# PIC12F6XX/16F6XX

## PIC12F6XX/16F6XX Memory Programming Specification

**This document includes the programming specifications for the following device:**

- PIC12F635
- PIC12F683
- PIC16F631
- PIC16F636
- PIC16F639
- PIC16F677
- PIC16F684
- PIC16F685
- PIC16F687
- PIC16F688
- PIC16F689
- PIC16F690

## 1.0 PROGRAMMING THE PIC12F6XX/16F6XX DEVICES

The PIC12F6XX/16F6XX devices are programmed using a serial method. The Serial mode will allow the PIC12F6XX/16F6XX devices to be programmed while in the user's system. This programming specification applies to the PIC12F6XX/16F6XX devices in all packages.

### 1.1 Hardware Requirements

PIC12F6XX/16F6XX devices require one power supply for VDD (5.0V) and one for VPP (12.0V).

### 1.2 Program/Verify Mode

The Program/Verify mode for the PIC12F6XX/16F6XX devices allow programming of user program memory, data memory, user ID locations and the Configuration Word.

Programming and verification can take place on any memory region, independent of the remaining regions. This allows independent programming of program and data memory regions. Therefore, unprotected data memory can be reprogrammed and protected without losing the content in the program memory.

**TABLE 1-1: PIN DESCRIPTIONS IN PROGRAM/VERIFY MODE**

| Pin Name | During Programming | | |
|---|---|---|---|
| | Function | Pin Type | Pin Description |
| GP1/RA1 | ICSPCLK | I | Clock input – Schmitt Trigger input |
| GP0/RA0 | ICSPDAT | I/O | Data input/output – Schmitt Trigger input |
| $\overline{MCLR}$ | Program/Verify mode | P(1) | Program Mode Select |
| VDD | VDD | P | Power Supply |
| VSS | VSS | P | Ground |

**Legend:** I = Input, O = Output, P = Power

**Note 1:** In the PIC12F6XX/16F6XX, the programming high voltage is internally generated. To activate the Program/Verify mode, high voltage needs to be applied to $\overline{MCLR}$ input. Since the $\overline{MCLR}$ is used for a level source, $\overline{MCLR}$ does not draw any significant current.

**FIGURE 1-1: 8-PIN PDIP, SOIC, DFN-S DIAGRAM FOR PIC12F635/683**

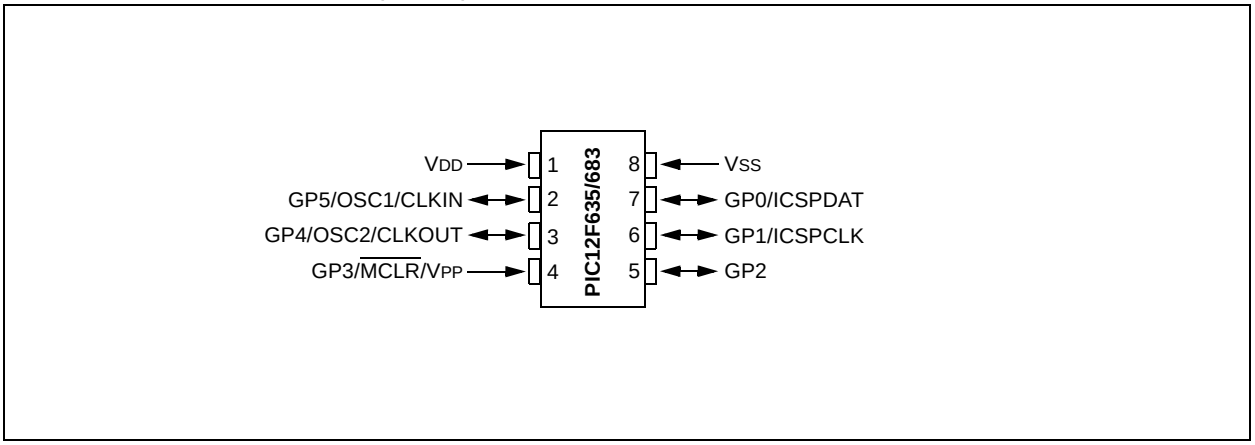



**FIGURE 1-2: 14-PIN PDIP, SOIC, TSSOP DIAGRAM FOR PIC16F636/684/688**

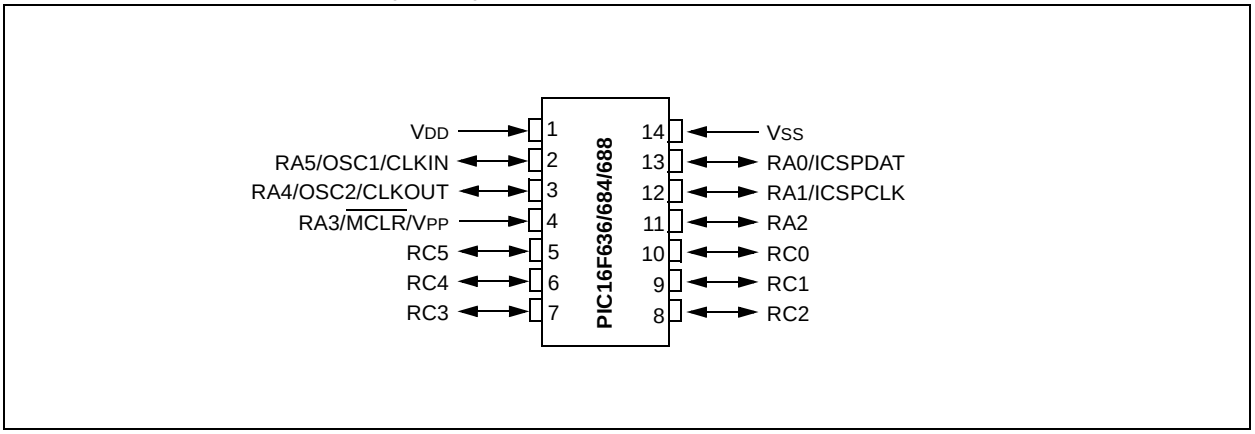



**FIGURE 1-3: 16-PIN QFN DIAGRAMS FOR PIC16F636/684/688**

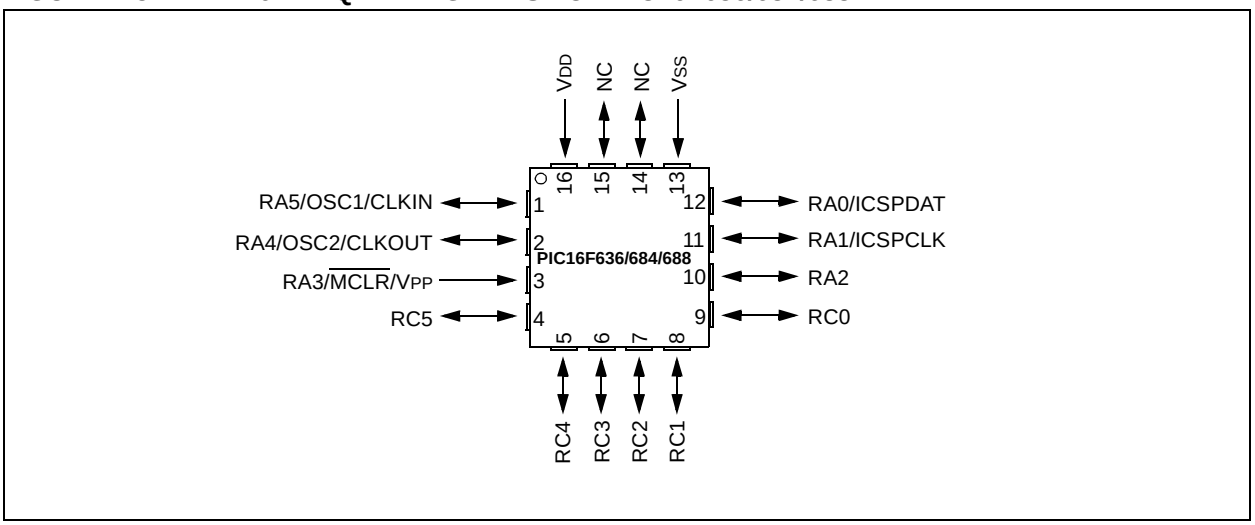
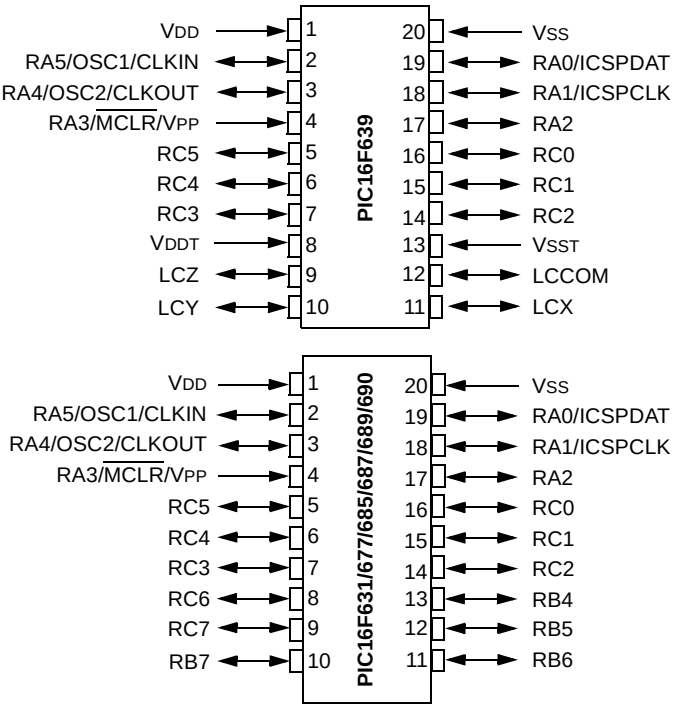
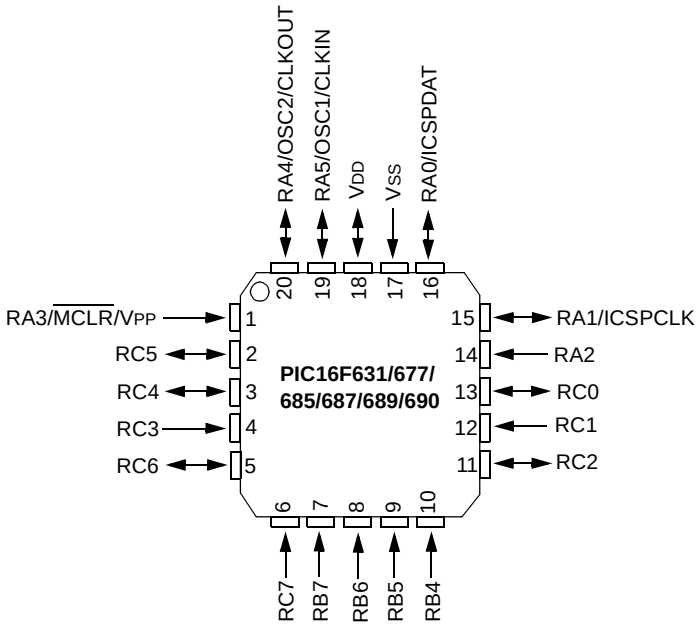


**FIGURE 1-4: 20-PIN PDIP, SOIC(1), SSOP DIAGRAMS**




**Note 1:** PIC16F631, PIC16F677, PIC16F685, PIC16F687, PIC16F689 and PIC16F690 only.

**FIGURE 1-5: 20-PIN QFN DIAGRAM FOR PIC16F631/677/685/687/689/690**



## 2.0 MEMORY DESCRIPTION

### 2.1 Program Memory Map

The user memory space extends from 0x0000 to 0x1FFF. In Program/Verify mode, the program memory space extends from 0x0000 to 0x3FFF, with the first half (0x0000-0x1FFF) being user program memory and the second half (0x2000-0x3FFF) being configuration memory. The PC will increment from 0x0000 to 0x1FFF and wrap to 0x000, 0x2000 to 0x3FFF and wraparound to 0x2000 (not to 0x0000). Once in configuration memory, the highest bit of the PC stays a '1', thus always pointing to the configuration memory. The only way to point to user program memory is to reset the part and re-enter Program/Verify mode as described in **Section 3.0 "Program/Verify Mode"**.

For the PIC12F6XX/16F6XX (not including PIC12F635/636/639) devices, the configuration memory space, 0x2000 to 0x2008 are physically implemented. However, only locations 0x2000 to 0x2003, 0x2007 and 0x2008 are available. Other locations are reserved.

For the PIC12F635/636/639 devices, the configuration memory space (0x2000-0x2009) are physically implemented. However, only locations 0x2000 to 0x2003 and locations 0x2006 to 0x2009 are available. Other locations are reserved.

### 2.2 User ID Locations

A user may store identification information (user ID) in four designated locations. The user ID locations are mapped in 0x2000 to 0x2003. It is recommended that the user use only the seven Least Significant bits (LSb) of each user ID location. The user ID locations read out normally, even after code protection is enabled. It is recommended that ID locations are written as 'xx xxxx xbbb bbbb' where 'bbb bbbb' is user ID information.

The 14 bits may be programmed, but only the 7 LSb's are displayed by MPLAB® IDE. The xxxx's are "don't care" bits and are not read by MPLAB® IDE.

### 2.3 Calibration Word

For the PIC16F631/677/685/687/689/690 (not including PIC12F635/636/639) devices, the 8 MHz Internal Oscillator (INTOSC), the Power-on Reset (POR) and the Brown-out Reset (BOR) modules are factory calibrated. These values are stored in the Calibration Word (0x2008). See the applicable device data sheet for more information.

For the PIC12F635/636/639 devices, the 8 MHz Internal Oscillator (INTOSC), the Power-on Reset and the Brown-out Reset modules are factory calibrated and stored in the Calibration Word (0x2008). The Wake-up Reset (WUR) and Low-Voltage Detect (LVD) modules are factory calibrated and stored in the Calibration Word (0x2009). See the applicable device data sheet for more information.

The Calibration Word locations are written at the time of manufacturing and are not erased when a Bulk Erase is performed. See **Section 3.1.5.10 "Bulk Erase Program Memory"** for more information on the various erase sequences. However, it is possible to inadvertently write to these locations. The device may not function properly or may operate outside of specifications if the Calibration Word locations do not contain the correct value. Therefore, it is recommended that the Calibration Words be read prior to any programming procedure and verified after programming is complete. See Figure 3-21 for a flowchart of the recommended verification procedure.

The device should not be used if the verification of the Calibration Word values fail after the device is programmed. The 0x3FFF value is a special case, it is a valid calibration value but, it is also the erased state of the register.

**TABLE 1: MEMORY CAPACITY**

| Device | EEDATA | Program Flash |
|---|---|---|
| PIC12F635 | 128 x 8 | 1k x 14 |
| PIC12F683 | 256 x 8 | 2k x 14 |
| PIC16F631 | 128 x 8 | 1k x 14 |
| PIC16F636 | 256 x 8 | 2k x 14 |
| PIC16F639 | 256 x 8 | 2k x 14 |
| PIC16F677 | 256 x 8 | 2k x 14 |
| PIC16F684 | 256 x 8 | 2k x 14 |
| PIC16F685 | 256 x 8 | 4k x 14 |
| PIC16F687 | 256 x 8 | 2k x 14 |
| PIC16F688 | 256 x 8 | 4k x 14 |
| PIC16F689 | 256 x 8 | 4k x 14 |
| PIC16F690 | 256 x 8 | 4k x 14 |

**FIGURE 2-1: PIC16F631 PROGRAM MEMORY MAPPING**

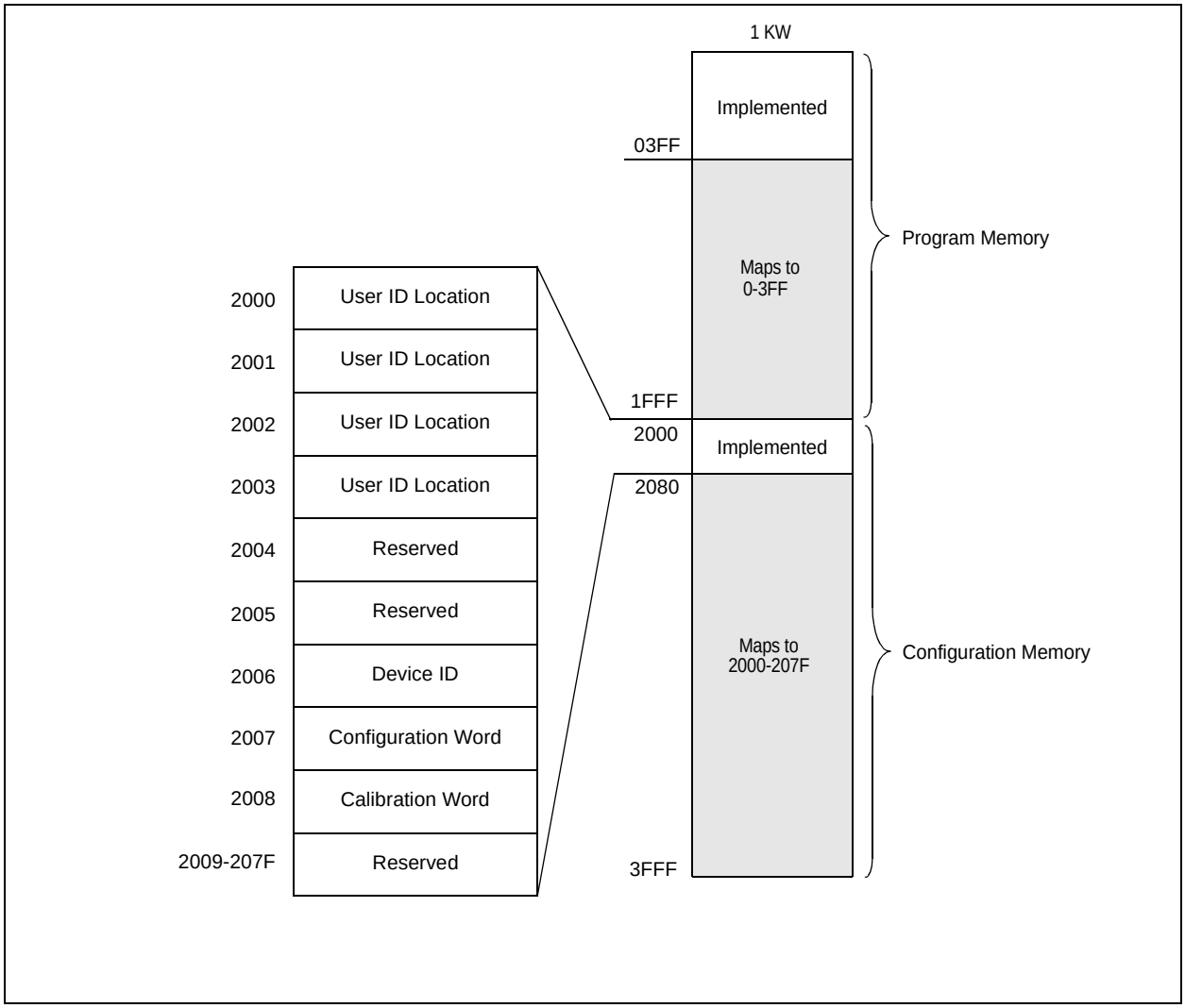


**FIGURE 2-2: PIC12F635 PROGRAM MEMORY MAPPING**

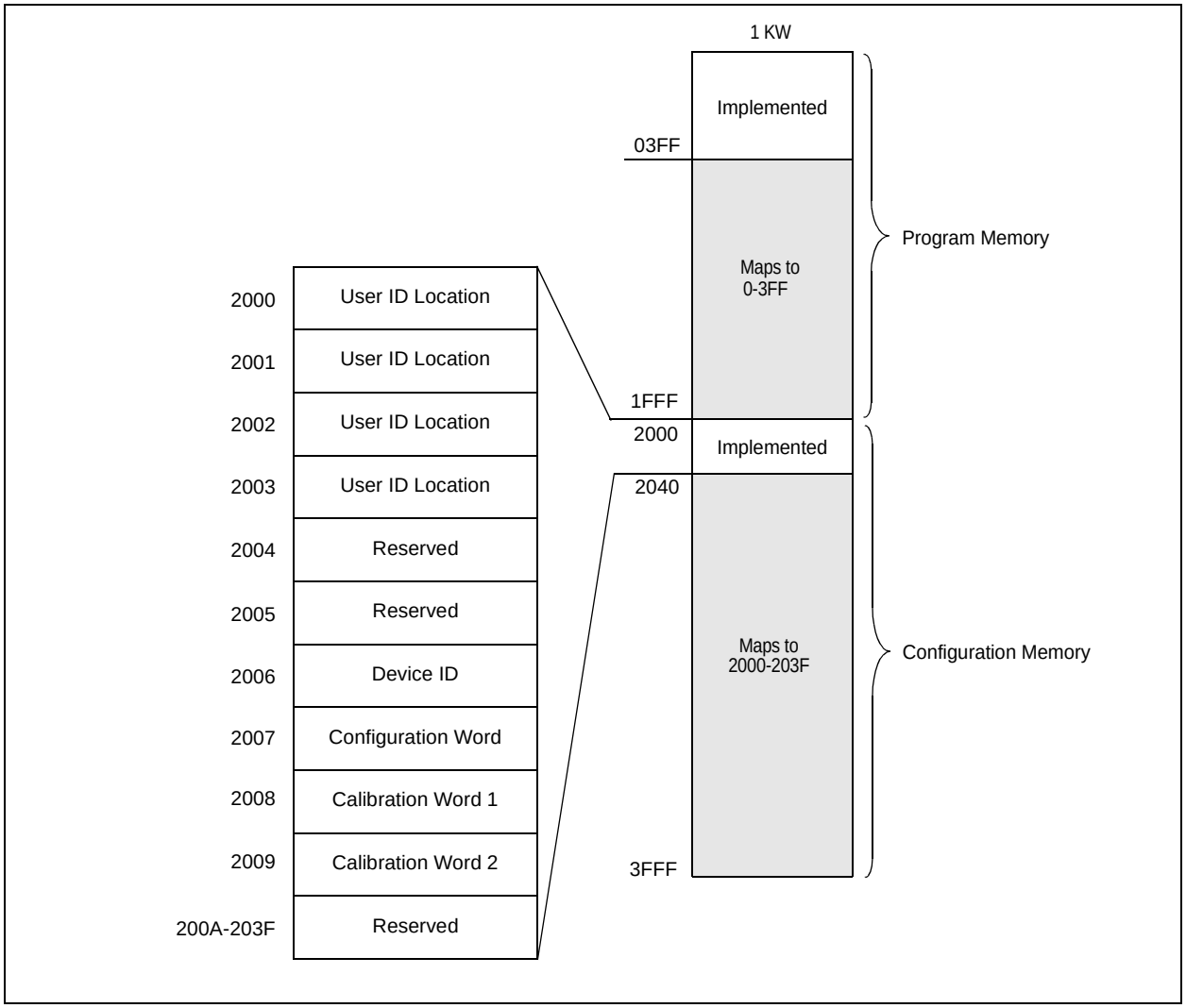


**FIGURE 2-3: PIC16F636/639 PROGRAM MEMORY MAPPING**

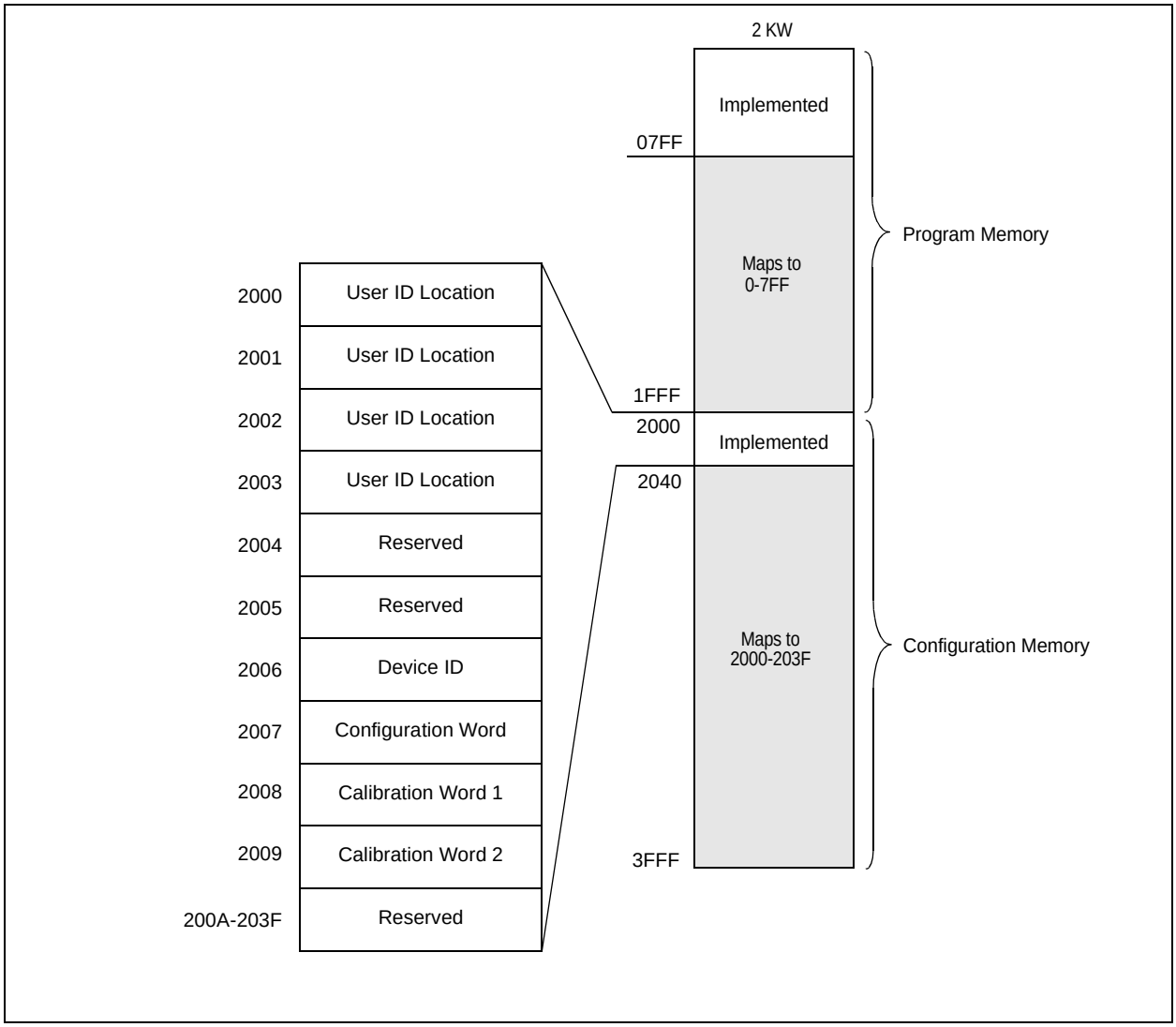


**FIGURE 2-4: PIC12F683/684 PROGRAM MEMORY MAPPING**

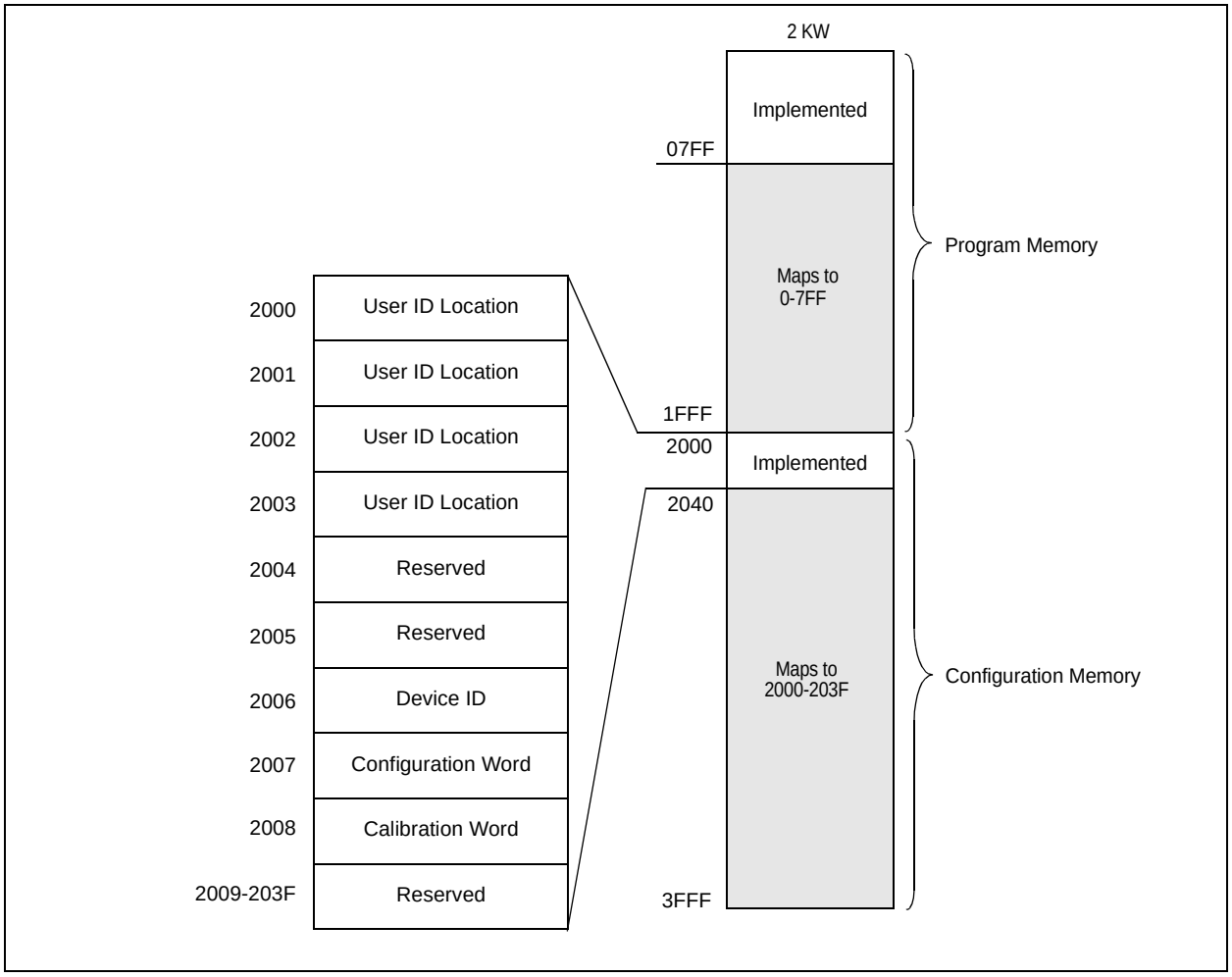


**FIGURE 2-5: PIC16F677/687 PROGRAM MEMORY MAPPING**

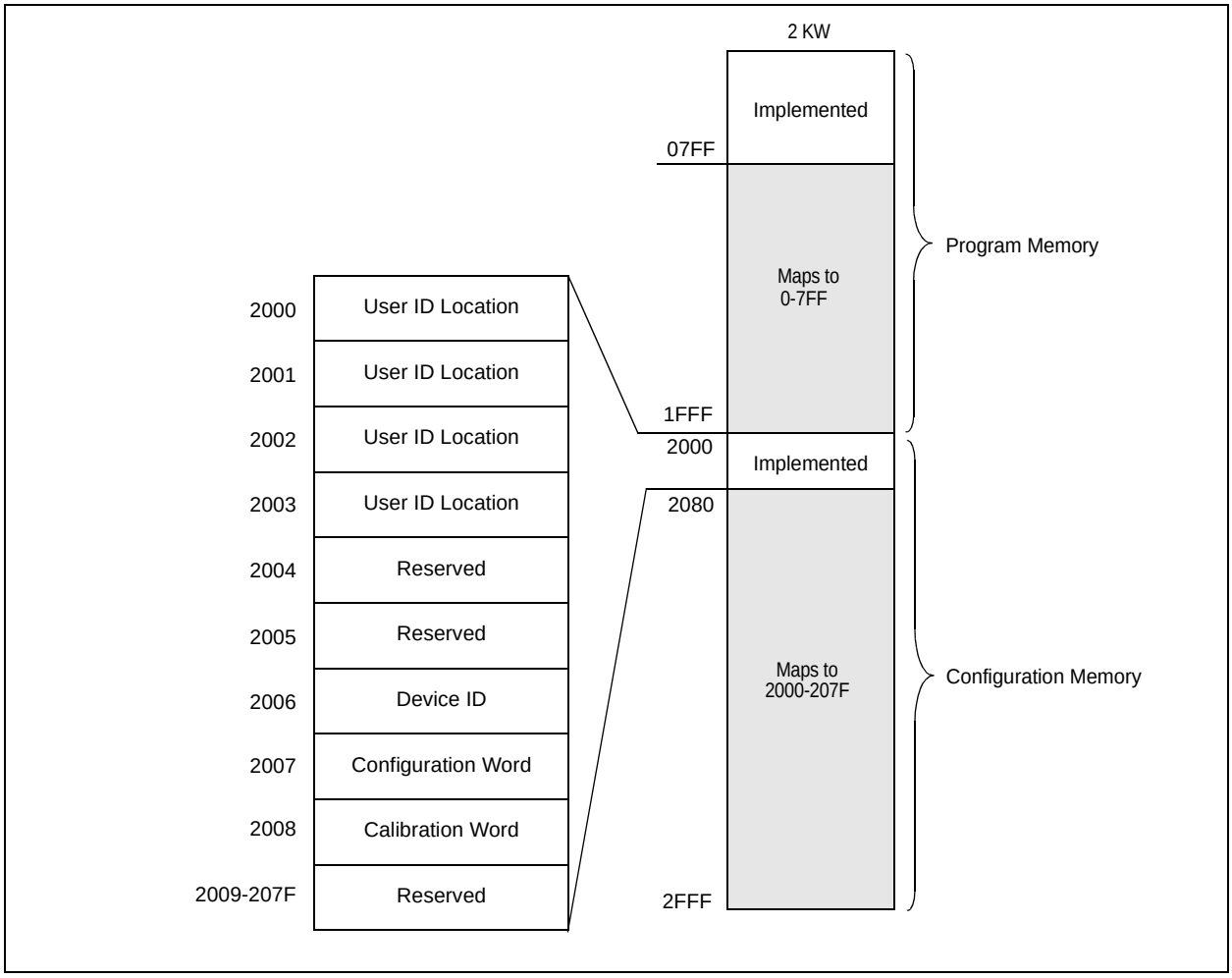


**FIGURE 2-6: PIC16F688 PROGRAM MEMORY MAPPING**

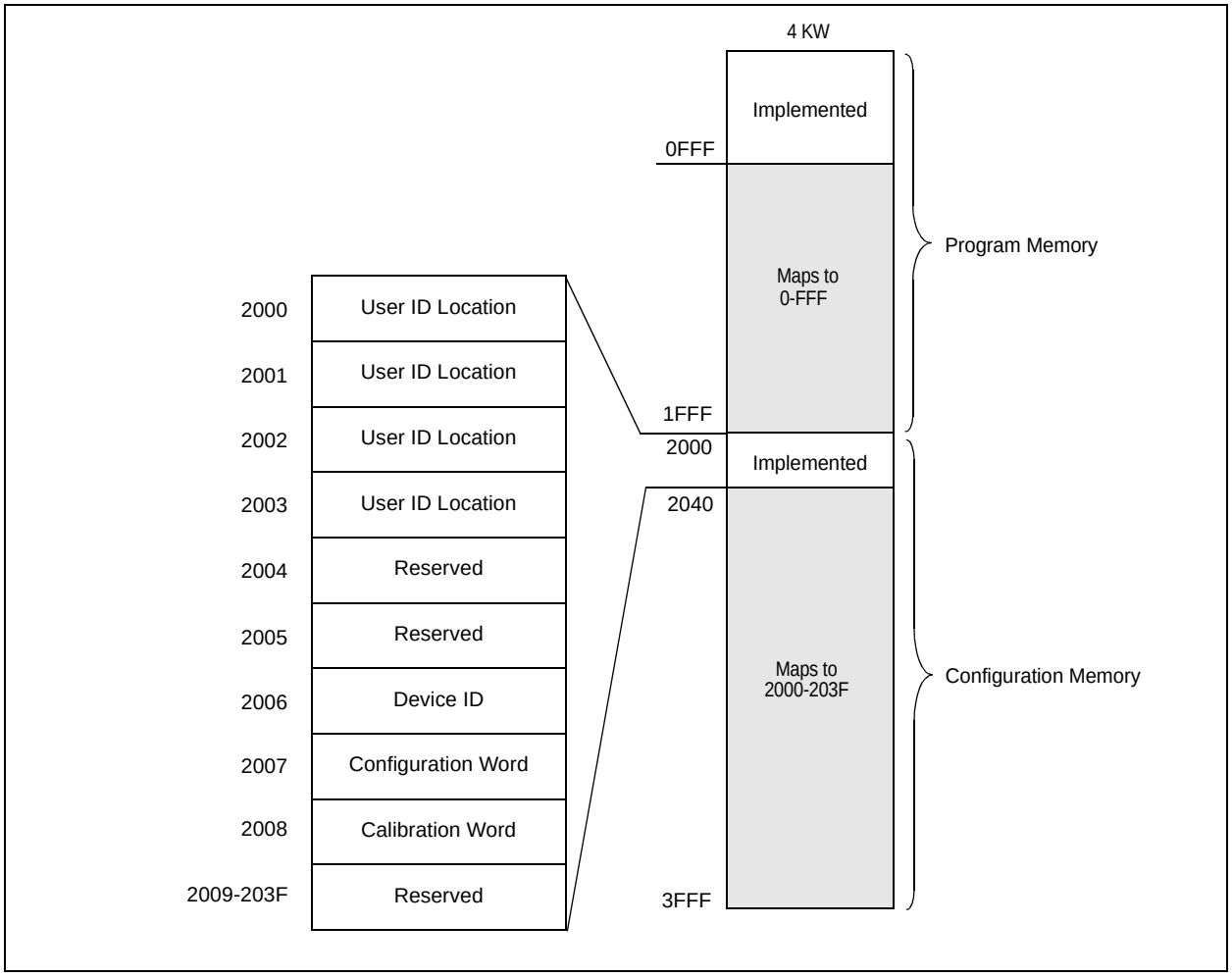
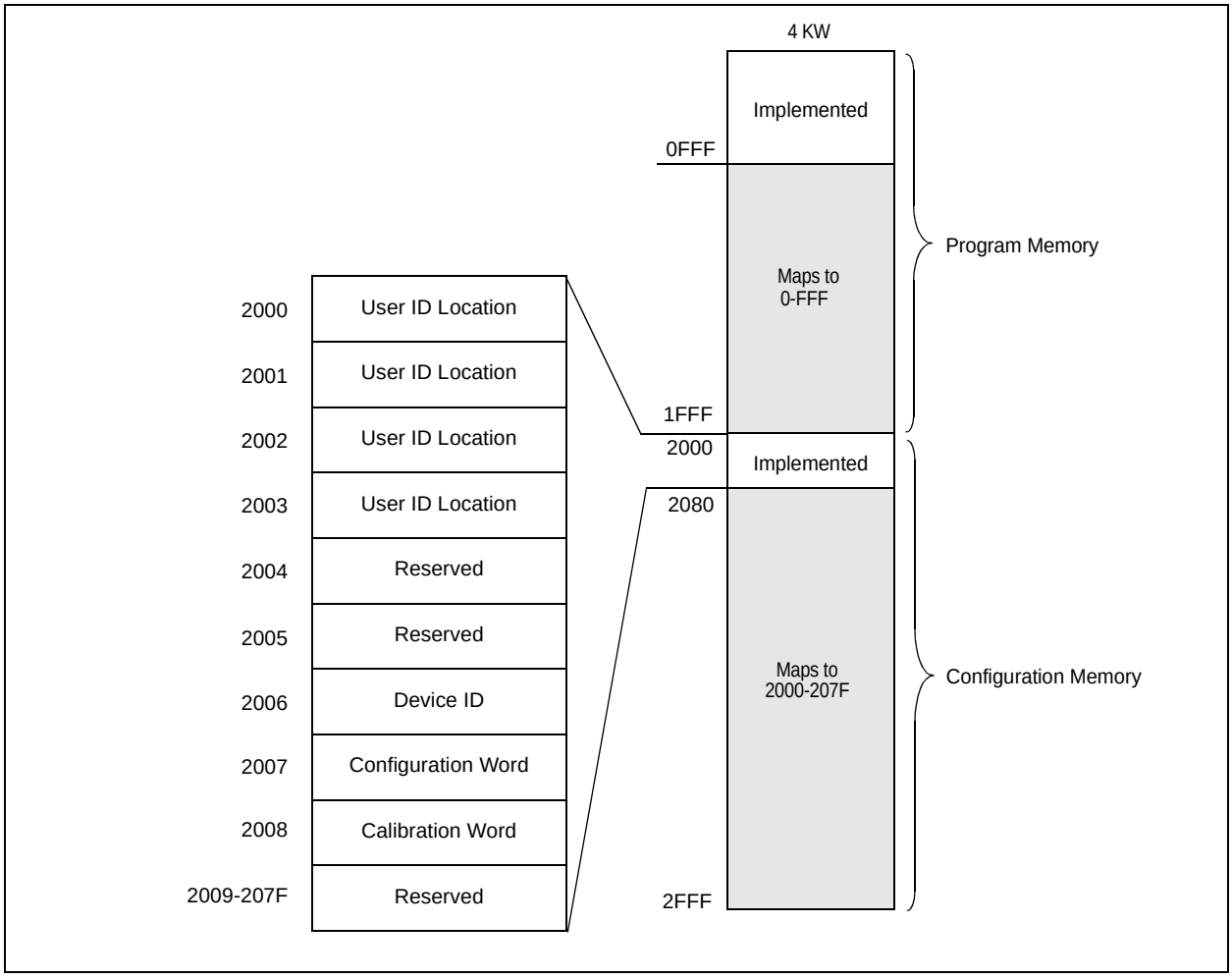


**FIGURE 2-7: PIC16F685/689/690 PROGRAM MEMORY MAPPING**



## 3.0 PROGRAM/VERIFY MODE

Two methods are available to enter Program/Verify mode. The "VPP-first" is entered by holding ICSPDAT and ICSPCLK low while raising MCLR pin from VIL to VIHH (high voltage), then applying VDD and data. This method can be used for any Configuration Word selection and **must** be used if the INTOSC and internal MCLR options are selected (FOSC<2:0> = 100 or 101 and MCLRE = 0). The VPP-first entry prevents the device from executing code prior to entering Program/Verify mode. See the timing diagram in Figure 3-1.

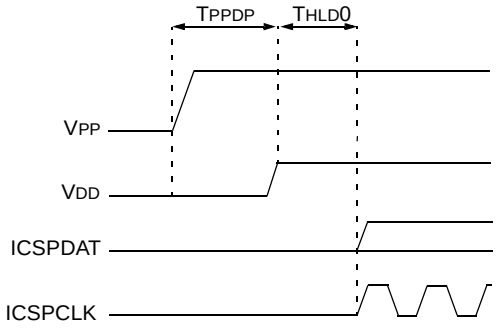
The second entry method, "VDD-first", is entered by applying VDD, holding ICSPDAT and ICSPCLK low, then raising MCLR pin from VIL to VIHH (high voltage), followed by data. This method can be used for any Configuration Word selection **except** when INTOSC and internal MCLR options are selected (FOSC<2:0> = 100 or 101 and MCLRE = 0). This technique is useful when programming the device when VDD is already applied, for it is not necessary to disconnect VDD to enter Program/Verify mode. See the timing diagram in Figure 3-2.

Once in this mode, the program memory, data memory and configuration memory can be accessed and programmed in serial fashion. ICSPDAT and ICSPCLK are Schmitt Trigger inputs in this mode. RA4 is tri-state regardless of fuse setting.

The sequence that enters the device into the Programming/Verify mode places all other logic into the Reset state (the MCLR pin was initially at VIL). Therefore, all I/O's are in the Reset state (high-impedance inputs) and the Program Counter (PC) is cleared.

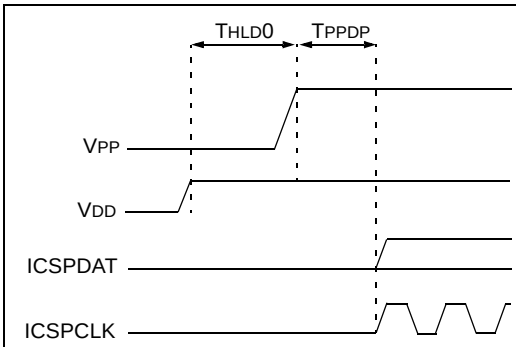
To prevent a device configured with INTOSC and internal MCLR from executing after exiting Program/Verify mode, VDD needs to power-down before VPP. See Figure 3-3 for the timing.

**FIGURE 3-1: VPP-FIRST PROGRAM/ VERIFY MODE ENTRY**



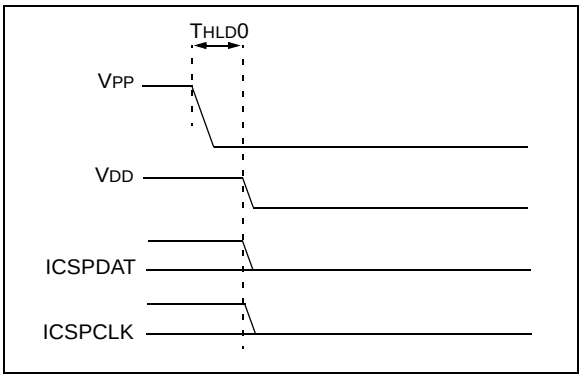

**Note:** This method of entry is valid, regardless of Configuration Word selected.

**FIGURE 3-2: VDD-FIRST PROGRAM/ VERIFY MODE ENTRY**




**Note:** This method of entry is valid if INTOSC and internal MCLR are not selected.

**FIGURE 3-3: PROGRAM/VERIFY MODE EXIT**




### 3.1 Program/Erase Algorithms

The PIC12F6XX/16F6XX program memory may be written in two ways. The fastest method writes four words at a time. However, one-word writes are also supported for backward compatibility with previous 8-pin and 14-pin Flash devices. The four-word algorithm is used to program the program memory only. The one-word algorithm can write any available memory location (i.e., program memory, configuration memory and data memory).

After writing the array, the PC may be reset and read back to verify the write. It is not possible to verify immediately following the write because the PC can only increment, not decrement.

A device Reset will clear the PC and set the address to '0'. The Increment Address command will increment the PC. The Load Configuration command will set the PC to 0x2000. The available commands are shown in Table 3-1.

### 3.1.1 FOUR-WORD PROGRAMMING

Only the program memory can be written using this algorithm. Data and configuration memory (>0x2000) must use the one-word programming algorithm (**Section 3.1.2 "One-Word Programming"**).

This algorithm writes four sequential addresses in program memory. The four addresses must point to a four-word block with addresses modulo 4 of 0, 1, 2 and 3. For example, programming address 4 through 7 can be programmed together. Programming addresses 2 through 5 will create an unexpected result.

The sequence for programming four words of program memory at a time is as follows:

1. Load a word at the current program memory address using Load Data for Program Memory command.
2. Issue a Increment Address command.
3. Load a word at the current program memory address using Load Data for Program Memory command.
4. Repeat Step 2 and Step 3 two times.
5. Issue a Begin Programming command either internally or externally timed.
6. Wait TPROG1 (internally timed) or TPROG2 (externally timed).
7. Issue a End Programming command if externally timed.
8. Issue a Increment Address command.
9. Repeat this sequence as required to write program memory.

See Figure 3-17 for more information.

### 3.1.2 ONE-WORD PROGRAMMING

The program memory may also be written one word at a time to allow compatibility with other 8-pin and 14-pin Flash PIC® devices. Configuration memory (>0x2000) and data memory must be written one word (or byte) at a time.

| | |
|---|---|
| **Note:** | The four write latches must be reset after programming the user ID (0x2000-0x2003) or Configuration Word (0x2007). See **Section 3.1.3 "Resetting Write Latches"**. |

The sequence for programming one word of program memory at a time is as follows:

1. Load a word at the current program memory address using Load Data For Program Memory command.
2. Issue a Begin Programming command either internally or externally timed.
3. Wait TPROG1 (internally timed) or TPROG2 (externally timed).
4. Issue a End Programming command if externally timed.
5. Issue a Increment Address command.
6. Repeat this sequence as required to write program, data or configuration memory.

See Figure 3-16 for more information.

### 3.1.3 RESETTING WRITE LATCHES

The user ID (0x2000-0x2003) and Configuration Word (0x2007) are mapped into the configuration memory, but do not physically reside in it. As a result, the write latches are not reset when programming these locations and must be reset by the programmer. This can be done in two ways, either loading all four latches with '1's or by exiting Program/Verify mode.

The sequence for manually resetting the write latches is as follows:

1. Load a word using Load Data for Program Memory or Load Data for Configuration Memory command with a data word of all '1's.
2. Issue a Increment Address command.
3. Repeat this sequence three times to reset all four write latches.

### 3.1.4 ERASE ALGORITHMS

The PIC12F6XX/16F6XX will erase different memory locations depending on the Program Counter (PC), $\overline{CP}$ and $\overline{CPD}$ values and which erase command executed. The following sequences can be used to erase noted memory locations. In each sequence, the data memory will be erased if the $\overline{CPD}$ bit in the Configuration Word is programmed (clear).

To erase the program memory and Configuration Word (0x2007), the following sequence must be performed. Note the Calibration Words (0x2008-0x2009) and user ID (0x2000-0x2003) **will not** be erased.

1. Do a Bulk Erase Program Memory command.
2. Wait TERA to complete erase.

To erase the user ID (0x2000-0x2003), Configuration Word (0x2007) and program memory, use the following sequence. Note that the Calibration Words (0x2008-0x2009) **will not** be erased.

1. Perform Load Configuration with dummy data to point the Program Counter (PC) to 0x2000.
2. Perform a Bulk Erase Program Memory command.
3. Wait TERA to complete erase.

To erase the data memory, use the following sequence:

1. Perform a Bulk Erase Data Memory command.
2. Wait TERA to complete erase.

### 3.1.5 SERIAL PROGRAM/VERIFY OPERATION

The ICSPCLK pin is used as a clock input and the ICSPDAT pin is used for entering command bits and data input/output during serial operation. To input a command, ICSPCLK is cycled six times. Each command bit is latched on the falling edge of the clock with the LSb of the command being input first. The data input onto the ICSPDAT pin is required to have a minimum setup and hold time (see Table 6-1), with respect to the falling edge of the clock. Commands that have data associated with them (Read and Load) are specified to have a minimum delay of 1 μs between the command and the data. After this delay, the clock pin is cycled 16 times with the first cycle being a Start bit and the last cycle being a Stop bit.

During a read operation, the LSb will be transmitted onto ICSPDAT pin on the rising edge of the second cycle. For a load operation, the LSb will be latched on the falling edge of the second cycle. A minimum 1 μs delay is also specified between consecutive commands, except for the End Programming command, which requires a 100 μs TDIS.

All commands and data words are transmitted LSb first. Data is transmitted on the rising edge and latched on the falling edge of the ICSPCLK. To allow for decoding of commands and reversal of data pin configuration, a time separation of at least 1 μs is required between a command and a data word.

The commands that are available are described in Table 3-1.

**TABLE 3-1: COMMAND MAPPING FOR PIC12F6XX/16F6XX**

| Command | Mapping (MSb … LSb) | | | | | | Data |
|---|---|---|---|---|---|---|---|
| Load Configuration | x | x | 0 | 0 | 0 | 0 | 0, data (14), 0 |
| Load Data for Program Memory | x | x | 0 | 0 | 1 | 0 | 0, data (14), 0 |
| Load Data for Data Memory | x | x | 0 | 0 | 1 | 1 | 0, data (8), zero (6), 0 |
| Read Data from Program Memory | x | x | 0 | 1 | 0 | 0 | 0, data (14), 0 |
| Read Data from Data Memory | x | x | 0 | 1 | 0 | 1 | 0, data (8), zero (6), 0 |
| Increment Address | x | x | 0 | 1 | 1 | 0 | |
| Begin Programming | x | 0 | 1 | 0 | 0 | 0 | Internally Timed |
| Begin Programming | x | 1 | 1 | 0 | 0 | 0 | Externally Timed |
| End Programming | x | 0 | 1 | 0 | 1 | 0 | |
| Bulk Erase Program Memory | x | x | 1 | 0 | 0 | 1 | Internally Timed |
| Bulk Erase Data Memory | x | x | 1 | 0 | 1 | 1 | Internally Timed |
| Row Erase Program Memory | x | 1 | 0 | 0 | 0 | 1 | Internally Timed |

### 3.1.5.1 Load Configuration

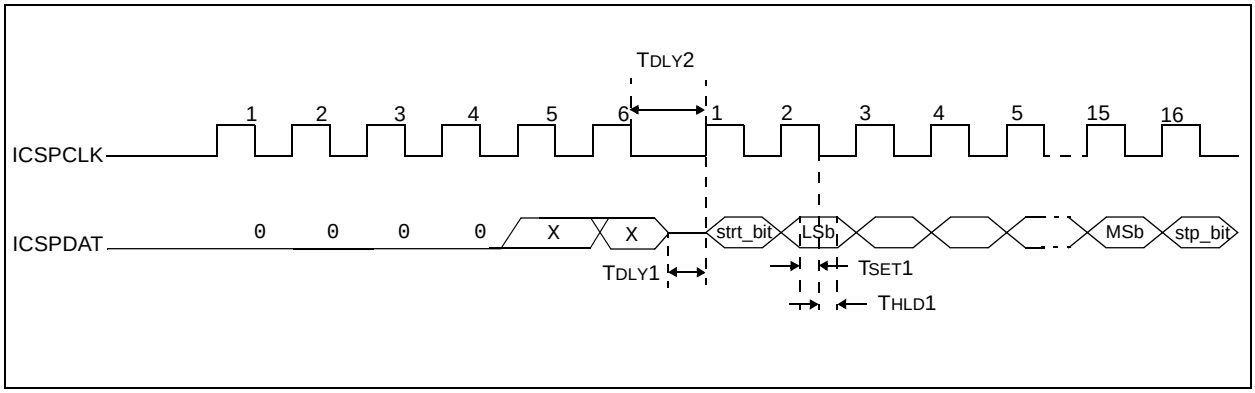
The Load Configuration command is used to access the Configuration Word (0x2007) and user ID (0x2000-0x2003). This command sets the Program Counter (PC) to address 0x2000 and loads the data latches with one word of data.

To access the configuration memory, send the Load Configuration command. Individual words within the configuration memory can be accessed by sending Increment Address commands and issuing load or read data for program memory.

After the 6-bit command is input, the ICSPCLK pin is cycled an additional 16 times for the Start bit, 14 bits of data and a Start bit (see Figure 3-4).

After the configuration memory is entered, the only way to get back to the program memory is to exit the Program/Verify mode by taking MCLR low ($V_{IL}$).

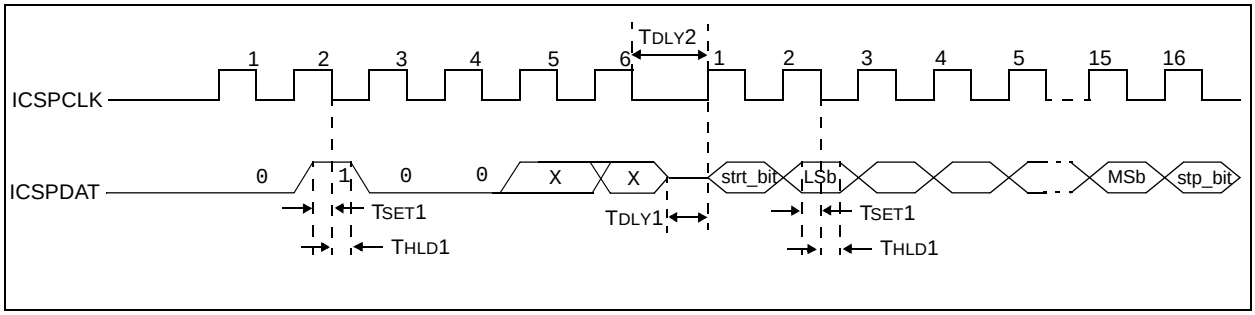
**FIGURE 3-4: LOAD CONFIGURATION COMMAND**




### 3.1.5.2 Load Data for Program Memory

After receiving this command, the chip will load in a 14-bit “data word” when 16 cycles are applied, as described previously. A timing diagram for the Load Data For Program Memory command is shown in Figure 3-5.

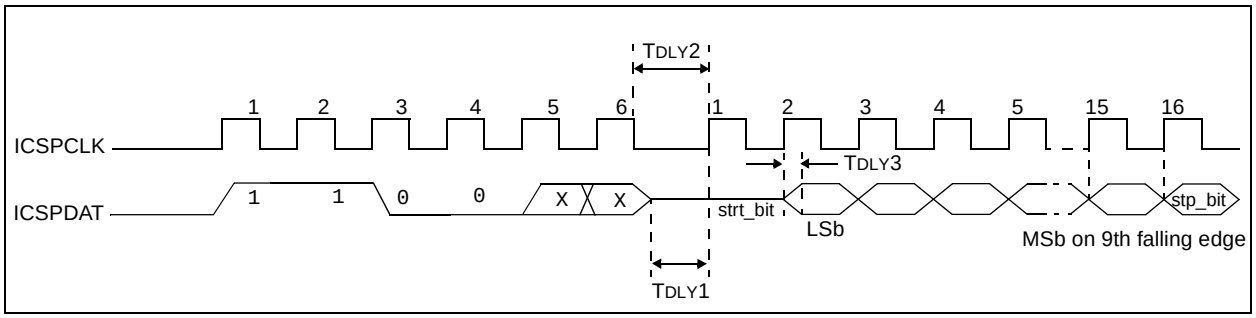
**FIGURE 3-5: LOAD DATA FOR PROGRAM MEMORY COMMAND**



### 3.1.5.3 Load Data for Data Memory

After receiving this command, the chip will load in a 14-bit "data word" when 16 cycles are applied. However, the data memory is only 8 bits wide and thus, only the first 8 bits of data after the Start bit will be programmed into the data memory. It is still necessary to cycle the clock the full 16 cycles in order to allow the internal circuitry to reset properly. The data memory contains 256 bytes.

**FIGURE 3-6: LOAD DATA FOR DATA MEMORY COMMAND**

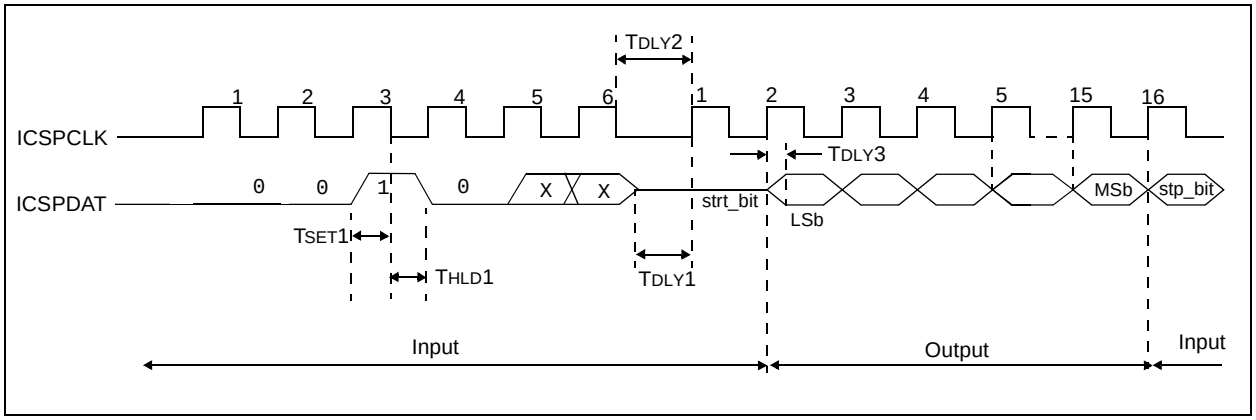



### 3.1.5.4 Read Data from Program Memory

After receiving this command, the chip will transmit data bits out of the program memory (user or configuration) currently accessed, starting with the second rising edge of the clock input. The data pin will go into Output mode on the second rising clock edge, and it will revert to Input mode (high-impedance) after the 16th rising edge.

If the program memory is code-protected ($\overline{CP}$ = 0), the data is read as zeros.

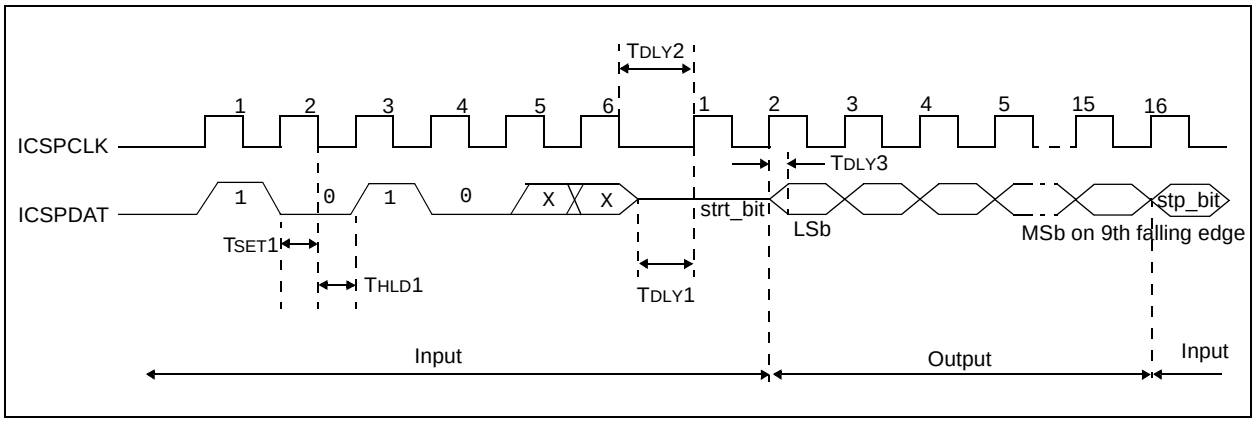
**FIGURE 3-7: READ DATA FROM PROGRAM MEMORY COMMAND**



### 3.1.5.5 Read Data from Data Memory

After receiving this command, the chip will transmit data bits out of the data memory, starting with the second rising edge of the clock input. The ICSPDAT pin will go into Output mode on the second rising edge and it will revert to Input mode (high-impedance) after the 16th rising edge. As previously stated, the data memory is 8 bits wide and, therefore, only the first 8 bits that are output are actual data. If the data memory is code-protected, the data is read as all zeros. A timing diagram of this command is shown in Figure 3-8.

**FIGURE 3-8: READ DATA FROM DATA MEMORY COMMAND**

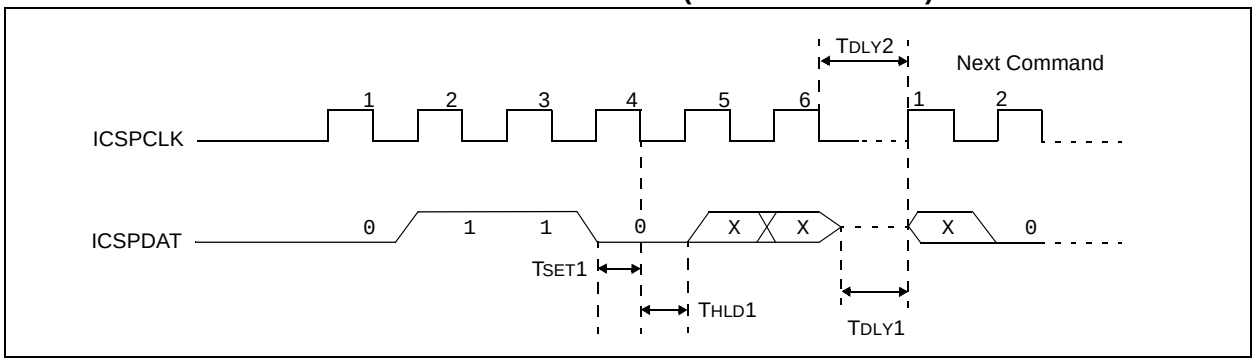



### 3.1.5.6 Increment Address

The PC is incremented when this command is received. A timing diagram of this command is shown in Figure 3-9.

It is not possible to decrement the address counter. To reset this counter, the user should exit and re-enter Program/Verify mode.

**FIGURE 3-9: INCREMENT ADDRESS COMMAND (PROGRAM/VERIFY)**

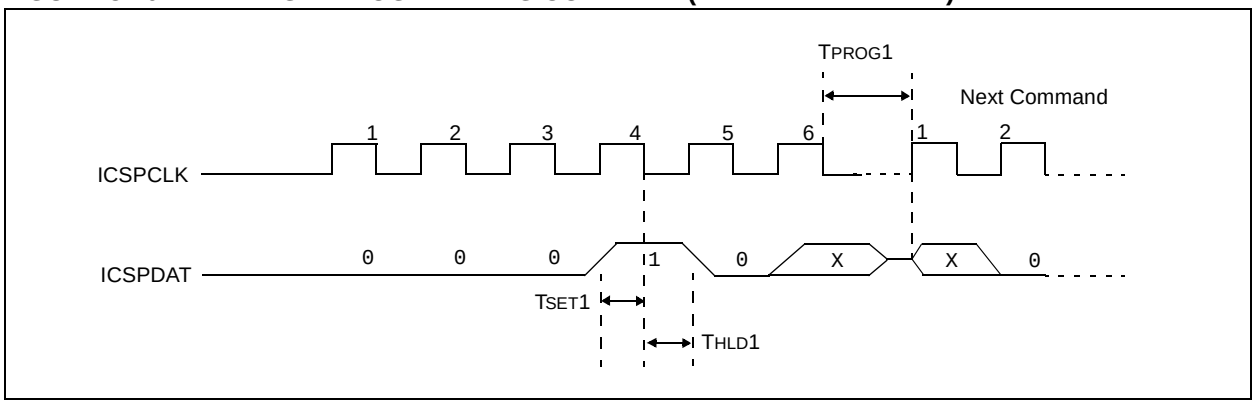


### 3.1.5.7 Begin Programming (Internally Timed)

A Load command must be given before every Begin Programming command. Programming of the appropriate memory (user program memory, configuration memory or data memory) will begin after this command is received and decoded. An internal timing mechanism executes a write. The user must allow for program cycle time for programming to complete. No End Programming command is required.

The addressed location is not erased before programming.

**FIGURE 3-10: BEGIN PROGRAMMING COMMAND (INTERNALLY TIMED)**

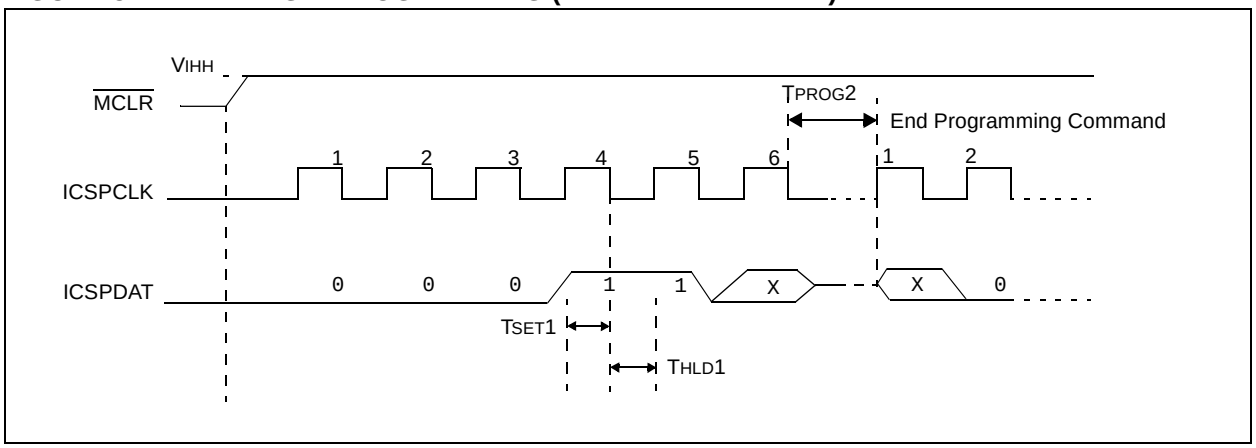



### 3.1.5.8 Begin Programming (Externally Timed)

A Load command must be given before every Begin Programming command**.** Programming of the appropriate memory (program memory, configuration or data memory) will begin after this command is received and decoded. Programming requires (TPROG2) time and is terminated using an End Programming command.

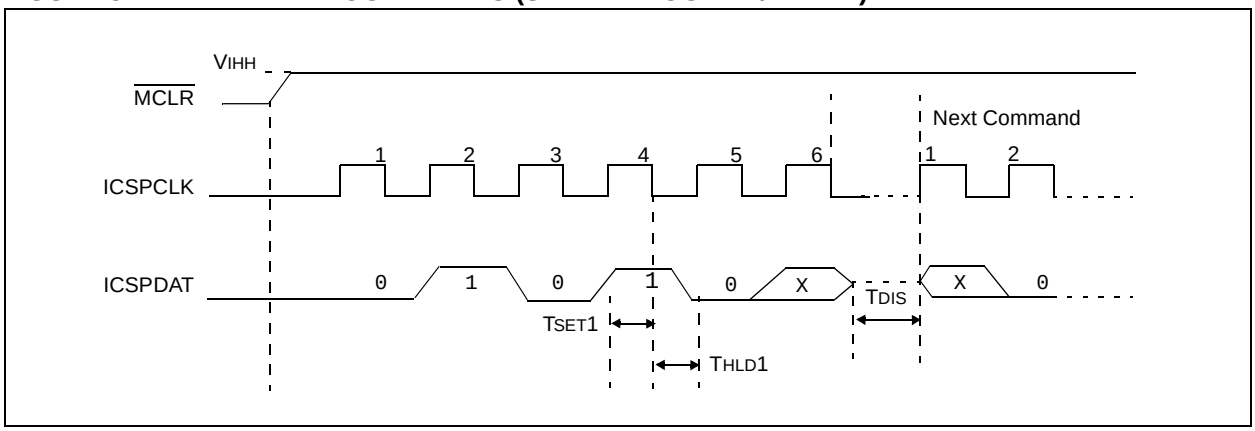
The addressed location is not erased before programming.

**FIGURE 3-11: BEGIN PROGRAMMING (EXTERNALLY TIMED)**



3.1.5.9 End Programming

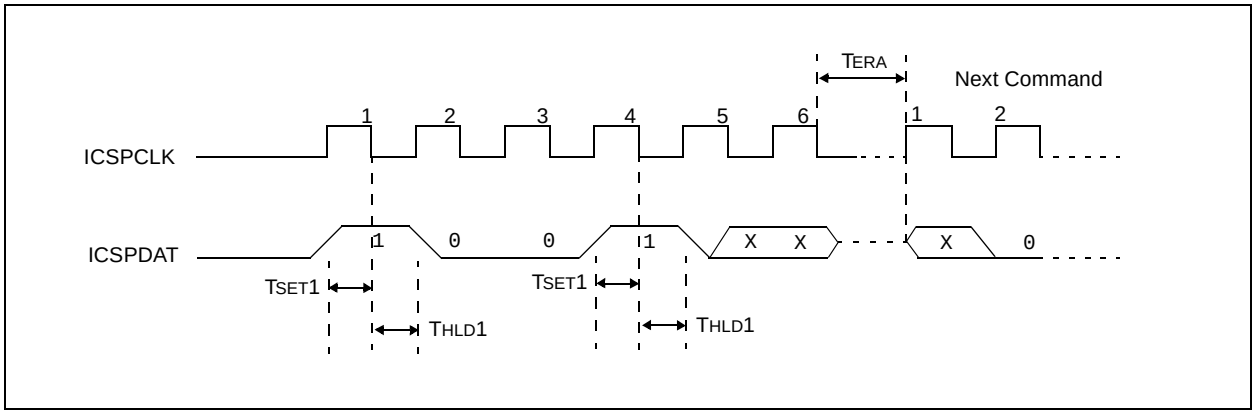
**FIGURE 3-12: END PROGRAMMING (SERIAL PROGRAM/VERIFY)**




3.1.5.10 Bulk Erase Program Memory

After this command is performed, the entire program memory and Configuration Word (0x2007) is erased. Data memory will also be erased if the $\overline{CPD}$ bit in the Configuration Word is programmed (clear). See **Section 3.1.4 "Erase Algorithms"** for erase sequences.

**FIGURE 3-13: BULK ERASE PROGRAM MEMORY COMMAND**



### 3.1.5.11 Bulk Erase Data Memory

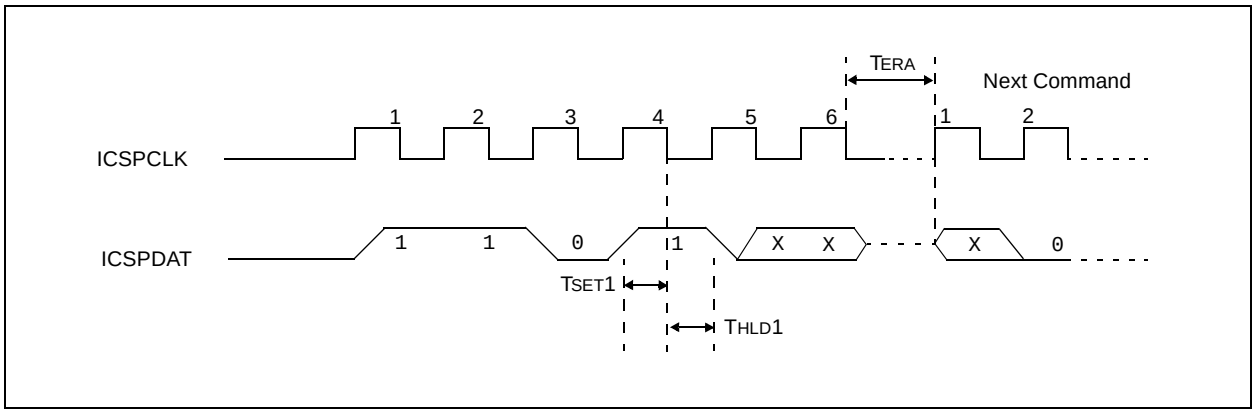
To perform an erase of the data memory, the following sequence must be performed.

1. Perform a Bulk Erase Data Memory command.
2. Wait TERA to complete Bulk Erase.

Data memory won't erase if code-protected ($\overline{CPD}$ = 0).

| **Note:** | All Bulk Erase operations must take place between 4.5V and 5.5V VDD for PIC12F6XX/16F6XX and 2.0V to 5.5V VDD for PIC12F6XX/16F6XX-ICD. |
|---|---|

**FIGURE 3-14: BULK ERASE DATA MEMORY COMMAND**




### 3.1.5.12 Row Erase Program Memory

This command erases the 16-word row of program memory pointed to by PC<11:4>. If the program memory array is protected ($\overline{CP}$ = 0) or the PC points to configuration memory (>0x2000), the command is ignored.

To perform a Row Erase Program Memory, the following sequence must be performed.

1. Execute a Row Erase Program Memory command.
2. Wait TERA to complete a row erase.

**FIGURE 3-15: ROW ERASE PROGRAM MEMORY COMMAND**

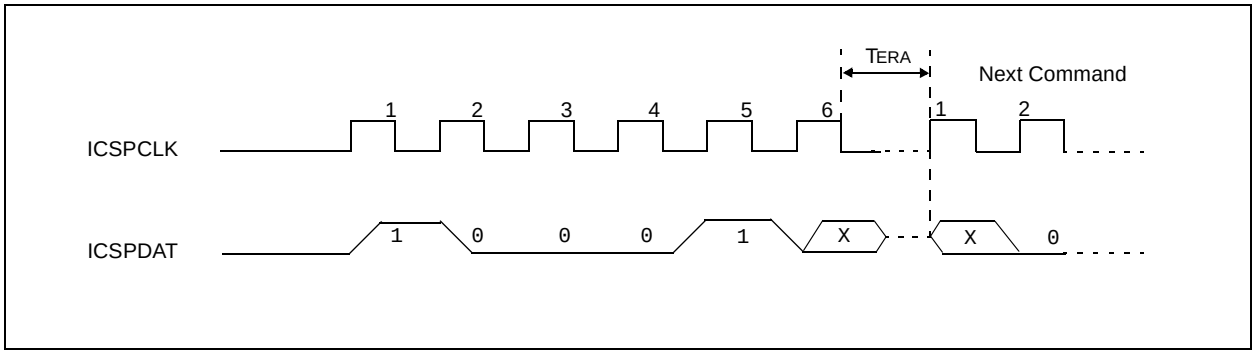
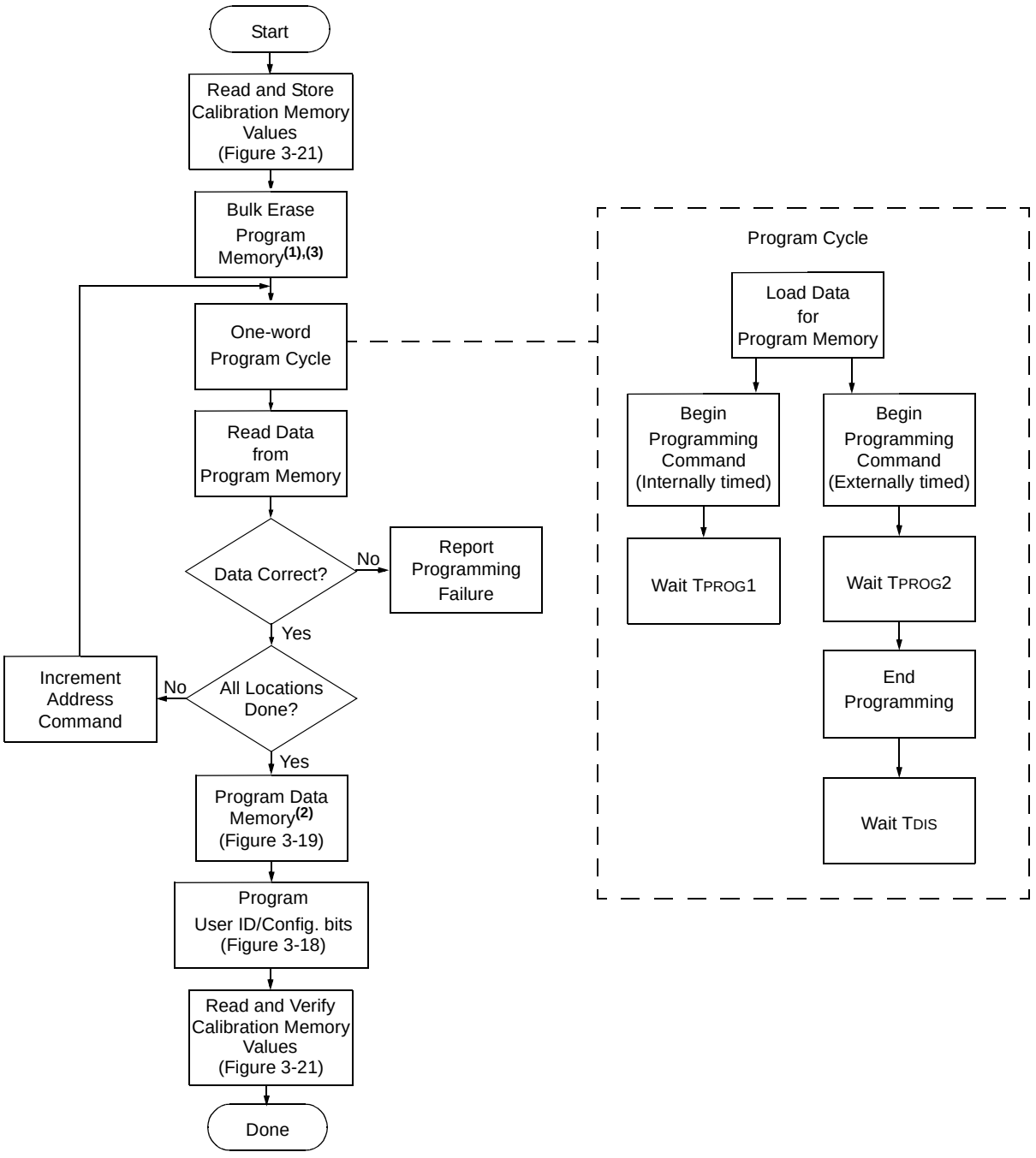


**FIGURE 3-16: ONE-WORD PROGRAMMING FLOWCHART**

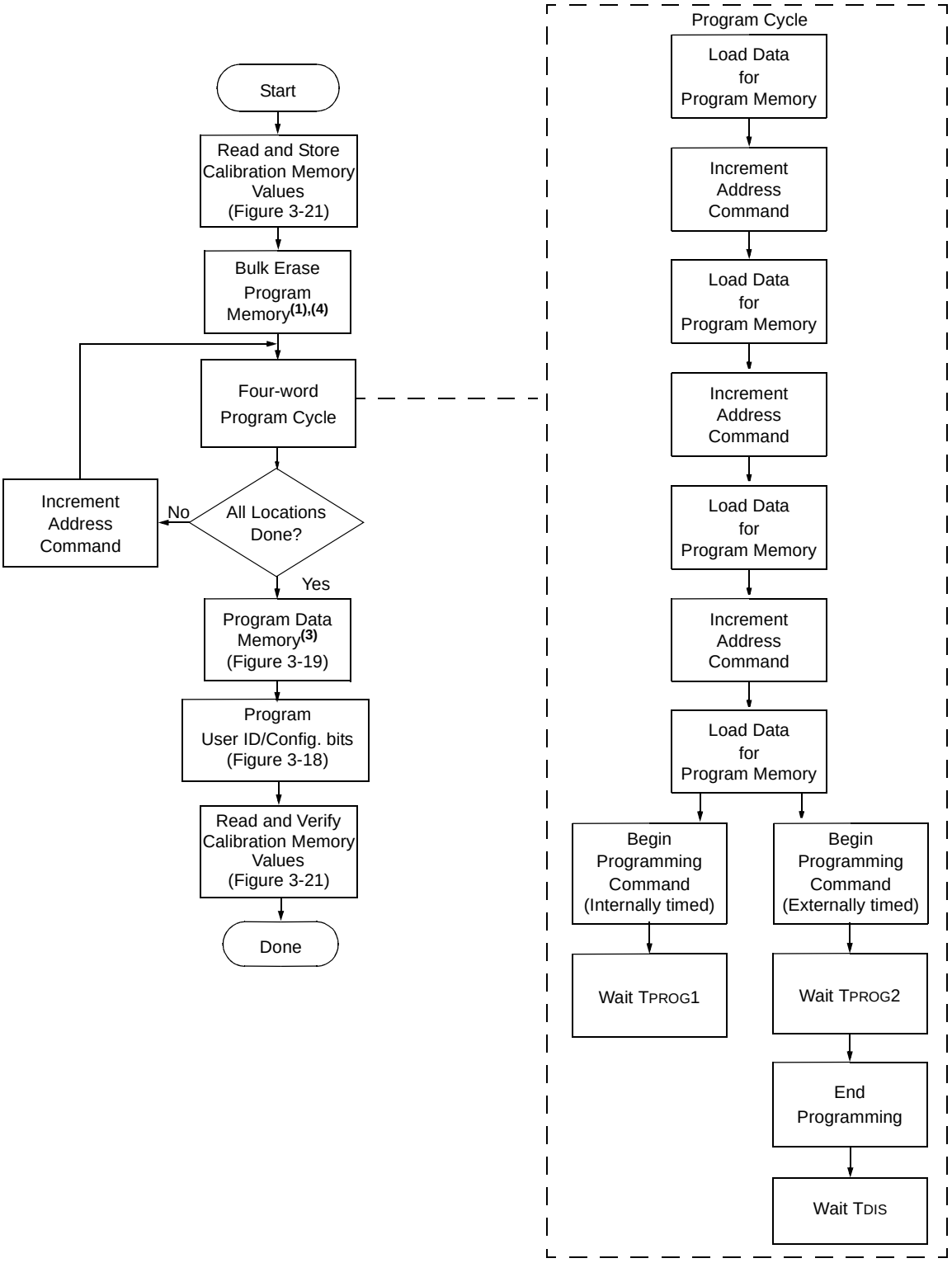



**Note 1:** This step is optional if the device has already been erased or has not been previously programmed.

**2:** This step is optional if the data memory does not require updates.

**3:** If the device is code-protected or must be completely erased, then Bulk Erase the device per Figure 3-20.

**FIGURE 3-17: FOUR-WORD PROGRAMMING FLOWCHART**



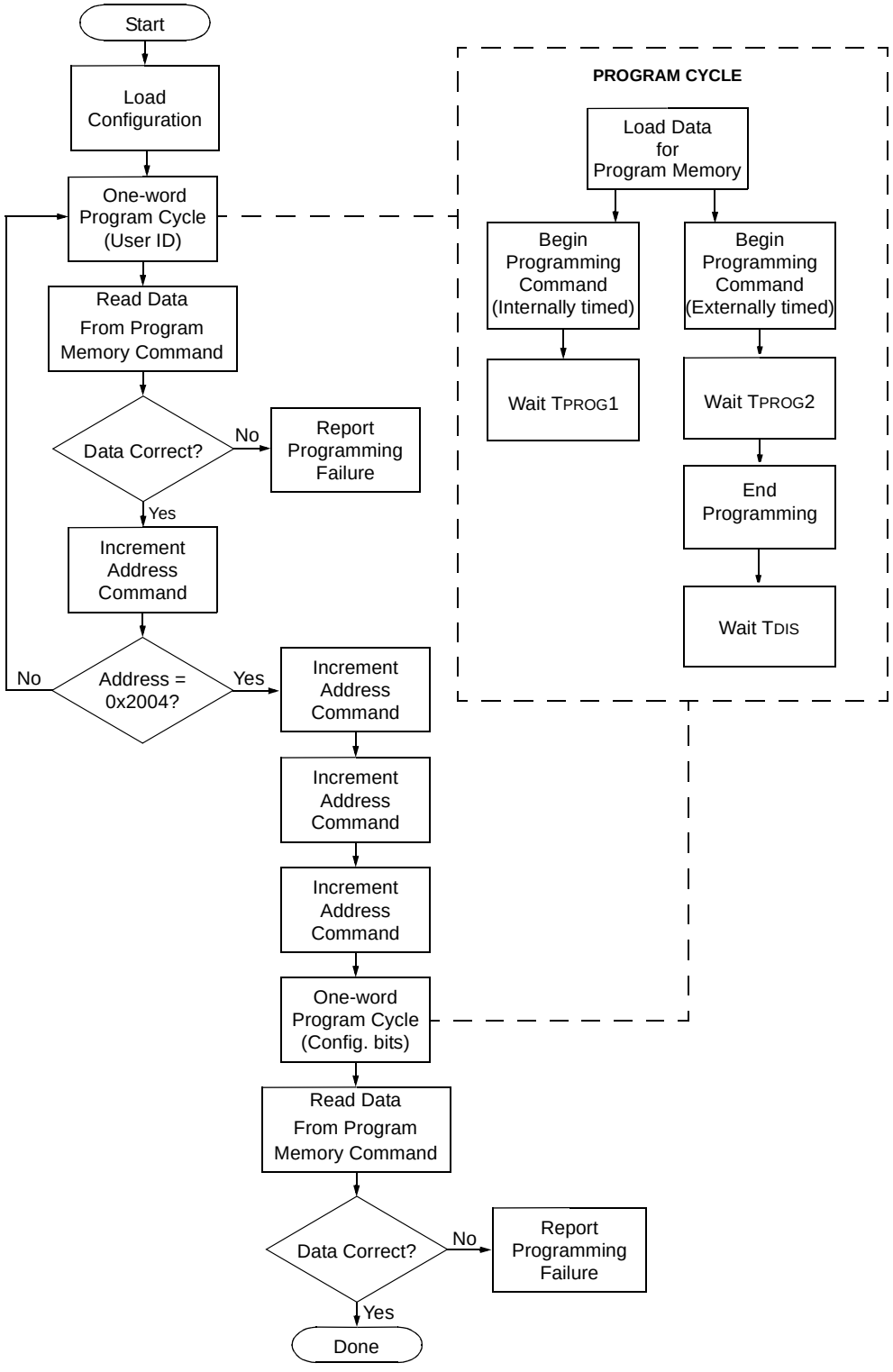

**Note 1:** This step is optional if the device is erased or not previously programmed.

**2:** Verification in Four-word mode is accomplished after programming by reading back the entire memory.

**3:** This step is optional if the data memory does not require updates.

**4:** If the device is code-protected or must be completely erased, then Bulk Erase the device per Figure 3-20.

**FIGURE 3-18: PROGRAM FLOWCHART – PIC12F6XX/16F6XX CONFIGURATION MEMORY**




**Note:** Ensure that a device Bulk Erase has been performed or that the device is blank prior to programming the configuration memory.

**FIGURE 3-19: PROGRAM FLOWCHART – PIC12F6XX/16F6XX DATA MEMORY**

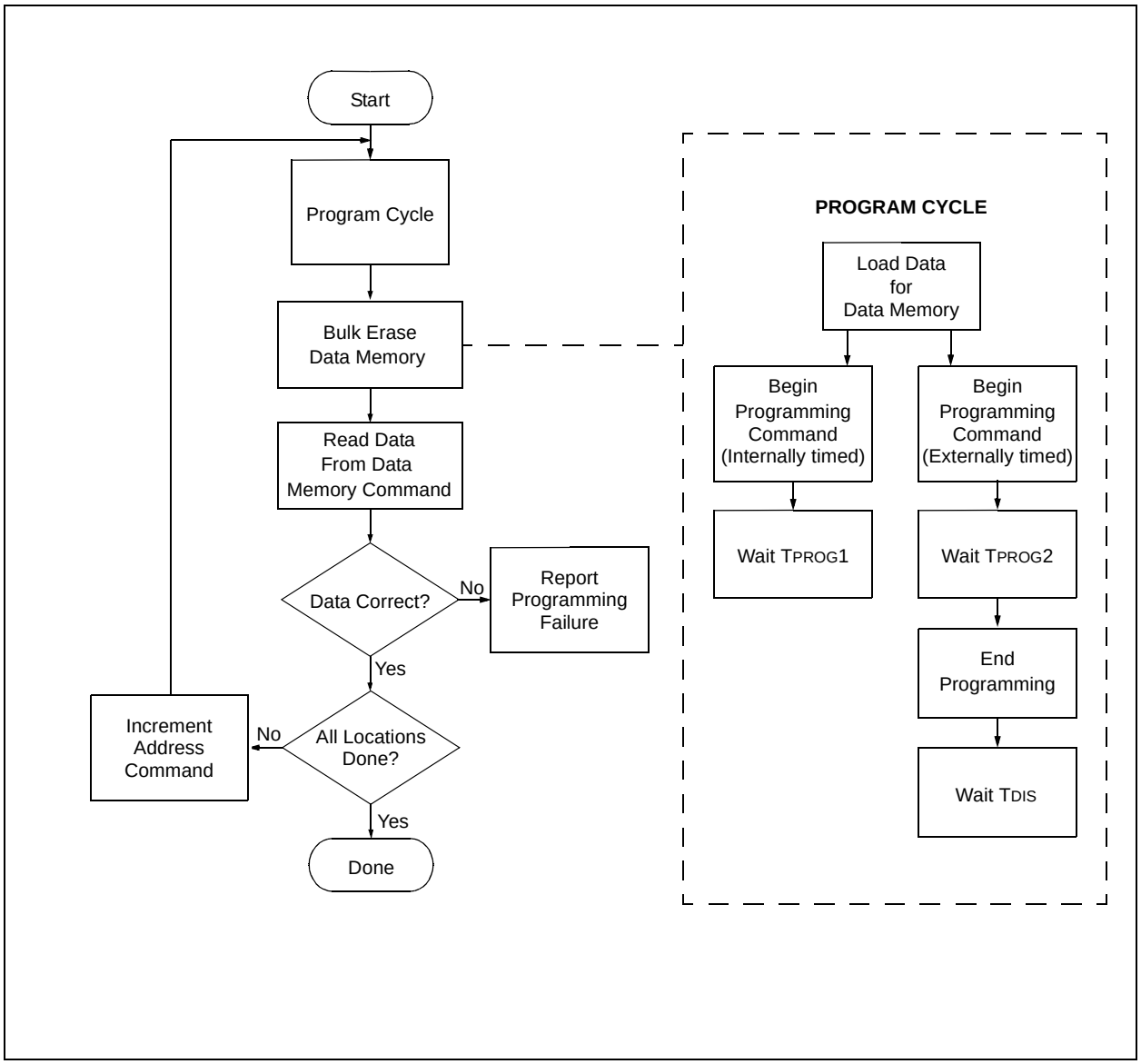


**FIGURE 3-20: PROGRAM FLOWCHART – ERASE FLASH DEVICE(1)**

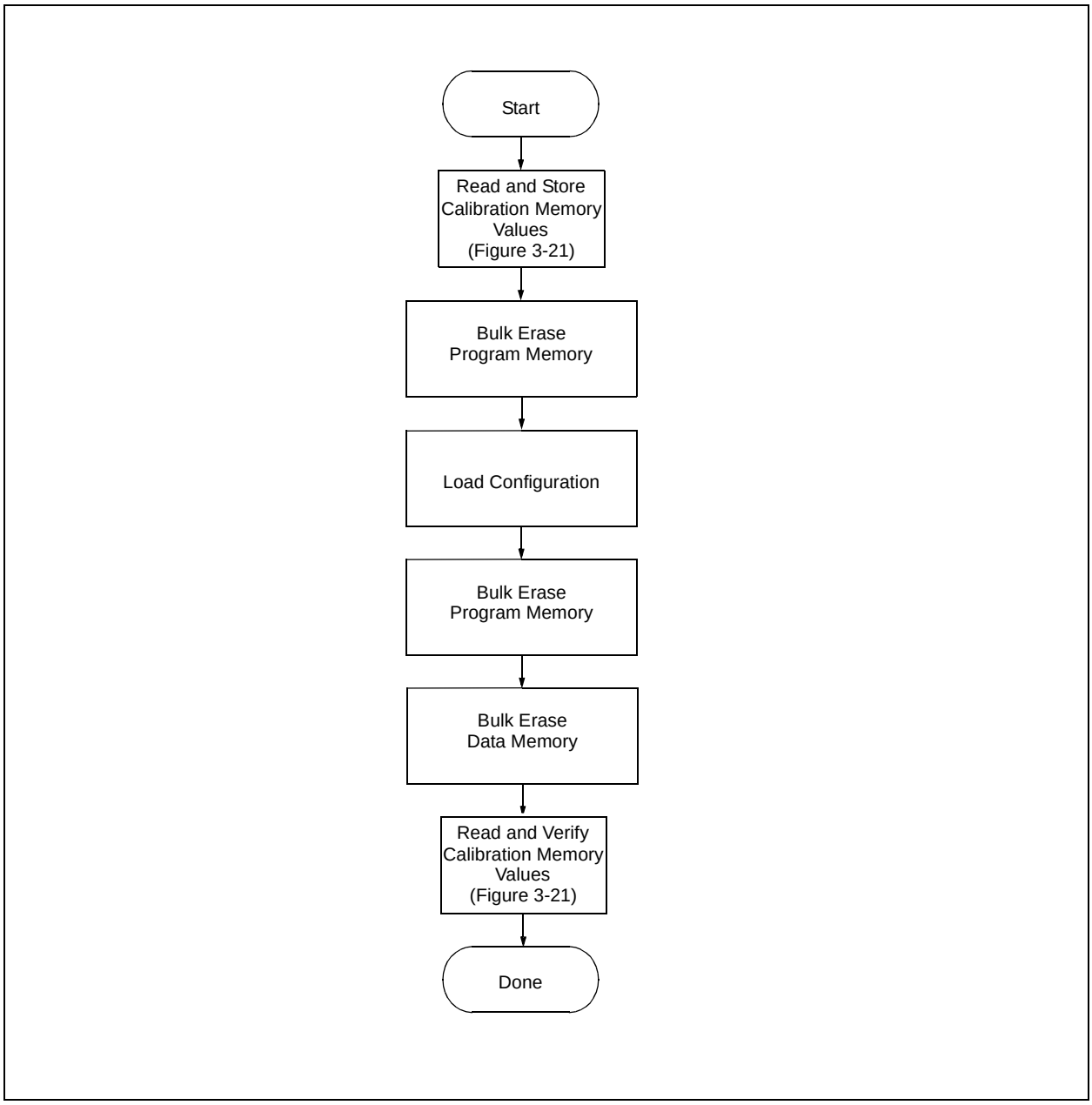
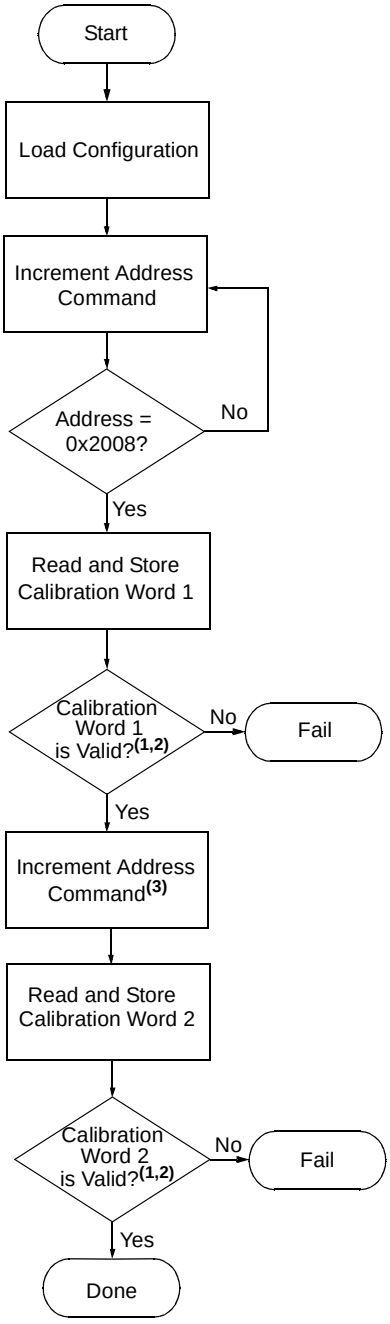


**FIGURE 3-21: CALIBRATION WORD VERIFICATION FLOWCHART**




**Note 1:** This step is not required for the Read and Store Calibration Memory Values procedure.

**2:** The device should not be used if verification of the Calibration Word locations fails. This information should be reported to the user through the user interface of the device programmer.

**3:** Several devices within this family do not possess Calibration Word 2. The remainder of this procedure is unnecessary for those devices without Calibration Word 2.

## 4.0 CONFIGURATION WORD

The PIC12F6XX/16F6XX has several Configuration bits. These bits can be programmed (reads '0') or left unchanged (reads '1'), to select various device configurations.

**REGISTER 4-1: CONFIG(1): CONFIGURATION WORD (ADDRESS:2007h) – PIC12F635/PIC16F636/PIC16F639**

| U-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|
| — | $\overline{\text{WURE}}$ | FCMEN | IESO | BOREN1 | BOREN0 | $\overline{\text{CPD}}$ |
| bit 13 | | | | | | bit7 |

| R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|
| $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE | FOSC2 | F0SC1 | F0SC0 |
| bit 6 | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 13 **Unimplemented:** Read as '1'

bit 12 **$\overline{\text{WURE}}$:** Wake-up Reset Enable bit
1 = Standard wake-up and continue enabled
0 = Wake-up and Reset enabled

bit 11 **FCMEN:** Fail-Safe Clock Monitor Enable bit
1 = Fail-Safe Clock Monitor enabled
0 = Fail-Safe Clock Monitor disabled

bit 10 **IESO:** Internal-External Switch Over bit
1 = Internal External Switchover mode enabled
0 = Internal External Switchover mode disabled

bit 8-9 **BOREN<1:0>:** Brown-out Reset Enable bits
11 = BOR enabled and SBOREN bit disabled
10 = BOR enabled while running and disabled in Sleep. SBOREN bit disabled.
01 = SBOREN in the PCON register controls BOR function
00 = BOR and SBOREN disabled

bit 7 **$\overline{\text{CPD}}$**: Data Code Protection bit(2)
1 = Data memory is not protected
0 = Data memory is external read-protected

bit 6 **$\overline{\text{CP}}$**: Code Protection bit(3)
1 = Program memory is not code-protected
0 = Program memory is external read and write-protected

bit 5 **MCLRE**: $\overline{\text{MCLR}}$ Pin Function Select bit(5)
1 = $\overline{\text{MCLR}}$ pin is $\overline{\text{MCLR}}$ function and weak internal pull-up is enabled
0 = $\overline{\text{MCLR}}$ pin is alternate function, $\overline{\text{MCLR}}$ function is internally disabled

bit 4 **$\overline{\text{PWRTE}}$**: Power-up Timer Enable bit(4)
1 = PWRT disabled
0 = PWRT enabled

bit 3 **WDTE**: Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled and can be enabled using SWDTEN in the WDTCON register

bit 2-0 **FOSC<2:0>**: Oscillator Selection bits
000 = LP oscillator: Low-power crystal on RA5(GP5)/OSC1/CLKIN and RA4(GP4)/OSC2/CLKOUT
001 = XT oscillator: Crystal/resonator on RA5(GP5)/OSC1/CLKIN and RA4(GP4)/OSC2/CLKOUT
010 = HS oscillator: High-speed crystal/resonator on RA5(GP5)/OSC1/CLKIN and RA4(GP4)/OSC2/CLKOUT
011 = EC: I/O function on RA4(GP4)/OSC2/CLKOUT, CLKIN on RA5(GP5)/OSC1/CLKIN
100 = INTOSCIO oscillator: I/O function on RA4(GP4)/OSC2/CLKOUT, I/O function on RA5(GP5)/OSC1/CLKIN
101 = INTOSC oscillator: CLKOUT function on RA4(GP4)/OSC2/CLKOUT, I/O function on RA5(GP5)/OSC1/CLKIN
110 = EXTRCIO oscillator: I/O function on RA4(GP4)/OSC2/CLKOUT, RC on RA5(GP5)OSC1/CLKIN
111 = EXTRC oscillator: CLKOUT function on RA4(GP4)/OSC2/CLKOUT, RC on RA5(GP5)/OSC1/CLKIN

**Note 1:** This Configuration Word register applies to PIC12F635/PIC16F636/PIC16F639 devices only.
**2:** The entire data memory will be erased when the code protection is turned off.
**3:** The entire program memory will be erased when the code protection is turned off.
**4:** Enabling Brown-out Detect does not automatically enable Power-up Timer.
**5:** When $\overline{\text{MCLR}}$ is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.

**REGISTER 4-2: CONFIG(1): CONFIGURATION WORD (ADDRESS:2007h) – PIC12F683, PIC16F631/677/684/685/687/688/689/690**

| U-1 | U-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|
| — | — | FCMEN | IESO | BOREN1 | BOREN0 | $\overline{CPD}$ |
| bit 13 | | | | | | bit7 |

| R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|
| $\overline{CP}$ | MCLRE | $\overline{PWRTE}$ | WDTE | FOSC2 | F0SC1 | F0SC0 |
| bit 6 | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 13-12 **Unimplemented:** Read as '1'

bit 11 **FCMEN:** Fail-Safe Clock Monitor Enable bit
1 = Fail-Safe Clock Monitor enabled
0 = Fail-Safe Clock Monitor disabled

bit 10 **IESO:** Internal-External Switch Over bit
1 = Internal External Switchover mode enabled
0 = Internal External Switchover mode disabled

bit 9-8 **BOREN<1:0>:** Brown-out Reset Enable bits(4)
11 = BOR enabled and SBOREN bit disabled
10 = BOR enabled while running and disabled in Sleep. SBOREN bit disabled.
01 = SBOREN in the PCON register controls BOR function
00 = BOR and SBOREN disabled

bit 7 **$\overline{CPD}$**: Code Protection Data bit(2)
1 = Data memory is not protected
0 = Data memory is external read-protected

bit 6 **$\overline{CP}$**: Code Protection bit(3)
1 = Program memory is not code-protected
0 = Program memory is external read and write-protected

bit 5 **MCLRE**: $\overline{MCLR}$ Pin Function Select(5) bit
1 = $\overline{MCLR}$ pin is $\overline{MCLR}$ function and weak internal pull-up is enabled
0 = $\overline{MCLR}$ pin is alternate function, $\overline{MCLR}$ function is internally disabled

bit 4 **$\overline{PWRTE}$**: Power-up Timer Enable bit(4)
1 = PWRT disabled
0 = PWRT enabled

bit 3 **WDTE**: Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled and can be enabled using SWDTEN in the WDTCON register

bit 2-0 **FOSC<2:0>**: Oscillator Selection bits
000 = LP oscillator: Low-power crystal on RA5(GP5)/OSC1/CLKIN and RA4(GP4)/OSC2/CLKOUT
001 = XT oscillator: Crystal/resonator on RA5(GP5)/OSC1/CLKIN and RA4(GP4)/OSC2/CLKOUT
010 = HS oscillator: High-speed crystal/resonator on RA5(GP5)/OSC1/CLKIN and RA4(GP4)/OSC2/CLKOUT
011 = EC: I/O function on RA4(GP4)/OSC2/CLKOUT, CLKIN on RA5(GP5)/OSC1/CLKIN
100 = INTOSCIO oscillator: I/O function on RA4(GP4)/OSC2/CLKOUT, I/O function on RA5(GP5)/OSC1/CLKIN
101 = INTOSC oscillator: CLKOUT function on RA4(GP4)/OSC2/CLKOUT, I/O function on RA5(GP5)/OSC1/CLKIN
110 = EXTRCIO oscillator: I/O function on RA4(GP4)/OSC2/CLKOUT, RC on RA5(GP5)OSC1/CLKIN
111 = EXTRC oscillator: CLKOUT function on RA4(GP4)/OSC2/CLKOUT, RC on RA5(GP5)/OSC1/CLKIN

**Note 1:** This Configuration Word register applies to PIC16F631/677/685/687/689/690 (not including PIC12F635/PIC16F636/PIC16F639) only.
**2:** The entire data memory will be erased when the code protection is turned off.
**3:** The entire program memory will be erased when the code protection is turned off.
**4:** Enabling Brown-out Detect does not automatically enable Power-up Timer.
**5:** When $\overline{MCLR}$ is asserted in INTOSC or RC mode, the internal clock oscillator is disabled.
**6:** For PIC16F685/PIC16F687/PIC16F689/PIC16F690, the pin is RA4/AN3/T1G/OSC2/CLKOUT.

**REGISTER 4-3: CALIB[1]: CALIBRATION WORD (ADDRESS: 2008h) – PIC12F683/684/688[2], [3]**

| U-1 | P/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|
| — | FCAL6 | FCAL5 | FCAL4 | FCAL3 | FCAL2 | FCAL1 |
| bit 13 | | | | | | bit7 |

| R/P-1 | U-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|
| FCAL0 | — | POR1 | POR0 | BOR2 | BOR1 | BOR0 |
| bit 6 | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 13 **Unimplemented:** Read as '0'

bit 12-6 **FCAL<6:0>:** Internal Oscillator Calibration bits
0111111 = Maximum frequency
•
•
0000001
0000000 = Center frequency
1111111
•
•
1000000 = Minimum frequency

bit 5 **Unimplemented:** Read as '0'

bit 4-3 **POR<1:0>:** POR Calibration bits
00 = Lowest POR voltage
11 = Highest POR voltage

bit 2-0 **BOR<2:0>:** BOR Calibration bits
000 = Reserved
001 = Lowest BOR voltage
111 = Highest BOR voltage

**Note 1:** This Calibration Word register applies to PIC12F683/PIC16F684/PIC16F688 devices only.
**2:** This location does not participate in Bulk Erase operations if the procedure in Figure 3-20 is used.
**3:** Calibration bits are reserved for factory calibration. These values can and will change across the entire range, therefore, specific values and available adjustment range cannot be specified.

**REGISTER 4-4: CALIB[1]: CALIBRATION WORD (ADDRESS: 2008h)– PIC16F631/677/685/687/689/690[2], [3], [4]**

| U-1 | P/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|
| — | FCAL6 | FCAL5 | FCAL4 | FCAL3 | FCAL2 | FCAL1 |
| bit 13 | | | | | | bit7 |

| R/P-1 | P/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|
| FCAL0 | POR2 | POR1 | POR0 | BOR2 | BOR1 | BOR0 |
| bit 6 | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 13 **Unimplemented:** Read as '0'

bit 12-6 **FCAL<6:0>:** Internal Oscillator Calibration bits
0111111 = Maximum frequency
•
•
0000001
0000000 = Center frequency
1111111
•
•
1000000 = Minimum frequency

bit 5-3 **POR<2:0>:** POR Calibration bits
111 = Maximum POR voltage
110
101
100 = Center POR voltage
000 = Center POR voltage
001
010
011 = Minimum POR voltage

bit 2-0 **BOR<2:0>:** BOR Calibration bits
111 = Maximum BOR voltage
110
101
100 = Center BOR voltage
000 = Center BOR voltage
001
010
011 = Minimum BOR voltage

**Note 1:** This Calibration Word register applies to PIC16F631/677/685/687/689/690 devices only.
**2:** This location does not participate in Bulk Erase operations if the procedure in Figure 3-20 is used.
**3:** Calibration bits are reserved for factory calibration. These values can and will change across the entire range, therefore, specific values and available adjustment range cannot be specified.
**4:** The calibration bits must be read, preserved, then replaced by the user during Program Memory Bulk Erase operation with PC = 2008h.

**REGISTER 4-5: CALIB1: CALIBRATION WORD 1 (ADDRESS: 2008H) – PIC12F635/636/639(1)**

| U-1 | P/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|
| — | FCAL6 | FCAL5 | FCAL4 | FCAL3 | FCAL2 | FCAL1 |
| bit 13 | | | | | | bit7 |

| R/P-1 | P/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|
| FCAL0 | POR2 | POR1 | POR0 | BOR2 | BOR1 | BOR0 |
| bit 6 | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 13 **Unimplemented:** Read as '0'

bit 12-6 **FCAL<6:0>:** Internal Oscillator Calibration bits
0111111 = Maximum frequency
•
•
0000001
0000000 = Center frequency. Oscillator is running at the calibrated frequency
1111111
•
•
1000000 = Minimum frequency

bit 5-3 **POR<2:0>:** POR Calibration bits
111 = Maximum POR voltage
110
101
100 = Center POR voltage
000 = Center POR voltage
001
010
011 = Minimum POR voltage

bit 2-0 **BOR<2:0>:** BOR Calibration bits
111 = Maximum BOR voltage
110
101
100 = Center BOR voltage
000 = Center BOR voltage
001
010
011 = Minimum BOR voltage

**Note 1:** This location does not participate in Bulk Erase operation, unless PC = 2008h.

**REGISTER 4-6: CALIB2 – CALIBRATION WORD 2 (ADDRESS: 2009h) – PIC12F635/636/639(1)**

| U-1 | U-1 | U-1 | U-1 | U-1 | U-1 | U-1 |
|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — |
| bit 13 | | | | | | bit7 |

| U-1 | P/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|
| — | WUR2 | WUR1 | WUR0 | LVD2 | LVD1 | LVD0 |
| bit 6 | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 13-6 **Unimplemented:** Read as '0'

bit 5-3 **WUR<2:0>:** WUR Calibration bits
111 = Maximum WUR voltage
110
101
100 = Center WUR voltage
000 = Center WUR voltage
001
010
011 = Minimum WUR voltage

bit 2-0 **LVD<2:0>:** LVD Calibration bits
111 = Maximum LVD voltage
110
101
100 = Center LVD voltage
000 = Center LVD voltage
001
010
011 = Minimum LVD voltage

**Note 1:** This location does not participate in Bulk Erase operation, unless PC = 2009h.

## 4.1 Device ID Word

The device ID word for the PIC12F6XX/16F6XX is located at 2006h. This location cannot be erased.

**TABLE 4-1: DEVICE ID VALUES**

| Device | Device ID Values | |
|---|---|---|
| | Dev | Rev |
| PIC12F635 | 00 1111 101 | x xxxx |
| PIC12F683 | 00 0100 011 | x xxxx |
| PIC16F631 | 01 0100 001 | x xxxx |
| PIC16F636 | 01 0000 101 | x xxxx |
| PIC16F639 | 01 0000 101 | x xxxx |
| PIC16F677 | 01 0100 010 | x xxxx |
| PIC16F684 | 01 0000 100 | x xxxx |
| PIC16F685 | 00 0100 101 | x xxxx |
| PIC16F687 | 01 0011 001 | x xxxx |
| PIC16F688 | 01 0001 100 | x xxxx |
| PIC16F689 | 01 0011 010 | x xxxx |
| PIC16F690 | 01 0100 000 | x xxxx |

## 5.0 CODE PROTECTION

For PIC12F6XX/16F6XX, once the $\overline{CP}$ bit is programmed to '0', all program memory locations read all '0's. The user ID locations and the Configuration Word read out in an unprotected fashion. Further programming is disabled for the entire program memory.

Data memory is protected with its own code-protect bit ($\overline{CPD}$). When enabled, the data memory can still be programmed and read using the EECON1 register (see the applicable data sheet for more information).

The user ID locations and the Configuration Word can be programmed regardless of the state of the $\overline{CP}$ and $\overline{CPD}$ bits.

### 5.1 Disabling Code Protection

It is recommended to use the procedure in Figure 3-20 to disable code protection of the device. This sequence will erase the program memory, data memory, Configuration Word (0x2007) and user ID locations (0x2000-0x2003). The Calibration Words (0x2008-0x2009) **will not** be erased.

| | |
|---|---|
| **Note:** | To ensure system security, if $\overline{CPD}$ bit = 0, Bulk Erase Program Memory command will also erase data memory. |

### 5.2 Embedding Configuration Word and User ID Information in the Hex File

To allow portability of code, the programmer is required to read the Configuration Word and user ID locations from the hex file when loading the hex file. If Configuration Word information was not present in the hex file, a simple warning message may be issued. Similarly, while saving a hex file, Configuration Word and user ID information must be included. An option to not include this information may be provided.

Specifically for the PIC12F6XX/16F6XX, the data memory should also be embedded in the hex file (see **Section 5.3.2 "Embedding Data Memory Contents in Hex File"**).

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

### 5.3 Checksum Computation

#### 5.3.1 CHECKSUM

Checksum is calculated by reading the contents of the PIC12F6XX/16F6XX memory locations and adding up the opcodes up to the maximum user addressable location (e.g., 0x7FF for the PIC16F684). Any Carry bits exceeding 16 bits are neglected. Finally, the Configuration Word (appropriately masked) is added to the checksum. Checksum computation for the PIC12F6XX/16F6XX devices is shown in Table 5-1.

The checksum is calculated by summing the following:

- The contents of all program memory locations
- The Configuration Word, appropriately masked
- Masked user ID locations (when applicable)

The Least Significant 16 bits of this sum is the checksum.

The following table describes how to calculate the checksum for each device. Note that the checksum calculation differs depending on the code-protect setting. Since the program memory locations read out zeroes when code-protected, the table describes how to manipulate the actual program memory values to simulate values that would be read from a protected device. When calculating a checksum by reading a device, the entire program memory can simply be read and summed. The Configuration Word and user ID locations can always be read regardless of the code-protect setting.

| | |
|---|---|
| **Note:** | Some older devices have an additional value added in the checksum. This is to maintain compatibility with older device programmer checksums. |

**TABLE 5-1: CHECKSUM COMPUTATIONS**

| Device | Code Protect | Checksum* | Blank Value | 0x25E6 at 0 and Max. Address |
|---|---|---|---|---|
| PIC12F635 | $\overline{CP}$ = 1, $\overline{CPD}$ = 1 | SUM[0x000:0x03FF] + (CFGW & 1FFF) | 0x1BFF | 0xE7CD |
| | $\overline{CP}$ = 0, $\overline{CPD}$ = 1 | (CFGW & 1FFF) + SUM_ID | 0x3BBE | 0x078C |
| PIC12F683 | $\overline{CP}$ = 1, $\overline{CPD}$ = 1 | SUM[0x000:0x07FF] + (CFGW & 0FFF) | 0x07FF | 0xD3CD |
| | $\overline{CP}$ = 0, $\overline{CPD}$ = 1 | (CFGW & 0x0FFF) + SUM_ID | 0x17BE | 0xE38C |
| PIC16F631 | $\overline{CP}$ = 1, $\overline{CPD}$ = 1 | SUM[0x000:0x03FF] + (CFGW & 0FFF) | 0x0BFF | 0xD7CD |
| | $\overline{CP}$ = 0, $\overline{CPD}$ = 1 | (CFGW & 0FFF) + SUM_ID | 0x1BBE | 0xE78C |
| PIC16F636 | $\overline{CP}$ = 1, $\overline{CPD}$ = 1 | SUM[0x000:0x07FF] + (CFGW & 1FFF) | 0x17FF | 0xE3CD |
| | $\overline{CP}$ = 0, $\overline{CPD}$ = 1 | (CFGW & 0x1FFF) + SUM_ID | 0X37BE | 0X038C |
| PIC16F639 | $\overline{CP}$ = 1, $\overline{CPD}$ = 1 | SUM[0x000:0x07FF] + (CFGW & 1FFF) | 0x17FF | 0xE3CD |
| | $\overline{CP}$ = 0, $\overline{CPD}$ = 1 | (CFGW & 0x1FFF) + SUM_ID | 0x37BE | 0x038C |
| PIC16F677 | $\overline{CP}$ = 1, $\overline{CPD}$ = 1 | SUM[0x000:0x07FF] + (CFGW & 0FFF) | 0x07FF | 0xD3CD |
| | $\overline{CP}$ = 0, $\overline{CPD}$ = 1 | (CFGW & 0x0FFF) + SUM_ID | 0x17BE | 0xE38C |
| PIC16F684 | $\overline{CP}$ = 1, $\overline{CPD}$ = 1 | SUM[0x000:0x07FF] + (CFGW & 0FFF) | 0x07FF | 0xD3CD |
| | $\overline{CP}$ = 0, $\overline{CPD}$ = 1 | (CFGW & 0x0FFF) + SUM_ID | 0x17BE | 0xE38C |
| PIC16F685 | $\overline{CP}$ = 1, $\overline{CPD}$ = 1 | SUM[0x000:0x0FFF] + (CFGW & 0FFF) | 0xFFFF | 0xCBCD |
| | $\overline{CP}$ = 0, $\overline{CPD}$ = 1 | (CFGW & 0x0FFF) + SUM_ID | 0x0FBE | 0xDB8C |
| PIC16F687 | $\overline{CP}$ = 1, $\overline{CPD}$ = 1 | SUM[0x000:0x07FF] + (CFGW & 0FFF) | 0x07FF | 0xD3CD |
| | $\overline{CP}$ = 0, $\overline{CPD}$ = 1 | (CFGW & 0x0FFF) + SUM_ID | 0x17BE | 0xE38C |
| PIC16F688 | $\overline{CP}$ = 1, $\overline{CPD}$ = 1 | SUM[0x000:0x0FFF] + (CFGW & 0FFF) | 0xFFFF | 0xCBCD |
| | $\overline{CP}$ = 0, $\overline{CPD}$ = 1 | (CFGW & 0x0FFF) + SUM_ID | 0x0FBE | 0xDB8C |
| PIC16F689 | $\overline{CP}$ = 1, $\overline{CPD}$ = 1 | SUM[0x000:0x0FFF] + (CFGW & 0FFF) | 0xFFFF | 0xCBCD |
| | $\overline{CP}$ = 0, $\overline{CPD}$ = 1 | (CFGW & 0x0FFF) + SUM_ID | 0x0FBE | 0xDB8C |
| PIC16F690 | $\overline{CP}$ = 1, $\overline{CPD}$ = 1 | SUM[0x000:0x0FFF] + (CFGW & 0FFF) | 0xFFFF | 0xCBCD |
| | $\overline{CP}$ = 0, $\overline{CPD}$ = 1 | (CFGW & 0x0FFF) + SUM_ID | 0x0FBE | 0xDB8C |

**Legend:** CFGW = Configuration Word. Example calculations assume Configuration Word is erased (all '1's).
SUM[a:b] = [Sum of locations a to b inclusive]
SUM_ID = User ID locations masked by 0xF then made into a 16-bit value with ID0 as the Most Significant nibble.
For example, ID0 = 0x1, ID1 = 0x2, ID3 = 0x3, ID4 = 0x4, then SUM_ID = 0x1234.
The 4 LSb's of the unprotected checksum is used for the example calculations.
*Checksum = [Sum of all the individual expressions] MODULO [0xFFFF]
+ = Addition
& = Bitwise AND

### 5.3.2 EMBEDDING DATA MEMORY CONTENTS IN HEX FILE

The programmer should be able to read data memory information from a hex file and conversely (as an option), write data memory contents to a hex file along with program memory information and Configuration Word (0x2007) and user ID (0x2000-0x2003) information.

The 256 data memory locations are logically mapped starting at address 0x2100. The format for data memory storage is one data byte per address location, LSb aligned.

## 6.0 PROGRAM/VERIFY MODE ELECTRICAL CHARACTERISTICS

**TABLE 6-1: AC/DC CHARACTERISTICS TIMING REQUIREMENTS FOR PROGRAM/VERIFY MODE**

| AC/DC CHARACTERISTICS | | Standard Operating Conditions (unless otherwise stated)<br>Operating Temperature -40°C ≤ TA ≤ +85°C<br>Operating Voltage 4.5V ≤ VDD ≤ 5.5V | | | | |
|---|---|---|---|---|---|---|
| **Sym.** | **Characteristics** | **Min.** | **Typ.** | **Max.** | **Units** | **Conditions/Comments** |
| **General** | | | | | | |
| VDD | VDD level for read/write operations, program and data memory | 2.0 | — | 5.5 | V | |
| | VDD level for Bulk Erase operations, program and data memory | 2.0<br>4.5 | —<br>— | 5.5<br>5.5 | V<br>V | PIC12F6XX/16F6XX-ICD<br>PIC12F6XX/16F6XX |
| VIHH | High voltage on $\overline{MCLR}$ for Program/Verify mode entry | 10 | — | 13 | V | |
| TVHHR | $\overline{MCLR}$ rise time (VSS to VHH) for Program/Verify mode entry | — | — | 1.0 | μs | |
| TPPDP | Hold time after VPP changes | 5 | — | — | μs | |
| VIH1 | (ICSPCLK, ICSPDAT) input high level | 0.8 VDD | — | — | V | |
| VIL1 | (ICSPCLK, ICSPDAT) input low level | 0.2 VDD | — | — | V | |
| TSET0 | ICSPCLK, ICSPDAT setup time before $\overline{MCLR}$↑ (Program/Verify mode selection pattern setup time) | 100 | — | — | ns | |
| THLD0 | Hold time after VDD changes | 0 | — | 2 | μs | |
| **Serial Program/Verify** | | | | | | |
| TSET1 | Data in setup time before clock↓ | 100 | — | — | ns | |
| THLD1 | Data in hold time after clock↓ | 100 | — | — | ns | |
| TDLY1 | Data input not driven to next clock input (delay required between command/data or command/ command) | 1.0 | — | — | μs | |
| TDLY2 | Delay between clock↓ to clock↑ of next command or data | 1.0 | — | — | μs | |
| TDLY3 | Clock↑ to data out valid (during a Read Data command) | | — | 80 | ns | |
| TERA | Erase cycle time | — | 5 | 6 | ms | |
| TPROG1 | Programming cycle time (internally timed) | 3<br>6 | —<br>— | —<br>— | ms<br>ms | Program memory<br>Data memory |
| TPROG2 | Programming cycle time (externally timed) | 3 | — | — | ms | 10°C ≤ TA ≤ +40°C<br>Program memory |
| TDIS | Time delay from program to compare (HV discharge time) | 100 | — | — | μs | |

**NOTES:**

**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.



**Trademarks**



QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

*Microchip received ISO/TS-16949:2002 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.*

# WORLDWIDE SALES AND SERVICE

## AMERICAS

**Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

**Atlanta**
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

**Boston**
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

**Chicago**
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

**Cleveland**
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

**Dallas**
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

**Detroit**
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

**Kokomo**
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

**Los Angeles**
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

**Santa Clara**
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

**Toronto**
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

## ASIA/PACIFIC

**Asia Pacific Office**
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

**Australia - Sydney**
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

**China - Beijing**
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

**China - Chengdu**
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

**China - Hong Kong SAR**
Tel: 852-2401-1200
Fax: 852-2401-3431

**China - Nanjing**
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

**China - Qingdao**
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

**China - Shanghai**
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

**China - Shenyang**
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

**China - Shenzhen**
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

**China - Wuhan**
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

**China - Xiamen**
Tel: 86-592-2388138
Fax: 86-592-2388130

**China - Xian**
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

**China - Zhuhai**
Tel: 86-756-3210040
Fax: 86-756-3210049

## ASIA/PACIFIC

**India - Bangalore**
Tel: 91-80-3090-4444
Fax: 91-80-3090-4080

**India - New Delhi**
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

**India - Pune**
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

**Japan - Yokohama**
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

**Korea - Daegu**
Tel: 82-53-744-4301
Fax: 82-53-744-4302

**Korea - Seoul**
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

**Malaysia - Kuala Lumpur**
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

**Malaysia - Penang**
Tel: 60-4-227-8870
Fax: 60-4-227-4068

**Philippines - Manila**
Tel: 63-2-634-9065
Fax: 63-2-634-9069

**Singapore**
Tel: 65-6334-8870
Fax: 65-6334-8850

**Taiwan - Hsin Chu**
Tel: 886-3-6578-300
Fax: 886-3-6578-370

**Taiwan - Kaohsiung**
Tel: 886-7-536-4818
Fax: 886-7-536-4803

**Taiwan - Taipei**
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

**Thailand - Bangkok**
Tel: 66-2-694-1351
Fax: 66-2-694-1350

## EUROPE

**Austria - Wels**
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

**Denmark - Copenhagen**
Tel: 45-4450-2828
Fax: 45-4485-2829

**France - Paris**
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

**Germany - Munich**
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

**Italy - Milan**
Tel: 39-0331-742611
Fax: 39-0331-466781

**Netherlands - Drunen**
Tel: 31-416-690399
Fax: 31-416-690340

**Spain - Madrid**
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

**UK - Wokingham**
Tel: 44-118-921-5869
Fax: 44-118-921-5820

03/26/09