Digital Systems and Computer Architecture					
Pre-requisite: None		L	T	P	C
Total hours: 40		3	1	0	4
Course Content					Hrs.
Unit 1	Boolean Algebra and Logic Gates: Basic definition, Axiomatic Definition, Basic theorem and Properties of Boolean Algebra, Minterms and Maxterms, Logic Operations, Digital logic gates IC, digital logic families.				6
Unit 2	Simplification of Boolean functions: Different types map method, product of sum simplification, NAND or NOR implementation, Don't care condition, Tabulation method, Adder, subtractor, Code Conversion, Universal Gates.				6
Unit 3	Sequential Logic: Flip-flops, Triggering of Flip-flops, Analysis of clocked sequential circuits, State reduction and Assignment, Flip-flop excitation, Design of counters, Design with state equations.				6
Unit 4	Basic Computer Organization and Design: Instruction codes, Computer registers, Computer instructions, Addressing modes, Timing and Control, Instruction cycle, Memory-Reference Instructions, Input-output and interrupt. Design of Basic computer, Design of Accumulator Unit.				8
Unit 5	Programming the Basic Computer: Introduction, Machine Language, Assembly Language, the Assembler. Programming Arithmetic and logic operations. Subroutines. I-O Programming.				7
Unit 6	Central Processing Unit: Introduction, GeneralRegister Organization, Stack Organization, Instruction format. Addressing Modes, Data transfer and manipulation. Program Control, Reduced Instruction Set Computer (RISC). Pipelining, Parallel Processing.				7

Course Outcomes

- Understand the fundamental principles of digital systems and computer architecture, including sequential logic, Boolean algebra, and basic of computer organization and design.
- Analyze and design digital systems using combinational and sequential logic components, and understand their role in modern computer architecture.
- Explore advanced topics in computer architecture, such as pipelining, addressing modes, and parallel processing, and their impact on system performance and efficiency.

• Design and implement computer architecture components such as processors, memory subsystems, and I/O programming.	
Text Books	
1.	Mano, Morris M., Computer system architecture. Prentice-Hall, Inc., 1993.
2.	Mano, Morris M., Digital logic and computer design, PHI, 2004
Reference Books	
1.	Tanenbaum A.S., Structured Computer Organization, 2016.
2.	Stallings, Computer Organization, Computer Organization & Architecture 7e, 2008.
3.	Hayes J., Computer Architecture and Organization, 2007.
4.	Hwang K., Advanced Computer Architecture, 2003.

Theory of Computations					
Prerequisite: None		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs.
Unit 1	Basic foundation: Review of Set Theory, Automata Theory, Alphabet, Power of Alphabet, Kleen Closure, Positive Closure, String, Empty String, Concatenation, Language. Finite Automata (FA): Introduction, Deterministic Finite Automata (DFA) -Formal Definition, Simpler Notations (State Transition Diagram, Transition Table), Language of A DFA. Nondeterministic Finite Automata (NFA)- Definition of NFA, Language of an NFA, Equivalence of Deterministic and Nondeterministic Finite Automata, Applications of Finite Automata, Finite Automata with Epsilon Transitions, Eliminating Epsilon Transitions, Minimization of Deterministic Finite Automata, Finite Automata with Output (Moore and Mealy Machines) and Inter Conversion.				10
Unit 2	Regular Expressions (RE): Introduction, Identities of Regular Expressions, Finite Automata and Regular Expressions- Converting from DFA’s to Regular Expressions, Converting Regular Expressions to Automata, Minimization of Finite Automata, Applications of Regular Expressions. Regular grammars: Chomsky Classification of Languages, Regular Grammars and FA, FA for Regular Grammar, Regular Grammar for FA. Proving Languages to be Non-Regular -Pumping Lemma, Applications, Closure Properties of Regular Languages.				8
Unit 3	Context free grammar (cfg): Derivation Trees, Sentential Forms, Rightmost and Leftmost Derivations of Strings. Ambiguity in CFGs, Minimization of CFGs, Normal Forms (CNF, GNF), Pumping Lemma for CFLs.				8
Unit 4	Pushdown automata theory: Push Down Automata, Deterministic and Nondeterministic PDA, PDA And Languages, Construction of PDA, Acceptance of CFL, Acceptance by Final State and Acceptance by Empty Stack and its Equivalence, Equivalence of CFG and PDA.				8

	Turing machines (tm): formal Definition and Behaviour, Languages of a TM, TM as Accepters, TM as a Computer of Integer Functions, TM with Storage in its State, TM as Subroutine, Minsky's Theorem, Types of TMs, Multitrack, Multitape, Nondeterministic TM, Encoding of TM, Computability and Acceptability.	
Unit 5	Recursive and Recursively Enumerable Languages (REL): Properties of Recursive and Recursively Enumerable Languages. Undecidability and undecidable problems: Post's Correspondence Problem (PCP), Universal Turing Machine, The Halting Problem, Undecidable Problems about TMs. Context Sensitive Language and Linear Bounded Automata (LBA), Chomsky Hierarchy, Decidability.	8

Course Outcomes:

- Develop a solid understanding of the theoretical foundations of computation, including formal languages and automata theory.
- Analyze and classify computational problems using formal models such as finite automata, regular expressions, and context-free grammars.
- Explore advanced topics in the theory of computation, such as Recursive and Recursively Enumerable Languages and Chomsky Hierarchy.
- Apply theoretical concepts to solve computational problems and analyze the computational complexity of algorithms, and understand the limits of computation.

Text Books

1.	Hopcroft J.E., Motwani R., Ullman J.D., Introduction to Automata Theory, Languages and Computation, Pearson Education, India, 2007
2.	Cohen, D. I. A. Introduction to Computer Theory. 2nd ed. New York: John Wiley & Sons, 1997.

Reference books

1.	Martin J.C., Introduction to Languages and the Theory of Computation. 4th ed., McGraw-Hill, 2010
2.	Lewis H. R., & Papadimitriou, C. H. Elements of the Theory of Computation. 2nd ed., Prentice Hall, 1998

3.	Linz P. An Introduction to Formal Languages and Automata. Jones & Barlett Learning, 2016.
4.	Sipser, M. Introduction to the Theory of Computation. 3rd ed., Cengage Learning, 2013.