Lab-2 Behavioral Simulation

CS1050

Lab Report

Name – Vidushini A.O.

Index No – 210670N

Assigned task

Assigned task was to create and stimulate a logic circuit to indicate working state of a Power Station with 3 generators . There , if all three generators are working , green light has to on while if two out of three generators work, the amber light has to on. However if less than 2 generators work, then the red light should on.

Building the Boolean Representation

For Green Light

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

This can be implemented by the Boolean Expression

A AND B AND C

For Amber Light

A	B	C	Y	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	→ NOT(A) AND B AND C
1	0	0	0	
1	0	1	1	→ A AND NOT(B) AND C
1	1	0	1	→ A AND B AND NOT(C)
1	1	1	0	

This can be implemented by the Boolean Expression

(NOT A AND B AND C) OR (A AND (NOT B) AND C) OR (A AND B AND (NOT C))

For Red Light

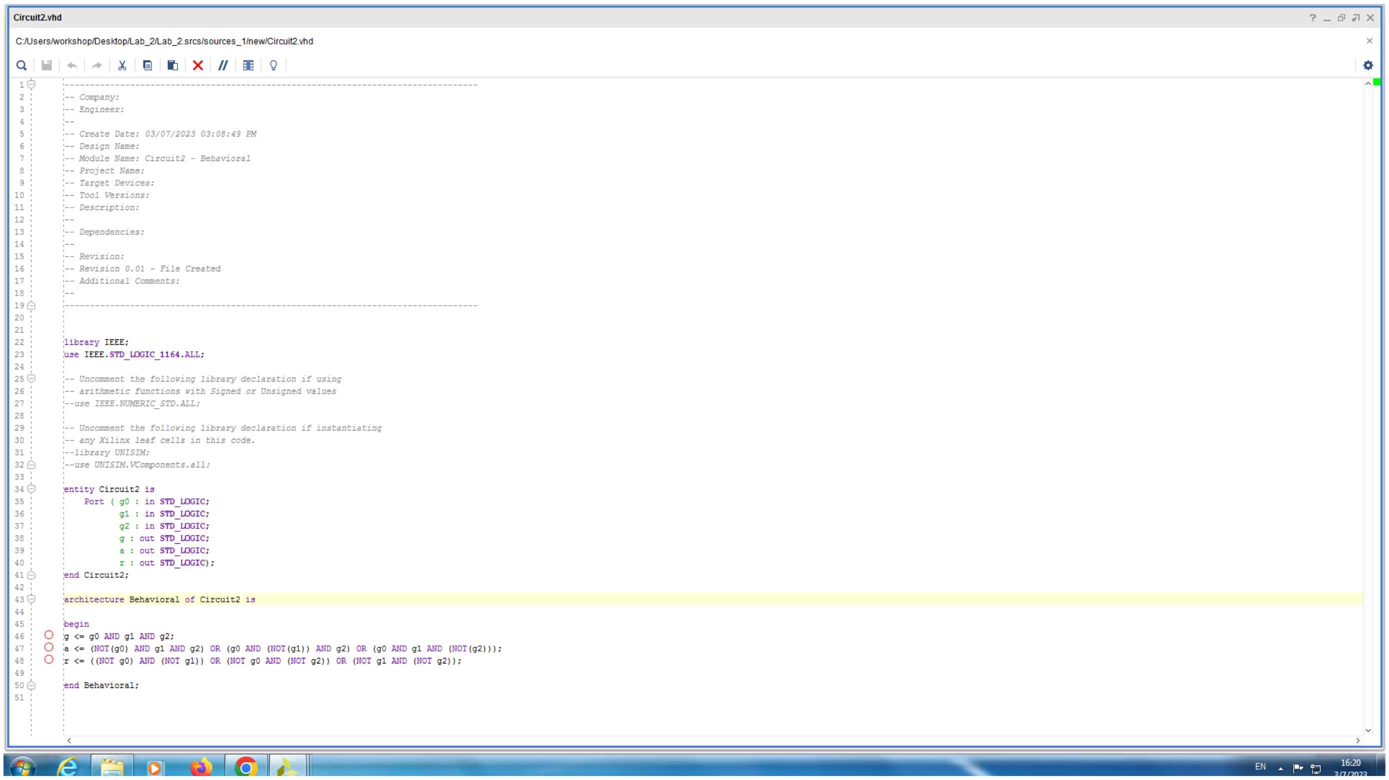
A	B	C	Z	
0	0	0	1	→ NOT(A) AND NOT(B) AND NOT(C)
0	0	1	1	→ NOT(A) AND NOT(B) AND C
0	1	0	1	→ NOT(A) AND B AND NOT(C)
0	1	1	0	
1	0	0	1	→ A AND NOT(B) AND NOT(C)
1	0	1	0	
1	1	0	0	
1	1	1	0	

This can be implemented by the Boolean Expression

(NOT A AND (NOT B)) OR ((NOT B) AND (NOT C)) OR ((NOT(A) AND (NOT C))

VHDL design Source code for the circuit.

(Here, switches A, B, C are represented as g0, g1 and g2 respectively while the LED s X, Y and Z are represented as g, a, and r respectively)

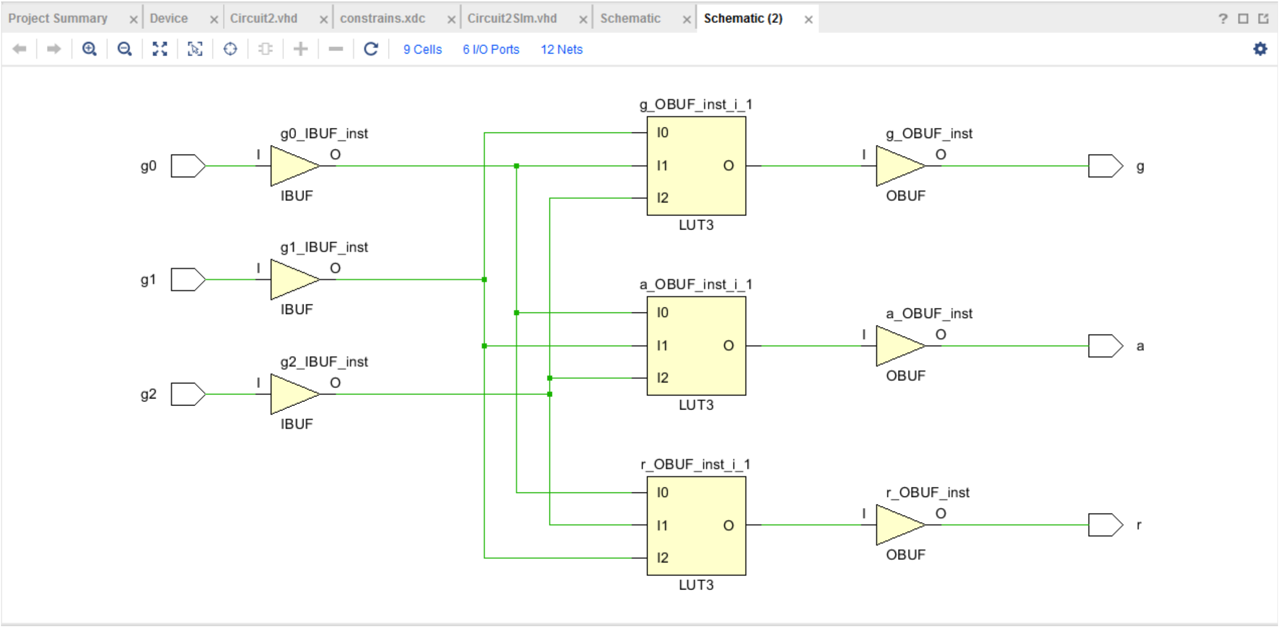


Circuit2.vhd

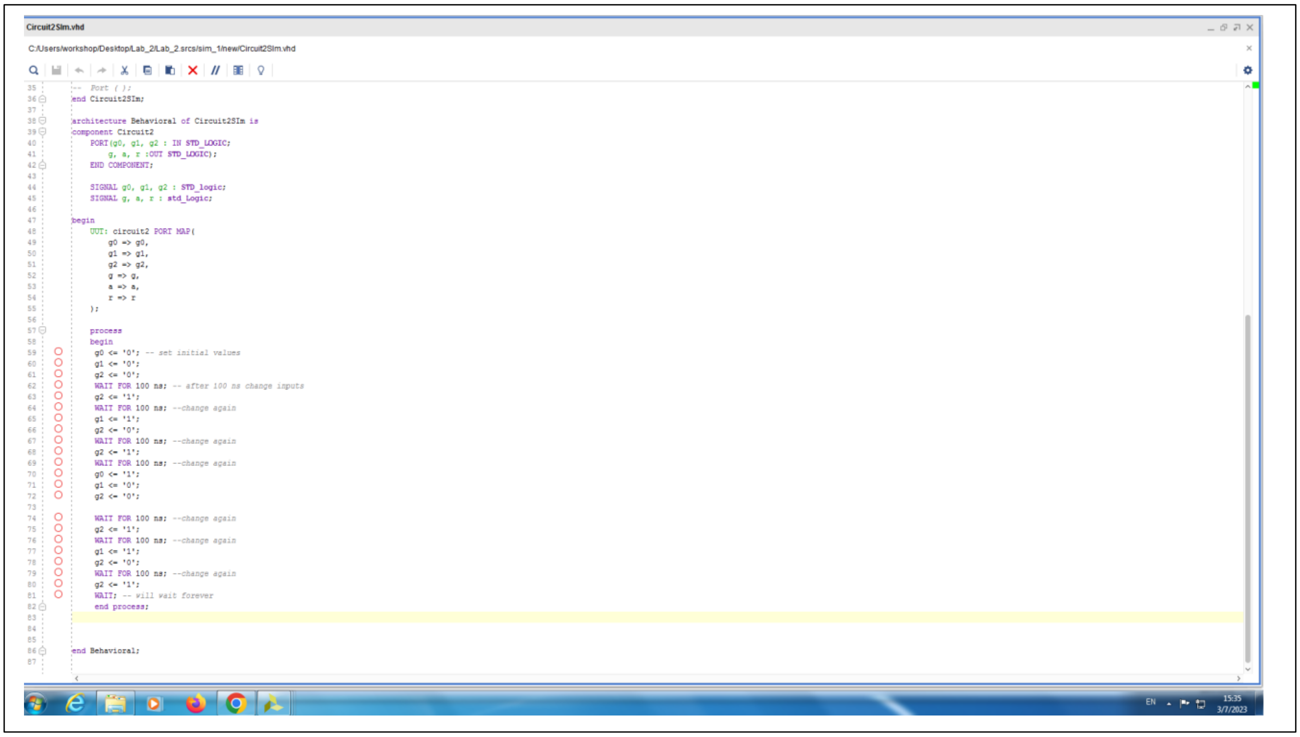
C:/Users/workshop/Desktop/Lab_2/Lab_2.srcs/sources_1/new/Circuit2.vhd

```
----------------------------------------------------------------------------------
-- Company: 
-- Engineer: 
-- 
-- Create Date: 03/07/2023 03:08:49 PM
-- Design Name: 
-- Module Name: Circuit2 - 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 Circuit2 is
    Port ( g0 : in STD_LOGIC;
           g1 : in STD_LOGIC;
           g2 : in STD_LOGIC;
           g : out STD_LOGIC;
           a : out STD_LOGIC;
           r : out STD_LOGIC);
end Circuit2;

architecture Behavioral of Circuit2 is

begin
g <= g0 AND g1 AND g2;
a <= (NOT(g0) AND g1 AND g2) OR (g0 AND (NOT(g1)) AND g2) OR (g0 AND g1 AND (NOT(g2)));
r <= ((NOT g0) AND (NOT g1)) OR (NOT g0 AND (NOT g2)) OR (NOT g1 AND (NOT g2));

end Behavioral;

```

Schematic circuit from Vivado



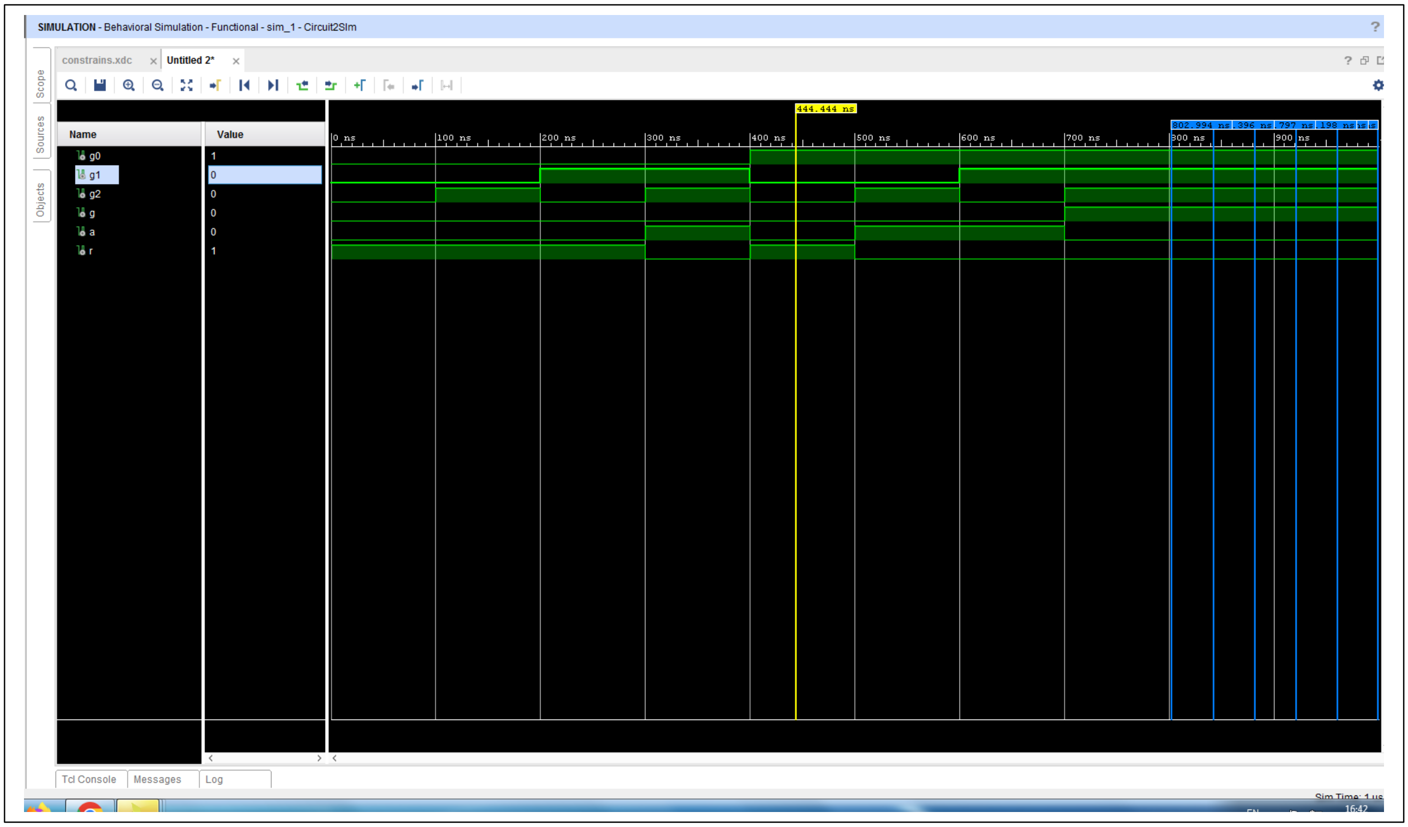
Test Bench Code



Circuit2SIm.vhd
C:/Users/workshop/Desktop/Lab_2/Lab_2.srcs/sim_1/new/Circuit2SIm.vhd
EN
15:35
3/7/2023

```vhdl
-- Port ( );
end Circuit2SIm;

architecture Behavioral of Circuit2SIm is
component Circuit2
    PORT(g0, g1, g2 : IN STD_LOGIC;
        g, a, r :OUT STD_LOGIC);
    END COMPONENT;

    SIGNAL g0, g1, g2 : STD_logic;
    SIGNAL g, a, r : std_Logic;

begin
    UUT: circuit2 PORT MAP(
        g0 => g0,
        g1 => g1,
        g2 => g2,
        g => g,
        a => a,
        r => r
    );

    process
    begin
     g0 <= '0'; -- set initial values
     g1 <= '0';
     g2 <= '0';
     WAIT FOR 100 ns; -- after 100 ns change inputs
     g2 <= '1';
     WAIT FOR 100 ns; --change again
     g1 <= '1';
     g2 <= '0';
     WAIT FOR 100 ns; --change again
     g2 <= '1';
     WAIT FOR 100 ns; --change again
     g0 <= '1';
     g1 <= '0';
     g2 <= '0';

     WAIT FOR 100 ns; --change again
     g2 <= '1';
     WAIT FOR 100 ns; --change again
     g1 <= '1';
     g2 <= '0';
     WAIT FOR 100 ns; --change again
     g2 <= '1';
     WAIT; -- will wait forever
     end process;



end Behavioral;
```

Timing Diagram for XSim



SIMULATION - Behavioral Simulation - Functional - sim_1 - Circuit2SIm

Name	Value
g0	1
g1	0
g2	0
g	0
a	0
r	1

Conclusion

Logic stimulator allowed to observe all the outputs of the circuit which was built before it was implemented on hardware, BASYS 3 . This was time saving and less error prone as it was easy to debug rather than testing the circuit on the hardware, Moreover, using a simulator is a cheaper option rather than designing hardware prototype.