TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

HIMALAYA COLLEGE OF ENGINEERING

THIRD YEAR EMBEDED SYSTEM PROJECT REPORT

ON

TRAFFIC LIGHT CONTROLLER IN VHDL

SUBMITTED TO:

Er. SHIVA RAJ LUITEL

SUBMITTED BY:

SANDEEP GAUTAM(073/BCT/35)

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

HIMALAYA COLLEGE OF ENGINEERING

CHYASAL, LALITPUR

JULY,2019

TITLE: TRAFFIC LIGHT CONTROLLER IN VHDL

OBJECTIVES:

- To design and implement traffic light controller using VHDL.
- To simulate traffic light controller using VHDL.

THEORY:

The normal function of traffic light requires more than slight control and coordination to ensure that traffic and pedestrians move as smoothly, and safely as possible. A variety of different control systems are used to accomplish this, ranging from simple clockwork mechanisms to sophisticated computerized control and coordination systems that self-adjust to minimize delay to people using the junction.

A simple traffic light controller can be implemented by a state machine that has a state diagram as shown in figure mentioned below. Its state progresses according to the value of timer used. If the value of timer reaches to specific value then, the state of system changes.

A set of four traffic lights has been used for a 4-way junction as shown in figure below:

The states are defined in terms of output. The timer goes from 0 to 100 and then reset to 0. Each of the light are programmed to operate as follow:

1. It is GREEN for 20 s.
2. Then it remains YELLOW for 5s.

During this time all the other lights remain RED.

This process runs successively for all 4 lights.

FINITE STATE MACHINE

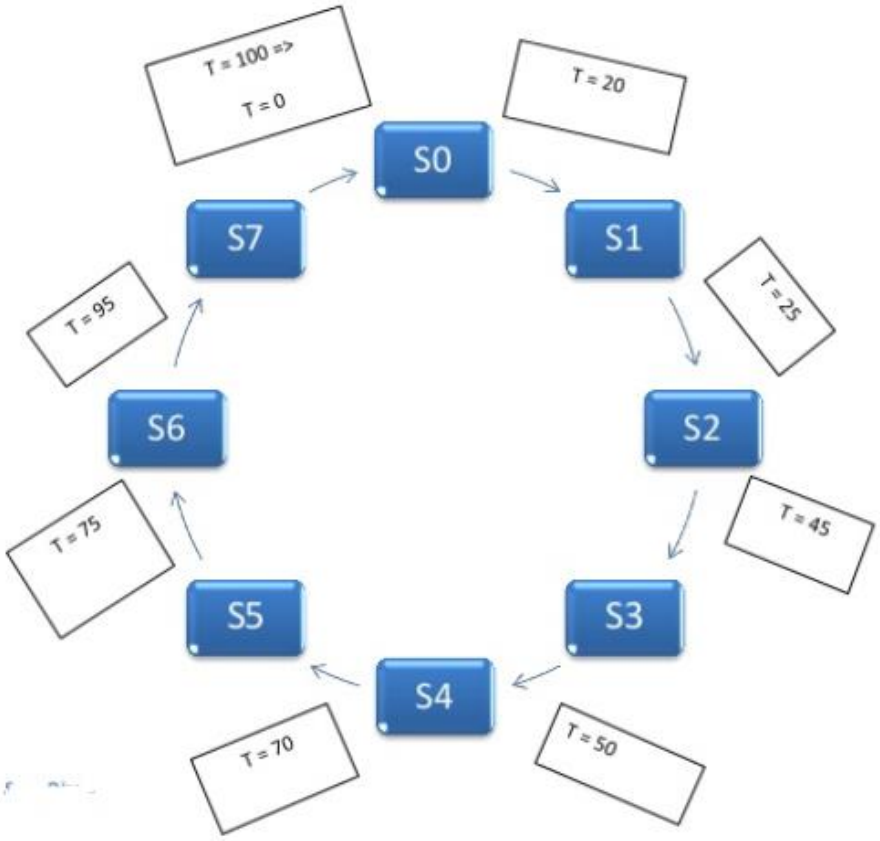

Fig: Finite State Machine

STATE TABLE

Time (Input)	Current State	Next State	Output
0	S0	S0	GRRR
20	S0	S1	GRRR
25	S1	S2	YRRR
45	S2	S3	RGRR
50	S3	S4	RYRR
70	S4	S5	RRGR
75	S5	S6	RRYR
95	S6	S7	RRRG
100	S7	S0	RRRY

The letter G, R, B represents the status of light as green, red and blue respectively in output. And also time will return to initial state 0 after 100.

VHDL CODE

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;
entity traffic is
    Port ( clk : in  STD_LOGIC;
           clr : in  STD_LOGIC;
           red1 : out  STD_LOGIC;
           yellow1 : out  STD_LOGIC;
           green1 : out  STD_LOGIC;
           red2 : out  STD_LOGIC;
           yellow2 : out  STD_LOGIC;
           green2 : out  STD_LOGIC;
           red3 : out  STD_LOGIC;
           yellow3 : out  STD_LOGIC;
           green3 : out  STD_LOGIC;
           red4 : out  STD_LOGIC;
           yellow4 : out  STD_LOGIC;
           green4 : out  STD_LOGIC);
end traffic;

architecture Behavioral of traffic is
type state_type is(s0,s1,s2,s3,s4,s5,s6,s7);
signal state: state_type :=s0;
signal count: integer:=0;
signal lights: std_logic_vector(11 downto 0);
begin
STATEpro:process(state)
begin
```

```
case state is
   when s0=>lights<="001100100100";
   when s1=>lights<="010100100100";
   when s2=>lights<="100001100100";
   when s3=>lights<="100010100100";
   when s4=>lights<="100100001100";
   when s5=>lights<="100100010100";
   when s6=>lights<="100100100001";
   when s7=>lights<="100100100010";
   when others=>lights<=lights;
end case;
end process;
LT:process(clk)
begin
 case count is
     when 0=> state <=s0; count <= count+1;
     when 20=> state <=s1; count <= count+1;
     when 25=> state <=s2; count <= count+1;
     when 45=> state <=s3; count <= count+1;
     when 50=> state <=s4; count <= count+1;
     when 70=> state <=s5; count <= count+1;
     when 75=> state <=s6; count <= count+1;
     when 95=> state <=s7; count <= count+1;
     when 100=>count<=0;
     when others => count <= count+1;
end case;
green4 <=lights(0);
yellow4 <=lights(1);
red4 <=lights(2);
```

```
green3 <=lights(3);
yellow3 <=lights(4);
red3 <=lights(5);
green2 <=lights(6);
yellow2 <=lights(7);
red2 <=lights(8);
green1 <=lights(9);
yellow1 <=lights(10);
red1 <=lights(11);
end process;
end Behavioral;
```

RTL SCHEMATICS

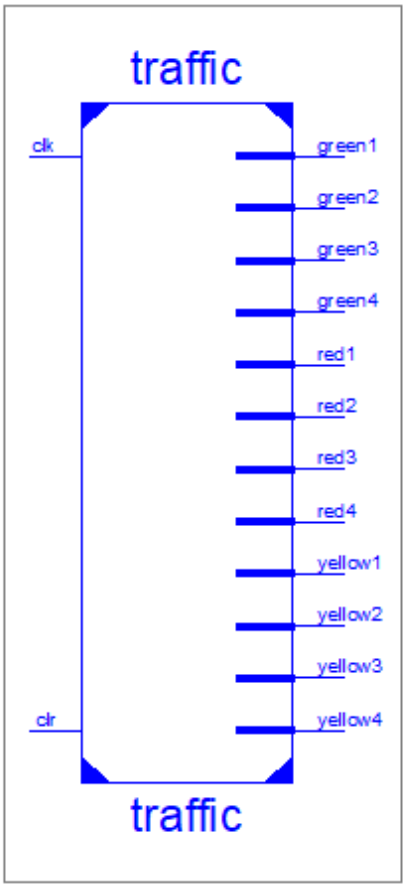

Fig: Black box view of traffic light controller

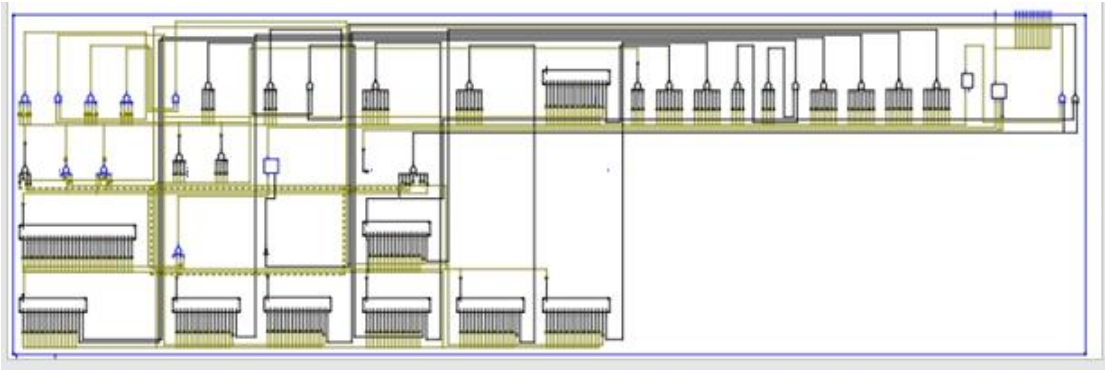

Fig: Internal connections of traffic light controller

VHDL TEST BENCH

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY trffica IS
END trffica;
ARCHITECTURE behavior OF trffica IS

   -- Component Declaration for the Unit Under Test (UUT)

   COMPONENT traffic
   PORT(
       clk : IN  std_logic;
       clr : IN  std_logic;
       red1 : OUT  std_logic;
       yellow1 : OUT  std_logic;
       green1 : OUT  std_logic;
       red2 : OUT  std_logic;
       yellow2 : OUT  std_logic;
       green2 : OUT  std_logic;
       red3 : OUT  std_logic;
       yellow3 : OUT  std_logic;
       green3 : OUT  std_logic;
       red4 : OUT  std_logic;
       yellow4 : OUT  std_logic;
       green4 : OUT  std_logic
      );
   END COMPONENT;
```

```
  --Inputs
  signal clk : std_logic := '0';
  signal clr : std_logic := '0';
 --Outputs
  signal red1 : std_logic;
  signal yellow1 : std_logic;
  signal green1 : std_logic;
  signal red2 : std_logic;
  signal yellow2 : std_logic;
  signal green2 : std_logic;
  signal red3 : std_logic;
  signal yellow3 : std_logic;
  signal green3 : std_logic;
  signal red4 : std_logic;
  signal yellow4 : std_logic;
  signal green4 : std_logic;
 -- Clock period definitions
  constant clk_period : time := 10 ns;
BEGIN
-- Instantiate the Unit Under Test (UUT)
  uut: traffic PORT MAP (
        clk => clk,
        clr => clr,
        red1 => red1,
        yellow1 => yellow1,
        green1 => green1,
        red2 => red2,
        yellow2 => yellow2,
        green2 => green2,
```

```
          red3 => red3,
          yellow3 => yellow3,
          green3 => green3,
          red4 => red4,
          yellow4 => yellow4,
          green4 => green4
        );
   -- Clock process definitions
 clk_process :process
   begin
clk <= '0';
wait for clk_period/2;
clk <= '1';
wait for clk_period/2;
   end process;
   -- Stimulus process
   stim_proc: process
   begin
      -- hold reset state for 100 ns.
      wait for 100 ns;
clk<='0';
clr<='0';
 wait for 100 ns;
clk<='0';
clr<='1';
     -- wait for clk_period*10;
      wait;
   end process;
END;
```

SIMULATION

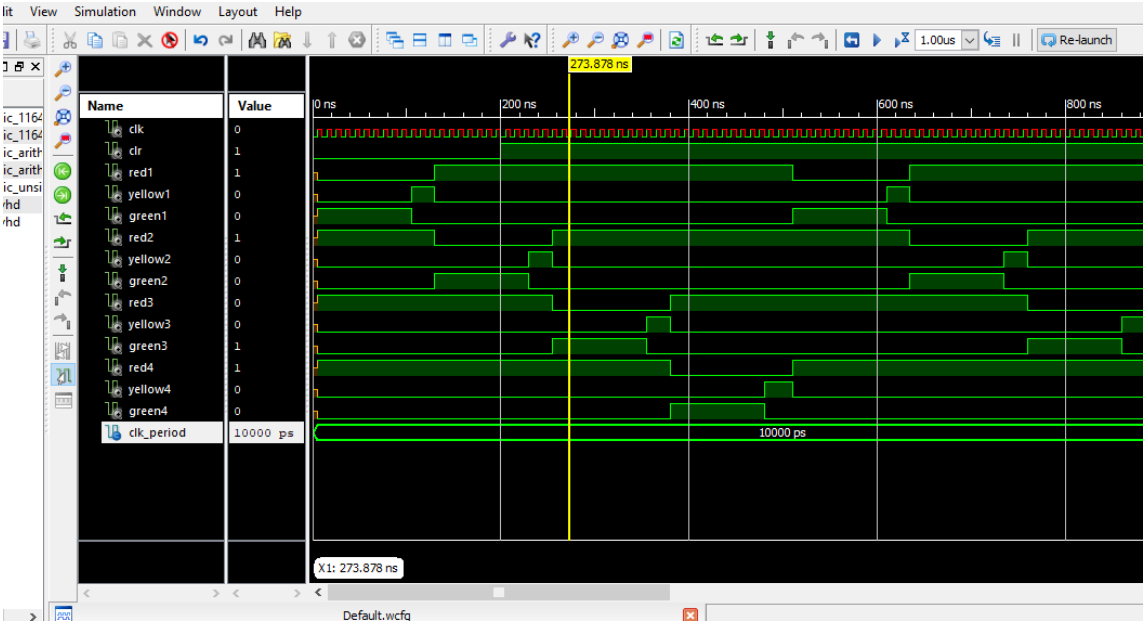

Fig: Simulation of traffic light controller

DISCUSSTION AND CONCLUSION

Through this project of VHDL we knew about the concept of XILINX ISE Design Suite. As we were given to do an individual VHDL project, I have selected the traffic light controller for a junction road consisting of four traffic light signal. For implementation of it, I have created VHDL TEST MODULE for generating RTL of traffic light controller and VHDL TEST BENCH for creating simulation of traffic light controller waveform.

At the end of my project we were able to view RTL and simulation of traffic light controller by doing code on VHDL.