

CS M152A Introductory Digital Design Laboratory, Winter 2020

Administrative

Instructor: Miodrag Potkonjak (miodrag@cs.ucla.edu)

Instructor's Office Hours: By appointment

Lab / Office Hour Location: Boelter Hall 3424

Instructed Lab Hours:

Lab 4	TR	10:00 – 11:50 AM	Mohit Garg	gargmohit@g.ucla.edu
-------	----	------------------	------------	--

Open Lab/Office Hours:

Monday	8:00 – 9:50 AM	Rahul Garg
Monday	4:00 – 5:50 PM	Srishti Majumdar
Tuesday	4:00 – 5:50 PM	Mohit Garg
Wednesday	4:00 – 5:50 PM	Logan Kuo
Thursday	4:00 – 5:50 PM	Rohan Surve

The office hours of the other TAs will be updated soon.

Introduction

In this lab class you will get hands-on experience with hardware design implementation on Field Programmable Gate-Arrays (FPGAs). You will explore the applications of the fundamentals learned in CS51A (combinatorial and sequential logic) in practical digital design problems.

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, generate the programming file, as well as proving the design both in simulation and on real hardware.

The projects are not designed in such a way that you will spend a significant amount of time in the lab while keeping yourself awake with lots of caffeinated drinks. Instead you should treat the lab sessions as a time to work with the hardware and interact with the TAs. **You will find that working on the designs on your own computer using the free version of ISE (Webpack), and sufficiently simulating the designs**

ahead of time before testing them on the FPGA boards makes one's life much easier in this course.

Team Work

You will be working in teams of 2-3. The team is formed at the beginning of the quarter, and is not subject to any change throughout the quarter (unless there are any unforeseen exceptions).

Each team will be assigned a workbench with an FPGA board and a PC workstation. Keep in mind that your team will be sharing the same workbench with teams from the other sessions. Try to keep the workload balanced within the team.

Lab Conducts

1. Respect the lab equipment that you are assigned to and be considerate to the other teams that share the same workstation with you.
2. Keep your assigned workstation clean. Keep the lab free of trash.
3. Under no circumstances should you take the lab equipment outside of the lab.

Lab Access

The lab is only accessible to the students in a TA's presence. You are strongly encouraged to work outside of the lab to find all the bugs first before testing the design in a lab session.

In general, you should only attend your assigned instructed lab session. You may attend other instructed lab session. However, the students who are assigned to that session have priority over the hardware.

You are free to attend any open lab hours. Try to stick with the same workstation that you have always used, unless the workstation is already occupied.

Computer Use

The PC workstation are only for course work related purposes, and they have restricted internet access. Do not leave your design files on the computer. Do not clutter the screen with files. Use a USB thumb drive or a cloud-based file storage service to transfer files between your own computer and the workstation.

No grades will be given to a team if the design files are lost on the lab computers without a backup. Lab time is limited. Use your best judgment on what's the best use of your allotted lab time.

Academic Integrity

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

- [Student Guide to Academic Integrity](http://www.deanofstudents.ucla.edu/Student%20Guide%20to%20Academic%20Integrity.htm): <http://www.deanofstudents.ucla.edu/Student%20Guide%20to%20Academic%20Integrity.htm>
- [Cheating](http://www.deanofstudents.ucla.edu/Better%20of%20an%20Exam.pdf): <http://www.deanofstudents.ucla.edu/Better%20of%20an%20Exam.pdf>
- [Plagiarism](http://www.deanofstudents.ucla.edu/Begin%20that%20Paper.pdf): <http://www.deanofstudents.ucla.edu/Begin%20that%20Paper.pdf>
- [UCLA Student Discipline FAQ](http://www.deanofstudents.ucla.edu/faq2.htm): <http://www.deanofstudents.ucla.edu/faq2.htm>

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](http://www.deanofstudents.ucla.edu/SCC98.htm) (<http://www.deanofstudents.ucla.edu/SCC98.htm>). 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:

1. 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
2. Project Report (Electronic Version): the project report should be uploaded in the corresponding assignment page

Project Code

This submission will account for 5% of the total credit for each lab. The groups which have written the most efficient code for the particular design problem will receive maximum credit for this section. So make sure that your code is modular and easy to follow.

Project Reports

Project reports should contain the following sections. The percentage in the parentheses represents the percentage score for the section. (Total 40%)

1. Problem statement and design description (10%).
 - a. Document the design aspects including the basic description of the design, modular architecture, interactions among the modules, and interface of each major module.
 - b. You should include schematics for the system architecture. You can also include figures for state machines and Verilog code when needed.
2. Simulation documentation (10%).
 - a. Document all the simulation efforts (what requirements are tested and what the test cases are)

- b. document bugs found during simulation, and
 - c. provide simulation waveforms. Include explanation of all simulation testbench code source files.
- 4 . Contribution (10%).
 - a. In this section, list your individual contribution to the development and implementation of the algorithm. The purpose of this section is to ensure all team - members work equally towards the completion of each lab assignment.
- 5 . Conclusion (10%).
 - d. Summary of the design. Difficulties you encountered, and how you dealt with them.

Grading

- Verilog quiz (10%)
- Project 0 (0%) - No report, Demo only
- Project 1 (15%)
- Project 2 (20%)
- Project 3 (25%)
- Project 4 (30%)

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

- Project reports (40%). See the Lab Report sections.
Students within a team will generally get **different** scores on their reports. Team members can work collaboratively on certain sections of the report, but cannot turn in the same report.
- Project Code (5%). See the Project Code section. Members within the same team will get **equal** credit.
- Demo (55%). The following are the rules for demo points. Note that before the deadline, only complete and correct demos are accepted when all the team members are present. If unfortunately, you can't finish the project on time, you have to demo whatever accomplished by the end of the next session. Members within the same team will get **equal** credit.
 - Demo on time: 55%*
 - One-session late demo: 40% x (correctness), where correctness refers to the percent of the demo you are able to finish by the late session.*