LAB 0: FPGA BASIC METHODOLOGY

James E. Stine, Jr.
Edward Joullian Endowed Chair in Engineering
Oklahoma State University
Electrical and Computer Engineering Department
Stillwater, OK 74078 USA
james.stine@okstate.edu

Objective

- The first lab has three main objectives:
  - The first objective is to learn how to design a basic digital design and verify it using a testbench.
  - The next objective is to implement it on your DSDB board and compare it to what you get from simulation.
  - The last objective is to understand the lab structure and report.
- History has indicated many people will wait – you cannot complete any lab overnight, be forewarned.
  - We will revisit lab report later today.
  - Also, you probably cannot finish this lab in one lab session.

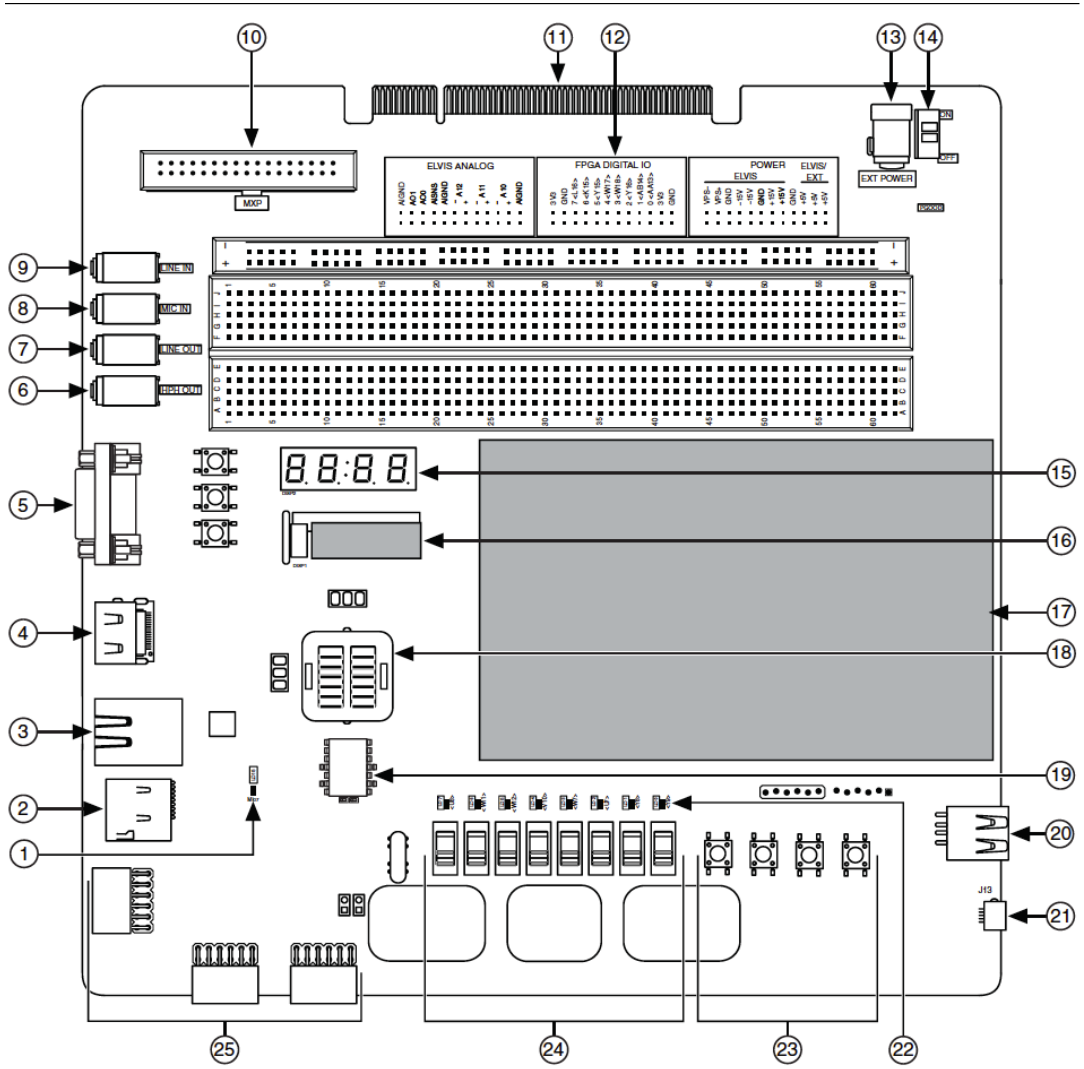
Things to keep in mind

- For future labs, start early and work with your partner to get things done quickly.
  - Communication is key to a good relationship.
  - Do not ask what I can do, just jump in and contribute even if you duplicate work.
- Labs are due one week after the lab is complete.
  - For lab 0, this Monday September 11
- Try to schedule the last week of lab for writing up your work, so you basically only have 2 weeks to complete this lab.
  - Back up often and work upwards (inverted pyramid).
  - Ask questions on slack or in person ***early***!
  - Know we are here to help – post screenshots/questions on slack or stop by.

Basics

- For this laboratory, we will be using the ELVIS III board.
  - Your ELVIS III board will also have a Digital System Development board fabricated by Digilent (https://digilent.com).
  - It was designed for National Instruments (NI) but we will be using it independently with the AMD Vivado environment.
- We have made most of it very simple if you follow the procedure and understand what is going on.
  - However, you want to ask questions as I am sure it is new for many or almost all of you.
- The laboratory document and its associated files are on the GitHub repo.
  - There are also important ancillary files in the repository.
- Make sure you have a DSDB board at your desk before you begin.
  - Ask your TA to install it, if its not there.

DSDB



18: Zynq XC7Z020-1CLG484C with included Heat Sink

All the FPGA pins are accessible via the DSDB board either through the schematic or attaching to the debugger

About Git

- Created by Linus Torvalds, creator of Linux in 2005
  - Came out of Linux development community and his dissatisfaction that there was no good open-source tool (actually, read more of the history!)
  - Designed to handle version control on Linux
- Goals of git
  - Speed
  - Support for non-linear development (could be thousand of branches)
  - Fully distributed
  - Able to handle large projects efficiently
- Etymology
  - A “git” is a cranky old man, and Linux was alluding to himself in a funny sort of way.

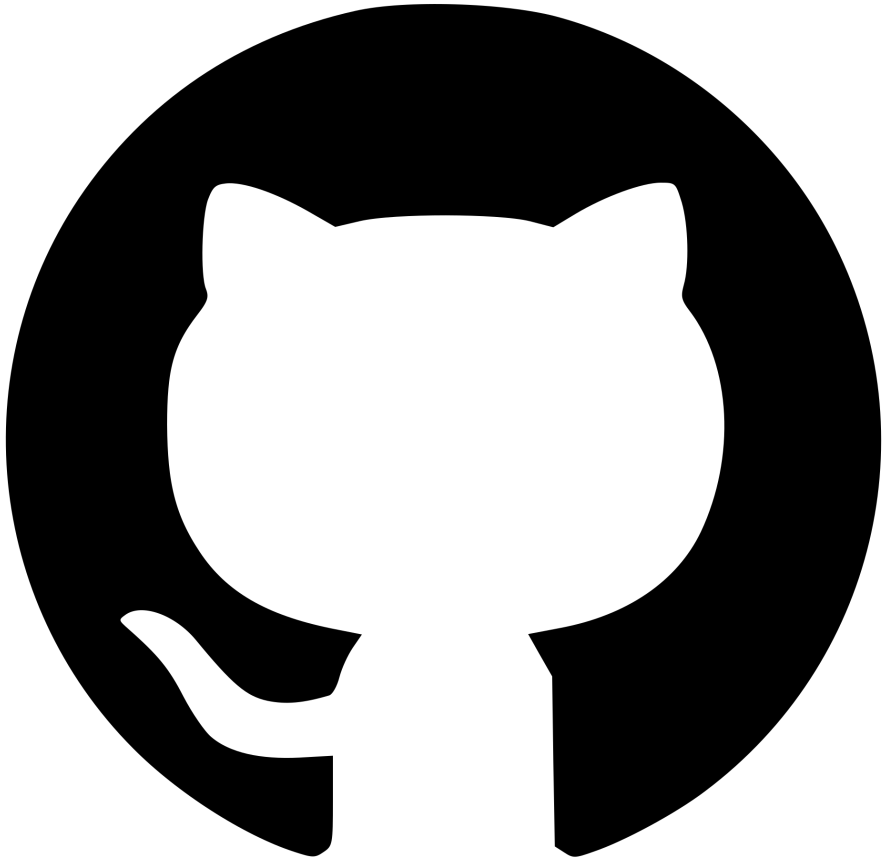
”Here’s the story…”

- `git` is not the first program to handle version control
  - Version control is nothing more than being able to work on a computer project within a team.
- Some former version control programs
  - RCS
  - SVN (sub-version)
- All of these tools really help people collaborate on a project as well as annotate each-other’s code.
- In reality, the best tool here is one where all parties contribute!
  - However, I believe `git` has evolved to be a better program than previous versions.
  - There are also web interfaces:
    - http://github.com
    - http://gitlab.com

[Paramount Home Entertainment]

GitHub

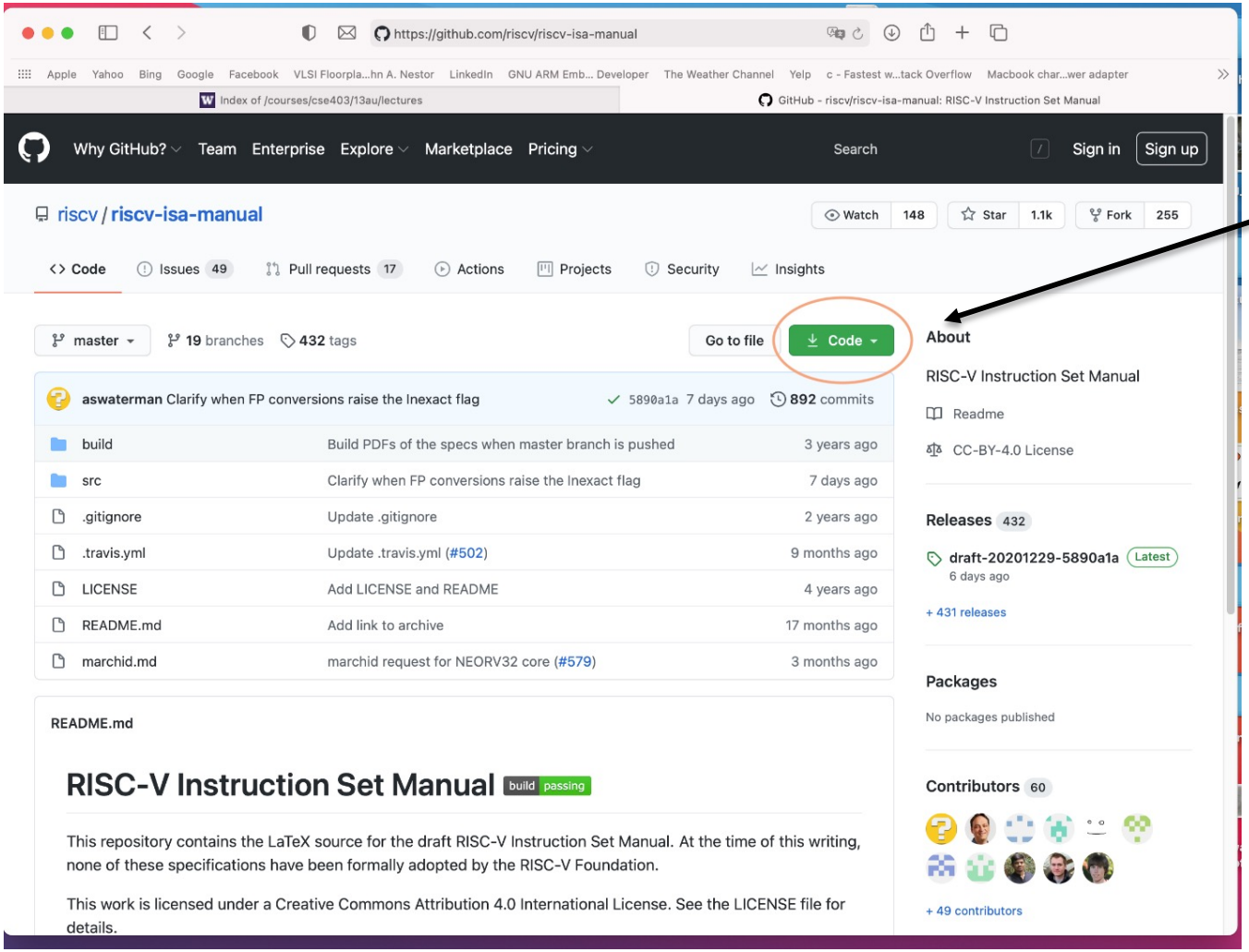
- GitHub was invented as a front-end to git.
- You can still access git on your terminal (more on this later).
- Microsoft purchased it in 2018, and it has taken off
  - You can do lots of cool stuff with GitHub (e.g., you can store all your notes, make code available for others)
- Companies have integrated GitHub inside their organization to help users be more productive.
- All GitHub does is really provide a place to store stuff and integrate git commands.



Example

- Find a repo you wish to clone
  - This can be a version stored in a directory somewhere
    - Directly copying the repo can be dangerous, as if you modify any permissions, it wipes out data information and thus ruins the repo.
  - It is far easier to pull it from a web-based repo (with security in mind, of course)
- Here is an example of the RISC-V documentation:
  - https://github.com/riscv/riscv-isa-manual
  - It is also easier to perform a clone at the terminal than using some form of GUI
- Most online repos will have a "CODE" button that gives you the location that you can insert into a clone command:
  - `git clone https://github.com/riscv/riscv-isa-manual.git`

GitHub site



Click here to see clone option

Git repo

- Don’t forget to review your git lecture that was recorded on Week 1.
- The repository is: https://github.com/stineje/dldfall2024
  - It is also linked on Canvas
- Remember to clone the files to your current location
  - `git clone git@github.com:stineje/dldfall2024.git`
- For those that want to do git on the terminal, check out my Google Doc on how to add ssh keys:
  - https://docs.google.com/document/d/1LjDd0m2bIPFHWujhgNF8_TFv81QWNCJMum7kMYL7goo/edit?usp=sharing
  - Otherwise
    - `git clone https://github.com/stineje/dldfall2024.git`
- If you get any errors, just let us know on Slack

Trifecta of commands

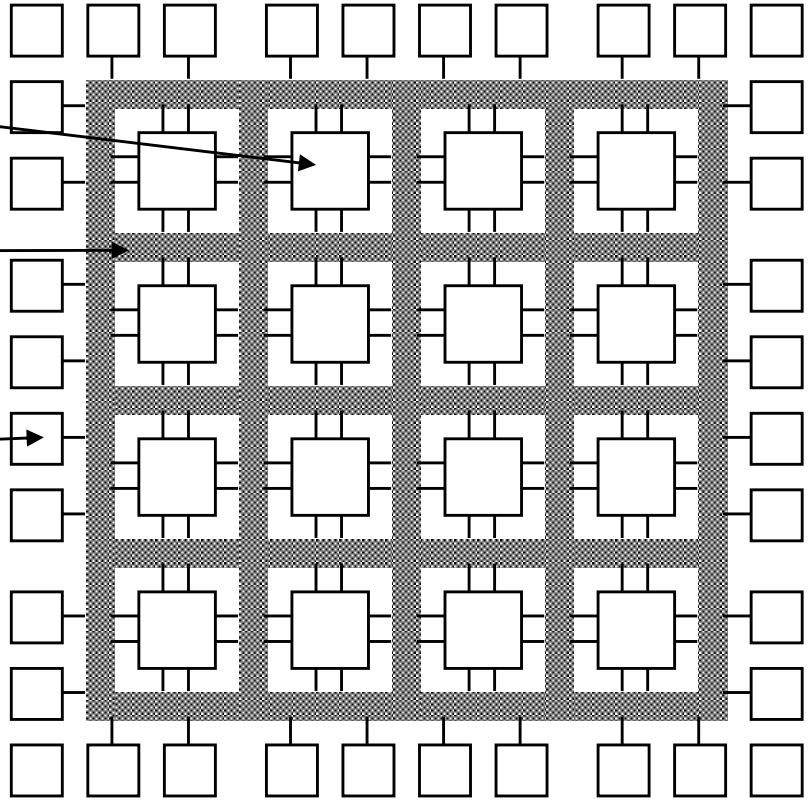
Trifecta		
1st Place	1	1st Place Payout
2st Place	2	2st Place Payout
3st Place	3	3st Place Payout
4st Place		4st Place Payout
5st Place		5st Place Payout
6st Place		6st Place Payout
7st Place		7st Place Payout
8st Place		8st Place Payout

[Wikipedia]

- There is a trifecta of commands that makes things easier when you want to move your staged stuff into a commit.
- I try to remember these but its good before you commit to see if anyone else has updated otherwise you may need to manually merge changes (later)
  - To get the most recent version of your repo, first, just ”pull” it
  - `git pull`
- Trifecta of commands!
  - `git add file`
  - `git commit -m “Commit message”`
  - `git push`
- It is highly advisable to add as much information about what you are committing in the commit message to help the next user.
  - Do not create a message like, “Update to files” – make it detailed!

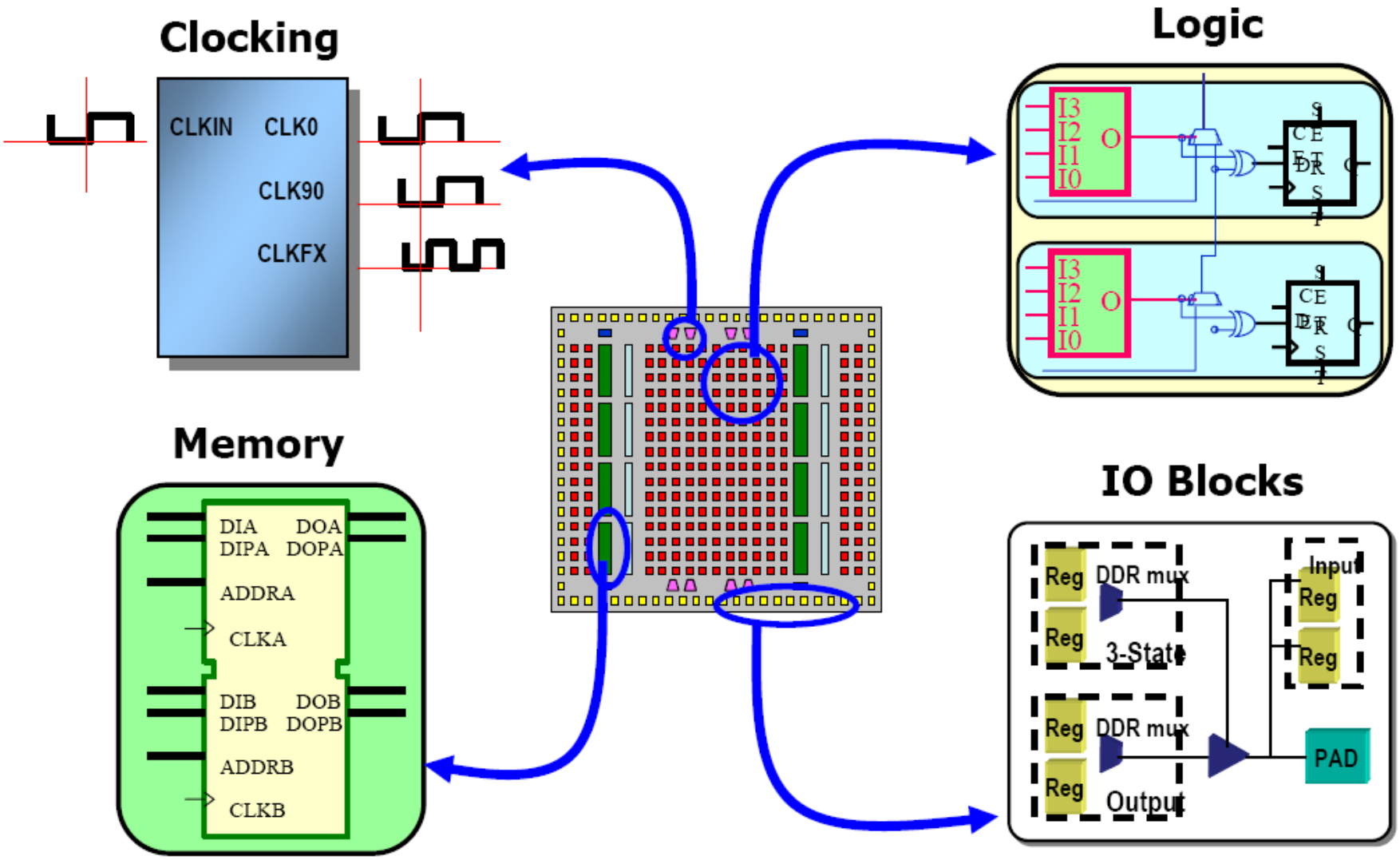
Field-Programmable Gate Arrays

- Logic blocks
  - To implement combinational and sequential logic
- Interconnect
  - Wires to connect inputs and outputs to logic blocks
- I/O blocks
  - Special logic blocks at periphery of device for external connections



- Key questions:
  - How to make logic blocks programmable?
  - How to connect the wires?
  - *After the chip has been fabbed* or fabricated!

Evolution of FPGA architectures



Tasks

- Download `labs0.zip` from Canvas
  - Unzip your files somewhere (e.g., Desktop)
- First test `sillyfunction` SystemVerilog with ModelSim
  - Navigate to your unzipped files.
  - Try compiling `silly.sv` with `vlog silly.sv`
    - Examine testbench with text editor to see what is happening
  - Simulate and Examine with ModelSim: `vsim -do silly.do`
- Implement the full adder (3 bits in, 2 bits out) in SystemVerilog
  - Modify the `silly_tb.sv` for the FA (just change the silly tb vectors)
  - Simulate with ModelSim and Check that it works
    - You might want to write down the correct truth table on paper
    - Examine that the truth table matches the waveform for every combination ($2^3 = 8$ combinations)
- Much of these tasks can be done on your own outside or inside laboratory

Key Ideas for Labs

- The key idea for this lab is to get used to the ModelSim and Vivado software.
  - ModelSim is used for development and Vivado is used for programming the FPGA board.
  - You should spend most of your time in the ModelSim environment.
- This laboratory is also familiarizing yourself with the lab report and how to write up your lab report.
- It is also an introduction to your lab partner and communication.
- A subtle part of the laboratory is understanding time management and getting things done.
  - The first lab report may take some time to understand the procedure.

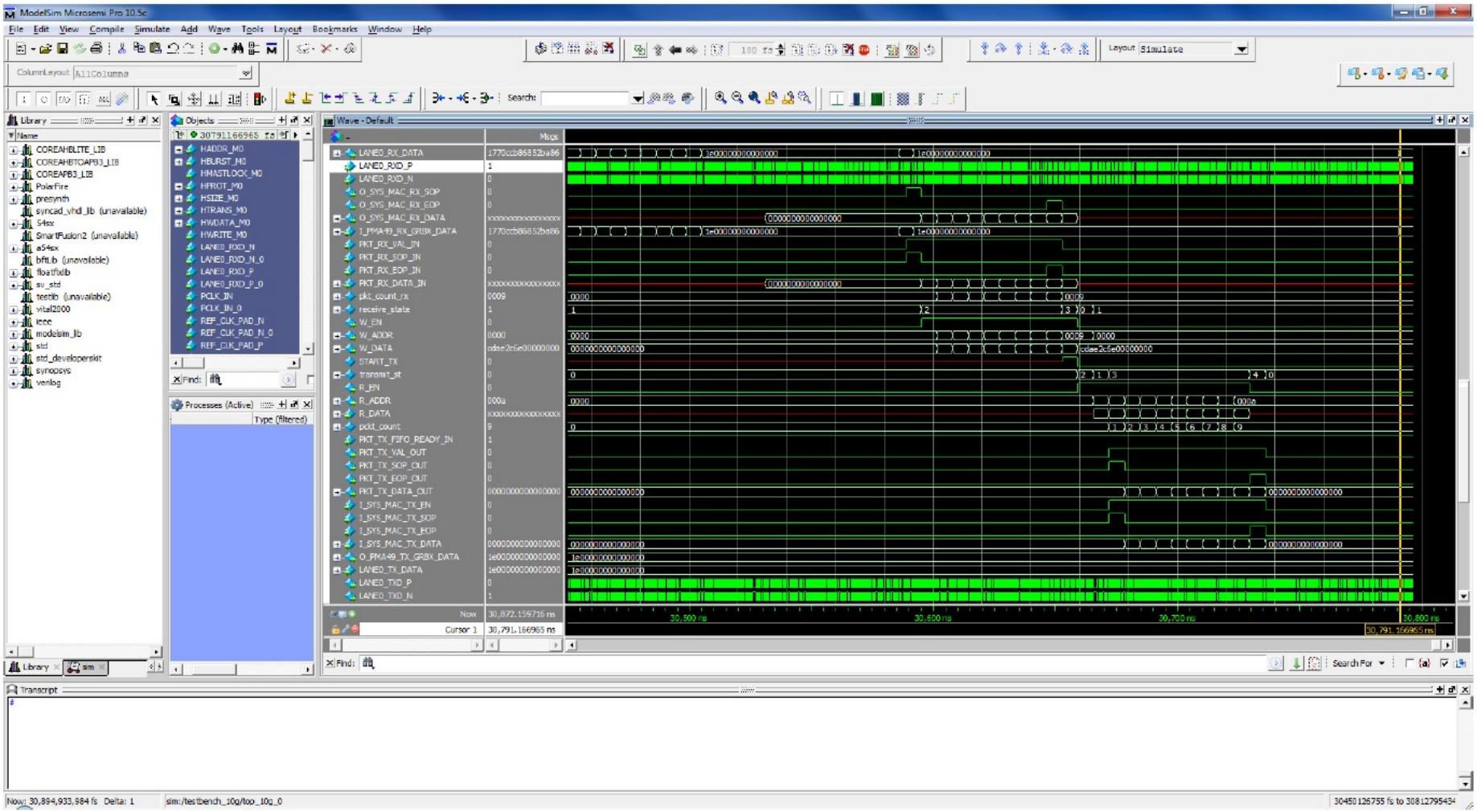
The three secrets to building successful engineering and product relationships

- Engineering and product/project managers embrace the blurry line.
  - The product or project manager is technical your boss and your job is getting the job done.
  - Work independently but get together to go over what you have done – do not under any circumstances accept the work from the other engineer and not do any work or try their implementation on your end.
  - You can work on separate items, but you should know how your team did something.
- EMs and PMs work to disagree productively.
  - The goal of an EM and PM partnership is to ship good software/hardware that meets the needs of users.
  - When you take two highly motivated people who care about doing this well, it's inevitable that you'll disagree from time to time.
  - Teamwork and communication is absolutely critical!!!
- EMs and PMs give good feedback.
  - As you spend time working with your product partner, it's almost certain you'll notice things that make them less effective in working with you or others.

Demo of Simulation

- Let's try to simulate `silly.do` in the repository

MGC ModelSim `vsim` window



Debugging?

- You will be implementing your design on your FPGA on the National Instruments DSDB board.
  - The implementation will be inside the silicon which you cannot see.
  - So, how do you figure out if something is working? How do you tell something does not work the way you expected it to?
- Use debugging techniques!
  - This includes switches, LEDs, push buttons to help you examine what is happening with the board.
  - Try to use as many LEDs as you can to see what is on and what is off.
  - Use your brain to figure out possible problems.

Implementation on DSDB

- Once you simulate your design, you should hook up to the DSDB.
- Inside your `lab0.zip` there should be a Demo program.
  - Unzip and open with Vivado
- Modify the `top_demo.sv` to instantiate your "working" SV and hook up inputs and outputs to things you can see on the DSDB:
  - Hook your input/output of your instantiated design to debugging items (e.g., LEDs).
  - Test all 8 (why 8?) possible combinations of your full adder to make sure it is working as

Port	Type	Description
`sw[3:0]`	Input	push buttons (#23)
`btn[7:0]`	Input	SPDT slide switches (#24)
`led[7:0]`	Output	Light Emitting Diodes (LEDs) (#22)

Table 1: Ports Used for Lab 0

- Using Vivado (which is Xilinx's synthesis and place/route tool) - this should be on your desktop in ENDV 350 to run
  1. Run Synthesis
  2. Run Implementation
  3. Generate Bitstream
- Per the lab0.pdf, download to the board using the Hardware Manager

Check on DSDB board

- Note for all Labs: DO NOT TRY AN IMPLEMENTATION ON YOUR DSDB BOARD UNTIL IT WORKS IN SIMULATION!
  - It takes several minutes to go through the three steps inside Vivado (e.g., synthesis).
  - You want to hopefully do the implementation once or twice.
  - Use your debugging items and your brain to make sure things are working.
- Check on DSDB board whether things work (check against the truth table).
  - Use the Light-Emitting Diodes (LEDs) to make sure all 8 possible cases match your simulation from ModelSim and theory.
- If you like, save any output from the Vivado screen for your report
  - e.g., Schematic, Implementation Results
- Write your report!

Lab Report

- The lab report should be simple and efficient with 5 sections (much of this information is in the rubric)
  - Section 1: Introduction
  - Section 2: Baseline Design
  - Section 3: Design
  - Section 4: Testing Strategy
  - Section 5: Evaluation
  - Team Evaluation

One report for each team but all team members *must* submit a Team Evaluation to get full credit!

- Check Lab Report Formats
  - Recommend a 10-pt font
  - Your report should be approximately a **maximum** of four pages not including figures/tables.
- Do not read too much into the rubric: just get it done to showcase what you did.

Sample Lab Report

- Some previous TAs created a sample lab report for Lab 0
- Yes, it looks like much of what is in this lab report is the lab
  - Of course; I want you to see the structure and what to do in laboratory.
  - This laboratory is about learning the procedure and gaining exposure to ENDV and the ELVIS/DSDB boards
- Do not assume you can get these labs done in 2 hours – work at understanding how to get things done.
  - Ask people questions!
  - Learn, Learn, Learn!

Last-minute Items

- Challenge yourself and get things done early!
  - Last semesters students waited until the second week when its too late!
  - You may have to visit the lab more than once! (i.e., you cannot always complete your work in one lab session)
- Submit all HDL, scripts, lab reports, and team evaluation for each member through Canvas.
  - Do not forget your HDL, testbenches, DO files or you will get a deduction.
  - Also, do not forget the team evaluation from each team member!!!!
- Believe in yourself – we all tend to be negative about our abilities (don't!)
  - You can do it!
  - OSU students are the best!
- Enjoy the opportunity!