Semester- V

Major/DS Course (Core): COMP 5011 (Theory)- Data Communications & Networking

Credit: 04 **60 Hours**

Course Objective

The course objective of this course is to understand the basic concepts behind computer networks and data communication. It helps students to learn the use of different layers in standard reference models used for communication. It also assists students to know main features of protocols used at various layers and understand the utility of different networking devices.

Course Learning Outcomes

On successful completion of the course, the student will be able to:

i. Describe the hardware and software components used in a network.

ii. Compare OSI and TCP/IP reference models at various layers.

iii. Describe, analyse and compare different data link, network, transport and application layer protocols.

iv. Design/implement data link and network layer protocols in a simulated networking environment.

Syllabus:

1. Data Communication Fundamentals and Techniques (15 Hours)

Introduction: Communication systems, Analogue data, digital data, Communication channels, Synchronous data, Asynchronous data.

Analog and digital signal; data-rate limits; digital to digital line encoding schemes; pulse code

modulation; parallel and serial transmission, digital to analog modulation , ASK, FSK, PSK, Quadrature PSK, Differential PSK. multiplexing techniques- FDM, TDM;

Transmission media: Twin wire, Coaxial cable, Radio, VHF and microwaves, satellite communication, Optical fiber.

2. Introduction to Computer Networks (5 Hours)

Network definition; network topologies; network classifications; network protocol; layered

Network architecture; overview of OSI reference model; overview of TCP/IP protocol suite.

3. Network Switching Techniques and Access mechanisms (10 Hours)

Circuit switching; packet switching- connectionless datagram switching, connection-oriented

Virtual circuit switching; dial-up modems; digital subscriber line; cable TV for data transfer.

4. Data Link Layer Functions and Protocol (10 Hours)

Error detection and error correction techniques (CRC, Hamming-codes); data-link control-

Framing and flow control; error recovery protocols- stop and wait ARQ, go-back-n ARQ; Point

to Point Protocol on Internet.

5. Multiple Access Protocol and Networks (5 Hours)

ALOHA & CSMA protocols; CDMA; Ethernet LANs; connecting LAN and back-bone

networks- repeaters, hubs, switches, bridges, router and gateways;

6. Networks Layer Functions and Protocols (5 Hours)

Routing; static and dynamic routing algorithms; network layer protocol of Internet- IP protocol,

Internet control protocols.

7. Transport Layer Functions and Protocols (5 Hours)

Transport services- error and flow control, Connection establishment and release- three way

handshake, TCP and UDP.

8. Overview of Application layer protocol (5 Hours)

Overview of DNS protocol; overview of WWW & HTTP protocol.

Reference Books

1. B. A. Forouzan: Data Communications and Networking, Fourth edition, THM ,2007.

2. A. S. Tanenbaum: Computer Networks, Fourth edition, PHI , 2002

COMP 5011 (Practical): Computer Networks

Credit:1 **30 Hours**

All programs should be developed in C / Java / Python

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.

2. Simulate Hamming-code based error detection & correction algorithm for noisy channel.

3. Simulate and implement stop and wait protocol for noisy channel.

4. Simulate and implement go back N sliding window protocol.

5. Simulate and implement selective repeat sliding window protocol.

6. Simulate and implement MST construction (Prim‟s, Kruskal‟s) for Ethernet

7. Simulate and implement the various routing algorithms (RIP, Distance-Vector routing, Dijkstra‟s, Bellman-Ford, Floyd-Warshall, Flooding).

Major/DS Course (Core): COMP 5012 (Theory)- Computer Organization

Credit: 04 **60 Hours**

Course Objective

This course introduces the students to the fundamental concepts of digital computer organization, design and architecture. It aims to develop a basic understanding of the building blocks of the computer system and highlights how these blocks are organized together to architect a digital computer system.

Course Learning Outcomes

On successful completion of the course, students will be able to:

i. Explain instruction cycle, pipelining and interrupts.

ii. Explain data communication between CPU, memory and I/O devices.

iii. Simulate the design of a basic computer using a software tool.

Syllabus:

Introduction: Evolution of Computers, Stored program concept and Von-Neumann architecture, Information representation and codes, Building blocks of Computers. [10 Hours]

Register Transfer and micro operations: Concepts of bus, Data movement among registers, A language to represent conditional data transfer, Data movement from/to memory, Arithmetic and logical operations with register transfer, Timing in register transfer. [15 Hours]

CPU Architecture: Instruction format, Instruction execution, Fetch and execution cycles, CPU organization with large registers, Stacks and handling of interrupts and subroutines, Instruction pipelining: stages, hazards and methods to remove hazards. [15 Hours]

Micro-programmed control unit: Basic organization of micro-programmed controller, Horizontal and vertical formats, Address sequencer. [5 Hours]

I/O Organization: Strobe based and handshake-based communication, Vector and Priority interrupt, DMA based transfer. [5 Hours]

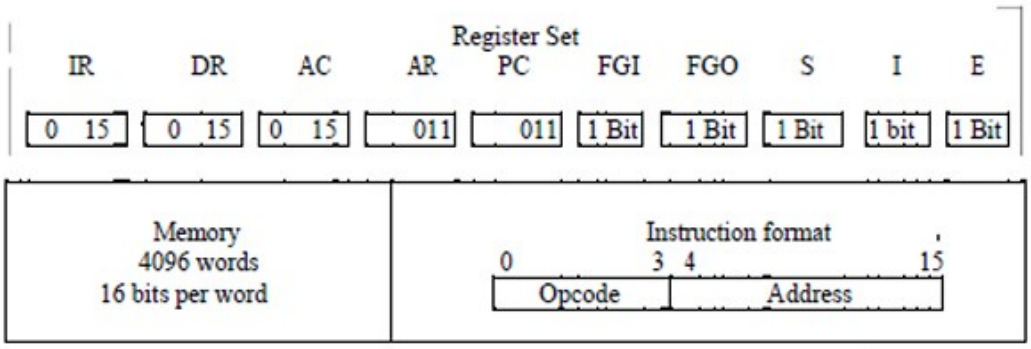
Programming the Basic Computer: Instruction formats, addressing modes, instruction

codes, machine language, assembly language, input output programming. [10 Hours].

COMP 5012 (Practical): Computer Organization

Credit:01 **30 Hours**

1. Create a machine based on the following architecture:



Basic Computer Instructions

Memory Reference			Register Reference		Input-Output		
Symbol	Hex		Symbol	Hex	Symbol	Hex	
AND	0xxx		CLA	E800	INP	F80 0	
ADD	2xxx		CLE	E400	OUT	F40 0	
LDA	4xxx	Direct	CMA	E200	SKI	F20 0	
STA	6xxx	Addressing	CME	E100	SKO	F10 0	
BUN	8xxx		CIR	E080	ION	F08 0	Optional
BSA	Axxx		CIL	E040	IOF	F04 0	
ISZ	Cxxx		INC	E020			
AND_I	1xxx		SPA	E010			
ADD_I	3xxx		SNA	E008			
LDA_I	5xxx	Indirect	SZA	E004			
STA_I	7xxx	Addressing	SZE	E002			
BUN_I	9xxx		HLT	E001			
BSA_I	Bxxx						
ISZ_I	Dxxx						

Refer to Chapter-5 of Morris Mano for description of instructions.

2) Create the micro operations and associate with instructions as given in the chapter (except interrupts). Design the register set, memory and the instruction set. Use this machine for the assignments of this section.

3) Create a Fetch routine of the instruction cycle.

4) Simulate the machine to determine the contents of AC, E, PC, AR and IR registers in hexadecimal after the execution of each of following register reference instructions:

a. CLA b. CLE c. CMA d. CME e. CIR f. CIL g. INC h. SPA i. SNA j. SZA k. SZE l. HLT

Initialize the contents of AC to (A937)16, that of PC to (022)16 and E to 1.

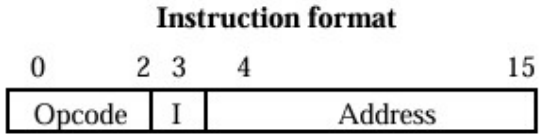
5. Simulate the machine for the following memory-reference instructions with I= 0 and address part = 082. The instruction to be stored at address 022 in RAM. Initialize the memory word at address 082 with the operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.

a. ADD b. AND c. LDA d. STA e. BUN f. BSA g. ISZ

6. Simulate the machine for the memory-reference instructions referred in above question with I= 1 and address part = 082. The instruction to be stored at address 026 in RAM. Initialize the

memory word at address 082 with the value 298. Initialize the memory word at address 298 with operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.

7. Modify the machine created in Practical 1 according to the following instruction format:



a. The instruction format contains a 3-bit opcode, a 1-bit addressing mode and a 12-bit address. There are only two addressing modes, I = 0 (direct addressing) and I = 1 (indirect addressing).

b. Create a new register I of 1 bit.

c. Create two new microinstructions as follows :

i. Check the opcode of instruction to determine type of instruction (Memory Reference/Register Reference/Input-Output) and then jump accordingly.

ii. Check the I bit to determine the addressing mode and then jump accordingly.

Major/DS Course (Core): COMP 5013 (Theory) - Software Engineering

Credit: 04 **60 Hours**

Course Objective

This course will provide fundamental approaches and techniques used to develop good quality software. This includes learning of various software development process frameworks, requirement analysis, design modelling, qualitative and quantitative software metrics, risk management, and testing techniques.

Course Learning Outcomes

On successful completion of the course, a student will be able to:

i. Understand the software development models.

ii. Analyse and model customer requirements; build design models.

iii. Estimate and prepare schedule for software projects.

iv. Analyse the impact of risks involved in software development.

v. Design and build test cases, and to perform software testing.

Syllabus:

Software Engineering Fundamentals: Definition of software product, Software Engineering Paradigms; Software engineering, Knowledge engineering, and End user development approaches. [5 Hours]

System Analysis: An abstraction, Partitioning and projection, Systems specification, Software Requirements Specification (SRS) standards, Formal Specification methods, Specification tools, Flow based, Data based and Object –Oriented Analysis. [7 Hours]

System Documentation: Principles of system documentation, types of documentation and their importance [5 Hours]

System Planning: Data and fact gathering techniques-Interviewing, communications, presentations and site visit. Feasibility study, feasibility reports, prototyping, cost-benefit analysis-tools and techniques. [5 Hours]

Systems Design:

Idealized and constrained design, Process oriented design (Game and Sarson and Yourdon notations), Data oriented design (Warnier–Orr, E-R modelling), Object oriented design (Booch approach), Cohesion and Coupling, Design matrices, Design documentation standard. [5 Hours]

Role of CASE Tools: Relevance of CASE Tools, High-end and Low-end CASE Tools [5 Hours]

Coding and Programming : Choice of programming languages, Mixed language programming and cell semantics ,Re-engineering legacy systems, Coding standard. [5 Hours]

Software Quality and testing: Software quality assurance .Types of Software Testing (White Box and Black Box Testing, Unit Testing, Integration Testing, Verification and Validation of Software) , Debugging and Software Reliability analysis , Software quality and matrices, Software maturity model and extensions. [8 Hours]

Software Cost and Time estimation: Functions points , Issues in software cost estimation : Introduction to the Rayleigh curve, Algorithmic cost models (COCOMO, Putnam- Slim, Watson, and Felix), Other approaches to software cost and Size estimation (software complexity, Delphi , costing by analogy). [8 Hours]

Software Project Management : Planning software , projects, Work breakdown structures, Integrating software design and project planning ,Software project teams, Projecting monitoring and control. [7 Hours]

Books

1. Software Engineering, A practioner's Approach- R. S. Pressman (Mc-Graw Hill Inc)

2. An Integrated Approach to Software Engineering-P.Jalote (Narosa Publication House)

3. R. Mall, Fundamentals of Software Engineering (2nd Edition), Prentice-Hall of India,

2004.

COMP 5013 (Tutorial) : Software Engineering

Tutorial: 15 Lectures

Tutorial based on theory classes.

Credit: 1

Semester- VI