CSE 141L: Building a Microprocessor

Steven Swanson
Mark Gahagan
Raymond Paseman
Thinh Nguyen
You will design and implement a microprocessor this quarter!
Course Goals

• Apply what you’ll learn in 141
  • Design principles
  • Performance measurement
• Extend you’ll learn in 141
  • Understand deeply how a processor works
  • See architecture play itself out in a real design
• Learn (more) Verilog
• Get experience working on a large-scale project
• Have Fun!

   I hear, I forget.
   I see, I remember.
   I do, I understand.
Course Content

• In 10 weeks you will implement a pipelined MIPS processor in Verilog
  • It will be able to run simple (but “real”) programs compiled with gcc.
  • It will be able to do simple IO.
• Over the course of the quarter the design will become more and more your own.
  • We will give you code for some pieces
  • You’ll give you the design of some others
  • We’ll give you specifications for some others
  • You’ll invent, design, and implement some of your own
Course Format

• Six labs
  • More about these in a moment.
• Lectures
  • Verilog coding and course coding standards
  • Discuss current or upcoming lab
  • Work through part of the lab
  • Answer questions about the lab
  • Sort of like group office hours
Warning!!!

- This course is **ALOT** of work.
  - Don’t let the 2 units fool you.
- Do not fall behind
  - The labs build on each other.
  - If you fall behind, it’s very hard to catch up.
- Do not wait till the last minute
  - It’s called *hardware* for a good reason
  - The tools are complicated.
  - The tools are (a bit) buggy.
  - Your code will be buggy (unless you allow enough time for testing)

You **CAN** complete this project. Most students do. But you **must** manage your time and **start early**.
Lab 1: Learning the Tools

- Two tutorials
  - Building projects in Quartus
  - Entering and compiling Verilog
  - Simulation in Model Sim
  - Measuring the characteristics of your design
- Build, simulate, and synthesize simple circuits
- Measure their properties
- These are skills you will need throughout the class
- Start now!
- Duration: 1 week
Lab 2: Assemble the Pieces

- Implement the datapath for a subset of MIPS
  - No branches.
  - Some other simplifications.
- We’ll give you the design and some key components
- You’ll implement the design
  - Implement a library of useful building blocks
  - Test them
  - Use them to implement the design
- Duration: 1 Week
Lab 3: Signs of Life

• Add control logic to the datapath from Lab 2
• Test your simple processor
• Execute simple programs
• Duration: 1 Week
Lab 4: It Lives!

• Add the missing pieces of MIPS
  • Branches
  • Complex memory operations
  • Some other bits
  • Test thoroughly.

• You now have a working processor!
  • Compile some simple programs for it!
  • Run them!
  • Measure performance!
  • Be disappointed by how slow it is :-(

• Duration: 2 Weeks
Lab 5: Teaching it to Run

- Pipeline your processor
  - Watch the clock rate sore
  - Understand some of the deep, dark secrets of the MIPS ISA

- Measure performance again
  - See how clock rate, CPI, and the compiler interact to determine system performance
  - Dream of ways to improve performance.

- Duration: 2 weeks
Lab 6: Making it Fly

- The sky is the limit
  - Deeper pipelines
  - Build a multiprocessor
  - Branch prediction
  - Speculation
  - Multi-media instructions
  - ???

- Fastest processor wins.
  - There will be fabulous prizes.

- Duration: 2-3 Weeks
Link to 141

- You do not need to be in 141 to take 141L
  - You need to have already taken (or be taking) 141.
- You should follow along in 141
  - Information about MIPS
  - The single-cycle and pipelined MIPS design
- 141 will be useful for inspiration in Lab 6.
Doing the work

- Lab 1 will be done independently
- Lab 2-6 will be in groups of 2-3
  - Choose your groups carefully
    - Splitting up is allowed.
    - Merging groups is not.
- The overarching philosophy is “learn by doing”
  - You (and your group) must do all your own coding and design.
  - You should absolutely talk to other students in the class about Quartus/ModelSim problems, design options, etc.
  - Labs 1-3 are specifically for this: you are all building the same thing. Learn from each other!
Lab Space and Software

- We will use the Altera tools for development (Quartus II)
  - Verilog entry
  - Design analysis
- We’ll use ModelSim for simulation
  - Simulation
  - Debugging
- Tools are a pain
  - Like all hardware design tools, there are bugs.
  - These are among the best tools available, hard as that may be to believe.
  - These are also “Industrial Strength.” They are tools you might use in a future job. This means they are complex and powerful.
- The labs in the CSE basement have the tools installed
- They are also available for free (see link on the website)
  - *We are using Quartus II version 12.0sp2*
  - *We are using ModelSim-Altera Starter 10.0sp2*
  - *Follow the directions in the lab for getting these versions (they are not quite the latest)*
  - *No other versions are supported in this class.*
  - *Also, Windows only (The Linux GUI is a mess)*
Grading

• Two grading schemes
• By the numbers
  • Six labs
  • Equal weight for each
• Outcome-based
  • Do a reasonable job on the labs (at least a C)
  • Deliver a working pipelined processor (Complete Lab 5).
  • You get an A.
  • Deliver a working processor with some cool additional features (Complete Lab 6)
  • You get an A+.
Course Staff

• Prof: Steven Swanson
• TAs
  • Mark Gahagan
  • Raymond Paseman
  • Thinh Nguyen
• We will be in the lab a lot
  • We hate to be lonely!
  • We will fix the room and time shortly.
• See the course website for details
  • http://cseweb.ucsd.edu/classes/sp13/cse141L-a/
Lab Hours

• Mornings?
• After noon?
• After dinner?
• Weekends?