Indian Institute of Technology Kharagpur

Computer Science and Engineering

CS 31007 COMPUTER ORGANIZATION AND ARCHITECTURE Autumn 2019

(L-T-P: 3-1-0; Credit = 4)

Semester Commencement: Monday, 15 July 2019; Closing: Friday, 15 Nov. 2019

Class Schedule: 12:00-12:55 (Mon), 10:00-11:55 (Tue), and 8:00-8:55 (Thurs)

Unless otherwise mentioned, **Tutorial classes** will be held on **every Thursday**

Instructors: *Section* I - Rajat Subhra Chakraborty (CSE – A209), Class Room NR 423
Email: rschakraborty@cse.iitkgp.ac.in;
Section II - Bhargab B. Bhattacharya (CSE – 124); Class Room NR 124
Email: bhargab@cse.iitkgp.ac.in;

Teaching Assistants: Pranesh Santikellur (pra.net01@gmail.com)
B. V. Sreekanth (balijavenkatasreekanth@gmail.com)
Rajat Sadhukhan (rajatssr835@gmail.com)
Paulson Mathew (paulsonmathew21irtn@gmail.com)
Sunandan Adhikary (mesunandan@gmail.com)
Sayandeep Sanyal (sayandeep.sanyal@iitkgp.ac.in)
Dingari Gauthama Vishnu Simha (dingarigauthamavishnusimha@gmail.com)

Prerequisites: Basic logic design, combinational and sequential circuits, knowledge of high-level programming language.

Textbook:

1. D. A. Patterson and J. L. Hennessy, *Computer Organization and Design - the Hardware Software Interface*, 5th Edition, Elsevier, Morgan Kaufmann, 2014.
2. Older editions of the same book [2nd Ed. (1998), 3rd Ed. (2005), 4th Ed. (2012)], and newer RISC V (2018) Edition may also be referred to while discussing some topics).

Further Reading:
1. Smruti R. Sarangi, *Computer Organisation and Architecture*, McGraw Hill India, 2014.
2. William Stallings, *Computer Organization and Architecture: Designing for Performance*, Eight Edition, Prentice Hall, 2010.
3. John P. Hayes, *Computer Architecture and Organization*, 3rd Edition, Tata McGraw Hill, 2012.

Additionally, numerous video lectures and lecture slides are available online for studying the subject and practicing problem solving.

Evaluation
- Homework and Quiz for practicing (will not be graded)
- In-Class Quiz on Tutorial Days: 10% (=5 × 2)
- Class-Test 1 and Class-Test 2: 10% (=2 × 5)
- Mid-Sem Exam (2 hour): 30%; 16-24 Sept. 2019
- End-Sem Exam (3 hour): 50%; 18-27 November 2019

Attendance

- Regularity in attendance should be honored. **Attendance below 80% without valid reasons may lead to de-registration from the course**.

Goals: Understanding the principles of computer architecture and organization lies at the foundation of computer system design. The aim of this course is to enable students to learn how programs written in a high-level programming language are executed on a physical machine. What is the hardware-software interface? Does the underlying "organization" of physical resources define the "architecture" of a machine, or vice-versa? What system design issues need to be considered for defining and designing the *Instruction Set Architecture* (ISA) so that the performance is optimized? We will highlight techniques for designing instruction sets, pipelining, computer arithmetic, and memory hierarchy. Special emphasis will be given on MIPS (*Microprocessor without Interlocked Pipelined Stages*), and RISC (*Reduced Instruction Set Computer*). At the end of the course, students are expected to learn how a complete processor is designed in order to support an instruction set architecture.

Course Content:

The course will cover the material in the text by Patterson and Hennessy. Topics include: performance analysis, instruction-set architecture, computer arithmetic, CPU design, pipelining, memory systems, and multiprocessing.

1. **Introduction:** Evolution of computers, computer families, Hierarchical design of computer systems, hardware/software interface. Model of computation – Turing Machine. Overall system organization: Basic functional blocks of a computer - CPU, memory, input-output subsystems, control unit. Instruction set architecture (ISA): registers, execution cycle, addressing modes, instruction set. Throughput, response time, and performance; CPU Performance Equation, Amdahl's Law. RISC versus CISC machines.
2. **Instruction sets:** Introduction to MIPS, SPIM; operations, operands, and op-codes; Addressing modes; Instruction types; branches, procedures; MIPS assembly language programming. Case studies.
3. **Computer arithmetic:** Number systems – Fixed and floating point binary number representation; Integer addition and subtraction, Design of arithmetic logic unit (ALU): adders, comparators, multipliers; ripple-carry adder, carry look-ahead adder; carry-select adders; logarithmic adders – Brent-Kung adder. Integer multiplication; Booth multiplier, floating point adder and multipliers.
4. **Processor design:** Data path and control unit design; Single cycle/multi-cycle implementation of a simple CPU based on MIPS Instruction Set Architecture.
5. **Pipelining:** Overview, Design of a simple pipeline; Pipeline stages and control; Forwarding techniques; Hazards; Branch prediction.
6. **Memory hierarchy:** Introduction, Cache architecture; Improving cache performance; Main memory, Memory interleaving; Virtual memory.
7. **Multi-Core and Multi-Processing** (if time permits): SISD, MIMD, SIMD, Vector processing; Multithreading, Shared Memory Multiprocessors; Graphics Processing Units; Clusters and Cloud; Multiprocessor Network Topologies.