Assignment -01

Name	Samar Abbas
Sr no	- 27
En. no	-- GN2500
Fac no	--22COB255
Subject	Microprocessor theory and applications
Submitted to	Dr. Abdus Samad & Dr. Sayyed Usman Ahmed
college	- Zakir Husain of Engineering & Technology ,Aligarh Muslim University , Aligarh ,U.P

CO-1

Solution:01 Finding highest among five numbers (13h, 2ch, 63h, 58h, 5ch)

```
Assume CS: code, DS: data
DATA SEGMENT

    Array DB 13h, 2ch, 63h, 58h, 5ch
    Max DB 0

DATA ENDS

CODE SEGMENT

START:
    MOV AX, data ; load data segment address into AX
    MOV DS, AX
    MOV SI, OFFSET Array
    MOV AL, [SI] : Move the first number (13h) into AL
    MOV CX, 4    : loop counter set to 4
    INC SI       : Increament SI to point to second element

Find-MAX:
    MOV BL, [SI]
    CMP AL, BL
    JAE NEXT : IF AL ≥ BL jump to next
    MOV AL, BL : else AL ← BL

NEXT:
    INC SI : SI Points to next number
    LOOP FIND-MAX : Repeat until CX reaches zero
    MOV M, AL
    INT 21h

CODE ENDS
END START
```

Assembler directives

① Segment : Tells us in which physical segment we are working. e.g → DATA SEGMENT, CODE SEGMENT

② DB : Defines a byte-sized variable (e.g Max)

③ ENDS : Mark the end of a Segment (DATA ENDS)

④ Assume : Tells the assembler the name of the logical segment it should use for a specified segment.

⑤ MOV : Move instruction to copy data ~~flow~~ from source to destin

⑥ END : Marks the end of the program and specified the start address (END START)

Q:2 ⓐ CS : 1200H, IP : DE00H

Solution: 02 ⓐ physical address = CS X 10H + OFFSET
= 1200H X 10H + DE00H
= 12000H + DE00H
= 1FE00H

```
 12000
+ DE00
-------
 1FE00
```

ⓑ DS = 73A2H, SI = 3216H

Solution

Physical address = DS X10H + OFFSET
= 73A2H X10H + 3216H
= 73A20H + 3216H
= 76C36H

```
 73A20H
+ 3216H
--------
 76C36H
```

ⓒ CS = 7370H, IP = 561EH

Solution physical address = CS X10H + IP
= 7370H X 10H + 561EH
= 73700H + 561EH
= 78D1EH

Solution

(i) MOV BX, 2050H ⇒ BX = 2050H = 0010 0000 0101 0000 (in binary)

(ii) MOV CL, 05H ⇒ CL = 05H = 5 (decimal)

(iii) SHL BX, CL

Shifting the bits of BX to left 5 times

for 1 shift: 0100 0000 1010 0000

for 2 shift: 1000 0001 0100 0000

for 3 shift: 0000 0010 1000 0000

for 4 shift: 0000 0101 0000 0000

for 5 shift: 0000 1010 0000 0000

0 A 0 0

BX = 0000 1010 0000 0000 = 0A00H in hexadecimal

(Answer)

4. (i) Serial data into parallel using SID line of 8085

Solution

: Program to read 8 bits serially from SID and store as parallel data

: Result will be stored in memory location 2030H

```
MVI B, 08 ; Counter for 8 bits
MVI A, 00H ; Initialize Accumulator
LOOP: RIM  : Read SID bit into accumulator
      RAL  : Rotate accumulator left (bring CY to LSB)
      DCR B : decrement counter
      JNZ LOOP
      STA 2050H : store parallel data in memory
      HLT  : stop program
```

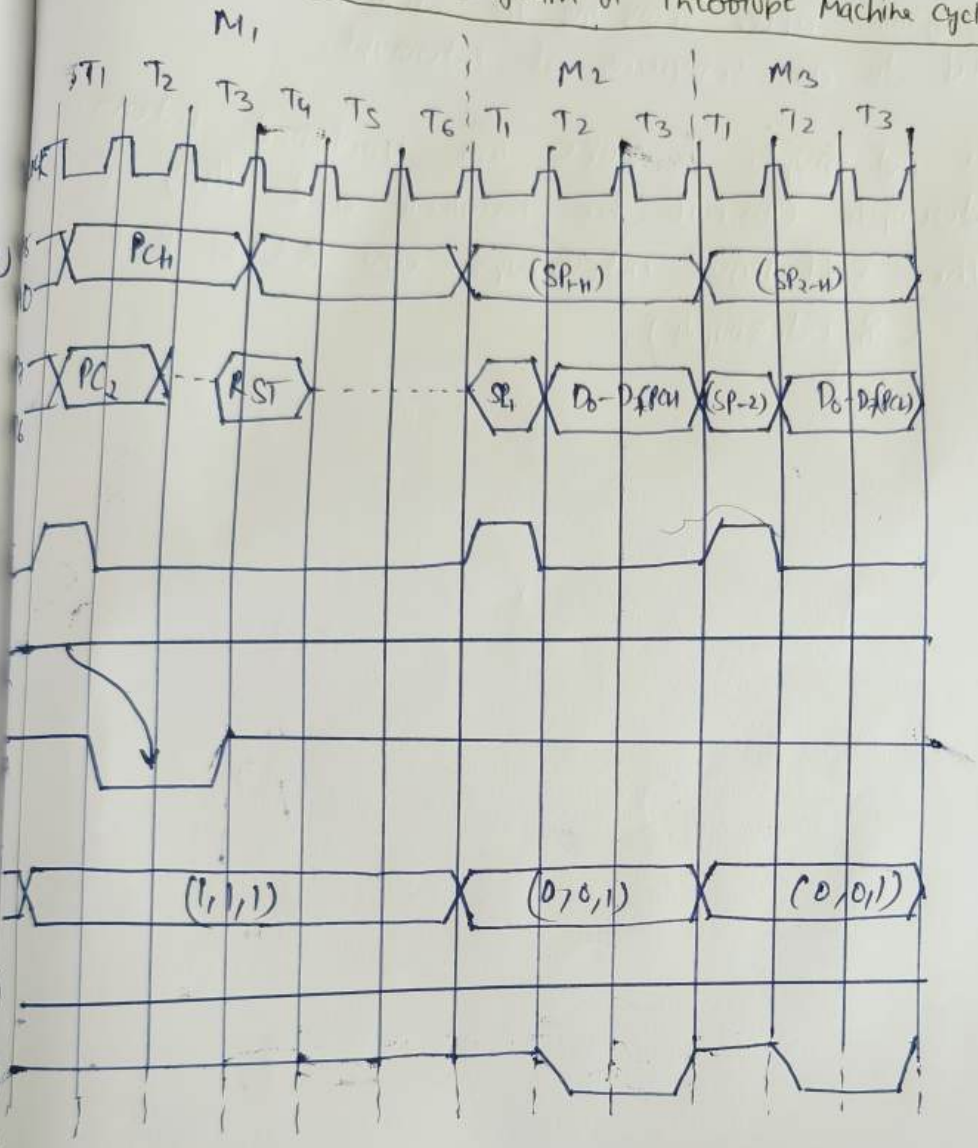
(i) Parallel data into serial and transmit into SOD line

: Data to be transmitted is stored in memory location 2050

```
LDA 2050H  :  A ← [2050]
MVI B, 08H :  Counter for 8-bit

Loop : RAL     : Rotate left accumulator (MSB goes to CY)
       MOV C,A : Save remaining bits
       MVI A,00H : clear accumulator
       INC Zero  : if CY=0, jump to Zero.
       MVI A,80H : Set SOD bit (A1=1 for SOD output)
Zero : SIM     : Output bit through SOD
       MOV A,C : restore remaining bits
       DCR B   : Decrement B Counter
       JNZ Loop : if not Zero, Continue sending bits
       HLT      : Stop program
```

Timing Diagram of Interrupt Machine cycle



Use of $RST_0 - RST_7$ interrupts :

The Software instructions are included at the appropriate (or required) e in the main program. When the processor encounters the software ructions, it pushes the content of PC (Program Counter) to Stack. en loads the Vector address in PC and start executing the interrupt rvice routine (ISR) stored in this Vector address. At the d of ISR, a return instruction — RET will be placed.

When the RET instruction is executed, the processor pop the content of stack to PC. Hence the processor control returns to the main program after servicing the interrupt. Execution of ISR is referred to as servicing of interrupt.

All Software interrupts of 8085 are vectored interrupts. The Software interrupts can not be masked and they can not be disabled. The Software interrupts are RST_0, RST_1, RST_2, RST_3 --- RST_7 (8 interrupts).