Interfacing Keyboard
and
Seven Segment display

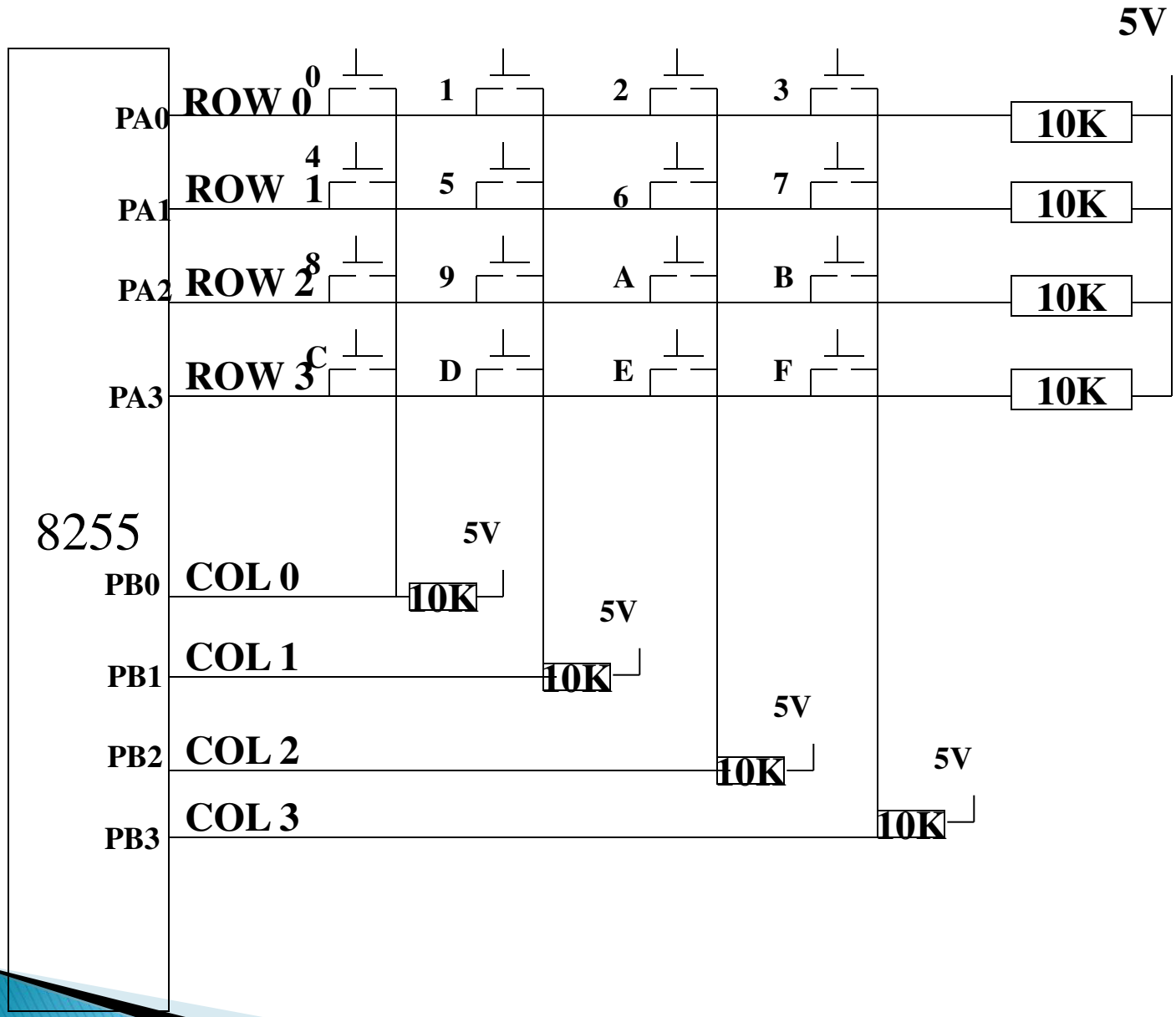
Introduction

- There are lot of research scope in field of microprocessor, today word is electronic world and computer and electronic equipment are using in every field. Improve the quality and feature of the existing microprocessor new research are going on. so there are lot of research scope in this field.

Applications

- Using in calculators keypad.
- Computer keyboard.
- Video games.
- Mobile phones
- Telephone
- PDA’s

INTERFACING KEYBOARD



8086 HAS TO

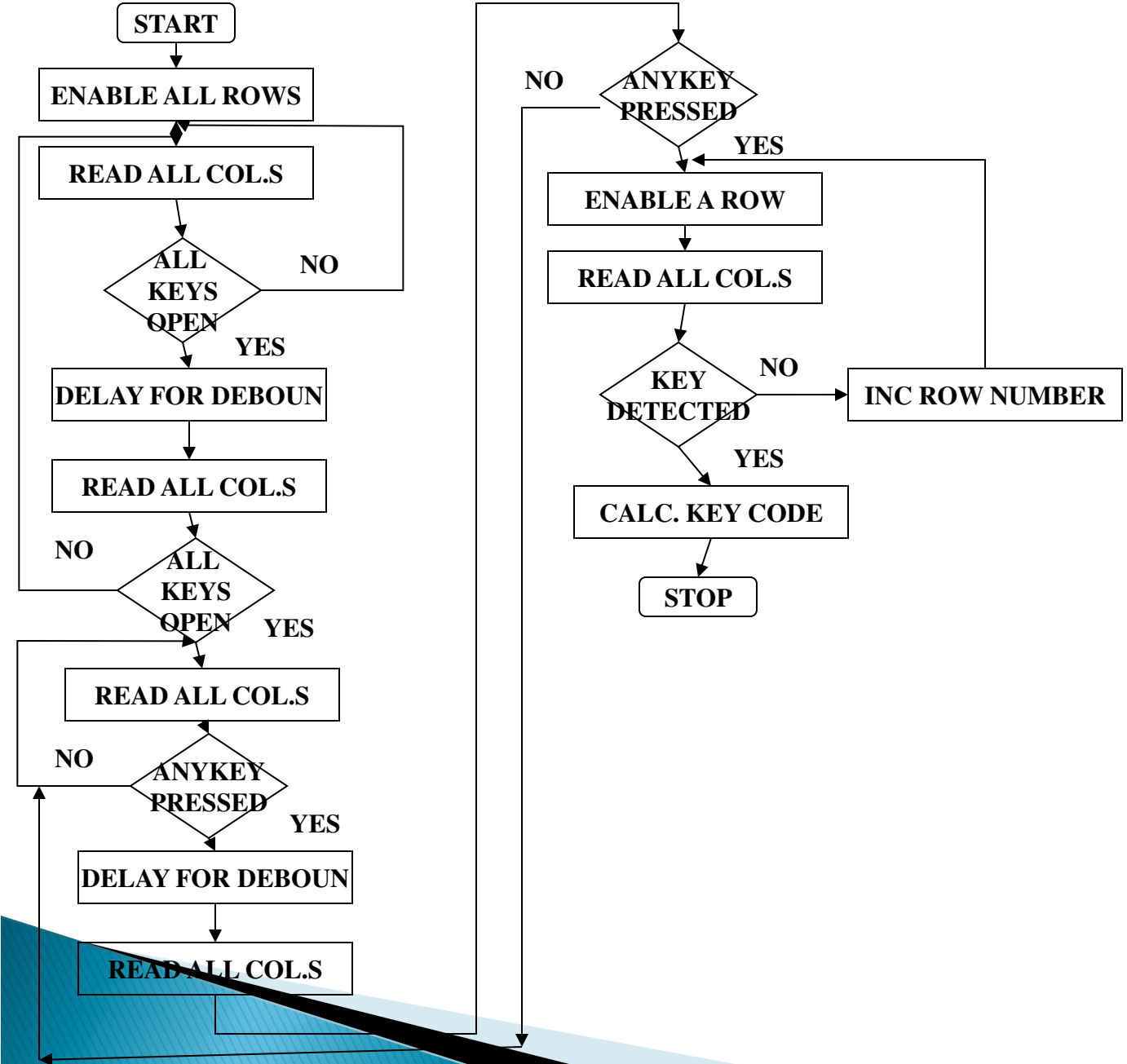
1. DETECT A KEY PRESS
2. DEBOUNCE A KEY PRESS
3. GENERATE A CODE CORRESPONDING TO THE KEY BEING PRESSED

SOFTWRAE ASPECTS

ALGORITHM

1. WAIT till all keys are released. Use s.w debounce for each key che
2. Wait for key closure
3. Confirm key closure
4. Find number of row and column to which key belongs
5. Convert the row and col information to entry number of the table which contains ASCII code
6. Get code and repeat in infinite loop

Flow chart

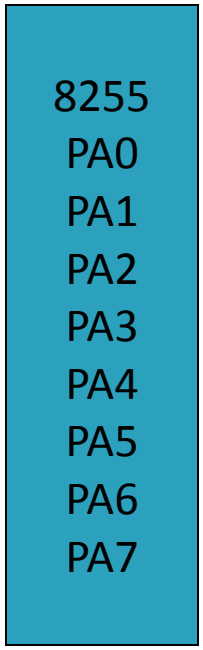

START
ENABLE ALL ROWS
READ ALL COL.S
ALL KEYS OPEN
NO
YES
DELAY FOR DEBOUN
READ ALL COL.S
NO
ALL KEYS OPEN
YES
READ ALL COL.S
NO
ANYKEY PRESSED
YES
DELAY FOR DEBOUN
READ ALL COL.S
NO
ANYKEY PRESSED
YES
ENABLE A ROW
READ ALL COL.S
KEY DETECTED
NO
INC ROW NUMBER
YES
CALC. KEY CODE
STOP

PROGRAM

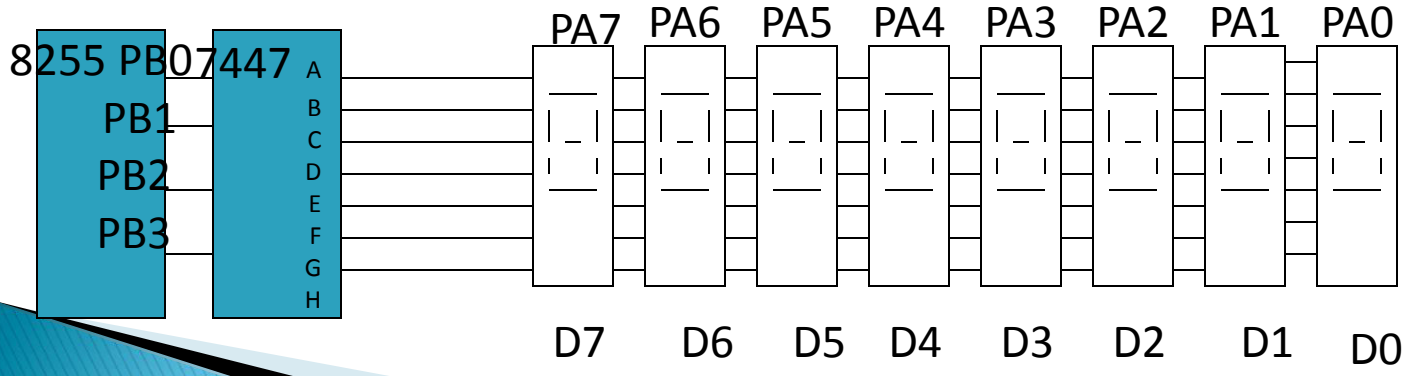
```
            DATA SEGMENT
            CNTRPRT EQU 8003H
            PORTA EQU 8000H
            PORTB EQU 8001H
            DELAY EQU 6666
            TABLE DB 30H,31H,32H,.....39H,41H,....46H  ;ASCII CODES FROM 0 TO F
            DATA ENDS
            CODE  SEGMENT
            ASUUME CS:CODE,DS:DATA
START:      MOV AX,DATA
            MOV DS,AX
            MOV AL,82H
            MOV DX,CNTRPRT          :PORT A AS I/P PORT PORT B AS O/P PORT
            OUT DX,AL
            XOR AL,AL
            MOV DX,PORTA
            OUT DX,AL               ;ENABLE ALL ROWS
            MOV DX,PORTB
RDCOL:      IN AL,DX                ;GTE COL STATUS
            AND AL,0FH              ;MASK UNWANTED BITS
            CMP AL,0FH              ;GET READY FOR CHKING COL SATTUS
            JNE RDCOL               ;IS ANY COL ACTIVE?IF YES CHK AGAIN
            MOV CX,DELAY            ;NO DEBOUNCE DEALY
SELF:       LOOP SELF
            IN AL,DX
            AND AL,0FH              ;CONFIRM COL STATUS AGAIN
            CMP AL,0FH
            JNE RDCOL               ;IF NOT CONFIRMED CHECK AGAIN
RDAGN:      IN AL,DX                ;CONFIRMED THAT ALL KEYS ARE OPEN,GET COL STATUS AGAIN
            AND AL,0FH
            CMP AL,0FH              ;CHECK FOR ANY KEY CLOSURE,IF NO CONTINUE TO CHECK,IF YES
            JE RDAGN                ;NEXT STEP
            MOV CX,DELAY
SELF1:      LOOP SELF1
```

```
                IN AL,DX
                AND AL,0FH              ;CONFIRM COL STATUS AGAIN
                JE RDAGN
                MOV AL,0FEH ;KEY CLOSURE CONFIRMES,SELECT ROW PATTTERN TO ENABLE A ROW
                MOV BL,AL               ;SAVE IT
ENROW:          MOV DX,PORTA
                OUT DX,AL               ;ENABLE CORRESPONDING ROW
                MOV DX,PORTB
                IN AL,DX                ;GET COL STATUS
                AND AL,0FH
                CMP AL,0FH              ;CHECK IF COL IS ACTIVE
                JNE CCODE               ;IF YES, GO TO CALCULATE ASCII CODE OF KEY PRESSED
                ROL BL,1                ;PREPARE TO ENABLE NEXT ROW
                MOV AL,BL
                JMP ENROW
CCODE:          MOV CL,0                ;AL CONTAINS COL PATTERN,BL CONTAINS ROW PATTERN
                                        ;INITIALIZE COL COUNT TO 0
NXTCOL:         ROR AL,1                ;COL STATUS GOES TO CARRY FLAG
                JNC CHKROW  ;IS COL ACTIVE, IF YES, CL CONTAINS COL.NUMBER
                INC CL                  ;NO INCREMENT COL COUNT
                JMP NXTCOL   ;CHECK NEXT COL
CHKROW:         MOV DL,0                ;CL CONTAINS COL NUMBER
                                        ;INITIALIZE ROW COUNT TO ZERO
NXTROW:         ROR BL,1                ;ROW STATUS GOES TO CARRY FLAG
                JNC CALADR    ;IS ROW ACTIVE? IF YES, DL CONTAINS ROW NUMBER
                ADD DL,04H              ;ROW COUNT+4 →ROW COUNT
                JMP NXTROW CHECK NEXT ROW
CALADR:         ADD DL,CL               ;ROW +COL
                MOV AL,DL
                LEA BX,TABLE
                XLAT                    ;GET ASCII CODE OF THE KEY PRESSED
                INT3H
                JMP START
                CODE ENDS
                END SATRT
```

INTERFACING THE LED DISPALY



CONNECT PA TO DISPLAY THROUGH PNP TRANSISTOR



ALGORITHM

1. TURN ON Q0 BY APPLYING A LOGICAL LOW TO BASE OF PNP TRANSISTOR
2. SEND 7-SEGMENT CODE FOR D0 (DIGIT 0)
3. AFTER 1MS TURN OFF QO,TURN ON Q1, OFF QO,Q2-Q7
4. SEND 7-SEGMENT CODE FOR D1(DIGIT 1)
5. AFTER 1MS TURN OFF Q1,TURN ON Q2 REMAINING Q'S OFF
6. REPEAT THE PROCESS FOR ALL 8-DIGITS.IT COMPLETES ONE CYCLE
7. START CYCLE AGAIN

PROGRAM

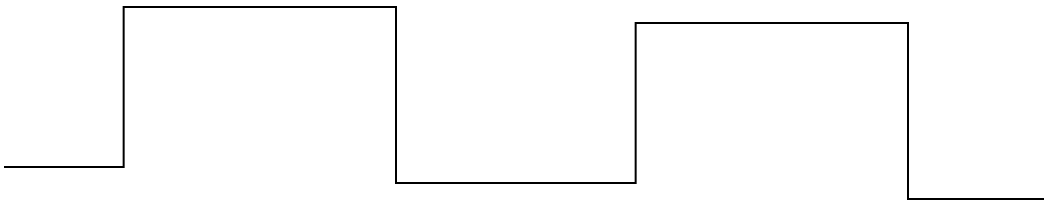
```
          DATA SEGMENT
          PORTA EQU 0FFF8H
          PORTB EQU 0FFF9F
          CTRLPORT EQU 0FFFBH
          DELAY EQU 012CH
          DIGITS DB 1,2,3,4,5,6,7,8
          DATA ENDS
          CODE  SEGMENT
          ASSUME CS:CODE,DS:DATA
START:    MOV AX,DATA
          MOV DS,AX
          MOV DX,CNTRLPORT          ;PORTA ,PORTB O/P PORTS
          MOV AL,80H
          OUT DX,AL
REPEAT:   MOV BH,8                  ;INITIALIZE DIGIT COUNT
          LEA SI,DIGITS       ;GET ADDRESS OF THE DIGIT TABLE
          MOV BL,0FEH         ;CODE TO TURN ON Q0
BACK:     MOV AL,BL
          MOV DX,PORTA              ;TURN ON Q0
          OUT DX,AL
          MOV AL,[SI]
          MOV DX,PORTB              ;GET DIGIT TO BE DISPLAYED
          OUT DX,AL                 ;SEND IT TO 7447 FOR DISPLAY
SELF:     MOV CX,DELAY              ;DELAY CONSTATNT FOR 1MS
          LOOP SELF
          INC SI
          ROL BL,1                  ;CODE TO TURN ON NEXT TRANSISTOR
          DEC BH                    ;DECREMENT DIGIT COUNT
          JNZ BACK
          JMP REPEAT
          CODE ENDS
          END START
```

D TO A CONVERTER

D/A CONVERTER CAN BE DIRECTLY CONNECTED TO 8255

LET US ASSUME THAT 8-BIT D/A CONVERTER USED IS HAVING FULL SCALE O/P VOLTAG EOF 0-5V. IT IS CONNECTED TO PORT A OF 8255. THE BASE ADDRESS SOF 8255 IS 8000H. CLOCK FREQUENCY IS 5MHZ

GENERATE A SQUARE WAVE OF 5VOLTS, 1KHZ FREQ



5VOLTS,500MICRO SEC

ALGORITHM

- SEND A VALUE 0 TO PORT A
- DELAY 500MICRO SEC
- SEND A VLAUE FFH TO PORT A(FOR +5V)
- REPEAT CYCLE INDIFINITELY

DELAY CALCULATIONS

LOOP INSTRUCTION USED FOR GENERATING REQUIRED DELAY,TAKES 17 CYCLES
TIME FOR 17 CYCLES = 17 X 200ns(CPU FREQ = 5MHZ, 1 CYCLE = 200NS)
- 3.4 MICRO SEC
HENCE ONE LOOP INSTRUCTION = 3.4 MICRO SEC
DELAY REQUIRED = 500MICRO SEC
LOOP INSTRUCTION SHOULD BE REPEATED FOR N WHERE
N = 500/3.4 = 147

PROGRAM

```
DATA SEGMENT
PORT EQU 8000H
CNTPRT EQU 8003H
DELAY EQU 147
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START:      MOV AX,DATA
MOV DS,AX
MOV AL,80H
MOV DX,CNTPRT
OUT DX,AL
MOV DX,PORTA
BACK:       MOV AL,00
OUT DX,AL
MOV CX,DELAY
SELF:       LOOP SELF
MOV AL,0FFH
OUT DX,AL
MOV CX,DELAY
SELF:       LOOP SEWLF
JMP BACK
INT 3H
CODE ENDS
END START
```

GENERATE RECTANGULAR WAVE OF 1V TO 4V,25% DUTY CYCLE, 500KHZ FREQ

ALGORITHM

1. SEND A VALUE CORRESAPONDING TO 1VOLT TO PORT A
2. AFTER 1500 MICRO SEC DELAY SEND
   A VALUE CORRESPONDING TO 4VOLTS TO PORT A
3. AFTER 500 MICRO SEC SEND FIRST VALUE(CORRESPONDING TO 1VOLT)
4. REPEAT CYCLE INDIFINITELY

DELAY CALCULATIONS

DELAY CONSTANT FOR 500 MICRO = 147
DELAY CONSTANT FOR 1500 MICRO = 147 X 3 = 441
BINARY VALUE FOR 5VOLT = FFH
BINARY VALUE FOR 1 VOLT = FF/5H= 255/5 = 51 = 33H
BINARY VALUE FOR 4VOLTS = 33H X 4 = CCH

PROGRAM

```
DADA SEGMENT
PORT EQU 8000H
CNTPRT EQU 8003H
DELAYH EQU 147
DELAYL EQU 441
LVOLT DB 33H
HVOLT DB 0CCH
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START:      MOV AX,DATA
MOV DS,AX
MOV AL,80H
MOV DX,CNTPRT
OUT DX,AL
BACK:       MOV AL,LVOLT
MOV DX,PORTA
OUT DX,AL
MOV CX,DELAYL
SELF:       LOOP SELF
MOV AL,HVOLTH
OUT DX,AL
MOV CX,DELAYH
SELF:       LOOP SEWLF
JMP BACK
INT 3H
CODE ENDS
END START
```

GENERATE TRIANGULAR WAVE OF 0 TO 5V

ALGORITHM

1. SEND A VALUE CORRESPONDING TO 0V ON PORT A
2. INCREMENT THE VALUE BY 1 AND KEEP SENDING IT TILL IT REACHES HIGH VOLTAGE
3. DECREMENT THE VALUE BY 1 AND KEEP SENDING IT TILL VALLU REACHES 0VOLT
4. INCREMENT AGAIN AND REPEAT THE CYCLE INDIFINITELY
5. BINARY VALUE FOR 0V = 00H
6. BINARY VALUE FOR 5V =FFH

PROGRAM

```
DADA SEGMENT
PORT EQU 8000H
CNTPRT EQU 8003H
DELAYH EQU 147
DELAYL EQU 441
LVOLT DB 00H
HVOLT DB 0FFH
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START:      MOV AX,DATA
MOV DS,AX
MOV AL,80H
MOV DX,CNTPRT
OUT DX,AL
MOV AL,LVOLT
MOV DX,PORTA
BACK:       OUT DX,AL
INC AL
CMP AL,HVOLT
JNZ BACK
BK:OUT DX,AL
DEC AL
CMP AL,LVOLT
JNZ BK
JMP BACK
INT 3H
CODE ENDS
END START
```

GENERATE STAIRCASE WAVE WITH THE FOLLOWING SPECIFICATIONS
NUM.OF STEPS = 5
HEIGHT OF STEP = 1VOLT
WIDTH OF STEP = 5MILLI SEC

ALGORITHM

1. **SEND A VALUE OF 0 CORRESPONDING TO 0 VOLTS TO PORT A**
2. **GIVE DELAY OF 5 MILLI SEC**
3. **CALCULATE NEXT VALUE BY ADDING STEP HEIGHT**
4. **SEND IT TO PORT A AND DELAY AGAIN**
5. **REPEAT THIS TILL ALL STEPS ARE OVER**
6. **CONTINUE THE CYCLE INDIFINITELY**

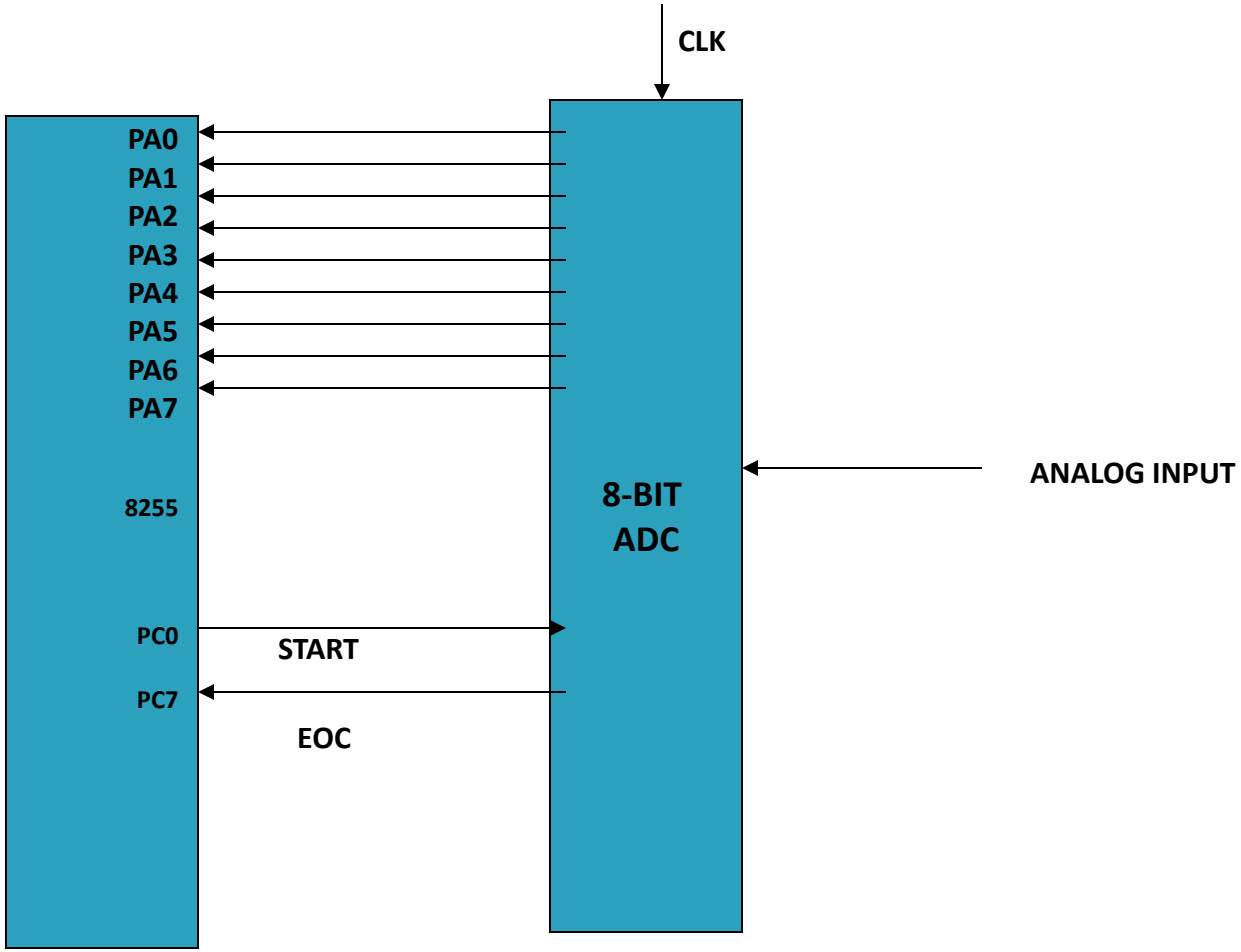
DEALY CALCULATIONS

3.4 MICRO SEC X DELAY CONSTANT = 5000 MICRO SEC
DELAY CONSTANT = 5000 MICRO SEC/ 3.4 = 1470
STEP HEIGHT = 1 VOLT = FF/5 H = 255 / 5 = 51 = 33H
(LVOLT) LOW VALUE = 0H
HVOLT HIGH VALUE = 0FFH

PROGRAM

```
DATA SEGMENT
PORTA EQU 8000H
CNTPRT EQU 8003H
LVOLT EQU 0H
HVOLT DB 0FFH
STEPH DB 33H
STEPCNT DB 06H ; NO.OF STEPS PLUS ONE = STEPCOUNT
DELAY EQU 1470
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START:    MOV AX,DATA
MOV DS,AX
MOV AL,80H
MOV DX,CNTPRT
OUT DX,AL
MOV AL,LVOLT
MOV DX,PORTA
BEGIN:    MOV BL,STEPCNT
MOV AL,00H
BACK:     OUT DX,AL
MOV CX,DELAY
SELF:     LOOP SELF
ADD AL,STEPH
DEC BL
JNZ BACK
JMP BEGIN
INT 3H
CODE ENDS
END START
```

Analog to Digital Converter



WRITE A PROGRAM FOR 8-BIT ADC TO SAMPLE ANALOG INPUT AND STORE THE DIGITAL VALUE IN MEMORY

ALGORITHM

1. SEND THE START PULSE TO ADC
2. WAIT FOR EOC TO BECOME ACTIVE
3. READ THE DATA FROM ADC AND STORE IT IN MEMORY

MD=98H

PCBSR = 00 (RESET)/ 01(SET)

```
DATA SEGMENT
PORTA EQU 0FFE0H
PORTC EQU 0FFE4H
CNTPRT EQU 0FFE6H
MEM DW 2000H
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START:    MOV AX,DATA
MOV DS,AX
MOV DX,CNTPRT
MOV AL,98H
OUT DX,AL
MOV AL,01H
OUT DX,AL
MOV AL,00
OUT DX,AL
MOV DX,PORTC
CHK:      IN AL,DX
AND AL,80H
JZ CHK
MOV DX,PORTA
IN AL,DX
MOV MEM,AL
INT 3H
CODE ENDS
END START
```

Scope of research

- Here we can developed the technique which required lesser hardware to interconnect microprocessor and and seven segment LED, and we can also design a such LED which has less then seven segments and capable to show all the letters and numbers.