Course Objectives

CS 233 has several goals:

1. Grasp basic principles of combinational and sequential logic design.
2. Have a high-level understanding of how to design a general-purpose computer, starting with simple logic gates.
3. To provide you with a mental model of how high-level language programs get executed on computer hardware.
4. To introduce you to the organization and performance analysis of modern computers.
5. Expose you to the hardware-level mechanisms for exposing parallelism.

The first two goals we will address together as we teach you the principles of logic design through the implementation of a processor capable of executing a subset of the MIPS instruction set architecture (ISA). Along the way we will discuss how computers represent numbers, understand the decode/execute model, the design of finite state machines and more. Specifically we will test the following objectives:

a. students should be able to design modest combinational circuits (20-30 gates) from a natural language (e.g., English) specification.
b. students should be able to design finite state machines of moderate complexity (10-20 states) from a natural language specification.

Furthermore, they should be able to implement these FSMs using a collection of gates and flip flops.
c. students should be able to analyze the design of a simple processor and modify a Verilog implementation to implement new instructions.

As computers execute programs in machine language, we will address the third goal through an extensive discussion of machine language and its human readable counterpart, assembly language. We will demonstrate how features of modern programming languages (e.g., function calls, recursion, pointers, dynamic memory allocation, etc.) are implemented in assembly language. In addition, topics like compilation, linking, I/O programming, and interrupt programming will be covered. We will specifically test the following objectives:

d. students should be able to translate small (20
line) C programs that include conditionals, loops, function calls, recursion, data structures, and pointers into MIPS assembly, observing calling conventions and stack management.

e. students should be able to translate short (30 line) MIPS programs to corresponding C programs including function prototypes.

The fourth goal will be addressed in three ways: 1) We will present an overview of the organization of modern computers (processor, memory, I/O system) demonstrating the key challenges and ideas (e.g., pipelining, caching, indirection, etc.) that influence their design, 2) we will present both theoretical and practical performance analysis techniques and analyze the performance of many parts of modern machines (processors, memory, caches, disks, networks), and 3) we will write code in high-level languages and assembly to optimize program execution. We will specifically test the following objectives:

f. students should be able to define and provide examples of the key implementation ideas of modern computers (abstraction, caching, hierarchy, indirection, pipelining, parallelism, speculation).

g. students should be able to analyze performance questions to identify what phenomena and/or numerical values are required and compute a result if these are provided.

h. students should be able to identify the hardware structures in a pipelined processor and cache/memory systems and explain their role.

While the previous goals focus primarily on single-processor systems, the last goal presents hardware mechanisms for implementing parallel processing. Specifically, we will discuss SIMD execution, cache coherence, memory consistency, and hardware synchronization primitives. As part of this discussion, we introduce concepts of race conditions, false sharing, and the need for memory barriers. We will specifically test the following objectives:

i. Provided a small piece of code executing on a shared memory parallel computer, students
should be able to recognize synchronization, coherence, and consistency pitfalls that could impact the execution's result or performance.

<table>
<thead>
<tr>
<th>Lectures</th>
<th>We have 50 minute lectures, three times a week. Bring a writing implement and an iClicker to lecture, because they will be interactive.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sections</td>
<td>First hour in 1111SC; second hour in 0218SC</td>
</tr>
<tr>
<td></td>
<td>Unless otherwise announced, there will be a quiz in each week's section, typically conducted near the end of the first hour:</td>
</tr>
<tr>
<td></td>
<td>• The quiz covers material the same material that was covered in the worksheet of that same section and one or more lectures preceding that section</td>
</tr>
<tr>
<td></td>
<td>• The quiz will differ by discussion section based on your TA, but should be similar in difficulty</td>
</tr>
<tr>
<td></td>
<td>• There will be some opportunities for extra credit based on the groups you are assigned to</td>
</tr>
<tr>
<td></td>
<td>• These quizzes are intended to be formative (i.e., help you understand the degree to which you understood the material) and together are only worth 2% of the course grade.</td>
</tr>
</tbody>
</table>
The first part of this course follows the first book *Logic and Computer Design Fundamentals*, but much of this material is available in an appendix of *Computer Organization & Design: The Hardware/Software Interface*. Some of the discussion sections in the first part of the course consist of Verilog problems, for which the *Verilog HDL* book can be used as a reference. However, the Verilog material covered in this course is very basic, and you should be able to find on-line any references needed. The second and third part of this course follows the book on *Computer Organization & Design: The Hardware/Software Interface* very closely, so it is highly recommended, but many students find they don't refer to the book. Any of the second, third, or fourth editions are fine.

Copies of the textbook (as well as others) are on reserve at Grainger Library.

We will use the Engineering Workstations (EWS) for the labs in this class. The web page that provides locations for of EWS labs also provides the names of machines if you wish to work remotely. Throughout the semester, we will be using iverilog, gtkwave, gcc/g++, and SPIM, all of which are installed on the Linux EWS machines, but many of which you can set up on your own machine. We can, however, provide little assistance in getting this software running on your machine.

https://wiki.cites.illinois.edu/wiki/display/cs233fa17 has the schedule and lecture and section notes. Just about all course materials will be distributed on the web.
We'll be using Piazza. Piazza is designed to reduce the time to getting your question answered and for easy navigation, but this in part relies on you appropriate usage. Please follow the following guidelines:

- Piazza is meant for general questions, such as asking for homework clarifications or addressing administrative issues. More specific or personal problems are usually better handled via email or in person.
- Use meaningful subject lines. "Segfault when accessing pixels in extractMessage" is much more informative than just "My head is falling off!" This makes it easier for other people to find posts that they are interested in.
- On a related note, **before you post a question, check see if somebody else has already asked the same thing.**
This class will use i>clickers. These are radio-frequency remotes, keyed to each individual student, which will allow anonymous (to other students, but not to the staff) responses to questions asked in Lecture. Past research has shown these to be very effective at improving student/lecturer interactions, which in turn are known to improve student learning (which is what we care about) and, consequently, student grades (what you care about).

You need to get one at the bookstore (new or used), and bring it to every lecture. 3% of your final grade will depend on lecture participation (participating means answering the question using your i>clicker, without considering whether your answer was correct or incorrect). You can miss up-to 3 lectures where i>clicker was used and still obtain full credit for lecture participation. You need to answer at least 50% of the day’s i>clicker questions to get attendance credit. In addition, extra credit of up-to 2% will be awarded for the correct answer. Notice this 2% will be extra credit. We will use the i>clicker from the first day of lectures, so bring it as soon as you have it.

You will need to register your i>clicker at the i>clicker web site. Your Student ID is your netID, which is your email address without the @illinois.edu. Your remote ID is in the back of your i>clicker. Even if you registered your i>clicker in a previous semester, you need to do it again this semester; otherwise we will not be able to assign you your i>clicker points for cs233. If you cannot read the remote ID on the back of your i>clicker, please go to the Customer Service Counter at the Illini Union Bookstore. A station has been set up there so that you can retrieve the Remote ID. Notice that you can use your i>clicker even if you have not registered, although you will need to register at some point to obtain the i>clicker points.

These are the components to your course grade:

1. **Labs (25%)**: There will be 15 labs throughout the semester, each weighted equally. See Lab Submission Policy below.
2. **Web homeworks (5%)**: we will have short web-based homeworks after some lectures that will be worth roughly 5% of your overall grade.
3. **Midterm exams (55%)**:
   1. We’ll have 7 midterms. Exam 1: Combinational design (10%, 5% for short answer and 5% for long form), Exam 2: FSM (5%), Exam 3: datapath modification (5%), Exam 4: MIPS fundamentals (10%, 5% for short answer and 5% for long form), Exam 5: MIPS advanced (5%), Exam 6: Pipelining/Performance (15%), and Exam 7: Caching (5%).
   2. Midterms 1, 2, 3, 4, 5, and 7 are computerized and autograded; Exam 6 is handwritten.
   3. For each midterm (except Exam 7), we’ll offer a “second chance” test to allow you to demonstrate mastery for the key concepts on the exam (details below). Second chance exams are only offered for the long-form questions and not the short answer questions.
   4. For Exam 6 only, you are allowed to bring one 8.5”x11” handwritten sheet of notes (both sides) to the exam.

4. **Final (10%)**: We’ll have a comprehensive final to cover the material that follows midterm 7 and a sampling of the material from the rest of the semester. Due to the timing, there will be no 2nd chance opportunity for this exam.

5. **Attendance (5%)**: Attendance will be taken in both lecture (using iClickers) and discussion sections (using quizzes). iClicker participation is 3% of this, and discussion quiz grades are the remaining 2%. You will have 3 free lecture absences and 2 free discussion absences. These free absences are provided to accommodate for cases where a student may miss class due to interviews, conference, illness, etc.
Extra credit:

1. As noted above, i>clicker performance (correct answers) will be worth up to 2% extra credit. For each day, we'll divide your number of correct answers by the maximum number of possible points that day to get your percentage for the day. We'll then average all the day percentages to get your overall performance percentage, which we'll multiply by 2% and award as extra credit.

2. Discussion sections are organized into groups sharing a table. If you get >= 8 points on a quiz, you get 1 point of extra credit for everyone else on your table who also got >= 8 points. This extra credit counts towards the 2% for discussion quiz grades.

3. Some labs will have small extra credit components. Details will be present in those lab handouts.

4. On very rare occasions there is opportunity for extra credit on exams.
**Lab Submission Policy**

**Submission:** code will be submitted through Subversion. Like any workplace would, we expect you to correctly check in your code. We will grade whatever was checked in most recently at the deadline. We won't be lenient if you forget to check in your code or check in the wrong version. You can verify which version of your code is checked in using the web interface for SVN.

**Partners:** For most labs, you can work individually or with 1 partner. The exceptions are labs declared as individual, and the SPIMbot tournament lab, where you **must** work in groups of two or three. You will specify your partners using the partners.txt file, by including the netids of all collaborators (your netid + any partners), one per line.

**Late Submission:** One lab's *second part* (other than the SPIMbot tournament) can be submitted up to 48 hours late without penalty. Otherwise, a 10% penalty every 12 hours (or part thereof) will be assessed, up to a maximum of 48 hours. See our Submission Policy page for more details.

**Style:** Good style is important whenever writing code; **we reserve the right to assign a 0 to any assignment that demonstrates total disregard for standard good style conventions.**
| Second Chance Testing | For each long-form (i.e., not short answer) midterm question, we offer an optional second chance exam that provides you the chance to improve your score on it. *Multiple choice questions are not offered on second chances.* Here’s how this works:

- You may show up for the second chance exam and read through the question(s) and complete as many or as few as you want (even none). For the computerized exams, you can choose whether you want to submit your work at the end; for Exam 6, you specify on the cover sheet which questions you want graded.
- For each question we grade, we will replace your original score for that question, even if the new score is lower. This means that it is important to not only know the material, but to know that you know the material (which is important in the real world).
- We cap the scores on each question on the second chance exam to 90% of the possible points for that question. Specifically, we grade the question normally and then assign you a score that is \(\min(\text{grade you earned on the second chance question}, 0.9 \times \text{maximum points for the question})\). As such, you should study for the original exam and use the second chance exam only as necessary.

- For the computerized exams, the first chances are purely autograded. The second chances are autograded, and submissions which don't get full credit on the autograder are hand-graded for partial credit. |

| Regrades | For computerized exams, regrade requests must be submitted within a week of the grade being released. For labs and computerized exams, the grade release announcements will have directions for asking grade-related questions and requested regrades. For Exam 6, they should be submitted in writing:

- please attach a sheet of paper detailing the problem(s) and section(s) that need to be regraded in order to be considered
- the regraded grade, if anything changed (whether positive or negative), will become your new grade for that question
- for written exams, once you have taken the exam out of the lab section (where we hand back the exams for you to look), or out of the office (Siebel 0212) if you were not present when it was initially handed back, it can no longer be considered for regrade |
| Grades | Assignment and exam grades will be available via the grade book. Please verify the accuracy of your grades often during the semester! Typos are unfortunately very possible in a class with many students. Letter grades will be assigned based on your overall numeric score. Also, the top and bottom 1% in each range will receive plus and minus grades.  
- 90% - 100% A  
- 80% - 89% B  
- 70% - 79% C |
| Cheating | Academic integrity is an important issue in general. The University expects you all to be familiar with Rule 33 in the Code of Policies and Regulations Applying to All Students. If we are able to pick out two nearly identical assignments out of 400, then cheating has likely occurred. **All parties involved will receive a 0 on that assignment or exam and their final course grade reduced by one letter grade** (e.g., A->B, B->C, etc.). A second offense will **result in a failing grade for the class**.  
Do NOT post your code into public code sharing tools such as Github. While you may not be cheating, you are enabling other students to cheat which is against the student code. |
| Accomodations | To obtain disability-related academic adjustments and/or auxiliary aids, students with disabilities must contact the course instructor and the Disability Resources and Educational Services (DRES) as soon as possible. To contact DRES, you may visit 1207 S. Oak St., Champaign, call 333-4603, e-mail disability@illinois.edu or go to the [DRES website](https://cbtf.engr.illinois.edu/#dres).  
For instructions on how to handle accomodations with the computer-based testing facility, please see [https://cbtf.engr.illinois.edu/#dres](https://cbtf.engr.illinois.edu/#dres) |
| Late Add FAQ | If you want to take the course but have not been able to register, please refer to the [Registration FAQ](https://cbtf.engr.illinois.edu/#dres) |