

1.	Aim of the Experiment/Simulation (Task details):
To : <ul style="list-style-type: none"> Design a Traffic Light Controller using FSM. Write the Verilog code and Verify with Test Bench 	
2.	Name of the Simulation Platform/Tools: (If applicable)
Model-Sim	
3.	Theory:
<pre> graph TD Start([Start]) --> Emergency{Emergency in any road?} Emergency -- Y --> Green1[Go to Green State of that Road] Emergency -- N --> Jam{Jam in any road?} Jam -- Y --> Green2[Go to Green State of that Road] Jam -- N --> LightGreen{Is Light on Road Green?} LightGreen -- Y --> Empty{Is Road Empty or 20s have passed?} LightGreen -- N --> FourSec{4s have passed?} Empty -- Y --> Yellow[Go to Yellow State of that Road] Empty -- N --> LightGreen FourSec -- Y --> GreenNext[Go to Green State of the next Road] FourSec -- N --> LightGreen Yellow --> LightGreen GreenNext --> LightGreen </pre>	

Description of states

State	State (Binary)	State Name	East	North	West	South
0	000	East_green	Green	Red	Red	Red
1	001	East_yellow	Yellow	Red-Yellow	Red	Red
2	010	North_green	Red	Green	Red	Red
3	011	North_yellow	Red	Yellow	Red-Yellow	Red
4	100	West_green	Red	Red	Green	Red
5	101	West_yellow	Red	Red	Yellow	Red-Yellow
6	110	South_green	Red	Red	Red	Green
7	111	South_yellow	Red-Yellow	Red	Red	Yellow

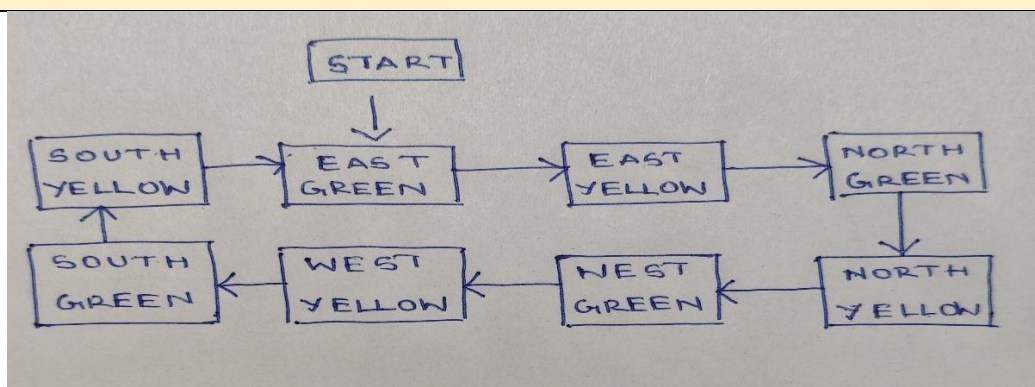
Emergency condition :

Direction	Emergency				State		
	Emergency [3]	Emergency [2]	Emergency [1]	Emergency [0]	State [2]	State [1]	State [0]
East	1	0	0	0	0	0	0
North	0	1	0	0	0	1	0
West	0	0	1	0	1	0	0
South	0	0	0	1	1	1	0

Jam Condition :

Direction	Jam				State		
	Jam [3]	Jam [2]	Jam [1]	Jam [0]	State [2]	State [1]	State [0]
East	1	0	0	0	0	0	0
North	0	1	0	0	0	1	0
West	0	0	1	0	1	0	0
South	0	0	0	1	1	1	0

4. State diagram:



5. Verilog HDL Code and Outputs:

```
// Tarun Rithvik B (23BEC0014)
module
Traffic_Light_Controller(East_road,North_road,West_road,South_road,clk,rst,Emergency,Jam,E
mpty);
input clk,rst;
input [3:0]Emergency,Jam,Empty;//Emergency/Jam/Empty=0000, for east 1000,north 0100,west
0010,south 0001
output reg[2:0]
East_road,North_road,West_road,South_road;//Red=100,Yellow=010,Green=001,Red&Yellow=
110,none=000

reg[2:0] state;

parameter [2:0] east_green=3'b000;
parameter [2:0] east_yellow=3'b001;
parameter [2:0] north_green=3'b010;
parameter [2:0] north_yellow=3'b011;
parameter [2:0] west_green=3'b100;
parameter [2:0] west_yellow=3'b101;
parameter [2:0] south_green=3'b110;
parameter [2:0] south_yellow=3'b111;

reg[4:0] count;

always@(posedge clk, negedge rst)
begin
if(!rst)
begin
count=5'b000000;
end

else if(|Emergency)
begin
state={Emergency[1]|Emergency[0],Emergency[2]|Emergency[0],1'b0};
count=5'b000000;
end

else if(|Jam)
begin
state={Jam[1]|Jam[0],Jam[2]|Jam[0],1'b0};
count=5'b000000;
end

else
begin
case(state)

east_green:
begin
if(count==5'b10011||Empty==4'b1000)
```

```

    begin
    count=5'b00000;
    state=east_yellow;
    end
else
    begin
    count=count+5'b00001;
    state=east_green;
    end
end

east_yellow:
begin
if(count==5'b00011)
begin
count=5'b00000;
state=north_green;
end
else
begin
count=count+5'b00001;
state=east_yellow;
end
end

north_green:
begin
if(count==5'b10011||Empty==4'b0100)
begin
count=5'b00000;
state=north_yellow;
end
else
begin
count=count+5'b00001;
state=north_green;
end
end

north_yellow:
begin
if(count==5'b00011)
begin
count=5'b00000;
state=west_green;
end
else
begin
count=count+5'b00001;
state=north_yellow;
end
end

```

```
west_green:
begin
if(count==5'b10011||Empty==4'b0010)
begin
count=5'b00000;
state=west_yellow;
end
else
begin
count=count+5'b000001;
state=west_green;
end
end
```

```
west_yellow:
begin
if(count==5'b00011)
begin
count=5'b00000;
state=south_green;
end
else
begin
count=count+5'b00001;
state=west_yellow;
end
end
```

```
south_green:
begin
if(count==5'b10011||Empty==4'b0001)
begin
count=5'b00000;
state=south_yellow;
end
else
begin
count=count+5'b00001;
state=south_green;
end
end
```

```
south_yellow:
begin
if(count==5'b00011)
begin
count=5'b00000;
state=east_green;
end
else
begin
```

```

        count=count+5'b00001;
        state=south_yellow;
    end
end

default:
    begin
        count=5'b00000;
        state=east_green;

    end

endcase
end
end

//Red=100,Yellow=010,Green=001,Red&Yellow=110,none=000
always@(state)
    begin
        if(!rst)
            begin
                East_road=3'b000;
                North_road=3'b000;
                West_road=3'b000;
                South_road=3'b000;
            end
        else
            begin
                case(state)

                    east_green:
                        begin
                            East_road=3'b001;
                            North_road=3'b100;
                            West_road=3'b100;
                            South_road=3'b100;
                        end

                    east_yellow:
                        begin
                            East_road=3'b010;
                            North_road=3'b110;//
                            West_road=3'b100;
                            South_road=3'b100;
                        end

                    north_green:
                        begin
                            East_road=3'b100;
                            North_road=3'b001;
                            West_road=3'b100;
                            South_road=3'b100;

```

```

    end

    north_yellow:
    begin
        East_road=3'b100;
        North_road=3'b010;
        West_road=3'b110;//
        South_road=3'b100;
    end

    west_green:
    begin
        East_road=3'b100;
        North_road=3'b100;
        West_road=3'b001;
        South_road=3'b100;
    end

    west_yellow:
    begin
        East_road=3'b100;
        North_road=3'b100;
        West_road=3'b010;
        South_road=3'b110; //
    end

    south_green:
    begin
        East_road=3'b100;
        North_road=3'b100;
        West_road=3'b100;
        South_road=3'b001;
    end

    south_yellow:
    begin
        East_road=3'b110;//
        North_road=3'b100;
        West_road=3'b100;
        South_road=3'b010;
    end

    endcase
end
endmodule

module tb_Traffic_Light_Controller;

    reg clk;
    reg rst;
    reg [3:0] Emergency;

```

```

reg [3:0] Jam;
reg [3:0] Empty;

wire [2:0] East_road;
wire [2:0] North_road;
wire [2:0] West_road;
wire [2:0] South_road;

Traffic_Light_Controller uut (
    .clk(clk),
    .rst(rst),
    .Emergency(Emergency),
    .Jam(Jam),
    .Empty(Empty),
    .East_road(East_road),
    .North_road(North_road),
    .West_road(West_road),
    .South_road(South_road)
);

always #5 clk = ~clk;
initial begin

    clk = 0;
    rst = 1;
    Emergency = 4'b0000;
    Jam = 4'b0000;
    Empty = 4'b0000;
    #10;
    rst = 0;
    #10;
    rst = 1;
    #20;
    Empty = 4'b0000;
    #200;
    Emergency = 4'b1000;
    #20;
    Emergency = 4'b0000;
    #20;
    Jam = 4'b0100;
    #20;
    Jam = 4'b0000;
    #20;
    Empty = 4'b1000;
    #40;
    Empty = 4'b0100;
    #40;
    Empty = 4'b0010;
    #40;
    Empty = 4'b0001;
    #40;

```



```

Empty = 4'b0000;
#200;
$stop;
end

initial begin
    $monitor("Time=%0t | East=%b | North=%b | West=%b | South=%b | Emergency=%b |
Jam=%b | Empty=%b",
        $time, East_road, North_road, West_road, South_road, Emergency, Jam, Empty);
end
endmodule

```

```

1 // Tarun Rithvik B (23BEC0014)
2 module Traffic_Light_Controller(East_road,North_road,West_road,South_road,clk,rst,Emergency,Jam,Empty);
3     input clk,rst;
4     input [3:0]Emergency,Jam,Empty;//Emergency/Jam/Empty=0000, for east 1000,north 0100,west 0010,south 0001
5     output reg[2:0] East_road,North_road,West_road,South_road;//Red=100,Yellow=010,Green=001,Red&Yellow=110,none=000
6
7     reg[2:0] state;
8
9
10    parameter [2:0] east_green=3'b000;
11    parameter [2:0] east_yellow=3'b001;
12    parameter [2:0] north_green=3'b010;
13    parameter [2:0] north_yellow=3'b011;
14    parameter [2:0] west_green=3'b100;
15    parameter [2:0] west_yellow=3'b101;
16    parameter [2:0] south_green=3'b110;
17    parameter [2:0] south_yellow=3'b111;
18
19    reg[4:0] count;
20
21    always@(posedge clk, negedge rst)
22    begin
23        if(!rst)
24        begin
25            count=5'b00000;
26        end
27
28        else if(!Emergency)
29        begin
30            state={Emergency[1]|Emergency[0],Emergency[2]|Emergency[0],1'b0};
31            count=5'b00000;
32        end
33

```

```

34        else if(!Jam)
35        begin
36            state={Jam[1]|Jam[0],Jam[2]|Jam[0],1'b0};
37            count=5'b00000;
38        end
39
40        else
41        begin
42            case(state)
43
44                east_green:
45                begin
46                    if(count==5'b10011||Empty==4'b1000)
47                    begin
48                        count=5'b00000;
49                        state=east_yellow;
50                    end
51                    else
52                    begin
53                        count=count+5'b00001;
54                        state=east_green;
55                    end
56                end
57
58                east_yellow:
59                begin
60                    if(count==5'b00011)
61                    begin
62                        count=5'b00000;
63                        state=north_green;
64                    end
65                    else
66                    begin

```

```

67         count=count+5'b00001;
68         state=east_yellow;
69     end
70 end
71
72 north_green:
73 begin
74     if (count==5'b10011 || Empty==4'b0100)
75     begin
76         count=5'b00000;
77         state=north_yellow;
78     end
79     else
80     begin
81         count=count+5'b00001;
82         state=north_green;
83     end
84     end
85
86 north_yellow:
87 begin
88     if (count==5'b00011)
89     begin
90         count=5'b00000;
91         state=west_green;
92     end
93     else
94     begin
95         count=count+5'b00001;
96         state=north_yellow;
97     end
98     end
99

```

```

100 west_green:
101 begin
102     if (count==5'b10011 || Empty==4'b0010)
103     begin
104         count=5'b00000;
105         state=west_yellow;
106     end
107     else
108     begin
109         count=count+5'b000001;
110         state=west_green;
111     end
112     end
113
114 west_yellow:
115 begin
116     if (count==5'b00011)
117     begin
118         count=5'b00000;
119         state=south_green;
120     end
121     else
122     begin
123         count=count+5'b00001;
124         state=west_yellow;
125     end
126     end
127
128 south_green:
129 begin
130     if (count==5'b10011 || Empty==4'b0001)
131     begin
132         count=5'b00000;
133         state=south_yellow;
134     end
135     else
136     begin
137         count=count+5'b00001;

```

```

138         state=south_green;
139     end
140 end
141
142 south_yellow:
143     begin
144         if(count==5'b00011)
145             begin
146                 count=5'b00000;
147                 state=east_green;
148             end
149         else
150             begin
151                 count=count+5'b00001;
152                 state=south_yellow;
153             end
154         end
155
156     default:
157         begin
158             count=5'b00000;
159             state=east_green;
160
161         end
162
163     endcase
164 end
165 end
166
167 //Red=100,Yellow=010,Green=001,Red&Yellow=110,none=000
168 always@(state)
169     begin
170         if(!rst)
171             begin
172                 East_road=3'b000;
173                 North_road=3'b000;
174                 West_road=3'b000;
175                 South_road=3'b000;

```

```

176     end
177 else
178     begin
179         case(state)
180
181         east_green:
182             begin
183                 East_road=3'b001;
184                 North_road=3'b100;
185                 West_road=3'b100;
186                 South_road=3'b100;
187             end
188
189         east_yellow:
190             begin
191                 East_road=3'b010;
192                 North_road=3'b110;//
193                 West_road=3'b100;
194                 South_road=3'b100;
195             end
196
197         north_green:
198             begin
199                 East_road=3'b100;
200                 North_road=3'b001;
201                 West_road=3'b100;
202                 South_road=3'b100;
203             end
204
205         north_yellow:
206             begin
207                 East_road=3'b100;
208                 North_road=3'b010;
209                 West_road=3'b110;//
210                 South_road=3'b100;
211             end
212
213         west_green:

```

```

214     begin
215         East_road=3'b100;
216         North_road=3'b100;
217         West_road=3'b001;
218         South_road=3'b100;
219     end
220
221     west_yellow:
222     begin
223         East_road=3'b100;
224         North_road=3'b100;
225         West_road=3'b010;
226         South_road=3'b110; //
227     end
228
229     south_green:
230     begin
231         East_road=3'b100;
232         North_road=3'b100;
233         West_road=3'b100;
234         South_road=3'b001;
235     end
236
237     south_yellow:
238     begin
239         East_road=3'b110; //
240         North_road=3'b100;
241         West_road=3'b100;
242         South_road=3'b010;
243     end
244
245     endcase
246 end
247 end
248 endmodule
249
250 module tb_Traffic_Light_Controller;
251

```

```

252     reg clk;
253     reg rst;
254     reg [3:0] Emergency;
255     reg [3:0] Jam;
256     reg [3:0] Empty;
257
258     wire [2:0] East_road;
259     wire [2:0] North_road;
260     wire [2:0] West_road;
261     wire [2:0] South_road;
262
263     Traffic_Light_Controller uut (
264         .clk(clk),
265         .rst(rst),
266         .Emergency(Emergency),
267         .Jam(Jam),
268         .Empty(Empty),
269         .East_road(East_road),
270         .North_road(North_road),
271         .West_road(West_road),
272         .South_road(South_road)
273     );
274
275
276     always #5 clk = ~clk;
277     initial begin
278
279         clk = 0;
280         rst = 1;
281         Emergency = 4'b0000;
282         Jam = 4'b0000;
283         Empty = 4'b0000;
284         #10;
285         rst = 0;
286         #10;
287         rst = 1;
288         #20;
289         Empty = 4'b0000;

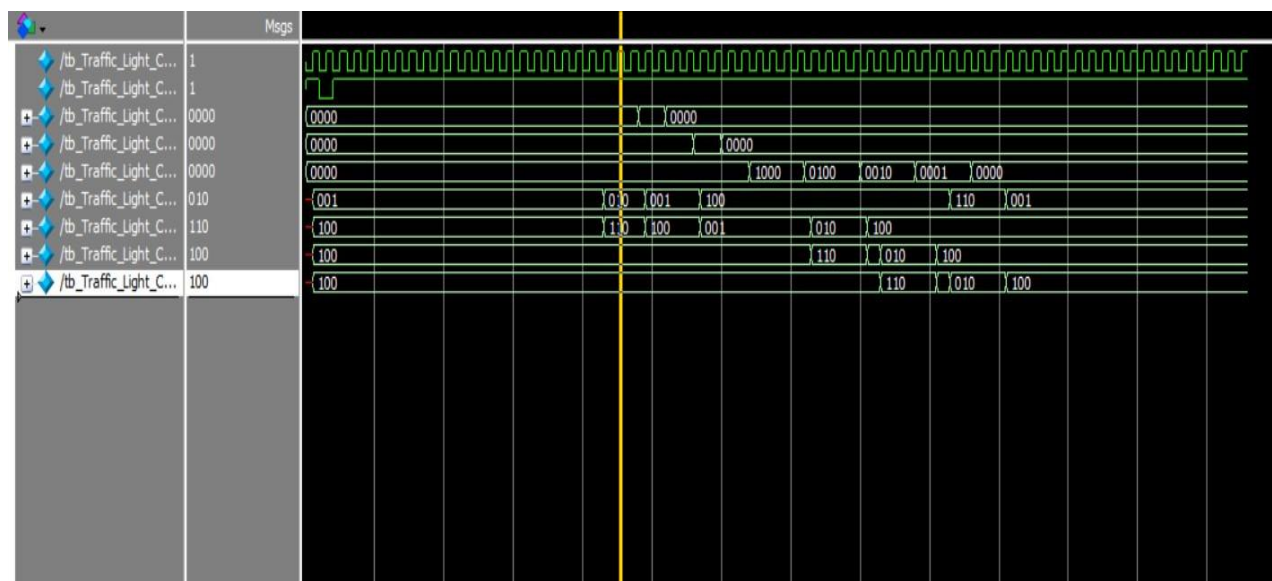
```

```

290     #200;
291     Emergency = 4'b1000;
292     #20;
293     Emergency = 4'b0000;
294     #20;
295     Jam = 4'b0100;
296     #20;
297     Jam = 4'b0000;
298     #20;
299     Empty = 4'b1000;
300     #40;
301     Empty = 4'b0100;
302     #40;
303     Empty = 4'b0010;
304     #40;
305     Empty = 4'b0001;
306     #40;
307     Empty = 4'b0000;
308     #200;
309     $stop;
310 end
311
312 initial begin
313     $monitor("Time=%0t | East=%b | North=%b | West=%b | South=%b | Emergency=%b | Jam=%b | Empty=%b",
314             $time, East_road, North_road, West_road, South_road, Emergency, Jam, Empty);
315 end
316 endmodule
317

```

Output:



6. Conclusions:

Thus using the above logic we have designed a 4 way traffic control system.