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	20MHz
Connectivity	-
Peripherals	POR, WDT
Number of I/O	20
Program Memory Size	3KB (2K x 12)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	72 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.600", 15.24mm)
Supplier Device Package	28-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f57-i-p

TABLE 2-3: PIC16F59 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0	RA0	TTL	CMOS	Bidirectional I/O pin
RA1	RA1	TTL	CMOS	Bidirectional I/O pin
RA2	RA2	TTL	CMOS	Bidirectional I/O pin
RA3	RA3	TTL	CMOS	Bidirectional I/O pin
RB0	RB0	TTL	CMOS	Bidirectional I/O pin
RB1	RB1	TTL	CMOS	Bidirectional I/O pin
RB2	RB2	TTL	CMOS	Bidirectional I/O pin
RB3	RB3	TTL	CMOS	Bidirectional I/O pin
RB4	RB4	TTL	CMOS	Bidirectional I/O pin
RB5	RB5	TTL	CMOS	Bidirectional I/O pin
RB6/ICSPCLK	RB6	TTL	CMOS	Bidirectional I/O pin
	ICSPCLK	ST	—	Serial programming clock
RB7/ICSPDAT	RB7	TTL	CMOS	Bidirectional I/O pin
	ICSPDAT	ST	CMOS	Serial programming I/O
RC0	RC0	TTL	CMOS	Bidirectional I/O pin
RC1	RC1	TTL	CMOS	Bidirectional I/O pin
RC2	RC2	TTL	CMOS	Bidirectional I/O pin
RC3	RC3	TTL	CMOS	Bidirectional I/O pin
RC4	RC4	TTL	CMOS	Bidirectional I/O pin
RC5	RC5	TTL	CMOS	Bidirectional I/O pin
RC6	RC6	TTL	CMOS	Bidirectional I/O pin
RC7	RC7	TTL	CMOS	Bidirectional I/O pin
RD0	RD0	TTL	CMOS	Bidirectional I/O pin
RD1	RD1	TTL	CMOS	Bidirectional I/O pin
RD2	RD2	TTL	CMOS	Bidirectional I/O pin
RD3	RD3	TTL	CMOS	Bidirectional I/O pin
RD4	RD4	TTL	CMOS	Bidirectional I/O pin
RD5	RD5	TTL	CMOS	Bidirectional I/O pin
RD6	RD6	TTL	CMOS	Bidirectional I/O pin
RD7	RD7	TTL	CMOS	Bidirectional I/O pin
RE4	RE4	TTL	CMOS	Bidirectional I/O pin
RE5	RE5	TTL	CMOS	Bidirectional I/O pin
RE6	RE6	TTL	CMOS	Bidirectional I/O pin
RE7	RE7	TTL	CMOS	Bidirectional I/O pin
T0CKI	T0CKI	ST	—	Clock input to Timer0. Must be tied to VSS or VDD, if not in use, to reduce current consumption.
$\overline{\text{MCLR}}$/VPP	$\overline{\text{MCLR}}$	ST	—	Active-low Reset to device. Voltage on the $\overline{\text{MCLR}}$/VPP pin must not exceed VDD to avoid unintended entering of Programming mode.
	VPP	HV	—	Programming voltage input
OSC1/CLKIN	OSC1	XTAL	—	Oscillator crystal input
	CLKIN	ST	—	External clock source input
OSC2/CLKOUT	OSC2	—	XTAL	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
	CLKOUT	—	CMOS	In RC mode, OSC2 pin outputs CLKOUT, which has 1/4 the frequency of OSC1.
VDD	VDD	Power	—	Positive supply for logic and I/O pins
VSS	VSS	Power	—	Ground reference for logic and I/O pins

Legend: I = input, O = output, ST = Schmitt Trigger input, I/O = input/output, — = Not Used, TTL = TTL input, CMOS = CMOS output, XTAL = Crystal input/output, HV = High Voltage

3.3 STATUS Register

This register contains the arithmetic status of the ALU, the Reset status and the page preselect bits for program memories larger than 512 words.

The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as `000u u1uu` (where `u` = unchanged).

Therefore, it is recommended that only `BCF`, `BSF`, `MOVWF` and `SWAPF` instructions be used to alter the STATUS register because these instructions do not affect the Z, DC or C bits from the STATUS register. For other instructions which do affect Status bits, see **Section 9.0 "Instruction Set Summary"**.

REGISTER 3-1: STATUS REGISTER (ADDRESS: 03h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
PA2	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

bit 7 **PA2**: Reserved, do not use
Use of the PA2 bit as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.

bit 6-5 **PA<1:0>**: Program Page Preselect bits (PIC16F57/PIC16F59)
`00` = Page 0 (000h-1FFh)
`01` = Page 1 (200h-3FFh)
`10` = Page 2 (400h-5FFh)
`11` = Page 3 (600h-7FFh)
Each page is 512 words. Using the PA<1:0> bits as general purpose read/write bits in devices which do not use them for program page preselect is not recommended. This may affect upward compatibility with future products.

bit 4 **$\overline{TO}$**: Time-Out bit
`1` = After power-up, `CLRWDT` instruction or `SLEEP` instruction
`0` = A WDT time-out occurred

bit 3 **$\overline{PD}$**: Power-Down bit
`1` = After power-up or by the `CLRWDT` instruction
`0` = By execution of the `SLEEP` instruction

bit 2 **Z**: Zero bit
`1` = The result of an arithmetic or logic operation is zero
`0` = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit Carry/$\overline{Borrow}$ bit (for `ADDWF` and `SUBWF` instructions)
ADDWF
`1` = A carry to the 4th low order bit of the result occurred
`0` = A carry from the 4th low order bit of the result did not occur
SUBWF
`1` = A borrow to the 4th low order bit of the result did not occur
`0` = A borrow from the 4th low order bit of the result occurred

bit 0 **C**: Carry/$\overline{Borrow}$ bit (for `ADDWF`, `SUBWF` and `RRF`, `RLF` instructions)

ADDWF	SUBWF	RRF or RLF
`1` = A carry occurred	`1` = A borrow did not occur	Loaded with LSb or MSb, respectively
`0` = A carry did not occur	`0` = A borrow occurred	

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

3.5 Program Counter

As a program instruction is executed, the Program Counter (PC) will contain the address of the next program instruction to be executed. The PC value is increased by one, every instruction cycle, unless an instruction changes the PC.

For a GOTO instruction, bits 8:0 of the PC are provided by the GOTO instruction word. The PC Latch (PCL) is mapped to PC<7:0> (Figure 3-6 and Figure 3-7).

For the PIC16F57 and PIC16F59, a page number must be supplied as well. Bit 5 and bit 6 of the STATUS register provide page information to bit 9 and bit 10 of the PC (Figure 3-6 and Figure 3-7).

For a CALL instruction, or any instruction where the PCL is the destination, bits 7:0 of the PC again are provided by the instruction word. However, PC<8> does not come from the instruction word, but is always cleared (Figure 3-6 and Figure 3-7).

Instructions where the PCL is the destination or modify PCL instructions, include MOVWF PCL, ADDWF PCL, and BSF PCL,5.

For the PIC16F57 and PIC16F59, a page number again must be supplied. Bit 5 and bit 6 of the STATUS register provide page information to bit 9 and bit 10 of the PC (Figure 3-6 and Figure 3-7).

Note:	Because PC<8> is cleared in the CALL instruction or any modified PCL instruction, all subroutine calls or computed jumps are limited to the first 256 locations of any program memory page (512 words long).

FIGURE 3-6: LOADING OF PC BRANCH INSTRUCTIONS – PIC16F54

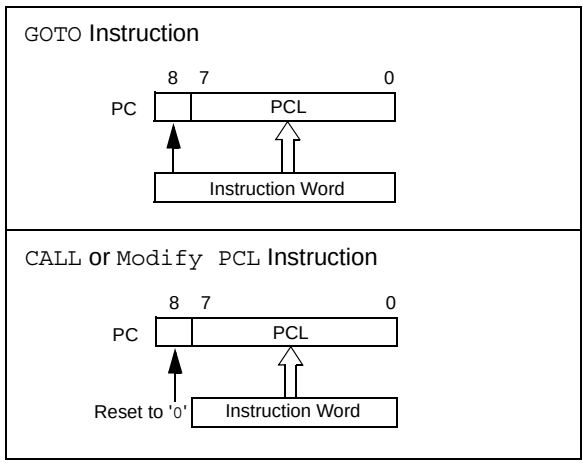
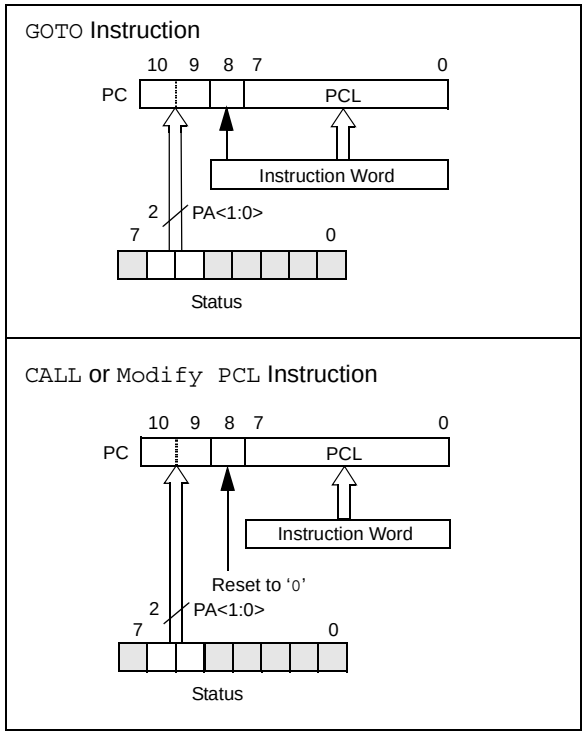


FIGURE 3-7: LOADING OF PC BRANCH INSTRUCTIONS – PIC16F57 AND PIC16F59



3.5.1 PAGING CONSIDERATIONS PIC16F57 AND PIC16F59

If the PC is pointing to the last address of a selected memory page, when it increments, it will cause the program to continue in the next higher page. However, the page preselect bits in the STATUS register will not be updated. Therefore, the next GOTO, CALL or MODIFY PCL instruction will send the program to the page specified by the page preselect bits (PA0 or PA<1:0>).

For example, a NOP at location 1FFh (page 0) increments the PC to 200h (page 1). A GOTO xxx at 200h will return the program to address xxh on page 0 (assuming that PA<1:0> are clear).

To prevent this, the page preselect bits must be updated under program control.

3.5.2 EFFECTS OF RESET

The PC is set upon a Reset, which means that the PC addresses the last location in the last page (i.e., the Reset vector).

The STATUS register page preselect bits are cleared upon a Reset, which means that page 0 is preselected.

Therefore, upon a Reset, a GOTO instruction at the Reset vector location will automatically cause the program to jump to page 0.

3.6 Stack

The PIC16F54 device has a 9-bit wide, two-level hardware PUSH/POP stack. The PIC16F57 and PIC16F59 devices have an 11-bit wide, two-level hardware PUSH/POP stack.

A CALL instruction will PUSH the current value of stack 1 into stack 2 and then PUSH the current program counter value, incremented by one, into stack level 1. If more than two sequential CALL's are executed, only the most recent two return addresses are stored.

A RETLW instruction will POP the contents of stack level 1 into the program counter and then copy stack level 2 contents into level 1. If more than two sequential RETLW's are executed, the stack will be filled with the address previously stored in level 2.

Note:	The W register will be loaded with the literal value specified in the instruction. This is particularly useful for the implementation of data look-up tables within the program memory.

For the RETLW instruction, the PC is loaded with the Top-of-Stack (TOS) contents. All of the devices covered in this data sheet have a two-level stack. The stack has the same bit width as the device PC, therefore, paging is not an issue when returning from a subroutine.

3.7 Indirect Data Addressing; INDF and FSR Registers

The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR Register (FSR is a *pointer*). This is indirect addressing.

EXAMPLE 3-1: INDIRECT ADDRESSING

- Register file 08 contains the value 10h
- Register file 09 contains the value 0Ah
- Load the value 08 into the FSR register
- A read of the INDF register will return the value of 10h
- Increment the value of the FSR register by one (FSR = 09h)
- A read of the INDF register now will return the value of 0Ah.

Reading INDF itself indirectly (FSR = 0) will produce 00h. Writing to the INDF register indirectly results in a no-operation (although Status bits may be affected).

A simple program to clear RAM locations 10h-1Fh using indirect addressing is shown in Example 3-2.

EXAMPLE 3-2: HOW TO CLEAR RAM USING INDIRECT ADDRESSING

```
          MOVLW  H'10'  ;initialize pointer
          MOVWF  FSR    ;to RAM
NEXT      CLRF   INDF   ;clear INDF Register
          INCF   FSR,F  ;inc pointer
          BTFSC  FSR,4  ;all done?
          GOTO   NEXT   ;NO, clear next
CONTINUE
          :             ;YES, continue
```

The FSR is either a 5-bit (PIC16F54), 7-bit (PIC16F57) or 8-bit (PIC16F59) wide register. It is used in conjunction with the INDF register to indirectly address the data memory area.

The FSR<4:0> bits are used to select data memory addresses 00h to 1Fh.

PIC16F54: This does not use banking. FSR<7:5> bits are unimplemented and read as '1's.

PIC16F57: FSR<7> bit is unimplemented and read as '1'. FSR<6:5> are the bank select bits and are used to select the bank to be addressed (00 = Bank 0, 01 = Bank 1, 10 = Bank 2, 11 = Bank 3).

PIC16F59: FSR<7:5> are the bank select bits and are used to select the bank to be addressed (000 = Bank 0, 001 = Bank 1, 010 = Bank 2, 011 = Bank 3, 100 = Bank 4, 101 = Bank 5, 110 = Bank 6, 111 = Bank 7).

Note:	A CLRF FSR instruction may not result in an FSR value of 00h if there are unimplemented bits present in the FSR.

5.0 RESET

The PIC16F5X devices may be reset in one of the following ways:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ Reset (normal operation)
- $\overline{\text{MCLR}}$ Wake-up Reset (from Sleep)
- WDT Reset (normal operation)
- WDT Wake-up Reset (from Sleep)

Table 5-1 shows these Reset conditions for the PCL and STATUS registers.

Some registers are not affected in any Reset condition. Their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a "Reset state" on Power-on Reset (POR), $\overline{\text{MCLR}}$ or WDT Reset. A $\overline{\text{MCLR}}$ or WDT wake-up from Sleep also results in a device Reset and not a continuation of operation before Sleep.

The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits (STATUS <4:3>) are set or cleared depending on the different Reset conditions (Table 5-1). These bits may be used to determine the nature of the Reset.

Table 5-3 lists a full description of Reset states of all registers. Figure 5-1 shows a simplified block diagram of the on-chip Reset circuit.

TABLE 5-1: STATUS BITS AND THEIR SIGNIFICANCE

Condition	$\overline{\text{TO}}$	$\overline{\text{PD}}$
Power-on Reset	1	1
$\overline{\text{MCLR}}$ Reset (normal operation)	u	u
$\overline{\text{MCLR}}$ Wake-up (from Sleep)	1	0
WDT Reset (normal operation)	0	1
WDT Wake-up (from Sleep)	0	0

Legend: u = unchanged, x = unknown, — = unimplemented read as '0'.

TABLE 5-2: SUMMARY OF REGISTERS ASSOCIATED WITH RESET

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on $\overline{\text{MCLR}}$ and WDT Reset
03h	STATUS	PA2	PA1	PA0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	000q quuu

Legend: u = unchanged, x = unknown, q = see Table 5-1 for possible values.

7.1 Using Timer0 with an External Clock

When an external clock input is used for Timer0, it must meet certain requirements. The external clock requirement is due to internal phase clock (TOSC) synchronization. Also, there is a delay in the actual incrementing of Timer0 after synchronization.

7.1.1 EXTERNAL CLOCK SYNCHRONIZATION

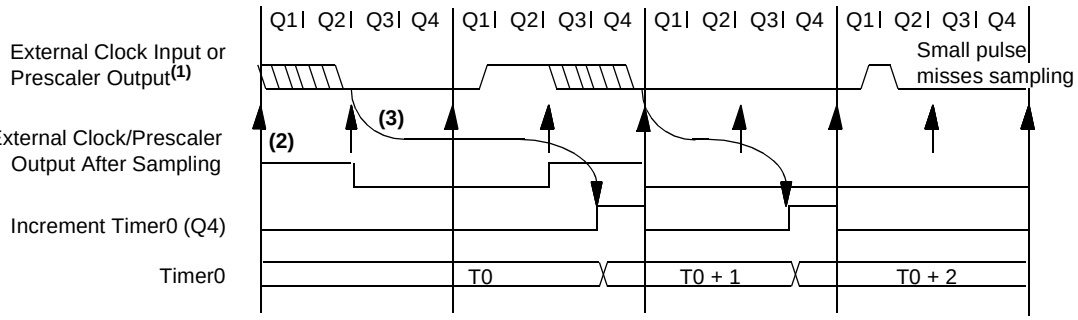
When no prescaler is used, the external clock is the Timer0 input. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 7-4). Therefore, it is necessary for T0CKI to be high for at least 2TOSC (and a small RC delay of 20 ns) and low for at least 2TOSC (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by the asynchronous ripple counter-type prescaler so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least 4TOSC (and a small RC delay of 40 ns) divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of 10 ns. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device.

7.1.2 TIMER0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the Timer0 module is actually incremented. Figure 7-4 shows the delay from the external clock edge to the timer incrementing.

FIGURE 7-4: TIMER0 TIMING WITH EXTERNAL CLOCK



Note 1: External clock if no prescaler selected; prescaler output otherwise.
2: The arrows indicate the points in time where sampling occurs.
3: Delay from clock input change to Timer0 increment is 3TOSC to 7TOSC (duration of Q = TOSC). Therefore, the error in measuring the interval between two edges on Timer0 input = ± 4TOSC max.

7.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer (WDT), respectively (**Section 8.2.1 "WDT Period"**). For simplicity, this counter is being referred to as "prescaler" throughout this data sheet. Note that the prescaler may be used by either the Timer0 module or the WDT, but not both. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the WDT, and vice-versa.

The PSA and PS<2:0> bits (OPTION<3:0>) determine prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF 1`, `MOVWF 1`, `BSF 1,x`, etc.) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the WDT. The prescaler is neither readable nor writable. On a Reset, the prescaler contains all '0's.

TABLE 9-2: INSTRUCTION SET SUMMARY

Mnemonic, Operands		Description	Cycles	12-Bit Opcode			Status Affected	Notes
				MSb		LSb		
ADDWF	f, d	Add W and f	1	0001	11df	ffff	C,DC,Z	1, 2, 4
ANDWF	f, d	AND W with f	1	0001	01df	ffff	Z	2, 4
CLRF	f	Clear f	1	0000	011f	ffff	Z	4
CLRW	—	Clear W	1	0000	0100	0000	Z	
COMF	f, d	Complement f	1	0010	01df	ffff	Z	
DECF	f, d	Decrement f	1	0000	11df	ffff	Z	2, 4
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	0010	11df	ffff	None	2, 4
INCF	f, d	Increment f	1	0010	10df	ffff	Z	2, 4
INCFSZ	f, d	Increment f, Skip if 0	1(2)	0011	11df	ffff	None	2, 4
IORWF	f, d	Inclusive OR W with f	1	0001	00df	ffff	Z	2, 4
MOVF	f, d	Move f	1	0010	00df	ffff	Z	2, 4
MOVWF	f	Move W to f	1	0000	001f	ffff	None	1, 4
NOP	—	No Operation	1	0000	0000	0000	None	
RLF	f, d	Rotate left f through Carry	1	0011	01df	ffff	C	2, 4
RRF	f, d	Rotate right f through Carry	1	0011	00df	ffff	C	2,4
SUBWF	f, d	Subtract W from f	1	0000	10df	ffff	C,DC,Z	1, 2, 4
SWAPF	f, d	Swap f	1	0011	10df	ffff	None	2, 4
XORWF	f, d	Exclusive OR W with f	1	0001	10df	ffff	Z	2, 4
BIT-ORIENTED FILE REGISTER OPERATIONS								
BCF	f, b	Bit Clear f	1	0100	bbbf	ffff	None	2, 4
BSF	f, b	Bit Set f	1	0101	bbbf	ffff	None	2, 4
BTFSC	f, b	Bit Test f, Skip if Clear	1(2)	0110	bbbf	ffff	None	
BTFSS	f, b	Bit Test f, Skip if Set	1(2)	0111	bbbf	ffff	None	
LITERAL AND CONTROL OPERATIONS								
ANDLW	k	AND literal with W	1	1110	kkkk	kkkk	Z	
CALL	k	Subroutine Call	2	1001	kkkk	kkkk	None	1
CLRWDT	—	Clear Watchdog Timer	1	0000	0000	0100	$\overline{TO}$, $\overline{PD}$	
GOTO	k	Unconditional branch	2	101k	kkkk	kkkk	None	
IORLW	k	Inclusive OR Literal with W	1	1101	kkkk	kkkk	Z	
MOVLW	k	Move Literal to W	1	1100	kkkk	kkkk	None	
OPTION	—	Load OPTION register	1	0000	0000	0010	None	
RETLW	k	Return, place Literal in W	2	1000	kkkk	kkkk	None	
SLEEP	—	Go into Standby mode	1	0000	0000	0011	$\overline{TO}$, $\overline{PD}$	
TRIS	f	Load TRIS register	1	0000	0000	0fff	None	3
XORLW	k	Exclusive OR Literal to W	1	1111	kkkk	kkkk	Z	

Note 1: The 9th bit of the program counter will be forced to a '0' by any instruction that writes to the PC except for GOTO (see **Section 3.5 "Program Counter"** for more on program counter).

2: 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'.

3: The instruction TRIS f, where f = 5, 6 or 7 causes the contents of the W register to be written to the tri-state latches of PORTA, B or C, respectively. A '1' forces the pin to a high-impedance state and disables the output buffers.

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

COMF **Complement f**

Syntax: [*label*] COMF f, d

Operands: $0 \le f \le 31$
$d \in [0,1]$

Operation: $(\bar{f}) \rightarrow (dest)$

Status Affected: Z

Encoding:

0010	01df	ffff

Description: The contents of register 'f' are complemented. 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:
```
COMF   REG1,0
```
Before Instruction
REG1 = 0x13
After Instruction
REG1 = 0x13
W = 0xEC

DECF **Decrement f**

Syntax: [*label*] DECF f, d

Operands: $0 \le f \le 31$
$d \in [0,1]$

Operation: (f) – 1 → (dest)

Status Affected: Z

Encoding:

0000	11df	ffff

Description: Decrement 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:
```
DECF    CNT, 1
```
Before Instruction
CNT = 0x01
Z = 0
After Instruction
CNT = 0x00
Z = 1

DECFSZ **Decrement f, Skip if 0**

Syntax: [*label*] DECFSZ f, d

Operands: $0 \le f \le 31$
$d \in [0,1]$

Operation: (f) – 1 → d; skip if result = 0

Status Affected: None

Encoding:

0010	11df	ffff

Description: The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1'. the result is placed back in register 'f'. If the result is '0', the next instruction, which is already fetched, is discarded and a NOP is executed instead making it a two-cycle instruction.

Words: 1

Cycles: 1(2)

Example:
```
HERE     DECFSZ   CNT, 1
         GOTO     LOOP
CONTINUE •
         •
         •
```
Before Instruction
PC = address(HERE)
After Instruction
CNT = CNT - 1;
if CNT = 0,
PC = address (CONTINUE);
if CNT ≠ 0,
PC = address (HERE+1)

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f

Operands: $0 \le f \le 31$

Operation: (W) → (f)

Status Affected: None

Encoding:

0000	001f	ffff

Description: Move data from the W register to register 'f'.

Words: 1

Cycles: 1

Example:

```
MOVWF  TEMP_REG
```

Before Instruction
TEMP_REG = 0xFF
W = 0x4F
After Instruction
TEMP_REG = 0x4F
W = 0x4F

NOP **No Operation**

Syntax: [*label*] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

0000	0000	0000

Description: No operation.

Words: 1

Cycles: 1

Example:

```
NOP
```

OPTION **Load OPTION Register**

Syntax: [*label*] OPTION

Operands: None

Operation: (W) → OPTION

Status Affected: None

Encoding:

0000	0000	0010

Description: The content of the W register is loaded into the Option register.

Words: 1

Cycles: 1

Example:

```
OPTION
```

Before Instruction
W = 0x07
After Instruction
OPTION = 0x07

RETLW **Return with Literal in W**

Syntax: [*label*] RETLW k

Operands: $0 \le k \le 255$

Operation: k → (W);
TOS → PC

Status Affected: None

Encoding:

1000	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.
TABLE •
      ADDWF PC  ;W = offset
      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: $0 \le f \le 31$
$d \in [0,1]$

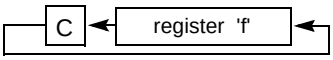
Operation: See description below

Status Affected: C

Encoding:

0011	01df	ffff

Description: The contents of register ‘f’ are rotated one bit to the left through the Carry Flag (STATUS<0>). 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

RRF **Rotate Right f through Carry**

Syntax: [*label*] RRF f, d

Operands: $0 \le f \le 31$
$d \in [0,1]$

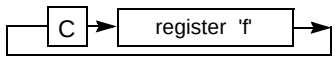
Operation: See description below

Status Affected: C

Encoding:

0011	00df	ffff

Description: The contents of register ‘f’ are rotated one bit to the right through the Carry Flag (STATUS<0>). 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 **Go into Standby Mode**

Syntax: [*label*] Sleep

Operands: None

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

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

Encoding:

0000	0000	0011

Description: Time-out Status bit ($\overline{TO}$) is set. The power-down Status bit ($\overline{PD}$) is cleared. The WDT and its prescaler are cleared.
The processor is put into Sleep mode with the oscillator stopped. See section on Sleep for more details.

Words: 1

Cycles: 1

Example: `SLEEP`

10.2 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for all PIC MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

10.3 MPLAB C18 and MPLAB C30 C Compilers

The MPLAB C18 and MPLAB C30 Code Development Systems are complete ANSI C compilers for Microchip's PIC18 and PIC24 families of microcontrollers and the dsPIC30 and dsPIC33 family of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use not found with other compilers.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

10.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

10.5 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 Assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire dsPIC30F instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

10.6 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C18 and MPLAB C30 C Compilers, and the MPASM and MPLAB ASM30 Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

11.2 DC Characteristics: PIC16F5X (Extended)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Sym.**	**Characteristic/Device**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
D001	VDD	**Supply Voltage**	2.0	—	5.5	V	
D002	VDR	**RAM Data Retention Voltage(1)**	—	1.5*	—	V	Device in Sleep mode
D003	VPOR	**VDD Start Voltage** to ensure Power-on Reset	—	VSS	—	V	See **Section 5.1 "Power-on Reset (POR)"** for details on Power-on Reset
D004	SVDD	**VDD Rise Rate** to ensure Power-on Reset	0.05*	—	—	V/ms	See **Section 5.1 "Power-on Reset (POR)"** for details on Power-on Reset
D010	IDD	**Supply Current(2)**					
			— — — —	170 0.4 1.7 15	450 2.0 7.0 40	µA mA mA µA	FOSC = 4 MHz, VDD = 2.0V, XT or RC mode(3) FOSC = 10 MHz, VDD = 3.0V, HS mode FOSC = 20 MHz, VDD = 5.0V, HS mode FOSC = 32 kHz, VDD = 2.0V, LP mode, WDT disabled
D020	IPD	**Power-down Current(2)**					
			— —	1.0 0.5	15.0 8.0	µA µA	VDD = 2.0V, WDT enabled VDD = 2.0V, WDT disabled

* These parameters are characterized but not tested.

† Data in "Typ" column is based on characterization results at 25°C. This data is for design guidance only and is 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 bus loading, oscillator type, bus rate, internal code execution pattern and temperature, also have an impact on the current consumption.

a) The test conditions for all IDD measurements in Active Operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VSS, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

b) For standby current measurements, the conditions are the same, except that the device is in Sleep mode. The Power-down Current in Sleep mode does not depend on the oscillator type.

3: Does not include current through REXT. The current through the resistor can be estimated by the formula: IR = VDD/2REXT (mA) with REXT in kΩ.

11.4 Timing Parameter Symbology and Load Conditions

The timing parameter symbols have been created with one of the following formats:

1. TppS2ppS
2. TppS

T	
F Frequency	T Time

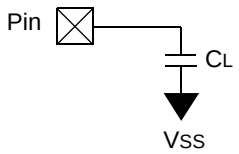
Lowercase letters (pp) and their meanings:

pp	
2 to	mc $\overline{\text{MCLR}}$
ck CLKOUT	osc oscillator
cy cycle time	os OSC1
drt device reset timer	t0 T0CKI
io I/O port	wdt watchdog timer

Uppercase letters and their meanings:

S	
F Fall	P Period
H High	R Rise
I Invalid (High-impedance)	V Valid
L Low	Z High-impedance

FIGURE 11-2: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS – PIC16F5X



Legend:

C_L = 50 pF for all pins and OSC2 for RC mode

0-15 pF for OSC2 in XT, HS or LP modes when external clock is used to drive OSC1

11.5 Timing Diagrams and Specifications

FIGURE 11-3: EXTERNAL CLOCK TIMING

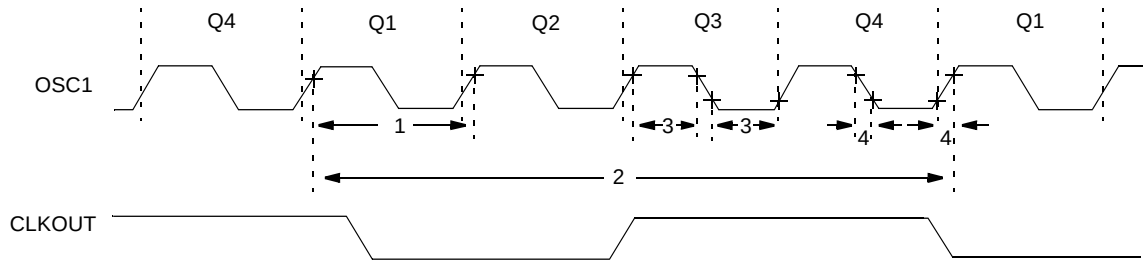


TABLE 11-1: EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS		Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Parameter No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
	FOSC	External CLKIN Frequency(1)	DC	—	4.0	MHz	XT Osc mode
			DC	—	20	MHz	HS Osc mode
			DC	—	200	kHz	LP Osc mode
		Oscillator Frequency(1)	DC	—	4.0	MHz	RC Osc mode
			0.1	—	4.0	MHz	XT Osc mode
			4.0	—	20	MHz	HS Osc mode
			5.0	—	200	kHz	LP Osc mode
1	TOSC	External CLKIN Period(1)	250	—	—	ns	XT Osc mode
			50	—	—	ns	HS Osc mode
			5.0	—	—	µs	LP Osc mode
		Oscillator Period(1)	250	—	—	ns	RC Osc mode
			250	—	10,000	ns	XT Osc mode
			50	—	250	ns	HS Osc mode
			5.0	—	—	µs	LP Osc mode
2	TCY	Instruction Cycle Time(2)	—	4/FOSC	—	—	
3	TosL, TosH	Clock in (OSC1) Low or High Time	50*	—	—	ns	XT oscillator
			20*	—	—	ns	HS oscillator
			2.0*	—	—	µs	LP oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	—	—	25*	ns	XT oscillator
			—	—	5*	ns	HS oscillator
			—	—	50*	ns	LP oscillator

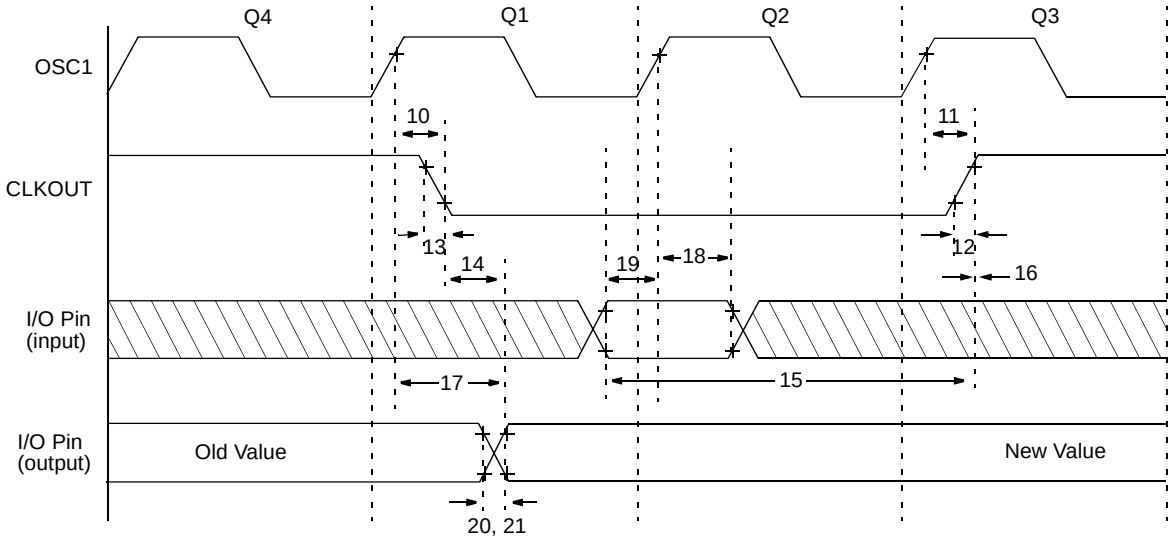
* These parameters are characterized but not tested.

† Data in the Typical ("Typ") column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: 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.
When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

2: Instruction cycle period (TCY) equals four times the input oscillator time base period.

FIGURE 11-4: CLKOUT AND I/O TIMING – PIC16F5X



Note: Please refer to Figure 11-2 for load conditions.

TABLE 11-2: CLKOUT AND I/O TIMING REQUIREMENTS – PIC16F5X

Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units
10	TosH2ckL	OSC1↑ to CLKOUT↓ **(1)**	—	15	30**	ns
11	TosH2ckH	OSC1↑ to CLKOUT↑ **(1)**	—	15	30**	ns
12	TckR	CLKOUT rise time **(1)**	—	5.0	15**	ns
13	TckF	CLKOUT fall time **(1)**	—	5.0	15**	ns
14	TckL2ioV	CLKOUT↓ to Port out valid **(1)**	—	—	40**	ns
15	TioV2ckH	Port in valid before CLKOUT↑ **(1)**	0.25 Tcy+30*	—	—	ns
16	TckH2ioI	Port in hold after CLKOUT↑ **(1)**	0*	—	—	ns
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid **(2)**	—	—	100*	ns
18	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	TBD	—	—	ns
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	TBD	—	—	ns
20	TioR	Port output rise time **(2, 3)**	—	10	25**	ns
20	TioR	Port output rise time **(2, 4)**	—	10	50**	ns
21	TioF	Port output fall time **(2, 3)**	—	10	25**	ns
21	TioF	Port output fall time **(2, 4)**	—	10	50**	ns

Legend: TBD = To Be Determined.

* These parameters are characterized but not tested.

** These parameters are design targets and are not tested. No characterization data available at this time.

† Data in the Typical ("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.

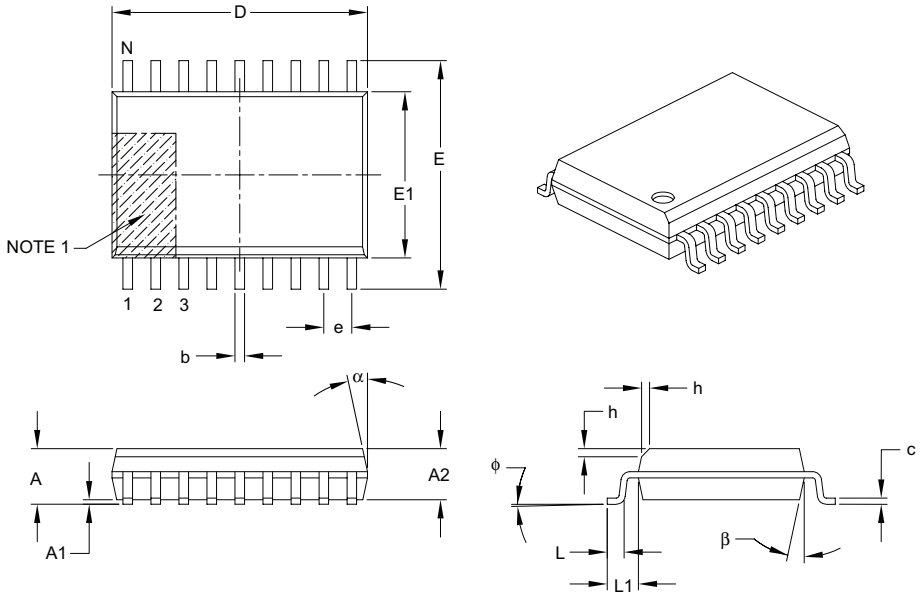
2: Please refer to Figure 11-2 for load conditions.

3: PIC16F54/57 only.

4: PIC16F59 only.

18-Lead Plastic Small Outline (SO) – Wide, 7.50 mm Body [SOIC]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		18	
Pitch	e		1.27 BSC	
Overall Height	A	–	–	2.65
Molded Package Thickness	A2	2.05	–	–
Standoff §	A1	0.10	–	0.30
Overall Width	E		10.30 BSC	
Molded Package Width	E1		7.50 BSC	
Overall Length	D		11.55 BSC	
Chamfer (optional)	h	0.25	–	0.75
Foot Length	L	0.40	–	1.27
Footprint	L1		1.40 REF	
Foot Angle	ϕ	0°	–	8°
Lead Thickness	c	0.20	–	0.33
Lead Width	b	0.31	–	0.51
Mold Draft Angle Top	α	5°	–	15°
Mold Draft Angle Bottom	β	5°	–	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

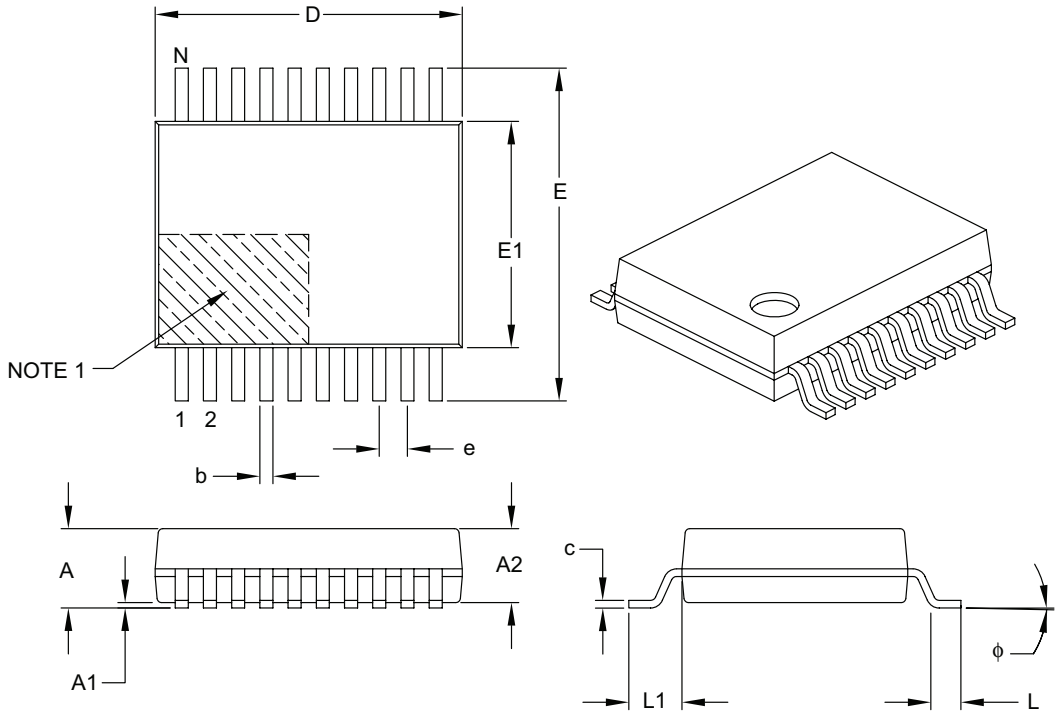
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-051B

20-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		20	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	6.90	7.20	7.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1		1.25 REF	
Lead Thickness	c	0.09	–	0.25
Foot Angle	φ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

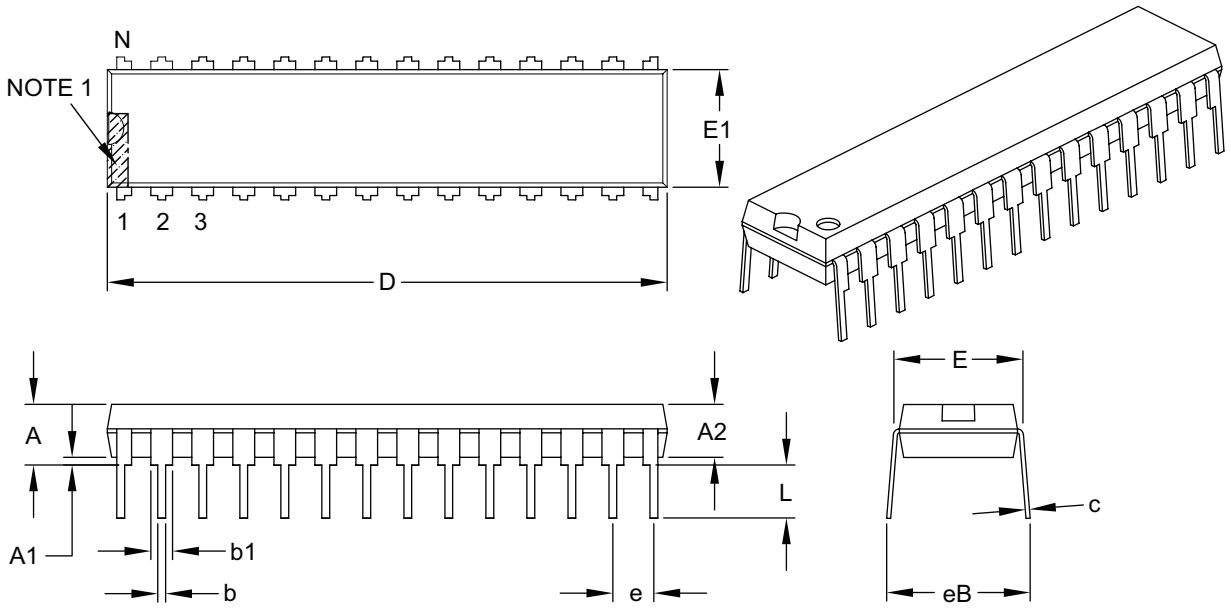
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-072B

28-Lead Skinny Plastic Dual In-Line (SP) – 300 mil Body [SPDIP]

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



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-070B

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC16F5X Literature Number: DS41213D

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

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	PIC16F54	– VDD range 2.0V to 5.5V
	PIC16F54T(1)	– VDD range 2.0V to 5.5V
	PIC16F57	– VDD range 2.0V to 5.5V
	PIC16F57T(1)	– VDD range 2.0V to 5.5V
Temperature Range	I	= -40°C to +85°C (Industrial)
	E	= -40°C to +125°C (Extended)
Package	SO	= SOIC
	SS	= SSOP
	P	= PDIP
	SP	= Skinny Plastic DIP (SPDIP)(2)
	SOG	= SOIC (Pb-free)
	SSG	= SOIC (Pb-free)
	PG	= SOIC (Pb-free)
	SPG	= SOIC (Pb-free)
Pattern	QTP, SQTP, Code or Special Requirements (blank otherwise)	

Examples:

a) PIC16F54–I/P = Industrial temp, PDIP package
b) PIC16F54T–I/SSG = Industrial temp, SSOP package (Pb -free), tape and reel
c) PIC16F57–E/SP6 = Extended temp, Skinny Plastic DIP package (Pb-free)
d) PIC16F57T–E/SS = Extended temp, SSOP package, tape and reel
e) PIC16F54–I/SOG = Industrial temp, SOIC package (Pb-free)

Note 1: T = in tape and reel SOIC and SSOP packages only.
2: PIC16F57 only

PART NO.	X	/XX	XXX
Device	Temperature Range	Package	Pattern

Device	PIC16F59	– VDD range 2.0V to 5.5V
	PIC16F59T(1)	– VDD range 2.0V to 5.5V
Temperature Range	I	= -40°C to +85°C (Industrial)
	E	= -40°C to +125°C (Extended)
Package	P	= PDIP
	PT	= TQFP
Pattern	QTP, SQTP, Code or Special Requirements (blank otherwise)	

Examples:

a) PIC16F59–I/P = Industrial temp, PDIP package (Pb-free).
b) PIC16F59T–I/PT = Industrial temp, TQFP package (Pb-free), tape and reel.

Note 1: T = in tape and reel TQFP packages only.