Lab Report 2: Introduction to VHDL

EE 102

Section 02

Bilkent University

February 21st, 2024

Doğa Zeynep Tarman

22303157

Purpose of the Experiment

The purpose of this laboratory assignment was to introduce us students to the concept of VHDL. Using a Basys 3, we were expected to test and debug certain circuits using VHDL code.

Methodology

I started the laboratory by opening the folder that was sent to us from Moodle. In this file, there were all necessary files that we were going to debug and test. After launching the app, I started to work on the tasks in order. All the codes I have written or have changed are available in the Appendix part of this report, with changed parts highlighted.

Firstly, I opened the code named “submodule1.vhd”, and observed the values in lines 14, 15 and 16 respectfully. I used the table given in the laboratory document and changed those values according to my school ID number, which ends with a 7. Afterwards, I moved on to the second task, which asked us to fix 6 bugs in the circuit we were given. I took screenshots of the error messages and copied the code that I have changed to the Appendix part of this report. Next, I synthesized the code and tried demonstrating this on my Basys 3. I checked which switch corresponds to what with some trial and error. I then moved on to the fourth task, where I was asked to write a test bench code to show the inputs in an order and display the results of this in waveform. I tried writing a code to complete the task and used the switches to get the results that were asked for in the laboratory document. I observed the waveform and after making sure everything was correct by getting a check from the TA, I moved on to the next task. In this task, we were asked to observe different schematics and compare these. I drew the necessary schematics and tried to understand the differences between them. Lastly, I made sure I included all the code that I have modified or wrote myself in this report for the sixth task.

- In VHDL, we specify the inputs and outputs by using ports. In these ports, we use the keywords “IN” and “OUT” to indicate which one is the input, and which is the output in a module.
- To use a module in another module, we use port maps. A port map is a tool that helps us divide complex VHDL codes into smaller parts and combine them afterwards (*VHDL component and Port Map tutorial*). This gives us users ease of implementing the code.
- A constraints file is a file that specifies which inputs will be represented by which switch and which outputs will be represented by which LED (Colvin, *What Is a Constraints File? - Digilent Reference*), in our case.

- The main purpose of creating a test bench is to enable the program to run a simulation.

Results

Task 1:

For this task, we were asked to modify the code named “submodule1.vhd”. Specifically, we were asked to change the values on lines 14, 15 and 16 according to the last digit of our ID number. Mine was 7, therefore I changed the values to “OR OR XOR” as specified in the laboratory document. The modified code can be found in the Appendix, named Code 1: submodule1.vhd.

Task 2:

In this task, we were asked to spot and fix six different bugs within the laboratory code that was assigned to us. The first bug that I faced was after trying to synthesize the code.

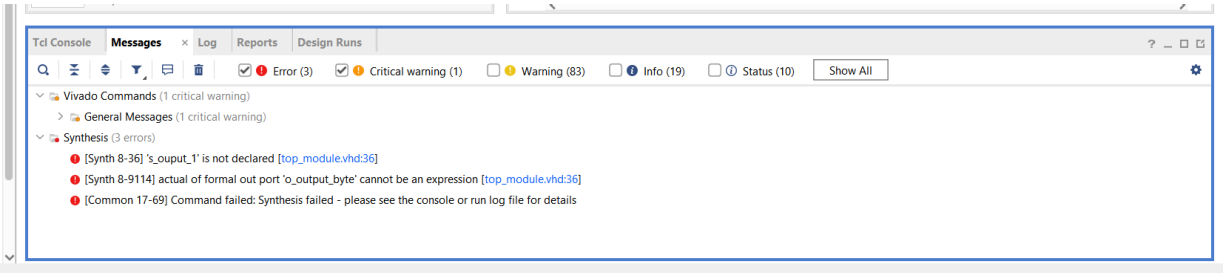

Figure 1.1 First Bug

It appears that on line 36, there was a word that was misspelled, and was missing a letter “t”. After adding the missing letter, the problem was resolved. The corrected code can be found in the Appendix, named Code 2. The second bug was about definition functions not being clear.

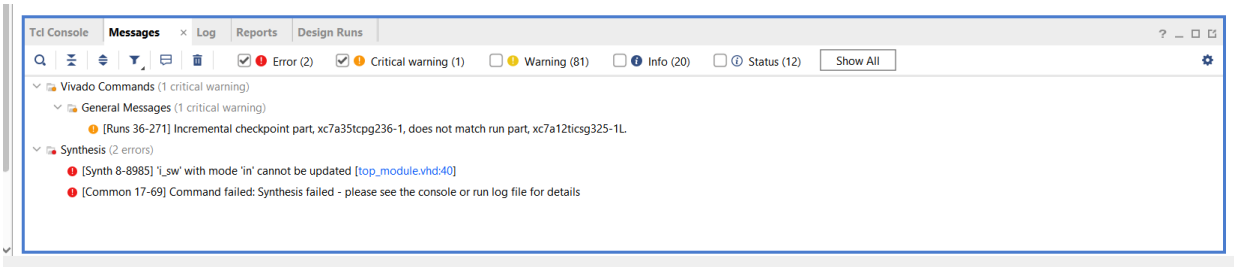

Figure 1.2 Second Bug

After defining the properties appropriately, the bug was fixed. The code can be seen in the Appendix named Code 3. The third bug was about a misnaming of variables, where some had “_the_beast” added to them, which caused a bug.

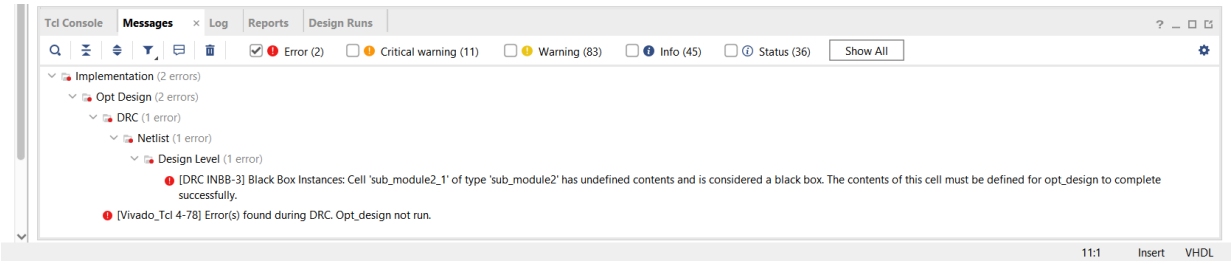

Figure 1.3 Third Bug

After removing those “the_beast” additions, this bug was successfully resolved as well. The other bug was about the project device being selected wrong.

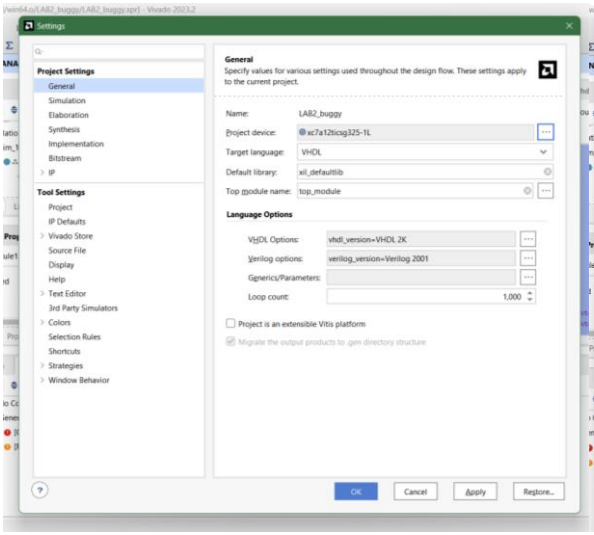

Figure 1.4 Fourth Bug (Unfixed)

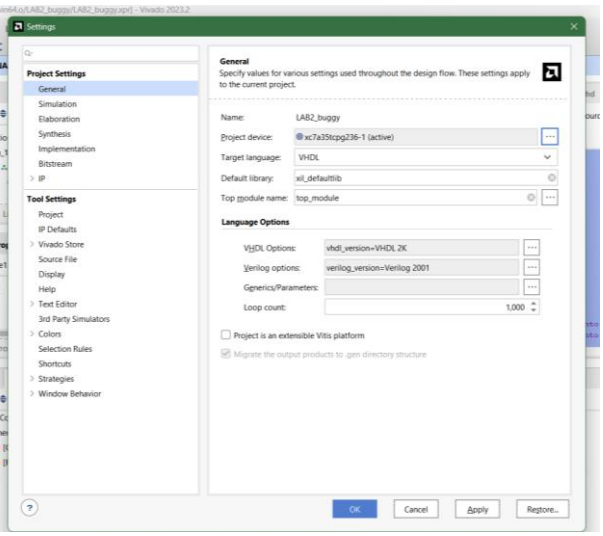

Figure 1.5 Fourth Bug (Fixed)

After picking the right device, this too was resolved. The next two bugs were about LED ports missing a defined value to them. To be more specific, some ports were named incorrectly.

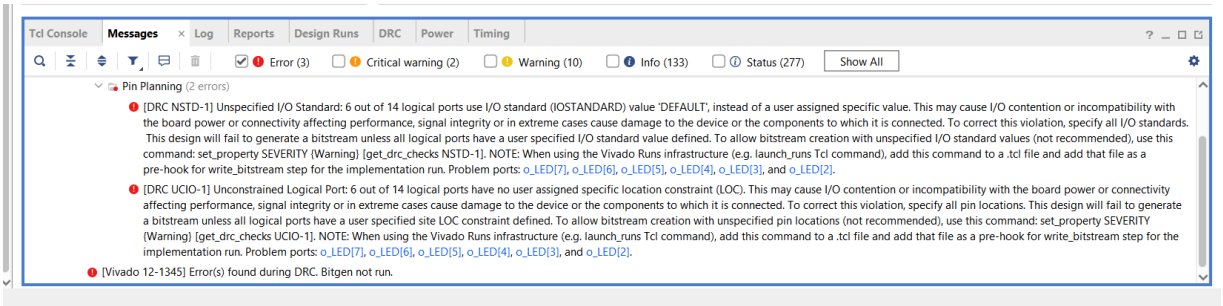

Figure 1.6 Fifth and Sixth Bugs

After renaming these to their correct values, I was also able to fix this bug. The altered code can be found in the Appendix named Code 4.

Task 3:

For the third task, we were asked to demonstrate the circuit on our Basys 3 FPGA. For this, I have programmed my device and took photos of my FPGA which was working correctly. However, I did this task after the fourth one and then observed the results.

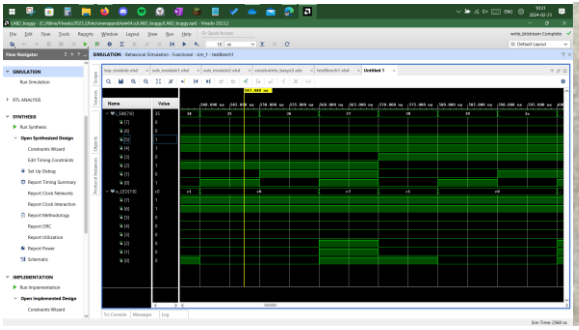

Figure 2.1 Waveform 1

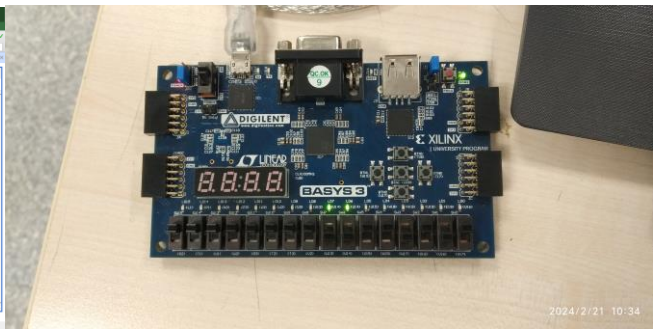

Figure 2.2 Representation 1

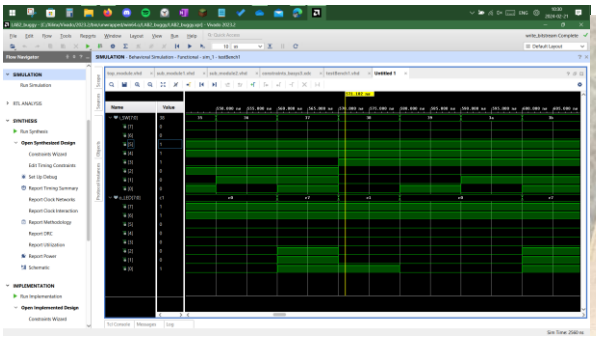

Figure 2.3 Waveform 2

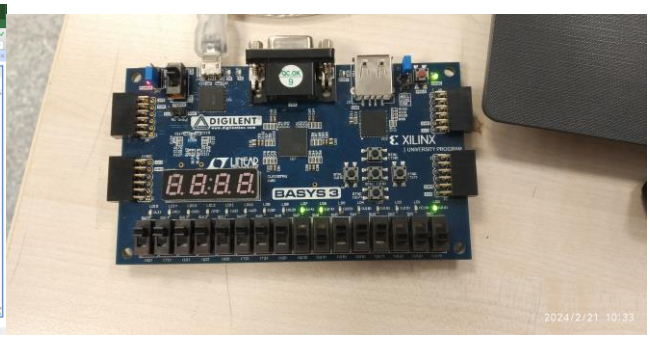

Figure 2.4 Representation 2

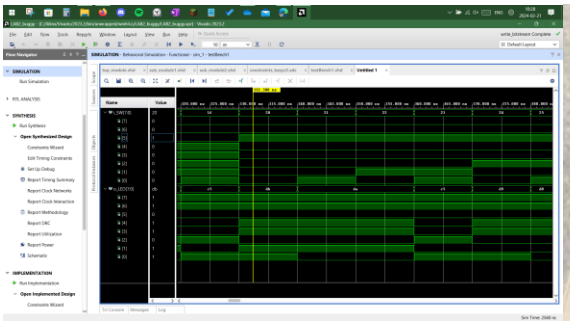

Figure 2.5 Waveform 3

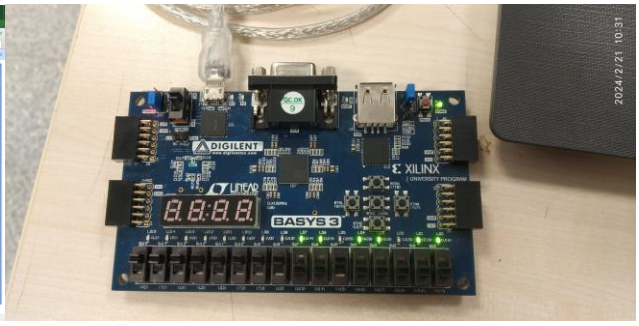

Figure 2.6 Representation 3

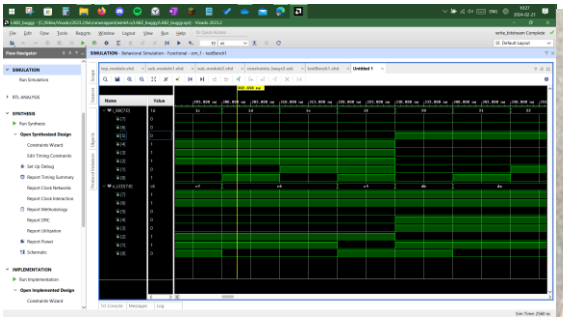

Figure 2.7 Waveform 4

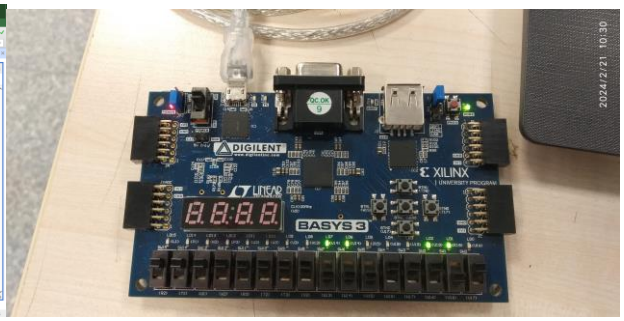

Figure 2.8 Representation 4

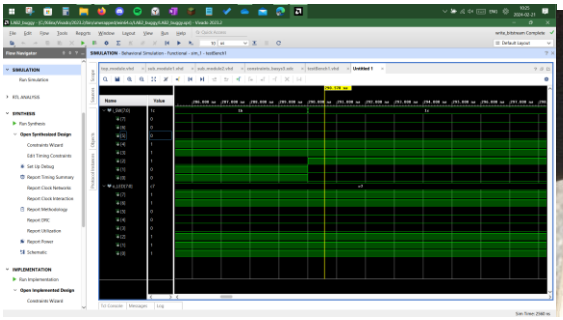

Figure 2.9 Waveform 5

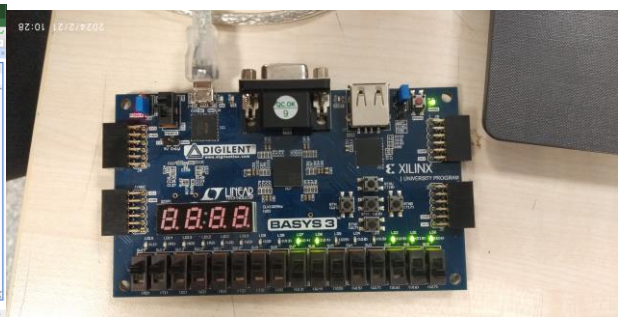

Figure 2.10 Representation 5

Task 4:

For this task, I tried writing a testbench code which I named testBench1 and got my TA's approval that my code was working as intended.

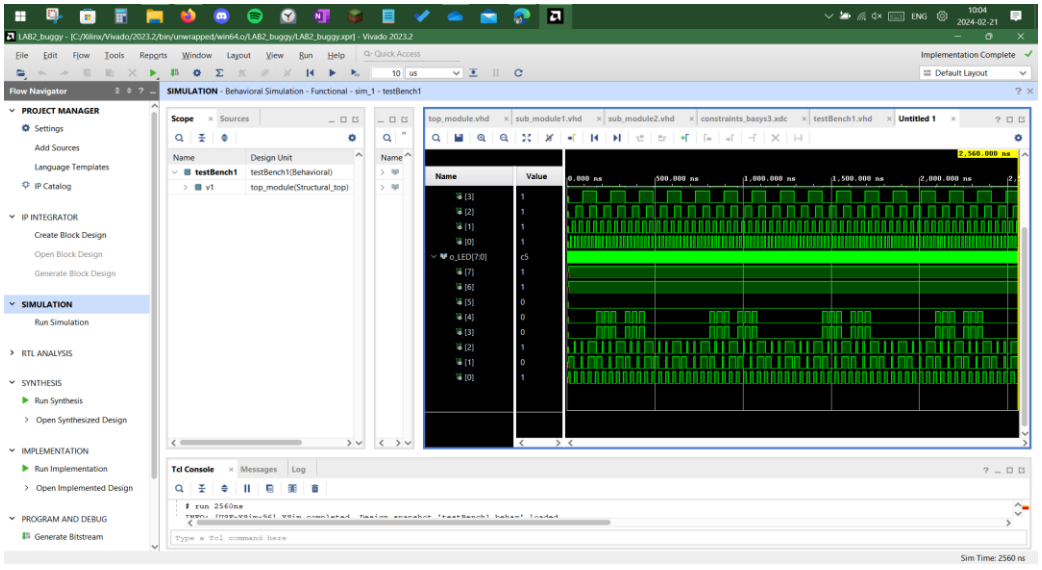

Figure 3.1 8 Inputs and All 256 Outputs

Task 5:

For this task, I drew three schematics and tried comparing them.

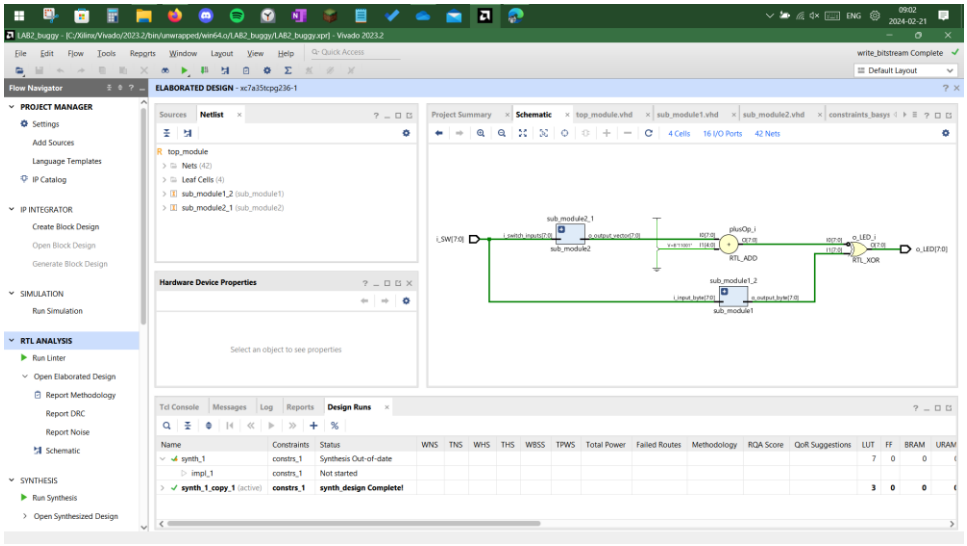

Figure 4.1 RTL Analysis

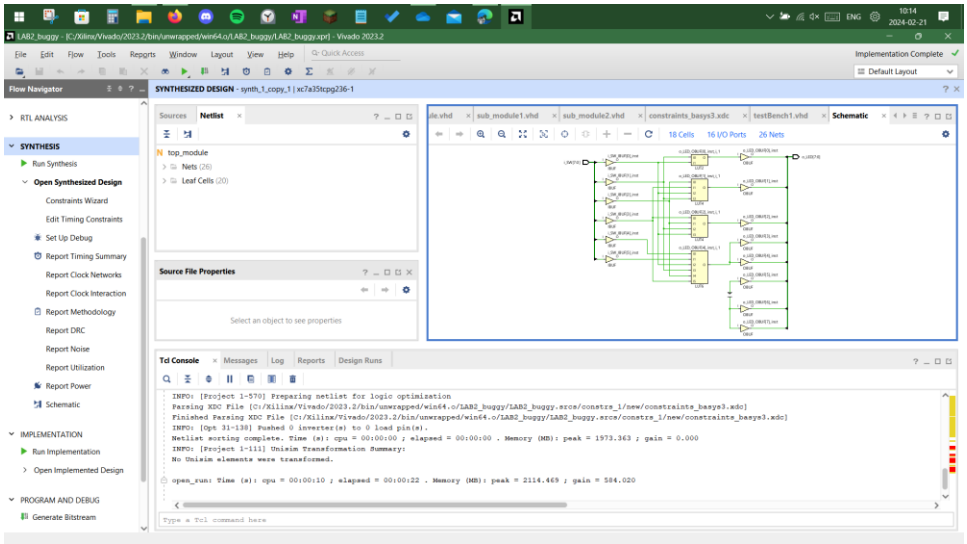

Figure 4.2 Synthesized Design

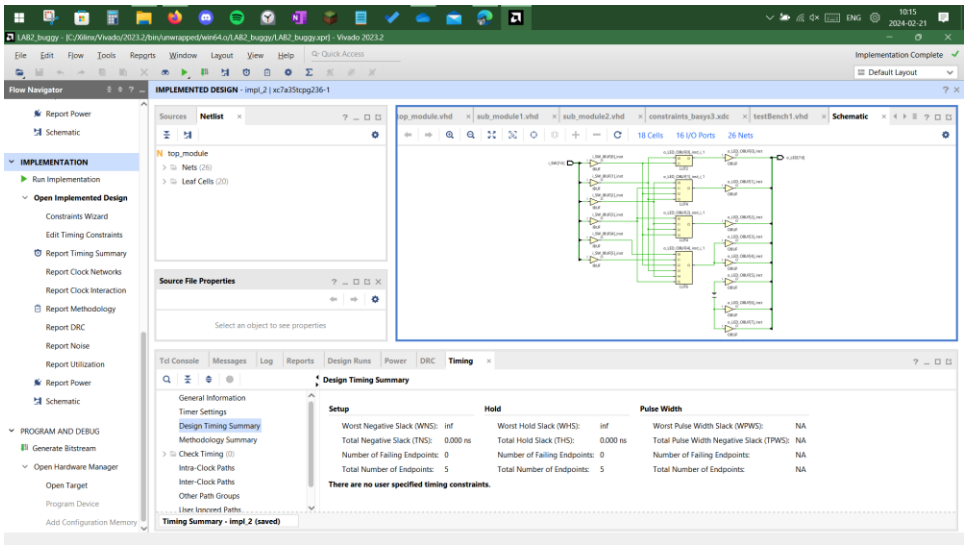

Figure 4.3 Implemented Design

By looking at these designs, I observed that the implemented design and the synthesized designs were the same looking, but RTL analysis was different. This is due to the RTL analysis one displaying logic circuits. The other designs, just like the RTL analysis one, support the simulation results as well.

Conclusion

By completing this laboratory, I was able to get a better grasp on VHDL and Basys 3. Some parts were much more challenging than others. To give an example, the first task took only a few minutes to complete for me as it was easy since the laboratory document explicitly stated which lines we were supposed to be looking, while also including which functions we were going to be using in a table. However, the third and fourth tasks consumed much more time than I had expected due to me misunderstanding certain parts and not being able to spot which part of the code I was supposed to change or rewrite unlike the first task. The reason why fixing the bugs was quite challenging at times was due to me overlooking small details such as a single letter or a parathesis being missing or not realizing that certain things were named differently than they should have been. However, I was able to overcome all these difficulties by carefully inspecting the code repeatedly and switching between tasks when I felt like I was not making any progress. Overall, the errors and time-consuming mistakes were caused by me due to my lack of knowledge on certain aspects of writing a code using VHDL. However, I believe that this laboratory was beneficial for me as it has taught me how to handle files and errors using Vivado. It has also made me able to get more familiar with VHDL, thus, I believe that I will not be spending too much time on the mistakes caused by my limited knowledge of the topic in the next laboratory.

Appendix

Code 1: sub_module.vhd

```
o_output_byte(0)        <= i_input_byte(0) or i_input_byte(1);
o_output_byte(1)        <= i_input_byte(2) or i_input_byte(3);
o_output_byte(2)        <= i_input_byte(4) xor i_input_byte(5);
```

Code 2:

```
    sub_module2_1 : sub_module2
        port map (
                i_switch_inputs  => i_SW,
                o_output_vector => s_output_2
            );
```

Code 3:

```
entity sub_module2 is
    Port (
        i_switch_inputs : in  STD_LOGIC_VECTOR (7 downto 0);
        o_output_vector : out STD_LOGIC_VECTOR (7 downto 0)
    );
end sub_module2;
```

Code 4:

```
# LEDs
set_property PACKAGE_PIN U16 [get_ports {o_LED[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[2]}]
set_property PACKAGE_PIN V14 [get_ports {o_LED[7]}]
```

Code 5 (testBench1):

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity testBench1 is
end testBench1;

architecture Behavioral of testBench1 is
    component top_module
    Port(
    i_SW : in std_logic_vector(7 downto 0);
    o_LED : out std_logic_vector(7 downto 0)
    );
    end component;
    signal i_SW : std_logic_vector(7 downto 0);
    signal o_LED : std_logic_vector(7 downto 0);

begin
v1: top_module port map(i_SW => i_SW, o_LED => o_LED);
    stimilus: process
    begin
    for i in 0 to 255 loop
        wait for 10 ns;
        i_SW <= std_logic_vector(to_unsigned(i, 8));
        end loop;
    wait;
    end process;
end Behavioural;
```

Works Cited

Team, Support, et al. “VHDL Component and Port Map Tutorial.” *Invent Logics*, 10 Jan. 2018, allaboutfpga.com/vhdl-component-port-map-tutorial/

What Is a Constraints File? - Digilent Reference. 17 Nov. 2021, digilent.com/reference/programmable-logic/guides/vivado-xdc-file.