CBCS SCHEME

USN | | | | | | | | | | |

21EE43

Fourth Semester B.E. Degree Examination, June/July 2023
Microcontrollers

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. Explain the block diagram 8051 microcontroller. (10 Marks)
b. List the features of 8051 microcontroller. (06 Marks)
c. Explain the P.S.W and flag's bits. (04 Marks)

OR

2 a. Explain the various addressing modes of 8051 microcontroller with examples. (10 Marks)
b. Explain with the help of diagram, how to interface external code memory to 8051 microcontroller. (10 Marks)

Module-2

3 a. What are assembler directives? Explain various assembler directives. (08 Marks)
b. Explain the following Instructions of 8051 with examples:
(i) DJNZ R_, rel (ii) DA A (iii) MOVX A, @DPTR
(iv) SWAP A (v) XCHD @R_i (vi) INC R_n (12 Marks)

OR

4 a. With a neat diagram explain the range of JUMP and CALL Instruction. (08 Marks)
b. Write an 8051 assembly program to find average of five numbers stored from Internal Data Memory address 40H. (08 Marks)
c. Explain Rotate Instruction of 8051 with examples. (04 Marks)

Module-3

5 a. Explain the various data types in 8051 C. (08 Marks)
b. Write an 8051 C program to toggle the bits of P_1 ports continuously with a 250ms delay. (06 Marks)
c. Write an 8051 C program to toggle bit P2.4 continuously without disturbing the rest of bits of P_2. (06 Marks)

OR

6 a. Explain TMOD register. (06 Marks)
b. Explain Mode-1 programming of 8051 Timer. (06 Marks)
c. Write an 8051 C program to convert packed BCD to ASCII and display the bytes on P_1 and P_2. (08 Marks)

Module-4

7 a. What is serial data communication? Explain simplex, half duplex and full duplex transfer. (08 Marks)
b. Draw and explain the interface of RS232 to 8051 using MAX232. (06 Marks)
c. Write a C-program the 8051 to transfer the letter 'C' serially at 9600 baud continuously. Use 8-bit data and 1 stop bit. (06 Marks)

OR

8 a. What is an Interrupt? List the various interrupts of 8051 with their corresponding vector address. (08 Marks)

b. Explain the bit status of SCON Register. (06 Marks)

c. Write a C-program that continuously get a single bit of data from P1.7 and send it to P1.0. While simulation creating a square wave of 200 μs period on P1.0. Use timer-0 to create square wave. Assume XTAL = 11.0592 MHz. (06 Marks)

Module-5

9 a. Explain pin diagram of 8255 chip. (07 Marks)

b. Draw and explain the interface diagram of LCD with 8051 microcontroller. (07 Marks)

c. Write an C-program to rotate stepper motor continuously in clockwise direction. (06 Marks)

OR

10 a. Draw the block diagram to show how 8051 is connected to DAC 0808 at port P. (07 Marks)

b. Write a C-program to generate a sine wave using DAC. (06 Marks)

c. Explain the Internal architecture of ADC 0804. (07 Marks)

* * * * *

Scheme of evaluation. 01

Prof. Ravindra Molekar

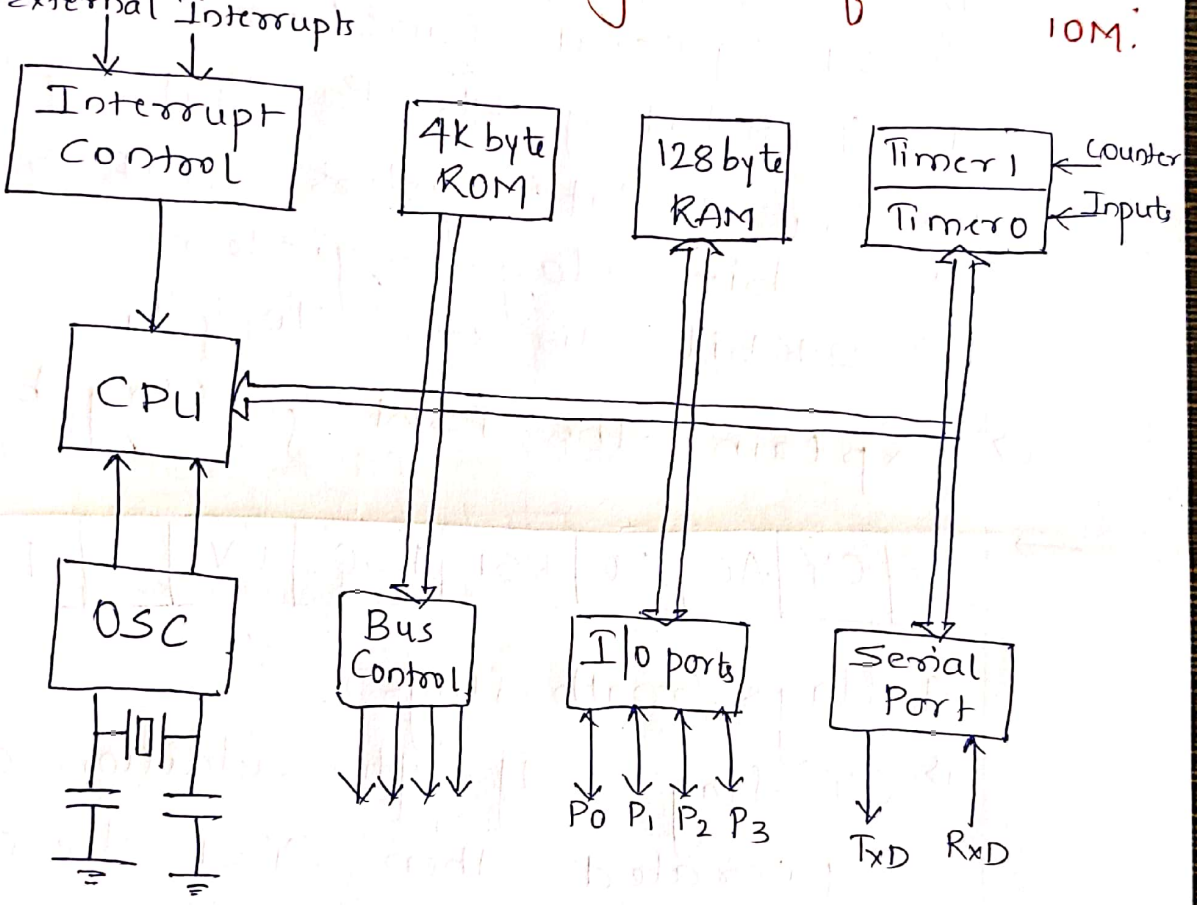
4th Sem Microcontroller

21EE43

June | July 2023 Examination

Module 1.

1a. Explain Block Diagram of 8051 μC? 10M.



8051 μC consists of an 8-bit CPU, along with an Oscillator of 11.0592 MHz. It also has a 4k bytes of ON chip ROM & 128 bytes of RAM. 8051 also has 4 I/O ports which are of 8-bits namely P0, P1, P2 & P3. It also has 2 internal & 3 external interrupts. It has two timers which are of 16-bits. 8051 supports full duplex serial Communic

b) List the features of 8051 μC? — 6M.

→ 1) 8-bit CPU with registers A & B.
2) Clock circuits.
3) full duplex serial communication.
4) 4K bytes of ROM.
5) 128 bytes of RAM
6) 8-bit stack Pointer.
7) 2 internal & 3 external interrupts.
8) 21 special function registers
9) 4, 8-bit ports P_0, P_1, P_2 & P_3.
10) Two 16-bit registers DPTR & PC.
11) 8-bit flag register
12) 2-16 bit Timers T_0 & T_1.

c) Explain the PSW & flag bits 4M.

→

CY	AC	F0	RS1	RS0	OV	-	P.

4 flags. (math flags).

1) CY flag → If after addition CY is generated then CY=1 else CY=0.

2) AC → If CY is generated from D_3 to D_4 then Auxillary Carry AC=1 else AC=0.

3) OV → If CY is generated from D6 to D7 but after D7 there is no carry or CY is not generated from D6 to D7 but after D7 there is a carry in both cases OV=1 else it is zero.

4) Parity: If after addition total no of 1's in Acc are odd P=0 else P=1.

2a. Explain various addressing modes with example — 10M.

→ The various ways of accessing any data is called Addressing Mode.

1) Immediate A.M :→ This type of A.M is used when we need to transfer an immediate data.

Ex: MOV A, #30h.

2) Direct AM → This type of A.M is used when we need to transfer the direct address.

Ex: MOV A, 30h.

3) Register A.M :→ This type of A.M is used when data is to be transferred in between two registers.

Ex: MOV A, R0.

4) Register Indirect A.M: Here @Symbol is used before a register. Only R0 & R1 registers can be used.

Ex: MOV A, @R0.

5) Indexed A.M :→

MOVC A, @A+DPTR.
MOVC A, @A+PC.

6) Long Addressing mode: Ex: LJMP 30h.

7) Relative Addressing.

8) Absolute Addressing

9) Bit Inherent addressing:

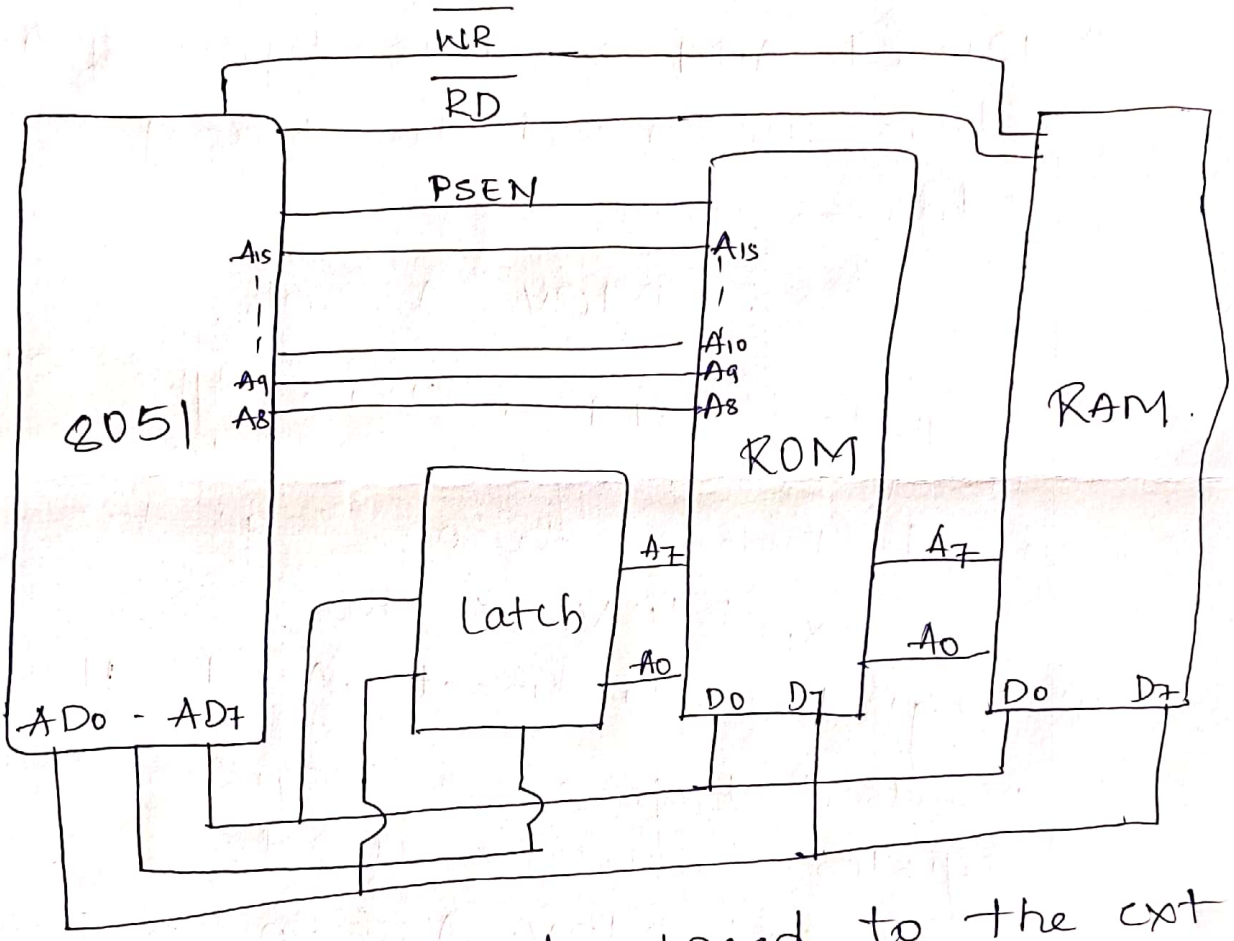
CLR A.

10) Bit Direct Addressing.

CLR Acc.7.

2b. Explain with neat diagram how to interface external code memory to 8051 µC — 10M.

→



8051 µC is interfaced to the ext memory (ROM & RAM) as shown in fig. The Latch IC is used to separate the address & data lines. PSEN is connected to ROM & $\overline{WR}$ & $\overline{RD}$ pins are connected to the ext. RAM.

Module 2.

3a. What are assembler directives? Explain various assembler directives — 8m.

→ Assembler directives are the pseudo codes (non-executable codes) which direct the assembler the operation need to b performed.

→ 1) ORG 2) EQU 3) db , 4) dw 5) END

1) ORG → Originate of the main pgm.

Ex: ORG 0000h.

2) EQU → It will equate the name to the particular address.

Ex: Suraj equ 30h.

→ Suraj = 30h.

3) db → define a byte.

ORG 0000h.

db 12h.

0000h 12h.

4) dw → define a word.

ORG 0000h.

dw 1234h.

0000h 12h.

0001h 34h.

5) END: It states the end of the program.

3b. Explain the following instructions of 8051 with examples. 12M.

1) DJNZ R1, res.

DJNZ R1, back.

if R1 = 01h. it decrements its to 0 & check if Not zero then it will jump to the address back else it will execute the next instruction.

2) DA A → Decimal Adjust accumulator after addition. It is used to obtain a valid BCD no

for ex. 38h
+45h
7Dh. → Invalid BCD
+ 6
83h. → valid BCD

3) MOVX A, @dptr.

Ex: MOVX A, @9000h.

BE. A = 05h.
9000h = 01h.

AE A = 01h.
9000h = 01h.

4) SWAP A

Ex: SWAP A

BE
if A = 65h.

AE,
A = 56h.

5) XCHD A, @R1

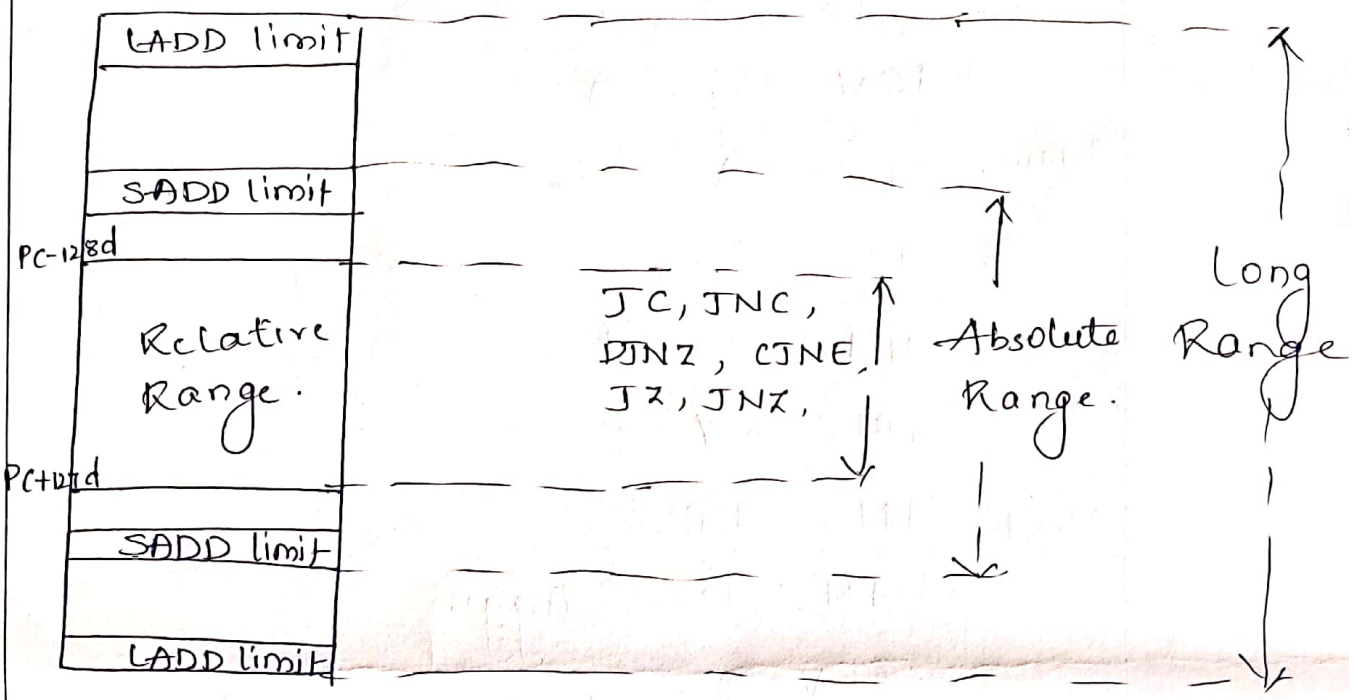
Ex: A = 05h.
R1 = 90h.
90h = 01h.

AE
Ex: A = 01h.
R1 = 90h.
90h = 05h.

6) INC R2.
if R2 = 05h. after execution R2 = 06h.

4a. With a neat diagram explain the range of Jump & CALL instruction - 8m

→



Relative Range → In this range there will be all conditional & unconditional jumps. It starts from PC+127d to PC-128d.

Absolute Range: In this type the jump is within a page. Each page is of 2K bytes. Therefore, total 32 pages. in which first 5 bits represent the page no & rest 11 bits represent the actual bit numbers.

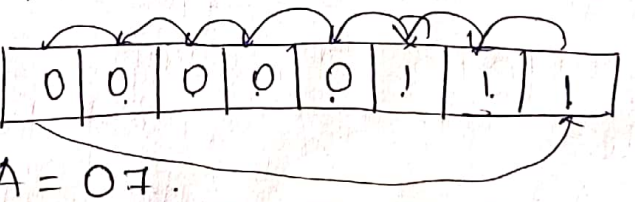
Long Range: In this type of range, the jump is anywhere from 0000h to FFFFh.

4b. Write an 8051 ALP to find average of five nos stored from internal Data memory address 40h. 8M.

→
```
MOV  R0, #40h.
MOV  R2, #05h.
CLR  A
MOV  R7, A.
Again: ADD  A, @R0.
       DA   A
       JNC  Next.
       INC  R7
Next.  INC  R0.
       DJNZ R2, Again
       mov  R0, #01h.
       DIV  AB.
       mov  R2, A.
```

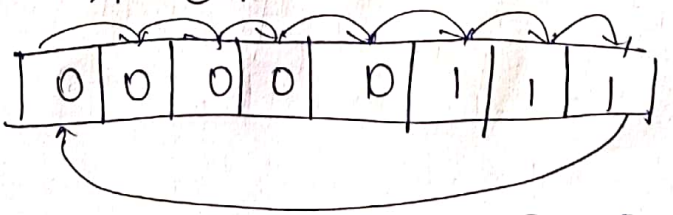
4c. Explain Rotate Instruction of 8051 with examples. — 4M.

→ 1) RL A → A = 07.



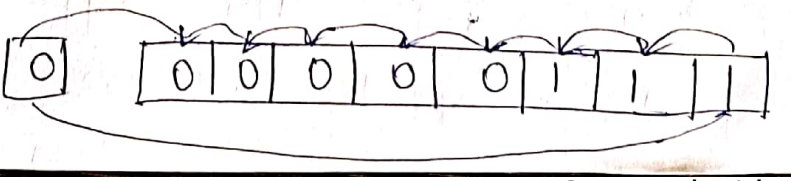
A = 0Eh.

2) RR A → A = 07.



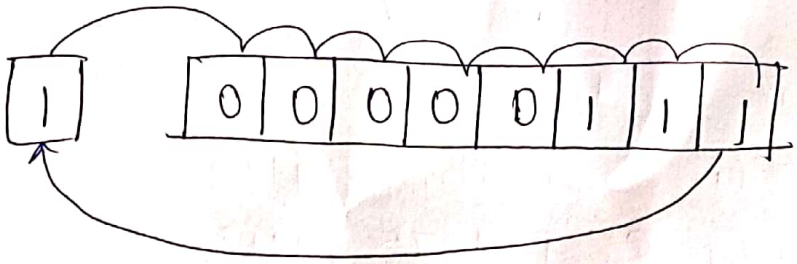
A = 83h.

3) RLC A A = 07h. C = 0



C = 0
A = 0Eh.

4) RRC A. A = 07h C = 1



CY = 1 ; A = 83h.

Module 3

5a Explain the various data types in 8051 C? — 8M.

→

Unsigned Char	8-bit	0-255
Signed Char	8-bit	-128 to +127
Unsigned int	16-bit	0 to 65,535
signed int	16-bit	-32768 to +32767
S-bit	1-bit	0 or 1
Bit	1-bit	0 or 1
Sfr	8-bit	FF

5b. Write an 8051 C-pgm to toggle the bits of P1 continuously with a 250msec delay ↳ 06M.

→

```
#include <reg51.h>
void Msdelay (unsigned int);
void main (void)
{ while(1)
{
 P1=0X55;
msdelay (250);
 P1=0XAA;
msdelay (250);
}}
```

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

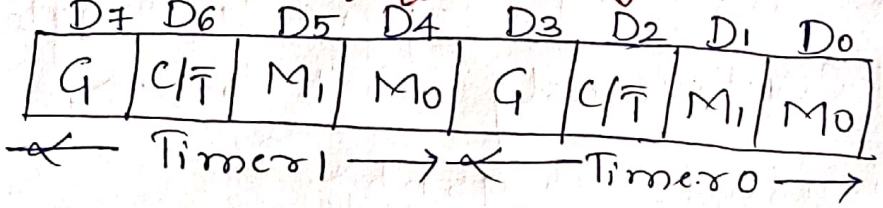
5c. Write an 8051 C pgm to toggle bit P2.4 Continuously without disturbing rest of bits of P2 — 6m.

→
```
#include <reg51.h>
sbit mybit = P2^4;
void main (void).
{
while(1).
{
mybit = 1;
mybit = 0;
}
}
```

6a. Explain TMOD register

Module 4.

→



D7→G → If G=1 → T1 will run provided INT1 pin should be 1 & TR1 = 1
If G=0 → T1 will run regardless of INT1 pin. but TR1=1.

D6→C/T → If C/T = 1 → Timer will act as counter counting no of events
C/T = 0 → Timer mode.

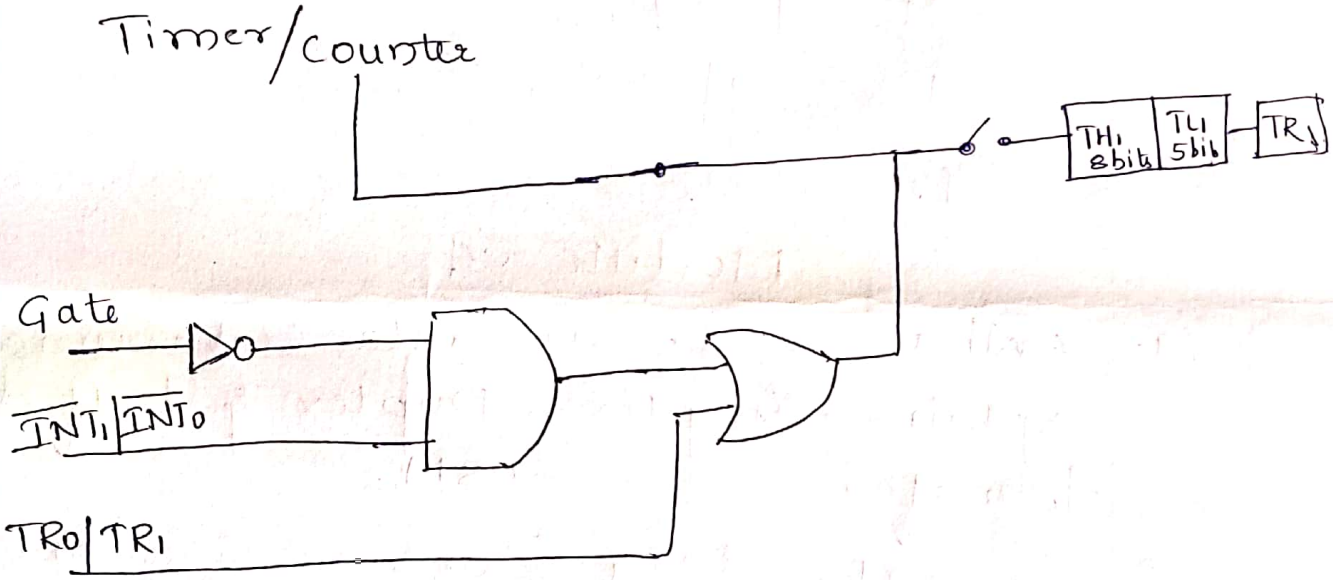
D5 & D4

M0	M1	Mode	Specification.
0	0	0	13-bit timer mode
0	1	1	16-bit timer mode
1	0	2	Split timer mode
1	1	3	Autoreload mode

The same next 4 bits hold for Timer 0.

6b. Explain Mode 1 programming of 8051 timer — 6M.

→



1) The TMOD register is set for the timer (0 or 1) & mode 1.
2) Load TL1 & TH1 values.
3) Start the timer.
4) JNB TF1, here keep doing until it changes from all 1s to all 0's.
5) Stop the timer.
6) Clear the timer overflow flag.
7) End of the program.

6c. Write an 8051 C pgm to convert packed BCD to ASCII & display the bytes on P1 & P2 — 6M.

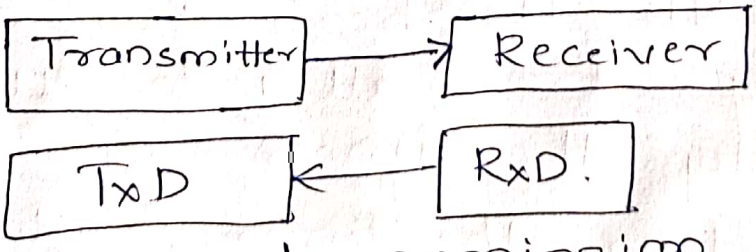
→
```
# include <reg51.h>
void main (void)
{
Unsigned char x, y, z;
Unsigned char mybyte = 0x29;
x = Mybyte & 0X0F;
P1 = x | 0X30;
y = Mybyte & 0XF0;
y = y>>4;
P2 = y | 0X30;
}.
```

Module 04

7a. What is Serial data communication? Explain Simplex, Duplex & half duplex transfer. — 08M.

→ Serial data Communication refers to transmission of 8-bit data serially. 8051 supports full duplex serial communication.

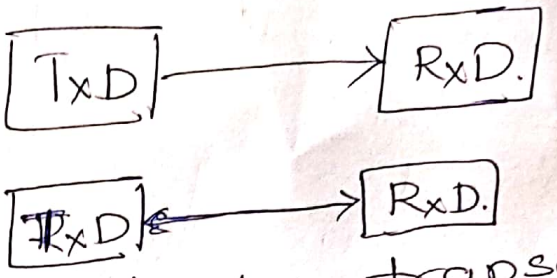
1) Simplex Transmission:-



In Simplex transmission, the data is either transmitted or received.

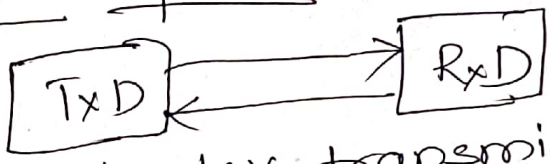
Ex: Pointer

2) Half Duplex :-



In half duplex transmission the data can be transmitted as well as received but not simultaneously.
Ex: Walkie-Talkie.

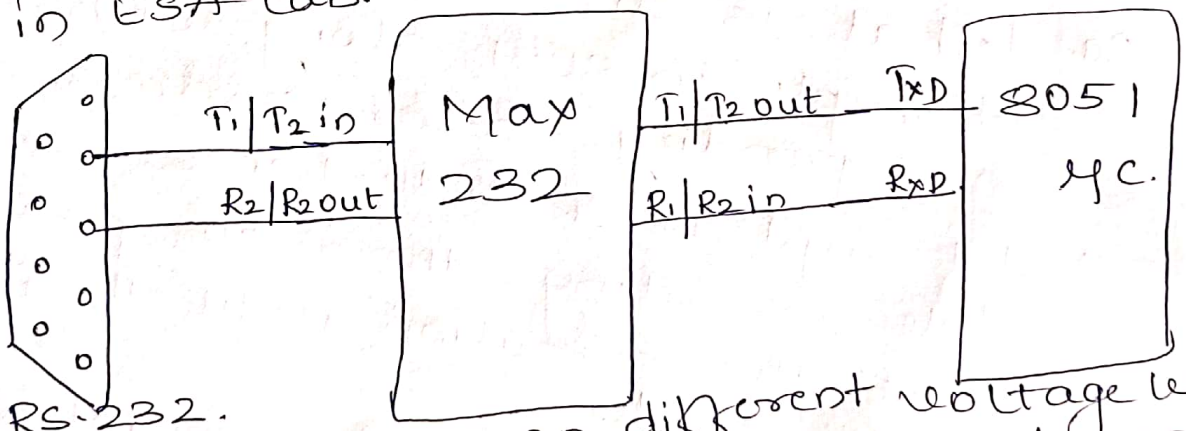
3) full Duplex :-



In full duplex transmission the data is both transmitted & received simultaneously.

7b. Draw & explain the interface of Rs-232 to 8051 using MAX-232? —06M.

→ RS-232 was first established in 1960 in ESA Lab.



RS-232 works on different voltage levels. -3V to -25V for logic 1. for logic 0 & +3V to +25V whereas 8051 works on +5V Therefore, to have an effective communication MAX-232 is used which converts to the desired voltage levels. It acts as an intermediator b/w RS-232 & 8051 µC.

7c. Write a 8051 C pgm to transfer the letter 'c' serially at 9600 baud rate continuously. Use 8-bit data & 1 stop bit — 6M.

→
```
#include <reg51.h>
void main()
{
 TMOD = 0X20;
  TH1 = 0XFD;
  SCON = 0X50;
  TR1 = 1;
while(1)
{
 SBUF = 'C';
while (TI==0);
  TI=0;
 }
}
```

8a. What is an Interrupt. List the various interrupts in 8051 along with their corresponding vector address. → 8M

→ Interrupt is a disturbance created while executing the main program & stops the execution until the interrupt is addressed.

There are namely 5 interrupts EX1, INT1, EX0, INT0 & serial port.

1	EX1 or external interrupt 1	000B.
2	INT1	001B
3	EX0 or ext. interrupt 0	0003.
4	INT0	0013
5	Serial port	0023.

8b. Explain the bit status of SCON Register 08

→ 6M.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0 & SM1 → Serial Mode specifier
SM2 → Multiprocessor Communication
REN → Enables Reception.
TB8 → not used in Mode1
RB8 → not used in mode1
TB8 → Transmit interrupt flag, set by H/w
RB8 → Receive interrupt flag, set by H/w.

8c. Write a C pgm that continuously gets a single byte data from P1.7 & send it to P1.0. While simulation creating a Square wave of 200 μs period on P1.4 & P2.5. Use timer 0 & create Square wave Assume XTAL = 11.0592 MHz — 6M

→

```
#include <reg51.h>
sbit SW = P1^7;
sbit IND = P1^0;
sbit wave = P2^5
void timer0 (void) Interrupt
{
wave = ~wave;
}
void main ()
{
SW = 1;
TMOD = 0x02;
TH0 = 0xA4;
IE = 0x82;
while (1)
{ IND = SW;
}
```

Module 5

9a. Explain pin diagram of 8255 chip — 7M.

→ 8255 chip also known as PPI is used to interface peripheral devices with micro processors. Its pin diagram consist of 40 pins divided in 4 groups.

1) Data Bus buffer :- Pin DB0 – DB7 serve as bidirectional data lines. These line facilitate communication b/w µp & external devices.

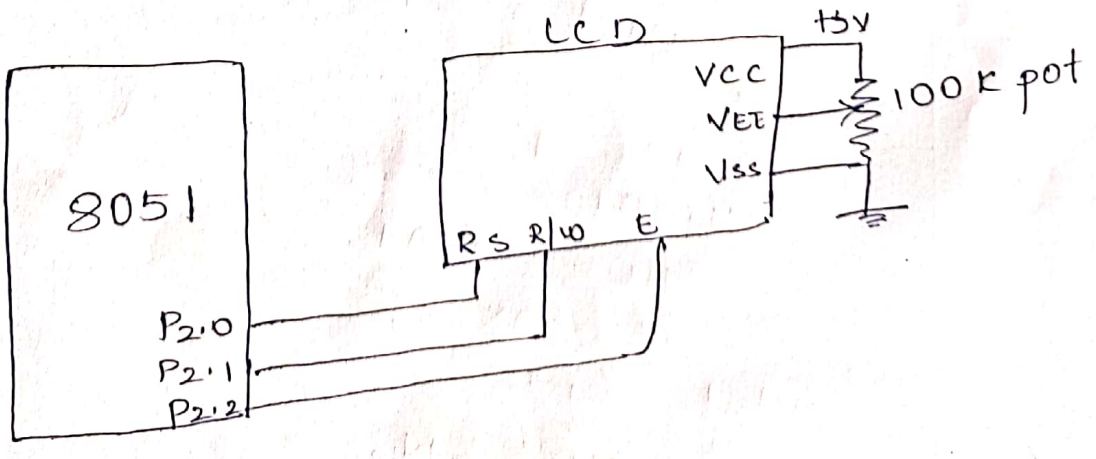
2) Control & status signals : Pins such as. CS, WR, & Reset, RD & A0 - A1 control the Operation of 8255 chip & indicate its status.

3) Peripheral Interface signals : PA0 - PA7, PB0 – PB7, PC0 – PC7 & various control pins are used to interface with external peripheral devices.

4) Ground & power Supply : Pins Vcc & Gnd provide the necessary supply & Ground.

9b. Draw & explain the interface diagram of LCD with 8051 µC? 8M.

→



Port 2 pins of 8051 are used. RS is used as a register select pin to use either as command register or data register, R/w represents read & write pin & E represents enable pin. Vcc is +5V, VEE is used to adjust the Control contrast of LCD & Vss is ground. The driver IC is inbuit present in the LCD module.

9c. Write a C-program to rotate Stepper motor continuously in clockwise direction 6M.

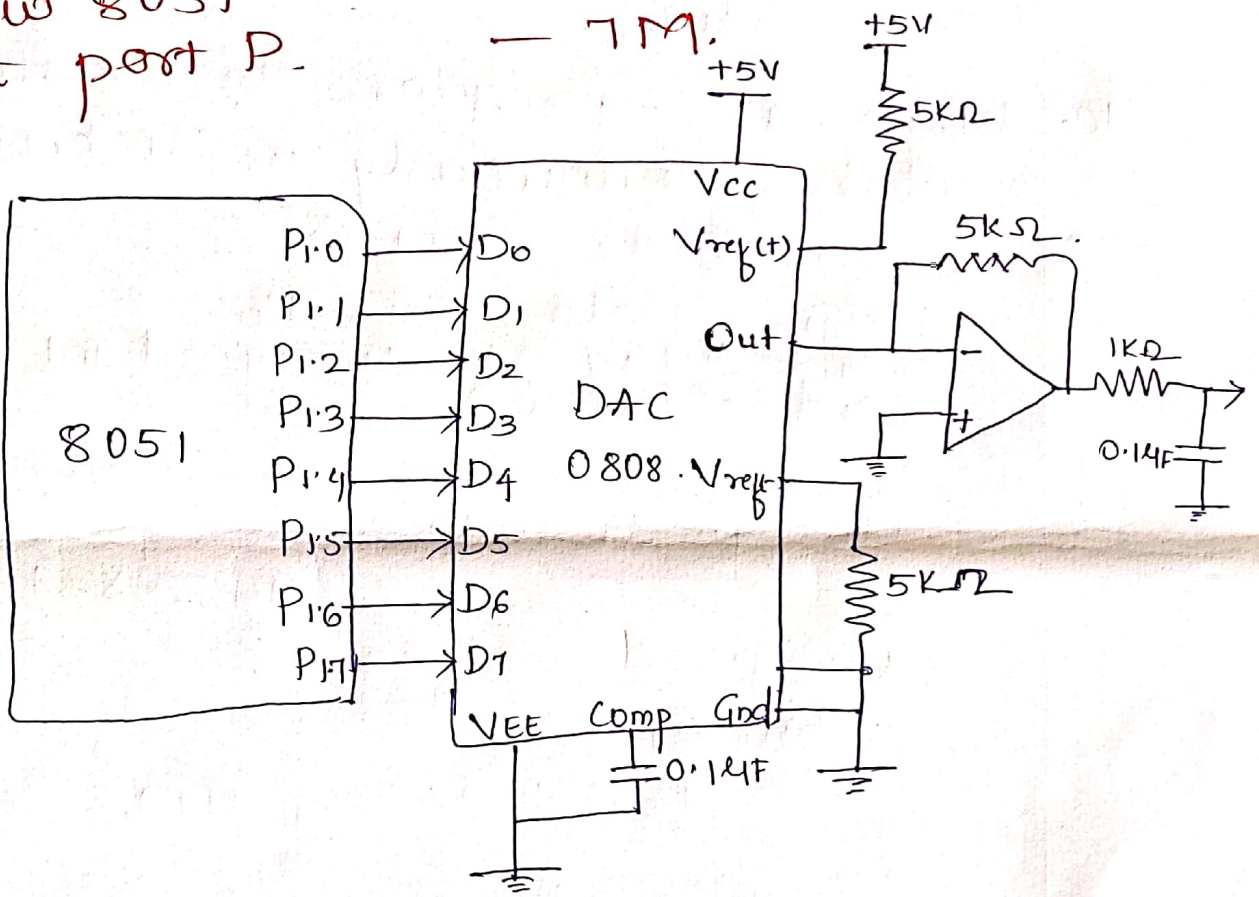
→

```
#include <reg51.h>
Define Data_Bus P0
void delay MS (unsigned Int)
void main ()
{
Unsigned char full step close = {0X18;
                                 0X14, 0X24; 0X28};
Unsigned char i;
while (1)
for (i=0 < i≤3 ; i++)
{
Data bus = full step clock (1)
delay 4sec (50);
}
void delay Ms sec (unsigned int count)
Unsigned Int r;
TMOD = 0X01;
TH0 = 0X80;
TH0 = 0XF8;
TH0 = 0XCD;
TR0 = 1;
for (r=0; r< count; r++).
```

```
{
while (;TF0)
TF0 = 0 ;
TH0 = 0XF8;
TL0 = 0XCD;
}
TR0 = 0.
```

10a) Draw a block diagram to show how 8051 is connected to DAC0808 at port P. — 7M.

→



10b. Write a C pgm to generate a sine wave using DAC — 6M.

→

```
#include <reg51.h>
Define DA-databus P1
void main (void)
{
Unsigned char i,j;
Unsigned char sample_point[] = {0X80, 0XC0, 0XEE
0XFF, 0XEE, 0XC0, 0X80; 0X40,
0X11, 0X00, 0X40, 0X80}
```

```
while (1)
{
for (i=0; i<=11; i++)
{
  dac_databus = sample_point[i];
    for (j=0; j<=10; j++);
  }.
}.
```

10c. Explain the internal Architecture of ADC 0804 — OTM.

→ ADC 0804 is an 8-bit analog to digital (ADC) converter commonly used in µC based systems. Its internal architecture typically consist of several key components:

1) Analog inputs: The ADC 0804 usually has eight analog i/p channels, labelled A0 through A7. These inputs allow the Converter to sample & convert analog signals into digital values.

2) Sample & Hold circuits: Before conversion, the ADC 0804 employs a sample & hold circuit to capture the input vltg & hold it steady during the conversion process. This prevents variations in the i/p vltg from affecting the accuracy of the conversion.

3) Comparator N/w :- This compares the i/p vltg with ref vltg & determines whether i/p vltg is higher or lower than ref vltg.

4) Successive Approximation Register (SAR) :- It performs binary search algorithm to determine the digital output value.

5) Control Logic :- Controls the sample & hold circuit, manages the SAR Operation & handles data transfer.

6) Output Registers :- These registers holds the converted digital value until it is read by an external device such as Microcontroller.

Motekar.

(Ravindra Motekar)