GITAM
(DEEMED TO BE UNIVERSITY)
(Estd. u/s 3 of the UGC Act, 1956)
VISAKHAPATNAM ✹ HYDERABAD ✹ BENGALURU

VLSI DESIGN
(course code: EECE3051)

Case study

Title: simulation and implementation Traffic light controller using Xilinx vivado software with FPGA kit

Done by
R INDUMATHI - BU21EECE0100360
SNEHA SALIMATH- BU21EECE0100563

Under the guidance of
DR. Arun kumar Manoharan

-

We have considered a T - shaped road and heavy load of traffic is present over this road where we have to control it using traffic signal.

Six cases have been considered here and analyzed using state diagram and state table.

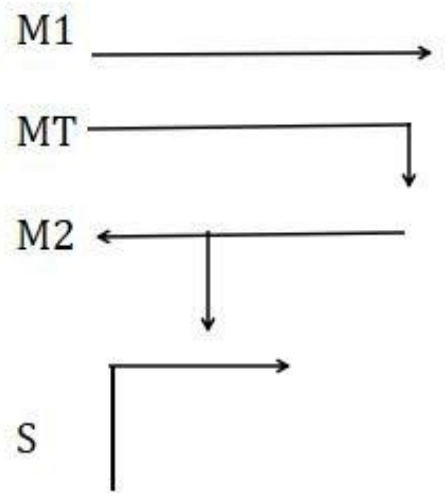


Figure 3.1 : Directions chart

The directions, M1, MT, M2, S, that is been considered for analysis of our problem is shown in the figure 3.1. And, the problem statement is explained in the figure 3.2.

Six states, S1, S2, S3, S4, S5, S6 are taken into consideration and state diagram(Figure 3.3), state table(Figure 3.4) is made using the following logic explained in the figure 3.2.

- Green light indicates that there is no traffic and there is easy flow of vehicles in that route/direction.
- Red light indicates that there is a traffic jam and that route is blocked for the vehicles tomove and,
- Yellow light indicates that the route has medium flow of vehicles.

Time delays for changing from one state to another(shown in Figure 3.2) is considered as, TMG(from S1 to S2), TY(from S2 to S3), TTG(from S3 to S4), TY(from S4 to S5), TSG(from S5 to S6) and TY(from S6 to S1) and the cycle continues.

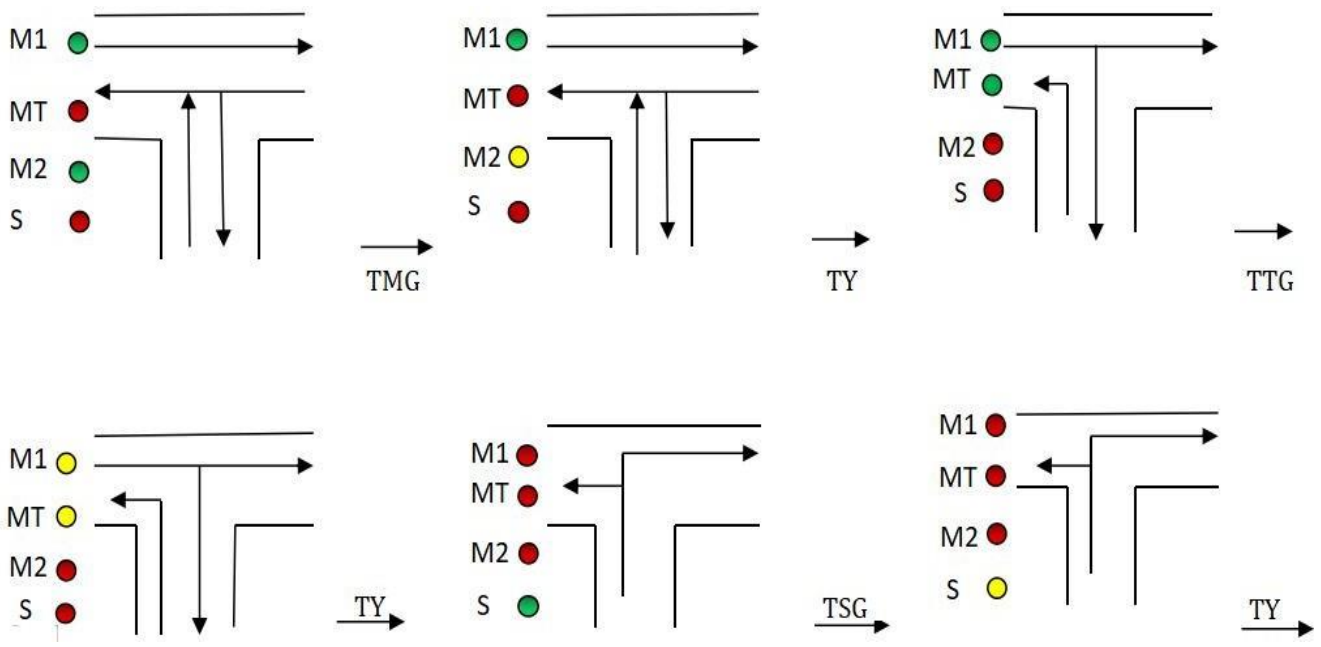


Figure 3.2 : Problem statement (Logic)

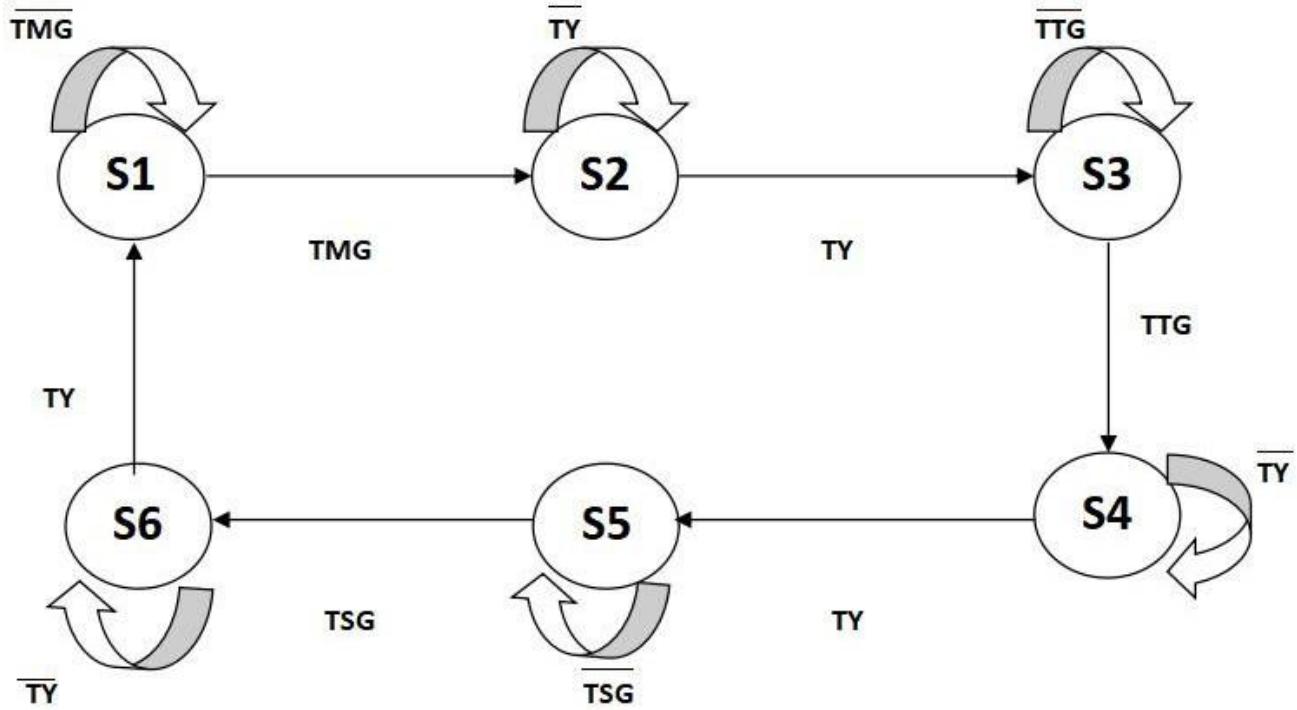


Figure 3.3 : State Diagram

In Figure 3.3, The time delays are considered as follows :

- TMG = 7 seconds
- TY = 2 seconds
- TTG = 5 seconds
- TSG = 3 seconds

Until TMG seconds, the signal will remain in S1 state, and after TMG seconds, it will move to S2 state. Until TY seconds it will remain in S2 state and after TY seconds, it will move to S3 state, and so on. After TY seconds, in state S6, it will go back to S1 state and the cycle continues.

The state table for the problem statement is shown below in Figure 3.4.

PRESENT STATE ABC	INPUT	NEXT STATE A*B*C*	S T	M1 RYG	M2 RYG	MT RYG	S RYG
000	-	001	1	000	000	000	000
001	$\overline{TMG}$ TMG	001 010	0 1	001	001	100	100
010	$\overline{TY}$ TY	010 011	0 1	001	010	100	100
011	$\overline{TTG}$ TTG	011 100	0 1	001	100	001	100
100	$\overline{TY}$ TY	100 101	0 1	010	100	010	100
101	$\overline{TSG}$ TSG	101 110	0 1	100	100	100	001
110	$\overline{TY}$ TY	110 001	0 1	100	100	100	010
111	-	000	0	000	000	000	000

Figure 3.4 : State Table

In Figure 3.4,

- R = RED,
- Y = YELLOW and,
- G = GREEN.

ST = State Transition; A, B and C are considered as the present state.
The state table is made, considering the Logic diagram/problem statement given in Figure 3.2.
From the Figure 3.2, Figure 3.3 and Figure 3.4,

In state S1(001); M1 = GREEN, implies, RYG value = 001,
MT = RED, implies, RYG value = 100,
M2 = GREEN, implies, RYG value = 001 and,
S = RED, implies, RYG value = 100.

After TMG seconds,

In state S2(010); M1 = GREEN, implies, RYG value = 001,
MT = RED, implies, RYG value = 100,
M2 = YELLOW, implies, RYG value = 010 and,
S = RED, implies, RYG value = 100.

After TY seconds,

In state S3(011); M1 = GREEN, implies, RYG value = 001,
MT = GREEN, implies, RYG value = 001,
M2 = RED, implies, RYG value = 100 and,
S = RED, implies, RYG value = 100.

After TTG seconds,

In state S4(100); M1 = YELLOW, implies, RYG value = 010,
MT = YELLOW, implies, RYG value = 010,
M2 = RED, implies, RYG value = 100 and,
S = RED, implies, RYG value = 100.

After TY seconds,

In state S5(101); M1 = RED, implies, RYG value = 100,
MT = RED, implies, RYG value = 100,
M2 = RED, implies, RYG value = 100 and,
S = GREEN, implies, RYG value = 001.

After TSG seconds,

In state S6(110); M1 = RED, implies, RYG value = 100,
MT = RED, implies, RYG value = 100,
M2 = RED, implies, RYG value = 100 and,
S = YELLOW, implies, RYG value = 010.

And after S6 state, the cycle repeats and goes to S1 state.

Program

```
module Traffic_Light_Controller(
      input clk,rst,
      output reg [2:0]light_M1,
      output reg [2:0]light_S,
      output reg [2:0]light_MT,
      output reg [2:0]light_M2
       );

      parameter    S1=0, S2=1, S3 =2, S4=3, S5=4,S6=5;
      reg [3:0]count;
      reg[2:0] ps;
      parameter    sec7=7,sec5=5,sec2=2,sec3=3;

      always@(posedge clk or posedge rst)
            begin
            if(rst==1)
            begin
            ps<=S1;
            count<=0;
            end
            else
                  case(ps)
                        S1: if(count<sec7)
                                    begin
                                    ps<=S1;
                                    count<=count+1;
                                    end
                              else
                                    begin
                                    ps<=S2;
                                    count<=0;
                                    end
                        S2: if(count<sec2)
                                    begin
                                    ps<=S2;
                                    count<=count+1;
                                    end
                              else
                                    begin
                                    ps<=S3;
                                    count<=0;
                                    end
```

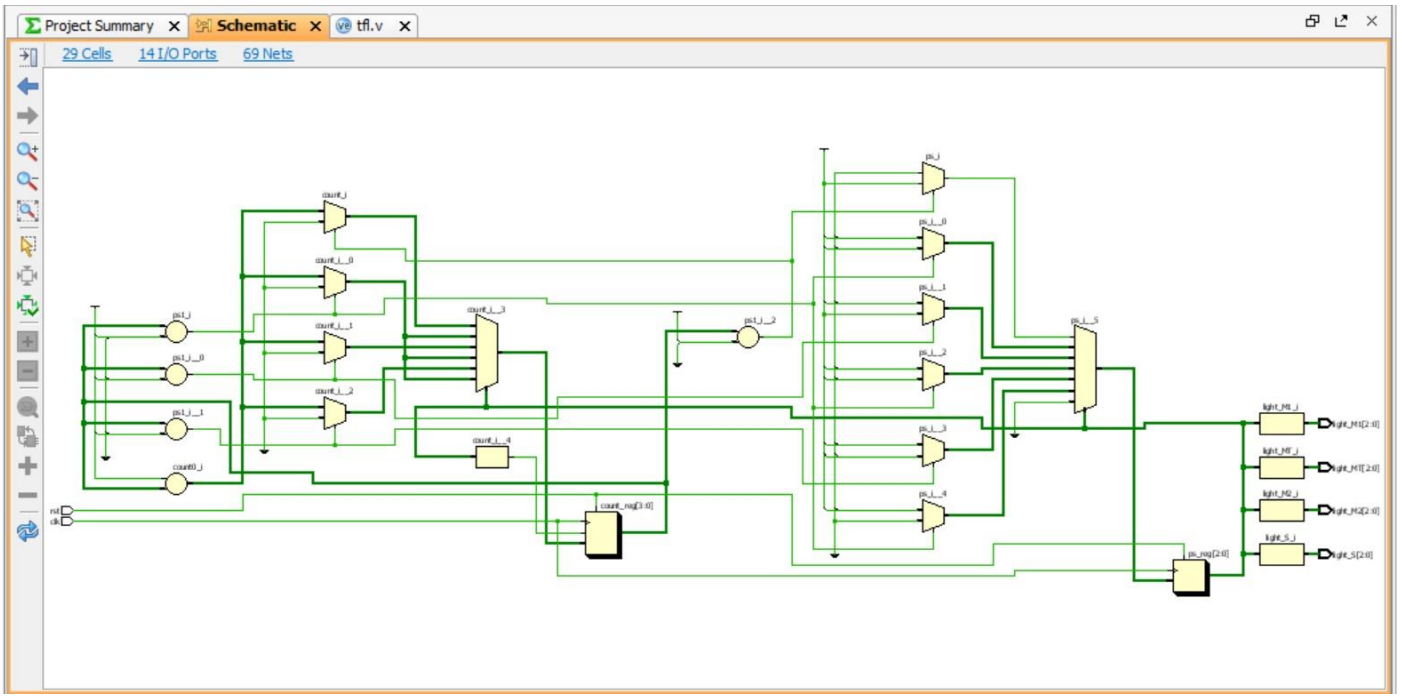
```
        S3: if(count<sec5)
                begin
                ps<=S3;
                count<=count+1;
                end
            else
                begin
                ps<=S4;
                count<=0;
                end
        S4:if(count<sec2)
                begin
                ps<=S4;
                count<=count+1;
                end
            else
                begin
                ps<=S5;
                count<=0;
                end
        S5:if(count<sec3)
                begin
                ps<=S5;
                count<=count+1;
                end
            else
                begin
                ps<=S6;
                count<=0;
                end
        S6:if(count<sec2)
                begin
                ps<=S6;
                count<=count+1;
                end
            else
                begin
                ps<=S1;
                count<=0;
                end
        default: ps<=S1;
        endcase
end


always@(ps)
begin
```

```
case(ps)
        S1:
        begin
            light_M1<=3'b001;
            light_M2<=3'b001;
            light_MT<=3'b100;
            light_S<=3'b100;
        end
        S2:
        begin
            light_M1<=3'b001;
            light_M2<=3'b010;
            light_MT<=3'b100;
            light_S<=3'b100;
        end
        S3:
        begin
            light_M1<=3'b001;
            light_M2<=3'b100;
            light_MT<=3'b001;
            light_S<=3'b100;
        end
        S4:
        begin
            light_M1<=3'b010;
            light_M2<=3'b100;
            light_MT<=3'b010;
            light_S<=3'b100;
        end
        S5:
        begin
            light_M1<=3'b100;
            light_M2<=3'b100;
            light_MT<=3'b100;
            light_S<=3'b001;
        end
        S6:
        begin
            light_M1<=3'b100;
            light_M2<=3'b100;
            light_MT<=3'b100;
            light_S<=3'b010;
        end
```

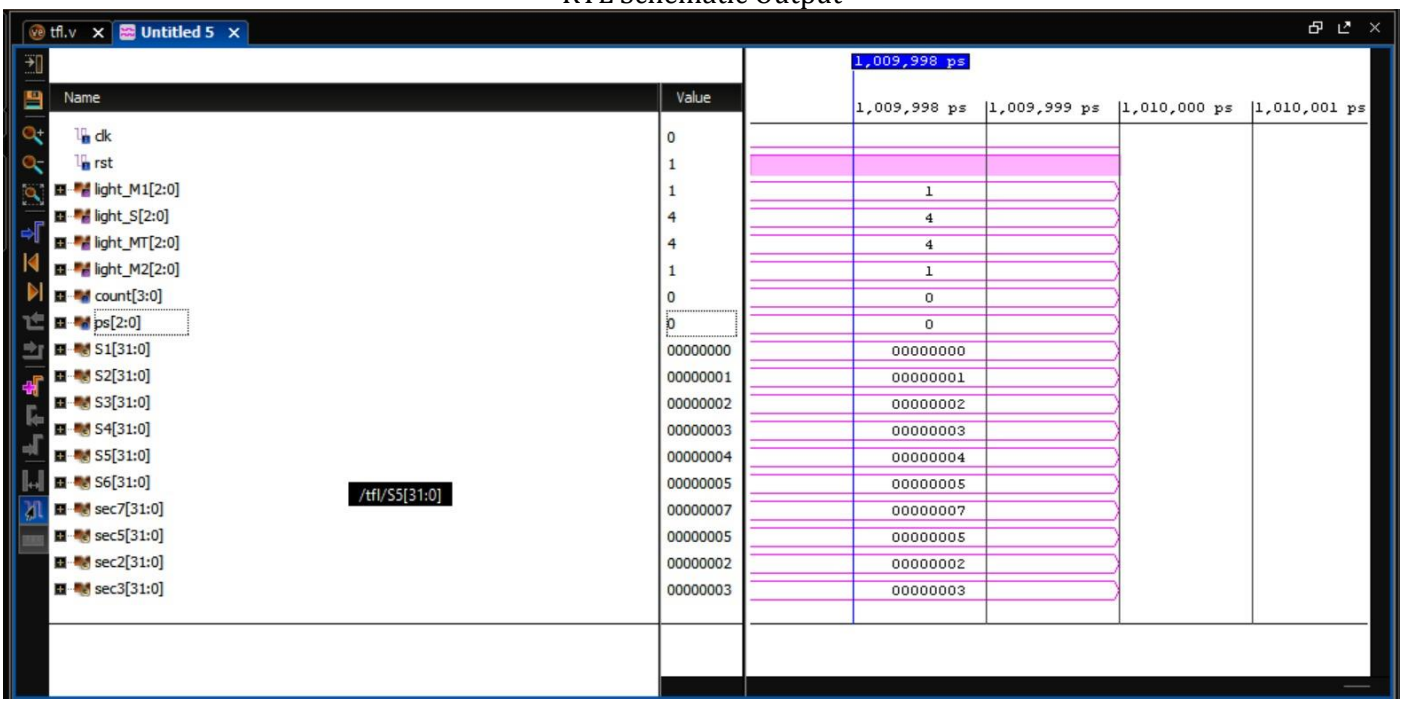
```
                default:
                        begin
                            light_M1<=3'b000;
                            light_M2<=3'b000;
                            light_MT<=3'b000;
                            light_S<=3'b000;
                        end
                    endcase
            end
    endmodule
```

VIDEO LINK: https://drive.google.com/file/d/1K8qDdTygBRO40o2ZNm_y8rn2tRUgK15F/view

RTL Schematic diagram



RTL Schematic Output



RTL Schematic Output