Lab 2 Introduction to FPGA

Calvin Reese
cjreese@fortlewis.edu

2/3/22

1 Introduction

This lab we loaded 3 sets of simple programs to learn how to design logic code and program the FPGA

2 Materials and Methods

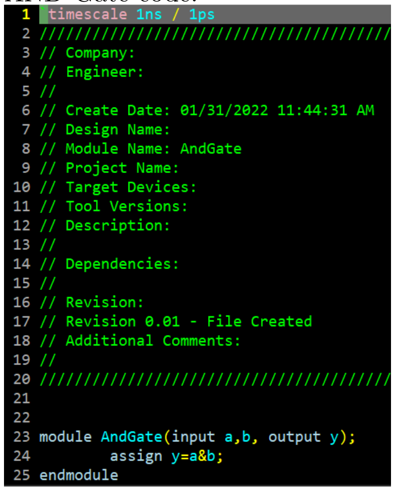
The tutorial for making these examples are in `http://www.yilectronics.com/Courses/CE433_Labs/s2022/Lab2_FPGA_Basics/Lab2.html`

3 Results

TASK 3: Video explaining volitile vs non-volitial programs loaded on the FPGA:

3.1 Task 1

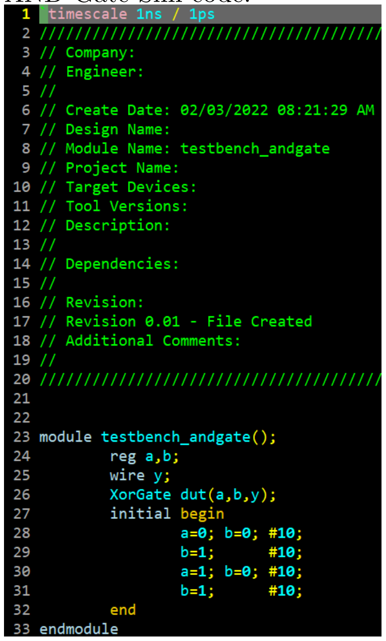
AND Gate code:



```
`timescale 1ns / 1ps
////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 01/31/2022 11:44:31 AM
// Design Name:
// Module Name: AndGate
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////


module AndGate(input a,b, output y);
        assign y=a&b;
endmodule
```

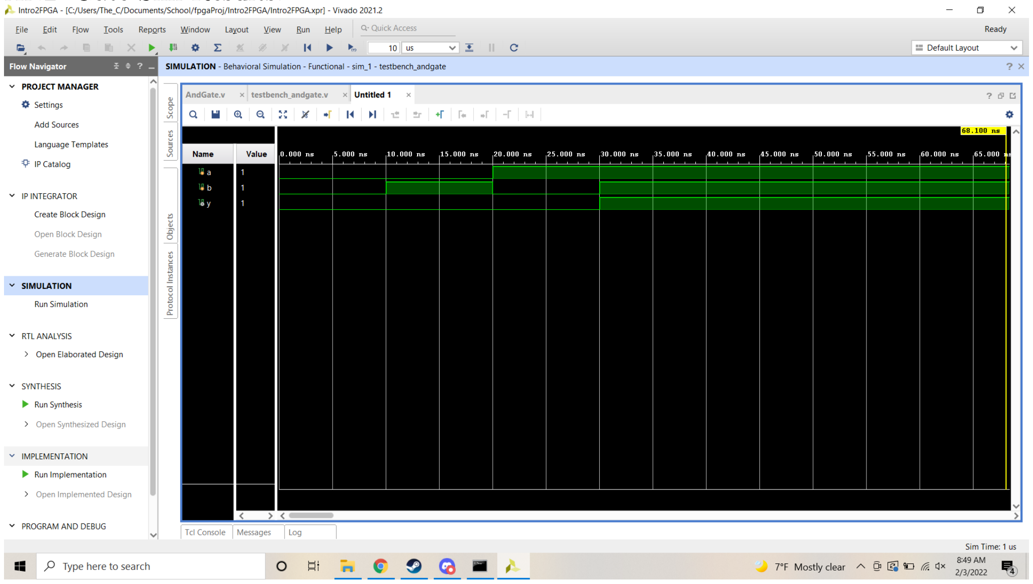
AND Gate Sim code:



```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 02/03/2022 08:21:29 AM
// Design Name:
// Module Name: testbench_andgate
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module testbench_andgate();
        reg a,b;
        wire y;
        XorGate dut(a,b,y);
        initial begin
                a=0; b=0; #10;
                b=1;      #10;
                a=1; b=0; #10;
                b=1;      #10;
        end
endmodule
```

AND Gate Sim Results:

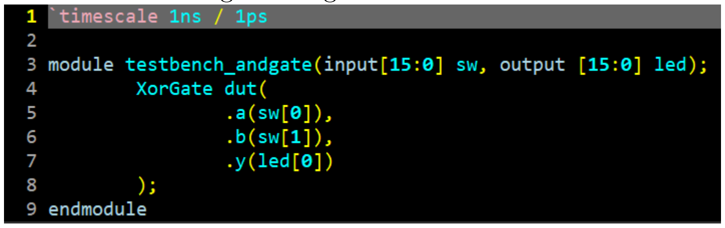


Intro2FPGA - [C:/Users/The_C/Documents/School/fpgaProj/Intro2FPGA/Intro2FPGA.xpr] - Vivado 2021.2

SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench_andgate

Name	Value
a	1
b	1
y	1

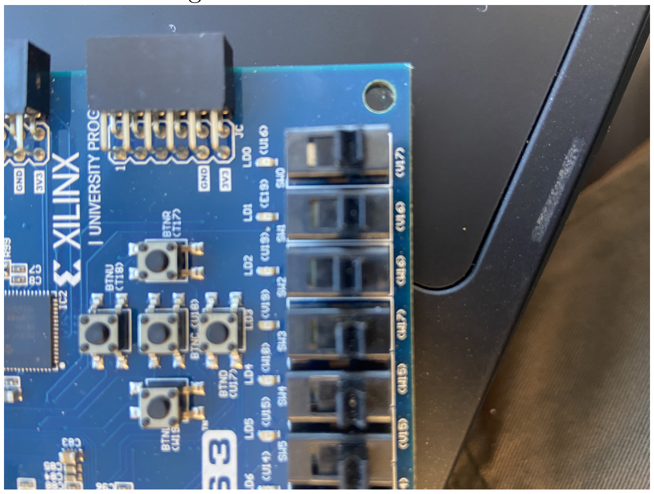
AND Gate Programming Code:



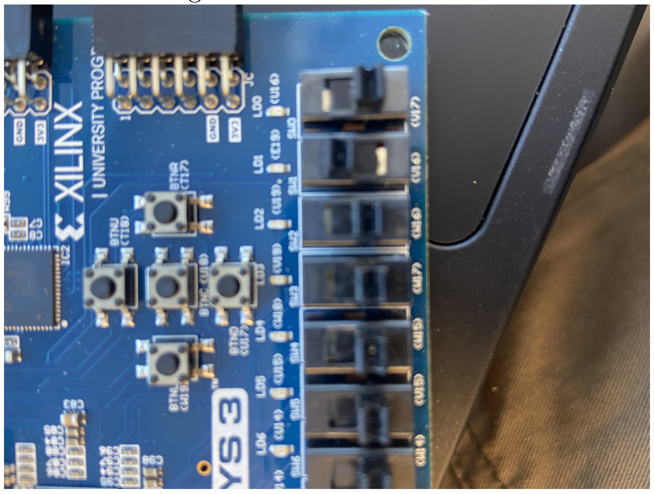
```
`timescale 1ns / 1ps

module testbench_andgate(input[15:0] sw, output [15:0] led);
        XorGate dut(
                .a(sw[0]),
                .b(sw[1]),
                .y(led[0])
        );
endmodule
```

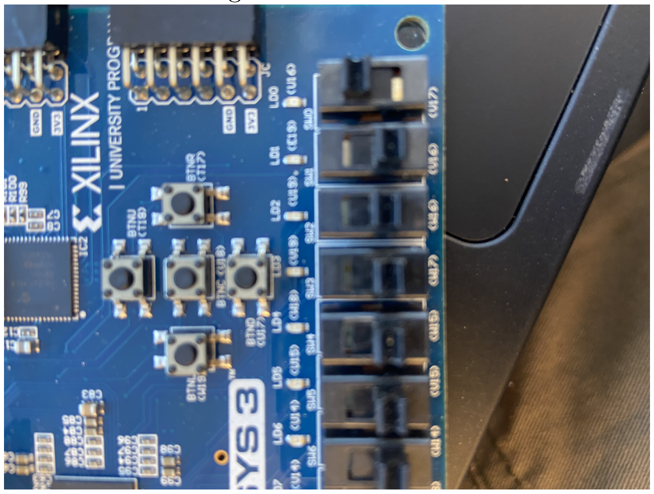
AND FPGA Logic 00



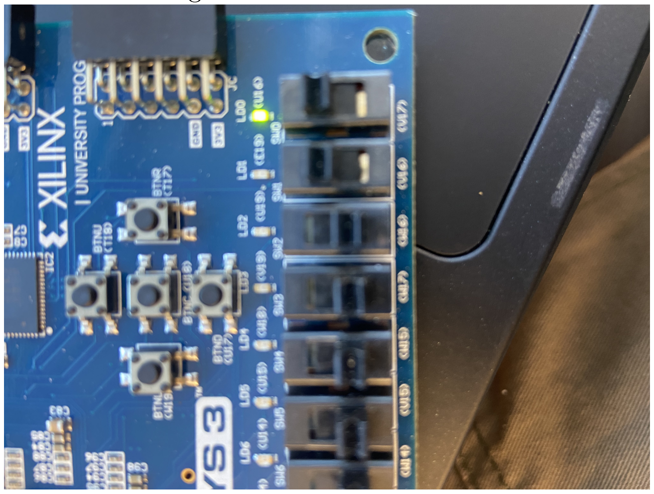
AND FPGA Logic 10



AND FPGA Logic 01

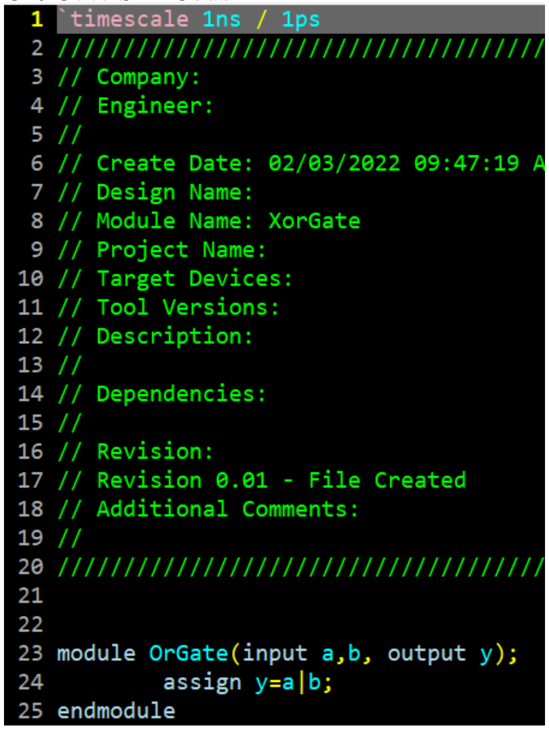


AND FPGA Logic 11



3.2 Task 2

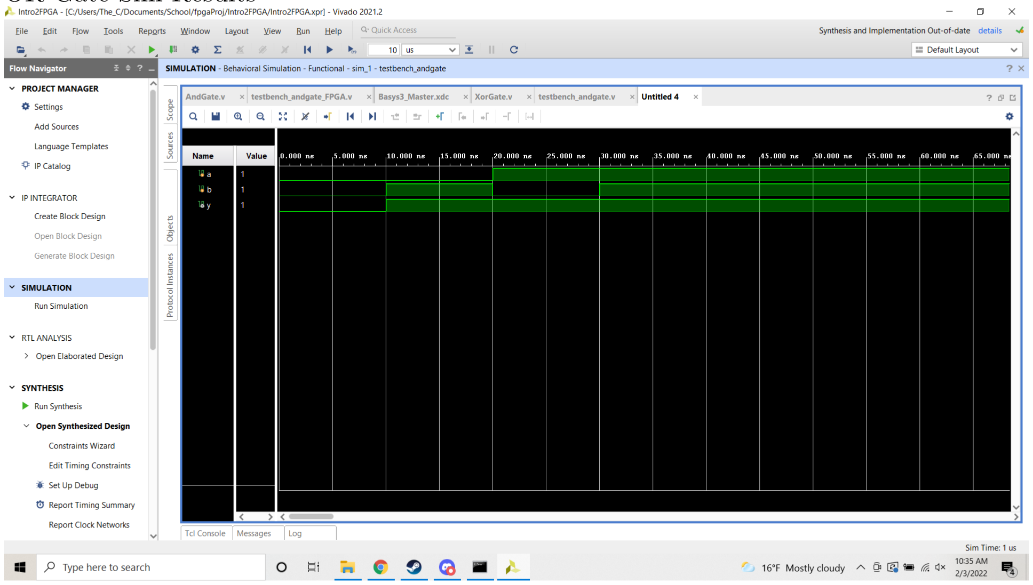
OR Gate Sim Code



```
`timescale 1ns / 1ps
////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 02/03/2022 09:47:19 A
// Design Name:
// Module Name: XorGate
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////


module OrGate(input a,b, output y);
        assign y=a|b;
endmodule
```

OR Gate Sim Results



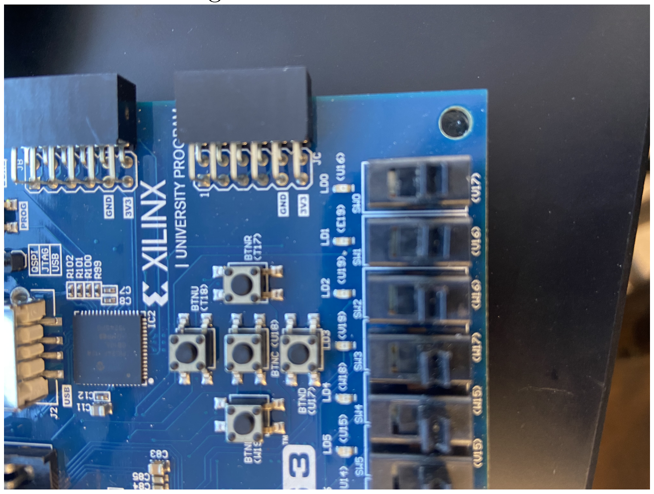
Intro2FPGA - [C:/Users/The_C/Documents/School/fpgaProj/Intro2FPGA/Intro2FPGA.xpr] - Vivado 2021.2

SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench_andgate

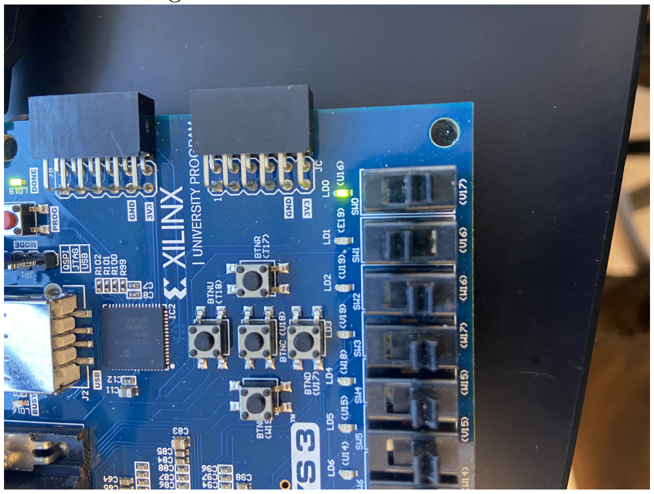
Name	Value
a	1
b	1
y	1

Sim Time: 1 us

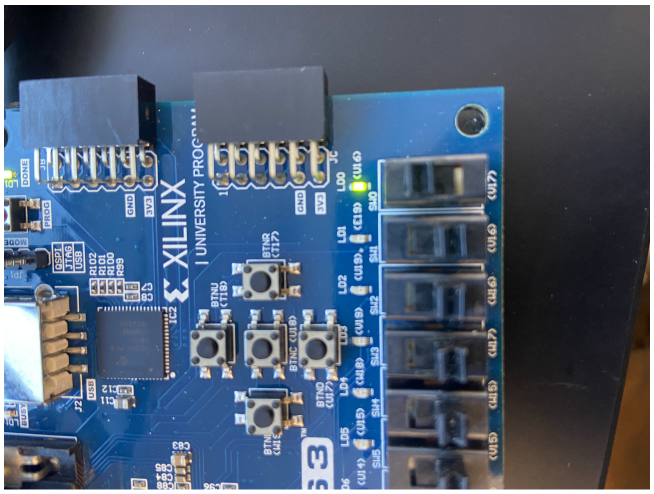
OR FPGA Logic 00

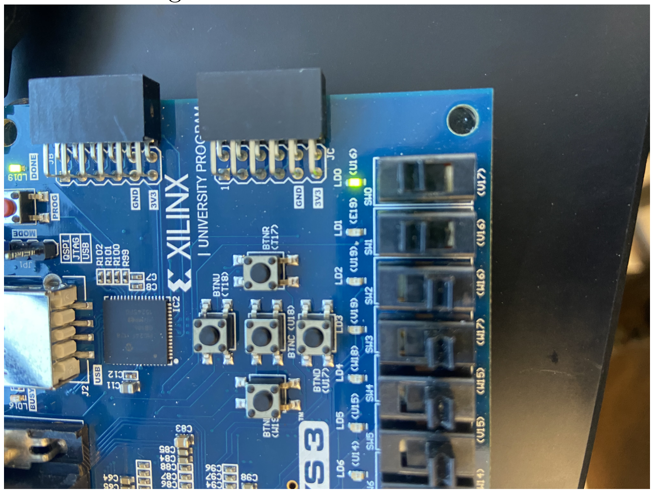


OR FPGA Logic 10

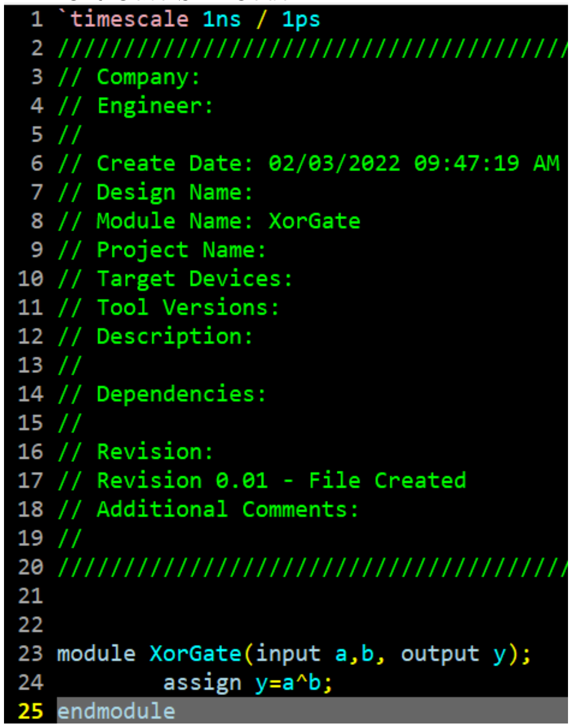


OR FPGA Logic 01


OR FPGA Logic 11



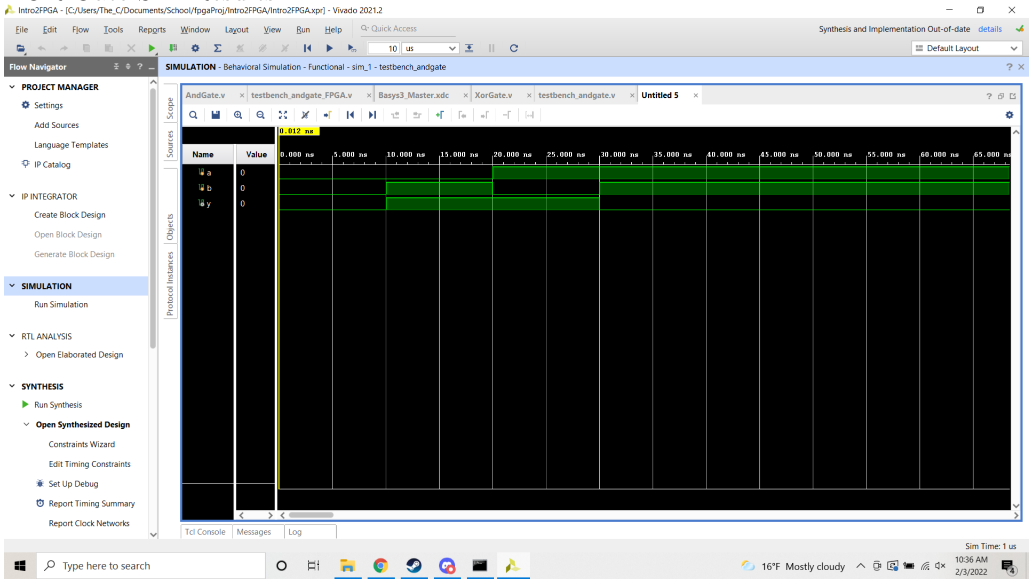
XOR Gate Sim Code



```
`timescale 1ns / 1ps
////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 02/03/2022 09:47:19 AM
// Design Name:
// Module Name: XorGate
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////


module XorGate(input a,b, output y);
        assign y=a^b;
endmodule
```

XOR Gate Sim Results



Intro2FPGA - [C:/Users/The_C/Documents/School/fpgaProj/Intro2FPGA/Intro2FPGA.xpr] - Vivado 2021.2

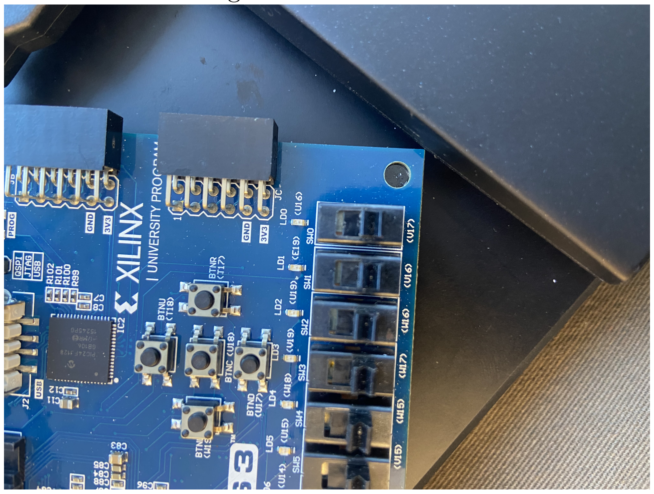
Synthesis and Implementation Out-of-date details

SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench_andgate

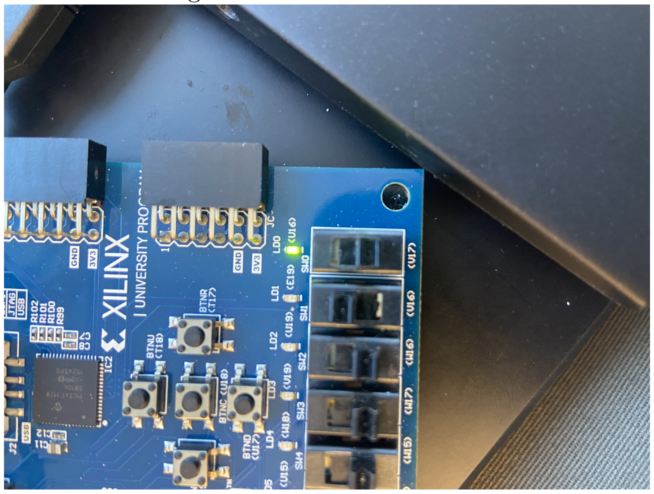
Name	Value
a	0
b	0
y	0

Sim Time: 1 us

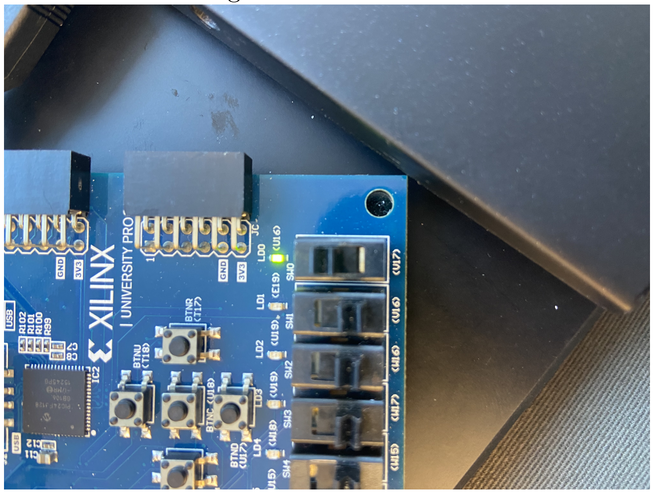
XOR FPGA Logic 00



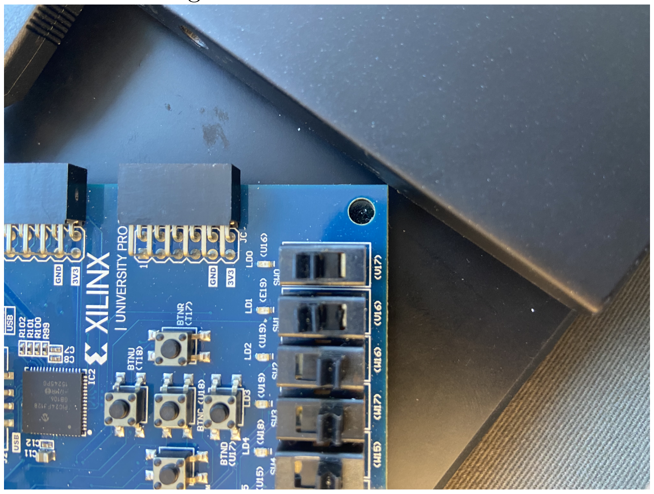
XOR FPGA Logic 10



XOR FPGA Logic 01



XOR FPGA Logic 11



4 Discussion

All of this was fairly simple just to follow your tutorial. I just reused the FPGA and Sim code for the OR and XOR for efficiency. I am excited to see where this takes us.