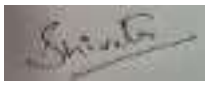
Department of Computer Science and Engineering

Details of Lesson Plan

S.No.	Particulars	Details
1.	Course Name	Microprocessors
2.	Course Code	COM-503
3.	Academic Year	2024-2025
4.	Semester	5th
5.	Number of Lesson plans	42
6.	Faculty Assigned	Ms. Shiveta Bhat



Faculty Signature

Lesson Plan No. 1	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Articulate the basic concept of microprocessor b. Explain the Evolution of Microprocessor c. Illustrate the various applications of microprocessor d. appreciate advantages of microprocessor based real life applications
Teaching Aids (if any)	a. Video on applications of microprocessor b. ICT usage
Teaching Development	1. **Introduction** (5 minutes) - Ask questions Have you heard about microprocessor? What do you mean by an integrated circuit? What do you mean by a processor? Which processor does your laptop have? - Introduce the basics of microprocessor. Show 8085. - Discuss about the Evolution of microprocessors - Talk about the processors available in the market these days. t - Introduce the formal definition of Microprocessor - Highlight the importance of interfacing 2. **Development** (30 minutes) a. Microprocessor - Introduce the concept of Microprocessor - Show video on evolution of microprocessors https://www.youtube.com/watch?v=QtYzvODwdPI - Introduce concept of interfacing b. Interfacing - Introduce the concepts of Interfacing - show the various devices that can be interfaced with a microprocessor - Give example about the projects that can be made using microprocessors. - Discuss the applications of microprocessors. https://www.youtube.com/playlist?list=PLm_MSClsnwm8Dw5nmh8A5E7gO06j1TeIa Major players in microprocessors - Intel Advantages of microprocessors - Speed - Compatibility - Low size - Versatile - Give examples to illustrate the advantages from a user-perspective. 3. Exercise (5 minutes) – - Give students time to discuss the various applications of microprocessors.

Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested material https://nptel.ac.in/courses/108/107/108107029/ 3. Homework Make a chart explaining the evolution of microprocessors discussing the advancement in features over the time. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Devise the architecture of 8085 b. Explain the various blocks in the architecture of 8085
Teaching Aids (if any)	a. Video on architecture of microprocessor b. ICT Usage
Teaching Development	1. **Introduction** (5 minutes) - Ask questions. - Have you ever studied the architecture of a microprocessor before? - Why is it important to understand the architecture of a microprocessor? - Explain that understanding the architecture of the 8085 microprocessor is essential for comprehending how it processes information and controls various operations. - Show a simplified block diagram of the 8085 microprocessors 2. **Development** (30 minutes) **a. Devise the Architecture of the 8085 Microprocessor:** **- Overview of the 8085 Architecture:** Display a detailed block diagram of the 8085 architectures. **- Components of the Architecture:** - Arithmetic and Logic Unit (ALU) - Registers - Interrupt Control - Serial I/O Control - Timing and Control Unit - Data and Address Buses **b. Explain the Various Blocks in the Architecture of the 8085:** **- Detailed Explanation of Each Block:** **-Arithmetic and Logic Unit (ALU):** - Explain its role in performing all arithmetic (addition, subtraction) and logical (AND, OR, NOT) operations. **- Registers:** - Discuss the purpose of each type of register.cal operations. - **Interrupts:** - Discuss different types of interrupts (RST 5.5, RST 6.5, RST 7.5, TRAP, INTR). - **Serial I/O Control**

	- **Timing and Control Unit** - **Data and Address Buses:** Explain the function of data and address buses in transferring data and addresses within the system. - Show a detailed, labeled diagram of the 8085 architectures, highlighting each block as it is discussed. 3. Exercise (5 minutes) – - Divide students into small groups and assign each group a specific block of the 8085 architectures to analyze. - Facilitate a discussion to compare and contrast the functions of different blocks.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading https://archive.nptel.ac.in/content/storage2/courses/106108100/pdf/Lecture_Notes/LNm1.pdf 3. Homework Assign a short-written task for students to explain the function and importance of a selected block and provide an example of its use. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Open the floor for any remaining questions Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 3	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the pin diagram of microprocessor b. Analyse the various pins of 8085
Teaching Aids (if any)	a. PPT on pin diagram of microprocessor b. ICT Usage
Teaching Development	1. **Introduction** (5 minutes) - Ask questions Have you seen 8085? What do you mean by pin diagram? - Introduce the pin diagram of microprocessor. Show 8085. - Discuss about the 40 pins of 8085 2. **Development** (30 minutes) a. Pin diagram of 8085 - Introduce the pin diagram of Microprocessor - Show slide on pin diagram of microprocessors 3. Exercise (5 minutes) – - Give students time to discuss the various pins of 8085. 4. **Development** (30 minutes) b. Pin diagram of 8085 - Introduce the pin diagram of Microprocessor - Show slide on pin diagram of microprocessors 5. Exercise (5 minutes) – - Give students time to discuss the various pins of 8085. 6. Exercise (5 minutes) – **Group Activity:** - Divide students into small groups and assign each group a set of pins to analyze. - Allow each group to present their findings to the class. - Facilitate a discussion to compare and contrast the functions of different pins.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading https://archive.nptel.ac.in/content/storage2/courses/106108100/pdf/Lecture_Notes/LNm1.pdf 3. Homework - Assign a short-written task for students to explain the function of a

	selected set of pins and provide examples of their application. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Review homework assignments to ensure comprehension of pin functions and their applications. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 4	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Explain the instruction set of 8085 b. Understand the various instructions available in 8085
Teaching Aids (if any)	a. ICT Usage b. PPT on the topic
Teaching Development	- **Introduction** (5 minutes) - Ask questions. - Have you ever worked with or heard about microprocessor instructions before? - Why do you think understanding the instruction set is important in programming a microprocessor? - Encourage students to think about the relevance and application of instruction sets - Explain that the instruction set is a collection of commands that the microprocessor can execute. - Explore the instruction set of the 8085 microprocessor and understand the various instructions it offers." - **Development** (30 minutes) **a. Explain the Instruction Set of the 8085 Microprocessor:** - **Overview of the Instruction Set:** - Display a slide with an overview of these categories. - **Categories of Instructions:** - Data Transfer Instructions - Arithmetic Instructions - Logical Instructions - Branching Instructions - Control Instructions **b. Understand the Various Instructions Available in the 8085 Microprocessor:** **- Detailed Explanation of Key Instructions:** - Data Transfer Instructions - Arithmetic Instructions - Logical Instructions - Branching Instructions - Control Instructions - **Examples and Demonstrations:** - Show a few simple programs using these instructions. - Explain how each instruction works in the context of the program.

	Display slides showing the syntax and usage of key instructions. - Exercise (5 minutes) – **Group Activity:** - Divide students into small groups and assign each group a set of instructions to analyze.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading - https://archive.nptel.ac.in/content/storage2/courses/106108100/pdf/Teacher_Slides/mod2/M2L1.pdf 3. Homework - Assign a short written task for students to create a few simple programs using different instructions and explain how they work. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. - What is the purpose of this instruction? - How does this instruction work in a program? - Can you write a simple program using this instruction? Facilitate a discussion to compare and contrast the different instructions. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 5	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Categorize the addressing modes of 8085 b. Explain the addressing modes of 8085
Teaching Aids (if any)	a. ICT Usage
Teaching Development	1. **Introduction** (5 minutes) - Ask questions. - Have you heard the term "addressing mode" before? - Why do you think addressing modes are important in microprocessor programming? 2. **Development** (30 minutes) - Explain that addressing modes define how the microprocessor accesses data from memory or registers. **a. Categorize the Addressing Modes of the 8085 Microprocessor:** **Overview of Addressing Modes:** - "The 8085 microprocessor uses several addressing modes to specify operands for instructions. These modes determine how the operand is accessed." - Display a slide with the main categories of addressing modes. **b. Main Categories of Addressing Modes** - Immediate Addressing Mode - Register Addressing Mode - Direct Addressing Mode - Indirect Addressing Mode - Implicit Addressing Mode - **b. Explain the Addressing Modes of the 8085 Microprocessor:** **Detailed Explanation of Each Addressing Mode with examples** Show a detailed slide or diagram for each addressing mode with example 3. Exercise (5 minutes) – **Group Activity:** - Divide students into small groups and assign each group a set of instructions to categorize and explain the addressing modes used.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading - https://archive.nptel.ac.in/content/storage2/courses/106108100/pdf/Lecture_Notes/LNm1.pdf

	3. Homework Assign a short written task for students to create a few instructions using different addressing modes and explain the modes used. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. What addressing mode is used in this instruction?" How is the operand accessed in this addressing mode?" Why might this addressing mode be used in this context? 2. Open the floor for any remaining questions Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 6	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Implement Basic Loops in 8085 assembly language programmes. b. Perform Counting Operations using 8085 instructions to implement counting mechanisms. c. Apply Indexing Techniques for array manipulation and data handling in 8085 assembly language programs.
Teaching Aids (if any)	a. PPT on program techniques in 8085 b. 8085 online simulator
Teaching Development	1. **Introduction** (5 minutes) - Ask questions. - What are program techniques - Why do you think program techniques are necessary - What could be the possible benefits of using programming techniques? - Can anyone explain what a loop is in the context of programming? - Why do you think counting and indexing are important in programming? 2. **Development** (30 minutes) - Explain the concept of looping, indexing and counting with respect to 8085. - Discuss the importance of looping, counting, and indexing in programming. - Describe the practical applications of looping, counting, and indexing in real-world scenarios. - Define what a loop is in programming. - Discuss the need for loops in programs. - Explain the use of registers for counting. - Describe how to increment and decrement counters using instructions like INR and DCR. - Discuss the concept of indexing in memory operations. - Explain the use of HL register pair for indirect addressing. 3. Exercise (5 minutes) – - Write a program to display numbers from 1 to 5 using a loop. - Implement a countdown timer from 10 to 0 and display each number. - Write a program to find the sum of a given array of 5 numbers. Allow students to work on these problems and ask for help if needed.

Closure	1. Reinforce the importance of looping, counting, and indexing. 2. Suggested Reading Microprocessor Architecture, Programming, and Applications with the 8085 (5th Edition) by Ramesh S. Gaonkar. Pg. 227-230 3. Homework Write a program to find the maximum number in an array of 5 elements using looping and indexing techniques. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, why, Who?) Allow students to answer and discuss. 2. Provide clarifications and further explanations as needed. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 7	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Summarize various program techniques used in the 8085 microprocessors. b. Apply different program techniques in the 8085 microprocessors through practical examples.
Teaching Aids (if any)	a. PowerPoint slides on 8085 programming techniques b. Online 8085 simulators
Teaching Development	- **Introduction** (5 minutes) - Ask questions. - What is a program? - How many of you have some experience with programming? - What languages have you used? - Why do we need program techniques in 8085? - What do you hope to learn from today's lesson on 8085 programming techniques? **Development** (30 minutes) - Introduce the importance of learning different programming techniques in the 8085 microprocessors. **a. Overview of 8085 Programming Techniques (20 minutes)** Introduction to Programming Techniques: - Define what is meant by programming techniques in the context of the 8085 microprocessor. - Discuss the basic structure of an 8085 program (opcode, operands, etc.). **b. Common Techniques:** **Looping:** - Explain the concept of looping in programs. - Provide examples using instructions like JMP, CALL, and RET. **Condition Checking:** - Discuss the use of conditional jumps (JZ, JNZ, JC, JNC, etc.). - Illustrate with simple examples. **Subroutines:** - Define subroutines and their significance. - Explain CALL and RET instructions with examples. **Interrupt Handling:** - Briefly introduce interrupts and their handling in 8085. - Provide an example of an interrupt service routine. **c. Detailed Explanation and Examples (30 minutes)** **Looping Example:** - Present a detailed example of a program that uses looping to add

	numbers from an array. - Walk through the program step by step, explaining each instruction. **Condition Checking Example:** - Provide an example of a program that checks if a number is even or odd. - Explain the logic and flow of the program. **Subroutine Example:** - Discuss a program that uses subroutines to perform a specific task, such as finding the maximum number in an array. - Illustrate the use of CALL and RET instructions. **Interrupt Handling Example:** - Provide a simple example of an interrupt-driven program. - Explain how the 8085 handles interrupts and the role of the interrupt service routine. **d. Hands-On Practice (20 minutes)** - **Interactive Coding Session:** - Divide students into small groups. - Assign each group a specific programming task using the 8085 microprocessors (e.g., summing an array, checking for prime numbers, etc.). - Provide access to 8085 simulation software or kits. - Guide students as they write and debug their programs. Exercise (5 minutes) – - Have each group present their program and explain how they applied the discussed techniques. - Discuss any challenges faced and how they were overcome.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading - NPTEL Notes on programming techniques - https://archive.nptel.ac.in/courses/108/107/108107029/ 3. Homework Homework on writing and analyzing 8085 programs using the techniques discussed in the lesson. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. - Recap the key points of the lesson. - Highlight the importance of understanding and applying different programming techniques in the 8085 microprocessor. - Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 8	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Describe the function and purpose of interrupts in a microprocessor system. b. Differentiate between various types of interrupts in the 8085 microprocessors. c. Write assembly programs to handle interrupts using the 8085 microprocessor.
Teaching Aids (if any)	a. PPT on Interrupt handling in 8085 b. 8085 online simulator
Teaching Development	1. **Introduction** (5 minutes) Ask questions. - What do you understand by the term 'interrupt' in a computing context? - Can you think of real-world examples where interrupts might be used? - Why might interrupts be important in a microprocessor system? - Why do you think is interrupt handling necessary? 2. **Development** (30 minutes) - Define what an interrupt is. - Explain the purpose and importance of interrupts in a microprocessor system. - Explain the basic concept of interrupts. - Discuss the difference between hardware and software interrupts - Describe the interrupt process in 8085: how the microprocessor responds to an interrupt, interrupt acknowledge cycle, and returning from an interrupt. - Explain the various types of interrupts: **Vectored Interrupts**: TRAP, RST7.5, RST6.5, RST5.5 **Non-vectored Interrupts**: INTR - Describe the priority and masking of interrupts. **Exercise (5 minutes) –** - Write a program to handle both RST7.5 and RST6.5 interrupts. - Write a program to demonstrate the masking and unmasking of interrupts using the SIM instruction. - Have students use a simulator to debug the program and identify any issues.

Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading https://nptel.ac.in/courses/108107029 3. Homework Write a program that handles an interrupt to perform a specific task, such as updating a counter or toggling an LED, and explain each step in detail. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Reinforce the importance of interrupt handling. Clarify any doubts or concerns. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 9	Course Name: Microprocessors Topic: Basics of Programmable I/O	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of Programmable I/O in microprocessors. b. Explain how microprocessors interact with input and output devices. c. Identify the various types of I/O devices and their roles. d. Appreciate the significance of programmable I/O in real-life applications.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - Ask questions. - Introduce the basics of Programmable I/O. - Discuss the types of I/O devices (input, output, and input/output). - Introduce the formal definition of Programmable I/O. - Highlight the importance of Programmable I/O in microprocessor-based systems. - **Development** (30 minutes) a Programmable I/O - Introduce the concept of Programmable I/O in microprocessors. - Show a video explaining the basics of Programmable I/O. - Discuss the significance of control registers in I/O. - Talk about the different modes of I/O operation (e.g., programmed I/O, interrupt-driven I/O, direct memory access). b Interfacing - Discuss the methods of interfacing I/O devices with microprocessors. - Highlight real-life examples of systems using Programmable I/O. - Major players in Programmable I/O technologies: Intel, ARM, etc. - Advantages of Programmable I/O: o Flexibility o Efficient communication - Exercise (5 minutes) – Give students time to discuss various applications of Programmable I/O in different industries.

Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video Lecture https://youtu.be/51tjzIdATwg 3. Homework Ask students to create a diagram illustrating how a microprocessor communicates with an I/O device, including the role of control registers and data transfer methods. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. What is Programmable I/O, and how does it differ from non-programmable I/O? 2. What roles do control registers play in the functioning of Programmable I/O? 3. Why is Programmable I/O important in modern microprocessor-based systems? 4. Why is flexibility considered an advantage of Programmable I/O? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 10	Course Name: Microprocessors Topic: General Purpose Programmable Peripheral Devices – 8255A	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the function and structure of the 8255A Programmable Peripheral Interface (PPI). b. Explain the different modes of operation of the 8255A.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - Ask questions. - What is a peripheral device in the context of microprocessors? - Have you heard of any programmable peripheral devices? - What do you think is the role of a programmable peripheral interface? - Introduce the 8255A Programmable Peripheral Interface (PPI). - Discuss the need for peripheral interfacing in microprocessor systems. - **Development** (30 minutes) a 8255A Overview - Introduce the 8255A PPI: its function and importance. - Discuss the internal structure of the 8255A, including its ports (Port A, Port B, Port C) and control register. - Explain the different modes of operation: o Mode 0: Basic Input/Output o Mode 1: Strobed Input/Output o Mode 2: Bidirectional Data Bus - Discuss how the 8255A connects to a microprocessor and the role of control signals. - Exercise (5 minutes) – Give students time to discuss and sketch a basic interfacing circuit for the 8255A.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video Lecture https://youtu.be/Y_H_Pjn9JE0 3. Homework Ask students to make a schematic diagram showing the interfacing of the 8255A with a microprocessor, including the connection to peripherals and control signals.

	Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. What is the primary function of the 8255A PPI in a microprocessor system? 2. What are the different modes of operation of the 8255A, and how do they differ from each other? 3. What role do the control signals play in the operation of the 8255A? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 11	Course Name: Microprocessors Topic: General Purpose Programmable Peripheral Devices – 8255A	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Illustrate how the 8255A interfaces with a microprocessor and other peripheral devices. b. Appreciate the significance of the 8255A in real-life applications.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - Ask questions. - What is a peripheral device in the context of microprocessors? - Have you heard of any programmable peripheral devices? - What do you think is the role of a programmable peripheral interface? - Introduce the 8255A Programmable Peripheral Interface (PPI). - Discuss the need for peripheral interfacing in microprocessor systems. - **Development** (30 minutes) a **Interfacing and Applications** - Demonstrate how to interface the 8255A with a microprocessor. - Provide examples of projects and systems using the 8255A. - Discuss major players and advancements related to the 8255A. - Highlight the advantages of using the 8255A: o Versatility o Ease of interfacing o Efficient handling of multiple I/O devices - Exercise (5 minutes) – Give students time to discuss and sketch a basic interfacing circuit for the 8255A.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video Lecture https://youtu.be/Y_H_Pjn9JE0 3. Homework Ask students to make a schematic diagram showing the interfacing of the 8255A with a microprocessor, including the connection to peripherals and control signals.. Spend 5 minutes to wrap up and consolidate the learnings

Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. Why is it necessary to use a programmable peripheral interface like the 8255A in microprocessor systems? 2. Why might a system designer choose Mode 1 (Strobed Input/Output) over Mode 0 (Basic Input/Output)? 3. Why is the flexibility of the 8255A's ports important in designing microprocessor systems? 4. How do the control register and the ports (Port A, Port B, Port C) of the 8255A interact during different modes of operation? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 12	Course Name: Microprocessors Topic: General Purpose Programmable Peripheral Devices – 8259A	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the function and structure of the 8259A Programmable Interrupt Controller (PIC). b. Explain the different modes of operation and priority schemes of the 8259A.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - Ask questions. - What is an interrupt in the context of microprocessors? - Why is interrupt management important in computing systems? - Have you heard of any interrupt controllers? - Introduce the 8259A Programmable Interrupt Controller. - Discuss the need for an interrupt controller in microprocessor-based systems. - **Development** (30 minutes) a 8259A Overview - Introduce the 8259A PIC: its function and importance in interrupt handling. - Discuss the internal structure of the 8259A, including the interrupt request (IR) lines, interrupt service routine (ISR) handling, and control registers. - Explain the modes of operation: o Single Mode: Single 8259A handling interrupts. o Cascaded Mode: Multiple 8259As handling interrupts (master-slave configuration). o Describe the priority schemes: o Fixed Priority: Fixed priority levels for each interrupt. o Rotating Priority: Priority changes over time. - Exercise (5 minutes) – Give students time to discuss and sketch a basic interfacing circuit for the 8259A, including connections to interrupts and control signals.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video Lecture https://youtu.be/Uq8X76BrQR8

	3. Homework Ask students to create a flowchart illustrating the process of handling an interrupt using the 8259A, including the steps from interrupt request to the execution of the interrupt service routine. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. What is the primary function of the 8259A PIC in a microprocessor-based system? 2. What are the different modes of operation of the 8259A, and how do they affect interrupt handling? 3. What is the difference between fixed priority and rotating priority schemes in the 8259A? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 13	Course Name: Microprocessors Topic: General Purpose Programmable Peripheral Devices – 8259A	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Illustrate how the 8259A interfaces with a microprocessor and handles interrupt requests. b. Appreciate the role of the 8259A in managing interrupts in real-time systems.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - Ask questions. - What is an interrupt in the context of microprocessors? - Why is interrupt management important in computing systems? - Have you heard of any interrupt controllers? - Introduce the 8259A Programmable Interrupt Controller. - Discuss the need for an interrupt controller in microprocessor-based systems. - **Development** (30 minutes) a Interfacing and Applications - Demonstrate how to interface the 8259A with a microprocessor. - Provide examples of real-world applications where the 8259A is used for managing interrupts. - Discuss major advancements and players related to interrupt controllers. - Highlight the advantages of using the 8259A: o Efficient interrupt handling o Flexible priority management o Improved system performance - Exercise (5 minutes) – Give students time to discuss and sketch a basic interfacing circuit for the 8259A, including connections to interrupts and control signals.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video Lecture https://youtu.be/u5R924YVx0k 3. Homework Ask students to create a flowchart illustrating the process of

	handling an interrupt using the 8259A, including the steps from interrupt request to the execution of the interrupt service routine. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. Why is interrupt management crucial in modern computing systems? 2. Why might a system designer choose to use multiple 8259A PICs in a cascaded mode rather than a single one? 3. Why is it important to have different priority schemes for handling interrupts? 4. How does the 8259A manage multiple interrupt requests simultaneously? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 14	Course Name: Microprocessors Topic: Direct Memory Access Controller – 8237	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the function and structure of the 8237 Direct Memory Access (DMA) Controller. b. Explain the different modes of operation and channel configurations of the 8237.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - Ask questions. - What is Direct Memory Access (DMA) and why is it used? - Can you name any scenarios where DMA might be beneficial? - Have you heard of the 8237 DMA Controller? - Introduce the 8237 Direct Memory Access Controller. - Discuss the need for DMA in microprocessor systems. - **Development** (30 minutes) a 8237 Overview - Introduce the 8237 DMA Controller: its function and importance in data transfer. - Discuss the internal structure of the 8237, including its channels and control registers. - Explain the modes of operation: o **Burst Mode:** DMA transfers data in bursts, allowing the CPU to use the bus during non-transfer periods. o **Cycle Stealing Mode:** DMA takes control of the bus only for one bus cycle at a time, allowing the CPU to perform other tasks in between. o **Demand Mode:** DMA transfers data only when the peripheral requests it. - Describe the configuration of DMA channels and how they interact with the CPU and memory. - Exercise (5 minutes) – Give students time to discuss and sketch a basic DMA configuration using the 8237, including connections to the microprocessor, memory, and peripherals
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video Lecture

	https://youtu.be/F8iR_3488PA 3. Homework Ask students to create a flowchart illustrating how the 8237 DMA Controller handles a data transfer request, including the steps from initiation to completion of the transfer. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. What is the main function of the 8237 DMA Controller in a microprocessor system? 2. What are the different modes of operation of the 8237, and how do they impact data transfer? 3. What role do the DMA channels play in the operation of the 8237? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 15	Course Name: Microprocessors Topic: Direct Memory Access Controller – 8237	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Illustrate how the 8237 interfaces with a microprocessor and memory. b. Appreciate the significance of DMA in improving system performance.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - Ask questions. - What is Direct Memory Access (DMA) and why is it used? - Can you name any scenarios where DMA might be beneficial? - Have you heard of the 8237 DMA Controller? - Introduce the 8237 Direct Memory Access Controller. - Discuss the need for DMA in microprocessor systems. - **Development** (30 minutes) b Interfacing and Applications - Demonstrate how to interface the 8237 with a microprocessor and memory. - Provide examples of real-world applications where the 8237 is used for efficient data transfer. - Discuss major advancements and players related to DMA controllers. - Highlight the advantages of using the 8237: o Increased data transfer speed o Reduced CPU overhead o Improved system performance - Exercise (5 minutes) – Give students time to discuss and sketch a basic DMA configuration using the 8237, including connections to the microprocessor, memory, and peripherals
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video Lecture https://youtu.be/bz1FX1kGfIw https://youtu.be/TXPXf8di1m4 3. Homework Ask students to create a flowchart illustrating how the 8237 DMA

	Controller handles a data transfer request, including the steps from initiation to completion of the transfer. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. Why is Direct Memory Access (DMA) used in microprocessor systems instead of traditional CPU-controlled data transfer? 2. Why is it important to configure the DMA channels properly when using the 8237? 3. How does the 8237 DMA Controller improve system efficiency compared to CPU-controlled data transfer? 4. How does the 8237 manage data transfer between memory and peripherals without heavily involving the CPU? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 16	Course Name: Microprocessors Topic: Architecture of 8086	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the architecture of the 8086 microprocessors. b. Explain how the Bus Interface Unit (BIU) and Execution Unit (EU) work together. c. Understand the concept of segmentation and its significance in memory management.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - Ask questions. - What is a microprocessor? - Why is it important in computing? - Have you heard of the **8086 microprocessor**? Why is it significant in computer history? - Introduce the **8086 microprocessor** and explain its role in the development of modern computers. - Explain the concept of **16-bit architecture** and why it was an advancement at the time. - **Development** (30 minutes) **Internal Architecture of 8086**: • Bus Interface Unit (BIU) • Execution Unit (EU) **Segmentation**: • Explain how memory is divided into segments and how segment registers like CS, DS, SS, and ES are used to manage different types of data. • llustrate how segmentation helps in efficient memory management in larger programs. - Exercise (5 minutes) – Ask students to outline the block diagram of the 8086 microprocessors, labelling the BIU, EU, buses, and segment registers. Discuss the function of each component and how they interact.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video https://youtu.be/P_iZzqqjQSI

	https://archive.nptel.ac.in/courses/108/103/108103157/ 3. Homework Ask students to create create a flowchart that explains the fetch-decode-execute cycle in the 8086 microprocessor, illustrating how an instruction is fetched from memory, decoded, and executed. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. Why does the 8086 use segmentation for memory management instead of a flat memory model? 2. How do the Bus Interface Unit (BIU) and Execution Unit (EU) work together in executing instructions? 3. Why is the 8086 architecture significant for modern computing? 4. How does the 8086 handle data flow between memory and peripherals? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 17	Course Name: Microprocessors Topic: Memory Address space and data organization	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the memory address space of the 8086 microprocessor. b. Explain the concept of segmented memory and how addresses are formed. c. Illustrate how data is organized and accessed in the 8086 memory.
Teaching Aids (if any)	a. ICT Usage

Teaching Development	- **Introduction** (5 minutes)
	- Ask questions.
	- What is memory address space, and why is it important in microprocessor systems?
	- Can anyone explain how the 8086 manages its memory space differently from earlier processors?
	- Introduce the concept of 8086 memory address space.
	- Explain the idea of segmented memory architecture, where memory is divided into segments.
	- Highlight why the 8086 uses a 20-bit address bus and how it allows for 1 MB of memory access despite being a 16-bit processor.
	
	- **Development** (30 minutes)
	
	a. Memory Address Space in 8086:
	- Segmented Memory:
	- Explain that the 8086 divides memory into segments, with each segment being 64 KB in size.
	- Describe the four main segment types:
	  - Code Segment (CS).
	  - Data Segment (DS)
	  - Stack Segment (SS)
	  - Extra Segment (ES)
	
	-
	b. Data Organization:
	- Explain how the 8086 can access data as bytes (8 bits) or words (16 bits).
	- Discuss how even and odd memory addresses are used for organizing data, and how words are stored in two consecutive memory locations.
	
	c. Exercise (5 minutes) –

	Give students time to calculate physical memory addresses given segment and offset values,and discuss how data is organized in memory (byte/word access and little-endian format).
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video Lecture https://youtu.be/PVSuHva0Wrk 3. Homework Ask students to draw a memory map for a simple program, labeling the Code Segment, Data Segment, Stack Segment, and showing the physical addresses for different segment and offset values. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. What is segmented memory architecture, and how does it allow the 8086 to access more than 64 KB of memory? 2. How are physical addresses calculated using the segment and offset method? 3. Why does the 8086 use a 20-bit address bus despite being a 16-bit processor? 4. How does data organization (bytes and words) affect memory access and performance? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 18	Course Name: Microprocessors Topic: Memory Address space and data organization	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the memory address space of the 8086 microprocessor. b. Explain the concept of segmented memory and how addresses are formed. c. Illustrate how data is organized and accessed in the 8086 memory.
Teaching Aids (if any)	a. ICT Usage

Teaching Development	- **Introduction** (5 minutes)
	- Ask questions.
	- What is memory address space, and why is it important in microprocessor systems?
	- Can anyone explain how the 8086 manages its memory space differently from earlier processors?
	- Introduce the concept of 8086 memory address space.
	- Explain the idea of segmented memory architecture, where memory is divided into segments.
	- Highlight why the 8086 uses a 20-bit address bus and how it allows for 1 MB of memory access despite being a 16-bit processor.
	
	- **Development** (30 minutes)
	
	a. Memory Address Space in 8086:
	- Segmented Memory:
	- Explain that the 8086 divides memory into segments, with each segment being 64 KB in size.
	- Describe the four main segment types:
	  - Code Segment (CS).
	  - Data Segment (DS)
	  - Stack Segment (SS)
	  - Extra Segment (ES)
	
	-
	b. Data Organization:
	- Explain how the 8086 can access data as bytes (8 bits) or words (16 bits).
	- Discuss how even and odd memory addresses are used for organizing data, and how words are stored in two consecutive memory locations.
	
	c. Exercise (5 minutes) –

	Give students time to calculate physical memory addresses given segment and offset values,and discuss how data is organized in memory (byte/word access and little-endian format).
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video Lecture https://youtu.be/PVSuHva0Wrk 3. Homework Ask students to draw a memory map for a simple program, labeling the Code Segment, Data Segment, Stack Segment, and showing the physical addresses for different segment and offset values. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. What is segmented memory architecture, and how does it allow the 8086 to access more than 64 KB of memory? 2. How are physical addresses calculated using the segment and offset method? 3. Why does the 8086 use a 20-bit address bus despite being a 16-bit processor? 4. How does data organization (bytes and words) affect memory access and performance? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 19	Course Name: Microprocessors Topic: Segment registers and memory segmentation,	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the function of segment registers in the 8086 microprocessors. b. Explain the concept of memory segmentation and how it manages memory efficiently. c. Calculate physical addresses using segment and offset values.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - Ask questions. - What do you understand by memory segmentation? - Why do you think microprocessors need special segment registers? - Introduce the 8086 microprocessor and its use of segmented memory architecture. - Briefly explain that segmentation allows the 8086 to access up to 1 MB of memory even though it is a 16-bit processor. - **Development** (30 minutes) **a. Introduction to Segment Registers**: - Explain that the 8086 microprocessor has **four segment registers**, each responsible for a different part of the memory. - Define **memory segmentation** as dividing memory into smaller, manageable sections (or segments) to organize data and instructions. - Explain that each segment can be **64 KB** in size and is addressed through a **segment** addressing scheme. **b. Address Formation**: - Discuss how **segment registers** provide the **base address** for each segment, and the **offset** represents the specific location within that segment. - Provide the formula for calculating a physical address: Physical Address=(Segment Address×16)+Offset\text{Physical Address} = (\text{Segment Address} \times 16) + \text{Offset}Physical Address=(Segment Address×16)+Offset - **c. Segment-Offset Addressing**: - Walk students through several **example calculations** where they determine physical addresses given different segment and offset values.

	- Explain how each segment register works in conjunction with the offset to form a complete 20-bit address. **d. Benefits of Segmentation**: ○ Modularity ○ Larger Program Size ○ Memory Protection ○ Multitasking - Exercise (5 minutes) – Give students time to calculate a few **physical addresses** given various segment and offset values using the segment addressing method.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video Lecture https://youtu.be/PVSuHva0Wrk 3. Homework Ask students to create a diagram showing the **four segment registers** (CS, DS, SS, ES) and provide examples of how each is used in a program. Include sample **address calculations** for practice.
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. What is memory segmentation, and why is it used in the 8086 microprocessor? 2. How do the segment registers (CS, DS, SS, ES) work together to manage different segments of memory? 3. Why does the 8086 use the segment method for calculating physical addresses? 4. How does memory segmentation improve system performance and organization? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 20	Course Name: Microprocessors Topic: Generating memory addresses & IO address space	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Explain the concept of memory segmentation and how it manages memory efficiently. b. Calculate physical addresses using segment and offset values.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - Ask questions. - What do you understand by memory segmentation? - Why do you think microprocessors need special segment registers? - Introduce the 8086 microprocessor and its use of segmented memory architecture. - Briefly explain that segmentation allows the 8086 to access up to 1 MB of memory even though it is a 16-bit processor. - **Development** (30 minutes) **a. Address Formation**: - Discuss how **segment registers** provide the **base address** for each segment, and the **offset** represents the specific location within that segment. - Provide the formula for calculating a physical address: Physical Address=(Segment Address×16)+Offset\text{Physical Address} = (\text{Segment Address} \times 16) + \text{Offset}Physical Address=(Segment Address×16)+Offset - **c. Segment-Offset Addressing**: - Walk students through several **example calculations** where they determine physical addresses given different segment and offset values. - Explain how each segment register works in conjunction with the offset to form a complete 20-bit address. - Exercise (5 minutes) – Give students time to calculate a few **physical addresses** given various segment and offset values using the segment addressing method.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video Lecture https://youtu.be/PVSuHva0Wrk

	3. Homework Ask students to create a diagram showing the **four segment registers** (CS, DS, SS, ES) and provide examples of how each is used in a program. Include sample **address calculations** for practice.
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. Why does the 8086 use the segment method for calculating physical addresses? 2. How does memory segmentation improve system performance and organization? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 21	Course Name: Microprocessors Topic: Addressing Modes	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Identify and differentiate between sequential control flow instructions and control transfer instructions. b. Describe the role of each instruction type in the execution of programs.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - Begin by asking students what they understand about the flow of program execution. - Explain that in the 8086 microprocessor, instructions are executed in a particular flow, which can either be: o **Sequential Control Flow** (where the next instruction is executed in sequence), or o **Control Transfer** (where the flow of control jumps to a different part of the program based on certain conditions or subroutine calls). - Introduce the idea that control flow instructions impact how a program progresses from one instruction to the next. - **Development** (30 minutes) - **Sequential Control Flow Instructions** o **Definition**: Instructions executed one after the other in memory address order. o Examples of sequential control instructions include basic arithmetic, data movement, and logical operations. - Exercise (5 minutes) – Provide students with a series of mixed instructions and ask them to categorize each one as either sequential or control transfer.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video Lecture https://archive.nptel.ac.in/courses/108/103/108103157/# 3. Homework Ask students to write a simple program that includes both sequential and control transfer instructions. They should annotate each instruction with its type (sequential or control transfer).
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss.

	1. Why do we need both sequential and control transfer instructions in a program? 2. How does conditional control transfer improve program flexibility? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 22	Course Name: Microprocessors Topic: Addressing Modes	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Identify and differentiate between sequential control flow instructions and control transfer instructions. b. Describe the role of each instruction type in the execution of programs.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - Begin by asking students what they understand about the flow of program execution. - Introduce the idea that control flow instructions impact how a program progresses from one instruction to the next. - **Development** (30 minutes) - **Sequential Control Flow Instructions** o Provide example instructions such as `ADD`, `MOV`, `SUB`, and explain their role in a sequential execution. - **Control Transfer Instructions (15 minutes):** o Definition o Discuss types of control transfer instructions: ▪ **Unconditional Jumps** (e.g., `JMP`) – Directly transfer control to another part of the program. ▪ **Conditional Jumps** (e.g., `JZ`, `JC`, `JNZ`) – Transfer control based on the result of a previous operation. ▪ **Procedure Calls** (e.g., `CALL`, `RET`) – Used to call subroutines and return. ▪ **Interrupt Instructions** – Control can be transferred in response to an interrupt signal. o Illustrate with examples, showing how these instructions redirect the program flow depending on specific conditions or calls.s - Exercise (5 minutes) – Provide students with a series of mixed instructions and ask them to categorize each one as either sequential or control transfer.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video Lecture https://archive.nptel.ac.in/courses/108/103/108103157/#

	3. Homework Ask students to write a simple program that includes both sequential and control transfer instructions. They should annotate each instruction with its type (sequential or control transfer).
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. Why do we need both sequential and control transfer instructions in a program? 2. How does conditional control transfer improve program flexibility? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 23	Course Name: Microprocessors Topic: Minimum mode	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Describe the concept of minimum mode operation in the 8086 microprocessors. b. Identify the hardware configuration required for minimum mode. c. Explain the role of control signals in minimum mode and how they support single-processor operation.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - What do you understand by operating mode in microprocessors? - Why would a microprocessor need different operating modes?" - Introduce the concept that the 8086 microprocessors can operate in two modes: - Minimum Mode (for single-processor systems) - Maximum Mode (for multiprocessor systems) Explain that minimum mode is the focus of this lesson and is often used in simpler, single-CPU configurations. - **Development** (30 minutes) - Overview of Minimum Mode o Define minimum mode as the configuration where the 8086 operates as a single processor without any other processors in the system. o Explain that in minimum mode, the 8086 internally generates control signals for memory and I/O, which simplifies the design of the system. o Describe how this mode is selected by setting the **MN/$\overline{MX}$** pin to high (logic 1). - **Minimum Mode Signals** o **Explain each control signal** generated by the 8086 in minimum mode: ▪ **HOLD and HLDA**: Explain the hold and hold acknowledge mechanism, used for DMA (Direct Memory Access). ▪ **INT$\overline{A}$ (Interrupt Acknowledge)**: Used when an external device interrupts the CPU. ▪ **M/I$\overline{O}$ (Memory/Input-Output)**: Distinguishes between memory and I/O operations. ▪ **R$\overline{D}$ and W$\overline{R}$ (Read and Write)**: Controls data transfer direction. ▪ **ALE (Address Latch Enable)**: Activates to

	latch the address on the multiplexed bus. ▪ **DT/$\overline{R}$ (Data Transmit/Receive)**: Determines data direction on the bus. o Provide examples to demonstrate the role of these signals during read and write operations. - **Address/Data Bus Multiplexing** o Explain how the 8086 uses multiplexed address and data lines (AD0-AD15) in minimum mode to save on pin count. o Describe the timing in which the address is available during the early part of the bus cycle, while data is transferred later. o Illustrate with a basic timing diagram showing address and data phases. - - Exercise (5 minutes) – • Ask students to summarize the importance of minimum mode for single-processor systems and the specific control signals it generates.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video Lecture https://youtu.be/P_iZzqqjQSI https://archive.nptel.ac.in/courses/108/103/108103157/ 3. Homework Ask students to draw a simplified block diagram of an 8086 system in minimum mode, showing connections with memory and I/O.
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. Why might a designer choose to use minimum mode instead of maximum mode? 2. How does multiplexing of address and data lines save resources in minimum mode? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 24	Course Name: Microprocessors Topic: Maxima mode	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Describe the concept of maximum mode operation in the 8086 microprocessors. b. Identify the hardware configuration required for maximum mode. c. Explain the role of external bus controller and control signals in supporting multiprocessor operation.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - What do you think might be the difference between minimum and maximum mode in the 8086? - Why might a system need to support multiple processors? - Introduce the concept of **maximum mode**, which is used when multiple processors or co-processors (like the 8087 math co-processor) are part of the system. - Explain that the **maximum mode** is selected by setting the **MN/M$\overline{X}$** pin to low (logic 0) and requires an external bus controller to manage the control signals. - **Development** (30 minutes) - **Overview of Maximum Mode** - Define **maximum mode** as the configuration in which the 8086 microprocessor operates in a multiprocessor environment. - Explain that in maximum mode, the 8086 relies on an external controller (Intel 8288 Bus Controller) to generate control signals. - Emphasize the need for maximum mode in systems where the 8086 shares the bus with other processors or peripherals, allowing for resource sharing. - **Role of 8288 Bus Controller** o **Introduction to the 8288 Bus Controller**: Explain that the 8086 outputs **status signals** (S0, S1, S2) in maximum mode, which are used by the 8288 Bus Controller to generate the necessary control signals for memory and I/O operations. o Discuss key control signals generated by the 8288 Bus Controller: ▪ **MRDC$\overline{}$ (Memory Read Command)** – Controls reading from memory. ▪ **MWTC$\overline{}$ (Memory Write Command)** – Controls writing to memory. ▪ **IORC$\overline{}$ (I/O Read Command)** and **IOWC$\overline{}$ (I/O**

	Write Command) – Control data transfer with I/O devices. ▪ **INTA̅ (Interrupt Acknowledge)** – Indicates response to an interrupt request. o Show a diagram of the 8086 in maximum mode, illustrating the connection between the microprocessor, 8288 Bus Controller, and other system components. - **Status and Control Signals in Maximum Mode** o Explain the **status signals** (S0, S1, S2) that the 8086 outputs in maximum mode and how they determine the type of operation (e.g., memory read, memory write, I/O read, I/O write). o Discuss the **RQ/GT̅ (Request/Grant)** lines: ▪ Explain that these lines are used for **bus arbitration** when multiple processors or DMA controllers request access to the system bus. ▪ Describe how these signals allow efficient communication in a multiprocessor setup. o Provide examples of how the status signals are decoded by the 8288 to generate appropriate control signals for different types of operations. - - Exercise (5 minutes) – - Present a series of status signals (S0, S1, S2) and ask students to determine the type of operation (memory read, memory write, I/O read, or I/O write) that would be performed.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video Lecture https://youtu.be/P_iZzqqjQSI https://archive.nptel.ac.in/courses/108/103/108103157/ 3. Homework Ask students to draw a simplified block diagram of an 8086 system in maxima mode, showing connections with memory and I/O.
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. Why is an external bus controller necessary in maximum mode? 2. How does the maximum mode configuration improve system performance in multiprocessor environments? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 25	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the fundamental concepts of RISC (Reduced Instruction Set Computer) architecture. b. Differentiate between RISC and CISC (Complex Instruction Set Computer) architectures.
Teaching Aids (if any)	a. PPT on The RISC revolution
Teaching Development	1. **Introduction** (5 minutes) - Start with a brief recap of computer architecture basics. - Present an engaging question: "What do you think made processors more efficient in the 1980s and beyond?" - Outline the key points that will be covered in the lesson. 2. **Development** (30 minutes) - Describe the development of computer architecture before RISC. - Introduce the concept of the RISC revolution in the 1980s. - Highlight the shift from complex, multi-cycle instructions to simpler, single-cycle operations. - Explain RISC: Emphasis on a small, highly optimized set of instructions. - Compare with CISC: Larger, more complex set of instructions. - Discuss the impact on processing speed and energy efficiency. - Use examples and diagrams to illustrate key differences. - Faster execution due to reduced instruction complexity. - Simplicity in pipelining and improved parallelism. - Discuss how the RISC philosophy is still relevant today. - Mention ARM processors as a successful RISC implementation in modern devices. - Briefly discuss the role of RISC in embedded systems and mobile computing. **Exercise (5 minutes) –** - Question: *"Which type of architecture is more power-efficient and commonly used in smartphones?"* - Encourage students to share their thoughts.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.

	2. Assign a reading on the evolution of microprocessors with a focus on RISC. 3. Suggested material https://onlinecourses.nptel.ac.in/noc22_cs88/preview Spend 5 minutes to wrap up and consolidate the learning
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 26	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Discuss the primary characteristics of RISC architecture. b. Differentiate between RISC and CISC architectures. c. Understand practical examples and applications of RISC-based processors.
Teaching Aids (if any)	a. PPT on RISC architecture
Teaching Development	1. **Introduction** (5 minutes) - Introduce RISC architecture, highlighting its purpose in simplifying the instruction set for more efficient processing. - Provide a short history of RISC development (mention key contributors like John Cocke). - Briefly remind students about CISC architecture for comparison purposes. **2. Development** (30 minutes) - Explain Characteristics of RISC Architecture - Discuss how these reduce the need for memory access. - Explain the use of pipelining to improve performance. - Mention popular RISC processors (e.g., ARM, MIPS, PowerPC). - Present a side-by-side comparison of RISC and CISC features. - Facilitate a discussion on the advantages and disadvantages of each architecture. - Take questions to clarify any doubts. - Discuss applications of RISC architecture in modern devices (e.g., smartphones, embedded systems). - Mention companies that implement RISC processors (like ARM Holdings). **Exercise (5 minutes) –** - Group Discussion: Divide students into small groups and ask them to brainstorm where they think RISC architecture has the most impact.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested material https://www.youtube.com/watch?v=bkjqOKGQuNY

	Spend 5 minutes to wrap up and consolidate the learning
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 27	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Explain the concept of pipelining specifically in RISC architecture. b. Define pipeline bubbles and understand how and why they occur in RISC-based pipelines.
Teaching Aids (if any)	a. PPT on pipeline bubbles in RISC
Teaching Development	1. **Introduction** (5 minutes) - Remind students about RISC architecture from the previous lesson. - Emphasize RISC's simplified instructions that generally execute in one clock cycle, making it ideal for pipelining. - Explain how RISC processors use pipelining to execute multiple instructions concurrently by overlapping different stages of instruction execution. - Use a simple diagram to show the stages of a RISC pipeline, such as instruction fetch, decode, execute, memory access, and write-back. - Introduce *pipeline bubbles* as idle cycles that occur when the pipeline cannot proceed smoothly, resulting in a performance drop. - Describe how RISC's goal of achieving one instruction per clock cycle can be disrupted by these bubbles. - Explain how dependencies between instructions cause pipeline stalls. - Use a simple RISC code sequence to illustrate a data hazard, showing how one instruction depends on the result of another. 2. **Development** (30 minutes) - Describe how branch instructions and jumps disrupt the smooth flow of instructions in the pipeline. - Show how a conditional branch can cause a bubble if the next instruction cannot be fetched in time. - Discuss resource conflicts, although these are less common in RISC due to simplified architecture. - Explain how bubbles increase the number of cycles needed to execute instructions, reducing overall efficiency. - Use a pipeline diagram to show how bubbles interrupt the flow and reduce throughput. - Engage students in a discussion on the significance of minimizing bubbles in a RISC processor compared to a CISC processor. - Explain how data can be forwarded between stages to minimize stalls. - Discuss how the processor temporarily pauses pipeline stages to handle dependencies. - Explain the use of branch prediction to guess the outcome of

	branches and reduce bubbles. - Describe how RISC processors can reorder instructions to minimize the impact of branches. **Exercise (5 minutes) –** - Have students write down one question they still have about pipeline bubbles or a method they found interesting. - Encourage students to share their thoughts.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. Ask students to research how a modern RISC processor, like the ARM Cortex series, handles pipeline bubbles. 2. Suggested material https://www.youtube.com/watch?v=46virIBg_3w Spend 5 minutes to wrap up and consolidate the learning
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 28	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Explain how external memory is accessed in RISC systems. b. Understand the role of the Load/Store architecture in external memory access. c. Discuss the impact of memory access on performance and techniques used to optimize it.
Teaching Aids (if any)	a. PPT on Accessing External Memory in RISC Systems
Teaching Development	1. **Introduction** (5 minutes) - Ask, "What types of memory are commonly used in a computer system?" to engage students and activate prior knowledge. - Briefly explain the difference between internal (register) memory and external memory, such as RAM, ROM, and cache. - Discuss why external memory is crucial for running large programs and storing data in RISC systems. - Explain the Load/Store architecture, a core concept of RISC, where only specific instructions (Load and Store) interact with memory. - Transfers data from memory to a register. - Transfers data from a register to memory. - Discuss how this design simplifies the instruction set and speeds up the execution of operations that don't involve memory access. - Use a diagram to show the flow of data between CPU registers and external memory in a RISC system. 2. **Development** (30 minutes) - Introduce common addressing modes used in RISC systems for external memory access. - Write simple RISC assembly code examples on the board to demonstrate memory access using Load and Store instructions. - Discuss how accessing external memory is slower compared to accessing data in registers, due to latency. - Introduce the concept of cache memory as a way to speed up data access. - Briefly explain how cache memory reduces access time. - Discuss strategies like loop unrolling and data alignment to optimize memory access. **Exercise (5 minutes) –** - Provide a RISC code snippet that accesses external memory inefficiently. Ask students to work in pairs to suggest

	improvements. - Review a few answers as a class and provide feedback. - Encourage students to share their thoughts.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Assign a short research task on how memory management is handled in a specific RISC processor, like the ARM Cortex-M series. 3. Suggested material https://www.youtube.com/watch?v=tas2eUavhRE Spend 5 minutes to wrap up and consolidate the learning
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 29	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Define branch penalties and understand how they affect pipeline performance. b. Explain why branch instructions create challenges in pipelined RISC systems. c. Discuss techniques used in RISC systems to minimize or reduce branch penalties
Teaching Aids (if any)	a. PPT on Reducing Branch Penalties in RISC Systems
Teaching Development	1. **Introduction** (5 minutes) - Ask students, "What happens in a pipelined system when the CPU encounters a branch instruction?" to review prior knowledge on pipelining. - Explain branch penalties as the delay or performance loss caused by branch instructions in a pipelined processor. - Use a diagram to show how a pipeline is stalled or flushed when a branch is encountered, creating a penalty. - Explain that branches are conditional jumps in the program that disrupt the instruction flow, leading to pipeline stalls. - Walk through a simple RISC code example that involves a conditional branch and highlight how the pipeline must wait for the branch outcome. 2. **Development** (30 minutes) - Discuss how frequent branches can significantly degrade performance, especially in RISC systems with deep pipelines. - Explain simple prediction strategies like always assuming a branch is taken or not taken. - Discuss more advanced techniques that use history to make predictions (e.g., 1-bit and 2-bit predictors). - Briefly introduce how a BHT keeps track of the outcomes of previous branches. - Provide a simple example of how dynamic prediction works and discuss its accuracy. - Explain how a BTB is used to store the addresses of previously executed branch instructions to speed up target address computation. - Use a diagram to show how a BTB works within the pipeline. - Describe delayed branching, where the compiler rearranges instructions so that useful operations are executed while waiting for the branch decision.

	- Show a RISC code snippet where delayed branching is applied to reduce penalties. - Explain loop unrolling as a compiler optimization technique that reduces the number of branches in loops. - Use an example of a loop to show how unrolling decreases branch frequency and improves pipeline efficiency. - Introduce speculative execution, where the processor makes educated guesses about the path to take and continues execution accordingly. - Explain the risks and the need for mechanisms to roll back if a speculation is incorrect. **Exercise (5 minutes) –** Ask a question: "Which technique do you think is most effective for reducing branch penalties and why?" Have students discuss in small groups and then share their answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. 2. Assign a short research activity on how a modern RISC processor, like ARM or RISC-V, implements branch prediction. 3. Suggested material https://www.youtube.com/watch?v=0Ce22P_jeLQ Spend 5 minutes to wrap up and consolidate the learning
Evaluation	-Recap the techniques discussed, emphasizing the importance of minimizing branch penalties to maintain pipeline efficiency in RISC systems. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 30	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the basic architecture and features of ARM processors. b. Discuss the design philosophy of ARM and its focus on energy efficiency and performance. c. Explain the instruction set and modes of operation used in ARM processors.
Teaching Aids (if any)	a. PPT on Introduction to ARM Processors
Teaching Development	1. **Introduction** (5 minutes) - Provide a brief history of ARM (originally Acorn RISC Machine, now Advanced RISC Machine) and how it became a dominant player in the processor market. - Mention examples of everyday devices that use ARM processors, like smartphones, tablets, and embedded systems. - Ask, "Why do you think ARM processors are popular in mobile and embedded devices?" - Recap the Reduced Instruction Set Computing (RISC) principles that ARM architecture is based on, emphasizing simplicity and efficiency. 2. **Development** (30 minutes) - Explain how ARM uses pipelining to speed up instruction execution. Mention how modern ARM processors have multiple pipelines to handle instructions efficiently. - Describe the register set of ARM processors, focusing on general-purpose registers and special-purpose registers like the Program Counter (PC) and Link Register (LR). - Explain how ARM uses the Load/Store model, where memory operations are handled separately from computational instructions. - Discuss how the Thumb instruction set provides a more compact code size, which is beneficial for memory-constrained environments. - Highlight ARM's design for energy efficiency, making it ideal for mobile and battery-powered devices. - Introduce the different operating modes (e.g., User, Supervisor, Interrupt) and how they support multitasking and exception handling. - Give an overview of commonly used ARM instructions (e.g., data processing, branch, load/store instructions). - Show a simple ARM assembly code example and walk students through it, explaining how each instruction works. - Briefly compare ARM's architecture to x86, focusing on the differences in design philosophy and application areas. - Discuss how ARM processors are used in smartphones, tablets, and wearables. - Highlight the use of ARM in embedded systems like automotive controls, IoT devices, and industrial automation. - Mention ARM's growing presence in data centers and supercomputers.

	Exercise (5 minutes) – - Conduct a quick quiz using questions related to ARM architecture, features, and instruction set, using tools like Mentimeter or a simple class discussion. - Pose a question: "How do you think ARM's energy efficiency contributes to its widespread use in mobile devices?" Encourage students to share their ideas.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. Recap the ARM architecture's main points, including its RISC principles, power efficiency, and key applications Ask students to research the latest developments in ARM technology, such as ARMv9 architecture or ARM's role in supercomputing. 2. Suggested material https://www.youtube.com/watch?v=4VRtujwa_b8 Spend 5 minutes to wrap up and consolidate the learning
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 31	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Identify and describe the different types of registers used in ARM processors. b. Understand the roles of general-purpose and special-purpose registers in ARM architecture.
Teaching Aids (if any)	a. PPT on registers in ARM
Teaching Development	1. **Introduction** (5 minutes) - Start with a question to activate prior knowledge: "What is the role of registers in a processor?" - Explain that registers are small, fast storage locations within the CPU that are used to hold data temporarily for quick access. - Mention that ARM processors use a RISC architecture, which means a simpler, more efficient set of registers. 2. **Development** (30 minutes) - Explain that ARM has 16 general-purpose registers (R0 to R15) in most 32-bit architectures. - Explain how the Link Register (LR) is used in function calls and how it helps return control to the caller. - Discuss how CPSR holds status flags like the Negative (N), Zero (Z), Carry (C), and Overflow (V) flags. - **Usage**: Explain how conditional execution of instructions depends on these flags - Mention that SPSR is used to store the status register during exceptions and interrupts. - Demonstrate how registers are used with memory access instructions, e.g., LDR R0, [R1] to load data from memory into a register. - Show how the Program Counter (PC) is used for branch instructions to control program flow, e.g., B loop. - Explain that ARM processors operate in different modes (User, Supervisor, IRQ, etc.), and each mode can have its own set of registers (banked registers). - Discuss briefly how some registers (like R13 and R14) have different values in different modes to handle interrupts and exceptions efficiently. **Exercise (5 minutes) –** Give students a set of register-related tasks or questions to identify the purpose of specific registers.

Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Recap the roles of general-purpose and special-purpose registers in ARM and how they support the efficient operation of the processor. 3. Suggested material https://archive.nptel.ac.in/courses/117/106/117106111/ Spend 5 minutes to wrap up and consolidate the learning
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 32	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the classification of ARM instructions and their functionalities. b. Discuss key categories like data processing, load/store, branch, and conditional execution instructions.
Teaching Aids (if any)	a. PPT on ARM Instructions
Teaching Development	1. **Introduction** (5 minutes) - Briefly review the ARM architecture and its reliance on a RISC-based instruction set. - Explain that ARM has a uniform and straightforward instruction set, which contributes to its efficiency. - Categories of Instructions: Introduce the main types of ARM instructions: Data Processing Instructions Load/Store Instructions Branch Instructions Special Instructions **2. Development** (30 minutes) - Explain that ARM uses a Load/Store architecture, where memory operations are separate from data processing. - Describe instructions like `LDR` (Load Register) and `LDRB` (Load Register Byte) to load data from memory into a register. - Discuss instructions like `STR` (Store Register) and `STRB` (Store Register Byte) to store register data into memory. - Introduce indexed addressing modes with examples - Explain the B (Branch) instruction for unconditional jumps. - Conditional Branching: Discuss instructions like BEQ (Branch if Equal), BNE (Branch if Not Equal), and how they depend on condition flags in the CPSR. - Explain how ARM allows most instructions to be conditionally executed based on status flags. - Software Interrupts: Introduce the SWI (Software Interrupt) instruction and explain its role in system calls. - Special Instructions: Briefly mention other special instructions like NOP (No Operation) and BX (Branch and Exchange) for switching between ARM and Thumb modes. **Exercise (5 minutes) –** Recap the main types of ARM instructions and their roles in efficient program execution.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested material https://www.youtube.com/watch?v=UdY5RkkT7bg

	Spend 5 minutes to wrap up and consolidate the learning
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 33	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Explain the purpose and importance of the built-in shift mechanism in ARM processors. b. Discuss the types of shifts supported by ARM and their syntax. c. Analyze the effects of shift operations on binary data through code examples
Teaching Aids (if any)	a. PPT on ARM Built-In Shift Mechanism
Teaching Development	1. **Introduction** (5 minutes) - Start with a quick review of what shifting means in binary operations (moving bits left or right). - Explain that ARM has a built-in shift mechanism integrated into most data processing instructions, enabling efficient data manipulation and reducing the need for separate shift instructions. - Briefly mention practical scenarios where shifts are useful, such as bit manipulation, fast multiplication or division by powers of 2, and data alignment. 2. **Development** (30 minutes) - Explain that ARM instructions can include shift operations without the need for extra instructions. - Discuss how shifts are used to clear, set, or toggle specific bits in a register. - Efficient Code: Highlight how using the built-in shift mechanism reduces instruction count and improves performance. Exercise (5 minutes) – - "Why might preserving the sign bit with ASR be crucial in certain arithmetic operations?" to encourage critical thinking. -
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Recap the types of shift operations and their integration with data processing instructions in ARM. 3. Suggested material https://www.youtube.com/watch?v=aGJm3fuKLhA

	Spend 5 minutes to wrap up and consolidate the learning
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 34	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the role and functionality of branch instructions in ARM. b. Discuss the different types of branch instructions in ARM. c. Learn how ARM uses conditional branches to improve program flow control.
Teaching Aids (if any)	a. PPT on ARM Branch Instructions
Teaching Development	1. **Introduction** (5 minutes) - Begin with a brief introduction to the importance of control flow in programming and how branch instructions help alter the flow of execution in ARM. - Explain that branch instructions are used to jump to different parts of the program based on certain conditions or unconditionally. - Highlight how ARM uses these instructions to implement loops, conditional execution, and subroutine calls. 2. **Development** (30 minutes) - Discuss Types of ARM Branch Instructions - Explain where an unconditional branch is used to jump to a part of the program: - Discuss the various conditional branch instructions, each tied to a status flag: - Walk through how the program loops until the condition is met - Show how conditional branches help implement decision-making. - Discuss how different conditions lead to different branches based on the comparison results. - Show how conditional branches can be used to implement a decision-making structure, such as a simple if-else logic. **Exercise (5 minutes) –** - Ask students to predict the behavior of a program with multiple branch instructions. For example, “What will happen if the Zero flag is set and the BEQ instruction is executed?” -
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested material https://www.youtube.com/watch?v=UdY5RkkT7bg

	Spend 5 minutes to wrap up and consolidate the learning
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 35	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of sequence control in ARM architecture. b. Discuss the role of program counters, branch instructions, and control flow in ARM. c. Learn how ARM handles execution order and sequence control through conditional and unconditional branching.
Teaching Aids (if any)	a. PPT on Sequence Control in ARM Architecture
Teaching Development	1. **Introduction** (5 minutes) - Define the concept of sequence control - Discuss why sequence control is necessary - Explain that the **PC** holds the address of the next instruction to be executed in ARM. 2. **Development** (30 minutes) - Walk through how the program jumps to the subroutine, executes it, and then returns to the main program flow. - Analyze simple code examples to observe sequence control in ARM programs. - Show a simple example where the PC changes during a branch - Explain a loop where the B instruction is used to repeat a set of instructions - Show how conditional branching works - Discuss how the program repeats until R0 equals 5 and then exits the loop. - how a program that demonstrates both unconditional and conditional branches within a loop: **Exercise (5 minutes) –** Present students with a small ARM assembly code snippet that includes branches. Ask them to predict the sequence of execution and determine the values in the registers at different points in the program.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Recap the role of sequence control in ARM architecture using program counters, conditional and unconditional branches, and subroutine calls. 3. Suggested material https://www.youtube.com/watch?v=UQ16Cous_tY Spend 5 minutes to wrap up and consolidate the learning

Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 36	Course Name: Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept and importance of data movement and memory reference instructions in ARM architecture. b. Learn the different types of data movement instructions. c. Understand the role of load and store instructions in ARM.
Teaching Aids (if any)	a. PPT on Data Movement and Memory Reference Instructions in ARM
Teaching Development	- **Introduction** (5 minutes) - Introduction to Data Movement and Memory Reference Instructions - Ask: When you're running an application on your phone, how do you think the data you input gets processed and stored temporarily - **Development** (30 minutes) - Introduce various data movement instructions in ARM processors. - Discuss their syntax and practical uses. - Explain how data is simply copied from one register to another. - Explain how this can be used in applications where inversion of a value is required - Illustrate the stack operation, typically used in function calls and interrupt handling. - Discuss scenarios where saving and restoring multiple registers in one instruction can optimize the code. - Introduce the different memory reference instructions used in ARM architecture. - Explain how these instructions access data in memory. - Discuss how instruction is used to fetch data from memory to registers for processing. - Explain how storing data back to memory is essential in programs that need to modify data. - **Exercise (5 minutes) –** - Provide a short exercise where students write an ARM assembly program to load data from memory, modify it, and store it back. For example, manipulating an array of numbers stored in memory. -

Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested material https://www.youtube.com/watch?v=sp_NVs3SkcA Spend 5 minutes to wrap up and consolidate the learning
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 37	Course Name: Microprocessors Topic: Concept of Core Processor	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Define the concept of a core processor and explain its significance in modern computing. b. Differentiate between single-core, dual-core, and multi-core processors. c. Analyze the benefits and challenges of using multi-core processors in various applications.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - What do you think "core" refers to in a processor? - How does having multiple cores improve the performance of a computer? - **Introduction to Core Processors:** - Explain that a core is the basic computational unit of a CPU, capable of executing instructions. - Discuss the evolution from single-core to multi-core processors as a response to performance demands and limitations of clock speed improvements. - **Development** (30 minutes) - Define core processors as the building blocks of modern CPUs, each capable of independent task execution. - Explain the difference between: - **Single-Core Processors** - Dual-Core and Multi-Core Processors - - **Benefits of Multi-Core Processors** - Parallelism - Improved Performance - Energy Efficiency - **Challenges of Multi-Core Processors** - Software Compatibility - Heat Dissipation - - **Application Areas of Multi-Core Processors** - Gaming, Video Editing, Scientific Computing, Cloud Computing, etc. - Use real-life examples to illustrate how multi-core processors

	enhance performance in these fields. - - Exercise (5 minutes) – - Present a scenario where a computer with different core configurations (single-core, dual-core, quad-core) performs a set of tasks. Ask students to predict and explain which configuration would perform best and why.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. https://youtu.be/VWxx-6qa_Lg 2. Homework Draw a labeled diagram of a multi-core processor, showing how cores connect to shared resources like cache and memory.
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. Why do multi-core processors require efficient task scheduling? 2. In what scenarios might a single-core processor still be preferable? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 38	Course Name: Microprocessors Topic: Basic features of Advanced Microprocessors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Describe the evolution and significance of advanced microprocessors. b. Identify key features of advanced microprocessors (e.g., Pentium, Intel Core series). c. Analyze how these features contribute to improved performance and efficiency.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - How do modern microprocessors differ from earlier versions like the 8086 or 8085? - Why do we need continuous improvements in microprocessor design? - Briefly explain the evolution from basic microprocessors (e.g., 8085, 8086) to advanced ones (e.g., Pentium, Core i3/i5/i7/i9). - Highlight their importance in modern computing for applications such as gaming, AI, and data processing. - **Development** (30 minutes) - **Key Features of Advanced Microprocessors** - **a. High Clock Speed** - Discuss how clock speed (measured in GHz) influences the execution of instructions. - Mention improvements over older processors with much lower clock speeds. - **b. Multiple Cores** - Explain multi-core technology (dual-core, quad-core, octa-core). - Benefits of parallel processing and multitasking. - **c. Cache Memory** - Describe the role of L1, L2, and L3 cache in reducing data access time. - Show a comparison of cache sizes in different microprocessors. - **d. Hyper-Threading and Simultaneous Multithreading (SMT)** - Define and explain how these technologies improve performance by allowing each core to handle multiple threads. - **e. Instruction Set Architecture (ISA)** - Introduction to modern ISAs such as x86-64 and ARM. - Explain the impact of extended instruction sets (e.g., SSE,

	AVX) on performance in specialized tasks like multimedia processing. - **f. Power Efficiency and Thermal Management** - Highlight the importance of low power consumption in mobile and embedded systems. - Introduce features like dynamic frequency scaling (Intel Turbo Boost) and advanced cooling solutions. - **g. Integrated Graphics Processing Unit (GPU)** - Explain how advanced microprocessors often include an integrated GPU for better graphics performance in laptops and desktops. - **h. Advanced Security Features** - Discuss built-in security features such as Intel SGX, AMD SEV, and hardware-based encryption. - **i. Enhanced Pipelining and Branch Prediction** - Describe how these features improve instruction execution efficiency. - - Exercise (5 minutes) – - Ask students to identify which processor would be most suitable for specific applications (e.g., gaming, office work, video editing) and justify their choice
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. https://youtu.be/VWxx-6qa_Lg 2. Homework Research and create a comparison chart of any two advanced microprocessors (e.g., Intel Core i5 vs. AMD Ryzen 5) based on their key features.
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. How do advanced microprocessors manage power efficiency while maintaining high performance? 2. Why is cache memory crucial in modern microprocessor design? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 39	Course Name: Microprocessors Topic: Pentium Processors	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Explain the evolution and significance of the Pentium processor series. b. Identify the key architectural and performance features of Pentium processors. c. Analyze the impact of Pentium processors on modern computing.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - How do you think Pentium processors improved over the earlier 8086 and 80386 processors? - What advancements might have made Pentium a significant step in microprocessor development? - Highlight the launch of the first Pentium processor by Intel in 1993. - Explain that Pentium marked the transition to more powerful, efficient, and widely used processors for personal computers. - **Development** (30 minutes) - Describe the importance of the Pentium brand in Intel's lineup, including its role in advancing computing power. - Mention key Pentium series: Pentium, Pentium Pro, Pentium II, Pentium III, and Pentium 4. - - **Key Features of Pentium Processors** - **a. Superscalar Architecture** - Explain how Pentium introduced superscalar architecture, allowing it to execute more than one instruction per clock cycle. - **b. Pipelining** - Discuss the use of a 5-stage instruction pipeline for faster processing compared to previous microprocessors. - **c. Dual Instruction Pipelines** - Describe how Pentium features two instruction pipelines (U and V pipelines) for parallel instruction execution. - **d. Improved Floating-Point Unit (FPU)** - Highlight enhancements in handling complex arithmetic operations, which improved performance in multimedia and scientific applications. - **e. MMX Technology** - Introduce MMX (Multi-Media Extensions), which enhanced performance in multimedia, image, and audio processing.

	- **f. L1 Cache and Branch Prediction** - Explain how the built-in L1 cache (16 KB in early Pentium models) and dynamic branch prediction improved instruction execution efficiency. - **g. Pentium 4 Specific Features** - Discuss Hyper-Threading Technology and high clock speeds (up to 3 GHz) introduced in Pentium 4. - - Exercise (5 minutes) – - Provide a set of scenarios where specific features of Pentium (e.g., dual pipelines, MMX technology) would benefit tasks like video playback, gaming, or office work. - Ask students to match the feature to the task and explain its relevance.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. https://youtu.be/VWxx-6qa_Lg 2. Homework Research and write a short essay on the differences between Pentium and the Intel Core series, focusing on architectural improvements.
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. Why was superscalar architecture a game-changer in Pentium processors? 2. How did MMX technology impact multimedia applications? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 40	Course Name: Microprocessors Topic: I3	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Explain the significance of Intel Core i3 processors in the Core series lineup. b. Identify the key architectural and performance features of Intel Core i3. c. Analyze the applications and limitations of Intel Core i3 processors in real-world scenarios.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - How do you think Intel Core i3 differs from earlier Pentium processors? - Why might users prefer a Core i3 processor for certain tasks over more advanced CPUs like Core i5 or i7? - Briefly introduce the Intel Core series, emphasizing its segmentation into i3, i5, i7, and later i9. - Explain that Core i3 is designed for budget-friendly systems offering balanced performance for everyday tasks. - **Development** (30 minutes) - Explain that Core i3 processors are entry-level CPUs in Intel's Core lineup, suitable for light to moderate computing tasks like web browsing, document editing, and media playback. - Highlight that i3 processors are available in both desktop and mobile variants. - **Key Features of Intel Core i3 Processors** - **a. Dual-Core and Quad-Core Configurations** - Early models featured dual cores, while newer generations include quad cores for improved multitasking. - **b. Hyper-Threading Technology** - Explain that Hyper-Threading allows each core to handle two threads, effectively doubling the number of threads the processor can manage simultaneously. - **c. Integrated Graphics** - Highlight the presence of Intel HD or UHD Graphics, enabling decent performance for casual gaming and video playback without a dedicated GPU. - **d. Cache Memory** - Discuss the use of L3 cache (typically 3–8 MB), which improves data access speed and overall processor efficiency. - **e. Power Efficiency** - Explain how Core i3 processors balance performance and

	power consumption, making them ideal for laptops and budget desktops. - **f. Turbo Boost (In Some Models)** - Mention that some newer Core i3 models support Intel Turbo Boost Technology for dynamic overclocking under heavy workloads. - **g. Enhanced Instruction Set** - Introduce support for advanced instruction sets such as SSE4.1, SSE4.2, and AVX2, which improve performance in multimedia and computational tasks. - - Exercise (5 minutes) – - Ask students to identify key improvements and explain how these changes impact performance.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. https://youtu.be/VWxx-6qa_Lg 2. Homework Research and compare an Intel Core i3 processor with a Core i5 processor. Highlight key differences and their impact on performance.
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. How does Hyper-Threading improve multitasking in Core i3 processors? 2. Why might a Core i3 processor be preferred for a budget-friendly laptop? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 41	Course Name: Microprocessors Topic: I5	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the role of Intel Core i5 processors in the Core series lineup. b. Identify the key architectural and performance features of Intel Core i5. c. Evaluate the advantages of Core i5 in various computing scenarios.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - Briefly introduce the Intel Core lineup, emphasizing that Core i5 offers a mid-range balance between affordability and high performance. - Explain that Core i5 is suitable for a wide range of tasks, from general computing to gaming and content creation. - **Development** (30 minutes) - Explain that Core i5 processors cater to users who need more power than Core i3 but do not require the advanced capabilities of Core i7 or i9. - Highlight their popularity in both desktop and laptop markets. - **Key Features of Intel Core i5 Processors** - **a. Quad-Core and Higher Configurations** - Early Core i5 models featured 4 cores, while later generations introduced 6 or even 8 cores in desktop variants. - **b. Turbo Boost Technology** - Explain how Intel Turbo Boost dynamically increases the clock speed of cores under heavy workloads for better performance. - **c. Hyper-Threading (Select Models)** - Mention that some higher-end Core i5 models support Hyper-Threading, enabling two threads per core for improved multitasking. - **d. Integrated Graphics** - Discuss the inclusion of Intel UHD Graphics, allowing casual gaming and video playback without a dedicated GPU. - **e. Larger Cache Size** - Highlight the larger cache size (6–12 MB L3 cache), which improves data retrieval speed and overall efficiency. - **f. Enhanced Power Efficiency** - Explain how Core i5 processors balance high performance with power-saving features, particularly in mobile variants. - **g. Enhanced Instruction Sets**

	- Support for AVX, AVX2, and SSE4 instruction sets, boosting performance in tasks like video encoding, 3D rendering, and scientific computations. - - Exercise (5 minutes) – - Provide a table comparing specifications of two Core i5 processors from different generations. Ask students to identify key improvements and discuss how these changes impact real-world performance.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. https://youtu.be/VWxx-6qa_Lg 2. Homework Compare an Intel Core i5 processor with a Core i3 processor in terms of core count, clock speed, and features. Present findings in a table format.
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. Why is Turbo Boost important for performance-intensive tasks? 2. How does a larger cache size improve processor performance? Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 42	Course Name: Microprocessors Topic: I7	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the role of Intel Core i7 processors in the Core series lineup. b. Identify the key architectural and performance features of Intel Core i7. c. Evaluate the applications of Core i7 processors in high-performance computing.
Teaching Aids (if any)	a. ICT Usage
Teaching Development	- **Introduction** (5 minutes) - How do you think Core i7 differs from Core i5 in terms of performance and use cases? - What kind of tasks might require an upgrade to Core i7? - Introduce the Core i7 as a high-performance CPU aimed at demanding users, such as gamers, professionals, and content creators. - Emphasize its role in providing superior multitasking and computational power. - **Development** (30 minutes) - Explain that Core i7 is designed for performance-intensive applications, offering more cores, higher clock speeds, and advanced features. - Highlight its use in both desktop and laptop configurations. - **Key Features of Intel Core i7 Processors** - **a. Higher Core and Thread Counts** - Typically features 4 to 8 cores in earlier generations and up to 16 cores in newer models, with Hyper-Threading enabling up to 32 threads in modern processors. - **b. Turbo Boost Technology** - Explain how Intel Turbo Boost allows dynamic overclocking, improving performance during peak workloads. - **c. Hyper-Threading Across All Models** - Every Core i7 processor supports Hyper-Threading, enabling better multitasking and faster execution of parallel tasks. - **d. Integrated Graphics** - Discuss the inclusion of high-performance Intel Iris or UHD Graphics, sufficient for light gaming and multimedia tasks. - **e. Large Cache Size** - Highlight the increased cache size (8–25 MB L3 cache) for faster data retrieval and improved efficiency.

	- **f. Overclocking Support (Select Models)** - Some Core i7 models (unlocked versions, denoted by 'K' or 'KF') support manual overclocking for enthusiasts. - **g. Power Efficiency and Thermal Management** - Explain the power optimization features, particularly in mobile variants, to balance performance and battery life. - **h. Enhanced Instruction Sets** - Support for advanced instruction sets (AVX2, AVX-512, SSE4, etc.), improving performance in specialized tasks like machine learning, 3D rendering, and video editing. - - - Exercise (5 minutes) – - Provide a table comparing specifications of two Core i7 processors from different generations. Ask students to identify key improvements and discuss how these changes impact real-world performance.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. https://youtu.be/VWxx-6qa_Lg 2. Homework Compare an Intel Core i7 processor with a Core i3, i5 processor in terms of core count, clock speed, and features. Present findings in a table format.
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 1. Why is Hyper-Threading particularly useful in Core i7 processors? 2. How does a larger cache size contribute to the performance of Core i7? Spend 5 minutes to evaluate student assimilation of the lesson contents