MICROCHIP

PIC16C84

EEPROM Memory Programming Specification

This document includes the programming specifications for the following devices:

- PIC16C84

1.0 PROGRAMMING THE PIC16C84

The PIC16C84 is programmed using the serial method. The serial mode will allow the PIC16C84 to be programmed while in the users system. This allows for increased design flexibility.

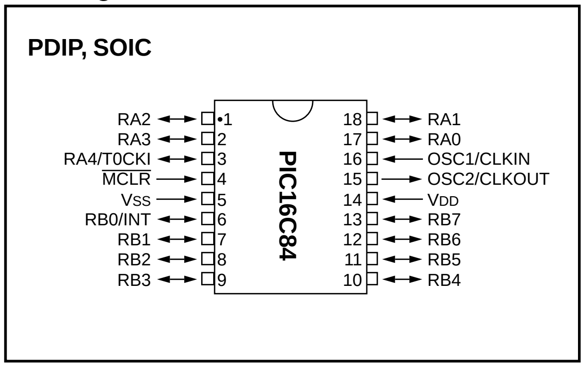
1.1 Hardware Requirements

The PIC16C84 requires 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 PIC16C84 allows programming of user program memory, data memory, special locations used for ID, and the configuration word.

Pin Diagram



PIN DESCRIPTIONS (DURING PROGRAMMING): PIC16C84

Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
RB6	CLOCK	I	Clock input
RB7	DATA	I/O	Data input/output
$\overline{\text{MCLR}}$	VTEST MODE	P*	Program Mode Select
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground

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

*In PIC16C84, programming high voltage is internally generated. To activate the programming mode, high voltage needs to be applied to $\overline{\text{MCLR}}$ input. This means that $\overline{\text{MCLR}}$ does not draw any significant current.

This document was created with FrameMaker 4.0.4

2.0 PROGRAM MODE ENTRY

2.1 User Program Memory Map

The user memory space extends from 0x0000 to 0x1FFF (8K), of which 1K (0x0000 - 0x03FF) 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 0x3FF, 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 to 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, 0x2000-0x200F are physically implemented. Locations beyond 0x200F will physically access user memory. (See Figure 2-1).

2.2 ID Locations

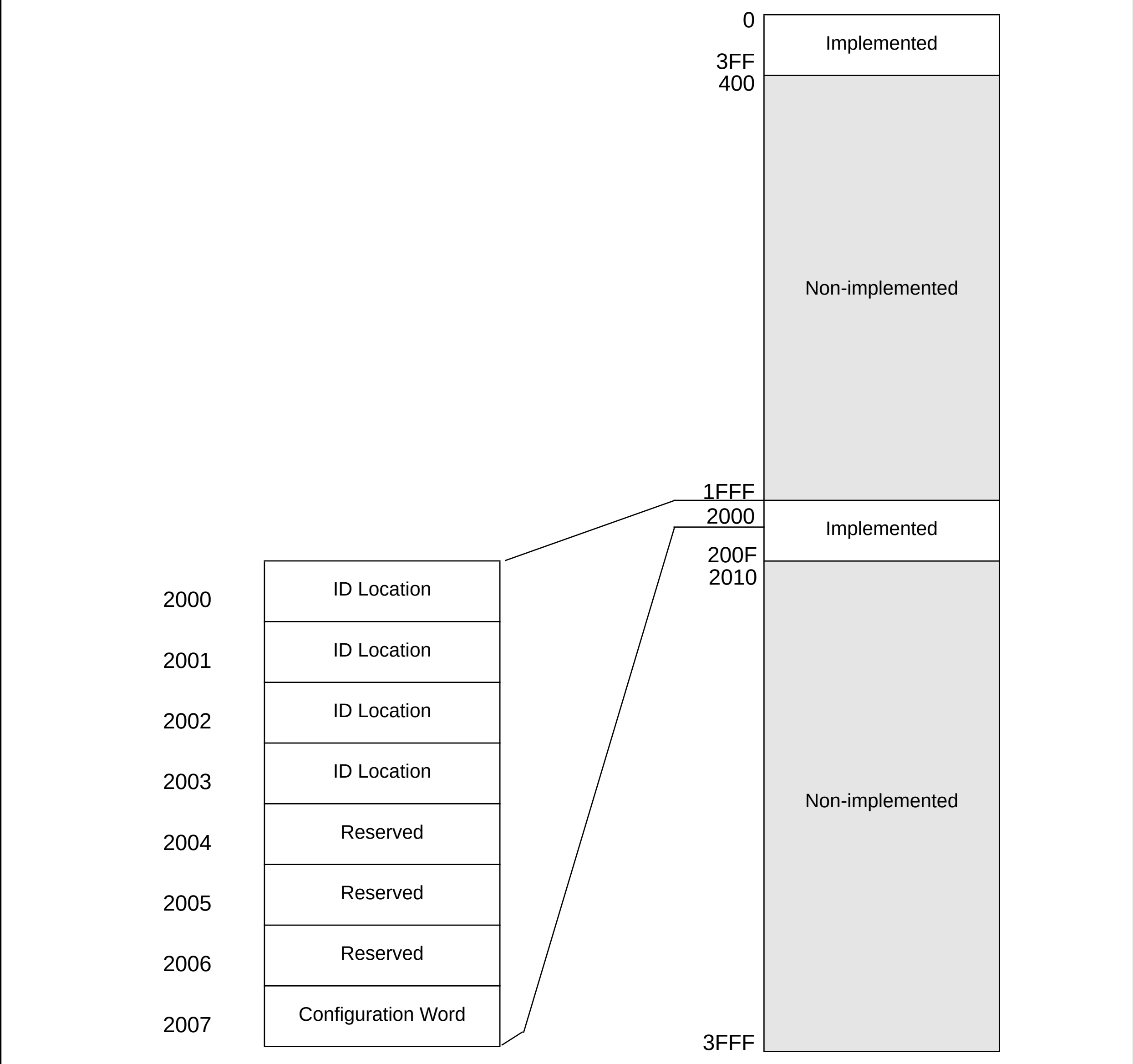
A user may store identification information (ID) in four ID locations. The ID locations are mapped in [0x2000 : 0x2003]. It is recommended that the user use only the four least significant bits of each ID location. In some devices, the ID locations read-out in a scrambled fashion after code protection is enabled. For these devices, it is recommended that ID location is written as “11 1111 1000 bbbb” where 'bbbb' is ID information.

In other devices, the ID locations read out normally, even after code protection. To understand how the devices behave, refer to Table 4.3.

To understand the scrambling mechanism after code protection, refer to Section 4.0.

© 1996 Microchip Technology Inc.

FIGURE 2-1: PROGRAM MEMORY MAPPING



2.3 Program/Verify Mode

The program/verify mode is entered by holding pins RB6 and RB7 low while raising $\overline{\text{MCLR}}$ 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. RB6 and RB7 are Schmitt Trigger Inputs in this mode.

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

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) 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 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 1 µs is required between a command and a data word (or another command).

The commands that are available are:

2.3.1.1 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 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 $\overline{\text{MCLR}}$ low (V_{IL}).

TABLE 2-1: COMMAND MAPPING (SERIAL OPERATION)

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 Programming	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	

© 1996 Microchip Technology Inc.

FIGURE 2-2: PROGRAM FLOW CHART - PIC16C84 PROGRAM MEMORY

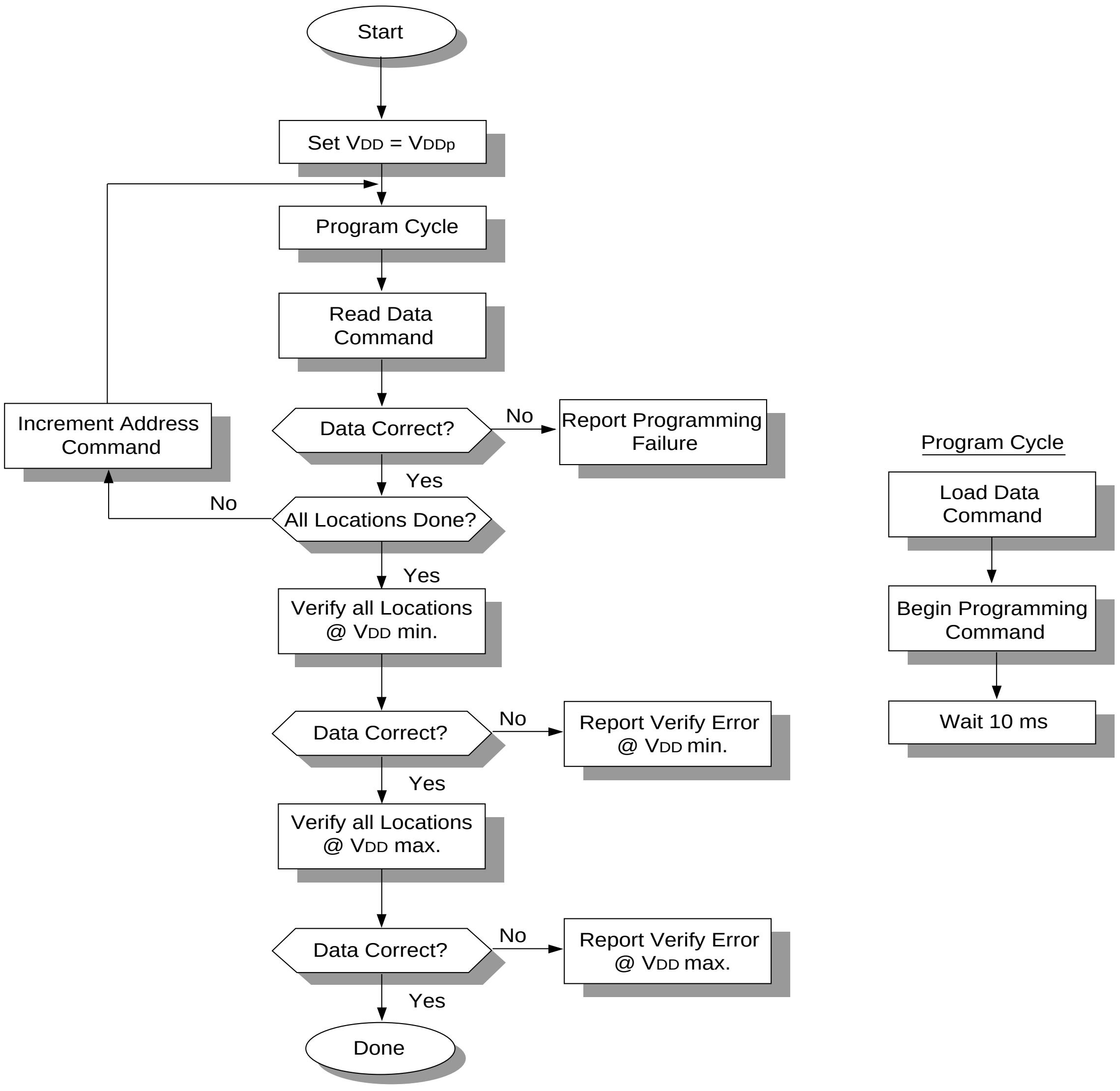
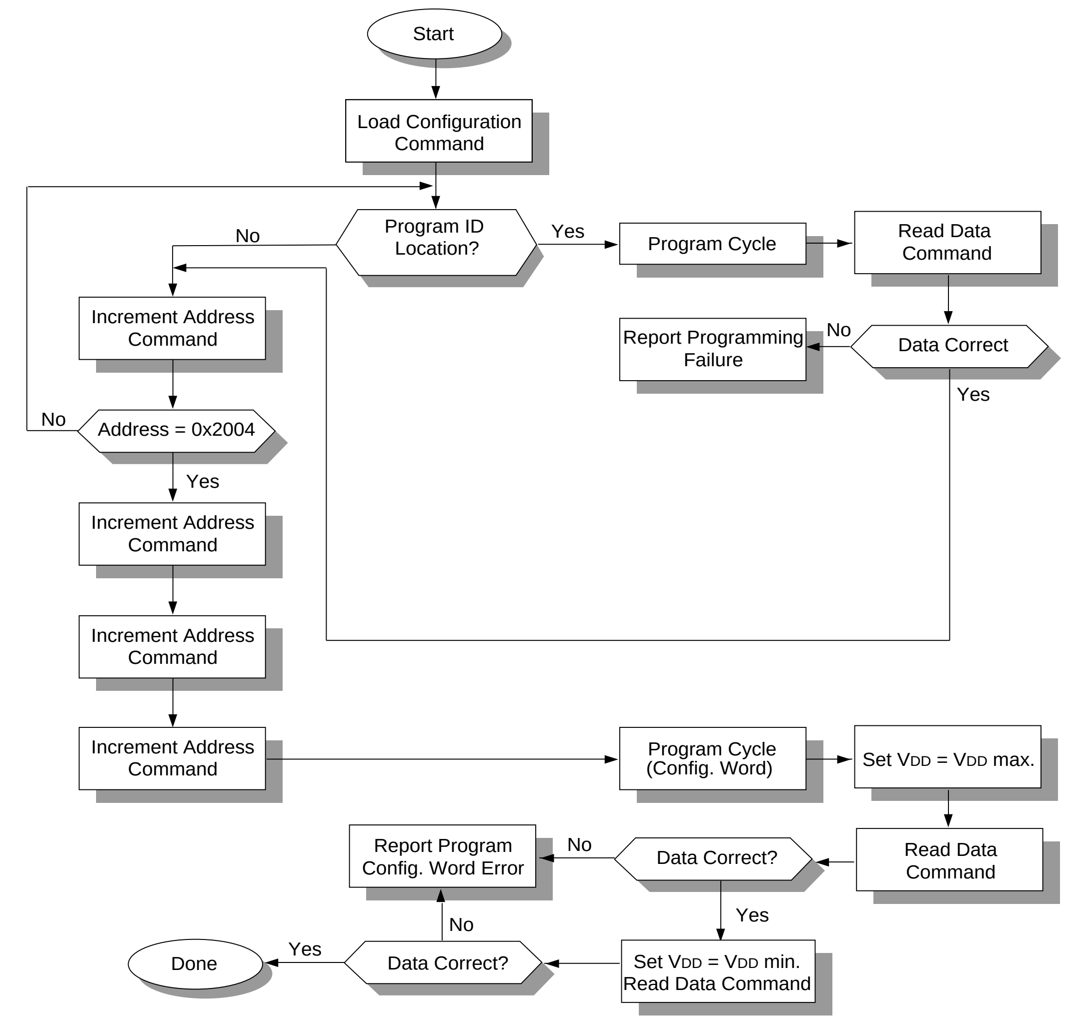

Start
Set VDD = VDDp
Program Cycle
Read Data Command
Data Correct?
No
Report Programming Failure
Yes
All Locations Done?
No
Increment Address Command
Yes
Verify all Locations @ VDD min.
Data Correct?
No
Report Verify Error @ VDD min.
Yes
Verify all Locations @ VDD max.
Data Correct?
No
Report Verify Error @ VDD max.
Yes
Done
Program Cycle
Load Data Command
Begin Programming Command
Wait 10 ms

FIGURE 2-3: PROGRAM FLOW CHART - PIC16C84 CONFIGURATION MEMORY


Start
Load Configuration Command
Program ID Location?
Yes
No
Program Cycle
Read Data Command
Data Correct
No
Report Programming Failure
Yes
Increment Address Command
No
Address = 0x2004
Yes
Increment Address Command
Increment Address Command
Increment Address Command
Program Cycle (Config. Word)
Set VDD = VDD max.
Read Data Command
Data Correct?
No
Report Program Config. Word Error
Yes
Set VDD = VDD min. Read Data Command
Data Correct?
No
Yes
Done

© 1996 Microchip Technology Inc.

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.

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 PROGRAMMING

A load command must be given before every begin programming command. Programming of the appropriate memory (test program 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 10ms for programming to complete. No "end programming" command is required.

2.3.1.8 BULK ERASE PROGRAM MEMORY

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

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 test program memory (0x2000 - 0x200F), then both the user memory and the test memory will be erased. The configuration word will not be erased, even if the address is pointing to location 0x2007.

If the address is pointing to the test program memory (0x2000 - 0x200F), then both the user memory and the test memory will be erased. The configuration word will not be erased, even if the address is pointing to location 0x2007.

2.3.1.9 BULK ERASE DATA MEMORY

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

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.

2.4 Programming Algorithm Requires Variable V_{DD}

The PIC16C84 uses an intelligent algorithm. The algorithm calls for program verification at V_{DD} (min.) as well as V_{DD} (max.). Verification at V_{DD} (min.) guarantees good "erase margin". Verification at V_{DD} (max) guarantees good "program margin".

The actual programming must be done with V_{DD} in the V_{DDP} range (4.5 - 5.5V).

V_{DDP} =	V_{CC} range required during programming.
V_{DD} min. =	minimum operating V_{DD} spec for the part.
V_{DD} max.=	maximum operating V_{DD} spec for the part.

Programmers must verify the PIC16C84 at its specified V_{DD} max. and V_{DD} min. levels. Since Microchip may introduce future versions of the PIC16C84 with a broader V_{DD} 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 PIC16C84 has five configuration bits. These bits can be set (reads '0') or left unchanged (reads '1') to select various device configurations.

FIGURE 3-1: CONFIGURATION WORD BIT MAP

Bit Number:	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CP	PWRTE	WDTE	FOSC1	FOSC0

bit 4: **CP,** Code Protection Configuration Bit
1 = code protection off
0 = code protection on

bit 3: **PWRTE**, Power Up Timer Enable Configuration Bit
1 = Power up timer enabled
0 = Power up timer disabled

bit 3-2: **WDTE**, WDT Enable Configuration Bits
1 = WDT enabled
0 = WDT disabled

bit 1-0 **FOSC<1:0>**, Oscillator Selection Configuration Bits
11: RC oscillator
10: HS oscillator
01: XT oscillator
00: LP oscillator

© 1996 Microchip Technology Inc.

4.0 CODE PROTECTION

For PIC16C84 devices, once code protection is enabled, all program memory locations read out in a scrambled fashion. The ID locations and the configuration word also read out in a scrambled 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.

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:

a) Execute load configuration (with a '1' in bit 4, code protect).
b) Increment to configuration word location (0x2007)
c) Execute command (000001)
d) Execute command (000111)
e) Execute 'Begin Programming' (001000)
f) Wait 10ms
g) Execute command (000001)
h) Execute command (000111)

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, all configuration word and ID information must be included. An option to not include this information may be provided.

Specifically for the PIC16C84, 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.

TABLE 4-1: CONFIGURATION WORD

PIC16C84

To code protect: `XXXXXXXX0XXX`

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
All memory.	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
ID Locations [0x2000 : 0x2003]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled

Legend: X = Don’t care

4.3 Checksum

4.3.1 CHECKSUM CALCULATIONS

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

Device	Code Protect	Checksum*	Blank Value	0x25E6 at 0 and max address
PIC16C84	OFF ON	SUM[0x000:0x3FF] + CFGW & 0x1F + 0x3FE0 SUM_XNOR7[0x000:0x3FF] + (CFGW & 0x1F | 0x60)	0x3BFF 0xFC6F	0x07CD 0xFC15

Legend: CFGW = Configuration Word
SUM[a:b] = [Sum of locations a to b inclusive]
SUM_XNOR7[a:b] = XNOR of the seven high order bits of memory location with the seven low order bits summed over locations a through b inclusive. For example, location_a = 0x123 and location_b = 0x456, then SUM_XNOR7 [location_a : location_b] = 0x001F.
*Checksum = [Sum of all the individual expressions] **MODULO** [0xFFFF]

© 1996 Microchip Technology Inc.

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 0x2100. 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 ≤ T_A ≤ +40°C, unless otherwise stated, (25°C is recommended)

Operating Voltage 4.5V ≤ V_{DD} ≤ 5.5V, unless otherwise stated.

Characteristic	Sym.	Min.	Typ.	Max.	Units	Conditions/Comments
Supply voltage during programming	V_{DDP}	4.5	5.0	5.5	V	
Supply voltage during verify	V_{DDV}	V_{DD} min.		V_{DD} max.	V	Note 1
High voltage on $\overline{MCLR}$ for test mode entry	VIHH	12		14.0	V	Note 2
Supply current (from V_{DD}) during program/verify	I_{DDP}			50	mA	
Supply current from V_{IHH} (on $\overline{MCLR}$)	I_{HH}			200	µA	
$\overline{MCLR}$ rise time (V_{SS} to V_{HH}) for test mode entry	t_{VHHR}			1.0	µs	
(RB6, RB7) input high level	V_{IH1}	0.8 V_{DD}			V	Schmitt Trigger input
(RB6, RB7) input low level $\overline{MCLR}$ (test mode selection	V_{IL1}	0.2 V_{DD}			V	Schmitt Trigger input
RB6, RB7 setup time (before pattern setup time)	tset0	100			ns	
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	

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

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

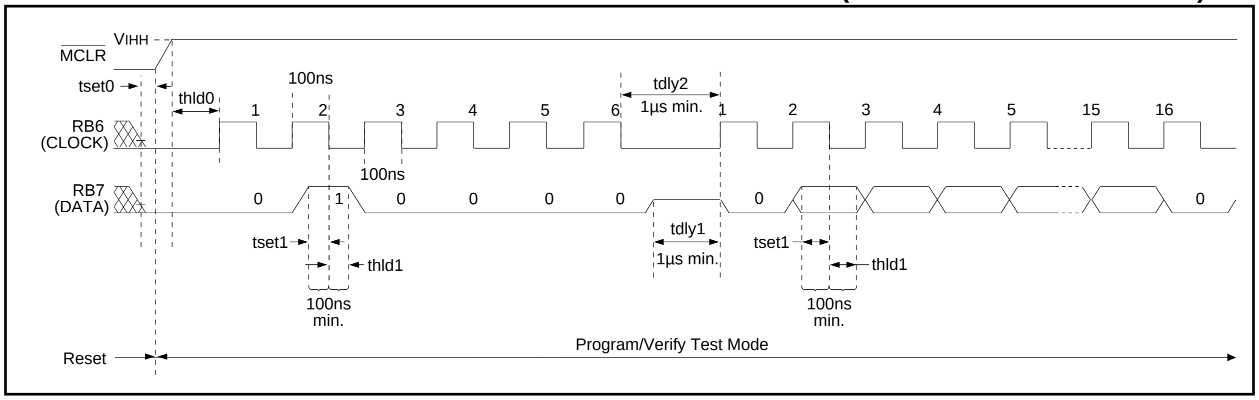


FIGURE 5-2: READ DATA FROM PROGRAM MEMORY COMMAND (SERIAL PROGRAM/VERIFY)

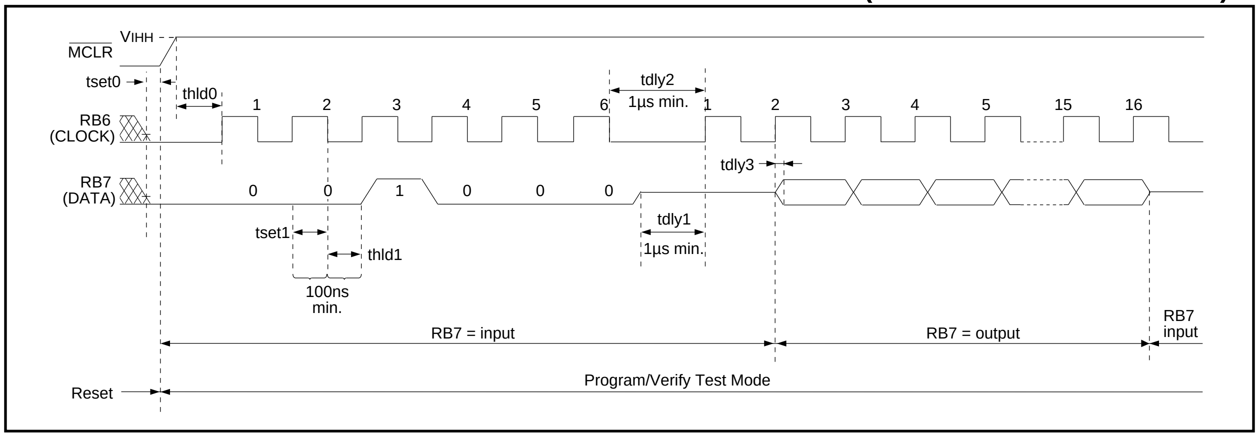
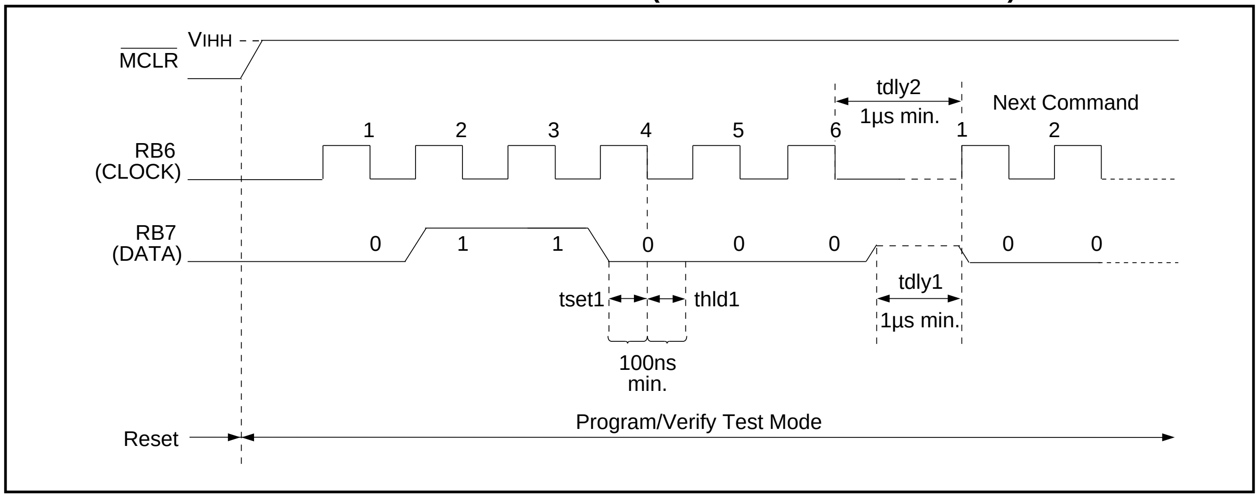


FIGURE 5-3: INCREMENT ADDRESS COMMAND (SERIAL PROGRAM/VERIFY)



© 1996 Microchip Technology Inc.

NOTES: