Verilog code

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'b100;

end

default:

begin

light\_M1<=3'b000;

light\_M2<=3'b000;

light\_MT<=3'b000;

light\_S<=3'b010;

end

endcase

end

endmodule