HARDWARE/INTERFACING PROGRAMS

LOGIC CONTROLLER

Read the status of eight input bits from the Logic Controller Interface and display 'FF' if it is even parity bits otherwise displays 00. Also display number of 1's in the input data.

```
.MODEL SMALL
.STACK 64                 ;sets the size for stack segment
.CODE
MOV AL,82H                ;to initialize control word
MOV DX,1093H            ;03 for control word
OUT DX,AL
MOV DX,1091H            ;01 for port b
IN AL,DX                  ;input from port, reads 8-bit value from dx
MOV CX,08H
MOV BL,00H                ;to have the count value of number of 1's
AGAIN:SHR AL,01H         ;shift right by 1
     JNC SKIP                       ;these three lines are used to check the
     INC BL                         ;number of 1's in input
SKIP:LOOP AGAIN
    MOV CL,BL
    SHR CL,01H
    JC ODD_P               ;jump if carry
    MOV AL,0FFH
    JMP HERE
ODD_P:XOR AL,AL             ;xor is performed and al will contain result
HERE:MOV DX,1090H     ;00 for port A, for output
    OUT DX,AL
    MOV AL,BL                ;number of 1's in input will be the output
    OUT DX,AL
    MOV AH,4CH
    INT 21H
    END
```

OUTPUT:

1. Even parity: all LEDs in the o/p port will be **ON** (ffh) or odd parity: all LEDs in the o/p port will be **OFF** (00h).
2. Displays total number of 1's on.

Perform the following functions using the Logic Controller Interface.

i. BCD up-down Counter

```
.MODEL SMALL
.CODE
MOV DX,1093H          ;03 for initializing control word
MOV AL,82H            ;initializing control word
OUT DX,AL             ;output to port, writes to accumulator
```

```
MOV AL,0
MOV DX,1090H            ;00 for port a

UP:OUT DX,AL            ;up counter procedure
  ADD AL,1              ;decimal adjust for addition, converts to packed decimals digits
   CMP AL,99H
  JNE UP                         ;jump if not equal

DOWN:OUT DX,AL         ;down counter procedure
    ADD AL,99H
    DAA
    CMP AL,99H
    JNE DOWN            ;jump if not equal
    MOV AH,4CH
    INT 21H
    END
```

OUTPUT:

1) Up counter-displays 00 to 99 in decimal at the o/p port.
2) Down counter-displays 99 to 00 in decimal at o/p port.

ii. Ring Counter

```
.MODEL SMALL
.CODE
MOV DX,1093H            ;03 for initializing control word
MOV AL,82H              ;control word initializing
OUT DX,AL
MOV DX,1090H            ;00 for port a
MOV AL,80H

AGAIN:OUT DX,AL
    ROR AL,01H
    JMP AGAIN           ;unconditional jump to label "AGAIN"
    MOV AH,4CH
    INT 21H
    END
```

OUTPUT:

O/P shifts one bit at a time to the right at the output port.

Read the status of two 8-bit inputs (X & Y) from the Logic Controller Interface and display X*Y.

```
.MODEL SMALL

.CODE
MOV AL,82H              ;initializing control word
MOV DX,1093H          ;03 to initalize control word
OUT DX,AL
MOV DX,1091H          ;01 for port b
IN AL,DX                   ;inputs from accumulator
MOV BL,AL
MOV DX,1091H         ;01 for port b
IN AL,DX
MUL BL                    ;multiplies contents of bl with al
MOV DX,1090H
OUT DX,AL
MOV AH,4CH
INT 21H
END
```

OUTPUT:

The 2 inputs given in hexa (answer<255) are multiplied and the result is received in hexa at the o/p port.

SEVEN-SEGMENT DISPLAY

Display messages FIRE and HELP alternately with flickering effects on a 7-segment display interface for a suitable period of time. Ensure a flashing rate that makes it easy to read both the messages.

```
.model small
.data
msg1 db 86h,0b8h,0cfh,8eh
msg2 db 8ch,0c7h,86h,89h
.code
mov ax,@data
mov ds,ax
mov dx,1093h
mov al,80h
out dx,al
mov ch,04
lea si,msg1
ag:call disp
call delay
inc si
```

```
dec ch
jnz ag
mov ch,04
lea si,msg2
ag5:call disp
call delay
inc si
dec ch
jnz ag5
mov ah,4ch
int 21h
disp proc
mov cl,08
mov bl,[si]
ag1:rol bl,01
mov al,bl
mov dx,1091h
out dx,al
mov dx,1092h
mov al,01
out dx,al
dec al
out dx,al
dec cl
jnz ag1
ret
disp endp
delay proc
push bx
push si
mov bx,0ffffh
nxt1:mov si,0ffffh
nxt:dec si
jnz nxt
dec bx
jnz nxt1
pop si
pop bx
ret
delay endp
end
```

OUTPUT:

Displays “**FIRE**” and “**HELP**” alternatively with flickering effects on seven-segment display interface

STEPPER MOTOR INTERFACE

Drive a Stepper Motor interface to rotate the motor in clockwise direction by N steps (N is specified by the examiner). Introduce suitable delay between successive steps.

```
.MODEL SMALL
.DATA
 N DW 200d                      ;200d=1 rotation
.CODE
 MOV AX,@DATA
 MOV DS,AX
 MOV DX,1093H          ;to initialize control word
 MOV AL,80H
 OUT DX,AL
 MOV DX,1090H
 MOV AL,77H            ; u can use 88H also
 MOV CX,N
 AGAIN:OUT DX,AL
      CALL DELAY
 ROR AL,01
 LOOP AGAIN
 MOV AH,4CH
 INT 21H
 DELAY PROC
        PUSH BX
        PUSH SI
        MOV BX,0FFFH
    NXT1: MOV SI,0FFFH
    NXT: DEC SI
        JNZ NXT
        DEC BX
        JNZ NXT1
        POP SI
        POP BX
        RET                ;returns control from procedure to calling procedure
  DELAY ENDP
 END
```

OUTPUT:

A stepper motor interface is driven to rotate the motor in clockwise direction (from A to C) by 200d (1 rotation) steps.

Drive a stepper motor interface to rotate the motor in anticlockwise direction by N steps (N is specified by the examiner). Introduce suitable delay between successive steps.

```
.MODEL SMALL
.DATA
 N DW 200d                                            ;200d= 1 rotation
.CODE
 MOV AX,@DATA
 MOV DS,AX
 MOV DX,1093H                             ;intialize control world
 MOV AL,80H
 OUT DX,AL
 MOV DX,1090H
 MOV AL,77H
 MOV CX,N
 AGAIN:OUT DX,AL
      CALL DELAY
 ROL AL,01
 LOOP AGAIN
 MOV AH,4CH
 INT 21H

 DELAY PROC
        PUSH BX
        PUSH SI
        MOV BX,0FFFH
    NXT1: MOV SI,0FFFH
    NXT: DEC SI
        JNZ NXT
        DEC BX
        JNZ NXT1
        POP SI
        POP BX
        RET                   ;returns control from procedure to calling procedure
        DELAY ENDP
 END
```

OUTPUT:

A stepper motor interface is driven to rotate the motor in anticlockwise direction (from C to A) by 200d (1 rotation) steps.

Drive a stepper motor interface to rotate the motor by N steps left direction and N steps right direction (N is specified by the examiner). Introduce suitable delay between successive steps.

```
.MODEL SMALL
.DATA
 N DW 200d                                      ;200d=1 rotation
.CODE
 MOV AX,@DATA
 MOV DS,AX
 MOV DX,1093H                        ;initialize control word
 MOV AL,80H
 OUT DX,AL
 MOV DX,1090H
 MOV AL,77H
 MOV CX,N
 AGAIN:OUT DX,AL                     ;for clockwise
      CALL DELAY
 ROR AL,01
 LOOP AGAIN
 mov CX,N
 HERE:OUT DX,AL                      ;for anti-clockwise
     CALL DELAY
     ROL AL,01
     LOOP HERE
 MOV AH,4CH
 INT 21H
 DELAY PROC
        PUSH BX
        PUSH SI
        MOV BX,0FFFH
    NXT1: MOV SI,0FFFH
    NXT: DEC SI
        JNZ NXT
        DEC BX
        JNZ NXT1
        POP SI
        POP BX
        RET                  ;returns control from procedure to calling procedure
        DELAY ENDP
 END
```

OUTPUT:

A stepper motor interface is driven to rotate the motor by 200d (one rotation) steps left direction and 200d steps right direction.

KEYBOARD INTERFACE

Scan a 8 x 3 keypad for key closure and to store the code of the key pressed in a memory location.

```
.model small
.data
key db ?
row db ?
col db ?
.code
mov ax,@data
mov ds,ax

mov al,90h
mov dx,1093h
out dx,al

mov ch,01               ; row
rept: mov bl,01
mov ah,00               ; key
mov cl,03
mov bh,01               ; col

nxtcol:
mov al,bl
mov dx,1092h
out dx,al

mov dx,1090h
in al,dx
cmp al,0
jnz keyp
add ah,08
rol bl,01
inc bh
dec cl
jnz nxtcol
jmp rept

keyp:ror al,01
jc here
inc ah
inc ch
jmp keyp
```

```
here:
mov key,ah
mov col,bh
mov row,ch

mov ah,4ch
int 21h
end
```

OUTPUT:

The following will be stored in memory location.
The entered character is: 0
The row is: 1
The column is: 1

DAC INTERFACE

Generate the Sine Wave using DAC interface (The output of the DAC is to be displayed on the CRO).

```
.model small
.data
table db 80h,96h,0abh,0c0h,0d2h,0e2h,0eeh,0f8h,0feh,0ffh
   db 0feh,0f8h,0eeh,0e2h,0d2h,0c0h,0abh,96h,80h
   db 69h,54h,40h,2dh,1dh,11h,07h,01h,00h
   db 01h,07h,11h,1dh,2dh,40h,54h,69h,80h      ;values of the points in HEXA
msg db 10,13,'pres any key to terminate$'
.code
mov ax,@data
mov ds,ax
mov dx,1093h             ;initialize control world
mov al,80h
out dx,al
lea dx,msg                    ;displays msg
mov ah,09h
int 21h
rest: mov cx,25h              ;values are moved into cx
   lea si,table
ag: mov al,[si]
  mov dx,1090h                ;for port A
  out dx,al
  inc si
  loop ag
  mov ah,1            ;reads from keyboard,doesn't wait for d key to be pressed
  int 16h
```

```
   jz rest
   mov ah,4ch
   int 21h
end
```

OUTPUT:

A sine waveform (varies from 00h to ffh) is generated using DAC interface and is displayed on CRO.

Generate a Half Rectified Sine wave form using the DAC interface. (The output of the DAC is to be displayed on the CRO).

```
.model small
.data
table db 80h,96h,0abh,0c0h,0d2h,0e2h,0eeh,0f8h,0feh,0ffh
     db 0feh,0f8h,0eeh,0e2h,0d2h,0c0h,0abh,96h,80h
     db 80h,80h,80h,80h,80h,80h,80h,80h,80h
     db 80h,80h,80h,80h,80h,80h,80h,80h,80h        ;values of the points in HEXA
msg db 10,13,'pres any key to terminate$'
.code
mov ax,@data
mov ds,ax
mov dx,1093h                                  ;initialize control world
mov al,80h
out dx,al
lea dx,msg                                    ;displays msg
mov ah,09h
int 21h
rest: mov cx,25h                              ;values are moved into cx
     lea si,table
ag: mov al,[si]
   mov dx,1090h
   out dx,al
   inc si
   loop ag
   mov ah,1          ;reads from keyboard,doesn't wait for d key to be pressed
   int 16h
   jz rest
   mov ah,4ch
   int 21h
end
```

OUTPUT:

A half rectified sine waveform (varies from 00h to ffh) is generated using DAC interface and is displayed on CRO.

Generate a Fully Rectified Sine waveform using the DAC interface. (The output of the DAC is to be displayed on the CRO).

```
.model small
.data
table db 80h,96h,0abh,0c0h,0d2h,0e2h,0eeh,0f8h,0feh,0ffh
     db 0feh,0f8h,0eeh,0e2h,0d2h,0c0h,0abh,96h,80h
           ;values of the points in HEXA
msg db 10,13,'pres any key to terminate$'
.code
mov ax,@data
mov ds,ax
mov dx,1093h                                    ;initialize control world
mov al,80h
out dx,al
lea dx,msg                                       ;displays msg
mov ah,09h
int 21h
rest: mov cx,13h                                 ;values are moved into cx
     lea si,table
ag: mov al,[si]
   mov dx,1090h
   out dx,al
   inc si
   loop ag
   mov ah,1              ;reads from keyboard,doesn't wait for d key to be pressed
   int 16h
   jz rest
   mov ah,4ch
   int 21h
end
```

OUTPUT:

A fully rectified sine waveform (varies from 00h to ffh) is generated using DAC interface and is displayed on CRO.

ELEVATOR INTERFACE

Drive an elevator interface to move the elevator from ground floor to top floor (UP) and top floor to ground floor (DOWN).

```
.model small
.code
mov dx,1093h
mov al,82h
out dx,al

mov al,0f0h
mov dx,1090h
mov cl,09
ag:out dx,al
call delay
inc al
dec cl
jnz ag
mov cl,0ah
ag1:out dx,al
call delay
dec al
dec cl
jnz ag1
mov ah,4ch
int 21h
delay proc
mov bx,0ffffh
nxt1:mov si,0ffffh
nxt:dec si
jnz nxt
dec bx
jnz nxt1
ret
delay endp
end
```