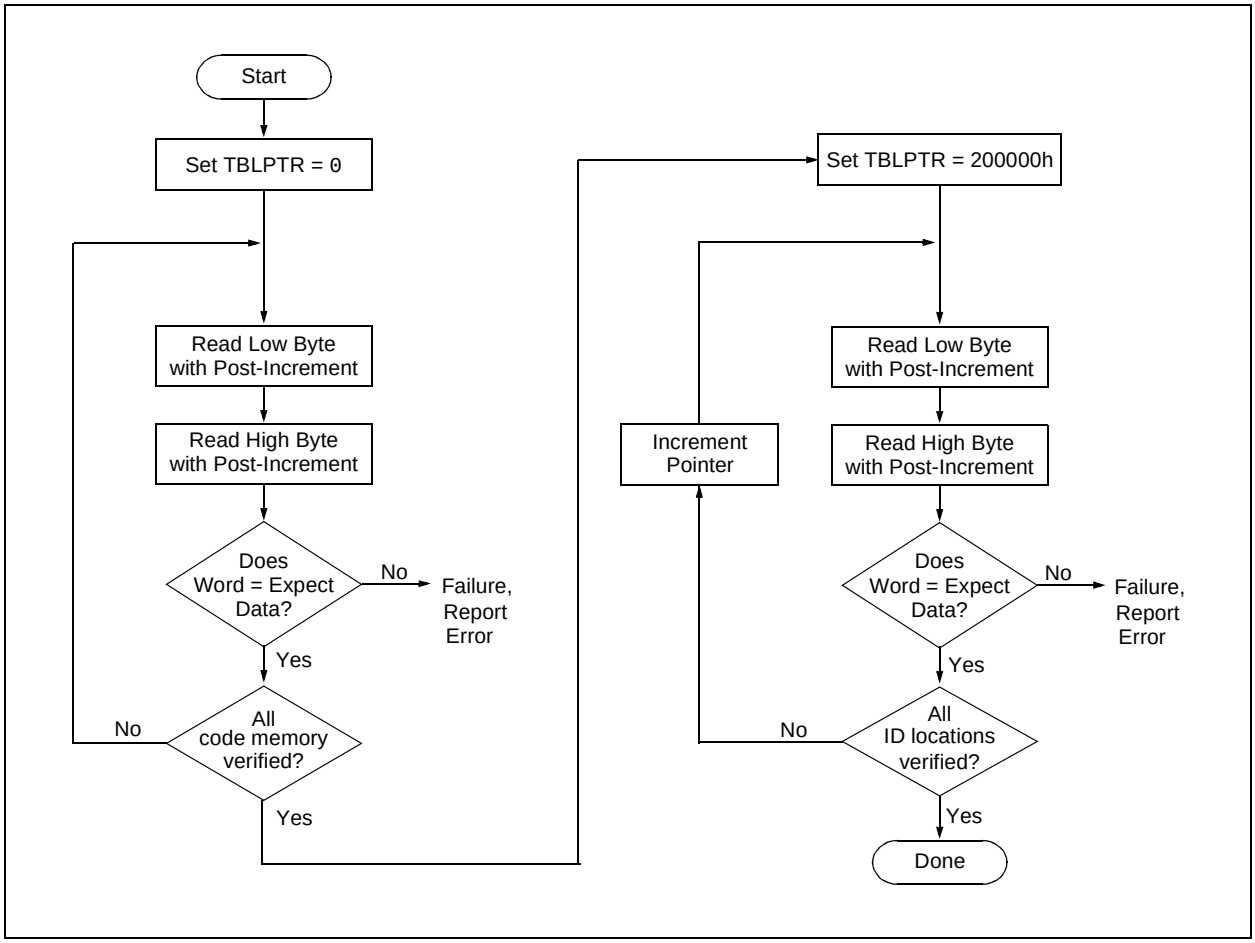


4.2 Verify Code Memory and ID Locations

The verify step involves reading back the code memory space and comparing it against the copy held in the programmer's buffer. Memory reads occur a single byte at a time, so two bytes must be read to compare against the word in the programmer's buffer. Refer to **Section 4.1 "Read Code Memory, ID Locations and Configuration Bits"** 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 128-Kbyte device, for example, a post-increment read of address, 1FFFFh, will wrap the Table Pointer back to 000000h, rather than point to the unimplemented address, 020000h.

FIGURE 4-2: VERIFY CODE MEMORY FLOW



4.3 Verify Configuration Bits

A configuration address may be read and output on PGD via the 4-bit command, '1001'. Configuration data is read and written in a byte-wise fashion, so it is not necessary to merge two bytes into a word prior to a compare. The result may then be immediately compared to the appropriate configuration data in the programmer's memory for verification. Refer to **Section 4.1 "Read Code Memory, ID Locations and Configuration Bits"** 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, EEADRH:EEADR) and a Data Latch (EEDATA). Data EEPROM is read by loading EEADRH:EEADR with the desired memory location and initiating a memory read by appropriately configuring the EECON1 register (Register 3-1). The data will be loaded into EEDATA, where it may be serially output on PGD via the 4-bit command, '0010' (Shift Out Data Holding register). A delay of P6 must be introduced after the falling edge of the 8th PGC of the operand to allow PGD to transition from an input to an output. During this time, PGC must be held low (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

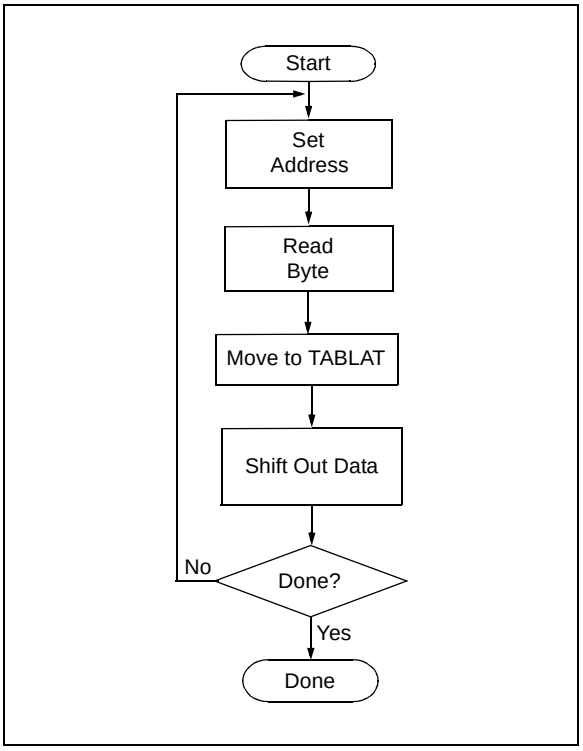
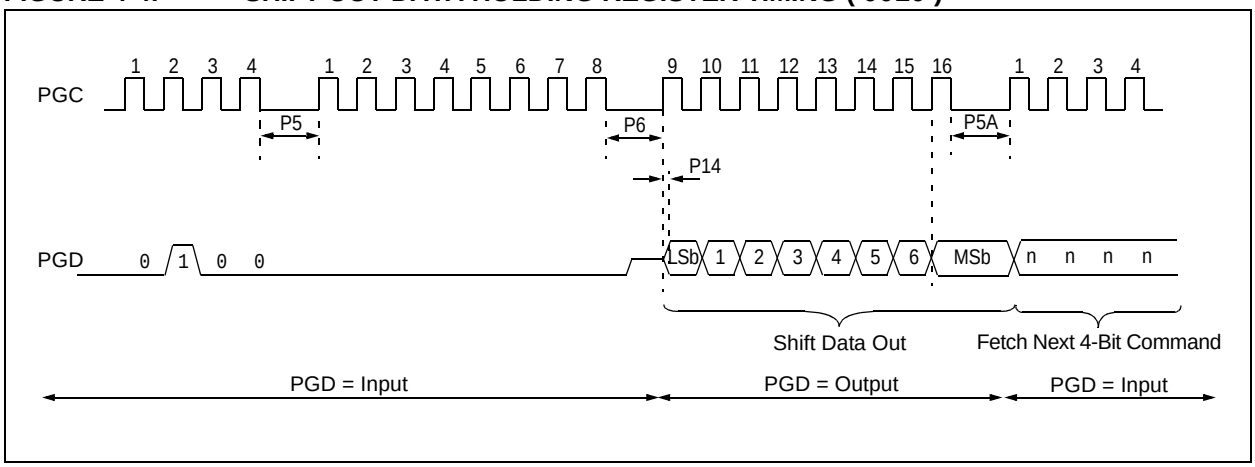


TABLE 4-2: READ DATA EEPROM MEMORY

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to data EEPROM.		
0000 0000	9E 7F 9C 7F	BCF EECON1, EEPGD BCF EECON1, CFGS
Step 2: Set the data EEPROM Address Pointer.		
0000 0000 0000 0000	0E <Addr> 6E 74 OE <AddrH> 6E 75	MOVLW <Addr> MOVWF EEADR MOVLW <AddrH> MOVWF EEADRH
Step 3: Initiate a memory read.		
0000	80 7F	BSF EECON1, RD
Step 4: Load data into the Serial Data Holding register.		
0000 0000 0000 0010	50 73 6E F5 00 00 <MSB><LSB>	MOVF EEDATA, W, 0 MOVWF TABLAT NOP Shift Out Data(1)

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

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 PGD via the 4-bit command, '0010' (TABLAT register). The result may then be immediately compared to the appropriate data in the programmer's memory for verification. Refer to **Section 4.4 "Read Data EEPROM Memory"** for implementation details of reading data EEPROM.

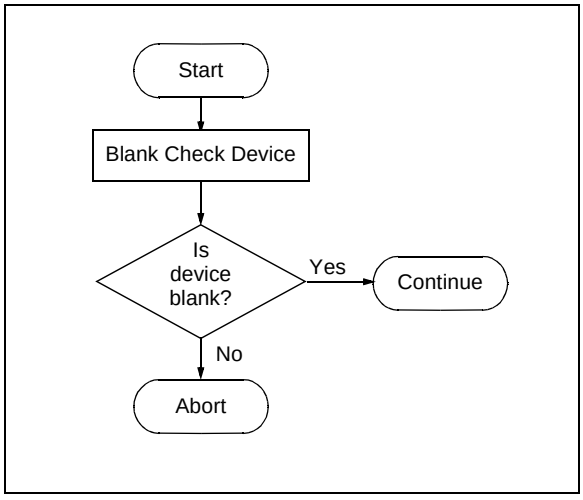
4.6 Blank Check

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 5-1 for blank configuration expect data for the various PIC18FXXK80 family devices.

Given that Blank Checking is merely code and data EEPROM verification with FFh expect data, refer to **Section 4.4 "Read Data EEPROM Memory"** and **Section 4.2 "Verify Code Memory and ID Locations"** for implementation details.

FIGURE 4-5: BLANK CHECK FLOW



5.0 CONFIGURATION WORD

The PIC18FXXK80 family of devices 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 the Configuration Words. These bits may be read out normally, even after read or code protection. See Table 5-1 for a list of Configuration bits and Device IDs, and Table 5-3 for the Configuration bit descriptions.

5.1 ID Locations

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

TABLE 5-1: 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
300000h	CONFIG1L	—	XINST	—	SOSCSEL1	SOSCSEL0	INTOSCSEL	—	$\overline{\text{RETEN}}$	-1-1 11-1
300001h	CONFIG1H	IESO	FCMEN	—	PLLCFG	FOSC3	FOSC2	FOSC1	FOSC0	00-0 1000
300002h	CONFIG2L	—	BORPW1	BORPW0	BORV1	BORV0	BOREN1	BOREN0	$\overline{\text{PWRTEN}}$	-111 1111
300003h	CONFIG2H	—	WDTPS4	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN1	WDTEN0	-111 1111
300005h	CONFIG3H	MCLRE	—	—	—	MSSPMSK	T3CKMX(1,3)	T0CKMX(1)	CANMX	1--- 1111
300006h	CONFIG4L	$\overline{\text{DEBUG}}$	—	—	BBSIZ	—	—	—	STVREN	1--1 ---1
300008h	CONFIG5L	—	—	—	—	CP3	CP2	CP1	CP0	---- 1111
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—	11-- ----
30000Ah	CONFIG6L	—	—	—	—	WRT3	WRT2	WRT1	WRT0	---- 1111
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	—	—	—	—	—	111- ----
30000Ch	CONFIG7L	—	—	—	—	EBTR3	EBTR2	EBTR1	EBTR0	---- 1111
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—	-1-- ----
3FFFFEh	DEVID1(2)	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxxx xxxx
3FFFFFh	DEVID2(2)	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	xxxx xxxx

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

Note 1: Only implemented in 64-pin devices.

2: See Register 28-13 in the *"PIC18F66K80 Family Data Sheet"* for DEVID1 values. DEVID registers are read-only and cannot be programmed by the user.

3: This bit must be maintained as '0' on 28-pin PIC18F2XK80 and 40/44-pin PIC18F4XK80 devices.

5.2 Device ID Word

The Device ID word (DEVID<2:1>) for the PIC18FXXK80 family of devices 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 protection. See Table 5-2 for a complete list of Device ID values.

FIGURE 5-1: READ DEVICE ID WORD FLOW

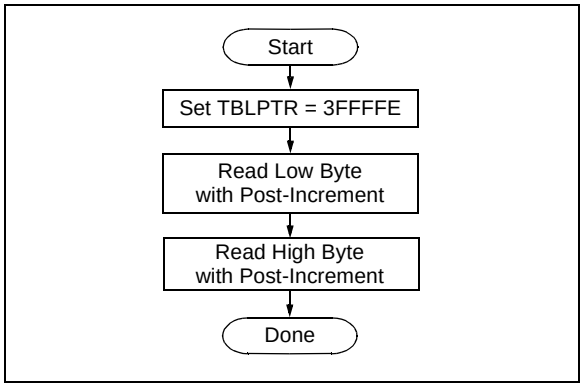


TABLE 5-2: DEVICE ID VALUE

Device	Device ID Value	
	DEVID2	DEVID1
PIC18F66K80	60h	111x xxxx
PIC18F46K80	61h	000x xxxx
PIC18F26K80	61h	001x xxxx
PIC18F65K80	61h	010x xxxx
PIC18F45K80	61h	011x xxxx
PIC18F25K80	61h	100x xxxx
PIC18LF66K80	61h	110x xxxx
PIC18LF46K80	61h	111x xxxx
PIC18LF26K80	62h	000x xxxx
PIC18LF65K80	62h	001x xxxx
PIC18LF45K80	62h	010x xxxx
PIC18LF25K80	62h	011x xxxx

Note: The ‘x’s in DEVID1 contain the device revision code.

TABLE 5-3: PIC18FXXK80 FAMILY CONFIGURATION BIT DESCRIPTIONS

Bit Name	Configuration Words	Description
XINST	CONFIG1L	Extended Instruction Set Enable bit 1 = Instruction set extension and Indexed Addressing mode enabled 0 = Instruction set extension and Indexed Addressing mode disabled (Legacy mode)
SOSCSEL<1:0>	CONFIG1L	SOSC Power Selection and Mode Configuration bits 11 = High-power SOSC circuit selected 10 = Digital (SCLKI) mode 01 = Low-power SOSC circuit selected 00 = Reserved
INTOSCSEL	CONFIG1L	LF-INTOSC Low-Power Enable bit 1 = LF-INTOSC in High-Power mode during Sleep 0 = LF-INTOSC in Low-Power mode during Sleep
$\overline{\text{RETEN}}$	CONFIG1L	VREG Sleep Enable bit 1 = Ultra low-power regulator is disabled. Regulator power in Sleep mode is controlled by VREGSLP (WDTCON<7>). 0 = Ultra low-power regulator is enabled. Regulator power in Sleep mode is controlled by SRETEN (WDTCON<4>).
IESO	CONFIG1H	Internal External Switchover bit 1 = Two-Speed Start-up is enabled 0 = Two-Speed Start-up is disabled
FCMEN	CONFIG1H	Fail-Safe Clock Monitor Enable bit 1 = Fail-Safe Clock Monitor is enabled 0 = Fail-Safe Clock Monitor is disabled
PLLCFG	CONFIG1H	4 x PLL Enable bit 1 = Oscillator is multiplied by 4 0 = Oscillator is used directly
FOSC<3:0>	CONFIG1H	Oscillator Selection bits 1101 = EC1, EC oscillator **(low power, DC-160 kHz)** 1100 = EC1IO, EC oscillator with CLKOUT function on RA6 **(low power, DC-160 kHz)** 1011 = EC2, EC oscillator **(medium power, 160 kHz-16 MHz)** 1010 = EC2IO, EC oscillator with CLKOUT function on RA6 **(medium power, 160 kHz-16 MHz)** 1001 = INTIO1 internal RC oscillator with CLKOUT function on RA6 1000 = INTIO2 internal RC oscillator 0111 = RC external RC oscillator 0110 = RCIO external RC oscillator with CKLOUT function on RA6 0101 = EC3, EC oscillator **(high power, 16 MHz-64 MHz)** 0100 = EC3IO, EC oscillator with CLKOUT function on RA6 **(high power, 16 MHz-64 MHz)** 0011 = HS1, HS oscillator **(medium power, 4 MHz-16 MHz)** 0010 = HS2, HS oscillator **(high power, 16 MHz-25 MHz)** 0001 = XT oscillator 0000 = LP oscillator
BORPWR<1:0>	CONFIG2L	BORMV Power Level bits 11 = ZPBORMV instead of BORMV is selected 10 = BORMV is set to high-power level 01 = BORMV is set to medium power level 00 = BORMV is set to low-power level
BORV<1:0>	CONFIG2L	Brown-out Reset Voltage bits 11 = V_{BOR} set to 1.8V 10 = V_{BOR} set to 2.0V 01 = V_{BOR} set to 2.7V 00 = V_{BOR} set to 3.0V

Note 1: The BBSIZ bit cannot be changed once any of the following code-protect bits are enabled: CPB or CP0, WRTB or WRT0, EBTRB or EBTR0.

2: Available on PIC18F6XKXX devices only.

3: This bit must be maintained as '0' on 28-pin PIC18F2XK80 and 40-pin PIC18F4XK80 devices.

TABLE 5-3: PIC18FXXK80 FAMILY CONFIGURATION BIT DESCRIPTIONS (CONTINUED)

Bit Name	Configuration Words	Description
BOREN<1:0>	CONFIG2L	Brown-out Reset Enable bits 11 = Brown-out Reset is enabled in hardware only (SBOREN is disabled) 10 = Brown-out Reset is enabled in hardware only and disabled in Sleep mode (SBOREN is disabled) 01 = Brown-out Reset is enabled and controlled by software (SBOREN is enabled) 00 = Brown-out Reset is disabled in hardware and software
$\overline{\text{PWRTEN}}$	CONFIG2L	Power-up Timer Enable bit 1 = PWRT is disabled 0 = PWRT is enabled
WDTPS<4:0>	CONFIG2H	Watchdog Timer Postscale Select bits 10101-11111: Reserved 10100 = 1:1048576 10011 = 1:524288 10010 = 1:262144 10001 = 1:131072 10000 = 1:65536 01111 = 1:32,768 01110 = 1:16,384 01101 = 1:8,192 01100 = 1:4,096 01011 = 1:2,048 01010 = 1:1,024 01001 = 1:512 01000 = 1:256 00111 = 1:128 00110 = 1:64 00101 = 1:32 00100 = 1:16 00011 = 1:8 00010 = 1:4 00001 = 1:2 00000 = 1:1
WDTEN<1:0>	CONFIG2H	Watchdog Timer Enable bits 11 = WDT is enabled in hardware; SWDTEN bit is disabled 10 = WDT is controlled with the SWDTEN bit setting 01 = WDT is enabled only while device is active and disabled in Sleep; SWDTEN bit is disabled 00 = WDT is disabled in hardware; SWDTEN bit is disabled
MCLRE	CONFIG3H	$\overline{\text{MCLR}}$ Pin Enable bit 1 = $\overline{\text{MCLR}}$ pin is enabled, RE3 input pin is disabled 0 = RE3 input pin is enabled, $\overline{\text{MCLR}}$ pin is disabled
MSSPMSK	CONFIG3H	MSSP V3 7-Bit Address Masking Mode Enable bit 1 = 7-Bit Address Masking mode enable 0 = 5-Bit Address Masking mode enable
T3CKMX(2,3)	CONFIG3H	Timer3 Clock Input MUX bit 1 = Timer3 gets its clock input from the T1CKI input when T3CON(SOSCEN) = 0 0 = Timer3 gets its clock input from the T3CKI input when T3CON(SOSCEN) = 0
T0CKMX(2)	CONFIG3H	Timer0 Clock Input MUX bit 1 = Timer0 gets its clock input from the RB5/T0CKI pin 0 = Timer0 gets its clock input from the RG4/T0CKI pin
CANMX	CONFIG3H	ECAN MUX bit 1 = ECAN TX and RX pins are located on RB2 and RB3, respectively 0 = ECAN TX and RX pins are located on RC6 and RC7, respectively (28-pin and 44-pin packages) or on RE5 and RE4, respectively (64-pin package)

Note 1: The BBSIZ bit cannot be changed once any of the following code-protect bits are enabled: CPB or CP0, WRTB or WRT0, EBTRB or EBTR0.

2: Available on PIC18F6XKXX devices only.

3: This bit must be maintained as '0' on 28-pin PIC18F2XK80 and 40-pin PIC18F4XK80 devices.

TABLE 5-3: PIC18FXXK80 FAMILY CONFIGURATION BIT DESCRIPTIONS (CONTINUED)

Bit Name	Configuration Words	Description
$\overline{\text{DEBUG}}$	CONFIG4L	Background Debugger Enable bit 1 = Background debugger is disabled, RB6 and RB7 are configured as general purpose I/O pins 0 = Background debugger is enabled, RB6 and RB7 are dedicated to In-Circuit Debug
BBSIZ(1)	CONFIG4L	Boot Block Size Select bit 1 = 2K word Boot Block size 0 = 1K word Boot Block size
STVREN	CONFIG4L	Stack Overflow/Underflow Reset Enable bit 1 = Reset on stack overflow/underflow is enabled 0 = Reset on stack overflow/underflow is disabled
CP3	CONFIG5L	Code Protection bit (Block 3 code memory area) 1 = Block 3 is not code-protected 0 = Block 3 is code-protected
CP2	CONFIG5L	Code Protection bit (Block 2 code memory area) 1 = Block 2 is not code-protected 0 = Block 2 is code-protected
CP1	CONFIG5L	Code Protection bit (Block 1 code memory area) 1 = Block 1 is not code-protected 0 = Block 1 is code-protected
CP0	CONFIG5L	Code Protection bit (Block 0 code memory area) 1 = Block 0 is not code-protected 0 = Block 0 is code-protected
CPD	CONFIG5H	Code Protection bit (Data EEPROM) 1 = Data EEPROM is not code-protected 0 = Data EEPROM is code-protected
CPB	CONFIG5H	Code Protection bit (Boot Block memory area) 1 = Boot Block is not code-protected 0 = Boot Block is code-protected
WRT3	CONFIG6L	Write Protection bit (Block 3 code memory area) 1 = Block 3 is not write-protected 0 = Block 3 is write-protected
WRT2	CONFIG6L	Write Protection bit (Block 2 code memory area) 1 = Block 2 is not write-protected 0 = Block 2 is write-protected
WRT1	CONFIG6L	Write Protection bit (Block 1 code memory area) 1 = Block 1 is not write-protected 0 = Block 1 is write-protected
WRT0	CONFIG6L	Write Protection bit (Block 0 code memory area) 1 = Block 0 is not write-protected 0 = Block 0 is write-protected
WRTD	CONFIG6H	Write Protection bit (Data EEPROM) 1 = Data EEPROM is not write-protected 0 = Data EEPROM is write-protected
WRTB	CONFIG6H	Write Protection bit (Boot Block memory area) 1 = Boot Block is not write-protected 0 = Boot Block is write-protected
WRTC	CONFIG6H	Write Protection bit (Configuration registers) 1 = Configuration registers are not write-protected 0 = Configuration registers are write-protected
EBTR3	CONFIG7L	Table Read Protection bit (Block 3 code memory area) 1 = Block 3 is not protected from table reads executed in other blocks 0 = Block 3 is protected from table reads executed in other blocks

Note 1: The BBSIZ bit cannot be changed once any of the following code-protect bits are enabled: CPB or CP0, WRTB or WRT0, EBTRB or EBTR0.

2: Available on PIC18F6XKXX devices only.

3: This bit must be maintained as '0' on 28-pin PIC18F2XK80 and 40-pin PIC18F4XK80 devices.

TABLE 5-3: PIC18FXXK80 FAMILY CONFIGURATION BIT DESCRIPTIONS (CONTINUED)

Bit Name	Configuration Words	Description
EBTR2	CONFIG7L	Table Read Protection bit (Block 2 code memory area) 1 = Block 2 is not protected from table reads executed in other blocks 0 = Block 2 is protected from table reads executed in other blocks
EBTR1	CONFIG7L	Table Read Protection bit (Block 1 code memory area) 1 = Block 1 is not protected from table reads executed in other blocks 0 = Block 1 is protected from table reads executed in other blocks
EBTR0	CONFIG7L	Table Read Protection bit (Block 0 code memory area) 1 = Block 0 is not protected from table reads executed in other blocks 0 = Block 0 is protected from table reads executed in other blocks
EBTRB	CONFIG7H	Table Read Protection bit (Boot Block memory area) 1 = Boot Block is not protected from table reads executed in other blocks 0 = Boot Block is protected from table reads executed in other blocks
DEV<10:3>	DEVID2	Device ID bits These bits are used with the DEV<2:0> bits in the DEVID1 register to identify the part number.
DEV<2:0>	DEVID1	Device ID bits These bits are used with the DEV<10:3> bits in the DEVID2 register to identify the part number.
REV<4:0>	DEVID1	Revision ID bits These bits are used to indicate the revision of the device.

Note 1: The BBSIZ bit cannot be changed once any of the following code-protect bits are enabled: CPB or CP0, WRTB or WRT0, EBTRB or EBTR0.

2: Available on PIC18F6XKXX devices only.

3: This bit must be maintained as ‘0’ on 28-pin PIC18F2XK80 and 40-pin PIC18F4XK80 devices.

5.3 Embedding Configuration Word Information in the HEX File

To allow portability of code, a PIC18FXXK80 device 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.

5.4 Embedding Data EEPROM Information in the HEX File

To allow portability of code, a PIC18FXXK80 device 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.

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 (starting on Page 41) describes how to calculate the checksum for each device. For these examples, the ID memory has been set to ‘Use Unprotected Checksum’ in MPLAB IDE®. Please use this value to determine the value of the ‘SUM(IDs)’ term for each appropriate code-protected example.

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	0xAA at 0 and Max Address
PIC18F66K80	None	SUM(0000:0FFF) + SUM(1000:3FFF) + SUM(4000:7FFF) + SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=8F & 8F) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=C0 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40)	0x0490	0x03E6
	Boot Block 2K word	SUM(1000:3FFF) + SUM(4000:7FFF) + SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=8F & 8F) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x145D	0x1412
	Boot/Panel0/ Panel1	SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=8F & 8F) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0C & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x845A	0x840F
	All	(CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=8F & 8F) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=00 & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x044E	0x0458
PIC18F65K80	None	SUM(0000:0FFF) + SUM(1000:1FFF) + SUM(2000:3FFF) + SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=8F & 8F) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=C0 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40)	0x8490	0x83E6
	Boot Block 2K word	SUM(1000:1FFF) + SUM(2000:3FFF) + SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=8F & 8F) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x9465	0x941A
	Boot/Panel0/ Panel1	SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=8F & 8F) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0C & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0xC462	0xC417
	All	(CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=8F & 8F) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=00 & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x0456	0x0460

TABLE 5-4: CHECKSUM COMPUTATION (CONTINUED)

Device	Code-Protect	Checksum	Blank Value	0xAA at 0 and Max Address
PIC18F46K80	None	SUM(0000:0FFF) + SUM(1000:3FFF) + SUM(4000:7FFF) + SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=C0 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40)	0x048A	0x03E0
	Boot Block 2K word	SUM(1000:3FFF) + SUM(4000:7FFF) + SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x1460	0x1406
	Boot/Panel0/ Panel1	SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0C & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x845D	0x8403
	All	(CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=00 & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x0451	0x044C
PIC18F45K80	None	SUM(0000:0FFF) + SUM(1000:1FFF) + SUM(2000:3FFF) + SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=C0 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40)	0x848A	0x83E0
	Boot Block 2K word	SUM(1000:1FFF) + SUM(2000:3FFF) + SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x9468	0x940E
	Boot/Panel0/ Panel1	SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0C & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0xC465	0xC40B
	All	(CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=00 & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x0459	0x0454

TABLE 5-4: CHECKSUM COMPUTATION (CONTINUED)

Device	Code-Protect	Checksum	Blank Value	0xAA at 0 and Max Address
PIC18F26K80	None	SUM(0000:0FFF) + SUM(1000:3FFF) + SUM(4000:7FFF) + SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=C0 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40)	0x048A	0x03E0
	Boot Block 2K word	SUM(1000:3FFF) + SUM(4000:7FFF) + SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x1460	0x1406
	Boot/Panel0/ Panel1	SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0C & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x845D	0x8403
	All	(CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=00 & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x0451	0x044C
PIC18F25K80	None	SUM(0000:0FFF) + SUM(1000:1FFF) + SUM(2000:3FFF) + SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=C0 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40)	0x848A	0x83E0
	Boot Block 2K word	SUM(1000:1FFF) + SUM(2000:3FFF) + SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x9468	0x940E
	Boot/Panel0/ Panel1	SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0C & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0xC465	0xC40B
	All	(CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=00 & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x0459	0x0454

TABLE 5-4: CHECKSUM COMPUTATION (CONTINUED)

Device	Code-Protect	Checksum	Blank Value	0xAA at 0 and Max Address
PIC18LF66K80	None	SUM(0000:0FFF) + SUM(1000:3FFF) + SUM(4000:7FFF) + SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=8F & 8F) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=C0 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40)	0x0490	0x03E6
	Boot Block 2K word	SUM(1000:3FFF) + SUM(4000:7FFF) + SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=8F & 8F) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x145D	0x1412
	Boot/Panel0/ Panel1	SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=8F & 8F) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0C & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x845A	0x840F
	All	(CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=8F & 8F) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=00 & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x044E	0x0458
PIC18LF65K80	None	SUM(0000:0FFF) + SUM(1000:1FFF) + SUM(2000:3FFF) + SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=8F & 8F) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=C0 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40)	0x8490	0x83E6
	Boot Block 2K word	SUM(1000:1FFF) + SUM(2000:3FFF) + SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=8F & 8F) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x9465	0x941A
	Boot/Panel0/ Panel1	SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=8F & 8F) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0C & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0xC462	0xC417
	All	(CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=8F & 8F) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=00 & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x0456	0x0460

TABLE 5-4: CHECKSUM COMPUTATION (CONTINUED)

Device	Code-Protect	Checksum	Blank Value	0xAA at 0 and Max Address
PIC18LF46K80	None	SUM(0000:0FFF) + SUM(1000:3FFF) + SUM(4000:7FFF) + SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=C0 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40)	0x048A	0x03E0
	Boot Block 2K word	SUM(1000:3FFF) + SUM(4000:7FFF) + SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x1460	0x1406
	Boot/Panel0/ Panel1	SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0C & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x845D	0x8403
	All	(CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=00 & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x0451	0x044C
PIC18LF45K80	None	SUM(0000:0FFF) + SUM(1000:1FFF) + SUM(2000:3FFF) + SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=C0 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40)	0x848A	0x83E0
	Boot Block 2K word	SUM(1000:1FFF) + SUM(2000:3FFF) + SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x9468	0x940E
	Boot/Panel0/ Panel1	SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0C & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0xC465	0xC40B
	All	(CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=00 & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x0459	0x0454

TABLE 5-4: CHECKSUM COMPUTATION (CONTINUED)

Device	Code-Protect	Checksum	Blank Value	0xAA at 0 and Max Address
PIC18LF26K80	None	SUM(0000:0FFF) + SUM(1000:3FFF) + SUM(4000:7FFF) + SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=C0 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40)	0x048A	0x03E0
	Boot Block 2K word	SUM(1000:3FFF) + SUM(4000:7FFF) + SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x1460	0x1406
	Boot/Panel0/ Panel1	SUM(8000:BFFF) + SUM(C000:FFFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0C & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x845D	0x8403
	All	(CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=00 & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x0451	0x044C
PIC18LF25K80	None	SUM(0000:0FFF) + SUM(1000:1FFF) + SUM(2000:3FFF) + SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=C0 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40)	0x848A	0x83E0
	Boot Block 2K word	SUM(1000:1FFF) + SUM(2000:3FFF) + SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0F & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x9468	0x940E
	Boot/Panel0/ Panel1	SUM(4000:5FFF) + SUM(6000:7FFF) + (CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=0C & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0xC465	0xC40B
	All	(CONFIG1L=5D & 5D) + (CONFIG1H=08 & DF) + (CONFIG2L=7F & 7F) + (CONFIG2H=7F & 7F) + (CONFIG3L=00 & 00) + (CONFIG3H=89 & 89) + (CONFIG4L=91 & 91) + (CONFIG4H=00 & 00) + (CONFIG5L=00 & 0F) + (CONFIG5H=80 & C0) + (CONFIG6L=0F & 0F) + (CONFIG6H=E0 & E0) + (CONFIG7L=0F & 0F) + (CONFIG7H=40 & 40) + SUM(IDs)	0x0459	0x0454

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 $\overline{\text{MCLR}}$/VPP/RE3	VDD + 1.5	9	V	
D111	VDD	Supply Voltage during Programming	2.1	5.5	V	Row Erase/Write for “F” parts
			2.7	5.5	V	Block Erase operations for “F” parts
			2.1	3.6	V	Row Erase/Write for “LF” parts
			2.7	3.6	V	Block Erase operations for “LF” parts
D112	IPP	Programming Current on $\overline{\text{MCLR}}$/VPP/RE3	—	600	μA	
D113	IDDP	Supply Current during Programming	—	3.0	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 (PGD)	—	50	pF	To meet AC specifications
P1	TR	$\overline{\text{MCLR}}$/VPP/RE3 Rise Time to Enter Program/Verify mode	—	1.0	μs	**(Note 1)**
P2	TPGC	Serial Clock (PGC) Period	100	—	ns	VDD = 5.0V
			1	—	μs	VDD = 2.0V
P2A	TPGCL	Serial Clock (PGC) Low Time	40	—	ns	VDD = 5.0V
			400	—	ns	VDD = 2.0V
P2B	TPGCH	Serial Clock (PGC) 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 PGC ↓	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 PGC ↓ of Command Byte to First PGC ↑ of Read of Data Word	20	—	ns	
P9	TDLY5	PGC High Time (minimum programming time)	1	—	ms	Externally timed
P9A	TDLY5A	PGC High Time	5	—	ms	Configuration Word programming time
P10	TDLY6	PGC Low Time after Programming (high-voltage discharge time)	100	—	μs	

Note 1: Do not allow excess time when transitioning $\overline{\text{MCLR}}$ 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.

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

Standard Operating Conditions Operating Temperature: 25°C is recommended						
Param No.	**Sym**	**Characteristic**	**Min**	**Max**	**Units**	**Conditions**
P11	TDLY7	Delay to allow Self-Timed Data Write or Block Erase to Occur	5	—	ms	
P11A	TDRWT	Data Write Polling Time	4	—	ms	
P12	THLD2	Input Data Hold Time from $\overline{MCLR}$/VPP/RE3 ↑	250	—	μs	
P13	TSET2	VDD ↑ Setup Time to $\overline{MCLR}$/VPP/RE3 ↑	100	—	ns	
P14	TVALID	Data Out Valid from PGC ↑	10	—	ns	
P15	TDLY8	Delay between Last PGC ↓ and $\overline{MCLR}$/VPP/RE3 ↓	0	—	s	
P16	THLD3	$\overline{MCLR}$/VPP/RE3 ↓ to VDD ↓	—	100	ns	
P17	THLD3	$\overline{MCLR}$/VPP/RE3 ↑ to VDD ↑	—	100	ns	

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.

APPENDIX A: REVISION HISTORY

Revision A (March 2010)

Original programming specification for the PIC18FXXK80 family devices.

Revision B (January 2011)

Updated **Section 2.3 "On-Chip Voltage Regulator"** with correct capacitor information. Updated Table 5-4 and **Section 6.0 "AC/DC Characteristics Timing Requirements for Program/Verify Test Mode"**. Minor grammatical corrections made throughout text.

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

Printed on recycled paper.

ISBN: 978-1-60932-837-5

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

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

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

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

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

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

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

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

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

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

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

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

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

ASIA/PACIFIC

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

ASIA/PACIFIC

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

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

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

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

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

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

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

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

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

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

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

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

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

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

EUROPE

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

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

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

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

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

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

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

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

08/04/10