CMR
INSTITUTE OF
TECHNOLOGY

USN □□□□□□□□□□□

Internal Assesment Test –II

Sub:	MICROPROCESSOR							Code:	10EC62
Date:	08 / 05 / 2017	Duration:	90 mins	Max Marks:	50	Sem:	VI	Branch:	ECE/TCE

Answer Any FIVE FULL Questions

		Marks	OBE	
			CO	RBT
1 (a)	Write a macro to i) Read a character from keyboard ii) Read a string from keyboard iii) Display a string on console iv) To detect a key press from keyboard	[10]	CO2	L3
2	Explain different operational modes of 82C55 PPI.	[10]	CO5	L2
3	Explain the interfacing diagram of 4x4 matrix keyboard to 8086 using 8255. Write the flow chart to detect a key press on a 4x4 matrix keyboard.	[10]	CO5	L3
4 (a)	What are assembler directives ?Explain following assembler directives with an example for each: i) ALIGN, ii) ASSUME, iii) OFFSET, iv) LENGTH	[05]	CO2	L2
(b)	Explain all the string primitives.	[05]	CO2	L2
5	Interface four 7-segment display using 8255 to 8086.Write an ALP to display the digits 5, 6, 7, 8.	[10]	CO5	L3
6	Write an ALP to read a string from keyboard and check the string is palindrome or not. Also display the reversed string on monitor screen.	[10]	CO2	L3
7	Write an ALP to convert binary to ASCII coded BCD number.	[10]	CO2	L3

(1)(a) Write a macro to.

(i) Read a character from keyboard
(ii) Read a string from keyboard
(iii) Display a string on console
(iv) To detect a key press from key board — 10 marks

→ (i) To read a character from keyboard.

```
~~RDchar Macro~~
RDCHAR  MACRO
    mov AH, 01
     INT 21H ; ASCII value of pressed key is in AL.
                                        — 2.5 marks
  ENDM
```

(ii) To read a string from keyboard.

```
RDSTR MACRO ~~opr~~ buf
   LEA DX, buf.
    mov AH,0AH.
     INT 21H.                          — 2.5 marks
   ENDM.
```

(iii) To display a string on console

```
disp  MACRO  str
   LEA DX, Str.
    mov AH,09H
     INT 21H.                          — 2.5 marks
  ENDM.
```

(iv) To detect a keypress from keyboard.

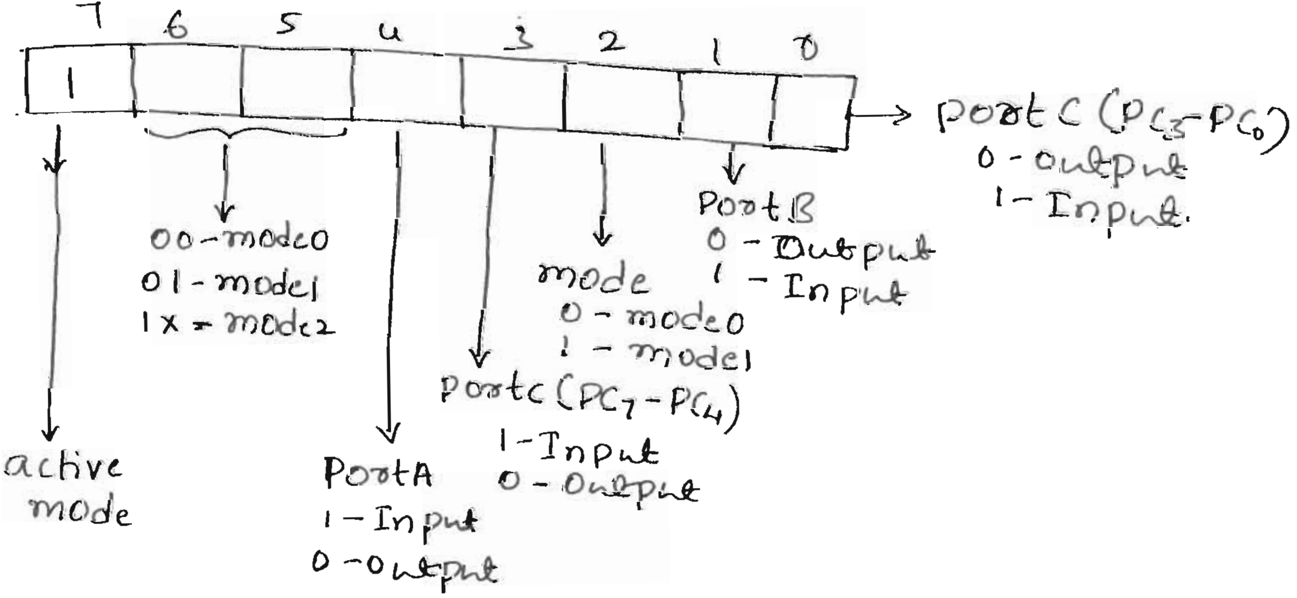
```
detect  MACRO          LOCAL ~~skip~~ .no-key
        MOV AH,0BH
        INT 21H.   ; if key is pressed
        ~~CMP AL,0FFH~~.   AL=0FF otherwise 00H.
        ~~JE nokey~~
        ~~MOV BL,0~~.
ENDM.
```

— 2.5 marks

(2) Explain the different operational modes of 82C55 PPI

-10 marks

→ Control word Register formats of 82C55.



When MSB of Controlword register is set to 1, then 82C55 operates in active mode. In this configuration ports can operate in mode 0, mode1, ~~and~~ or mode2.. ~~op~~. Port A (PA7 - PA0) together with PC (PC7 - PC4) are called as GroupA

~~PortsB (PB~~

Port B (PB7 - PB0) together with PC (PC3 - PC0) are called as GroupB.

Mode 0 operation: Mode 0 operation causes 82C55 to function either as a buffered input device or as a latched output device.

~~PortA can operat~~

portA and Port B can acts as either ~~port~~ output or input ports.

Port C can act as either output or input port with 8 lines (PC7-PC0]. otherwise Port C can be divided into two halves as Port C lower (PC3-PC0), Port C upper(PC7-PC4). Each halves can be configured as either input or output ports

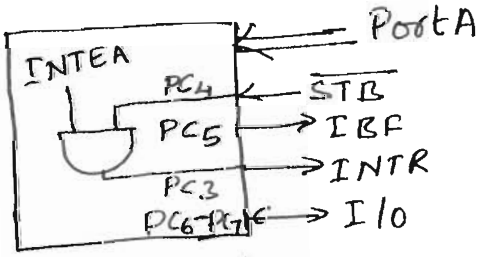
— 02marks

Mode 1 operation:

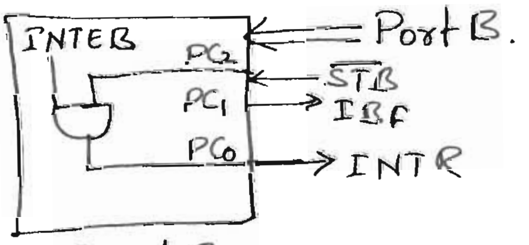
In this mode PortA or Port B acts as either strobed input or strobed output ports.

~~PortCupper (PC7-PC4~~

Port C lines are used to carry handshaking signals for strobed input or output operation

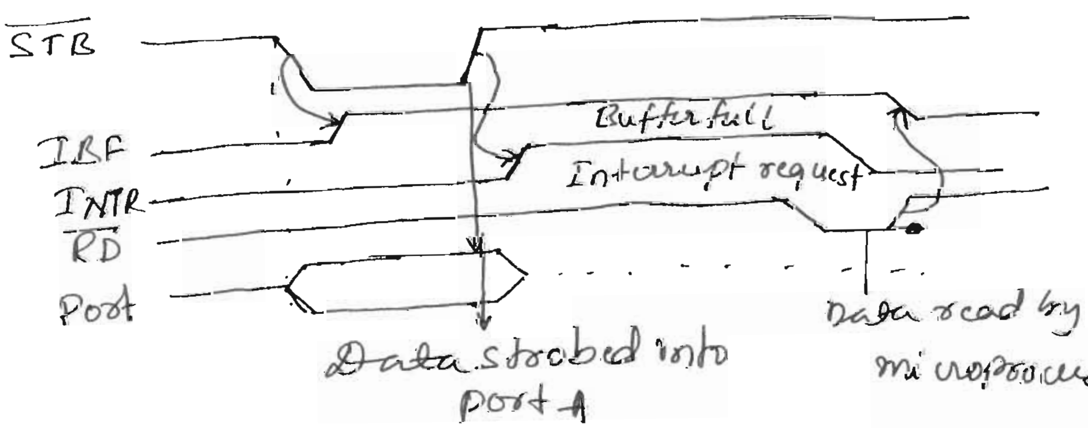


Model: PortA



PortB

strobed INput

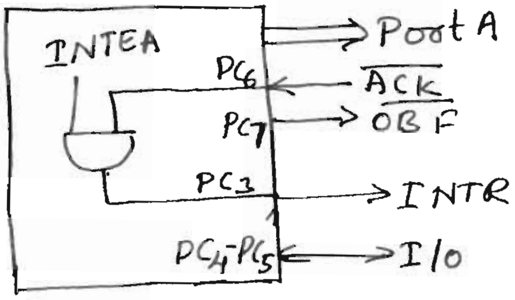


During model strobed input operation :

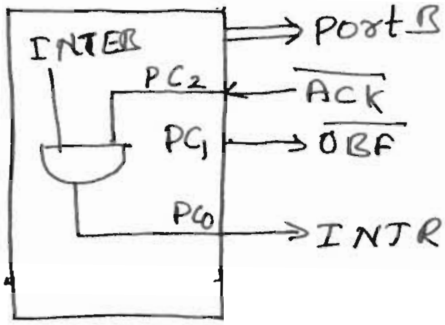
Strobe input loads data into theport latch. IBF (Input buffer full) is an output indicates the input latchhas the data.

Interrupt request (INTR) is an output requests an interrupt And Interrupt is cleared when ~~th~~ the data are input from the port by microprocessor. ~~Re~~ INTE(Interrupt enable) is niether an input nor an output programmed using PC_4 or PC_2. PC_7 and PC_6 are general purpose I/O pins.

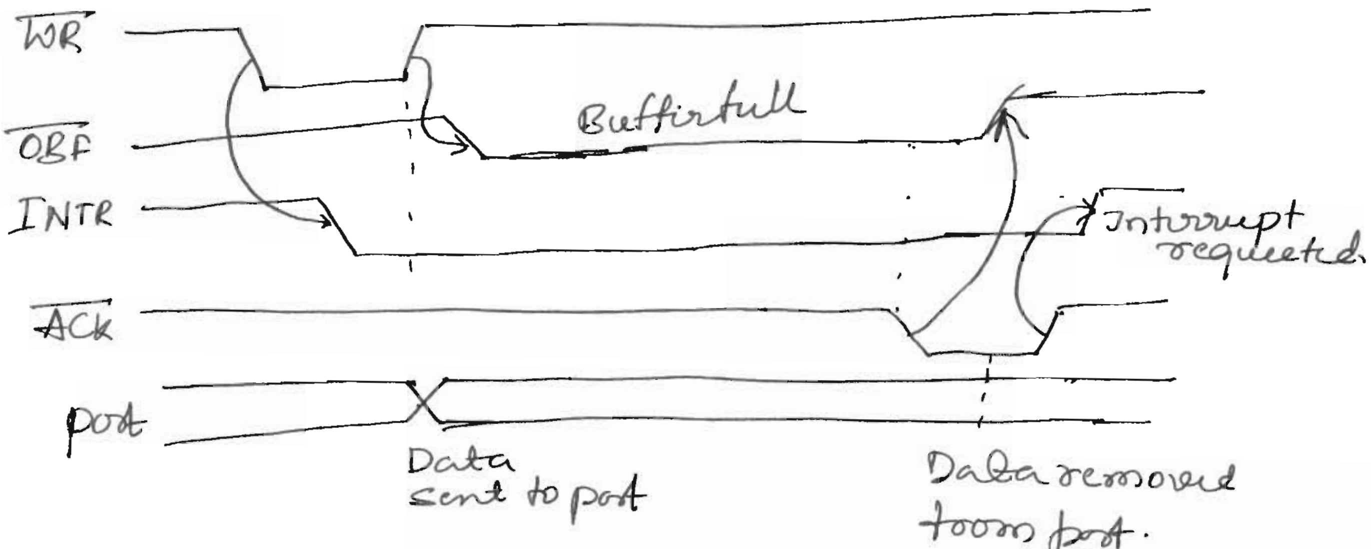
Strobed output:



Port A



port B

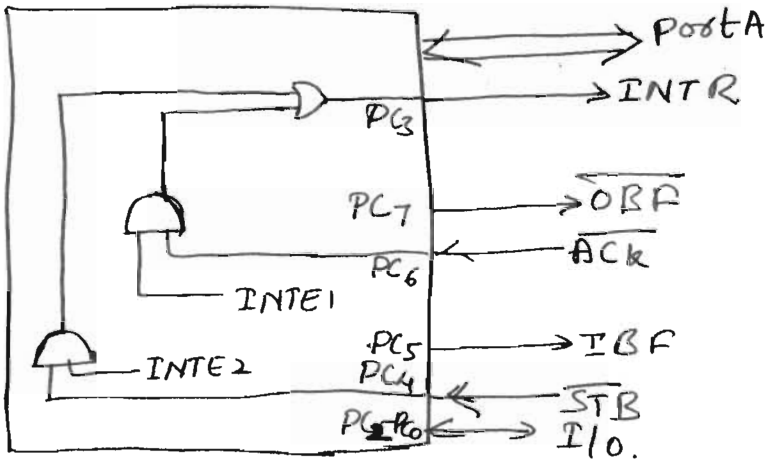


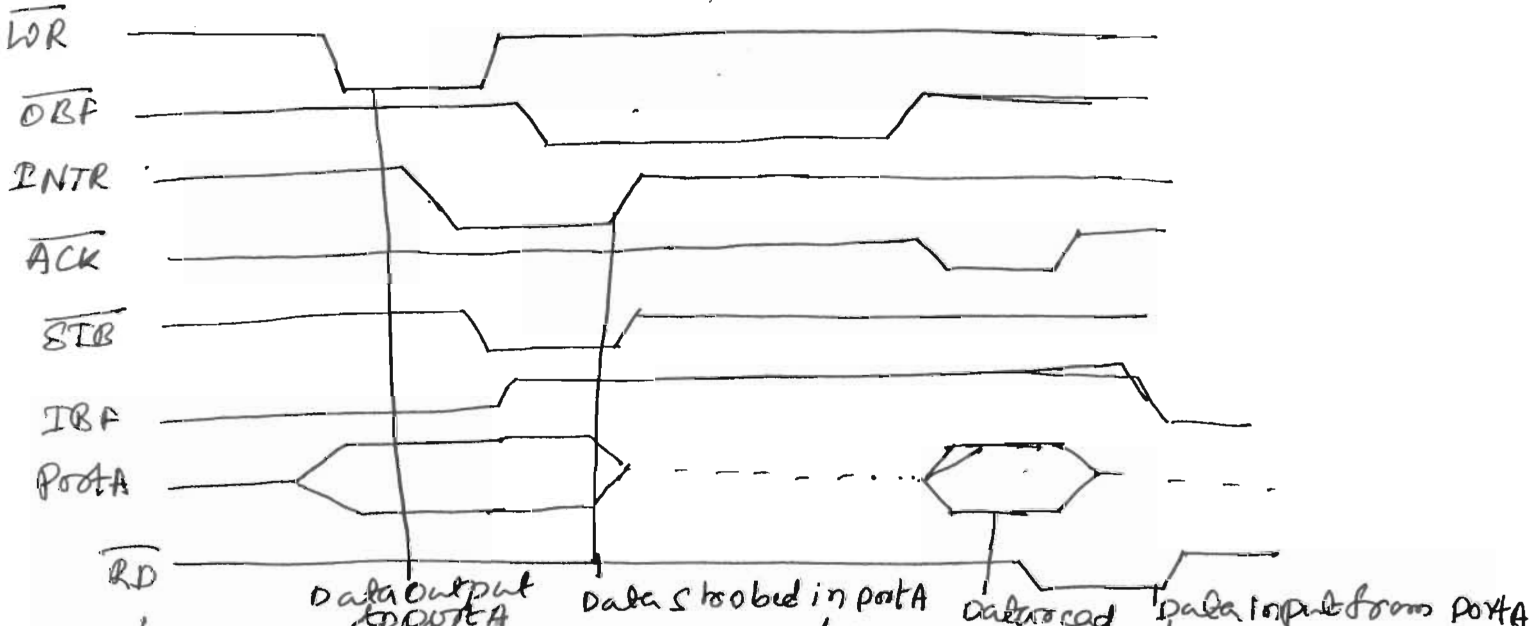
$\overline{OBF}$ (output buffer full) goes low whenever data are output to the portA or portB. ~~when~~ $\overline{ACK}$ signal is a response from an external device indicating that it has received the data from 8255 port. $\overline{ACK}$ causes $\overline{OBF}$ Pinto return to logic '1' level. INTR (Interrupt request) signal interrupts the micropro-cessor when the external device receives the data via $\overline{ACK}$ signal.

(05marks)

Mode 2 operation : Bidirectional operation

Mode 2 operation is allowed with Group A only, port A Becomes bidirectional, allowing data to be transmitted and received over the same eight wires.



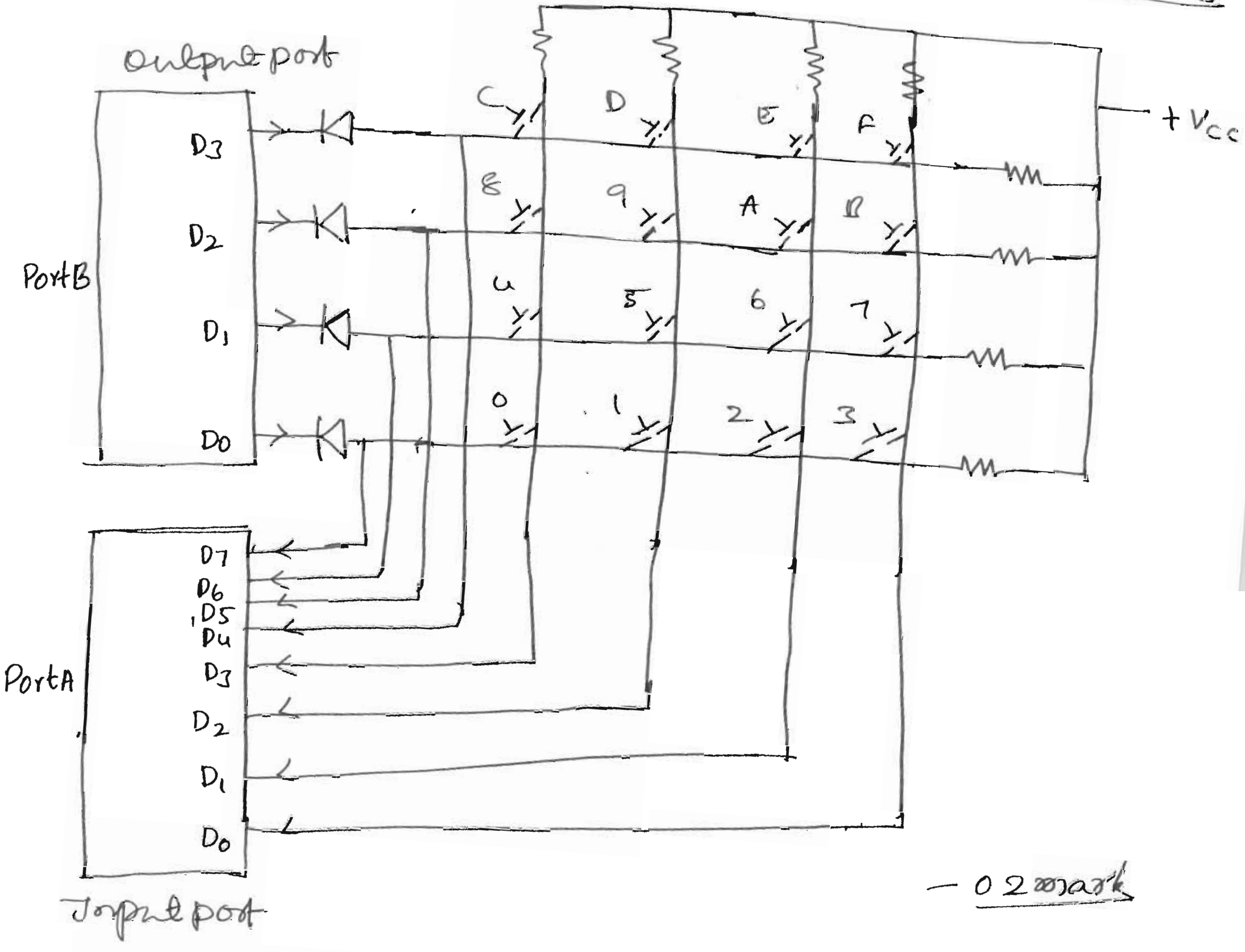


mode 2 ~~uses the same~~ combines strobed input and Output operations of Port A to use it as Bidirectional port.

03 marks

③ Explain the interfacing diagram of 4x4 matrix keyboard to 8086 using 8255. Write the flow chart to detect a keypress on a 4x4 matrix keyboard.

— 10 marks



— 02 mark

The 4X4 matrix keyboard is as shown in above diagram. There are 4 row line connected to Lower ~~port~~ 4 port lines of Port B (PB3-PB0) and Four ~~lines are~~ row lines are tapped and along with four column lines of keyboard are connected to Port A.

Port B is an output port and Port A acts as input port.

To detect a key press, zeros are sent to all rows and columns are read. If one of the column line is zero then a it ~~assu~~ decides that a key is pressed. Otherwise it waits for the key to be pressed. If a key is pressed then wait for ~~sometime to~~ debounce of the key pressed. If key is still pressed then ~~so~~ output zero to one of the row and scan the column ~~to check~~ ~~what~~ for zero to check the key press in that particular column. If key is not pressed then send zero to the next row and scan the column for ~~o~~ zero in that column to detect keypress. Continue until ~~a key~~ the pressed key is detected. And ~~e~~ decode the key ~~to~~ as a BCD code as shown in figure

If key "6" is pressed then to deetect the key '6' send zero to D_1 row connected to line D_1. If Zero received in the lines D_6 and D_1 then it encondes that key as '6'.

To detect ~~6~~ key '6'

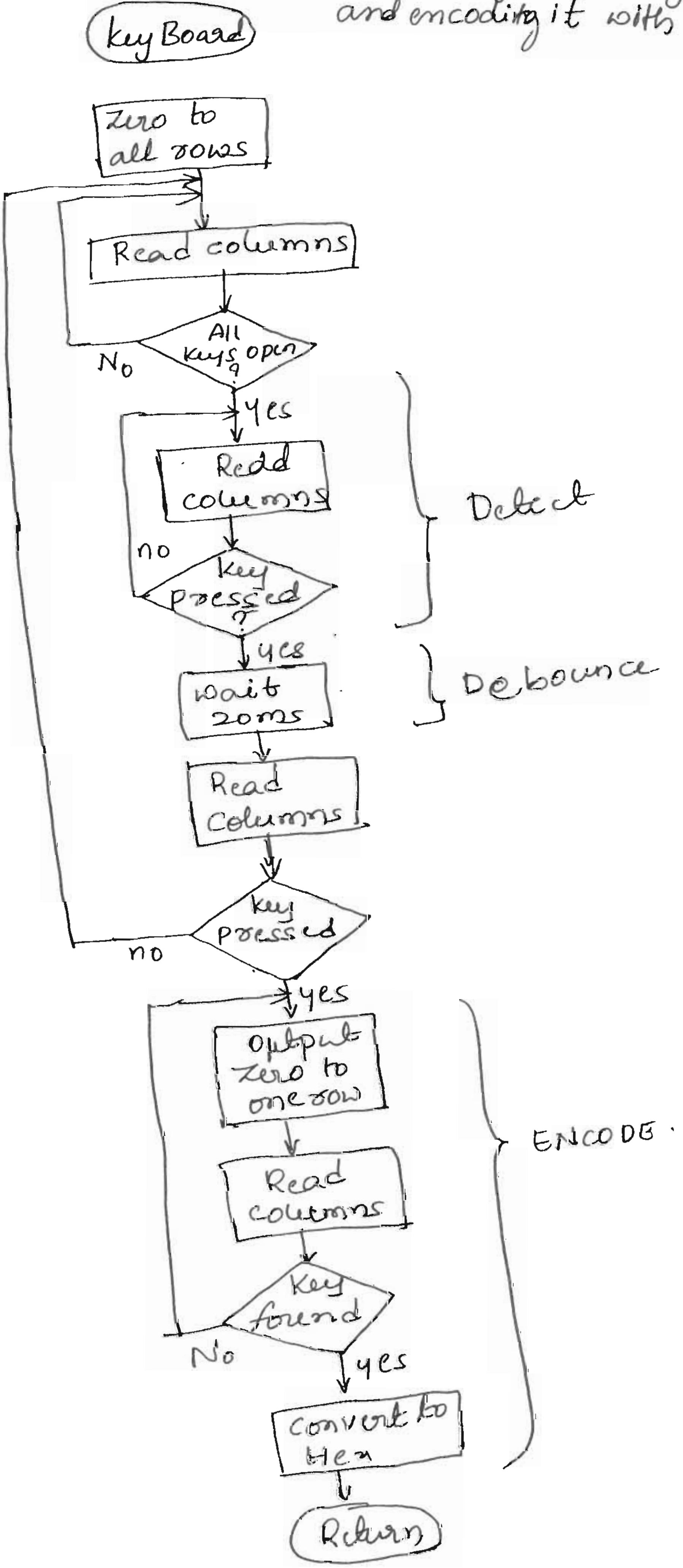
make ~~Send~~ $D_3 D_2 D_1 D_0$ of PortB as = 1101B

Then scan the lines of port A for = 10111101

—04 marks

Flow chart:

Flow chart to detect a matrix keyboard keypress, debounce, and encoding it with microcomputer



(4)(a) What are assembler directives? Explain following assembler directives with an example for each
(i) ALIGN, (ii) ASSUME, (iii) OFFSET, (iv) LENGTH. - 05 marks

→ (a) Assembler directives are pseudo operations that control the assembly process. They indicate how an operand or section of program is to be processed by the assembler. — 1 mark

(i) ALIGN: ~~Sea~~ Align to boundary

Ex: Align 2 → Aligns the data to word boundary. address divisible by 2. — 1 mark

(ii) ASSUME: Informs the assembler to ~~assign~~ name each segment

∴ ASSUME ~~se~~: CS: code1, DS: data1.
assigns name 'code1' to codesegment & 'data1' to data segment — 1 mark.

(iii) OFFSET: Specifies an offset address

mov BX, offset num
Returns an offset address to BX register. pointed by label num. — 1 mark

(iv) LENGTH: Returns number of units assigned to a variable

Ex: FEES DW 100 DUP(0)
would cause 100 words to be ~~assigned~~ associated with the variable ~~with~~ ~~F~~ FEES & the statement
MOV CX, LENGTH FEES.
would be equivalent to
MOV CX,100. 1-mark

(4)(b) Explain all the string primitives.

(i) LODS SRC/LODSB/LODSW

for Byte $(AL) \leftarrow ((SI))$, $(SI) \leftarrow (SI) \pm 1$

for word $(AX) \leftarrow ((SI))$, $(SI) \leftarrow (SI) \pm 2$.

(ii) STOS DST/STOSB/STOSW

for byte $(AL) \rightarrow ((DI))$, $(DI) \leftarrow (DI) \pm 1$

for word $(AX) \rightarrow ((DI))$, $(DI) \leftarrow (DI) \pm 2$

(iii) MOVS DST, SRC/MOVSB/MOVSW

for byte $((DI)) \leftarrow ((SI))$, $(SI) \leftarrow (SI) \pm 1$
$(DI) \leftarrow (DI) \pm 1$

for word $((DI)) \leftarrow ((SI))$, $(SI) \leftarrow (SI) \pm 2$
$(DI) \leftarrow (DI) \pm 2$.

(iv) CMBSRC, DST/CMPSB/SMPSW

~~(SRC), (DST)~~

$((SI)) - ((DI))$ flags will be affected.

for byte $(SI) \leftarrow (SI) \pm 1$, $(DI) \leftarrow (DI) \pm 1$

for word $(DI) \leftarrow (DI) \pm 2$, $(SI) \leftarrow (SI) \pm 2$

(v) SCAS DST/SCASB/SCASW

$(AL) - ((DI))$ flags will be affected

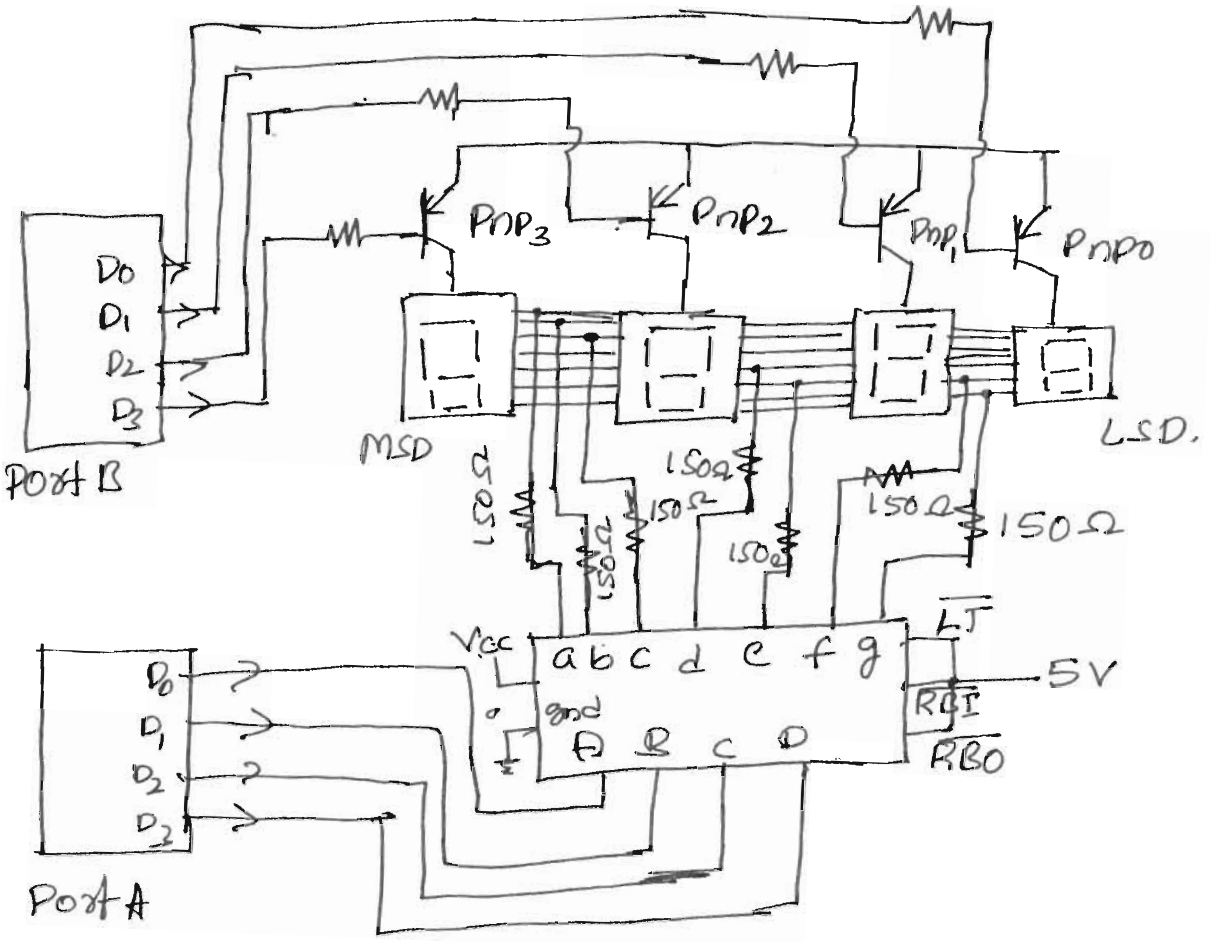
for byte $(DI) \leftarrow (DI) \pm 1$

for word $(DI) \leftarrow (DI) \pm 2$

1 for each = 1 × 5 = 5 mark

(5) Interface 4-7-Segment display using 8255 with 8086, write an ALP to display the digits 5.6.7.8

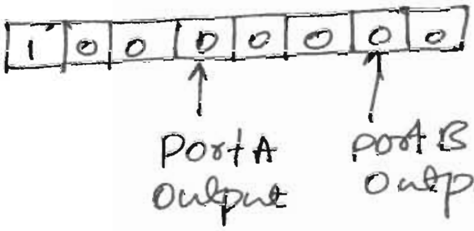
—10 marks



```
Data1 Segment
 disp db 5,6,7,8
 Data1 Ends
 stack1 Segment
      dw 100 DUP(0)
   TOS label word.
  stack1 Ends
  code1 Segment
  ASSUME  DS:Data1, CS: code1, SS:stack1.
   mov AX, Data1
    mov DS, AX
    mov AX, stack1
    mov SS, AX
mov SP, offset TOS.
 mov AL, 80H.
 mov DX, 0E883H ; controlwo
                ; -rd register
                ; address

Out DX, AL.
call display
  nop
display PROC Near.
    mov BL, 77H.
backl: mov CX, 04H.
        mov SI, 00.
 back: mov AL, BL
        ~~mov DX, 0E88~~
         mov DX, 0E881H ; port B address
        out DX, AL
         mov AL, ~~[illegible]~~ disp[SI]
          mov DX, 0E880H ; port A address
           out DX, AL.
           Call delay
            ROR BL, 1
           INC SI
           loob back
            Jmp backl
            RET
       display ENDP
      Delay PROC Near
        push CX
        push F
        ~~push BX~~
        mov CX, 1022H. ; for 10ms delay
back2: loop back2
        POPF
        POPCX
        RET
    delay ENDP
    code1 ENDS
       END.
```

control word Register



$(10000000)_2 = (80)_{16}$

⑥ Write an ALP to read a string from keyboard & check the string is palindrome or not. Also display the reversed string on the monitor screen

10 marks

→

```
.model small
.stack 64h
.data
  ~~rd str db~~
  ~~count db 05~~
  ~~revstr db~~

          N Equ 05
          rdstr db N, N+1 DUP(0), '$',
                              0DH,0AH.
          .revstr db N dup(0)

.code
~~ma~~ start: mov AX, @data
        mov DS, AX
        mov CX, ~~count~~ N
        LEA SI, rdstr+2.
        LEA DI, revstr            — 02 marks
        ADD DI, N-1
        ~~mov~~ LEA DX, rdstr  }
          mov AH, 0AH          }  — 02 marks
          INT 21H

  back:   movSB                    JmpSkip1
          DEC DI                   skip: mov AL,00h
          DEC DI                   skip1: movAH,4Ch
          Loop back.                      INT21h
          mov CX, N                   End start.
          LEA SI, rdstr+2
          LEA DI, revstr.             02 marks
        rcpE compSB
          cmp CX, 00
          JE skip
          ~~mov AL, 01~~
          mov AH,09h
        LEA ~~mov~~ DX, revstr.
          INT21 H.
```

04 marks

⑥ Write an ALP to read a string from keyboard & check the string is palindrome or not. Also display the reversed string on the monitor screen

10 marks

```
→  .model small
   .stack 64h
   .data
     ~~rd str db~~
     ~~count db 05~~
     ~~revstr db~~
                         N Equ 05
                         rdstr db N, N+1 DUP(0),'$',
                                             ODH,OAH.
                         .revstr db N dup(0)
   .code
   ~~ma~~ start: mov AX,@data
              mov DS, AX
              mov CX, ~~count~~ N
              LEA SI, rdstr+2.
              LEA DI, revstr          — 02 marks
              ADD DI, N-1
              ~~mov~~ LEA DX, rdstr }
                 mov AH,0AH         } — 02 marks
                 INT 21H
         back: movSB
               DEC DI
               DEC DI
               Loop back.
               mov CX, N
04 marks       LEA SI, rdstr+2
               LEA DI, revstr.
              repE compSB
                 cmp CX,00
                  JE Skip
                 ~~mov AH,09~~
                 mov AH,09h
              LEA ~~mov~~ DX, revstr.
                 INT21H.

         JmpSkip1
         skip: mov AL,00h
         Skip1: mov AH,4Ch
                   INT21h
               End start.

               02 marks
```

⑦ Write an ALP to convert binary to ASCII coded BCD number — 10 marks

```
.model small
.stack 64h
~~.da~~
.data
 num dw 0FFFFH
 res db 5dup(0)                       — 03 marks
 .code
 start: mov AX, data
         mov DS, AX
         mov CX, 5
     mov ~~XOR~~ DI, ~~[illegible]~~ 04
         mov AX, num
         mov BX, 10
   again: XOR DX, DX
          DIV BX
          ADD DL, 30
          ~~mov ASCII BCD[DI], DL.~~
           ~~INC DI~~
           ~~loop again~~
          mov res[DI], DL
              DEC DI
             loop again
             mov AH, 4ch
              INT 21h
             End start.                — 7 marks
```

1.