Purpose

The purpose of this lab session is to debug and test a combinational circuit on Basys 3 on Vivado using the language VHDL.

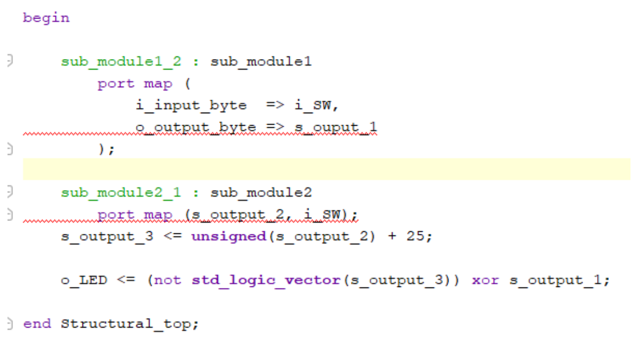
Questions

1- In every design on VHDL, there has to be an entity and at least one architecture section. Through the entity section, the users decides the amount and names of these, and chooses the connections and design between these input and outputs from the architecture.
2- The entity sections lets modules communicate with the outside. Every can be implemented into other modules by the PORT MAPs by choosing the input and outputs, and can be used multiple times.
3- After completing the design and simulating it, one can implement the design on Basys 3 using a constraint file to use which parts of the FPGA to use. Pins, timing constraints can be coded with the constraint file, in this lab switches and leds will be used.
4- The purpose of writing a testbench is to test how a design would work before implementing it on the FPGA. This could let the users to detect any errors before moving on.

Methodology

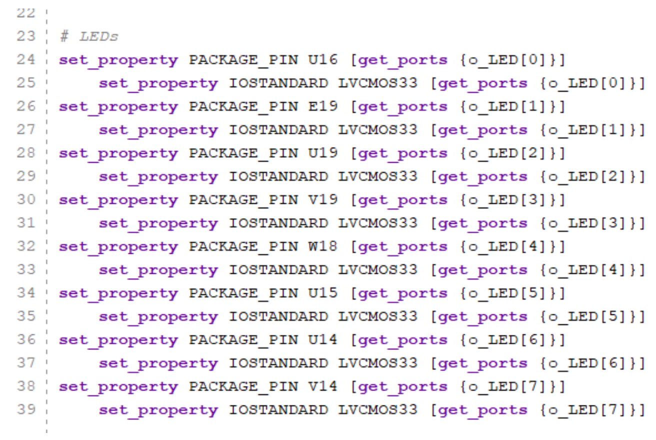
The last digit of my ID is 2, so I firstly changed operators in the lines 14, 15 and 16 of the first submodule to XOR, XNOR and AND. Investigating the circuit, I came across 6 bugs.

First one was a simple typing error, that a word missed the letter ‘t’. Another similar one was that there was a syntax error in the constraint file and a ‘}’ was missing. In similar fashion the chosen FPGA on the Vivado application was not the Basys 3 we are using, and it had to be changed in order for the constraint file to work correctly. The 4th bug was another error in the constraint file, the output LEDs were in incorrect order and were giving incorrect outputs when the design was implemented onto Basys 3. 5th bug was that the port map of the second submodule in the top_module.vhd was written wrong and I had to rewrite it. Lastly, the 6th bug was that there was an extra ‘the_beast’ phrase written at the end of the entity section’s declaration in the second submodule. Solving these designs were intended to obtain a system where the first 8 switches (from the rightmost side) were the inputs, and the rightmost 8 LEDs were the outputs for the desired gates and design.



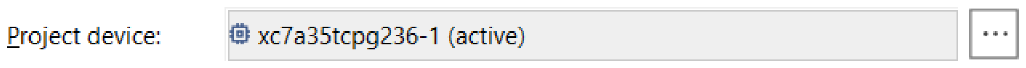
```
begin

    sub_module1_2 : sub_module1
        port map (
            i_input_byte  => i_SW,
            o_output_byte => s_ouput_1
        );

    sub_module2_1 : sub_module2
        port map (s_output_2, i_SW);
    s_output_3 <= unsigned(s_output_2) + 25;

    o_LED <= (not std_logic_vector(s_output_3)) xor s_output_1;

end Structural_top;
```

Figure 1: Bugs 1 and 5



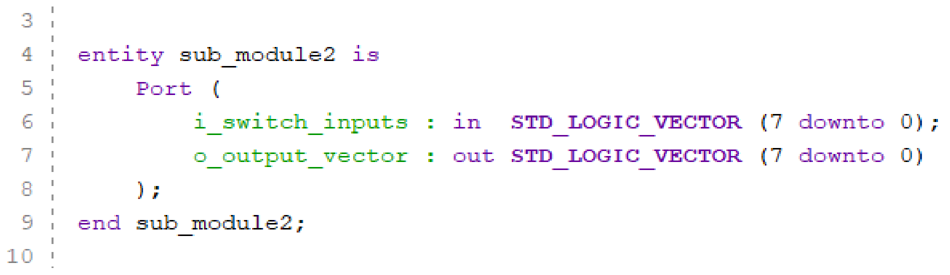
```
22
23  # LEDs
24  set_property PACKAGE_PIN U16 [get_ports {o_LED[0]}]
25      set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[0]}]
26  set_property PACKAGE_PIN E19 [get_ports {o_LED[1]}]
27      set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[1]}]
28  set_property PACKAGE_PIN U19 [get_ports {o_LED[2]}]
29      set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[2]}]
30  set_property PACKAGE_PIN V19 [get_ports {o_LED[3]}]
31      set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[3]}]
32  set_property PACKAGE_PIN W18 [get_ports {o_LED[4]}]
33      set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[4]}]
34  set_property PACKAGE_PIN U15 [get_ports {o_LED[5]}]
35      set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[5]}]
36  set_property PACKAGE_PIN U14 [get_ports {o_LED[6]}]
37      set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[6]}]
38  set_property PACKAGE_PIN V14 [get_ports {o_LED[7]}]
39      set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[7]}]
```

Figure 2: Bug 2 and 4 (2 in line 28 is fixed, 4 fixed in 24 and 38)



Project device: xc7a35tcpg236-1 (active)

Figure 3: Bug 3 (fixed)



```

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;

```

Figure 5: Bug 6 (fixed)

As now an implementation was able to be created without error, I later created a testbench in order to check if the design works as intended. I wanted to check the 256 outputs sequentially and wanted to make a comparison. I wrote a testbench where I make the simulation check every possible input's results.

Results

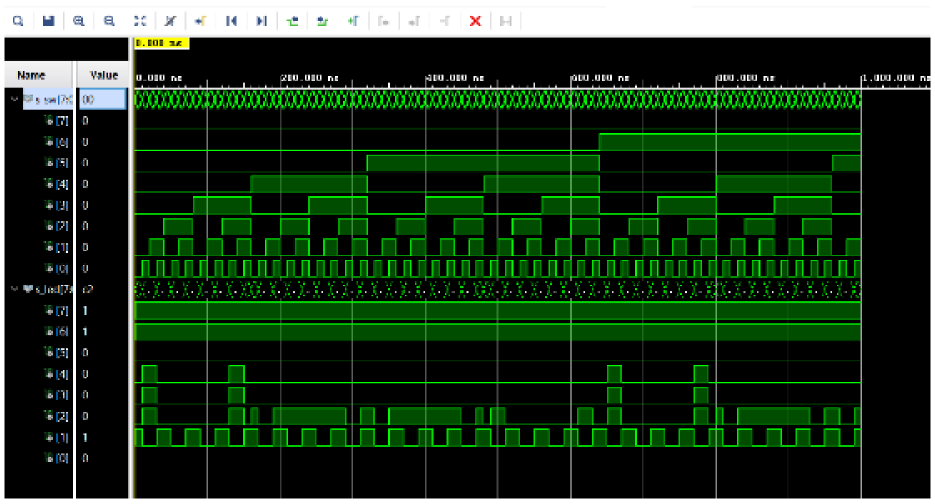


Figure 6: Simulation Results

Later, I obtained the simulation results and was able to visually see the behavior of the final design of the system and check how every input creates any outputs. Furthermore, I created the desired schematics.

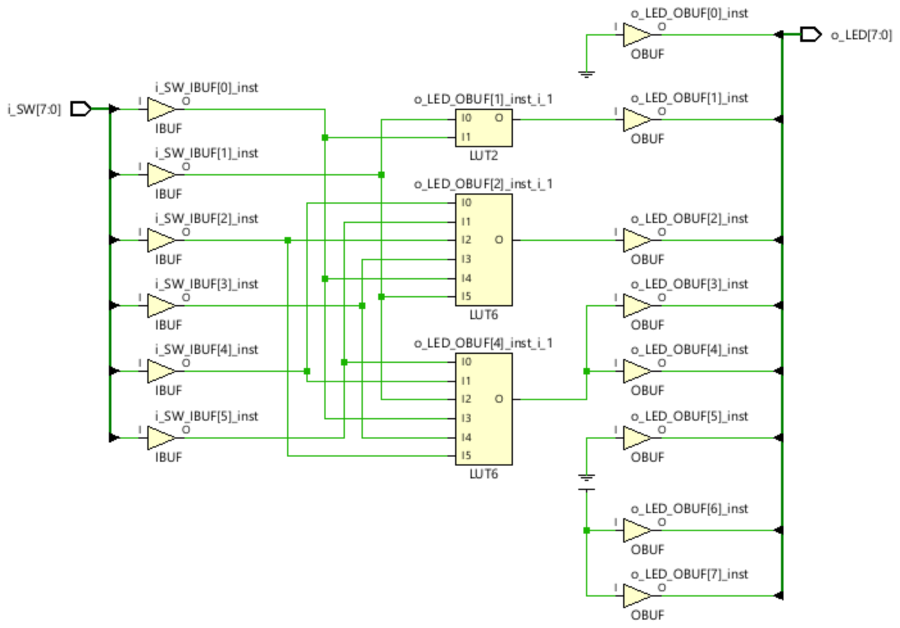


Figure 7: Synthesized Design Schematic

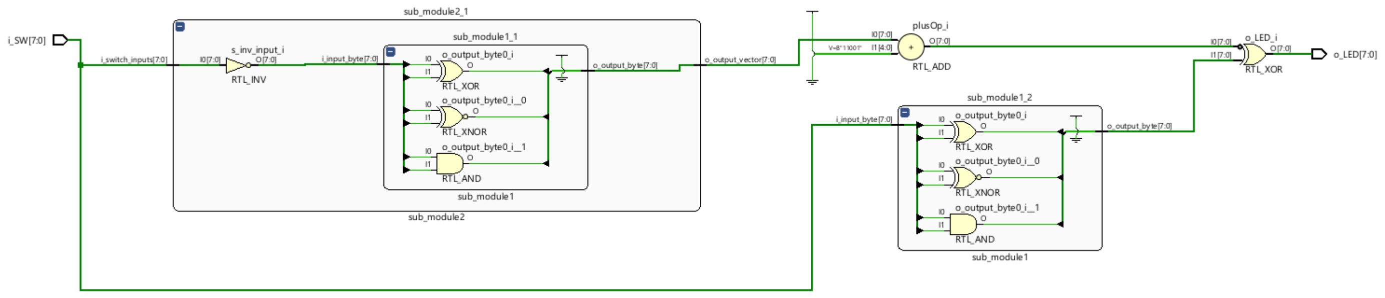


Figure 8: Schematic 2

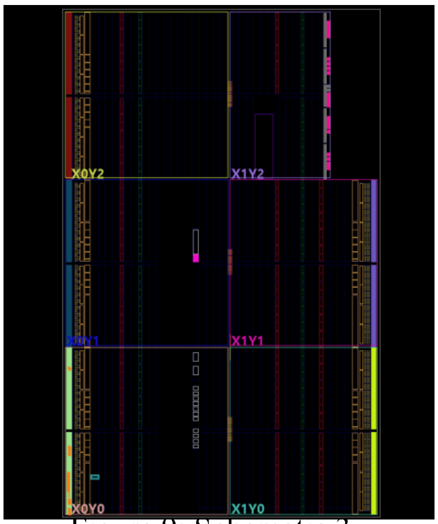


Figure 9: Schematic 3

These schematics display how the design works, simply how every input goes through multiple operations, and creates what kind of an output. They let the designer see what kind of a system he/she will obtain after implementing the result, and let that person visually analyze the design, make changes without difficultly visualizing the code and undergoing similar difficulties. In our schematics we can observe how our 8 bit input goes to the 8 bit output and decide simply what changes we want to make by checking these pictures.

Conclusion

The aim of this lab session was to get along with VHDL and do this by debugging and testing a certain given combinational circuit on Vivado. The given circuit had mistakes in it and we were expected to fix these problems through checking every bit of code, and this was get into writing and reading code written with VHDL. In this lab, we learned the simple compartments of every VHDL design, constraint files, testbenches, simulations and certain schematics. We had the opportunity to use the Basys 3 and understand some of its parts. In this lab I had a problem with finding the last bug and did not see the problem on the constraint file. I later checked the simulation and checked the desired outputs of the inputs, and saw that there was a change of the LEDs spots on the constraint file that gave incorrect results. I changed this and got the desired outputs. This lab was highly beneficial in terms of learning about VHDL and Basys 3 that we will use this semester, and understanding the basic logic behind digital designs.

Appendix

testb.vhd

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

use IEEE.NUMERIC_STD.ALL;


entity test_bench1 is

-- Port ( );

end test_bench1;



architecture Behavioral of test_bench1 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 s_sw :std_logic_vector(7 downto 0):= (others =>'0');
signal s_led: std_logic_vector(7 downto 0);



begin
uut: top_module port map(
i_sw => s_sw,
o_led =>s_led
);
stim_proc:process



begin
for i in 0 to 255 loop
   s_sw <= std_logic_vector(to_unsigned(i,8));
wait for 10 ns;
end loop;
end process stim_proc;
end Behavioral;
```

changed section of top_module.vhd

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