Department of Electrical and Computer Engineering
University of Texas at Austin

ECE 382N, Spring 2002
Y. N. Patt, Dave Armstrong
January 14, 2002
Overview

Instructor: Y. N. Patt, 541a ENS. Telephone: 471-4085, or patt@ece.utexas.edu.
TA: Dave Armstrong, 532 ENS Building, Telephone: 471-6814, or dna@ece.utexas.edu.

Class meets: 5 to 6:30 pm, MW, in (until further notice): CPE 2.206. Discussion Section: Tuesday, 5 to 6:30pm in (also, until further notice) ENS 126. (You will note from the course syllabus that I will sometimes use Tuesday for lecturing and Monday for Discussion section, depending on what I think makes sense during any particular week. I may also substitute a guest lecture if someone with an important message happens to be in Austin and available on a particular day. Bottom line is that I expect you to plan on being available all three days (MTW) for lecture or discussion. Please do not sign up for this course unless you are able to make all three class meetings each week.)

Objectives of the course:

EE 382N is intended to provide a solid introduction to microarchitecture to the serious graduate student who is interested either in PhD research in microarchitecture or an industrial position on a leading edge microarchitecture project. We expect to do that in two ways:

(1) Each student will participate as a member of a design team to complete a substantial design of a cpu for a subset of a commercially available modern microprocessor. We will use Intel's IA 32 (nee x86) ISA as our starting point. Each team will start with a clean sheet of paper and will design the data path, microsequencer, microprogrammed or hardwired control, microcode or logic, as appropriate, interface to memory and I/O, and the selection and interconnection of parts to implement all of the above. The design will be done at the logic gate level, in structural level Verilog where the design will be concerned with timing issues (propagation delay, cycle time). The design may be an aggressive pipeline, or a more conservative microarchitecture, at the discretion of the design team. Our expectation is that each student will come out of this experience more fully appreciating the problems that come up in designing the microarchitecture for a general purpose ISA.

(2) Lectures, in addition to dealing with design issues relevant to the project, will provide in depth coverage of the latest hot topics in high performance microarchitecture, and an awareness and appreciation of the field of computer architecture, particularly alternative design styles and implementation tradeoffs. We will deal with problems involving instruction supply, data supply, and instruction processing, compile-time vs. run-time tradeoffs, very aggressive branch prediction, wide-issue processors, in-order vs. out-of-order execution, instruction retirement. Case studies will be taken mostly from current microprocessors, although we may examine (as time allows) a classical older implementation.

Relevance of the course.

This course provides a fundamental body of knowledge useful to graduate students who plan to do PhD research in microarchitecture or who plan to seek employment in the microprocessor industry upon completion of their degree.

With respect to PhD research, several major IEEE and ACM conferences deal specifically with research results from this field, including ISCA, ASPLOS, Micro-n, HPCA, and PACT. Several prestigious journals publish research based on the foundation material taught in this course. There does not appear to be any lessening of interest in this material in the research community.

With respect to the microprocessor industry, companies seek graduates who have the insights acquired from this course. Many major employers of our graduates (Intel, for example) have an increasing need for

graduates who have these insights.

Where I am coming from.

An outline of some of the topics we will cover is included below. We will undoubtedly not get to all of them for several reasons: (1) there is too much here to cover in one semester. (2) "covering" the material is not something I particularly aspire to. Furthermore, we will probably not cover the topics in the order listed in the course syllabus, regardless how much I plan to.

My objective in our class meetings is to explore ideas that will be useful to your future research and/or your future work in industry. My view of research is that if you know the outcome before you start the project, then I am not interested in the work as "research." I suspect that many of our class meetings will follow some unintended path as we explore dynamically some issue that comes up. I want you to think critically about what you read, and explore creatively what might be possible. If that causes us to spend three times as long on a topic as we might otherwise if we covered the topic from my notes, it will not make me unhappy. If we get the material from my notes to yours without going through the brains of either of us, that will make me very unhappy.

Lest you think this is intended to encourage wild-eyed departures from fundamental knowledge, let me assure you that the one thing we always try to do is tie things to the fundamentals. My hope is to encourage you to combine mastery of the fundamentals, critical reading and analysis, and creative thinking.

CAD Tools: For the project, we will be using a modern set of CAD design tools, provided by Synopsys, which use the Verilog design language. We have put together sufficient introductory material and examples to help you get started with these tools. Mastery of the tools is not an end in itself; on the contrary, the tools are expected to be a means to enhance your productivity in completing the project. **You are encouraged to help each other master the tools,** so that we can all get on with the business of carrying out our designs.

Prerequisites: Satisfactory completion of courses covering the material of 319K and 360N.

Caveat: My experience from teaching this course has been that the design project requires a much larger amount of time to complete than most students expect to be the case in the beginning. If this semester goes as the ones before it, you will be pleased with what you have accomplished after the term is over. But during the term, sometimes after a few consecutive sleepless nights, you may wonder what lapse in sanity caused you to sign up. Please consider this as you organize your workload for the semester.

Grading: Three items will contribute to your grade in this course: the design project, scores on the two mid-term exams, and homework and problem sets. They will be weighted, approximately as follows:

exams, 42%
project, 42%
homework, problem sets, etc., 10%
other, 6%

Office hours: MTW right after class, plus other times as you need them.

References:

There is no required text. References will be suggested where appropriate, depending on the topic. I expect to provide handouts on additional material. Also, Some of the lectures will use transparencies. In those cases, you will be provided with copies of the transparencies.

For those of you who decide to continue, Good Luck. I hope you find the experience an important part of your computer engineering education. I also hope you have a good time doing it.

Some possible lecture topics.

1. Fundamental properties of microarchitecture.

    Instruction supply
    Instruction processing
    Data supply
    Control

2. Basic concepts

    Critical path design
    Pipelining, superpipelining, superscalar
    Bread and butter design
    Partitioning of functionality
    Role of Microprogramming in 1996
    Native mode vs. emulation
    Approaches to concurrency
    Architectural choices
    Support for multiprogramming
    Support for multiprocessing

3. Fundamental paradigms.

    SIMD, MIMD, SPMD
    Vector processing
    VLIW, DAE
    VLIW vs. Superscalar
    HPS (superscalar, dynamic scheduling, precise exceptions)
    The Multiscalar approach

4. Measurement Methodology

    SPEC 2000 benchmarks
    Other methods of benchmarking
    Abuse of statistics

5. Instruction supply mechanisms

    Removing pipeline stalls
      Redirection determination
      Target determination
    Branch Prediction (static, dynamic)
    Predicated execution
    Multiple decode
    Post-decode caches, trace caches
    Methods for approximating perfect caches
    Methods for approximating perfect branch prediction

6. Data Processing

    Block-structured ISA
    Dependency checking
    Function unit capability

Internal communication (bypass mechanisms)
Result distribution
State maintenance mechanisms
(checkpoint, reorder buffer, history buffer)
Retirement (precise vs. imprecise exceptions)
Methods for approximating perfect data flow

7. Data Supply mechanisms

Cache Memory
Alternative characteristics
New approaches to cache structure
Memory disambiguation
Mechanisms for dealing with memory contention
Functional unit capabilities
Pin bandwidth problem
Impact of processor/memory cycle time disparity

8. Influence of ISA on performance tradeoffs.

9. Compile time/run time tradeoffs.

10. Computer Arithmetic

Fast vs. Correct arithmetic
The IEEE Floating point standard
Impact of the Floating point standard on performance

11. Influence of a Multiprocessor Environment

12. Influence of the I/O subsystem

13. Influence of the Application environment

Strictly integer code
Scientific computation
Multimedia applications

14. Case studies

Detailed study of a classic microprogrammed machine
Detailed study of a current microprocessor implementation