



MICROCHIP

PIC16C5X

EPROM Memory Programming Specification

This document includes the programming specifications for the following devices:

- PIC16C54
- PIC16C54A
- PIC16C55
- PIC16C56
- PIC16C57
- PIC16C58A
- PIC16CR54
- PIC16CR57A
- PIC16CR58A

INTRODUCTION

Overview

The PIC16C5X Series is a family of single-chip CMOS microcontrollers with on-chip EPROM for program storage. The programming specification also applies to ROM products for verification only.

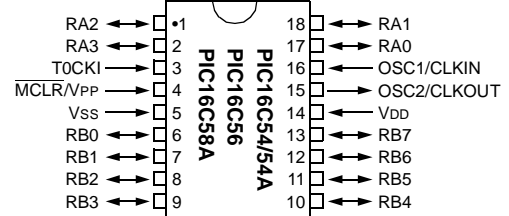
Due to the special architecture of these microcontrollers (12-bit wide instruction word) and the low pin counts (starting at 18 pins), the EPROM programming methodology is different from that of standard (byte-wide) EPROMs (e.g., 27C256).

The PIC16C5X Series can be programmed by applying the 12-bit wide data word to the 12 available I/O pins while the address is generated by the on-chip Program Counter. The MCLR/VPP pin provides the programming supply voltage (VPP). Programming/verify chip enable is controlled by the T0CKI pin while the OSC1 pin controls the Program Counter.

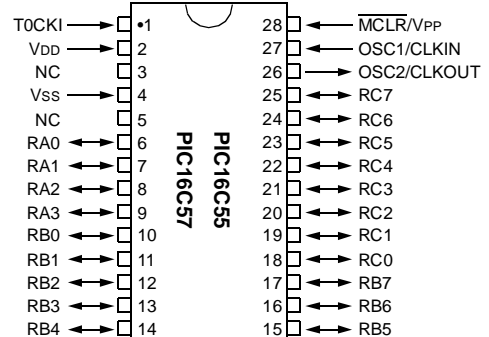
This document describes all the programming details of the PIC16C5X Series and the requirements for programming equipment to be used from programming prototypes in the engineering lab up to high volume programming on the factory floor.

PIN CONFIGURATIONS

PDIP, SOIC, Cerdip Window



PDIP, SOIC, Cerdip Window



PIN DESCRIPTIONS (DURING PROGRAMMING): PIC16C54/54A/56/58a, PIC16C55/57

Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
T0CKI	PROG/VER	I	Program pulse input/verify pulse input
RA0 - RA3	D0 - D3	I/O	Data input/output
RB0 - RB7	D4 - D11	I/O	Data input/output
OSC1	INCPC	I	Increment Program Counter input
MCLR/VPP	VPP	P	Programming Power
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground

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

1.0 PROGRAM/VERIFY MODES

The PIC16C5X Series uses the internal Program Counter (PC) to generate the EPROM address. VPP is supplied through the $\overline{\text{MCLR}}$ pin.

The T0CKI pin acts as chip enable, alternating between programming and verifying.

The OSC1 pin is used for incrementing the PC.

Data is applied to, or can be read on PORTA and PORTB (MSB on RB7, LSB on RA0).

The programming/verify mode is entered by raising the level on the $\overline{\text{MCLR}}$ pin from VIL to VHH (= VPP) while the T0CKI pin is held at VIH and the OSC1 pin is held at VIL.

The Program Counter now has the value "0xFFF", because $\overline{\text{MCLR}}$ was at VIL before. This condition selects the configuration word as the very first EPROM location to be accessed after entering the program/verify mode.

Since the $\overline{\text{MCLR}}$ pin was initially at VIL, the device is in the reset state (the I/O pins are in the reset state).

Incrementing the PC once (by pulsing the OSC1 pin) selects location "0x000" of the user program memory. Afterwards all other memory locations from 001h through end of memory can be addressed by incrementing the PC .

If the Program Counter has reached the last address of the user memory area (e.g. "0x1FF" for the PIC16C54/55), and is incremented again, the on-chip special EPROM area will be addressed. (See Figure 1-2 to determine where the special EPROM area is located for the various PIC16C5X devices).

1.1 Program/Verify without PC Increment

After entering the program/verify mode, pulsing the T0CKI pin LOW programs the data present on PORTA and PORTB into the memory location selected by the Program Counter. The duration of the T0CKI LOW time determines the length of the programming pulse.

Pulsing the T0CKI pin LOW again without changing the signals on $\overline{\text{MCLR}}$ and OSC1 puts the contents of the selected memory location out on PORTA and PORTB for verification of a successful programming cycle. This verification pulse on T0CKI can be much shorter than the programming pulse. If the programming was not successful, T0CKI can be pulsed LOW again to apply another programming pulse, followed again by a shorter T0CKI LOW pulse for another verification cycle.

This sequence can be repeated as many times as required until the programming is successful.

1.2 Verify with PC Increment

If a verification cycle shows that programming was successful, the Program Counter can be incremented by keeping the T0CKI input at a HIGH level while pulsing the OSC1 input HIGH. When both T0CKI and OSC1 are HIGH, the contents of the selected memory location is put out on Ports A and B (= Verify). The falling edge of OSC1 will increment the Program Counter.

A fast VERIFY- ONLY with automatic increment of the PC can be performed by entering the program/verify mode as described above and then clocking the OSC1 input. If OSC1 is HIGH, the selected memory location is output on Ports A and B, while the falling edge of OSC1 will increment the Program Counter. Thus, the first memory location to be verified after entering the program/verify mode, is the configuration word. The next location is 000h followed by 001h and so on. The program memory location "N" can be reached by generating "N + 1" falling edges on OSC1. When OSC1 is brought HIGH again, the contents of address "N" are output on Ports A and B as long as OSC1 stays HIGH.

1.3 Programming/Verifying Configuration Word

The configuration word is logically mapped at program memory location "0xFFF". The PC points to the configuration word after $\overline{\text{MCLR}}$ pin goes from LOW to VHH (HIGH). The configuration word can be programmed or verified using the techniques described in Section 1.1 and Section 1.2.

If PC is incremented, the next location it will point to is "0x000" in user memory. Incrementing PC 4096 times will not allow the user to point to the configuration word. The only way to point to it again is to reset and re-enter program mode.

Programming Specification

1.4 Programming Method

The programming technique is described in the following section. It is designed to guarantee good programming margins. It does, however, require a variable power supply for VCC.

1.4.1 PROGRAMMING METHOD DETAILS

Essentially, this technique includes the following steps:

1. Perform blank check at $V_{DD} = V_{DD \text{ min}}$. Report failure. The device may not be properly erased.
2. Program location with pulses (100 μ s typically) and verify after each pulse at $V_{DD} = V_{DDP}$: where $V_{DDP} = V_{DD}$ range required during programming (4.5V - 5.5V).
 - a) Programming condition:
 $V_{PP} = 13.0V$ to 13.25V
 $V_{DD} = V_{DDP} = 4.5V$ to 5.5V
 V_{PP} must be $\geq V_{DD} + 7.25V$ to keep "programming mode" active.
 - b) Verify condition:
 $V_{DD} = V_{DDP}$
 $V_{PP} \geq V_{DD} + 7.5V$ but not to exceed 13.25V
If location fails to program after "N" pulses, (suggested maximum program pulses of 25) then report error as a programming failure.

Note: Device must be verified at minimum and maximum specified operating voltages as specified in the data sheet.

3. Once location passes "Step 2", apply 3X over-programming, i.e., apply three times the number of pulses that were required to program the location. This will guarantee a solid programming margin. The overprogramming should be made "software programmable" for easy updates.
4. Program all locations.
5. Verify all locations (using speed verify mode) at $V_{DD} = V_{DD \text{ min}}$.
6. Verify all locations at $V_{DD} = V_{DD \text{ max}}$.
 $V_{DD \text{ min}}$ is the minimum operating voltage spec. for the part. $V_{DD \text{ max}}$ is the maximum operating voltage spec. for the part.

1.4.2 SYSTEM REQUIREMENTS

Clearly, to implement this technique, the most stringent requirements will be that of the power supplies:

VPP: V_{PP} can be a fixed 13.0V to 13.25V supply. It must not exceed 14.0V to avoid damage to the pin and should be current limited to approximately 100mA.

VDD: 2.0V to 6.5V with 0.25V granularity. Since this method calls for verification at different V_{DD} values, a programmable V_{DD} power supply is needed.

Current Requirement: 40mA maximum

Microchip may release PIC16C5Xs in the future with different V_{DD} ranges which make it necessary to have a programmable V_{DD} .

It is important to verify an EPROM at the voltages specified in this method to remain consistent with Microchip's test screening. For example, a PIC16C5X specified for 4.5V to 5.5V should be tested for proper programming from 4.5V to 5.5V.

Note: Any programmer not meeting the programmable V_{DD} requirement and the verify at $V_{DD \text{ max}}$ and $V_{DD \text{ min}}$ requirement may only be classified as "prototype" or "development" programmer but not a production programmer.

1.4.3 SOFTWARE REQUIREMENTS

Certain parameters should be programmable (and therefore easily modified) for easy upgrade.

- a) Pulse width, current value 100 μ s.
- b) Maximum number of pulses, current limit 25.
- c) Number of over-programming pulses: should be $= (A \cdot N) + B$, where N = number of pulses required in regular programming. In our current algorithm $A = 3$, $B = 0$.

1.5 Programming Pulse Width

Normal EPROM Cells: When programming one word of EPROM, a programming pulse width (TPW) of 100 μ s is recommended.

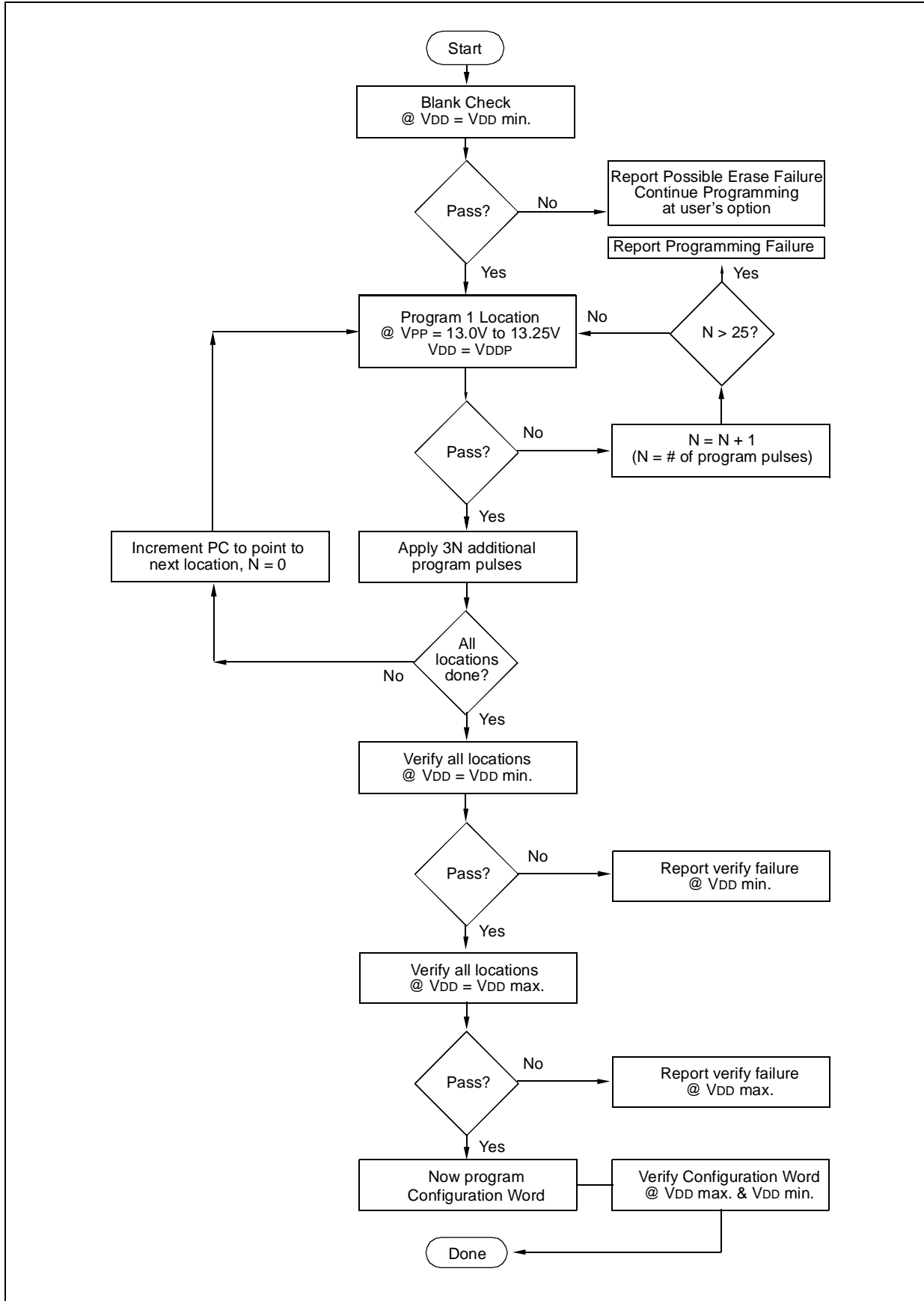
The maximum number of programming attempts should be limited to 25 per word.

After the first successful verify, the same location should be over-programmed with 3X over-programming.

Configuration Word: The configuration word for oscillator selection, WDT (watchdog timer) disable and code protection, requires a programming pulse width (TPWF) of 10ms. A series of 100 μ s pulses is preferred over a single 10ms pulse.

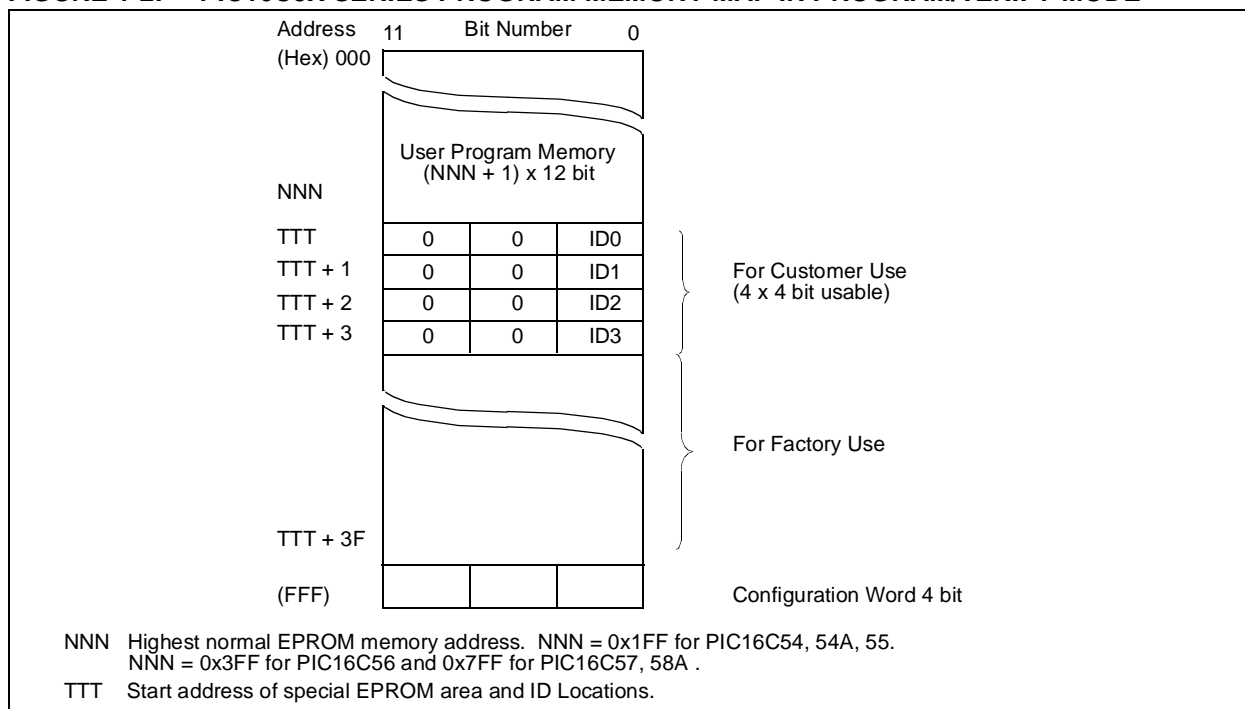
PIC16C5X

FIGURE 1-1: PROGRAMMING METHOD



Programming Specification

FIGURE 1-2: PIC16C5X SERIES PROGRAM MEMORY MAP IN PROGRAM/VERIFY MODE



1.6 Special Memory Locations

The ID Locations area is only enabled if the device is in a test or programming/verify mode. Thus, in normal operation mode only the memory location 0x000 to 0xNNN will be accessed and the Program Counter will just roll over from address 0xNNN to 0x000 when incremented.

The configuration word can only be accessed immediately after $\overline{\text{MCLR}}$ going from V_{IL} to V_{HH} . The Program Counter will be set to all '1's upon $\overline{\text{MCLR}} = V_{IL}$. Thus, it has the value "0xFFF" when accessing the configuration EPROM. Incrementing the Program Counter once by pulsing OSC1 causes the Program Counter to roll over to all '0's. Incrementing the Program Counter 4K times after reset ($\overline{\text{MCLR}} = V_{IL}$) does not allow access to the configuration EPROM.

1.6.1 CUSTOMER ID CODE LOCATIONS

Per definition, the first four words (address TTT to TTT + 3) are reserved for customer use. It is recommended that the customer use only the four lower order bits (bits 0 through 3) of each word and filling the eight higher order bits with '0's.

A user may want to store an identification code (ID) in the ID locations and still be able to read this code after the code protection bit was programmed. This is possible if the ID code is only four bits long per memory location, is located in the least significant nibble boundary of the 12-bit word, and the remaining eight bits are all '0's.

EXAMPLE 1-1: CUSTOMER CODE 0xD1E2

The Customer ID code "0xD1E2" should be stored in the ID locations 200-203 like this:

```

200:  0000 0000 1101
201:  0000 0000 0001
202:  0000 0000 1110
203:  0000 0000 0010
  
```

Reading these four memory locations, even with the code protection bit programmed would still output on Port A the bit sequence "1101", "0001", "1110", "0010" which is "0xD1E2".

Note: Microchip will assign a unique pattern number for QTP and SQTP requests and for ROM devices. This pattern number will be unique and traceable to the submitted code.

PIC16C5X

2.0 CONFIGURATION WORD

The configuration word is the very first memory location which is accessed after entering the program/verify mode of the PIC16C5X. It contains the two bits for the selection of the oscillator type, the watchdog timer enable bit, and the code protection bit. All other bits (4 through 11) are read as '1's.

One-Time-Programmable (OTP) devices may have the oscillator configuration bits "FOSC0" and "FOSC1" set by the factory and are tested accordingly. The pack-

ages are marked "PIC16C5XHS", "PIC16C5XXT", "PIC16C5XLP", or "PIC16C5XRC". Therefore, it is essential that the inputs RA0 and RA1 are held at '1's when programming the "WDTE" and/or the "CP" bit of the configuration word. Otherwise, the factory tested and selected oscillator configuration could be overwritten and the functionality of the device is not guaranteed any more.

FIGURE 2-1: CONFIGURATION WORD BIT MAP

Bit Number:	11	10	9	8	7	6	5	4	3	2	1	0
PIC16CR58A	CP	CP	CP	CP	CP	CP	CP	CP	CP	WDTE	FOSC1	FOSC0
Other PIC16C5X	—	—	—	—	—	—	—	—	CP	WDTE	FOSC1	FOSC0
	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	RA3	RA2	RA1	RA0

TABLE 2-1: CONFIGURATION BIT FUNCTIONALITY (PIC16C54/55/56/57/C54A/C58A/CR54/CR57A)

RA3-RA11 CP	RA2 WDTE	RA1 FOSC1	RA0 FOSC2	Function	Remarks
1	x	x	x	Memory Unprotected	Default
0	x	x	x	Memory protected	
x	1	x	x	Watchdog Timer enabled	Default
x	0	x	x	Watchdog Timer disabled	
x	x	1	1	RC Oscillator	Default
x	x	1	0	HS - High Speed Crystal	
x	x	0	1	XT - Standard Crystal	
x	x	0	0	LP - Low Frequency Crystal	

Legend: 1= Erased (apply HIGH Level to I/O pin during program)
 0 = Written (apply LOW Level to I/O pin during program).
 x = Don't Care

Programming Specification

TABLE 2-2: CONFIGURATION WORD

PIC16C54 (CP enable pattern: XXXXXXXX0XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFF)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
ID Words [0x200 : 0x203]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
[0x040 : 0x1FF]	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
[0x000 : 0x03F]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16C54A (CP enable pattern: XXXXXXXX0XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFF)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
ID Words [0x200 : 0x203]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
[0x040 : 0x1FF]	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
[0x000 : 0x03F]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16CR54 (CP enable pattern: XXXXXXXX0XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFF)	Read Scrambled	Read Unscrambled
ID Words [0x200 : 0x203]	Read Scrambled	Read Unscrambled
[0x040 : 0x1FF]	Read Scrambled	Read Unscrambled
[0x000 : 0x03F]	Read Scrambled	Read Unscrambled

PIC16C55 (CP enable pattern: XXXXXXXX0XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFF)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
ID Words [0x200 : 0x203]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
[0x040 : 0x1FF]	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
[0x000 : 0x03F]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16C56 (CP enable pattern: XXXXXXXX0XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFF)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
ID Words [0x400 : 0x403]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
[0x040 : 0x3FF]	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
[0x000 : 0x03F]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16C57 (CP enable pattern: XXXXXXXX0XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFF)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
ID Words [0x800 : 0x803]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
[0x040 : 0x7FF]	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
[0x000 : 0x03F]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16CR57A (CP enable pattern: XXXXXXXX0XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFF)	Read Scrambled	Read Unscrambled
ID Words [0x800 : 0x803]	Read Scrambled	Read Unscrambled
[0x040 : 0x7FF]	Read Scrambled	Read Unscrambled
[0x000 : 0x03F]	Read Scrambled	Read Unscrambled

PIC16C5X

PIC16C58A (CP enable pattern: XXXXXXXX0XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFFF)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
ID Words [0x800 : 0x803]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
[0x040 : 0x7FF]	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
[0x000 : 0x03F]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16CR58A (CP enable pattern: 00000000XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFFF)	Read Unscrambled	Read Unscrambled
ID Words [0x800 : 0x803]	Read Unscrambled	Read Unscrambled
[0x040 : 0x7FF]	Read Disabled	Read Unscrambled
[0x000 : 0x03F]	Read Unscrambled	Read Unscrambled

Legend: X = Don't care

2.1 Code Protection

The program code written into the EPROM can be protected by writing to the "CP" bit of the configuration word. All memory locations starting at 0x40 and above are protected against programming. It is still possible to program locations 0x00 through 0x3F, the ID locations, and the configuration word.

Note: Locations [0x000 : 0x03F] are not secure after code protection.

2.1.1 PROGRAMMING LOCATIONS 0x000 TO 0x03F AFTER CODE PROTECTION

In a code protected part, these locations will program with the exception of the PIC16CRXX devices. They will read back scrambled data, with the exception of PIC16CR54A and PIC16CR58A. In any event, the programmer cannot verify the device once it is code protected.

In code protected parts, specifically PIC16C54/ C54A/ CR54/C55/C56/C57 devices, the contents of the program memory cannot be read out in a way that the program code can be reconstructed. A location when read out will read as: 0000 0000 xxxx where xxxx is the XOR of the three nibbles.

For example, if the memory location contains 0xC04 (movlw 4), after code protection the output will be 0x008.

In addition, all memory locations starting at 0x40 and above are protected against programming. It is still possible to program locations 0x000 through 0x03F and the configuration word. However, performing a verify with activated code protection logic puts a 4-bit wide "checksum" on PORTA while the 8-bits of PORTB are read as '0's. The checksum is computed as follows:

The four high order bits of an instruction word are "XOR'ed" with the four middle and the four low order bits, and the result is transferred to PORTA. All memory locations are affected.

To program location 0x000 to 0x03F in a code protected part, the programmer should program one nibble at a time and verify the result through the XOR'ed output. For example, to program a location with 0xA93, first program the location with 0xFF3, verify checksum to be 0x003; then program the location with 0xF93 and verify the XOR'ed output to be 0x00C and finally program the location with 0xA93 and verify the read-out to be 0x006.

2.2 CHECKSUM COMPUTATION

2.2.1 CHECKSUM

Checksum is calculated by reading the contents of the PIC16C5X memory locations and adding up the opcodes up to the maximum user addressable location, e.g., 0x1FF for the PIC16C54/55. 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 PIC16C5X devices is shown in Table 2-3.

Programming Specification

TABLE 2-3: CHECKSUM COMPUTATION

Device	Code Protect	Checksum*	Blank Value	0x723 at 0 and max address
PIC16C54	OFF ON	SUM[0x000:0x1FF] + CFGW & 0xFFF SUM_XOR4[0x000:0x1FF] + CFGW & 0x00F	0x0DFF 0x1E07	0xFC47 0x1DF5
PIC16C54A	OFF ON	SUM[0x000:0x1FF] + CFGW & 0xFFF SUM_XOR4[0x000:0x1FF] + CFGW & 0x00F	0x0DFF 0x1E07	0xFC47 0x1DF5
PIC16CR54	OFF ON	SUM[0x000:0x1FF] + CFGW & 0xFFF SUM_XOR4[0x000:0x1FF] + CFGW & 0x00F	0x0DFF 0x1E07	— —
PIC16C55	OFF ON	SUM[0x000:0x1FF] + CFGW & 0xFFF SUM_XOR4[0x000:0x1FF] + CFGW & 0x00F	0x0DFF 0x1E07	0xFC47 0x1DF5
PIC16C56	OFF ON	SUM[0x000:0x3FF] + CFGW & 0xFFF SUM_XOR4[0x000:0x3FF] + CFGW & 0x00F	0x0BFF 0x3C07	0xFA47 0x3BF5
PIC16C57	OFF ON	SUM[0x000:0x7FF] + CFGW & 0xFFF SUM_XOR4[0x000:0x7FF] + CFGW & 0x00F	0x07FF 0x7807	0xF647 0x77F5
PIC16CR57A	OFF ON	SUM[0x000:0x7FF] + CFGW & 0xFFF SUM_XOR4[0x000:0x7FF] + CFGW & 0x00F	0x07FF 0x7807	— —
PIC16C58A	OFF ON	SUM[0x000:0x7FF] + CFGW & 0xFFF SUM_XOR4[0x000:0x7FF] + CFGW & 0x00F	0x07FF 0x7807	0xF647 0x77F5
PIC16CR58A	OFF ON	SUM[0x000:0x7FF] + CFGW & 0xFFF SUM[0x000:0x3F] + CFGW & 0xFFF + SUM_ID	0x07FF 0x800E	— —

Legend: CFGW = Configuration Word

SUM[a:b] = Sum of locations a through b inclusive

SUM_XOR4[a:b] = XOR of the four high order bits with the four middle and the four low of memory location order bits summed over the locations a through b inclusive. For example, location_a = 0x123 and location_b = 0x456, then SUM_XOR [location_a : location_b] = 0x0007.

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 individual expressions modulo [0xFFFF]

2.3 Embedding Configuration Word and ID Information in the Hex File

To allow portability of code, a PIC16C5X 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. Configuration word should have the address of 0xFFF. ID locations are mapped at addresses described in Section 1.6.1 and Table 2-2. 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.

PIC16C5X

3.0 PIC16C5X HEX DATA FORMATS

MPASM assembler is capable of outputting several different object file formats, suitable for a variety of programmers. A PIC16C5X programmer must be able to accept and send data in one of the following formats: 8-bit Split Format (.HXL/.HXH) and Intel® HEX Format (.HEX). The 8-bit merged (INHX8M) format is preferred.

3.1 8-bit Split Format (.HXL/.HXH)

The Intellec™ split 8-bit file format produces two output files: .HXL and .HXH. The format is the same as the normal 8-bit format, except that the low bytes of the data word are stored in the .HXL file, and the high bytes of the data word are stored in the .HXH file.

EXAMPLE 3-1:

```
<FILENAME>. HXL:
:0A0000000000000000000000000000F6
:0A0000000000000000000000000000F6
:1000190000284068A8E8C82868A989EA28086ABFAA
:10002900E0E82868BFE8C8080808034303E8E8FFD0
:03003900FFFF19AD
:00000001FF
```

EXAMPLE 3-2:

```
<FILENAME>. HXH:
:0A0000000000000000000000000000F6
:0A0000000000000000000000000000F6
:10001900000000000000000101010101020202CA
:100029000202030303030304040404050607070883
:0300390008080AAA
:00000001FF
```

3.2 Intel HEX Format (.HEX)

This format produces one 8-bit HEX file with a low byte, high byte combination. Since each address can only contain 8 bits in this format, all addresses are doubled. This format is useful for transferring PIC16C5X series object code to third party EPROM programmers.

EXAMPLE 3-3:

```
<FILENAME>.HEX
:040010000000000000EC
:1000000000000000000000000000000000000000000000F0
:040010000000000000EC
:100032000000280040006800A800E800C80028016D
:100042006801A9018901EA01280208026A02BF02C5
:10005200E002E80228036803BF03E803C8030804B8
:1000620008040804030443050306E807E807FF0839
:06007200FF08FF08190A57
:00000001FF
```

3.3 8-Bit Word Format

Each data record begins with a nine character prefix and ends with a two character checksum. Each record has the following format:

```
:BBAAAATTHHHH...HHHCC
```

Where:

- BB Two-digit hexadecimal byte count representing the number of data words that will appear on the line.
- AAAA Four-digit hexadecimal address representing the starting address for the data record.
- TT Two-digit record type that will always be '00' except for the end-of-file record which is set to '01'.
- HH Two-digit hexadecimal data word.
- CC Two-digit hexadecimal checksum that is the two's complement of the sum of all preceding bytes in the record including the prefix.

Programming Specification

4.0 PROGRAM/VERIFY MODE ELECTRICAL CHARACTERISTICS

4.1 DC Program Characteristics

TABLE 4-1: DC CHARACTERISTICS (TA = +10°C TO +40°C)

Characteristics	Symbol	Min.	Typ.	Max.	Units	Conditions
Supply voltage during programming	VDDP	4.75	5.0	5.25	V	Note 1
Supply Current (from VDD)	IDDP			25.0	mA	VDD = 5.0V, Fosc1 = 5MHz
Supply Voltage during verify	VDDV	VDD min		VDD max		
Voltage on MCLR to stay in Program/Verify Modes	VHH1	VDD + 3		VDD+7.25	V	
Voltage on MCLR during programming	VHH2	12.5		13.5	V	
Supply current from programming voltage source	IHH			100	mA	
Current into MCLR pin during programming (T0CKI=0)	IHH2		10.0	25.0	mA	VHH = 13.5V, VDD = 6.0V
Input Low Voltage	VIL	VSS		0.15VDD	V	
Input High Voltage	VIH	0.85VDD	5.0	VDD	V	

Note 1: Device must be verified at minimum and maximum operating voltages specified in the data sheet.

4.2 AC Program and Test Mode Characteristics

TABLE 4-2: AC CHARACTERISTICS (TA = +10°C TO +40°C, VDD = 5.0V ± 5%)

Characteristics	Symbol	Min	Typ	Max	Units	Conditions
MCLR Rise Time	TR	0.15	1.0	5	μs	
MCLR Fall Time	TF	0.5	2.0	5	μs	
Program Mode Setup Time	TPS	1.0			μs	
Data Access Time	TACC			250	ns	
Data Setup Time	TDS	1.0			μs	
Data Hold Time	TDH	1.0			μs	
Output Enable Time	TOE	0		100	ns	
Output Disable Time	TOZ	0		100	ns	
Programming Pulse Width	TPW	10.0			μs	
Programming Pulse Width	TPWF		10,000		μs	Configuration Word only
Recovery Time	TRC	10.0			μs	
Frequency on OSC1	FOSC	DC		5	MHz	For incrementing of the PC

4.3 Timing Diagrams

FIGURE 4-1: PROGRAMMING AND VERIFY TIMING WAVEFORM

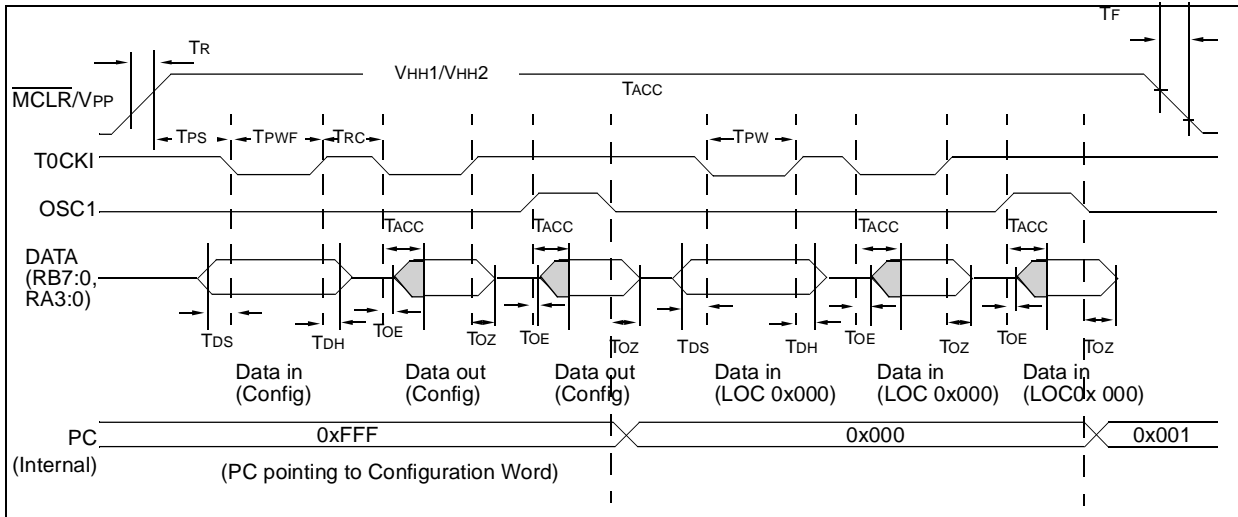
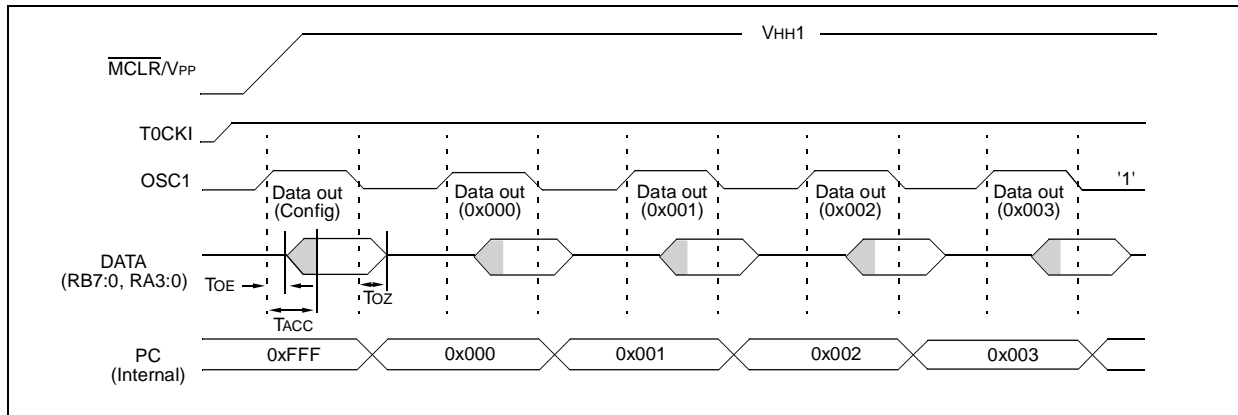


FIGURE 4-2: SPEED VERIFY TIMING WAVEFORM



Programming Specification

NOTES:

Worldwide Sales and 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

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.
35 Rockridge Road
Englewood, OH 45322
Tel: 513 832-2543 Fax: 513 832-2841

Los Angeles

Microchip Technology Inc.
18201 Von Karman, Suite 455
Irvine, CA 92715
Tel: 714 263-1888 Fax: 714 263-1338

New York

Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516 273-5305 Fax: 516 273-5335

AMERICAS (continued)

San Jose

Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408 436-7950 Fax: 408 436-7955

ASIA/PACIFIC

Hong Kong

Microchip Technology
Unit No. 3002-3004, Tower 1
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 0718
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 0 1628 851077 Fax: 44 0 1628 850259

France

Arizona Microchip Technology SARL
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 Pegaso Ingresso No. 2
Via Paracelso 23, 20041
Agrate Brianza (MI) Italy
Tel: 39 039 689 9939 Fax: 39 039 689 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

Also Available are:

Microchip Electronic Bulletin Board Service (BBS)

MCHIPBBS on CompuServe®

Internet Address

<http://www.mchip.com/biz/mchip>

8/18/95



MICROCHIP

Printed in USA © 1995, Microchip Technology Incorporated. All Rights Reserved.

"Information contained in this publication regarding device applications and the like is intended through 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 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. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.