



Welcome to [E-XFL.COM](https://www.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

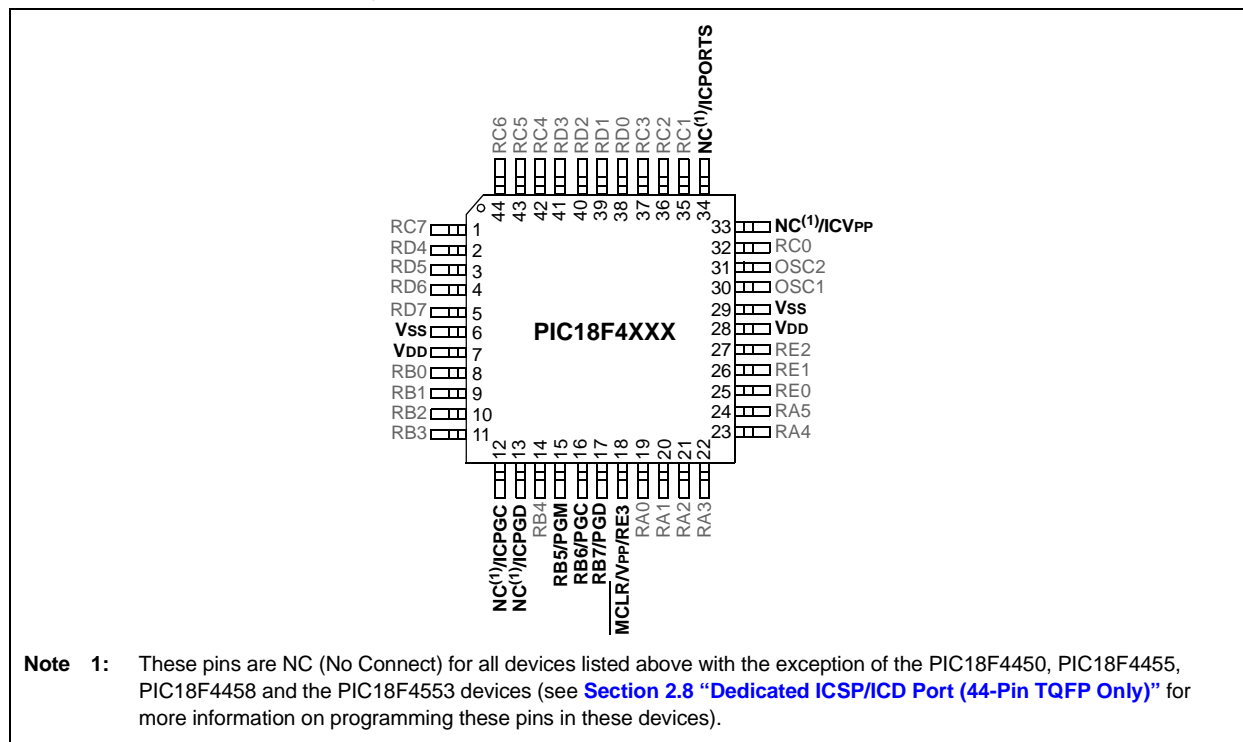
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	48KB (24K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic18f4515-i-p">https://www.e-xfl.com/product-detail/microchip-technology/pic18f4515-i-p</a>

# PIC18F2XXX/4XXX FAMILY

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**

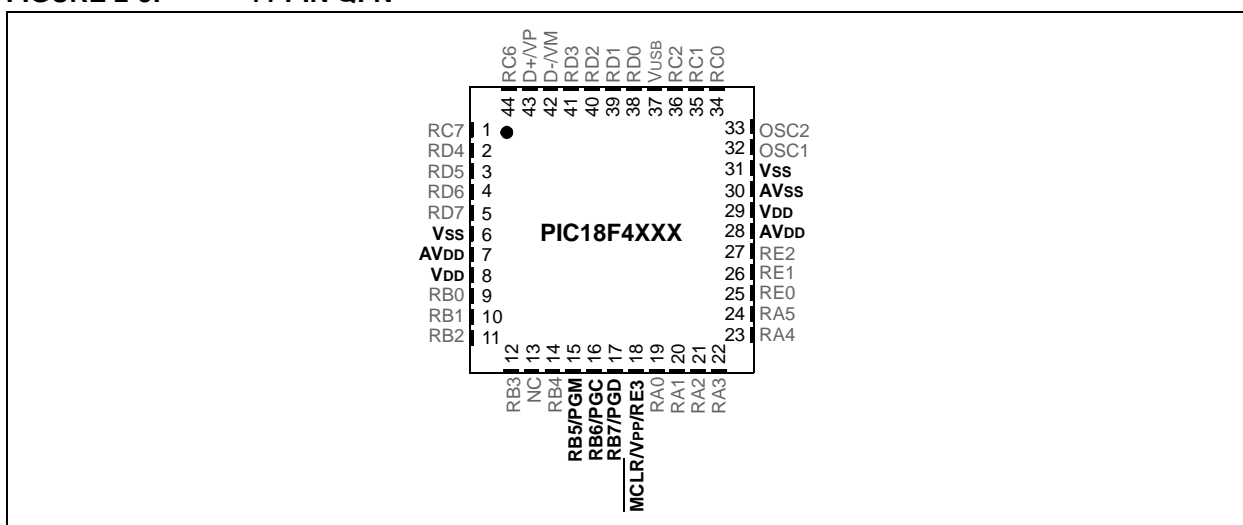


# PIC18F2XXX/4XXX FAMILY

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.

# PIC18F2XXX/4XXX FAMILY

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**

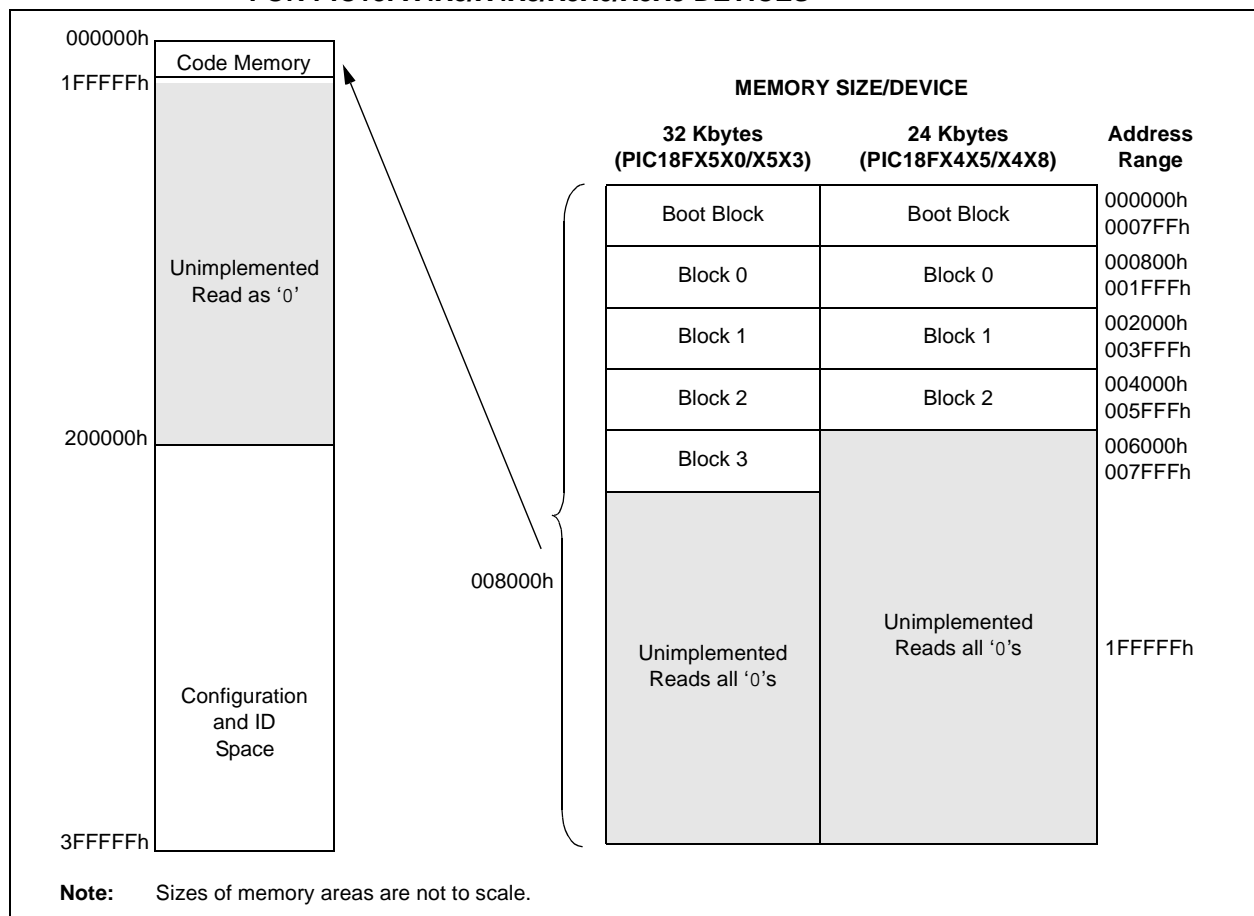
Device	Code Memory Size (Bytes)
PIC18F2682	000000h-013FFFh (80K)
PIC18F4682	
PIC18F2685	000000h-017FFFh (96K)
PIC18F4685	

# PIC18F2XXX/4XXX FAMILY

**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**



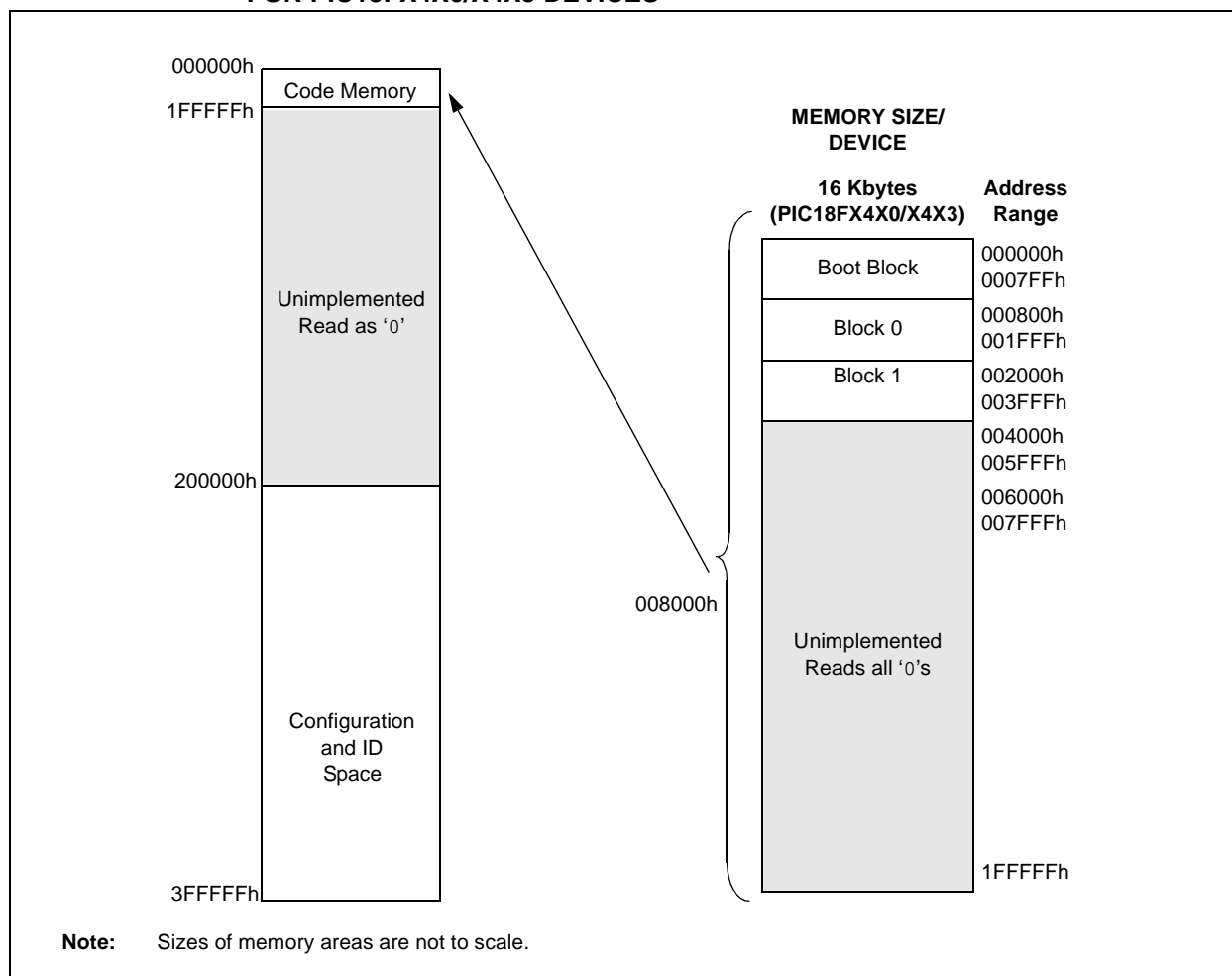
For PIC18FX4X0/X4X3 devices, the code memory space extends from 000000h to 003FFFh (16 Kbytes) in two 8-Kbyte blocks. Addresses, 000000h through 0003FFFh, however, define a "Boot Block" region that is treated separately from Block 0. All of these blocks define code protection boundaries within the code memory space.

# PIC18F2XXX/4XXX FAMILY

**TABLE 2-5: IMPLEMENTATION OF CODE MEMORY**

Device	Code Memory Size (Bytes)
PIC18F2410	000000h-003FFFh (16K)
PIC18F2420	
PIC18F2423	
PIC18F2450	
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 07FFFh, 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.

# PIC18F2XXX/4XXX FAMILY

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 3FFFFFFh, is addressed via the Table Pointer register, which is comprised of three pointer registers:

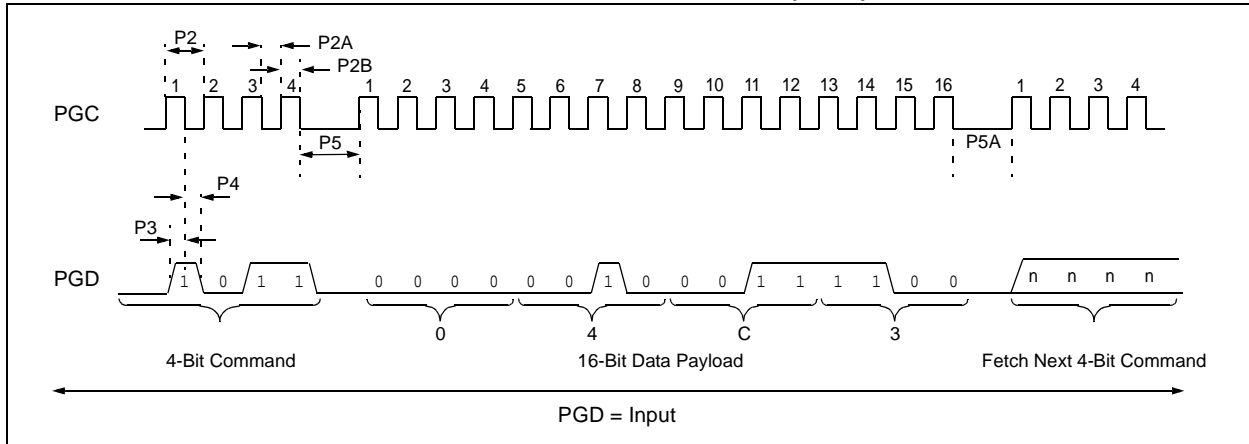
- TBLPTRU at RAM address 0FF8h
- TBLPTRH at RAM address 0FF7h
- TBLPTRL at RAM address 0FF6h

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.

# PIC18F2XXX/4XXX FAMILY

**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 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  $V_{IH}$  is seen on the MCLR/VPP/RE3 pin prior to applying  $V_{IH}$  to the ICRST/ICVPP pin, then the state of the ICRST/ICVPP pin is ignored. Likewise, when the  $V_{IH}$  is seen on ICRST/ICVPP prior to applying  $V_{IH}$  to MCLR/VPP/RE3, then the state of the 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
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



# PIC18F2XXX/4XXX FAMILY

## 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 <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

**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](#).

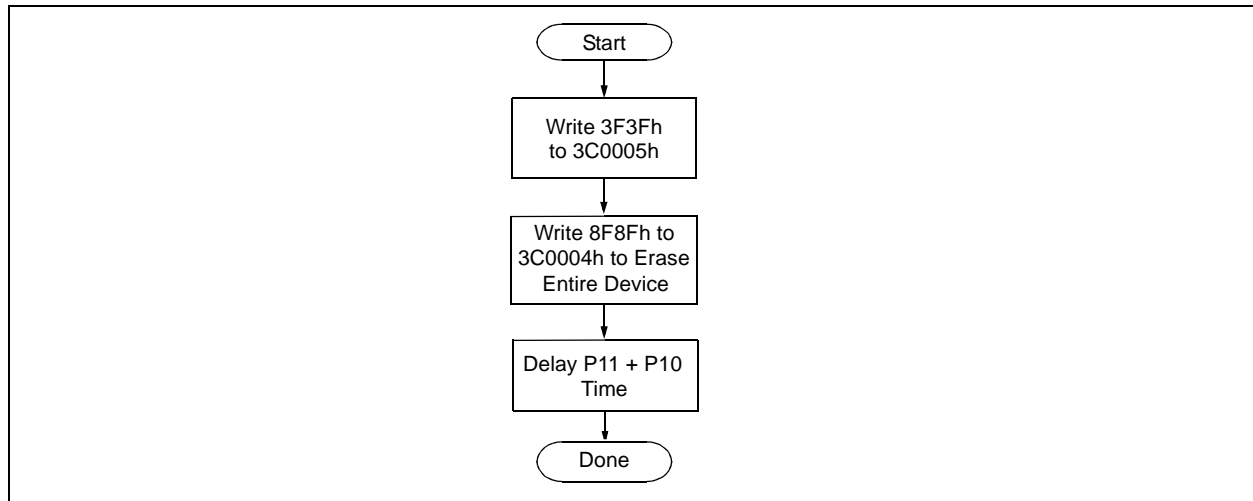
**Note:** A Bulk Erase is the only way to reprogram code-protect bits from an ON state to an OFF state.

# PIC18F2XXX/4XXX FAMILY

**TABLE 3-2: BULK ERASE COMMAND SEQUENCE**

4-Bit Command	Data Payload	Core Instruction
0000	0E 3C	MOVLW 3Ch
0000	6E F8	MOVWF TBLPTRU
0000	0E 00	MOVLW 00h
0000	6E F7	MOVWF TBLPTRH
0000	0E 05	MOVLW 05h
0000	6E F6	MOVWF TBLPTRL
1100	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	

**FIGURE 3-1: BULK ERASE FLOW**



# PIC18F2XXX/4XXX FAMILY

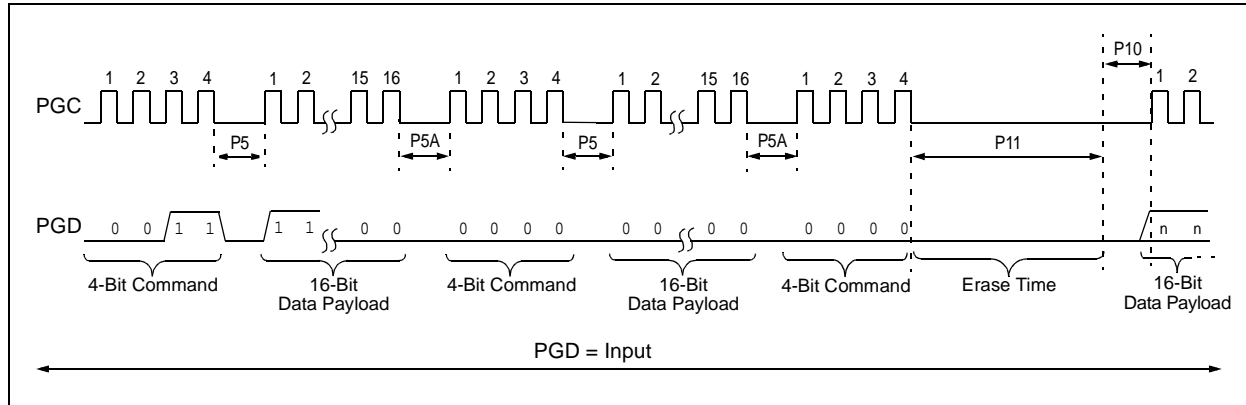
## 3.1.2 LOW-VOLTAGE ICSP BULK ERASE

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

If it is determined that a program memory erase must be performed at a supply voltage below the Bulk Erase limit, refer to the erase methodology described in [Section 3.1.3 “ICSP Row Erase”](#) and [Section 3.2.1 “Modifying Code Memory”](#).

If it is determined that a data EEPROM erase (selected devices only, see [Section 3.3 “Data EEPROM Programming”](#)) must be performed at a supply voltage below the Bulk Erase limit, follow the methodology described in [Section 3.3 “Data EEPROM Programming”](#) and write ‘1’s to the array.

**FIGURE 3-2: BULK ERASE TIMING**



## 3.1.3 ICSP ROW ERASE

Regardless of whether high or low-voltage ICSP is used, it is possible to erase one row (64 bytes of data), provided the block is not code or write-protected. Rows are located at static boundaries, beginning at program memory address, 000000h, extending to the internal program memory limit (see [Section 2.3 “Memory Maps”](#)).

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

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

The code sequence to Row Erase a PIC18F2XXX/4XXX Family device is shown in [Table 3-3](#). The flowchart, shown in [Figure 3-3](#), depicts the logic necessary to completely erase 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 can point to any byte within the row intended for erase.

# PIC18F2XXX/4XXX FAMILY

## 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		

# PIC18F2XXX/4XXX FAMILY

FIGURE 3-4: PROGRAM CODE MEMORY FLOW

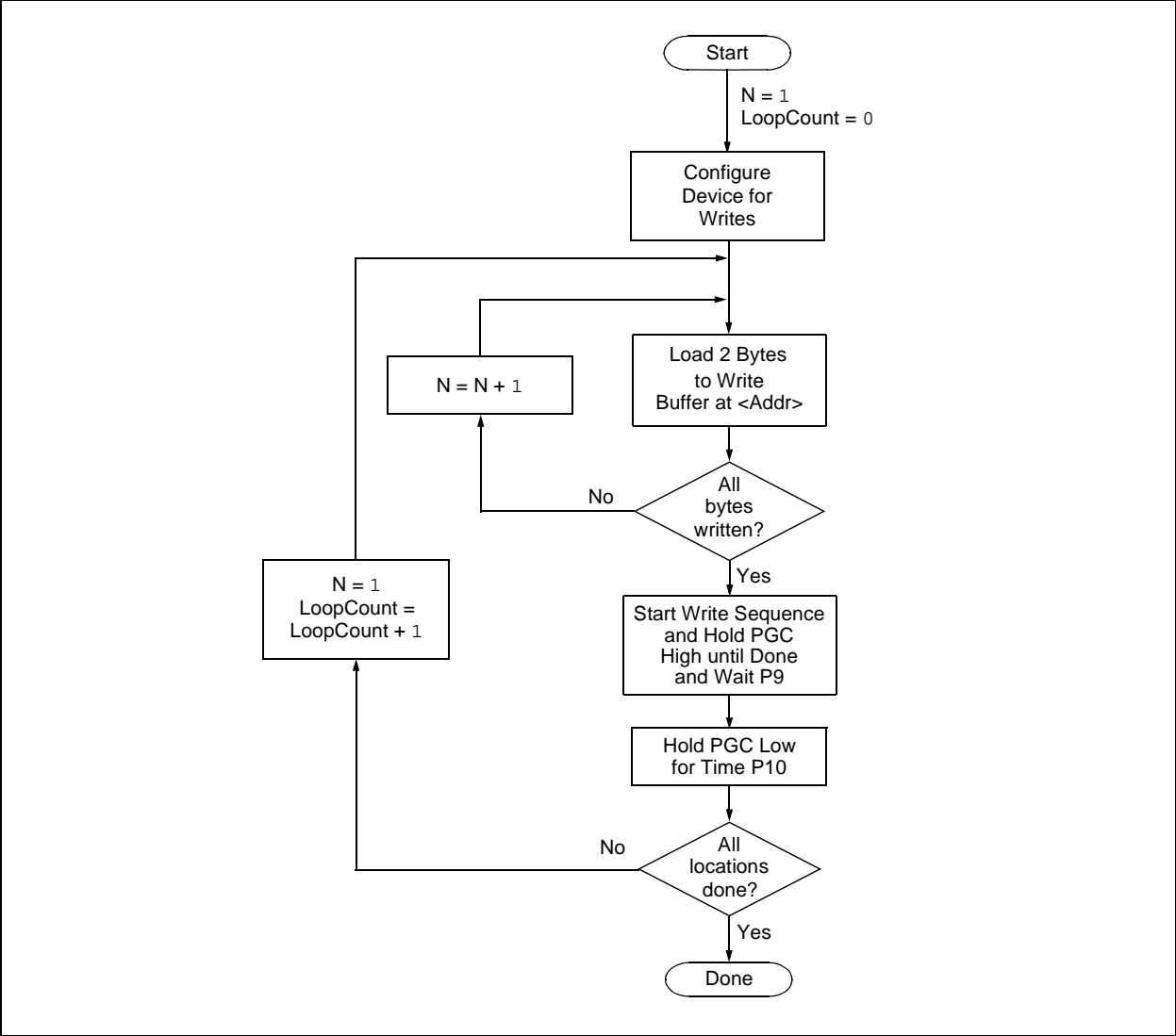
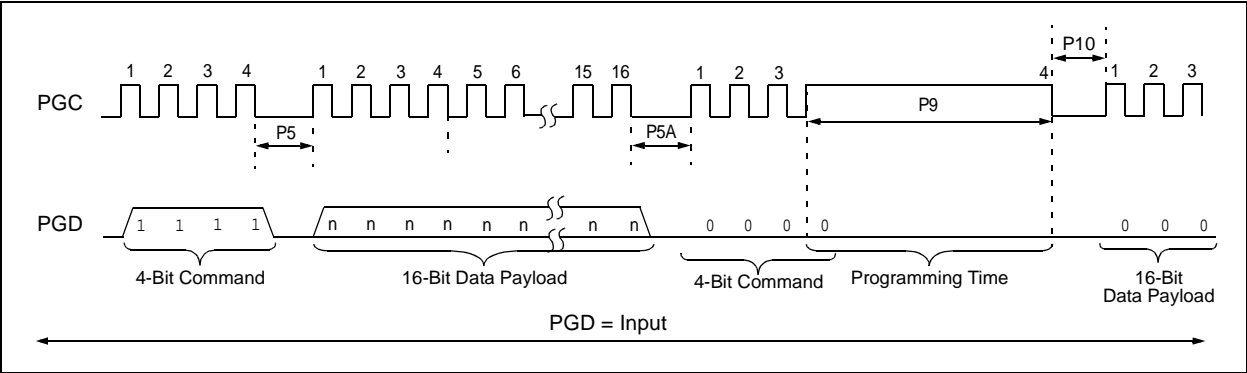


FIGURE 3-5: TABLE WRITE AND START PROGRAMMING INSTRUCTION TIMING (1111)



## 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.

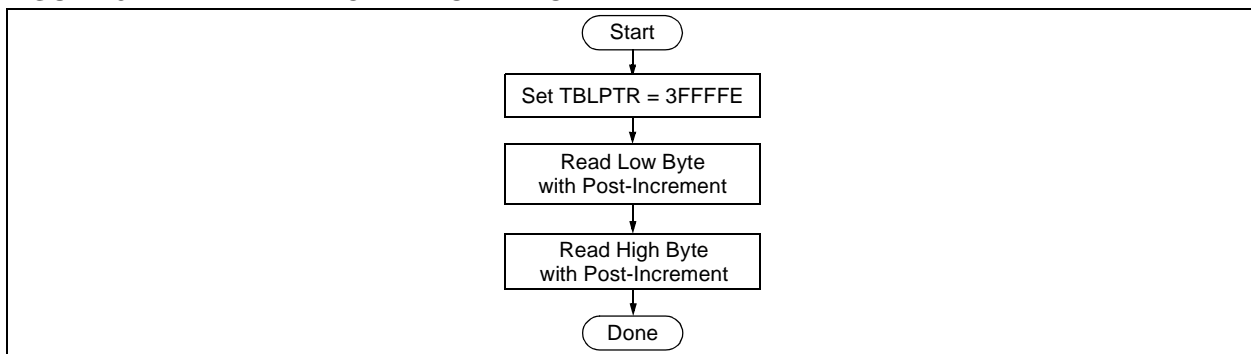
### 5.2 Device ID Word

The Device ID Word for the PIC18F2XXX/4XXX Family devices is located at 3FFFFEh:3FFFFFh. These bits may be used by the programmer to identify what device type is being programmed and read out normally, even after code or read protection.

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**



# PIC18F2XXX/4XXX FAMILY

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

# PIC18F2XXX/4XXX FAMILY

TABLE 5-2: DEVICE ID VALUES (CONTINUED)

Device	Device ID Value	
	DEVID2	DEVID1
PIC18F4585	0Eh	101x xxxx
PIC18F4610	0Ch	001x xxxx
PIC18F4620	0Ch	000x xxxx
PIC18F4680	0Eh	100x xxxx
PIC18F4682	27h	010x xxxx
PIC18F4685	27h	011x 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).



# PIC18F2XXX/4XXX FAMILY

**TABLE 5-3: PIC18F2XXX/4XXX FAMILY BIT DESCRIPTIONS**

Bit Name	Configuration Words	Description
IESO	CONFIG1H	Internal External Switchover bit 1 = Internal External Switchover mode is enabled 0 = Internal External Switchover mode is disabled
FCMEN	CONFIG1H	Fail-Safe Clock Monitor Enable bit 1 = Fail-Safe Clock Monitor is enabled 0 = Fail-Safe Clock Monitor is disabled
FOSC<3:0>	CONFIG1H	Oscillator Selection bits 11xx = External RC oscillator, CLKO function on RA6 101x = External RC oscillator, CLKO function on RA6 1001 = Internal RC oscillator, CLKO function on RA6, port function on RA7 1000 = Internal RC oscillator, port function on RA6, port function on RA7 0111 = External RC oscillator, port function on RA6 0110 = HS oscillator, PLL is enabled (Clock Frequency = 4 x FOSC1) 0101 = EC oscillator, port function on RA6 0100 = EC oscillator, CLKO function on RA6 0011 = External RC oscillator, CLKO function on RA6 0010 = HS oscillator 0001 = XT oscillator 0000 = LP oscillator
FOSC<3:0>	CONFIG1H	Oscillator Selection bits <b>(PIC18F2455/2550/4455/4550, PIC18F2458/2553/4458/4553 and PIC18F2450/4450 devices only)</b> 111x = HS oscillator, PLL is enabled, HS is used by USB 110x = HS oscillator, HS is used by USB 1011 = Internal oscillator, HS is used by USB 1010 = Internal oscillator, XT is used by USB 1001 = Internal oscillator, CLKO function on RA6, EC is used by USB 1000 = Internal oscillator, port function on RA6, EC is used by USB 0111 = EC oscillator, PLL is enabled, CLKO function on RA6, EC is used by USB 0110 = EC oscillator, PLL is enabled, port function on RA6, EC is used by USB 0101 = EC oscillator, CLKO function on RA6, EC is used by USB 0100 = EC oscillator, port function on RA6, EC is used by USB 001x = XT oscillator, PLL is enabled, XT is used by USB 000x = XT oscillator, XT is used by USB
USBDIV	CONFIG1L	USB Clock Selection bit <b>(PIC18F2455/2550/4455/4550, PIC18F2458/2553/4458/4553 and PIC18F2450/4450 devices only)</b> Selects the clock source for full-speed USB operation: 1 = USB clock source comes from the 96 MHz PLL divided by 2 0 = USB clock source comes directly from the OSC1/OSC2 oscillator block; no divide
CPUDIV<1:0>	CONFIG1L	CPU System Clock Selection bits <b>(PIC18F2455/2550/4455/4550, PIC18F2458/2553/4458/4553 and PIC18F2450/4450 devices only)</b> 11 = CPU system clock divided by 4 10 = CPU system clock divided by 3 01 = CPU system clock divided by 2 00 = No CPU system clock divide

**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.

# PIC18F2XXX/4XXX FAMILY

---

## 5.6.3 ID LOCATIONS

Normally, the contents of these locations are defined by the user, but MPLAB® IDE provides the option of writing the device's unprotected 16-bit checksum in the 16 Most Significant bits of the ID locations (see MPLAB IDE Configure/ID Memory" menu). The lower 16 bits are not used and remain clear. This is the sum of all program memory contents and Configuration Words (appropriately masked) before any code protection is enabled.

If the user elects to define the contents of the ID locations, nothing about protected blocks can be known. If the user uses the preprotected checksum, provided by MPLAB IDE, an indirect characteristic of the programmed code is provided.

## 5.6.4 CODE PROTECTION

Blocks that are code-protected read back as all '0's and have no effect on checksum calculations. If any block is code-protected, then the contents of the ID locations are included in the checksum calculation.

All Configuration Words and the ID locations can always be read out normally, even when the device is fully code-protected. Checking the code protection settings in Configuration Words can direct which, if any, of the program memory blocks can be read, and if the ID locations should be used for checksum calculations.

# PIC18F2XXX/4XXX FAMILY

**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.

# PIC18F2XXX/4XXX FAMILY

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

Standard Operating Conditions Operating Temperature: 25°C is recommended						
Param No.	Sym	Characteristic	Min	Max	Units	Conditions
P11A	TDRWT	Data Write Polling Time	4	—	ms	
P12	THLD2	Input Data Hold Time from $\overline{\text{MCLR}}/\text{VPP}/\text{RE3} \uparrow$	2	—	$\mu\text{s}$	
P13	TSET2	VDD $\uparrow$ Setup Time to $\overline{\text{MCLR}}/\text{VPP}/\text{RE3} \uparrow$	100	—	ns	(Note 2)
P14	TVALID	Data Out Valid from PGC $\uparrow$	10	—	ns	
P15	TSET3	PGM $\uparrow$ Setup Time to $\overline{\text{MCLR}}/\text{VPP}/\text{RE3} \uparrow$	2	—	$\mu\text{s}$	(Note 2)
P16	TDLY8	Delay Between Last PGC $\downarrow$ and $\overline{\text{MCLR}}/\text{VPP}/\text{RE3} \downarrow$	0	—	s	
P17	THLD3	$\overline{\text{MCLR}}/\text{VPP}/\text{RE3} \downarrow$ to VDD $\downarrow$	—	100	ns	
P18	THLD4	$\overline{\text{MCLR}}/\text{VPP}/\text{RE3} \downarrow$ to PGM $\downarrow$	0	—	s	

- Note 1:** Do not allow excess time when transitioning  $\overline{\text{MCLR}}$  between  $\text{VIL}$  and  $\text{VIHH}$ . This can cause spurious program executions to occur. The maximum transition time is:  
 1  $\text{T}_{\text{CY}} + \text{T}_{\text{PWRT}}$  (if enabled) + 1024  $\text{T}_{\text{OSC}}$  (for LP, HS, HS/PLL and XT modes only) +  
 2 ms (for HS/PLL mode only) + 1.5  $\mu\text{s}$  (for EC mode only)  
 where  $\text{T}_{\text{CY}}$  is the instruction cycle time,  $\text{T}_{\text{PWRT}}$  is the Power-up Timer period and  $\text{T}_{\text{OSC}}$  is the oscillator period. For specific values, refer to the Electrical Characteristics section of the device data sheet for the particular device.
- 2:** When  $\text{ICPRT} = 1$ , this specification also applies to  $\text{ICVPP}$ .
- 3:** At 0°C-50°C.

---

**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.

#### **Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBlox, KEELOQ, KEELOQ logo, Klear, 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, KlearNet, KlearNet 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

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

*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 manufacture of development systems is ISO 9001:2000 certified.*