



**MODEL ANSWER**

**SUMMER– 17 EXAMINATION**

**Subject Title: Microcontroller and Applications**

Subject Code: **17509**

**Important Instructions to examiners:**

- 1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
- 2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
- 3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills).
- 4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
- 5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
- 6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
- 7) For programming language papers, credit may be given to any other program based on equivalent concept.



Q1.A) Attempt any THREE:

12M

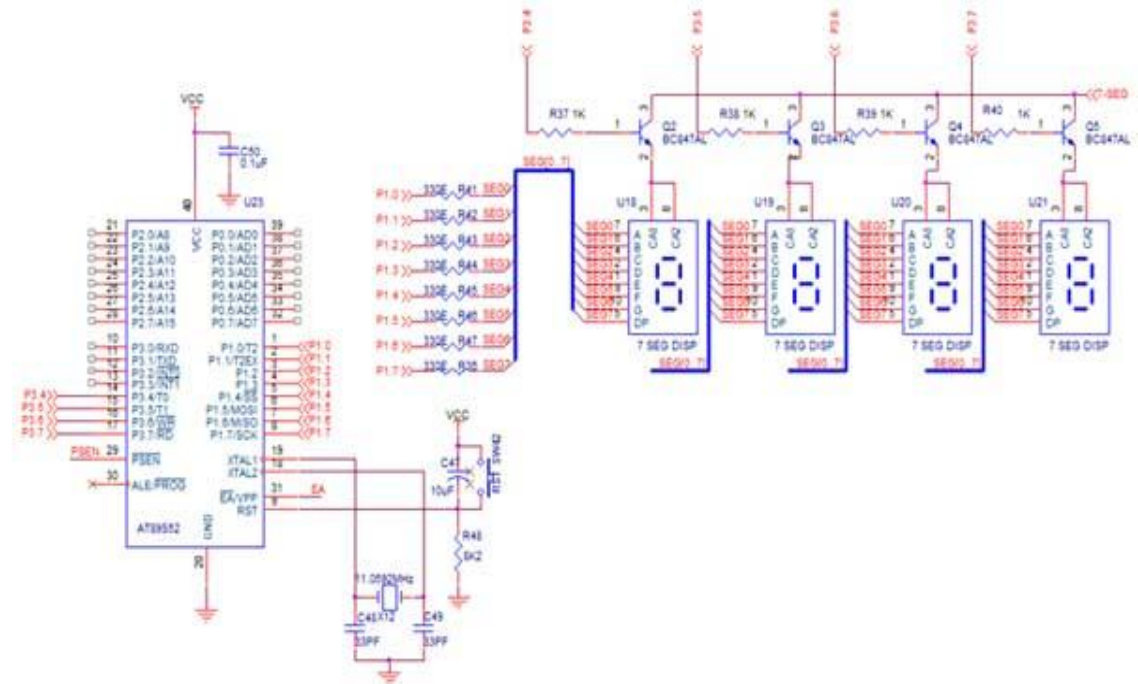
i) Compare between microprocessor and microcontroller on any four points.

Ans: 1M-each correct point

Sr.No	<i>Microprocessor</i>	<i>Microcontroller</i>
1	Microprocessor don't have inbuilt ROM and RAM	Micro controller has inbuilt ROM and RAM
2	Microprocessor don't have inbuilt Timer and Counter	Micro controller has inbuilt Timer and Counter
3	Microprocessor don't have inbuilt UART	Micro controller has inbuilt UART
4	Microprocessor don't have inbuilt I/O PORTs	Micro controller has inbuilt 4 I/O PORTS
5	It has many instructions to move data between memory and CPU.	It has one or two instructions to move data between memory and CPU.
6	It has one or two bit handling instructions.	It has many bit handling instructions.
7	Access times for memory and I/O devices are more.	Less access times for built-in memory and I/O devices.
8	Microprocessor based system requires more hardware.	Micro controller based system requires less hardware reducing PCB size and increasing the reliability.

ii) Draw the interfacing diagram of four 7-segment display with 8051 microcontroller.

Ans: 4M- correct diagram



iii) What are the different data types used in ‘C’? Give their value range.

Ans: 2M- datatypes, 2M-range

**Data types in C Language**

Data types specify how we enter data into our programs and what type of data we enter. C language has some predefined set of data types to handle various kinds of data that we use in our program. These datatypes have different storage capacities. C language supports 2 different type of data types,

**Primary data types**

These are fundamental data types in C namely integer(int), floating(float), charater(char) and void.

**Derived data types**

Derived data types are like arrays, functions, stuctures and pointers. These are dicussed in detail later.

**Integer type**

Integers are used to store whole numbers.

Size and range of Integer type

Type	Size(bytes)	Range
int or signed int	2	-32,768 to 32767
unsigned int	2	0 to 65535
short int or signed short int	1	-128 to 127
long int or signed long int	4	-2,147,483,648 to 2,147,483,647
unsigned long int	4	0 to 4,294,967,295

**Floating type**

Floating types are used to store real numbers.

**Size and range of Integer type**

Type	Size(bytes)	Range
------	-------------	-------



Float	4	3.4E-38 to 3.4E+38
Double	8	1.7E-308 to 1.7E+308
long double	10	3.4E-4932 to 1.1E+4932

### Character type

Character types are used to store characters value.

### Size and range of Integer type

Type	Size(bytes)	Range
char or signed char	1	-128 to 127
unsigned char	1	0 to 255

### Void type

Void type means no value. This is usually used to specify the type of functions.

iv) **State the alternate pin functions of port3 of 8051 microcontroller.**

**Ans: ½ M-each pin function**

Pin	Name	Alternate Function
P3.0	RXD	Serial input line(Receive)
P3.1	TXD	Serial output line(Transmit)
P3.2	— INT0	External interrupt 0
P3.3	— INT1	External interrupt 1
P3.4	T0	Timer 0 external input
P3.5	T1	Timer 1 external input
P3.6	— WR	External data memory write strobe
P3.7	— RD	External data memory read strobe

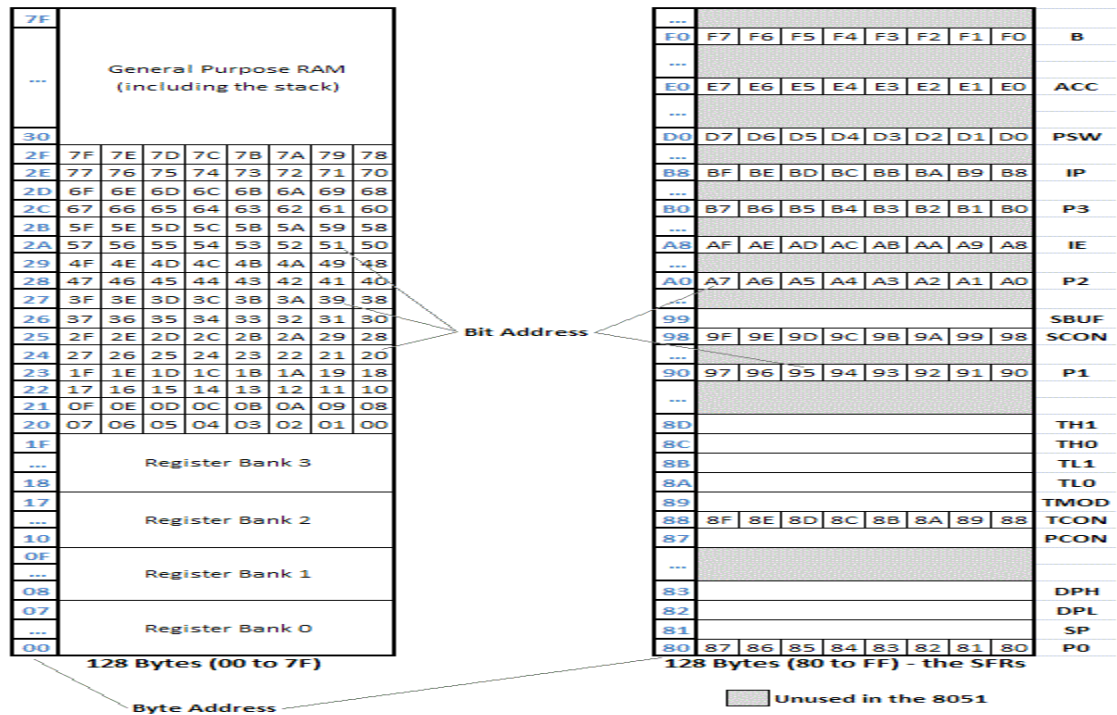
**B) Attempt any ONE:**

**6M**

**a) Draw the organization of internal RAM of 8051 and describe in brief.**

**Ans: 4M-diagram, 2M-explanation**

There are 256 bytes of internal RAM on the 8051.  $2^8 = 256$ , therefore the internal RAM address bus is 8 bits wide and internal RAM locations go from 00H to FFH. The first 128 locations (00H to 7FH) of internal RAM are used by the programmer for storing data while the second 128 locations (80H to FFH) are the Special Function Registers (SFRs). The diagram below is a summary of the 8051 on-chip RAM.



### Register Banks

There are four register banks from 00H to 1FH. On power-up, registers R0 to R7 are located at 00H to 07H. However, this can be changed so that the register set points to any of the other three banks (if you change to Bank 2, for example, R0 to R7 is now located at 10H to 17H).

### Bit-addressable Locations

The 8051 contains 210 bit-addressable locations of which 128 are at locations 20H to 2FH while the rest are in the SFRs. Each of the 128 bits from 20H to 2FH have a unique number (address) attached to them, as shown in the table above. The 8051 instruction set allows you to set or reset any single bit in this section of RAM.

With the general purpose RAM from 30H to 7FH and the register banks from 00H to 1FH, you may only read or write a full byte (8 bits) at these locations.

However, with bit-addressable RAM (20H to 2FH) you can read or write any single bit in this region by using the unique address for that bit. We will later see that this is a very powerful feature.

### Special Function Registers (SFRs)

Locations 80H to FFH contain the special function registers. As you can see from the diagram above, not all locations are used by the 8051 (eleven locations are blank). These extra locations are used by other family members (8052, etc.) for the extra features these microcontrollers possess.



Also note that not all SFRs are bit-addressable. Those that are having a unique address for each bit.

**b) Explain any four assembler directives with suitable examples.**

**Ans: 4M-each correct directive ,1/2 M -each example**

i) **ORG:-** ORG stands for Origin

Syntax:                    

ORG
-----

Address
---------

The ORG directive is used to indicate the beginning of the address. The number that comes after ORG can be either in hex or in decimal. If the number is not followed by H, it is decimal and the assembler will convert it to hex. Some assemblers use “. ORG” (notice the dot) instead of “ORG” for the origin directive.

ii) **DB:-** (Data Byte)

Syntax:                    

Label:
--------

DB
----

Byte
------

Where byte is an 8-bit number represented in either binary, Hex, decimal or ASCII form. There should be at least one space between label & DB. The colon (:) must present after label. This directive can be used at the beginning of program. The label will be used in program instead of actual byte. There should be at least one space between DB & a byte. Following are some DB examples:

```

                ORG 500H
DATA1:  DB  28                ;DECIMAL(1C in hex)
DATA2:  DB  00110101B        ;BINARY (35 in hex)
DATA3:  DB  39H              ;HEX

                ORG 510H
DATA4:  DB  "2591"           ;ASCII NUMBERS

                ORG 518H
DATA6:  DB  "My name is Joe" ;ASCII CHARACTERS

```

iii) **EQU: Equate**

It is used to define constant without occupying a memory location.

Syntax:                    

Name
------

EQU
-----

Constant
----------

By means of this directive, a numeric value is replaced by a symbol.

For e.g. MAXIMUM EQU 99 After this directive every appearance of the label “MAXIMUM” in the program, the assembler will interpret as number 99 (MAXIMUM=99). iv) **END:**

This directive must be at the end of every program meaning that in the source code anything after the END directive is ignored by the assembler.

This indicates to the assembler the end of the source file(asm).

Once it encounters this directive, the assembler will stop interpreting program into machine code.

e.g. END ; End of the program.



**Q2. Attempt any TWO:**

**16M**

**a) Describe the addressing modes of 8051 microcontroller with 2 examples of each.**

**Ans: 8M- correct explanation**

There are a number of addressing modes available to the 8051 instruction set, as follows:

1. Immediate Addressing mode
2. Register Addressing mode
3. Direct Addressing mode
- 4 Register Indirect addressing mode
5. Relative Addressing mode
6. Absolute addressing mode
7. Long Addressing mode
8. Indexed Addressing mode

1) Immediate Addressing mode:

Immediate addressing simply means that the operand (which immediately follows the Instruction op. code) is the data value to be used.

For example the instruction:

MOV A, #25H ; Load 25H into A

Moves the value 25H into the accumulator. The # symbol tells the assembler that the immediate addressing mode is to be used.

**2 ) Register Addressing Mode:**

One of the eight general-registers, R0 to R7, can be specified as the instruction Operand.

The assembly language documentation refers to a register generically as Rn.

An example instruction using register addressing is :

ADD A, R5 ; Add the contents of register R5 to contents of A (accumulator)

Here the contents of R5 are added to the accumulator. One advantage of register addressing is that the instructions tend to be short, single byte instructions.

**3) Direct Addressing Mode:**

Direct addressing means that the data value is obtained directly from the memory location specified in the operand.

For example consider the instruction:

MOV R0, 40H; Save contents of RAM location 40H in R0.

The instruction reads the data from Internal RAM address 40H and stores this in theR0.

Direct addressing can be used to access Internal RAM, including the SFR registers.

**4) Register Indirect Addressing Mode:**

Indirect addressing provides a powerful addressing capability, which needs to be appreciated.

An example instruction, which uses indirect addressing, is as follows:

MOV A, @R0; move contents of RAM location whose address is held by R0 into A

Note the @ symbol indicated that the indirect addressing mode is used. If the data is inside the CPU, only registers R0 & R1 are used for this purpose.

**5) Relative Addressing Mode:**

This is a special addressing mode used with certain jump instructions. The relative address, often referred to as an offset, is an 8-bit signed number, which is automatically added to the PC to make the address of the next instruction. The 8-bit signed offset value gives an address range of + 127 to -128 locations.

Consider the following example:

### SJMP LABEL\_X

An advantage of relative addressing is that the program code is easy to relocate in memory in that the addressing is relative to the position in memory.

#### 6) Absolute addressing Mode:

Absolute addressing within the 8051 is used only by the AJMP (Absolute Jump) and ACALL (Absolute Call) instructions.

**7) Long Addressing Mode:** The long addressing mode within the 8051 is used with the instructions LJMP and LCALL. The address specifies a full 16 bit destination address so that a jump or a call can be made to a location within a 64KByte code memory space ( $2^{16} = 64K$ ).

An example instruction is:

LJMP 5000h; full 16 bit address is specified in operand

#### 8) Indexed Addressing Mode:

With indexed addressing a separate register, either the program counter, PC, or the data pointer DPTR, is used as a base address and the accumulator is used as an offset address. The effective address is formed by adding the value from the base address to the value from the offset address. Indexed addressing in the 8051 is used with the JMP or MOVX instructions. Look up tables are easy to implement with the help of index addressing.

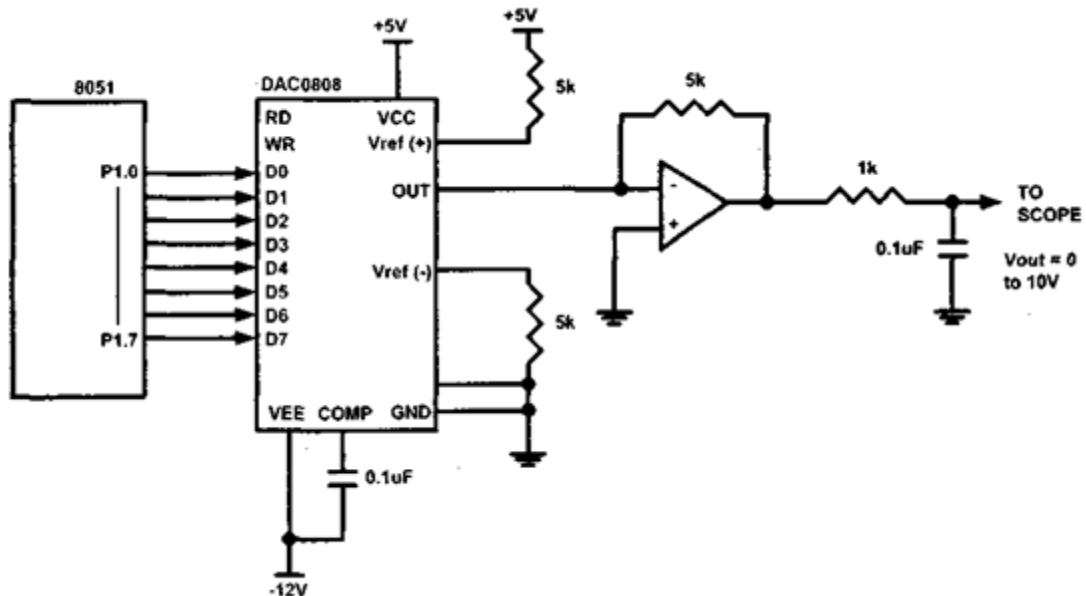
Consider the example instruction:

MOVX A, @A+DPTR

MOVX is a move instruction, which moves data from the external code memory space. The address operand in this example is formed by adding the content of the DPTR register to the accumulator value. Here the DPTR value is referred to as the base address and the accumulator value is referred to as the index address.

b) Draw neat diagram to interface 8 bit DAC 0808 with 8051. Write 'C' language program to generate staircase waveform.

Ans: 4M- diagram, 4M –correct program







**DAC interfacing:**

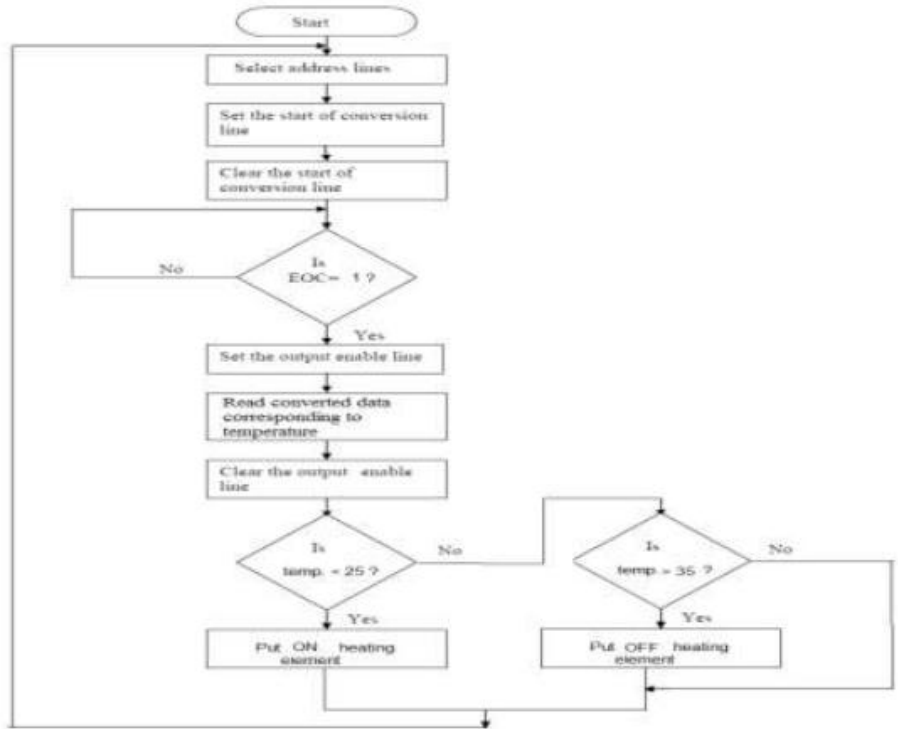
```
#include<reg51.h>
void Delay();                // delay for frequency
{
    int c;
    for(c=0;c<1000;c++);
}

void main()
{
    P1=0x00;                // P1 as output port
    while(1)
    {
        P1=0x00;            // step by step increment
        Delay();
        P1=0x20;            // values sent to P2
        Delay();
        P1=0x40;
        Delay();
        P1=0x80;
        Delay();
    }
}
```

- c) Assuming temperature control system using LM35 as temperature sensor and ADC 0809. Draw flowchart of this system to maintain temperature within 25°C to 30°C.

**Ans: 8M- correct flowchart**

(Note: Any other correct logic can be used)



**3. Attempt any FOUR:**

a) Draw the format of PSW SFR and describe the function of each bit.



Ans:(2M format,2 M functions)

PSW register –

**Format:-**

D<sub>7</sub>

D<sub>0</sub>

Cy	AC	F0	RS1	RS0	OV	X	P
----	----	----	-----	-----	----	---	---

Bit7: Carry/Borrow bit: When two 8-bit operands are added, the result may exceed 8-bit and the 9<sup>th</sup> bit is copied in the carry bit. During subtraction, if the borrow occurs, the carry bit is set and otherwise, it is cleared.

Bit 6: Auxiliary carry/borrow bit: This bit indicates a carry from lower nibble during 8 bit addition. If AC flag is set, it means there is a carry from 3<sup>rd</sup> to 4<sup>th</sup> bit position.

Bit5: F0: Flag 0 is available to user for general purpose.

Bit 4-3: RS1:RS0: Register Bank selects bits

RS1	RS0	Bank selected
0	0	Register Bank 0
0	1	Register Bank 1
1	0	Register Bank 2
1	1	Register Bank 3

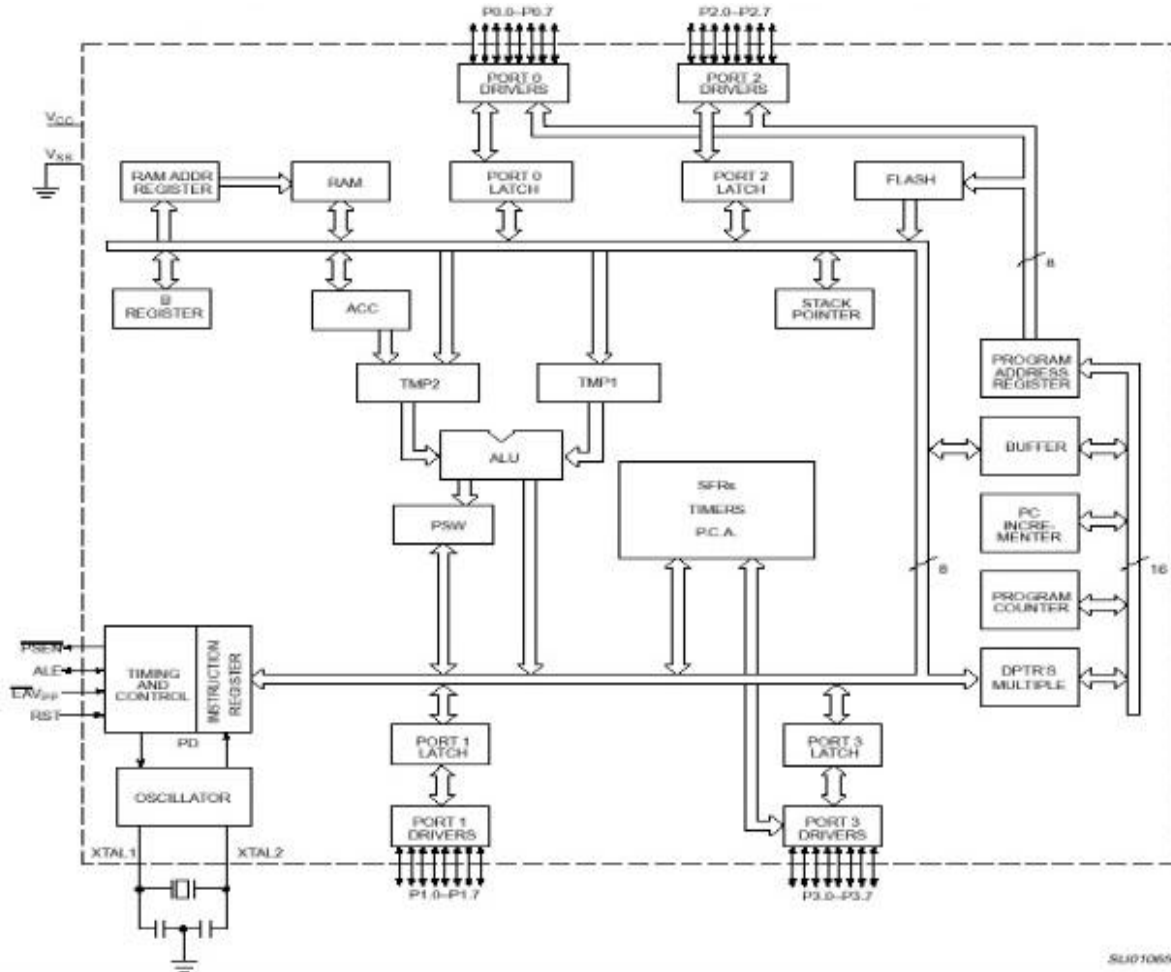
Bit 2: OV: Over flow flag is used to detect errors in signed arithmetic operations. When two signed numbers are added, if the result exceeds the destination, overflow flag is set, else it is reset.

Bit1: Undefined flag

Bit 0: Parity Flag: P flag is set, if the result contains an odd number of 1 bit, else it is reset, if the result contains an even number of 1 bit.

b) Draw the block diagram of internal architecture of 8051 microcontroller.

Ans: (4 M correct answer)



c) Write the instructions for following operations using 'C' operators.

- i) Bitwise shift data left 4 times
- ii) Bitwise shift data right 3 times

Ans: (2 M each correct answer)  
 $P0 = 0x56 \gg 4$ ; shift 3 times to right  
 $P0 = 0x56 \ll 4$ ; shift 4 times to left



d) Compare RISC and CISC machines.

Ans: (1M each correct point)

d) Compare RISC and machines.

Ans) (any 4 mark)

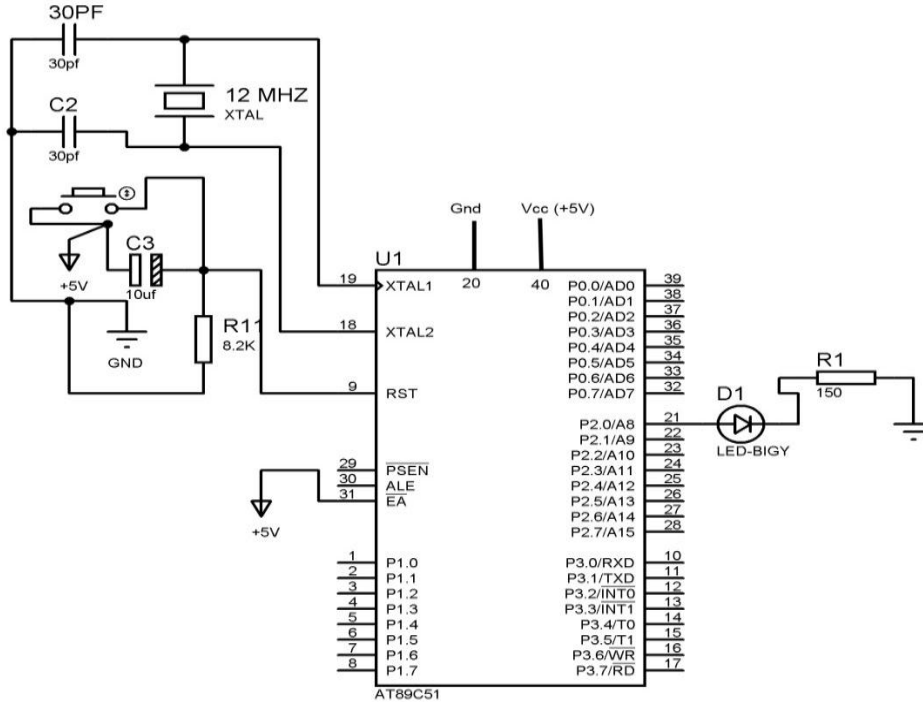
RISC	CISC
Reduced instruction set computer	Complex instruction set computer.
Instruction set is simple and limited.	Instruction set is very large.
Instruction set is not flexible, hence the program is long.	Instruction set is flexible, hence the program is short
Provides few addressing modes normally register.	Provides many addressing modes
Large memory is required	Less memory required
Provides large number of registers	No. of registers are less.
Process architecture & control unit is simple	Processor architecture and control unit is complicated.
Instruction are shorter, hence execution speed is fast.	Instructions are lengthy hence execution speed is slow.
External memory is accessed rarely	External memory is accessed frequently
Each instruction required few bus cycle.	Each instruction requires many bus cycle.
e. g. ARM, ATMEL, AVR, MIPS, PIC, POWER PC, etc.	Interx86, Motorola, 68000 series.

between CISC

points each 1

e) Draw the interfacing diagram to connect 8 LEDs to port 2 of 8051 microcontroller. Also write 'C' language program to turn ON and OFF LEDs with some delay.

**Ans: (2M-diagram, 2M- for correct program)**



(Note: LEDs should be connected to all the 8 lines of Port 2)

**NOTE: Program may change. Student can also use the other logic.**

Please check the logic and understanding of students.

```
#include <reg51.h>

void delay (unsigned int);

void main (void)
{
while(1) //repeat loop
{
```

```

P2=0xff; //LEDs will turn ON

delay (400); //add delay

P2=0x00; //LEDs will turn OFF

delay (400); //add delay

}

}

void delay (unsigned inti)

{

Unsignedint x,y;

for(x=0;x<i; x++)

for (y=0;y<1275;y++);

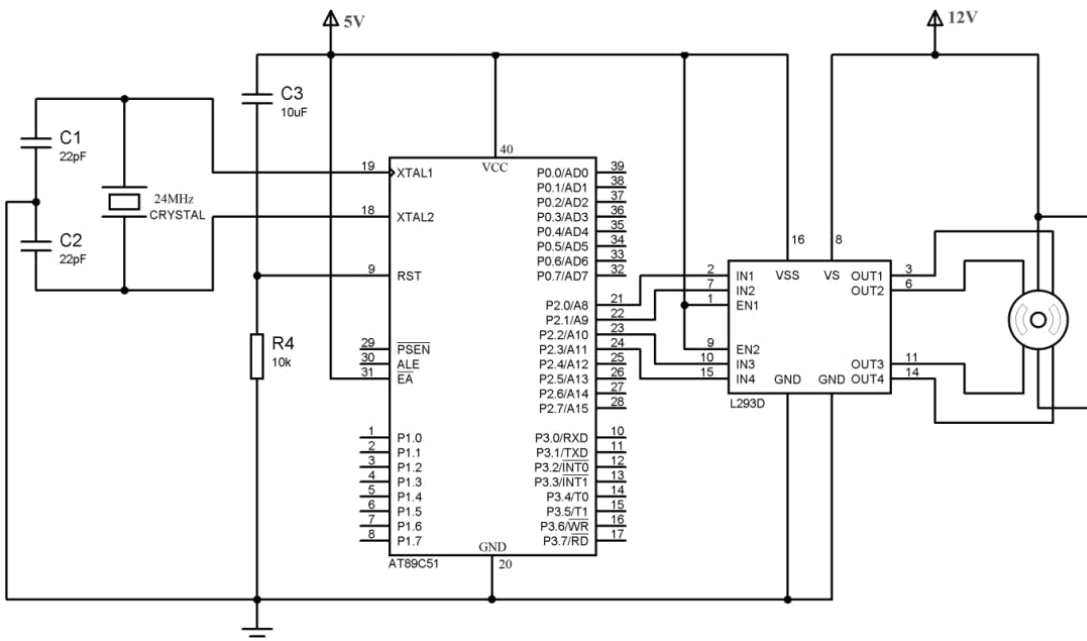
}

```

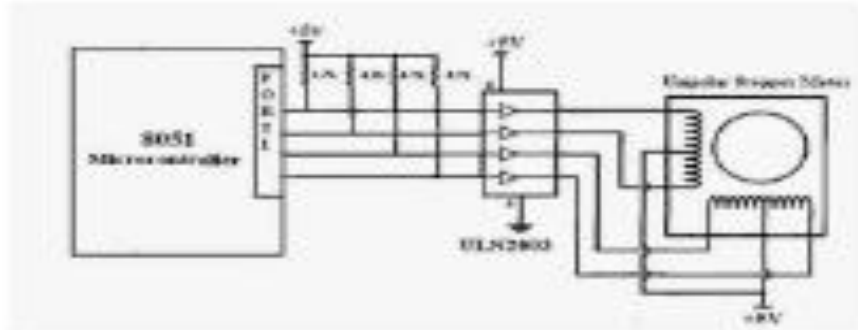
**4. A) Attempt any THREE:**

a) Draw the interfacing of stepper motor with 8051 microcontroller. Also draw the flow chart to rotate the motor through 360 degree in the clockwise

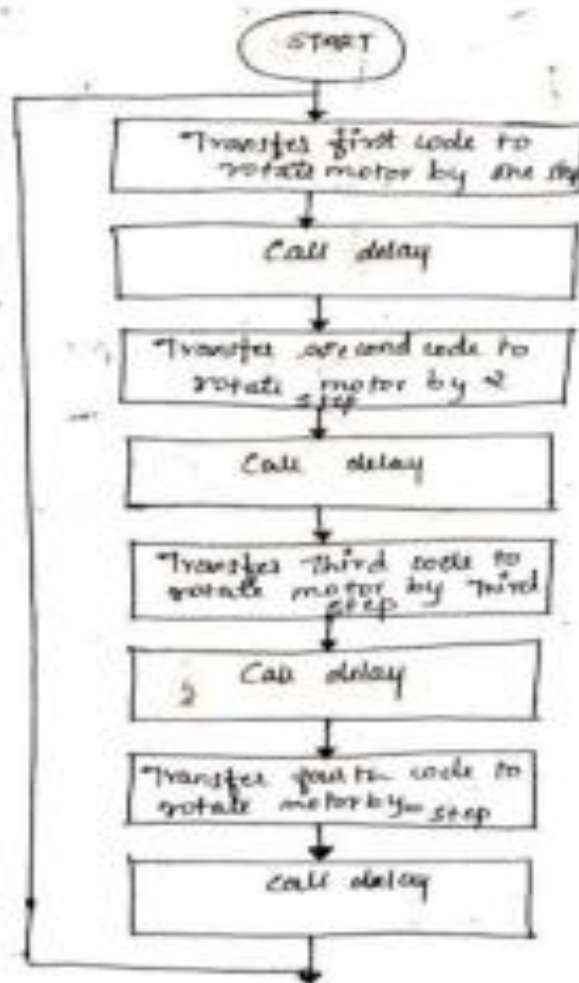
Ans : (2M correct diagram,2M flowchart)



OR



Flow chart:







b) Write 'C' language program for 8051 microcontroller to toggle port 1.0 with some delay.

Ans :(Proper relevant program – 4 M)

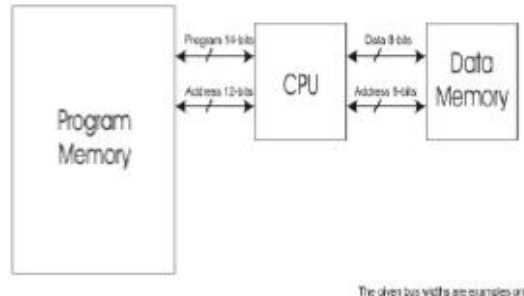
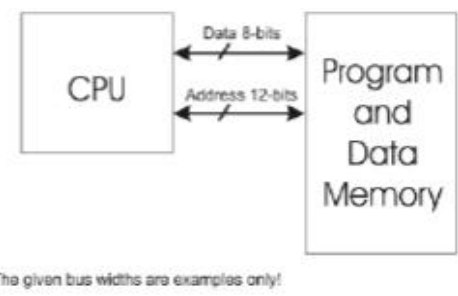
**Program:**

```
# include <reg51.h>
sbit sw=P1^0;
void delay(int) // delay function
void main (void)
{
while (1) // infinite while loop
{
{
sw=1; //port pin is on
delay (100); //delay
sw= 0 ; //port pin is off
delay (100); //delay

}
}
void delay(int k)
{
int i,j;
for(i=0;i<k;i++);
for(j=0;j<100;j++);
}
}
```

c) Compare between Von-Neumann and Harvard architecture with neat diagram.

Ans (1 M each correct point)

Sr. No	Harvard Architecture	Van Neumann's Architecture
1.	 <p style="text-align: center; font-size: small;">The given bus widths are examples only!</p>	 <p style="text-align: center; font-size: small;">The given bus widths are examples only!</p>
2.	The Harvard architecture uses physically separate memories for their instructions and data.	The Van Neumann's architecture uses single memory for their instructions and data.
3.	Requires separate & dedicated buses for memories for instructions and data	Requires single bus for instructions and data.
4.	Its design is complicated	Its design is simpler.
5.	Instructions and data can be fetched simultaneously as there is separate buses for instructions and data which increasing operation bandwidth.	Instructions and data have to be fetched in sequential order limiting the operation bandwidth.

d) List the interrupts of 8051 microcontroller with their vector address and priority upon reset. Explain SFR used to enable interrupts.

Ans: (2M List, 2M SFR explanation)

Interrupt Source	Vector address	Interrupt priority
External Interrupt 0 –INT0	0003H	1
Timer 0 Interrupt	000BH	2
External Interrupt 1 –INT1	0013H	3
Timer 1 Interrupt	001BH	4
Serial Interrupt	0023H	5



THE SFR used to enable interrupts is the IE SFR.

**IE: INTERRUPT ENABLE REGISTER. BIT ADDRESSABLE.**

If the bit is 0, the corresponding interrupt is disabled. If the bit is 1, the corresponding interrupt is enabled.

EA	—	ET2	ES	ET1	EX1	ET0	EX0
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.					
—	IE.6	Not implemented, reserved for future use.*					
ET2	IE.5	Enable or disable the Timer 2 overflow or capture interrupt (8052 only).					
ES	IE.4	Enable or disable the serial port interrupt.					
ET1	IE.3	Enable or disable the Timer 1 overflow interrupt.					
EX1	IE.2	Enable or disable External Interrupt 1.					
ET0	IE.1	Enable or disable the Timer 0 overflow interrupt.					
EX0	IE.0	Enable or disable External Interrupt 0.					

\*User software should not write 1s to reserved bits. These bits may be used in future MCS-51 products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

**B) Attempt any ONE:**

a) Write an assembly language program for 8051 microcontroller to find the average of ten 8 bit numbers stored in internal RAM location 20H onwards. Store the result in 31H.

Ans : (6M correct program)

```
                ORG 0000h
                LMP MAIN
                ORG 0030H
MAIN:          MOV R1,#0AH    ;LOAD COUNTER OF 10
                MOV R0,#20H
                XRA A         ; CLEAR ACC AND CY FLAG
NEXT:         ADD A,@R0      ;ADD NUMBAR TO ACC AND STORE SUM IN ACC
                INC R0
                DJNZ R1,NEXT
                MOV B,#0AH
                DIV A,B
                MOV 30H,A
                RET
```

b) Compare 8051 and 8052 microcontroller on the following points:



- i) Onchip ROM
- ii) Onchip RAM
- iii) Number of timers
- iv) Interrupts

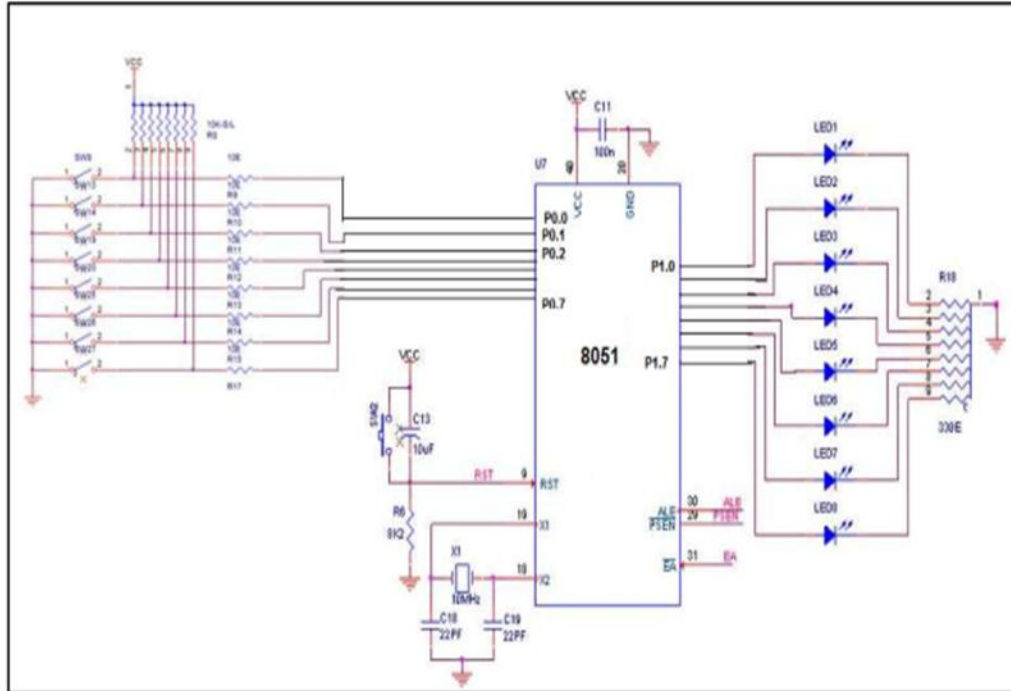
Parameter	8051	8052
Onchip RAM	4 KB	8 KB
On chip ROM	128 Bytes	256 Bytes
No, of timers	Two timers	Three timers
Interrupts	Five interrupts	Six interrupts

5. Attempt any TWO:

16

- a) Draw the interfacing diagram to interface 8 switches to port 0 and 8 LEDs to port 1.  
Write 'C' language program to read switch status and display the same on the LEDs.

Ans:- ( Interfacing diagram- 4 mks, program – 4 mks)



( Note : LEDs can be connected in common anode mode also )

'C' Language program:

```
#include <reg51.h>
void main(void)
{
    unsigned char mybyte;
    P0=0xFF; //make Port0 input port
    P1=0X00; // make Port1 output port
    while (1)
    {
        mybyte=P0; //get a byte from P0
        P1= ~ mybyte; //send compliment of it to P1
    }
}
```

**Note:** When switch is not pressed port pin of P0 status is logic 1 and when switch is pressed port pin of P0 is logic 0. So we have to complement the status of P0 i.e mybyte variable.

b) Write and assembly language program to transfer array of ten numbers stored in memory location 50 H to memory location 60 H.

Ans:- ( Algorithm – 4 mks, program- 4 mks)



### Algorithm:

1. Start
2. Initialize count R2=0AH
3. Initialize source pointer R0 with starting address 50h
4. Initialize destination pointer R1 with starting address 60H.
5. Move number pointed by R0 to accumulator
6. Move number from accumulator to destination address pointed by R1
7. Increment source pointer R0
8. Increment destination pointer R1
9. Decrement count & if count not equal to zero repeat step 5 to step 9
10. Else stop

### Program

```
Assembly language program:
MOV R2,#0AH ; load count 10 in register R2
MOV R0,#50H; load source address in register R0
MOV R1,#60H ; load destination address in register R1
UP:  MOV A , @R0 ; take source data pointed by R0 in Accumulator ✓
     MOV @R1 , A ; Move data in accumulator to address pointed by R1
     INC R0      ; Increment source pointer R0 ✓
     INC R1      ; Increment destination pointer R1 ✓
     DJNZ R2 , UP ;Decrement count and jump UP if count = 0 ✓
HERE: SJMP HERE ; terminate program ✓
```

c) Write an assembly language program for 8051 to generate square wave of 1 KHZ on port pin P1.1. User timer 1 and assume crystal frequency to be 12 MHZ. Clearly show the necessary calculation with comments.



Ans:- ( Calculations – 2 mks, program- 6 mks)

Calculations:-Crystal frequency= 12 MHz

i/p clock=  $12 \times 10^6 / 12 = 1 \text{ MHz}$

Tin= 1  $\mu$ sec

For 1 KHz square wave

Fout=1 KHz

Tout=  $1/1 \times 10^3 = 1 \text{ msec}$

Consider half of it=Tout=500  $\mu$ sec

N=Tout/Tin=500/1=500  $\mu$ sec

65536-500=65036=FE0CH

Program

MOV TMOD,01H ; Set timer 0 in mode 1,ie 16 bit timer

L2: MOV TL0,#0C H ; load TL register with LSB of count

MOV TH0,#0FFH ; Load TH register with MSB of count

SETB TR0 ;Start timer 0

L1: JNB TF0,L1 ; Poll till timer roll over

CLR TR0 ; Stop timer 0

CPL P1.1 ; Complement port 1.1 to get high or low

CLR TF0 ; Clear timer flag 0

SJMP L2 ; Reload timer with count as mode 1,is not auto reload

6. Attempt any FOUR: 16

a) Draw the internal structures of port 0 of 8051. Label the same. Describe the process to ready port 0 pin status.

Ans:- ( Diagram with proper label- 2 M ,Procedure- 2 M)

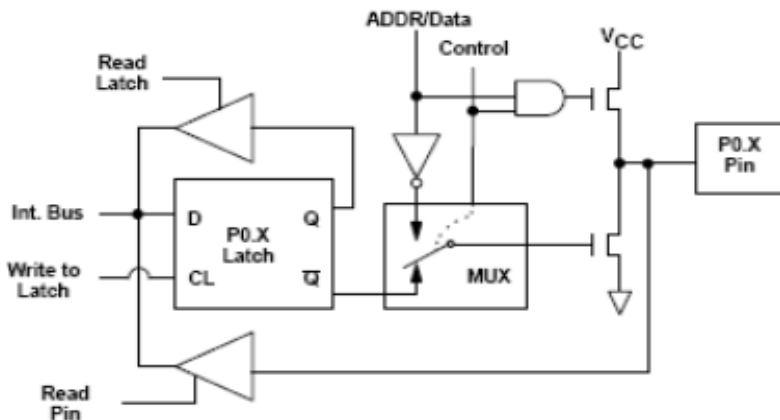


Port 0 can be used as

- Simple I/O Port and
- Bidirectional low order address and data bus ( AD0 – AD7) for external memory access.

Port -0 has 8 pins (P0.0-P0.7).

The structure of a Port-0 pin is shown in fig below



**Port0 pin circuit**

### Process to read pin status:

When the port is used as an input port, '1' must be written to the latch. In this situation both the output MOSFETs are 'off'. Hence the output pin floats in high impedance state, and pin is connected to input buffer. Data on the pin can be read by asserting "Read pin" signal when read pin instruction is executed

- Write 'c' language program for 8051 to transmit message "MSBTE" serially a 9600 baud, 8 bit data, 1 stop bit, assuming crystal frequency to be 11,0592 MHZ.

Ans:- ( Proper program- 4 mks)

```
#Include < REG51.H>
Void main(void)
{
Unsigned char text[]= " MSBTE";
Unsigned char i;
TMOD=0x20;
TH1=0Xfd;
SCON=0x50;
TR1=1;
```





```
While (1)
{
For (i=0: i<5;i++)
{
SBUF=text[I];
While)T1==0);
T1=0;
}
}
}
```

c) **With neat algorithm, write an assembly language program to add two BCD numbers stored at internal RAM location 30 H and 31 H. Store the result in internal RAM location 40 H.**

**Ans:- ( Algorithm- 2 M, program-2 M)**

### **Algorithm**

Initialize the first memory pointer  
Initialize the second memory pointer  
Move the first data into accumulator  
Add the first and second data  
Adjust result to BCD  
Store the result in memory location

### **Program**

```
MOV R0,#30H ; Initialize the first memory pointer
MOV R1, #31H ; Initialize the second memory pointer
MOV A,@R0 ;Move the contents of 30H into accumulator
ADD A,R1 :Add both the numbers
DAA ; Adjust result in BCD
MOV 40H,A ; Store the result in 40H
```

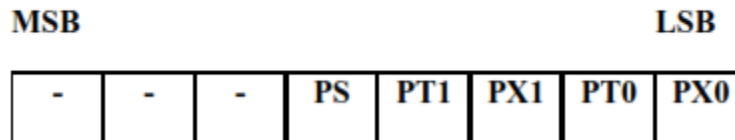


END ; Stop

d) Draw and describe the format of IP SFR of 8051.

.Ans:- Diagram- 2 mks, description- 2 mks)

**The IP ( INTERRUPT PRIORITY) SFR FORMAT**



DESCRIPTION:

Bit	Name	Description
IP.7	-	Reserved
IP.6	-	Reserved
IP.5	-	Reserved
IP.4	PS	Serial Port Interrupt priority
IP.3	PT1	Timer 1 Interrupt priority (TF1)
IP.2	PX1	External Interrupt 1 priority (INT1)
IP.1	PT0	Timer 0 Interrupt priority (TF0)
IP.0	PX0	External Interrupt 0 priority (INT0)

e) Draw interfacing diagram to interface relay at P1.0 and opto-isolator at P1.7 of 8051 microcontroller.

Ans:- ( Proper interfacing diagram- 4 mks)

