PIC16F8X

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

1.1 Hardware Requirements

The PIC16F8X devices require 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 devices allows programming of user program memory, data memory, special locations used for ID, and the configuration word. On PIC16CR8X devices, only data EEPROM and CDP can be programmed.

Pin Diagram

PDIP, SOIC

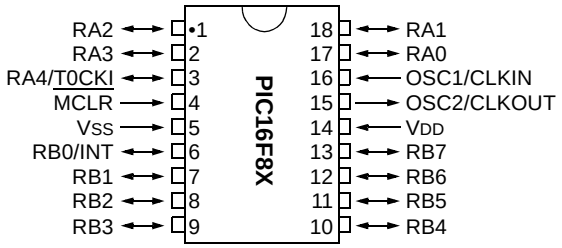


TABLE 1-1: PIN DESCRIPTIONS (DURING PROGRAMMING): PIC16F8X

Pin Name	During Programming		
	Function	Pin Type	Pin Description
RB6	CLOCK	I	Clock Input
RB7	DATA	I/O	Data Input/Output
$\overline{MCLR}$	VTEST MODE	P(1)	Program Mode Select
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground

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

Note 1: In the PIC16F8X, 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 MODE ENTRY

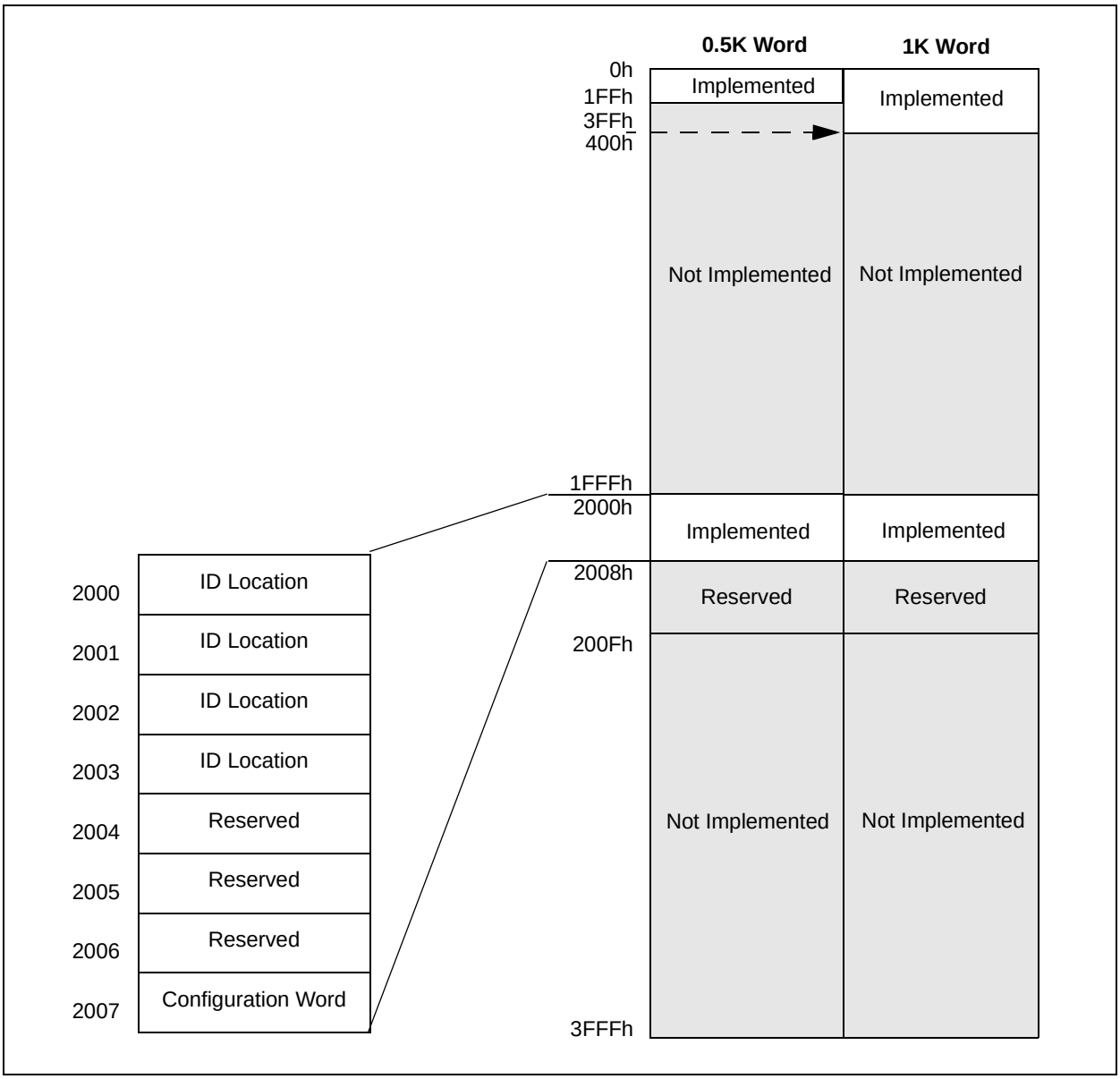
2.1 User Program Memory Map

The user memory space extends from 0000h to 1FFFh (8 Kbytes), of which 1 Kbyte (0000h - 03FFh) 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 03FFh, 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 0000h to 3FFFh, with the first half (0000h-1FFFh) being user program memory and the second half (2000h-3FFFh) being configuration memory. The PC will increment from 0000h to 1FFFh and wrap to 0000h, or 2000h to 3FFFh and wrap around to 2000h (not to 0000h). Once in configuration memory, the highest bit of the PC stays a '1', thus always pointing to the configuration memory. The only way to point to user program memory is to reset the part and re-enter Program/Verify mode, as described in Section 2.3.

In the configuration memory space, 2000h-200Fh are physically implemented. However, only locations 2000h through 2007h are available. Other locations are reserved. Locations beyond 2000Fh will physically access user memory (see Figure 2-1).

FIGURE 2-1: PROGRAM MEMORY MAPPING



2.2 ID Locations

A user may store identification information (ID) in four ID locations, mapped in addresses 2000h through 2003h. It is recommended that the user use only the four Least Significant bits of each ID location. The ID locations read out in an unscrambled fashion after code protection is enabled. It is recommended that ID location is written as "`11 1111 1000 bbbb`", where "`bbbb`" is ID information.

2.3 Program/Verify Mode

The Program/Verify mode is entered by holding pins RB6 and RB7 low, while raising MCLR pin from V_{IL} to V_{IHH} (high voltage). 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.

Note:	Do not allow excess time when transitioning MCLR between V_{IL} and V_{IHH}; this can cause spurious program executions to occur. The maximum transition time is $1T_{CY} + T_{PWRT}$ (if enabled) + 1024 T_{OSC} (for LP, HS and XT modes only) where T_{CY} is the Instruction Cycle Time, T_{PWRT} is the Power-up Timer Period, and T_{OSC} is the Oscillator Period (all values in µs or ns). For specific values, refer to the Electrical Characteristics section of the Device Data Sheet for the particular device.

The sequence that enters the device into the Programming/Verify mode places all other logic into the RESET state (the MCLR pin was initially at 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 in Table 5-1), with respect to the falling edge of the clock. Commands that have data associated with them (read and load) are specified to have a minimum delay of 1 µs between the command and the data. After this delay, the clock pin is cycled 16 times with the first cycle being a START bit and the last cycle being a STOP bit. 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 µs 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 µs is required between a command and a data word (or another command).

The available commands (Load Configuration and Load Data for Program Memory) are discussed in the following sections.

2.3.1.1 Load Configuration

After receiving this command, the program counter (PC) will be set to 2000h. 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 $\overline{MCLR}$ below 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 Erase Programming Cycle	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

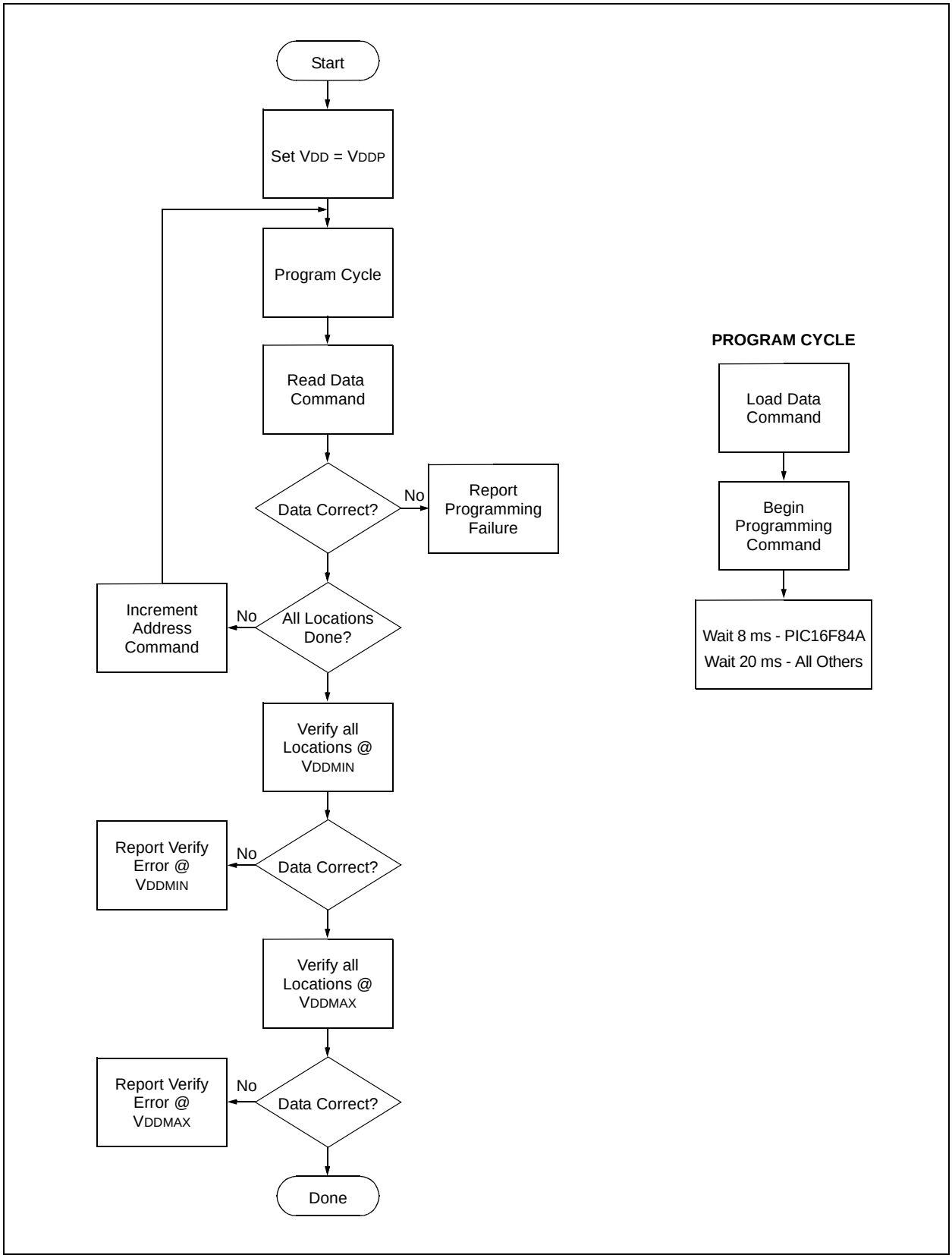
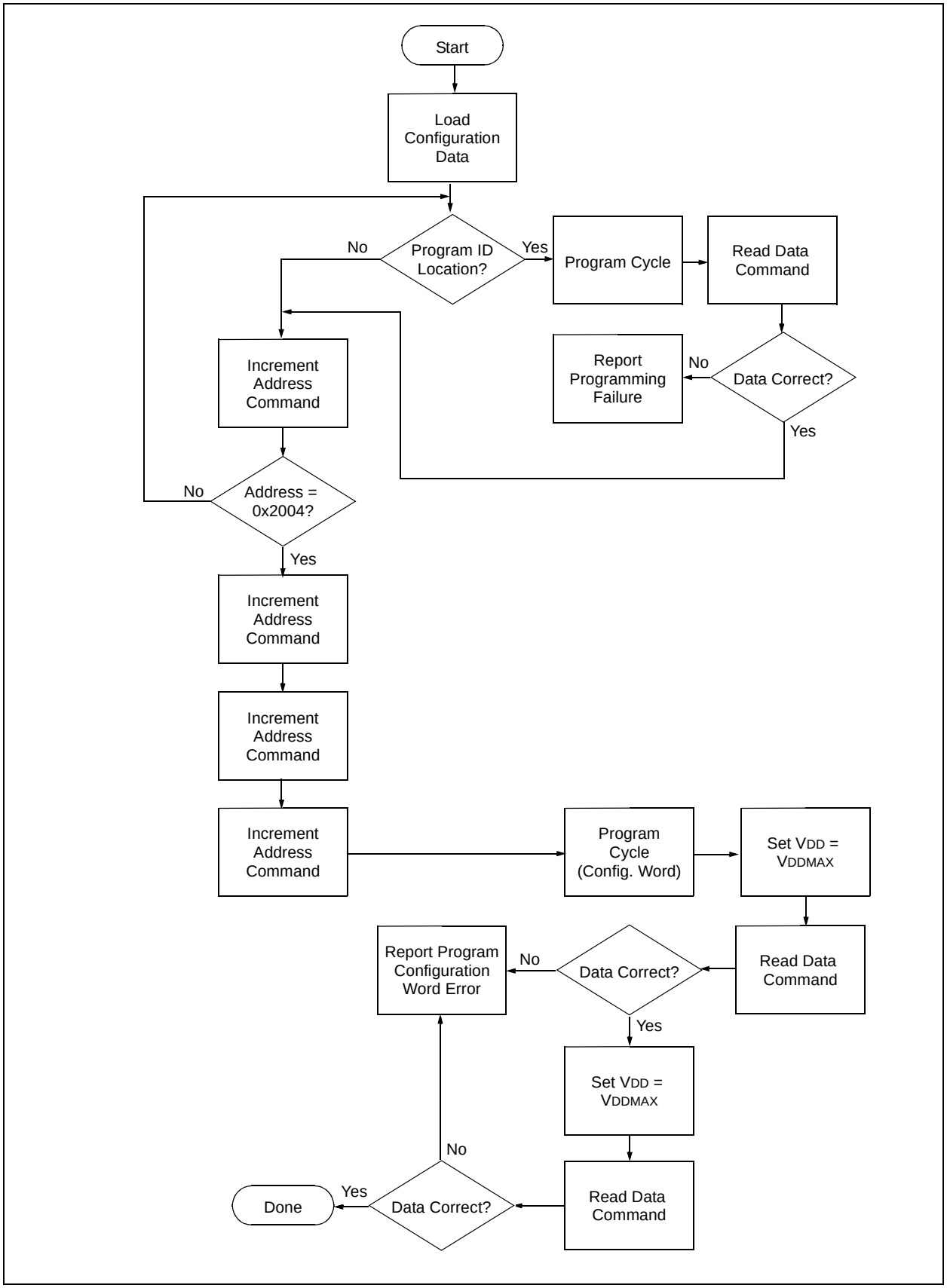


FIGURE 2-3: PROGRAM FLOW CHART - PIC16F8X CONFIGURATION MEMORY



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 (configuration 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

This command is available only on the PIC16F84A. A load command must be given before every begin programming command. Programming of the appropriate memory (configuration 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.

For PIC16F84A, perform the following commands:

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 configuration memory (2000h - 200Fh), 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.

For PIC16CR83/CR84 and PIC16F84, perform the following commands:

1. Issue Command 2 (write program memory)
2. Send out 3FFFH data
3. Issue Command 1 (toggle select even rows)
4. Issue Command 7 (toggle select even rows)
5. Issue Command 8 (begin programming)
6. Delay 10 ms
7. Issue Command 1 (toggle select even rows)
8. Issue Command 7 (toggle select even rows)

Note:	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.

For PIC16F84A, perform the following commands:

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

For PIC16CR83/CR84 and PIC16F84, perform the data memory:

5. Send out 3FFFH data
6. Issue Command 1 (toggle select even rows)
7. Issue Command 7 (toggle select even rows)
8. Issue Command 8 (begin data)
9. Delay 10 ms
10. Issue Command 1 (toggle select even rows)
11. Issue Command 7 (toggle select even rows)

Note:	All BULK ERASE operations must take place at 4.5 to 5.5 V_{DD} range.

2.4 Programming Algorithm Requires Variable V_{DD}

The PIC16F8X devices use an intelligent algorithm. The algorithm calls for program verification at V_{DDMIN}, as well as V_{DDMAX}. Verification at V_{DDMIN} ensures good “erase margin”. Verification at V_{DDMAX} ensures good “program margin”.

The actual programming must be done with V_{DD} in the V_{DDP} range (see Table 5-1):

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

V_{DDMIN} = minimum operating V_{DD} spec for the part

V_{DDMAX} = maximum operating V_{DD} spec for the part

Programmers must verify the PIC16F8X devices at their specified V_{DDMAX} and V_{DDMIN} levels. Since Microchip may introduce future versions of the PIC16F8X devices with a broader V_{DD} range, it is best that these levels are user selectable (defaults are acceptable).

Note:	Any programmer not meeting these requirements may only be classified as “prototype” or “development” programmer, but not a “production” quality programmer.

3.0 CONFIGURATION WORD

Most of the PIC16F8X devices have five configuration bits. These bits can be set (reads '0'), or left unchanged (reads '1') to select various device configurations. Their usage in the Device Configuration Word is shown in Register 3-1.

3.1 Device ID Word

The device ID word for the PIC16F84A device is located at 2006h. Older devices do not have device ID.

TABLE 3-1: DEVICE ID WORD

Device	Device ID Value	
	Dev	Rev
PIC16F84A	00 0101 011	X XXXX

REGISTER 3-1: CONFIGURATION WORD: PIC16F83/84/84A, PIC16CR83/84

For PIC16F83/84/84A:

CP	CP	CP	CP	CP	CP	CP	CP	CP	CP	$\overline{\text{PWTREN}}$	WDTEN	FOSC1	FOSC0

FOR PIC16CR83/84:

CP	CP	CP	CP	CP	CP	DP	CP	CP	CP	$\overline{\text{PWTREN}}$	WDTEN	FOSC1	FOSC0

bit13 bit0

bit 13-8, bit 6-4 **CP:** Code Protection bits(1)
1 = Code protection off
0 = Code protection on

bit 7 For PIC16F83/84/84A:
CP: Code Protection bits(1)
1 = Code protection off
0 = Code protection on
For PIC16CR83/84:
DP: Data Memory Code Protection bit
1 = Code protection off
0 = Data memory is code protected

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

Note 1: All of the CP bits have to be given the same value to enable the code protection scheme listed.

4.0 CODE PROTECTION

For PIC16F8X devices, once code protection is enabled, all program data 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.

For PIC16CR8X devices, once code protection is enabled, all program memory locations read all '0's; data memory locations read all '1's.

A description of the code protection schemes for the various PIC16F8X devices is provided on page 11. For each device, the bit configuration for the device configuration word to enable code protection is provided. This is followed with a comparison of read and write operations for selected memory spaces in both protected and unprotected modes.

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:

1. Execute load configuration (with a '1' in bits 4-13, code protect)
2. Increment to configuration word location (2007h)
3. Execute command (000001)
4. Execute command (000111)
5. Execute 'Begin Programming' (001000)
6. Wait 10 ms
7. Execute command (000001)
8. Execute command (000111)

4.2 Embedding Configuration Word and ID Information in the HEX File

Note: 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.

Device: PIC16F83

To code protect: 0000000000XXXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (2007h)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
All memory	Read All '0's, Write Disabled	Read Unscrambled, Write Enabled
ID Locations [2000h : 2003h]	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled

Device: PIC16CR83

To code protect: 0000000000XXXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (2007h)	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 [2000h : 2003h]	Read Unscrambled	Read Unscrambled

Device: PIC16CR84

To code protect: 0000000000XXXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (2007h)	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 [2000h : 2003h]	Read Unscrambled	Read Unscrambled

Device: PIC16F84

To code protect: 0000000000XXXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (2007h)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
All memory	Read All '0's, Write Disabled	Read Unscrambled, Write Enabled
ID Locations [2000h : 2003h]	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled

Device: PIC16F84A

To code protect: 0000000000XXXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (2007h)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
All memory	Read All '0's, Write Disabled	Read Unscrambled, Write Enabled
ID Locations [2000h : 2003h]	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., 1FFh for the PIC16F83. 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 4-1.

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 4-1: CHECKSUM COMPUTATION

Device	Code Protect	Checksum*	Blank Value	25E6h at 0 and Max Address
PIC16F83	OFF ON	SUM[000h:1FFh] + CFGW & 3FFFh CFGW & 3FFFh + SUM_ID	3DFFh 3E0Eh	09CDh 09DCh
PIC16CR83	OFF ON	SUM[000h:1FFh] + CFGW & 3FFFh CFGW & 3FFFh + SUM_ID	3DFFh 3E0Eh	09CDh 09DCh
PIC16F84	OFF ON	SUM[000h:3FFh] + CFGW & 3FFFh CFGW & 3FFFh + SUM_ID	3BFFh 3C0Eh	07CDh 07DCh
PIC16CR84	OFF ON	SUM[000h:3FFh] + CFGW & 3FFFh CFGW & 3FFFh + SUM_ID	3BFFh 3C0Eh	07CDh 07DCh
PIC16F84A	OFF ON	SUM[000h:3FFh] + CFGW & 3FFFh CFGW & 3FFFh + SUM_ID	3BFFh 3C0Eh	07CDh 07DCh

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 =01h, ID1 = 02h, ID3 = 03h, ID4 = 04h, then SUM_ID = 1234h.
*Checksum = [Sum of all the individual expressions] **MODULO** [FFFFh]
+ = Addition
& = Bitwise AND

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 data EEPROM contents to a HEX file, along with program memory information and fuse information.

The 64 data memory locations are logically mapped, starting at address 2100h. The format for data memory storage is one data byte per address location, LSB aligned.

TABLE 5-1: AC/DC CHARACTERISTICS TIMING REQUIREMENTS FOR PROGRAM/VERIFY TEST MODE

Standard Operating Conditions
Operating Temperature: +10°C ≤ TA ≤ +40°C, unless otherwise stated, (25°C is recommended)
Operating Voltage: 4.5V ≤ VDD ≤ 5.5V, unless otherwise stated

Parameter No.	Sym.	Characteristic	Min.	Typ.	Max.	Units	Conditions/Comments
	VDDP	Supply voltage during programming	4.5	5.0	5.5	V	
	VDDV	Supply voltage during verify	VDDMIN		VDDMAX	V	**(Note 1)**
	VIHH	High voltage on $\overline{MCLR}$ for Test mode entry	12		14.0	V	**(Note 2)**
	IDDP	Supply current (from VDD) during program/verify			50	mA	
	IHH	Supply current from VIHH (on $\overline{MCLR}$)			200	µA	
	VIH1	(RB6, RB7) input high level	0.8 VDD			V	Schmitt Trigger input
	VIL1	(RB6, RB7) input low level $\overline{MCLR}$ (Test mode selection)	0.2 VDD			V	Schmitt Trigger input
P1	TVHHR	$\overline{MCLR}$ rise time (VIL to VIHH) for Test mode entry			8.0	µs	
P2	Tset0	RB6, RB7 setup time (before pattern setup time)	100			ns	
P3	Tset1	Data in setup time before clock ↓	100			ns	
P4	Thld1	Data in hold time after clock ↓	100			ns	
P5	Tdly1	Data input not driven to next clock input (delay required between command/data or command/command)	1.0			µs	
P6	Tdly2	Delay between clock ↓ to clock ↑ of next command or data	1.0			µs	
P7	Tdly3	Clock to data out valid (during read data)	80			ns	
P8	Thld0	RB<7:6> hold time after $\overline{MCLR}$ ↑	100			ns	
—	—	Erase cycle time	—	—	4	ms	PIC16F84A only
—	—	Program cycle time	—	—	4	ms	PIC16F84A only
—	—	Erase and program time	— —	— —	8 20	ms ms	PIC16F84A only All other devices

Note 1: Program must be verified at the minimum and maximum VDD limits for the part.
2: VIHH must be greater than VDD + 4.5V to stay in Programming/Verify mode.

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

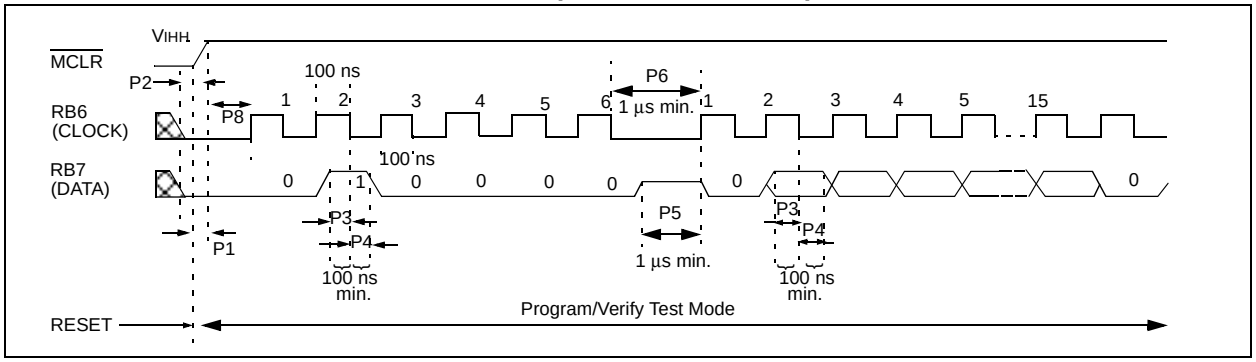


FIGURE 5-2: READ DATA COMMAND (PROGRAM/VERIFY)

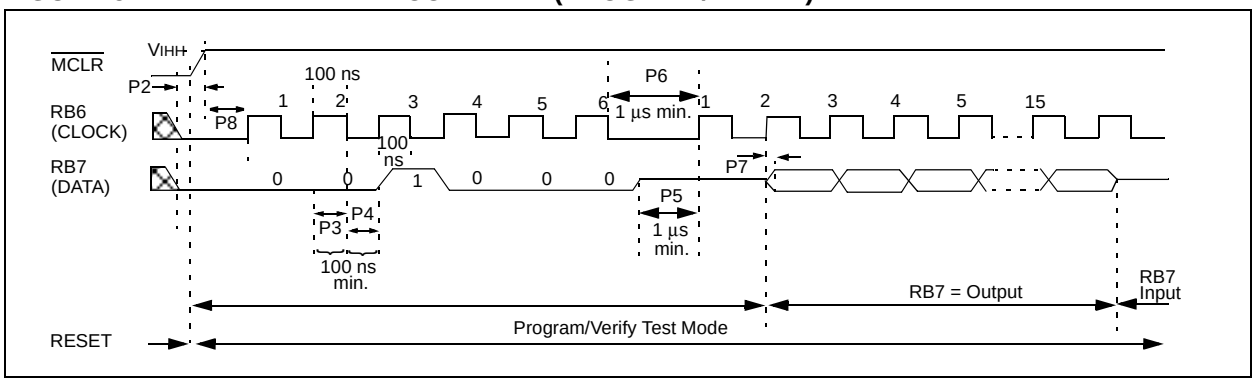
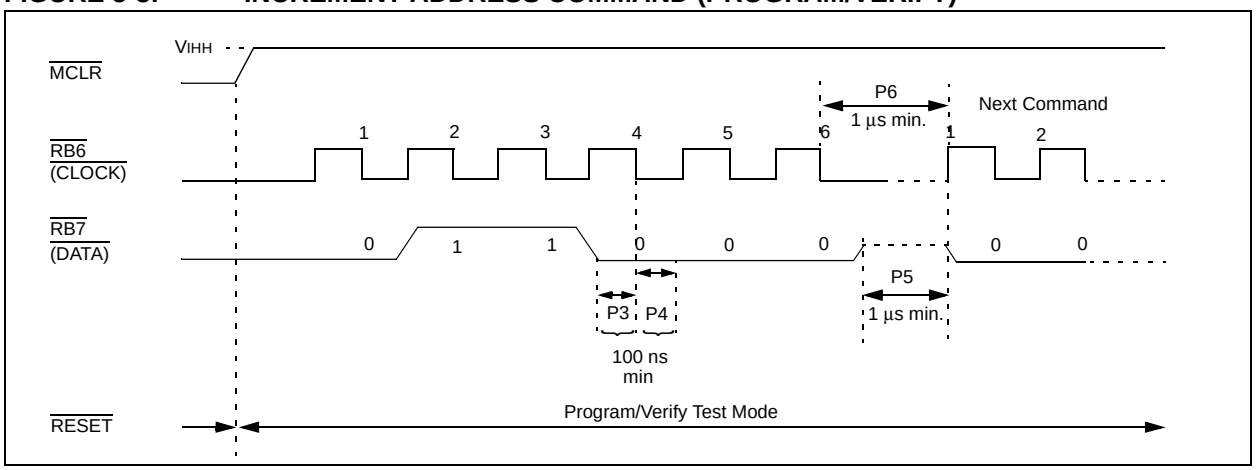


FIGURE 5-3: INCREMENT ADDRESS COMMAND (PROGRAM/VERIFY)



Note the following details of the code protection feature on PICmicro® MCUs.

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products 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 PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be 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 product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

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.

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

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

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

Australia
Microchip Technology Australia Pty Ltd
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, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-6766200 Fax: 86-28-6766599

China - Fuzhou
Microchip Technology Consulting (Shanghai) Co., Ltd., Fuzhou Liaison Office
Rm. 531, North Building
Fujian Foreign Trade Center Hotel
73 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7557563 Fax: 86-591-7557572

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. 1315, 13/F, Shenzhen Kerry Centre,
Renminnan Lu
Shenzhen 518001, China
Tel: 86-755-2350361 Fax: 86-755-2366086

Hong Kong
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

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 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: 65-334-8870 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 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
Gustav-Heinemann Ring 125
D-81739 Munich, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy
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

10/01/01