NORTH CAROLINA STATE UNIVERSITY

GRADUATE COURSE ACTION FORM

NOTE: Click once on shaded fields to type data. To check boxes, right click at box, click “Properties”, and click “Checked” under Default Values.

<table>
<thead>
<tr>
<th>DEPARTMENT/PROGRAM</th>
<th>Electrical and Computer Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>COURSE PREFIX/NUMBER</td>
<td>ECE 786</td>
</tr>
<tr>
<td>PREVIOUS PREFIX/NUMBER</td>
<td></td>
</tr>
<tr>
<td>DATE OF LAST ACTION</td>
<td></td>
</tr>
<tr>
<td>COURSE TITLE</td>
<td>Advanced Computer Architecture</td>
</tr>
<tr>
<td>ABBREVIATED TITLE</td>
<td>ADVANCED COMP ARCH</td>
</tr>
<tr>
<td>SCHEDULING</td>
<td>Fall [ ] Spring [x] Summer [x] Every Year [x]</td>
</tr>
<tr>
<td></td>
<td>Alt. Year Odd [ ] Alt. Year Even [x] [ ] Other [ ]</td>
</tr>
<tr>
<td>COURSE OFFERED</td>
<td>BY DISTANCE EDUCATION ONLY [ ] ON CAMPUS ONLY [x]</td>
</tr>
<tr>
<td></td>
<td>BOTH ON CAMPUS AND BY DISTANCE EDUCATION [ ]</td>
</tr>
<tr>
<td>CREDIT HOURS</td>
<td>3</td>
</tr>
<tr>
<td>CONTACT HOURS</td>
<td>Lecture/Recitation 3 [ ] Seminar [ ] Laboratory [ ] Problem [ ]</td>
</tr>
<tr>
<td></td>
<td>Studio [ ] Independent Study/Research [ ] Internship/Practicum/Field Work [ ]</td>
</tr>
<tr>
<td>GRADING</td>
<td>ABCDF [x] S/U [ ]</td>
</tr>
<tr>
<td>INSTRUCTOR (NAME/RANK)</td>
<td>Eric Rotenberg, Associate Professor</td>
</tr>
<tr>
<td>Graduate Faculty Status</td>
<td>Associate [ ] Full [x]</td>
</tr>
<tr>
<td>ANTICIPATED ENROLLMENT</td>
<td>Per semester 20 [ ] Max. per Section 20 [ ] Multiple sections Yes [ ] No [x]</td>
</tr>
<tr>
<td>PREREQUISITE(S)</td>
<td>ECE 521, ECE/CSC 506</td>
</tr>
<tr>
<td>COREQUISITE(S)</td>
<td></td>
</tr>
<tr>
<td>RESTRICTIVE STATEMENT</td>
<td></td>
</tr>
<tr>
<td>CURRICULA/MINORS</td>
<td>Required [ ] Qualified Elective [ ]</td>
</tr>
<tr>
<td>PROPOSED EFFECTIVE DATE</td>
<td>Fall 2006</td>
</tr>
<tr>
<td>APPROVED EFFECTIVE DATE</td>
<td></td>
</tr>
<tr>
<td>CATALOG DESCRIPTION:</td>
<td>In-depth study of research topics in computer architecture; mechanisms and their implementations; advantages and disadvantages of various mechanisms; technology shifts, trends, and constraints.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TYPE OF PROPOSAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Course [x]</td>
</tr>
<tr>
<td>Drop Course [ ]</td>
</tr>
<tr>
<td>Course Revision [ ]</td>
</tr>
<tr>
<td>Dual-Level Course [ ]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RECOMMENDED BY:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department Head/Director of Graduate Programs</td>
</tr>
<tr>
<td>ENDORSED BY:</td>
</tr>
<tr>
<td>Chair, College Graduate Studies Committee</td>
</tr>
<tr>
<td>APPROVED:</td>
</tr>
<tr>
<td>Dean of the Graduate School</td>
</tr>
</tbody>
</table>
Course Justification
Computer architecture is a broad sub-specialty of computer engineering, and NC State’s ECE department is recognized as an international center for research in this field. We have a large number of faculty and students who regularly contribute to the field and publish in premier journals and conferences. The scope of faculty expertise is too broad to be captured in a single advanced course. Therefore, we propose this Advanced Topics in Computer Architecture course to allow faculty to regularly offer in-depth coverage of specialty areas within computer architecture.

The sub-topic for a specific offering of ECE 786 will be chosen by the instructor. The sub-topic will build on the core computer architecture classes, ECE 521 (uniprocessors) and ECE 506 (multiprocessors), and will require an understanding of instruction sets, pipelining, memory hierarchies, and multiprocessor systems. It will also be distinct from existing advanced computer architecture courses: ECE 721 (uniprocessors) and ECE 706 (multiprocessors). Examples of possible sub-topics include: instruction-level parallelism (ILP), power-aware architecture, and architectural support for specific application areas, such as networking, security, or graphics.

The attached syllabus uses instruction-level parallelism as its topic. Computer architecture has a rich history of seminal approaches for exploiting ILP in programs. Time and again, this combined wisdom has been revisited to influence real computers, as rapidly changing technology transforms what is possible. Examples of contemporary trends that rekindle intense interest in parallelism options include: poor scaling of wire delay; power and complexity limits of scaling frequency; transistor performance limits. This course is designed to provide graduate students with in-depth knowledge of ILP paradigms: vector computers, SIMD, dataflow, multithreading, VLIW, superscalar.

Based on the number of active faculty and their students, and enrollment in other advanced computer architecture courses, we anticipate enrollment of approximately 25-30 consistently per year.

Student Learning Objectives
- See syllabus

Previous Enrollment
This course is offered as ECE792S, Spring 2006, and has an enrollment of 13.

New Resources
No new resources are required for offering this course.

Consultation with Other Departments
Computer Science was consulted and has no objections to the course being added.
ECE 786: Advanced Topics in Computer Architecture

Instruction-Level Parallelism

NC State University
Spring 2006

Course Syllabus

<table>
<thead>
<tr>
<th>INSTRUCTOR</th>
<th>OFFICE HRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. Eric Rotenberg <a href="mailto:ericro@ncsu.edu">ericro@ncsu.edu</a></td>
<td>TH 2:30-3:30, Partners I</td>
</tr>
<tr>
<td>Assoc. Professor, ECE Dept.</td>
<td></td>
</tr>
<tr>
<td>Partners I, Suite 2300</td>
<td></td>
</tr>
<tr>
<td>(919) 513-2822</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TA</th>
<th>OFFICE HRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD</td>
<td>TBD</td>
</tr>
</tbody>
</table>

Course Overview

This course addresses advanced topics in computer architectures for exploiting instruction-level parallelism (ILP). The course material will primarily be “classic” papers from the research literature, covering seminal approaches. A representative sample of topics for the course includes:

- Vector computing
- SIMD (single instruction stream, multiple data streams)
- Dataflow programming models and architectures
- Multithreading
- VLIW (very long instruction word)
- Superscalar

Prerequisites

The prerequisites for this class are ECE 521 and ECE/CSC 506. If you have not had these classes, or an equivalent class at another institution, see me immediately. You will need a working knowledge of instruction set architecture, pipelining, memory hierarchies, and multiprocessing before taking this class.

Learning Objectives

Upon completion of this course, a student will be able to:

- describe the potential and limits of instruction-level parallelism;
- describe salient points about vector, SIMD, dataflow, multithreaded, VLIW, and superscalar execution;
- discuss advantages/disadvantages of above approaches;
- compare and contrast above approaches;
- identify workloads amenable to each approach above;
- identify traits of above approaches that have directly or indirectly influenced real designs, past and present;
- predict technology shifts/trends/conditions that may cause described approaches to be revisited.
Course Content

The following is a tentative list of topics and the number of lectures spent discussing each topic (assuming twenty-eight 75-minute lecture periods in a semester). This list is subject to change. Papers below are representative of papers covered in the course – additional papers are likely.

- Early electronic digital computers and the “von Neumann architecture” (2 lectures)
  - Electronic digital computers: ABC (Berry/Atanasoff, Iowa State College), ENIAC (Eckert/Mauchly, Univ. of Penn.)
  - Stored program concept: EDVAC (Eckert/Mauchly, Univ. of Penn.; Burks/Goldstine/von Neumann), EDSAC I & II (Wilkes, Cambridge University)
  - Papers

- Vector processing (3 lectures)
  - Instruction set architectures for vector machines
  - Vectorizing compilers
  - Cray-1 architecture: vector register file, scalar and vector pipelines
  - Performance optimizations: vector chaining, conditional execution, scatter/gather, vector memory architectures
  - Influence on contemporary processors: vector co-processors, micro-SIMD, load string instruction
  - Papers:

- SIMD (2 lectures)
  - Instruction set architectures for SIMD machines
  - ILIAC architecture
  - Influence on contemporary processors: micro-SIMD
  - Papers:

- Dataflow (5 lectures)
  - Data-driven vs. imperative sequencing
  - Functional programming languages and compilers
  - Dataflow architectures (Tagged-Token; Monsoon)
  - Limitations of dataflow: programming model and memory semantics, parallelism explosion, deadlock, technology mismatch
  - Towards multithreading for coarse-grain synchronization
• Influence on contemporary architecture: Wavescalar, restricted dataflow (superscalar), nanoscale architectures

• Papers:

- Multithreading (4 lectures)
  • Latency tolerance through multithreading
  • Various forms: fine-grain multithreading, switch-on-event multithreading, simultaneous multithreading
  • Multithreading for ILP: multiscalar, speculative multithreading, software (implicit) multithreading, towards VLIW
  • Influence on contemporary processors: desktops (Pentium 4), servers (Power 5), embedded processors (Ubicom), network processors (Clearwater)

• Papers:

• Mid-term exam (1 lecture)

• Limits of instruction-level parallelism in ordinary programs (3 lectures)
  • Limit studies (Nicolau & Fisher; Wall; Butler et al.; Lam & Wilson)
  • Bottlenecks: structural, data-flow, control-flow, memory wall
  • ILP taxonomy (Fisher & Rau): Dependence architectures, independence architectures, explicitly-parallel architectures

• Papers:


• Superscalar (4 lectures)
  o Instruction issue logic: in-order execution, out-of-order execution
  o Issuing multiple instructions
  o Speculative execution support: reorder buffers, checkpoint/repair
  o Pipeline optimizations: Instruction fetch
  o Survey of 90s superscalar processors: IBM POWER architecture, MIPS R10K, Ultrasparc, Pentium processors
  o Influence on contemporary processors: desktop (Pentium-Pro, Pentium-4), embedded (embedded PowerPC)

  o Papers

• VLIW (4 lectures)
  o Classic VLIW architectures: Cydrome, Multiflow
  o VLIW instruction sets (IA-64)
  o Acyclic scheduling: trace scheduling, predication, superblocks, hyperblocks, treeregions, tree instruction sets
  o Cyclic scheduling: modulo scheduling, software pipelining
Assignments and Grading

The overall grade will be a weighted average of the following components:

- Midterm Exam (20%)
- Final Exam (20%)
- Presentations (20%)
- Project (40%)

Exams (40%)
The final exam will emphasize material covered after the midterm exam.

Presentations (20%)
Each student will prepare at least two presentations based on papers discussed in class. Presentation grades will include both instructor evaluation and peer evaluation, as well as attendance and participation in the peer evaluation of other student presentations.

Project (40%)
The project for this class will be a journal-quality survey paper, on a topic chosen by the student and approved by the instructor.

Course Grade
The total course grade is a weighted average, with the weights described above.

Resources and Policies

Computer Resources
Course home page: http://courses.ncsu.edu/ece786/lec/001/
Message Board: (see home page)

All class announcements will be posted to the message board. The web site will also contain project assignments and other relevant information. The message board is intended for questions and comments.
about projects, lectures, or anything else. Anyone in the class may post to the message board. If inappropriate material is posted, the message board will be deleted.

I may choose to broadcast an email message to the entire class for time-critical announcements. For the most part, however, you are responsible for getting information in class, from the web site, or from the message board.

**Late Assignments**

Assignments are due at the beginning of class on the specified date. Late assignments will be not be accepted, except for university-excused absences. If you have a medical excuse or instructor approval, you may receive full credit if the assignment is turned in as soon as possible.

http://www.ncsu.edu/policies/academic_affairs/pols_regs/REG205.00.4.php

**Office Hours**

Scheduled office hours are held in my office (Centennial Campus) and are reserved for students in this class. I will also be available after class for questions, as needed. You can drop by my office anytime, but if you want to make sure I’m available, call or email me to arrange an appointment.

**Academic Integrity**

All exams, projects, and presentations are individual assignments, unless otherwise stated in writing. Evidence of cheating, plagiarism, or other violations of the Code of Student Conduct will be investigated and, if appropriate, referred to the Office of Student Conduct for disciplinary review.

Recycling of projects from another class will be considered an academic integrity violation. If you wish to extend or refine the work done for another class project, this must be approved in advance, and you must provide the results from the previous project.

Code of Student Conduct:

Office of Student Conduct:
http://www2.ncsu.edu/student_affairs/osc/

**Inclement Weather**

The class will follow the University’s closure policy. If classes are not cancelled, I will make every effort to be in class on time, and so should you. Please do not send me email asking whether class is going to meet. Instead, check the University website or the weather hotline (513-8888).

**Students with Disabilities**

Reasonable accommodations will be made for students with verifiable disabilities. In order to take advantage of available accommodations, students must register with Disability Services for Students at 1900 Student Health Center, Campus Box 7509, 515-7653.

http://www.ncsu.edu/provost/offices/affirm_action/dss/

For more information on NC State’s policy on working with students with disabilities, please see:
http://www.ncsu.edu/provost/hat/current/appendix/appen_k.html