Experiment 3

Burglar Alarm

Digital Systems

EEE3342C-20FALL 0011

Spenser Tacinelli
PID: sp498810

Equipment:

The Xilinx's FPGA VIVADO HLx Editions design tools are available in the laboratory. These tools can also be downloaded from Xilinx's web site at www.xilinx.com. The WebPack version of this tool that we use for the laboratory experiments are located under the support download section at the related website (https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloa dNav/vivado-design-tools/2017-4.html). Please take note that the file download size exceeds 1 GB and also during the installation process updates may have to be installed. It can take you up to a few hours to download and install the software on your computer. The user does not need to have the BASYS development board interface to the computer to design and simulate an FPGA.

Objective:

To further investigate the operation of Xilinx's Vivado HLx by creating and designing a combinational logic circuit. To create a circuit of a burglar alarm controller circuit.

Pre-Laboratory Assignment:

1. Read this experiment carefully to become familiar with the requirement of this experiment.

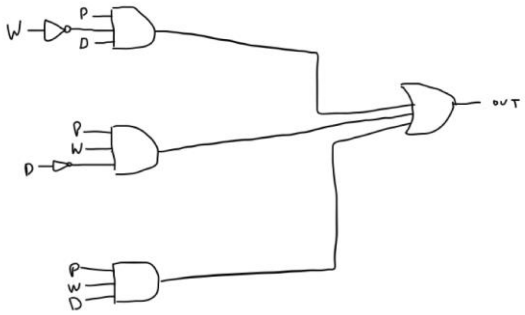
2. In your laboratory notebook, fill in the truth table for the burglar alarm for all possible inputs for P, W, and D.

P	W	D	OUT
1	0	1	1
1	1	0	1
1	1	1	1
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

3. Write the Boolean expression describing the burglar alarm controller:

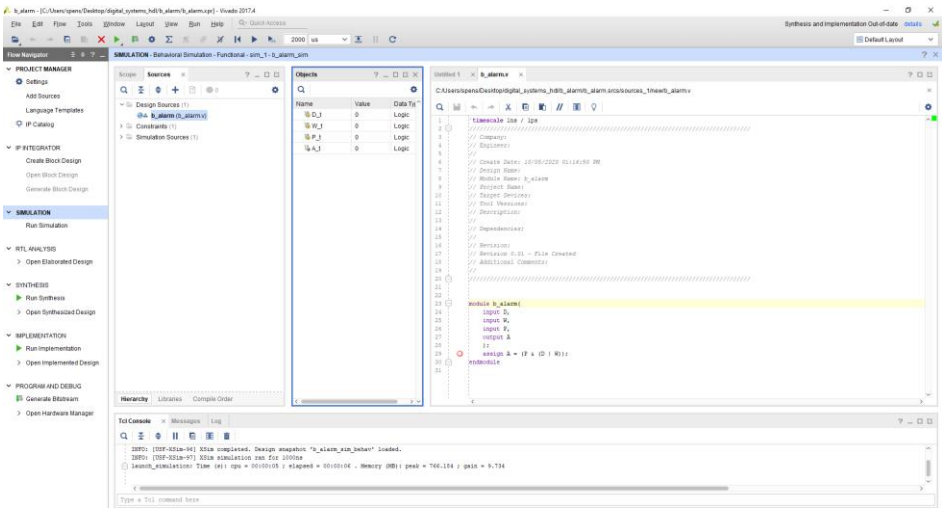
A = PW’D + PWD’+PWD

4. Draw the logic diagram for the burglar alarm.

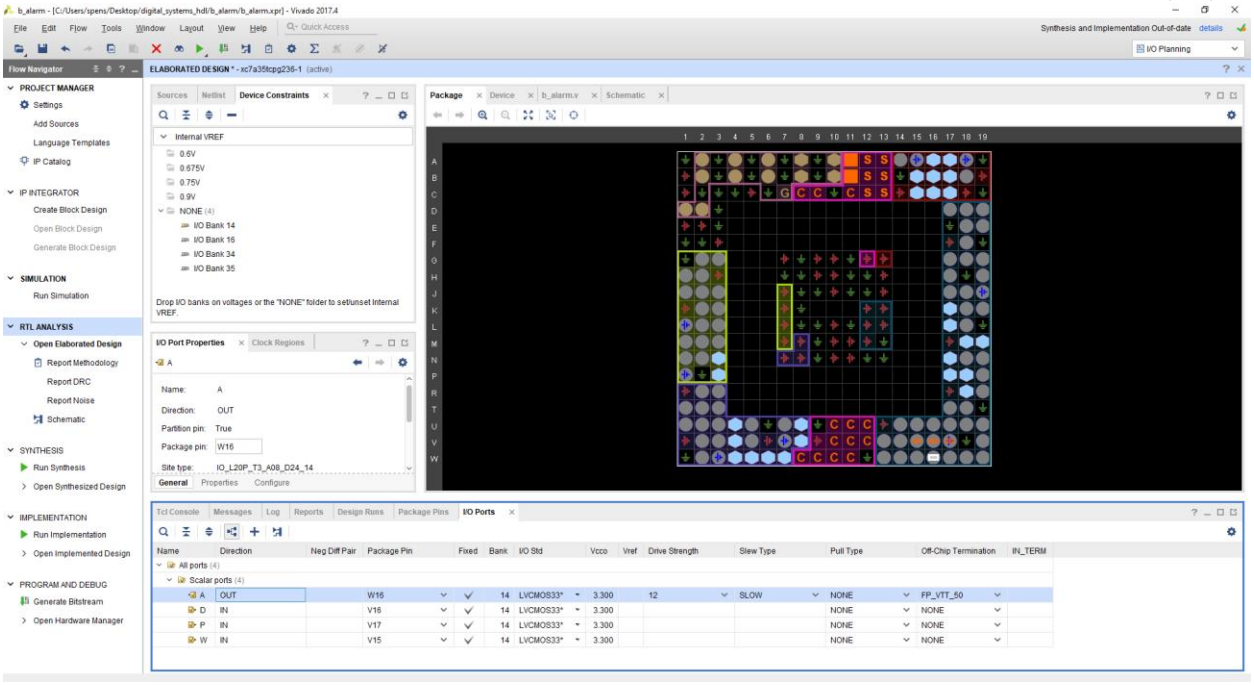


Design Steps:

I created the project (b_alarm) and added the three inputs and the single output. I then programmed the logic of the multi-functional gate.

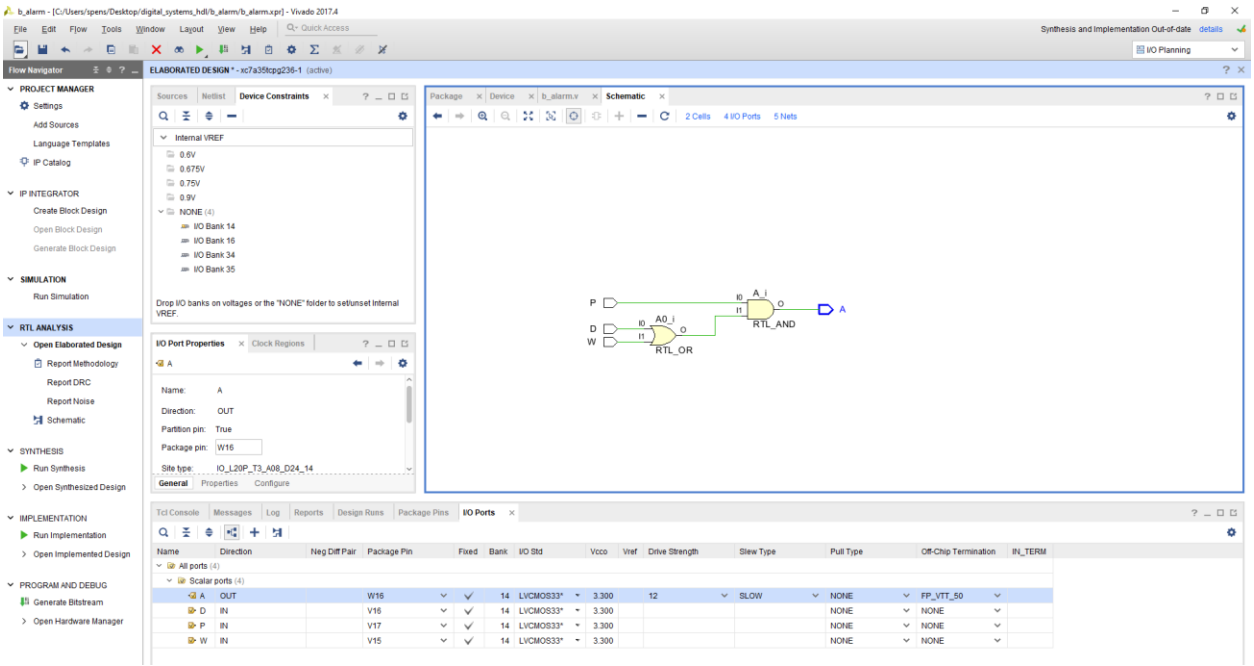


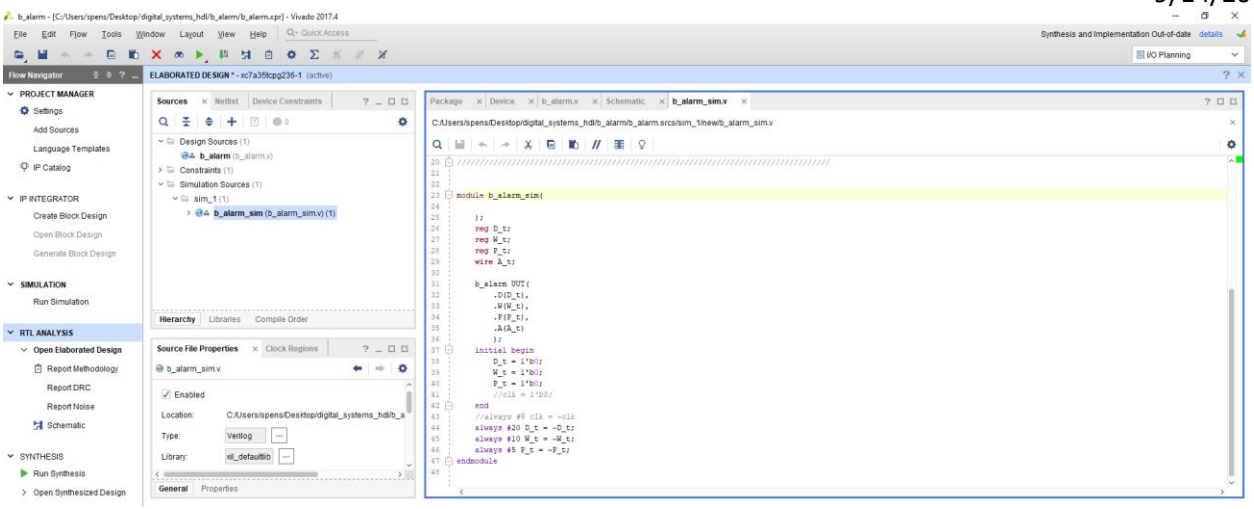
I then ran synthesis and implementation, to implement I/O Planning and Schematics. I then connected all the inputs and outputs to the I/O Planning.



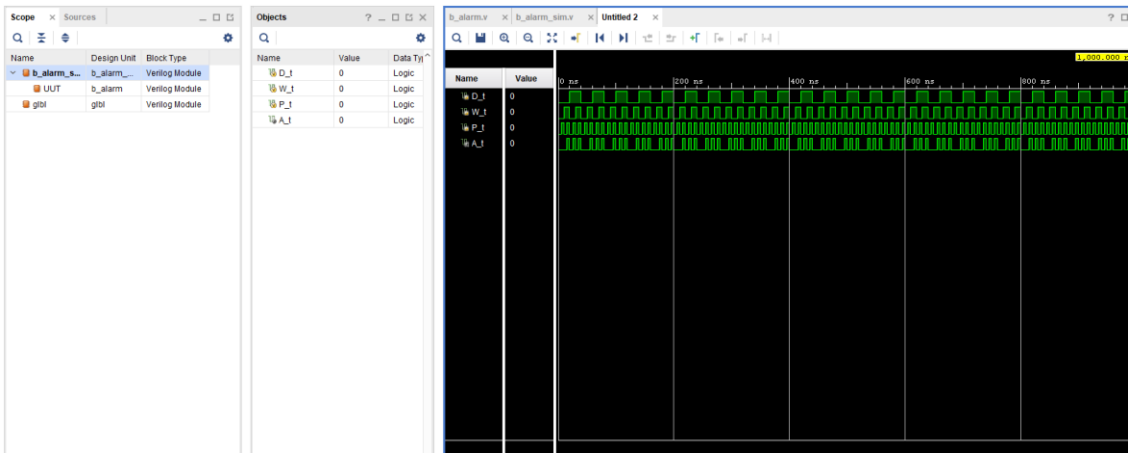
Then I generated the schematic of the burglar alarm controller circuit.

I then programmed the logic for the simulation.

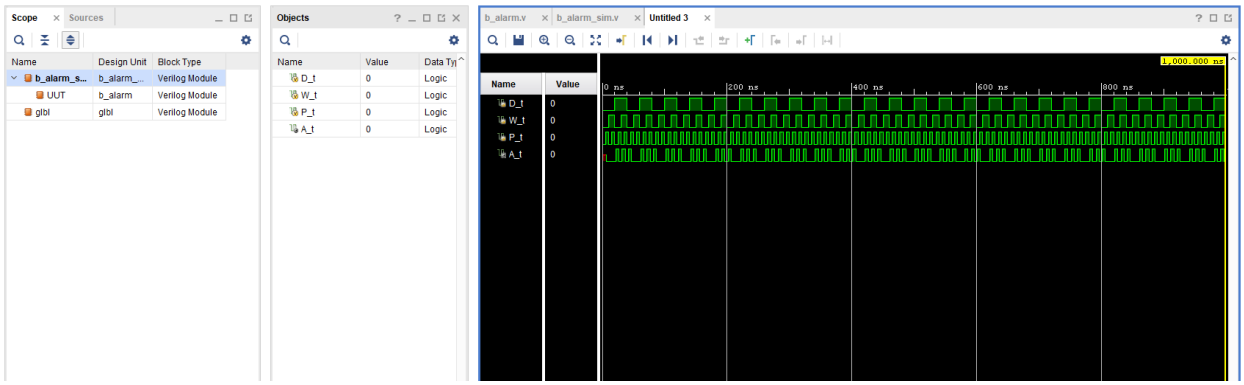




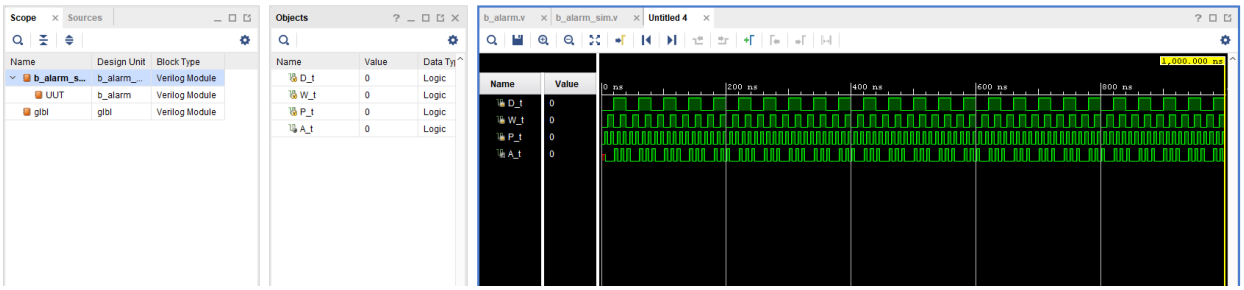
I then ran a behavioral, post-synthesis, and a post-implementation timing simulation.



(behavioral)



(post-synthesis)

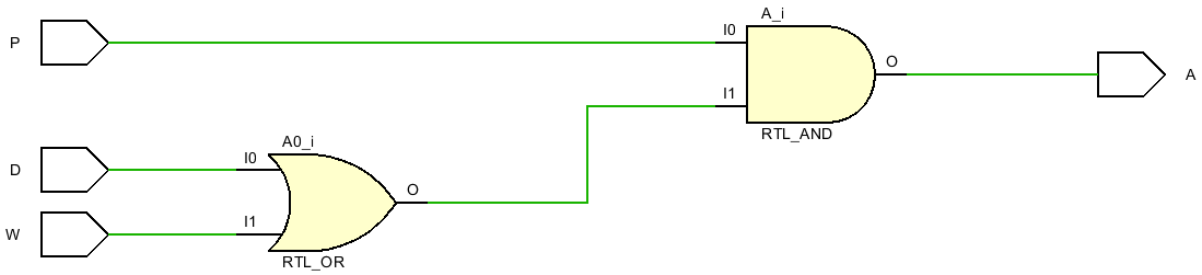


(post-implementation)

Burglar alarm controller circuit Schematic

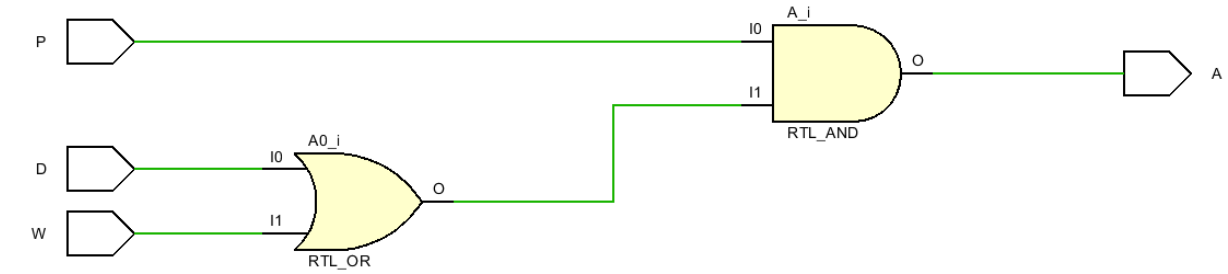
Logic Diagram:

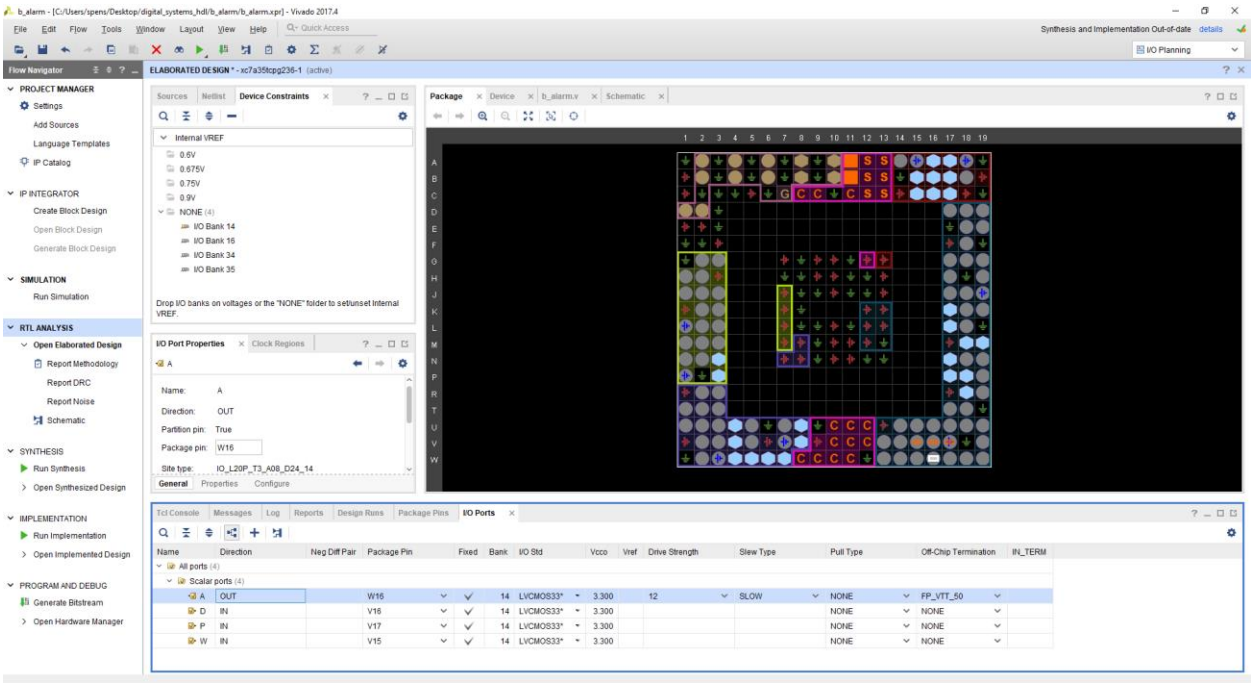
The resulting schematic diagram for the burglar alarm controller circuit gate.



Design Specification Plan:

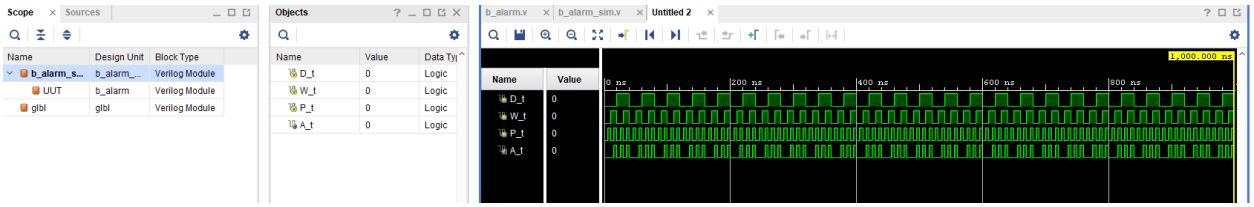
This is the design specification for a burglar alarm controller circuit gate. It has three inputs (P, W, D) and one output (A).



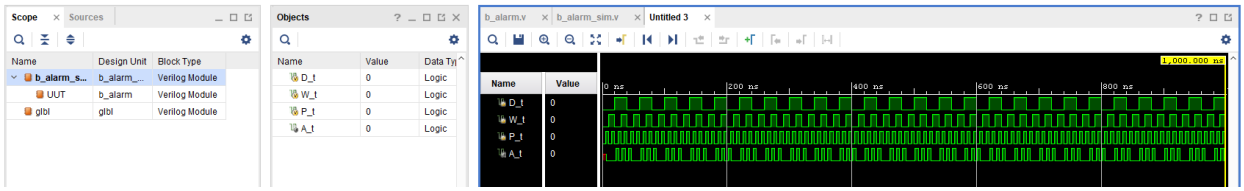


Results Statement:

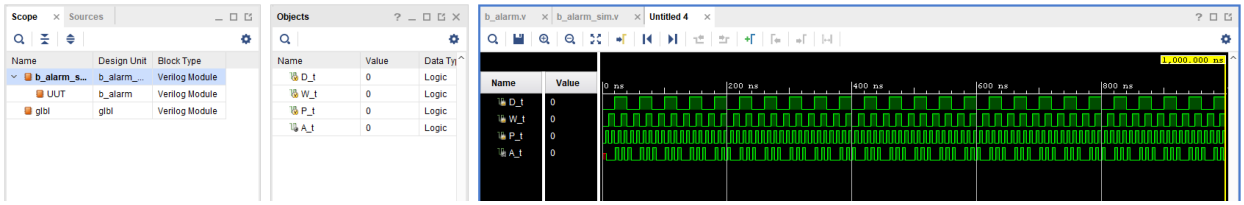
Here is the behavioral simulation results for the burglar alarm controller circuit_gate.



Here is the post-synthesis simulation results for the burglar alarm controller circuit_gate.



Here is the post-implementation simulation results for the burglar alarm controller circuit gate.

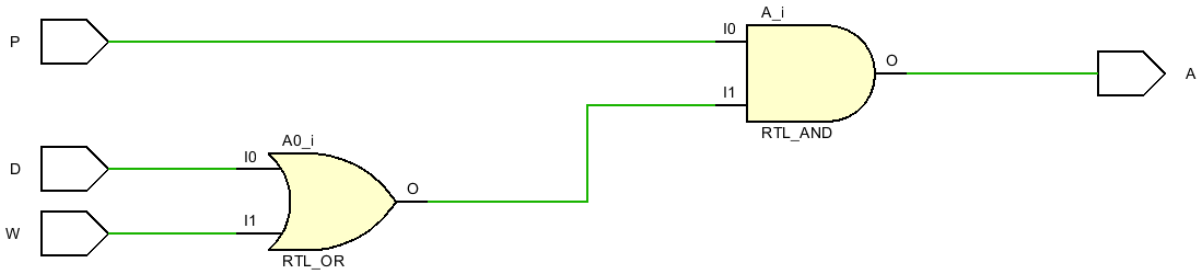


P	W	D	OUT
1	0	1	1
1	1	0	1
1	1	1	1
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

Burglar alarm controller circuit Verilog

Logic Diagram

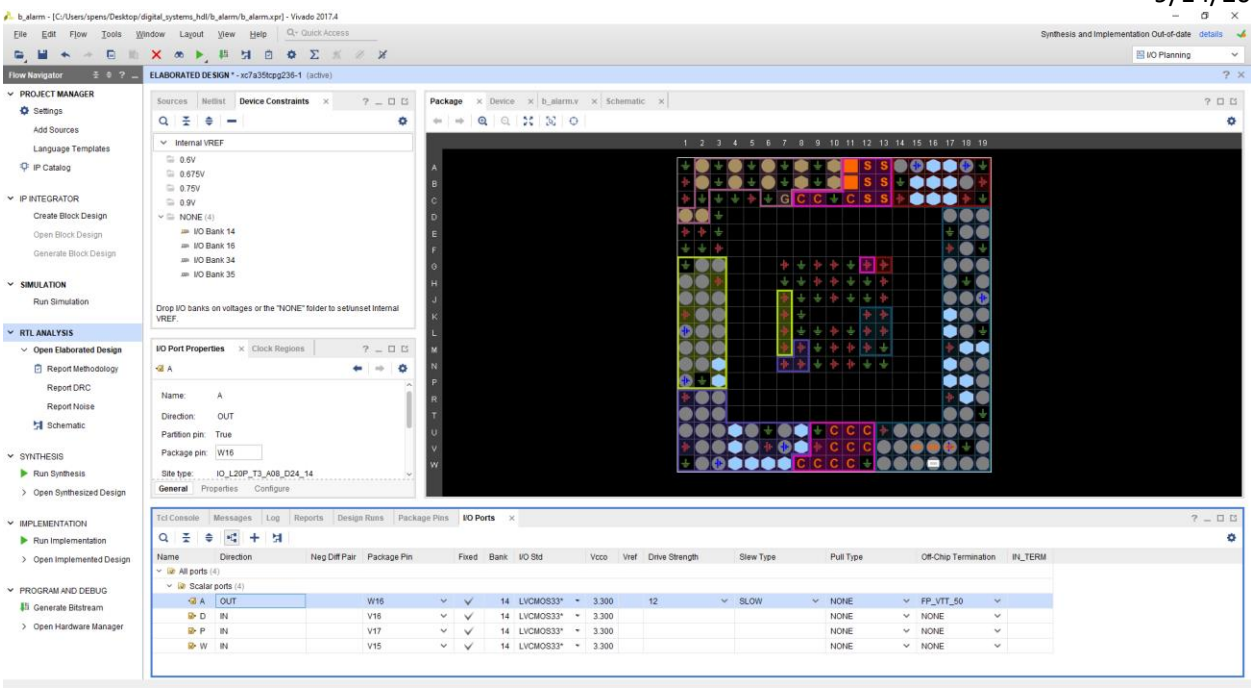
The results from the simulation testing for the simulation for a burglar alarm controller circuit.



Design Specification Plan

This is the design specification for a burglar alarm controller circuit. It has three inputs (P, W, D) and one output (A).

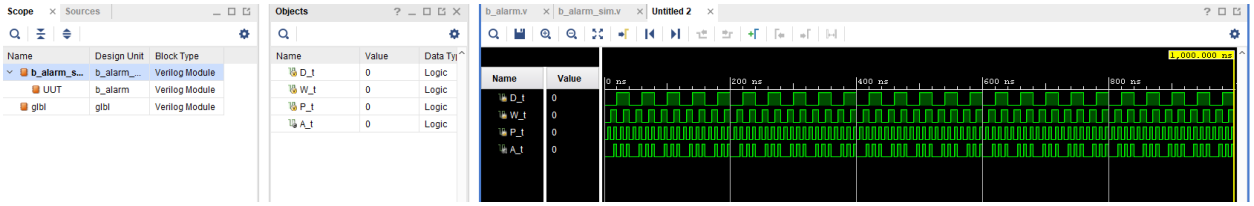
```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Company:
4  // Engineer:
5  //
6  // Create Date: 10/05/2020 01:16:50 PM
7  // Design Name:
8  // Module Name: b_alarm
9  // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////////////////////////////////////
21
22
23 module b_alarm(
24     input D,
25     input W,
26     input P,
27     output A
28     );
29     assign A = (P & (D | W));
30 endmodule
```



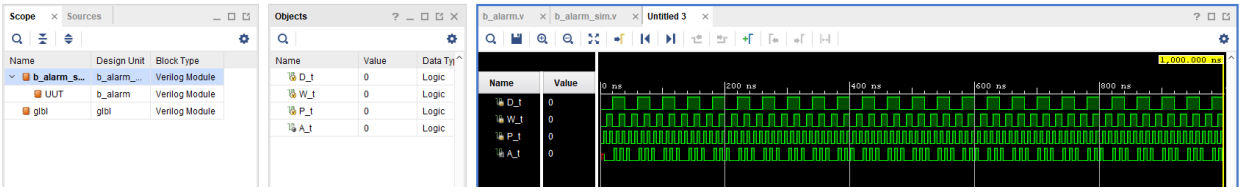
Results Statement

Explanation of pictures and tables.

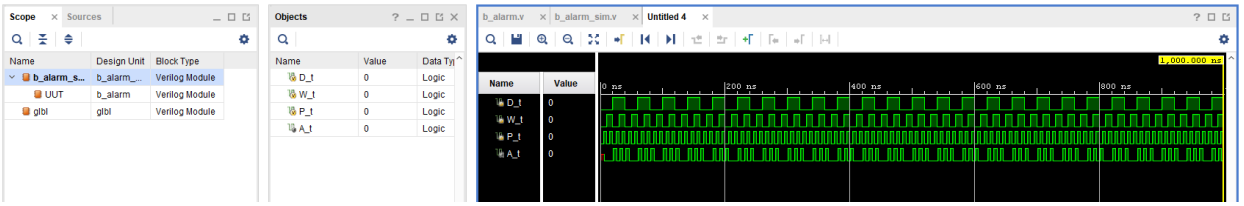
Here is the behavioral simulation results.



Here is the post-synthesis simulation results.



Here is the post-implementation simulation results.



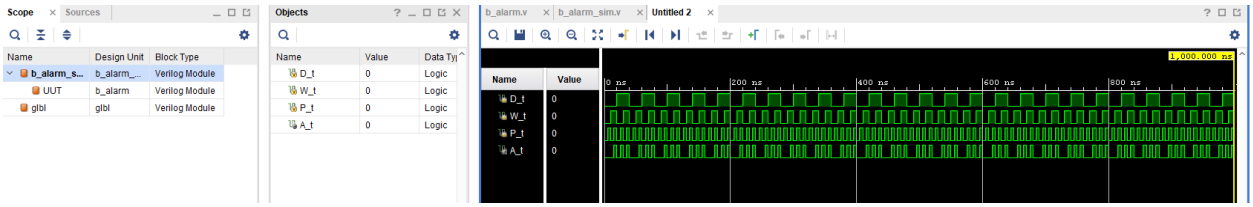
P	W	D	OUT
1	0	1	1
1	1	0	1
1	1	1	1
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

Test Plan

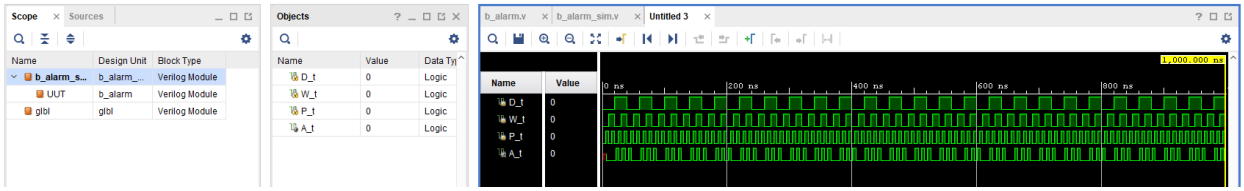
The initial design was tested through a simulation based off of the test bench, which was programmed for every possible input, which would give every output.

Burglar alarm controller circuit

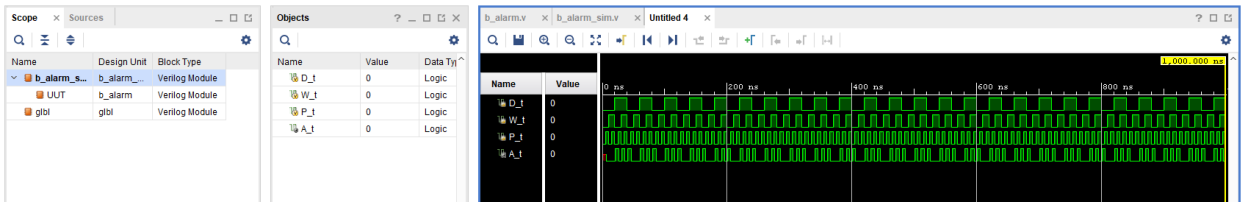
Here is the behavioral simulation results.



Here is the post-synthesis simulation results.



Here is the post-implementation simulation results.



Conclusion

My results were what I expected, and all the test and simulation results were correct and agreed with the logic that I had programmed. By running the gate through multiple simulations and then analyzing them was interesting, and lead to further understanding of the logic gates functionality of how it would work in the real world.

1. Can the logic be simplified in any way? Explain.

A = PW'D + PWD'+PWD

A = PD[W'+W] + PWD'

A = PD[1] + PWD'

A = P[D + WD']

A = P[D+W]

A = PD + PW

Yes.

2. How would the controller logic be simplified if the power is always on?

The power is always on in the last four columns. Since there are four rows in the results, it would be easier to write the function for the alarm using SOP

A = P' + D + W

The logic gate design would be one OR gate and an inverter for the power.

3. Have you met all the requirements of this lab (Design Specification Plan)?

I have all required conditions after simplification.

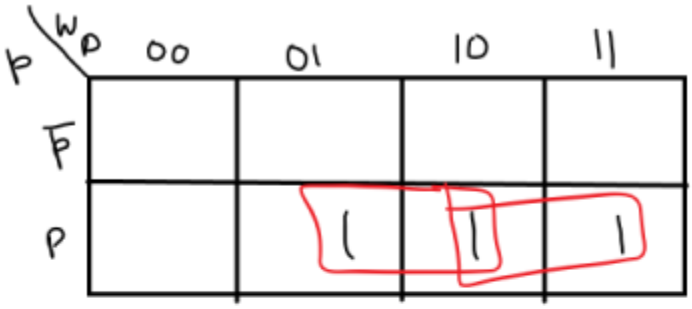
4. How should your design be tested (Test Plan)?

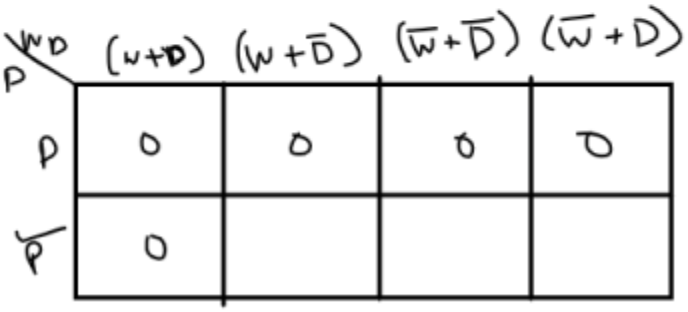
Start by giving input to the LED and testing which inputs turn it on and which ones do not. Then compare them with the results of the simulation.

5. The function is represented by the ORing of the terms associated with the 1's in the truth table. Can an expression for A be found which is derived by the ANDing of terms associated with the 0's in the truth table?

A = PW'D + PWD' + PWD

= 0 + 0 + 1 = 1





A = P(W+D)

6. Write the Boolean expression describing a burglar alarm, which also turns on an LED when an additional sensor has been crossed.

A = PD'W'S + PD'WS' + PD'WS + PDW'S' + PDW'S + PDW'S + PDWS

A = P(D'W'S + D'WS' + D'WS + DW'S' + DW'S + DW'S + DWS)

A = DWS

P	D	W	S	A
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

7. Discuss the differences, if any, between the behavioral, post-synthesis timing, and post-implementation timing simulations. Do they differ from the real behavior of the implemented circuit?

Considering all factors, physical and theoretical, then they most definitely differ from the real behavior of the implemented circuit. The post synthesis and the post implementation differ due to physical constraints of real life behavior, evidence of this can be seen in their corresponding simulation results with the red portion at the begging.