Modified

CBCS SCHEME

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

17EE52

Fifth Semester B.E. Degree Examination, Dec.2019/Jan.2020
Microcontroller

Time: 3 hrs. Max. Marks: 100

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

Module-1

1 a. Draw the block diagram of 8051 μC. Explain the working of:
(i) Program counter and data pointer
(ii) Accumulator and register B
(iii) Register bank, stack and stack pointer **(10 Marks)**

b. Draw and explain program status word register of 8051 μC. Calculate the status of carry, auxiliary carry and parity flags after the addition of (i) 55h and 52h (ii) 91h and 92h. What is the hexadecimal sum in each case? **(10 Marks)**

OR

2 a. Explain register indirect addressing mode. State its advantages. **(05 Marks)**

b. Explain indexed addressing mode with MOVC and MOVX instructions. **(05 Marks)**

c. What is memory address decoding? Explain the steps in interfacing memory chips to μC. Develop the interfacing circuit to connect 4K × 8 memory IC using logic gates as decoder. Assume the memory address from 3000 h to 3FFF h. **(10 Marks)**

Module-2

3 a. Define assembler directive. Explain DB and ORG directives. **(05 Marks)**

b. Write a program to multiply 35 by 10 using repeated addition. Save the result in R6. Neglect carry. **(05 Marks)**

c. Explain the working of MUL AB and DIV AB instructions. **(05 Marks)**

d. State the following instructions as valid or invalid. Give reasons:
(i) MOV A, @ R4 (ii) PUSH R0 (iii) MOV R5, R6
(iv) POP 00h (v) MOV P1, #0FFh **(05 Marks)**

OR

4 a. Explain the working of port 0 as input port. State its dual role. **(05 Marks)**

b. Calculate the delay for the following program. Assume clock frequency as 11.0592 MHz.

		Machine cycle
	MOV R3, #255	1
GO:	NOP	1
	NOP	1
	DJNZ R3, GO	2
	RET	1

(05 Marks)

c. How the following numbers are represented in 8051?
(i) 4 (ii) –4 (iii) 82 (iv) –82 (v) –128 **(05 Marks)**

d. Explain the working of overflow flag. After the addition of +45 with +04, what is the status of overflow flag and what is the sum, according to μC? **(05 Marks)**

Module-3

5 a. State and explain the advantages of using 'C' program for 8051 μC. **(05 Marks)**

b. Write 8051 'C' program to toggle bit D7 of port 0, 60,000 times. **(05 Marks)**

c. Explain the differences between sbit, bit and sfr declarations. **(05 Marks)**

d. Write 8051 'C' program to convert ASCII digits '9' and '2' to packed BCD and display it on port P2. **(05 Marks)**

OR

6 a. Explain the bit status of TMOD register. **(05 Marks)**

b. Write an assembly program to generate square wave with ON time = 5 ms and OFF time = 20 ms on all pins of port-1. Use Timer0 in Mode1. Assume crystal frequency = 11.0592 MHz. Calculate the duty cycle. Explain TH0, TL0 and TMOD calculations. **(10 Marks)**

c. Explain the characteristics and operations of mode-2 program in 8051 timer. **(05 Marks)**

Module-4

7 a. Explain the bit status of SCON register. With XTAL = 11.0592 MHz, calculate the THI value needed for the band rates; (i) 9600 (ii) 2400. **(10 Marks)**

b. A square wave is being generated at pin P1.2. This square wave is to be sent to a receiver connected in serial form to 8051.Write an assembly language program for this. Explain the calculations of TMOD, SCON, TH1 value. Assume Timer0 and Timer1 in Mode2. Assume baud rate = 9600 and XTAL = 11.0592 MHz. **(10 Marks)**

OR

8 a. Compare interrupts versus polling methods, in 8051 interrupts. **(05 Marks)**

b. Explain the 6 interrupts in 8051. Also state its ROM location. **(05 Marks)**

c. Write an assembly program to get data continuously from port 0 and send it to port P1 while simultaneously creating a square wave of 200 μs period on P2.1 Use Timer0 to create square wave. Assume XTAL = 11.0592 MHz. Explain IE, TMOD, TH0 calculations. **(10 Marks)**

Module-5

9 a. State advantages of LCD over multi-segment LEDS. Explain the architecture and working of 14 pin LCD. Draw its schematic diagram. **(10 Marks)**

b. Explain the interfacing circuit of DAC to 8051 μC. If I_{ref} = 2 mA, calculate the DAC output if all the inputs to DAC are high. **(05 Marks)**

c. Calculate V_0 of sawtooth wave (with respect to DAC interface) with the following program. Assume R_F = 5 KΩ in I/V converter in DAC circuit interfacing.[Refer fig.Q9(c)]

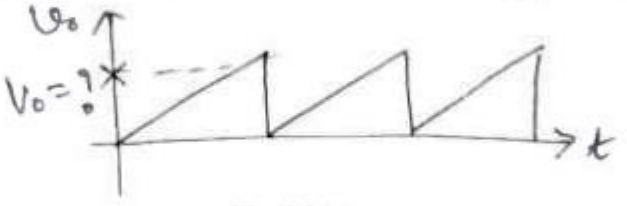


Fig.Q9(c)

```
Program: MOV A, #00h
         MOV P1, A
      GO: INC A
         SJMP GO
```

(05 Marks)

OR

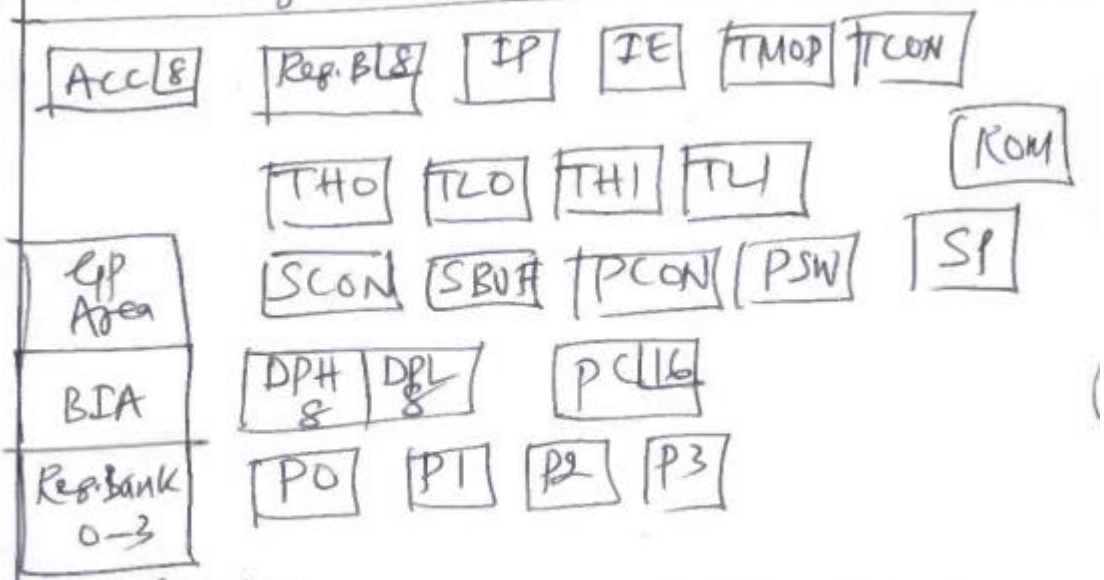
10 a. Explain the construction and working of stepper motor. Also explain 2-ϕ, 4 step stepping sequence, step angle and steps per revolution. **(10 Marks)**

b. Explain the control word format of 8255 IC. What is the control word for all the ports as output ports? **(05 Marks)**

c. Explain the principle of opto isolator and its purpose in interfacing to 8051 μC. **(05 Marks)**

* * * * *

1.(a) block diagram of 8051



Program counter (PC) automatically increments after every inst byte is fetched & altered by some inst. it does not have internal addr.

Data pointer: has DPH & DPL used to give addr for internal & external code data access; specified as DPTR.

Accumulator & Reg. B: 'Acc' is versatile of all registers & normally stores of arithmetic of logical operations stored in Acc & Reg. B is like any other register, but is used in MUL & DIV instructions to save results.

Register bank, stack & SP: There are 4 register bank ie bank 0 to bank 3, each having 8 registers giving 32 general purpose registers; stack refers to area ie used with certain opcodes to store & retrieve data quickly.

SP is to hold internal RAM address called top of the stack. The addr holding SP register is the location in internal RAM where the last byte of data stored.

1.(b) PSW register

D7							D0
CY	AC	F0	RS1	RS0	OV	–	P

if CY out of D7 CY → 1
if CY from d3 to d4 in BCD; AC → Set 1
if odd no of 1's in Acc: P → set
RS1 & RS0 are for bank select & OV for signed arithmetic. (untestable)

(i) 55h + 52h = A7h
0101 0101 ∴ CY=0
0101 0010 AC=0
1010 0111 P=1

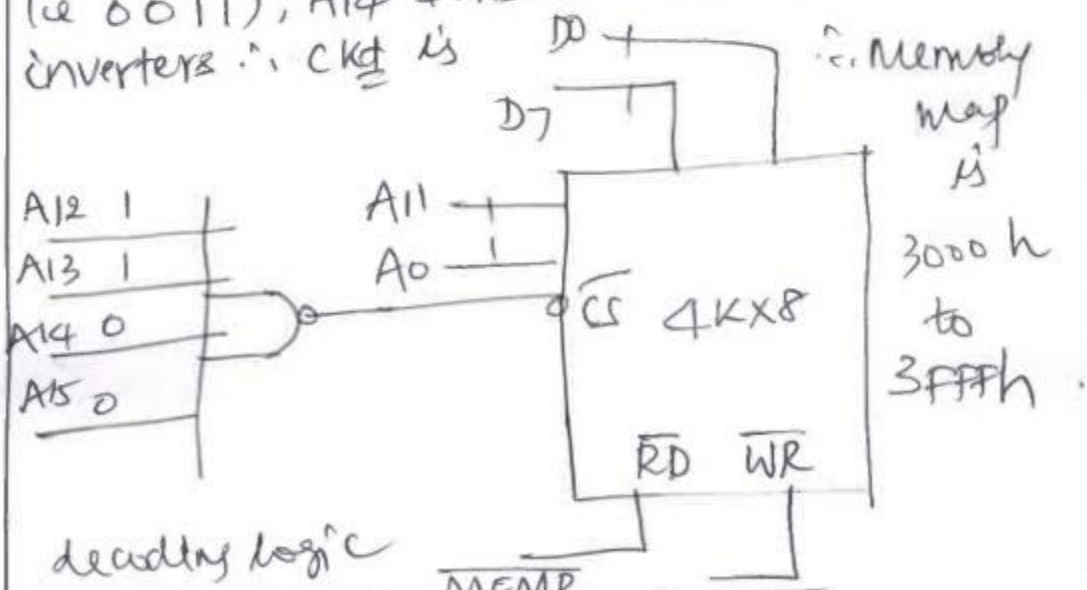
(ii) 91h + 92h = 23h
1001 0001 ∴ CY=1
1001 0010 AC=0
(1) 0010 0011 P=1

2 (a) Register R0 or R1 used as pointer to the data
Eg MOV A,@R0 & MOV @R1, B;
Merits: it makes accessing data dynamic rather than static as in direct addressing mode. other registers cannot be used.

2(b) indexed addressing mode used in accessing data elements of a look up table located in program ROM space of 8051 by inst MOVC A, @A+DPTR; 8051 has another 64 KB of memory space for data storage only, can be accessed by MOVX A,@A+DPTR

5. c) Memory address decoding means mapping the address range of various memory ICs in connecting to µC. Steps are: 1) data bus of µC is connected directly to data pins of memory (2) control signals RD & WR from µC connected to $\overline{OE}$ of $\overline{CE}$ and WE pins of memory respectively 3) depending on no of address lines to be connected, required address lines are connected directly to memory IC & remaining address lines are connected to address logic decoder using logic gates of 3 to 8 decoder & this o/p is connected to $\overline{CS}$ of memory IC.

For 4KX8 memory, no of address lines $= 2^2 \cdot 2^{10} = 2^{12}$ ∴ A0 to A11 are connected directly; since memory map is 3000 to 3FFFh let $A_{12} = 0/1$, $A_{13} = 0/1$ $A_{14} = 0$ & $A_{15} = 0$ (i.e 0011); A14 & A15 are connected through inverters ∴ ckt is



decoding logic

A15	A14	A13	A12	A11	A10	———	A0
0	0	1	1	0	0	0 ———	0
0	0	1	1	1	1	1 ———	1

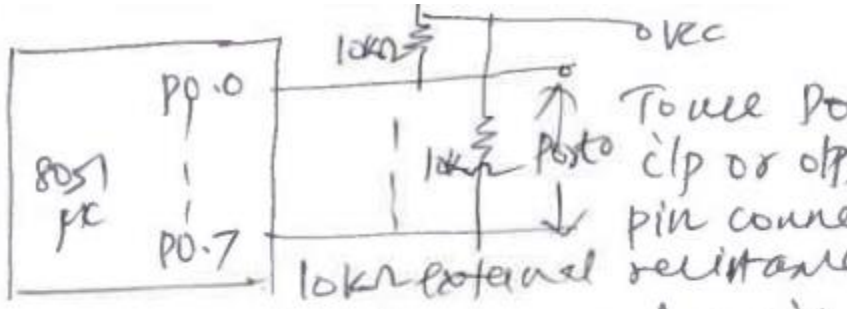
3.(a) assembler directive is a message to assembler that tells something it needs to know in order to carry out assembly process; are pseudo instr & does not take any memory space.
DB means define byte, we can define variable in decimal or hex or binary or ASCII
Eg data1 DB 28 ; ie 28 is decimal (assembler = 1Ch)
data2 DB 28h; ie assembler = 28h as it
ORG: origin gives beginning address of program
Eg. ORG 100h) etc

3(b)
```
    MOV A, #00; clear Acc
    MOV R2, # 10; Multiplier R2 = 10 = 0Ah
40; ADD A, # 35; 35 + 00 = 35 in Acc
    DJNZ R2, 40; repeat addition '10' times
    MOV R6, A; answer in Reg. R6 = 350
```
(Here answer is 5Eh & carry is neglected)

3(c) MUL AB → means, lower byte of ans in Acc & upper byte in Reg. B.
DIV AB → No (Acc) / Dr (B); after division quotient goes to Acc & remainder in Reg. B

3(d) MOV A, @R4; invalid, because only registers R0 or R1 used as pointer
PUSH R0; invalid, because absolute addr should be used, we cannot use name in push
MOV R5, R6; invalid, because we cannot move data among registers ie not Acc only.
POP 00h; valid as absolute addr is used.
MOV P1, #0FFh; valid, as directly data can be written in port P1.

4(a)



To use Port 0 as i/p or o/p, each pin connected to

P0 is open drain, P0 used as i/p port by writing all 1's to it & then data is received from that port.

```
eg: MOV A, #0FFh
    MOV P0, A  ; Port 0 is i/p now
    MOV A, P0  ; get data from P0 to Acc
```

dual role:- P0 is also called AD0 - AD7. When ALE = 0 it provides data D0 - D7 & when ALE = 1 it provides addr A0-A7

4(b)

$$T = \frac{12}{11.0592 \times 10^6} = 1.085\ \mu s$$

$$\therefore \text{delay} = [1 + (1+1+2) \times 255 + 1] \times 1.085\ \mu s$$

$$\therefore \text{delay} = 1.10887\ ms$$

4(c)

+4 : 04h ; −4 : 0000 0100 (Taking 2's complement)

1111 1100 = FCh

+82 : 0101 0010 = 52h

1010 1110 = AEh

−128 : & +128 : 1000 0000 (2's)

1000 0000 ⇒ 80h

∴ +4 : 04h { +82 → 52h } −128 → 80h

−4 : FCh { −82 → AEh }

4(d) OV = CY out of D7 ⊕ CY from D6 to D7

+45 = 0010 1101 = 2Dh

+04 = 0000 0100 + = 04h

0011 0001 = 31h = 49 in decimal

Here NO CY from d6 to d7 & no CY out of D7

∴ OV = 0 ⊕ 0 = 0 (Resets)

Sum as per ADSI is 31h & OV resets as sum is < +127.

5.(a) 'C' is easier to modify & update, less time consuming; we can use code available in libraries; C code is portable to other µC with little modifications.

5(b)
```
#include <reg51.h>
 sbit mybit = P0^7; // i.e Bit 7 of port0
void main (void)
   {unsigned int q;
    for(q=0;q<=60000;q++)
      {mybit = 0;
       mybit= 1; // toggle P0.7
      }
   }
```

5.(c) sbit: we can define 1 bit, SFR bit addressable only; bit is 1 bit declaration for RAM bit addressable only & sfr is 8 bit data, but RAM addr 80h – FFh only

5(d)

```
#include<reg51.h>
void main (void)
{unsigned char bcd
unsigned char w='9';
unsigned char z='2';
w = w &0X0F; //ie W=39 get 09h made 0
w = w <<4; // ie get 90h, shift left
z = z &0X0F; // ie 2=32 get 02h
bcd = w|z; // OR W & 2 =92h
P2 = bcd; // send it to P2
}
```

6.(a)

Gate	C/T	M1	M0	Gate	C/T	M1	M0

← Timer - 1 → ← Timer - 0 →

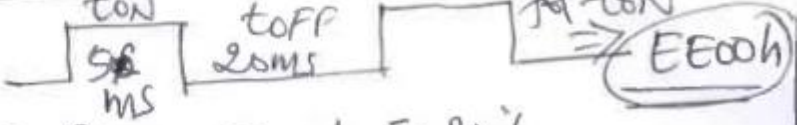
M1 M0
- 0 0 Mode 0 13 bit timer
- 0 1 Mode 1 16 bit timer
- 1 0 Mode 2 8 bit autoreload
- 1 1 Mode 3 Split timer mode

C/T = 0 timer, = 1 Counter

Gate = 0 Software method to start/stop

Gate = 1 Hardware method to start/stop

6.(b)

Software is 

$\therefore$ duty cycle $= \frac{5}{5+20} = \frac{5}{25} = \frac{1}{5} = 20\%$

XTAL = 11.0592 MHZ $\therefore T = 1/\frac{11.0592}{12} = 1.085\ \mu s$

For OFF time

$\Rightarrow \frac{20ms}{1.085\ \mu s} = 18,433 \therefore 65536 - 18433 = 47103 = B7FFh$

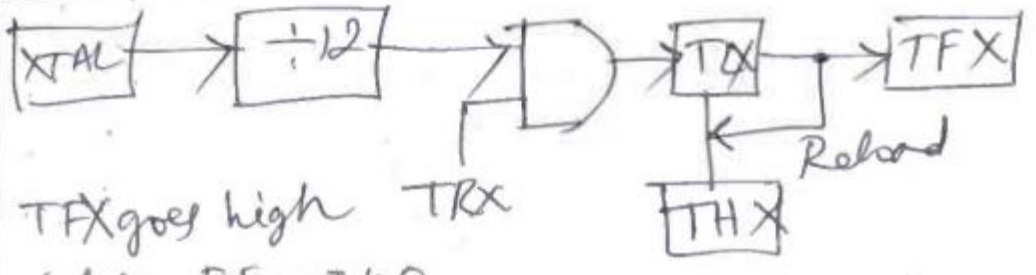
For ON time: $5ms/1.085\ \mu s = 4608$; $65536 - 4608 = 60928$

TMOD = 0000 0001 = 01h = timer 0 in Mode 1

Program: MOV TMOD, #01h; timer 0 in mode 1
REPEAT: MOV TL0, #0FFh; } off time
MOV TH0, #0B7h; } B7FFh
ACALL DELAY; goto delay loop
MOV TL0, #00h; } ON time
MOV TH0, #0EEh; } EE00h
MOV P1, #0FFh; all bits of P1 are high
ACALL DELAY; goto delay loop
SJMP REPEAT; continuously generate

; delay
DELAY: SETB TR0; start timer 0
WAIT: JNB TF0, WAIT; wait time 0 to roll over
CLR TR0; stop timer 0
CLR TF0; clear TF0 also
RET; go to main program
END; end of main program.

6(c)



X is 0 or 1 for timer 0 or timer 1

7(a)

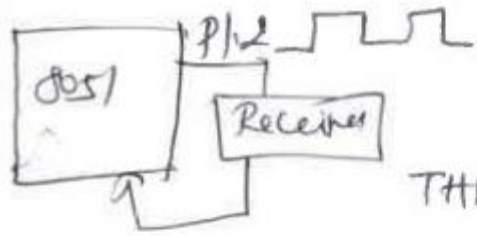
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

T = 1/(11.0592/12) = 1.085 μs & 921.6K/32 = 28800

for 9600; TH1 = 28,800/9600 = 3 = −3 = FDh

for 2400; TH1 = 28800/2400 = 12 − 12 = F4h.

7(b)



TMOD: 00100010
SCON: 01010000
TH1 = 28800/9600 = 3 = -5 = FDh

ie SQ wave is generated at P1.2, it P1.2 = high, data FFh is transmitted serially & if P1.2 = low, data 00h is transmitted & this data converted to ~~ll~~ pulses at receiver side to regenerate SQ wave.

```
=> MOV TMOD, #22 h; timer 0 & timer 1 in mode 2
   MOV SCON, # 50h; baud variable
   MOV TH1, # FDh; 9600 baud
   MOV TH0, # 00h; let count value = 00h
   SETB TR1; start timer 1
   MOV A, # 00; clear Acc
   CLR P1.2; let bit 2 of port 1 = LOW
REPEAT: SETB TR0; start timer 0
WAIT  JNB TF0, WAIT; wait timer 0 roll over
      CPL A; complement Acc
      CPL P1.2; toggle bit 2 of Port 1
      MOV SBUF, A; keep (Acc) in SBUF Reg
      CLR TR0; stop timer 0
      CLR TF0; clear TF0 flag also
DO:   JNB TI, DO; send data
      CLR TI; clear TI flag
      SJMP REPEAT; repeat continuously
      END;
```

8(a) Compare interrupts & polling methods
at least 5 points

8(b) RESET → 0000h INT1 → 0013h
INT0 → 0003h TF1 → 001Bh
TF0 → 000Bh Serial COM RI & TI → 0023h

8(c) $TH0 = \frac{200}{2} = \frac{100\,\mu s}{1.085\,\mu} = 92 = -92 = A4h$

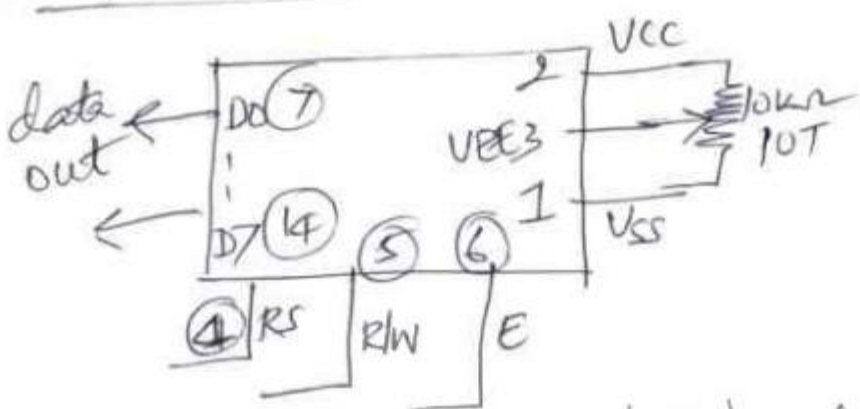
TMOD = 02h; enable timer 0 interrupt

IE = 1000 0010 = 82h

```
        ORG 00h
        LJMP Main
; ISR for timer 0 to get SQ wave
        ORG 000Bh; timer 0 ISR addr
        CPL 2.1; toggle bit 1 of port 2
        RETI; return from interrupt
; main program                    auto-reload
MAIN:   MOV TMOD, #02; timer 0 in mode 2
        MOV P0, #0FFh; P0 is i/p port
        MOV TH0, # A4h; for half period delay
        MOV IE, # 82h; enable timer 0 int
        SETB TR0; start timer 0
GO:     MOV A, P0; get data from P0.
        MOV P1, A; send data to P1
        SJMP GO; do it continuously unless
                 interrupted by TF0.
```

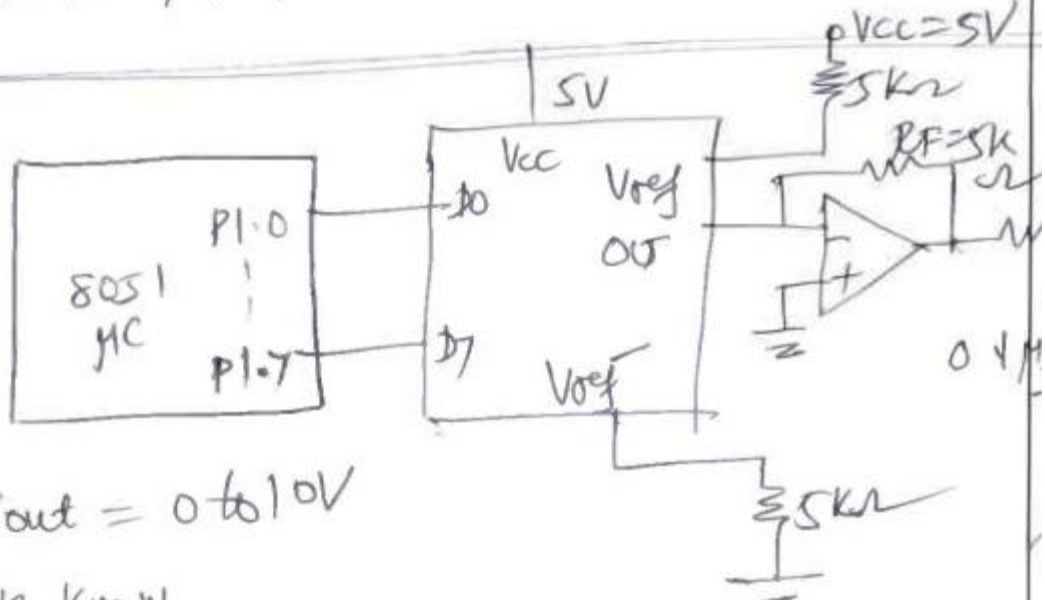
9(a) Merits: declining price of LCDs; ability to display numbers, chars & graphics, where LED limited to nos & few chars; ease of programming; use of refreshing controller with LED.

14 pin LCD



RS; Register select; RS=0 'CR' is selected
RS=1 data register selected
R/W : =0 for Writing & 1 for reading
E : H → L ; D0 - D7 : data out

9. (b)



Vout = 0 to 10V

We know,

$$I_{out} = I_{ref}\left(\frac{D7}{2} + \frac{D6}{4} + \frac{D5}{8} + \frac{D4}{16} + \frac{D3}{32} + \frac{D2}{64} + \frac{D1}{128} + \frac{D0}{256}\right)$$

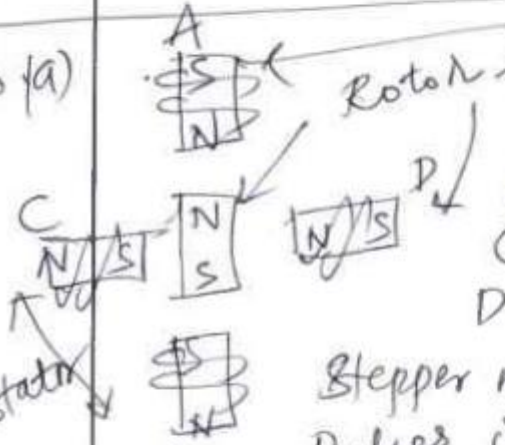
Now D7 -- D0 = 1111 1111

$$\therefore I_{out} = 2mA\,(0.5 + 0.25 + 0.125 + 0.0625 + 0.03125 + 0.015625 + 7.8125 \times 10^{-3} + 3.90625 \times 10^{-3})$$

$$= 1.99mA$$

9(C) Here ACC is incremented from 00h to FFh & then FF to 00, & continuously.
for FFh $I_{out} = I_{ref}(\ ) = 1.99mA$ (same as in Q.9 (C)) $\therefore V_{out} = 1.99mA \times 5K\Omega$
$= 9.96$ volts.

10(a)

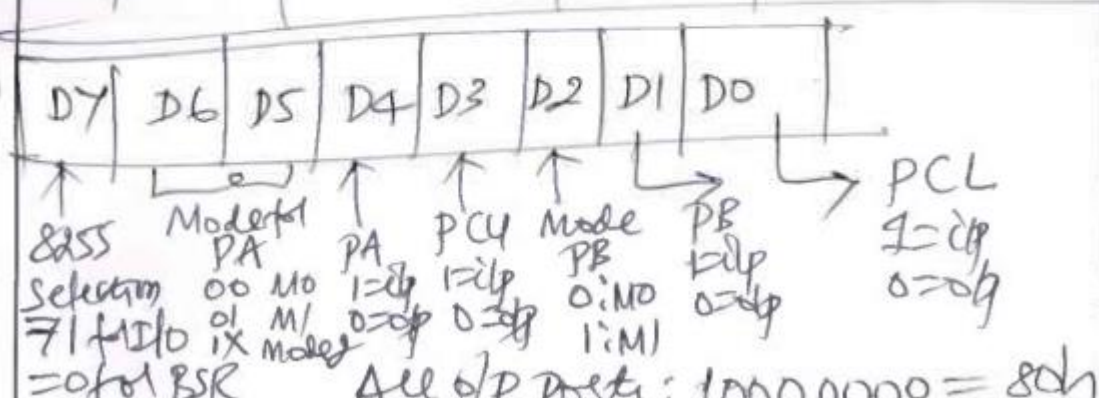


Stepper motor translates electrical pulses into mechanical movement used in dot matrix printers, disk drives etc; has permanent magnet rotor surrounded by stator

4 step, 2 Φ sequence

Step No	Winding A	B	C	D
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1

10(b)



All o/p ports: 10000000 = 80h

10(c) IL74 opto-isolator: used to isolate 2 parts of a system. For ex in motor driving ckt motors can produce back emf. ie a HV spike produced by sudden change of current ie $V = L(di/dt)$, here we use opto isolators. it has LED as Tx & photo sensor receiver separated by gap. when current flows through LED, it transmits light signal a/s the gap & receiver produces same signal with same phase but different current & amplitude.