TRAFFIC LIGHT CONTROLLER

-Tarun Kumar(14114068)

Introduction:

This traffic light controller model is designed for junction as shown:

On every side traffic has privilege to go left freely, while for other directions the signals are implemented in this project.

Overview:

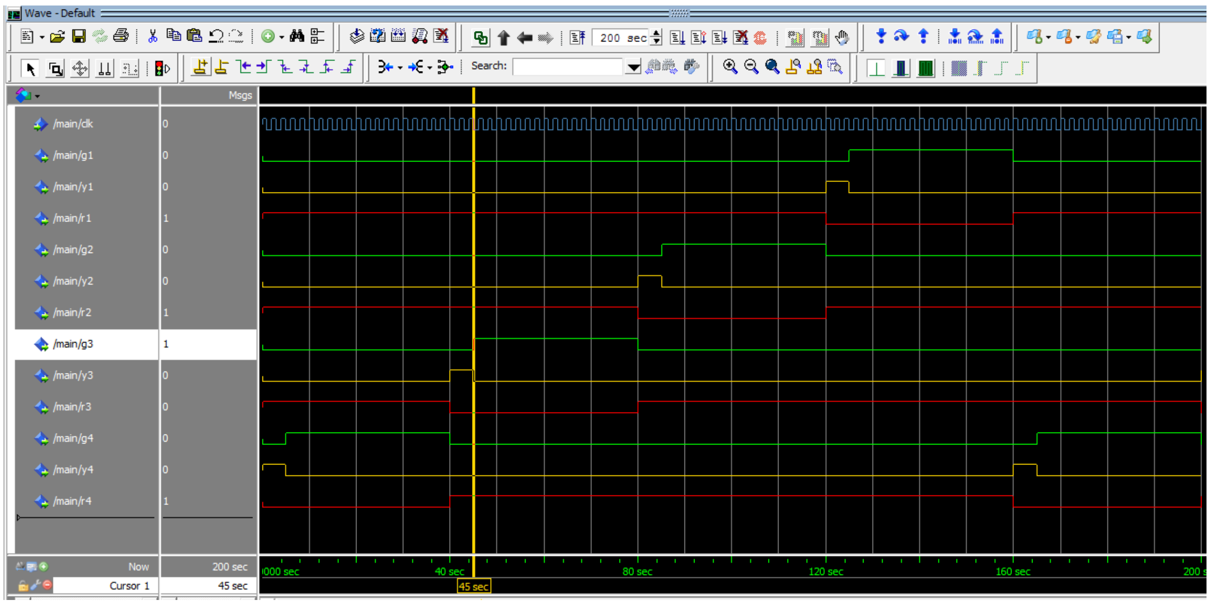
Traffic light controller model is designed for intersection of two roads. We are considering 3 signals for each side. Hence in total there are 12 *signals. Each of these signals will be connected to LED light which will be on if high is there on signal.*

Description:

We are taking 12 signals marked as r1,y1,g1,r2,y2,g2,r3,y3,g3,r4,g4 for corresponding red , yellow and green light for each of four sides. In VHDL design we are using array binary signals of 12 bits. Each of the pattern of this binary signal array represents a different state of our machine.

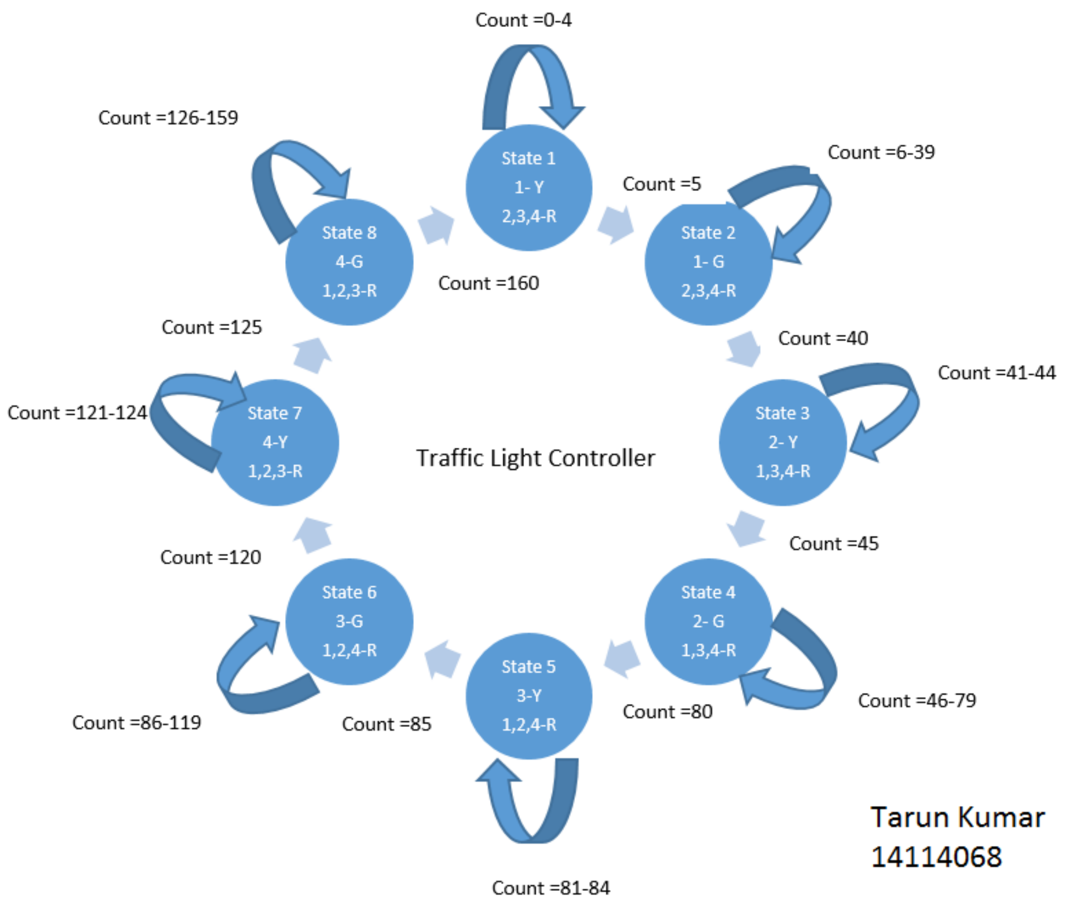
This project is compiled and simulated with Model Sim PE Student Edition.

Wave form of clock and the outputs is shown below:-



High of any wave tells that corresponding LED is on. Eg. - At 45 sec , we can see that r1,r2,r4 are high, and yellow of third side is turning off and its green is becoming high.

FSM:



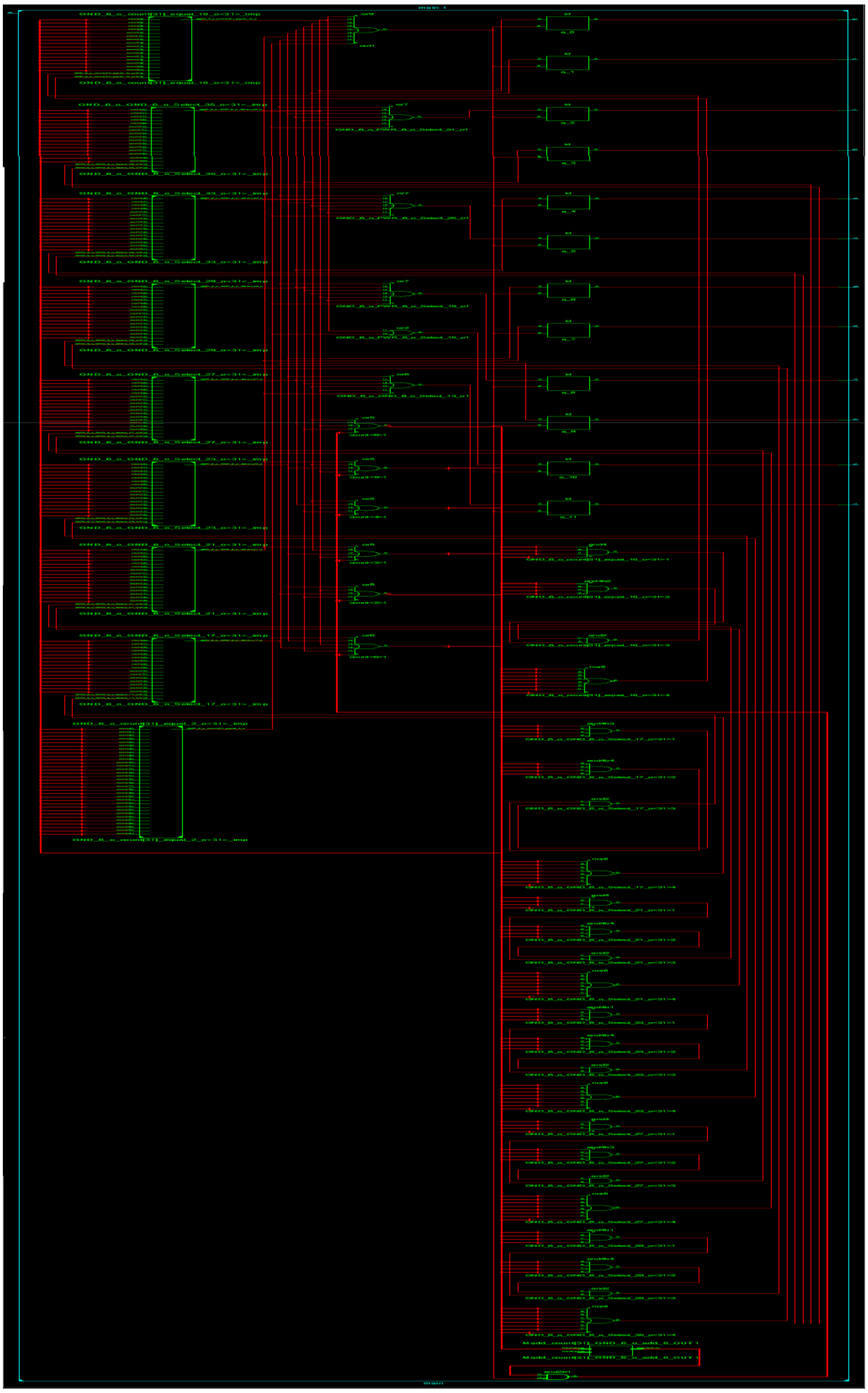
Code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.NUMERIC_STD.ALL;

entity main is
  Port (
          clk : in std_logic; --- Clock with provided frequency
          g1 : out std_logic; --
```

```
                y1 : out std_logic; --- Outputs for fist side
                r1 : out std_logic; --
                g2 : out std_logic; --
                y2 : out std_logic; --- Outputs for second side
                r2 : out std_logic; --
                g3 : out std_logic; --
                y3 : out std_logic; --- Outputs for third side
                r3 : out std_logic; --
                g4 : out std_logic; --
                y4 : out std_logic; --- Outputs for fourth side
                r4 : out std_logic);
end main;

architecture Traffic of main is
        signal count : integer := 0;
        signal a: std_logic_VECTOR (11 downto 0) := "010100100100" ;
        begin PROCESS (clk)
        BEGIN case count is -- Cases and their corresponding signal variations.
                when 0 => a <= "010100100100"; count <= count+1;
                when 5 => a <= "001100100100";  count <= count+1;
                when 40 => a <= "100010100100";  count <= count+1;
                when 45 => a <= "100001100100"; count <= count+1;
                when 80 => a <= "100100010100"; count <= count+1;
```

```
            when 85 => a <= "100100001100"; count <= count+1;
            when 120 => a <= "100100100010"; count <= count+1;
            when 125 => a <= "100100100001"; count <= count+1;
            when 160 => a <= "010100100100"; count <= 0;
                when others => count <= count+1;
        END case;
        r1 <= a(2); y1 <= a(1); g1 <= a(0);
        r2 <= a(5); y2 <= a(4); g2  <=  a(3);  --  Assigning  all  signals  their
corresponding values.
        r3 <= a(8); y3 <= a(7); g3 <= a(6);
        r4 <= a(11);        y4 <= a(10);        g4 <= a(9);
        END PROCESS;
end Traffic;
```

<u>RTL Synthesis Output:</u>



Thank You.