Computer Sciences Department
University of Wisconsin-Madison
CS/ECE 552 – Introduction to Computer Architecture
Project Description

1. Summary

The CS/ECE 552 term project is the complete functional design of a microprocessor called the WISC-SP22. All components of your design will be written in Verilog. As with the course homework assignments, the CS/ECE 552 Verilog restrictions apply, and all final code is expected to pass the Vcheck program.

The project will be completed individually.

The specifics of the microarchitecture and WISC-SP22 architecture are found in separate documents, and will also be posted on the course website.

The project will progress in several distinct stages. Some of these stages are enforced through grading deadlines; others are not. The deadlines are:

Date	**Project Component**
2/21/22	Design Review (5% of project grade)
3/6/22	Phase #1 (15% of project grade)
4/3/22	Phase #2 (30% of project grade)
N/A	Phase #2.1
N/A	Phase #2.2
4/8/22	Form Project Group
4/24/22	Phase #2.3 Caching (10% of project grade)
5/3/22	Phase #3 (30% of project grade)

Each stage of the design makes the processor progressively more complicated. For your own benefit, it is strongly recommended that you not proceed to a new stage before you are confident the current stage is working to specification. Debugging errors in a complex design is much harder. It is almost always better to test smaller, simpler components first.

Many of the Verilog problems in the homework assignments were designed to be compatible with the project. Please feel free to reuse these modules (of course, fixing any errors first!).

In addition to the previous homework problems, you will be provided with several reusable modules that you can use in your design. Most of these are Verilog implementations of memory system components. Please note that these files do *not* follow the CS/ECE 552 Verilog restrictions, so don't include them when you run Vcheck. Download the project tar file (`/u/s/i/sinclair/public/html/courses/cs552/spring2022/handouts/verilog_`

`code/project/project.tgz`) to get the complete collection of the files you will need for all stages.

An assembler for the WISC-SP22 ISA is provided for your use. Sample test programs are also provided, although you are strongly encouraged to write custom tests to augment these. Be aware that these test programs were written for a slightly different ISA specification and therefore may not work as advertised. It will be your job as diligent designers to determine if unexpected behavior occurs due to a bug in your design or as the result of the change in ISA. See the course website for a description of how to use the assembler.

2. IMPORTANT NOTES

- Start early: This project is designed to take a considerable amount of time.
- Plan ahead: You may find that the instructor, TAs, and peer mentors will be very inaccessible the night before a deadline.
- Ask questions: If you are getting stuck on some problem ask for help. Ask me, the TAs, the peer mentors, or your classmates.
- Functionality: Getting a working design is of paramount importance. Optimizations, clock-speed and bonus questions come 2nd. First make sure your design works!

If you finish really early, you will get the opportunity to earn extra credit by adding extra features to your processor, synthesizing your design. Additional details are available in the Extra Credit document.

If you are able to complete this project without the unnecessary stress that procrastination imposes, it is our belief that you will find this to be a highly rewarding experience.

All of the files you will need are included in the project tar file.

Finally, a reminder of some important documents:

1. Follow the instructions on ModelSim Setup Tutorial to get your environment setup.
2. Read the Command-line Verilog Simulation Tutorial. Additional references are on course website.
3. Read the Verilog Cheat sheet and Verilog rules pages. Everything you need to know about Verilog are in these documents.
4. Read the Verilog file naming conventions and Verilog file naming convention checking webpages and adhere to those conventions. We will be checking this in your submissions to ensure you followed the rules.
5. Read the Verilog rules checking page on the course website and adhere to the conventions. This page also provides information on how you can check that your files conform to these rules.
6. Read the Handin Verification page on the course website. You will need to run this before submitting your answers.