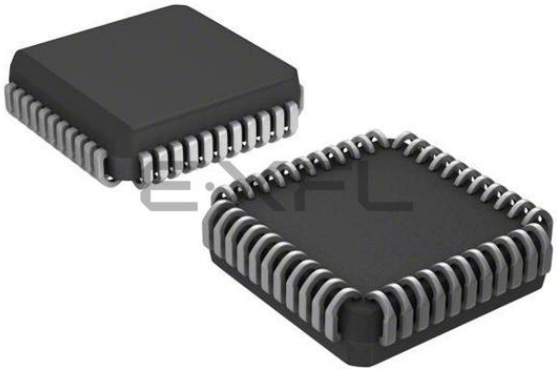


Welcome to E-XFL.COM

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	C251
Core Size	8/16-Bit
Speed	24MHz
Connectivity	EBI/EMI, I²C, Microwire, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	32
Program Memory Size	32KB (32K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	44-PLCC (16.6x16.6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/tsc87251g2d-24ib

- **Typical Operating Current:11 mA at 3V**
- **Typical Power-down Current: 1 µA**
- **Temperature Ranges: Commercial (0°C to +70°C), Industrial (-40°C to +85°C), Automotive ((-40°C to +85°C) ROM only)**
- **Option: Extended Range (-55°C to +125°C)**
- **Packages: PDIL 40, PLCC 44 and VQFP 44**
- **Options: Known Good Dice and Ceramic Packages**

Description

The TSC80251G2D products are derivatives of the Atmel Microcontroller family based on the 8/16-bit C251 Architecture. This family of products is tailored to 8/16-bit microcontroller applications requiring an increased instruction throughput, a reduced operating frequency or a larger addressable memory space. The architecture can provide a significant code size reduction when compiling C programs while fully preserving the legacy of C51 assembly routines.

The TSC80251G2D derivatives are pin and software compatible with standard 80C51/Fx/Rx/Rx+ with extended on-chip data memory (1 Kbyte RAM) and up to 256 kilobytes of external code and data. Additionally, the TSC83251G2D and TSC87251G2D provide on-chip code memory: 32 kilobytes ROM and 32 kilobytes EPROM/OTPROM respectively.

They provide transparent enhancements to Intel's 8xC251Sx family with an additional Synchronous Serial Link Controller (SSLC supporting TWI, µWire and SPI protocols), a Keyboard interrupt interface, a dedicated Baud Rate Generator for UART, and Power Management features.

TSC80251G2D derivatives are optimized for speed and for low power consumption on a wide voltage range.

Note: 1. This Datasheet provides the technical description of the TSC80251G2D derivatives. For further information on the device usage, please request the TSC80251 Programmer's Guide and the TSC80251G1D Design Guide and errata sheet.

Typical Applications

- ISDN Terminals
- High-Speed Modems
- PABX (SOHO)
- Line Cards
- DVD ROM and Players
- Printers
- Plotters
- Scanners
- Banking Machines
- Barcode Readers
- Smart Cards Readers
- High-End Digital Monitors
- High-End Joysticks
- High-end TV's

Table 2. Product Name Signal Description (Continued)

Signal Name	Type	Description	Alternate Function
T1:0	I/O	**Timer 1:0 External Clock Inputs** When timer 1:0 operates as a counter, a falling edge on the T1:0 pin increments the count.	–
T2	I/O	**Timer 2 Clock Input/Output** For the timer 2 capture mode, T2 is the external clock input. For the Timer 2 clock-out mode, T2 is the clock output.	P1.0
T2EX	I	**Timer 2 External Input** In timer 2 capture mode, a falling edge initiates a capture of the timer 2 registers. In auto-reload mode, a falling edge causes the timer 2 register to be reloaded. In the up-down counter mode, this signal determines the count direction: 1 = up, 0 = down.	P1.1
TXD	O	**Transmit Serial Data** TXD outputs the shift clock in serial I/O mode 0 and transmits data in serial I/O modes 1, 2 and 3.	P3.1
VDD	PWR	**Digital Supply Voltage** Connect this pin to +5V or +3V supply voltage.	–
VPP	I	**Programming Supply Voltage** The programming supply voltage is applied to this input for programming the on-chip EPROM/OTPROM.	–
VSS	GND	**Circuit Ground** Connect this pin to ground.	–
VSS1	GND	**Secondary Ground 1** This ground is provided to reduce ground bounce and improve power supply bypassing. Connection of this pin to ground is recommended. However, when using the TSC80251G2D as a pin-for-pin replacement for a 8xC51 product, VSS1 can be unconnected without loss of compatibility. Not available on DIP package.	–
VSS2	GND	**Secondary Ground 2** This ground is provided to reduce ground bounce and improve power supply bypassing. Connection of this pin to ground is recommended. However, when using the TSC80251G2D as a pin-for-pin replacement for a 8xC51 product, VSS2 can be unconnected without loss of compatibility. Not available on DIP package.	–
WAIT#	I	**Real-time Synchronous Wait States Input** The real-time WAIT# input is enabled by setting RTWE bit in WCON (S:A7h). During bus cycles, the external memory system can signal 'system ready' to the microcontroller in real time by controlling the WAIT# input signal.	P1.6
WCLK	O	**Wait Clock Output** The real-time WCLK output is enabled by setting RTWCE bit in WCON (S:A7h). When enabled, the WCLK output produces a square wave signal with a period of one half the oscillator frequency.	P1.7
WR#	O	**Write** Write signal output to external memory.	P3.6
XTAL1	I	**Input to the on-chip inverting oscillator amplifier** To use the internal oscillator, a crystal/resonator circuit is connected to this pin. If an external oscillator is used, its output is connected to this pin. XTAL1 is the clock source for internal timing.	–

Table 16. Notation for Immediate Addressing

Immediate Address	Description	C251	C51
#data	An 8-bit constant that is immediately addressed in an instruction	3	3
#data16	A 16-bit constant that is immediately addressed in an instruction	3	–
#0data16 #1data16	A 32-bit constant that is immediately addressed in an instruction. The upper word is filled with zeros (#0data16) or ones (#1data16).	3	–
#short	A constant, equal to 1, 2, or 4, that is immediately addressed in an instruction.	3	–

Table 17. Notation for Bit Addressing

Direct Address	Description	C251	C51
bit51	A directly addressed bit (bit number = 00h-FFh) in memory or an SFR. Bits 00h-7Fh are the 128 bits in byte locations 20h-2Fh in the on-chip RAM. Bits 80h-FFh are the 128 bits in the 16 SFRs with addresses that end in 0h or 8h, S:80h, S:88h, S:90h,..., S:F0h, S:F8h.	–	3
bit	A directly addressed bit in memory locations 00:0020h-00:007Fh or in any defined SFR.	3	

Table 18. Notation for Destination in Control Instructions

Direct Address	Description	C251	C51
rel	A signed (two's complement) 8-bit relative address. The destination is -128 to +127 bytes relative to the next instruction's first byte.	3	3
addr11	An 11-bit target address. The target is in the same 2-Kbyte block of memory as the next instruction's first byte.	–	3
addr16	A 16-bit target address. The target can be anywhere within the same 64-Kbyte region as the next instruction's first byte.	–	3
addr24	A 24-bit target address. The target can be anywhere within the 16-Mbyte address space.	3	–

Table 22. Summary of Compare Instructions

CompareCMP <dest>, <src>dest opnd - src opnd						
			Binary Mode		Source Mode	
Mnemonic	**<dest>, <src>(2)**	**Comments**	**Bytes**	**States**	**Bytes**	**States**
CMP	Rmd, Rms	Register with register	3	2	2	1
	WRjd, WRjs	Word register with word register	3	3	2	2
	DRkd, DRks	Dword register with dword register	3	5	2	4
	Rm, #data	Register with immediate data	4	3	3	2
	WRj, #data16	Word register with immediate 16-bit data	5	4	4	3
	DRk, #0data16	Dword register with zero-extended 16-bit immediate data	5	6	4	5
	DRk, #1data16	Dword register with one-extended 16-bit immediate data	5	6	4	5
	Rm, dir8	Direct address (on-chip RAM or SFR) with byte register	4	3(1)	3	2(1)
	WRj, dir8	Direct address (on-chip RAM or SFR) with word register	4	4	3	3
	Rm, dir16	Direct address (64K) with byte register	5	3(2)	4	2(2)
	WRj, dir16	Direct address (64K) with word register	5	4(3)	4	3(3)
	Rm, at WRj	Indirect address (64K) with byte register	4	3(2)	3	2(2)
	Rm, at DRk	Indirect address (16M) with byte register	4	4(2)	3	3(2)

Notes:
1. If this instruction addresses an I/O Port (Px, x = 0-3), add 1 to the number of states. Add 2 if it addresses a Peripheral SFR.
2. If this instruction addresses external memory location, add N+2 to the number of states (N: number of wait states).
3. If this instruction addresses external memory location, add 2(N+2) to the number of states (N: number of wait states).

Logical AND[1]ANL <dest>, <src>dest opnd ← dest opnd Λ src opnd
Logical OR[1]ORL <dest>, <src>dest opnd ← dest opnd ς src opnd
Logical Exclusive OR[1]XRL <dest>, <src>dest opnd ← dest opnd ∀ src opnd
Clear[1]CLR A(A) ← 0
Complement[1]CPL A(A) ← ∅ (A)
Rotate LeftRL $A(A)_{n+1} \leftarrow (A)_n$, n = 0..6
$(A)_0 \leftarrow (A)_7$
Rotate Left CarryRLC $A(A)_{n+1} \leftarrow (A)_n$, n = 0..6
$(CY) \leftarrow (A)_7$
$(A)_0 \leftarrow (CY)$
Rotate RightRR $A(A)_{n-1} \leftarrow (A)_n$, n = 7..1
$(A)_7 \leftarrow (A)_0$
Rotate Right CarryRRC $A(A)_{n-1} \leftarrow (A)_n$, n = 7..1
$(CY) \leftarrow (A)_0$
$(A)_7 \leftarrow (CY)$

Mnemonic	<dest>, <src>[1]	Comments	Binary Mode		Source Mode	
			Bytes	States	Bytes	States
ANL ORL XRL	A, Rn	register to ACC	1	1	2	2
	A, dir8	Direct address (on-chip RAM or SFR) to ACC	2	1[3]	2	1[3]
	A, at Ri	Indirect address to ACC	1	2	2	3
	A, #data	Immediate data to ACC	2	1	2	1
	dir8, A	ACC to direct address	2	2[4]	2	2[4]
	dir8, #data	Immediate 8-bit data to direct address	3	3[4]	3	3[4]
	Rmd, Rms	Byte register to byte register	3	2	2	1
	WRjd, WRjs	Word register to word register	3	3	2	2
	Rm, #data	Immediate 8-bit data to byte register	4	3	3	2
	WRj, #data16	Immediate 16-bit data to word register	5	4	4	3
	Rm, dir8	Direct address (on-chip RAM or SFR) to byte register	4	3[3]	3	2[3]
	WRj, dir8	Direct address (on-chip RAM or SFR) to word register	4	4	3	3
	Rm, dir16	Direct address (64K) to byte register	5	3[5]	4	2[5]
	WRj, dir16	Direct address (64K) to word register	5	4[6]	4	3[6]
	Rm, at WRj	Indirect address (64K) to byte register	4	3[5]	3	2[5]
	Rm, at DRk	Indirect address (16M) to byte register	4	4[5]	3	3[5]
CLR	A	Clear ACC	1	1	1	1
CPL	A	Complement ACC	1	1	1	1
RL	A	Rotate ACC left	1	1	1	1
RLC	A	Rotate ACC left through CY	1	1	1	1
RR	A	Rotate ACC right	1	1	1	1
RRC	A	Rotate ACC right through CY	1	1	1	1

Table 26. Summary of Move Instructions (2/3)

Move[1] MOV <dest>, <src> dest opnd ← src opnd						
			Binary Mode		Source Mode	
Mnemonic	<dest>, <src>[2]	Comments	Bytes	States	Bytes	States
MOV	A, Rn	Register to ACC	1	1	2	2
	A, dir8	Direct address (on-chip RAM or SFR) to ACC	2	1[3]	2	1[3]
	A, at Ri	Indirect address to ACC	1	2	2	3
	A, #data	Immediate data to ACC	2	1	2	1
	Rn, A	ACC to register	1	1	2	2
	Rn, dir8	Direct address (on-chip RAM or SFR) to register	2	1[3]	3	2[3]
	Rn, #data	Immediate data to register	2	1	3	2
	dir8, A	ACC to direct address (on-chip RAM or SFR)	2	2[3]	2	2[3]
	dir8, Rn	Register to direct address (on-chip RAM or SFR)	2	2[3]	3	3[3]
	dir8, dir8	Direct address to direct address (on-chip RAM or SFR)	3	3[4]	3	3[4]
	dir8, at Ri	Indirect address to direct address (on-chip RAM or SFR)	2	3[3]	3	4[3]
	dir8, #data	Immediate data to direct address (on-chip RAM or SFR)	3	3[3]	3	3[3]
	at Ri, A	ACC to indirect address	1	3	2	4
	at Ri, dir8	Direct address (on-chip RAM or SFR) to indirect address	2	3[3]	3	4[3]
	at Ri, #data	Immediate data to indirect address	2	3	3	4
	DPTR, #data16	Load Data Pointer with a 16-bit constant	3	2	3	2

Notes:
1. Instructions that move bits are in Table 27.
2. Move instructions from the C51 Architecture.
3. If this instruction addresses an I/O Port (Px, x = 0-3), add 1 to the number of states. Add 2 if it addresses a Peripheral SFR.
4. Apply note 3 for each dir8 operand.

Move[1] MOV <dest>, <src> dest opnd ← src opnd						
			Binary Mode		Source Mode	
Mnemonic	<dest>, <src>[1]	Comments	Bytes	States	Bytes	States
MOV	Rmd, Rms	Byte register to byte register	3	2	2	1
MOV	WRjd, WRjs	Word register to word register	3	2	2	1
MOV	DRkd, DRks	Dword register to dword register	3	3	2	2
MOV	Rm, #data	Immediate 8-bit data to byte register	4	3	3	2
MOV	WRj, #data16	Immediate 16-bit data to word register	5	3	4	2
MOV	DRk, #0data16	zero-ext 16bit immediate data to dword register	5	5	4	4
MOV	DRk, #1data16	one-ext 16bit immediate data to dword register	5	5	4	4
MOV	Rm, dir8	Direct address (on-chip RAM or SFR) to byte register	4	3[3]	3	2[3]
MOV	WRj, dir8	Direct address (on-chip RAM or SFR) to word register	4	4	3	3
MOV	DRk, dir8	Direct address (on-chip RAM or SFR) to dword register	4	6	3	5
MOV	Rm, dir16	Direct address (64K) to byte register	5	3[4]	4	2[4]
MOV	WRj, dir16	Direct address (64K) to word register	5	4[5]	4	3[5]
MOV	DRk, dir16	Direct address (64K) to dword register	5	6[6]	4	5[6]
MOV	Rm, at WRj	Indirect address (64K) to byte register	4	3[4]	3	2[4]
MOV	Rm, at DRk	Indirect address (16M) to byte register	4	4[4]	3	3[4]
MOV	WRjd, at WRjs	Indirect address (64K) to word register	4	4[5]	3	3[5]
MOV	WRj, at DRk	Indirect address (16M) to word register	4	5[5]	3	4[5]
MOV	dir8, Rm	Byte register to direct address (on-chip RAM or SFR)	4	4[3]	3	3[3]
MOV	dir8, WRj	Word register to direct address (on-chip RAM or SFR)	4	5	3	4
MOV	dir8, DRk	Dword register to direct address (on-chip RAM or SFR)	4	7	3	6
MOV	dir16, Rm	Byte register to direct address (64K)	5	4[4]	4	3[4]
MOV	dir16, WRj	Word register to direct address (64K)	5	5[5]	4	4[5]
MOV	dir16, DRk	Dword register to direct address (64K)	5	7[6]	4	6[6]
MOV	at WRj, Rm	Byte register to indirect address (64K)	4	4[4]	3	3[4]
MOV	at DRk, Rm	Byte register to indirect address (16M)	4	5[4]	3	4[4]
MOV	at WRjd, WRjs	Word register to indirect address (64K)	4	5[5]	3	4[5]
MOV	at DRk, WRj	Word register to indirect address (16M)	4	6[5]	3	5[5]
MOV	Rm, at WRj +dis16	Indirect with 16-bit displacement (64K) to byte register	5	6[4]	4	5[4]
MOV	WRj, at WRj +dis16	Indirect with 16-bit displacement (64K) to word register	5	7[5]	4	6[5]
MOV	Rm, at DRk +dis24	Indirect with 16-bit displacement (16M) to byte register	5	7[4]	4	6[4]

MOV	WRj, at WRj +dis24	Indirect with 16-bit displacement (16M) to word register	5	8[5]	4	7[5]
MOV	at WRj +dis16, Rm	Byte register to indirect with 16-bit displacement (64K)	5	6[4]	4	5[4]
MOV	at WRj +dis16, WRj	Word register to indirect with 16-bit displacement (64K)	5	7[5]	4	6[5]
MOV	at DRk +dis24, Rm	Byte register to indirect with 16-bit displacement (16M)	5	7[4]	4	6[4]
MOV	at DRk +dis24, WRj	Word register to indirect with 16-bit displacement (16M)	5	8[5]	4	7[5]

Notes:
1. Instructions that move bits are in Table 27.
2. Move instructions unique to the C251 Architecture.
3. If this instruction addresses an I/O Port (Px, x = 0-3), add 1 to the number of states. Add 2 if it addresses a Peripheral SFR.
4. If this instruction addresses external memory location, add N+2 to the number of states (N: number of wait states).
5. If this instruction addresses external memory location, add 2(N+1) to the number of states (N: number of wait states).
6. If this instruction addresses external memory location, add 4(N+2) to the number of states (N: number of wait states).

Add 3 if it addresses a Peripheral SFR.

5. If this instruction addresses an I/O Port (Px, x = 0-3), add 3 to the number of states. Add 5 if it addresses a Peripheral SFR.
6. In internal execution only, add 1 to the number of states of the 'jump taken' if the destination address is internal and odd.

Table 31. Summary of Unconditional Jump Instructions

Absolute jumpAJMP <src>(PC) ← (PC) +2; $(PC)_{10:0}$ ← src opnd						
Extended jumpEJMP <src>(PC) ← (PC) + size (instr); $(PC)_{23:0}$ ← src opnd						
Long jumpLJMP <src>(PC) ← (PC) + size (instr); $(PC)_{15:0}$ ← src opnd						
Short jumpSJMP rel(PC) ← (PC) +2; (PC) ← (PC) +rel						
Jump indirectJMP at A +DPTR$(PC)_{23:16}$ ← FFh; $(PC)_{15:0}$ ← (A) + (DPTR)						
No operationNOP(PC) ← (PC) +1						
Mnemonic	**<dest>, <src>(1)**	**Comments**	**Binary Mode**		**Source Mode**	
			Bytes	**States**	**Bytes**	**States**
AJMP	addr11	Absolute jump	2	3(2)(3)	2	3(2)(3)
EJMP	addr24	Extended jump	5	6(2)(4)	4	5(2)(4)
	at DRk	Extended jump (indirect)	3	7(2)(4)	2	6(2)(4)
LJMP	at WRj	Long jump (indirect)	3	6(2)(4)	2	5(2)(4)
	addr16	Long jump (direct address)	3	5(2)(4)	3	5(2)(4)
SJMP	rel	Short jump (relative address)	2	4(2)(4)	2	4(2)(4)
JMP	at A +DPTR	Jump indirect relative to the DPTR	1	5(2)(4)	1	5(2)(4)
NOP		No operation (Jump never)	1	1	1	1

Notes:
1. A shaded cell denotes an instruction in the C51 Architecture.
2. In internal execution only, add 1 to the number of states if the destination address is internal and odd.
3. Add 2 to the number of states if the destination address is external.
4. Add 3 to the number of states if the destination address is external.

Table 32. Summary of Call and Return Instructions

Absolute callACALL <src>(PC) ← (PC) +2; push $(PC)_{15:0}$;
$(PC)_{10:0}$ ← src opnd
Extended callECALL <src>(PC) ← (PC) + size (instr); push $(PC)_{23:0}$;
$(PC)_{23:0}$ ← src opnd
Long callLCALL <src>(PC) ← (PC) + size (instr); push $(PC)_{15:0}$;
$(PC)_{15:0}$ ← src opnd
Return from subroutineRETpop $(PC)_{15:0}$
Extended return from subroutineERETpop $(PC)_{23:0}$
Return from interruptRETIIF [INTR = 0] THEN pop $(PC)_{15:0}$
IF [INTR = 1] THEN pop $(PC)_{23:0}$; pop (PSW1)
Trap interruptTRAP(PC) ← (PC) + size (instr);
IF [INTR = 0] THEN push $(PC)_{15:0}$
IF [INTR = 1] THEN push (PSW1); push $(PC)_{23:0}$

Mnemonic	<dest>, <src>(1)	Comments	Binary Mode		Source Mode	
			Bytes	States	Bytes	States
ACALL	addr11	Absolute subroutine call	2	9(2)(3)	2	9(2)(3)
ECALL	at DRk	Extended subroutine call (indirect)	3	14(2)(3)	2	13(2)(3)
	addr24	Extended subroutine call	5	14(2)(3)	4	13(2)(3)
LCALL	at WRj	Long subroutine call (indirect)	3	10(2)(3)	2	9(2)(3)
	addr16	Long subroutine call	3	9(2)(3)	3	9(2)(3)
RET		Return from subroutine	1	7(2)	1	7(2)
ERET		Extended subroutine return	3	9(2)	2	8(2)
RETI		Return from interrupt	1	7(2)(4)	1	7(2)(4)
TRAP		Jump to the trap interrupt vector	2	12(4)	1	11(4)

Notes:
1. A shaded cell denotes an instruction in the C51 Architecture.
2. In internal execution only, add 1 to the number of states if the destination/return address is internal and odd.
3. Add 2 to the number of states if the destination address is external.
4. Add 5 to the number of states if INTR = 1.

To preserve the secrecy of the encryption key byte sequence, the Encryption Array can not be verified.

Notes:
1. When a MOVC instruction is executed, the content of the ROM is not encrypted. In order to fully protect the user program code, the lock bit level 1 (see Table 33) must always be set when encryption is used.
2. If the encryption feature is implemented, the portion of the on-chip code memory that does not contain program code should be filled with “random” byte values to prevent the encryption key sequence from being revealed.

Signature Bytes

The TSC80251G2D derivatives contain factory-programmed Signature Bytes. These bytes are located in non-volatile memory outside the memory address space at 30h, 31h, 60h and 61h. To read the Signature Bytes, perform the procedure described in section Verify Algorithm, using the verify signature mode (see Table 37). Signature byte values are listed in Table 35.

Table 35. Signature Bytes (Electronic ID)

		Signature Address	Signature Data
Vendor	Atmel	30h	58h
Architecture	C251	31h	40h
Memory	32 kilobytes EPROM or OTPROM	60h	F7h
	32 kilobytes MaskROM or ROMless		77h
Revision	TSC80251G2D derivative	61h	FDh

Programming Algorithm

Figure 6 shows the hardware setup needed to program the TSC87251G2D EPROM/OTPROM areas:

- The chip has to be put under reset and maintained in this state until completion of the programming sequence.
- PSEN# and the other control signals (ALE and Port 0) have to be set to a high level.
- Then PSEN# has to be to forced to a low level after two clock cycles or more and it has to be maintained in this state until the completion of the programming sequence (see below).
- The voltage on the EA# pin must be set to V_{DD}.
- The programming mode is selected according to the code applied on Port 0 (see Table 36). It has to be applied until the completion of this programming operation.
- The programming address is applied on Ports 1 and 3 which are respectively the Most Significant Byte (MSB) and the Least Significant Byte (LSB) of the address.
- The programming data are applied on Port 2.
- The EPROM Programming is done by raising the voltage on the EA# pin to V_{PP}, then by generating a low level pulse on ALE/PROG# pin.
- The voltage on the EA# pin must be lowered to V_{DD} before completing the programming operation.
- It is possible to alternate programming and verifying operation (See Paragraph Verify Algorithm). Please make sure the voltage on the EA# pin has actually been lowered to V_{DD} before performing the verifying operation.

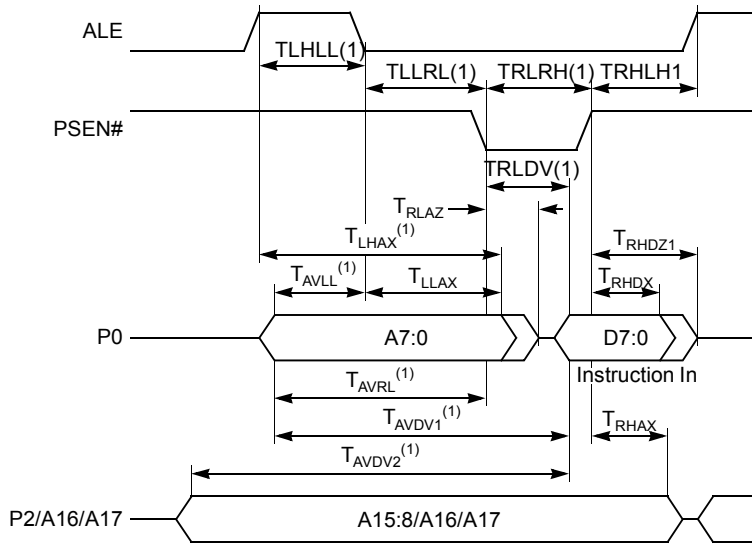
Table 39. Bus Cycles AC Timings; V_{DD} = 4.5 to 5.5 V, T_A = -40 to 85°C

Symbol	Parameter	12 MHz		16 MHz		24 MHz		Unit
		Min	Max	Min	Max	Min	Max	
T_{OSC}	$1/F_{OSC}$	83		62		41		ns
T_{LHLL}	ALE Pulse Width	78		58		38		ns(2)
T_{AVLL}	Address Valid to ALE Low	78		58		37		ns(2)
T_{LLAX}	Address hold after ALE Low	19		11		3		ns
T_{RLRH}(1)	RD#/PSEN# Pulse Width	162		121		78		ns(3)
T_{WLWH}	WR# Pulse Width	165		124		81		ns(3)
T_{LLRL}(1)	ALE Low to RD#/PSEN# Low	22		14		6		ns
T_{LHAX}	ALE High to Address Hold	99		70		40		ns(2)
T_{RLDV}(1)	RD#/PSEN# Low to Valid Data		146		104		61	ns(3)
T_{RHDX}(1)	Data Hold After RD#/PSEN# High	0		0		0		ns
T_{RHAX}(1)	Address Hold After RD#/PSEN# High	0		0		0		ns
T_{RLAZ}(1)	RD#/PSEN# Low to Address Float		0		0		0	ns
T_{RHDZ1}	Instruction Float After RD#/PSEN# High		45		40		30	ns
T_{RHDZ2}	Data Float After RD#/PSEN# High		215		165		115	ns
T_{RHLH1}	RD#/PSEN# high to ALE High (Instruction)	49		43		31		ns
T_{RHLH2}	RD#/PSEN# high to ALE High (Data)	215		169		115		ns
T_{WHLH}	WR# High to ALE High	215		169		115		ns
T_{AVDV1}	Address (P0) Valid to Valid Data In		250		175		105	ns(2)(3)
T_{AVDV2}	Address (P2) Valid to Valid Data In		306		223		140	ns(2)(3)
T_{AVDV3}	Address (P0) Valid to Valid Instruction In		150		109		68	ns(3)
T_{AXDX}	Data Hold after Address Hold	0		0		0		ns
T_{AVRL}(1)	Address Valid to RD# Low	100		70		40		ns(2)
T_{AVWL1}	Address (P0) Valid to WR# Low	100		70		40		ns(2)
T_{AVWL2}	Address (P2) Valid to WR# Low	158		115		74		ns(2)
T_{WHQX}	Data Hold after WR# High	90		69		32		ns
T_{QVWH}	Data Valid to WR# High	133		102		72		ns(3)
T_{WHAX}	WR# High to Address Hold	167		125		84		ns

Notes:
1. Specification for PSEN# are identical to those for RD#.
2. If a wait state is added by extending ALE, add $2 \cdot T_{OSC.}$
3. If wait states are added by extending RD#/PSEN#/WR#, add $2N \cdot T_{OSC}$ (N = 1..3).

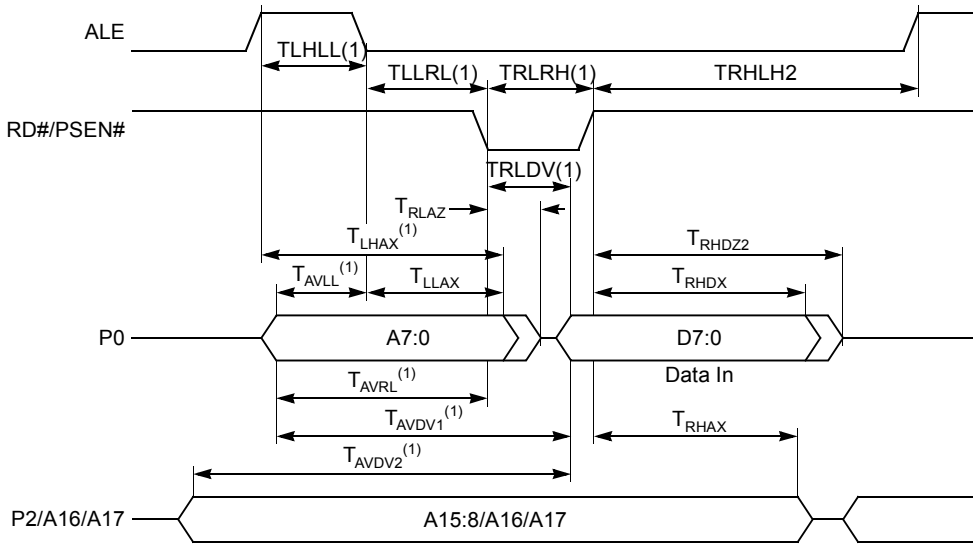
Waveforms in Non-Page Mode

Figure 8. External Bus Cycle: Code Fetch (Non-Page Mode)



Note: 1. The value of this parameter depends on wait states. See Table 39 and Table 40.

Figure 9. External Bus Cycle: Data Read (Non-Page Mode)



Note: 1. The value of this parameter depends on wait states. See Table 39 and Table 40.

Timings

Table 42. Real-Time Synchronous Wait AC Timings; V_{DD} = 2.7 to 5.5 V, T_A = -40 to 85°C

Symbol	Parameter	Min	Max	Unit
T_{CLYV}	Wait Clock Low to Wait Set-up	0	T_{OSC} - 20	ns
T_{CLYX}	Wait Hold after Wait Clock Low	2W·T_{OSC} + 5	(1+2W)·T_{OSC} - 20	ns
T_{RLYV}	PSEN#/RD# Low to Wait Set-up	0	T_{OSC} - 20	ns
T_{RLYX}	Wait Hold after PSEN#/RD# Low	2W·T_{OSC} + 5	(1+2W)·T_{OSC} - 20	ns
T_{WLYV}	WR# Low to Wait Set-up	0	T_{OSC} - 20	ns
T_{WLYX}	Wait Hold after WR# Low	2W·T_{OSC} + 5	(1+2W)·T_{OSC} - 20	ns

Waveforms

Figure 14. Real-time Synchronous Wait State: Code Fetch/Data Read

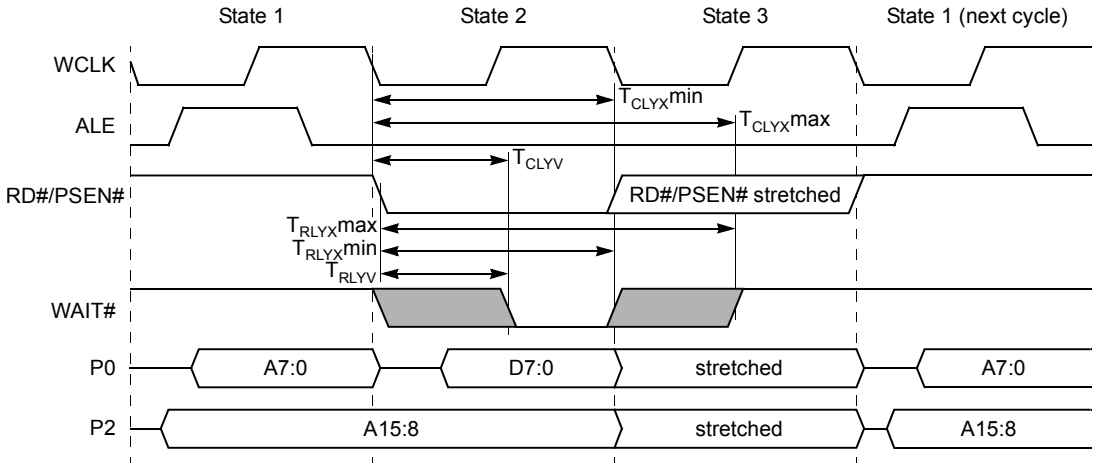
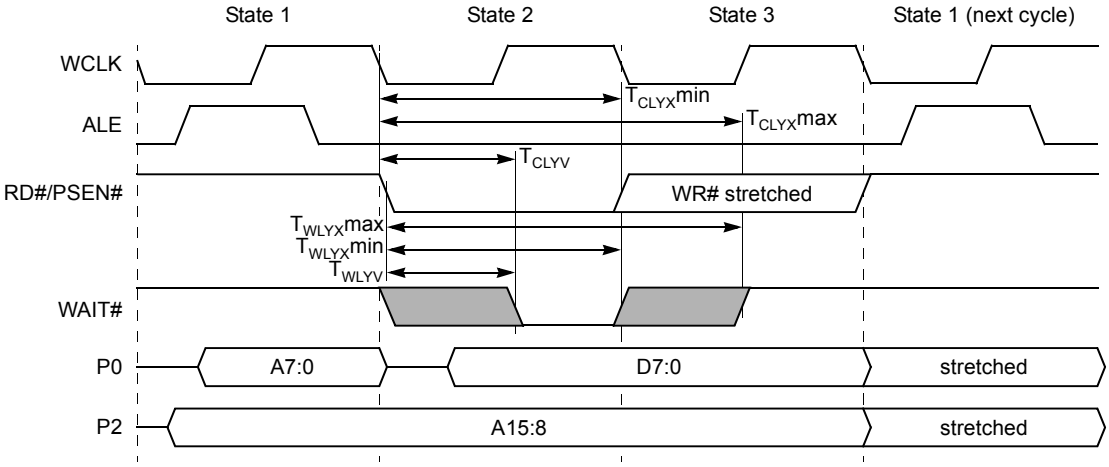


Figure 15. Real-time Synchronous Wait State: Data Write



AC Characteristics - Real-Time Asynchronous Wait State

Definition of Symbols

Table 43. Real-Time Asynchronous Wait Timing Symbol Definitions

Signals	
S	PSEN#/RD#/WR#
Y	AWAIT#

Conditions	
L	Low
V	Valid
X	No Longer Valid

Timings

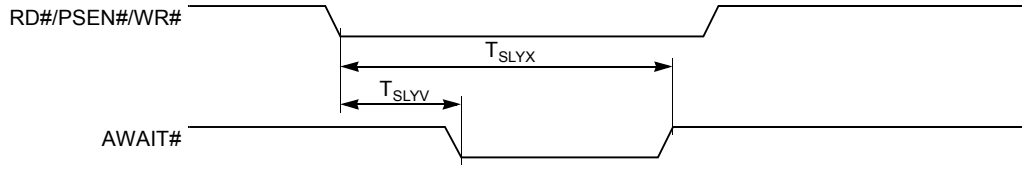
Table 44. Real-Time Asynchronous Wait AC Timings; V_{DD} = 2.7 to 5.5 V, T_A = -40 to 85°C

Symbol	Parameter	Min	Max	Unit
T_{SLYV}	PSEN#/RD#/WR# Low to Wait Set-up		T_{OSC} - 10	ns
T_{SLYX}	Wait Hold after PSEN#/RD#/WR# Low	$(2N-1)\cdot T_{OSC}$ + 10		ns(1)

Note: 1. N is the number of wait states added (N≥ 1).

Waveforms

Figure 16. Real-time Asynchronous Wait State Timings



AC Characteristics - Serial Port in Shift Register Mode

Definition of Symbols

Table 45. Serial Port Timing Symbol Definitions

Signals	
D	Data In
Q	Data Out
X	Clock

Conditions	
H	High
L	Low
V	Valid
X	No Longer Valid

AC Characteristics - SSLC: TWI Interface

Timings

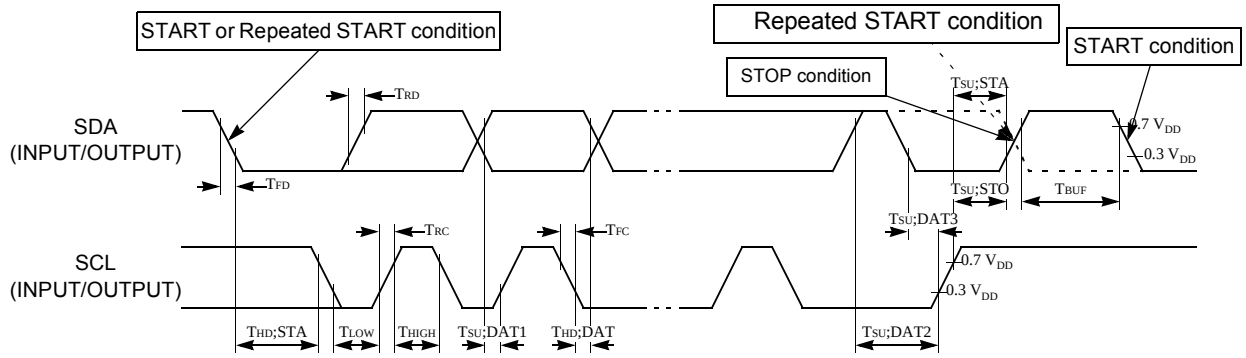
Table 47. TWI Interface AC Timing; V_{DD} = 2.7 to 5.5 V, T_A = -40 to 85°C

Symbol	Parameter	INPUT Min	INPUT Max	OUTPUT Min	OUTPUT Max
T_{HD}; STA	Start condition hold time	14·T_{CLCL}(4)		4.0 µs(1)	
T_{LOW}	SCL low time	16·T_{CLCL}(4)		4.7 µs(1)	
T_{HIGH}	SCL high time	14·T_{CLCL}(4)		4.0 µs(1)	
T_{RC}	SCL rise time	1 µs		-(2)	
T_{FC}	SCL fall time	0.3 µs		0.3 µs(3)	
T_{SU}; DAT1	Data set-up time	250 ns		20·T_{CLCL}(4)- T_{RD}	
T_{SU}; DAT2	SDA set-up time (before repeated START condition)	250 ns		1 µs(1)	
T_{SU}; DAT3	SDA set-up time (before STOP condition)	250 ns		8·T_{CLCL}(4)	
T_{HD}; DAT	Data hold time	0 ns		8·T_{CLCL}(4) - T_{FC}	
T_{SU}; STA	Repeated START set-up time	14·T_{CLCL}(4)		4.7 µs(1)	
T_{SU}; STO	STOP condition set-up time	14·T_{CLCL}(4)		4.0 µs(1)	
T_{BUF}	Bus free time	14·T_{CLCL}(4)		4.7 µs(1)	
T_{RD}	SDA rise time	1 µs		-(2)	
T_{FD}	SDA fall time	0.3 µs		0.3 µs(3)	

Notes:
1. At 100 kbit/s. At other bit-rates this value is inversely proportional to the bit-rate of 100 kbit/s.
2. Determined by the external bus-line capacitance and the external bus-line pull-up resistor, this must be < 1 µs.
3. Spikes on the SDA and SCL lines with a duration of less than 3·T_{CLCL} will be filtered out. Maximum capacitance on bus-lines SDA and SCL = 400 pF.
4. T_{CLCL} = T_{OSC} = one oscillator clock period.

Waveforms

Figure 18. TWI Waveforms



Timings

Table 49. SPI Interface AC Timing; V_{DD} = 2.7 to 5.5 V, T_A = -40 to 85°C

Symbol	Parameter	Min	Max	Unit
Slave Mode(1)				
T_{CHCH}	Clock Period	8		T_{OSC}
T_{CHCX}	Clock High Time	3.2		T_{OSC}
T_{CLCX}	Clock Low Time	3.2		T_{OSC}
T_{SLCH}, T_{SLCL}	SS# Low to Clock edge	200		ns
T_{IVCL}, T_{IVCH}	Input Data Valid to Clock Edge	100		ns
T_{CLIX}, T_{CHIX}	Input Data Hold after Clock Edge	100		ns
$T_{CLOV,}$ T_{CHOV}	Output Data Valid after Clock Edge		100	ns
T_{CLOX}, T_{CHOX}	Output Data Hold Time after Clock Edge	0		ns
T_{CLSH}, T_{CHSH}	SS# High after Clock Edge	0		ns
T_{IVCL}, T_{IVCH}	Input Data Valid to Clock Edge	100		ns
T_{CLIX}, T_{CHIX}	Input Data Hold after Clock Edge	100		ns
T_{SLOV}	SS# Low to Output Data Valid		130	ns
T_{SHOX}	Output Data Hold after SS# High		130	ns
T_{SHSL}	SS# High to SS# Low	(2)		
T_{ILIH}	Input Rise Time		2	µs
T_{IHIL}	Input Fall Time		2	µs
T_{OLOH}	Output Rise time		100	ns
T_{OHOL}	Output Fall Time		100	ns
Master Mode(3)				
T_{CHCH}	Clock Period	4		T_{OSC}
T_{CHCX}	Clock High Time	1.6		T_{OSC}
T_{CLCX}	Clock Low Time	1.6		T_{OSC}
T_{IVCL}, T_{IVCH}	Input Data Valid to Clock Edge	50		ns
T_{CLIX}, T_{CHIX}	Input Data Hold after Clock Edge	50		ns
$T_{CLOV,}$ T_{CHOV}	Output Data Valid after Clock Edge		65	ns
T_{CLOX}, T_{CHOX}	Output Data Hold Time after Clock Edge	0		ns
T_{ILIH}	Input Data Rise Time		2	µs
T_{IHIL}	Input Data Fall Time		2	µs
T_{OLOH}	Output Data Rise time		50	ns
T_{OHOL}	Output Data Fall Time		50	ns

Notes: 1. Capacitive load on all pins = 200 pF in slave mode.
2. The value of this parameter depends on software.
3. Capacitive load on all pins = 100 pF in master mode.

Timings

Table 51. EPROM Programming AC timings; V_{DD} = 4.5 to 5.5 V, T_A = 0 to 40°C

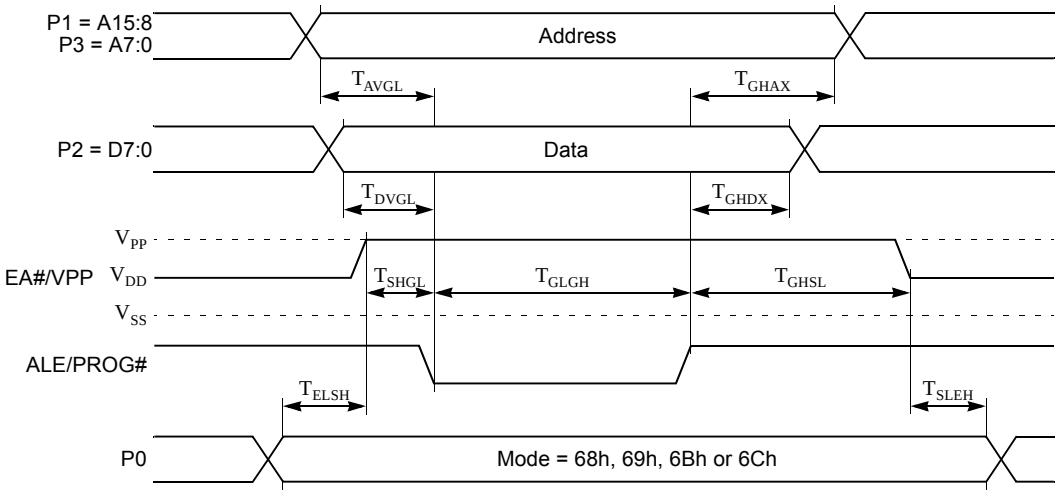
Symbol	Parameter	Min	Max	Unit
T_{OSC}	XTAL1 Period	83.5	250	ns
T_{AVGL}	Address Setup to PROG# low	48		T_{OSC}
T_{GHAX}	Address Hold after PROG# low	48		T_{OSC}
T_{DVGL}	Data Setup to PROG# low	48		T_{OSC}
T_{GHDX}	Data Hold after PROG#	48		T_{OSC}
T_{ELSH}	ENABLE High to V_{PP}	48		T_{OSC}
T_{SHGL}	V_{PP} Setup to PROG# low	10		µs
T_{GHSL}	V_{PP} Hold after PROG#	10		µs
T_{SLEH}	ENABLE Hold after V_{PP}	0		ns
T_{GLGH}	PROG# Width	90	110	µs

Table 52. EPROM Verifying AC timings; V_{DD} = 4.5 to 5.5 V, V_{DD} = 2.7 to 5.5 V, T_A = 0 to 40°C

Symbol	Parameter	Min	Max	Unit
T_{OSC}	XTAL1 Period	83.5	250	ns
T_{AVQV}	Address to Data Valid		48	T_{OSC}
T_{AXQX}	Address to Data Invalid	0		ns
T_{ELQV}	ENABLE low to Data Valid	0	48	T_{OSC}
T_{EHQZ}	Data Float after ENABLE	0	48	T_{OSC}

Waveforms

Figure 23. EPROM Programming Waveforms



Packages

List of Packages

- PDIL 40
- CDIL 40 with window
- PLCC 44
- CQPJ 44 with window
- VQFP 44 (10x10)

PDIL 40 - Mechanical Outline

Figure 33. Plastic Dual In Line

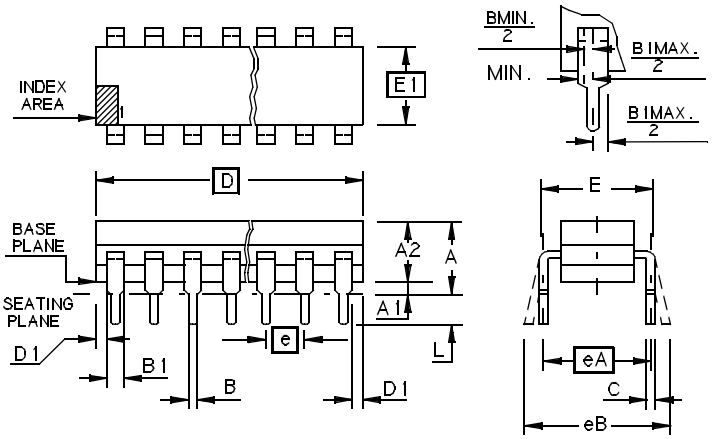


Table 57. PDIL Package Size

	MM		Inch	
	Min	**Max**	**Min**	**Max**
A	-	5.08	-	.200
A1	0.38	-	.015	-
A2	3.18	4.95	.125	.195
B	0.36	0.56	.014	.022
B1	0.76	1.78	.030	.070
C	0.20	0.38	.008	.015
D	50.29	53.21	1.980	2.095
E	15.24	15.87	.600	.625
E1	12.32	14.73	.485	.580
e	2.54 B.S.C.		.100 B.S.C.	
eA	15.24 B.S.C.		.600 B.S.C.	
eB	-	17.78	-	.700
L	2.93	3.81	.115	.150
D1	0.13	-	.005	-

CDIL 40 with Window - Mechanical Outline

Figure 34. Ceramic Dual In Line

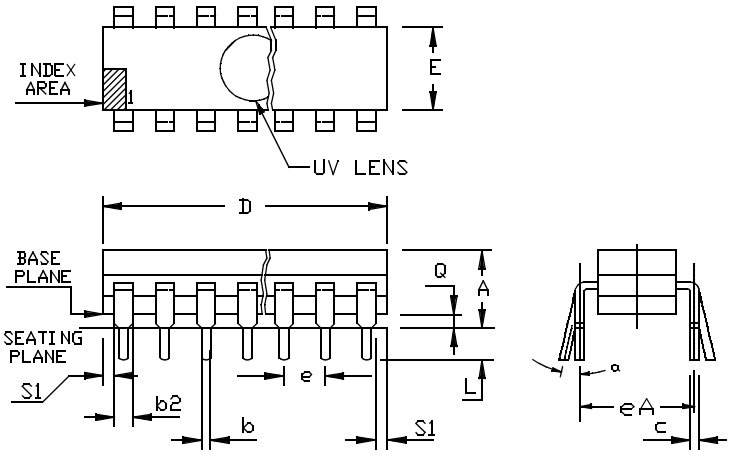


Table 58. CDIL Package Size

	MM		Inch	
	Min	**Max**	**Min**	**Max**
A	-	5.71	-	.225
b	0.36	0.58	.014	.023
b2	1.14	1.65	.045	.065
c	0.20	0.38	.008	.015
D	-	53.47	-	2.105
E	13.06	15.37	.514	.605
e	2.54 B.S.C.		.100 B.S.C.	
eA	15.24 B.S.C.		.600 B.S.C.	
L	3.18	5.08	.125	.200
Q	0.38	1.40	.015	.055
S1	0.13	-	.005	-
a	0 - 15		0 - 15	
N	40			