

EC513 Computer Architecture

2008/2009 Catalog Data:

Computer architecture and design. Topics include computer arithmetic and ALU design; performance evaluation; instruction set design; CPU design, including pipelining, branch prediction, and speculative execution; memory hierarchy, including cache basics, cache design for performance, and virtual memory support; I/O, including devices, interfaces, specification, and modeling. Examples from high-end microprocessors and embedded systems.

Class/Lab Schedule:

LEC: 4hrs/wk (TTh10-12)

Status in the Curriculum: Elective

Textbook:

Computer Architecture: A Quantitative Approach, Morgan-Kaufman, Hennessy & Patterson, 4th Edition

Reference:

Computer Organization: The Hardware Software Interface, Morgan-Kaufman, Patterson & Hennessy

Coordinator:

Martin Herbordt, Associate Professor, ECE

Prerequisites by topic:

EC413

Goals:

Review

- MIPS computer organization

Learn about

- Computer Organization – what the parts of the computer are and how they work
- Computer Architecture -- the design trade-offs within and among the components, and how to evaluate those trade-offs.

Gain experience with

- Evaluating computer system components including CPUs, memory hierarchies, and I/O systems.
- Evaluating Computer Systems

Course Outcomes:

As an outcome of completing this course, students should be able to:

1. Understand the use of circuits to perform computer arithmetic at a level advanced enough to fully understand the primary alternatives, their methods of combination, and their efficacy with respect to possible technologies.
2. Design arithmetic circuits as a function of unit size and latency.
3. Understand caching and virtual memory at the level of contemporary microprocessor systems.
4. Evaluate memory hierarchies using trace-driven simulation and microkernels.

5. Understand CPUs (especially pipelining) at the level of contemporary microprocessors, including the most important design trade-offs.
6. Understand the interaction of program and hardware with respect to both dynamic and static scheduling.
7. Understand the performance of I/O systems up to the level of the application of simple results from queuing theory.
8. Perform simple capacity planning studies of I/O systems.
9. Understand issues in the performance evaluation of computer systems, including benchmarking.
10. Understand the overall organization of computers, including the interaction of memory, CPU, and I/O.
11. Understand architectural issues in hardware interaction with the operating system such as virtual memory and interrupts.
12. Understand architectural issues in hardware interaction with I/O systems such as interrupts and I/O interfaces.
13. Understand the basic issues in tradeoffs in computer organization from both an economic and an engineering point of view.

Course Outcomes mapped to Program Outcomes:

Program Outcomes	A	B	C	D	E	F	G	H	I	J	K
Course Outcomes	1-13	4	2								
Emphasis (1-5)	5	2	2								

Contribution of Course to Meeting the Professional Component:

Engineering topics: 90%
 Math & Basic Science: 5%
 General Education: 5%

Prepared by: Martin Herbordt, Associate Professor

Date: May 29, 2009