UNIVERSITY OF DETROIT MERCY
COLLEGE OF ENGINEERING AND SCIENCE
Electrical and Computer Engineering
EE 480/580 COMPUTER ARCHITECTURE
TERM II 2005

1998 Catalog Data: The design of Computational Systems and circuits. Investigations of Alternative Structures for Computers.

Textbook: *Computer Organization & Design: The Hardware/Software Interface, 3rd Edition.* David A Patterson & John L. Hennessey. Morgan Kaufman Publishers, 2004.

References:

1. *Computer Architecture and Organization, 4th Ed*., by Hayes. McGraw-Hill, 1998.

2. *Computer Organization, 6th Ed.*, by Hamacher, V. C., Vranesic, Z. G., and Zaky, S.G. McGraw-Hill, 2004.

3. *Computer Organization and Architecture, 6th Ed*., Stallings, William. Prentice Hall, 2004

4. *Structured Computer Organization, 4th Ed*., Tanenbaum, Andrew S.

Goal: To show the relationship between hardware and software and to focus on the concepts that are the basis of current computers. This course will provide an in-depth understanding of the inner workings of modern digital computer systems and tradeoffs present the hardware-software interface. You will get an understanding of the design process in the context of a complex hardware system.

Prerequisite by Topic:

1. Digital Logic Circuits
2. Computer Programming
3. Assembly language programming

Instructor: Dr. Nizar Al-Halou, Professor
Office and Contact: E 330, 313-993-3365, alholoun@udmercy.edu

Office Hours: As posted. Other hours are by appointment.

Withdraw Dates: February 4, 05 (without a "W") and April 1, 05 (with a "W")

Electronic Bulletin Board: There will be Electronic newsgroup to post announcements and to allow students to post questions or discussion and homework help at http://knowledge.udmercy.edu. Special announcements may be made in the lectures and at the website. Students should visit the web site frequently. These announcements may overwrite previous handouts and announcements. Students are responsible for material made in announcements.

Tentative Grading:

1.	Midterm	20%, February 20, 2005
2.	Project Report(s), poster and Presentation(s)	20%
3.	Final	40%, April 26, 2005
4.	Homework & Quizzes	20%

Graduate Students are expected to answer extra exam questions, and write an additional report.

Academic Integrity: Students are expected to conform to high standards of honesty and integrity in this course. Please refer to the University Catalog and E&S Student Handbook for further explanation of academic integrity.

Homework Assignments: Homework assignments will be given periodically in lecture. Assignments are due at the time announced. Assignments are due at the beginning of the lecture period when it is due and will not be accepted after the due date. Homework assignments are provided to reinforce or expand on topics discussed in lecture. They may also touch on topics that will be covered in exams. It is to your advantage to work on the problems yourself. You are encouraged to discuss homework problems with your classmates. However, everyone must write up their own unique solution. Any question regarding the grading of a problem set may be raised within a week of its return date.

Learning Outcomes: Upon the successful completion of this course, you will be able to:

1. Understand the basic components of the van Neumann computer architecture.
2. Write MIPS Assembly Language Programming
3. Design arithmetic logic unit (ALU)
4. Perform arithmetic operations using the 2's complement and IEEE floating point notation
5. Design control unit (CU)
6. Evaluate the performance of different architectures
7. Enhance the performance using advanced techniques such as pipeline, memory hierarchy
8. Design I/O sub-systems

Topics:

1. Computer Abstractions and Technology: Interaction between computer hardware and software, performance, advances in speed, capability, and cost of processors and memory.
2. Instructions: Language of the Machine, Operations and operands, MIPS addressing
3. The Role of Performance: Program execution and performance, cost/performance, MIPS, MFLOPS, & Benchmarks
4. The Processor: Datapath and Control Datapath, multicycle implementation of the MIPS instruction set, execution steps, and control, hardwired vs. Microprogrammed control unit
5. Large and Fast: Exploiting Memory Hierarchy, General principles, main memory, caches, and virtual memory.
6. Interfacing processors and peripherals: Types and characteristics of I/O devices, I/O performance measures, buses, I/O interface to the memory, processor, and operating system, I/O interrupts, DMA, and handshaking

7. Enhancing performance with Piplelining, Basic pipelining principles, pipelined datapath and control, hazards, pipeline implementation problems, performance of pipelined systems
8. Multiprocessors: Programming multiprocessors, multiprocessors connected by a single bus, multiprocessors connected by a network, clusters.

Project

The most significant work you will do in this course will be a semester project, which will count as 20% of your grade. The project consists of two phases:

Phase I:

Discuss what you are going to do in the project and the expected outcome. This step will help you in the second phase of the project. As soon as you select a home page, I'll reserve it for you. It must be done by 1-20-05.

The presentation is scheduled on 1-27-05 with a power point report. The presentation may include overview, objective, expected outcome and deliverable.

Phase II:

Select a project. The project should be about Computer Organization and Architecture and may include: design issues, hardware/software implementation, cache coherency, CPU design & prototyping using FPGA, fast ALU design, design evaluation and demonstration, fuzzy logic based cache, fuzzy logic based virtual memory management.

- Title and summary of Project is due 2/3/05
- Presentation and report is due 4/5/05. The report should include a hard copy, a softcopy in Microsoft word and HTML, a poster and a copy of the presentation in Power Point.

Report Format

The format for the final project report is as follows:

1. Title of the paper, student name, date, instructor name, and department
2. Abstract
3. Key Words—three to nine major key words
4. Introduction
   a. Problem Statement
   b. Background-state of the art of the problem at hand
   c. Organization of the paper
5. The body of the paper
6. Results
7. Simulation (if necessary)
8. Conclusion
9. References (All references must be referenced in the paper using IEEE style)

This course presents the principles, characteristics, and trends of computer systems design at a level appropriate for all computer scientists and computer engineers. It expands on the role of a traditional computer architecture course, focusing largely on the hardware design, to include aspects of the complete system, comprising the hardware, operating system, compilers, and application software.

The intention is to present material that will benefit all computer scientists, even those who are more concerned with software development than with hardware design. By learning more about what is inside "the box," software developers can make more effective use of computers, often achieving dramatic performance gains.

Students will also gain an understanding of the hardware technology that has fueled the rapid progress of computer systems. Since the technology will continue to change and improve, it is important to see how these changes will affect the characteristics of future computer systems.

After completing the course, students should have gained a firm grounding in the following:

Machine level programming: With the advent of optimizing compilers, there is seldom need to write large amounts of assembly code. On the other hand, it is important to be able to understand the code generated by those compilers, and to write small sections of code. We will use the MIPS instruction set as our main case study, mostly using assembly code generated by the C compilers.

Instruction Set Design: We will study the general characteristics of several instruction sets, including the Intel x86 and the IBM PowerPC.

Processor Implementation: The design of the internal components of a processor's datapath and control will be covered. A non-pipelined implementation of a processor using the MIPS instruction set will be used as basis for this discussion.

Memory Subsystem: The combination of primary memory, caches, and virtual memory are intended to provide the illusion of high speed access to a large memory space. Understanding how this subsystem operates is critical to efficiently implementing applications with high memory requirements.

I/O and Storage Subsystem: With the advent of multimedia applications, we are seeing stringent, real-time requirements placed on the various I/O and display devices, as well as the busses connecting them to the CPU. Understanding the organization and performance characteristics of this subsystem is therefore critical to current and future software developers.

Parallel Computing: Increasing application performance by using resources operating in parallel has been attempted in a number of different ways for nearly 50 years. Although the track record for these efforts is mixed, there are many indications that parallelism will play an increasingly critical role as the technology and the techniques for exploiting it mature.