

- Balaguruswamy E. Introduction to Computing and Problem Solving using Python, 2nd edition, McGraw Hill Education, 2018.

### **Suggestive Readings**

- Brown, Martin C. Python: The Complete Reference, 2nd edition, McGraw Hill Education, 2018.
- Guttag, J.V. Introduction to computation and programming using Python, 2nd edition, MIT Press, 2016.

**Note:** Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

## **DISCIPLINE SPECIFIC CORE COURSE – 2: COMPUTER SYSTEM ARCHITECTURE**

### **CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE**

| Course title & Code          | Credits | Credit distribution of the course |          |                     | Eligibility criteria | Pre-requisite of the course (if any) |
|------------------------------|---------|-----------------------------------|----------|---------------------|----------------------|--------------------------------------|
|                              |         | Lecture                           | Tutorial | Practical/ Practice |                      |                                      |
| Computer System Architecture | 4       | 3                                 | 0        | 1                   | Class XII pass       | NIL                                  |

### **Learning Objectives**

The Learning Objectives of this course are as follows:

- Introduces the students to the fundamental concepts of digital computer organization, design and architecture.
- 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.

### **Learning Outcomes**

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

- Design Combinational Circuits using basic building blocks. Simplify these circuits using Boolean algebra and Karnaugh maps. Differentiate between combinational circuits and sequential circuits.
- Represent data in binary form, convert numeric data between different number systems and perform arithmetic operations in binary.
- Determine various stages of instruction cycle, pipelining and describe interrupts and their handling.
- Explain how CPU communicates with memory and I/O devices and distinguish between different types of processors.
- Simulate the design of a basic computer using a software tool.

## SYLLABUS OF DSC - 2

### Theory

#### **Unit – 1 (6 hours)**

##### **Digital Logic Circuits**

Logic Gates, Truth Tables, Boolean Algebra, Digital Circuits, Combinational Circuits, Introduction to Sequential Circuits, Circuit Simplification using Karnaugh Map, Don't Care Conditions, Flip-Flops, Characteristic Tables, Excitation Table.

#### **Unit – 2 (9 hours)**

##### **Digital Components (Fundamental building blocks)**

Designing of combinational circuits- Half Adder, Full Adder, Decoders, Encoders, Multiplexers, Registers and Memory (RAM, ROM and their types), Arithmetic Microoperations, Binary Adder, Binary Adder-Subtractor.

#### **Unit – 3 (6 hours)**

##### **Data Representation and Basic Computer Arithmetic**

Number System,  $r$  and  $(r-1)$ 's Complements, data representation and arithmetic operations.

#### **Unit – 4 (9 hours)**

##### **Basic Computer Organization and Design**

Bus organization, Microprogrammed vs Hardwired Control, Instruction Codes, Instruction Format, Instruction Cycle, Instruction pipelining, Memory Reference, Register Reference and Input Output Instructions, Program Interrupt and Interrupt Cycle..

#### **Unit – 5 (6 hours)**

##### **Processors**

General register organization, Stack Organization, Addressing Modes, Overview of Reduced Instruction Set Computer (RISC), Complex Instruction Set Computer (CISC), Multicore processor and Graphics Processing Unit (GPU).

#### **Unit – 6 (9 hours)**

##### **Memory and Input-Output Organization**

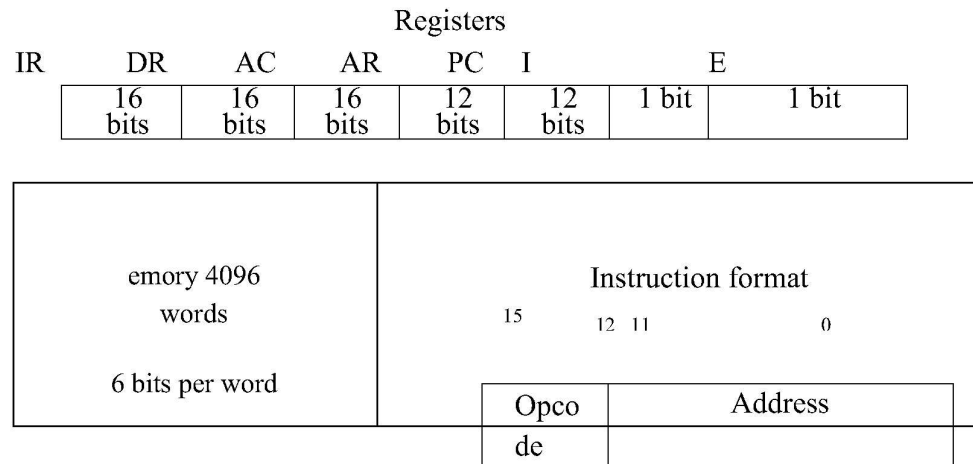
Memory hierarchy (main, cache and auxiliary memory), Input-Output Interface, Modes of Transfer: Programmed I/O, Interrupt initiated I/O, Direct memory access.

#### **Practical (30 hours)**

##### **List of Practicals:**

(Use Simulator – CPU Sim 3.6.9 or any higher version for the implementation)

1. Create a machine based on the following architecture:



#### Basic Computer Instructions

| Memory Reference |      |                     | Register Reference |      |
|------------------|------|---------------------|--------------------|------|
| Symbol           | Hex  |                     | Symbol             | Hex  |
| AND              | 0xxx | Direct Addressing   | CLA                | 7800 |
| ADD              | 1xxx |                     | CLE                | 7400 |
| LDA              | 2xxx |                     | CMA                | 7200 |
| STA              | 3xxx |                     | CME                | 7100 |
| BUN              | 4xxx |                     | CIR                | 7080 |
| BSA              | 5xxx |                     | CIL                | 7040 |
| ISZ              | 6xxx |                     | INC                | 7020 |
| AND_I            | 8xxx | Indirect Addressing | SPA                | 7010 |
| ADD_I            | 9xxx |                     | SNA                | 7008 |
| LDA_I            | Axxx |                     | SZA                | 7004 |
| STA_I            | Bxxx |                     | SZE                | 7002 |
| BUN_I            | Cxxx |                     | HLT                | 7001 |
| BSA_I            | Dxxx |                     | INP                | F800 |
| ISZ_I            | Exxx |                     | OUT                | F400 |

Refer to Chapter-5 of reference 1 for description of instructions.

Design the register set, memory and the instruction set. Use this machine for the assignments of this section.

2. Create a Fetch routine of the instruction cycle.

3. Write an assembly program to simulate ADD operation on two user-entered numbers.
4. Write an assembly program to simulate SUBTRACT operation on two user-entered numbers.
5. Write an assembly program to simulate the following logical operations on two user-entered numbers.
  - i. AND
  - ii. OR
  - iii. NOT
  - iv. XOR
  - v. NOR
  - vi. NAND
6. Write an assembly program for simulating following memory-reference instructions.
  - i. ADD
  - ii. LDA
  - iii. STA
  - iv. BUN
  - v. ISZ
7. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:
  - i. CLA
  - ii. CMA
  - iii. CME
  - iv. HLT
8. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:
  - i. INC
  - ii. SPA
  - iii. SNA
  - iv. SZE

9. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:
- i. CIR
  - ii. CIL
10. Write an assembly program that reads in integers and adds them together; until a negative non-zero number is read in. Then it outputs the sum (not including the last number).
11. Write an assembly program that reads in integers and adds them together; until zero is read in. Then it outputs the sum.

#### **Essential Readings**

- David A. Patterson and John L. Hennessy. “Computer Organization and Design: The Hardware/Software interface”, 5th edition, Elsevier, 2012.
- Mano, M. Computer System Architecture, 3rd edition, Pearson Education, 1993.

#### **Suggestive Readings**

- Mano, M. Digital Design, Pearson Education Asia, 1995.
- Null, L., & Lobur, J. The Essentials of Computer Organization and Architecture. 5th edition, (Reprint) Jones and Bartlett Learning, 2018.
- Stallings, W. Computer Organization and Architecture Designing for Performance 8th edition, Prentice Hall of India, 2010.

**Note:** Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.