Lab 4 Report

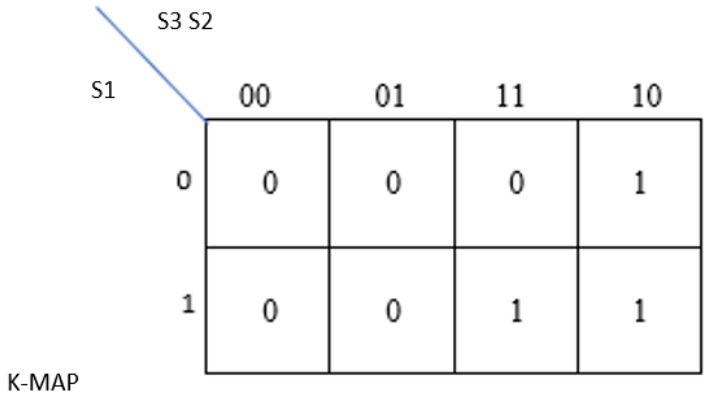
Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2

- ## Design Methodology

I establish this labs purpose with the scenario that I used in Lab02. In the scenario there is a Supervisor (S1), Rookie(S2), Cashier(S3) and the goal is getting inside the bank vault. Only the supervisor has an access the bank vault so if someone else wants to get in the vault they must be together with the supervisor. Also, rookie always must be with the cashier in order to be monitored. If rookie is in the vault cashier must be in the vault as well as the supervisor. At the seven-segment display S1 S2 S3 L4 will be displayed with '1' s and '0' s according to their values which means if the value is 0 then LED_on ="0000001", if it is 1 then LED_on = "1001111".

- **Differences from Preliminary:**
- I added clock module and I am using it to display 4 different seven-segments at the same time on Basys3
- About Activator and LED_bin I changed the locations and the way I used them in my code.
- I added 4 LEDs to visualize inputs and outputs values.

- ## Results



K-MAP

Lab 4 Report

Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2

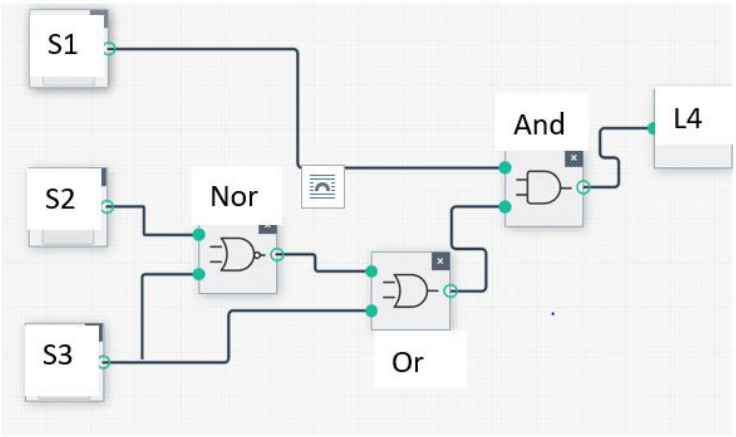
Truth Table

S	C	R	H(S,C,R)	Minterms
0	0	0	0	m_0
0	0	1	0	m_1
0	1	0	0	m_2
0	1	1	0	m_3
1	0	0	1	m_4
1	0	1	0	m_5
1	1	0	1	m_6
1	1	1	1	m_7

Sum of products(minterms) $= \sum m(4,6,7) = m_4 + m_6 + m_7 = H(S,C,R)$

$$\mathrm{H(S,C,R)} = S.\bar{C}.\bar{R} + S.C.\bar{R} + S.C.R$$
$$= S.\bar{C}.\bar{R} + S.C.(\bar{R} + R)$$
$$= S.\bar{C}.\bar{R} + S.C$$
$$= S.(\bar{C}.\bar{R} + C)$$
$$\mathrm{H(S,C,R)} = S.(\bar{C}.\bar{R} + C)$$

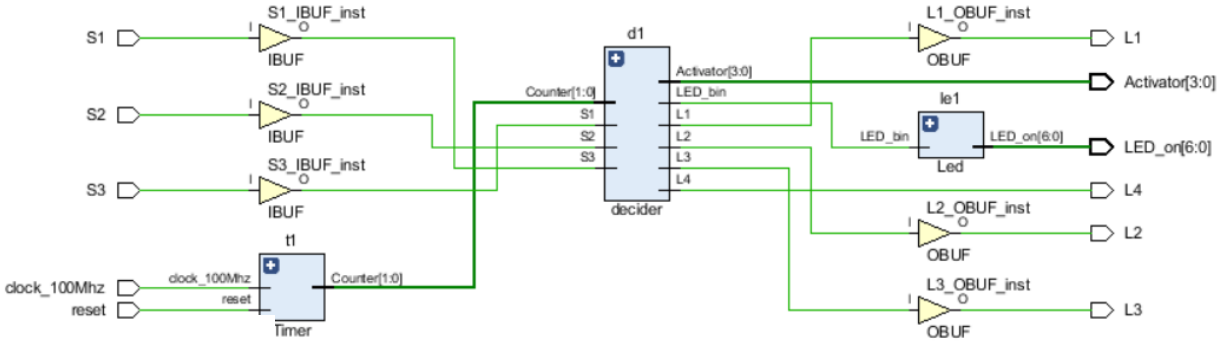
Logic Schematic Diagram



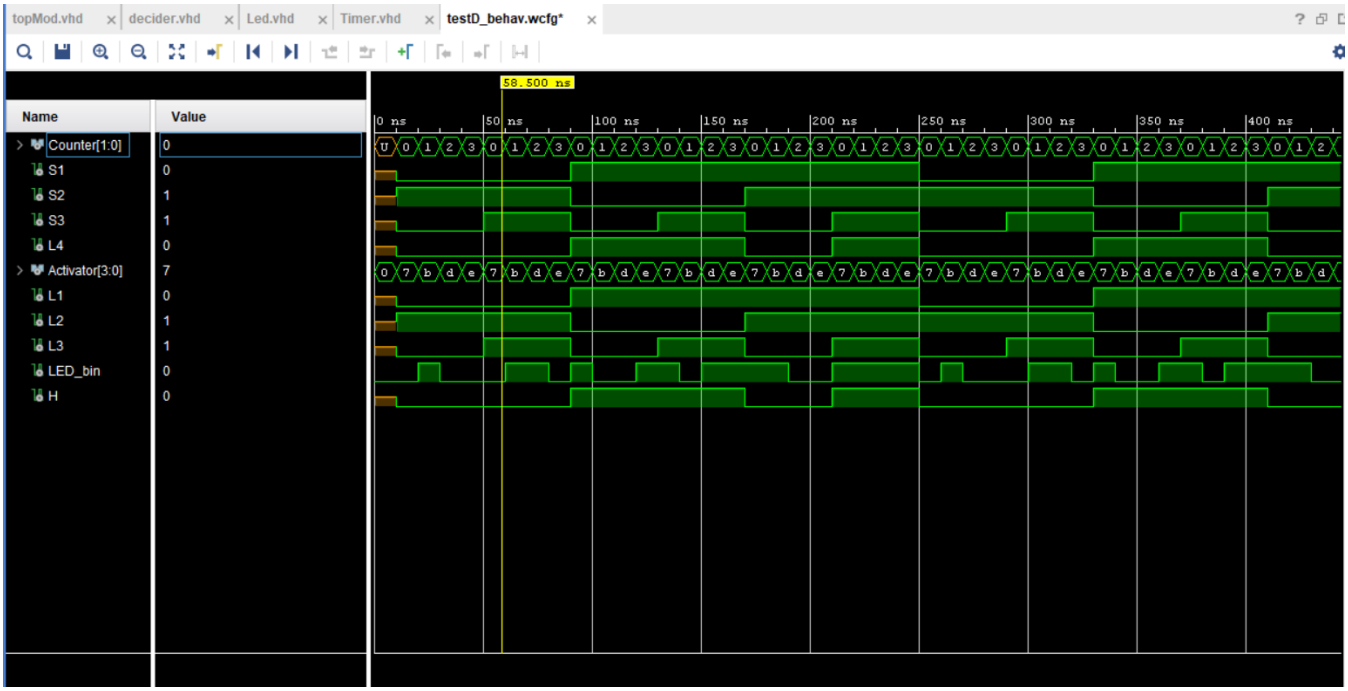
Lab 4 Report

Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2

Schematic

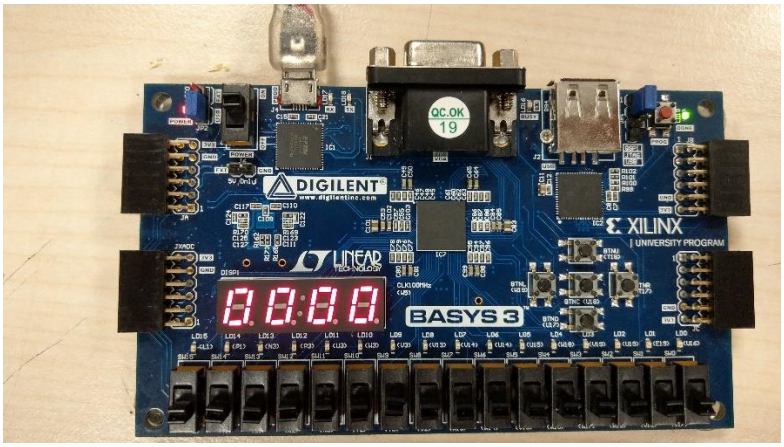


Test bench of decider

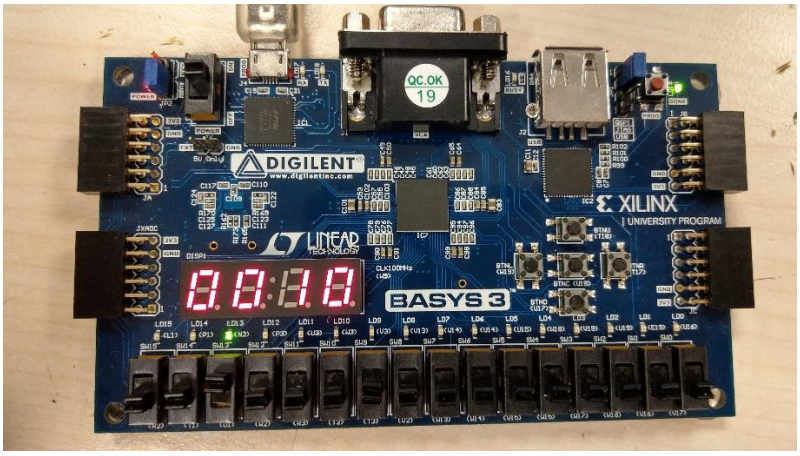


Lab 4 Report

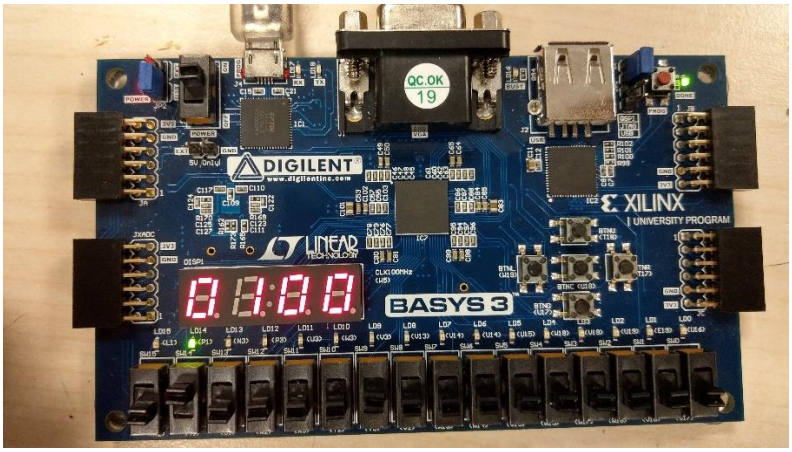
Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2



S1:0 S2:0 S3:0 L4:0



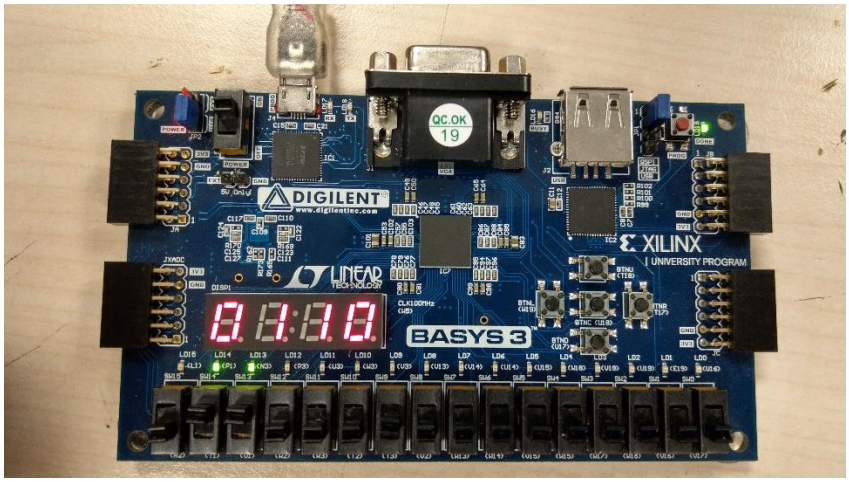
S1:0 S2:0 S3:1 L4:0



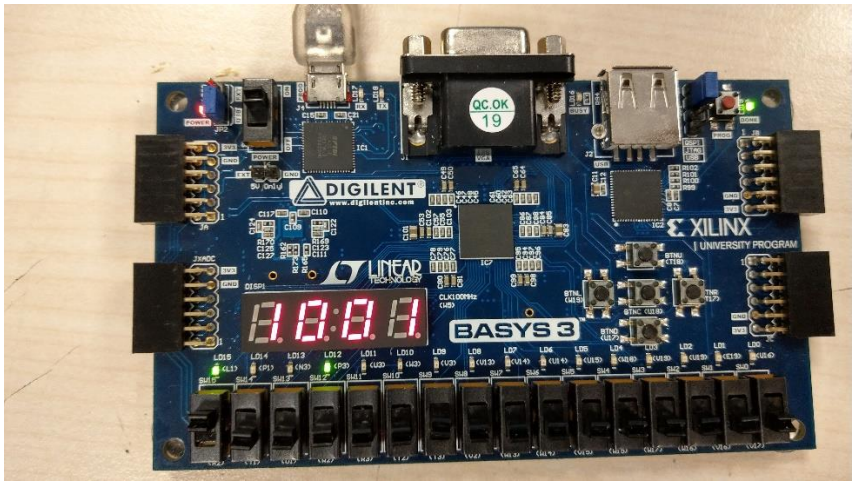
S1:0 S2:1 S3:0 L4:0

Lab 4 Report

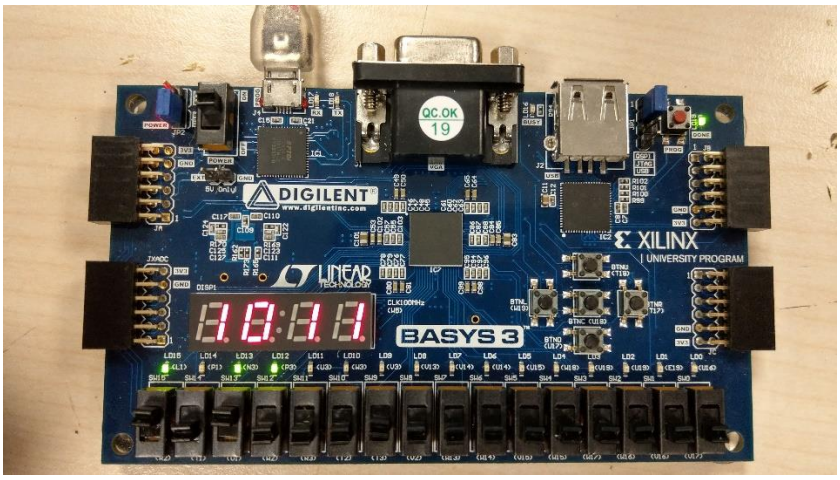
Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2



S1:0 S2:1 S3:1 L4:0



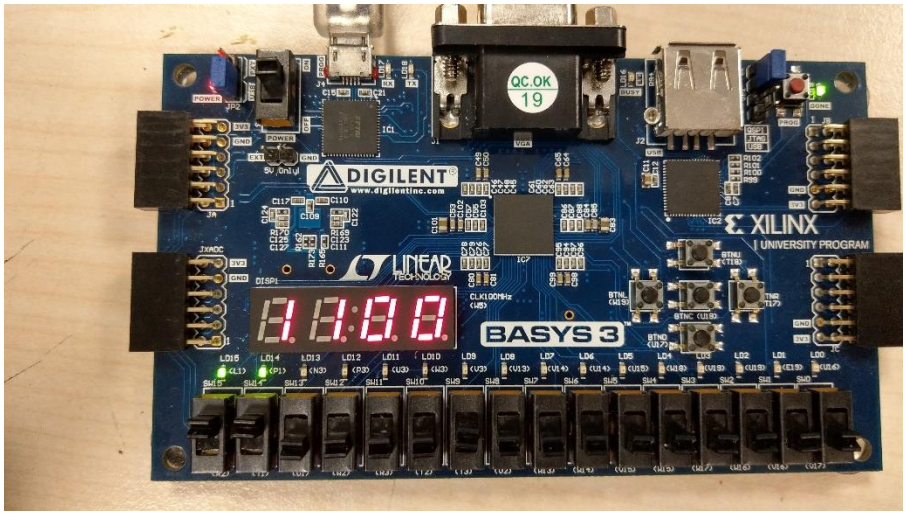
S1:1 S2:0 S3:0 L4:1



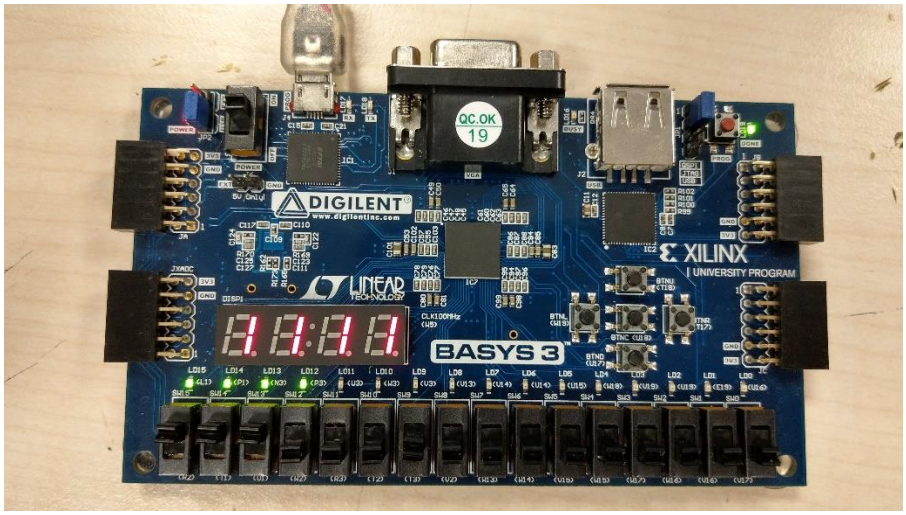
S1:1 S2:0 S3:1 L4:1

Lab 4 Report

Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2



S1:1 S2:1 S3:0 L4:0



S1:1 S2:1 S3:1 L4:1

Lab 4 Report

Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2

- # Conclusion

I have learnt the differences between variable types like in, out, inout, signals… and places where can we use them, or which one must use. I have faced with some errors like "usf-xsim-62 'elaborate' step failed with error(s)" and I learnt to track the log files and find the errors. I have learnt whether I included all possible values I should also add "others" case in cases. I have learnt how to use top and sub modules and how to write test bench in better way. I also have learnt how to use seven-segment display in which how to display numbers and letters, how to order them and showing four seven segments at the same time using clock on the Basys3.

- # Appendices
  - VHDL:

TopModule:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity topMod is
    Port ( clock_100Mhz : in STD_LOGIC;
           reset : in STD_LOGIC;
           S3,S2,S1 : in STD_LOGIC;
           L4,L3,L2,L1 : out STD_LOGIC;
           Activator : out STD_LOGIC_VECTOR (3 downto 0);
           LED_on : out STD_LOGIC_VECTOR (6 downto 0));
end topMod;

architecture Behavioral of topMod is
signal Counter_temp:  std_logic_vector(1 downto 0);
signal BIN_LED: std_logic;


component decider is
PORT
(
           Counter : in std_logic_vector(1 downto 0);
           S3,S2,S1 : in STD_LOGIC;
           Activator : out std_logic_vector(3 downto 0);
           L4,L3,L2,L1 : out STD_LOGIC;
           LED_bin : out STD_LOGIC);
```

Lab 4 Report

Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2

```
end component;

component Led is
PORT
        (LED_bin : in STD_LOGIC;
         LED_on : out STD_LOGIC_VECTOR (6 downto 0));
end component;


component Timer is
PORT
        (clock_100Mhz : in STD_LOGIC;
         reset : in STD_LOGIC;
         Counter: out std_logic_vector(1 downto 0));
end component;
begin

d1 : decider PORT MAP (Counter_temp,S1,S2,S3,Activator,L4,L3,L2,L1,BIN_LED);
le1: Led  PORT MAP (BIN_LED,LED_on);
t1 : Timer PORT MAP ( clock_100Mhz,reset,Counter_temp);

end Behavioral;
```

Decider module:

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

entity decider is
   Port ( Counter : in std_logic_vector(1 downto 0);
          S1 : in STD_LOGIC;
          S2 : in STD_LOGIC;
          S3 : in STD_LOGIC;
          Activator : out std_logic_vector(3 downto 0);
          L1 : out STD_LOGIC;
          L2 : out STD_LOGIC;
          L3 : out STD_LOGIC;
          L4 : out STD_LOGIC;
          LED_bin : out STD_LOGIC);
end decider;

architecture Behavioral of decider is
signal H: STD_LOGIC;

begin
 H  <= S1 and (( not S2 and not S3) or S3);
```

Lab 4 Report

Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2

```
 L1 <= S1;
 L2 <= S2;
 L3 <= S3;
L4 <= H;

process(Counter,S1)
begin
   case Counter is
   when "00" =>
      Activator <= "0111";
      LED_bin <= S1;
   when "01" =>
      Activator <= "1011";
      LED_bin <= S2;
   when "10" =>
      Activator <= "1101";
      LED_bin <= S3;
   when "11" =>
      Activator <= "1110";
      LED_bin <= H;
    when others =>
     Activator <= "0000";
       LED_bin <= '0';
   end case;
end process;

end Behavioral;
```

Led Module:

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

entity Led is
   Port ( LED_bin : in STD_LOGIC;
          LED_on : out STD_LOGIC_VECTOR (6 downto 0));
end Led;

architecture Behavioral of Led is

begin

process(LED_bin)
begin
   case LED_bin is
   when '0' => LED_on <= "0000001";
```

Lab 4 Report

Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2

```
   when '1' => LED_on <= "1001111";
   when others => LED_on <= "0000001";

   end case;
end process;

end Behavioral;
```

Timer Module:

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

entity Timer is
   Port ( clock_100Mhz : in STD_LOGIC;
          reset : in STD_LOGIC;
          Counter: out std_logic_vector(1 downto 0));
end Timer;

architecture Behavioral of Timer is
signal refresh_counter: STD_LOGIC_VECTOR (19 downto 0);
begin

process(clock_100Mhz,reset)
begin
   if(reset='1') then
      refresh_counter <= (others => '0');
   elsif(rising_edge(clock_100Mhz)) then
      refresh_counter <= refresh_counter + 1;
   end if;
end process;
 Counter <= refresh_counter(19 downto 18);

end Behavioral;
```

TESTBENCHES:

TESTD:

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

entity testD is
end testD;
```

Lab 4 Report

Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2

```
architecture Behavioral of testD is
component decider is
 Port ( Counter : in std_logic_vector(1 downto 0);
          S1 : in STD_LOGIC;
          S2 : in STD_LOGIC;
          S3 : in STD_LOGIC;
         Activator : out std_logic_vector(3 downto 0);
         L1 : out STD_LOGIC;
          L2 : out STD_LOGIC;
         L3 : out STD_LOGIC;
        L4:out STD_LOGIC;
        LED_bin : out STD_LOGIC
);
end component;

signal Counter :  std_logic_vector(1 downto 0);
signal S1 : STD_LOGIC;
signal S2 : STD_LOGIC;
signal S3 : STD_LOGIC;
signal L4 :  STD_LOGIC;
signal Activator :  std_logic_vector(3 downto 0) := "0000";
signal L1 :  STD_LOGIC;
signal L2 :  STD_LOGIC;
signal L3 :  STD_LOGIC;
signal LED_bin :  STD_LOGIC := '0';

begin
UUT: decider PORT MAP(
Counter => Counter,
S1 => S1,
S2 => S2,
S3 => S3,
L4 => L4,
Activator => Activator,
 L1 => L1,
 L2 => L2,
 L3  => L3,
LED_bin => LED_bin
--Activator => Activator
);

Bench1: PROCESS
begin
--wait for 100 ns;
--Counter <= "00";
--S1 <= '0';
```

Lab 4 Report

Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2

```
--S2 <= '0';
--S3 <= '0';
--wait for 100 ns;
--Counter <= "01";
--S1 <= '0';
--S2 <= '0';
--S3 <= '0';
--wait for 100 ns;
--Counter <= "10";
--S1 <= '0';
--S2 <= '0';
--S3 <= '0';wait for 100 ns;
--Counter <= "10";
--S1 <= '0';
--S2 <= '0';
--S3 <= '0';
----------------
--wait for 100 ns;
--Counter <= "00";
--S1 <= '0';
--S2 <= '0';
--S3 <= '1';
--wait for 100 ns;
--Counter <= "01";
--S1 <= '0';
--S2 <= '0';
--S3 <= '1';
--wait for 100 ns;
--Counter <= "10";
--S1 <= '0';
--S2 <= '0';
--S3 <= '1';
--wait for 100 ns;
--Counter <= "11";
--S1 <= '0';
--S2 <= '0';
--S3 <= '1';

-------------
wait for 10 ns;
Counter <= "00";
S1 <= '0';
S2 <= '1';
S3 <= '0';
wait for 10 ns;
Counter <= "01";
```

Lab 4 Report

Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2

```
S1 <= '0';
S2 <= '1';
S3 <= '0';
wait for 10 ns;
Counter <= "10";
S1 <= '0';
S2 <= '1';
S3 <= '0';
wait for 10 ns;
Counter <= "11";
S1 <= '0';
S2 <= '1';
S3 <= '0';
-------------
wait for 10 ns;
Counter <= "00";
S1 <= '0';
S2 <= '1';
S3 <= '1';
wait for 10 ns;
Counter <= "01";
S1 <= '0';
S2 <= '1';
S3 <= '1';
wait for 10 ns;
Counter <= "10";
S1 <= '0';
S2 <= '1';
S3 <= '1';
wait for 10 ns;
Counter <= "11";
S1 <= '0';
S2 <= '1';
S3 <= '1';

------------
wait for 10 ns;
Counter <= "00";
S1 <= '1';
S2 <= '0';
S3 <= '0';
wait for 10 ns;
Counter <= "01";
S1 <= '1';
S2 <= '0';
S3 <= '0';
```

Lab 4 Report

Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2

```
wait for 10 ns;
Counter <= "10";
S1 <= '1';
S2 <= '0';
S3 <= '0';
wait for 10 ns;
Counter <= "11";
S1 <= '1';
S2 <= '0';
S3 <= '0';
--------------
wait for 10 ns;
Counter <= "00";
S1 <= '1';
S2 <= '0';
S3 <= '1';
wait for 10 ns;
Counter <= "01";
S1 <= '1';
S2 <= '0';
S3 <= '1';
wait for 10 ns;
Counter <= "10";
S1 <= '1';
S2 <= '0';
S3 <= '1';
wait for 10 ns;
Counter <= "11";
S1 <= '1';
S2 <= '0';
S3 <= '1';
--------------------
wait for 10 ns;
Counter <= "00";
S1 <= '1';
S2 <= '1';
S3 <= '0';
wait for 10 ns;
Counter <= "01";
S1 <= '1';
S2 <= '1';
S3 <= '0';
wait for 10 ns;
Counter <= "10";
S1 <= '1';
S2 <= '1';
```

Lab 4 Report

Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2

```
S3 <= '0';
wait for 10 ns;
Counter <= "11";
S1 <= '1';
S2 <= '1';
S3 <= '0';
-----------------
wait for 10 ns;
Counter <= "00";
S1 <= '1';
S2 <= '1';
S3 <= '1';
wait for 10 ns;
Counter <= "01";
S1 <= '1';
S2 <= '1';
S3 <= '1';
wait for 10 ns;
Counter <= "10";
S1 <= '1';
S2 <= '1';
S3 <= '1';
wait for 10 ns;
Counter <= "11";
S1 <= '1';
S2 <= '1';
S3 <= '1';
End process;
end Behavioral;

TESTLED:
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity testLed is
end testLed;

architecture Behavioral of testLed is
component Led is
   Port (
         LED_bin : in STD_LOGIC;
         LED_on : out STD_LOGIC_VECTOR (6 downto 0));
  end component;
  signal LED_bin :  STD_LOGIC;
  signal   LED_on :  STD_LOGIC_VECTOR (6 downto 0):= b"0000000" ;
begin
UUT: Led PORT MAP(
```

Lab 4 Report

Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2

```
LED_bin => LED_bin,
LED_on=> LED_on);

Bench1: PROCESS
begin
wait for 10 ns;
LED_bin<='0';
wait for 10 ns;
LED_bin<='1';
end PROCESS;
end Behavioral;

TEST_TIME:
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity testTime is
end testTime;
architecture Behavioral of testTime is
component Timer is
 Port ( clock_100Mhz : in STD_LOGIC;
        reset : in STD_LOGIC;
        Counter: out std_logic_vector(1 downto 0));
end component;
signal clock_100Mhz :  STD_LOGIC;
signal reset :  STD_LOGIC;
signal Counter:  std_logic_vector(1 downto 0);
begin
UUT: Timer PORT MAP(
clock_100Mhz => clock_100Mhz,
reset => reset,
Counter => Counter
);

Bench2: PROCESS
begin
wait for 50 ns;
clock_100Mhz <= '0';
reset <= '0';
wait for 50 ns;
clock_100Mhz <= '0';
reset <= '1';
wait for 50 ns;
clock_100Mhz <= '1';
reset <= '0';
wait for 50 ns;
clock_100Mhz <= '1';
```

Lab 4 Report

Labwork 4
Seven Segment Display
31.10.2018
Şevki Gavrem Kulkuloğlu
21601793 EE 102 -2

```
reset <= '1';
wait for 50 ns;
clock_100Mhz <= '0';
reset <= '0';
end PROCESS;
end Behavioral;
```