Lab 7 – 7 Segment Display

CS1050 Computer Organization and Digital Design

Name : K. Oshadi Saumya Perera

Index No : 200458M

Lab Task

- We design a 7-Segment Display for our 4-bit Arithmetic Unit developed in previous lab. BASYS 3 has 4, 7-segment displays and 7 segments can be used to display hexagonal numbers by switching on or off the desired segments.
- In this lab, we display the output of 4-bit sum of 4-bit AU as a hexadecimal number using a 7-segment display.
- We design and develop a lookup table using Read Only Memory (ROM) to map the 4-bit sum to the 7-segments on the display.
- We build 4-bit AU symbol by importing all relevant VHDL files from previous lab (AU and all its low-level entities and slow clock).
- We design and develop a 7-segment display using the output from the lookup table.
- Connecting inputs and outputs.
- We verify their functionality via simulation and change the switches on the BASYS 3 board.
- Modify the high level circuit to show the output only on the right most 7-segment display.

➢ Lookup table

Output from RCA					Segments to Switch On						
S_3	S_2	S_1	S_0	Hex. Value	A	B	C	D	E	F	G
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	1	0	0	1	1	1	1
0	0	1	0	2	0	0	1	0	0	1	0
0	0	1	1	3	0	0	0	0	1	1	0
0	1	0	0	4	1	0	0	1	1	0	0
0	1	0	1	5	0	1	0	0	1	0	0
0	1	1	0	6	0	1	0	0	0	0	0
0	1	1	1	7	0	0	0	1	1	1	1
1	0	0	0	8	0	0	0	0	0	0	0
1	0	0	1	9	0	0	0	0	1	0	0
1	0	1	0	A	0	0	0	1	0	0	0
1	0	1	1	B	1	1	0	0	0	0	0
1	1	0	0	C	0	1	1	0	0	0	1
1	1	0	1	D	1	0	0	0	0	1	0
1	1	1	0	E	0	1	1	0	0	0	0
1	1	1	1	F	0	1	1	1	0	0	0

➢ VHDL Codes

AU Design Source Code (AU.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/16/2022 02:43:52 PM
-- Design Name:
-- Module Name: AU - 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 AU is
   Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
          RegSel : in STD_LOGIC;
          Clk : in STD_LOGIC;
          S : out STD_LOGIC_VECTOR (3 downto 0);
          Zero : out STD_LOGIC;
          Carry : out STD_LOGIC);
end AU;

architecture Behavioral of AU is
component Slow_Clk
```

```
   Port ( Clk_in : in STD_LOGIC;
          Clk_out : out STD_LOGIC);
end component;

component RCA_4
   Port ( A0 : in STD_LOGIC;
          A1 : in STD_LOGIC;
          A2 : in STD_LOGIC;
          A3 : in STD_LOGIC;
          B0 : in STD_LOGIC;
          B1 : in STD_LOGIC;
          B2 : in STD_LOGIC;
          B3 : in STD_LOGIC;
          C_in : in STD_LOGIC;
          S0 : out STD_LOGIC;
          S1 : out STD_LOGIC;
          S2 : out STD_LOGIC;
          S3 : out STD_LOGIC;
          C_out : out STD_LOGIC);
end component;

component Reg
   Port ( D : in STD_LOGIC_VECTOR (3 downto 0);
          En : in STD_LOGIC;
          Clk : in STD_LOGIC;
          Q : out STD_LOGIC_VECTOR (3 downto 0));
end component;
signal slow_clock : STD_LOGIC;
signal En_A, EN_B, C_out : STD_LOGIC;
```

```
signal Q_A,Q_B,S_RCA : STD_LOGIC_VECTOR (3 downto 0);

begin
Slow_Clk_0 : Slow_Clk
   PORT MAP(
      Clk_in => Clk,
      Clk_out => slow_clock
   );
Reg_A : Reg
   PORT MAP(
      D => A,
      En => EN_A ,
      Clk => slow_clock,
      Q => Q_A
   );

Reg_B : Reg
   PORT MAP(
      D => A,
      En => En_B,
      Clk => slow_clock,
      Q => Q_B
   );

RCA_4_0 : RCA_4
   PORT MAP(
      A0 =>Q_A(0),
      A1 =>Q_A(1),
      A2 =>Q_A(2),
```

```
        A3 =>Q_A(3),
        B0 =>Q_B(0),
        B1 =>Q_B(1),
        B2 =>Q_B(2),
        B3 =>Q_B(3),
        C_in =>'0',
        S0 =>S_RCA(0),
        S1 =>S_RCA(1),
        S2 =>S_RCA(2),
        S3 =>S_RCA(3),
        C_out => C_out

    );
    EN_A <= RegSel;
    EN_B <= NOT(RegSel);
    S <= S_RCA;
    Carry <= C_out;
    Zero <= NOT(S_RCA(0)) AND NOT(S_RCA(1)) AND NOT(S_RCA(2)) AND NOT(S_RCA(3));

end Behavioral;
```

Slow clock Design Source code (Slow_Clk.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/09/2022 02:47:51 PM
-- Design Name:
-- Module Name: Slow_Clk - 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 Slow_Clk is
   Port ( Clk_in : in STD_LOGIC;
          Clk_out : out STD_LOGIC);
end Slow_Clk;

architecture Behavioral of Slow_Clk is

signal count : integer :=1;
signal clk_status : STD_LOGIC := '0';

begin
   --For 100MHz input clock this generates 1Hz clock
   process (Clk_in)
   begin
      if(rising_edge(Clk_in)) then
         count <= count+1;   --Increment counter
         --if(count = 50000000) then     --Count 50M pulses (1/2 of period)
         if(count = 4) then
            clk_status <= NOT(clk_status);    --Invert clock status
            Clk_out <= clk_status;
            count <= 1;    --Reset counter
         end if;

       end if;
   end process;
end Behavioral;
```

Lookup Table Design Source Code (LUT_16_7.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/23/2022 11:21:51 AM
-- Design Name:
-- Module Name: LUT_16_7 - 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.numeric_std.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 LUT_16_7 is
    Port ( address : in STD_LOGIC_VECTOR (3 downto 0);
           data : out STD_LOGIC_VECTOR (6 downto 0));
end LUT_16_7;

architecture Behavioral of LUT_16_7 is
    type rom_type is array (0 to 15) of std_logic_vector(6 downto 0);
        signal sevenSegment_ROM : rom_type := (
            "1000000", --0
            "1111001", --1
            "0100100", --2
            "0110000", --3
            "0011001", --4
            "0010010", --5
            "0000010", --6
            "1111000", --7
            "0000000", --8
            "0010000", --9
            "0001000", --a
            "0000011", --b
            "1000110", --c
            "0100001", --d
            "0000110", --e
            "0001110"  --f
        );
begin
    data <= sevenSegment_ROM(to_integer(unsigned(address)));
end Behavioral;
```

Lookup Table Test Bench Code (TB_LUT_16_7.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/23/2022 11:17:51 PM
-- Design Name:
-- Module Name: TB_LUT_16_7 - 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_LUT_16_7 is
--  Port ( );
end TB_LUT_16_7;

architecture Behavioral of TB_LUT_16_7 is
component LUT_16_7
   Port ( address : in STD_LOGIC_VECTOR (3 downto 0);
          data : out STD_LOGIC_VECTOR (6 downto 0));
end component;
signal address : STD_LOGIC_VECTOR (3 downto 0);
signal data : STD_LOGIC_VECTOR (6 downto 0);
begin
   UUT : LUT_16_7
   PORT MAP(
      address => address,
      data => data
   );

   process
   begin
      --index 200458M = 11 0000 1111 0000 1010
      address <= "0101";
      wait for 100ns;

      address <= "0000";
      wait for 100ns;

      address <= "1111";
```

```
        wait for 100ns;

        address <= "0000";
        wait for 100ns;

        wait;
    end process;

end Behavioral;
```

7-Segment Display Design Source Code (AU_7_seg.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/23/2022 02:13:27 PM
-- Design Name:
-- Module Name: AU_7_seg - 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 AU_7_seg is
    Port (
        A : in STD_LOGIC_VECTOR (3 downto 0);
        RegSel : in STD_LOGIC;
        Clk : in STD_LOGIC;
        S_LED : out STD_LOGIC_VECTOR (3 downto 0);
        Zero : out STD_LOGIC;
        Carry : out STD_LOGIC;
        S_7Seg : out STD_LOGIC_VECTOR (6 downto 0);
        Anode : out STD_LOGIC_VECTOR (3 downto 0)
    );
end AU_7_seg;

architecture Behavioral of AU_7_seg is
component AU
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           RegSel : in STD_LOGIC;
           Clk : in STD_LOGIC;
```

```
        S : out STD_LOGIC_VECTOR (3 downto 0);
        Zero : out STD_LOGIC;
        Carry : out STD_LOGIC);
end component;
component LUT_16_7
   Port ( address : in STD_LOGIC_VECTOR (3 downto 0);
        data : out STD_LOGIC_VECTOR (6 downto 0));
end component;
signal Sum : STD_LOGIC_VECTOR (3 downto 0);
signal data : STD_LOGIC_VECTOR (6 downto 0);
begin
AU_0 : AU
   PORT MAP(
       A => A,
       RegSel => RegSel,
       Clk => Clk,
       S => Sum,
       Zero => Zero,
       Carry => Carry
   );
LUT_16_7_0 : LUT_16_7
   PORT MAP(
      address => Sum,
      data => data
   );
S_LED <= Sum;
S_7Seg <= data;
Anode <= "1110";
end Behavioral;
```

7-Segment Display Test Bench Code (TB_AU_7_seg.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/24/2022 12:04:34 AM
-- Design Name:
-- Module Name: TB_AU_7_seg - 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;
```

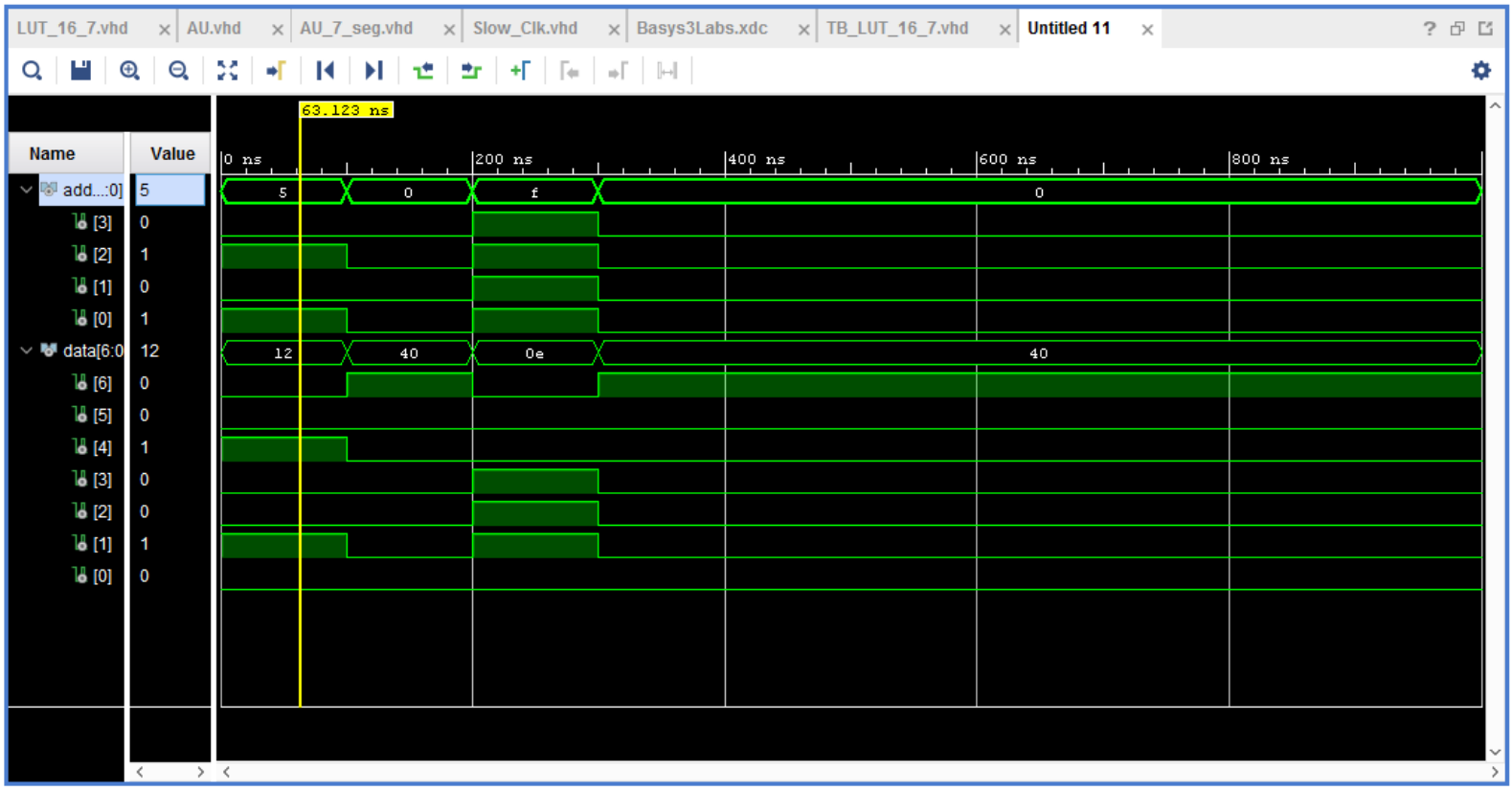
```
entity TB_AU_7_seg is
--  Port ( );
end TB_AU_7_seg;

architecture Behavioral of TB_AU_7_seg is
component AU_7_seg
   Port (
      A : in STD_LOGIC_VECTOR (3 downto 0);
      RegSel : in STD_LOGIC;
      Clk : in STD_LOGIC;
      S_LED : out STD_LOGIC_VECTOR (3 downto 0);
      Zero : out STD_LOGIC;
      Carry : out STD_LOGIC;
      S_7Seg : out STD_LOGIC_VECTOR (6 downto 0);
      Anode : out STD_LOGIC_VECTOR (3 downto 0)
   );
end component;
signal A,S_LED : STD_LOGIC_VECTOR (3 downto 0) := "0000";
signal Anode : STD_LOGIC_VECTOR (3 downto 0);
signal S_7Seg : STD_LOGIC_VECTOR (6 downto 0);
signal RegSel, Clk, Zero, Carry : STD_LOGIC := '0';
begin
UUT : AU_7_seg
   PORT MAP(
      A => A ,
      RegSel => RegSel,
      Clk => Clk,
      S_LED => S_LED,
```

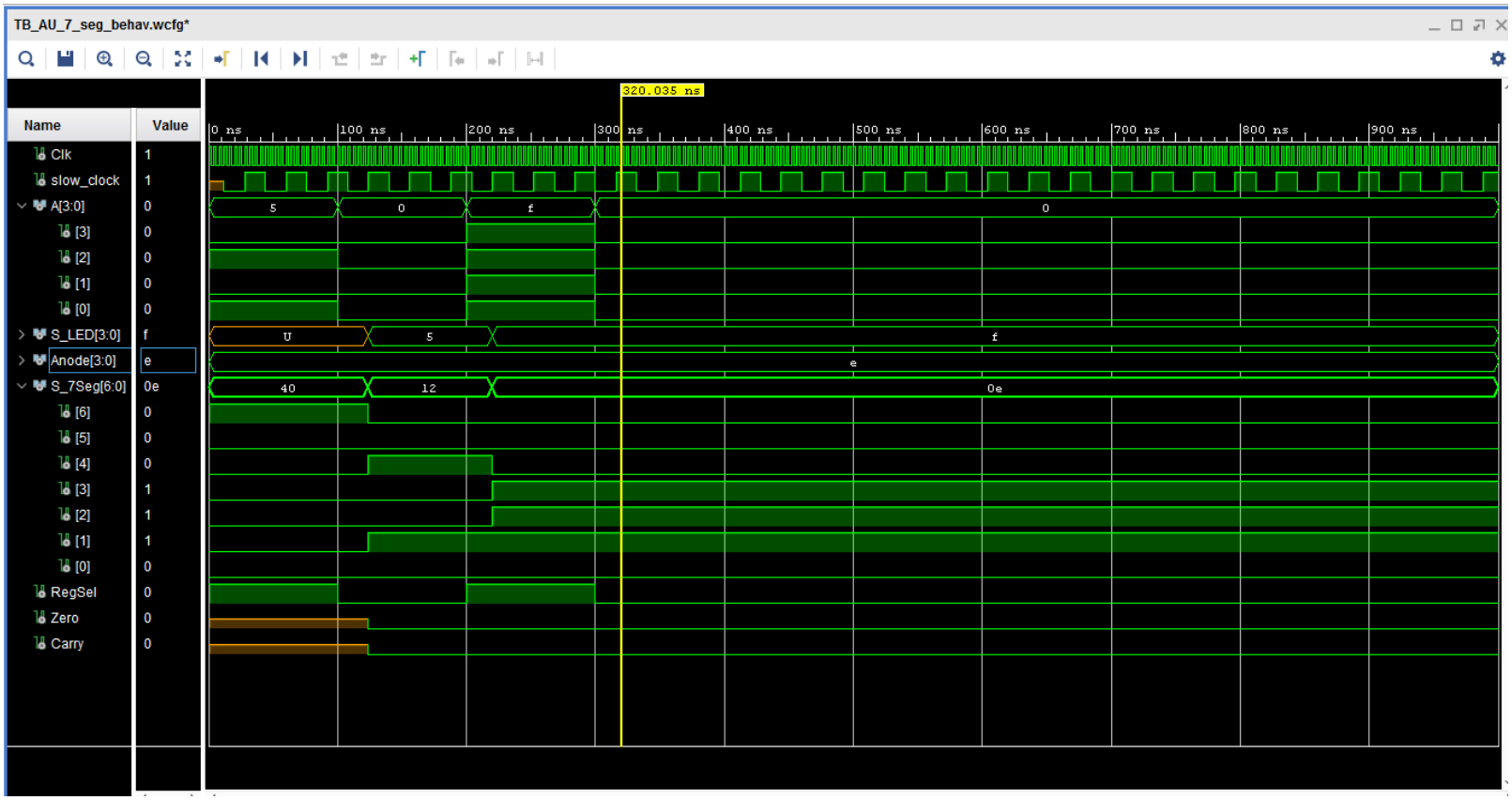
```
        Zero => Zero ,
        Carry => Carry,
        S_7Seg => S_7Seg,
        Anode => Anode
    );
    process
    begin
        Clk <= NOT(Clk);
        wait for 2ns;
    end process;
    process
    begin
        --index 200458M = 11 0000 1111 0000 1010
        --1010 + 0000
        A <= "0101";
        RegSel <= '1';
        wait for 100ns;
        A <= "0000";
        RegSel <= '0';
        wait for 100ns;
        --1111 + 0000
        A <= "1111";
        RegSel <= '1';
        wait for 100ns;
        A <= "0000";
        RegSel <= '0';
        wait;
end process;
end Behavioral;
```

- Timing Diagrams

Lookup table



7-Segment Display



- How to show results using only a single 7-segment

Seven Segment LED display in BASYS 3 board contains one four-common anode and the LED cathodes which are remained separately. By the use of transistors, BASYS 3 drives enough current into the common anode point. Therefore when active, both anode and cathode signals are driven low. We can modify the design source code of 7-segment display to show the output only on the right most 7-segment display. We can give high to left most three anodes which needed to be switched off and give low to right most anode which needed to be switched on (light up). We can add an **Anode** a 4-bit output bus to the 7-segment display and give **"1110"** to it.

➢ Conclusions from the lab

- The segment which we want to light up or down also depends on the specific 7-segment implementation, where it may light up depending on an active high (common cathode circuit) or active low input (common anode circuit). The BASYS 3 board uses a common anode circuit.
- Instead of writing logic equations using k-maps for inputs of each segments we can use a lookup table which can be built using a ROM (Read Only Memory).
- ROM can be used to store constant values.
- In an Asynchronous ROM, its operation is not controlled by a clock.