CSE 141L – Winter 2022

Final Submission Instructions

Submission Options

Option 1 (Very Strongly Preferred)

Sign up with a TA to give a live demo that describes and demonstrates your design. Your demo should cover all the elements described at the end of this document. As a backup, we ask that you still submit the project files along with a README to help us navigate through your work. As incentive, if you choose this option, **you do not need to write or submit a formal report.**

Option 2 (Preferred):

Record a video describing and demonstrating your design. Submit the project files along with a README to help us navigate through your work. Tell us clearly what worked and what did not. Please carefully follow the detailed instructions at the end of this document. As incentive, if you choose this option, **you do not need to write or submit a formal report.**

Option 3 (No demo, requires detailed formal report):

Submit the project files along with a README to help us navigate through your work. Tell us clearly what worked and what did not. You must also submit a formal report describing your design. The report must cover everything described in the video content section at the end of this document, but in the format of a formal, written report. Persons choosing this option must include clear "diffs" of the content from their Milestone 3 report.

Rationale:

- Recording a video is hopefully faster and easier than further revising the formal report; M1–M3 have exercised that skill sufficiently.
- When you demonstrate effectively that your programs work in video, we can be mostly confident that they work as you show, and we may not necessarily need to re-run your code.
- In cases where we do run your code, with the video describing your design it is much easier and clearer to see how to get your assembler and processor running correctly.
- If you opt for the formal report, the only way we can verify your programs work is by running your code. If any part of your report is unclear and leads us to not be able to set up your project properly, it is very possible that we will not be able to run your processor and thus cannot award you points for it.
    - In cases where this does occur, we will need to go through a regrade process in a very tight time window during a very stressful part of the quarter. We will expect you to be highly responsive to our inquiries.

Regarding Program 2

Some folks identified that the spec for Program 2 could be interpreted to mean putting the original 11-bit message in memory, or putting the full 16-bit with corrected parities if needed in memory. As this was brought up well into the implementation period, we will accept either interpretation. **Be sure to make very clear which you chose.**

Grading

Your course grade is based on the behavior of your final processor, specifically how well your design executes the three assigned programs:

A Demonstrate clearly that all three programs run correctly with a variety of input operands (i.e. different initial memory conditions).

A– Clear demonstration that any two programs work, but the third doesn't.

B Clear demonstration that one program works, but the others do not.

C Hardware design compiles cleanly in Questa/ModelSim and Quartus (no errors; warnings okay); reasonable assembly and machine code written; programs give wrong answers or x's on some inputs, but a reasonable level of effort demonstrated. Include an explanation of at least one known issue or path forward you would try next.

Positive Modifiers

We wish to reward the groups that go above-and-beyond, so there are a few ways to earn extra points for outstanding designs and reports.

NOTE: To be eligible for ***any*** performance awards, you must have perfect correctness on ***all three*** programs. Persons with correctness errors are still eligible for other awards.

Best of P1: Top-performing processor on Program 1.	+half letter
Best of P2: Top-performing processor on Program 2.	+half letter
Best of P3: Top-performing processor on Program 3.	+half letter
Best overall processor: Top-performing across all programs.	+half letter

Outstanding Architectural Innovation: Staff discretion for something particularly interesting, creative, or unique. Few, if any, given.	+half letter

Negative Modifiers

Each missing / very incomplete section of the final report	-half letter
Each completely missing milestone	-half letter
More than one late milestone and/or late final submission	-half letter

Additional Modifiers

The course staff reserves the right to add additional positive or negative modifiers given specific situations that may arise. These additional modifiers may be applied to a whole group or to individuals within a group (e.g. to resolve imbalance in contributions) as appropriate.

Video Submission Guidelines

Note: Since there could be technical issues for video recording and the link to the video takes time to be generated, please be sure to start early on recording the video (as there are less TA available during the weekend and there would be no TAs after midnight to help). Video preparation issues *are not* an acceptable excuse for late submissions. We suggest planning to submit by Tuesday evening at the latest, to give yourself time for any unexpected issues.

Video Content:

Submit a short (2–4 minute) video that gives an overview of your design.

- Go over your architectural highlights
  - E.g. Machine type, number of registers, instruction format(s) and bit breakdowns, branching logic, supported operations, etc
- Show us a block diagram or datapath for your processor
  - Can be hand-drawn or generated by Quartus
  - Highlight any places you had to make interesting modifications for your architecture, and elaborate on any custom hardware modules you might have included
- Explain how your testbench works
  - In particular, what does your testbench do to prove that it runs each program correctly?
- Show us the results of executing your testbench (output files; maybe some waveforms)
  - Prove to us that your processor executes the programs correctly.
- Evaluate the performance of your processor
  - “Convince an architect that your processor works well for these programs.”

Video Recording How-To:

Videos should be recorded on zoom and include the link in your README file (see below).
To record:

- One person of the team should log in to Zoom **using UCSD's SSO (this step is important for the recording to be uploaded to the cloud)**, and start a zoom meeting
- Add all your teammates (please show all your faces for at least the beginning part)
- Share your screen and start recording. **Make sure the "Record to the Cloud" option is selected when you start recording.**
- Double check that the session is being recorded
- Do your presentation
- Stop recording/end the meeting.

You will then be prompted by Zoom that "You will get an Email notification when the cloud recording is ready”. After 20-40 minutes, an email will be sent to you with a link and a passcode to the video (At the bottom of the email, where it says "Share recordings with viewers: ucsd.zoom.us/xxxx Passcode: xxxx). You should include that link and passcode in the README file.

What to Submit (Video Option):

Submit all of your source code on Gradescope including:

1. All hardware (Verilog) modules
2. Your testbench
3. Assembly code for your 3 programs
4. Machine code for your 3 programs
5. Your assembler (python, or other high-level language file)
6. A screenshot or printout of your testbench output
7. Additional files needed to run your processor (e.g. memory initialization files)

In addition, include a text document named **README.txt** in which you:

1. Explain which programs/features work
2. Explain which programs/features don't work and what challenges you faced when implementing your design.
3. **Include the link and passcode to your zoom video (see above)**

What to Submit (Written Report Option):

Everything described above in the Video Option, **plus:**

1. A formal, written report that includes everything described in the "Video Content" section above.