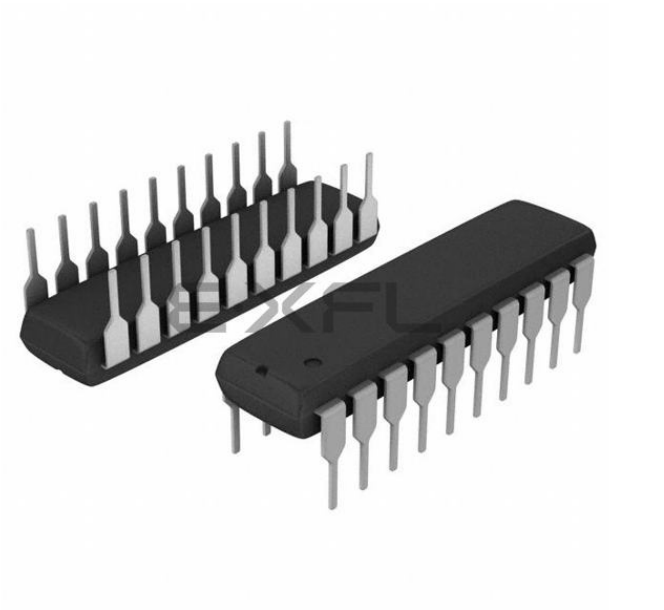


What is "Embedded - Microcontrollers"?

"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	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	18
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 17x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f18345-i-p

TABLE 2: 20-PIN ALLOCATION TABLE (PIC16(L)F18345)

I/O[2]	20-Pin PDIP/SOIC/SSOP	20-Pin UQFN	ADC	Reference	Comparator	NCO	DAC	DSM	Timers	CCP	PWM	CWG	MSSP	EUSART	CLC	CLKR	Interrupt	Pull-up	Basic
RA0	19	16	ANA0	—	C1IN0+	—	DAC1OUT	—	—	—	—	—	—	—	—	—	IOC	Y	ICDDAT ICSPDAT
RA1	18	15	ANA1	VREF+	C1IN0- C2IN0-	—	DAC1REF+	—	—	—	—	—	$\overline{\text{SS2}}$	—	—	—	IOC	Y	ICDCLK ICSPCLK
RA2	17	14	ANA2	VREF-	—	—	DAC1REF-	—	T0CKI[1]	CCP3[1]	—	CWG1IN[1] CWG2IN[1]	—	—	CLCIN0[1]	—	IOC INT[1]	Y	—
RA3	4	1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IOC	Y	$\overline{\text{MCLR}}$ VPP
RA4	3	20	ANA4	—	—	—	—	—	T1G[1] T3G[1] T5G[1] SOSCO	CCP4[1]	—	—	—	—	—	—	IOC	Y	CLKOUT OSC2
RA5	2	19	ANA5	—	—	—	—	—	T1CKI[1] T3CKI[1] T5CKI[1] SOSCIN SOSCI	—	—	—	—	—	—	—	IOC	Y	CLKIN OSC1
RB4	13	10	ANB4	—	—	—	—	—	—	—	—	—	SDI1[1] SDA1[1,3,4]	—	CLCIN2[1]	—	IOC	Y	—
RB5	12	9	ANB5	—	—	—	—	—	—	—	—	—	SDI2[1] SDA2[1,3,4]	RX[1]	CLCIN3[1]	—	IOC	Y	—
RB6	11	8	ANB6	—	—	—	—	—	—	—	—	—	SCK1[1] SCL1[1,3,4]	—	—	—	IOC	Y	—
RB7	10	7	ANB7	—	—	—	—	—	—	—	—	—	SCK2[1] SCL2[1,3,4]	—	—	—	IOC	Y	—
RC0	16	13	ANC0	—	C2IN0+	—	—	—	—	—	—	—	—	—	—	—	IOC	Y	—
RC1	15	12	ANC1	—	C1IN1- C2IN1-	—	—	—	—	—	—	—	—	—	—	—	IOC	Y	—
RC2	14	11	ANC2	—	C1IN2- C2IN2-	—	—	MDCIN1[1]	—	—	—	—	—	—	—	—	IOC	Y	—

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.
2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers.
3: These peripheral functions are bidirectional. The output pin selections must be the same as the input pin selections.
4: These pins are configured for I^2C logic levels; clock and data signals may be assigned to any of these pins. Assignments to other pins (e.g. RA5) will operate, but logic levels will be standard TTL/ST as selected by the INLVL register.

TABLE 4-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18325	PIC16(L)F18345	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
Bank 9													
CPU CORE REGISTERS; see Table 4-2 for specifics													
48Ch to 497h	—	—		Unimplemented								—	—
498h	NCO1ACCL			NCO1ACC<7:0>								`0000 0000`	`0000 0000`
499h	NCO1ACCH			NCO1ACC<15:8>								`0000 0000`	`0000 0000`
49Ah	NCO1ACCU			—	—	—	—	NCO1ACC<19:16>				`---- 0000`	`---- 0000`
49Bh	NCO1INCL			NCO1INC<7:0>								`0000 0001`	`0000 0001`
49Ch	NCO1INCH			NCO1INC<15:8>								`0000 0000`	`0000 0000`
49Dh	NCO1INCU			—	—	—	—	NCO1INC<19:16>				`---- 0000`	`---- 0000`
49Eh	NCO1CON			N1EN	—	N1OUT	N1POL	—	—	—	N1PFM	`0-00 ---0`	`0-00 ---0`
49Fh	NCO1CLK			N1PWS<2:0>			—	—	—	N1CKS<1:0>		`000- --00`	`000- --00`

Legend: x = unknown, u = unchanged, q =depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.
Note 1: Only on PIC16F18325/18345.
2: Register accessible from both User and ICD Debugger.

Preliminary

7.2.2 INTERNAL CLOCK SOURCES

The device may be configured to use the internal oscillator block as the system clock by performing one of the following actions:

- Program the RSTOSC<2:0> bits in Configuration Words to select the INTOSC clock source, which will be used as the default system clock upon a device Reset.
- Write the NOSC<2:0> bits in the OSCCON1 register to switch the system clock source to the internal oscillator during run-time. See **Section 7.3 "Clock Switching"** for more information.

The function of the OSC2/CLKOUT pin is determined by the $\overline{\text{CLKOUTEN}}$ bit in Configuration Words.

The internal oscillator block has two independent oscillators that can produce two internal system clock sources.

1. The HFINTOSC (High-Frequency Internal Oscillator) is factory calibrated and operates up to 32 MHz.
2. The LFINTOSC (Low-Frequency Internal Oscillator) is factory calibrated and operates at 31 kHz.

7.2.2.1 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a precision digitally-controlled internal clock source that produces a stable clock up to 32 MHz. The HFINTOSC can be enabled through one of the following methods:

- Programming the RSTOSC<2:0> bits in Configuration Word 1 to '`110`' (1 MHz) or '`000`' (32 MHz) to set the oscillator upon device Power-up or Reset
- Write to the NOSC<2:0> bits of the OSCCON1 register during run-time

The HFINTOSC frequency can be selected by setting the HFFRQ<2:0> bits of the OSCFRQ register.

The NDIV<3:0> bits of the OSCCON1 register allow for division of the output of the selected clock source by a range between 1:1 and 1:512.

7.2.2.2 2x PLL

The oscillator module contains a PLL that can be used with the HFINTOSC clock source to provide a system clock source. The input frequency to the PLL is limited to 8, 12, or 16 MHz, which will yield a system clock source of 16, 24, or 32 MHz, respectively.

The PLL may be enabled for use by one of two methods:

1. Program the RSTOSC bits in the Configuration Word 1 to '`000`' to enable the HFINTOSC (32 MHz). This setting configures the HFFRQ<2:0> bits to '`110`' (16 MHz) and activates the 2x PLL.
2. Write '`000`' the NOSC<2:0> bits in the OSCCON1 register to enable the 2x PLL, and write the correct value into the HFFRQ<3:0> bits of the OSCFRQ register to select the desired system clock frequency. See Register 6-6 for more information.

REGISTER 8-11: PIR4: PERIPHERAL INTERRUPT REQUEST REGISTER 4

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	HS = Hardware set

bit 7 **CWG2IF:** CWG 2 Interrupt Flag bit
1 = CWG2 has gone into shutdown
0 = CWG2 is operating normally, or interrupt cleared

bit 6 **CWG1IF:** CWG1 Interrupt Flag bit
1 = CWG1 has gone into shutdown
0 = CWG1 is operating normally, or interrupt cleared

bit 5 **TMR5GIF:** Timer5 Gate Interrupt Flag bit
1 = The TMR5 gate has gone inactive (the gate is closed).
0 = The TMR5 gate has not gone inactive.

bit 4 **TMR5IF:** Timer5 Overflow Interrupt Flag bit
1 = TMR5 overflow occurred (must be cleared in software)
0 = No TMR5 overflow occurred

bit 3 **CCP4IF:** CCP4 Interrupt Flag bit

Value	CCPM Mode		
	Capture	Compare	PWM
1	Capture occurred (must be cleared in software)	Compare match occurred (must be cleared in software)	Output trailing edge occurred (must be cleared in software)
0	Capture did not occur	Compare match did not occur	Output trailing edge did not occur

bit 2 **CCP3IF:** CCP3 Interrupt Flag bit

Value	CCPM Mode		
	Capture	Compare	PWM
1	Capture occurred (must be cleared in software)	Compare match occurred (must be cleared in software)	Output trailing edge occurred (must be cleared in software)
0	Capture did not occur	Compare match did not occur	Output trailing edge did not occur

9.0 POWER-SAVING OPERATION MODES

The purpose of the Power-Down modes is to reduce power consumption. There are three Power-Down modes: DOZE mode, IDLE mode, and Sleep mode.

9.1 DOZE Mode

Doze mode allows for power savings by reducing CPU operation and program memory access, without affecting peripheral operation. Doze mode differs from Sleep mode because the system oscillators continue to operate, while only the CPU and program memory are affected. The reduced execution saves power by eliminating unnecessary operations within the CPU and memory.

When the Doze Enable (DOZEN) bit is set (DOZEN = `1`), the CPU executes only one instruction cycle out of every N cycles as defined by the DOZE<2:0> bits of the CPUDOZE register. For example, if DOZE<2:0> = `100`, the instruction cycle ratio is 1:32. The CPU and memory execute for one instruction cycle and then lay idle for 31 instruction cycles. During the unused cycles, the peripherals continue to operate at the system clock speed.

9.1.1 DOZE OPERATION

The Doze operation is illustrated in Figure 9-1. For this example:

- Doze enable (DOZEN) bit set (DOZEN = `1`)
- DOZE<2:0> = `001` (1:4) ratio
- Recover-on-Interrupt (ROI) bit set (ROI = `1`)

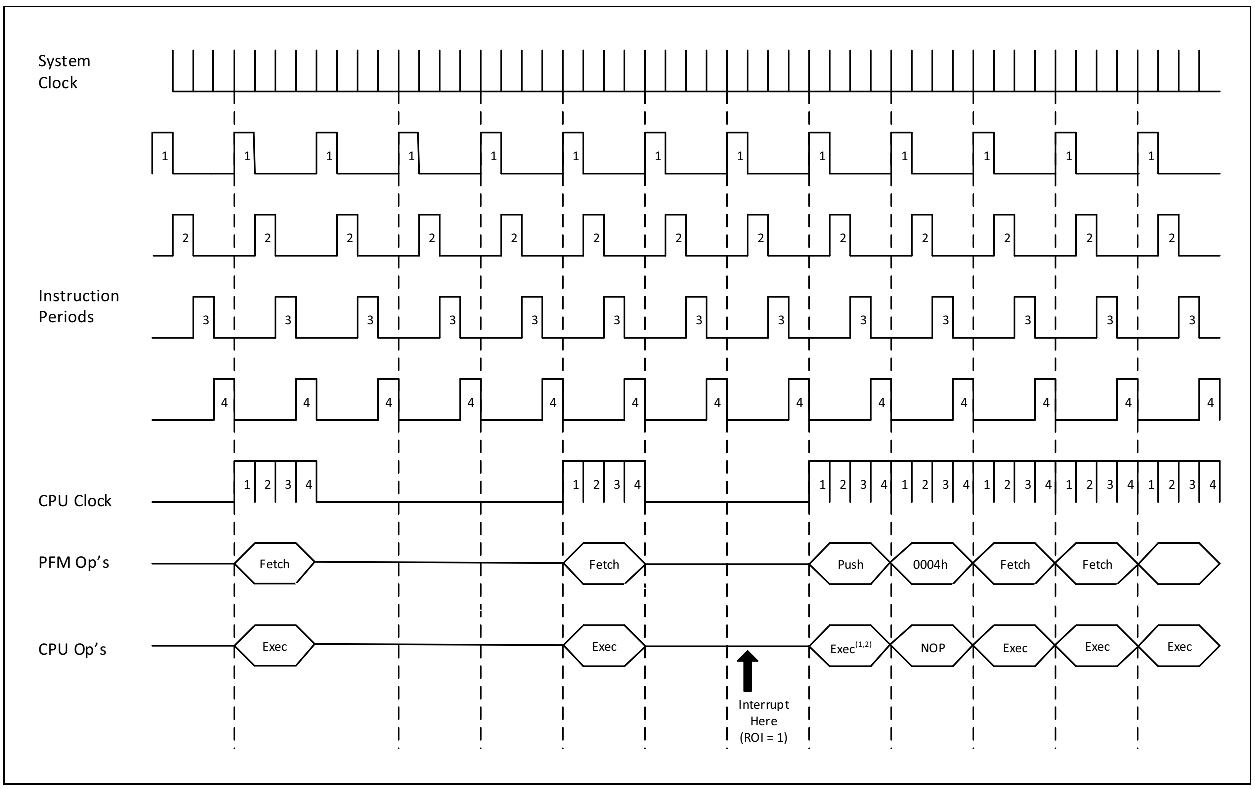
As with normal operation, the program memory fetches for the next instruction cycle. The instruction clocks to the peripherals continue throughout.

9.1.2 INTERRUPTS DURING DOZE

If an interrupt occurs and the Recover-on-Interrupt (ROI) bit is clear (ROI = `0`) at the time of the interrupt, the Interrupt Service Routine (ISR) continues to execute at the rate selected by DOZE<2:0>. Interrupt latency is extended by the DOZE<2:0> ratio.

If an interrupt occurs and the ROI bit is set (ROI = `1`) at the time of the interrupt, the DOZEN bit is cleared and the CPU executes at full speed. The prefetched instruction is executed and then the interrupt vector sequence is executed. In Figure 9-1, the interrupt occurs during the 2nd instruction cycle of the Doze period, and immediately brings the CPU out of Doze. If the Doze-on-Exit (DOE) bit is set (DOE = `1`) when the RETFIE operation is executed, DOZEN is set, and the CPU executes at the reduced rate based on the DOZE<2:0> ratio.

FIGURE 9-1: DOZE MODE OPERATION EXAMPLE



11.4.3 NVMREG WRITE TO EEPROM

Writing to the EEPROM is accomplished by the following steps:

1. Set the NVMREGS and WREN bits of the NVMCON1 register.
2. Write the desired address (address +7000h) into the NVMADRH:NVMADRL register pair (Table 11-2).
3. Perform the unlock sequence as described in **Section 11.4.2 "NVM Unlock Sequence"**.

A single EEPROM byte is written with NVMDATA. The operation includes an implicit erase cycle for that byte (it is not necessary to set the FREE bit), and requires many instruction cycles to finish. CPU execution continues in parallel and, when complete, WR is cleared by hardware, NVMIF is set, and an interrupt will occur if NVMIE is also set. Software must poll the WR bit to determine when writing is complete, or wait for the interrupt to occur. WREN will remain unchanged.

Once the EEPROM write operation begins, clearing the WR bit will have no effect; the operation will run to completion.

11.4.4 NVMREG ERASE OF PROGRAM FLASH MEMORY

Program Flash memory can only be erased one row at a time. No automatic erase occurs upon the initiation of the write to program Flash memory.

To erase a program Flash memory row:

1. Clear the NVMREGS bit of the NVMCON1 register to erase program Flash memory locations, or set the NVMREGS bit to erase user ID locations.
2. Write the desired address into the NVMADRH:NVMADRL register pair (Table 11-2).
3. Set the FREE and WREN bits of the NVMCON1 register.
4. Perform the unlock sequence as described in **Section 11.4.2 "NVM Unlock Sequence"**.

If the program Flash memory address is write-protected, the WR bit will be cleared and the erase operation will not take place.

While erasing program Flash memory, CPU operation is suspended, and resumes when the operation is complete. Upon completion, the NVMIF is set, and an interrupt will occur if the NVMIE bit is also set.

Write latch data is not affected by erase operations, and WREN will remain unchanged.

FIGURE 11-3: NVM ERASE FLOWCHART

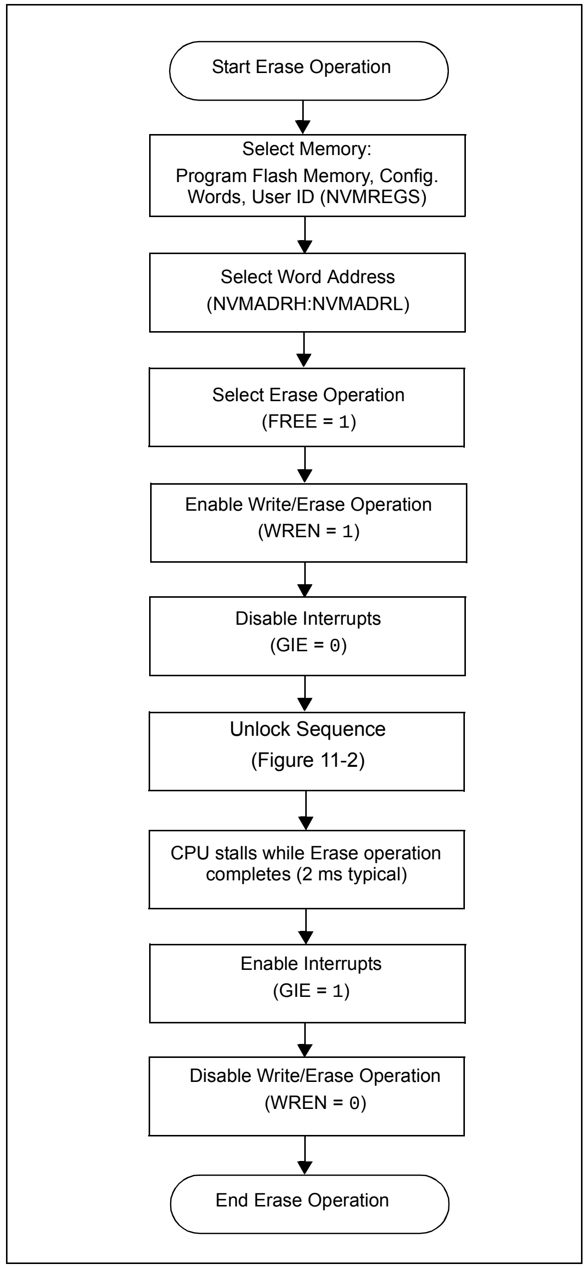


TABLE 19-3: SUMMARY OF REGISTERS ASSOCIATED WITH PWMx

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	143
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	144
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	149
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	150
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	156
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	157
PWM5CON	PWM5EN	—	PWM5OUT	PWM5POL	—	—	—	—	196
PWM5DCH	PWM5DC<9:2>								196
PWM5DCL	PWM5DC<1:0>		—	—	—	—	—	—	196
PWM6CON	PWM6EN	—	PWM6OUT	PWM6POL	—	—	—	—	196
PWM6DCH	PWM6DC<9:2>								196
PWM6DCL	PWM6DC<1:0>		—	—	—	—	—	—	196
PWMTMRS	—	—	—	—	P6TSEL<1:0>		P5TSEL<1:0>		197
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	101
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	108
PIR2	TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BCL2IF	TMR4IF	NCO1IF	109
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	103
PIE2	TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BCL2IE	TMR4IE	NCO1IE	104
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		298
T4CON	—	T4OUTPS<3:0>				TMR4ON	T4CKPS<1:0>		292
T6CON	—	T6OUTPS<3:0>				TMR6ON	T6CKPS<1:0>		292
TMR2	TMR2<7:0>								299
TMR4	TMR4<7:0>								299
TMR6	TMR6<7:0>								299
PR2	PR2<7:0>								299
PR4	PR4<7:0>								299
PR6	PR6<7:0>								299
CWGxDAT	—	—	—	—	DAT<3:0>				215
CLCxSELy	—	—	LCxDyS<5:0>						229
MDSRC	—	—	—	—	MDMS<3:0>				272
MDCARH	—	MDCHPOL	MDCHSYNC	—	MDCH<3:0>				273
MDCARL	—	MDCLPOL	MDCLSYNC	—	MDCL<3:0>				274

Legend: - = Unimplemented locations, read as ‘0’. Shaded cells are not used by the PWM module.
Note 1: PIC16(L)F18345 only.
2: Unimplemented, read as ‘1’.

22.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- Port configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Result formatting

22.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. Refer to **Section 12.0 “I/O Ports”** for more information.

Note:	Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current.

22.1.2 CHANNEL SELECTION

There are several channel selections available:

- Five PORTA pins (RA0-RA2, RA4-RA5)
- Four PORTB pins (RB4-RB7, PIC16(L)F18345 only)
- Six PORTC pins (RC0-RC5, PIC16(L)F18325)
- Eight PORTC pins (RC0-RC7, PIC16(L)F18345 only)
- Temperature Indicator
- DAC output
- Fixed Voltage Reference (FVR)
- VSS (ground)

The CHS<5:0> bits of the ADCON0 register (Register 22-1) determine which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion. Refer to **Section 22.2 “ADC Operation”** for more information.

Note:	It is recommended that when switching from an ADC channel of a higher voltage to a channel of a lower voltage, the software selects the VSS channel before switching to the channel of the lower voltage. If the ADC does not have a dedicated VSS input channel, the VSS selection (DAC1R<4:0> = `b'00000'`) through the DAC output channel can be used. If the DAC is in use, a free input channel can be connected to VSS, and can be used in place of the DAC.

22.1.3 ADC VOLTAGE REFERENCE

The ADPREF<1:0> bits of the ADCON1 register provides control of the positive voltage reference. The positive voltage reference can be:

- VREF+ pin
- VDD
- FVR 2.048V
- FVR 4.096V (Not available on LF devices)

The ADNREF bit of the ADCON1 register provides control of the negative voltage reference. The negative voltage reference can be:

- VREF- pin
- VSS

See **Section 16.0 “Fixed Voltage Reference (FVR)”** for more details on the Fixed Voltage Reference.

22.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS<2:0> bits of the ADCON1 register. There are seven possible clock options:

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- ADCRC (dedicated RC oscillator)

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 12 TAD periods as shown in Figure 22-2.

For correct conversion, the appropriate TAD specification must be met. Refer to Table 35-13 for more information. Table 22-1 gives examples of appropriate ADC clock selections.

Note:	Unless using the ADCRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.

FIGURE 27-4: TIMER1 GATE TOGGLE MODE

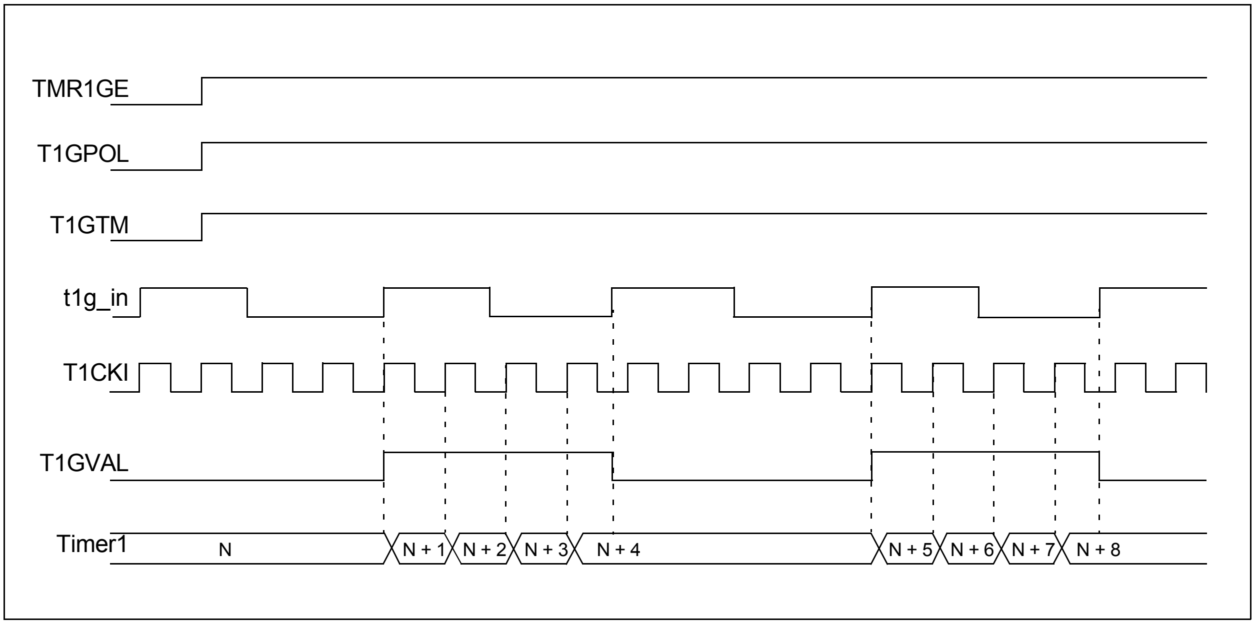


FIGURE 27-5: TIMER1 GATE SINGLE-PULSE MODE

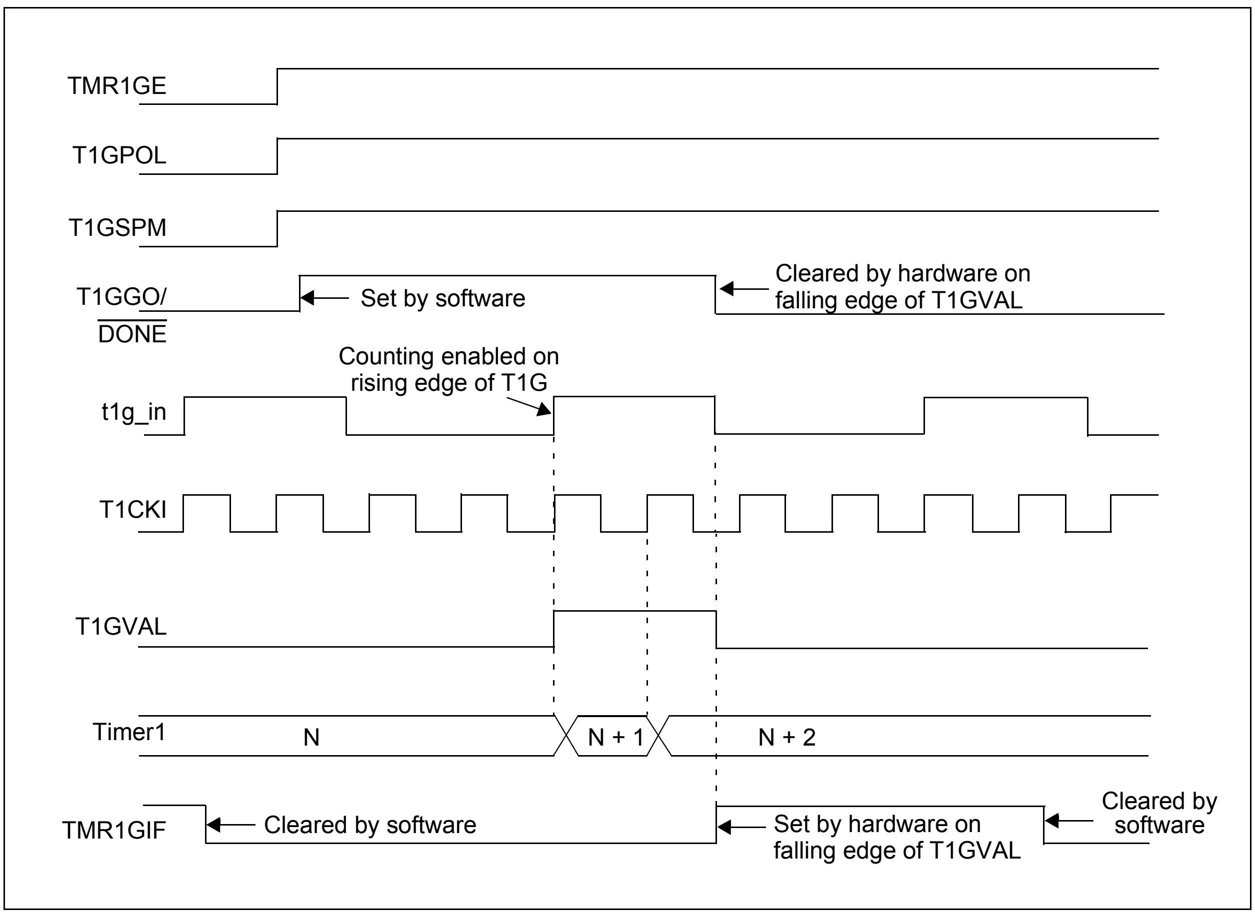


TABLE 27-5: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1/3/5

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	143
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	144
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	149
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	150
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	156
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	157
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	101
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	108
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	103
PIR3	OSFIF	CSWIF	TMR3GIF	TMR3IF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	110
PIE3	OSFIE	CSWIE	TMR3GIE	TMR3IE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	105
PIR4	CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF	111
PIE4	CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE	106
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1SOSC	$\overline{\text{T1SYNC}}$	—	TMR1ON	292
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>		293
TMR1L	TMR1L<7:0>								294
TMR1H	TMR1H<7:0>								294
T1CKIPPS	—	—	—	T1CKIPPS<4:0>					162
T1GPPS	—	—	—	T1GPPS<4:0>					162
T3CON	TMR3CS<1:0>		T3CKPS<1:0>		T3SOSC	$\overline{\text{T3SYNC}}$	—	TMR3ON	292
T3GCON	TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/$\overline{\text{DONE}}$	T3GVAL	T3GSS<1:0>		293
TMR3L	TMR3L<7:0>								294
TMR3H	TMR3H<7:0>								294
T3CKIPPS	—	—	—	T3CKIPPS<4:0>					162
T3GPPS	—	—	—	T3GPPS<4:0>					162
T5CON	TMR5CS<1:0>		T5CKPS<1:0>		T5SOSC	$\overline{\text{T5SYNC}}$	—	TMR5ON	292
T5GCON	TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/$\overline{\text{DONE}}$	T5GVAL	T5GSS<1:0>		293
TMR5L	TMR5L<7:0>								294
TMR5H	TMR5H<7:0>								294
T5CKIPPS	—	—	—	T5CKIPPS<4:0>					162
T5GPPS	—	—	—	T5GPPS<4:0>					162
T0CON0	T0EN	—	T0OUT	T016BIT	T0OUTPS<3:0>				280
CMxCON0	CxON	CxOUT	—	CxPOL	—	CxSP	CxHYS	CxSYNC	190
CCPTMRS	C4TSEL<1:0>		C3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>		311
CCPxCON	CCPxEN	—	CCPxOUT	CCPxFMT	CCPxMODE<3:0>				308
CLCxSELy	—	—	LCxDyS<5:0>						229
ADACT	—	—	—	ADACT<4:0>					246

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the Timer1 module.
Note 1: PIC16(L)F18345 only.
2: Unimplemented, read as '1'.

30.5.3.3 7-bit Transmission with Address Hold Enabled

Setting the AHEN bit of the SSPxCON3 register enables additional clock stretching and interrupt generation after the eighth falling edge of a received matching address. Once a matching address has been clocked in, CKP is cleared and the SSPxIF interrupt is set.

Figure 30-19 displays a standard waveform of a 7-bit address slave transmission with AHEN enabled.

1. Bus starts Idle.
2. Master sends Start condition; the S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Master sends matching address with R/$\overline{W}$ bit set. After the eighth falling edge of the SCL line the CKP bit is cleared and SSPxIF interrupt is generated.
4. Slave software clears SSPxIF.
5. Slave software reads ACKTIM bit of SSPxCON3 register, and R/$\overline{W}$ and D/$\overline{A}$ of the SSPxSTAT register to determine the source of the interrupt.
6. Slave reads the address value from the SSPxBUF register clearing the BF bit.
7. Slave software decides from this information if it wishes to $\overline{ACK}$ or not $\overline{ACK}$ and sets the ACKDT bit of the SSPxCON2 register accordingly.
8. Slave sets the CKP bit releasing SCL.
9. Master clocks in the $\overline{ACK}$ value from the slave.
10. Slave hardware automatically clears the CKP bit and sets SSPxIF after the $\overline{ACK}$ if the R/$\overline{W}$ bit is set.
11. Slave software clears SSPxIF.
12. Slave loads value to transmit to the master into SSPxBUF setting the BF bit.

Note:	SSPxBUF cannot be loaded until after the $\overline{ACK}$.

13. Slave sets the CKP bit releasing the clock.
14. Master clocks out the data from the slave and sends an $\overline{ACK}$ value on the ninth SCL pulse.
15. Slave hardware copies the $\overline{ACK}$ value into the ACKSTAT bit of the SSPxCON2 register.
16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
17. If the master sends a not $\overline{ACK}$ the slave releases the bus allowing the master to send a Stop and end the communication.

Note:	Master must send a not $\overline{ACK}$ on the last byte to ensure that the slave releases the SCL line to receive a Stop.

30.5.4 SLAVE MODE 10-BIT ADDRESS RECEPTION

This section describes a standard sequence of events for the MSSPx module configured as an I^2C slave in 10-bit Addressing mode.

Figure 30-20 is used as a visual reference for this description.

This is a step-by-step process of what must be done by slave software to accomplish I^2C communication.

1. Bus starts Idle.
2. Master sends Start condition; S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Master sends matching high address with R/$\overline{\text{W}}$ bit clear; UA bit of the SSPxSTAT register is set.
4. Slave sends $\overline{\text{ACK}}$ and SSPxIF is set.
5. Software clears the SSPxIF bit.
6. Software reads received address from SSPxBUF clearing the BF flag.
7. Slave loads low address into SSPxADD, releasing SCL.
8. Master sends matching low address byte to the slave; UA bit is set.

Note:	Updates to the SSPxADD register are not allowed until after the $\overline{\text{ACK}}$ sequence.

9. Slave sends $\overline{\text{ACK}}$ and SSPxIF is set.

Note:	If the low address does not match, SSPxIF and UA are still set so that the slave software can set SSPxADD back to the high address. BF is not set because there is no match. CKP is unaffected.

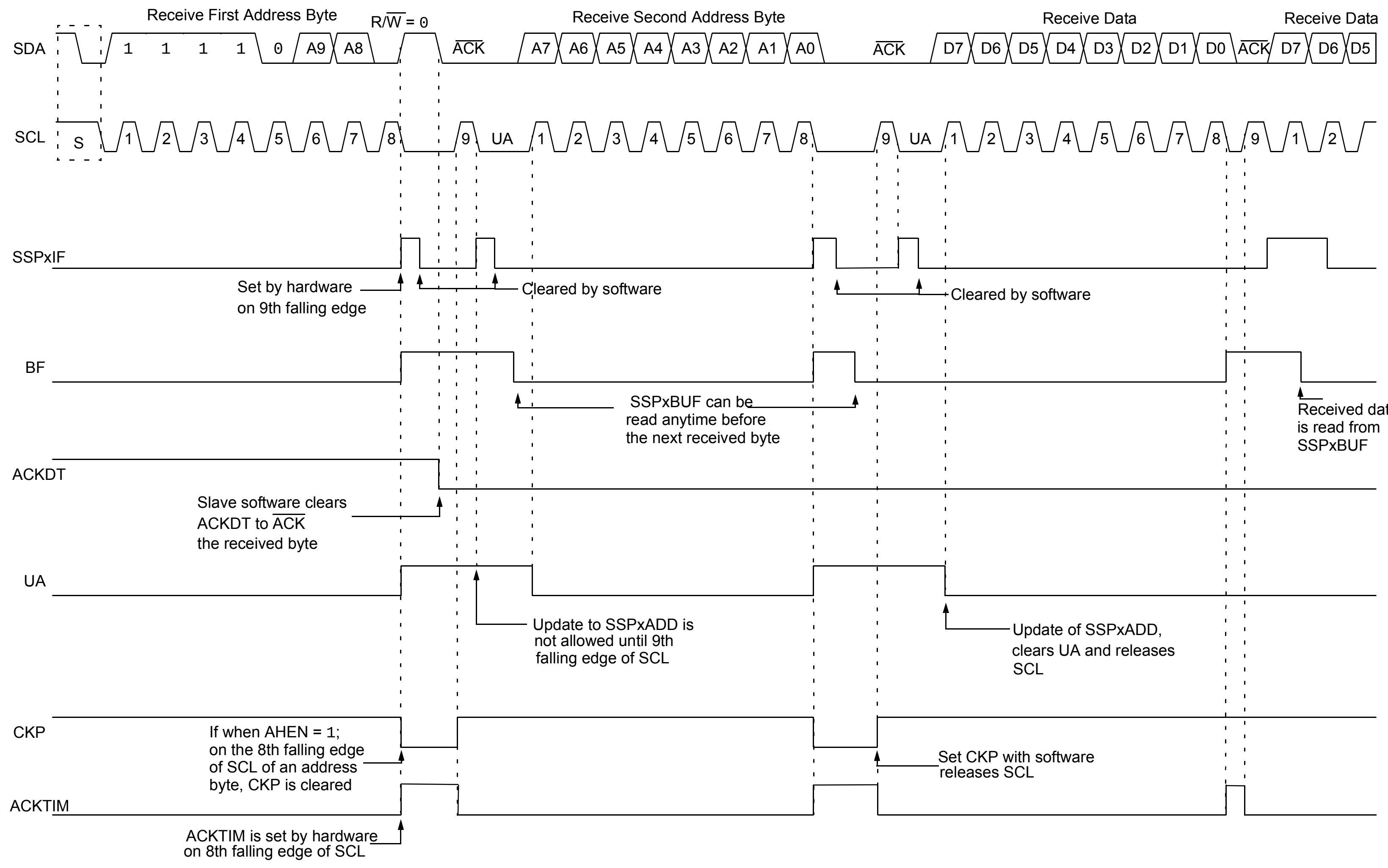
10. Slave clears SSPxIF.
11. Slave reads the received matching address from SSPxBUF clearing BF.
12. Slave loads high address into SSPxADD.
13. Master clocks a data byte to the slave and clocks out the slaves $\overline{\text{ACK}}$ on the ninth SCL pulse; SSPxIF is set.
14. If SEN bit of SSPxCON2 is set, CKP is cleared by hardware and the clock is stretched.
15. Slave clears SSPxIF.
16. Slave reads the received byte from SSPxBUF clearing BF.
17. If SEN is set the slave sets CKP to release the SCL.
18. Steps 13-17 repeat for each received byte.
19. Master sends Stop to end the transmission.

30.5.5 10-BIT ADDRESSING WITH ADDRESS OR DATA HOLD

Reception using 10-bit addressing with AHEN or DHEN set is the same as with 7-bit modes. The only difference is the need to update the SSPxADD register using the UA bit. All functionality, specifically when the CKP bit is cleared and SCL line is held low are the same. Figure 30-21 can be used as a reference of a slave in 10-bit addressing with AHEN set.

Figure 30-22 shows a standard waveform for a slave transmitter in 10-bit Addressing mode.

FIGURE 30-21: I²C SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 1, DHEN = 0)



Preliminary

FIGURE 30-22: I²C SLAVE, 10-BIT ADDRESS, TRANSMISSION (SEN = 0, AHEN = 0, DHEN = 0)

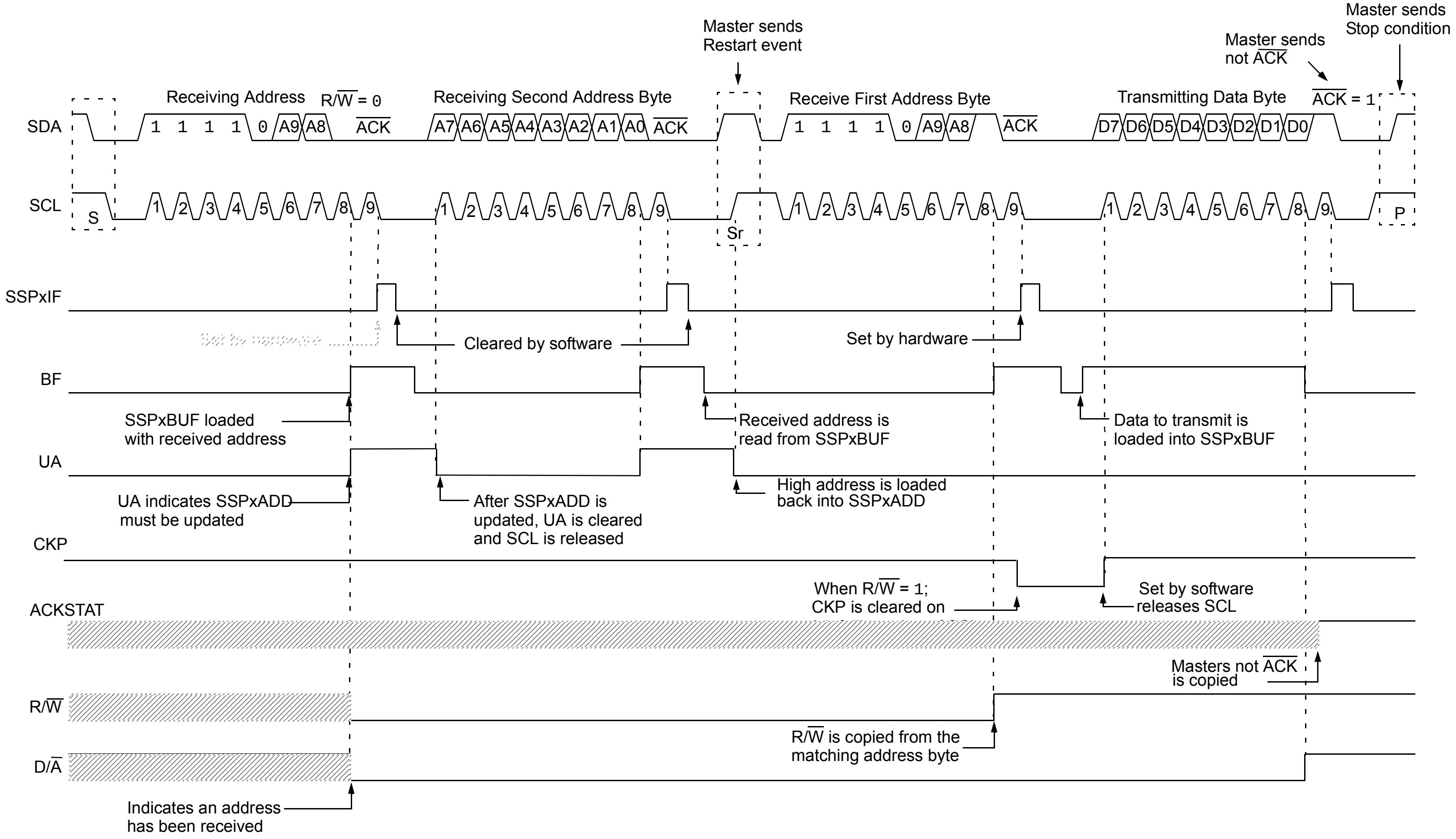
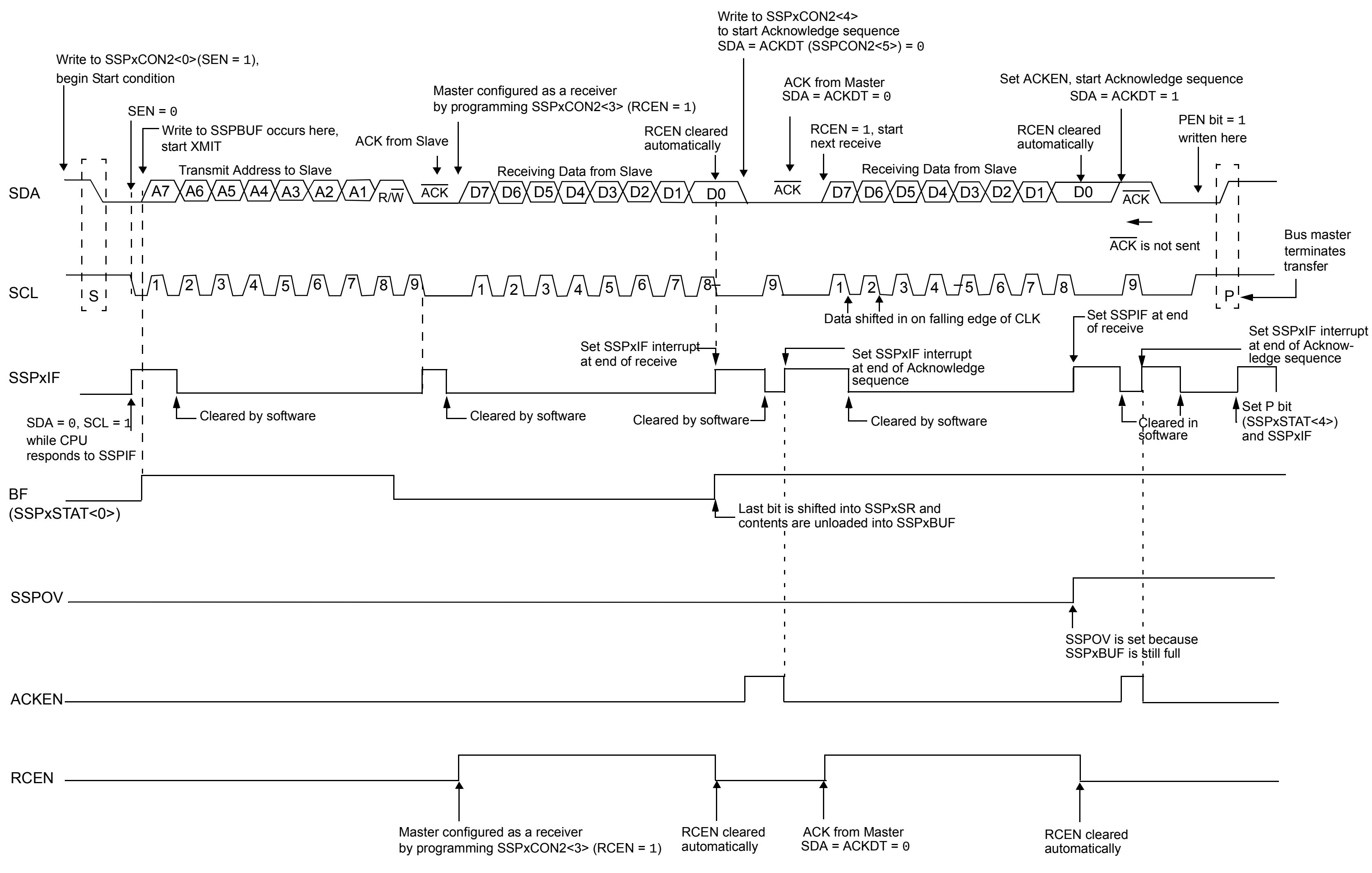


FIGURE 30-29: I²C MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)



Preliminary

30.7 Baud Rate Generator

The MSSPx module has a Baud Rate Generator available for clock generation in both I²C and SPI Master modes. The Baud Rate Generator (BRG) reload value is placed in the SSPxADD register (Register 30-6). When a write occurs to SSPxBUF, the Baud Rate Generator will automatically begin counting down.

Once the given operation is complete, the internal clock will automatically stop counting and the clock pin will remain in its last state.

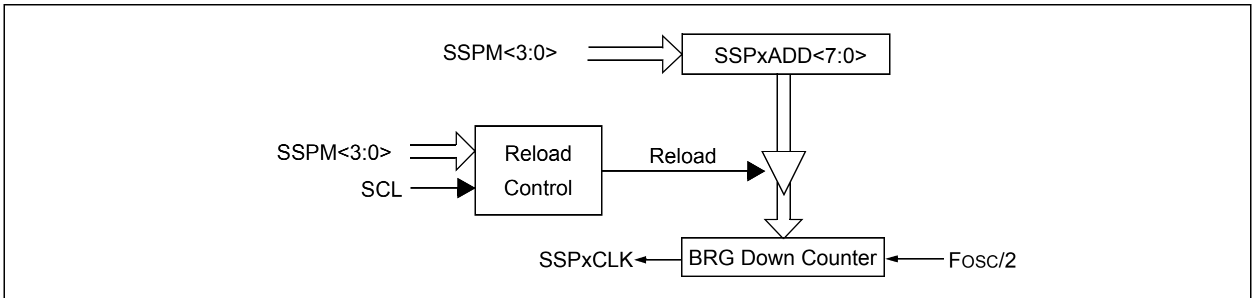
An internal signal "Reload" in Figure 30-40 triggers the value from SSPxADD to be loaded into the BRG counter. This occurs twice for each oscillation of the module clock line. The logic dictating when the reload signal is asserted depends on the mode the MSSPx is being operated in.

Table 30-4 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPADD.

EQUATION 30-1:

$$F_{CLOCK} = \frac{F_{OSC}}{(SSPxADD + 1)(4)}$$

FIGURE 30-40: BAUD RATE GENERATOR BLOCK DIAGRAM



Note:	Values of 0x00, 0x01 and 0x02 are not valid for SSPxADD when used as a Baud Rate Generator for I²C. This is an implementation limitation.

REGISTER 30-4: SSPxCON3: SSP CONTROL REGISTER 3

R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ACKTIM(3)	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7 **ACKTIM:** Acknowledge Time Status bit (I^2C mode only)(3)
1 = Indicates the I^2C bus is in an Acknowledge sequence, set on eighth falling edge of SCL clock
0 = Not an Acknowledge sequence, cleared on ninth rising edge of SCL clock

bit 6 **PCIE**: Stop Condition Interrupt Enable bit (I^2C mode only)
1 = Enable interrupt on detection of Stop condition
0 = Stop detection interrupts are disabled(2)

bit 5 **SCIE**: Start Condition Interrupt Enable bit (I^2C mode only)
1 = Enable interrupt on detection of Start or Restart conditions
0 = Start detection interrupts are disabled(2)

bit 4 **BOEN:** Buffer Overwrite Enable bit
In SPI Slave mode:(1)
1 = SSPBUF updates every time that a new data byte is shifted in ignoring the BF bit
0 = If new byte is received with BF bit of the SSPSTAT register already set, SSPOV bit of the SSPCON1 register is set, and the buffer is not updated
In I^2C Master mode and SPI Master mode:
This bit is ignored.
In I^2C Slave mode:
1 = SSPBUF is updated and $\overline{ACK}$ is generated for a received address/data byte, ignoring the state of the SSPOV bit only if the BF bit = 0.
0 = SSPBUF is only updated when SSPOV is clear

bit 3 **SDAHT:** SDA Hold Time Selection bit (I^2C mode only)
1 = Minimum of 300 ns hold time on SDA after the falling edge of SCL
0 = Minimum of 100 ns hold time on SDA after the falling edge of SCL

bit 2 **SBCDE:** Slave Mode Bus Collision Detect Enable bit (I^2C Slave mode only)
If, on the rising edge of SCL, SDA is sampled low when the module is outputting a high state, the BCL1IF bit of the PIR1 register is set, and bus goes idle
1 = Enable slave bus collision interrupts
0 = Slave bus collision interrupts are disabled

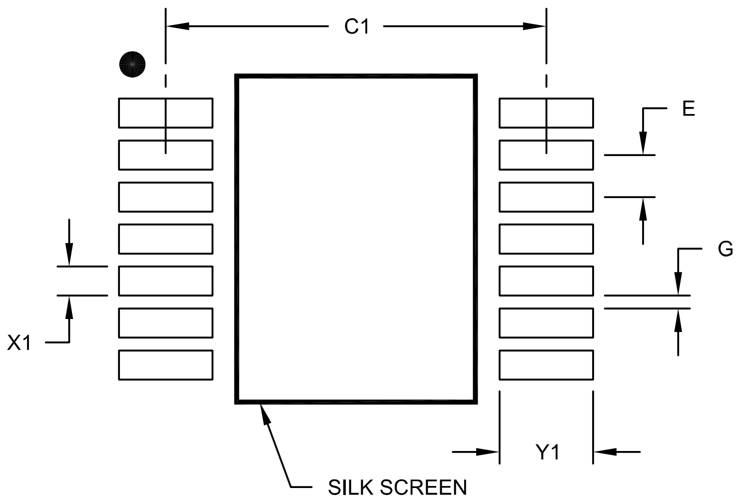
bit 1 **AHEN:** Address Hold Enable bit (I^2C Slave mode only)
1 = Following the eighth falling edge of SCL for a matching received address byte; CKP bit of the SSPCON1 register will be cleared and the SCL will be held low.
0 = Address holding is disabled

bit 0 **DHEN:** Data Hold Enable bit (I^2C Slave mode only)
1 = Following the eighth falling edge of SCL for a received data byte; slave hardware clears the CKP bit of the SSPCON1 register and SCL is held low.
0 = Data holding is disabled

Note 1: For daisy-chained SPI operation; allows the user to ignore all but the last received byte. SSPOV is still set when a new byte is received and BF = 1, but hardware continues to write the most recent byte to SSPBUF.
2: This bit has no effect in Slave modes that Start and Stop condition detection is explicitly listed as enabled.
3: The ACKTIM Status bit is only active when the AHEN bit or DHEN bit is set.

14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C1		5.90	
Contact Pad Width (X14)	X1			0.45
Contact Pad Length (X14)	Y1			1.45
Distance Between Pads	G	0.20		

Notes:

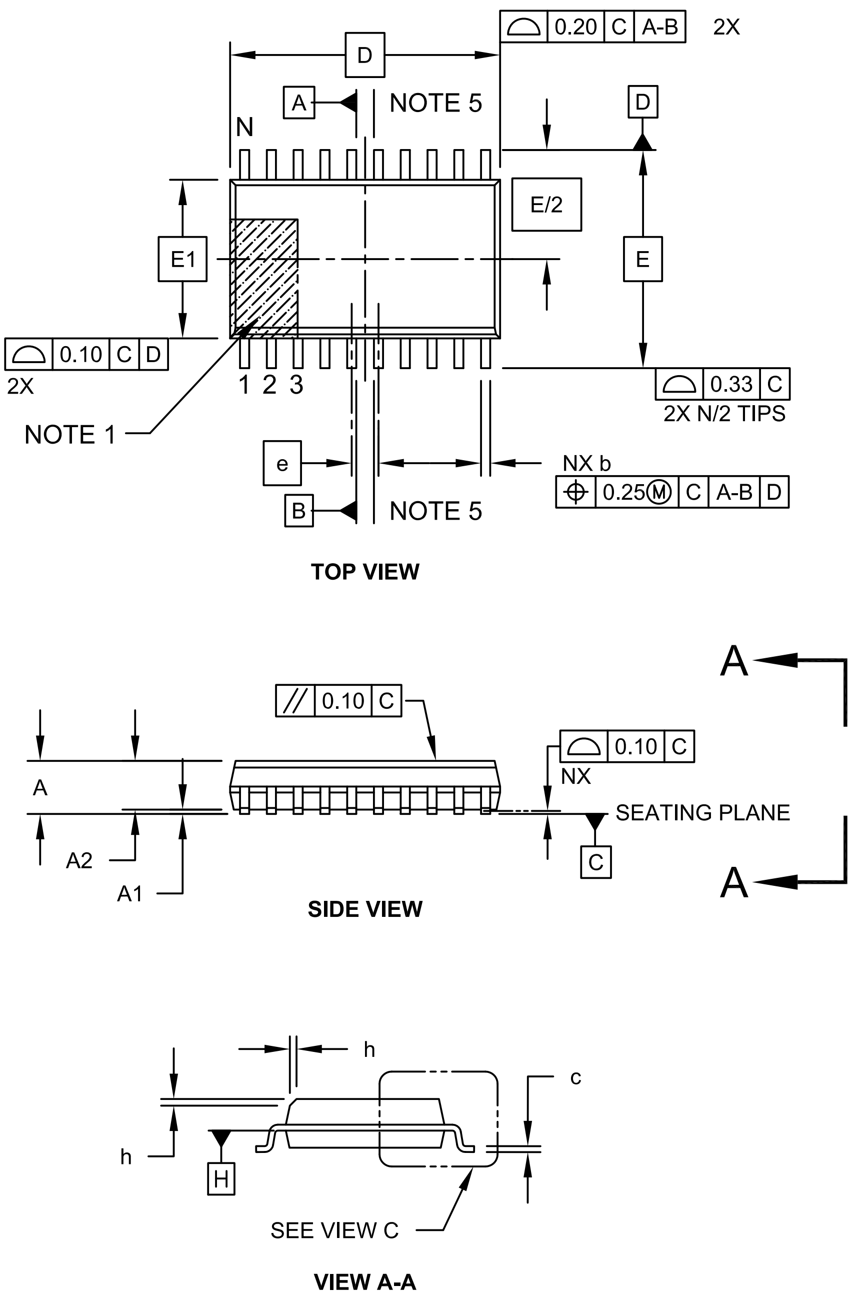
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2087A

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

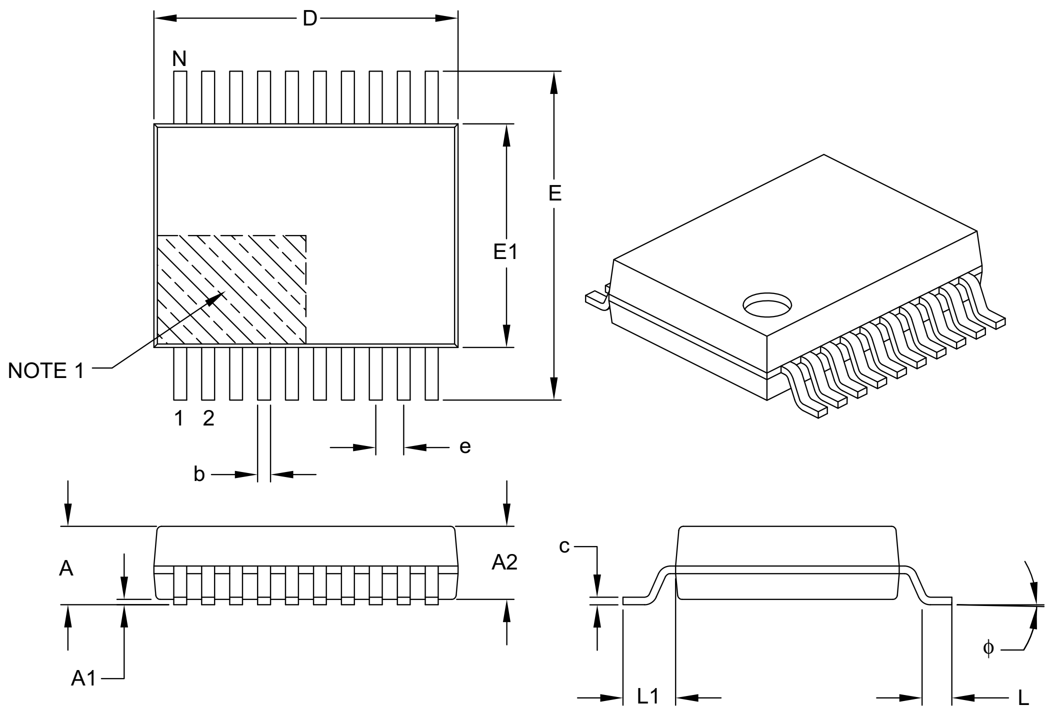
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-094C Sheet 1 of 2

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

