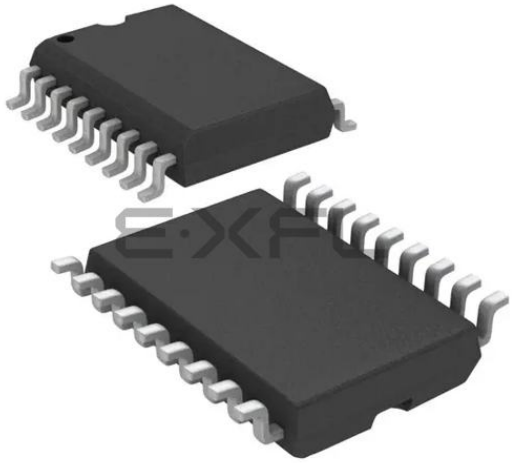


Welcome to E-XFL.COM

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	896B (512 x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	80 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°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/pic16c554-04-so

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.

 © 1996-2013 Microchip Technology Inc.

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16C55X family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16C55X uses a Harvard architecture in which program and data are accessed from separate memories using separate busses. This improves bandwidth over traditional von Neumann architecture where program and data are fetched from the same memory. Separating program and data memory further allows instructions to be sized differently from 8-bit wide data words. Instruction opcodes are 14-bit wide making it possible to have all single word instructions. A 14-bit wide program memory access bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions (35) execute in a single-cycle (200 ns @ 20 MHz) except for program branches. The table below lists the memory (EPROM and RAM).

Device	Program Memory (EPROM)	Data Memor (RAM)
PIC16C554	512	80
PIC16C557	2 K	128
PIC16C558	2 K	128

The PIC16C554 addresses 512 x 14 on-chip program memory. The PIC16C557 and PIC16C558 addresses 2 K x 14 program memory. All program memory is internal.

The PIC16C55X can directly or indirectly address its register files or data memory. All special function registers, including the program counter, are mapped into the data memory. The PIC16C55X has an orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any Addressing mode. This symmetrical nature and lack of ‘special optimal situations’ make programming with the PIC16C55X simple yet efficient. In addition, the learning curve is reduced significantly.

The PIC16C55X devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register). The other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

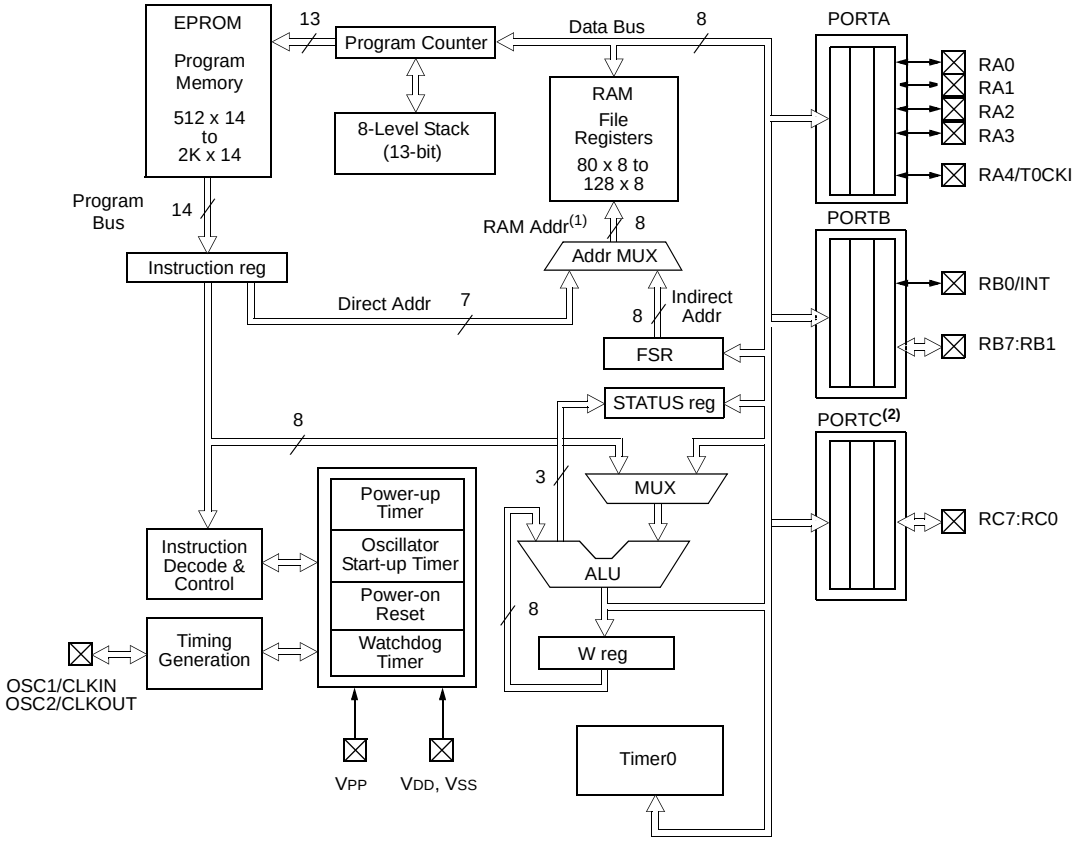
The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{Borrow}}$ and $\overline{\text{Digit}}$ $\overline{\text{Borrow}}$ out bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.

A simplified block diagram is shown in Figure 3-1, with a description of the device pins in Table 3-1.

FIGURE 3-1: BLOCK DIAGRAM

Device	Program Memory	Data Memory
PIC16C554	512 x 14	80 x 8
PIC16C557	2 K x 14	128 x 8
PIC16C558	2 K x 14	128 x 8



Note 1: Higher order bits are from STATUS Register.
2: PIC16C557 only.

 © 1996-2013 Microchip Technology Inc.

3.1 Clocking Scheme/Instruction Cycle

The clock input (OSC1/CLKIN pin) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 3-2.

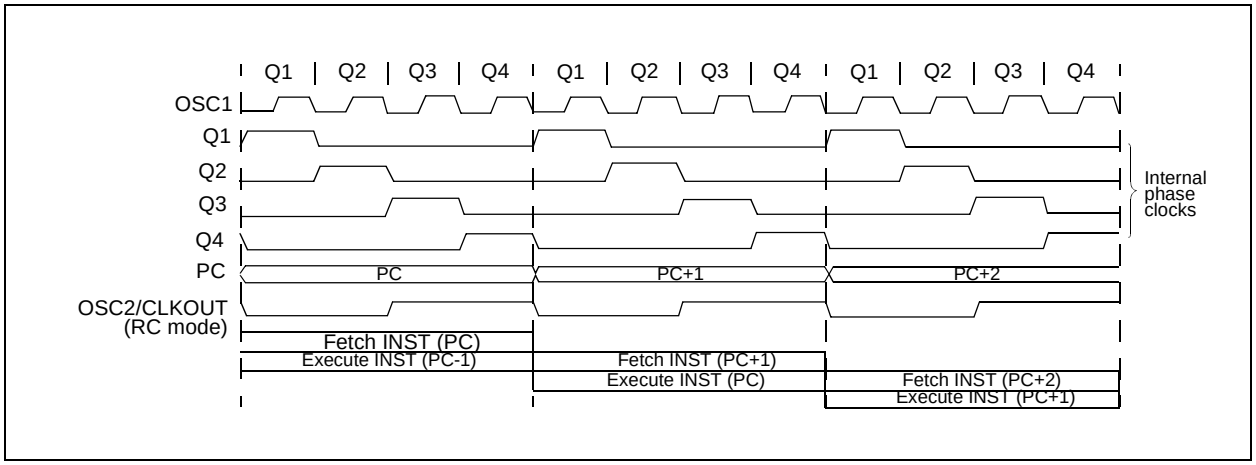
3.2 Instruction Flow/Pipelining

An “Instruction Cycle” consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., GOTO), then two cycles are required to complete the instruction (Example 3-1).

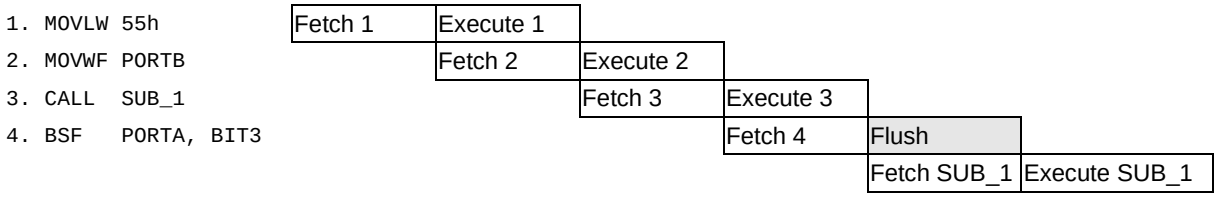
A fetch cycle begins with the program counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the “Instruction Register (IR)” in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW



All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is “flushed” from the pipeline while the new instruction is being fetched and then executed.

 © 1996-2013 Microchip Technology Inc.

4.0 MEMORY ORGANIZATION

4.1 Program Memory Organization

The PIC16C55X has a 13-bit program counter capable of addressing an 8 K x 14 program memory space. Only the first 512 x 14 (0000h - 01FFh) for the PIC16C554 and 2K x 14 (0000h - 07FFh) for the PIC16C557 and PIC16C558 are physically implemented. Accessing a location above these boundaries will cause a wrap-around within the first 512 x 14 spaces in the PIC16C554, or 2K x 14 space of the PIC16C558 and PIC16C557. The RESET vector is at 0000h and the interrupt vector is at 0004h (Figure 4-1, Figure 4-2).

FIGURE 4-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16C554

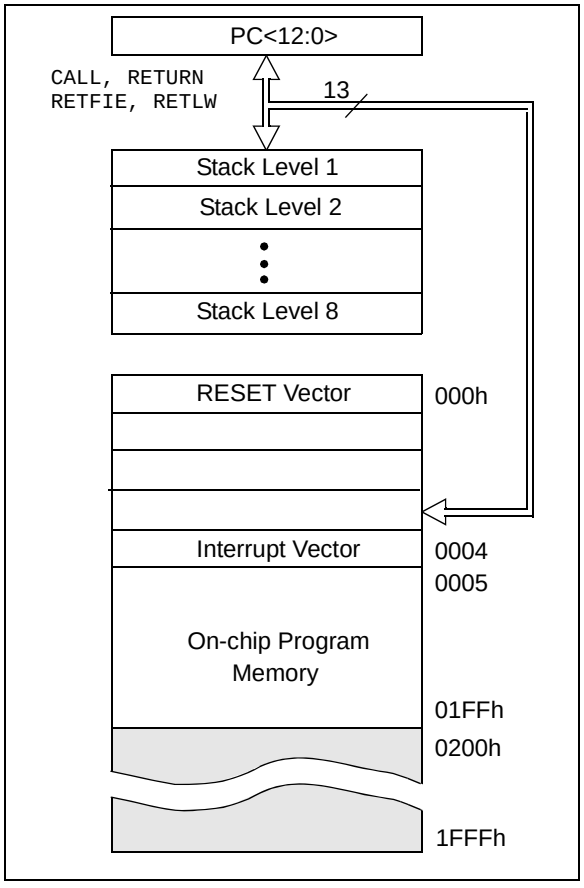
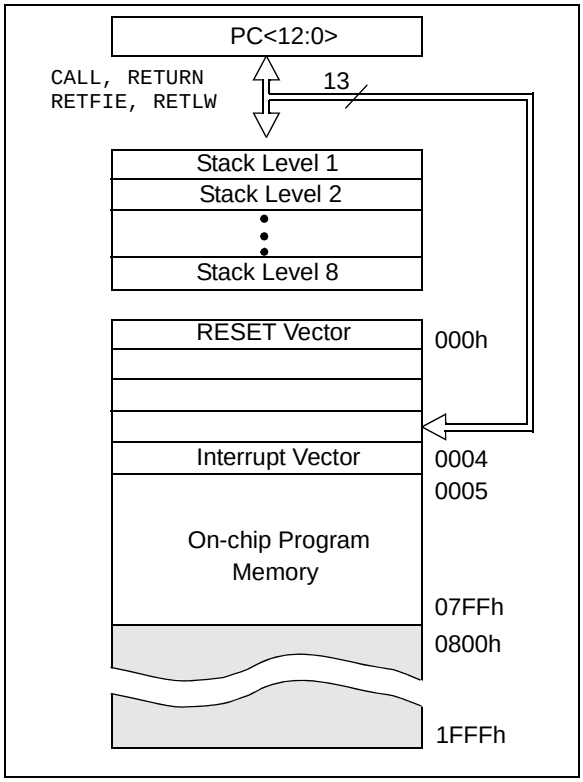


FIGURE 4-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC16C557 AND PIC16C558



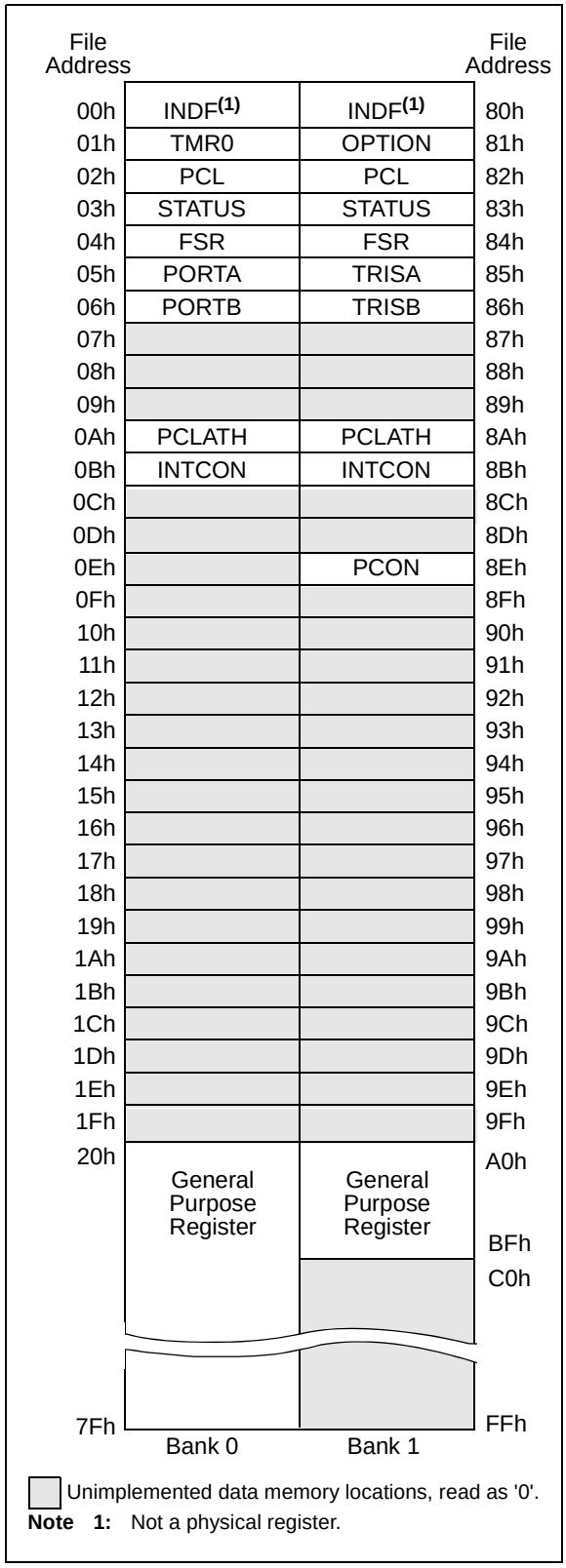
4.2 Data Memory Organization

The data memory (Figure 4-3 through Figure 4-5) is partitioned into two banks which contain the General Purpose Registers (GPR) and the Special Function Registers (SFR). Bank 0 is selected when the RP0 bit (STATUS <5>) is cleared. Bank 1 is selected when the RP0 bit is set. The Special Function Registers are located in the first 32 locations of each Bank. Register locations 20-6Fh (Bank 0) on the PIC16C554 and 20-7Fh (Bank 0) and A0-BFh (Bank 1) on the PIC16C558 and PIC16C557 are General Purpose Registers implemented as static RAM. Some special purpose registers are mapped in Bank 1.

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 80 x 8 in the PIC16C554 and 128 x 8 in the PIC16C557 and PIC16C558. Each can be accessed either directly or indirectly through the File Select Register, FSR (Section 4.4).

FIGURE 4-5: DATA MEMORY MAP FOR THE PIC16C558



4.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (Table 4-1). These registers are static RAM.

The Special Function Registers can be classified into two sets (core and peripheral). The special function registers associated with the “core” functions are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

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.

 © 1996-2013 Microchip Technology Inc.

6.2 Oscillator Configurations

6.2.1 OSCILLATOR TYPES

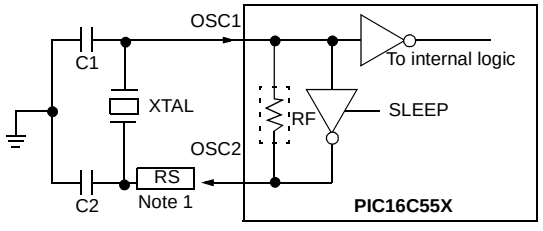
The PIC16C55X can be operated in four different oscillator options. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

6.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, LP or HS modes a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation (Figure 6-1). The PIC16C55X oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1 pin (Figure 6-2).

FIGURE 6-1: CRYSTAL OPERATION (OR CERAMIC RESONATOR) (HS, XT OR LP OSC CONFIGURATION)



Note 1: A series resistor may be required for AT strip cut crystals.
2: See Table 6-1 and Table 6-2 for recommended values of C1 and C2.

FIGURE 6-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

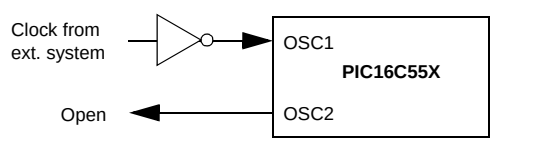


TABLE 6-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS (PRELIMINARY)

Ranges Characterized:			
Mode	**Freq**	**OSC1(C1)**	**OSC2(C2)**
XT	455 kHz 2.0 MHz 4.0 MHz	22 - 100 pF 15 - 68 pF 15 - 68 pF	22 - 100 pF 15 - 68 pF 15 - 68 pF
HS	8.0 MHz 16.0 MHz	10 - 68 pF 10 - 22 pF	10 - 68 pF 10 - 22 pF

Note 1: Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult with the resonator manufacturer for appropriate values of external components.

TABLE 6-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR (PRELIMINARY)

Mode	Freq	OSC1(C1)	OSC2(C2)
LP	32 kHz 200 kHz	68 - 100 pF 15 - 30 pF	68 - 100 pF 15 - 30 pF
XT	100 kHz 2 MHz 4 MHz	68 - 150 pF 15 - 30 pF 15 - 30 pF	150 - 200 pF 15 - 30 pF 15 - 30 pF
HS	8 MHz 10 MHz 20 MHz	15 - 30 pF 15 - 30 pF 15 - 30 pF	15 - 30 pF 15 - 30 pF 15 - 30 pF

Note 1: Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low-drive level specification. Since each crystal has its own characteristics, the user should consult with the crystal manufacturer for appropriate values of external components.

TABLE 6-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	000h	0001 1xxx	---- --0-
$\overline{MCLR}$ Reset during normal operation	000h	000u uuuu	---- --u-
$\overline{MCLR}$ 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	$\overline{MCLR}$ Reset during normal operation $\overline{MCLR}$ 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.

 © 1996-2013 Microchip Technology Inc.

6.5.1 RB0/INT INTERRUPT

An external interrupt on RB0/INT pin is edge triggered: either rising if INTEDG bit (OPTION<6>) is set, or falling if INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, the INTF bit (INTCON<1>) is set. This interrupt can be disabled by clearing the INTE control bit (INTCON<4>). The INTF bit must be cleared in software in the interrupt service routine before re-enabling this interrupt. The RB0/INT interrupt can wake-up the processor from SLEEP, if the INTE bit was set prior to going into SLEEP. The status of the GIE bit decides whether or not the processor branches to the interrupt vector following wake-up. See Section 6.8 for details on SLEEP and Figure 6-14 for timing of wake-up from SLEEP through RB0/INT interrupt.

6.5.2 TMR0 INTERRUPT

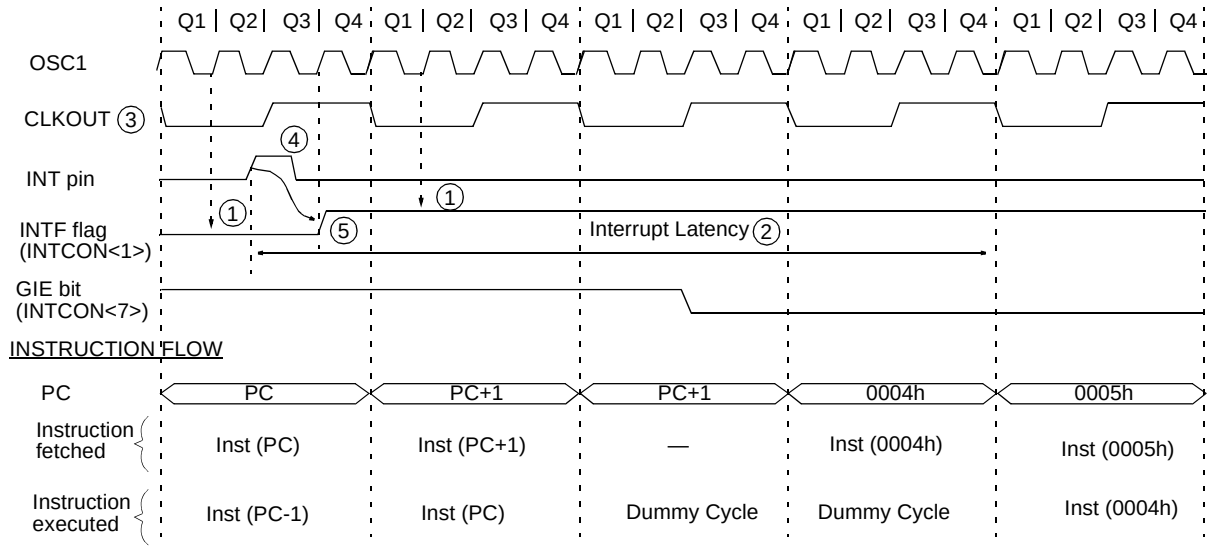
An overflow (FFh $\rightarrow$ 00h) in the TMR0 register will set the T0IF (INTCON<2>) bit. The interrupt can be enabled/disabled by setting/clearing T0IE (INTCON<5>) bit. For operation of the Timer0 module, see Section 7.0.

6.5.3 PORTB INTERRUPT

An input change on PORTB <7:4> sets the RBIF (INTCON<0>) bit. The interrupt can be enabled/disabled by setting/clearing the RBIE (INTCON<4>) bit. For operation of PORTB (Section 5.2).

Note:	If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may get set.

FIGURE 6-12: INT PIN INTERRUPT TIMING



Note 1: INTF flag is sampled here (every Q1).

2: Interrupt latency = 3-4 TCY where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

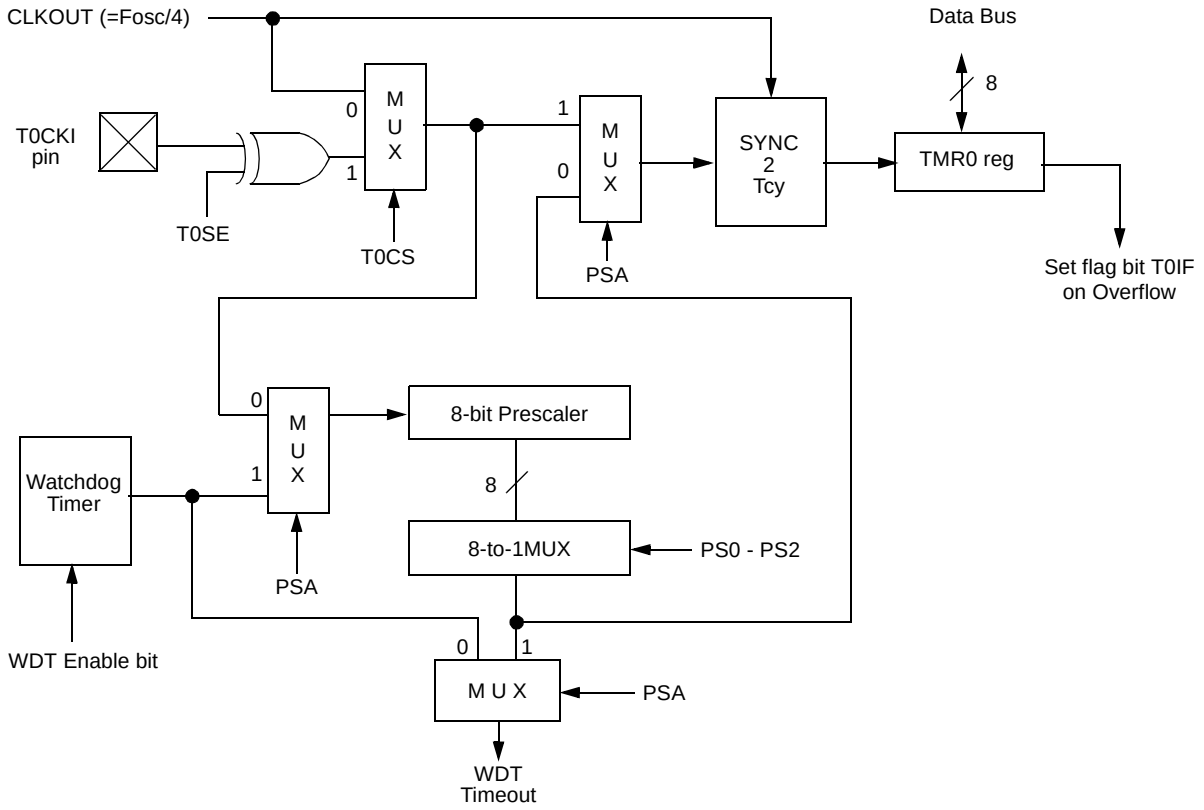
3: CLKOUT is available only in RC Oscillator mode.

4: For minimum width of INT pulse, refer to AC specs.

5: INTF is enabled to be set anytime during the Q4-Q1 cycles.

 © 1996-2013 Microchip Technology Inc.

FIGURE 7-6: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER



Note 1: T0SE, T0CS, PSA, PS0-PS2 are bits in the OPTION register.

 © 1996-2013 Microchip Technology Inc.

7.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed “on the fly” during program execution). To avoid an unintended device RESET, the following instruction sequence (Example 7-1) must be executed when changing the prescaler assignment from Timer0 to WDT. Lines 5-7 are required only if the desired postscaler rate is 1:1 (PS<2:0> = 000) or 1:2 (PS<2:0> = 001).

EXAMPLE 7-1: CHANGING PRESCALER (TIMER0→WDT)

```
BCF    STATUS, RP0 ;Skip if already in
                   ;Bank 0 CLRWDT Clear WDT
CLRF   TMR0        ;Clear TMR0 & Prescaler
BSF    STATUS, RP0 ;Bank 1
MOVLW  '00101111'b ;These 3 lines (5, 6, 7)
MOVWF  OPTION      ;Are required only if
                   ;Desired PS<2:0> are
                   ;CLRWDT 000 or 001
MOVLW  '00101xxx'b ;Set Postscaler to
MOVWF  OPTION      ;Desired WDT rate
BCF    STATUS, RP0 ;Return to Bank 0
```

To change prescaler from the WDT to the TMR0 module use the sequence shown in Example 7-2. This precaution must be taken even if the WDT is disabled.

EXAMPLE 7-2: CHANGING PRESCALER (WDT→TIMER0)

```
CLRWDT                ;Clear WDT and
                      ;prescaler
BSF     STATUS, RP0
MOVLW   b'xxxx0xxx'   ;Select TMR0, new
                      ;prescale value and
                      ;clock source
MOVWF   OPTION
BCF     STATUS, RP0
```

TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER0

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
01h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
0Bh/8Bh	INTCON	GIE	Reserved	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
81h	OPTION	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111

Legend: — = Unimplemented locations, read as ‘0’,
Note 1: Shaded bits are not used by TMR0 module.

BCF **Bit Clear f**

Syntax: [*label*] BCF f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: $0 \rightarrow$ (f<b>)

Status Affected: None

Encoding:

01	00bb	bfff	ffff

Description: Bit 'b' in register 'f' is cleared.

Words: 1

Cycles: 1

Example
```
BCF     FLAG_REG, 7
```
Before Instruction
FLAG_REG = 0xC7
After Instruction
FLAG_REG = 0x47

BSF **Bit Set f**

Syntax: [*label*] BSF f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: $1 \rightarrow$ (f<b>)

Status Affected: None

Encoding:

01	01bb	bfff	ffff

Description: Bit 'b' in register 'f' is set.

Words: 1

Cycles: 1

Example
```
BSF     FLAG_REG,   7
```
Before Instruction
FLAG_REG = 0x0A
After Instruction
FLAG_REG = 0x8A

BTFSC **Bit Test, Skip if Clear**

Syntax: [*label*] BTFSC f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

01	10bb	bfff	ffff

Description: If bit 'b' in register 'f' is '0' then the next instruction is skipped. If bit 'b' is '0' then the next instruction fetched during the current instruction execution is discarded, and a `NOP` is executed instead, making this a two-cycle instruction.

Words: 1

Cycles: 1(2)

Example
```
HERE    BTFSC   FLAG,1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```
Before Instruction
PC = address `HERE`
After Instruction
if FLAG<1> = 0,
PC = address `TRUE`
if FLAG<1> = 1,
PC = address `FALSE`

 © 1996-2013 Microchip Technology Inc.

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

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

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

RLF **Rotate Left f through Carry**

Syntax: [*label*] RLF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

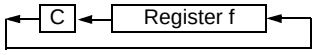
Operation: See description below

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

 © 1996-2013 Microchip Technology Inc.

RRF **Rotate Right f through Carry**

Syntax: [*label*] RRF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

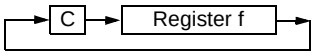
Operation: See description below

Status Affected: C

Encoding:

00	1100	dfff	ffff

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.



Words: 1

Cycles: 1

Example
```
RRF             REG1,0
```
Before Instruction
```
REG1 = 1110 0110
C    = 0
```
After Instruction
```
REG1 = 1110 0110
W    = 0111 0011
C    = 0
```

SLEEP

Syntax: [*label* SLEEP]

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

00	0000	0110	0011

Description: The power-down status bit, $\overline{PD}$ is cleared. Timeout status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared.
The processor is put into SLEEP mode with the oscillator stopped. See Section 6.8 for more details.

Words: 1

Cycles: 1

Example:
```
SLEEP
```

SUBLW **Subtract W from Literal**

Syntax: [*label*] SUBLW k

Operands: $0 \le k \le 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Encoding:

11	110x	kkkk	kkkk

Description: The W register is subtracted (2's complement method) from the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example 1:
```
SUBLW   0x02
```
Before Instruction
```
W = 1
C = ?
```
After Instruction
```
W = 1
C = 1; result is positive
```

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

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

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:
OSC1 = external square wave, from rail to rail; all I/O pins configured as input, pulled to VDD,
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 ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial and -40°C ≤ TA ≤ +125°C for automotive Operating voltage VDD range as described in DC spec Table 10-1				
Param. No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Unit**	**Conditions**
	VIL	**Input Low Voltage**					
		I/O ports					
D030		with TTL buffer	VSS	—	0.8V 0.15 VDD	V	VDD = 4.5V to 5.5V otherwise
D031		with Schmitt Trigger input	VSS		0.2 VDD	V	
D032		$\overline{MCLR}$, RA4/T0CKI,OSC1 (in RC mode)	VSS	—	0.2 VDD	V	**(Note1)**
D033		OSC1 (in XT* and HS)	VSS	—	0.3 VDD	V	
		OSC1 (in LP*)	VSS	—	0.6 VDD-1.0	V	
	VIH	**Input High Voltage**					
		I/O ports		—			
D040		with TTL buffer	2.0V 0.8 + 0.25 VDD	— —	VDD VDD	V V	VDD = 4.5V to 5.5V otherwise
D041		with Schmitt Trigger input	0.8V		VDD		
D042		$\overline{MCLR}$ RA4/T0CKI	0.8 VDD	—	VDD	V	
D043 D043A		OSC1 (XT*, HS and LP*) OSC1 (in RC mode)	0.7 VDD 0.9 VDD	—	VDD	V	 **(Note1)**
D070	IPURB	PORTB weak pull-up current	50	200	400	µA	VDD = 5.0V, VPIN = VSS
	IIL	**Input Leakage Current(2)(3)**					
		I/O ports (Except PORTA)			±1.0	µA	VSS ≤ VPIN ≤ VDD, pin at hi-impedance
D060		PORTA	—	—	±0.5	µA	Vss ≤ VPIN ≤ VDD, pin at hi-impedance
D061		RA4/T0CKI	—	—	±1.0	µA	Vss ≤ VPIN ≤ VDD
D063		OSC1, $\overline{MCLR}$	—	—	±5.0	µA	Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
	VOL	**Output Low Voltage**					
D080		I/O ports	—	—	0.6	V	IOL=8.5 mA, VDD=4.5V, -40° to +85°C
			—	—	0.6	V	IOL=7.0 mA, VDD=4.5V, +125°C
D083		OSC2/CLKOUT	—	—	0.6	V	IOL=1.6 mA, VDD=4.5V, -40° to +85°C
		(RC only)	—	—	0.6	V	IOL=1.2 mA, VDD=4.5V, +125°C
	VOH	**Output High Voltage(3)**					
D090		I/O ports (Except RA4)	VDD-0.7	—	—	V	IOH=-3.0 mA, VDD=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 $\overline{MCLR}$ 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.

 © 1996-2013 Microchip Technology Inc.

10.4 Timing Diagrams and Specifications

FIGURE 10-6: EXTERNAL CLOCK TIMING

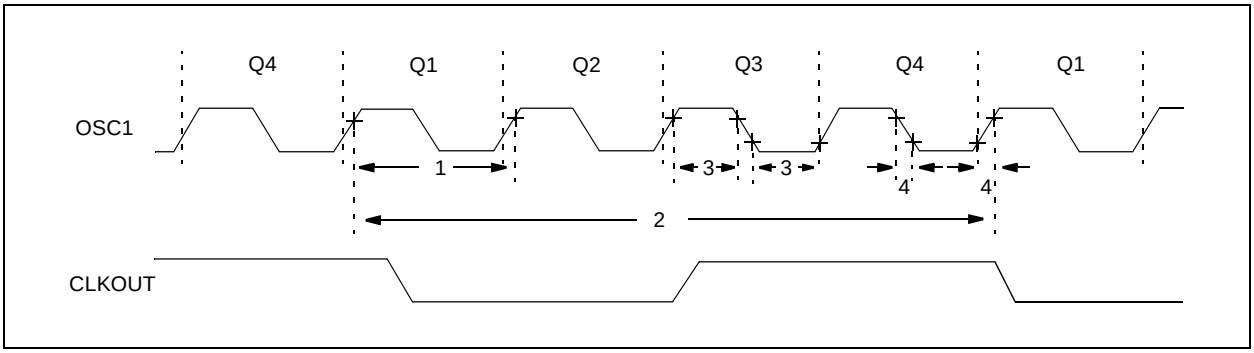


TABLE 10-1: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fos	External CLKIN Frequency(1)	DC	—	4	MHz	XT and RC osc mode, VDD=5.0V
			DC	—	20	MHz	HS osc mode
			DC	—	200	kHz	LP osc mode
		Oscillator Frequency(1)	DC	—	4	MHz	RC osc mode, VDD=5.0V
			0.1	—	4	MHz	XT osc mode
			1	—	20	MHz	HS osc mode
			DC	–	200	kHz	LP osc mode
1	Tosc	External CLKIN Period(1)	250	—	—	ns	XT and RC osc mode
			50	—	—	ns	HS osc mode
			5	—	—	µs	LP osc mode
		Oscillator Period(1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			50	—	1,000	ns	HS osc mode
			5	—	—	µs	LP osc mode
2	Tcy	Instruction Cycle Time(1)	1.0	Fos/4	DC	µs	TCY=FOS/4
3*	TosL, TosH	External Clock in (OSC1) High or Low Time	100*	—	—	ns	XT osc mode
			2*	—	—	µs	LP osc mode
			20*	—	—	ns	HS osc mode
4*	TosR, TosF	External Clock in (OSC1) Rise or Fall Time	25*	—	—	ns	XT osc mode
			50*	—	—	ns	LP osc mode
			15*	—	—	ns	HS osc mode

* These parameters are characterized but not tested.

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

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1 pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

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.

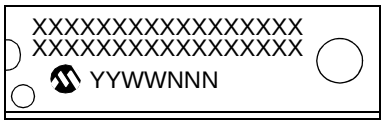
† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

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

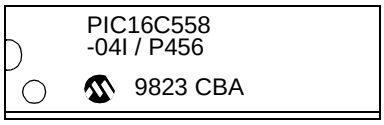
11.0 PACKAGING INFORMATION

11.1 Package Marking Information

18-Lead PDIP

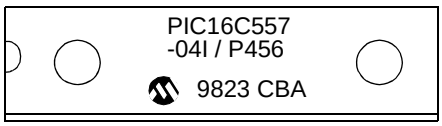


Example

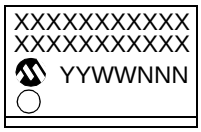


28-Lead PDIP

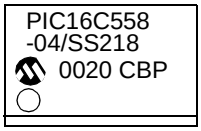
Example



20-Lead SSOP



Example



28-Lead SSOP

Example

Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.