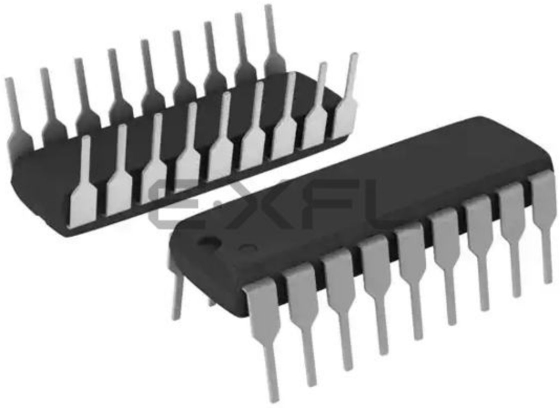


Welcome to E-XFL.COM

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)	2.5V ~ 5.5V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	18-DIP (0.300", 7.62mm)
Supplier Device Package	18-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lc558-04i-p

Special Microcontroller Features:

- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Serial in-circuit programming (via two pins)
- Four user programmable ID locations

Note: For additional information on enhancements, see Appendix A

CMOS Technology:

- Low power, high speed CMOS EPROM technology
- Fully static design
- Wide operating voltage range
  - 2.5V to 5.5V
- Commercial, Industrial and Extended temperature range
- Low power consumption
  - < 2.0 mA @ 5.0V, 4.0 MHz
  - 15 µA typical 3.0V, 32 kHz
  - < 1.0 µA typical standby current @ 3.0V

Device Differences

Device	Voltage Range	Oscillator
PIC16C554	2.5 - 5.5	**(Note 1)**
PIC16C557	2.5 - 5.5	**(Note 1)**
PIC16C558	2.5 - 5.5	**(Note 1)**

Note 1: If you change from this device to another device, please verify oscillator characteristics in your application.

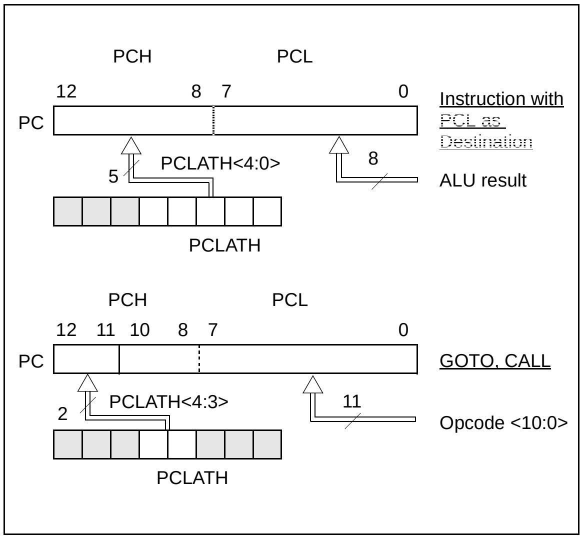
NOTES:

NOTES:

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:
```

5.4 I/O Programming Considerations

5.4.1 BI-DIRECTIONAL I/O PORTS

Any instruction which writes, operates internally as a read followed by a write operation. The `BCF` and `BSF` instructions, for example, read the register into the CPU, execute the bit operation and write the result back to the register. Caution must be used when these instructions are applied to a port with both inputs and outputs defined. For example, a `BSF` operation on bit5 of PORTB will cause all eight bits of PORTB to be read into the CPU. Then the `BSF` operation takes place on bit5 and PORTB is written to the output latches. If another bit of PORTB is used as a bi-directional I/O pin (e.g., bit 0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and re-written to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the Input mode, no problem occurs. However, if bit 0 is switched into Output mode later on, the content of the data latch may now be unknown.

Reading the port register, reads the values of the port pins. Writing to the port register writes the value to the port latch. When using read-modify-write instructions (ex. `BCF, BSF`, etc.) on a port, the value of the port pins is read, the desired operation is done to this value, and this value is then written to the port latch.

Example 5-1 shows the effect of two sequential read-modify-write instructions (ex., `BCF`, `BSF`, etc.) on an I/O port.

A pin actively outputting a low or high should not be driven from external devices at the same time in order to change the level on this pin (“wired-or”, “wired-and”). The resulting high output currents may damage the chip.

NOTES:

6.0 SPECIAL FEATURES OF THE CPU

What sets a microcontroller apart from other processors are special circuits to deal with the needs of real-time applications. The PIC16C55X family has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection.

These are:

1. OSC selection
2. RESET
3. Power-on Reset (POR)
4. Power-up Timer (PWRT)
5. Oscillator Start-Up Timer (OST)
6. Interrupts
7. Watchdog Timer (WDT)
8. SLEEP
9. Code protection
10. ID Locations
11. In-circuit serial programming™

The PIC16C55X has a Watchdog Timer which is controlled by configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), which is intended to keep the chip in RESET until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only, designed to keep the part in RESET while the power supply stabilizes. With these two functions on-chip, most applications need no external RESET circuitry.

The SLEEP mode is designed to offer a very low current Power-down mode. The user can wake-up from SLEEP through external RESET, Watchdog Timer wake-up or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits are used to select various options.

6.1 Configuration Bits

The configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped in program memory location 2007h.

The user will note that address 2007h is beyond the user program memory space. In fact, it belongs to the special test/configuration memory space (2000h – 3FFFh), which can be accessed only during programming.

REGISTER 6-1: CONFIGURATION WORD

CP1	CP0	CP1	CP0	CP1	CP0	—	Reserved	CP1	CP0	PWRTE	WDTE	F0SC1	F0SC0
bit 13													bit 0

bit 13-8
bit 5-4

CP<1:0>: Code protection bits[(1)]
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 **$\overline{\text{PWRTE}}$**: 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

TABLE 6-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	000h	`0001 1xxx`	`---- --0-`
$\overline{\text{MCLR}}$ Reset during normal operation	000h	`000u uuuu`	`---- --u-`
$\overline{\text{MCLR}}$ Reset during SLEEP	000h	`0001 0uuu`	`---- --u-`
WDT Reset	000h	`0000 uuuu`	`---- --u-`
WDT Wake-up	PC + 1	`uuu0 0uuu`	`---- --u-`
Interrupt Wake-up from SLEEP	PC + 1(1)	`uuu1 0uuu`	`---- --u-`

Legend: `u` = unchanged, `x` = unknown, `-` = unimplemented bit, reads as ‘0’, `q` = value depends on condition.

Note 1: When the wake-up is due to an interrupt and global enable bit, GIE is set, the PC is loaded with the interrupt vector (0004h) after execution of PC+1.

TABLE 6-6: INITIALIZATION CONDITION FOR REGISTERS

Register	Address	Power-on Reset	$\overline{\text{MCLR}}$ Reset during normal operation $\overline{\text{MCLR}}$ Reset during SLEEP WDT Reset	Wake-up from SLEEP through interrupt Wake-up from SLEEP through WDT timeout
W	—	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
INDF	00h	—	—	—
TMR0	01h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
PCL	02h	`0000 0000`	`0000 0000`	`PC + 1`(2)
STATUS	03h	`0001 1xxx`	`000q quuu`(3)	`uuuq quuu`(3)
FSR	04h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
PORTA	05h	`---x xxxx`	`---u uuuu`	`---u uuuu`
PORTB	06h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
PORTC(4)	06h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
PCLATH	0Ah	`---0 0000`	`---0 0000`	`---u uuuu`
INTCON	0Bh	`0000 000x`	`0000 000u`	`uuuu uuuu`(1)
OPTION	81h	`1111 1111`	`1111 1111`	`uuuu uuuu`
TRISA	85h	`---1 1111`	`---1 1111`	`---u uuuu`
TRISB	86h	`1111 1111`	`1111 1111`	`uuuu uuuu`
TRISC(4)	86h	`1111 1111`	`1111 1111`	`uuuu uuuu`
PCON	8Eh	`---- --0-`	`---- --u-`	`---- --u-`

Legend: `u` = unchanged, `x` = unknown, `-` = unimplemented bit, reads as ‘0’, `q` = value depends on condition.

Note 1: One or more bits in INTCON will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

3: See Table 6-5 for RESET value for specific condition.

4: PIC16C557 only.

FIGURE 6-7: TIMEOUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO VDD): CASE 1

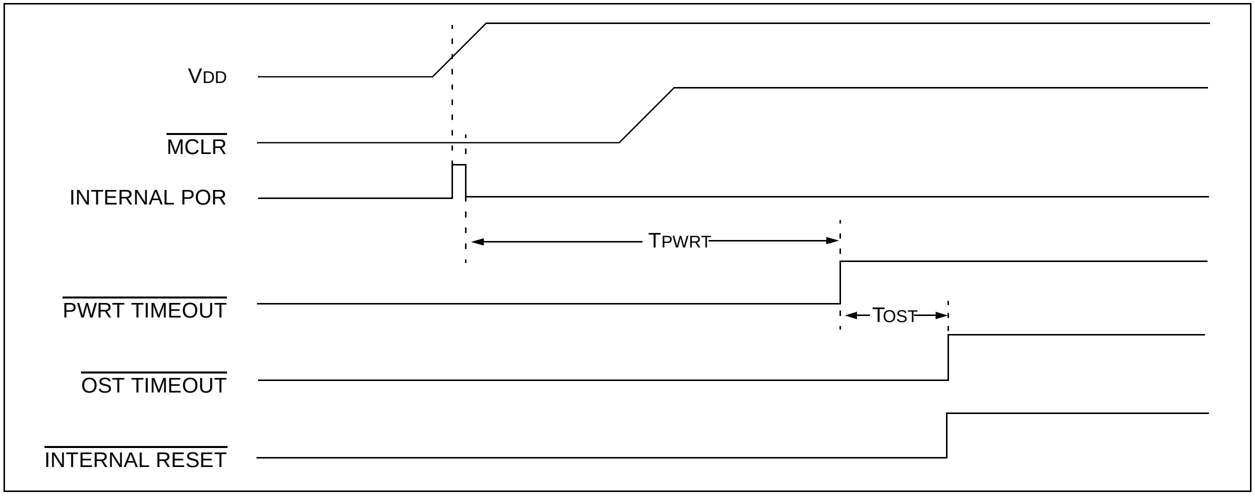
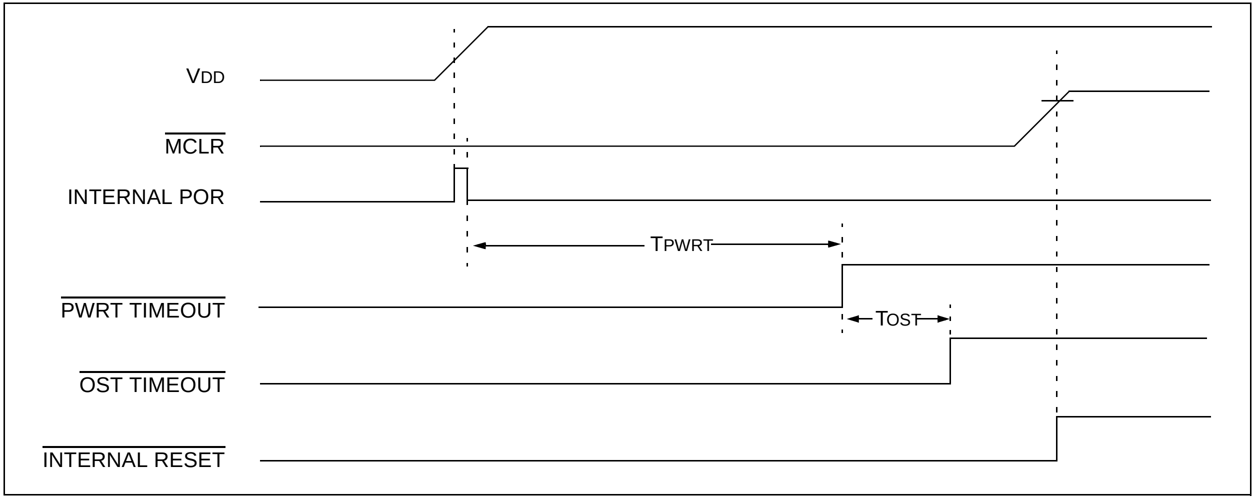

VDD
$\overline{\text{MCLR}}$
INTERNAL POR
TPWRT
$\overline{\text{PWRT TIMEOUT}}$
TOST
$\overline{\text{OST TIMEOUT}}$
$\overline{\text{INTERNAL RESET}}$

FIGURE 6-8: TIMEOUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO VDD): CASE 2


VDD
$\overline{\text{MCLR}}$
INTERNAL POR
TPWRT
$\overline{\text{PWRT TIMEOUT}}$
TOST
$\overline{\text{OST TIMEOUT}}$
$\overline{\text{INTERNAL RESET}}$

6.6 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt (e.g., W register and STATUS register). This will have to be implemented in software.

Example 6-1 stores and restores the STATUS and W registers. The user register, W_TEMP, must be defined in both banks and must be defined at the same offset from the bank base address (i.e., W_TEMP is defined at 0x20 in Bank 0 and it must also be defined at 0xA0 in Bank 1). The user register, STATUS_TEMP, must be defined in Bank 0. The Example 6-1:

- Stores the W register
- Stores the STATUS register in Bank 0
- Executes the ISR code
- Restores the STATUS (and bank select bit register)
- Restores the W register

EXAMPLE 6-1: SAVING THE STATUS AND W REGISTERS IN RAM

```
MOVWF  W_TEMP        ;copy W to TEMP
                     ;register, could be in
                     ;either bank
SWAPF  STATUS,W      ;swap STATUS to be
                     ;saved into W
BCF    STATUS,RP0    ;change to bank0
                     ;regardless of
                     ;current bank
MOVWF  STATUS_TEMP   ;save STATUS to bank0
                     ;register
:
:
:
SWAPF  STATUS_TEMP,W;swap STATUS_TEMP
                     ;register into W, sets
                     ;bank to original state
MOVWF  STATUS        ;move W into STATUS
                     ;register
SWAPF  W_TEMP,F      ;swap W_TEMP
SWAPF  W_TEMP,W      ;swap W_TEMP into W
```

6.7 Watchdog Timer (WDT)

The Watchdog Timer is a free running on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the CLKIN pin. That means that the WDT will run, even if the clock on the OSC1 and OSC2 pins of the device has been stopped, for example, by execution of a SLEEP instruction. During normal operation, a WDT timeout generates a device RESET. If the device is in SLEEP mode, a WDT timeout causes the device to wake-up and continue with normal operation. The WDT can be permanently disabled by programming the configuration bit WDTE as clear (Section 6.1).

6.7.1 WDT PERIOD

The WDT has a nominal timeout period of 18 ms, (with no prescaler). The timeout periods vary with temperature, VDD and process variations from part-to-part (see DC specs). If longer timeout periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT under software control by writing to the OPTION register. Thus, timeout periods up to 2.3 seconds can be realized.

The CLRWDT and SLEEP instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET.

The $\overline{\text{TO}}$ bit in the STATUS register will be cleared upon a Watchdog Timer timeout.

6.7.2 WDT PROGRAMMING CONSIDERATIONS

It should also be taken in account that under worst case conditions (VDD = Min., Temperature = Max., max. WDT prescaler) it may take several seconds before a WDT timeout occurs.

TABLE 8-2: PIC16C55X INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb			LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0000	0011	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1(2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1(2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into Standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

RETFIE — Return from Interrupt

Syntax: [*label*] RETFIE

Operands: None

Operation: TOS → PC,
1 → GIE

Status Affected: None

Encoding:

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

RETLW — Return with Literal in W

Syntax: [*label*] RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → (W);
TOS → 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

```
      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
```

Before Instruction

W	=	0x07

After Instruction

W	=	value of k8

RETURN — Return from Subroutine

Syntax: [*label*] RETURN

Operands: None

Operation: TOS → PC

Status Affected: None

Encoding:

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

```
RETURN
```

After Interrupt

PC	=	TOS

RLF — Rotate Left f through Carry

Syntax: [*label*] RLF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

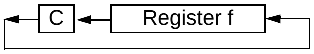
Operation: See description below

Status Affected: C

Encoding:

00	1101	dfff	ffff

Description: 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'.



Words: 1

Cycles: 1

Example

```
RLF     REG1,0
```

Before Instruction

REG1	=	1110 0110
C	=	0

After Instruction

REG1	=	1110 0110
W	=	1100 1100
C	=	1

XORLW Exclusive OR Literal with W

Syntax:	[*label*] XORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .XOR. k → (W)
Status Affected:	Z
Encoding:	`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:	`XORLW 0xAF`
	Before Instruction
	W = 0xB5
	After Instruction
	W = 0x1A

XORWF Exclusive OR W with f

Syntax:	[*label*] XORWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .XOR. (f) → (dest)
Status Affected:	Z
Encoding:	`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	`XORWF REG 1`
	Before Instruction
	REG = 0xAF
	W = 0xB5
	After Instruction
	REG = 0x1A
	W = 0xB5

10.1 DC Characteristics: PIC16C55X-04 (Commercial, Industrial, Extended)
PIC16C55X-20 (Commercial, Industrial, Extended)
HCS1365-04 (Commercial, Industrial, Extended)

DC Characteristics			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial and -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
D020	IPD	**Power-Down Current(3)**					
		16LC55X	—	0.7	2	μA	VDD = 3.0V, WDT disabled
		16C55X	—	1.0	2.5 15	μA μA	VDD = 4.0V, WDT disabled (+85°C to +125°C)
	ΔIWDT	**WDT Current(5)**					
		16LC55X	—	6.0	15	μA	VDD = 3.0V
		16C55X	—	6.0	20	μA	VDD = 4.0V (+85°C to +125°C)

* These parameters are characterized but not tested.

† Data is “Typ” column is at 5V, 25°C, unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active Operation mode are:
$\overline{\text{OSC1}}$ = external square wave, from rail to rail; all I/O pins configured as input, pulled to VDD,
$\overline{\text{MCLR}}$ = VDD; WDT enabled/disabled as specified.

3: The power-down current in SLEEP mode does not depend on the oscillator type. Power-down current is measured with the part in SLEEP mode, with all I/O pins configured as input and tied to VDD or VSS.

4: For RC osc configuration, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.

5: The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

10.2 DC Characteristics: PIC16C55X (Commercial, Industrial, Extended) PIC16LC55X(Commercial, Industrial, Extended)

DC Characteristics			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ T_A ≤ +85°C for industrial and 0°C ≤ T_A ≤ +70°C for commercial and -40°C ≤ T_A ≤ +125°C for automotive Operating voltage V_{DD} range as described in DC spec Table 10-1				
Param. No.	Sym	Characteristic	Min	Typ†	Max	Unit	Conditions
	V_{IL}	**Input Low Voltage**					
		I/O ports					
D030		with TTL buffer	V_{SS}	—	0.8V 0.15 V_{DD}	V	V_{DD} = 4.5V to 5.5V otherwise
D031		with Schmitt Trigger input	V_{SS}		0.2 V_{DD}	V	
D032		$\overline{\text{MCLR}}$, RA4/T0CKI,OSC1 (in RC mode)	V_{SS}	—	0.2 V_{DD}	V	**(Note1)**
D033		OSC1 (in XT* and HS)	V_{SS}	—	0.3 V_{DD}	V	
		OSC1 (in LP*)	V_{SS}	—	0.6 V_{DD}-1.0	V	
	V_{IH}	**Input High Voltage**					
		I/O ports		—			
D040		with TTL buffer	2.0V 0.8 + 0.25 V_{DD}	— —	V_{DD} V_{DD}	V V	V_{DD} = 4.5V to 5.5V otherwise
D041		with Schmitt Trigger input	0.8V		V_{DD}		
D042		$\overline{\text{MCLR}}$ RA4/T0CKI	0.8 V_{DD}	—	V_{DD}	V	
D043		OSC1 (XT*, HS and LP*)	0.7 V_{DD}	—	V_{DD}	V	
D043A		OSC1 (in RC mode)	0.9 V_{DD}				**(Note1)**
D070	I_{PURB}	PORTB weak pull-up current	50	200	400	μA	V_{DD} = 5.0V, V_{PIN} = V_{SS}
	I_{IL}	**Input Leakage Current(2)(3)**					
		I/O ports (Except PORTA)			±1.0	μA	V_{SS} ≤ V_{PIN} ≤ V_{DD}, pin at hi-impedance
D060		PORTA	—	—	±0.5	μA	V_{ss} ≤ V_{PIN} ≤ V_{DD}, pin at hi-impedance
D061		RA4/T0CKI	—	—	±1.0	μA	V_{ss} ≤ V_{PIN} ≤ V_{DD}
D063		OSC1, $\overline{\text{MCLR}}$	—	—	±5.0	μA	V_{ss} ≤ V_{PIN} ≤ V_{DD}, XT, HS and LP osc configuration
	V_{OL}	**Output Low Voltage**					
D080		I/O ports	—	—	0.6	V	I_{OL}=8.5 mA, V_{DD}=4.5V, -40° to +85°C
			—	—	0.6	V	I_{OL}=7.0 mA, V_{DD}=4.5V, +125°C
D083		OSC2/CLKOUT	—	—	0.6	V	I_{OL}=1.6 mA, V_{DD}=4.5V, -40° to +85°C
		(RC only)	—	—	0.6	V	I_{OL}=1.2 mA, V_{DD}=4.5V, +125°C
	V_{OH}	**Output High Voltage(3)**					
D090		I/O ports (Except RA4)	V_{DD}-0.7	—	—	V	I_{OH}=-3.0 mA, V_{DD}=4.5V, -40° to +85°C

* 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: In RC oscillator configuration, the OSC1 pin is a Schmitt Trigger input. It is not recommended that the PIC16C55X be driven with external clock in RC mode.

2: The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

FIGURE 10-8: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

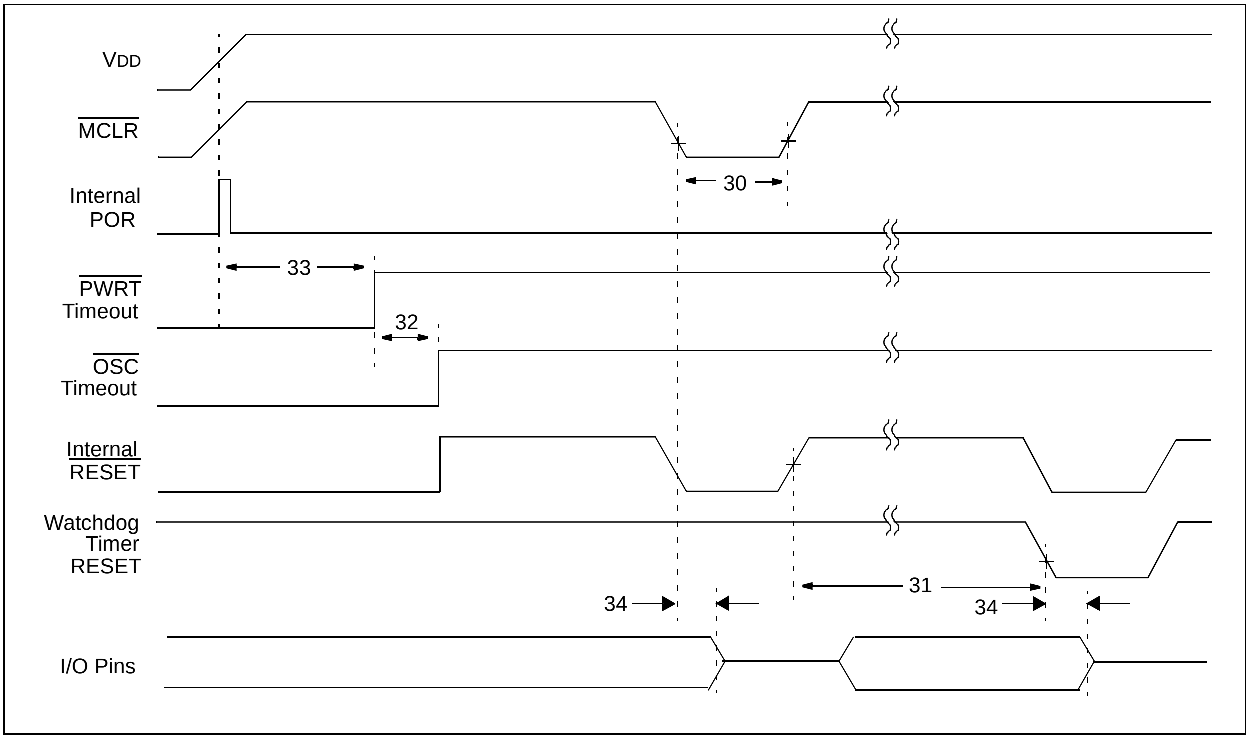


TABLE 10-3: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{\text{MCLR}}$ Pulse Width (low)	2000	—	—	ns	-40° to +85°C
31	Twdt	Watchdog Timer Timeout Period (No Prescaler)	7*	18	33*	ms	VDD = 5.0V, -40° to +85°C
32	Tost	Oscillation Start-up Timer Period	—	1024 TOSC	—	—	TOSC = OSC1 period
33	Tpwrt	Power-up Timer Period	28*	72	132*	ms	VDD = 5.0V, -40° to +85°C
34	TIOZ	I/O hi-impedance from $\overline{\text{MCLR}}$ low		—	2.0*	μs	

* 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.

NOTES:

INDEX

PRODUCT IDENTIFICATION SYSTEM

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

PART NO.	X	/XX	XXX
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)