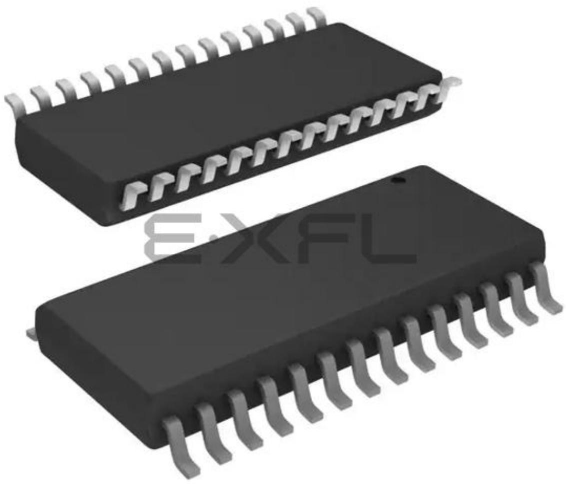


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f2510-i-so

TABLE 2-1: PIN DESCRIPTIONS (DURING PROGRAMMING): PIC18F2XXX/4XXX FAMILY

Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
$\overline{MCLR}$/VPP/RE3	VPP	P	Programming Enable
VDD[2]	VDD	P	Power Supply
VSS[2]	VSS	P	Ground
RB5	PGM	I	Low-Voltage ICSP™ Input when LVP Configuration bit equals '1'[1]
RB6	PGC	I	Serial Clock
RB7	PGD	I/O	Serial Data

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

Note 1: See Figure 5-1 for more information.

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

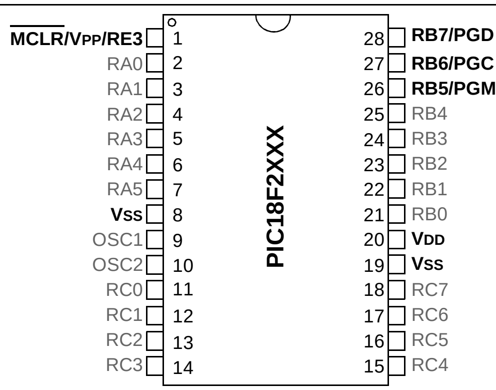
The following devices are included in 28-pin SPDIP, PDIP and SOIC parts:

- PIC18F2221
- PIC18F2321
- PIC18F2410
- PIC18F2420
- PIC18F2423
- PIC18F2450
- PIC18F2455
- PIC18F2458
- PIC18F2480
- PIC18F2510
- PIC18F2515
- PIC18F2520
- PIC18F2523
- PIC18F2525
- PIC18F2550
- PIC18F2553
- PIC18F2580
- PIC18F2585
- PIC18F2610
- PIC18F2620
- PIC18F2680
- PIC18F2682
- PIC18F2685

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

- PIC18F2221
- PIC18F2321

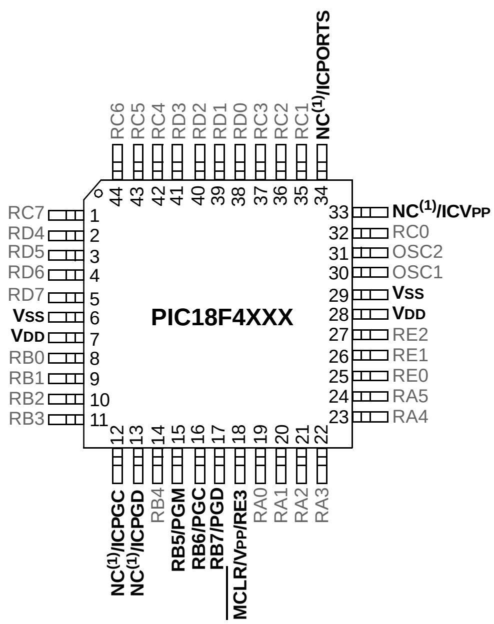
FIGURE 2-1: 28-Pin SPDIP, PDIP, SOIC,SSOP



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

- PIC18F4221
- PIC18F4321
- PIC18F4410
- PIC18F4420
- PIC18F4423
- PIC18F4450
- PIC18F4455
- PIC18F4458
- PIC18F4480
- PIC18F4510
- PIC18F4520
- PIC18F4515
- PIC18F4523
- PIC18F4525
- PIC18F4550
- PIC18F4553
- PIC18F4580
- PIC18F4585
- PIC18F4610
- PIC18F4620
- PIC18F4680
- PIC18F4682
- PIC18F4685

FIGURE 2-4: 44-PIN TQFP



Note 1: These pins are NC (No Connect) for all devices listed above with the exception of the PIC18F4450, PIC18F4455, PIC18F4458 and the PIC18F4553 devices (see **Section 2.8 "Dedicated ICSP/ICD Port (44-Pin TQFP Only)"** for more information on programming these pins in these devices).

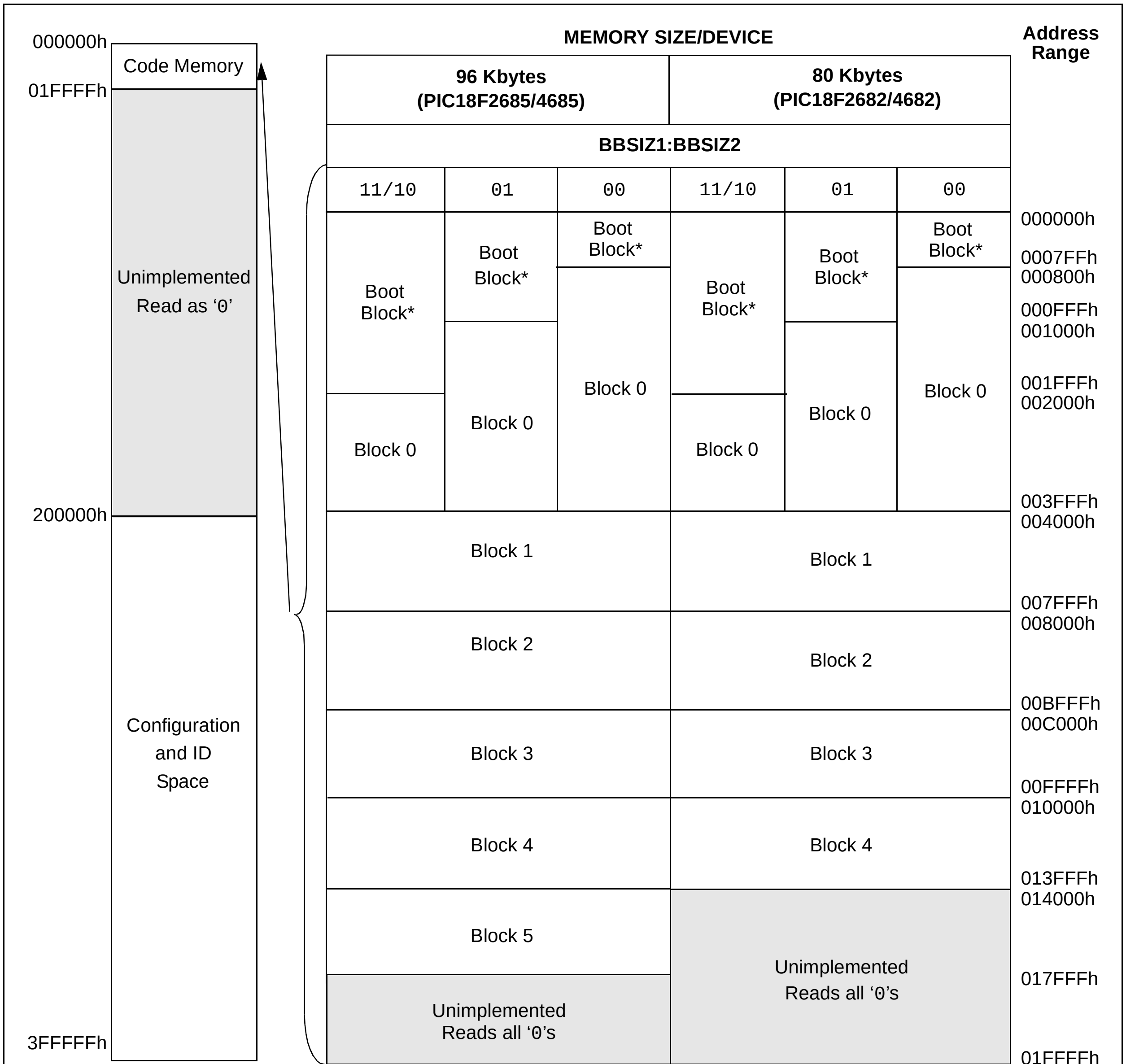
For PIC18F2685/4685 devices, the code memory space extends from 0000h to 017FFFh (96 Kbytes) in five 16-Kbyte blocks. For PIC18F2682/4682 devices, the code memory space extends from 0000h to 0013FFFh (80 Kbytes) in four 16-Kbyte blocks. Addresses, 0000h through 0FFFh, however, define a “Boot Block” region 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 PIC18F2685/4685 and PIC18F2682/4682 devices can be configured as 1, 2 or 4K words (see Figure 2-7). This is done through the BBSIZ<2:1> 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)
PIC18F2682	000000h-013FFFh (80K)
PIC18F4682	
PIC18F2685	000000h-017FFFh (96K)
PIC18F4685	

FIGURE 2-7: MEMORY MAP AND THE CODE MEMORY SPACE FOR PIC18F2685/4685 AND PIC18F2682/4682 DEVICES



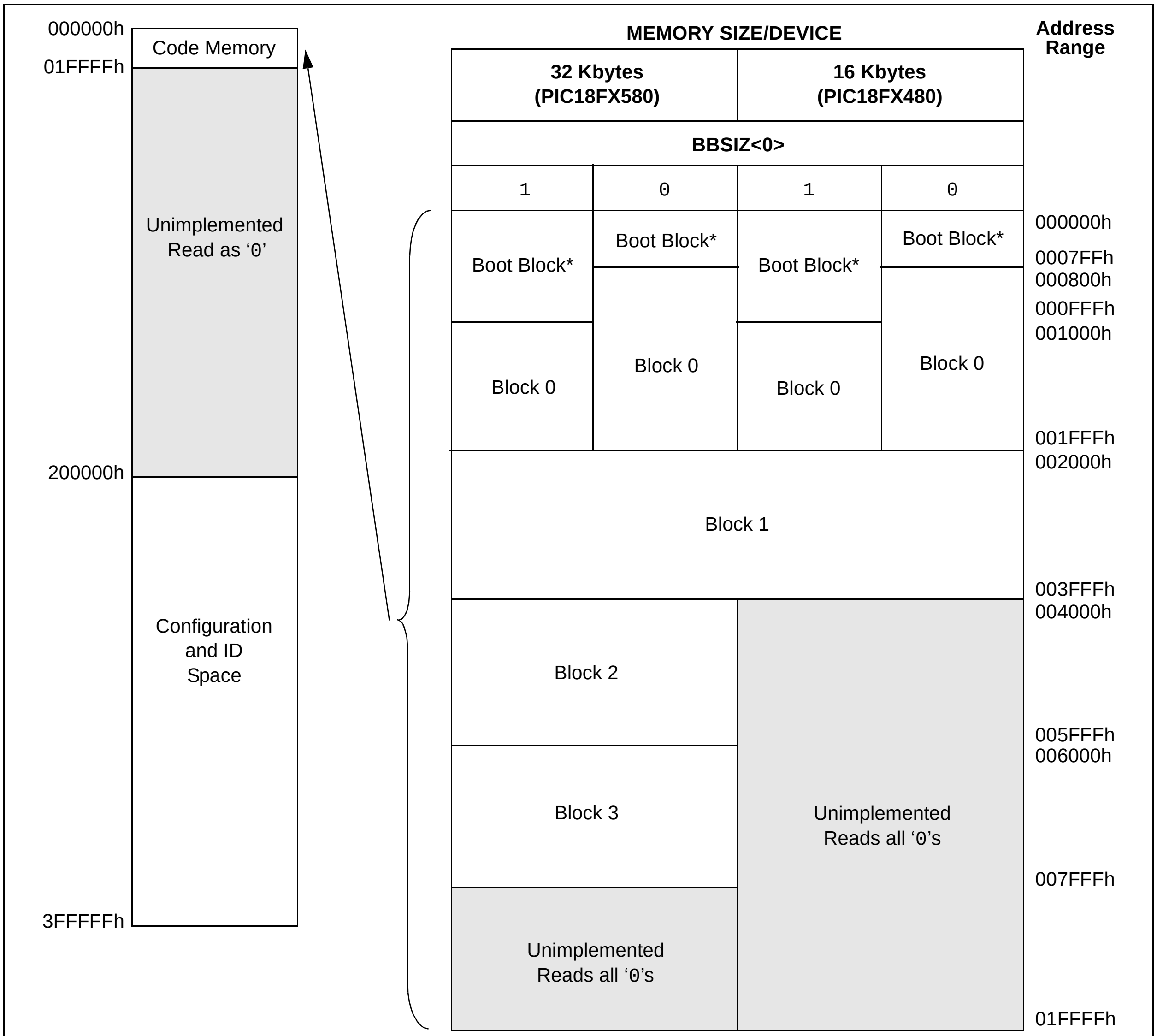
Note: Sizes of memory areas are not to scale.
* Boot Block size is determined by the BBSIZ<1:2> bits in the CONFIG4L register.

For PIC18FX5X0/X5X3 devices, the code memory space extends from 000000h to 007FFFh (32 Kbytes) in four 8-Kbyte blocks. For PIC18FX4X5/X4X8 devices, the code memory space extends from 000000h to 005FFFh (24 Kbytes) in three 8-Kbyte blocks. Addresses, 000000h through 0007FFh, 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.

TABLE 2-6: IMPLEMENTATION OF CODE MEMORY

Device	Code Memory Size (Bytes)
PIC18F2480	000000h-003FFFh (16K)
PIC18F4480	
PIC18F2580	000000h-007FFFh (32K)
PIC18F4580	

FIGURE 2-10: MEMORY MAP AND THE CODE MEMORY SPACE FOR PIC18F2480/2580/4480/4580 DEVICES



Note: Sizes of memory areas are not to scale.

* Boot Block size is determined by the BBSIZ<0> bit in the CONFIG4L register.

For PIC18F2221/4221 devices, the code memory space extends from 0000h to 00FFFh (4 Kbytes) in one 4-Kbyte block. For PIC18F2321/4321 devices, the code memory space extends from 0000h to 01FFFh (8 Kbytes) in two 4-Kbyte blocks. Addresses, 0000h through 07FFh, however, define a variable "Boot Block" region that is treated separately from Block 0. All of these blocks define code protection boundaries within the code memory space.

In addition to the code memory space, there are three blocks that are accessible to the user through Table Reads and Table Writes. Their locations in the memory map are shown in Figure 2-12.

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

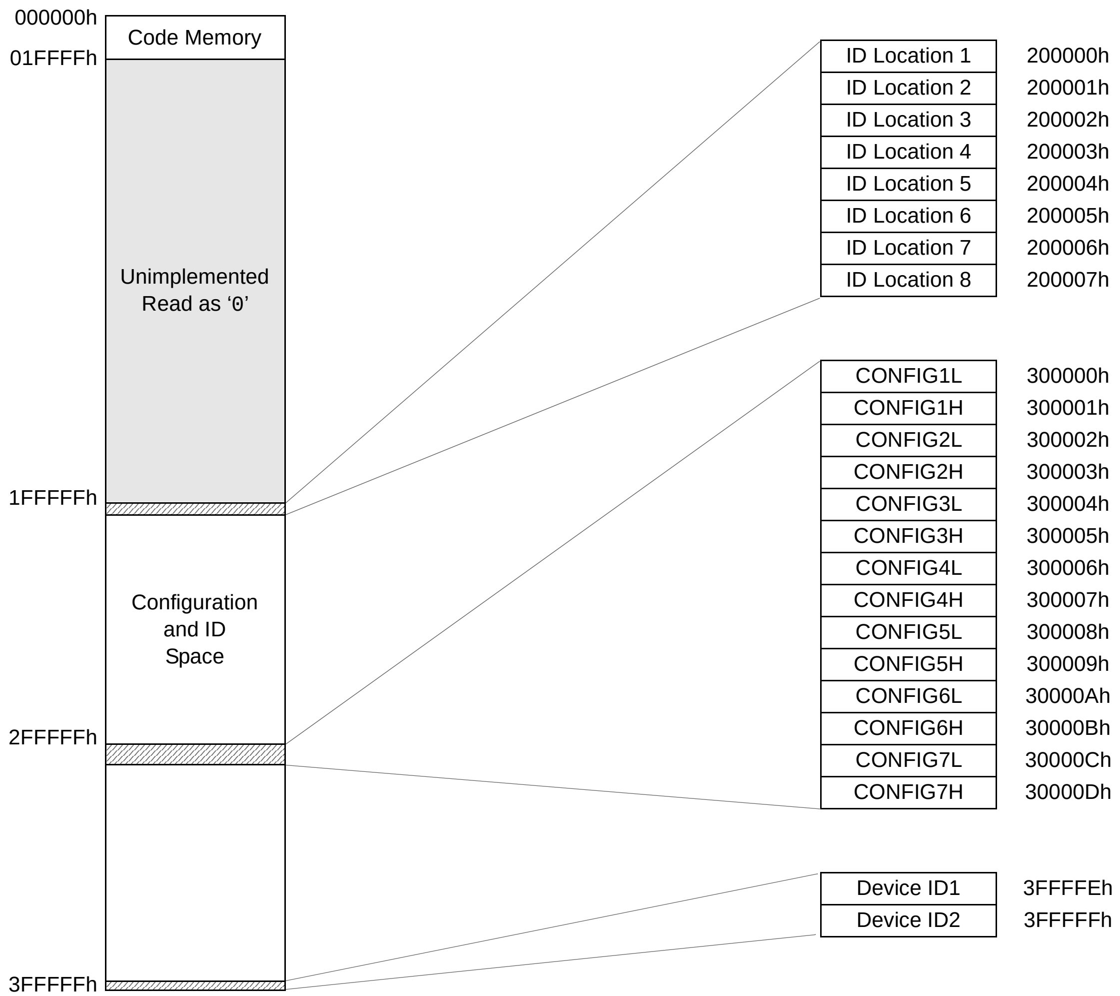
Memory in the address space, 0000000h to 3FFFFFh, is addressed via the Table Pointer register, which is comprised of three pointer registers:

- 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 2-12: CONFIGURATION AND ID LOCATIONS FOR PIC18F2XXX/4XXX FAMILY DEVICES

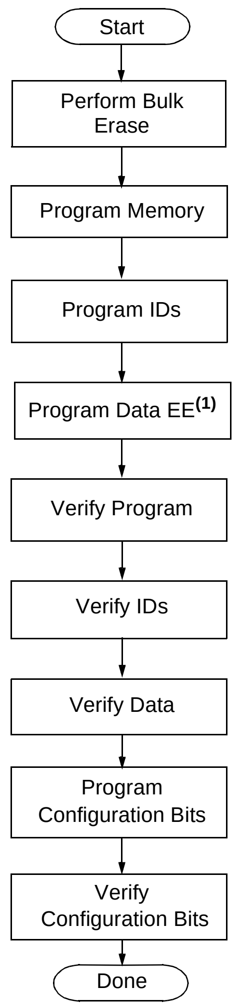


Note: Sizes of memory areas are not to scale.

2.4 High-Level Overview of the Programming Process

Figure 2-13 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 (selected devices only, see **Section 3.3 "Data EEPROM Programming"**). 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 2-13: HIGH-LEVEL PROGRAMMING FLOW



Note 1: Selected devices only, see **Section 3.3 "Data EEPROM Programming"**.

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

As shown in Figure 2-14, the High-Voltage ICSP Program/Verify mode is entered by holding PGC and PGD low and then raising MCLR/VPP/RE3 to VIHH (high voltage). Once in this mode, the code memory, data EEPROM (selected devices only, see **Section 3.3 "Data EEPROM Programming"**), ID locations and Configuration bits can be accessed and programmed in serial fashion. Figure 2-15 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 2-14: ENTERING HIGH-VOLTAGE PROGRAM/VERIFY MODE

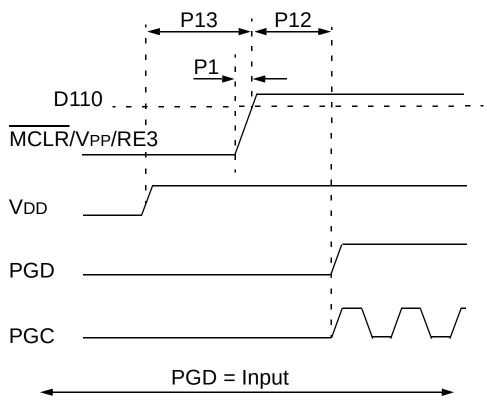


FIGURE 2-15: EXITING HIGH-VOLTAGE PROGRAM/VERIFY MODE

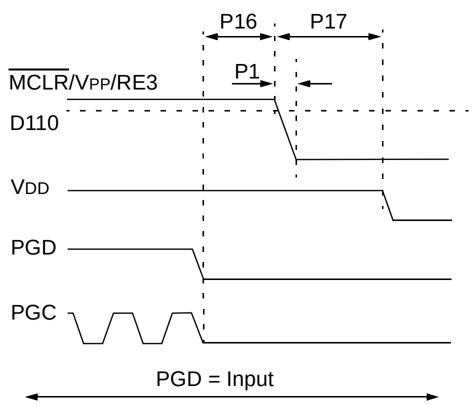
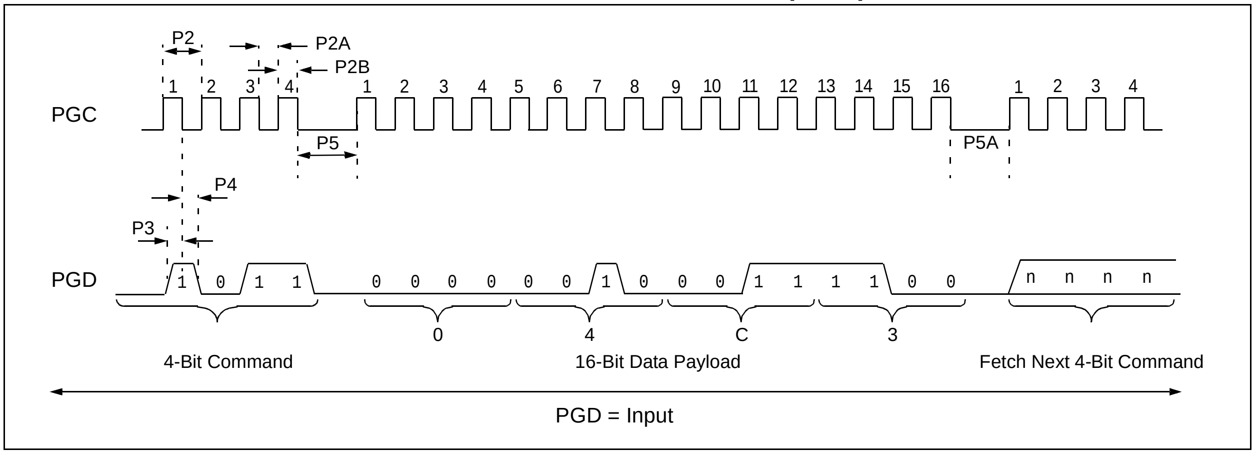


FIGURE 2-18: TABLE WRITE, POST-INCREMENT TIMING (1101)



2.8 Dedicated ICSP/ICD Port (44-Pin TQFP Only)

The PIC18F4455/4458/4550/4553 44-pin TQFP devices are designed to support an alternate programming input: the dedicated ICSP/ICD port. The primary purpose of this port is to provide an alternate In-Circuit Debugging (ICD) option and free the pins (RB6, RB7 and $\overline{MCLR}$) that would normally be used for debugging the application. In conjunction with ICD capability, however, the dedicated ICSP/ICD port also provides an alternate port for ICSP.

Setting the ICPRT Configuration bit enables the dedicated ICSP/ICD port. The dedicated ICSP/ICD port functions the same as the default ICSP/ICD port; however, alternate pins are used instead of the default pins. Table 2-10 identifies the functionally equivalent pins for ICSP purposes:

The dedicated ICSP/ICD port is an alternate port. Thus, ICSP is still available through the default port even though the ICPRT Configuration bit is set. When the VIH is seen on the $\overline{MCLR}$/VPP/RE3 pin prior to applying VIH to the ICRST/ICVPP pin, then the state of the ICRST/ICVPP pin is ignored. Likewise, when the VIH is seen on ICRST/ICVPP prior to applying VIH to $\overline{MCLR}$/VPP/RE3, then the state of the $\overline{MCLR}$/VPP/RE3 pin is ignored.

Note: The ICPRT Configuration bit can only be programmed through the default ICSP port. Chip Erase functions through the dedicated ICSP/ICD port do not affect this bit.

When the ICPRT Configuration bit is set (dedicated ICSP/ICD port enabled), the NC/ICPORTS pin must be tied to either VDD or VSS.

The ICPRT Configuration bit must be maintained clear for all 28-pin and 40-pin devices; otherwise, unexpected operation may occur.

TABLE 2-10: ICSP™ EQUIVALENT PINS

Pin Name	During Programming			
	Pin Name	Pin Type	Dedicated Pins	Pin Description
$\overline{MCLR}$/VPP/RE3	VPP	P	NC/ICRST/ICVPP	Programming Enable
RB6	PGC	I	NC/ICCK/ICPGC	Serial Clock
RB7	PGD	I/O	NC/ICDT/ICPGD	Serial Data

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

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 erase.

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 being 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 OPTIONS

Description	Data (3C0005h:3C0004h)
Chip Erase	3F8Fh
Erase Data EEPROM(1)	0084h
Erase Boot Block	0081h
Erase Configuration 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

Note 1: Selected devices only, see **Section 3.3 "Data EEPROM Programming"**.

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 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.

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 PIC18F2XXX/4XXX Family device is shown in Table 3-5. The flowchart, shown in Figure 3-4, depicts the logic necessary to completely write a PIC18F2XXX/4XXX Family device. The timing diagram that details the Start Programming command and Parameters P9 and P10 is shown in Figure 3-5.

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

TABLE 3-4: WRITE AND ERASE BUFFER SIZES

Devices (Arranged by Family)	Write Buffer Size (Bytes)	Erase Buffer Size (Bytes)
PIC18F2221, PIC18F2321, PIC18F4221, PIC18F4321	8	64
PIC18F2450, PIC18F4450	16	64
PIC18F2410, PIC18F2510, PIC18F4410, PIC18F4510	32	64
PIC18F2420, PIC18F2520, PIC18F4420, PIC18F4520		
PIC18F2423, PIC18F2523, PIC18F4423, PIC18F4523		
PIC18F2480, PIC18F2580, PIC18F4480, PIC18F4580		
PIC18F2455, PIC18F2550, PIC18F4455, PIC18F4550		
PIC18F2458, PIC18F2553, PIC18F4458, PIC18F4553		
PIC18F2515, PIC18F2610, PIC18F4515, PIC18F4610	64	64
PIC18F2525, PIC18F2620, PIC18F4525, PIC18F4620		
PIC18F2585, PIC18F2680, PIC18F4585, PIC18F4680		
PIC18F2682, PIC18F2685, PIC18F4682, PIC18F4685		

3.3 Data EEPROM Programming

Note: Data EEPROM programming is **not** available on the following devices:	
PIC18F2410	PIC18F4410
PIC18F2450	PIC18F4450
PIC18F2510	PIC18F4510
PIC18F2515	PIC18F4515
PIC18F2610	PIC18F4610

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

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

The write begins on the falling edge of the 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 3-6: PROGRAM DATA FLOW

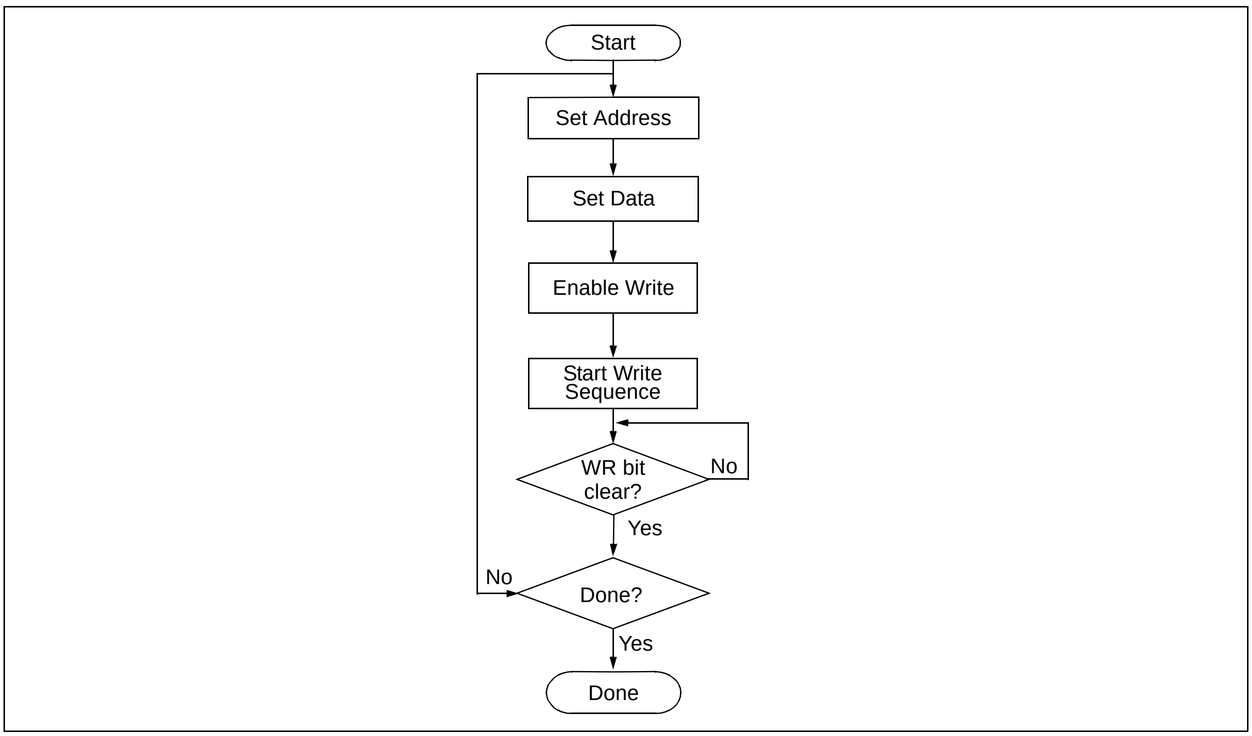


TABLE 3-7: PROGRAMMING DATA MEMORY

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.

TABLE 5-1: CONFIGURATION BITS AND DEVICE IDS

<table>
<tr><th colspan="2">File Name</th><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th><th>Default/ Unprogrammed Value</th></tr>
<tr><td>300000h[1,8]</td><td>CONFIG1L</td><td>—</td><td>—</td><td>USBDIV</td><td>CPUDIV1</td><td>CPUDIV0</td><td>PLLDIV2</td><td>PLLDIV1</td><td>PLLDIV0</td><td>--00 0000</td></tr>
<tr><td rowspan="2">300001h</td><td rowspan="2">CONFIG1H</td><td rowspan="2">IESO</td><td rowspan="2">FCMEN</td><td rowspan="2">—</td><td rowspan="2">—</td><td rowspan="2">FOSC3</td><td rowspan="2">FOSC2</td><td rowspan="2">FOSC1</td><td rowspan="2">FOSC0</td><td>00-- 0111</td></tr>
<tr><td>00-- 0101[1,8]</td></tr>
<tr><td rowspan="2">300002h</td><td rowspan="2">CONFIG2L</td><td rowspan="2">—</td><td rowspan="2">—</td><td>—</td><td rowspan="2">BORV1</td><td rowspan="2">BORV0</td><td rowspan="2">BOREN1</td><td rowspan="2">BOREN0</td><td rowspan="2">$\overline{\text{PWRTEN}}$</td><td>---1 1111</td></tr>
<tr><td>VREGEN[1,8]</td><td>--01 1111[1,8]</td></tr>
<tr><td>300003h</td><td>CONFIG2H</td><td>—</td><td>—</td><td>—</td><td>WDTPS3</td><td>WDTPS2</td><td>WDTPS1</td><td>WDTPS0</td><td>WDTEN</td><td>---1 1111</td></tr>
<tr><td rowspan="2">300005h</td><td rowspan="2">CONFIG3H</td><td rowspan="2">MCLRE</td><td rowspan="2">—</td><td rowspan="2">—</td><td rowspan="2">—</td><td rowspan="2">—</td><td rowspan="2">LPT1OSC</td><td rowspan="2">PBADEN</td><td>CCP2MX[7]</td><td>1--- -011[7]</td></tr>
<tr><td>—</td><td>1--- -01-</td></tr>
<tr><td rowspan="5">300006h</td><td rowspan="5">CONFIG4L</td><td rowspan="5">$\overline{\text{DEBUG}}$</td><td rowspan="5">XINST</td><td>ICPRT[1]</td><td>—</td><td>—</td><td rowspan="5">LVP</td><td rowspan="5">—</td><td rowspan="5">STVREN</td><td>100- -1-1[1]</td></tr>
<tr><td>BBSIZ1</td><td>BBSIZ0</td><td>—</td><td>1000 -1-1</td></tr>
<tr><td>—</td><td>BBSIZ[3]</td><td>—</td><td>10-0 -1-1[3]</td></tr>
<tr><td>ICPRT[8]</td><td>—</td><td>BBSIZ[8]</td><td>100- 01-1[8]</td></tr>
<tr><td>BBSIZ1[2]</td><td>BBSIZ2[2]</td><td>—</td><td>1000 -1-1[2]</td></tr>
<tr><td>300008h</td><td>CONFIG5L</td><td>—</td><td>—</td><td>CP5[10]</td><td>CP4[9]</td><td>CP3[4]</td><td>CP2[4]</td><td>CP1</td><td>CP0</td><td>--11 1111</td></tr>
<tr><td>300009h</td><td>CONFIG5H</td><td>CPD</td><td>CPB</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>11-- ----</td></tr>
<tr><td>30000Ah</td><td>CONFIG6L</td><td>—</td><td>—</td><td>WRT5[10]</td><td>WRT4[9]</td><td>WRT3[4]</td><td>WRT2[4]</td><td>WRT1</td><td>WRT0</td><td>--11 1111</td></tr>
<tr><td>30000Bh</td><td>CONFIG6H</td><td>WRTD</td><td>WRTB</td><td>WRTC[5]</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>111- ----</td></tr>
<tr><td>30000Ch</td><td>CONFIG7L</td><td>—</td><td>—</td><td>EBTR5[10]</td><td>EBTR4[9]</td><td>EBTR3[4]</td><td>EBTR2[4]</td><td>EBTR1</td><td>EBTR0</td><td>--11 1111</td></tr>
<tr><td>30000Dh</td><td>CONFIG7H</td><td>—</td><td>EBTRB</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>-1-- ----</td></tr>
<tr><td>3FFFFEh</td><td>DEVID1[6]</td><td>DEV2</td><td>DEV1</td><td>DEV0</td><td>REV4</td><td>REV3</td><td>REV2</td><td>REV1</td><td>REV0</td><td>See Table 5-2</td></tr>
<tr><td>3FFFFFh</td><td>DEVID2[6]</td><td>DEV10</td><td>DEV9</td><td>DEV8</td><td>DEV7</td><td>DEV6</td><td>DEV5</td><td>DEV4</td><td>DEV3</td><td>See Table 5-2</td></tr>
</table>

Legend: - = unimplemented. Shaded cells are unimplemented, read as '0'.

Note 1: Implemented only on PIC18F2455/2550/4455/4550 and PIC18F2458/2553/4458/4553 devices.

2: Implemented on PIC18F2585/2680/4585/4680, PIC18F2682/2685 and PIC18F4682/4685 devices only.

3: Implemented on PIC18F2480/2580/4480/4580 devices only.

4: These bits are only implemented on specific devices based on available memory. Refer to **Section 2.3 "Memory Maps"**.

5: In PIC18F2480/2580/4480/4580 devices, this bit is read-only in Normal Execution mode; it can be written only in Program mode.

6: DEVID registers are read-only and cannot be programmed by the user.

7: Implemented on all devices with the exception of the PIC18FXX8X and PIC18F2450/4450 devices.

8: Implemented on PIC18F2450/4450 devices only.

9: Implemented on PIC18F2682/2685 and PIC18F4682/4685 devices only.

10: Implemented on PIC18F2685/4685 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/RE3 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/RE3 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 PIC18F2XXX/4XXX Family 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 PIC18F2XXX/4XXX Family 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 Words, appropriately masked
- ID locations (if any block is code-protected)

The Least Significant 16 bits of this sum is the checksum. The contents of the data EEPROM are not used.

5.6.1 PROGRAM MEMORY

When program memory contents are summed, each 16-bit word is added to the checksum. The contents of program memory, from 000000h to the end of the last program memory block, are used for this calculation. Overflows from bit 15 may be ignored.

5.6.2 CONFIGURATION WORDS

For checksum calculations, unimplemented bits in Configuration Words should be ignored as such bits always read back as '1's. Each 8-bit Configuration Word is ANDed with a corresponding mask to prevent unused bits from affecting checksum calculations.

The mask contains a '0' in unimplemented bit positions, or a '1' where a choice can be made. When ANDed with the value read out of a Configuration Word, only implemented bits remain. A list of suitable masks is provided in Table 5-5.

TABLE 5-4: DEVICE BLOCK LOCATIONS AND SIZES

Device	Memory Size (Bytes)	Pins	Ending Address							Size (Bytes)			
			Boot Block	Block 0	Block 1	Block 2	Block 3	Block 4	Block 5	Boot Block	Block 0	Remaining Blocks	Device Total
PIC18F2221	4K	28	0001FF	0007FF	000FFF	—	—	—	—	512	1536	2048	4096
			0003FF							1024	1024		
PIC18F2321	8K	28	0001FF	000FFF	001FFF	—	—	—	—	512	3584	4096	8192
			0003FF							1024	3072		
			0007FF							2048	2048		
PIC18F2410	16K	28	0007FF	001FFF	003FFF	—	—	—	—	2048	6144	8192	16384
PIC18F2420	16K	28	0007FF	001FFF	003FFF	—	—	—	—	2048	6144	8192	16384
PIC18F2423	16K	28	0007FF	001FFF	003FFF	—	—	—	—	2048	6144	8192	16384
PIC18F2450	16K	28	0007FF	001FFF	003FFF	—	—	—	—	2048	6144	8192	16384
			000FFF							4096	4096		
PIC18F2455	24K	28	0007FF	001FFF	003FFF	005FFF	—	—	—	2048	6144	16384	24576
PIC18F2458	24K	28	0007FF	001FFF	003FFF	005FFF	—	—	—	2048	6144	16384	24576
PIC18F2480	16K	28	0007FF	001FFF	003FFF	—	—	—	—	2048	6144	8192	16384
			000FFF							4096	4096		
PIC18F2510	32K	28	0007FF	001FFF	003FFF	005FFF	007FFF	—	—	2048	6144	24576	32768
PIC18F2515	48K	28	0007FF	003FFF	007FFF	00BFFF	—	—	—	2048	14336	32768	49152
PIC18F2520	32K	28	0007FF	001FFF	003FFF	005FFF	007FFF	—	—	2048	14336	16384	32768
PIC18F2523	32K	28	0007FF	001FFF	003FFF	005FFF	007FFF	—	—	2048	14336	16384	32768
PIC18F2525	48K	28	0007FF	003FFF	007FFF	00BFFF	—	—	—	2048	14336	32768	49152
PIC18F2550	32K	28	0007FF	001FFF	003FFF	005FFF	007FFF	—	—	2048	6144	24576	32768
PIC18F2553	32K	28	0007FF	001FFF	003FFF	005FFF	007FFF	—	—	2048	6144	24576	32768
PIC18F2580	32K	28	0007FF	001FFF	003FFF	005FFF	007FFF	—	—	2048	6144	24576	32768
			000FFF							4096	4096		
PIC18F2585	48K	28	0007FF	003FFF	007FFF	00BFFF	—	—	—	2048	14336	32768	49152
			000FFF							4096	12288		
			001FFF							8192	8192		
PIC18F2610	64K	28	0007FF	003FFF	007FFF	00BFFF	00FFFF	—	—	2048	14336	49152	65536
PIC18F2620	64K	28	0007FF	003FFF	007FFF	00BFFF	00FFFF	—	—	2048	14336	49152	65536
PIC18F2680	64K	28	0007FF	003FFF	007FFF	00BFFF	00FFFF	—	—	2048	14336	49152	65536
			000FFF							4096	12288		
			001FFF							8192	8192		
PIC18F2682	80K	28	0007FF	003FFF	007FFF	00BFFF	00FFFF	013FFF	—	2048	14336	65536	81920
			000FFF							4096	12288		
			001FFF							8192	8192		
PIC18F2685	96K	28	0007FF	003FFF	007FFF	00BFFF	00FFFF	013FFF	017FFF	2048	14336	81920	98304
			000FFF							4096	12288		
			001FFF							8192	8192		
PIC18F4221	4K	40	0001FF	0007FF	000FFF	—	—	—	—	512	1536	2048	4096
			0003FF							1024	1024		
PIC18F4321	8K	40	0001FF	000FFF	001FFF	—	—	—	—	512	3584	4096	8192
			0003FF							1024	3072		
			0007FF							2048	2048		
PIC18F4410	16K	40	0007FF	001FFF	003FFF	—	—	—	—	2048	6144	8192	16384
PIC18F4420	16K	40	0007FF	001FFF	003FFF	—	—	—	—	2048	6144	8192	16384
PIC18F4423	16K	40	0007FF	001FFF	003FFF	—	—	—	—	2048	6144	8192	16384
PIC18F4450	16K	40	0007FF	001FFF	003FFF	—	—	—	—	2048	6144	8192	16384
			000FFF							4096	4096		

Legend: — = unimplemented.

TABLE 5-4: DEVICE BLOCK LOCATIONS AND SIZES (CONTINUED)

Device	Memory Size (Bytes)	Pins	Ending Address							Size (Bytes)			
			Boot Block	Block 0	Block 1	Block 2	Block 3	Block 4	Block 5	Boot Block	Block 0	Remaining Blocks	Device Total
PIC18F4455	24K	40	0007FF	001FFF	003FFF	005FFF	—	—	—	2048	6144	16384	24576
PIC18F4458	24K	40	0007FF	001FFF	003FFF	005FFF	—	—	—	2048	6144	16384	24576
PIC18F4480	16K	40	0007FF	001FFF	003FFF	—	—	—	—	2048	6144	8192	16384
			000FFF							4096	4096		
PIC18F4510	32K	40	0007FF	001FFF	003FFF	005FFF	007FFF	—	—	2048	6144	24576	32768
PIC18F4515	48K	40	0007FF	003FFF	007FFF	00BFFF	—	—	—	2048	14336	32768	49152
PIC18F4520	32K	40	0007FF	001FFF	003FFF	005FFF	007FFF	—	—	2048	14336	16384	32768
PIC18F4523	32K	40	0007FF	001FFF	003FFF	005FFF	007FFF	—	—	2048	14336	16384	32768
PIC18F4525	48K	40	0007FF	003FFF	007FFF	00BFFF	—	—	—	2048	14336	32768	49152
PIC18F4550	32K	40	0007FF	001FFF	003FFF	005FFF	007FFF	—	—	2048	6144	24576	32768
PIC18F4553	32K	40	0007FF	001FFF	003FFF	005FFF	007FFF	—	—	2048	6144	24576	32768
PIC18F4580	32K	40	0007FF	001FFF	003FFF	005FFF	007FFF	—	—	2048	6144	24576	32768
			000FFF							4096	4096		
PIC18F4585	48K	40	0007FF	003FFF	007FFF	00BFFF	—	—	—	2048	14336	32768	49152
			000FFF							4096	12288		
			001FFF							8192	8192		
PIC18F4610	64K	40	0007FF	003FFF	007FFF	00BFFF	00FFFF	—	—	2048	14336	49152	65536
PIC18F4620	64K	40	0007FF	003FFF	007FFF	00BFFF	00FFFF	—	—	2048	14336	49152	65536
PIC18F4680	64K	40	0007FF	003FFF	007FFF	00BFFF	00FFFF	—	—	2048	14336	49152	65536
			000FFF							4096	12288		
			001FFF							8192	8192		
PIC18F4682	80K	40	0007FF	003FFF	007FFF	00BFFF	00FFFF	013FFF	—	2048	14336	65536	81920
			000FFF							4096	12288		
			001FFF							8192	8192		
PIC18F4685	96K	44	0007FF	003FFF	007FFF	00BFFF	00FFFF	013FFF	017FFF	2048	14336	81920	98304
			000FFF							4096	12288		
			001FFF							8192	8192		

Legend: — = unimplemented.

TABLE 5-5: CONFIGURATION WORD MASKS FOR COMPUTING CHECKSUMS

Device	Configuration Word (CONFIGxx)													
	1L	1H	2L	2H	3L	3H	4L	4H	5L	5H	6L	6H	7L	7H
	Address (30000xh)													
	0h	1h	2h	3h	4h	5h	6h	7h	8h	9h	Ah	Bh	Ch	Dh
PIC18F2221	00	CF	1F	1F	00	87	F5	00	03	C0	03	E0	03	40
PIC18F2321	00	CF	1F	1F	00	87	F5	00	03	C0	03	E0	03	40
PIC18F2410	00	CF	1F	1F	00	87	C5	00	03	C0	03	E0	03	40
PIC18F2420	00	CF	1F	1F	00	87	C5	00	03	C0	03	E0	03	40
PIC18F2423	00	CF	1F	1F	00	87	C5	00	03	C0	03	E0	03	40
PIC18F2450	3F	CF	3F	1F	00	86	ED	00	03	40	03	60	03	40
PIC18F2455	3F	CF	3F	1F	00	87	E5	00	07	C0	07	E0	07	40
PIC18F2458	3F	CF	3F	1F	00	87	E5	00	07	C0	07	E0	07	40
PIC18F2480	00	CF	1F	1F	00	86	D5	00	03	C0	03	E0	03	40
PIC18F2510	00	1F	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F2515	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F2520	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F2523	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F2525	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F2550	3F	CF	3F	1F	00	87	E5	00	0F	C0	0F	E0	0F	40
PIC18F2553	3F	CF	3F	1F	00	87	E5	00	0F	C0	0F	E0	0F	40
PIC18F2580	00	CF	1F	1F	00	86	D5	00	0F	C0	0F	E0	0F	40
PIC18F2585	00	CF	1F	1F	00	86	C5	00	0F	C0	0F	E0	0F	40
PIC18F2610	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F2620	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F2680	00	CF	1F	1F	00	86	C5	00	0F	C0	0F	E0	0F	40
PIC18F2682	00	CF	1F	1F	00	86	C5	00	3F	C0	3F	E0	3F	40
PIC18F2685	00	CF	1F	1F	00	86	C5	00	3F	C0	3F	E0	3F	40
PIC18F4221	00	CF	1F	1F	00	87	F5	00	03	C0	03	E0	03	40
PIC18F4321	00	CF	1F	1F	00	87	F5	00	03	C0	03	E0	03	40
PIC18F4410	00	CF	1F	1F	00	87	C5	00	03	C0	03	E0	03	40
PIC18F4420	00	CF	1F	1F	00	87	C5	00	03	C0	03	E0	03	40
PIC18F4423	00	CF	1F	1F	00	87	C5	00	03	C0	03	E0	03	40
PIC18F4450	3F	CF	3F	1F	00	86	ED	00	03	40	03	60	03	40
PIC18F4455	3F	CF	3F	1F	00	87	E5	00	07	C0	07	E0	07	40
PIC18F4458	3F	CF	3F	1F	00	87	E5	00	07	C0	07	E0	07	40
PIC18F4480	00	CF	1F	1F	00	86	D5	00	03	C0	03	E0	03	40
PIC18F4510	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F4515	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F4520	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F4523	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F4525	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F4550	3F	CF	3F	1F	00	87	E5	00	0F	C0	0F	E0	0F	40
PIC18F4553	3F	CF	3F	1F	00	87	E5	00	0F	C0	0F	E0	0F	40
PIC18F4580	00	CF	1F	1F	00	86	D5	00	0F	C0	0F	E0	0F	40
PIC18F4585	00	CF	1F	1F	00	86	C5	00	0F	C0	0F	E0	0F	40
PIC18F4610	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40

Legend: Shaded cells are unimplemented.

TABLE 5-5: CONFIGURATION WORD MASKS FOR COMPUTING CHECKSUMS (CONTINUED)

Device	Configuration Word (CONFIGxx)													
	1L	1H	2L	2H	3L	3H	4L	4H	5L	5H	6L	6H	7L	7H
	Address (30000xh)													
	0h	1h	2h	3h	4h	5h	6h	7h	8h	9h	Ah	Bh	Ch	Dh
PIC18F4620	00	CF	1F	1F	00	87	C5	00	0F	C0	0F	E0	0F	40
PIC18F4680	00	CF	1F	1F	00	86	C5	00	0F	C0	0F	E0	0F	40
PIC18F4682	00	CF	1F	1F	00	86	C5	00	3F	C0	3F	E0	3F	40
PIC18F4685	00	CF	1F	1F	00	86	C5	00	3F	C0	3F	E0	3F	40

Legend: Shaded cells are unimplemented.