Index Number - 210098R

Name - De Silva APC

Lab task –

Designing and developing a seven-segment display that displays numbers with the use of a lookup table to map which light segments should be on when displaying each number. Further more, the previously developed components were used (previously designed AU+) to make it more functional by displaying the number which was added together and outputted by the AU.

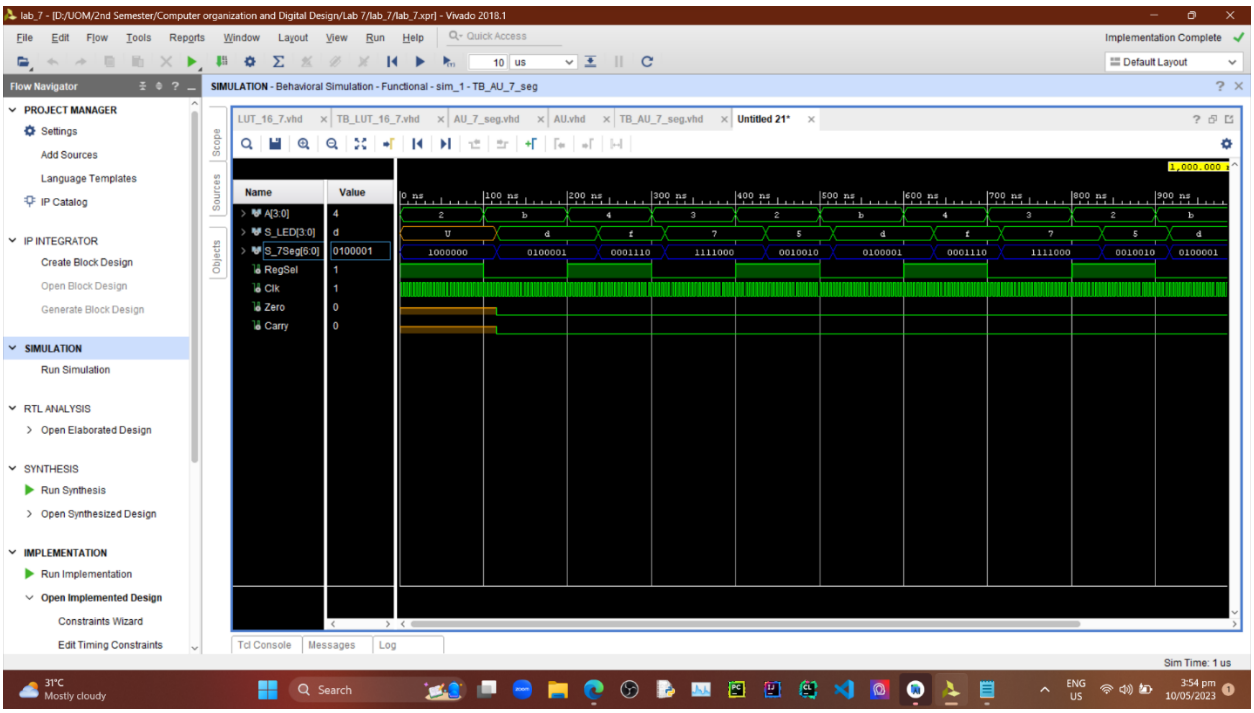
Filled Lookup Table –

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

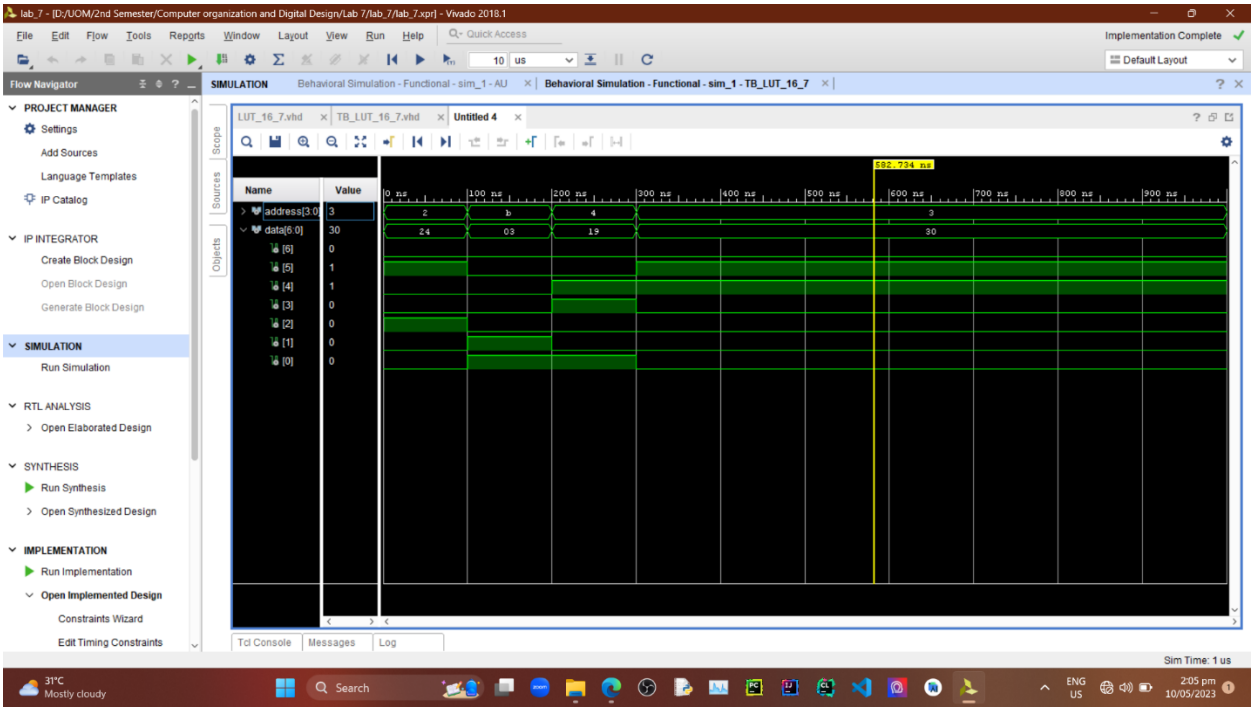
Screenshots –

(Final simulation)

*(Segmants of four bits from the end of binary conversion of the index number has been used to test)



(Testing the Lookup Table implementation)



VHD Files –

AU Design file(AU_7_Seg.vhd) –

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 05/10/2023 02:12:58 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);
          Clk : in STD_LOGIC;
          RegSel : in STD_LOGIC;
          S_LED : out STD_LOGIC_VECTOR (3 downto 0);
          S_7Seg : out STD_LOGIC_VECTOR (6 downto 0);
          Carry : out STD_LOGIC;
          Zero : out STD_LOGIC);
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 OutputFeeder : std_logic_vector ( 3 downto 0 ):="0000";
begin
AU_0 : AU
   PORT MAP(
   A => A,
   RegSel =>RegSel,
   Clk =>Clk,
   Zero => Zero,
   Carry => Carry,
   S => OutputFeeder
   );
S_LED <= OutputFeeder;

LUT_16_7_0 : LUT_16_7
   PORT MAP(
   address =>Outputfeeder,
   data =>S_7Seg
   );

end Behavioral;
```

LUT Design file(LUT_16_7.vhd) –

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 05/10/2023 01:22:53 PM
-- 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; -- Importing numeric library

-- 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;
```

LUT Testbench file(TB_LUT_16_7.vhd) –

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 05/10/2023 01:40:13 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
   address <= "0010";
   WAIT FOR 100 ns;
   address <= "1011";
   WAIT FOR 100 ns;
   address <= "0100";

   WAIT FOR 100 ns;
   address <= "0011";
   WAIT;
end process;
end Behavioral;
```

Final Testbench file(TB_AU_7_seg.vhd) –

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 05/10/2023 02:32:43 PM
-- 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);
          Clk : in STD_LOGIC;
          RegSel : in STD_LOGIC;
          S_LED : out STD_LOGIC_VECTOR (3 downto 0);
          S_7Seg : out STD_LOGIC_VECTOR (6 downto 0);
          Carry : out STD_LOGIC;
          Zero : out STD_LOGIC);
end component;
signal A,S_LED : 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,
      S_LED =>S_LED,
      Regsel => Regsel,
      Clk => Clk,
      Zero => Zero,
      Carry =>Carry,
      S_7Seg => S_7Seg
   );
process
begin
   Clk <= not(Clk);
   wait for 2ns;
end process;

process
```

```
begin
   A<= "0010";
   RegSel <= '1';
   wait for 100ns;

   RegSel <= '0';
   A<= "1011";
   wait for 100ns;
   RegSel <= '1';
   A <= "0100";
   wait for 100ns;
   RegSel <= '0';
   A <= "0011";
   wait for 100ns;



end process;

end Behavioral;
```

How to Display results using a single 7 segment Display –

Here we are using 4 bit numbers as input. Hence the output would be a 4 bit number and may have another carry bit. Using 4 bits we can display 16 different numbers and that exactly matches with how many different numbers can be displayed using a single 7 segmant display using hexadecimal representations. So by using extra indicator to indicate the carry bit we can display the results without a issue using both indicator and the seven segment display.

Conclusions –

Basys 3 board has a common anode 7 segmant displays which can display up to 4 digits.(i.e. four 7 segmant displays in a sequence.)

Using a 7 segmant display we can display every digit used in hexadecimal number system too.

Lookup tables can be used to implement circuits by using it as a memory(ROM).