Art of Engineering: CE / CS Dept. Project
Simulation of a Very Simple Microprocessor Core

Instructor
Caleb Tulloss
caleb.tulloss@columbia.edu

Introduction

In this project, we will explore some of the main tools and techniques of computer engineering to implement a simulation of a very simple microprocessor, using Logisim. Computer engineering is an exciting field at the intersection of hardware and software design, and in this project, you will get to experience a little of each. We will also employ two classic design principles used in computer engineering: abstraction and design hierarchy. By the completion of the project, you will get a taste of...

- the way computer engineers *analyze* systems.
- the types of *design* problems faced by computer engineers.
- the style of *design tools* used by computer engineers for both hardware and software design.

More specifically, you will be able to...

- recognize both *how* and *why* computer engineers use *abstraction* and *design hierarchy* in the design of hardware and software systems.

Finally, you will have a sense of whether you would like to continue learning about computer engineering. It is meant to be fun!

Prerequisites and Context

The goal of this project is not to teach you how to design a microprocessor. If you continue with computer engineering, you will take at least one whole class on this! We do not expect that you will understand the whole design; we simply want to give you a sense of what it would be like to study computer engineering or computer science.

No background in programming is necessary, and we will be programming very close to the hardware, in machine language -- i.e. directly using 0's and 1's. Students who have completed the project have remarked that it is closer to CE than CS, but even if you think you want to stick to software, knowing how the underlying hardware works, and how the software is executed on that hardware, will make you a better programmer.

Assignments and Structure

There are four main sections of the project, corresponding to processor components you will design and simulate. The section topics, and their corresponding components, are:

1. **Logic**: Arithmetic/Logic Unit
2. **Memory**: Register File
3. **Programming**: Your Program
4. **Datapath**: Complete Microprocessor

For each section, there are three things to complete:

1. An asynchronous module covering the content.
2. A synchronous session to begin designing the component within the context of individual and/or group activities.
3. A design report showing you have completed the component, and verified its functionality.

You are expected to complete the content-based asynchronous module **before** the corresponding synchronous session. Content will NOT be covered during the synchronous session; they are NOT lectures. The report showing your final implementation of each component will be due one week after the corresponding synchronous session.

Timeline (subject to change)

Fri Feb 17: Deadline to sign up
Synchronous Location: Fayerweather 313 1:30pm-3:00pm

Date	Asynchronous Module Due	Synchronous Session	Report Due
Fri Feb 17	Logic	Logic	
Fri Feb 24			ALU
Fri Mar 3	Memory	Memory	
Fri Mar 10			Register File
Fri Mar 24	Programming	Programming	
Fri Mar 31			Program
Fri Apr 7	Datapath	Datapath	
Fri Apr 21			Final

Assessment

There are many intermediate deadlines, each of which is for a small section of the final project. The purpose of all these intermediate deliverables is to ensure that you are making steady progress throughout the semester. Each of these sub-project deliverables will be graded on a 1/0 basis (after all, this is a digital design project…).

We will assess your final report work based on evidence of clear effort and exploration of the important concepts. Specifically, we will be looking for whether you have made use of abstraction and design hierarchy to simplify your design process, and whether you have run appropriate simulations to characterize your design. We would rather see you submit an “incomplete” project that is carefully designed and analyzed than a complete microprocessor implemented without knowing what is going on!

Detailed Grading Breakdown

Required: ALU Report turned in	+1
Required: Register File Report turned in	+1
Required: Program Report turned in	+1
Required: Final Report turned in	+1
Required: Attendance at 2 synchronous sessions	+1
Attendance at each additional session	+0.5 each
Final report: follows all directions	+1
Final report: processor contains all sub-blocks	+1
Final report: effort made to debug wiring issues	+1
Final report: demonstration of working program	+1
Total	**+10**
Bonus: Extra functionality and/or very cool program	+1

Score	**Dept Project Grade**
11	**A+**
7-10	**A**
5-6.5	**C**
<5	**No Credit**

To get any credit for completing the department project (grade of C or higher), you must sign up by the deadline (Feb 17), turn in all 4 required reports, and attend at least 2 synchronous sessions in person.

Instructor Support

Please do not hesitate to ask me questions **on the Ed Discussion (on courseworks)**; you can do so anonymously if you choose to, and your classmates will also benefit from the answers. Asking questions on the shared discussion board, where everyone can see the question and the answer, means that I don't have to spend time answering the same question several times via email, which leads to faster response time. **If you email me a question about the project, I will respond by telling you to ask your question on Ed Discussion.**

I will have an office hour on Monday from 12pm - 1pm eastern (NYC) time. The zoom link is here.

Textbook

You are NOT required to use a textbook for this project. However, if you are looking for a useful reference, check out *Computer Organization and Design, RISC-V Edition*, by Patterson and Hennessy. Not to be confused with the other computer architecture book, by Hennessy and Patterson. You do NOT need to buy this book; I assure you there are other ways of finding it.