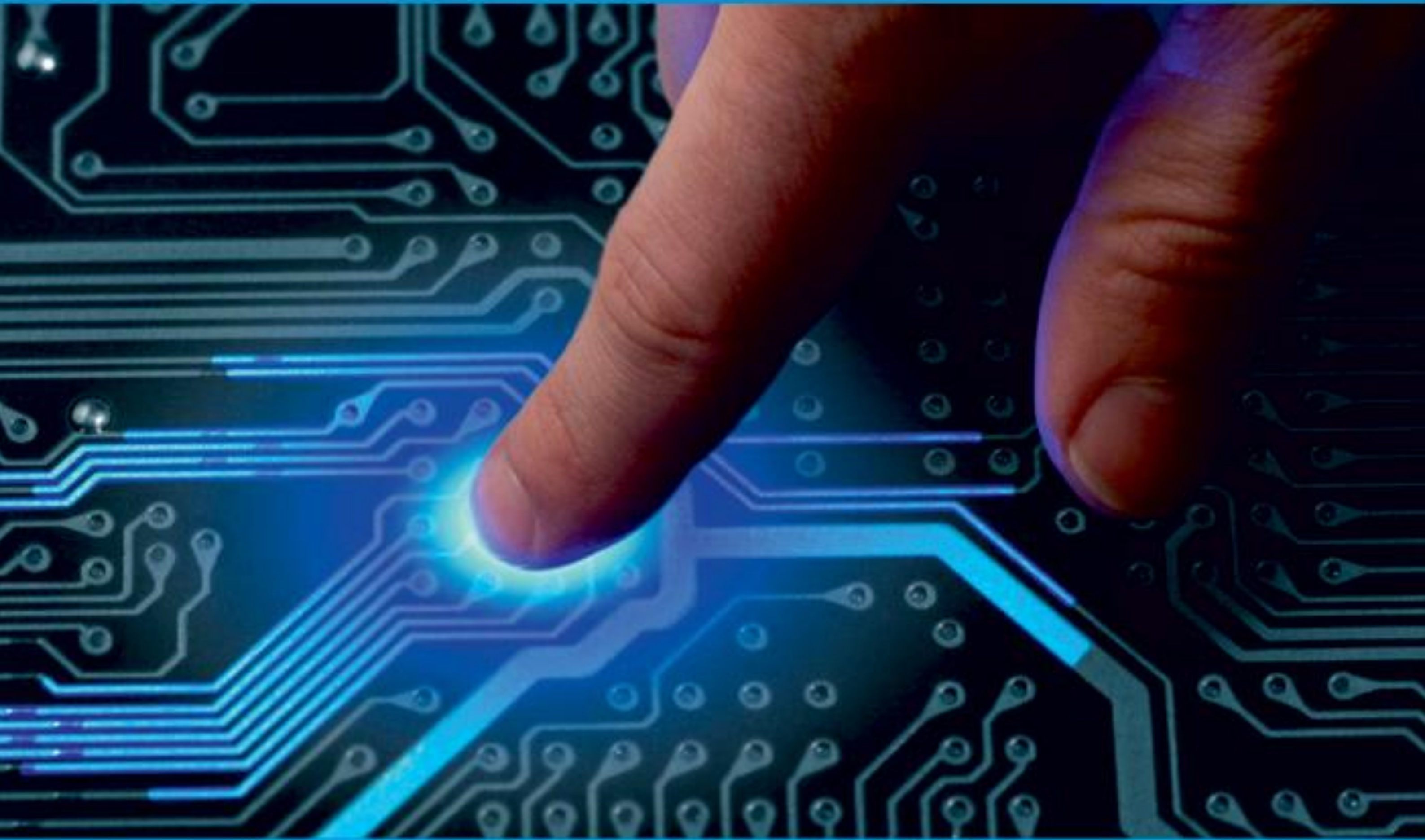
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 4, April 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204310 |

Development and Optimization of Traffic LightController Logic for 45nm Technology

Mr P.Surya Prakash Reddy[1], K.Akhil Kumar Reddy [2], A.Bhanuja[3], M .Anees[4], G.Anjana[5],G.Gandheeswar Reddy[6]

Assistant Professor, Department of ECE, GATES Institute of Technology, Gooty, India[1]

Student, Department of ECE, GATES Institute of Technology, Gooty, India [2,3,4,5,6]

ABSTRACT: In many cities, traffic regulation is a difficult challenge to solve. This is due to the large number of cars and the traffic system's high dynamics. Poor traffic network are a major cause of accidents and lost time. Vehicles will spend a minimum waiting period at traffic lights if this methodology is used. Verilog Hardware Description Language (HDL) programming is used to create the hardware design. Verilog is a hardware description language that deals with hardware design and simulation, as the name indicates. Mounting the numerous electronic components on a breadboard or PCB circuit becomes quite complex. It also takes an excessive amount of time to simulate, and various faults might arise due to poor component connections on the circuit. As a result, hardware descriptive language concludes to overcome this obstacle. The fundamental design of a T-shaped road for traffic light regulation is the subject of this project. Mentor Graphics, Questasim and Oasys has been used to test the system's output.

KEYWORDS: VLSI, Mentor Graphics, Questasim, Oasys, 45nm Technology, Traffic Systems, Traffic Light Controller, Verilog

I. INTRODUCTION

Advances in the history of CPU production may be traced using CMOS technology, as well as the design of the processor itself. Reducing the dimensions of the small transistors that build up a CPU has many implications because it can pack more into the same space. At the most basic level, it wasnot even possible to create today's processor designs using process technology just a few years ago. Intel's Core 2 Extreme QX9650 has a capacity of about 800 million, which is about 3,000 times thatnumber [1]. The smaller the transistor, the less wattage it consumes in the cycle. This means that you can actually get more power from the transistors than you would with large process technologies [2]. Combining enough transistors with the 386 QX9650 consumes about 3000W, buta complete Core 2 Extreme QX9650 PC further elements require just over 200W at full load [3]. Low power utilization has another practical side effect. If the transistor consumes less power, it willnot get too hot. Therefore you may set them to a higher phase without over burdening them up or overloading the motherboard power circuits that power them [4]. There are other aspects to consider, but the new technologies nearly every time mean higher clock frequency caps. Keeping up the basic CPU design brings the benefits of the last, but not a few, smaller transistors [5]. When this happens, the processor itself becomes smaller and is called a "shrink". Since the manufacturing system uses standardized semiconductor wafers, 300mm is currently the maximum and can accommodate more wafers. Since the wafer fabrication cost itself is the same, the fabrication cost of each processor is low [6]. A 45nm, for instance, takes up half the space of a 65nm processor of thesame architecture [7].Therefore, shifting to 45nm processor will cut manufacturing costs in half, but you also need to consider the cost of implementing a new process and building an industrial unit to do it. Perhaps this is quite pricy [8]. Major advantages of 45nm technology is that it always seems better for semiconductors to be smaller, and I wonder why such miniaturization doesn't happen faster [10].However, there are always issues that must be overcome to reduce the transistor size. These include parasitic capacitance, current leakage, and latch-up that hold the charge when some of the miniature integrated circuits should not hold the charge. The latter two are especially problematic in recent process cuts [11]. This is because the distances among the Narrow lines is extremely small that it becomes increasingly difficult to keep current from flowing in unexpected places. AMD and IBM are using Silicon-on-Insulator (SOI) technology to counteract this and allow it to be downgraded to 65nm. However, with Intel's move from 65nm to 45nm, the industry is still using it. Legacy bulk CMOS technology, with the addition of high K dielectric and metal gate technology.

Traditionally, silicon dioxide has been used as a dielectric for small transistors, but today's manufacturing scale is prone to leaks. High dielectric constant (High K) alternatives block this. Metal gates, on the other hand, take elements of processor that should be conductive in the other way. Previously, low-conductivity polysilicon was used in circuits because it was unchallenging to manufacture

II. METHODOLOGY

We may program the process in Verilog and then mount it on a circuit or just upload it to the circuit to make it operate according to the code we wrote. HDL is commonly used for sequential circuits such as shift registers and combinational logic circuits such as adders and subtractors. It essentially defines digital systems such as a microprocessor or a memory. Whatever design is described in HDL is independent, it has its state of work, is more easier to model, design, and debug than schematics, and is far more helpful for big circuits., thus overcoming difficulties or problems in manually designing circuits with breadboard and PCB, the adoption of Verilog design in today's complicated environment is skyrocketing. The fundamental design of a T- shaped road for traffic light regulation is the subject of this project. Mentor Graphics, Questasim and Oasys has been used to test the system's output. In many cities, traffic regulation is a difficult challenge to solve.

This is due to a big number of cars and the traffic system's high dynamics. Poor traffic systems are a major cause of accidents and lost time. Vehicles will spend a minimum waiting period at traffic lights if this methodology is used. It also takes an excessive amount of time to simulate, and variousfaults might arise due to poor component connections on the circuit.

As a result, hardware descriptive language concludes to overcome this obstacle. We may program the process in Verilog and then mount it on a circuit or just upload it to the circuit to make it operateaccording to the code we wrote.

HDL is commonly used for sequential circuits such as shift registers and combinational logic circuits such as adders and subtractors. It essentially defines digital systems such as a microprocessor or a memory, thus overcoming difficulties or problems in manually designing circuits with breadboard and PCB, the adoption of Verilog design in today's complicatedenvironment is skyrocketing

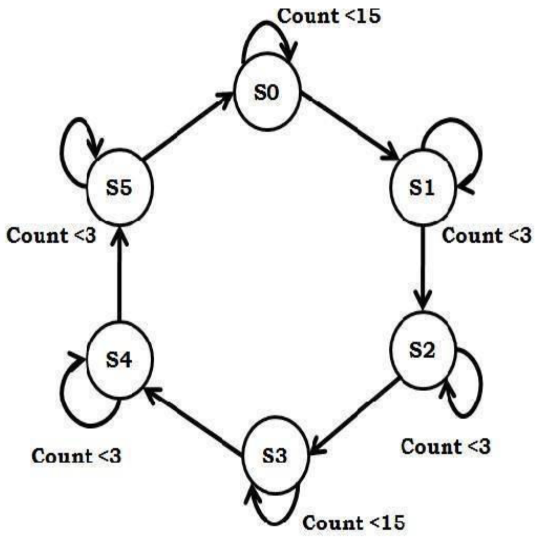


Fig.1. State Diagram

- Green light indicates that there is no traffic and there is easy flow of vehicles in thatroute/direction.
- x Red light indicates that there is a traffic jam and that route is blocked for the vehiclesto move and,
- Yellow light indicates that the route has medium flow of vehicles

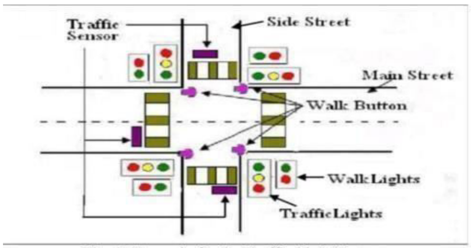
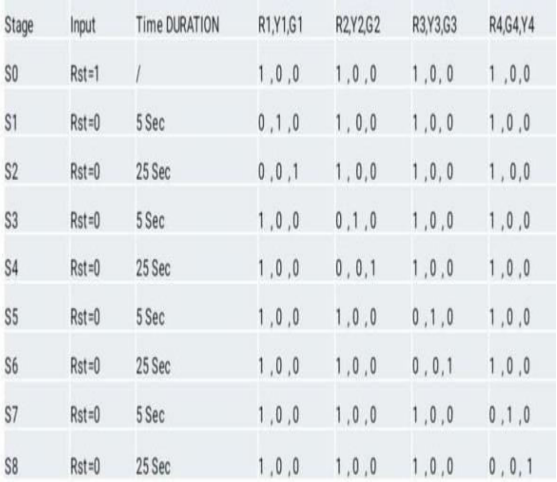


Fig. 1: Example for the Traffic Light System.

- R = RED,
- Y = YELLOW
- G = GREEN.



Stage	Input	Time DURATION	R1,Y1,G1	R2,Y2,G2	R3,Y3,G3	R4,G4,Y4
S0	Rst=1	/	1 ,0 ,0	1 ,0 ,0	1 ,0 , 0	1 ,0 ,0
S1	Rst=0	5 Sec	0 ,1 ,0	1 , 0 ,0	1 ,0 , 0	1 , 0 ,0
S2	Rst=0	25 Sec	0 ,0 ,1	1 , 0 ,0	1 ,0 , 0	1 , 0 ,0
S3	Rst=0	5 Sec	1 , 0 ,0	0 ,1 ,0	1 ,0 ,0	1 , 0 ,0
S4	Rst=0	25 Sec	1 , 0 ,0	0 , 0 ,1	1 ,0 ,0	1 , 0 ,0
S5	Rst=0	5 Sec	1 , 0 ,0	1 , 0 ,0	0 ,1 ,0	1 , 0 ,0
S6	Rst=0	25 Sec	1 , 0 ,0	1 , 0 ,0	0 , 0 ,1	1 , 0 ,0
S7	Rst=0	5 Sec	1 , 0 ,0	1 , 0 ,0	1 ,0 ,0	0 ,1 ,0
S8	Rst=0	25 Sec	1 , 0 ,0	1 , 0 ,0	1 ,0 ,0	0 , 0 , 1

State Table

It illustrates the structure of the selected traffic light model for four road intersections (one Main Street and three side streets). In general, Traffic Light Controller System consists of three lights (red, green and yellow) in each direction .The red light indicates to Stop, green light indicates to allow the traffic and yellow light indicates the caution that the traffic is going to be stopped in few seconds. While, turning in yellow and red lights at the same time indicates the caution that the traffic is going to be moving in few seconds. The intersection is fitted with a sensor for side street traffic and with walk request button. This traffic light controller also has provision for walk light (which consists of two lights red and green, where, green light allows the walkers to pass the street while red light avoids the walkers from passing the street) and for the traffic sensors in each one of the side streets. A simple block diagram of the traffic light controller system is exposed in Figure 2. The design is composed of finite state machine (FSM), data storage (D_RAM), timer, divider, and various synchronizers (latch,and synchronizer). 3.1. Finite state machines (FSM) Finite State Machines (FSM) is the heart of the traffic light controller system. This FSM controls the loading of static data storage locations with timing parameters, displaying these parameters by reading RAM locations, and the control of the actual traffic lights. There are four timing parameters in this system as displayed in Table 1. They are the base interval (TBASE) for side green, an extended interval for main green and walk green light (TEXT), the time for yellow light (TYEL), and a blink interval (TBLINK). The user can specify the four timing parameters using two switches (L0, L1) manually. The FSM can execute four functions specified by two functions switches (F0, F1). These functions are listed in Table 2, where, the user can execute one of four possible functions: writing new timing parameters, reading old timing parameters, running traffic light in normal mode, and running the traffic light in blinking mode as obtained in Figure 3a. Besides, the idle state of the FSM is called the reset state, in this state, the lights are turned off and the system does not do anything. The system will stay in the reset state until the GO button is pressed.

Using the writing function, the user can specify the any one of the four timing parameters as shown in Table 1 using (L1, L0) switches, the value of the parameter is set using the (C4_C0) switches. For the reading operation, the user can use the same L1 and L0 switches to denote whichof the four timing parameters to view on a set Hex-LEDs. In normal mode or blinking mode, the system just cycles through the various traffic light states. The regular controller has been designedwith nine states as presented in Table 3 without taking the traffic sensors and walk request in the point view.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

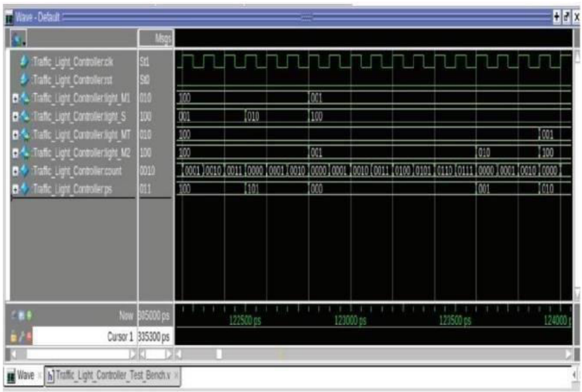
|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204310 |

In the normal mode that is displayed in Figure 3b, the side street has a shorter green interval than the main street, but if there is traffic on the side street when the controller is about to cycle to turn that green light off, it will extend the green light by the shorter (side street) green interval. Thus the green light on the side street will stay on until traffic on the side street clears. Traffic sensor switch is used to simulate the effect waiting traffic on the side street, the system complies by keeping side street green until the traffic sensor is switch off. The walk light comes onafter the main street yellow interval, and then only if the walk request button has been pushed. Late at night or when something in the system is not working, the light goes into the blinking mode this involves the lights blinking on and off, alternating between main yellow side red, and main red side yellow.

III. SIMULATION RESULTS

Simulation can be defined as a process of verifying the functionality of the digital design and is a technique for applying different input stimulus to the design at various clock times to check if the RTL code behaves in the expected way or not. To verify the behavior we require a test bench,for which the waveform obtained Test bench is a Verilog module and is a program or a code in which the inputs and outputs are declared as reg and wire respectively. In this, some ideal predefined values are given in order to verify the behavior of the design if it is working as expected or not.


Wave - Default
Msgs
Traffic_Light_Controller.clk St1
Traffic_Light_Controller.rst St0
Traffic_Light_Controller.light_M1 010
Traffic_Light_Controller.light_S 100
Traffic_Light_Controller.light_MT 010
Traffic_Light_Controller.light_M2 100
Traffic_Light_Controller.count 0010
Traffic_Light_Controller.ps 011
100 001
001 010 100
100 001
100 001 010 100
0001 0010 0011 0000 0001 0010 0000 0001 0010 0011 0100 0101 0110 0111 0000 0001 0010 0000
100 101 000 001 010
Now 305000 ps
Cursor 1 335300 ps
122500 ps
123000 ps
123500 ps
124000 ps
Wave
Traffic_Light_Controller_Test_Bench.v

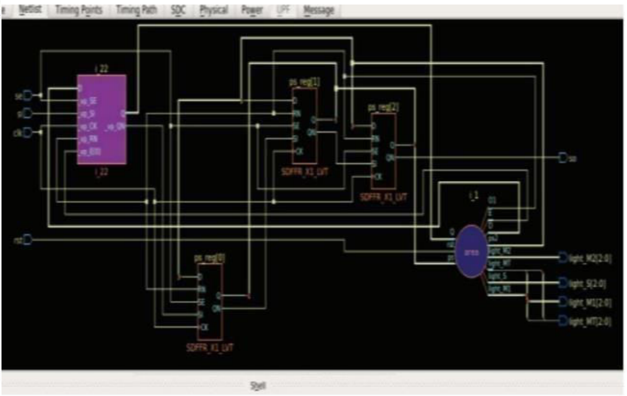


Fig. 5. Simulation Waveform

Whenever the clk is set to “1” and the rst is set to “0” we obtain the intended output waveform asexpected and designed in our test bench and RTL Code. Here, according toour problem statement and state table, when the ps = 011 we obtain M1 = 010, S = 100, MT = 010 and M2 = 100 for count = 0010. Similarly, for other states ranging from 000 to 111 we obtain expected M1, S, MT and M2 values respectively. Indicating the flow of traffic whether it is heavy, medium or easy flow of traffic with Red, Yellow and Green (R, Y, G) colors. For example, if M1 = 010 then the flow of traffic is medium as Yellow color is set to high, i.e., “1”. Similarly, for the other directions S, MT, M2 the flow of traffic is analyzed.

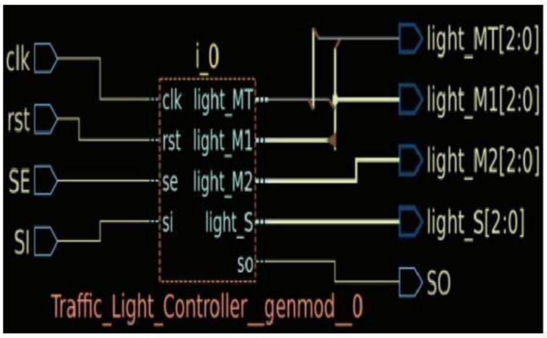
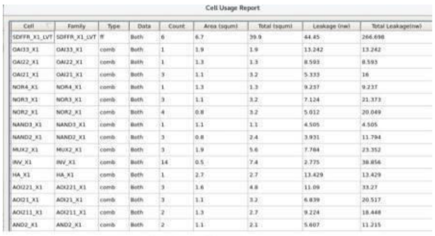


Fig. 7. Block diagram of Real-time Traffic Light Controller after synthesis.

The area, registers are shown in Fig 6. and Fig 7. above. The outputs are 3-Bit and the inputs are given as rst and clk to the area. The internal registers are shown in Fig 6.depicting the flow of the design terms of nW. For instance ps_reg[1] the total power obtained is 54nW, leakage power = 44nW, switching power = 6nW and internal power = 2nW. And similarly, for other instances the power report The Area report obtained after synthesis including the top instance module has 51 cells and the cell area includes 86squm. The Hierarchy report for the module Traffic_Light_Controller with 51 cells and total area 86squm has sequential area = 40squm and combinational area = 46squm.The RTL Partition report for Traffic_Light_Controller_genmod_0, concludes that for 51instances the area obtained after synthesis is 85.918squm.

IV. ANALYSIS REPORT

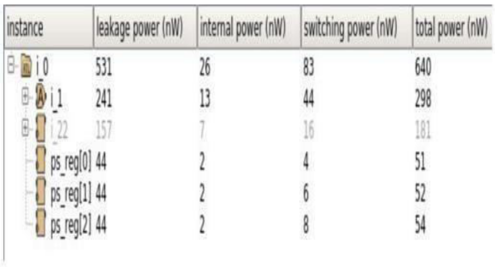


Cell Usage Report

Cell	Family	Type	Data	Count	Area (squm)	Total (squm)	Leakage (nw)	Total Leakage(nw)
SDFFR_X1_LVT	SDFFR_X1_LVT	ff	Both	6	6.7	39.9	44.45	266.698
OAI33_X1	OAI33_X1	comb	Both	1	1.9	1.9	13.242	13.242
OAI22_X1	OAI22_X1	comb	Both	1	1.3	1.3	8.593	8.593
OAI21_X1	OAI21_X1	comb	Both	3	1.1	3.2	5.333	16
NOR4_X1	NOR4_X1	comb	Both	1	1.3	1.3	9.237	9.237
NOR3_X1	NOR3_X1	comb	Both	3	1.1	3.2	7.124	21.373
NOR2_X1	NOR2_X1	comb	Both	4	0.8	3.2	5.012	20.049
NAND3_X1	NAND3_X1	comb	Both	1	1.1	1.1	4.505	4.505
NAND2_X1	NAND2_X1	comb	Both	3	0.8	2.4	3.931	11.794
MUX2_X1	MUX2_X1	comb	Both	3	1.9	5.6	7.784	23.352
INV_X1	INV_X1	comb	Both	14	0.5	7.4	2.775	38.856
HA_X1	HA_X1	comb	Both	1	2.7	2.7	13.429	13.429
AOI221_X1	AOI221_X1	comb	Both	3	1.6	4.8	11.09	33.27
AOI21_X1	AOI21_X1	comb	Both	3	1.1	3.2	6.839	20.517
AOI211_X1	AOI211_X1	comb	Both	2	1.3	2.7	9.224	18.448
AND2_X1	AND2_X1	comb	Both	2	1.1	2.1	5.607	11.215

Fig. 8. Cell usage report

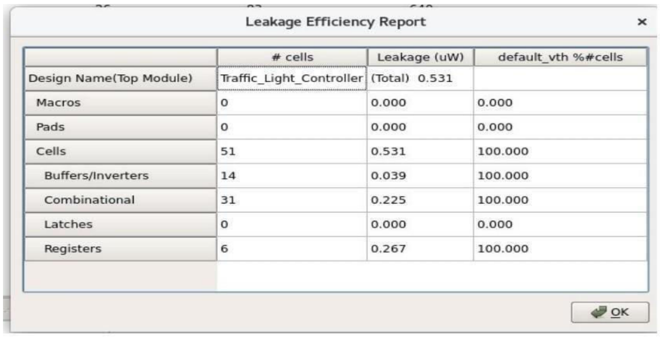
Fig 8. shows the cell usage report after the synthesis where the Total leakage power is given in terms of nw and the area in terms of squm. For a AND2_X1 cell, the area obtained is 1.1squm, Total area = 2.1squm, Leakage = 5.607nw and Total Leakage = 11.215nw. And, similarly for the other cells such as the NOR2_X1, HA_X1, etc. the obtained values are as shown in the Fig 8.



instance	leakage power (nW)	internal power (nW)	switching power (nW)	total power (nW)
i 0	531	26	83	640
i 1	241	13	44	298
i 22	157	7	16	181
ps_reg[0]	44	2	4	51
ps_reg[1]	44	2	6	52
ps_reg[2]	44	2	8	54

Fig. 9. Power Report for instance

In the Fig 9. The total power, switching power, leakage power and internal power obtained is in

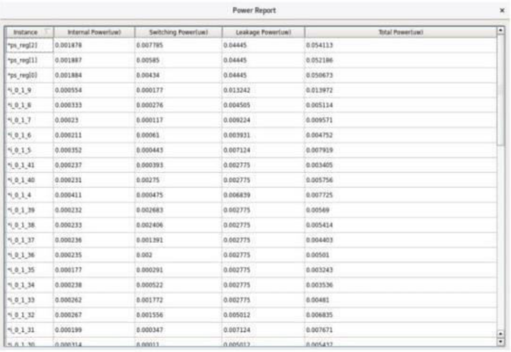


Leakage Efficiency Report

	# cells	Leakage (uW)	default_vth %#cells
Design Name(Top Module)	Traffic_Light_Controller	(Total) 0.531	
Macros	0	0.000	0.000
Pads	0	0.000	0.000
Cells	51	0.531	100.000
Buffers/Inverters	14	0.039	100.000
Combinational	31	0.225	100.000
Latches	0	0.000	0.000
Registers	6	0.267	100.000

Fig. 10. Leakage Efficiency Report

The Leakage efficiency report shown in Fig 10. For the Top Modules/Designs for Macros, Pads, Cells, Buffers, Combinational, Latches and Registers states the obtained number of cells, leakage efficiency in uW and the default vth percentage after synthesis. It has 100% default_vth cells for the Top Modules Cells, Buffers/Inverters, Combinational and Registers. The Scan Chain Report for scanChain_1 with Index 1, has Scan Instance = 6, Length = 6 with a Test Clock as "clk" and Clock Edgeas "rise" and finally zero Lockup. The Clock report after synthesis obtained concludes a period of 1400ns with Ports/Pins as "clk. Fig 11. shows the overall final Power Report for all the Instances after synthesis. The total Power consumption is very much less for a 45nm when compared to any other higher technology nodes than 45nm, such as 90nm or 180nm.



Power Report

Instance	Internal Power(uw)	Switching Power(uw)	Leakage Power(uw)	Total Power(uw)
*ps_reg[2]	0.001878	0.007785	0.04445	0.054113
*ps_reg[1]	0.001887	0.00585	0.04445	0.052186
*ps_reg[0]	0.001884	0.00434	0.04445	0.050673
*i_0_1_9	0.000554	0.000177	0.013242	0.013972
*i_0_1_8	0.000333	0.000276	0.004505	0.005114
*i_0_1_7	0.00023	0.000117	0.009224	0.009571
*i_0_1_6	0.000211	0.00061	0.003931	0.004752
*i_0_1_5	0.000352	0.000443	0.007124	0.007919
*i_0_1_41	0.000237	0.000393	0.002775	0.003405
*i_0_1_40	0.000231	0.00275	0.002775	0.005756
*i_0_1_4	0.000411	0.000475	0.006839	0.007725
*i_0_1_39	0.000232	0.002683	0.002775	0.00569
*i_0_1_38	0.000233	0.002406	0.002775	0.005414
*i_0_1_37	0.000236	0.001391	0.002775	0.004403
*i_0_1_36	0.000235	0.002	0.002775	0.00501
*i_0_1_35	0.000177	0.000291	0.002775	0.003243
*i_0_1_34	0.000238	0.000522	0.002775	0.003536
*i_0_1_33	0.000262	0.001772	0.002775	0.00481
*i_0_1_32	0.000267	0.001556	0.005012	0.006835
*i_0_1_31	0.000199	0.000347	0.007124	0.007671
*i_0_1_30	0.000314	0.00011	0.005012	0.005437

Fig. 11. Overall Power Report

V. CONCLUSION

The latest solutions of various-ways of traffic control vastly enhance the traffic situation across the globe. The study's key characteristic is to enhance the flow of traffic control at T-junction. The synthesis and simulation of Traffic Light Controller using 45nm technology has been successfully implemented and verified using industry specific standard licensed tools, such as Mentor Graphic, Questasim and Oasys. It is found that using this 45nm technology in the Traffic Light Controller we can concludethat it is efficient than 180nm and 90nm respectively. The future work would be to implement the same on FPGA Board.

REFERENCES

1. Prashant Kumar Singh and P. Daniel, "Advanced real Traffic Light Controller system design using Cortex-M0 ip on FPGA," 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, 2014, pp. 1023-1026, doi: 10.1109/ICACCCT.2014.7019251.
2. Taehee Han and Chiho Lin, "Design of an intelligence traffic light controller (ITLC) with VHDL," 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering. TENCOM '02. Proceedings., 2002, pp. 1749-1752 vol.3, doi:10.1109/TENCON.2002.1182673.
3. W. El-Medany and M. Hussain, "FPGA-Based Advanced Real Traffic Light Controller System Design," 2007 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2007, pp. 100-105, doi: 10.1109/IDAACS.2007.4488383.
4. S. Singh and S. C. Badwaik, "Design and implementation of FPGAbased adaptive dynamic traffic light controller," 2011 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), 2011, pp. 324-330, doi: 10.1109/ETNCC.2011.6255914.
5. M. F. M. Sabri, M. H. Husin, W. A. W. Z. Abidin, K. M. Tay and H. M. Basri, "Design of FPGA-based Traffic Light Controller System," 2011 IEEE International Conference on Computer Science and Automation Engineering, 2011, pp. 114-118, doi: 10.1109/CSAE.2011.5952814.
6. Girish H, Shashikumar D R, “A Novel Optimization Framework for Controlling Stabilization Issue in Design Principle of FinFET based SRAM”, International Journal of Electrical and Computer Engineering (IJECE) Vol. 9, No. 5 October 2019, pp. 4027~4034. ISSN: 2088-8708, DOI: 10.11591/ijece.v9i5.pp.4027-4034 [7]
7. Girish H, Shashikumar D R, “PAOD: a predictive approach for optimization of design in FinFET/SRAM”, International Journal of Electrical and Computer Engineering (IJECE) Vol. 9, No. 2, April 2019, pp. 960~966. ISSN: 2088-8708, DOI: 0.11591/ijece.v9i2.pp.960-966
8. Girish H, Shashikumar D R, “SOPA: Search Optimization Based Predictive Approach for Design Optimization in FinFET/SRAM”, © Springer International Publishing AG, part of Springer Nature 2019 Silhavy (Ed.): CSOC 2018, AISC 764, pp. 21–29, 2019. https://doi.org/10.1007/978-3-319-91189-2_3.
9. Girish H, Shashikumar D R, “Cost-Effective Computational Modelling of Fault Tolerant Optimization of FinFET-based SRAM Cells”, © Springer International Publishing AG 2017 R. Silhavy et al. (eds.), Cybernetics and Mathematics Applications in Intelligent Systems, Advances in Intelligent Systems andComputing 574, DOI 10.1007/978-3-319-57264-2_1.[10]
10. Girish H, Shashikumar D R, “A Survey on the Performance Analysis of FinFET SRAM Cells for Different Technologies”, International Journal of Engineering and Advanced Technology (IJEAT)ISSN: 2249 – 8958, Volume-4 Issue-6, 2016. [11]
11. K. S. Reddy and B. B. Shabarinath, "Timing and Synchronization for Explicit FSM Based Traffic Light Controller," 2017 IEEE 7th International Advance Computing Conference (IACC), 2017,pp. 526-529, doi: 10.1109/IACC.2017.0114.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details