राष्ट्रीय प्रौद्योगिकी संस्थान पटना / NATIONAL INSTITUE OF TECHNOLOGY PATNA
संगणक विज्ञान एंव अभियांत्रिकी विभाग / DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
अशोक राजपथ, पटना-८००००५, बिहार / ASHOK RAJPATH, PATNA-800005, BIHAR
Phone No.: 0612-2372715, 2370419, 2370843, 2371929 Ext- 200, 202 Fax-0612-2670631 Website: www.nitp.ac.in

No:- Date:

CS24108 Computer Organization

L-T-P-Cr: 3-0-2-4

Prerequisites: Programming in C, Digital Design.

Course Overview: The purpose of this course is to impart fundamental concepts of computer (particularly, structures and functions of computer system, microarchitecture, interconnectivity) and their functioning, as well as hands-on experiments in simulation platforms or hardware kits.

Course Outcomes – After completing this course, students should be able to:

CO-1. *recall* computer organization terminologies, as well as fundamental concepts, operating principles and methods about structures and functions of computer systems and their components; [Bloom level: Remember; Mapped to: PO-1]

CO-2. *explain* concepts and techniques in functioning of arithmetic logic unit (ALU), floating-point unit (FPU), control unit (CU), instruction set architecture (ISA), memory hierarchy and input/output unit; [Bloom level: Understand; Mapped to: PO-1, PO-2]

CO-3. *solve* problems on principles of arithmetic/logic functions, control mechanisms, cache, primary, secondary memory addressing, I/O addressing, instruction pipelining, as well as their performances; [Bloom level: Apply; Mapped to: PO-1, PO-2, PO-3]

CO-4. *use* software tools for designing and evaluating microarchitecture in computer system for given ISA; [Bloom level: Apply; Mapped to: PO-3, PO-5]

CO-5. *implement* assembly-language programs to fulfill program objectives in computer system for given ISA; [Bloom level: Apply; Mapped to: PO-3, PO-5]

CO-6. *determine* pipeline hazards in computer system for given execution sequence. [Bloom level: Analyze; Mapped to: PO-1, PO-2, PO-3]

Syllabus:

Unit I: (CPU data path) Lectures: 11

(Module-1) *Introduction*: Introduction to computer organization; Microarchitecture and architecture of computer system; Structural and functional views of computer system.
(Module-2) *Arithmetic Data Path*: Unsigned and signed integer representations; Integer addition, subtraction; Booth's multiplication; Restoring and nonrestoring division; Data path for computer arithmetic on unsigned and signed integers.
(Module-3) *IEEE 754 FP Data Path*: Real number representations; IEEE 754 FP standard representation; Addition, subtraction, multiplication, division in IEEE 754 FP; Data path for IEEE 754 FP arithmetic.

Unit II: (Control Unit & Instruction Set Architecture) Lectures: 10

(Module-1) *Control Unit*: Control unit and instruction set; Hardwired and microprogrammed control; Microoperation sequencing logic; Control memory.
(Module-2) *Instruction Set Architecture*: Instruction set; CISC and RISC systems; Instruction cycle and state diagram; Instruction set design factors; Address space and addressing modes; Instruction sets in CISC and RISC systems.

Unit III: (Memory & I/O) Lectures: 10

(Module-1) *Memory Hierarchy*: Memory system characteristics; Memory system technologies; Memory hierarchy; Cache memory; Main memory, RAM, ROM; Flash memory; External memory; SSD; RAID.
(Module-2) *Input/Output Techniques*: I/O structures and functions; I/O techniques; Direct Memory Access: Direct Cache Access.

Unit IV: (Instruction pipeline) Lectures: 11

(Module-1) *Processing Performance Measurement*: Performance metrics of processing system; Processing system classification; Flynn taxonomy.
(Module-2) *Instruction Pipeline*: RAW, WAW, WAR dependencies; Instruction pipelining; Pipeline hazards and remedy techniques.

Text/ Reference Book(s):

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Naraig Manjikian, Computer Organization and Embedded Systems, McGraw Hill, Sixth edition, 2011 (ISBN: 9780073380650).
2. David A. Patterson, John L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, Morgan Kaufmann, Sixth edition, 2021.
3. William Stallings, *Computer Organization and Architecture: Designing for Performance*, Pearson Education, Eleventh edition, 2019.
4. John L. Hennessy, David A. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann, Sixth edition, 2019.
5. Andrew S. Tanenbaum, Todd Austin, *Structured Computer Organization*, Pearson Education, Sixth edition, 2013.
6. Barry B. Brey, *The Intel Microprocessors: Architecture, Programming, and Interfacing*, Pearson Education, Eighth edition, 2009.

List of experiments:

1. Use a logic simulation platform to implement 32-bit ADDER (supporting unsigned and two complement addition).
2. Implement 32-bit twos-complement subtraction through 32-bit ADDER using a logic simulation platform.
3. Implement multiplication of two 32-bit integers (in twos complement form) following Booth's algorithm using a logic simulation platform.
4. Implement 32-bit ALU (supporting add/sub/mul/div) using a logic simulation platform.
5. Use a logic simulation platform to implement 32-bit FADD (to support single-precision floating-point addition in IEEE 754 standard representation).
6. Implement a machine-language program to perform arithmetic operations on signed integers using an assembler platform based on given ISA.
7. Implement a machine-language program to perform arithmetic operations on signed floating-point numbers based on given ISA.
8. Implement a machine-language program to calculate average of binary integer series through usage of different addressing modes based on given ISA.
9. Implement a machine-language program to calculate average of odd integers only from binary integer series (with subroutine to check for odd numbers) based on given ISA.
10. Implement a machine-language program to arrange binary integer series in ascending/ descending order based on given ISA.

Mini Projects:

I. Implement calculator to support basic arithmetic operations, exponentiation, unary operations and conversions on decimal/binary signed integers and floating-point numbers based on given ISA.
II. Implement interrupt service routine.