Yiğit Narter
22102718
EE102-01
Lab Report 2
03.10.2022

LAB 2: INTRODUCTION TO VHDL

Purpose

This experiment aims to design and implement a combinational circuit on the BASYS3 board. This design should be made using VHDL. For my part, I wanted to design a circuit based on a logic equation called “Consensus Theorem” which we’ve learned in class. With this circuit, I want to show that this theorem holds true by displaying the right and left hand side of the equation using two outputs of the BASYS-3 board.

Design Specifications

In this experiment, I designed the circuit using the VHDL programming language. Also, to implement this circuit on the BASYS3 board, I used Vivado Suite Design 2018.3 Webpack. There will be three inputs in this implementation, namely X, Y and Z, which will have binary values (0 represents False, and 1 represents True). As for outputs, there will be two. The output R represents the right hand side of the equation, and the output L represents the left hand side. On the Basys-3 FGPA board, the switches will be used as inputs and the LEDs as outputs. For inputs, the switch V17 will correspond to X, the switch V16 will correspond to Y and the switch W16 will correspond to Z. For outputs, the LED U16 will represent R and the LED E19 will represent L.

Methodology

The circuit I want to design will be an implementation of the Consensus Theorem (also known as Redundancy Theorem), which is an equation used as a shortcut in digital logic. The equation can be expressed using three variables X,Y and Z, and these variables will be represented in the circuit as three separate inputs:

$$X.Y + Y.Z + \overline{X}.Z = X.Y + \overline{X}.Z$$

The conditions for this theorem is that there should be three variables, with each of them repeated twice in the equation and one them being present with its complement form. It is observed that the term in the middle disappears. We can prove this theorem by using a truth table, which can be seen in Figure 1. The function L represents the left hand side of the equation ($L = X.Y + Y.Z + \overline{X}.Z$), and the function R corresponds to the right hand side ($R = X.Y + \overline{X}.Z$). These functions will also be present in the circuits as outputs.

X	Y	Z	R	L
0	0	0	0	0
0	0	1	1	1

Yiğit Narter
22102718
EE102-01
Lab Report 2
03.10.2022

0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	1
1	1	1	1	1

Figure 1: Truth table of the functions

As seen from the table the functions L and R are equivalent, meaning that the equation holds true for all possible X, Y and Z values. As a result of this, in the circuit we expect the outputs L and R behave the same as the inputs X, Y, Z change.

In VHDL, the inputs and outputs are defined in the part called “entity”. The inputs and outputs normally aren’t provided in VHDL, so we have to use a common code called “library”. The library, which is named IEEE in this case, must be stated before architectures that involve inputs and outputs. For this reason, I added the necessary statement to use the library IEEE, and defined the inputs and outputs as STD_LOGIC forms. The port map in the code is where I state the local signals that the inputs and outputs will be connected to. It makes connections between the ports and the architecture part where the signals are located.

The restrictions related to placement and timing on the board are specified in the constraint file. The inputs and outputs are also controlled using it. In Vivado, an XDC file (Xilinx Design Constraints file) is what is used as the constraint file. Using a constraint file, the inputs and outputs on a hardware interface can be connected to the pins on the FGPA board.

In order to simulate the circuit I’ve designed, it is necessary to use a test bench code. This can be described as generating input waveforms and observing the waveforms created by the simulation tool of Xilinx ISE. After observing the output waveform created by the test bench code, the circuit will be implemented on the BASYS-3 board.

Results

First, I created a design source for the circuit and observed its elaborated design which is shown in Figure 2. After that, I wrote the test bench code and simulated the inputs. The output waveform is seen in Figure 3. In the test bench code, it is seen that the values of R and L are the same for all the possible values of X, Y and Z.

Yiğit Narter
22102718
EE102-01
Lab Report 2
03.10.2022

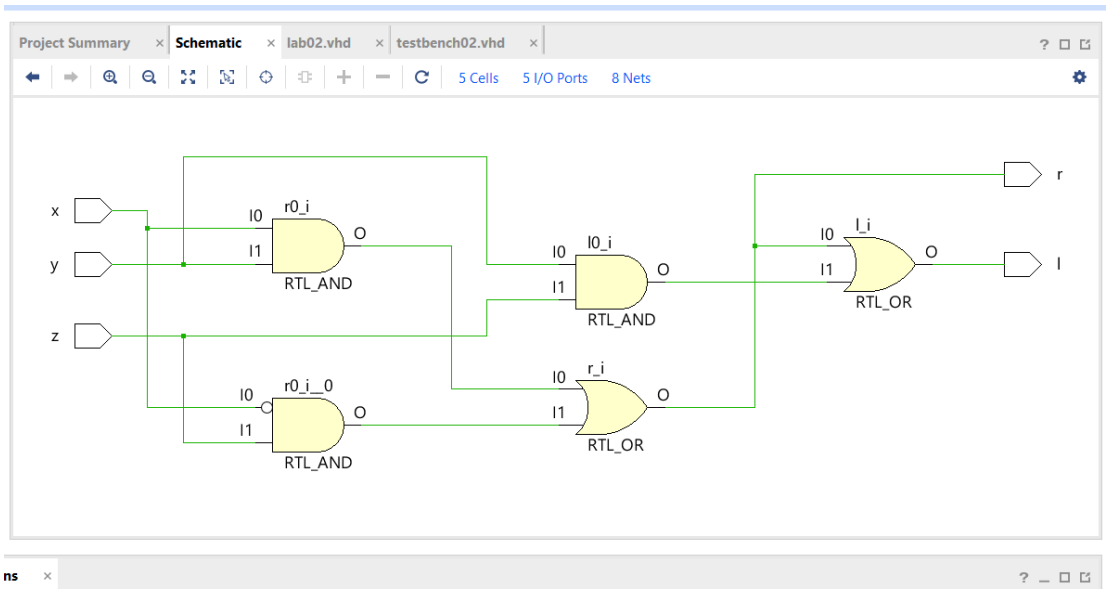

Figure 2: Elaborated design of the circuit

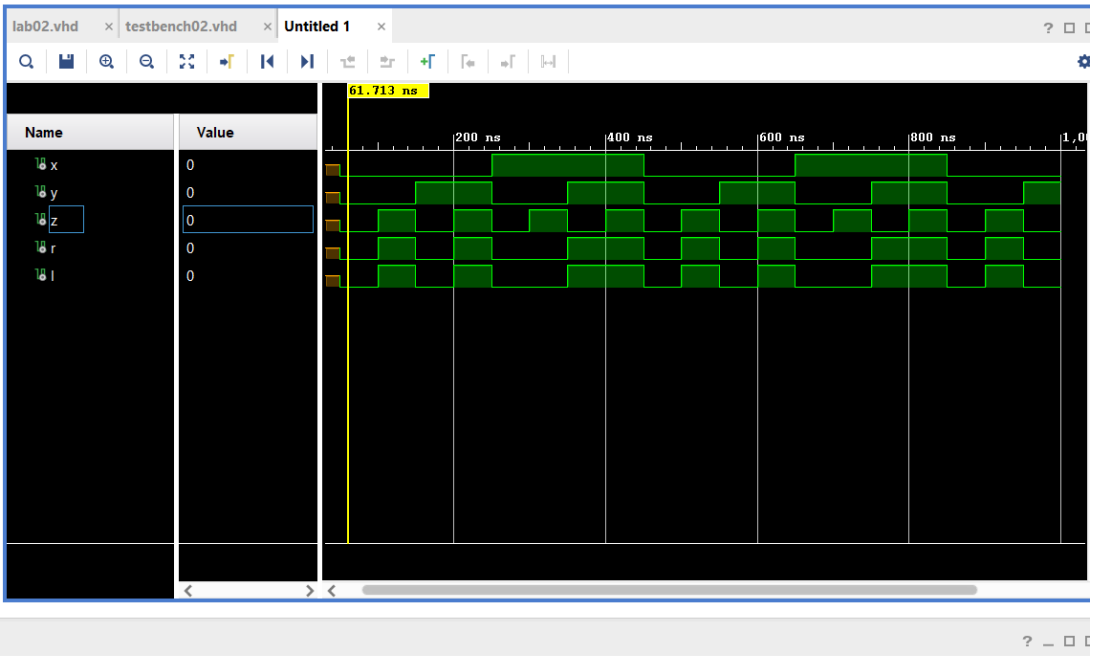

Figure 3: Test bench code run on the simulator

Yiğit Narter
22102718
EE102-01
Lab Report 2
03.10.2022

Figures below show the outputs on the BASYS-3 board. The values of the inputs X,Y and Z are specified by adjusting the switches. The switches beginning from the right correspond to X,Y,Z.

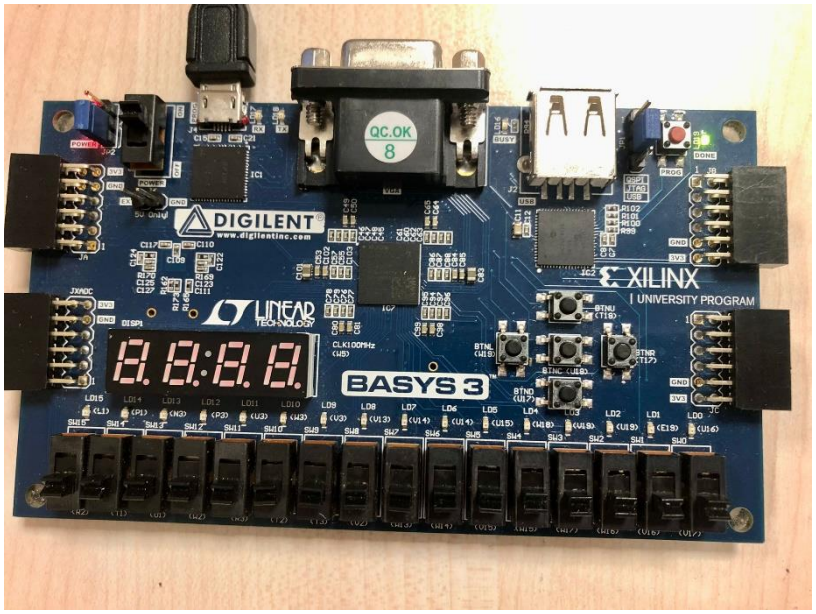

Figure 4: Outputs R=L=0 when X=0, Y=0, Z=0

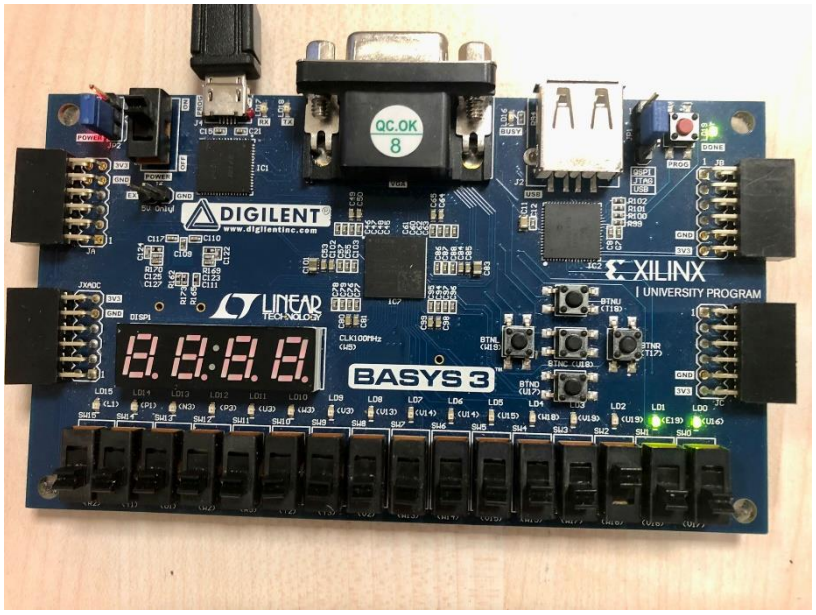

Figure 5: Outputs R=L=1 when X=0, Y=0, Z=1

Yiğit Narter
22102718
EE102-01
Lab Report 2
03.10.2022

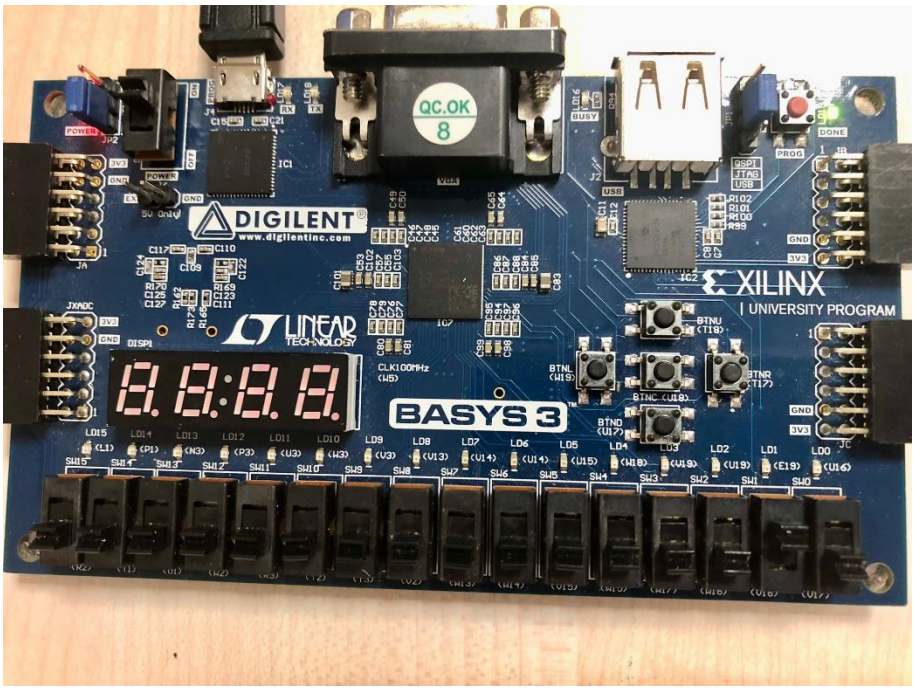

Figure 6: Outputs R=L=0 when X=0, Y=1, Z=0

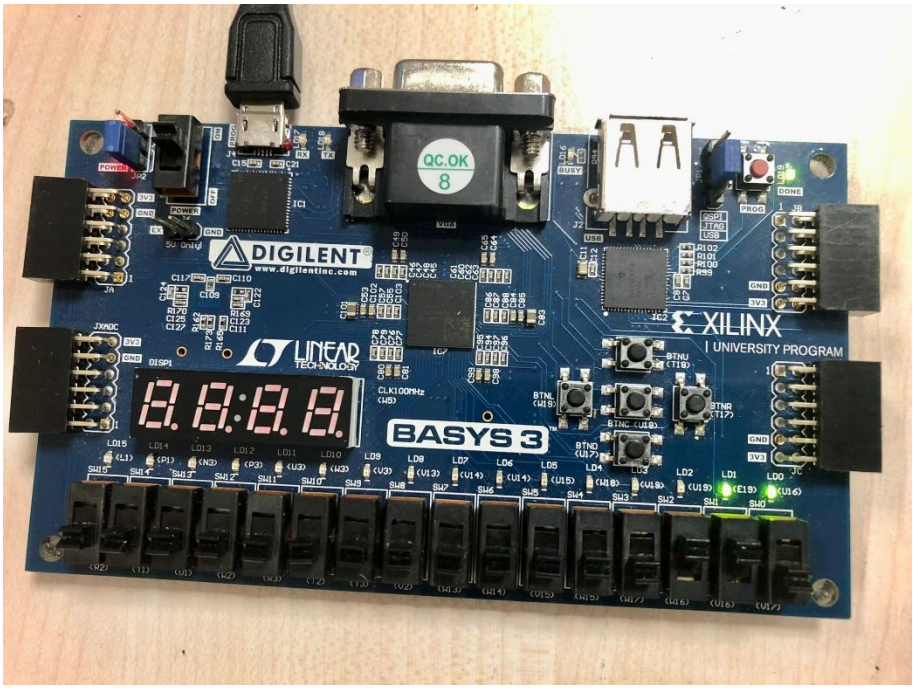

Figure 7: Outputs R=L=1 when X=0, Y=1, Z=1

Yiğit Narter
22102718
EE102-01
Lab Report 2
03.10.2022

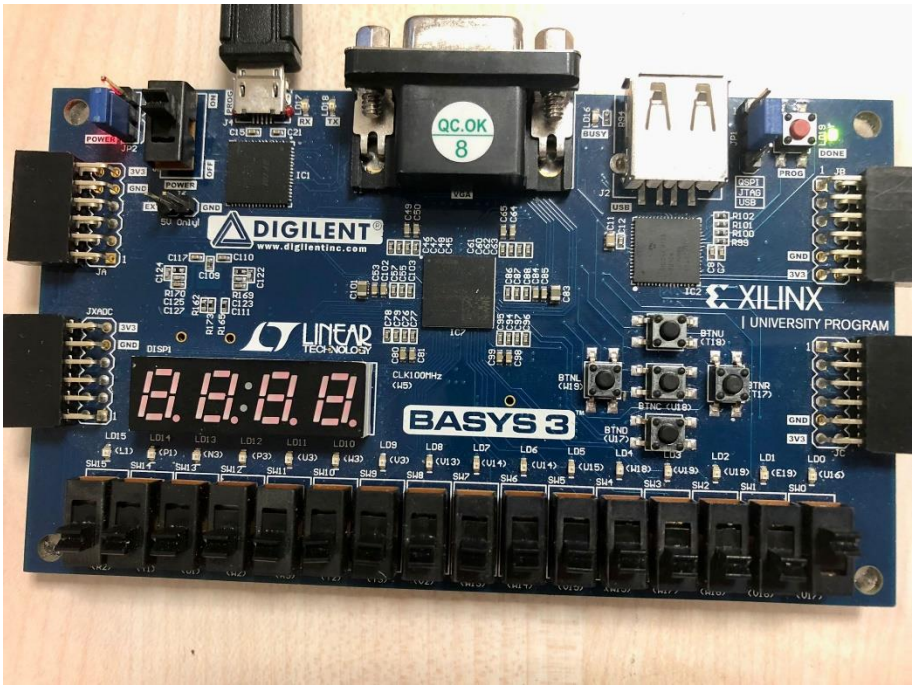

Figure 8: Outputs R=L=0 when X=1, Y=0, Z=0

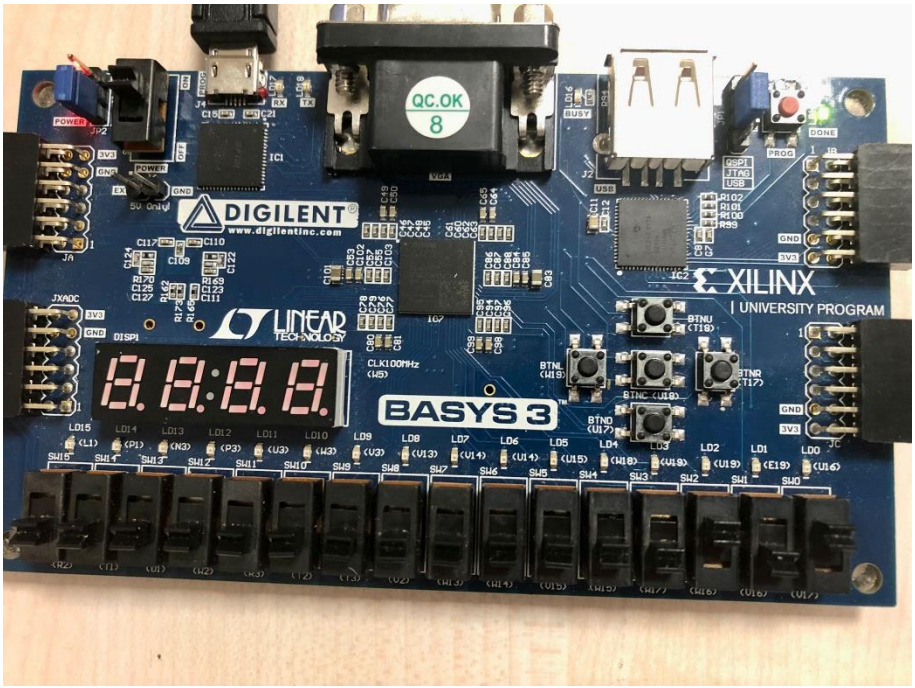

Figure 9: Outputs R=L=0 when X=1, Y=0, Z=1

Yiğit Narter
22102718
EE102-01
Lab Report 2
03.10.2022

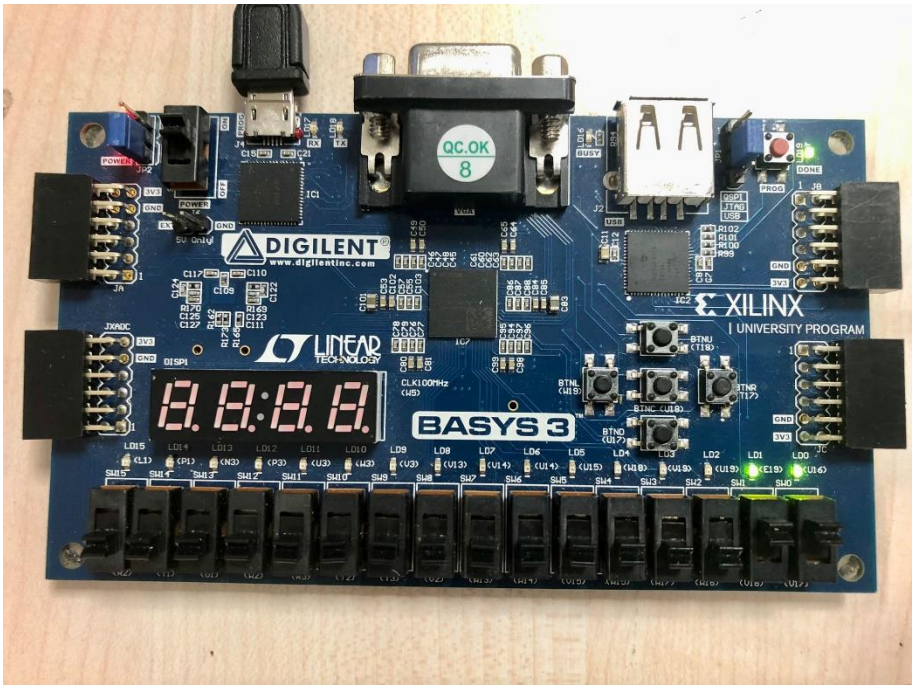

Figure 10: Outputs R=L=1 when X=1, Y=1, Z=0

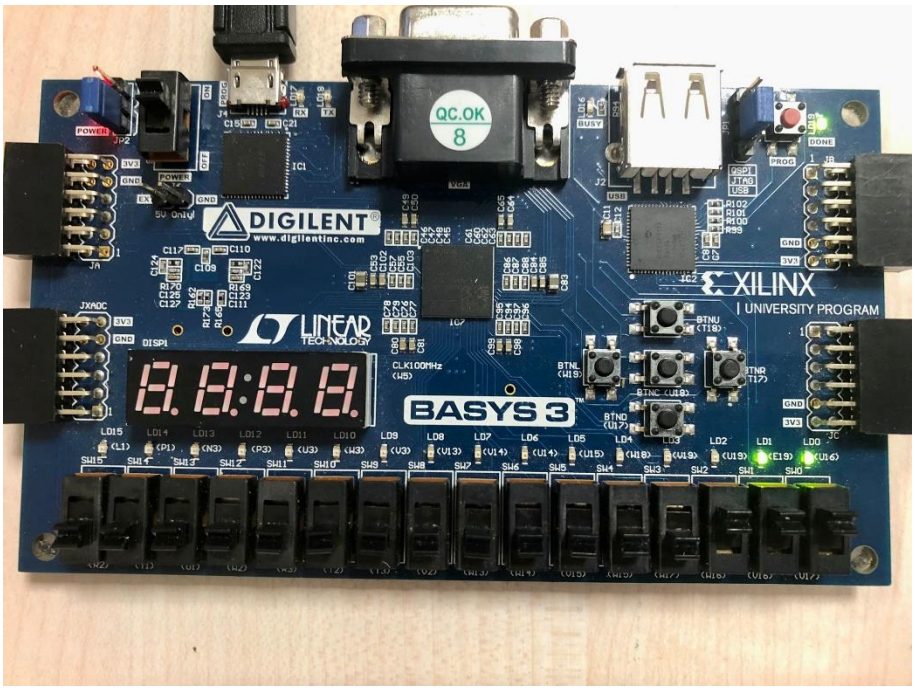

Figure 11: Outputs R=L=0 when X=1, Y=1, Z=1

As expected, the LEDs corresponding to the outputs R and L lighten simultaneously for all values of X, Y and Z. It was already shown that the functions R and L are equivalent using a truth table. This design also proves this as both outputs' behaviors (whether the corresponding LEDs are on or off) are the same for all values of X, Y and Z.

Conclusion

The experiment's aim was to design and implement a combinational circuit using Vivado and VHDL. I was able to achieve this by designing a logic equation that we have learned in the course. Even though the circuit included only 3 inputs and 2 outputs, it still had a complex algorithm behind it. Creating the truth table for the function and writing test bench codes was quite helpful for implementing the circuit. The results turned out as expected, showing that the equation holds true. Even though there is still much to learn about VHDL and Vivado, I can say that this experiment was beneficial for me because I had an idea of how these systems work.

Appendices

Design Code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity lab02 is
   Port ( x : in STD_LOGIC;
        y : in STD_LOGIC;
        z : in STD_LOGIC;
        r : out STD_LOGIC;
        l : out STD_LOGIC);
end lab02;


architecture Behavioral of lab02 is
begin
r <= (x and y) or ((not x) and z);
```

```
l <= (x and y) or ((not x) and z) or (y and z);

end Behavioral;
```

Test Bench Code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity testbench02 is
end testbench02;

architecture Behavioral of testbench02 is
COMPONENT lab02
 PORT( x : IN STD_LOGIC;
 y : IN STD_LOGIC;
 z : IN STD_LOGIC;
 r : OUT STD_LOGIC;
 l : OUT STD_LOGIC);
 END COMPONENT;
 SIGNAL x : STD_LOGIC;
 SIGNAL y : STD_LOGIC;
 SIGNAL z : STD_LOGIC;
 SIGNAL r : STD_LOGIC;
 SIGNAL l : STD_LOGIC;
BEGIN
 UUT: lab02 PORT MAP(
x => x,
y => y,
z => z,
```

```
r => r,
l => l
 );
 testbench02 : PROCESS
 BEGIN
 wait for 50 ns;
 x <='0';
 y <='0';
 z <='0';
 wait for 50 ns;
 x <='0';
 y <='0';
 z <='1';
 wait for 50 ns;
 x <='0';
 y <='1';
 z <='0';
 wait for 50 ns;
 x <='0';
 y <='1';
 z <='1';
 wait for 50 ns;
 x <='1';
 y <='0';
 z <='0';
 wait for 50 ns;
 x <='1';
 y <='0';
 z <='1';
```

```
 wait for 50 ns;
 x <='1';
 y <='1';
 z <='0';
 wait for 50 ns;
 x <='1';
 y <='1';
 z <='1';
 END PROCESS;
end Behavioral;
```

References

Test Bench Tutorial.pdf

BASYS 3 - Vivado Tutorial.pdf

https://vhdlwhiz.com/port-map/