Syllabus of all courses offered by CSE

CS11001/CS11002 PROGRAMMING AND DATA STRUCTURES **L-T-P: 3-1-0, Credit: 4**

Introduction to digital computers; introduction to programming - variables, assignments; expressions; input/output; conditionals and branching; iteration; functions; recursion; arrays; introduction to pointers; structures; introduction to data-procedure encapsulation; dynamic allocation; linked structures; introduction to data structures - stacks and queues; time and space requirements.

(A programming language like C/C++ may be used as a basis language. The same language must be used for the laboratory).

CS19001/CS19002 PDS Laboratory
L-T-P: 0-0-3, Credit: 2

Suggested assignments to be conducted on a 3-hour slot. It will be conducted in tandem with the theory course so the topics for problems given in the lab are already initiated in the theory class. The topics taught in the theory course should be appropriately be sequenced for synchronization with the laboratory. A sample sequence of topics and lab classes for the topic are given below:

1. Familiarization of a computer and the environment and execution of sample programs
2. Expression evaluation
3. Conditionals and branching
4. Iteration
5. Functions
6. Recursion
7. Arrays
8. Structures
9. Linked lists
10. Data structures

It is suggested that some problems related to continuous domain problems in engineering and their numerical solutions are given as laboratory assignments. It may be noted that some of basic numerical methods are taught in the Mathematics course.

References

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India.
2. E. Balaguruswamy, Programming in ANSI C, Tata McGraw-Hill.
3. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill.
4. Seymour Lipschutz, Data Structures, Schaum's Outlines Series, Tata McGraw-Hill.

5. Ellis Horowitz, Satraj Sahni and Susan Anderson-Freed, Fundamentals of Data Structures in C, W. H. Freeman and Company.
6. R. G. Dromey, How to Solve it by Computer, Prentice-Hall of India.
7. On-line notes

CS20006 SOFTWARE ENGINEERING **L-T-P: 3-0-0, Credit: 3**

Introduction, software life-cycle models, software requirements specification, formal requirements specification and verification - axiomatic and algebraic specifications, function-oriented software design, object-oriented design, UML, design patterns, user interface design, coding and unit testing, integration and systems testing, debugging techniques, software quality - SEI CMM and ISO-9001. Software reliability and fault-tolerance, software project planning, monitoring, and control, software maintenance, computer-aided software engineering (CASE), software reuse, component-based software development, extreme programming.

CS29006 Software Engineering Laboratory
L-T-P: 0-0-3, Credit: 2

Development of requirements specification, function oriented design using SA/SD, object-oriented design using UML, test case design, implementation using Java and testing. Use of appropriate CASE tools and other tools such as configuration management tools, program analysis tools in the software life cycle.

References

1. Rajib Mall, Fundamentals of Software Engineering, Prentice Hall India.
2. Pankaj Jalote, An integrated approach to Software Engineering, Springer/Narosa.
3. Roger S. Pressman, Software Engineering: A practitioner's approach, McGraw Hill.
4. Ian Sommerville, Software Engineering, Addison-Wesley.

CS21001 DISCRETE STRUCTURES **L-T-P: 3-1-0, Credit: 4**

Propositional logic: Syntax, semantics, valid, satisfiable and unsatisfiable formulas, encoding and examining the validity of some logical arguments.

Proof techniques: forward proof, proof by contradiction, contrapositive proofs, proof of necessity and sufficiency.

Sets, relations and functions: Operations on sets, relations and functions, binary relations, partial ordering relations, equivalence relations, principles of mathematical induction.

Size of a set: Finite and infinite sets, countable and uncountable sets, Cantor's diagonal argument and the power set theorem, Schroeder-Bernstein theorem.

Introduction to counting: Basic counting techniques - inclusion and exclusion, pigeon-hole principle, permutation, combination, summations. Introduction to recurrence relation and generating function.

Algebraic structures and morphisms: Algebraic structures with one binary operation - semigroups, monoids and groups, congruence relation and quotient structures. Free and cyclic monoids and groups, permutation groups, substructures, normal subgroups. Algebraic structures with two binary operations - rings, integral domains and fields. Boolean algebra and Boolean ring.

Introduction to graphs: Graphs and their basic properties - degree, path, cycle, subgraphs, isomorphism, Eulerian and Hamiltonian walks, graph coloring, planar graphs, trees.

References

1. Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata McGraw-Hill.
2. C. L. Liu, Elements of Discrete Mathematics, Tata McGraw-Hill.
3. Norman L. Biggs, Discrete Mathematics, Oxford University Press.
4. Kenneth Bogart, Clifford Stein and Robert L. Drysdale, Discrete Mathematics for Computer Science, Key College Publishing.
5. Thomas Koshy, Discrete Mathematics with Applications, Elsevier.
6. Ralph P. Grimaldi, Discrete and Combinatorial Mathematics, Pearson Education, Asia.

CS21002 SWITCHING CIRCUITS AND LOGIC DESIGN L-T-P: 3-1-0, Credit: 4

Switching Circuits: Logic families: TTL, nMOS, CMOS, dynamic CMOS and pass transistor logic (PTL) circuits, inverters and other logic gates, area, power and delay characteristics, concepts of fan-in, fan-out and noise margin.

Switching theory: Boolean algebra, logic gates, and switching functions, truth tables and switching expressions, minimization of completely and incompletely specified switching functions, Karnaugh map and Quine-McCluskey method, multiple output minimization, representation and manipulation of functions using BDD's, two-level and multi-level logic circuit synthesis.

Combinational logic circuits: Realization of Boolean functions using NAND/NOR gates, Decoders, multiplexers. logic design using ROMs, PLAs and FPGAs. Case studies.

Sequential circuits: Clocks, flip-flops, latches, counters and shift registers, finite-state machine model, synthesis of synchronous sequential circuits, minimization and state assignment, asynchronous sequential circuit synthesis.

ASM charts: Representation of sequential circuits using ASM charts, synthesis of output and next state functions, data path control path partition-based design.

CS29002 Switching Laboratory
L-T-P: 0-0-3, Credit: 2

Pulse Circuits: Bistable, astable and monostable MVs and Schmitt Triggers using transistors, OP Amps and 555 timers.

TTL and CMOS Gates: Study the characteristics of TTL and MOS gates.

Combinational logic circuits: Design and implementation of combinational circuits such as ALU and 7-segment LED display driver.

Sequential Circuits: Design of sequence generators and detectors, counters, design of ASMs such as, traffic light controllers, lift controllers, etc.

References

1. H. Taub and D. Schilling, Digital Integrated Electronics, McGraw-Hill .
2. Z. Kohavi, Switching and Finite Automata Theory, Tata McGraw-Hill.
3. Randy H. Katz and Gaetano Borriello, Contemporary Logic Design, Prentice Hall of India.
4. Giovanni De Micheli, Synthesis and Optimization of Digital Circuits, Tata McGraw-Hill.

CS21003 ALGORITHMS I **L-T-P: 3-1-0, Credit: 4**

Asymptotic notations and their significance, introduction to RAM model of computation, complexity analysis of algorithms, worst case and average case.

Basic introduction to algorithmic paradigms like divide and conquer, recursion, greedy, etc.

Searching: binary search trees, balanced binary search trees, AVL trees and red-black trees, B-trees, skip lists, hashing. Priority queues, heaps, Interval trees, tries.

Order statistics.

Sorting: comparison based sorting - quick sort, heap sort, merge sort: worst and average case analysis. Decision tree model and (worst case) lower bound on sorting. Sorting in linear time - radix sort, bucket sort, counting sort, etc.

String matching

Graph Algorithms: BFS, DFS, connected components, topological sort, minimum spanning trees, shortest paths - single source and all pairs.

CS29003 Algorithms Laboratory
L-T-P: 0-0-3, Credit: 2

The laboratory component will emphasize two areas:

Implementation of algorithms covered in class: This will involve running the algorithms under varying input sets and measuring running times, use of different data structures for the same algorithm (wherever applicable) to see its effect on time and space, comparison of different algorithms for the same problem etc.

Design of Algorithms: This will involve design and implementation of algorithms for problems not covered in class but related to topics covered in class.

The exact set of algorithms to design and implement is to be decided by the instructor. In addition, there will be at least one significantly large design project involving some real world application. An efficient design of the project should require the use of multiple data structures and a combination of different algorithms/techniques.

References

1. T. H. Cormen, C. L. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, MIT Press.
2. J. Kleinberg and E. Tardos, Algorithm Design, Addison-Wesley.
3. Harry R. Lewis and Larry Denenberg, Data Structures and Their Algorithms, Harper Collins.
4. A. Gibbons, Algorithmic Graph Theory, Cambridge University Press.
5. Michael T. Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, John Wiley.
6. R. Sedgewick, Algorithms in C (Parts 1-5), Addison Wesley.
7. M. H. Alsuwaiyel, Algorithm Design Techniques and Analysis, World Scientific.
8. Gilles Brassard and Paul Bratley, Algorithmics : theory and practice, Prentice-Hall.
9. Udi Manber, Introduction to Algorithms: A Creative Approach, Addison-Wesley.
10. Sara Baase and Allen Van Gelder, Computer Algorithms: Introduction to Design and Analysis, Addison-Wesley.

CS21004 FORMAL LANGUAGES AND AUTOMATA THEORY **L-T-P: 3-1-0, Credit: 4**

Introduction: Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages.

Regular languages and finite automata: Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata.

Context-free languages and pushdown automata: Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs.

Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG.

Turing machines: The basic model for Turing machines (TM), Turing-recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators.

Undecidability: Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice's theorem, undecidable problems about languages.

References

1. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, Pearson Education Asia.
2. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education Asia.
3. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.
4. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.
5. John Martin, Introduction to Languages and The Theory of Computation, Tata McGraw Hill.

CS30002 OPERATING SYSTEMS **L-T-P: 3-0-0, Credit: 3**

Evolution of Operating Systems, Structural overview, Concept of process and Process synchronization, Process Management and Scheduling, Hardware requirements: protection, context switching, privileged mode; Threads and their Management; Tools and Constructs for Concurrency, Detection and Prevention of deadlocks, Dynamic Resource Allocation, Design of IO systems, File Management, Memory Management: paging, virtual memory management, Distributed and Multiprocessor Systems, Case Studies.

CS39002 Operating Systems Laboratory
L-T-P: 0-0-3, Credit: 2

Familiarization with UNIX system calls for process management and inter-process communication; Experiments on process scheduling and other operating system tasks through simulation/implementation under a simulated environment (like Nachos).

References

1. Avi Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts, Wiley Asia Student Edition.
2. William Stallings, Operating Systems: Internals and Design Principles, Prentice Hall of India.
3. D. M. Dhamdhere, Operating Systems: A Concept-Based Approach, Tata McGraw-Hill.
4. Charles Crowley, Operating System: A Design-oriented Approach, Irwin Publishing.
5. Gary J. Nutt, Operating Systems: A Modern Perspective, Addison-Wesley.
6. Maurice Bach, Design of the Unix Operating Systems, Prentice-Hall of India.
7. Daniel P. Bovet, Marco Cesati, Understanding the Linux Kernel, O'Reilly and Associates.

CS30003 COMPILERS **L-T-P: 3-0-0, Credit: 3**

The aim is to learn how to design and implement a compiler and also to study the underlying theories. The main emphasis is for the imperative languages.

Introduction: Phases of compilation and overview.

Lexical Analysis (scanner): Regular language, finite automata, regular expression, from regular expression to finite automata, scanner generator (lex,flex).

Syntax Analysis (Parser): Context-free language and grammar, push-down automata, LL(1) grammar and top-down parsing, operator grammar, LR(O), SLR(1), LR(1), LALR(1) grammars and bottom-up parsing, ambiguity and LR parsing, LALR(1) parser generator (yacc,bison)

Semantic Analysis: Attribute grammar, syntax directed definition, evaluation and flow of attribute in a syntax tree.

Symbol Table: Its structure, symbol attributes and management.

Run-time environment: Procedure activation, parameter passing, value return, memory allocation, and scope.

Intermediate Code Generation: Translation of different language features, different types of intermediate forms.

Code Improvement (optimization): Analysis: control-flow, data-flow dependence etc.; Code improvement local optimization, global optimization, loop optimization, peep-hole optimization etc. Architecture dependent code improvement: instruction scheduling (for pipeline), loop optimization (for cache memory) etc.

Register allocation and target code generation

Advanced topics: Type systems, data abstraction, compilation of object oriented features and non-imperative programming languages.

CS39003 Compilers Laboratory
L-T-P: 0-0-3, Credit: 2

The aim is to write a compiler for a small language.

Familiarity with compiled codes (assembly language) of RISC and CISC machines, writing a scanner, writing predictive parser for a small language, small experiment with scanner (lex/flex) and parser (yacc/byson) generator (such as translation of regular expression to NFA or the construction or parse tree), writing scanner-parse specification for a small language, translation of the language to an intermediate form (e.g. three-address code), generation of target code (in assembly language). Code improvement (optional).

References

1. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, Compilers: Principles, Techniques and Tools, Addison-Wesley.
2. Michael L. Scott, Programming Language Pragmatics, Elsevier.
3. Andrew W. Appel, Modern Compiler Implementation in C/Java, Cambridge University Press.
4. Keith D. Cooper and Linda Torczon, Engineering a Compiler, Elsevier.
5. Allen I. Holob, Compiler Design in C, Prentice-Hall.
6. Steven S. Muchnik, Advanced Compiler Design and Implementation, Elsevier.
7. Randy Allen and Ken Kennedy, Optimizing Compilers for Modern Architectures, Elsevier.

CS30701 FOUNDATIONS OF COMPUTING **L-T-P: 3-0-0, Credit: 3**

Logic, sets, relations and functions, induction, iteration and recursion, graphs. Algebraic structures, combinatorics. Grammars and languages, automata, Turing machines, undecidability. Algorithms and their correctness, complexity, intractability.

References

1. Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata McGraw-Hill.
2. Ralph P. Grimaldi, Discrete and Combinatorial Mathematics, Pearson Education, Asia.
3. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.

CS31001 COMPUTER ORGANIZATION AND ARCHITECTURE **L-T-P: 4-0-0, Credit: 4**

Basic functional blocks of a computer: CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU - registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. Case study - instruction sets of some common CPUs.

Data representation: signed number representation, fixed and floating point representations, character representation. Computer arithmetic - integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication - shift-and-add, Booth multiplier, carry save multiplier, etc. Division - non-restoring and restoring techniques, floating point arithmetic.

CPU control unit design: hardwired and micro-programmed design approaches, Case study - design of a simple hypothetical CPU.

Memory system design: semiconductor memory technologies, memory organization.

Peripheral devices and their characteristics: Input-output subsystems, I/O transfers - program controlled, interrupt driven and DMA, privileged and non-privileged instructions, software interrupts and exceptions. Programs and processes - role of interrupts in process state transitions.

Performance enhancement techniques

Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards.

Memory organization: Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs block size, mapping functions, replacement algorithms, write policy.

CS39001 Computer Organization Laboratory
L-T-P: 0-0-6, Credit: 4

1. Familiarization with assembly language programming.
2. Synthesis/design of simple data paths and controllers, processor design.
3. Interfacing - DAC, ADC, keyboard-display modules, etc.

Development kits as well as Microprocessors/PCs may be used for the laboratory, along with design/simulation tools as and when necessary.

References

1. David A. Patterson and John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, Elsevier.
2. Carl Hamachar, Zvonco Vranesic and Safwat Zaky, Computer Organization, McGraw Hill.
3. John P. Hayes, Computer Architecture and Organization, McGraw Hill.
4. William Stallings, Computer Organization and Architecture: Designing for Performance, Pearson Education.

5. Vincent P. Heuring and Harry F. Jordan, Computer Systems Design and Architecture, Pearson Education.

CS31004 THEORY OF COMPUTATION **L-T-P: 3-1-0, Credit: 4**

Computability theory: Review of Turing machines, some other computing models and formalisms, their equivalence with Turing machines, undecidability, Post correspondence problem, Turing computability, primitive recursive functions, Cantor and Goedel numbering, Ackermann function, mu-recursive functions, recursiveness of Ackermann and Turing computable functions, lambda calculus, term rewriting, oracle machines and the arithmetic hierarchy.

Complexity theory: Time- and space-bounded Turing machines, reduction and complete problems, oracle machines and the polynomial hierarchy, randomized computation, parallel computation.

Logic: First-order predicate calculus - syntax, semantics, validity and satisfiability, decision problems in logic, quantified Boolean formulas and their relation with the polynomial hierarchy.

References

1. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.
2. Fred C. Hennie. Introduction to Computability. Addison-Wesley.
3. Bernard M. Moret, The Theory of Computation, Pearson Education Asia.
4. Christos H. Papadimitriou, Computational Complexity, Addison-Wesley Longman.
5. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.
6. John Martin, Introduction to Languages and The Theory of Computation, Tata McGraw Hill.
7. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education Asia.

CS31005 ALGORITHMS II **L-T-P: 3-1-0, Credit: 4**

Models of computation: RAM model and its logarithmic cost.

Formal introduction to algorithmic paradigms: divide and conquer, recursion, dynamic programming, greedy, branch and bound, etc.

Advanced data structures: Fibonacci heap, union-find, splay trees.

Amortized complexity analysis

Randomized algorithms: Randomized algorithms to be introduced a bit early, i.e. before NP-completeness to highlight randomization as an algorithmic technique.

Application areas

1. *Geometric algorithms:* convex hulls, nearest neighbor, Voronoi diagram, etc.
2. *Algebraic and number-theoretic algorithms:* FFT, primality testing, etc.
3. *Graph algorithms:* network flows, matching, etc.
4. *Optimization techniques:* linear programming

Reducibility between problems and NP-completeness: discussion of different NP-complete problems like satisfiability, clique, vertex cover, independent set, Hamiltonian cycle, TSP, knapsack, set cover, bin packing, etc.

Backtracking, branch and bound

Approximation algorithms: Constant ratio approximation algorithms.

Miscellaneous: Introduction to external memory algorithms, parallel algorithms.

References

1. Rajeev Motwani and Prabhakar Raghavan, Randomized Algorithms, Cambridge University Press.
2. Allan Borodin, Ran El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press.
3. Nancy Lynch, Distributed Algorithms, Morgan Kaufmann.
4. Robert Endre Tarjan, Data Structures and Network Algorithms, SIAM.
5. L. Grotchel, L. Lovasz, and A. Schrijver, Geometric algorithms and Combinatorial Optimization, Springer.
6. M. Kearns and U. Vazirani, An Introduction to Computational Learning Theory. MIT Press.
7. N. Alon and J. H. Spencer, The Probabilistic Method, John Wiley.
8. Vijay Vazirani, Approximation Algorithms, Springer.
9. Fan Chung, Spectral Graph Theory, American Mathematical Society.

CS31702 COMPUTER ARCHITECTURE AND OPERATING SYSTEMS **L-T-P: 4-0-0, Credit: 4**

Architecture: Basic organization, fetch-decode-execute cycle, data path and control path, instruction set architecture, I/O subsystems, interrupts, memory hierarchy, overview of pipelined architecture.

Operating systems: An overview, process management, user and supervisor modes, process synchronization, semaphores, memory management, virtual memory, file systems, I/O systems.

Issues in multiprocessing environments.

References

1. David A. Patterson and John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, Elsevier.
2. Carl Hamachar, Zvonco Vranesic and Safwat Zaky, Computer Organization, McGraw-Hill.
3. John P. Hayes, Computer Architecture and Organization, McGraw-Hill.
4. Avi Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts, Wiley Asia Student Edition.
5. William Stallings, Operating Systems: Internals and Design Principles, Prentice Hall of India.

CS40001 COMPUTER NETWORKS **L-T-P: 3-0-0, Credit: 3**

Introduction to networks and layered architecture. Data communication concepts, transmission media and topology, multiplexing. Circuit switching and packet switching, data link layer, layer 2 switches and ATM switches, SONET/SDH. Medium access control. CSMA CD, TDMA, FDMA, CDMA. Network layer and addressing, IP version 4 and 6. Routing algorithms. Transmission layer, TCP and UDP. Congestion control techniques. WAN, ATM. Internetworking. Wireless communications. Network management and security.

CS49001 Networks Laboratory
L-T-P: 0-0-3, Credit: 2

Simulation experiments for protocol performance, configuring, testing and measuring network devices and parameters/policies; network management experiments; Exercises in network programming.

References

1. William Stallings, Data and Computer Communication, Prentice Hall of India.
2. Behrouz A. Forouzan, Data Communication and Networking, McGraw-Hill.
3. Andrew S. Tanenbaum, Computer Networks, Prentice Hall.
4. Douglas Comer, Internetworking with TCP/IP, Volume 1, Prentice Hall of India.
5. W. Richard Stevens, TCP/IP Illustrated, Volume 1, Addison-Wesley.

CS40104 PARALLEL ALGORITHMS **L-T-P: 3-0-0, Credit: 3**

Parallel Models (SIMD, MIMD, PRAMs, Interconnection Networks); Performance Measures (Time, Processors, Space, Work); Interconnection Architectures (Linear Array, Meshes, Trees, Mesh of Trees, Hypercubes, Butterfly Networks, Cube Connected Cycles, Benes Networks); Techniques (Balanced Trees, Pointer Jumping, Divide and Conquer, Partitioning, Pipelining, Systolic Computation, Accelerated Cascading, Prefix Computation, List Ranking, Euler Tour, Tree Contraction); Sorting, Searching, Merging; Matrix Operations; Graph Algorithms (Connected Components, Spanning Trees, Shortest Paths); Complexity (Lower bounds, NC Class and P-Completeness).

CS40105 SYMBOLIC LOGIC AND AUTOMATED **L-T-P: 3-0-0, Credit:**

REASONING **3**

Introduction and motivation: Role of logic in Computer Science, problem representation.

Basic notions: language, models, interpretations, validity, proof, decision problems in logic. decidability.

Propositional logic: Syntax, semantics, proof systems, Validity, satisfiability and unsatisfiability, soundness and completeness.

Machanization: truth tables, normal forms, semantic tableau, resolution, proof by contradiction, example.

First order predicate logic theory: Quantifiers, first order models, validity and satisfiability, semantic tableaux.

Normal forms, skolemization: Elimination of quantifiers, unification, resolution and various resolution strategies, equality axioms and para-modulation. Horn formulas and programs. Prolog as a restricted resolution-based theorem prover. Undecidability and incompleteness in logic, compactness Theorem.

Other topics: Introduction to Modal Logic, Temporal Logic and other logics for concurrency. Some exposure to theorem proving systems such as Prolog, PVS, SPIN, etc.

References

1. Michael Huth and Mark Ryan, Logic in Computer Science: Modelling and Reasoning about Systems, Cambridge University Press.
2. Arindama Singh, Logics for Computer Science, Prentice Hall of India.
3. C. L. Chang and R. C. T. Lee, Symbolic Logic and Mechanical Theorem Proving, Academic Press.
4. M. Ben-Ari, Mathematical Logic for Computer Science, Springer.
5. E. Mendelson, Introduction to Mathematical Logic, Chapman and Hall.

CS40106 PRINCIPLES OF PRORAMMING LANGUAGES L-T-P: 3-0-0, Credit: 3

Theory: The aim is to study and appreciate different types of languages and the underlying mathematical theories. This may help to design and also to appreciate new language features.

Introduction: Overview of different programming paradigms e.g. imperative, object oriented, functional, logic and concurrent programming.

Syntax and semantics of programming languages: A quick overview of syntax specification and semiformal semantic specification using attribute grammar.

Imperative and OO Languages: Names, their scope, life and binding. Control-flow, control abstraction; in subprogram and exception handling. Primitive and constructed data types, data abstraction, inheritance, type checking and polymorphism.

Functional Languages: Typed-calculus, higher order functions and types, evaluation strategies, type checking, implementation, case study.

Logic Programming Languages: Computing with relation, first-order logic, SLD-resolution, unification, sequencing of control, negation, implementation, case study.

Concurrency: Communication and synchronization, shared memory and message passing, safety and liveness properties, multithreaded program.

Formal Semantics: Operational, denotational and axiomatic semantics of toy languages, languages with higher order constructs and types, recursive type, subtype, semantics of nondeterminism and concurrency.

References

1. Glynn Winskel, A Formal Semantics of Programming Languages: An Introduction, MIT Press.
2. John C. Mitchell, Foundations for Programming Languages, MIT Press.
3. Benjamin C. Pierce, Types and Programming Languages, MIT Press.
4. Daniel P. Friedman, Mitchell Wand and Christopher T. Haynes, Essentials of Programming Languages, Prentice Hall of India.
5. Ravi Sethi, Programming Languages: Concepts and Constructs, Addison-Wesley.
6. H. P. Barendregt, The Lambda Calculus: Its Syntax and Semantics, North-Holland.

CS40202 FAULT TOLERANT SYSTEMS **L-T-P: 3-0-0, Credit: 3**

Introduction to redundancy theory; decision theory in redundant systems.

Hardware fault tolerance, redundancy techniques, detection of faults, replication and compression techniques, self-repairing techniques, concentrated and distributed voters, models of fault tolerant computing systems, case study. Fault diagnosis of digital circuits and systems: fault modeling, test generation, design for testability, signature analysis, built in self test. Testing of embedded systems. Software fault tolerance: fault tolerance versus fault intolerance, errors and their management strategies, software defense, protective redundancy. Fault recovery techniques. Coding theory: application to fault tolerant system design.

CS40203 OBJECT ORIENTED SYSTEM DESIGN **L-T-P: 3-0-0, Credit: 3**

This course will cover object-oriented approach to modeling, problem solving, requirement analysis, system design, system implementation, database design, system engineering and software engineering.

Fundamental concepts of object oriented programming: Introduction to the principles of object-oriented programming (classes, objects, messages, encapsulation, inheritance, polymorphism, exception handling, and object-oriented containers).

Object design implementation in a programming language, e.g., C++ or Java.

Object oriented analysis, modeling and design: UML may be introduced. Use cases, use case driven analysis.

Structural modeling classes, relationships, interfaces, class diagrams, and object diagrams, in UML.

Behavioral/Functional modeling use case diagrams, sequence diagrams, in UML.

Dynamic modeling: State charts

Architectural modeling

Analysis patterns, Design patterns.

Distributed object model: CORBA and COM / DCOM

Object oriented database systems: Object oriented data model, query languages, storage organization and indexing techniques; object relational databases.

References

1. Bertrand Meyer, Object Oriented Software Construction, Prentice-Hall.
2. Grady Booch, Object Oriented Analysis and Design, Addison-Wesley.
3. Grady Booch, James Rumbaugh and Ivar Jacobson, Unified Modeling Language Guide, Addison-Wesley.
4. Erich Gamma et al., Design Patterns: Elements of Reusable OO Software, Addison-Wesley.
5. Michael L. Scott, Programming Language Pragmatics, Morgan-Kaufmann.
6. Kim Bruce, Foundations of Object Oriented Languages, Prentice-Hall.
7. Benjamin C. Pierce, Types and Programming Languages, Prentice-Hall.
8. Bjarne Stroustrup, The Design and Evolution of C++, Addison-Wesley.
9. Bill Venners, Inside the JAVA 2 Virtual Machine, McGraw Hill.
10. James E. Smith and Ravi Nair, Virtual Machines, Elsevier/Morgan-Kaufmann.
11. Saba Zamir, Handbook of Object Technology, CRC Press.

CS40301 ARTIFICIAL INTELLIGENCE **L-T-P: 3-0-0, Credit: 3**

This course will cover basic ideas and techniques underlying the design of intelligent computer systems. Topics include:

- Introduction to AI and intelligent agents.
- Problem Solving: Solving Problems by Searching, heuristic search techniques, constraint satisfaction problems, stochastic search methods.
- Game Playing: minimax, alpha-beta pruning.
- Knowledge and Reasoning: Building a Knowledge Base: Propositional logic, first order
- Logic, situation calculus. Theorem Proving in First Order Logic.
- Planning, partial order planning.
- Uncertain Knowledge and Reasoning, Probabilities, Bayesian Networks.
- Learning: Overview of different forms of learning, Learning Decision Trees, Neural Networks
- Introduction to Natural Language Processing.

References

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Prentice-Hall.
2. Nils J. Nilsson, Artificial Intelligence: A New Sythesis, Morgan-Kaufmann.

CS40306 ELECTRONIC DESIGN AUTOMATION L-T-P: 3-0-0, Credit: 3

Two-level and multi-level logic optimization of combinational circuits, state assignment of finite state machines. Technology mapping for FPGAs. Techniques for partitioning, floor planning, placement and routing. Architectural models, scheduling, allocation and binding for high-level synthesis.

Hardware-software codesign. Test generation, fault simulation, built-in self test, test structures. Verilog and VHDL.

References

1. R. H. Katz, Contemporary Logic Design, Addison-Wesley.
2. M. J. S. Smith, Application-Specific Integrated Circuits, Addison-Wesley.
3. W. Wolf, Modern VLSI Design: Systems on Silicon, Pearson Education.
4. J. Bhasker, Verilog VHDL Synthesis: A Practical Primer, B S Publications.
5. D. D. Gajski, N. D. Dutt, A. C. Wu and A. Y. Yin, High-level Synthesis: Introduction to Chip and System Design, Kluwer Academic Publishers.
6. M. Abramovici, M. A. Breuer and A. D. Friedman, Digital Systems Testing and Testable Design, IEEE Press.
7. P. Bardell, W. H. McAnney and J. Savir, Built-in Test for VLSI: Pseudo-random Techniques, John Wiley and Sons.
8. M. Sarrafzadeh and C. K. Wong, An Introduction to Physical Design, McGraw Hill.
9. N. A. Sherwani, Algorithms for VLSI Physical Design Automation, Kluwer Academic Publishers.

10. S. M. Sait and H. Youssef, VLSI Physical Design Automation: Theory and Practice, World Scientific.

CS40305 IMAGE PROCESSING **L-T-P: 3-0-0, Credit: 3**

Digital Image Fundamentals: A simple image model, Sampling and Quantization, Imaging Geometry, Digital Geometry, Image Acquisition Systems, Different types of digital images.

Bilevel Image Processing: Basic concepts of digital distances, distance transform, medial axis transform, component labeling, thinning, morphological processing, extension to grey scale morphology.

Binarization and Segmentation of Grey level images: Histogram of grey level images, Optimal thresholding using Bayesian classification, multilevel thresholding, Segmentation of grey level images, Water shade algorithm for segmenting grey level image.

Detection of edges and lines in 2D images: First order and second order edge operators, multi-scale edge detection, Canny's edge detection algorithm, Hough transform for detecting lines and curves, edge linking.

Images Enhancement: Point processing, Spatial Filtering, Frequency domain filtering, multi-spectral image enhancement, image restoration.

Color Image Processing: Color Representation, Laws of color matching, chromaticity diagram, color enhancement, color image segmentation, color edge detection, color demosaicing.

Image Registration and depth estimation: Registration Algorithms, Setreo Imaging, Computation of disparity map.

Image compression: Lossy and lossless compression schemes, prediction based compression schemes, vector quantization, sub-band encoding schemes, JPEG compression standard, Fractal compression scheme, Wavelet compression scheme.

References

1. Gonzalez and Woods, Digital Image Processing, Prentice-Hall.

CS40308 INTERNET TECHNOLOGY **L-T-P: 3-0-0, Credit: 3**

Evolution of Internet, TCP/IP: addressing and routing. Internet applications: FTP, Telnet, Email, Chat. World Wide Web: HTTP protocol. Designing web pages: HTML, forms, CGI scripts and clickable maps, JAVA applets, JAVAscript, JAVA servlets, Perl. DHTML, XML. E-Commerce and security issues including symmetric and asymmetric key, encryption and digital signature, authentication. Emerging trends, Internet telephony, virtual reality over the web, etc. Intranet and extranet, firewall design issues.

CS40310 MODELING AND SIMULATION **L-T-P: 3-0-0, Credit: 3**

System models and role of simulation. Entities, Attributes, States and Activities. Types of systems: Deterministic, Stochastic, Continuous and Discrete systems. Steps in simulation studies. Statistical tools and techniques: generation of pseudorandom numbers, random variate generation for uniform, Poisson and normal distributions, sampling and estimation, maximum likelihood estimation, confidence intervals and hypothesis testing, stochastic processes and Markov models. Discrete event simulation languages. Simulation of inventory and queuing systems: single and multiserver queues, network of queues. Modeling and performance evaluation of computers and computer communication networks. Workload characterization. Continuous system simulation languages, growth and decay models, system dynamics diagrams. Biological and Sociological system simulation. Verification and validation of simulation models: input/output validation, sensitivity analysis, performance measures and their estimation. Case studies.

CS41101 APPLIED GRAPH THEORY **L-T-P: 3-1-0, Credit: 4**

Fundamental concepts (basic definitions, operations, properties, proof styles); Trees (properties, distances and centroids, spanning trees, enumeration); Matchings (bipartite graphs, general graphs, weighted matching); Connectivity (vertex and edge connectivity, cuts, blocks, k-connected graphs, network flows); Traversibility (Eulerian tours, Hamiltonian cycles); Coloring (vertex and edge coloring, chromatic number, chordal graphs); Planarity (duality, Euler's formula, characterization, 4-color theorem); Advanced topics (perfect graphs, matroids, Ramsay theory, extremal graphs, random graphs); Applications.

References

1. Douglas B. West, Introduction to Graph Theory, Prentice Hall of India.
2. Narsingh Deo, Graph Theory with Applications to Engineering and Computer Science. Prentice-Hall.
3. Frank Harary, Graph Theory, Narosa.
4. R. Ahuja, T. Magnanti, and J. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice-Hall.

CS41102 COMPUTATIONAL GEOMETRY **L-T-P: 3-1-0, Credit: 4**

Introduction: historical perspective, geometric preliminaries. Convex hulls algorithms in 2d and 3d, lower bounds. Triangulations: polygon triangulations, representations, point-set triangulations. Voronoi diagrams: algorithms, closest pair problems. Delaunay triangulations: algorithms (divide-and-conquer, flip, incremental), duality of Voronoi diagrams, properties (min-max angle). Geometric searching: point-location, 2d linear programming with prune and search. Visibility: algorithms for weak and strong visibility, visibility with reflections, art-gallery problems. Arrangements of lines: 2d arrangements, zone theorem, many-faces complexity, algorithms. Sweep techniques: plane sweep for segment intersections, Fortune's sweep for Voronoi diagrams, topological sweep for line arrangements. Combinatorial geometry: Ham-sandwich cuts, Helly's theorems, k-sets. Rectilinear geometry: intersection and union of

rectangles, rectangle searching. Robust geometric computing. Applications of computational geometry.

References

1. Mark de Berg, Otfried Schwarzkopf, Marc van Kreveld and Mark Overmars, Computational Geometry: Algorithms and Applications, Springer.
2. F. P. Preparata and Michael I. Shamos, Computational Geometry: An Introduction, Springer.
3. Joseph O' Rourke, Computational Geometry in C, Cambridge University Press.
4. Lecture Notes by David Mount.

CS41103 COMPUTATIONAL COMPLEXITY L-T-P: 3-1-0, Credit: 4

Computational Models (machine models, logic); Problems, computability, Algorithms, Resources, and Complexity; Turing machines (time and space bounds, nondeterminism); Logic (Boolean logic, circuits, first and second order logic); Complexity classes (hierarchy theorem, reachability, P, NP, Co-NP); Reduction and completeness; Randomized computation; Approximability; Cryptography and protocols; Parallel Computation; Polynomial Hierarchy; Logarithmic space; Polynomial space; Exponential time and space.

References

1. Christos H. Papadimitriou, Computational Complexity, Addison-Wesley Longman.
2. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.
3. John E. Hopcroft and Jeffrey D. Ullman, Introduction to Automata, Languages and Computation, Addison-Wesley, 1979.
4. J. Balcazar, J. Diaz, and J. Gabarro, Structural Complexity, Volumes I and II, Springer.

CS41201 ADVANCED COMPUTER ARCHITECTURE L-T-P: 3-1-0, Credit: 4

Overview of von Neumann architecture: Instruction set architecture; The Arithmetic and Logic Unit, The Control Unit, Memory and I/O devices and their interfacing to the CPU; Measuring and reporting performance; CISC and RISC processors.

Pipelining: Basic concepts of pipelining, data hazards, control hazards, and structural hazards; Techniques for overcoming or reducing the effects of various hazards.

Hierarchical Memory Technology: Inclusion, Coherence and locality properties; Cache memory organizations, Techniques for reducing cache misses; Virtual memory organization, mapping and management techniques, memory replacement policies.

Instruction-level parallelism: Concepts of instruction-level parallelism (ILP), Techniques for increasing ILP; Superscalar, super-pipelined and VLIW processor architectures; Vector and symbolic processors; Case studies of contemporary microprocessors

Multiprocessor Architecture: Taxonomy of parallel architectures; Centralized shared-memory architecture, synchronization, memory consistency, interconnection networks; Distributed shared-memory architecture, Cluster computers.

Non von Neumann Architectures: Data flow Computers, Reduction computer architectures, Systolic Architectures.

References

1. John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann.
2. John Paul Shen and Mikko H. Lipasti, Modern Processor Design: Fundamentals of Superscalar Processors, Tata McGraw-Hill.
3. M. J. Flynn, Computer Architecture: Pipelined and Parallel Processor Design, Narosa Publishing House.
4. Kai Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill.

CS41205 VLSI SYSTEM DESIGN **L-T-P: 3-1-0, Credit: 4**

Introduction to CMOS VLSI Design; nMOS and CMOS transistor structures and process technologies. Operation of MOS transistor as a switch. Design and analysis of nMOS and CMOS inverters, common gates, latches and flip-flops. Fabrication of MOS transistors; stick diagrams, design rules and layout. Circuit characterization and performance estimation of MOS circuits. CMOS circuit and logic design, BiCMOS logic gates. Dynamic MOS structures, Registers, counters and memory realizations using MOS logic. Design structuring; Regular structure circuits, PLAs and FSMs, system timing and clocking issues, scaling. CMOS subsystem design. Low power circuits and systems. System case studies. Design automation of VLSI Systems: basic concepts. Deep Sub-micron Technologies: Some Design Issues.

References

1. N. H. E. Weste and K. Eshraghian, Principles of CMOS VLSI Design : A Systems Perspective, Pearson Education.
2. W. Wolf, Modern VLSI Design: Systems on Silicon, Pearson Education.
3. J. Rabaey, A. Chandrakasan and B. Nikolic, Digital Integrated Circuits: A Design Perspective, Prentice Hall of India.
4. M. Sarafzadeh and C. K. Wong, An Introduction to VLSI Physical Design, MCGraw-Hill.
5. D. D. Gajaski, N. D. Dutt, A. C.-H. Wu and S. Y.-L. Lin, High-Level Synthesis: Introduction to Chip and System Design, Kluwer Academic Publishers.

CS41312 MULTIMEDIA APPLICATIONS **L-T-P: 3-1-0, Credit: 3**

Introduction to Multimedia System: Architecture and components, Multimedia distributed processing model, Synchronization, Orchestration and Quality of Service (QOS) architecture.

Audio and Speech: Data acquisition, Sampling and Quantization, Human Speech production mechanism, Digital model of speech production, Analysis and synthesis, Psycho-acoustics, low bit rate speech compression, MPEG audio compression.

Images and Video: Image acquisition and representation, Composite video signal NTSC, PAL and SECAM video standards, Bilevel image compression standards: ITU (formerly CCITT) Group III and IV standards, JPEG image compression standards, MPEG video compression standards.

Multimedia Communication: Fundamentals of data communication and networking, Bandwidth requirements of different media, Real time constraints: Audio latency, Video data rate, multimedia over LAN and WAN, Multimedia conferencing.

Hypermedia presentation: Authoring and Publishing, Linear and non-linear presentation, Structuring Information, Different approaches of authoring hypermedia documents, Hyper-media data models and standards.

Multimedia Information Systems: Operating system support for continuous media applications: limitations is usual OS, New OS support, Media stream protocol, file system support for continuous media, data models for multimedia and hypermedia information, content based retrieval of unstructured data.

References

1. Ralf Steinmetz and Klara Nahrstedt, Multimedia Systems, Springer.
2. J. D. Gibson, Multimedia Communications: Directions and Innovations, Springer.
3. K. Sayood, Introduction to Data Compression, Morgan-Kaufmann.
4. A. Puri and T. Chen, Multimedia Systems, Standards, and Networks, Marcel Dekker.
5. Iain E.G. Richardson, H.264 and MPEG-4 Video Compression, John Wiley.
6. Borivoje Furht, Handbook of Multimedia Computing, CRC Press.

CS43204 MICROPROCESSORS AND MICROCONTROLLERS **L-T-P: 3-0-3, Credit: 5**

Historical background; organization and architectural features of microprocessor and microcontrollers; the instruction set: instruction format, addressing modes; assembly language programming of 8085 and 8051; interfacing of memory devices; data transfer techniques and I/O ports; interfacing of keyboard and display devices; programmable interrupt and DMA controllers; interfacing of sensors, transducers, actuators, A/D and D/A Converters, analog signal conditioning circuits, data acquisition systems; standard interfaces - RS232, USB; development

aids and troubleshooting techniques; application examples; advanced microprocessors and microcontrollers.

Laboratory: Assembly and machine language programming, signal generators, interfacing basic I/O devices like keypad, LED display, usage of timers and USART peripherals, multi-port device access, stepper motor movement control, DC motor speed control, bootstrap programming and interfacing various peripherals for embedded applications; building a complete microcontroller-based system.

References

1. R. Gaonkar, Microprocessor Architecture, Programming, and Applications with the 8085, Penram.
2. A. Pal, Microprocessors: Principles and Applications, Tata McGraw-Hill.
3. K. J. Ayala, The 8051 Microcontroller Architecture, Programming and Applications, Penram.
4. Mazidi and Mazidi, Microcontroller and Embedded Systems, Pearson Education.
5. R. Kapadia, 8051 Microcontroller and Embedded Systems, Jaico.

CS43302 COMPUTER GRAPHICS L-T-P: 3-0-3, Credit: 5

Graphics hardware and display devices; graphics primitives: drawing lines and curves; 2d and 3d transformations; segments and their applications; generating curves, surfaces and volumes in 3d, wire-frame models, Bezier and spline curves and surfaces; geometric modeling: elementary geometric algorithms for polygons, boundary representations, constructive solid geometry, spatial data structures; hidden surface and line elimination; rendering: shading, light models, realistic image synthesis techniques, textures and image-based rendering; video games and computer animation.

Laboratory: Programming for generating lines, curves and rendered surfaces. Interactive graphics programming: modeling and updating objects in an object hierarchy, video games, computer animation and realistic image synthesis.

Programming environments: OpenGL (or equivalent), Java graphics environments, X windows (or equivalents).

CS43303 DIGITAL SYSTEM TESTING AND TESTABLE DESIGN L-T-P: 3-0-3, Credit: 5

Physical faults and their modeling. Fault equivalence and dominance; fault collapsing. Fault simulation: parallel, deductive and concurrent techniques; critical path tracing. Test generation for combinational circuits: Boolean difference, D-algorithm, Podem, etc. Exhaustive, random and weighted test pattern generation; aliasing and its effect on fault coverage. PLA testing: cross-point fault model, test generation, easily testable designs. Memory testing: permanent, intermittent and pattern-sensitive faults; test generation. Delay faults and hazards; test generation techniques. Test pattern generation for sequential circuits: time-frame expansion method, ad-hoc

and structures techniques, scan path and LSSD, boundary scan. Built-in self-test techniques. Testing issues in embedded core based systems.

References

1. N. K. Jha and S. Gupta, Testing of Digital Systems, Cambridge University Press.
2. M. L. Bushnell and V. D. Agrawal, Essentials of Electronic Testing, Kluwer Academic Publishers.
3. M. Abramovici, M. A. Breuer and A. D. Friedman, Digital Systems Testing and Testable Design, Wiley-IEEE Press.
4. P. H. Bardell, W. H. McAnney and J. Savir, Built-in Test for VLSI: Pseudorandom Techniques, Wiley Interscience.
5. P. K. Lala, Fault Tolerant and Fault Testable Hardware Design, Prentice-Hall.
6. A. Krstic and K-T Cheng, Delay Fault Testing for VLSI Circuits, Kluwer Academic Publishers.
7. A. Osseiran (Ed.), Analog and Mixed Signal Boundary Scan, Kluwer Academic Publishers.

CS43304 DATABASE MANAGEMENT SYSTEMS L-T-P: 3-0-3, Credit: 5

Database system architecture Data Abstraction, Data Independence, Data Definition and Data Manipulation Languages.

Data models Entity-relationship, network, relational and object oriented data models, integrity constraints and data manipulation operations.

Relational query languages Relational algebra, tuple and domain relational calculus, SQL and QBE.

Relational database design Domain and data dependency, Armstrong's axioms, normal forms, dependency preservation, lossless design.

Query processing and optimization Evaluation of relational algebra expressions, query equivalence, join strategies, query optimization algorithms.

Storage strategies Indices, B-trees, hashing.

Transaction processing Recovery and concurrency control, locking and timestamp based schedulers, multiversion and optimistic Concurrency Control schemes.

Advanced topics Object-oriented and object relational databases, logical databases, web databases, distributed databases, data warehousing and data mining.

Laboratory

Database schema design, database creation, SQL programming and report generation using a commercial RDBMS like ORACLE/SYBASE/DB2/SQL-Server/INFORMIX. Students are to be exposed to front end development tools, ODBC and CORBA calls from application Programs, internet based access to databases and database administration.

References

1. Abraham Silberschatz, Henry Korth, and S. Sudarshan, Database System Concepts, McGraw-Hill.
2. Raghu Ramakrishnan, Database Management Systems, WCB/McGraw-Hill.
3. Bipin Desai, An Introduction to Database Systems, Galgotia.
4. J. D. Ullman, Principles of Database Systems, Galgotia.
5. R. Elmasri and S. Navathe, Fundamentals of Database Systems8, Addison-Wesley.
6. Serge Abiteboul, Richard Hull and Victor Vianu, Foundations of Databases. Addison-Wesley.

CS60001 ADVANCES IN ALGORITHMS **L-T-P: 4-0-0, Credit: 4**

Algorithmic paradigms: Dynamic Programming, Greedy, Branch-and-bound; Asymptotic complexity, Amortized analysis; Graph Algorithms: Shortest paths, Flow networks; NP-completeness; Approximation algorithms; Randomized algorithms; Linear programming; Special topics: Geometric algorithms (range searching, convex hulls, segment intersections, closest pairs), Numerical algorithms (integer, matrix and polynomial multiplication, FFT, extended Euclid's algorithm, modular exponentiation, primality testing, cryptographic computations), Internet algorithms (text pattern matching, tries, information retrieval, data compression, Web caching).

CS60002 DISTRIBUTED SYSTEMS **L-T-P: 4-0-0, Credit: 4**

Basic concepts. Models of computation: shared memory and message passing systems, synchronous and asynchronous systems. Logical time and event ordering. Global state and snapshot algorithms, mutual exclusion, clock synchronization, leader election, deadlock detection, termination detection, spanning tree construction. Programming models: remote procedure calls, distributed shared memory. Fault tolerance and recovery: basic concepts, fault models, agreement problems and its applications, commit protocols, voting protocols, checkpointing and recovery, reliable communication. Security and Authentication: basic concepts, Kerberos. Resource sharing and load balancing. Special topics: distributed objects, distributed databases, directory services, web services.

References

1. Mukesh Singhal and Niranjan Shivaratri, Advanced Concepts in Operating Systems, McGraw-Hill.
2. Nancy Lynch, Distributed Algorithms, Morgan Kaufmann.
3. Andrew S. Tanenbaum, Distributed Operating Systems, ACM Press.

4. Jie Wu, Distributed Systems, CRC Press.
5. Hagit Attiya, Jennifer Welch, Distributed Computing: Fundamentals, Simulations and Advanced Topics, McGraw-Hill.
6. Sape Mullender (ed.), Distributed Systems, Addison-Wesley.

CS60003 HIGH PERFORMANCE COMPUTER ARCHITECTURE **L-T-P: 4-0-0, Credit: 4**

Introduction: review of basic computer architecture, quantitative techniques in computer design, measuring and reporting performance. CISC and RISC processors. Pipelining: Basic concepts, instruction and arithmetic pipeline, data hazards, control hazards, and structural hazards, techniques for handling hazards. Exception handling. Pipeline optimization techniques. Compiler techniques for improving performance. Hierarchical memory technology: Inclusion, Coherence and locality properties; Cache memory organizations, Techniques for reducing cache misses; Virtual memory organization, mapping and management techniques, memory replacement policies. Instruction-level parallelism: basic concepts, techniques for increasing ILP, superscalar, super-pipelined and VLIW processor architectures. Array and vector processors. Multiprocessor architecture: taxonomy of parallel architectures. Centralized shared-memory architecture: synchronization, memory consistency, interconnection networks. Distributed shared-memory architecture. Cluster computers. Non von Neumann architectures: data flow computers, reduction computer architectures, systolic architectures.

References

1. John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann.
2. John Paul Shen and Mikko H. Lipasti, Modern Processor Design: Fundamentals of Superscalar Processors, Tata McGraw-Hill.
3. M. J. Flynn, Computer Architecture: Pipelined and Parallel Processor Design, Narosa Publishing House.
4. Kai Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill.

CS60031 LOGICS FOR COMPUTER SCIENCE **L-T-P: 3-1-0, Credit: 4**

Axiomatic Theory: Propositional Calculus, Predicate Calculus, First Order Theories, Peano Arithmetic. Decision Procedures in First Order Logic: Resolution Theorem Provers: some theoretical issues. Modal Logic, Temporal Logic: their applications, Model Checking. Model Theory, Proof Theory. Mu-Calculus, Lambda Calculus, Non-monotonic Reasoning, Intuitionistic First Order Logic, Fuzzy Logic.

References

1. Michael Huth and Mark Ryan, Logic in Computer Science: Modelling and Reasoning about Systems, Cambridge University Press.
2. Arindama Singh, Logics for Computer Science, Prentice Hall of India.

3. C. L. Chang and R. C. T. Lee, Symbolic Logic and Mechanical Theorem Proving, Academic Press.
4. M. Ben-Ari, Mathematical Logic for Computer Science, Springer.
5. E. M. Clarke Jr., Orna Grumberg and D. A. Peled, Model Checking, MIT Press.
6. E. Mendelson, Introduction to Mathematical Logic, Chapman and Hall.

CS60032 DATABASE ENGINEERING L-T-P: 3-0-0, Credit: 3

Relational Databases: Integrity Constraints revisited: Functional, Multi-valued and Join Dependency, Template Algebraic, Inclusion and Generalized Functional Dependency, Chase Algorithms and Synthesis of Relational Schemes. Query Processing and Optimization: Evaluation of Relational Operations, Transformation of Relational Expressions, Indexing and Query Optimization, Limitations of Relational Data Model, Null Values and Partial Information. Deductive Databases: Datalog and Recursion, Evaluation of Datalog program, Recursive queries with negation. Objected Oriented and Object Relational Databases: Modeling Complex Data Semantics, Specialization, Generalization, Aggregation and Association, Objects, Object Identity, Equality and Object Reference, Architecture of Object Oriented and Object Relational Databases. Case Studies: Gemstone, O2, Object Store, SQL3, Oracle xxi, DB2. Parallel and Distributed Databases: Distributed Data Storage: Fragmentation and Replication, Location and Fragment Transparency, Distributed Query Processing and Optimization, Distributed Transaction Modeling and Concurrency Control, Distributed Deadlock, Commit Protocols, Design of Parallel Databases, Parallel Query Evaluation. Advanced Transaction Processing: Nested and Multilevel Transactions, Compensating Transactions and Saga, Long Duration Transactions, Weak Levels of Consistency, Transaction Work Flows, Transaction Processing Monitors. Active Databases: Triggers in SQL, Event Constraint and Action: ECA Rules, Query Processing and Concurrency Control, Compensation and Databases Recovery. Real Time Databases: Temporal Constraints: Soft and Hard Constraints, Transaction Scheduling and Concurrency Control. Image and Multimedia Databases: Modeling and Storage of Image and Multimedia Data, Data Structures - R-tree, k-d tree, Quadtrees, Content Based Retrieval: Color Histograms, Textures etc, Image Features, Spatial and Topological Relationships, Multimedia Data Formats, Video Data Model, Audio and Handwritten Data, Geographic Information Systems (GIS). WEB Databases: Accessing Databases through WEB, WEB Servers, XML Databases, commercial Systems: Oracle xxi, DB2. Data Mining: Knowledge Representation Using Rules, Association and Classification Rules, Sequential Patterns, Algorithms for Rule Accessing.

References

1. Abraham Silberschatz, Henry Korth, and S. Sudarshan, Database System Concepts, McGraw-Hill.
2. Raghu Ramakrishnan, Database Management Systems, WCB/McGraw-Hill.
3. Bipin Desai, An Introduction to Database Systems, Galgotia.
4. J. D. Ullman, Principles of Database Systems, Galgotia.
5. R. Elmasri and S. Navathe, Fundamentals of Database Systems8, Addison-Wesley.

6. Serge Abiteboul, Richard Hull and Victor Vianu, Foundations of Databases. Addison-Wesley.

CS60033 LOGIC PROGRAMMING **L-T-P: 3-0-0, Credit: 3**

Propositional logic, First Order Logic: syntax and semantics, deduction, Herbrand interpretation and resolution methods, Syntax and Semantics of Logic Programs, Inference Rules, Unification and SLD-Resolution, Negation as Failure, Logic programming language PROLOG - a case study. Basic concepts, Recursive programming, Cuts and negation, Non-deterministic programming, Abstract computational model - Warren's Abstract Machine (WAM), Implementation of Prolog on WAM. Introduction to Constraint Logic Programming: Constraint logic programming scheme, Constraint satisfaction, constraint propagation, Constraint Logic Programming over the reals, Constraint Logic Programming over finite domains. Introduction to nonclassical logics. Modal logic. Accessibility. Relation and Kripke possible world semantics. The logic of knowledge and belief, Autoepistemic knowledge, Temporal logic.

CS60034 ADVANCED MICROPROCESSOR BASED SYSTEMS **L-T-P: 3-0-0, Credit: 3**

Introduction: Basics of Von Neumann Architecture and the early Microprocessors, CISC and RISC concepts; Parallelism in Processor Architecture: Pipelining, Super-scalar, Super-pipeline and VLIW Architectures, Low-power Architecture; Built-in Multiprocessing support; Co-processors; Processor Architecture with hierarchical memory organization: Cache memory, Virtual memory; Built-in Multi-user and multitasking support in 16-bit and 32-bit microprocessors, Built-in memory mapping and management support; Evolution of platform architecture; Special-purpose processor Architectures: Signal processing Microprocessors; Communication processors; Case studies with contemporary Microprocessors.

CS60035 SELECTED TOPICS IN ALGORITHMS **L-T-P: 3-0-0, Credit: 3**

The objective of this course is to familiarize students with some contemporary research in the area of algorithm design and analysis. The treatment will be theoretical with emphasis on problem solving and will be primality assignments based.

Models of computation and efficiency: Searching faster than O(log n), sorting faster than O(n log n).

Randomized algorithms in graphs and geometry: The impact of using randomization for designing algorithms that are simpler and often more efficient than the deterministic counterparts for several fundamental problems like MST, mincuts, spanners, convex hulls, triangulations, etc. Typically analysis is often harder than design.

Approximation algorithms: A set of rapidly evolving techniques that lead to provable approximation guarantees for hard optimization problems within polynomial running times. Unlike other communities dealing with the same problems the emphasis here is on provability of general instances and goes hand-in-hand with the "hardness of approximation" theory.

Each of the topics on their own could be easily a full semester course, so depending on the class response, we may pick and choose from the above list of topics.

References

1. Rajeev Motwani and Prabhakar Raghavan, Randomized Algorithms, Cambridge University Press.
2. Kurt Mehlhorn, Data Structures and Algorithms I (Sorting and Searching), Springer.
3. Vijay Vazirani, Approximation Algorithms, Springer.

CS60036 INTELLIGENT SYSTEMS **L-T-P: 3-0-0, Credit: 3**

Data, information and knowledge. Model of an intelligent system. Models of knowledge representations: Representation and reasoning in logic. Semantic representations: semantic networks, frames; Frame/script systems; Conceptual dependency and conceptual graphs. Ontologies. Knowledge based systems: Software architecture of a knowledge-based system, Rule-based programming and production systems, Rule chaining and inference control, Inference: reasoning about knowledge, Temporal reasoning, Inference under uncertainty: Bayesian techniques, Fuzzy reasoning, Case-based reasoning. Intelligent agents, The agent metaphor and attributes of agent-hood, Agent theory and languages, Inter-agent communication, Ontological issues. Alternatives to the symbolic approach: Foundations of connectionist networks; their history. Applications of AI: Example application domains, e.g. Configuration, Diagnosis, Planning, intelligent interfaces, user modeling, practical implications of choosing and applying AI solutions. Knowledge representation and the Web, Semantic Web.

CS60037 EMBEDDED SYSTEMS **L-T-P: 3-1-0, Credit: 4**

Introduction to Embedded Systems - definitions and constraints; hardware and processor requirements; special purpose processors; input-output design and I/O communication protocols; design space exploration for constraint satisfaction; co-design approach; example system design; Formal approach to specification; specification languages; specification refinement and design; design validation; Real Time operating system issues with respect to embedded system applications; time constraints and performance analysis.

References

1. Peter Marwedel, Embedded System Design, Kluwer.
2. Wayne Wolf, Computers as Components: Principles of Embedded Computing Systems Design, Morgan-Kaufmann.
3. Frank Vahid and Tony Givargis, Embedded System Design: A Unified Hardware/Software Introduction, John Wiley.

CS60038 ADVANCES IN OPERATING SYSTEMS DESIGN **L-T-P: 3-0-0, Credit: 3**

Theory and implementation aspects of distributed operating systems. Process synchronization in multiprocessing/multiprogramming systems. Inter-process communication and co-ordination in large distributed systems. Distributed resource management. Fundamentals of real time operating systems. Case studies. Information management in distributed systems: security, integrity and concurrency problems. Fault tolerance issues. OS issues related to the Internet, intranets, pervasive computing, embedded systems, mobile systems and wireless networks. Case studies of contemporary operating systems.

CS60039 TESTING AND VERIFICATION OF CIRCUITS L-T-P: 3-1-0, Credit: 4

Physical faults and their modeling. Fault equivalence and dominance; fault collapsing. Fault simulation: parallel, deductive and concurrent techniques; critical path tracing. Test generation for combinational circuits: Boolean difference, D-algorithm, Podem, etc. Exhaustive, random and weighted test pattern generation; aliasing and its effect on fault coverage. PLA testing: cross-point fault model, test generation, easily testable designs. Memory testing: permanent, intermittent and pattern-sensitive faults; test generation. Delay faults and hazards; test generation techniques. Test pattern generation for sequential circuits: ad-hoc and structures techniques, scan path and LSSD, boundary scan. Built-in self-test techniques. Verification: logic level (combinational and sequential circuits), RTL-level (data path and control path). Verification of embedded systems. Use of formal techniques: decision diagrams, logic-based approaches.

References

1. N. K. Jha and S. Gupta, Testing of Digital Systems, Cambridge University Press.
2. M. L. Bushnell and V. D. Agrawal, Essentials of Electronic Testing, Kluwer Academic Publishers.
3. M. Abramovici, M. A. Breuer and A. D. Friedman, Digital Systems Testing and Testable Design, Wiley-IEEE Press.
4. P. H. Bardell, W. H. McAnney and J. Savir, Built-in Test for VLSI: Pseudorandom Techniques, Wiley Interscience.
5. P. K. Lala, Fault Tolerant and Fault Testable Hardware Design, Prentice-Hall.
6. A. Krstic and K-T Cheng, Delay Fault Testing for VLSI Circuits, Kluwer Academic Publishers.
7. A. Osseiran (Ed.), Analog and Mixed Signal Boundary Scan, Kluwer Academic Publishers.

CS60040 PARALLEL AND DISTRIBUTED ALGORITHMS L-T-P: 3-0-0, Credit: 3

Fundamentals: Models of parallel and distributed computation, complexity measures; The PRAM Model: balancing, divide and conquer, parallel prefix computation, pointer jumping, symmetry breaking, list ranking, sorting and searching, graph algorithms, parallel complexity and complexity classes, lower bounds; Interconnection Networks: topologies (arrays and mesh networks, trees, systolic networks, hypercubes, butterfly) and fundamental algorithms, matrix algorithms, sorting, graph algorithms, routing, relationship with PRAM models; Asynchronous Parallel Computation; Distributed Algorithms: models and complexity measures, safety, liveness, termination, logical time and event ordering, global state and snapshot algorithms,

mutual exclusion, clock synchronization, election, termination detection, routing, Distributed graph algorithms; Applications of Distributed algorithms.

CS60041 CRYPTOGRAPHY AND NETWORK SECURITY L-T-P: 3-0-0, Credit: 3

Introduction: Basic objectives of cryptography, secret-key and public-key cryptography, one-way and trapdoor one-way functions, cryptanalysis, attack models, classical cryptography.

Block ciphers: Modes of operation, DES and its variants, RCS, IDEA, SAFER, FEAL, BlowFish, AES, linear and differential cryptanalysis.

Stream ciphers: Stream ciphers based on linear feedback shift registers, SEAL, unconditional security.

Message digest: Properties of hash functions, MD2, MD5 and SHA-1, keyed hash functions, attacks on hash functions.

Public-key parameters: Modular arithmetic, gcd, primality testing, Chinese remainder theorem, modular square roots, finite fields.

Intractable problems: Integer factorization problem, RSA problem, modular square root problem, discrete logarithm problem, Diffie-Hellman problem, known algorithms for solving the intractable problems.

Public-key encryption: RSA, Rabin and EIGamal schemes, side channel attacks.

Key exchange: Diffie-Hellman and MQV .

Digital signatures: RSA, DSA and NR signature schemes, blind and undeniable signatures.

Entity authentication: Passwords, challenge-response algorithms, zero-knowledge protocols.

Standards: IEEE, RSA and ISO standards.

Network security: Certification, public-key infra-structure (PKI), secure socket layer (SSL), Kerberos.

Advanced topics: Elliptic and hyper-elliptic curve cryptography, number field sieve, lattices and their applications in cryptography, hidden monomial cryptosystems, cryptographically secure random number generators.

References

1. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press.

2. William Stallings, Cryptography and Network Security: Principles and Practice, Prentice Hall of India.
3. Neal Koblitz, A course in number theory and cryptography, Springer.
4. Johannes A. Buchmann, Introduction to Cryptography, Undergraduate Text in Mathematics, Springer.
5. Doug Stinson, Cryptography Theory and Practice, CRC Press.
6. A. Das and C. E. Veni Madhavan, Public-Key Cryptography: Theory and Practice, Pearson Education Asia.

CS60042 ADVANCES IN COMPILER CONSTRUCTION L-T-P: 3-1-0, Credit: 4

Review of compiler fundamentals - lexical analysis, parsing, semantic analysis, error recovery and intermediate code generation; Runtime storage management; Code generation; Code improvement - peephole optimization, dependence analysis and redundancy elimination, loop optimization, procedural and inter-procedural optimization, instruction scheduling, optimization for memory hierarchy; Compilation for high performance architecture; Portability and retargetability; Selected topics from compilers for imperative, object-oriented and mark-up languages, parallel and distributed programming and concurrency.

References

1. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, Compilers: Principles, Techniques and Tools, Addison-Wesley.
2. Michael L. Scott, Programming Language Pragmatics, Elsevier.
3. Andrew W. Appel, Modern Compiler Implementation in C/Java, Cambridge University Press.
4. Keith D. Cooper and Linda Torczon, Engineering a Compiler, Elsevier.
5. Allen I. Holob, Compiler Design in C, Prentice-Hall.
6. Steven S. Muchnik, Advanced Compiler Design and Implementation, Elsevier.
7. Randy Allen and Ken Kennedy, Optimizing Compilers for Modern Architectures, Elsevier.

CS60043 ALGORITHMS FOR BIOINFORMATICS L-T-P: 3-0-0, Credit: 3

Sequence similarity, homology, and alignment. Pairwise alignment: scoring model, dynamic programming algorithms, heuristic alignment, and pairwise alignment using Hidden Markov Models. Multiple alignment: scoring model, local alignment gapped and ungapped global alignment. Motif finding: motif models, finding occurrence of known sites, discovering new sites. Gene Finding: predicting reading frames, maximal dependence decomposition. Analysis of DNA microarray data using hierarchical clustering, model-based clustering, expectation-maximization clustering, Bayesian model selection.

CS60044 PERFORMANCE EVALUATION AND RELIABILITY OF INFORMATION SYSTEMS L-T-P: 3-0-0, Credit: 3

Review of probability and statistics, stochastic processes, Markov Models, Parameter estimation and hypothesis testing. Models of information systems, introduction to reliability measures. Estimation of MTF and other reliability parameters. Software metrics and software reliability models. Queuing network models, Workload design, Benchmarks, Estimations of performance metrics, case studies.

CS60045 ARTIFICIAL INTELLIGENCE **L-T-P: 3-0-0, Credit: 3**

Problem solving by search: state space, problem reduction, game playing, constraint satisfaction; Automated Reasoning: proposition and first order logic, inference and deduction, resolution refutation, answer extraction, knowledge based systems, logic programming and constrained logic programming, non-monotonic reasoning; Planning: state-space, plan space and partial order planning, planning algorithms; Reasoning under uncertainty: probabilistic reasoning, belief networks; Learning: inductive learning, decision trees, logical approaches, computational learning theory, neural networks, reinforcement learning; Intelligent agents; natural language understanding; Applications.

References

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Prentice-Hall.
2. Nils J. Nilsson, Artificial Intelligence: A New Sythesis, Morgan-Kaufmann.

CS60046 REAL TIME SYSTEMS **L-T-P: 3-0-0, Credit: 3**

Introduction to real time system, embedded systems and reactive systems; Hard and soft real time systems; handling real time; specification and modeling; design methods; real time operating systems; validation and verification; real time process and applications; distributed real time systems.

CS60047 ADVANCED GRAPH THEORY **L-T-P: 3-1-0, Credit: 4**

Basic Concepts: Graphs and digraphs, incidence and adjacency matrices, isomorphism, the automorphism group; Trees: Equivalent definitions of trees and forests, Cayley's formula, the Matrix-Tree theorem, minimum spanning trees; Connectivity: Cut vertices, cut edges, bonds, the cycle space and the bond space, blocks, Menger's theorem; Paths and Cycles: Euler tours, Hamilton paths and cycles, theorems of Dirac, Ore, Bondy and Chvatal, girth, circumference, the Chinese Postman Problem, the Traveling Salesman problem, diameter and maximum degree, shortest paths; Matchings: Berge's Theorem, perfect matchings, Hall's theorem, Tutte's theorem, Konig's theorem, Petersen's theorem, algorithms for matching and weighted matching (in both bipartitie and general graphs), factors of graphs (decompositions of the complete graph), Tutte's f-factor theorem; Extremal problems: Independent sets and covering numbers, Turan's theorem, Ramsey theorems; Colorings: Brooks theorem, the greedy algorithm, the Welsh-Powell bound, critical graphs, chromatic polynomials, girth and chromatic number, Vizing's theorem; Graphs on surfaces: Planar graphs, duality, Euler's formula, Kuratowski's theorem, toroidal graphs, 2-cell embeddings, graphs on other surfaces; Directed graphs: Tournaments, directed paths and

cycles, connectivity and strongly connected digraphs, branchings; Networks and flows: Flow cuts, max flow min cut theorem, perfect square; Selected topics: Dominating sets, the reconstruction problem, intersection graphs, perfect graphs, random graphs.

References

1. Douglas B. West, Introduction to Graph Theory, Prentice Hall of India.
2. Narsingh Deo, Graph Theory with Applications to Engineering and Computer Science. Prentice-Hall.
3. Frank Harary, Graph Theory, Narosa.
4. R. Ahuja, T. Magnanti, and J. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice-Hall.

CS60048 THEORY OF PROGRAMMING LANGUAGES L-T-P: 3-0-0, Credit: 3

Syntax of Programming Languages, Formal languages and automata theory: Finite automata, regular languages, pushdown automata, context free languages, linear bounded automata, context sensitive languages, Turing machines and recursively enumerable sets. Theory of LR(k) parsing, attribute grammars. Semantics of programming languages: Basic mathematical introduction: Propositional and predicate calculus, lambda calculus, algebraic structures. Sequential languages (imperative and applicative): operational semantics, Vienna definition methods. Denotational semantics: Scott-Strachy theory, axiomatic semantics: Floyd-Hoare approach, temporal logic, algebraic semantics and data types.

References

1. Glynn Winskel, A Formal Semantics of Programming Languages: An Introduction, MIT Press.
2. John C. Mitchell, Foundations for Programming Languages, MIT Press.
3. Benjamin C. Pierce, Types and Programming Languages, MIT Press.
4. Daniel P. Friedman, Mitchell Wand and Christopher T. Haynes, Essentials of Programming Languages, Prentice Hall of India.
5. Ravi Sethi, Programming Languages: Concepts and Constructs, Addison-Wesley.
6. H. P. Barendregt, The Lambda Calculus: Its Syntax and Semantics, North-Holland.

CS60049 COMPUTATIONAL COMPLEXITY L-T-P: 3-0-0, Credit: 3

Models of computation, resources (time and space), algorithms, computability, complexity; complexity classes, P/NP/PSPACE, reductions, hardness, completeness, hierarchy, relationships between complexity classes; Randomized computation and complexity; Logical characterizations, incompleteness; approximability; circuit complexity, lower bounds; parallel computation and complexity; counting problems; interactive proofs; probabilistically checkable proofs; communication complexity; Quantum computation.

References

1. Christos H. Papadimitriou, Computational Complexity, Addison-Wesley Longman.
2. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.
3. John E. Hopcroft and Jeffrey D. Ullman, Introduction to Automata, Languages and Computation, Addison-Wesley, 1979.
4. J. Balcazar, J. Diaz, and J. Gabarro, Structural Complexity, Volumes I and II, Springer.

CS60050 MACHINE LEARNING **L-T-P: 3-0-0, Credit: 3**

The concept learning task. General-to-specific ordering of hypotheses. Version spaces. Inductive bias. Decision Tree Learning. Rule Learning: Propositional and First-Order, Over-fitting, Cross-Validation. Experimental Evaluation of Learning Algorithms Instance-Based Learning: k-Nearest neighbor algorithm, Radial basis functions. Case-based learning. Computational Learning Theory: probably approximately correct (PAC) learning. Sample complexity. Computational complexity of training. Vapnik-Chervonenkis dimension. Artificial Neural Networks: Linear threshold units, Perceptrons, Multilayer networks and back-propagation, recurrent networks. Probabilistic Machine Learning Maximum Likelihood Estimation, MAP, Bayes Classifiers Naive Bayes. Bayes optimal classifiers. Minimum description length principle. Bayesian Networks, Inference in Bayesian Networks, Bayes Net Structure Learning Unlabelled data: EM, preventing overfitting, cotraining Gaussian Mixture Models, K-means and Hierarchical Clustering, Clustering and Unsupervised Learning, Hidden Markov Models, Reinforcement Learning Support Vector Machines Ensemble learning: boosting, bagging.

References

1. Tom Mitchell, Machine Learning, McGraw-Hill.
2. Soumen Chakrabarti, Mining the Web: Discovering Knowledge from Hypertext Data, Morgan-Kaufmann.

CS60051 DISCRETE STRUCTURES **L-T-P: 3-1-0, Credit: 4**

Propositional Logic, Proof Methods of Implications, Sets, Basic operations on sets, Functions, Relations, Binary relations: Equivalence Relations, Partial orders and posets. Mathematical induction, pigeonhole principle, first order logic and other proof methods. Cardinality of sets, finite and infinite sets, countable and uncountable sets, Cantor's theorem. Algebraic structures: Semigroups, monoids, Groups, Substructures and morphisms, rings, fields and vector spaces; lattices, Boolean algebras, morphisms of Boolean algebras; basic counting principles, permutations, combinations, recurrence relations and their solutions.

References

1. Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata McGraw-Hill.
2. C. L. Liu, Elements of Discrete Mathematics, Tata McGraw-Hill.
3. Norman L. Biggs, Discrete Mathematics, Oxford University Press.

4. Kenneth Bogart, Clifford Stein and Robert L. Drysdale, Discrete Mathematics for Computer Science, Key College Publishing.
5. Thomas Koshy, Discrete Mathematics with Applications, Elsevier.
6. Ralph P. Grimaldi, Discrete and Combinatorial Mathematics, Pearson Education, Asia.

CS60052 ADVANCED DIGITAL IMAGE PROCESSING AND COMPUTER VISION **L-T-P: 3-0-0, Credit: 3**

Sensor and Imaging: Imaging Optics, Radiometry of Imaging, Illumination sources and techniques, Camera Principles, Color Imaging, Single Sensor Color Imaging and Color Demosaicing, Range Images, 3D Imaging. Signal Representation: Vector Space and Unitary Transforms, Multi-Resolutional Signal Representation, Wavelet Decomposition, Scale space and diffusion, Representation of color, Retinex Processing, Markov Random Field Modeling of Images. Non-linear Image Processing: Median and Order Statistics Filters, Rank-Ordered-Mean Filters and Signal Dependent Rank-Ordered-Mean Filters, Two Dimensional Teager Filters, Applications of nonlinear filters in image enhancement, edge detections, noise removal etc. Feature Estimation: Morphological Operations, Edge Detection, Edges in multichannel images, Texture Analysis, Optical flow based motion estimation, Reflectance based shape recovery, Depth from focus, Stereo matching and depth estimation. Image and Video Compression Standards: Lossy and lossless compression schemes: Transform Based, Sub-band Decomposition, Entropy Encoding, JPEG, JPEG2000, MPEG-1, MPEG-4, and MPEG-7. Object Analysis, Classification: Bayesian Classification, Fuzzy Classification, Neural Network Classifiers, Shape Reconstruction from volumetric data, knowledge-based interpretation of images.

CS60053 VLSI SYSTEM DESIGN **L-T-P: 3-1-0, Credit: 4**

Introduction to VLSI Design, Different types of VLSI design styles: Full custom, standard cell based, gate array based, programmable logic, field programmable gate arrays etc. VLSI Design flow. CMOS logic: PMOS, NMOS and CMOS, Electrical characteristics, operation of MOS transistors as a switch and an amplifier, MOS inverter, stick diagram, design rules and layout, delay analysis, different type of MOS circuits: Dynamic logic, BiCMOS, pass transistors etc. CMOS process, Combinational logic cells, Sequential logic cells, Datapath logic cells, I/O cells. ASIC Library Design: Transistors as Resistors and parasitic Capacitance, Logical effort, gate array, standard cell and datapath cell design. Introduction to hardware description language (HDL) Verilog/VHDL. A logic synthesis example. Floor-planning and Placement: I/O and power planning, clock planning. Routing global and detailed. Example design technique: mapping of architecture to silicon.

References

1. N. H. E. Weste and K. Eshraghian, Principles of CMOS VLSI Design : A Systems Perspective, Pearson Education.
2. W. Wolf, Modern VLSI Design: Systems on Silicon, Pearson Education.

3. J. Rabaey, A. Chandrakasan and B. Nikolic, Digital Integrated Circuits: A Design Perspective, Prentice Hall of India.
4. M. Sarafzadeh and C. K. Wong, An Introduction to VLSI Physical Design, MCGraw-Hill.
5. D. D. Gajaski, N. D. Dutt, A. C.-H. Wu and S. Y.-L. Lin, High-Level Synthesis: Introduction to Chip and System Design, Kluwer Academic Publishers.

CS60054 LOW POWER CIRCUITS AND SYSTEMS L-T-P: 3-0-0, Credit: 3

Basics of MOS circuits: MOS transistor structure and device modeling, MOS inverters, MOS combinational circuits - different logic families.

Sources of power dissipation in CMOS circuits: static power dissipation - diode leakage power, subthreshold leakage power, gate and other tunnel currents; dynamic power dissipation - short circuit power, switching power, gliching power; degrees of freedom.

Supply voltage scaling approaches: technology Level - feature size scaling, threshold voltage scaling; logic level - gate sizing for voltage scaling; architecture level - parallelism and pipelining; algorithm level - transformations to exploit concurrency; dynamic voltage scaling.

Switched capacitance minimization approaches: system level - power down, system partitioning; algorithm level - concurrency, locality, regularity, data representation; architecture level - concurrency, signal correlation; logic level - gate sizing, logic styles; layout level - layout optimization; technology level - advanced packaging, SOI.

Leakage power control techniques: threshold voltage scaling: MTCMOS, VTCMOS and Multiple-Vt CMOS circuits; gate sizing.

Special Topics: adiabatic switching, battery aware synthesis.

References

1. Sung-Mo Kang, Yusuf Leblebici, CMOS Digital Integrated Circuits, Tata McGraw-Hill .
2. Neil H. E. Weste and K. Eshraghian, Principles of CMOS VLSI Design, Addison-Wesley (Indian reprint).
3. A. Bellamour, and M. I. Elmasri, Low Power VLSI CMOS Circuit Design, Kluwer Academic Press.
4. Anantha P. Chandrakasan and Robert W. Brodersen, Low Power Digital CMOS Design, Kluwer Academic Publishers.
5. Kaushik Roy and Sharat C. Prasad, Low-Power CMOS VLSI Design, Wiley-Interscience.

CS60055 UBIQUITOUS COMPUTING L-T-P: 3-0-0, Credit: 3

Overview of wireless technologies, Signal propagation, Multiplexing, Modulation, and Spread spectrum techniques. Media access control: FDMA, TDMA, CDMA. Cellular systems: AMPS, GSM, DECT, UMTS, IMT-2000. CDMA-based cellular systems. Satellite systems: basic routing, localization, and handoff issues. Wireless Networks: packet radio network, Wireless LAN, IEEE 802.11b, Blue-tooth, Wireless ATM. Wireless Application Protocol (WAP) and WML. Mobile Networking: Mobile IP, Ad-Hoc Networks: AODV, DSR, DSDV routing. Wireless TCP: indirect TCP, Snooping TCP, Mobile TCP. Information Management, Location-Independent and Location-dependent computing models, Mobile applications and services, Security.

CS60056 COMPUTER GRAPHICS **L-T-P: 3-1-0, Credit: 4**

Introduction: Display of entities, Geometric computation and representation, Graphics Environments; Working Principles of display devices: refreshing raster scan devices, vector devices, Cathode Ray Tube Terminals, Plotters; Display of colors: Look Up Tables, display of gray shades, Half toning; Display and drawing of graphics primitives: point, line, polygon, circle, curves and text; Coordinate Conventions: world coordinates, device coordinates, normalized device coordinates, view-port and window, zooming and panning by changing coordinate reference frames; Computations on polygons: point inclusion problem, polygon filling, polygon intersection, clipping, polygonization of a point set, convex hull computation, triangulation of polygons; Transformations in 2D and 3D: translation, rotation, scaling, reflection, Projection: perspective and parallel projections, isometric projection, Transformation matrices; Volume and Surface Representation: polygonal meshes, parametric curves and surfaces, Cubic and Bicubic Splines, Voxel, Octree and Medial Axis representation, Sweep Representation, Surfaces and Volumes by rotation of curves and surfaces, fractal modeling; Hidden surface and line elimination: Elimination of back surfaces, painters' algorithms, Binary Space Partitioning Tree; Rendering and Visualization: Shading model, Constant, Goraud and Phong Shading, Ray tracing algorithm, Radiosity Computation; Computer Animation: fundamental concepts.

CS60057 SPEECH AND NATURAL LANGUAGE PROCESSING **L-T-P: 3-0-0, Credit: 3**

Speech and Natural Language Processing: Introduction; Brief Review of Regular Expressions and Automata; Finite State Transducers; Word level Morphology and Computational Phonology; Basic Text to Speech; Introduction to HMMs and Speech Recognition. Indian language case studies; Part of Speech Tagging; Parsing with CFGs; Probabilistic Parsing. Representation of Meaning; Semantic Analysis; Lexical Semantics; Word Sense; Disambiguation; Discourse understanding; Natural Language Generation; Techniques of Machine Translation; Indian Language case studies.

References

1. Daniel Jurafsky and James H. Martin, Speech and Language Processing, Prentice-Hall.
2. Chris Manning and Hinrich Schuetze, Foundations of Statistical Natural Language Processing, MIT Press.

CS60058 FAULT TOLERANT SYSTEMS **L-T-P: 3-0-0, Credit: 3**

Fundamental concepts in the theory of reliable computer systems design. Introduction to redundancy theory, limit theorems; decision theory in redundant systems. Hardware fault tolerance, redundancy techniques, detection of faults, replication and compression techniques, self-repairing techniques, concentrated and distributed voters, models of fault tolerant computing systems. Case studies. Software fault tolerance: fault tolerance versus fault intolerance, errors and their management strategies. Implementation techniques: software defense, protective redundancy, architectural support. Fault recovery techniques. Coding theory: application to fault tolerant system design. Fault-tolerance and reliability of multicomputer networks (direct and indirect) including fault-tolerant routing and sparing techniques. Yield and reliability enhancement techniques for VLSI/WSI array processors.

CS60059 OBJECT ORIENTED SYSTEMS **L-T-P: 3-0-0, Credit: 3**

Review of programming practices and code-reuse; Object model and object-oriented concepts; Object-oriented programming languages and implementation; Object-oriented analyses and design using UML structural, behavioral and architectural modeling; Unified development process, Software reuse design patterns, components and framework; Distributed object computing, interoperability and middleware standards COM/DCOM and CORBA; Object-oriented database system data model, object definition and query language, object-relational system.

References

1. Bertrand Meyer, Object Oriented Software Construction, Prentice-Hall.
2. Grady Booch, Object Oriented Analysis and Design, Addison-Wesley.
3. Grady Booch, James Rumbaugh and Ivar Jacobson, Unified Modeling Language Guide, Addison-Wesley.
4. Erich Gamma et al., Design Patterns: Elements of Reusable OO Software, Addison-Wesley.
5. Michael L. Scott, Programming Language Pragmatics, Morgan-Kaufmann.
6. Kim Bruce, Foundations of Object Oriented Languages, Prentice-Hall.
7. Benjamin C. Pierce, Types and Programming Languages, Prentice-Hall.
8. Bjarne Stroustrup, The Design and Evolution of C++, Addison-Wesley.
9. Bill Venners, Inside the JAVA 2 Virtual Machine, McGraw Hill.
10. James E. Smith and Ravi Nair, Virtual Machines, Elsevier/Morgan-Kaufmann.
11. Saba Zamir, Handbook of Object Technology, CRC Press.

CS60060 FORMAL SYSTEMS **L-T-P: 3-0-0, Credit: 3**

Formal languages and their related automata, Turing machines, type-0 languages, linear bounded automata and CSLs. Time and tape bounded Turing machines, time and space bounds for recognizing CFLs. Turing Computability: number theoretic computations by Turing machines and indexing. Axiomatic systems, their soundness and completeness. Recursive function theory: primitive recursive functions and primitive recursive predicates. Ackermann's function, recursive

and general recursive functions. Computability and decidability: computable functions, computable sets, decision problems. Fix-point theory of programs, functions and functionals, verification methods, Lambda calculus and applications.

References

1. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.
2. Fred C. Hennie. Introduction to Computability. Addison-Wesley.
3. Bernard M. Moret, The Theory of Computation, Pearson Education Asia.
4. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.
5. John Martin, Introduction to Languages and The Theory of Computation, Tata McGraw Hill.

CS60062 MULTIMEDIA SYSTEMS **L-T-P: 3-0-0, Credit: 3**

An overview of multimedia system and media streams; Source representation and compression techniques text, speech and audio, still image and video; Graphics and animation; Multi-modal communication; Multimedia communication, video conferencing, video-on-demand broadcasting issues, traffic shaping and networking support; Transcoding; Multimedia OS and middleware; Synchronization and QoS; Multimedia servers, databases and content management; Multimedia information system and applications.

References

1. Ralf Steinmetz and Klara Nahrstedt, Multimedia Systems, Springer.
2. J. D. Gibson, Multimedia Communications: Directions and Innovations, Springer.
3. K. Sayood, Introduction to Data Compression, Morgan-Kaufmann.
4. A. Puri and T. Chen, Multimedia Systems, Standards, and Networks, Marcel Dekker.
5. Iain E.G. Richardson, H.264 and MPEG-4 Video Compression, John Wiley.
6. Borivoje Furht, Handbook of Multimedia Computing, CRC Press.

CS60064 COMPUTATIONAL GEOMETRY **L-T-P: 3-0-0, Credit: 3**

Convex hulls: construction in 2d and 3d, lower bounds; Triangulations: polygon triangulations, representations, point-set triangulations, planar graphs; Voronoi diagrams: construction and applications, variants; Delaunay triangulations: divide-and-conquer, flip and incremental algorithms, duality of Voronoi diagrams, min-max angle properties; Geometric searching: point location, fractional cascading, linear programming with prune and search, finger trees, concatenable queues, segment trees, interval trees; Visibility: algorithms for weak and strong visibility, visibility with reflections, art-gallery problems; Arrangements of lines: arrangements of hyperplanes, zone theorems, many-faces complexity and algorithms; Combinatorial geometry: Ham-sandwich cuts, Helly's theorems, k-sets, polytopes and hierarchies, polytopes and linear programming in d-dimensions, complexity of the union of convex sets, simply connected sets and visible regions; Sweep techniques: plane sweep for segment intersections, Fortune's sweep

for Voronoi diagrams, topological sweep for line arrangements; Randomization in computational geometry: algorithms, techniques for counting; Robust geometric computing; Applications of computational geometry.

References

1. Mark de Berg, Otfried Schwarzkopf, Marc van Kreveld and Mark Overmars, Computational Geometry: Algorithms and Applications, Springer.
2. F. P. Preparata and Michael I. Shamos, Computational Geometry: An Introduction, Springer.
3. Joseph O' Rourke, Computational Geometry in C, Cambridge University Press.
4. Lecture Notes by David Mount.

CS60066 SOFTWARE ENGINEERING **L-T-P: 3-0-0, Credit: 3**

Introduction. Life cycle models, Requirements analysis and specification, Formal requirements specification. Fundamental issues in software design: goodness of design, cohesion, coupling. Function-oriented design: structured analysis and design. Overview of object-oriented concepts. Unified Modeling Language (UML). Unified design process. User interface design. Coding standards and guidelines. Code walkthrough and reviews. Unit testing. Black box and white box testing. Integration and system testing. Software quality and reliability. SEI CMM and ISO 9001. PSP and Six Sigma. Cleanroom technique. Software project management. Configuration management. Software maintenance issues and techniques. Software reuse. Client-server software development.

References

1. Rajib Mall, Fundamentals of Software Engineering, Prentice Hall India.
2. Pankaj Jalote, An integrated approach to Software Engineering, Springer/Narosa.
3. Roger S. Pressman, Software Engineering: A practitioner's approach, McGraw Hill.
4. Ian Sommerville, Software Engineering, Addison-Wesley.

CS60068 CAD FOR VLSI **L-T-P: 3-0-0, Credit: 3**

Introduction: VLSI design flow, challenges. Verilog/VHDL: introduction and use in synthesis, modeling combinational and sequential logic, writing test benches. Logic synthesis: two-level and multilevel gate-level optimization tools, state assignment of finite state machines. Basic concepts of high-level synthesis: partitioning, scheduling, allocation and binding. Technology mapping. Testability issues: fault modeling and simulation, test generation, design for testability, built-in self-test. Testing SoC's. Basic concepts of verification. Physical design automation. Review of MOS/CMOS fabrication technology. VLSI design styles: full-custom, standard-cell, gate-array and FPGA. Physical design auto-mation algorithms: floor-planning, placement, routing, compaction, design rule check, power and delay estimation, clock and power routing, etc. Special considerations for analog and mixed-signal designs.

CS60070 QUANTUM COMPUTING AND QUANTUM INFORMATION PROCESSING **L-T-P: 3-1-0, Credit: 4**

Mathematical foundations; quantum mechanical principles; quantum entanglement; reversible computation, qubits, quantum gates and registers; universal gates for quantum computing; quantum parallelism and simple quantum algorithms; quantum Fourier transforms and its applications, quantum search algorithms; elements of quantum automata and quantum complexity theory; introduction to quantum error correcting codes; entanglement assisted communication; elements of quantum information theory and quantum cryptography.

References

1. M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press.
2. Jozef Gruska, Quantum Computing, McGraw-Hill.
3. Lecture notes by John Preskill and N. D. Mermin available in the Internet.
4. Los Alamos Quant_ph archive.
5. Current Literature.

CS60076 ADVANCES IN DIGITAL AND MIXED SIGNAL TESTING **L-T-P: 3-0-0, Credit: 3**

Delay fault testing: path delay test, transition faults, delay test methodologies. IDDQ testing: basic concept, faults detected, test generation, limitations, IDDQ design for testability. Functional testing of arithmetic and regular arrays. Functional testing of microprocessors and microcontrollers. Sequential circuit testing: time frame expansion and simulation-based approaches to ATPG, design of testable FSMs, use of coding theory. Advanced BIST techniques: theory of linear machines, practical BIST architectures. System-on-chip design and test: SOC testing problem, core-based design and system wrapper, proposed test architectures for SOC, platform-based design and testability issues.

DSP-based analog and mixed-signal test: func-tional DSP-based testing, static ADC and DAC testing methods, realizing emulated instruments, CODEC testing, future challenges. Model-based analog and mixed-signal test: analog fault models, levels of abstraction, analog fault simulation, analog ATPG. Analog test bus standard: analog circuit DFT, analog test bus, IEEE 1149.4 standard.

References

1. M. L. Bushnell and V. D. Agrawal, Essentials of Electronic Testing, Kluwer Academic Publishers.
2. A. Osseiran, Analog and mixed-signal boundary scan: a guide to the IEEE 1149.4 test standard, Kluwer Academic Publishers.
3. A. Krstic and K-T. Cheng, Delay fault testing for VLSI circuits, Kluwer Academic Publishers.

4. S. Chakravarty and P. J. Thadikaran, Introduction to IDDQ testing, Kluwer Academic Publishers.

CS60078 COMPLEX NETWORKS **L-T-P: 3-0-0, Credit: 3**

Objective

- Study of the models and behaviors of networked systems.
- Empirical studies of social, biological, technological and information networks.
- Exploring the concepts of small world effect, degree distribution, clustering, network correlations, random graphs, models of network growth, and preferential attachment and dynamical processes taking place on networks.

Content

Types of network: Social networks, Information networks, Technological networks, Biological networks.

Properties of network: Small world effect, transitivity and clustering, degree distribution, scale free networks, maximum degree; network resilience; mixing patterns; degree correlations; community structures; network navigation.

Random Graphs: Poisson random graphs, generalized random graphs, the configuration model, power-law degree distribution, directed graph, bipartite graph, degree correlations.

Models of network growth: Price's model, Barabasi and Albert's model, other growth models, vertex copying models.

Processes taking place on networks: Percolation theory and network resilience, Epidemiological processes.

Applications: Search on networks, exhaustive network search, guided network search, network navigation; network visualization.

References

1. S. N. Dorogovtsev and J. F. F. Mendes, Evolution of Networks, Oxford University Press.
2. Narsingh Deo, Graph Theory, Prentice Hall of India.
3. Current Literature.

CS60080 INFORMATION RETRIEVAL **L-T-P: 3-0-0, Credit: 3**

Introduction to Information Retrieval: The nature of unstructured and semi-structured text. Inverted index and Boolean queries.

Text Indexing, Storage and Compression: Text encoding: tokenization, stemming, stop words, phrases, index optimization. Index compression: lexicon compression and postings, lists compression. Gap encoding, gamma codes, Zipf's Law. Index construction. Postings size estimation, merge sort, dynamic indexing, positional indexes, n-gram indexes, real-world issues.

Retrieval Models: Boolean, vector space, TFIDF, Okapi, probabilistic, language modeling, latent semantic indexing. Vector space scoring. The cosine measure. Efficiency considerations. Document length normalization. Relevance feedback and query expansion. Rocchio.

Performance Evaluation: Evaluating search engines. User happiness, precision, recall, F-measure. Creating test collections: kappa measure, interjudge agreement.

Text Categorization and Filtering: Introduction to text classification. Naive Bayes models. Spam filtering. Vector space classification using hyperplanes; centroids; k Nearest Neighbors. Support vector machine classifiers. Kernel functions. Boosting.

Text Clustering: Clustering versus classification. Partitioning methods. k-means clustering. Mixture of Gaussians model. Hierarchical agglomerative clustering. Clustering terms using documents.

Advanced Topics: Summarization, Topic detection and tracking, Personalization, Question answering, Cross language information retrieval.

Web Information Retrieval: Hypertext, web crawling, search engines, ranking, link analysis, PageRank, HITS, XML and Semantic web.

References

1. Manning, Raghavan and Schutze, Introduction to Information Retrieval, Cambridge University Press.
2. Baeza-Yates and Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley.
3. Soumen Charabarti, Mining the Web, Morgan-Kaufmann.
4. Survey by Ed Greengrass available in the Internet.

CS60082 COMPUTATIONAL NUMBER THEORY L-T-P: 3-0-0, Credit: 3

Algorithms for integer arithmetic: Divisibility, gcd, modular arithmetic, modular exponentiation, Montgomery arithmetic, congruence, Chinese remainder theorem, Hensel lifting, orders and primitive roots, quadratic residues, integer and modular square roots, prime number theorem, continued fractions and rational approximations.

Representation of finite fields: Prime and extension fields, representation of extension fields, polynomial basis, primitive elements, normal basis, optimal normal basis, irreducible polynomials.

Algorithms for polynomials: Root-finding and factorization, Lenstra-Lenstra-Lovasz algorithm, polynomials over finite fields.

Elliptic curves: The elliptic curve group, elliptic curves over finite fields, Schoof's point counting algorithm.

Primality testing algorithms: Fermat test, Miller-Rabin test, Solovay-Strassen test, AKS test.

Integer factoring algorithms: Trial division, Pollard rho method, p-1 method, CFRAC method, quadratic sieve method, elliptic curve method.

Computing discrete logarithms over finite fields: Baby-step-giant-step method, Pollard rho method, Pohlig-Hellman method, index calculus methods, linear sieve method, Coppersmith's algorithm.

Applications: Algebraic coding theory, cryptography.

References

1. Victor Shoup, A Computational Introduction to Number Theory and Algebra, Cambridge University Press.
2. Maurice Mignotte, Mathematics for Computer Algebra, Springer-Verlag.
3. Ivan Niven, Herbert S. Zuckerman and H. L. Montgomery, An Introduction to the Theory of Numbers, John Wiley.
4. Joachim von zur Gathen and Juergen Gerhard, Modern Computer Algebra, Cambridge University Press.
5. Rudolf Lidl and Harald Niederreiter, Introduction to Finite Fields and their Applications, Cambridge University Press.
6. Alfred J. Menezes, editor, Applications of Finite Fields, Kluwer Academic Publishers.
7. Joseph H. Silverman and John Tate, Rational Points on Elliptic Curves, Springer International Edition.
8. D. R. Hankerson, A. J. Menezes and S. A. Vanstone, Guide to Elliptic Curve Cryptography, Springer-Verlag.
9. A. Das and C. E. Veni Madhavan, Public-key Cryptography: Theory and practice, Pearson Education Asia.
10. Henri Cohen, A Course in Computational Algebraic Number Theory, Springer-Verlag.

CS60084 FOUNDATIONS OF CRYPTOGRAPHY L-T-P: 3-0-0, Credit: 3

Introduction to Cryptography: Basics of Symmetric Key Cryptography, Basics of Assymetric Key Cryptography, Hardness of Functions

Notions of Semantic Security (SS) and Message Indistinguishability (MI): Proof of Equivalence of SS and MI, Hard Core Predicate, Trap-door permutation, Goldwasser-Micali Encryption

Goldreich-Levin Theorem: Relation between Hardcore Predicates and Trap-door permutations

Formal Notions of Attacks: Attacks under Message Indistinguishability: Chosen Plaintext Attack(IND-CPA), Chosen Ciphertext Attacks (IND-CCA1 and IND-CCA2), Attacks under Message Non-malleability: NM-CPA and NM-CCA2, Inter-relations among the attack model

Random Oracles: Provable Security and asymmetric cryptography, hash functions

One-way functions: Weak and Strong one way functions

Pseudo-random Generators (PRG): Blum-Micali-Yao Construction, Construction of more powerful PRG, Relation between One-way functions and PRG, Pseudo-random Functions (PRF)

Building a Pseudorandom Permutation: The Luby Rackoff Construction: Formal Definition, Application of the Luby Rackoff Construction to the construction of Block Ciphers, The DES in the light of Luby Rackoff Construction

Left or Right Security (LOR)

Message Authentication Codes (MACs): Formal Definition of Weak and Strong MACs, Using a PRF as a MAC, Variable length MAC

Public Key Signature Schemes: Formal Definitions, Signing and Verification, Formal Proofs of Security of Full Domain Hashing

Assumptions for Public Key Signature Schemes: One way functions Imply Secure One-time Signatures

Shamir's Secret Sharing Scheme

Formally Analyzing Cryptographic Protocols

Zero Knowledge Proofs and Protocols

References

1. Hans Delfs and Helmut Knebl, Introduction to Cryptography: Principles and Applications, Springer Verlag.
2. Wenbo Mao, Modern Cryptography, Theory and Practice, Pearson Education (Low Priced Edition)
3. Shaffi Goldwasser and Mihir Bellare, Lecture Notes on Cryptography, Available at http://citeseerx.ist.psu.edu/.

4. Oded Goldreich, Foundations of Cryptography, CRC Press (Low Priced Edition Available), Part 1 and Part 2

CS69003 Computer Systems Laboratory I
L-T-P: 0-0-6, Credit: 4

Object-oriented programming concepts and implementation of abstract data types. Implementation of graph algorithms. Linear programming with applications. Basics of OS programming: process creation and synchronization, shared memory and semaphore, shell programming.

CS69004 Computer Systems Laboratory II
L-T-P: 0-0-6, Credit: 4

Socket programming, database creation and update, building large client server applications. Basics of compiler writing using lex and yacc.