# CSE341: Microprocessor

**Quiz - 4** **Section:** **Marks: 15** **Duration: 30 minutes**

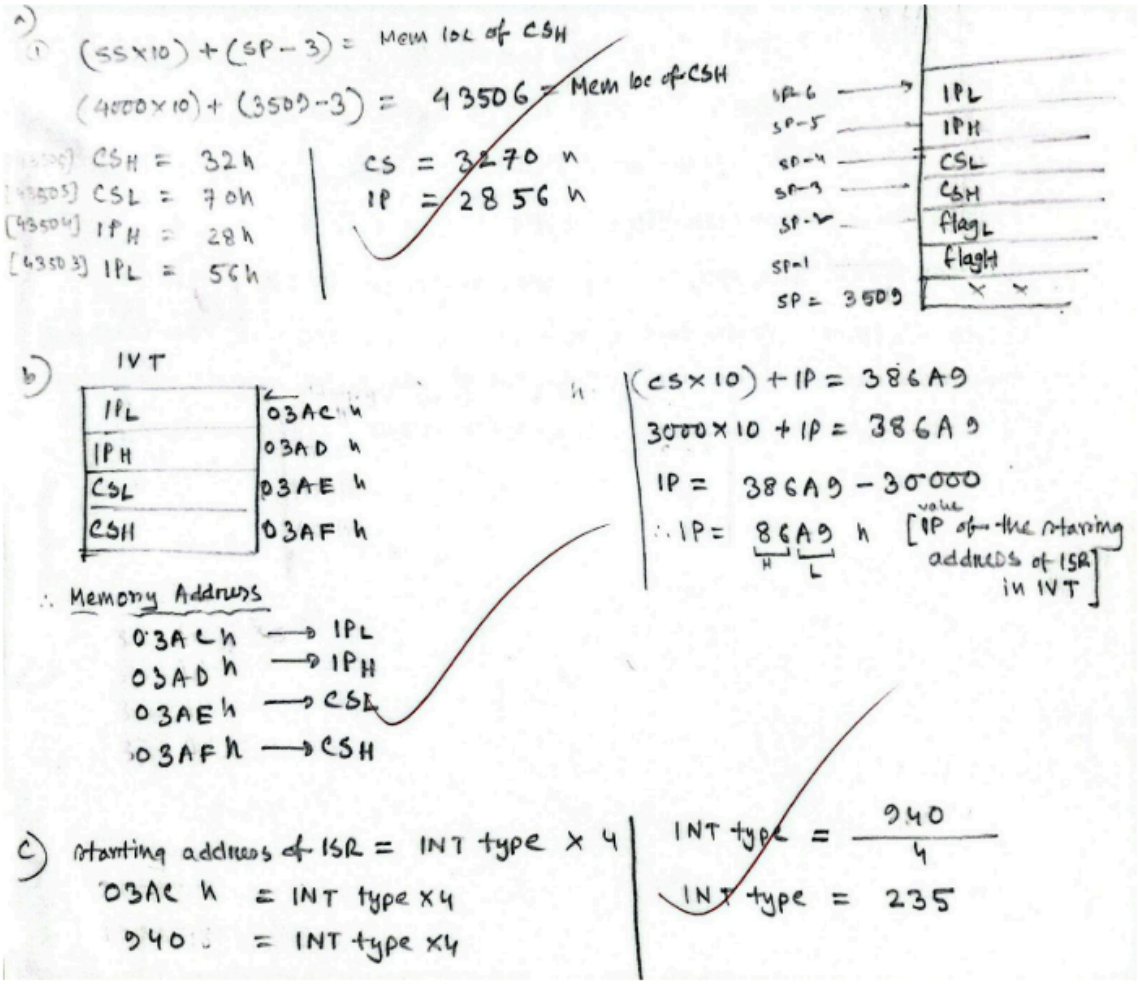
**Name:** **ID:**

**Question-1:** An 8086 microprocessor has the following segment addresses: **DS = 2000h, CS = 3000h, SS = 4000h** and while program **X** was executing, an **interrupt N** occurred. To handle this interrupt, the processor must jump to the **ISR** located at **386A9h**, which resides within the same code segment. Before fetching the ISR, the stack top offset was **3509h**. Some portions of memory are provided below:

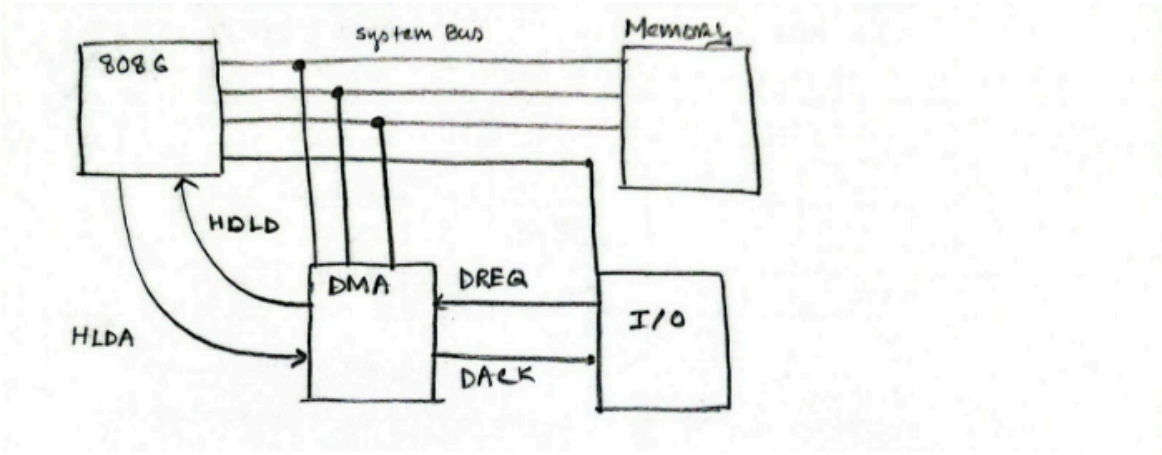
| Address | 43509h | 43508h | 43507h | 43506h | 43505h | 43504h |
|---|---|---|---|---|---|---|
| Data | 57h | 76h | 32h | 32h | 70h | 28h |

| Address | 43503h | 03AFh | 03AEh | 03ADh | 03ACh | 03ABh |
|---|---|---|---|---|---|---|
| Data | 56h | 30h | 00h | 86h | A9h | 98h |

a. Deduce the value of CS and IP of program X. [3]
b. Deduce the memory addresses or locations of the IVT table where the CS and IP of Interrupt N is situated. [3]
c. Deduce the Interrupt Type N. [3]



**Question-2:** Illustrate the way DMA operates. Draw a diagram and explain how it operates. [6]




1. CPU invokes a read/write command to the peripherals.
2. The peripherals when ready sends a DMA request (DREQ) signal to the DMA controller.
3. The DMA controller sends a Hold Request (HRQ) signal to the CPU to gain control of the system bus.
4. The CPU replies with a Hold Acknowledgement signal to the DMA giving the DMA controller total control over the system bus.
5. The DMA controller then sends a DMA Acknowledgement signal to the peripherals in reply to the previous DMA request signal.
6. Data can now be transferred between Memory and I/O with the help of the DMA controller.
7. DMA controller sends interrupt signal to the CPU when finished with data transfer.
8. The DMA disables the DMA Acknowledgement signal and the CPU disables the Hold Acknowledgement signal taking back control of the system bus.

**CSE341: Microprocessor**

**Quiz - 3** **Section:** **Marks: 15** **Duration: 25 minutes**

**Name:** **ID:**

**Question-1:** Suppose an **8086** wants to read **2 bytes** of **data** from an **odd memory location**. Now the memory is working at a **slower rate** than the 8086. Hence when reading the **1st byte of data**, the 8086 has to **wait** for an **additional 2 clock pulses**. But the **2nd byte** was read **without any delay**. Now assuming the 8086 is working at a **frequency** of **10MHz calculate** the following:

a. The value of the Instruction Cycle (in nanoseconds). [3]

b. The values of A0 and BHE' pins while transferring the 2 bytes of data. [3]

c. The value of TON and TOFF assuming the 8086 is running at a duty cycle of 40%. [3]

d. Draw the timing diagram for the operation described in Question. [6]

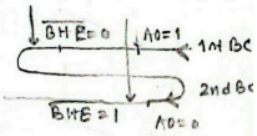
a) At 1st BC
no of Tstates = 4 + 2 = 6
1 Tstate = 100 ns
∴ 6 Tstate = (6 × 100 ns) = 600 ns

f = 10 MHz = $10 \times 10^6$ Hz
$T = \frac{1}{f} = \frac{1}{10 \times 10^6}\ s = \frac{10^9}{10 \times 10^6}\ ns$
∴ T = 100 ns

→ Reading 2 bytes (16 bits) from odd address (unaligned word)
∴ 2 BC

At 1st BC = 600 ns
∴ 2nd BC = (400 ns) = (600 + 400) ns ∴ IC = 1000 ns

b) At 1st BC : A0 = 1 , $\overline{BHE} = 0$
2nd BC : A0 = 0 , $\overline{BHE} = 1$



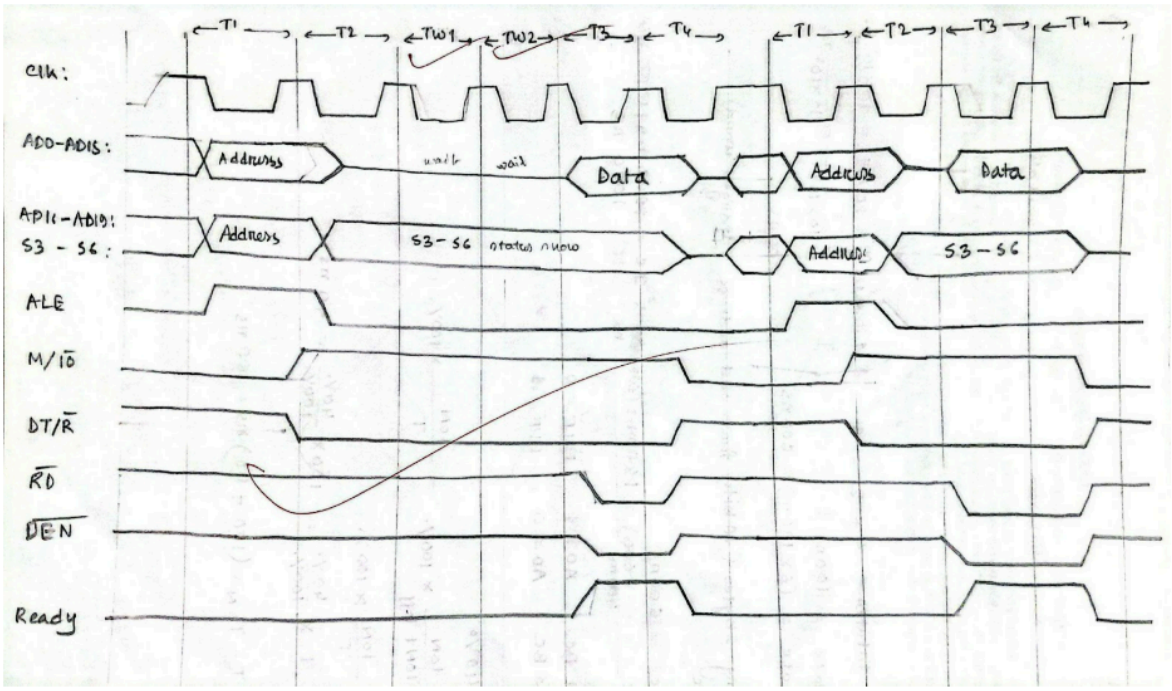

c) DC = 40%

$DC = \frac{T_{ON}}{T_{ON} + T_{OFF}} \times 100\% = \frac{T_{ON}}{T} \times 100\%$

$\Rightarrow 40\% = \frac{T_{ON}}{T} \times 100\%$

$\Rightarrow T_{ON} = T \times \frac{40\%}{100\%} = 100 \times \frac{40\%}{100\%} = 40\ ns$

$\therefore T_{OFF} = T - T_{ON} = (100 - 40)\ ns = 60\ ns$


T1
T2
TW1
TW2
T3
T4
T1
T2
T3
T4
CLK:
AD0-AD15:
Address
wait
wait
Data
Address
Data
AD16-AD19:
S3 - S6:
Address
S3 - S6
Address
S3 - S6
ALE
M/IO
DT/R
RD
DEN
Ready