

Tauseef Tajwar

180041109

CSE 4503

①

Answer to the question no: 1

"a"

§ Intel 80286 has 2 operating modes:

Real Mode:

- In this mode all memory management and protection mechanisms are disabled.
- It allows the microprocessor to address only the first 1MB of memory space.
- The first 1MB is called the real memory.
- Windows does not use the real mode.
- The concept of segment and offset is used.

Protected Mode

- In protected mode 80286 used all 24 address lines to access upto 16MB of physical memory.
- This is where windows operates.

The ~~8~~ protected mode was first introduced by 80286 and continued to pentium processors.

The protected mode supports segmentation and paging. In pentium processors it supports 32-bit mode and segment unit translates 32-bit logical address to 32-bit linear address. And paging unit translates 32-bit linear address to 32-bit physical ~~as~~ address.

In this mode there is a segment descriptor table. The typical program structure follows code, data, stack areas ; CS, DS, SS segment descriptors ; Global descriptor table ; and Local descriptor table.

"b"

A thread of execution is the smallest sequence of programmed instructions that can be managed independently by a scheduler; which is typically a part of the OS. Multiple threads can exist within one process executing concurrently and sharing resources such as memory, while different processes do not share memory.

The idea of threading is that a process can be divided into chunks of independently processable parts and each of those ~~parts~~ parts can be executed parallelly by multiple ~~cores~~ cores of the processor. In case of single core ~~processor~~ processor, if a single core is responsible for running the entire process hence it may take T amount of time. But if the processor has n cores in that case the same process can be divided into

n parallelly executable sections and hence the processing time would be decreased to  $T/n$ . So, threading can improve & ensure faster processing in a multi-core processor system.

### "C"

Parallel programming is the type of programming where the programmer has to utilize the multiple processors to execute a program parallelly to get improved performance.

Let us have a program that computes n values and adds them together. For a serial solution, the program requires n iterations. But let's now consider that we have 8-cores and  $n=24$ . If in this scenario the master core runs the serial program then it would require 7 receives and 7 additions to complete the task.

But if the power of multiprocessors is utilized then the task can be divided among each processor where each part of processors ~~will~~ will compute the sum in a tree like fashion.

This binary logarithmic decrease of ~~of~~ of time allows the whole computation to be done in 3 recces and 3 additions ~~which~~ which is an increase factor of more than two.

Imagine if we have 1000 cores, then ~~if~~ the first case would give us 999 recces and 999 additions. But having 1000 cores can do it in 10 recces and ~~10~~ 10 additions yielding an improvement of a factor of 100.

Answer to the question no: 2"a"

for 8086 microprocessor, the interrupt vector table (IVT) is usually located in the first 1k byte of memory segment (from  $00000\text{H}$  to  $003FF\text{H}$ ).

i

INT 10H is given. 10H refers to 2H.

Which means at first we do  $(2 \times 4)\text{H} = 8\text{H}$ .

So, the IP values can be found in memory location 00008H and 00009H. And the CS values can be found in 0000AH and 0000BH memory location.

(ii)

INT 3 is given.

which means at first we do  $(3 \times 4)H = CH$ .

So, the IP values can be found in memory locations

0030CH and 000DH. And CS values can be found in 0000EH and 0000FH memory locations.

"B"

Data segment is initialized at 0700H:0000H memory location of 8086.

At first the initial offset of A will be 0000H. (at SI)

The initial offset of B will be 000AH. (at DI)

The initial offset of C will be 0010H. (at BX)

Initial offset of D will be 0012H. (at BP)

And the value of the DS register will be 0700H.

(A.1).

"C"

The code to take a single character input and to display the previous character of the ASCII table is as follows:

```
org 100h
• CODE
main proc
    mov ax, @DATA
    mov ds, ax
```

```
    mov ah, 1
```

```
    int 21h
```

```
    mov ah, 2
```

```
    mov dl, 0AH
```

```
    int 21h
```

```
    mov dl, 0DH
```

```
    int 21h
```

```
    mov dl, 9H
```

```
    dec dl
```

```
    int 21h
```

```
ret
```

```
main endp
```

```
end main
```

```
.ret
```

Answer to the question no: 3"a"

Write operation to be made in the I/O port address of  $1111\text{h}$

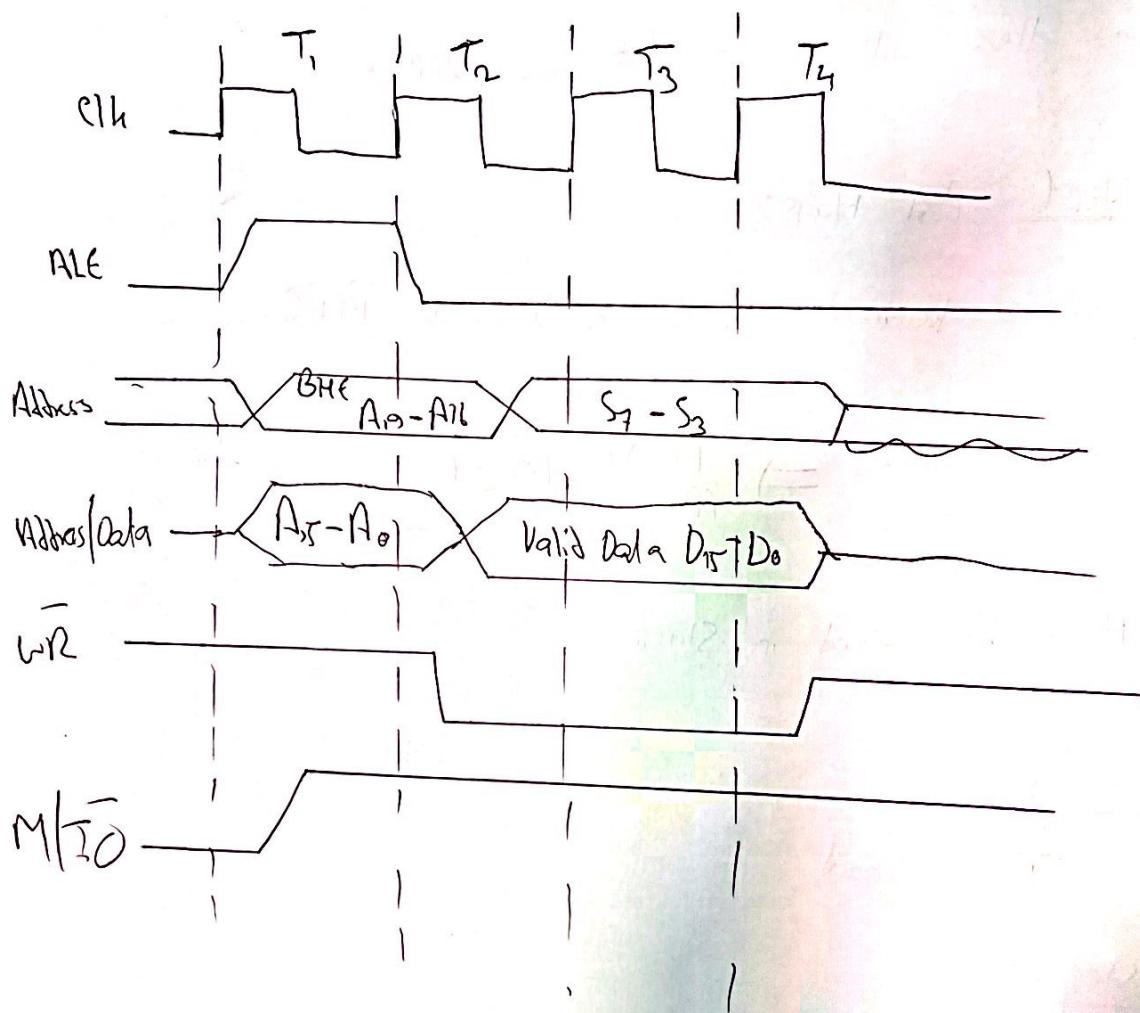


fig: Write cycle of 8086.

"b"

Given SP value is fffah and the CS, IP and FL values are 7000h, 1001h, 1001h respectively. When an interrupt occurs there will be a few steps.

- Step 1: Push flags

SP is decremented by 2 to become FFF8.

Stack contains  $\Rightarrow$ 

FFF9	1001
FFF8	

Flags are pushed in Stack.

- Step 2:

IF is cleared.

- Step 3:

TF is cleared.

- Step 4:

Push CS.

SP is decremented by 2 to become FFF6

Stack contains  $\Rightarrow$

FFF9	1001
FFF8	7000

FFF9	1001
FFF8	7000
FFF7	
FFF6	

CS value is pushed.

Step 5:

push IP.

SP is decremented by 2 to become FFFF4.

Stack contains  $\Rightarrow$

FFFF9	1001
FFFF8	
FFFF7	7000
FFFF6	
FFFF5	1000
FFFF4	

IP value is pushed.

After that the values are ~~pushed~~ popped.

- POP IP  $\Rightarrow$  SP incremented by 2 = FFFF6.

Stack  $\Rightarrow$

FFFF9	1001
FFFF8	
FFFF7	7000
FFFF6	

- POP CS  $\Rightarrow$  SP incremented by 2 = FFFF8

Stack  $\Rightarrow$

FFFF9	1001
FFFF8	

- POP FLAGS  $\Rightarrow$  SP incremented by 2 = FFFFA.

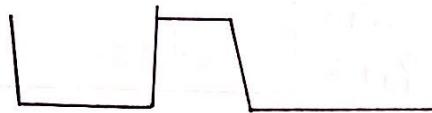
Stack  $\Rightarrow$  initial state. (Ans).

"C"

i) Clock Cycle:

One complete oscillation of the clock pulse is the clock cycle. It is also called a state.

eg:

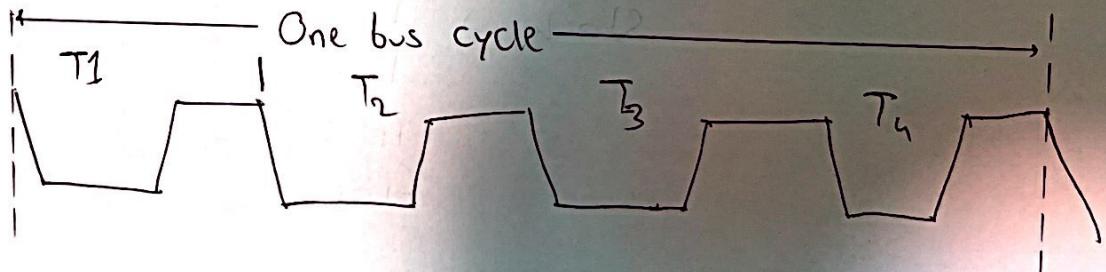


It is one clock cycle.

ii) Bus Cycle:

A basic CPU operation such as reading or writing a byte/word from or to memory or I/O port is called a bus cycle.

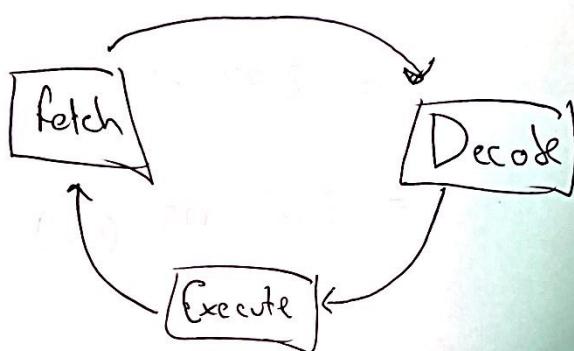
eg:



(iii)

Instruction Cycle:

The time a MP requires to complete fetch-decode-execute operation of a single instruction is known as instruction cycle.

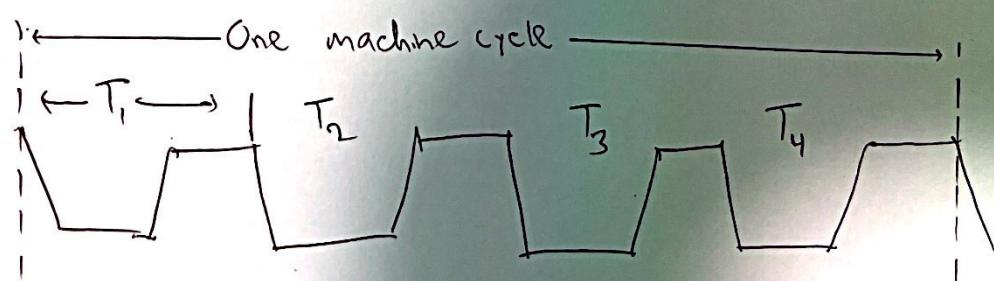


(iv)

Machine Cycle:

A machine cycle is the same as a bus cycle. Consists of at least 4 clock cycles, called T-states.

eg:



⑦ 'T' duration:

for a clock speed of 12 MHz, the

$$T_{\text{duration}} = \frac{1}{12 \times 10^6}$$

$$= 8.33 \times 10^{-8} \text{ seconds.}$$

$$= 83.3 \text{ ns} \quad (\text{Ans}).$$