PIC16F7X

FLASH Memory Programming Specification

This document includes the programming specifications for the following devices:

- PIC16F73
- PIC16F74
- PIC16F76
- PIC16F77

1.0 PROGRAMMING THE PIC16F7X

The PIC16F7X is programmed using a serial method. The Serial mode allows the PIC16F7X to be programmed while in the users' system, allowing for increased design flexibility. This programming specification applies to PIC16F7X devices in all packages.

1.1 Hardware Requirements

The PIC16F7X requires two programmable power supplies, one for VDD (2.0 V to 5.5 V) and the other for VPP of 12.75 V to 13.25 V. Both supplies should have a minimum resolution of 0.25 V.

1.2 Programming Mode

The Programming mode for the PIC16F7X allows programming of user program memory, special locations used for ID, and the configuration word.

Pin Diagram

PDIP, SOIC

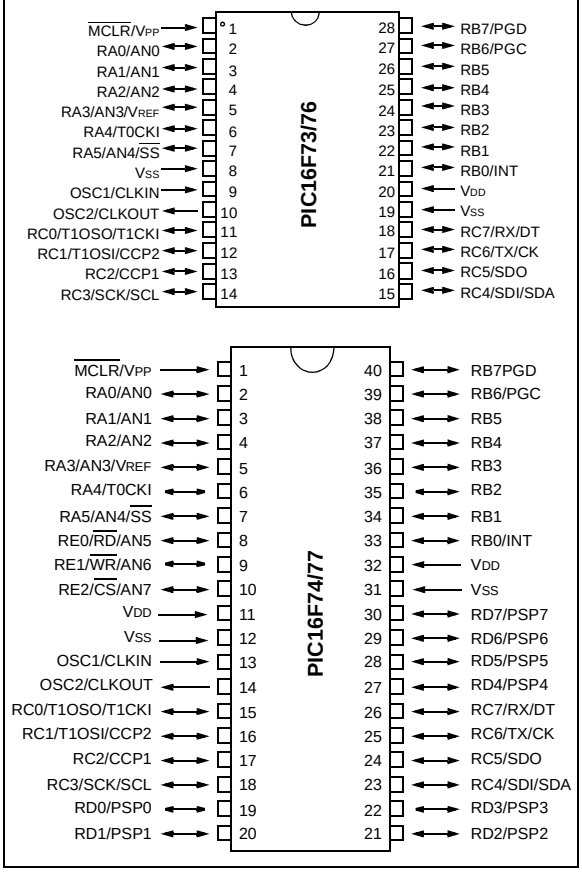


TABLE 1-1: PIN DESCRIPTIONS (DURING PROGRAMMING): PIC16F7X

Pin Name	During Programming		
	Function	Pin Type	Pin Description
RB6/PGC	CLOCK	I	Clock Input
RB7/PGD	DATA	I/O	Data Input/Output
MCLR/VPP	VTEST MODE	P	Program Mode Select
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground

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

2.0 PROGRAM MODE ENTRY

2.1 User Program Memory Map

The user memory space extends from 0x0000 to 0x1FFF (8 K), or 0x0000 to 0x0FFF (4 K). Table 2-1 shows the actual implementation of program memory in the PIC16F7X family. Configuration memory begins at 0x2000, and continues to 0x3FFF. The PC will increment from 0x0000 to 0x1FFF and wrap to 0x0000, 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 program memory is to reset the part and reenter Program/Verify mode, as described in Section 2.3.

Configuration memory is selected when the PC points to any address in the range of 0x2000-0x201F; however, only locations 0x2000 through 0x2007 are implemented. Addressing locations beyond 0x201F will access program memory (see Figure 2-1).

TABLE 2-1: IMPLEMENTATION OF PROGRAM MEMORY IN THE PIC16F7X FAMILY

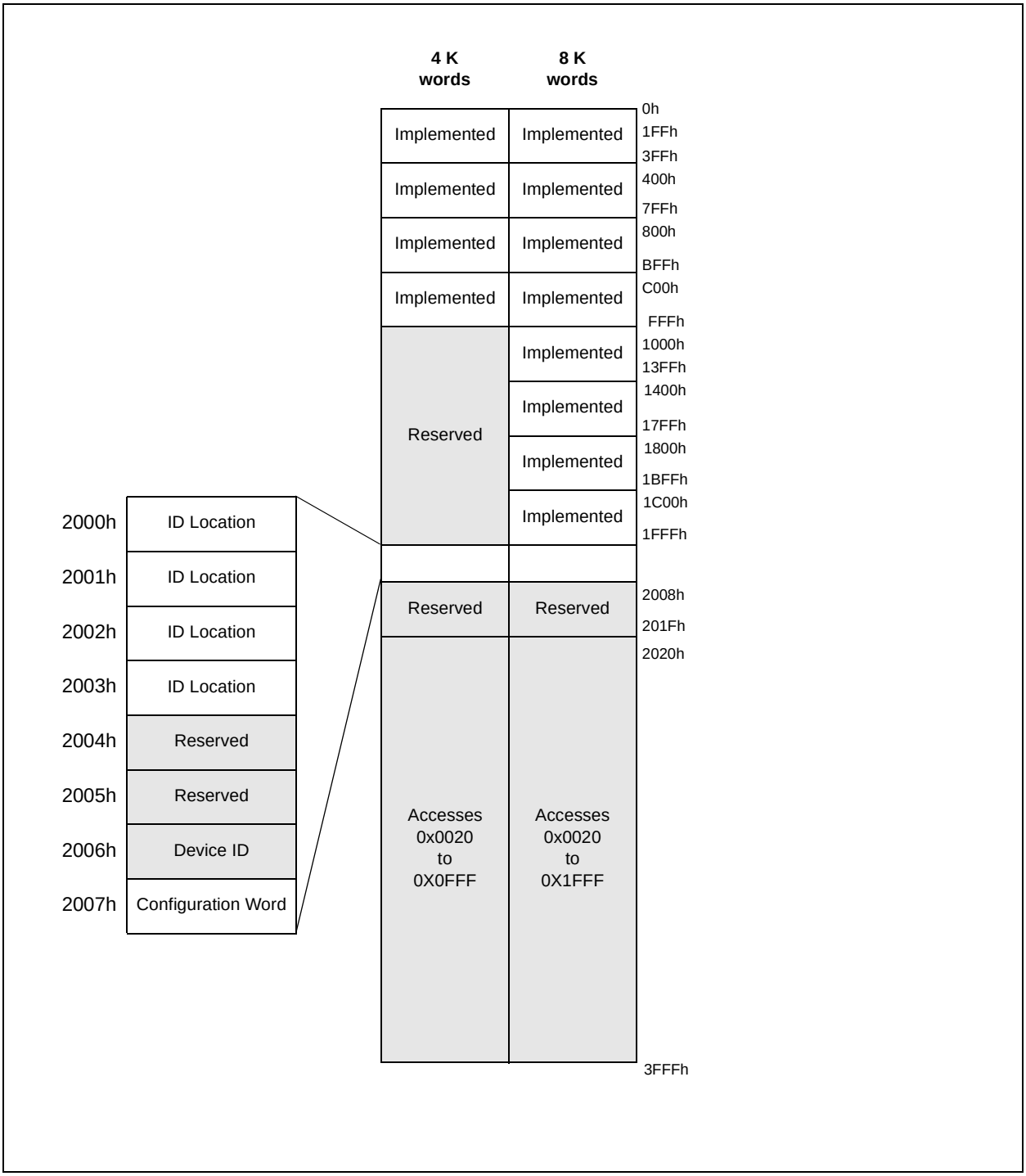
Device	Program Memory Size
PIC16F73	0x0000 – 0x0FFF (4K)
PIC16F74	0x0000 – 0x0FFF (4K)
PIC16F76	0x0000 – 0x1FFF (8K)
PIC16F77	0x0000 – 0x1FFF (8K)

2.2 ID Locations

A user may store identification information (ID) in four ID locations mapped to [0x2000:0x2003]. It is recommended that each ID location word is written as '`11 1111 1000 bbbb`' where '`bbbb`' is ID information. The ID locations can be read after code protection is enabled.

To understand the program memory read mechanism after code protection is enabled, refer to Section 4.0. Table 4-1 shows specific calculations and behavior for each of the PIC16F7X devices.

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 MCLR pin from VIL to VPP. Once in this mode, the user program memory and the configuration memory can be accessed and programmed in serial fashion. (RB6 and RB7 are Schmitt Trigger Inputs in this mode.)

The sequence that enters the device into the Programming/Verify mode places all other logic into the RESET state. All I/O are in the RESET state (high impedance inputs).

A device RESET will clear the PC and point to address 0x0000. 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 2-2.

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

1. Issue the 'Load Data' command to load a word at the current (even) program memory address.
2. Issue an 'Increment Address' command.
3. Load a word at the current (odd) program memory address using the 'Load Data' command.
4. Issue a 'Begin Programming' command to begin programming.
5. Wait t_{prog} (about 1 ms).
6. Issue an 'End Programming' command.
7. Increment to the next address.
8. Repeat this sequence as required to write program and configuration memory.

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

1. Set a word for the current memory location using the 'Load Data' command.
2. Issue a 'Begin Programming' command to begin programming.
3. Wait t_{prog}.
4. Issue an 'End Programming' command.
5. Increment to the next address.
6. Repeat this alternative sequence as required to write program and configuration memory.

The address and program counter is 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.

2.3.1 SERIAL PROGRAM/VERIFY OPERATION

RB6 is used as a clock input pin, and RB7 is used for entering command bits and data input/output. To enter a command, the clock pin (RB6) is pulsed six times. Each command bit is latched on the falling edge of the clock (RB6) with the least significant bit (LSb) of the command being input first. The data on pin RB7 needs a minimum setup (t_{set1}) and hold time (t_{hold1}) with respect to the falling edge of the clock. The read and load commands are specified to have a minimum delay (t_{dly1}) between the command and 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 (see Figure 5-1).

During a read operation, the LSb will be output on pin RB7 on the rising edge of the second clock pulse and during a load operation, the LSb will be latched on the falling edge of the second clock pulse. A minimum delay (t_{dly2}) is required between consecutive commands (see Figure 5-2).

To allow for decoding of commands and reversal of data pin configuration, a time separation of at least (t_{dly1}) is required between a command and a data word, or another command (see Figure 5-3).

The available commands are listed below:

- Load Configuration
- Load Data for Memory
- Read Data from Memory
- Increment Address
- Begin Programming
- Bulk Erase Program Memory
- End Programming

TABLE 2-2: COMMAND MAPPING FOR PIC16F7X

Command	Mapping (MSb … LSb)						Data
Load Configuration (Set PC = 2000h)	X	X	0	0	0	0	0, data (14), 0
Load Data for Memory	X	X	0	0	1	0	0, data (14), 0
Read Data from Memory	X	X	0	1	0	0	0, data (14), 0
Increment Address	X	X	0	1	1	0	
Begin Programming	X	X	1	0	0	0	
Bulk Erase Program Memory (Chip Erase)	X	X	1	0	0	1	
End Programming	X	X	1	1	1	0	

2.3.1.1 LOAD CONFIGURATION

After receiving the Load Configuration command, the PC will be set to 0x2000 and the data sent with the command is discarded. The four ID locations and the configuration word can then be programmed using the normal programming sequence, as described in Section 2.3. 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 $\overline{MCLR}$ low.

2.3.1.2 LOAD DATA FOR MEMORY

The device 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.

2.3.1.3 READ DATA FROM MEMORY

The device will transmit data bits out of the memory (program or configuration) currently addressed by the PC, starting with the second rising edge of the clock input. RB7 will go into output mode on the second rising clock edge and will revert back to input mode (hi-impedance) after the 16th rising edge. A timing diagram for this command is shown in Figure 5-2.

If the device is code protected, user program memory will read all '0's. Configuration memory can still be read.

2.3.1.4 INCREMENT ADDRESS

The PC is incremented by one. A timing diagram for this command is shown in Figure 5-3.

2.3.1.5 BEGIN PROGRAMMING

A 'Load Data' command must be issued before every 'Begin Programming' command. Programming of memory (configuration or program) will begin after this command is received and decoded. Programming requires (t_{prog}) time and is terminated using an 'End Programming' command.

2.3.1.6 CHIP ERASE (PROGRAM MEMORY)

Erasure of configuration and program memory begins after this command is received and decoded. The erase sequence is self-timed and it is not necessary to issue an 'End Programming' command, only to wait for the appropriate time interval (t_{era}) for the entire erase sequence, before issuing another command.

This procedure will disable code protection (code protect bit = 1); however, all data within the program memory will be erased when this command is executed and thus, the security of the data or code is not compromised.

Note:	All CHIP ERASE operations must take place with V_{DD} between 4.75 V and 5.25 V.

2.4 Programming Algorithm Requires Variable V_{DD}

The PIC16F7X uses an intelligent algorithm. The algorithm calls for program verification at V_{DDAPP}.

The actual chip erase and programming must be done with V_{DD} in the V_{DDP} range (See Table 5-1).

V_{DDP} = V_{DD} range required during programming

V_{DDAPP} = V_{DD} in the target application

Programmers must verify the PIC16F7X at V_{DDAPP}. Since Microchip may introduce future versions of the PIC16F7X with a broader V_{DD} range, it is best that these levels are user selectable (defaults are OK).

Note:	Any programmer not meeting this requirement may only be classified as a "prototype" or "development" programmer, but not a "production quality" programmer.

FIGURE 2-2: PROGRAMMING METHOD FLOW CHART (SHEET 1 OF 3)

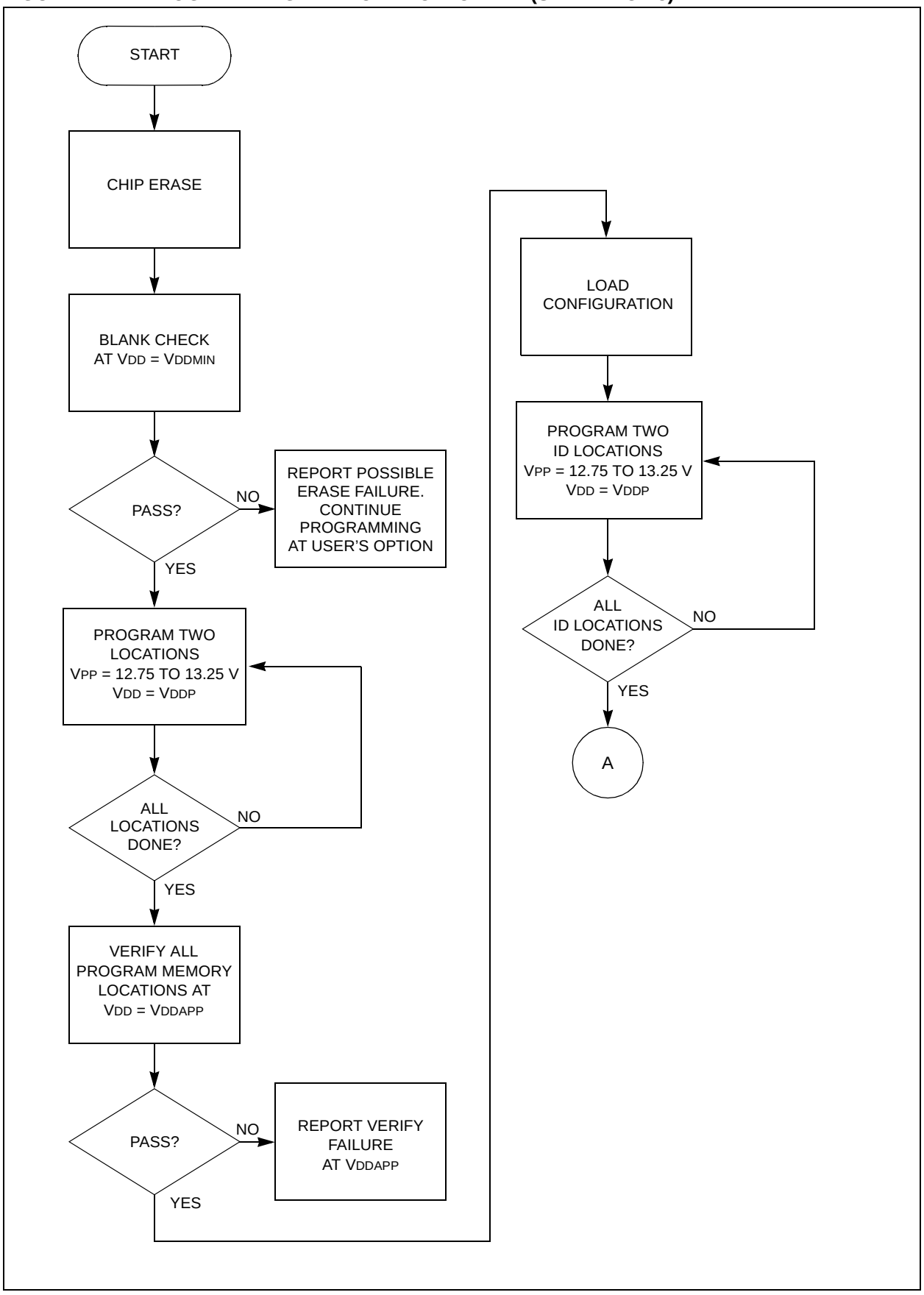


FIGURE 2-3: PROGRAMMING METHOD FLOW CHART (SHEET 2 OF 3)

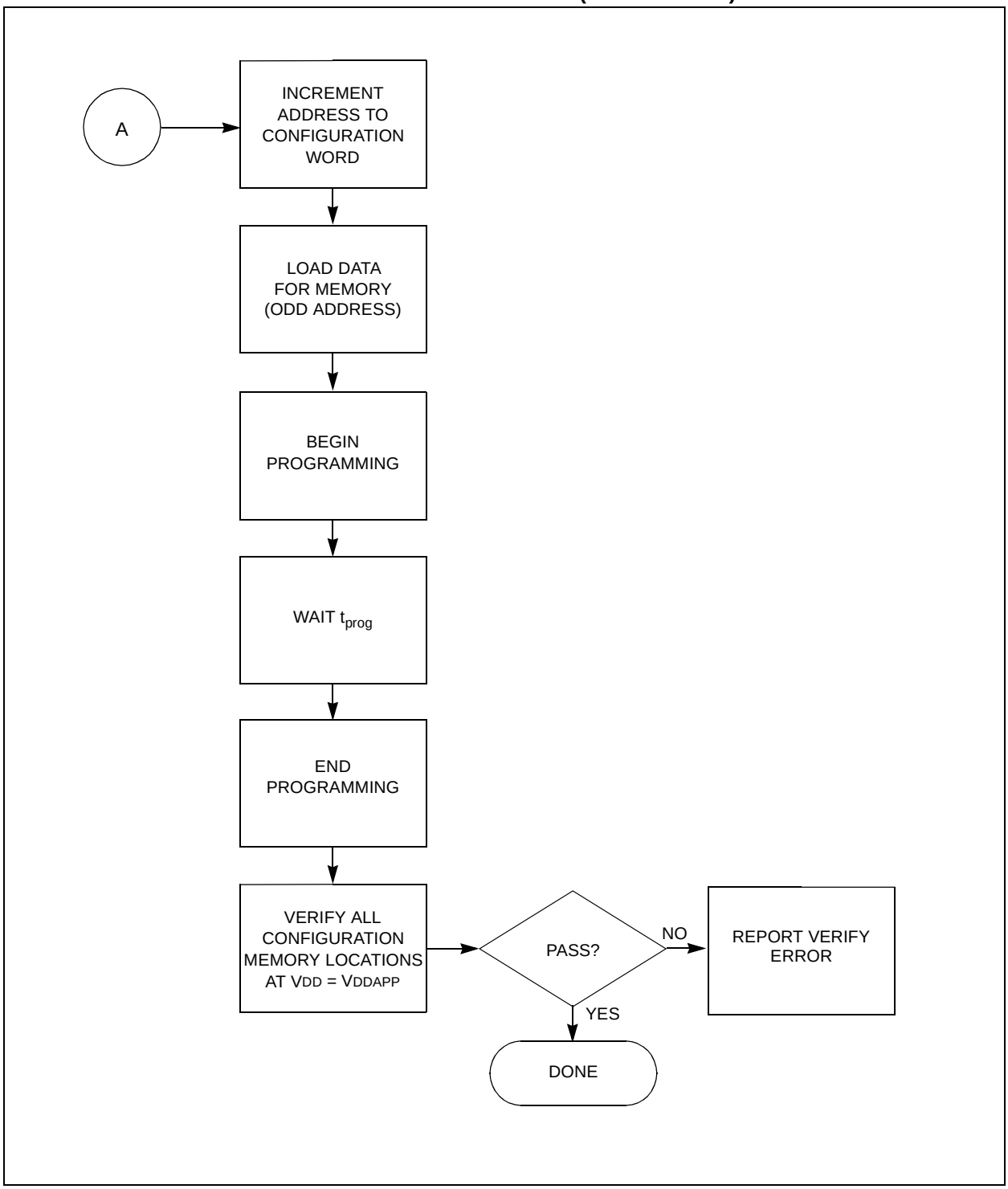
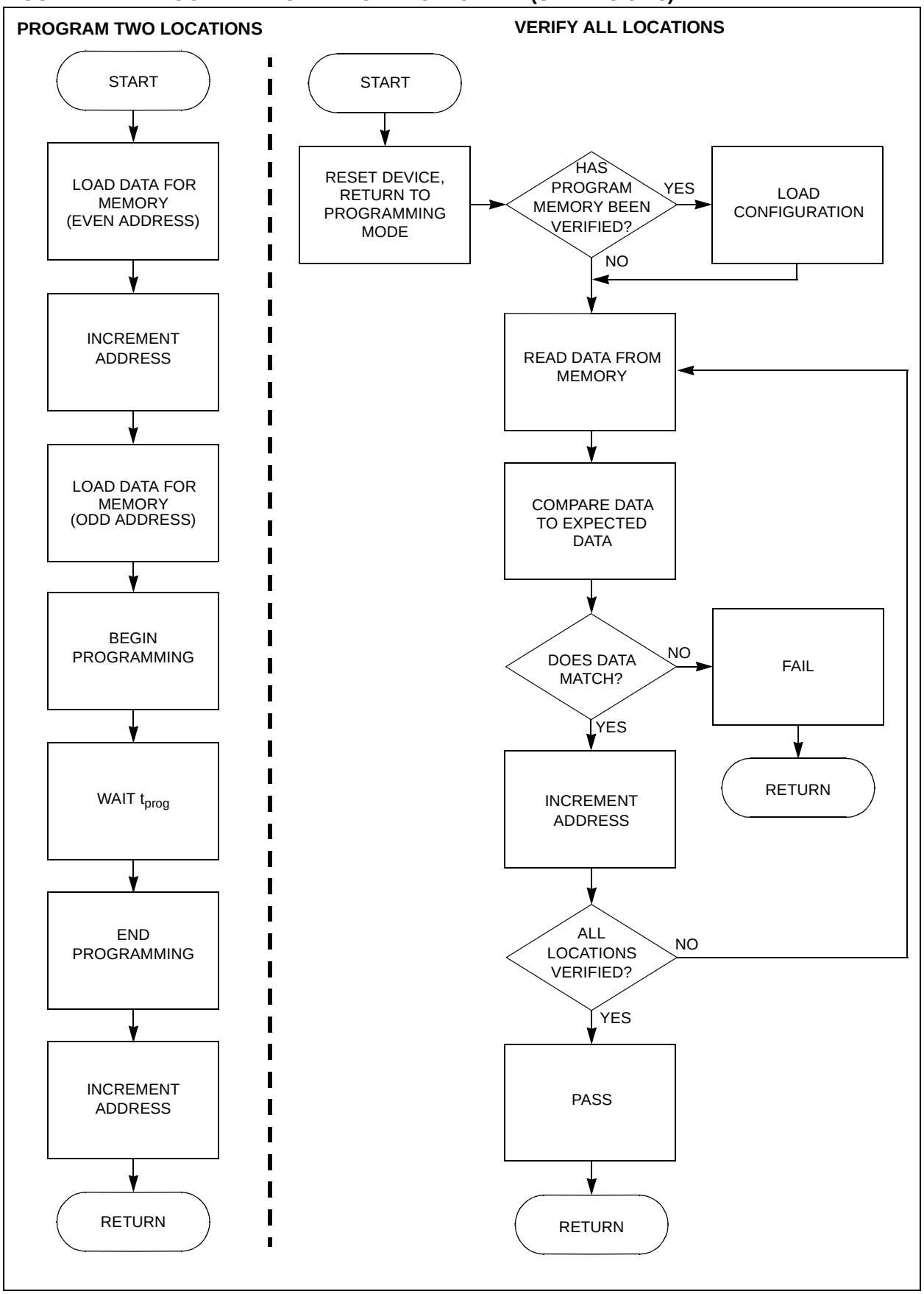


FIGURE 2-4: PROGRAMMING METHOD FLOW CHART (SHEET 3 OF 3)



3.0 CONFIGURATION WORD

The PIC16F7X has configuration bits in a configuration word located at 0x2007. These bits can be cleared (reads '0'), or left unchanged (reads '1'), to select various device configurations.

3.1 Device ID Word

The device ID word for the PIC16F7X is located at 2006h. The nine most significant bits are the device ID number, while the five least significant bits are the device revision number.

TABLE 3-1: DEVICE ID VALUE

Device	Device ID Word (0x2006)	
	Dev	Rev
PIC16F73	00 0110 000	n nnnn
PIC16F74	00 0110 001	n nnnn
PIC16F76	00 0110 010	n nnnn
PIC16F77	00 0110 011	n nnnn

FIGURE 3-1: CONFIGURATION WORD FOR PIC16F7X

-	-	-	-	-	-	-	BODEN	-	CP0	$\overline{\text{PWRTE}}$	WDTE	F0SC1	F0SC0
bit13													bit0

Register: CONFIG
Address 2007h

bit 13-7: **Unimplemented**

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

bit 5: **Unimplemented**

bit 4: **CP0**: Program Memory Code Protection bit
1 = Code protection off
0 = 0000h to 1FFFh code protected (All)

bit 3: **$\overline{\text{PWRTE}}$**: 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 1-0: **FOSC1:FOSC0**: Oscillator Selection bits
11 = RC oscillator
10 = HS oscillator
01 = XT oscillator
00 = LP oscillator

Note 1: Enabling Brown-out Reset automatically enables Power-up Timer (PWRT), regardless of the value of bit $\overline{\text{PWRTE}}$.

4.0 CODE PROTECTION

Once code protection is enabled, all program memory locations read all '0's; further programming of program memory is disabled. ID locations and the configuration word may still be read and programmed (1's to 0's only).

4.1 Disabling Code Protection

The following procedure should be performed before any other programming is attempted. This procedure also turns off code protection (code protect bit = 1); however, ***all program memory will be erased when this procedure is executed and thus, the security of the code is not compromised.***

Procedure to disable code protection:

a) Issue the 'Chip Erase' command.
b) Wait for the erase cycle time (t_{era}) to pass. The program memory is erased, then the configuration memory is erased.

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.

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

4.3 Checksum Computation

The checksum is calculated by reading the contents of the PIC16F7X memory locations and adding up the opcodes, up to the maximum user addressable location (e.g., 0x1FFF for the PIC16F7X). 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 PIC16F7X devices is shown in Table 4-1.

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.

Table 4-1 describes how to calculate the checksum for each device. Note that the checksum calculation differs depending on the code protection setting. Since the program memory locations read out differently depending on the code protection 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 of a non-protected device, the entire program memory can simply be read and summed. The configuration word and ID locations can always be read.

TABLE 4-1: CHECKSUM COMPUTATION

Device	Code Protect	Checksum	Blank Value	0x05E6 at 0x0000 and max address
PIC16F73	OFF	SUM[0x000:0x0FFF] + CFWD & 0x005F	0xF05F	0x7C2D
	ALL	CFWD & 0x005F + SUM_ID	0x005E	0x005E
PIC16F74	OFF	SUM[0x000:0x0FFF] + CFWD & 0x005F	0xF05F	0x7C2D
	ALL	CFWD & 0x005F + SUM_ID	0x005E	0x005E
PIC16F76	OFF	SUM[0x000:0x1FFF] + CFWD & 0x005F	0xE05F	0x8C2D
	ALL	CFWD & 0x005F + SUM_ID	0x005E	0x005E
PIC16F77	OFF	SUM[0x000:0x1FFF] + CFWD & 0x005F	0xE05F	0x8C2D
	ALL	CFWD & 0x005F + SUM_ID	0x005E	0x005E

Legend: CFWD = Configuration Word
SUM[a:b] = [Sum of locations a to b inclusive]
SUM_ID = ID locations masked by 0x0F, then concatenated into a 16-bit value with ID0 as the most significant nibble. For example, ID0 = 0x01, ID2 = 0x02, ID3 = 0x03, ID4 = 0x04, then SUM_ID = 0x1234
Checksum = [Sum of all the individual expressions] **MODULO** [0xFFFF]
\+ = Addition
& = Bitwise AND

5.0 PROGRAM/VERIFY MODE ELECTRICAL CHARACTERISTICS

5.1 AC/DC Characteristics

TABLE 5-1: TIMING REQUIREMENTS FOR PROGRAM/VERIFY MODE

Standard Operating Conditions (unless otherwise stated) **Operating Temperature: +10°C ≤ TA ≤ +40°C** **Operating Voltage: 4.5 V ≤ VDD ≤ 5.5 V**						
Characteristics	**Sym**	**Min**	**Typ**	**Max**	**Units**	**Conditions/Comments**
General						
VDD level for read and verification	VDD	2.0		5.5	V	
VDD level for programming and erasing	VDDP	4.75		5.25	V	
High voltage on $\overline{MCLR}$ for chip erase and program write operations	VPP	12.75		13.25	V	**Note 1, 2**
$\overline{MCLR}$ rise time (VSS to VPP) for Test mode entry	tVHHR			1.0	µs	
(RB6, RB7) input high level	VIH1	0.8 VDD			V	Schmitt Trigger input
(RB6, RB7) input low level	VIL1	0.2 VDD			V	Schmitt Trigger input
Serial Program/Verify						
Data in setup time before clock↓	t_{set1}	100			ns	
Data in hold time after clock↓	t_{hld1}	100			ns	
Data input not driven to next clock input (delay required between command/data or command/command)	t_{dly1}	1.0			µs	
Delay between clock↓ to clock↑ of next command or data	t_{dly2}	1.0			µs	
Clock↑ to data out valid (during read data)	t_{dly3}	200			ns	
Erase cycle time	t_{era}	30			ms	**Note 3**
Programming cycle time	t_{prog}	1	—	3**(4)**	ms	

Note 1: VPP should be current limited to about 100 mA.

2: VPP must remain above VDDP + 4.0 V to remain in Programming mode, while not actually erasing or programming.

3: The chip erase is self-timed.

4: t_{prog} is expected to be reduced to 1 mS max.

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

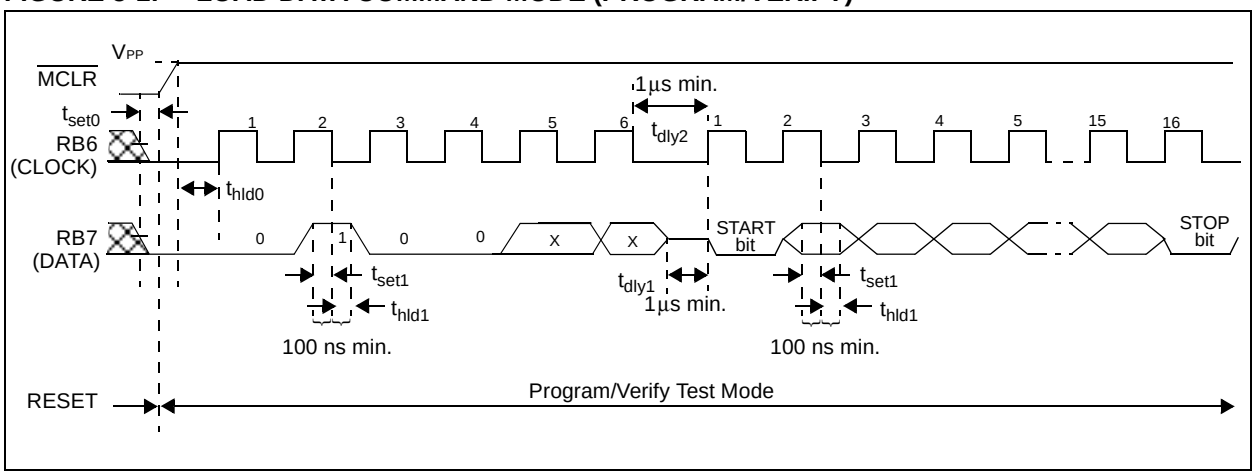


FIGURE 5-2: READ DATA COMMAND MODE (PROGRAM/VERIFY)

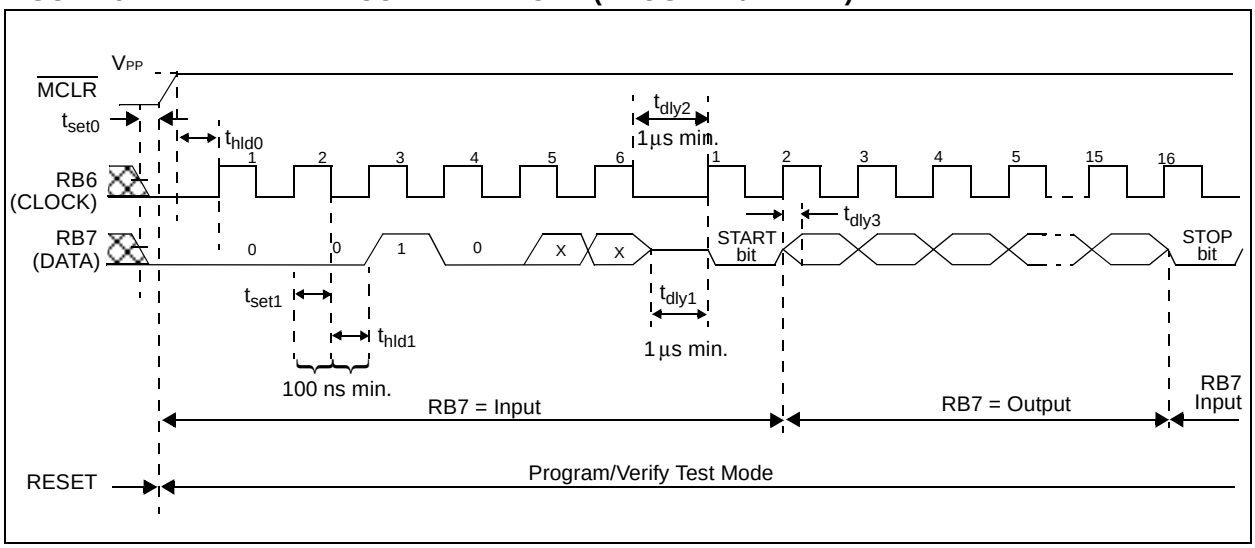
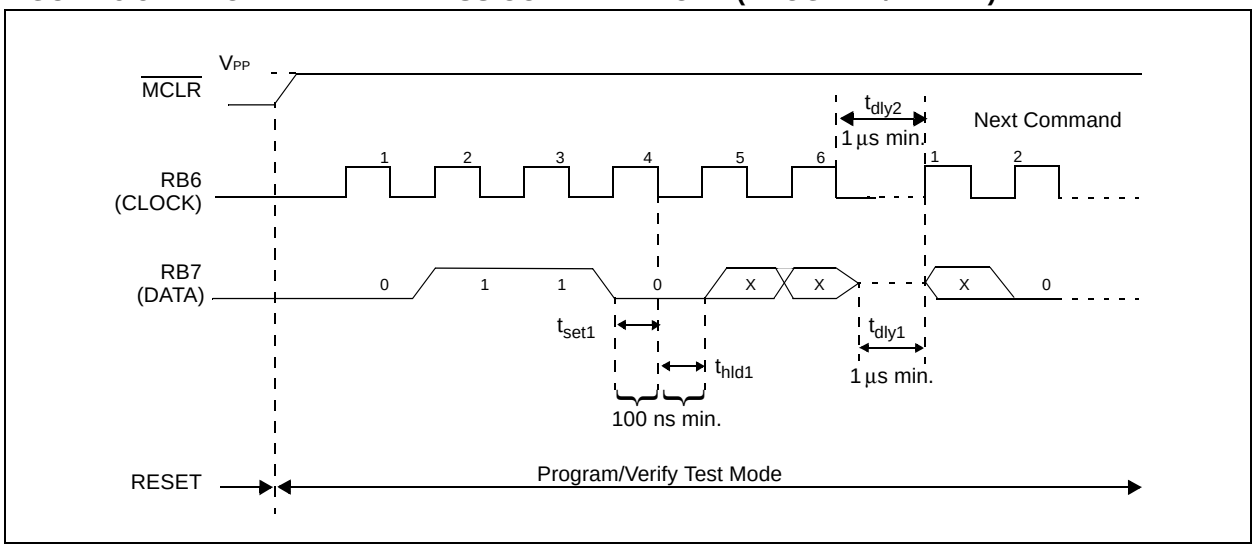


FIGURE 5-3: INCREMENT ADDRESS COMMAND MODE (PROGRAM/VERIFY)



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: 480-792-7456

Atlanta
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
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

Dayton
Two Prestige Place, Suite 130
Miamisburg, OH 45342
Tel: 937-291-1654 Fax: 937-291-9175

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

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

New York
150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 631-273-5305 Fax: 631-273-5335

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

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

ASIA/PACIFIC

China - Beijing
Microchip Technology Beijing Office
Unit 915
New China Hong Kong Manhattan Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

China - Shanghai
Microchip Technology Shanghai Office
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

Hong Kong
Microchip Asia Pacific
RM 2101, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

India
Microchip Technology Inc.
India Liaison Office
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 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

ASIA/PACIFIC (continued)

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

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

EUROPE

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

France
Arizona 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
Arizona Microchip Technology GmbH
Gustav-Heinemann Ring 125
D-81739 Munich, 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: 39-039-65791-1 Fax: 39-039-6899883

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

10/01/00

DNV Certification, Inc. USA — ANSI-RAB QMS ACCREDITED — DNV MSC The Netherlands Accredited by the RvA — DNV — 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. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.