

CS152A Lab 1 Workshop 1

In the previous session you were given a tutorial of FPGA design and implementation, and went through the whole process using Xilinx's toolchain. In this session, you will explore the example's simulation process to learn more about behavioral simulation, debugging, and design.

Answer the following questions as much as you can and include the answers in your lab report.

Clock Enable

1. In the nexys3.v file, there is a signal/reg named `clk_en`. Read the section of code that's relevant to `clk_en` and try to understand what this signal does.
2. Add `clk_en` to the simulation's waveform tab and then run the simulation again. Use the cursor to find the periodicity of this signal. Capture a waveform picture that shows two occurrences of `clk_en`, and include it in the lab report. Indicate the exact period of the signal in the report.
3. What is the value of `clk_dv` signal during the clock cycle that `clk_en` is high?
4. Draw a simple schematic/diagram that illustrates the logical relationship amongst `clk_dv`, `clk_en`, and `clk_en_d` signals. Try to read the code for more clues.

Instruction Valid

1. Now move on to the signal `inst_vld`. Read the relevant code and use the simulation as your aid, answer the following questions in your lab report.
2. Write down the first simulation time interval (exact number) during which the expression `inst_vld = ~step_d[0] & step_d[1] & clk_en_d` evaluates to 1.
3. What is the purpose of `clk_en_d` signal when used in expression `~step_d[0] & step_d[1] & clk_en_d`? Why don't we use `clk_en`?
4. Include waveform captures that clearly show the timing relationship between `clk_en`, `step_d[1]`, `step_d[0]`, `btnS`, `clk_en_d`, and `inst_vld`.
5. Draw a simple schematic/diagram to illustrate the logical relationship amongst the signals above.

Register File

1. The sequencer's register file is located in a file called `seq_rf.v`. It stores the values of the four registers. Take a look at the source code and see if you can understand how it is implemented. Answer the following questions in the lab report.
2. Find the line of code where a register is written a non-zero value. Is this sequential logic or combinatorial logic?
3. Find the lines of code where the register values are read out from the register file. Is this sequential or combinatorial logic? If you were to manually implement the readout logic, what kind of logic elements would you use?

4. Draw a circuit diagram of the register file block.
5. Capture a waveform that shows the first time register 3 is written with a non-zero value.