Question Bank

23CST307 - MICROPROCESSORS AND MICROCONTROLLERS

Course Type	Course Nature	CA Conduct	System	L	T	P	Credits	CA Total	CA Pass	SEE Total	SEE Pass	Total Pass
Theory	1	End Semester	Mark	3	1	0	4	50	0	100	40	75

Question Bank Summary

Sect. Part A	Sect. Part B	Easy	Med.	Chall.	Th.	Appli.
73	60	26	82	25	101	32

Part A

#	Unit	Question	COS	Categorized
1	1.1	Compare the 8085 and 8086 microprocessors.	CO1	Easy - Analysing - T
2	1.1	List any three important features of the 8085 microprocessor and explain how each feature contributes to its operation.	CO1	Easy - Understanding - T
3	1.2	Differentiate between the functions of the following 8086 microprocessor signals and analyze their impact on system performance: ii) ALE ii) BHE iii) NMI	CO1	Medium - Analysing - T
4	1.2	The value of Code Segment (CS) Register is 3054H and the value of different registers is as follows: **BX: 4025H,IP: 1580H ,DI: 5467H**.Compute the physical address of the next instruction to be fetched using the segmented addressing scheme.	CO1	Medium - Applying - A
5	1.2	Differentiate between the architectural and operational features of 8086 and 8088 microprocessors based on data bus width, instruction queue, and execution performance	CO1	Medium - Understanding - T
6	1.2	Illustrate the concept of pipelined architecture in the 8086 microprocessor and explain how it enhances instruction execution efficiency	CO1	Medium - Understanding - T
7	1.2	Explain how the 8086 microprocessor accesses a 16-bit word from both even and odd memory addresses, and analyze the impact of alignment on memory access efficiency	CO1	Medium - Applying - T
8	1.2	Demonstrate how various flags in the 8086 microprocessor influence the execution of conditional instructions by identifying and explaining the purpose of each flag.	CO1	Medium - Applying - T
9	1.2	While developing an 8086-based arithmetic module, a student observes unexpected results during addition operations involving BCD values and binary numbers. As part of the debugging process, explain how the Carry flag, Auxiliary Carry flag, and Parity flag function in the 8086 microprocessor.	CO1	Medium - Applying - A
10	1.2	During the development of a string manipulation and interrupt-driven I/O program using the 8086 microprocessor, a developer encounters unexpected behavior in the execution flow. Explain the functions of the Overflow flag, Interrupt flag, and Direction flag, and analyze how the status of each flag can affect arithmetic operations, interrupt handling, and string processing in the program.	CO1	Medium - Analysing - A
11	1.2	Explain how the **READY, HOLD**, and **INTR** signals of the 8086 microprocessor contribute to its interfacing and control mechanisms.	CO1	Medium - Applying - T

12	1.2	Explain the role of CLK, LOCK, and RESET signals in the 8086 microprocessor and illustrate how they support reliable system operation and peripheral interfacing	CO1	Medium - Applying - T
13	1.2	Draw the 8086 flag register structure and explain how control flags are used to modify the behavior of instructions during program execution.	CO1	Medium - Applying - T
14	2.2	Illustrate the usage of the following rotate instructions with suitable examples in 8086 assembly language: (i) RCR (ii) RCL Explain how these instructions affect the Carry Flag and the operand bits.	CO2	Medium - Applying - T
15	2.2	Differentiate between the **LDS** and **LES** instructions in 8086 microprocessor. Analyze how each instruction loads data and segment registers from memory.	CO2	Medium - Analysing - T
16	2.2	Compare the use of **LAHF**/**SAHF** with the **PUSHF**/**POPF** instructions for saving and restoring flags. Analyze the scenarios where one is preferred over the other.	CO2	Challenging - Evaluating - A
17	2.2	Illustrate the use of the **IN** instruction in the 8086 microprocessor with suitable examples. Demonstrate how it is used for both direct and indirect port addressing modes.	CO2	Medium - Analysing - T
18	2.2	Demonstrate how the **OUT** instruction is used in the 8086 microprocessor to transfer data to an output port. Provide one example each for 8-bit and 16-bit data transfers.	CO2	Medium - Applying - T
19	2.2	Analyze the operation of the **CMP** instruction in 8086. Explain why it does not store the result of the comparison and discuss how it influences program control flow through flag status.	CO2	Easy - Analysing - T
20	2.2	The 8086 microprocessor supports various string operations like MOVSB, CMPSB, and LODSB which operate on sequences of data. In this context, what is the purpose of the **REP prefix** when used with such string instructions? Illustrate with an example.	CO2	Medium - Analysing - T
21	2.2	The 8086 microprocessor uses the **Direction Flag (DF)** to control the movement of source (SI) and destination (DI) pointers during string operations like MOVSB. Discuss the function of the Direction Flag in such operations, and explain how its setting (set or cleared) influences the direction of data transfer when executing repeated string instructions.	CO2	Easy - Understanding - T
22	2.2	While designing a system that requires frequent data movement between registers, memory, stack, and I/O devices, an embedded systems developer must understand and utilize the appropriate data transfer instructions supported by the 8086 microprocessor. Classify and list these data transfer instructions based on the source and destination of data (registers, memory, stack, and I/O ports). Explain the significance of each category with suitable examples that illustrate how they enable efficient data exchange in real-time processing tasks.	CO2	Medium - Analysing - T
23	2.2	Analyze the differences between **JUMP** and **LOOP** instructions in 8086 microprocessor in terms of execution control and register dependency, and explain how understanding these distinctions supports efficient program flow design in embedded systems.	CO2	Challenging - Understanding - T
24	2.2	Demonstrate the operation of the **PUSH** and **POP** instructions in the 8086 microprocessor. Use a suitable example to show how these instructions affect the stack, including changes in the Stack Pointer (SP) and memory contents. Illustrate the process with diagrams.	CO2	Medium - Applying - T
25	2.2	Explain the functionality and typical use-cases of the 8086 instructions **NOP, XCHG,** and **STC**. How does each instruction contribute to effective program execution?	CO2	Medium - Understanding - T
26	2.2	A student attempts to initialize the data segment in an 8086 assembly program using the instruction **MOV DS, 6000 H**, but the assembler throws an error. As a system-level programmer, analyze why this instruction is invalid. Explain the correct procedure to load an immediate value into the DS register, and justify each step based on 8086 architecture constraints.	CO2	Challenging - Analysing - A
27	2.2	Imagine a scenario where you need to convert encoded input values into corresponding characters using a predefined lookup table. Explain how the **XLAT** instruction in the 8086 microprocessor facilitates this translation. Analyze its role in simplifying table-driven conversions and assess its usefulness in such applications.	CO2	Challenging - Evaluating - A
28	2.3	Demonstrate an understanding of the assembler directives **ENDP** and **ASSUME** used in 8086 assembly language by explaining their purpose and role in program organization.	CO2	Easy - Applying - T
29	2.3	Discuss the importance of **assembler directives** in 8086 assembly language programming. Illustrate your answer with two relevant examples.	CO2	Medium - Understanding - T

30	2.3	Discuss the significance of assembler directives in 8086 assembly language programming. Explain the functions of **SEGMENT, OFFSET,** and **PROC** directives with appropriate usage.	CO2	Medium - Understanding - T
31	2.3	Explain the purpose of the following assembler directives in 8086 programming and explain their role in program development: (i) EQU (ii) ENDP	CO2	Medium - Understanding - T
32	3.1	What happens to the **Stack Pointer (SP)** when a PUSH AX instruction is executed? Explain briefly.	CO3	Medium - Applying - A
33	3.1	Differentiate between **PUSH and POP** instructions with respect to the 8086 stack.	CO3	Medium - Understanding - T
34	3.1	If SP = 2000H and a PUSH BX instruction is executed, what will be the new value of SP?	CO3	Medium - Applying - A
35	3.1	Explain the role of the stack in managing subroutine calls and returns.	CO3	Medium - Understanding - T
36	3.1	Explain the concept of a “top-down” data structure in the context of a stack.	CO3	Medium - Understanding - T
37	3.1	Identify the situation that leads to a stack overflow condition in 8086.	CO3	Medium - Remembering - T
38	3.2	Write notes on the following based on 8086: a. software interrupt b. hardware interrupt c. nested interrupt	CO3	Medium - Understanding - T
39	3.2	Differentiate between **maskable** and **non-maskable interrupts** in 8086.	CO3	Easy - Understanding - T
40	3.2	Define an **interrupt** in 8086. What is the need of push and pop?	CO3	Easy - Understanding - T
41	3.2	What is an **Interrupt Service Routine (ISR)**? State its purpose	CO3	Easy - Understanding - T
42	3.2	Write a short note on **software interrupts** in 8086.	CO3	Medium - Understanding - T
43	3.3	What is the need for a **Programmable Interrupt Controller (8259)** in a microprocessor system	CO3	Easy - Understanding - T
44	3.3	Define the role of the **Interrupt Request Register (IRR)** and **In-Service Register (ISR)** in 8259.	CO3	Easy - Understanding - T
45	3.3	Describe the significance of the 8259 Programmable Interrupt Controller (PIC) in the context of advanced computer systems design and operation.	CO3	Medium - Understanding - T
46	4.1	Analyze and interpret the mode selection and port configurations of 8255 when its control word register is loaded with **86H**.	CO4	Challenging - Analysing - A
47	4.1	Determine the port configurations (Port A, Port B, Port C) when the control register is loaded with **10111XXX**.	CO4	Challenging - Applying - A
48	4.1	Explain the primary function of the 8255 Programmable Peripheral Interface (PPI).	CO4	Easy - Understanding - T

49	4.1	Differentiate between Group A and Group B control in 8255 PPI, and analyze how their configuration affects port operations in microprocessor-based systems.	CO4	Medium - Understanding - T
50	4.1	Assess the importance of the **STB#, IBF**, and **INTR** signals in **Mode 1** input operation of 8255, and explain how they ensure reliable handshaking and synchronization between the CPU and peripheral devices.	CO4	Medium - Evaluating - T
51	4.1	Explain the role of **OBF#** and **ACK#** signals in **Mode 1** output operation of 8255, and analyze how they facilitate reliable handshaking and data transfer between the CPU and peripheral devices.	CO4	Medium - Analysing - T
52	4.1	Interpret the mode and configurations of 8255 after its control word register is loaded with **83H**.	CO4	Challenging - Understanding - A
53	4.2	Explain why a programmable interval timer (PIT) is preferred over software delay routines in microprocessor-based systems.	CO4	Medium - Understanding - T
54	4.2	List the features of 8254 Programmable Interval Timer.	CO4	Easy - Remembering - T
55	4.3	List the major features of the 8257 DMA controller.	CO4	Easy - Remembering - T
56	4.3	Explain the purpose of the **DRQ, TC,** and **MARK** signals of the **8257 DMA controller**, and analyze their role in coordinating data transfer between CPU, memory, and I/O devices.	CO4	Medium - Understanding - T
57	4.3	Explain the role of the **MARK** signal in the 8257 DMA controller, and analyze its significance in synchronizing data transfer during DMA operations.	CO4	Medium - Analysing - T
58	4.3	Compare the role of **IOR** and **IOW** signals in slave mode versus master mode of 8257.	CO4	Challenging - Evaluating - T
59	4.3	Analyze the impact of the **READY** signal on DMA cycles and CPU-peripheral synchronization.	CO4	Easy - Analysing - T
60	4.3	Justify the statement: “DMA controlled data transfer faster than CPU controlled data transfer”.	CO4	Medium - Evaluating - T
61	5.1	Describe internal memory organization of 8051 microcontroller	CO5	Easy - Understanding - T
62	5.1	List any three key differences between a microprocessor and a microcontroller	CO5	Easy - Understanding - T
63	5.1	List important characteristics of the 8051 microcontroller.	CO5	Easy - Understanding - T
64	5.1	Compare the roles and functionalities of DPTR and PC in the 8051.	CO5	Medium - Understanding - T
65	5.1	Analyze the roles of different I/O ports in the 8051 microcontroller.	CO5	Easy - Analysing - T
66	5.2	During arithmetic computation, you need to move data from register to the **accumulator** for addition. • Which addressing method is used here? • Write the instruction and explain how the operand is fetched.	CO5	Medium - Applying - A

67	5.2	A variable is stored in **internal RAM location 30H**, and you want to copy it to the accumulator. • Identify the suitable addressing mode. • Write the corresponding instruction and explain how 8051 locates the data.	CO5	Medium - Applying - A
68	5.2	You are creating an array starting from address **40H** in internal RAM, and you want to access its elements sequentially using **R0**. • Which addressing approach is appropriate? • Demonstrate how MOV A, @R0 would work if R0 = 43H.	CO5	Medium - Analysing - A
69	5.3	Explain any **three arithmetic instructions** of the **8051 microcontroller** with their **operation, syntax, and example**.	CO5	Easy - Understanding - T
70	5.3	Explain any **three logical instructions** of the **8051 microcontroller** with their **operation, syntax, and example**.	CO5	Medium - Understanding - T
71	5.4	List **three features** of ARM processors that make them suitable for embedded systems.	CO5	Easy - Understanding - T
72	5.4	Explain the reasons why **ARM Architecture is considered valuable** in modern embedded and computing systems.	CO5	Medium - Understanding - T
73	5.4	Mention the advantage of having a **large number of registers** in ARM architecture.	CO5	Medium - Understanding - T

Part B

#	Unit	Question	COS	Categorized
1	1.1,2	A. Illustrate the architecture of the 8085 microprocessor with the help of a labeled diagram and explain the function of each block to demonstrate your understanding of its internal organization (7 marks) B. Analyze the physical memory organization of the 8086 microprocessor using a labeled diagram, and explain how segmentation supports memory addressing. (7 marks)	CO1,CO1	Medium - - T
2	1.2	Illustrate and explain the internal architecture of the 8086 microprocessor, highlighting the functions of its major components and their role in instruction execution	CO1	Medium - Applying - T
3	1.2	A. Draw the register organization of the 8086 microprocessor and explain the function of each register, demonstrating how they contribute to instruction execution and data handling. (10 marks) B. Apply the concept of segmentation in the 8086 microprocessor to compute the 20-bit physical address from given segment and offset values, using appropriate examples. (4 marks)	CO1,CO1	Medium - - T
4	1.2	A. Demonstrate your understanding of the 8086 microprocessor's physical memory organization by illustrating it with a suitable diagram and explaining how segmentation enables access to a 1MB memory space. (6 marks) B. Analyze the read and write timing diagrams of the 8086 microprocessor in minimum mode using labeled diagrams, and explain the function of each control signal during memory operations (8 marks)	CO1,CO1	Medium - - T

5	1.2	A. Explain the maximum mode configuration of the 8086 microprocessor, highlighting the role of control signals to demonstrate your understanding of system-level operation. (9 marks) B. Using a labeled block diagram, analyze the architecture of the 8085 microprocessor and explain how each component contributes to its overall functionality (5 marks)	CO1,CO1	Medium - - T
6	1.2	A. Illustrate the memory read and write timing diagrams of the 8086 microprocessor in maximum mode, and explain the role of control and status signals involved in each operation (9 marks) B. Draw the format of the 8086 flag register and describe the function of each flag, highlighting their role in decision-making and control flow during program execution (5 marks)	CO1,CO1	Medium - - T
7	1.2	A. Using a labeled internal block diagram, analyze the architecture of the 8086 microprocessor and explain how its functional units interact during instruction execution. (8 marks) B. Illustrate and analyze the read and write timing diagrams of the 8086 microprocessor in minimum mode, explaining the role of each control signal during memory access (6 marks)	CO1,CO1	Medium - - T
8	1.2	Illustrate the pin diagram of the 8086 microprocessor and explain the function of each signal to demonstrate your understanding of its hardware interface and control mechanisms.	CO1	Medium - Applying - T
9	1.2	A. Using a labeled diagram, analyze the architecture of the 8086 microprocessor and explain how its functional units interact to perform instruction execution (10 marks) B. Draw the format of the 8085 flag register and describe the function of each flag, highlighting their significance in controlling and monitoring processor operations (4 marks)	CO1,CO1	Medium - - T
10	1.2	A. Using a labeled diagram, analyze the physical memory organization of the 8086 microprocessor and explain how segment and offset addressing enable access to a 20-bit address space. (10 marks) B. Explain how the 8086 accesses words at even and odd addresses, highlighting the impact of data alignment and bus usage on memory operations (4 marks)	CO1,CO1	Medium - - T
11	1.2	A. Analyze the minimum mode configuration of the 8086 microprocessor and explain how control signals are managed to support complex system operations. (9 marks) B. Compare the architectures and key features of the 8086 and 8088 microprocessors to demonstrate your understanding of their structural and functional differences (5 marks)	CO1,CO1	Medium - Analysing - T
12	2.1,4	A. A simple embedded system is designed to **calculate the factorial of a number** input by the user. The number is stored at memory location **4000H**. As the system is built using 8086 architecture, you are asked to write an assembly program to compute the factorial and store the result at memory location **5000H** so that it can be accessed by other modules. (9 marks) B. You are working as a system-level programmer for an embedded application using the 8086 microprocessor. While analyzing the program's memory usage, you are asked to compute the physical memory addresses accessed by the following two instructions. The system provides the following register values: **DS = 2AD3 H, CS = 0058 H, BX = 0312 H, and SI = 0058 H**. Instructions: 1. **MOV AL, [BX]** 2. **MOV [BX] [SI] 15H, DL** Determine the segment used by each instruction, calculate the physical addresses, and explain how the addressing mode influences the final result. (5 marks)	CO2,CO2	Challenging - - A

13	2.1	A. A programmer is developing an 8086 assembly routine and needs to access operands using different techniques to optimize performance and memory usage. Explain the various addressing modes available in the 8086 microprocessor that the programmer can utilize. For each mode, describe its functionality, syntax, and provide relevant examples to demonstrate how it helps in efficient instruction execution. (10 marks) B. During the execution of a data transfer routine in an 8086-based embedded system, the segment and offset registers are set with the following values: **DS = 9D8F H, CS = 500C H, BX = 03B2 H, and SI = 1004 H.** Two instructions are executed— 1. **MOV [BX], AL** 2. **MOV AL, [BX] [SI] A3 H.** The system engineer needs to verify the exact physical memory locations being accessed for debugging. Evaluate the physical addresses referred by each instruction, showing how the effective address is formed using the register values and how it is then combined with the segment base to compute the final physical address in memory. (4 marks)	CO2,CO2	Medium - - T
14	2.2,4	A. A system requires user input to be processed in reverse order for string comparison in a legacy embedded device using the 8086 microprocessor. As a system programmer, develop an 8086 assembly language program that accepts a string, stores it in memory, and displays its reverse. Explain how your code manages string traversal and memory access. (7 marks) B. As an assembly language programmer, explain how you would use **branching instructions** in 8086 to manage the flow of control in this application. Clearly differentiate between **conditional and unconditional branching** instructions, and provide suitable examples of each from your implementation. (7 marks)	CO2,CO2	Challenging - - A
15	2.2	As an embedded programmer, discuss the various **data transfer instructions** available in 8086. Support your explanation with relevant examples showing how each instruction facilitates efficient data movement.	CO2	Easy - Understanding - T
16	2.2,4	A. Assume that 8086 registers having values AX=0030H, BX =0031 H, CX=0032 H, DX=0033 H, Flag - 0000 H. Predict the values of Registers and Flags [AX, BX, CX, DX, Carry flag (CF), Zero Flag (ZF), Sign Flag (SF)] after the execution of following instructions: (Assume each instruction are being executed independently) i) ROR AX,04 h ii) CMP BX, CX iii) XCHG CX, DX iv) AND AX, BX v) LOOP Addr vi) XOR AX, AX vii) STC Hint - Draw a table with columns Instructions, AX, BX, CX, DX, CF, ZF, SF and fill the answers. (7 marks) B. In a sensor monitoring system, the readings collected are stored in a consecutive memory block starting from address **2001 H**. The total number of readings (n) is stored at address **2000 H**. As a system programmer, develop an 8086 assembly language program to find the **largest number among 'n' one-byte values**. Clearly explain how the program accesses memory, compares values, and stores the final result. (7 marks)	CO2,CO2	Challenging - - A
17	2.2,4	A. A data-acquisition unit stores **15 unordered 16-bit sensor readings** in consecutive memory starting at address **2500 H**. As the embedded-systems programmer, write an 8086 assembly program that finds both the **largest and the smallest value** in this array. The results must remain in registers for later use. Also draw a flow-chart that clearly shows the program logic. (10 marks) B. Explain the functional and operational differences between **TEST** and **AND** instructions. Provide suitable examples for each and analyze their impact on the processor flags and operand values. (4 marks)	CO2,CO2	Challenging - - A
18	2.2	Analyze the various **arithmetic instructions** supported by the 8086 microprocessor and explain how each instruction affects the condition code flags. Support your explanation with relevant examples.	CO2	Easy - Analysing - T

19	2.2	A. Explain the functionality of the **SUB, SBB,** and **CMP** instructions in the 8086 microprocessor. How does each instruction operate, and in what ways do they differ from one another? Support your explanation with suitable examples. (7 marks) B. During the development of a billing module for a legacy accounting system, you're required to perform decimal arithmetic operations using Binary Coded Decimal (BCD) values. The addition and subtraction operations store intermediate results in the AL register, which may not be valid BCD numbers. As a system-level programmer, analyze how the **DAA (Decimal Adjust after Addition)** and **DAS (Decimal Adjust after Subtraction)** instructions adjust the result in AL to obtain valid BCD output. (7 marks)	CO2,CO2	Medium - - T
20	2.2,4	A. Explain the purpose and functioning of the ASCII adjust instructions in the 8086 microprocessor — **AAA, AAS, AAM**, and **AAD**. Describe how each instruction modifies the result of arithmetic operations involving unpacked BCD (Binary Coded Decimal) values. Support your explanation with suitable examples showing the adjustments made by each instruction. (9 marks) B. You are working with an embedded monitoring system that captures 10 sensor readings, each stored as an **8-bit value** in consecutive memory locations. Write an 8086 assembly language program to **calculate the total sum of these 10 readings** and store the result in the AX register. Assume the numbers are stored starting at a known memory address. (5 marks)	CO2,CO2	Challenging - - A
21	2.2	A. You are developing a simple digital lock system where each key press updates a register value. To encode and decode the values, you need to move and rotate bits within the register. For this, rotate and shift instructions are used. As part of your task, explain the **rotate** and **shift instructions in 8086** assembly language. Use simple examples to show how these instructions work and how they affect the bits in a register. (10 marks) B. You are creating a basic step counter program in 8086 assembly that increases or decreases a value each time a user presses a button. For this, you plan to use **INC** and **DEC** instructions. Your supervisor asks you to explain these instructions clearly before you proceed. Write short notes on the INC and DEC instructions in 8086. Explain their operation, syntax, addressing modes, and effect on flags. Also, justify why these instructions do not affect the Carry Flag during execution. (4 marks)	CO2,CO2	Medium - - T
22	2.3,4	A. You are developing a password verification feature for a simple 8086-based embedded device. The system needs to compare a user-entered **5-character password** with a stored password. Both passwords are stored as **byte strings** in memory, with the source string pointed to by DS:SI and the destination string pointed to by ES:DI. If the two strings are equal, the system should store 00H in the AL register to indicate a match; otherwise, it should store FFH. Write an 8086 assembly language program that compares the two strings using the CMPSB instruction along with the REPE prefix. Ensure that the program correctly sets the AL register based on the comparison result. (10 marks) B. Illustrate how the assembler directives **EVEN** affect data alignment and how **SEG** helps in segment-relative addressing, using suitable code snippets. (4 marks)	CO2,CO2	Challenging - - A
23	2.4	You are tasked with analyzing a set of **20 sensor readings**, each **16-bit wide**, stored in memory **starting at address 5500H**. These readings are used by a control system to determine the stability of the environment. The system must identify how many readings are even (indicating stable conditions) and how many are odd (indicating fluctuations), in order to trigger corresponding control responses. As an embedded system programmer, develop an 8086 assembly language program to **count the total number of even and odd numbers** from the given array. Additionally, draw a flowchart to represent the logic of your implementation.	CO2	Challenging - Applying - A
24	3.1	A. With neat diagram, explain the **stack structure and addressing mechanism** in 8086 using SS and SP registers. (10 marks) B. Analyse the function of the RET instruction in relation to the stack?(4 marks)	CO3,CO3	Medium - - T
25	3.1,2	A. Draw and explain the stack structure of 8086.(7 marks) B. Describe the purpose of the Interrupt Vector Table of 8086 and explain its structure.(7 marks)	CO3,CO3	Medium - Understanding - T

26	3.2	Describe the interrupt handling process in 8086 microprocessors	CO3	Easy - Understanding - T
27	3.2	What is the **Interrupt Vector Table (IVT)**? Explain its structure and how the CPU uses it to locate ISRs.	CO3	Easy - Understanding - T
28	3.3	With a neat diagram, explain the **architecture of 8259 PIC**. Highlight the function of each major block.	CO3	Medium - Understanding - T
29	3.3	A. Discuss the roles of **IRR, ISR, and IMR registers** in handling interrupts. (7 marks) B. Explain how the priority resolver, control logic, and cascade buffer of the 8259 Programmable Interrupt Controller contribute to efficient interrupt management. (7 marks)	CO3,CO3	Medium - - T
30	3.3	Describe the operation of the 8259 Programmable Interrupt Controller (PIC) and how it manages multiple interrupt requests efficiently.	CO3	Medium - Applying - T
31	3.4	Interface **two 4K X 8 EPROMS** and **two 4K X 8 RAM** chips with 8086. Select suitable maps.	CO3	Challenging - Applying - A
32	3.4	Interface two 32K X 8 EPROMS and two 32K X 8 RAM chips with 8086, microprocessor and draw the suitable circuit showing their interfacing.	CO3	Medium - Applying - A
33	3.4	Design an interface b/w 8086 CPU and **two** chips of **16K X 8 EPROM** and **two** chips of **32K X 8 RAM**. Select the starting address of EPROM suitably. The **RAM** address must start at **00000H**.	CO3	Challenging - Analysing - A
34	3.4	Design an interface between CPU 8086 and **two chips of 32K X 8 ROM** and **four chips of 32 K X 8 RAM** according to the following memory map. **ROM1 and ROM2 ⇒ F0000H - FFFFFH RAM1 and RAM2 ⇒ D0000H - DFFFFH RAM3 and RAM4 ⇒ E0000H - EFFFFH**	CO3	Challenging - Creating - A
35	4.1	Explain the different modes of operation of **8255 PPI**, and analyze how each mode facilitates input/output operations in a microprocessor-based system.	CO4	Challenging - Analysing - T
36	4.1	Describe the architecture of **8255 PPI** with a neat diagram, and analyze how its ports and control logic facilitate communication between a microprocessor and peripheral devices.	CO4	Easy - Understanding - T
37	4.1	A. Assess the significance of the Control Word Register (CWR) in 8255 PPI and analyze its role in determining port configurations and operating modes. (7 marks) B. Suppose 8 LEDs connected to 8255 Port-C Pins (PC 0 – PC 7), what will be the sequence of control words to be pushed to the 8255 for turning-on the LEDs connected at Odd Pins (PC0, PC2, PC4, PC6). Hint: Assume a High in Port C Pin will turn-on the LED. (7 marks)	CO4,CO4	Challenging - - T
38	4.1	A. Assess the importance of handshake signals in **Mode 1** of 8255 PPI for ensuring synchronized and reliable data transfer between the CPU and peripheral devices. (9 marks) B. Assess the significance of **BSR mode** in practical interfacing applications. (5 marks)	CO4,CO4	Medium - - T
39	4.1,2	A. Illustrate the functions of the control signals **RD, WR, A0, A1, RESET,** and **CS** in **8255 PPI** and assess their role in controlling data transfer and port operations. (7 marks) B. Illustrate how the 8254 programmable timer can be used as a square wave generator. (7 marks)	CO4,CO4	Challenging - - T
40	4.1	A. Identify the salient features of Mode 0 (Basic I/O) and Mode 2 (Strobed Bidirectional I/O) of 8255 PPI, and analyze how these modes facilitate efficient data transfer between the CPU and peripherals. (10 marks) B. Interpret the mode and configurations of 8255 after its control word register is loaded with **82H**. (4 marks)	CO4,CO4	Medium - - T

41	4.1	A. Compare the three modes of operation of 8255 PPI: Mode 0, Mode 1, and Mode 2, focusing on their mechanisms for data transfer and handshaking, and assess their effectiveness in different I/O scenarios. (7 marks) B. Identify the registers available in the 8257 DMA Controller and discuss their functions in managing memory-to-I/O and I/O-to-memory data transfers. (7 marks)	CO4,CO4	Medium - - T
42	4.2	Explain the architecture and operation modes of the **8254** programmable timer with a neat block diagram.	CO4	Medium - Understanding - T
43	4.2,3	A. Explain the overall working of the 8257 DMA controller, illustrating its role in transferring data between memory and I/O devices with neat diagrams, and analyze how it reduces CPU involvement in data transfer operations. (9 marks) B. Draw the operational waveform of the 8254 programmable timer in **Mode 0** and explain how it generates time delays, illustrating the behavior of the output signal during the counting process. (5 marks)	CO3,CO4	Medium - - T
44	4.2	Illustrate the internal architecture of 8254 and analyze how each counter operates in different modes.	CO4	Easy - Analysing - T
45	4.3	Draw the internal architecture of the **8257 DMA controller** and explain how its components coordinate to perform memory-to-I/O and I/O-to-memory data transfers efficiently.	CO4	Medium - Understanding - T
46	4.3	A. Examine how the following signals—MEMR, MEMW, HRQ, HLDA, Address Strobe (ADSTB), and Address Enable (AEN)—interact to ensure correct and efficient DMA data transfer. (7 marks) B. Identify the roles of the DMA Address Register and the Terminal Count Register (TCR) in DMA operations, and assess their importance in ensuring accurate and efficient data transfer. (7 marks)	CO4,CO4	Medium - - T
47	5.1	A. Explain the **memory organization** of the 8051 microcontroller with a neat diagram. Discuss the division of internal RAM, ROM, and the address range of each memory segment. (9 marks) B. List any five special function registers (SFRs) of the 8051 microcontroller and explain their purposes (5 marks)	CO5,CO5	Medium - Understanding - T
48	5.1	Explain the architecture of the 8051 microcontroller.	CO5	Medium - Remembering - T
49	5.1,3	A. Develop an 8051 assembly language program to perform matrix addition. (9 marks) B. Illustrate how external memory can be interfaced with the 8051 microcontroller. (5 marks)	CO5,CO5	Challenging - - A
50	5.1,3	A. Write an assembly language program for 8051 to find the transpose of a 2x2 matrix. (10 marks) B. Illustrate the importance of RAM and ROM memory in the 8051 microcontroller. (4 marks)	CO5,CO5	Challenging - - A
51	5.1,3	A. Write an 8051 assembly language program to find the largest number in an array. (10 marks) B. Illustrate the significance of PSEN and EA control signals in the 8051. (4 marks)	CO5,CO5	Medium - - A
52	5.1,3	A. Explain in detail the **register organization of the 8051 microcontroller** with a neat diagram. Describe the purpose and function of each register. (7 marks) B. Develop an 8051 program to determine the count of ones and zeros in an 8-bit data value. (7 marks)	CO5,CO5	Medium - - A
53	5.1	A. Explain the concept of a **stack** in the 8051 microcontroller. Describe how **stack operations (PUSH and POP)** work with the **Stack Pointer (SP)**. Illustrate the process with a neat diagram. (9 marks) B. Discuss the organization of register banks in the 8051 microcontroller and show a clear diagram. (5 marks)	CO5,CO5	Medium - Understanding - T

54	5.1,3	A. Write an assembly language program for 8051 to add ten bytes in internal RAM. Assume that the starting location of the block is 40H. Assume the sum to be 16 bits. Store the result in registers R2 and R3 of register bank 0, with the LSB stored in register R2 and the MSB in register R3. (7 marks) B. Draw and explain the format of program status word in 8051.(7 marks)	CO5,CO5	Medium - - A
55	5.1,3	A. Write an assembly language program for 8051 to find whether a given byte is available in the given sequence or not. If it is available, write FF in R3. Otherwise write 00 in R3 (10 marks) B. Discuss the functioning and organization of the stack in the 8051 microcontroller.(4 marks)	CO5,CO5	Challenging - - A
56	5.2	A. Analyze the different addressing modes used in the 8051 microcontroller and give examples for each. (10 marks) B. Write an assembly language program for 8051 to divide two 8 – bit numbers stored in external m/y locations 3000H and 3001 H. Store the quotient at 3020H and remainder at 3021H. (4 marks)	CO5,CO5	Medium - - T
57	5.3	**Explain in detail** the various categories of **8051 instructions — data transfer, arithmetic, logical, Boolean (bit manipulation), and control transfer instructions —** with suitable syntax, examples,	CO5	Medium - Understanding - T
58	5.3	A. Develop an 8051 assembly language program to count the positive numbers in a given array. (10 marks) B. Differentiate MOV, MOVC, MOVX instructions of 8051 microcontroller. (4 marks)	CO5,CO5	Challenging - - A
59	5.3	A. Illustrate and explain the data transfer instructions available in the 8051 microcontroller. (10 marks) B. Discuss the various 8051 instructions that operate on data bytes. (4 marks)	CO5,CO5	Medium - Understanding - T
60	5.4	Draw and explain the architecture of the ARM microprocessor in detail.	CO5	Medium - Understanding - T