

## EPROM Memory Programming Specification

This document includes the programming specifications for the following devices:

- PIC16C642
- PIC16C662

### 1.0 PROGRAMMING THE PIC16C64X/66X

The PIC16C64X/66X can be programmed using a serial method. In serial mode, the PIC16C64X/66X can be programmed while in the users system. This allows for increased design flexibility. This programming specification applies to PIC16C64X/66X devices in all packages.

#### 1.1 Hardware Requirements

The PIC16C64X/66X requires two programmable power supplies, one for VDD (2.0V to 6.5V recommended) and one for VPP (12V to 14V). Both supplies should have a minimum resolution of 0.25V.

#### 1.2 Programming Mode

The programming mode for the PIC16C64X/66X allows programming of user program memory, special locations used for ID, and the configuration word for the PIC16C64X/66X.

### Pin Diagrams

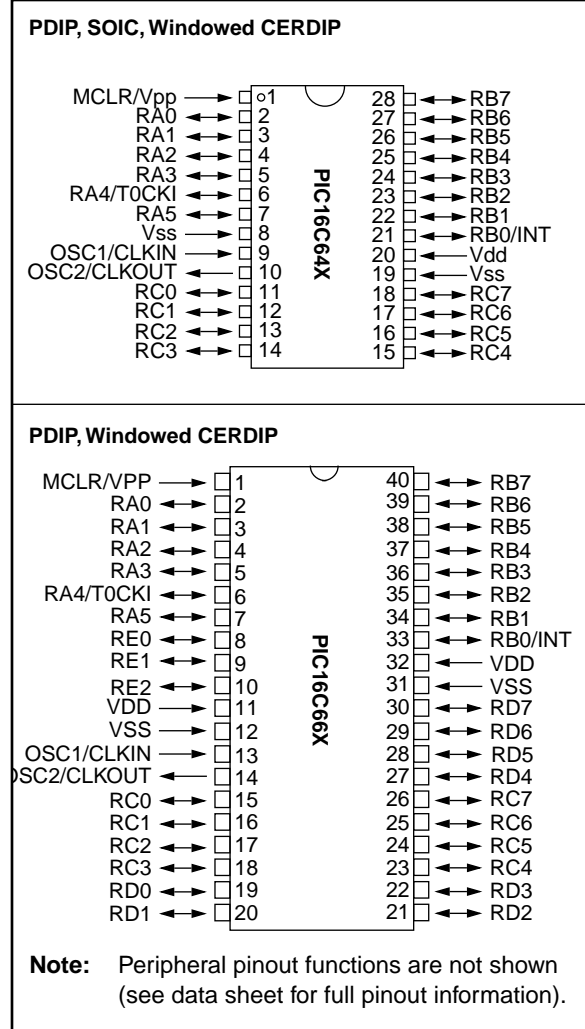


TABLE 1-1: PIN DESCRIPTIONS (DURING PROGRAMMING): PIC16C64X/66X

Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
RB6	CLOCK	I	Clock input
RB7	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

# PIC16C64X/66X

## 2.0 PROGRAM MODE ENTRY

### 2.1 User Program Memory Map

The user memory space extends from 0x0000 to 0x1FFF (8K). Table shows actual implementation of program memory in the PIC16C64X/66X microcontroller family.

When the PC reaches the last location of the implemented program memory, it will wrap around and address a location within the physically implemented memory (Figure 2-1).

In Programming Mode, the program memory space extends from 0x0000 to 0x3FFF, with the first half (0x0000-0x1FFF) being user program memory and the second half (0x2000-0x3FFF) being configuration memory. The PC will increment from 0x0000 to 0x1FFF and wrap to 0x000 or 0x2000 to 0x3FFF and wrap around to 0x2000 (not to 0x0000). Once in configuration memory, the highest bit of the PC stays a '1', thus always pointing to the configuration memory. The only way to point to user program memory is to reset the part and reenter program/verify mode, as described in Section 2.3.

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

**TABLE 2-1: IMPLEMENTATION OF PROGRAM MEMORY IN THE PIC16C64X/66X**

Device	Program Memory Size
PIC16C642	0x000 -0xFFFF (4K)
PIC16C662	0x000 -0xFFFF (4K)

A user may store identification information (ID) in four ID locations. The ID locations are mapped in [0x2000: 0x2003]. It is recommended that the user use only the six least significant bits of each ID location where the least two significant bits are the parity bits. In some devices, the ID locations read-out in a scrambled fashion after code protection is enabled. For these devices, it is recommended that ID location is written as

“11 1111 1666 bbbb pp”

where 'bbbb' is ID information.

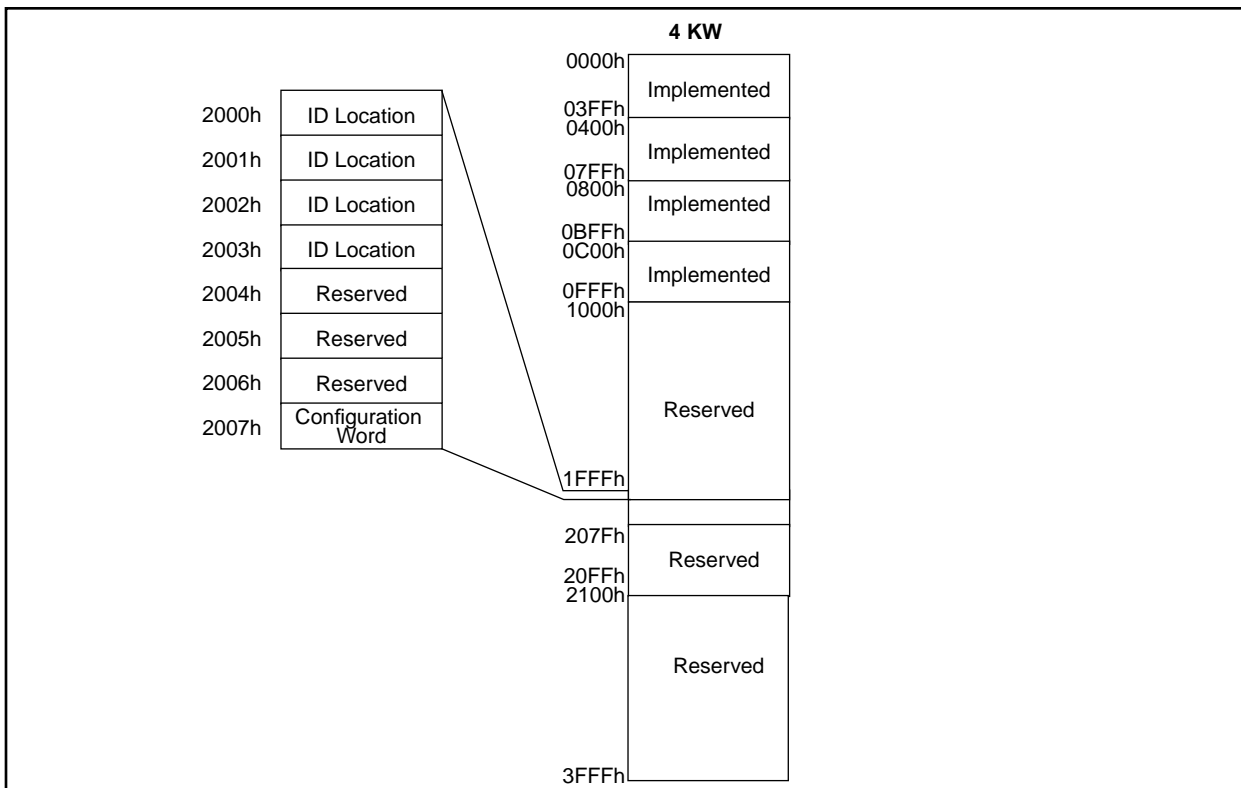
**Note 1:** All other locations are reserved and should not be programmed.

**2:** Parity bits must be clocked in/out for identification, but are ignored and not checked.

In other devices, 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**



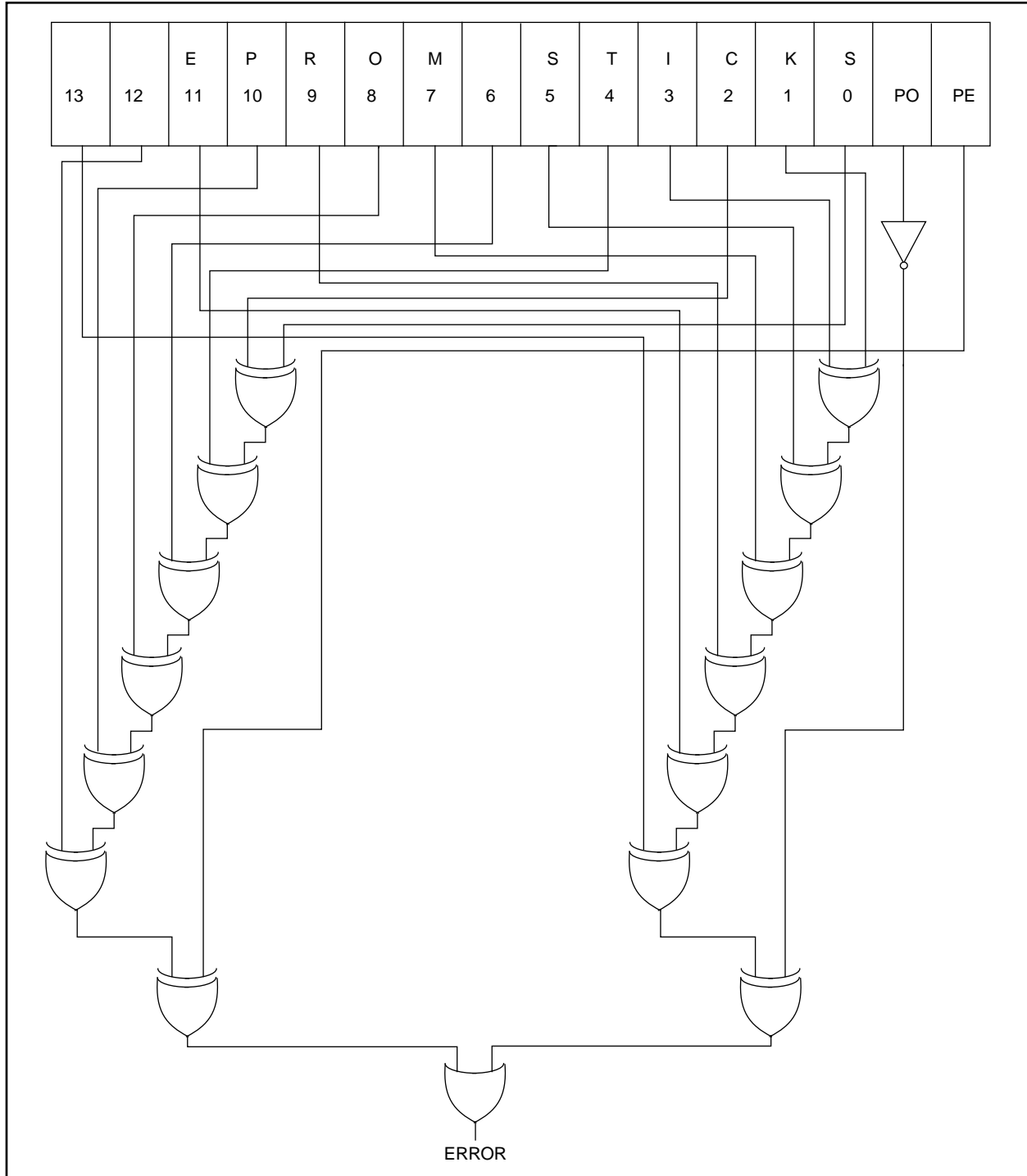
## 2.2 Program Memory Parity

The PIC16C66X has on-chip parity bits that can be used to verify the contents of the program memory during run time. Parity bits may be useful in applications in order to increase overall reliability of the system.

Due to the on-chip parity bits the entire program memory word has been enlarged to 16 bits.

The user is responsible to generate and program the correct parity for a given program memory word. The two parity bits are computed on alternating bits of the program word. One computation is performed using even parity, the other using odd parity as shown in Figure 2-2.

**FIGURE 2-2: EPROM MEMORY WITH PARITY CHECKING**



# PIC16C64X/66X

## 2.3 Program/Verify Mode

The Program/Verify mode is entered by holding pins RB6 and RB7 low while raising  $\overline{\text{MCLR}}$  pin from  $V_{\text{IL}}$  to  $V_{\text{IH}}$  (high voltage). Once in this mode the user program memory and the configuration memory can be accessed and programmed in serial fashion. The mode of operation is serial, and the memory that is accessed is the user program memory. RB6 is a Schmitt Trigger input in this mode.

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

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

### 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 6 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 of 100 ns with respect to the falling edge of the clock. Commands that have data associated with them (read and load) are specified to have a minimum delay of 1  $\mu\text{s}$  between the command and the data. After this delay the clock pin is cycled 18 times with the first cycle being a start bit and the last cycle being a stop bit. Data is also input and output LSB first. Therefore, during a read operation the LSB will be transmitted onto pin RB7 on the rising edge of the second cycle, and during a load operation the LSB will be latched on the falling edge of the second cycle. A minimum 1  $\mu\text{s}$  delay is also specified between consecutive commands.

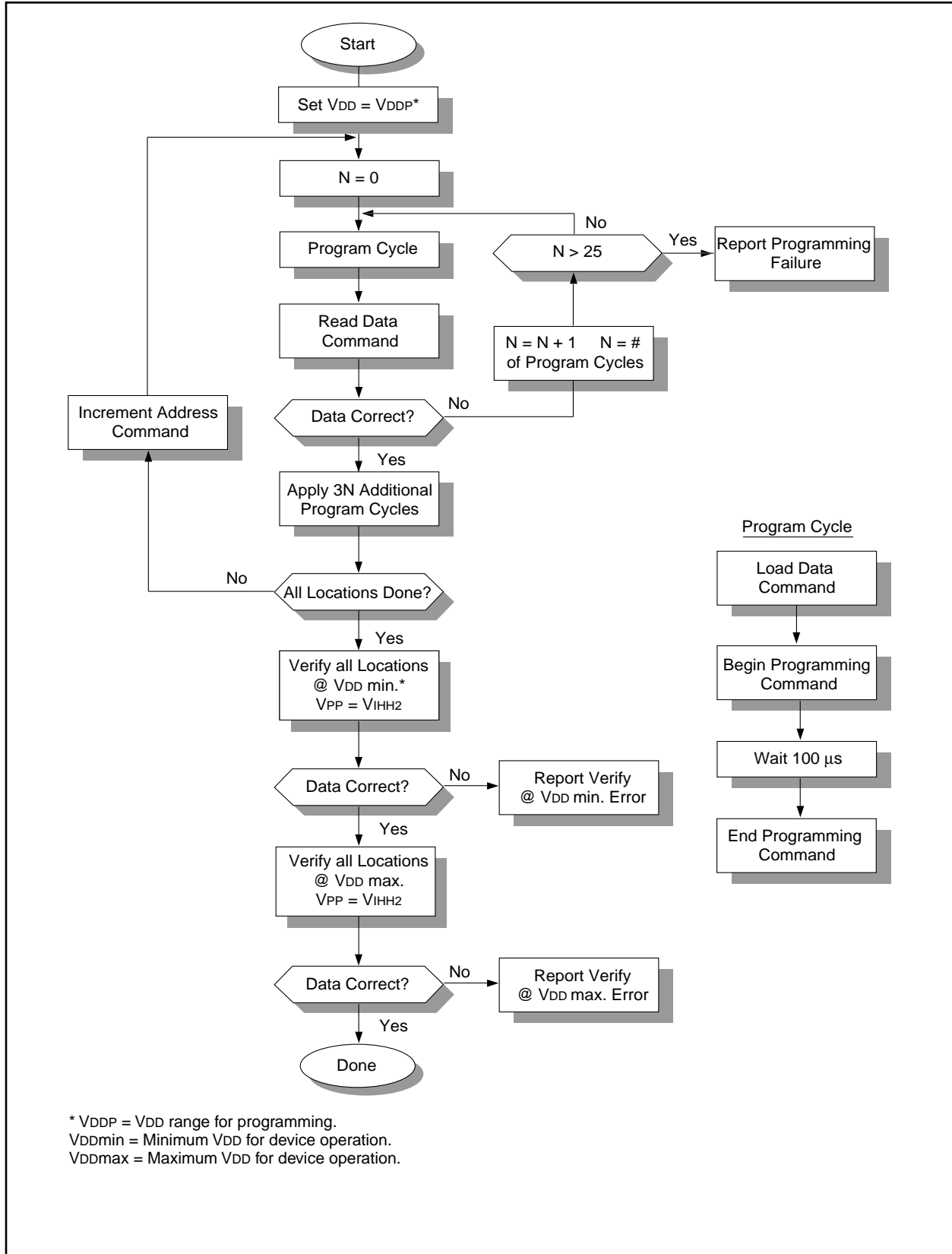
### 2.3.2 LOAD CONFIGURATION

After receiving this command, the program counter (PC) will be set to 0x2000. By then applying 18 cycles to the clock pin, the chip will load 16-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  $\overline{\text{MCLR}}$  low ( $V_{\text{IL}}$ ).

TABLE 2-2: COMMAND MAPPING (SERIAL OPERATION)

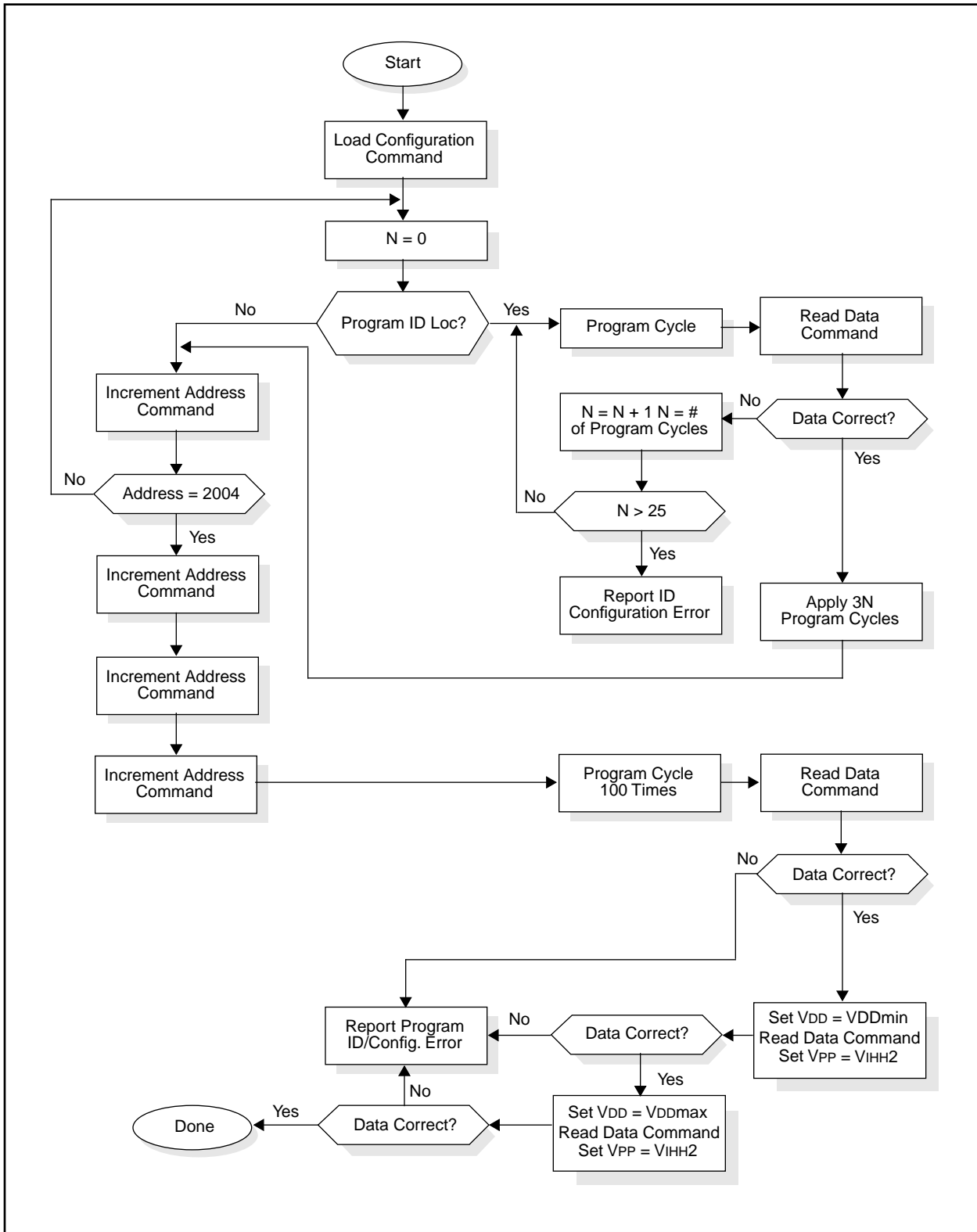
Command	Mapping (MSb...LSb)	Data
Load Data, Set PC = 2000h	X X 0 0 0 X	start_bit, data (16), stop_bit
Load Data for Program Memory	X X 0 0 1 X	start_bit, data (16), stop_bit
Read Data	X X 0 1 0 X	start_bit, data (16), stop_bit
Increment Address	X X 0 1 1 X	
Begin Programming	X X 1 0 0 X	
End Programming	X X 1 1 1 X	

**FIGURE 2-3: PROGRAM FLOW CHART - PIC16C64X/66X PROGRAM MEMORY**



# PIC16C64X/66X

FIGURE 2-4: PROGRAM FLOW CHART - PIC16C64X/66X CONFIGURATION WORD & ID LOCATIONS



## 2.3.2.1 LOAD DATA

After receiving this command, the chip will load in a 16-bit "data word" when 18 cycles are applied, as described previously. A timing diagram for the load data command is shown in Figure 5-1.

## 2.3.2.2 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 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 18th rising edge. A timing diagram of this command is shown in Figure 5-2.

## 2.3.2.3 INCREMENT ADDRESS

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

## 2.3.2.4 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 100  $\mu$ s programming pulses. A programming pulse is defined as the time between the begin programming command and the end programming command.

## 2.3.2.5 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.

## 2.4 Programming Algorithm Requires Variable VDD

The PIC16C64X/66X 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 PIC16C64X/66X at its specified VDDmax and VDDmin levels. Since Microchip may introduce future versions of the PIC16C64X/66X 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.

# PIC16C64X/66X

## 3.0 CONFIGURATION WORD

The PIC16C64X/66X family members have 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.

**Note:** Parity bits for identification and configuration word must be clocked in/out, but are ignored and not checked.

**FIGURE 3-1: CONFIGURATION WORD BIT MAP**

Bit Number:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																	
PIC16C642/662	CP1	CP0	CP1	CP0	CP1	CP0	MPEEN	BODEN	CP1	CP0	PWRTE	WDTE	FOSC1	FOSC0	P	P																	
bit 9:	<b>MPEEN</b> , Memory Parity Error Enable 1: Memory parity checking is enabled 0: Memory parity checking is disabled																																
bit 7-6:	<b>CP1:CP0</b> , Code Protect																																
	<table border="1"> <thead> <tr> <th>Device</th> <th>CP1</th> <th>CP0</th> <th>Code Protection</th> </tr> </thead> <tbody> <tr> <td rowspan="4">PIC16C642/662</td> <td>0</td> <td>0</td> <td>All memory protected</td> </tr> <tr> <td>0</td> <td>1</td> <td>Upper 3/4 memory protected</td> </tr> <tr> <td>1</td> <td>0</td> <td>Upper 1/2 memory protected</td> </tr> <tr> <td>1</td> <td>1</td> <td>Code protection off</td> </tr> </tbody> </table>																Device	CP1	CP0	Code Protection	PIC16C642/662	0	0	All memory protected	0	1	Upper 3/4 memory protected	1	0	Upper 1/2 memory protected	1	1	Code protection off
Device	CP1	CP0	Code Protection																														
PIC16C642/662	0	0	All memory protected																														
	0	1	Upper 3/4 memory protected																														
	1	0	Upper 1/2 memory protected																														
	1	1	Code protection off																														
bit 8:	<b>BODEN</b> , Brown-out Enable bit 1 = Enables Brown-out 0 = Disables Brown-out																																
bit 5:	<b>PWRTE</b> , Power-up Timer Enable bit 0 = Power up timer enabled 1 = Power up timer disabled																																
bit 4:	<b>WDTE</b> , WDT Enable bit 1 = WDT enabled 0 = WDT disabled																																
bit 3-2:	<b>FOSC1:FOSC0</b> , Oscillator Selection bit 11: RC oscillator 10: HS oscillator 01: XT oscillator 00: LP oscillator																																
bit 1-0:	Parity bits for Configuration Word																																



## 4.0 CODE PROTECTION

The program code written into the EPROM can be protected by writing to the CP0 & CP1 bits of the configuration word.

**Note:** Code protection should be the last test for a device, since the areas protected cannot be reprogrammed. Code protection is permanent for any device.

## 4.1 Programming Locations 0x000 to 0xFF after Code Protection

For all PIC16C64X/66X devices, 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.

## 4.2 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: CONFIGURATION WORD**

PIC16C642

To code protect:

- Protect all memory                   0000001X00XXXXXX
- Protect upper 1/2 memory        101010XX10XXXXXX
- Protect upper 3/4 memory        010101XX01XXXXXX
- No code protection                1111111X11XXXXXX

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
ID Locations (0x2000 : 0x2003)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16C662

To code protect:

- Protect all memory                   0000001X00XXXXXX
- Protect upper 3/4 memory        0101011X01XXXXXX
- Protect upper 1/2 memory        1010101X10XXXXXX
- No code protection                1111111X11XXXXXX

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
ID Locations (0x2000 : 0x2003)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled

# PIC16C64X/66X

## 4.3 Checksum

### 4.3.1 CHECKSUM CALCULATIONS

Checksum is calculated by reading the contents of the PIC16C64X/66X memory locations and adding up the opcodes up to the maximum user addressable location, e.g., 0xFFFF for the PIC16C662. 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 PIC16C64X/66X devices is shown in Table 4-2.

The checksum is calculated by summing the following:

- The contents of all program memory locations
- The configuration word, appropriately masked
- Masked ID locations (when applicable)

The least significant 16 bits of this sum is the checksum.

**TABLE 4-2: CHECKSUM COMPUTATION**

Device	Code Protect	Checksum*	Blank Value	0x25E6 at 0 and Max Address
PIC16C642	OFF	SUM[0x0000:0x0FFF] + (CONFIG & 0x3FFF)	0x2FFF	0xFBCD
	1/2	SUM[0x0000:0x07FF] + (CONFIG & 0x3FFF) + SUM_ID	0x52EE	0x04A3
	3/4	SUM[0x0000:0x03FF] + (CONFIG & 0x3FFF) + SUM_ID	0x41DE	0xF393
	ALL	(CONFIG & 0x3FFF) + SUM_ID	0x30CE	0xFC9C
PIC16C662	OFF	SUM[0x0000:0x0FFF] + (CONFIG & 0x3FFF)	0x2FFF	0xFBCD
	1/2	SUM[0x0000:0x07FF] + (CONFIG & 0x3FFF) + SUM_ID	0x52EE	0x04A3
	3/4	SUM[0x0000:0x03FF] + (CONFIG & 0x3FFF) + SUM_ID	0x41DE	0xF393
	ALL	(CONFIG & 0x3FFF) + SUM_ID	0x30CE	0xFC9C

Legend: CFGW = Configuration Word

SUM[a:b] = [Sum of locations a through 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 = 0x12, ID1 = 0x37, ID2 = 0x4, ID3 = 0x26, then SUM\_ID = 0x2746.

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

+ = Addition

& = Bitwise AND

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.

## 5.0 PROGRAM/VERIFY MODE

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

Standard Operating Conditions							
Operating Temperature: $+10^{\circ}\text{C} \leq T_A \leq +40^{\circ}\text{C}$ , unless otherwise stated, ( $20^{\circ}\text{C}$ recommended)							
Operating Voltage: $4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$ , unless otherwise stated.							
Parameter No.	Sym.	Characteristic	Min.	Typ	Max.	Units	Conditions
<b>General</b>							
P1	VDDP	Supply voltage during programming	4.75	5.0	5.25	V	
P2	IDDP	Supply current (from VDD) during programming	—	—	20	mA	
P3	VDDV	Supply voltage during verify	VDDmin	—	VDDmax	V	Note 1
P4	VIHH1	Voltage on $\overline{\text{MCLR}}/\text{VPP}$ during programming	12.75	—	13.25	V	Note 2
P5	VIHH2	Voltage on $\overline{\text{MCLR}}/\text{VPP}$ during verify	VDD + 4.0	—	13.5	—	
P6	IPP	Programming supply current (from VPP)	—	—	50	mA	
P9	VIH1	(RB6, RB7) input high level	0.8 VDD	—	—	V	Schmitt Trigger input
P8	VIL1	(RB6, RB7) input low level	0.2 VDD	—	—	V	Schmitt Trigger input

<b>Serial Program Verify</b>							
P1	TR	$\overline{\text{MCLR}}/\text{VPP}$ rise time (VSS to VHH) for test mode entry	—	—	8.0	$\mu\text{s}$	
P2	TF	$\overline{\text{MCLR}}$ Fall time	—	—	8.0	$\mu\text{s}$	
P3	TSET1	Data in setup time before clock $\downarrow$	100	—	—	ns	
P4	THLD1	Data in hold time after clock $\downarrow$	100	—	—	ns	
P5	TDLY1	Data input not driven to next clock input (delay required between command/data or command/command)	1.0	—	—	$\mu\text{s}$	
P6	TDLY2	Delay between clock $\downarrow$ to clock $\uparrow$ of next command or data	1.0	—	—	$\mu\text{s}$	
P7	TDLY3	Clock $\uparrow$ to data out valid (during read data)	200	—	—	ns	
P8	THLD0	Hold time after $\overline{\text{MCLR}}$ $\uparrow$	2	—	—	$\mu\text{s}$	

**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.



NOTES:

# PIC16C64X/66X

---

NOTES:

NOTES:



## WORLDWIDE SALES AND SERVICE

### AMERICAS

#### Corporate Office

Microchip Technology Inc.  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-786-7200 Fax: 480-786-7277  
Technical Support: 480-786-7627  
Web Address: <http://www.microchip.com>

#### Atlanta

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

#### Boston

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

#### Chicago

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

#### Dallas

Microchip Technology Inc.  
4570 Westgrove Drive, Suite 160  
Addison, TX 75248  
Tel: 972-818-7423 Fax: 972-818-2924

#### Dayton

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

#### Detroit

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

#### Los Angeles

Microchip Technology Inc.  
18201 Von Karman, Suite 1090  
Irvine, CA 92612  
Tel: 949-263-1888 Fax: 949-263-1338

#### New York

Microchip Technology Inc.  
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

### AMERICAS (continued)

#### Toronto

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

### ASIA/PACIFIC

#### Hong Kong

Microchip Asia Pacific  
Unit 2101, Tower 2  
Metroplaza  
223 Hing Fong Road  
Kwai Fong, N.T., Hong Kong  
Tel: 852-2-401-1200 Fax: 852-2-401-3431

#### Beijing

Microchip Technology, Beijing  
Unit 915, 6 Chaoyangmen Bei Dajie  
Dong Erhuan Road, Dongcheng District  
New China Hong Kong Manhattan Building  
Beijing 100027 PRC  
Tel: 86-10-85282100 Fax: 86-10-85282104

#### India

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

#### Japan

Microchip Technology Intl. Inc.  
Benex S-1 6F  
3-18-20, Shinyokohama  
Kohoku-Ku, Yokohama-shi  
Kanagawa 222-0033 Japan  
Tel: 81-45-471-6166 Fax: 81-45-471-6122

#### Korea

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

#### Shanghai

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

### 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, R.O.C

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

### EUROPE

#### United Kingdom

Arizona Microchip Technology Ltd.  
505 Eskdale Road  
Wokingham  
Berkshire, England RG41 5TU  
Tel: 44 118 921 5858 Fax: 44-118 921-5835

#### Denmark

Microchip Technology Denmark ApS  
Regus Business Centre  
Lautrup hof 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 - 1er 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 München, Germany  
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

#### Italy

Arizona Microchip Technology SRL  
Centro Direzionale Colleoni  
Palazzo Taurus 1 V. Le Colleoni 1  
20041 Agrate Brianza  
Milan, Italy  
Tel: 39-039-65791-1 Fax: 39-039-6899883

11/15/99



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and water 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, KEELOC® 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.

All rights reserved. © 1999 Microchip Technology Incorporated. Printed in the USA. 11/99 Printed on recycled paper.

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