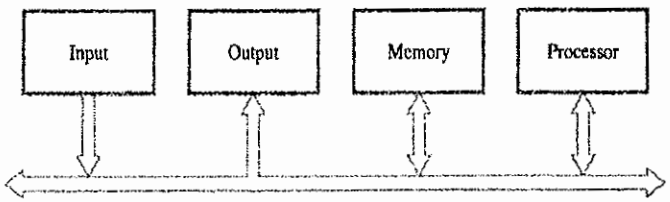
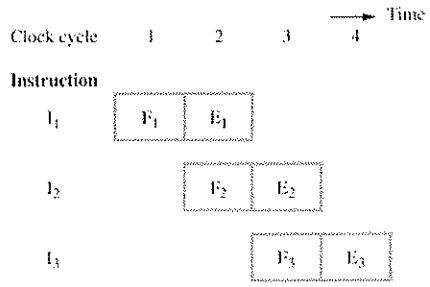
Scoring Indicators

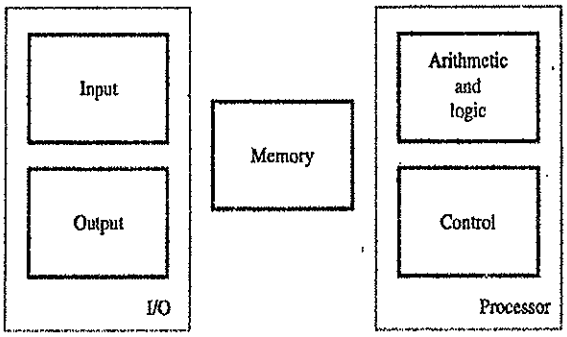
Course Name : Computer Organization
Course Code : 3131 **QID :**

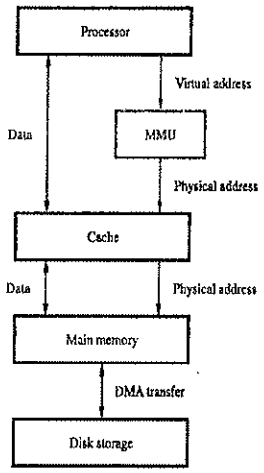
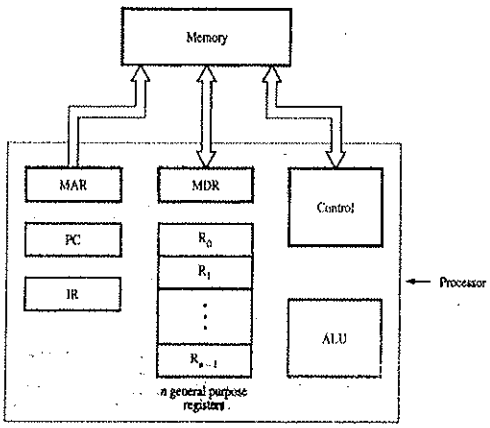
Q. No.	Scoring Indicators	Split score	Sub Total	Total score
	PART A			9
I.1	Instruction Register (IR)	1	1	
I.2	To hold the address of memory from which data or instruction to be read	1	1	
I.3	Direct Memory Access (DMA)	1	1	
I.4	Universal Serial Bus (USB)	1	1	
I.5	True	1	1	
I.6	Control Store	1	1	
I.7	Control signal used to indicate that the memory operation (read or write) is completed	1	1	
I.8	Trap Flag (TF)	1	1	
I.9	20 bits	1	1	
	PART B			24
II.1	 All units connected to single bus. Bus can be used only for one transfer at a time, only two units can actively use the bus at any given time. Bus control lines are used to arbitrate multiple requests for the use of bus. Low cost and flexible for attaching peripheral devices.	Fig: 1.5 1.5	3	
II.2	Flash Memory: a flash cell based on single transistor controlled by trapped charge, it is possible to read the contents, previous contents of the block can be erased, have greater density, higher capacity, lower cost per bit, consume less power EEPROM: can be programmed and erased electrically, contents can be erased selectively, different voltages needed for erasing, writing and reading the stored data	1.5 1.5	3	

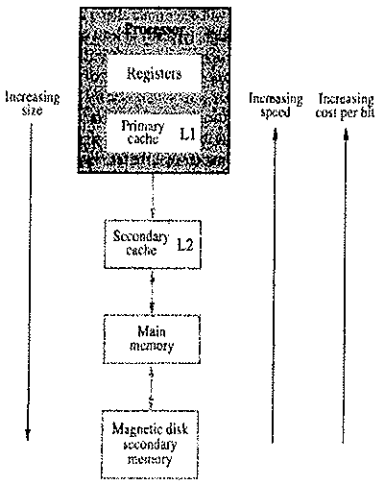
<table>
<tr>
<td>II.3</td>
<td>
<table>
<tr><th>Interrupt</th><th>Polling</th></tr>
<tr><td>In interrupt, the device notices the CPU that it requires its attention.</td><td>In polling, CPU steadily checks whether the device needs attention.</td></tr>
<tr><td>In interrupt, the device is serviced by interrupt handler.</td><td>While in polling, the device is serviced by CPU.</td></tr>
<tr><td>Interrupt can take place at any time.</td><td>Whereas CPU steadily ballots the device at regular interval.</td></tr>
<tr><td>In interrupt, interrupt request line is used as indication for indicating that device requires servicing.</td><td>While in polling, Command ready bit is used as indication for indicating that device requires servicing.</td></tr>
<tr><td>In interrupts, processor is simply disturbed once any device interrupts it.</td><td>processor waste countless processor cycles by repeatedly checking the command-ready little bit of each device.</td></tr>
</table>
</td>
<td>3</td><td>3</td><td></td>
</tr>
<tr>
<td>II.4</td>
<td>It provides a simple, low cost, easy to interconnect, accommodate wide range of data transfer for I/O devices, higher data transfer bandwidth, more ports can be added, plug and play mode of operation – able to detect new devices, identify the appropriate device-driver software and other facilities needed to service the device, establish appropriate addresses and logical connects to enable them to communicate.</td>
<td>3</td><td>3</td><td></td>
</tr>
<tr>
<td>II.5</td>
<td>First type consists of array of mechanical switches mounted on a PCB. The switches are organized in rows and columns and connected to a microcontroller on the board. When a switch is pressed, the controller identifies the row and column and determines which key is pressed.
Second type uses a flat structure consisting of three layers – top layer: a plasticized material, with key positions marked on the top surface and conducting traces on the underside, middle layer: made of rubber, with holes at key positions, bottom layer – metallic with raised bumps at key position. When pressure is applied, the trace comes in contact with bump and completes the circuit. The current flows in the circuit is sensed by microcontroller.</td>
<td>1.5

1.5</td><td>3</td><td></td>
</tr>
<tr>
<td>II.6</td>
<td>Consider the instruction Move (R1), R2. The control signals are activated as follows:
1. $R1_{out}$, MAR_{in}, Read
2. MDR_{inE}, WMFC
3. MDR_{out}, $R2_{in}$</td>
<td>3</td><td>3</td><td></td>
</tr>
</table>

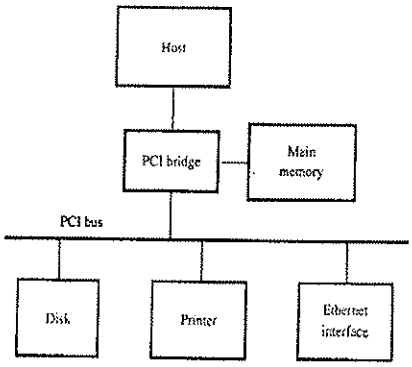
II.7	Pipelining is an effective way to organize concurrent activity by breaking down the instruction into multiple simple tasks which can be carried out independently. Multiple instructions are overlapped during the execution.  	3	3	
II.8	Following is the list of conditional flags – 1. Carry flag – This flag indicates an overflow condition for arithmetic operations. 2. Auxiliary flag – When an operation is performed at ALU, it results in a carry/barrow from lower nibble (i.e., D0 – D3) to upper nibble (i.e., D4 – D7), then this flag is set, i.e., carry given by D3 bit to D4 is AF flag. 3. Parity flag – This flag is used to indicate the parity of the result, i.e., when the lower order 8-bits of the result contains even number of 1's, then the Parity Flag is set. For odd number of 1's, the Parity Flag is reset. 4. Zero flag – This flag is set to 1 when the result of arithmetic or logical operation is zero else it is set to 0. 5. Sign flag – This flag holds the sign of the result, i.e., when the result of the operation is negative, then the sign flag is set to 1 else set to 0. 6. Overflow flag – This flag represents the result when the system capacity is exceeded	0.5*6	3	
II.9	In Register Addressing mode both the operands are registers. No memory access is involved. Example: MOV AX, BX ; copy the contents of BX to AX	3	3	
II.10	Features of the Pentium processor: 32-bit microprocessor, 32-bit address bus, 64-bit data bus Superscalar architecture – two pipelines integer units, capable of less than one clock per instruction, pipelines floating point unit Separate code and data caches – 8K code cache, 8K write back data Advanced design features – static and dynamic branch prediction, allow 2MB and 4MB page sizes, system management mode	3	3	
	PART C			42

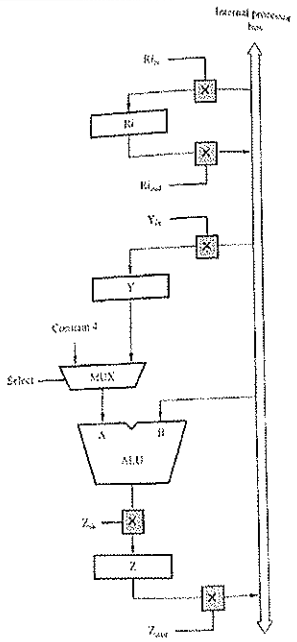
III.1		2	7	
	Input unit: - The source program/high level language program/coded information/simply data is fed to a computer through input devices keyboard is a most common type. Whenever a key is pressed, one corresponding word or number is translated into its equivalent binary code over a cable & fed either to memory or processor. Joysticks, trackballs, mouse, scanners etc are other input devices. Memory unit: - Its function into store programs and data. It is basically to two types 1. Primary memory, 2. Secondary memory Primary memory is exclusively associated with the processor and operates at the electronics speeds programs must be stored in this memory while they are being executed. The memory contains a large number of semiconductors storage cells. Secondary memory: - Is used where large amounts of data & programs have to be stored, particularly information that is accessed infrequently. Examples: - Magnetic disks & tapes, optical disks (i.e., CD-ROM's), floppies etc., Arithmetic logic unit (ALU): - Most of the computer operators are executed in ALU of the processor like addition, subtraction, division, multiplication, etc. the operands are brought into the ALU from memory and stored in high-speed storage elements called register. Then according to the instructions, the operation is performed in the required sequence. Output unit: - These are the counterparts of input unit. Its basic function is to send the processed results to the outside world. Examples: - Printer, speakers, monitor etc. Control unit: - It effectively is the nerve center that sends signals to other units and senses their states. The actual timing signals that govern the transfer of data between input unit, processor, memory and output unit are generated by the control unit.	5		

III.2	 Techniques that automatically move program and data blocks into the physical memory when they are required for execution are called virtual memory. Programs and processor reference an instruction and data space that is independent of the available physical main memory space. The binary address that the processor issues for either instruction or data are called virtual or logical addresses. These addresses are translated into physical addresses by a combination of hardware and software components. A special hardware unit called the memory management unit (MMU) translates the virtual addresses into physical addresses. When the desired data or instructions are in the main memory these data are fetched from the appropriate location. If the data are not in the main memory the MMU causes the operating system to bring the data into the memory from the disk.	2 5	7	
III.3	 The transfer between memory and processor are started by sending the address of the memory location to be accessed to the memory unit and issuing the appropriate control signals. The data are then transferred to or from the memory. The processor contains registers that are used for several purposes. IR hold the current instruction being executed; its output is available in the control circuits. PC keep track of the execution of a program. It contains the memory address of the next instruction. General purpose registers are used. MAR and	2 5	7	

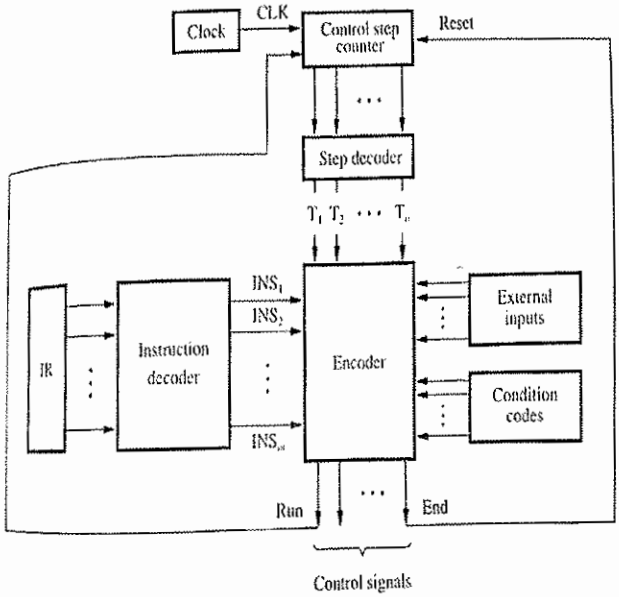
	MDR facilitate the communication with the memory. MAR holds the address of the location to be accessed, MDR contains the data to be written into or read out of the memory.			
III.4	 Different types of memory units are employed in a computer. The fastest access is to data held in processor registers but the size is very small. The next level if relatively small amount of memory that can be implemented directly on the processor chip, called processor cache which holds the copes of instructions and data stored in main memory, primary cache or L1 cache. Secondary cache is placed between primary cache and man memory, referred as L2 cache. It is usually implemented using SRAM chips. Main memory is larger memory implemented using DRAM chips, significantly slower than cache memory. Disk devices provided huge amount of inexpensive storage. They are very slow compared to the semiconductor devices.	3 4	7	

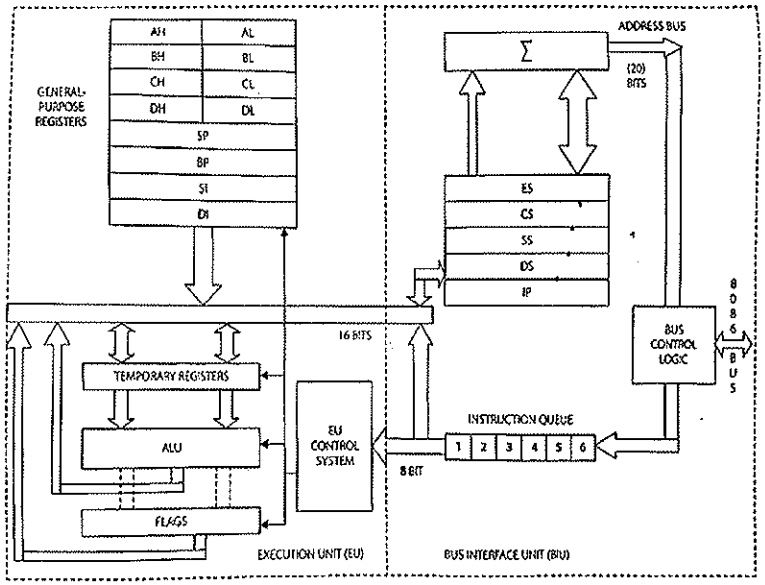
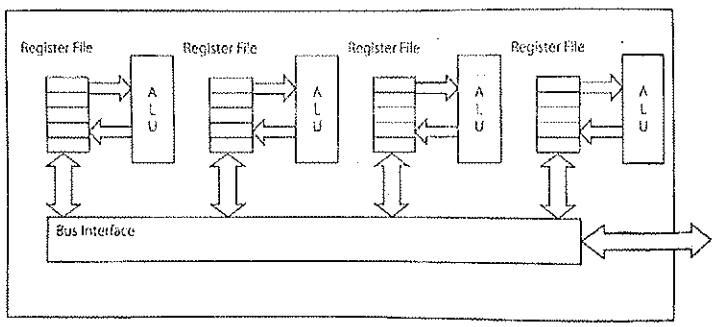
III.5	I/O devices operate at speed much lower than that of processor. For keyboard a status flag SIN is included in the interface circuit as part of the status register. This flag is set to 1 when a character is entered at the keyboard and cleared to 0 once the character is read by the processor. This is accomplished in a program loop that repeatedly reads the status register and checks the status of SIN. Similar procedure is used to control output operations using an output status flag, SOUT.
	`Move #LINE,R0 Initialize memory pointer.`
	`WAITK TestBit #0,STATUS Test SIN.`
	`Branch=0 WAITK Wait for character to be entered.`
	`Move DATAIN,R1 Read character.`
	`WAITD TestBit #1,STATUS Test SOUT.`
	`Branch=0 WAITD Wait for display to become ready.`
	`Move R1,DATAOUT Send character to display.`
	`Move R1,(R0)+ Store charater and advance pointer.`
	`Compare #$0D,R1 Check if Carriage Return.`
	`Branch≠0 WAITK If not, get another character.`
	`Move #$0A,DATAOUT Otherwise, send Line Feed.`
	`Call PROCESS Call a subroutine to process the the input line.`
		7	7	

	The program reads a line of characters from the keyboard and stores it in a memory buffer starting at location LINE. Then it calls subroutine PROCESS to process the input line. As each character is read it is echoed back to the display. R0 is used as a pointer to memory buffer area. The contents of R0 are updated using the autoincrement addressing mode so that successive characters are stored in successive memory locations. If the character is carriage return (CR), then a line feed character is end to the display to move down the cursor and subroutine PROCESS is called. Otherwise, the program loops back to wait for another character from the keyboard. This program illustrates program-controlled I/O in which the processor repeatedly checks a status flag to achieve the required synchronization between the processor and an input or output device.			
III.6	 The PCI bridge provides a separate physical connection for the main memory. The bus may be further divided into segments connected via bridges. At any given time, one device is the bus master and it initiate the data transfer by issues read and write commands. A master is called an initiator in PCI which is either a processor or DMA controlled. The addressed device that responds to read and write commands is called a target. Consider a bus transaction in which processor reads 32-bit words from memory, the initiator is the processor and the target is the memory. A clock signal provides the timing reference used to coordinate different phases of a transaction. All signal transitions are triggered by the rising edge of the clock. In clock cycle 1, the processor asserts FRAME# to indicate the beginning of a transaction. It sends the address on the AD lines and a command on the C/BE# lines which indicate a read operation is requested and the memory address space is being used. In clock cycle 2, the processor removes the address and disconnects its drives from the AD lines. The selected target enables its drivers on the AD lines and fetches the requested data to be placed on the bus during cycle 3. It asserts DEVSEL# and maintains it in the asserted state until the end of the transaction. The initiator asserts IRDY# to indicate that it is ready to receive data. If the target has data ready to send, it asserts	2 5	7	

	TRDY# and sends a word of data. The initiator loads the data into its input buffer at the end of the clock cycle.			
III.7	Flat panel displays are tinner and lighter weight, provides higher resolution, developed using liquid-crystal panels, plasma panels and electroluminescent panels. Liquid-crystal panels: constructed by sandwiching a thin layer of liquid crystal between two transparent plates. Top plate has transparent electrodes deposited on it, and the back plate is a mirror. By applying appropriate electrical signals across the plates, various segments of the liquid crystal can be activated, causing changes in their light-diffusing properties. An image is produced by passing light through selected segments of the liquid crystal and then reflecting it back from the mirror to the viewer. Used in watches, calculators, notebook computers etc. In thin-film transistor (TFT) display the transistors are prepared on a thin film deposited on one of the plates. This provides faster response and better display. Plasma panels consists of two glass plates separated by a thin gap filles with a gas such as neon. The electrodes on the two plates run at right angles to each other. A voltage pulse applied between two electrodes causes a small segment of gas at the intersection of the two electrodes to glow. The glow of gas segments is maintained by a lower voltage. As similar pulsing arrangement is used to selectively turn points off. Electroluminescent panels use a thin layer of phosphor between two electrically conducting panels. The image is created by applying electrical signals to the plates, making the phosphor glow.	1 3 3	7	
III.8	 The ALU is a combinational circuit that has no internal storage. It performs arithmetic and logic operations on the two operands applied to its A and B inputs. One of the operands is the output of the	3 4	7	

	multiplexer MUX and the other operand is obtained directly from the bus. The result produced by the ALU is stored temporarily in register Z. The sequence of operations to add the contents of register R1 to R2 and store the result in register R3 is [R3] ← [R1] + [R2] 1. $R1_{out}$, Y_{in} 2. $R2_{out}$, Select Y, Add, Z_{in} 3. Z_{out}, $R3_{in}$ In step 1, the output of register R1 and the input of register Y are enabled causing the contents of R1 to be transferred over the bus to Y. In step 2, the multiplexer's Select signal is set to Select Y, causing the multiplexer to gate the contents of register Y into input A of the ALU. At the same time, the contents of register R2 are gated onto the bus and to input B. The function performed by the ALU depends on the signals applied to its control lines.			
III.9	The execution of instruction Add (R3), R1 requires the following actions: 1. Fetch the instruction 2. Fetch the first operand 3. Perform the addition 4. Load the result into R1 The sequence of control steps required to perform these operations is given below: **Step** / **Action** **1** PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in} **2** Z_{out}, PC_{in}, Y_{in}, WMFC **3** MDR_{out}, IR_{in} **4** $R3_{out}$, MAR_{in}, Read **5** $R1_{out}$, Y_{in}, WMFC **6** MDR_{out}, Select Y, Add, Z_{in} **7** Z_{out}, $R1_{in}$, End Instruction execution proceeds as follows: Step 1: The instruction-fetch operation is initiated by loading contents of PC into MAR & sending a Read request to memory. The Select signal is set to Select4, which causes the MUX to select constant 4. This value is added to operand at input B (PC's content), and the result is stored in Z. Step 2: Updated value in Z is moved to PC. This completes the PC increment operation and PC will now point to next instruction. Step 3: Fetched instruction is moved into MDR and then to IR. The step 1 through 3 constitutes the Fetch Phase. At the beginning of step 4, the instruction decoder interprets the contents of the IR. This enables the control circuitry to activate the control-signals for steps 4 through 7. The step 4 through 7 constitutes the Execution Phase. Step 4: Contents of R3 are loaded into MAR & a memory read signal is issued. Step 5: Contents of R1 are transferred to Y to prepare for addition.	7	7	

	Step 6: When Read operation is completed, memory-operand is available in MDR, and the addition is performed. Step 7: Sum is stored in Z, then transferred to R1. The End signal causes a new instruction fetch cycle to begin by returning to step 1.			
III.10	 Hardwired control is a method of control unit design. The required control signals are determined by contents of step counter, instruction register, condition code flags, external input signals such as MFC and interrupt requests. Decoder/Encoder Block is a combinational-circuit that generates required control-outputs depending on state of all its inputs. Instruction Decoder: It decodes the instruction loaded in the IR. If IR is an 8 bit register, then instruction decoder generates 2^8(256 lines); one for each instruction. It consists of a separate output-lines INS_1 through INS_m for each machine instruction. According to code in the IR, one of the output-lines INS_1 through INS_m is set to 1, and all other lines are set to 0. Step-Decoder provides a separate signal line for each step in the control sequence. Encoder: It gets the input from instruction decoder, step decoder, external inputs and condition codes. It uses all these inputs to generate individual control-signals: Y_{in}, PC_{out}, Add, End and so on. For example, $Z_{in}=T_1+T_6.ADD+T_4.BR$; This signal is asserted during time-slot T1 for all instructions, during T6 for an Add instruction, during T4 for unconditional branch instruction. When RUN=1, counter is incremented by 1 at the end of every clock cycle. When RUN=0, counter stops counting. After execution of each instruction, end signal is generated. End signal resets step counter. Sequence of operations carried out by this machine is determined by wiring of logic circuits, hence the name "hardwired"	2 5	7	

III.11	 Memory Segmentation is the process in which the main memory of the computer is divided into different segments and each segment has its own address. The Bus Interface Unit (BIU) contains four 16-bit special purpose registers called as Segment Registers - Code segment register (CS), Data segment register (DS), Extra Segment Register (ES), Stack Segment Register (SS). The four segment registers contain the upper 16 bits of the starting addresses of the four memory segments of 64 KB each with which the 8086 is working at that instant of time.	5 2	7	
III.12	In multicore technology, two or more processing elements from in the same chip, physically very close and communication is very fast. Higher efficiency, low power consumption, effective multithreading, thread level parallelism, reliable, software running in different cores can communicate, geographical and temporal isolation may increase as the threads on one core are not delayed by threads running on another core. 	7	7	