# Lab04 – Hands on Experience with Basys3

## 1 7-Segment Display

### Top Module –

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

entity BCD_7SEG_TOP is Port (
CLK_100MHZ : in std_logic;
BCD_SW: in std_logic_vector (15 downto 0);
SEG_OUT: out std_logic_vector (6 downto 0);
ANODE_ACT: out std_logic_vector (3 downto 0)
 ); end BCD_7SEG_TOP;

architecture Behavioral of BCD_7SEG_TOP is
component  Anode_Activate_SEG is Port (
CLK_100MHZ : in std_logic;
BCD_SW: in std_logic_vector (15 downto 0);
Anode_Activate: out std_logic_vector (3 downto 0);
SEG_BCD : out std_logic_vector (3 downto 0)
);end component;

component BCD_Deccoder is Port (
SEG_BCD : in std_logic_vector (3 downto 0);
SEG_OUT: out std_logic_vector (6 downto 0)
 ); end component;

signal BCD_CODE : std_logic_vector (3 downto 0);
begin
uu1: Anode_Activate_SEG Port map( CLK_100MHZ => CLK_100MHZ, BCD_SW=>BCD_SW, SEG_BCD=>BCD_CODE
    , Anode_Activate=>ANODE_ACT );
uu2: BCD_Deccoder Port map(SEG_BCD => BCD_CODE, SEG_OUT => SEG_OUT );
end Behavioral;
```

### SubModule – BCD Decoder

```
entity BCD_Deccoder is Port (
SEG_BCD : in std_logic_vector (3 downto 0);
SEG_OUT: out std_logic_vector (6 downto 0)
 ); end BCD_Deccoder;

architecture Behavioral of BCD_Deccoder is
begin
process(SEG_BCD)
begin
    case SEG_BCD  is
    when "0000" => SEG_OUT <= "1000000"; -- "0"
    when "0001" => SEG_OUT <= "1111001"; -- "1"
    when "0010" => SEG_OUT <= "0100100"; -- "2"
    when "0011" => SEG_OUT <= "0110000"; -- "3"
    when "0100" => SEG_OUT <= "0011001"; -- "4"
    when "0101" => SEG_OUT <= "0010010"; -- "5"
    when "0110" => SEG_OUT <= "1000010"; -- "6"
    when "0111" => SEG_OUT <= "1111000"; -- "7"
    when "1000" => SEG_OUT <= "0000000"; -- "8"
    when "1001" => SEG_OUT <= "0010000"; -- "9"
    when "1010" => SEG_OUT <= "0100000"; -- a
    when "1011" => SEG_OUT <= "0000011"; -- b
    when "1100" => SEG_OUT <= "1000110"; -- C
    when "1101" => SEG_OUT <= "1000010"; -- d
    when "1110" => SEG_OUT <= "0000110"; -- E
    when "1111" => SEG_OUT <= "0001110"; -- F
    end case;
end process;
end Behavioral;
```

## Sub Module– Anode Activate SEG

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

entity Anode_Activate_SEG is Port (
CLK_100MHZ : in std_logic;
BCD_SW: in std_logic_vector (15 downto 0);
Anode_Activate: out std_logic_vector (3 downto 0);
SEG_BCD : out std_logic_vector (3 downto 0)
);
end Anode_Activate_SEG;

architecture Behavioral of Anode_Activate_SEG is
signal refresh_counter: std_logic_vector(19 downto 0);
signal SEG_active_Count: std_logic_vector(1 downto 0);

begin

process(CLK_100MHZ)
begin
    if(rising_edge(CLK_100MHZ)) then
        refresh_counter <= refresh_counter + 1;
     end if;
end process;
 SEG_active_Count <= refresh_counter(19 downto 18);

process(SEG_active_Count)
    begin
        case SEG_active_Count is
        when "00" => Anode_Activate <= "0111";
                    SEG_BCD <= BCD_SW(15 downto 12);
        when "01" => Anode_Activate <= "1011";
                    SEG_BCD <= BCD_SW(11 downto 8);
        when "10" =>  Anode_Activate <= "1101";
                    SEG_BCD <= BCD_SW(7 downto 4);
        when "11" =>  Anode_Activate <= "1110";
                    SEG_BCD <= BCD_SW(3 downto 0);
    end case;
end process;
end Behavioral;
```