MICROCHIP

PIC14000

EPROM Memory Programming Specification

This document includes the programming specifications for the following devices:

- PIC14000

1.0 PROGRAMMING THE PIC14000

The PIC14000 can be programmed using a serial method. In serial mode the PIC14000 can be programmed while in the users system. This allows for increased design flexibility. This programming specification applies to PIC14000 devices in all packages.

1.1 Hardware Requirements

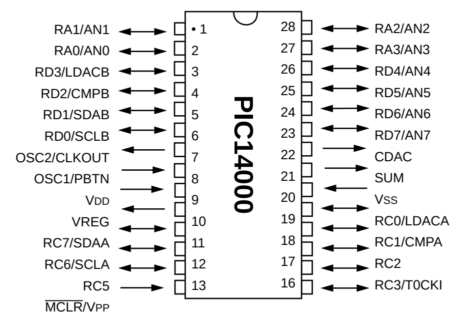
The PIC14000 requires two programmable power supplies, one for VDD (2.0V to 6.5V recommended) and one for VPP (12V to 14V).

1.2 Programming Mode

The programming mode for the PIC14000 allows programming of user program memory, configuration word, and calibration memory.

Pin Diagram

PDIP, SOIC, SSOP, Windowed CERDIP



Note: Peripheral pinout functions are not shown (see data sheets for full pinout information).

PIN DESCRIPTIONS (DURING PROGRAMMING): PIC14000

Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
RC6	CLOCK	I	Clock input
RC7	DATA	I/O	Data input/output
MCLR/VPP	VPP	P	Programming Power
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 program and calibration memory space extends from 0x000 to 0xFFF (4096 words). Table 2-1 shows actual implementation of program memory in the PIC14000.

TABLE 2-1: IMPLEMENTATION OF PROGRAM AND CALIBRATION MEMORY IN THE PIC14000P

Area	Memory Space	Access to Memory
Program	0x000-0xFBF	PC<12:0>
Calibration	0xFC0 -0xFFF	PC<12:0>

When the PC reaches address 0xFFF, 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 0x0000, 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 2.2.

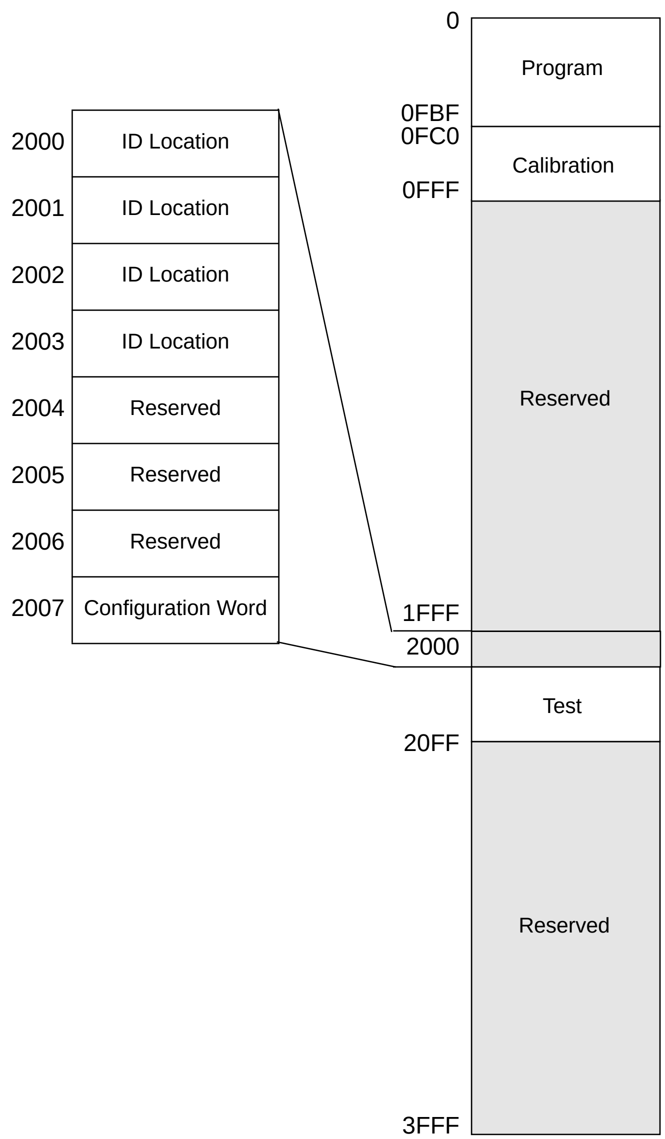
In the configuration memory space, 0x2000-0x20FF are utilized. When in configuration memory, as in the user memory, the 0x2000-0x2XFF segment is repeatedly accessed as PC exceeds 0x2XFF (see Figure 2-1).

A user may store identification information (ID) in four ID locations. The ID locations are mapped in [0x2000 : 0x2003]. All other locations are reserved and should not be programmed.

The ID locations read out normally, even after code protection. To understand how the devices behave, refer to Table 4-1.

To understand the scrambling mechanism after code protection, refer to Section 4.1.

FIGURE 2-1: PROGRAM MEMORY MAPPING



2.2 Program/Verify Mode

The program/verify mode is entered by holding pins RC6 and RC7 low while raising $\overline{MCLR}$ 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. The mode of operation is serial, and the memory that is accessed is the user program memory. RC6 and RC7 are both 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 (the $\overline{MCLR}$ pin was initially at V_{IL}). This means that all I/O are in the reset state (High impedance inputs).

Note: The $\overline{MCLR}$ pin should be raised as quickly as possible from V_{IL} to V_{IHH}. This is to ensure that the device does not have the PC incremented while in valid operation range.

2.2.1 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 (RC6) 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 RC7 is required to have a minimum setup and hold time (see AC/DC specs) 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 1µs 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 RC7 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 1µs 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 1µs is required between a command and a data word (or another command).

The commands that are available are listed in Table 2-2.

2.2.1.1 LOAD CONFIGURATION

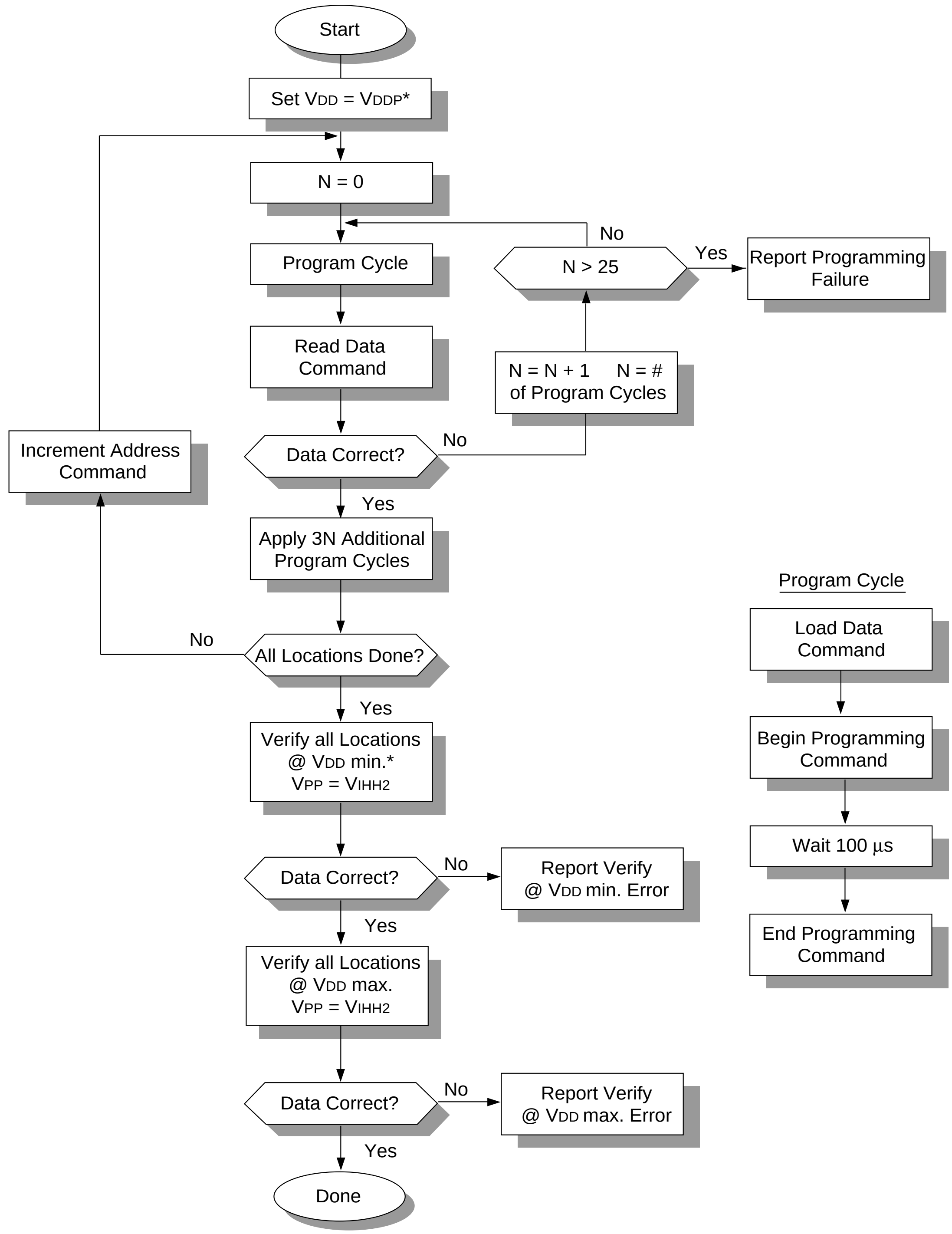
After receiving this command, the program counter (PC) will be set to 0x2000. By then applying 16 cycles to the clock pin, the chip will load 14-bits a “data word” as described above, to be programmed into the configuration memory. A description of the memory mapping schemes for normal operation and 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 (V_{IL}).

TABLE 2-2: COMMAND MAPPING

Command	Mapping (MSB ... LSB)						Data
Load Configuration	0	0	0	0	0	0	0, data(14), 0
Load Data	0	0	0	0	1	0	0, data(14), 0
Read Data	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	
End Programming	0	0	1	1	1	0	

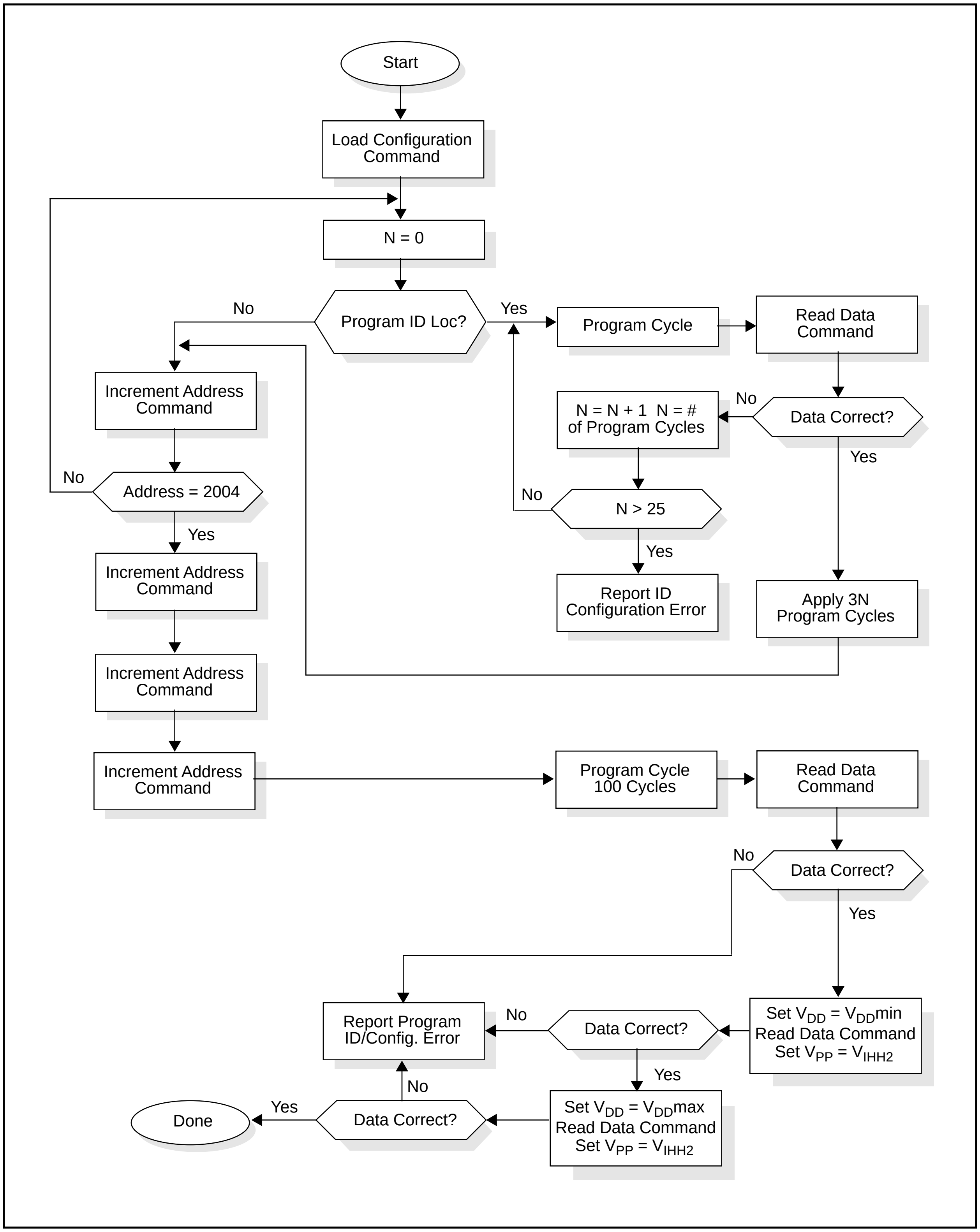
Note: The CPU clock must be disabled during in-circuit programming (to avoid incrementing the PC).

FIGURE 2-2: PROGRAM FLOW CHART - PIC14000 PROGRAM MEMORY AND CALIBRATION



* V_{DDP} = V_{DD} range for programming (typically 4.75V - 5.25V).
V_{DD}min = Minimum V_{DD} for device operation.
V_{DD}max = Maximum V_{DD} for device operation.

FIGURE 2-3: PROGRAM FLOW CHART - PIC14000 CONFIGURATION WORD & ID LOCATIONS



2.2.1.2 LOAD DATA

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.

2.2.1.3 READ DATA

After receiving this command, the chip will transmit data bits out of the memory currently accessed starting with the second rising edge of the clock input. The RC7 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.

2.2.1.4 INCREMENT ADDRESS

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

2.2.1.5 BEGIN PROGRAMMING

A load command (load configuration or load data) must be given before every begin programming command. Programming of the appropriate memory (test program memory or user program memory) will begin after this command is received and decoded. Programming should be performed with a series of 100µs programming pulses. A programming pulse is defined as the time between the begin programming command and the end programming command.

2.2.1.6 END PROGRAMMING

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

2.3 Programming Algorithm Requires Variable VDD

The PIC14000 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 (4.75 - 5.25V).

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 PIC14000 at its specified VDDmax and VDDmin levels. Since Microchip may introduce future versions of the PIC14000 with a broader VDD range, it is best that these levels are user selectable (defaults are ok).

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

The PIC14000 has several configuration bits. These bits can be programmed (reads '0') or left unprogrammed (reads '1') to select various device configurations. Figure 3-1 provides an overview of configuration bits.

FIGURE 3-1: CONFIGURATION WORD BIT MAP

Bit Number:	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIC14000	CPC	CPP1	CPP0	CPP0	CPP1	CPC	CPC	F	CPP1	CPP0	$\overline{\text{PWRTE}}$	WDTE	F	FOSC

CPP<1:0>
11: All Unprotected
10: N/A
01: N/A
00: All Protected

bit 1,6: **F** Internal trim, factory programmed. DO NOT CHANGE! Program as ‘1’. Note 1.

bit 3: **$\overline{\text{PWRTE}}$**, Power Up Timer Enable Bit
0 = Power up timer enabled
1 = Power up timer disabled (unprogrammed)

bit 2: **WDTE**, WDT Enable Bit
0 = WDT disabled
1 = WDT enabled (unprogrammed)

bit 0: **FOSC<1:0>**, Oscillator Selection Bit
0: HS oscillator (crystal/resonator)
1: Internal RC oscillator (unprogrammed)

Note 1: See Section 4.1.2 for cautions.

4.0 CODE PROTECTION

The memory space in the PIC14000 is divided into two areas: program space (0-0xFBF) and calibration space (0xFC0-0xFFF).

For program space or user space, once code protection is enabled, all protected segments read ‘0’s (or “garbage values”) and are prevented from further programming. All unprotected segments, including ID locations and configuration word, read normally. These locations can be programmed.

4.1 Calibration Space

The calibration space can contain factory-generated and programmed values. For non-JW devices, the CPC bits in the configuration word are set to ‘0’ at the factory, and the calibration data values are write-protected; they may still be read out, but not programmed. JW devices contain the factory values, but DO NOT have the CPC bits set.

Microchip does not recommend setting code protect bits in windowed devices to ‘0’. Once code-protected, the device cannot be reprogrammed.

4.1.1 CALIBRATION SPACE CHECKSUM

The data in the calibration space has its own checksum. When properly programmed, the calibration memory will always checksum to 0x0000. When this checksum is 0x0000, and the checksum of memory [0x0000:0xFBF] is 0x2FBF, the part is effectively blank, and the programmer should indicate such.

If the CPC bits are set to ‘1’, but the checksum of the calibration memory is 0x0000, the programmer should NOT program locations in the calibration memory space, even if requested to do so by the operator. This would be the case for a new JW device.

If the CPC bits are set to ‘1’, and the checksum of the calibration memory is NOT 0x0000, the programmer is allowed to program the calibration space as directed by the operator.

The calibration space contains specially coded data values used for device parameter calibration. The programmer may wish to read these values and display them for the operator’s convenience. For further information on these values and their coding, refer to AN621 (DS00621B).

4.1.2 REPROGRAMMING CALIBRATION SPACE

The operator should be allowed to read and store the data in the calibration space, for future reprogramming of the device. This procedure is necessary for reprogramming a windowed device, since the calibration data will be erased along with the rest of the memory. When saving this data, Configuration Word <1,6> must also be saved, and restored when the calibration data is reloaded.

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.

TABLE 4-1: CODE PROTECT OPTIONS

- Protect calibration memory `0XXXX00XXXXXXX`
- Protect program memory `X0000XXX00XXXX`
- No code protection `1111111X11XXXX`

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Unprotected memory segment	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Protected memory segment	Read All 0’s, Write Disabled	Read Unscrambled, Write Enabled
Protected calibration memory	Read Unscrambled, 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

4.3.1 CHECKSUM CALCULATIONS

Checksum is calculated by reading the contents of the PIC14000 memory locations and adding up the opcodes up to the maximum user addressable location, 0xFBF. Any carry bits exceeding 16-bits are neglected. Finally, the configuration word (appropriately masked) is added to the checksum. Checksum computation for the PIC14000 device 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

Code Protect	Checksum*	Blank Value	0x25E6 at 0 and max address
OFF	SUM[0000:0FBF] + CFGW & 0x3FBD	0x2FFD	0xFBCB
OFF OTP	SUM[0000:0FBF] + CFGW & 0x3FBD	0x0E7D	0xDA4B
ON	CFGW & 0x3FBD + SUM(IDs)	0x300A	0xFBD8

Legend: CFGW = Configuration Word
SUM[A:B] = [Sum of locations a through b inclusive]
SUM(ID) = ID locations masked by 0x7F then made into a 28-bit value with ID0 as the most significant byte
*Checksum = [Sum of all the individual expressions] MODULO [0xFFFF]
+ = Addition
& = Bitwise AND

5.0 PROGRAM/VERIFY MODE ELECTRICAL CHARACTERISTICS

TABLE 5-1: AC/DC CHARACTERISTICS
AC/DC Timing Requirements for Program/Verify Mode

Standard Operating Conditions
Operating Temperature: +10°C ≤ T_A ≤ +40°C, unless otherwise stated, (25°C recommended)
Operating Voltage: 4.5V ≤ V_{DD} ≤ 5.5V, unless otherwise stated.

Parameter No.	Sym.	Characteristic	Min.	Typ.	Max.	Units	Conditions
General							
PD1	V_{DDP}	Supply voltage during programming	4.75	5.0	5.25	V	
PD2	I_{DDP}	Supply current (from V_{DD}) during programming	–	–	20	mA	
PD3	V_{DDV}	Supply voltage during verify	V_{DD}min		V_{DD}max	V	Note 1
PD4	V_{IHH1}	Voltage on $\overline{MCLR}$/V_{PP} during programming	12.75	–	13.25	V	Note 2
PD5	V_{IHH2}	Voltage on $\overline{MCLR}$/V_{PP} during verify	V_{DD} + 4.0		13.5		
PD6	I_{PP}	Programming supply current (from V_{PP})	–	–	50	mA	
PD9	V_{IH1}	(RC6, RC7) input high level	0.8 V_{DD}	–	–	V	Schmitt Trigger input
PD8	V_{IL1}	(RC6, RC7) input low level	0.2 V_{DD}	–	–	V	Schmitt Trigger input

Serial Program Verify							
P1	T_R	$\overline{MCLR}$/V_{PP} rise time (V_{SS} to V_{HH}) for test mode entry	–	–	8.0	µs	
P2	Tf	$\overline{MCLR}$ Fall time	–	–	8.0	µs	
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 date out valid (during read data)	200	–	–	ns	
P8	Thld0	Hold time after $\overline{MCLR}$ ↑	2	–	–	µs	

Note 1: Program must be verified at the minimum and maximum V_{DD} limits for the part.
Note 2: V_{IHH} must be greater than V_{DD} + 4.5V to stay in programming/verify mode.

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

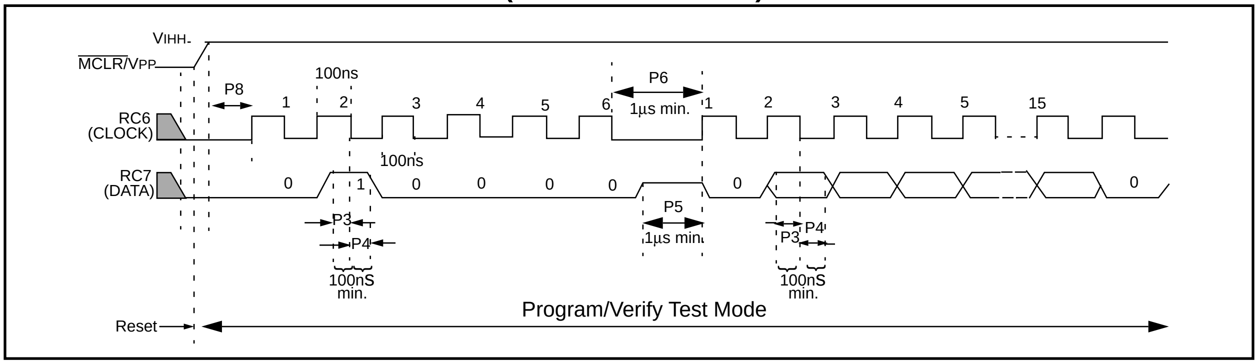


FIGURE 5-2: READ DATA COMMAND (PROGRAM/VERIFY)

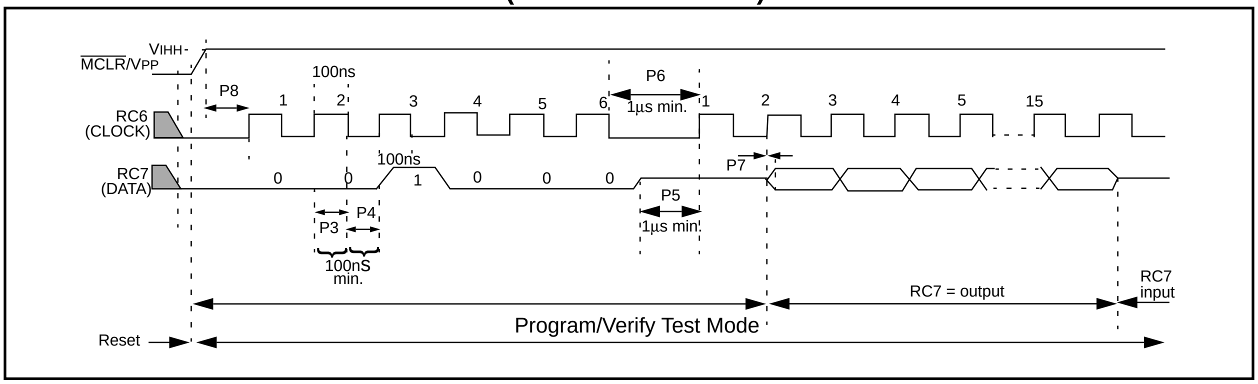
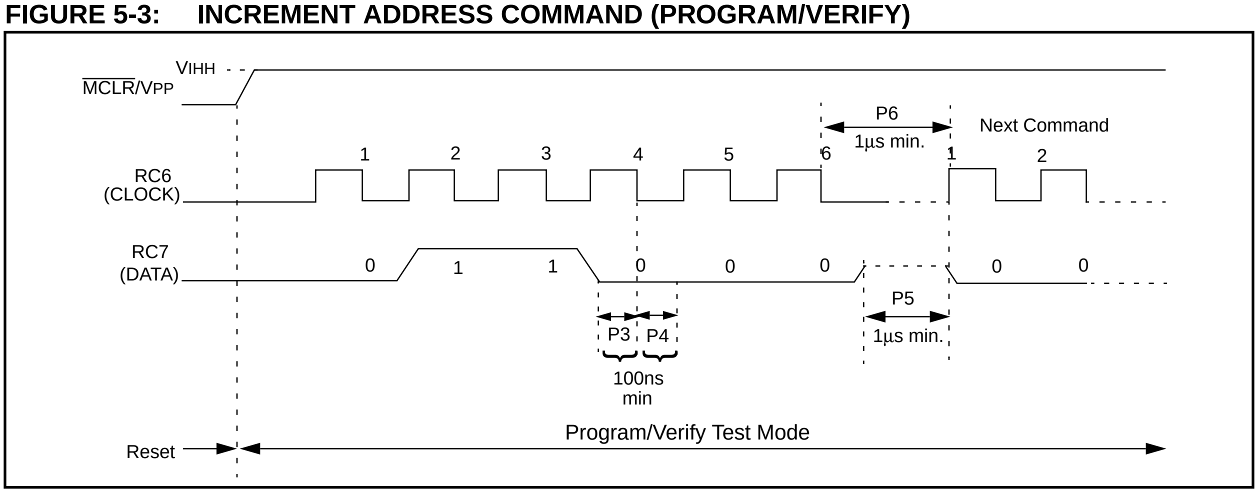


FIGURE 5-3: INCREMENT ADDRESS COMMAND (PROGRAM/VERIFY)



NOTES:

WORLDWIDE SALES & 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: 708 285-0071 Fax: 708 285-0075

Dallas
Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 214 991-7177 Fax: 214 991-8588

Dayton
Microchip Technology Inc.
Suite 150
Two Prestige Place
Miamisburg, OH 45342
Tel: 513 291-1654 Fax: 513 291-9175

Los Angeles
Microchip Technology Inc.
18201 Von Karman, Suite 1090
Irvine, CA 92715
Tel: 714 263-1888 Fax: 714 263-1338

AMERICAS *(continued)*

New York
Microchip Technology Inc.
150 Motor Parkway, Suite 416
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 Technology
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

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

Singapore
Microchip Technology
200 Middle Road
#10-03 Prime Centre
Singapore 188980
Tel: 65 334 8870 Fax: 65 334 8850

Taiwan
Microchip Technology
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2 717 7175 Fax: 886 2 545 0139

EUROPE

United Kingdom
Arizona Microchip Technology Ltd.
Unit 6, The Courtyard
Meadow Bank, Furlong Road
Bourne End, Buckinghamshire SL8 5AJ
Tel: 44 1 628 850303 Fax: 44 1 628 850178

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 Muenchen, 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 39 689 9939 Fax: 39 39 689 9883

JAPAN

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shin Yokohama
Kohoku-Ku, Yokohama
Kanagawa 222 Japan
Tel: 81 45 471 6166 Fax: 81 45 471 6122

5/10/96