EEE 102

LAB 2

Student Name: Robin Umut Kızıl

Student ID: 22003260

Class code and section: EE-102-01

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

  In VHDL, Inputs and outputs specified using port identification part in entity. All ports that will be used have to be identified whether they are ‘in’, ‘out’ or ‘in out’.

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

  A module is code written to perform a task. We use the port map part to match the signals that will be used with the signals inside the module.

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

  The constraint file determines the physical properties and state of the generated circuit. It serves to control the signals connected to the pins, and files such as clock frequency and keep the circuit in order.

- **What is the purpose of writing a testbench?**

  With testbench, the signal wavelength can be checked step by step. so it can be checked whether there is a problem in the circuit and whether the desired result is obtained.

Purpose of Lab:

This experiment aims to get used to vhdl and basys3 using vivado, solve a real-life problem with a combinatorial circuit, prepare a testbreach to solve the problem, simulate the experiment, and become familiar with the use of LEDs and switches.

Real-Life Problem:

The real life problem, a game made in America and played by flipping 4 coin, is considered. If 4 consecutive coin flips land at least 2 heads in a row, the playing player wins.

Design Specification:

On any coin, heads are represented by 1 and tails by 0. In Basys3, rightmost 4 key and the rightmost 5 led used. Since the led on each switch is connected, it can be shown whether the switch is open or not. 5th led from the right show the winning status of the player, that is, he has come to the head 2 times in a row. (Since the leds and switches on basys3 are sorted by index number, switch numbers (sw0-sw1-sw2-sw3) and led numbers (led0-led1-led2-led3-led4) are ordered from 0 to n-1.) Logic function of problem is:

$$f(sw0, sw1, sw2, sw3) = (\ sw0\ and\ sw1\ )\ or\ (\ sw1\ and\ sw2)\ or\ (\ sw2\ and\ sw3\ )$$

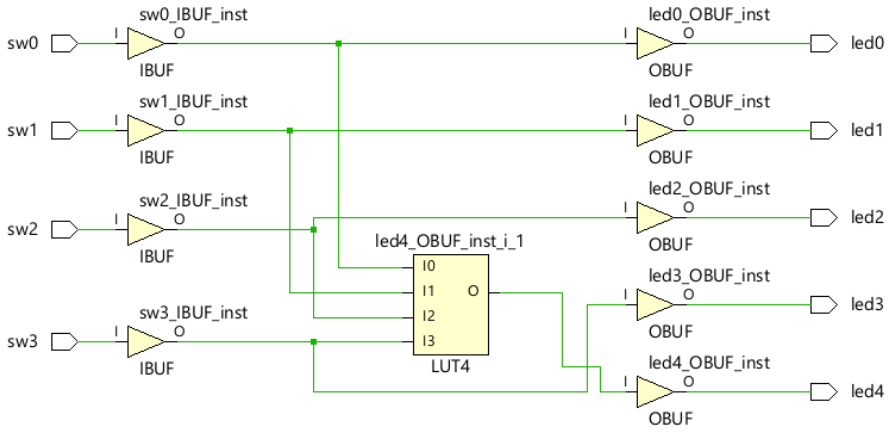

Table 1: Schematic of logic function

Input Switches				Output Led that indicating the win status
Sw0	Sw1	Sw2	Sw3	Led4
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Table 2: Truth table of win condition represented on led4

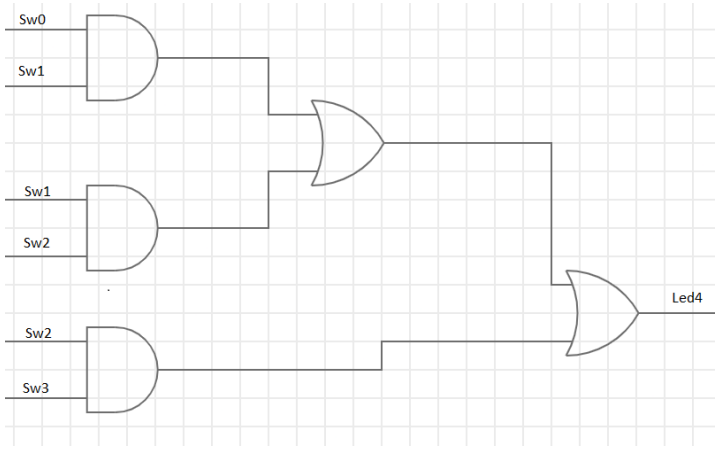

Table 3: Logic circuit of the function

Methodology:

First, an empty project is opened in vivado to use basys3 and vhdl. Then a new design source was created from the source section. In this file, the necessary 4 switches and 5 leds are defined in the port section. Then 4 switches that are physically below the 4 leds are connected, so it can be seen whether the switches are open or not. Then, the required logic function was written on the 5th LED (it can be seen in the additional files section). The code was brought to a working state. Then the constraint file is assigned. By changing the necessary places in the file, the connection between the code and the ports on the physical card was established.

Also added simulation source. testbench was created and observed by writing the necessary codes (which can be seen in additional files).

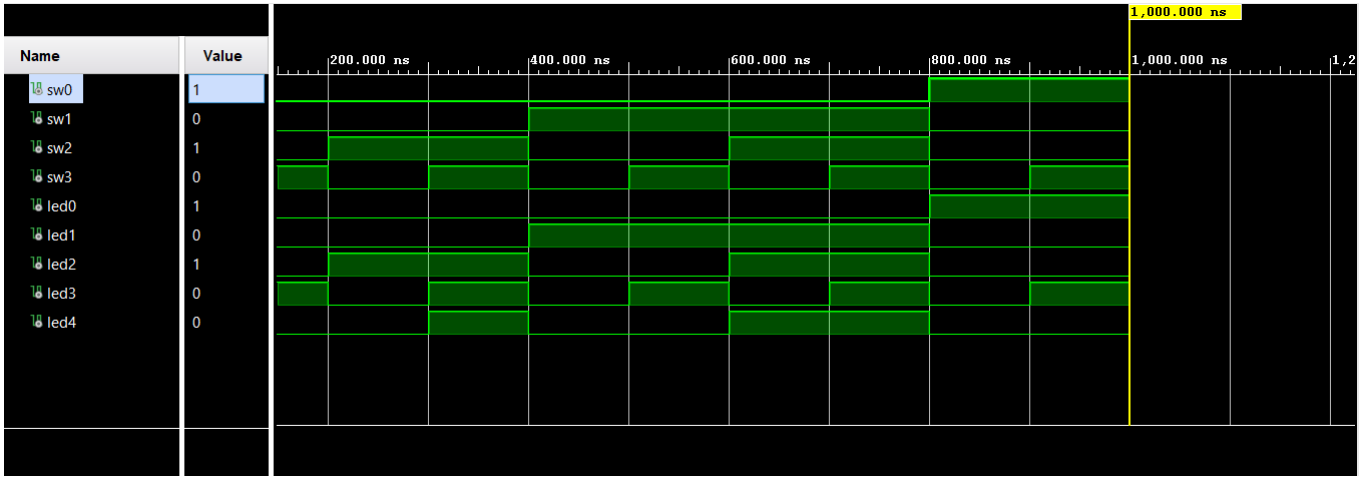

Table 4: Testbench timing diagram

At the same time, the code successfully passed the stages of synthesis, implementation, generation bitstream respectively, and finally successfully run on basys3 with the help of hardware manager.

Result:

As a result of the experiment, in the designed problem, the combination of 2 head states made the logic function 1, as can be seen in the additional files, this situation was provided and the code ran successfully. Also, while doing this, it was learned to write basic vhdl coding with switch and led control, basic logic function with vhdl on vivado and to apply it on basys3. So lab is successful.

Appendices:

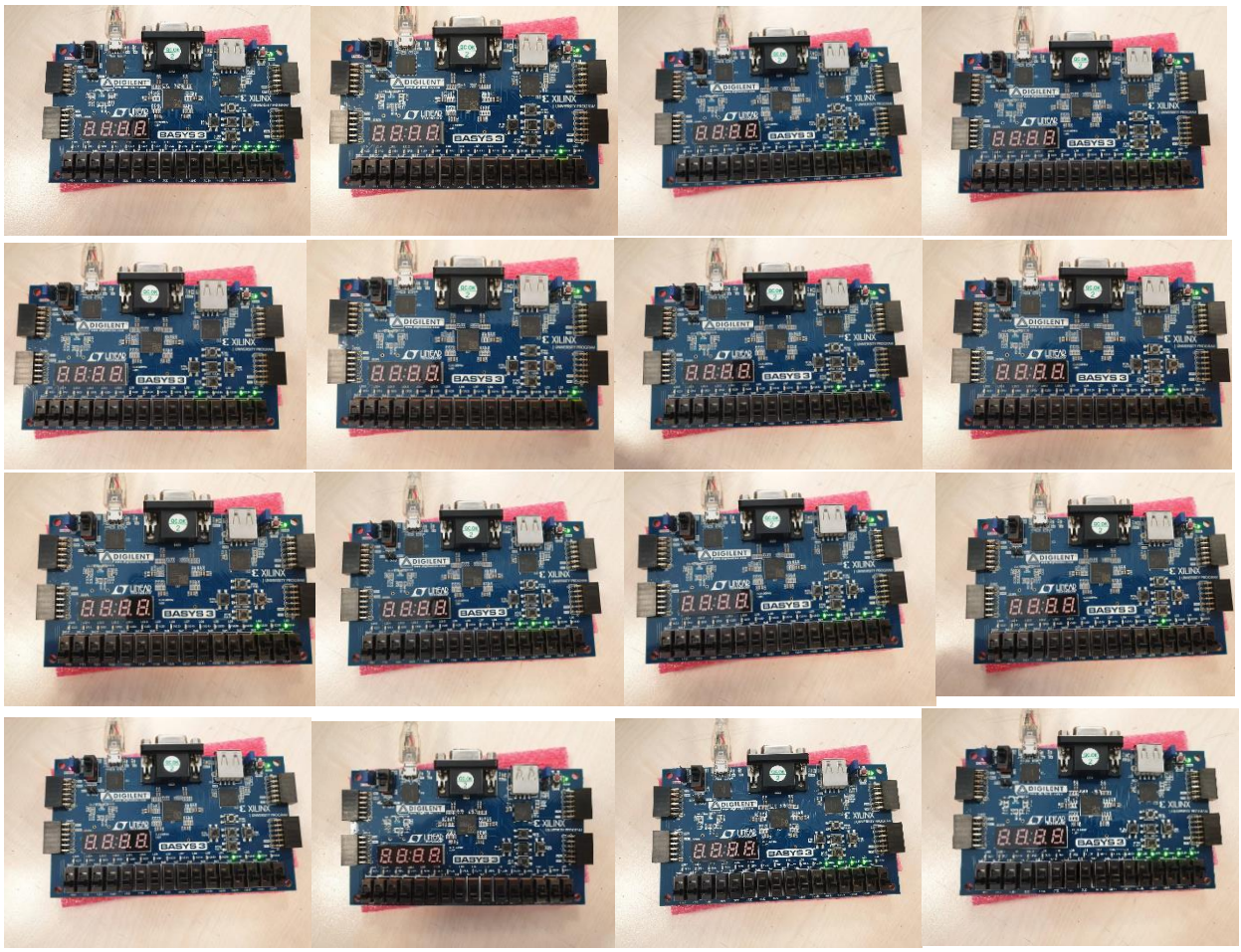

Table 5: basys 3 images in all situations

I3	I2	I1	I0	O=I0 & I1 + I2 & I3 + I1 & I2
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Table 6: Truth Table of LUT4(From table 1)

Design Source Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity Lab_2_Head_or_Tail is
   Port (
        sw0 : in STD_LOGIC;
        sw1 : in STD_LOGIC;
        sw2 : in STD_LOGIC;
        sw3 : in STD_LOGIC;
        led0 : out STD_LOGIC;
        led1 : out STD_LOGIC;
        led2 : out STD_LOGIC;
        led3 : out STD_LOGIC;
        led4 : out STD_LOGIC);
end Lab_2_Head_or_Tail;

architecture Behavioral of Lab_2_Head_or_Tail is

begin

led0 <= sw0;
led1 <= sw1;
led2 <= sw2;
led3 <= sw3;

led4 <= ( sw0 and sw1 ) or ( sw1 and sw2) or ( sw2 and sw3 );

end Behavioral;
```

Testbrench Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity Testbench1 is
   end Testbench1;

architecture Behavioral of Testbench1 is
component Lab_2_Head_or_Tail
   Port (
        sw0 : in STD_LOGIC;
        sw1 : in STD_LOGIC;
        sw2 : in STD_LOGIC;
        sw3 : in STD_LOGIC;
        led0 : out STD_LOGIC;
        led1 : out STD_LOGIC;
        led2 : out STD_LOGIC;
        led3 : out STD_LOGIC;
        led4 : out STD_LOGIC);
end component;

signal sw0,sw1,sw2,sw3 : STD_LOGIC;
signal led0,led1,led2,led3,led4 : STD_LOGIC;

begin

uut: Lab_2_Head_or_Tail PORT MAP (
sw0 => sw0,
```

```
sw1 => sw1,
sw2 => sw2,
sw3 => sw3,
led0 => led0,
led1 => led1,
led2 => led2,
led3 => led3,
led4 => led4
);
stim_proc: process


begin
sw0<='0';sw1<='0';sw2<='0';sw3 <='0';
wait for 100ns;
sw0<='0';sw1<='0';sw2<='0';sw3 <='1';
wait for 100ns;
sw0<='0';sw1<='0';sw2<='1';sw3 <='0';
wait for 100ns;
sw0<='0';sw1<='0';sw2<='1';sw3 <='1';
wait for 100ns;

sw0<='0';sw1<='1';sw2<='0';sw3 <='0';
wait for 100ns;
sw0<='0';sw1<='1';sw2<='0';sw3 <='1';
wait for 100ns;
sw0<='0';sw1<='1';sw2<='1';sw3 <='0';
wait for 100ns;
sw0<='0';sw1<='1';sw2<='1';sw3 <='1';
```

```
wait for 100ns;

sw0<='1';sw1<='0';sw2<='0';sw3 <='0';
wait for 100ns;
sw0<='1';sw1<='0';sw2<='0';sw3 <='1';
wait for 100ns;
sw0<='1';sw1<='0';sw2<='1';sw3 <='0';
wait for 100ns;
sw0<='1';sw1<='0';sw2<='1';sw3 <='1';
wait for 100ns;

sw0<='1';sw1<='1';sw2<='0';sw3 <='0';
wait for 100ns;
sw0<='1';sw1<='1';sw2<='0';sw3 <='1';
wait for 100ns;
sw0<='1';sw1<='1';sw2<='1';sw3 <='0';
wait for 100ns;
sw0<='1';sw1<='1';sw2<='1';sw3 <='1';
wait for 100ns;
 end process;


end Behavioral;
```