ECE 4750 Computer Architecture, Fall 2023

Lab Assignment Logistics

School of Electrical and Computer Engineering
Cornell University

revision: 2023-08-21-07-08

This document describes what students are expected to submit for the laboratory assignments and how their submissions will be evaluated at a high level. A handout is provided for each laboratory assignment describing the motivation for the assignment and providing background on the baseline design, alternative design, testing strategy, and evaluation. Each laboratory assignment requires you to submit three parts: the lab milestone (i.e., one or more initial implementations along with the corresponding tests), the lab code (i.e., all implementations along with all tests), and the lab report (i.e., a document with several sections, possibly including an introduction, testing strategy discussion, alternative design discussion, optimization discussion, and conclusion). The lab code is worth 25% of your final grade (22.5% for ECE 5740), the lab reports 10% of your final grade (7.5% for ECE 5740), and the lab milestones are ungraded. In total, the labs are worth 35% of your final grade (30% for ECE 5740).

1. Lab Code Quality

Code quality is one of the criteria used to assess the labs. You should follow the coding conventions as illustrated in the tutorial on Verilog. Please remove unnecessary comments that are provided by instructors in the code distributed to students. Do not commit executable binaries or any other unnecessary files.

2. Lab Report Submission

In addition to the actual code, we also require a lab report. The report offers an opportunity to convey the high-level implementation approach, motivation for specific design decisions, and quantitative and qualitative evaluations of different designs.

The lab report should be written assuming the reader is familiar with the lecture material and has read the lab handout. You may need to paraphrase some of the content in the handout in your own words to demonstrate understanding. Details about the actual code should be in the code comments, not the report. The report should focus on the high-level implementation and evaluation aspects of the assignment. All reports should include a title and the name(s) and NetID(s) of the student(s) who worked on the assignment at the top of the first page. Do not put this information on a separate title page. The report should be written using a serif font (e.g., Times, Palatino), be single-spaced, use margins in the range of 0.5–1 in, use a 10 pt font size. All figures must be legible. Avoid scanning hand-written figures, and do not use a digital camera to capture a hand-written figure. Do not just use a screen capture of the code. Instead, cut and paste the code into your report as text and format it appropriately. Clearly mark each section with a *numbered* section header.

Your report should not look like an outline. It should look like a report with paragraphs and prose. **Although there is no page limit, most high-quality reports require around four pages.** You should interleave your figures, plots, and tables in the main body text where appropriate and not place all of your figures, plots, and tables at the end.

Each report should include the following sections:

- **Introduction (one paragraph maximum) –** The introduction should be brief but still provide a good summary of the lab. Summarize the purpose of the lab. Why are we doing this assignment? How does it connect to the lecture material? Feel free to paraphrase from the handout as necessary. Include a sentence or two that describes at a very high level the baseline and alternative designs.
- **Alternative Design –** Describe the alternative design and its implementation. Include a datapath diagram or a block diagram, and possibly an FSM diagram for the alternative design. Consider a paragraph that provides an overview of the design before doing a deep dive into the details of one or two interesting aspects of the design. Think critically about what are the key items to mention for the reader to understand how the alternative design works. Examples are usually great to include here to illustrate how the alternative design works. We encourage you to describe how their design incorporates specific design patterns and principles discussed in lecture and the discussion section, but be specific. Simply saying the design exhibits modularity, hierarchy, encapsulation, regularity, and/or extensibility is not sufficient; be specific and explain *how* the design exhibits these design principles. Do not include detailed information about Verilog signals or code; your lab report should be at a higher level. If you include waveforms or line traces, then you must annotate them so that the reader can understand what they mean. This section will likely be about one page. **Remember that you must provide a balanced discussion between what you implemented *and* why you chose that implementation.**
- **Testing Strategy –** Describe the testing framework provided for testing your design. Describe the overall testing strategy. Be specific and explain *why* you used a specific testing strategy. Explain, at a high level, the kind of directed test cases implemented and why those were chosen. Consider including a table with a test case summary or some kind of quantitative summary of the number of test cases that are passing. We recommend starting this section with a short paragraph that provides an overview of your *strategy* for testing (so how all of the testing fits together). Then you might have one paragraph for each kind of testing. Each paragraph starts with the "why" (why that kind of testing) and then goes on to the "what" (what did you actually test using that kind of testing). Then you can end with a paragraph that pulls it all together and tries to make a compelling case for why you believe your design is functionally correct. Do not include the actual test code itself; your lab report should be at a higher level. If you include waveforms line traces, then you must annotate them so that the reader can understand what they mean (i.e., what corner case does the waveform or line trace illustrate?). For lab 4, you will also need to discuss your *software* testing strategy in addition to your *hardware* testing strategy. **Remember to provide a balanced discussion between how you tested your design *and* why you chose that testing strategy and test cases.**
- **Optimizations (optional) –** For lab 4, you may include a dedicated section to discuss any software or hardware optimizations they experimented with. We suggest dedicating a paragraph to each optimization. Start by motivating what overhead the optimization focuses on, then discuss how the optimization can mitigate this overhead, and finally discuss the details of how the optimization was implemented in your implementations. Discuss optimizations even if the optimization did not actually improve performance. Do not quantitatively evaluate your optimizations in this section. This section may just be a single paragraph or could be several paragraphs. **This section is optional. Students can still earn full credit without this section.**

- **Evaluation –** Report simulation results, using an appropriate mix of text, tables, and plots. Do not simply include the raw data. A plot is almost always helpful. You must include some kind of analysis of the results: Why is one design better or worse than another? Can you predict how the results might change for other designs or parameters? What can we learn from these results? Include some kind of qualitative analysis of the impact of the alternative design on cycle time (i.e., clock frequency), area, and energy. Simply saying one design uses more area or energy is not sufficient; be specific and explain *why* one design might use more area or energy. We recommend starting with the performance analysis and then having three (possibly short) paragraphs: one each on area, energy, and cycle time. This is where your qualitative analysis comes in on these important metrics. This section will probably be one of the longer (and most important) sections. **Remember to provide a balanced discussion between what the results are *and* what those results mean.**
- **Conclusion (1 paragraph maximum) –** All reports end with a conclusion. Include a brief qualitative *and* a quantitative overview of the evaluation results (Which implementation performed best? By how much? On which inputs?). Include also some high-level conclusions they can draw from their qualitative and quantitative evaluation. Do not over-generalize. Can you predict how the results might change for other inputs? What can we learn from these results? Which design should we use in the future? If it depends, explain why it depends.
- **Work Distribution (1 paragraph maximum) –** Provide a one-paragraph description of which student did what work. It is perfectly fine if one student ends up doing more work for that lab; the key is transparency and honesty about the work distribution across teammates.

It is also always great to include extra material to help demonstrate your understanding. For example, you could include line traces and reference them in the alternative design to illustrate a key feature of your design, or reference them in the testing strategy section to illustrate a subtle bug or a kind of testing, or reference them in your evaluation to illustrate *why* a specific input pattern performs the way it does. If you include line traces, you must annotate them. Label the columns and maybe even draw on them to show what is going You could include a particularly clever test case and reference it in the testing strategy section. You could include a pen-and-paper example to illustrate how your baseline design or alternative design works. Also, be sure to highlight "extra" work you did in your design, testing, or evaluation. If you tried two different alternative designs, discuss them in the alternative design section and make sure to use them to create a richer comparative analysis in the evaluation section. If you used a new kind of testing technique (e.g., randomly generating different mixed instruction sequences), then make sure you highlight that in the testing strategy. If you added an interesting new evaluation input, make sure you highlight that in your evaluation section. There are many creative things you can do to set your report apart!

Many students initially struggle with the idea of preparing the lab report. In previous courses, students may have simply described their code at a low level in a lab report. In this course, we challenge students to prepare reports that better demonstrate their understanding of the course content. Before starting to write, we encourage you to prepare a detailed outline. The outline should include one section for each section that will eventually make up the report. Under each section, there should be one bullet for each paragraph the student is planning to include in that section. This bullet should describe the topic of the paragraph. Under each bullet, there should be several sub-bullets, one for each topic to be discussed in that paragraph. The outline should also explicitly include references to the figures, tables, and plots the student plans to include in the report. This is called a *structured approach* to technical writing.

3. Lab Grading Scheme

The lab milestone is ungraded and is purely a way for students to receive early feedback from the course instructors. The lab code is assessed using four criteria weighted as follows:

- Lab Code: Baseline Functionality 30%
- Lab Code: Alternative Functionality 30%
- Lab Code: Verification Quality 30%
- Lab Code: Code Quality 10%

As discussed in the syllabus, each criteria/subcriteria is scored on a scale from 0 (nothing) to 4.25 (exceptional work). The functionality of the designs is assessed based on the number of test cases that pass in both the student and instructor test suites in combination with the severity of any errors. The verification quality is assessed for labs 1–3 based on the judgment of the instructor in terms of how well the test cases actually test the design. The optimization quality is assessed for lab 4 based on holistic quantitative analysis of the performance of the software and hardware on the sorting benchmark. The code quality is based on: how well the code follows the course coding guidelines; inclusion of comments that clearly document the structure, interfaces, and implementation of all modules; following the naming convention and build system structure appropriately; decomposing complicated monolithic expressions into smaller sub-expressions to increase readability; cleanly separating combinational and sequential logic; using local parameters for constants; organizing the code logically to match the dataflow in the design. Overall, good code quality means little work is necessary to figure out how the code works and how we might improve or maintain the design.

The lab report is weighted as follows:

- Lab Report: Introduction 10%
- Lab Report: Alternative Design 25%
- Lab Report: Testing Strategy or Optimizations 25%
- Lab Report: Evaluation 25%
- Lab Report: Conclusion 10%
- Lab Report: Writing Quality 5%

Again, each criteria/subcriteria is scored on a scale from 0 (nothing) to 4.25 (exceptional work). A detailed rubric will be provided with the PA final report grade to explain how each section was assessed.