UNIVERSITY OF BRITISH COLUMBIA
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
CPEN 311 - Digital Systems Design

Laboratory Information

A significant part of this course is a hands-on lab. The lab will give you a chance to try out the concepts discussed in class, and actually build and test working hardware. Even if you normally shy away from lab work, I suspect that you will find these labs interesting and worthwhile.

There will be five lab assignments, **each** worth 7% of your final grade for a total of **35% of your final grade**.

You must submit your code to the course's Canvas site before the deadline for each lab. We will be using these submissions to look for unauthorized collaboration in the assignments; suspected cases will be investigated further and reported to the Dean's office for further consideration.

Labs 1, 2, and 3 will be done individually. Labs 4 and 5 can be done in pairs ONLY by those students who have an average of at least 80% in Labs 1 and 2. Students with an average less than 80% in Labs 1 and 2 will do labs 4 and 5 individually. This should be a good incentive to put in effort early in the course.

Lab Submission File Structure

The lab submission needs to be a ZIP file (not a RAR file or any other compressed file format). It needs to have the filename

"FirstName_LastName_StudentNumber_Lab_LabNumber.zip", for example:

john_doe_123456_Lab_2.zip

For labs 4 and 5, if submitted in pairs, the filename should be:

"FirstName1_LastName1_StudentNumber1_FirstName2_LastName2_StudentNumber2_Lab_LabNumber.zip"

For example:

john_doe_123456_james_smith_567890_Lab_2.zip

When the ZIP file is unzipped, the root directory needs to have a project report in a README.DOC or README.PDF file which contains the following:

1. Where (which directory path) is your SOF file located, and what it is called
2. What is the status of the lab (what works, what doesn't)

3. Annotated simulation screenshots as required by the lab
4. Information on how to run the simulations (i.e. where the files are located, which program you used to run your simulation)
4. Any additional information that would be relevant for the TA marking your project.

Inside the ZIP file, the subdirectory structure should be:
rtl/ : contains all the source code and compiled FPGA code (including all project subdirectories generated by Quartus, including the SOF file)
doc/ : any additional documentation
sim/ : files needed to run the simulation, if not present in the "rtl/" subdirectory

Policies on Missed Labs

If you know ahead of time that you will miss a lab for a medical reason or another approved reason you must contact the instructor at ylinn@ece.ubc.ca *before the lab*. If you fall sick on the day of the lab, contact the lab TA immediately, so alternative arrangements can be made. **Do not show up to a lab section other than your own** without receiving permission.

Marking scheme for Labs

The laboratory are graded in the following manner:

Not all students will demo the labs to the TAs. The TAs will call upon students during lab hours to demo according to their discretion. In general a random subset of students will be asked to demo their work, but TAs may also ask any student to demo if for example they have some questions about their code or for example if they believe the code is not original.

If a demo occurs, the breakdown of the grading is (unless the lab instructions say otherwise):
40%: Functionality - the circuit does what we wanted it to do
10%: Simulations: Show and explain simulation of main module(s) to TA
10%: Efficiency - the code does not use more FPGA resources than is necessary
10%: Reliability - the code is written reliably
10%: Modularity - the code is written in a modular fashion
10%: Readability - the code is written legibly and neatly, with lots of comments, and it is clear which hardware structures are instantiated
10%: oral presentation in class (the whole group must be present and participate!)

100%

If no demo occurs, the breakdown of the grading is (unless the lab instructions say otherwise):
40%: Functionality - the circuit does what we wanted it to do
20%: Simulations: Show and explain simulation of main module(s) to TA
10%: Efficiency - the code does not use more FPGA resources than is necessary
10%: Reliability - the code is written reliably
10%: Modularity - the code is written in a modular fashion

10%: Readability - the code is written legibly and neatly, with lots of comments, and it is clear which hardware structures are instantiated

---

100%

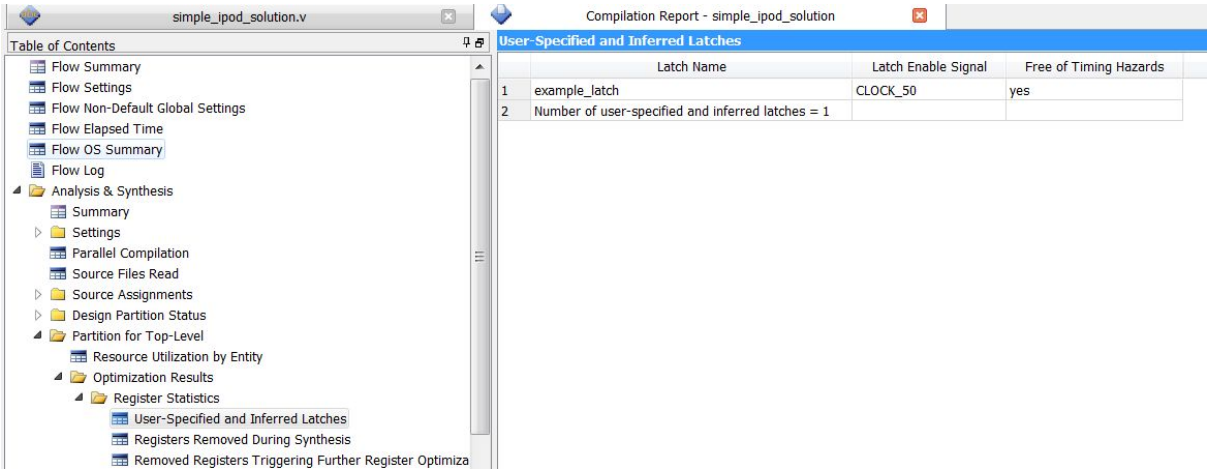
Some labs contain extra bonus tasks that may give you extra points on the lab.

Obviously, although functionality is 40% of the grade, if the lab does not attain a minimum of functionality, the grade for the laboratory will be 0. If a student is asked to demo and the demo shows that the student is completely unfamiliar with the code, or other extremely serious issues (e.g. absolutely nothing works and that contradicts the lab report) then the grade will also be 0.

Although you need to work independently on each laboratory, you can use any set of modules that other people wrote for **previous** labs. This is part of the emphasis on modular and reusable design. The only thing is that you must, if you use a previous lab module, write a short comment in the module code that says that (for example, in Verilog: "/ / This module was written by X in laboratory Y")

Latch Penalty

A 20 point penalty will be enforced for the labs 2 to 5 if your code contains latches. TAs will specifically look for this. If your code contains latches this means that you don't know what you are doing - there is never in this course an instance where creating latches on purpose is a good idea. SystemVerilog provides safeguards that we discussed in class to avoid creating latches, if your code contains latches this means that something is seriously wrong. Look at the following screenshot in order to see how you can make sure that your code does not contain latches:



Make sure that no latches are in your code as shown in the example screenshot under "User Specified and Inferred Latches". This penalty was added because in previous semesters some students had a lot of difficulty with the labs because they did not maintain in the forefront of their mind the connection between the code they are writing and the structures that are implemented, and had latches all over the place which completely messed up the implementation of their lab. If you do have latches in your code, and you

don't know how to fix that, then this is a serious issue that usually points to a bigger problem of not understanding what it is you are implementing, and you should understand why you have latches and how to remove them before continuing, otherwise you will likely just be wasting your time getting the lab to work. So, before you demo to the TA, look at the compilation report and make absolutely sure that there are no latches in your code, in order to avoid this 20 point penalty.

Plagiarism Detection Service

Please familiarize yourself with UBC's policy on academic honesty at the following two websites:

http://www.students.ubc.ca/calendar/index.cfm?tree=3,54,111,959

http://www.students.ubc.ca/calendar/index.cfm?tree=3,54,111,960

All labs will be submitted to an on-line plagiarism detection service. Suspected plagiarism will be reported to the Dean's office.