EEE 102 LAB 5- SEVEN SEGMENT DISPLAY

Name: Burak Gökkaya

ID: 22202209

Section: 001

Introduction

The purpose of this lab was to learn seven segment diplay with using BASYS3 and VHDL. For that, we used to Vivado and learnt that what is the meaning of persistence of vision. Also we answered some questions.

Questions

1) What is the internal clock frequency of Basys3?

Basys3's internal clock frequency is 100 MHz.

2) How can you create a slower clock signal from this one?

We can creates a slower clock signal with increasing a number of digits and make changes in the MSB

3) Can you create a clock with any arbitrary frequency lower than that of the internal clock? If not, which frequencies can you create?

If we want a fix frequency clock, only frequencys we can get are limited by $100MHz/2$^n.

Implementation

We are wanted to make seven segment display and for that we made research and learnt that how can we do. Hence, I made my research as a result I understand to mechanism of seven-segment display. After finishing code part, I implemented it successfully and got the schematic.

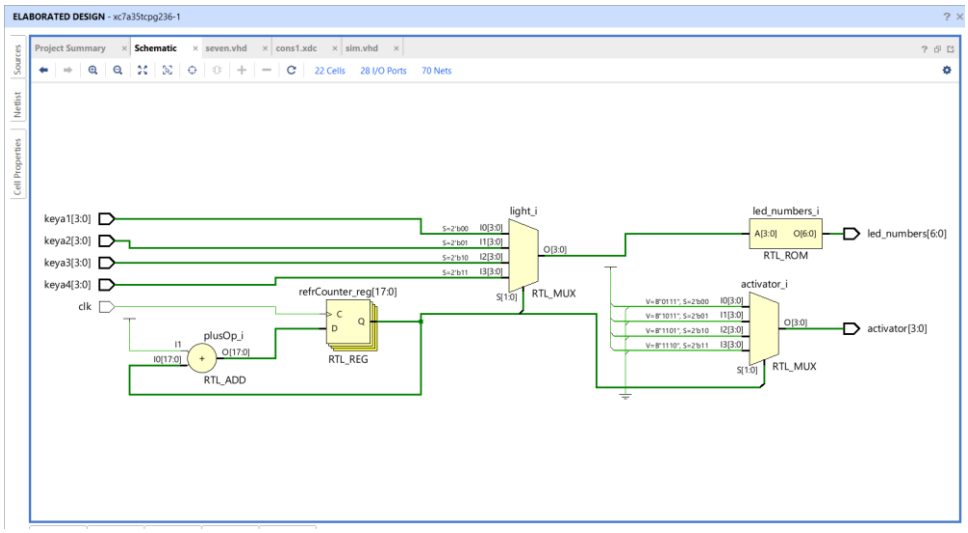

Figure 1. Schematic of seven-segment display

Then I wrote test bench code and run simulation.

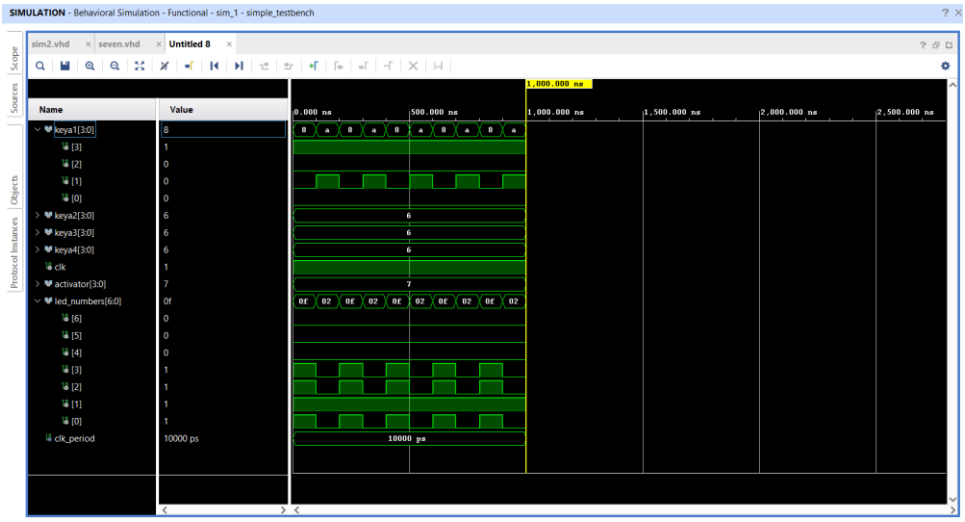

Figure 2. Simulation of seven-segment display

After successful tests, I implemented code to BASYS 3 and get the results which I wanted.

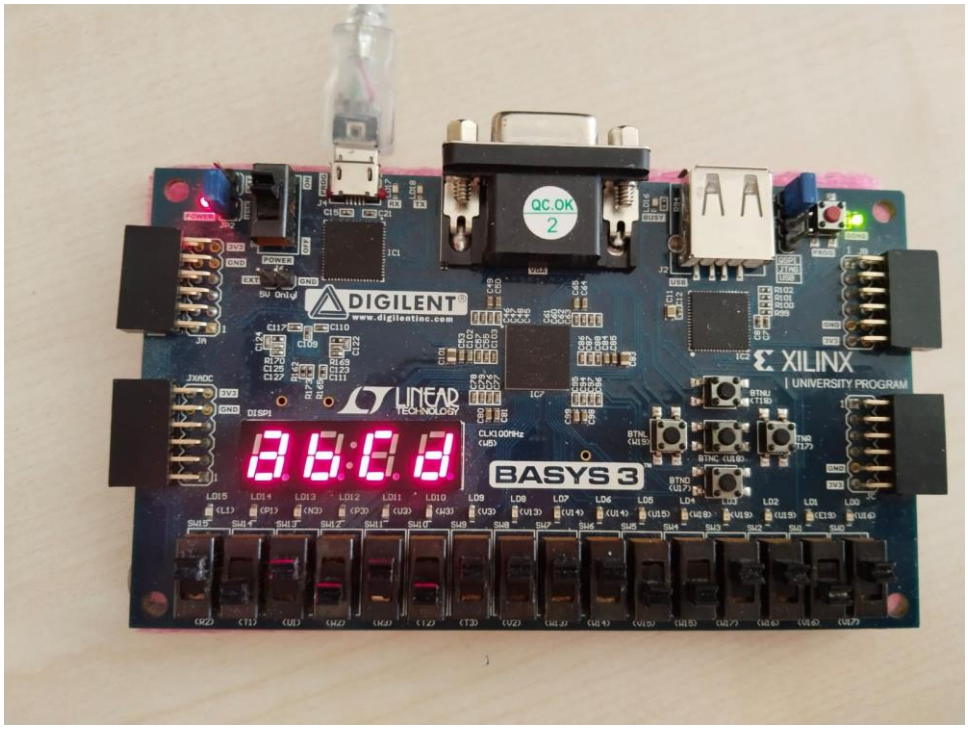

Figure 3. ABCD

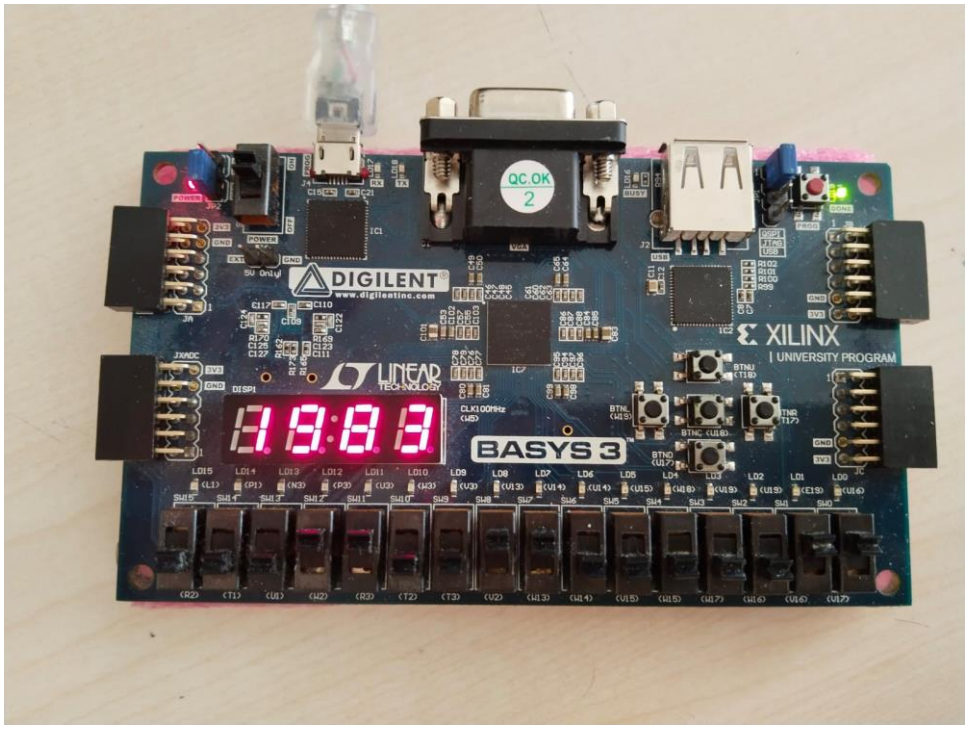

Figure 4. 1983

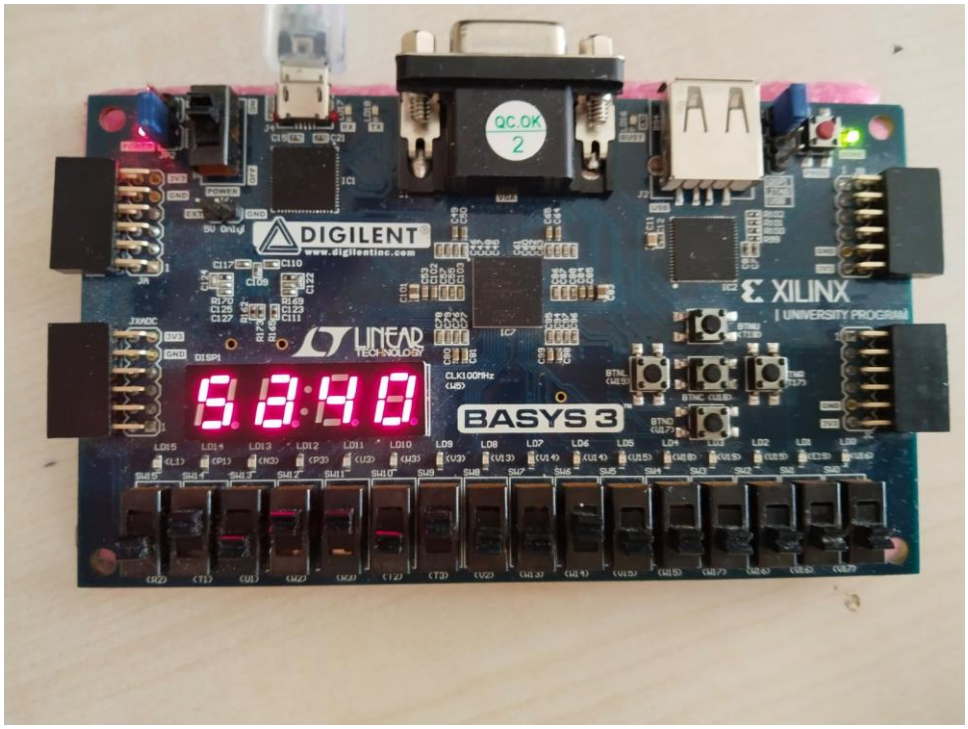

Figure 5. 5A40

Conclusion

In conclusion, this was very helpful laboratory for us. We learned the seven-segment display and also we improved for using BASYS3. I think, this project helped me to use BASYS 3 in a more efficient way. So it was successful experiment for me.

Appendix

#Main Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.std_logic_unsigned.all;
entity sevensegment_display_VHDL is
    Port (  clk : in STD_LOGIC;
            keya1 : in STD_LOGIC_VECTOR (3 downto 0);
            keya2 : in STD_LOGIC_VECTOR (3 downto 0);
            keya3 : in STD_LOGIC_VECTOR (3 downto 0);
            keya4 : in STD_LOGIC_VECTOR (3 downto 0);
            activator : out STD_LOGIC_VECTOR (3 downto 0);
            led_numbers : out STD_LOGIC_VECTOR (6 downto 0));
end sevensegment_display_VHDL;
architecture Behavioral of sevensegment_display_VHDL is
    signal light: STD_LOGIC_VECTOR (3 downto 0) := (others => '0');
    signal refrCounter: STD_LOGIC_VECTOR (17 downto 0):= (others => '0');
    signal LEDActivater: std_logic_vector(1 downto 0):= (others => '0');
begin
process(light)
begin
    case light is
        when "0000" => led_numbers <= "0000001";
        when "0001" => led_numbers <= "1001111";
        when "0010" => led_numbers <= "0010010";
        when "0011" => led_numbers <= "0000110";
        when "0100" => led_numbers <= "1001100";
        when "0101" => led_numbers <= "0100100";
        when "0110" => led_numbers <= "0100000";
        when "0111" => led_numbers <= "0001111";
        when "1000" => led_numbers <= "0000000";
```

```
                when "1001" => led_numbers <= "0000100";
                when "1010" => led_numbers <= "0000010";
                when "1011" => led_numbers <= "1100000";
                when "1100" => led_numbers <= "0110001";
                when "1101" => led_numbers <= "1000010";
                when "1110" => led_numbers <= "0110000";
                when "1111" => led_numbers <= "0111000";
                when others => led_numbers <= "1111111";
        end case;
end process;
process(clk)
begin
if(rising_edge(clk)) then
        refrCounter <= refrCounter + 1;
end if;
end process;
LEDActivater <= refrCounter(17 downto 16);
        process(LEDActivater)
        begin
                case LEDActivater is
                        when "00" =>
                        activator <= "0111";
                        light <= keya1;
                        when "01" =>
                        activator <= "1011";
                        light <= keya2;
                        when "10" =>
                        activator <= "1101";
                        light <= keya3;
                        when "11" =>
                        activator <= "1110";
```

```
                light <= keya4;
                when others =>
                activator <= "1111";
                light <= keya4;
            end case;
        end process;
end Behavioral;
```

SIMULATION CODE

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
ENTITY simple_testbench IS
END simple_testbench;
ARCHITECTURE behavior OF simple_testbench IS
-- Component Declaration for the Unit Under Test (UUT)
COMPONENT simple
PORT(
        clk : in STD_LOGIC;
        keya1 : in STD_LOGIC_VECTOR (3 downto 0);
        keya2 : in STD_LOGIC_VECTOR (3 downto 0);
        keya3 : in STD_LOGIC_VECTOR (3 downto 0);
        keya4 : in STD_LOGIC_VECTOR (3 downto 0);
        activator : out STD_LOGIC_VECTOR (3 downto 0);
        led_numbers : out STD_LOGIC_VECTOR (6 downto 0)
);
END COMPONENT;
--Inputs
   signal keya1 : STD_LOGIC_VECTOR (3 downto 0) := "0000";
   signal keya2 : STD_LOGIC_VECTOR (3 downto 0) := "0000";
   signal keya3 : STD_LOGIC_VECTOR (3 downto 0) := "0000";
   signal keya4 : STD_LOGIC_VECTOR (3 downto 0) := "0000";
```

```
   signal clk : STD_LOGIC := '1';
   constant clk_period : time := 10 ns;
--Outputs
   signal activator : STD_LOGIC_VECTOR (3 downto 0);
   signal led_numbers : STD_LOGIC_VECTOR (6 downto 0);
BEGIN
uut: simple port map (
          clk => clk,
          keya1 => keya1,
          keya2 => keya2,
          keya3 => keya3,
          keya4 => keya4,
          activator => activator,
          led_numbers => led_numbers
       );
stim_proc: process
begin
       activator <= "0111";
       keya1 <= "1000";
       keya2 <= "0110";
       keya3 <= "0110";
       keya4 <= "0110";
       led_numbers <= led_numbers;
       wait for 100 ns;
       keya1 <= "1010";
       keya2 <= "0110";
       keya3 <= "0110";
       keya4 <= "0110";
       activator <= "0111";
       led_numbers <= led_numbers;
       wait for 100 ns;
```

end process;

END;

CONSTRAINT

set_property PACKAGE_PIN V17 [get_ports {keya4[0]}]

set_property IOSTANDARD LVCMOS33 [get_ports {keya4[0]}]

set_property PACKAGE_PIN V16 [get_ports {keya4[1]}]

set_property IOSTANDARD LVCMOS33 [get_ports {keya4[1]}]

set_property PACKAGE_PIN W16 [get_ports {keya4[2]}]

set_property IOSTANDARD LVCMOS33 [get_ports {keya4[2]}]

set_property PACKAGE_PIN W17 [get_ports {keya4[3]}]

set_property IOSTANDARD LVCMOS33 [get_ports {keya4[3]}]

set_property PACKAGE_PIN W15 [get_ports {keya3[0]}]

set_property IOSTANDARD LVCMOS33 [get_ports {keya3[0]}]

set_property PACKAGE_PIN V15 [get_ports {keya3[1]}]

set_property IOSTANDARD LVCMOS33 [get_ports {keya3[1]}]

set_property PACKAGE_PIN W14 [get_ports {keya3[2]}]

set_property IOSTANDARD LVCMOS33 [get_ports {keya3[2]}]

set_property PACKAGE_PIN W13 [get_ports {keya3[3]}]

set_property IOSTANDARD LVCMOS33 [get_ports {keya3[3]}]

set_property PACKAGE_PIN V2 [get_ports {keya2[0]}]

set_property IOSTANDARD LVCMOS33 [get_ports {keya2[0]}]

set_property PACKAGE_PIN T3 [get_ports {keya2[1]}]

set_property IOSTANDARD LVCMOS33 [get_ports {keya2[1]}]

set_property PACKAGE_PIN T2 [get_ports {keya2[2]}]

set_property IOSTANDARD LVCMOS33 [get_ports {keya2[2]}]

set_property PACKAGE_PIN R3 [get_ports {keya2[3]}]

set_property IOSTANDARD LVCMOS33 [get_ports {keya2[3]}]

set_property PACKAGE_PIN W2 [get_ports {keya1[0]}]

set_property IOSTANDARD LVCMOS33 [get_ports {keya1[0]}]

set_property PACKAGE_PIN U1 [get_ports {keya1[1]}]

set_property IOSTANDARD LVCMOS33 [get_ports {keya1[1]}]

set_property PACKAGE_PIN T1 [get_ports {keya1[2]}]

set_property IOSTANDARD LVCMOS33 [get_ports {keya1[2]}]

set_property PACKAGE_PIN R2 [get_ports {keya1[3]}]

set_property IOSTANDARD LVCMOS33 [get_ports {keya1[3]}]

 # Clock signal

set_property PACKAGE_PIN W5 [get_ports clk]

set_property IOSTANDARD LVCMOS33 [get_ports clk]

#seven-segment LED display

set_property PACKAGE_PIN W7 [get_ports {led_numbers[6]}]

set_property IOSTANDARD LVCMOS33 [get_ports {led_numbers[6]}]

set_property PACKAGE_PIN W6 [get_ports {led_numbers[5]}]

set_property IOSTANDARD LVCMOS33 [get_ports {led_numbers[5]}]

set_property PACKAGE_PIN U8 [get_ports {led_numbers[4]}]

set_property IOSTANDARD LVCMOS33 [get_ports {led_numbers[4]}]

set_property PACKAGE_PIN V8 [get_ports {led_numbers[3]}]

set_property IOSTANDARD LVCMOS33 [get_ports {led_numbers[3]}]

set_property PACKAGE_PIN U5 [get_ports {led_numbers[2]}]

set_property IOSTANDARD LVCMOS33 [get_ports {led_numbers[2]}]

set_property PACKAGE_PIN V5 [get_ports {led_numbers[1]}]

set_property IOSTANDARD LVCMOS33 [get_ports {led_numbers[1]}]

set_property PACKAGE_PIN U7 [get_ports {led_numbers[0]}]

set_property IOSTANDARD LVCMOS33 [get_ports {led_numbers[0]}]

set_property PACKAGE_PIN U2 [get_ports {activator[0]}]

set_property IOSTANDARD LVCMOS33 [get_ports {activator[0]}]

set_property PACKAGE_PIN U4 [get_ports {activator[1]}]

set_property IOSTANDARD LVCMOS33 [get_ports {activator[1]}]

set_property PACKAGE_PIN V4 [get_ports {activator[2]}]

set_property IOSTANDARD LVCMOS33 [get_ports {activator[2]}]

set_property PACKAGE_PIN W4 [get_ports {activator[3]}]

set_property IOSTANDARD LVCMOS33 [get_ports {activator[3]}]