Punjab Technical University

1. NAME OF DEPTT./CENTER: **Computer Science Engineering**
2. subject code: **BTCS 301 Course Title: Computer Architecture**
3. Contact Hours **L:3 T:2 P:1**
4. Examination Duration(Hrs): **Theory** 3 **Practial** 1
5. credits: 2
6. semester **3**
7. **Objectives : This subject gave the knowledge about computer components.**

- **Register Transfer and Microoperations** Register transfer language & operations, arithmetic microoperations, logic microoperations, shift microoperations, arithmetic logic shift unit. Design of a complete basic computer and its working. [5]
- **Basic Computer Organisation and Design** Instruction codes, Computer registers, Computer Instructions, Timing and control, Instruction Cycle, Memory reference instructions, Input Output and Interrupt, Design of basic Computer, Design of Accumulator Logic. [6]
- **Design of Control Unit** Control memory, design of control unit microprogrammed, hardwired, and their comparative study. [3]
- **Central Processing Unit** General Register Organisation, Stack Organisation, Instruction formats, Addressing Modes, Data transfer and manipulations, Program control, RISC and CISC architecture. [6]
- **Input-Output Organisation** Peripheral devices, IO Interface, asynchronous data transfer, modes of transfer, priority interrupt, DMA, IO processor, serial communication. [5]
- **Memory Organisation** Memory hierarchy, main memory, auxiliary memory, associative memory, cache memory, virtual memory, memory management hardware. [6]
- **Advanced concepts of Computer Architecture** Concept of pipeline, Arithmetic pipeline, Instruction , vector processors and array processors. Introduction to parallel processing, Interprocessor communication & synchronization. [5]
- **Sets, relations and functions** Introduction, Combination of Sets, ordered pairs, proofs of general identities of sets, relations, operations on relations, properties of relations and functions, Hashing Functions, equivalence relations, compatibility relations, partial order relations. [7]
- **Combinatorial Mathematics** Basic counting principles Permutations and combinations Inclusion and Exclusion Principle Recurrence relations, Generating Function, Application. [7]
- **Monoids and Groups** Groups Semigroups and monoids Cyclic semigraphs and submonoids, Subgroups and Cosets. Congruence relations on semigroups. Morphisms. Normal subgroups. Dihedral groups. [7]
- **Graph Theory** Graph- Directed and undirected, Eulerian chains and cycles, Hamiltonian chains and cycles Trees, Chromatic number Connectivity, Graph coloring, Plane and connected graphs, Isomorphism and Homomorphism. Applications. [3]
- **Boolean Algebra** Boolean postulates and laws De-Morgans Theorem, Principle of Duality, Boolean expression Boolean function, Minimization of Boolean expressions Sum of Products (SOP), Product of Sums (POS), Minterm, Maxterm, Canonical forms, Conversion between canonical forms, Karnaugh map Minimization, Quine-McCluskey method - Dont care conditions. [5]
- **Signal Conversions** Analog & Digital signals. A,D and D,A conversion techniques (Weighted type, R-2R Ladder type, Counter Type, Dual Slope type, Successive Approximation type). [5]

- **Stacks** Sequential and linked representations, operations on stacks, application of stacks such as parenthesis checker, evaluation of postfix expressions, conversion from infix to postfix representation, implementing recursive functions. [4]

Suggested Readings/ Books

- M. Moris Mano, **Computer System Architecture,** P. Pal Choudhri.
- William Stallings, **Computer Organisation and Architecture,** P. Pal Choudhri.
- David A Patterson, **Computer Architecture,** P. Pal Choudhri.
- P. Pal Choudhri, **Computer Organisation and Design,** P. Pal Choudhri.
- Kai Hawang, **Advanced Computer Architecture,** A & C BLACK.

1. NAME OF DEPTT./CENTER: **Computer Science Engineering**
2. subject code: **BTCS302 Course Title: Discrete Structures**
3. Contact Hours **L:3 T:2 P:1**
4. Examination Duration(Hrs): **Theory** 3 **Practial** 1
5. credits: 2
6. semester **3**
7. **Objectives : The objective of this course is to provide the necessary back ground of discrete structures with particular reference to the relationships between discrete structures and their data structure counterparts including algorithm development.**

- **Rings and Boolean algebra** Rings, Subrings, morphism of rings ideals and quotient rings. Euclidean domains Integral domains and fields Boolean Algebra direct product morphisms Boolean sub-algebra Boolean Rings Application of Boolean algebra (Logic Implications, Logic Gates, Karnaugh-map) [8]

Suggested Readings/ Books

- Lipschutz, **Discrete Mathematics (Schaum series),** fknkfnks.

1. NAME OF DEPTT./CENTER: **Computer Science Engineering**
2. subject code: **BTCS303 Course Title: Digital Circuits & Logic Design**
3. Contact Hours **L:3 T:2 P:1**
4. Examination Duration(Hrs): **Theory** 3 **Practial** 1
5. credits: 2
6. semester **3**
7. **Objectives : Demonstrate the operation of simple digital gates, identify the symbols, develop the truth table for those gates; combine simple gates into more complex circuits; change binary, hexadecimal, octal numbers to their decimal equivalent an vice versa, demonstrate the operation of a flip-flop. Design counters and clear the concept of shift resisters. Study different types of memories and their applications. Convert digital into analog and vice versa.**

- **Number Systems** Binary, Octal, Decimal, Hexadecimal. Number base conversions, 1s, 2s, rths complements, signed Binary numbers. Binary Arithmetic, Binary codes: Weighted BCD, Gray code, Excess 3 code, ASCII conversion from one code to another. [5]
- **Logic GATES** AND, OR, NOT, NAND, NOR, Exclusive-OR and Exclusive-NOR. Implementations of Logic Functions using gates, NAND-NOR implementations. Study of logic families like RTL, DTL, DCTL, TTL, MOS, CMOS, ECL and their characteristics. [5]
- **Combinational Circuits** Design procedure Adders, Subtractors, Serial adder,Subtractor, Parallel adder, Subtractor Carry look ahead adder, BCD adder, Magnitude Comparator, Multiplexer,Demultiplexer, encoder,decoder, parity checker, code converters. Implementation of combinational logic using MUX. [6]
- **Sequential Circuits** Flip flops SR, JK, T, D and Master slave, Excitation table, Edge triggering, Level Triggering, Realization of one flip flop using other flip flops. Asynchronous,Ripple counters, Synchronous counters, Modulo-n counter, Ring Counters. Classification of sequential circuits-Moore and Mealy, Design of Synchronous counters: state diagram, Circuit implementation. Shift registers. [4]
- **Memory Devices** Classification of memories, RAM organization, Write operation, Read operation, Memory cycle. Static RAM Cell-Bipolar, RAM cell, MOSFET RAM cell, Dynamic RAM cell. ROM organization, PROM, EPROM, EEPROM, Field Programmable Gate Arrays (FPGA). [4]

Suggested Readings/ Books

Pariticals

Notice: Undefined variable: data1 in /var/www/wordpress/wp-content/plugins/addpage/file1.php on line 89

- Study of Logic Gates: Truth-table verification of OR, AND, NOT, XOR, NAND and NOR gates; Realization of OR, AND, NOT and XOR functions using universal gates.
- Half Adder / Full Adder: Realization using basic and XOR gates.
- Half Subtractor / Full Subtractor: Realization using NAND gates.
- 4-Bit Binary-to-Gray & Gray-to-Binary Code Converter: Realization using XOR gates.
- 4-Bit and 8-Bit Comparator: Implementation using IC7485 magnitude comparator chips.
- Multiplexer: Truth-table verification and realization of Half adder and Full adder using IC74153 chip.
- Demultiplexer: Truth-table verification and realization of Half subtractor and Full subtractor using IC74139 chip.
- Flip Flops: Truth-table verification of JK Master Slave FF, T-type and D-type FF using IC7476 chip.
- Asynchronous Counter: Realization of 4-bit up counter and Mod-N counter using IC7490 & IC7493 chip.
- Synchronous Counter: Realization of 4-bit up/down counter and Mod-N counter using IC74192 & IC74193 chip.
- Shift Register: Study of shift right, SIPO, SISO, PIPO, PISO & Shift left operations using IC7495 chip.
- DAC Operation: Study of 8-bit DAC (IC 08/0800 chip), obtain staircase waveform using IC7493 chip.
- ADC Operations: Study of 8-bit ADC.

1. NAME OF DEPTT./CENTER: **Computer Science Engineering**
2. subject code: **BTCS 304 Course Title: Data Structures**
3. Contact Hours **L:3 T:2 P:1**
4. Examination Duration(Hrs): **Theory** 3 **Practial** 1
5. credits: 2
6. semester **3**
7. **Objectives : nnnnnnnnnnnnnnnnnnnnnn**

- **Dynamic Memory Management** Understanding pointers, usage of pointers, arithmetic on pointers, memory allocation, memory management functions and operators, debugging pointers - dangling pointers, memory leaks, etc. [2]
- **Introduction** Concept of data type, definition and brief description of various data structures, data structures versus data types, operations on data structures, algorithm complexity, Big O notation. [2]
- **Arrays** Linear and multi-dimensional arrays and their representation, operations on arrays, sparse matrices and their storage. [3]
- **Linked List** Linear linked list, operations on linear linked list, doubly linked list, operations on doubly linked list, application of linked lists. [4]
- **Queues** equential representation of queue, linear queue, circular queue, operations on linear and circular queue, linked representation of a queue and operations on it, deque, priority queue, applications of queues. [4]
- **Trees** Basic terminology, sequential and linked representations of trees, traversing a binary tree using recursive and non-recursive procedures, inserting a node, deleting a node, brief introduction to threaded binary trees, AVL trees and B-trees. [4]
- **Heaps** Representing a heap in memory, operations on heaps, application of heap in implementing priority queue and heap sort algorithm. [2]
- **Graphs** Basic terminology, representation of graphs (adjacency matrix, adjacency list), traversal of a graph (breadth-first search and depth-first search), and applications of graphs. [3]
- **Hashing & Hash Tables** Comparing direct address tables with hash tables, hash functions, concept of collision and its resolution using open addressing and separate chaining, double hashing, rehashing. [3]
- **Searching & Sorting** Searching an element using linear search and binary search techniques, Sorting arrays using bubble sort, selection sort, insertion sort, quick sort, merge sort, heap sort, shell sort and radix sort, complexities of searching & sorting algorithms. [5]

Suggested Readings/ Books

- Sartaj Sahni, **Dynamic Memory Management,** hitesh.
- A. A. V. I. M., **Data Structures using C and C++,** A & C BLACK.

Pariticals

Notice: Undefined variable: data1 in /var/www/wordpress/wp-content/plugins/addpage/file1.php on line 89

- Write a menu driven program that implements following operations (using separate functions) on a linear array:
- Write a menu driven program that maintains a linear linked list whose elements are stored in on ascending order and implements the following operations (using separate functions)
- Write a program to demonstrate the use of stack (implemented using linear array) in converting arithmetic expression from infix notation to postfix notation.
- Program to demonstrate the use of stack (implemented using linear linked lists) in evaluating arithmetic expression in postfix notation.
- Program to demonstration the implementation of various operations on a linear queue represented using a linear array.
- Program to demonstration the implementation of various operations on a circular queue represented
- Program to demonstration the implementation of various operations on a queue represented using a linear linked list (linked queue).
- Program to illustrate the implementation of different operations on a binary search tree.
- Program to illustrate the traversal of graph using breadth-first search.
- Program to illustrate the traversal of graph using depth-first search.
- Program to sort an array of integers in ascending order using bubble sort.
- Program to sort an array of integers in ascending order using selection sort.
- Program to sort an array of integers in ascending order using insertion sort.
- Program to sort an array of integers in ascending order using radix sort.
- Program to sort an array of integers in ascending order using merge sort.
- Program to sort an array of integers in ascending order using quick sort.
- Program to sort an array of integers in ascending order using heap sort.
- Program to sort an array of integers in ascending order using shell sort.
- Program to demonstrate the use of linear search to search a given element in an array.
- Program to demonstrate the use of binary search to search a given element in a sorted array in ascending order.

1. NAME OF DEPTT./CENTER: **Computer Science Engineering**
2. subject code: **BTCS 305 Course Title: Object Oriented Programming Using C++**
3. Contact Hours **L:3 T:2 P:1**
4. Examination Duration(Hrs): **Theory** 3 **Practial** 1
5. credits: 2
6. semester **3**
7. **Objectives : mlllllllllllllllllllllllllllllll**

- **Object-Oriented Programming Concepts** Introduction, comparision between procedural programming paradigm and object-oriented proramming paradigm, basic concepts of object-oriented programming concepts of an object and a class, interface and implementation of a class, operations on objects, relationship among objects, abstraction, encapsulation, data hiding, inheritance, overloading, polymorphism, messaging. [2]
- **Standard Input/Output** Concept of streams, hierarchy of console stream classes, input,output using overloaded operators and and memberv functions of i,o stream classes, formatting output, formatting using ios class functions and flags, formatting using manipulators. [3]
- **Classes and Objects** Specifying a class, creating class objects, accessing class members, access specifiers, static members, use of const keyword, friends of a class, empty classes, nested classes, local classes, abstract classes, container classes, bit fields and classes. [4]
- **Pointers and Dynamic Memory Management** Declaring and initializing pointers, accessing data through pointers, pointer arithmetic, memory allocation (static and dynamic), dynamic memory management using new and delete operators, pointer to an object, this pointer, pointer related problems - dangling,wild pointers, null pointer assignment, memory leak and allocation failures. [5]
- **Constructors and Destructors** Need for constructors and destructors, copy constructor, dynamic constructors, explicit constructors, destructors, constructors and destructors with static members, initializer lists. [2]
- **Operator Overloading and Type Conversion** Overloading operators, rules for overloading operators, overloading of various operators, type conversion - basic type to class type, class type to basic type, class type to another class type. [4]
- **Inheritance** Introduction, defining derived classes, forms of inheritance, ambiguity in multiple and multipath inheritance, virtual base class, object slicing, overriding member functions, object composition and delegation, order of execution of constructors and destructors. [5]
- **Virtual functions & Polymorphism** Concept of binding - early binding and late binding, virtual functions, pure virtual functions, abstract clasess, virtual destructors [3]
- **Exception Handling** Review of traditional error handling, basics of exception handling, exception handling mechanism, throwing mechanism, catching mechanism, rethrowing an exception, specifying exceptions. [2]
- **Templates and Generic Programming** Template concepts, Function templates, class templates, illustrative examples [3]
- **Files** File streams, hierarchy of file stream classes, error handling during file operations, reading,writing of files, accessing records randomly, updating files [3]

Suggested Readings/ Books

- Lafore R, **Object Oriented Programming in C++,** William Stallings.
- E. Balagurusamy, **Object Oriented Programming with C++,** Bjarne Stroustrup.
- R. S. Salaria, **Mastering Object-Oriented Programming with C++,** Sartaj Sahni.

Pariticals

Notice: Undefined variable: data1 in /var/www/wordpress/wp-content/plugins/addpage/file1.php on line 89

- Write a program that uses a class where the member functions are defined inside a class
- Write a program that uses a class where the member functions are defined outside a class.
- Write a program to demonstrate the use of static data members.
- Write a program to demonstrate the use of const data members
- Write a program to demonstrate the use of zero argument and parameterized constructors.
- Write a program to demonstrate the use of dynamic constructor.
- Write a program to demonstrate the use of explicit constructor.
- Write a program to demonstrate the use of initializer list.
- Write a program to demonstrate the overloading of increment and decrement operators.
- Write a program to demonstrate the overloading of binary arithmetic operators.
- Write a program to demonstrate the overloading of memory management operators.
- Write a program to demonstrate the typecasting of basic type to class type.
- Write a program to demonstrate the typecasting of class type to basic type.
- Write a program to demonstrate the multilevel inheritance.