Lab 1 – Introduction to Verilog HDL & the Lab Design Flow

Name: Von Kaukeano *Date*: 9/4/18

Course: Digital Circuit Design Lab – ECE 2613 *Section #*: 001

Summary/Abstract

In this lab, we are introduced to the programs used for labs and successfully use Verilog HDL along with gate primitives to show our design and test it with the cloud9 software. We demonstrate our bit file by placing the file on a USB drive and into our Basys3 board to test results.

Introduction

The background information needed is the basic knowledge of Verilog programming language and gate primitives along with how the gates determine the output by their logic truth tables. These determine how the Basys3 board responds to the switches being used by turning on the LED lights accordingly.

Procedure

In this lab we were given four gates and the schematic for the code. Using the truth table below we were able to determine the LED light from being on based on the positions of the switches.

A0,1,2,or3	B0,1,2 or 3	f0	f1	f2	f3
0	0	0	0	0	1
0	1	0	1	1	0
1	0	0	1	1	0
1	1	1	1	0	0

We opened our cloud9 software and entered our Verilog design. After creating the design code, we tested it in the test bench. During simulation we test for any mismatches to make sure that our code is correct. After receiving no mismatches, we create a bit file to be saved on our USB drive to be placed into our Basys3 board. We then tested our code by moving the switches to have the LED light to turn on accordingly.

Results

I had mismatches at first because I typed the Verilog incorrectly for the NAND gate (~). After fixing the syntax I had none and was able to get the board to communicate with the bit file with a few attempts.

Design Code

```
//
// lab1 : version 09/03/2018
//
`timescale 1ns / 1ps
module gates(
    output logic f0, output logic f1, output logic f2, output logic f3,
    input logic a0, input logic b0, input logic a1, input logic b1,
    input logic a2, input logic b2, input logic a3, input logic b3
    );

    // Write code starting here ...

assign f0 = (a0 & b0);
assign f1 = (a1 | b1);
assign f2 = (a2 ^ b2);
assign f3 = ~(a3 & b3);

endmodule
```

Simulation Results

```
****** xsim v2018.2 (64-bit)
  **** SW Build 2258646 on Thu Jun 14 20:02:38 MDT 2018
  **** IP Build 2256618 on Thu Jun 14 22:10:49 MDT 2018
    ** Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.

source xsim.dir/work.tb_gates/xsim_script.tcl
# xsim {work.tb_gates} -autoloadwcfg -tclbatch {tb_gates.tcl} -onerror quit
Vivado Simulator 2018.2
Time resolution is 1 ps
source tb_gates.tcl
## run all
Simulation complete - no mismatches!!!
$finish called at time : 80 ns : File "/home/tuh42003/2613_2018f/lab1/tb_gates.sv" Line 62
## exit
INFO: [Common 17-206] Exiting xsim at Tue Sep  4 13:47:55 2018...


Process exited with code: 0
```

Hardware Implementation

My design was demonstrated to Catherine on the afternoon of 9/4 during the lab period.

Conclusion

The Verilog matched my truth table logic and I was able to visually see how the demonstration of the lights react to the position of the switches according to each logic gate. I had no errors and were able to use the cloud9 test bench and the Basys3 board successfully.