University of California, Los Angeles
CS M152A (ECE M116L) - Introductory Digital Design Laboratory

[Fall 2020] Course Syllabus

Instructor Information

- **Instructor:** Majid Sarrafzadeh
- **E-mail:** majid@cs.ucla.edu
- **Office Hours:** By appointment
- **Lab Location:** Online - Zoom (More instructions will be announced on CCLE)
- **Instructed Lab Hours:**

Section	Hours	Teaching Assistant
Lab 1	MW 10-11:50am	Rahul Garg
Lab 2	MW 12-1:50pm	Mohit Garg
Lab 3	MW 2-3:50pm	Rohan Surve
Lab 4	TR 10-11:50am	Mohammad Kachuee
Lab 5	TR 12-1:50pm	Migyeong Gwak
Lab 6	TR 2-3:50pm	Rajas Mhaskar

Course Description

In this lab class, you will get hands-on experience with design implementation on Field Programmable Gate-Arrays (FPGAs). You will apply what you've learned in CS51A or ECE M16 (combinatorial and sequential logic) and implement some designs using modern design tools and Hardware Description Language (HDL).

We have designed several projects for you to implement using the Xilinx ISE design and implementation environment. These projects are intended as tutorials for you to learn how to put designs together, implement them in Verilog HDL, and providing the design in simulation only.

Due to the major concerns regarding the coronavirus outbreak, unfortunately, you will

not be able to access the lab or work with the real hardware this quarter. You will work on the designs on your own computer using the free version of Xilinx ISE (Webpack), and sufficiently simulating the designs before submitting the project.

Important Notes:

- There will be no teamwork. Each project is graded individually.
- While the course syllabus and structure are similar between sections, the sections are separate in terms of grading and other class activities. Please only contact the instructor or the TA in charge of the section you are enrolled in.
- While the class is being offered online, the attendance is required and it will be part of the grading. If you think, for a justified reason, you would not be able to join the class regularly, please contact the instructor and TA (you need to have an approval by 10/9/2020)
- All deadlines are fixed and there will be no extensions.

Academic Integrity

The Office of the Dean of Students has summarized University policy on academic integrity. Here are the relevant links:

- Student Conduct Code: view
- *Academic Dishonesty*: All forms of academic misconduct or research misconduct, including, but not limited to, cheating, fabrication or falsification, plagiarism, multiple submissions or facilitating academic misconduct which occurs in academic exercises or submissions.
- *Cheating*: Cheating includes, but is not limited to, the use of unauthorized materials, information, or study aids in any academic exercise; the alteration of any answers on a graded document before submitting it for re-grading; or the failure to observe the expressed procedures or instructions of an academic exercise (e.g., examination instructions regarding alternate seating or conversation during an examination).
- *Plagiarism*: Plagiarism includes, but is not limited to, the use of another person's work (including words, ideas, designs, or data) without giving appropriate attribution or citation. This includes, but is not limited to, representing, with or without the intent to deceive, part or all of an entire work obtained by purchase or otherwise, as the Student's original work; the omission of or failure to acknowledge the true source of the work; or representing an altered but identifiable work of another person or the Student's own previous work as if it were the Student's original or new work. Unless otherwise specified by the

faculty member, all submissions, whether in draft or final form, to meet course requirements (including a paper, project, exam, computer program, oral presentation, or other work) must either be the Student's own work, or must clearly acknowledge the source.

These summaries don't specifically address programming assignments in detail, so we state our policy here. Of course, you understand that your work on programming assignments must be your own. But we understand that high-level discussions about approaches to a problem have educational value and are acceptable. So where do we draw the line? We'll decide each case on its merits, but here are some categorizations:

Acceptable:

- Clarifying what an assignment is requiring
- Discussing algorithms for solving a problem, perhaps accompanied by pictures, without writing any code
- Helping someone find a minor problem with their code, provided that offering such assistance doesn't require examining more than a few lines of code
- Turning in someone's work without crediting the author of that work, if the source of that work is a CS152A instructor or TA

Unacceptable:

- Turning in any portion of someone's work without crediting the author of that work, if the source of that work is not a CS152A instructor or TA
- Using project solutions from earlier offerings of this or any other class
- Writing for another student a code fragment that solves any portion of a project assignment
- Receiving from another person (other than a CS152A instructor or TA) a code fragment that solves any portion of a programming assignment
- Helping the same person find problems with their code more than a few times for a particular assignment
- You must abide both by this policy and the policies expressed in the UCLA Student Conduct Code (view). In accordance with University policy, we will submit cases of suspected cheating to the Dean.

Project Submission

For each project, the following should be submitted:

- **Project code**: the Xilinx ISE project folder should be cleaned up (Project > Cleanup Project Files), zipped, and uploaded in the corresponding assignment page on the course website.
    - A sample filename: *Firstname_Lastname_Project1_code.zip*
- **Video demos (about 10 minutes for each project)**: You need to prepare a video (screen recording) using the software that you have chosen for the project, and describe the following on your code in details:
    - Your thought process in designing the top modules
    - The role of each module, individually
    - How to integrate different modules
    - What were the challenges you faced? How did you overcome them?
    - What results do you see in your simulations, and what can we deduce based on them?
    - A sample filename: *Firstname_Lastname_Project1_videos.zip*

  You may use Zoom screen recording to make video files!
- **Lab Report**: the project report should be submitted as a PDF document following the below format.
    - A sample filename: *Firstname_Lastname_Project1_report.pdf*

Lab Reports

Project reports should contain the following sections. Percentages are subject to change and can vary based on projects' needs and specifications.

1. Introduction and requirement
    - Summarize background information about the lab and the detailed design requirements. It's very important to make sure you are designing the right thing before starting.
2. Design description
    - Document the design aspects including the basic description of the design, modular architecture, interactions among the modules, and interface of each major module. You should include schematics for the system architecture. You can also include figures for state machines and Verilog code when needed.
3. Simulation documentation
    - Document all the simulation efforts (what requirements are tested and what the test cases are), document bugs found during simulation, and

provide simulation waveforms. Note that the design summary report (output from Xilinx ISE showing device utilization after the implementation) should also be included.

4. Conclusion
    - Summary of the design. Difficulties you encountered, and how you dealt with them. General suggestions for improving the lab, if any.

Grading

You will be graded on a curve based on your performance in the course. The grading breakdown is as follows:

- Attendance Quizzes (10%)
- Project 1 (20%)
- Project 2 (20%)
- Project 3 (25%)
- Project 4 (25%)

Each lab/project is graded based on the following components:

- Lab Report (PDF document) (50%)
- Video Demo and Correctness (videos + project code in a Zip file) (50%)

Your final letter grade will be based on the following:

- If your total grade is above 90%, you will receive A- or above
- If your total grade is above 80%, you will receive B- or above
- If your total grade is above 70%, you will receive C+ or above
- For grades within each range or outside these ranges, we use the distribution of the grades within each session to make adjustments.