ECE 429/629 Organization of Computers
Fall 2009

Instructor: Stan Birchfield, 207A Riggs Hall, 656-5912, stb at clemson
Office Hours: 10:45-11:45 TTh, or by appointment
Course website: **http://www.ces.clemson.edu/~stb/ece429**
Text:

- Hennessy and Patterson, *Computer Architecture: A Quantitative Approach*, 4th ed., Morgan Kaufmann, 2007 (required)
- Patterson and Hennessy, *Computer Organization and Design: The Hardware / Software Interface*, 2nd ed., Morgan Kaufmann, 1998 (recommended, on reserve at the library)

Prerequisites: ECE 272

Overview: This course introduces the principles of advanced computer architecture. Students are expected to enter this class with a basic understanding of computer architecture and familiarity with assembly language. Building upon these fundamentals, students learn advanced architectural techniques for making computers run orders of magnitude faster than would be possible from technological improvements alone.

Objectives: By the end of the course, students should be able to do the following:

- *Fundamental concepts.* Describe the basic concepts of processor architecture, including datapath, control, instruction set, pipelining, and memory hierarchy. Explain the difference between RISC and CISC, their historical development, factors to consider in designing a processor, and effective methods for evaluating processor performance. Calculate performance using the performance equation, Amdahl's Law, and SPEC ratings.
- *CPU internals.* Describe how to build a CPU from individual transistors. Determine values of control and data lines on single-cycle, multi-cycle, and pipelined CPUs. Identify pipeline hazards from MIPS assembly code, and describe solutions to overcome them.
- *Advanced concepts.* Explain the gap in processor-memory performance. Identify software and hardware techniques for minimizing the impact of the gap. Compute state of cache after accesses. Explain concepts of dynamic scheduling, thread-level parallelism, speculation, branch prediction, and compiler techniques for exploiting instruction-level parallelism. Determine stage of execution for each instruction using scoreboarding and Tomasulo's algorithm.

Grading: assignments (35%), two exams (20% each), final exam (25%)

Topics:	**Lectures**
• introduction	2
• performance evaluation	1
• instruction sets	3
• datapath and control	6

• pipelining	9
• advanced concepts for ILP	4
• memory hierarchy	2
• thread-level parallelism	1
• tests	2

	30

Students taking the graduate-level version of the course (ECE 629) must, in addition to the above, write a 2-3 page summary of an article describing recent developments in computer architecture.

Academic integrity: "As members of the Clemson University community, we have inherited Thomas Green Clemson's vision of this institution as a 'high seminary of learning.' Fundamental to this vision is a mutual commitment to truthfulness, honor, and responsibility, without which we cannot earn the trust and respect of others. Furthermore, we recognize that academic dishonesty detracts from the value of a Clemson degree. Therefore, we shall not tolerate lying, cheating, or stealing in any form." – from the university's web page on academic integrity: http://www.cs.clemson.edu/html/academics/academic_integrity_2002.html