CSCI 2500 — Computer Organization
Group Project (document version 1.0) — Due December 9, 2022
How do computers actually compute?

- This project is due by the midnight EST on the above date via a Submitty gradeable.
- This project is a *group assignment*. You can have groups of up to five people.
- Start the project early. You can ask questions during office hours, in the Submitty forum, and during your lab session.

1 Project Overview

For the group project, we'll be taking Lab 9 and Lab 10 to the logical conclusion – you'll be exploring a Verilog description of the datapath for a fully functional 32-bit MIPS ISA that we reviewed in class. While you will be allowed to use Verilog code from the textbook, your job will be to analyze execution of a short sample MIPS program.

Note that this project is meant to be exploratory. Your team will have a fair amount of freedom in making certain choices but it also means that there is a number of decisions to make. Make sure you document your entire thought process and justifications for any decisions that you will be making.

2 Verilog models

You may use any Verilog code that we reviewed in class or that appears anywhere in our textbook. Make sure you label each Verilog model that you use, so that we know whether it is something that your team created or it is coming from the textbook. You have to show all Verilog code that makes up your model, even if this code is from the textbook.

3 Datapath

You can implement either non-pipelined or pipelined datapath.

4 Memory

You need to choose whether you will be implementing memory hierarchy (i.e., use cache) or single-level memory for both your instruction and data memory.

5 Simulation

First, you need to come up with a sample short (3-4 instructions) MIPS program that you will be simulating. Note that the program has to comprise specific instructions that are stored in instruction memory as 32-bit words. You will need to show the contents of the corresponding addresses of the instruction memory. Your sample program has to provide a mix of at least two different instruction types (i.e., instructions cannot be all arithmetic and logic instructions, or jump instructions, etc.)

Second, you will describe how your Verilog processor model executes the sample program. You are welcome to simulate the execution manually or use some tools to help you run your simulation.

If you decide to use some tools, please feel free to explore what is available out there. There is a number of IDEs, online simulators, etc. that you can use. In your writeup, you will need to specify which tool or tools you are using. To help you get started with designing a testbench, you might want to check out the corresponding lab in zyBooks ("11.11 For instructors: creating new Verilog zyLabs" https://learn.zybooks.com/zybook/RPICSCI2500KuzminFall2022/chapter/11/section/11), although you won't be able to create an actual zyLab for your project on zyBooks. You can also explore other resources available on the web (like https://hardwarebee.com/ultimate-guide-verilog-test-bench/ but there are many others, of course.

Please note that your simulation has to be detailed enough to show the state of inputs and outputs of each key component of the system during each clock cycle. Feel free to use any means of representing the state that you find appropriate, e.g., tables, timing diagrams, etc. Your goal here is to demonstrate your understanding of the process of executing memory stored instructions by your Verilog model of the processor.

6 References

Your project writeup must include the References section that lists all resources that you used when working on this team project. You don't have to list every single resource that you accessed (e.g., if you opened a web page just to see that it is not what you want) but every resource that you actually used (read, found useful, etc.) must be listed. Resources include not just our zyBook but also any other books, articles, blogs, forum posts, YouTube videos, tutorials, etc. For each reference you need to provide as much information as possible (title, author, publisher, publication date, URL, page numbers or chapters, if it is a longer paper or a book, etc.)

7 Submission and Grading Criteria

7.1 Submission Requirements

- You MUST type up your writeup. Handwritten solutions will not be accepted or graded, even if they are scanned into a PDF file.

- We recommend using LaTeX (https://www.latex-project.org/). If you have never used LaTeX, you might want follow some tutorial, like this one: https://www.latex-tutorial.com/tutorials/. There is a convenient online LaTeX editor called Overleaf (https://www.overleaf.com/) that has a free plan for students.
- Submit your writeup on Submitty as a single PDF file named `project.pdf`.
- You may submit additional files (e.g., your Verilog model and testbench files, your IDE project), if you feel that they would be helpful in grading your project. All additional files need to be well organized into directories and packed into a single archived ZIP file.

There will be two different checkpoints and deadlines that you'll need to be aware of. The final submission will consist of the project writeup.

- December 2 - Finalize your groups on Submitty
- December 9 - Final submissions to Submitty for project writeup.

Since final submission is due on the last day of the semester, it will not be possible to use any late days for the team project.

7.2 Grading Breakdown

The grading breakdown is as follows:

Project writeup

- Fully explains all choices made
- Includes all Verilog code necessary to model a MIPS processor with memory
- Gives a sample short MIPS program
- Describes the process of executing this sample short MIPS program
- Explains contributions of each group member
- Provides a References section that lists all resources that were used while working on this project