University of Illinois at Urbana-Champaign
Department of Electrical and Computer Engineering

ECE 511, Fall 2004
Project Overview

The course project is a significant component of ECE 511. The projects are open-ended, and can involve groups of one, two, three or four students (my experience is that groups of two work best). A typical project will involve re-validation of results in an existing paper, extension of an existing paper, and possibly some original research.

Objective

The objective of the project is to give you some experience in conducting architectural research. It is expected that your work be rigorous and insightful; lots of data will not make up for lack of substance. You will write up your results in a paper, written in a style and format similar to the papers in the course reading list. Many of the better projects from past semesters have served as the basis for a Master's thesis, and occasionally, the basis of conference submissions.

Components

The project will have three main components: a proposal, a paper, and a poster presentation. The proposal is a one or two page description of your project. It should contain the following components: a description of the project, an argument as to why the project is important, a description of how the idea will be evaluated, and a brief listing of references to similar work.

As stated, the paper is expected to be styled similar to a typical research paper, with abstract, bibliography, and, if necessary, appendices. The paper will be the major "output" of your efforts on this project. The paper should be as long as it needs to be, but a good rule of thumb would be approximately 4000-6000 words.

We will have class presentations in the form of a poster session at the end of the semester. I will invite other faculty and graduate students at UIUC interested in computer architecture to attend. We will discuss the format of the poster session as the end of the semester approaches.

Schedule:
Proposals are due: Wednesday, November 3rd, 2004
Status reports due: Monday, November 29th, 2004
Poster session: Friday, December 10th, 2004
Paper due: Saturday, December 18th, 2004

Grading

The projects will be graded on several bases, including: (1) problem definition and motivation, (2) insight, (3) carefulness of experimental evaluation, (4) creativity and approach, (5) organization and clarity of the paper.

Resources

For experimental projects, you will need to build a model representing your ideas in order to measure their effectiveness. You will also need to consider what types of inputs to apply to your

model. There are many resources available to help you get started. Some are listed below, others can be found on the Computer Architecture Homepage : http://www.cs.wisc.edu/~arch/www

Benchmarks	SPEC95 SPEC2000 MediaBench
Processor Simulators	SimpleScalar Simulator SimAlpha ISA model Course simulator

More information will be provided on these resources as the term progresses. I will provide access to some of these things. Some of them you will need to investigate and find yourselves.

Project Ideas.

The following list is by no means complete. Its purpose is to get you thinking on the right track.

- New ways to use threads. Many new microarchitectures are providing support for threading. How can an application that is not threaded make use of a threading support?
- New, creative ways to predict branches, or to reduce interference in existing predictors. Including: Branch characterization. What types of branches are predictable, which are not? Is there a "theory" behind branch prediction? How predictable are branches in the limit? How does one predict indirect jumps?
- Ways to deal with memory latency; prefetching; adaptive memory systems
- New paradigms for reliable hardware operation; fault-tolerance; error detection and recovery
- Power-efficient or energy-aware microarchitectures that provide high-performance, too
- Compiler transformations for EPIC architectures. Hyperblock formation, predication; tradeoffs – when to predicate, when not to predicate. Co-existing predication with prediction
- Dynamic optimization systems
- Predictive architectures; address prediction, value prediction, reuse.
- New techniques for uncovering/exploiting Instruction-Level Parallelism
- Complexity effective processor techniques; simplify an existing microarchitecture
- High-bandwidth cache design; techniques for splitting the data stream
- Trace caches, high-bandwidth fetch
- Parallelization of a particular application for a parallel machine; use a network of computers (EWS or Turing) to test out your ideas.
- Architectural support for networks, encryption, media, speech processing, DSP, etc…
- Applications analysis; data set size, frequent operations, branch predictability
- Processor/Memory integration
- Automatically extracting course-grained parallelism from a regular program; hardware or software
- Simulation methodologies – how to simulate a complex microarchitecture in reasonable time
- Reconfigurable architectures