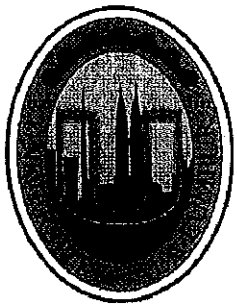


UNIVERSITI KUALA LUMPUR
MALAYSIAN INSTITUTE OF INDUSTRIAL TECHNOLOGY

FINAL EXAMINATION

JANUARY 2016 SEMESTER

COURSE CODE	:	JCB 20603
COURSE TITLE	:	MICROPROCESSOR AND MICROCONTROLLER
PROGRAMME LEVEL	:	BACHELOR
DATE	:	20 MAY 2016
TIME	:	2.30 PM – 5.30 PM
DURATION	:	3 HOURS

INSTRUCTIONS TO CANDIDATES

1. Please read the instructions given in the question paper CAREFULLY.
2. This question paper is printed on both sides of the paper.
3. This question paper consists of TWO (2) sections.
4. Answer ALL questions in Section A. Answer ALL THREE (3) questions in Section B.
5. Please write your answers on the answer booklet provided.
6. Please answer all questions in English only.
7. The 8051 mnemonics and related datasheets are appended

THERE ARE 5 PAGES OF QUESTIONS EXCLUDING THIS PAGE.

SECTION A (Total: 40 marks)

INSTRUCTION: Answer ALL questions.
Please use the answer booklet provided.

Question 1

(a) Microprocessor and microcontroller have been designed for real time application. They share common features and available in different version starting from 6 pin to as high 80 to 100 pins or even higher depending on the features. Explain **THREE (3)** differences between microprocessor and microcontroller.

(6 marks)

(b) Identify **EIGHT (8)** examples of microcontroller applications in manufacturing industries.

(4 marks)

(c) Describe the following function of each port of a microcontroller.

i. Port 2
ii. Memory system
iii. Accumulator
iv. Timer
v. Interrupt
vi. Bus system

(6 marks)

(d) Describe the contents of the RAM locations line by line of the following Algorithm 1;

Algorithm 1 : Addressing Mode

```
MOV R0,#99H
MOV R1,#85H
MOV R2,#3FH
MOV R7,#63H
MOV R5,#12H
```

(4 marks)

Question 2

(a) The XRL instruction can be used to clear the contents of a register by XORing it with itself. Show the instruction "XRL A,A" clears A, assuming that A=45H.

(5 marks)

(b) Identify the content in Register A and the flags (Carry Flag (CY), Parity Flag (P), Auxiliary carry flag (AC)) after executing the following assembly code as shown in Algorithm 2.

Algorithm 2: Arithmetic operation

```
MOV  R1,#0F5H
MOV  R5,#0BH
MOV  A,#1AH
MOV A,#0
ADD A,R1
ADD A,R5
```

(5 marks)

(c) The various addressing modes of a microprocessor are determined when it is designed, and therefore cannot be changed by the programmer. The 8051 provides a total of five distinct addressing modes. Explain briefly **FIVE (5)** addressing modes of 8051 with example of each.

(10 marks)

SECTION B (Total: 60 marks)

INSTRUCTION: Answer THREE (3) questions ONLY.
Please use the answer booklet provided.

Question 1

(a) For a machine cycle of $1.085\mu s$, examine the time delay of the following subroutine.

```
DELAY
            MOV  R2,#200
AGAIN:      MOV  R3,#250
HERE:       NOP
            NOP
            DJNZ R3,HERE
HERE:       DJNZ R2,AGAIN

            RET
```

(5 marks)

(b) Analyze the result of following Algorithm 4.

Algorithm 4: Arithmetic Operation

```
            CLR  C
            MOV  A,#4CH
            SUBB A,#6EH
            JNC  NEXT
            CPL  A
            INC  A
NEXT:       MOV  R1,A
```

(5 marks)

(c) Sketch the oscillator diagram (pulse train) for 8051 microcontroller. Explain the influence of the oscillator circuit to the machine cycle.

(10 marks)

Question 2

(a) Draw the diagram to interface 16X2 LCD display with 8051 microcontroller. Compare LCD module interface with 7 segment LED multiplex interface based on the following points.

- Hardware requirement for displaying 16 characters.
- Refreshing the display
- Power Consumption
- Software Complexity.

(10 marks)

(b) Assume that XTAL = 11.0592 MHz, write a program to generate a square wave of 2 kHz frequency on pin P1.5. Solve for the Timer Low and Timer High and develop our assembly language program.

(10 marks)

Question 3

(a) Draw interfacing diagram of 7 segment LED display with 8051 microcontroller.

(4 marks)

(b) Write a subroutine which checks the content of 20H. If it is a positive number, the subroutine find its two's complement and store it in same location and returns.

(4 marks)

(c) Outline the pseudocode and write an assembly language or C program to read the Port 0 and turn on LEDs for 5 seconds and off for 5 second if port 0 is not zero.

(10 marks)

Question 4

(a) Describe four advantages of embedded system.

(4 marks)

(b) Draw pinout of 14 pin LCD display and state the functions of RS, EN, R/W.

(4 marks)

(c) Whenever a robotics hobbyist talk about making a robot, the first thing comes to his mind is making the robot move on the ground. And there are always two options in front of the designer whether to use a DC motor or a stepper motor. When it comes to speed, weight, size, cost, DC motors are always preferred over stepper motors. Explain the interfacing of DC motor to 8051 using opto insulator. Write a C program to move DC motor with 25% duty cycle pulse

(12 marks)

END OF EXAMINATION PAPER

Features

- Compatible with MCS-51® Products
- 4K Bytes of In-System Programmable (ISP) Flash Memory
  - Endurance: 1000 Write/Erase Cycles
- 4.0V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Watchdog Timer
- Dual Data Pointer
- Power-off Flag
- Fast Programming Time
- Flexible ISP Programming (Byte and Page Mode)

Description

The AT89S51 is a low-power, high-performance CMOS 8-bit microcontroller with 4K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S51 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, two 16-bit timer/counters, a five-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next external interrupt or hardware reset.

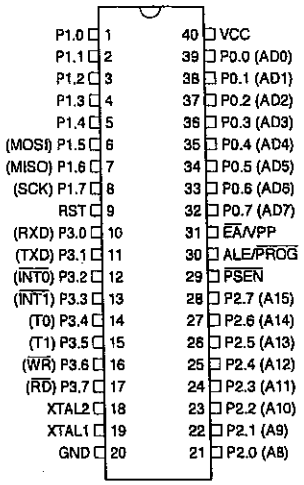
8-bit Microcontroller with 4K Bytes In-System Programmable Flash

AT89S51

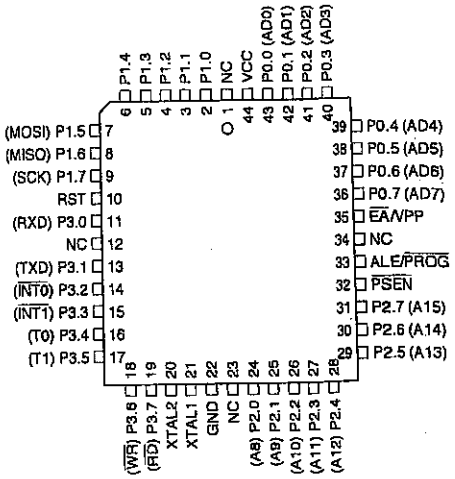
Rev. 2487A–10/01

Pin Configurations

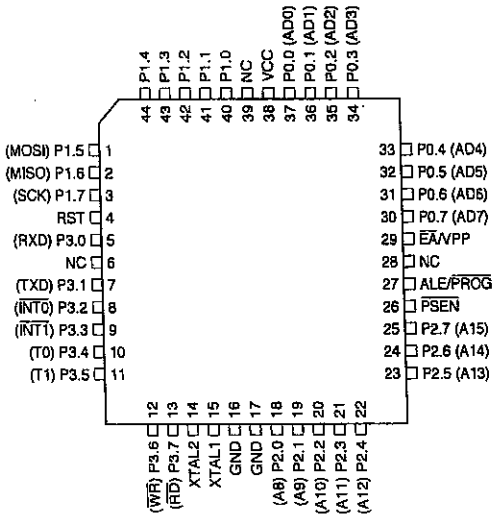
PDIP



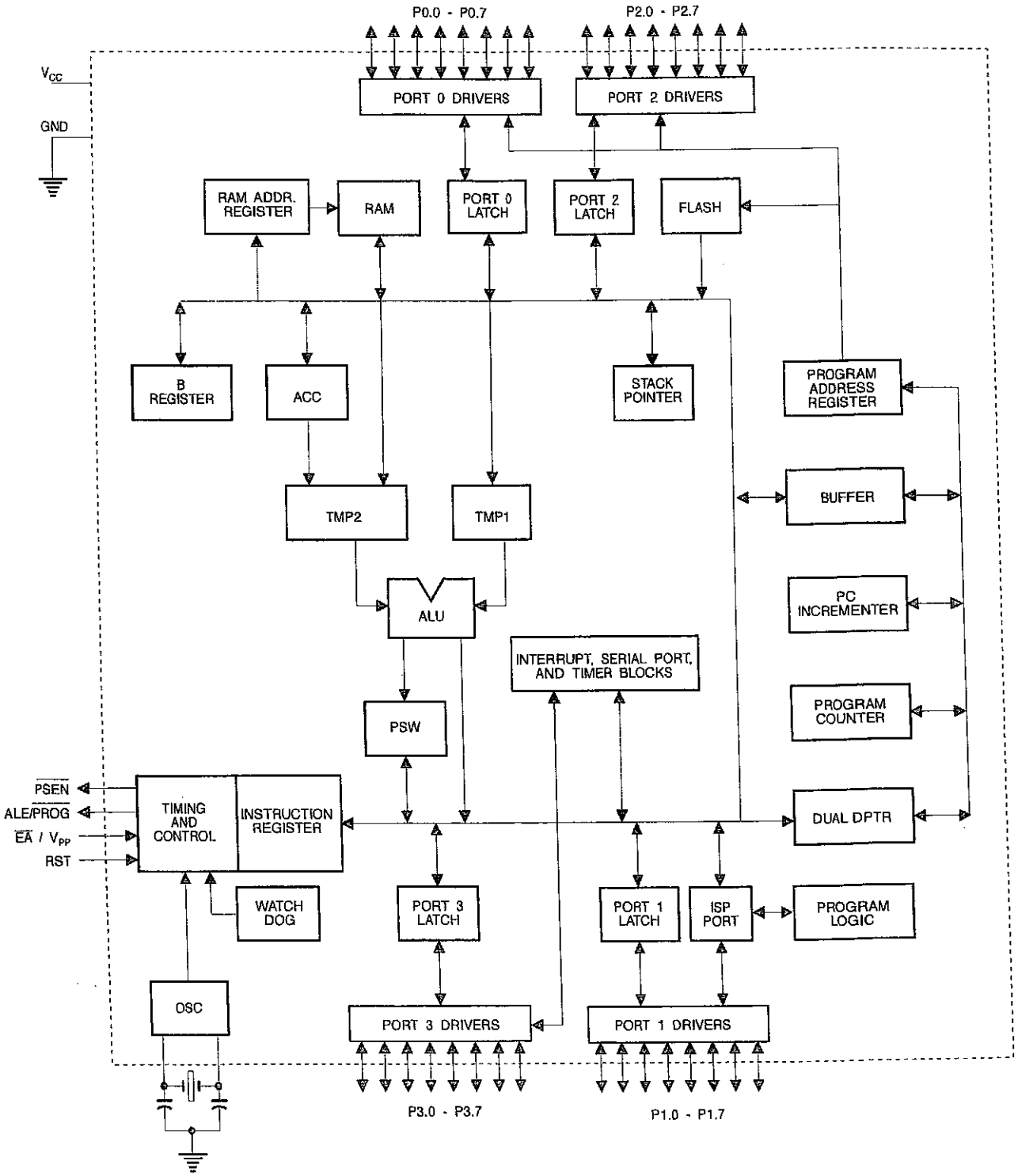
PLCC

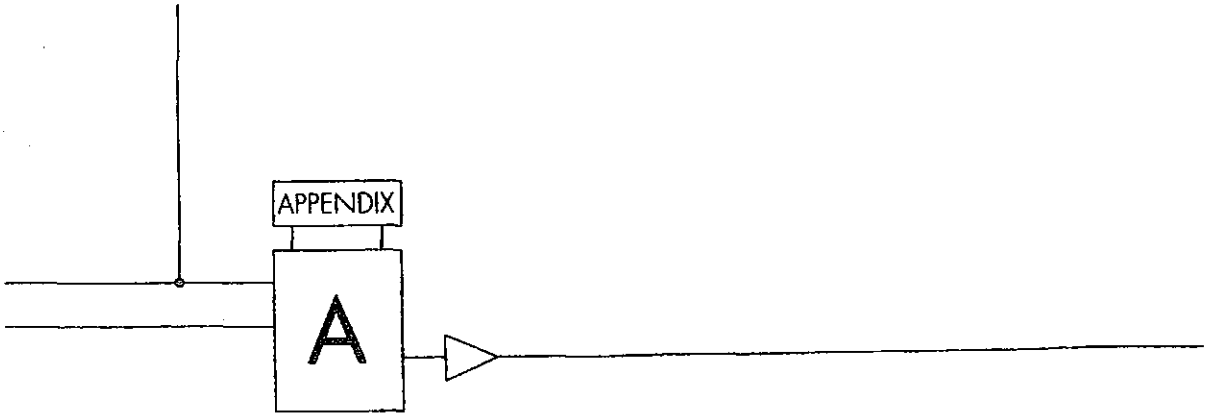


TQFP



Block Diagram





8051 Operational Code Mnemonics

Appendix A lists two arrangements of mnemonics for the 8051: by function, and alphabetically. The mnemonic definitions differ from that of the original manufacturer (Intel Corporation) by the names used for addresses or data; for example, "add" is used to represent an address in internal RAM, while Intel uses the name "direct." The author believes that the names used are clearer than those used by Intel. Appendix A also includes an alphabetical listing of the mnemonics using Intel names. There is no difference between the mnemonics when real numbers replace the names. For example; MOV add,#n and MOV direct,#data become MOV 10h,#40h when the number 10h replaces the internal RAM address (add/direct), and 40h replaces the number (#n/# data).

Mnemonics, Arranged by Function

Arithmetic

MNEMONIC	DESCRIPTION	BYTES	CYCLES	FLAGS
ADD A,Rr	A+Rr → A	1	1	C OV AC
ADD A,add	A+(add) → A	2	1	C OV AC
ADD A,@Rp	A+(Rp) → A	1	1	C OV AC
ADD A,#n	A+n → A	2	1	C OV AC
ADDC A,Rr	A+Rr+C → A	1	1	C OV AC
ADDC A,add	A+(add)+C → A	2	1	C OV AC
ADDC A,@Rp	A+(Rp)+C → A	1	1	C OV AC
ADDC A,#n	A+n+C → A	2	1	C OV AC
DA A	Abin → Adec	1	1	C
DEC A	A−1 → A	1	1	
DEC Rr	Rr−1 → Rr	1	1	
DEC add	(add)−1 → (add)	2	1	
DEC @Rp	(Rp)−1 → (Rp)	1	1	

Continued

Arithmetic

MNEMONIC	DESCRIPTION	BYTES	CYCLES	FLAGS
				Continued
DIV AB	A/B → AB	1	4	0 0V
INC A	A+1 → A	1	1	
INC Rr	Rr+1 → Rr	1	1	
INC add	(add)+1 → (add)	2	1	
INC @Rp	(Rp)+1 → (Rp)	1	1	
INC DPTR	DPTR+1 → DPTR	1	2	
MUL AB	A×B → AB	1	4	0 0V
SUBB A,Rr	A−Rr−C → A	1	1	C OV AC
SUBB A,add	A−(add)−C → A	2	1	C OV AC
SUBB A,@Rp	A−(Rp)−C → A	1	1	C OV AC
SUBB A,#n	A−n−C → A	2	1	C OV AC

Logic

MNEMONIC	DESCRIPTION	BYTES	CYCLES	FLAGS
ANL A,Rr	A AND Rr → A	1	1	
ANL A,add	A AND (add) → A	2	1	
ANL A,@Rp	A AND (Rp) → A	1	1	
ANL A,#n	A AND n → A	2	1	
ANL add,A	(add) AND A → (add)	2	1	
ANL add,#n	(add) AND n → (add)	3	2	
ORL A,Rr	A OR Rr → A	1	1	
ORL A,add	A OR (add) → A	2	1	
ORL A,@Rp	A OR (Rp) → A	1	1	
ORL A,#n	A OR n → A	2	1	
ORL add,A	(add) OR A → (add)	2	1	
ORL add,#n	(add) OR n → (add)	3	2	
XRL A,Rr	A XOR Rr → A	1	1	
XRL A,add	A XOR (add) → A	2	1	
XRL A,@Rp	A XOR (Rp) → A	1	1	
XRL A,#n	A XOR n → A	2	1	
XRL add,A	(add) XOR A → (add)	2	1	
XRL add,#n	(add) XOR n → (add)	3	2	
CLR A	00 → A	1	1	
CPL A	$\overline{A}$ → A	1	1	
NOP	PC+1 → PC	1	1	
RL A	A0←A7←A6. .←A1←A0	1	1	
RLC A	C←A7←A6. .←A0←C	1	1	C
RR A	A0→A →A6. .→A1→A0	1	1	
RRC A	C→A7→A6. .→A0→C	1	1	C
SWAP A	Alsn ↔ Amsn	1	1	

Data Moves

MNEMONIC	DESCRIPTION	BYTES	CYCLES	FLAGS
MOV A,Rr	Rr → A	1	1	
MOV A,add	(add) → A	2	1	
MOV A,@Rp	(Rp) → A	1	1	

Continued

MNEMONIC	DESCRIPTION	BYTES	CYCLES	FLAGS
MOV A,#n	n → A	2	1	
MOV Rr,A	A → Rr	1	1	
MOV Rr,add	(add) → Rr	2	2	
MOV Rr,#n	n → Rr	2	1	
MOV add,A	A → (add)	2	1	
MOV add,Rr	Rr → (add)	2	2	
MOV add1,add2	(add2) → (add1)	3	2	
MOV add,@Rp	(Rp) → (add)	2	2	
MOV add,#n	n → (add)	3	2	
MOV @Rp,A	A → (Rp)	1	1	
MOV @Rp,add	(add) → (Rp)	2	2	
MOV @Rp,#n	n → (Rp)	2	1	
MOV DPTR,#nn	nn → DPTR	3	2	
MOVC A,@A+DPTR	(A+DPTR) → A	1	2	
MOVC A,@A+PC	(A+PC) → A	1	2	
MOVX A,@DPTR	(DPTR)^ → A	1	2	
MOVX A,@Rp	(Rp)^ → A	1	2	
MOVX @Rp,A	A → (Rp)^	1	2	
MOVX @DPTR,A	A → (DPTR)^	1	2	
POP add	(SP) → (add)	2	2	
PUSH add	(add) → (SP)	2	2	
XCH A,Rr	A ↔ Rr	1	1	
XCH A,add	A ↔ (add)	2	1	
XCH A,@Rp	A ↔ (Rp)	1	1	
XCHD A,@Rp	Alsn ↔ (Rp)lsn	1	1	

Calls and Jumps

MNEMONIC	DESCRIPTION	BYTES	CYCLES	FLAGS
ACALL sadd	PC+2 → (SP); sadd → PC	2	2	
CJNE A,add,radd	[A<>(add)]: PC+3+radd → PC	3	2	C
CJNE A,#n,radd	[A<>n]: PC+3+radd → PC	3	2	C
CJNE Rr,#n,radd	[Rr<>n]: PC+3+radd → PC	3	2	C
CJNE @Rp,#n,radd	[(Rp)<>n]: PC+3+radd → PC	3	2	C
DJNZ Rr,radd	[Rr−1<>00]: PC+2+radd → PC	2	2	
DJNZ add,radd	[(add)−1<>00]: PC+3+radd → PC	3	2	
LCALL ladd	PC+3 → (SP); ladd → PC	3	2	
AJMP sadd	sadd → PC	2	2	
LJMP ladd	ladd → PC	3	2	
SJMP radd	PC+2+radd → PC	2	2	
JMP @A+DPTR	DPTR+A → PC	1	2	
JC radd	[C=1]: PC+2+radd → PC	2	2	
JNC radd	[C=0]: PC+2+radd → PC	2	2	
JB b,radd	[b=1]: PC+3+radd → PC	3	2	
JNB b,radd	[b=0]: PC+3+radd → PC	3	2	
JBC b,radd	[b=1]: PC+3+radd → PC; 0 → b	3	2	
JZ radd	[A=00]: PC+2+radd → PC	2	2	
JNZ radd	[A>00]: PC+2+radd → PC	2	2	
RET	(SP) → PC	1	2	
RETI	(SP) → PC; EI	1	2	

Boolean

MNEMONIC	DESCRIPTION	BYTES	CYCLES	FLAGS
ANL C,b	C AND b → C	2	2	C
ANL C,/b	C AND $\overline{b}$ → C	2	2	C
CLR C	0 → C	1	1	0
CLR b	0 → b	2	1	
CPL C	$\overline{C}$ → C	1	1	C
CPL b	$\overline{b}$ → b	2	1	
ORL C,b	C OR b → C	2	2	C
ORL C,/b	C OR $\overline{b}$ → C	2	2	C
MOV C,b	b → C	2	1	C
MOV b,C	C → b	2	2	
SETB C	1 → C	1	1	1
SETB b	1 → b	2	1	

Mnemonics, Arranged Alphabetically

MNEMONIC	DESCRIPTION	BYTES	CYCLES	FLAGS
ACALL sadd	PC+2 → (SP); sadd → PC	2	2	
ADD A,add	A+(add) → A	2	1	C OV AC
ADD A,@Rp	A+(Rp) → A	1	1	C OV AC
ADD A,#n	A+n → A	2	1	C OV AC
ADD A,Rr	A+Rr → A	1	1	C OV AC
ADDC A,add	A+(add)+C → A	2	1	C OV AC
ADDC A,@Rp	A+(Rp)+C → A	1	1	C OV AC
ADDC A,#n	A+n+C → A	2	1	C OV AC
ADDC A,Rr	A+Rr+C → A	1	1	C OV AC
AJMP sadd	sadd → PC	2	2	
ANL A,add	A AND (add) → A	2	1	
ANL A,@Rp	A AND (Rp) → A	1	1	
ANL A,#n	A AND n → A	2	1	
ANL A,Rr	A AND Rr → A	1	1	
ANL add,A	(add) AND A → (add)	2	1	
ANL add,#n	(add) AND n → (add)	3	2	
ANL C,b	C AND b → C	2	2	C
ANL C,/b	C AND $\overline{b}$ → C	2	2	C
CJNE A,add,radd	[A<>(add)]: PC+3+*radd* → PC	3	2	C
CJNE A,#n,radd	[A<>n]: PC+3+radd → PC	3	2	C
CJNE @Rp,#n,radd	[(Rp)<>n]: PC+3+radd → PC	3	2	C
CJNE Rr,#n,radd	[Rr<>n]: PC+3+radd → PC	3	2	C
CLR A	0 → A	1	1	
CLR b	0 → b	2	1	
CLR C	0 → C	1	1	0
CPL A	$\overline{A}$ → A	1	1	
CPL b	$\overline{b}$ → b	2	1	
CPL C	$\overline{C}$ → C	1	1	C
DA A	Abin → Adec	1	1	C
DEC A	A−1 → A	1	1	
DEC add	(add)−1 → (add)	2	1	
DEC @Rp	(Rp)−1 → (Rp)	1	1	
DEC Rr	Rr−1 → Rr	1	1	

Continued

MNEMONIC	DESCRIPTION	BYTES	CYCLES	FLAGS
DIV AB	A/B → AB	1	4	0 OV
DJNZ add,radd	[(add)−1<>00]: PC+3+radd → PC	3	2	
DJNZ Rr,radd	[Rr−1<>00]: PC+2+radd → PC	2	2	
INC A	A+1 → A	1	1	
INC add	(add)+1 → (add)	2	1	
INC DPTR	DPTR+1 → DPTR	1	2	
INC @Rp	(Rp)+1 → (Rp)	1	1	
INC Rr	Rr+1 → Rr	1	1	
JB b,radd	[b=1]: PC+3+radd → PC	3	2	
JBC b,radd	[b=1]: PC+3+radd → PC; 0 → b	3	2	
JC radd	[C=1]: PC+2+radd → PC	2	2	
JMP @A+DPTR	DPTR+A → PC	1	2	
JNB b,radd	[b=0]: PC+3+radd → PC	3	2	
JNC radd	[C=0]: PC+2+radd → PC	2	2	
JNZ radd	[A>00]: PC+2+radd → PC	2	2	
JZ radd	[A=00]: PC+2+radd → PC	2	2	
LCALL ladd	PC+3 → (SP); ladd → PC	3	2	
LJMP ladd	ladd → PC	3	2	
MOV A,add	(add) → A	2	1	
MOV A,@Rp	(Rp) → A	1	1	
MOV A,#n	n → A	2	1	
MOV A,Rr	Rr → A	1	1	
MOV add,A	A → (add)	2	1	
MOV add1,add2	(add2) → (add1)	3	2	
MOV add,@Rp	(Rp) → (add)	2	2	
MOV add,#n	n → (add)	3	2	
MOV add,Rr	Rr → (add)	2	2	
MOV b,C	C → b	2	2	
MOV C,b	b → C	2	1	C
MOV @Rp,A	A → (Rp)	1	1	
MOV @Rp,add	(add) → (Rp)	2	2	
MOV @Rp,#n	n → (Rp)	2	1	
MOV DPTR,#nn	nn → DPTR	3	2	
MOV Rr,A	A → Rr	1	1	
MOV Rr,add	(add) → Rr	2	2	
MOV Rr,#n	n → Rr	2	1	
MOVC A,@A+DPTR	(A+DPTR) → A	1	2	
MOVC A,@A+PC	(A+PC) → A	1	2	
MOVX A,@DPTR	(DPTR)^ → A	1	2	
MOVX A,@Rp	(Rp)^ → A	1	2	
MOVX @DPTR,A	A → (DPTR)^	1	2	
MOVX @Rp,A	A → (Rp)^	1	2	
NOP	PC+1 → PC	1	1	
MUL AB	AxB → AB	1	4	0 OV
ORL A,add	A OR (add) → A	2	1	
ORL A,@Rp	A OR (Rp) → A	1	1	
ORL A,#n	A OR n → A	2	1	
ORL A,Rr	A OR Rr → A	1	1	
ORL add,A	(add) OR A → (add)	2	1	
ORL add,#n	(add) OR n → (add)	3	2	
ORL C,b	C OR b → C	2	2	C
ORL C,$\overline{b}$	C OR $\overline{b}$ → C	2	2	C

Continued

MNEMONIC	DESCRIPTION	BYTES	CYCLES	FLAGS
				Continued
POP add	(SP) → (add)	2	2	
PUSH add	(add) → (SP)	2	2	
RET	(SP) → PC	1	2	
RETI	(SP) → PC; EI	1	2	
RL A	A0←A7←A6..←A1←A0	1	1	
RLC A	C←A7←A6..←A0←C	1	1	C
RR A	A0→A7→A6..→A1→A0	1	1	
RRC A	C→A7→A6..→A0→C	1	1	C
SETB b	1 → b	2	1	
SETB C	1 → C	1	1	1
SJMP radd	PC+2+radd → PC	2	2	
SUBB A,add	A−(add)−C → A	2	1	C OV AC
SUBB A,@Rp	A−(Rp)−C → A	1	1	C OV AC
SUBB A,#n	A−n−C → A	2	1	C OV AC
SUBB A,Rr	A−Rr−C → A	1	1	C OV AC
SWAP A	Alsn ↔ Amsn	1	1	
XCH A,add	A ↔ (add)	2	1	
XCH A,@Rp	A ↔ (Rp)	1	1	
XCH A,Rr	A ↔ Rr	1	1	
XCHD A,@Rp	Alsn ↔ (Rp)lsn	1	1	
XRL A,add	A XOR (add) → A	2	1	
XRL A,@Rp	A XOR (Rp) → A	1	1	
XRL A,#n	A XOR n → A	2	1	
XRL A,Rr	A XOR Rr → A	1	1	
XRL add,A	(add) XOR A → (add)	2	1	
XRL add,#n	(add) XOR n → (add)	3	2	

MNEMONIC ACRONYMS

add	Address of the internal RAM from 00h to FFh.
ladd	Long address of 16 bits from 0000h to FFFFh.
radd	Relative address, a signed number from −128d to +127d.
sadd	Short address of 11 bits; complete address = PC11−PC15 and sadd.
b	Addressable bit in internal RAM or a SFR.
C	The carry flag.
lsn	Least significant nibble.
msn	Most significant nibble.
n	Any immediate 8 bit number from 00h to FFh.
Rr	Any of the eight registers, R0 to R7 in the selected bank.
Rp	Either of the pointing registers R0 or R1 in the selected bank.
[ ]:	IF the condition inside the brackets is *true*, THEN the action listed will occur; ELSE go to the next instruction.
^	External memory location.
( )	Contents of the location inside the parentheses.

Note that flags affected by each instruction are shown where appropriate; any operations that affect the PSW address may also affect the flags.

Intel Corporation Mnemonics, Arranged Alphabetically

MNEMONIC	DESCRIPTION	BYTES	CYCLES	FLAGS
ACALL addr11	PC+2 → (SP); addr11 → PC	2	2	
ADD A,direct	A+(direct) → A	2	1	C OV AC
ADD A,@Ri	A+(Ri) → A	1	1	C OV AC
ADD A,#data	A+#data → A	2	1	C OV AC
ADD A,Rn	A+Rn → A	1	1	C OV AC
ADDC A,direct	A+(direct)+C → A	2	1	C OV AC
ADDC A,@Ri	A+(Ri)+C → A	1	1	C OV AC
ADDC A,#data	A+#data+C → A	2	1	C OV AC
ADDC A,Rn	A+Rn+C → A	1	1	C OV AC
AJMP addr11	addr11 → PC	2	2	
ANL A,direct	A AND (direct) → A	2	1	
ANL A,@Ri	A AND (Ri) → A	1	1	
ANL A,#data	A AND #data → A	2	1	
ANL A,Rn	A AND Rn → A	1	1	
ANL direct,A	(direct) AND A → (direct)	2	1	
ANL direct,#data	(direct) AND #data → (direct)	3	2	
ANL C,bit	C AND bit → C	2	2	C
ANL C,$\overline{bit}$	C AND $\overline{bit}$ → C	2	2	C
CJNE A,direct,rel	[A<>(direct)]: PC+3+rel → PC	3	2	C
CJNE A,#data,rel	[A<>n]: PC+3+rel → PC	3	2	C
CJNE @Ri,#data,rel	[(Ri)<>n]: PC+3+rel → PC	3	2	C
CJNE Rn,#data,rel	[Rn<>n]: PC+3+rel → PC	3	2	C
CLR A	0 → A	1	1	
CLR bit	0 → bit	2	1	
CLR C	0 → C	1	1	0
CPL A	$\overline{A}$ → A	1	1	
CPL bit	$\overline{bit}$ → bit	2	1	
CPL C	$\overline{C}$ → C	1	1	C
DA A	Abin → Adec	1	1	C
DEC A	A−1 → A	1	1	
DEC direct	(direct)−1 → (direct)	2	1	
DEC @Ri	(Ri)−1 → (Ri)	1	1	
DEC Rn	Rn−1 → Rn	1	1	
DIV AB	A/B → AB	1	4	0 OV
DJNZ direct,rel	[(direct)−1<>00]: PC+3+rel → PC	3	2	
DJNZ Rn,rel	[Rn−1<>00]: PC+2+rel → PC	2	2	
INC A	A+1 → A	1	1	
INC direct	(direct)+1 → (direct)	2	1	
INC DPTR	DPTR+1 → DPTR	1	2	
INC @Ri	(Ri)+1 → (Ri)	1	1	
INC Rn	Rn+1 → Rn	1	1	
JB bit,rel	[b=1]: PC+3+rel → PC	3	2	
JBC bit,rel	[b=1]: PC+3+rel → PC; 0 → bit	3	2	
JC rel	[C=1]: PC+2+rel → PC	2	2	
JMP @A+DPTR	DPTR+A → PC	1	2	
JNB bit,rel	[b=0]: PC+3+rel → PC	3	2	
JNC rel	[C=0]: PC+2+rel → PC	2	2	
JNZ rel	[A>00]: PC+2+rel → PC	2	2	
JZ rel	[A=00]: PC+2+rel → PC	2	2	

Continued

MNEMONIC	DESCRIPTION	BYTES	CYCLES	FLAGS
LCALL addr16	PC+3 → (SP); addr16 → PC	3	2	
LJMP addr16	addr16 → PC	3	2	
MOV A,direct	(direct) → A	2	1	
MOV A,@Ri	(Ri) → A	1	1	
MOV A,#data	#data → A	2	1	
MOV A,Rn	Rn → A	1	1	
MOV direct,A	A → (direct)	2	1	
MOV direct,direct	(direct) → (direct)	3	2	
MOV direct,@Ri	(Ri) → (direct)	2	2	
MOV direct,#data	#data → (direct)	3	2	
MOV direct,Rn	Rn → (direct)	2	2	
MOV bit,C	C → bit	2	2	
MOV C,bit	bit → C	2	1	C
MOV @Ri,A	A → (Ri)	1	1	
MOV @Ri,direct	(direct) → (Ri)	2	2	
MOV @Ri,#data	#data → (Ri)	2	1	
MOV DPTR,#data16	#data16 → DPTR	3	2	
MOV Rn,A	A → Rn	1	1	
MOV Rn,direct	(direct) → Rn	2	2	
MOV Rn,#data	#data → Rn	2	1	
MOVC A,@A+DPTR	(A+DPTR) → A	1	2	
MOVC A,@A+PC	(A+PC) → A	1	2	
MOVX A,@DPTR	(DPTR)^ → A	1	2	
MOVX A,@Ri	(Ri)^ → A	1	2	
MOVX @DPTR,A	A → (DPTR)^	1	2	
MOVX @Ri,A	A → (Ri)^	1	2	
NOP	PC+1 → PC	1	1	
MUL AB	A×B → AB	1	4	0 OV
ORL A,direct	A OR (direct) → A	2	1	
ORL A,@Ri	A OR (Ri) → A	1	1	
ORL A,#data	A OR #data → A	2	1	
ORL A,Rn	A OR Rn → A	1	1	
ORL direct,A	(direct) OR A → (direct)	2	1	
ORL direct,#data	(direct) OR #data → (direct)	3	2	
ORL C,bit	C OR bit → C	2	2	C
ORL C,/bit	C OR /bit → C	2	2	C
POP direct	(SP) → (direct)	2	2	
PUSH direct	(direct) → (SP)	2	2	
RET	(SP) → PC	1	2	
RETI	(SP) → PC; EI	1	2	
RL A	A0←A7←A6..←A1←A0	1	1	
RLC A	C←A7←A6..←A0←C	1	1	C
RR A	A0→A7→A6..→A1→A0	1	1	
RRC A	C→A7→A6..→A0→C	1	1	C
SETB bit	1 → bit	2	1	
SETB C	1 → C	1	1	1
SJMP rel	PC+2+rel → PC	2	2	
SUBB A,direct	A−(direct)−C → A	2	1	C OV AC
SUBB A,@Ri	A−(Ri)−C → A	1	1	C OV AC
SUBB A,#data	A−#data−C → A	2	1	C OV AC
SUBB A,Rn	A−Rn−C → A	1	1	C OV AC

Continued

MNEMONIC	DESCRIPTION	BYTES	CYCLES	FLAGS
SWAP A	Alsn ↔ Amsn	1	1	
XCH A,direct	A ↔ (direct)	2	1	
XCH A,@Ri	A ↔ (Ri)	1	1	
XCH A,Rn	A ↔ Rn	1	1	
XCHD A,@Ri	Alsn ↔ (Ri)lsn	1	1	
XRL A,direct	A XOR (direct) → A	2	1	
XRL A,@Ri	A XOR (Ri) → A	1	1	
XRL A,#data	A XOR #data → A	2	1	
XRL A,Rn	A XOR Rn → A	1	1	
XRL direct,A	(direct) XOR A → (direct)	2	1	
XRL direct,#data	(direct) XOR #data → (direct)	3	2	

ACRONYMS

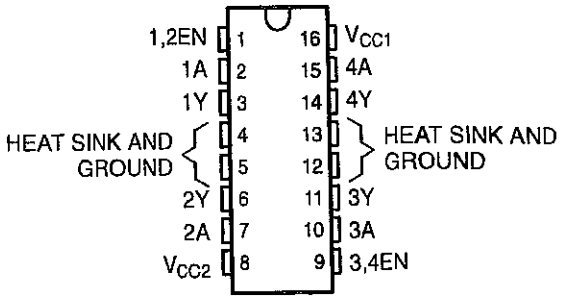
addr11	Page address of 11 bits, which is in the same 2K page as the address of the following instruction.
addr16	Address for any location in the 64K memory space.
bit	The address of a bit in the internal RAM bit address area or a bit in an SFR.
C	The carry flag.
#data	An 8-bit binary number from 00 to FFh.
#data16	A 16-bit binary number from 0000 to FFFFh.
direct	An internal RAM address or an SFR byte address.
lsn	Least significant nibble.
msn	Most significant nibble.
rel	Number that is added to the address of the next instruction to form an address +127d or −128d from the address of the next instruction.
Rn	Any of registers R0 to R7 of the current register bank.
@Ri	Indirect address using the contents of R0 or R1.
[ ]:	IF the condition inside the brackets is *true*, THEN the action listed will occur; ELSE go to the next instruction.
^	EXTERNAL memory location.
( )	Contents of the location inside the parentheses.

Note that flags affected by each instruction are shown where appropriate; any operations which affect the PSW address may also affect the flags.

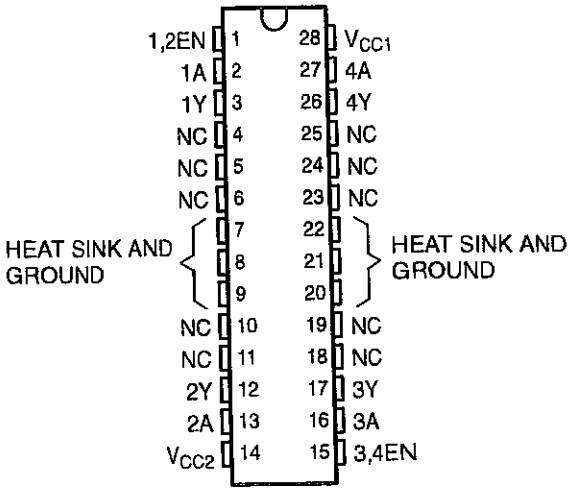
SLRS008C – SEPTEMBER 1986 – REVISED NOVEMBER 2004

- **Featuring Unitrode L293 and L293D Products Now From Texas Instruments**
- **Wide Supply-Voltage Range: 4.5 V to 36 V**
- **Separate Input-Logic Supply**
- **Internal ESD Protection**
- **Thermal Shutdown**
- **High-Noise-Immunity Inputs**
- **Functionally Similar to SGS L293 and SGS L293D**
- **Output Current 1 A Per Channel (600 mA for L293D)**
- **Peak Output Current 2 A Per Channel (1.2 A for L293D)**
- **Output Clamp Diodes for Inductive Transient Suppression (L293D)**

L293 . . . N OR NE PACKAGE
L293D . . . NE PACKAGE
(TOP VIEW)



L293 . . . DWP PACKAGE
(TOP VIEW)



description/ordering information

The L293 and L293D are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	HSOP (DWP)	Tube of 20	L293DWP	L293DWP
	PDIP (N)	Tube of 25	L293N	L293N
	PDIP (NE)	Tube of 25	L293NE	L293NE
		Tube of 25	L293DNE	L293DNE

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

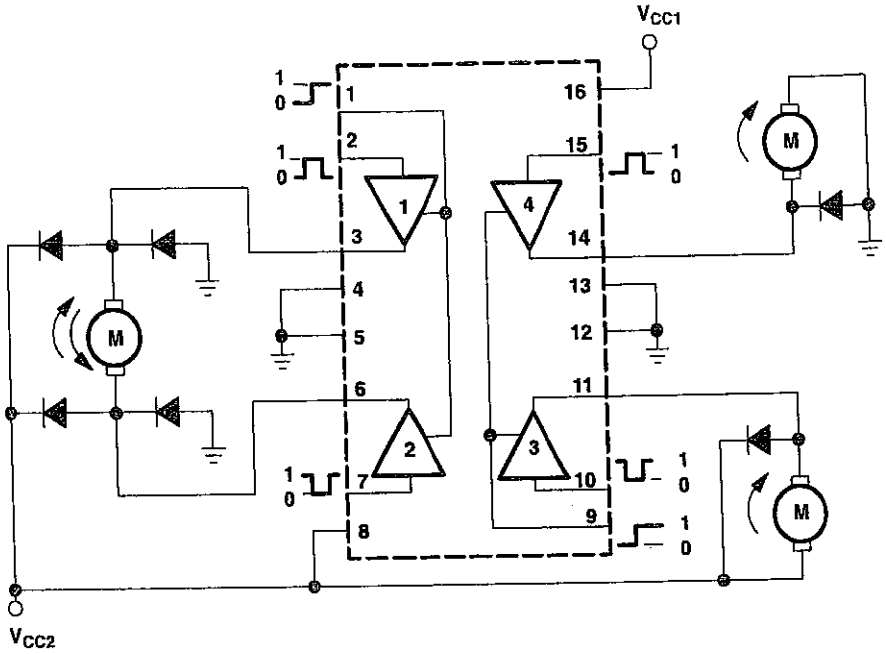
description/ordering information (continued)

On the L293, external high-speed output clamp diodes should be used for inductive transient suppression.

A V_{CC1} terminal, separate from V_{CC2}, is provided for the logic inputs to minimize device power dissipation.

The L293and L293D are characterized for operation from 0°C to 70°C.

block diagram



NOTE: Output diodes are internal in L293D.

FUNCTION TABLE
(each driver)

INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

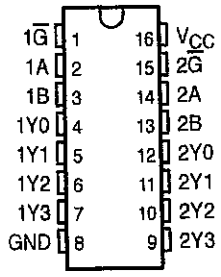
H = high level, L = low level, X = irrelevant,
Z = high impedance (off)

† In the thermal shutdown mode, the output is in the high-impedance state, regardless of the input levels.

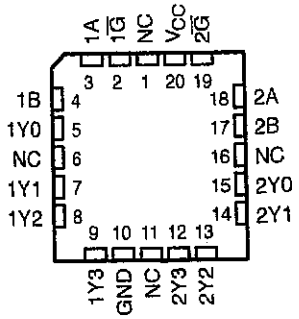
SDAS204A – APRIL 1982 – REVISED DECEMBER 1994

- **Designed Specifically for High-Speed Memory Decoders and Data Transmission Systems**
- **Incorporate Two Enable Inputs to Simplify Cascading and/or Data Reception**
- **Package Options Include Plastic Small-Outline (D) Packages, Ceramic Chip Carriers (FK), and Standard Plastic (N) and Ceramic (J) 300-mil DIPs**

SN54ALS139 . . . J PACKAGE
SN74ALS139 . . . D OR N PACKAGE
(TOP VIEW)



SN54ALS139 . . . FK PACKAGE
(TOP VIEW)



NC – No internal connection

description

The 'ALS139 are dual 2-line to 4-line decoders/demultiplexers designed for use in high-performance memory-decoding or data-routing applications requiring very short propagation delay times. In high-performance memory systems, these devices can minimize the effects of system decoding. When employed with high-speed memories utilizing a fast-enable circuit, the delay times of these decoders and the enable time of the memory are usually less than the typical access time of the memory. Therefore, the effective system delay introduced by the Schottky-clamped system decoder is negligible.

The 'ALS139 comprise two individual 2-line to 4-line decoders in a single package. The active-low enable ($\overline{G}$) input can be used as a data line in demultiplexing applications. These decoders/demultiplexers feature fully buffered inputs, each of which represents only one normalized load to its driving circuit. All inputs are clamped with high-performance Schottky diodes to suppress line ringing and simplify system design.

The SN54ALS139 is characterized for operation over the full military temperature range of −55°C to 125°C. The SN74ALS139 is characterized for operation from 0°C to 70°C.

FUNCTION TABLE

INPUTS			OUTPUTS			
ENABLE $\overline{G}$	SELECT					
	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

Kitronik Ltd – 7 Segment Display

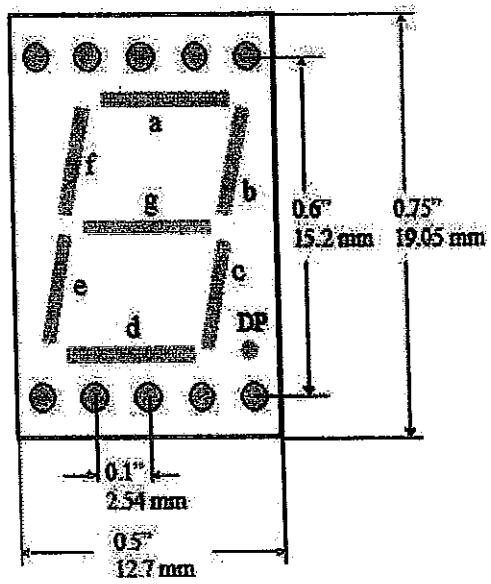
This data sheet details the following two parts:

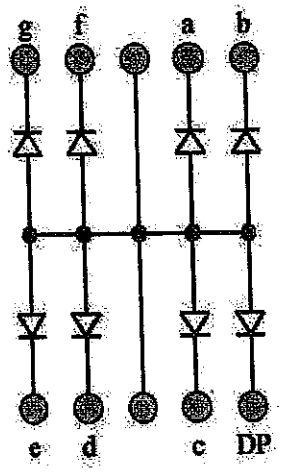
Description	Kingbright Part No	Kitronik Part No
14.2 mm green 7 segment common anode display	SA56-11GWA	3512
14.2 mm green 7 segment common cathode display	SC56-11GWA	3513

Both parts have the same mechanical attributes and use a standard 0.1" lead pitch for easy mounting. The difference between the parts as (shown below) is whether the LED anodes or LED cathodes are connected together.

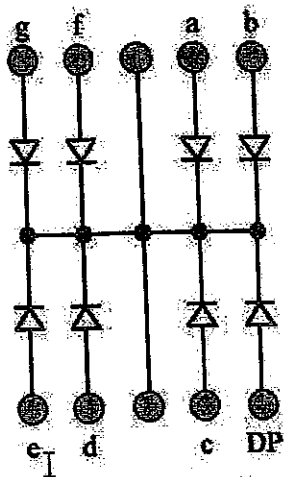
LED Characteristics

Voltage drop per LED segment:	2.2V (typical)
Max current per LED segment:	25mA
Light output @ 10mA:	10 mCd (typical)
Suitable current limit resistor for 5V supply:	220ohm





Common anode



Common cathode