

## EEPROM Memory Programming Specification

This document includes the programming specifications for the following devices:

- PIC16F83
- PIC16CR83
- PIC16F84
- PIC16CR84
- PIC16F84A

### 1.0 PROGRAMMING THE PIC16F8X

The PIC16F8X is programmed using a serial method. The serial mode will allow the PIC16F8X to be programmed while in the users system. This allows for increased design flexibility. This programming specification applies to PIC16F8X devices in all packages.

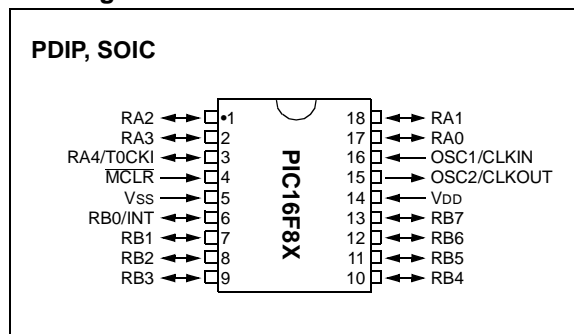
#### 1.1 Hardware Requirements

The PIC16F8X requires one programmable power supply for VDD (4.5V to 5.5V) and a VPP of 12V to 14V. Both supplies should have a minimum resolution of 0.25V.

#### 1.2 Programming Mode

The programming mode for the PIC16F8X allows programming of user program memory, data memory, special locations used for ID, and the configuration word.

Pin Diagram



### PIN DESCRIPTIONS (DURING PROGRAMMING): PIC16F8X

Pin Name	During Programming		
	Function	Pin Type	Pin Description
RB6	CLOCK	I	Clock input
RB7	DATA	I/O	Data input/output
MCLR	VTEST MODE	P*	Program Mode Select
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground

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

\*In the PIC16F8X, the programming high voltage is internally generated. To activate the programming mode, high voltage needs to be applied to MCLR input. Since the MCLR is used for a level source, this means that 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), of which 1K (0x0000 - 0x03FF) is physically implemented. In actual implementation the on-chip user program memory is accessed by the lower 10-bits of the PC, with the upper 3-bits of the PC ignored. Therefore if the PC is greater than 0x3FF, it will wrap around and address a location within the physically implemented memory. (See Figure 2-1).

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 0x1FFF and wrap to 0x000 or 0x2000 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 reenter program/verify mode as described in Section 2.3.

In the configuration memory space, 0x2000-0x200F are physically implemented. However, only locations 0x2000 through 0x2007 are available. Other locations are reserved. Locations beyond 0x200F will physically access user memory. (See Figure 2-1).

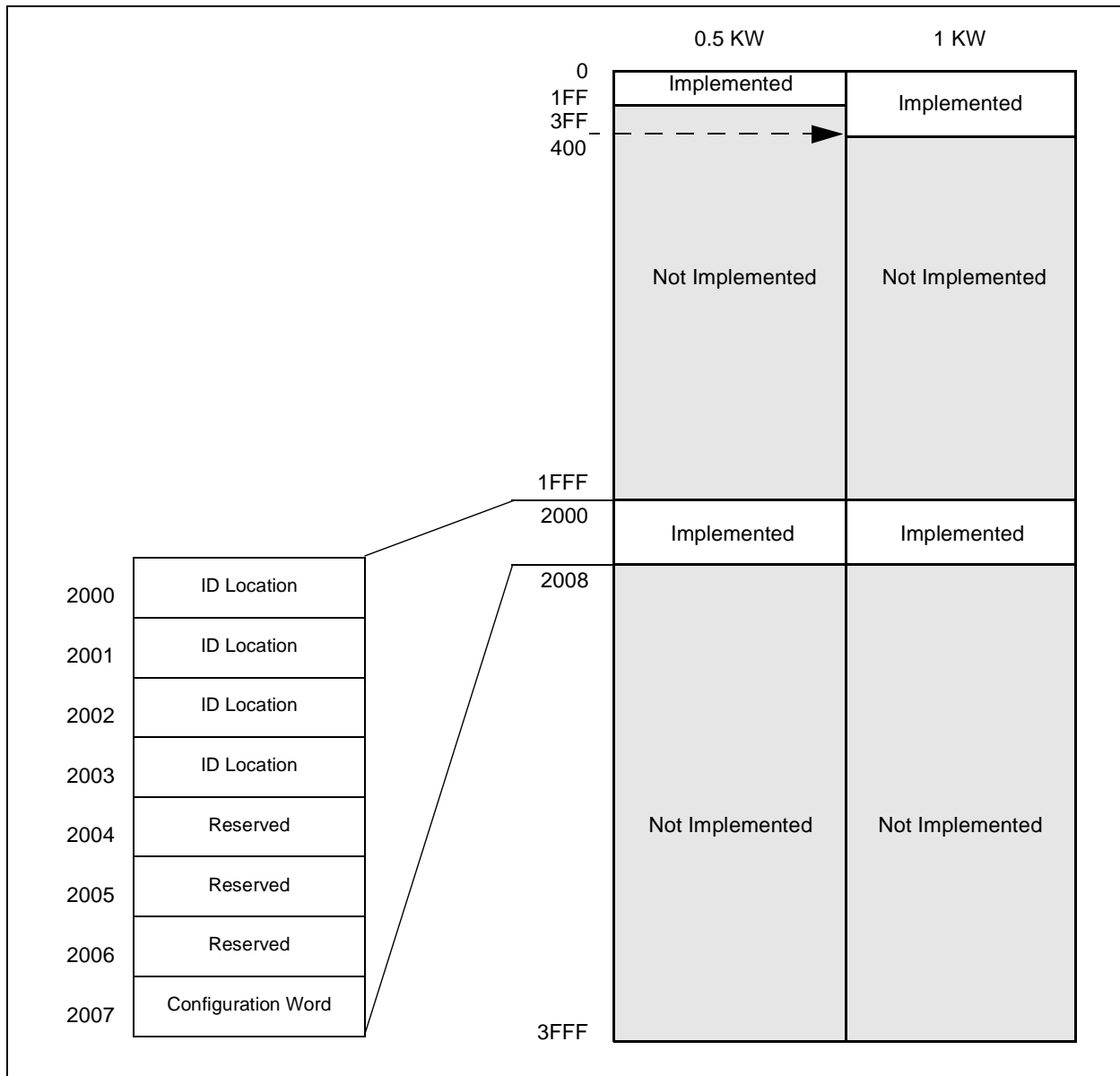
### 2.2 ID Locations

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 4-2.

To understand the scrambling mechanism after code protection, refer to Section 4.0.

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



## 2.3 Program/Verify Mode

The program/verify mode is entered by holding pins RB6 and RB7 low while raising  $\overline{\text{MCLR}}$  pin from  $V_{IL}$  to  $V_{IH}$  (high voltage). Once in this 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 that is 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  $\overline{\text{MCLR}}$  is between  $V_{IL}$  and  $V_{IH}$ .

The sequence that enters the device into the programming/verify mode places all other logic into the reset state (the  $\overline{\text{MCLR}}$  pin was initially at  $V_{IL}$ ). This means that all I/O are in the reset state (High impedance inputs).

The normal sequence for programming is to use the load data command to set a value to be written at the selected address. Issue the begin programming command followed by read data command to verify, and then increment the address.

### 2.3.1 SERIAL PROGRAM/VERIFY OPERATION

The RB6 pin is used as a clock input pin, and the RB7 pin is used for entering command bits and data input/output 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 pin RB7 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) are specified to have a minimum delay of 1  $\mu\text{s}$  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. Data is also input and output LSB first.

Therefore, during a read operation the LSB will be transmitted onto pin 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  $\mu\text{s}$  delay is also specified between consecutive commands.

All commands are transmitted LSB first. Data words are also transmitted LSB first. The data is transmitted on the rising edge and latched on the falling edge of the clock. To allow for decoding of commands and reversal of data pin configuration, a time separation of at least 1  $\mu\text{s}$  is required between a command and a data word (or another command).

The commands that are available are:

#### 2.3.1.1 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/verify test mode by taking  $\overline{\text{MCLR}}$  low ( $V_{IL}$ ).

## 2.3.1.2 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 command is shown in Figure 5-1.

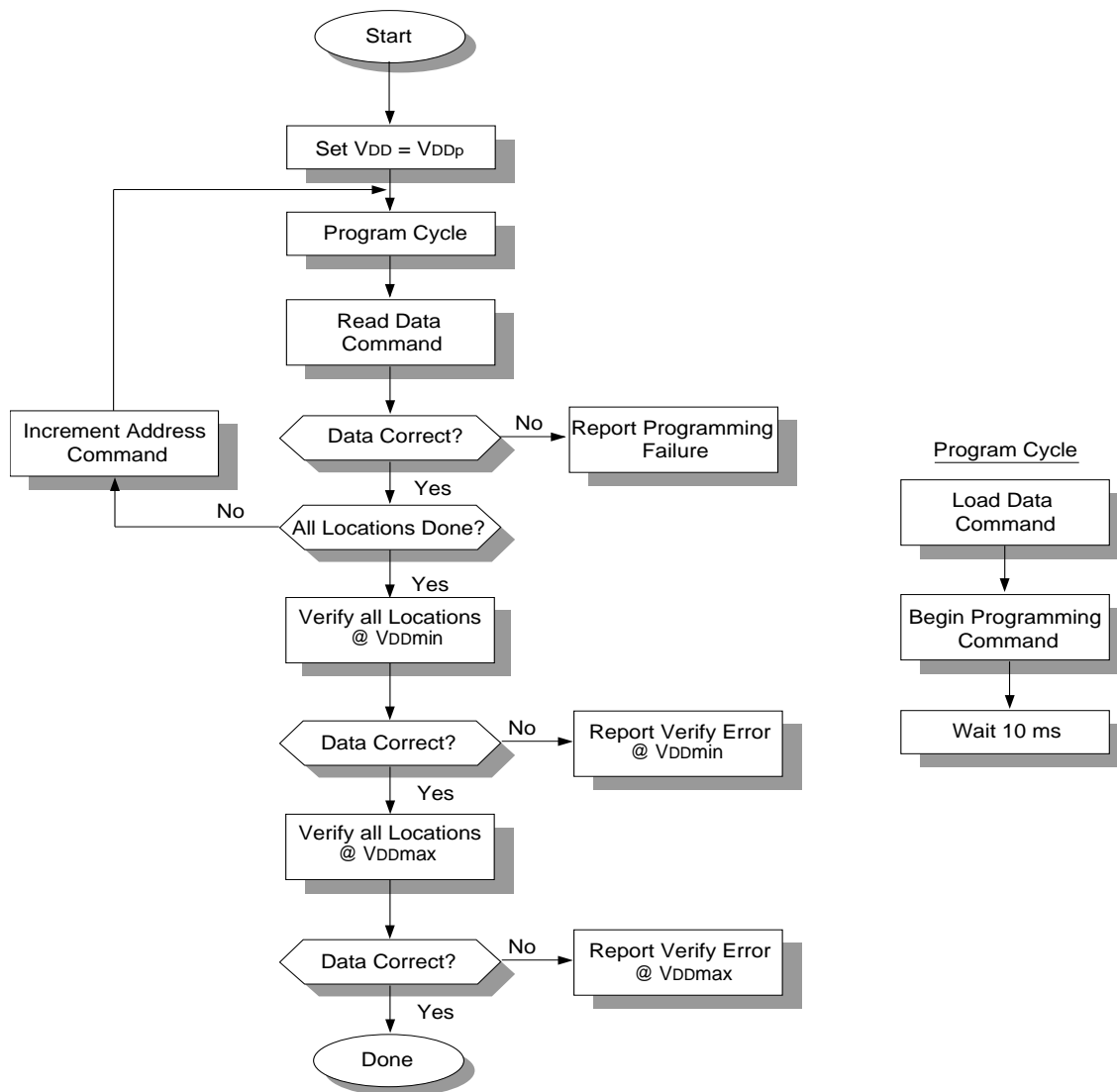
**TABLE 2-1: COMMAND MAPPING FOR PIC16F83/CR83/F84/CR84**

Command	Mapping (MSB ... LSB)						Data
Load Configuration	0	0	0	0	0	0	0, data (14), 0
Load Data for Program Memory	0	0	0	0	1	0	0, data (14), 0
Read Data from Program Memory	0	0	0	1	0	0	0, data (14), 0
Increment Address	0	0	0	1	1	0	
Begin Programming	0	0	1	0	0	0	
Load Data for Data Memory	0	0	0	0	1	1	0, data (14), 0
Read Data from Data Memory	0	0	0	1	0	1	0, data (14), 0
Bulk Erase Program Memory	0	0	1	0	0	1	
Bulk Erase Data Memory	0	0	1	0	1	1	

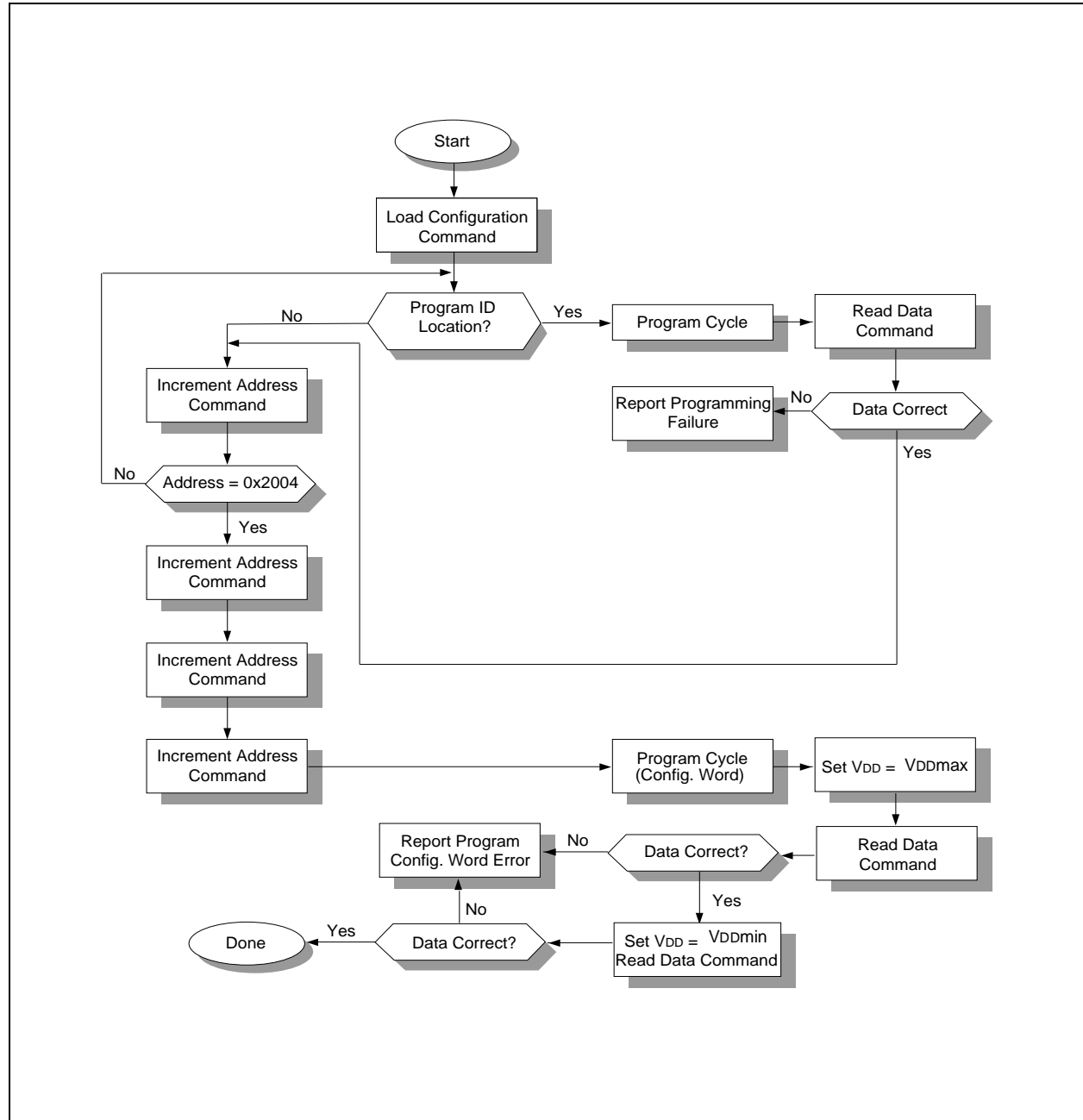
**TABLE 2-2: COMMAND MAPPING FOR PIC16F84A**

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
Read Data from Program Memory	X	X	0	1	0	0	0, data (14), 0
Increment Address	X	X	0	1	1	0	
Begin Erase Programming Cycle	0	0	1	0	0	0	
Begin Programming Only Cycle	0	1	1	0	0	0	
Load Data for Data Memory	X	X	0	0	1	1	0, data (14), 0
Read Data from Data Memory	X	X	0	1	0	1	0, data (14), 0
Bulk Erase Program Memory	X	X	1	0	0	1	
Bulk Erase Data Memory	X	X	1	0	1	1	

FIGURE 2-2: PROGRAM FLOW CHART - PIC16F8X PROGRAM MEMORY



**FIGURE 2-3: PROGRAM FLOW CHART - PIC16F8X CONFIGURATION MEMORY**



## 2.3.1.3 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 64 words. Only the lower 8-bits of the PC are decoded by the data memory, and therefore if the PC is greater than 0x3F, it will wrap around and address a location within the physically implemented memory.

## 2.3.1.4 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 5-2.

## 2.3.1.5 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.

## 2.3.1.6 INCREMENT ADDRESS

The PC is incremented when this command is received. A timing diagram of this command is shown in Figure 5-3.

## 2.3.1.7 BEGIN ERASE/PROGRAM CYCLE

**A load command must be given before every begin programming command.** Programming of the appropriate memory (test program memory, user program memory or data memory) will begin after this command is received and decoded. An internal timing mechanism executes an erase before write. The user must allow for both erase and programming cycle times for programming to complete. No “end programming” command is required.

## 2.3.1.8 BEGIN PROGRAMMING

**A load command must be given before every begin programming command.** Programming of the appropriate memory (test program memory, user program 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.

This command is similar to the ERASE/PROGRAM CYCLE command, except that a word erase is not done. It is recommended that a bulk erase be performed before starting a series of programming only cycles.

## 2.3.1.9 BULK ERASE PROGRAM MEMORY

After this command is performed, the next program command will erase the entire program memory.

To perform a bulk erase of the program memory, the following sequence must be performed.

1. Do a “Load Data All 1’s” command.
2. Do a “Bulk Erase User Memory” command.
3. Do a “Begin Programming” command.
4. Wait 10 ms to complete bulk erase.

If the address is pointing to the test program memory (0x2000 - 0x200F), then both the user memory and the test memory will be erased. The configuration word will not be erased, even if the address is pointing to location 0x2007.

<b>Note:</b> If the device is code-protected (PIC16F84A), the BULK ERASE command will not work.
---

## 2.3.1.10 BULK ERASE DATA MEMORY

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

1. Do a “Load Data All 1’s” command.
2. Do a “Bulk Erase Data Memory” command.
3. Do a “Begin Programming” command.
4. Wait 10 ms to complete bulk erase.

<b>Note:</b> All BULK ERASE operations must take place at 4.5 to 5.5 VDD range.
---



## 2.4 Programming Algorithm Requires Variable VDD

The PIC16F8X uses an intelligent algorithm. The algorithm calls for program verification at VDDmin. as well as VDDmax. Verification at VDDmin. guarantees good “erase margin”. Verification at VDDmax guarantees good “program margin”.

The actual programming must be done with VDD in the VDDP range (See Table 5-1).

VDDP = VCC range required during programming.

VDDmin. = minimum operating VDD spec for the part.

VDDmax.= maximum operating VDD spec for the part.

Programmers must verify the PIC16F8X at its specified VDD max. and VDDmin levels. Since Microchip may introduce future versions of the PIC16F8X with a broader VDD range, it is best that these levels are user selectable (defaults are ok).

<p><b>Note:</b> Any programmer not meeting these requirements may only be classified as “prototype” or “development” programmer but not a “production” quality programmer.</p>
--

# PIC16F8X

## 3.0 CONFIGURATION WORD

The PIC16F8X has five configuration bits. These bits can be set (reads '0') or left unchanged (reads '1') to select various device configurations.

## 3.1 Device ID Word

The device ID word for the PIC16F84A is located at 2006h.

TABLE 3-1:

Device	Device ID Value	
	Dev	Rev
PIC16F84A	00 0101 010	0 0000

FIGURE 3-1: CONFIGURATION WORD BIT MAP

Bit Number:	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIC16F83/F84/F84A	CP	CP	CP	CP	CP	CP	CP	CP	CP	CP	PWRTE	WDTE	FOSC1	FOSC0
PIC16CR83/CR84	CP	CP	CP	CP	CP	CP	DP	CP	CP	CP	PWRTE	WDTE	FOSC1	FOSC0
bit 4-13: <b>CP</b> , Code Protection Configuration Bits 1 = code protection off 0 = code protection on														
bit 7: <b>PIC16CR83/CR84 only</b> <b>DP</b> , Data Memory Code Protection Bit 1 = code protection off 0 = data memory is code protected														
bit 3: <b>PWRTE</b> , Power Up Timer Enable Configuration Bit 1 = Power up timer disabled 0 = Power up timer enabled														
bit 2: <b>WDTE</b> , WDT Enable Configuration Bits 1 = WDT enabled 0 = WDT disabled														
bit 1-0 <b>FOSC&lt;1:0&gt;</b> , Oscillator Selection Configuration Bits 11: RC oscillator 10: HS oscillator 01: XT oscillator 00: LP oscillator														

## 4.0 CODE PROTECTION

For PIC16F8X devices, once code protection is enabled, all program memory locations read all 0's. The ID locations and the configuration word read out in an unscrambled fashion. Further programming is disabled for the entire program memory as well as data memory. It is possible to program the ID locations and the configuration word.

### 4.1 Disabling Code-Protection

It is recommended that the following procedure be performed before any other programming is attempted. It is also possible to turn code protection off (code protect bit = 1) using this procedure; however, ***all data within the program memory and the data memory will be erased when this procedure is executed, and thus, the security of the data or code is not compromised.***

Procedure to disable code protect:

- Execute load configuration (with a '1' in bit 4, code protect).
- Increment to configuration word location (0x2007)
- Execute command (000001)
- Execute command (000111)
- Execute 'Begin Programming' (001000)
- Wait 10 ms
- Execute command (000001)
- Execute command (000111)

### 4.2 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 PIC16F8X, the EEPROM data memory should also be embedded in the hex file (see Section 5.1).

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

**TABLE 4-1: CONFIGURATION WORD**

#### PIC16F83

To code protect: 000000000XXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
All memory	Read All 0's, Write Disabled	Read Unscrambled, Write Enabled
ID Locations [0x2000 : 0x2003]	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled

#### PIC16CR83

To code protect: 000000000XXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Unscrambled	Read Unscrambled
All memory	Read All 0's for Program Memory, Read All 1's for Data Memory - Write Disabled	Read Unscrambled, Data Memory - Write Enabled
ID Locations [0x2000 : 0x2003]	Read Unscrambled	Read Unscrambled

# PIC16F8X

---

## PIC16CR84

To code protect: 000000000XXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Unscrambled	Read Unscrambled
All memory	Read All 0's for Program Memory, Read All 1's for Data Memory - Write Disabled	Read Unscrambled, Data Memory - Write Enabled
ID Locations [0x2000 : 0x2003]	Read Unscrambled	Read Unscrambled

## PIC16F84

To code protect: 000000000XXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
All memory	Read All 0's, Write Disabled	Read Unscrambled, Write Enabled
ID Locations [0x2000 : 0x2003]	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled

## PIC16F84A

To code protect: 000000000XXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
All memory	Read All 0's, Write Disabled	Read Unscrambled, Write Enabled
ID Locations [0x2000 : 0x2003]	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled

Legend: X = Don't care

## 4.3 CHECKSUM COMPUTATION

### 4.3.1 CHECKSUM

Checksum is calculated by reading the contents of the PIC16F8X memory locations and adding up the opcodes up to the maximum user addressable location, e.g., 0x1FF for the PIC16F8X. 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 PIC16F8X devices is shown in Table 4-2.

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 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 4-2: CHECKSUM COMPUTATION**

Device	Code Protect	Checksum*	Blank Value	0x25E6 at 0 and max address
PIC16F83	OFF ON	SUM[0x000:0x1FF] + CFGW & 0x3FFF CFGW & 0x3FFF + SUM_ID	0x3DFF 0x3E0E	0x09CD 0x09DC
PIC16CR83	OFF ON	SUM[0x000:0x1FF] + CFGW & 0x3FFF CFGW & 0x3FFF + SUM_ID	0x3DFF 0x3E0E	0x09CD 0x09DC
PIC16F84	OFF ON	SUM[0x000:0x3FF] + CFGW & 0x3FFF CFGW & 0x3FFF + SUM_ID	0x3BFF 0x3C0E	0x07CD 0x07DC
PIC16CR84	OFF ON	SUM[0x000:0x3FF] + CFGW & 0x3FFF CFGW & 0x3FFF + SUM_ID	0x3BFF 0x3C0E	0x07CD 0x07DC
PIC16F84A	OFF ON	SUM[0x000:0x3FF] + CFGW & 0x3FFF CFGW & 0x3FFF + SUM_ID	0x3BFF 0x3C0E	0x07CD 0x07DC

Legend: CFGW = Configuration Word

SUM[a:b] = [Sum of locations a to b inclusive]

\*Checksum = [Sum of all the individual expressions] **MODULO** [0xFFFF]

+ = Addition

& = Bitwise AND

# PIC16F8X

## 5.0 PROGRAM/VERIFY MODE ELECTRICAL CHARACTERISTICS

## 5.1 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

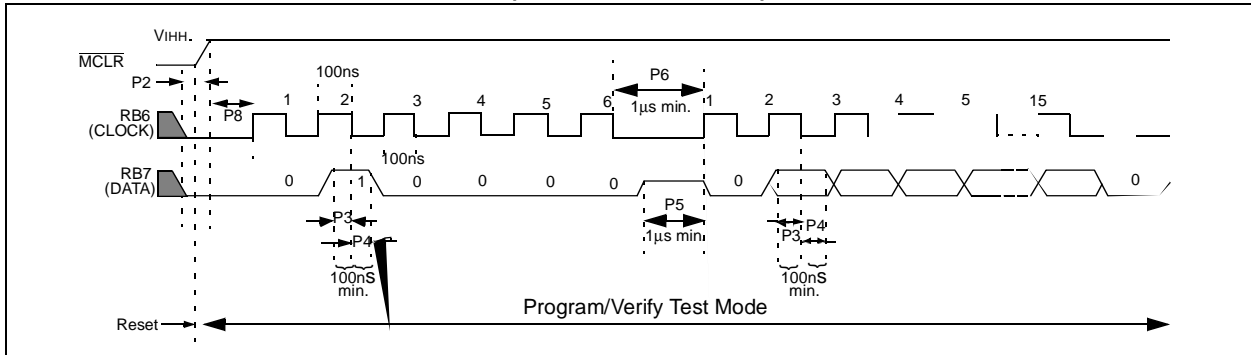
**TABLE 5-11. TIMING REQUIREMENTS FOR PROGRAM/VERIFY TEST MODE**

<b>Standard Operating Conditions</b>							
						<b>Operating Voltage:</b>	$4.5V \leq V_{DD} \leq 5.5V$
$T_A \leq +40^{\circ}\text{C}$ , unless otherwise stated, ( $25^{\circ}\text{C}$ is recommended) $V_{DD} \leq 5.5\text{V}$ , unless otherwise stated.							
Paramet							
	DDP	Supply voltage during programming	4.5	5.0	5.5	V	
	VDDV	Supply voltage during verify	VDDmin		VDDmax	V	Note 1 High voltage on MCLR Note 2
		for test mode entry					14.0 V
							50 mA
	DDP	Supply current (from VDD) during program/verify					
	IHH	Supply current from VIH (on MCLR)			200	$\mu\text{A}$	V
	IH1	(RB6, RB7) input high level	0.8 VDD			V	Schmitt Trigger input
	VIL1	(RB6, RB7) input low level MCLR	(test mode selection)			V	Schmitt Trigger input
P1	TvHHR	MCLR rise time (VSS to VHH) for test mode entry					8.0 $\mu\text{s}$
	P2	Tset0 RB6, RB7 setup time (before pattern)					
		Erase cycle time	100		10	ms	
		Program cycle time			10	ms	

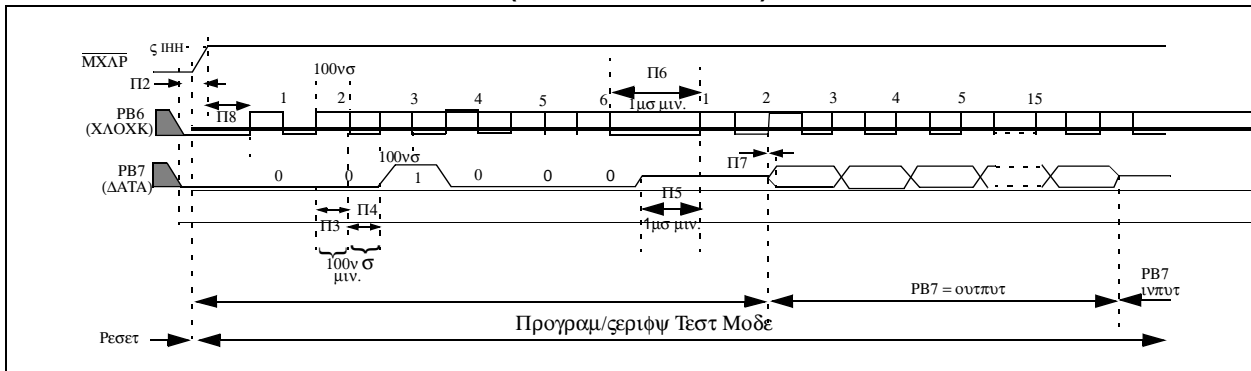
Note 1: Program must be verified at the minimum and maximum VDD limits for the part.

Note 2:  $V_{IH}$  must be greater than  $V_{DD} + 4.5V$  to stay in programming/verify mode.

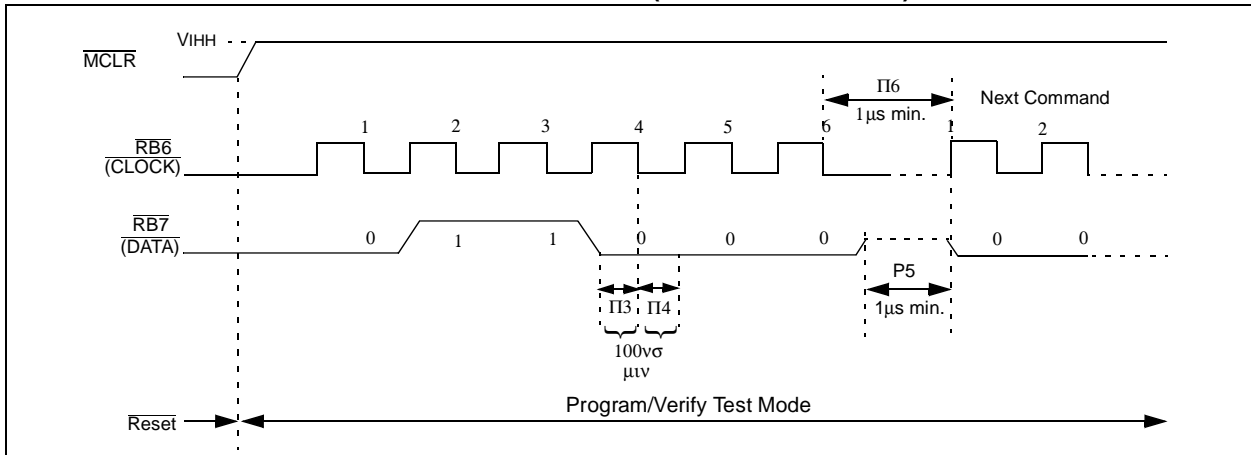
**FIGURE 5-1: LOAD DATA COMMAND (PROGRAM/VERIFY)**



**FIGURE 5-2: READ DATA COMMAND (PROGRAM/VERIFY)**



**FIGURE 5-3: INCREMENT ADDRESS COMMAND (PROGRAM/VERIFY)**





**MICROCHIP**

## WORLDWIDE SALES AND SERVICE

### AMERICAS

#### Corporate Office

Microchip Technology Inc.  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 602-786-7200 Fax: 602-786-7277  
Technical Support: 602 786-7627  
Web: <http://www.microchip.com>

#### Atlanta

Microchip Technology Inc.  
500 Sugar Mill Road, Suite 200B  
Atlanta, GA 30350  
Tel: 770-640-0034 Fax: 770-640-0307

#### Boston

Microchip Technology Inc.  
5 Mount Royal Avenue  
Marlborough, MA 01752  
Tel: 508-480-9990 Fax: 508-480-8575

#### Chicago

Microchip Technology Inc.  
333 Pierce Road, Suite 180  
Itasca, IL 60143  
Tel: 630-285-0071 Fax: 630-285-0075

#### Dallas

Microchip Technology Inc.  
14651 Dallas Parkway, Suite 816  
Dallas, TX 75240-8809  
Tel: 972-991-7177 Fax: 972-991-8588

#### Dayton

Microchip Technology Inc.  
Two Prestige Place, Suite 150  
Miamisburg, OH 45342  
Tel: 937-291-1654 Fax: 937-291-9175

#### Los Angeles

Microchip Technology Inc.  
18201 Von Karman, Suite 1090  
Irvine, CA 92612  
Tel: 714-263-1888 Fax: 714-263-1338

#### New York

Microchip Technology Inc.  
150 Motor Parkway, Suite 202  
Hauppauge, NY 11788  
Tel: 516-273-5305 Fax: 516-273-5335

#### San Jose

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

#### Toronto

Microchip Technology Inc.  
5925 Airport Road, Suite 200  
Mississauga, Ontario L4V 1W1, Canada  
Tel: 905-405-6279 Fax: 905-405-6253

### ASIA/PACIFIC

#### Hong Kong

Microchip Asia Pacific  
RM 3801B, Tower Two  
Metroplaza  
223 Hing Fong Road  
Kwai Fong, N.T., Hong Kong  
Tel: 852-2-401-1200 Fax: 852-2-401-3431

#### India

Microchip Technology Inc.  
India Liaison Office  
No. 6, Legacy, Convent Road  
Bangalore 560 025, India  
Tel: 91-80-229-0061 Fax: 91-80-229-0062

#### Japan

Microchip Technology Intl. Inc.  
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  
Tel: 82-2-554-7200 Fax: 82-2-558-5934

#### Shanghai

Microchip Technology  
RM 406 Shanghai Golden Bridge Bldg.  
2077 Yan'an Road West, Hong Qiao District  
Shanghai, PRC 200335  
Tel: 86-21-6275-5700  
Fax: 86 21-6275-5060

#### Singapore

Microchip Technology Singapore Pte Ltd.  
200 Middle Road  
#07-02 Prime Centre  
Singapore 188980  
Tel: 65-334-8870 Fax: 65-334-8850

### ASIA/PACIFIC (continued)

#### Taiwan, R.O.C

Microchip Technology Taiwan  
10F-1C 207  
Tung Hua North Road  
Taipei, Taiwan, ROC  
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

### EUROPE

#### United Kingdom

Arizona Microchip Technology Ltd.  
505 Eskdale Road  
Winnersh Triangle  
Wokingham  
Berkshire, England RG41 5TU  
Tel: 44-1189-21-5858 Fax: 44-1189-21-5835

#### France

Arizona Microchip Technology SARL  
Zone Industrielle de la Bonde  
2 Rue du Buisson aux Fraises  
91300 Massy, France  
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

#### Germany

Arizona Microchip Technology GmbH  
Gustav-Heinemann-Ring 125  
D-81739 München, Germany  
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

#### Italy

Arizona Microchip Technology SRL  
Centro Direzionale Colleoni  
Palazzo Taurus 1 V. Le Colleoni 1  
20041 Agrate Brianza  
Milan, Italy  
Tel: 39-39-6899939 Fax: 39-39-6899883

4/3/98



*Microchip received ISO 9001 Quality System certification for its worldwide headquarters, design, and wafer fabrication facilities in January, 1997. Our field-programmable PICmicro™ 8-bit MCUs, Serial EEPROMs, related specialty memory products and development systems conform to the stringent quality standards of the International Standard Organization (ISO).*

All rights reserved. © 5/98, Microchip Technology Incorporated, USA. Printed on recycled paper.

Information contained in this publication regarding device applications and the like is intended for suggestion only and may be superseded by updates. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights. The Microchip logo and name are registered trademarks of Microchip Technology Inc. in the U.S.A. and other countries. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.