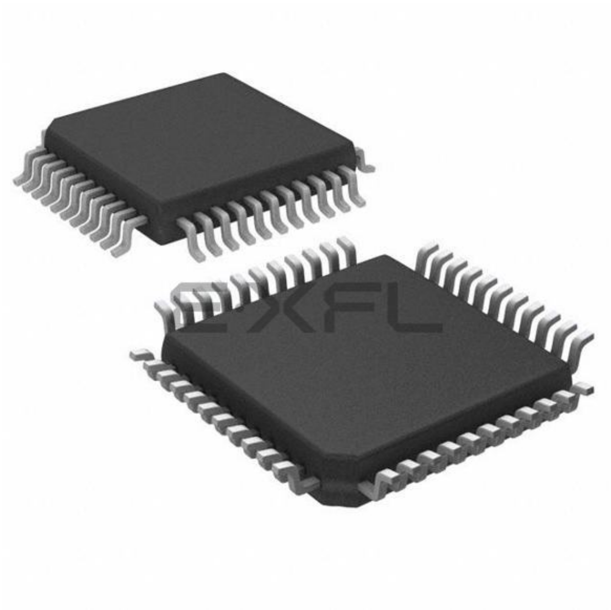


Welcome to E-XFL.COM

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	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	33
Program Memory Size	14KB (8K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	368 x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-MQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lc67t-04i-pq

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16CXX family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16CXX uses a Harvard architecture, in which, program and data are accessed from separate memories using separate buses. This improves bandwidth over traditional von Neumann architecture where program and data may be fetched from the same memory using the same bus. Separating program and data busses further allows instructions to be sized differently than 8-bit wide data words. Instruction opcodes are 14-bits 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 (Example 3-1). Consequently, all instructions execute in a single cycle (200 ns @ 20 MHz) except for program branches.

The PIC16C61 addresses 1K x 14 of program memory. The PIC16C62/62A/R62/64/64A/R64 address 2K x 14 of program memory, and the PIC16C63/R63/65/65A/R65 devices address 4K x 14 of program memory. The PIC16C66/67 address 8K x 14 program memory. All program memory is internal.

The PIC16CXX can directly or indirectly address its register files or data memory. All special function registers including the program counter are mapped in the data memory. The PIC16CXX 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” makes programming with the PIC16CXX simple yet efficient, thus significantly reducing the learning curve.

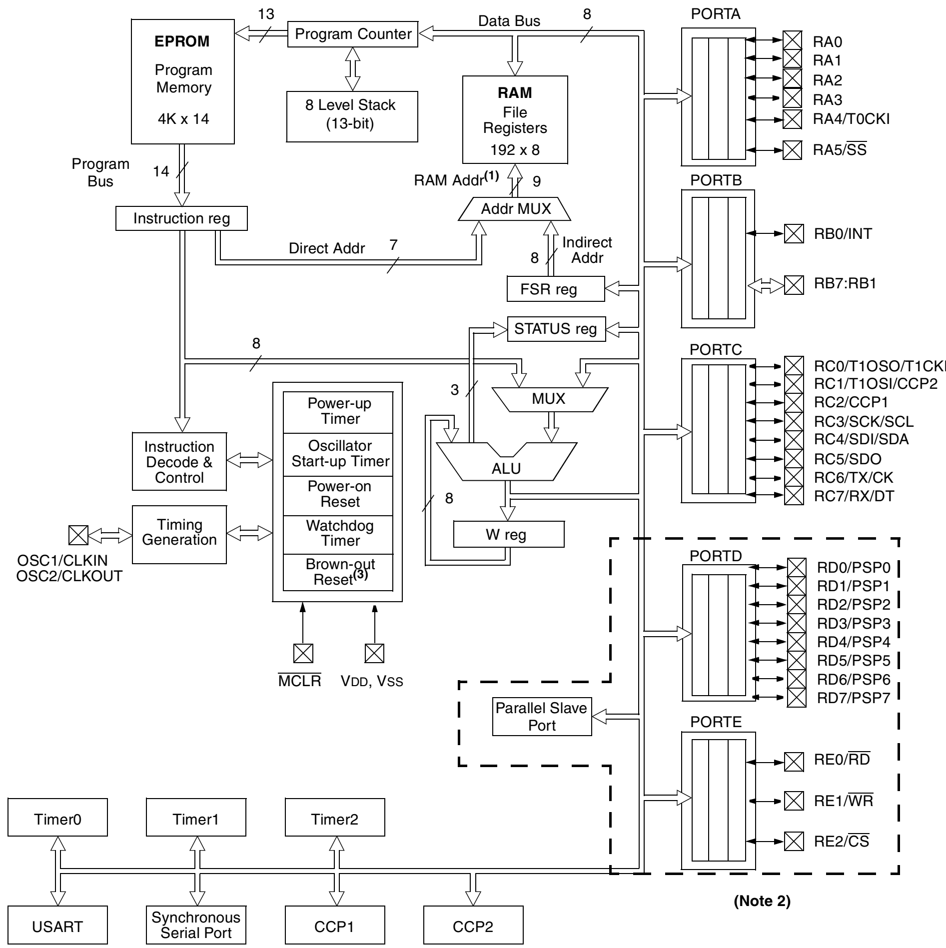
The PIC16CXX device contains an 8-bit ALU and working register (W). The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

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

FIGURE 3-3: PIC16C63/R63/65/65A/R65 BLOCK DIAGRAM



Note 1: Higher order bits are from the STATUS register.
2: PORTD, PORTE and the Parallel Slave Port are not available on the PIC16C63/R63.
3: Brown-out Reset is not available on the PIC16C65.

TABLE 4-2: SPECIAL FUNCTION REGISTERS FOR THE PIC16C62/62A/R62 (Cont.'d)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets(3)
Bank 1											
80h(1)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
81h	OPTION	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h(1)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
83h(1)	STATUS	IRP(5)	RP1(5)	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	000q quuu
84h(1)	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah(1,2)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
8Bh(1)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
8Ch	PIE1	**(6)**	**(6)**	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE	00-- 0000	00-- 0000
8Dh	—	Unimplemented								—	—
8Eh	PCON	—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$(4)	---- --qq	---- --uu
8Fh	—	Unimplemented								—	—
90h	—	Unimplemented								—	—
91h	—	Unimplemented								—	—
92h	PR2	Timer2 Period Register								1111 1111	1111 1111
93h	SSPADD	Synchronous Serial Port (I^2C mode) Address Register								0000 0000	0000 0000
94h	SSPSTAT	—	—	D/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	UA	BF	--00 0000	--00 0000
95h-9Fh	—	Unimplemented								—	—

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented location read as '0'. Shaded locations are unimplemented, read as '0'.

Note 1: These registers can be addressed from either bank.

2: The upper byte of the Program Counter (PC) is not directly accessible. PCLATH is a holding register for the PC whose contents are transferred to the upper byte of the program counter. (PC<12:8>)

3: Other (non power-up) resets include external reset through $\overline{\text{MCLR}}$ and the Watchdog Timer reset.

4: The $\overline{\text{BOR}}$ bit is reserved on the PIC16C62, always maintain this bit set.

5: The IRP and RP1 bits are reserved on the PIC16C62/62A/R62, always maintain these bits clear.

6: PIE1<7:6> and PIR1<7:6> are reserved on the PIC16C62/62A/R62, always maintain these bits clear.

6.0 OVERVIEW OF TIMER MODULES

Applicable Devices													
61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67

All PIC16C6X devices have three timer modules except for the PIC16C61, which has one timer module. Each module can generate an interrupt to indicate that an event has occurred (i.e., timer overflow). Each of these modules are detailed in the following sections. The timer modules are:

- Timer0 module (Section 7.0)
- Timer1 module (Section 8.0)
- Timer2 module (Section 9.0)

6.1 Timer0 Overview

Applicable Devices													
61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67

The Timer0 module is a simple 8-bit overflow counter. The clock source can be either the internal system clock (Fosc/4) or an external clock. When the clock source is an external clock, the Timer0 module can be selected to increment on either the rising or falling edge.

The Timer0 module also has a programmable prescaler option. This prescaler can be assigned to either the Timer0 module or the Watchdog Timer. Bit PSA (OPTION<3>) assigns the prescaler, and bits PS2:PS0 (OPTION<2:0>) determine the prescaler value. TMR0 can increment at the following rates: 1:1 when the prescaler is assigned to Watchdog Timer, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, and 1:256.

Synchronization of the external clock occurs after the prescaler. When the prescaler is used, the external clock frequency may be higher then the device's frequency. The maximum frequency is 50 MHz, given the high and low time requirements of the clock.

6.2 Timer1 Overview

Applicable Devices													
61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67

Timer1 is a 16-bit timer/counter. The clock source can be either the internal system clock (Fosc/4), an external clock, or an external crystal. Timer1 can operate as either a timer or a counter. When operating as a counter (external clock source), the counter can either operate synchronized to the device or asynchronously to the device. Asynchronous operation allows Timer1 to operate during sleep, which is useful for applications that require a real-time clock as well as the power savings of SLEEP mode.

TImer1 also has a prescaler option which allows TMR1 to increment at the following rates: 1:1, 1:2, 1:4, and 1:8. TMR1 can be used in conjunction with the Capture/Compare/PWM module. When used with a CCP module, Timer1 is the time-base for 16-bit capture or 16-bit compare and must be synchronized to the device.

6.3 Timer2 Overview

Applicable Devices													
61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67

Timer2 is an 8-bit timer with a programmable prescaler and a programmable postscaler, as well as an 8-bit Period Register (PR2). Timer2 can be used with the CCP module (in PWM mode) as well as the Baud Rate Generator for the Synchronous Serial Port (SSP). The prescaler option allows Timer2 to increment at the following rates: 1:1, 1:4, and 1:16.

The postscaler allows TMR2 register to match the period register (PR2) a programmable number of times before generating an interrupt. The postscaler can be programmed from 1:1 to 1:16 (inclusive).

6.4 CCP Overview

Applicable Devices													
61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67

The CCP module(s) can operate in one of three modes: 16-bit capture, 16-bit compare, or up to 10-bit Pulse Width Modulation (PWM).

Capture mode captures the 16-bit value of TMR1 into the CCPRxH:CCPRxL register pair. The capture event can be programmed for either the falling edge, rising edge, fourth rising edge, or sixteenth rising edge of the CCPx pin.

Compare mode compares the TMR1H:TMR1L register pair to the CCPRxH:CCPRxL register pair. When a match occurs, an interrupt can be generated and the output pin CCPx can be forced to a given state (High or Low) and Timer1 can be reset. This depends on control bits CCPxM3:CCPxM0.

PWM mode compares the TMR2 register to a 10-bit duty cycle register (CCPRxH:CCPRxL<5:4>) as well as to an 8-bit period register (PR2). When the TMR2 register = Duty Cycle register, the CCPx pin will be forced low. When TMR2 = PR2, TMR2 is cleared to 00h, an interrupt can be generated, and the CCPx pin (if an output) will be forced high.

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.

Note:	To avoid an unintended device RESET, the following instruction sequence (shown in Example 7-1) must be executed when changing the prescaler assignment from Timer0 to the WDT. This precaution must be followed even if the WDT is disabled.

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

Lines 2 and 3 do NOT have to be included if the final desired prescale value is other than 1:1. If 1:1 is final desired value, then a temporary prescale value is set in lines 2 and 3 and the final prescale value will be set in lines 10 and 11.

```
1)  BSF     STATUS, RP0    ;Bank 1
2)  MOVLW   b'xx0x0xxx'    ;Select clock source and prescale value of
3)  MOVWF   OPTION_REG     ;other than 1:1
4)  BCF     STATUS, RP0    ;Bank 0
5)  CLRF    TMR0           ;Clear TMR0 and prescaler
6)  BSF     STATUS, RP1    ;Bank 1
7)  MOVLW   b'xxxx1xxx'    ;Select WDT, do not change prescale value
8)  MOVWF   OPTION_REG     ;
9)  CLRWDT                 ;Clears WDT and prescaler
10) MOVLW   b'xxxx1xxx'    ;Select new prescale value and WDT
11) MOVWF   OPTION_REG     ;
12) BCF     STATUS, RP0    ;Bank 0
```

To change prescaler from the WDT to the Timer0 module, use the sequence shown in Example 7-2.

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

```
CLRWDT                 ;Clear WDT and prescaler
BSF      STATUS, RP0   ;Bank 1
MOVLW    b'xxxx0xxx'   ;Select TMR0, new prescale value and clock source
MOVWF    OPTION_REG    ;
BCF      STATUS, RP0   ;Bank 0
```

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, BOR	Value on all other resets
01h, 101h	TMR0	Timer0 module’s register								xxxx xxxx	uuuu uuuu
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE(1)	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
81h, 181h	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	—	—	PORTA Data Direction Register(1)						--11 1111	--11 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by Timer0.
Note 1: TRISA<5> and bit PEIE are not implemented on the PIC16C61, read as '0'.

EXAMPLE 10-2: PWM PERIOD AND DUTY CYCLE CALCULATION

Desired PWM frequency is 78.125 kHz,
Fosc = 20 MHz
TMR2 prescale = 1

$1/78.125\ \text{kHz} = [(PR2) + 1] \cdot 4 \cdot 1/20\ \text{MHz} \cdot 1$

$12.8\ \mu s = [(PR2) + 1] \cdot 4 \cdot 50\ \text{ns} \cdot 1$

$PR2 = 63$

Find the maximum resolution of the duty cycle that can be used with a 78.125 kHz frequency and 20 MHz oscillator:

$1/78.125\ \text{kHz} = 2^{\text{PWM RESOLUTION}} \cdot 1/20\ \text{MHz} \cdot 1$

$12.8\ \mu s = 2^{\text{PWM RESOLUTION}} \cdot 50\ \text{ns} \cdot 1$

$256 = 2^{\text{PWM RESOLUTION}}$

$\log(256) = (\text{PWM Resolution}) \cdot \log(2)$

$8.0 = \text{PWM Resolution}$

At most, an 8-bit resolution duty cycle can be obtained from a 78.125 kHz frequency and a 20 MHz oscillator, i.e., $0 \leq$ CCPR1L:CCP1CON<5:4> ≤ 255. Any value greater than 255 will result in a 100% duty cycle.

In order to achieve higher resolution, the PWM frequency must be decreased. In order to achieve higher PWM frequency, the resolution must be decreased.

Table 10-3 lists example PWM frequencies and resolutions for Fosc = 20 MHz. The TMR2 prescaler and PR2 values are also shown.

10.3.3 SET-UP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPR1L register and CCP1CON<5:4> bits.
3. Make the CCP1 pin an output by clearing the TRISC<2> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCP1 module for PWM operation.

TABLE 10-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHz

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	5.5

TABLE 10-4: REGISTERS ASSOCIATED WITH TIMER1, CAPTURE AND COMPARE

Add	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh,8Bh 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(2)	(3)	RCIF(1)	TXIF(1)	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh(4)	PIR2	—	—	—	—	—	—	—	CCP2IF	---- ---0	---- ---0
8Ch	PIE1	PSPIE(2)	(3)	RCIE(1)	TXIE(1)	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh(4)	PIE2	—	—	—	—	—	—	—	CCP2IE	---- ---0	---- ---0
87h	TRISC	PORTC Data Direction register								1111 1111	1111 1111
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	--00 0000	--uu uuuu
15h	CCPR1L	Capture/Compare/PWM1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh(4)	CCPR2L	Capture/Compare/PWM2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch(4)	CCPR2H	Capture/Compare/PWM2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh(4)	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used in these modes.
Note 1: These bits are associated with the USART module, which is implemented on the PIC16C63/R63/65/65A/R65/66/67 only.
2: Bits PSPIE and PSPIF are reserved on the PIC16C62/62A/R62/63/R63/66, always maintain these bits clear.
3: The PIR1<6> and PIE1<6> bits are reserved, always maintain these bits clear.
4: These registers are associated with the CCP2 module, which is only implemented on the PIC16C63/R63/65/65A/R65/66/67.

FIGURE 11-8: SSPCON: SYNC SERIAL PORT CONTROL REGISTER (ADDRESS 14h)(PIC16C66/67)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
- n =Value at POR reset

bit 7: **WCOL**: Write Collision Detect bit
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

bit 6: **SSPOV**: Receive Overflow Indicator bit

In SPI mode
1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In master mode the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.
0 = No overflow

In I^2C mode
1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don’t care" in transmit mode. SSPOV must be cleared in software in either mode.
0 = No overflow

bit 5: **SSPEN**: Synchronous Serial Port Enable bit

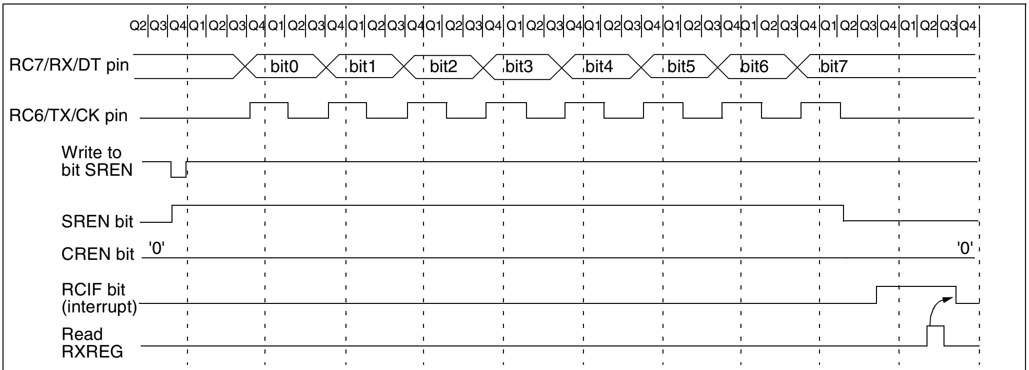
In SPI mode
1 = Enables serial port and configures SCK, SDO, and SDI as serial port pins
0 = Disables serial port and configures these pins as I/O port pins

In I^2C mode
1 = Enables the serial port and configures the SDA and SCL pins as serial port pins
0 = Disables serial port and configures these pins as I/O port pins
In both modes, when enabled, these pins must be properly configured as input or output.

bit 4: **CKP**: Clock Polarity Select bit
In SPI mode
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level
In I^2C mode
SCK release control
1 = Enable clock
0 = Holds clock low (clock stretch) (Used to ensure data setup time)

bit 3-0: **SSPM3:SSPM0**: Synchronous Serial Port Mode Select bits
`0000` = SPI master mode, clock = FOSC/4
`0001` = SPI master mode, clock = FOSC/16
`0010` = SPI master mode, clock = FOSC/64
`0011` = SPI master mode, clock = TMR2 output/2
`0100` = SPI slave mode, clock = SCK pin. $\overline{SS}$ pin control enabled.
`0101` = SPI slave mode, clock = SCK pin. $\overline{SS}$ pin control disabled. $\overline{SS}$ can be used as I/O pin
`0110` = I^2C slave mode, 7-bit address
`0111` = I^2C slave mode, 10-bit address
`1011` = I^2C firmware controlled master mode (slave idle)
`1110` = I^2C slave mode, 7-bit address with start and stop bit interrupts enabled
`1111` = I^2C slave mode, 10-bit address with start and stop bit interrupts enabled

FIGURE 12-14: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



Note: Timing diagram demonstrates SYNC master mode with bit SREN = '1' and bit BRG = '0'.

TABLE 13-1: CERAMIC RESONATORS PIC16C61

Ranges Tested:			
Mode	**Freq**	**OSC1**	**OSC2**
XT	455 kHz 2.0 MHz 4.0 MHz	47 - 100 pF 15 - 68 pF 15 - 68 pF	47 - 100 pF 15 - 68 pF 15 - 68 pF
HS	8.0 MHz 16.0 MHz	15 - 68 pF 10 - 47 pF	15 - 68 pF 10 - 47 pF
These values are for design guidance only. See notes at bottom of page.			
Resonators Used:			
455 kHz	Panasonic EFO-A455K04B	± 0.3%	
2.0 MHz	Murata Erie CSA2.00MG	± 0.5%	
4.0 MHz	Murata Erie CSA4.00MG	± 0.5%	
8.0 MHz	Murata Erie CSA8.00MT	± 0.5%	
16.0 MHz	Murata Erie CSA16.00MX	± 0.5%	
All resonators used did not have built-in capacitors.			

TABLE 13-2: CERAMIC RESONATORS PIC16C62/62A/R62/63/R63/64/64A/R64/65/65A/R65/66/67

Ranges Tested:			
Mode	**Freq**	**OSC1**	**OSC2**
XT	455 kHz 2.0 MHz 4.0 MHz	68 - 100 pF 15 - 68 pF 15 - 68 pF	68 - 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
These values are for design guidance only. See notes at bottom of page.			
Resonators Used:			
455 kHz	Panasonic EFO-A455K04B	± 0.3%	
2.0 MHz	Murata Erie CSA2.00MG	± 0.5%	
4.0 MHz	Murata Erie CSA4.00MG	± 0.5%	
8.0 MHz	Murata Erie CSA8.00MT	± 0.5%	
16.0 MHz	Murata Erie CSA16.00MX	± 0.5%	
All resonators used did not have built-in capacitors.			

TABLE 13-3: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR FOR PIC16C61

Mode	Freq	OSC1	OSC2
LP	32 kHz 200 kHz	33 - 68 pF 15 - 47 pF	33 - 68 pF 15 - 47 pF
XT	100 kHz 500 kHz 1 MHz 2 MHz 4 MHz	47 - 100 pF 20 - 68 pF 15 - 68 pF 15 - 47 pF 15 - 33 pF	47 - 100 pF 20 - 68 pF 15 - 68 pF 15 - 47 pF 15 - 33 pF
HS	8 MHz 20 MHz	15 - 47 pF 15 - 47 pF	15 - 47 pF 15 - 47 pF
These values are for design guidance only. See notes at bottom of page.			

TABLE 13-4: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR FOR PIC16C62/62A/R62/63/R63/64/64A/R64/65/65A/R65/66/67

Osc Type	Crystal Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF
These values are for design guidance only. See notes at bottom of page.			
Crystals Used			
32 kHz	Epson C-001R32.768K-A	± 20 PPM	
200 kHz	STD XTL 200.000KHz	± 20 PPM	
1 MHz	ECS ECS-10-13-1	± 50 PPM	
4 MHz	ECS ECS-40-20-1	± 50 PPM	
8 MHz	EPSON CA-301 8.000M-C	± 30 PPM	
20 MHz	EPSON CA-301 20.000M-C	± 30 PPM	

Note 1: Recommended values of C1 and C2 are identical to the ranges tested Table 13-1 and Table 13-2.
2: Higher capacitance increases the stability of oscillator but also increases the start-up time.
3: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.
4: Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification.

13.8 Power-down Mode (SLEEP)

Applicable Devices														
61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67	

Power-down mode is entered by executing a `SLEEP` instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, status bit $\overline{PD}$ (STATUS<3>) is cleared, status bit $\overline{TO}$ (STATUS<4>) is set, and the oscillator driver is turned off. The I/O ports maintain the status they had before the `SLEEP` instruction was executed (driving high, low, or hi-impedance).

For lowest current consumption in this mode, place all I/O pins at either V_{DD}, or V_{SS}, ensure no external circuitry is drawing current from the I/O pin, and disable external clocks. Pull all I/O pins, that are hi-impedance inputs, high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at V_{DD} or V_{SS} for lowest current consumption. The contribution from on-chip pull-ups on PORTB should be considered.

The $\overline{MCLR}/V_{PP}$ pin must be at a logic high level (V_{IHMC}).

13.8.1 WAKE-UP FROM SLEEP

The device can wake from SLEEP through one of the following events:

1. External reset input on $\overline{MCLR}/V_{PP}$ pin.
2. Watchdog Timer Wake-up (if WDT was enabled).
3. Interrupt from RB0/INT pin, RB port change, or some peripheral interrupts.

External $\overline{MCLR}$ Reset will cause a device reset. All other events are considered a continuation of program execution and cause a “wake-up”. The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register can be used to determine the cause of device reset. The $\overline{PD}$ bit, which is set on power-up is cleared when `SLEEP` is invoked. The $\overline{TO}$ bit is cleared if WDT time-out occurred (and caused wake-up).

The following peripheral interrupts can wake the device from SLEEP:

1. TMR1 interrupt. Timer1 must be operating as an asynchronous counter.
2. SSP (Start/Stop) bit detect interrupt.
3. SSP transmit or receive in slave mode (SPI/I^2C).
4. CCP capture mode interrupt.
5. Parallel Slave Port read or write.
6. USART TX or RX (synchronous slave mode).

Other peripherals can not generate interrupts since during SLEEP, no on-chip Q clocks are present.

When the `SLEEP` instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the `SLEEP` instruction. If the GIE bit is set (enabled), the device executes the instruction after the `SLEEP` instruction and then branches to the interrupt address (0004h). In cases where the execution of the instruction following `SLEEP` is not desirable, the user should have a `NOP` after the `SLEEP` instruction.

13.8.2 WAKE-UP USING INTERRUPTS

When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a `SLEEP` instruction, the `SLEEP` instruction will complete as a NOP. Therefore, the WDT and WDT postscaler will not be cleared, the $\overline{TO}$ bit will not be set and $\overline{PD}$ bits will not be cleared.
- If the interrupt occurs **during or after** the execution of a `SLEEP` instruction, the device will immediately wake up from sleep. The `SLEEP` instruction will be completely executed before the wake-up. Therefore, the WDT and WDT postscaler will be cleared, the $\overline{TO}$ bit will be set and the $\overline{PD}$ bit will be cleared.

Even if the flag bits were checked before executing a `SLEEP` instruction, it may be possible for flag bits to become set before the `SLEEP` instruction completes. To determine whether a `SLEEP` instruction executed, test the $\overline{PD}$ bit. If the $\overline{PD}$ bit is set, the `SLEEP` instruction was executed as a NOP.

To ensure that the WDT is cleared, a `CLRWDT` instruction should be executed before a `SLEEP` instruction.

14.0 INSTRUCTION SET SUMMARY

Each PIC16CXX instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CXX instruction set summary in Table 14-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 14-1 shows the opcode field descriptions.

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

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

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

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

TABLE 14-1: OPCODE FIELD DESCRIPTIONS

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

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

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

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

Table 14-2 lists the instructions recognized by the MPASM assembler.

Figure 14-1 shows the general formats that the instructions can have.

> **Note:** To maintain upward compatibility with future PIC16CXX products, do not use the `OPTION` and `TRIS` instructions.

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

0xhh

where h signifies a hexadecimal digit.

FIGURE 14-1: GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations

13 — 8	7	6 — 0
OPCODE	d	f (FILE #)

d = 0 for destination W
d = 1 for destination f
f = 7-bit file register address

Bit-oriented file register operations

13 — 10	9 — 7	6 — 0
OPCODE	b (BIT #)	f (FILE #)

b = 3-bit bit address
f = 7-bit file register address

Literal and control operations

General

13 — 8	7 — 0
OPCODE	k (literal)

k = 8-bit immediate value

`CALL` and `GOTO` instructions only

13 — 11	10 — 0
OPCODE	k (literal)

k = 11-bit immediate value

GOTO — Unconditional Branch

Syntax:	[*label*] GOTO k
Operands:	0 ≤ k ≤ 2047
Operation:	k → PC<10:0> PCLATH<4:3> → PC<12:11>
Status Affected:	None

Encoding:

10	1kkk	kkkk	kkkk

Description: `GOTO` is an unconditional branch. The eleven bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. `GOTO` is a two cycle instruction.

Words: 1

Cycles: 2

Q Cycle Activity:

	Q1	Q2	Q3	Q4
1st Cycle	Decode	Read literal 'k'	Process data	Write to PC
2nd Cycle	No-Operation	No-Operation	No-Operation	No-Operation

Example: `GOTO THERE`

After Instruction
PC = Address `THERE`

INCF — Increment f

Syntax:	[*label*] INCF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (destination)
Status Affected:	Z

Encoding:

00	1010	dfff	ffff

Description: The contents of register 'f' are incremented. 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

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

Example: `INCF CNT, 1`

Before Instruction
CNT = 0xFF
Z = 0
After Instruction
CNT = 0x00
Z = 1

Applicable Devices	61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67

19.3 DC Characteristics: PIC16C65-04 (Commercial, Industrial) PIC16C65-10 (Commercial, Industrial) PIC16C65-20 (Commercial, Industrial) PIC16LC65-04 (Commercial, Industrial)

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 Operating voltage VDD range as described in DC spec Section 19.1 and Section 19.2					
Param No.	Characteristic	Sym	Min	Typ †	Max	Units	Conditions
	Input Low Voltage						
	I/O ports	VIL					
D030	with TTL buffer		VSS	-	0.15VDD	V	For entire VDD range
D030A			VSS	-	0.8V	V	4.5V ≤ VDD ≤ 5.5V
D031	with Schmitt Trigger buffer		VSS	-	0.2VDD	V	
D032	MCLR, OSC1(in RC mode)		Vss	-	0.2VDD	V	
D033	OSC1 (in XT, HS and LP)		Vss	-	0.3VDD	V	Note1
	Input High Voltage						
	I/O ports	VIH		-			
D040	with TTL buffer		2.0	-	VDD	V	4.5V ≤ VDD ≤ 5.5V
D040A			0.25VDD + 0.8V	-	VDD	V	For entire VDD range
D041	with Schmitt Trigger buffer		0.8VDD	-	VDD		For entire VDD range
D042	MCLR		0.8VDD	-	VDD	V	
D042A	OSC1 (XT, HS and LP)		0.7 VDD	-	VDD	V	Note1
D043	OSC1 (in RC mode)		0.9VDD	-	VDD	V	
D070	PORTB weak pull-up current	IPURB	50	250	400	μA	VDD = 5V, VPIN = VSS
	Input Leakage Current (Notes 2, 3)						
D060	I/O ports	IIL	-	-	±1	μA	Vss ≤ VPIN ≤ VDD, Pin at hi-impedance
D061	MCLR, RA4/T0CKI		-	-	±5	μA	Vss ≤ VPIN ≤ VDD
D063	OSC1		-	-	±5	μA	Vss ≤ VPIN ≤ VDD, XT, HS, and LP osc configuration
	Output Low Voltage						
D080	I/O ports	VOL	-	-	0.6	V	IOL = 8.5 mA, VDD = 4.5V, -40°C to +85°C
D083	OSC2/CLKOUT (RC osc config)		-	-	0.6	V	IOL = 1.6 mA, VDD = 4.5V, -40°C to +85°C
	Output High Voltage						
D090	I/O ports (Note 3)	VOH	VDD-0.7	-	-	V	IOH = -3.0 mA, VDD = 4.5V, -40°C to +85°C
D092	OSC2/CLKOUT (RC osc config)		VDD-0.7	-	-	V	IOH = -1.3 mA, VDD = 4.5V, -40°C to +85°C
D150*	**Open-Drain High Voltage**	VOD	-	-	14	V	RA4 pin

* These parameters are characterized but not tested.

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

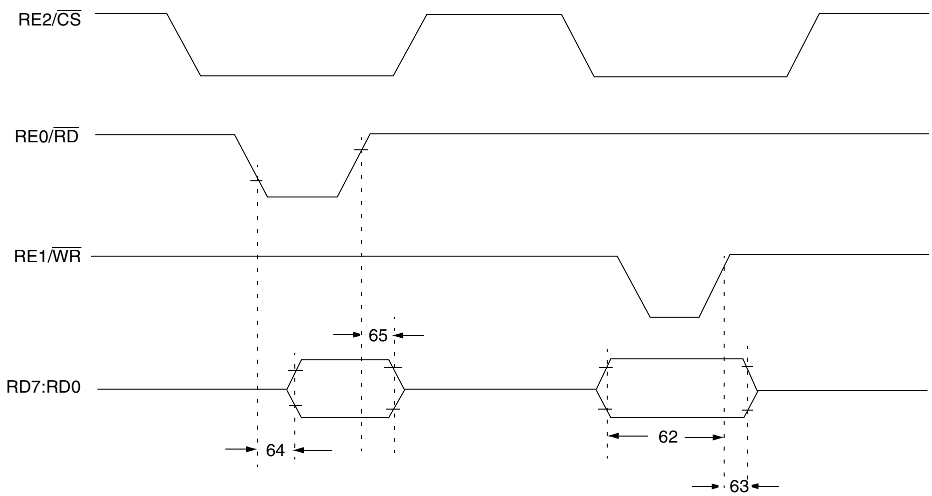
Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16C6X be driven with external clock in RC mode.

2: The leakage current on the MCLR/VPP pin is strongly dependent on the 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 current sourced by the pin.

Applicable Devices	61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67

FIGURE 20-8: PARALLEL SLAVE PORT TIMING (PIC16C65A)



Note: Refer to Figure 20-1 for load conditions

TABLE 20-7: PARALLEL SLAVE PORT REQUIREMENTS (PIC16C65A)

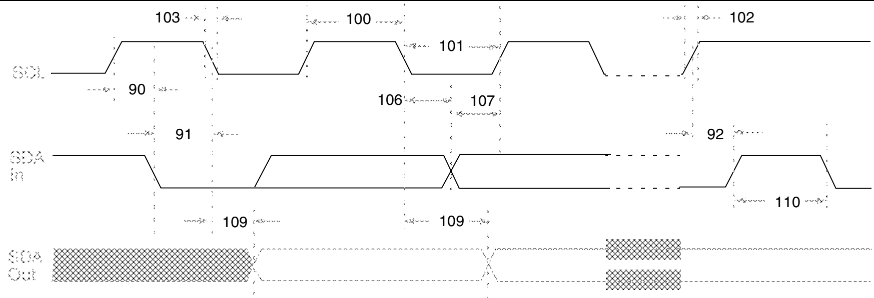
Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
62*	TdtV2wrH	Data in valid before $\overline{\text{WR}}$↑ or $\overline{\text{CS}}$↑ (setup time)		20	—	—	ns	
				25	—	—	ns	Extended Range Only
63*	TwrH2dtI	$\overline{\text{WR}}$↑ or $\overline{\text{CS}}$↑ to data–in invalid (hold time)	PIC16**C**65A	20	—	—	ns	
			PIC16**LC**65A	35	—	—	ns	
64	TrdL2dtV	$\overline{\text{RD}}$↓ and $\overline{\text{CS}}$↓ to data–out valid		—	—	80	ns	
				—	—	90	ns	Extended Range Only
65*	TrdH2dtI	$\overline{\text{RD}}$↑ or $\overline{\text{CS}}$↑ to data–out invalid		10	—	30	ns	

* These parameters are characterized but not tested.

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

Applicable Devices	61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67

FIGURE 20-11: I^2C BUS DATA TIMING



Note: Refer to Figure 20-1 for load conditions

TABLE 20-10: I^2C BUS DATA REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Max	Units	Conditions
100*	THIGH	Clock high time	100 kHz mode	4.0	—	µs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	µs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5TCY	—		
101*	TLOW	Clock low time	100 kHz mode	4.7	—	µs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	µs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5TCY	—		
102*	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1Cb	300	ns	Cb is specified to be from 10-400 pF
103*	TF	SDA and SCL fall time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1Cb	300	ns	Cb is specified to be from 10-400 pF
90*	TSU:STA	START condition setup time	100 kHz mode	4.7	—	µs	Only relevant for repeated START condition
			400 kHz mode	0.6	—	µs	
91*	THD:STA	START condition hold time	100 kHz mode	4.0	—	µs	After this period the first clock pulse is generated
			400 kHz mode	0.6	—	µs	
106*	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	µs	
107*	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	Note 2
			400 kHz mode	100	—	ns	
92*	TSU:STO	STOP condition setup time	100 kHz mode	4.7	—	µs	
			400 kHz mode	0.6	—	µs	
109*	TAA	Output valid from clock	100 kHz mode	—	3500	ns	Note 1
			400 kHz mode	—	—	ns	
110*	TBUF	Bus free time	100 kHz mode	4.7	—	µs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	µs	
	Cb	Bus capacitive loading		—	400	pF	

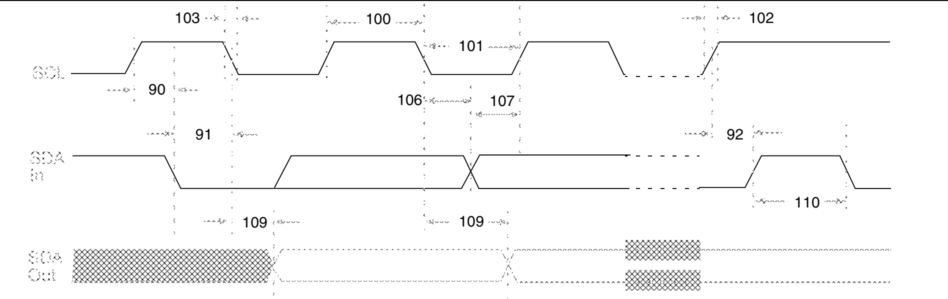
* These parameters are characterized but not tested.

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

2: A fast-mode (400 kHz) I^2C-bus device can be used in a standard-mode (100 kHz) I^2C-bus system, but the requirement Tsu:DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line TR max.+tsu;DAT = 1000 + 250 = 1250 ns (according to the standard-mode I^2C bus specification) before the SCL line is released.

Applicable Devices	61	62	62A	R62	63	R63	64	64A	R64	65	65A	R65	66	67

FIGURE 21-11: I^{2}C BUS DATA TIMING



Note: Refer to Figure 21-1 for load conditions

TABLE 21-10: I^{2}C BUS DATA REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Max	Units	Conditions
100*	THIGH	Clock high time	100 kHz mode	4.0	—	µs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	µs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5TCY	—		
101*	TLOW	Clock low time	100 kHz mode	4.7	—	µs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	µs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5TCY	—		
102*	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1Cb	300	ns	Cb is specified to be from 10-400 pF
103*	TF	SDA and SCL fall time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1Cb	300	ns	Cb is specified to be from 10-400 pF
90*	TSU:STA	START condition setup time	100 kHz mode	4.7	—	µs	Only relevant for repeated START condition
			400 kHz mode	0.6	—	µs	
91*	THD:STA	START condition hold time	100 kHz mode	4.0	—	µs	After this period the first clock pulse is generated
			400 kHz mode	0.6	—	µs	
106*	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	µs	
107*	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	Note 2
			400 kHz mode	100	—	ns	
92*	TSU:STO	STOP condition setup time	100 kHz mode	4.7	—	µs	
			400 kHz mode	0.6	—	µs	
109*	TAA	Output valid from clock	100 kHz mode	—	3500	ns	Note 1
			400 kHz mode	—	—	ns	
110*	TBUF	Bus free time	100 kHz mode	4.7	—	µs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	µs	
	Cb	Bus capacitive loading		—	400	pF	

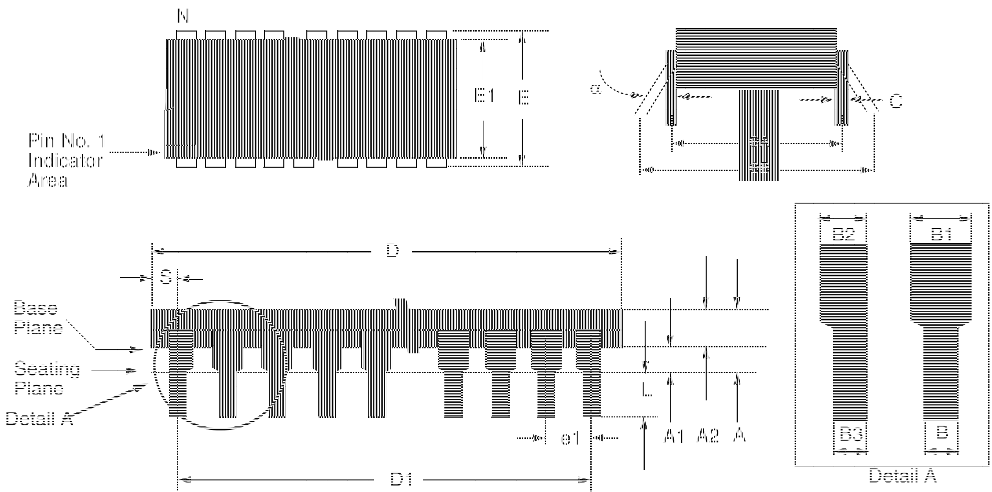
* These parameters are characterized but not tested.

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

2: A fast-mode (400 kHz) I^{2}C-bus device can be used in a standard-mode (100 kHz) I^{2}C-bus system, but the requirement Tsu:DAT $\geq$ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line TR max.+tsu;DAT = 1000 + 250 = 1250 ns (according to the standard-mode I^{2}C bus specification) before the SCL line is released.

24.2 28-Lead Plastic Dual In-line (300 mil) (SP)

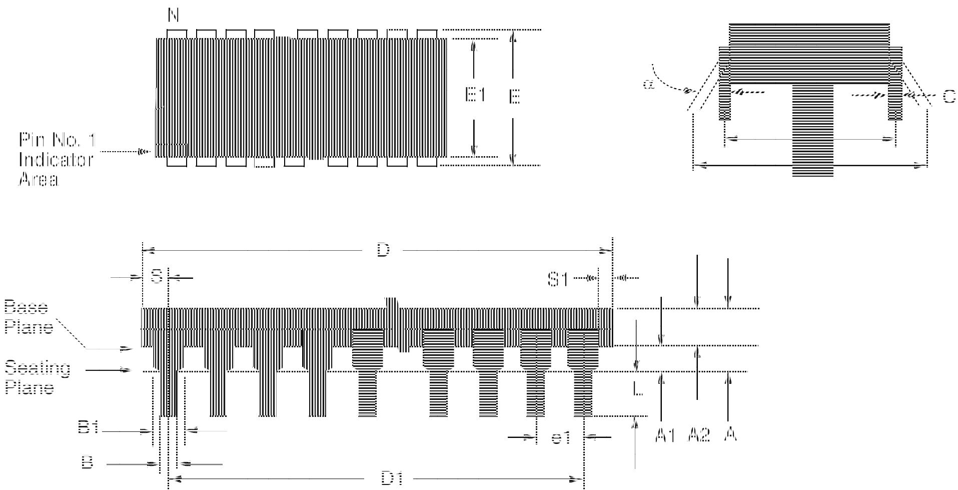
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Package Group: Plastic Dual In-Line (PLA)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.632	4.572		0.143	0.180	
A1	0.381	–		0.015	–	
A2	3.175	3.556		0.125	0.140	
B	0.406	0.559		0.016	0.022	
B1	1.016	1.651	**Typical**	0.040	0.065	**Typical**
B2	0.762	1.016	**4 places**	0.030	0.040	**4 places**
B3	0.203	0.508	**4 places**	0.008	0.020	**4 places**
C	0.203	0.331	**Typical**	0.008	0.013	**Typical**
D	34.163	35.179		1.385	1.395	
D1	33.020	33.020	**Reference**	1.300	1.300	**Reference**
E	7.874	8.382		0.310	0.330	
E1	7.112	7.493		0.280	0.295	
e1	2.540	2.540	**Typical**	0.100	0.100	**Typical**
eA	7.874	7.874	**Reference**	0.310	0.310	**Reference**
eB	8.128	9.652		0.320	0.380	
L	3.175	3.683		0.125	0.145	
N	28	28		28	28	
S	0.584	1.220		0.023	0.048	

24.3 40-Lead Plastic Dual In-line (600 mil) (P)

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



Package Group: Plastic Dual In-Line (PLA)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	–	5.080		–	0.200	
A1	0.381	–		0.015	–	
A2	3.175	4.064		0.125	0.160	
B	0.355	0.559		0.014	0.022	
B1	1.270	1.778	**Typical**	0.050	0.070	**Typical**
C	0.203	0.381	**Typical**	0.008	0.015	**Typical**
D	51.181	52.197		2.015	2.055	
D1	48.260	48.260	**Reference**	1.900	1.900	**Reference**
E	15.240	15.875		0.600	0.625	
E1	13.462	13.970		0.530	0.550	
e1	2.489	2.591	**Typical**	0.098	0.102	**Typical**
eA	15.240	15.240	**Reference**	0.600	0.600	**Reference**
eB	15.240	17.272		0.600	0.680	
L	2.921	3.683		0.115	0.145	
N	40	40		40	40	
S	1.270	–		0.050	–	
S1	0.508	–		0.020	–	

INDEX

Numerics

D

E

F

G

H

I