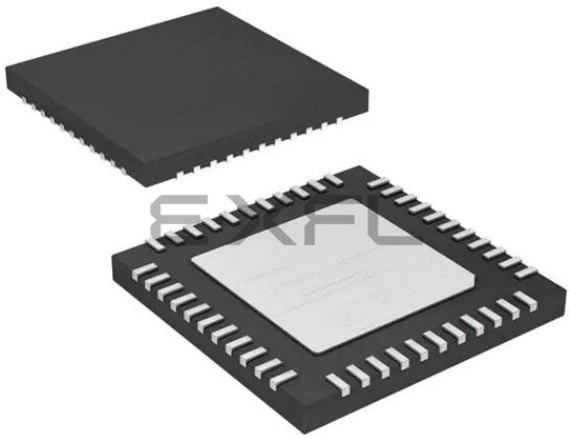


Welcome to E-XFL.COM

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	36
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f4410-i-ml

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.

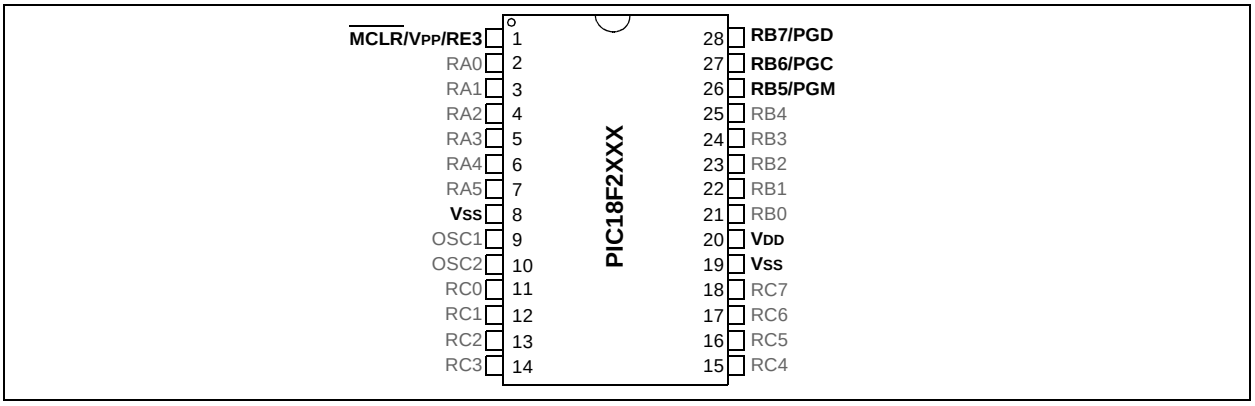
The following devices are included in 28-pin SPDIP, PDIP and SOIC parts:

- 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

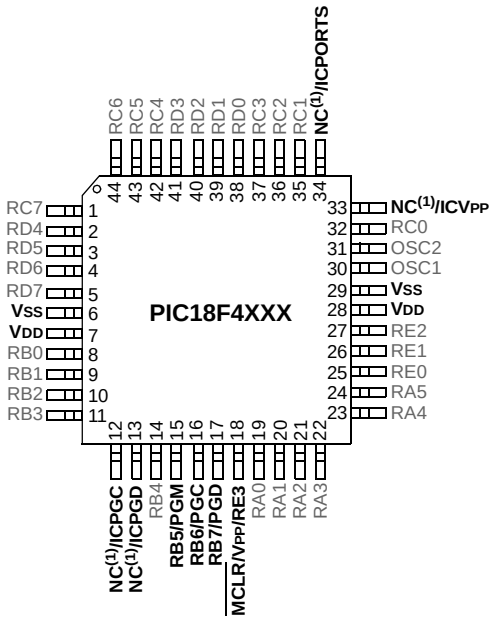
FIGURE 2-1: 28-Pin SPDIP, PDIP, SOIC,SSOP



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

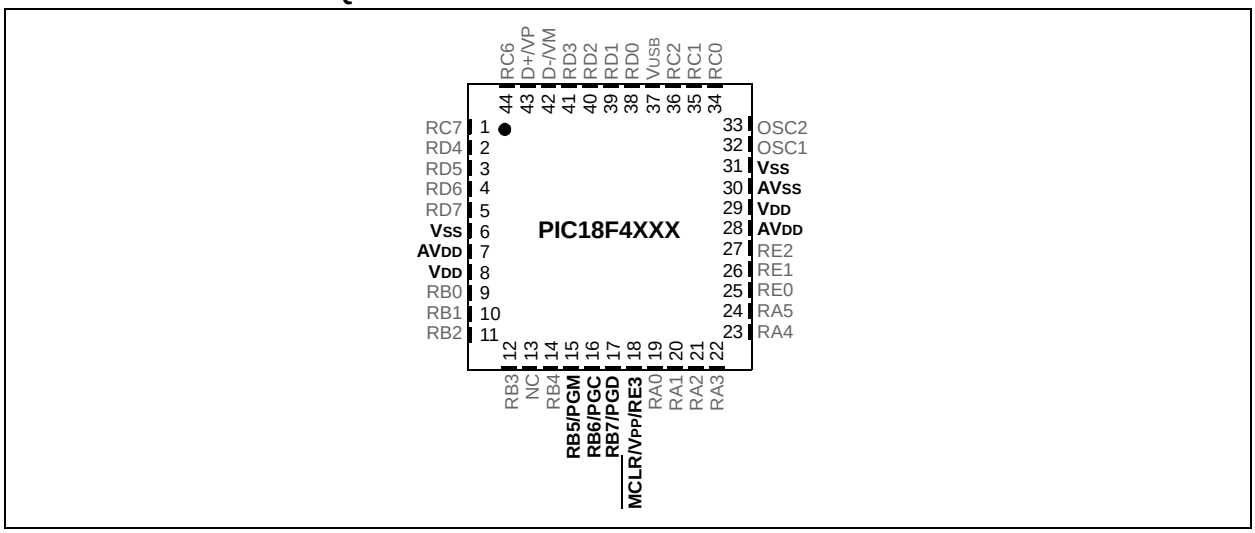


Note 1: These pins are NC (No Connect) for all devices listed above with the exception of the PIC18F4450, PIC18F4455, PIC18F4458 and the PIC18F4553 devices (see **Section 2.8 "Dedicated ICSP/ICD Port (44-Pin TQFP Only)"** for more information on programming these pins in these devices).

The following devices are included in 44-pin QFN 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 2-5: 44-PIN QFN



2.3 Memory Maps

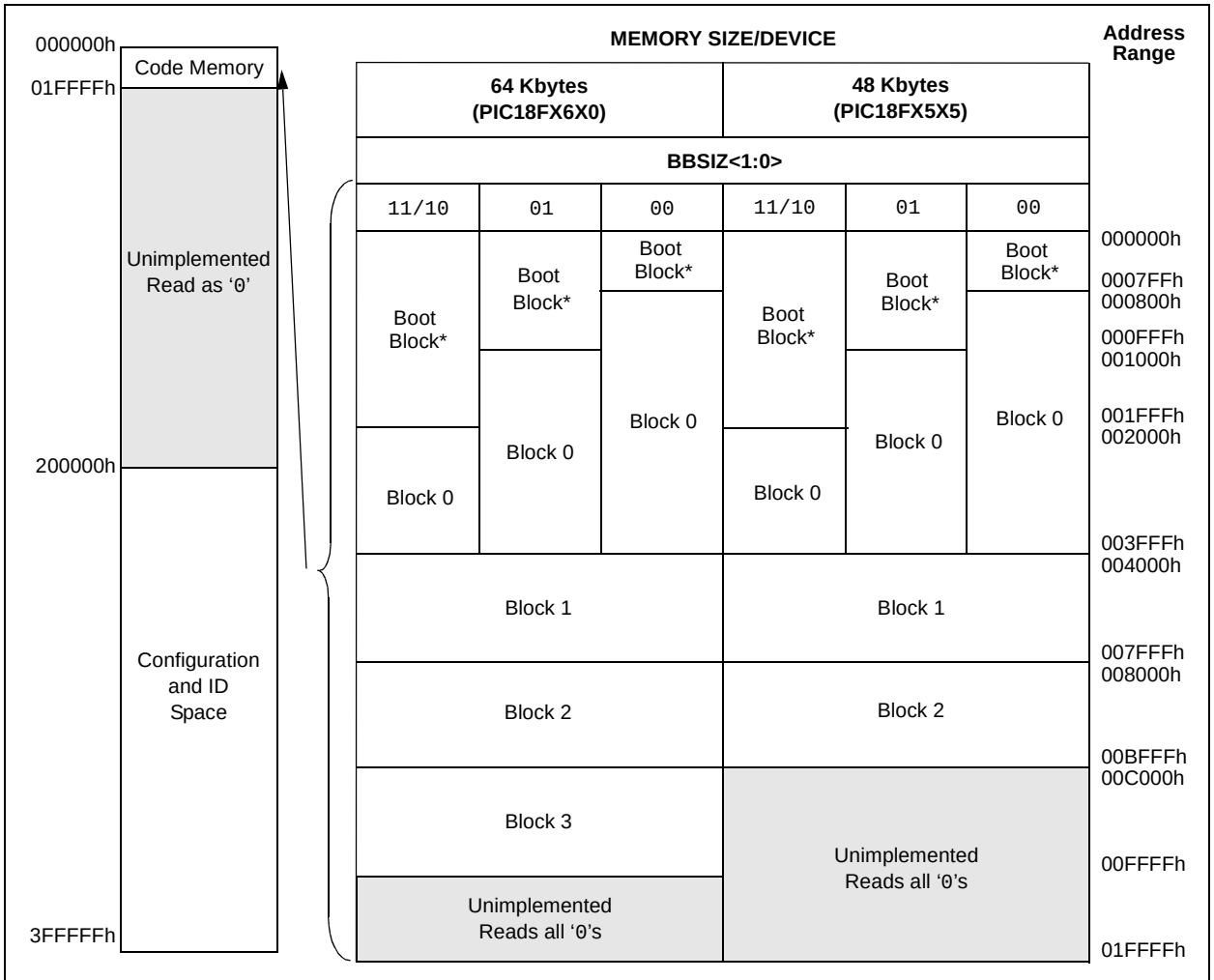
For PIC18FX6X0 devices, the code memory space extends from 0000h to 0FFFFh (64 Kbytes) in four 16-Kbyte blocks. For PIC18FX5X5 devices, the code memory space extends from 0000h to 0BFFFFh (48 Kbytes) in three 16-Kbyte blocks. Addresses, 0000h through 07FFh, however, define a “Boot Block” region that 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 PIC18F2585/2680/4585/4680 devices can be configured as 1, 2 or 4K words (see Figure 2-6). 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)
PIC18F2515	000000h-00BFFFh (48K)
PIC18F2525	
PIC18F2585	
PIC18F4515	
PIC18F4525	
PIC18F4585	
PIC18F2610	000000h-00FFFFh (64K)
PIC18F2620	
PIC18F2680	
PIC18F4610	
PIC18F4620	
PIC18F4680	

FIGURE 2-6: MEMORY MAP AND THE CODE MEMORY SPACE FOR PIC18FX5X5/X6X0 DEVICES



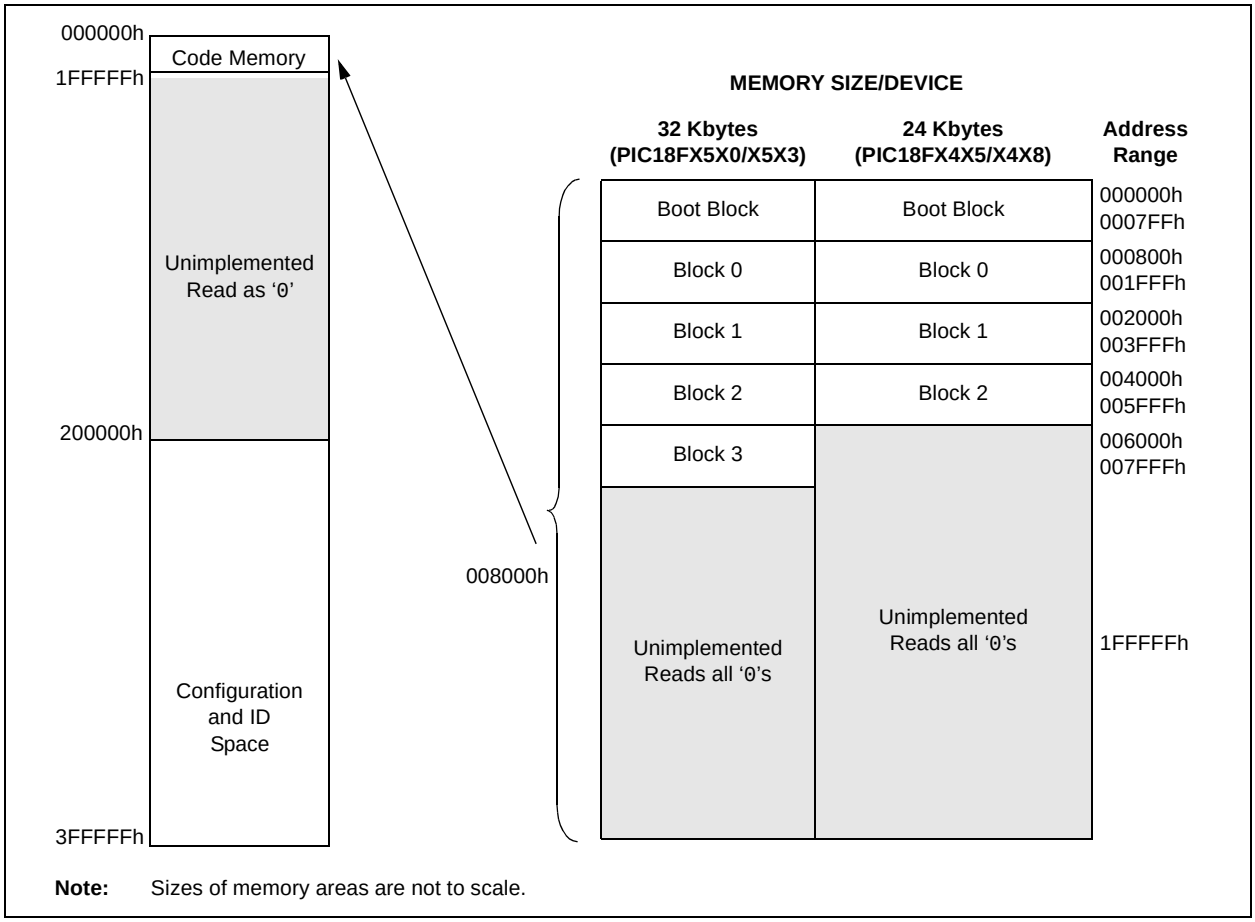
Note: Sizes of memory areas are not to scale.

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

TABLE 2-4: IMPLEMENTATION OF CODE MEMORY

Device	Code Memory Size (Bytes)
PIC18F2455	000000h-005FFFh (24K)
PIC18F2458	
PIC18F4455	
PIC18F4458	
PIC18F2510	000000h-007FFFh (32K)
PIC18F2520	
PIC18F2523	
PIC18F2550	
PIC18F2553	
PIC18F4510	
PIC18F4520	
PIC18F4523	
PIC18F4550	
PIC18F4553	

FIGURE 2-8: MEMORY MAP AND THE CODE MEMORY SPACE FOR PIC18FX4X5/X4X8/X5X0/X5X3 DEVICES



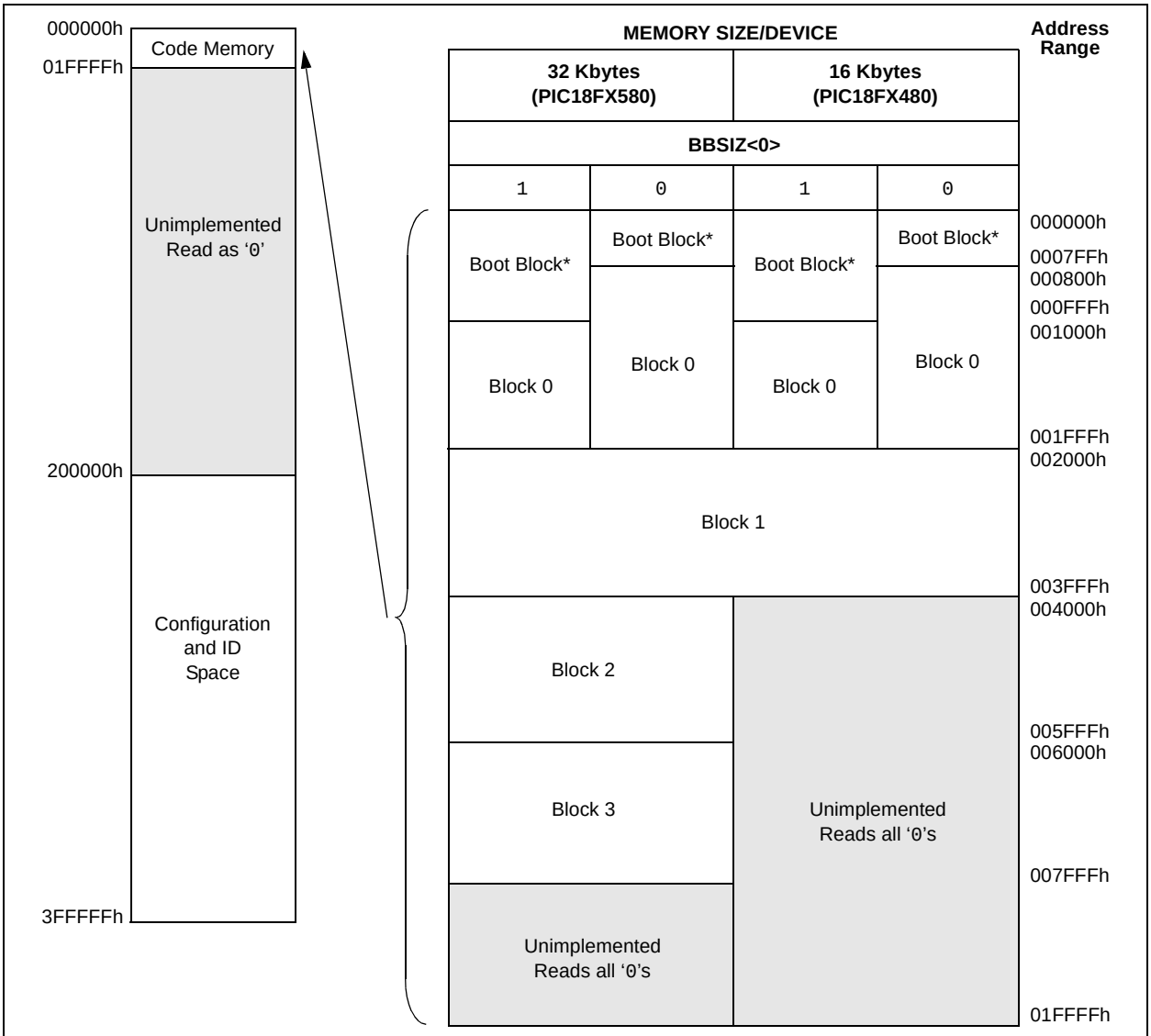
Note: Sizes of memory areas are not to scale.

For PIC18FX4X0/X4X3 devices, the code memory space extends from 000000h to 003FFFh (16 Kbytes) in two 8-Kbyte blocks. Addresses, 000000h through 0003FFh, 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.

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 $\overline{MCLR}$/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

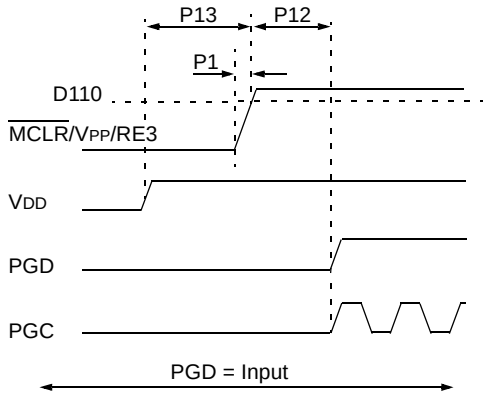
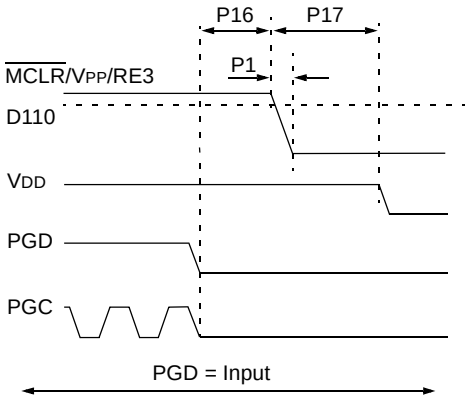


FIGURE 2-15: EXITING HIGH-VOLTAGE PROGRAM/VERIFY MODE



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		

TABLE 3-5: WRITE 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
Step 2: Load write buffer.		
0000	0E <Addr[21:16]>	MOVLW <Addr[21:16]>
0000	6E F8	MOVWF TBLPTRU
0000	0E <Addr[15:8]>	MOVLW <Addr[15:8]>
0000	6E F7	MOVWF TBLPTRH
0000	0E <Addr[7:0]>	MOVLW <Addr[7:0]>
0000	6E F6	MOVWF TBLPTRL
Step 3: Repeat for all but the last two bytes.		
1101	<MSB><LSB>	Write 2 bytes and post-increment address by 2.
Step 4: Load write buffer for last two bytes.		
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 writing data, repeat Steps 2 through 4, where the Address Pointer is incremented by 2 at each iteration of the loop.		

3.2.1 MODIFYING CODE MEMORY

The previous programming example assumed that the device had 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.

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

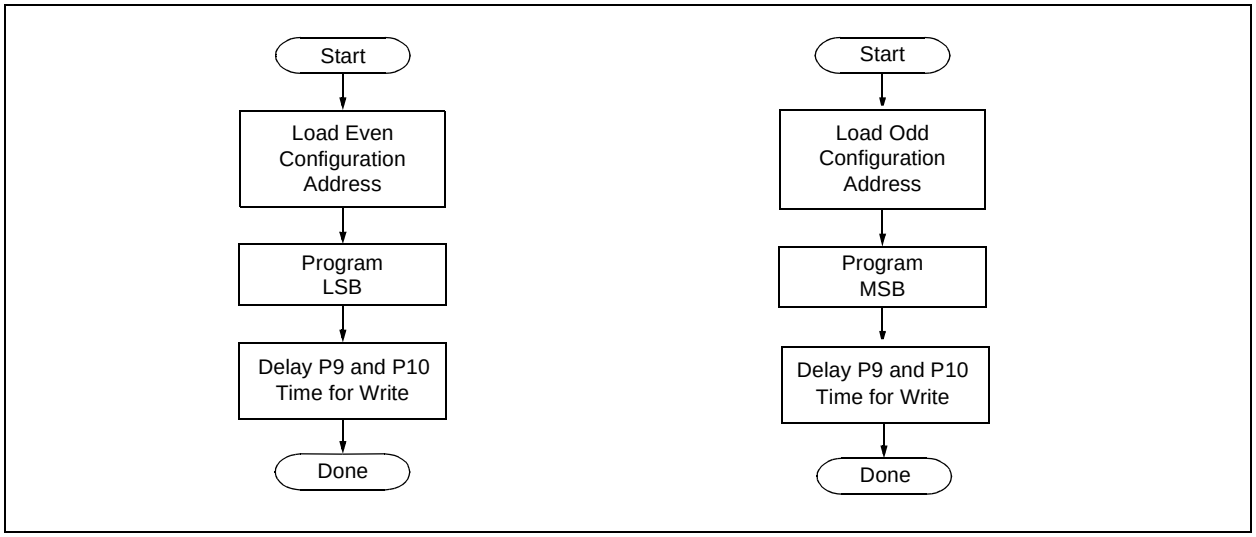
4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to code memory.		
Step 2: Read and modify code memory (see **Section 4.1 "Read Code Memory, ID Locations and Configuration Bits"**).		
0000 0000	8E A6 9C A6	BSF EECON1, EEPGD BCF EECON1, CFGS
Step 3: Set the Table Pointer for the block to be erased.		
0000 0000 0000 0000 0000 0000	0E <Addr[21:16]> 6E F8 0E <Addr[8:15]> 6E F7 0E <Addr[7:0]> 6E F6	MOVLW <Addr[21:16]> MOVWF TBLPTRU MOVLW <Addr[8:15]> MOVWF TBLPTRH MOVLW <Addr[7:0]> MOVWF TBLPTRL
Step 4: Enable memory writes and set up an erase.		
0000 0000	84 A6 88 A6	BSF EECON1, WREN BSF EECON1, FREE
Step 5: Initiate erase.		
0000 0000	82 A6 00 00	BSF EECON1, WR NOP - hold PGC high for time P9 and low for time P10.
Step 6: Load write buffer. The correct bytes will be selected based on the Table Pointer.		
0000 0000 0000 0000 0000 0000 1101 . . . 1111 0000	0E <Addr[21:16]> 6E F8 0E <Addr[8:15]> 6E F7 0E <Addr[7:0]> 6E F6 <MSB><LSB> . . . <MSB><LSB> 00 00	MOVLW <Addr[21:16]> MOVWF TBLPTRU MOVLW <Addr[8:15]> MOVWF TBLPTRH MOVLW <Addr[7:0]> MOVWF TBLPTRL Write 2 bytes and post-increment address by 2. Repeat as many times as necessary to fill the write buffer Write 2 bytes and start programming. NOP - hold PGC high for time P9 and low for time P10.
To continue modifying data, repeat Steps 2 through 6, where the Address Pointer is incremented by the appropriate number of bytes (see Table 3-4) at each iteration of the loop. The write cycle must be repeated enough times to completely rewrite the contents of the erase buffer.		
Step 7: Disable writes.		
0000	94 A6	BCF EECON1, WREN

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: Set Table Pointer for configuration byte to be written. Write even/odd addresses.(1)		
0000	0E 30	MOVLW 30h
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPRTH
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 the Configuration bits. Always write all the Configuration bits before enabling the write protection for Configuration bits.

FIGURE 3-8: CONFIGURATION PROGRAMMING FLOW



4.0 READING THE DEVICE

4.1 Read Code Memory, ID Locations and Configuration Bits

Code memory is accessed, one byte at a time, via the 4-bit command, '1001' (Table Read, post-increment). The contents of memory pointed to by the Table Pointer (TBLPTRU:TBLPTRH:TBLPTRL) are serially output on PGD.

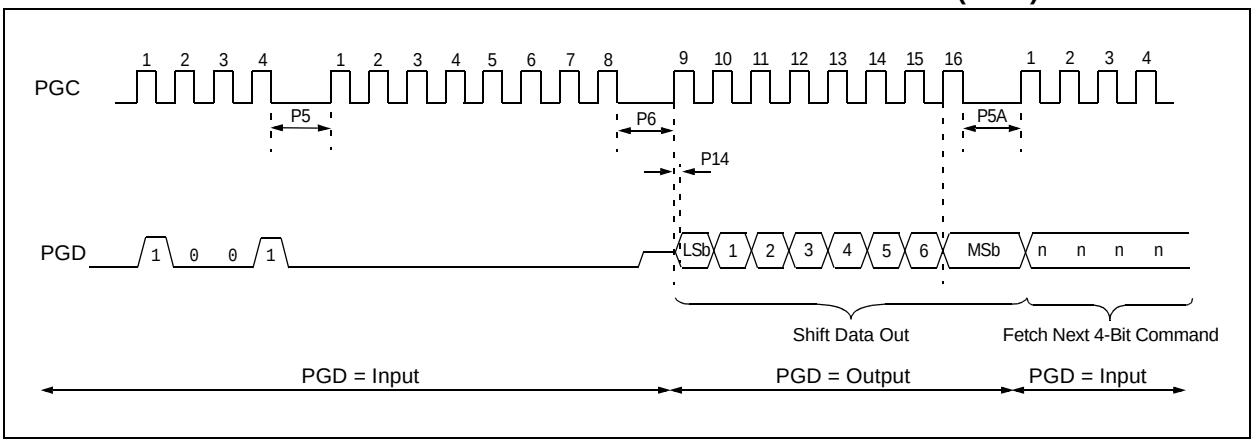
The 4-bit command is shifted in, LSb first. The read is executed during the next eight clocks, then shifted out on PGD during the last eight clocks, LSb to MSb. A delay of P6 must be introduced after the falling edge of the 8th PGC of the operand to allow PGD to transition from an input to an output. 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)



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. The data will be loaded into EEDATA, where it may be serially output on PGD via the 4-bit command, '0010' (Shift Out Data Holding register). A delay of P6 must be introduced after the falling edge of the 8th PGC of the operand to allow PGD to transition from an input to an output. During this time, PGC must be held low (see Figure 4-4).

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

FIGURE 4-3: READ DATA EEPROM FLOW

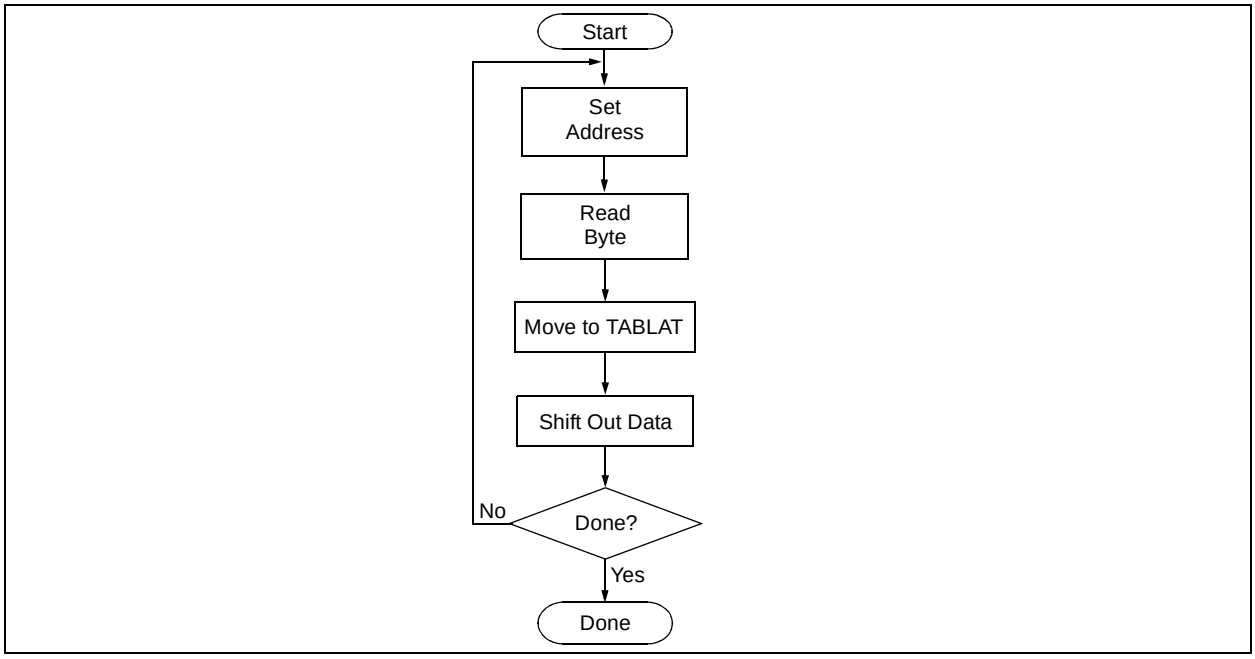


TABLE 4-2: READ DATA EEPROM MEMORY

4-Bit Command	Data Payload	Core Instruction
Step 1: Direct access to data EEPROM.		
0000 0000	9E A6 9C A6	BCF EECON1, EEPGD BCF EECON1, CFGS
Step 2: Set the data EEPROM Address Pointer.		
0000 0000 0000 0000	0E <Addr> 6E A9 OE <AddrH> 6E AA	MOVLW <Addr> MOVWF EEADR MOVLW <AddrH> MOVWF EEADRH
Step 3: Initiate a memory read.		
0000	80 A6	BSF EECON1, RD
Step 4: Load data into the Serial Data Holding register.		
0000 0000 0000 0010	50 A8 6E F5 00 00 <MSB><LSB>	MOVF EEDATA, W, 0 MOVWF TABLAT NOP Shift Out Data(1)

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

TABLE 5-1: CONFIGURATION BITS AND DEVICE IDS

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

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.

TABLE 5-3: PIC18F2XXX/4XXX FAMILY BIT DESCRIPTIONS (CONTINUED)

Bit Name	Configuration Words	Description
PLLDIV<2:0>	CONFIG1L	Oscillator Selection bits **(PIC18F2455/2550/4455/4550, PIC18F2458/2553/4458/4553 and PIC18F2450/4450 devices only)** Divider must be selected to provide a 4 MHz input into the 96 MHz PLL: 111 = Oscillator divided by 12 (48 MHz input) 110 = Oscillator divided by 10 (40 MHz input) 101 = Oscillator divided by 6 (24 MHz input) 100 = Oscillator divided by 5 (20 MHz input) 011 = Oscillator divided by 4 (16 MHz input) 010 = Oscillator divided by 3 (12 MHz input) 001 = Oscillator divided by 2 (8 MHz input) 000 = No divide – oscillator used directly (4 MHz input)
VREGEN	CONFIG2L	USB Voltage Regulator Enable bit **(PIC18F2455/2550/4455/4550, PIC18F2458/2553/4458/4553 and PIC18F2450/4450 devices only)** 1 = USB voltage regulator is enabled 0 = USB voltage regulator is disabled
BORV<1:0>	CONFIG2L	Brown-out Reset Voltage bits 11 = V_{BOR} is set to 2.0V 10 = V_{BOR} is set to 2.7V 01 = V_{BOR} is set to 4.2V 00 = V_{BOR} is set to 4.5V
BOREN<1:0>	CONFIG2L	Brown-out Reset Enable bits 11 = Brown-out Reset is enabled in hardware only (SBOREN is disabled) 10 = Brown-out Reset is enabled in hardware only and disabled in Sleep mode SBOREN is disabled) 01 = Brown-out Reset is enabled and controlled by software (SBOREN is enabled) 00 = Brown-out Reset is disabled in hardware and software
PWRTEN	CONFIG2L	Power-up Timer Enable bit 1 = PWRT is disabled 0 = PWRT is enabled
WDPS<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

Note 1: The BBSIZ bits, BBSIZ<1:0> and BBSIZ<2:1> bits, cannot be changed once any of the following code-protect bits are enabled: CPB or CP0, WRTB or WRT0, EBTRB or EBTR0.

2: Not available in PIC18FXX8X and PIC18F2450/4450 devices.

TABLE 5-3: PIC18F2XXX/4XXX FAMILY BIT DESCRIPTIONS (CONTINUED)

Bit Name	Configuration Words	Description
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 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 part number.
REV<4:0>	DEVID1	Revision ID bits These bits are used to indicate the revision of the device. The REV4 bit is sometimes used to fully specify the device type.

Note 1: The BBSIZ bits, BBSIZ<1:0> and BBSIZ<2:1> bits, cannot be changed once any of the following code-protect bits are enabled: CPB or CP0, WRTB or WRT0, EBTRB or EBTR0.

2: Not available in PIC18FXX8X and PIC18F2450/4450 devices.

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 5-5: 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.