Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

Design Methodology:

In this lab work we needed to show multiple digits of seven segment displays simultaneously. So, my aim is showing hexadecimal number on seven segment display with 16-bit binary number input with switches (when one of the switches is ON, the LED at the top of switch will be ON to distinguish which switches are ON.) Also I added push button input to reset clock counter signal and this feature gives us a chance to see which digit activated for a moment that we pressed push button.

When using seven segment displays, we can only send information for one digit at a time. So, to display multiple digits simultaneously we are using phenomenon called persistence of vision. In order to use this phenomenon, I wrote clock module that sends information to digits in loop with very high frequency. And I defined 4-2 multiplexer module in order to display every digits simultaneously with using my clock module. I defined switch of multiplexer as a clock counter signal (1downto0) and other inputs of multiplexer are switches with grouped by 4. (Such as 15downto12 is first digit, 11downto8 for second digit.) And as the outputs of multiplexer first output leads switch inputs to LED inputs such as "0001" and send the 4-bit output to our other module, decoder's input and second output directs the clock counter signals to activate anode. (Second output decides which digit will be displayed for a moment.) Also, showing letters and numbers in seven segment display is a bit complicated. Seven segment displays has seven LEDs with common anode and different cathodes. (there is one more LED for decimal point but I will ignore them for this lab work) We need to turn on appropriate LEDs to show our desirable digit. This situation does not have any math or logic. So we need to match our desired digits with 7-bit outputs. As I use 4-bit binary numbers to express every digit of 4-digit hexadecimal number, I defined decoder module to match 4-bit binary inputs from multiplexer module with 7-bit binary outputs. There are some figures that represent how seven segment displays work:

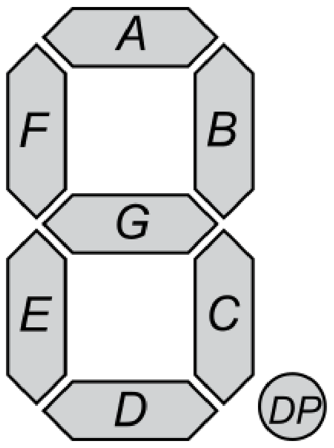


Figure 1: Seven Segment Displays' Common Anode LEDs

Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

					OUTPUTS						
DIGITS	INPUTS				a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0
A	1	0	1	0	0	0	0	0	0	1	0
b	1	0	1	1	1	1	0	0	0	0	0
C	1	1	0	0	0	1	1	0	0	0	1
d	1	1	0	1	1	0	0	0	0	1	0
E	1	1	1	0	0	1	1	0	0	0	0
F	1	1	1	1	0	1	1	1	0	0	0

Figure 2: Truth Table for Decoder (4-bit binary to 7-segments matching)

After writing three modules and one top module, there is the RTL schematic of my design:

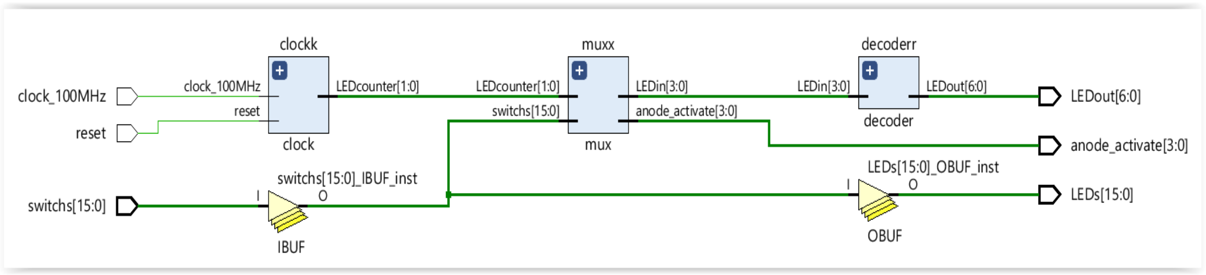


Figure 3: RTL Schematic of Design

Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

Results:

I wrote one top module, three modules and three test bench codes for each three module. After fixing some errors, it worked as what I expected.

There are schematics and behavioural simulations for each module:

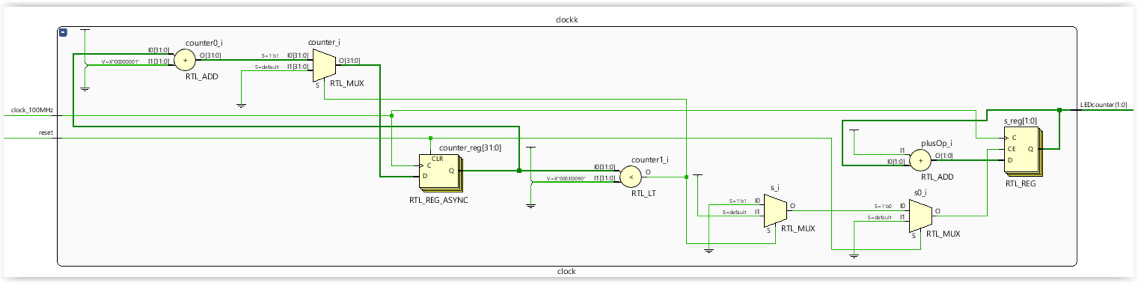
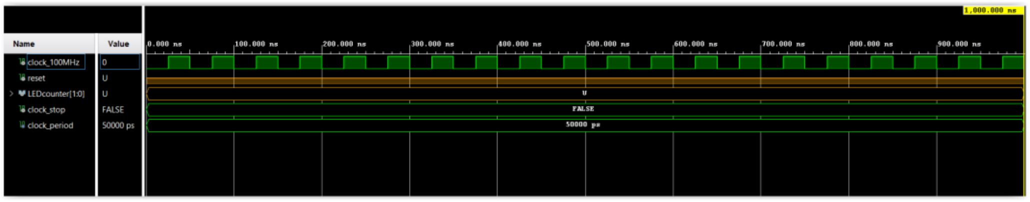


Figure 4: Schematic of Clock Module



Name	Value
clock_100MHz	0
reset	U
LEDcounter[1:0]	U
clock_stop	FALSE
clock_period	50000 ps

Figure 5: Behavioural Simulation of Clock Module

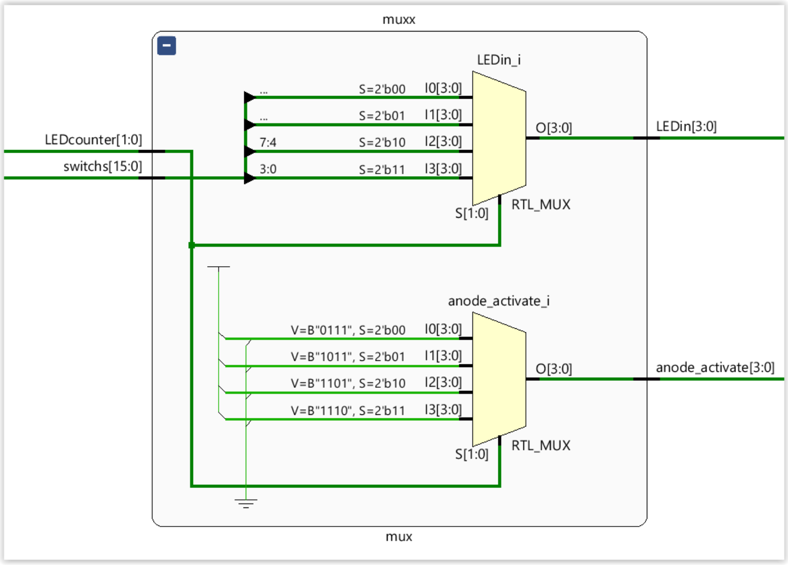


Figure 6: Schematic of Multiplexer Module

Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

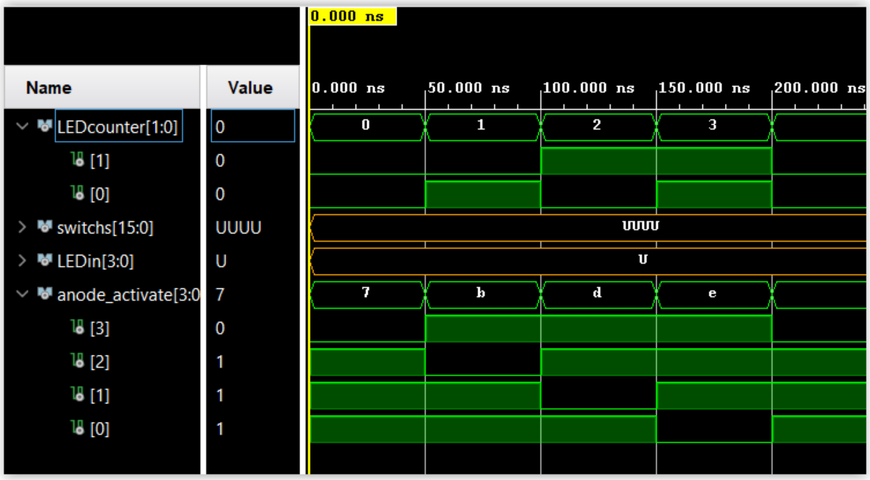


Figure 7: Behavioural Simulation of Multiplexer Module

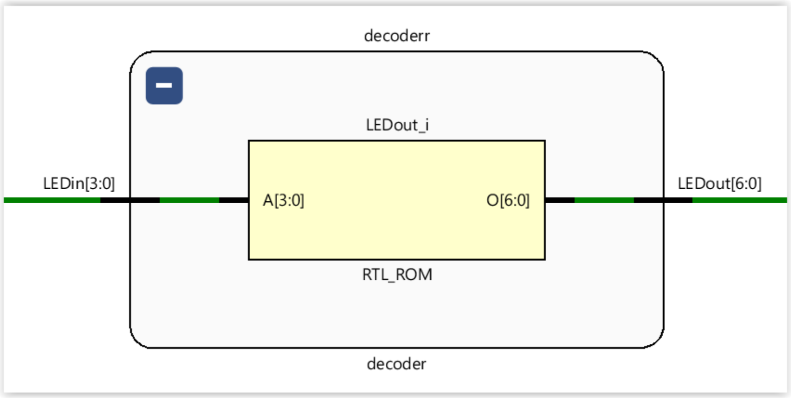


Figure 8: Schematic of Decoder Module

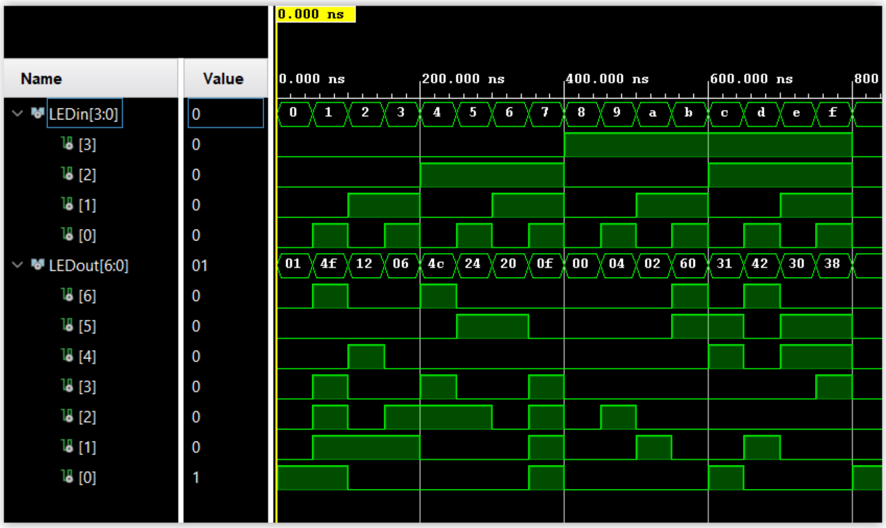


Figure 9: Behavioural Simulation of Decoder Module

Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

Now, we generated bitstream to work our code on BASYS3. There are the test results:

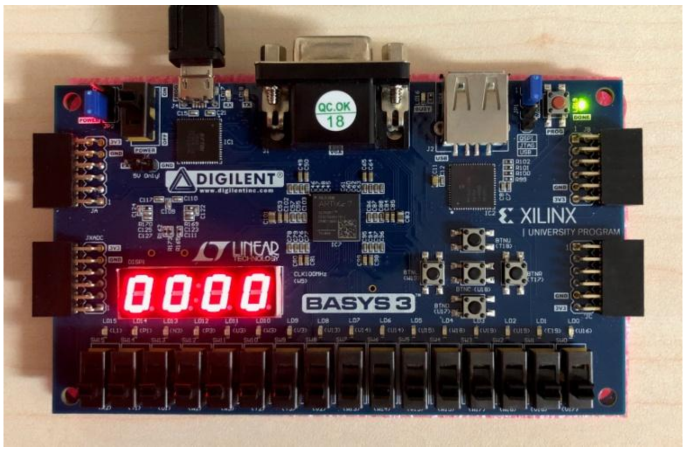


Figure 10: Test 1 (0000-0000-0000-0000 => 0000)

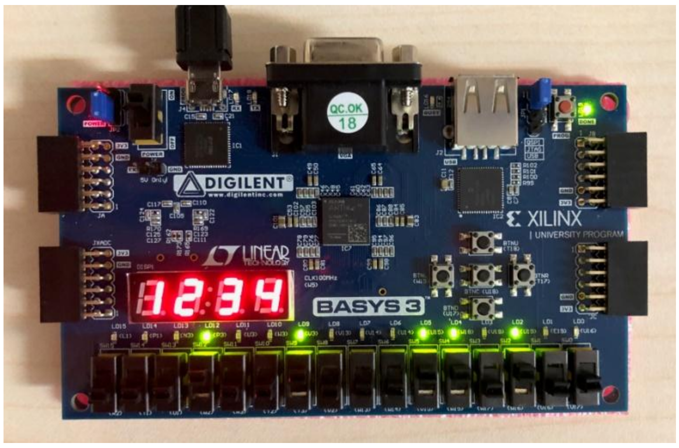


Figure 11: Test 2 (0001-0010-0011-0100 => 1234)

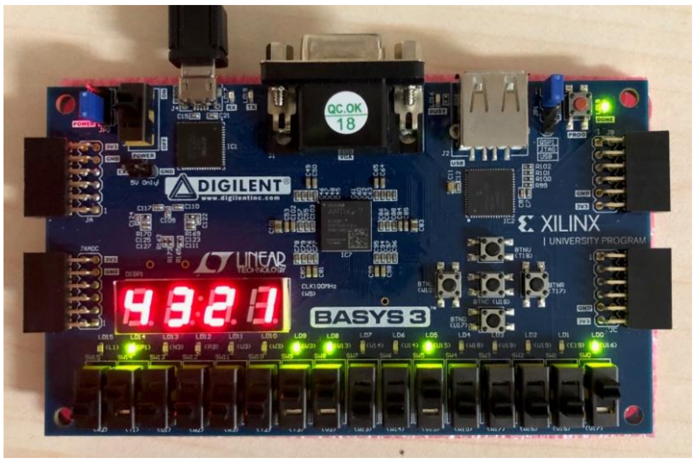


Figure 12: Test 3 (0100-0011-0010-0001 => 4321)

Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

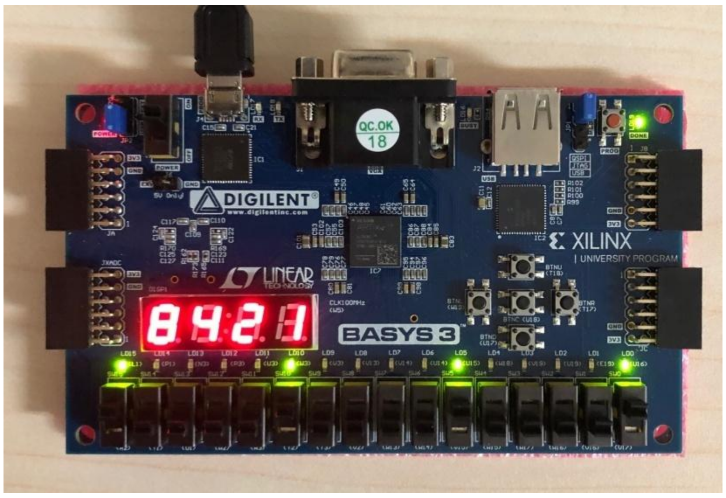


Figure 13: Test 4 (1000-0100-0010-0001 => 8421)

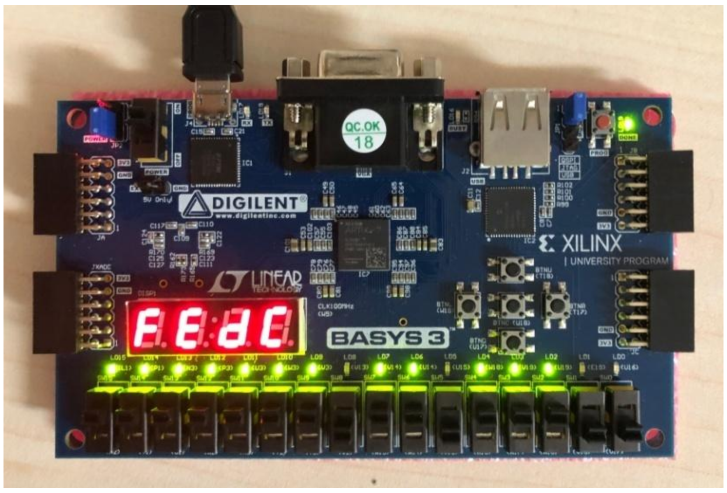


Figure 14: Test 5 (1111-1110-1101-1100 => FEdC)

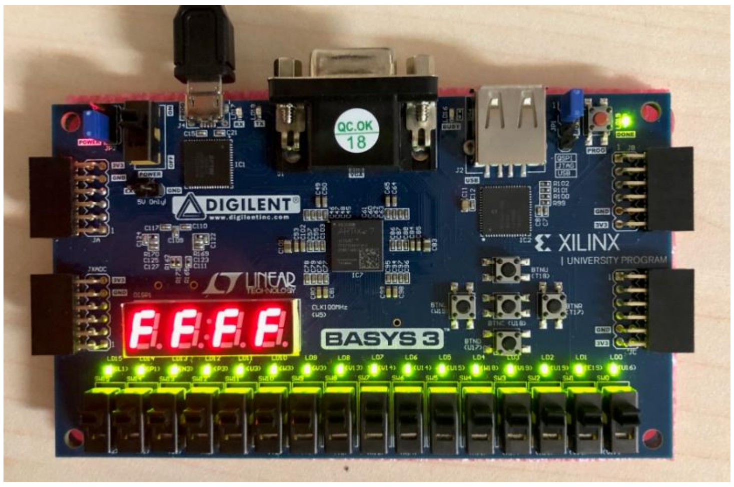


Figure 15: Test 6 (1111-1111-1111-1111 => FFFF)

Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

Conclusion:

In this lab work, I learned how to code seven segment displays with VHDL. Seven segment displays is a very essential concept for electronics because it is used in many electronic devices in real life. Also I learned how to do modular design. It is very useful for long and complicated projects. If we use modules, our code will be simpler and we can check for errors later easier than non modular design. In order to do behavioural simulation in modular design, I learned that I need to write test bench for every module separately. In the test part on BASYS3, I confronted an error. My digits displays were inverse. **(E –> Ǝ , 7 -> Ⱶ)** Then I realized that I wrote orders of seven segment displays constraints inversely (0->6 instead of 6->0) I corrected order of constraints and it worked as I expected.

Appendices:

Top Module:

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;



entity seven_segment is

 Port (

   clock_100MHz: in std_logic;

   reset: in std_logic;

   switchs: in std_logic_vector (15 downto 0);

   LEDout: out std_logic_vector (6 downto 0);

   anode_activate: out std_logic_vector (3 downto 0);

   LEDs: out std_logic_vector (15 downto 0)

   );

end seven_segment;
```

Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

```
architecture Behavioral of seven_segment is


component clock is

Port (

   clock_100MHz: in std_logic;

   reset: in std_logic;

   LEDcounter: out std_logic_vector (1 downto 0)

   );

end component;


component decoder is -- decoder kullanarak 0001 ile 0000001'i eşleştiriyoruz

Port (

   LEDin: in std_logic_vector (3 downto 0);

   LEDout: out std_logic_vector (6 downto 0)

   );

end component;


component mux is -- 4-1 multiplexer ile switchleri ledlere göre 4'lü grupluyoruz

Port (

   LEDcounter: in std_logic_vector (1 downto 0);

   switchs: in std_logic_vector (15 downto 0);

   LEDin: out std_logic_vector (3 downto 0);

   anode_activate: out std_logic_vector (3 downto 0)

   );
```

```
end component;


signal LEDcounter: std_logic_vector (1 downto 0);

signal LEDin: std_logic_vector (3 downto 0);




begin

LEDs <= switchs;

clockk: clock port map ( clock_100MHz => clock_100MHz, reset => reset, LEDcounter =>
LEDcounter);

decoderr: decoder port map ( LEDin => LEDin, LEDout => LEDout);

muxx: mux port map ( LEDcounter => LEDcounter, switchs => switchs, LEDin => LEDin,
anode_activate => anode_activate);

end Behavioral;
```

Clock Module:

```
----------------------------------------------------------------------------------

-- Company:

-- Engineer:

--

-- Create Date: 20.03.2021 18:54:57

-- Design Name:

-- Module Name: clock - Behavioral

-- Project Name:
```

```
-- Target Devices:

-- Tool Versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

----------------------------------------------------------------------------------



library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

use IEEE.std_logic_unsigned.all;

-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC_STD.ALL;


-- Uncomment the following library declaration if instantiating

-- any Xilinx leaf cells in this code.

--library UNISIM;

--use UNISIM.VComponents.all;
```

Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

```
entity clock is

   Port (

   clock_100MHz: in std_logic;

   reset: in std_logic;

   LEDcounter: out std_logic_vector (1 downto 0)

   );

end clock;


architecture Behavioral of clock is

   signal counter: integer:= 0;

   signal s    : std_logic_vector (1 downto 0);

begin

   process(clock_100MHz)

   begin

      if reset='1' then

      counter <= 0;

      elsif rising_edge(clock_100MHz) then

         if counter < 250000 then

            counter <= counter + 1;

         else

            counter <= 0;

            s <= s+1;

         end if;
```

```
        end if;
    end process;

LEDcounter <= std_logic_vector(s);

end Behavioral;
```

Multiplexer Module:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 20.03.2021 18:41:34
-- Design Name:
-- Module Name: mux - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
```

Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

```
-- Additional Comments:

--

----------------------------------------------------------------------------------




library IEEE;

use IEEE.STD_LOGIC_1164.ALL;


-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC_STD.ALL;


-- Uncomment the following library declaration if instantiating

-- any Xilinx leaf cells in this code.

--library UNISIM;

--use UNISIM.VComponents.all;


entity mux is

   Port (

   LEDcounter: in std_logic_vector (1 downto 0);

   switchs: in std_logic_vector (15 downto 0);

   LEDin: out std_logic_vector (3 downto 0);

   anode_activate: out std_logic_vector (3 downto 0)

   );
```

```
end mux;


architecture Behavioral of mux is


begin
  process(LEDcounter)
  begin
    case LEDcounter is
    when "00" =>
      anode_activate <= "0111"; --LED1 is activated
      LEDin <= switchs(15 downto 12);
    when "01" =>
      anode_activate <= "1011"; --LED2 is activated
      LEDin <= switchs(11 downto 8);
    when "10" =>
      anode_activate <= "1101"; --LED3 is activated
      LEDin <= switchs(7 downto 4);
    when "11" =>
      anode_activate <= "1110"; --LED2 is activated
      LEDin <= switchs(3 downto 0);
    when others =>
      anode_activate <= "1111"; --no LED activated
      LEDin <= "0000";
    end case;
    end process;



end Behavioral;
```

Decoder Module:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 20.03.2021 17:49:24
-- Design Name:
-- Module Name: decoder - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
```

Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

```
----------------------------------------------------------------------------------



library IEEE;

use IEEE.STD_LOGIC_1164.ALL;


-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC_STD.ALL;


-- Uncomment the following library declaration if instantiating

-- any Xilinx leaf cells in this code.

--library UNISIM;

--use UNISIM.VComponents.all;


entity decoder is

   Port (

      LEDin: in std_logic_vector (3 downto 0);

      LEDout: out std_logic_vector (6 downto 0)

   );

end decoder;


architecture Behavioral of decoder is
```

Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

```
begin
    process(LEDin)
    begin
        case LEDin is
        when "0000" => LEDout <= "0000001"; --0
        when "0001" => LEDout <= "1001111"; --1
        when "0010" => LEDout <= "0010010"; --2
        when "0011" => LEDout <= "0000110"; --3
        when "0100" => LEDout <= "1001100"; --4
        when "0101" => LEDout <= "0100100"; --5
        when "0110" => LEDout <= "0100000"; --6
        when "0111" => LEDout <= "0001111"; --7
        when "1000" => LEDout <= "0000000"; --8
        when "1001" => LEDout <= "0000100"; --9
        when "1010" => LEDout <= "0000010"; --a
        when "1011" => LEDout <= "1100000"; --b
        when "1100" => LEDout <= "0110001"; --C
        when "1101" => LEDout <= "1000010"; --d
        when "1110" => LEDout <= "0110000"; --E
        when "1111" => LEDout <= "0111000"; --F
        when others => LEDout <= "1111111"; --no LED activated
        end case;
    end process;
end Behavioral;
```

Test Bench for Clock Module:

```
----------------------------------------------------------------------------------
-- Company: 
-- Engineer: 
-- 
-- Create Date: 26.03.2021 13:58:41
-- Design Name: 
-- Module Name: tb_clock - Behavioral
-- Project Name: 
-- Target Devices: 
-- Tool Versions: 
-- Description: 
-- 
-- Dependencies: 
-- 
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
-- 
----------------------------------------------------------------------------------
```

Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;


-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC_STD.ALL;


-- Uncomment the following library declaration if instantiating

-- any Xilinx leaf cells in this code.

--library UNISIM;

--use UNISIM.VComponents.all;


entity tb_clock is end;



architecture Behavioral of tb_clock is

component clock

Port(

   clock_100MHz: in std_logic;

   reset: in std_logic;

   LEDcounter: out std_logic_vector(1 downto 0)

   );

end component;
```

Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

```
signal clock_100MHz: std_logic;

signal reset: std_logic;

signal LEDcounter: std_logic_vector(1 downto 0);


constant clock_period: time:=50ns;

signal clock_stop: boolean;


begin

portmap: clock port map(

    clock_100MHz => clock_100MHz,

    reset => reset,

    LEDcounter => LEDcounter

            );


process

  begin

    while not clock_stop

    loop

    clock_100MHz <= '0','1'

    after clock_period/2;

    wait for clock_period;

    end loop;

    wait;
    end process;

end Behavioral;
```

Test Bench for Multiplexer Module:

```
----------------------------------------------------------------------------------

-- Company:

-- Engineer:

--

-- Create Date: 26.03.2021 14:19:49

-- Design Name:

-- Module Name: tb_mux - Behavioral

-- Project Name:

-- Target Devices:

-- Tool Versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

----------------------------------------------------------------------------------
```

Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;


-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC_STD.ALL;


-- Uncomment the following library declaration if instantiating

-- any Xilinx leaf cells in this code.

--library UNISIM;

--use UNISIM.VComponents.all;


entity tb_mux is end;




architecture Behavioral of tb_mux is

component mux


   Port (

      LEDcounter: in std_logic_vector (1 downto 0);

      switchs: in std_logic_vector (15 downto 0);

      LEDin: out std_logic_vector (3 downto 0);
```

```
        anode_activate: out std_logic_vector (3 downto 0)
        );
end component;

signal LEDcounter: std_logic_vector (1 downto 0);
signal switchs: std_logic_vector (15 downto 0);
signal LEDin: std_logic_vector (3 downto 0);
signal anode_activate: std_logic_vector (3 downto 0);

begin

    portmap: mux port map
        (
        LEDcounter => LEDcounter,
        switchs => switchs,
        LEDin => LEDin,
        anode_activate => anode_activate
        );

    process
        begin
        LEDcounter <= "00";
        wait for 50ns;
        LEDcounter <= "01";
        wait for 50ns;

        LEDcounter <= "10";

        wait for 50ns;

        LEDcounter <= "11";

        wait for 50ns;

        wait;

    end process;


end Behavioral;
```

Test Bench for Decoder Module:

```
----------------------------------------------------------------------------------

-- Company:

-- Engineer:

--

-- Create Date: 26.03.2021 15:06:22

-- Design Name:

-- Module Name: tb_decoder - Behavioral

-- Project Name:

-- Target Devices:

-- Tool Versions:

-- Description:

--

-- Dependencies:
```

```
--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

----------------------------------------------------------------------------------




library IEEE;

use IEEE.STD_LOGIC_1164.ALL;


-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC_STD.ALL;


-- Uncomment the following library declaration if instantiating

-- any Xilinx leaf cells in this code.

--library UNISIM;

--use UNISIM.VComponents.all;


entity tb_decoder is end;


architecture Behavioral of tb_decoder is

component decoder
```

```
    Port    (
            LEDin: in std_logic_vector (3 downto 0);
            LEDout: out std_logic_vector (6 downto 0)
            );
end component;

signal LEDin: std_logic_vector (3 downto 0);
signal LEDout: std_logic_vector (6 downto 0);

begin

portmap: decoder port map
        (
        LEDin => LEDin,
        LEDout => LEDout
        );

    process
    begin
    LEDin <= "0000";
    wait for 50ns;
    LEDin <= "0001";
    wait for 50ns;
```

```
LEDin <= "0010";

wait for 50ns;

LEDin <= "0011";

wait for 50ns;

LEDin <= "0100";

wait for 50ns;

LEDin <= "0101";

wait for 50ns;

LEDin <= "0110";

wait for 50ns;

LEDin <= "0111";

wait for 50ns;

LEDin <= "1000";

wait for 50ns;

LEDin <= "1001";

wait for 50ns;

LEDin <= "1010";

wait for 50ns;

LEDin <= "1011";

wait for 50ns;

LEDin <= "1100";

wait for 50ns;

LEDin <= "1101";

wait for 50ns;
```

```
    LEDin <= "1110";

    wait for 50ns;

    LEDin <= "1111";

    wait for 50ns;

    LEDin <= "0000";

    wait;

    end process;

end Behavioral;
```

Constraints :

```
# Clock signal

set_property PACKAGE_PIN W5 [get_ports clock_100MHz]

    set_property IOSTANDARD LVCMOS33 [get_ports clock_100MHz]

set_property PACKAGE_PIN U18 [get_ports reset]

    set_property IOSTANDARD LVCMOS33 [get_ports reset]


# Switches

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

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

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

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

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

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

set_property PACKAGE_PIN W17 [get_ports {switchs[3]}]
```

```
   set_property IOSTANDARD LVCMOS33 [get_ports {switchs[3]}]
set_property PACKAGE_PIN W15 [get_ports {switchs[4]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {switchs[4]}]
set_property PACKAGE_PIN V15 [get_ports {switchs[5]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {switchs[5]}]
set_property PACKAGE_PIN W14 [get_ports {switchs[6]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {switchs[6]}]
set_property PACKAGE_PIN W13 [get_ports {switchs[7]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {switchs[7]}]
set_property PACKAGE_PIN V2 [get_ports {switchs[8]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {switchs[8]}]
set_property PACKAGE_PIN T3 [get_ports {switchs[9]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {switchs[9]}]
set_property PACKAGE_PIN T2 [get_ports {switchs[10]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {switchs[10]}]
set_property PACKAGE_PIN R3 [get_ports {switchs[11]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {switchs[11]}]
set_property PACKAGE_PIN W2 [get_ports {switchs[12]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {switchs[12]}]
set_property PACKAGE_PIN U1 [get_ports {switchs[13]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {switchs[13]}]
set_property PACKAGE_PIN T1 [get_ports {switchs[14]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {switchs[14]}]
set_property PACKAGE_PIN R2 [get_ports {switchs[15]}]
```

```
    set_property IOSTANDARD LVCMOS33 [get_ports {switchs[15]}]

  set_property IOSTANDARD LVCMOS33 [get_ports {switchs[15]}]


# LEDs

set_property PACKAGE_PIN U16 [get_ports {LEDs[0]}]

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

set_property PACKAGE_PIN E19 [get_ports {LEDs[1]}]

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

set_property PACKAGE_PIN U19 [get_ports {LEDs[2]}]

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

set_property PACKAGE_PIN V19 [get_ports {LEDs[3]}]

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

set_property PACKAGE_PIN W18 [get_ports {LEDs[4]}]

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

set_property PACKAGE_PIN U15 [get_ports {LEDs[5]}]

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

set_property PACKAGE_PIN U14 [get_ports {LEDs[6]}]

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

set_property PACKAGE_PIN V14 [get_ports {LEDs[7]}]

    set_property IOSTANDARD LVCMOS33 [get_ports {LEDs[7]}]

set_property PACKAGE_PIN V13 [get_ports {LEDs[8]}]

    set_property IOSTANDARD LVCMOS33 [get_ports {LEDs[8]}]

set_property PACKAGE_PIN V3 [get_ports {LEDs[9]}]

    set_property IOSTANDARD LVCMOS33 [get_ports {LEDs[9]}]
```

Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

```
set_property PACKAGE_PIN W3 [get_ports {LEDs[10]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {LEDs[10]}]
set_property PACKAGE_PIN U3 [get_ports {LEDs[11]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {LEDs[11]}]
set_property PACKAGE_PIN P3 [get_ports {LEDs[12]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {LEDs[12]}]
set_property PACKAGE_PIN N3 [get_ports {LEDs[13]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {LEDs[13]}]
set_property PACKAGE_PIN P1 [get_ports {LEDs[14]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {LEDs[14]}]
set_property PACKAGE_PIN L1 [get_ports {LEDs[15]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {LEDs[15]}]


#7 segment display
set_property PACKAGE_PIN W7 [get_ports {LEDout[6]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {LEDout[6]}]
set_property PACKAGE_PIN W6 [get_ports {LEDout[5]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {LEDout[5]}]
set_property PACKAGE_PIN U8 [get_ports {LEDout[4]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {LEDout[4]}]
set_property PACKAGE_PIN V8 [get_ports {LEDout[3]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {LEDout[3]}]
set_property PACKAGE_PIN U5 [get_ports {LEDout[2]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {LEDout[2]}]
```

Mehmet Bayık
21802166
EE-102 Section 3
Lab Work 4
26.03.2021

```
set_property PACKAGE_PIN V5 [get_ports {LEDout[1]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {LEDout[1]}]
set_property PACKAGE_PIN U7 [get_ports {LEDout[0]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {LEDout[0]}]
set_property PACKAGE_PIN U2 [get_ports {anode_activate[0]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {anode_activate[0]}]
set_property PACKAGE_PIN U4 [get_ports {anode_activate[1]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {anode_activate[1]}]
set_property PACKAGE_PIN V4 [get_ports {anode_activate[2]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {anode_activate[2]}]
set_property PACKAGE_PIN W4 [get_ports {anode_activate[3]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {anode_activate[3]}]
```