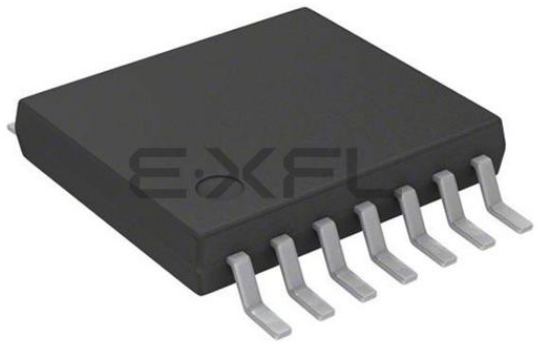


Welcome to E-XFL.COM

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	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	12
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 11x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	14-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	14-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f15323-e-st

Table of Contents

TABLE 4-5: PIC16(L)F15313/23 MEMORY MAP, BANKS 8-15

	BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	Core Register (Table 4-3)	480h	Core Register (Table 4-3)	500h	Core Register (Table 4-3)	580h	Core Register (Table 4-3)	600h	Core Register (Table 4-3)	680h	Core Register (Table 4-3)	700h	Core Register (Table 4-3)	780h	Core Register (Table 4-3)	
40Bh		48Bh		50Bh		58Bh		60Bh		68Bh		70Bh		78Bh		
40Ch	—	48Ch	—	50Ch	—	58Ch	NCO1ACCL	60Ch	CWG1CLK	68Ch	—	70Ch	PIR0	78Ch	—	
40Dh	—	48Dh	—	50Dh	—	58Dh	NCO1ACCH	60Dh	CWG1DAT	68Dh	—	70Dh	PIR1	78Dh	—	
40Eh	—	48Eh	—	50Eh	—	58Eh	NCO1ACCU	60Eh	CWG1DBR	68Eh	—	70Eh	PIR2	78Eh	—	
40Fh	—	48Fh	—	50Fh	—	58Fh	NCO1INCL	60Fh	CWG1DBF	68Fh	—	70Fh	PIR3	78Fh	—	
410h	—	490h	—	510h	—	590h	NCO1INCH	610h	CWG1CON0	690h	—	710h	PIR4	790h	—	
411h	—	491h	—	511h	—	591h	NCO1INCU	611h	CWG1CON1	691h	—	711h	PIR5	791h	—	
412h	—	492h	—	512h	—	592h	NCO1CON	612h	CWG1AS0	692h	—	712h	PIR6	792h	—	
413h	—	493h	—	513h	—	593h	NCO1CLK	613h	CWG1AS1	693h	—	713h	PIR7	793h	—	
414h	—	494h	—	514h	—	594h	—	614h	CWG1STR	694h	—	714h	—	794h	—	
415h	—	495h	—	515h	—	595h	—	615h	—	695h	—	715h	—	795h	—	
416h	—	496h	—	516h	—	596h	—	616h	—	696h	—	716h	PIE0	796h	PMD0	
417h	—	497h	—	517h	—	597h	—	617h	—	697h	—	717h	PIE1	797h	PMD1	
418h	—	498h	—	518h	—	598h	—	618h	—	698h	—	718h	PIE2	798h	PMD2	
419h	—	499h	—	519h	—	599h	—	619h	—	699h	—	719h	PIE3	799h	PMD3	
41Ah	—	49Ah	—	51Ah	—	59Ah	—	61Ah	—	69Ah	—	71Ah	PIE4	79Ah	PMD4	
41Bh	—	49Bh	—	51Bh	—	59Bh	—	61Bh	—	69Bh	—	71Bh	PIE5	79Bh	PMD5	
41Ch	—	49Ch	—	51Ch	—	59Ch	TMR0	61Ch	—	69Ch	—	71Ch	PIE6	79Ch	—	
41Dh	—	49Dh	—	51Dh	—	59Dh	PR0	61Dh	—	69Dh	—	71Dh	PIE7	79Dh	—	
41Eh	—	49Eh	—	51Eh	—	59Eh	T0CON0	61Eh	—	69Eh	—	71Eh	—	79Eh	—	
41Fh	—	49Fh	—	51Fh	—	59Fh	T0CON1	61Fh	—	69Fh	—	71Fh	—	79Fh	—	
420h	Unimplemented Read as '0'	4A0h	Unimplemented Read as '0'	520h	Unimplemented Read as '0'	5A0h	Unimplemented Read as '0'	620h	Unimplemented Read as '0'	6A0h	Unimplemented Read as '0'	720h	Unimplemented Read as '0'	7A0h	Unimplemented Read as '0'	
46Fh		4EFh		56Fh		5EFh		66Fh		6EFh		76Fh		7EFh		
470h	Common RAM Accesses 70h-7Fh	4F0h	Common RAM Accesses 70h-7Fh	570h	Common RAM Accesses 70h-7Fh	5F0h	Common RAM Accesses 70h-7Fh	670h	Common RAM Accesses 70h-7Fh	6F0h	Common RAM Accesses 70h-7Fh	770h	Common RAM Accesses 70h-7Fh	7F0h	Common RAM Accesses 70h-7Fh	
47Fh		4FFh		57Fh		5FFh		67Fh		6FFh		77Fh		7FFh		

Note 1: Unimplemented locations read as '0'.

TABLE 4-10: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on: MCLR
Bank 3											
CPU CORE REGISTERS; see Table 4-3 for specifics											
18Ch	SSP1BUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	xxxx xxxx
18Dh	SSP1ADD	ADD<7:0>								0000 0000	0000 0000
18Eh	SSP1MSK	MSK<7:0>								1111 1111	1111 1111
18Fh	SSP1STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	0000 0000
190h	SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
191h	SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000
192h	SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000 0000	0000 0000
193h — 19Fh	—	Unimplemented								—	—

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

4.5 Stack

All devices have a 16-level x 15-bit wide hardware stack (refer to Figure 4-4 through Figure 4-7). The stack space is not part of either program or data space. The PC is PUSHed onto the stack when CALL or CALLW instructions are executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer if the STVREN bit is programmed to '0' (Configuration Words). This means that after the stack has been PUSHed sixteen times, the seventeenth PUSH overwrites the value that was stored from the first PUSH. The eighteenth PUSH overwrites the second PUSH (and so on). The STKOVF and STKUNF flag bits will be set on an Overflow/Underflow, regardless of whether the Reset is enabled.

Note 1: There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, CALLW, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address.

4.5.1 ACCESSING THE STACK

The stack is accessible through the TOSH, TOSL and STKPTR registers. STKPTR is the current value of the Stack Pointer. TOSH:TOSL register pair points to the TOP of the stack. Both registers are read/writable. TOS is split into TOSH and TOSL due to the 15-bit size of the PC. To access the stack, adjust the value of STKPTR, which will position TOSH:TOSL, then read/write to TOSH:TOSL. STKPTR is five bits to allow detection of overflow and underflow.

Note: Care should be taken when modifying the STKPTR while interrupts are enabled.

During normal program operation, CALL, CALLW and interrupts will increment STKPTR while RETLW, RETURN, and RETFIE will decrement STKPTR. STKPTR can be monitored to obtain to value of stack memory left at any given time. The STKPTR always points at the currently used place on the stack. Therefore, a CALL or CALLW will increment the STKPTR and then write the PC, and a return will unload the PC value from the stack and then decrement the STKPTR.

Reference Figure 4-4 through Figure 4-7 for examples of accessing the stack.

FIGURE 4-4: ACCESSING THE STACK EXAMPLE 1

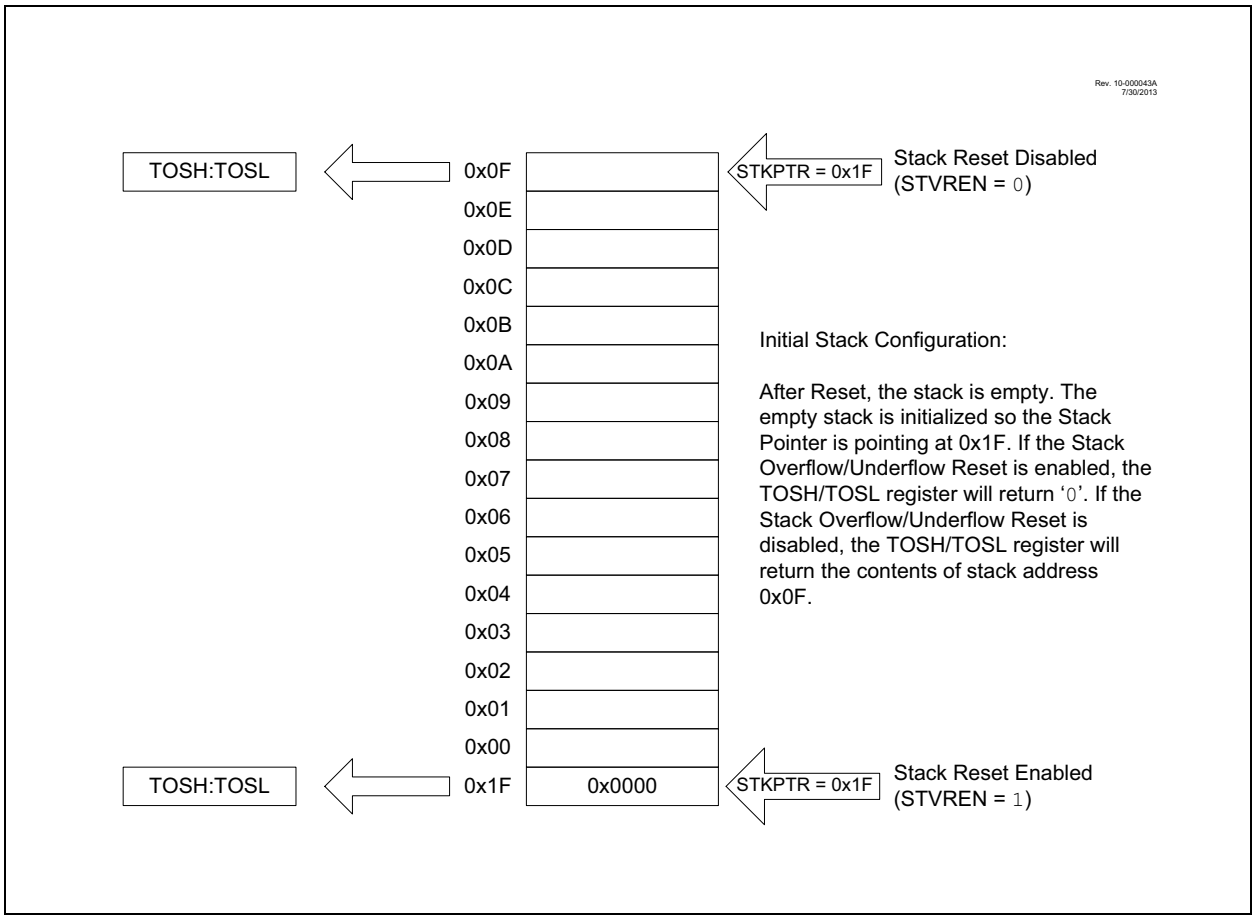


TABLE 11-1: SUMMARY OF REGISTERS ASSOCIATED WITH POWER-DOWN MODE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	121
PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	122
PIE1	OSFIE	CSWIE	—	—	—	—	—	ADIE	123
PIE2	—	ZCDIE	—	—	—	—	C2IE(1)	C1IE	124
PIE3	—	—	RC1IE	TX1IE	—	—	BCL1IE	SSP1IE	125
PIE4	—	—	—	—	—	—	TMR2IE	TMR1IE	126
PIR0	—	—	TMR0IF	IOCIF	—	—	—	INTF	130
PIR1	OSFIF	CSWIF	—	—	—	—	—	ADIF	131
PIR2	—	ZCDIF	—	—	—	—	C2IF(1)	C1IF	132
PIR3	—	—	RC1IF	TX1IF	—	—	BCL1IF	SSP1IF	133
PIR4	—	—	—	—	—	—	TMR2IF	TMR1IF	134
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	204
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	204
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	205
IOCCP(1)	—	—	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	206
IOCCN(1)	—	—	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	206
IOCCF(1)	—	—	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	207
STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	32
VREGCON	—	—	—	—	—	—	VREGPM	—	143
CPUDOZE	IDLEN	DOZEN	ROI	DOE	—	DOZE<2:0>			144
WDTCON0	—	—	WDTPS<4:0>					SWDTEN	150

Legend: — = unimplemented location, read as '0'. Shaded cells are not used in Power-Down mode.

Note 1: Present only in PIC16(L)F15323.

EXAMPLE 13-5: DEVICE ID ACCESS

```
; This write routine assumes the following:
; 1.  A full row of data are loaded, starting at the address in DATA_ADDR
; 2.  Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
; stored in little endian format
; 3.  A valid starting address (the least significant bits = 00000) is loaded in ADDRH:ADDRL
; 4.  ADDRH and ADDRL are located in common RAM (locations 0x70 - 0x7F)
; 5.  NVM interrupts are not taken into account

          BANKSEL             NVMADRH
          MOVF                ADDRH,W
          MOVWF               NVMADRH                 ; Load initial address
          MOVF                ADDRL,W
          MOVWF               NVMADRL
          MOVLW               LOW DATA_ADDR           ; Load initial data address
          MOVWF               FSR0L
          MOVLW               HIGH DATA_ADDR
          MOVWF               FSR0H
          BCF                 NVMCON1,NVMREGS         ; Set PFM as write location
          BSF                 NVMCON1,WREN            ; Enable writes
          BSF                 NVMCON1,LWLO            ; Load only write latches

LOOP
          MOVIW               FSR0++
          MOVWF               NVMDATL                 ; Load first data byte
          MOVIW               FSR0++
          MOVWF               NVMDATH                 ; Load second data byte

          CALL                UNLOCK_SEQ              ; If not, go load latch
          INCF                NVMADRL,F               ; Increment address
          MOVF                NVMADRL,W
          XORLW               0x1F                    ; Check if lower bits of address are 00000
          ANDLW               0x1F                    ; and if on last of 32 addresses
          BTFSC               STATUS,Z                ; Last of 32 words?
          GOTO                START_WRITE             ; If so, go write latches into memory

          GOTO                LOOP

START_WRITE
          BCF                 NVMCON1,LWLO            ; Latch writes complete, now write memory
          CALL                UNLOCK_SEQ              ; Perform required unlock sequence
          BCF                 NVMCON1,LWLO            ; Disable writes

UNLOCK_SEQ
          MOVLW               55h
          BCF                 INTCON,GIE              ; Disable interrupts
          MOVWF               NVMCON2                 ; Begin unlock sequence
          MOVLW               AAh
          MOVWF               NVMCON2
          BSF                 NVMCON1,WR
          BSF                 INTCON,GIE              ; Unlock sequence complete, re-enable interrupts
          return
```

TABLE 15-2: PPS INPUT SIGNAL ROUTING OPTIONS (PIC16(L)F15323)

INPUT SIGNAL NAME	Input Register Name	Default Location at POR	Reset Value (xxxPPS<4:0>)	Remappable to Pins of PORTx	
				PIC16(L)F15323	
				PORTA	PORTC
INT	INTPPS	RA2	00010	•	•
T0CKI	T0CKIPPS	RA2	00010	•	•
T1CKI	T1CKIPSS	RA5	00101	•	•
T1G	T1GPPS	RA4	00100	•	•
T2IN	T2INPPS	RA5	00101	•	•
CCP1	CCP1PPS	RC5	10101	•	•
CCP2	CCP2PPS	RC3	10011	•	•
CWG1IN	CWG1INPPS	RA2	00010	•	•
CLCIN0	CLCIN0PPS	RC3	10011	•	•
CLCIN1	CLCIN1PPS	RC4	10100	•	•
CLCIN2	CLCIN2PPS	RC1	10001	•	•
CLCIN3	CLCIN3PPS	RA5	00101	•	•
ADACT	ADACTPPS	RC2	10010	•	•
SCK1/SCL1	SSP1CLKPPS	RC0	10000	•	•
SDI1/SDA1	SSP1DATPPS	RC1	10001	•	•
SS1	SSP1SS1PPS	RC3	10011	•	•
RX1/DT1	RX1DTPPS	RC5	10101	•	•
CK1	TX1CKPPS	RC4	10100	•	•

15.3 Bidirectional Pins

PPS selections for peripherals with bidirectional signals on a single pin must be made so that the PPS input and PPS output select the same pin. Peripherals that have bidirectional signals include:

- EUSART (synchronous operation)
- MSSP (I^2C)

Note:	The I^2C SCLx and SDAx functions can be remapped through PPS. However, only the RB1, RB2, RC3 and RC4 pins have the I^2C and SMBus specific input buffers implemented (I^2C mode disables INLVL and sets thresholds that are specific for I^2C). If the SCLx or SDAx functions are mapped to some other pin (other than RB1, RB2, RC3 or RC4), the general purpose TTL or ST input buffers (as configured based on INLVL register setting) will be used instead. In most applications, it is therefore recommended only to map the SCLx and SDAx pin functions to the RB1, RB2, RC3 or RC4 pins.

15.4 PPS Lock

The PPS includes a mode in which all input and output selections can be locked to prevent inadvertent changes. PPS selections are locked by setting the PPSLOCKED bit of the PPSLOCK register. Setting and clearing this bit requires a special sequence as an extra precaution against inadvertent changes. Examples of setting and clearing the PPSLOCKED bit are shown in Example 15-1.

EXAMPLE 15-1: PPS LOCK/UNLOCK SEQUENCE

```
; suspend interrupts
    BCF     INTCON,GIE
;   BANKSEL PPSLOCK     ; set bank
; required sequence, next 5 instructions
    MOVLW   0x55
    MOVWF   PPSLOCK
    MOVLW   0xAA
    MOVWF   PPSLOCK
; Set PPSLOCKED bit to disable writes or
; Clear PPSLOCKED bit to enable writes
    BSF     PPSLOCK,PPSLOCKED
; restore interrupts
    BSF     INTCON,GIE
```

15.5 PPS Permanent Lock

The PPS can be permanently locked by setting the PPS1WAY Configuration bit. When this bit is set, the PPSLOCKED bit can only be cleared and set one time after a device Reset. This allows for clearing the PPSLOCKED bit so that the input and output selections can be made during initialization. When the PPSLOCKED bit is set after all selections have been made, it will remain set and cannot be cleared until after the next device Reset event.

15.6 Operation During Sleep

PPS input and output selections are unaffected by Sleep.

15.7 Effects of a Reset

A device Power-on-Reset (POR) clears all PPS input and output selections to their default values (Permanent Lock Removed). All other Resets leave the selections unchanged. Default input selections are shown in Table 15-1 and Table 15-2.

REGISTER 15-2: RxyPPS: PIN Rxy OUTPUT SOURCE SELECTION REGISTER

U-0	U-0	U-0	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	RxyPPS<4:0>				
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 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RxyPPS<4:0>:** Pin Rxy Output Source Selection bits
See Table 15-4 and Table 15-5.

Note 1: TRIS control is overridden by the peripheral as required.

REGISTER 15-3: PPSLOCK: PPS LOCK REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	PPSLOCKED
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 7-1 **Unimplemented:** Read as ‘0’

bit 0 **PPSLOCKED:** PPS Locked bit
1= PPS is locked. PPS selections can not be changed.
0= PPS is not locked. PPS selections can be changed.

REGISTER 16-4: PMD3: PMD CONTROL REGISTER 3

U-0	U-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
—	—	PWM6MD	PWM5MD	PWM4MD	PWM3MD	CCP2MD	CCP1MD
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	q = Value depends on condition

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **PWM6MD:** Disable Pulse-Width Modulator PWM6 bit
1 = PWM6 module disabled
0 = PWM6 module enabled

bit 4 **PWM5MD:** Disable Pulse-Width Modulator PWM5 bit
1 = PWM5 module disabled
0 = PWM5 module enabled

bit 3 **PWM4MD:** Disable Pulse-Width Modulator PWM4 bit
1 = PWM4 module disabled
0 = PWM4 module enabled

bit 2 **PWM3MD:** Disable Pulse-Width Modulator PWM3 bit
1 = PWM3 module disabled
0 = PWM3 module enabled

bit 1 **CCP2MD:** Disable CCP2 bit
1 = CCP2 module disabled
0 = CCP2 module enabled

bit 0 **CCP1MD:** Disable CCP1 bit
1 = CCP1 module disabled
0 = CCP1 module enabled

17.5 Register Definitions: Interrupt-on-Change Control

REGISTER 17-1: IOCAP: INTERRUPT-ON-CHANGE PORTA POSITIVE EDGE REGISTER

U-0	U-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
—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1(1)	IOCAP0(1)
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 7-6 **Unimplemented:** read as '0'

bit 5-0 **IOCAP<5:0>:** Interrupt-on-Change PORTA Positive Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a positive-going edge. IOCAFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

REGISTER 17-2: IOCAN: INTERRUPT-ON-CHANGE PORTA NEGATIVE EDGE REGISTER

U-0	U-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
—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1(1)	IOCAN0(1)
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 7-6 **Unimplemented:** read as '0'

bit 5-0 **IOCAN<5:0>:** Interrupt-on-Change PORTA Negative Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a negative-going edge. IOCAFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

REGISTER 27-3: T2HLT: TIMERx HARDWARE LIMIT CONTROL REGISTER

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	R/W-0/0
PSYNC(1, 2)	CKPOL(3)	CKSYNC(4, 5)	MODE<4:0>(6, 7)				
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 **PSYNC:** Timerx Prescaler Synchronization Enable bit(1, 2)
- 1 = TMRx Prescaler Output is synchronized to Fosc/4
- 0 = TMRx Prescaler Output is not synchronized to Fosc/4

bit 6 **CKPOL:** Timerx Clock Polarity Selection bit(3)
- 1 = Falling edge of input clock clocks timer/prescaler
- 0 = Rising edge of input clock clocks timer/prescaler

bit 5 **CKSYNC:** Timerx Clock Synchronization Enable bit(4, 5)
- 1 = ON register bit is synchronized to TMR2_clk input
- 0 = ON register bit is not synchronized to TMR2_clk input

bit 4-0 **MODE<4:0>:** Timerx Control Mode Selection bits(6, 7)
See Table 27-1.

Note 1: Setting this bit ensures that reading TMRx will return a valid value.

2: When this bit is '1', Timer2 cannot operate in Sleep mode.

3: CKPOL should not be changed while ON = 1.

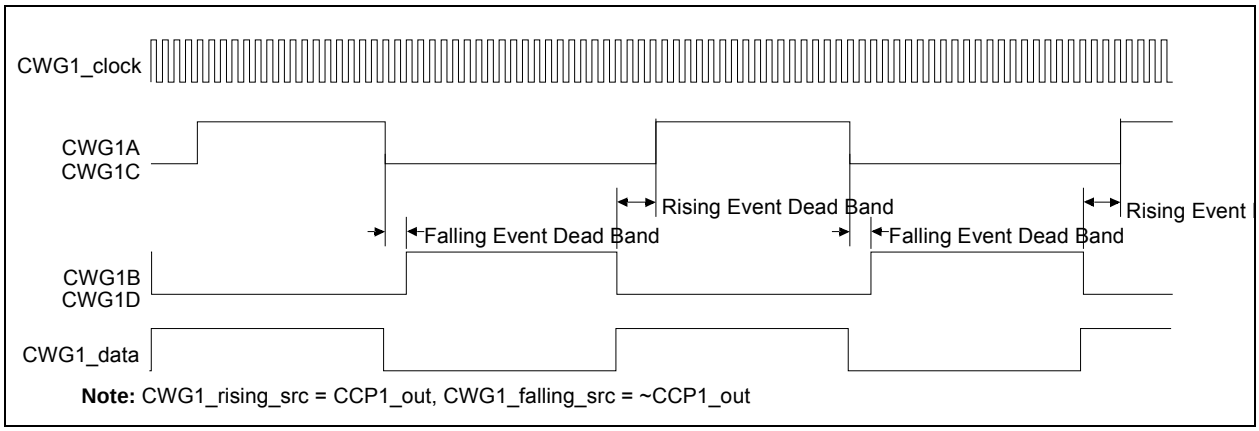
4: Setting this bit ensures glitch-free operation when the ON is enabled or disabled.

5: When this bit is set then the timer operation will be delayed by two TMRx input clocks after the ON bit is set.

6: Unless otherwise indicated, all modes start upon ON = 1 and stop upon ON = 0 (stops occur without affecting the value of TMRx).

7: When TMRx = PRx, the next clock clears TMRx, regardless of the operating mode.

FIGURE 30-9: CWG HALF-BRIDGE MODE OPERATION



Note: CWG1_rising_src = CCP1_out, CWG1_falling_src = ~CCP1_out

30.10 Auto-Shutdown

Auto-shutdown is a method to immediately override the CWG output levels with specific overrides that allow for safe shutdown of the circuit. The shutdown state can be either cleared automatically or held until cleared by software. The auto-shutdown circuit is illustrated in Figure 30-12.

30.10.1 SHUTDOWN

The shutdown state can be entered by either of the following two methods:

- Software generated
- External Input

30.10.1.1 Software Generated Shutdown

Setting the SHUTDOWN bit of the CWG1AS0 register will force the CWG into the shutdown state.

When the auto-restart is disabled, the shutdown state will persist as long as the SHUTDOWN bit is set.

When auto-restart is enabled, the SHUTDOWN bit will clear automatically and resume operation on the next rising edge event.

30.10.2 EXTERNAL INPUT SOURCE

External shutdown inputs provide the fastest way to safely suspend CWG operation in the event of a Fault condition. When any of the selected shutdown inputs goes active, the CWG outputs will immediately go to the selected override levels without software delay. Several input sources can be selected to cause a shutdown condition. All input sources are active-low. The sources are:

- Comparator C1OUT_sync
- Comparator C2OUT_sync
- Timer2 – TMR2_postscaled
- CWG1IN input pin

Shutdown inputs are selected using the CWG1AS1 register (Register 30-6).

Note:	Shutdown inputs are level sensitive, not edge sensitive. The shutdown state cannot be cleared, except by disabling auto-shutdown, as long as the shutdown input level persists.

30.11 Operation During Sleep

The CWG module operates independently from the system clock and will continue to run during Sleep, provided that the clock and input sources selected remain active.

The HFINTOSC remains active during Sleep when all the following conditions are met:

- CWG module is enabled
- Input source is active
- HFINTOSC is selected as the clock source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and the CWG clock source, when the CWG is enabled and the input source is active, then the CPU will go idle during Sleep, but the HFINTOSC will remain active and the CWG will continue to operate. This will have a direct effect on the Sleep mode current.

32.5.6 CLOCK STRETCHING

Clock stretching occurs when a device on the bus holds the SCL line low, effectively pausing communication. The slave may stretch the clock to allow more time to handle data or prepare a response for the master device. A master device is not concerned with stretching as anytime it is active on the bus and not transferring data it is stretching. Any stretching done by a slave is invisible to the master software and handled by the hardware that generates SCL.

The CKP bit of the SSP1CON1 register is used to control stretching in software. Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. Setting CKP will release SCL and allow more communication.

32.5.6.1 Normal Clock Stretching

Following an $\overline{ACK}$ if the R/$\overline{W}$ bit of SSP1STAT is set, a read request, the slave hardware will clear CKP. This allows the slave time to update SSP1BUF with data to transfer to the master. If the SEN bit of SSP1CON2 is set, the slave hardware will always stretch the clock after the $\overline{ACK}$ sequence. Once the slave is ready; CKP is set by software and communication resumes.

32.5.6.2 10-bit Addressing Mode

In 10-bit Addressing mode, when the UA bit is set the clock is always stretched. This is the only time the SCL is stretched without CKP being cleared. SCL is released immediately after a write to SSP1ADD.

32.5.6.3 Byte NACKing

When AHEN bit of SSP1CON3 is set; CKP is cleared by hardware after the eighth falling edge of SCL for a received matching address byte. When DHEN bit of SSP1CON3 is set; CKP is cleared after the eighth falling edge of SCL for received data.

Stretching after the eighth falling edge of SCL allows the slave to look at the received address or data and decide if it wants to ACK the received data.

32.5.7 CLOCK SYNCHRONIZATION AND THE CKP BIT

Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. However, clearing the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external I^2C master device has already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I^2C bus have released SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 32-23).

FIGURE 32-23: CLOCK SYNCHRONIZATION TIMING

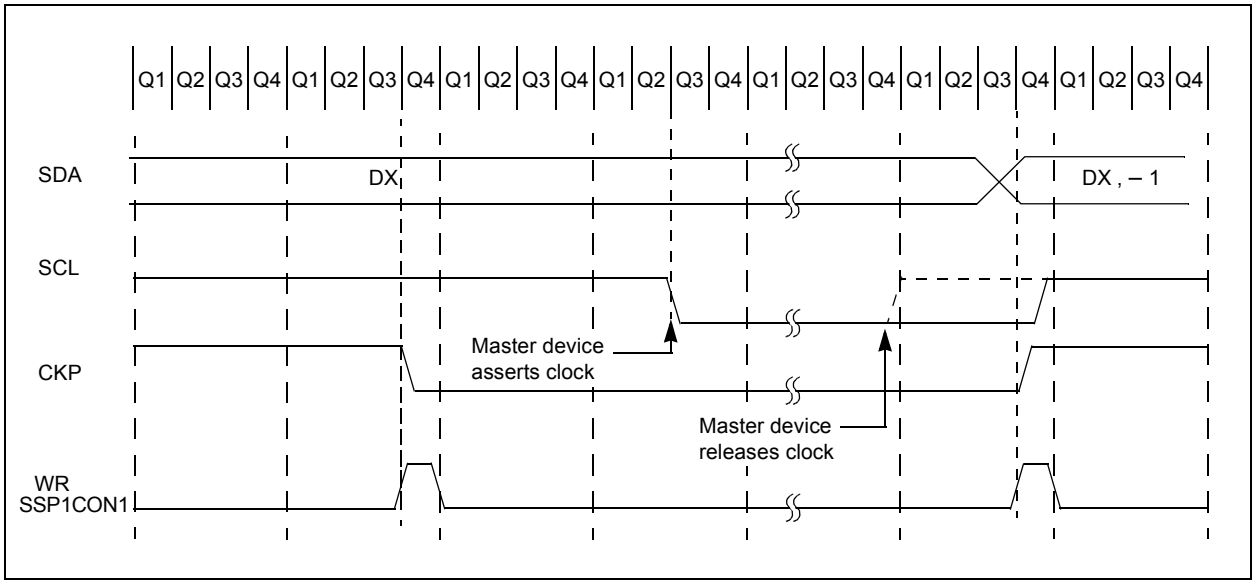


FIGURE 33-7: AUTO-WAKE-UP BIT (WUE) TIMING DURING NORMAL OPERATION

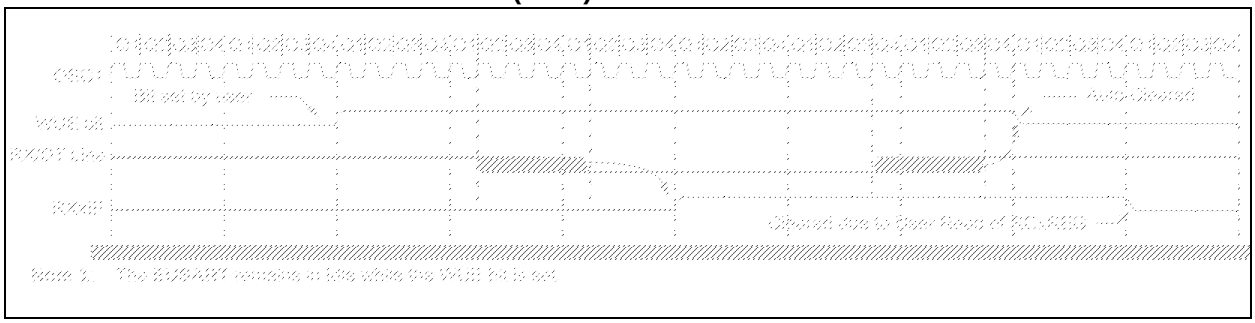
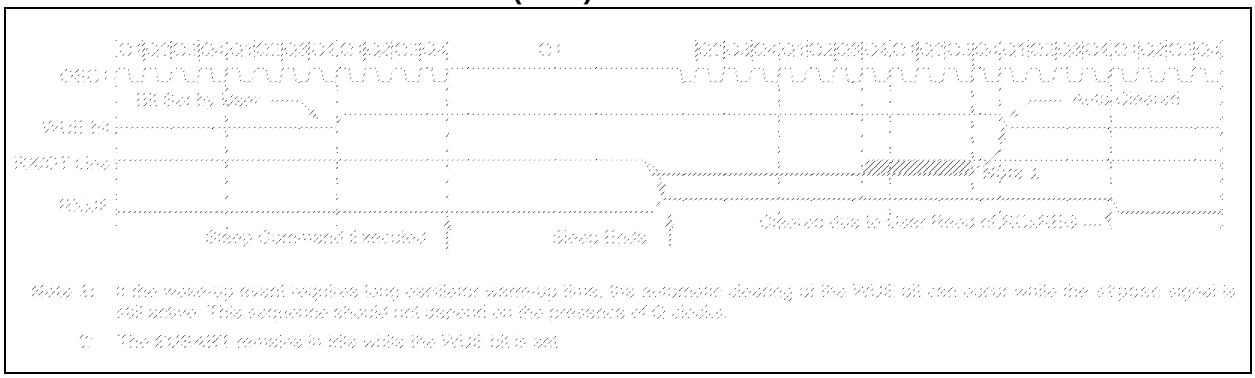


FIGURE 33-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP



33.3.4 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. A Break character consists of a Start bit, followed by 12 '0' bits and a Stop bit.

To send a Break character, set the SENDB and TXEN bits of the TX1STA register. The Break character transmission is then initiated by a write to the TX1REG. The value of data written to TX1REG will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

The TRMT bit of the TX1STA register indicates when the transmit operation is active or idle, just as it does during normal transmission. See Figure 33-9 for the timing of the Break character sequence.

33.3.4.1 Break and Sync Transmit Sequence

The following sequence will start a message frame header made up of a Break, followed by an auto-baud Sync byte. This sequence is typical of a LIN bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENDB bits to enable the Break sequence.
3. Load the TX1REG with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TX1REG to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware and the Sync character is then transmitted.

When the TX1REG becomes empty, as indicated by the TX1IF, the next data byte can be written to TX1REG.

BCF **Bit Clear f**

Syntax:	[*label*] BCF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	0 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

BRA **Relative Branch**

Syntax:	[*label*] BRA label [*label*] BRA $+k
Operands:	-256 ≤ label - PC + 1 ≤ 255 -256 ≤ k ≤ 255
Operation:	(PC) + 1 + k → PC
Status Affected:	None
Description:	Add the signed 9-bit literal 'k' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 1 + k. This instruction is a 2-cycle instruction. This branch has a limited range.

BRW **Relative Branch with W**

Syntax:	[*label*] BRW
Operands:	None
Operation:	(PC) + (W) → PC
Status Affected:	None
Description:	Add the contents of W (unsigned) to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 1 + (W). This instruction is a 2-cycle instruction.

BSF **Bit Set f**

Syntax:	[*label*] BSF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	1 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

BTFSC **Bit Test f, Skip if Clear**

Syntax:	[*label*] BTFSC f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2-cycle instruction.

BTFSS **Bit Test f, Skip if Set**

Syntax:	[*label*] BTFSS f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b < 7
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction.

FIGURE 37-14: CLC PROPAGATION TIMING

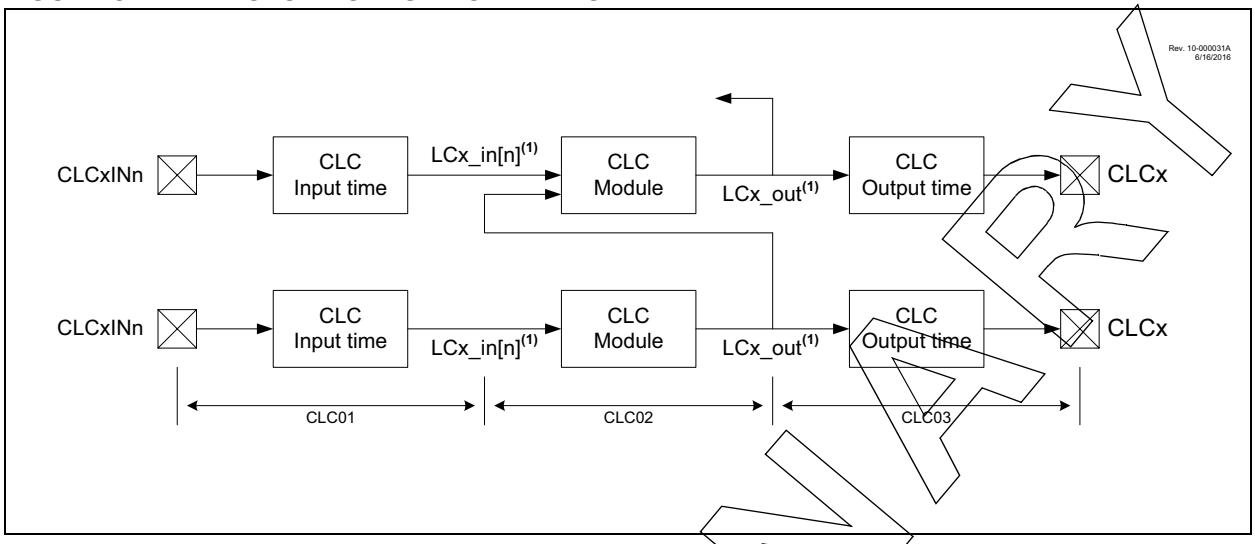


TABLE 37-20: CONFIGURABLE LOGIC CELL (CLC) CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
CLC01*	TCLCIN	CLC input time	—	7	IO5	ns	**(Note 1)**
CLC02*	TCLC	CLC module input to output propagation time	— —	24 12	— —	ns ns	VDD = 1.8V VDD > 3.6V
CLC03*	TCLCOUT	CLC output time Rise Time	—	IO7	—	—	**(Note 1)**
		Fall Time	—	IO8	—	—	**(Note 1)**
CLC04*	FCLCMAX	CLC maximum switching frequency	—	32	FOSC	MHz	

* These parameters are characterized but not tested.

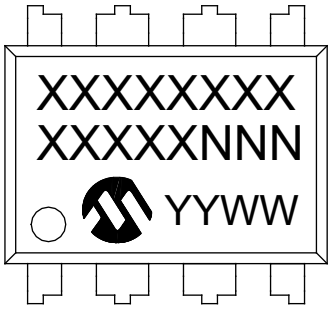
† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: See Table 37-10 for IO5, IO7 and IO8 rise and fall times.

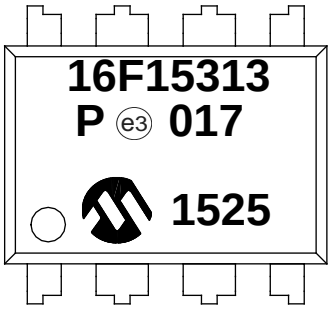
40.0 PACKAGING INFORMATION

40.1 Package Marking Information

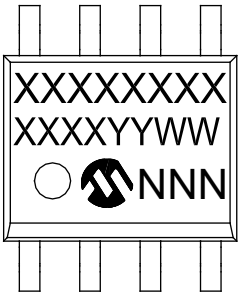
8-Lead PDIP (300 mil)



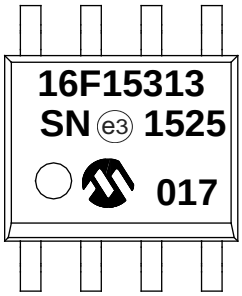
Example



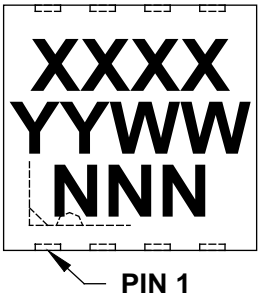
8-Lead SOIC (3.90 mm)



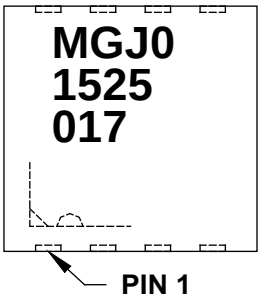
Example



8-Lead UDFN (3x3x0.9 mm)



Example

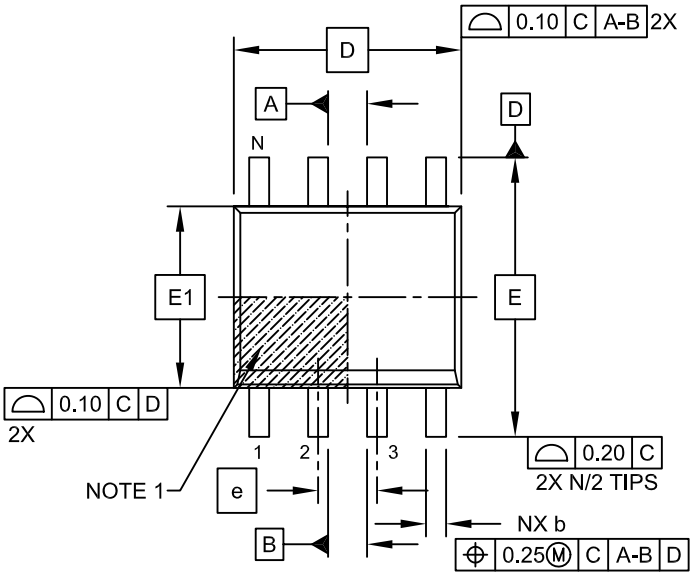


Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
		Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free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

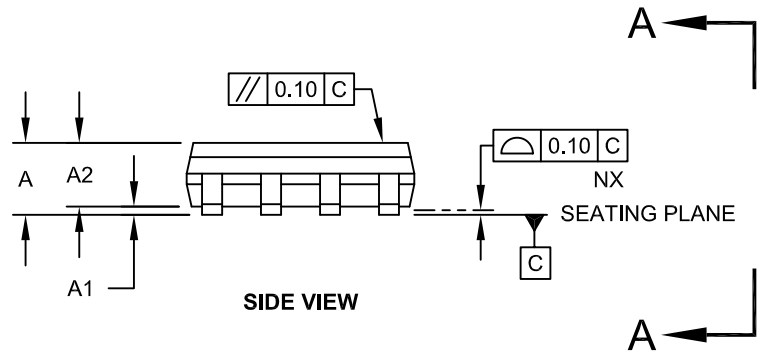
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

8-Lead Plastic Small Outline (SN) - Narrow, 3.90 mm Body [SOIC]

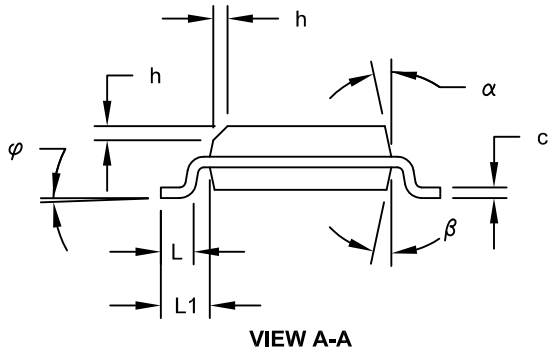
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW



VIEW A-A

Microchip Technology Drawing No. C04-057C Sheet 1 of 2