CS 385: Computer Architecture

1. Course number and name: CS 385: Computer Architecture

2. Credits and contact hours: 3.000 Credit Hours, 3.000 Lecture hours

3. Instructor or course coordinator: Zdravko Markov


   a. other supplemental materials
      o Course syllabus and lecture notes are available online at:
        http://www.cs.ccsu.edu/~markov/ccsu_courses/385Syllabus.html
      o The course uses Blackboard/Vista for projects, quizzes, tests, and grade book.

5. Specific course information

   a. brief description of the content of the course (Catalog Description):
      The architecture of the computer is explored by studying its various levels: physical level, operating-system level, conventional machine level and higher levels. An introduction to microprogramming and computer networking is provided.

   b. prerequisites or co-requisites: CS 354

   c. indicate whether a required, elective, or selected elective course in the program: Elective

6. Specific goals for the course

   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.

      Upon successful completion of the course the student will be able to

      - Understand the fundamentals of different instruction set architectures and their relationship to the CPU design.
      - Understand the principles and the implementation of computer arithmetic.
      - Understand the operation of modern CPUs including pipelining, memory systems and busses.
      - Understand the principles of operation of multiprocessor systems.
      - Design a CPU by a given specification using the hardware description language (HDL).
      - Write simple parallel programs.
b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

The course outcomes will allow students taking CS385 to make progress toward the following department outcomes:

- **Outcome a**: An ability to apply knowledge of computing and mathematics appropriate to the discipline
- **Outcome b**: An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
- **Outcome c**: An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
- **Outcome i**: An ability to use current techniques, skills, and tools necessary for computing practice.
- **Outcome j**: An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.

7. Brief list of topics covered

- MIPS instruction set
- Computer arithmetic and ALU design
- Datapath and control
- Pipelining
- Memory hierarchy, caches and virtual memory
- Interfacing CPU and peripherals, buses
- Multiprocessors, networks of multiprocessors, parallel programming
- Performance issues

8. Expected Performance Criteria

Students will write assembly language programs to demonstrate their mastery of the MIPS processor instruction set architecture. The student’s understanding of the principles of computer architecture and the hardware/software interface will be demonstrated by answering questions on a midterm test and a final exam. The student’s ability to apply these principles in practice will be demonstrated in a semester project that requires designing a 16-bit version of the MIPS machine and implementing it in Verilog HDL.