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. **You'll have to simulate with the original source code to answer the questions**.

Clock Dividers

In the nexys3.v file, there is a signal/reg named clk_en. The clk_en represents a clock that is much slower than the master clock clk. Read the section of code that's relevant to clk_en and try to understand how the clock divider is implemented.

1. 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 (you can select the signal and use arrow keys to reach the exact edges). 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.
2. A duty cycle is the percentage of one period in which a signal or system is active: $D = \frac{T}{P} \times 100\%$, where D is the duty cycle, T is the interval where the signal is high, and p is the period. What is the exact duty cycle of clk_en signal?
3. What is the value of clk_dv signal during the clock cycle that clk_en is high?
4. Draw a simple schematic/diagram of signals clk_dv, clk_en, and clk_en_d signals. It should be a translation of the corresponding Verilog code.

Debouncing

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.

1. 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?
2. Instead of clk_en <= clk_dv_inc[17], can we do clk_en <= clk_dv[16], making the duty cycle of clk_en 50%? Why?
3. 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.
4. Draw a simple schematic/diagram of the signals above. It should be a translation of the corresponding Verilog code.

Register File

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.

1. Find the line of code where a register is written a non-zero value. Is this sequential logic or combinatorial logic?
2. 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?
5. Draw a circuit diagram of the register file block. It should be a translation of the corresponding Verilog code.
3. Capture a waveform that shows the first time register 3 is written with a non-zero value.