CBCS SCHEME

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

BEE403

Fourth Semester B.E./B.Tech. Degree Examination, June/July 2024
Microcontroller

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.	Explain the important features of 8051 microcontroller.	06	L1	CO1
	b.	Explain the working of stack and stack pointer.	06	L1	CO1
	c.	Explain addressing modes of 8051 microcontroller with an example.	08	L1	CO1
		OR			
Q.2	a.	Describe various pins of 8051 microcontroller with Pin diagram.	08	L1	CO1
	b.	Explain the PSW resister with bit pattern. Discuss the functions of each flag in detail.	06	L1	CO1
	c.	Compare microcontroller and microprocessor.	06	L1	CO1
		Module –2			
Q.3	a.	Explain the following assembler directives : i) ORG ii) EQU iii) END iv) DB	08	L2	CO2
	b.	Explain the following instructions of 8051 with examples: (1) XCHD A, @R_i (2) MOVC A, @A+PC (3) RL A (4) MUL A B (5) DA A (6) SWAP A	12	L2	CO2
		OR			
Q.4	a.	Write an 8051 ALP to find average of marks scored by students in 6 subjects. Assume the marks are stored from location 40h and its average is to be stored at location 50h.	08	L3	CO2
	b.	Discuss Call and Jump instruction types and name of branches in each case.	12	L2	CO2
		Module – 3			
Q.5	a.	Discuss the Data types in 8051C.	06	L1	CO3
	b.	Explain TMOD and TCON with its bit pattern.	06	L1	CO3
	c.	Write an 8051 C program to convert ASCII digit of 4 and 7 to packed BCD and display them on P_1.	08	L3	CO3
		OR			
Q.6	a.	Write an 8051 C program to toggle all bits of P_2 continuously every 500 ms. Use Timer1 mode-1 to create the delay.	08	L3	CO3
	b.	Write a 8051 C program to bring in a byte of data serially one-bit at a time via $P_{2.0}$ the MSB should come in first.	08	L3	CO3
	c.	Explain the characteristics and operation of Mode-2 program in 8051 timers.	04	L1	CO3
		Module – 4			
Q.7	a.	Explain RS232 handshaking signal and specify the purpose of Max232 while interfacing.	06	L2	CO4
	b.	Write an 8051 C program to transfer the message INDIA serially at 9600 baud rate.	08	L3	CO4
	c.	Explain simplex, half duplex and full duplex serial data transfer.	06	L2	CO4

		OR			
Q.8	a.	Write a program to retrieve the data serially and put them in P_1. Set the baud rate at 4800, 8-bit data and one stop bit.	06	L3	CO4
	b.	What is an Interrupt? List various interrupts of the 8051 with their corresponding vector address.	08	L1	CO4
	c.	Write an ALP program to send 'Y' serially on 8051. Use baud rate of 2400 bauds.	06	L3	CO4
		Module – 5			
Q.9	a.	Explain the internal architecture of ADC 0804.	06	L1	CO1
	b.	Explain the construction and working of stepper motor along with 4-step sequence table.	07	L1	CO5
	c.	Write a C program to rotate stepper motor continuously.	07	L3	CO5
		OR			
Q.10	a.	Explain the block diagram of 8255 chip.	10	L1	CO5
	b.	A switch is connected to pin $P_{2.7}$. Write a C program to monitor the status of SW and perform the following : If SW = 0 ; The stepper motor moves clockwise If SW = 1 ; The stepper motor moves counter clockwise	10	L3	CO5

* * * *

Fourth Sem B.E/B.Tech Degree Examination June/July 2024

Sub. Code: BEEL403

Microcontroller. Prepared by: Rajeshwari P.

Time: 3hrs Max. Marks = 100.

Module - 1.

Q1(a) Explain Important features of 8051 microcontroller. (6marks)

Soln:

1. 8bit CPU with register A & B
2. It has clock circuit.
3. It has internal 4k byte Rom.
4. " " " 128 " RAm.
5. Two 16 bit timers (T_0 & T_1)
6. full duplex serial communication.
7. Four I/O ports (P_0, P_1, P_2, P_3)
8. 2 external & 3 internal interrupts.
9. Two 16 bit registers Program Counter (PC) & Data Pointer (DPTR)
10. 8-bit stack pointer register, 8-bit program status word (PSW) & 21 8-bit special function register (SFRs)

b) Explain working of stack and stack pointer. (6marks)

Soln:

1. Stack pointer (SP) : is the Internal RAm area used by CPU to store & retrieve data.
2. Register used to access the stack is called stack pointer register (SP)
3. SP is used to hold an internal RAM address i.e called TOP of the stack.
4. When 8051 is reset, SP → 07h.
5. RAm location 08h is the first location used by the stack to store the data.
6. storing of CPU register into stack - PUSH

When data is to be placed on stack, then SP increments

SP = SP + 1.

(7) Loading content of stack bank into CPU is called Pop.

(8) If data is retrieved from SP then SP is decremented as SP = SP-1.

Q.No(8) Explain addressing modes of 8051 microcontroller with a example. (8 marks)

Soln: The CPU can access data in many ways. the data could be in a register, or in a memory or provid as an immediate value.

Addressing Modes. (AM)

① Immediate AM: In this AM, the source operand is a constant, the immediate data must be preceded by a "pound" sign "#"

Ex: 1) MOV A, #0FFh, MOV R3, #03h.

② Register Addressing Mode: This AM involves the use of registers to hold the data, that has to be manipulated. Ex: MOV A, R0, MOV R2, A.

③ Direct AM: The entire 128 bytes of RAM can be accessed using direct AM. The RAM locations 30H to 7Fh are most of them used.

Ex: MOV R0, 10h, MOV A, 40h..

④ Register Indirect A.M.: In register indirect AM, a register is used to hold the add of the data. The register itself is not the address, but rather the no in the register

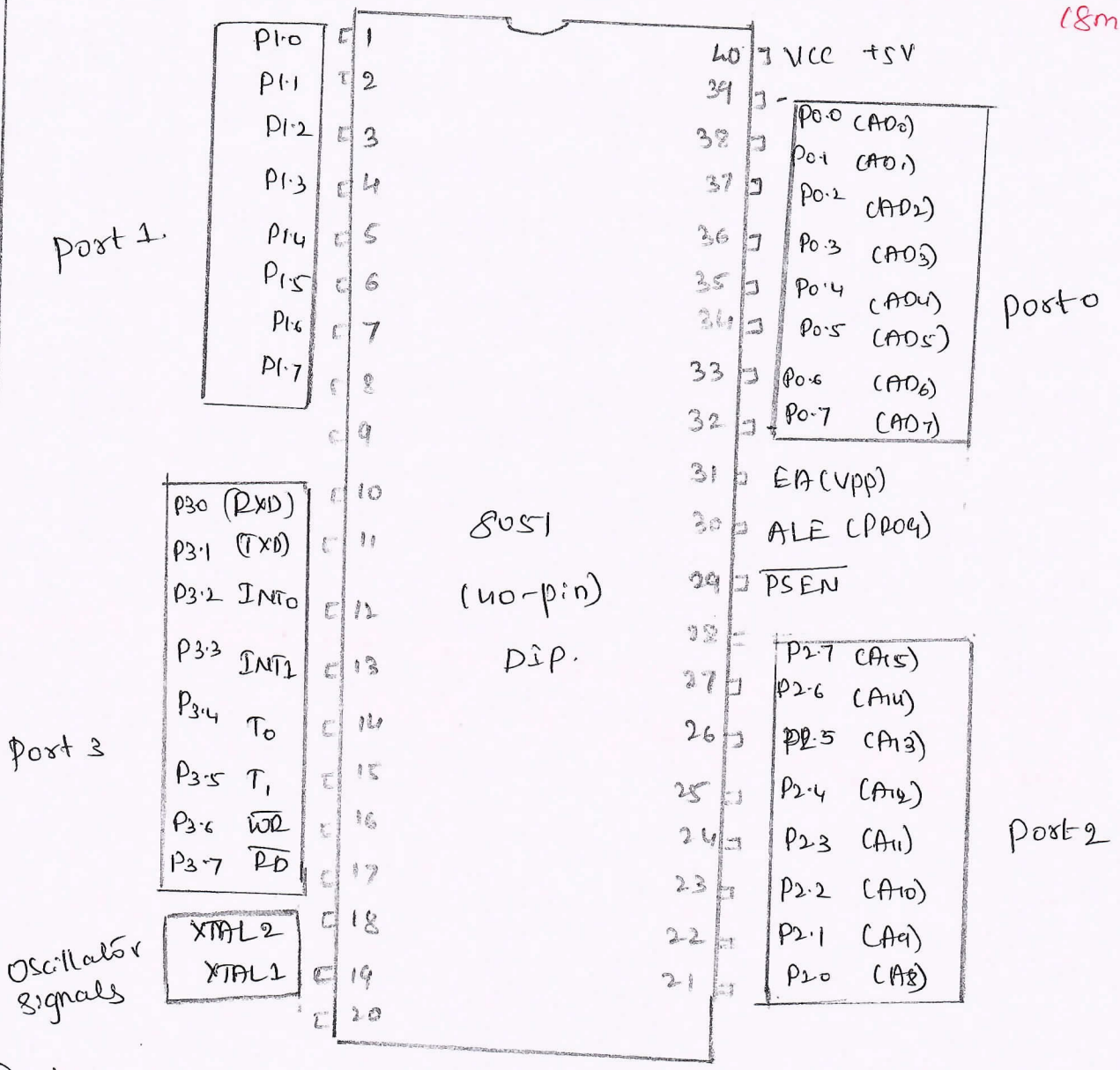
Ex: MOV @R0, A. MOV 40h, @R0.

⑤ Indexed AM: Its widely used in accessing data elements of look up table entries located in program Rom space.

Ex: MOVC A, @A+DPTR

MOVC A, @A+PC.

Q2(a) Describe various pins of 8051 microcontroller with Pin diag (8marks)



① Port 1 pin 1-8 can be configured as I/P or O/P pins.

② pin 9 is Reset it a active high signal & used as interrupt.

③ Port 3 Pins 10-17: Each of these pins can be used as I/O.

④ Pins 10 & 11: RxD & TxD. These are used for serial data Communications Transmitted out (TxD) or Received (RxD) by 8051.

⑤ Pin 12, $\overline{INT0}$ } 2 interrupt pins triggered by external Pin 13, $\overline{INT1}$ } circuits.

⑥ Pin 14 & 15, T0 & T1 are two timers Timer 0 & Timer 1. used either as Timer / Counter.

⑦ Pin 16-17 are $\overline{RD}$ & $\overline{WR}$ These are active low - used to read & write data.

⑧ Pin 18 & 19: XTAL2 & XTAL1 It a Quartz crystal oscillator.

(9) Pin 20 used as Ground Pin.

(10) Pins 21-28 Port 2 : If external memory is not used these pins can be used as i/p or o/ps.

(11) Pin 29 $\overline{PSEN}$ (Program store Enable) It active low pin used for interfacing to external Rom.

(12) Pin 30 (ALE) =1 , port 0 provides lower order address
=0 " " " as data lines.

(13) Pin 31 $\overline{EA}$ (External Access) It used whenever external memory used.

(14) Pin 32-39 Port 0: If external memory not used these are used as I/Os.

(15) Pin 40 = Vss.

Q2(6) Explain the PSW register with bit pattern. Discuss the functions of each flag in detail.

Soln:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CY	AC	F0	RS1	RS0	OV	—	P

(1) PSW is an 8-bit register. It is also called as flag register. 6bits of PSW registers are used & 2bits are unused.

(2) 4 flags are called conditional flags or math flags.

(3) RS0 & RS1 are used to change Register Bank.

(4) Two unused bits are called user defined flags.

(5) CY used to perform arithmetic & logic operations

(6) AC (Auxillary carry) : It set when carry is generated from $D_3 \rightarrow D_4$ bit.

(7) RS_1 & RS0 are used used to select the Register Banks. ie 0, 1, 2, 3.

(8) OV overflow flag: It 8bit signed number OV is set to 1 if carry from $D_6 \rightarrow D_7$ but no carry out of D_7 (CY=0) If there is carry out from D_7 (CY=1).

(9) Parity flag : Indicates the no of 1's present in Acc.
P=1 if acc has no of 1's odd
P=0 " " " " " even.

Q2(c) Compare Microcontrollers & Microprocessor (6 marks)

Soln:

	Microprocessor	Microcontrollers.
①	Its an integrated circuit or working register, timing & control ckts, clock ckts are fabricated on a single chip.	Its an integrated ckt. in which all ALU working regist Timing & clock ckts memory (RAM) IIo ports are fabricat on single chip.
②	Used for General purpose	Used for specific purpose.
③	Provides flexibility	Don't provide flexibility.
④	Many instructions to move data blw memory & CPU.	Not designed for rapid movement only 1 or 2 instns related to it.
⑤	It has one or two bit handling instructions.	It has many bit handling instructions.
⑥	It requires more hardware	It requires less hardware

Module - 2

Q3(a) Explain the following assembles directives.
(i) ORG (ii) EQU (iii) END (iv) DB. (8marks)

Soln: (i) ORG: xxxh: Originate the following code at starting address xxxxh. This directive directs the assembles that the instruction of following is to be stored at 8200h.

Ex: org 8200h.

② EQU: Equ xxxxh.

The equ directive equate the name to the number xxxxh. for eg: Suraj equ 30h.

This directive will equate the name Suraj to the number 30h.

③ DB xx; The db stands for define a byte. It directs the assembles to place the number xx next in memory.

for eg : ORG 0000h
db 12h

Address byte
0000 12h

(4) dw xx: ~~dw~~ END: This indicates the assembles the end of the source file. ie .asm file.

ORG 0000h.

eg: MOV A, #30h
MOV B, #40h
ADD A, B.
END.

Q (3)(b) Explain the following instructions of 8051 with examples (12 marks)

Soln: (a) XCHD A, @Ri : It also called the Exchang Digit.

XCHD exchanges only the lowes nibble of accumulator ie bit (0-3) with lowes nibble of RAM location.

Ex: XCHD A, @R1

B.E A = FFh
R1 = 50h
50h = 00h

A.E: A = F0h
R1 = 50h
50h = 0Fh.

(b) MOVC A, @A+PC: Its the instruction is used to transfes data from code memory into the accumulator.

Syntax: MOV A, @A+PC

Compute the address as PC+A = 0x100 + 0x02 = 0x102

(c) RL A : It rotates the bits of accumulator to the left.

Ex: BE = A = (28h)
00101000

A.E = 85(h).
10000101

(d) MUL AB : Its used for multiplication of contents of Accumulator & B-register.

MOV A, #05h
MOV B, #07h

5x7 = 35d
= 23(h)

B.E : A = 05h, B = 07h
A.E : A = 23h, B = 00h.

Q3(b) contd....

(5) DAA : Decimal Adjust Accumulator after addition

This instruction is used after addition of BCD numbers to convert the result back to BCD.

Ex:
```
MOV A, #47h
MOV B, #38h
ADD A, B
DAA.
```

```
  47 h
+ 38 h
  7F h → (invalid BCD)
+  6
  85(h) → valid BCD.
```

AC = 1

(6) SWAP A: It swaps the lower nibble (A0 - A3) with upper nibble (A4 - A7) inside register (A).

Ex:
```
MOV A, #95h.
SWAP A.
```
A = 59(h).

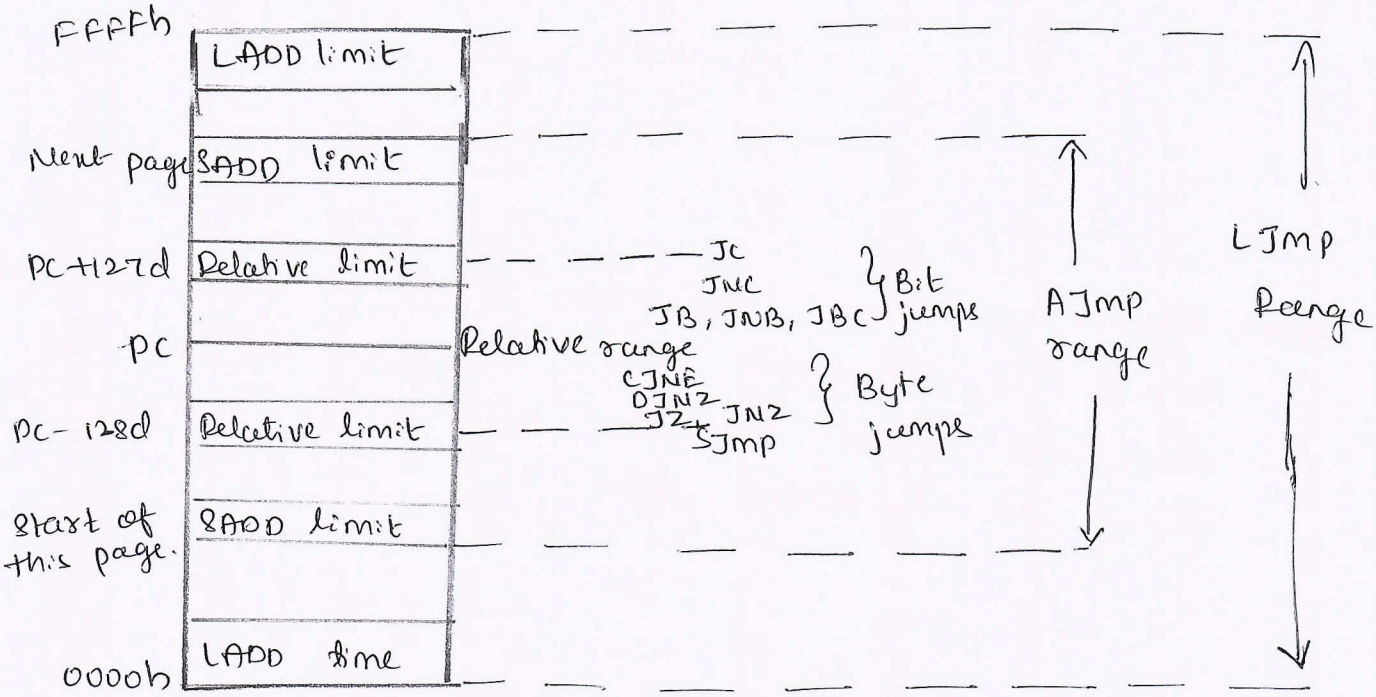
Q4(a) Write an 8051 ALP to find average of marks scored by students in 6 subjects. Assume the marks are stored from location 40h & its average is to be stored at location 50h.

Soln:
```
      MOV R0, #40h
      MOV R2, #6h
      CLR A.
      MOV R7, A
Again: ADD A, @R0.
      DA A
      JNC Next
      INC R7
Next : INC R0
      DJNZ R2, Again
      mov R0, #6
      DIV A, B
      mov R0, A.
```

Q4(b) Discuss call and Jump instruction types & name of branch in each case. (12 marks)

Soln:

- Jump & Call instruction replaces the contents of program Counter (PC) with new address & program execution to start from that new address
- The difference (in bytes) of this new add from add in program where jump & call instructions is called Range of Jump or Call.



① Relative Range: The jump can be within -128 bytes (backward) to +127 bytes (forward jump) of memory relative to the address of current program counter. 1st byte holds opcode & 2nd byte relative add of target location.

② Absolute Range: In 8051 program memory is divided into logical divisions called pages of each of 2Kbyte. Max size pgm memory is 64Kbytes. ∴ each page = 2Kbytes. 1st byte 5bits of PC holds page no & lower 11 bits hold add within that page.

③ Long Absolute Range: This range allows the jump to anywhere in the program location from 0000h to FFFFh. Its 3 byte instruction 1st byte opcode & 2nd, 3rd byte represents 16 bit add of target location.

Module 3

Qs (a) Discuss the Data types in 8051 C. (6 marks)

Soln: (1) Unsigned chas: Its an 8 bit datatype that takes values in range of 00 - FF h (ie 2 to 255) used for setting a counter value to represent ASCII characters.

(2) Signed chas: Its also an 8 bit data type that uses MSB 'D7' to represent +ve or -ve values. So only 7 bit are used to represent the magnitude of number. ranges from -128 to 127.

(3) Unsigned int: Its a 16 bit data type that ranges from 0000 to FFFFh ie (0 to 65535) & it is used to define a 16 bit variable such as memory address.

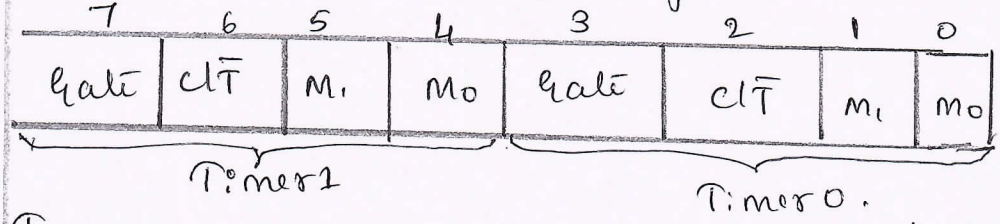
(4) Signed int: Signed int is a 16 bit data byte type that uses MSB bit D15 to represent +ve or -ve value. Signed int ranges from -32768 to +32767. used to represent +ve or -ve value.

(5) Sbit: Its used to access single bit of SFR register & ports P_0-P_3. Size is 1 bit either 0 or 1.

(6) Sfr: Its used to access byte size SFR register & size is 1 byte. RAM address 80h - FFh is used.

Qs (b) Explain TMOD & TCON with its bit pattern. (6 marks)

Soln: (1) TMOD. (Timer mode) Register.



Timer 1 & 0 uses same register TMOD to set various times operation modes. In this lower 4 bits are for Timer 0 & higher 4 bits for Timer 1.

Timer 1: Bit 7 Gate = 1. Timer 1 will run only when INT 1 ($P_{3.3}$) pin is high. Gate = 0 " " " regardless of " .

Bit 6: C/T = 1 Counter mode.

C/T = 0 Timer "

Bit 5 & 4 : are used for selecting timer mode.

for Timer 0. : Bit 3 Gate : When Gate = 1. Timer 0 will run only when $\overline{INT}0$ ($P_{3.0}$) pin is high.

Gate = 0 Timer 0 will run regardless of $\overline{INT}0$.

Bit 2 C/$\overline{T}$: C/$\overline{T}$ = 1 counter mode.

C/$\overline{T}$ = 0 Timer mode.

Bit 1 & 0 : Used as M_1 & M_0 to select timer mode.

(3) TCON

7	6	5	4	3	2	1	0
TF_1	TR_1	TF_0	TR_0	IE_1	IT_1	IE_0	IT_0

TF_1 → Timer 1 overflow flag.

TR_1 → " run control bit

TF_0 → Timer 0 overflow flag.

TR_0 → Timer 0 run control bit.

IE_1 → External interrupt 1 Edge flag.

IT_1 → External interrupt 1 signal type control bit

IE_0 → " " 0 Edge flag.

IT_0 → External Interrupt 0 signal type control bit.

Q5 (c) Write an 8051 program to convert ASCII digit of 4 & 7 to packed BCD & display them on P_1 (8marks)

Soln:

```
ORG 0000H.
MOV A, #34h
SUBB A, #30h
  MOV R0, A.
MOV A, #37h
SUBB A, #30h
SWAP A
ORL A, R0
MOV P1, A.
END.
```

(Q6a) Write an 8051 C program to toggle all bits of P_2 continuously every 500 ms. Use Timer1 mode-1 to create the delay. (8 marks)

Soln: 8051 has 12MHz crystal, each machine cycle = 1 µs.

To generate 500 ms delay 500 ms = 500,000 µs / 1 µs = 500000 counts

Initial count = 65536 − 50000 = 15536 (3CB0H in hex)

TH1 = 0X3C, TL1 = 0XB0

C- Program.

```
#include <reg51.h>
Void Timer1_Delay500ms();
  Void main() {
   while(1)  {
    P2 = ~P2;
    Timer1_Delay500ms();
   }
}
  Void Timer1_Delay500ms() {
     TMOD =0X10;
     TH1 = 0X3C;
     TL1 = 0XB0;
     TR1 = 1;
   while (TF1==0)
     TR1=0;
     TF1=0;
  }.
```

(b) Write a 8051 Cprogm to ~~toggle all bits of P_2~~ continuously ~~every 500ms~~ bring in a byte of data serially one bit at a time via $P_{2.0}$ the MSB should come first. (8 marks)

Soln:

```
#include <reg51.h>
unsigned char receivedByte =0;
  Void delay()
{
    unsigned int i;
  for(i=0; i<100; i++)
```

contd..

```
{
    receivedByte = receivedByte<<1;
    if (P2 & 0x01)
    {
        receivedByte = 0x01;
    }
    delay();
}
}
void main()
{
    while()
    {
        receiveSerialByte();
    }
}
```

Q6(c) Explain the characteristics & operation of Mode2 program in 8051 timers. (4 marks)

Soln: ① Its an 8bit timer ie it allows value from 00h to FFh

② After TH is loaded it gives a copy of TH to TL

③ Then timer start by SETB TRx.

④ TL register starts to count up it starts to increment & when it reaches FFh it overflows from FFh to 00h during that time TFx = 1.

⑤ When TF = 1, TL register is automatically reloaded with original value kept by TH register.

⑥ To repeat the process we must clear TF register.

MODULE-4.

Q7(a) Explain RS 232 handshaking signal & specify the purpose of Max 232 while interfacing. (6 marks).

Soln: RS-232 is a standard for serial communication used to connect for serial communication used to connect computer & peripheral devices.

Common RS-232 Handshaking signals.

① Data Terminal Ready (DTR) indicates the DTE is ready to communicate.

(2) Data set ready (DSR) indicates Data communication Equipment such as modem ready for communication.

(3) Request to send (RTS) The DTE sends this signal to DCE to indicate it is ready to send data.

(4) Clear to send (CTS): The DCE responds with this signal to indicate its ready to receive data.

(5) Carrier Detect (CD) - Indicates connection has been established

(6) Ring Indicator (RI) to indicate an incoming call.

(7) Transmit Data (TXD) The line on which data is transmitted from DTE to DCE.

(8) Received Data (RXD) line on which data is received by DTE from DCE.

(9) Gnd The reference ground for all.

The purpose of Max 232 in RS 232 interfacing is to convert voltage levels between RS-232 serial ports and TTL devices. It helps RS232 operate at voltage levels of 13 to 25V

(Q7)(b) Write an 8051 C program to transfer the message INDIA serially at 9600 baud rate. (8marks)

Soln:

```
#include <reg51.h>
void serial_init(void);
void serial_send(char ch);
void send_string(char *str);
void main()
{
  serial_init();
  while(1)
  {
    send_string("INDIA");
    while(1);
  }
}
```

contd...

```
Void serial_init (void)
{
    TMOD = 0x20;
    TH1  = 0xFD;
    SCON = 0x50;
    TR1  = 1;
}

Void serial_send (char ch)
{
    SBUF = ch;
    while (!TI);
    TI = 0;
}

    Void send_string (char *str)
    {
        while (*str)
        {
            serial_send (*str++);
        }
    }
```

Baud Rate Calculation

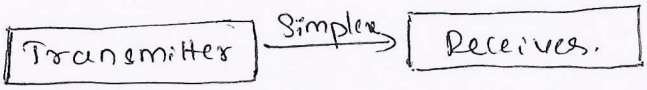
$$\text{Baud Rate} = \frac{\text{Crystal Frequency}}{12 \times (256 - TH1)}$$

For a 11.0592 MHz Crystal & 9600 baud rate

$$TH1 = 256 - \frac{11059200}{12 \times 32 \times 9600} = 0xFD$$

c) Explain simplex, half duplex, and full duplex serial data transfer. (6 marks)

Soln: ① Simplex:

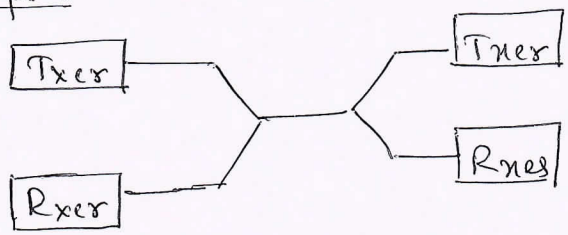


* In simplex transmission data is transmitted in only one direction. There is no possibility of data transfer in other directions.

Ex: from computer to printer.

(2) Duplex: In data transmission of the data can be transmitted & received. Its a duplex transmission.

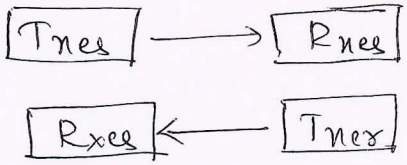
(a) Half -Duplex.



In half duplex, data transfer is possible only one way at a time. It allows data transmission in both direction. but not simultaneously.

Ex: Walkie-Takie.

(b) Full Duplex.



Full duplex transmission allows data transfer in both directions simultaneously.

Ex: Transmission through telephone lines.

Q8(a) Write a program to retrieve the data serially & put them in P_1. Set the baudrate at 4800, 8 bit data & one stop bit. (6marks).

Soln: Set the baud rate to 4800

- Using 11.0592 MHz crystal, the formula for baud rate

$$\text{Baud Rate} = \frac{\text{Oscillator Frequency}}{12 \times (256 - TH1)}$$

- for 4800baud, TH1 = 0xFA (-6 in decimal)

```
ORG 0000H
MOV TMOD, #20h
MOV TH1, #0FAh
MOV SCON, #50h
SETB TR1.
MAIN_LOOP:
    JNB RI, MAIN_LOOP
    MOV A, SBUF
    MOV P1, A.
    CLR RI
    SJMP MAIN-LOOP
END.
```

Q8(b) What is an Interrupt? List various interrupt of the 8051 with their corresponding vector address (8marks).

Soln: An interrupt is a signal that temporarily stops the the execution of the main program & forces the μc to execute a special routine called an Interrupt.

Six interrupts in 8051.

In reality only five interrupts are available in 8051 but many manufacturers data sheet includes Reset as well

1) When Reset pin activated, 8051 jumps to address location 0000h.
2) Two interrupts are set aside for To & T1. Memory location 000Bh & 001Bh belongs to To & T1.
3) Two interrupts are set aside for hlw external interrupt INT0 & INT1. They are also referred as Ex0 & Ex1 Memory location 0003h & 0013h
4) Serial Communication has a single interrupt that belongs to both receive & transmit. & memory add. 0023h.

Q8(c) Write an ALP to send '4' serially on 8051. Used baud rate of 2400 bauds. (6marks)

Soln:

```
ORG 0000h
 LJMP MAIN
ORG 0023h
LJMP SERIAL_ISR
MAIN:
   MOV TMOD, #20h
   MOV TH1 , #0FAh
   MOV SCON, #50h
     SETB TR1
     SETB EA
     SETB ES
      SJMP $
```

```
SERIAL_ISR:
  JNB RI, CHECK_TI
    MOV A, SBUF
    MOV P1 , A
     CLR RI
CHECK_TI
    JNB TI, EXIT_ISR
     CLR TI
EXIT_ISR
     RETI.
```

MODULE -5:

Q9(a) Explain the internal architecture of ADC 0804. (6 marks)

Soln: The internal str of the ADC0804 consists of several key components that work together to perform analog to digital conversion, these includes.

① Analog Input (Vint) and Reference voltage (Vref/2): Vint pin receives the analog voltage (0V to 5V) & Vref = 5V that needs to be converted.

② Sample & Hold circuit: It captures the analog i/p vtg to ensures accurate conversion.

③ Successive Approximation Register (SAR): Its the heart of ADC which determines the digital output by comparing the input voltage against a reference. It starts with MSB and tests each bit in sequence until the full 8-bit output is determined.

④ Comparator compares i/p vtg with the o/p of DAC.

⑤ Control Logic & Clock Generates necessary timing for conversion.

⑥ Output Latches & 8 bit data bus available for the μc.

Q9(b) Explain the construction & working of stepper motor along with 4-step sequence table. (7 marks)

soln: The operation of a stepper motor is based on electro magnetic attraction & repulsion. When a coil in the stator is energised, it creates a magnetic field, that attracts or repels the motor causing it to move step by step

4-step sequence table.

Step	Coil A	Coil B	Coil C	Coil D
1	1	0	1	0
2	0	1	1	0
3	0	1	0	1
4	1	0	0	1

1 = ON

0 = OFF

Step 1: Coils A & C are energized pulling the rotor to specif position.

Step 2: Coil B is energised while A is turned off causing th rotor to move slightly.

Step 3: Coil C is turned off. & Coil D is energized, furthes rotating the rotor.

Step 4: Coil A is energized again while B is turned off, completing one full step cycle.

Q9(c) Write a C program to rotate stepper motor continuously. (7marks)

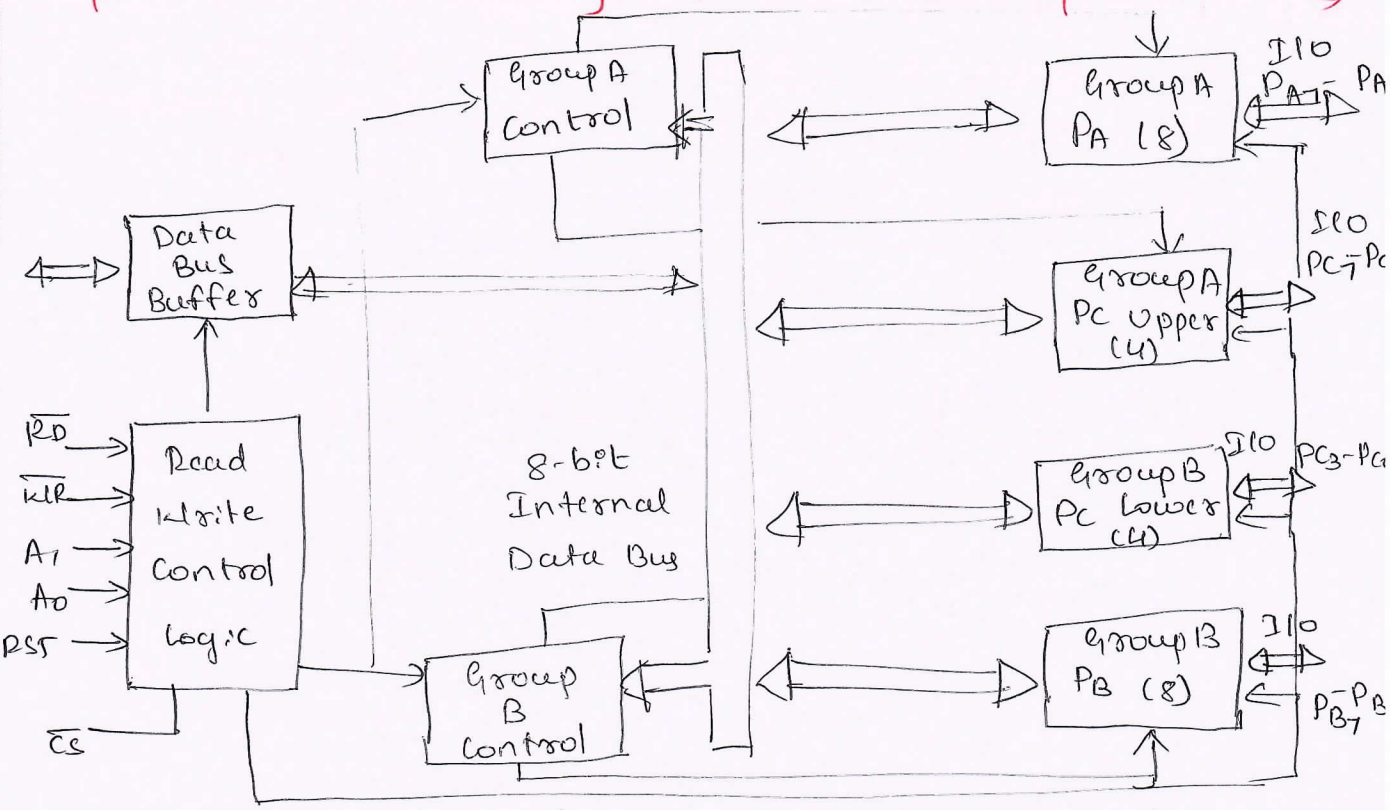
Soln:

```
#include <reg51.h>
#define STEP_DELAY 5
void delay_ms (unsigned int ms)
{
    unsigned int i,j;
    for (i=0 ; i<ms; i++)
      for (j =0; j< 1275; j++);
    void stepper_rotate ()
    {
        while (1) {
        P2 = 0x09;
       delay_ms (STEP_DELAY);
        P2 =0x0C;
       delay_ms (STEP_DELAY);
          P2 = 0x06;
       delay_ms (STEP_DELAY);
          P2 =0x03;
          delay_ms (STEP_DELAY);
    }
}

        void main() {
          P2 = 0x00;
          stepper_rotate();
        }
```

Q10 (a) Explain the block diagram of 8255 chip. (10 marks)

Soln:



① It is a 40 pin chip (Dual Inline package)

② It has 3 8-bit ports P_A, P_B & P_C

③ P_A can be programmed as either i/p or o/p port also it can be programmed as a bidirectional port.

④ P_B can be programmed as i/p or o/p port.

⑤ P_C is grouped into two 4 bit ports ($P_{C4} - P_{C7}$) (port C upper) & $PC_3 - PC_0$ (port C lower) each can be programmed as i/p or o/p port.

⑥ It has two 8 bit ports P_A & P_B, two 4-bit ports PC (upper) & Pc (lower), Data bus buffer & read/write control logic.

⑦ Ports can be programmed to an i/p or o/p ports.

⑧ When ports are defined as o/p they act as Latches & when i/p they act as buffers.

⑨ RD/WR control has six lines. Their fn are $\overline{CS}$ → chip select, A1/A0 select specific ports & control register

⑩ RESET is an active high signal. When signal goes high it clears all ports (A, B & C) & control registers.

Q10(b) A switch is connected to pin $P_{2.7}$. Write a C program to monitor the status of SW & perform the following

If SW=0; the stepper motor moves clockwise.

If SW=1; " " " " Anti " . (Comarks)

Soln:

```
# include <reg51.h>
# define STEP_DELAY 5

sbit SW = P2^7;
sbit coil-A = P2^0;
sbit coil-B = P2^1;
sbit coil-C = P2^2;
sbit coil_D = P2^3;

void delay_ms (unsigned int ms) {
    unsigned int i, j;
    for (i=0; i<ms; i++)
        for (j=0; j<1275; j++);
}
    void stepper_clockwise() {
        Coil_A = 1; Coil_B = 0; Coil_C = 0; Coil_D = 1;
        delay_ms (STEP_DELAY);
        Coil_A = 1; Coil_B = 1; Coil_C = 0; Coil_D = 0;
        delay_ms (STEP_DELAY);
        Coil_A = 0; Coil_B = 1; Coil_C = 1; Coil_D = 0;
        delay_ms (STEP_DELAY);
        Coil_A = 0; Coil_B = 0; Coil_C = 1; Coil_D = 1;
        delay_ms (STEP_DELAY);
    }
void stepper_counterclockwise() {
    Coil_A = 0; Coil B = 0; Coil C = 1; Coil D = 1;
    delay_ms (STEP_DELAY);
    Coil_A = 0; Coil_B = 1; Coil_C = 1; Coil_D = 0;
    delay_ms (STEP_DELAY);
    Coil_A = 1; Coil_B = 1; Coil_C = 0; Coil_D = 0;
    delay_ms (STEP_DELAY);
```

```
    Coil_A = 1; Coil_B = 0;  Coil_C = 0;  Coil_D = 1;
    delay_ms (STEP_DELAY);
}

    void main() {
        P2 = 0x80;
    while (1) {
      if (sw == 0) {
        stepper_clockwise ();
      } else {
        stepper_counterclockwise ();
    }.
```

——— O. END ——— O ———

(Rajeshwari P)

HEAD
Dept. of Electrical & Electronics Engg
KLS's V. D. Institute of Technology
HALIYAL-581 329.

Dean (Academics)
KLS VDIT HALIYAL