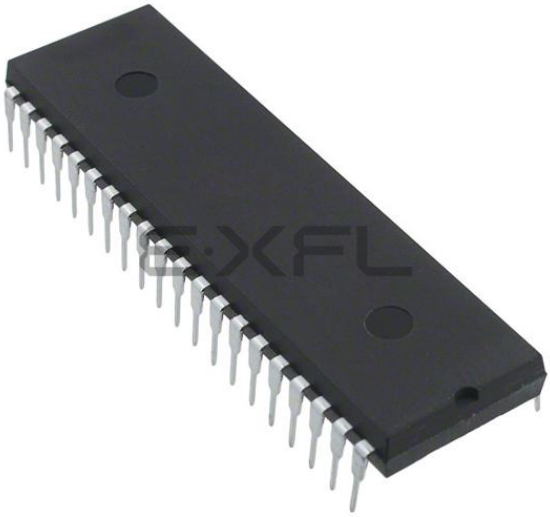


Welcome to E-XFL.COM

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.

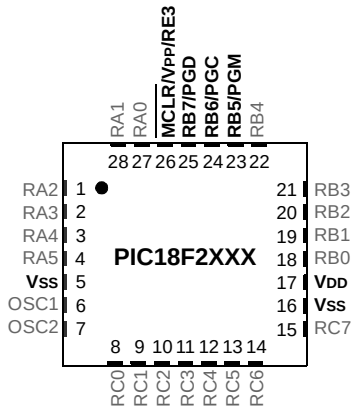
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	25MHz
Connectivity	I²C, 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)	4.2V ~ 5.5V
Data Converters	A/D 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f4610-e-p

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

- PIC18F2221
- PIC18F2321
- PIC18F2410
- PIC18F2420
- PIC18F2423
- PIC18F2450
- PIC18F2480
- PIC18F2510
- PIC18F2520
- PIC18F2523
- PIC18F2580
- PIC18F2682
- PIC18F2685

FIGURE 2-2: 28-Pin QFN



The following devices are included in 40-pin PDIP parts:

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

FIGURE 2-3: 40-Pin PDIP

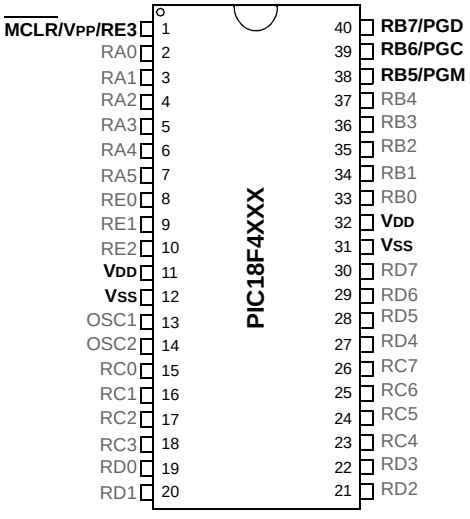
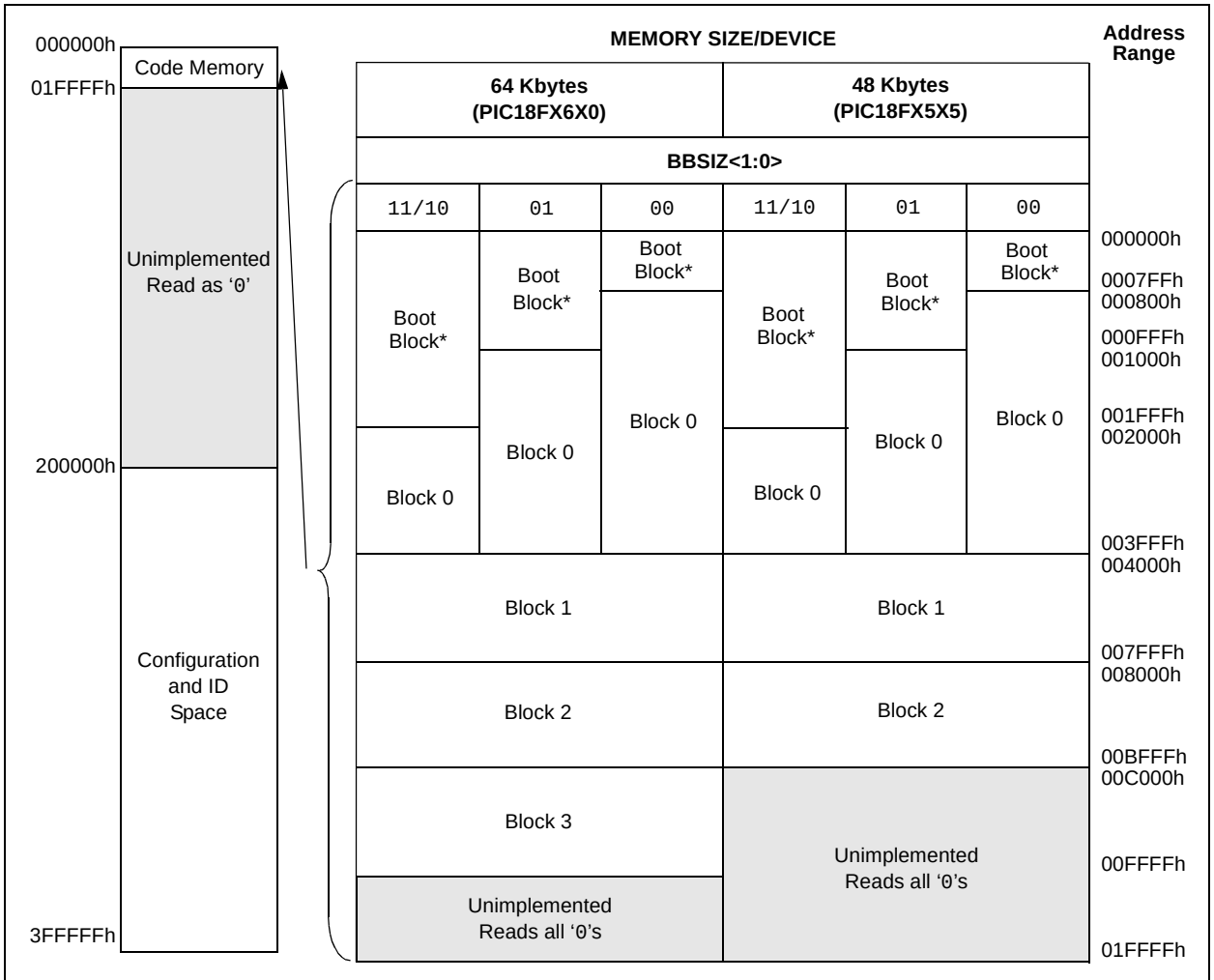


TABLE 2-2: IMPLEMENTATION OF CODE MEMORY

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.

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.

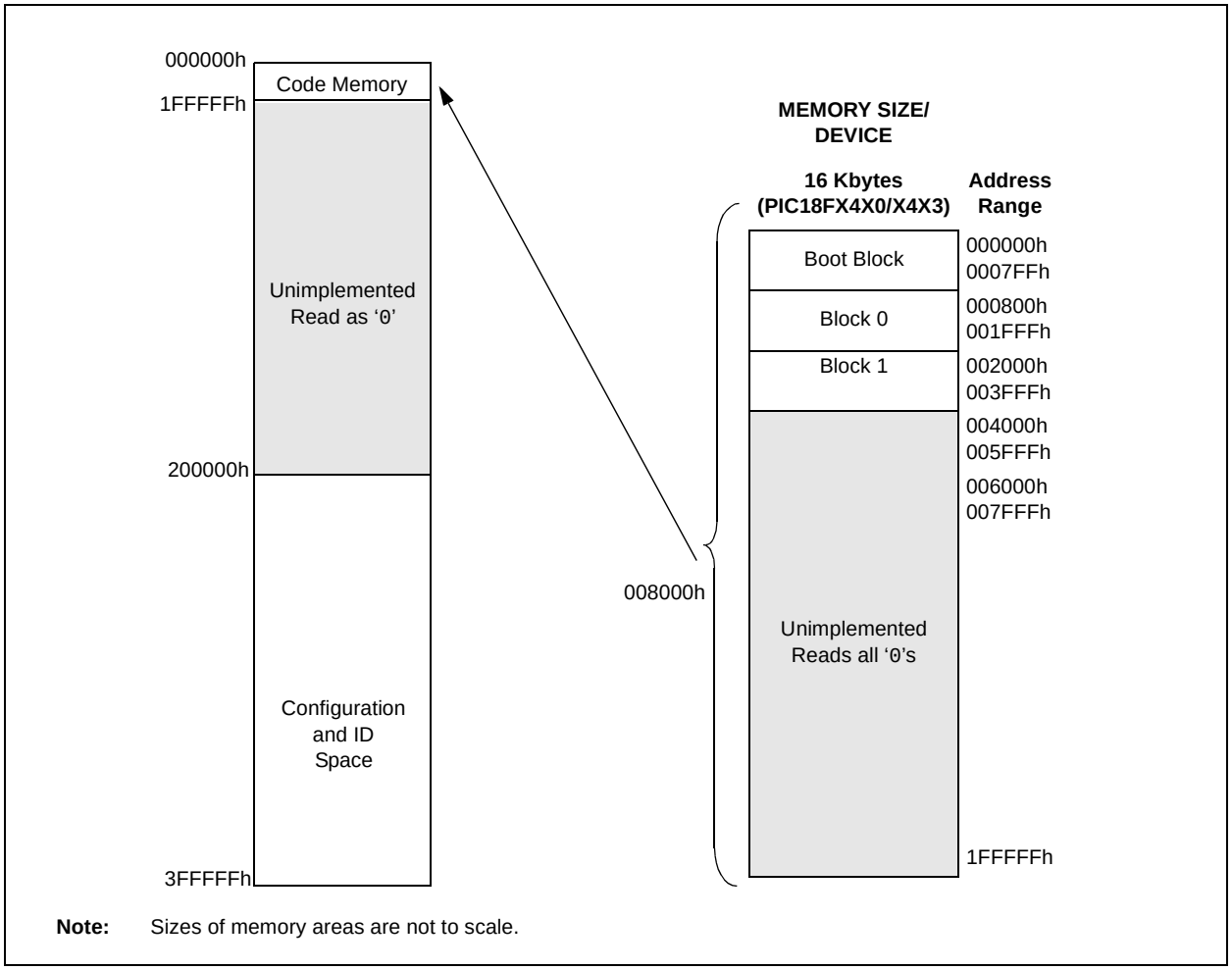
TABLE 2-3: IMPLEMENTATION OF CODE MEMORY

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

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



Note: Sizes of memory areas are not to scale.

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.

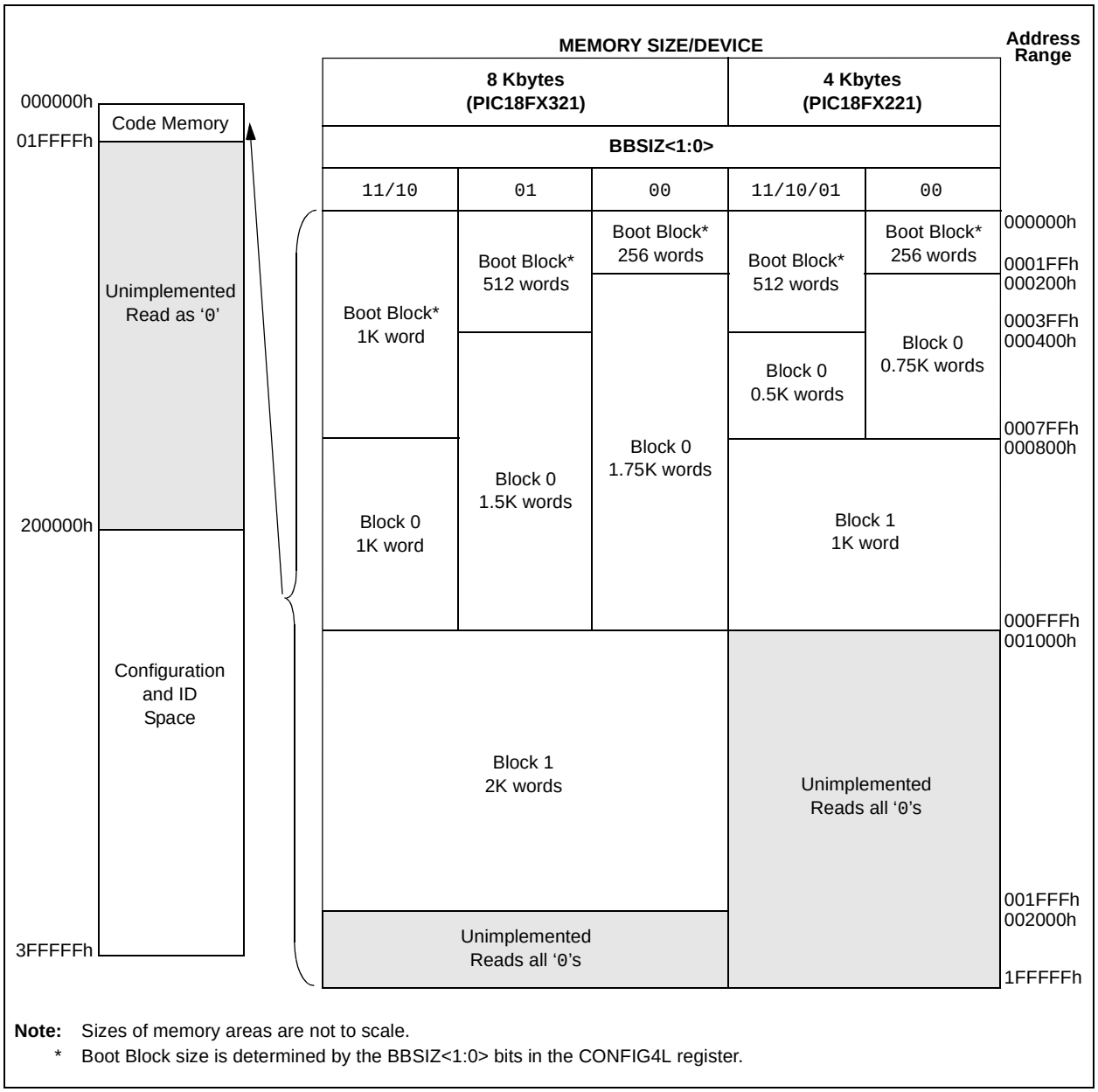
The size of the Boot Block in PIC18F2480/2580/4480/4580 devices can be configured as 1 or 2K words (see Figure 2-10). This is done through the BBSIZ<0> bit in the Configuration register, CONFIG4L. It is important to note that increasing the size of the Boot Block decreases the size of Block 0.

The size of the Boot Block in PIC18F2221/2321/4221/4321 devices can be configured as 256, 512 or 1024 words (see Figure 2-11). This is done through the BBSIZ<1:0> bits in the Configuration register, CONFIG4L (see Figure 2-11). It is important to note that increasing the size of the Boot Block decreases the size of Block 0.

TABLE 2-7: IMPLEMENTATION OF CODE MEMORY

Device	Code Memory Size (Bytes)
PIC18F2221	000000h-000FFFh (4K)
PIC18F4221	
PIC18F2321	000000h-001FFFh (8K)
PIC18F4321	

FIGURE 2-11: MEMORY MAP AND THE CODE MEMORY SPACE FOR PIC18F2221/2321/4221/4321 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.

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

Users may store identification information (ID) in eight ID registers. These ID registers are mapped in addresses, 200000h through 200007h. The ID locations read out normally, even after code protection is applied.

Locations, 300000h through 30000Dh, are reserved for the Configuration bits. These bits select various device options and are described in **Section 5.0 "Configuration Word"**. These Configuration bits read out normally, even after code protection.

Locations, 3FFFFEh and 3FFFFFh, are reserved for the Device ID bits. These bits may be used by the programmer to identify what device type is being programmed and are described in **Section 5.0 "Configuration Word"**. These Device ID bits read out normally, even after code protection.

2.3.1 MEMORY ADDRESS POINTER

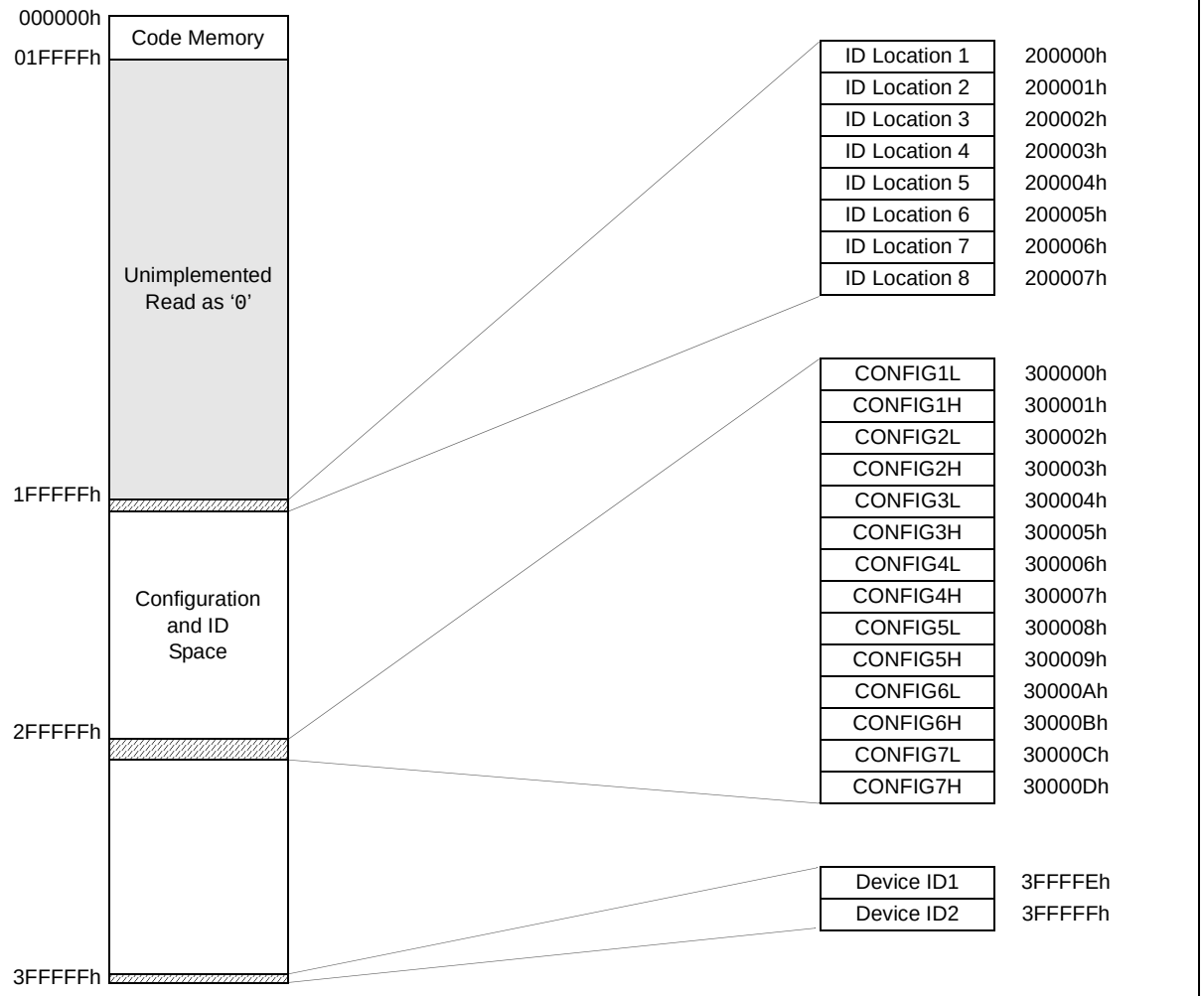
Memory in the address space, 0000000h to 3FFFFFh, is addressed via the Table Pointer register, which is comprised of three pointer registers:

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

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

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

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



Note: Sizes of memory areas are not to scale.

3.0 DEVICE PROGRAMMING

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

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

When using the EECON1 register to act on code memory, the EEPGD bit must be set (EECON1<7> = 1) and the CFGS bit must be cleared (EECON1<6> = 0). The WREN bit must be set (EECON1<2> = 1) to enable writes of any sort (e.g., erases) and this must be done prior to initiating a write sequence. The FREE bit must be set (EECON1<4> = 1) in order to erase the program space being pointed to by the Table Pointer. The erase or write sequence is initiated by setting the WR bit (EECON1<1> = 1). It is strongly recommended that the WREN bit only be set immediately prior to a program erase.

3.1 ICSP Erase

3.1.1 HIGH-VOLTAGE ICSP BULK ERASE

Erasing code or data EEPROM is accomplished by configuring two Bulk Erase Control registers located at 3C0004h and 3C0005h. Code memory may be erased, portions at a time, or the user may erase the entire device in one action. Bulk Erase operations will also clear any code-protect settings associated with the memory block being erased. Erase options are detailed in Table 3-1. If data EEPROM is code-protected (CPD = 0), the user must request an erase of data EEPROM (e.g., 0084h as shown in Table 3-1).

TABLE 3-1: BULK ERASE OPTIONS

Description	Data (3C0005h:3C0004h)
Chip Erase	3F8Fh
Erase Data EEPROM(1)	0084h
Erase Boot Block	0081h
Erase Configuration Bits	0082h
Erase Code EEPROM Block 0	0180h
Erase Code EEPROM Block 1	0280h
Erase Code EEPROM Block 2	0480h
Erase Code EEPROM Block 3	0880h
Erase Code EEPROM Block 4	1080h
Erase Code EEPROM Block 5	2080h

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

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

The code sequence to erase the entire device is shown in Table and the flowchart is shown in Figure 3-1.

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

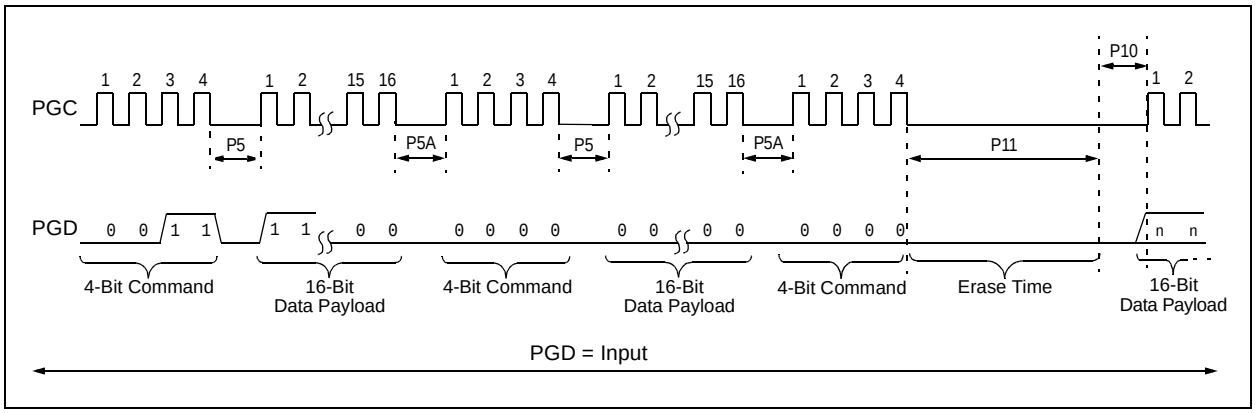
3.1.2 LOW-VOLTAGE ICSP BULK ERASE

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.

FIGURE 3-4: PROGRAM CODE MEMORY FLOW

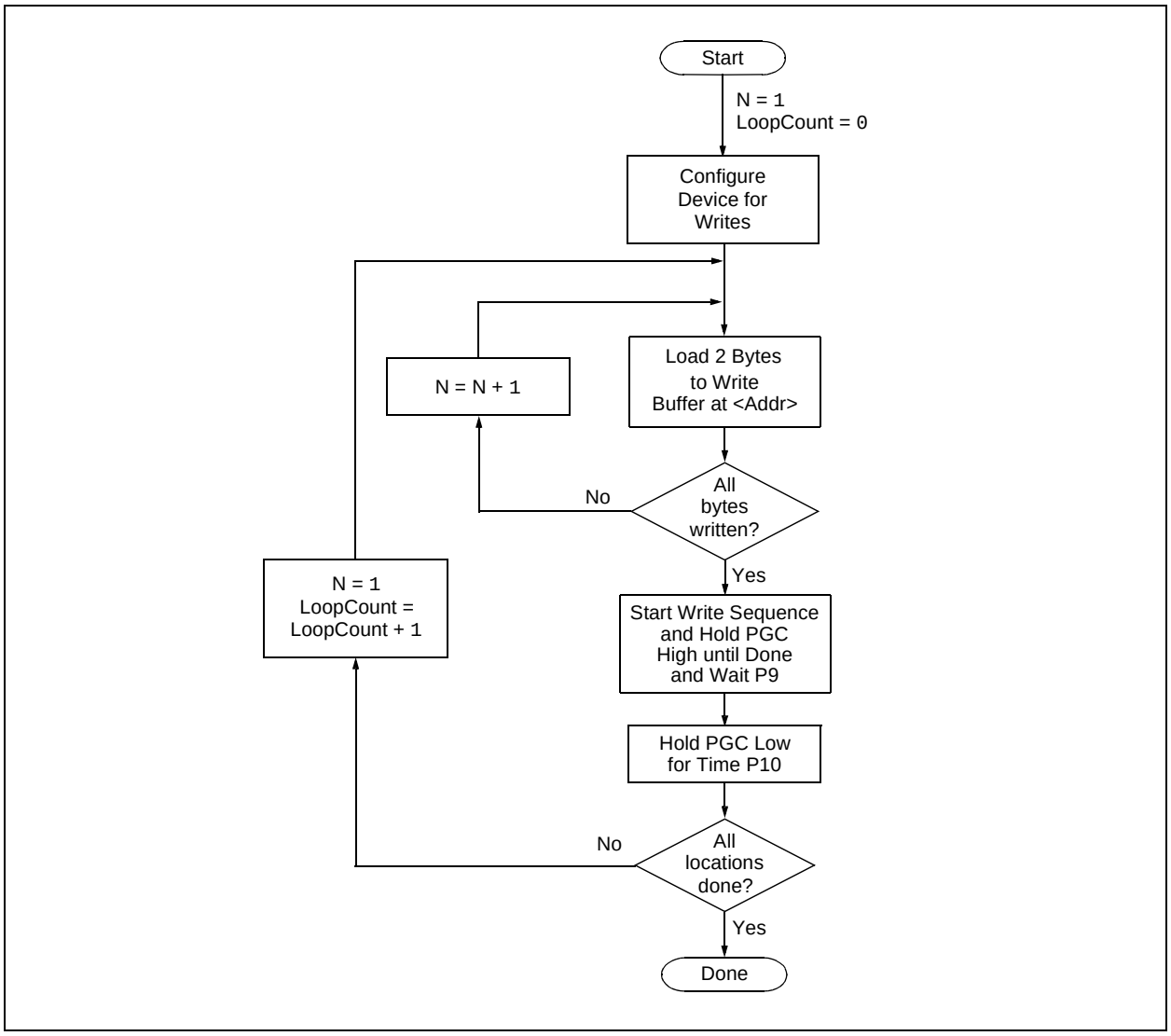
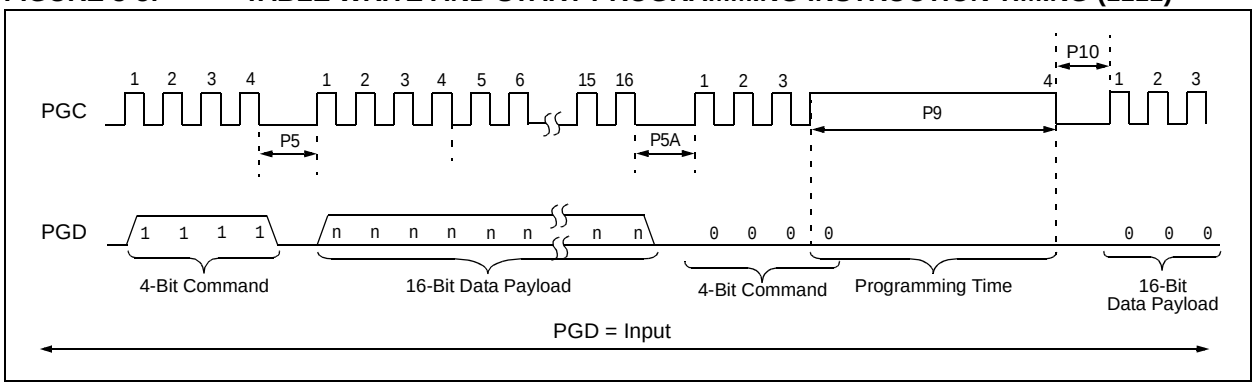


FIGURE 3-5: TABLE WRITE AND START PROGRAMMING INSTRUCTION TIMING (1111)



3.3 Data EEPROM Programming

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

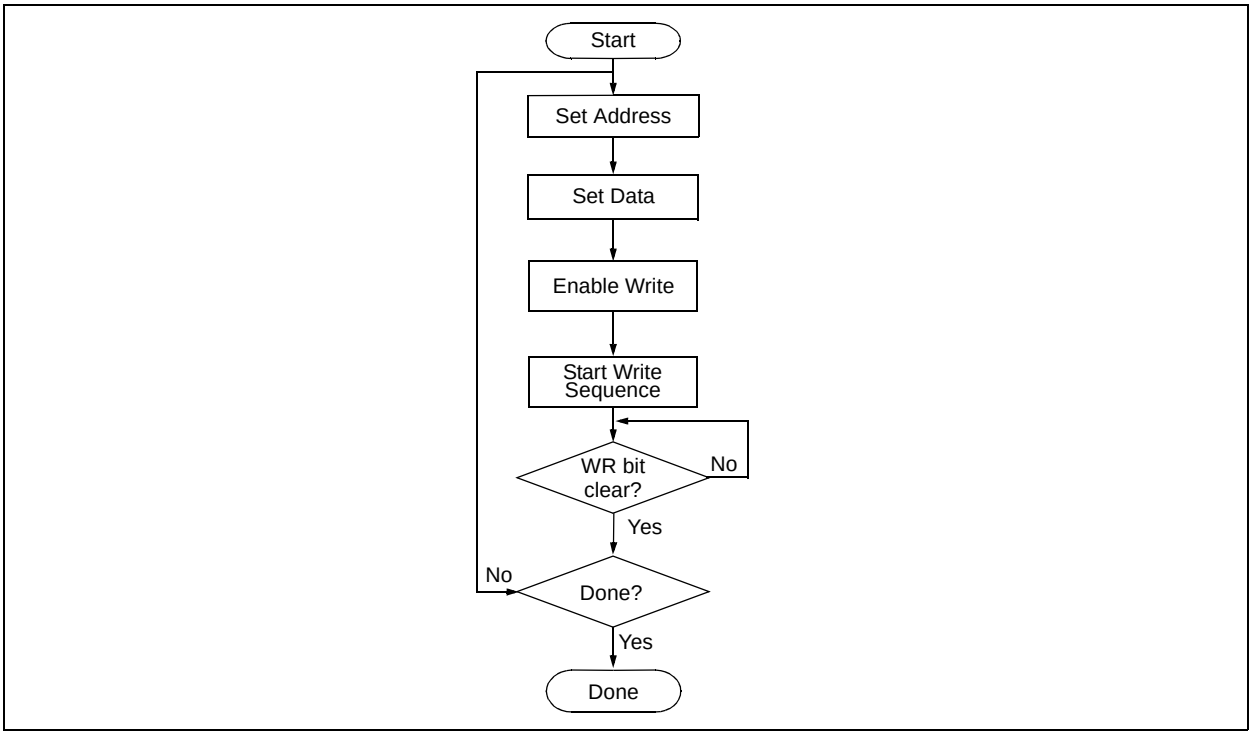
Data EEPROM is accessed one byte at a time via an Address Pointer (register pair: EEADRH:EEADR) and a data latch (EEDATA). Data EEPROM is written by loading EEADRH:EEADR with the desired memory location, EEDATA, with the data to be written and initiating a memory write by appropriately configuring the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

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

The write begins on the falling edge of the 4th PGC after the WR bit is set. It ends when the WR bit is cleared by hardware.

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

FIGURE 3-6: PROGRAM DATA FLOW



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

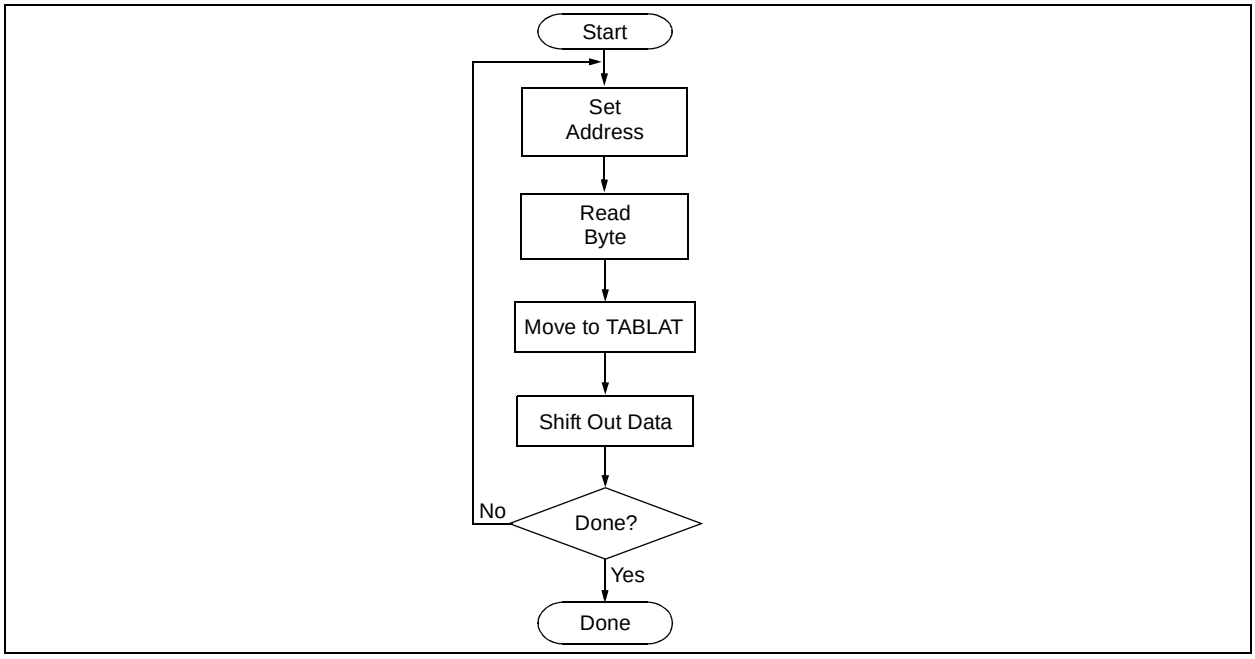


TABLE 4-2: READ DATA EEPROM MEMORY

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.

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

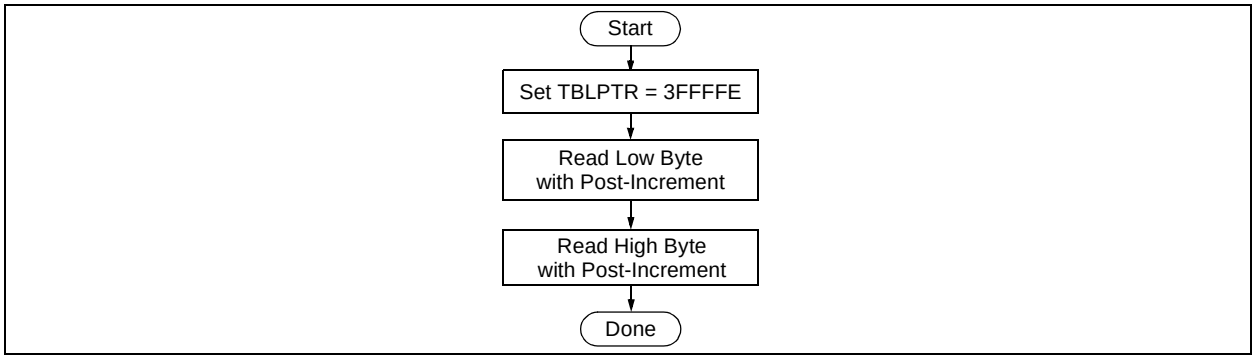


TABLE 5-2: DEVICE ID VALUES

Device	Device ID Value	
	DEVID2	DEVID1
PIC18F2221	21h	011x xxxx
PIC18F2321	21h	001x xxxx
PIC18F2410	11h	011x xxxx
PIC18F2420	11h	010x xxxx(1)
PIC18F2423	11h	010x xxxx(2)
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(2)
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(2)
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(1)
PIC18F4523	10h	100x xxxx(2)
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).

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

Bit Name	Configuration Words	Description
WDTEN	CONFIG2H	Watchdog Timer Enable bit 1 = WDT is enabled 0 = WDT is disabled (control is placed on the SWDTEN bit)
MCLRE	CONFIG3H	$\overline{\text{MCLR}}$ Pin Enable bit 1 = $\overline{\text{MCLR}}$ pin is enabled, RE3 input pin is disabled 0 = RE3 input pin is enabled, $\overline{\text{MCLR}}$ pin is disabled
LPT1OSC	CONFIG3H	Low-Power Timer1 Oscillator Enable bit 1 = Timer1 is configured for low-power operation 0 = Timer1 is configured for high-power operation
PBADEN	CONFIG3H	PORTB A/D Enable bit 1 = PORTB A/D<4:0> pins are configured as analog input channels on Reset 0 = PORTB A/D<4:0> pins are configured as digital I/O on Reset
PBADEN	CONFIG3H	PORTB A/D Enable bit **(PIC18FXX8X devices only)** 1 = PORTB A/D<4:0> and PORTB A/D<1:0> pins are configured as analog input channels on Reset 0 = PORTB A/D<4:0> pins are configured as digital I/O on Reset
CCP2MX	CONFIG3H	CCP2 MUX bit 1 = CCP2 input/output is multiplexed with RC1**(2)** 0 = CCP2 input/output is multiplexed with RB3
$\overline{\text{DEBUG}}$	CONFIG4L	Background Debugger Enable bit 1 = Background debugger is disabled, RB6 and RB7 are configured as general purpose I/O pins 0 = Background debugger is enabled, RB6 and RB7 are dedicated to In-Circuit Debug
XINST	CONFIG4L	Extended Instruction Set Enable bit 1 = Instruction set extension and Indexed Addressing mode are enabled 0 = Instruction set extension and Indexed Addressing mode are disabled (Legacy mode)
ICPRT	CONFIG4L	Dedicated In-Circuit (ICD/ICSP™) Port Enable bit **(PIC18F2455/2550/4455/4550, PIC18F2458/2553/4458/4553 and PIC18F2450/4450 devices only)** 1 = ICPORT is enabled 0 = ICPORT is disabled
BBSIZ<1:0>**(1)**	CONFIG4L	Boot Block Size Select bits **(PIC18F2585/2680/4585/4680 devices only)** 11 = 4K words (8 Kbytes) Boot Block 10 = 4K words (8 Kbytes) Boot Block 01 = 2K words (4 Kbytes) Boot Block 00 = 1K word (2 Kbytes) Boot Block
BBSIZ<2:1>**(1)**	CONFIG4L	Boot Block Size Select bits **(PIC18F2682/2685/4582/4685 devices only)** 11 = 4K words (8 Kbytes) Boot Block 10 = 4K words (8 Kbytes) Boot Block 01 = 2K words (4 Kbytes) Boot Block 00 = 1K word (2 Kbytes) Boot Block

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.

5.3 Single-Supply ICSP Programming

The LVP bit in Configuration register, CONFIG4L, enables Single-Supply (Low-Voltage) ICSP Programming. The LVP bit defaults to a '1' (enabled) from the factory.

If Single-Supply Programming mode is not used, the LVP bit can be programmed to a '0' and RB5/PGM becomes a digital I/O pin. However, the LVP bit may only be programmed by entering the High-Voltage ICSP mode, where MCLR/VPP/RE3 is raised to VIHH. Once the LVP bit is programmed to a '0', only the High-Voltage ICSP mode is available and only the High-Voltage ICSP mode can be used to program the device.

Note 1: The High-Voltage ICSP mode is always available, regardless of the state of the LVP bit, by applying VIHH to the MCLR/VPP/RE3 pin.

2: While in Low-Voltage ICSP mode, the RB5 pin can no longer be used as a general purpose I/O.

5.4 Embedding Configuration Word Information in the HEX File

To allow portability of code, a PIC18F2XXX/4XXX Family programmer is required to read the Configuration Word locations from the hex file. If Configuration Word information is not present in the hex file, then a simple warning message should be issued. Similarly, while saving a hex file, all Configuration Word information must be included. An option to not include the Configuration Word information may be provided. When embedding Configuration Word information in the hex file, it should start at address, 300000h.

Microchip Technology Inc. feels strongly that this feature is important for the benefit of the end customer.

5.5 Embedding Data EEPROM Information In the HEX File

To allow portability of code, a PIC18F2XXX/4XXX Family programmer is required to read the data EEPROM information from the hex file. If data EEPROM information is not present, a simple warning message should be issued. Similarly, when saving a hex file, all data EEPROM information must be included. An option to not include the data EEPROM information may be provided. When embedding data EEPROM information in the hex file, it should start at address, F00000h.

Microchip Technology Inc. believes that this feature is important for the benefit of the end customer.

5.6 Checksum Computation

The checksum is calculated by summing the following:

- The contents of all code memory locations
- The Configuration Words, appropriately masked
- ID locations (if any block is code-protected)

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

5.6.1 PROGRAM MEMORY

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

5.6.2 CONFIGURATION WORDS

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

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

TABLE 5-4: DEVICE BLOCK LOCATIONS AND SIZES

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

Legend: — = unimplemented.

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

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

Legend: — = unimplemented.

TABLE 5-5: CONFIGURATION WORD MASKS FOR COMPUTING CHECKSUMS (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.