ECEN 449 Lab Report 1

Philip Smith - 624002014 (Sec. 511)

January 26, 2020

Introduction

The focus of Lab 1 was to set up the Xiling FPGA development environment and test it out with several simple programming excercises:

1. LED Switcher - Take DIP switches and display their state on the onboard LEDs
2. LED Counter - Divide the system clock and display a counter on the onbaord LEDs with the option to count up, count down, and reset
3. Simple jackpot game

Procedure

After setting up the development environment, I set up the constraints file (.xdc) which defined the pins to use in the verilog code. The first problem was to run a demonstration module which took user input at the DIP switches (defined as an input array) and displayed their logical status on the on-board LEDs (defined as an output array).

The second problem was to create a verilog module that would allow a user to count up or down (and reset the count) in a binary fashion. My approach was to use a register in the module (32 bits wide) to divide the clock frequency into something reasonable. This would increment the internal counter at a human speed. The inputs were constantly polled to see if the user wanted to count up, down, or reset the counter. It should be noted that for an n bit counter being used as a clock divider, the output frequency of bit n follows:

$$f_n = \frac{f_{clock}}{2^n}$$

The third problem was to create a "jackpot" style game. Using the same logic to divide the clock, though this time faster, I used a binary shift algorithm to create a one-hot display mode for the 4 on-board LEDs. The switches were again constantly polled at their rising edge to see if switch n was triggered at the same time LED n was. If this was the case the internal "won" state register would be set to HIGH and all 4 LEDs would be turned on until the user reset the game. The logic used to determine if a win condition has been met is as follows:

$$W = S \wedge L > 0$$

Where W is the win condition, S is the 4 bit array set by the DIP switch positions, and L is the current LED output array.

Results

All three programs ran as intended with minimal need to debug. The XDC file was created based on a repository by Digilent detailing the Zybo board's I/O connections[1].

[1] https://github.com/Digilent/digilent-xdc/blob/master/Zybo-Z7-Master.xdc

Conclusions

This lab offered much in the way of learning how to set up and use the Vivado design suite. In addition I learned the specifics of constraint files in regards to their use in FPGA development. I did not run into many issues with this particular lab other than having to refresh myself on Verilog. In addition I discovered that accessing Verilog over the TAMU VPN is not a viable option. Ultimately my roommate and I who are both in the same lab section have decided to purchase a Zybo board so that we can work on it at home and not feel rushed during the lab period.

Questions

1. The following table shows which pins are mapped to which input buttons on the development board:

Button	Pin
BTN0	K18
BTN1	P16
BTN2	K19
BTN3	Y16

2. The purpose of an edge detection circuit is to specify on which edge the program should react. There are two options available, rising and falling, which translate to the Verilog directives **posedge** and **negedge**, respectively.