
Section 1

— CSE 369 Workflow —

Meet your TAs!

Aadithya Manoj (he/him)

- 3rd year Computer Engineering student
- Took CSE369 in Au24
- This is my first time TAing!
- Musician in my free time



Meet your TAs!

Isabel Froelich (she/her)

- 4th year Computer Engineering student
- Took CSE 369 in Au24
- 5th time TAing, first-time TAing this class!
- I ride horses :)



Meet each other!

In groups of 3-5, introduce yourselves:

- Name
- Year
- Favorite class so far
- Why you are taking this class?
- How would you say DE1-SoC?

Administrivia

- **Pre-course survey:** Due today (1/9, [link](#)).
- **Lab kits:** If you didn't pick up a kit, read this [Ed post](#). We will chain some lab kits to desks in CSE 003 to use as "shared stations". We also have access to a remote FPGA development platform called LabsLand, which lets you synthesize Verilog designs and upload them to remote FPGAs that have webcams pointed at them.
- **Lab 1&2:** Report due next Wednesday (1/21) @ 2:30 pm, demo by last OH on Friday (1/23), but expected during your assigned slot.
 - Lab demo slots will be assigned on Monday (1/19).

What to expect from section

- **When:** Every Friday at 1:30 pm in Allen 203
 - Recordings will be made available afterward on Panopto
 - Different TAs each week!
- **What:** SystemVerilog is a tricky language and we don't have much time to talk about it during lectures (only 80 min/week), so we are introducing optional sections focused on it.
- Please come and share your feedback! Let us know if there is anything you would like to focus on!

369 Workflow

More on this in Lecture 2

Quartus



- The Intel Quartus Software is a tool for designing, synthesizing, and programming FPGAs.
- With it, you are able to (1) **write SystemVerilog code**, (2) **compile it**, and (3) **program your DE1-SoC board**.
- More information: [Quartus Tutorial](#)

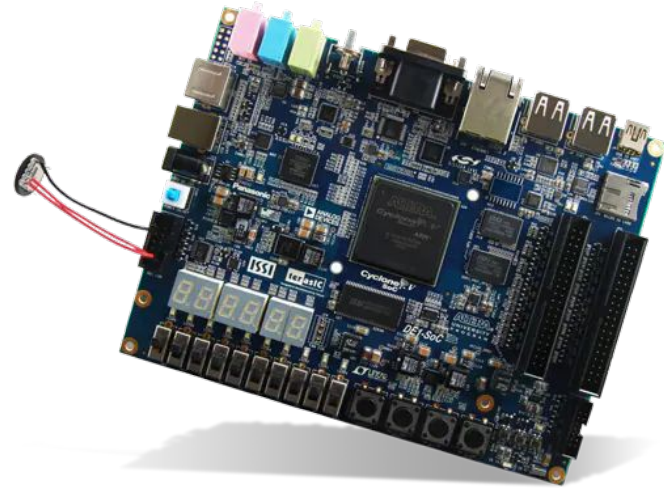
ModelSim ModelSim

- ModelSim is a simulation and debugging tool for a variety of hardware description languages, including SystemVerilog.
- With it, you are able to run simulations to **verify your code's logical behavior** without endangering your hardware.
 - This is your main tool for **debugging!**
- More information: [ModelSim Usage Guide](#)

Some details on how FPGAs work will be covered Lecture 9

DE1-SoC Overview

- The DE1-SoC is a development kit built around an FPGA.
 - An FPGA is a large array of logical elements with reprogrammable connections. This means we can use it to build many different kinds of hardware all on the same chip!
 - The kit also contains other functionalities that we will use in our labs.
- More information: [FPGA overview](#)



Installation Process

Installing Quartus and ModelSim

- If you wish to skip installation altogether, you can use the lab computers in CSE 003.
- Follow the installation tutorial on the website: [link](#)
 - This class only supports the **Windows** version of Quartus and ModelSim.
 - For **MacOS**, you can use a Windows virtual machine (described in the tutorial).
- Installation tips:
 - All 3 downloaded files need to be in the same directory so that ModelSim and Cyclone V options are available during Quartus installation.
 - Try not to miss the USB Blaster II driver installation option at the end.


Workflow Demo

Shortened version of the [Quartus Tutorial](#)

File Organization

- Unzip Lab1_files_Q17.zip to get started.
 - Download from the Lab 1&2 specs.
 - Can be unzipped again to start any new project or you can copy an existing project directory.
- Create subdirectories for each lab within a class directory (e.g., CSE369/lab1).
 - All project SystemVerilog files should be placed in this directory and added to Quartus project.
- Every Verilog module should have a test bench in a *separate* file.
 - Suggested naming scheme: new_module.sv and new_module_tb.sv for test bench.


Programming Workflow (Quartus)

- Open the Quartus project via the **.qpf** file in the project directory (double-click or find using File → Open Project...).
- In the Project Navigator “Files” tab, need to right-click and select “**Set as Top-Level Entity**” on the proper file/module.
 - The top-level entity should NOT be a test bench.
- When done coding a module, save the file and then run the **Analysis and Synthesis tool**: 
 - Quartus’ interface for compilation warnings and errors is better than ModelSim’s.
 - Use this tool to fix errors with syntax and signal connections before simulation.

Simulation Workflow (ModelSim)

- Double-click `Launch_ModelSim.bat` in the project directory.
- In a text editor, `modify runlab.do` for your project:
 - Add files to compile (modules + test benches).
 - Change which test bench you wish to simulate.
 - Change the waveform script file (`*_wave.do`) – this won't exist at first.
- Execute `do runlab.do` in the *Transcript* pane.
 - Use waveforms to verify/debug logical behavior of your module(s).
- Update waveform script file as desired.
 - Click on different modules in the *sim* pane to access different signals.
 - Drag signals from the *Objects* pane into the *Wave* pane.
 - With the *Wave* pane selected, `Ctrl+S` to overwrite your waveform script file.

Hardware Workflow (Quartus)

- Make sure the board is off before connecting to the computer's USB, then power the board on (red push button).
- In Quartus, ensure that your top-level module is set as the project's top-level module then use the **Compilation tool**: 
 - This typically takes a while to run (2–10 minutes).
- Use File → Open... to open the **Programmer interface** via `ProgramTheDE1_SoC.cdf`.
 - Need to change the file type to "Programming Files":



- Assuming no issues, click "**Start**" to program your DE1-SoC!