

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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

E·XFI

Product Status	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	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/pic18lf4610t-i-ml

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

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

- PIC18F4221
- PIC18F4321
- PIC18F4410
- PIC18F4420
- PIC18F4423
- PIC18F4450
- PIC18F4455
- PIC18F4458
- PIC18F4480
- PIC18F4510
- PIC18F4520
- PIC18F4515

PIC18F4553
PIC18F4580
PIC18F4585
PIC18F4610
PIC18F4620
PIC18F4680

• PIC18F4523

PIC18F4525

PIC18F4550

- PIC18F4682
- PIC18F4685

FIGURE 2-5: 44-PIN QFN RD2 RD1 VUSB VUSB RC1 RC1 RC1 RC1 33 OSC2 32 OSC1 RD4 2 RD5 3 RD6 4 31 Vss 30 AVss RD7 5 29 VDD PIC18F4XXX 28 AVDD Vss 6 AVDD 7 27 RE2 **VDD** 8 RB0 9 26 RE1 25 RE0 24 RA5 RB1 10 RB2 RA4 23 11 ទទ RA3 S S G<sup>R</sup>B4 RA2 ш RB5/P RB6/P RB7/P MCLR/VPP/R

## 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	
PIC18F2525	
PIC18F2585	
PIC18F4515	00000011-00BFFF11 (40K)
PIC18F4525	
PIC18F4585	
PIC18F2610	
PIC18F2620	
PIC18F2680	
PIC18F4610	0000001-00FFF11(04K)
PIC18F4620	
PIC18F4680	

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

Device	Code Memory Size (Bytes)
PIC18F2682	000000h 012EEEh (80K)
PIC18F4682	00000011-013FFF11(80K)
PIC18F2685	
PIC18F4685	0000001-017FFFI (90K)

## TABLE 2-5: IMPLEMENTATION OF CODE MEMORY

Device	Code Memory Size (Bytes)	
PIC18F2410		
PIC18F2420		
PIC18F2423		
PIC18F2450	000000h-003FFFh (16K)	
PIC18F4410		
PIC18F4420		
PIC18F4450		

#### FIGURE 2-9: MEMORY MAP AND THE CODE MEMORY SPACE FOR PIC18FX4X0/X4X3 DEVICES



For PIC18F2480/4480 devices, the code memory space extends from 0000h to 03FFFh (16 Kbytes) in one 16-Kbyte block. For PIC18F2580/4580 devices, the code memory space extends from 0000h to 07FFFh (32 Kbytes) in two 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 PIC18F2480/2580/4480/4580 devices can be configured as 1 or 2K words (see Figure 2-10). This is done through the BBSIZ<0> bit in the Configuration register, CONFIG4L. It is important to note that increasing the size of the Boot Block decreases the size of Block 0.



## 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.





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

As shown in <u>Figure 2-14</u>, 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



## 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 2-8.

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 2-9. The 4-bit command is shown Most Significant bit (MSb) first. The command operand, or "Data Payload", is shown as <MSB><LSB>. Figure 2-18 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 2-8: 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 2-9: SAMPLE COMMAND SEQUENCE

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



## **FIGURE 2-18:**

#### 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 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 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 MCLR/VPP/RE3, then the state of the MCLR/VPP/RE3 pin is ignored.

The ICPRT Configuration bit can only be programmed through the default ICSP port. Chip Erase functions Note: 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.

Din Namo	During Programming			
	Pin Name	Pin Type	Dedicated Pins	Pin Description
MCLR/Vpp/RE3	Vpp	Р	NC/ICRST/ICVPP	Programming Enable
RB6	PGC	I	NC/ICCK/ICPGC	Serial Clock
RB7	PGD	I/O	NC/ICDT/ICPGD	Serial Data

#### **TABLE 2-10: ICSP™ EQUIVALENT PINS**

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).

Description	Data (3C0005h:3C0004h)
Chip Erase	3F8Fh
Erase Data EEPROM <sup>(1)</sup>	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

## TABLE 3-1: BULK ERASE OPTIONS

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.

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	3F 3F	Write 3F3Fh 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	8F 8F	Write 8F8Fh TO 3C0004h to erase entire device.
		NOP
		Hold PGD low until erase completes.
0000	00 00	
0000	00 00	

## TABLE 3-2: BULK ERASE COMMAND SEQUENCE

#### FIGURE 3-1: BULK ERASE 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 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		
PIC18F2420, PIC18F2520, PIC18F4420, PIC18F4520	32	64
PIC18F2423, PIC18F2523, PIC18F4423, PIC18F4523		
PIC18F2480, PIC18F2580, PIC18F4480, PIC18F4580		
PIC18F2455, PIC18F2550, PIC18F4455, PIC18F4550		
PIC18F2458, PIC18F2553, PIC18F4458, PIC18F4553		
PIC18F2515, PIC18F2610, PIC18F4515, PIC18F4610		
PIC18F2525, PIC18F2620, PIC18F4525, PIC18F4620	64	64
PIC18F2585, PIC18F2680, PIC18F4585, PIC18F4680		04
PIC18F2682, PIC18F2685, PIC18F4682, PIC18F4685	]	

## 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 3.2.1** "**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 acc	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></lsb></msb>	Write 2 bytes and post-increment address by 2.			
1101	<msb><lsb></lsb></msb>	Write 2 bytes and post-increment address by 2.			
1101	<msb><lsb></lsb></msb>	Write 2 bytes and post-increment address by 2.			
1111	<msb><lsb></lsb></msb>	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.

## 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 eight 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.

## 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.

4-Bit Command	Data Payload	Core Instruction						
Step 1: Set Table Pointer.								
0000	0E <addr[21:16]></addr[21:16]>	MOVLW Addr[21:16]						
0000	6E F8	MOVWF TBLPTRU						
0000	0E <addr[15:8]></addr[15:8]>	MOVLW <addr[15:8]></addr[15:8]>						
0000	6E F7	MOVWF TBLPTRH						
0000	0E <addr[7:0]></addr[7:0]>	MOVLW <addr[7:0]></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 *+						

TABLE 4-1:READ CODE MEMORY SEQUENCE





## 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 acc	ess to data EEPROM.								
0000 0000	9E A6 9C A6	BCF EECON1, EEPGD BCF EECON1, CFGS							
Step 2: Set the da	ata EEPROM Address Pointe	er.							
0000 0000 0000 0000	0E <addr> 6E A9 OE <addrh> 6E AA</addrh></addr>	MOVLW <addr> MOVWF EEADR MOVLW <addrh> MOVWF EEADRH</addrh></addr>							
Step 3: Initiate a r	memory read.								
0000	80 A6	BSF EECON1, RD							
Step 4: Load data	Step 4: Load data into the Serial Data Holding register.								
0000 0000 0000 0010	50 A8 6E F5 00 00 <msb><lsb></lsb></msb>	MOVF EEDATA, W, O MOVWF TABLAT NOP Shift Out Data <sup>(1)</sup>							

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

## 5.0 CONFIGURATION WORD

The PIC18F2XXX/4XXX Family devices have several Configuration Words. These bits can be set or cleared to select various device configurations. All other memory areas should be programmed and verified prior to setting 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.

## 5.2 Device ID Word

The Device ID Word for the PIC18F2XX/4XXX Family devices is located at 3FFFFEh:3FFFFh. 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.

In some cases, devices may share the same DEVID values. In such cases, the Most Significant bit of the device revision, REV4 (DEVID1<4>), will need to be examined to completely determine the device being accessed.

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 VALUES

<b>_</b>	Device ID Value						
Device	DEVID2	DEVID1					
PIC18F2221	21h	011x xxxx					
PIC18F2321	21h	001x xxxx					
PIC18F2410	11h	011x xxxx					
PIC18F2420	11h	010x xxxx(1)					
PIC18F2423	11h	010x xxxx <sup>(2)</sup>					
PIC18F2450	24h	001x xxxx					
PIC18F2455	12h	011x xxxx					
PIC18F2458	2Ah	011x xxxx					
PIC18F2480	1Ah	111x xxxx					
PIC18F2510	11h	001x xxxx					
PIC18F2515	0Ch	111x xxxx					
PIC18F2520	11h	000x xxxx(1)					
PIC18F2523	11h	000x xxxx <sup>(2)</sup>					
PIC18F2525	0Ch	110x xxxx					
PIC18F2550	12h	010x xxxx					
PIC18F2553	2Ah	010x xxxx					
PIC18F2580	1Ah	110x xxxx					
PIC18F2585	0Eh	111x xxxx					
PIC18F2610	0Ch	101x xxxx					
PIC18F2620	0Ch	100x xxxx					
PIC18F2680	0Eh	110x xxxx					
PIC18F2682	27h	000x xxxx					
PIC18F2685	27h	001x xxxx					
PIC18F4221	21h	010x xxxx					
PIC18F4321	21h	000x xxxx					
PIC18F4410	10h	111x xxxx					
PIC18F4420	10h	110x xxxx(1)					
PIC18F4423	10h	110x xxxx <sup>(2)</sup>					
PIC18F4450	24h	000x xxxx					
PIC18F4455	12h	001x xxxx					
PIC18F4458	2Ah	001x xxxx					
PIC18F4480	1Ah	101x xxxx					
PIC18F4510	10h	101x xxxx					
PIC18F4515	0Ch	011x xxxx					
PIC18F4520	10h	100x xxxx <sup>(1)</sup>					
PIC18F4523	10h	100x xxxx <sup>(2)</sup>					
PIC18F4525	0Ch	010x xxxx					
PIC18F4550	12h	000x xxxx					
PIC18F4553	2Ah	000x xxxx					
PIC18F4580	1Ah	100x xxxx					

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

**Note 1:** DEVID1 bit 4 is used to determine the device type (REV4 = 0).

**2**: DEVID1 bit 4 is used to determine the device type (REV4 = 1).

#### Configuration Bit Name Description Words BBSIZ<1:0>(1) CONFIG4L Boot Block Size Select bits (PIC18F2321/4321 devices only) 11 = 1K word (2 Kbytes) Boot Block 10 = 1K word (2 Kbytes) Boot Block 01 = 512 words (1 Kbyte) Boot Block 00 = 256 words (512 bytes) Boot Block Boot Block Size Select bits (PIC18F2221/4221 devices only) 11 = 512 words (1 Kbyte) Boot Block 10 = 512 words (1 Kbyte) Boot Block 01 = 512 words (1 Kbyte) Boot Block 00 = 256 words (512 bytes) Boot Block BBSIZ<sup>(1)</sup> CONFIG4I Boot Block Size Select bits (PIC18F2480/2580/4480/4580 and PIC18F2450/4450 devices only) 1 = 2K words (4 Kbytes) Boot Block 0 = 1K word (2 Kbytes) Boot Block LVP CONFIG4L Low-Voltage Programming Enable bit 1 = Low-Voltage Programming is enabled, RB5 is the PGM pin 0 = Low-Voltage Programming is disabled, RB5 is an I/O pin STVREN CONFIG4L Stack Overflow/Underflow Reset Enable bit 1 = Reset on stack overflow/underflow is enabled 0 = Reset on stack overflow/underflow is disabled CP5 CONFIG5L Code Protection bit (Block 5 code memory area) (PIC18F2685 and PIC18F4685 devices only) 1 = Block 5 is not code-protected 0 = Block 5 is code-protected CP4 CONFIG5L Code Protection bit (Block 4 code memory area) (PIC18F2682/2685 and PIC18F4682/4685 devices only) 1 = Block 4 is not code-protected 0 = Block 4 is code-protected CP3 CONFIG5L Code Protection bit (Block 3 code memory area) 1 = Block 3 is not code-protected 0 = Block 3 is code-protected CP2 CONFIG5L Code Protection bit (Block 2 code memory area) 1 = Block 2 is not code-protected 0 = Block 2 is code-protected CP1 CONFIG5L Code Protection bit (Block 1 code memory area) 1 = Block 1 is not code-protected 0 = Block 1 is code-protected CP0 CONFIG5L Code Protection bit (Block 0 code memory area) 1 = Block 0 is not code-protected 0 = Block 0 is code-protected CPD CONFIG5H Code Protection bit (Data EEPROM) 1 = Data EEPROM is not code-protected 0 = Data EEPROM is code-protected СРВ CONFIG5H Code Protection bit (Boot Block memory area) 1 = Boot Block is not code-protected 0 = Boot Block is code-protected

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

**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.

	Configuration Word (CONFIGxx)													
Daviaa	1L	1H	2L	2H	3L	ЗH	4L	4H	5L	5H	6L	6H	7L	7H
Device	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

## TABLE 5-5: CONFIGURATION WORD MASKS FOR COMPUTING CHECKSUMS

Legend: Shaded cells are unimplemented.

#### 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.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

## QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV = ISO/TS 16949=

#### Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBlox, KEELOQ, KEELOQ logo, Kleer, LANCheck, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC<sup>32</sup> logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

The Embedded Control Solutions Company and mTouch are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, ECAN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, KleerNet, KleerNet logo, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, RightTouch logo, REAL ICE, SQI, Serial Quad I/O, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, 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.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

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

© 2010-2015, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-63277-856-7

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELoQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and mulfacture of development systems is ISO 9001:2000 certified.