Traffic Light Control System

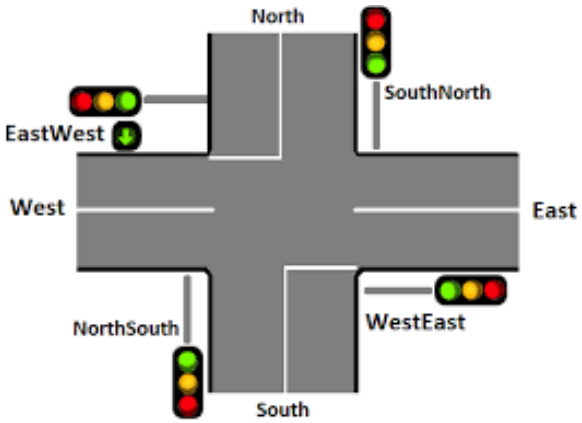
Using Verilog (VIVADO SOFTWARE)

Designed by Akash Sonowal

Duration of Project : May 2022 - June 2022.

Overview

Designed a Automated Traffic Light Control System for Highway Intersection using Viavdo Software . Generally It is Designed for controlling Traffic for four way intersection or for Junction of Two Roads. The Following Diagram Shows a Rough View of the Designed Traffic Control System.



STATE	WEST-EAST	NORTH-SOUTH	DELAY(SEC)
0	GREEN	RED	5
1	YELLOW	RED	1
2	RED	RED	1
3	RED	GREEN	5
4	RED	YELLOW	1
5	RED	RED	1

Specifications

1) Considered Lights as Binary Code :

   Green : 100

   Red : 001

   Yellow : 010

2) Consider two set of Traffic Lights :

   Light_A and Light_B

3) Divided the Possibility of Crossing and Traffic Signal in 6 possible combinations :

   S0 : Light_A= 001 ; Light_B=100.

   S1 : Light_A= 010 ; Light_B=100.

   S2 : Light_A= 100 ; Light_B=100.

   S3 : Light_A= 100 ; Light_B=001.

   S4 : Light_A= 100 ; Light_B=010.

   S5 : Light_A= 100 ; Light_B=100.

.

Verilog Code :

```
module traffic_lights(
    input clk,
    input rst_n,
    output reg[2:0] light_A,
    output reg[2:0] light_B

);
   reg [2:0] Green;
   reg [2:0] Red ;
   reg [2:0] Yellow;

localparam s0 = 6'b000001,
           s1 = 6'b000010,
           s2 = 6'b000100,
           s3 = 6'b001000,
           s4 = 6'b010000,
           s5 = 6'b100000;

  reg[7:0] curr_state;
  reg[7:0] next_state;

  reg [7:0] count;
  reg [7:0] count_en;
  reg count_done;

   localparam SEC5 = 8'd50,     SEC1 = 8'd10;


 always@(posedge clk or negedge rst_n) begin
```

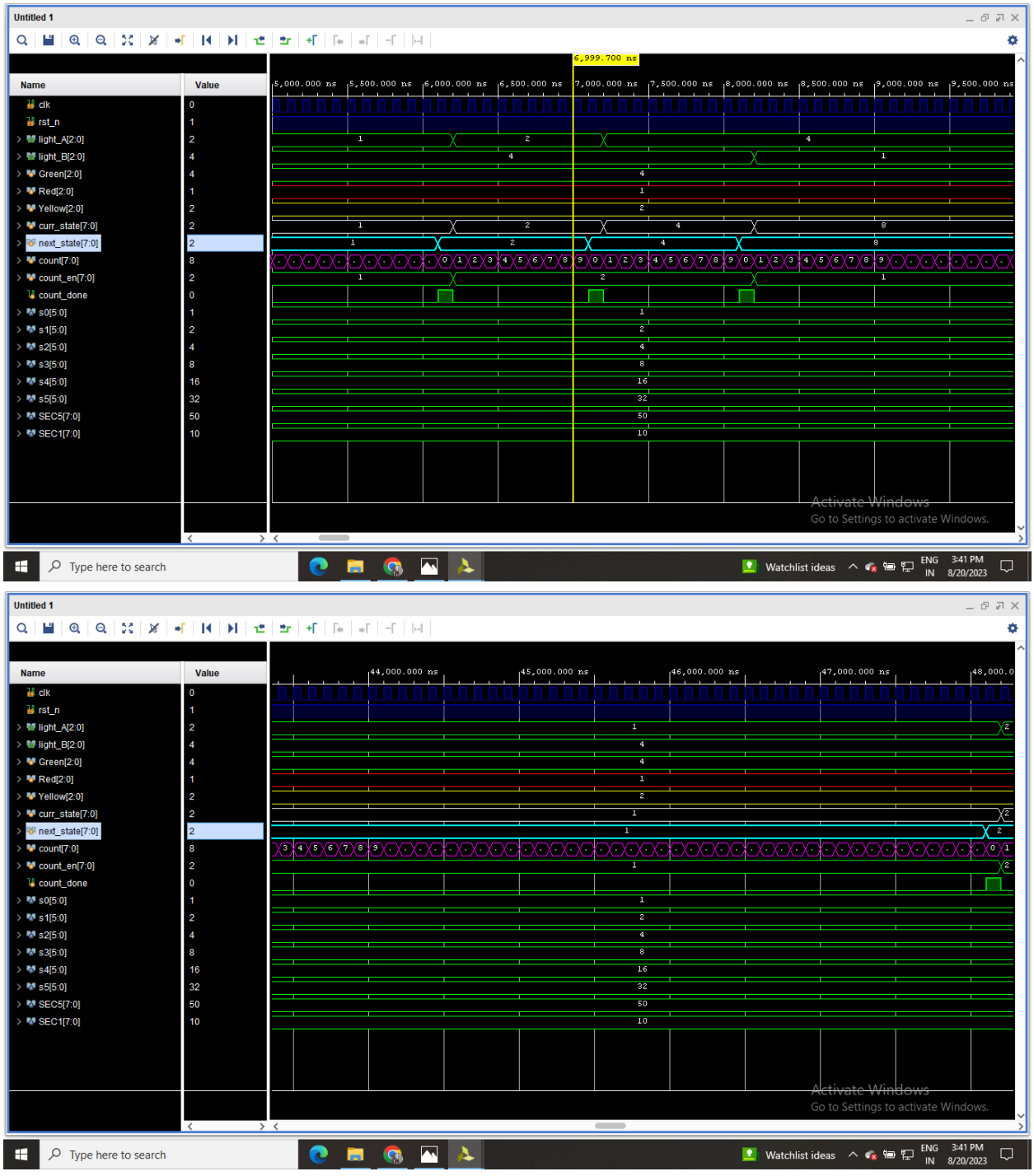
```
if(!rst_n) begin
count <= 8'b0;
count_done <= 8'b0;

end

else begin
    Green =8'b100;
     Red =8'b001;
      Yellow =8'b010;
case(count_en)
 2'b01:
   begin
    count_done <=0;
    if(count <SEC5 - 1)
       count <= count + 1'b1;
    else begin
     count <= 0;
    count_done <= 1;
     end
   end
 2'b10:
    begin
    count_done<=0;
      if(count <SEC1 - 1)
       count <=count + 1'b1;
      else begin
       count <= 0;
      count_done <= 1;
       end
     end
     default: begin
        count <= 0;
       count_done <=0;
```

```
        end
    endcase
  end
end

  always@(posedge clk or negedge rst_n) begin
        if (!rst_n) begin

          curr_state <= s0;
        end
        else begin
          curr_state <= next_state;
      end
    end

 always @(*)
         case(curr_state)
         s0: begin
          count_en = 2'b01;
          if(count_done)
           next_state = s1;
           end
         s1: begin
            count_en = 2'b10;
            if(count_done)
              next_state = s2;
            end
         s2: begin
           count_en=2'b10;
           if(count_done)
            next_state = s3;
            end
         s3: begin
            count_en=2'b01;
```

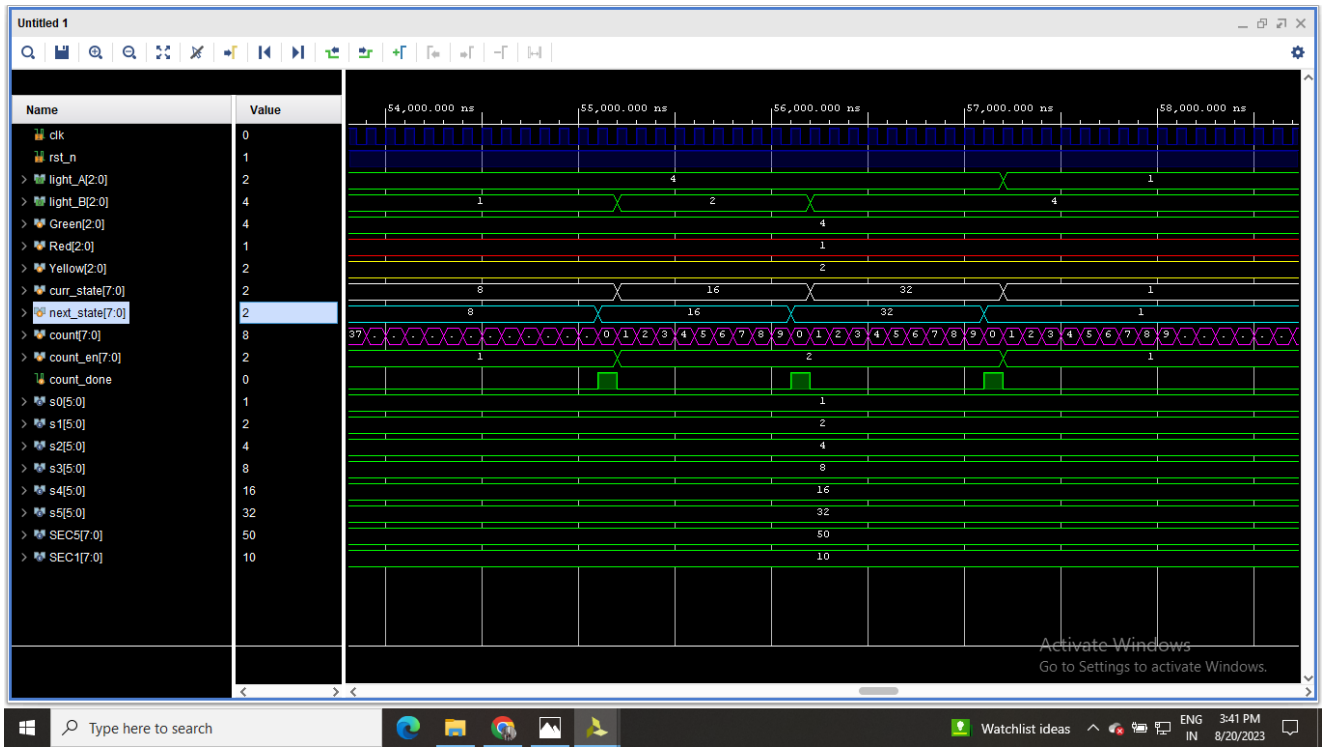
```
    if(count_done)
     next_state = s4;
     end
 s4: begin
    count_en=2'b10;
    if(count_done)
    next_state = s5;
   end
s5: begin
   count_en=2'b10;
   if(count_done)
   next_state = s0;
  end
default: begin
       next_state = s0;
       count_en =2'b00;
       end
endcase

always@(posedge clk or negedge rst_n) begin
if (!rst_n) begin
 light_A = 3'b000;
 light_B = 3'b000;
  end
  else begin
    case (next_state)
       s0:begin
       light_A<=3'b001;
       light_B<=3'b100;
       end
       s1:begin
        light_A<=3'b010;
         light_B<=3'b100;
         end
```

```
                s2:begin
                light_A<=3'b100;
                 light_B<=3'b100;
                 end
                s3:begin
               light_A<=3'b100;
               light_B<=3'b001;
               end
                s4:begin
                 light_A<=3'b100;
                 light_B <=3'b010;
                 end
                s5:begin
                light_A <= 3'b100;
                light_B <= 3'b100;
                end
            default: begin
                  light_A = 3'b000;
                  light_B = 3'b000;
                end
          endcase
       end
     end
endmodule
```

OUTPUT :


Untitled 1
Name
Value
clk
rst_n
light_A[2:0]
light_B[2:0]
Green[2:0]
Red[2:0]
Yellow[2:0]
curr_state[7:0]
next_state[7:0]
count[7:0]
count_en[7:0]
count_done
s0[5:0]
s1[5:0]
s2[5:0]
s3[5:0]
s4[5:0]
s5[5:0]
SEC5[7:0]
SEC1[7:0]
6,999.700 ns
Activate Windows
Go to Settings to activate Windows.
Type here to search
Watchlist ideas
ENG
IN
3:41 PM
8/20/2023

OUTPUT :


Untitled 1
Name
Value
clk
rst_n
light_A[2:0]
light_B[2:0]
Green[2:0]
Red[2:0]
Yellow[2:0]
curr_state[7:0]
next_state[7:0]
count[7:0]
count_en[7:0]
count_done
s0[5:0]
s1[5:0]
s2[5:0]
s3[5:0]
s4[5:0]
s5[5:0]
SEC5[7:0]
SEC1[7:0]
54,000.000 ns
55,000.000 ns
56,000.000 ns
57,000.000 ns
58,000.000 ns
Activate Windows
Go to Settings to activate Windows.
Type here to search
Watchlist ideas
ENG
IN
3:41 PM
8/20/2023