Lab 2 – Behavioral Simulation

CS1050 Computer Organization and Digital Design

Name: K. Oshadi Saumya Perera

Index no: 200458M

Lab Task

In this lab, we will build a logic circuit that can indicate whether all generators of a power station are correctly functioning. Consider a power station with 3 generators. Suppose the three generators are providing power to a city and any 2 generators out of the 2 can provide enough power. The third generator is added for redundancy in case one generator fails. However, a single power station cannot provide enough power for the city.

Suppose working status of each power station is indicated using a switch, where switches A (SW2), B (SW1), and C (SW0) represent the three generators.

- Green light should be on when all three generators are functioning correctly.
- Amber light should be on only when 2 generators are functioning correctly.
- Red light should come up if less than 2 generators are functioning correctly.

Simplified Boolean representation

1. Truth Table

A	B	C	Y1	Y2	Y3
0	0	0	0	0	1
0	0	1	0	0	1
0	1	0	0	0	1
0	1	1	0	1	0
1	0	0	0	0	1
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	1	0	0

2. K-maps

Y1: Green

BC / A	00	01	11	10
0	0	0	0	0
1	0	0	1	0

Y1 = A.B.C

Y2: Amber

BC / A	00	01	11	10
0	0	0	1	0
1	0	1	0	1

Y2 = A.B'.C + A'.B.C + A.B.C'

Y3: Red

BC / A	00	01	11	10
0	1	1	0	1
1	1	0	0	0

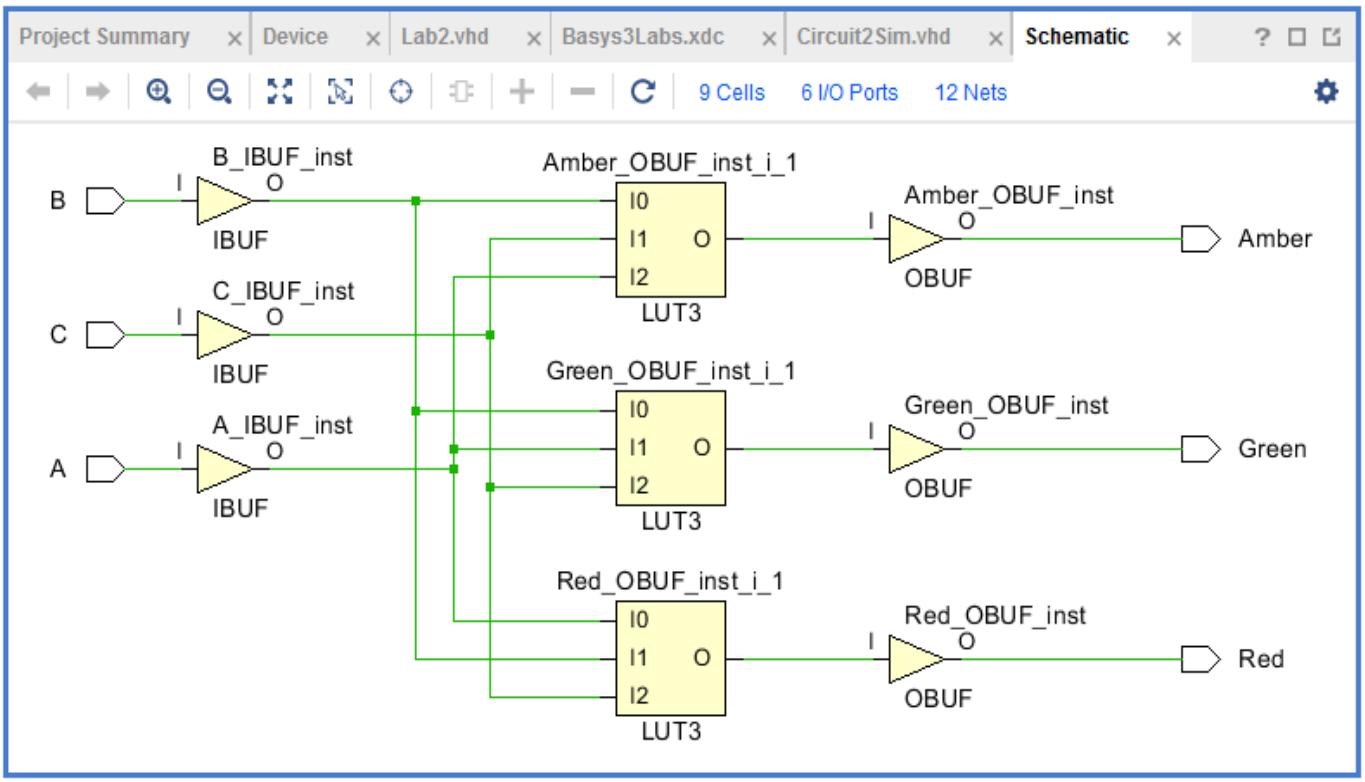
Y3 = B'.C' + A'.B' + A'.C'

Full VHDL design source code

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 05/19/2022 02:13:58 PM
-- Design Name:
-- Module Name: Lab2 - 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 Lab2 is
   Port ( A : in STD_LOGIC;
          B : in STD_LOGIC;
          C : in STD_LOGIC;
          Green : out STD_LOGIC;
          Amber : out STD_LOGIC;
          Red : out STD_LOGIC);
end Lab2;

architecture Behavioral of Lab2 is
```

```
begin
Green <= A AND B AND C;
--Amber <= A XNOR B XNOR C;
Amber <= (A AND NOT(B) AND C) OR (NOT(A) AND B AND C) OR (A AND B AND NOT(C));
Red <= (NOT(A) AND NOT(C)) OR (NOT(A) AND NOT(B)) OR (NOT(B) AND NOT(C));

end Behavioral;
```

Schematic circuit from Vivado



Test bench code

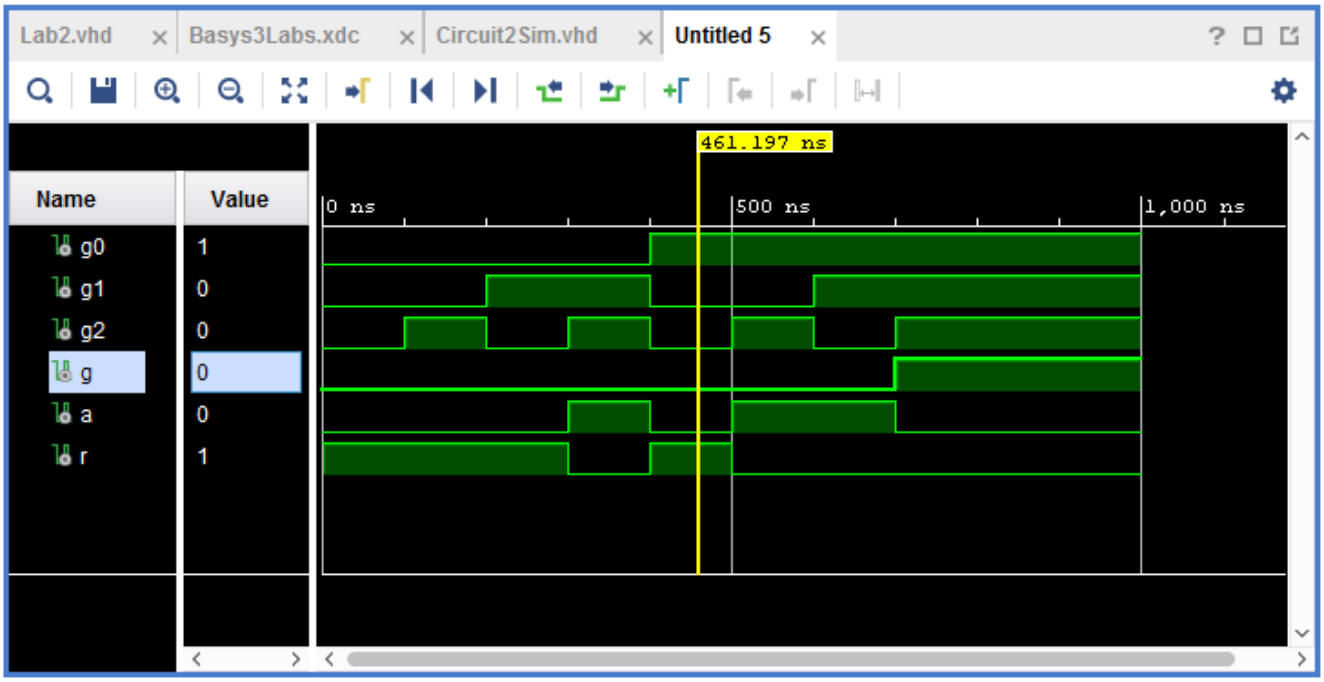
```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 05/19/2022 02:36:30 PM
-- Design Name:
-- Module Name: Circuit2Sim - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- 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 Circuit2Sim is
--  Port ( );
end Circuit2Sim;

architecture Behavioral of Circuit2Sim is

   COMPONENT Lab2
      PORT(
         A,B,C : IN STD_LOGIC;
         Green,Amber,Red : OUT STD_LOGIC
      );
   END COMPONENT;
   signal g0,g1,g2 : STD_LOGIC;
   signal g,a,r : STD_LOGIC;
begin

UUT : Lab2
   PORT MAP(
      A => g0,
      B => g1,
      C => g2,
      Green => g,
      Amber => a,
      Red => 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 from XSim showing all possible inputs and outputs



Conclusions

By completing this lab, we learnt how to simulate a logic circuit using the software.

Goals achieved:

- Design and develop a simple logic circuit using schematics.
- Verify its functionality via simulation