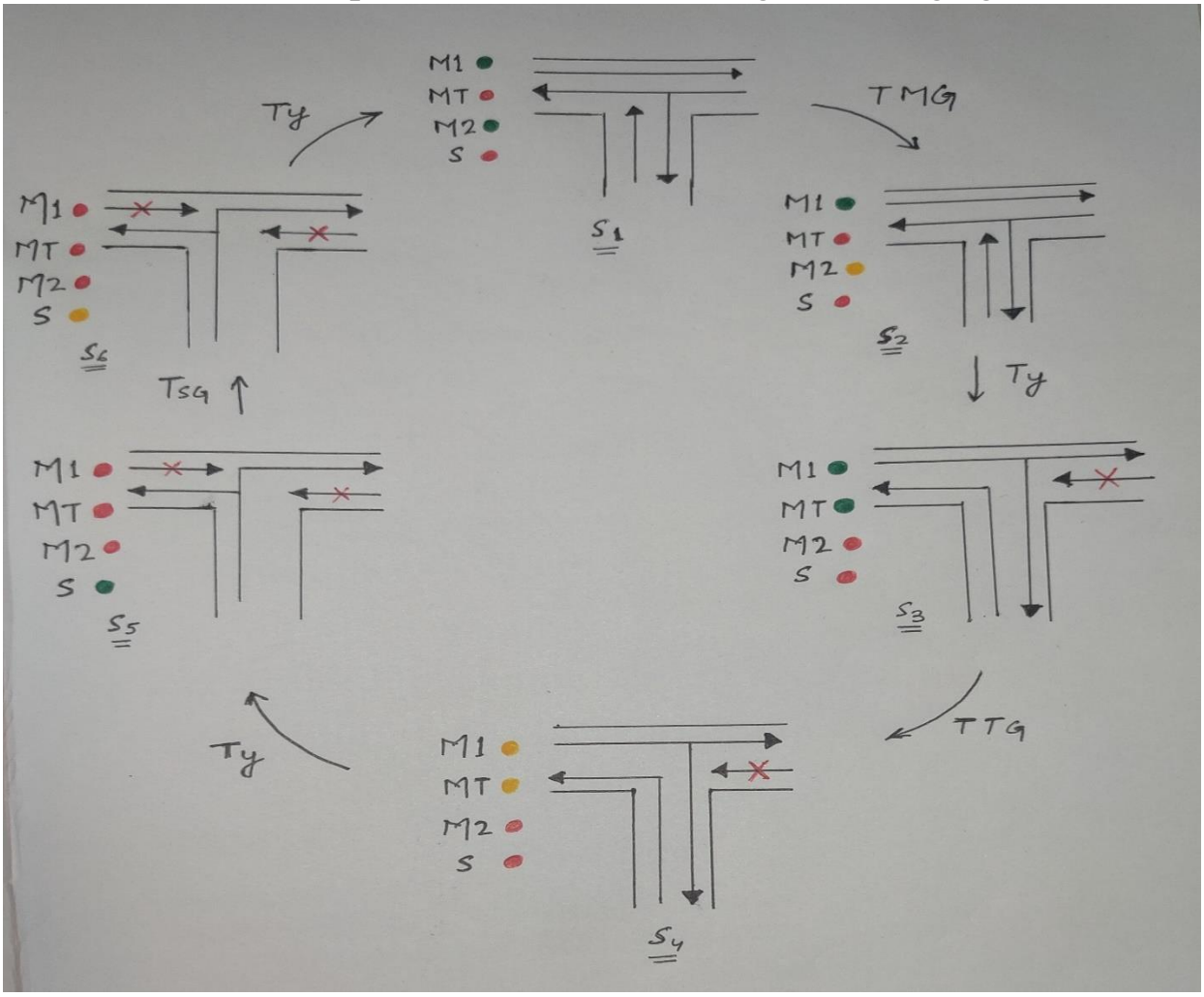
TRAFFIC LIGHT CONTROLLER USING VERILOG

National Institute of Technology, Rourkela
Pre- final year, Electrical Engineering

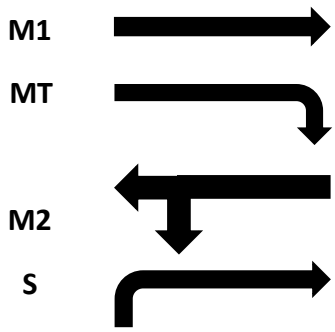
-by MEHUL KUMAR

Problem Statement: The aim of the project is to design a traffic controller for a T-intersection.

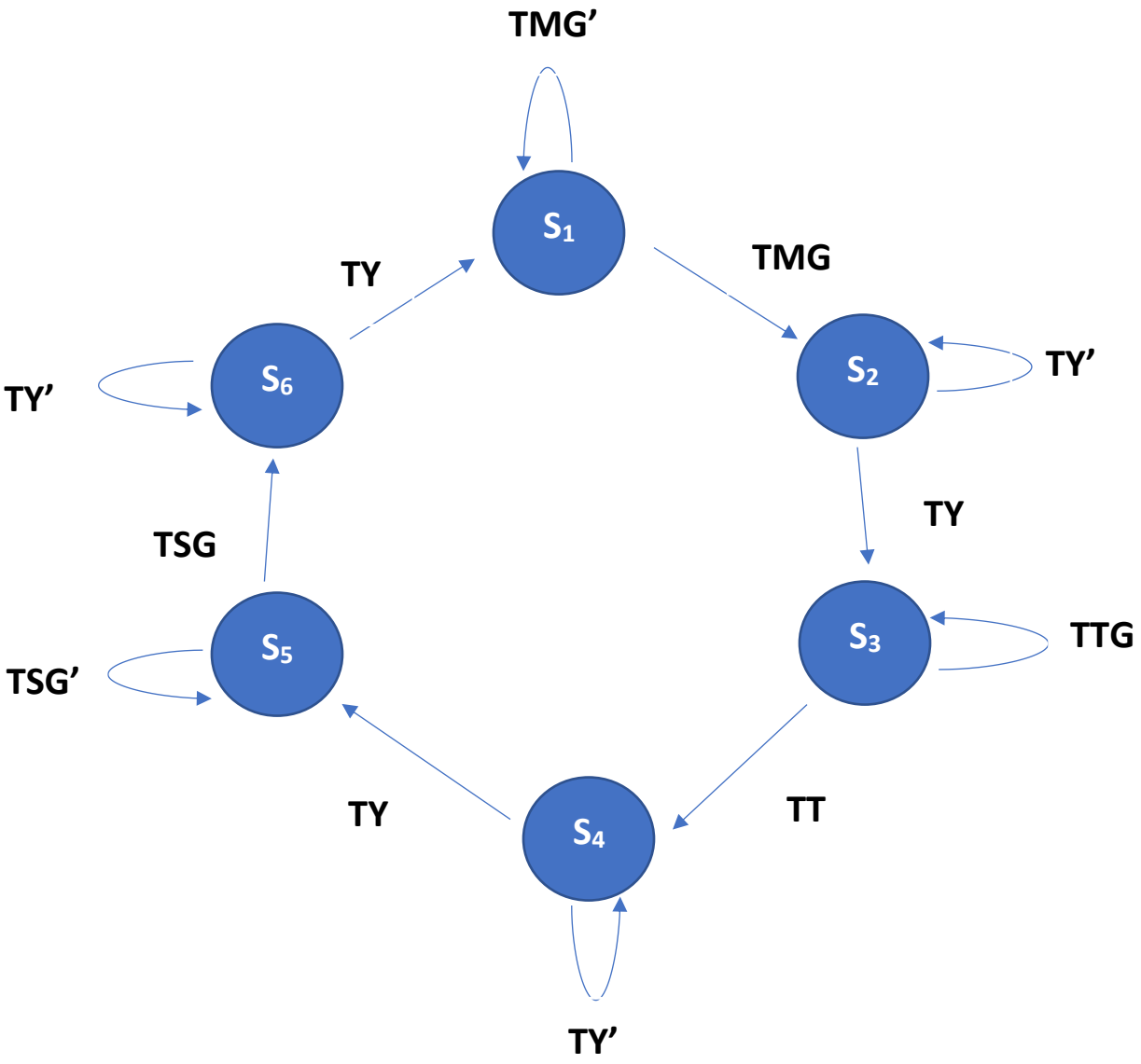
Let's understand the problem statement through the image given below.



M1, MT, M2 and S are the signals for the different directions show below,



State Diagram:



TMG= 7 secs. //Preference is given to the main road traffic

TY= 2 secs.

TTG= 5 secs.

TSG=3 secs.

NOTE:

TMG', TTG' TY' and TSG' are the time negations when the respective times are not reached.

State Table:

Present State A B C	Input	Next State $A^+ B^+ C^+$	M1 R Y G	M2 R Y G	MT R Y G	S R Y G
0 0 1	TMG' TMG	0 0 1 0 1 0	0 0 1	0 0 1	1 0 0	1 0 0
0 1 0	TY' TY	0 1 0 0 1 1	0 0 1	0 1 0	1 0 0	1 0 0
0 1 1	TTG' TTG	0 1 1 1 0 0	0 0 1	1 0 0	0 0 1	1 0 0
1 0 0	TY' TY	1 0 0 1 0 1	0 1 0	1 0 0	0 1 0	1 0 0
1 0 1	TSG' TSG	1 0 1 1 1 0	1 0 0	1 0 0	1 0 0	0 0 1
1 1 0	TY' TY	1 1 0 0 0 1	1 0 0	1 0 0	1 0 0	0 1 0
1 1 1	-	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0

Verilog Code:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////

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;   // sec7=TMG,
sec5=TTG, sec2=TY, sec3=TSG.
```

```
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
```

Test Bench:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////
module Traffic_Light_Controller_TB;
reg clk,rst;
wire [2:0]light_M1;
wire [2:0]light_S;
wire [2:0]light_MT;
wire [2:0]light_M2;

Traffic_Light_Controller dut(.clk(clk) , .rst(rst) , .light_M1(light_M1) ,
.light_S(light_S) ,.light_M2(light_M2),.light_MT(light_MT) );

initial
begin

	clk=1'b0;
	forever #(1000000000/2) clk=~clk;
end

initial
begin

	rst=0;
	#1000000000;
	rst=1;
	#1000000000;
	rst=0;
	#(1000000000*200);
	$finish;
end
endmodule
```