Course code (15) 5131

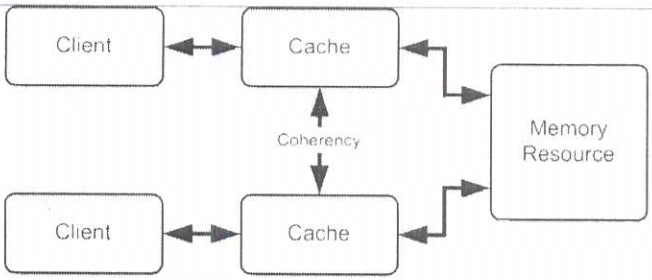
(MICROPROCESSORS AND INTERFACING)

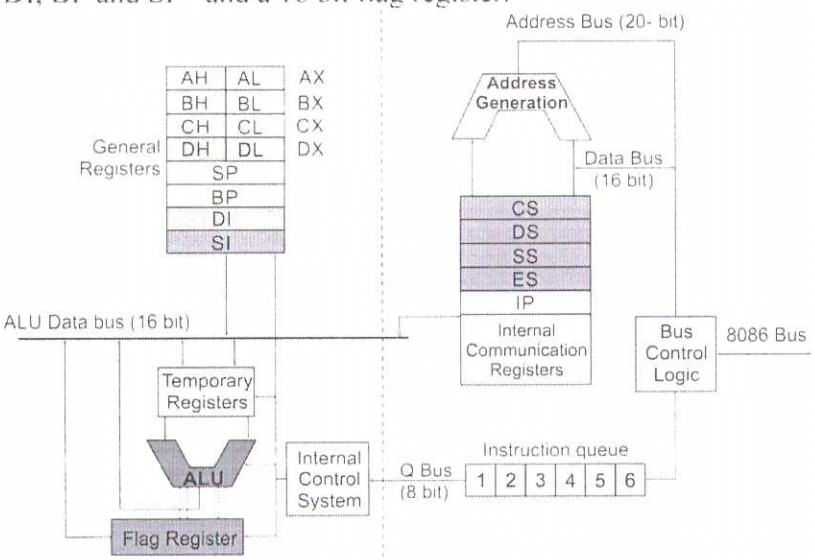
Question No.	Scoring Indicator	Split up Score	Total
I	PART - A		
1	CS,ES,DS,SS registers	0.5x4	2
2	It is used to select the mode of operation of 8086 – Minimum or maximum mode	2	2
3	<Macro name> MACRO [Parameter List] Instructions or body of macro ENDM	2	2
4	An interrupt is a signal sent to the processor that interrupts the current process, generated by a hardware device or a software program.	2	2
5	First 32 bit microprocessor and faster execution speed is obtained, It is 132 pin IC, It is capable of addressing 4 GB of physical memory and 64 TB of virtual memory, It provides multitasking support, memory management ,pipe lined architecture and high speed bus interface in a single IC.	1x2	2
II	PART - B		
1	Segmentation is the process in which the main memory of the computer is logically divided into different segments and each segment has its own base address. It is basically used to enhance the speed of execution of the computer system, so that the processor is able to fetch and execute the data from the memory easily and fast. The Bus Interface Unit (BIU) contains four 16 bit special purpose registers (mentioned below) called as Segment Registers. • Code segment register (CS): is used for addressing memory location in the code segment of the memory, where the executable program is stored. • Data segment register (DS): points to the data segment of the memory where the data is stored. • Extra Segment Register (ES): also refers to a segment in the memory which is another data segment in the memory. • Stack Segment Register (SS): is used for addressing stack segment of the memory. The stack segment is that segment of memory which is used to store stack data..	6	6
2	LOOP Instruction: It loops through a sequence of instructions until CX=0.		

	General form : LOOP label ; Decrease CX, jump to label if CX not zero.
	
	Example : Display all digits from 0 to 10
	
	mov cx,10
	
	mov DL,'0'
	
	repeat1:mov ah,02h
	
	int 21h
	
	inc DL
	
	loop repeat1
		3
		
		3
			6

3	**AND** The bitwise AND operation returns 1, if the matching bits from both the operands are 1, otherwise it returns 0. Format is : AND operand1, operand2 **NOT** Used to invert each bit of a byte or word. Format is : NOT operand1 **OR** The bitwise OR operator returns 1, if the matching bits from either or both operands are one. It returns 0, if both the bits are zero. Format is : OR operand1, operand2 **XOR** Used to perform Exclusive-OR operation over each bit in a byte/word with the corresponding bit in another byte/word. Format is: XOR operand1, operand2	4x1.5	6
4	An interrupt is a condition that halts the microprocessor temporarily to work on a different task and then return to its previous task. Whenever an interrupt occurs the processor completes the execution of the current instruction and starts the execution of an **Interrupt Service Routine (ISR)** or **Interrupt Handler**. After every instruction cycle, the processor checks if any interrupt is enabled. If so the following steps are carried out. a. The flag register is pushed onto the stack. b. The interrupt flag is disabled. (IF = 0) c. The trap flag is disabled (TF = 0) d. The CS register is pushed onto the stack e. The IP register is pushed onto the stack. f. Control is transferred to the interrupt vector.	6	6

	g. The program corresponding to the ISR is executed. The last instruction in ISR will be IRET. h. IP and CS is popped out from stack. i. The flag register is popped out. j. Control returns to the point at which it had left off.		
5	**Bit set reset (BSR) mode** – If MSB of control word (D7) is 0, PPI works in BSR mode. In this mode only port C bits are used for set or reset. **Input-Outpt mode** – If MSB of control word (D7) is 1, PPI works in input-output mode. This is further divided into three modes 1. **Mode 0** – In this mode all the three ports (port A, B, C) can work as simple input function or simple output function. There is no interrupt handling capacity. 2. **Mode 1** – Handshake I/O mode or strobbed I/O mode. In this mode either port A or port B can work as simple input port or simple output port, and port C bits are used for handshake signals before actual data transmission. 3. **Mode 2** – Bi-directional data bus mode. In this mode only port A works, and port B can work either in mode 0 or mode 1. 6 bits of port C is used as handshake signals.	3x2	6
6	MMX MMX is a Pentium microprocessor from Intel that is designed to run faster when playing multimedia applications. PC with an MMX microprocessor runs a multimedia application up to 60% faster than one with a microprocessor having the same clock speed but without MMX. MMX microprocessor runs other applications about 10% faster. Hyper-Threading Hyper-Threading is a technology used by some Intel microprocessor s that allows a single microprocessor to act like two separate processors to the operating system and the application program s that use it. With Hyper-Threading, a microprocessor's "core" processor can execute two concurrent streams or threads of instructions sent by the operating system. Having two streams of execution units to work on allows more work to be done by the processor during each clock cycle .	3 3	6
7	Cache coherency is a situation where multiple processor cores share the same memory hierarchy, but have their own L1 data and instruction caches. Incorrect execution could occur if two or more copies of a given cache block exist, in two processor's caches, and one of these blocks is modified.		

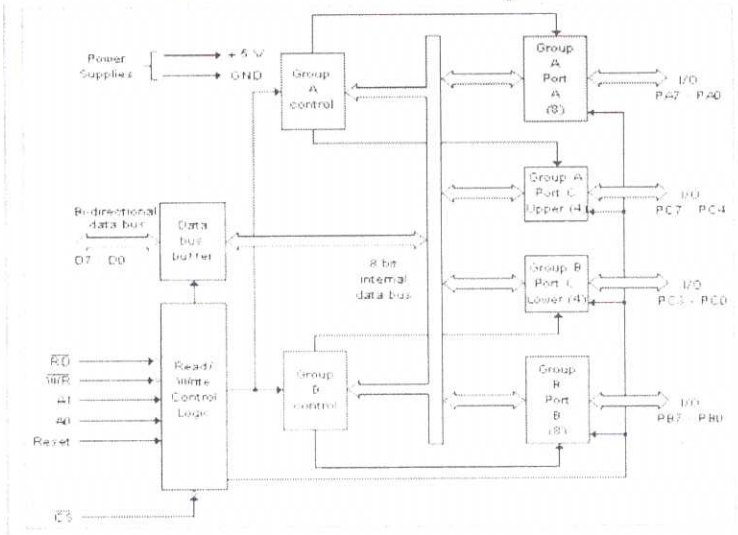
		6	6
	PART C		
III a	The way in which an operand is specified in an instruction is called Addressing Mode. Register addressing Source and destination operands are registers Eg: MOV AL,AH Immediate addressing 8-bit or 16-bit data is specified as part of the instruction. Eg: MOV DL, 08H Direct addressing Here, the effective address of operand is stored is given directly in the instruction. Eg: MOV BX, [1354H] Register Indirect addressing Register which holds the effective address (EA) will be specified in the instruction. Registers used to hold EA are BX, BP, DI and SI. Eg: MOV AL,[BX] Indexed Addressing SI or DI register is used to hold an index value for memory data and a signed 8-bit or unsigned 16-bit displacement will be specified in the instruction. Eg: MOV CX, [SI + 0A2H] Based Indexed addressing Base register and an index register together carry the effective address. The contents of these two registers are added and called the effective address. Eg: MOV DX, [BX + SI + 0AH]	4x2	8
b	➢ 8086 is the first 16-bit processor having 16-bit ALU, 16-bit registers, internal data bus, and 16-bit external data bus resulting in faster processing. ➢ It has 20 bit address bus (A0 to A19). Can address up to $2^{20} = 1$ megabytes of memory space. ➢ It has an instruction queue, which is capable of storing six instruction bytes from the memory resulting in faster processing. ➢ It uses two stages of pipelining, i.e. Fetch Stage and Execute Stage, which improves performance. ➢ It has 256 vectored interrupts.	7	7

	➢ It supports two modes of operation, i.e. Maximum mode and Minimum mode.		
IV	The internal block diagram has been partitioned into two logical units - Bus Interface Unit (BIU) and Execution Unit (EU). Bus Interface Unit (BIU) Contains Instruction Queue, Segment Registers, Bus control Logic etc. Segment Registers - Memory is divided into four segments namely Data ,Code, Stack and Extra segments. Segmentation keeps data and code separately. Instruction queue - 6 byte instruction code are pre fetched from the memory ahead of time to speed up the execution by overlapping instruction fetch with execution. This mechanism is known as pipelining Execution Unit EU decodes and executes instructions. Contains ALU, Control unit, Internal bus ,Registers. A decoder in the EU control system translates instructions. Registers include four 16 bit general purpose or scratch pad registers - AX,BX,CX and DX, pointer and index registers – SI, DI, BP and SP and a 16 bit flag register.  	8 7	15
V a	**Procedure** is a part of code that can be called from your program to do some specific task(like function in c). Procedures make program more structural and easier to understand. The syntax for procedure declaration: **procname PROC** <instructions> RET **procname ENDP** **CALL** instruction is used within main to call a procedure Syntax: CALL procname Example	4 4	8

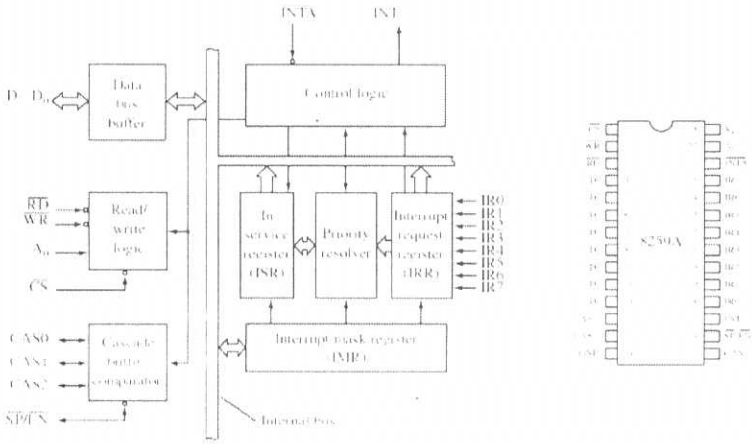
V
b
	.MODEL SMALL
	.DATA
	NUMBERS DB 10 DUP(0)
	.CODE
	LEA SI,NUMBERS
	MOV CX,05H
	Repeat :MOV AH,01H
	INT 21H
	SUB AL,30H
	CALL SQ
	MOV [SI],AL
	INC SI
	LOOP Repeat
	// DISPLAY NUMBERS
	//EXIT
	END
	
	SQ PROC
	MOV AL,BL
	MUL BL
	RET
	SQ ENDP
		7	7

VI a	**MOV** Used to copy a byte or word from the provided source to provided destination. The source and destination can be either memory or register or an immediate data. Example:- MOV AL,BL, MOV AH,09H, MOV BX,LOC **XCHG** Used to exchange the data between two locations. Example: XCHG AX, BX **PUSH reg16/ mem** Push the contents of register or memory location to top of stack. Example: PUSH AX **POP reg16/ mem** Pop top of stack to a 16 bit register or to a memory location. Example: POP BX	4x2	8

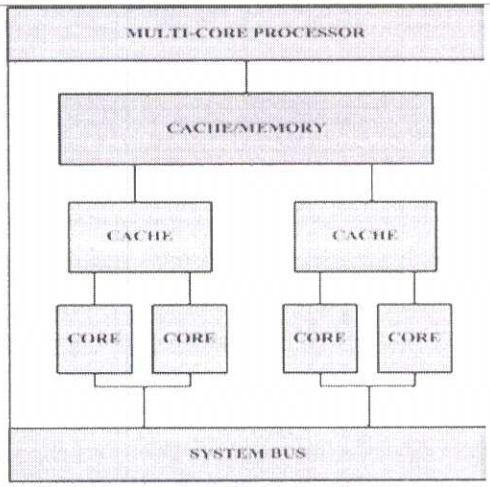
b	.MODEL SMALL
	.DATA
	WRD1 DB "SAMSON"
	WRD2 DB "SAMRON "
	MSG1 DB "SAME"
	MSG2 DB "NOT SAME"
	.CODE
	MOV AX,@DATA
	MOV DS,AX
	MOV ES,AX
	LEA SI,WRD1
	LEA DI,WRD2
	MOV CX,6
	CLD
	REPE CMPSB
	JNZ L1
	LEA DX,MSG1
	JMP DISP
	L1: LEA DX,MSG2
		7	7

	~~JMP DISP~~ ~~L1: LEA DX,MSG2~~ DISP: MOV AH,09H INT 21H END		
VII a	*8255 – Internal block diagram*  Programmable Peripheral Interface is used_Interface slow I/O devices to the fast processor to achieve an efficient data transfer between them. It consists of data bus buffer, control logic and Group A and Group B controls. Data Bus Buffer: This tri-state bi-directional buffer is used to interface the internal data bits of 8255 to the system data bus. Control Logic: The control logic block accepts control bus signals as well as inputs from the address bus, and issues commands to the individual group control blocks (Group A control and Group B control). Group A and Group B Controls: Each of the Group A and Group B control blocks receives control words from the CPU and issues appropriate commands to the ports associated with it. 8255 has 3 ports – port A, port B and port C . These ports have pins connecting to it to external devices. Port A has 8 pins PA0 – PA7 . Same for ports B and C. Port C lower and port C upper and port C can work in either BSR (bit set rest) mode or in mode 0 of input-output mode of 8255.	4 4	8
VII b	<u>Dedicated interrupt types</u> <u>INT 0 (Divide by zero error)</u> This interrupt is generated automatically if the quotient register is not large enough to contain the quotient when performing division. The ISR can be either display a message indicating 'divide overflow' or to write a program to increase the size of the quotient register.		

	INT 1 (Single Stepping) It is for single stepping or trace which is needed for debugging. It puts microprocessor in single stepping mode. The trap flag will be reset as part of the interrupt response, so the ISR can execute without stopping after each program line. INT 2 (Non Maskable Interrupt) When an interrupt is received on the pin NMI of the processor , type 2 interrupt occurs. For eg: Save program and data when power fails. An external circuit is used to detect the power failure and send an interrupt signal to the 8086 NMI pin. INT 3 (Break point Interrupt) Breakpoint interrupt is useful for debugging. Breakpoints are inserted to check the contents of registers and memory during debugging. INT 4 (Overflow Interrupt) This interrupt corresponds to overflow flag (OF). If OF = 1 , this interrupt occurs . For this, INTO (Interrupt on Overflow) must be written in the program segment which may cause the overflow flag to be set.	7	7
VIII	8259 is a Standard chip manages interrupt requests. PIC has 8 interrupt request lines IR0 to IR7 on which peripherals can place their interrupt requests. The processor sends back an INTA signal , then PIC sends the interrupt type number to the corresponding interrupt to 8086. PIC has to do resolving priority also when multiple interrupt comes. The Block Diagram consists of 8 blocks which are – Data Bus Buffer - used as a mediator between 8259 and 8085/8086 microprocessor by acting as a buffer. Read/Write Logic - This block is responsible for the flow of data depending upon the inputs of RD and WR. Cascade Buffer Comparator – To increase the Interrupt handling capability, cascade buffer can be used to cascade more number of pins. Control Logic - controls the functioning of every block. Interrupt request register (IRR) – It stores all the interrupt level which are requesting for Interrupt services. Interrupt service register (ISR) – It stores the interrupt level which are currently being executed. Interrupt mask register (IMR) – It stores the interrupt level which have to be masked by storing the masking bits of the interrupt level. Priority resolver – Set the priority of interrupts and according to the priority of the interrupts, interrupt with highest priority is set in ISR register.	8	

		7	15
IX a	◦ Fifth generation x86 processor developed in 1993. ◦ 32 bit processor, 64 bit data bus and 32 bit address bus to address up to 4 GB of physical memory space. ◦ 32 bit internal registers. ◦ Super scalar architecture - Superscalar means it has more than one execution unit.Two pipe lined integer execution units, U and V pipes. ◦ Faster floating point unit – FPU has an 8 stage pipeline to speed up operations. ◦ Separate data and instruction cache - There are separate instruction and data caches with 8kb in size and 32 byte line size. ◦ Branch prediction unit - Useful in preventing pipeline stalling in the case of branch instruction. ◦ System management mode - System management mode is for efficient power management. k.	8	8
IX b	Pipeline hazards are situations that prevent the next instruction in the instruction stream from executing during its designated clock cycles. There are primarily three types of hazards: <u>Data Hazards :</u> A data hazard is any condition in which either the source or the destination operands of an instruction are not available at the time expected in the pipeline. As a result of which some operation has to be delayed and the pipeline stalls. <u>Control Hazards or instruction Hazards</u> : Arises when memory is accessed at the same time by two instructions. One instruction may need to access the memory as part of the Execute or Write back phase while other instruction is being fetched. <u>Structural Hazards</u> : This situation arises mainly when two	1 2 2	7

	instructions require a given hardware resource at the same time and hence for one of the instructions the pipeline needs to be stalled.	2	
X a	Operating modes of 80386 Three operating modes - **Real mode, virtual 8086 mode** and **Protected Virtual address mode** Real Mode ◦ When the processor is reset or after power on, it is in the real mode. ◦ In real mode ,it acts as faster 8086 processor with a few new additional instructions. ◦ The main feature is it has access to only to 1 MB of physical memory and address calculation is done as in 8086.The segment size is 64KB. Protected virtual address mode (PVAM) ◦ The 8086 works as a 32 bit processor and all instructions and features of 80386 are available in this mode. ◦ While working in PVAM, the processor can switch to Virtual 8086 mode to run 8086 applications and then return to PVAM. ◦ In PVAM, it has 4 GB of physical memory and 64 TB of virtual memory address space. ◦ Each physical address is represented by 48 bit virtual address. Virtual 8086 mode. ◦ 80386 has a virtual mode which allows switching between real and protected modes. ◦ Time slices are allotted for programs and switching between real and protected mode will occur continuously. ◦ Virtual 8086 mode divides the computer into multiple address spaces and maintains virtual registers for each virtual machine. ◦ Virtual mode is entered by setting the VM (Virtual Machine) bit in the EFlags register.	8	8
X b	Multicore processing A multicore processor is a single computing component comprised of two or more CPUs that execute the instructions. The individual cores can execute multiple instructions in parallel, increasing the performance of software which has been written to take advantage of the unique architecture. Different multicore processor architectures vary in terms of Number of cores, Number and level of caches, instruction and data caches, bus architectures, Isolation etc.	4	7

		 MULTI-CORE PROCESSOR CACHE/MEMORY CACHE CACHE CORE CORE CORE CORE SYSTEM BUS	3	