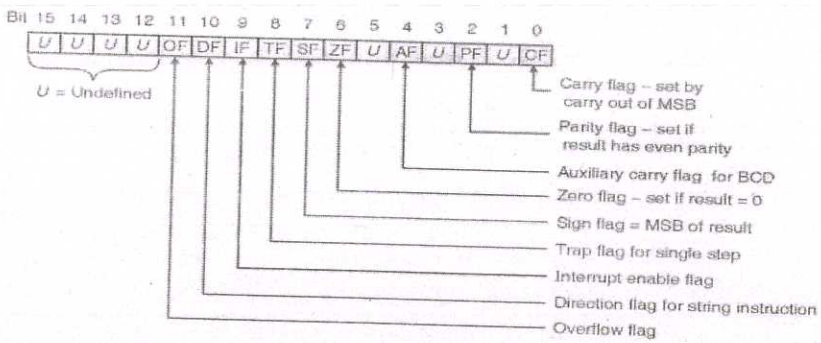


SCHEME OF VALUATION
(Scoring Indicators)

Revision: 2015

Course Title: ADVANCED MICROPROCESSORS

Course code: 6041

| Qst. No. | Scoring Indicator | Split up score | Sub Total | Total |
|-----------|---|----------------|-----------|-------|
| I | <u>PART A</u> | | | |
| 1 | Pin No. 33 when made high, 8086 operate in minimum mode and when it is low , CPU operate in maximum mode | 1+1 | 2 | |
| 2 | 1. Hardware and Software interrupts. 2. Vectored and Non-vectored interrupts. 3.Maskable and Non-maskable interrupts. <i>Any two interrupts</i> | 1+1 | 2 | |
| 3 | Segmentation unit Paging unit | 1+1 | 2 | |
| 4 | U- pipe line & V- pipe line | 1+1 | 2 | |
| 5 | A multi core processor where every core is just an image of the other is called a homogeneous multi core processor. | 2 | 2 | 10 |
| II | <u>PART B</u> | | | |
| 1 |  <p align="center">8086 Flag Register format <i>(Description 3 marks + figure 3 marks)</i></p> | 3+3 | 6 | |

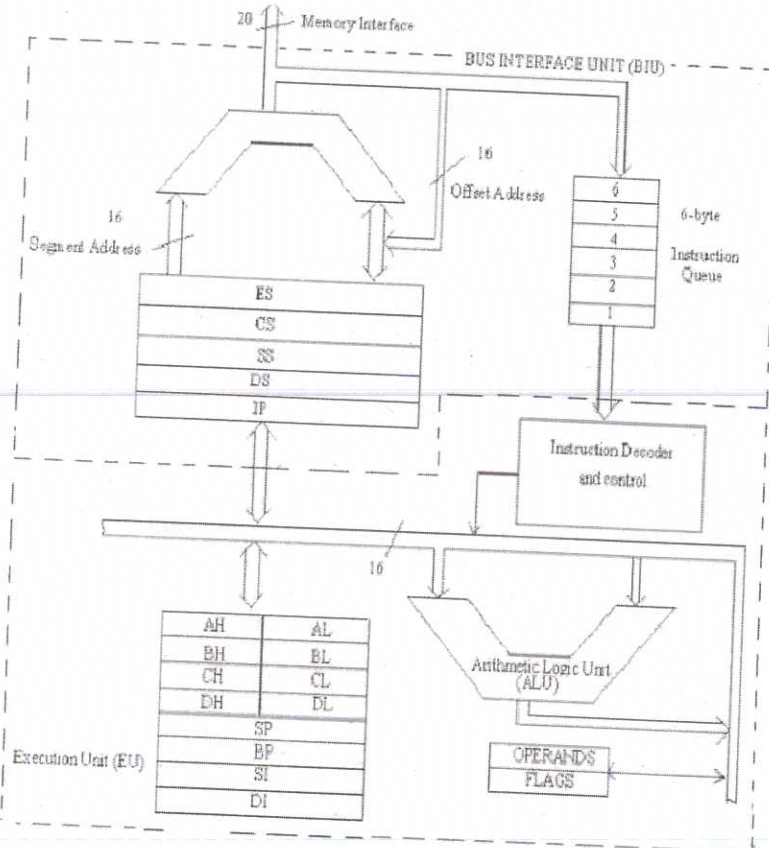
| | | | | |
|---|--|-----------|---|--|
| 2 | <p>Addressing mode – The method of specifying the data to be operated by the instruction. Register addressing – the instruction will specify the name of the register which holds the data to be operated by the instruction. Eg: MOV AX, BX Implied addressing – the instruction itself will specify the data to be operated by the instruction. Eg: CLC</p> <p><u>Definition (2 marks) + addressing mode explanation with example 2 marks each</u></p> | 2+2+2 | 6 | |
| 3 | <p>Assembler – software that converts assembly language program codes to machine language codes.</p> <p>(i) ASSUME – Indicates the name of each segment to the assembler. (ii) DB – Define byte. (iii) MACRO – Defines the name, parameters and start of a macro (iv) ENDP – End of procedure (v) EQU – Equate. Used to equate numeric value or constant to a variable.</p> <p><u>Assembler (1 mark) + 1 mark each for directives</u></p> | 1+1+1+1+1 | 6 | |
| 4 | <ul style="list-style-type: none"> ■ 32 bit Processor ■ 32 bit address and data bus. ■ All registers are 32 bit size ■ The maximum size of physical memory is 4Gb. ■ Virtual memory of 64 TB can be used ■ Since 80386 runs at higher clock frequency, faster execution speed is obtained and most instructions take fewer clock cycles to execute. ■ The on chip memory management supports paging. ■ 3 modes of operation - real mode, protected virtual address mode and virtual 8086 mode. <p><u>(any 6 features. 1 mark each)</u></p> | 1x6 | 6 | |

| | | | | |
|---|---|-----|---|----|
| 5 | <ol style="list-style-type: none"> 1. Hyper-threading provides two logical processor in a single logical processor 2. It is a technique used to improve parallelization of computations performed on PC 3. It is a technique which enables CPU to act like multiple CPUs. 4. Enables different parts of the CPU to work on different tasks concurrently 5. Threaded software application to execute threads in parallel in a single multi-core processor instead of thread processing in linear fashion 6. Hyper-threading technology takes advantage of multithreading capability 7. Divide the workload into processes and threads 8. Execute the threads on different processors <p style="text-align: center;"><i>(6 points. 1 mark each)</i></p> | 1x6 | 6 | |
| 6 | <ol style="list-style-type: none"> 1. Intel P6 micro architecture 2. Three way superscalar architecture 3. Five parallel execution units and 12-stage pipeline. 4. 36 bit address bus and 64 GB physical memory 5. Register renaming 6. Error checking and correcting codes. 7. Improved power management with two extra modes (Stop Grant and Auto HALT modes). 8. Internal thermal protection 9. Integrated level two cache <p style="text-align: center;"><i>(Any 6 features. 1 mark each)</i></p> | 1x6 | 6 | |
| 7 | <p>--Multi core CPU Designs require only much less PCB space</p> <p>--Multi core processors use less power compared to a single core CPU</p> <p>-- Multi cores shares some circuitry like L2 cache and the interface to the Front Side Bus(FSB)</p> <p>-- Multi core chips allow high performance at lower energy</p> <p>-- Multiple cores can execute multiple instructions at the same time and increases the speed of programs in parallel processing.</p> <p>--The proximity of multiple CPU cores on the same die allows the cache coherency circuitry to operate at much higher clock rate</p> <p style="text-align: center;"><i>(6 points. Each point carries 1mark)</i></p> | 1x6 | 6 | 30 |

PART-C

III(a)

The internal architecture of 8086 is divided into two parts namely Bus Interface Unit (BIU) & Execution Unit (EU). Execution Unit receives program instruction codes and data from the BIU, executes them and stores the results in the general registers. EU has no connection with the system buses. It receives and outputs all its data through BIU. Bus Interface Unit (BIU) - BIU and EU are connected with an internal bus. BIU connects EU with the memory or I/O circuits. It is responsible for transmitting data, addresses and control signal on the busses.



6+4

10

Diagram (6 marks) + brief description of blocks (4 marks)

III(b)

The physical address of 8086 is 20-bits long and it is generated using segment and offset registers, each 16-bits long. The content of segment register is shifted left bit-wise four times and the resulting 20 bit base address is added with the 16 bit offset address to get the 20 bit physical address,
So, Physical Address = Base Address + Offset.

Suppose the Data Segment holds the Base Address as 1000h and the data you need is present in the 0020h memory location (Offset) of the Data Segment. The calculation of the actual address

is done as follows.

1. Left shift the 16-bit address present in the segment register by 4-bits

0001 0000 0000 0000 (0000)

2. Add the 16-bit offset address to this shifted base address

0001 0000 0000 0000 0000

+ 0000 0000 0010 0000

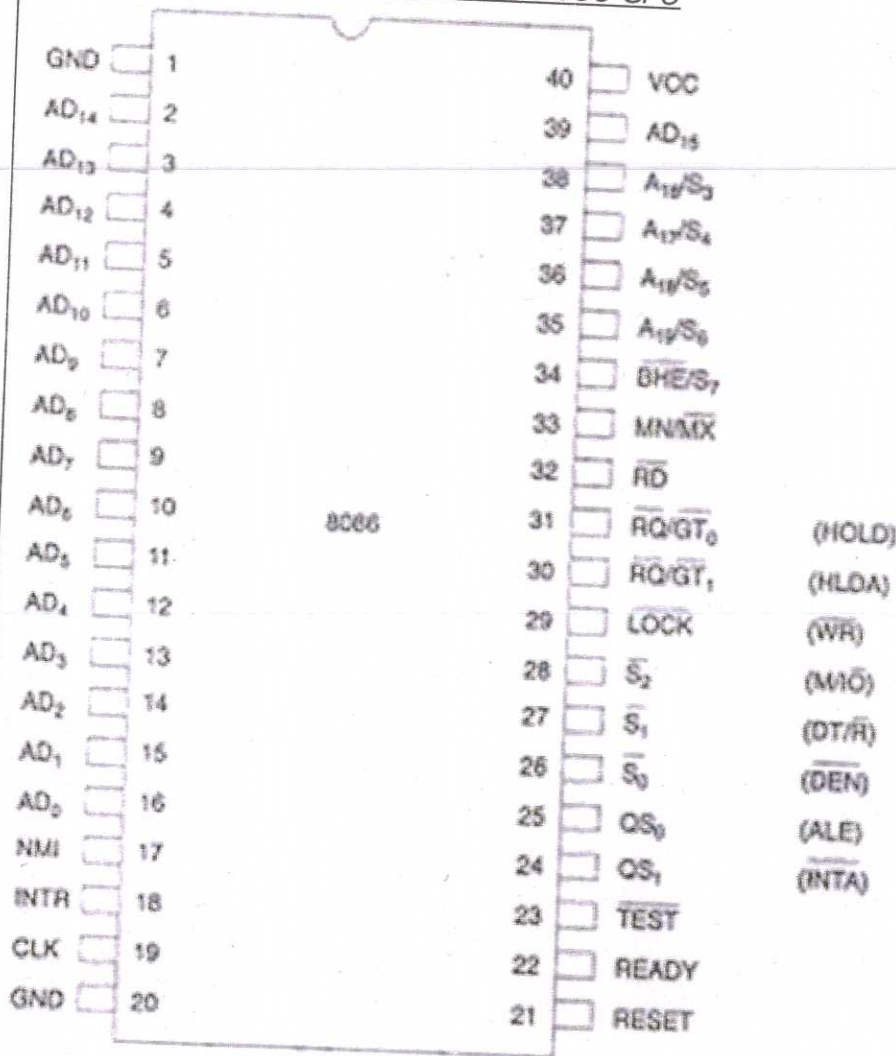
0001 0000 0000 0010 0000

So the actual address turns out to be 10020h.

(Exp 3 marks and example 2 marks)

Pin diagram of 8086 CPU

IV(a)



(Pin diagram 4 marks and Exp. 4 marks)

3+2

5

15

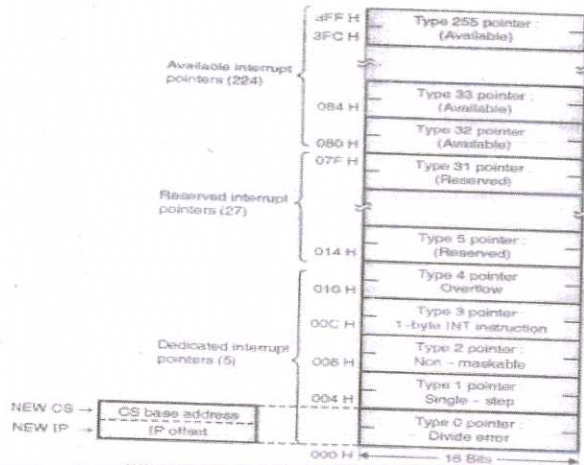
4+4

8

| | | | | |
|--------------|--|-----|---|----|
| IV(b) | <p>General purpose registers in 8086 microprocessor</p> <p>AX – This is the accumulator. ... BX – This is the base register. ... CX – This is the counter register. ... DX – This is the data register. ... SP – This is the stack pointer. ... BP – This is the base pointer. ... SI – This is the source index register. ... DI – This is the destination index register.</p> <p><i>(Register listing 4 marks and Exp. 3 marks)</i></p> | 4+3 | 7 | 15 |
| V(a) | <p><u>Arithmetic instructions of 8086</u></p> <p>ADD : Add the two operands ADC : Add with Carry SUB : Subtract SBB : Subtract with Borrow INC : Increment. DEC : Decrement NEG : Negate CMP : Compare MUL : Unsigned Multiplication Byte or Word IMUL : Signed Multiplication DIV : Unsigned division AAA : ASCII Adjust After Addition AAS : ASCII Adjust AL after Subtraction AAM : ASCII Adjust after Multiplication AAD : ASCII Adjust before Division DAA : Decimal Adjust Accumulator DAS : Decimal Adjust after Subtraction</p> <p><i>(Any 9 instructions 1mark each)</i></p> | 1x9 | 9 | |

| | | | | |
|-------|---|-----|---|----|
| | | | | |
| V(b) | <p style="text-align: right;"><u>ALP to find largest of 10 numbers</u></p> <pre> CODE SEGMENT ASSUME CS:CODE ORG 100H START: MOV SI,1100H MOV DI,2000H MOV CL,10 INC SI MOV AL,[SI] DEC CL AGAIN:INC SI MOV BL,[SI] CMP AL,BL JNC AHEAD MOV AL,BL AHEAD:DEC CL JNZ AGAIN MOV [DI],AL HLT CODE ENDS END START </pre> | | | |
| V(a) | | 6 | 6 | 15 |
| VI(a) | <ol style="list-style-type: none"> 1. SP is decremented by two and the content of flag register is pushed to stack memory. 2. The interrupt system is disabled by clearing Interrupt flag. Trap flag is disabled. 3. SP is decremented by two and the content of CS register is pushed to stack memory. Again, the SP is decremented by two and the content of IP is pushed to stack memory. 4. In case of INTR, the processor runs an interrupt acknowledge cycle to get the interrupt type number. 5. The processor generates a 20-bit memory address by multiplying the type number by and sign extending to 20-bit. 6. The first word pointed by vector table address is loaded in IP and the next word is loaded in CS register. 7. The 20 bit physical memory address is calculated by multiplying the content of CS register by 16 and adding to the content of IP. 8. The processor executes the ISR to service the interrupt. 9. At the end of ISR, the previous status of the processor is restored and the processor will resume execution of normal program from the instruction where it was suspended. <p style="text-align: center;"><i>(9 steps. 1 mark for each step)</i></p> | 1x9 | 9 | |

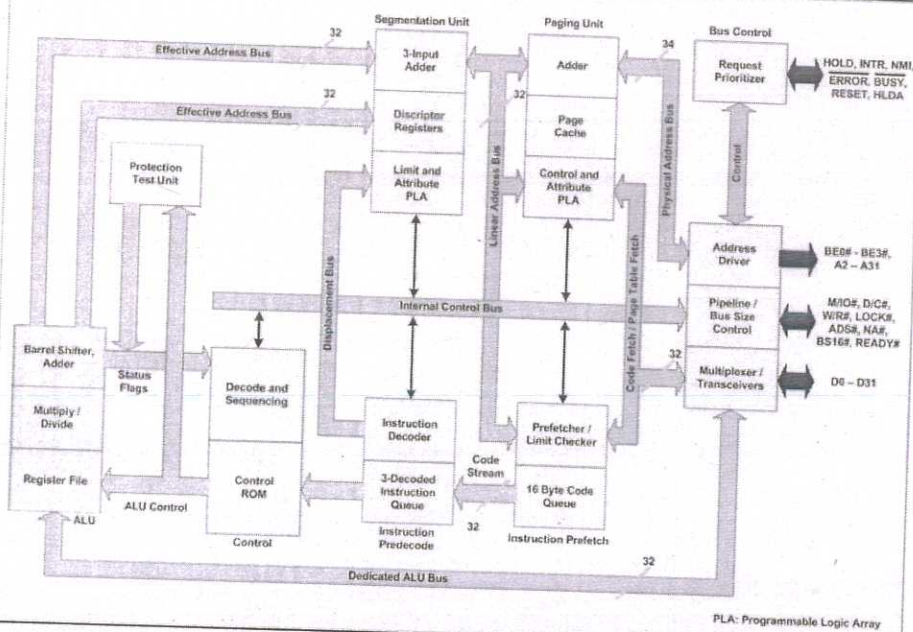
VI(b) In IVT, 16-bit offset address and 16-bit segment base address of each ISR are stored in four consecutive memory locations. The addresses stored in this table are called vector addresses. IVT requires 1Kb memory space.



4+2 6 15

Diagram (4 marks) + Description (2 marks)

VII(a)



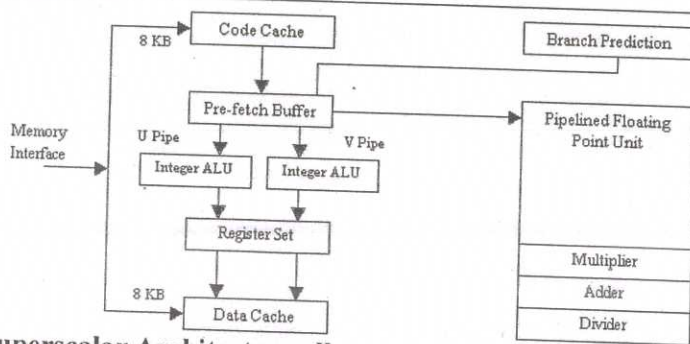
6+6 12

(Block diagram 6 marks and Exp. 6 marks)

VII(b) CR0 – Machine Control register. 32 bit long
 CR1 Reserved
 CR2 – contains a value called Page Fault Linear Address. When a page fault occurs, the address the program attempted to access is stored in CR2 register.
 CR3 – used to hold the root address of page directory)

3 3 15

VIII a



Superscalar Architecture allows parallel execution of 2 instructions.

Figure (4 marks) + Description (4 Marks)

4+4

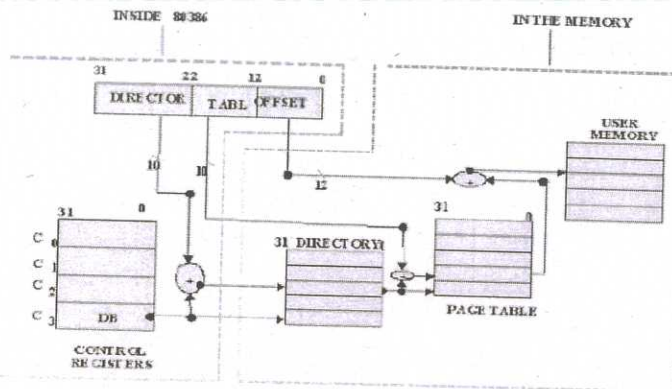
8

VIII(b)

In paging mechanism the memory segments can be organized as pages of size 4Kb and allows 2^{20} pages. Involves 3 elements- page directory, page table and page. Four bytes of page directory are used to store information about a page table. Each page table can define 1024 pages of size 64Kb.

The segmentation unit will supply a 32-bit linear address to paging unit. The upper 10 bits ($A_{22}-A_{31}$) of linear address is the index for page directory, the middle 10-bits ($A_{12}-A_{21}$) of linear address is the index for page table and lower 12-bits (A_0-A_{11}) of linear address is the lower 12-bits of physical address.

The control register CR_3 holds the root address of page directory. The root address is added to 10-bit page directory index to get the address page directory entry of the page table. Page directory entry has base address of page table which is added to 10-bit page table index to get the address of page table entry of the page. The page table entry has upper 20-bits of page frame address which is concatenated with lower 12-bits of linear address to form physical address.



Description (4 marks)+ Figure (3 marks)

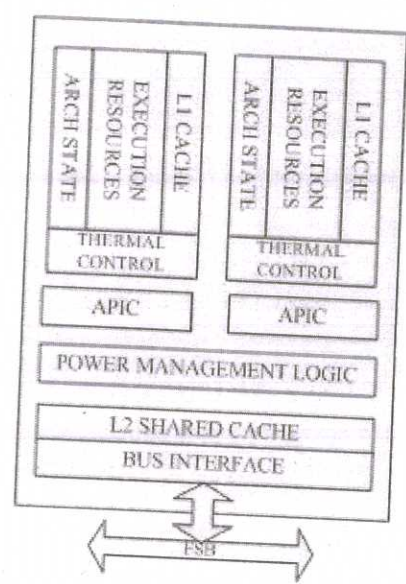
4+3

7

15

IX(a)

Intel Core 2 Duo processor block diagram



(Block diagram 5 marks and Exp. 5 marks)

5+5 10

IX(b)

Comparison of core i3,i5 and i7 processors

| | <i>Core i3</i> | <i>Core i5</i> | <i>Core i7</i> |
|--------------------------|---------------------------------|----------------------|------------------------------|
| <i>No. of cores</i> | Dual core | Dual/Quad core | Quad core |
| <i>Hyper threading</i> | Yes | No | Yes |
| <i>Clock speed range</i> | 3.4 GHz – 4.2 GHZ | 2.4GHz – 3.8GHz | 2.9 GHz – 4.2GHz |
| <i>Cache memory</i> | Has up to 3MB cache | Has 3MB to 6MB cache | Has 4 MB to 8MB cache |
| <i>Turbo boost</i> | No Turbo boost | Supports Turbo boost | Supports Turbo boost |
| | Entry level CPU.Economic choice | Mid grade CPU | High end CPU. Most expensive |

(Any 5 points. 1 mark each)

5x1 5 15

| | | | | |
|------|--|-------|---|----|
| X(a) | <p align="center"><u>Issues With Many-Core Processors</u></p> <p><u>•Power</u></p> <ul style="list-style-type: none"> - Difficult to predict on chip power - Difficult to predict the utilization of the different processors at any given instance. - May lead to overheating if all cores are running at full potential. - Increased power dissipation with decreasing feature size <p><u>•Connectivity</u></p> <ul style="list-style-type: none"> •Increase in the number of wires •Increase in power dissipation •Need for long wires <p><u>Common Memory</u></p> <p>As the different cores of a multi core system share the DRAM memory system, multiple programs executing on different cores can interfere with each other's memory access requests, which adversely affects the performance.</p> <p><u>•Cache Coherence</u></p> <p>Since each core has its own cache, the copy of the data in that cache may not always be the most up to date version, which may lead to cache miss</p> <p><u>•Programmability</u></p> <p>Programmers has to write applications to be run in different cores, meaning that data dependencies have to be resolved or accounted for. If one core is used much more than the other, the programmer is not taking the full advantage of the multi core system.</p> <p><u>Starvation</u></p> <p>If a program is not developed correctly for a multi core system, then one or more cores may srarve for data. If a single threaded application is run in a multi core system, the thread will simply run in one of the cores and other cores may simply sit idle.</p> <p align="center"><i>(1.5 marks for each point)</i></p> | | | |
| X(b) | <p><u>Technological features of IA 64 processors</u></p> <ul style="list-style-type: none"> - EPIC (Explicit Parallel Instruction Computing) - 64 bit address flat memory model - 2.1 GB/S Front Side Bus - Large register file - Register rotation - Powerful Integer Architecture - Advanced Floating Point Architecture - Supports Multimedia(MMX) Technology - 3 level cache (L1 & on chip and L3 on cartridge) - 10 stages in order pipe line <p align="center"><i>(Any 6 points. 1 mark each)</i></p> | 1.5x6 | 9 | |
| | | 1x6 | 6 | 15 |