

# CS64: Computer Organization and Design Logic

## Fall 2015

**Instructor:** Kyle Dewey ([kyledewey@cs.ucsb.edu](mailto:kyledewey@cs.ucsb.edu))

**Teaching Assistants:**

- Michael Christensen ([mchristensen@cs.ucsb.edu](mailto:mchristensen@cs.ucsb.edu))
- Benjamin Lampel ([denimalpaca@cs.ucsb.edu](mailto:denimalpaca@cs.ucsb.edu))

**Website:** <http://cs.ucsb.edu/~kyledewey/cs64>

**Lecture:** Tuesday / Thursday 11:00 AM - 12:15 PM in Phelps 3505

**Lab:** Friday 11:00 AM - 11:50 AM; 12:00 PM - 12:50 PM in Phelps 3525

**Textbook:**

- Computer Organization and Design: The Hardware/Software Interface, 5th edition (by David A. Patterson and John L. Hennessy)

**Course Description:**

(In my own words) An introduction to computer hardware, starting from the programmer's low-level interface and ending with basic circuitry.

(From the course catalog) Assembly language programming and advanced computer organization; Digital logic design topics including gates, combinational circuits, flip-flops, and the design and analysis of sequential circuits.

**Prerequisites:**

- CS16, with a grade of 'C' or better
- Math 3C or 4A with a grade of 'C' or better

**Grading:**

- 5% - Lab Attendance, 10 sessions @ 0.5% per session
- 30% - Lab Assignments, 10 assignments @ 3% apiece
- 15% - Exam #1, closed book, one hand-written page of notes (double sided), along with MIPS reference card
- 15% - Exam #2, closed book, one hand-written page of notes (double sided), along with MIPS reference card
- 35% - Final Exam - closed book, two hand-written pages of notes (each double sided), along with MIPS reference card

If your exam average is below 60% and is well below the class average, you will receive an 'F' in the class. This is to protect against students who lean on their partners too much, and whose tests demonstrate that they have not learned the material.

**Course Outline:** (subject to change)

Week	Day	Topic	Reading
0	9/24	Introduction: bases, binary, data representation, bitwise operations	Ch. 2.4, 2.6
1	9/29	Bitwise operations, addition, multiplication	Ch. 2.6, 3.2, 3.3
1	10/1	Assembly: MIPS introduction with <code>spim</code>	Ch. 2.2, 2.3, 2.6
2	10/6	Assembly: arithmetic, branches	Ch. 2.2, 2.3, 2.6
2	10/8	Assembly: MIPS memory	Ch. 2.8
3	10/13	Assembly: MIPS memory Data representation: floating-point, big/little endian	Ch. 2.8
3	10/15	Assembly: procedures, exam review	Ch. 2.8, A.6
4	10/20	<b>Exam 1</b>	
4	10/22	Assembly: calling convention rules	Ch. 2.8, A.6
5	10/27	Digital design (DigDesign) introduction	B.1 - B.3
5	10/29	DigDesign: boolean algebra	B.1 - B.3
6	11/3	DigDesign: combinatorial logic	B.3, B.5 - B.7
6	11/5	DigDesign: MUXs, ALUs, clocks, exam review	B.3, B.5 - B.7
7	11/10	<b>Exam 2</b>	
7	11/12	DigDesign: latches, flip-flops, memory	B.8, B.9
8	11/17	DigDesign: Drawing FSMs	B.10
8	11/19	DigDesign: FSMs	B.10
9	11/24	DigDesign: More FSMs	B.10
10	12/1	Building a working MIPS core	Ch. 4.3, 4.4
10	12/3	Final exam review	
11	12/9	<b>Final exam</b>	

**On Using Partners:**

The labs will often allow for the use of a partner. Under this setup, students are allowed to form **groups of two**, and may discuss assignment solution details within the group. Each member of the group is still accountable for his or her own work, though free communication is permitted within the group. This is more privileged than a group member communicating with a non-group-member (who cannot exchange assignment solution details), but less privileged than a team (who submit assignments as a single unit, as opposed to individuals). The intention with this setup is that if someone is stuck, they can ask their partner for help without needing to contact the TA or the instructor. Students may change groups between assignments, though not within an assignment, which avoids situations where information could be exchanged between more than two students on a single assignment. Forming groups is optional; students may work alone if they want.

**On Collaboration:**

In actual software development, very little work is done as an individual effort. Most modern software systems are simply too big to be undertaken in this style, so collaboration is key. This is one of the reasons why we allow partners for some of the labs - this represents the common case.

That said, collaboration is **not** simply taking the credit for someone else's work. When this happens, **everyone** loses. The obvious loser is whoever did all the work. The less immediate loser is the one who took half of the credit for the work, because that person did not learn whatever the assignment was trying to teach. This can (and I have seen it happen during my own undergrad!) snowball long-term. Eventually the student who took the credit without working is truly completely lost, often without realizing it, and is held up entirely by the work of other students. This is a worst-case scenario, but in my experience it is far too common.

The point is this: collaboration means to work together, not to blindly take someone else's work or to give your own work away.

**Attendance policy:**

Attendance will not be taken in lecture, and there will be no graded assignments given during lecture. However, any material covered in lecture is ultimately **your** responsibility, regardless of whether or not the lecture was attended. Attendance will be taken for lab, and being present at each lab counts towards a total of 5% of your grade.

**Due Dates / Late Policy:**

For items turned in late, each person has 24 hours worth of "grace" time in total. For example, if someone were to submit the first lab 4 hours late and the second lab 6 hours late, then a total of 10 "grace" hours have been used. Both submissions would be accepted without incident, and there would be 14 "grace" hours remaining. Except in extenuating circumstances, submissions for students who have gone beyond their grace time will **not** be accepted.

A little background on this policy - the grace time is intended to be used as a sort of last-minute “oops” relating to a poor time estimate of (what should be) final touches. This policy tries to reduce the number of submissions hastily done just to meet a deadline, and to prevent issues of submissions that missed the deadline by a relatively small amount of time. It is not intended to be used as a way to extend the deadline for one assignment for a day, although it certainly can be used that way. Be forewarned: once it’s gone it’s gone, so use it wisely!

### **Extenuating Circumstances:**

“Extenuating circumstances”, for the purpose of this class, is defined as anything beyond our immediate control. In these cases, at my discretion I can grant an extension. To be absolutely clear, there is **no** guarantee that I will do so, and I am not obligated to grant them. For the things we can predict (e.g., trips), I expect to be contacted at least a week in advance. For the things we cannot predict (e.g., illness), I need official documentation explaining the situation (e.g., a doctor’s note).

### **Communication Policy:**

I have two office hours per week, though I may increase this to 3-4 if questions abound. I’m also available by appointment.

With email, assume that I will take at least 24 hours to respond. Typically my response time is much, much faster than this, but I do occasionally take this long. Historically, this has only been an issue the last hours before a project deadline, and only for students who started far too late. The point being: start early!

### **Academic Honesty:**

In as few words as possible, cheating and plagiarism will **not** be tolerated. I understand that the temptation may be high (“it’s just this one assignment” or “I just need this class”), but this is **no** excuse. At the very least, this is unfair to all the students who did not resort to such unethical means, who instead took the time and struggled through. I will be following UCSB’s Academic Conduct policy on this (from [http://www.sa.ucsb.edu/Regulations/student\\_conduct.aspx](http://www.sa.ucsb.edu/Regulations/student_conduct.aspx), under “General Standards of Conduct”), quoted below for convenience:

It is expected that students attending the University of California understand and subscribe to the ideal of academic integrity, and are willing to bear individual responsibility for their work. Any work (written or otherwise) submitted to fulfill an academic requirement must represent a student’s original work. Any act of academic dishonesty, such as cheating or plagiarism, will subject a person to University disciplinary action. Cheating includes, but is not limited to, looking at another student’s examination, referring to unauthorized notes during an exam, providing answers, having another person take an exam for you, etc. Representing the words, ideas, or concepts of another person without appropriate attribution is plagiarism. Whenever another person’s written work is utilized, whether it be a single phrase or longer, quotation marks must be used and sources cited. Paraphrasing another’s work, i.e., borrowing the ideas or concepts and putting them into one’s “own” words, must also be acknowledged. Although a person’s state of mind and intention will be considered in determining the

University response to an act of academic dishonesty, this in no way lessens the responsibility of the student.

Note that collaboration with a non-lab partner also constitutes cheating. Any incident of cheating **will** be reported. While this may sound steep, real-life cases of cheating (i.e. taking someone else's code without permission) have led to job termination and lawsuits among other things, so this is not unrealistic.