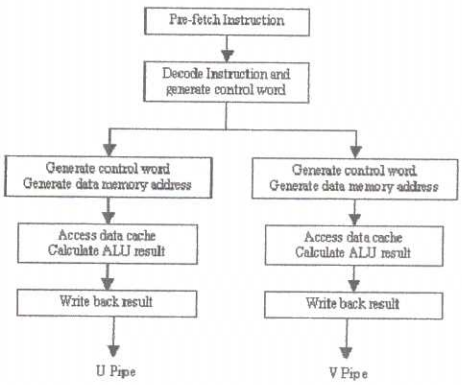
SCHEME OF EVALUATION

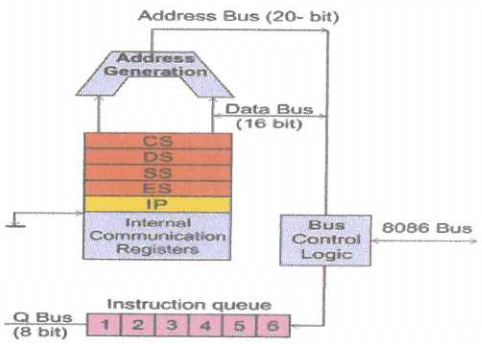
(SCORING INDICATORS)

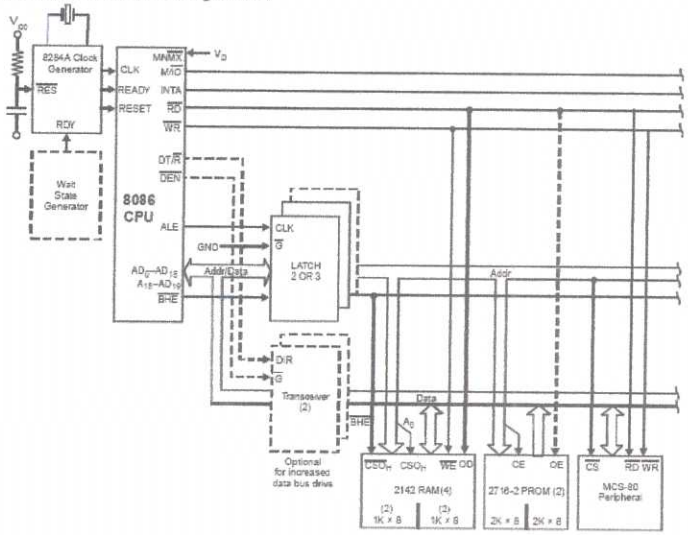
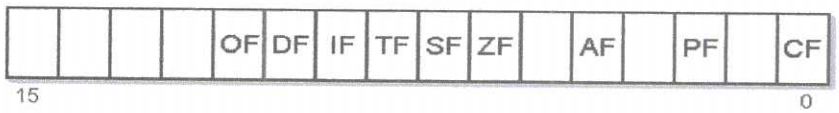
Revision: 2015 **Course Title : Advanced Microprocessors**		**Course code:6041**		
Qst No	Scoring Indicator	Split up score	Sub Total	Total
	PART A			
I(1)	16 bit processor, 20 bit address, pipe lining, 1 Mb memory (any 2)	2	2	
I(2)	NMI, INTR	2	2	
I(3)	Real-address mode, Protected mode, System management mode, Virtual-8086.	2	2	
I(4)	**Core** - A physical section of a processor which is used to process instructions.	2	2	
I(5)	**Turbo Boost in processors**-technology that allows a processor to increase its core clock speed dynamically whenever the need arises.	2	2	10
	PART B			
II(1)	8086's BIU produces the 20-bit physical memory address by combining a 16-bit segment address with a 16-bit offset address. The physical 20-bit address is calculated by shifting the segment address 4-bit left and then adding that to the offset address. For CS=4569H, and IP= 10A0H, physical address is Segment address : 45690 H Offset address :+10A0 H Physical address : 46730 H	4 2	6	
II(2)	Segmentation is the process in which the main memory is divided into different segments and each segment has its own base address. 4 segments and 4 segment registers **Advantages** Allows the memory capacity to be 1Mb although the actual addresses to be handled are of 16 bit size. Allows the placing of code, data and stack portions of the same program in different parts (segments) of the memory, for data and code protection. Permits a program and/or its data to be put into different areas of memory each time program is executed, i.e. provision for relocation. Completes the current instruction that is in progress. Pushes the flag register on the stack. Clearing the interrupt flag (IF), and the trap flag (TF) in the flag	3 3	6	

II(3)	Pushes the current code segment (CS)register contents and instruction pointer(IP) contents on the stack. Branches to the ISR using CS and IP address from the interrupt vector table. At the end of ISR ,IRET – is used to return from interrupt service to the main program	6	6	

II(4)	**ADD** – Used to add the provided byte to byte/word to word.
	Add operand1 , operand2; operand1 = operand1 + operand2
	Example:
	ADD AL, -3; AL= AL-3
	
	MUL- Unsigned multiply. when operand is a byte: AX = AL * operand.
	
	when operand is a word:(DX AX) = AX * operand.
	Example:
	MOV AL, 200 ; AL = 0C8h
	MOV BL, 4
	MUL BL ; AX = 0320h (800)
	
	SBB- Subtract with Borrow.
	Algorithm:
	operand1 = operand1 - operand2 - CF
	Example:
	STC
	MOV AL, 5
	SBB AL, 3 ; AL = 5 - 3 - 1 = 1 (any 3x2)
		6	6	

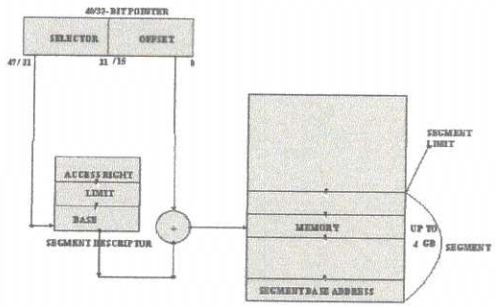
II(5)	 Super scalar architecture- Dual instruction pipeline, U and V -Two Integer execution units, with separate ALU, address generation and access to cache. Allows the execution of two instructions to be executed simultaneously in a single processor clock. U-line can n execute any Pentium instruction, V-line only executes *simple* instructions. Each line has 5 stages-Pre-fetch, Instruction Decode, Address Generation, Execute, Cache and ALU Access, and Write back.	3 3	6	

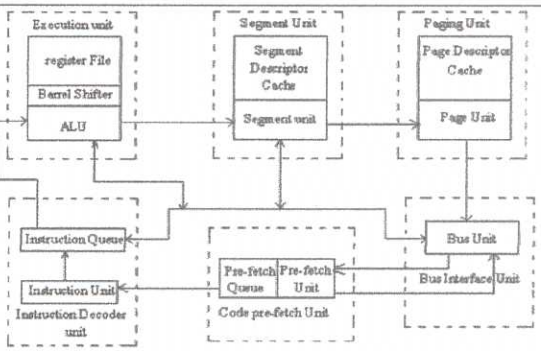
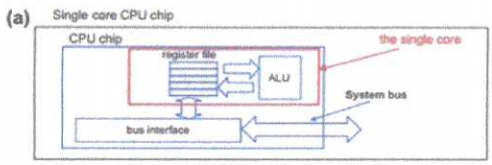
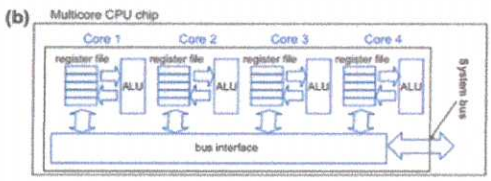
II(6)	After reset, the 80386 is in real address mode, works as a fast 8086 with 32-bit registers and data types, the default operand size is 16 bit but 32- bit operands and addressing modes may be used with the help of override prefixes. The segment size in real mode is 64k.can address 1Mbytes of physical memory using address lines A0-A19.To form a physical memory address, appropriate segment registers contents (16-bits) are shifted left by four positions and then added to the 16-bit offset address .	6	6	
II(7)	<table><tr><th></th><th>Intel Core i3</th><th>Intel Core i5</th><th>Intel Core i7</th></tr><tr><td>Number of Cores</td><td>2</td><td>4</td><td>4</td></tr><tr><td>Clock Speed Range(Several Models)</td><td>3.4GHz – 4.2GHz</td><td>2.4GHz – 3.8GHz</td><td>2.9GHz – 4.2GHz</td></tr><tr><td>urbo Boost</td><td>No</td><td>Yes</td><td>Yes</td></tr><tr><td>Cache Memory</td><td>3 – 4MB</td><td>4 – 6MB</td><td>8MB</td></tr><tr><td>Hyper-Threading</td><td>Yes</td><td>No</td><td>Yes</td></tr></table> Any 4	6	6	42
	PART C			
III(a)	**MN/~~MX~~**–indicates what mode the processor is to operate in; when it is high, it works in the minimum mode and vice-versa.	2		
	ALE - address latch enable. A positive pulse is generated each time the processor begins any operation. This signal indicates the availability of a valid address on the address/data lines.	2		
	BHE –active low-used to enable data onto the most significant half of data bus, D_8-D_{15}.	2		
	RESET - used to restart the execution, ie, processor terminate its present activity. To be active high for 4 clock cycles to RESET the microprocessor.It causes the 8086 to initialize registers DS, SS, ES, IP and flags to all zeros, and sets CS	2	8	
III(b)		3		
	BIU performs sending addresses, fetching instructions from the memory, reading data from the ports and the memory and writing data to the ports and the memory. Contains instruction Q, segment registers, IP, and adder for address generation.	4	7	15

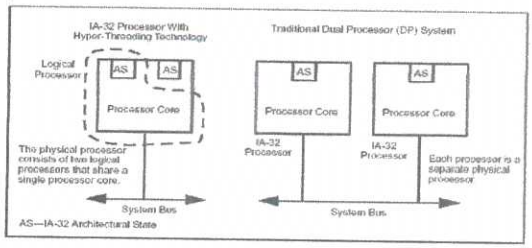
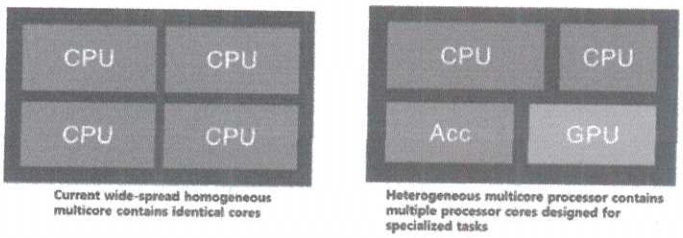
IV(a)	***Minimum Mode 8086 Configuration***  Single processor system. All control signals are generated by the processor	5 3	8	
IV(b)	 Carry flag, Auxiliary flag, Parity flag ,Zero flag ,Sign flag, Overflow ,Trap flag ,Interrupt flag ,Direction flag	4 3	7	15
V(a)	**Register addressing mode**-The instruction will specify the name of the register which holds the data to be operated by the instruction. Example - MOV CX, AX ; copies the contents of the 16-bit AX register into the 16-bit CX register **Direct addressing mode**-The addressing mode in which the effective address of the memory location is written directly in the instruction. Example: - MOV AX, [1592H], MOV AL, [0300H] **Register indirect addressing mode**-This addressing mode allows data to be addressed at any memory location through an offset address held in any of the following registers: BP, BX, DI & SI. Example:- MOV AX, [BX] ; Suppose the register BX contains 4895H, then the contents at 4895H in data segment are moved to AX ADD CX,[BX] **Based addressing mode** -In this addressing mode, the offset address of the operand is given by the sum of contents of the BX/BP registers and 8-bit/16-bit displacement. DS, or SS is used as the segment register.Useful in accessing structures. Accessing stack data without modifying SP.vcExample:- MOV DX, [BX+04], ADD AL, [BX+08] (any 4x2)	8	8	

V(b)	DATA SEGMENT
	ORG 1100H
	SUM DW 0
	CARRY DB 0
	DATA ENDS.
	CODE SEGMENT
	ASSUME CS:CODE
	ASSUME DS:DATA
	ORG l000H
	MOV AX,205AH ;Load the first data in AX register
	MOV BX,40EDH. ;Load the second data in BX register
	MOV CL,00H ;Clear the CL register for carry
	ADD AX,BX ;Add the two data, sum,wi11 be in AX
	MOV SUM,AX ;Store the sum in memory location (1104H)
	JNC AHEAD ;Check the status of carry flag
	INC CL ;If carry flag is set, increment CL by one
	AHEAD: MOV CARRY,CL ;Store the carry in memory location (1106}
	HLT
	CODE ENDS
	END
		7	7	15

VI(a)	**DB(DEFINE BYTE)** The DB directive is used to declare a byte type variable in memory. Example:- PRICES DB 49H, 98H, 29H;Declare array of 3 bytes named PRICES and initialize them with specified values. **DW (DEFINE WORD)** The DW directive is used to reserve storage locations of type word in memory. Example:-WORDS DW 1234H, 3456H; Declares an array of 2 words and initialize them with the specified values. **DD (DEFINE DOUBLE WORD)** The DD directive is used to declare a variable of type double word **Example:**- ARRAY DD 25629261H **DQ (DEFINE QUADWORD)** The DQ directive is used to tell the assembler to declare a variable 4 words in length. **Example:**-BIG_NUMBER DQ 243598740192A92BH. **DT (DEFINE TEN BYTES)** The DT directive is used to tell the assembler to declare a variable, which is 10 bytes in length . **Example:**-PACKED_BCD DT 11223344556677889900 ; initialize the 10 bytes with the values 11, 22, 33, 44, 55, 66, 77, 88, 99, and 00.(any 4x2)	8	8	
VI(b)	**Divide-By-Zero Interrupt-Type 0:** The 8086 will automatically do a type 0 interrupt if the result of a DIV operation or an IDIV operation is too large to fit in the destination			

	register. **Single Step Interrupt-Type 1:** When the trap flag(TF) is set to 1, processor generate a type 1 interrupt after execution of every instruction. Useful in debugging. **Non-maskable Interrupt-Type 2:** The 8086 will automatically do a type 2 interrupt response when it receives a low to high transition on its NMI pin **Breakpoint Interrupt-Type 3:** The type 3 interrupt is produced by execution of the INT3 instruction. Used to implement a breakpoint function in a system. **Overflow Interrupt-Type4:** The 8086 overflow flag will be set if the signed result of an arithmetic operation on two signed numbers is too large to be represented in the destination register or memory location.	7	7	15
VII(a)	In PVAM, each physical address is represented by 48 bit Virtual address- Offset is 32 and Selector is 16 bit. The Selector is used to fetch an 8 byte Segment Descriptor from the Descriptor table, which contain the 32 bit Segment base address, Segment limit, and Access Rights byte of the segment. The Offset is added with Segment Base address to calculate Linear address. This linear address is used as physical address.  Protected Mode Addressing Without Paging Unit	5 3	8	
VII(b)	Microprocessor is packaged in 387-pin PGA (pin grid array). L1 cache of 16 KB.(8K+8K) It includes L2 cache of 256/512 KB It uses 12 stage pipeline. Supports speculative execution. Multiple branch prediction. 36 address lines- capable of addressing 64GB of memory Process two integer instructions and one floating-point instruction simultaneously.2M paging Built-in error correction circuit (ECC) Can correct one-bit error and detect two-bit error (any 7)	7	7	15

VIII(a)	 1. Bus Interface unit 2. Code Prefetch Unit 3. Instruction Decode Unit 4. Segmentation Unit 5. Paging Unit 6. Execution Unit explain	5 5	10	
VIII(b)	The segmentation scheme may divide the physical memory into a variable size segments but the paging divides the memory into a fixed small size pages. The segments are supposed to be the logical segments of the program, but the pages do not have any logical relation with the program. The pages are fixed size portions of the program module or data. Segmentation -difficult for memory swapping. May cause, fragmentation. The advantage of paging scheme is that the complete segment of a task need not be in the physical memory at any time. Memory requirement of the task is substantially reduced, using the available memory for other tasks.	5	5	15
IX(a)	In single core, only one ALU, register file etc   In multi core, single physical processor contains the core logic of two or more processors, packaged into a single integrated circuit (IC). Each core contains its own dedicated processing resources similar to an individual CPU.	4 4	8	

IX(b)	**Hyper Threading Technology** Hyper-Threading allows a single microprocessor to act like two separate processors to the operating system, so that can execute two concurrent streams (or thread s) of instructions sent by the operating system, allows more work to be done by the processor during each clock cycle . claims up to a 30% performance improvement compared with an otherwise identical, non-simultaneous multithreading processor.  	4		
X(a)	 **Homogeneous milticore** This involves geometrically increasing the number of cores , either duplicating or quadrupling the same core, and interconnecting them to produce a single, more powerful core **Heterogeneous multicore** It is a combination of a large core, plus several low-power cores, ie, multiple cores on a single chip, but those cores might be of different designs of both high and low complexity.	3 3 5	7 8	15
X(b)	More processing power ,Usually faster work, because of parallel processing with threads . Smaller energy consumption at the same clock rate (compared to single core), which is good for mobile . Less heat production because of less energy consumption CPUs on a single die means that signals between different CPUs travel shorter distances, and therefore those signals degrade less, allow more data transfer than multi-chip designs. Multi-core CPU designs require much less space than multi-chip designs.	7	7	15