

# CS410 -- Computer Architecture

## Course Syllabus and Calendar – Winter 2008

Assistant Professor Timothy Stanley, PhD

Brigham Young University Hawaii

### **Brief Overview**

In compute organization (CS210) we learned about the lowest levels of computing, transistors that make gates that giving functions such as AND and OR, combining these gates to produce modules like an adder, and numbering systems such as BASE 2. We also learned about how a computer can be programmed at the lowest levels in machine and assembly languages. In Computer Architecture we solidify this knowledge through designing a computer from gates. This includes designing an instruction set and assembling the virtual hardware to accomplish the instruction set, and producing simple programs to demonstrate your computer. Through this exercise you will really understand what is going on under the hood of a computer, and really understand what your program needs to do, and how it does it.

We will also learn about techniques to improve speed and efficiency through techniques like pipelining and use of cash memory. Our computer designs will be implemented Mulimedia Logic, freeware available from <http://www.softronix.com/>. You will be provided with many example computer designs to start with as well as suggestions to improve designs.

### **The Course**

Course Number: CS-410

Title: Computer Architecture

Course Description: Memory system organization and architecture, interfacing and communications, data paths and instruction decoding, multiprocessing, pipelining, and alternative architectures. (Prerequisite: CS-210.)

Textbook: Computer Organization and Design, The Essentials of Computer Organization and Architecture, Second Edition by Linda Null and Julia Lobur, ISBN-13: 978-0-7637-3769-6

Reference Material: Computer Organization and Design, The Hardware / Software Interface third edition, by: David A. Patterson and John L. Hennessy, ISBN: 1-55860-428-6 or 1-55860-604-1.

Class Time: MWF 1:00-1:50 PM

Final Exam: 3:00 to 6:00 on Monday April 21th

Classroom: GCB 143

## **The Instructor**

Instructor: Tim Stanley

My email: [tim@cs.byuh.edu](mailto:tim@cs.byuh.edu)

My Office: GCB 105

## **Office Hours**

My office hours for winter 2008 are MWF 10-12 and MTWT 2-3. Updated office hours (when necessary) are posted outside my office door. Students for whom the posted hours are not convenient can contact me by email to make an appointment.

I also have an open-door policy, posted on my office door as follows: ``If my door is open (even just a bit) feel free to knock and come in. -- Bro. Stanley''

Subject to Change

It is possible that I will revise some aspects of the course as we go along. Any changes I make are likely to be to your advantage. If any of my changes seems unfair to you, let me know. I will try to correct it.

## Teaching Approach

This class will be taught using a combination of lecture, assembly language projects using the MARS MIPS emulator from <http://courses.missouristate.edu/KenVollmar/MARS/download.htm>, and logic emulations using "Multimedia Logic" from <http://www.softronix.com/>, CEDAR Logic Emulator from <http://sourceforge.net/projects/cedarlogic>, and JLS at <http://www.cs.mtu.edu/~pop/jlsp/bin/JLS.html>, . Since we have has a lot of success with Multimedia logic it will be the default logic simulation package we use, but the others are certainly alternatives.

Shared drives will be created that you can use to develop and execute projects in these two packages in our classroom. The focus of this class will be design of computers in Multimedia Logic. The computer designs can be individual work or team projects with not more than five people on a team. This class is now three credit hours and so more will be expected. The final exam will be presentations of computer designs to faculty members and the Dean of the School of Computing.

## Why Take This Course?

Tools amplify our abilities. An airplane is a tool that lets us fly. A car is a tool that lets us travel quickly from place to place on the ground. A secretary or an assistant is a "tool" to which we can give instructions and get back results. What is a computer? Computers are not human. But they are probably the most human tool that man has ever made.

When is it safe to rely on results from a computer? What are its limits? When we ask it to do such-and-such, how long will it take? Will the answer be right? When the program "breaks," what might have caused the problem? If we really know cars and engines, we drive differently, with a trained ear and a sense of what is really happening. We can push the car to its limits and we know when to back off. With a computer, to really drive it well, it helps to have a similar sense for what the machine is really doing.

We will learn the simple, native language of computers. We will learn how computers think and some of the limits they have. We strive to truly understand computers, and we will make a very good start in this course. We will lay the foundation. We will gain an understanding of bits and bytes, of ands and ors and nots, of integers and floats and doubles, of addition, subtraction, multiplication, and division, of gates, latches, flip-flops, and memories. We will learn to think like a computer, and thereby realize the limitations on the thinking of all computers.

## Prerequisites

The math we will typically do is limited to adding and subtracting by one, and multiplying and dividing by two. But more than that, we will ``manipulate" numbers. We will take them apart and put them back together in different ways. We will see the world of mathematics as a computer sees it, which is probably just a little differently than you ever saw it before. We will also do some programming to see how small and simple things can be combined into complex and powerful results.

Knowledge of algebra and previous programming experience will be very helpful to you. Without them, it may take a bit more time for the classroom presentations and labs to sink in and make sense.

## Course Content

CS 410 will cover the following CC2001 knowledge units):.:

AR4. Memory system organization and architecture [core]

Minimum core coverage time: 5 hours

Topics:

- \_ Storage systems and their technology
- \_ Coding, data compression, and data integrity
- \_ Memory hierarchy
- \_ Main memory organization and operations
- \_ Latency, cycle time, bandwidth, and interleaving
- \_ Cache memories (address mapping, block size, replacement and store policy)
- \_ Virtual memory (page table, TLB)
- \_ Fault handling and reliability

#### AR5. Interfacing and communication [core]

Minimum core coverage time: 3 hours

Topics:

- \_ I/O fundamentals: handshaking, bussing, programmed I/O, interrupt-driven I/O
- \_ Interrupt structures: vectored and prioritized, interrupt acknowledgment
- \_ External storage, physical organization, and drives
- \_ Buses: bus protocols, arbitration, direct-memory access (DMA)
- \_ Introduction to networks
- \_ Multimedia support
- \_ RAID architectures

#### AR6. Functional organization [core]

Minimum core coverage time: 7 hours

Topics:

- \_ Implementation of simple datapaths
- \_ Control unit: hardwired realization vs. microprogrammed realization
- \_ Instruction pipelining
- \_ Introduction to instruction-level parallelism (ILP)

3. Explain basic instruction level parallelism using pipelining and the major hazards that may occur.

#### AR7. Multiprocessing and alternative architectures [core]

Minimum core coverage time: 3 hours

Topics:

- \_ Introduction to SIMD, MIMD, VLIW, EPIC
- \_ Systolic architecture
- \_ Interconnection networks (hypercube, shuffle-exchange, mesh, crossbar)
- \_ Shared memory systems
- \_ Cache coherence
- \_ Memory models and memory consistency

Learning objectives:

1. Discuss the concept of parallel processing beyond the classical von Neumann model.
2. Describe alternative architectures such as SIMD, MIMD, and VLIW.
3. Explain the concept of interconnection networks and characterize different approaches.
4. Discuss the special concerns that multiprocessing systems present with respect to memory management and describe how these are addressed.

## Course Schedule

Week	Chapters	Starting Date	Topics	Exams	Holidays, Homework, Projects
1	1,2	1/9	Review beginning chapters		CS210 Final as home work
2	3,4,5	1/14	Review continued		Review HW Due
3		1/21	Multimedia Logic review and design projects assigned		
4		1/28	Computer design project definition		Design definitions due
5	5	2/4	Pipelining		Instruction set design due
6		2/11	Assembler Design Examples		Chapter 6 homework due
7		2/18	Microcode design review		
8	6	2/25	Memory hierarchy and performance		
9		3/3	Hardware Design Review		Chapter 7 homework due
10	7	3/10	Interfacing Processors and Peripherals	Midterm	
11		3/17	Hardware Implementation Review		Chapter 8 homework due
12	9	3/24	Multiprocessor systems		
13		3/31	Demonstration program review		Chapter 9 homework due
14		4/7	Finalize Projects		
15		4/14	Presentation practice		
16		4/21	Monday April 24 Final 3:00-6:00		Presentations on designs to guests

## Grading

The allocation of points for grading has not been decided. This paragraph reflects some past practice, but it is under reconsideration. Take it with a grain of salt.

Grades will be computed on the basis of points earned as follows:

500 computer design project & Presentation

100 - Definition assignment

200 - Computer Design Implementation

200 – Presentation and Demonstration

200 midterm exam

200 Homework

100 Attendance and participation

1000 total

{Grading Scale:}

I use the following grading scale for this class.

930+ A

900-929 – A

870-899 - B+

830-869 – B

800-829 - B-

770-799 - C+

730-769 – C

700-729 - C-

670-699 - D+

630-669 – D

600-629 - D-

0-599 - F

Tests:

The primary purpose of tests (examinations) is to gauge student learning by measuring performance in a timed and supervised situation. Some memorization may be required.

It is understood that such a situation creates additional stress for many students. For this reason testing is not used exclusively in the grading process. Each test will receive a scaled (normalized) score and a letter grade indicating the final course grade that would be earned by consistent performance at the level reflected on that test.

Testing Center: Except the final exam, most tests are given in the BYUH testing center. The day of the exam we will preview it in class. Attendance is not required. Taking the test counts as attendance in class. However you may find that getting a preview copy of the exam, and being able to ask me questions about it, are both good reasons to attend. You can take the exam that same day or the next day. Generally I allow unlimited time but no books or notes.

Test Makeup: Exams cannot be made up except when I approve it in unusual circumstances. This is *very* rare.

### Discussing/Haggling:

I welcome the opportunity to discuss specific instances of grading with you, and to hear your requests for different grades than were initially assigned. In fact, I encourage it. Some very good learning occurs in these settings (for you and for me). About half the time I end up agreeing with the request. The best time to do this is during my office hours. As an alternative, you can submit your argument in writing, together with the original graded work.

### Final Grades:

Final grades are generally issued by email soon after the final exam, or in-person if I feel that some discussion might be beneficial. Students are invited to visit my office to claim any exams or homework that I am still holding, and to discuss their academic progress. Interim progress reports are issued to the students several times during the course, generally after the midterm exams and before the final exam.

### Lab Submission Rules

There are several rules that I use in this class. They are designed to allow you to learn, but to prevent you from doing things that might let you pass the class without learning. Violation of any of these rules is a violation of the honor code. You will receive a score of zero for any such assignment, and it cannot be made up. Repeated violations may lead to failing the class. Please be careful what you submit.

### The Keystroke Rule

Every keystroke in every lab you submit must come from your own fingertips. (If you are handicapped in some way that makes typing difficult or impossible for you, check with me. We can make a special exception for you if necessary.) You can re-use code that you wrote in a prior assignment (or in a prior class or in a prior job). You are forbidden to submit any code that was not typed by you yourself.

## The Open-Neighbor Rule

All labs are "open-neighbor" in the sense that you can confer with other students and lab assistants. You can read their code (if they let you). You can share your code with them. You can talk about your code, your approach, your difficulties, and your ideas. You can draw pictures and make analogies and ask the TA or me (even me) questions. You can use their ideas. However, you cannot submit their code even if you first modify it.

## The Looking Rule

Except for looking at the textbook, you are not allowed to look at your own code that you will be submitting and somebody else's code at the same time.

## The Challenge Rule

If I think that you may have violated these rules on some particular assignment, I will ask you (by email or in person) to state that you followed these rules. If I don't hear back from you, I will assume that you cheated.

## Communication by Email

When I want to say something, or when you want to say something, if we are not in the same room, my first choice is to do it by email. I far prefer it to telephone calls, for instance. When there is an announcement, I will generally tell you in class or send it to you by email. Such announcements might include clarifications on the homework assignments. You will need to maintain an email account and to provide me with a valid email address.

## Computer Accounts

You should have a computer account in the Computer Science lab (GCB~101). This account gives you access to UNIX systems, software (including compilers and assemblers), email, some storage, and some paper printing (currently 100 pages per CS class). There are also a few modems for dial-in access. You will use your CS account to do the lab work in this class. See me or a lab worker (GCB~101) to get set up.

## Subject to Change

It is very unlikely that I will make any major changes, but aside from course number, title, and description, I reserve the right to change anything in this syllabus including the grading policies and the course calendar. Important changes are generally communicated in class and by email to those affected. If my changes are unfair to you, let me know. I will try to fix it.

## Special Needs

Brigham Young University Hawaii is committed to providing a working and learning atmosphere, which reasonably accommodates qualified persons with disabilities. If you have any disability that may impair your ability to complete this course successfully, please contact the students with Special Need Coordinator, Leilani A'una at 293-3518. Reasonable academic accommodations are reviewed for all students who have qualified documented disabilities. If you need assistance or if you feel you have been unlawfully discriminated against on the basis of disability, you may seek resolution through established grievance policy and procedures. You should contact the Human Resource Services at 780-8875.

## Preventing Sexual Harassment

Title IX of the education amendments of 1972 prohibits sex discrimination against any participant in an educational program or activity that receives federal funds, including Federal loans and grants. Title IX also covers student-to-student sexual harassment.

If you encounter unlawful sexual harassment or gender-based discrimination, please contact the Human Resource Services at 780-8875 (24 hours).