PIC18F872X FAMILY

Flash Microcontroller Programming Specification

1.0 DEVICE OVERVIEW

This document includes the programming specifications for the following devices:

- PIC18F6527
- PIC18F6622
- PIC18F6627
- PIC18F6628
- PIC18F6722
- PIC18F6723
- PIC18F8527
- PIC18F8622
- PIC18F8627
- PIC18F8628
- PIC18F8722
- PIC18F8723

2.0 PROGRAMMING OVERVIEW

The PIC18F872X family of devices can be programmed using either the high-voltage In-Circuit Serial Programming™ (ICSP™) method or the low-voltage ICSP method. Both methods can be done with the device in the users' system. The low-voltage ICSP method is slightly different than the high-voltage method and these differences are noted where applicable. This programming specification applies to the PIC18F872X family of devices in all package types.

2.1 Hardware Requirements

In High-Voltage ICSP mode, the PIC18F872X family requires two programmable power supplies; one for V_{DD} and one for $\overline{MCLR}/V_{PP}/RG5$. 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 PIC18F872X family can be programmed using a V_{DD} source in the operating range. The $\overline{MCLR}/V_{PP}/RG5$ 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 PIC18F872X family are shown in Figure 2-1 and Figure 2-2.

TABLE 2-1: PIN DESCRIPTIONS (DURING PROGRAMMING): PIC18F872X FAMILY

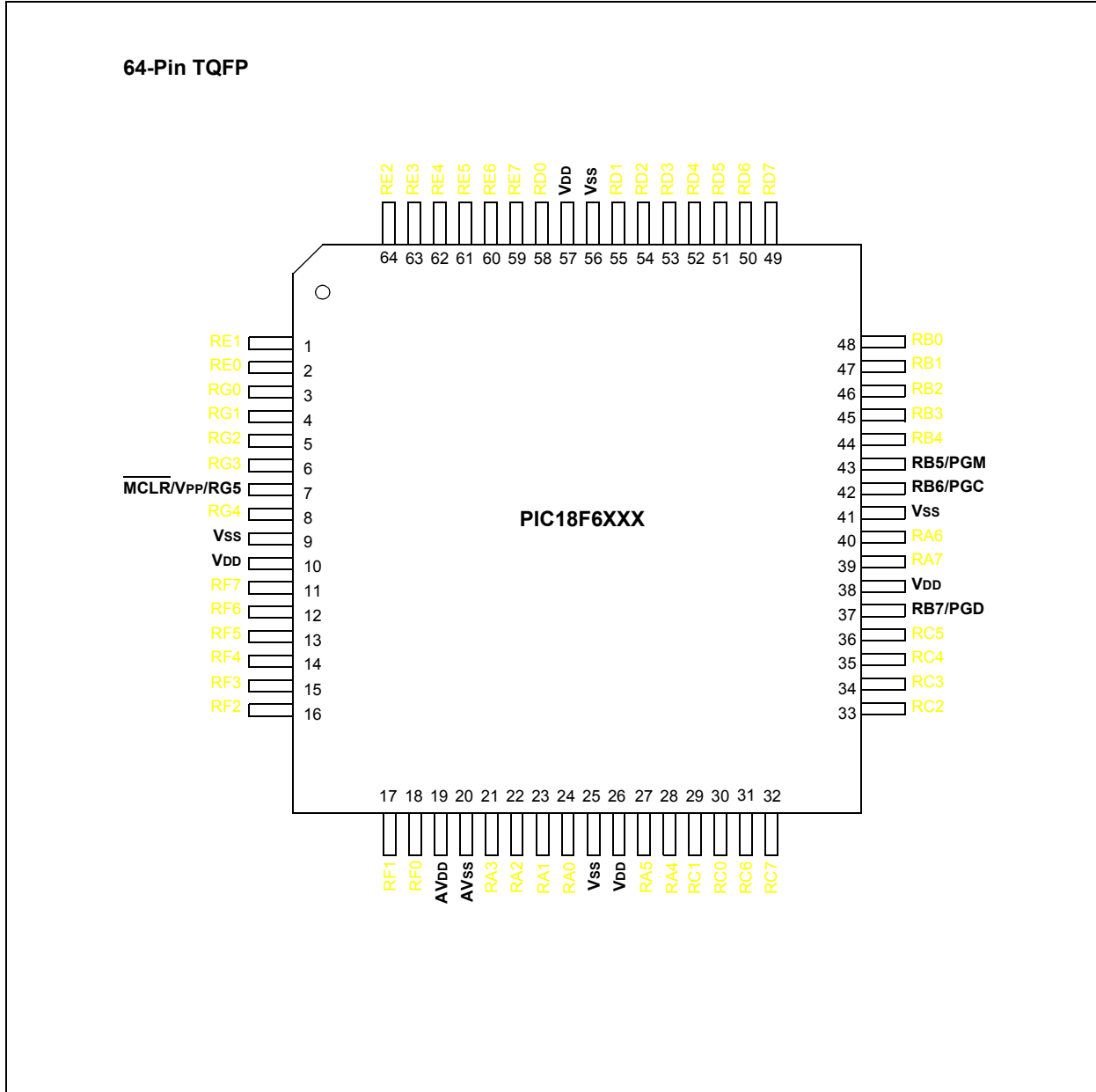| Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
$\overline{MCLR}/V_{PP}/RG5$	V_{PP}	P	Programming Enable
V_{DD}**(1)**	V_{DD}	P	Power Supply
V_{SS}**(1)**	V_{SS}	P	Ground
AV_{DD}	AV_{DD}	P	Analog Power Supply
AV_{SS}	AV_{SS}	P	Analog Ground
RB5	PGM	I	Low-Voltage ICSP™ Input when LVP Configuration bit equals '1'**(2)**
RB6	PGC	I	Serial Clock
RB7	PGD	I/O	Serial Data

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

Note 1: All power supply (V_{DD}) and ground (V_{SS}) pins must be connected.
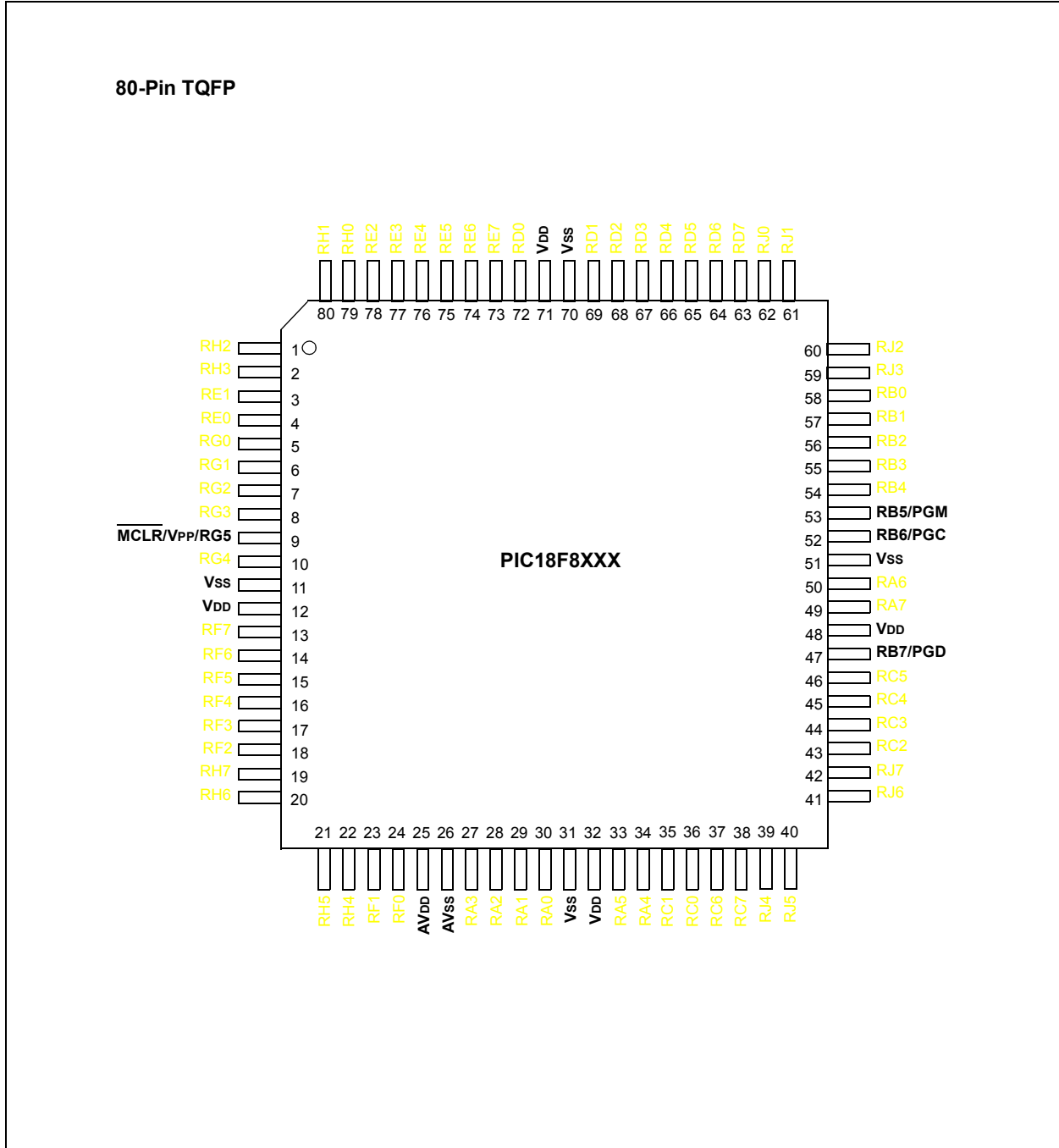
2: See Table 5-1 for more information.

PIC18F872X FAMILY

FIGURE 2-1: **PIC18F6527/6622/6627/6722/6628/6723 FAMILY PIN DIAGRAM**

64-Pin TQFP

Top pins (64–49): RE2, RE3, RE4, RE5, RE6, RE7, RD0, VDD, VSS, RD1, RD2, RD3, RD4, RD5, RD6, RD7

Left pins (1–16):
1 RE1
2 RE0
3 RG0
4 RG1
5 RG2
6 RG3
7 MCLR/VPP/RG5
8 RG4
9 VSS
10 VDD
11 RF7
12 RF6
13 RF5
14 RF4
15 RF3
16 RF2

Center: PIC18F6XXX

Right pins (48–33):
48 RB0
47 RB1
46 RB2
45 RB3
44 RB4
43 RB5/PGM
42 RB6/PGC
41 VSS
40 RA6
39 RA7
38 VDD
37 RB7/PGD
36 RC5
35 RC4
34 RC3
33 RC2

Bottom pins (17–32): RF1, RF0, AVDD, AVSS, RA3, RA2, RA1, RA0, VSS, VDD, RA5, RA4, RC1, RC0, RC6, RC7

FIGURE 2-2: **PIC18F8527/8622/8627/8722/8628/8723 FAMILY PIN DIAGRAM**
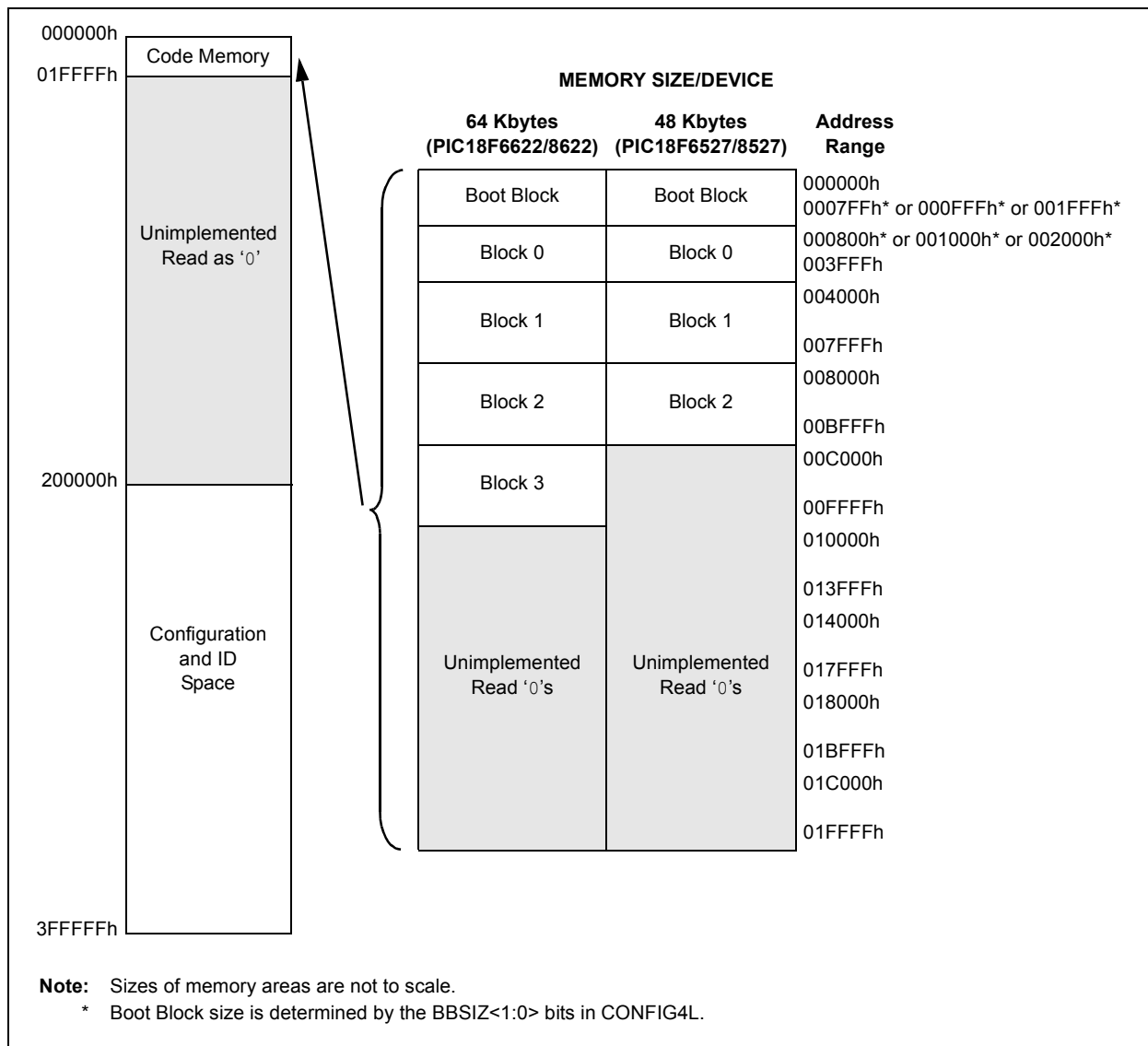

80-Pin TQFP

2.3 Memory Maps

For PIC18F6622/8622 devices, the code memory space extends from 000000h to 00FFFFh (64 Kbytes) in four 16-Kbyte blocks. For PIC18F6527/8527 devices, the code memory space extends from 000000h to 00BFFFh (48 Kbytes) in three 16-Kbyte blocks. Addresses, 0000h through 07FFh, 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 PIC18F6527/6622/8527/ 8622 devices can be configured as 1, 2 or 4K words (see Table 5-1). This is done through the BBSIZ<1:0> bits in the Configuration register, CONFIG4L. 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)
PIC18F6527	000000h-00BFFFh (48K)
PIC18F8527	
PIC18F6622	000000h-00FFFFh (64K)
PIC18F8622	

FIGURE 2-3: MEMORY MAP AND THE CODE MEMORY SPACE FOR PIC18F6527/6622/8527/8622 DEVICES

Note: Sizes of memory areas are not to scale.

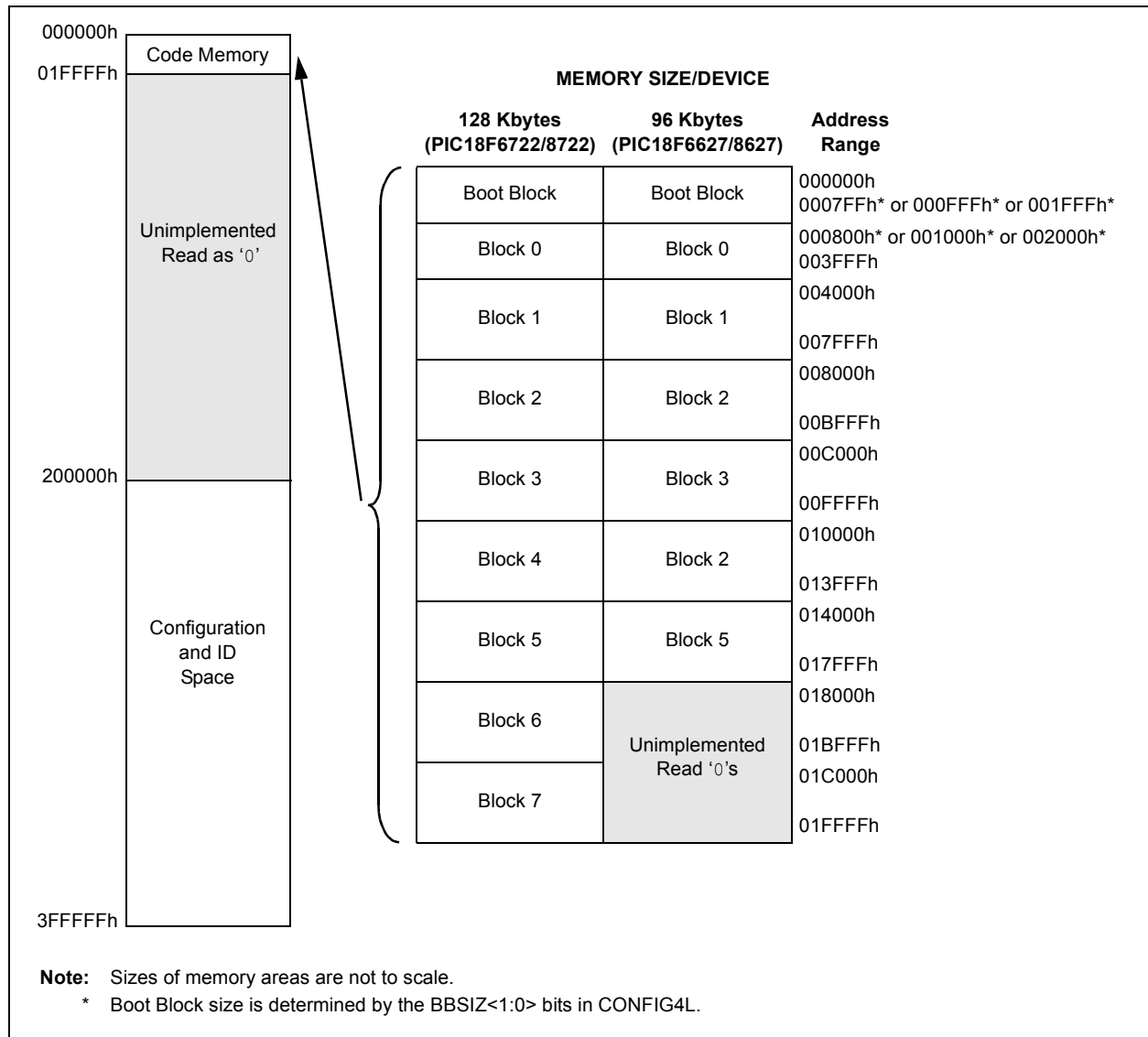* Boot Block size is determined by the BBSIZ<1:0> bits in CONFIG4L.

For PIC18F6722/8722/6723/8723 devices, the code memory space extends from 000000h to 01FFFFh (128 Kbytes) in eight 16-Kbyte blocks. For PIC18F6627/8627/6628/8628 devices, the code memory space extends from 000000h to 017FFFh (96 Kbytes) in six 16-Kbyte blocks. Addresses, 0000h through 07FFh, 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 PIC18F6627/6722/8627/8722/8723/8628/6723/6628 devices can be configured as 1, 2 or 4K words (see Table 5-1). This is done through the BBSIZ<1:0> bits in the Configuration register, CONFIG4L. It is important to note that increasing the size of the Boot Block decreases the size of Block 0.

TABLE 2-3: IMPLEMENTATION OF CODE MEMORY

Device	Code Memory Size (Bytes)
PIC18F6627	000000h-017FFFh (96K)
PIC18F8627	
PIC18F6628	
PIC18F8628	
PIC18F6722	000000h-01FFFFh (128K)
PIC18F8722	
PIC18F6723	
PIC18F8723	

FIGURE 2-4: MEMORY MAP AND THE CODE MEMORY SPACE FOR PIC18F6627/6722/8627/8722/8723/8628/6723/6628 DEVICES

128 Kbytes (PIC18F6722/8722)	96 Kbytes (PIC18F6627/8627)	Address Range
Boot Block	Boot Block	000000h 0007FFh* or 000FFFh* or 001FFFh*
Block 0	Block 0	000800h* or 001000h* or 002000h* 003FFFh
Block 1	Block 1	004000h 007FFFh
Block 2	Block 2	008000h 00BFFFh
Block 3	Block 3	00C000h 00FFFFh
Block 4	Block 2	010000h 013FFFh
Block 5	Block 5	014000h 017FFFh
Block 6	Unimplemented Read '0's	018000h 01BFFFh
Block 7		01C000h 01FFFFh

Memory map left side: Code Memory (000000h / 01FFFFh), Unimplemented Read as '0', 200000h, Configuration and ID Space, 3FFFFFh

MEMORY SIZE/DEVICE

Note: Sizes of memory areas are not to scale.

* Boot Block size is determined by the BBSIZ<1:0> bits in CONFIG4L.

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 "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.

2.3.1 MEMORY ADDRESS POINTER

Memory in the address space, 0000000h to 3FFFFFh, is addressed via the Table Pointer register, which is comprised of three Pointer registers:
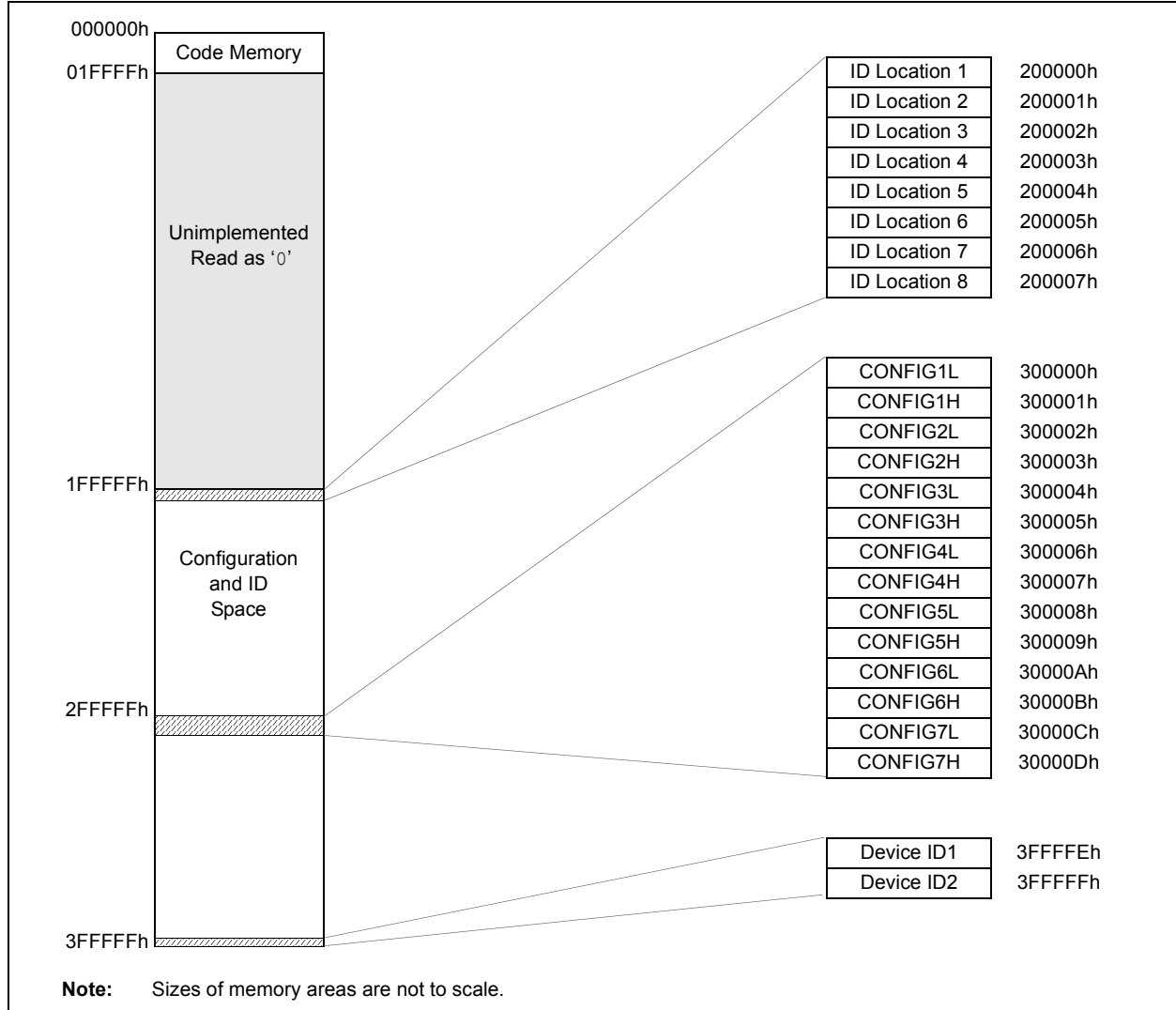
• TBLPTRU, at RAM address 0FF8h
• TBLPTRH, at RAM address 0FF7h
• TBLPTRL, at RAM address 0FF6h

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

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

FIGURE 2-5: **CONFIGURATION AND ID LOCATIONS FOR PIC18F872X FAMILY DEVICES**

|	
ID Location 1	200000h
ID Location 2	200001h
ID Location 3	200002h
ID Location 4	200003h
ID Location 5	200004h
ID Location 6	200005h
ID Location 7	200006h
ID Location 8	200007h

CONFIG1L	300000h
CONFIG1H	300001h
CONFIG2L	300002h
CONFIG2H	300003h
CONFIG3L	300004h
CONFIG3H	300005h
CONFIG4L	300006h
CONFIG4H	300007h
CONFIG5L	300008h
CONFIG5H	300009h
CONFIG6L	30000Ah
CONFIG6H	30000Bh
CONFIG7L	30000Ch
CONFIG7H	30000Dh

Device ID1	3FFFFEh
Device ID2	3FFFFFh

Memory map (left):
- 000000h — Code Memory
- 01FFFFFh
- Unimplemented Read as '0'
- 1FFFFFh
- Configuration and ID Space
- 2FFFFFh
- 3FFFFFh

Note: Sizes of memory areas are not to scale.

2.4 High-Level Overview of the Programming Process

Figure 2-6 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.

FIGURE 2-6: HIGH-LEVEL PROGRAMMING FLOW
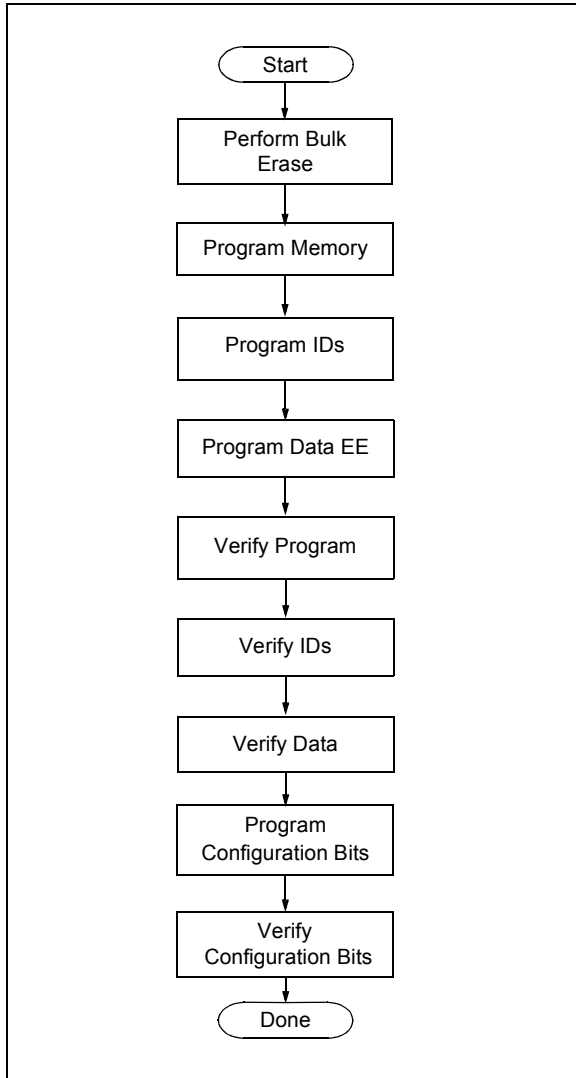
2.5 Entering and Exiting High-Voltage ICSP Program/Verify Mode

As shown in Figure 2-7, the High-Voltage ICSP Program/Verify mode is entered by holding PGC and PGD low and then raising $\overline{MCLR}$/VPP/RG5 to VIHH (high voltage). Once in this mode, the code memory, data EEPROM, ID locations and Configuration bits can be accessed and programmed in serial fashion. Figure 2-8 shows the exit sequence.

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

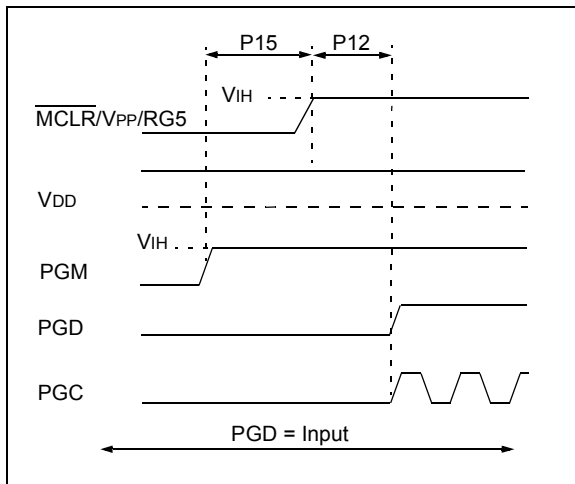**FIGURE 2-7: ENTERING HIGH-VOLTAGE PROGRAM/VERIFY MODE**
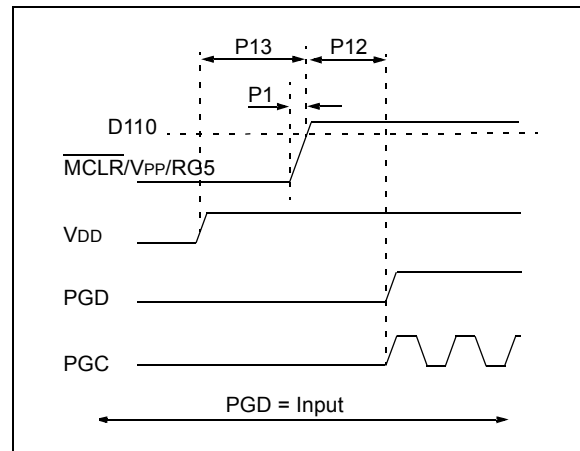
FIGURE 2-8: EXITING HIGH-VOLTAGE PROGRAM/VERIFY MODE
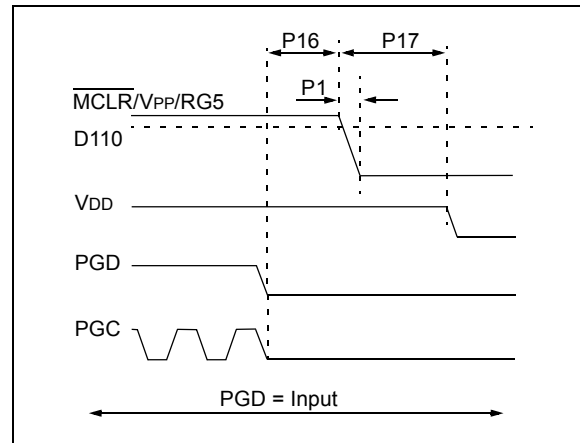
2.6 Entering and Exiting Low-Voltage ICSP Program/Verify Mode

When the LVP Configuration bit is '1' (see **Section 5.3 "Single-Supply ICSP Programming"**), the Low-Voltage ICSP mode is enabled. As shown in Figure 2-9, Low-Voltage ICSP Program/Verify mode is entered by holding PGC and PGD low, placing a logic high on PGM and then raising MCLR/VPP/RG5 to VIH. In this mode, the RB5/PGM pin is dedicated to the programming function and ceases to be a general purpose I/O pin. Figure 2-10 shows the exit sequence.

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

FIGURE 2-9: ENTERING LOW-VOLTAGE PROGRAM/VERIFY MODE

FIGURE 2-10: EXITING LOW-VOLTAGE PROGRAM/VERIFY MODE
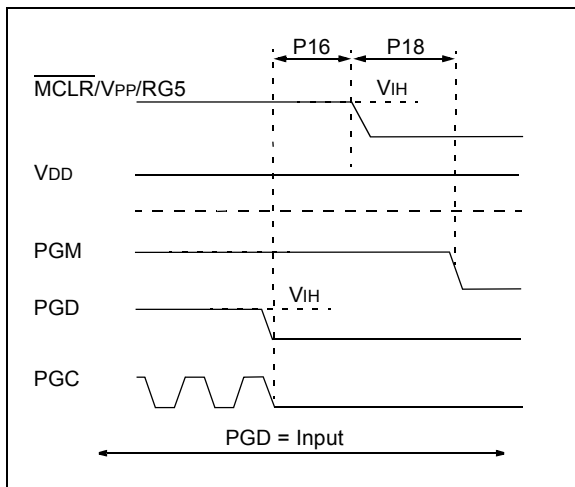
2.7 Serial Program/Verify Operation

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.

2.7.1 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 2-4. 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 2-5. The 4-bit command and data are shown, Most Significant bit (MSb) first. The command operand, or "Data Payload", is shown <LSB><MSB>. Figure 2-11 demonstrates how to serially present a 20-bit command/operand to the device.

2.7.2 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 2-4: COMMANDS FOR PROGRAMMING

Description	4-Bit Command
Core Instruction (shift in16-bit instruction)	0000
Shift Out TABLAT Register	0010
Table Read	1000
Table Read, Post-Increment	1001
Table Read, Post-Decrement	1010
Table Read, Pre-Increment	1011
Table Write	1100
Table Write, Post-Increment by 2	1101
Table Write, Start Programming, Post-Increment by 2	1110
Table Write, Start Programming	1111

TABLE 2-5: SAMPLE COMMAND SEQUENCE

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

FIGURE 2-11: **TABLE WRITE, POST-INCREMENT TIMING (1101)**
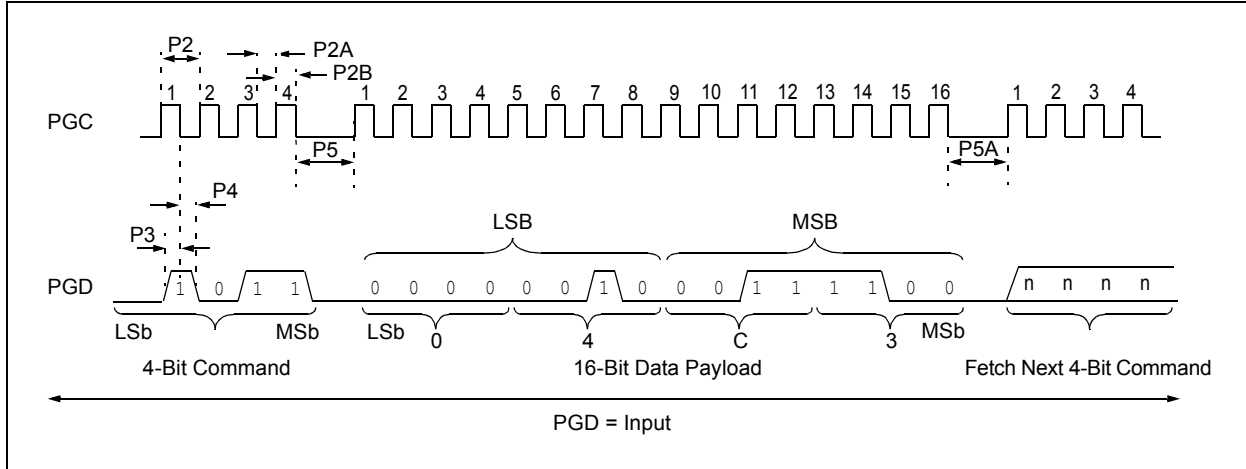
3.0 DEVICE PROGRAMMING

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

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

When using the EECON1 register to act on 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: DATA EEPROM CONTROL REGISTER 1

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:		
	S = Settable bit	
R = Readable bit	W = Writable bit	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 cycle or 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. 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.

PIC18F872X FAMILY

3.1 ICSP Erase

3.1.1 HIGH-VOLTAGE ICSP BULK ERASE

Erasing code or data EEPROM is accomplished by configuring two Bulk Erase Control registers located at 3C0004h and 3C0005h. 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. If data EEPROM is code-protected (CPD = 0), the user must request an erase of data EEPROM (e.g., 0084h as shown in Table 3-1).

TABLE 3-1: BULK ERASE OPERATIONS

Description	Data (3C0005h:3C0004h)
Chip Erase	FF87h
Erase Data EEPROM	0084h
Erase Boot Block	0081h
Erase Config Bits	0082h
Erase Code EEPROM Block 0	0180h
Erase Code EEPROM Block 1	0280h
Erase Code EEPROM Block 2	0480h
Erase Code EEPROM Block 3	0880h
Erase Code EEPROM Block 4	1080h
Erase Code EEPROM Block 5	2080h
Erase Code EEPROM Block 6	4080h
Erase Code EEPROM Block 7	8080h

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

The code sequence to erase the entire device is shown in Table 3-2 and the flowchart 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.

TABLE 3-2: BULK ERASE COMMAND SEQUENCE

| 4-Bit Command	Data Payload	Core Instruction
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 05	MOVLW 05h
0000	6E F6	MOVWF TBLPTRL
1100	FF FF	Write FFh to 3C0005h
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 04	MOVLW 04h
0000	6E F6	MOVWF TBLPTRL
1100	87 87	Write 87h TO 3C0004h to erase entire device. NOP
0000	00 00	Hold PGD low until
0000	00 00	erase completes.

FIGURE 3-1: BULK ERASE FLOW
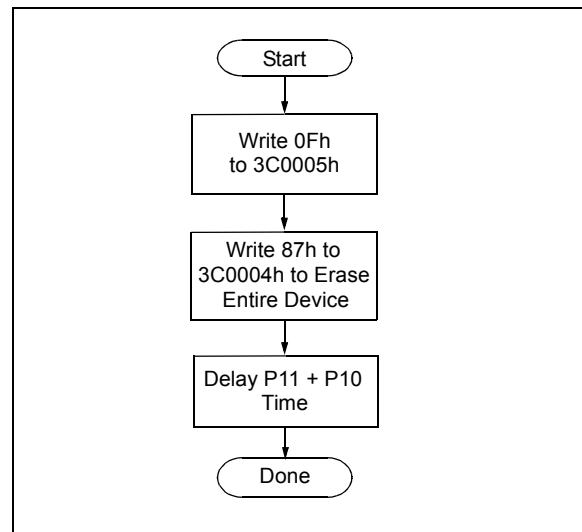
FIGURE 3-2:   **BULK ERASE TIMING**
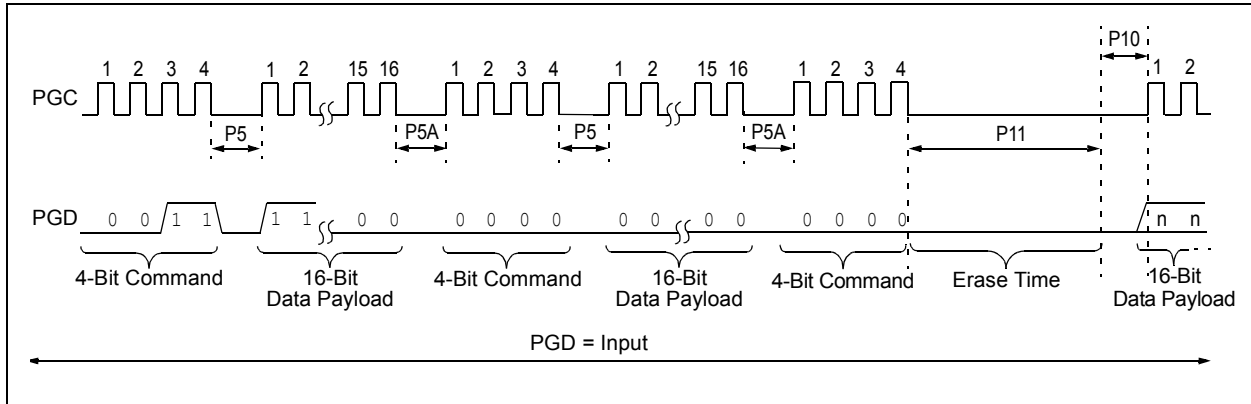
3.1.2   LOW-VOLTAGE ICSP BULK ERASE

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

If it is determined that a program memory erase must be performed at a supply voltage below the Bulk Erase limit, refer to the erase methodology described in **Section 3.1.3 "ICSP Row Erase"** and **Section 3.2.2 "Modifying Code Memory"**.

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 "Data EEPROM Programming"** and write '1's to the array.

3.1.3   ICSP ROW ERASE

Regardless of whether high or low-voltage ICSP is used, it is possible to erase one row (64 bytes of data) provided the block is not code or write-protected. Rows are located at static boundaries, beginning at program memory address, 000000h, extending to the internal program memory limit (see **Section 2.3 "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 PIC18F872X family device is shown in Table 3-3. The flowchart shown in Figure 3-3 depicts the logic necessary to completely erase a PIC18F872X 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.

PIC18F872X FAMILY

TABLE 3-3: **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 A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
0000	84 A6	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 A6	BSF EECON1, FREE
0000	82 A6	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**
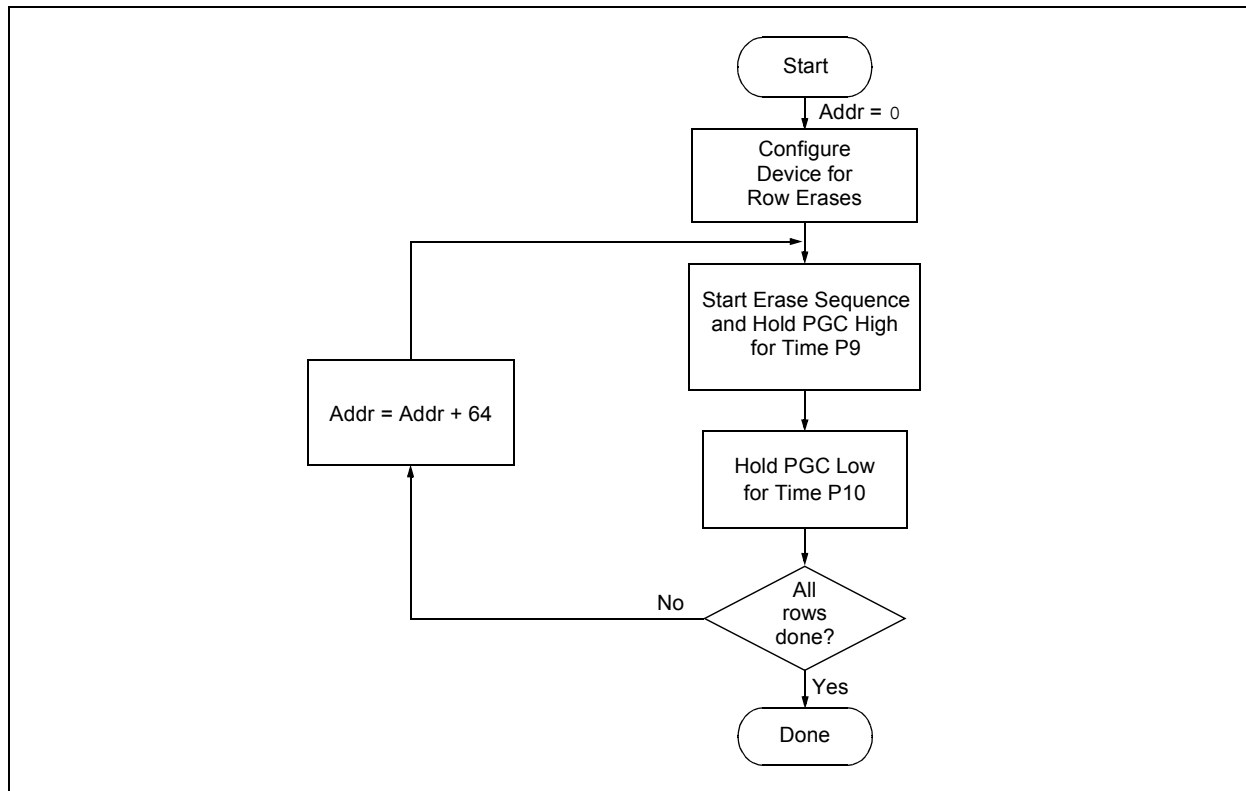
3.2   Code Memory Programming

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 3-4, can be mapped to any location of the same size beginning at 000000h. The actual memory write sequence takes the contents of this buffer and programs the proper amount of 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 PIC18F872X family device is shown in Table 3-5. The flowchart, shown in Figure 3-6, depicts the logic necessary to completely write a PIC18F872X 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.

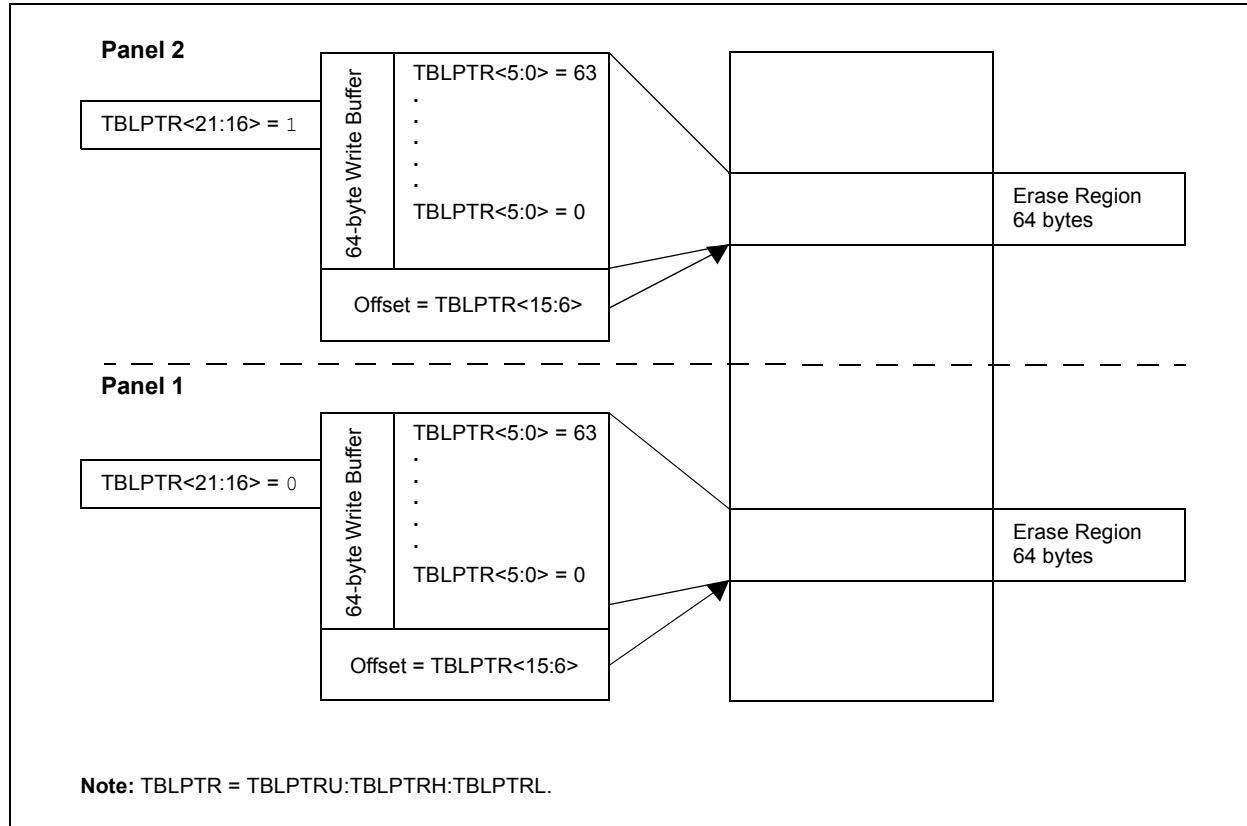**TABLE 3-4:   WRITE AND ERASE BUFFER SIZES**

Device (Arranged by Family)	Write Buffer Size in Bytes	Erase Buffer Size in Bytes
PIC18FX527	64	64
PIC18FX622		
PIC18FX627		
PIC18FX628		
PIC18FX722		
PIC18FX723		

FIGURE 3-4:   TABLE WRITE AND START PROGRAMMING INSTRUCTION TIMING (1111)
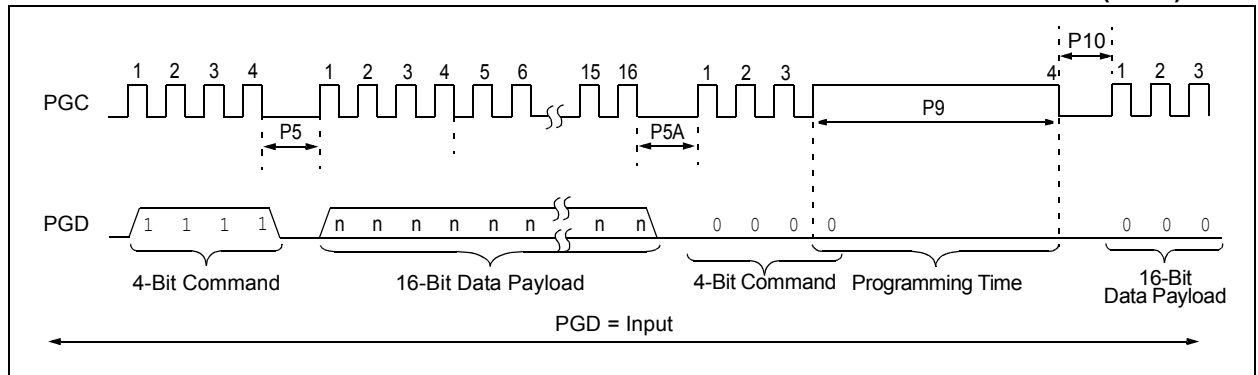
FIGURE 3-5: **ERASE AND WRITE BOUNDARIES**

Note: TBLPTR = TBLPTRU:TBLPTRH:TBLPTRL.
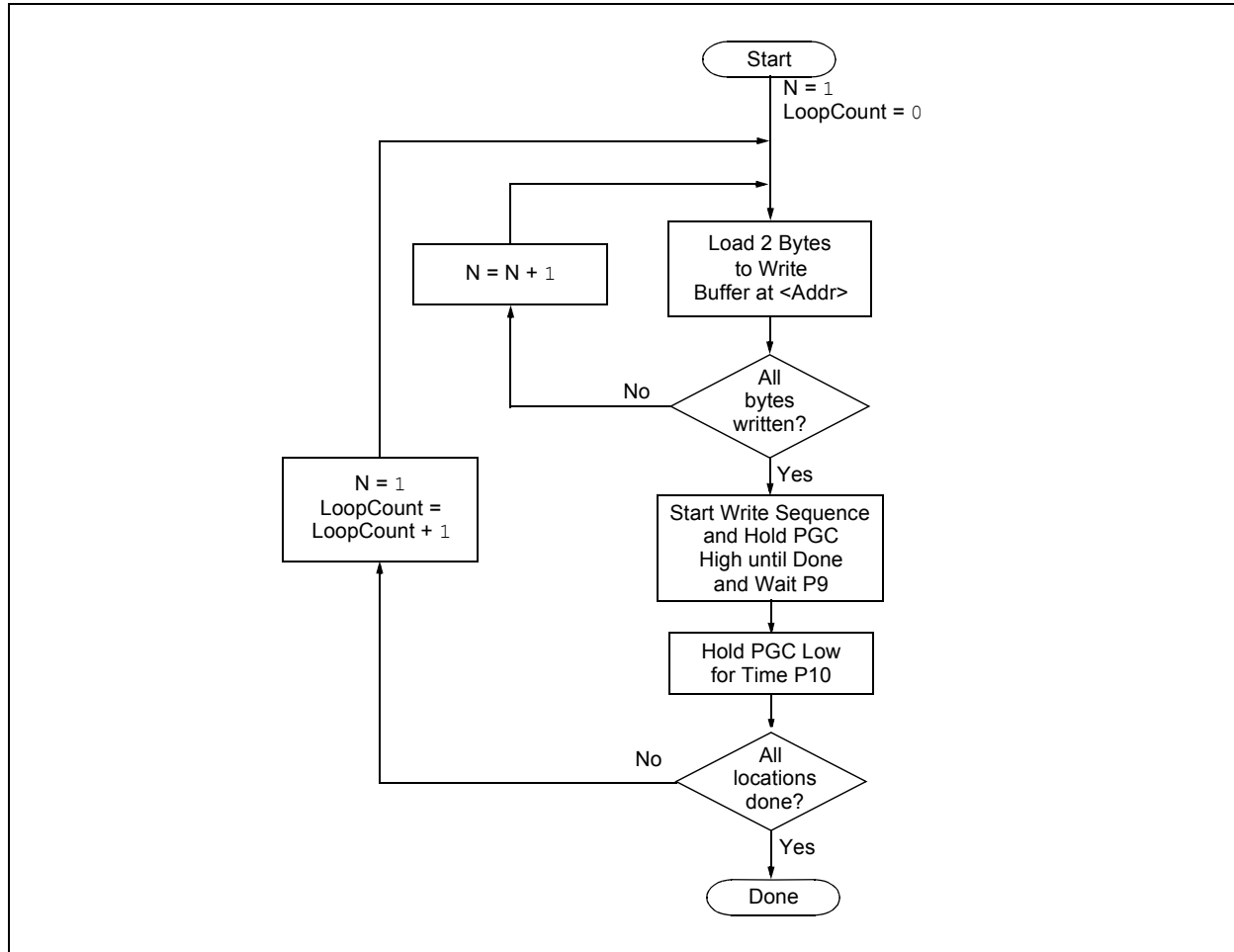
3.2.1 PROGRAMMING

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 3-5:   WRITE CODE MEMORY CODE SEQUENCE FOR PROGRAMMING

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to configuration memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	8C A6	BSF EECON1, CFGS
0000	84 A6	BSF EECON1, WREN
Step 2: Direct access to code memory and enable writes.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
Step 3: Load write buffer for panel.		
0000	0E	MOVLW <Addr[21:16]>
0000	<Addr[21:16]>	MOVWF TBLPTRU
0000	6E F8	MOVLW <Addr[15:8]>
0000	0E <Addr[15:8]>	MOVWF TBLPTRH
0000	6E F7	MOVLW <Addr[7:0]>
0000	0E <Addr[7:0]>	MOVWF TBLPTRL
1101	6E F6	Write 2 bytes and post-increment address by 2
.	<MSB><LSB>	.
.	.	. Repeat 28 times
.	.	.
1111	.	Write 2 bytes and start programming
0000	<MSB><LSB>	NOP - hold SCLK high for time P9, low for time P10
	00 00	
To continue writing data, repeat step 3, 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 has been Bulk Erased prior to programming (see **Section 3.1.1 "High-Voltage ICSP Bulk Erase"**). It may be the case, however, that the user wishes to modify only a section of an already programmed device.

The appropriate number of bytes required for the erase buffer must be read out of 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 3-6: 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 A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
Step 3: Set the Table Pointer for the block to be erased.		
0000	0E <Addr[21:16]>	MOVLW <Addr[21:16]>
0000	6E F8	MOVWF TBLPTRU
0000	0E <Addr[8:15]>	MOVLW <Addr[8:15]>
0000	6E F7	MOVWF TBLPTRH
0000	0E <Addr[7:0]>	MOVLW <Addr[7:0]>
0000	6E F6	MOVWF TBLPTRL
Step 4: Enable memory writes and set up an erase.		
0000	84 A6	BSF EECON1, WREN
0000	88 A6	BSF EECON1, FREE
Step 5: Initiate erase.		
0000	82 A6	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 A6	BSF EECON1, EEPGD
0000	8C A6	BSF EECON1, CFGS
0000	84 A6	BSF EECON1, WREN
Step 7: Direct access to code memory and enable writes.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	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 28 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 the 64 bytes at each iteration of the loop.		
Step 9: Disable writes.		
0000	94 A6	BCF EECON1, WREN

PIC18F872X FAMILY

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.
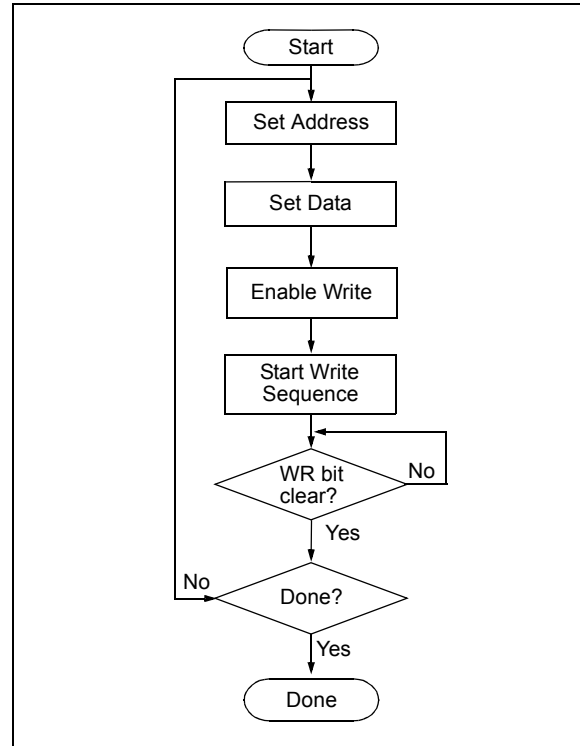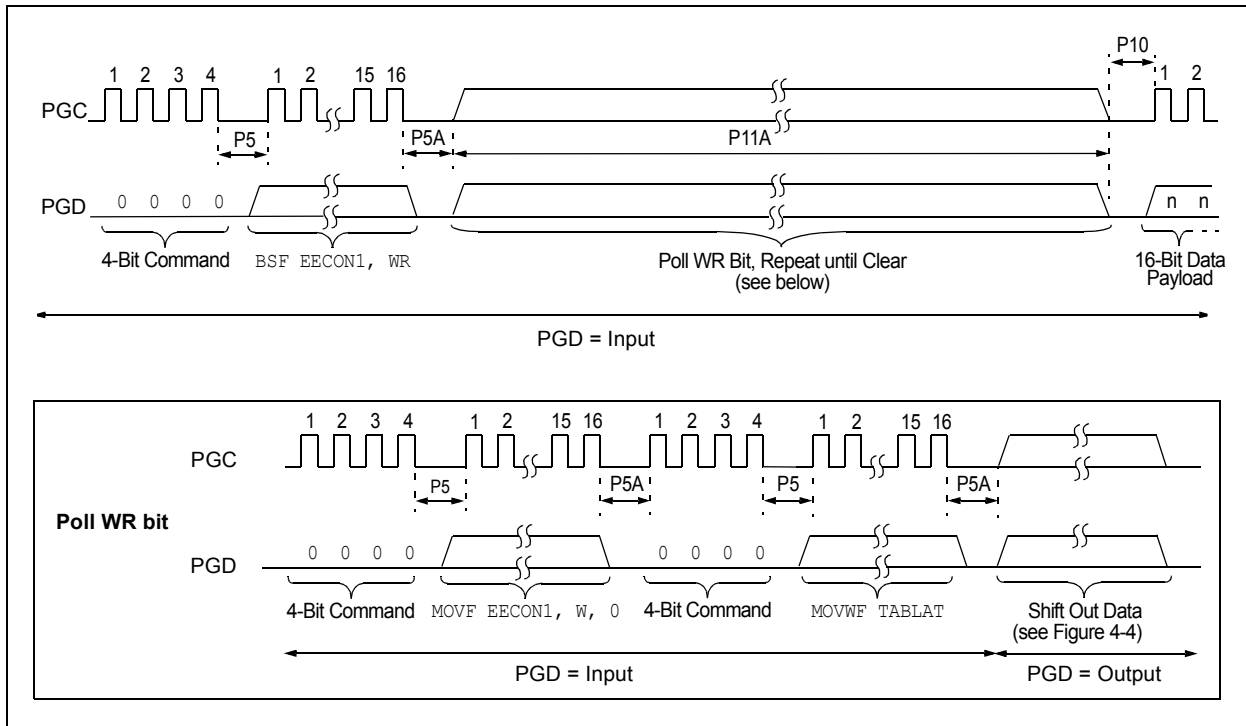
FIGURE 3-7: PROGRAM DATA FLOW

FIGURE 3-8: DATA EEPROM WRITE TIMING

TABLE 3-7: **PROGRAMMING DATA MEMORY**

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to data EEPROM.		
0000 0000	9E A6 9C A6	BCF EECON1, EEPGD BCF EECON1, CFGS
Step 2: Set the data EEPROM Address Pointer.		
0000 0000 0000 0000	0E \<Addr> 6E A9 OE \<AddrH> 6E AA	MOVLW \<Addr> MOVWF EEADR MOVLW \<AddrH> MOVWF EEADRH
Step 3: Load the data to be written.		
0000 0000	0E \<Data> 6E A8	MOVLW \<Data> MOVWF EEDATA
Step 4: Enable memory writes.		
0000	84 A6	BSF EECON1, WREN
Step 5: Initiate write.		
0000	82 A6	BSF EECON1, WR
Step 6: Poll WR bit, repeat until the bit is clear.		
0000 0000 0000 0010	50 A6 6E F5 00 00 \<MSB>\<LSB>	MOVF EECON1, W, 0 MOVWF TABLAT NOP Shift out data[1]
Step 7: Hold PGC low for time, P10.		
Step 8: Disable writes.		
0000	94 A6	BCF EECON1, WREN
Repeat steps 2 through 8 to write more data.		

Note 1: See Figure 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 3-8 demonstrates the code sequence required to write the ID locations.

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

TABLE 3-8:   WRITE ID SEQUENCE

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to code memory and enable writes.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	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 3-5 should be used, except that the address used in "Step 2" will be in the range of 000000h to 0007FFh, 000000h to 000FFFh or 000000h to 001FFFh, as defined by the BBSIZ<1:0> bits 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 3-9.

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

TABLE 3-9: 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 A6	BSF EECON1, EEPGD
0000	8C A6	BSF EECON1, CFGS
Step 2**(1)**: Set Table Pointer for configuration 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 TBLPTRU
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 P9 and low for time P10.

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

FIGURE 3-9: CONFIGURATION PROGRAMMING FLOW
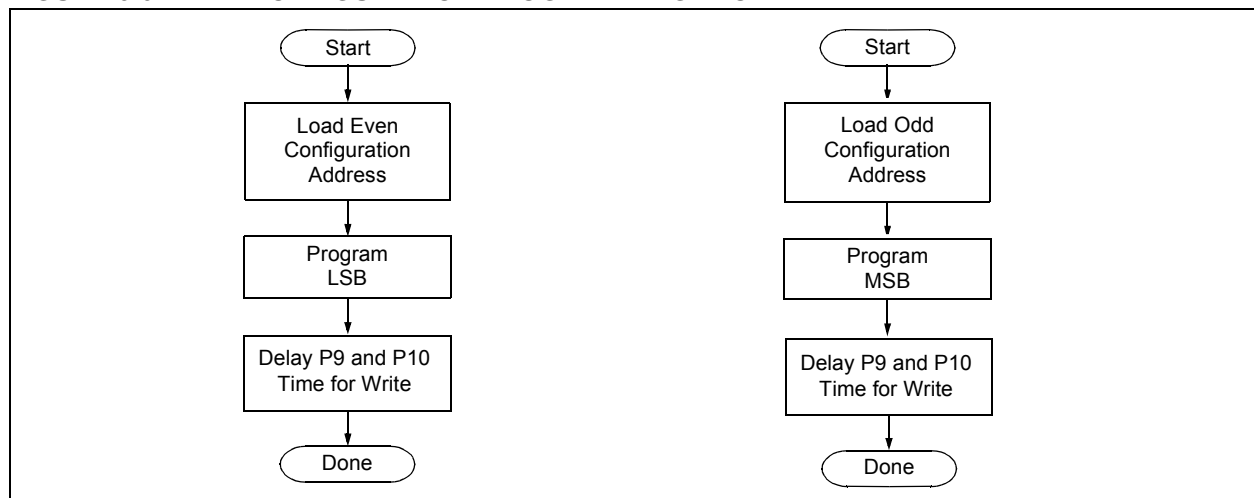
PIC18F872X FAMILY

3.7 External Memory Bus (PIC18F8525/8622 Only)

The PIC18F8525/8622 External Memory Bus (EMB) can be activated during In-Circuit Serial Programming™ (ICSP™) to program external Flash memory devices. The EMB is always disabled upon Program mode entry. TCFG3L, located at address 380004h, is a special register available in ICSP which overrides CONFIG3L.

The code sequence to enable the EMB for Byte Write mode, with 0 wait counts during ICSP, is shown in Table 3-10.

TABLE 3-10: CONFIGURE EXTERNAL MEMORY BUS CODE SEQUENCE

4-Bit Command	Data Payload	Core Instruction
0000	0E B0	MOVLW B0h
0000	6E 9C	MOVWF MEMCON
0000	0E 38	MOVLW 38h
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	70 70	Write 70h to 380004h

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 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 and then shift out on PGD, LSb to MSb.		
1001	00 00	TBLRD *+

### FIGURE 4-1: TABLE READ, POST-INCREMENT INSTRUCTION TIMING (1001)
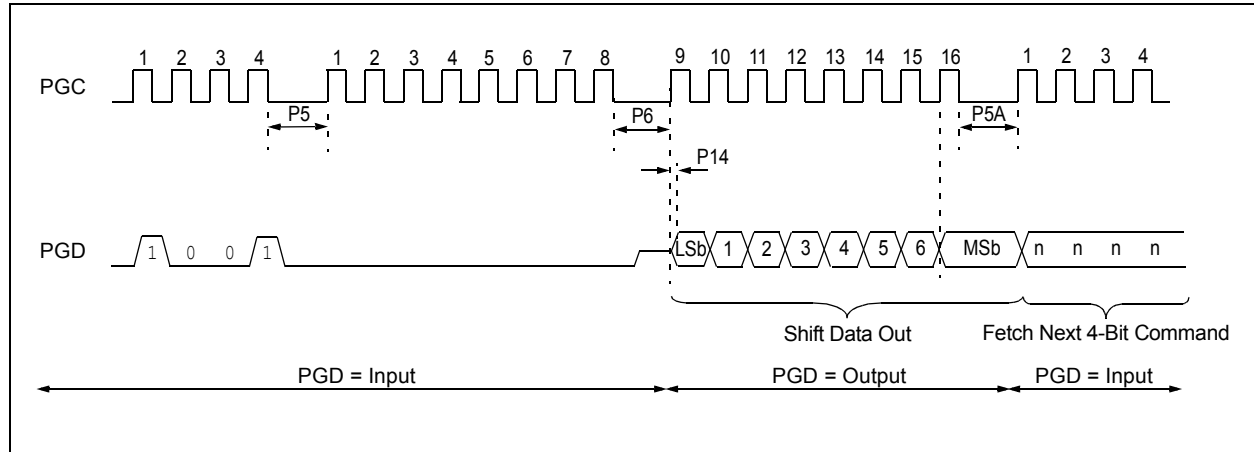
PIC18F872X FAMILY

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 unimplemented address, 020000h.

FIGURE 4-2: **VERIFY CODE MEMORY FLOW**
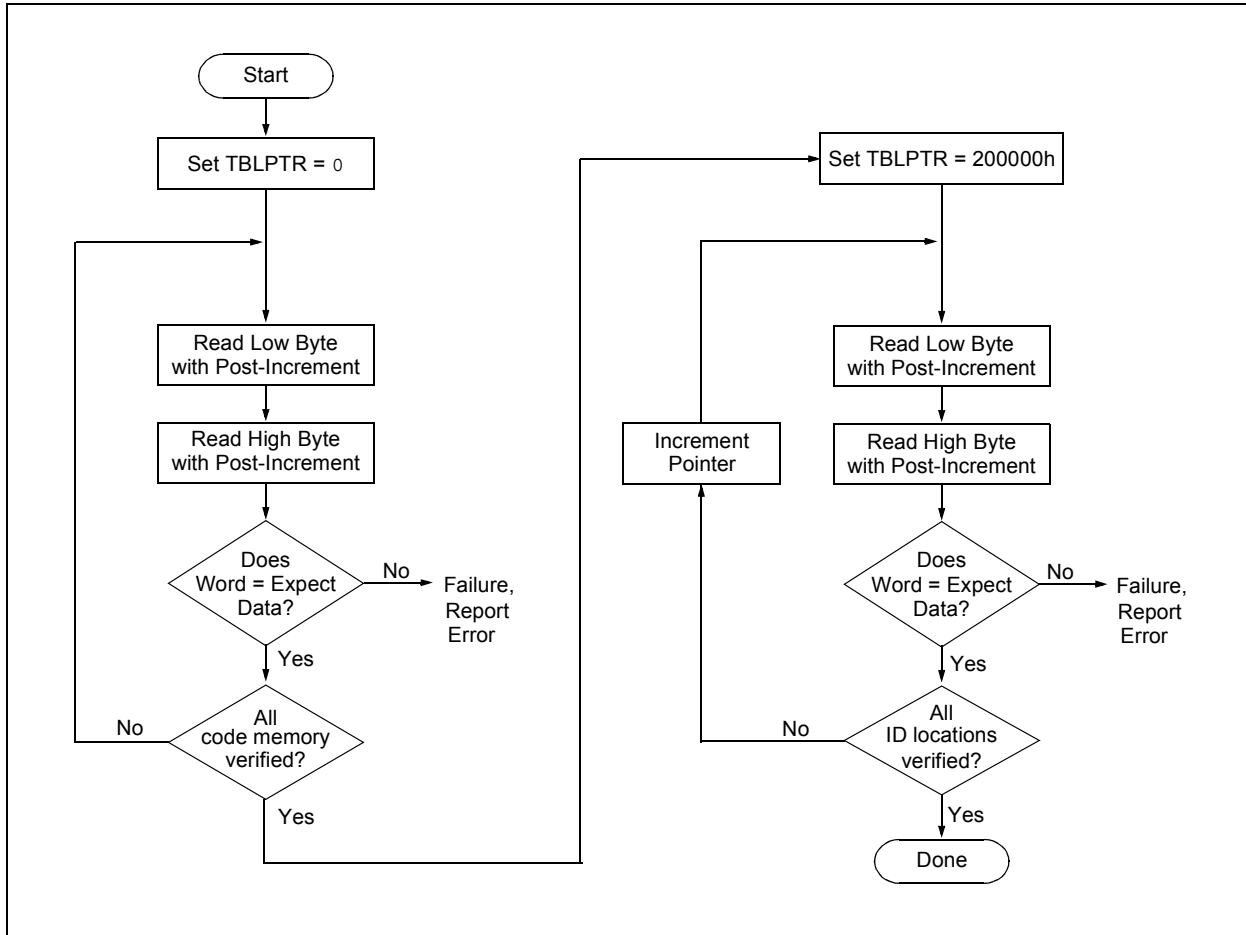
4.3 Verify Configuration Bits

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.
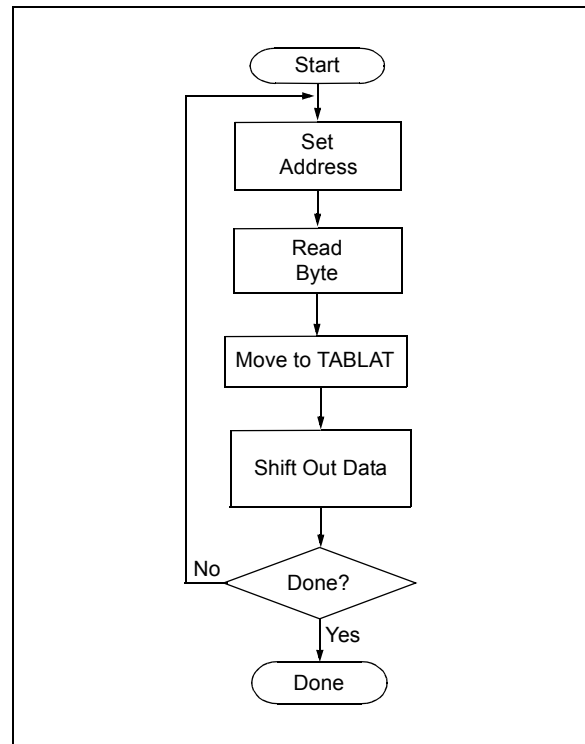
FIGURE 4-3: READ DATA EEPROM FLOW

TABLE 4-2: READ DATA EEPROM MEMORY

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to data EEPROM.		
0000	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: Initiate a memory read.		
0000	80 A6	BSF EECON1, RD
Step 4: Load data into the Serial Data Holding register.		
0000	50 A8	MOVF EEDATA, W, 0
0000	6E F5	MOVWF TABLAT
0000	00 00	NOP
0010	<MSB><LSB>	Shift Out Data[1]

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

PIC18F872X FAMILY

FIGURE 4-4: **SHIFT OUT DATA HOLDING REGISTER TIMING (0010)**
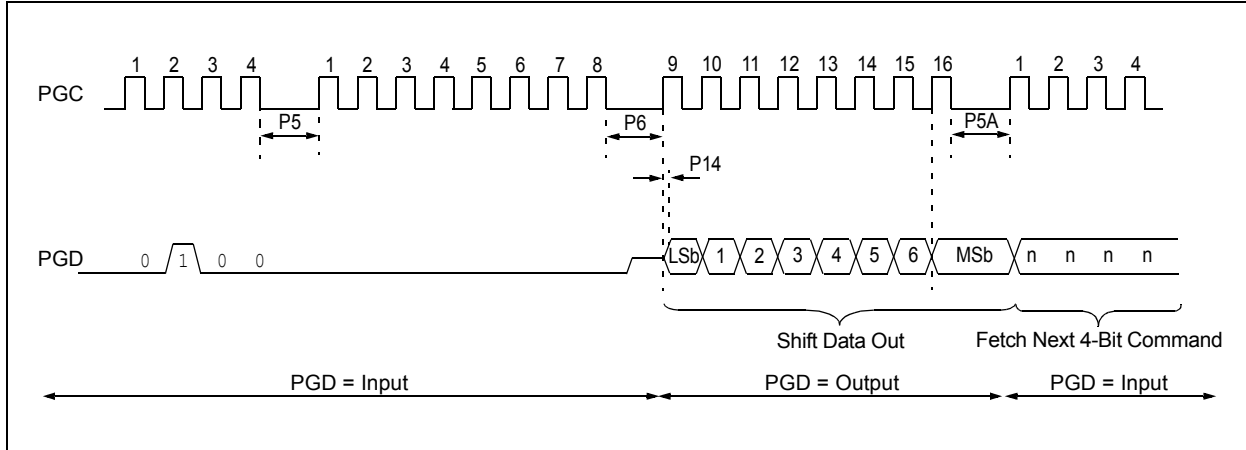
4.5 Verify Data EEPROM

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 PIC18F872X 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**
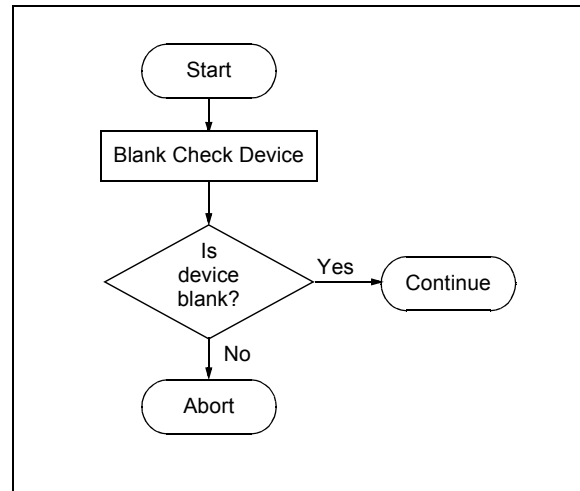
5.0 CONFIGURATION WORD

The PIC18F872X 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 information (ID) 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
300001h	CONFIG1H	IESO	FCMEN	—	—	FOSC3	FOSC2	FOSC1	FOSC0	00-- 0111
300002h	CONFIG2L	—	—	—	BORV1	BORV0	BOREN1	BOREN0	$\overline{\text{PWRTEN}}$	---1 1111
300003h	CONFIG2H	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN	---1 1111
300004h	CONFIG3L[5]	WAIT	BW	ABW1	ABW0	—	—	PM1[5]	PM0[5]	1111 --11
300005h	CONFIG3H	MCLRE	—	—	—	—	LPT1OSC	ECCPMX[5]	CCP2MX	1--- -011
300006h	CONFIG4L	$\overline{\text{DEBUG}}$	XINST	BBSIZ1	BBSIZ0	—	LVP	—	STVREN	1000 -1-1
300008h	CONFIG5L	CP7[1]	CP6[1]	CP5[2]	CP4[2]	CP3[3]	CP2	CP1	CP0	1111 1111
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—	11-- ----
30000Ah	CONFIG6L	WRT7[1]	WRT6[1]	WRT5[2]	WRT4[2]	WRT3[3]	WRT2	WRT1	WRT0	1111 1111
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	—	—	—	—	—	111- ----
30000Ch	CONFIG7L	EBRT7[1,6]	EBRT6[1,6]	EBTR5[2,6]	EBTR4[2,6]	EBTR3[3,6]	EBTR2[6]	EBTR1[6]	EBTR0[6]	1111 1111
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—	-1-- ----
3FFFFEh	DEVID1[4]	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxxx xxxx
3FFFFFh	DEVID2[4]	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: Unimplemented in PIC18F6527/6622/6627/8527/8622/8627 devices.
2: Unimplemented in PIC18F6527/6622/8527/8622 devices.
3: Unimplemented in PIC18F6527/8527 devices.
4: See Register 25-13 in the PIC18F872X Family Data Sheet for DEVID1 values. DEVID registers are read-only and can not be programmed by the user.
5: Implemented in 80-pin devices only. On 64-pin devices, these bits are reserved and should always be maintained as '1'.
6: It is recommended to enable the corresponding CPx bit to protect blocks from external read operations.

PIC18F872X FAMILY

5.2 Device ID Word

The Device ID Word (DEVID<2:1>) for the PIC18F872X 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.

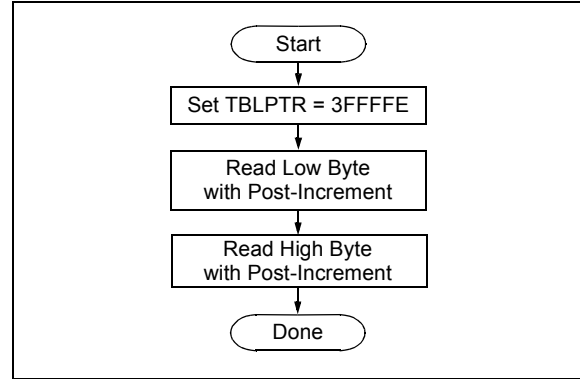**FIGURE 5-1: READ DEVICE ID WORD FLOW**

TABLE 5-2: DEVICE ID VALUE

Device	Device ID Value	
	DEVID2	DEVID1
PIC18F6527	13h	010x xxxx
PIC18F6622	13h	100x xxxx
PIC18F6627	13h	110x xxxx
PIC18F6628	49h	110x xxxx
PIC18F6722	14h	000x xxxx
PIC18F6723	4Ah	000x xxxx
PIC18F8527	13h	011x xxxx
PIC18F8622	13h	101x xxxx
PIC18F8627	13h	111x xxxx
PIC18F8628	49h	111x xxxx
PIC18F8722	14h	001x xxxx
PIC18F8723	4Ah	001x xxxx

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

TABLE 5-3: PIC18F872X FAMILY CONFIGURATION BIT DESCRIPTIONS

Bit Name	Configuration Words	Description
IESO	CONFIG1H	Internal External Switchover bit `1` = Internal External Switchover mode enabled `0` = Internal External Switchover mode disabled
FCMEN	CONFIG1H	Fail-Safe Clock Monitor Enable bit `1` = Fail-Safe Clock Monitor enabled `0` = Fail-Safe Clock Monitor disabled
FOSC<3:0>	CONFIG1H	Oscillator Selection bits `11xx` = External RC oscillator, CLKO function on RA6 `101x` = External RC oscillator, CLKO function on RA6 `1001` = Internal RC oscillator, CLKO function on RA6, port function on RA7 `1000` = Internal RC oscillator, port function on RA6, port function on RA7 `0111` = External RC oscillator, port function on RA6 `0110` = HS oscillator, PLL enabled (clock frequency = 4 x FOSC1) `0101` = EC oscillator, port function on RA6 `0100` = EC oscillator, CLKO function on RA6 `0011` = External RC oscillator, CLKO function on RA6 `0010` = HS oscillator `0001` = XT oscillator `0000` = LP oscillator
BORV<1:0>	CONFIG2L	Brown-out Reset Voltage bits `11` = V$_{BOR}$ set to 2.0V `10` = V$_{BOR}$ set to 2.7V `01` = V$_{BOR}$ set to 4.2V `00` = V$_{BOR}$ set to 4.5V
BOREN<1:0>	CONFIG2L	Brown-out Reset Enable bits `11` = Brown-out Reset enabled in hardware only (SBOREN is disabled) `10` = Brown-out Reset enabled in hardware only and disabled in Sleep mode (SBOREN is disabled) `01` = Brown-out Reset enabled and controlled by software (SBOREN is enabled) `00` = Brown-out Reset disabled in hardware and software
PWRTEN	CONFIG2L	Power-up Timer Enable bit `1` = PWRT disabled `0` = PWRT enabled
WDTPS<3:0>	CONFIG2H	Watchdog Timer Postscaler Select bits `1111` = 1:32,768 `1110` = 1:16,384 `1101` = 1:8,192 `1100` = 1:4,096 `1011` = 1:2,048 `1010` = 1:1,024 `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 the SWDTEN bit)

Note 1: The BBSIZ<1:0> bits 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 PIC18F8XXX devices only.

TABLE 5-3: PIC18F872X FAMILY CONFIGURATION BIT DESCRIPTIONS (CONTINUED)

Bit Name	Configuration Words	Description
WAIT	CONFIG3L	External Bus Wait Enable bit 1 = Wait selections from MEMCON.WAIT<1:0> are unavailable and the device will not wait 0 = Wait programmed by MEMCON.WAIT<1:0>
BW	CONFIG3L	Data Bus Width Select bit 1 = 16-Bit External Bus mode 0 = 8-Bit External Bus mode
ABW<1:0>	CONFIG3L	Address Bus Width Select bits 11 = 20-bit address bus 10 = 16-bit address bus 01 = 12-bit address bus 00 = 8-bit address bus
PM<1:0>	CONFIG3L	Processor Data Memory Mode Select bits 11 = Microcontroller mode 10 = Microprocessor mode 01 = Microprocessor with Boot Block mode 00 = Extended Microcontroller mode
MCLRE	CONFIG3H	$\overline{\text{MCLR}}$ Pin Enable bit 1 = $\overline{\text{MCLR}}$ pin enabled, RE3 input pin disabled 0 = RE3 input pin enabled, $\overline{\text{MCLR}}$ pin disabled
LPT1OSC	CONFIG3H	Low-Power Timer1 Oscillator Enable bit 1 = Timer1 configured for low-power operation 0 = Timer1 configured for higher power operation
ECCPMX	CONFIG3H	ECCP Mux bit[2] 1 = Enhanced CCP1/3 (P1B/P1C/P3B/P3C) are multiplexed on to RE6, RE5, RE4 and RE3, respectively 0 = Enhanced CCP1/3 (P1B/P1C/P3B/P3C) are multiplexed on to RH7, RH6, RH5 and RH4, respectively
CCP2MX	CONFIG3H	CCP2 MUX bit 1 = CCP2 input/output is multiplexed with RC1 0 = CCP2 input/output is multiplexed with RB3 in Extended Microcontroller, Microprocessor or Microprocessor with Boot Block mode
$\overline{\text{DEBUG}}$	CONFIG4L	Background Debugger Enable bit 1 = Background debugger disabled, RB6 and RB7 configured as general purpose I/O pins 0 = Background debugger enabled, RB6 and RB7 are dedicated to In-Circuit Debug
XINST	CONFIG4L	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)
BBSIZ<1:0>[1]	CONFIG4L	Boot Block Size Select bits 11 = 4K words (8 Kbytes) Boot Block 10 = 4K words (8 Kbytes) Boot Block 01 = 2K words (4 Kbytes) Boot Block 00 = 1K words (2 Kbytes) Boot Block
LVP	CONFIG4L	Low-Voltage Programming Enable bit 1 = Low-Voltage Programming enabled, RB5 is the PGM pin 0 = Low-Voltage Programming disabled, RB5 is an I/O pin
STVREN	CONFIG4L	Stack Overflow/Underflow Reset Enable bit 1 = Reset on stack overflow/underflow enabled 0 = Reset on stack overflow/underflow disabled
CP7	CONFIG5L	Code Protection bits (Block 7 code memory area) 1 = Block 7 is not code-protected 0 = Block 7 is code-protected
CP6	CONFIG5L	Code Protection bits (Block 6 code memory area) 1 = Block 6 is not code-protected 0 = Block 6 is code-protected

Note 1: The BBSIZ<1:0> bits 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 PIC18F8XXX devices only.

TABLE 5-3: **PIC18F872X FAMILY CONFIGURATION BIT DESCRIPTIONS (CONTINUED)**

Bit Name	Configuration Words	Description
CP5	CONFIG5L	Code Protection bits (Block 5 code memory area) 1 = Block 5 is not code-protected 0 = Block 5 is code-protected
CP4	CONFIG5L	Code Protection bits (Block 4 code memory area) 1 = Block 4 is not code-protected 0 = Block 4 is code-protected
CP3	CONFIG5L	Code Protection bits (Block 3 code memory area) 1 = Block 3 is not code-protected 0 = Block 3 is code-protected
CP2	CONFIG5L	Code Protection bits (Block 2 code memory area) 1 = Block 2 is not code-protected 0 = Block 2 is code-protected
CP1	CONFIG5L	Code Protection bits (Block 1 code memory area) 1 = Block 1 is not code-protected 0 = Block 1 is code-protected
CP0	CONFIG5L	Code Protection bits (Block 0 code memory area) 1 = Block 0 is not code-protected 0 = Block 0 is code-protected
CPD	CONFIG5H	Code Protection bits (data EEPROM) 1 = Data EEPROM is not code-protected 0 = Data EEPROM is code-protected
CPB	CONFIG5H	Code Protection bits (Boot Block memory area) 1 = Boot Block is not code-protected 0 = Boot Block is code-protected
WRT7	CONFIG6L	Write Protection bits (Block 7 code memory area) 1 = Block 7 is not write-protected 0 = Block 7 is write-protected
WRT6	CONFIG6L	Write Protection bits (Block 6 code memory area) 1 = Block 6 is not write-protected 0 = Block 6 is write-protected
WRT5	CONFIG6L	Write Protection bits (Block 5 code memory area) 1 = Block 5 is not write-protected 0 = Block 5 is write-protected
WRT4	CONFIG6L	Write Protection bits (Block 4 code memory area) 1 = Block 4 is not write-protected 0 = Block 4 is write-protected
WRT3	CONFIG6L	Write Protection bits (Block 3 code memory area) 1 = Block 3 is not write-protected 0 = Block 3 is write-protected
WRT2	CONFIG6L	Write Protection bits (Block 2 code memory area) 1 = Block 2 is not write-protected 0 = Block 2 is write-protected
WRT1	CONFIG6L	Write Protection bits (Block 1 code memory area) 1 = Block 1 is not write-protected 0 = Block 1 is write-protected
WRT0	CONFIG6L	Write Protection bits (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

Note 1: The BBSIZ<1:0> bits 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 PIC18F8XXX devices only.

TABLE 5-3: **PIC18F872X FAMILY CONFIGURATION BIT DESCRIPTIONS (CONTINUED)**

Bit Name	Configuration Words	Description
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
EBTR7	CONFIG7L	Table Read Protection bit (Block 7 code memory area) 1 = Block 7 is not protected from table reads executed in other blocks 0 = Block 7 is protected from table reads executed in other blocks
EBTR6	CONFIG7L	Table Read Protection bit (Block 6 code memory area) 1 = Block 6 is not protected from table reads executed in other blocks 0 = Block 6 is protected from table reads executed in other blocks
EBTR5	CONFIG7L	Table Read Protection bit (Block 5 code memory area) 1 = Block 5 is not protected from table reads executed in other blocks 0 = Block 5 is protected from table reads executed in other blocks
EBTR4	CONFIG7L	Table Read Protection bit (Block 4 code memory area) 1 = Block 4 is not protected from table reads executed in other blocks 0 = Block 4 is protected from table reads executed in other blocks
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
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<1:0> bits 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 PIC18F8XXX devices only.

5.3 Single-Supply ICSP Programming

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

If Single-Supply Programming mode is not used, the LVP bit can be programmed to a '0' and RB5/PGM becomes a digital I/O pin. However, the LVP bit may only be programmed by entering the High-Voltage ICSP mode, where MCLR/VPP/RG5 is raised to VIHH. Once the LVP bit is programmed to a '0', only the High-Voltage ICSP mode is available and only the High-Voltage ICSP mode can be used to program the device.

Note 1:	The High-Voltage ICSP mode is always available, regardless of the state of the LVP bit, by applying VIHH to the MCLR/VPP/RG5 pin.
2:	While in Low-Voltage ICSP mode, the RB5 pin can no longer be used as a general purpose I/O.

5.4 Embedding Configuration Word Information in the HEX File

To allow portability of code, a PIC18F872X 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.

5.5 Embedding Data EEPROM Information in the HEX File

To allow portability of code, a PIC18F872X 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.

5.6 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 36 through 39) 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	0xAA at 0 and Max Address
PIC18F6527	None	SUM(0000:07FF)+SUM(0800:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+ (CONFIG0 & 0000)+(CONFIG1 & 00CF)+(CONFIG2 & 001F)+ (CONFIG3 & 001F)+(CONFIG4 & 00F3)+(CONFIG5 & 0087)+ (CONFIG6 & 00F5)+(CONFIG7 & 0000)+(CONFIG8 & 0007)+ (CONFIG9 & 00C0)+(CONFIG10 & 0007)+(CONFIG11 & 00E0)+ (CONFIG12 & 0007)+(CONFIG13 & 0040)	4340h	4296h
	Boot Block 4K words	SUM(0800:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+(CONFIG0 & 0000)+ (CONFIG1 & 00CF)+(CONFIG2 & 001F)+(CONFIG3 & 001F)+ (CONFIG4 & 00F3)+(CONFIG5 & 0087)+(CONFIG6 & 00F5)+ (CONFIG7 & 0000)+(CONFIG8 & 0007)+(CONFIG9 & 00C0)+ (CONFIG10 & 0007)+(CONFIG11 & 00E0)+(CONFIG12 & 0007)+ (CONFIG13 & 0040)+SUM(IDs)	4B78h	4B23h
	Boot/ Panel0/ Panel1	SUM(8000:BFFF)+(CONFIG0 & 0000)+(CONFIG1 & 00CF)+ (CONFIG2 & 001F)+(CONFIG3 & 001F)+(CONFIG4 & 00F3)+ (CONFIG5 & 0087)+(CONFIG6 & 00F5)+(CONFIG7 & 0000)+ (CONFIG8 & 0007)+(CONFIG9 & 00C0)+(CONFIG10 & 0007)+ (CONFIG11 & 00E0)+(CONFIG12 & 0007)+(CONFIG13 & 0040)+SUM(IDs)	C375h	C320h
	All	(CONFIG0 & 0000)+(CONFIG1 & 00CF)+(CONFIG2 & 001F)+ (CONFIG3 & 001F)+(CONFIG4 & 00F3)+(CONFIG5 & 0087)+ (CONFIG6 & 00F5)+(CONFIG7 & 0000)+(CONFIG8 & 0007)+ (CONFIG9 & 00C0)+(CONFIG10 & 0007)+(CONFIG11 & 00E0)+ (CONFIG12 & 0007)+(CONFIG13 & 0040)+SUM(IDs)	0371h	0371h
PIC18F6622	None	SUM(0000:07FF)+SUM(0800:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+ SUM(C000:FFFF)+(CONFIG0 & 0000)+(CONFIG1 & 00CF)+ (CONFIG2 & 001F)+(CONFIG3 & 001F)+(CONFIG4 & 00F3)+ (CONFIG5 & 0087)+(CONFIG6 & 00F5)+(CONFIG7 & 0000)+ (CONFIG8 & 0007)+(CONFIG9 & 00C0)+(CONFIG10 & 000F)+ (CONFIG11 & 00E0)+(CONFIG12 & 000F)+(CONFIG13 & 0040)	0358h	02AEh
	Boot Block 4K words	SUM(0800:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+SUM(C000:FFFF)+ (CONFIG0 & 0000)+(CONFIG1 & 00CF)+(CONFIG2 & 001F)+ (CONFIG3 & 001F)+(CONFIG4 & 00F3)+(CONFIG5 & 0087)+ (CONFIG6 & 00F5)+(CONFIG7 & 0000)+(CONFIG8 & 0007)+ (CONFIG9 & 00C0)+(CONFIG10 & 0007)+(CONFIG11 & 00E0)+ (CONFIG12 & 0007)+(CONFIG13 & 0040)+SUM(IDs)	0B90h	0B3Bh
	Boot/ Panel0/ Panel1	SUM(8000:BFFF)+SUM(C000:FFFF)+(CONFIG0 & 0000)+ (CONFIG1 & 00CF)+(CONFIG2 & 001F)+(CONFIG3 & 001F)+ (CONFIG4 & 00F3)+(CONFIG5 & 0087)+(CONFIG6 & 00F5)+ (CONFIG7 & 0000)+(CONFIG8 & 000F)+(CONFIG9 & 00C0)+ (CONFIG10 & 000F)+(CONFIG11 & 00E0)+(CONFIG12 & 000F)+ (CONFIG13 & 0040)+SUM(IDs)	838Dh	8338h
	All	(CONFIG0 & 0000)+(CONFIG1 & 00CF)+(CONFIG2 & 001F)+ (CONFIG3 & 001F)+(CONFIG4 & 00F3)+(CONFIG5 & 0087)+ (CONFIG6 & 00F5)+(CONFIG7 & 0000)+(CONFIG8 & 000F)+ (CONFIG9 & 00C0)+(CONFIG10 & 000F)+(CONFIG11 & 00E0)+ (CONFIG12 & 000F)+(CONFIG13 & 0040)+SUM(IDs)	0381h	0381h

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	0xAA at 0 and Max Address
PIC18F6627/ PIC18F6628	None	SUM(0000:07FF)+SUM(0800:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+ SUM(C000:FFFF)+SUM(10000:13FFF)+SUM(14000:17FFF)+ (CONFIG0 & 0000)+(CONFIG1 & 00CF)+(CONFIG2 & 001F)+ (CONFIG3 & 0000)+(CONFIG4 & 00F3)+(CONFIG5 & 0087)+ (CONFIG6 & 00F5)+(CONFIG7 & 0000)+(CONFIG8 & 003F)+ (CONFIG9 & 00C0)+(CONFIG10 & 003F)+(CONFIG11 & 00E0)+ (CONFIG12 & 003F)+(CONFIG13 & 0040)	83E8h	833Eh
	Boot Block 4K words	SUM(0800:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+SUM(C000:FFFF)+ SUM(10000:13FFF)+SUM(14000:17FFF)+CONFIG0 & 0000)+ (CONFIG1 & 00CF)+(CONFIG2 & 001F)+(CONFIG3 & 0000)+ (CONFIG4 & 00F3)+(CONFIG5 & 0087)+(CONFIG6 & 00F5)+ (CONFIG7 & 0000)+(CONFIG8 & 003F)+(CONFIG9 & 00C0)+ (CONFIG10 & 003F)+(CONFIG11 & 00E0)+(CONFIG12 & 003F)+ (CONFIG13 & 0040)+SUM(IDs)	8C20h	8BCBh
	Boot/ Panel0/ Panel1	SUM(8000:BFFF)+SUM(C000:FFFF)+SUM(10000:13FFF)+ SUM(14000:17FFF)+(CONFIG0 & 0000)+(CONFIG1 & 00CF)+ (CONFIG2 & 001F)+(CONFIG3 & 0000)+(CONFIG4 & 00F3)+ (CONFIG5 & 0087)+(CONFIG6 & 00F5)+(CONFIG7 & 0000)+ (CONFIG8 & 003F)+(CONFIG9 & 00C0)+(CONFIG10 & 003F)+ (CONFIG11 & 00E0)+(CONFIG12 & 003F)+(CONFIG13 & 0040)+SUM(IDs)	041Dh	03C8h
	All	(CONFIG0 & 0000)+(CONFIG1 & 00CF)+(CONFIG2 & 001F)+ (CONFIG3 & 0000)+(CONFIG4 & 00F3)+(CONFIG5 & 0087)+ (CONFIG6 & 00F5)+(CONFIG7 & 0000)+(CONFIG8 & 003F)+ (CONFIG9 & 00C0)+(CONFIG10 & 003F)+(CONFIG11 & 00E0)+ (CONFIG12 & 003F)+(CONFIG13 & 0040)+SUM(IDs)	03E1h	03E1h
PIC18F6722/ PIC18F6723	None	SUM(0000:07FF)+SUM(0800:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+ SUM(C000:FFFF)+SUM(10000:13FFF)+SUM(14000:17FFF)+ SUM(18000:1BFFF)+SUM(1C000:1FFFF)+(CONFIG0 & 0000)+ (CONFIG1 & 00CF)+(CONFIG2 & 001F)+(CONFIG3 & 0000)+ (CONFIG4 & 00F3)+(CONFIG5 & 0087)+(CONFIG6 & 00F5)+ (CONFIG7 & 0000)+(CONFIG8 & 00FF)+(CONFIG9 & 00C0)+ (CONFIG10 & 00FF)+(CONFIG11 & 00E0)+ (CONFIG12 & 00FF)+(CONFIG13 & 0040)	0628h	057Eh
	Boot Block 4K words	SUM(0800:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+SUM(C000:FFFF)+ SUM(10000:13FFF)+SUM(14000:17FFF)+SUM(18000:1BFFF)+ SUM(1C000:1FFFF)+(CONFIG0 & 0000)+(CONFIG1 & 00CF)+ (CONFIG2 & 001F)+(CONFIG3 & 0000)+(CONFIG4 & 00F3)+ (CONFIG5 & 0087)+(CONFIG6 & 00F5)+(CONFIG7 & 0000)+ (CONFIG8 & 00FF)+(CONFIG9 & 00C0)+(CONFIG10 & 00FF)+ (CONFIG11 & 00E0)+(CONFIG12 & 00FF)+(CONFIG13 & 0040)+SUM(IDs)	0E60h	0E0Bh
	Boot/ Panel0/ Panel1	SUM(8000:BFFF)+SUM(C000:FFFF)+SUM(10000:13FFF)+ SUM(14000:17FFF)+SUM(18000:1BFFF)+SUM(1C000:1FFFF)+ (CONFIG0 & 0000)+(CONFIG1 & 00CF)+(CONFIG2 & 001F)+ (CONFIG3 & 0000)+(CONFIG4 & 00F3)+(CONFIG5 & 0087)+ (CONFIG6 & 00F5)+(CONFIG7 & 0000)+(CONFIG8 & 00FF)+ (CONFIG9 & 00C0)+(CONFIG10 & 00FF)+(CONFIG11 & 00E0)+ (CONFIG12 & 00FF)+(CONFIG13 & 0040)+SUM(IDs)	865Dh	8608h
	All	(CONFIG0 & 0000)+(CONFIG1 & 00CF)+(CONFIG2 & 001F)+ (CONFIG3 & 0000)+(CONFIG4 & 00F3)+(CONFIG5 & 0087)+ (CONFIG6 & 00F5)+(CONFIG7 & 0000)+(CONFIG8 & 003F)+ (CONFIG9 & 00C0)+(CONFIG10 & 003F)+(CONFIG11 & 00E0)+ (CONFIG12 & 003F)+(CONFIG13 & 0040)+SUM(IDs)	0561h	0561h

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	0xAA at 0 and Max Address
PIC18F8527	None	SUM(0000:07FF)+SUM(0800:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+ (CONFIG0 & 0000)+(CONFIG1 & 00CF)+(CONFIG2 & 001F)+ (CONFIG3 & 001F)+(CONFIG4 & 00F3)+(CONFIG5 & 0087)+ (CONFIG6 & 00F5)+(CONFIG7 & 0000)+(CONFIG8 & 0007)+ (CONFIG9 & 00C0)+(CONFIG10 & 0007)+(CONFIG11 & 00E0)+ (CONFIG12 & 0007)+(CONFIG13 & 0040)	4435h	438Bh
	Boot Block 4K words	SUM(0800:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+(CONFIG0 & 0000)+ (CONFIG1 & 00CF)+(CONFIG2 & 001F)+(CONFIG3 & 001F)+ (CONFIG4 & 00F3)+(CONFIG5 & 0087)+(CONFIG6 & 00F5)+ (CONFIG7 & 0000)+(CONFIG8 & 0007)+(CONFIG9 & 00C0)+ (CONFIG10 & 0007)+(CONFIG11 & 00E0)+(CONFIG12 & 0007)+ (CONFIG13 & 0040)+SUM(IDs)	4C6Dh	4C18h
	Boot/ Panel0/ Panel1	SUM(1C000:1FFFF)+(CONFIG0 & 0000)+(CONFIG1 & 00CF)+ (CONFIG2 & 001F)+(CONFIG3 & 0000)+(CONFIG4 & 00F3)+ (CONFIG5 & 0087)+(CONFIG6 & 00F5)+(CONFIG7 & 0000)+ (CONFIG8 & 00FF)+(CONFIG9 & 00C0)+(CONFIG10 & 00FF)+ (CONFIG11 & 00E0)+(CONFIG12 & 00FF)+(CONFIG13 & 0040)+SUM(IDs)	C46Ah	C415h
	All	(CONFIG0 & 0000)+(CONFIG1 & 00CF)+(CONFIG2 & 001F)+ (CONFIG3 & 001F)+(CONFIG4 & 00F3)+(CONFIG5 & 0087)+ (CONFIG6 & 00F5)+(CONFIG7 & 0000)+(CONFIG8 & 0007)+ (CONFIG9 & 00C0)+(CONFIG10 & 0007)+(CONFIG11 & 00E0)+ (CONFIG12 & 0007)+(CONFIG13 & 0040)+SUM(IDs)	0466h	0466h
PIC18F8622	None	SUM(0000:07FF)+SUM(0800:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+ SUM(C000:FFFF)+(CONFIG0 & 0000)+(CONFIG1 & 00CF): (CONFIG2 & 001F)+(CONFIG3 & 001F)+(CONFIG4 & 00F3)+ (CONFIG5 & 0087)+(CONFIG6 & 00F5)+(CONFIG7 & 0000)+ (CONFIG8 & 000F)+(CONFIG9 & 00C0)+(CONFIG10 & 000F)+ (CONFIG11 & 00E0)+(CONFIG12 & 000F)+(CONFIG13 & 0040)	04DDh	03A3h
	Boot Block 4K words	SUM(0800:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+SUM(C000:FFFF)+ (CONFIG0 & 0000)+(CONFIG1 & 00CF)+(CONFIG2 & 001F)+ (CONFIG3 & 001F)+(CONFIG4 & 00F3)+(CONFIG5 & 0087)+ (CONFIG6 & 00F5)+(CONFIG7 & 0000)+(CONFIG8 & 000F)+ (CONFIG9 & 00C0)+(CONFIG10 & 000F)+(CONFIG11 & 00E0)+ (CONFIG12 & 000F)+(CONFIG13 & 0040)+SUM(IDs)	0C85h	0C30h
	Boot/ Panel0/ Panel1	SUM(8000:BFFF)+SUM(C000:FFFF)+(CONFIG0 & 0000)+ (CONFIG1 & 00CF)+(CONFIG2 & 001F)+(CONFIG3 & 001F)+ (CONFIG4 & 00F3)+(CONFIG5 & 0087)+(CONFIG6 & 00F5)+ (CONFIG7 & 0000)+(CONFIG8 & 000F)+(CONFIG9 & 00C0)+ (CONFIG10 & 000F)+(CONFIG11 & 00E0)+(CONFIG12 & 000F)+ (CONFIG13 & 0040)+SUM(IDs)	8482h	842Dh
	All	(CONFIG0 & 0000)+(CONFIG1 & 00CF)+(CONFIG2 & 001F)+ (CONFIG3 & 001F)+(CONFIG4 & 00F3)+(CONFIG5 & 0087)+ (CONFIG6 & 00F5)+(CONFIG7 & 0000)+(CONFIG8 & 000F)+ (CONFIG9 & 00C0)+(CONFIG10 & 000F)+(CONFIG11 & 00E0)+ (CONFIG12 & 000F)+(CONFIG13 & 0040)+SUM(IDs)	0476h	0476h

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	0xAA at 0 and Max Address
PIC18F8627/ PIC18F8628	None	SUM(0000:07FF)+SUM(0800:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+ SUM(C000:FFFF)+SUM(10000:13FFF)+SUM(14000:17FFF)+ (CONFIG0 & 0000)+(CONFIG1 & 00CF)+(CONFIG2 & 001F)+ (CONFIG3 & 001F)+(CONFIG4 & 00F3)+(CONFIG5 & 0087)+ (CONFIG6 & 00F5)+(CONFIG7 & 0000)+(CONFIG8 & 003F)+ (CONFIG9 & 00C0)+(CONFIG10 & 003F)+(CONFIG11 & 00E0)+ (CONFIG12 & 003F)+(CONFIG13 & 0040)	84DDh	8433h
	Boot Block 4K words	SUM(0800:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+SUM(C000:FFFF)+ SUM(10000:13FFF)+SUM(14000:17FFF)+(CONFIG0 & 0000)+ (CONFIG1 & 00CF)+(CONFIG2 & 001F)+(CONFIG3 & 001F)+ (CONFIG4 & 00F3)+(CONFIG5 & 0087)+(CONFIG6 & 00F5)+ (CONFIG7 & 0000)+(CONFIG8 & 003F)+(CONFIG9 & 00C0)+ (CONFIG10 & 003F)+(CONFIG11 & 00E0)+(CONFIG12 & 003F)+ (CONFIG13 & 0040)+SUM(IDs)	8D15h	8CC0h
	Boot/ Panel0/ Panel1	SUM(8000:BFFF)+SUM(C000:FFFF)+SUM(10000:13FFF)+ SUM(14000:17FFF)+(CONFIG0 & 0000)+(CONFIG1 & 00CF): (CONFIG2 & 001F)+(CONFIG3 & 001F)+(CONFIG4 & 00F3)+ (CONFIG5 & 0087)+(CONFIG6 & 00F5)+(CONFIG7 & 0000)+ (CONFIG8 & 003F)+(CONFIG9 & 00C0)+(CONFIG10 & 003F)+ (CONFIG11 & 00E0)+(CONFIG12 & 003F)+(CONFIG13 & 0040)+SUM(IDs)	0512h	04BDh
	All	(CONFIG0 & 0000)+(CONFIG1 & 00CF)+(CONFIG2 & 001F)+ (CONFIG3 & 001F)+(CONFIG4 & 00F3)+(CONFIG5 & 0087)+ (CONFIG6 & 00F5)+(CONFIG7 & 0000)+(CONFIG8 & 003F)+ (CONFIG9 & 00C0)+(CONFIG10 & 003F)+(CONFIG11 & 00E0)+ (CONFIG12 & 003F)+(CONFIG13 & 0040)+SUM(IDs)	04D6h	04D6h
PIC18F8722/ PIC18F8723	None	SUM(0000:07FF)+SUM(0800:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+ SUM(C000:FFFF)+SUM(10000:13FFF)+SUM(14000:17FFF)+ SUM(18000:1BFFF)+SUM(1C000:1FFFF)+(CONFIG0 & 0000)+ (CONFIG1 & 00CF)+(CONFIG2 & 001F)+(CONFIG3 & 001F)+ (CONFIG4 & 00F3)+(CONFIG5 & 0087)+(CONFIG6 & 00F5)+ (CONFIG7 & 0000)+(CONFIG8 & 00FF)+(CONFIG9 & 00C0)+ (CONFIG10 & 00FF)+(CONFIG11 & 00E0)+ (CONFIG12 & 00FF)+(CONFIG13 & 0040)	071Dh	0673h
	Boot Block 4K words	SUM(0800:3FFF)+SUM(4000:7FFF)+SUM(8000:BFFF)+SUM(C000:FFFF)+ SUM(10000:13FFF)+SUM(14000:17FFF)+SUM(18000:1BFFF)+ SUM(1C000:1FFFF)+(CONFIG0 & 0000)+(CONFIG1 & 00CF)+ (CONFIG2 & 001F)+(CONFIG3 & 001F)+(CONFIG4 & 00F3)+ (CONFIG5 & 0087)+(CONFIG6 & 00F5)+(CONFIG7 & 0000)+ (CONFIG8 & 00FF)+(CONFIG9 & 00C0)+(CONFIG10 & 00FF)+ (CONFIG11 & 00E0)+(CONFIG12 & 00FF)+(CONFIG13 & 0040)+SUM(IDs)	0F55h	0F00h
	Boot/ Panel0/ Panel1	SUM(8000:BFFF)+SUM(C000:FFFF)+SUM(10000:13FFF)+ SUM(14000:17FFF)+SUM(18000:1BFFF)+SUM(1C000:1FFFF)+ (CONFIG0 & 0000)+(CONFIG1 & 00CF)+(CONFIG2 & 001F)+ (CONFIG3 & 001F)+(CONFIG4 & 00F3)+(CONFIG5 & 0087)+ (CONFIG6 & 00F5)+(CONFIG7 & 0000)+(CONFIG8 & 00FF)+ (CONFIG9 & 00C0)+(CONFIG10 & 00FF)+(CONFIG11 & 00E0)+ (CONFIG12 & 00FF)+(CONFIG13 & 0040)+SUM(IDs)	8752h	86FDh
	All	(CONFIG0 & 0000)+(CONFIG1 & 00CF)+(CONFIG2 & 001F)+ (CONFIG3 & 001F)+(CONFIG4 & 00F3)+(CONFIG5 & 0087)+ (CONFIG6 & 00F5)+(CONFIG7 & 0000)+(CONFIG8 & 00FF)+ (CONFIG9 & 00C0)+(CONFIG10 & 00FF)+(CONFIG11 & 00E0)+ (CONFIG12 & 00FF)+(CONFIG13 & 0040)+SUM(IDs)	0656h	0656h

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

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

Standard Operating Conditions
Operating Temperature: 25°C is recommended

Param No.	Sym	Characteristic	Min	Max	Units	Conditions
D110	VIHH	High-Voltage Programming Voltage on MCLR/VPP/RG5	VDD + 4.0	12.5	V	
D110A	VIHL	Low-Voltage Programming Voltage on MCLR/VPP/RG5	2.00	5.50	V	
D111	VDD	Supply Voltage during Programming	2.00	5.50	V	Self-timed
			4.50	5.50	V	Externally timed
			4.50	5.50	V	Bulk Erase operations
D112	IPP	Programming Current on MCLR/VPP/RG5	—	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 (PGD)	—	50	pF	To meet AC specifications
P1	TR	MCLR/VPP/RG5 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
P10	TDLY6	PGC Low Time after Programming (high-voltage discharge time)	100	—	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 MCLR/VPP/RG5 ↑	2	—	μs	
P13	TSET2	VDD ↑ Setup Time to MCLR/VPP/RG5 ↑	100	—	ns	

Note 1: Do not allow excess time when transitioning 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
P14	T_{VALID}	Data Out Valid from PGC ↑	10	—	ns	
P15	$T_{SET}3$	PGM ↑ Setup Time to $\overline{MCLR}$/V$_{PP}$/RG5 ↑	2	—	µs	
P16	$T_{DLY}8$	Delay between Last PGC ↓ and $\overline{MCLR}$/V$_{PP}$/RG5 ↓	0	—	s	
P17	$T_{HLD}3$	$\overline{MCLR}$/V$_{PP}$/RG5 ↓ to V$_{DD}$ ↓	—	100	ns	
P18	$T_{HLD}4$	$\overline{MCLR}$/V$_{PP}$/RG5 ↓ to PGM ↓	0	—	s	

Note 1: Do not allow excess time when transitioning $\overline{MCLR}$ between V$_{IL}$ and V$_{IHH}$; this can cause spurious program executions to occur. The maximum transition time is:
1 T_{CY} + T_{PWRT} (if enabled) + 1024 T_{OSC} (for LP, HS, HS/PLL and XT modes only) +
2 ms (for HS/PLL mode only) + 1.5 µs (for EC mode only)
where T_{CY} is the instruction cycle time, T_{PWRT} is the Power-up Timer period and T_{OSC} is the oscillator period. For specific values, refer to the Electrical Characteristics section of the device data sheet for the particular device.

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

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, rfPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Octopus, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, PIC^{32} logo, REAL ICE, rfLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2009, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

♻ Printed on recycled paper.

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

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

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

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

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4080

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

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

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

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

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

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

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

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

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

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

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

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

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

EUROPE

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

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

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

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

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

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

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

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

03/26/09