INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 05 ISSUE: 09 | SEPT - 2021 ISSN: 2582-3930

SYNTHESIS OF TRAFFIC LIGHT CONTROLLER USING CADENCE

Archana M[1], Dr. Kiran V[2]
[1]*Dept. of Electronics and Communication Engineering, RV College of Engineering, Bangalore, Karnataka, India*
[2]*Professor, Dept. of Electronics and Communication Engineering, RV College of Engineering, Bangalore, Karnataka, India*

***Abstract*—Traffic lights are used to control the traffic and provide safety for the pedestrians on the road. This makes the pedestrians feel safe to walk on the roads. Traffic light controller manages the multipath ways and helps us to reduce the accidents. These systems can be implemented using Verilog HDL. This system is mainly used in junctions of the road. In this paper we adapt a traffic light controller using cadence, provided with synthesizable netlist. The Cadence uses genus synthesis tool which provides up to 5X faster synthesis which provides a detailed information about the architectural level of the design with detailed physical implementation. We realize a traffic light controller in high-level synthesis in genus cadence.**

Keywords—Traffic light, Verilog, genus, cadence.

I. INTRODUCTION

As the vehicles are increasing the traffic is increasing and it is important to control it [4]. Traffic light controller is used to monitor the junction of the roads and control the vehicles congestion. Traffic light controller directs the people to slow down and which will avoid road accidents[6]. The multipath way traffic light systems will play a major role as more accidents occur in the junctions. There are mainly three colors in the traffic lights: First, RED indicates stop for vehicles and pedestrians can cross the road. Second, Yellow indicates vehicles ready to move and for caution for pedestrians to stop. Third, Green indication for vehicles to move and pedestrians to stop. So, all the traffic lights will occur based on the timing fixed[5].

In this paper we realized a traffic light controller using Verilog and synthesized using genus tool. Verilog is Hardware Description Language(HDL) which is used for simulation, timing analysis, testing analysis and synthesis. Synthesis is a process of converting the register transfer level (RTL) code to gate level netlist and provide elaborated form of design. Simulation is done to check whether the design is meeting the provided constraints. Synthesis it translates the design and maps based on the RTL code. Genus tool reduces 20% iterations without affecting the performance. The input of synthesis is RTL code with timing constraints (SDC) file. Traffic light controller is implemented using Verilog HDL using different states. The implemented code is simulated in NC Launch simulator and the synthesized using genus tool which provides a detailed information regarding power, area based on timing constraints. This code can be implemented on FPGA which is more economical compared to other IOT systems. In this system we get detailed information regarding the traffic light control system regarding power, area and netlist generated. Due to massive growth of vehicles it is important to design the system accordingly, and it is equally important to know the working of the system with known power and area. Existing system designed and implemented using new ideas which helps to reduces the traffic violations. The following code is written in cadence software.

II. SYSTEM IMPLEMENTATION

The proposed system consists of 4 junction road respectively. The traffic controller system mainly consists of 3 lights (Red, Yellow and Green) in each road. In this red indicates stop, green indicates you can go, yellow indicates slow down[1]. These lights will keep switch after few seconds. In this paper, it starts from road1 as green primarily and switches from road2 to road4 and at last road3.

Fig 1. States of Traffic lights

Road1	Road2	Road3	Road4
Green	Red	Red	Red
Yellow	Red	Red	Red
Red	Green	Red	Red
Red	Red	Red	Green
Red	Red	Red	Yellow
Red	Red	Green	Red
Red	Red	Yellow	Red
Green	Red	Red	Red

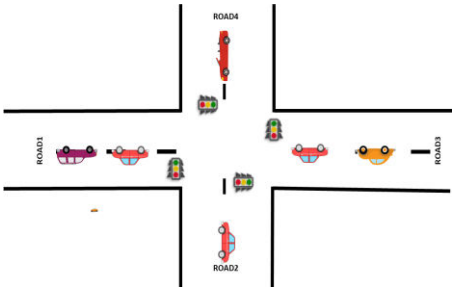

Fig 2. Structure of junction consisting of four roads

First green on road1 is predominantly declared and after 8 clock cycles it will switch to yellow and after 4 clock cycles it will switch to red , after this Road2 will switch to green for 8clock cycles and then switch to yellow for 4clock cycles then it will switch back to red and vice versa. The code for following specifications are written and executed in the following NC Launch software in cadence environment.

III. TOOLS USED

A. NC Launch software

NC Launch is an simulator in Cadence. This tool helps us to compile, elaborate and simulate the Verilog design. There are certain tools in the NC Launch:

- ncvlog it is used to compile the Verilog code.
- ncelab as the name suggests it is used to elaborate the design and provides a generated simulated snapshot.
- ncsim is used for simulation and the graph is obtained in the Simvision window[3].

The NC Launch will help us to confirm the design is meeting all the constraints.

B. Genus Synthesis Tool

It is physical innovation which is used provide a synthesized netlists. Certain attributes are set before beginning the process[2]. The HDL (code) path is set then the Synopsys Design Constraints(SDC) file is written which contains the delay and input output constraints. After SDC file is written and set, the netlist is generated and mapped using standard cell libraries.

IV. RESULTS

Simulation is a process of verifying the RTL code is meeting its functionality. Test bench is written with input port declared as reg and output port declared as wires where we provide different values to check the behavioral function of the design. Simulation is carried out by NC Launch tool in Cadence environment

Sl.No.	Outputs	Description
1	Road 1	RED=4, Yellow = 2, Green = 1
2	Road 2	RED=4, Yellow = 2, Green = 1
3	Road 3	RED=4, Yellow = 2, Green = 1
4	Road 4	RED=4, Yellow = 2, Green = 1

Fig.3 Traffic light description

When traffic light is green it displays 1, when it is yellow it displays 2 and when it is at red it displays 4. So, initially road1 is in green for 8clock cycles and other roads will be in red, after 8 clock cycles it will switch to yellow for 4 clock cycles and other road will still be in red. After 4 clock cycles the road1 will switch to red while road2 will switch to green for 8clock cycles whereas other roads will still be in red. So, after 8 clock cycles the road2 will switch to yellow for 4clock cycles and then red and the same process is carried out by for road3 and road4 and the simulated waveform and schematic are obtained for the same as shown in fig. 4 and fig. 5

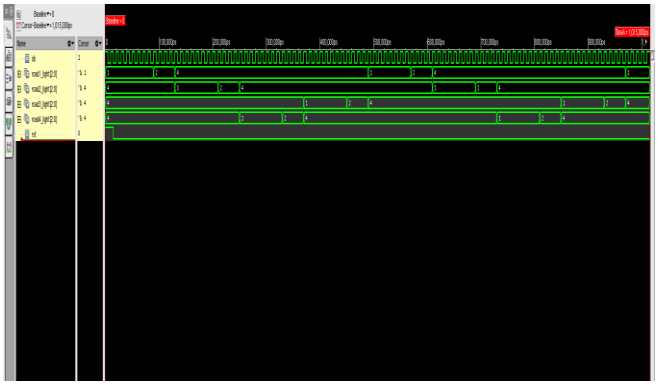

Fig 4. simulation waveform

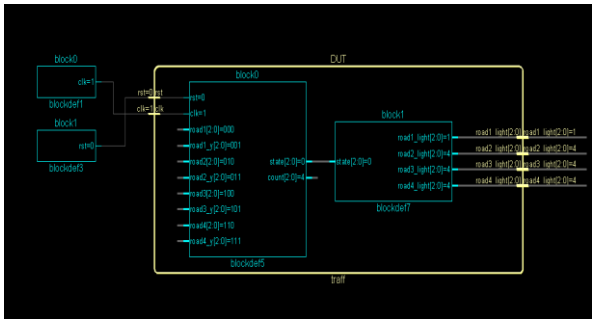

Fig 5. Simulation schematic

Synthesis is a process of generating and mapping a synthesized netlist to std library cells for the simulated design code and provide a waveforms with inserting delay and not inserting the delay constraints. It carried out by genus tool.

There are two steps in synthesis :

- Syngeneric: It is used to provide the RTL optimization for the design.
- Synmap: It's is used to provide logical optimization for the design. Logical optimisation means process of finding equivalent representation to simplify the design complexity. So, here we map the following circuit to std cell library to obtain the optimised design.

The design circuits for syn_generic and syn_map is shown in the fig. 6 and fig. 7. The synthesized netlist is obtained after this process.

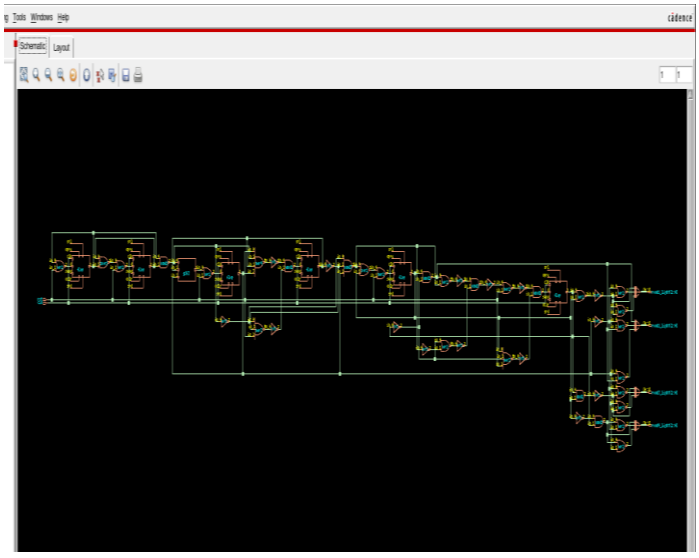

Fig 6. Synthesized generic schematic

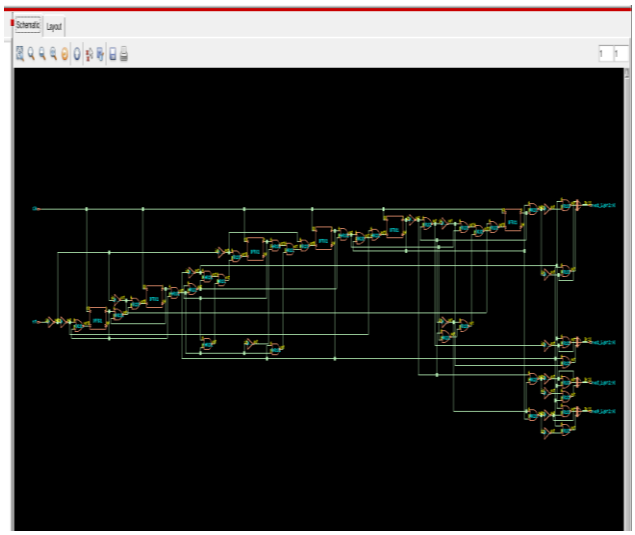

Fig 7. Synthesized mapping schematic

Power and timing report after optimizing the circuits are obtained as follows: The total cells used will be 55 and the total power consumed the 110uW and the timing constraints with setup and latency are given in the below fig.9.

```
legacy_genus:/> report_power
=============================================================
  Generated by:           Genus(TM) Synthesis Solution GENUS15.22 - 15.20-s024_1
  Generated on:           Sep 01 2021  09:58:00 pm
  Module:                 traff
  Technology library:     stdlib_CMOS 1.0
  Operating conditions:   PVT_1P8V_25C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
=============================================================

              Leakage    Dynamic     Total
Instance Cells Power(nW)  Power(nW)  Power(nW)
-------------------------------------------------
traff       55     2.094 110538.386 110540.480
```

Fig 8. Power report

```
legacy_genus:/> report_timing
Warning : Possible timing problems have been detected in this design. [TIM-11]
        : The design is 'traff'.
        : Use 'report timing -lint' for more information.
============================================================
  Generated by:           Genus(TM) Synthesis Solution GENUS15.22 - 15.20-s024_1
  Generated on:           Sep 01 2021  09:58:33 pm
  Module:                 traff
  Technology library:     stdlib_CMOS 1.0
  Operating conditions:   PVT_1P8V_25C (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

        Pin             Type      Fanout Load Slew Delay Arrival
                                          (fF) (ps)  (ps)   (ps)
------------------------------------------------------------------
(clock clk)             launch                              0 R
                        latency                          +300   300 R
(add.sdc_line_8_1_1)    ext delay                        +1000  1300 R
rst                     in port        1 14.0    0     +0    1300 R
g1283/in                                                 +0    1300
g1283/out               INV1X1         3 41.7   49    +47    1347 F
g1280/in                                                 +0    1347
g1280/out               INV1X1         2 42.6   82    +76    1423 R
g1266/A                                                  +0    1423
g1266/out               NOR2X1         1 25.7   62    +82    1505 F
count_reg[0]/D          DFFBX1                           +0    1505
count_reg[0]/CK         setup                         0 +114   1619 R
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(clock clk)             capture                            40000 R
                        latency                          +300  40300 R
------------------------------------------------------------------
Cost Group   : 'clk' (path_group 'clk')
Timing slack :   38681ps
Start-point  : rst
End-point    : count_reg[0]/D
```

Fig 9. Timing Report

After synthesizing the netlist is generated and that netlist is verified by simulating it again in the NC Launch to make sure the functionality is matching or not. The netlist is simulated by including delay constraints. The output of this simulation is provided in fig. 10 and fig. 11.

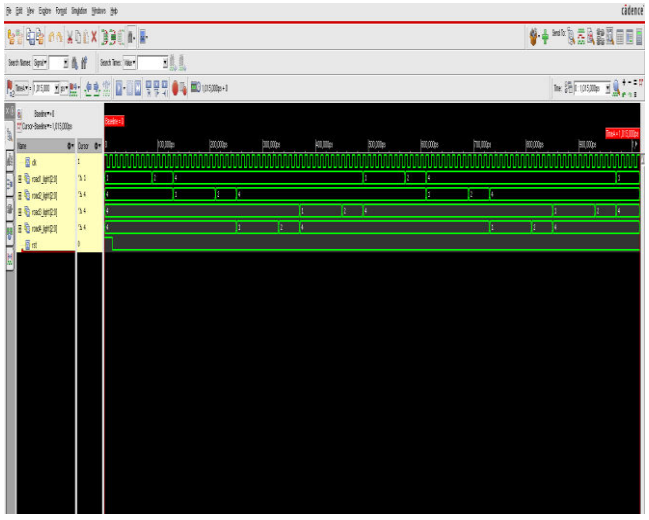

Fig 10. waveform after synthesis with no delay

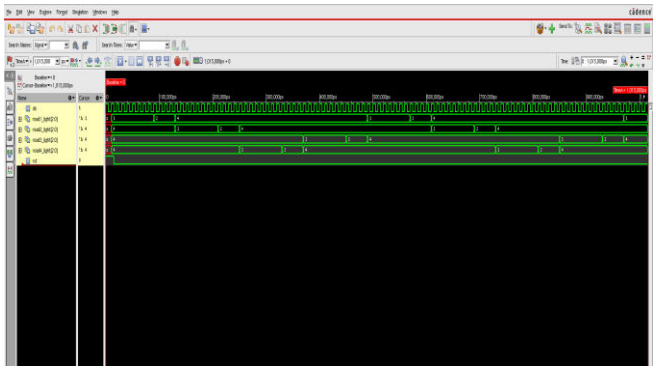

Fig 11. waveform after synthesis with delay

V. CONCLUSION

In this paper we designed traffic light controller using Verilog HDL and we checked the functionality using simulator and synthesized the design to get a efficient traffic light controller. This model is basically for the 4 junction roads where most of the accidents occurs and most of the traffic arises. It is usually used to avoid all types of accidents and make sure to run the traffic smoothly. So the gate-level netlist is generated and mapped to the standard library cells. The power and timing report is generated and to provide a detailed information about system and the waveforms are observed by giving delay and with no delay. It provides detailed information about architectural level and physical implementation level of the design.

ACKNOWLEDGEMENT

We are thankful to the college for providing the Cadence tools for executing the following results. This work is carried out under dept. of Electronics and Communication at RVCE.

REFERENCES

[1] S. V. Kishore, V. Sreeja, V. Gupta, V. Videesha, I. B. K. Raju and K. M. Rao, "FPGA based traffic light controller," 2017 International Conference on Trends in Electronics and Informatics (ICEI), 2017, pp. 469-475, doi: 10.1109/ICOEI.2017.8300971.

[2] Geralla, Leo & Guzman, Melvin & Hora, Jefferson. (2018). Optimization of Physically-Aware Synthesis for Digital Implementation Flow. International Journal of Engineering and Technology(UAE). 7. 31-35. 10.14419/ijet.v7i2.11.11002.

[3] Swetha, G. Sai Priyanka, Prathyusha, Mahesh Shetkar, " Design of Real-Time Traffic Control System using Verilog", International Journal of VLSI System Design and Communication Systems ISSN 2322-0929 Vol.05, Issue.04, April-2017, Pages:0343-0345.

[4] M. A. Kumaar, G. A. Kumar and S. M. Shyni, "Advanced traffic light control system using barrier gate and GSM," 2016 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC), 2016, pp. 291-294, doi: 10.1109/ICCPEIC.2016.7557213.

[5] A Arefín, Utsho & Hafizullah, M.R.M & Ahamed, Md. (2013). Implementation of Automatic Traffic Light Controller. 10.13140/2.1.4076.0640.

[6] B. K. Koay and M. M. Isa, "Traffic light system design on FPGA," 2009 IEEE Student Conference on Research and Development (SCOReD), 2009, pp. 269-271, doi: 10.1109/SCORED.2009.5443035.