Course Goals: As computer scientists, we tend to program using abstractions significantly removed from the hardware on which our code executes. Yet there are many cases where misunderstanding how the hardware works leads to poor performance, errors, security vulnerabilities, and other modes of failure. Software can't completely hide physics! A deeper understanding of the underlying mechanisms of computers also helps us distinguish real technological advances from marketing hype. In this course, we pull back the covers to see how machines operate at the lowest level, and examine the electronic, mechanical, and physical principles behind the operations. We will also explore enhancements to machine architecture beyond the basics.

Course Notes: Available at https://people.cs.umass.edu/~weems. I keep a blog there with an outline of what we cover in class, plus the slides, any homework assignments, and handouts. If you miss a class, it is important to check the blog for updates. I do my best to update it as soon as possible, but other commitments sometimes cause me to fall behind for a few days.

Grading: Midterm: 15%  Final: 15%  Exam Prep Homework: 10%  Logic labs: 16%  Assembly labs: 14%  Participation in class: 15%  Final Project: 15%

These percentages are estimates, and I reserve the right to make minor changes as necessary.

Exams are open book and open notes. This is so you can look up information such as machine instructions -- I don't expect you to memorize this sort of information for exams. I do expect you to have read the book, attended class, and to know where to look up the information you need. There won't be time in the exams to read material you're unfamiliar with. The purpose of the exams is for you to demonstrate that you have attained an operational level of understanding of the material. Bring a calculator to all exams unless I specifically tell you it won't be needed.

Homework and Term Project

Due Dates: Homework is generally due at the start of class one week after it is handed out, although holidays, etc., may cause exceptions. The project due dates will be specified in class and on the blog.

Late Policy: You must do your homework on time because we'll be correcting some of it in class. Likewise, projects may be preparation for in-class lab work, so we can make the best use of that time, which means that you can't do them late. If you know you have a
specific time conflict, make arrangements with me in advance for a separate assignment for late submission. Because answers are usually given out in class on the due date, credit cannot be given for regular homework turned in after that.

**Collaboration:** There are two kinds of homework: exam preparation exercises and labs. The purpose of the exam preparation homework is to give each person a chance to gauge their comprehension of material from class, and to practice answering questions like those on the exams. It is thus important to do this work alone. If someone asks you for help, try to take the role of a teacher and help them find the solution on their own; don’t just give them an answer. If I feel that people are submitting answers that are merely copies of each other, I will grade the one solution and divide the credit among the copies. If you do poorly on an exam prep assignment, and there is time before an exam, I may be able to give you a makeup. The objective is for you to be prepared for the exam.

Lab homework involves using tools to gain a deeper understanding of how a computer works at the level of the digital logic, and the native instruction set. These are more about having an experience, than getting the right answer (although the goal of the experience is for you to get something working). The same principles apply here, as to the exam preparation homework, in that you should only help others in a way that enables them to take the next step through the experience, rather than supplying answers. Of course, the TA and I are here to help too, and welcome questions.

The final project can be done individually or in teams of two. I will expect much more depth from a team. You should view working as a team as an opportunity to do more and go further, rather than as a way to share the work. I will offer some ideas for ways that teams can go deeper, but really want to see teams take initiative to explore the project in novel ways. If your team just completes the basic assignment, you will each get less credit than if you had done the work individually.

**Grading:** Homework may be corrected as an exercise in class, where people exchange homework and then mark each other’s work, or it may be corrected by the TA and/or me. Be considerate of your fellow classmates and us by writing legibly and organizing your answers in a clear manner. **Important advice:** I can be generous with partial credit only when you’ve made an attempt. If you leave a blank on a homework or exam, or don’t submit a lab, then it can’t be given any credit and is automatically a zero. **Always try!**

**Participation:** At various points during the semester, we’ll have in-class exercises. These may involve working through a practice sheet where you can ask for help, building things in the logic simulator or on the prototyping board, getting the MBED environment set up, trying out code, etc. When we do these, there will usually be an accompanying hand out to fill-in, and these will be submitted at the end of class so you can get credit for the work. If you miss a class, depending on the nature of the exercise, you may be able to make it up separately. Ask if this is possible, rather than assuming it is or isn’t.

**Project Teams:** If you choose to participate in a team, then your grade for the semester is partially dependent on your performance as a team. Some ways that you can help to ensure that your team does well:

Be realistic with your partner in discussing your abilities and time commitments when you divide up the project work. If possible, get together and work collaboratively, taking time to brainstorm ideas for how you will go beyond the requirements. Paired programming is a recognized software engineering technique in industry, and is known to yield higher
productivity, especially on small, intensive assignments. In many cases, you’ll learn more and learn it more quickly through working together directly.

Make sure you have each other’s schedules, phone numbers, e-mail addresses, etc. Establish a clear policy of when it is OK to contact one another. Let each other know when you are going to be unavailable with enough advance warning for good contingency planning.

Communicate! Communicate! Communicate! Tell each other what you are thinking. Don’t keep thoughts inside. Remember to praise each other for jobs well done. If you feel the need to criticize, use statements that start with "I think…." or "I feel…" Avoid criticism that starts with an accusatory "You did…," or "You always…"

Think of yourselves as a team. Develop some team spirit. Name your team. Develop a unique style for your presentations and demos (e.g., team colors, a logo). Get to know each other as individuals. Do lunch. Discuss hobbies, career goals. If you each take the time to care about one another, then you’ll do what it takes to excel as a team.

CmpSci 335 Assignments

Logic Labs: There are four digital logic labs, each of which has an associated handout that walks you through a process of creating a series of circuits in Logisim. The handout asks you to experiment with the circuits and answer questions at various stages in the construction process. They are briefly summarized here:

Logic lab 1: Use Logisim to simulate and explore simple combinational Boolean logic circuits: AND, OR, NOT, NAND, NOR, XOR, discover DeMorgan’s Law. (1 hour)

Logic Lab 2: Use Logisim to build a full adder, then a multibit adder/subtracter. (1 hour)

Logic Lab 3: Use Logisim to build an RS flip flop, a D flip flop, a register, and then a simple processor datapath with multiple functions. (2 hours)

Logic Lab 4: Extend the data path to include a program counter, memory, and branch logic. (2 hours)

Exam Preparation These are homework assignments that provide experience with questions like those that will appear on the midterm and final exams. (75 minutes each)

Programming Labs: Two projects that are designed to get you up to speed on mixed C and Assembly programming for the ARM processor using the MBED cloud-based development environment. The goal is to guide you in developing the skills necessary for the semester project.

Programming Lab 1: Use assembly to get the lights on the microcontroller to illuminate to show the low order 4 bits of a value passed in by a C program for a time that is passed as a second parameter (I/O port driver and assembly delay loop subroutine). (3 hours)

Programming Lab 2: Sort a table of ten numbers passed from C using assembly language, and display the outer loop counter using the lights and an appropriate delay. (4 hours)
**Semester Project:** Using the microcontroller on the prototyping board and additional components, develop a project of your choice that you think will impress the rest of the class. I have some components available for loan, and will take request to order some more. You may add components of your own if you wish. You will present your project to the department on the last day of class. Presentations will be evaluated in part by other students in the class on the basis of its impressiveness. You will submit a brief report describing the development process and what you learned from it. (8 to 10 hours)

**Exams**

Midterm (75 minutes) covers first half of class
Final Exam (two hours) covers second half of class since midterm (not cumulative)

**Bibliography**


ARMv7-M Architecture Reference Manual, ARM Inc., 2010

LPC17xx User Manual, NXP Semiconductors Inc., 2010

[www.cburch.com/logisim](http://www.cburch.com/logisim)

**Accommodation Statement**

The University of Massachusetts Amherst is committed to providing an equal educational opportunity for all students. If you have a documented physical, psychological, or learning disability on file with Disability Services (DS), you may be eligible for reasonable academic accommodations to help you succeed in this course. If you have a documented disability that requires an accommodation, please notify me within the first two weeks of the semester so that we may make appropriate arrangements.

**Academic Honesty Statement**

Since the integrity of the academic enterprise of any institution of higher education requires honesty in scholarship and research, academic honesty is required of all students at the University of Massachusetts Amherst. Academic dishonesty is prohibited in all programs of the University. Academic dishonesty includes but is not limited to: cheating, fabrication, plagiarism, and facilitating dishonesty. Appropriate sanctions may be imposed on any student who has committed an act of academic dishonesty. Instructors should take reasonable steps to address academic misconduct. Any person who has reason to believe that a student has committed academic dishonesty should bring such information to the attention of the appropriate course instructor as soon as possible. Instances of academic dishonesty not related to a specific course should be brought to the attention of the appropriate department Head or Chair. Since students are expected to be familiar with this policy and the commonly accepted standards of academic integrity, ignorance of such standards is not normally sufficient evidence of lack of intent ([http://www.umass.edu/dean_students/codeofconduct/acadhonesty/](http://www.umass.edu/dean_students/codeofconduct/acadhonesty/)).