

The following is just a **sample** lab report for ECE 385; by no means do you need to follow the outline exactly. Keep in mind that this is just an outline, and you will need to provide as much detail as you think is necessary to explain your circuit and its operation. The purpose of this is to provide a general example of what should be contained in a lab report for ECE 385. Please refer to pages GG.18-GG.21 of the ECE 385 lab manual for a description of what should be included in your lab reports.

ECE 385

Fall 2014

Experiment #

Lab Title

Your Names

Your Lab Section/Day & Time

Your TA's Name

Purpose of Circuit

This is where you should briefly (a few sentences) describe the purpose of this lab, i.e. is this lab about flip-flops? Memory? Computation unit? How is this lab useful in the context of a processor architecture?

Written Description of Circuit

This is where you should explain the operation of your circuit in detail. Include a description of any logic that you have designed, and explain **how you came up with your design**. This means things like K-Maps, state diagrams, truth tables, etc... A rule of thumb is that for anything you have done towards the designing your circuit, you should include in your lab report in an organized fashion that makes sense (we've seen many lab reports with scribbled pages as attachment. This is not a good presentation of your work)

Do not just list the components in your circuit (e.g. "this design uses 4 NAND gates and an inverter to perform the function; we drive input B with this signal, and this is what we got for the output"). This is the job of the logic diagrams and the layout sheet. Rather, a well-informed circuit description should be a more detailed explanation of the 'Purpose of Circuit', i.e., if you are working on a memory system, what components does the system need (e.g., registers for the actual storage; MUXs for data selection; control unit for read/write request)? How are the registers arranged (i.e., in series/parallel? How many of them? How large is each storage space)? How do you operate the system (e.g., the circuit is manually resetted. If we wish to store data X into address Y, we need to operate control Z, then the circuit goes through processes A→D to complete the storing action...)

High Level Block Diagram

For more complex designs, divide your design into blocks and include a top-level block diagram showing only the major components of your design and their interconnections. Block diagrams are higher level representation of your logic diagram. It divides your logic diagram into large 'chunks' into black boxes (e.g. Fig. 1 on pp. 3.2), where each box serves a meaningful purpose in the large picture of your design. The block diagram does not give you the details of the implementation at all, but its purpose is to introduce your design and your circuit in an abstract way, so people will be able to fully grasp the big idea in a short amount of time. Please note that block diagrams are needed in your lab reports. The more logical you split up your design modules, the more likely you can clear-cut them apart for wiring and debugging purposes (it also enhances the organization and hence the readability of your design).

Logic Diagrams

Include a logic diagram showing the gate-level layout of your circuit. Be sure to label all inputs, outputs, pin numbers, as well as the chip locations on the protoboard (B1, C3, etc. in coordination with the component layout sheet). For an example see Figure 9 on page GG.21 of the ECE 385 lab manual.

State Diagrams and Tables

For any design that uses a state machine, you should include a state diagram and, optionally, transition tables to show how your state machine works.

Component Layout

Include a component layout of your circuit using the sheets provided in your lab manual. You should include the layout of each circuit (for two or more circuits for a single lab, please draw clear lines to separate them). For an example, see Figure 8 on page GG.20 of the ECE 385 lab manual.

For each chip of each circuit, you should draw out the pins (different chips will have different numbers of pins on each side). DO NOT label every single pin in your design, as it is impossible for anyone to read. DO label fundamental pins such as external IOs to the IO box, VDD, and GND. Remember to write the chip number in each component.

Answers to Pre-Lab Questions

You should include the answers to all of the pre-lab questions in your lab report. Answer these questions before you come into the lab to do the experiment, but correct the answers if you discover during or after the lab that you made a mistake.

Answers to Lab Questions

Answer any questions asked in the LAB section of the ECE 385 lab manual.

Answers to Post-Lab Questions

Answer any questions asked in the POST-LAB section of the ECE 385 lab manual.

Conclusions

What have you learned? What worked? What didn't? What can you possibly do to enhance/simplify/fix your circuit to make it better?

Extra Requirements for FPGA Labs: Block Diagram

Your SystemVerilog codes for the FPGA labs will be organized into separate entities, each being a block module connected to each other in your top-level module. You will need to print out a schematic diagram for your top-level module. See IQT. 23 for the conversion information. This detailed block diagram will replace the rough block diagram from the TTL labs.

Extra Requirements for FPGA Labs: Design Module Information

You will need to write the purpose and operation for each SystemVerilog module. For each module, you will need to list the inputs, outputs, and what the module does (i.e., is it a register? Control unit? Computation unit? What does it do to the inputs? What are the role of the outputs in the whole circuit?)

Extra Requirements for FPGA Labs: Design Statistics

Complete the design statistics table. Refer to the Design Resources and Statistics in IQT.25-27 in the lab manual for steps to obtain the information.

Extra Requirements for FPGA Labs: Annotated Simulation Waveform

For each circuit designed, you will need to provide a simulation waveform showing the essential operations of the circuit. That is, giving a set of inputs, you should provide enough operation details (state transitions, controlling signals, intermediate values, etc.) to show that your circuit is functionally correct.

To facilitate the reader reading your waveform, it is required that you annotate the waveform. "Annotation" means that you need to tell the reader what's going on in each step of the waveform. You should put comments along the simulation saying what's going on in each particular locations (e.g. here's the value for A... here's the value for B... Here we raise the signal to add A and B... Here we're adding A and B.... here's the result of A+B... here we're hitting "execute"... here we move to state x... here we're doing something in state x... and so on in the waveform).

Note that although you need to show the essential operations of the circuit, you do not necessarily need to show the operation of the entire circuit from head to toe. You will see in the later labs that sometimes showing everything will turn out to be impractical. For example, we will be dealing with keyboard input and VGA output in Lab 7. It is very tedious to simulate these inputs and outputs. To show the essential operations of your circuit, you can start your simulation assuming your circuit has already received a legitimate input (essentially skipping the simulation for the input), and show that your circuit does operate correctly by showing some correct information just prior to the final output. You might have to modify your final circuit a bit to be able to simulate the reduced circuit. Or, for good practice, you should simulate the core of your circuit and make sure it is functionally correct before you move on to connect the core to its inputs and outputs.

Take Lab 7 for example, you can reduce your simulation to the movement of the ball. Let the input to your reduced waveform be the direction of the keypress, and show that your ball_X_pos and ball_Y_pos update accordingly, especially for the boundary cases.