Course Code	Course Title	L	T	P	C
BACSE103	Computation Structures	3	0	2	4
Pre-requisite		Syllabus Version			
		1.0			

Course Objectives

1. To Understand and design of digital logic systems ranging from simple combinational circuits to sequential designs.
2. To impart the knowledge of data representation and to understand the implementation of arithmetic algorithms in a typical computer.
3. To introduce and explore the fundamental concepts of microcontrollers and techniques of parallel processing.

Course Outcomes

1. To design and optimize minimal combinational circuits.
2. To analyze and design sequential circuits utilizing various types of flip-flops.
3. To analyze different instruction formats, addressing modes and validate efficient algorithms for fixed-point arithmetic operations.
4. To understand the concept of memory organization and I/O fundamentals
5. To understand the microcontroller architectural designs and parallel computing systems.

Module:1	Digital Systems and Combinational Circuits	7 hours

Introduction to Digital systems: Number system, Data representation: Fixed and Floating point, Logic Gates, Simplification of Boolean function using K-Map, Design of combinational circuit: Adder, Carry look-ahead adder, Encoder and Decoder, Multiplexer and Demultiplexer.

Module:2	Sequential Circuits	9 hours

Flip Flops, Sequential circuit design and Analysis: Finite State Machines-Mealy and Moore, Registers, and Counters: Synchronous and Asynchronous, Timing Sequences.

Module:3	Computer Architectures and Arithmetic Algorithms	10 hours

Introduction to Von Neumann machine and Harvard architecture, Fixed point arithmetic operations: Multiplication (Booths, Modified Booths), Division (restoring and non-restoring). Instruction set architecture, Instruction formats, Instruction types, addressing modes, Instruction cycle, Single cycle Datapath design, Multicycle Datapath design.

Module:4	Memory and I/O Peripherals	8 hours

Memory Organization, Memory Hierarchy, Memory Design, Cache Memory-Mapping Policies, Replacement Algorithms, I/O Fundamentals- programmed I/O and Interrupt driven I/O, Direct Memory Access, Shared Memory Architectures.

Module:5	Advanced Architecture	9 hours

Introduction to Microcontroller, ARM Architecture Versions– Instruction Set – Stacks and Subroutines. Parallel processing Techniques: Instruction Level parallelism, and Thread Level Parallelism.

Module:6	Contemporary Issues	2 hours
	Total Lecture hours:	45 hours

Text Books

1.	M. Morris Mano, Digital Logic and Computer Design, Fifth Edition, Pearson India Education Publishers, 2019
2.	Patterson, David A., and John L. Hennessy, Computer organization and design ARM edition: the hardware software interface, First Edition, Morgan kaufmann, 2016.
Reference Books	
1.	William Stallings, Computer Architecture and Organization-Designing for Performance, Eleventh edition, Pearson Education series, 2022.
2.	Ananth Grama, Anshul Gupta, George Karypis and Vipin Kumar, Introduction to Parallel Computing, 2nd Edition, Pearson, 2008.
Mode of Evaluation: Quiz, Assignment, CAT and FAT	
List of Experiments (Indicative)	
1	Combinational Circuit Design - Full Adder, Full Subtractor, Binary Parallel Adder. Consider there are two counter circuit employed in a food mall one at the front door near the staircase and another beside the elevator, to count the number of customers. At time 't' the number of customers entered through staircase and elevator are counted by the two counter circuits as 2 and 3 respectively. Design a suitable combinational circuit to find the sum of these two-counter output at time 't'.
2	Combinational Circuit Design - Decoder, Encoder. Build a decoder with three input lines but with only six output lines. If the value of the input corresponds to 6 or 7, then all output lines should be asserted to signal an error.
3	Combinational Circuit Design - MUX, DMUX. You are required by your manager to design a 4-bit prime number checker – you want to output 1 if the binary number is primer else 0 (note 0 and 1 are not prime numbers). a. Show the truth table for the primer number checker. b. Design the circuit using single 4x1 MUX and a minimal number of extra AND, OR or NOT gates if needed (i.e., no NAND, NOR, XOR or XNOR, etc.). Show your work including any simplifications done. You must show the logic diagram with all the pins of the mux labeled properly. State any assumptions that you make.
4	Sequential Circuit Design - Flipflop, Shift Register. Use a SIPO shift register to display a binary counter on 8 LEDs. Each second, the counter increments by 1, and the value is sent serially to the register.
5	Sequential Circuit Design - Synchronous and Asynchronous Counters. A bank queuing system has a capacity of 5 customers which serves on first come first served basis. A display unit is used to display the number of customers waiting in the queue. Whenever a customer leaves the queue, the count is reduced by one and the count is increased by one if a customer joins a queue. Two sensors (control signals) are used to sense customers leaving

	and joining the queue respectively. Design a circuit that displays the number of customers waiting in the queue in binary format using LEDs. Binary '1' is represented by LED glow and '0' otherwise.
6	Introduction to Microcontroller -To design and implement a simple calculator
7	Interfacing with GPIO - To interface with GPIO pins by controlling an LED and detecting a button press using a microcontroller.
8	Interfacing Bluetooth module with microcontroller - Connect a Bluetooth module (e.g., HC-05) to the microcontroller and send/receive simple text messages using serial window.
9	Introduction to parallel programming using OpenMP - Write a OpenMP program to find the largest of three numbers and smallest of three numbers using explicit thread identification.
10	OpenMP work sharing constructs - Write a program to perform matrix operations and compare the sequential and parallel execution time.
11	OpenMP synchronization constructs- Demonstrate an example by using Single, Master, Barrier, and critical constructs of OpenMP.
12	Using Performance Analyzer tools find the execution time and memory usage of a parallel program.
	Total hours: 30 hours

Mode of Evaluation:			
Recommended by Board of Studies			
Approved by Academic Council	**No.**	**Date**	