CBCS SCHEME

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

BEE403

Fourth Semester B.E./B.Tech. Degree Examination, June/July 2025

Microcontrollers

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks , L: Bloom's level , C: Course outcomes.

		Module – 1	M	L	C
Q.1	a.	Differentiate between a microprocessor and a microcontroller.	06	L4	CO1
	b.	With the neat block diagram, explain the architecture of 8051 microcontroller.	08	L2	CO1
	c.	Explain PSW register of 8051 microcontroller.	06	L2	CO1
		OR			
Q.2	a.	Write a short note on stack memory.	05	L2	CO1
	b.	Explain the internal memory organization of 8051 microcontroller with a neat memory map.	08	L2	CO1
	c.	What are addressing modes? Mention different addressing modes used in 8051 microcontroller and explain any two in detail with example.	07	L1 L2	CO2
		Module – 2			
Q.3	a.	Explain the following instructions with an example: (i) XCHD A, @R_p (ii) MOVC A, @A+DPTR (iii) RRC A (iv) AJMP absolute address (v) DA A	10	L2	CO2
	b.	Write an assembly program to add two 16 bit numbers stored in memory 20H, 21H, 22H, 23H and store result in 30H, 31H, carry on 32H.	05	L2	CO2
	c.	Explain PUSH and POP instructions with example.	05	L2	CO2
		OR			
Q.4	a.	Explain any four assembler directives used in 8051 microcontroller with example.	06	L2	CO2
	b.	Write an assembly program to transfer 5 bytes from one memory to another memory within internal RAM.	06	L3	CO2
	c.	Explain any four arithmetic instructions and logical instructions.	08	L2	CO2
		Module – 3			
Q.5	a.	Explain various C-data types used in 8051 microcontroller.	06	L2	CO3
	b.	Explain TMOD and TCON with its bit pattern.	08	L2	CO3
	c.	Write 8051 C program to send – 4 to +4 to port 0.	06	L3	CO3

		OR			
Q.6	**a.**	Write 8051 assembly program to toggle all bits of P_2 continuously 500 ms. Use timer1, 16-bit-mode to generate delay f = 11.0592 MHz.	**06**	L3	CO3
	b.	Explain mode 2 programming of 8051 time. Describe the different steps to program in mode 2.	**06**	L2	CO3
	c.	Write 8051 program to generate square wave with t_{ON} = 3 ms and t_{OFF}=7 ms on all pins of port 0. System clock is 22 MHz. Use timer 0 in mode 1.	**08**	L3	CO3
		Module – 4			
Q.7	**a.**	Define the following terms with respect to communication: (i) Serial communication (ii) Parallel communication (iii) Simplex communication (iv) Half duplex communication (v) Full duplex communication	**05**	L2	CO4
	b.	Explain all handshaking signals of RS-232 communication standard.	**07**	L2	CO4
	c.	Write the steps required by 8051 microcontroller to receive and send data serially.	**08**	L2	CO4
		OR			
Q.8	**a.**	Mention different interrupts used in 8051 microcontroller with their interrupt vector table.	**05**	L2	CO4
	b.	Write an ALP that continuously gets 8-bit data from P_0 and sends it to P_1. While simultaneously creating a square wave of 200 μs on $P_{2.1}$. Use timer 0. XTAL = 11.0592 MHz.	**07**	L3	CO4
	c.	Explain the bit contents of SCON and PCON registers.	**08**	L2	CO4
		Module – 5			
Q.9	**a.**	Interface a stepper motor to 8051 and rotate it by checking the status of a simple toggle switch connected to $P_{2.7}$ as follows : (i) If switch is open rotate the motor clockwise (ii) If switch is closed, rotate the motor counter clockwise directions.	**08**	L3	CO5
	b.	Explain the salient features of ADC 0804. What are the signals importance while interfacing such as ADC to a 8051 controller.	**08**	L3	CO5
	c.	Write 8051 program to generate a ramp signal.	**04**	L3	CO5
		OR			
Q.10	**a.**	Explain the functional block diagram of 8255.	**08**	L2	CO5
	b.	Write an ALP to display the message "DONE".	**08**	L3	CO5
	c.	Draw the block diagram to show how 8051 is connected to DAC 0808 at port P_1 using output buffer for DAC.	**04**	L1	CO5

* * * * *

June / July - 2025
Microcontroller (BEE403)

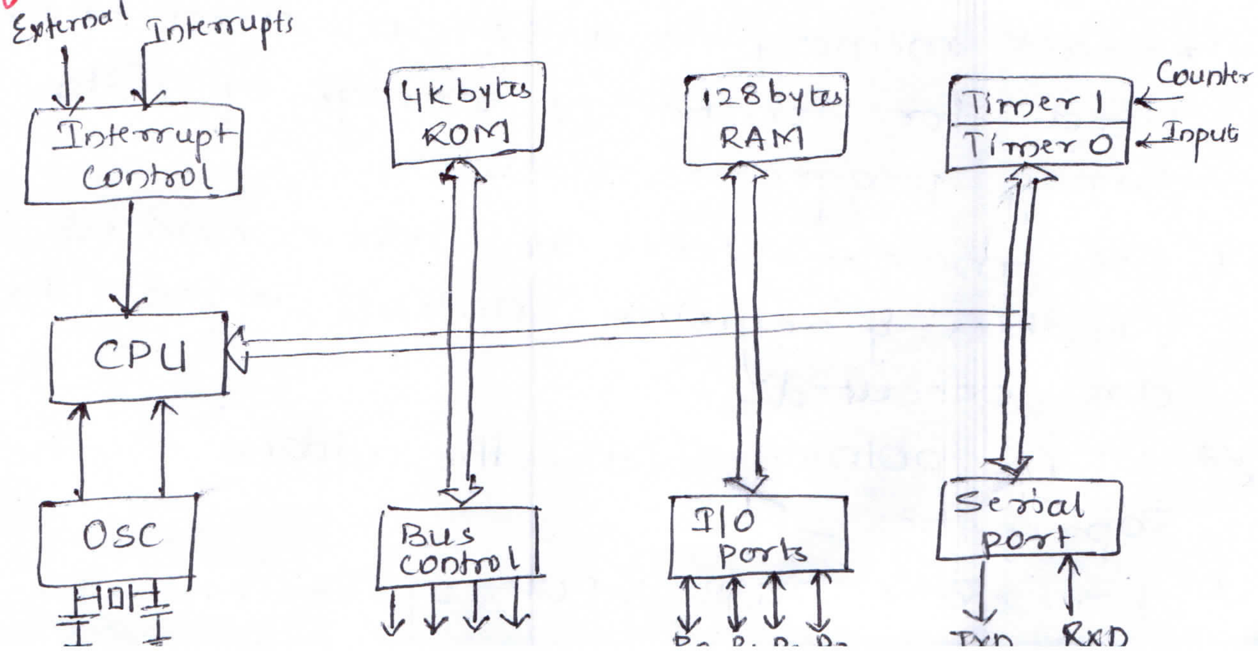
Subject Name : Microcontroller
Subject Code : BEE403
Semester : 4th.
Department : Electrical & Electronics Engg.
Prepared By : Prof. Ravindra Motekar

1a) Differentiate between Microprocessor & Microcontroller. (6M)

→ Microprocessor	Microcontroller
1) Used for General purpose	1) Used for specific purpose
2) Provides flexibility	2) Don't provide.
3) Many instructions	3) few instructions
4) Requires more hardware	4) less required
5) It has one or two bit handling instructions.	5) Many bit-handling instructions.
6) ALU, Registers, Clock circuits are fabricated on a single chip	6) Memory, I/p-o/p device & ALU processor all are in a single chip.

1b) With neat diagram explain architecture of 8051 microcontroller. (8M)

→



8051 μC consists of an 8-bit CPU along with an oscillator of 11.0592 MHz. It also has 4K bytes of ROM & 128 bytes of RAM. 8051 also has 4 i/o ports which are of 8-bits namely P_0, P_1, P_2 & P_3. It also has 2 internal & 3 external interrupts. It has 2 timers T_0 & T_1 which are of 16 bits & 8051 supports full duplex serial communication.

1c) Explain PSW register of 8051 microcontroller (8M)

→

CY	AC	F0	RS1	RS0	OV	–	P

4 math flags.

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

2> AC → If CY is generated from D3 to D4 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 carry, in both cases OV=1, else its zero.

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

OR

2a) Write a short note on stack memory. (5M)

→* Stack memory is a special area of RAM used for temporary storage of data during program execution.

* It mainly stores the return address, local variables & register contents when interrupts are executed.

* Stack pointer holds the address at the Top of the stack.

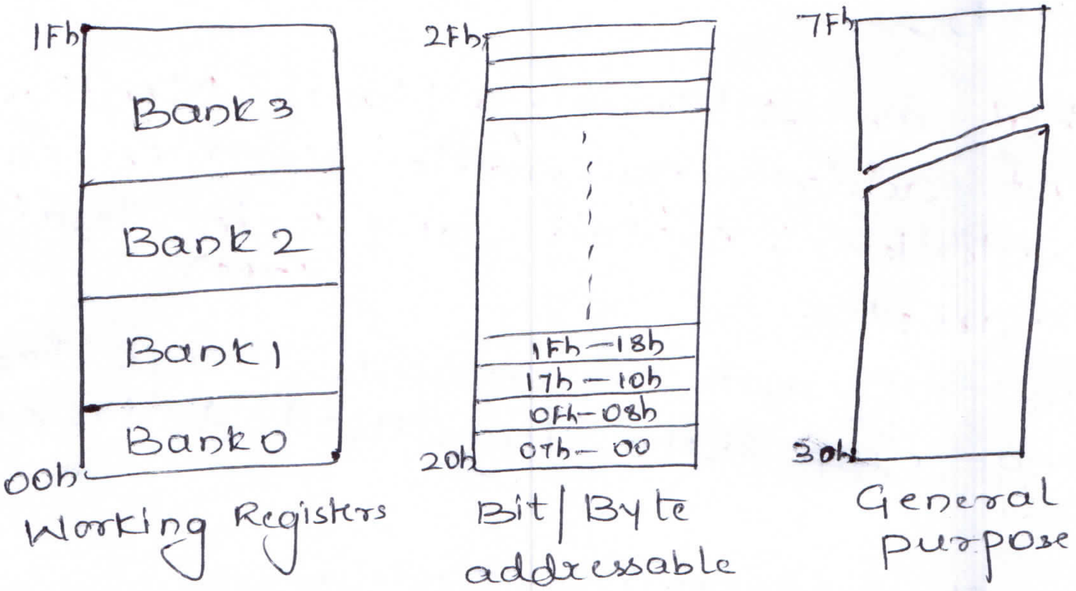
* For every PUSH & POP operation, the

stack pointer is incremented/decremented 2 by one.

* Proper stack management is essential, Otherwise stack Overflow or data corruption may occur.

* After every Reset stack pointer points at address 07h.

2b) Explain the internal memory Organization of 8051 microcontroller with a neat memory map. (8M)

→ 

The internal memory organization of RAM is classified into 3 types.

1) Working Registers
2) Bit | Byte Addressable
3) General Purpose

1) Working Registers :-
Working Registers consist of 4 Banks

a) Bank 0
b) Bank 1
c) Bank 2
d) Bank 3

All the Banks consists of registers R0 - R7. After reset by default Bank 0 is selected.

2) Bit | Byte Addressable :-

This area can be used as both bit | byte addressable. Mostly single bit instructions & logical instructions are used. The address area is from 20h to 2Fh.

3) General Purpose :-

This area is used for any data transfer instructions. Address area is from 30h - 7Fh.

2c) What are Addressing Modes? Mention different addressing modes used in 8051 microcontroller & explain any 2 in detail with example. (7M)

→ The different or various ways for accessing any data is called Addressing modes.

Types of Addressing modes.

1) Immediate addressing mode.
2) Register AM
3) Register Indirect AM
4) Direct AM.
5) Indexed AM.
6) Relative AM
7) Absolute AM
8) Long AM
9) Bit inherent AM
10) Bit Direct AM

1) Immediate A.M :→

In immediate addressing mode a direct data is transferred to the destination address. the data is preceeded with a pound sign (# sign).

Ex: MOV A, #30h.

Before execution:

A = 05h

After Execution

A = 30h.

2) Register A.M :→

In Register addressing mode, the data in the register is transferred to the destination register. Here both source & destination are registers only.

MOV Rd, Rs → syntax.

MOV R2, R1 → invalid instruction

Ex: MOV A, R5

Before Execution

A = 30h

R5 = 01h

After Execution

A = 05h

R5 = 05h.

Module - 2

3a) Explain the following instructions with an example. (10M)

(i) XCHD A, @Rp

Ex: XCHD A, @R0

Before Execution

A = 05h

R0 = 30h.

30h = 09h

After execution

A = 09h

R0 = 30h

30h = 05h

ii) MOVC A, @A+DPTR

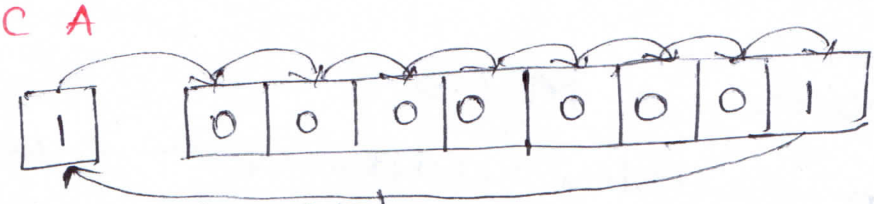
It is mainly used for lookup tables stored in ROM

Example! If DPTR = 8500h
& A = 03h , 8503h = 99h.

then, MOVC A, @A+DPTR

then A = 99h.

iii) RRC A



let say A = 05h.
C = 1

After execution → A = 80h
C = 1

iv) AJMP absolute address

AJMP stands for absolute jump. It is a 2 byte instruction used to jump to a specified address within the same 2kb block of program memory.

Ex! AJMP loop.

if instruction address is 0800h, then jump can occur anywhere b/w 0800h to 0FFFh.

v) DA A !-

Decimal Adjust accumulator after addition. This instruction adds 6 to a number if it is greater than 9 to convert it from invalid BCD to valid BCD no.

Ex! 47h
38h
85h

But 47h
+38h
7Fh
+6 = 85h.

3b) Write an assembly program to add two 16-bit numbers stored in memory 20h, ~~21~~ 21h, 22h, 23h & store result in 30h, 31h & Carry on 32h. (5M)

```
        ORG 0000h.
→       MOV  A, 20h.
        ADD  A, 22h
        MOV  30h, A

        MOV  A, 21h
        ADDC A, 23h
        MOV  31h, A.
        MOV  A, #00h
        ADDC A, #00h
        MOV  32h, A
        NOP
        END
```

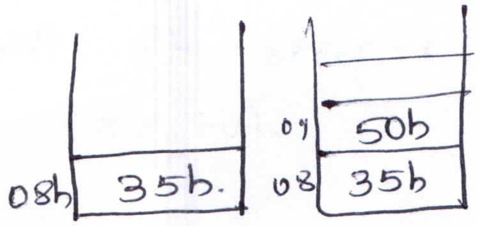
3c) Explain PUSH & POP instruction with an example. (5M)

→ PUSH → Push instruction increments the stack pointer by '1'.

SP → SP + 1

for every push stack pointer gets incremented.

```
Ex: MOV R1, #35h.
    MOV R2, #50h
    PUSH 01h
    PUSH 02h
```

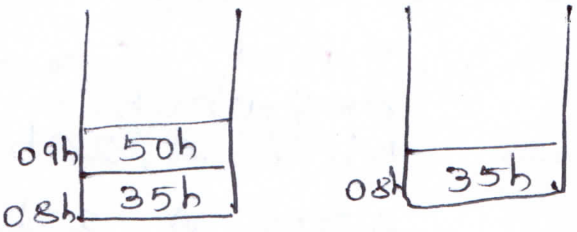


→ POP → POP instruction decrements the stack pointer by '1'

SP → SP − 1

for every POP instruction stack pointer gets decremented.

```
MOV R1, #35h
MOV R2, #50h.
PUSH 01h
PUSH 02h
POP 02h
POP 01b
```



OR

4a) explain any 4 assembler directive used in 8051 microcontroller with example (6M)

→ Assembler directives are the pseudocode which don't perform any operation but directs the assembler what has to be done.

1) ORG
2) Equate
3) Define a byte
4) Define a word
5) END.

1) ORG → It stands for the originate of the program. ORG 8000h represents that program starts from address 8000h.

2) Equate : This directive equates the name to the address.

for ex: e. Suraj equ 30h
→ Suraj = 30h

3) Define a byte :

```
ORG 0000h
db 12h.
```

It defines 0000h = 12h.

4) END :→

It states the end of the program

4b) Write an assembly program to transfer 5 bytes from one memory location to another within internal RAM. (6M)

→
```
         ORG  0000h
         MOV  R0, #20h
         MOV  R1, #30h
         MOV  R2, #05h
Back:    MOV  A, @R0
         MOV  @R1, A
         INC  R0
         INC  R1
         DJNZ R2, Back.
         NOP
         END.
```

4c) Explain any 4 Arithmetic instruction & logical instructions. (8M)

→ 1) ADD A, R1

BE → A = 05h
R1 = 01h

AE → A = 06h
R1 = 01h

2) SUBB A, R0.

BE : A = 08h
R0 = 05h

AE :→ A = 03h
R0 = 05h

3) MUL AB.

BE A = 02h
B = 02h

AE → A = 04h
B = 00h.

4) DIV AB.

BE :- A = 07h AE → A = 03h
B = 02h. B = 01h.

5) ORL c, 00h

BE :- c = 1 AE → c = 1
00h = 1

6) ANL c, 00h

BE → c = 0 AE :- c = 0
00h = 1

7) XRL c, 00h.

BE → c = 0 AE c = 1
00h = 1

8) CPL A

BE → A = 1 AE → A = 0.

Module 3

5a) Explain various c data types used in 8051 microcontroller. (6M)

→ unsigned char	8bit	0-255h
signed char	8.bit	-128 to +127
unsigned int	16bit	0 to 65,535
signed int	16 bit	-32,768 to +32,767
s-bit	1-bit	0 or 1
Bit	1-bit	0 or 1
sfr	8-bit	FFh.

5b) Explain TMOD & TCON Register with its bit pattern. (8M)

→ TMOD :→

D7	D6	D5	D4	D3	D2	D1	D0
Gate	C/$\overline{T}$	M_1	M_0	Gate	C/$\overline{T}$	M_1	M_0
← Timer 1 →				← Timer 0 →			

D7 → Gate :

If G = 1, T_1 will run provided INT_1 pin should be 1 & $TR_1 = 1$

If G = 0, T_1 will run regardless of INT_1 pin but $TR_1 = 1$.

D6 → C/$\overline{T}$:-

If C/$\overline{T}$ = 1 →

Timer will act as counter, counting the no of events.

If C/$\overline{T}$ = 0 → Timer mode

D5 & D4

M_1	M_0	Mode	Specifications.
0	0	0	13-bit timer mode
0	1	1	16-bit timer mode
1	0	2	split timer mode
1	1	3	Auto Reload mode.

TCON :→

TF_1	TR_1	TF_0	TR_0	IE_1	IT_1	IE_0	IT_0

TF_1/TF_0 Timer overflow flag : it becomes 1 when timer 1 rolls over from all 1's to all 0's.

TR1/TR0 → Starts the timer if bit is 1
if 0 Stops the timer.

IE1/IE0 → Interrupt enable.
Enables the interrupt if 1

IT1/IT0 → Interrupt transmit flag.
used for serial port interface.

5c) Write 8051 C program to send -4 to +4 to port 0. (6M)

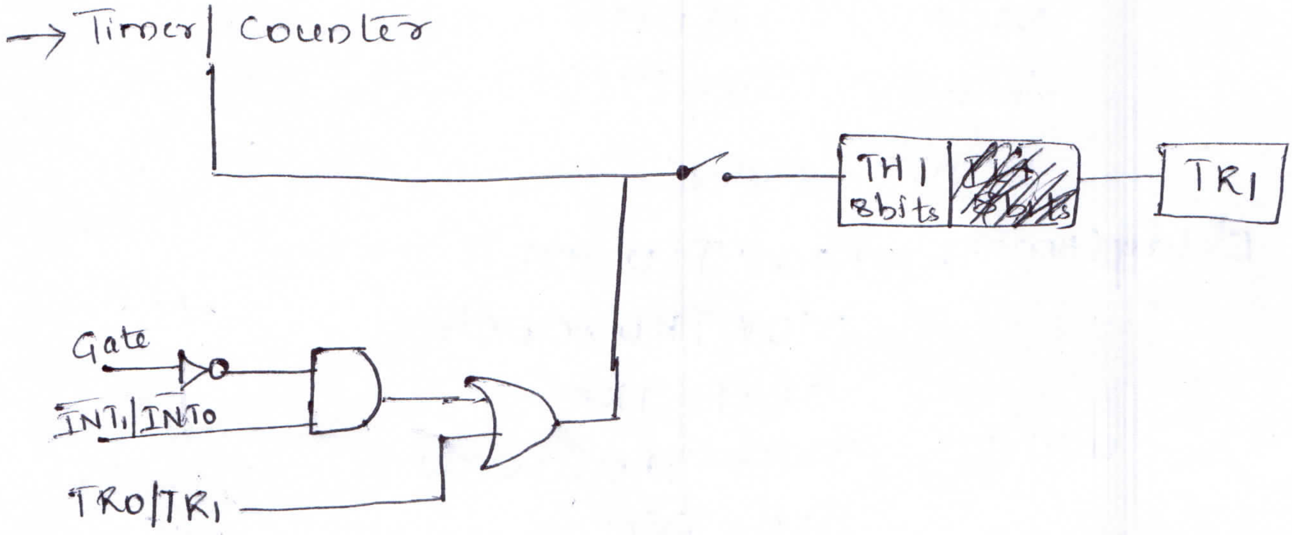
→
```
# include <reg51.h>
void main()
{
 Char i;
 while(1);
{
for (i=-4; i<=4; i++)
 {
   Po = i;
  }
 }
}
```

OR

6a) Write 8051 Assembly program to toggle all bits of P2 continuously for 500ms, Use timer 1, 16-bit mode to generate delay f=11.0592 MHz. (6M)

→
```
        ORG 0000h
        MOV TMOD, #10h
        MOV P2, #00h
        MOV R2, #10
Main:   CPL P2
        ACALL Delay_500ms
        SJMP  Main
  Delay. 500ms
   D1!        MOV  R2,#10$
              MOV  TH1,#4Cb
              MOV  TL1,#00b
              SETB  TR1
   Wait!      JNB  TF1, wait
               CLR  TR1
               CLR  TF1
              DJNZ  R2, D1
               RET
               END.
```

6b) Explain Mode-2 programming of 8051 timer. Describe the different steps to program in Mode 2 (6M)



steps

1) The TMOD register is set for the timer (0 or 1)

2) load TH1 values.

3) Start the timer.

4) JNB TF1, here. keep doing until it jumps

from all 1's to all 0's.

5) ~~Clea~~ Stop the timer

6) Clear the timer overflow flag.

7) End of the program.

6c) Write an 8051 program to generate a square wave with Ton = 3ms & toff = 7ms on all pins of port 0. System clock is 22MHZ. Use timer 0 in mode 1 (8M)

→ for TON = 3ms. & for Toff = 7ms.

```
THO = OEAh          THO = OCDh
TLO = 7Fh           TLO = OE4h.

        ORG 0000h
        MOV TMOD, # 01h
Back:   MOV Po, #OFFh
        Acall delay(3ms)
        MOV PO, #00h
        Acall delay(7ms)
        SJMP Back.
Delay(3ms):   MOV TLO, #7Fh
              MOV THO, #OEAh
              SETB TRO
      wait:   JNB TFO, wait.
              CLR TRO
              CLR TFO
              RET
delay (7ms):  MOV THO, #OCDh
              MOV TLO, #OE4h.
              SETB TRO.
      here    JNB TFO, here
              CLR TRO   CLR TFO   RET
```

Module - 4

7a) Define the following terms wrt to Communication. (5M)

1) Serial Communication:-

It is a method in which bits are transmitted one after other over a single communication line.

2) Parallel Communication:-

It is a method in which multiple bits are transmitted simultaneously using multiple data lines.

3) Simplex Communication:-

It allows data transmission in only one direction.

Ex: Printer.

4) Half Duplex Communication.

It allows data transmission in both directions, but not at the same time.

Ex: Walkie-Talkie

5) Full Duplex Communication:-

It allows data transmission in both directions simultaneously at once.

Ex: Telephone.

7b) Explain all hand shaking signals of RS-232 Communication standard. (7M)

→ In RS-232 serial Communication, handshaking signals are used to Control & co-ordinate data flow b/w the Data Terminal Equipment & Data Communication equipment.

These signals ensure that data is transmitted only when both sides are ready, preventing data loss.

* The main handshaking signals are.
RTS → Request to send
CTS → Clear to send.

Another pair is DTR → Data Terminal Ready
& DSR → Data set Ready.
DCD → Data Carrier Detect
RI → Ring Indicator.

7c) Write the steps required by 8051 μc to receive & send data serially. (8M)

→ steps to Transmit Data

1) Configure serial port mode & enable SCON Register.
2) set the Baud Rate using timer 1 in mode 2.
3) Load the data to be transmitted into the SBUF Register.
4) 8051 transmits the data serially to TxD pin.
5) Monitor TI flag, when it becomes 1, then transmission is complete.
6) Clear the TI flag to prepare for the next instruction.

→ steps to Receive Data (serially)

1) Configure serial port mode & enable SCON register

2) set the required baud rate using timer 1

3) 8051 receives serial data through the RxD pin & stores in SBUF register.

4) When data is received, RI = 1

5) Read the received data from the SBUF.

6) Clear the RI flag for next reception.

OR

8a) Mention the different interrupts used in 8051 Microcontroller with their interrupt vector table. (5M)

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

There are mainly 5 interrupts

EX1 → External interrupt 1

INT1 → Interrupt 1

EX0 → External interrupt 0

INT0 → Interrupt 0

Serial port Interrupt

Sl.No	Name of Interrupt	Vector Address
1	EX1	000B
2	INT1	001B
3	EX0	0003
4	INT0	0013
5	Serial port	0023

8b) Write an ALP that continuously gets an 8-bit data from Po & sends it to P1, while simultaneously creating a square wave of 200ms on P2.1. Use timer 0, XTAL = 11.0592 MHz (7M)

→
```
              ORG  0000h
              MOV TMOD,#01h.
              CLR P2.1
  Main:       MOV A, Po
              MOV P1, A
              CPL P2.1
              ACALL Delay-200ms.
              SJMP Main.
  Delay-200ms
               MOV TH0,#0FFh.
               MOV TL0,#0A4h
               SETB TR0
       here: JNB TF0, here
                CLR TR0
                CLR TF0
                 RET
```

8c) Explain the bit contents of SCON & PCON Register. (8M)

→ SCON :→

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0 & SM1 → serial mode specifier

SM2 → Multiprocessor communication.

REN → Enables Reception.

TB8 → not used in mode 1

RB8 → not used in mode 1

TI → Transmit Interrupt flag set by hardwar

RI → Receive Interrupt flag set by hardware

→ PCON Register

SMOD	—	—	—	GP1	GP0	PD	IDL

SMOD → if SMOD = 1 → it doubles the serial baud rate when it is set to 1.

GP1 → General purpose bit 1

GP0 → General purpose bit 0

PD → Power Down mode control

ID → Idle mode control.

Module - 5

9a) Interface a stepper motor to 8051 & rotate it by checking the status of a simple toggle switch connected to P2.7 as follows.
(i) if switch is open rotate motor. Clockwise
(ii) if switch is closed, rotate the motor in counter clockwise directions. (8M)

→

```
        ORG 0000h
        MOV P1, #00h
        SETB P2.7
Main:   JB P2.7, CW
        SJMP CW
```

```
CW:     MOV P1,#09h
        ACALL Delay
        MOV P1 , #0Ch
        Acall Delay
        MOV P1 , #06h
        Acall Delay
        MOV P1 ,#03h
        Acall Delay
        SJMP Main

CCW :   MOV P1,#03h
        Acall Delay
        MOV P1 , #06h
        Acall Delay
        MOV P1 , #0Ch
        Acall Delay
        MOV P1 , #09h
        Acall Delay
        SJMP Main

Delay:  MOV R7 ,#0FFh.
here1:  MOV R6 , #0FFh
here :  DJNZ R6 , here
        DJNZ R7, here 1
        RET
        END.
```

9b) Explain the salient features of ADC 0804. What are the signals importance whi interfacing such as ADC to a 8051 Controller (8M)

→ Salient features →

1) ADC 0804 is an 8-bit Successive approxi type ADC converter.

2) Converts analog i/p Voltage into 8-bit digital o/p.

3) Conversion time is typically 100μs.

4) It operates at 5V supply & has an internal clock.

5) The ADC 0804 has a reference voltage (Vref/2) pin, which allows section of i/p voltage range usually 0-5V.

→ Importance :-

* The important Control & status signals of ADC 0804 include CS (Chip select), RD (Read) & WR (Write).

* The WR signal is used to start the conversion while RD is used to read the converted digital data.

* CS signals enables the ADC during read & write operation.

* INTR pin goes low at the end of conversior indicating that data is ready to be read.

* Vin(+) & Vin(-) accepts analog i/p signal

* Vref/2 sets max. i/p voltage & hence resolution

* CLK IN & CLKR used to generate ADC clock signals

9c) Write an 8051 program to generate a RAMP signal. (4M)

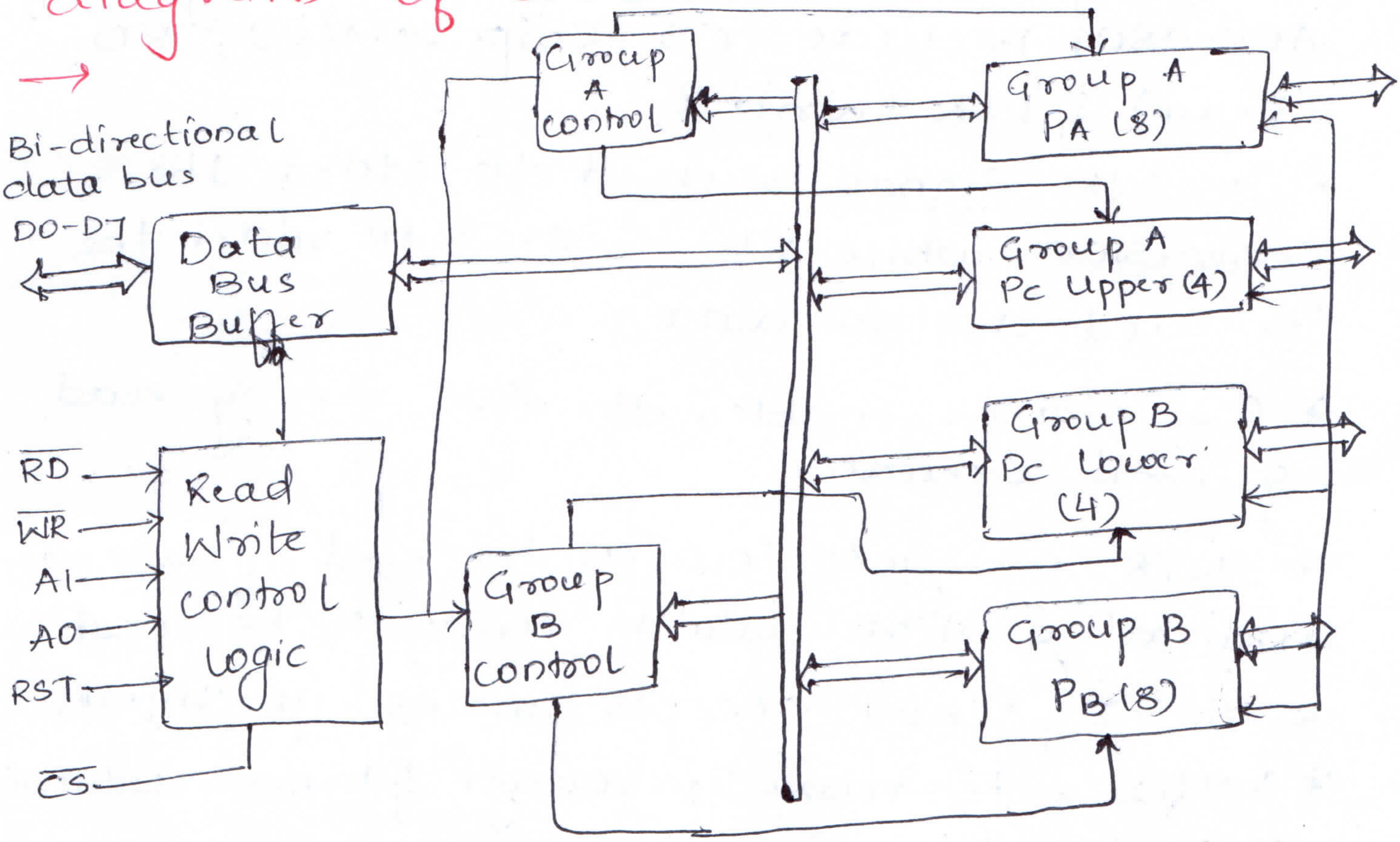
→

```
         ORG 0000h
         MOV A, #00h
Back:    MOV P1, A
         Acall Delay
         INC A
         SJMP Back
Delay:   MOV R7, #0FFh
D1:      MOV R6, #0FFh
D2:      ~~MON~~ DJNZ R6, D2
         DJNZ R7, D1
         RET
         END
```

OR

10 a) Explain the functional block diagram of 8255. (8M)

→



Internal Organization of 8255 is shown in fig

* It has 2 8-bit ports PA & PB.
* Two 4 bit ports Pc (upper) & Pc (lower)
* Ports can be programmed both as i/p & o/p
* When ports are defined as o/p they are called as Latch & when i/p they act as buffer.
* There are 2 Group A & Group B control

Group A Control → Controls port A & upper port C

Group B control → Controls port B & lower port C

* Read/write control has six lines.

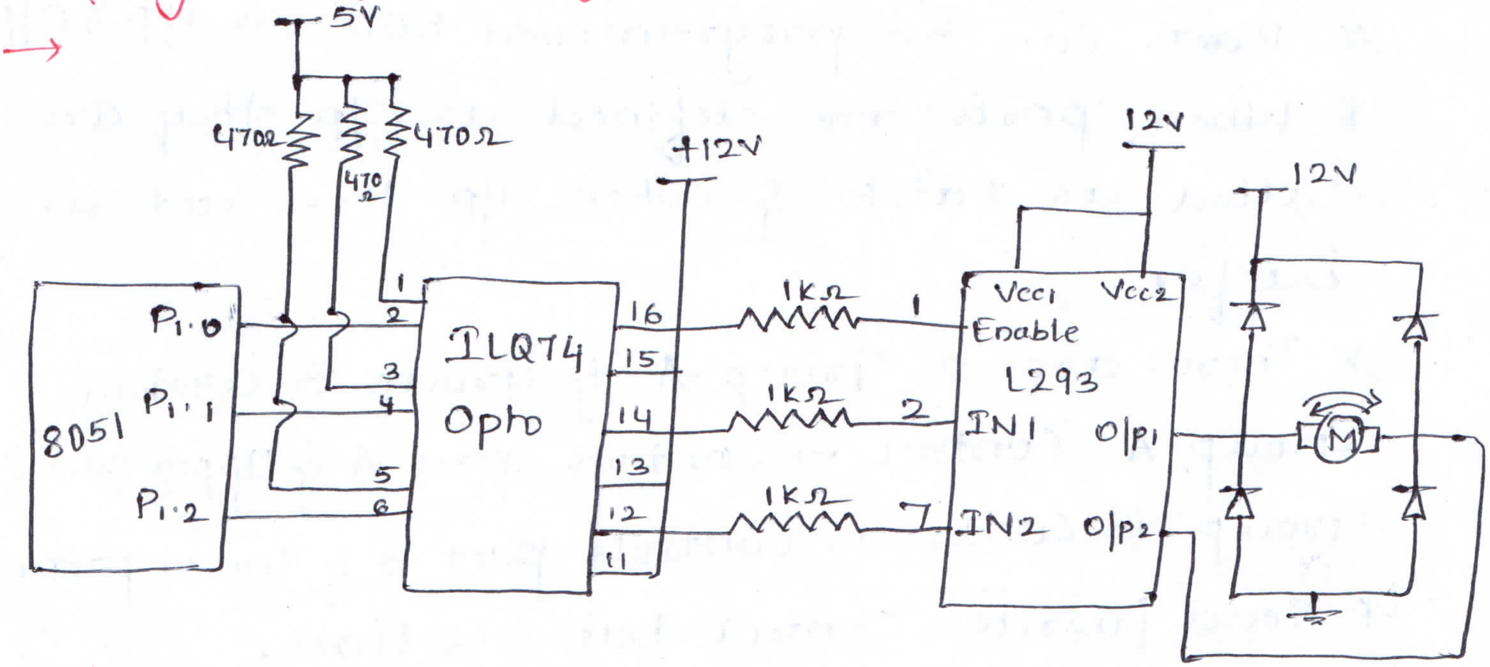
10b) Write an ALP to display message DONE (8M)

→

```
          ORG 0000h
         MOV DPTR ,#`DONE'
          MOV R5, #04h
   Main :  MOV A, @DPTR
           MOV P0, A
           ACALL Delay
           INC DPTR
           DJNZ R5, Main
           SJMP Main.
  Delay:   MOV R7, #0FFh
      D1!  MOV R6, #0FFh
      D2!  DJNZ R6, D2
           DJNZ R7, D1
           RET
           END
```

10 c) Draw the block diagram to show how 8051 is connected to DAC 0808 at port P1 Using output buffer for DAC. (4M)

→



——————* ——————* END ——————* ——————* ——————

Rotekep

RD