CLUSTER UNIVERSITY SRINAGAR

SYLLABUS (FYUP UNDER NEP - 2020)

Offered by Department of Information Technology

Semester 5th (Major Course – CT2)

Title: Microprocessor Systems

Course Code: UGICT22J502	**Max. Marks: 150**
Credits: 6 (Theory: 4, Practical: 2)	**Theory External: 80; Min Marks: 32**
Contact Hrs: 120 (Theory: 60, Practical: 60)	**Theory Internal (Continuous Assessment): 20 Marks, Min Marks: 08**
	Practical Experimental Basis= 30, Min. Marks: 12
	Practical Experimental (Continuous assessment) = 20, Min. Marks: 08

Course Objectives

1. To introduce students to the basic concepts of microprocessors, microcomputers, and embedded systems, providing a foundational understanding necessary for advanced study in computer architecture and embedded systems.
2. To educate students on the Von Neumann architecture and the detailed workings of the 8086 microprocessor, including its architecture, functional units, and addressing modes.
3. To equip students with the skills needed to write and debug assembly language programs, focusing on 8086 and ARM processors, understanding the instructions and tools involved.

Course Outcomes

1. Students will be able to explain the core concepts of microprocessors, microcomputers, and embedded systems, and understand their roles in modern computing.
2. Students will demonstrate an understanding of the Von Neumann architecture, explaining its components and how it underpins modern computer systems.
3. Students will be able to describe the architecture of the 8086 microprocessors, including its block diagram, functional units, and register organization.
4. Students will gain a foundational understanding of assembly language programming, including the syntax, instructions, and tools required for the assembly process.
5. Students will gain a comprehensive understanding of the ARM architecture, focusing on register organization, data types, memory organization, and addressing modes.
6. Students will demonstrate proficiency in using the basic ARM instruction set and writing assembly language programs for ARM processors.

UNIT 1 **(15 Hrs.)**

Basic concepts of microprocessors, microcomputers, and embedded systems. Von Neumann architecture. Introduction to 8086 architecture. Basic block diagram and functional units. Register organization and addressing modes. Microcomputer System Organization: Memory organization and addressing. Pin Diagram. Basic I/O devices and their interfacing.

UNIT II **(15 Hrs.)**

Assembly Language Programming: Introduction to Assembly Language Programming, Basic Assembly Language Instructions and Syntax Assembly process and tools.

8086 Instruction Sets: Data transfer, arithmetic, logical, and control flow instructions.

UNIT III **(15 Hrs.)**

ARM Processor Fundamentals: Overview of ARM Architecture, Register Organization and Data Types, Memory Organization and Addressing Modes.

ARM Instruction Set: Basic ARM Instructions and Assembly Language, ARM Development Tools: Using Assemblers, Compilers, and Debuggers.

Differences and Similarities between 8086, and ARM Architectures.

UNIT IV **(15 Hrs.)**

Microprocessor Applications: Embedded systems design concepts. Case studies of microprocessor-based systems (8086 and ARM). Trends in Microprocessor Technology: Emerging microprocessor architectures and technologies

Industrial Visit:

Objective:

The primary objective of the industrial visit is to provide students with practical insights into the microprocessors and its applications, and the latest advancements in microprocessor technology. This visit aims to bridge the gap between theoretical knowledge and real-world industry practices.

Recommended Books:

1. Microcomputer Systems: The 8086/8088 Family Architecture, Programming, and Interfacing by Barry B. Brey
2. The 8086/8088 Primer: An Introduction to System Design by Triebel, Wagner, and Hyde: Offers a practical approach to 8086 system design.
3. ARM System Developer's Guide by Andrew N. Sloss, Dominic Symes, and Chris Wright: A standard reference for ARM architecture and system design.

LIST OF PRACTICALS (2 CREDITS - 60 HOURS)

Write Assembly Programs for 8086 and ARM microprocessors using Microprocessor Kits/Simulators (EMU8086/TASM for 8086 and QEMU/GNU Arm Embedded Toolchain for ARM).

1. Add two 8-bit numbers.
2. Subtract two 16-bit numbers.
3. Multiply two 8-bit numbers using repeated addition.
4. Divide a 16-bit number by an 8-bit number using repeated subtraction.
5. Find the maximum of two numbers.
6. Find the minimum of two numbers.
7. Swap the contents of two registers.
8. Reverse the order of bits in a byte.
9. Convert a packed BCD number to binary.
10. Convert a binary number to packed BCD.
11. Load a value into a register and store it in memory.
12. Load a block of data from memory to a register array.
13. Store a block of data from a register array to memory.
14. Move a string of characters from one memory location to another.
15. Find the length of a null-terminated string.
16. Compare two strings for equality.
17. Implement a simple loop to print numbers from 1 to 10.
18. Calculate the factorial of a number.
19. Find the sum of even numbers from 1 to n.
20. Check if a number is prime.
21. Implement a simple sorting algorithm (e.g., bubble sort).
22. Implement a linear search algorithm.
23. Read a character from the keyboard and display it on the screen.
24. Display a message on the screen.
25. Read a number from the keyboard and store it in memory.
26. Write a number from memory to the screen.
27. Implement a simple calculator.
28. Convert numbers between different number systems (e.g., binary, decimal, and hexadecimal).
29. Simulate a traffic light controller.
30. Create a simple game (e.g., Number Guessing Game).