



Welcome to E-XFL.COM

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	-
Peripherals	POR, WDT
Number of I/O	13
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c558-04i-so

Table of Contents

1.0	General Description.....	5
2.0	PIC16C55X Device Varieties	7
3.0	Architectural Overview	9
4.0	Memory Organization	13
5.0	I/O Ports	23
6.0	Special Features of the CPU	31
7.0	Timer0 Module	47
8.0	Instruction Set Summary	53
9.0	Development Support.....	67
10.0	Electrical Specifications.....	73
11.0	Packaging Information.....	87
Appendix A:	Enhancements.....	97
Appendix B:	Compatibility	97
Index		99
On-Line Support.....		101
Systems Information and Upgrade Hot Line		101
Reader Response		102
Product Identification System		103

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at docerrors@mail.microchip.com or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

<http://www.microchip.com>

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at www.microchip.com/cn to receive the most current information on all of our products.

2.0 PIC16C55X DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in the PIC16C55X Product Identification System section at the end of this data sheet. When placing orders, please use this page of the data sheet to specify the correct part number.

2.1 UV Erasable Devices

The UV erasable version, offered in CERDIP package, is optimal for prototype development and pilot programs. This version can be erased and reprogrammed to any of the oscillator modes.

Microchip's PICSTART[®] and PROMATE[®] programmers both support programming of the PIC16C55X.

2.2 One-Time Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers who need the flexibility for frequent code updates and small volume applications. In addition to the program memory, the configuration bits must also be programmed.

2.3 Quick-Turnaround Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium-to-high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices, but with all EPROM locations and configuration options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. Please contact your Microchip Technology sales office for more details.

2.4 Serialized Quick-Turnaround Production (SQTPSM) Devices

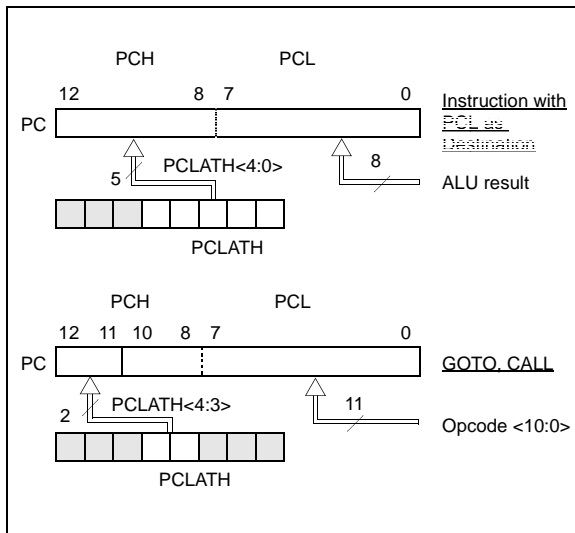
Microchip offers a unique programming service where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number which can serve as an entry code, password or ID number.

4.3 PCL and PCLATH

The program counter (PC) is 13-bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high bits (PC<12:8>) are not directly readable or writable and come from PCLATH. On any RESET, the PC is cleared. Figure 4-6 shows the two situations for the loading of the PC. The upper example in the figure shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 4-6 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

FIGURE 4-6: LOADING OF PC IN DIFFERENT SITUATIONS



4.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256 byte block). Refer to the application note "Implementing a Table Read" (AN556).

4.3.2 STACK

The PIC16C55X family has an 8-level deep x 13-bit wide hardware stack (Figure 4-1 and Figure 4-2). The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

Note 1: There are no status bits to indicate stack overflow or stack underflow conditions.

2: There are no instructions mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions, or vectoring to an interrupt address.

4.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the file select register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 4-7. However, IRP is not used in the PIC16C55X.

A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 4-1.

EXAMPLE 4-1: INDIRECT ADDRESSING

```

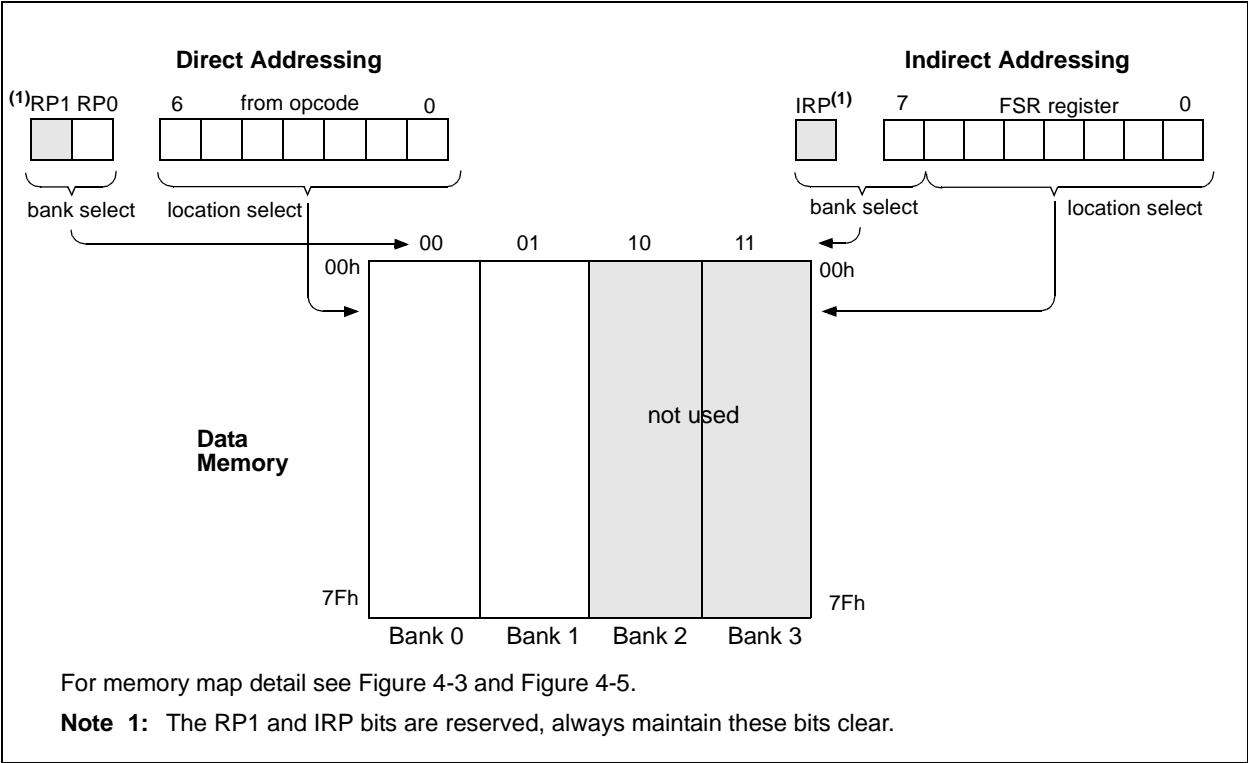
                                movlw    0x20    ;initialize pointer
                                movwf    FSR      ;to RAM
NEXT                          clrf      INDF     ;clear INDF register
                                incf      FSR      ;inc pointer
                                btfss    FSR,4    ;all done?
                                goto      NEXT     ;no clear next
                                           ;yes continue

```

CONTINUE:

PIC16C55X

FIGURE 4-7: DIRECT/INDIRECT ADDRESSING PIC16C55X



5.3 PORTC and TRISC Registers⁽¹⁾

PORTC is a 8-bit wide latch. All pins have data direction bits (TRIS registers) which can configure these pins as input or output.

A '1' in the TRISC register puts the corresponding output driver in a Hi-impedance mode. A '0' in the TRISC register puts the contents of the output latch on the selected pin(s).

Reading the PORTC register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

FIGURE 5-5: BLOCK DIAGRAM OF PORT PINS RC<7:0>

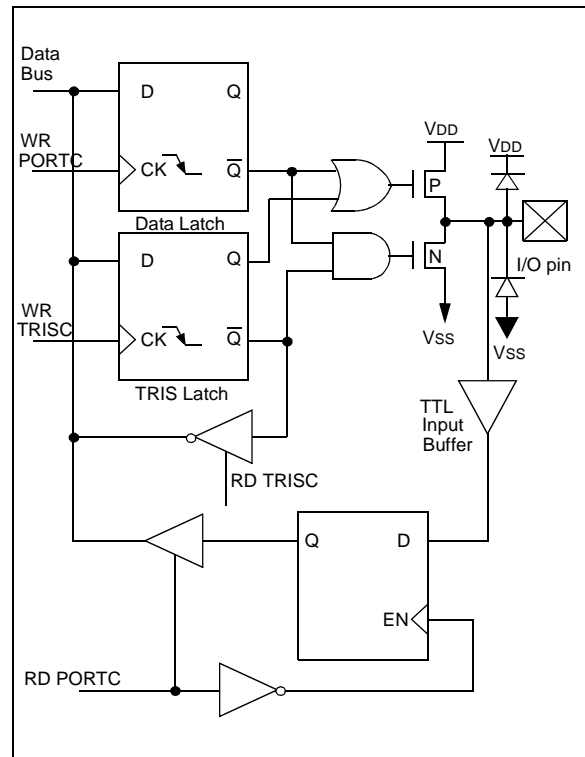


TABLE 5-5: PORTC FUNCTIONS

Name	Bit #	Buffer Type	Function
RC0	Bit 0	TTL	Bi-directional I/O port.
RC1	Bit 1	TTL	Bi-directional I/O port.
RC2	Bit 2	TTL	Bi-directional I/O port.
RC3	Bit 3	TTL	Bi-directional I/O port.
RC4	Bit 4	TTL	Bi-directional I/O port.
RC5	Bit 5	TTL	Bi-directional I/O port.
RC6	Bit 6	TTL	Bi-directional I/O port.
RC7	Bit 7	TTL	Bi-directional I/O port.

Legend: ST = Schmitt Trigger, TTL = TTL input

TABLE 5-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC AND TRISC

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on All Other RESETS
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
87h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged

Note 1: PIC16C557 ONLY.

EXAMPLE 5-1: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

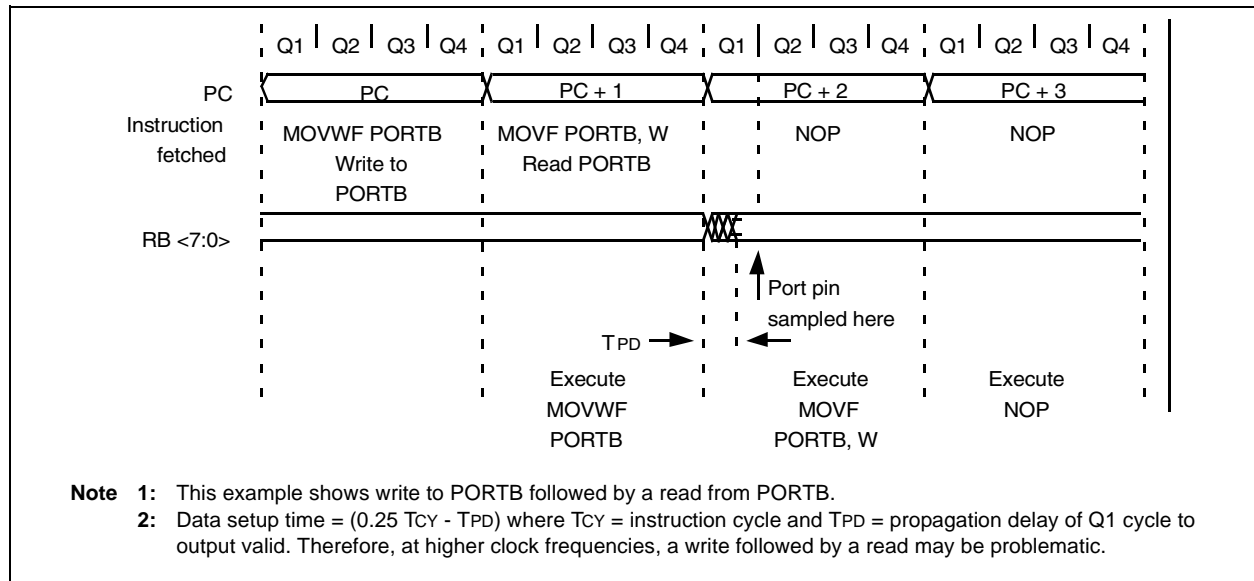
```
; Initial PORT settings: PORTB<7:4> Inputs
;
;                                PORTB<3:0> Outputs
; PORTB<7:6> have external pull-up and are
; not connected to other circuitry
;
;                                PORT latch PORT pins
;                                -----
;

BCF PORTB, 7    ; 01pp pppp 11pp pppp
BCF PORTB, 6    ; 10pp pppp 11pp pppp
BSF STATUS, RP0 ;
BCF TRISB, 7    ; 10pp pppp 11pp pppp
BCF TRISB, 6    ; 10pp pppp 10pp pppp
```

5.4.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle, as shown in Figure 5-6. Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such to allow the pin voltage to stabilize (load dependent) before the next instruction which causes that file to be read into the CPU is executed. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with an NOP or another instruction not accessing this I/O port.

FIGURE 5-6: SUCCESSIVE I/O OPERATION



PIC16C55X

REGISTER 6-1: CONFIGURATION WORD

CP1	CP0	CP1	CP0	CP1	CP0	—	Reserved	CP1	CP0	PWRTE	WDTE	FOSC1	FOSC0
bit 13													bit 0

bit 13-8 **CP<1:0>**: Code protection bits⁽¹⁾

bit 5-4
 11 = Program Memory code protection off
 10 = 0400h - 07FFh code protected
 01 = 0200h - 07FFh code protected
 11 = 0000h - 07FFh code protected

bit 7 **Unimplemented**: Read as '1'

bit 6 **Reserved**: Do not use

bit 3 **PWRTE**: Power-up Timer Enable bit
 1 = PWRT disabled
 0 = PWRT enabled

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

bit 1-0 **FOSC1:FOSC0**: Oscillator Selection bits
 11 = RC oscillator
 10 = HS oscillator
 01 = XT oscillator
 00 = LP oscillator

Note 1: All of the CP1:CP0 pairs have to be given the same value to enable the code protection scheme listed.

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR reset

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

6.4 Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST)

6.4.1 POWER-ON RESET (POR)

A Power-on Reset pulse is generated on-chip when V_{DD} rise is detected (in the range of 1.6V – 1.8V). To take advantage of the POR, just tie the \overline{MCLR} pin through a resistor to V_{DD} . This will eliminate external RC components usually needed to create Power-on Reset. A maximum rise time for V_{DD} is required. See Electrical Specifications for details.

The POR circuit does not produce internal RESET when V_{DD} declines.

When the device starts normal operation (exits the RESET condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in RESET until the operating conditions are met.

For additional information, refer to Application Note AN607 “Power-up Trouble Shooting”.

6.4.2 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 72 ms (nominal) timeout on power-up only, from POR. The Power-up Timer operates on an internal RC oscillator. The chip is kept in RESET as long as PWRT is active. The PWRT delay allows the V_{DD} to rise to an acceptable level. A configuration bit, \overline{PWRTE} can disable (if set) or enable (if cleared or programmed) the Power-up Timer. The Power-Up Time delay will vary from chip to chip and due to V_{DD} , temperature and process variation. See DC parameters for details.

6.4.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-Up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This ensures that the crystal oscillator or resonator has started and stabilized.

The OST timeout is invoked only for XT, LP and HS modes and only on Power-on Reset or wake-up from SLEEP.

6.4.4 TIMEOUT SEQUENCE

On power-up, the timeout sequence is as follows: First PWRT timeout is invoked after POR has expired, then OST is activated. The total timeout will vary based on oscillator configuration and \overline{PWRTE} bit status. For example, in RC mode with \overline{PWRTE} bit erased (PWRT disabled), there will be no timeout at all. Figure 6-7, Figure 6-8 and Figure 6-9 depict timeout sequences.

Since the timeouts occur from the POR pulse, if \overline{MCLR} is kept low long enough, the timeouts will expire. Then bringing \overline{MCLR} high will begin execution immediately (see Figure 6-8). This is useful for testing purposes or to synchronize more than one PIC16C55X device operating in parallel.

Table 6-5 shows the RESET conditions for some special registers, while Table 6-6 shows the RESET conditions for all the registers.

PIC16C55X

FIGURE 6-9: TIMEOUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ TIED TO V_{DD}): CASE 3

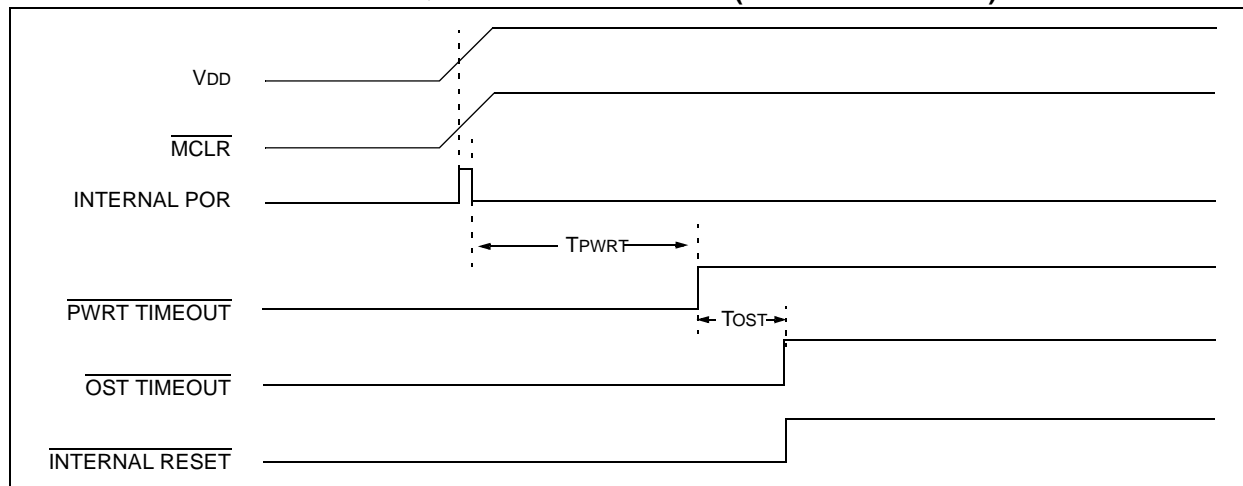
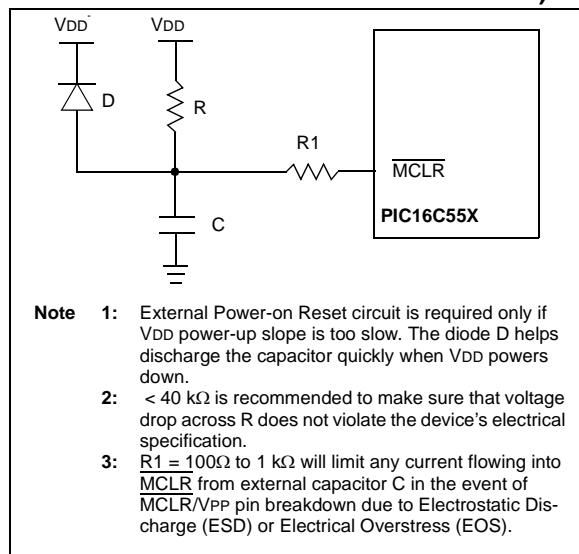


FIGURE 6-10: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW V_{DD} POWER-UP)



PIC16C55X

NOTES:

8.0 INSTRUCTION SET SUMMARY

Each PIC16C55X instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16C55X instruction set summary in Table 8-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 8-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For **literal and control** operations, 'k' represents an eight or eleven bit constant or literal value.

TABLE 8-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
label	Label name
TOS	Top of Stack
PC	Program Counter
PCLATH	Program Counter High Latch
GIE	Global Interrupt Enable bit
WDT	Watchdog Timer/Counter
TO	Timeout bit
PD	Power-down bit
dest	Destination either the W register or the specified register file location
[]	Options
()	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
<i>italics</i>	User defined term (font is courier)

The instruction set is highly orthogonal and is grouped into three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs.

Table 8-1 lists the instructions recognized by the MPASM™ assembler.

Figure 8-1 shows the three general formats that the instructions can have.

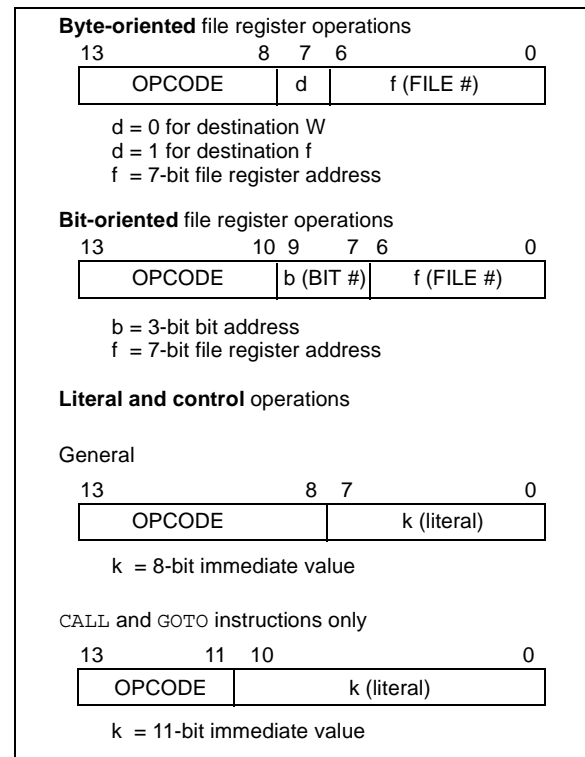
Note: To maintain upward compatibility with future PIC® MCU products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

FIGURE 8-1: GENERAL FORMAT FOR INSTRUCTIONS



PIC16C55X

RETFIE Return from Interrupt

Syntax:	[<i>label</i>] RETFIE				
Operands:	None				
Operation:	TOS → PC, 1 → GIE				
Status Affected:	None				
Encoding:	<table><tr><td>00</td><td>0000</td><td>0000</td><td>1001</td></tr></table>	00	0000	0000	1001
00	0000	0000	1001		
Description:	Return from Interrupt. Stack is POPed and Top of Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two-cycle instruction.				
Words:	1				
Cycles:	2				
Example	RETFIE After Interrupt PC = TOS GIE = 1				

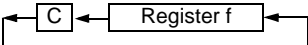
RETURN Return from Subroutine

Syntax:	[<i>label</i>] RETURN				
Operands:	None				
Operation:	TOS → PC				
Status Affected:	None				
Encoding:	<table><tr><td>00</td><td>0000</td><td>0000</td><td>1000</td></tr></table>	00	0000	0000	1000
00	0000	0000	1000		
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.				
Words:	1				
Cycles:	2				
Example	<pre>RETURN After Interrupt PC = TOS</pre>				

RETLW Return with Literal in W

Syntax:	[<i>label</i>] RETLW k			
Operands:	$0 \leq k \leq 255$			
Operation:	$k \rightarrow (W)$; TOS \rightarrow PC			
Status Affected:	None			
Encoding:	11	01xx	kkkk	kkkk
Description:	The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.			
Words:	1			
Cycles:	2			
Example	<pre>CALL TABLE;W contains table ;offset value ;W now has table value . . . ADDWF PC ;W = offset TABLE RETLW k1 ;Begin table RETLW k2 ; . . RETLW kn ; End of table</pre>			
	Before Instruction			
	W = 0x07			
	After Instruction			
	W = value of k8			

RLF Rotate Left f through Carry

Syntax:	[<i>label</i>] RLF <i>f</i> , <i>d</i>				
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$				
Operation:	See description below				
Status Affected:	C				
Encoding:	<table><tr><td>00</td><td>1101</td><td>dfff</td><td>ffff</td></tr></table>	00	1101	dfff	ffff
00	1101	dfff	ffff		
Description:	<p>The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is stored back in register 'f'.</p> 				
Words:	1				
Cycles:	1				
Example	<pre>RLF REG1,0</pre> <p>Before Instruction</p> <pre>REG1 = 1110 0110 C = 0</pre> <p>After Instruction</p> <pre>REG1 = 1110 0110 W = 1100 1100 C = 1</pre>				

PIC16C55X

SUBWF Subtract W from f

Syntax:	[<i>label</i>] SUBWF f,d				
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$				
Operation:	$(f) - (W) \rightarrow (\text{dest})$				
Status	C, DC, Z				
Affected:					
Encoding:	<table><tr><td>00</td><td>0010</td><td>dfff</td><td>ffff</td></tr></table>	00	0010	dfff	ffff
00	0010	dfff	ffff		
Description:	Subtract (2's complement method) W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.				
Words:	1				
Cycles:	1				
Example 1:	SUBWF REG1,1				

Before Instruction

REG1 = 3
W = 2
C = ?

After Instruction

REG1 = 1
W = 2
C = 1; result is positive

Example 2:	Before Instruction
	REG1 = 2
	W = 2
	C = ?
	After Instruction
	REG1 = 0
	W = 2
	C = 1; result is zero

Example 3:	Before Instruction
	REG1 = 1
	W = 2
	C = ?
	After Instruction
	REG1 = 0xFF
	W = 2
	C = 0; result is negative

SWAPF Swap Nibbles in f

Syntax:	[<i>label</i>] SWAPF f,d			
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$			
Operation:	$(f<3:0>) \rightarrow (\text{dest}<7:4>),$ $(f<7:4>) \rightarrow (\text{dest}<3:0>)$			
Status Affected:	None			
Encoding:	00	1110	dfff	ffff
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in W register. If 'd' is 1 the result is placed in register 'f'.			
Words:	1			
Cycles:	1			
Example	SWAPF REG, 0			

Before Instruction

REG1 = 0xA5

After Instruction

REG1 = 0xA5
W = 0x5A

TRIS	Load TRIS Register				
Syntax:	[<i>label</i>] TRIS f				
Operands:	$5 \leq f \leq 7$				
Operation:	(W) → TRIS register f;				
Status Affected:	None				
Encoding:	<table><tr><td>00</td><td>0000</td><td>0110</td><td>0fff</td></tr></table>	00	0000	0110	0fff
00	0000	0110	0fff		
Description:	The instruction is supported for code compatibility with the PIC16C5X products. Since TRIS registers are readable and writable, the user can directly address them.				
Words:	1				
Cycles:	1				
Example	<table><tr><td>To maintain upward compatibility with future PIC MCU products, do not use this instruction.</td></tr></table>	To maintain upward compatibility with future PIC MCU products, do not use this instruction.			
To maintain upward compatibility with future PIC MCU products, do not use this instruction.					

XORLW Exclusive OR Literal with W

Syntax:	[<i>label</i>] XORLW <i>k</i>				
Operands:	$0 \leq k \leq 255$				
Operation:	(W) .XOR. <i>k</i> \rightarrow (W)				
Status Affected:	Z				
Encoding:	<table border="1"><tr><td>11</td><td>1010</td><td>kkkk</td><td>kkkk</td></tr></table>	11	1010	kkkk	kkkk
11	1010	kkkk	kkkk		
Description:	The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.				
Words:	1				
Cycles:	1				
Example:	<pre>XORLW 0xAF</pre> <p>Before Instruction</p> <p>W = 0xB5</p> <p>After Instruction</p> <p>W = 0x1A</p>				

XORWF Exclusive OR W with f

Syntax:	[<i>label</i>] XORWF <i>f</i> , <i>d</i>				
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$				
Operation:	(W) .XOR. (<i>f</i>) \rightarrow (<i>dest</i>)				
Status Affected:	Z				
Encoding:	<table border="1"><tr><td>00</td><td>0110</td><td>dfff</td><td>ffff</td></tr></table>	00	0110	dfff	ffff
00	0110	dfff	ffff		
Description:	Exclusive OR the contents of the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.				
Words:	1				
Cycles:	1				
Example	<pre>XORWF REG 1</pre> <p>Before Instruction</p> <p>REG = 0xAF W = 0xB5</p> <p>After Instruction</p> <p>REG = 0x1A W = 0xB5</p>				

TABLE 10-2: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter #	Sym	Characteristic	Min	Typ†	Max	Units
10*	TosH2ckL	OSC1↑ to CLKOUT↓ ⁽¹⁾	— —	75 —	200 400	ns ns
11*	TosH2ckH	OSC1↑ to CLKOUT↑ ⁽¹⁾	— —	75 —	200 400	ns ns
12*	TckR	CLKOUT rise time ⁽¹⁾	— —	35 —	100 200	ns ns
13*	TckF	CLKOUT fall time ⁽¹⁾	— —	35 —	100 200	ns ns
14*	TckL2ioV	CLKOUT ↓ to Port out valid ⁽¹⁾	—	—	20	ns
15*	TioV2ckH	Port in valid before CLKOUT ↑ ⁽¹⁾	Tosc +200 ns Tosc +400 ns	— —	— —	ns ns
16*	TckH2ioI	Port in hold after CLKOUT ↑ ⁽¹⁾	0	—	—	ns
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	— —	50	150 300	ns ns
18*	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	100 200	— —	— —	ns ns
19*	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	0	—	—	ns
20*	TioR	Port output rise time	— —	10 —	40 80	ns ns
21*	TioF	Port output fall time	— —	10 —	40 80	ns ns
22*	Tinp	RB0/INT pin high or low time	25 40	— —	— —	ns ns
23*	Trbp	RB<7:4> change interrupt high or low time	Tcy	—	—	ns

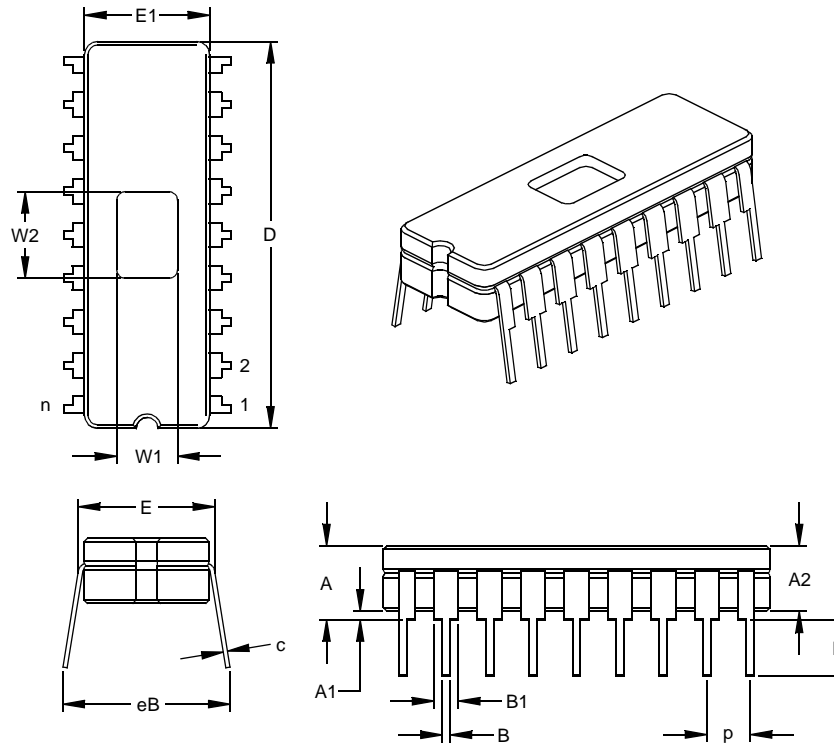
* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Measurements are taken in RC mode where CLKOUT output is 4 x Tosc.

18-Lead Ceramic Dual In-line with Window (JW) – 300 mil (CERDIP)

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



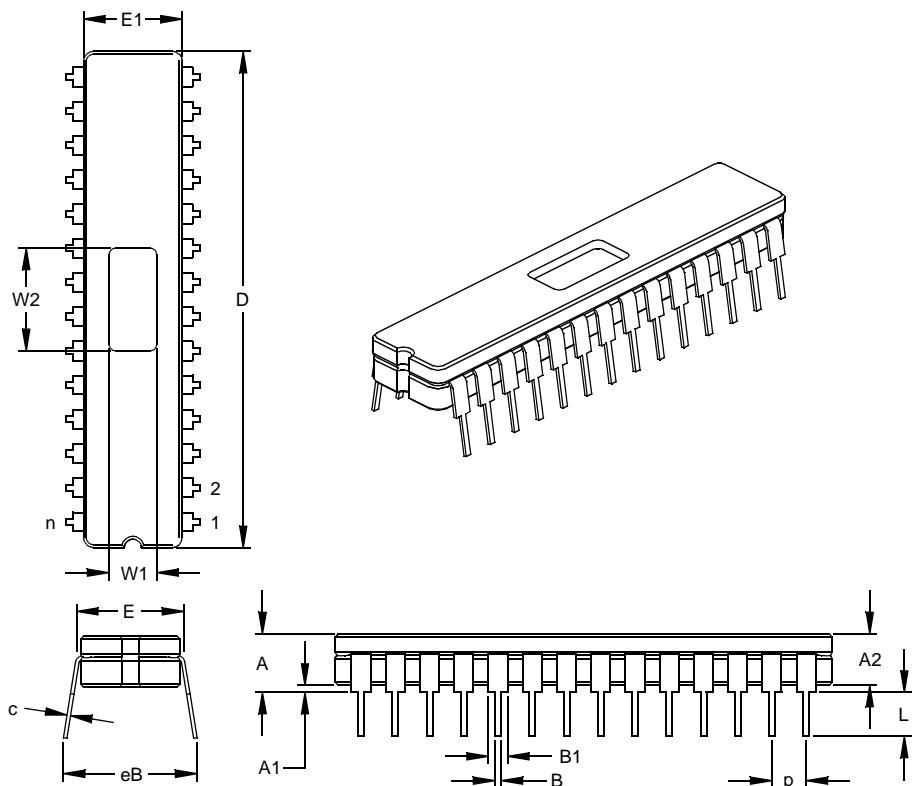
Units		INCHES*			MILLIMETERS		
Dimension	Limits	MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.170	.183	.195	4.32	4.64	4.95
Ceramic Package Height	A2	.155	.160	.165	3.94	4.06	4.19
Standoff	A1	.015	.023	.030	0.38	0.57	0.76
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Ceramic Pkg. Width	E1	.285	.290	.295	7.24	7.37	7.49
Overall Length	D	.880	.900	.920	22.35	22.86	23.37
Tip to Seating Plane	L	.125	.138	.150	3.18	3.49	3.81
Lead Thickness	c	.008	.010	.012	0.20	0.25	0.30
Upper Lead Width	B1	.050	.055	.060	1.27	1.40	1.52
Lower Lead Width	B	.016	.019	.021	0.41	0.47	0.53
Overall Row Spacing	§ eB	.345	.385	.425	8.76	9.78	10.80
Window Width	W1	.130	.140	.150	3.30	3.56	3.81
Window Length	W2	.190	.200	.210	4.83	5.08	5.33

* Controlling Parameter
 § Significant Characteristic
 JEDEC Equivalent: MO-036
 Drawing No. C04-010

PIC16C55X

28-Lead Ceramic Dual In-line with Window (JW) – 300 mil (CERDIP)

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



		Units	INCHES*			MILLIMETERS		
Dimension Limits			MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n			28			28	
Pitch	p			.100			2.54	
Top to Seating Plane	A		.170	.183	.195	4.32	4.64	4.95
Ceramic Package Height	A2		.155	.160	.165	3.94	4.06	4.19
Standoff	A1		.015	.023	.030	0.38	0.57	0.76
Shoulder to Shoulder Width	E		.300	.313	.325	7.62	7.94	8.26
Ceramic Pkg. Width	E1		.285	.290	.295	7.24	7.37	7.49
Overall Length	D		1.430	1.458	1.485	36.32	37.02	37.72
Tip to Seating Plane	L		.135	.140	.145	3.43	3.56	3.68
Lead Thickness	c		.008	.010	.012	0.20	0.25	0.30
Upper Lead Width	B1		.050	.058	.065	1.27	1.46	1.65
Lower Lead Width	B		.016	.019	.021	0.41	0.47	0.53
Overall Row Spacing	§ eB		.345	.385	.425	8.76	9.78	10.80
Window Width	W1		.130	.140	.150	3.30	3.56	3.81
Window Length	W2		.290	.300	.310	7.37	7.62	7.87

* Controlling Parameter

§ Significant Characteristic

JEDEC Equivalent: MO-058

Drawing No. C04-080

APPENDIX A: ENHANCEMENTS

The following are the list of enhancements over the PIC16C5X microcontroller family:

1. Instruction word length is increased to 14 bits. This allows larger page sizes both in program memory (4K now as opposed to 512 before) and register file (up to 128 bytes now versus 32 bytes before).
2. A PC high latch register (PCLATH) is added to handle program memory paging. PA2, PA1, PA0 bits are removed from STATUS register.
3. Data memory paging is slightly redefined. STATUS register is modified.
4. Four new instructions have been added: RETURN, RETFIE, ADDLW, and SUBLW. Two instructions TRIS and OPTION are being phased out although they are kept for compatibility with PIC16C5X.
5. OPTION and TRIS registers are made addressable.
6. Interrupt capability is added. Interrupt vector is at 0004h.
7. Stack size is increased to 8 deep.
8. RESET vector is changed to 0000h.
9. RESET of all registers is revised. Three different RESET (and wake-up) types are recognized. Registers are reset differently.
10. Wake-up from SLEEP through interrupt is added.
11. Two separate timers, Oscillator Start-up Timer (OST) and Power-up Timer (PWRT) are included for more reliable power-up. These timers are invoked selectively to avoid unnecessary delays on power-up and wake-up.
12. PORTB has weak pull-ups and interrupt-on-change feature.
13. Timer0 clock input, T0CKI pin is also a port pin (RA4/T0CKI) and has a TRIS bit.
14. FSR is made a full 8-bit register.
15. "In-circuit programming" is made possible. The user can program PIC16C55X devices using only five pins: VDD, VSS, VPP, RB6 (clock) and RB7 (data in/out).
16. PCON status register is added with a Power-on Reset (POR) status bit.
17. Code protection scheme is enhanced such that portions of the program memory can be protected, while the remainder is unprotected.
18. PORTA inputs are now Schmitt Trigger inputs.

APPENDIX B: COMPATIBILITY

To convert code written for PIC16C5X to PIC16C55X, the user should take the following steps:

1. Remove any program memory page select operations (PA2, PA1, PA0 bits) for CALL, GOTO.
2. Revisit any computed jump operations (write to PC or add to PC, etc.) to make sure page bits are set properly under the new scheme.
3. Eliminate any data memory page switching. Redefine data variables to reallocate them.
4. Verify all writes to STATUS, OPTION, and FSR registers since these have changed.
5. Change RESET vector to 0000h.

APPENDIX C: REVISION HISTORY

Revision E (January 2013)

Added a note to each package outline drawing.

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

<u>PART NO.</u>	<u>X</u>	<u>/XX</u>	<u>XXX</u>
Device	Temperature Range	Package	Pattern
Device	PIC17C756: Standard VDD range PIC17C756T: (Tape and Reel) PIC17LC756: Extended VDD range		
Temperature Range	- = 0°C to +70°C I = -40°C to +85°C		
Package	CL = Windowed LCC PT = TQFP L = PLCC		
Pattern	QTP, SQTP, ROM Code (factory specified) or Special Requirements. Blank for OTP and Windowed devices.		

Examples:

- a) PIC17C756-16L Commercial Temp., PLCC package, 16 MHz, normal VDD limits
- b) PIC17LC756-08/PT Commercial Temp., TQFP package, 8MHz, extended VDD limits
- c) PIC17C756-33I/PT Industrial Temp., TQFP package, 33 MHz, normal VDD limits

* JW Devices are UV erasable and can be programmed to any device configuration. JW Devices meet the electrical requirement of each oscillator type.

Sales and Support

Data Sheets

Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Worldwide Site (www.microchip.com)

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families 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 Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is 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 products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC³² logo, rPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscent Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rLAB, Select Mode, SQL, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. & KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 1996-2013, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

ISBN: 9781620769737

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
= ISO/TS 16949 =

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.