24/06/2024

CO 3

# Interfacing

5 Interfaces

(i) 7 Segment display interfacing with 8051 MC

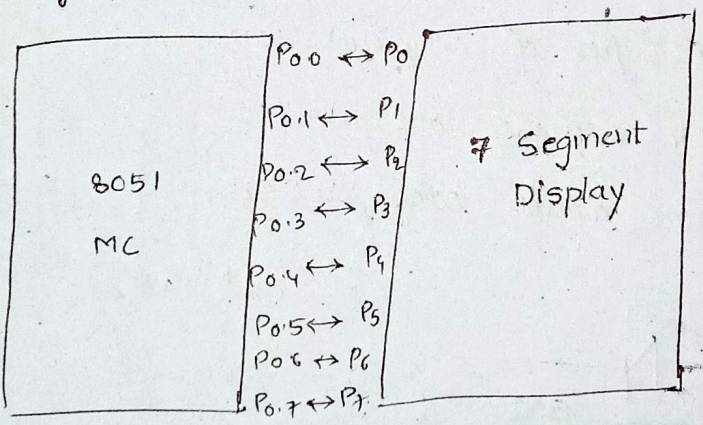
(ii) LCD (2X16) " " " "

(iii) A/D , D/A " " " "
(Analog → Digital) (Digital → Analog)

(iv) Stepper Motor " " " "

(v) Key Board " " " "

⇒ Interfacing is done through ports of 8051.



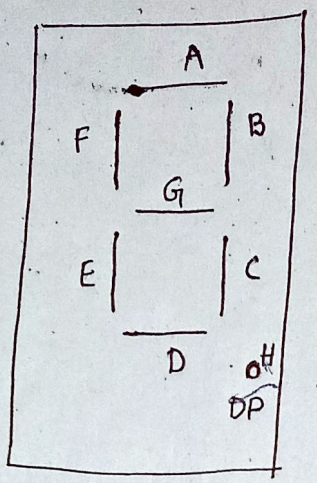

Eg:

```
MOV  A, #05H
MOV  P0, A
ACall Delay
MOV  A, #07H
MOV  P0, A

Delay :
    MOV
```

⇒ 7-Segment display

It is used to display the numbers 0 to 9

It consists of 7 segments and 1 dot point (DP)



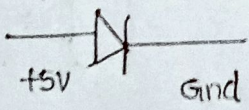

LED Configuration:-

There are 2 types of configuration:-

(i) Common cathode configuration

(ii) Common anode configuration

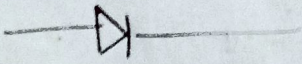
(i)




1 — ON (+5V)

0 — OFF

(ii)



0 — ON

1 — OFF

⇒ Equivalent Hexa Values:-

Common cathode:-

| I/P | H | G | F | E | D | C | B | A | Hexa(H) |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | BFH |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 806 H |
| 2 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | DBH |
| 3 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | CFH |
| 4 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | E6H |
| 5 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | EDH |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | FDH |
| 7 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 87H |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | FFH |
| 9 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | EFH |

⇒ Write an ALP program to display numbers 0-9 using 8051 microcontroller. Assume that LED configuration is cathode configuration.

```
        ORG   C0H
UP:     MOV   R3, #10d / 0AH
        MOV   DPTR, # NUM1
~~GO:~~   CLR   A
GO:     MOVC  A, @ A+DPTR
        MOV   P1, A          → Port 1
        ACall Delay
        INC   DPTR
        DJNZ  R3, GO
        SJMP  UP             → Short ~~jmp~~ jump
        ORG   0200H          → Define Byte
NUM1:   DB    BFH, 86H, DBH, CFH, E6H, EDH, FDH,
              87H, FFH, EFH
        End

Delay:  MOV   R0, #10d
GO:     DJNZ  R0, GO
        Ret
```

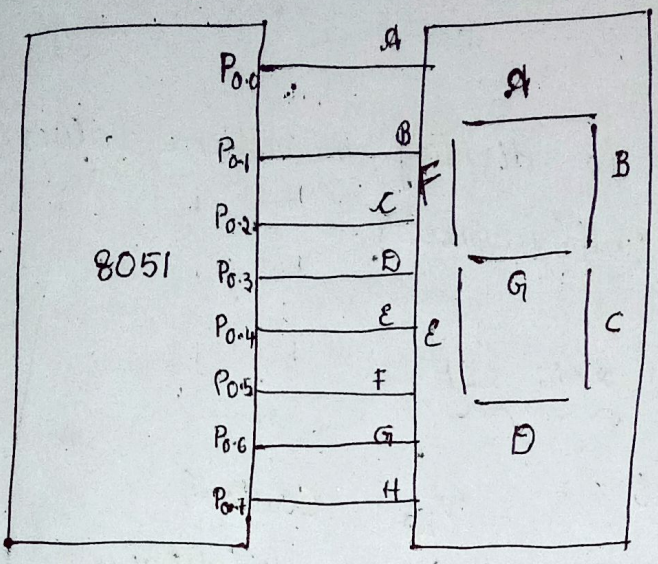
⇒ Write an ALP program to display the numbers 0-9 using 7 segment display using 8051 microcontroller. Assume that LED config. common anode.

```
        ORG   00H
UP:     MOV   R3, #10d / 0AH
        MOV   DPTR, #NUM1
        CLR   A
GO:     MOVC  A, @A+DPTR
        MOV   P1, A
        ACALL DELAY
        INC   DPTR
        DJNZ  R3, GO
        SJMP  UP

        ORG   0200H
NUM1:   DB    40H, 79H, 24H, 30H, 19H, 12H, 02H,
              78H, 00H, 10H

        END

DELAY:  MOV   R0, #10d
GO:     DJNZ  R0, GO
        RET
```

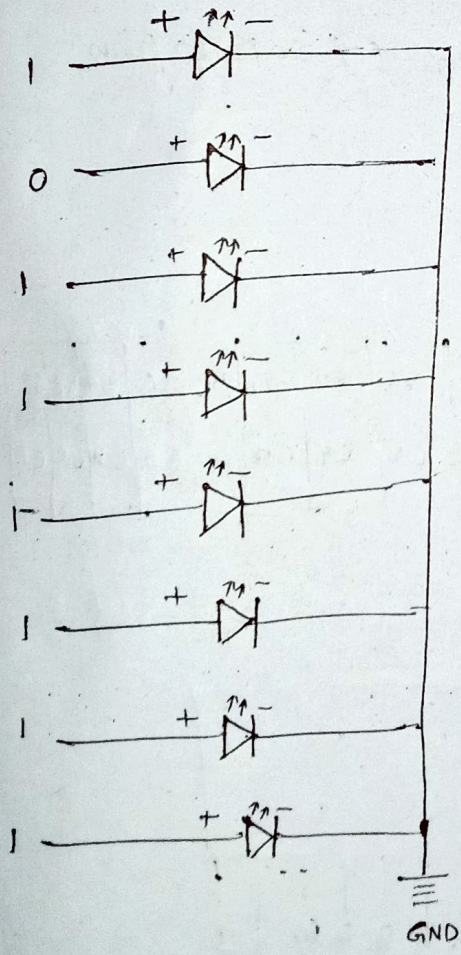
⇒ Common anode :

| I/P | H | G | F | E | D | C | B | A | Hexa(H) |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 40H |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 79H |
| 2 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 24H |
| 3 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 30H |
| 4 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 19H |
| 5 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 12H |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 02H |
| 7 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 78H |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 00H |
| 9 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 10H |

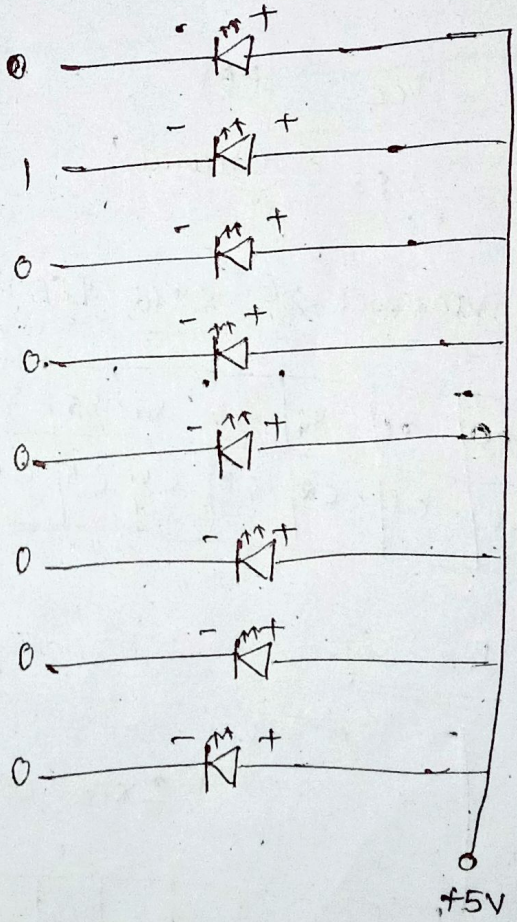
*** ⇒ Draw the interface diagram of 7-segment LED display and 8051 MC.




Common Cathode




Common Anode



***

⇒ Interfacing of 2X16 LCD with 8051 micro-controller:-

LCD is used to display numerical values and alphabets (alphanumeric).

Pin diagram of 2X16 LCD:-

RS - Register Select

R/$\bar{w}$ - Read / write

EN - Enable

$V_{CC}$ - +5V

$V_{SS}$ - Ground

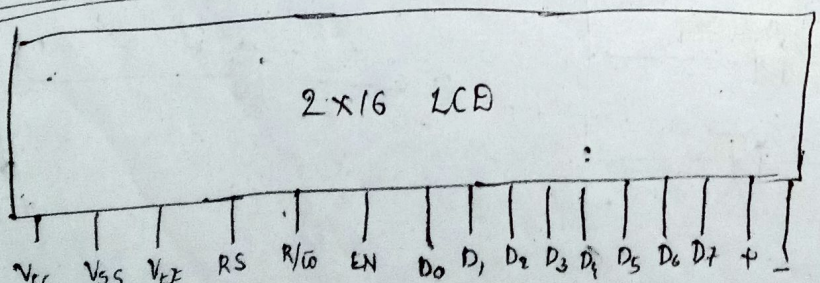
$V_{EE}$ - Control brightness (contrast)

(D0 - D7) → Data pins.

Addresses of 2X16 LCD:-

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 8A | 8B | 8C | 8D | 8E | 8F |
| C0 | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | CA | CB | CC | CD | CE | CF |

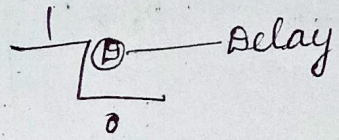
Pin Dig:-



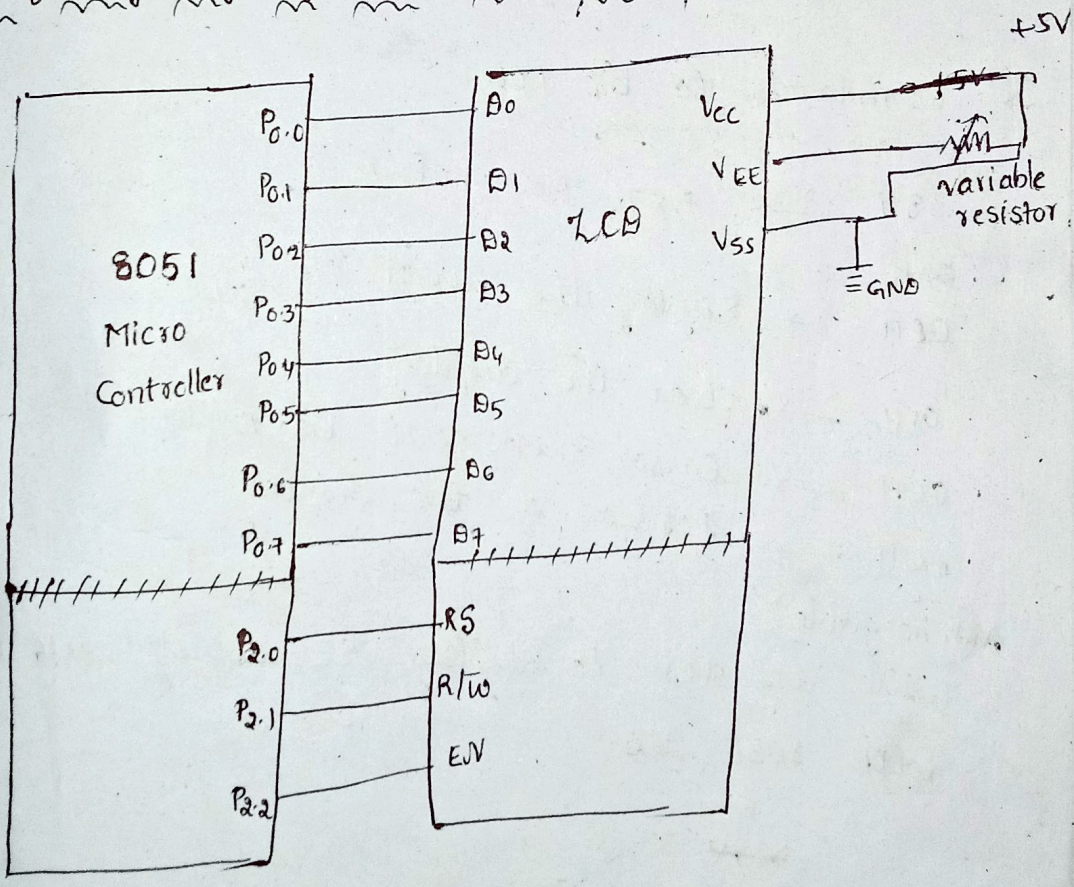
⇒ If Register Select (RS) → 0 → Command
→ 1 → Data

⇒ If R/$\overline{W}$ → 0 → Write
→ 1 → Read

⇒ If EN → 1 → 0 } → (High to low for any operation)
1 → 0




→ Interfacing Pin of 8051 with 2X16 LCD :-



⇒ ~~DATA~~ (Write Operation)

| Command | Data |
|---|---|
| CLR P2.0 | SETB P2.0 |
| CLR P2.1 | CLR P2.1 |
| SET P2.2 | SET P2.2 |
| ACall Delay | ACall Delay |
| CLR P2.2 | CLR P2.2 |

⇒ Write an ALP. ~~program~~

→ Commands to the LCD :-

38H → 5X7 Dot matrix

~~0AH~~

0EH → Display the cursor on

01H → Clear the display

06H → Cursor movement to write

80H → Position of the display

~~Add~~ (changable)

→ Write an ALP to display CSE using 2X16 LCD with 8051 MC.

~~MOV~~

```
ORG   00H
MOV   A, #38H
MOV   P1, A
ACall CMD
ACall Delay
```

```
MOV   A, # 0EH
MOV   P1, A
ACall  CMD
 ACall  Delay
 MOV  A, # 01H
 MOV  P1, A
 ACall  CMD
 ACall  Delay
 MOV   A, # 06H
 MOV   P1, A
 ACall  CMD
 ACall  Delay
 MOV   A, # 8&H
  MOV  P1, A
  ACall  CMD
  Acall  Delay
  MOV   A, # 'C'
  MOV   P1, A
   ACall  DATA
   ACall  Delay
 MOV A, #'S'
 MOV  P1, A
 ACall  DATA
 ACall  Delay
 MOV  A, #'E'
  MOV  P1, A
  ACall  DATA
  ACall  Delay
  End
```

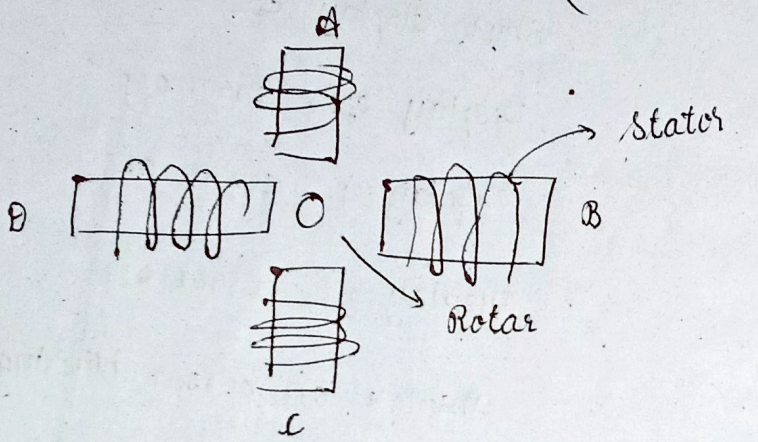
```
CMD:
    CLR   P2.0
    CLR   P2.1
    SET   P2.2
    ACall  Delay
    CLR   P2.2.
    Ret

DATA:
    SETB  P2.0
    CLR   P2.1
    SET   P2.2
    ACall  Delay
    CLR   P2.2
    Ret

Delay:  MOV  R0, #10d
GO:  DJNZ  R0, GO
     Ret
```

| Code (Hex) | Command to LCD Instruction Register |
|---|---|
| 1 | Clear display screen |
| 2 | Return home |
| 4 | Decrement cursor (shift cursor to left) |
| 6 | Increment cursor (shift cursor to right) |
| 5 | Shift display right |
| 7 | Shift display left |
| 8 | Display off, cursor off |
| A | Display off, cursor on |
| C | Display on, cursor off |
| E | Display on, cursor blinking |
| F | Display on, cursor blinking |
| 10 | Shift cursor position to left |
| 14 | Shift cursor position to right |
| 18 | Shift the entire display to left |
| 1C | Shift the entire display to right |
| 80 | Force cursor to beginning to 1st line |
| C0 | Force cursor to beginning to 2nd line |
| 38 | 2 lines and 5×7 matrix |

⇒ Interfacing of ~~8051~~ stepper motor with 8051 Microcontroller :-

→ Stepper motor converts electrical signal into mechanical work

→ It consists of (i) Magnet (Rotar)
(ii) Stator (winding)




⇒ Applications: Robotics

→ The stepper motor can be driven in 3 ways:

(i) Wave Drive mode
(ii) Normal 4 step sequence
(iii) Half 8-step sequence.

(i) Wave Drive Mode:

| Step | A | B | C | D | Equivalent value (Hex) |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 8 |
| 2 | 0 | 1 | 0 | 0 | 4 |
| 3 | 0 | 0 | 1 | 0 | 2 |
| 4 | 0 | 0 | 0 | 1 | 1 |

(ii) Normal 4 step sequence :-

| Step | A | B | C | D | Eq. value (hexa) |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | C |
| 2 | 0 | 1 | 1 | 0 | 6 |
| 3 | 0 | 0 | 1 | 1 | 3 |
| 4 | 1 | 0 | 0 | 1 | 9 |

(iii) Half 8 step sequence :-

| Step | A | B | C | D | Eq. value (hexa) |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 8 |
| 2 | 1 | 1 | 0 | 0 | C |
| 3 | 0 | 1 | 0 | 0 | 4 |
| 4 | 0 | 1 | 1 | 0 | 6 |
| 5 | 0 | 0 | 1 | 0 | 2 |
| 6 | 0 | 0 | 1 | 1 | 3 |
| 7 | 0 | 0 | 0 | 1 | 1 |
| 8 | 1 | 0 | 0 | 1 | 9 |

⇒ Write an ALP to drive the stepper motor clockwise using wave drive mode.

```
        ORG  00H
GO:     MOV  A, #88H
        MOV  P1, A
        ACall Delay
        MOV  A, #44H
        MOV  P1, A
        ACall Delay
        MOV  A, #22H
        MOV  P1, A
        ACall Delay
        MOV  A, #11H
        MOV  P1, A
        ACall Delay
        SJMP GO
        End

DELAY:  MOV  R0, #10d
~~GO:  DJNZ~~
UP:     MOV  R1, #10d
GO:     DJNZ R1, GO
        DJNZ R0, UP
        Ret
```

∵ 8, 4, 2, 1 are inputs but they can't be represented in bytes. So padding is done

8 → 88H

4 → 44H

2 → 22H

1 → 11H

⇒ Write an ALP to drive the stepper motor clockwise using normal 4 step sequence

```
        ORG 00H
GO:     MOV A, #CCH
        MOV P1, A
        ACall Delay
        MOV A, #66H
        MOV P1, A
        ACall Delay
        MOV A, #33H
        MOV P1, A
        ACall Delay
        MOV A, #99H
        MOV P1, A
        ACall Delay
        SJMP GO
        End

Delay:  MOV R0, #10d
UP:     MOV R1, #10d
GO:     DJNZ R1, GO
        DJNZ R0, UP
        Ret
```

C → CC

6 → 66

3 → 33

9 → 99

⇒ Write an ALP to drive the stepper motor anti clockwise using wave drive mode

```
        ORG  00H
GO:     MOV  A, #88H
        MOV  P1, A
        ACall Delay
        MOV  A, #CCH
        MOV  P1, A
        ACall Delay
        MOV  A, #44H
        MOV  P1, A
        ACall Delay
        MOV  A, #88H
        MOV  P1, A
        ACall Delay
        SJMP GO
        End
Delay:  MOV  R0, #10d
UP:     MOV  R1, #10d
GO:     DJNZ R1, GO
        DJNZ R0, UP
        Ret
```

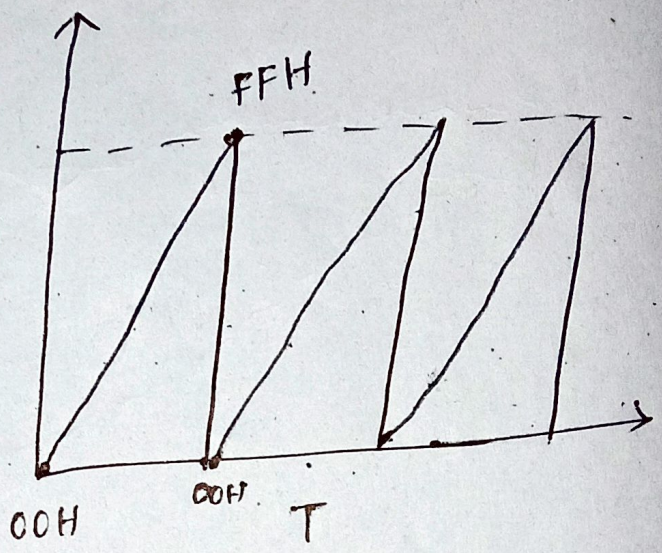
ACall

⇒ write an ALP to generate Saw Tooth wave form using DAC convertor. with 8051 MC.

↓

Digital Analog Converter

```
        ORG   00H
        MOV   A, #00H
Back:   MOV   P1, A
        ACall Delay
        INC   A
        SJMP  Back

Delay:  MOV   R0, #10d
UP:     MOV   R1, #10d
GO:     DJNZ  R1, GO
        DJNZ  R0, UP
        Ret
        End
```

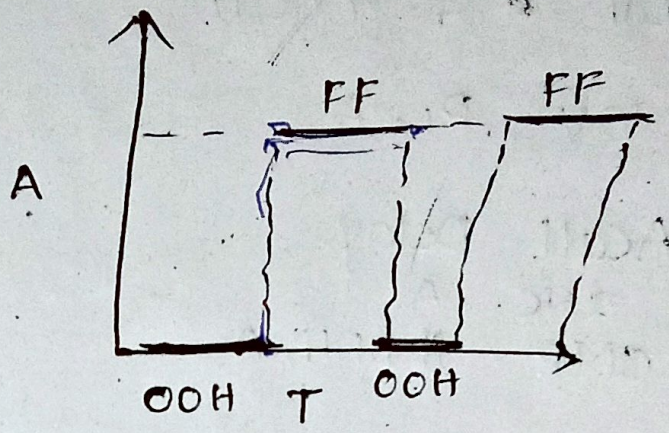



[ A → Amplitude
T → Time Period ]

1111 1111
1
CF = 1
0000 0000
1+FF+1 = 00H

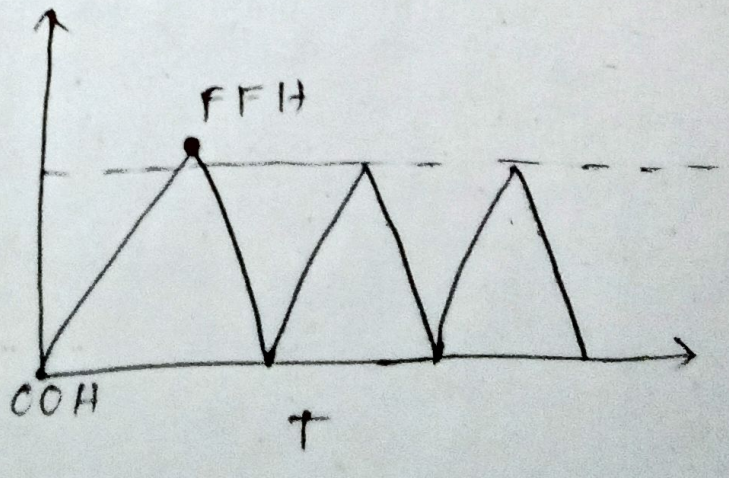
⇒ write an ALP to generate square wave using DAC with 8051 MC

```
    ORG  00H
    MOV  #00H
Back: MOV  P1, A
      ACall Delay
      CPL  A
      SJMP Back

Delay: MOV  R0, #10d
   UP: MOV  R1, #10d
   GO: ~~MOV~~ DJNZ ~~GO~~ R1, GO
       DJNZ  R0, UP
       Ret
       End
```



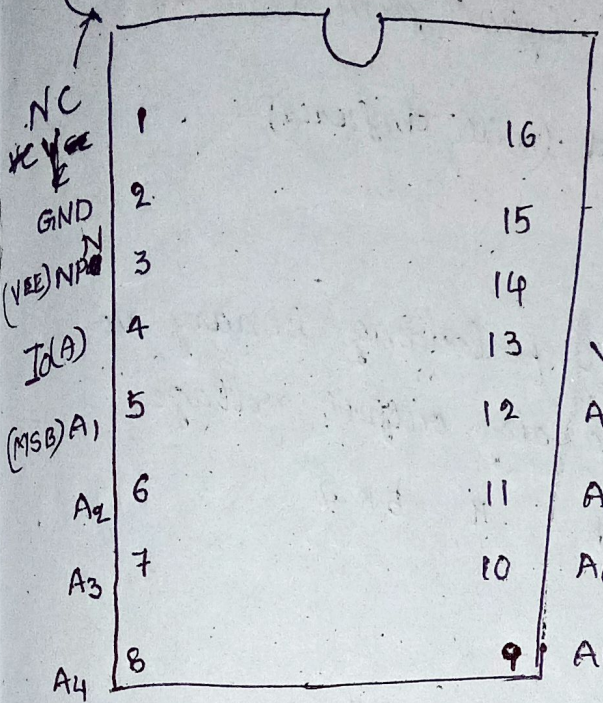

write an ALP to generate a triangular wave using DAC with 8051 MC

```
    ORG  00H
    MOV  A, #00H
BACK: MOV  P1, A
    ACall Delay
```



```
        ORG  00H
UP1:    MOV  A, #00H
BACK:   MOV  P1, A
        ACall Delay
        INC  A
        CMP  #FFH, A
        JNE  BACK
BACK1:  DEC  A
        MOV  P1, A
        ACall Delay
        Dec A
        CMP  #C0H, A
        JNE  BACK1
        SJMP UP1
        Ret
        end

Delay:  MOV  R0, #10d
UP GO:  MOV  R1, #10d
GO:     DJNZ R1, GO
        DJNZ R0, UP
        Ret
```

⇒ Draw the pin diagram of Digital to analog convertor (0808).



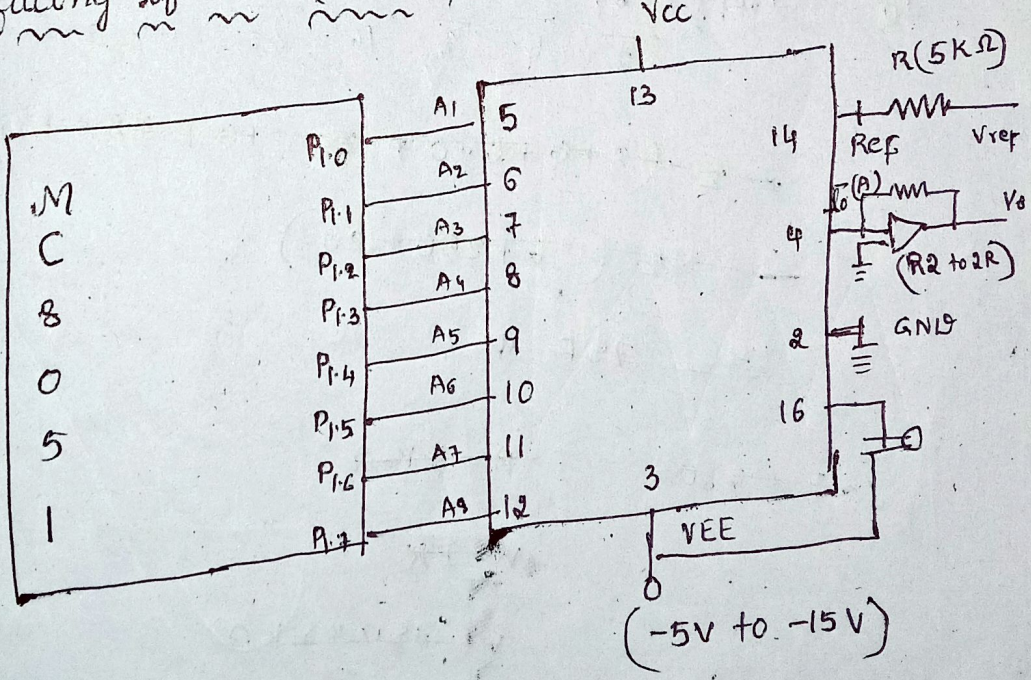

[Vref = 10V]

[Io → Current Output/ Analog output]

(NPN → If you want to give negative voltage -5V to -15V)

: DAC (0808)

⇒ Interfacing of DAC (0808) with 8051 MC :-



⇒ DAC (0808) is a 8-bit DAC.

⇒ It consists of 16 pins

→ $(A_1 - A_8)$ are digital input pins. (5-12)

→ $I_{ref}$ is generally 2 mA (milli Amperes)

⇒ ~~write an ALP~~ For the following binary no, apply DAC, calculate analog output voltage obtained from $I_{ref} = 2mA$, $R = 5k\Omega$.

(a) 1100 0011

(b) ~~0011~~
0001 0111

$$I_0 = I_{ref}\left(\frac{D_7}{2} + \frac{D_6}{4} + \frac{D_5}{8} + \frac{D_4}{16} + \frac{D_3}{32} + \frac{D_2}{64} + \frac{D_1}{128} + \frac{D_0}{256}\right)$$

$$= 2\left(\frac{2+4+0+0+32+0+0+128+256}{}\right)$$

$$= 2\left(6+256+128\right)$$

$$= 780\ A$$

$I = 780A$ $\quad R = 5K\Omega$

$V = IR$

$V = 780 \times 5K\Omega$

$$= \frac{1}{2}\left(\frac{128+64+2+1}{128}\right) = \frac{130+64+1}{128}$$

$$= \frac{195}{128} = 1.52$$

256 | 6
128 | 2
64 | 2
32 | 2
16 | 2
8 | 2
4 | 2
2 | 2
1 | 0

262
128
390
390
780
15
3900
256
125
126
256

$I = 1.5\,mA \quad R = 5k\Omega$

$V = IR = 1.52 \times 10^{-3} \times 10^{3} \times 5$

$= \cancel{7.6\,V}\ 7.6\,V$

$152 \times 5 = 760$

(b) $I_0 = I_{ref}\left(\frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{64}\right)$

$32)19(0.$
$0$
$190$

$= 2\left(\frac{8+4+2+1}{64}\right)$

$= \cancel{2}^{1}\left(\frac{19}{\cancel{64}\,32}\right) = \frac{19}{32}\,mA$

⇒ Find the voltage for full scale. DAC.

$$I_0 = I_{ref}\left(\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\frac{1}{16}+\frac{1}{32}+\frac{1}{64}+\frac{1}{128}+\frac{1}{256}\right)$$

$$= 2\left(\frac{128+64+32+16+8+4+2+1}{256}\right)$$

$$= 2\left(\frac{128+15+80+32+128}{128}\right)$$

$$= \frac{160+80+15}{128} = \frac{235}{128}\ mA = 1.99\ mA$$

128) 235 (1
128

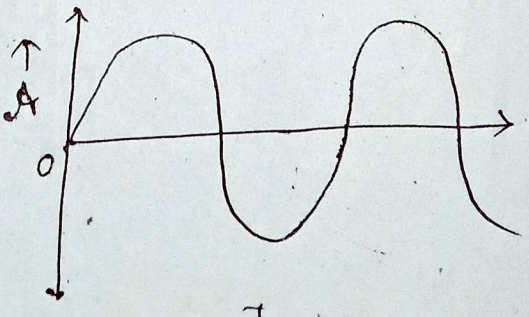
$$I_0 = \frac{235}{128}\ mA \qquad R = 5\,k\Omega$$

$$V = IR = \frac{235}{128} \times 5 \times 10^{3}$$

$$= 1.99 \times 5 \times 10^{3} \times 10^{-3}$$

$$= 9.5\ V$$

⇒ Write an ALP to generate sine wave using DAC with 8051 MC.



| θ | sinθ | $V_{Mag} = 5V + (5V \times \sin\theta)$ | DAC $V_{av} = V_{Mag} \times 25.6$ |
|---|---|---|---|
| 0 | 0 | 5 V | 128 |
| 30 | 0.5 | 7.5 V | 192 |
| 60 | 0.8 | 9 V | 230 |
| 90 | 1 | 10 V | 256 |
| 120 | 0.8 | 9 V | 230 |
| 150 | 0.5 | 7.5 V | 192 |
| 180 | 0 | 5V | 128 |
| 210 | -0.5 | 2.5 V | 64 |
| 240 | -0.8 | 1 V | 25 |
| 270 | -1 | 0 V | 00 |
| 300 | -0.8 | 1 V | 25 |
| 330 | -0.5 | 2.5V | 64 |
| 360 | 0 | 5V | 128 |

Program:

```
        ORG   00H
UP:     MOV   R3, # 13d
        MOV   DPTR, # NUM1
        CLR   A
GO:     MOVC  A, @A + DPTR
        MOV   P1, A
        ACall Delay
        INC   DPTR
        DJNZ  R3, GO
        SJMP  UP
        ORG   0200H
NUM1:   DB    128d, 192d, 230d, ---128d
        end
```

⇒ ADC (Analog to digital convertor) :-

→ It is an 8 bit ADC

→ Used to convert analog signals into digital signals.

→ Used at the communication to convert analog signals into digital signals.

⇒ Pin Diagram of ADC :-

It consists of 20 pins

8 → Digital pins

1 → Analog GND

~~1 → GND~~

1 → Digital GND

1 → $V_{cc}$

1 → CLLROR

$V_{IN}$ → +ve

$V_{IN}$ → -ve

1 → $V_{ref}$

1 → CLKIN

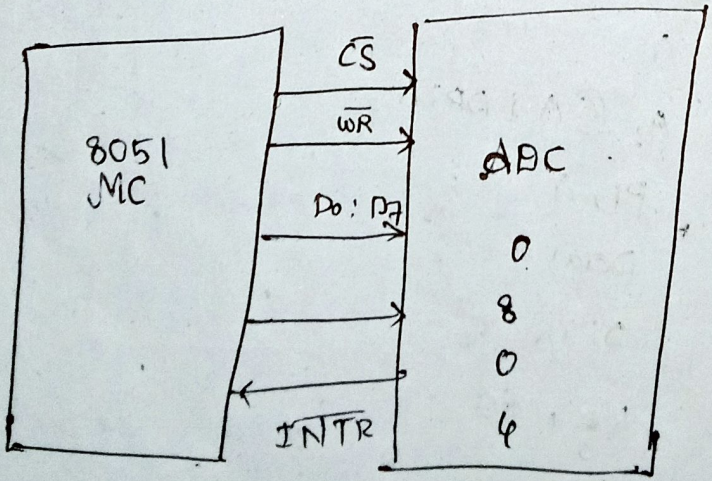
$\overline{CS}$ ⟶ Chip Select

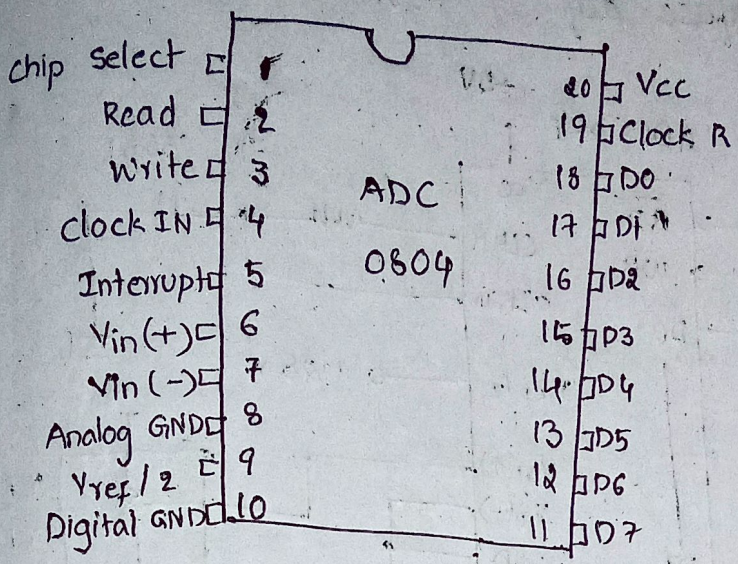
$\overline{RD}$ ⟶ Read

$\overline{WR}$ ⟶ write

$\overline{INTR}$ ⟶ Interrupt request

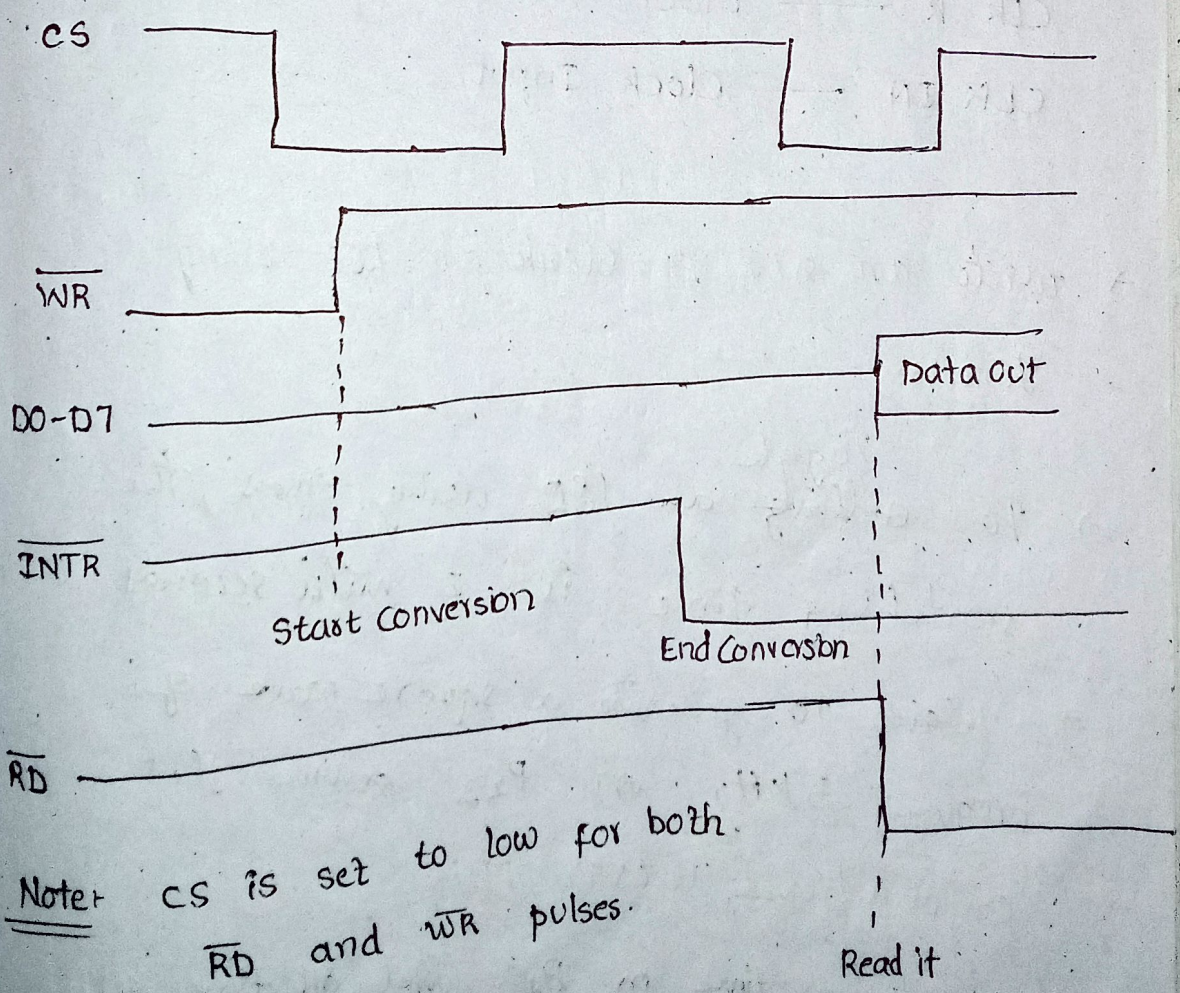
} Selection pins





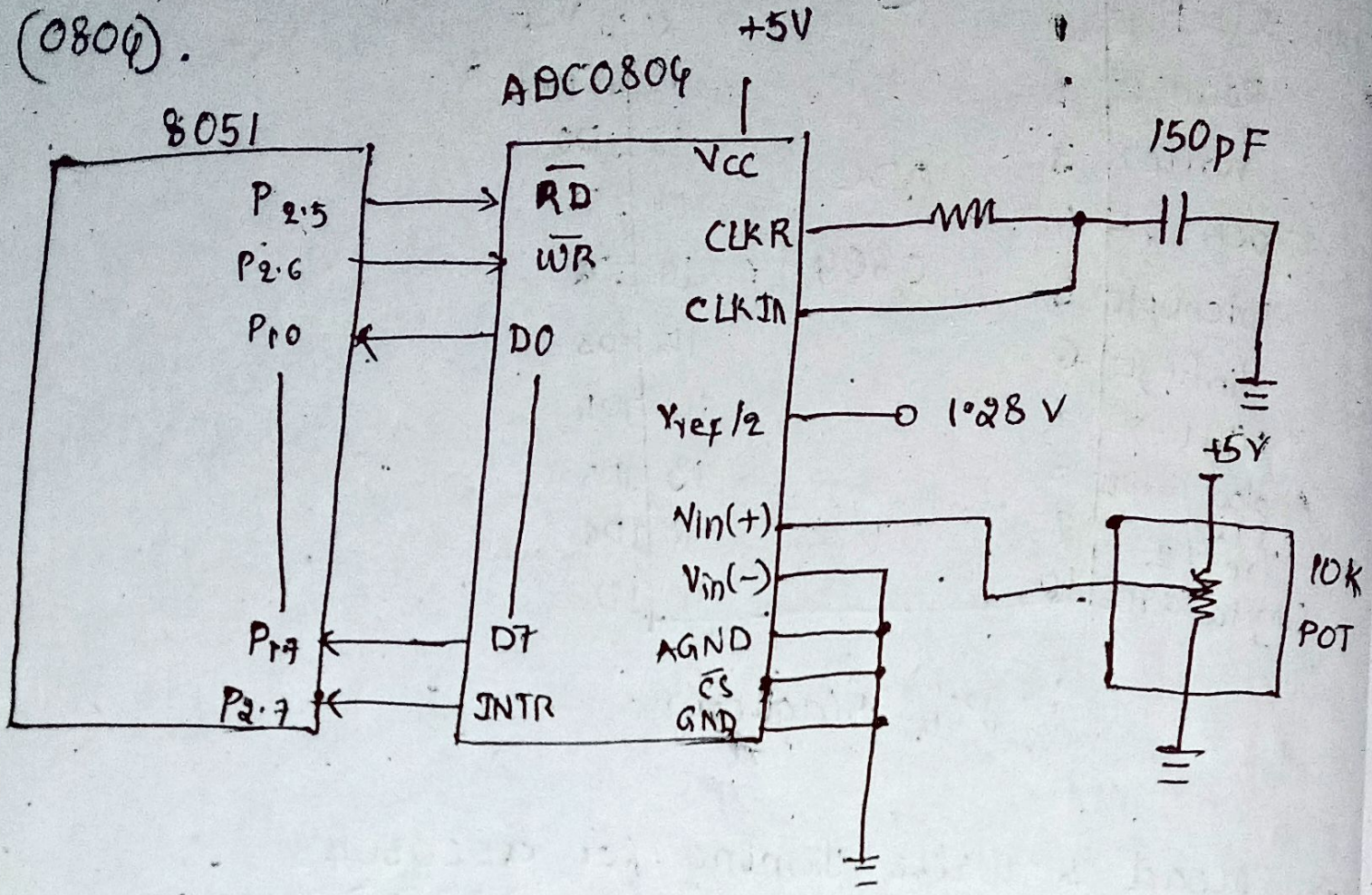

: Pin Diagram

⇒ Read & write Timing for ADC0804 :-




Note: CS is set to low for both $\overline{RD}$ and $\overline{WR}$ pulses.

⇒ Draw the interface dig. of 8051 with ADC (0804).




CLK R —— Clock Reference

CLK IN —— Clock Input