SCHOOL OF ELECTRONICS ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)

VLSI LABORATORY REPORT

(EC-3095)

Submitted By

Name: Debagnik Kar

Roll No: 1804373

Section: ETC-06

Semester: 6th semester

Experiment: 5th

Design of traffic light controller using verilog and it's FPGA implementation.

Experiment -06

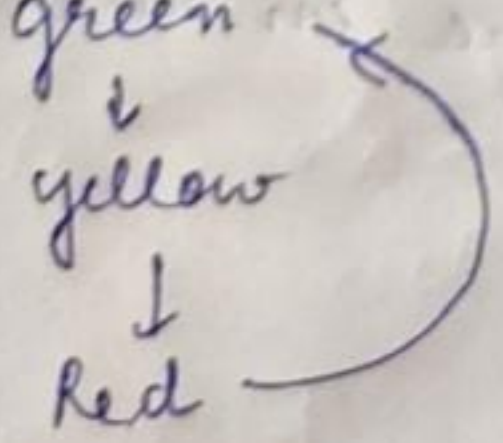
Aim: Design of traffic light controller using verilog and its FPGA implemention.

Software Used: EDA playground.

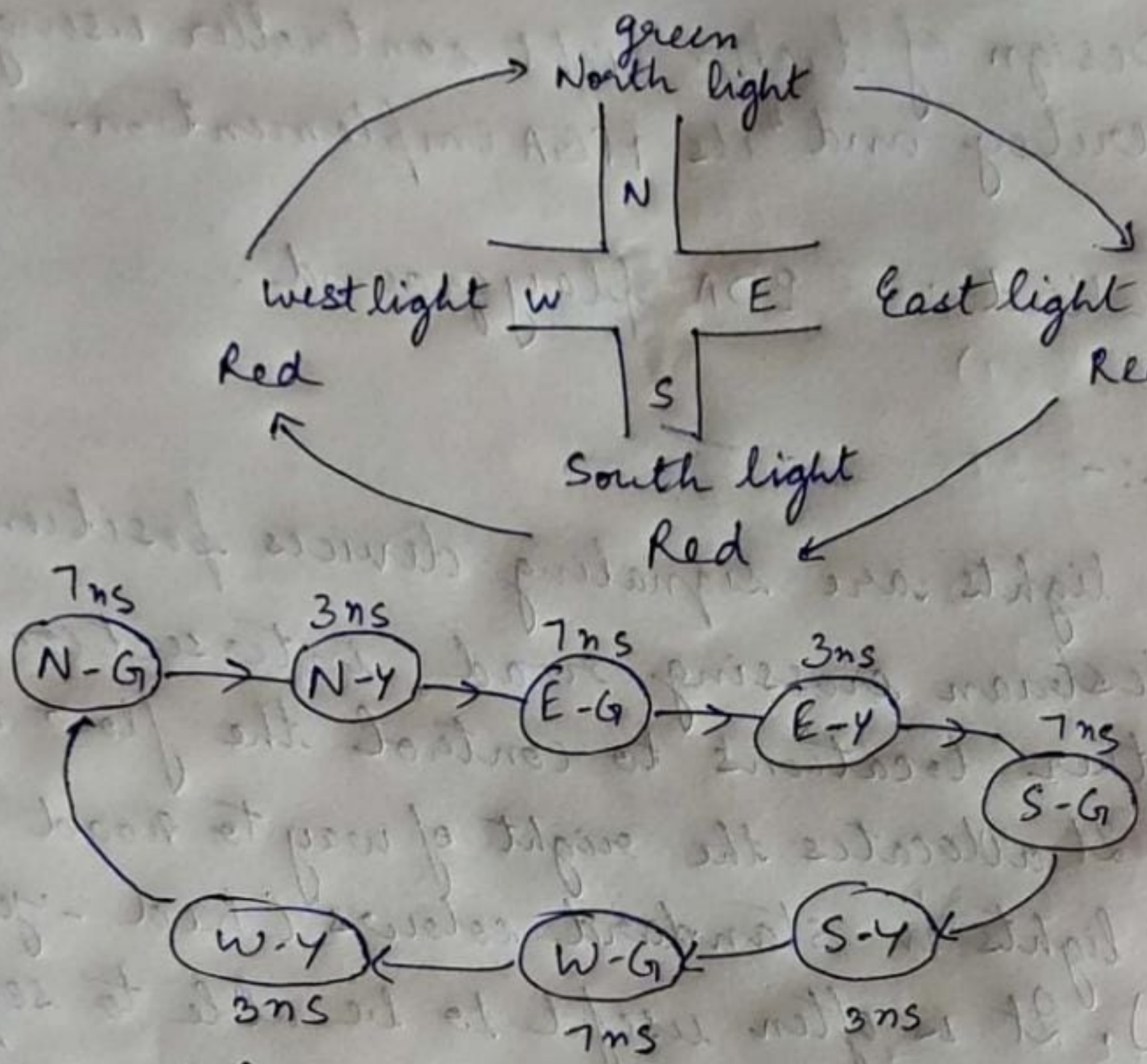
Theory:-

Traffic lights are signaling devices positioned at pedestrian crossing road, intersections, and other locations to control the flow of traffic. It allocates the right of way to road users using lights in standard colours (Red - yellow - green). It is often useful to be able to sequence through an arbitary number of states, staying in each through an arbitary ~~number~~ of amount of time.

- Four way Traffic Light



In this system If one way is green/yellow others are red.



There are 8 parameters, So [2:0] states Initially all 4 signals are kept red. Each signal will change the state after 10nsec. First the north becomes green for 7ns and yellow for next 3 ns. Simultaneously, all the other 3 signal will perform the above same in next 10 sec.

Debagnik Kar 1804373

Code:

Design.sv

```
module traffic_light(output [2:0] north,east,south,west, input clock,reset);
  reg [2:0] north,east,south,west;
  reg [2:0] state;
  reg [2:0] count;
  parameter n_g = 3'b000,n_y = 3'b001,e_g = 3'b010,e_y = 3'b011,s_g = 3'b100,s_y =
3'b101,w_g = 3'b110,w_y = 3'b111;
  parameter red = 3'b100, yellow = 3'b010, green = 3'b001;
  always @(posedge clock or posedge reset) begin
    if(reset) begin
      count=0;
      state=n_g;
    end
    else if(count==7) begin
      case(state)
        n_g: begin state=n_y; count=0; end
        e_g: begin state=e_y; count=0; end
        s_g: begin state=s_y; count=0; end
        w_g: begin state=w_y; count=0; end
        //default: begin state=n_y; count=0; end
      endcase
    end
    else if(count==3) begin
      case(state)
        n_y: begin state=e_g; count=0; end
        e_y: begin state=s_g; count=0; end
        s_y: begin state=w_g; count=0; end
        w_y: begin state=n_g; count=0; end
      endcase
    end
    count=count+1;
  end
  always@(state) begin
    case(state)
      n_g: begin north=green; east=red; south=red; west=red; end
      n_y: begin north=yellow; east=red; south=red; west=red; end
      e_g: begin north=red; east=green; south=red; west=red; end
      e_y: begin north=red; east=yellow; south=red; west=red; end
      s_g: begin north=red; east=red; south=green; west=red; end
      s_y: begin north=red; east=red; south=yellow; west=red; end
      w_g: begin north=red; east=red; south=red; west=green; end
      w_y: begin north=red; east=red; south=red; west=yellow; end
      default: begin north=green; east=red; south=red; west=red; end
    endcase
  end
endmodule
```

Testbench.sv

```
// Code your testbench here
// or browse Examples
module traffic_light_test;
  reg clock,reset;
  wire [2:0] north,east,south,west;
  traffic_light tt(north,east,south,west,clock,reset);
  always #10 clock=!clock;
  initial begin
    $dumpfile("dump.vcd"); $dumpvars;
    clock=0;reset=1;
    #10 reset=0;
  end
endmodule
```

Observation/outputs:

Fig 1: output

Conclusion

In this experiment we have successfully designed the 4 way traffic light controller on EDA playground by writing the code and also simulated it to obtain the required waveform and observed the output.