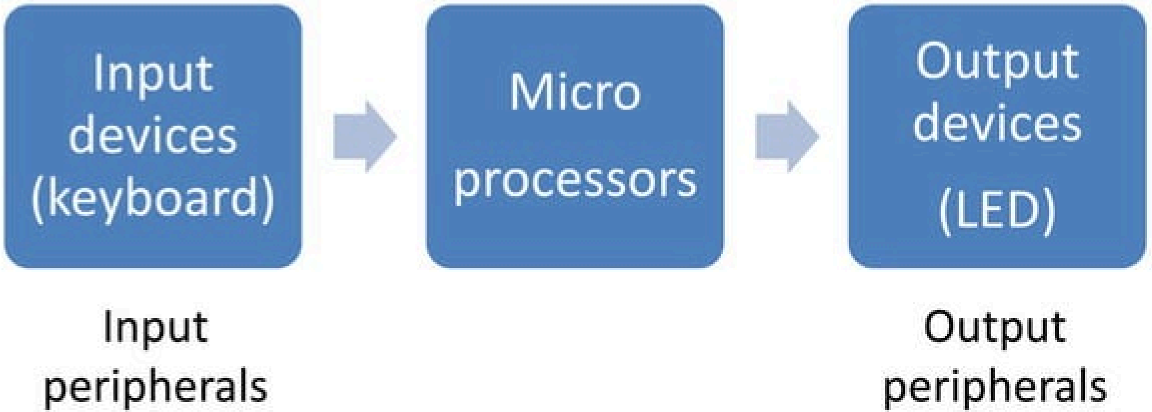
Introduction

- To communicate with the outside world, microprocessor use peripherals (I/O devices)
- Input devices – Keyboards, A/D converters etc.,
- Output devices – CRT, Printers, LEDs etc.,
- Peripherals are connected to the microprocessors through electronic circuit known as interfacing circuits.

Microprocessors unit with I/O devices

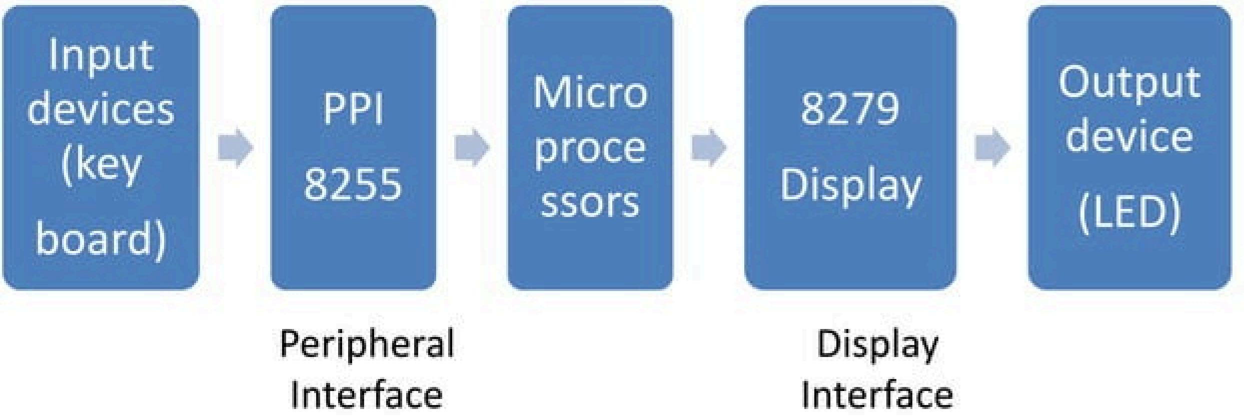


- Some of the general purpose interfacing devices
  - I/O ports
  - Programmable peripherals interface (PPI)
  - DMA controllers(Direct memory access)
  - Interrupt controller
- Some of the special purpose interfacing devices
  - CRT controller
  - Keyboard
  - Display
  - Floppy Disc controllers

Some peripheral interfacing chips of 8085 and 8086 microprocessors.

- Programmable peripherals interface Inter 8255 (PPI)
- Programmable Interrupt controller (PIC) Intel 8259
- Programmable communication interface (PCI) Intel 8251
- Keyboard display Controller Intel 8279
- Programmable counter /Inverter timer Intel 8253
- A/D and D/A Converter Interfacing

Microprocessors unit with I/O devices



Address Space Partitioning

- Two schemes for the allocation of addresses to memories and I/O devices
  - Memory mapped I/O
  - I/O mapped I/O

Memory mapped I/O

- It has only one address space
- Address space is defined as the set of all possible addresses that a microprocessor can generate
- Some addresses assigned to memories and Some addresses to I/O devices
- Memory locations are assigned with addresses from 8000 to 84FF
- It is suitable for small system.

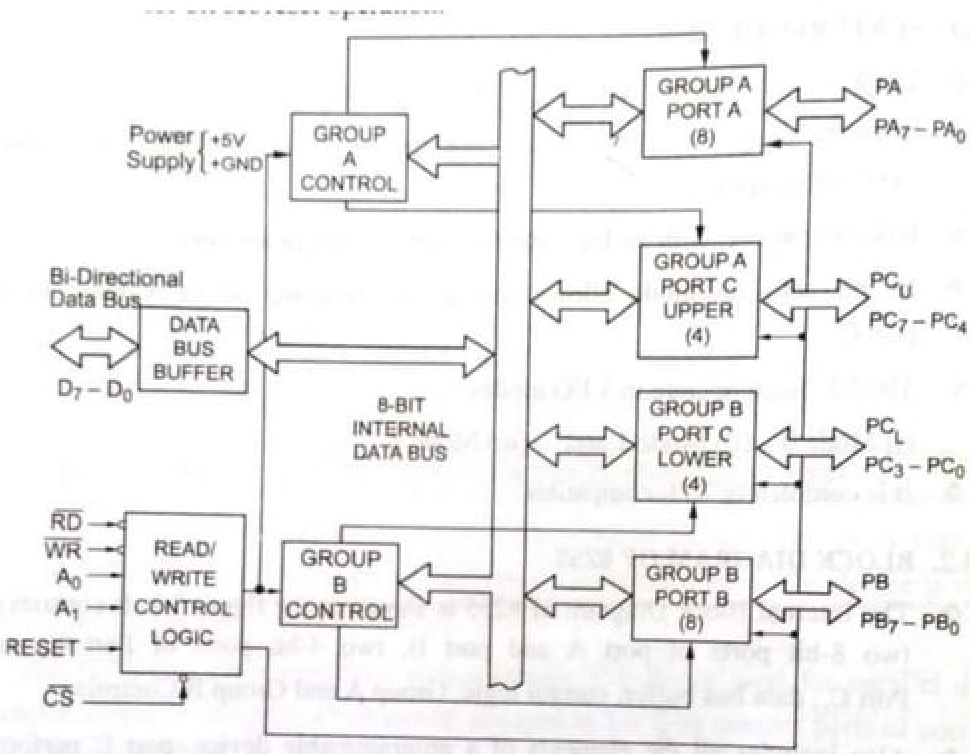
I/O mapped I/O scheme

- In this scheme, addresses assigned to memories locations can also be assigned to I/O devices
- When the signal is high, then address on the address bus is for an I/O devices
- When the signal is low, then address on the address bus is for memory locations.

I/O mapped I/O scheme

- Two extra instruction IN and OUT are used to address I/O devices.
- The IN instruction is used to read the data of an input devices.
- The OUT instruction is used to send the data of an input devices.
- This scheme is suitable for a large system.

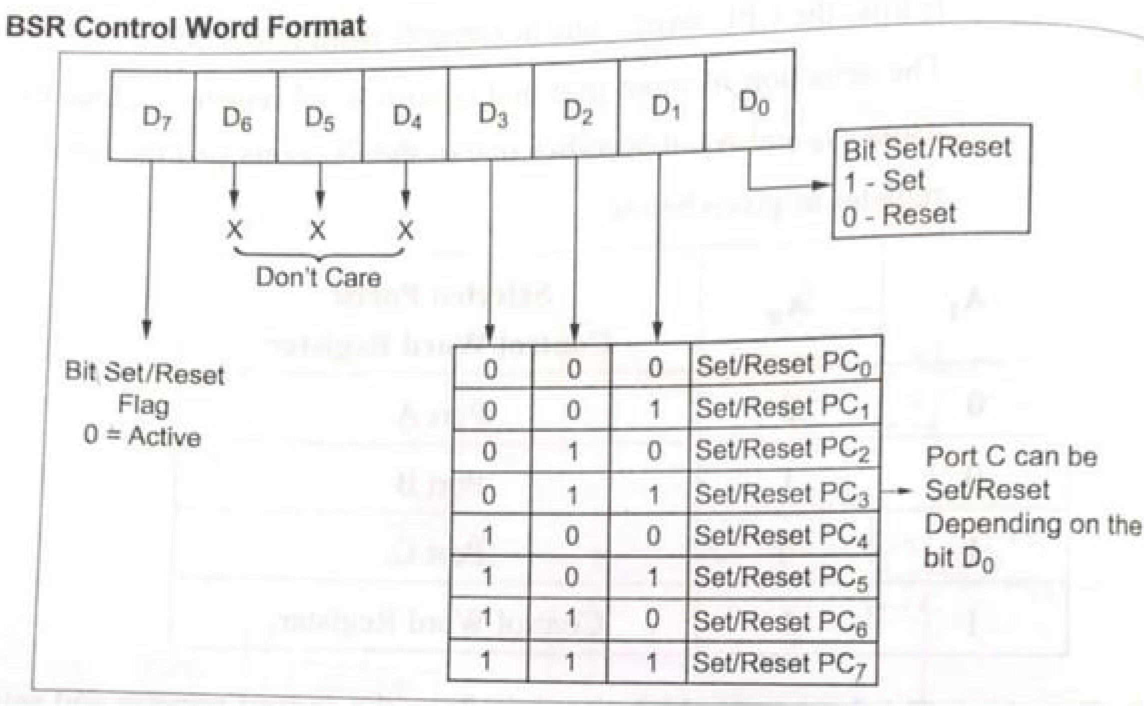
PROGRAMMABLE PERIPHERALS INTERFACE INTER 8255 (PPI)



Operating mode of 8255

- Bit Set Reset (BSR) Mode
- I/O Mode

Bit Set Reset (BSR) Mode



BSR control word format

I/O Mode

- The 8255 has the following 3 modes of operation
  - Mode 0 – Simple Input/output
  - Mode 1 – Input / Output with the Handshake or strobed
  - Mode 2 – Bi-directional I/O

I/O Mode

Mode 0 – Simple Input/output

- Port A and port B are used as two simple 8-bit I/O port
- Port C as two 4-bit port

- Features
  - Outputs are latched
  - Inputs are buffered not latched
  - Ports do not have handshake or interrupt capability

I/O Mode

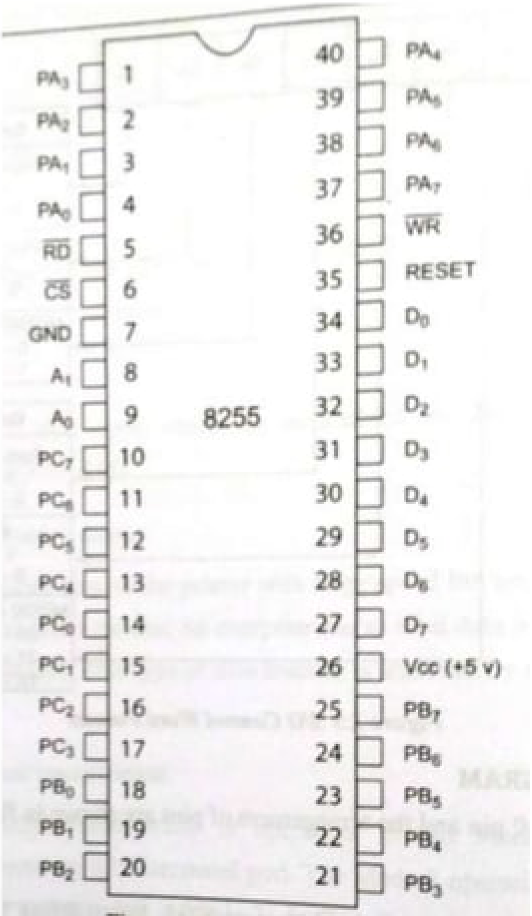
- Mode 1 – Input / Output with the Handshake
  - Input or output data transfer is controlled by handshaking signals.
  - Handshaking signals are used to transfer data between devices whose data transfer speeds are not same.
  - Port A and Port B are designed to operate with the Port C.
  - When Port A and Port B are programmed in Mode 1, 6 pins of port C is used for their control.

I/O Mode

- D0-D7 data bus
  - bi directional, tri state data bus line
  - It is used to transfer data and control word from 8085 to 8255
- RD (Read)
  - When this pin is low, the CPU can read data in the port or status word through the data buffer
- WR (write)
  - When this pin is low, the CPU can write data in the port or in the control register through the data buffer

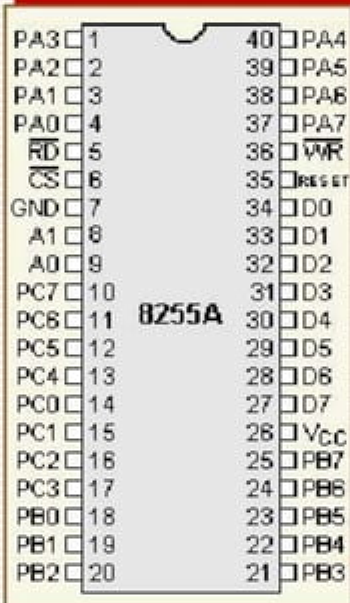
I/O Mode

- Mode 2 – Bi-directional I/O
- Port A can be programmed to operate as a bidirectional port.
- The mode 2 operation is only for port A
- When port A is programmed in Mode 2, the Port B can be used in either Mode 1 or Mode 0.
- Mode 2 operation the port a is controlled by PC_3 to PC_7 of port C.



PIN DIAGRAM OF 8255

Pin diagram of 8255



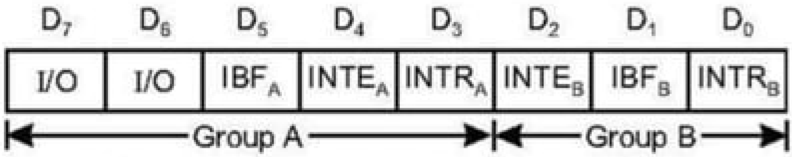
D7 – D0	**Data Bus**
PA7 – PA0	**Port A**
PB7 – PB0	**Port B**
PC7 – PC0	**Port C**
CS	**Chip Select**
A0, A1	**Address bits**
RD	**Read Input**
WR	**Write Input**
RESET	**Reset Input**
Vcc	**+5V**
GND	**0 Volts**

PROGRAMMING and OPERATION of 8255

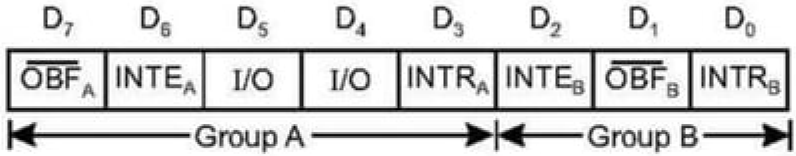
- Programming in MODE 0
- D7 –set to 1
- D6,D5,D2- all set to 0 –MODE 0
- D4,D3,D1 and D0- determine weather the corresponding ports are to configured as input or output

A		B		GROUP A		GROUP B	
D4	D3	D1	D0	PORT A	PORTC U	PORT B	PORT C L
0	0	0	0	OUT	OUT	OUT	OUT
0	0	0	1	OUT	OUT	OUT	IP
0	0	1	0	OUT	OUT	IP	OUT
0	0	1	1	OUT	OUT	IP	IP
0	1	0	0	OUT	IP	OUT	OUT
0	1	0	1	OUT	IP	OUT	IP
0	1	1	0	OUT	IP	IP	OUT
0	1	1	1	OUT	IP	IP	IP
1	0	0	0	IP	OUT	OUT	OUT
1	0	0	1	IP	OUT	OUT	IP
1	0	1	0	IP	OUT	IP	OUT
1	0	1	1	IP	OUT	IP	IP
1	1	0	0	IP	IP	OUT	OUT
1	1	0	1	IP	IP	OUT	IP
1	1	1	0	IP	IP	IP	OUT
1	1	1	1	IP	IP	IP	IP

Programming in MODE 1



(a) Status Word for Mode 1 Input Configuration



(b) Status Word for Mode 1 Output Configuration

Fig. 9a.8: Status word for mode 1 (a) Input (b) Output configuration

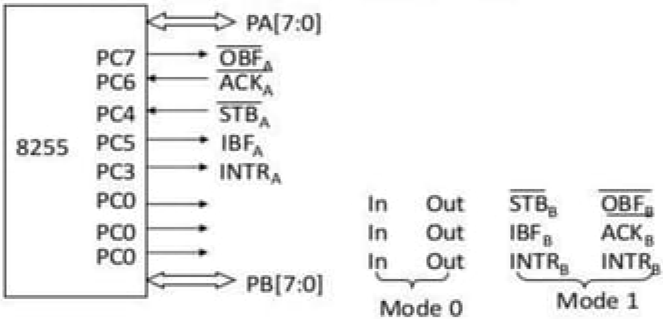
- IBF- input buffer full
- INTR- interrupt request
- INTE-interrupt enable

- OBF-output buffer full
- INTR-interrupt request
- INTE-interrupt enable

Programming in MODE 2

Programming 8255

- Mode 2:
  - Port A is programmed to be bi-directional
  - Port C is for handshaking
  - Port B can be either input or output in mode 0 or mode 1



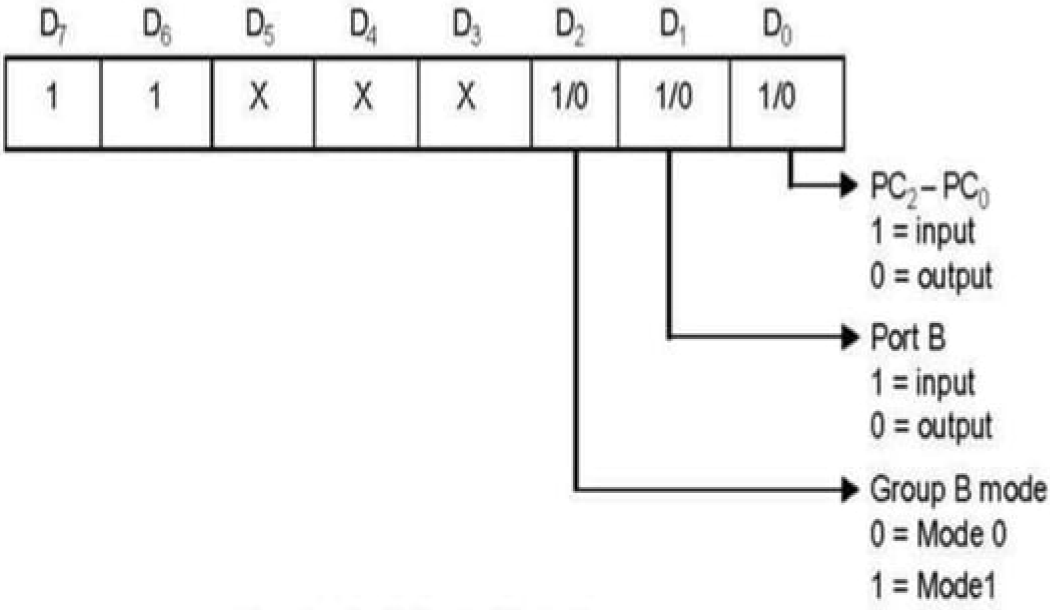
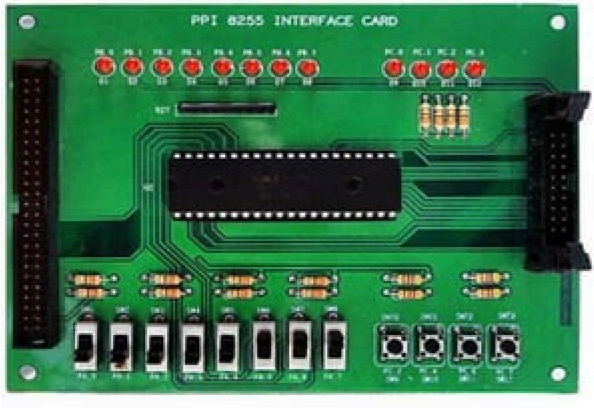
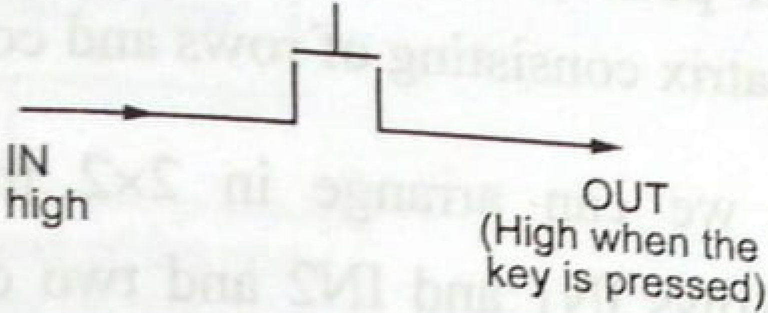


Fig. 9a.9: CWR in Mode 2

Interfacing cable





Basic Key operation

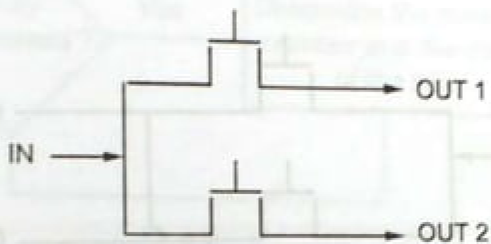


Figure 3.13 Basic Keyboard Operation – two keys

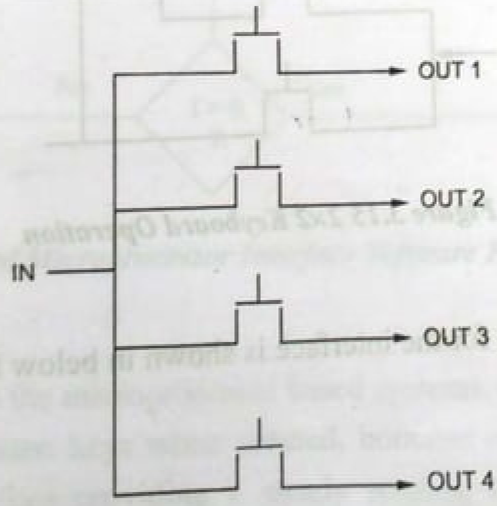
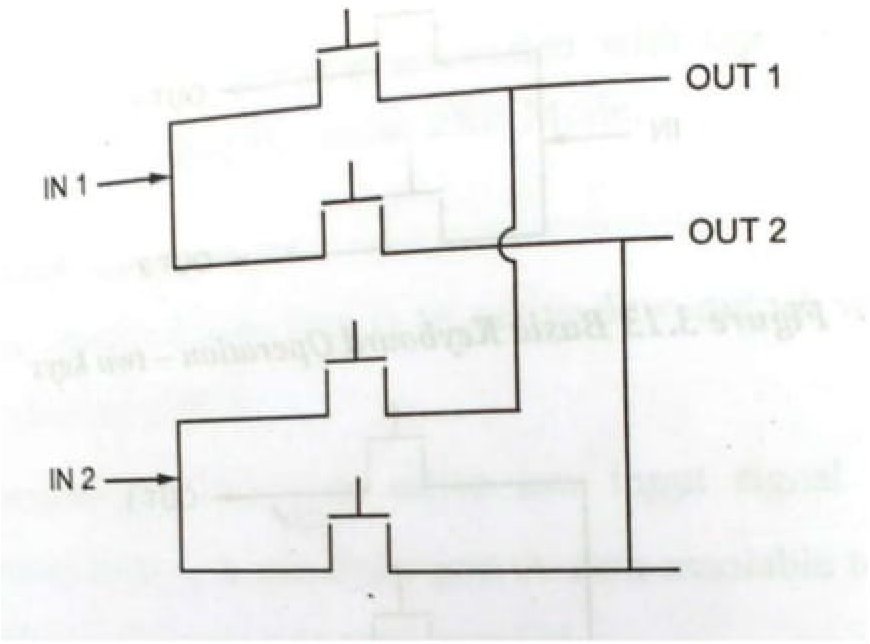
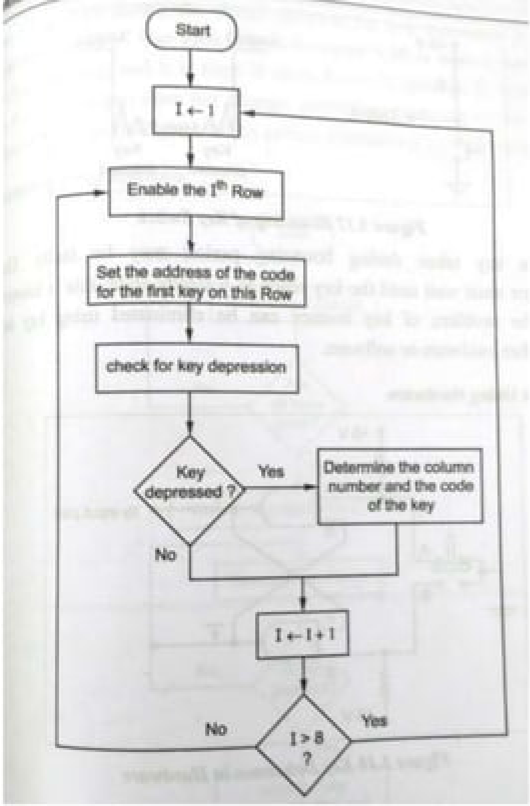


Figure 3.14 Basic Keyboard Operation – Four Keys

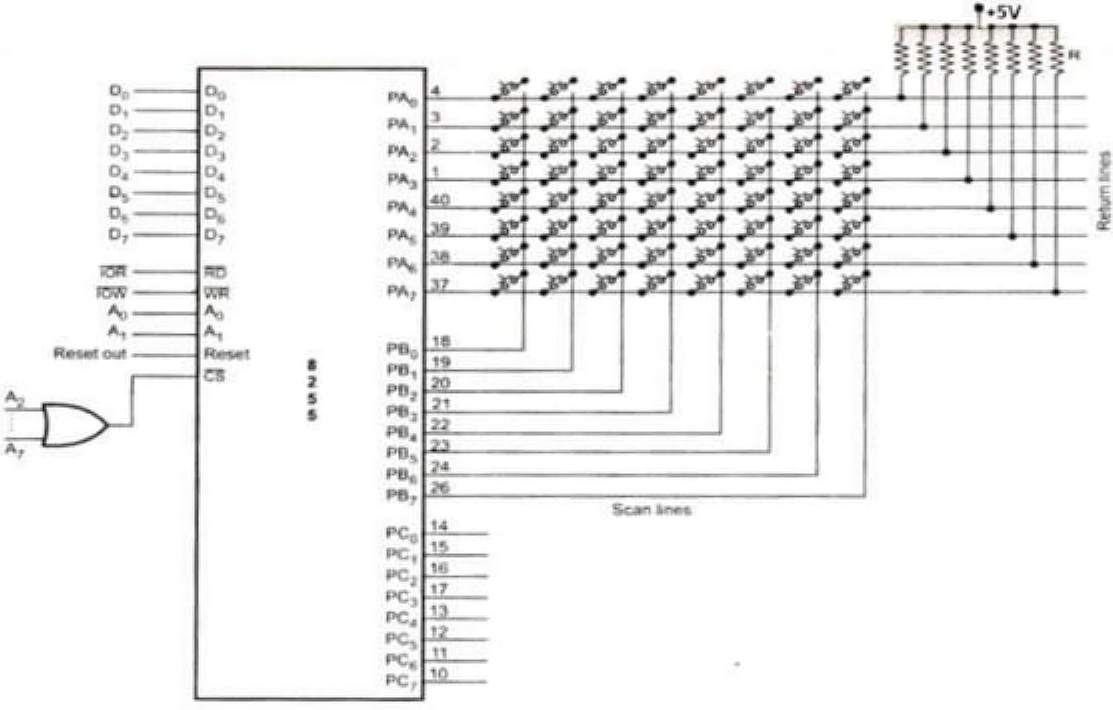


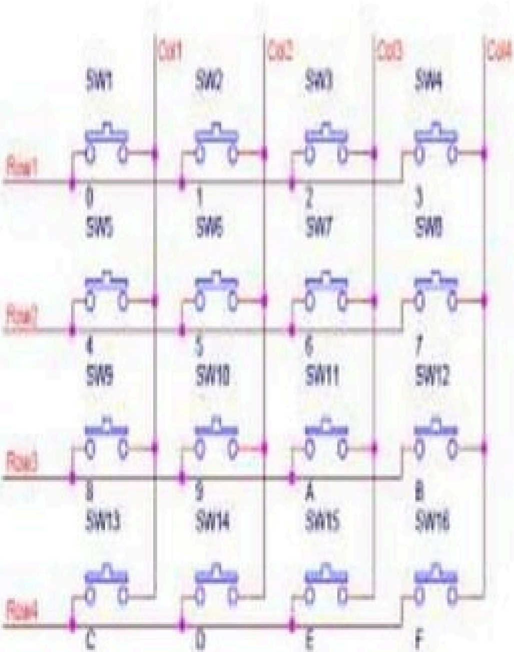
2 X 2 Key operation

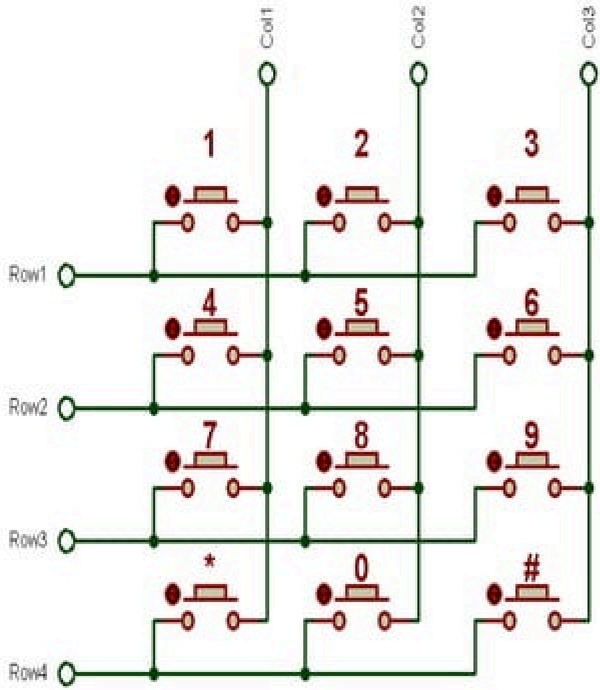


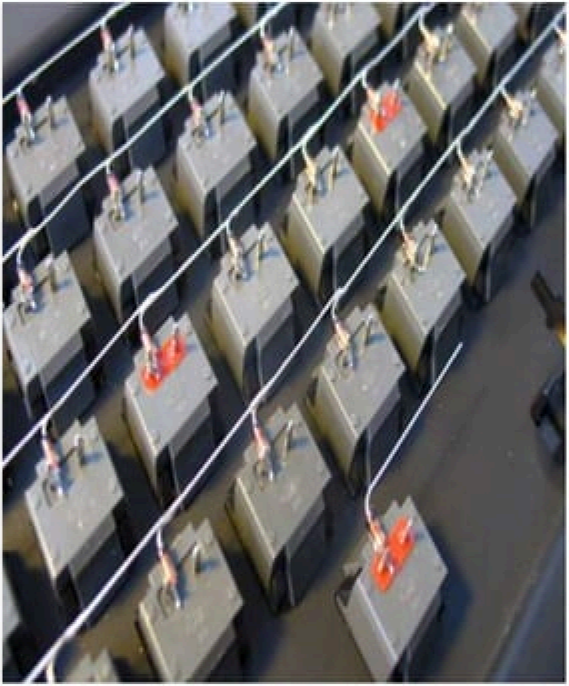
Keyboard Microprocessor Interface software Flowchart

INTERFACING-keyboard

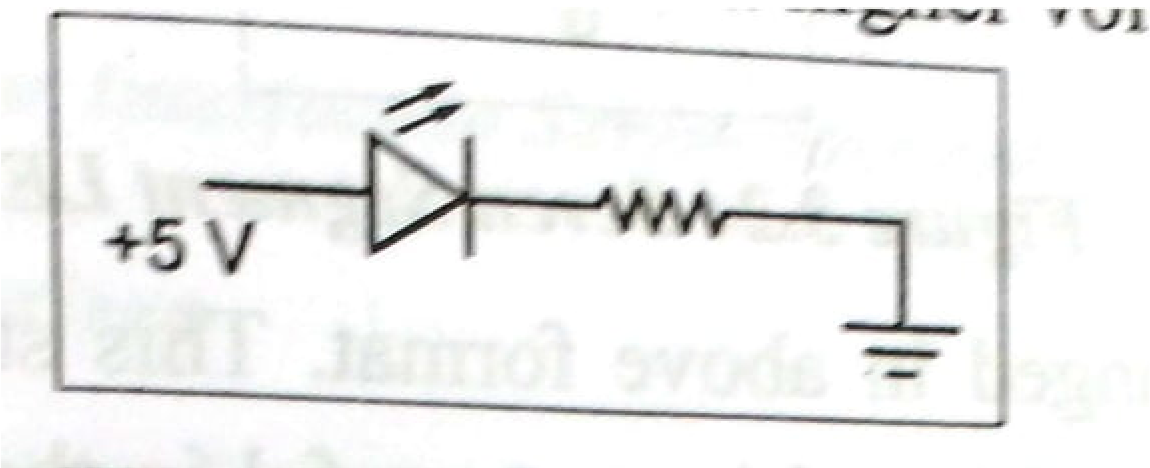



Col1
Col2
Col3
Col4
SW1
SW2
SW3
SW4
Row1
0
1
2
3
SW5
SW6
SW7
SW8
Row2
4
5
6
7
SW9
SW10
SW11
SW12
Row3
8
9
A
B
SW13
SW14
SW15
SW16
Row4
C
D
E
F

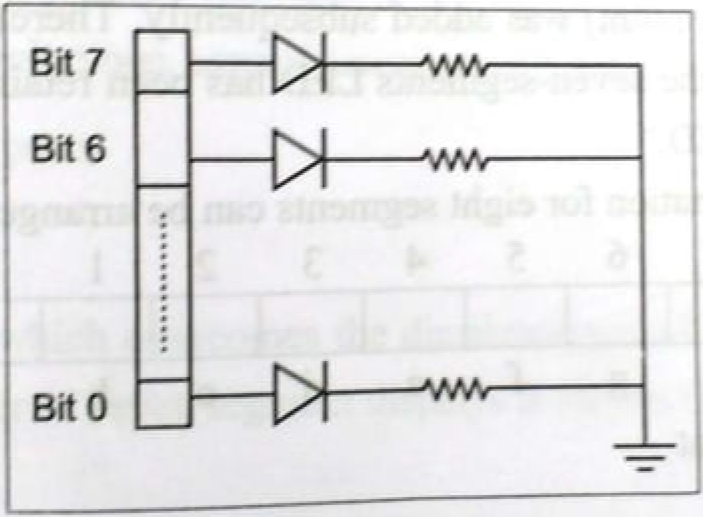

Col1
Col2
Col3
1
2
3
Row1
4
5
6
Row2
7
8
9
Row3
*
0
#
Row4




LED Operation

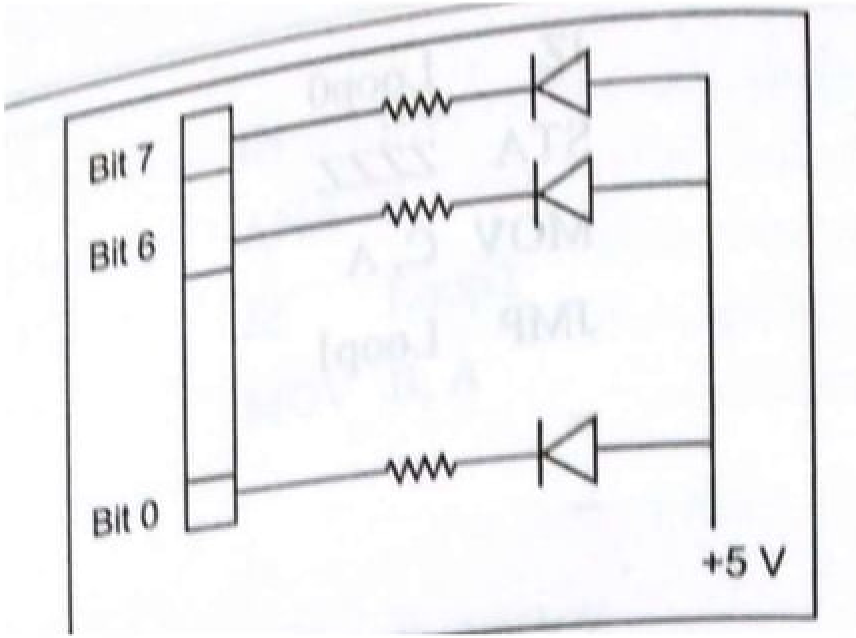


igure 3.22 :



re 3.22 Microprocessor Interface to LED (Common Cathode)

Microprocessor interface to LED (Common anode)



ited.

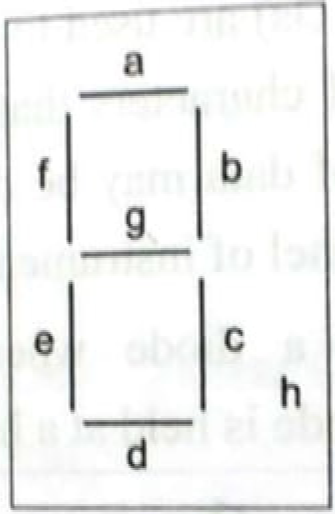
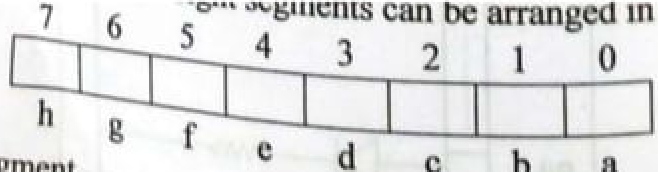


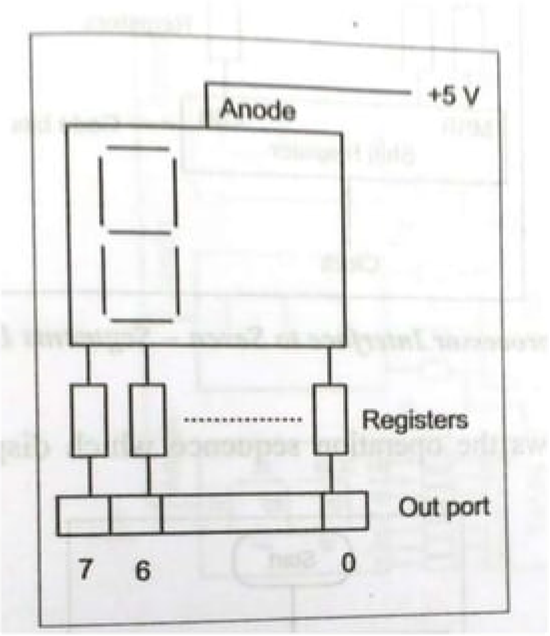
Figure 3.24 Seven Segment LED

...segments can be arranged in on

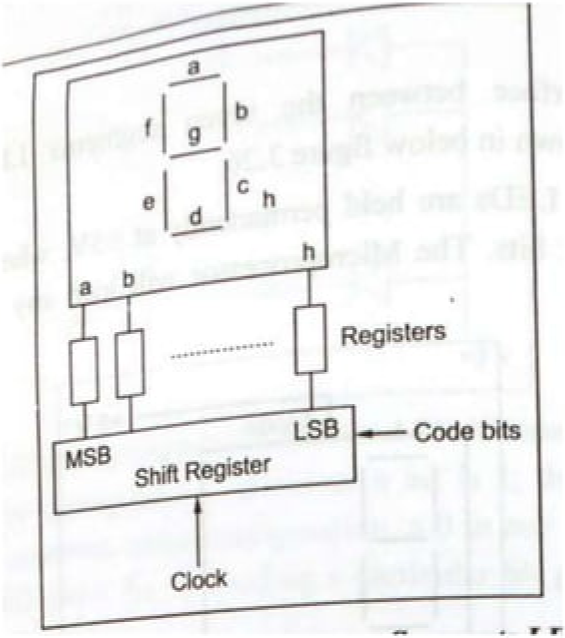


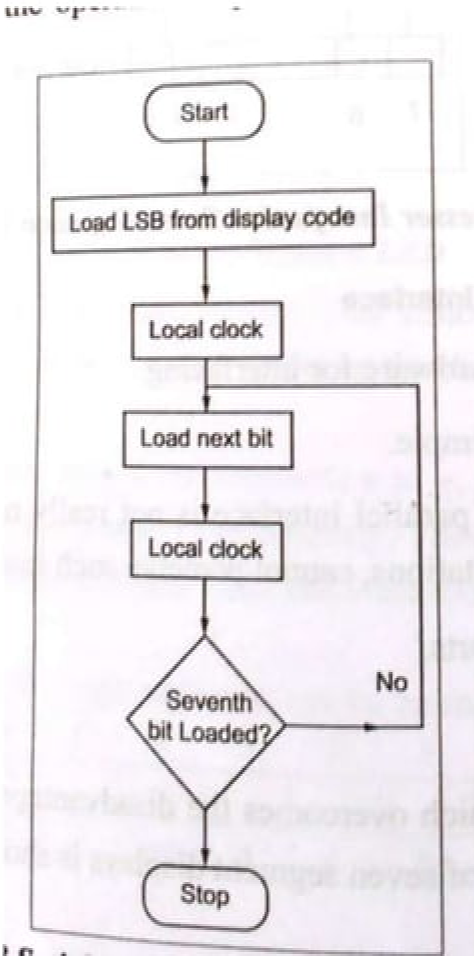
0 – Glow the segment
1 – Do not glow

Microprocessor interface to 7 segment LED (Parallel)



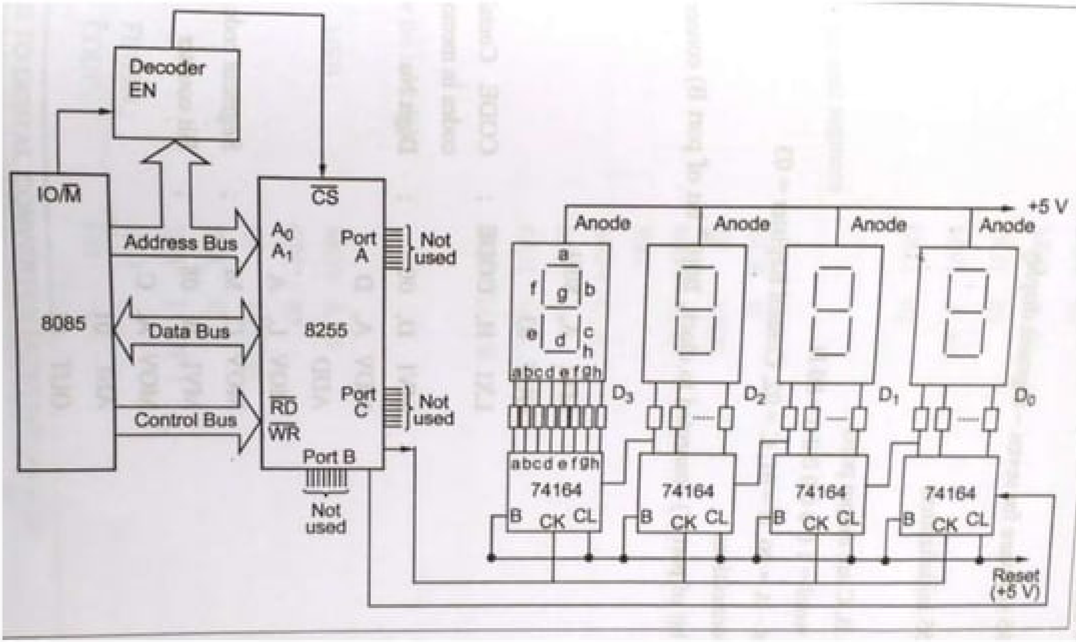
Microprocessor interface to 7 segment LED (serial)

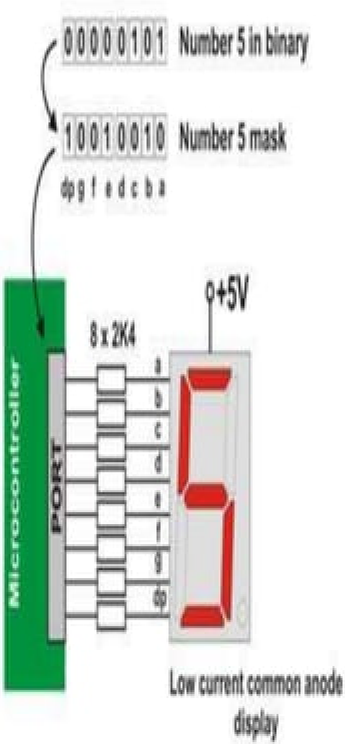




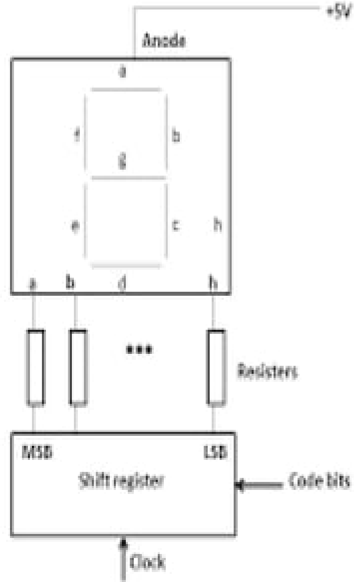
Serial interface of 7 segment LED to Microprocessor software flowchart

INTERFACE-LED display



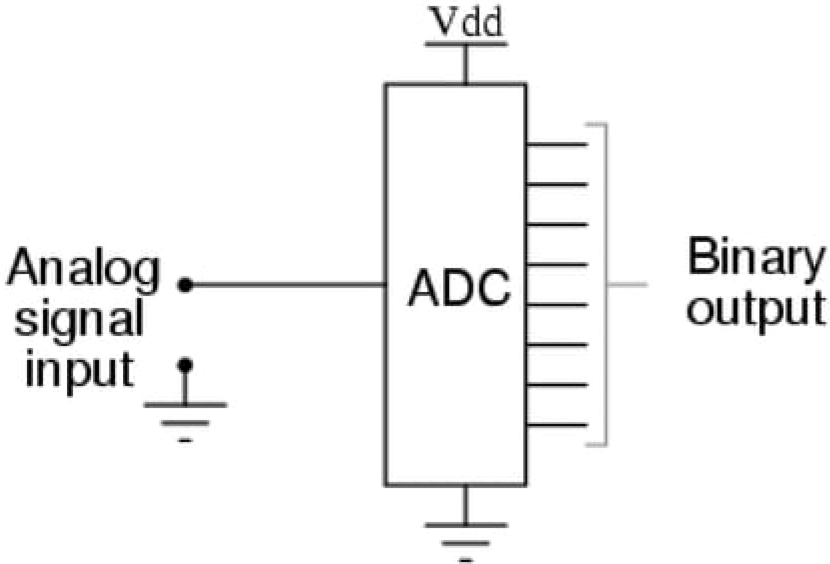


Hex value is 92H



Microprocessor interface to seven-segment LED(serial interface)

ADC INTERFACE



BLOCK diagram of ADC 0808

Block Diagram of ADC 0808/0809

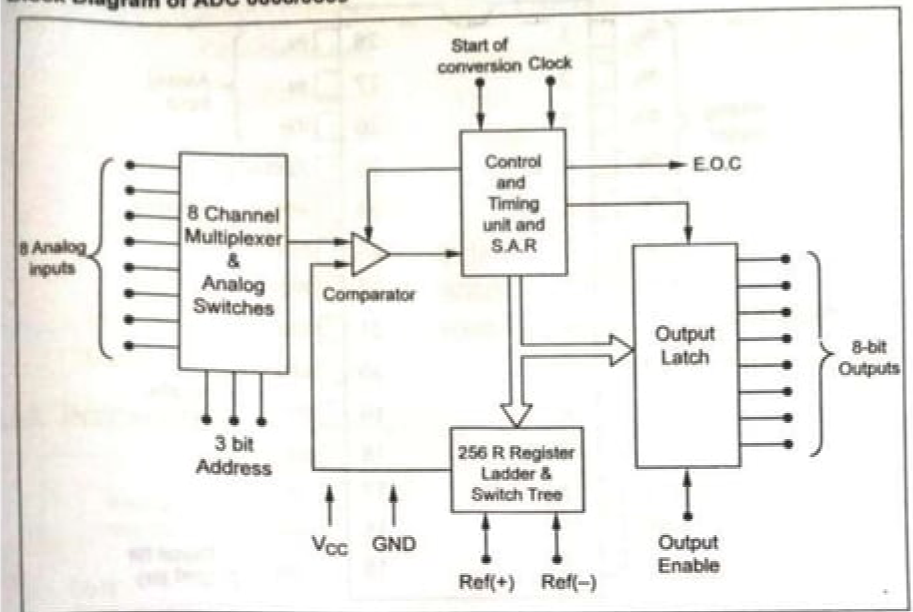


Figure 3.31 Block Diagram of ADDC 0808/ADC 0809

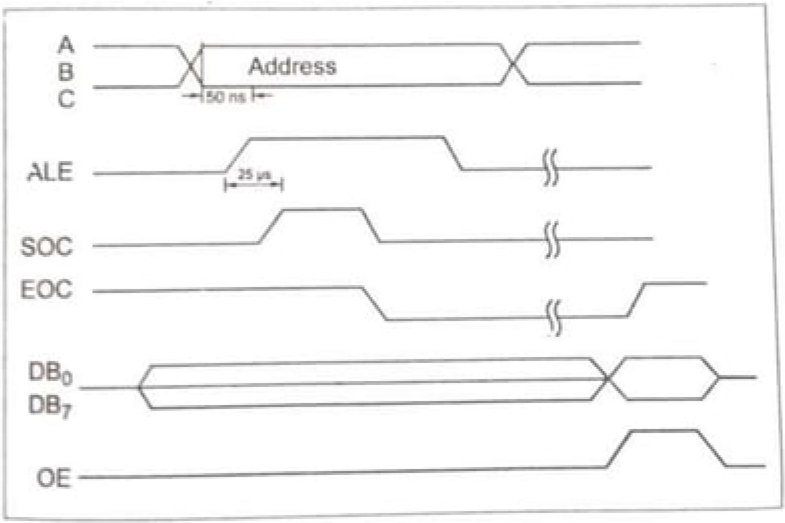
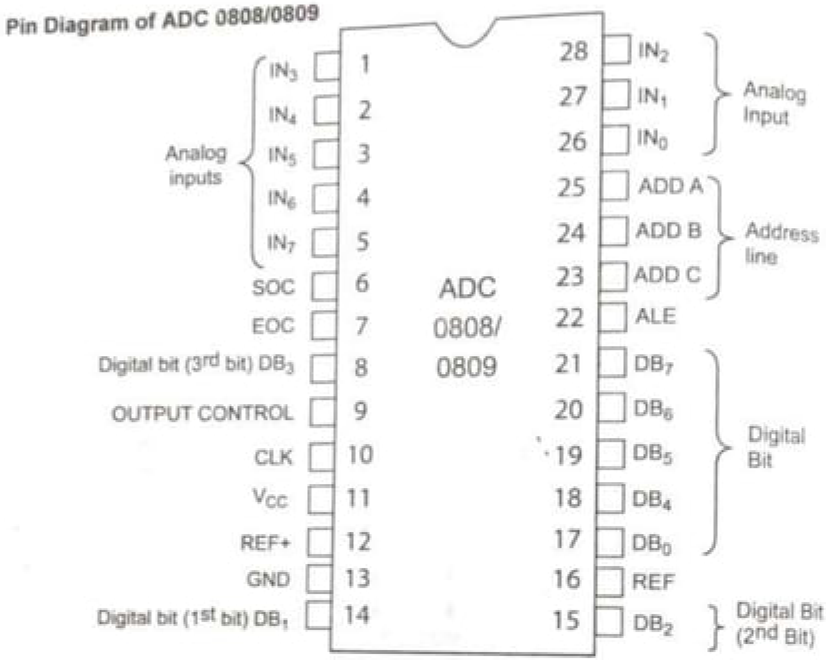
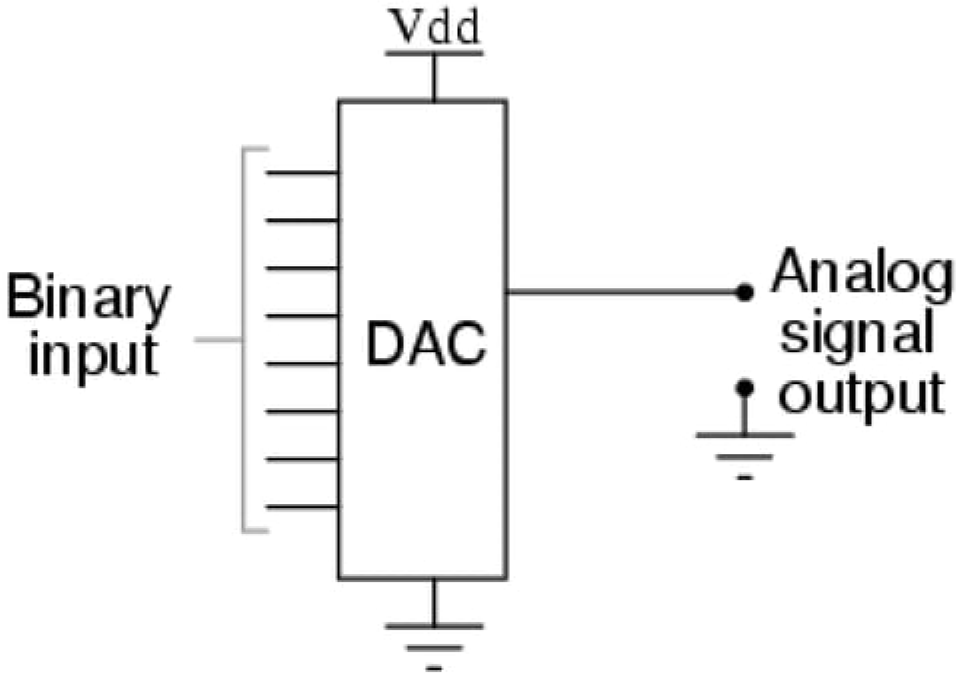


Figure 3.32 Timing Waveforms for ADC 0808

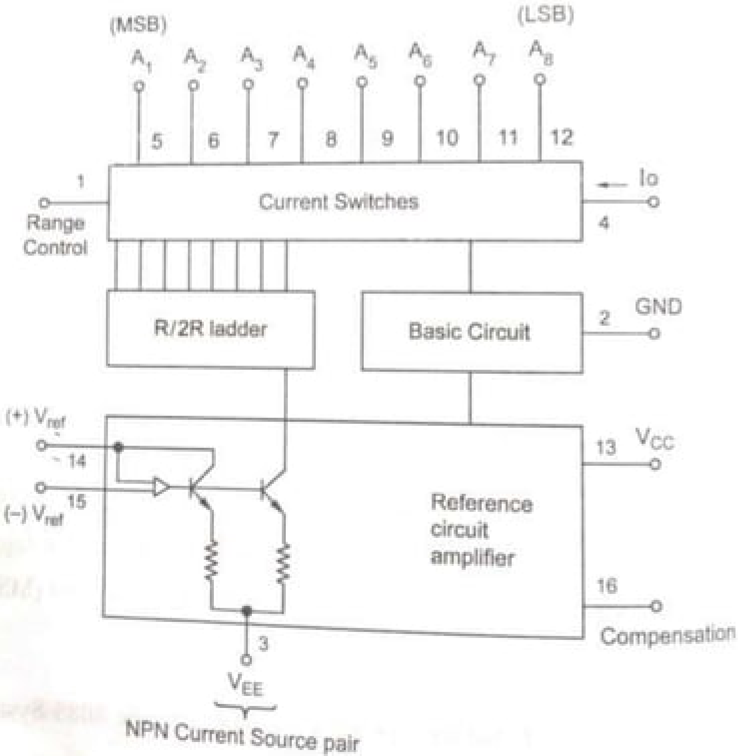
PIN diagram of ADC 0808



DAC INTERFACE



Pin diagram of DAC



Pin diagram of DAC

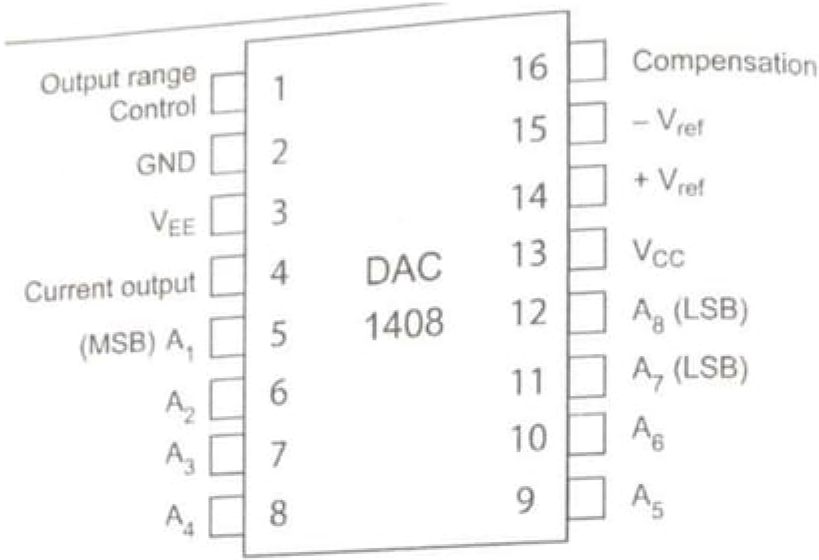
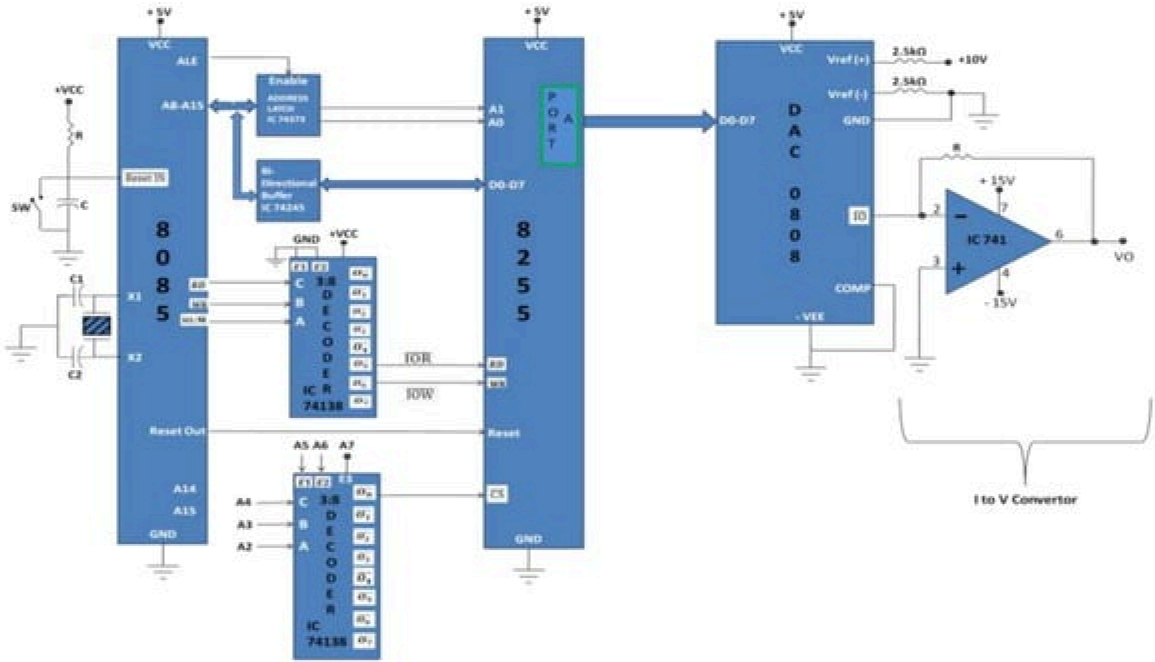


Figure 3.36 Pin Diagram of DAC 1408

INTERFACING diagram for DAC

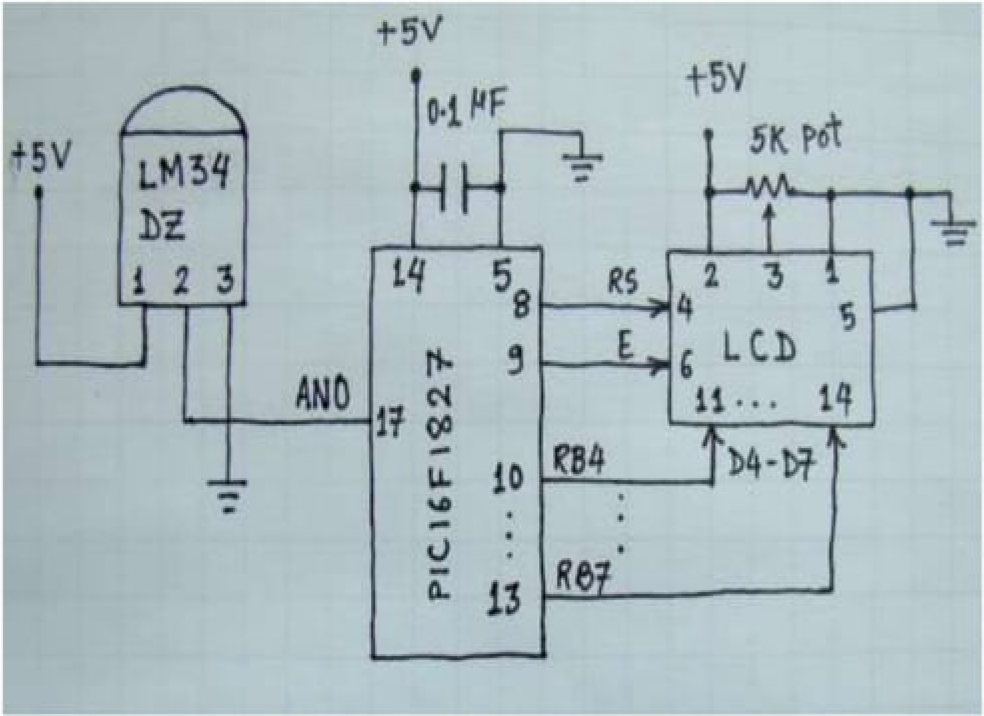


TEMPERATURE CONTROL

- **Temperature sensor** –convert temp to electrical signal by thermistor
- **Transducer** convert physical data into electrical signal
- **Physical data** –temp, light, flow, speed etc…
- **LM34 & LM35** –temperature sensor by NATIONAL SEMICONDUCTOR CO-OPERATION

- **LM34**
- Output voltage is linearly proportional to **Fahrenheit** temp
- No external **calibration**
- **10mV** for each degree of Fahrenheit temp

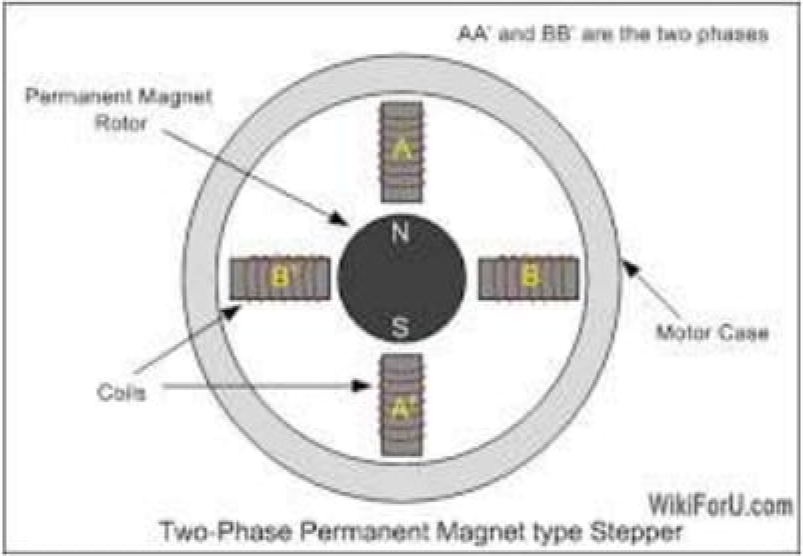
- **LM35**
- Output voltage is linearly proportional to **Celsius** temp
- No external **calibration**
- **10mV** for each degree of Centigrate temp



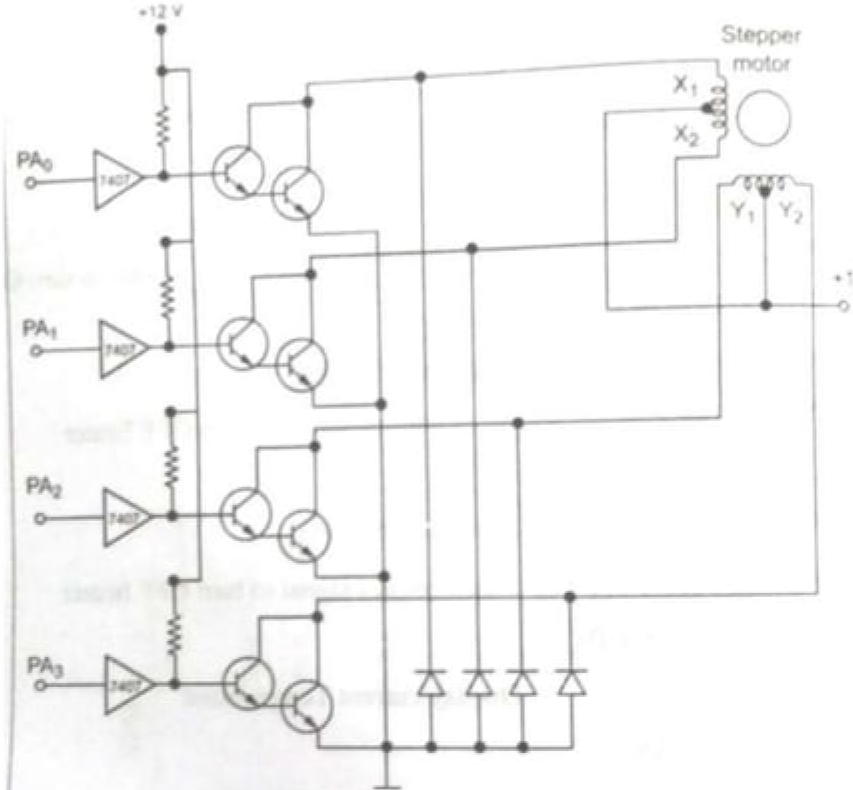
+5V
0·1 μF
+5V
5K Pot
+5V
LM34
DZ
1 2 3
14
5
8
9
17
10
13
RS
E
AN0
RB4
RB7
PIC16F1827
2 3 1
4
5
6
LCD
11 ... 14
D4-D7

STEPPER MOTOR CONTROL interface

- Digital motor used to translate electrical pulse into mechanical movement
- Center tap winding connected to 12 V supply
- Motor can be excited by grounding four terminals of the two windings
- **ROTOR**-Stepper motor has permanent magnet rotor .It is also known as shaft
- **STEP ANGLE-**It is minimum degree of rotation associated with a single step


AA' and BB' are the two phases
Permanent Magnet
Rotor
A
N
B'
B
S
Motor Case
Coils
A'
WikiForU.com
Two-Phase Permanent Magnet type Stepper

Stepper Motor Interface



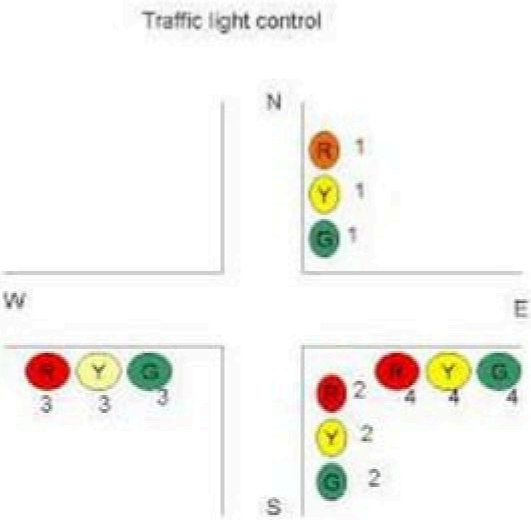
Excitation Table

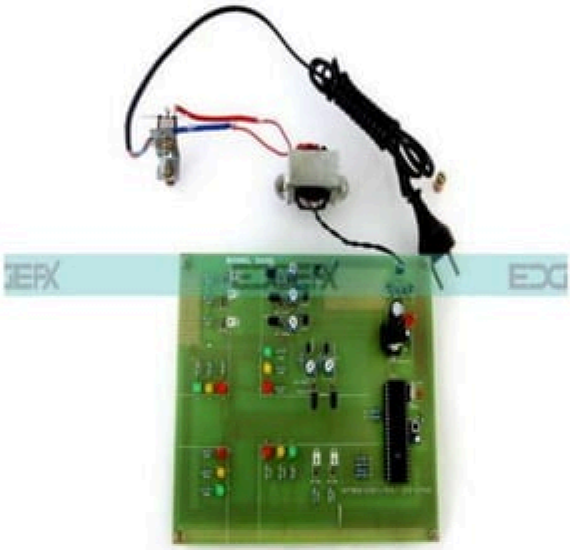
Step	X1	X2	X3	X4
1	0	1	0	1
2	1	0	0	1
3	1	0	1	0
4	0	1	1	0
1	0	1	0	1

Traffic Light Control System

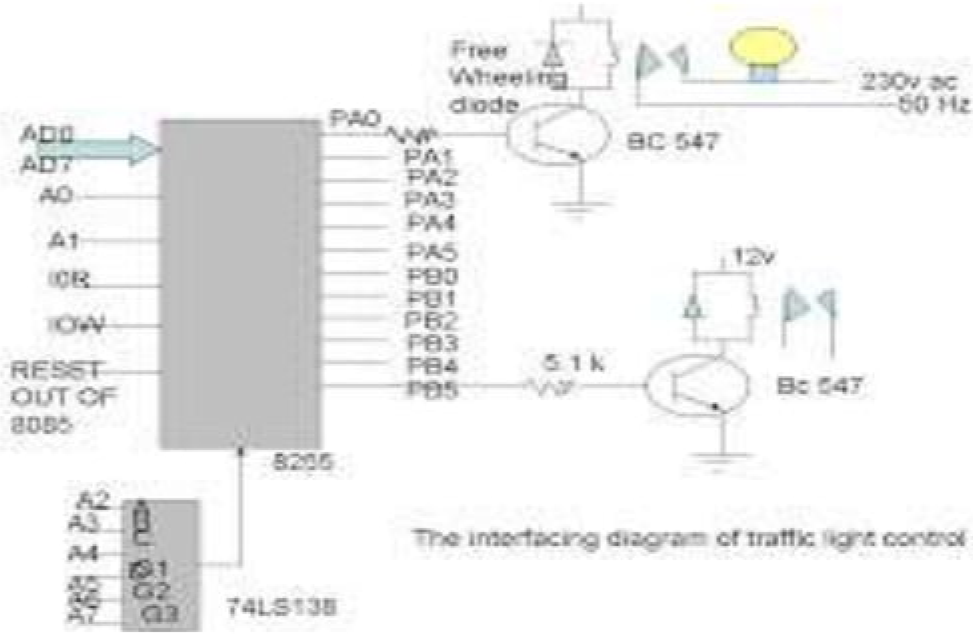
- Allow traffic from W to E and E to W transition for 20 seconds
- Give transition period of 5 seconds (yellow bulbs ON)
- Allow traffic from N to s and S to n for 20 seconds
- Give transition period of 5 seconds (yellow bulbs ON)
- Repeat the process

Traffic Light Control System





Interfacing diagram for Traffic Light Control System

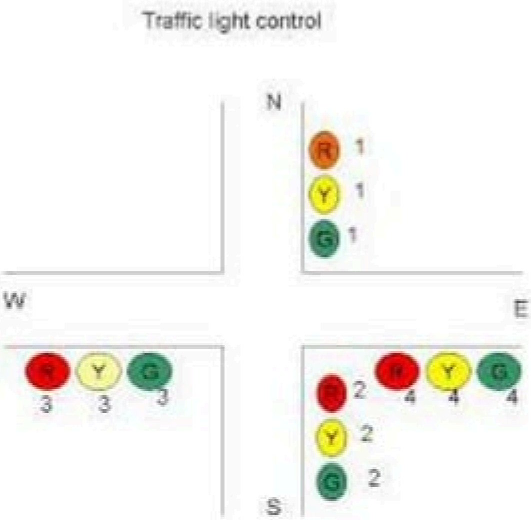


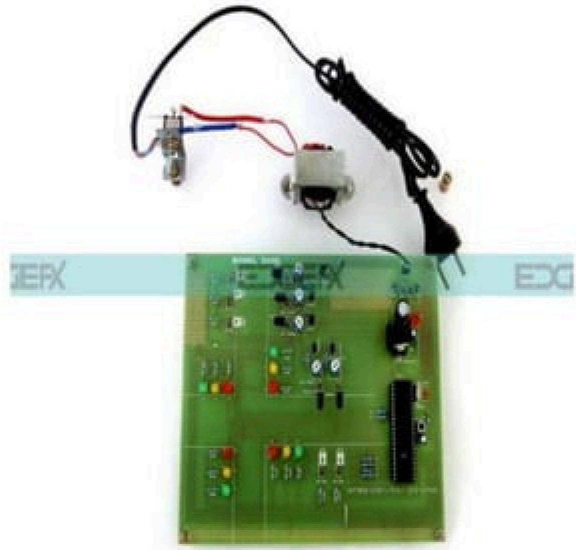
The interfacing diagram of traffic light control system

Pins	Light	Pins	Light
PA_0	R_1	PB_0	R_3
PA_1	Y_1	PB_1	Y_3
PA_2	G_1	PB_2	G_3
PA_3	R_2	PB_3	R_4
PA_4	Y_2	PB_4	Y_4
PA_5	G_2	PB_5	G_4

Table 1

Traffic Light Control System





KEYBOARD INTERFACING

Program for Keyboard Interface in 8085

```
;
  Intel 8255 Control Word = 1 0 0 1 0 0 0 0 = 90H, Port A = INPUT,
  Port B = OUTPUT,
; Mode = 0
; Port Nos – Port A = 00, Port B = 01, Port C = 02, control Register Port = 03
;
                MVI   A, 90H
                OUT   03; initialize Intel 8255
Loop0:          MVI   C, 01; Row counter
                MOV   A, C
                STA   ZZZZ;     store Row No in ZZZZ
;
; Set address XXXX of code Table (First Row)
;
Loop1:          LXI   H, XXXX
                SUI   01
                RAL
                RAL
                RAL   ; (A) = (Row No, – 1) × 8
                ADD   L
                MOV   L, A ;    Address of First Row Set
                MOV   A, C
                OUT   0 1
```

```
                IN      00
                ANA     A
                JZ      Loop2
                MOV     B, A
;
;   Delay Loop
;       DD:     LXI     D, YYYY
                DCX     D
                JNZ     DD
                IN      00
                CMP     B
                JNZ     Loop2
                MVI     D, 00

                STC
                CMC     ; Clear Carry Flag
        RR:     RRC
                JC      CODE
                INR     D
                JMP     RR  ; Calculate column No
        CODE:   MOV     A, D
                ADD     L
                MOV     L, A  ; Modify the address of code Table
                MOV     A, M
                CALL USE ;      use the code for processing
;
;   Increment the ROW No
;
        Loop 2: LDA     ZZZZ
                INR     A
                CPI     09
```

```
.           JZ     Loop0
            STA    ZZZZ
            MOV    C, A
            JMP    Loop1
            –
            –
            –
ZZZZ        DB     0;   Row No
XXXX        DB     –, –, –, –, –, –, –; XXXX is code Table.
```

LED INTERFACE

Program for Seven Segment LED display Interface in 8055

```
;
;  Intel 8085 software for seven – segment display
;
;  Intel 8255 initialization
;
;  ports A, B, C are output ports
;  Control word = 1 0 0 0 0 0 0 0 = 80 H
;  Port Nos – A = 00, B = 01, C = 02, Control Register = 03
;  circuit connections
;  C0 (0th bit of port C) connected to clock, B0(0th bit of port B) connected to code
   input
;
                MVI  A,  80H
                OUT  03
                LXI  H,  CODE    ;  CODE Contains display
                                    codes in memory
                MVI  D,  00      ;  Digit No
; Loop1         MOV  A,  D
                ADD  L
                MOV  L,  A
                MOV  C,  M       ;  Segment code in register C
                MVI  B,  08      ;  Bit counter
                MOV  A,  C
                ANI  01
                OUT  01
;
;  Output Clock
```

```
CLOCK:          MVI   A, 00
                OUT   02
                MVI   A, 01
                OUT   02
                MVI   A, 00
                OUT   02
;
; Repeat for the next segment
;
                DCR   B
                JZ    MSB
                MOV   A, C
                RRC
                MOV   C, A
                ANI   01
                OUT   01H, AL
                JMP CLOCK
;
; Repeat for the next digit
;
MSB :           MOV   A, D
                CPI   03
                JZ    FINISH
                INR   D
                JMP   LOOP 1
FINISH :        HLT
CODE            DB    -, -, -, -
```

TEMPERATURE CONTROL

Program

```
        LXI   SP, 2800H        ; Initialize Stack Pointer
        MVI   A, 98H           ; Initialize 8255
        OUT   83H (CWR)
START:  MVI   A, 03H           ; select channel IN3 and gets high to low
        OUT   82H(PORTC)
        MVI   A, 03H
        OUT   82H(PORTC)
L1:     IN    82H(PORTC)       ; checks EOC
        RAL
        JNC   L1               ; IS EOC = 1? If no then Jump to L1
        IN    80H(PORTA)       ; IS EOC = 1? If yes then Read temperature (t1)
        MOV   B, A             ; save reading of L1 into B register
        CPI   A, DATA          ; IS t1 = T1? If yes then Jump to START
        JZ    START
        JC    OFF              ; IS t1 > T1? If yes then Jump to OFF
        MVI   A, 01H           ; IS t1 > T1? If no then send signals to turn ON
        OOT   81H(PORT B)
        JMP   DISPLAY          ; Jump to DISPLAY
        MVI   A, 00H           ; sends control signal to turn OFF heater
        OUT   81H(PORT B)
        JMP   DISPLAY          ; Jump to DISPLAY
OFF:    MVI   A, 00H           ; Sends control signal to turn OFF heater
        OUT   81H(PORT B)
DISPLAY:MOV   A, B             ; Display current Temperature
        CALL  DISPLAY
        JMP   START            ; Repeat the process
```

STEPPER MOTOR

Program

; Stepper Motor Control Program

6000 H Excite-code DB 03H, 06H, 09H, OCH; code sequence for clockwise rotation

; Subroutine to rotate a stepper motor clockwise by 360°

```
; Set the counts
        MVI   C, 32H      ; Set Repetition count to 50₁₀
START:  MVI   B, 04H      ; Counts excitation sequence
        LXI   H, 6000H    ; Initialize pointer
BACK1:  MOV   A, M        ; Get the Excite-code
        OUT   PORTA       ; Send Excite-code
        CALL  DELAY       ; Wait
        INX   H           ; Increment pointer
        DCR   B           ; Repeat 4 times
        JNZ   BACK1
        DCR   C           ; Repeat 50 times
        JNZ   START
        RET
```

TRAFFIC CONTROL

Source Program

```
          MVI A, 80H       ;  Initialize 8255, port A and port B
          OUT 83H (CR)     ;  in output mode
START :   MVI A, 09H       ;
          OUT 80H (PA)     ;  send data on PA to glow R1 and R2
          MVI A, 24H
          OUT 81H (PB)     ;  send data on PB to glow G3 and G4
          MVI C, 28H       ;  Load multiplier count (40 base 10) for delay
          CALL DELAY       ;  call delay subroutine
          MVI A, 12H
          OUT (81H) PA     ;  send data on port A to glow Y1 and Y2
          OUT (81H) PB     ;  send data on port B to glow Y3 and Y4
          MVI C, OAH       ;  Load multiplier count (10 base 10) for delay
          CALL DELAY       ;  call delay subroutine
          MVI A, 24H
```

```
        OUT (80H) PA     ;  send data on port A to glow G1 and G2
        MVI A, 24H
        OUT (80H) PA     ;  send data on port B to glow R3 and R4
        MVI C, 28H       ;  load multiplier count (40 decimal) for delay
        CALL DELAY       ;  call delay subroutine
        MVI A, 12H
        OUT PA           ;  send data on port A to glow Y1 and Y2
        OUT PB           ;  send data on port B to glow Y3 and Y4
        MVI, C, OAH      ;  Load multiplier count (10 decimal) for delay
        CALL DELAY       ;  call delay subroutine
        JMP START
```

Delay Subroutine

```
DELAY :  LXI   D, count  ;  load count to give 0.5 sec delay
BACK  :  DCX   D         ;  Decrement counter
         MOV   A, D
         ORA   E         ;  check whether count is 0
         JNZ   BACK      ;  If not zero, repeat
         DCR   C         ;  check if multiplier zero;
         JNZ DELAY       ;  otherwise repeat
         RET             ;  Return to main program
```