

5CS1 COMPUTER ARCHITECTURE (Common to Comp. Engg. & Info. Tech)

Class: V Sem. B.Tech.	Evaluation
Branch: Computer Engg. Schedule per Week Lectures: 3, Tutorial: 1	Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]

Units	Contents of the subject
I	Introduction to Computer Architecture and Organization. Von Neuman Architecture, Flynn Classification. Register Transfer and Micro operations: Register transfer language, Arithmetic Micro-operations, Logic Micro-operations, Shift Micro-operations, Bus and memory transfers. Computer Organization and Design: Instruction cycle, computer registers, common bus system, computer instructions, addressing modes, design of a basic computer
II	Central Processing Unit: General register organization, stack organization, Instruction formats, Data transfer and manipulation, program control. RISC, CISC characteristics. Pipeline and Vector processing: Pipeline structure, speedup, efficiency, throughput and bottlenecks. Arithmetic pipeline and Instruction pipeline.
III	Computer Arithmetic: Adder, Ripple carry Adder, carry look Ahead Adder, Multiplication: Add and Shift, Array multiplier and Booth Multiplier, Division: restoring and Non-restoring Techniques. Floating Point Arithmetic: Floating point representation, Add, Subtract, Multiplication, Division.
IV	Memory Organization: RAM, ROM, Memory Hierarchy, Organization, Associative memory, Cache memory, and Virtual memory: Paging and Segmentation.
V	Input-Output Organization: Input-Output Interface, Modes of Transfer, Priority Interrupt, DMA, IOP processor.

Text/References:

1. Computer Organization and Architecture - William Stallings (Pearson Education Asia)
2. Computer Organization and Architecture -John P. Hayes (McGraw -Hill)
3. Computer Organization -V. Carl. Hamacher (McGraw-Hill)

RAJASTHAN TECHNICAL UNIVERSITY

Detailed Syllabus B.Tech. (Comp. Engg.) V-VI Sem. 2010-11

5CS2 Digital Logic Design (Comp. Engg.)

Class: V Sem. B.Tech.	Evaluation
Branch: Computer Engg. Schedule per Week Lectures: 3	Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]

Units	Contents of the subject
I	Hardware Description Languages and their use in digital logic design. VHDL: Modelling Concepts, Lexical Elements & Syntax Descriptions, Scalar Data types & Operations, Sequential Statements, Composite Data Types & Operations, Basic Modelling Constructs. Case Study: VHDL Simulation of Ripple Carry, & Look Ahead carry Adders.
II	VHDL: Subprograms, Packages & Use Clauses, Aliases, Resolved Signals, Components & Configurations, Generate Statements, Concurrent Statements. Use of VHDL in simulation and synthesis.
III	Clocked Sequential circuits. Design steps for synchronous sequential circuits. Design of a sequence detector. Moore and Mealy Machines. Design using JK flip-flops and D flip-flops. State reduction, State assignment, Algorithmic State Charts, converting ASM charts to hardware, one-hot state assignment. Considerations of clock skew, set-up time, hold-time and other flip-flop parameters, timing constraints. Programmable Logic Devices. Read-only memory. Boolean function implementation through ROM. PLD, PGA, PLA, PAL, FPGA.
IV	Event-driven Circuits. Design procedure for asynchronous circuits, stable and unstable states, races, race-free assignments. State reduction of incompletely specified machines. Compatibility and state reduction procedure. Hazards in combinational networks. Dynamic hazards, Function Hazards, and Essential Hazards. Eliminating hazards.
V	Field Programmable Gate Arrays: Introduction, Logic Elements & programmability, Interconnect structures & programmability, Extended Logic Elements, SRAM, Flash Memory & Antifuse Configuration, Case Studies of Altera Stratix & Xilinx Virtex-II pro. Technology Mapping for FPGAs: Logic Synthesis, Lookup Table Technology Mapping.

Text Book:

1. Brian Holdsworth and Clive Woods. Digital Logic Design. Newnes (Elsevier). [Available in Indian Edition].
2. Ashenden, The Designer's Guide to VHDL, Elsevier.
3. Stephen D. Brown, et.al., Field Programmable Gate Arrays, Kluwer Academic Publishers.
4. Scott Hauck, André DeHon, Reconfigurable computing: the theory and practice of FPGA-based computation, Morgan Kaufman
5. Zvi Kohavi: Switching and Finite Automata Theory. TMH.
6. Parag K. Lala, Practical Digital Logic Design and Testing. PHI
7. Stephen H. Unger, The essence of logic circuits. Wiatrowski & House.

5CS3 TELECOMMUNICATION FUNDAMENTALS (Common to Comp. Engg. & Info. Tech)

Class: V Sem. B.Tech.	Evaluation
Branch: Computer Engg. Schedule per Week Lectures: 3	Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]

I	Data Transmission: Terminology, Frequency, spectrum, bandwidth, analog and digital transmission, Transmission impairments, channel capacity including sampling theorem and Fourier series. Transmission Media: Transmission of signals through Twisted pair, Coaxial cable, optical fibre (SM, MM, Graded Index). Wireless Transmission: Antenna and antenna gain, introduction to terrestrial and satellite microwave, Propagation of wireless signals, free space loss for LOS communication. Review of Line Encoding Schemes. Concept of bit period, effect of clock skew, Synchronous and Asynchronous communication Network Reference Models (OSI/ISO and TCP/IP)
II	Data Link Layer: Functions performed by data link layer, Data link Layer design issues Error Control Coding: Error Detection, Two Dimensional Parity Checks, Internet Checksum. Polynomial Codes, Standardized polynomial codes, error detecting capability of a polynomial codes. Linear codes, performance of linear codes, error detection & correction using linear codes. Flow Control: Flow control in loss less and lossy channels using stop-and-wait, sliding window protocols. Performance of protocols used for flow control.
III	Data Link Control: HDLC & PPP including frame structures, MAC sublayer: Pure and slotted Aloha, CSMA, CSMA/CD, collision free multiple access. Throughput analysis of pure and slotted Aloha, delay & throughput analysis of CSMA and CSMA/CD
IV	Multiplexing: Frequency division, time division (Synchronous and statistical) multiplexing. ADSL, DS1 and DS3 carriers. Multiple Accesses: Performance of FDMA-FM-FDMA, Single channel per carrier. TDMA frame structure, TDMA Burst Structure, TDMA Frame efficiency, TDMA Superframe structure, Frame acquisition and synchronization, Slip rate and in digital terrestrial networks. Switching: Qualitative description of Space division, time division and space-time-space division switching.
V	Spread Spectrum Techniques: Direct sequence(DSSS) & frequency hopping(FHSS); Performance consideration in DSSS & FHSS; Code division Multiple access (CDMA): frequency & channel specifications, forward & reverse CDMA channel, pseudo noise(PN) sequences, m-sequence, gold sequence, orthogonal code, gold sequences, Walsh codes synchronization, power control, handoff, capacity of CDMA system, IMT-2000, WCDM

Text/References:

1. Stallings, Data and computer communication, 8th ed. Pearson
2. Tri.T.Ha, Digital Satellite Communications, 2/e, Tata McGraw Hill
3. Alberto Leon-Garcia, Indra Widjaja, COMMUNICATION NETWORKS, 2nd ed., TMH
4. Wireless Communications, 2/e, Rappaport, PHI
5. Analysis of Computer and Communication Networks, ISBN: 0387744363, FayeZ Gebali, 2008, Springer-verlag, 1st Ed.

5CS4 DATABASE MANAGEMENT SYSTEMS (Common to Comp. Engg. & Info. Tech)

Class: V Sem. B.Tech.		Evaluation
Branch: Computer Engg. Schedule per Week Lectures: 3		Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]
Units	Contents of the subject	
I	<p>Introduction: Applications, Purpose, File System v/s DBMS, Data Abstraction (views), Structure of a DBMS-Query Processor, Database Users and Administrator, Data Dictionary, Transaction Manager, Storage Manager. Data Models Introduction-Network Model, Hierarchical Model, Relational Model, Entity Relationship Model and Object Oriented Model. [1]</p> <p>Entity Relationship Model: Structure of RDMS and Database Schema, Entities, Attributes and Entity Sets, Relationship and Relationship Sets, Key Constraints, Participation Constraints (Mapping Cardinalities), Integrity Constraints, Weak Entity Set, Design issues, Extended Features- Aggregation, Generalization and Specialization, case study of an Enterprise. [1]</p>	
II	<p>Relational Algebra: Operations: Selection, Projection, Set, Renaming, Joins, Division. Relational calculus- Tuple Relational Calculus, Domain Relational Calculus. [2]</p> <p>Query Languages: Procedural and Non Procedural, DDL, DCL and DML. SQL-Clauses, Nested Queries, SQL Functions- Single Row Function, Multigroup Functions, Set Operations, Aggregate Operators, Null Values, Embedded SQL, Dynamic SQL. [2]</p>	
III	<p>Schema Refinement And Normal Forms: Introductions to Schema Refinement, Functional Dependencies, Boyce-Codd Normal Forms, Third Normal Form, Normalization-Decomposition into BCNF Decomposition into 3-NF, Denormalization, Triggers. [2]</p> <p>Transaction Processing: Introduction-Transaction State, Transaction properties, Concurrent Executions. Need of Serializability, Conflict vs. View Serializability, Testing for Serializability, Recoverable Schedules, Cascadeless Schedules. [2]</p>	
IV	<p>Concurrency Control: Implementation of Concurrency: Lock-based protocols, Timestamp-based protocols, Validation-based protocols, Deadlock handling, [1]</p> <p>Database Failure and Recovery: Database Failures, Recovery Schemes: Shadow Paging and Log-based Recovery, Recovery with Concurrent transactions. [1]</p>	
V	<p>Indexing and Hashing: Basic Concepts, Ordered Indices, B⁺-Tree Index Files, B-Tree Index Files, Multiple Key Access, Static Hashing, Dynamic Hashing, Comparison of Ordered Indexing and Hashing, Bitmap Indices, Index Definition in SQL. [1]</p>	

Text/References:

1. H.f. Korth and Silberschatz: Database Systems Concepts, McGraw Hill
2. Almasri and S.B. Navathe: Fundamentals of Database Systems,

RAJASTHAN TECHNICAL UNIVERSITY

Detailed Syllabus B.Tech. (Comp. Engg.) V-VI Sem. 2010-11

3. Ramakrishnan and Gehrke: Database Management System, McGraw Hill
4. C.J. Date: Data Base Design, Addison Wesley
5. Hansen and Hansen : DBM and Design, PHI

5CS5 OPERATING SYSTEMS (Common to Comp. Engg. & Info. Tech)

Class: V Sem. B.Tech.	Evaluation
Branch: Computer Engg. Schedule per Week Lectures: 3	Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]

Units	Contents of the subject
I	<p>Introduction and need of operating system, layered architecture/logical structure of operating system, Type of OS, operating system as resource manager and virtual machine, OS services, BIOS, System Calls/Monitor Calls, Firmware- BIOS, Boot Strap Loader.</p> <p>Process management- Process model, creation, termination, states & transitions, hierarchy, context switching, process implementation, process control block, Basic System calls- Linux & Windows.</p> <p>Threads- processes versus threads, threading, concepts, models, kernel & user level threads, thread usage, benefits, multithreading models.</p>
II	<p>Interprocess communication- Introduction to message passing, Race condition, critical section problem, mutual exclusion with busy waiting- disabling interrupts, lock variables, strict alteration, Peterson's solution, TSL instructions, busy waiting, sleep and wakeup calls, semaphore, monitors, classical IPC problems.</p> <p>Process scheduling- Basic concepts, classification, CPU and I/O bound, CPU scheduler- short, medium, long-term, dispatcher, scheduling:- preemptive and non-preemptive, Static and Dynamic Priority, Co-operative & Non-cooperative, Criteria/Goals/Performance Metrics, scheduling algorithms- FCFS, SJFS, shortest remaining time, Round robin, Priority scheduling, multilevel queue scheduling, multilevel feedback queue scheduling, Fair share scheduling.</p>
III	<p>Deadlock- System model, resource types, deadlock problem, deadlock characterization, methods for deadlock handling, deadlock prevention, deadlock avoidance, deadlock detection, recovery from deadlock.</p> <p>Memory management- concepts, functions, logical and physical address space, address binding, degree of multiprogramming, swapping, static & dynamic loading- creating a load module, loading, static & dynamic linking, shared libraries, memory allocation schemes- first fit, next fit, best fit, worst fit, quick fit. Free space management- bitmap, link list/free list, buddy's system, memory protection and sharing, relocation and address translation.</p>
IV	<p>Virtual Memory- concept, virtual address space, paging scheme, pure segmentation and segmentation with paging scheme hardware support and implementation details, memory fragmentation, demand paging, pre-paging, working set model, page fault frequency, thrashing, page replacement algorithms- optimal, NRU, FIFO, second chance, LRU, LRU- approximation clock, WS clock; Belady's anomaly, distance string; design issues for paging system- local versus global allocation policies, load control, page size, separate instruction and data spaces, shared pages, cleaning policy, TLB (translation look aside buffer) reach, inverted page table, I/O interlock, program structure, page fault handling, Basic idea of MM in Linux & windows.</p>
V	<p>File System- concepts, naming, attributes, operations, types, structure, file organization & access(Sequential, Direct ,Index Sequential) methods, memory mapped files, directory structures- one level, two level, hierarchical/tree, acyclic graph, general graph, file system mounting, file sharing, path name, directory operations, overview of file system in Linux & windows.</p> <p>Input/Output subsystems- concepts, functions/goals, input/output devices- block and character, spooling, disk structure & operation, disk attachment, disk storage capacity, disk scheduling algorithm- FCFS, SSTF, scan scheduling, C-scan schedule.</p>

Text/Reference Books:

1. A. Silberschatz and Peter B Galvin: Operating System Principals, Wiley India Pvt. Ltd.

RAJASTHAN TECHNICAL UNIVERSITY

Detailed Syllabus B.Tech. (Comp. Engg.) V-VI Sem. 2010-11

2. Achyut S Godbole: Operating Systems, Tata McGraw Hill
3. Tanenbaum: Modern Operating System, Prentice Hall.
4. DM Dhamdhere: Operating Systems – A Concepts Based Approach, Tata McGraw Hill
5. Charles Crowly: Operating System A Design – Oriented Approach, Tata McGraw Hill.

5CS6.1 ADVANCED DATA STRUCTURE (Common to Comp. Engg. & Info. Tech)

Class: V Sem. B.Tech.	Evaluation
Branch: Computer Engg. Schedule per Week Lectures: 3	Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]

Units	Contents of the subject
I	ADVANCED TREES: Definitions, Operations on Weight Balanced Trees (Huffman Trees), 2-3 Trees and Red- Black Trees. Dynamic Order Statistics, Interval Tree; Dictionaries.
II	MERGEABLE HEAPS: Mergeable Heap Operations, Binomial Trees, Implementing Binomial Heaps and its Operations, 2-3-4. Trees and 2-3-4 Heaps. Amortization analysis and Potential Function of Fibonacci Heap, Implementing Fibonacci Heap.
III	GRAPH THEORY DEFINITIONS: Definitions of Isomorphic Components. Circuits, Fundamental Circuits, Cut-sets. Cut- Vertices Planer and Dual graphs, Spanning Trees, Kuratovski's two Graphs. GRAPH THEORY ALGORITHMS: Algorithms for Connectedness, Finding all Spanning Trees in a Weighted Graph, Breadth First and Depth First Search, Topological Sort, Strongly Connected Components and Articulation Point. Single Min-Cut Max-Flow theorem of Network Flows. Ford-Fulkerson Max Flow Algorithms.
IV	SORTING NETWORK: Comparison network, zero-one principle, bitonic sorting and merging network sorter. Priority Queues and Concatenable Queues using 2-3 Trees. Operations on Disjoint sets and its union-find problem, Implementing Sets.
V	NUMBER THEORITIC ALGORITHM: Number theoretic notions, Division theorem, GCD, recursion, Modular arithmetic, Solving Modular Linear equation, Chinese Remainder Theorem, power of an element, Computation of Discrete Logarithms, primality Testing and Integer Factorization.

Text/References:

1. Cormen, Leiserson, Rivest: Introduction to Algorithms, Prentice Hall of India.
2. Horowitz and Sahani: Fundamental of Computer algorithms.
3. Aho A.V , J.D Ulman: Design and analysis of Algorithms, Addison Wesley
4. Brassard : Fundamental of Algorithmics, PHI.

5CS6.2 DIGITAL SIGNAL PROCESSING (Comp. Engg.)

Class: V Sem. B.Tech.	Evaluation
Branch: Computer Engg. Schedule per Week Lectures: 3	Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]

Units	Contents of the subject
I	INTRODUCTION : Discrete time signals and systems, properties of discrete time systems, Linear time invariant systems - discrete time. Properties of LTI systems and their block diagrams. Convolution, Discrete time systems described by difference equations.
II	Fourier Transform: Discrete time Fourier transform for periodic and aperiodic signals. Properties of DTFT. Z-transform: The region of convergence for the Z-transform. The Inverse Z-transform. Properties of Z transform.
III	SAMPLING: Mathematical theory of sampling. Sampling theorem. Ideal & Practical sampling. Interpolation technique for the reconstruction of a signal from its samples. Aliasing. Sampling in freq. domain. Sampling of discrete time signals.
IV	THE DISCRETE FOURIER TRANSFORMS (DFT): Properties of the DFT, Linear Convolution using DFT. Efficient computation of the DFT: Decimation-in-Time and Decimation-in frequency FFT Algorithms.
V	FILTER DESIGN TECHNIQUES: Structures for discrete-time systems- Block diagram and signal flow graph representation of LCCD (LCCD – Linear Constant Coefficient Difference) equations, Basic structures for IIR and FIR systems, Transposed forms. Introduction to filter Design: Butterworth & Chebyshev.IIR filter design by impulse invariance & Bilinear transformation. Design of FIR filters by Windowing: Rectangular, Hamming & Kaiser.

Text/References:

1. Oppenheim, Discrete-Time Signal Processing, 2/e, Pearson Education
2. Proakis, Digital Signal Processing, 4/e, Pearson Education
3. S.K.Mitra, Digital Signal Processing, 2/e, Tata McGraw Hill

5CS6.3 INFORMATION THEORY & CODING (Comp. Engg.)

Class: V Sem. B.Tech.	Evaluation
Branch: Computer Engg. Schedule per Week Lectures: 3	Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]

Units	Contents of the subject
I	Introduction to information theory. Uncertainty, Information and Entropy, Information measures for continuous random variables, source coding theorem. Discrete Memory less channels, Mutual information, Conditional entropy.
II	Source coding schemes for data compaction: Prefix code, Huffman code, Shanon-Fane code & Hempel-Ziv coding channel capacity. Channel coding theorem. Shannon limit.
III	Linear Block Code: Introduction to error correcting codes, coding & decoding of linear block code, minimum distance consideration, conversion of non systematic form of matrices into systematic form.
IV	Cyclic Code: Code Algebra, Basic properties of Galois fields (GF) polynomial operations over Galois fields, generating cyclic code by generating polynomial, parity check polynomial. Encoder & decoder for cyclic codes.
V	Convolutional Code: Convolutional encoders of different rates. Code Tree, Trllis and state diagram. Maximum likelihood decoding of convolutional code: The viterbi Algorithm fee distance of a convolutional code.

Text/References

1. Digital Communication, Simon Haykin, Wiley.

5CS7 DATABASE LAB (Common to Comp. Engg. & Info. Tech)

Class: V Sem. B.Tech.	Evaluation
Branch: Computer Engg. Schedule per Week Practical Hrs.: 3	Examination Time = Four (4) Hours Maximum Marks = 100 [Sessional/Mid-term (60) & End-term (40)]

Objectives: At the end of the semester, the students should have clearly understood and implemented the following:

1. Stating a database design & application problem.
2. Preparing ER diagram
3. Finding the data fields to be used in the database.
4. Selecting fields for keys.
5. Normalizing the database including analysis of functional dependencies.
6. Installing and configuring the database server and the front end tools.
7. Designing database and writing applications for manipulation of data for a stand alone and shared data base including concepts like concurrency control, transaction roll back, logging, report generation etc.
8. Get acquainted with SQL.

In order to achieve the above objectives, it is expected that each students will chose one problem. The implementation shall being with the statement of the objectives to be achieved, preparing ER diagram, designing of database, normalization and finally manipulation of the database including generation of reports, views etc. The problem may first be implemented for a standalone system to be used by a single user.

All the above steps may then be followed for development of a database application to be used by multiple users in a client server environment with access control. The application shall NOT use web techniques.

One exercise may be assigned on creation of table, manipulation of data and report generation using SQL.

Suggested Tools:

For standalone environment, Visual FoxPro or any similar database having both the database and manipulation language may be used.

For multi-user application, MYSql is suggested. However, any other database may also be used. For front end, VB.Net, Java, VB Script or any other convenient but currently used by industry may be chosen.

Indicative List of exercises:

1. Student information system for your college.
2. Student grievance registration and redressal system.

3. A video library management system for a shop.
4. Inventory management system for a hardware/ sanitary item shop.
5. Inventory management system for your college.
6. Guarantee management system for the equipments in your college.

5CS8 SYSTEM DESIGNS in UML LAB (Comp. Engg.)

Class: V Sem. B.Tech.	Evaluation
Branch: Computer Engg. Schedule per Week Practical Hrs : 3	Examination Time = Four (4) Hours Maximum Marks = 75 [Sessional/Mid-term (45) & End-term (30)]

Objectives:

1. The students shall be able to use following modules of UML for system description, implementation and finally for product development.
 - Capture a business process model.
 - The User Interaction or Use Case Model - describes the boundary and interaction between the system and users. Corresponds in some respects to a requirements model.
 - The Interaction or Communication Model - describes how objects in the system will interact with each other to get work done.
 - The State or Dynamic Model - State charts describe the states or conditions that classes assume over time. Activity graphs describe the workflows the system will implement.
 - The Logical or Class Model - describes the classes and objects that will make up the system.
 - The Physical Component Model - describes the software (and sometimes hardware components) that make up the system.
 - The Physical Deployment Model - describes the physical architecture and the deployment of components on that hardware architecture.

The students are expected to use the UML models, prepare necessary documents using UML and implement a system. Some hardware products like digital clock, digital camera, washing machine controller, air conditioner controller, an electronic fan regulator, an elementary mobile phone etc. may also be chosen.

The students shall be assigned one problem on software based systems and another involving software as well as hardware.

5CS9 OPERATING SYSTEMS SIMULATION LAB (Common to Comp. Engg. & Info. Tech)

Class: V Sem. B.Tech.	Evaluation
Branch: Computer Engg. Schedule per Week Practical Hrs : 3	Examination Time = Four (4) Hours Maximum Marks = 100 [Sessional/Mid-term (60) & End-term (40)]

Objectives:

- Understand the basic functions of operating systems.
- In depth knowledge of the algorithms used for implementing the tasks performed by the operating systems.
- Understand & simulate strategies used in Linux & Windows operating systems.
- Develop aptitude for carrying out research in the area of operating system.

Suggested Tools:

Operating system simulator- MOSS preferably on Linux platform (Available for free download from <http://www.ontko.com/moss/>).

Recommended Exercises:

A. Exercises shall be given on simulation of algorithms used for the tasks performed by the operating systems. Following modules of the simulator may be used:

- Scheduling
- Deadlock
- Memory Management Systems
- File system simulator

Algorithms described in the text may be assigned. The simulation results such as average latency, hit & Miss Ratios or other performance parameters may be computed.

B. One exercise shall be on simulation of algorithms reported in the recent conferences/ journals and reproducing the results reported therein.

5CS10 DIGITAL HARDWARE DESIGN LAB (Common to Comp. Engg. & Info. Tech)

Class: V Sem. B.Tech.	Evaluation
Branch: Computer Engg. Schedule per Week Practical Hrs : 3	Examination Time = Four (4) Hours Maximum Marks = 75 [Sessional/Mid-term (45) & End-term (30)]

Objectives: At the end of course, the students shall be able to

- Should be able to design datapath for digital systems
- Create a digital system using discrete digital ICs
- Design a hard wired / micro-programmed control circuit
- Simulate a digital datapath in Hardware Description Language
- Understand IC descriptions and select proper IC in a given circuit based on its timing characteristics

Suggested Methodology and tools: Hardware description language like Verilog /VHDL can be used for simulation.

The exercise shall involve design of datapath, its simulation and finally realization on breadboard. Library of digital ICs have to be built. Similarly, manuals of Digital IC families have to be placed in the laboratories for reference by students.

Suggested Exercises

- Create a microprocessor from ALU 74181. For this, the students may design a small instruction set and attach necessary registers and suitable control unit to realize a microprocessor.
- Simulate and realize a Cordic calculator.
- Simulate & realize a Four bit Adder
 - Design and simulation of a 4-bit Adder
 - VHDL/Verilog HDL (Hardware description language)
 - Interfacing 7-segment decoder
- Combinational Multiplier
 - 4x4-bit multiplier
 - Binary-to-BCD conversion
 - Timing Constraints
- CRC checksum generator & verifier
- Realizing a carry look ahead adder

6CS1 COMPUTER NETWORKS (Common to Comp. Engg. & Info. Tech)

Class: VI Sem. B.Tech.	Evaluation
Branch: Computer Engg. Schedule per Week Lectures: 3	Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]

NOTE: The first 2 lectures shall be devoted to review of the basis architectures and responsibilities of different layers.

Units	Contents of the subject
I	Network layer- Network layer design issue, routing algorithms: Optimally principle short path, flooding, Distance vector, link state, hierarchical, Broadcast routing. Congestion control: General principle of congestion control, congestion prevention policies, congestion control in Datagram subnets, load shedding, jitter control Quality of service: requirements and techniques.
II	Internetworking: Differences in networks, connecting networks, concatenated virtual circuit, connectionless internetworking, Tunneling, Internetwork routing, Fragmentation Network layer in the Internet: IPV4, IP addressing including Subnet addressing, CIDR, NAT, ICMP, OSPF, BGP, IGMP, ARP, RARP, BOOTP, DHCP(only working and purpose; packet headers etc. not included), Differences in IPV6 over IPV4
III	Transport layer: Services provided, Transport service primitives. Elements of Transport protocols: addressing, connection Establishment, connection release, Flow control & Buffering, Multiplexing, Crash Recovery, UDP, RPC, RTP. Principles of Reliable Data Transfer: Reliable data transfer over a perfectly reliable channel, Channel with bit errors and Lossy Channel with bit errors.
IV	Transport Layer in the Internet: Introduction to TCP, TCP service Model, TCP Header and segment structure, TCP connection establishment and release, transmission policy, timer management, Transactional TCP. TCP Congestion Control: Fairness, TCP delay modeling.
V	Application Layer: Service needs. Domain Name System: Name Space, resource record, name servers, resolution process, introduction to DNS poisoning. Electronic Mail: Architecture and services, MIME message formats, SMTP, POP3, IMAP. World Wide Web: Architecture, responsibilities of client and server sides, HTTP, performance enhancement. P2P File Sharing: Centralized Directory, Query flooding, exploiting heterogeneity.

Text/References:

1. Tanenbaum; Computer Network, 4th Ed., Pearson.
2. Kurose; Computer Networking, 3rd Ed., Pearson.
3. Peterson, Davie; Computer Networks, 4rd Ed., ELSEVIER

6CS2 DESIGN AND ANALYSIS OF ALGORITHMS (Common to Comp. Engg. & Info. Tech.)

Class: VI Sem. B.Tech.		Evaluation
Branch: Computer Engg. Schedule per Week Lectures: 3		Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]
Units	Contents of the subject	
I	BACKGROUND: Review of Algorithm Complexity, Order Notations: definitions and calculating complexity. DIVIDE AND CONQUER METHOD: Binary Search, Merge Sort, Quick sort and Strassen's matrix multiplication algorithms. GREEDY METHOD: Knapsack Problem, Job Sequencing, Optimal Merge Patterns and Minimal Spanning Trees.	
II	DYNAMIC PROGRAMMING: Matrix Chain Multiplication. Longest Common Subsequence and 0/1 Knapsack Problem. BRANCH AND BOUND: Traveling Salesman Problem and Lower Bound Theory. Backtracking Algorithms and queens problem.	
III	PATTERN MATCHING ALGORITHMS: Naïve and Rabin Karp string matching algorithms, KMP Matcher and Boyer Moore Algorithms. ASSIGNMENT PROBLEMS: Formulation of Assignment and Quadratic Assignment Problem.	
IV	RANDOMIZED ALGORITHMS. Las Vegas algorithms, Monte Carlo algorithms, randomized algorithm for Min-Cut, randomized algorithm for 2-SAT. Problem definition of Multicommodity flow, Flow shop scheduling and Network capacity assignment problems.	
V	PROBLEM CLASSES NP, NP-HARD AND NP-COMPLETE: Definitions of P, NP-Hard and NP-Complete Problems. Decision Problems. Cook's Theorem. Proving NP-Complete Problems - Satisfiability problem and Vertex Cover Problem. Approximation Algorithms for Vertex Cover and Set Cover Problem. PROBLEM CLASSES NP, NP-HARD AND NP-COMPLETE: Definitions of P, NP-Hard and NP-Complete Problems. Decision Problems. Cook's Theorem. Proving NP-Complete Problems - Satisfiability problem and Vertex Cover Problem. Approximation Algorithms for Vertex Cover and Set Cover Problem.	

Text/References:

1. Cormen, Leiserson, Rivest: Introduction to Algorithms, Prentice Hall of India.
2. Horowitz and Sahani: Fundamental of Computer algorithms.
3. Aho A.V , J.D Ulman: Design and analysis of Algorithms, Addison Wesley

6CS3 THEORY OF COMPUTATION (Common to Comp. Engg. & Info. Tech)

Class: VI Sem. B.Tech.	Evaluation
Branch: Computer Engg. Schedule per Week Lectures: 3, Tutorial:1	Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]

Units	Contents of the subject
I	Finite Automata & Regular Expression: Basic Concepts of finite state system, Deterministic and non-deterministic finite automation and designing regular expressions, relationship between regular expression & Finite automata minimization of finite automation mealy & Moore Machines.
II	Regular Sets of Regular Grammars: Basic Definition of Formal Language and Grammars. Regular Sets and Regular Grammars, closure proportion of regular sets, Pumping lemma for regular sets, decision Algorithms for regular sets, Myhell_Nerod Theory & Organization of Finite Automata.
III	Context Free Languages& Pushdown Automata: Context Free Grammars – Derivations and Languages – Relationship between derivation and derivation trees – ambiguity – simplification of CEG – Greiback Normal form – Chomsky normal forms – Problems related to CNF and GNF Pushdown Automata: Definitions – Moves – Instantaneous descriptions – Deterministic pushdown automata – Pushdown automata and CFL - pumping lemma for CFL - Applications of pumping Lemma.
IV	Turing Machines: Turing machines – Computable Languages and functions – Turing Machine constructions – Storage in finite control – multiple tracks – checking of symbols – subroutines – two way infinite tape. Undecidability: Properties of recursive and Recursively enumerable languages – Universal Turing Machines as an undecidable problem – Universal Languages – Rice's Theorems.
V	Linear bounded Automata Context Sensitive Language: Chomsky Hierarchy of Languages and automata, Basic Definition & descriptions of Theory & Organization of Linear bounded Automata Properties of context-sensitive languages

Text/References

1. Aho, Hopcroft and Ullman, Introduction to Automata Theory, Formal Languages and Computation, Narosa
2. Cohen, Introduction to Computer Theory, Addison Wesley.
3. Papadimitriou, Introduction to Theory of Computing, Prentice Hall.

6CS4 PROGRAMMING IN JAVA (Common to Comp. Engg. & Info. Tech)

Class: VI Sem. B.Tech.		Evaluation
Branch: Computer Engg. Schedule per Week Lectures: 3		Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]
Units	Contents of the subject	
I	<p>JAVA: Introduction to Object Orientated Programming, Abstraction, Object Oriented Programming Principles, Features of JAVA, Introduction to Java byte code, Java Virtual machine.</p> <p>PROGRAM ELEMENTS: Primitive data types, variables, assignment, arithmetic, short circuit logical operators, Arithmetic operators, bit wise operators, relational operators, Boolean logic operators, the assignment operators, operator precedence, Decision and control statements, arrays.</p>	
II	<p>CONTROL STATEMENTS: Java's Selection Statements, if statement, switch statement, Iteration Statements, while, do-while, for, for-each, Nested Loops, Jump Statements, Using break, Using continue, return.</p> <p>OBJECTS AND CLASSES: Objects, constructors, returning and passing objects as parameter, Nested and inner classes, Single and Multilevel Inheritance, Extended classes, Access Control, usage of super, Overloading and overriding methods, Abstract classes, Using final with inheritance.</p>	
III	<p>PACKAGE AND INTERFACES: Defining package, concept of CLASSPATH, access modifiers, importing package, Defining and implementing interfaces.</p> <p>STRING HANDLING: String constructors, special string operations, character extraction, searching and comparing strings, string Buffer class.</p>	
IV	<p>EXCEPTION HANDLING: Exception handling fundamentals, Exception types, uncaught exceptions, try, catch and multiple catch statements. Usage of throw, throws and finally .FILE HANDLING: I/O streams, File I/O.</p>	
V	<p>CONCURRENCY: Processes and Threads, Thread Objects, Defining and Starting a Thread, Pausing Execution with Sleep, Interrupts, Joins, Synchronization. APPLET: Applet Fundamentals, using paint method and drawing polygons.</p>	

Text/References

1. Herbert Schildt: JAVA 2 - The Complete Reference, TMH, Delhi
2. Deitel: How to Program JAVA, PHI
3. U.K. Chakraborty and D.G. Dastidar: Software and Systems – An Introduction, Wheeler Publishing, Delhi.
4. Joseph O'Neil and Herb Schildt: Teach Yourself JAVA, TMH, Delhi.

6CS5 Embedded System Design (Comp. Engg.)

Class: VI Sem. B.Tech.		Evaluation
Branch: Computer Engg. Schedule per Week Lectures: 3		Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]
Units	Contents of the subject	
I	<p>Overview of Embedded System: Definition, Design Challenges and Characteristics, Categories and Requirements of Embedded Systems. Embedded Hardware and Software Development environment.</p> <p>Difference between microprocessor, microcontroller and DSP. General capability of microcontroller; microcontrollers in embedded systems.</p> <p>Suitability/selection of a microcontroller based on - Cost, Performance, Power dissipation and architecture- 8-bit, 16-bit, 32-bit.</p> <p>Concepts of system-on-chip, How modern-day system-on-chip (SoC) microcontrollers can implement a whole signal chain.</p>	
II	<p>Microcontroller MSP 430: RISC architecture, Instruction Sets and Addressing modes, I/O ports, counter and timers, interrupts and interrupt structure, Assembly Language Programming and Compiler-friendly features. Clock system, Memory subsystem. Key differentiating factors between different MSP 430 families.</p> <p>Interfacing: I/O Interfacing, understanding the multiplexing scheme of MSP 430 pins, LED, LCD, seven segment display, real time clock and Keyboard Interfacing. ADC, DAC, and Sensor Interfacing, Interfacing to External Memory, Interfacing to Stepper Motor.</p>	
III	<p>ARM Fundamentals: Registers, Current Program Status Register, Pipeline, Exceptions, Interrupts, and the Vector table, Core Extensions, Instruction Set, Introduction to Thumb Instruction Set (Writing Programs not included in the theory Course)</p>	
IV	<p>8051 Microcontroller: Architecture, Addressing modes, I/O Port Programming, Single bit instructions and Programming, Interrupt Programming</p>	
V	<p>Performance Issues of an Embedded System: CPU performance–CPU Power Consumption, Analysis and Optimization of CPU Power Consumption, program execution time–Analysis, low-power modes (sleep modes), clock request feature, low power programming and interrupts.</p> <p>Applications of Embedded systems: Energy meters, Smoke detectors, Data acquisition system, wired sensor network, and wireless sensor networks with Chipcon RF interface.</p>	

Text Books:

1. John Davies, MSP430 Microcontroller Basics, Elsevier, 2008.
2. Andrew N. Sloss et.al. ARM System Developers Guide, ELSEVIER
3. Muhammad Ali Mazidi et.al., The 8051 Microcontroller & Embedded Systems, Pearson
4. Embedded System Design, A Unified Hardware/Software Introduction, Frank Vahid / Tony Givargis, 2006 reprint, John Wiley Student Edition.

5. An Embedded Software Primer, David .E. Simon, Fourth Impression 2007, Pearson Education.

Text/References:

1. Embedded Microcomputer Systems, Valvano, Thomson.
2. Performance Issues of an Embedded System <http://embedded.com>.
3. Computers As Components: Principles of Embedded Computing System Design, 2nd Edition. Morgan Kauffman.
4. Multi-core Embedded Systems,” edited by Prof. Georgios Kornaros , CRC Press
5. MSP430 Teaching CD-ROM, Texas Instruments, 2008 (Includes Power Point foils for Instructors.
6. can be requested from <http://www.uniti.in>)
7. Documentation from www.msp430.com
8. Course materials on MSP 430 available from Rice University's "Connexions" system (<http://cnx.org>)

6CS6.1 ADVANCE TOPICS IN OPERATING SYSTEMS (Common to Comp. Engg. & Info. Tech)

Class: VI Sem. B.Tech.	Evaluation
Branch: Computer Engg. Schedule per Week Lectures: 3	Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]

Units	Contents of the subject
I	<p>Operating system structures – policies & mechanism, Structures- monolithic, layered, virtual machines, micro kernel, exokernels, client- server model. Examples from Linux & Windows.</p> <p>Threads Advance Concepts– Libraries- Pthreads, win32 threads, Java threads, Introduction to threading issues, system calls, cancellation, signal handling, thread pool, thread specific data, window threads, Linux threads, Solaris Threads.</p> <p>Message Passing System – Need of Message Passing Systems, design issues, naming, synchronization, Implementation–buffering and delivery; mailboxes; RPC & RMI. Examples Systems – Linux, Windows.</p>
II	<p>File System- file system layouts, file system implementation, contiguous allocation, link list allocation, indexed allocation, file allocation table, virtual file system, directory implementation- linear list and hash table. File System reliability and integrity.</p> <p>I/O system: device drivers/ controllers, busses and interfaces- USB, IDE, SCSI, IEEE1394, RAID system, disk caching and buffering, disk management-disk formatting, RAID Structure, boot block, bad block, swap-space management.</p> <p>System Security: Security Problems, Program Threats, System Network Threats, Cryptography as a Security Tool, User Authentication, Implementing Security Defenses, Firewalling to Protect Systems and Network, Computer Security Classifications. Overview of security in Windows. [4]</p>
III	<p>The Linux OS: Unix Vs Linux, Design Principles, Kernel Structure, components Kernel Modules, Shell- usage, types; An overview of- Process Management, Thread Management and Scheduling, Memory Management, Process Scheduling in Linux, File System structure & implementation, I/O Management, Network File System, Inter-process Communications, Booting and login process, security.[3]</p>
IV	<p>The Window OS: Design Principles, System Components- Hardware Abstraction layer, Kernel, Executives; Environmental Subsystems- MS-DOS Environment, 16-bit Windows Environment, Win32 API, POSIX subsystem; Exception and Interrupts; An overview of-memory management, process management and thread; Process Scheduling in Windows; File Systems: Internal Layout, recovery, Volume Management and Fault Tolerance, FAT and NTFS, Security features, window registry, OS organizations.[3]</p>

V	<p>Multiprocessor Operating Systems: Architecture of Multiprocessor Systems, Overview of Multiprocessor OS, Kernel Structure and Multiprocessing support in Linux & Windows, Process Synchronization- Queued Lock, Spin Lock, Sleep Lock; Process Scheduling.</p> <p>Multimedia Operating System- Introduction to Multimedia & Data Compression- concepts, common graphics file formats, common audio file formats; Video server, Process management- real time scheduling; Multimedia file systems, Multimedia file storage mechanisms, Video server organization.[2]</p> <p>Mobile Operating System- Windows CE, Palm OS, Symbian OS, JAVA card, Multos.</p>
---	---

Text/Reference Books:**Text/Reference Books:**

1. DM Dhamdhare: Operating Systems – A Concepts Based Approach, Tata McGraw Hill
2. Achyut S Godbole: Operating Systems, Tata McGraw Hill
3. Tanenbaum: Modern Operating System, Prentice Hall
4. A. Silberschatz and Peter B Galvin: Operating System Principals, Wiley India Pvt. Ltd.
5. Charles Crowly: Operating System A Design – Oriented Approach, Tata McGraw Hill.
6. Bach, Design of Unix Operating Systems.

6CS6.2 ARTIFICIAL INTELLIGENCE (Comp. Engg.)

Class: VII Sem. B.Tech.	Evaluation
Branch: Computer Engg. Schedule per Week Lectures: 3	Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]

Units	Contents of the subject
I	Meaning and definition of artificial intelligence, Various types of production systems, Characteristics of production systems, Study and comparison of breadth first search and depth first search. Techniques, other Search Techniques like hill Climbing, Best first Search. A* algorithm, AO* algorithms etc, and various types of control strategies.
II	Knowledge Representation, Problems in representing knowledge, knowledge representation using propositional and predicate logic, comparison of propositional and predicate logic, Resolution, refutation, deduction, theorem proving, inferencing, monotonic and nonmonotonic reasoning.
III	Probabilistic reasoning, Baye's theorem, semantic networks scripts schemas, frames, conceptual dependency and fuzzy logic, forward and backward reasoning.
IV	Game playing techniques like minimax procedure, alpha-beta cut-offs etc, planning, Study of the block world problem in robotics, Introduction to understanding and natural languages processing.
V	Introduction to learning, Various techniques used in learning, introduction to neural networks, applications of neural networks, common sense, reasoning, some example of expert systems.

Text Books & References:

1. Artificial Intelligence: Elaine Rich, Kevin Knight, Mc-Graw Hill.
2. Introduction to AI & Expert System: Dan W. Patterson, PHI.
3. Artificial Intelligence by Luger (Pearson Education)
4. Russel & Norvig, Artificial Intelligence: A Modern Approach, Prentice-Hall

6CS6.3 HUMAN COMPUTER INTERFACE (Common to Comp. Engg. & Info. Tech)

Class: VI Sem. B.Tech.	Evaluation
Branch: Comp. Engg. Schedule per Week Lectures: 3	Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]

Units	Contents of the subject
I	The Human: input-output channels, Human memory, thinking, emotions, individual differences, psychology and the design of interactive systems. The Computer: Text entry devices with focus on the design of key boards, positioning, pointing and drawing, display devices. The Interaction: Models of interaction, ergonomics, interaction styles, elements of WIMP interfaces, interactivity, experience, engagement and fun. Paradigms for Interaction.
II	Design Process: The process of design, user focus, scenarios, navigation design screen design and layout, iteration & prototyping. Usability Engineering Design rules: Principles to support usability, standards, guidelines, rules and heuristics, HCI patterns.
III	Evaluation Techniques: Definition and goals of evaluation, evaluation through expert analysis and user participation, choosing an evaluation method. User support, requirement, approaches, adaptive help systems, designing user support systems
IV	Cognitive methods: Goals and task hierarchies, linguistic models, challenges of display based systems, physical and device models, cognitive architectures.
V	Communications and collaborations models: Face to Face communication, conversations, Text based communication, group working. Task Analysis: Differences between task analysis and other techniques, task decomposition, knowledge based analysis, ER based analysis, sources of information and data collection, use of task analysis.

Text/References:

1. Human Computer Interaction; Alan Dix et.al, 3rd ed., Pearson

6CS7 JAVA PROGRAMMING LAB (Comp. Engg. Tech)

Class: VI Sem. B.Tech.	Evaluation
Branch: Computer Engg. Schedule per Week Practical Hrs.: 3	Examination Time = Four (4) Hours Maximum Marks = 100 [Sessional/Mid-term (60) & End-term (40)]

Objectives: At the end of the semester, the students should have clearly understood and implemented the following:

- 1. Develop an in depth understanding of programming in Java:** data types, variables, operators, operator precedence, Decision and control statements, arrays, switch statement, Iteration Statements, Jump Statements, Using break, Using continue, return.
- 2. Write Object Oriented programs in Java:** Objects, Classes constructors, returning and passing objects as parameter, Inheritance, Access Control, Using super, final with inheritance Overloading and overriding methods, Abstract classes, Extended classes.
- 3. Develop understanding to developing packages & Interfaces in Java:** Package, concept of CLASSPATH, access modifiers, importing package, Defining and implementing interfaces.
- 4. Develop understanding to developing Strings and exception handling:** String constructors, special string operations, character extraction, searching and comparing strings, string Buffer class. Exception handling fundamentals, Exception types, uncaught exceptions, try, catch and multiple catch statements. Usage of throw, throws and finally.
- 5. Develop applications involving file handling:** I/O streams, File I/O.
- 6. Develop applications involving concurrency:** Processes and Threads, Thread Objects, Defining and Starting a Thread, Pausing Execution with Sleep, Interrupts, Joins, and Synchronization.
- 7. Develop applications involving Applet:** Applet Fundamentals, using paint method and drawing polygons.

It is expected that each laboratory assignments to given to the students with an aim to In order to achieve the above objectives

Indicative List of exercises:

7. Programs to demonstrate basic concepts e.g. operators, classes, constructors, control & iteration statements, recursion etc. such as complex arithmetic, matrix arithmetic, tower of Hanoi problem etc.
8. Development of programs/projects to demonstrate concepts like inheritance, exception handling, packages, interfaces etc. such as application for electricity department, library management, ticket reservation system, payroll system etc.
9. Development of a project to demonstrate various file handling concepts.
10. Development of a project to demonstrate various applet concepts.

6CS8 SHELL PROGRAMMING LAB (Info. Tech)

Class: VI Sem. B.Tech.	Evaluation
Branch: I.T. Schedule per Week Practical Hrs : 2	Examination Time = Four (4) Hours Maximum Marks = 50 [Sessional/Mid-term (30) & End-term (20)]

S. No.	List of Experiments
1.	Use of Basic Unix Shell Commands: <i>ls, mkdir, rmdir, cd, cat, banner, touch, file, wc, sort, cut, grep, dd, dfspace, du, ulimit.</i>
2.	Commands related to inode, I/O redirection and piping, process control commands, mails.
3.	Shell Programming: Shell script exercises based on following (i) Interactive shell scripts (ii) Positional parameters (iii) Arithmetic (iv) if-then-fi, if-then-else-fi, nested if-else (v) Logical operators (vi) else + if equals elif, case structure (vii) while, until, for loops, use of break (viii) Metacharacters (ix) System administration: disk management and daily administration
4.	Write a shell script to create a file in \$USER /class/batch directory. Follow the instructions (i) Input a page profile to yourself, copy it into other existing file; (ii) Start printing file at certain line (iii) Print all the difference between two file, copy the two files at \$USER/CSC/2007 directory. (iv) Print lines matching certain word pattern.
5.	Write shell script for- (i) Showing the count of users logged in, (ii) Printing Column list of files in your home directory (iii) Listing your job with below normal priority (iv) Continue running your job after logging out.
6.	Write a shell script to change data format. Show the time taken in execution of this script
7.	Write a shell script to print files names in a directory showing date of creation & serial number of the file.
8.	Write a shell script to count lines, words and characters in its input(do not use wc).
9.	Write a shell script to print end of a Glossary file in reverse order using Array. (Use awk tail)
10.	Write a shell script to check whether Ram logged in, Continue checking further after every 30 seconds till success.
11.	Write a shell script to compute gcd lcm & of two numbers. Use the basic function to find gcd & lcm of N numbers.
12.	Write a shell script to find whether a given number is prime. Take a large number such as 15 digits or higher and use a proper algorithm.

Ref: UNIX Shell programming, By Stephen G. Kochan, Patrick H. Wood

6CS9 DESIGN AND ANALYSIS OF ALGORITHMS (Common to Comp. Engg. & Info. Tech)

Class: VI Sem. B.Tech.	Evaluation
Branch: Comp. Engg. Schedule per Week Practical Hrs : 3	Examination Time = Four (4) Hours Maximum Marks = 100 [Sessional/Mid-term (60) & End-term (40)]

Objectives: Upon successful completion of this course, students should be able to:

- Prove the correctness and analyze the running time of the basic algorithms for those classic problems in various domains;
- Apply the algorithms and design techniques to solve problems;
- Analyze the complexities of various problems in different domains.

Suggested Tools: For implementation and estimation of running time on various sizes of input(s) or output(s) as the case may be, Linux platform is suggested.

Suggested Exercises:

- A. It is expected that teachers will assign algorithms to the students for estimation of time & space complexity. Algorithms reported in various research journals may be chosen by the teachers.
- B. Problem on designing algorithms to meet complexity constraints may be assigned. For example, a problem on design, analysis and implementation for transposing a sparse matrix requiring not more than one pass from the original matrix may be assigned.
- C. **A guide to such problems is given below:**
 1. **Exploring a Binary Heap:** Consider a binary heap containing n numbers (the root stores the greatest number). You are given a positive integer $k < n$ and a number x . You have to determine whether the k^{th} largest element of the heap is greater than x or not. Your algorithm must take $O(k)$ time. You may use $O(k)$ extra storage.
 2. **Merging two search trees:** You are given two height balanced binary search trees T and T' , storing m and n elements respectively. Every element of tree T is smaller than every element of tree T' . Every node u also stores height of the subtree rooted at it. Using this extra information how can you merge the two trees in time $O(\log m + \log n)$ (preserving both the height balance and the order)?
 3. **Complete binary tree as an efficient data-structure:** You are given an array of size n (n being a power of two). All the entries of the array are initialized to zero. You have to perform a sequence of the following online operations :
 1. (i) *Add*(i,x) which adds x to the entry $A[i]$.
 2. (ii) Report *sum*(i,j) = sum of the entries in the array from indices i to j for any $0 < i < j \leq n$.

It can be seen easily that we can perform the first operation in $O(1)$ time whereas the second operation may cost $O(n)$ in worst case. Your objective is to perform these operations efficiently. Give a data-structure which will guarantee $O(\log n)$ time per operation.

4. Problems on Amortized Analysis

- a. Delete-min in constant time!!! Consider a binary heap of size n , the root storing the smallest element. We know that the cost of insertion of an element in the heap is $O(\log n)$ and the cost of deleting the smallest element is also $O(\log n)$. Suggest a valid potential function so that the amortized cost of insertion is $O(\log n)$ whereas amortized cost of deleting the smallest element is $O(1)$.
- b. Implementing a queue by two stack
- c. Show how to implement a queue with two ordinary stacks so that the amortized cost of each Enqueue and each Dequeue operation is $O(1)$.

5. Computing a spanning tree having smallest value of largest edge weight:

Describe an efficient algorithm that, given an undirected graph G , determines a spanning tree of G whose largest edge weight is minimum over all spanning trees of G .

6. Shortest Path Problems:

i. From a subset of vertices to another subset of vertices

- a. Given a directed graph $G(V,E)$, where edges have nonnegative weights. S and D are two disjoint subsets of the set of vertices. Give an $O(|V| \log |V| + |E|)$ time algorithm to find the shortest path among the set of paths possible from any node in S to any node in D .

ii. Paths in Directed Acyclic Graph

a. Counting the number of paths

Given two nodes u, v in a directed acyclic graph $G(V,E)$. Give an $O(|E|)$ time algorithm to count all the paths from u to v .

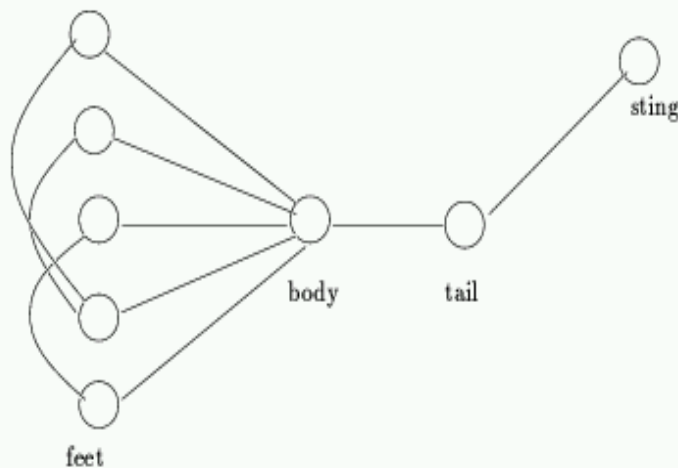
b. Path passing through a subset of nodes

Given two nodes u, v and a set of vertices w_1, w_2, \dots, w_k in a directed acyclic graph $G(V,E)$. Give an $O(|E|)$ time algorithm to output a path (if exists) from u to v which passes through each of the nodes w_1, \dots, w_k . If there is no such path then your algorithm must report that "no such path exists".

7. Searching for a friend:

You are standing at a crossing from where there emerge four roads extending to infinity. Your friend is somewhere on one of the four roads. You do not know on which road he is and how far he is from you. You have to walk to your friend and the total distance traveled by you must be at most a constant times the actual distance of your friend from you. In terminology of algorithms, you should traverse $O(d)$ distance, where d is the distance of your friend from you.

8. **A simple problem on sorted array:** Design an $O(n)$ -time algorithm that, given a real number x and a sorted array S of n numbers, determines whether or not there exist two elements in S whose sum is exactly x .
9. **Finding the decimal dominant in linear time:** You are given n real numbers in an array. A number in the array is called a decimal dominant if it occurs more than $n/10$ times in the array. Give an $O(n)$ time algorithm to determine if the given array has a decimal dominant.
10. **Finding the first one:** You are given an array of infinite length containing zeros followed by ones. How fast can you locate the first one in the array?
11. **Searching for the Celebrity:** Celebrity is a person whom everybody knows but he knows nobody. You have gone to a party. There are total n persons in the party. Your job is to find the celebrity in the party. You can ask questions of the form Does Mr. X know Mr. Y ? You will get a binary answer for each such question asked. Find the celebrity by asking only $O(n)$ questions.
12. **Checking the Scorpion:** An n -vertex graph is a *scorpion* if it has a vertex of degree 1 (the sting) connected to a vertex of degree two (the tail) connected to a vertex of degree $n-2$ (the body) connected to the other $n-3$ (the feet). Some of the feet may be connected to other feet. Design an algorithm that decides whether a given adjacency matrix represents a scorpion by examining only $O(n)$ entries.



13. **Endless list:** You are having a pointer to the head of singly linked list. The list either terminates at null pointer or it loops back to some previous location (not necessarily to the head of the list). You have to determine whether the list loops back or ends at a null location in time proportional to the length of the list. You can use at most a constant amount of extra storage.
14. **Nearest Common Ancestor:** Given a rooted tree of size n . You receive a series of online queries: "Give nearest common ancestor of u, v ". Your objective is to preprocess the tree in $O(n)$ time to get a data structure of size $O(n)$ so that you can answer any such query in $O(\log n)$ time.

6CS10 Embedded System Design Lab.(Comp. Engg.)

Class: VI Sem. B.Tech.	Evaluation
Branch: Comp. Engg. Schedule per Week Practical Hrs : 3	Examination Time = Four (4) Hours Maximum Marks = 100 [Sessional/Mid-term (60) & End-term (40)]

Course Objectives

Upon successful completion of the course, students will be able to design simple embedded systems and develop related software. Students also learn to work in a team environment and communicate the results as written reports and oral presentations.

Suggested Microcontroller Platform: Texas Instruments MSP430, ARM 9, 68HC12, 8051.

It is assumed that there are 14 weeks in the semester and about 5 to 6 experiments will be carried out. More experiments are provided to bring in variation.

Experiment #0

Get familiar with the microcontroller kit and the development software. Try the sample programs that are supplied to get familiar with the Microcontroller.

Experiment #1

a) Blink an LED which is connected to your microcontroller using the built-in *timer* in the microcontroller. Assume that the LED should be on for x milliseconds and off for y milliseconds; assume that these values are stored in memory locations X and Y . We should be able to change the value of x and y and rerun the program.

b) Consider an alternate way to program this application. Here, the microcontroller turns the LED on and waits in a busy loop to implement a delay of x milliseconds. Then it turns the LED off and waits in a busy loop to implement a delay of y milliseconds. How do you compare these two solutions?

Experiment #2

Assume that in Experiment #1, the values of x and y have been chosen to be 200 and 500 respectively. When the LED blinking program runs, pressing a key on the keyboard should generate an interrupt to the microcontroller. If the key that has been pressed is a numeric key, the value of x and y must be interchanged by the interrupt service routine. If the key that has been pressed is not a numeric key, then the LED must be turned off for 2 seconds before resuming the blinking.

Experiment #3

If your microcontroller kit has an LCD interface, write a program to display a character string on the LCD. Assume that the string is stored at a location

STRING and consists of alphanumeric characters. The string is null-terminated. Modify your program to scroll the displayed string from left to right.

Experiment #4

Modern microcontrollers usually have an in-built Digital-to-Analog and Analog-to-Digital converter. Use the built-in DAC to generate voltage waveforms such as (a) pulse train (b) triangular waveform (c) sinusoidal waveform. Observe these waveforms on an oscilloscope.

Experiment #5

Your microcontroller may have a built-in temperature sensor. If not, interface an external temperature sensor to the microcontroller. Write a program to take several measurements of temperature at regular intervals and display the average temperature on the LCD display. Test if the readings change when the ambient temperature changes.

Experiment #6

Your microcontroller may have a built-in ADC. Build a voltmeter that can measure stable voltages in a certain range. The measured value must be displayed on the LCD display. Measure the same voltage using a multimeter and record the error in measurement. Tabulate the error for several values of the voltage.

Experiment #7

Build a simple security device based on the microcontroller kit. Interface an external motion sensor to the microcontroller. An alarm must be generated if motion is sensed in a specified region. There must be a provision to record the time at which the intrusion was detected. Similarly, there must be a provision to turn the alarm off by pressing a key.

Experiment #8

A voltage waveform $v(t)$ is available as an input to the microcontroller. We must continuously check the waveform and record the maximum value of the waveform and display the maximum value on the LCD display. Test the program by using a DC supply to generate $v(t)$ and varying the DC value.