Explain the purpose of DUP operator with an example

Ans: DUP means duplicate.The DUP operator is very often used in the declaration of arrays This operator works with any of the data allocation directives. The initial value can be an integer, character constant, or another DUP operator, and must always appear within parentheses.
The syntax for DUP:
c DB 5 DUP(0) which is equal as
c DB 0, 0, 0, 0, 0
one more example:
d DB 5 DUP(1, 2) which is equal as
d DB 1, 2, 1, 2, 1, 2, 1, 2, 1, 2

Write short differentiations between the following 8086 assembly language instructions: i. ROR and SHR ii. LEA and OFFSET iii. NOT and NEG

Ans:

1:) ROR and SHR

RoR means rotate right ...it rotates the given bits in the register to the right side one by one.., such that rightmost bit i.e., being rotated is again stored as the MSB in the register, and it is also stored in the Carry Flag (CF).

SHR stands for the shift right....and this instruction shifts the mentioned bits in the register to the right side one by one, but instead of inserting the zeroes from the left end, the MSB is restored. The rightmost bit that is being shifted is stored in the Carry Flag (CF).

2:) LEA and OFFSET

The LEA instruction **places the address specified by its first operand into the register specified by its second operand**.

An address constant is a special type of immediate operand that consists of an *offset* or segment value.

3:) NOT and NEG

One's complement negation ..this is basically same as logical NOT...

NEG computes the two's complement negative value, not the bitwise negative.

Write an assembly language program that takes N as a decimal digit (0 ~ 9) input and shows
the summation of 1+2+ ... + N as output.

Suppose, while debugging an assembly language program the values of the registers are:
Flag=FEB9h, IP=0102h, CS=0S00h, SP=FFFCh. Now, if INT 21h is requested, derive the memory addresses from where the new IP and CS can be retrieved; Also show the new SP value and steps involved in handling the interrupt by the 8086 microprocessor

FFFB - FE **FFFA - B9**
FFF9 - 05 **FFF8 - 00**
FFF7 - 01 **FFF6 - 02**

Narrate the function of using 1, 2 and 9 under INT 21h instruction.
Ans: INT 21h may be used to invoke a large number of DOS functions; a particular function is requested by placing a function number in the AH register and invoking INT 21h. Here we are interested in the following functions:
1 single-key input
2 single-key output
9 character string output

Distinguish between the followings:
1. PolLing and Interrupt. 2. Memory-mapped VO and Isolated 1/0.

Memory-Mapped I/O	Isolated I/O
• 1. In memory-mapped I/O, the I/O devices and memory, both are treated as memory. • 2. In this, any instruction which references to memory can be used. • 3. Its devices are treated as memory locations on the memory map. Therefore, all the instructions related to the memory can be utilized for the data exchange between the processor and the I/O device. • 4. Lesser efficient. • 5. Here, data transfer is possible between any register and I/O device. • 6. Here, a large number of I/O ports (2^{16} ports) are available to be used for interfacing. • 7. In this, one can perform arithmetic and logical operations on the data. • 8. It is simple, as simpler internal logic is used because I/O is also treated as memory only. • 9. Smaller in size. • 10. Faster operations.	• 1. In isolated I/O, the I/O devices are treated as I/O devices and the memory is treated as memory. • 2. In this, limited instructions can be used, like IN, OUT, INS, OUTS. • 3. The addresses for these devices are called ports. Therefore, only the IN and the OUT instructions can be put to use for transferring information between the I/O device and the processor. • 4. More efficient due to separate buses. • 5. Here, data transfer is possible between the accumulator and I/O device only. • 6. Here, only 2^8 ports, are made available for interfacing. • 7. In this, one cannot perform arithmetic and logical operations on the data. • 8. It is complex, as separate internal logic is used to control both. • 9. Larger in size due to more buses. • 10. Slower operations.

i.

Polling	Interrupt
• 1. In polling, the CPU looks for a 'service request flag' by monitoring all served devices continuously. • 2. Whenever CPU encounters a request, it serves the device and then keeps polling. • 3. CPU is always 'busy' with polling executing the 'while any request' loop.	• 1. It informs the CPU if and when a device is ready and needs attention. • 2. CPU drops its currently executing task to serve the device, after that, it returns back to its original task. • 3. CPU is always 'free' when not serving any interrupt.

Briefly explain the operations ofIOPL and NT flags of80286 microprocessor

Ans: IOPL: IOPL is used in protected mode operation to select the privilege level for I/O devices.

If the current privilege level is higher or more trusted than the IOPL, I/O is executed without hindrance.IPOL 00 is the highest or more trusted and IOPL 11 is the lowest or least trusted. If the IOPL value is lower than the current privilege level, an interrupt occurs, causing execution to suspend.

NT: When NT is set, it indicates that one system task has invoked another through a CALL instruction as opposed to a JMP. For multitasking this can be manipulated to have advantage.

With an appropriate timing diagram clearly define the following terms:

Clock cycle: It is the amount of time between two pulses of an oscillator.

Machine cycle: A basic µP operation such as reading or writing a byte/word from or to memory or I/O port is called a Machine Cycle or Bus cycle

Instruction cycle: The time a µP requires to complete fetch-decode execute operation of a single instruction is known as Instruction Cycle

What are real mode, protected mode and virtual mode? Which microprocessor(s) first implements the virtual mode and how?

Real Mode: 80286 is just a fast 8086 --- up to 6 times faster. All memory management and protection mechanisms are disabled. Similar to 8086, the maximum physical address in this mode is 1 Mbyte.

Protected Virtual Address Mode: 80286 works with all of its memory management and protection capabilities with the advanced instruction set. It uses all 24 address lines to access upto 16 Mbytes of physical memory and provides upto Gigabyte range virtual memory.

Distinguish between the DX and SX version of 80386 microprocessor.

80386DX	80386SX
32 bit address bus 32 bit data bus	24 bit address bus 16 bit data bus
Packaged in 132 pin	100 pin flat package
Address 4GB of memory	16 MB of memory

Write an assembly language program structure to clearly state the operational differentiation between LABEL and LOOP?

Level example:

```
.DATA
Message  DB  '128 ASCII char in reverse order:',13,10,'$'

.CODE
MAIN PROC

    MOV AX, @DATA
    MOV DS, AX

    LEA DX, Message
    MOV AH, 9
    INT 21H

    MOV CX, 128              ; initialize CX

    MOV AH, 2
    MOV DL, 127              ; initialize DL with last ASCII character

    top:                  ; loop label
      INT 21H                ; print ASCII character

      DEC DL                 ; decrement DL to next ASCII character
      DEC CX                 ; decrement CX
      JNZ top                ; jump to label top if CX is 0

    MOV AH, 4CH
    INT 21H
```

```
MAIN ENDP
END MAIN
```
Top is the lebel, it is set of instructions which repeats when called

Loop example:

```
org 100h
.DATA ; Data segment starts
str1 db 'Islamic University of Technology',13,10,'$'

.CODE ; Code segment starts
MAIN PROC
mov ax, @DATA
mov ds, ax

xor cx, cx ; reset the CX register
mov cx, 10
mov ah, 9
lea dx, str1
top: int 21h
 loop top

mov ah, 4ch ; equivalent function number of RETURN
int 21h ; Input-Output Interrupt
MAIN ENDP
END MAIN
RET
```
Here top is also a lebel. loop is the keyword for Loop. on calling loop top, top repeats.

Briefly explain about multiple interrupt concepts.

Lecture 8 slide 3-4

Write the equivalent assembly language code structures using conditional jump and loop instructions to implement the ifelse,for and while loop operations.

For example: We write an assembly language program that will accept an input of 5 (five) digits (0 to 9) from the keyboard randomly and rearrange them in ascending order.

Sample Input / Output:

Input: 2 4 5 3 2

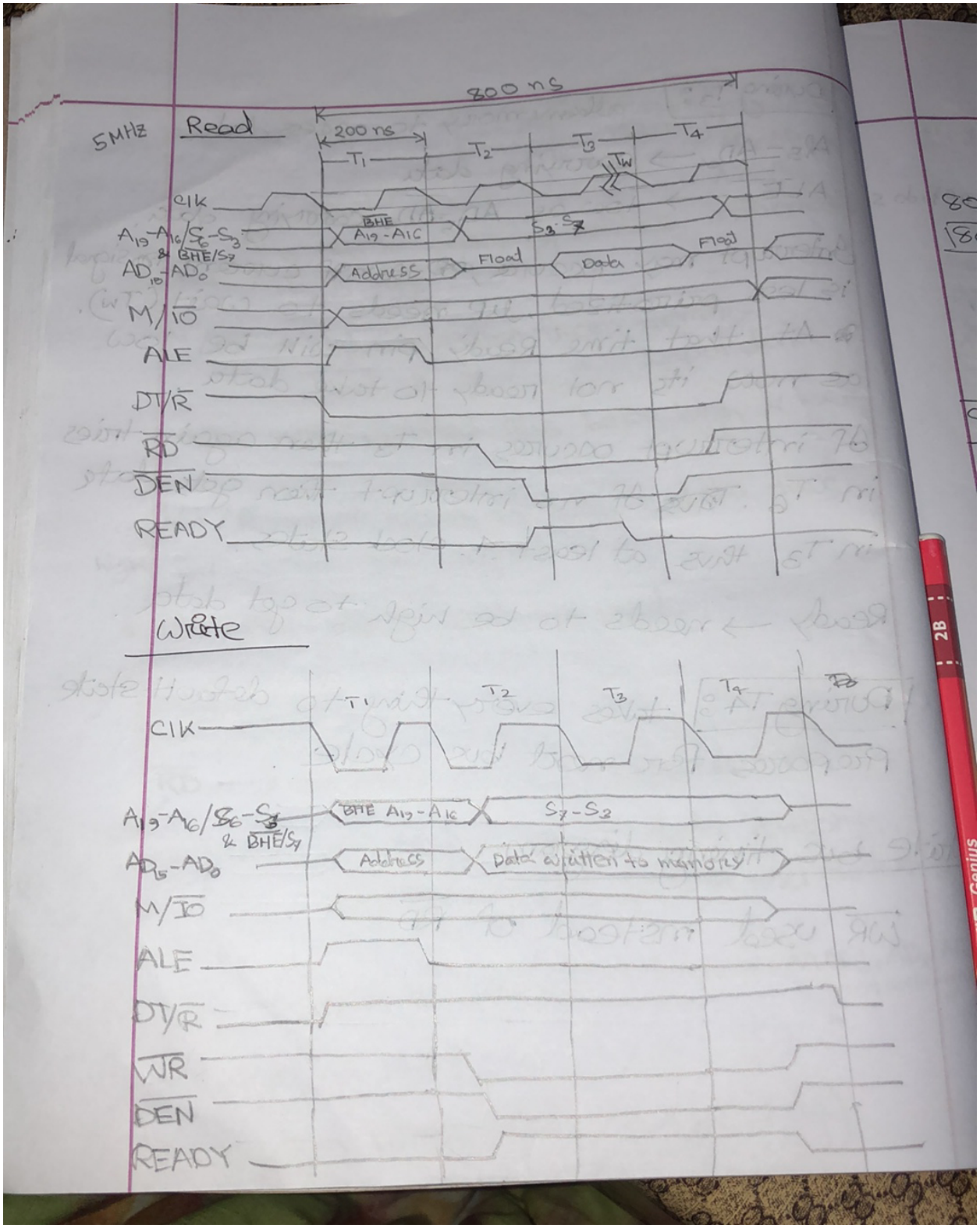
Output: Ascending: 2 2 3 4 5

```
include 'emu8086.inc'
org 100h
.data
arr db 5 dup(?)
input_array db 'Input:$'
```

```
output_array db 'Ascending:$'
.code
main proc
mov ah,9
lea dx,input_array ; output of the input string
int 21h
mov cx,5 ; takes the value 5 for the input loop
lea bx, arr; load effective address of arr taken in bx
mov ah,1
input:
int 21h
mov [bx],al; input taken into the arr array
inc bx
loop input
mov cx,5
dec cx
outerloop:
mov bx, cx
mov si,0
comploop:
mov al,arr[si]
mov dl,arr[si+1]
cmp al,dl
jc noSwap
mov arr[si],dl
mov arr[si+1],al
noSwap:
inc si
dec bx
jnz comploop
loop outerloop
mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h
mov ah,9
lea dx,output_array
int 21h
mov cx,5
mov bx, offset arr
Outputs:
mov dl,[bx]
mov ah,2
```

```
int 21h
inc bx
loop Outputs
main endp
ret
```

Draw the bus timing diagram for a microprocessor's operation while it performs a WRITE operation toward an OUTPUT unit.



What are the basic differences between MIN and MAX mode of8086 pin diagram?

Minimum mode	Maximum mode
In minimum mode there can be only one processor i.e. 8086.	In maximum mode there can be multiple processors with 8086, like 8087 and 8089.
MN/$\overline{MX}$ is 1 to indicate minimum mode.	MN/$\overline{MX}$ is 0 to indicate maximum mode.
ALE for the latch is given by 8086 as it is the only processor in the circuit.	ALE for the latch is given by 8288 bus controller as there can be multiple processors in the circuit.
DEN$\overline{}$ and DT/$\overline{R}$ for the trans-receivers are given by 8086 itself.	And DT/$\overline{R}$ for the trans-receivers are given by 8288 bus controller.
Direct control signals M/$\overline{IO}$, RD$\overline{}$ and WR$\overline{}$ are given by 8086.	Instead of control signals, each processor generates status signals called S2$\overline{}$, S1$\overline{}$ and S0$\overline{}$.

Minimum mode	Maximum mode
Control signals M/$\overline{IO}$, RD$\overline{}$ and WR$\overline{}$ are decoded by a 3:8 decoder like 74138.	Status signals S2$\overline{}$, S1$\overline{}$ and S0$\overline{}$ are decoded by a bus controller like 8288 to produce control signals.
INTA$\overline{}$ is given by 8086 in response to an interrupt on INTR line.	INTA$\overline{}$ is given by 8288 bus controller in response to an interrupt on INTR line.
HOLD and HLDA signals are used for bus request with a DMA controller like 8237.	RQ$\overline{}$/GT$\overline{}$, lines are used for bus requests by other processors like 8087 or 8089.
The circuit is simpler.	The circuit is more complex.
Multiprocessing cannot be performed hence performance is lower.	As multiprocessing can be performed, it can give very high performance.

In how many ways can you define an array using assembly language programming? Give example code for each of them.

Ans: Arrays can be seen as chains of variables. A text string is an example of a byte array; each character is presented as an ASCII code value (0..255). Here are some array definition examples:
a DB 48h, 65h, 6Ch, 6Ch, 6Fh, 00h
b DB 'Hello', 0
b is an exact copy of the an array, when compiler sees a string inside quotes it automatically converts it to a set of bytes. This chart shows a part of the memory where these arrays are Declared:

If you need to declare a large array with same value you can use DUP operator. The syntax for DUP: For example:
c DB 5 DUP(0)

c DB 0, 0, 0, 0, 0 ; is an alternative way of declaring:
one more example:
d DB 5 DUP(1, 2)
d DB 1, 2, 1, 2, 1, 2, 1, 2, 1, 2 ; is an alternative way of declaring:

'Utilization of parallel processors can be achieved through parallel programming'. How? Prove with appropriate examples.

The utilization of parallel processors can be achieved by parallel programming as follows:

1. Multithreading:

Multithreading helps in parallel programming by providing memory references the ability to run simultaneously as unrelated instructions.

2. Message passing:

The computation is divided into multiple processes. It causes the processes to be dependent on the same data. It needs the processes to send messages to each other via a communication medium.

In parallel processors, there are at least two microprocessors that manage parts of an overall task.

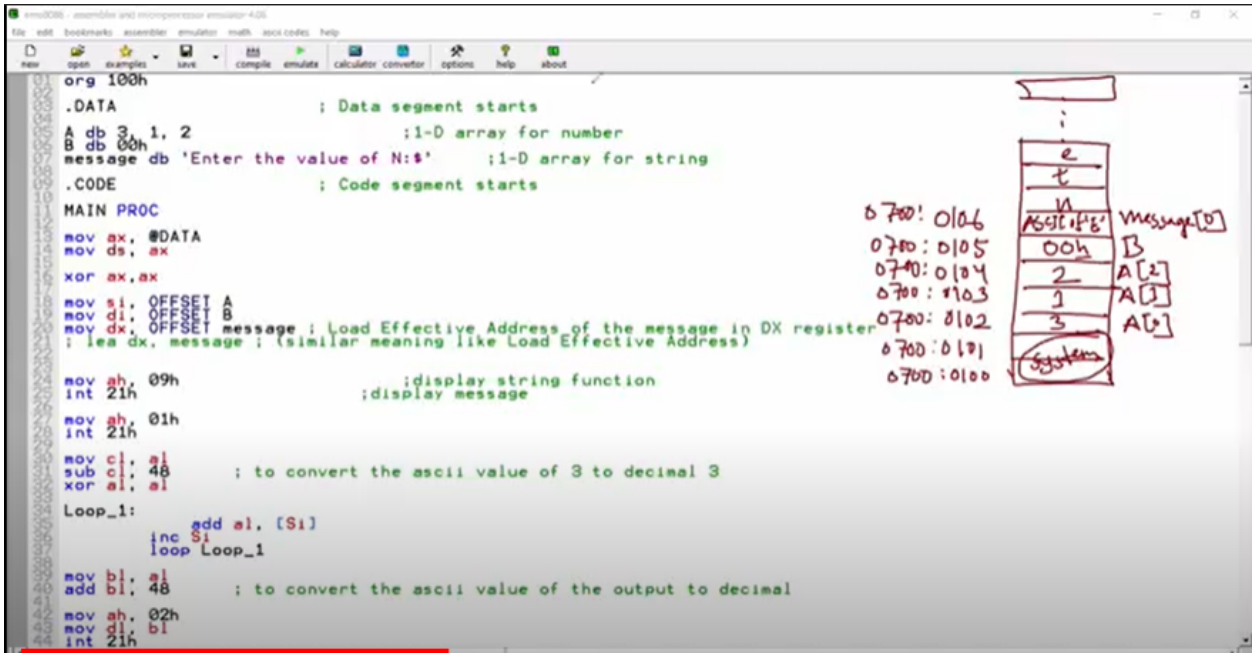
A complex or big problem is divided into components using special

Let, in an assembly language code data segment is initialized at `7000h:0000h` memory location of 8086 and variables are declared as follows:

```
.DATA
A db 1, 2, 3, 4, 5
B db 'IUT'
C dw FFFFh
D db 'Exam Date is 24 August, 2020','$'
```

Now, using appropriate instructions derive and store the initial offsets of A, B, C and D at SI, DI, BX, BP registers, respectively. Also, derive the DS register value.

// 0000 and 0001 system uses the memory location



```
org 100h
.DATA                   ; Data segment starts
A db 3, 1, 2                    ;1-D array for number
B db 00h
message db 'Enter the value of N:$'     ;1-D array for string
.CODE                   ; Code segment starts
MAIN PROC

mov ax, @DATA
mov ds, ax

xor ax,ax

mov si, OFFSET A
mov di, OFFSET B
mov dx, OFFSET message ; Load Effective Address of the message in DX register
; lea dx, message ; (similar meaning like Load Effective Address)

mov ah, 09h                     ;display string function
int 21h                 ;display message

mov ah, 01h
int 21h

mov cl, al
sub cl, 48      ; to convert the ascii value of 3 to decimal 3
xor al, al

Loop_1:
        add al, [Si]
        inc Si
        loop Loop_1

mov bl, al
add bl, 48      ; to convert the ascii value of the output to decimal

mov ah, 02h
mov dl, bl
int 21h
```

```
org 0000h

.DATA
A db 1, 2, 3, 4, 5
B db 'IUT'
C dw 0FFFFh
D db 'Exam Date is 24 August, 2020','$'

.code
main proc
    mov ax, @data
    mov ds, ax

    mov si, offset A
    mov di, offset B
    mov bx, offset C
    mov bp, offset D

    ret
main endp
end main

DS = 7000h
SI = 0000h
DI = 0005h
BX = 0008h
BP = 000Ah
```

Instruction cycle. Also, find out the T durations of the microprocessors having clock speeds of 40 MHz and 1 GHz, respectively.

T duration

Example: If each bus cycle needs 4 T duration. then,

T duration = Clock cycle

$$= \frac{1}{40 \times 10^6} s$$

$= 25$ ns

Which memory locations are reserved for 8086 Interrupt Vector Table? Derive the specific memory locations where the CS and IP values of the following Interrupt Types can be found:

i. INT 21h
ii. INT 3

I. (21*4)h = 84 h

Ip address : 00084 h 00085 h
Cs : 00086 h 00087 h

Ii) (3*4)= 12 = Ch
Ip : 0000ch 0000dh
Cs : 0000eh 0000fh

Let, in 8086 based system current SP value is FFF8h and the CS, IP and FL values are 7000h, 0100h and 1101h, respectively. Now, if at instance an interrupt occurs, then show the stack increase and decrease scenario along with changes in SP values and stack contents.

Setp	SP	- Content
1		FFF7 - 11
	(SP)	FFF6 - 01
2		FFF5 - 70
	(SP)	FFF4 - 00
3		FFF3 - 01
	(SP)	FFF2 - 00
4		FFF2 - $
		FFF3 - $
	(SP)	FFF4
5		FFF4 - $
		FFF5 - $
	(SP)	FFF6
6		FFF6 - $
		FFF7 - $
	(SP)	FFF8

Finally, SP = FFF8

Write an assembly language program that will display "Islamic University of Technology" 10 (ten) times with new lines each starts at home position.

```
org 100h
.DATA ; Data segment starts
str1 db 'Islamic University of Technology',13,10,'$'

.CODE ; Code segment starts
MAIN PROC
mov ax, @DATA
```

```
mov ds, ax

xor cx, cx ; reset the CX register
mov cx, 10
mov ah, 9
lea dx, str1
top: int 21h
 loop top

mov ah, 4ch ; equivalent function number of RETURN
int 21h ; Input-Output Interrupt
MAIN ENDP
END MAIN
RET
```

Differentiate between the features of core i3, i5 and i7 processors.

Features	Core 2	Core i3	Core i5	Core i7
* Core CPU	Two cores in one processor Totally 4 cores	Dual core CPU	Dual core CPU	Dual core & Quad core CPU
* Threading	One thread per physical core	(2 logical threads / physical core) Hyper	Hyper	Hyper
* Clock speed	2.2 GHz (High)	Medium speed	Slightly higher than i3	Higher clockspeed than i5
* Turbo boost capacity	—	—	Turbo Boost	Turbo Boost
* Virtualization	Multiple tasks can be done at same time	—	—	Can be done
* Performance	Good	Adequate	Good Performance	Great performance

Define Thread and Turbo Mode in the context of a multi-core processor system?

A thread of execution is the smallest sequence of programmer instructions that can be managed independently by the scheduler, which is typically a part of the OS. Multiple threads existing within a process means each thread executes concurrently but shares resources like memory (which different processes do not share). The schedule assigns these tasks to different cores in the required order. For example, for $(a + b) \times (c + d)$, the additions need to take place at the same time while the multiplication needs to take place afterwards.
The Core i3 also supports hyper-threading, which has efficient use of processor resources. It has a 3 to 4 MB cache and uses 32 nm Silicon, with the small size causing less heat and using less energy.

Turbo mode:

Turbo mode or turbo boost technology is a way to run the processor cores faster than the marked frequency automatically. This was introduced by intel.

When the two cores are running, it will run at up to the 3.9GHz, and when the four cores are running, it will run at 3.7GHz.

This Turbo mode was recently introduced on x86 multi core chips from intel and also AMD. It will over clock the cores by utilizing the available thermal headroom. These chips can averagely overclock a core by upto 40% or even more sometimes. As this increase will lead to the increase in performance.

Thank you :)

What do you mean by Coppermine? How do Coppermine and L2 cache memory differ from 8 each other?

Coppermine is an internal cache with 256K advanced transfer mechanism within the IC running at processor speed

L2 cache memory is far better than the coppermine.
With L2 cache memory, the cpu performance will be better.
The L2 cache memory allows more instructions and data to be retained and increases the probability that the cache will be correct.
Coppermine processors are built using traditional silicon but the L2 cache memory is built using modern technologies....

How are the main memory of 80386 and Pentium processors segmented? Mention the use of address bus pins for both 80386 and Pentium microprocessors.

GIven

Describe the process of segmentation in 80386 microprocessor.

Step 2

The memory management unit (MMU) consists of a segmentation unit and a paging unit. The \ssegmentation unit allows the use of two address components, such as segment and offset for revocability and sharing of code and data. The segmentation unit allows segments of size 4Gbytes at maximum. The segmentation unit provides a four-level protection mechanism for protecting and isolating the system's code and data from those of the application programme.

Step 3

Descriptors

There is a descriptor in each segment. The system software creates the respective descriptor with the help of compilers and loaders, linkers when the programmer creates a logical segment. This is the task of the OS. Segment information is included in the descriptors. There is a 20- bit segment limit and a 32-bit segment address in the 80385 descriptors. The 80386 descriptors are 8 bytes with access right or attribute bits together with segments base and limits.

Tables of Descriptor

A number of segments of different types for different applications will be created in any system. As many descriptors should be there as well. All descriptors are stored in tables called tables of descriptors. These tables are created and stored in memory using system software. The segmentation scheme offers a way to protect data and code of various types.

Segmentation: Pentium

This process helps in dividing programs into logical blocks and then placing them in different memory areas. This makes it possible to regulate access to critical sections of the application and help identify bugs during the development process. It includes several features like to define the exact location and size of each segment in memory and set a specific privilege level to a segment which protects its content from unauthorized access. [6]

Segment registers are now called segment selectors because they do not map directly to a physical address but point to an entry of the descriptor table.

Pentium CPU has six 16 bit segment registers called SELECTORS. The logical address consists of 16 bit of segment size and 32 bit offset. The below figure shows a multi-segment model which uses the full capabilities of the segmentation mechanism to provide hardware enforced protection of code, data structures, and programs and tasks. This is supported by IA-32 architecture. Here, each program is given its own table of segment descriptors and its own segments.

Write an assembly language program, where a MACRO is used to address a string and a PROCEDURE is used to display that string

```
MyMacro2 MACRO
    LOCAL label1, label2
    CMP AX, 2
    JE label1
    CMP AX, 3
    JE label2
    label1:


      label1:
          INC AX
      label2:
          ADD AX, 2
ENDM

ORG 100h
MyMacro2
MyMacro2
RET
```

ALP for Finding Factorial of number using procedures

```
CODE SEGMENT
ASSUME CS:CODE
 FACT MACRO
 MOV BX,AX
 DEC BX
 BACK: MUL BX
 DEC BX
 JNZ BACK
 ENDM
START: MOV AX,7
 FACT
 MOV AH,4CH
 INT 21H
 CODE ENDS
 END START
```