DIGITAL DESIGN PROJECT

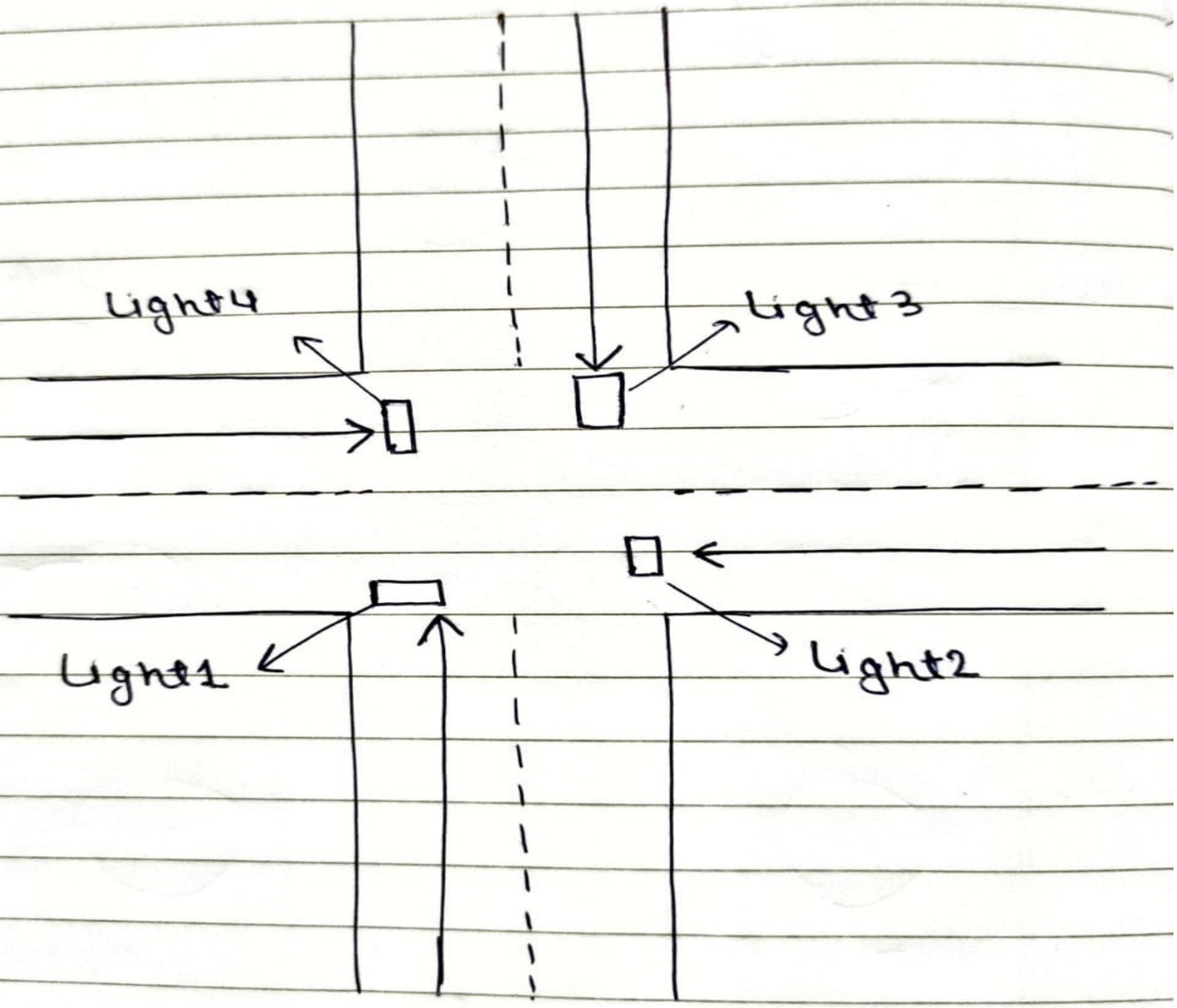
- NAME – SHASHI PRAKASH
- ROLL NO- B19EE076

PROJECT TOPIC

- 4-way Traffic Light Controller with Red Green and Yellow lights

TRAFFIC LIGHT WORKING

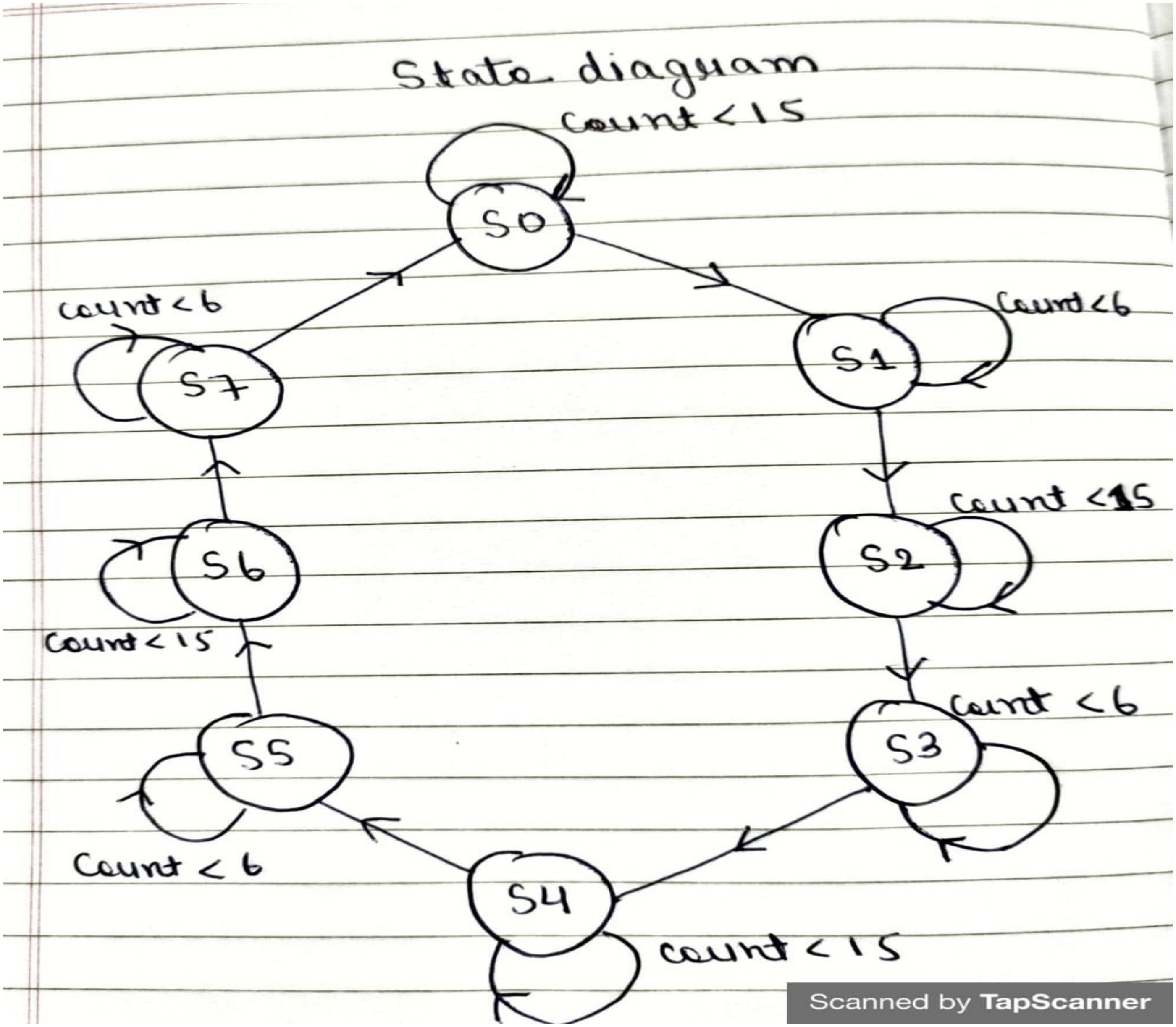
- There are three lights such as green,yellow,red.
- Green signal allows movement of vehicles.
- Yellow signal represents that signal is going to change from green to red.
- Red signal stop the movement of vehicles.


Light4
Light3
Light1
Light2

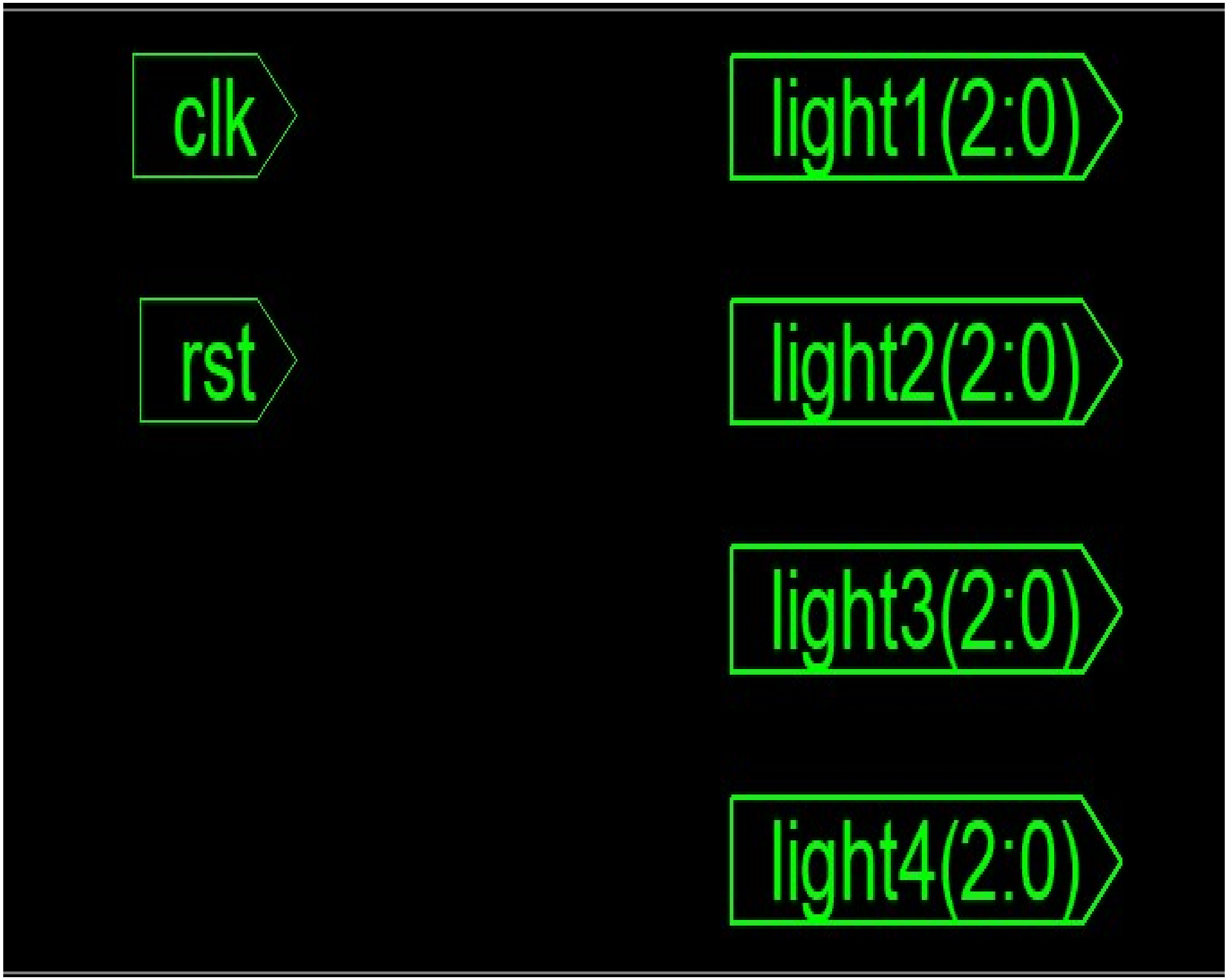
STATE DESCRIPTION

STATE	LIGHT1	LIGHT2	LIGHT3	LIGHT4
S0	GREEN	RED	RED	RED
S1	YELLOW	RED	RED	RED
S2	RED	GREEN	RED	RED
S3	RED	YELLOW	RED	RED
S4	RED	RED	GREEN	RED
S5	RED	RED	YELLOW	RED
S6	RED	RED	RED	GREEN
S7	RED	RED	RED	YELLOW

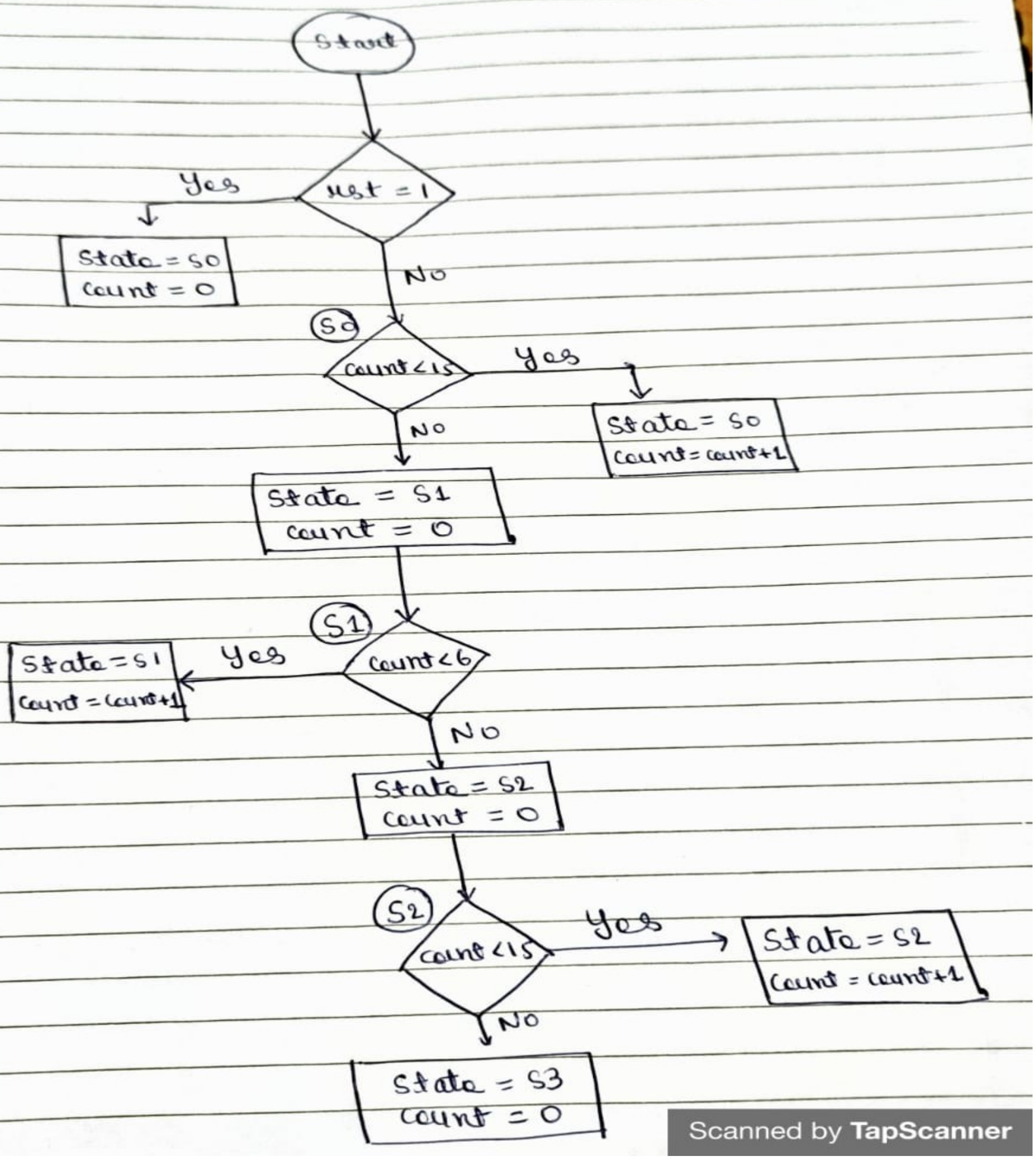
STATE DIAGRAM



- We have three light signal.
- Let green signal = 001 , yellow signal = 010, and red signal = 100.
- We have two input clock and reset.
- We have four output Light1, Light2, Light3 and Light4.


clk
rst
light1(2:0)
light2(2:0)
light3(2:0)
light4(2:0)

Flowchart


Start
rst = 1
Yes
State = S0
count = 0
No
S0
count<15
Yes
State = S0
count=count+1
No
State = S1
count = 0
S1
count<6
Yes
State=S1
count=count+1
No
State = S2
count = 0
S2
count<15
Yes
State = S2
Count = count+1
No
State = S3
count = 0

- The above figure denotes the flowchart for states S0, S1 and S2.
- Similarly we can draw flowchart for states S3, S4, S5, S6 and S7.
- At the end again the flowchart start from S0 states and counter start from 0.

VERILOG CODE

```
module waytrafficlight(
   clk,
           rst,
           light1,
           light2,
           light3,
           light4,
           count,
           state
   );
input clk, rst;
output reg[2:0] light1,light2,light3,light4;
output reg[3:0] count;
output reg[2:0] state;
parameter S0=0, S1=1, S2=2, S3=3, S4=4,
S5=5, S6=6, S7=7;
```

```
always@(posedge(clk) or posedge(rst))
begin
  if(rst == 1)
     begin
     state<=S0;
     count<=0;
     end
```

IMPORTANT BLOCK OF CODE

```
S0:
        begin
        if(count<15)
           begin
           state<=S0;
           count<=count + 1;
           end
        else
           begin
           state<=S1;
           count<=0;
           end
        end
```

```
S1:
        begin
        if(count<6)
           begin
           state<=S1;
           count<=count + 1;
           end
        else
           begin
           state<=S2;
           count<=0;
           end
        end
```

IMPORTANT BLOCK OF CODE

```
always@(state)
  begin
        case(state)
                S0:
                 begin
                 light1 = 3'b001; light2 = 3'b100; light3 = 3'b100; light4 = 3'b100;
                 end
                S1:
                 begin
                 light1 = 3'b010; light2 = 3'b100; light3 = 3'b100; light4 = 3'b100;
                 end
                S2:
                 begin
                 light1 = 3'b100; light2 = 3'b001; light3 = 3'b100; light4 = 3'b100;
                 end
                S3:
                 begin
                 light1 = 3'b100; light2 = 3'b010; light3 = 3'b100; light4 = 3'b100;
                 end
```

IMPORTANT BLOCK OF CODE

```
S4:
                       begin
                       light1 = 3'b100; light2 = 3'b100; light3 = 3'b001; light4 = 3'b100;
                       end
   S5:
                       begin
                       light1 = 3'b100; light2 = 3'b100; light3 = 3'b010; light4 = 3'b100;
                       end
   S6:
                       begin
                       light1 = 3'b100; light2 = 3'b100; light3 = 3'b100; light4 = 3'b001;
                       end
   S7:
                       begin
                       light1 = 3'b100; light2 = 3'b100; light3 = 3'b100; light4 = 3'b010;
                       end
   endcase
end

endmodule
```

TESTBENCH CODE

```
module testbench;
        reg clk;
        reg rst;
        wire [2:0] light1;
        wire [2:0] light2;
        wire [2:0] light3;
        wire [2:0] light4;
        wire [3:0] count;
        wire [2:0] state;
        waytrafficlight uut (
                .clk(clk),
                .rst(rst),
                .light1(light1),
                .light2(light2),
                .light3(light3),
                .light4(light4),
                .count(count),
                .state(state)
        );
```

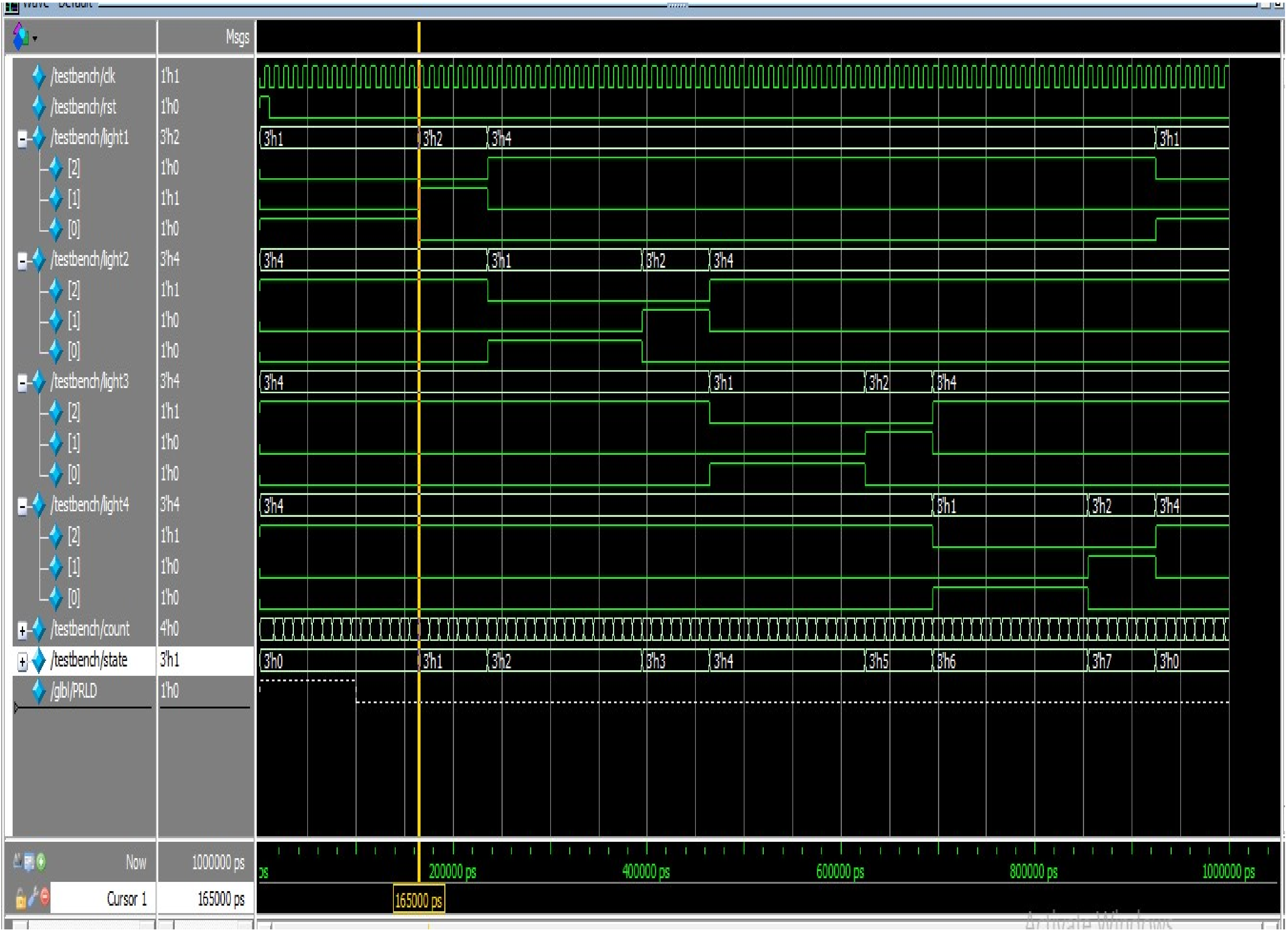
```
initial begin
                     clk = 0;
                     rst = 1;

        #10 rst=0;

        end
 always #5 clk = ~clk;
 initial #1000 $finish;

endmodule
```

SIMULATED WAVEFORM



	Msgs
/testbench/clk	1'h1
/testbench/rst	1'h0
/testbench/light1	3'h2
[2]	1'h0
[1]	1'h1
[0]	1'h0
/testbench/light2	3'h4
[2]	1'h1
[1]	1'h0
[0]	1'h0
/testbench/light3	3'h4
[2]	1'h1
[1]	1'h0
[0]	1'h0
/testbench/light4	3'h4
[2]	1'h1
[1]	1'h0
[0]	1'h0
/testbench/count	4'h0
/testbench/state	3'h1
/glbl/PRLD	1'h0
Now	1000000 ps
Cursor 1	165000 ps

CONCLUSION

- As we can see in simulated waveform when counter start from 0 to 14 then the state is in S0(green signal) state after that again counter start from 0 to 5 then the state is in S1(yellow signal) state and so on.
- When rst = 1 then it initializes the state = S0.

THANK YOU