

# ECE 463 Syllabus

<b>Course:</b>	ECE 463
<b>Credit Hours:</b>	3
<b>Course Title:</b>	Advanced Microprocessor System Design
<b>Course Description:</b>	

Advanced topics in microprocessor systems design. Measuring performance. Instruction-set architectures. Memory hierarchies, including caches, prefetching, program transformations for optimizing caches, and virtual memory. Processor architecture, including pipelining, hazards, branch prediction, static and dynamic scheduling, instruction-level parallelism, superscalar, and VLIW. Major projects.

**Prerequisite(s):** ECE 406

**Textbook(s) and/or other required material:**

Required textbook:

Computer Architecture: A Quantitative Approach, Fourth Edition. John Hennessy and David Patterson. Morgan Kauffman (Elsevier), 2006, ISBN-13: 978-0-12-370490-0  
\$88.95

Computing environment:

Generic programming tools (compilers, debuggers, etc.) are available on the Eos system. All students should have an Eos account.

Course web page:

Materials (handouts, assignments, project specifications, message boards, etc.) are posted on these two identical course web pages:

<http://www.courses.ncsu.edu/ece463>

<http://www.courses.ncsu.edu/ece521>

**Course objectives. By the end of this course, the student should be able to (use demonstrative verbs):**

1. Analyze, measure, and characterize performance of programs and workloads on computer architectures. Understand and use Amdahl's and Moore's Laws.
2. Design caches, memory hierarchies, and virtual memory management. Characterize memory system performance and understand techniques for improving hit time, miss rate, and miss penalty.
3. Design, analyze, and use instruction-set architectures; discuss trade-offs in instruction-set architecture design.
4. Explain pipeline hazards caused by data and control dependences and resources, design pipeline stages, and analyze pipeline performance.

# ECE 463 Syllabus

5. Design data bypasses and interlock logic.
6. Design one-level and two-level branch predictors.
7. Measure instruction-level parallelism; design dynamic and static instruction scheduling, register renaming, precise interrupts, superscalar processors, and VLIW processors.

## Topics covered:

1. Introduction and overview (1 lecture)
2. Measuring performance and cost (2 lectures)
3. Memory hierarchy design (8 lectures)
4. Instruction set design (2 lectures)
5. Pipelined microarchitectures (6 lectures)
6. Instruction-level parallelism (8 lectures)
7. Midterm exam (1 lecture)

## Class/laboratory schedule (sessions per week and duration of each session):

Two 75-minute lectures per week.

## Contribution of course to meeting the requirements of Criterion 5 - other:

**Contribution of course to meeting the requirements of Criterion 5 - math and basic sciences:**

**Contribution of course to meeting the requirements of Criterion 5 - engineering topics:**

3 hours.

**Contribution of course to meeting the requirements of Criterion 5 - general education:**

## Relationship of this course to program learning outcomes:

Learning Outcome	Level of Instruction	Related Course Content
Outcome A	Major	Students learn about program locality and how to apply this principle for designing microarchitectural

**Relationship of this course to program learning outcomes:**

<b>Learning Outcome</b>	<b>Level of Instruction</b>	<b>Related Course Content</b>
Outcome B	Major	<p>components such as caches and predictors, and how to extend the principle to new components.</p> <p>Students implement three experimental projects in which they simulate processor components, measure performance across a large design space, present results graphically and textually, explain trends in the results, and base design decisions accordingly.</p> <p>Students implement three projects in which they design processor components with optimal performance and robust behavior across varying workloads, considering cost, complexity, power, and performance trade-offs. Modeling these components from scratch using student-developed simulators additionally stresses creativity and efficient design.</p>
Outcome C	Major	<p>Students implement three projects in which they design processor components with optimal performance and robust behavior across varying workloads, considering cost, complexity, power, and performance trade-offs. Modeling these components from scratch using student-developed simulators additionally stresses creativity and efficient design.</p>
Outcome D	N/A	<p>Students analyze a simulator specification and apply their knowledge from the course to implement</p>
Outcome E	Major	<p>Students analyze a simulator specification and apply their knowledge from the course to implement</p>

**Relationship of this course to program learning outcomes:**

Learning Outcome	Level of Instruction	Related Course Content
Outcome F	N/A	<p>the simulator. Students analyze a microarchitecture and apply their knowledge from the course to identify performance problems and improve its performance.</p>
Outcome G	Major	<p>Students write comprehensive reports to explain their processor component designs, describe their simulator implementations (i.e., procedure), and present and discuss their experimental findings. Students are exposed to microprocessor applications and their positive impact on society.</p>
Outcome H	Basic	<p>Students are exposed to the importance of continuing to scale computing performance (exploiting Moore's Law) as it affects productivity, enables new applications of unprecedented complexity, and enables new scientific discovery.</p> <p>Students are exposed to the problem of power consumption of modern</p>

**Relationship of this course to program learning outcomes:**

<b>Learning Outcome</b>	<b>Level of Instruction</b>	<b>Related Course Content</b>
Outcome I	Basic	high-performance processors and their environmental impact. Students gain a balanced view of performance and power which will guide them as future computer architects.
Outcome J	N/A	Students are exposed to rapid technology evolution to appreciate life-long education.
Outcome K	Major	Students write large programs and use debuggers in three course projects.

**Person who last prepared this description and date of preparation:**

- Rotenberg, Eric (ericro) - Mar 26th, 2010 (06:40pm)