# PIC16F627A/628A/648A EEPROM Memory Programming Specification

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

- PIC16F627A
- PIC16F628A
- PIC16F648A
- PIC16LF627A
- PIC16LF628A
- PIC16LF648A

| **Note:** | All references to PIC16F627A/628A/648A also apply to PIC16LF62XA devices. |
|---|---|

## 1.0 PROGRAMMING THE PIC16F627A/628A/648A

The PIC16F627A/628A/648A is programmed using a serial method. The Serial mode will allow the PIC16F627A/628A/648A to be programmed while in the user's system. This allows for increased design flexibility. This programming specification applies to PIC16F627A/628A/648A devices in all packages.

### 1.1 Hardware Requirements

The PIC16F627A/628A/648A requires one programmable power supply for $V_{DD}$ (2.0V to 5.5V) and a $V_{PP}$ of 12V to 14V, or $V_{PP}$ of 4.5V to 5.5V, when using low voltage. Both supplies should have a minimum resolution of 0.25V.

### 1.2 Programming Mode

The Programming mode for the PIC16F627A/628A/648A allows programming of user program memory, data memory, special locations used for ID, and the Configuration Word.

**FIGURE 1-1: PIN DIAGRAM**

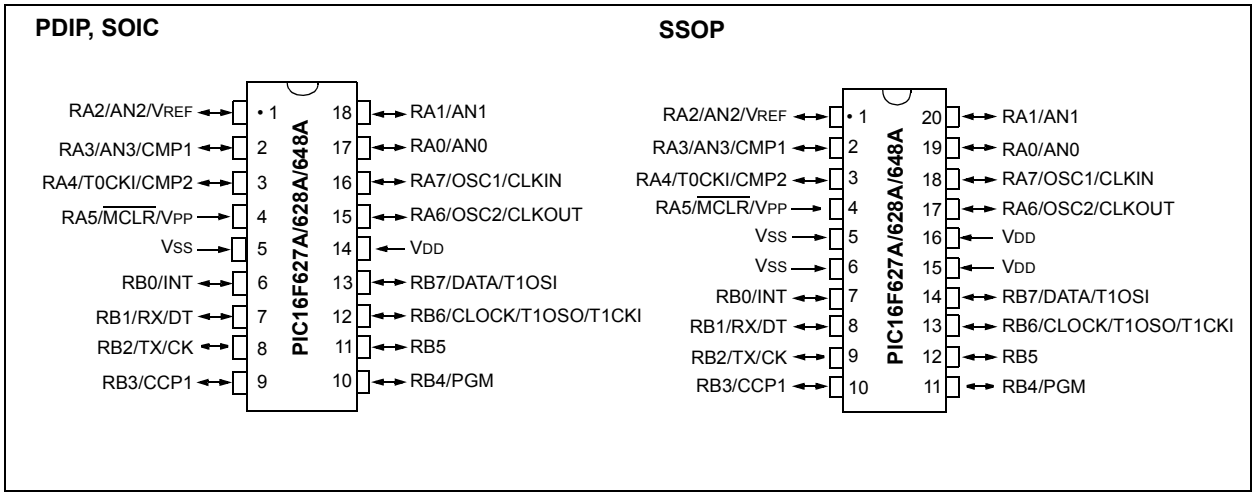


**FIGURE 1-2: 28-PIN QFN PIC16F627A/628A/648A DIAGRAM**

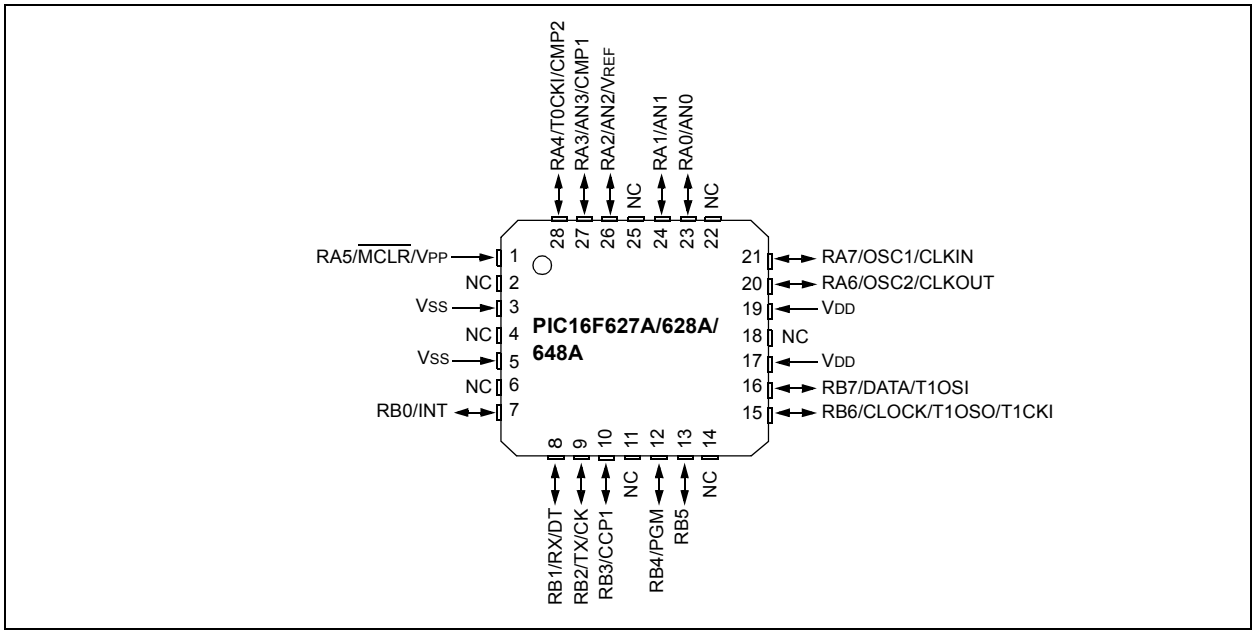



**TABLE 1-1: PIN DESCRIPTIONS (DURING PROGRAMMING): PIC16F627A/628A/648A**

| Pin Name | During Programming | | |
|---|---|---|---|
| | Function | Pin Type | Pin Description |
| RB4 | PGM | I | Low-voltage programming input if Configuration bit equals 1 |
| RB6 | CLOCK | I | Clock input |
| RB7 | DATA | I/O | Data input/output |
| $\overline{MCLR}/V_{PP}$ | Programming Mode | P(1) | Program Mode Select |
| $V_{DD}$ | $V_{DD}$ | P | Power Supply |
| $V_{SS}$ | $V_{SS}$ | P | Ground |

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

**Note 1:** In the PIC16F627A/628A/648A, the programming high voltage is internally generated. To activate the Programming mode, high voltage needs to be applied to $\overline{MCLR}$ input. Since the $\overline{MCLR}$ is used for a level source, this means that $\overline{MCLR}$ does not draw any significant current.

# 2.0 PROGRAM DETAILS

## 2.1 User Program Memory Map

The user memory space extends from 0x0000 to 0x1FFF. 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. In the user program memory space, the PC will increment from 0x0000 to the end of implemented user program memory (see Figure 2-1) and wraps around to 0x0000. Additionally, the high order bit is not affected by the Increment Address command. Thus, in configuration memory, the PC increments from 0x2000 to 0x3FFF and wraps around to 0x2000 (not to 0x0000). The only way to set the PC back to user program memory is to reset the part and re-enter Program/Verify mode as described in **Section 2.4 "Program/Verify Mode"**.

Configuration memory space is entered via the Load Configuration command (see **Section 2.4.3 "Load Data for Configuration Memory"**). Only addresses 0x2000-0x200F of configuration memory space are physically implemented. However, only locations 0x2000 through 0x2007 are available. Other locations are reserved. Locations beyond 0x200F will physically access user memory.

## 2.2 User ID Locations

A user may store identification information (user ID) in four user ID locations. The user ID locations are mapped in [0x2000 : 0x2003]. These locations read out normally even after the code protection is enabled.

**Note 1:** All other locations in PIC® MCU configuration memory are reserved and should not be programmed.

**2:** Only the low order 4 bits of the user ID locations may be included in the device checksum. See **Section 3.10 "Checksum Computation"** for checksum calculation details.

## 2.3 EE Data Memory

The EE Data memory space extends from 0x00 to 0xFF and is separate from both program memory space and RAM space.

Only the lower 128 bytes are implemented in the PIC16F627A/628A devices, while the PIC16F648A implements the full 256 bytes.

Programming the EE Data memory uses the same PC as program memory, though only the lower bits are decoded and used.

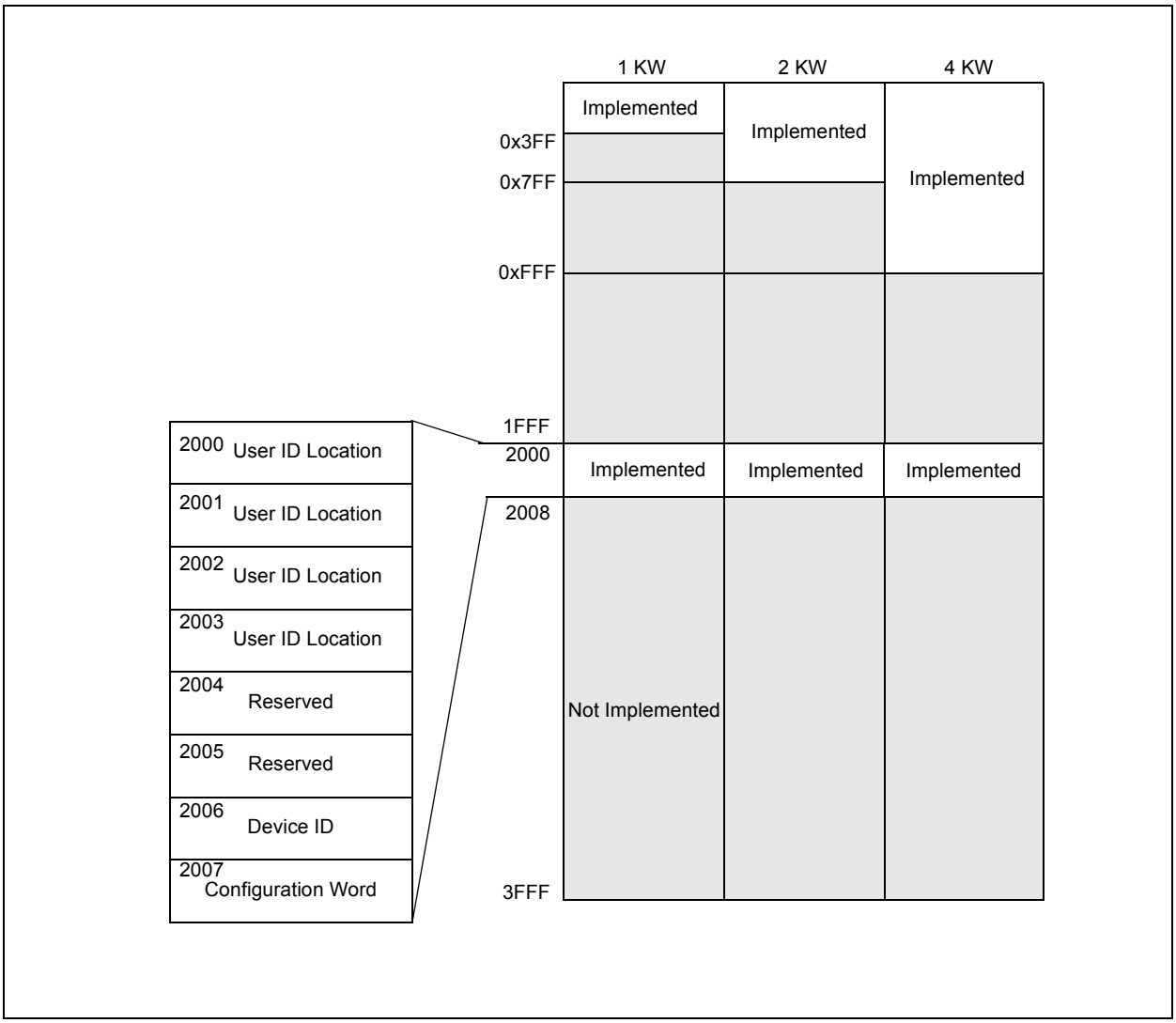
**TABLE 2-1: EE DATA CAPACITY**

| Device | EE Data Memory | PC Bits Decoded |
|---|---|---|
| PIC16F627A/628A | 128 | 7 |
| PIC16F648A | 256 | 8 |

**TABLE 2-2: PROGRAM FLASH**

| Device | Program Flash |
|---|---|
| PIC16F627A | 1K |
| PIC16F628A | 2K |
| PIC16F648A | 4K |

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



## 2.4 Program/Verify Mode

The programming module operates on simple command sequences entered in serial fashion with the data being latched on the falling edge of the clock pulse. The sequences are entered serially, via the clock and data lines, which are Schmitt Trigger inputs in this mode. The general form for all command sequences consists of a 6-bit command and conditionally a 16-bit data word. Both command and data word are clocked LSb first.

The signal on pin DATA is required to have a minimum setup and hold time (see AC/DC specifications), with respect to the falling edge of the clock. Commands that have data associated with them (Read and Load), require a minimum delay of TDLY1 between the command and the data.

The 6-bit command sequences are shown in Table 2-3.

**TABLE 2-3: COMMAND MAPPING FOR PIC16F627A/PIC16F628A/PIC16F648A**

| 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 |
| Increment Address | X | X | 0 | 1 | 1 | 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 |
| Begin Programming Only Cycle | X | 0 | 1 | 0 | 0 | 0 | |
| Bulk Erase Program Memory | X | X | 1 | 0 | 0 | 1 | |
| Bulk Erase Data Memory | X | X | 1 | 0 | 1 | 1 | |

The optional 16-bit data word will either be an input to, or an output from the PIC® microcontroller, depending on the command. Load Data commands will be input, and Read Data commands will be output. The 16-bit data word only contains 14 bits of data to conform to the 14-bit program memory word. The 14 bits are centered within the 16-bit word, padded with a leading and trailing zero.

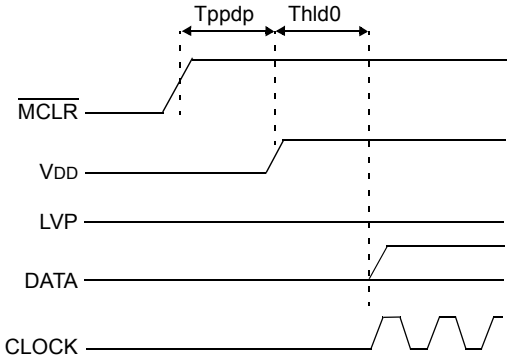
Program/Verify mode may be entered via one of two methods. High voltage Program/Verify is entered by holding CLOCK and DATA pins low while raising $\overline{MCLR}$ first, then VDD as shown in Figure 2-2. Low voltage Program/Verify mode is entered by raising VDD, then $\overline{MCLR}$ and PGM, as shown in Figure 2-3. The PC will be set to '0' upon entering into Program/Verify mode. The PC can be changed by the execution of either an increment PC command, or a Load Configuration command, which sets the PC to 0x2000.

All other logic is held in the Reset state while in Program/Verify mode. This means that all I/O are in the Reset state (high-impedance inputs).

If LVP is not being used for programming and the LVP Configuration bit is set (i.e., LVP feature is enabled), the PGM pin must not be allowed to toggle while programming. The PGM pin is edge sensitive and if an edge is detected during programming, it may cause the PC to reset. If the LVP feature is disabled, the PGM pin will have no effect on programming.

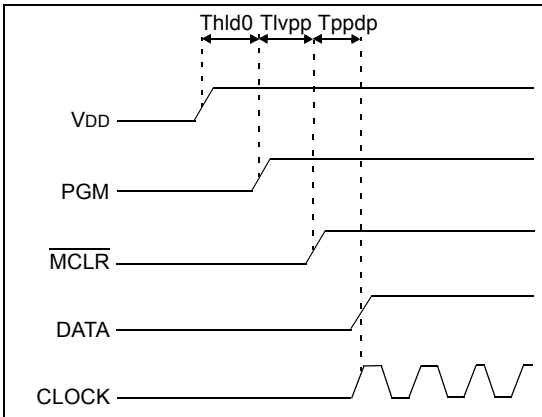
| **Note:** | The LVP feature is enabled by default when the LVP bit of the Configuration Word is set. |
|---|---|

**FIGURE 2-2: ENTERING HIGH VOLTAGE PROGRAM/ VERIFY MODE**




| **Note:** | If the LVP fuse is enabled, PGM should be held low to prevent inadvertent entry into LVP mode. |
|---|---|

**FIGURE 2-3: ENTERING LOW VOLTAGE PROGRAM/ VERIFY MODE**



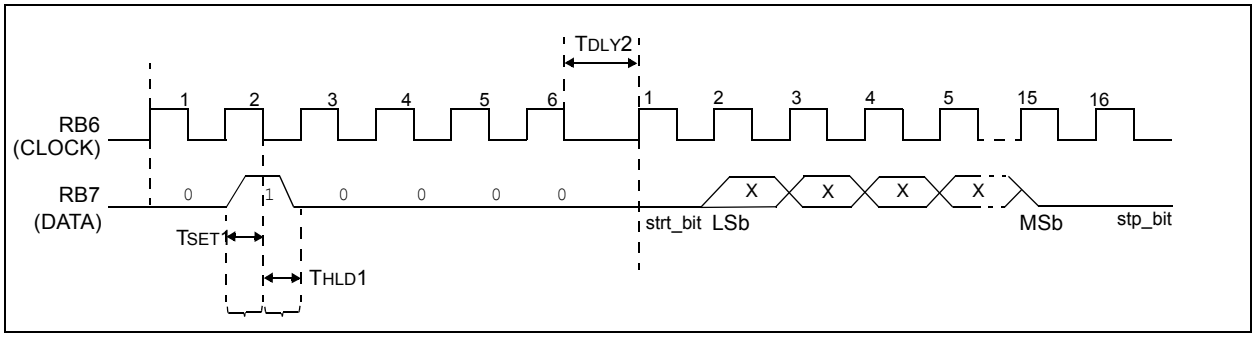

| **Note:** | If the device is in LVP mode, raising VPP to VIHH will override LVP mode. |
|---|---|

### 2.4.1 LOAD DATA FOR PROGRAM MEMORY

Load data for program memory receives a 14-bit word, and readies it to be programmed at the PC location. See Figure 2-4 for timing details.

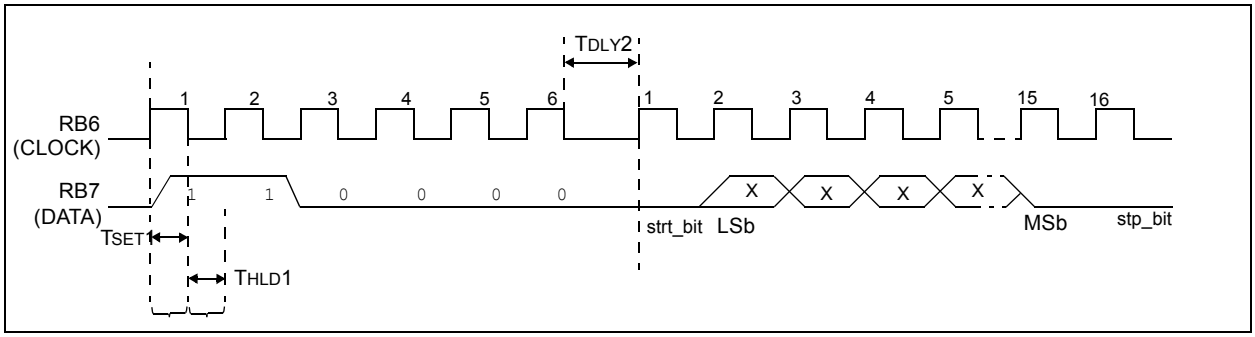
**FIGURE 2-4: LOAD DATA COMMAND FOR PROGRAM MEMORY**




### 2.4.2 LOAD DATA FOR DATA MEMORY

Load data for data memory receives an 8-bit byte and readies it to be programmed into data memory. Though the data byte is only 8-bits wide, all 16 clock cycles are required to allow the programming module to reset properly.

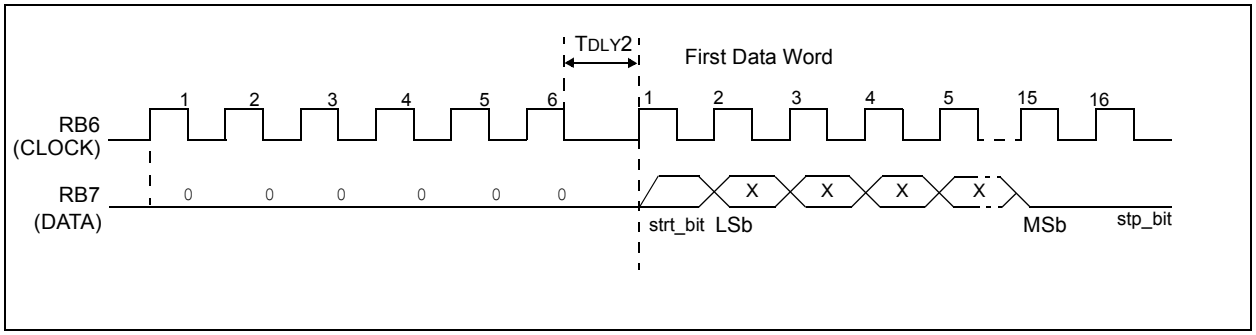
**FIGURE 2-5: LOAD DATA COMMAND FOR DATA MEMORY**



### 2.4.3 LOAD DATA FOR CONFIGURATION MEMORY

The Load Configuration command advances the PC to the start of configuration memory (0x2000-0x200F), and loads the data for the first ID location. Once it is set to the configuration region, only exiting and re-entering Program/Verify mode will reset PC to the user memory space.

**FIGURE 2-6: LOAD CONFIGURATION**

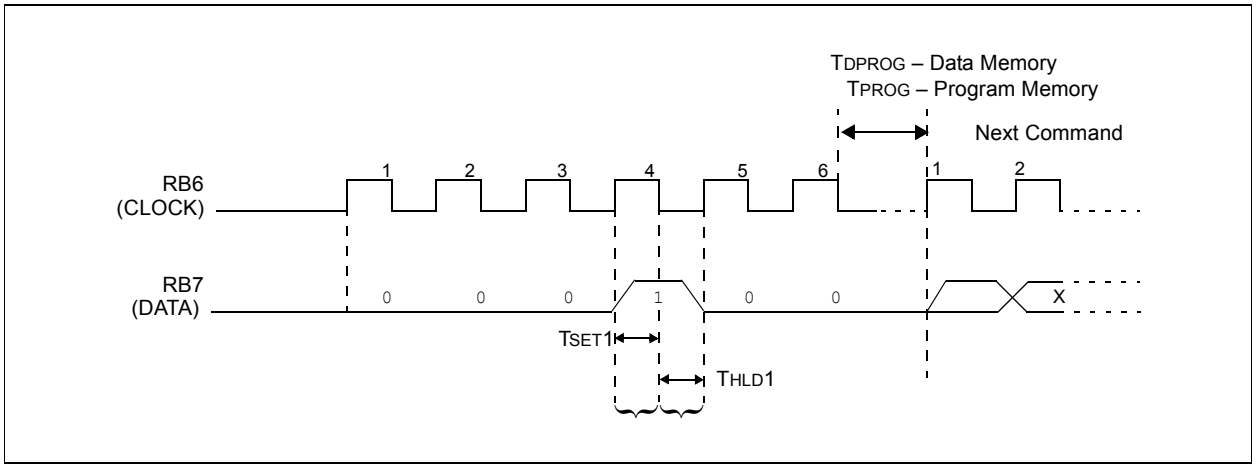



### 2.4.4 BEGIN PROGRAMMING ONLY CYCLE

Begin programming only cycle programs the previously loaded word into the appropriate memory (User Program, Data or Configuration memory). **A Load command must be given before every Programming command**. Programming begins after this command is received and decoded. An internal timing mechanism executes the write. The user must allow for program cycle time before issuing the next command. No “End Programming” command is required.

The device must be Bulk Erased before starting a series of programming only cycles.

**FIGURE 2-7: BEGIN PROGRAMMING ONLY CYCLE**

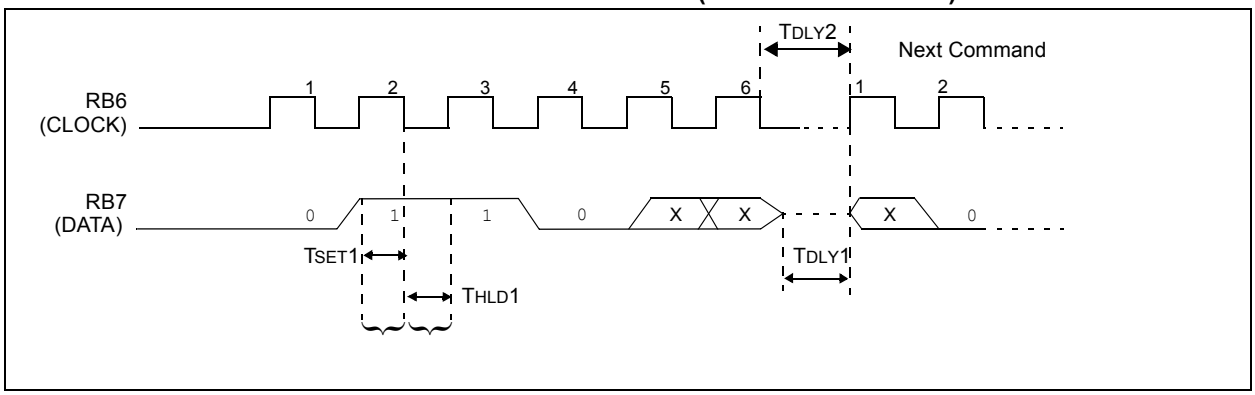


### 2.4.5 INCREMENT ADDRESS

The PC is incremented when this command is received. See Figure 2-8.

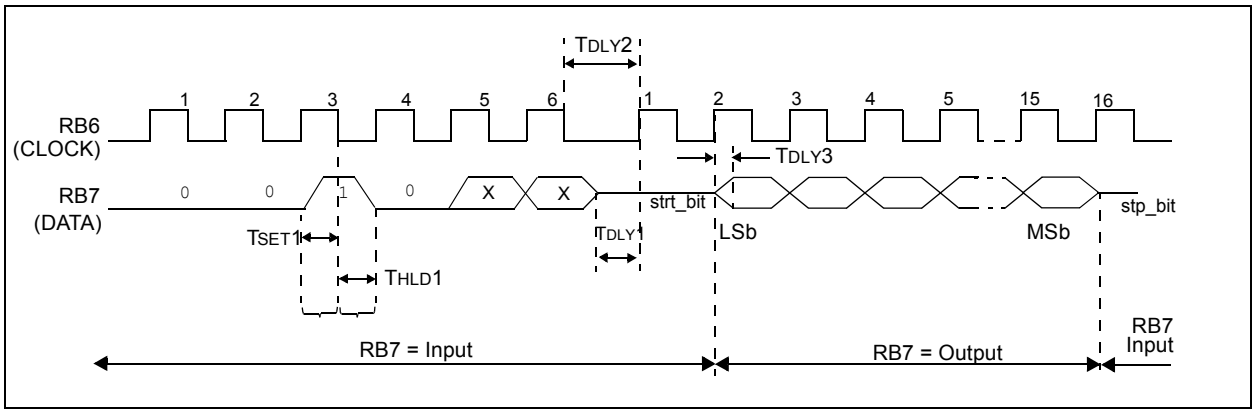
**FIGURE 2-8: INCREMENT ADDRESS COMMAND (PROGRAM/VERIFY)**




### 2.4.6 READ DATA FROM PROGRAM MEMORY

Read data from program memory reads the word addressed by the PC and transmits it on the DATA pin during the data phase of the command. This command will report words from either user or configuration memory, depending on the PC setting. The DATA pin will go into Output mode on the second rising clock edge and revert back to Input mode (high-impedance) after the 16th rising edge.

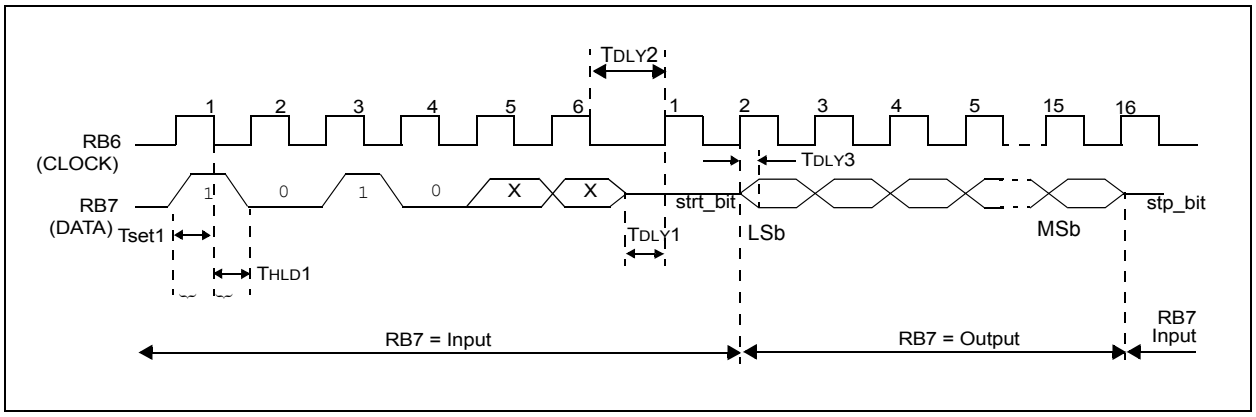
**FIGURE 2-9: READ DATA FROM PROGRAM MEMORY**



### 2.4.7 READ DATA FROM DATA MEMORY

Read data from data memory reads the byte in data memory addressed by the low order bits of PC and transmits it on the DATA pin during the data phase of the command. The DATA pin will go into Output mode on the second rising clock edge and revert back to Input mode (high-impedance) after the 16th rising edge. As only 8 bits are transmitted, the last 8 bits are zero padded.

**FIGURE 2-10: READ DATA FROM DATA MEMORY**



## 3.0 COMMON PROGRAMMING TASKS

These programming commands may be combined in several ways, in order to accomplish different programming goals.

### 3.1 Bulk Erase Program Memory

The program memory can be erased with the Bulk Erase Program Memory command.

| **Note:** | All Bulk Erase operations must take place with VDD between 4.5-5.5V. |
|---|---|

To perform a Bulk Erase of the program memory, the following sequence must be performed:

1. Execute a Load Data for Program Memory with the data word set to all '1's (0x3FFF).
2. Execute a Bulk Erase Program Memory command
3. Wait TERA for the erase cycle to complete.

If the address is pointing to the configuration memory (0x2000-0x200F), then both user ID locations and program memory will be erased.

**FIGURE 3-1: BULK ERASE PROGRAM MEMORY**

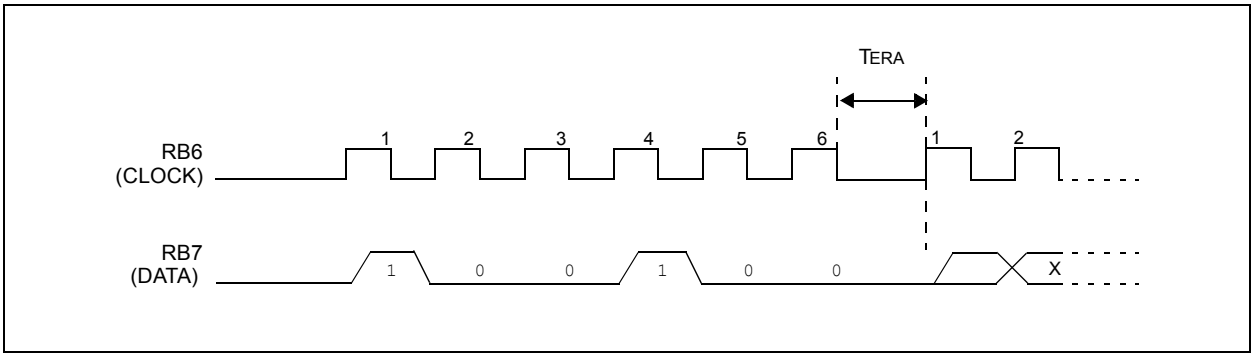



**TABLE 3-1: EFFECTS OF ERASING CODE PROTECTED MEMORY**

| ACTION Serial & Parallel Operation | Initial State | | | Result | | | | |
|---|---|---|---|---|---|---|---|---|
| | $\overline{CP}$ ON=0 OFF=1 | $\overline{CPD}$ ON=0 OFF=1 | PC= Config. Mem. | Program Memory | Data EE Memory | Config. Word | User ID location | Comment |
| Bulk Erase Data Memory | X | OFF | X | Unaffected | Erased | Unaffected | Unaffected | |
| Bulk Erase Data Memory | X | ON | X | Unaffected | Erased | Unaffected | Unaffected | CPD=ON |
| Bulk Erase Program Memory | X | ON | YES | Erased | Erased | Erased | Erased | |
| Bulk Erase Program Memory | X | OFF | YES | Erased | Unaffected | Erased | Erased | |
| Bulk Erase Program Memory | X | ON | NO | Erased | Erased | Erased | Unaffected | |
| Bulk Erase Program Memory | X | OFF | NO | Erased | Unaffected | Erased | Unaffected | |

## 3.2 Bulk Erase Data Memory

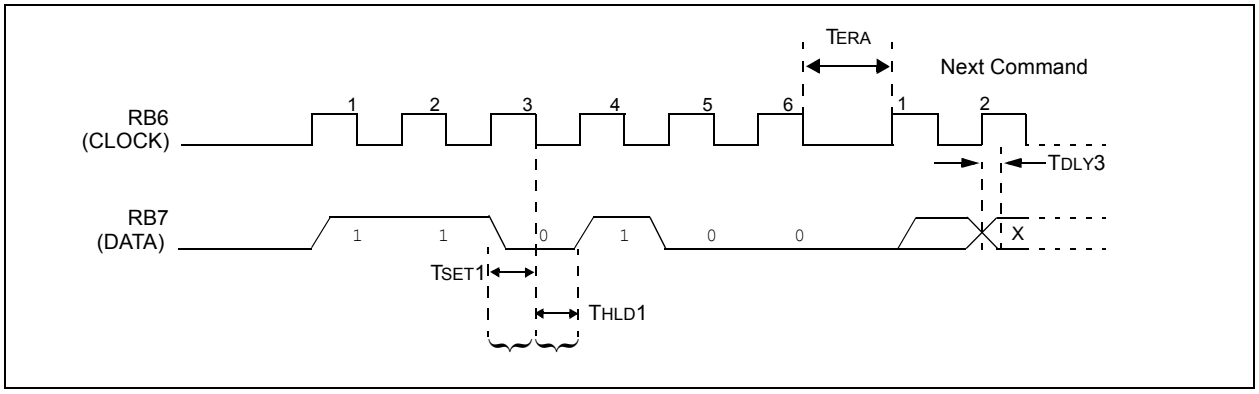
The data memory can be erased with the Bulk Erase Data memory command.

To perform a Bulk Erase of the data memory, the following sequence must be performed:

1. Execute a Bulk Erase Data memory command.
2. Wait TERA for the erase cycle to complete.

| **Note:** | All Bulk Erase operations must take place with VDD between 4.5-5.5V. |
|---|---|

**FIGURE 3-2: BULK ERASE DATA MEMORY COMMAND**



## 3.3 Programming Program Memory

**FIGURE 3-3: PROGRAM FLOWCHART – PIC16F627A/628A/648A PROGRAM MEMORY**

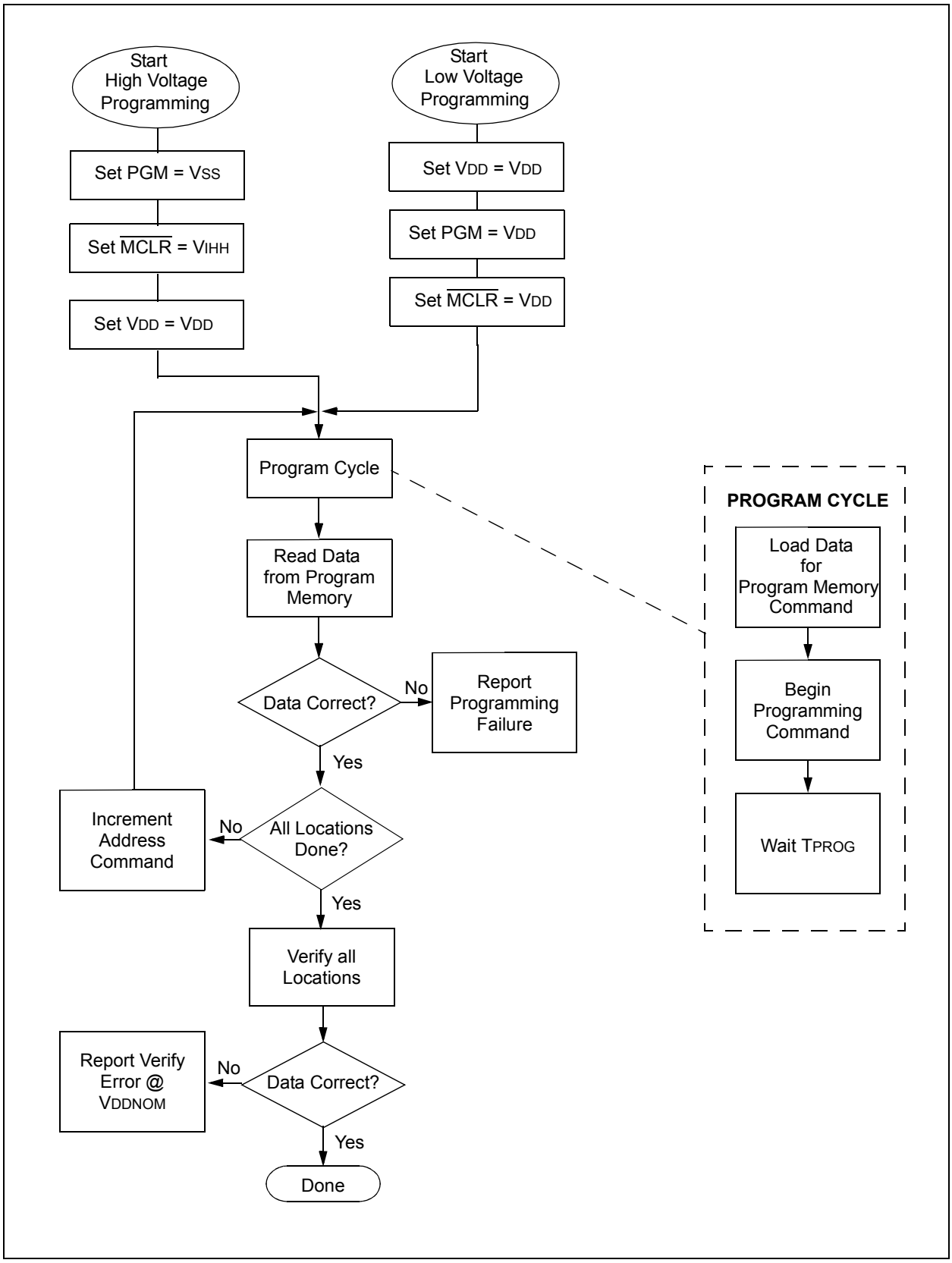
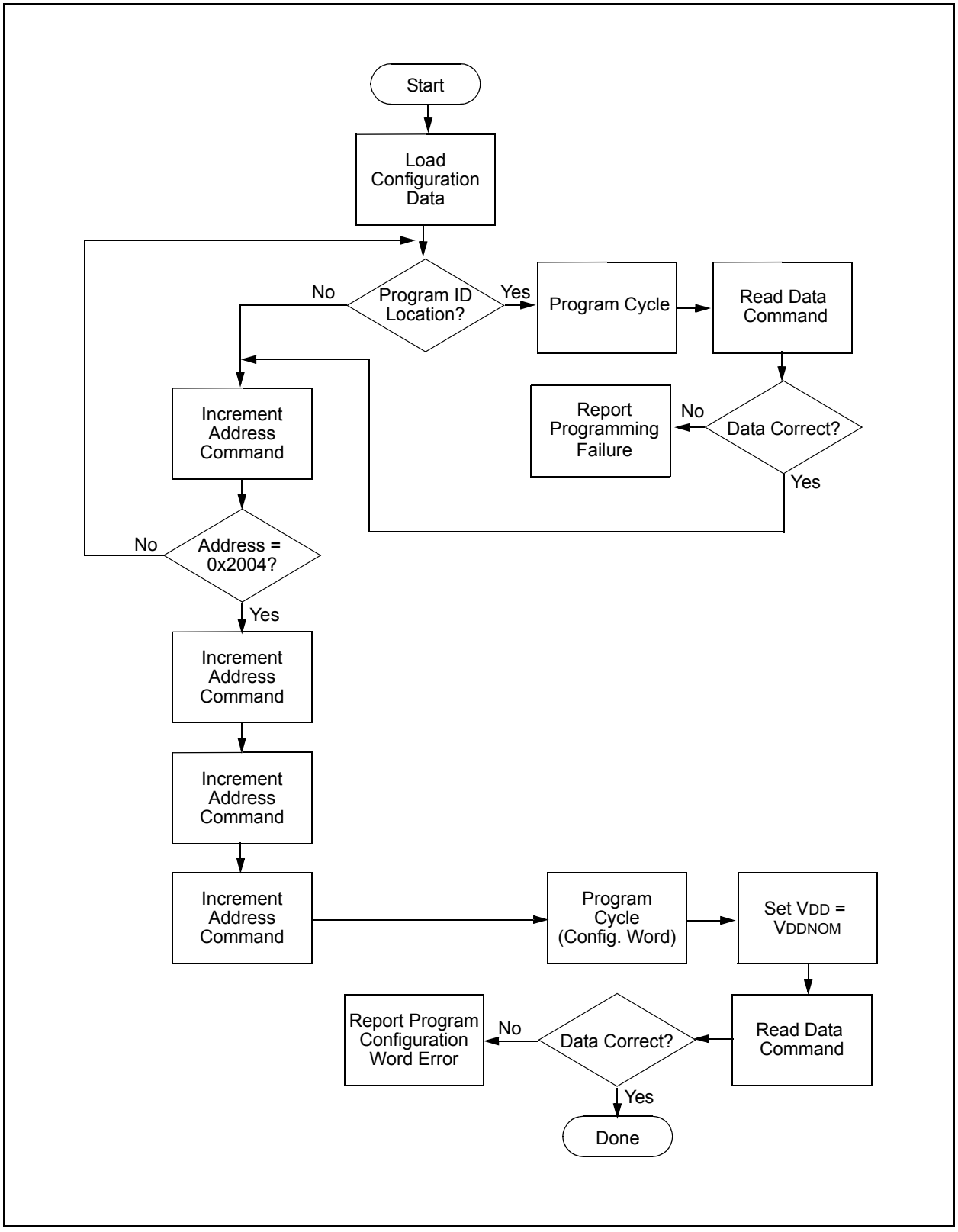


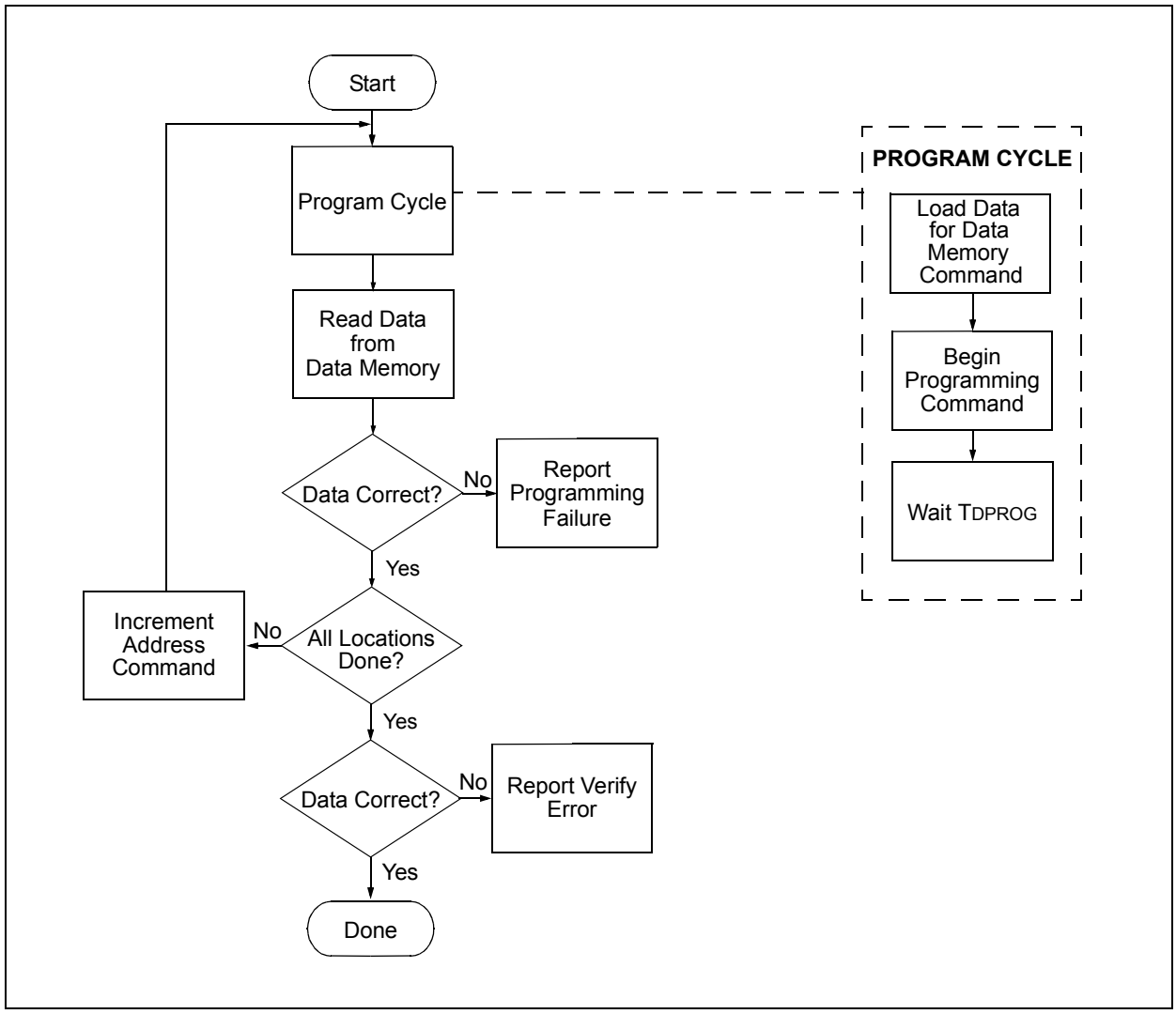
**FIGURE 3-4: PROGRAM FLOWCHART – PIC16F627A/628A/648A CONFIGURATION MEMORY**



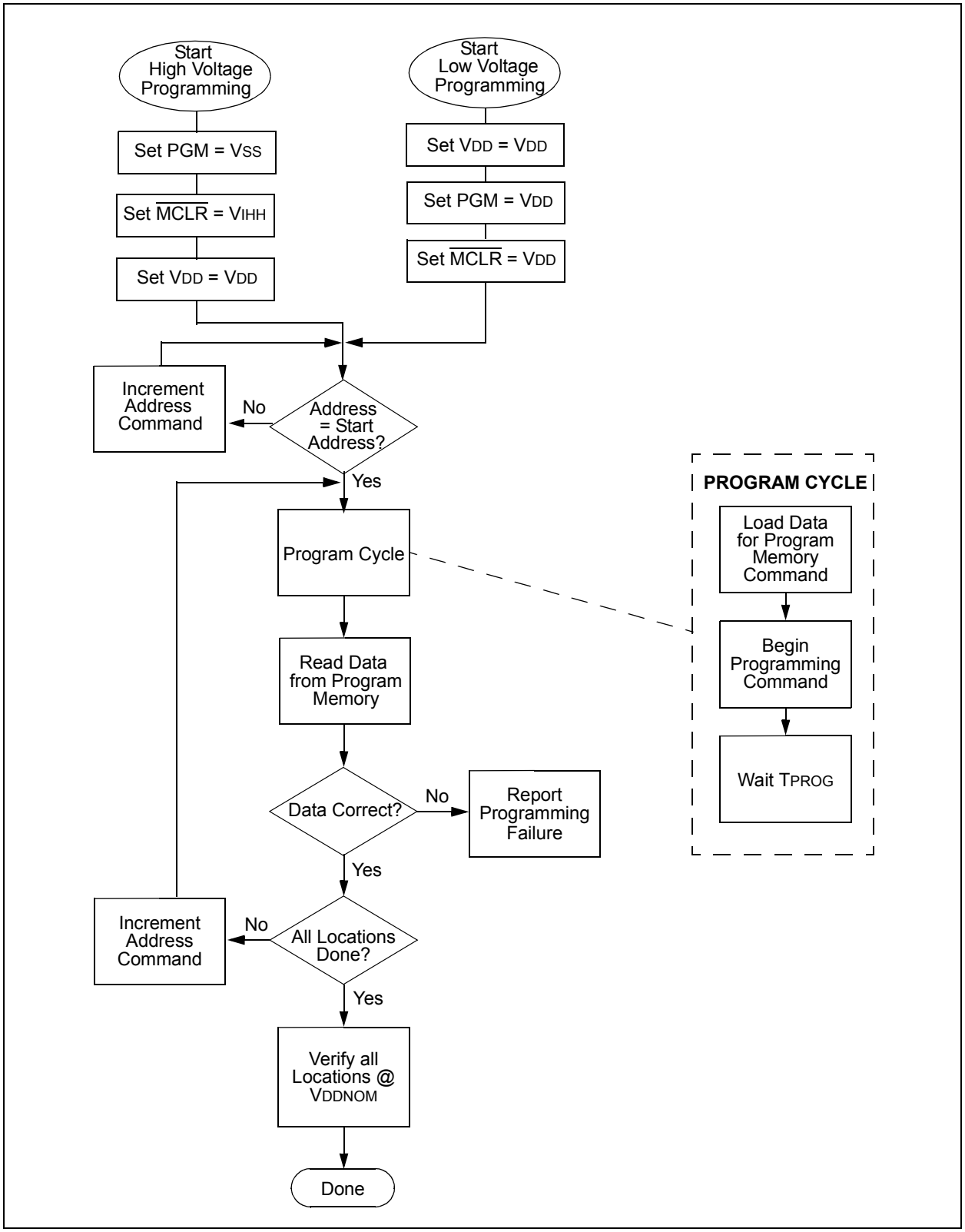
## 3.4 Program Data Memory

**FIGURE 3-5: PROGRAM FLOWCHART – PIC16F627A/628A/648A DATA MEMORY**



## 3.5 Programming Range of Program Memory

**FIGURE 3-6: PROGRAM FLOWCHART – PIC16F627A/628A/648A PROGRAM MEMORY**



## 3.6 Configuration Word

The PIC16F627A/628A/648A has several Configuration bits. These bits can be set (reads '0') or left unchanged (reads '1'), to select various device configurations.

## 3.7 Device ID Word

The device ID word for the PIC16F627A/628A/648A is hard coded at 2006h.

**TABLE 3-2: DEVICE ID VALUES**

| Device | Device ID Value | |
|---|---|---|
| | Dev | Rev |
| PIC16F627A | 01 0000 010 | x xxxx |
| PIC16F628A | 01 0000 011 | x xxxx |
| PIC16F648A | 01 0001 000 | x xxxx |

**REGISTER 3-1: CONFIGURATION WORD FOR PIC16F627A/PIC16F628A/PIC16F648A (ADDRESS: 2007h)**

| R/P-1 | U-1 | U-1 | U-1 | U-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 | R/P-1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\overline{CP}$ | — | — | — | — | $\overline{CPD}$ | LVP | BOREN | MCLRE | FOSC2 | $\overline{PWRTE}$ | WDTE | FOSC1 | FOSC0 |
| bit 13 | | | | | | | | | | | | | bit 0 |

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

bit 13 **$\overline{CP}$**: FLASH Program Memory Code Protection bit
(PIC16F648A)
1 = Code protection off
0 = 0000h to 0FFFh code-protected
(PIC16F628A)
1 = Code protection off
0 = 0000h to 07FFh code-protected
(PIC16F627A)
1 = Code protection off
0 = 0000h to 03FFh code-protected

bit 12-9 **Unimplemented**: Read as '1'

bit 8 **$\overline{CPD}$**: Data Code Protection bit(2)
1 = Data memory code protection off
0 = Data memory code-protected

bit 7 **LVP**: Low Voltage Programming Enable bit
1 = RB4/PGM pin has PGM function, low-voltage programming enabled
0 = RB4/PGM is digital I/O, HV on MCLR must be used for programming

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

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

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

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

bit 4, 1-0 **FOSC<2:0>**: Oscillator Selection bits(3)
111 = RC oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, Resistor & Capacitor on RA7/OSC1/CLKIN
110 = RC oscillator: I/O function on RA6/OSC2/CLKOUT pin, Resistor & Capacitor on RA7/OSC1/CLKIN
101 = INTOSC internal oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN
100 = INTOSC internal oscillator: I/O function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN
011 = EXTCLK: I/O function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN
010 = HS oscillator: High speed crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN
001 = XT oscillator: Crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN
000 = LP oscillator: Low power crystal on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN

**Note 1:** Enabling Brown-out Reset does not automatically enable the Power-up Timer (PWRT).
**2:** Only a Bulk Erase will reset the Configuration Word, including the $\overline{CP}$ bits.
**3:** While $\overline{MCLR}$ is asserted in INTOSC mode, the internal clock oscillator is disabled.

## 3.8 Embedding Configuration Word and ID Information in the 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 PIC16F627A/628A/648A, the EEPROM data memory should also be embedded in the hex file (see **Section 3.9 "Embedding Data EEPROM Contents in Hex File"**).

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

## 3.9 Embedding Data EEPROM Contents in 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 fuse information.

The 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.

## 3.10 Checksum Computation

### 3.10.1 CHECKSUM

Checksum is calculated by reading the contents of the PIC16F627A/628A/648A memory locations and adding up the opcodes up to the maximum user addressable location (e.g., 0x7FF for the PIC16F628A). 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 PIC16F627A/628A/648A devices is shown in Table 3-3.

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 is the checksum.

The following table describes how to calculate the checksum for each device.

**Note:** 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.

**TABLE 3-3: CHECKSUM COMPUTATION**

| Device | Code Protect | Checksum* | Blank Value | 0x25E6 at 0 and Max Address |
|---|---|---|---|---|
| PIC16F627A | OFF | SUM[0x0000:0x03FF] + CFGW & 0x21FF | 1DFF | E9CD |
| | ON | CFGW & 0x21FF + SUM_ID | 1FFE | EBCC |
| PIC16F628A | OFF | SUM[0x0000:0x7FF] + CFGW & 0x21FF | 19FF | E5CD |
| | ON | CFGW & 0x21FF + SUM_ID | 1BFE | E7CC |
| PIC16F648A | OFF | SUM[0x0000:0x0FFF] + CFGW & 0x21FF | 11FF | DDCD |
| | ON | CFGW & 0x21FF + SUM_ID | 13FE | DFCC |

**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

## 4.0 PROGRAM/VERIFY MODE ELECTRICAL CHARACTERISTICS

**TABLE 4-1: AC/DC CHARACTERISTICS TIMING REQUIREMENTS FOR PROGRAM/VERIFY MODE**

| AC/DC Characteristics | Standard Operating Conditions (unless otherwise stated)<br>Operating Temperature: 0°C ≤ TA ≤ +70°C<br>Operating Voltage: 4.5V ≤ VDD ≤ 5.5V | | | | | |
|---|---|---|---|---|---|---|
| **Characteristics** | **Sym** | **Min** | **Typ** | **Max** | **Units** | **Conditions/ Comments** |
| **General** | | | | | | |
| VDD level for word operations, program memory | VDD | 2.0 | — | 5.5 | V | |
| VDD level for word operations, data memory | VDD | 2.0 | — | 5.5 | V | |
| VDD level for Bulk Erase operations, program and data memory | VDD | 4.5 | — | 5.5 | V | |
| High voltage on $\overline{MCLR}$ | VIHH | 10.0 | — | 13.5 | V | |
| $\overline{MCLR}$ rise time (VSS to VIHH) for Programming mode entry | TVHHR | — | — | 1.0 | μs | |
| Hold time after $\overline{MCLR}$↑ | TPPDP | 5 | — | — | μs | |
| Hold time after LVP↑ | TLVPP | 5 | — | — | μs | |
| (CLOCK, DATA) input high level | VIH1 | 0.8 VDD | — | — | V | Schmitt Trigger input |
| (CLOCK, DATA) input low level | VIL1 | — | — | 0.2 VDD | V | Schmitt Trigger input |
| CLOCK, DATA setup time before $\overline{MCLR}$↑ | TSET0 | 100 | — | — | ns | |
| Hold time after VDD↑ | THLD0 | 5 | — | — | μs | |
| Serial Program/Verify | | | | | | |
| 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 | |
| Delay between clock↓ to clock↑ of next command or data | TDLY2 | 1.0 | — | — | μs | |
| Clock↑ to data out valid (during read data) | TDLY3 | — | — | 80 | ns | |
| Programming cycle time | TPROG | — | — | 4 | ms | |
| Data EEPROM Programming cycle time | TDPROG | — | — | 6 | ms | |
| Bulk Erase cycle time | TERA | — | — | 6 | ms | |

**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

**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 - Fuzhou**
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

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

**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 - Shunde**
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

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

**China - Xian**
Tel: 86-29-8833-7250
Fax: 86-29-8833-7256

## ASIA/PACIFIC

**India - Bangalore**
Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

**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 - Gumi**
Tel: 82-54-473-4301
Fax: 82-54-473-4302

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

**Malaysia - Penang**
Tel: 60-4-646-8870
Fax: 60-4-646-5086

**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: 886-3-572-9526
Fax: 886-3-572-6459

**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

12/08/06