Atalay Üstündağ

22203554

EE102-03

04.10.2023

Lab 2: Introduction to VHDL

1.Purpose

This experiment aims to teach VHDL coding language, applying it on Basys 3, and using Vivado. And it achieves this purpose by solving a daily life problem with using logic gates.

2. Requirements

3 inputs and 1 output is required for this experiment. My inputs are opt1, opt2, opt3 and my output is result.

3. Methodology

This experiment is designed to solve a daily problem by using logic gates. Here is the scenerio: A woman's car has a flat tire and a broken engine. In order to fix her car, she has three options. First option is going to tire shop and fixing only her tire, second option is going to an engineer who can fix engines but can not fix tires and the third option is going to a mechanic who can fix both tire and engine. In a nutshell, for the car to be repaired fully its tire and engine must be fixed.

Variable " opt1" represents the tire shop. Variable "opt2" represents the engineer. Variable "opt3" represents mechanic. Lastly variable "result" represents if the the car is fixed or not. The expression for the solution of problem is:

result = (opt1 . opt2) + opt3

opt1	opt2	opt3	result
0	0	0	0
1	0	0	0
0	1	0	0
0	0	1	1
1	1	0	1
1	0	1	1
0	1	1	1
1	1	1	1

Table1 : Truth table for the problem

After deciding the real life scenerio, a design source, a testbench code and a constraint file is written. Then Basis 3 is programmed.

4. Results and Researchs

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

PORT function specifies the input and output variables in VHDL. Following code can be used to specify the variables.

```
entity Design is

 Port (   in1 : in STD_LOGIC;

          In2 : in STD_LOGIC;

          In3 : out STD_LOGIC);

end Design;
```

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

Using a module inside of another module is possible with the help of the Port Map method.

For example,

```
uut: Design port map ( in1 => in1,

          in2 => in2,

          in3 => in3 );
```

With using Port Map the Design module is gets called in testbench.

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

A constraint file is written in order to specify which switch will represent which input and which LED will represent which output.

For example:

set_property PACKAGE_PIN W2 [get_ports {opt3}]

set_property IOSTANDARD LVCMOS33 [get_ports {opt3}]

By switching W2 switch, variable opt3 is arrenged 1 or 0:

• What is the purpose of writing a testbench?

Testbenchs are codes that allows us the chance to know if our design is working as we expected.

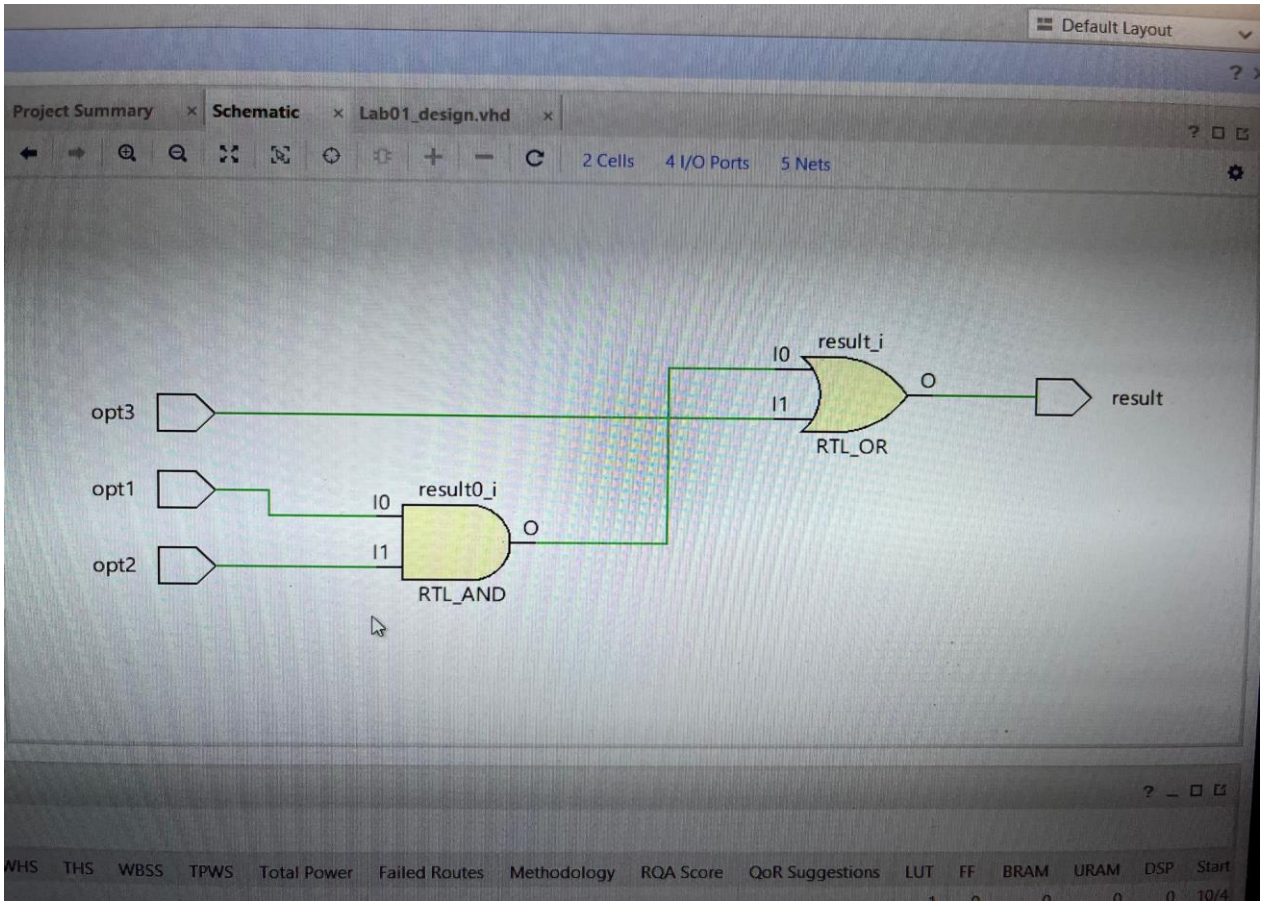

Figure 1: Schematic Diagram of Circuit

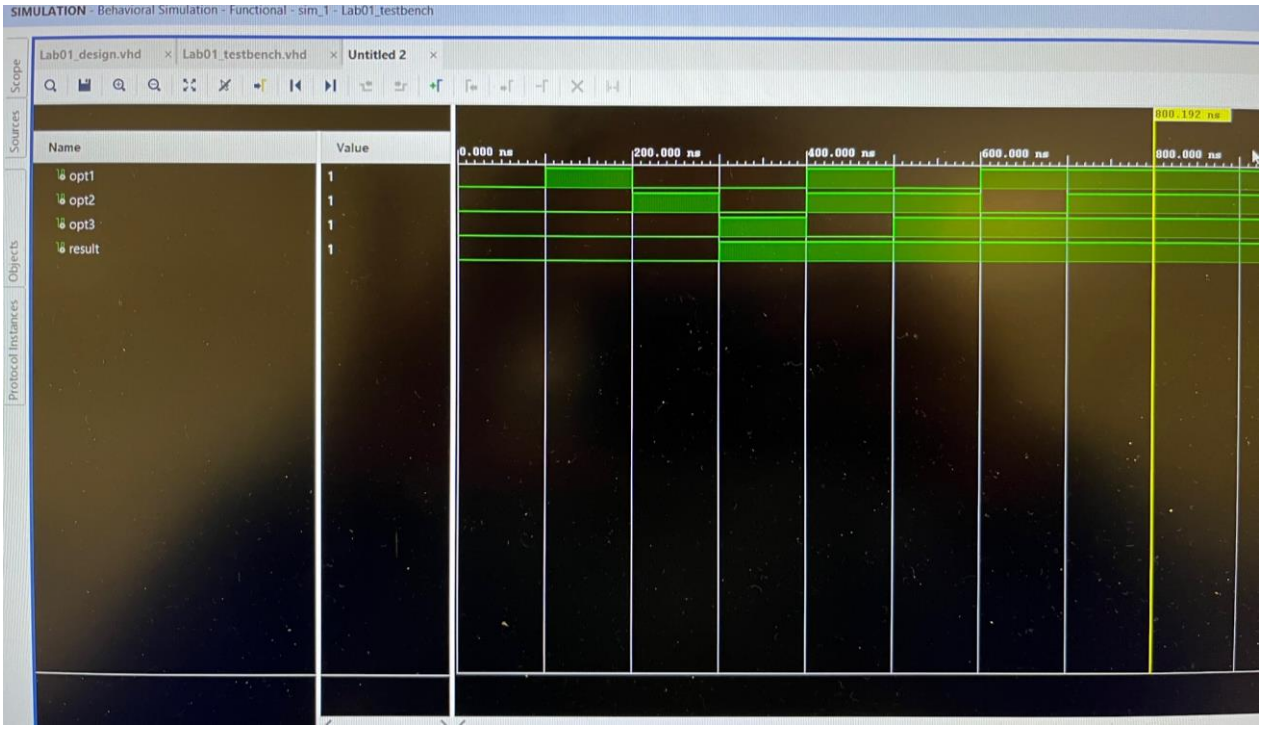

Figure 2: Waveform of Circuit

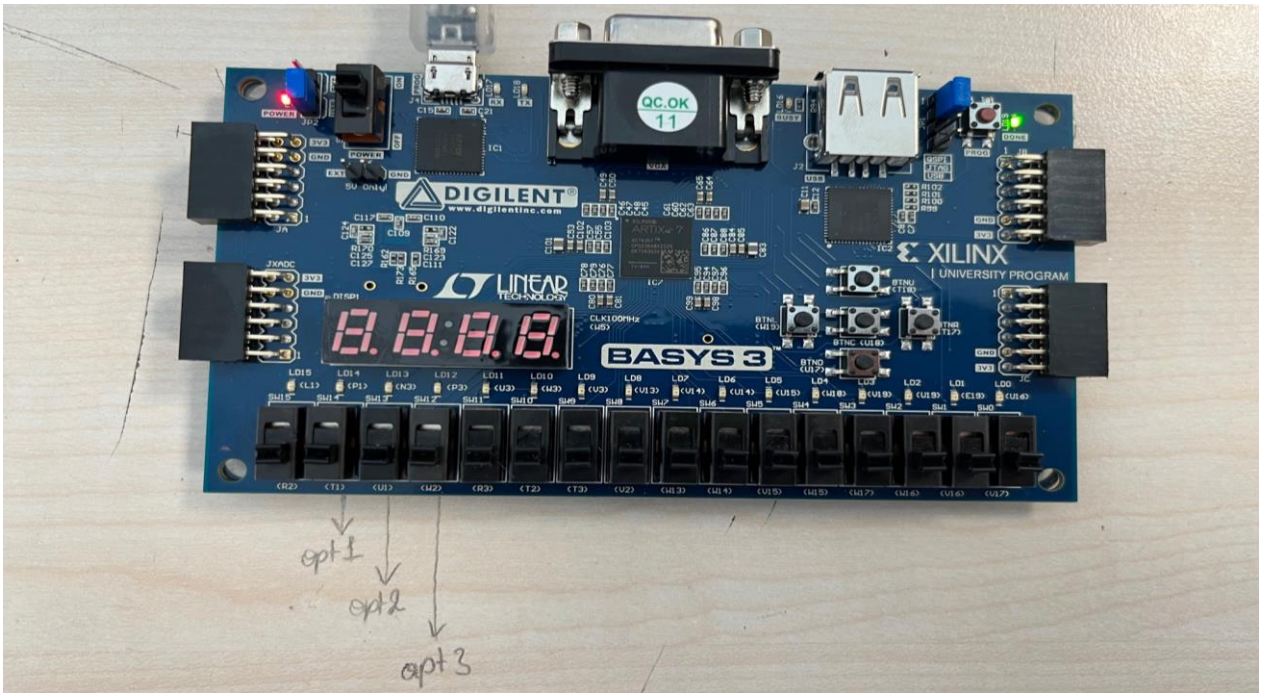

Figure 3: (opt1=0 , opt2=0 , opt3= 0, result = 0)

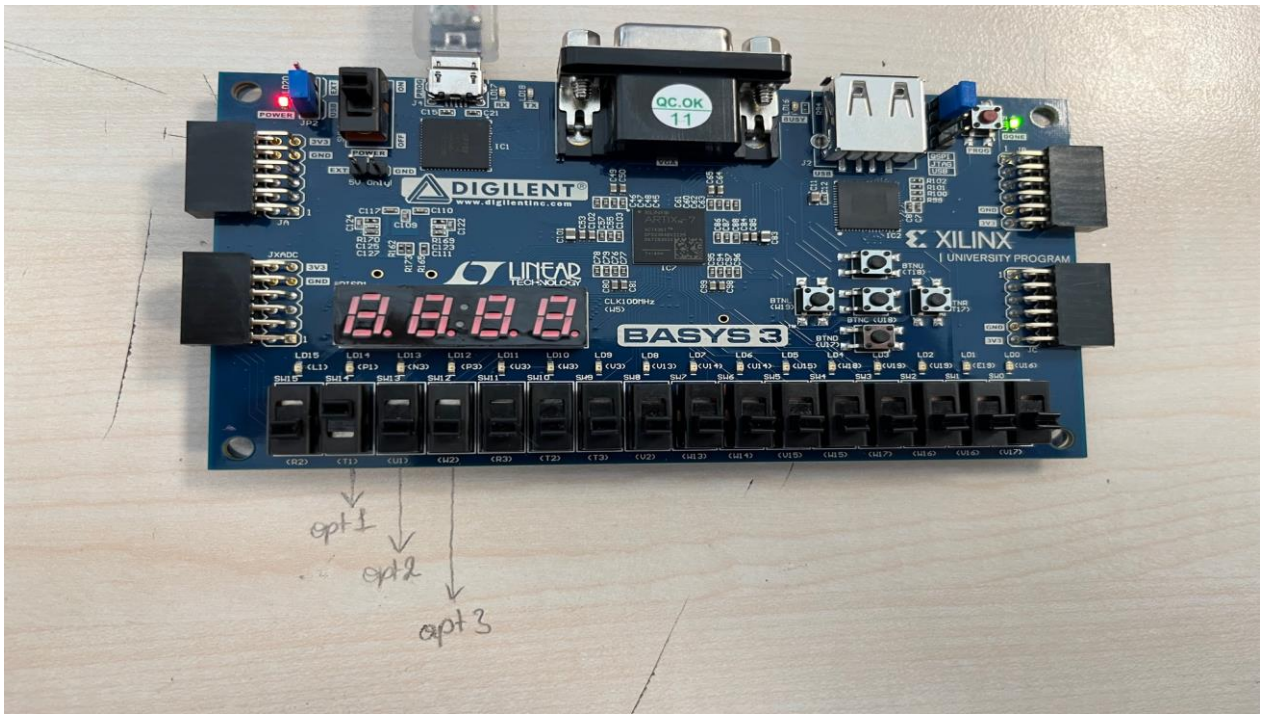

Figure 4 : (opt1=1 , opt2=0 , opt3= 0, result = 0)

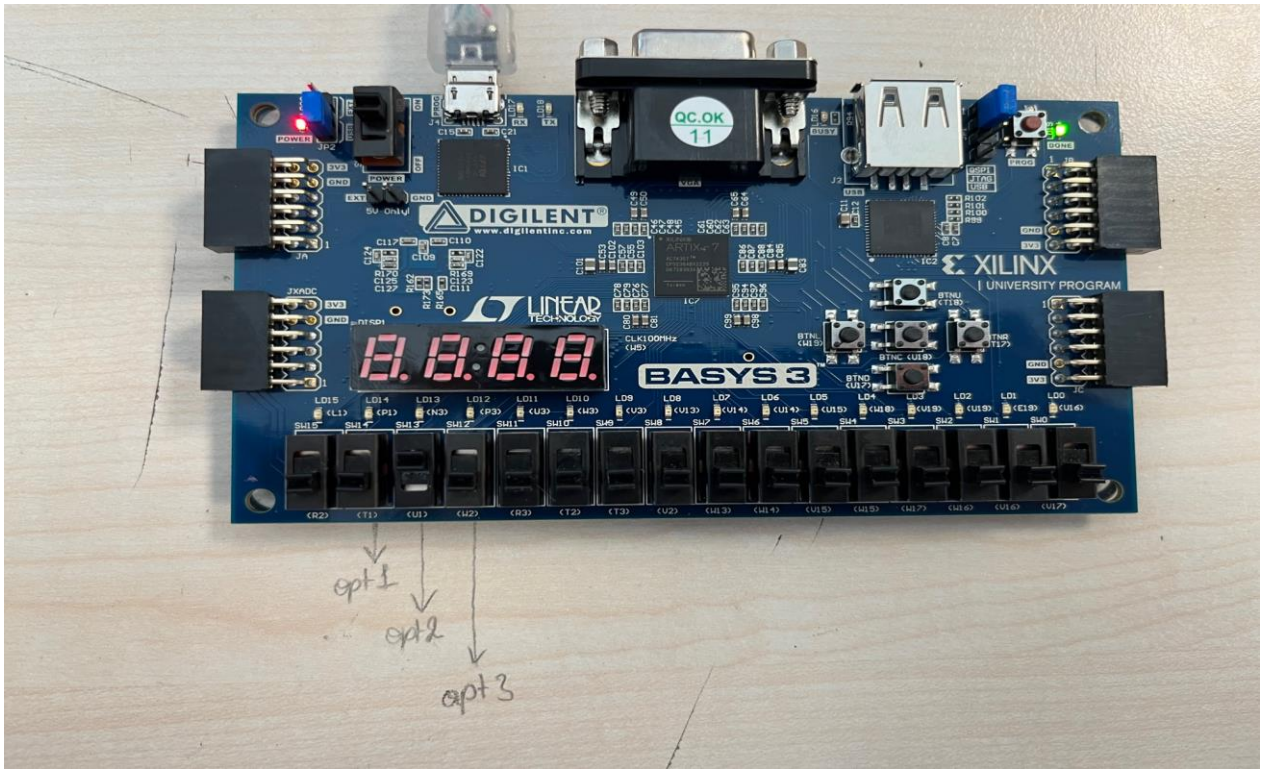

Figure 5: (opt1=0 , opt2=1 , opt3= 0, result = 0)

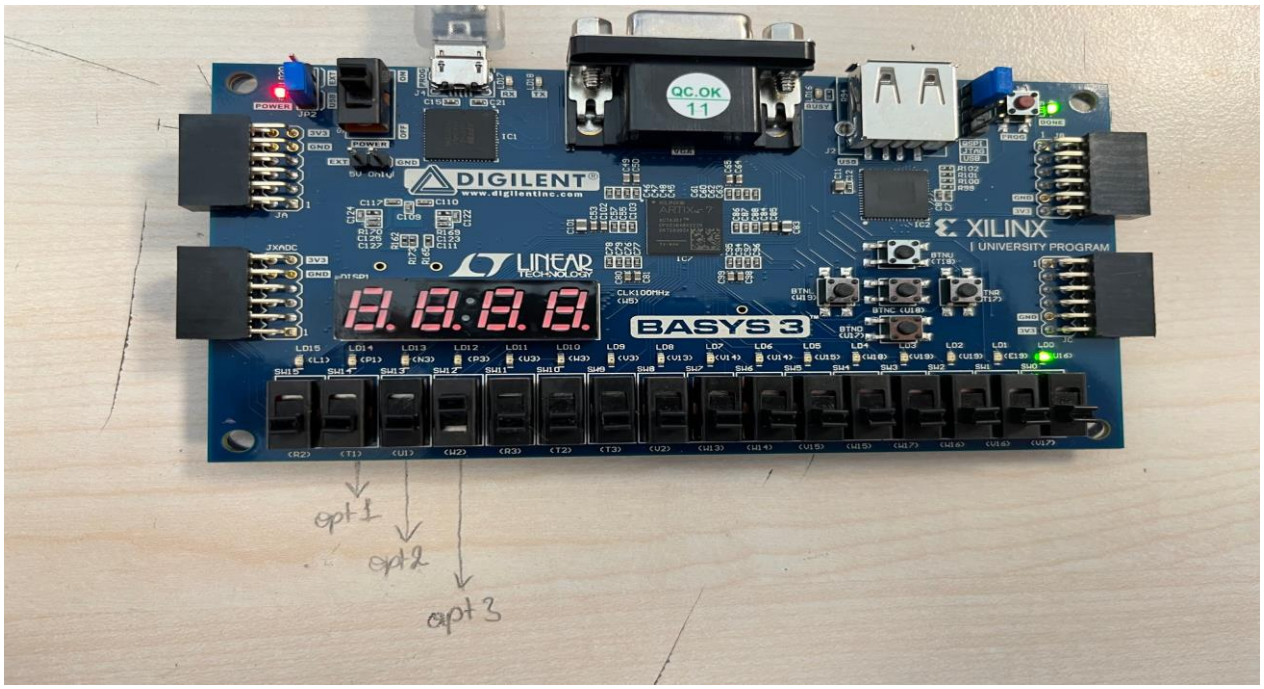

Figure 6: (opt1=0 , opt2=0 , opt3= 1, result = 1)

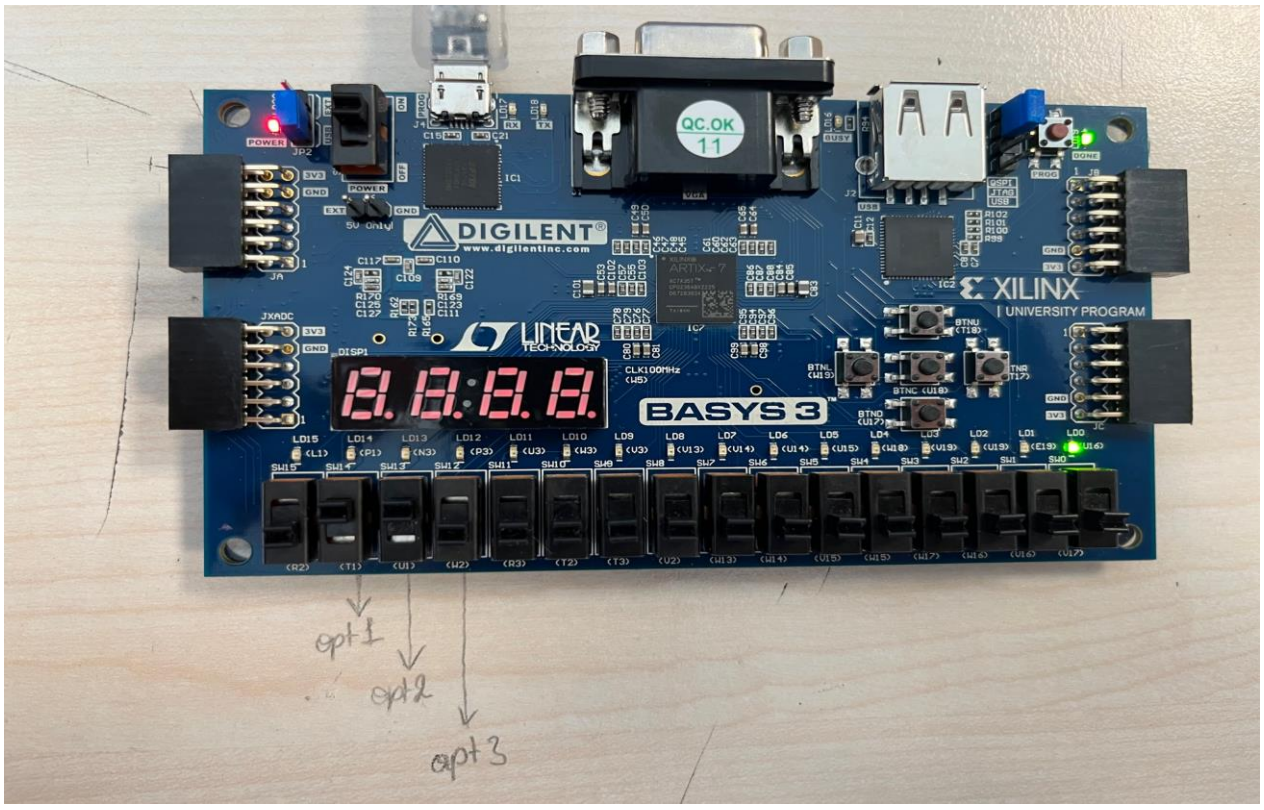

Figure 7: (opt1=1 , opt2=1 , opt3= 0, result = 1)

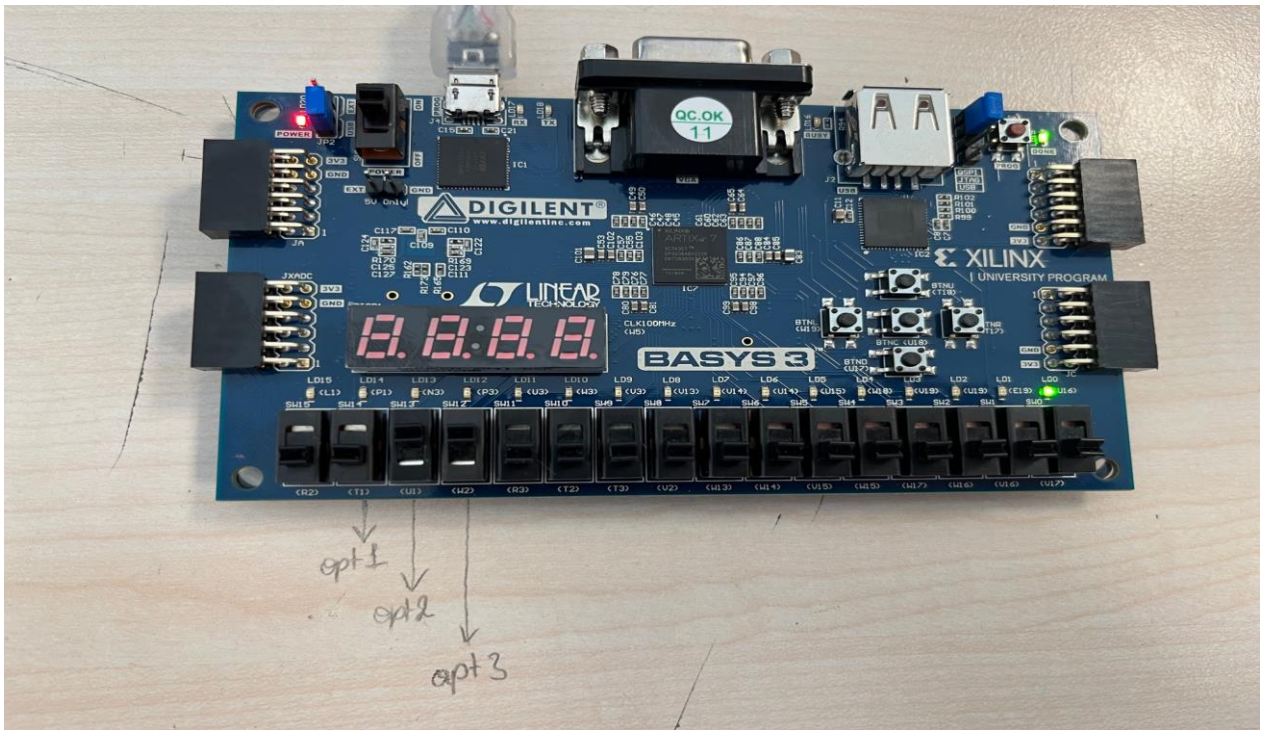

Figure 8: (opt1=0 , opt2=1 , opt3= 1, result = 1)

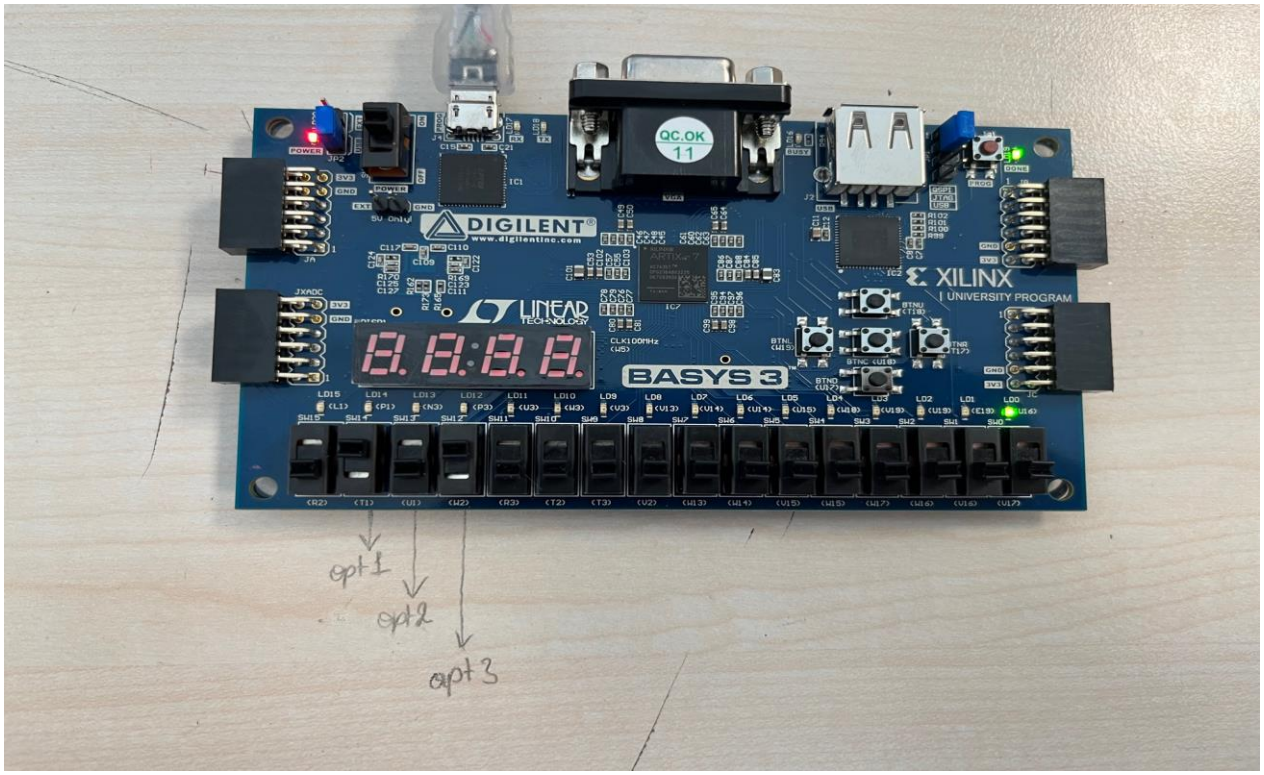

Figure 9: (opt1=1 , opt2=0 , opt3= 1, result = 1)

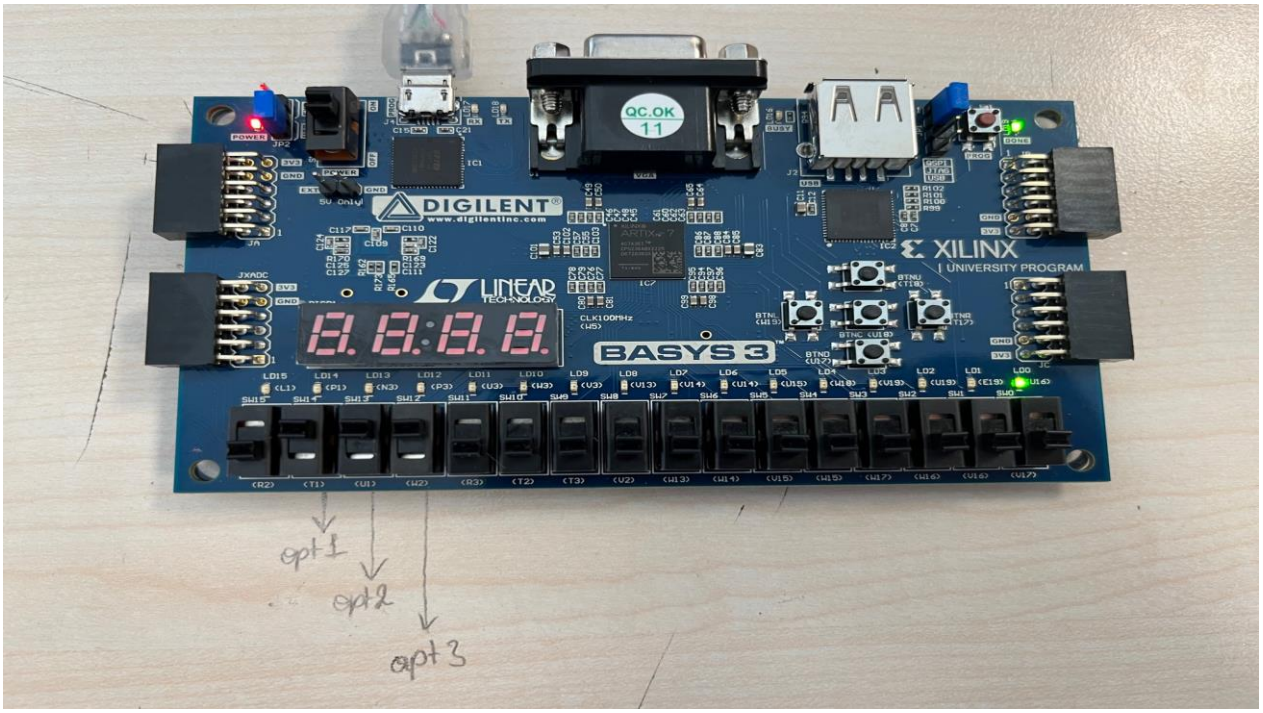

Figure 10: (opt1=1 , opt2=1 , opt3= 1, result = 1)

5. Conclusion

The experiment that was conducted significantly contributed to our understanding of Basys3's structural arrangement and the FPGA board. The goals of this lab include gaining programming abilities in VHDL, constructing bitstreams, programming Basys3, designing a simulation model, and executing synthesis. In my oppinion, this experiment was succesfull because the process ended without an error.

6.Appendices

VHDL Design Code

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

entity Lab01_design is
    Port ( opt1 : in STD_LOGIC;
           opt2 : in STD_LOGIC;
           opt3 : in STD_LOGIC;
           result : out STD_LOGIC);
end Lab01_design;

architecture Behavioral of Lab01_design is

begin
    result <= (opt1 AND opt2)OR opt3 ;

end Behavioral;
```

Testbench Code

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

entity Lab01_testbench is
end Lab01_testbench;

architecture Behavioral of Lab01_testbench is

    component Lab01_design
        port (opt1 : in std_logic;
              opt2 : in std_logic;
              opt3 : in std_logic;
              result : out std_logic);
    end component;

    signal opt1 : std_logic;
    signal opt2 : std_logic;
    signal opt3 : std_logic;
    signal result : std_logic;

begin
```

```
uut : Lab01_design
port map (opt1 => opt1,
        opt2 => opt2,
        opt3 => opt3,
        result => result);



stimulus : process
begin

    opt1 <= '0';
    opt2 <= '0';
    opt3 <= '0';

    wait for 100 ns ;

    opt1 <= '1';
    opt2 <= '0';
    opt3 <= '0';

    wait for 100 ns;

    opt1 <= '0';
    opt2 <= '1';
```

```
opt3 <= '0';

wait for 100 ns;

opt1 <= '0';
opt2 <= '0';
opt3 <= '1';

wait for 100 ns;

opt1 <= '1';
opt2 <= '1';
opt3 <= '0';

wait for 100 ns;

opt1 <= '0';
opt2 <= '1';
opt3 <= '1';

wait for 100 ns;

opt1 <= '1';
opt2 <= '0';
opt3 <= '1';
```

```
        wait for 100 ns;

        opt1 <= '1';
        opt2 <= '1';
        opt3 <= '1';

        wait for 100 ns;




        wait;
    end process;

end ;
```

Constraint Code

```
set_property PACKAGE_PIN W2 [get_ports {opt3}]
    set_property IOSTANDARD LVCMOS33 [get_ports {opt3}]
set_property PACKAGE_PIN U1 [get_ports {opt2}]
    set_property IOSTANDARD LVCMOS33 [get_ports {opt2}]
set_property PACKAGE_PIN T1 [get_ports {opt1}]
    set_property IOSTANDARD LVCMOS33 [get_ports {opt1}]
set_property PACKAGE_PIN U16 [get_ports {result}]
    set_property IOSTANDARD LVCMOS33 [get_ports {result}]
```