

PROJECT WORK

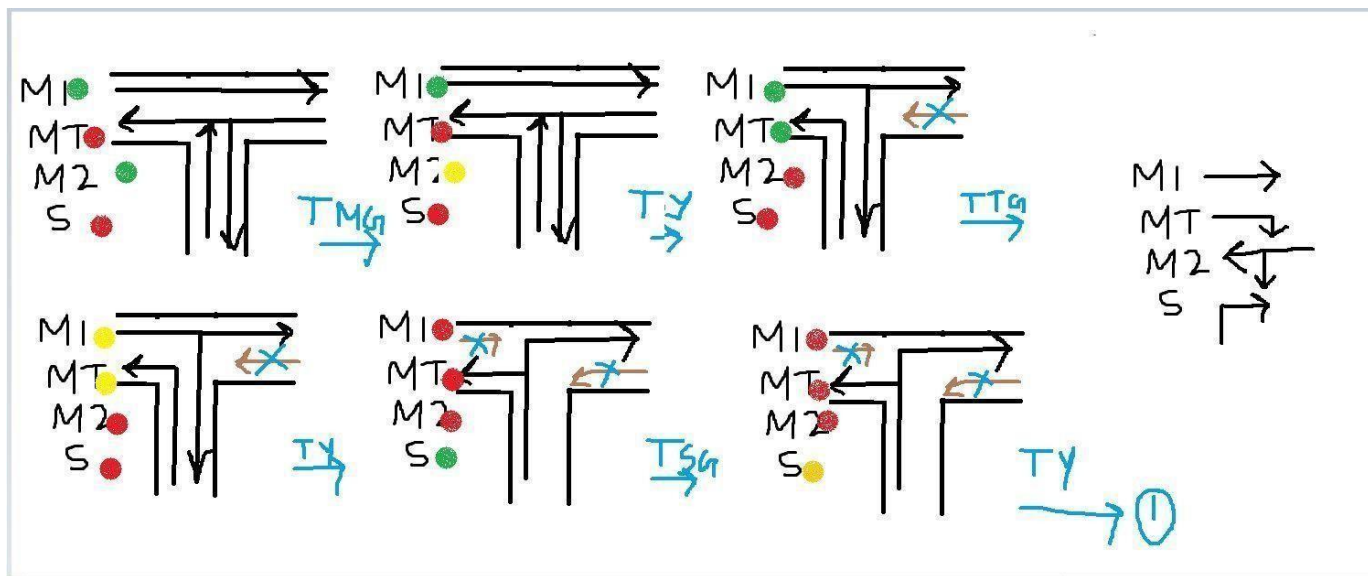
"Verilog Implementation of Digital Circuit Designs on FPGA using Vivado"

Topic: Traffic Light Controller Design using Verilog

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.



The six cases present here eventually turn to the six states .

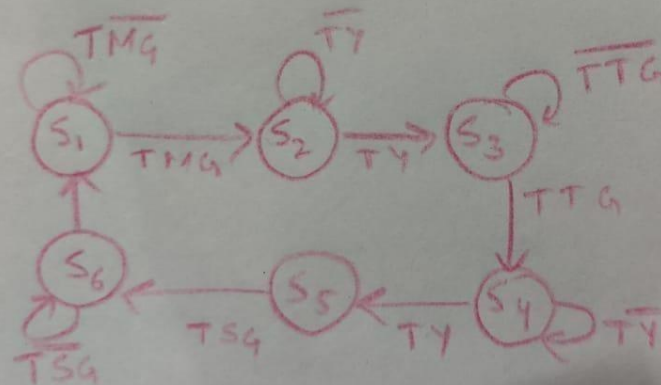
This is the state diagram:

$\overline{TMG} - 7A$

$\overline{TY} - 2A$

$\overline{TTG} - 5A$

$\overline{TS4} - 3A$



Shot on OnePlus
By Arjun

From the state diagram we for the state table:

State Table

Present state ABC	Input	NS A'B'C'		M1 RYG	M2 RYG	T RYG	S RYG	
001	$\overline{T}Y$	001	}	001	001	100	100	
001	TY	010						
010	$\overline{T}Y$	010	}	001	010	100	100	
	TY	011						
011	$\overline{T}Y$	011	}	001	100	001	100	
	TY	100						
100	$\overline{T}Y$	100	}	010	100	010	100	
	TY	101						
101	$\overline{T}S$	101	}	100	100	100	001	
	TS	110						
110	$\overline{T}Y$	110	}	100	100	100	010	
	TY	001						
111	-	000		000	000	000	000	

VERILOG CODE

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

```
////////////////////////////////////////////////////////////////
```

```
// Module Name: Traffic_Light_Controller // Project Name:
```

```
////////////////////////////////////////////////////////////////
```

```
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

```

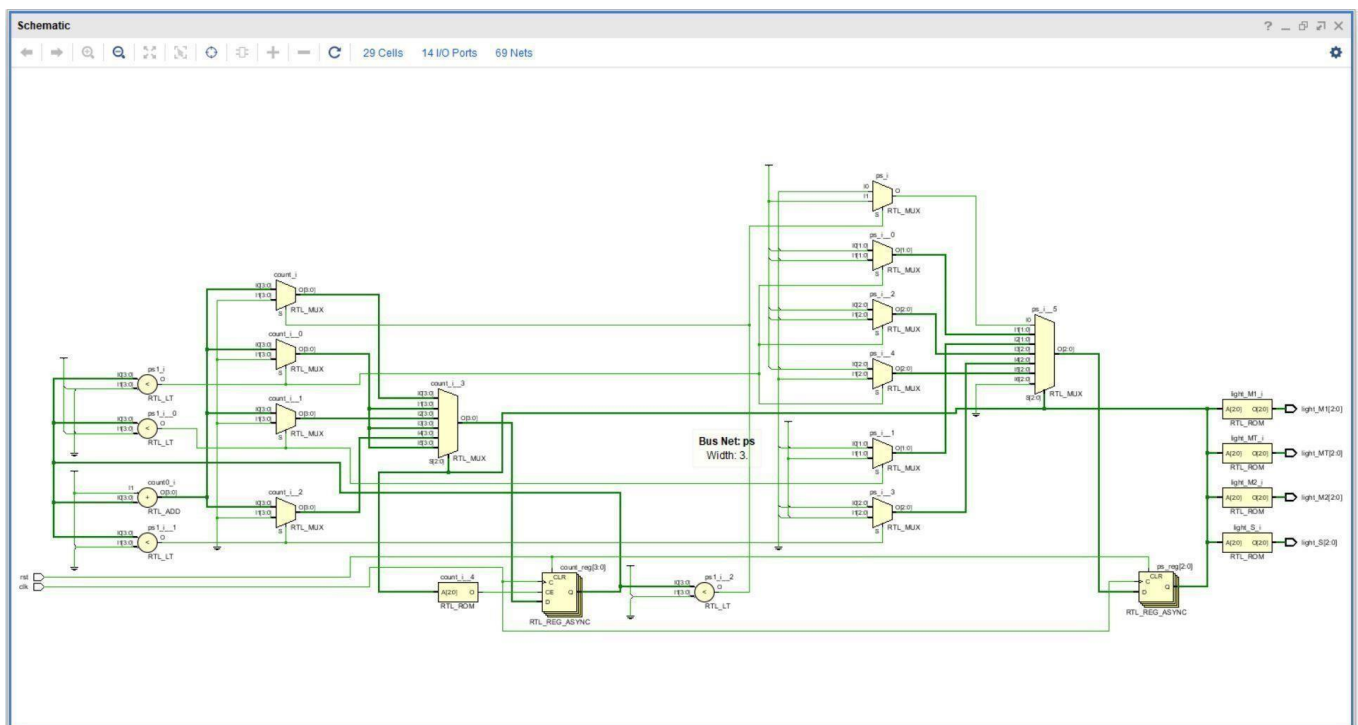
```

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

RTL-SCHEMATIC



TESTBENCH

`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;

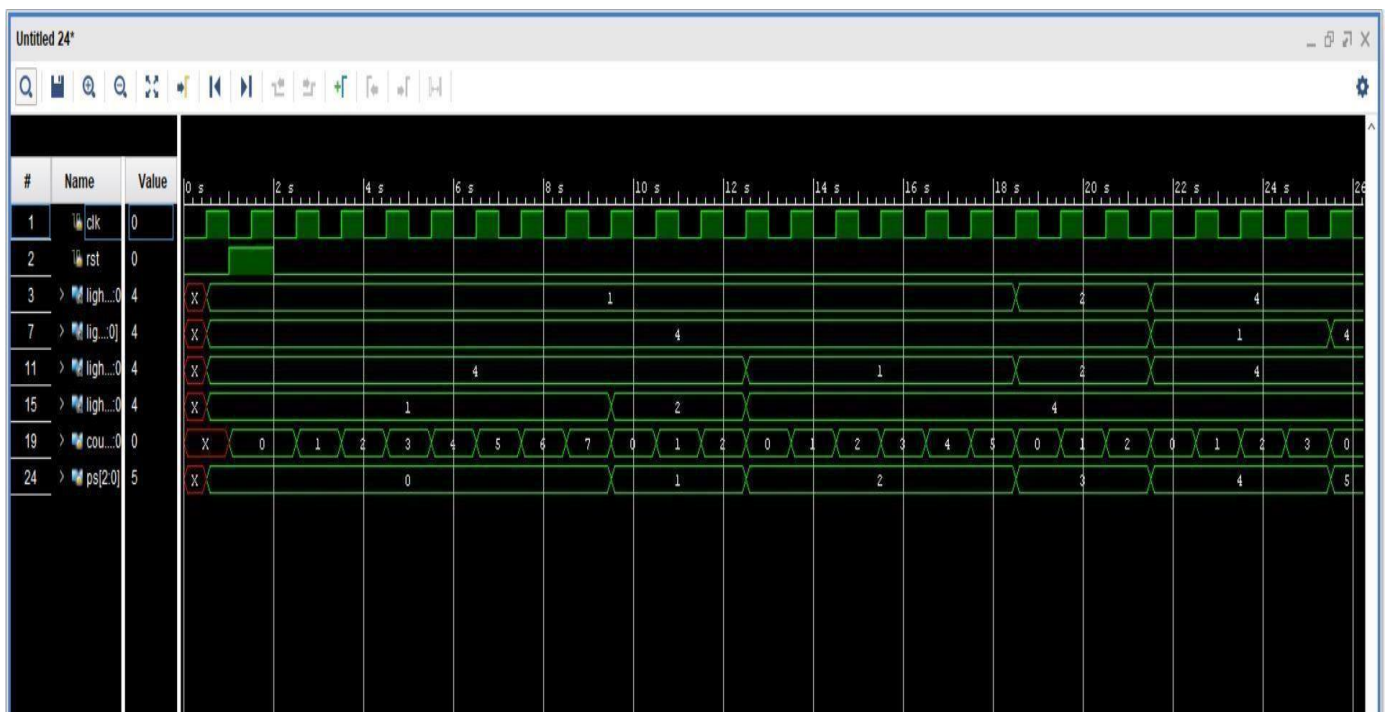
#(1000000000*200);

$finish;  end

endmodule

```

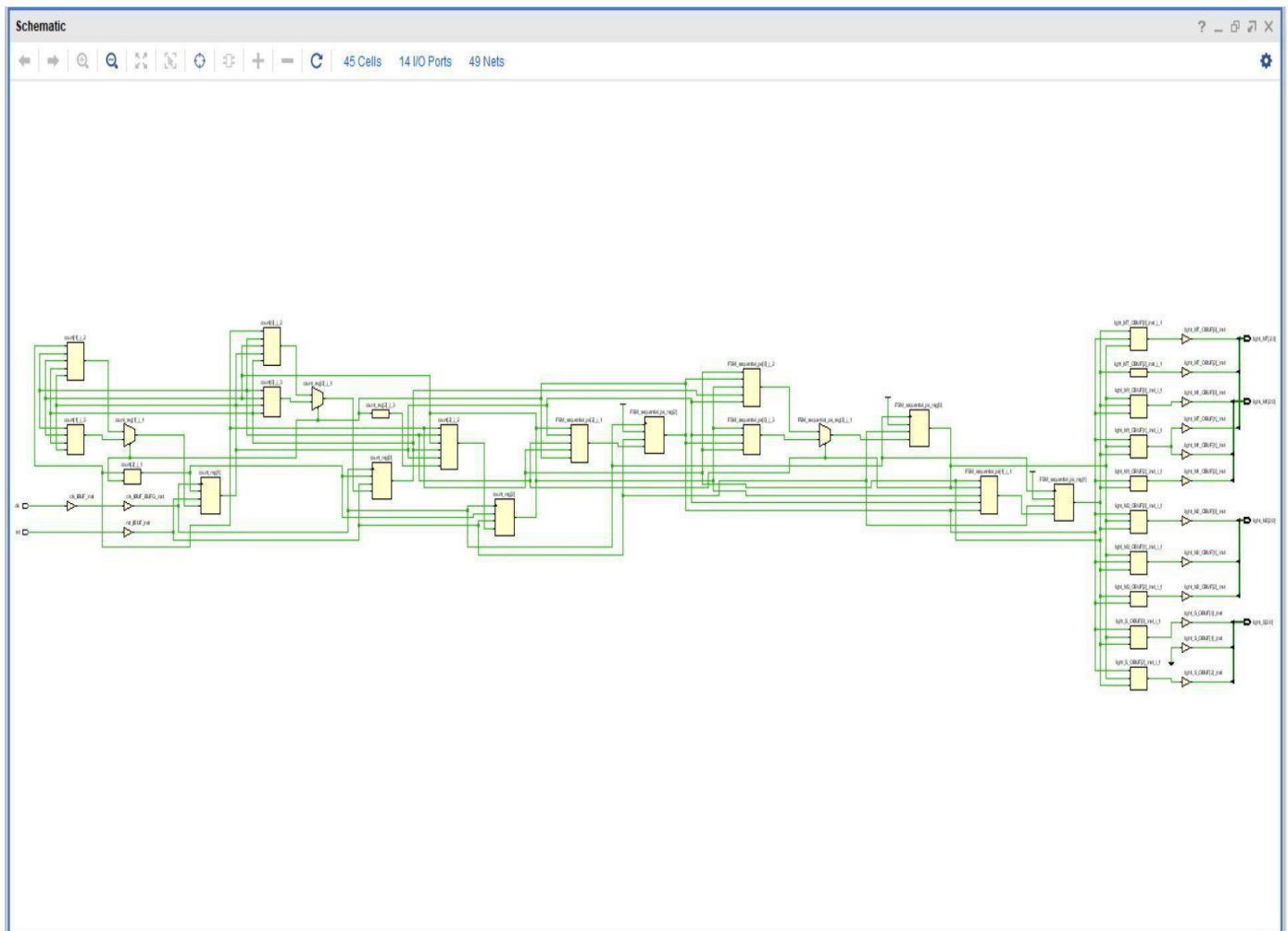
SIMULATED WAVEFORM



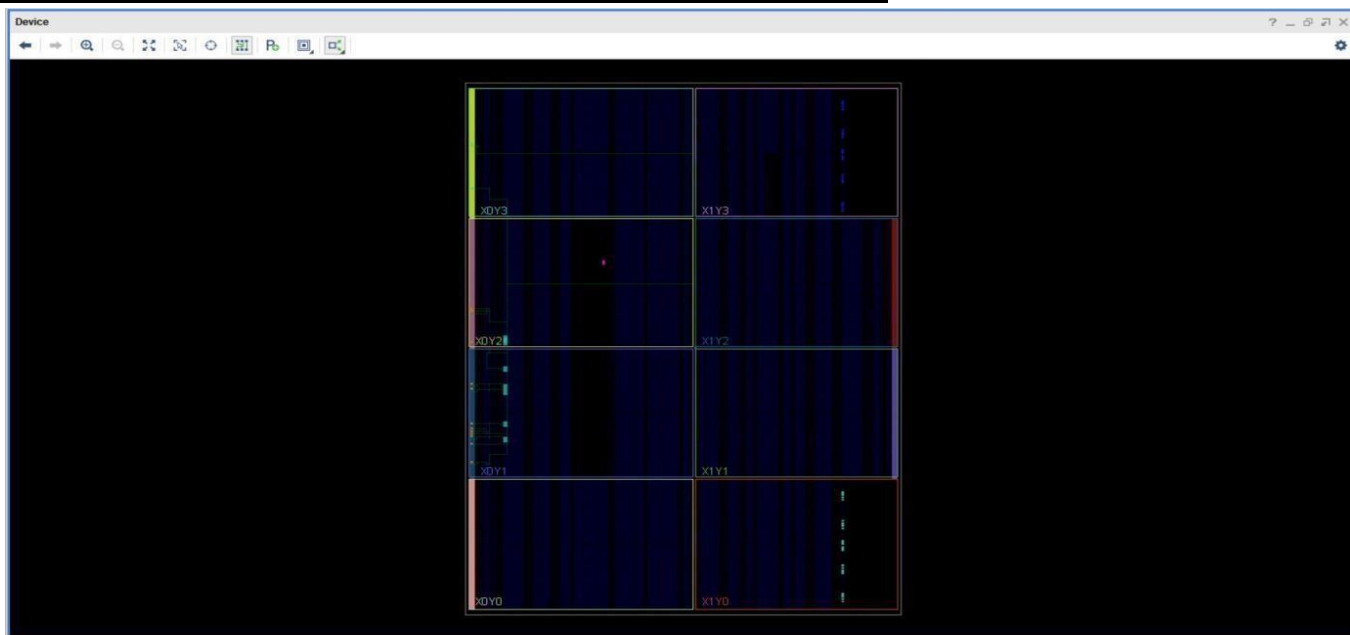
Upon analysing the waveform we can clearly see that the FSM works perfectly.

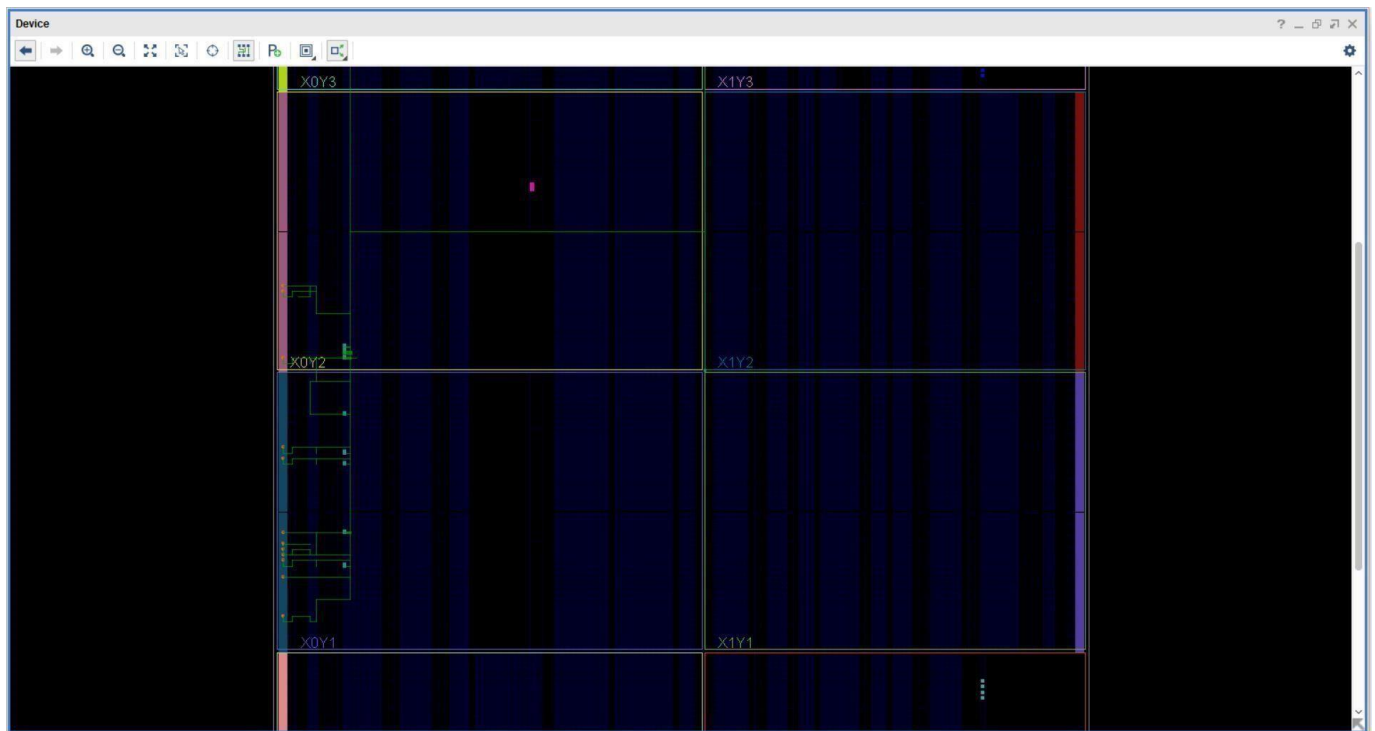
The ports are assigned from the ucf file .

SCHEMATIC AFTER SYNTHESIS



DEVICE LAYOUT AFTER IMPLEMENTATION





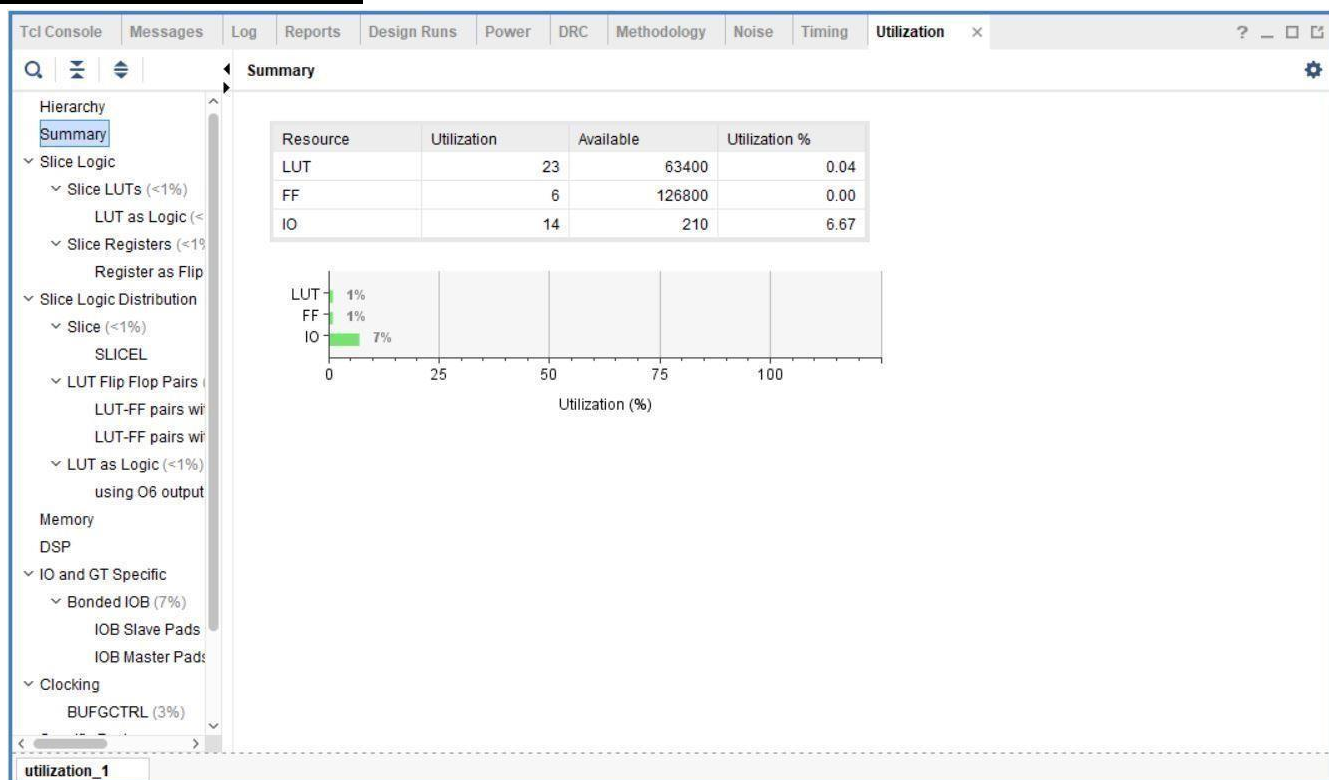
REPORTS AFTER SYNTHESIS

TIMING REPORT

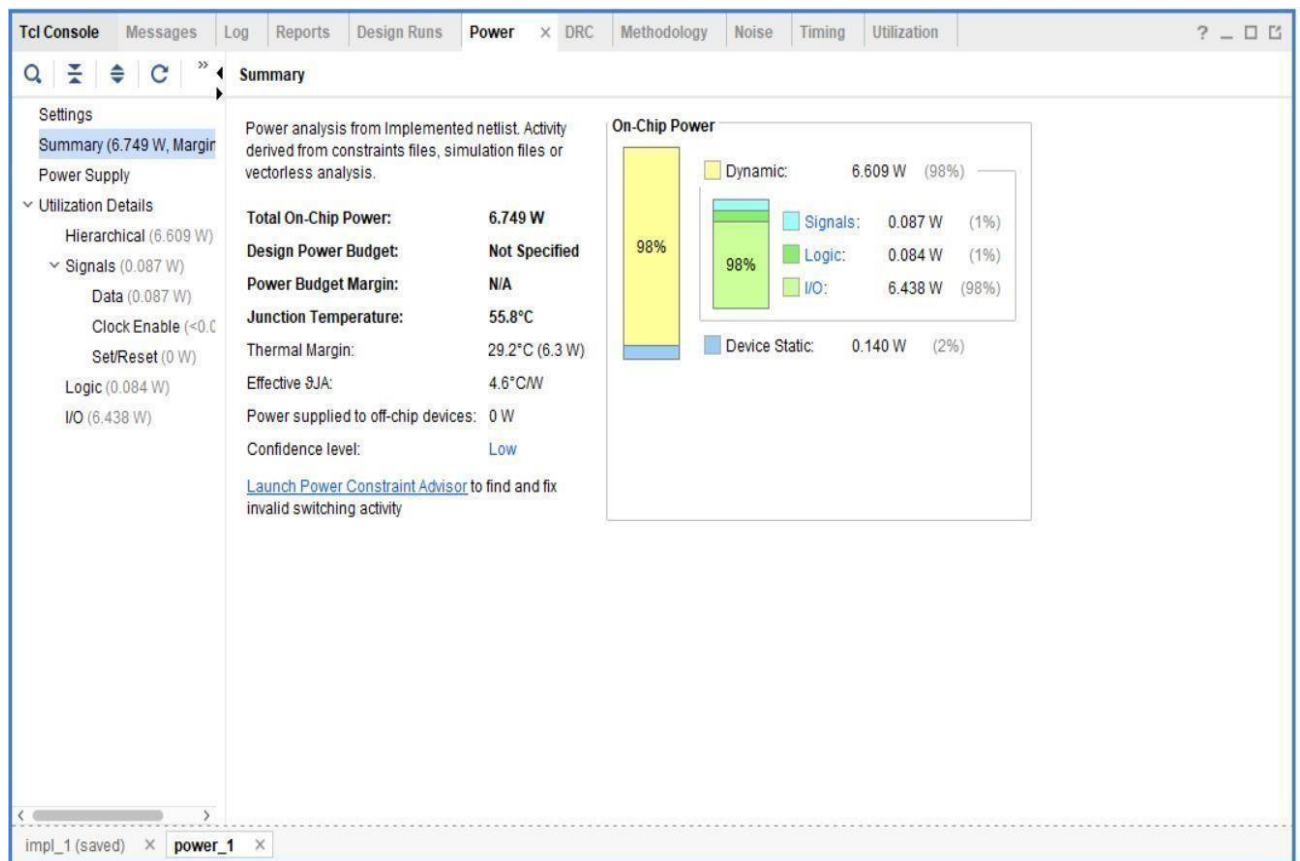
Tcl Console	Messages	Log	Reports	Design Runs	Power	DRC	Methodology	Timing
Design Timing Summary								
General Information								
Timer Settings								
Design Timing Summary								
Check Timing (33)								
Intra-Clock Paths								
Inter-Clock Paths								
Other Path Groups								
User Ignored Paths								
Unconstrained Paths								
Setup								
Worst Negative Slack (WNS): inf								
Total Negative Slack (TNS): 0.000 ns								
Number of Failing Endpoints: 0								
Total Number of Endpoints: 26								
Hold								
Worst Hold Slack (WHS): inf								
Total Hold Slack (THS): 0.000 ns								
Number of Failing Endpoints: 0								
Total Number of Endpoints: 26								
Pulse Width								
Worst Pulse Width Slack (WPWS): NA								
Total Pulse Width Negative Slack (TPWS): NA								
Number of Failing Endpoints: NA								
Total Number of Endpoints: NA								
There are no user specified timing constraints.								

Tcl Console	Messages	Log	Reports	Design Runs	Power	DRC	Methodology	Noise	Timing	
Summary	Messages (1)									
I/O Bank Details										
Links										
Name	Port	I/O Std	Vcco	Slew	Drive Strength (...)	Off-Chip Termina...	Remaining Margin ...	Notes		
I/O Bank 0 (0)										
I/O Bank 14 (9)		LVC MOS33	3.30	SLOW	12	FP_VTT_50				
V17	light_M2[0]	LVC MOS33	3.30	SLOW	12	FP_VTT_50	83.71			
R18	light_M2[1]	LVC MOS33	3.30	SLOW	12	FP_VTT_50	93.97			
N14	light_M2[2]	LVC MOS33	3.30	SLOW	12	FP_VTT_50	96.99			
T15	light_MT[0]	LVC MOS33	3.30	SLOW	12	FP_VTT_50	84.59			
V16	light_MT[1]	LVC MOS33	3.30	SLOW	12	FP_VTT_50	73.29			
U17	light_MT[2]	LVC MOS33	3.30	SLOW	12	FP_VTT_50	72.46			
V15	light_S[0]	LVC MOS33	3.30	SLOW	12	FP_VTT_50	73.62			
T16	light_S[1]	LVC MOS33	3.30	SLOW	12	FP_VTT_50	73.06			
U14	light_S[2]	LVC MOS33	3.30	SLOW	12	FP_VTT_50	90.26			
I/O Bank 15 (3)		LVC MOS33	3.30	SLOW	12	FP_VTT_50				
J13	light_M1[0]	LVC MOS33	3.30	SLOW	12	FP_VTT_50	88.79			
K15	light_M1[1]	LVC MOS33	3.30	SLOW	12	FP_VTT_50	91.60			
H17	light_M1[2]	LVC MOS33	3.30	SLOW	12	FP_VTT_50	91.35			
I/O Bank 16 (0)										
I/O Bank 34 (0)										
I/O Bank 35 (0)										

UTILIZATION REPORT



POWER REPORT



Github Link : <https://github.com/Jidnyesh-Chaudhari-2004/Traffic-LightController-System-Verilog.git>