Bilkent University EEE102

Lab 02 Report:

Introduction to VHDL

Burak Gökkaya 22202209

Purpose:

The aim of the this lab was using the BASYS 3 and Vivado. We are wanted to understand the mechanism of BASYS 3 with finding some errors in the given code. Also we are wanted to answer some questions about BASYS3

Methodology:

Firstly, we look the files in the Moodle which are related to VHDL, and we tried to answer the questions:

Q1.) How does one specify the inputs and outputs of a module in VHDL?

Answer: At the beginning, we declare the inputs and outputs in the Entity part.

Q2.) How does one use a module inside another code/module? What does PORT MAP do?

Answer: In VHDL, we can use the module inside another module by using PORT MAP. With help of port map we are mapping the signals to its ports.

Q3.) What is a constraint file? How does it relate your code to the pins on your FPGA?

Answer: A constraint file is a file that contains timing, placement etc. for a FPGA device. Then we specify the assignments for each input, output, LED.

Q4.) What is the purpose of writing a testbench?

Answer: The purpose of the testbench is checking the codes if it is working properly. Before generating bitstream and implement the code to BASYS3; we can see that is code is true or not.

After answering the all questions, we started to lab. In first step, we are given codes and there were six mistakes. Also we are wanted to modify the code according to the last digit of our

student ID. My last digit was 9 so I changed the code XNOR, XNOR, AND.

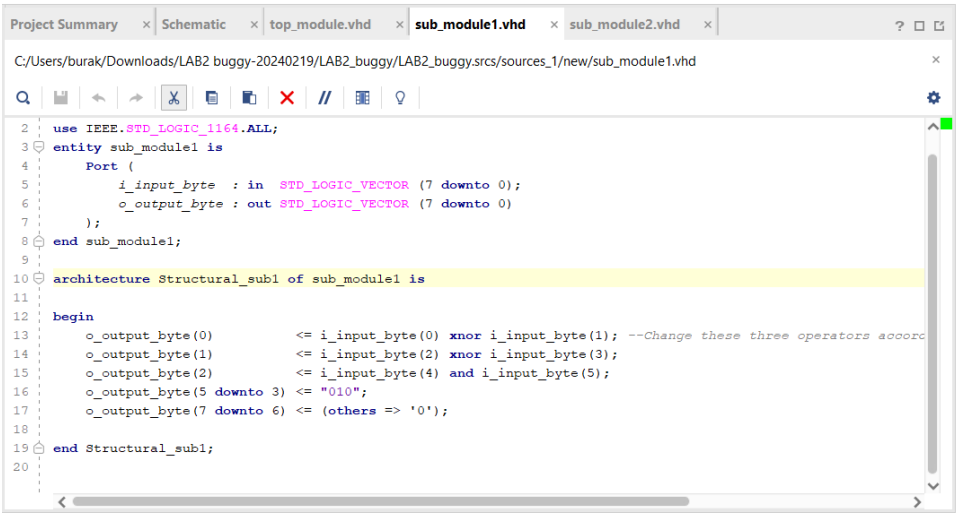

Figure 1 : XNOR, XNOR, AND

In second step, we are wanted investigate the given code and find six bugs. I found all of them. Bugs were:

1-) There was a letter mistake while writing s_output_1

2-) There was a mistake of sub_module2's names.

3-) There was a mistake at port map of sub_module2's while implementing it to top_module.

4-) There was a bug in constraint file. There isn't curly bracket.

5-) Also again there was bug in constraint file. Led's placement wasn't correct.

6-) The code wasn't working for the BASYS 3. So when we generate the bitstream we need to solve this problem.

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

Figure 2: Error 1

```
sub_module1_2 : sub_module1
    port map (
        i_input_byte  => i_SW,
        o_output_byte => s_output_1
    );
```

Figure 3: Solution of Error 1

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

Figure 4: Error 2 and 3

```
sub_module2_1 : sub_module2_the_beast
    port map (
    o_output_vector => s_output_2,
    i_switch_inputs =>  i_SW);
s_output_3 <= unsigned(s_output_2) + 25;
```

Figure 5: Solution of Error 2 and 3

```
20      set_property IOSTANDARD LVCMOS33 [get_ports {i_SW[7]}]
21
22
23  # LEDs
24  set_property PACKAGE_PIN V14 [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 U16 [get_ports {o_LED[7]}]
39      set_property IOSTANDARD LVCMOS33 [get_ports {o_LED[7]}]
```

Figure 6: Error 4 and 5

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

Figure 7: Solution of error 4 and 5

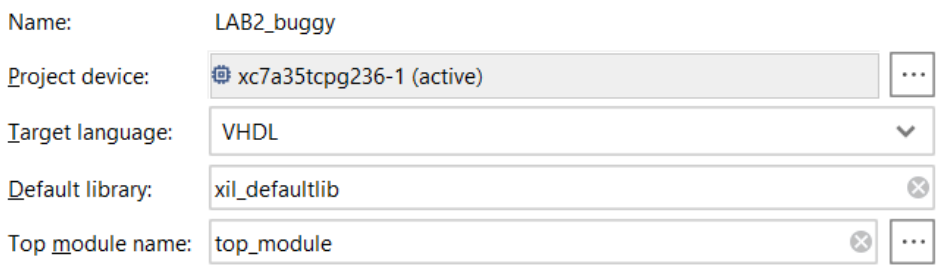

Figure 8: Solution of error 6

In the third step, we implemented the code to Basys 3.

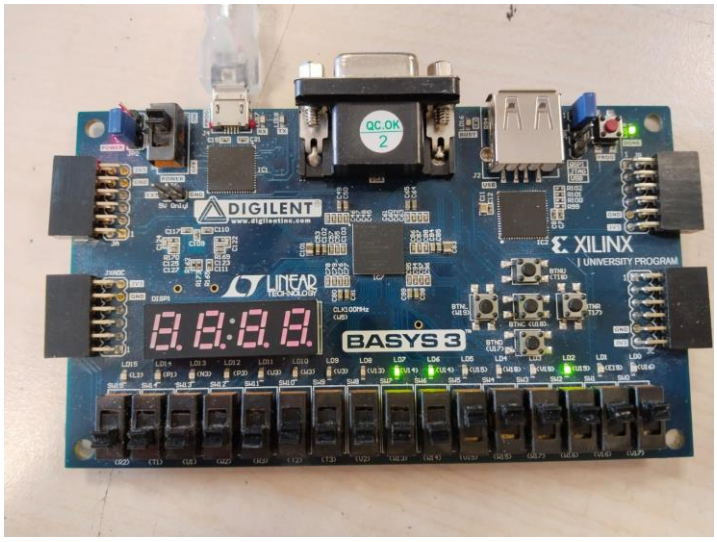

Figure 9: ‘00111111’

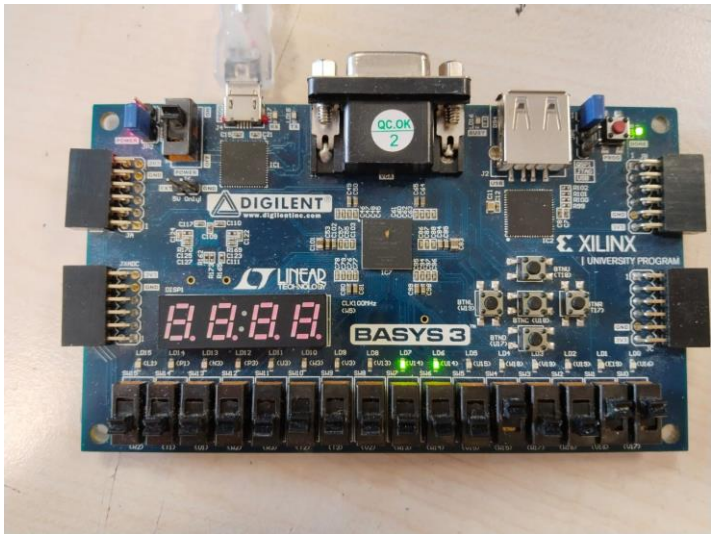

Figure 10: ‘00010011’

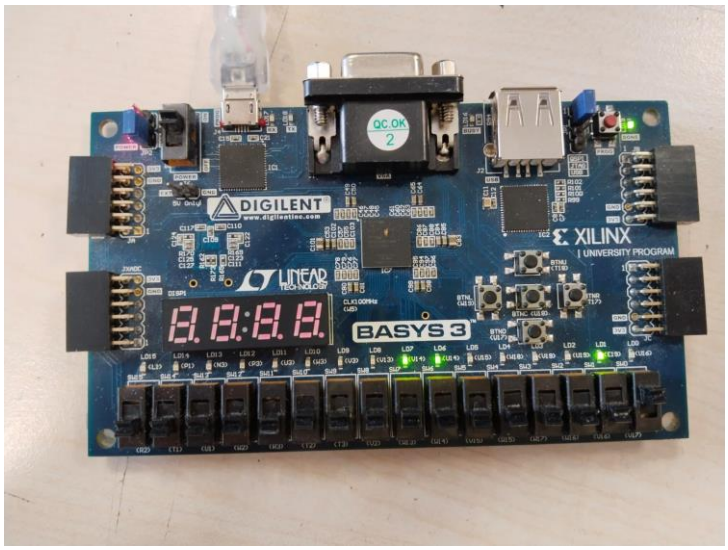

Figure 11: ‘00000001’

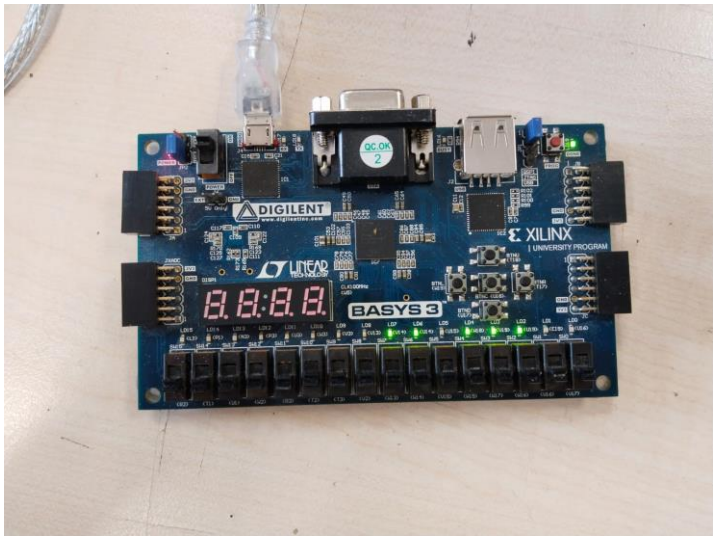

Figure 12: ‘00000000’

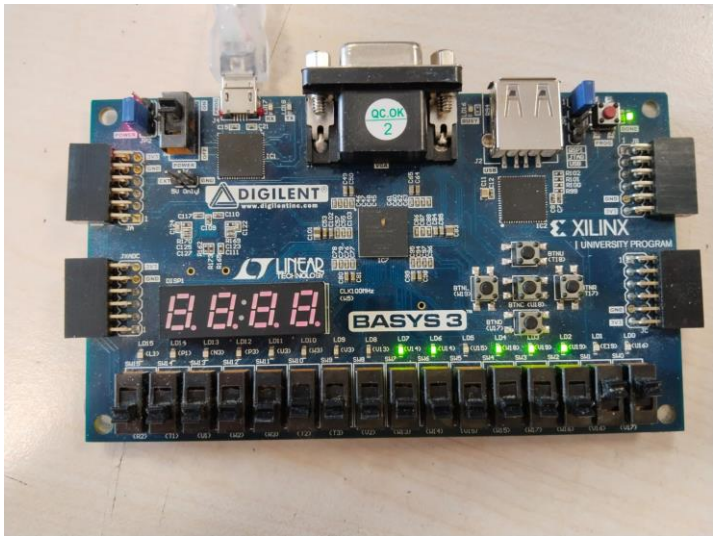

Figure 13: ‘00000011’

At the 4th step we are wanted to write a testbench code and simulate it.

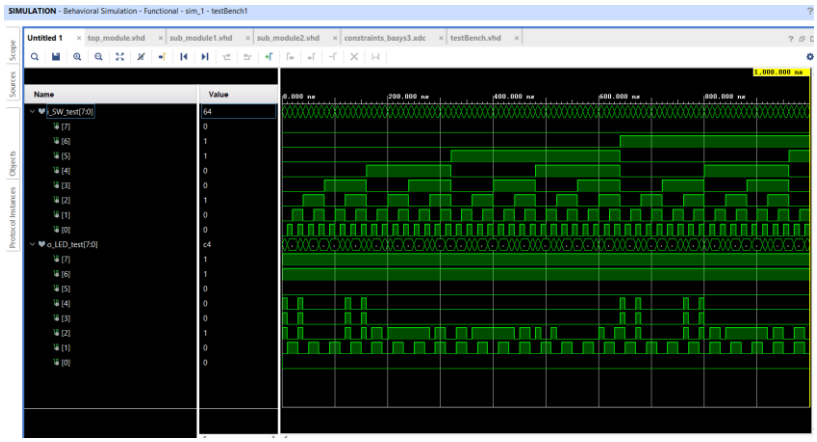

Figure 14: Simulation

And at the last step we are wanted to draw the RTL schematic, Synthesized Design and Implemented Design in Vivado. Then compare them.

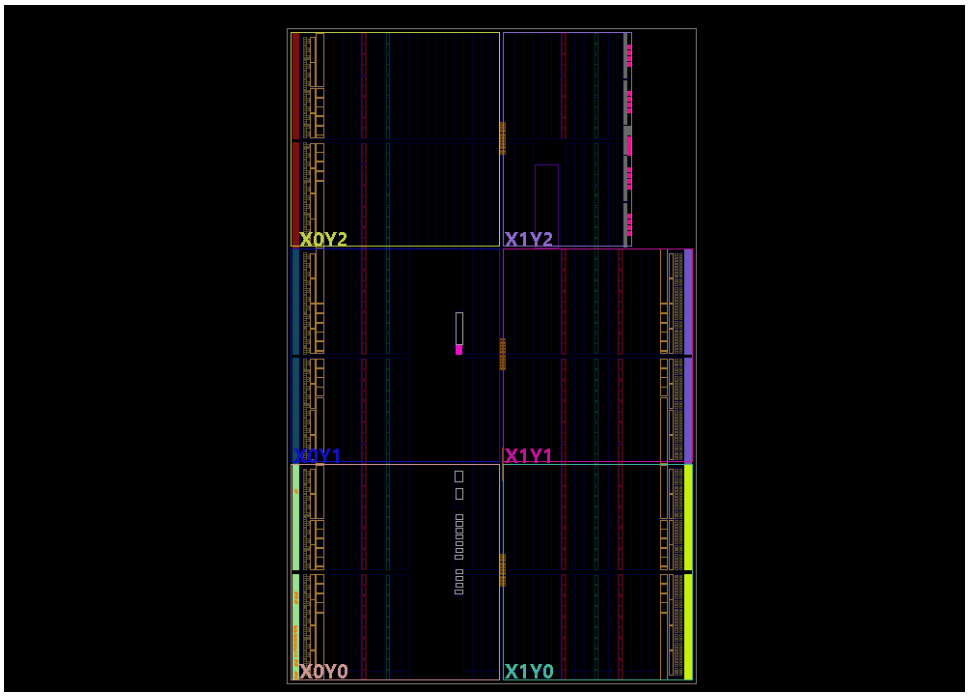

Figure 15: Synthesized Design

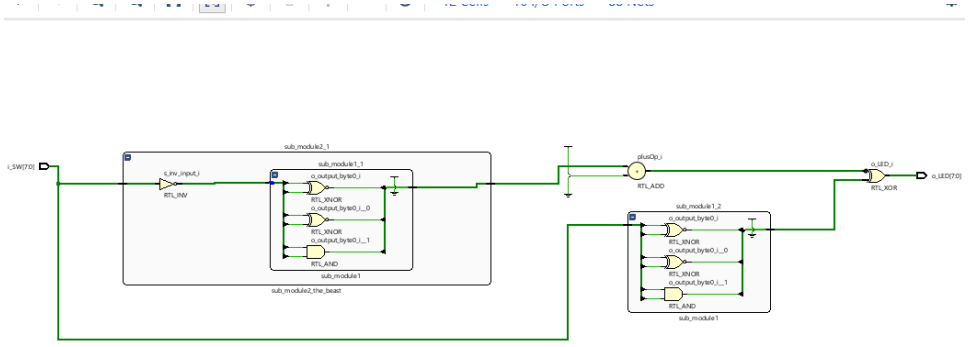

Figure 16: Schematic of circuit

Conclusion:

As a result, I understood Basys 3 and Vivado mechanism better. I answered some questions, therefore I understood the coding better. We wrote test bench codes and found 6 error. So that we learned way better. I think it was really successful experiment for us.

Appendix:

---top_module

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

entity top_module is
    Port (
        i_SW  : in  STD_LOGIC_VECTOR (7 downto 0);
        o_LED : out STD_LOGIC_VECTOR (7 downto 0)
    );
end top_module;

architecture Structural_top of top_module is

    component sub_module1 is
        port (
            i_input_byte  : in  STD_LOGIC_VECTOR (7 downto 0);
            o_output_byte : out STD_LOGIC_VECTOR (7 downto 0)
        );
    end component sub_module1;

    component sub_module2_the_beast is
        port (
            i_switch_inputs : in  STD_LOGIC_VECTOR (7 downto 0);
            o_output_vector : out STD_LOGIC_VECTOR (7 downto 0)
        );
    end component sub_module2_the_beast;

    signal s_output_1, s_output_2 : STD_LOGIC_VECTOR(7 downto 0) := (others => '0');
```

```
   signal s_output_3           : unsigned(7 downto 0)       := (others => '0');

begin

   sub_module1_2 : sub_module1
      port map (
         i_input_byte  => i_SW,
         o_output_byte => s_output_1
      );

   sub_module2_1 : sub_module2_the_beast
      port map (
      o_output_vector => s_output_2,
      i_switch_inputs =>  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;
```

```
---submodule1
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity sub_module1 is
Port (
i_input_byte  : in  STD_LOGIC_VECTOR (7 downto 0);
o_output_byte : out STD_LOGIC_VECTOR (7 downto 0)
);
end sub_module1;

architecture Structural_sub1 of sub_module1 is
```

```
begin
o_output_byte(0)        <= i_input_byte(0) xnor i_input_byte(1); --Change these three operators according to the table in the lab document
o_output_byte(1)        <= i_input_byte(2) nand i_input_byte(3);
o_output_byte(2)        <= i_input_byte(4) or i_input_byte(5);
o_output_byte(5 downto 3) <= "010";
o_output_byte(7 downto 6) <= (others => '0');

end Structural_sub1;
```

```
---submodule2
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
---submodule2's inpu
entity sub_module2_the_beast 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_the_beast;

architecture Structural_sub2 of sub_module2_the_beast is
   component sub_module1 is
      port (
         i_input_byte  : in  STD_LOGIC_VECTOR (7 downto 0);
         o_output_byte : out STD_LOGIC_VECTOR (7 downto 0)
      );
   end component sub_module1;

   signal s_inv_input : STD_LOGIC_VECTOR(7 downto 0) := (others => '0');
begin
   s_inv_input <= not i_switch_inputs;
```

```
   sub_module1_1 : sub_module1
      port map (
         i_input_byte  => s_inv_input,
         o_output_byte => o_output_vector
      );
end Structural_sub2;
```

```
--testbench
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity testBench1 is
end testBench1;

architecture Behavioral of testBench1 is
  SIGNAL i_SW_test : STD_LOGIC_VECTOR(7 downto 0);
  SIGNAL o_LED_test : STD_LOGIC_VECTOR(7 downto 0);
begin
  DUT : ENTITY work.top_module
  PORT MAP (
   i_SW => i_SW_test,
   o_LED => o_LED_test
   );
PROCESS
begin
     for i in 0 to 255 loop
       i_SW_test <= std_logic_vector(to_unsigned(i,8));
       WAIT FOR 10 ns;
     end loop;
     wait;
```

```
  end process;

end Behavioral;
```

---Constrain

Constraint file sets which input/output in the VHDL code connects to which pin in FPGA.

Three characters after PACKAGE_PIN gives the FPGA pin which is also written next to all LEDs/Switches on the board.

The second line (IOSTANDARD LVCMOS33) sets the voltage standard for the pins which is constant and always the same for BASYS3.

```
set_property PACKAGE_PIN V17 [get_ports {i_SW[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {i_SW[0]}]
set_property PACKAGE_PIN V16 [get_ports {i_SW[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {i_SW[1]}]
set_property PACKAGE_PIN W16 [get_ports {i_SW[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {i_SW[2]}]
set_property PACKAGE_PIN W17 [get_ports {i_SW[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {i_SW[3]}]
set_property PACKAGE_PIN W15 [get_ports {i_SW[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {i_SW[4]}]
set_property PACKAGE_PIN V15 [get_ports {i_SW[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {i_SW[5]}]
set_property PACKAGE_PIN W14 [get_ports {i_SW[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {i_SW[6]}]
set_property PACKAGE_PIN W13 [get_ports {i_SW[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {i_SW[7]}]


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