**ECEN 449 - Lab Report**

**Lab Number:** 1

**Lab Title:** Using Vivado

**Section Number:** 508

**Student's Name:** Samuel Fafel

**Date Performed:** 01-26-2023

**Date Due:** 02-02-2023

**TA:** Prajwal Holla

**Purpose/Introduction:**

This lab's purpose is to refamiliarize students with the Xilinx/Vivado systems in conjunction with the FPGA Zybo board. We will write simple Verilog code to run on the FPGA.

**Procedure:**

Summarize, in your own words, the procedures for each task and how you did them. Do not regurgitate the procedures step-by-step. Typically, you should include a couple of sentences per task.

1. Set up Xilinx/Vivado if applicable. Students are also able to use the computers in the lab, which already have Vivado installed.
2. Create a new project for 3 basic programs.
3. The first program maps the four switches on the FPGA board to the four LEDs on the board.
4. The second program creates a counter and uses the LEDs to keep track of the status of the counter. The counter is incremented, decremented, or reset using Button 0, Button 1, and Button 3, respectively.
5. The third and final program creates a "Jackpot" game where the LEDs light up rapidly in sequence. When a switch is toggled on, if the LED corresponding to the switch is currently lit up, the user "wins" and all four LEDs light up.

**Results:**

1. The first program (switch) creates a 4-bit input and a 4-bit output, and assigns the output bits to be equal to those of the input. The constraints file maps the four switches to the 4 input bits and the four LEDS to the four output bits.
2. The second program (counter) creates a clock and 3 button inputs, and keeps the same 4-bit LED output. Inside the module, a register is used to keep track of the count. At each positive edge of the clock, the module checks if any of the buttons are pressed, starting with BTN3 (corresponding to the reset button), then BTN0 (increment) and BTN1 (decrement). After adjusting the count according to the button pressed, it sets the LED output equal to the count.
3. The last program (jackpot) also used a clock and LEDs, but used switches instead of buttons. On each positive edge of the clock, the count would be incremented, then the LEDs would be assigned a value based on the count (4-bit, one at a time, sequentially) Lastly, if the switches input was equal to the LED output, all four leds would be turned on.

**Conclusion:**

In this lab, we once again became familiar with simple Verilog code for FPGA boards. We must remember how to use switches and buttons, as well as create clocks and map everything to the board's interface.

**Questions:**

1. BTN 0 is connected to K18, BTN 1 to P16, BTN 2 to K19, and BTN 3 to Y16. The signals are pulled up.
2. The purpose of an edge detection circuit is to detect the edge of a clock cycle (positive or negative). It could be used in this lab for the always blocks for posedge or negedge actions.

**Appendix:**

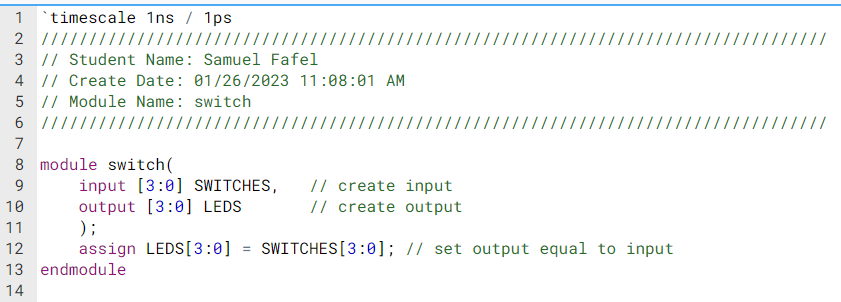


Figure 1: switch.v

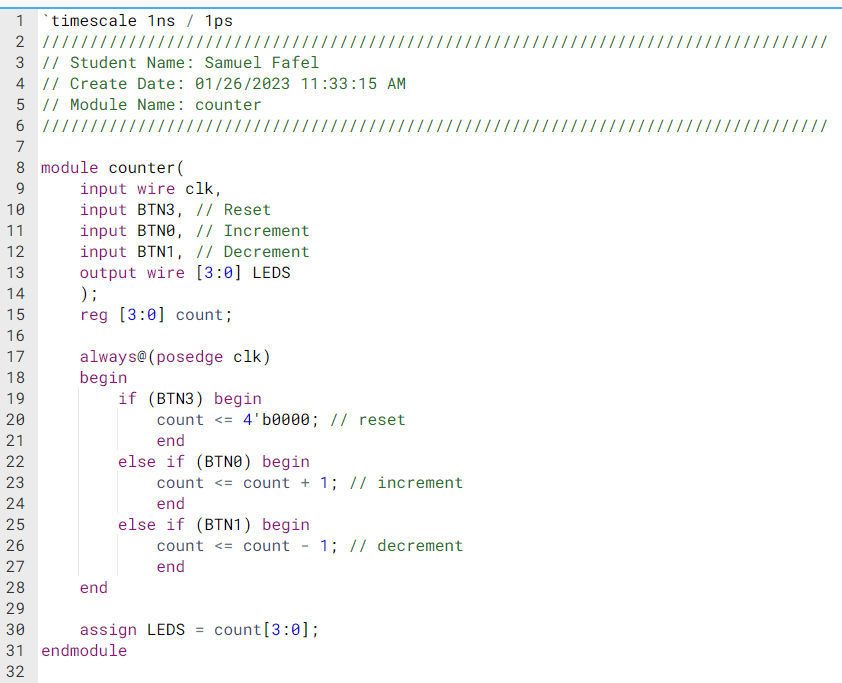


Figure 2: counter.v

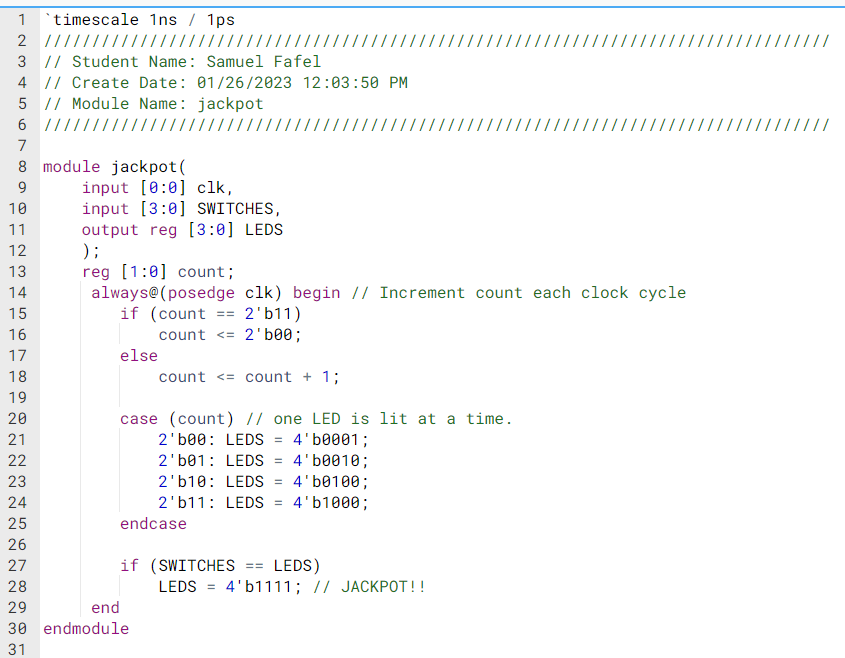


Figure 3: jackpot.v