# PIC16F818/819

## FLASH Memory Programming Specification

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

- PIC16F818
- PIC16F819

## 1.0 PROGRAMMING THE PIC16F818/819

The PIC16F818/819 is programmed using a serial method. The Serial mode will allow the PIC16F818/819 to be programmed while in the user's system. This allows for increased design flexibility. This programming specification applies to PIC16F818/819 devices in all packages.

### 1.1 Programming Algorithm Requirements

The programming algorithm used depends on the operating voltage ($V_{DD}$) of the PIC16F818/819 device.

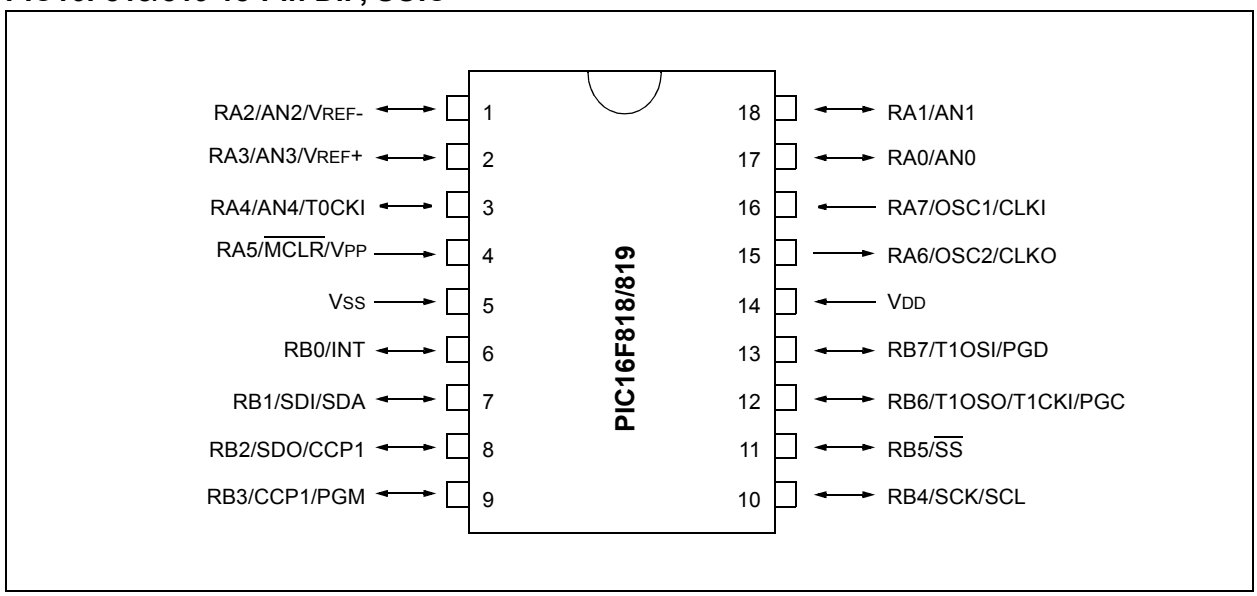
| Algorithm # | $V_{DD}$ Range |
|---|---|
| 1 | $2.0V \leq V_{DD} < 5.5V$ |
| 2 | $4.5V \leq V_{DD} \leq 5.5V$ |

Both algorithms can be used with the two available programming entry methods. The first method, called Low Voltage ICSP™ or LVP for short, applies $V_{DD}$ to $\overline{MCLR}$ and uses the I/O pin RB3 to enter Programming mode. When RB3 is driven to $V_{DD}$ from ground, the PIC16F818/819 device enters Programming mode. The second method follows the normal Microchip Programming mode entry of holding pins RB6 and RB7 low, while raising $\overline{MCLR}$ pin from $V_{IL}$ to $V_{IHH}$ (13V ± 0.5V).

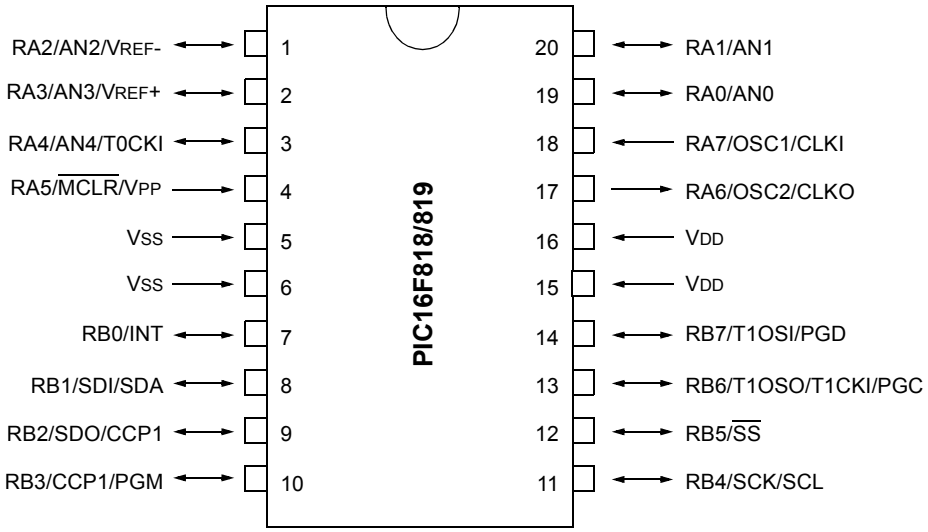
### 1.2 Programming Mode

The Programming mode for the PIC16F818/819 allows programming of user program memory, data memory, special locations used for ID, and the configuration word.

**PIC16F818/819 18-Pin DIP, SOIC**



### PIC16F818/819 20-Pin SSOP




### PIC16F818/819 28-Pin QFN

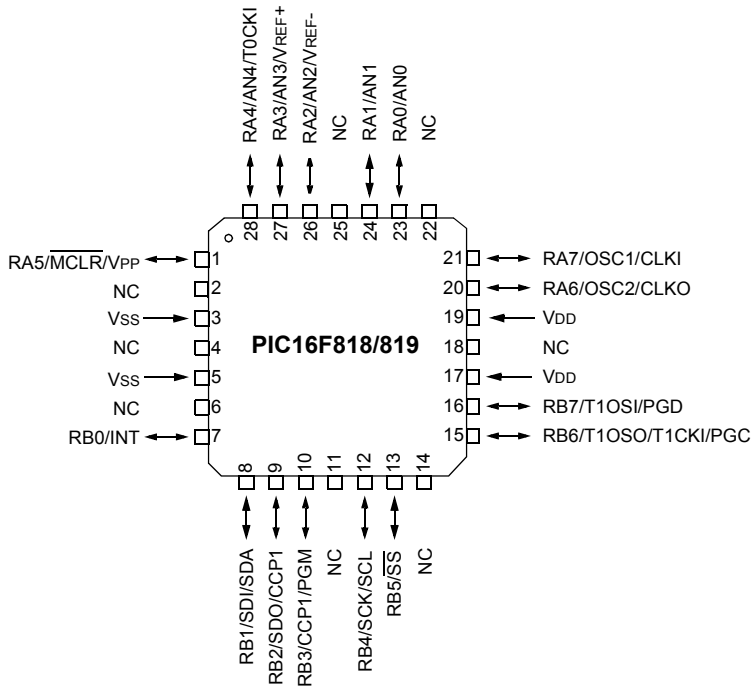


**TABLE 1-1: PIN DESCRIPTIONS (DURING PROGRAMMING): PIC16F818/819**

| Pin Name | During Programming | | |
|---|---|---|---|
| | Function | Pin Type | Pin Description |
| RB3 | PGM | I | Low Voltage ICSP programming input if LVP configuration bit equals '1' |
| RB6 | CLOCK | I | Clock input |
| RB7 | DATA | I/O | Data input/output |
| $\overline{MCLR}$ | $V_{PP}$ | P* | Program Mode Select |
| $V_{DD}$ | $V_{DD}$ | P | Power Supply |
| $V_{SS}$ | $V_{SS}$ | P | Ground |

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

* To activate the Programming mode, high voltage needs to be applied to the $\overline{MCLR}$ input. Since $\overline{MCLR}$ is used for a level source, this means that $\overline{MCLR}$ does not draw any significant current.

## 2.0 PROGRAM MODE ENTRY

### 2.1 User Program Memory Map

The user memory space extends from 0x0000 to 0x1FFF (8K). In Programming 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 0x07FF, then increment to 0x0800 and access 0x0000. Once the PC reaches 0x1FFF, it will increment to 0x2000. From 0x2000, the PC will increment up to 0x3FFF and wrap around 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 mode, as described in Section 2.4.

| Device | Program FLASH |
|---|---|
| PIC16F818 | 1K |
| PIC16F819 | 2K |

In the configuration memory space, 0x2000-0x201F are physically implemented. However, only locations 0x2000 through 0x2007 are available. Other locations are reserved. Locations beyond 0x201F will physically access user memory (see Figure 2-1).

### 2.2 Data EEPROM Memory

The EEPROM data memory space is a separate block of high endurance memory that the user accesses, using a special sequence of instructions. The amount of data EEPROM memory depends on the device and is shown below in number of bytes.

| Device | # of Bytes |
|---|---|
| PIC16F818 | 128 |
| PIC16F819 | 256 |

The contents of data EEPROM memory have the capability to be embedded into the HEX file.

The programmer should be able to read data EEPROM information from a HEX file and conversely (as an option), write data EEPROM contents to a HEX file, along with program memory information and configuration bit information.

The 256 data memory locations are logically mapped and use PC<7:0>. The format for data memory storage is one data byte per address location, LSb aligned.

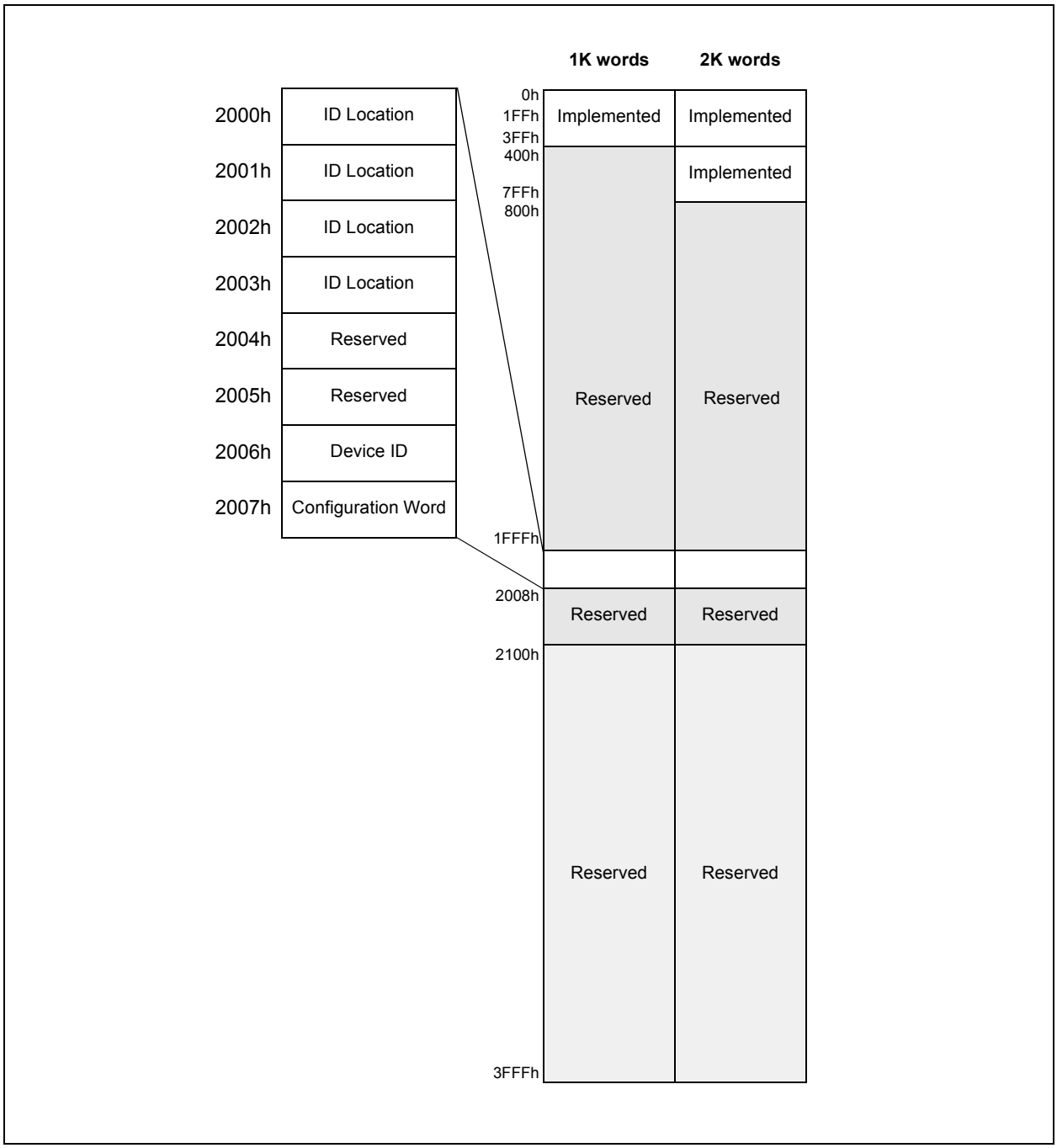
## 2.3 ID Locations

A user may store identification information (ID) in four ID locations. The ID locations are mapped in [0x2000 : 0x2003]. It is recommended that the user use only the four Least Significant bits of each ID location. In some devices, the ID locations read out in an unscrambled fashion after code protection is enabled.

For these devices, it is recommended that ID location is written as "11 1111 1000 bbbb", where 'bbbb' is ID information.

In other devices, the ID locations read out normally, even after code protection. To understand how the devices behave, refer to Table 5-1.

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



## 2.4 Program Mode

Program mode is entered by holding pins RB6 and RB7 low, while raising MCLR pin from VIL to VIHH (high voltage). In this mode, the state of the RB3 pin does not affect programming, which is used for low voltage ICSP programming. Once in Program mode, the user program memory and the configuration memory can be accessed and programmed in serial fashion. The mode of operation is serial, and the memory accessed is the user program memory. RB6 and RB7 are Schmitt Trigger inputs in this mode.

| | |
|---|---|
| **Note:** | The OSC must not have 72 osc clocks while the device MCLR is between VIL and VIHH. |

The sequence that enters the device into the Programming mode places all other logic into the RESET state (the MCLR pin was initially at VIL). This means all I/O are in the RESET state (high impedance inputs).

| | |
|---|---|
| **Note:** | The MCLR pin should be raised from below VIL to above the minimum VIHH (VPP), within 100 µs of VDD rise. This ensures that the device always enters Programming mode before any instructions that may be in program memory can be executed. Otherwise, unintended instruction execution could occur when the INTRC clock source is configured as the primary clock. |

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 2-1.

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

1. Set pointer to row location.
2. Issue a 'Begin Erase' command.
3. Wait tprog2.
4. Issue an 'End Programming' command.
5. Load a word at the current program memory address using the 'Load Data' command.
6. Issue an 'Increment Address' command.
7. Load a word at the current program memory address using the 'Load Data' command.
8. Repeat Step 6 and Step 7 two times.
9. Issue a 'Begin Programming' command to begin programming.
10. Wait tprog1.
11. Issue an 'End Programming' command.
12. Increment to the next address.
13. Repeat steps 5 through 12 seven times to program one row.

The address and program counter are reset to 0x0000 by resetting the device (taking MCLR below VIL) and re-entering Programming mode. Program and configuration memory may then be read or verified using the 'Read Data' and 'Increment Address' commands.

### 2.4.1 LOW VOLTAGE ICSP PROGRAMMING MODE

Low Voltage ICSP Programming mode allows a PIC16F818/819 device to be programmed using VDD only. However, when this mode is enabled by a configuration bit (LVP), the PIC16F818/819 device dedicates RB3 to control entry/exit into Programming mode.

When LVP bit is set to '1', the low voltage ICSP programming entry is enabled. Since the LVP configuration bit allows low voltage ICSP programming entry in its erased state, an erased device will have the LVP bit enabled at the factory. While LVP is '1', RB3 is dedicated to low voltage ICSP programming. Bring RB3 and then, MCLR to VDD to enter Programming mode. All other specifications for high voltage ICSP apply.

To disable Low Voltage ICSP mode, the LVP bit must be programmed to '0'. This must be done while entered with the High Voltage Entry mode (LVP bit = 1). RB3 is now a general purpose I/O pin.

### 2.4.2 SERIAL PROGRAM OPERATION

The RB6 pin is used as a clock input pin, and the RB7 pin is used to enter command bits, and to input or output data during serial operation. To input a command, the clock pin (RB6) is cycled six times. Each command bit is latched on the falling edge of the clock, with the Least Significant bit (LSb) of the command being input first. The data on RB7 is required to have a minimum setup (tset1) and hold (thold1) time (see AC/DC specifications), with respect to the falling edge of the clock. Commands with associated data (read and load) are specified to have a minimum delay (tdly1) 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 (0) and the last cycle being a STOP bit (0). Data is transferred LSb first.

During a read operation, the LSb will be transmitted onto RB7 on the rising edge of the second cycle, and during a load operation, the LSb will be latched on the falling edge of the second cycle. A minimum 1 µs delay (tdly2) is specified between consecutive commands.

All commands and data words are transmitted LSb first. The data is transmitted on the rising edge, and latched on the falling edge of the clock. To allow decoding of commands and reversal of data pin configuration, a time separation of at least 1 µs (tdly1) is required between a command and a data word, or another command.

The available commands are described in the following paragraphs and listed in Table 2-1.

#### 2.4.2.1 Load Configuration

After receiving this command, the program counter (PC) will be set to 0x2000. By then applying 16 cycles to the clock pin, the chip will load 14 bits in a “data word”, as described above, to be programmed into the configuration memory. A description of the memory mapping schemes of the program memory for normal operation and Configuration mode operation is shown in Figure 2-1. After the configuration memory is entered, the only way to get back to the user program memory is to exit the Program mode by taking $\overline{MCLR}$ low ($V_{IL}$).

#### 2.4.2.2 Load Data for Program Memory

After receiving this command, the chip will load one word (with 14 bits as a “data word”) to be programmed into user program memory when 16 cycles are applied. A timing diagram for this command is shown in Figure 6-1.

#### 2.4.2.3 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 up to 256 bytes. If the device is code protected, the data is read as all zeros. A timing diagram for this command is shown in Figure 6-2.

#### 2.4.2.4 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 RB7 pin will go into Output mode on the second rising clock edge, and it will revert back to Input mode (hi-impedance) after the 16th rising edge. A timing diagram of this command is shown in Figure 6-3.

#### 2.4.2.5 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 RB7 pin will go into Output mode on the second rising edge, and it will revert back to Input mode (hi-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. A timing diagram for this command is shown in Figure 6-4.

#### 2.4.2.6 Increment Address

The PC is incremented when this command is received. A timing diagram of this command is shown in Figure 6-5.

| | |
|---|---|
| **Note:** | Upon entry into Programming mode, a “Load Data for Program Memory” or “Load Data for Data Memory” command of 0x01 must be given before a Begin Erase or Begin Programming command is initiated. This will ensure that the programming pointer is pointing to the correct location in data or program memory. |

#### 2.4.2.7 Begin Erase (Program and Data Memory)

The erase block size for program memory is 32 words (row) and 1 word for data memory. The row or word to be programmed must first be erased. This is done by setting the pointer to a location in the row or word and then performing a ‘Begin Erase’ command. The row or word is then erased. The user must allow the combined time for row erase and programming, as specified in the electrical specifications, for programming to complete. This is an externally timed event.

The internal timer is not used for this command, so the ‘End Programming’ command must be used to stop erase.

| | |
|---|---|
| **Note 1:** | The code protect bits cannot be erased with this command. |
| **2:** | All Begin Erase operations can take place over the entire $V_{DD}$ range. |

A timing diagram for this command is shown in Figure 6-6.

#### 2.4.2.8 Begin Programming Only

Programming of program and data memory will begin after this command is received and decoded. The user must allow the time for programming, as specified in the electrical specifications, for programming to complete. An ‘End Programming’ command is required.

The internal timer is not used for this command, so the ‘End Programming’ command must be used to stop programming.

1. If the address is pointing to user memory, the user memory alone will be affected.
2. If the address is pointing to the physically implemented configuration memory (2000h - 2007h), the configuration memory will be written. The configuration word will not be written unless the address is specifically pointing to 2007h.

A timing diagram for this command is shown in Figure 6-7.

#### 2.4.2.9 End Programming

After receiving this command, the chip stops programming the memory (configuration memory or user program memory), that it was programming at the time.

| **Note:** | This command will also set the write data shift latches to all '1's to avoid issues with downloading only one word before the write. |
|---|---|

**TABLE 2-1: COMMAND MAPPING FOR PIC16F818/819**

| Command | Mapping (MSB … LSB) | | | | | Data | Voltage Range |
|---|---|---|---|---|---|---|---|
| Load Configuration | 0 | 0 | 0 | 0 | 0 | 0, data (14), 0 | 2.0V - 5.5V |
| Load Data for Program Memory | 0 | 0 | 0 | 1 | 0 | 0, data (14), 0 | 2.0V - 5.5V |
| Read Data from Program Memory | 0 | 0 | 1 | 0 | 0 | 0, data (14), 0 | 2.0V - 5.5V |
| Increment Address | 0 | 0 | 1 | 1 | 0 | | 2.0V - 5.5V |
| Begin Erase | 0 | 1 | 0 | 0 | 0 | externally timed | 2.0V - 5.5V |
| Begin Programming Only Cycle | 1 | 1 | 0 | 0 | 0 | externally timed | 2.0V - 5.5V |
| Bulk Erase Program Memory | 0 | 1 | 0 | 0 | 1 | externally timed | 4.5V - 5.5V |
| Bulk Erase Data Memory | 0 | 1 | 0 | 1 | 1 | externally timed | 4.5V - 5.5V |
| Chip Erase | 1 | 1 | 1 | 1 | 1 | internally timed | 4.5V - 5.5V |
| Load Data for Data Memory | 0 | 0 | 0 | 1 | 1 | 0, zeroes (6), data (14), 0 | 2.0V - 5.5V |
| Read Data from Data Memory | 0 | 0 | 1 | 0 | 1 | 0, zeroes (6), data (14), 0 | 2.0V - 5.5V |
| End Programming | 1 | 0 | 1 | 1 | 1 | | |

## 2.5 Erasing Program and Data Memory

Depending on the state of the code protection bits, program and data memory will be erased using different methods. The first two commands are used when both program and data memories are not code protected. The third command is used when either memory is code protected, or if you want to also erase the code protect bits. A device programmer should determine the state of the code protection bits and then apply the proper command to erase the desired memory.

### 2.5.1 ERASING NON-CODE PROTECTED PROGRAM AND DATA MEMORY

When both program and data memories are not code protected, they must be individually erased using the following commands. The only way that both memories are erased using a single command is if code protection is enabled for one of the memories. These commands do not erase the configuration word or ID locations.

#### 2.5.1.1 Bulk Erase Program Memory

When this command is performed, and is followed by a 'Begin Erase' command, the entire program memory will be erased.

If the address is pointing to user memory, only the user memory will be erased.

If the address is pointing to the configuration memory (2000h - 2007h), then both the user memory and the configuration memory will be erased. The configuration word will not be erased, even if the address is pointing to location 2007h.

Previously, a load data with 0FFh command was recommended before any Bulk Erase. On these devices, this will not be required.

The Bulk Erase command is disabled when the CP bit is programmed to '0', enabling code protect.

A timing diagram for this command is shown in Figure 6-8.

#### 2.5.1.2 Bulk Erase Data Memory

When this command is performed, and is followed by a ‘Begin Erase’ command, the entire data memory will be erased.

The Bulk Erase Data command is disabled when the CPD bit is programmed to ‘0’, enabling protected data memory. A timing diagram for this command is shown in Figure 6-9.

| | |
|---|---|
| **Note:** | All Bulk Erase operations must take place at the 4.5V to 5.5V VDD range. |

#### 2.5.1.3 Chip Erase

This command, when performed, will erase the program memory, EE data memory, and all of the code protection bits. All on-chip FLASH and EEPROM memory is erased, regardless of the address contained in the PC.

When a Chip Erase command is issued and the PC points to (0000h - 1FFFh), the configuration word (2007h) and the user program memory will be erased. When a Chip Erase command is issued and the PC points to (2000h - 2007h), all of the configuration memory, program memory and data memory will be erased.

The Chip Erase is internally self-timed to ensure that all program and data memory are erased before the code protect bits are erased. A timing diagram for this command is shown in Figure 6-10.

| | |
|---|---|
| **Note:** | The Chip Erase operation must take place at the 4.5V to 5.5V VDD range. |

### 2.5.2 ERASING CODE PROTECTED MEMORY

For the PIC16F818/819 devices, once code protection is enabled, all protected program and data memory locations read all '0's and further programming is disabled. The ID locations and configuration word read out unscrambled and can be reprogrammed normally. The only command to erase a code protected PIC16F818/819 device is the Chip Erase. This erases program memory, data memory, configuration bits and ID locations, as described in Section 2.5.1.3. **Since all data within the program and data memory will be erased when this command is executed, the security of the data or code is not compromised.**

**FIGURE 2-2: ALGORITHM 1 FLOW CHART – PROGRAM MEMORY (2.0V ≤ VDD < 5.5V)**

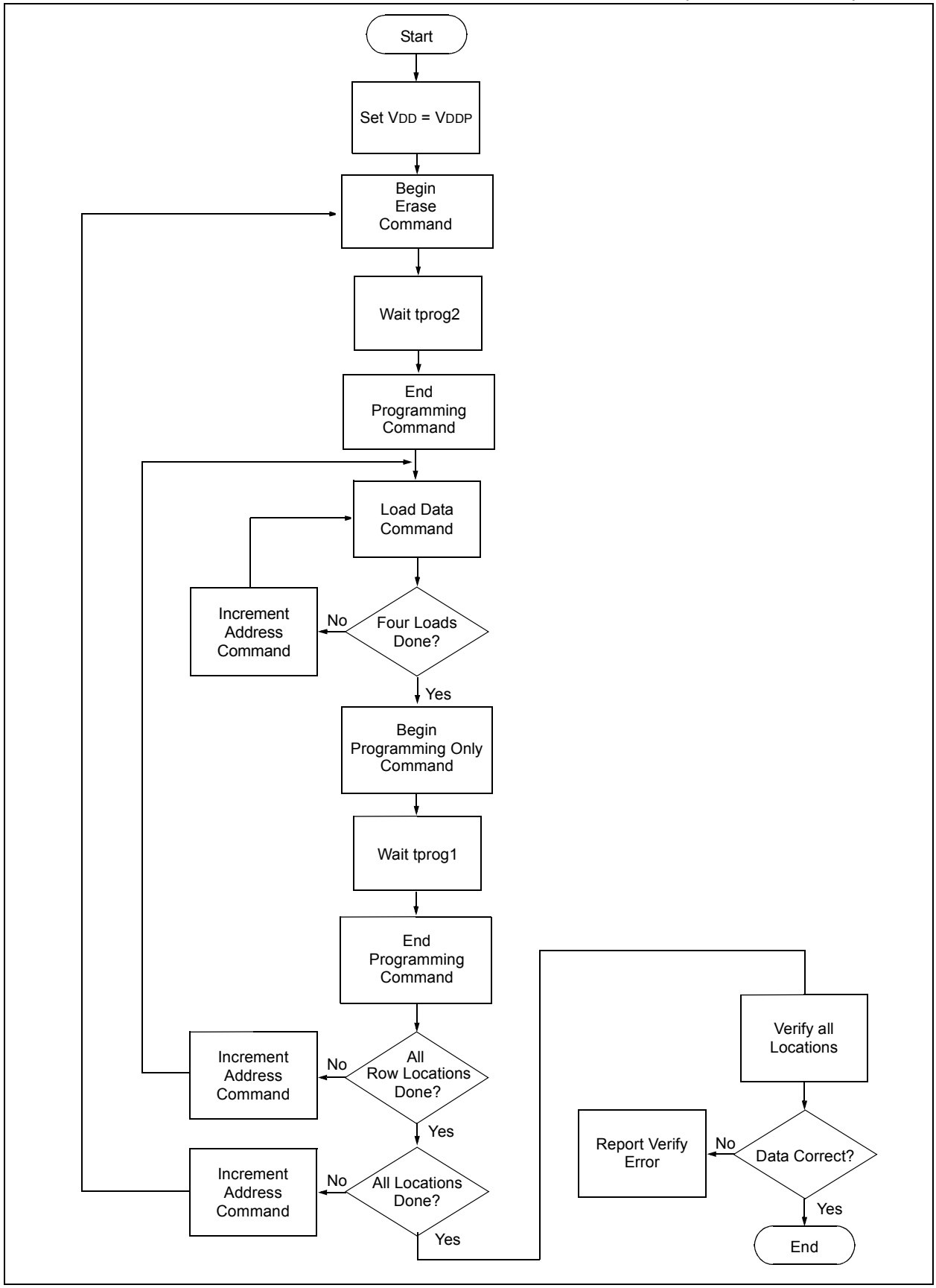


**FIGURE 2-3: ALGORITHM 2 FLOW CHART – PROGRAM MEMORY (4.5V ≤ VDD ≤ 5.5V)**

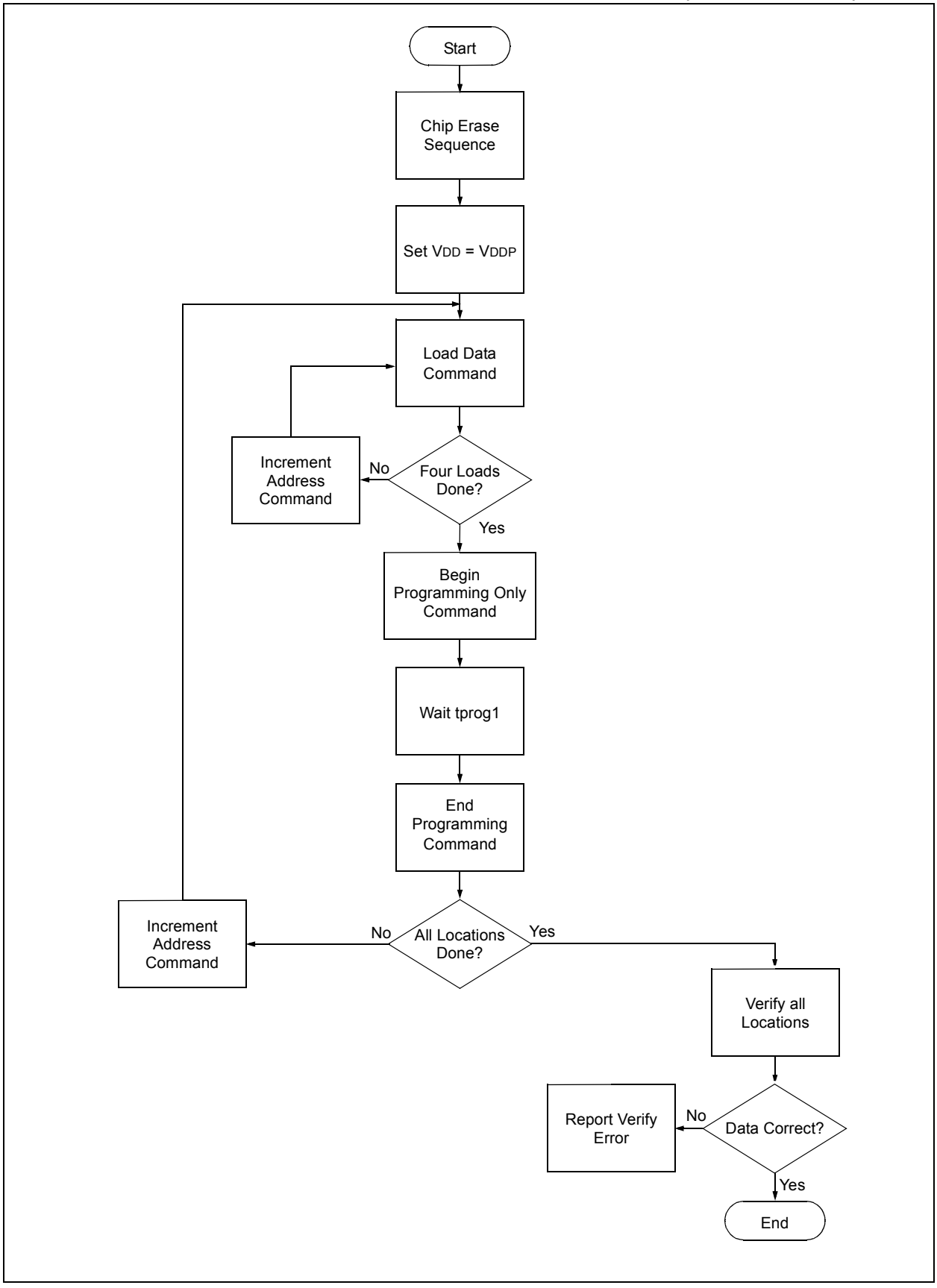
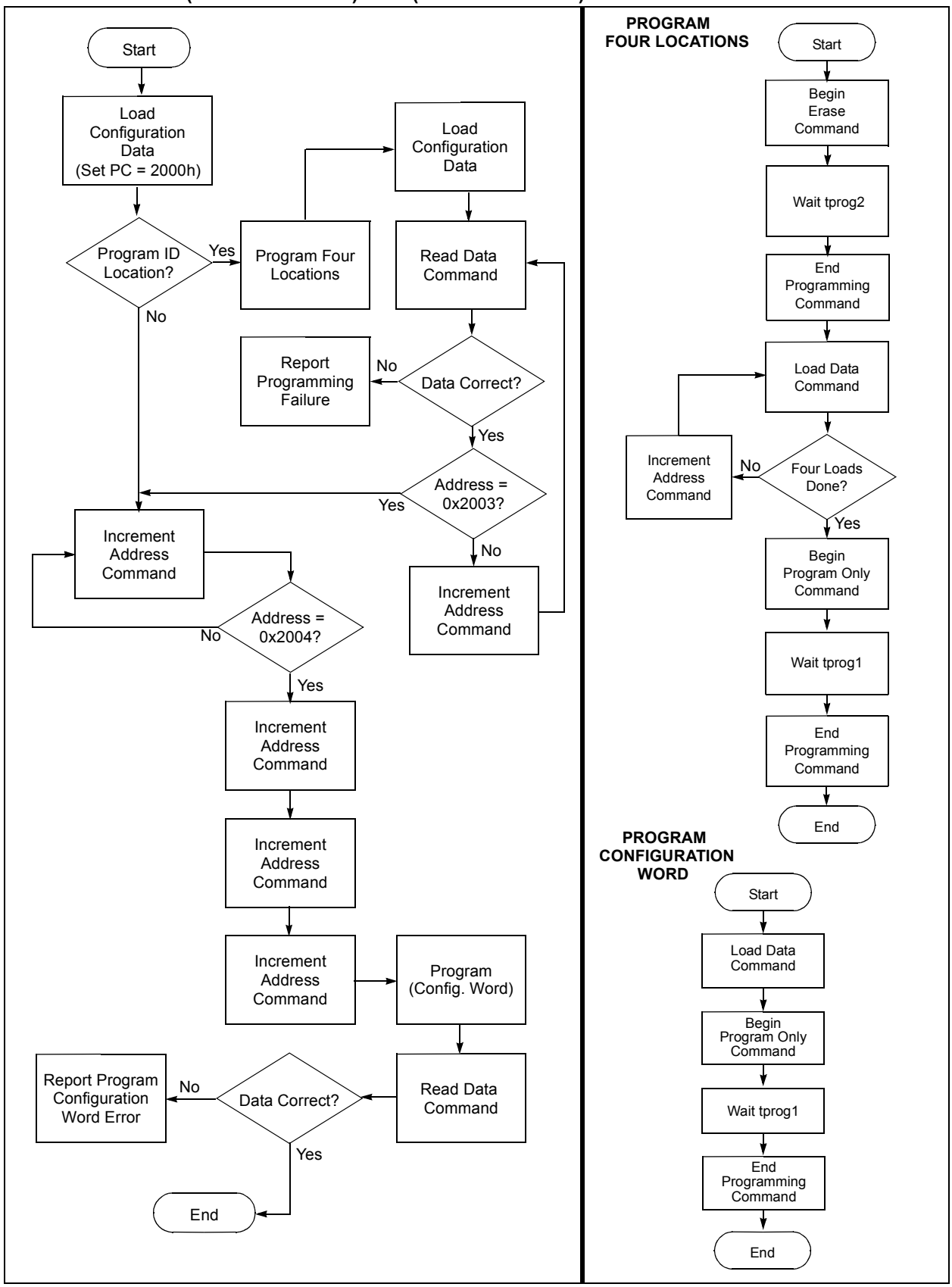


**FIGURE 2-4: FLOW CHART – PIC16F818/819 CONFIGURATION MEMORY (2.0V ≤ VDD < 5.5V) AND (4.5V ≤ VDD < 5.5V)**



## 3.0 CONFIGURATION WORD

The PIC16F818/819 has several configuration bits. These bits can be written to ‘0’ or ‘1’ with the Begin Program Only command. A Begin Erase command is not required when programming configuration memory.

### 3.1 Device ID Word

The device ID word for the PIC16F818/819 is located at 2006h.

**TABLE 3-1: DEVICE ID VALUE**

| Device | Device ID Value | |
|---|---|---|
| | Dev | Rev |
| PIC16F818 | 00 0100 1100 | XXXX |
| PIC16F819 | 00 0100 1110 | XXXX |

**REGISTER 3-1: CONFIGURATION WORD (ADDRESS 2007h)(1)**

| u-1 | u-1 | u-1 | u-1 | u-1 | u-1 | u-1 | u-1 | u-1 | u-1 | u-1 | u-1 | u-1 | u-1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CP | CCPMX | DEBUG | WRT1 | WRT0 | CPD | LVP | BOREN | MCLRE | FOSC2 | $\overline{\text{PWRTEN}}$ | WDTEN | F0SC1 | F0SC0 |
| bit13 | | | | | | | | | | | | | bit0 |

bit 13 **CP:** FLASH Program Memory Code Protection bit
1 = Code protection off
0 = All memory locations code protected

bit 12 **CCPMX:** CCP1 Pin Selection bit
1 = CCP1 function on RB2
0 = CCP1 function on RB3

bit 11 **DEBUG:** In-Circuit Debugger Mode bit
1 = In-Circuit Debugger disabled, RB6 and RB7 are general purpose I/O pins
0 = In-Circuit Debugger enabled, RB6 and RB7 are dedicated to the debugger

bit 10-9 **WRT1:WRT0:** FLASH Program Memory Write Enable bits
11 = Write protection off
10 = 0000h to 01FFh write protected, 0200h to 07FFh may be modified by EECON control
01 = 0000h to 03FFh write protected, 0400h to 07FFh may be modified by EECON control
00 = 0000h to 05FFh write protected, 0600h to 07FFh may be modified by EECON control

bit 8 **CPD:** Data EE Memory Code Protection bit
1 = Code protection off
0 = Data EE memory locations code protected

bit 7 **LVP:** Low Voltage Programming Enable bit
1 = RB3/PGM pin has PGM function, low voltage programming enabled
0 = RB3/PGM pin has digital I/O function, HV on $\overline{\text{MCLR}}$ must be used for programming

bit 6 **BOREN:** Brown-out Reset Enable bit(2)
1 = BOR enabled
0 = BOR disabled

bit 5 **MCLRE:** RA5/$\overline{\text{MCLR}}$ Pin Function Select bit
1 = RA5/$\overline{\text{MCLR}}$ pin function is $\overline{\text{MCLR}}$
0 = RA5/$\overline{\text{MCLR}}$ pin function is digital I/O, $\overline{\text{MCLR}}$ internally tied to $V_{DD}$

bit 3 **$\overline{\text{PWRTEN}}$:** Power-up Timer Enable bit
1 = PWRT disabled
0 = PWRT enabled

bit 2 **WDTEN:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 4, 1-0 **FOSC2:FOSC0:** Oscillator Selection bits
111 = EXTRC oscillator; CLKO function on RA6/OSC2/CLKO pin
110 = EXTRC oscillator; port I/O function on RA6/OSC2/CLKO pin
101 = INTRC oscillator; CLKO function on RA6/OSC2/CLKO pin and port I/O function on RA7/OSC1/CLKI pin
100 = INTRC oscillator; port I/O function on both RA6/OSC2/CLKO pin and RA7/OSC1/CLKI pin
011 = EXTCLK; port I/O function on RA6/OSC2/CLKO pin
010 = HS oscillator
001 = XT oscillator
000 = LP oscillator

**Note 1:** The erased (unprogrammed) value of the configuration word is 3FFFh.
**2:** Enabling Brown-out Reset automatically enables Power-up Timer (PWRT), regardless of the value of bit $\overline{\text{PWRTEN}}$. Ensure the Power-up Timer is enabled any time Brown-out Reset is enabled.

.

| Legend: | | |
|---|---|---|
| R = Readable bit | P = Programmable bit | U = Unimplemented bit, read as '1' |
| - n = Value when device is unprogrammed | | u = Unchanged from programmed state |

## 4.0 EMBEDDING CONFIGURATION WORD AND ID INFORMATION IN HEX FILE

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

Specifically for the PIC16F818/819, the EEPROM data memory should also be embedded in the HEX file (see Section 2.2).

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

## 5.0 CHECKSUM COMPUTATION

Checksum is calculated by reading the contents of the PIC16F818/819 memory locations and adding up the opcodes, up to the maximum user addressable location (e.g., 0x1FF for the PIC16F818/819). Any carry bits exceeding 16-bits are neglected. Finally, the configuration word (appropriately masked) is added to the checksum. Checksum computation for each member of the PIC16F818/819 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 ID locations (when applicable)

The Least Significant 16 bits of this sum are 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 differently depending on the code protect setting, the table describes how to manipulate the actual program memory values to simulate the 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 ID locations can always be read.

Note that 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 COMPUTATION**

| Device | Code Protect | Checksum* | Blank Value | 0x25E6 at 0 and Max Address |
|---|---|---|---|---|
| PIC16F818 | OFF | SUM[0000:03FF] + (CFGW & 3FFF) | 3BFF | 07CD |
| | ON | (CFGW & 3FFF) + SUM_ID | 5BFE | 27CC |
| PIC16F819 | OFF | SUM[0000:07FF] + (CFGW & 3FFF) | 37FF | 03CD |
| | ON | (CFGW & 3FFF) + SUM_ID | 57FE | 23CC |

Legend: CFGW = Configuration Word
SUM[a:b] = [Sum of locations a to b inclusive]
SUM_ID = 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.
*Checksum = [Sum of all the individual expressions] **MODULO** [0xFFFF]
\+ = Addition
& = Bitwise AND

## 6.0 PROGRAM MODE ELECTRICAL CHARACTERISTICS

**TABLE 6-1: TIMING REQUIREMENTS FOR PROGRAM MODE**

| **AC/DC CHARACTERISTICS POWER SUPPLY PINS** | **Standard Operating Procedure (unless otherwise stated)** Operating temperature $0 \le T_A \le +70°C$ Operating Voltage $2.0V \le V_{DD} \le 5.5V$ | | | | | |
|---|---|---|---|---|---|---|
| **Characteristics** | **Sym** | **Min** | **Typ** | **Max** | **Units** | **Conditions/Comments** |
| **General** | | | | | | |
| $V_{DD}$ level for Begin Erase, Begin Program operations and EECON1 writes of program memory | $V_{DD}$ | 2.0 | — | 5.5 | V | |
| $V_{DD}$ level for Begin Erase, Begin Program operations and EECON1 writes of data memory | $V_{DD}$ | 2.0 | — | 5.5 | V | |
| $V_{DD}$ level for Bulk Erase, Chip Erase, and Begin Program operations of program and data memory | $V_{DD}$ | 4.5 | — | 5.5 | V | |
| Begin Programming Only cycle time | tprog1 | 1 | — | — | ms | Externally Timed, > 4.5V |
| | | 2 | — | — | ms | Externally Timed, < 4.5V |
| Begin Erase | tprog2 | 1 | — | — | ms | Externally Timed, > 4.5V |
| | | 2 | — | — | ms | Externally Timed, < 4.5V |
| Bulk Erase cycle time | tprog3 | 2 | — | — | ms | Externally Timed |
| Chip Erase cycle time | tprog4 | 8 | — | — | ms | Internally Timed |
| High voltage on $\overline{MCLR}$ and RA4/T0CKI for Program mode entry | $V_{IHH}$ | $V_{DD}$ + 3.5 | — | 13.5 | V | |
| $\overline{MCLR}$ rise time ($V_{SS}$ to $V_{HH}$) for Program mode entry | tVHHR | — | — | 1.0 | μs | |
| (RB6, RB7) input high level | $V_{IH1}$ | 0.8 $V_{DD}$ | — | — | V | Schmitt Trigger input |
| (RB6, RB7) input low level | $V_{IL1}$ | 0.2 $V_{DD}$ | — | — | V | Schmitt Trigger input |
| RB<7:4> setup time before $\overline{MCLR}\uparrow$ (Program mode selection pattern setup time) | tset0 | 100 | — | — | ns | |
| RB<7:4> hold time after $\overline{MCLR}\uparrow$ (Program mode selection pattern setup time) | thld0 | 5 | — | — | μs | |
| **Serial Program** | | | | | | |
| Data in setup time before clock↓ | tset1 | 100 | — | — | ns | |
| Data in hold time after clock↓ | thld1 | 100 | — | — | ns | |
| Data input not driven to next clock input (delay required between command/data or command/ command) | tdly1 | 1.0 | — | — | μs | $2.0V \le V_{DD} < 4.5V$ |
| | | 100 | — | — | ns | $4.5V \le V_{DD} \le 5.5V$ |
| Delay between clock↓ to clock↑ of next command or data | tdly2 | 1.0 | — | — | μs | $2.0V \le V_{DD} < 4.5V$ |
| | | 100 | — | — | ns | $4.5V \le V_{DD} \le 5.5V$ |
| Clock↑ to data out valid (during read data) | tdly3 | 80 | — | — | ns | |

**FIGURE 6-1: LOAD DATA FOR USER PROGRAM MEMORY COMMAND (PROGRAM)**

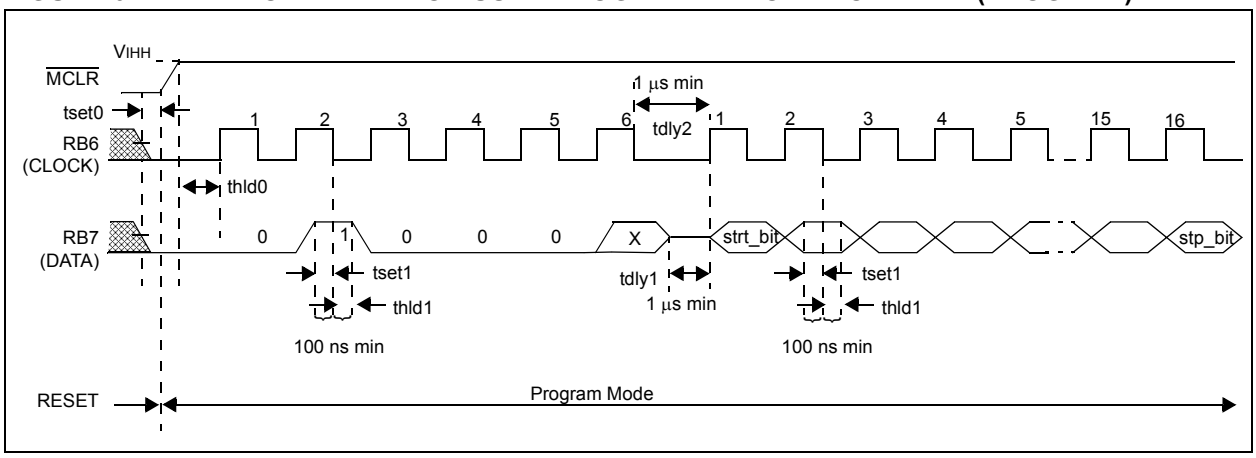



**FIGURE 6-2: LOAD DATA FOR USER DATA MEMORY COMMAND (PROGRAM)**

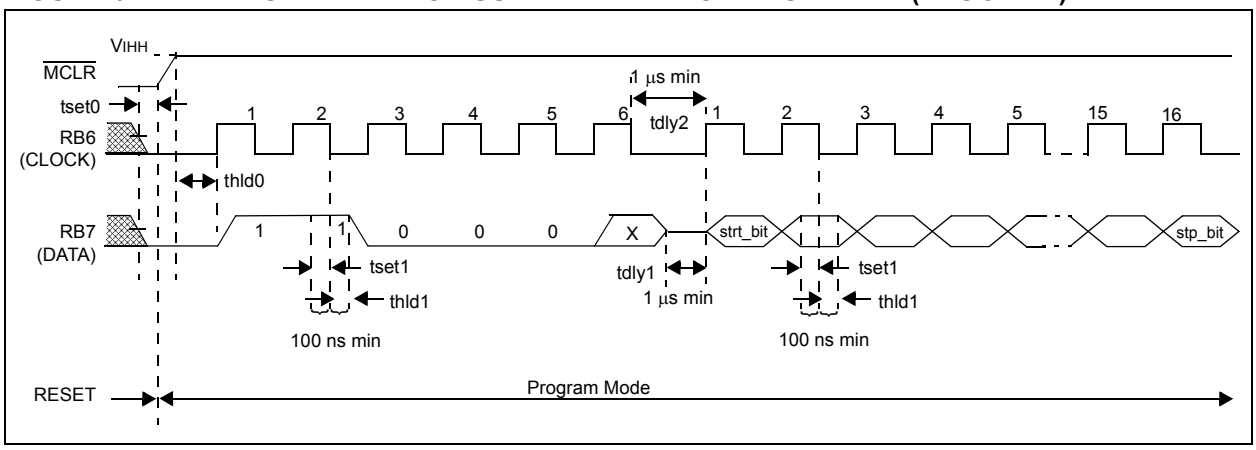



**FIGURE 6-3: READ DATA FROM PROGRAM MEMORY COMMAND (PROGRAM)**

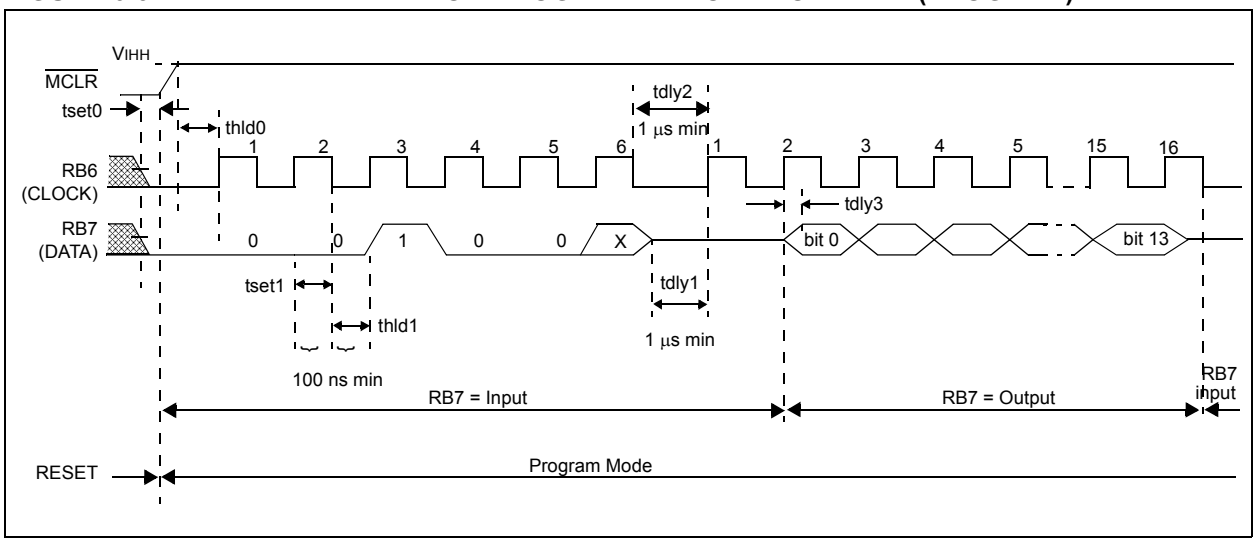


**FIGURE 6-4: READ DATA FROM DATA MEMORY COMMAND (PROGRAM)**

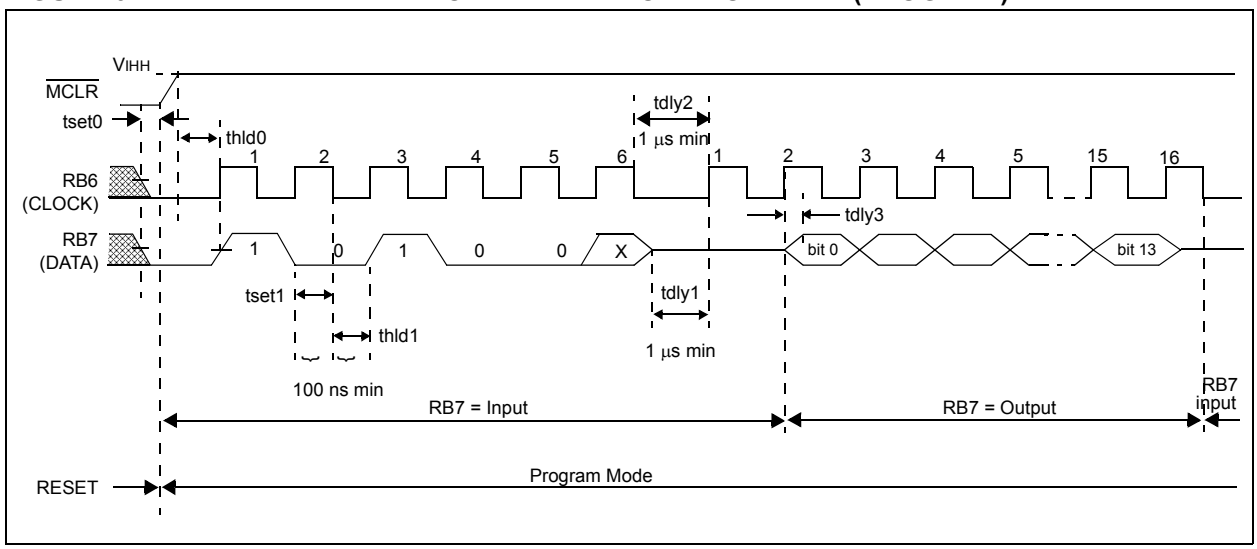



**FIGURE 6-5: INCREMENT ADDRESS COMMAND (SERIAL PROGRAM)**

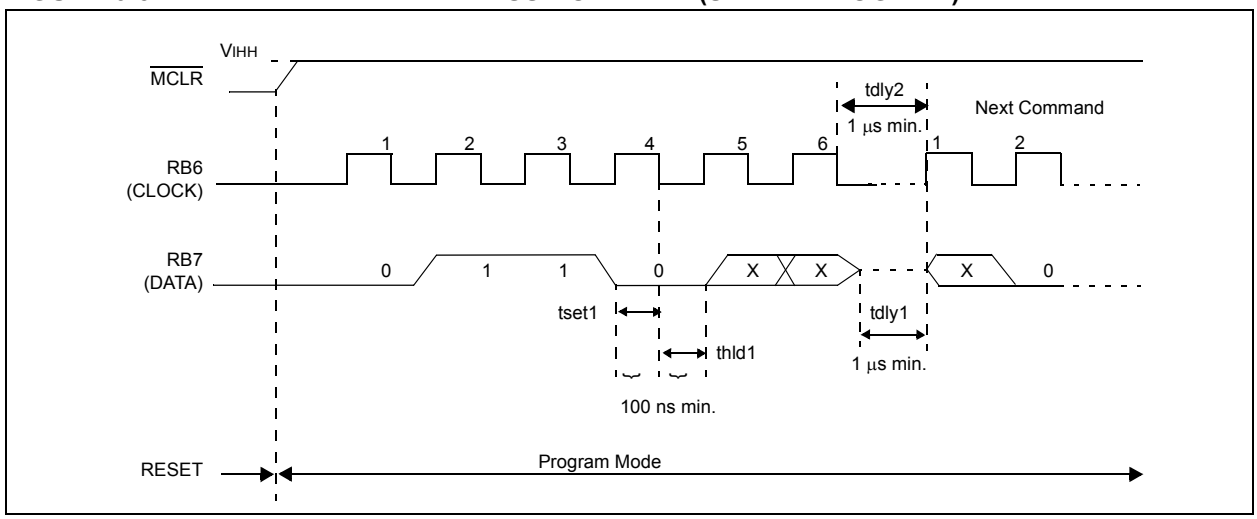



**FIGURE 6-6: BEGIN ERASE (SERIAL PROGRAM)**

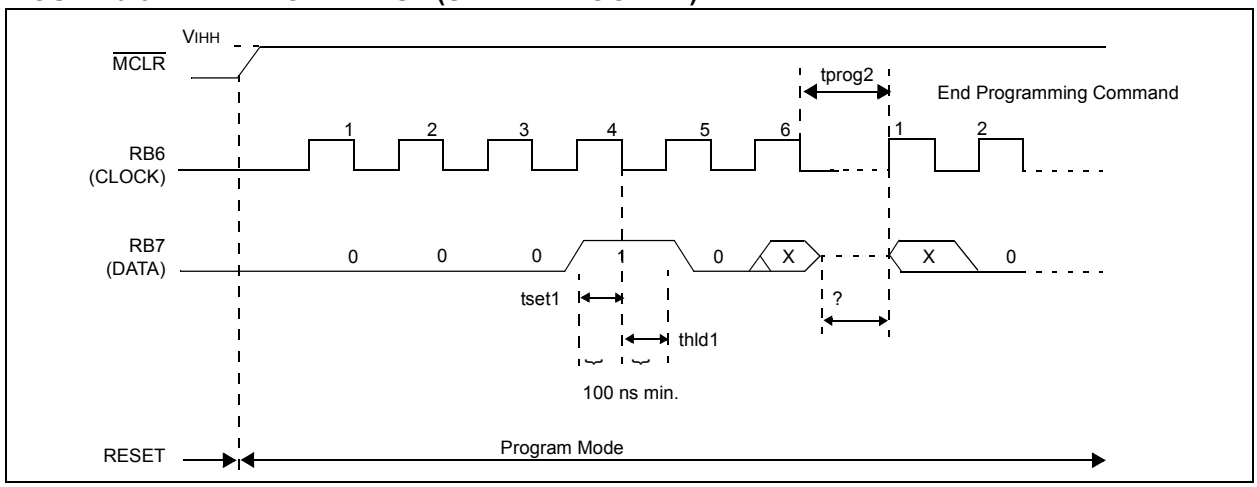


**FIGURE 6-7: BEGIN PROGRAMING ONLY COMMAND (SERIAL PROGRAM)**

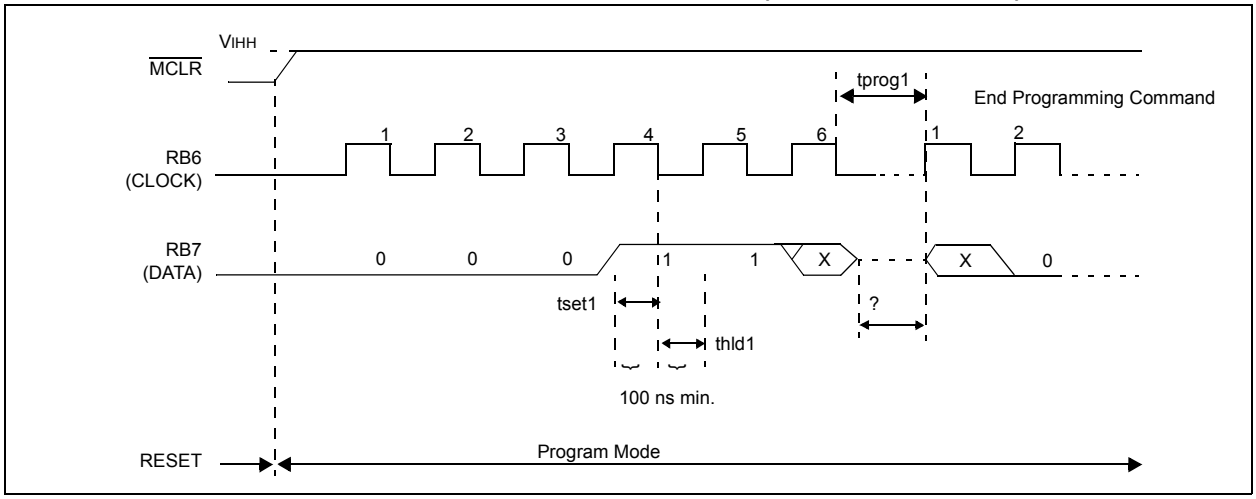



**FIGURE 6-8: BULK ERASE PROGRAM MEMORY COMMAND (SERIAL PROGRAM/VERIFY)**

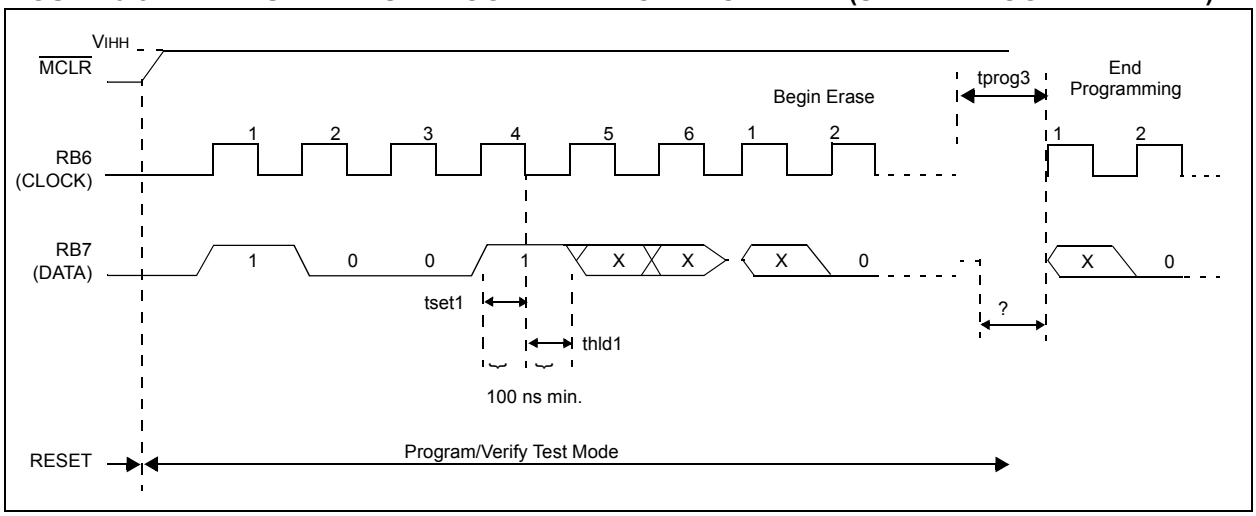



**FIGURE 6-9: BULK ERASE DATA MEMORY COMMAND (SERIAL PROGRAM/VERIFY)**

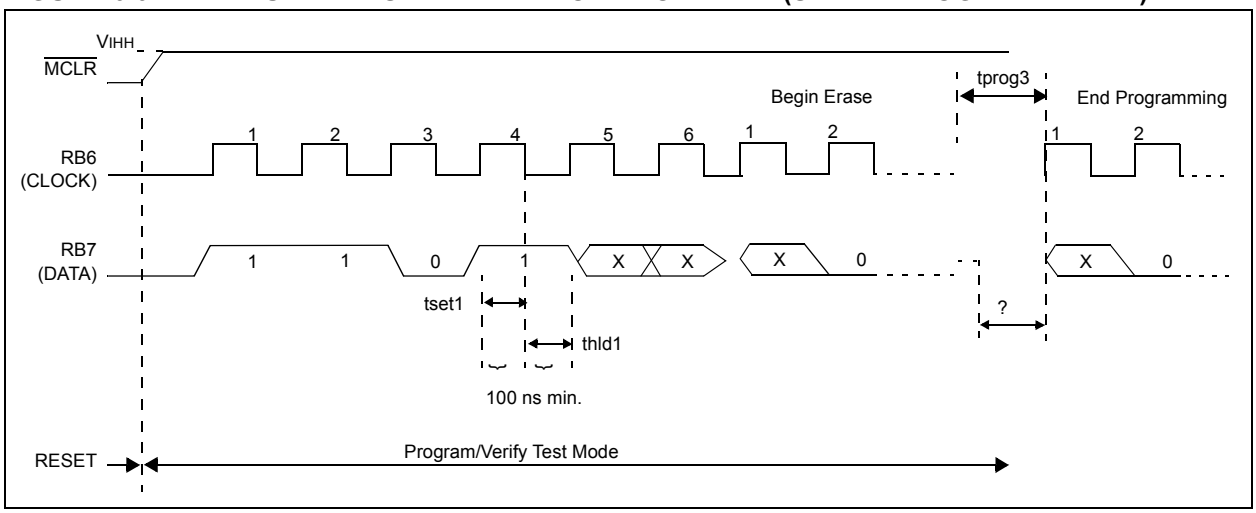
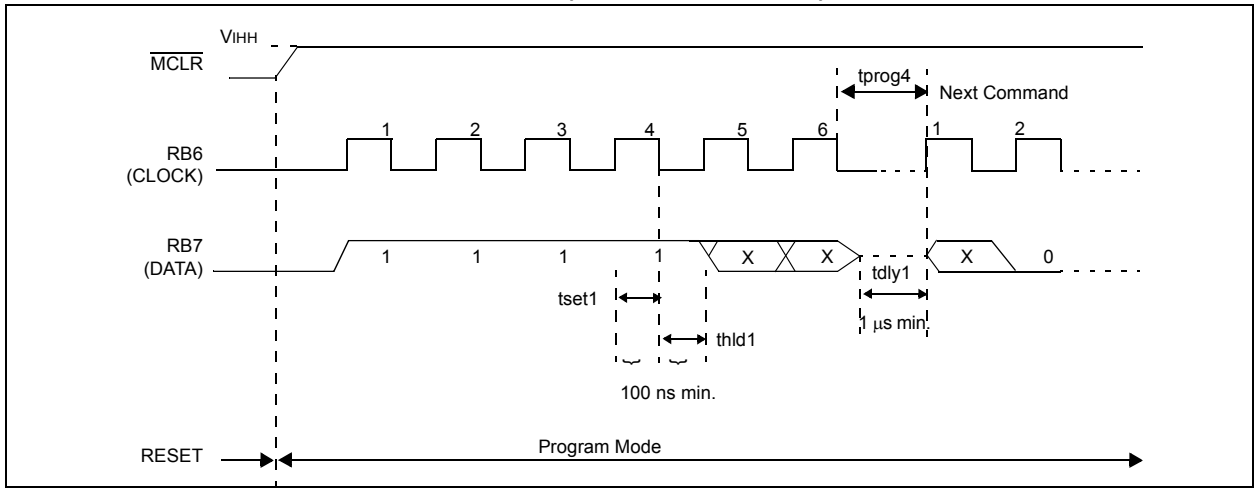


**FIGURE 6-10: CHIP ERASE COMMAND (SERIAL PROGRAM)**



**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.



DNV Certification, Inc. USA — ANSI-RAB QMS ACCREDITED — DNV — DNV MSC The Netherlands Accredited by the RvA — ISO 9001 / QS-9000 REGISTERED FIRM

*Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002. The Company’s quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001 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: 480-792-7627
Web Address: http://www.microchip.com

**Rocky Mountain**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966 Fax: 480-792-4338

**Atlanta**
3780 Mansell Road, Suite 130
Alpharetta, GA 30022
Tel: 770-640-0034 Fax: 770-640-0307

**Boston**
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848 Fax: 978-692-3821

**Chicago**
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

**Dallas**
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423 Fax: 972-818-2924

**Detroit**
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

**Kokomo**
2767 S. Albright Road
Kokomo, Indiana 46902
Tel: 765-864-8360 Fax: 765-864-8387

**Los Angeles**
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

**San Jose**
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

**Toronto**
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699 Fax: 905-673-6509

## ASIA/PACIFIC

**Australia**
Microchip Technology Australia Pty Ltd
Marketing Support Division
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

**China - Beijing**
Microchip Technology Consulting (Shanghai) Co., Ltd., Beijing Liaison Office
Unit 915
Bei Hai Wan Tai Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

**China - Chengdu**
Microchip Technology Consulting (Shanghai) Co., Ltd., Chengdu Liaison Office
Rm. 2401-2402, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-86766200 Fax: 86-28-86766599

**China - Fuzhou**
Microchip Technology Consulting (Shanghai) Co., Ltd., Fuzhou Liaison Office
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506 Fax: 86-591-7503521

**China - Hong Kong SAR**
Microchip Technology Hongkong Ltd.
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

**China - Shanghai**
Microchip Technology Consulting (Shanghai) Co., Ltd.
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

**China - Shenzhen**
Microchip Technology Consulting (Shanghai) Co., Ltd., Shenzhen Liaison Office
Rm. 1812, 18/F, Building A, United Plaza
No. 5022 Binhe Road, Futian District
Shenzhen 518033, China
Tel: 86-755-82901380 Fax: 86-755-82966626

**China - Qingdao**
Rm. B505A, Fullhope Plaza,
No. 12 Hong Kong Central Rd.
Qingdao 266071, China
Tel: 86-532-5027355 Fax: 86-532-5027205

**India**
Microchip Technology Inc.
India Liaison Office
Marketing Support Division
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

**Japan**
Microchip Technology Japan K.K.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

**Korea**
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5934

**Singapore**
Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

**Taiwan**
Microchip Technology (Barbados) Inc., Taiwan Branch
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

## EUROPE

**Austria**
Microchip Technology Austria GmbH
Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

**Denmark**
Microchip Technology Nordic ApS
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45 4420 9895 Fax: 45 4420 9910

**France**
Microchip Technology SARL
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

**Germany**
Microchip Technology GmbH
Steinheilstrasse 10
D-85737 Ismaning, Germany
Tel: 49-089-627-144-100
Fax: 49-089-627-144-44

**Italy**
Microchip Technology SRL
Via Quasimodo, 12
20025 Legnano (MI)
Milan, Italy
Tel: 39-0331-742611 Fax: 39-0331-466781

**United Kingdom**
Microchip Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5869 Fax: 44-118 921-5820

02/12/03