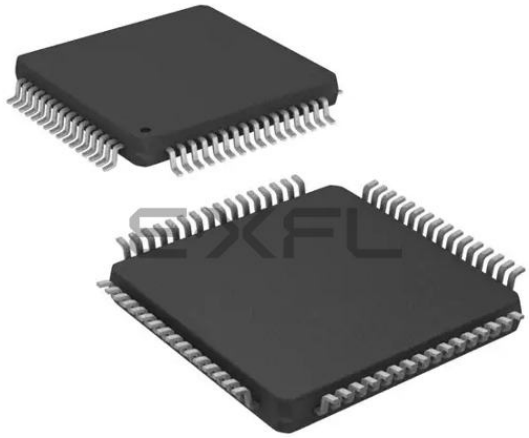
Microchip Technology - PIC18C658T-I/PT Datasheet



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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	52
Program Memory Size	32KB (16K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18c658t-i-pt

4.2.3 PUSH AND POP INSTRUCTIONS

Since the Top-of-Stack (TOS) is readable and writable, the ability to push values onto the stack and pull values off the stack without disturbing normal program execution is a desirable option. To push the current PC value onto the stack, a `PUSH` instruction can be executed. This will increment the stack pointer and load the current PC value onto the stack. TOSU, TOSH and TOSL can then be modified to place a return address on the stack.

The `POP` instruction discards the current TOS by decrementing the stack pointer. The previous value pushed onto the stack then becomes the TOS value.

4.2.4 STACK FULL/UNDERFLOW RESETS

These RESETs are enabled by programming the STVREN configuration bit. When the STVREN bit is disabled, a full or underflow condition will set the appropriate STKFUL or STKUNF bit, but not cause a device RESET. When the STVREN bit is enabled, a full or underflow will set the appropriate STKFUL or STKUNF bit and then cause a device RESET. The STKFUL or STKUNF bits are only cleared by the user software or a POR.

4.3 Fast Register Stack

A “fast return” option is available for interrupts and calls. A fast register stack is provided for the STATUS, WREG and BSR registers and is only one layer in depth. The stack is not readable or writable and is loaded with the current value of the corresponding register when the processor vectors for an interrupt. The values in the fast register stack are then loaded back into the working registers if the `fast return` instruction is used to return from the interrupt.

A low or high priority interrupt source will push values into the stack registers. If both low and high priority interrupts are enabled, the stack registers cannot be used reliably for low priority interrupts. If a high priority interrupt occurs while servicing a low priority interrupt, the stack register values stored by the low priority interrupt will be overwritten.

If high priority interrupts are not disabled during low priority interrupts, users must save the key registers in software during a low priority interrupt.

If no interrupts are used, the fast register stack can be used to restore the STATUS, WREG and BSR registers at the end of a subroutine call. To use the fast register stack for a subroutine call, a `fast call` instruction must be executed.

Example 4-1 shows a source code example that uses the fast register stack.

EXAMPLE 4-1: FAST REGISTER STACK CODE EXAMPLE

```
CALL SUB1, FAST    ;STATUS, WREG, BSR
                   ;SAVED IN FAST REGISTER
                   ;STACK
       •
       •

SUB1   •
       •
       •
   RETURN FAST     ;RESTORE VALUES SAVED
                   ;IN FAST REGISTER STACK
```


Filename	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)
CCPR1H	Capture/Compare/PWM Register 1 High Byte								xxxx xxxx	uuuu uuuu
CCPR1L	Capture/Compare/PWM Register 1 Low Byte								xxxx xxxx	uuuu uuuu
CCP1CON	—	—	DC1B1	DC1B0	CCPM3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
CCPR2H	Capture/Compare/PWM Register 2 High Byte								xxxx xxxx	uuuu uuuu
CCPR2L	Capture/Compare/PWM Register 2 Low Byte								xxxx xxxx	uuuu uuuu
CCP2CON	—	—	DC2B1	DC2B0	CCPM3	CCP2M2	CCP2M1	CCP2M0	0000 0000	0000 0000
VRCON	VREN	VROEN	VRR	VRSS	VR3	VR2	VR1	VR0	0000 0000	0000 0000
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
TMR3H	Timer3 Register High Byte								xxxx xxxx	uuuu uuuu
TMR3L	Timer3 Register Low Byte								xxxx xxxx	uuuu uuuu
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON	0000 0000	uuuu uuuu
PSPCON	IBF	OBF	IBOV	PSPMODE	—	—	—	—	0000 ----	0000 ----
SPBRG	USART Baud Rate Generator								0000 0000	0000 0000
RCREG	USART Receive Register								0000 0000	0000 0000
TXREG	USART Transmit Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	0000 000x
IPR3	IRXIP	WAKIP	ERRIP	TXB2IP	TXB1IP	TXB0IP	RXB1IP	RXB0IP	1111 1111	1111 1111
PIR3	IRXIF	WAKIF	ERRIF	TXB2IF	TXB1IF	TXB0IF	RXB1IF	RXB0IF	0000 0000	0000 0000
PIE3	IRXIE	WAKIE	ERRIE	TXB2IE	TXB1IE	TXB0IE	RXB1IE	RXB0IE	0000 0000	0000 0000
IPR2	—	CMIP	—	—	BCLIP	LVDIP	TMR3IP	CCP2IP	-1-- 1111	-1-- 1111
PIR2	—	CMIF	—	—	BCLIF	LVDIF	TMR3IF	CCP2IF	-0-- 0000	-0-- 0000
PIE2	—	CMIE	—	—	BCLIE	LVDIE	TMR3IE	CCP2IE	-0-- 0000	-0-- 0000
IPR1	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
TRISJ(4)	Data Direction Control Register for PORTJ								1111 1111	1111 1111
TRISH(4)	Data Direction Control Register for PORTH								1111 1111	1111 1111
TRISG	—	—	—	Data Direction Control Register for PORTG					---1 1111	---1 1111
TRISF	Data Direction Control Register for PORTF								1111 1111	1111 1111
TRISE	Data Direction Control Register for PORTE								1111 1111	1111 1111
TRISD	Data Direction Control Register for PORTD								1111 1111	1111 1111
TRISC	Data Direction Control Register for PORTC								1111 1111	1111 1111
TRISB	Data Direction Control Register for PORTB								1111 1111	1111 1111
TRISA	—	Bit 6(1)	Data Direction Control Register for PORTA						--11 1111	--11 1111

Legend: x = unknown, u = unchanged, - = unimplemented, q = value depends on condition

Note 1: Bit 6 of PORTA, LATA and TRISA are enabled in ECIO and RCIO oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

3: Other (non-power-up) RESETs include external RESET through MCLR and Watchdog Timer Reset.

4: These registers are reserved on PIC18C658.

5.2.2.1 Long Write Operation

The long write is what actually programs words of data into the internal memory. When a TBLWT to the MSB of the write block occurs, instruction execution is halted. During this time, programming voltage and the data stored in internal latches is applied to program memory.

For a long write to occur:

1. $\overline{MCLR}/V_{PP}$ pin must be at the programming voltage
2. LWRT bit must be set
3. TBLWT to the address of the MSB of the write block

If the LWRT bit is clear, a short write will occur and program memory will not be changed. If the TBLWT is not to the MSB of the write block, then the programming phase is not initiated.

Setting the LWRT bit enables long writes when the $\overline{MCLR}$ pin is taken to V_{PP} voltage. Once the LWRT bit is set, it can be cleared only by performing a POR or $\overline{MCLR}$ Reset.

To ensure that the memory location has been well programmed, a minimum programming time is required. The long write can be terminated after the programming time has expired by a RESET or an interrupt. Having only one interrupt source enabled to terminate the long write, ensures that no unintended interrupts will prematurely terminate the long write.

5.2.2.2 Sequence of Events

The sequence of events for programming an internal program memory location should be:

1. Enable the interrupt that terminates the long write. Disable all other interrupts.
2. Clear the source interrupt flag.
3. If Interrupt Service Routine execution is desired when the device wakes, enable global interrupts.
4. Set LWRT bit in the RCON register.
5. Raise $\overline{MCLR}/V_{PP}$ pin to the programming voltage, V_{PP}.
6. Clear the WDT (if enabled).
7. Set the interrupt source to interrupt at the required time.
8. Execute the Table Write for the lower (even) byte. This will be a short write.
9. Execute the Table Write for the upper (odd) byte. This will be a long write. The controller will HALT while programming. The interrupt wakes the controller.
10. If GIE was set, service the interrupt request.
11. Go to 7 if more bytes to be programmed.
12. Lower $\overline{MCLR}/V_{PP}$ pin to V_{DD}.
13. Verify the memory location (table read).
14. Reset the device.

5.2.3 LONG WRITE INTERRUPTS

The long write must be terminated by a RESET or any interrupt.

The interrupt source must have its interrupt enable bit set. When the source sets its interrupt flag, programming will terminate. This will occur regardless of the settings of interrupt priority bits, the GIE/GIEH bit or the PIE/GIEL bit.

Depending on the states of interrupt priority bits, the GIE/GIEH bit or the PIE/GIEL bit, program execution can either be vectored to the high or low priority Interrupt Service Routine (ISR), or continue execution from where programming commenced.

In either case, the interrupt flag will not be cleared when programming is terminated and will need to be cleared by the software.

5.3 Unexpected Termination of Write Operations

If a write is terminated by an unplanned event such as loss of power, an unexpected RESET, or an interrupt that was not disabled, the memory location just programmed should be verified and reprogrammed if needed.

TABLE 5-2: SLEEP MODE, INTERRUPT ENABLE BITS AND INTERRUPT RESULTS

GIE/ GIEH	PIE/ GIEL	Priority	Interrupt Enable	Interrupt Flag	Action
X	X	X	0 (default)	X	Long write continues even if interrupt flag becomes set during SLEEP.
X	X	X	1	0	Long write continues, will wake when the interrupt flag is set.
0 (default)	0 (default)	X	1	1	Terminates long write, executes next instruction. Interrupt flag not cleared.
0 (default)	1	1 high priority (default)	1	1	Terminates long write, executes next instruction. Interrupt flag not cleared.
1	0 (default)	0 low	1	1	Terminates long write, executes next instruction. Interrupt flag not cleared.
0 (default)	1	0 low	1	1	Terminates long write, branches to low priority interrupt vector. Interrupt flag can be cleared by ISR.
1	0 (default)	1 high priority (default)	1	1	Terminates long write, branches to high priority interrupt vector. Interrupt flag can be cleared by ISR.

REGISTER 7-5: PIR REGISTERS (CONT'D)

PIR3

bit 7 **IRXIF:** Invalid Message Received Interrupt Flag bit
1 = An invalid message has occurred on the CAN bus
0 = An invalid message has not occurred on the CAN bus

bit 6 **WAKIF:** Bus Activity Wake-up Interrupt Flag bit
1 = Activity on the CAN bus has occurred
0 = Activity on the CAN bus has not occurred

bit 5 **ERRIF:** CAN Bus Error Interrupt Flag bit
1 = An error has occurred in the CAN module (multiple sources)
0 = An error has not occurred in the CAN module

bit 4 **TXB2IF:** Transmit Buffer 2 Interrupt Flag bit
1 = Transmit Buffer 2 has completed transmission of a message, and may be reloaded
0 = Transmit Buffer 2 has not completed transmission of a message

bit 3 **TXB1IF:** Transmit Buffer 1 Interrupt Flag bit
1 = Transmit Buffer 1 has completed transmission of a message, and may be reloaded
0 = Transmit Buffer 1 has not completed transmission of a message

bit 2 **TXB0IF:** Transmit Buffer 0 Interrupt Flag bit
1 = Transmit Buffer 0 has completed transmission of a message, and may be reloaded
0 = Transmit Buffer 0 has not completed transmission of a message

bit 1 **RXB1IF:** Receive Buffer 1 Interrupt Flag bit
1 = Receive Buffer 1 has received a new message
0 = Receive Buffer 1 has not received a new message

bit 0 **RXB0IF:** Receive Buffer 0 Interrupt Flag bit
1 = Receive Buffer 0 has received a new message
0 = Receive Buffer 0 has not received a new message

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

TABLE 8-13: PORTG FUNCTIONS

Name	Bit#	Buffer Type	Function
RG0/CANTX0	bit0	ST	Input/output port pin or CAN bus transmit output.
RG1/CANTX1	bit1	ST	Input/output port pin or CAN bus complimentary transmit output or CAN bus bit time clock.
RG2/CANRX	bit2	ST	Input/output port pin or CAN bus receive input.
RG3	bit3	ST	Input/output port pin.
RG4	bit4	ST	Input/output port pin.

Legend: ST = Schmitt Trigger input

Note: Refer to "CAN Module", Section 17.0 for usage of CAN pin functions.

TABLE 8-14: SUMMARY OF REGISTERS ASSOCIATED WITH PORTG

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
TRISG	PORTG Data Direction Control Register								---1 1111	---1 1111
PORTG	Read PORTG pin / Write PORTG Data Latch								---x xxxx	---u uuuu
LATG	Read PORTG Data Latch/Write PORTG Data Latch								---x xxxx	---u uuuu
CIOCON	TX1SRC	TX1EN	ENDRHI	CANCAP	—	—	—	—	0000 ----	0000 ----

Legend: x = unknown, u = unchanged

NOTES:

14.3.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep bit CCP1IE (PIE registers) clear to avoid false interrupts and should clear the flag bit CCP1IF, following any such change in operating mode.

14.3.4 CCP PRESCALER

There are four prescaler settings, specified by bits CCP1M3:CCP1M0. Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. This means that any RESET will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared; therefore, the first capture may be from a non-zero prescaler. Example 14-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

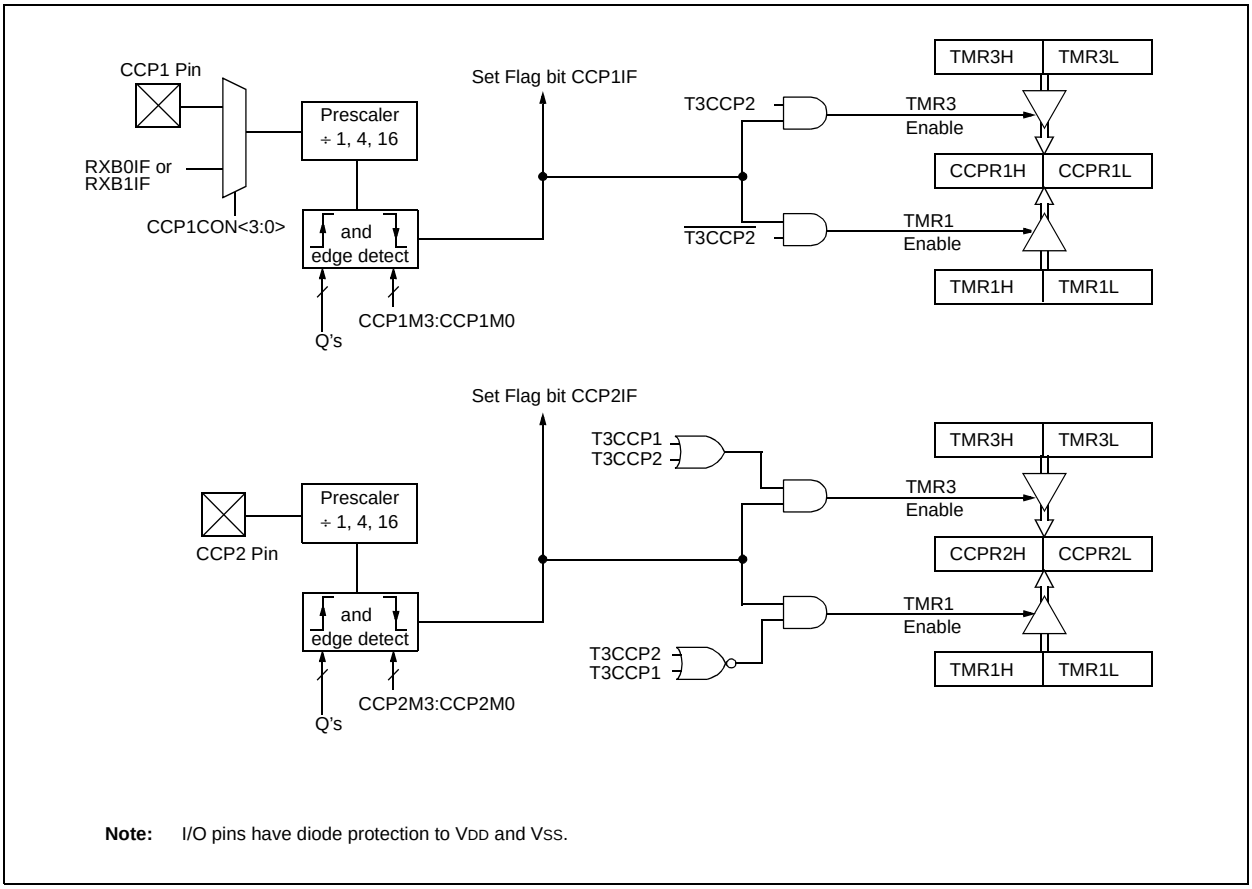
14.3.5 CAN MESSAGE RECEIVED

The CAN capture event occurs when a message is received in either receive buffer. The CAN module provides a rising edge to the CCP module to cause a capture event. This feature is provided to time-stamp the received CAN messages.

EXAMPLE 14-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
CLRF   CCP1CON, F  ; Turn CCP module off
MOVLW  NEW_CAPT_PS ; Load WREG with the
                   ; new prescaler mode
                   ; value and CCP ON
MOVWF  CCP1CON     ; Load CCP1CON with
                   ; this value
```

FIGURE 14-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



Note: I/O pins have diode protection to VDD and VSS.

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
1000 = I^2C Master mode, clock = FOSC / (4 * (SSPADD+1))
1001 = Reserved
1010 = Reserved
1011 = I^2C firmware controlled Master mode (Slave idle)
1100 = Reserved
1101 = Reserved
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

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

15.4.15 MULTI-MASTER MODE

In Multi-master mode, the interrupt generation on the detection of the START and STOP conditions allows the determination of when the bus is free. The STOP (P) and START (S) bits are cleared from a RESET, or when the MSSP module is disabled. Control of the I^2C bus may be taken when the P bit (SSPSTAT register) is set, or the bus is idle with both the S and P bits clear. When the bus is busy, enabling the SSP Interrupt will generate the interrupt when the STOP condition occurs.

In Multi-master operation, the SDA line must be monitored for arbitration, to see if the signal level is the expected output level. This check is performed in hardware, with the result placed in the BCLIF bit.

Arbitration can be lost in the following states:

- Address transfer
- Data transfer
- A START condition
- A Repeated START condition
- An Acknowledge condition

15.4.16 MULTI -MASTER COMMUNICATION, BUS COLLISION, AND BUS ARBITRATION

Multi-master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin = '0', then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag (BCLIF) and reset the I^2C port to its IDLE state. (Figure 15-20).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF bit is cleared, the SDA and SCL lines are de-asserted, and the SSPBUF can be written to. When the user services the bus collision interrupt service routine, and if the I^2C bus is free, the user can resume communication by asserting a START condition.

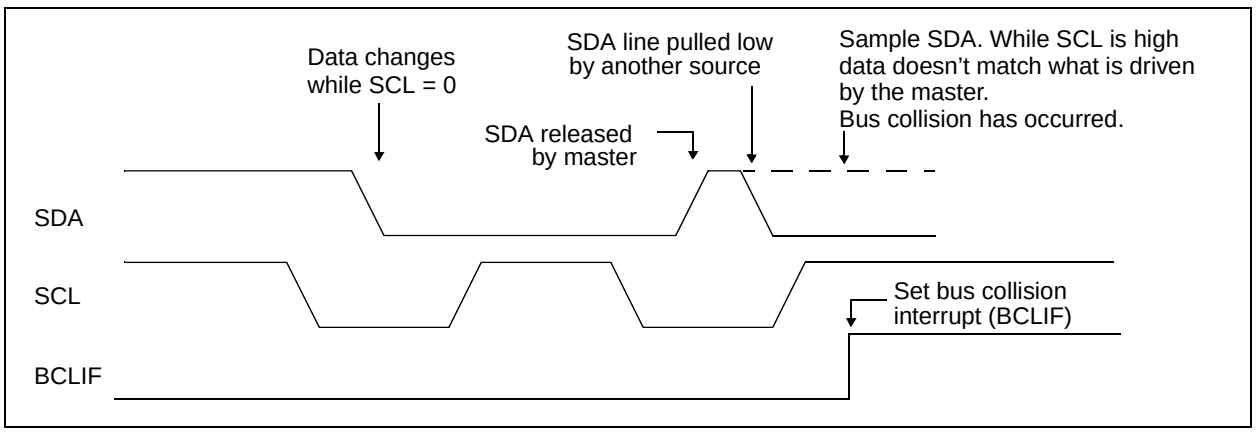
If a START, Repeated START, STOP, or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are de-asserted, and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision interrupt service routine, and if the I^2C bus is free, the user can resume communication by asserting a START condition.

The master will continue to monitor the SDA and SCL pins. If a STOP condition occurs, the SSPIF bit will be set.

A write to the SSPBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In Multi-master mode, the interrupt generation on the detection of START and STOP conditions allows the determination of when the bus is free. Control of the I^2C bus can be taken when the P bit is set in the SSPSTAT register, or the bus is idle and the S and P bits are cleared.

FIGURE 15-20: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE



REGISTER 17-9: TXBnDm – TRANSMIT BUFFER n DATA FIELD BYTE m REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
TXBnDm7	TXBnDm6	TXBnDm5	TXBnDm4	TXBnDm3	TXBnDm2	TXBnDm1	TXBnDm0
bit 7							bit 0

bit 1-0 **TXBnDm7:TXBnDm0:** Transmit Buffer n Data Field Byte m bits (where 0≤n<3 and 0<m<8)

Each Transmit Buffer has an array of registers. For example, Transmit buffer 0 has 7 registers: TXB0D0 to TXB0D7.

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

REGISTER 17-10: TXBnDLC – TRANSMIT BUFFER n DATA LENGTH CODE REGISTER

U-0	R/W-x	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
—	TXRTR	—	—	DLC3	DLC2	DLC1	DLC0
bit 7							bit 0

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

bit 6 **TXRTR:** Transmission Frame Remote Transmission Request bit
1 = Transmitted message will have TXRTR bit set
0 = Transmitted message will have TXRTR bit cleared.

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **DLC3:DLC0:** Data Length Code bits
1111 = Reserved
1110 = Reserved
1101 = Reserved
1100 = Reserved
1011 = Reserved
1010 = Reserved
1001 = Reserved
1000 = Data Length = 8 bytes
0111 = Data Length = 7 bytes
0110 = Data Length = 6 bytes
0101 = Data Length = 5 bytes
0100 = Data Length = 4 bytes
0011 = Data Length = 3 bytes
0010 = Data Length = 2 bytes
0001 = Data Length = 1 bytes
0000 = Data Length = 0 bytes

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

17.2.3 CAN RECEIVE BUFFER REGISTERS

This section shows the Receive Buffer registers with its associated control registers.

REGISTER 17-12: RXB0CON – RECEIVE BUFFER 0 CONTROL REGISTER

R/C-0	R/W-0	R/W-0	U-0	R-0	R/W-0	R-0	R/W-0
RXFUL	RXM1	RXM0	—	RXRTRRO	RXB0DBEN	JTOFF	FILHIT0
bit 7							bit 0

bit 7 **RXFUL:** Receive Full Status bit
1 = Receive buffer contains a received message
0 = Receive buffer is open to receive a new message

Note: This bit is set by the CAN module and should be cleared by software after the buffer is read.

bit 6-5 **RXM1:RXM0:** Receive Buffer Mode bits
11 = Receive all messages (including those with errors)
10 = Receive only valid messages with extended identifier
01 = Receive only valid messages with standard identifier
00 = Receive all valid messages

bit 4 **Unimplemented:** Read as '0'

bit 3 **RXRTRRO:** Receive Remote Transfer Request Read Only bit
1 = Remote transfer request
0 = No remote transfer request

bit 2 **RXB0DBEN:** Receive Buffer 0 Double Buffer Enable bit
1 = Receive Buffer 0 overflow will write to Receive Buffer 1
0 = No Receive Buffer 0 overflow to Receive Buffer 1

bit 1 **JTOFF:** Jump Table Offset bit (read only copy of RX0DBEN)
1 = Allows Jump Table offset between 6 and 7
0 = Allows Jump Table offset between 1 and 0

Note: This bit allows same filter jump table for both RXB0CON and RXB1CON.

bit 0 **FILHIT0:** Filter Hit bit
This bit indicates which acceptance filter enabled the message reception into receive buffer 0
1 = Acceptance Filter 1 (RXF1)
0 = Acceptance Filter 0 (RXF0)

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

18.0 10-BIT ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The analog-to-digital (A/D) converter module has twelve inputs for the PIC18C658 devices and sixteen for the PIC18C858 devices. This module has the ADCON0, ADCON1, and ADCON2 registers.

The A/D allows conversion of an analog input signal to a corresponding 10-bit digital number.

The A/D module has five registers:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)
- A/D Control Register 2 (ADCON2)

The ADCON0 register, shown in Register 18-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 18-2, configures the functions of the port pins. The ADCON2, shown in Register 16-3, configures the A/D clock source and justification.

REGISTER 18-1: ADCON0 REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

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

bit 5-2 **CHS3:CHS0:** Analog Channel Select bits
0000 = channel 00, (AN0)
0001 = channel 01, (AN1)
0010 = channel 02, (AN2)
0011 = channel 03, (AN3)
0100 = channel 04, (AN4)
0101 = channel 05, (AN5)
0110 = channel 06, (AN6)
0111 = channel 07, (AN7)
1000 = channel 08, (AN8)
1001 = channel 09, (AN9)
1010 = channel 10, (AN10)
1011 = channel 11, (AN11)
1100 = channel 12, (AN12)**(1)**
1101 = channel 13, (AN13)**(1)**
1110 = channel 14, (AN14)**(1)**
1111 = channel 15, (AN15)**(1)**

Note 1: These channels are not available on the PIC18C658 devices.

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
When ADON = 1
1 = A/D conversion in progress. Setting this bit starts an A/D conversion cycle. This bit is automatically cleared by hardware when the A/D conversion is complete.
0 = A/D conversion not in progress

bit 0 **ADON:** A/D On bit
1 = A/D converter module is operating
0 = A/D converter module is shut off and consumes no operating current

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

19.6 Comparator Interrupts

The comparator interrupt flag is set whenever there is a change in the output value of either comparator. Software will need to maintain information about the status of the output bits, as read from CMCON<7:6>, to determine the actual change that occurred. The CMIF bit (PIR registers) is the comparator interrupt flag. The CMIF bit must be RESET by clearing '0'. Since it is also possible to write a '1' to this register, a simulated interrupt may be initiated.

The CMIE bit (PIE registers) and the PEIE bit (INTCON register) must be set to enable the interrupt. In addition, the GIE bit must also be set. If any of these bits are clear, the interrupt is not enabled, though the CMIF bit will still be set if an interrupt condition occurs.

.

Note:	If a change in the CMCON register (C1OUT or C2OUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CMIF (PIR registers) interrupt flag may not get set.

The user, in the interrupt service routine, can clear the interrupt in the following manner:

a) Any read or write of CMCON will end the mismatch condition.

b) Clear flag bit CMIF.

A mismatch condition will continue to set flag bit CMIF. Reading CMCON will end the mismatch condition, and allow flag bit CMIF to be cleared.

19.7 Comparator Operation During SLEEP

When a comparator is active and the device is placed in SLEEP mode, the comparator remains active and the interrupt is functional if enabled. This interrupt will wake-up the device from SLEEP mode, when enabled. While the comparator is powered up, higher SLEEP currents than shown in the power-down current specification will occur. Each operational comparator will consume additional current, as shown in the comparator specifications. To minimize power consumption while in SLEEP mode, turn off the comparators, CM<2:0> = 111, before entering SLEEP. If the device wakes up from SLEEP, the contents of the CMCON register are not affected.

19.8 Effects of a RESET

A device RESET forces the CMCON register to its RESET state, causing the comparator module to be in the comparator RESET mode, CM<2:0> = 000. This ensures that all potential inputs are analog inputs. Device current is minimized when analog inputs are present at RESET time. The comparators will be powered down during the RESET interval.

NOTES:

FIGURE 25-3: LOW VOLTAGE DETECT CHARACTERISTICS

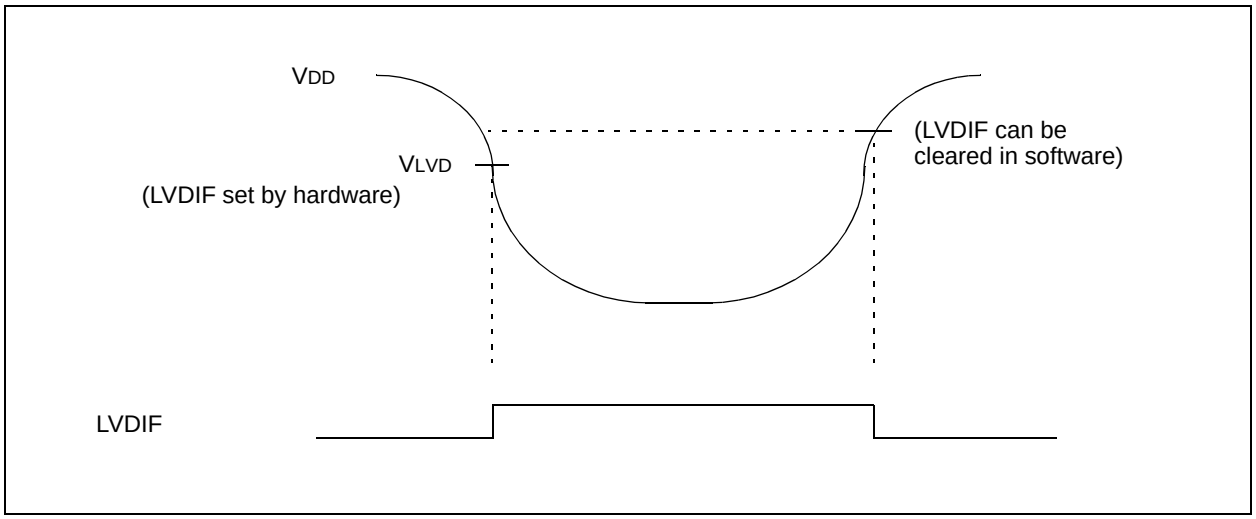


TABLE 25-1: LOW VOLTAGE DETECT CHARACTERISTICS

			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Symbol**	**Characteristic/**		**Min**	**Max**	**Units**	**Conditions**
D420	VLVD	LVD Voltage	LVDL<3:0> = 0100	2.5	2.66	V	
			LVDL<3:0> = 0101	2.7	2.86	V	
			LVDL<3:0> = 0110	2.8	2.98	V	
			LVDL<3:0> = 0111	3.0	3.2	V	
			LVDL<3:0> = 1000	3.3	3.52	V	
			LVDL<3:0> = 1001	3.5	3.72	V	
			LVDL<3:0> = 1010	3.6	3.84	V	
			LVDL<3:0> = 1011	3.8	4.04	V	
			LVDL<3:0> = 1100	4.0	4.26	V	
			LVDL<3:0> = 1101	4.2	4.46	V	
			LVDL<3:0> = 1110	4.5	4.78	V	

25.3.3 TIMING DIAGRAMS AND SPECIFICATIONS

FIGURE 25-5: EXTERNAL CLOCK TIMING

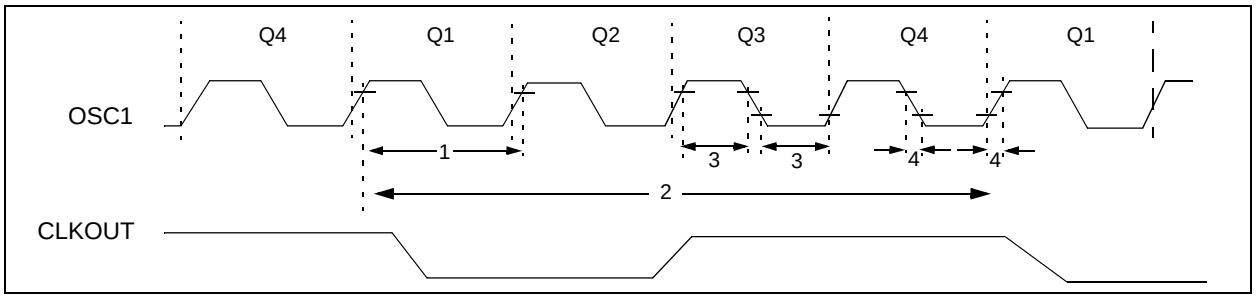
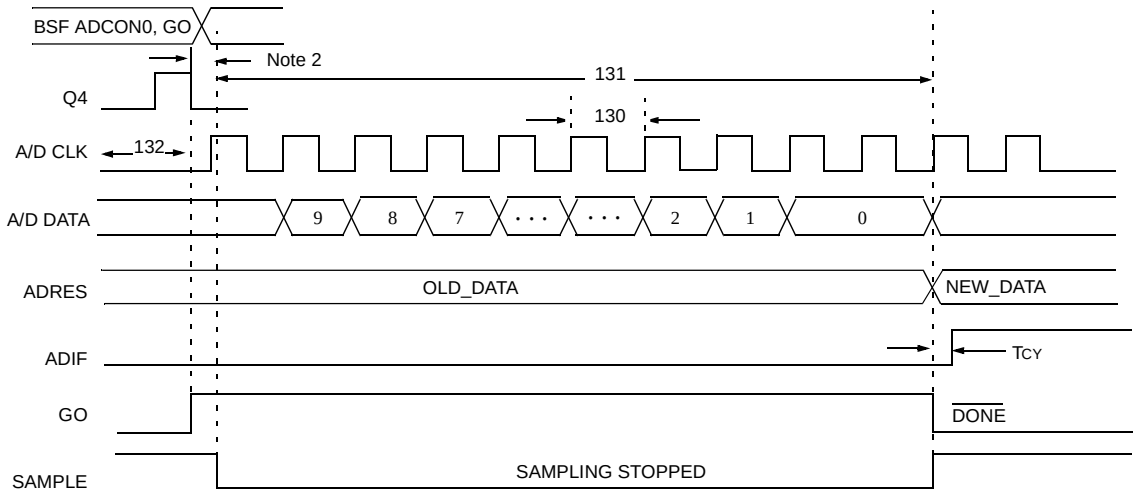


TABLE 25-4: EXTERNAL CLOCK TIMING REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
1A	Fosc	External CLKIN Frequency(1)	DC	40	MHz	XT osc
			DC	40	MHz	HS osc
			4	10	MHz	HS + PLL osc
			DC	40	kHz	LP osc
			DC	40	MHz	EC
		Oscillator Frequency(1)	DC	4	MHz	RC osc
			0.1	4	MHz	XT osc
			4	25	MHz	HS osc
			4	10	MHz	HS + PLL osc
			5	200	kHz	LP osc mode
1	Tosc	External CLKIN Period(1)	250	—	ns	XT and RC osc
			40	—	ns	HS osc
			100	—	ns	HS + PLL osc
			5	—	µs	LP osc
			5	—	ns	EC
		Oscillator Period(1)	250	—	ns	RC osc
			250	10,000	ns	XT osc
			100	10,000	ns	HS osc
			40	100	ns	HS + PLL osc
			5	—	µs	LP osc
2	TCY	Instruction Cycle Time(1)	100	—	ns	TCY = 4/FOSC
3	TosL, TosH	External Clock in (OSC1) High or Low Time	30	—	ns	XT osc
			2.5	—	ns	LP osc
			10	—	µs	HS osc
4	TosR, TosF	External Clock in (OSC1) Rise or Fall Time	—	20	ns	XT osc
			—	50	ns	LP osc
			—	7.5	ns	HS osc

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

FIGURE 25-22: A/D CONVERSION TIMING



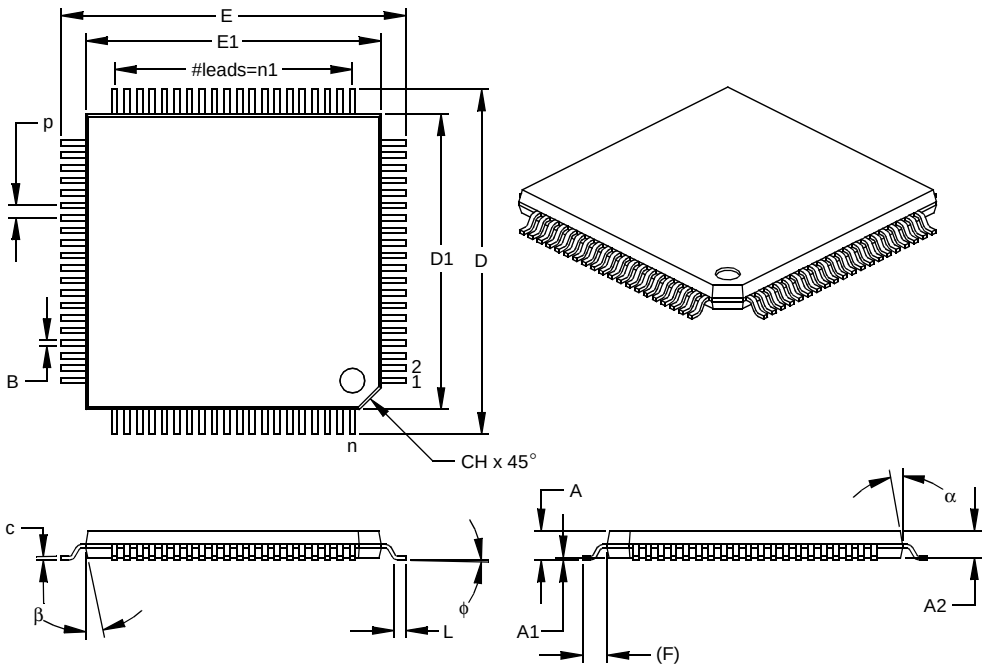
Note 1: If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts.This allows the SLEEP instruction to be executed.
2: This is a minimal RC delay (typically 100 ns), which also disconnects the holding capacitor from the analog input.

TABLE 25-22: A/D CONVERSION REQUIREMENTS

Param No.	Sym-bol	Characteristic		Min	Max	Units	Conditions
130	TAD	A/D clock period	PIC18**C**XX8	1.6	20(5)	µs	TOSC based, VREF ≥ 3.0V
			PIC18**LC**XX8	3.0	20(5)	µs	TOSC based, VREF full range
			PIC18**C**XX8	2.0	6.0	µs	A/D RC mode
			PIC18**LC**XX8	3.0	9.0	µs	A/D RC mode
131	TCNV	Conversion time (not including acquisition time)(1)		11	12	TAD	
132	TACQ	Acquisition time(3)		15 10	— —	µs µs	-40°C ≤ Temp ≤ 125°C 0°C ≤ Temp ≤ 125°C
135	TSWC	Switching time from convert → sample		—	**(Note 4)**		
136	TAMP	Amplifier settling time (Note 2)		1	—	µs	This may be used if the "new" input voltage has not changed by more than 1 LSb (i.e., 5 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).

Note 1: ADRES register may be read on the following TCY cycle.
2: See Section 18.0 for minimum conditions, when input voltage has changed more than 1 LSb.
3: The time for the holding capacitor to acquire the "New" input voltage, when the voltage changes full scale after the conversion (AVDD to AVSS, or AVSS to AVDD). The source impedance (*RS*) on the input channels is 50 Ω.
4: On the next Q4 cycle of the device clock.
5: The time of the A/D clock period is dependent on the device frequency and the TAD clock divider.

80-Lead Plastic Thin Quad Flatpack (PT) 12x12x1 mm Body, 1.0/0.10 mm Lead Form (TQFP)



Units		INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		80			80	
Pitch	p		.020			0.50	
Pins per Side	n1		20			20	
Overall Height	A	.039	.043	.047	1.00	1.10	1.20
Molded Package Thickness	A2	.037	.039	.041	0.95	1.00	1.05
Standoff §	A1	.002	.004	.006	0.05	0.10	0.15
Foot Length	L	.018	.024	.030	0.45	0.60	0.75
Footprint (Reference)	(F)		.039			1.00	
Foot Angle	φ	0	3.5	7	0	3.5	7
Overall Width	E	.541	.551	.561	13.75	14.00	14.25
Overall Length	D	.541	.551	.561	13.75	14.00	14.25
Molded Package Width	E1	.463	.472	.482	11.75	12.00	12.25
Molded Package Length	D1	.463	.472	.482	11.75	12.00	12.25
Lead Thickness	c	.004	.006	.008	0.09	0.15	0.20
Lead Width	B	.007	.009	.011	0.17	0.22	0.27
Pin 1 Corner Chamfer	CH	.025	.035	.045	0.64	0.89	1.14
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-026
Drawing No. C04-092

NOTES:

