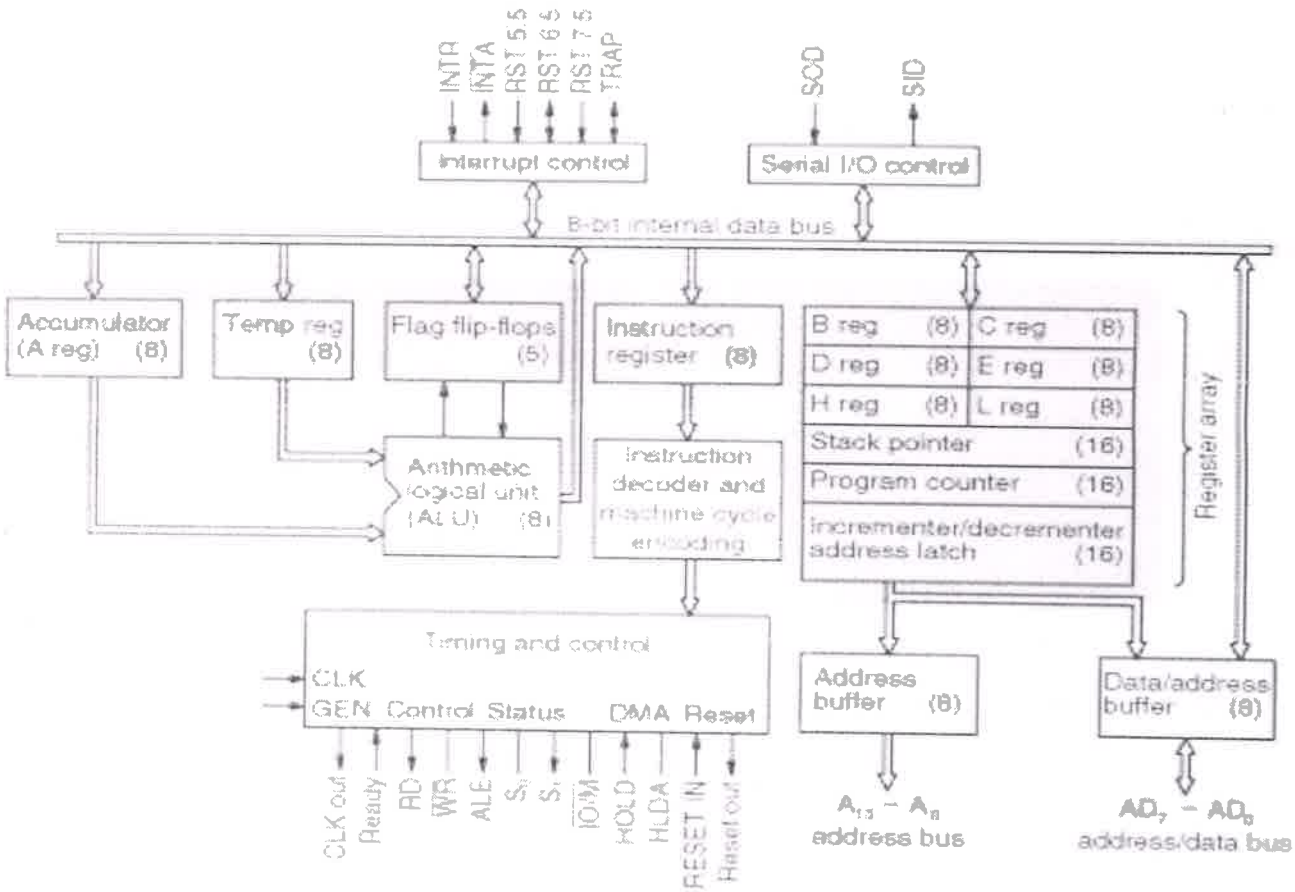
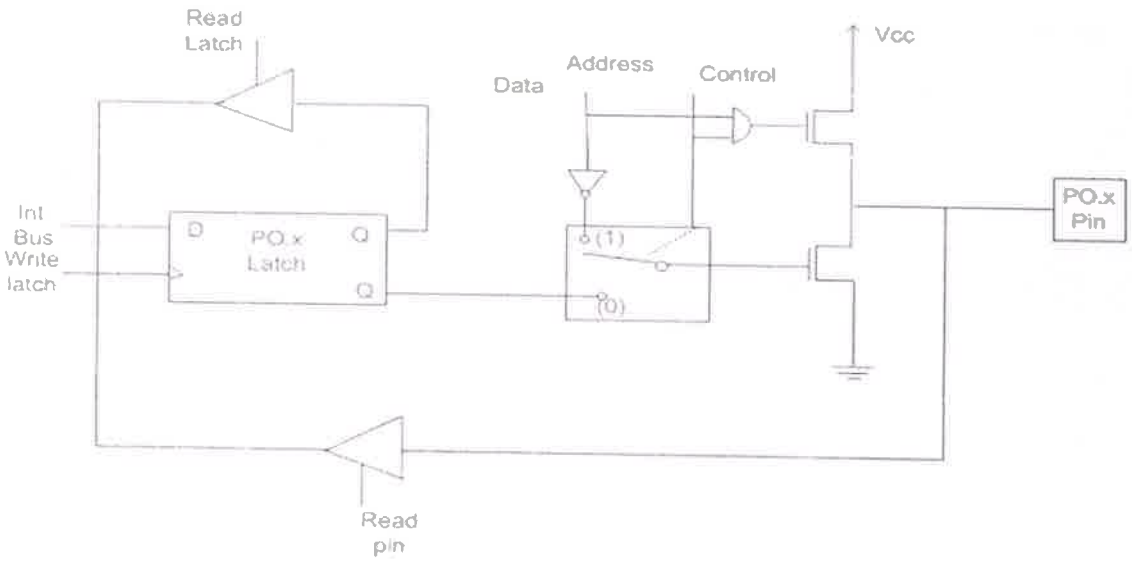
QP.23762

SOLUTION

Q.1 a) Block diagram of 8085 microprocessor:



Q.1 b) PORT 0 Structure:



Q.1 c)

```
MOV A, #38H
ADD A, #2FH ;after the addition A=67H, CY=0
```

Solution:

38	00111000
+ 2F	00101111
67	01100111

CY = 0 since there is no carry beyond the D7 bit

AC = 1 since there is a carry from the D3 to the D4 bi

P = 1 since the accumulator has an odd number of 1s (it has five 1s)

Q.1 d)

The machine cycle frequency of 8051 = 11.0592 / 12 = 921.6 kHz, and 921.6 kHz / 32 = 28.800 Hz is frequency by UART to timer 1 to set baud rate.

(a) 28.800 / 3 = 9600 where -3 = FD (hex) is loaded into TH1
(b) 28.800 / 12 = 2400 where -12 = F4 (hex) is loaded into TH1
(c) 28.800 / 24 = 1200 where -24 = E8 (hex) is loaded into TH1

Q.2 a) ***RISC & CISC Comparison:-***

CISC	RISC
It is prominent on Hardware	It is prominent on the Software
It has high cycles per second	It has low cycles per second
It has transistors used for storing Instructions which are complex	More transistors are used for storing memory
LOAD and STORE memory-to-memory is induced in instructions	LOAD and STORE register-register are independent
It has multi-clock	It has a single - clock

(3)

Q.2 b)

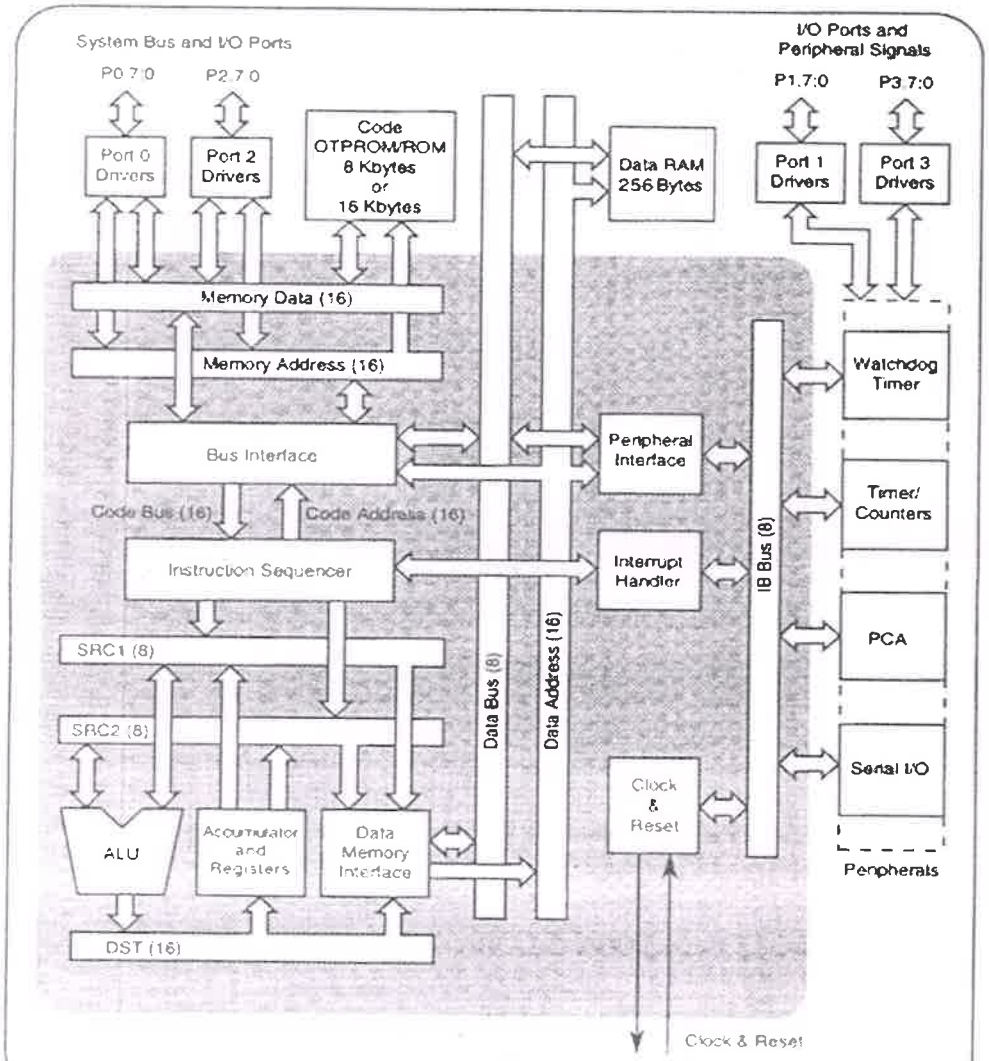
```
MOV     A,#29H   ;A=29H, packed BCD
MOV     R2,A     ;keep a copy of BCD data
ANL     A,#0FH   ;mask the upper nibble (A=09)
ORL     A,#30H   ;make it an ASCII, A=39H('9')
MOV     R6,A     ;save it
MOV     A,R2     ;A=29H, get the original
data
ANL     A,#0F0H ;mask the lower nibble
RR      A        ;rotate right   } SWAP A
RR      A        ;rotate right   }
RR      A        ;rotate right   }
RR      A        ;rotate right   }
ORL     A,#30H   ;A=32H, ASCII char. '2'
MOV     R2,A     ;save ASCII char in R2
```

Q.2 c)

Features	8051	8031	8032	8052
ROM	4K	0	0	8K
RAM	128	128	256	256
Timers	2	2	3	3
I/O pins	32	32	32	32
Serial Port	1	1	1	1
Interrupt Sources	6	6	5	8

4

Q.3 a)

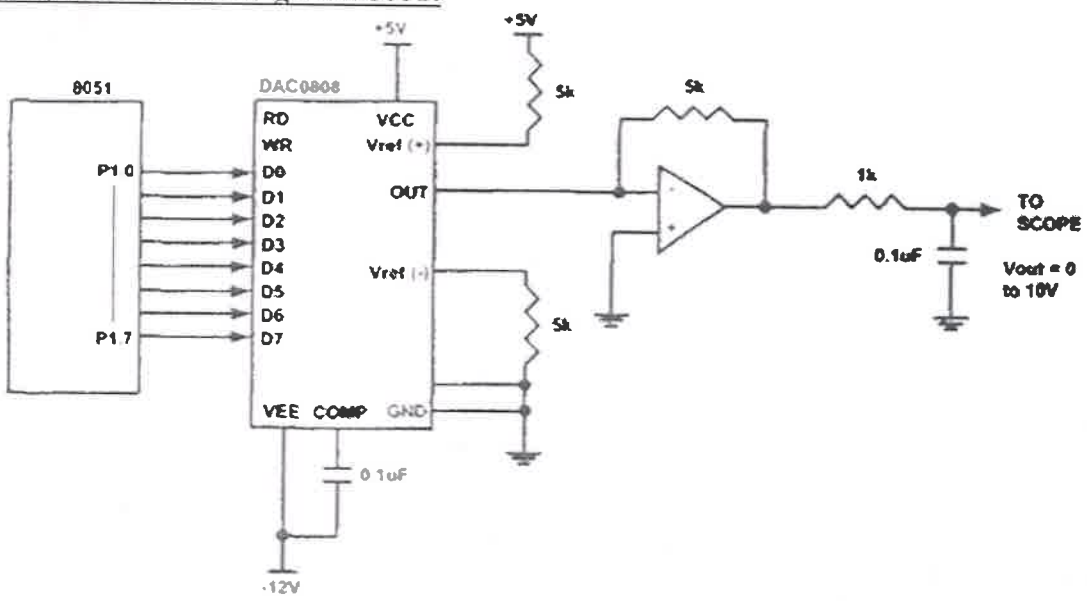


8XC151SA/SB Microcontroller

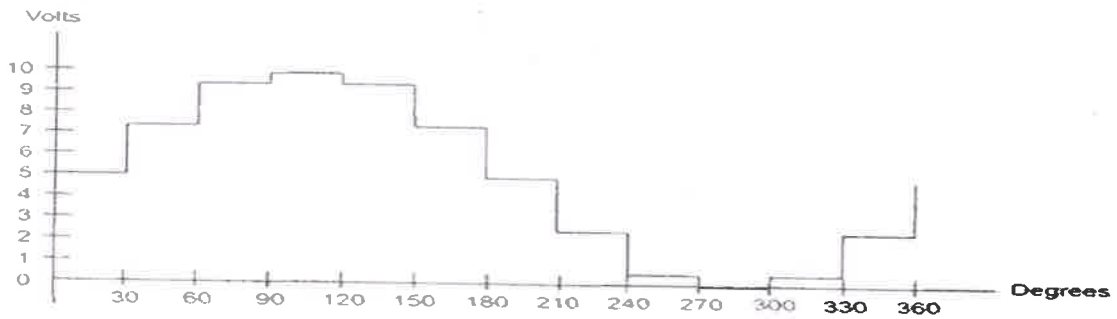
- MCS® 51 Microcontroller Compatible Instruction Set
- Pin Compatible with 44-lead PLCC and 40-lead PDIP MCS 51 Sockets
- Fast Instruction Pipeline
- 16-bit Internal Code Fetch
- 8-bit, Min 2-clock External Code Fetch in Page Mode
- User-selectable Configurations:
  - — External Wait States (0-3 wait states)
  - — Page Mode
- 64K External Code Memory Space
- 64K External Data Memory Space
- ROM/OTPROM Options: 8 Kbytes (SA), 16 Kbytes (SB) or without ROM/OTPROM
- 256 Bytes On-Chip RAM
- Power Management
  - — Idle Mode
  - — Powerdown Mode
- 32 Programmable I/O Lines
- Seven Maskable Interrupt Sources with Four Programmable Priority Levels
- Three Flexible 16-bit Timer/counters
- Hardware Watchdog Timer
- Programmable Counter Array
  - — High-speed Output
  - — Compare/Capture Operation
  - — Pulse Width Modulator
  - — Watchdog Timer
- Programmable Serial I/O Port
  - — Framing Error Detection
  - — Automatic Address Recognition
- High-performance CHMOS Technology
- Static Standby to 16-MHz Operation
- Package Options (PDIP, PLCC)

5

Q.3 b) DAC interfacing with 8051:



V_{out} 5 V + (5 × sin θ)



Angle vs. Voltage Magnitude for Sine Wave

Program:

```
AGAIN: MOV DOPTR,#TABLE
       MOV R2,#COUNT
 BACK: CLR A
       MOVC A,@A+DPTR
       MOV P1,A
       INC DPTR
       DJNZ R2,BACK
       SJMP AGAIN
       ORG 300
TABLE: DB 128,192,238,255,238,192
       DB 128, 64,17,0,17,64,128
```

Q.4 a)

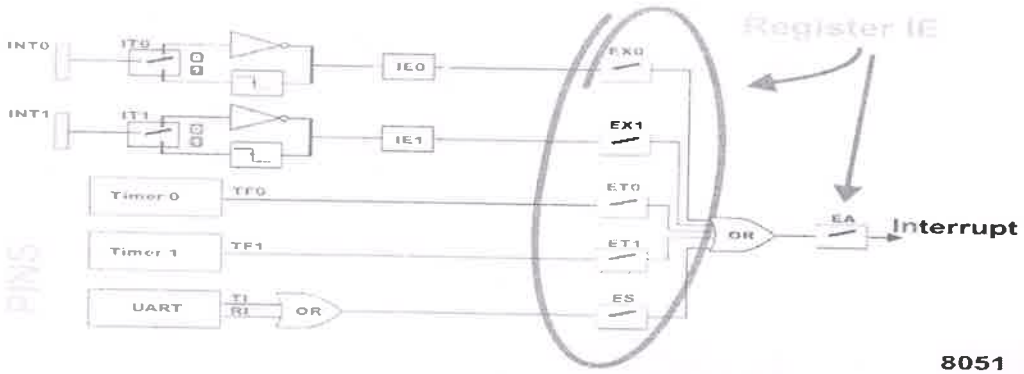
PROGRAM:

```
#include <reg51.h>
sbit MYSW=P2^0;
void main (main)
{
   unsigned char z;
```

```
    unsigned char name1 [] = "MUMBAI";
    unsigned char name2 [] = "UNIVERSITY";
    TMOD=0x20;                          //use Timer 1, 8-bit auto-reload
     TH1=0xFD;                          //9600 baud rate
     SCON=0x50;
     TR1=1;
     if (MYSW= = 0)
  {
      for ( z=0; z<6; z++)
   {
     SBUF=name1 [z];
     while (TI= =0);
     TI=0;
   }
  }
else
    {
     for ( z=0; z<10; z++)
     {
       SBUF= name2 [z];
       while (TI= =0);
       TI=0;
     }
    }
}
```

6

Q.4 b)



	0	X	0	0	0	0	0	0	Value after Reset
IE	EA		ET2	ES	ET1	EX1	ET0	EX0	Bit name
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

	X	X	0	0	0	0	0	0	Value after Reset
IP			PT2	PS	PT1	PX1	PT0	PX0	Bit name
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

Q.5 a)

(7)

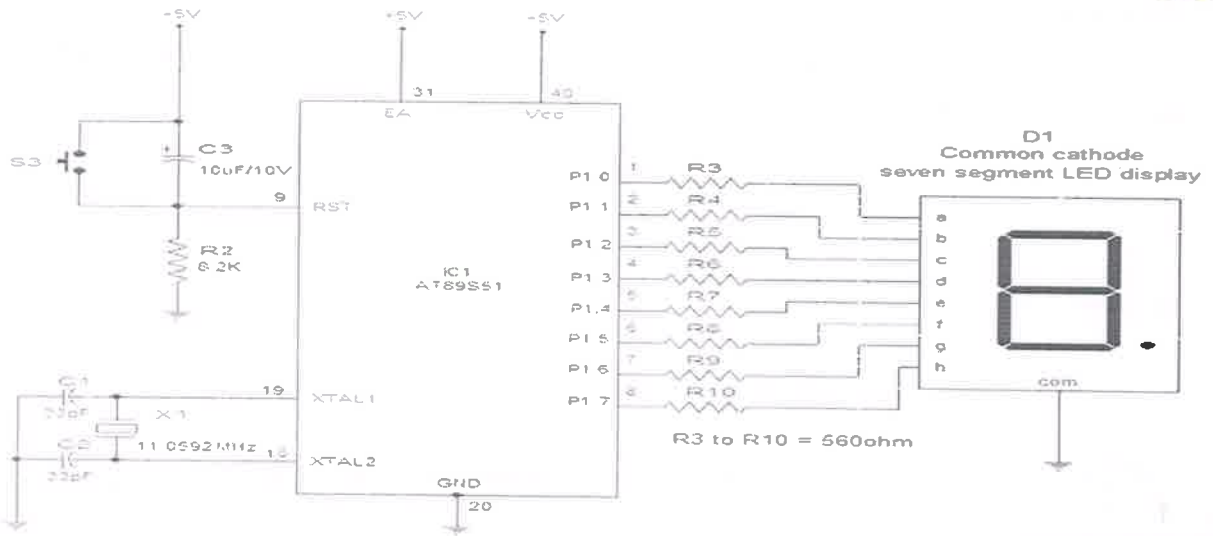
```
#include <reg51.h>
sfr ldata = 0x90;
sbit rs = P2^0;
sbit rw = P2^1;
sbit en = P2^2;
void main()
{
lcdcmd (0x38);
MSDelay (250);
lcdcmd (0x0E);
MSDelay (250);
lcdcmd (0x01);
MSDelay (250);
lcdcmd (0x06);
MSDelay (250);
lcdcmd (0x86);
MSDelay (250);
lcddata ('R');
MSDelay (250);
lcddata ('S');
MSDelay (250);
lcddata ('P');
}
void lcdcmd (unsigned char value)
{
ldata = value;
rs = 0;
rw = 0;
en = 1;
MSDelay (1);
en = 0;
 return;
}
void lcddata ( unsigned char value)
{
ldata = value;
rs = 1;
rw = 0;
en = 1;
MSDelay (1);
en = 0;
return;
}
void MSDelay ( unsigned int itime)
```

```
{
unsigned int i, j;
for (i=0; i<itime; i++)
for (j=0; j=1275; j++);
}
```

Q.5 b)

```
ORG 0000H
CLR A
MOV DPTR,#0400H
MOVC A,@A+DPTR
MOV 60H,A
INC DPTR
CLR A
MOVC A,@A+DPTR
MOV 61H,A
INC DPTR
CLR A
MOVC A,@A+DPTR
MOV 62H,A
INC DPTR
CLR A
MOVC A,@A+DPTR
MOV 63H,A
INC DPTR
CLR A
MOVC A,@A+DPTR
MOV 64H,A
HERE: SJMP HERE
ORG 400H
DB "RAJIV"
END
```

Q.6 a)



Interfacing 7 segment display to 8051

```
ORG 000H //initial starting address
START: MOV A,#00001001B // initial value of accumulator
MOV B,A
MOV R0,#0AH //Register R0 initialized as counter which counts from 10 to 0
LABEL: MOV A,B
INC A
MOV B,A
MOVC A,@A+PC // adds the byte in A to the program counters address
MOV P1,A
ACALL DELAY // calls the delay of the timer
DEC R0//Counter R0 decremented by 1
MOV A,R0 // R0 moved to accumulator to check if it is zero in next instruction.
JZ START //Checks accumulator for zero and jumps to START. Done to check if counting has
        been finished.
SJMP LABEL
DB 3FH // digit drive pattern for 0
DB 06H // digit drive pattern for 1
DB 5BH // digit drive pattern for 2
DB 4FH // digit drive pattern for 3
DB 66H // digit drive pattern for 4
DB 6DH // digit drive pattern for 5
DB 7DH // digit drive pattern for 6
DB 07H // digit drive pattern for 7
DB 7FH // digit drive pattern for 8
DB 6FH // digit drive pattern for 9
DELAY: MOV R4,#05H // subroutine for delay
WAIT1: MOV R3,#00H
WAIT2: MOV R2,#00H
WAIT3: DJNZ R2,WAIT3
DJNZ R3,WAIT2
DJNZ R4,WAIT1
RET
END
```

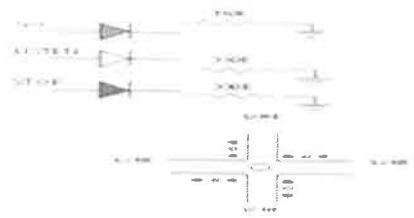
10

Q.6 b)

PIN Assignment with 8051:

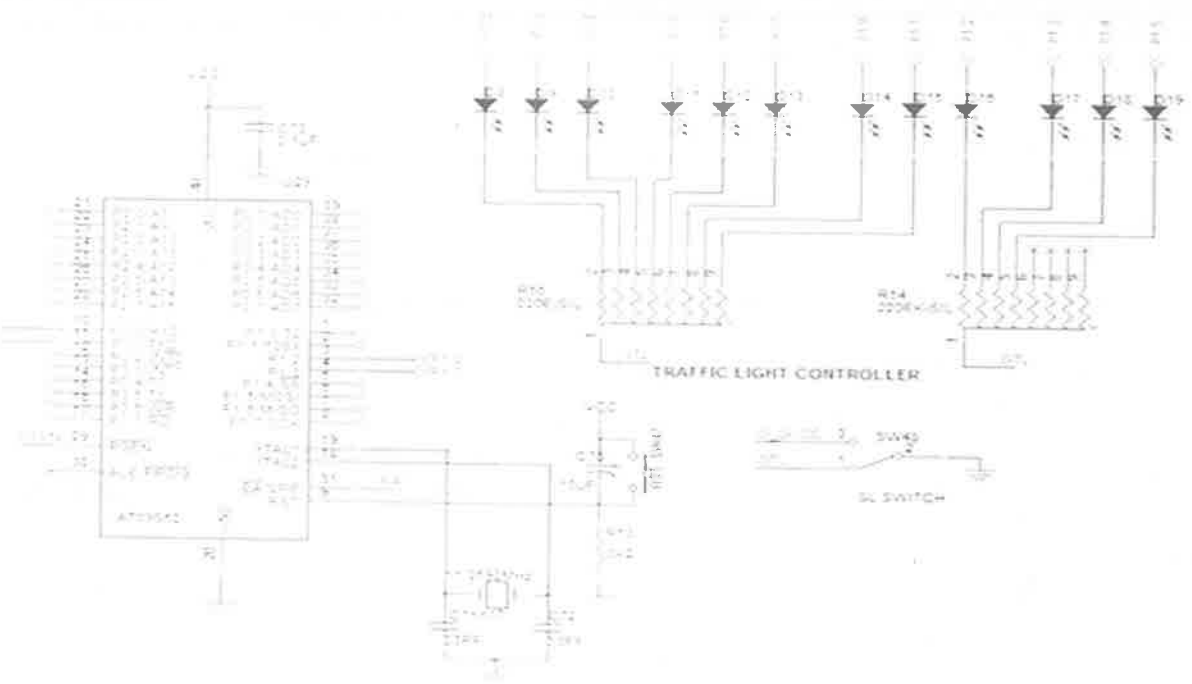
LAN Direction	S051 Lines	LED's
	P3.2	D8-Stop
NORTH	P3.3	D9-Listen
	P3.4	D10-Go
	P3.5	D11-Stop
WEST	P3.6	D12-Listen
	P3.7	D13-Go
	P1.0	D14-Stop
SOUTH	P1.1	D15-Listen
	P1.2	D16-Go
	P1.3	D17-Stop
EAST	P1.4	D18-Listen
	P1.5	D19-Go

Traffic Light Controller



Note : Make SW32 to "*Traffic*" label marking position

Circuit Diagram:



Program to control Traffic signals:

```
#include <reg51.h>
sbit RA =  P1^0;
sbit YA =  P1^1;
sbit GA =  P1^2;
sbit RB =  P3^2;
sbit YB =  P3^3;
sbit GB =  P3^4;
sbit RC =  P3^5;
sbit YC =  P3^6;
sbit GC =  P3^7;
sbit rD =  P1^3;
sbit YD =  P1^4;
sbit GD =  P1^5;
```

```
void Delay (void)
{
   unsigned int i,j;
   for (i=0;i<200;i++)
       for (j=0;j<500;j++);
}
void SuperDelay ()
{
   unsigned int i;
   for (i=0;i<25;i++)
       Delay();
}

void main ()
{
   P3 = 0;
   while (1)
   {
      RA = 0; GA = 1; YA = 0;
      RB = 1; GB = 0; YB = 0;
      RC = 1; GC = 0; YC = 0;
      rD = 1; GD = 0; YD = 0;

      SuperDelay();

      GA = 0; YA = 1;
      Delay();
      RA = 1; GA = 0; YA = 0;
      RB = 0; GB = 1; YB = 0;
      RC = 1; GC = 0; YC = 0;
      rD = 1; GD = 0; YD = 0;
      SuperDelay ();

GB = 0; YB = 1;
      Delay ();
      RA = 1; GA = 0; YA = 0;
      RB = 1; GB = 0; YB = 0;
      RC = 0; GC = 1; YC = 0;
      rD = 1; GD = 0; YD = 0;
SuperDelay ();
      GC = 0; YC = 1;
      Delay();
      RA = 1; GA = 0; YA = 0;
      RB = 1; GB = 0; YB = 0;
      RC = 1; GC = 0; YC = 0;
      rD = 0; GD = 1; YD = 0;
SuperDelay ();
      GD = 0; YD = 1;
      Delay();

   }
}
```

End