CMCS 611-101
Advanced Computer Architecture

Lecture 1

Introduction and Overview

September 2, 2009

www.csee.umbc.edu/~younis/CMSC611/
CMSC611.htm

Lecture’s Overview

- Course resources, syllabus and work load
- Grade structure and policy
- Teaching style and philosophy
- An introduction to computer architecture
- The importance of studying computer architecture
- Organization and anatomy of computers
- The impact of microelectronics technology on computers
- The evolution of the computer industry and generations

Course Resources

Instructor: **Dr. Mohamed Younis**

Office: ITE 318 E-mail: younis@cs.umbc.edu

Office hours: Monday and Wednesday 3:00 PM - 4:00 PM

Research interest:

Wireless Ad-hoc and Sensor Networks Real-time systems, Fault tolerant computing, Compiler-based analysis, Tool support for embedded systems

TA: **Mr. David Walser**

Office: ITE 349 E-mail: walser1@umbc.edu

Office hours: Tuesday and Thursday 11:30AM - 1:00 PM

Textbook:

Computer Architecture: A Quantitative Approach, *3rd or 4th Edition*

John L. Hennessy and David A. Patterson

Morgan Kaufmann Publishers, ISBN 1-55860-329-8

Web page: www.csee.umbc.edu/~younis/CMSC611/CMSC611.htm

Instructor will stay after class to answer questions

Course Goals

- To learn the organizational paradigms that determine the capabilities and performance of computer systems
- To understand the interactions between the computer's architecture and its software so that
  - future software designers (compiler writers, operating system designers, database programmers, …) can achieve the best cost-performance trade-offs
  - future architects understand the effects of their design choices on software applications
- To know some of the advanced design features on modern processors that boost the performance
- To understand contemporary design issues and how to conduct a trade-off among the various design objectives

* Slide is (partially) courtesy of Mary Jane Irwin

Course Syllabus

1. Quantitative Design Principles

- Technology and Usage Trends
- Cost and Trends in Cost
- Measuring & Reporting Performance
- Benchmarks and metrics

2. Instr. Set Principles & Examples

- Classification of Instruction Set Arch.
- Instruction Formats and Semantics

3. Pipelining & Instr. Parallelism

- Basic Pipeline Operations
- Data and Control Pipeline Hazards
- Instruction-Level Parallelism
- Dynamic Instruction Scheduling

4. Memory-Hierarchy Design

- Cache Design Issues
- Virtual Memory Addressing
- Memory Protection Mechanisms

5. Storage Systems

- Types of Storage Devices
- Connecting I/O Devices
- I/O Performance Measures
- Reliability, Availability, and RAID
- Interfacing to an Operating System

6. Interconnection Networks

- Interconnection Network Media
- Connecting Multiple Computers
- Practical Issues and Examples

7. Multiprocessor Systems (time permitting)

- Application Characteristics
- Centralized Shared-memory Arch.
- Distributed Shared-memory Arch.
- Execution Synchronization
- Models of Memory Consistency

Course Workload

❑ ***Assignments***

- 3 assignments will be given and normalized to %15 of the final grade
- An average assignments requires about 2-3 hours to perform
- Assignments are due in class on the due date (not later)

❑ ***Exams***

- A midterm exam is scheduled on October 28th during regular class time
- A final exam is scheduled on December 21st during UMBC specified hours
- The scopes of exams do not overlap (Final exam is not comprehensive)

❑ ***Projects***

- A design project will be given contributing 25% to the final grades
- The project involves architecture simulation and performance analysis
- The project is to be implemented using the programming language of your choice, given that it your submit a "makefile" that automate the compilation of the source files
- Project must be finished and submitted on time to earn a grade

Grade structure and policy

	Grade distribution
Final Exam	30%
Mid-term Exam	30%
Project	25%
Homework	15%

Course grade	Range
A	90% - 100%
B	80% - 89.9%
C	70% - 79.9%
D	60% - 69.9%

- Assignments are due in class (*Late assignments are not accepted*)
- UMBC rules apply to cheating/copying
  - You may work together and discuss homework and the project.
  - You must do your own work and not copy from anyone else
  - You better off skipping an assignment or get a partial credit
- Copying/cheating will result in a minimum punishment of a zero grade for the assignment or project

Teaching Style and Philosophy

❑ *Instructor’s role*

- Facilitate and guide the students to the fundamental concepts
- Make it simple and elaborate with examples
- Relate as much as possible to available products
- Prepare class notes to be as rich and comprehensive as possible

❑ *Student’s role*

- Focus on understanding and digesting the concept
- Do not worry about the grade more than concepts; you are a professional !!
- Slow down the instructor if you do not understand and raise questions
- Be prepared to answer an oral quiz, when you get involved in a side talk

❑ *TA’s role*

- Help students with questions related to their assignments
- Resolve computer and tool issues related to projects
- Grade assignments and projects

Exams will question the level of understanding of fundamental concepts