

## EE573/673 COMPUTER ORGANIZATION AND DESIGN

### Course Information

Course	EE 573/673
Credits	4
Lectures	Tue / Thu 5:30 – 7:10PM
Location	Wilson Clark Center, Room 403
Instructors	John Lynch Roy Kravitz
How to reach us	John: <a href="mailto:jdlynch@csee.ogi.edu">jdlynch@csee.ogi.edu</a> 503.748.7305 Roy: <a href="mailto:roy.kravitz@radisys.com">roy.kravitz@radisys.com</a> 503.615.1280
Office Hours	Roy: By Appointment (call or email to arrange) John: Mon. – Thu. 4:00 - 5:00pm (BCB160K)
Course Web Page	<a href="http://www.csee.ogi.edu/class/ee573">http://www.csee.ogi.edu/class/ee573</a>

### Catalog Description

Basic computer organization: memory hierarchy, including caches, computer arithmetic, pipelining, number representation, floating point arithmetic processors, controllers, input/output, buses, DMA. Data formats, addressing modes, instruction sets, and microcode. This course fills the gap between the CSEE Computer Engineering and Design courses and CSE 521 - Introduction to Computer Architecture. Prerequisites: EE 570 recommended. (4 credits)

### Overview

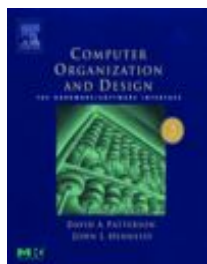
EE 573 teaches the student how modern digital computers are organized and controlled. It builds on the concepts learned in EE 570 by showing how the logic gates and building blocks (flip-flops, registers, state machines, multiplexers, etc.) can be interconnected to form registers, the data paths that connect them, and the control circuits needed to manage the data flow. In addition, the course provides a basic introduction to the Instruction Set Architecture (ISA) and assembly level programming. By focusing on computer hardware, the class serves as a “bridge” class to the CSEE computer architecture courses.

The course treats a computer as a hierarchy of levels, each one performing a well-defined function. We will start by introducing the basic components of a modern computer (processor, memories, and input/output devices). Once we’ve completed that discussion we will examine the digital computer in detail at several levels, with particular attention paid to RTL / microarchitecture. Specifically, we will examine the architecture and implementation of MIPS 3000 introduced in the textbook. The course will also discuss the memory hierarchy and hardware-based techniques such as pipelining and caching that are routinely used to increase instruction throughput. Detailed pipeline optimization and multiple instruction issue strategies are not covered in this course – they will be addressed in detail in CSE 521 – Introduction to Computer Architecture.

### **Prerequisites**

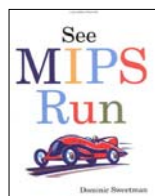
There are no prerequisites for this course, but EE 570 or equivalent is strongly recommended (EE570 may be taken concurrently). Students taking this class should have general experience programming computers since we will be looking at computers from an instruction set level. Students should be comfortable programming in a high level programming language, (preferably C or C++) to complete one of the class projects.

### **Required Text:**



*Computer Organization and Design, Third edition*  
David A. Patterson and John L. Hennessy  
Publisher: Morgan Kaufmann, 2004  
ISBN: 1558606041

### **Reference Text:**



*See MIPS Run*  
Dominic Sweetman  
Publisher: Morgan Kaufmann, 1999  
ISBN: 1558604103

### **Details**

The course will be co-taught by Roy Kravitz and John Lynch, with the instructors generally alternating between weeks.

Homework will generally be assigned once a week and will be due the same day the following week. Late homework will be accepted with a valid excuse (such as company travel), but, if at all possible, that excuse must be given before the homework is due. Late homework with no valid excuse will be accepted for up to a week after the due date. The penalty for unexcused late homework will be 2% deducted a day.

In addition to weekly homework assignments, students will complete two programming projects. The first will be a MIPS assembly-language programming project using the SPIM simulator. The second project will be to design and implement a PDP-8 microsimulator using a general purpose programming language such as C or C++. These projects will not be overly difficult, as computer programs go, but difficult enough to provide insight into how computers are implemented.

There will be a midterm and a final exam. There may be “pop quizzes” as the instructors see fit. Pop quizzes will be included as part of your homework score. We will try to accommodate students who cannot take the exams when scheduled if they have a valid excuse and make arrangements in advance.

Lecture notes and slides, as well as other documentation, will be posted on the course web site:  
<http://www.csee.ogi.edu/class/ee573>

### **Grading**

Grading will be as follows:

Homework	40%
Projects	15%
Midterm exam	20%
Final exam	25%

**This course moves at a very fast pace. Students will be required to demonstrate a significant level of competency in the basics skills being taught to receive a grade of B or higher.**

---

### ***Tentative Lesson Plan***

#### Week 1 (Sept. 27, 29)

Instructor: John (Tue) and Roy (Thu)

Reading: COD Chapters 1 & 4

Tue: Course overview; computer abstractions and technology; computer history

Thu: Assessing and understanding performance, measures, benchmarks

#### Week 2 (Oct. 4, 6)

Instructor: Roy

Reading: COD Chapter 2

Tue: Basic instruction sets, introduction to MIPS 3000 architecture and assembly code

Thu: More MIPS instruction set architecture, data addressing, load/store, indexing and arrays, branching, compiling MIPS code, SPIM simulator

Project: Assign MIPS assembly language project (Thu)

#### Week 3 (Oct. 11, 13)

Instructor: John

Reading: COD Chapter 3, Appendix B

Tue: Data representation, signed and unsigned numbers, addition, subtraction, multiplication

Thu: Division, floating point arithmetic

#### Week 4 (Oct. 18, 20)

Instructor: John

Reading: COD Chapter 5

Tue: Data path and control, multiple cycle implementations

Thu: Exceptions, microprogrammed control, design with an HDL

Week 5 (Oct. 25, 27)

Instructor: John and Roy

Reading: None

Tue: Review for Midterm

Thu: **\*\*\*MIDTERM EXAM\*\*\***

Project: MIPS assembly language project **due Tuesday**

Week 6 (Nov. 1, 3)

Instructor: Roy

Reading: COD 6.1-6.11, Appendix C

Tue: Go over exam solutions; Pipelined data path

Thu: Data hazards and forwarding, advanced pipelining

Week 7 (Nov. 8, 10)

Instructor: John

Reading: COD 7.1-7.5

Tue: Memory hierarchy, caches

Thu: Virtual memory

Week 8 (Nov. 15, 17)

Instructor: Roy

Reading: COD Chapter 8, PDP-8 reading

Tue: Introduction to PDP-8 and microsimulation

Thu: Input/output, external storage, buses

Project: Assign PDP-8 microsimulator project (Tue)

Week 9 (Nov. 22, 24)

Instructor: Roy

Reading: None

Tue: Q/A session on microsimulator, catch-up (optional)

Thu: **No class, Thanksgiving**

Week 10 (Nov. 29, Dec. 1)

Instructor: John and Roy available for consultation

Reading: None

Topics: **No class, Project**

Week 11 (Dec. 6, 8)

Instructor: John and Roy

Tue: Review for final exam

Thu: **\*\*\*FINAL EXAM\*\*\***

Project: Microsimulator project **due Friday 12/9**