Nicholas Tann

Lab 1

ECEN 449-503

1/28/2020

Introduction:

In Lab 1 we went through the process of setting up Xilinx, our software suite for programming Verilog circuits on to our FPGA board. We set up a new project and wrote a few simple circuits in Verilog to test on the FPGA.

Procedure:

First we set up a new development project. Then we wrote 3 Verilog circuits. The first Verilog circuit we wrote assigned 4 switches on the FPGA to control 4 LEDS also on the FPGA, by using an assign statement and defining the input and outputs of our Verilog module switches.v in our ucf file.

The second Verilog circuit we implemented a simple 4 bit counter with the LEDS. Its functionality included, a 1 hz clock signal (light changes every 1 sec), holding an UP button to count up, holding a DOWN button to count down, and an asynchronous reset button. To do this we had to implement a clock divider module, which is a variation of a counter. Then we had to implement an LED counter using the divided clock signal.

The final Verilog circuit we implemented a JackPot game. We set a new clock signal using the clock divider then wrote logic that would 1 by one shift the LEDS lights in a “one hot” fashion. If at anytime if you turn on a switch corresponding to the LED that is on, you trigger the winstate which will light all the LEDS.

Results:

Verilog source files Included in lab1.zip

Conclusion:

Although the lab may seem simple in terms of complexity, the actual implementation of using the FPGA is not. You must be careful when writing your Verilog to ensure that what you write can be synthesized by your software and FPGA, because there are many statements that are valid Verliog but can not actually be implemented. In particular non-blocking assignments should be avoided as well as using posedge on a multibit signal in your sensitivity list.

Questions:

1. The push buttons are located as follows,

   Btn 0 -> K18

   Btn 1 -> P16

   Btn 2 -> K19

   Btn 3 -> Y16

   The push-buttons use pull up resistors.

2. The purpose of an edge detection circuit is to know when a signal is transitioning states. It allows you to implement logic synchronously much easier. In our lab it was using to keep the changes in the leds occurring at every positive edge(change from low to high) of the Clock signal.