PIC16F62X

PIC16F62X EEPROM Memory Programming Specification

This document includes the programming specifications for the following devices:

- PIC16F627
- PIC16F628
- PIC16LF627
- PIC16LF628

Note:	All references to PIC16F62X also apply to PIC16LF62X.

1.0 PROGRAMMING THE PIC16F62X

The PIC16F62X is programmed using a serial method. The Serial mode will allow the PIC16F62X to be programmed while in the users system. This allows for increased design flexibility. This programming specification applies to PIC16F62X devices in all packages.

PIC16F62X devices may be programmed using a single +5 volt supply (Low Voltage Programming mode).

1.1 Hardware Requirements

The PIC16F62X requires one programmable power supply for VDD (4.5V to 5.5V) and a VPP of 12V to 14V, or VPP of 4.5V to 5.5V, when using low voltage. Both supplies should have a minimum resolution of 0.25V.

1.2 Programming Algorithm Requires Variable VDD

The PIC16F62X uses an intelligent algorithm. The algorithm calls for program verification at VDDMIN as well as VDDMAX. Verification at VDDMIN ensures good "erase margin". Verification at VDDMAX ensures good "program margin".

The actual programming must be done with VDD in the VDDP range.

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 PIC16F62X is at its specified VDDMAX and VDDMIN levels. Since Microchip may introduce future versions of the PIC16F62X 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 a "prototype" or "development" programmer, not a "production" quality programmer.

1.3 Programming Mode

The Programming mode for the PIC16F62X allows programming of user program memory, data memory, special locations used for ID, and the configuration word.

Pin Diagram

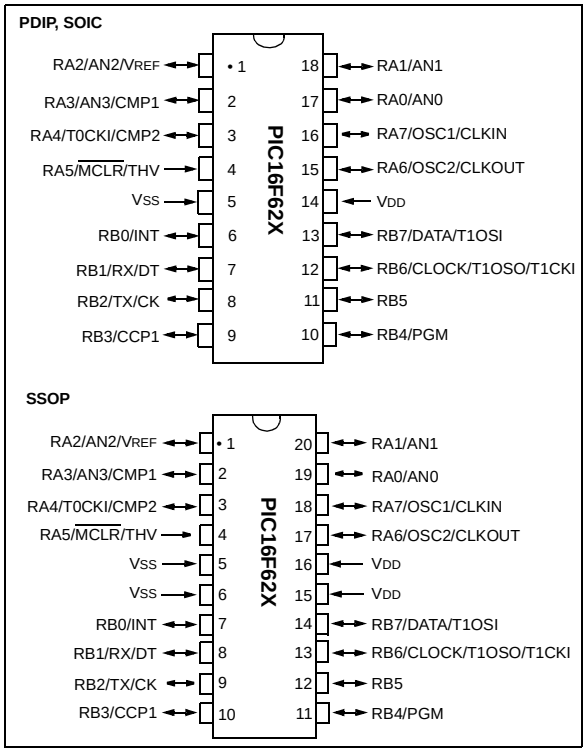


TABLE 1-1: PIN DESCRIPTIONS (DURING PROGRAMMING): PIC16F62X

Pin Name	During Programming		
	Function	Pin Type	Pin Description
RB4	PGM	I	Low Voltage Programming input if configuration bit equals 1
RB6	CLOCK	I	Clock input
RB7	DATA	I/O	Data input/output
$\overline{MCLR}$	Programming Mode	P*	Program Mode Select
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground

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

* In the PIC16F62X, the programming high voltage is internally generated. To activate the Programming mode, high voltage needs to be applied to $\overline{MCLR}$ input. Since the $\overline{MCLR}$ is used for a level source, this means that $\overline{MCLR}$ does not draw any significant current.

2.0 PROGRAM DETAILS

2.1 User Program Memory Map

The user memory space extends from 0x0000 to 0x1FFF. In Programming mode, the program memory space extends from 0x0000 to 0x3FFF, with the first half (0x0000-0x1FFF) being user program memory and the second half (0x2000-0x3FFF) being configuration memory. The PC will increment from 0x0000 to 0x1FFF and wrap to 0x000, 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 re-enter 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).

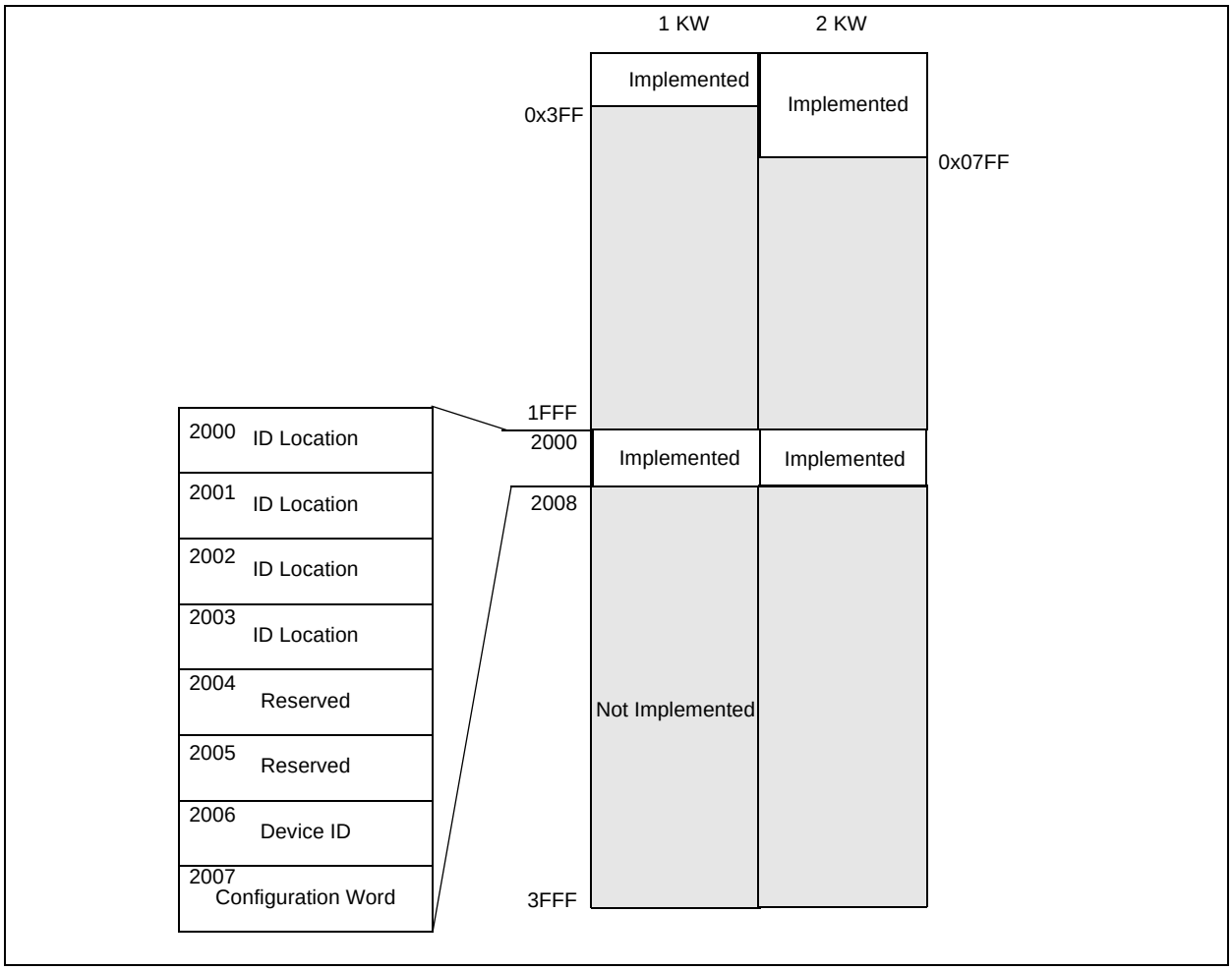
2.2 User ID Locations

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

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

2: Only the low order 4 bits of the User ID locations may be included in the device checksum. See Section 3.1 for checksum calculation details.

FIGURE 2-1: PROGRAM MEMORY MAPPING



2.3 Program/Verify Mode

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

The signal on the data pin is required to have a minimum setup and hold time (see AC/DC specifications), with respect to the falling edge of the clock. Commands that have data associated with them (read and load), require a minimum delay of Tdly1 between the command and the data.

The 6-bit command sequences are shown in Table 2-1.

TABLE 2-1: COMMAND MAPPING FOR PIC16F627/PIC16F628

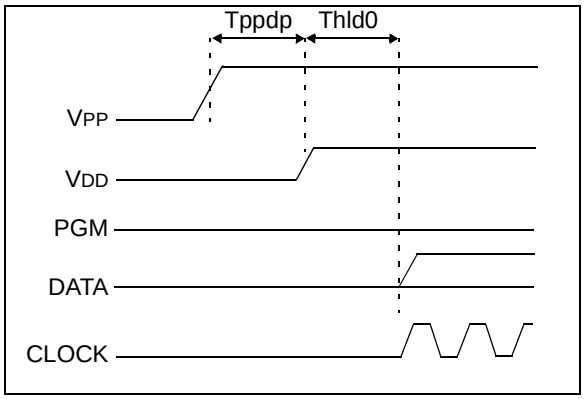
Command	Mapping (MSb … LSb)						Data
Load Configuration	X	X	0	0	0	0	0, data (14), 0
Load Data for Program Memory	X	X	0	0	1	0	0, data (14), 0
Load Data for Data Memory	X	X	0	0	1	1	0, data (8), zero (6), 0
Increment Address	X	X	0	1	1	0	
Read Data from Program Memory	X	X	0	1	0	0	0, data (14), 0
Read Data from Data Memory	X	X	0	1	0	1	0, data (8), zero (6), 0
Begin Erase/Programming Cycle	0	0	1	0	0	0	
Begin Programming Only Cycle	0	1	1	0	0	0	
Bulk Erase Program Memory	X	X	1	0	0	1	
Bulk Erase Data Memory	X	X	1	0	1	1	
Bulk Erase Setup 1	0	0	0	0	0	1	
Bulk Erase Setup 2	0	0	0	1	1	1	

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

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

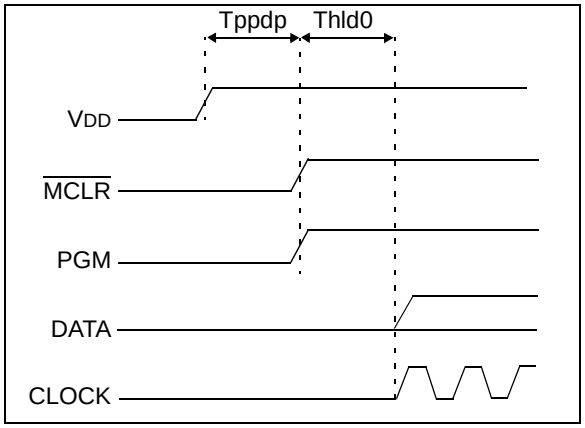
All other logic is held in the RESET state while in Program/Verify mode. This means that all I/O are in the RESET state (high impedance inputs).

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



Note:	PGM should be held low to prevent inadvertent entry into LVP mode.

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

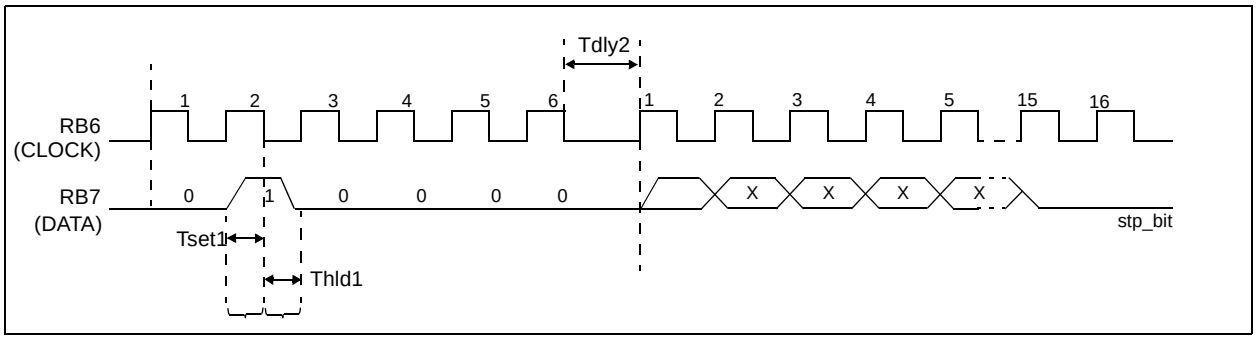


Note:	If the device is in LVP mode, raising VPP to VIHH does not override LVP mode.

2.3.1 LOAD DATA FOR PROGRAM MEMORY

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

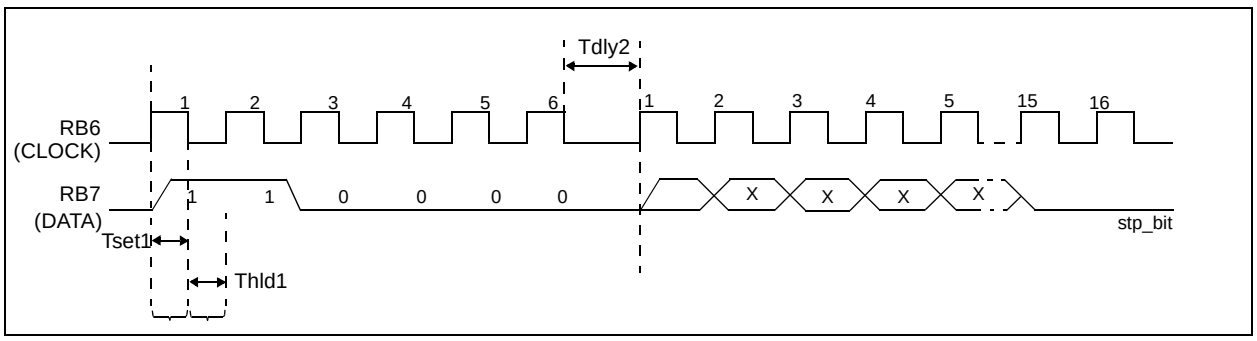
FIGURE 2-4: LOAD DATA COMMAND FOR PROGRAM MEMORY



2.3.2 LOAD DATA FOR DATA MEMORY

Load data for data memory receives an 8-bit byte, and readies it to be programmed into data memory at location specified by the lower 7 bits of the PC. Though the data byte is only 8-bits wide, all 16 clock cycles are required to allow the programming module to reset properly.

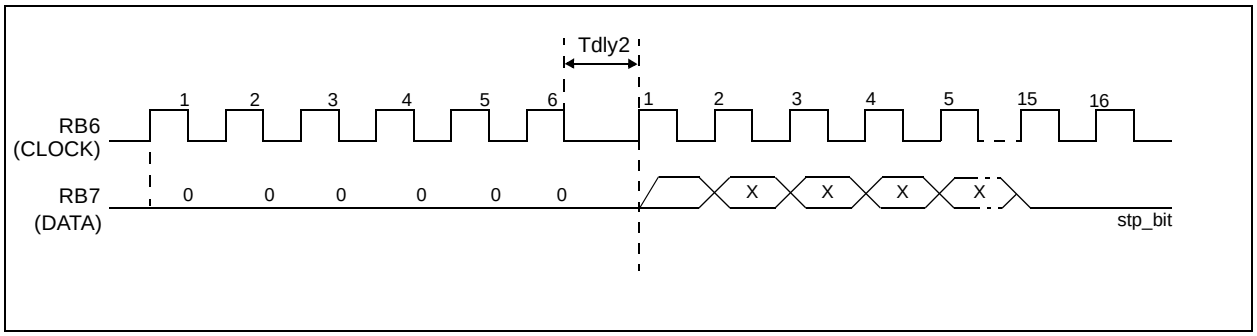
FIGURE 2-5: LOAD DATA COMMAND FOR DATA MEMORY



2.3.3 LOAD DATA FOR CONFIGURATION MEMORY

The load configuration command advances the PC to the start of configuration memory (0x2000-0x200F). Once it is set to the configuration region, only exiting and re-entering Program/Verify mode will reset PC to the user memory space (see Figure 2-6).

FIGURE 2-6: LOAD CONFIGURATION

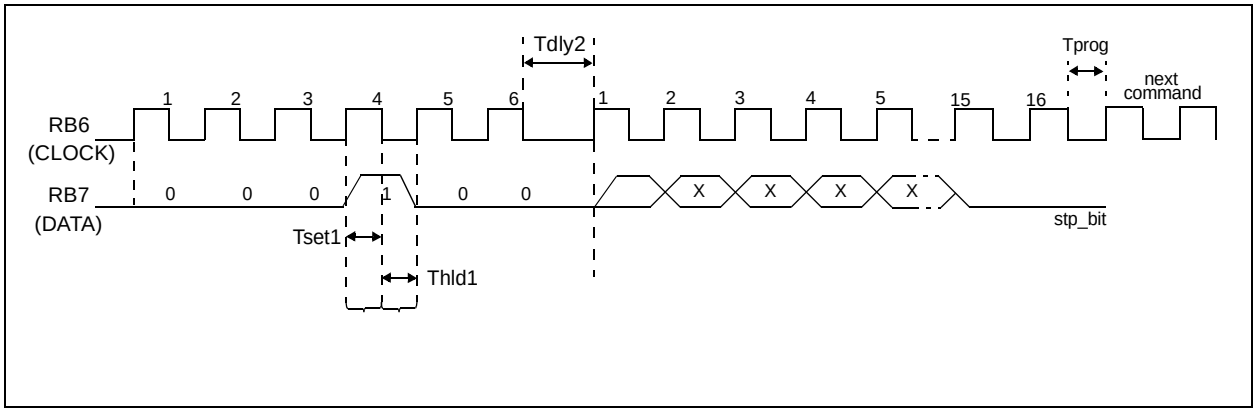


2.3.4 BEGIN PROGRAMMING ONLY CYCLE

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

This command is similar to the Erase/Program 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.

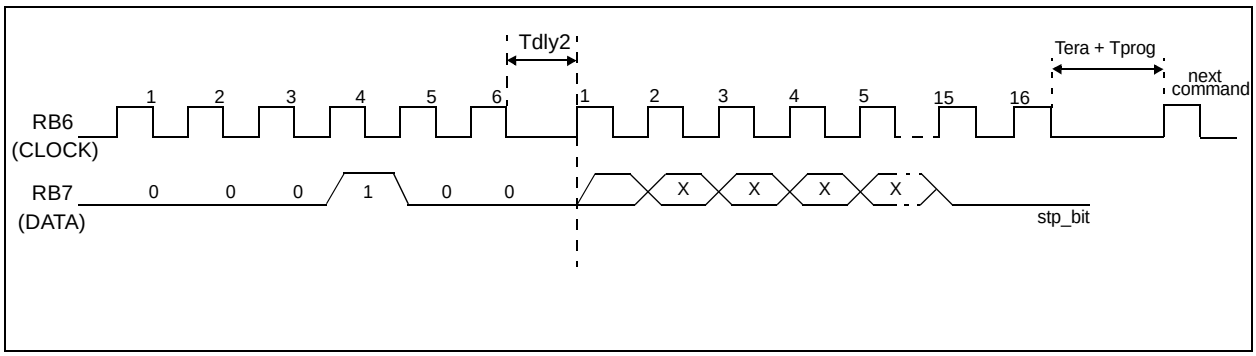
FIGURE 2-7: BEGIN PROGRAMMING ONLY CYCLE



2.3.5 BEGIN ERASE/PROGRAMMING CYCLE

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

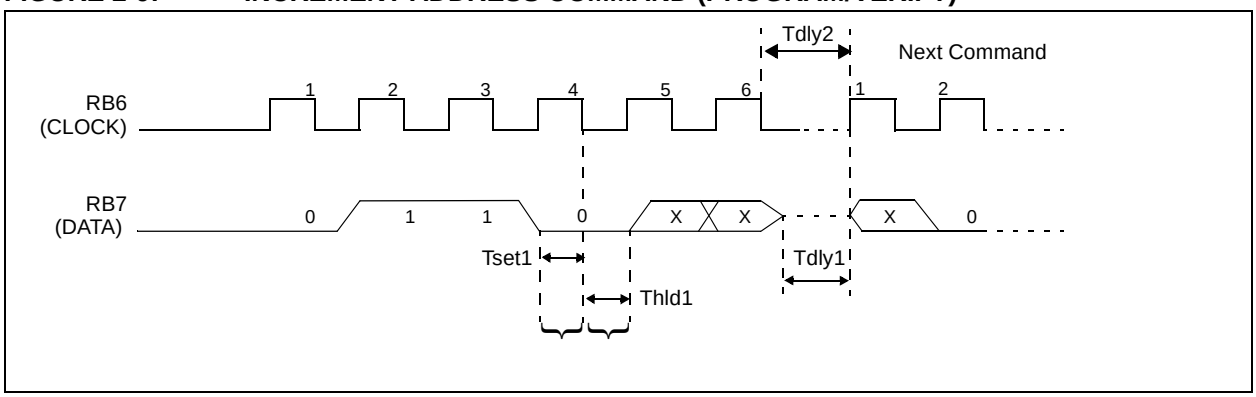
FIGURE 2-8: BEGIN ERASE/PROGRAMMING CYCLE



2.3.6 INCREMENT ADDRESS

The PC is incremented when this command is received. See Figure 2-9.

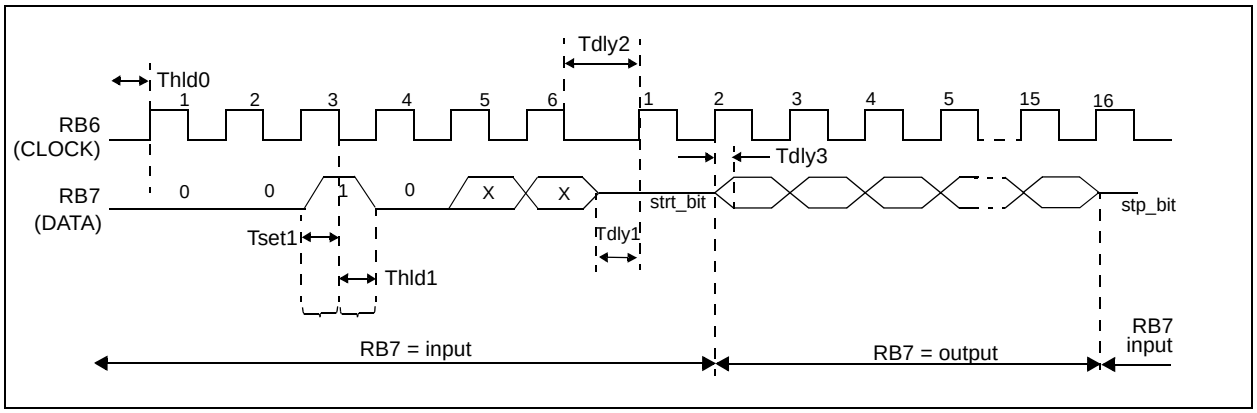
FIGURE 2-9: INCREMENT ADDRESS COMMAND (PROGRAM/VERIFY)



2.3.7 READ DATA FROM PROGRAM MEMORY

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

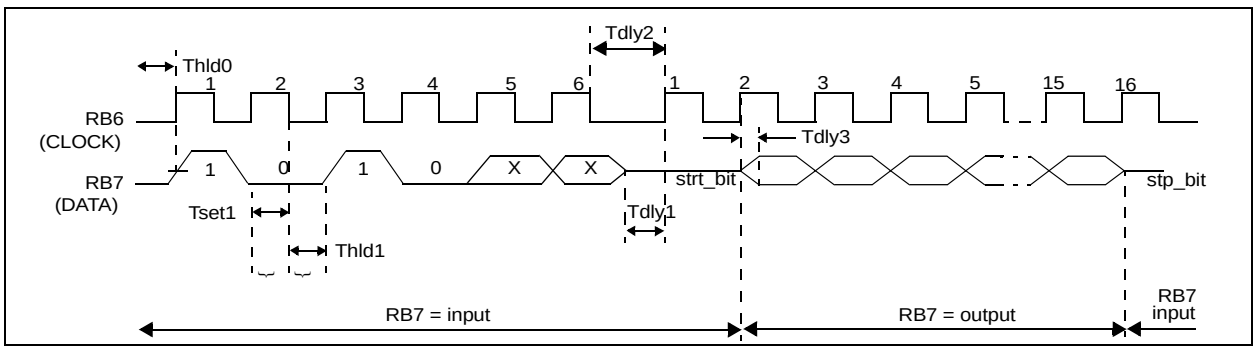
FIGURE 2-10: READ DATA FROM PROGRAM MEMORY



2.3.8 READ DATA FROM DATA MEMORY

Read data from data memory reads the byte in data memory addressed by the low order 7 bits of PC and transmits it on the data pin during the data phase of the command. The data pin will go into Output mode on the second rising clock edge, and revert back to input moved (hi-impedance) after the 16th rising edge. As only 8 bits are transmitted, the last 8 bits are zero padded.

FIGURE 2-11: READ DATA FROM DATA MEMORY



2.3.9 BULK ERASE SETUP 1 AND BULK ERASE SETUP 2

These commands are used in conjunction to reset the configuration word. See Section 3.3 for details on how to reset the configuration word.

FIGURE 2-12: BULK ERASE SETUP 1

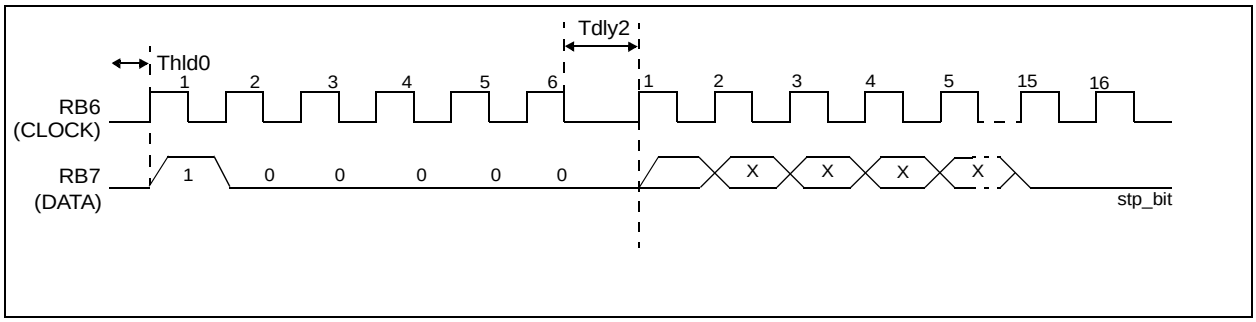
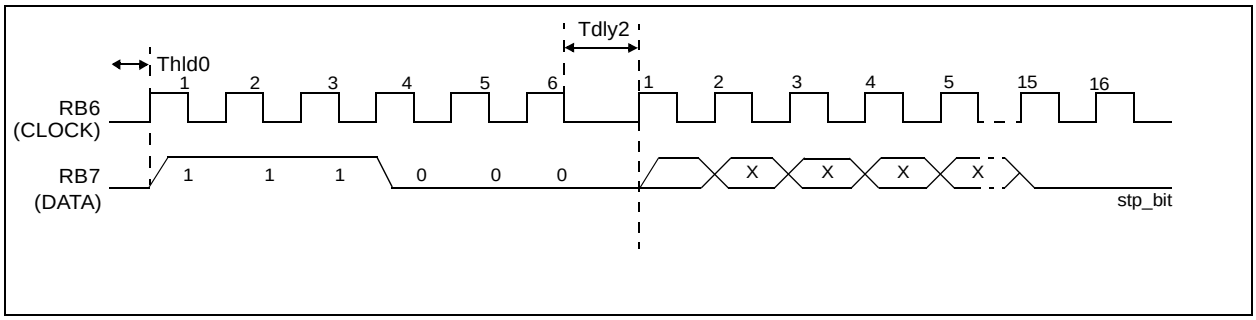


FIGURE 2-13: BULK ERASE SETUP 2



3.0 COMMON PROGRAMMING TASKS

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

3.1 Bulk Erase Program Memory

If the device is not code protected, the program memory can be erased with the Bulk Erase Program Memory command. See Section 3.4 for removing code protection if it is set.

Note:	All bulk erase operations must take place with VDD between 4.5-5.5V.

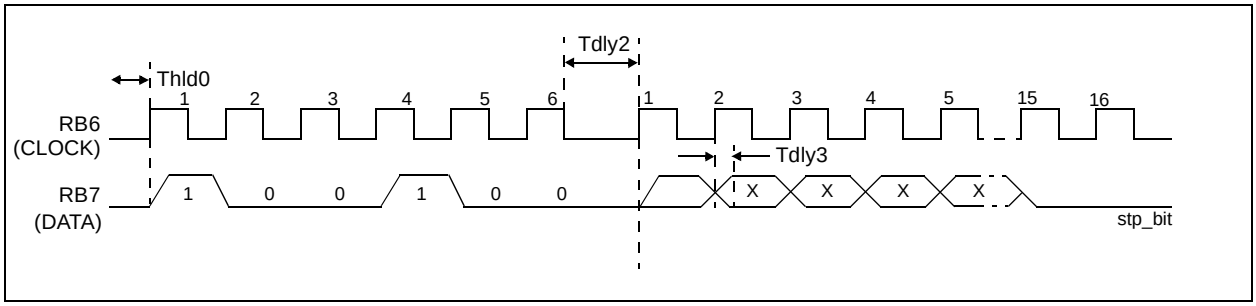
To perform a bulk erase of the program memory, the following sequence must be performed:

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

If the address is pointing to the ID/configuration word memory (0x2000-0x200F), then both ID locations and program memory will be erased. However, the configuration word will not be cleared by this method.

Note:	If the device is code protected, the Bulk Erase command will not work.

FIGURE 3-1: BULK ERASE PROGRAM MEMORY



3.2 Bulk Erase Data Memory

If the device is not data protected, the program memory can be erased with the Bulk Erase Data Memory command. See Section 3.3 for removing code protection, if it is not set.

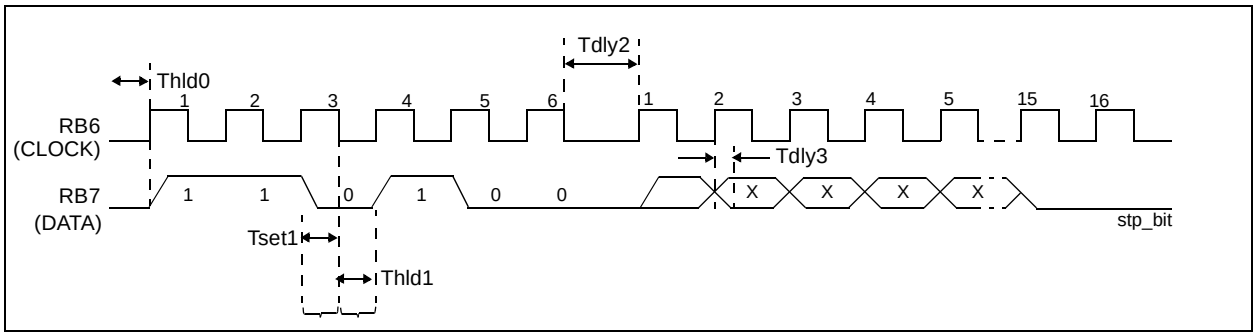
Note:	All bulk erase operations must take place with V_{DD} between 4.5-5.5V

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

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

Note:	If the device is code protected, the Bulk Erase command will not work.

FIGURE 3-2: BULK ERASE DATA MEMORY COMMAND



3.3 Disabling Code Protection

Once the device has been code protected, the code protected regions of program memory read out as zeros and the device may no longer be written until the following process has been completed. The Bulk Erase commands will not erase the device. Instead, the following procedure, to reset the code protection bits, must be used. Resetting the Code Protection bits will also erase Program, Data and Configuration memory, thus maintaining security of the code and data.

1. Execute a Load Configuration command (data word 0x0000) to set PC to 0x2000.
2. Execute Increment Address command 7 times to advance PC to 0x2007.
3. Execute Bulk Erase Setup 1 command.
4. Execute Bulk Erase Setup 2 command.
5. Execute Begin Erase Programming command.
6. Wait Tera + Tprog.
7. Execute Bulk Erase Setup 1 command.
8. Execute Bulk Erase Setup 2 command.

3.4 Programming Program Memory

FIGURE 3-3: PROGRAM FLOW CHART - PIC16F62X PROGRAM MEMORY

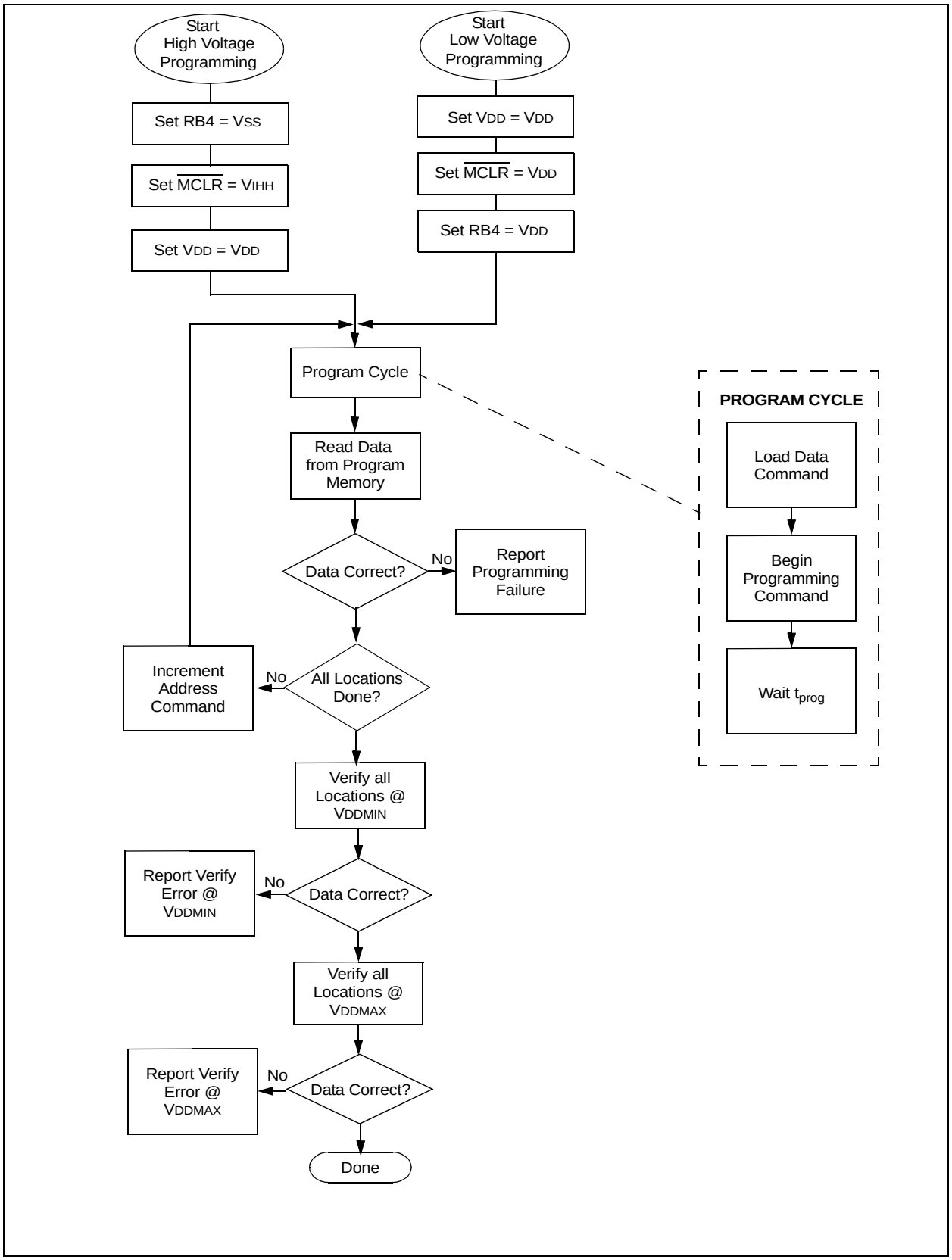
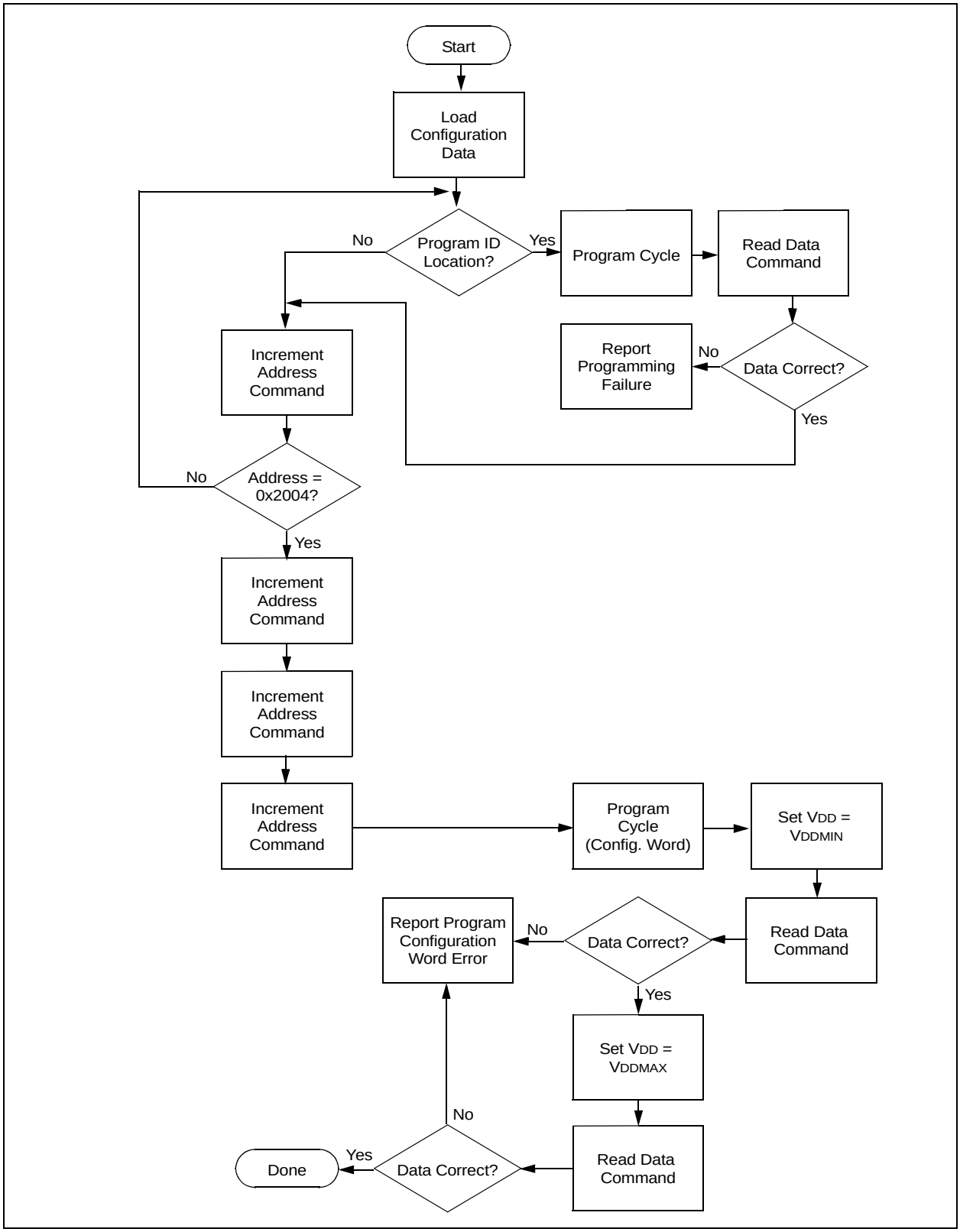
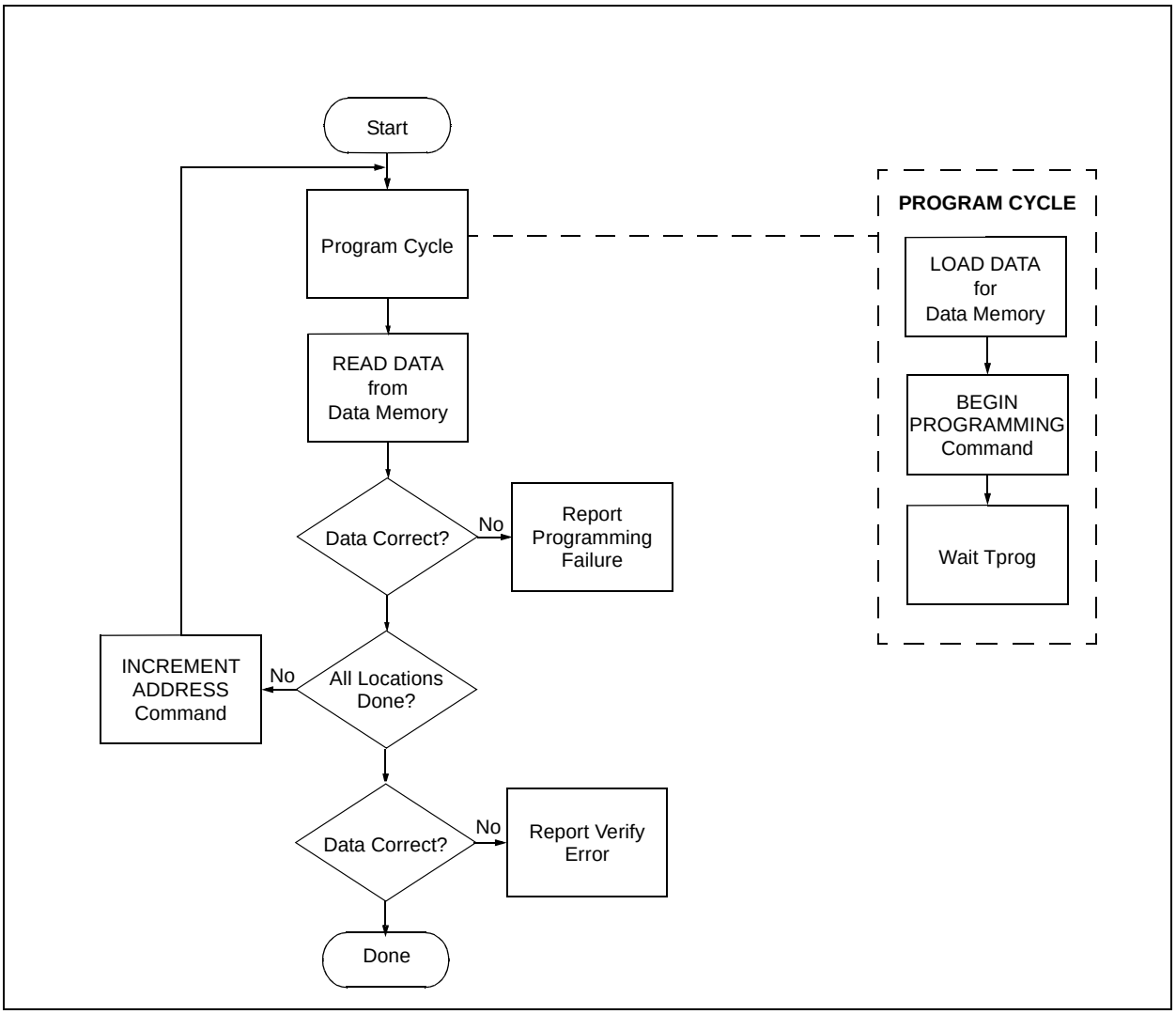


FIGURE 3-4: PROGRAM FLOW CHART - PIC16F62X CONFIGURATION MEMORY



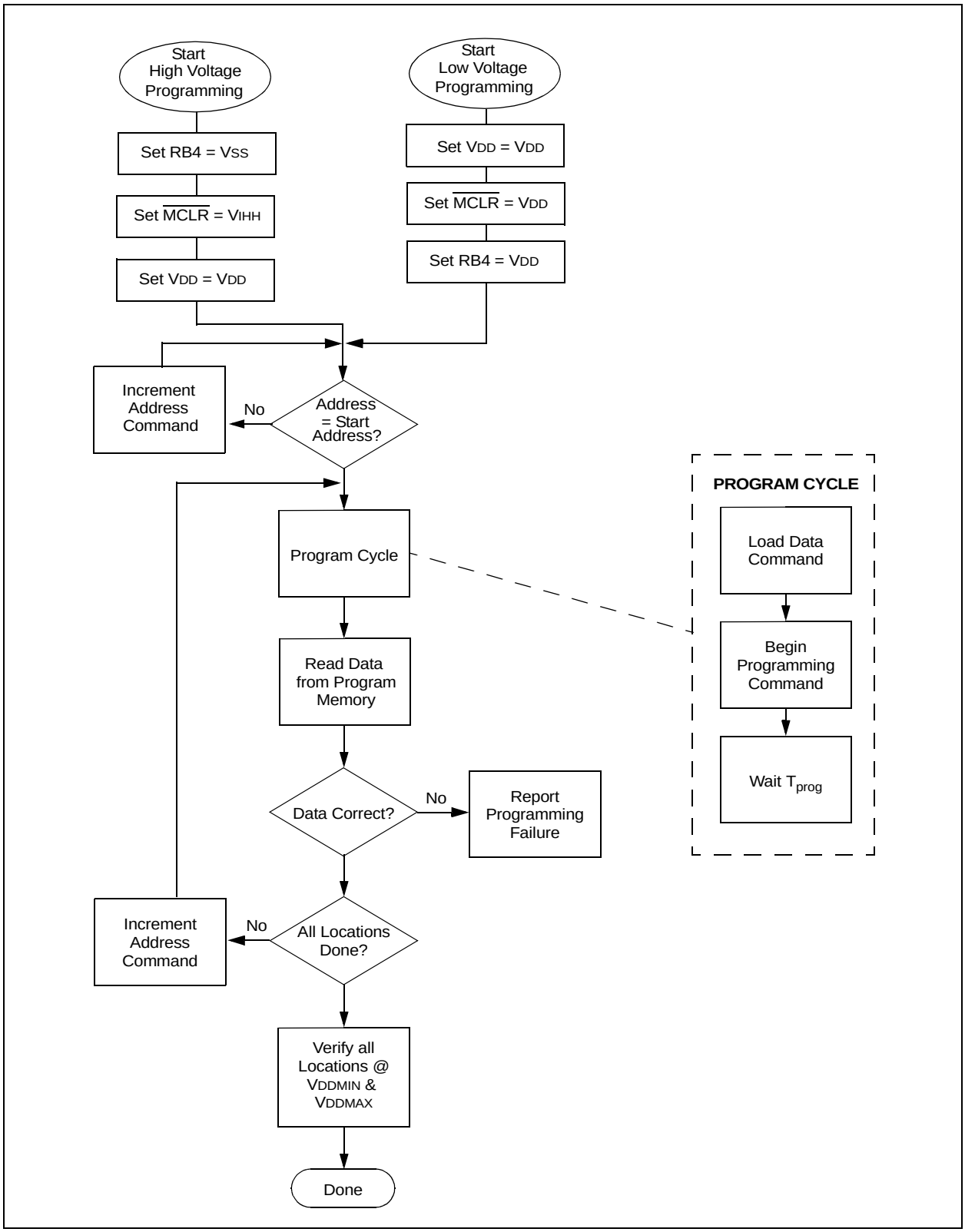
3.5 Program Data Memory

FIGURE 3-5: PROGRAM FLOW CHART - PIC16F62X DATA MEMORY



3.6 Programming Range of Program Memory

FIGURE 3-6: PROGRAM FLOW CHART - PIC16F62X PROGRAM MEMORY



3.7 Configuration Word

The PIC16F62X has several configuration bits. These bits can be set (reads '0'), or left unchanged (reads '1'), to select various device configurations.

3.8 Device ID Word

The device ID word for the PIC16F62X is hard coded at 2006h.

TABLE 3-1: DEVICE ID VALUES

Device	Device ID Value	
	Dev	Rev
PIC16F627	00 0111 101	x xxxx
PIC16F628	00 0111 110	x xxxx

REGISTER 3-1: CONFIGURATION WORD FOR PIC16F627/628 (ADDRESS: 2007h)

CP1	CP0	CP1	CP0	—	CPD	LVP	BOREN	MCLRE	FOSC2	$\overline{\text{PWRTEN}}$	WDTEN	F0SC1	F0SC0

bit 13 bit 0

bit 13-10 **CP1:CP0:** Code Protection bits (2)
Code protection for 2K program memory
11 = Program memory code protection off
10 = 0400h-07FFh code protected
01 = 0200h-03FFh code protected
00 = 0000h-07FFh code protected
Code protection for 1K program memory
11 = Program memory code protection off
10 = Program memory code protection off
01 = 0200h-03FFh code protected
00 = 0000h-03FFh code protected

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

bit 8 **CPD:** Data Code Protection bit (3)
1 = Data memory code protection off
0 = Data memory code protected

bit 7 **LVP**: Low Voltage Programming Enable bit
1 = RB4/PGM pin has PGM function, Low Voltage Programming enabled
0 = RB4/PGM is digital input, HV on $\overline{\text{MCLR}}$ must be used for programming

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

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

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

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

bit 4, 1-0 **FOSC2:FOSC0**: Oscillator Selection bits (4)
111 = ER oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, Resistor on RA7/OSC1/CLKIN
110 = ER oscillator: I/O function on RA6/OSC2/CLKOUT pin, Resistor on RA7/OSC1/CLKIN
101 = INTRC oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN
100 = INTRC oscillator: I/O function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN
011 = ExtClk: I/O function on RA6/OSC2/CLKOUT pin, CLKIN on RA7/OSC1/CLKIN
010 = HS oscillator: High speed crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN
001 = XT oscillator: Crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN
000 = LP oscillator: Low power crystal on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN

Note 1: Enabling Brown-out Detect Reset automatically enables Power-up Timer (PWRT) regardless of the value of bit $\overline{\text{PWRTEN}}$. Ensure the Power-up Timer is enabled any time Brown-out Reset is enabled.
2: All of the CP1:CP0 pairs have to be given the same value to enable the code protection scheme listed. The entire program EEPROM will be erased if the code protection is reset.
3: The entire data EEPROM will be erased when the code protection is turned off. The calibration memory is not erased.
4: When $\overline{\text{MCLR}}$ is asserted in INTRC or ER mode, the internal clock oscillator is disabled.

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

3.9 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 PIC16F62X, the EEPROM data memory should also be embedded in the HEX file (see Section 4.1).

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

3.10 Checksum Computation

3.10.1 CHECKSUM

Checksum is calculated by reading the contents of the PIC16F62X memory locations and adding up the opcodes up to the maximum user addressable location (e.g., 0x7FF for the PIC16F628). 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 PIC16F62X devices is shown in Table 3-2.

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: Some older devices have an additional value added in the checksum. This is to maintain compatibility with older device programmer checksums.

TABLE 3-2: CHECKSUM COMPUTATION

Device	Code Protect	Checksum*	Blank Value	0x25E6 at 0 and Max Address
PIC16F627	OFF	SUM[0x0000:0x3FFF] + CFGW & 0x3DFF	0x39FF	0x05CD
	0x200 : 0x3FF	SUM[0x0000:0x01FF] + CFGW & 0x3DFF + SUM_ID	0x4DFE	0xFFB3
	ALL	CFGW & 0x3DFF + SUM_ID	0x3BFE	0x07CC
PIC16F628	OFF	SUM[0x0000:0x07FF] + CFGW & 0x3DFF	0x35FF	0x01CD
	0x400 : 0x7FF	SUM[0x0000:0x03FF] + CFGW & 0x3DFF +SUM_ID	0x5BFE	0x0DB3
	0x200 : 0x7FF	SUM[0x0000:0x01FF] + CFGW & 0x3DFF + SUM_ID	0x49FE	0xFBB3
	ALL	CFGW & 0x3DFF + SUM_ID	0x37FE	0x03CC

Legend: CFGW = Configuration Word
SUM[a:b] = [Sum of locations a to b inclusive]
SUM_ID = ID locations masked by 0xF then made into a 16-bit value with ID0 as the most significant nibble.
For example, ID0 = 0x1, ID1 = 0x2, ID3 = 0x3, ID4 = 0x4, then SUM_ID = 0x1234
*Checksum = [Sum of all the individual expressions] **MODULO** [0xFFFF]
+ = Addition
& = Bitwise AND

4.0 PROGRAM/VERIFY MODE ELECTRICAL CHARACTERISTICS

4.1 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 128 data memory locations are logically mapped starting at address 0x2100. The format for data memory storage is one data byte per address location, LSB aligned.

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

AC/DC Characteristics	**Standard Operating Conditions (unless otherwise stated)** Operating Temperature: 0°C ≤ TA ≤ +70°C Operating Voltage: 4.5V ≤ VDD ≤ 5.5V					
Characteristics	**Sym**	**Min**	**Typ**	**Max**	**Units**	**Conditions/Comments**
General						
VDD level for word operations, program memory	VDD	2.0		5.5	V	
VDD level for word operations, data memory	VDD	2.0		5.5	V	
VDD level for bulk erase/write operations, program and data memory	VDD	4.5		5.5	V	
High voltage on $\overline{MCLR}$ and RA4/T0CKI for Programming mode entry	VIHH	VDD + 3.5		13.5	V	
$\overline{MCLR}$ rise time (VSS to VIHH) for Programming mode entry	TVHHR			1.0	µs	
Hold time after VPP↑	Tppdp	5			µs	
(CLOCK, DATA) input high level	VIH1	0.8 VDD			V	Schmitt Trigger input
(CLOCK, DATA) input low level	VIL1			0.2 VDD	V	Schmitt Trigger input
CLOCK, DATA setup time before $\overline{MCLR}$↑	Tset0	100			ns	
CLOCK, DATA hold time after $\overline{MCLR}$↑	Thld0	5			µs	
Serial Program/Verify						
Data in setup time before clock↓	Tset1	100			ns	
Data in hold time after clock↓	Thld1	100			ns	
Data input not driven to next clock input (delay required between command/data or command/command)	Tdly1	1.0			µs	
Delay between clock↓ to clock↑ of next command or data	Tdly2	1.0			µs	
Clock↑ to data out valid (during read data)	Tdly3			80	ns	
Erase cycle time	Tera		2	5	ms	
Programming cycle time	Tprog		4	8	ms	
Time delay from program to compare (HV discharge time)	Tdis	0.5			µs	

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.

Trademarks

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 and Mountain View, California in March 2002. The Company’s quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog 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: 480-792-4338

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

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: 86-28-86766200 Fax: 86-28-86766599

China - Fuzhou
Microchip Technology Consulting (Shanghai) Co., Ltd., Fuzhou Liaison Office
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506 Fax: 86-591-7503521

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

China - Hong Kong SAR
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: 65-6334-8870 Fax: 65-6334-8850

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

EUROPE

Austria
Microchip Technology Austria GmbH
Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

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

05/16/02