CBCS SCHEME

USN 2VD19EE028

18EE52

Fifth Semester B.E. Degree Examination, Feb./Mar. 2022
Microcontroller

Time: 3 hrs. Max. Marks: 100

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

Module-1

1 a. With a neat block diagram, explain 8051 microcontroller's each block. **(08 Marks)**

b. Explain Addressing modes of 8051 with example for each. **(06 Marks)**

c. Explain the bit pattern of Program Status Word (PSW). **(06 Marks)**

OR

2 a. Show the status of CY, AC and P flags after adding the following data:
(i) 9CH and 64H
(ii) 88H and 93H **(06 Marks)**

b. Describe the functions of various pins of 8051 microcontroller with Pin diagram. **(08 Marks)**

c. Compare the Microprocessor and Microcontroller. **(06 Marks)**

Module-2

3 a. Explain the following assembler directives:
(i) ORG (ii) EQU (iii) END (iv) DB **(08 Marks)**

b. Write an ALP program for the addition of two 16-bit numbers, the numbers are *FC45H and* 02ECH. Place the sum in R7 and R6, R6 should have lower Byte. **(06 Marks)**

c. Explain the following instructions with an example:
(i) RRC A (ii) XCHD A, @R_P (iii) SWAP A **(06 Marks)**

OR

4 a. Write an assembly language program to complement the value 55H, 700 times. **(06 Marks)**

b. Explain JUMP and CALL instructions. With a neat diagram explain the range of JUMP and CALL instructions. **(08 Marks)**

c. With a neat diagram, explain working of PORT 0. **(06 Marks)**

Module-3

5 a. Explain the different Data types supported by 8051 C microcontroller. **(08 Marks)**

b. Write an 8051 C program to create a square wave of frequency 2500Hz on port pin $P_{2.7}$. Use timer1 in mode 2 to create the delay f = 11.0592 MHz. **(08 Marks)**

c. Explain TMOD Register. **(04 Marks)**

OR

6 a. Assume crystal frequency as 11.0592 MHz. What value do we need to load into the timer register if we want to have a time delay of 5 msec? Write an assembly language program for the same with timer 0 to create a pulse width of 5 msec on $P_{2.3}$ **(06 Marks)**

b. Write a C program for counter 0 in mode 2 to display seconds and minutes on P_1 and P_2. Assume that 60Hz external clock is supplied to TO pin ($P_{3.4}$). **(06 Marks)**

c. Write a C program to toggle all bits of P_2 continuously every 500 msec. Use Timer 1 in mode 1 to create delay. **(08 Marks)**

Module-4

7 a. Explain the bit status of SCON register. **(06 Marks)**

b. Explain the importance of MAX 232 line driver for connecting RS232 to 8051 with a neat sketch show the interface of RS232 to 8051 using MAX 232. **(08 Marks)**

c. Write a C program for 8051 to transfer the letter "A" serially at 4800 baud continuously. Use 8-bit data and 1-stop bit. **(06 Marks)**

OR

8 a. Explain different interrupts of 8051 indicating their vector address. **(06 Marks)**

b. Write an 8051 C program to transfer the message "WELCOME" serially at 9600 baud, 8-bit data, 1-stop bit. Do this continuously. **(08 Marks)**

c. Write a 8051 C program to receive bytes of data serially and put them in port P_1. Set the baud rate at 4800, 8-bit data and 1 stop bit. **(06 Marks)**

Module-5

9 a. Explain the construction and working of stepper motor. Explain the 4-step sequence, step angle and steps per revolution. **(06 Marks)**

b. Explain the architecture and working of 14 pin LCD. Draw the interface diagram of LCD with 8051 microcontroller. **(08 Marks)**

c. With a neat diagram explain interfacing of DAC 0808 to 8051 microcontroller. **(06 Marks)**

OR

10 a. Explain the internal architecture of ADC 0804. Explain the interfacing diagram of 8051 microcontroller with ADC 0804. **(10 Marks)**

b. A switch is connected to pin $P_{2.7}$. Write a program to monitor the status of SW and perform the following :

(i) If SW = 0, the stepper motor moves in clockwise.

(ii) If SW = 1, the stepper motor moves in counter clockwise. **(10 Marks)**

* * * * *

Microcontroller (Feb/Mar-2022)

Scheme of Evaluation.

Subject: Microcontroller

Subject code: 18EE52

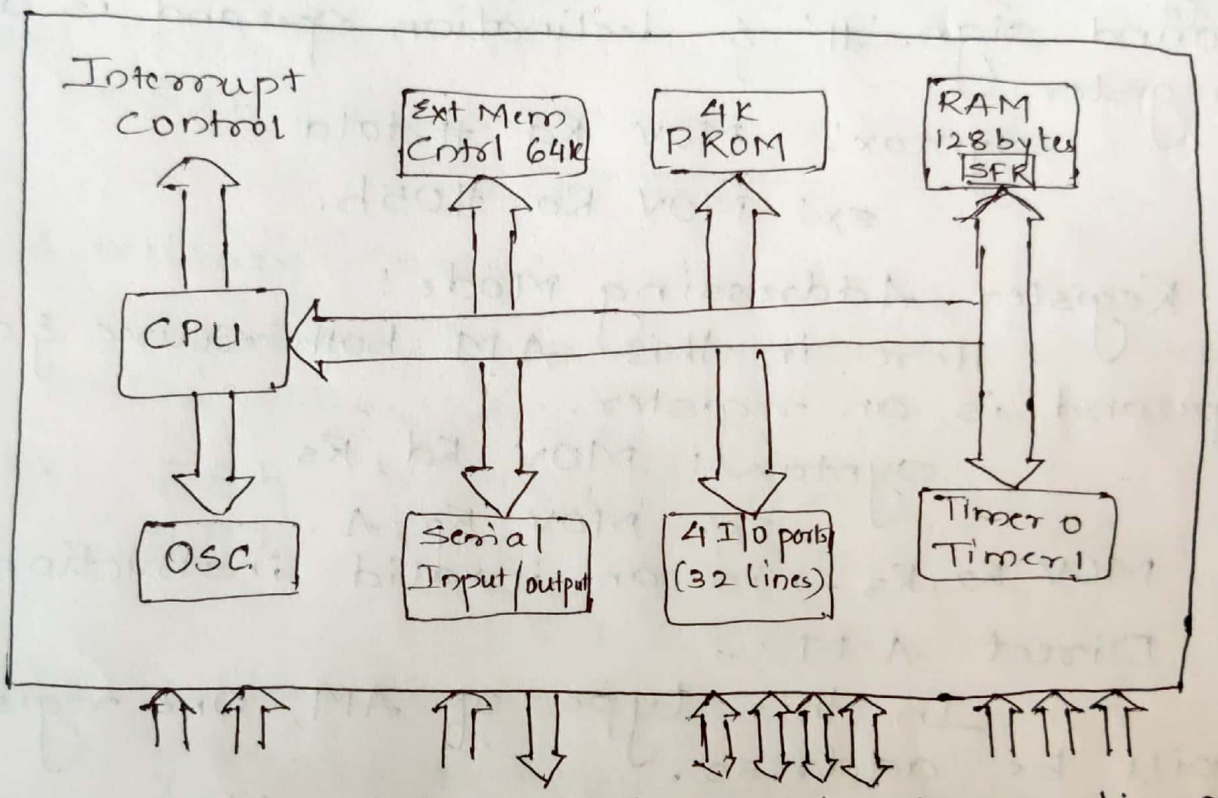
Sem: 5th (odd)

Department: Electrical & Electronics Engg.

Prepared By: Prof. Ravindra Motekar.

1a) With neat block diagram, explain 8051 microcontroller's each block. (08M)

→



The figure above shows the block diagram of 8051.

1) It has 8 bit CPU with registers A & B.

2) It has 4K bytes of ON chip ROM & can be extended upto 64K.

3) The RAM size is of 128 bytes, which has SFR's in it.

4) It has 12 16-bit timers, Timer 0 & Timer 1

5) 4 ports P_0, P_1, P_2 & P_3. each port is of 8-bits & can be used as I/O port.

6) Oscillatory circuit
7) Supports full duplex serial communication.

1b. Explain Addressing Modes of 8051? with an example for each. (6M)

Ans: Def: The various ways of accessing any data is called as Addressing Modes.

1. Immediate A.M →

In Immediate A.M, the source is a number or an immediate data with a pound sign. '#' & destination operand is a register.

Syntax: MOV Rd, #data

Ex: MOV R0, #05h.

2) Register Addressing Mode:

Here in this A.M both source & dest Operand is an register.

Syntax: MOV Rd, Rs.

Ex: MOV R2, A.

MOV R2, R3 → is an invalid instruction.

3) Direct A.M: →

In this type of AM, One register will be address.

Syntax: MOV Rd, address

Ex: MOV R1, 40h.

4) Register Indirect A.M: →

In this type of A.M, Only register R0 & R1 is used precedded by '@' symbol.

Syntax: MOV Rd, @Rs.

Ex: MOV A, @R0.

5) Indexed A.M: -

MOVC A @A+dptr.

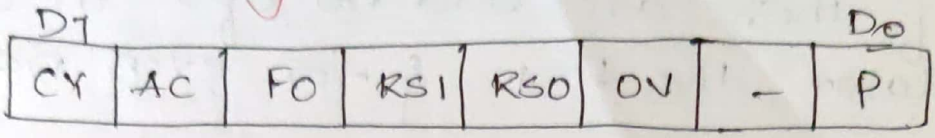
MOVC A, @A+PC.

6) Bit inherent A.M :-

CLR 07h.

1C. Explain the bit pattern of Program status word (PSW) register (6M)

→



1) Carry flag :-

If there is a carry generated after D7, then CY=1 else CY=0.

Ex: F0h 11110000

+82h 10000010 CY=1

[1] 01110010

2) Auxillary carry :-

If there is a carry generated from D3 to D4 then AC=1 else AC=0.

Ex: 88h 1000 1000 AC=1

(D4 D3 D2 D1 D0)

+29h 0010 1001

[1]

10110001

3) F0 → User Defined bit.

4) RS1 & RS0 → Register select bits.

RS1	RS0	Bank
0	0	0
0	1	1
1	0	2
1	1	3

5) OV → Overflow flag.

If there is a carry generated from D6 to D7 but after D7 there is no carry or viceversa then AC=1 else AC=0.

Ex: 0100 0001 AC=1
0100 1000
1
1000 1001

6) Parity flag (P) :-
If there are odd no of 1's in an Accumulator the P=1 & for even no of 1's P=0.

2a) Show the status of CY, AC & P flags after adding the following data (6M)

a) 9Ch & 64h

1001 1100
+ 0110 0100
1 1 1 1 1
0000 0000

CY = 1
AC = 0
P = 0

b) 88h & 93h

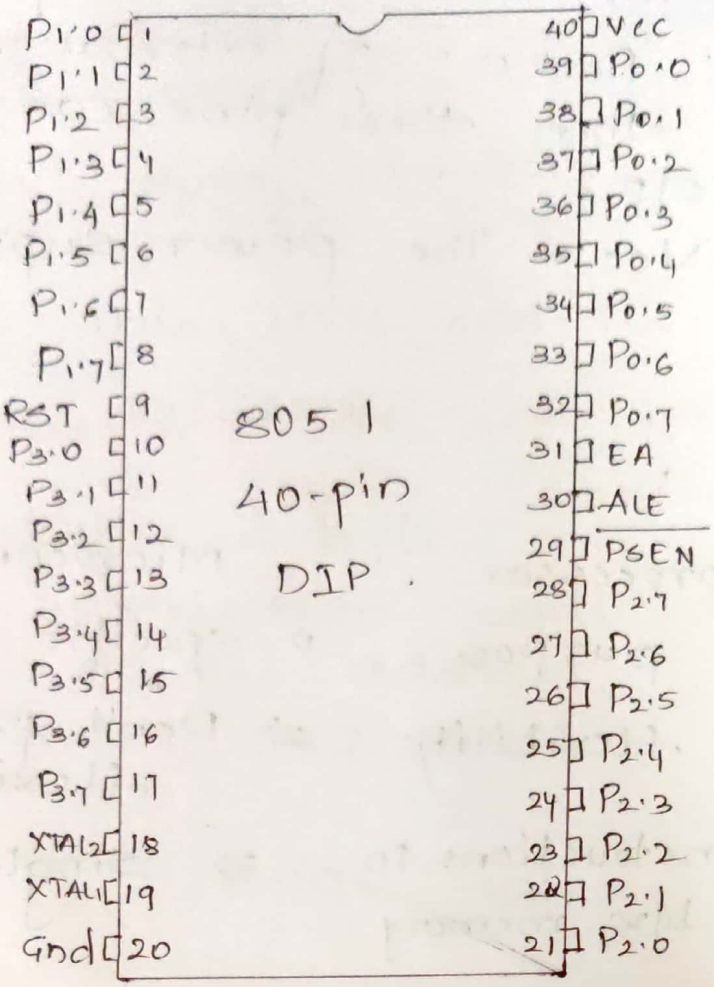
1000 1000
1001 0011
1
0001 1011

CY = 1
AC = 1
P = 0

2b) Describe the functions of various pins of 8051 microcontroller with pin diagram. (8M)

→ * Pins 1-8: Port 1:
Each of these pins can be configured as i/p or o/p pins.
* Port register name: P1
address 90h.

* Pin 9: Reset: It is an active high signal i/p, high pulse is applied to this pin. µc will terminate all its activities & resets the Program Counter.



* Pin 10-17 → Port 3 can be used as i/p & o/p port. It is a dual functionality port with pins TxD, RxD, $\overline{RD}$, $\overline{WR}$, T0, T1, $\overline{INT0}$ & $\overline{INT1}$.

* Pin 18 & 19: XTAL1 & XTAL2. The Crystal freq of Oscillator is about 11.0592 MHz.

* Pin 20: Vss : Ground signal.

* Pin 21 - 28: Port 2 pins can be used as i/p & o/p port. It is used for higher addressing.

* Pin 29: $\overline{PSEN}$ → Program store Enable pin It is active low pin used with EA during interfacing of ext ROM.

* Pin 30: ALE → Address latch enable.

ALE = 1; port 0 is used as address lines

ALE = 0; port 0 is used as data lines.

* Pin 31: $\overline{EA}$: external Access. This pin is used whenever external memory is used.
* Pin 32-39: Port 0: If external memory is not used then these pins can be used as i/p's or o/p's.
* Pin 40: Vcc – The power supply is +5V.

2c) Compare the Microprocessor & Microcontroller? (6M)

Microprocessor	Microcontroller
1) General purpose	1) specific purpose.
2) Provides flexibility	2) Don't provide flexibility.
3) Many instructions to move data b/w memory & CPU.	3) Single instruction.
4) One or two bit handling or bit manipulation instruction	4) Many bit handling instructions.
5) Require more hardware	5) Requires less hardware.
6) Versatile	6) Non versatile.

Module 2

3a) Explain the following assembler directives. (8M)

a) ORG: Originate of the program. This instruction will originate the program from the starting address.

Ex: ORG 0000h.

b) EQU: Equate:

This directive will equate the name to address

ex: suraj equ 30h.

c) END: This will end the program.

Ex: END.

d) dB: Define a byte:

This instruction will define a byte of data to the address

for ex: ORG 0000h
db 12h.
0000h 12h.

3b) Write an ALP program for the addition of two 16-bit nos. the nos are FC45h & 02ECh. Place the sum in R7 & R6. R6 should have lower byte. (6M)

→

FC45
02EC
FF31h.

8500 = 45h
8501 = ECh
8502 = FCh
8503 = 02h

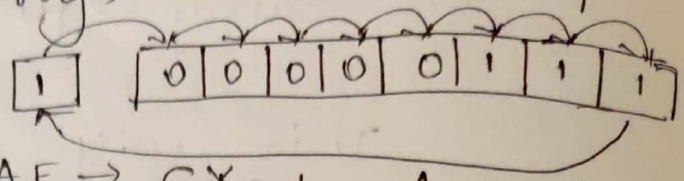
R6 = 31h
R7 = FFh.

```
ORG 0000h
mov dptr, #8500h
movx a, @dptr.
mov r6, a.
inc dptr.
movx a, @dptr.
add a, r6.
mov r6, a.
inc dptr
movx a, @dptr
mov r7, a
inc dptr
movx a, @dptr
addc a, r7
mov r7, a   nop   end.
```

3c. Explain the following instructions with example. (6M)

(i) RRC A : This instruction will rotate the Accumulator right with carry.

for ex: A = 07h
CY = 01



AE → CY = 1, A = 83h.

ii) XCHD A, @Ri

Ex: XCHD A, @R0

A = 07h
R0 = 50h
50h = 01h

→ After execution A = 01h
R0 = 50h
50h = 07h.

iii) SWAP A :

This will swap the lower nibbles of accumulator with higher.

Ex: SWAP A
A = 95h ⟶ AE → A = 59h

OR

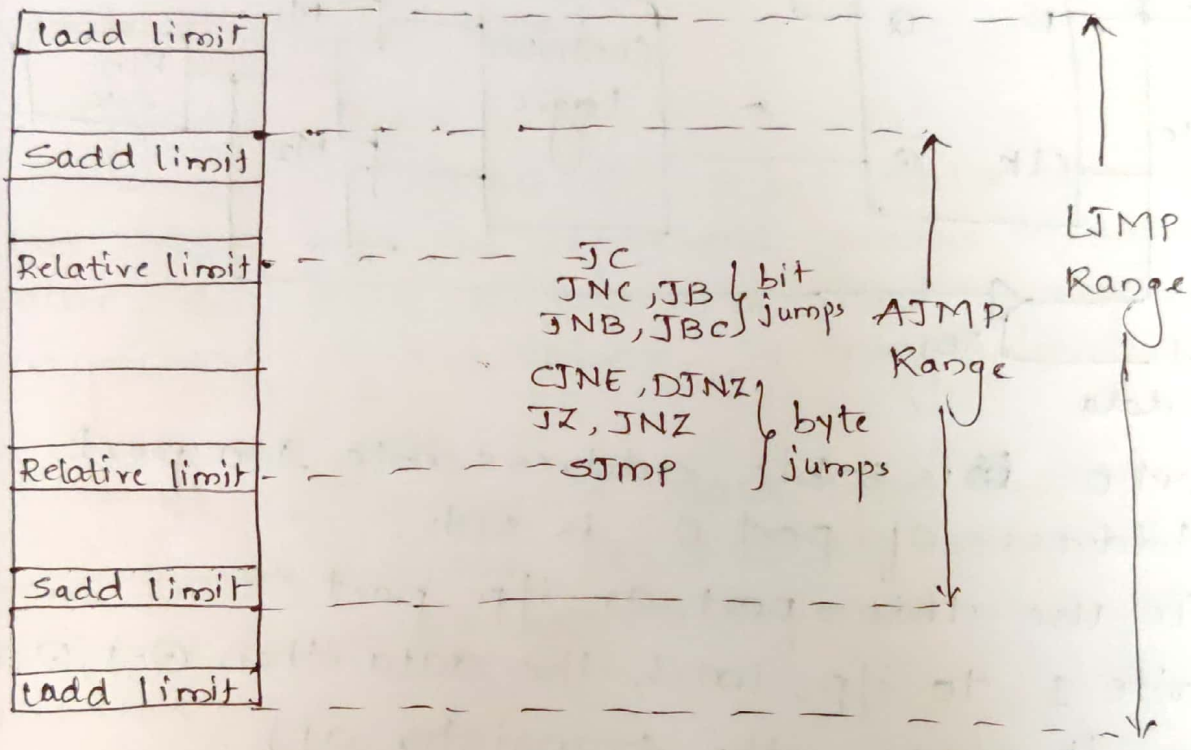
4a) Write an assembly language program to complement the value 55h, 700 times (6M)

```
        ORG 0000h
        MOV A, #55h
        MOV R3, #10
Next:   MOV R2, #70
Again:  CPL A
        DJNZ R2, Again.
        DJNZ R3, Next.
```

4b) Explain Jump & Call Instructions. With neat diagram explain range of JUMP & CALL instructions. (8M)

Jump & Call instructions 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 address from address in program where Jump & Call instruction is called Range of Jump or Call.



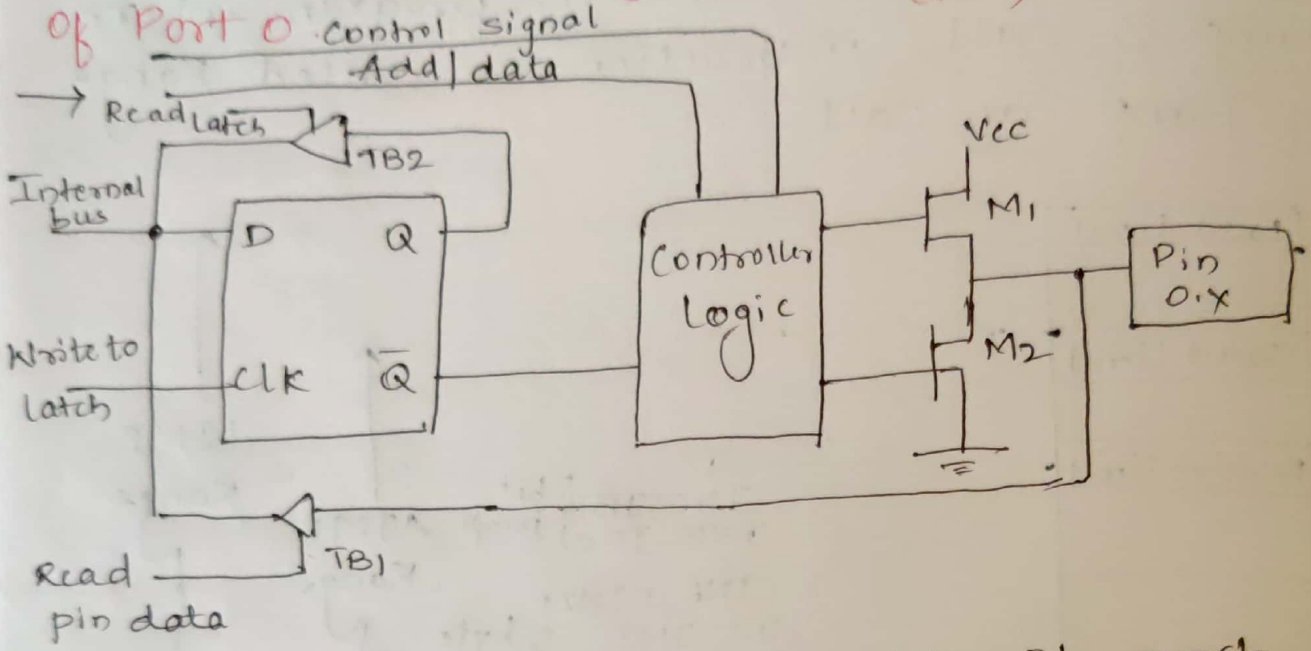
i) Relative Range: The jump can be within -128 bytes to +127 bytes of memory relative to the address of the current program counter. Jump or Call instruction with relative range will be of 2 byte instruction.

ii) Absolute Range:

In 8051, program memory is divided into logical divisions called pages each of 2K byte. Max. size of program memory is 64K bytes, size of each page is 2K bytes.

iii) Long Absolute Range:
This range allows the jump to anywhere in the program location from 0000h to FFFFh.

4C. With neat diagram, explain working of Port 0 (6M)



Port 0 is 8-bit addressable I/O port
Address of port 0 is 80h.

→ To use this port as i/p port →

1) Write 1 to i/p, latch the data then $Q=1, \bar{Q}=0$
Thus, it makes the transistor off.
2) When M_1, M_2 are off it acts like open ckt, there will be no connection b/w port 0 & ground.
3) So directly the i/p signal is directed to tristate buffer.

→ To use this port as o/p port.

1) Write a 0 to internal bus.
2) m_1, m_2 turns ON it acts as sc, thus port pin is connected to ground.
3) Any attempt to read the i/p pin will always get a low ground signal.

Module 3.

5a Explain the different data types supported by 8051 Microcontroller. (8M)

→ Data types.

(i) Unsigned char:

The char data type is of 8-bits. This data type takes a value in range 0-255 (00-FFh). It is the most widely used data type in 8051.

2) Signed char:

It is also an 8-bit data type that uses MSB D7 to represent positive or negative value. So only 7 bits are used to represent magnitude of number. It ranges from -128 to 127.

3) Unsigned int:

It is a 16-bit data type that ranges from 0000h - FFFFh (ie, 0 to 65,535. It is used to define a 16-bit variable such as memory address.

4) Signed int:

It is also an 16-bit data type, that uses MSB D15 to represent positive or negative value. Signed int ranges from -32768 to 32768.

5) sbit (single bit):

It is used to access single bit addressable register. Size is 1 bit ie, either 0 or 1.

6) Bit: Bit data type allows to access single bits of bit addressable memory 20-2Fh.

7) sfr: It is used to access byte size SFR registers. Its size is 8-bits.

5b. Write an 8051 C-program to create a square wave of frequency 2500 Hz on port pin $P_{2.7}$. Use timer 1 in mode 2 to Create the delay $f = 11.0592$ MHz. (8M)

→
```
#include <reg51.h>
void TIM2Delay (void);
sbit mybit = P2^7;
void main (void)
{
  Unsigned char x;
  while (1)
  {
    mybit = ~mybit;
    TIM2Delay();
  }
}
void TiM2Delay (void)
{
  TMOD = 0X20;
  TH1 = -184;
  TR1 = 1;
  while (TF1 == 0);
  TR1 = 0;
  TF1 = 0;
}
```

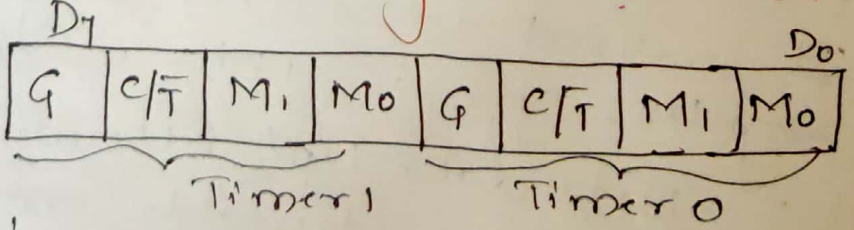
1/2500Hz = 400μs.

400μs/2 = 200μs.

200μs/1.085μs = 184.

5c. Explain TMOD Register? (4M)

→



D7: Gate:

When Gate = 1, timer 1 will run, Only when $\overline{INT_1}$ ($P_{3.3}$) pin is high.

When Gate = 0, timer 1 will run, regardless Of the state of $\overline{INT_1}$ pin.

TR1 in TCON register must be set to 1 for timer 1 to run ie, TR1 = 1.

Bit 6: C/$\bar{T}$.

When C/$\bar{T}$ = 1, counter mode. In counter mode, timer 1 will count event on T1 pin.

When C/$\bar{T}$ = 0, timer 1 will increment every machine cycle.

Bit 5 & 4: M1 & M0

M1	M0	Mode	Operating Mode
0	0	0	13-bit timer
0	1	1	16-bit timer
1	0	2	8-bit Auto reload mode
1	1	3	Split timer mode.

Note: The same for bit D0 - D3 to be written for timer 0, changing prefix 1 to 0.

OR

a. Assume Crystal frequency as 11.0592 MHz. What value do we need to load into the timer register if we want to have a time delay of 5msec. Write an ALP for the same with timer 0 to create a pulse width of 5msec on P2.3. (6M)

XTAL = 11.0592 MHz.

Counter counts up every 1.085 μsec.

∴ 5msec / 1.085μs = 4608 Clock.

∴ TL & TH = 65536 - 4608

= EE00h.

TL = 00H ; TH = EEh.

```
        CLR  P2.3
        MOV  TMOD, #01h
Here:   MOV  TL0, #00h
        MOV  TH0, #0EEh
        SETB P2.3
        SETB TR0
Again:  JNB  TF0, Again
        CLR  P2.3
        CLR  TR0
        CLR  TF0.
```

6b. Write a C-program for Counter 0 in mode 2 to display seconds & minutes On P_1 & P_2. Assume that 60Hz external clock is supplied to T0 pin (P3.4). (6M)

→

```
#include <reg51.h>
void ToTime (unsigned char);
 void main ()
 {
  unsigned char val;
  T0 = 1;
  TMOD = 0X06;
  TH0 = -60;
  while (1){
   {do
      {TR0 = 1;
       Sec = TL0;
       ToTime (val);
      }
    while (TF0 == 0);
    TR0 = 0;
    TF0 = 0;
   }
}
void ToTime (unsigned char val)
 { unsigned char sec, min;
  min = value/60; sec = value % 60; P1 = sec; P2 = m
 }
```

5c. Write a C-program to toggle all bits of P_2 continuously every 500msec. Use Timer 1 in mode 1 to create delay. (8M)

→

```
#include <reg51.h>
void Msdelay (void);
void main (void)
{
 unsigned char x;
 P2 = 0X55;
 while (1)
  {
    P2 = ~P2;
  for (x=0 ; x<20 ; x++)
     Msdelay ();
  }
 }
void Msdelay (void)
 {
   TMOD= 0X10;
   TL1 =0XFE;
   TH1 = 0XA5;
   TR1 = 1;
  while (TF1 ==0);
   TR1 = 0;
   TF1 = 0;
  }
```

A5FEh = 42494 (d)

65536 - 42496 = 23042

23042 X 1.085 µs = 25 ms & 20 X 25 msec

= 500 ms.

Module 4.

7a. Explain the bit status of SCON register (6M)

→

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0 & SM1 → Serial Mode Specifiers.

SM0	SM1	Mode	Description	Baud Rate
0	0	0	8 bit shift register	Baud Rate = fosc/12
0	1	1	8-bit UART	BR. = variable
1	0	2	9-bit UART	fosc/12 or fosc/32
1	1	3	9-bit UART	BR= variable.

SM2: It is used for multiprocessor Communicat
In 8051 we aren't using multiprocessor. So SM2=0

REN: Enables Reception.
REN= 1 (enables reception)
REN=0 (disables reception)

TB8: It refers to the 9th bit, that is to be transmitted in mode 2 & 3.

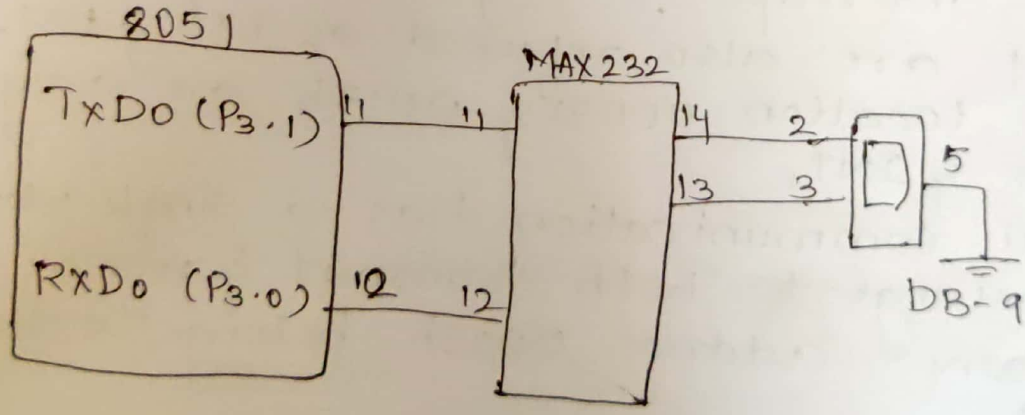
RB8: It refers to the 9th bit, that is to be received by mode 2 & 3.

TI: Transmit Interrupt flag:
Set by hardware, whenever byte is transmitted. If we want to clear TI flag it must be cleared with s/w.

RI: Receive Interrupt flag.
set by h/w whenever byte data is to be received.
If we want to clear RI flag it must be cleared with s/w.

7b. Explain the importance of MAX232 line driver for connecting RS232 to 8051 with neat sketch. Show interface of RS232 to 8051 using MAX 232.

→ Since RS232 is not compatible with today's µP & µC's, we need a line driver (voltage converter) to convert RS232's signal to TTL voltage levels that will be acceptable to 8051's TxD & RxD pin. Such a converter is Max 232. One advantage of MAX232 chip is that it uses a +5V power source which is same as the source voltage for 8051. MAX 232 has two sets of line drivers for transferring & receiving data. TxD & RxD. TxD are called T_1 & T_2 while RxD are designated as R1 & R2. RS-232 uses a 9-pin connector called as DB-9. The most widely used value for these capacitors is 22µF (8M)



7c. Write a C-program for 8051, to transfer the letter 'A' serially, at 4800 baud rate. Use 8-bit data & 1-stop bit (6M)

→
```
#include <reg51.h>
void main (void)
{
    TMOD = 0x20;
```

```
    TH1 = 0XFA;
    SCON = 0X50;
    TR1 = 1;
    while(1)
    {
     SBUF = 'A';
    while(TI==0);
     TI=0;
    }
}
```

OR

8a. Explain different interrupts of 8051 indicating their vector address. (6M)

→ 1) When Reset pin is activated, 8051 jumps to address location 0000h.

2) Two interrupts are set aside for T0 & T1. Memory location 000Bh & 001Bh belongs to T0 & T1 respectively.

3) Two interrupts are set aside for h/w external interrupts INT0 & INT1 respectively. They are also refereed as EX0 & EX1. Memory location 0003h & 0013h are assigned to INT0 & INT1.

4) Serial communication has a single interru that belongs to both transmit & receive. Memory address 0023h belongs to this interrupt.

* Interrupt vector table.

Sl.No	Interrupt	ROM location (hex)
1	Reset	0000h
2	INT0	0003h
3	TF0	000Bh
4	INT1	0013h
5	TF1	001Bh
6	serial port	0023h

3b. Write an 8051 C-program to transfer the message "WELCOME" serially at 9600 baud, 8-bit data, 1 stop bit. Do this continuously. (8M)

→

```
#include <reg51.h>
void serTx (unsigned char);
void main(void)
  {
    TMOD = 0X20;
    TH1 = 0XFD;
    SCON = 0X50;
    TR1 = 1;
    while (1)
    {
      SerTx ('W');
      SerTx ('E');
      SerTx ('L');
      SerTx ('C');
      SerTx ('O');
      SerTx ('M');
      SerTx ('E');
    }
  }
void SerTx (unsigned char x)
  {
    SBUF = x;
    while (TI==0);
    TI=0;
  }.
```

8c. Write an 8051 C- program to receive bytes of data serially & put them in P_1. Set the baud rate at 4800, 8-bit data & 1-stop bit (6M)

→
```
#include <reg51.h>
void main (void)
{
unsigned char mybyte;
TMOD = 0X20;
TH1 = 0XFA;
SCON = 0X50;
TR1 = 1;
while (1)
{
while (RI == 0);
mybyte = SBUF;
P1 = mybyte;
RI = 0;
}
}
```

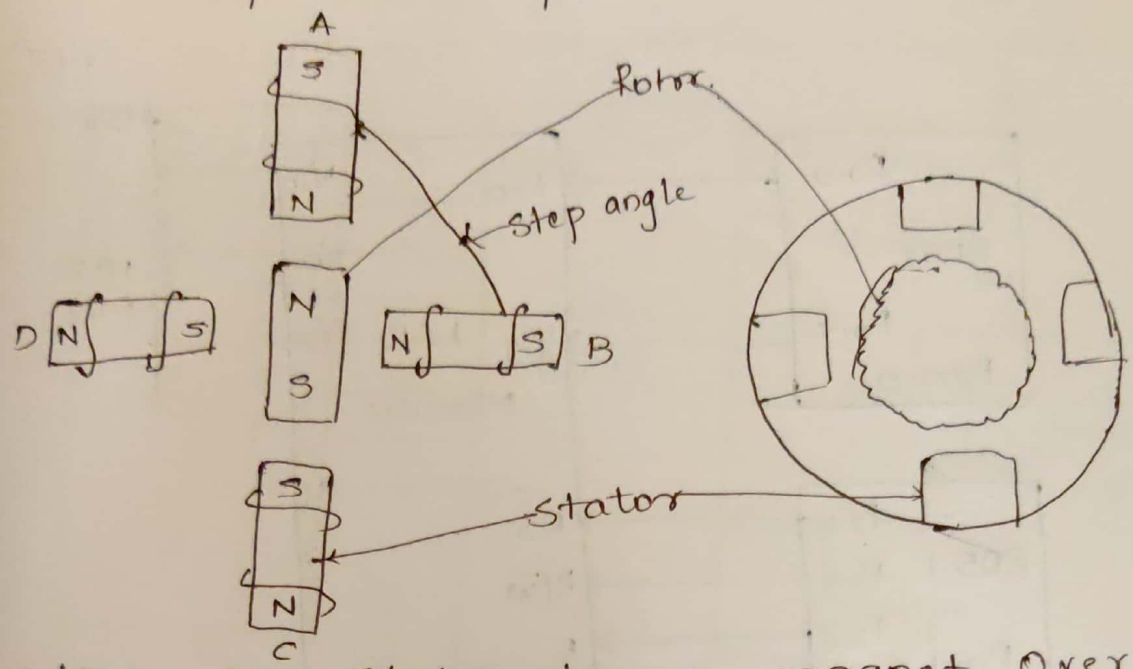
Module 5.

9a. Explain the construction & working of stepper motor. Explain the 4 step sequence, step angle & steps per revolution. (6M)

→ Stepper Motor is widely used device that translates electrical pulses into mechanical movement. Used in applications such as disk drives, dot matrix printers, robotics etc.

It has a permanent magnet rotor which is surrounded by a stator. Commonly used stepper motors have

Such motors are called as four phase or Unipolar stepper motor.



Now, the Stator is a magnet, over which electrical coil is wound. One end of the coil are connected to +5V or Gnd & other end is provided with fixed sequence such that motor runs in a particular direction.

Step angle is defined as minimum degree of rotation with a single step.

No of steps / rev = 360° / step angle.

$$\text{steps per second} = \frac{\text{rpm} \times \text{step/rev}}{60}$$

Ex: If step angle = 2°.

No of steps / rev = 180°.

Two phase ON

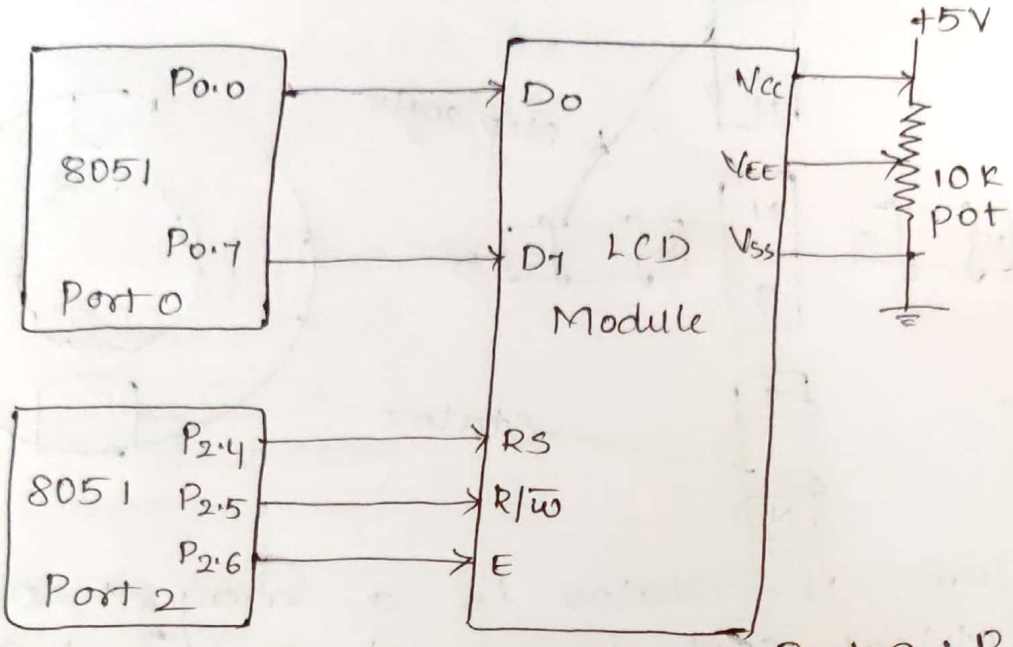
Step	A	B	C	D
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1

One phase ON

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

9b. Explain the architecture & working of 14 pin LCD. Draw the interface diagram of LCD with 8051 Microcontroller. (8M)

→



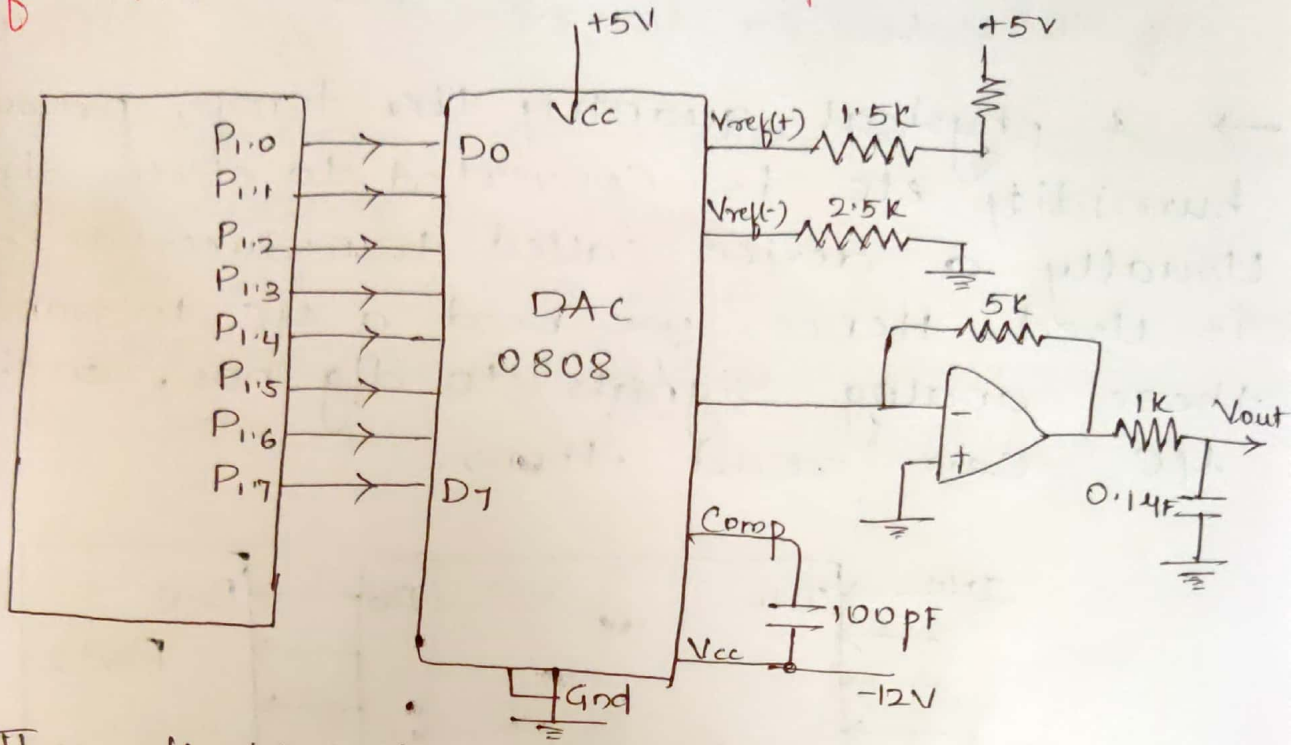
1) LCD is interfaced by using Port 0 & P2.4 to P2.6.
2) When voltage is applied, electrostatic field is created, this allows the light to pass through segment.
3) LCD needs a driving ckt. LCD & driving ckts are integrated in LCD display module.
4) The display has one Command & one Data Register, Data to be displayed is ASCII
5) Data & Command register are differentiated by RS input.
6) Interfacing ckt allows µC to send data to LCD & also to read.

* Pin details:-

1) Vss – Gnd
2) Vcc – +5V
3) VEE – to Control Contrast
4) RS pin → Rs=0 → Command Reg
Rs=1 → Data Reg
5) R/W – i/p pin
=0 for write
=1 for read
6) E → Enable pin.
7) 7 to 14 → I/O pin (8 bit data lines)

9c. With neat diagram explain interfacing of DAC 0808 to 8051 µC. (6M)

→



1) The digital inputs are converted to current Iout.

2) Then by connecting a resistor to Iout pin we convert the result to voltage.

3) The total current Iout is a function of binary nos at D0-D7. i/p's of DAC 0808 & the reference current Iref.

ie, $I_{out} = I_{ref}\left(\frac{D7}{2} + \frac{D6}{4} + \frac{D5}{8} + \frac{D4}{16} + \dots + \frac{D0}{256}\right)$

Usually reference current Iref = 2mA.

4) Ideally we connect o/p pin to resistor, Convert this current to voltage & monitor the o/p on scope.

5) But this can cause inaccuracy, hence an opamp is used to convert the o/p current to voltage.

10 a. Explain the internal architecture of ADC 0804. Explain the interfacing diagram of 8051 µC to ADC 0804. (10M)

→ A physical quantity like temp, pressure humidity etc is converted to electric signals usually a device called transducer or sensor is used. Hence, we need a µC to translate these analog signals to d/g nos, so that µC can read them.

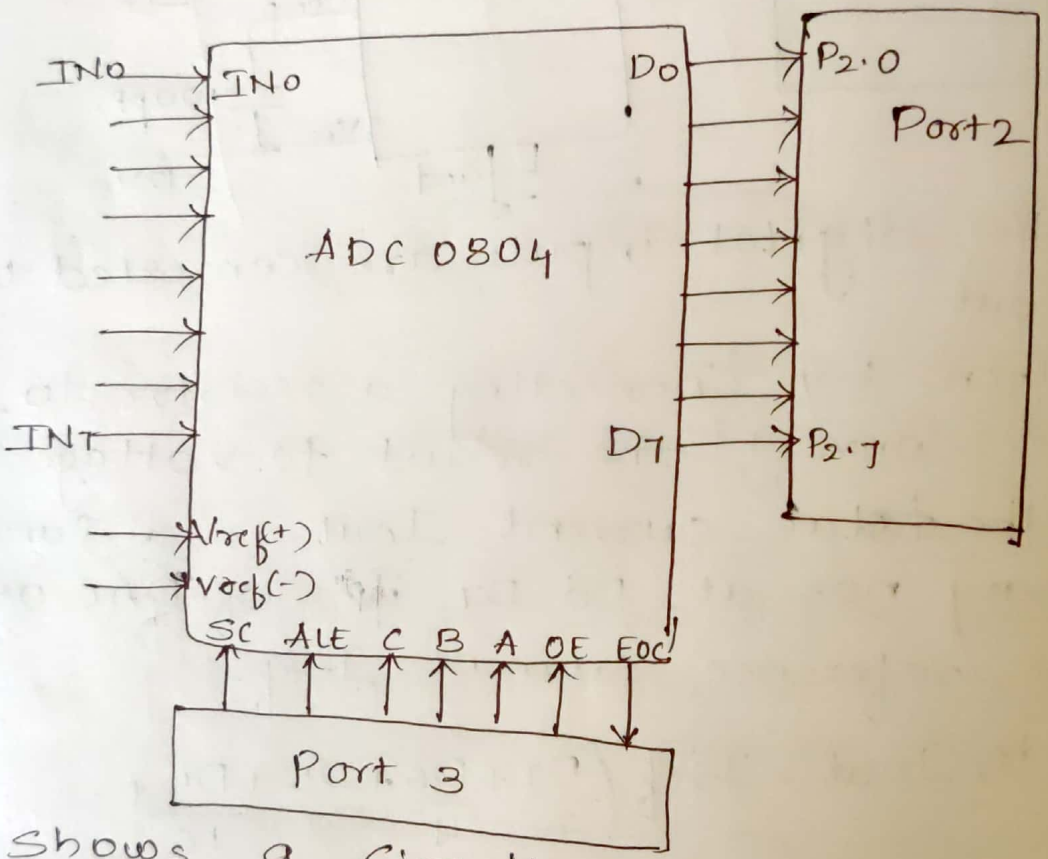


fig shows a Circuit to interface an 8-channel 8-bit A/D converter.

It has 8 i/p lines (IN0 - IN7) & three address lines (A, B & C) to select the i/p. ALE pin is used to latch the address. 8 Data lines are used.

SC → start of conversion.

EOC → End of Conversion.

OE → OE is output enable to perform read operation.

If $V_{ref}(+) = 5V$ & $V_{ref}(-) = 0V$ } then range of A/D is 0-5V

& step size is $\frac{5V}{256} = 19.53mV$.

The 8 o/p lines can assume 8 o/p combinations from 00000000 to 11111111.

for 0V i/p, the o/p of A/D is 00000000 $(00)_h$

for 2.5V i/p, the o/p of A/D is 10000000 $(80)_h$

& for +5V, o/p is 11111111 $(FF)_h$.

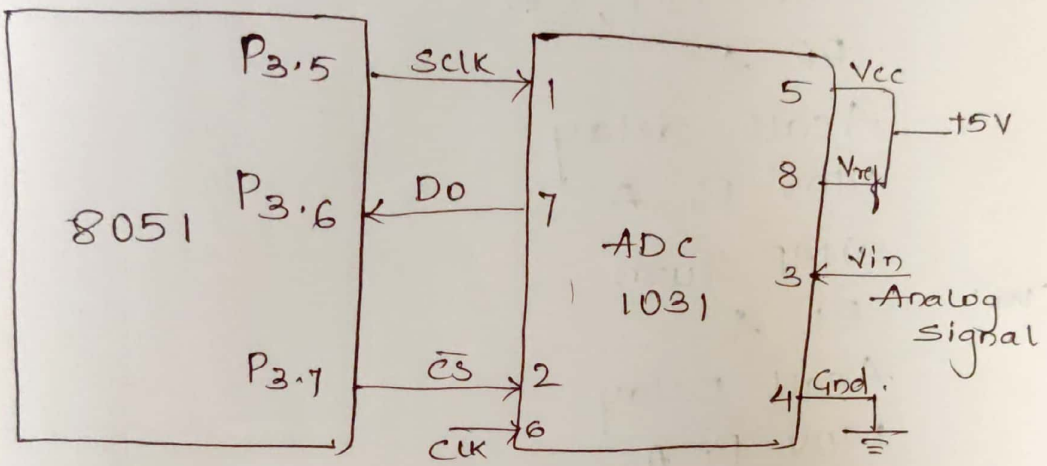


fig shows the ckt to interface ADC1031.

ADC 1031 is a serial ADC chip from national semiconductors & is of 10-bit (D0-D9) serial A/D converter.

MSB is shifted out first through data pin D

Serial clk (SCLK) is used to bring data out

freq applied to CLK pin can be upto 700KHz to 4MHz.

It operates on +5V power supply.

10b. A Switch is connected to pin $P_{2.7}$. Write a program to monitor the status of SW & perform the following. (10M)

(i) If SW=0; Stepper motor moves in CW

(ii) If SW=1; stepper motor moves ACW.

→

```
        ORG 0000h
Main:   SETB P2.7
        MOV A, #66h
        MOV P1, A
Turn:   JNB P2.7, CW
        RR A
        Acall delay
        MOV P1, A
        SJMP Turn
CW:     RL A
        Acall Delay
        MOV P1, A
        SJMP Turn
Delay:
        MOV R2, #100
        MOV R3, #255
H1:     DJNZ R3, H2
H2:     DJNZ R2, H1
        RET
        END
```