

CIS 216 Computer Organization and Design

2:00 – 2:50 PM, MWF

Spring 2007

Instructor: Dr. Alfred R. Watkins
Office: Tapley 317
Phone: 404.270.5768
Email: alwatkins@spelman.edu
Web Page: <http://www.spelman.edu/~compsci/cis216>
Office Hours: Tuesday/Thursday 2:30--3:45 and Friday 3:00--4:00

Description and Prerequisites

This course gives the student an introduction to the theoretical foundations of computer science with emphasis on topics such as finite state machines, formal languages, boolean logic and combinatorics. It also guides a study of the vision of the computer science field with exposure to the breadth of the basic areas of study. **Prerequisite:** CIS 122 or consent of the department.

Purpose

This course explains the basic components and inner workings of a computer, from the stored-program concept and the machine cycle, to the representation of code and data. By discussing the Instruction Set Architecture, it bridges the gap between hardware and software.

This course will also provide the students with an understanding of combinational and sequential logic gates and the internal operation of modern computer hardware. The emphasis is on the functional logic design level, and the mechanics of data transfer and control, but reinforces the theory with hands-on circuit design and implementation.

Objectives

At a very primitive level, computers work by executing machine language: programs represented in binary. In this course, we will

study the more readable counterpart of machine language called assembly language. We will also look at the hardware implementation of assembly language, building up from logic gates to flip-flops, registers, and, ultimately to complete computer architectures. We will also see how the architecture is designed for usability, efficiency, and speed.

Specifically, by the end of the course, you will be able to do the following:

- Describe how data structures are represented in binary--- objects such as numbers, character strings, and records, and operate on them using assembly language.
- Explain the Instruction Set Architecture and its importance in the design of computer systems.
- Translate high-level programming statements into their assembly language and machine language equivalents.
- Determine sum-of-products Boolean algebra expressions to compute functions described by truth tables over up to four variables, simplify those expressions using identities of boolean algebra, and design combinational digital logic circuits to implement the algebra.
- Describe the design of the ALU and the processor, datapath and control, including being able to give the values of control

lines to implement instructions and augmenting datapaths in simple ways to add new functionality.

- Explain micro-code at a conceptual level and give advantages and disadvantages of its use.
- Sketch the organization of the central processing unit, input/output devices and memory of modern computers.
- Describe the general architecture of the MIPS architecture, with emphasis on pipelining.
- Calculate the execution times of programs based on CPI of different instruction classes and percentage of instructions, with and without pipelining, and with and without cache misses. You'll be able to explain the design goals and tradeoffs of various aspects of computer architecture.
- Explain how caches work and how virtual memory works. Compute where an item will go in memory or cache, and whether there is a hit or miss.
- Define the premise of advanced architectures, such as SIMD and MIMD.

This course prepares students for more advanced courses such as compiler design, parallel processing and computer architecture.

Instruction

Instruction will consist mostly of lectures based on the text. We will use SPIM, a simulator of the MIPS architecture, for executing and debugging programs in MIPS assembly language. There will

required laboratory sessions, meeting in the Sun lab (Room 223). These lab sessions will cover either assembly language programming in SPIM or digital logic using black box design kits or a software simulator.

There will be several assembly language programming assignments during the semester. Assembly language programming will appear on the tests. There will be a final project done in groups.

The material to be covered in each lecture is given below. Students are expected to have read the assigned material prior to class.

Textbook

Computer Organization and Design: The Hardware/Software Interface 3rd Edition by David A. Patterson and John L. Hennessy, published by Morgan Kaufmann, 2004, ISBN 1558606041.

Requirements

There will be three in-class tests (about one a month) and a final exam. The dates of the in-class exams are tentatively scheduled below. The tests are cumulative; they will cover all material given in the lectures up to the test date. The final exam will cover all course material. There will be no make-up exams except for excused absences where I have been notified **in advance**.

The preceding components of your grade are weighted as follows:

Component	Percentage
<i>in-class tests</i>	35%
<i>final exam</i>	25%
<i>project assignments</i>	15%
<i>lab/homework /quizzes</i>	10%
<i>final project</i>	15%

Course Outline

Students are expected to come to class having read the material for the week.

Week	Reading	Topics
1	Ch.1	Computer Abstractions and Technology
2	Ch.2 & App.A	Instructions: Language of the Computer
3	Ch.2 & App.A	Instructions: Language of the Computer
4	Ch.3 & App.B	Arithmetic for Computers Test 1
5	Ch.3 & App.B	Arithmetic for Computers Lab Reading
6	Ch.3 & App.B	Arithmetic for Computers
7	Ch.4	Assessing and Understanding Performance Test 2
8	Ch.4	Assessing and Understanding Performance
9	Ch.5 & App.C	The Processor: Datapath and Control
10	Ch.5 & App.C	The Processor: Datapath and Control
11	Ch.6	Enhancing Performance with Pipelining
12	Ch.6	Enhancing Performance with Pipelining Test 3
13	Ch.7	Large and Fast: Exploiting Memory Hierarchy
14	Ch.7	Large and Fast: Exploiting Memory Hierarchy Final Project
15	1-7	Final Exam

Tests are currently scheduled as follows:

Test 1, Friday, February 9th
 Test 2, Friday, March 2nd
 Test 3, Friday, April 13th
 Final, Monday, May 7th, 10:30am

Program and homework due dates will be announced when they are assigned. Changes to the schedule will be announced in class and posted to the class website.

Policies

Attendance. Although your attendance is not mandatory, it is to your own benefit.

It will be your responsibility to get the information missed due to your absence from class. You are expected to bring your textbook to all lecture and lab classes.

In cases of more than a full week of absences, the instructor maintains the right to administratively withdraw a student who has missed two or more weeks of class without a Dean's Office excuse. The student should be aware that the Dean's Office will not intervene in these situations.

Academic Dishonesty (Cheating)
 Cheating constitutes any case in which a student receives unauthorized aid in his/her work contributing to a course grade with the intent to deceive. If another student gives the unauthorized aid, that student is considered to have cheated as well. Violators will be dealt with according to the policies of Spelman College; therefore it is the student's responsibility to become familiar with the rules and regulations of the College. If you are unsure of whether you are in violation, re-evaluate your ethics. If all else fails, ask!