3844 Ojasvi Tummala
3851 Preeti Raut
3853 Aparajita Sarkar

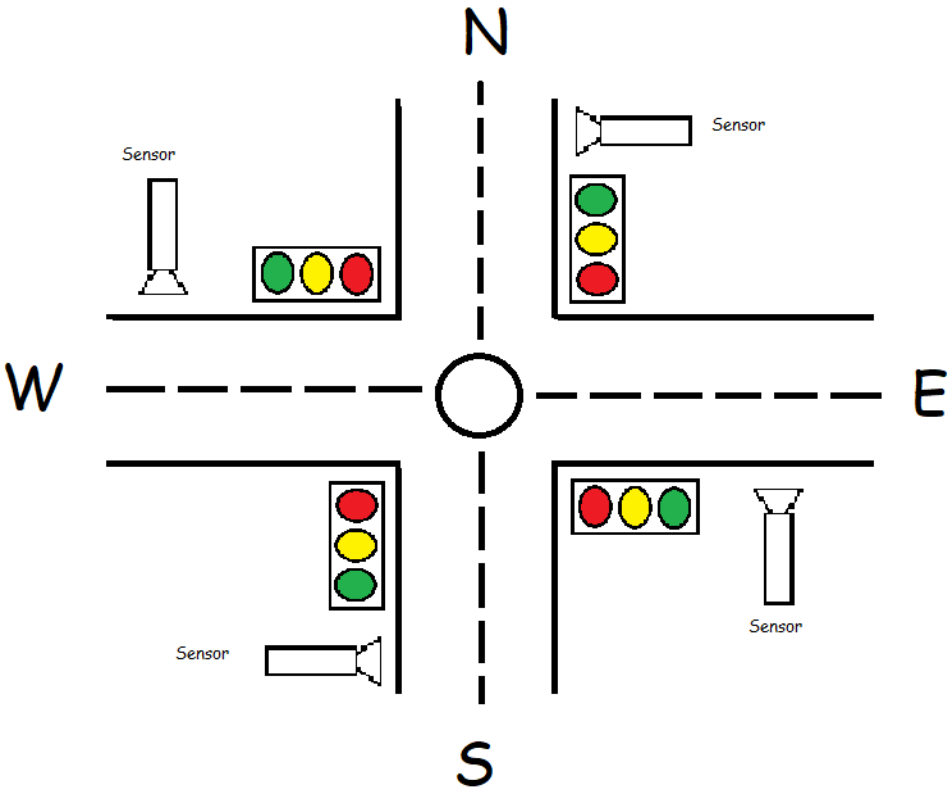
Problem statement:

Design of 4-way Traffic Light Controller based on Finite State Machine (FSM) using Verilog in the order of East, North, West and South.

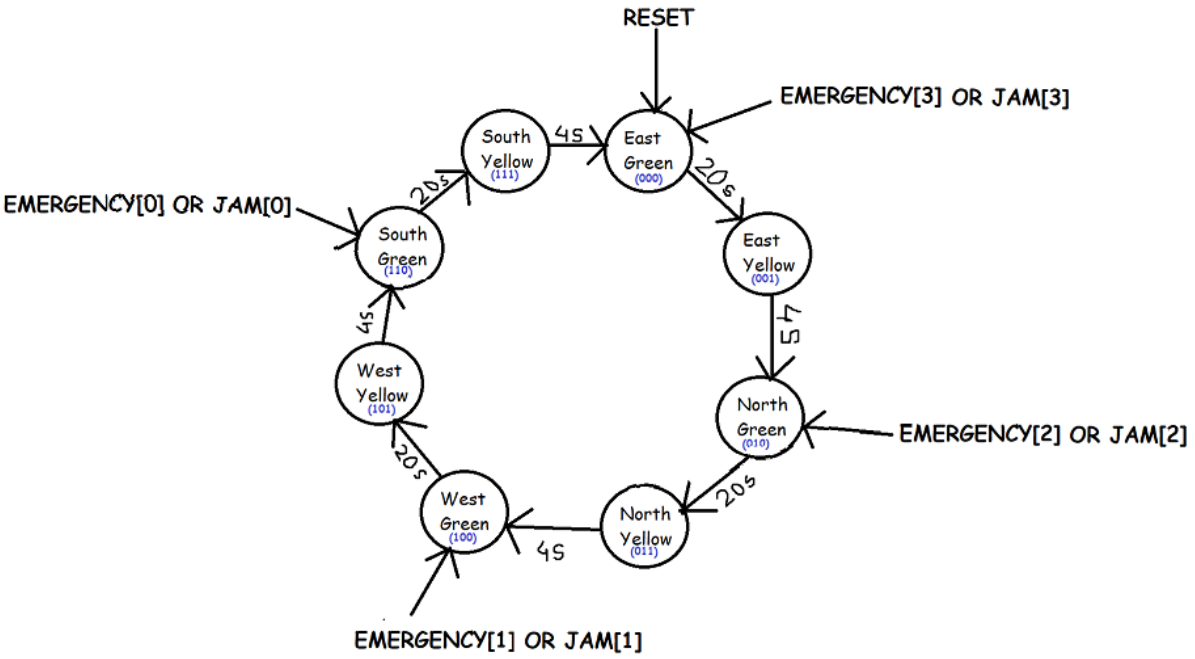
Features:

- If there is an emergency vehicle on a road, signal for that road is turned green immediately.
- If there is jam on a road, signal for that road is turned green immediately.
- If there are no vehicles on a road, signal for that road is turned state.

Diagrammatic representation:



State diagram:



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:

Description	Emergency				State		
	[3]	[2]	[1]	[0]	[2]	[1]	[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:

Description	Jam				State		
	[3]	[2]	[1]	[0]	[2]	[1]	[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

Verilog Code:

```
`timescale 1ns / 1ps
module Traffic (clk, rst, Emergency, Jam, Empty, East_road, North_road, West_road,
South_road);
   input clk;
   input rst;
   input [03:0] Emergency;
   input [03:0] Jam;
   input [03:0] Empty;
   output reg [02:0] East_road;
   output reg [02:0] North_road;
   output reg [02:0] West_road;
   output reg [02:0] South_road;

        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 [2:0] state;
        reg[4:0] count;
        reg[22:0] temp;

        always @ (posedge clk)
                temp<=temp+1;

        always @ (posedge temp[22], negedge rst)
        begin
                if (!rst)
                        count=5'b00000;
                else if (|Emergency)
                begin
                        state={Emergency[1] | Emergency[0] , Emergency[2]|
Emergency[0],1'b0};
                        count=5'b00000;
                end
                else if(|Jam)
                begin
                        state={Jam[1] | Jam[0] , Jam [2]| Jam [0],1'b0};
                        count=5'b00000;
                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'b00001;
                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
```

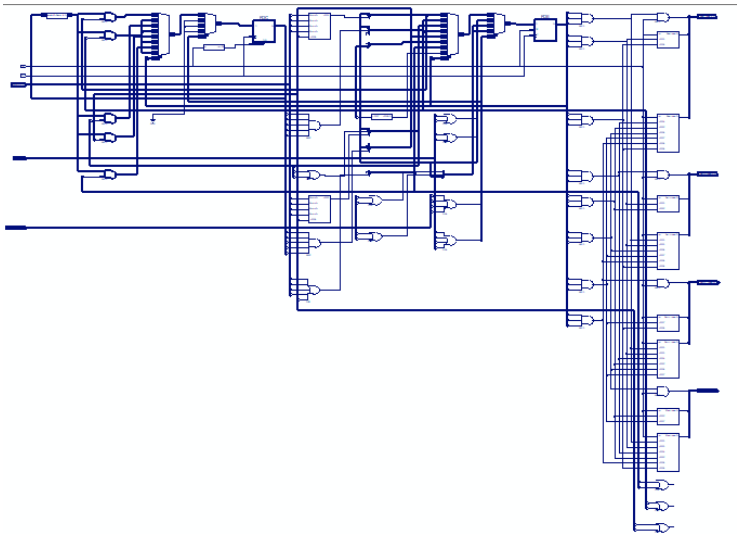
```
                        end
                        default:
                        begin
                                count=5'b00000;
                                state=east_green;
                        end
                endcase
        end
end
//red=100,yellow=010,green=001,red&amp;yellow=110,none=000

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
    end
endmodule
```

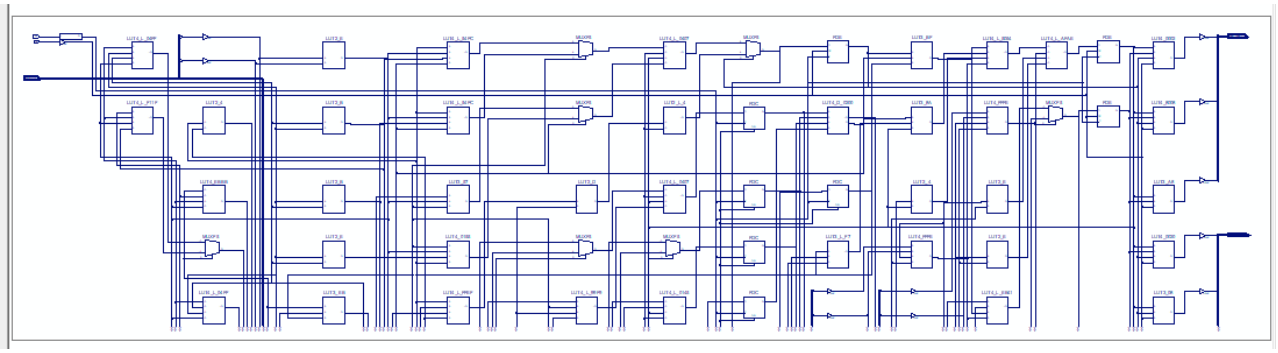
Device utilization Summary:

Number of Slices:	53 out of 2448	2%
Number of Slice Flip Flops:	12 out of 4896	0%
Number of 4 input LUTs:	101 out of 4896	2%
Number of bonded IOBs:	26 out of 158	16%
Number of GCLKs:	1 out of 24	4%

RTL Schematic:

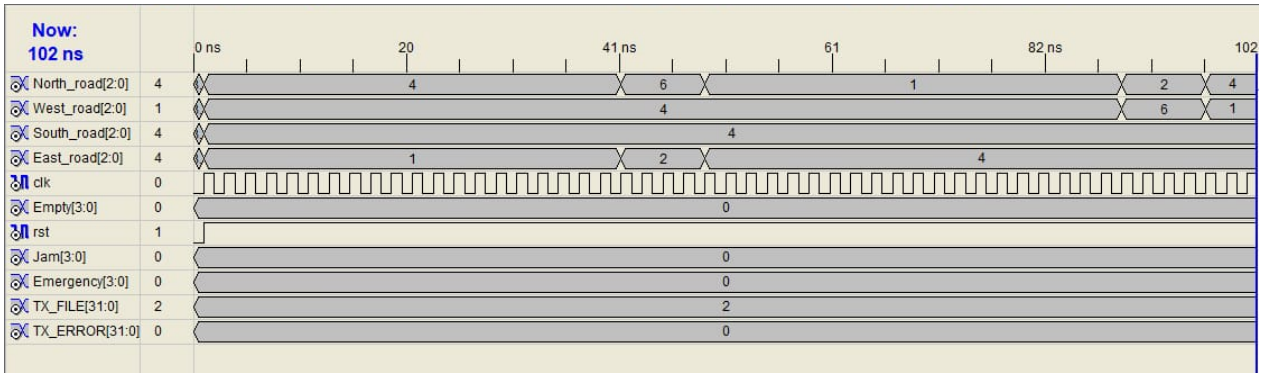


Technology Schematic

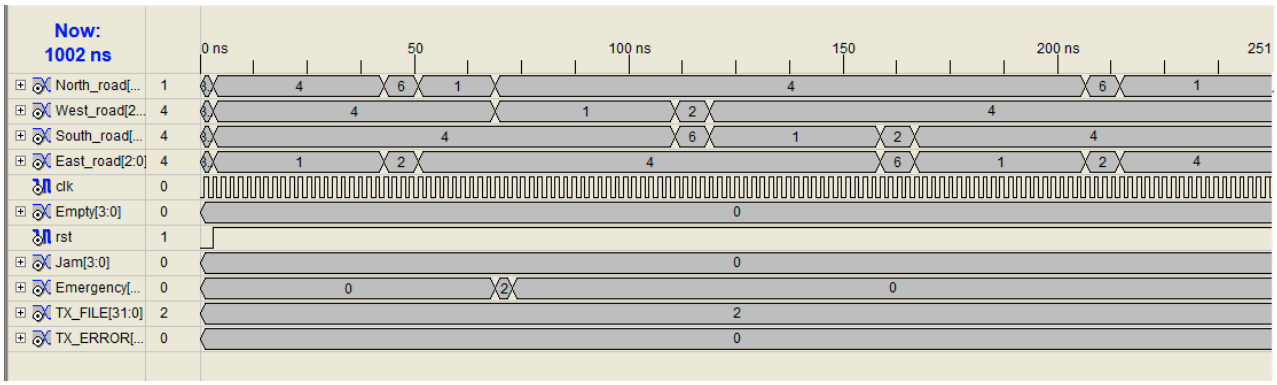


Simulation Report:

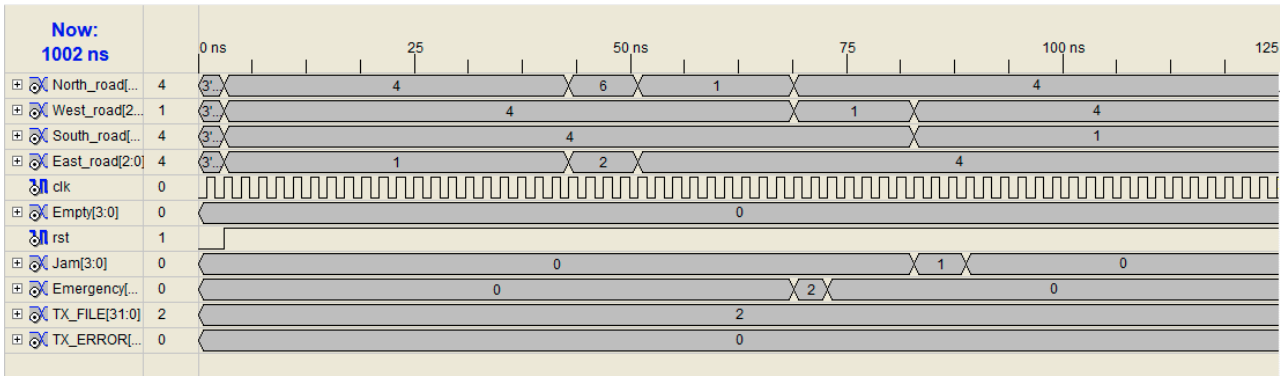
- **4 Way Traffic Light Controller without Emergency, Jam:**



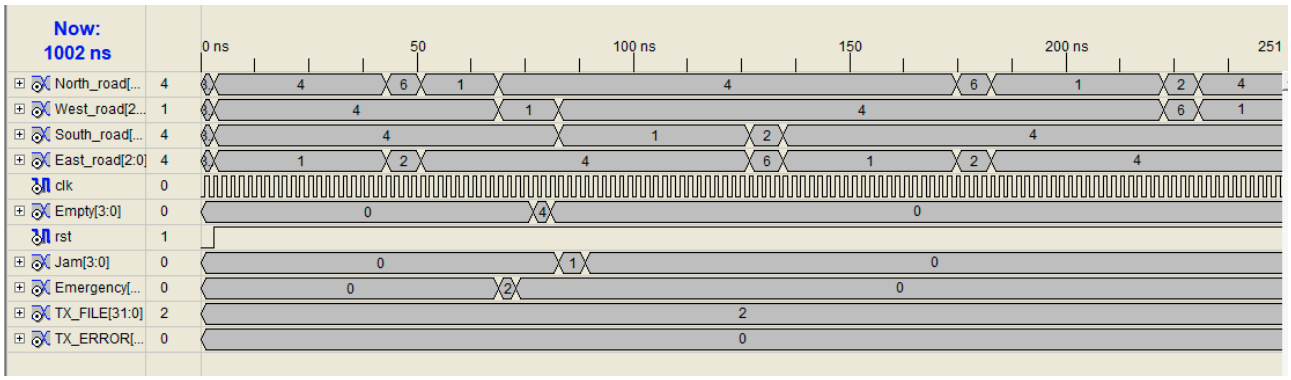
- **For Emergency in West Road:**



- **Jam condition in South Road:**



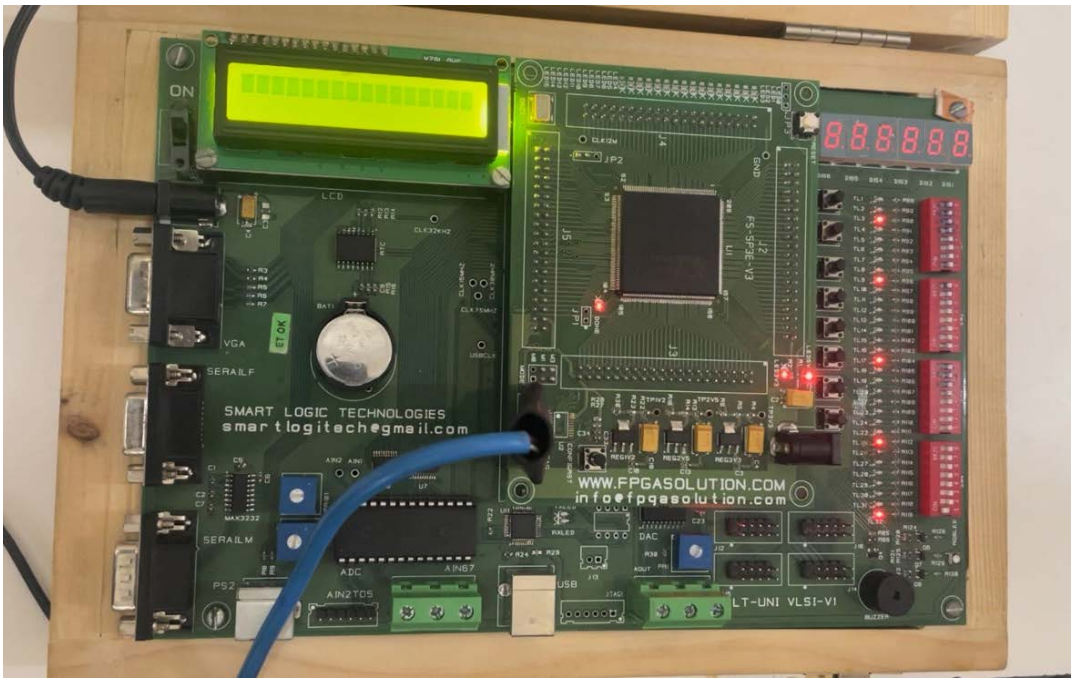
- **Empty Condition:**

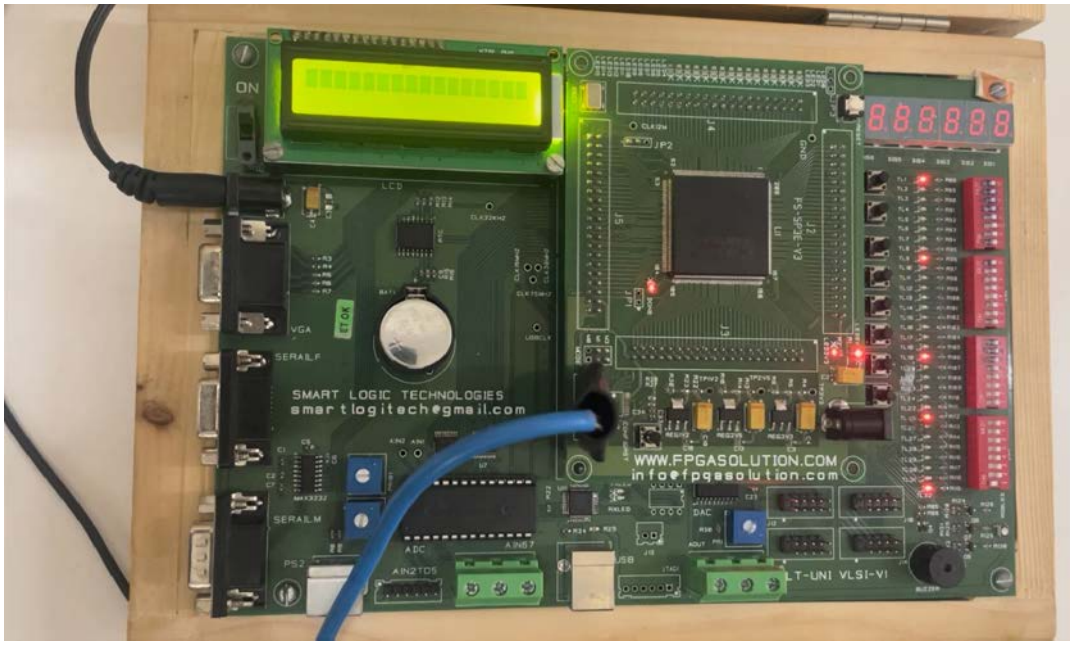


Implementation on FPGA XC3S250E:

Pin Assignment:

I/O Name	DIP Switch	LOC
clk	-	P80
East_road[0]	SW1	P203
East_road[1]		P206
East_road[2]		P205
Emergency[0]	SW1	P196
Emergency[1]		P199
Emergency[2]		P197
Emergency[3]		P202
Empty[0]	SW2	P181
Empty [1]		P185
Empty [2]		P187
Empty [3]		P186
Jam[0]	SW3	P172
Jam [1]		P171
Jam [2]		P168
Jam [3]		P152
North_road[0]	SW2	P189
North_road [1]		P193
North_road [2]		P192
rst	SW4	P153
South_road [0]	SW3	P163
South_road [1]		P167
South_road [2]		P165
West_road [0]	SW4	P177
West_road [1]		P180
West_road [2]		P179


ON
LCD
VGA
SERAILF
SERAILM
PS2
SMART LOGIC TECHNOLOGIES
smartlogitech@gmail.com
WWW.FPGASOLUTION.COM
info@fpgasolution.com
LT-UNI VLSI-VI


ON
LCD
VGA
SERAILF
SERAILM
PS2
SMART LOGIC TECHNOLOGIES
smartlogitech@gmail.com
WWW.FPGASOLUTION.COM
info@fpgasolution.com
LT-UNI VLSI-VI