Index Number – 210098R

Student Name – De Silva A.P.C.

Assigned lab task - Designing a circuit to indicate the functionality of three power generators in a power station (Basys3 board to be used.) We were asked to obtain the needed logic and then implement circuit logic using VHDL and then simulate the circuit and check whether its' running as intended by observing test input set and its' output.

Steps of simplifying Boolean Representation –

1. **First identified How many inputs and outputs are there.**
2. **Then drew the truth table for those inputs (3) and outputs(3).**
3. **Then using "Sum of product method" Wrote the exact Boolean representations related to the truth table.**
4. **Then simplified possible logic to get the final logic.**

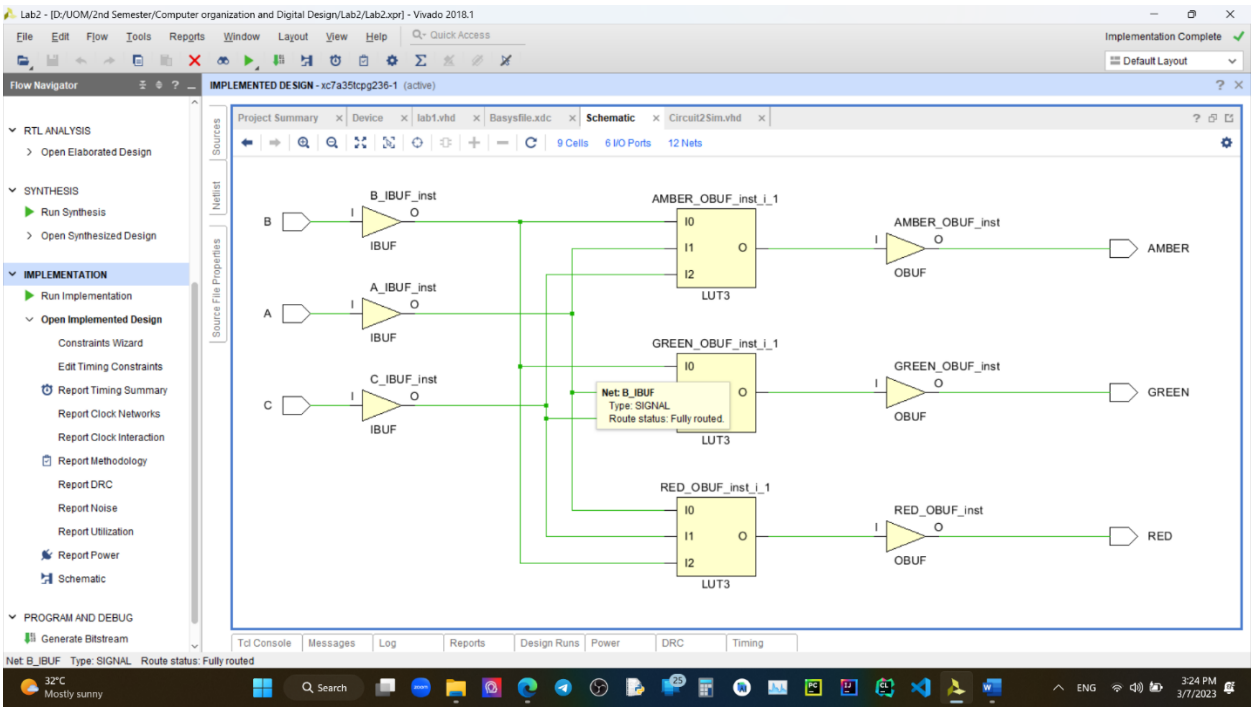
Full VHDL Design source code –

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 03/07/2023 02:14:34 PM
-- Design Name:
-- Module Name: lab1 - 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 lab1 is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
```

```
        C : in STD_LOGIC;
        RED : out STD_LOGIC;
        AMBER : out STD_LOGIC;
        GREEN : out STD_LOGIC);
end lab1;

architecture Behavioral of lab1 is
signal A_AND_B_out : std_logic;
signal B_AND_C_out : std_logic;
signal A_AND_C_out : std_logic;
begin
A_AND_B_out <= A and B;
B_AND_C_out <= C and B;
A_AND_C_out <= A and C;
GREEN <= A_AND_B_out and C;

AMBER <= (B_AND_C_out and not A) OR (A_AND_B_out and not C) OR (A_AND_C_out and not
B);

RED <= (not A and not B) OR (not A and not C) OR (not B and not C);

end Behavioral;
```

Schematic diagram –



Test Bench Code –

-- Company:

-- Engineer:

--

-- Create Date: 03/07/2023 02:23:09 PM

-- Design Name:

-- Module Name: Circuit2Sim - Behavioral

-- Project Name:

-- Target Devices:

-- Tool Versions:

-- Description:

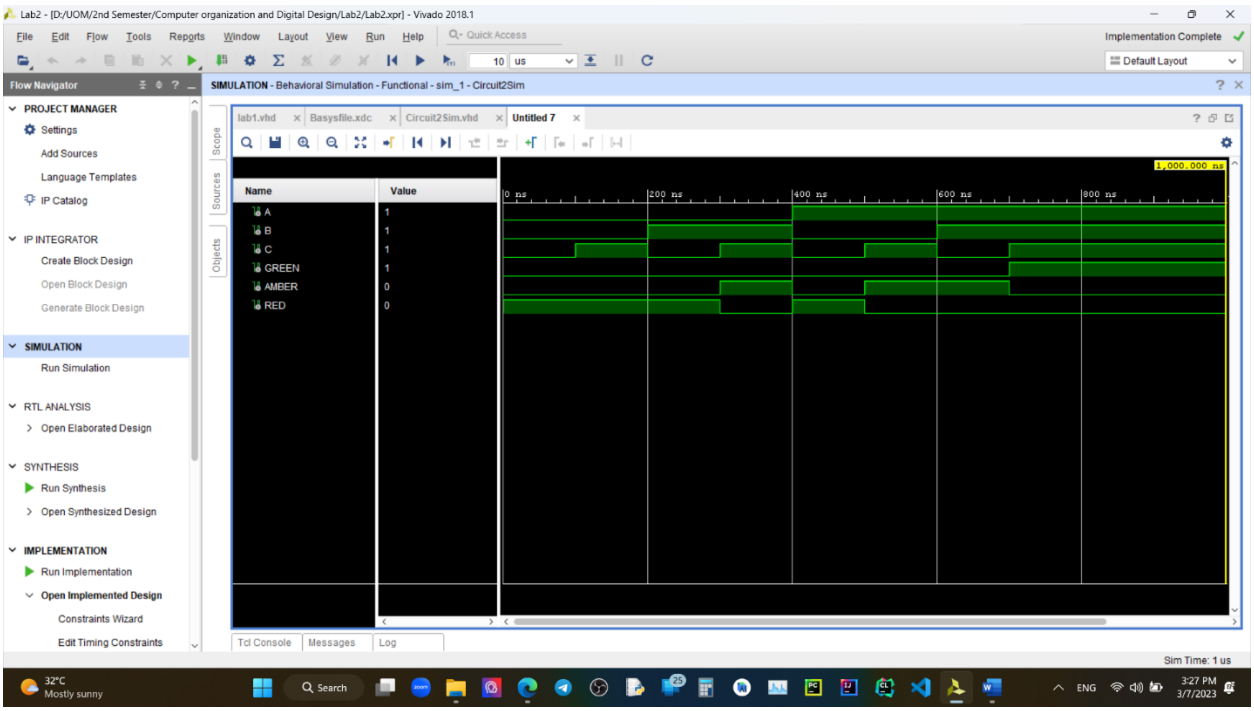
--

```
-- 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 lab1
```

```
   PORT( A,B,C : IN STD_LOGIC;
GREEN,AMBER,RED : OUT STD_LOGIC);
END COMPONENT;
SIGNAL A,B,C          : std_logic;
SIGNAL GREEN,AMBER,RED : std_logic;
begin
UUT : lab1 PORT MAP(
      A => A,
      B => B,
      C =>C,
      GREEN => GREEN,
      AMBER => AMBER,
      RED => RED
);

process
begin
       A <= '0';
       B <= '0';
       C <= '0';

       WAIT FOR 100 ns;
       C <='1';

       WAIT FOR 100 ns;
       B <= '1';
       C <= '0';
```

```
        WAIT FOR 100 ns;
        C <= '1';

        WAIT FOR 100 ns;
        A <= '1';
        B <= '0';
        C <= '0';

        WAIT FOR 100 ns;
        C <= '1';

        WAIT FOR 100 ns;
        B <= '1';
        C <= '0';

        WAIT FOR 100 ns;
        C <= '1';

        WAIT;
    end process;
end Behavioral;
```

Timing diagram from Xsim –



Conclusion - Using Vivado we can not only design logic circuits but also we can test them with ease before going to the hardware level implementations. This saves not only time but also extra costs attached to it in case where the design had some flows that is not recognized before real world implementation.