Ahmet Faruk Çolak

22102104

Lab 2: Introduction to VHDL

- **How does one specify the inputs and outputs of a module in VHDL?**

  Specifying inputs and outpust is doing with “constraint file” which is in our umbrella file.

```
# LEDs
set_property PACKAGE_PIN V14 [get_ports {o_LED[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[0]}]
set_property PACKAGE_PIN E19 [get_ports {o_LED[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[1]}]
set_property PACKAGE_PIN U19 [get_ports {o_LED[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[2]}]
set_property PACKAGE_PIN V19 [get_ports {o_LED[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[3]}]
set_property PACKAGE_PIN W18 [get_ports {o_LED[4]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[4]}]
set_property PACKAGE_PIN U15 [get_ports {o_LED[5]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[5]}]
set_property PACKAGE_PIN U14 [get_ports {o_LED[6]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[6]}]
set_property PACKAGE_PIN V16 [get_ports {o_LED[7]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[7]}]
```

  For example, in this picture, input is V14 switch and the output is LED0. If I change LED0 to LED8 for instance, V14 does not control LED0 anymore, but control LED8.

- **How does one use a module inside another code/module? What does PORT MAP do?**

  Modules are very helpful to build a complete code because one module can be used into another module and this makes lifes easier, basically.

  Port Map, is a guidance about which input is connected which output and helps us to find our way in our FPGA card, basically. We define something input/output and then do some exercise according to our definings, this is actually very important.

- **What is a constraint file? How does it relate your code to the pins on your FPGA?**

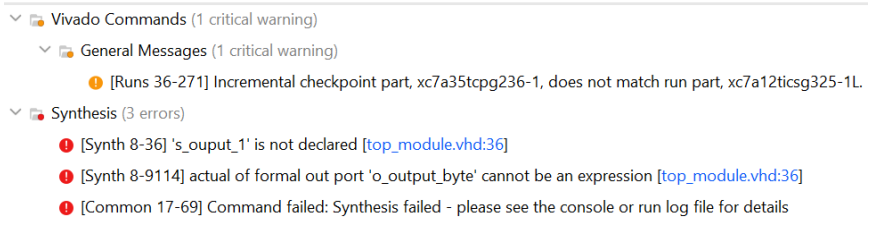
  - Constraint file sets which input/output in the VHDL code connects to which pin in FPGA. It typically with the extension .xdc, plays a crucial role in bridging the gap between VHDL code and the physical pins on your FPGA. It acts as a set of instructions for the synthesis and implementation tools, guiding them on how to map design onto the available resources of the FPGA.

- **What is the purpose of writing a testbench?**
  - There may be many purposes but in the perspective of this lab, the goal is debugging and see which parts are problematic.
  - Debugging isolates issues by allowing controlled testing of specific parts of the circuit.
  - Helps pinpoint the source of unexpected behavior by analyzing input-output relationships.

1) Modify the code submodule1.vhd line 14, 15 and 16. Change the logic gates to the corresponding gates given in the table below.

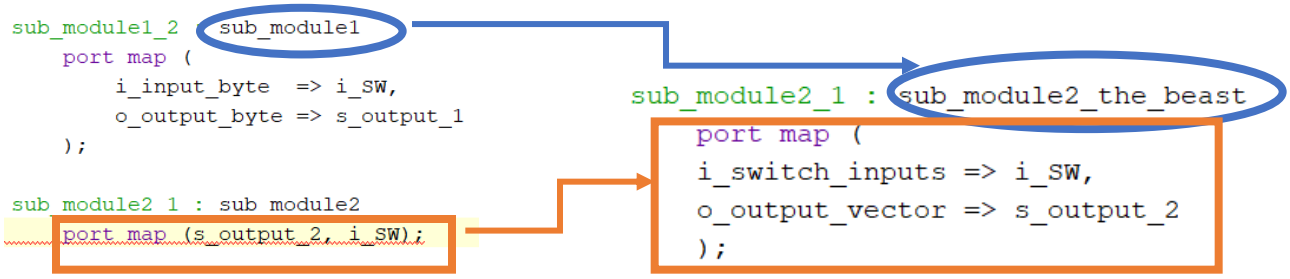
```
begin
    o_output_byte(0)          <= i_input_byte(0) or i_input_byte(1);
    o_output_byte(1)          <= i_input_byte(2) and i_input_byte(3);
    o_output_byte(2)          <= i_input_byte(4) nand i_input_byte(5);
    o_output_byte(5 downto 3) <= "010";
    o_output_byte(7 downto 6) <= (others => '0');

end Structural_sub1;
```

2) This is all errors:



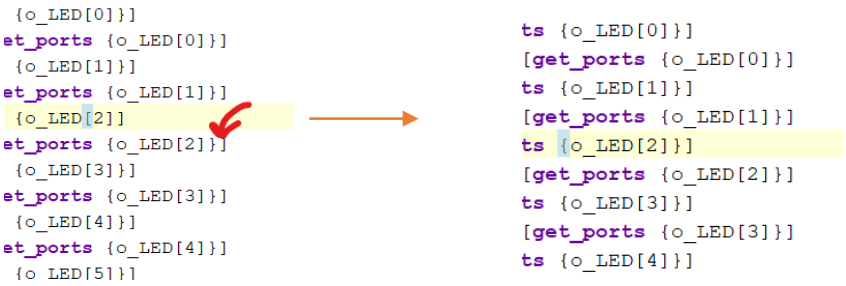
1- 's_ouput_1' is not declared There is a simple typo, it should be 's_output_1'

2- The parts coming after “sub_module1_2” do not matches each other. I changed “sub_module1” to sub_module1_2_the_beast”



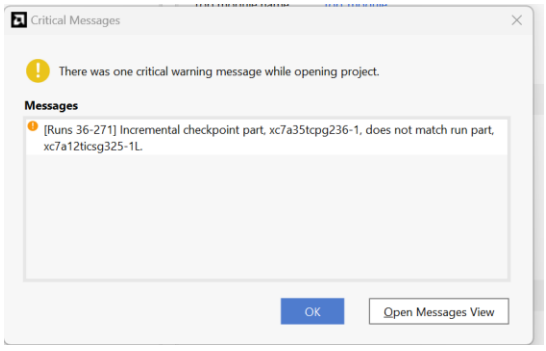
3- Also, there is an error about defining “sub_module_2_1” I re-define its inside.

4- Typo



“}” sign is missing on the left hand sided screenshot, then i fixed it.

5-



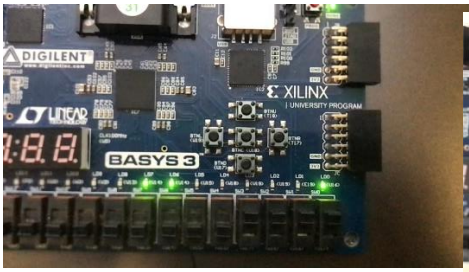
At the beginning, another FPGA card is selected in IDE, but BASYS 3 should be selected. I fixed it.

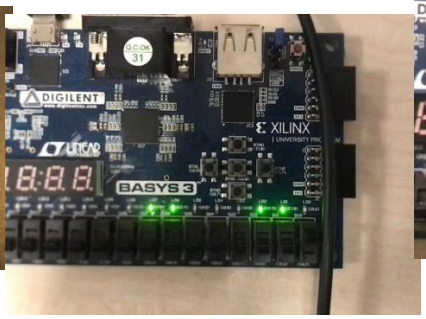
6-

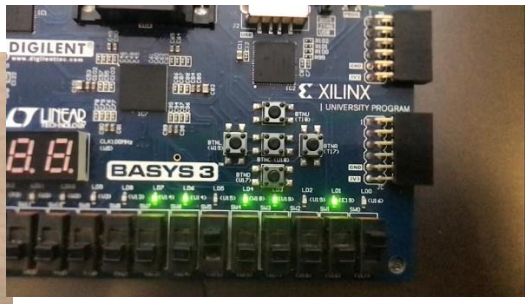
```
# LEDs
set_property PACKAGE_PIN V14 [get_ports {o_LED[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[0]}]
set_property PACKAGE_PIN E19 [get_ports {o_LED[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[1]}]
set_property PACKAGE_PIN U19 [get_ports {o_LED[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[2]}]
set_property PACKAGE_PIN V19 [get_ports {o_LED[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[3]}]
set_property PACKAGE_PIN W18 [get_ports {o_LED[4]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[4]}]
set_property PACKAGE_PIN U15 [get_ports {o_LED[5]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[5]}]
set_property PACKAGE_PIN U14 [get_ports {o_LED[6]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[6]}]
set_property PACKAGE_PIN U16 [get_ports {o_LED[7]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[7]}]
```

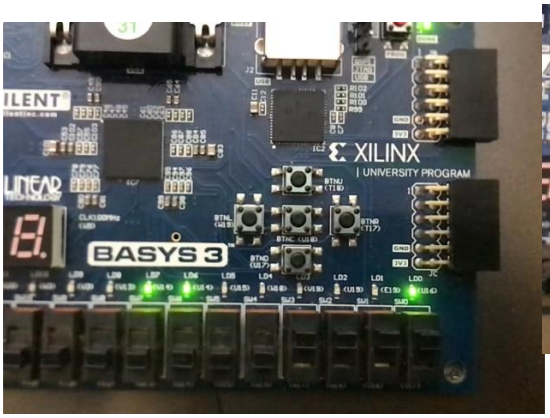
Highlighted pins should be switch each other, according to manuel of the BASYS 3.

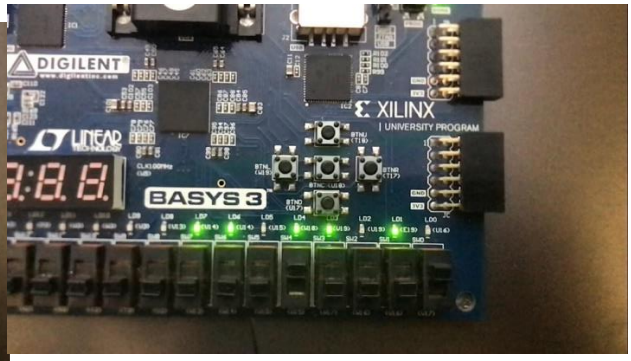
3) ...In your report, include example photos of your working FPGA (Photos of 5 different inputs/outputs are acceptable).



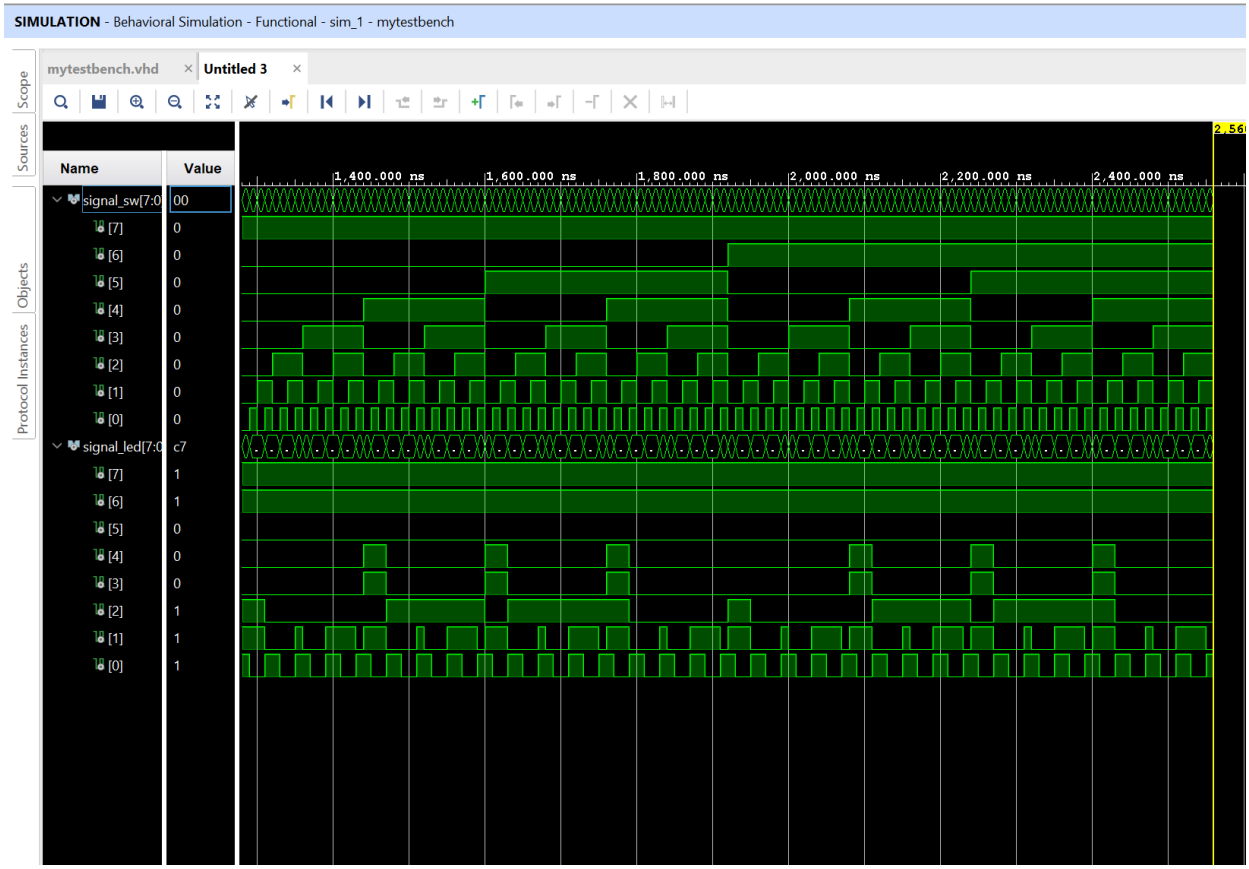








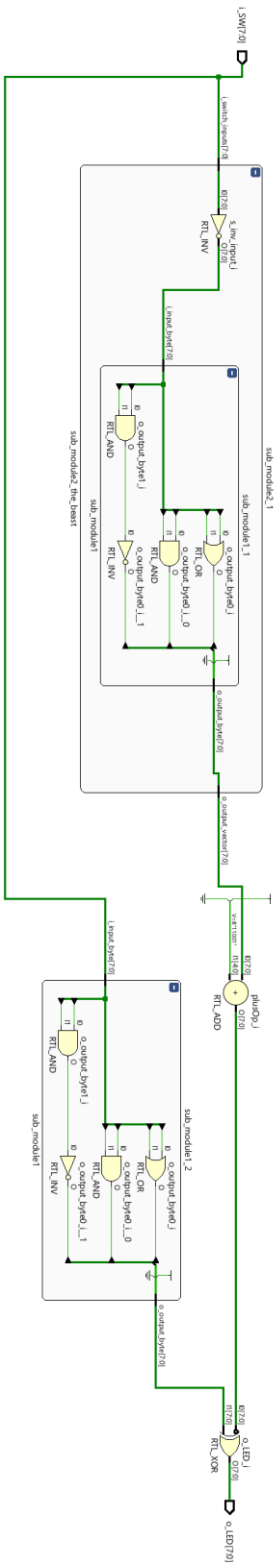
4) ...Include a screenshot of the waveform in your report (for 8 inputs there should be all 256 combinations in the waveform).



Test bench code:

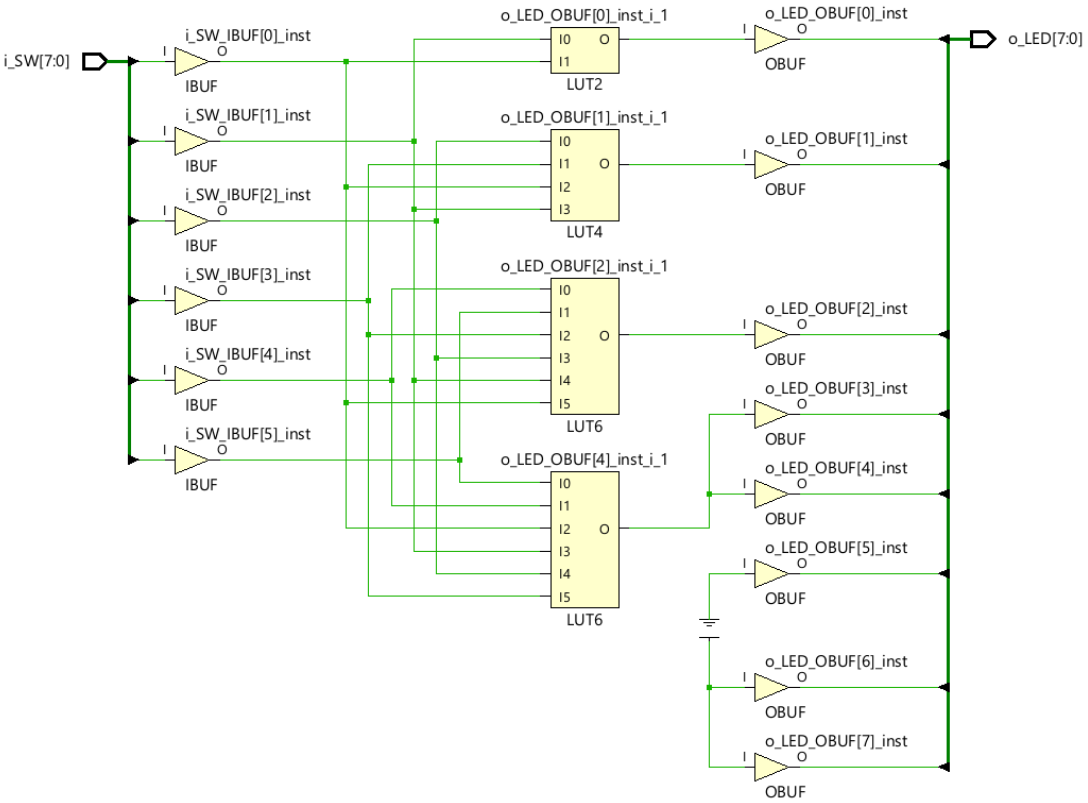
```
22  library IEEE;
23  use IEEE.STD_LOGIC_1164.ALL;
24  use IEEE.NUMERIC_STD.ALL;
25
26  entity mytestbench is
27  --  Port ( );
28  end mytestbench;
29
30  architecture Behavioral of mytestbench is
31  component top_module
32      Port(
33          i_sw : in STD_LOGIC_VECTOR (7 downto 0);
34          o_LED: out STD_LOGIC_VECTOR (7 downto 0)
35          );
36          end component;
37
38      signal signal_sw : STD_LOGIC_VECTOR (7 downto 0) :=(others => '0');
39      signal signal_led : STD_LOGIC_VECTOR (7 downto 0);
40
41  begin
42      UUT: top_module port map(
43      i_sw => signal_sw,
44      o_LED => signal_led
45      );
46
47      stim_proc: process
48      begin
49
50          for i in 0 to 255 loop
51              signal_sw <= STD_LOGIC_VECTOR (to_unsigned(i,8));
52              wait for 10ns;
53          end loop;
54      end process stim_proc;
55
56
57  end Behavioral;
```

5) RTL (please zoom in):



Synthesized Design:

Implemented Design:



Synthesized Design: "What" my design does. It explains the circuit basically.

Implemented Design: "How" my design is done on the specific FPGA. It explains circuit much detailed.

Waveform is generated according to implemented design.