**Guidelines of B.Sc. (H) Computer Science I Semester / B.Sc. Programme III Semester/ B.A. Programme III Semester/ Generic Elective II Semester (NEP UGCF 2022)**

**Computer System Architecture**
**DSC02/DSC03/GE 2c**
**(Effective from Academic Year 2024-25)**

| S. No. | Unit Name | Chapter Number and Name | Section Numbers | Reference | No. of Lectures* | Week No. |
|---|---|---|---|---|---|---|
| 1 | **Unit 1:** Digital Logic Circuits | Ch 1: Digital Logic Circuits | 1.1, 1.2, 1.3, 1.4, 1.5, 1.6<br>1.7 (up to pg. 28) | [1] | 6 | 1-2 |
| 2 | **Unit 2:** Digital Components (Fundamental building blocks): | Ch 2: Digital Components | 2.2, 2.3, 2.7 | | 5 | 3-5 |
| | | Ch 4: Register Transfer and Micro-operations | 4.4 (up to fig. 4.7) | | 4 | |
| 3 | **Unit 3:** Data Representation and Basic Computer Arithmetic | Ch 3: Data Representation | 3.1, 3.2, 3.3 | | 6 | 6-7 |
| 4 | **Unit 4:** Basic Computer Organization and Design | Ch 5: Basic Computer Organization and Design | 5.1, 5.2, 5.3,<br>5.4 (up to pg. 137),<br>5.5, 5.6, 5.7 | | 10 | 8-11 |
| | | Ch 9: Pipeline and Vector Processing | 9.2 | | 2 | |
| 5 | **Unit 5:** Processors: | Ch 8: Central Processing Unit | 8.1, 8.2, 8.3 (up to pg. 247), 8.5, 8.8 (only characteristics, i.e., pg. 282 – 284) | | 4 | 12-13 |
| | | Ch-7: Multicores, Multiprocessors, and Clusters | 7.1 (page 632- Introduction of Multicore Processor)<br>7.7 (page 654-656) Characteristics of GPU Vs. CPU | [2] | 1 | |
| 6 | **Unit 6:** Memory and Input-Output Organization | Ch 11: Input Output Organization | 11.2 (up to pg-388), 11.4, 11.6 (up to pg-416) | [1] | 6 | 14-15 |
| | | Ch 12: Memory Organization | 12.1 (up to pg-446) | | 1 | |

*The number of lectures and chapter wise weightage may be treated as indicative only.

## References:


[1] Computer System Architecture: Morris M. Mano (Pearson Education, 3rd Edition)
[2] Patterson and John L. Hennessy. "Computer Organization and Design: The Hardware/Software interface", 5th edition, Elsevier, 2012.

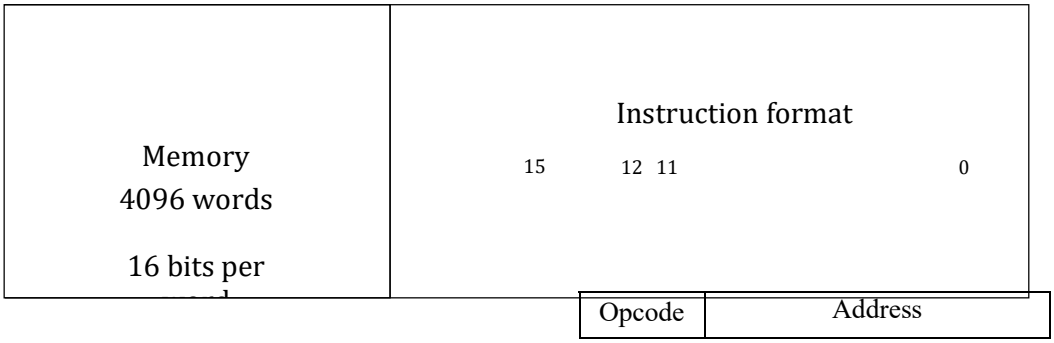
**Practical**

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

1. Create a machine based on the following architecture:

Registers

| IR | DR | AC | AR | PC | I | E |
|---|---|---|---|---|---|---|
| 16 bits | 16 bits | 16 bits | 12 bits | 12 bits | 1 bit | 1 bit |




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.