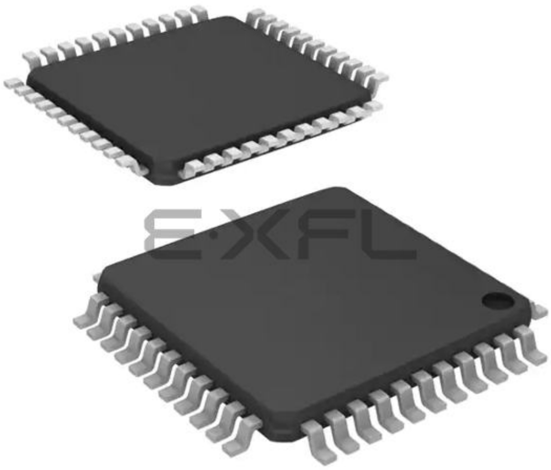


Welcome to E-XFL.COM

## What is "Embedded - Microcontrollers"?

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

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 40MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, HLVD, POR, PWM, WDT |
| Number of I/O | 36 |
| Program Memory Size | 8KB (4K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 512 x 8 |
| Voltage - Supply (Vcc/Vdd) | 4.2V ~ 5.5V |
| Data Converters | A/D 13x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-TQFP |
| Supplier Device Package | 44-TQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18f4321-i-pt |

For PIC18F2685/4685 devices, the code memory space extends from 0000h to 017FFFh (96 Kbytes) in five 16-Kbyte blocks. For PIC18F2682/4682 devices, the code memory space extends from 0000h to 0013FFFh (80 Kbytes) in four 16-Kbyte blocks. Addresses, 0000h through 0FFFh, however, define a “Boot Block” region that is treated separately from Block 0. All of these blocks define code protection boundaries within the code memory space.

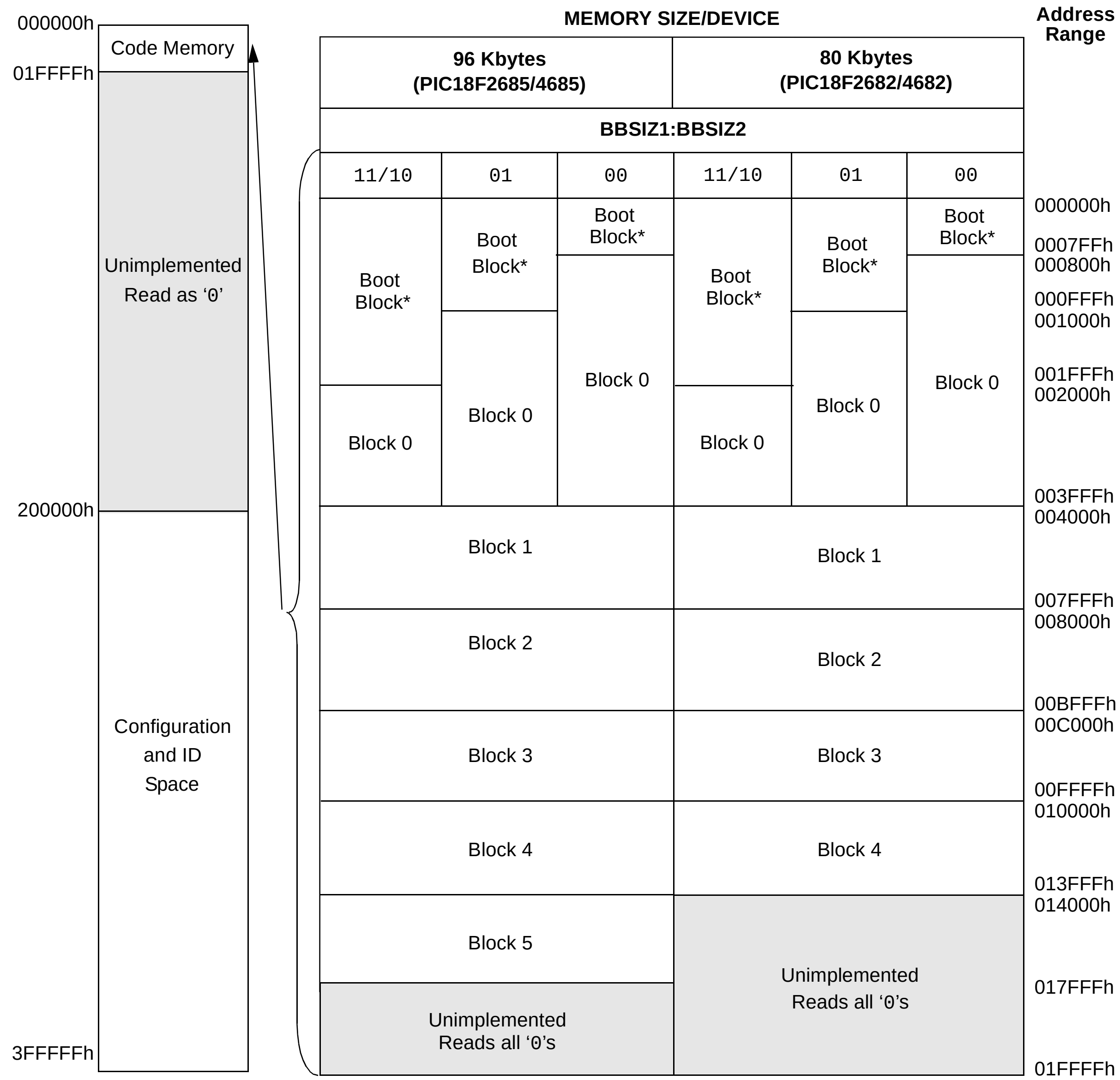
The size of the Boot Block in PIC18F2685/4685 and PIC18F2682/4682 devices can be configured as 1, 2 or 4K words (see Figure 2-7). This is done through the BBSIZ<2:1> bits in the Configuration register, CONFIG4L. It is important to note that increasing the size of the Boot Block decreases the size of Block 0.

**TABLE 2-3: IMPLEMENTATION OF CODE MEMORY**

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

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




**Note:** Sizes of memory areas are not to scale.

* Boot Block size is determined by the BBSIZ<1:2> bits in the CONFIG4L register.

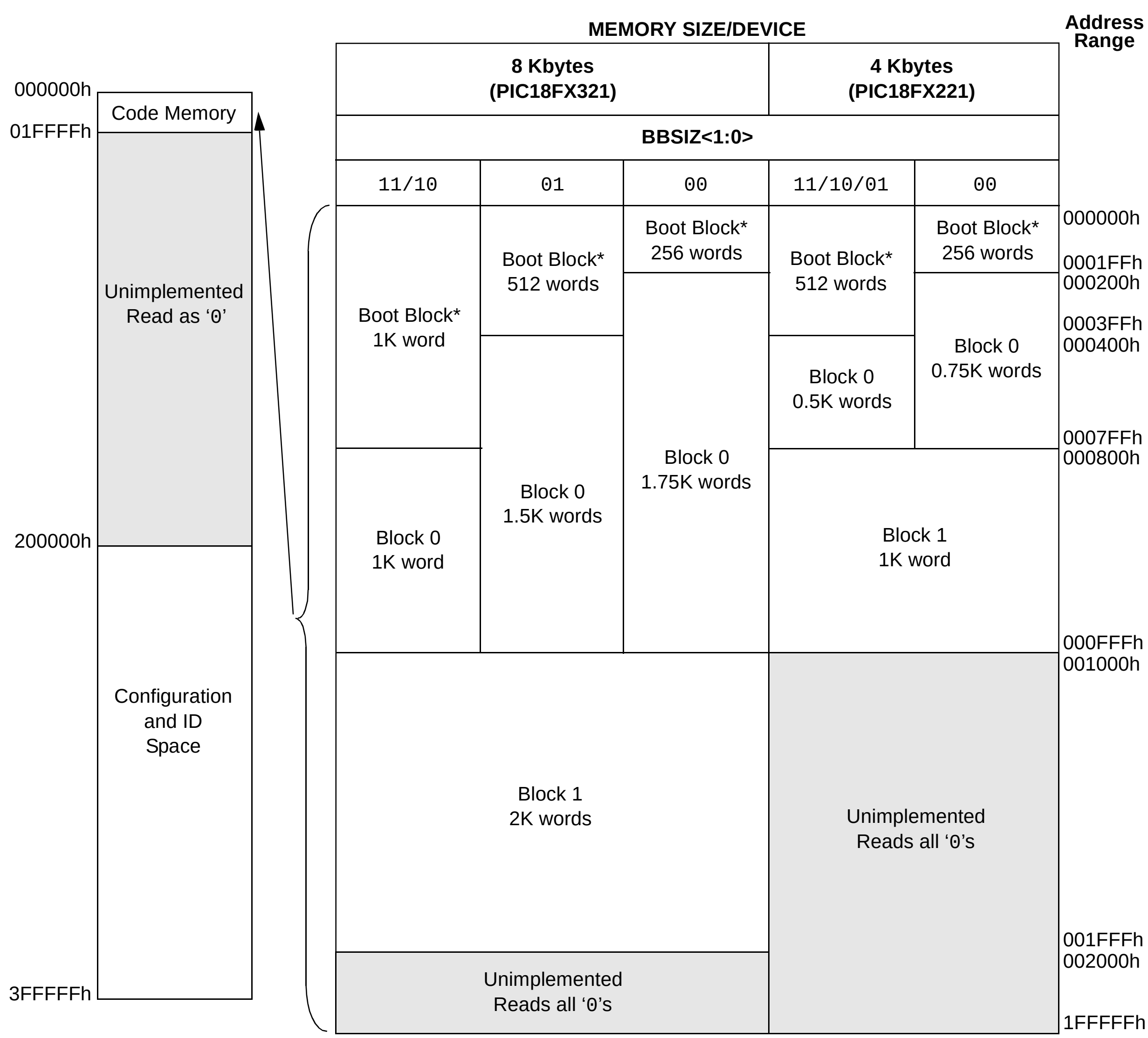
For PIC18FX5X0/X5X3 devices, the code memory space extends from 000000h to 007FFFh (32 Kbytes) in four 8-Kbyte blocks. For PIC18FX4X5/X4X8 devices, the code memory space extends from 000000h to 005FFFh (24 Kbytes) in three 8-Kbyte blocks. Addresses, 000000h through 0007FFh, however, define a “Boot Block” region that is treated separately from Block 0. All of these blocks define code protection boundaries within the code memory space.

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

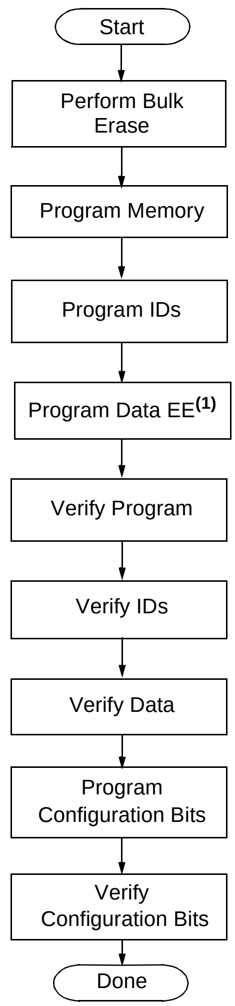



**Note:** Sizes of memory areas are not to scale.
* Boot Block size is determined by the BBSIZ<1:0> bits in the CONFIG4L register.

## 2.4 High-Level Overview of the Programming Process

Figure 2-13 shows the high-level overview of the programming process. First, a Bulk Erase is performed. Next, the code memory, ID locations and data EEPROM are programmed (selected devices only, see **Section 3.3 "Data EEPROM Programming"**). These memories are then verified to ensure that programming was successful. If no errors are detected, the Configuration bits are then programmed and verified.

**FIGURE 2-13: HIGH-LEVEL PROGRAMMING FLOW**




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

## 2.6 Entering and Exiting Low-Voltage ICSP Program/Verify Mode

When the LVP Configuration bit is '1' (see **Section 5.3 "Single-Supply ICSP Programming"**), the Low-Voltage ICSP mode is enabled. As shown in Figure 2-16, Low-Voltage ICSP Program/Verify mode is entered by holding PGC and PGD low, placing a logic high on PGM and then raising $\overline{MCLR}$/VPP/RE3 to VIH. In this mode, the RB5/PGM pin is dedicated to the programming function and ceases to be a general purpose I/O pin. Figure 2-17 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 2-16: ENTERING LOW-VOLTAGE PROGRAM/VERIFY MODE**

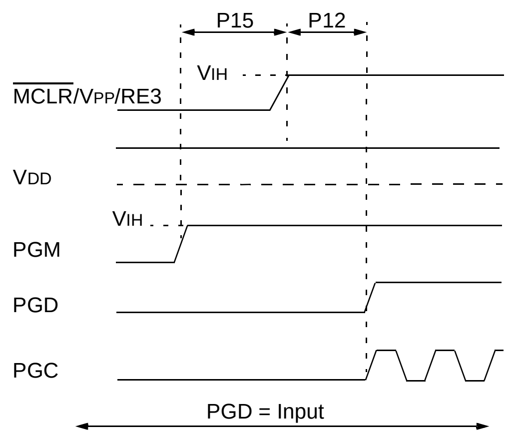



**FIGURE 2-17: EXITING LOW-VOLTAGE PROGRAM/VERIFY MODE**

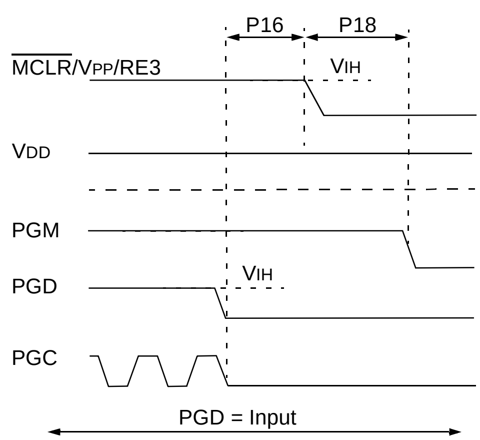
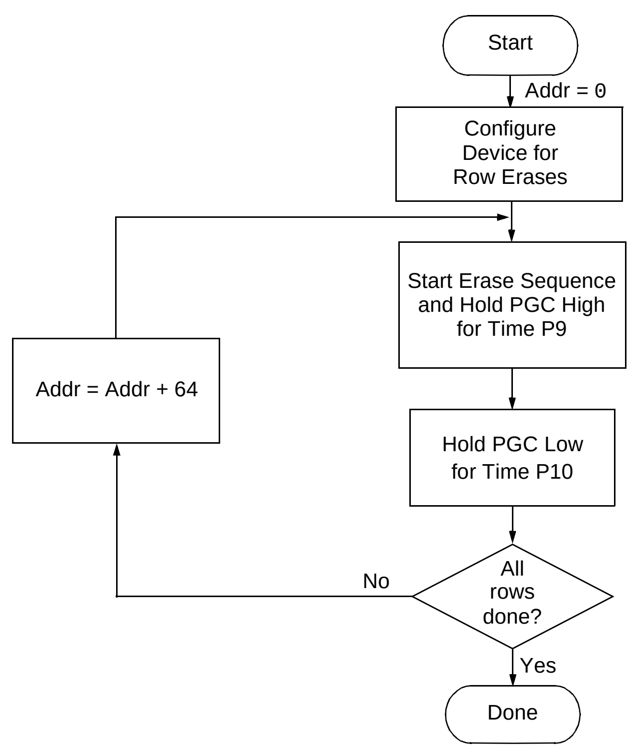


**TABLE 3-3: ERASE CODE MEMORY CODE SEQUENCE**

| 4-Bit Command | Data Payload | Core Instruction |
|---|---|---|
| Step 1: Direct access to code memory and enable writes. | | |
| 0000<br>0000<br>0000 | 8E A6<br>9C A6<br>84 A6 | BSF EECON1, EEPGD<br>BCF EECON1, CFGS<br>BSF EECON1, WREN |
| Step 2: Point to first row in code memory. | | |
| 0000<br>0000<br>0000 | 6A F8<br>6A F7<br>6A F6 | CLRF TBLPTRU<br>CLRF TBLPTRH<br>CLRF TBLPTRL |
| Step 3: Enable erase and erase single row. | | |
| 0000<br>0000<br>0000 | 88 A6<br>82 A6<br>00 00 | BSF EECON1, FREE<br>BSF EECON1, WR<br>NOP – hold PGC high for time P9 and low for time P10. |
| Step 4: Repeat Step 3, with the Address Pointer incremented by 64 until all rows are erased. | | |

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

**TABLE 3-5: WRITE CODE MEMORY CODE SEQUENCE**

| 4-Bit Command | Data Payload | Core Instruction |
|---|---|---|
| Step 1: Direct access to code memory and enable writes. | | |
| 0000<br>0000 | 8E A6<br>9C A6 | BSF EECON1, EEPGD<br>BCF EECON1, CFGS |
| Step 2: Load write buffer. | | |
| 0000<br>0000<br>0000<br>0000<br>0000<br>0000 | 0E <Addr[21:16]><br>6E F8<br>0E <Addr[15:8]><br>6E F7<br>0E <Addr[7:0]><br>6E F6 | MOVLW <Addr[21:16]><br>MOVWF TBLPTRU<br>MOVLW <Addr[15:8]><br>MOVWF TBLPTRH<br>MOVLW <Addr[7:0]><br>MOVWF TBLPTRL |
| Step 3: Repeat for all but the last two bytes. | | |
| 1101 | <MSB><LSB> | Write 2 bytes and post-increment address by 2. |
| Step 4: Load write buffer for last two bytes. | | |
| 1111<br>0000 | <MSB><LSB><br>00 00 | Write 2 bytes and start programming.<br>NOP - hold PGC high for time P9 and low for time P10. |
| To continue writing data, repeat Steps 2 through 4, where the Address Pointer is incremented by 2 at each iteration of the loop. | | |

**FIGURE 3-4: PROGRAM CODE MEMORY FLOW**

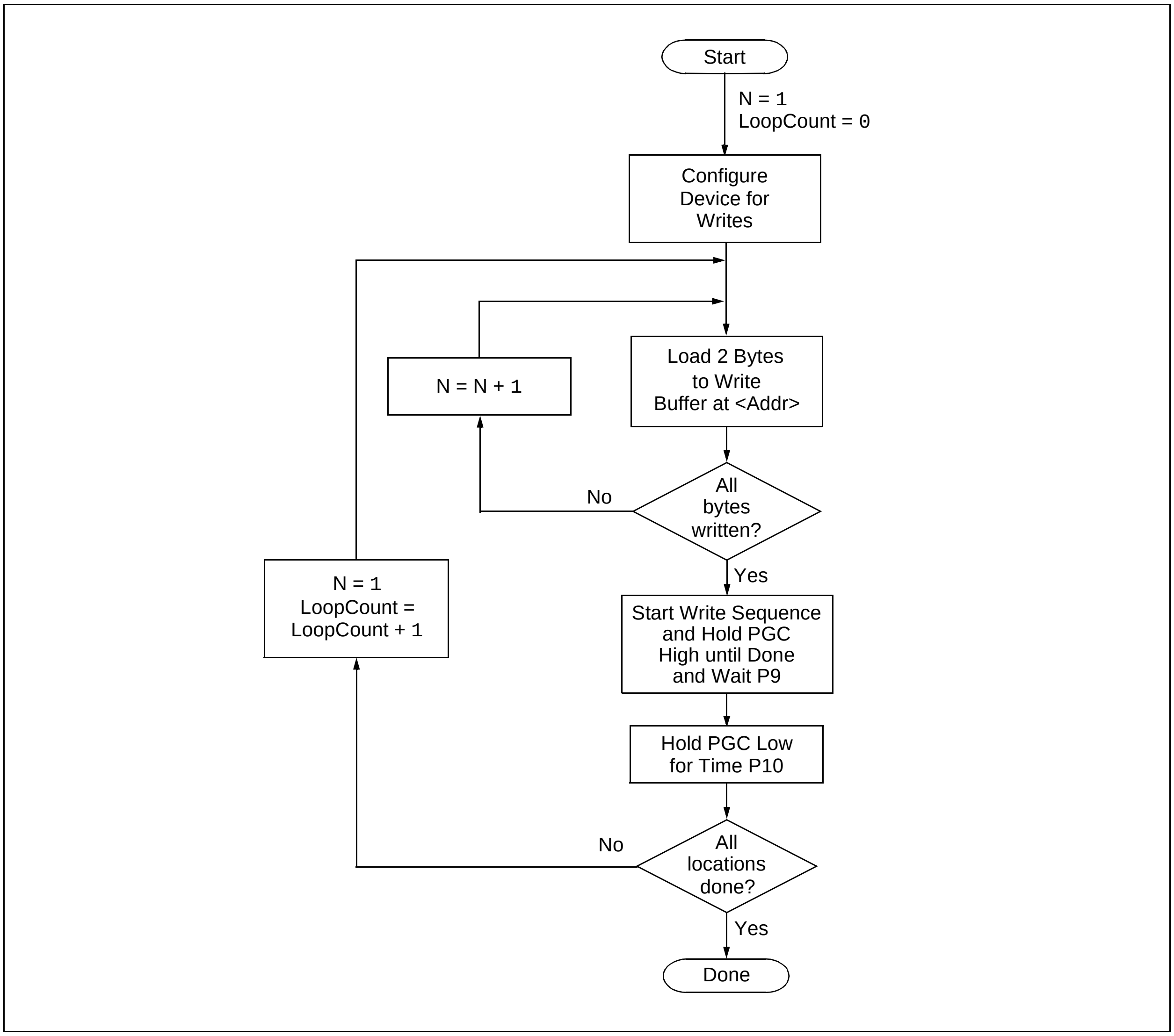
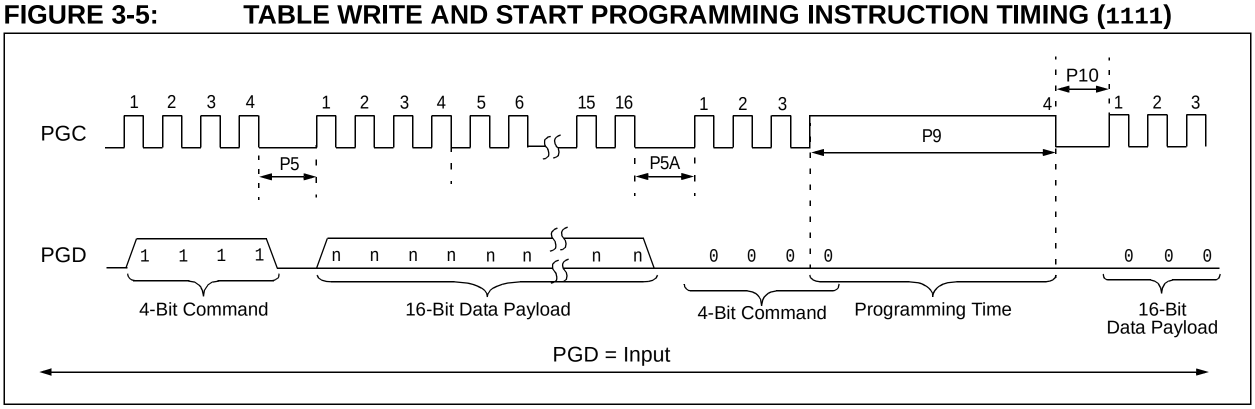



**FIGURE 3-5: TABLE WRITE AND START PROGRAMMING INSTRUCTION TIMING (1111)**



### 3.2.1 MODIFYING CODE MEMORY

The previous programming example assumed that the device had been Bulk Erased prior to programming (see **Section 3.1.1 "High-Voltage ICSP Bulk Erase"**). It may be the case, however, that the user wishes to modify only a section of an already programmed device.

The appropriate number of bytes required for the erase buffer must be read out of code memory (as described in **Section 4.2 "Verify Code Memory and ID Locations"**) and buffered. Modifications can be made on this buffer. Then, the block of code memory that was read out must be erased and rewritten with the modified data.

The WREN bit must be set if the WR bit in EECON1 is used to initiate a write sequence.

**TABLE 3-6: MODIFYING CODE MEMORY**

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

# 4.0 READING THE DEVICE

## 4.1 Read Code Memory, ID Locations and Configuration Bits

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

The 4-bit command is shifted in, LSb first. The read is executed during the next eight clocks, then shifted out on PGD during the last eight clocks, LSb to MSb. A delay of P6 must be introduced after the falling edge of the 8th PGC of the operand to allow PGD to transition from an input to an output. During this time, PGC must be held low (see Figure 4-1). This operation also increments the Table Pointer by one, pointing to the next byte in code memory for the next read.

This technique will work to read any memory in the 000000h to 3FFFFFh address space, so it also applies to the reading of the ID and Configuration registers.

**TABLE 4-1: READ CODE MEMORY SEQUENCE**

| 4-Bit Command | Data Payload | Core Instruction |
|---|---|---|
| Step 1: Set Table Pointer. | | |
| 0000<br>0000<br>0000<br>0000<br>0000<br>0000 | 0E <Addr[21:16]><br>6E F8<br>0E <Addr[15:8]><br>6E F7<br>0E <Addr[7:0]><br>6E F6 | MOVLW Addr[21:16]<br>MOVWF TBLPTRU<br>MOVLW <Addr[15:8]><br>MOVWF TBLPTRH<br>MOVLW <Addr[7:0]><br>MOVWF TBLPTRL |
| Step 2: Read memory and then shift out on PGD, LSb to MSb. | | |
| 1001 | 00 00 | TBLRD *+ |

**FIGURE 4-1: TABLE READ POST-INCREMENT INSTRUCTION TIMING (1001)**

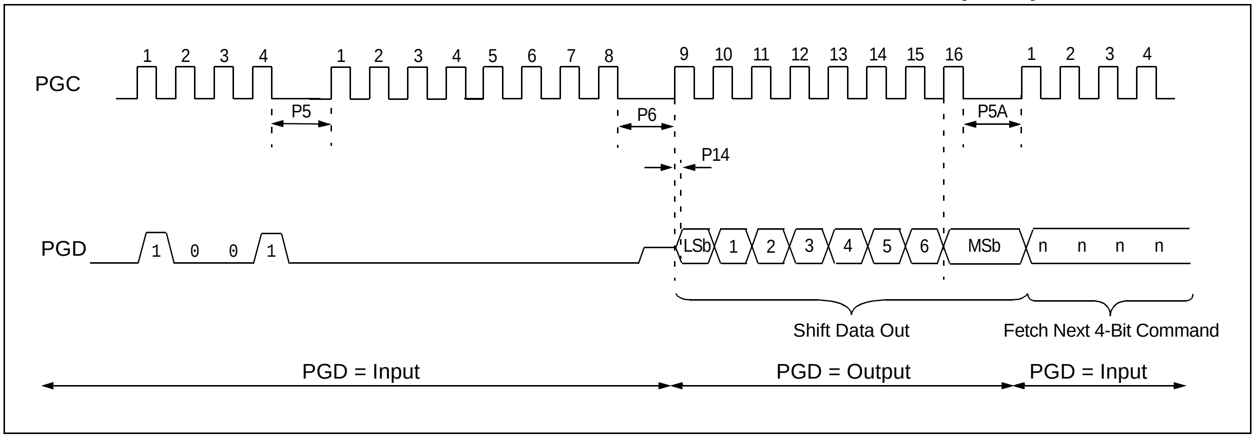
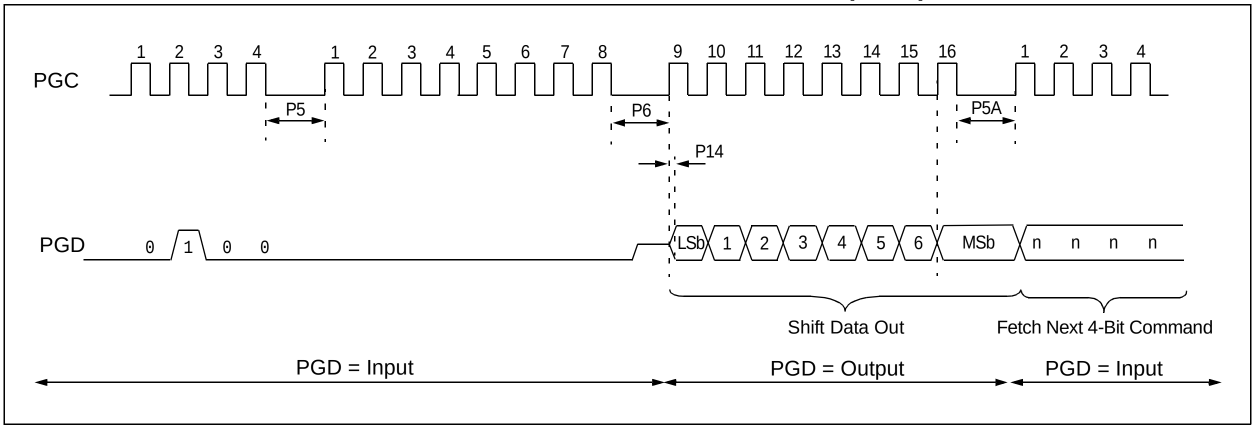


**FIGURE 4-4: SHIFT OUT DATA HOLDING REGISTER TIMING (0010)**




## 4.5 Verify Data EEPROM

A data EEPROM address may be read via a sequence of core instructions (4-bit command, '0000') and then output on PGD via the 4-bit command, '0010' (TABLAT register). The result may then be immediately compared to the appropriate data in the programmer's memory for verification. Refer to **Section 4.4 "Read Data EEPROM Memory"** for implementation details of reading data EEPROM.

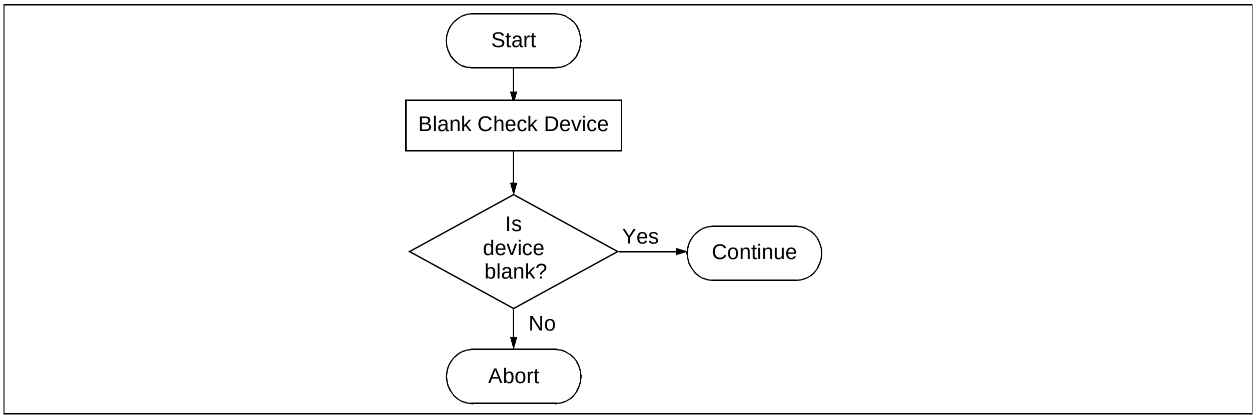
## 4.6 Blank Check

The term Blank Check means to verify that the device has no programmed memory cells. All memories must be verified: code memory, data EEPROM, ID locations and Configuration bits. The Device ID registers (3FFFFEh:3FFFFFh) should be ignored.

A "blank" or "erased" memory cell will read as '1'. Therefore, Blank Checking a device merely means to verify that all bytes read as FFh, except the Configuration bits. Unused (reserved) Configuration bits will read '0' (programmed). Refer to Figure 4-5 for blank configuration expect data for the various PIC18F2XXX/4XXX Family devices.

Given that Blank Checking is merely code and data EEPROM verification with FFh expect data, refer to **Section 4.4 "Read Data EEPROM Memory"** and **Section 4.2 "Verify Code Memory and ID Locations"** for implementation details.

**FIGURE 4-5: BLANK CHECK FLOW**



## 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**

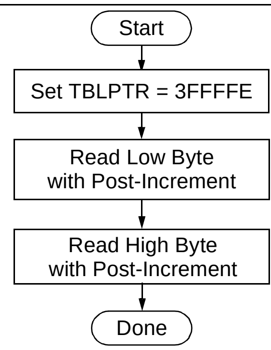


**TABLE 5-1: CONFIGURATION BITS AND DEVICE IDS**

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

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

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

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

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

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

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

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

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

**8:** Implemented on PIC18F2450/4450 devices only.

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

**10:** Implemented on PIC18F2685/4685 devices only.

**TABLE 5-2: 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**

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

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

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

**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)<br>1 = Block 0 is not protected from Table Reads executed in other blocks<br>0 = Block 0 is protected from Table Reads executed in other blocks |
| EBTRB | CONFIG7H | Table Read Protection bit (Boot Block memory area)<br>1 = Boot Block is not protected from Table Reads executed in other blocks<br>0 = Boot Block is protected from Table Reads executed in other blocks |
| DEV<10:3> | DEVID2 | Device ID bits<br>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<br>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<br>These bits are used to indicate the revision of the device. The REV4 bit is sometimes used to fully specify the device type. |

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

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

### TABLE 5-4: DEVICE BLOCK LOCATIONS AND SIZES

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

**Legend:** — = unimplemented.

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

| **Standard Operating Conditions**<br>Operating Temperature: 25°C is recommended | | | | | | |
|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Max** | **Units** | **Conditions** |
| P11A | $T_{DRWT}$ | Data Write Polling Time | 4 | — | ms | |
| P12 | $T_{HLD2}$ | Input Data Hold Time from $\overline{MCLR}$/$V_{PP}$/RE3 ↑ | 2 | — | μs | |
| P13 | $T_{SET2}$ | $V_{DD}$ ↑ Setup Time to $\overline{MCLR}$/$V_{PP}$/RE3 ↑ | 100 | — | ns | **(Note 2)** |
| P14 | $T_{VALID}$ | Data Out Valid from PGC ↑ | 10 | — | ns | |
| P15 | $T_{SET3}$ | PGM ↑ Setup Time to $\overline{MCLR}$/$V_{PP}$/RE3 ↑ | 2 | — | μs | **(Note 2)** |
| P16 | $T_{DLY8}$ | Delay Between Last PGC ↓ and $\overline{MCLR}$/$V_{PP}$/RE3 ↓ | 0 | — | s | |
| P17 | $T_{HLD3}$ | $\overline{MCLR}$/$V_{PP}$/RE3 ↓ to $V_{DD}$ ↓ | — | 100 | ns | |
| P18 | $T_{HLD4}$ | $\overline{MCLR}$/$V_{PP}$/RE3 ↓ to PGM ↓ | 0 | — | s | |

**Note 1:** Do not allow excess time when transitioning $\overline{MCLR}$ between $V_{IL}$ and $V_{IHH}$. This can cause spurious program executions to occur. The maximum transition time is:
1 $T_{CY}$ + $T_{PWRT}$ (if enabled) + 1024 $T_{OSC}$ (for LP, HS, HS/PLL and XT modes only) +
2 ms (for HS/PLL mode only) + 1.5 μs (for EC mode only)
where $T_{CY}$ is the instruction cycle time, $T_{PWRT}$ is the Power-up Timer period and $T_{OSC}$ is the oscillator period. For specific values, refer to the Electrical Characteristics section of the device data sheet for the particular device.

**2:** When ICPRT = 1, this specification also applies to $ICV_{PP}$.

**3:** At 0°C-50°C.