LAB 07 COD – 210495G – B.A Nipun Viraj

Lab Task

Using a 7 segment LED to show the output of a 4 bit number by choosing which of the segments to output.

Filled Up 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	0	0	0	1	1	0	0
0	1	0	1	5	1	1	0	0	1	0	0
0	1	1	0	6	1	1	0	0	0	0	0
0	1	1	1	7	1	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

LUT VHDL File

LUT_16_7.vhd

E:/Xilinx/Projects/Lab7/Lab7.srcs/sources_1/new/LUT_16_7.vhd

```
22  library IEEE;
23  use IEEE.STD_LOGIC_1164.ALL;
24  use ieee.numeric_std.all;
25
26  -- Uncomment the following library declaration if using
27  -- arithmetic functions with Signed or Unsigned values
28  --use IEEE.NUMERIC_STD.ALL;
29
30  -- Uncomment the following library declaration if instantiating
31  -- any Xilinx leaf cells in this code.
32  --library UNISIM;
33  --use UNISIM.VComponents.all;
34
35  entity LUT_16_7 is
36      Port ( address : in STD_LOGIC_VECTOR (3 downto 0);
37             data : out STD_LOGIC_VECTOR (6 downto 0));
38  end LUT_16_7;
39  architecture Behavioral of LUT_16_7 is
40  type rom_type is array (0 to 15) of std_logic_vector(6 downto 0);
41  signal sevenSegment_ROM : rom_type := (
42  "1000000", -- 0
43  "1111001", --1
44  "0100100", --2
45  "0110000", --3
46  "0011000", --4
47  "0010011", --5
48  "0000011", --6
49  "1111001", --7
50  "0000000", --8
51  "0010000", --9
52  "0001000", -- a
53  "0000011", --b
54  "1000110", --c
55  "0100001", --d
56  "0000110", --e
57  "0001110" -- f
58   );
59  begin
60  data <= sevenSegment_ROM(to_integer(unsigned(address)));
61  end Behavioral;
```

LUT – TB File

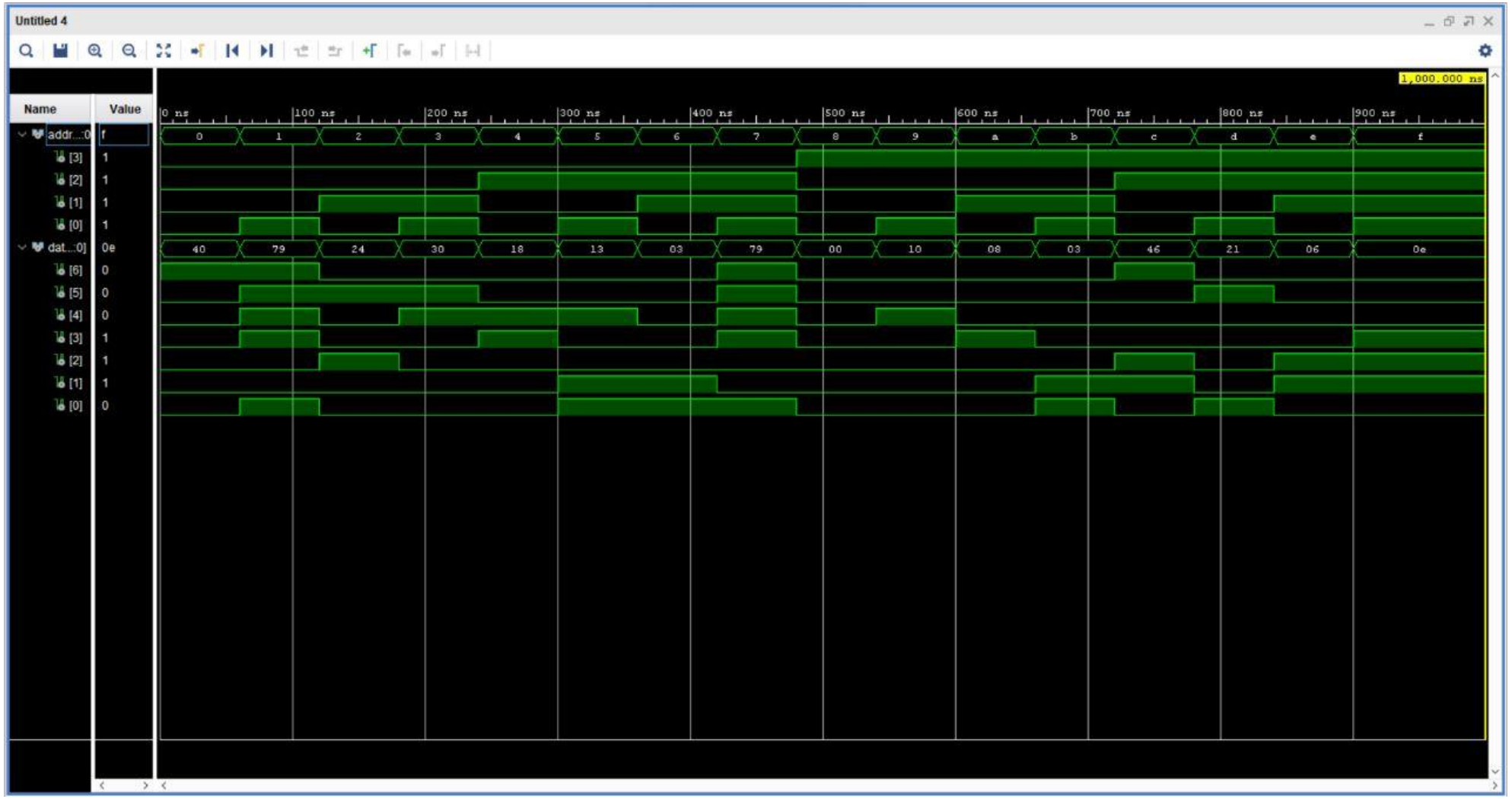
TB_LUT_16_7.vhd

E:/Xilinx/Projects/Lab7/Lab7.srcs/sim_1/new/TB_LUT_16_7.vhd

```
22  library IEEE;
23  use IEEE.STD_LOGIC_1164.ALL;
24  use ieee.numeric_std.all;
25
26  -- Uncomment the following library declaration if using
27  -- arithmetic functions with Signed or Unsigned values
28  --use IEEE.NUMERIC_STD.ALL;
29
30  -- Uncomment the following library declaration if instantiating
31  -- any Xilinx leaf cells in this code.
32  --library UNISIM;
33  --use UNISIM.VComponents.all;
34
35  entity TB_LUT_16_7 is
36  --  Port ( );
37  end TB_LUT_16_7;
38
39  architecture Behavioral of TB_LUT_16_7 is
40  component LUT_16_7 is Port
41  (address : in STD_LOGIC_VECTOR (3 downto 0);
42  data : out STD_LOGIC_VECTOR (6 downto 0));
43  end component;
44
45  signal address0:std_logic_vector(3 downto 0);
46  signal data0:std_logic_vector(6 downto 0);
47
48  begin
49  UUT:LUT_16_7 PORT MAP(
50  address => address0,
51  data => data0);
```

```
49  UUT:LUT_16_7 PORT MAP(
50  address => address0,
51  data => data0);
52  process
53      begin
54          for i in 0 to 15 loop
55              address0 <= std_logic_vector(to_unsigned(i, address0'length));
56              wait for 60 ns;
57          end loop;
58          wait;
59      end process;
60
61  end Behavioral;
```

LUT – Timing Diagram



AU_7_Seg_ VHDL

AU_7_seg.vhd

E:/Xilinx/Projects/Lab7/Lab7.srcs/sources_1/new/AU_7_seg.vhd

```
22  library IEEE;
23  use IEEE.STD_LOGIC_1164.ALL;
24
25  -- Uncomment the following library declaration if using
26  -- arithmetic functions with Signed or Unsigned values
27  --use IEEE.NUMERIC_STD.ALL;
28
29  -- Uncomment the following library declaration if instantiating
30  -- any Xilinx leaf cells in this code.
31  --library UNISIM;
32  --use UNISIM.VComponents.all;
33
34  entity AU_7_seg is
35   Port (
36   A : in STD_LOGIC_VECTOR (3 downto 0);
37   RegSel : in STD_LOGIC;
38   Clk : in STD_LOGIC;
39   S_LED : out STD_LOGIC_VECTOR (3 downto 0);
40   S_7Seg: out STD_LOGIC_VECTOR (6 downto 0);
41   Zero : out STD_LOGIC;
42   Carry : out STD_LOGIC);
43  end AU_7_seg;
44
45  architecture Behavioral of AU_7_seg is
46
47  component AU PORT
48  (A : in STD_LOGIC_VECTOR (3 downto 0);
49   RegSel : in STD_LOGIC;
50   Clk : in STD_LOGIC;
51   S : out STD_LOGIC_VECTOR (3 downto 0);
52   Zero : out STD_LOGIC;
53   Carry : out STD_LOGIC);
54  end component;
55
56  component LUT_16_7 PORT
57  (address : in STD_LOGIC_VECTOR (3 downto 0);
58   data : out STD_LOGIC_VECTOR (6 downto 0));
59  end component;
60
61  signal address0 : STD_LOGIC_VECTOR(3 DOWNTO 0);
```

```
60
61  signal address0 : STD_LOGIC_VECTOR(3 DOWNTO 0);
62
63  begin
64  AU_0:AU
65  PORT MAP(
66  A=>A,
67  RegSel=>RegSel,
68  Clk=>Clk,
69  Zero=>Zero,
70  Carry=>Carry);
71
72  LUT_16_7_0:LUT_16_7
73  PORT MAP(
74  address => address0,
75  data => S_7Seg
76  );
77
78  S_LED <= address0;
79
80  end Behavioral;
81
```

AU_7_Seg_Testbench

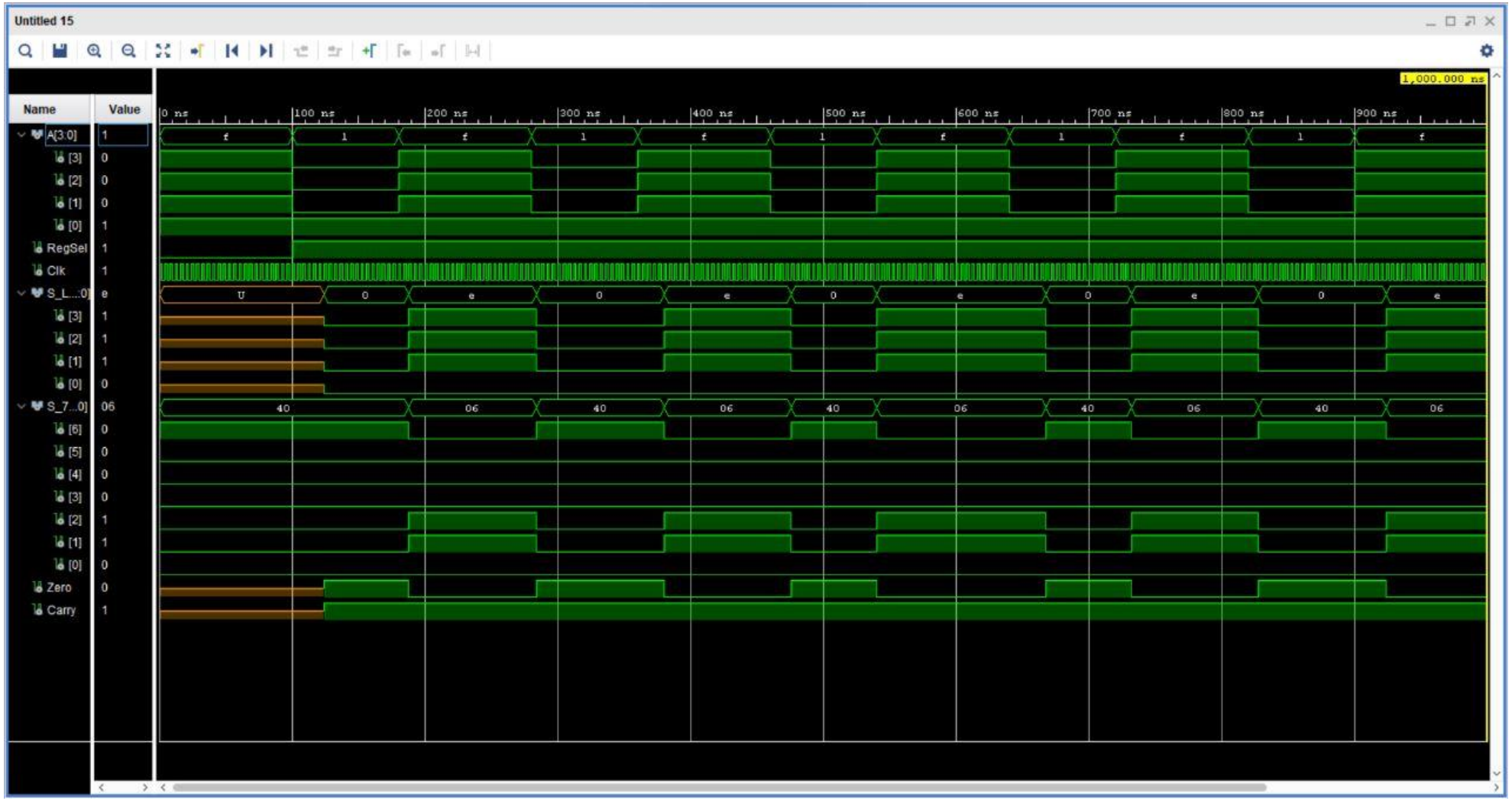
```
34  entity TB_AU_7_Seg is
35  --  Port ( );
36  end TB_AU_7_Seg;
37
38  architecture Behavioral of TB_AU_7_Seg is
39  component AU_7_seg
40      Port (
41        A : in STD_LOGIC_VECTOR (3 downto 0);
42        RegSel : in STD_LOGIC;
43        Clk : in STD_LOGIC;
44        S_LED : out STD_LOGIC_VECTOR (3 downto 0);
45        S_7Seg: out STD_LOGIC_VECTOR (6 downto 0);
46        Zero : out STD_LOGIC;
47        Carry : out STD_LOGIC
48      );
49  end component;
50
51  signal A : STD_LOGIC_VECTOR (3 downto 0) := "0000";
52  signal RegSel : STD_LOGIC := '0';
53  signal Clk : STD_LOGIC := '0';
54
55  signal S_LED : STD_LOGIC_VECTOR (3 downto 0);
56  signal S_7Seg : STD_LOGIC_VECTOR (6 downto 0);
57  signal Zero : STD_LOGIC;
58  signal Carry : STD_LOGIC;
59
60
61  begin
```

TB_AU_7_Seg.vhd

E:/Xilinx/Projects/Lab7/Lab7.srcs/sim_1/new/TB_AU_7_Seg.vhd

```
47        Carry : out STD_LOGIC
48      );
49  end component;
50
51  signal A : STD_LOGIC_VECTOR (3 downto 0) := "0000";
52  signal RegSel : STD_LOGIC := '0';
53  signal Clk : STD_LOGIC := '0';
54
55  signal S_LED : STD_LOGIC_VECTOR (3 downto 0);
56  signal S_7Seg : STD_LOGIC_VECTOR (6 downto 0);
57  signal Zero : STD_LOGIC;
58  signal Carry : STD_LOGIC;
59
60
61  begin
62  -- Instantiate the DUT
63    uut: AU_7_seg port map (
64      A => A,
65      RegSel => RegSel,
66      Clk => Clk,
67      S_LED => S_LED,
68      S_7Seg => S_7Seg,
69      Zero => Zero,
70      Carry => Carry
71    );
72
73    process begin
74    Clk <= not (Clk);
75    wait for 2ns;
76    end process;
77
78    process begin
79    A<="1111";
80    wait for 100 ns;
81    RegSel<='1';
82    A<="0001";
83    wait for 80 ns;
84    end process;
85
86  end Behavioral;
```

AU_7_Seg_Timing Diagram



Conclusion

- We can easily design a lookup table and use it to build a 7 segment LED output.