Final Project Submission for Fall 2015

ECE/CS 3710 - Computer Design Lab
Due Dates:
Archive your project (Unix tar.gz or .ZIP archive) and send to me by:
Saturday, December 19, 2015; no later than 5pm
Detailed instructions given below

I have already discussed the final report submission with each team, individually, regarding the content of the report and how it should be organized. Because all the projects have different applications and interfaces, I gave specific pointers to you on how to best describe your efforts in a concise manner.

This document describes what you should submit (and how), and some general guidelines for preparing your report. For the submission, send me an archive of your project. Email may or may not work if your archive is too big. Preferably, you should upload your archive on a website (e.g. our own CADE server), dropbox or Google-drive, so that I can download it. As soon as I download it, I'll send you an email so you can take it down. Please look for my confirmation email that I have received it. If I am unable to access your archive, then I'll send you an email too.

I. What to Submit

The entire project can be submitted as an electronic archive. You may use either Unix (.tar.gz format) or Windows/ZIP archives, and send me the link. If you have a web-page, you can upload the archive on the web-page and send me the URL in email. Once I download and untar everything, I'll send you an email so that you can take it down.

All of the following should be in the package:

1) The final project report. PDF preferred, but DOC is okay.
2) If you have your slides from the mid-semester presentation, please include them too.
3) An electronic copy of the final poster that you prepared for the demo.
4) All the Verilog source code, starting from your top-level CPU module, down to each bit that you finally compiled. Include the code for all the peripherals. Also include the final, overall pin-constraints UCF file.
5) Test benches: All of you wrote a lot of test benches to test your code. [If you have deleted any or all testbenches, well, then there is nothing to submit. But I don't think you would delete them!] Do not worry if they look too dirty; test benches usually look dirty. I would like to see what you tested for, and how you wrote your testbenches. If you just kept on over-writing new test benches over old ones, that is fine, do not worry.
6) Your software assembly program, pseudo-code and/or machine-code. You designed your own assembler, so include that too.
7) A simple README file that briefly describes your archive, and anything else you want me to know.

In your archive, do not put everything in one directory. Preferably, there should be a separate directory for each of the following: Verilog Code, Test benches, Software code/assembler, Documents. The final project report, mid-semester presentation slides, and the README file should be in the Documents directory.

II. Report Organization

Below are some guidelines to organize your final report. These are just guidelines, you can use your own organization. But ideally, these things should all be covered.

1) Have a separate Title page with authorship.
2) Write a 100-150 (approximately) word abstract. It should be an abstract (or a summary) of the whole project, briefly outlining WHAT you've done. The "HOW's" of the implementation will be in the manuscript, not in the abstract.
3) Introduction: This should be like the Problem Statement, or the Objective. Introduce your fully integrated system here. Describe its overall organization. The hardware/software part.
4) Then you should go "inside" your system and start describing the basic hardware (i.e. CPU + organization) and your extensions, along with the reasons for incorporating the extensions.
5) The CPU was then integrated with peripherals - Memory, IO, (S)NES, CODEC, SVGA, Keybd, etc. Concentrate on the interface and how the CPU control accesses/communicates with these.
6) You should use, with the help of a figure, an example to show some interesting cases of how your CPU + peripherals work:
   - For example, how does a load operation work on your CPU. Say, in cycle 1, PC is applied on the memory bus, then the data is stored in a register (or maybe you directly send it to the ALU/Regfile bus, ..., and so on.
   - If your CPU was more basic, but you had funky VGA buffers, describe how that works.
   - If you are proud of your SNES interface, show a block diagram of the interfacing signals and describe (briefly) how data is fetched.
   - If you used the NIC/Bluetooth/UARTs, make sure to clearly describe the CPU's hardware interface to it and state-machine/protocol for the sender and receiver.
   - Some of you used the MIDI input, how did you figure it out? How does it work? Others used wireless, one group used the head and motor of a VCR, and so on — how is that integrated with the CPU.
7) Software - You should describe the software as a separate section. Make sure to describe the software organization - how is the data represented? Code/Data Memory organization? Flow-charts or Pseudo-code for algorithms, etc. Some of you prototyped the entire application in software first, and then translated to your machine-code. Describe your approach.
8) Overall System integration and how does the system work.
9) Lessons learned, mistakes made and subsequently rectified, surprises (if any) on simulations and synthesis, what went on expected lines, etc. - all such issues can be described in one separate section. This is one important section in your report. This exercise will remind you of what you really learned in this class.
10) Have a separate section that discusses each group member's responsibilities and accomplishments. Be honest.
11) Conclusions & Future work: Conclude your report with 1-2 paragraphs and then discuss how could you improve your project and what future extensions do you envision.
12) References: While working on this project, each team came-up with their own ideas and may have referenced some external literature. This is where you list your references. For example, where did you get the SNES controller's operation manual from, where did you get the manual of the network controller, block-ram, etc. If you referred to a conference or journal paper, where was it published? If you found something on the Internet, refer to the URL.

The report can be written in a top-down (Application → Software → Architecture → CPU → . . .) or a bottom-up fashion - this is your choice. Everyone has his/her own writing style and I do not wish to impose one on you. Just be well-organized and consistent.

Please make good use of figures: block-level schematics, state machine diagrams, timing diagrams (say, if you wish to describe a resetting sequence, or a communication sequence for SNES, etc.), flow-charts to describe your software, etc.

How many pages should your report be? Somewhere around 15-17 pages, single-column, single spacing, give or take a few. Use whatever format you like, IEEE, or ACM, or your own. Keep 1-inch page margins.

Finally, please remember to return your lab kits. Also remember to fill out the confidential team-work assessment form (on the class website) and give it to me.

I hope you had fun in this course. If you have any questions or concerns, feel free to send me an email.