Interconnect Optimization For Dynamic Power Reduction

Submitted in partial fulfillment of the

requirements for the degree of

Master of Technology

in

Electronics and Communication Engineering

(VLSI Design)

Submitted By

Foram Kothari

18MECV04

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2020

Certificate

This is to certify that the M.Tech thesis entitled **”Interconnect Optimization For Dynamic Power Reduction”** submitted by **Foram Kothari (Roll No: 18MECV04)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Electronics and Communication (VLSI Design) of Nirma University, Ahmedabad, is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this M.Tech thesis, to the best of my knowledge, haven’t been submitted to any other university or institution for award of any degree or diploma.

Dr. Niranjan Devashrayee
Internal Project Guide
Institute of Technology,
Nirma University, Ahmedabad

Dr. Usha Mehta
PG Coordinator M.Tech - VLSI
Institute of Technology,
Nirma University, Ahmedabad

Dr. Dhaval Pujara
Head Of Department
Institute of Technology,
Nirma University, Ahmedabad

Director
Institute of Technology,
Nirma University, Ahmedabad

EXTERNAL CERTIFICATE

This is to certify that the Major Project Report entitled **"Interconnect Optimization For Dynamic Power Reduction"** submitted by **Foram Kothari** (Roll No. **18MECV04**) as the partial fulfillment of the requirements for the award of the degree of Master of Technology in VLSI Design, Electronics & Communication Engineering, Institute of Technology, Nirma University is the record of work carried out by her under my supervision and guidance. The work submitted in our opinion has reached a level required for being accepted for the examination.

Date: 08/05/2020
Place: Bangalore

Poonam Srinivas
Manager
poonam.srinivas@intel.com
Intel Technology India Pvt Ltd
Bangalore

Statement of Originality

I, **Foram Kothari**, Roll No: **18MECV04**, give undertaking that the M.Tech thesis entitled "**Interconnect Optimization For Dynamic Power Reduction**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Electronics & Communication Engineering (VLSI Design)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student
Date:
Place: Ahmedabad

Endorsed by
Dr. N. M. Devashrayee
(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Dr. N. M. Devashrayee**, Electronics and Communication Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come and also for his guidance throughout whole M.Tech.

It gives me an great pleasure to thank **Dr. Usha Mehta**, PG Coordinator of VLSI Design, Electronics and Communication Department, Institute of Technology, Nirma University, Ahmedabad for her kind support and guidance throughout whole M.Tech.

It gives me an immense pleasure to thank **Dr. Dhaval Pujara**, Hon'ble Head of Electronics and Communication Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

I would like to thank **Mrs. Poonam Srinivas** (Manager) and **Mr. Karan B Patel** (Physical Design Engineer) at Intel Technology India Private Limited, Bangalore, for their suggestions and the clarity of the concepts on the topic that helped me a lot in this work. Also I am thankful to my team mates for their valuable guidance and prompt help.

I would also thank the Institution, all faculty members of Electronics and Communication Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

Foram Kothari
18MECV04

Abstract

Modern ASIC designs having ultra-deep sub-micron processes and for this type of technology node layout had very important role to play. Full chip layout had its own hierarchical structure. The full chip hierarchy contains Standard cell, Functional unit blocks (fub) and Section(partition). Fub Integration Flow is useful to integrate the fub inside section. It basically loads all the fub data inside section, then perform hook power on it and check for the opens, DRC violations, design for manufacturability, etc. Sections have many fub to integrate in cyclic manner and have many DRCs violation over the fubs. To clean all the DRC violations manually is time consuming process.

Routing has become increasingly challenging with each technology node due to the ever-increasing number of metal layers and distinct layer thicknesses, design rule volume, and design complexity. In this context, the problem of congestion analysis or prediction also increases in significance, since designers need to be able to predict during early floor planning and timing closure iterations whether their design will route. Further, it is not enough just to route cleanly; the designer requires that the routing be completed with minimal disruption, to avoid massive post-routing timing and power degradation and also electrical fix ups.

This thesis covers the methods to reduce the dynamic power of the power hungry nets by reducing the dynamic capacitance (Cdyn). After that, an automation flow is developed for these power reduction techniques by using TCL scripting so that the overall human efforts can be reduced.

By these methods, I got an improvement for around 5 % nets of the project thereby giving me a considerable power optimization overall. And also a good amount of human efforts and time are saved by using the automation flow on each partition.

Contents

List of Figures

Chapter 1

Introduction

Back-end design style have sections with sea of fubs, as against big partition with the sea of cells. Full chip architecture is divided into 8-10 sections. Each section has around 100+ fubs. To integrate the fubs with all the DRCs clean-up is quite a big challenge for all the section layout owners. Time required for the integration of fub and to clean DRC for each and every fub is time consuming process. Lots of fubs are multi-instantiated, so to clean DRCs one by one is much more time consuming process. Complexity of today's technology node and to meet the time to market, which is followed up by the customer requirement - algorithm should be developed, which can claim to reach the TTM on time without putting manual layout effort in cleaning up the DRCs.

With today's newer fabrication technology improvements following Moore's law, integrated circuits (ICs) are continually shrinking, creating many new challenges for the place and route flow. We face a lot of issues in converging during the place and route flow. Each flow stage impacts downstream steps in the flow,making later stages like routing and post-route more challenging, it is hard to successfully route a design during the routing stage, and it has become even harder to reroute after disturbing it during the post-route stage. That's why we have to control and minimize the optimization disturbance.

With the dramatic increases in on-chip packing densities, routing congestion has become a major problem in chip design. The problem is especially acute as interconnects are also the performance bottleneck in integrated circuits. The solution lies in judicious resource management. This involves intelligent allocation of the available interconnect resources, up-front planning of the wire routes for even wire distributions, and transformations that make the physical synthesis flow congestion-aware.

A few years back, major challenges for semiconductor industry were area and performance, they were increasing speed and reducing the area. With increase in these demands, the need of power performance came into picture. And placement and routing plays a major role in dynamic power reduction. So, along with the timing improvement and routing optimization, power also plays very important role in place and route.

The main objective of this project is to reduce the dynamic power of the power hungry nets by reducing the dynamic capacitance (Cdyn). Here I am going to describe various power reduction technique by routing optimization. There are lakhs of nets in the section(partition) and to work on all the nets is too tedious and time consuming. So, I am going to write an automation flow for the various power reduction techniques by using TCL scripting so that the overall human efforts and time can be reduced.

The different methods for the power reduction by optimized routing are as follows:

1. Moving the attacker net away from the power hungry net and placing a float wire in that vacant track so that no other routing can be placed there.
2. Downgrading to the lower metal layers and increase the spacing as they have low capacitance.

1.1 VLSI Design Flow

As per shown in figure 1.1, VLSI design flow is described as below:

- **Specification**

  Specification is the detailed explanation of requirement which comes from the customer. In specification customer will explain their requirement – in terms of functionality, operation frequency, power budget, cost, area of the chip, duration required to finish the project with proper deadlines etc.

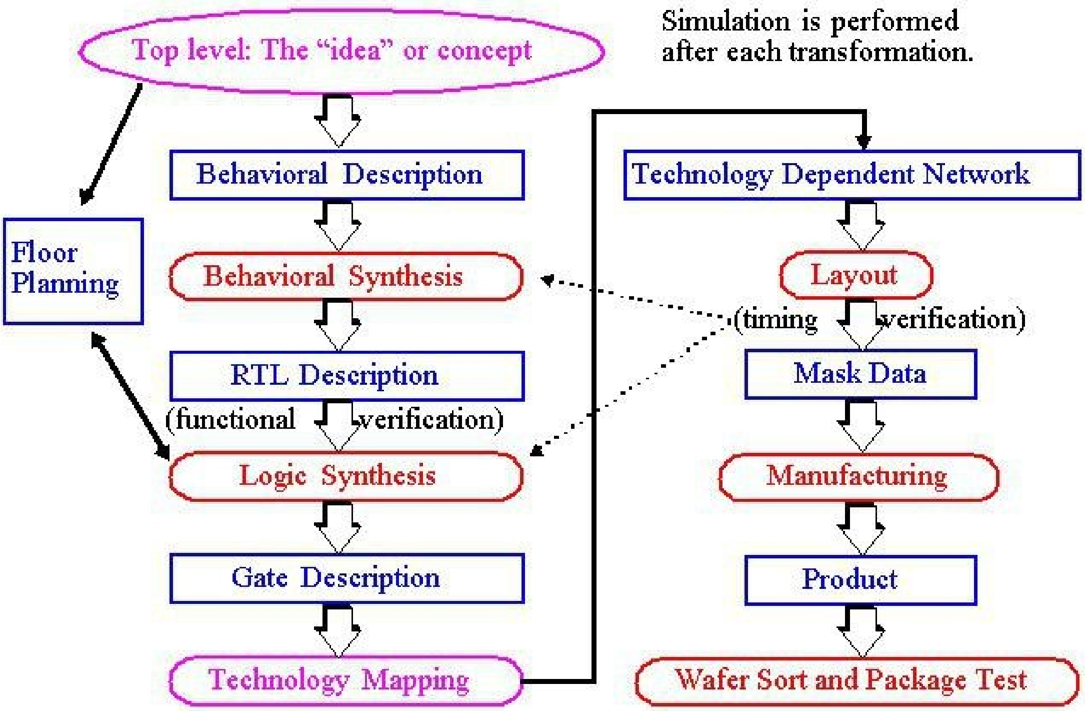


Figure 1.1: VLSI Design Flow

- **Behavioral Description**

  After getting complete description of the functioning and understanding, behavioral code is written in Verilog, VHDL or SystemVerilog. This coding will be the explanation of behavior of the chip to be implemented. This thing will be coded with if else, looping, giving some condition, etc.

- **Behavioral Synthesis**

  This all coded stuff will be synthesized and some circuit of it will be created with the help of synthesizer that will transform logic code into standard cell and their connection.

- **RTL Description**

  RTL description is converting behavioral model into Register transfer logic which will have information of gate level structure in Verilog code.

- **Functional verification**

  This thing must be verified with different test cases. For example 32 bit input function there will be 232 input combination. To give all the test cases is difficult. So the test cases will be generated such that it could give max coverage of all the possible functionality and different possible output. Verification continues even project is getting into last stage and if bug will be found it will re synthesize fub and will be implemented.

- **Floor-planning**

  While all this thing is being executing floor plan of the design is taking place in parallel from the past experience or the knowing behavior of its all Functional unit block and their connectivity. Floor planning is done by the layout people. This will save the time.

- **Gate Description**

  RTL description will have information of gate and it will be in visual form. From this information netlist will be created. Netlist will be on the base of the standard cell library. It means that to create netlist there must be cell, which is in standard cell. For the timing improvement or for power performance cell will be changed, or buffer or inverter will be added.

- **Technology mapping**

  In this stage all the standard cell be converted into technology dependent standard cell which had timing information as per the technology. This Cell can be changed as per the timing requirement with the same cell but they have different behavior. This Behavior is based on the design of cell, like low leakage cell, high speed cell.

- **Layout**

  Layout is being done in hierarchical way, e.g. fub layout, section layout, full chip lay-out. Each type of hierarchy is connected to each other and will be integrated inside each other. Layout is the process of cleaning DRC, floor-planning, Timing management, clock integration etc.

- **Mask Data**

  From the Layout GDS-II file will be created. This file will be the input to the process of Mask Creation. This mask are made up for different metal layer or poly layer. It can be more than one mask for the same metal layer.

- **Testing**

  Testing is the process in which Fabricated chip will be tested in different environment with different possible test cases. Testing can be for Chip reliability. Chip will be tested in different temperature pressure.

1.2 Back-end Design Flow

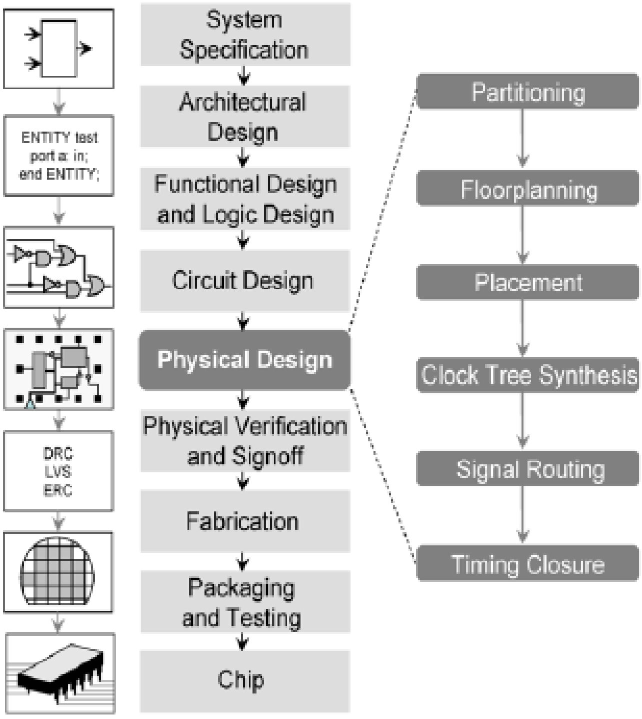


Figure 1.2: Back-end Design Flow

As per shown in figure 1.2, Back-end design flow is described as below:

- **Partitioning**

  Decomposition of a complex system into smaller subsystems.Each subsystem can be designed independently speeding up the design process.Decomposition scheme has to minimize the interconnections among the subsystems. Decomposition is carried out hierarchically until each subsystem is of manageable size.

- **Floor plan**

  A floor planning is the process of placing blocks/macros in the chip/core area as per the connection with another blocks. While all this thing is being executing floor plan of the design is taking place in parallel from the past experience or the knowing behavior of its all Functional unit block and their connectivity.Floor planning is done by the layout people. This will save the time.

- **Placement**

  Placement is the process of placing standard cells in the rows created at floor-planning stage. The goal is to minimize the total area and interconnects cost. The quality of routing is highly determined by the placement.

- **Clock Tree Synthesis**

  Clock Tree Synthesis is a process which makes sure that the clock gets distributed evenly to all sequential elements in a design to minimize the skew and latency. The clock tree constraints will be Latency, Skew, Maximum transition, Maximum capacitance, Maxi- mum fan-out, list of buffers and inverters etc. The clock tree synthesis contains clock tree building and clock tree balancing. Clock tree can be build by clock tree inverters so as to maintain the exact transition (duty cycle) and clock tree balancing is done by clock tree buffers (CTB) to meet the skew and latency requirements.

- **Routing**

  Routing is an important step in the design of integrated circuits (ICs). It generates wiring to interconnect pins of the same signal, while obeying the manufacturing design rules. Two approaches are used i.e. Global and Detailed routing. Global routing first partitions the routing region into tiles and decides tile-to-tile paths for all nets while attempting to optimize some given objective function (e.g., total wire length and circuit timing). Then, guided by the paths obtained in global routing, detailed routing assigns actual tracks and vias for nets.

- **Physical Verification**

  In the layout,It need to check all the Design Rule Check which are specified by the foundry for the fabrication process.So cleaning of different DRCs are needed after the routing it is called the physical verification.

1.3 Problem Statement and Solution

Back-end design style have sections with sea of fubs, as against big partition with the sea of cells.Full chip architecture is divided into 8-10 sections. One section has around 100+ fubs. These fubs has lakhs of nets so it becomes tedious to work on each and every net. Along with the timing constraints and routing optimization, power also plays very important role in place and route. The main objective of this project is to reduce the dynamic power of the power hungry nets by reducing the dynamic capacitance (Cdyn). Here I am going to describe various power reduction technique by routing optimization. There are lakhs of nets in the section(partition) and to work on all the nets is too tedious. So here I am going to write an automation flow for the various power reduction techniques by using TCL scripting so that the overall human efforts and time can be reduced. These utilities improves the power on the nets present in the layout using an algorithm. This helps the layout designer to reduce efforts in modifying the nets manually. This utility allows the layout designer to look in the case wherein congestion of layers is on its peak, and there is no track left to work upon. Such cases are left for manual attention.

The different methods for the power reduction by optimized routing are as follows:

1. Moving the attacker net away from the power hungry net and placing a float wire in that vacant track so that no other routing can be placed there.

2. Downgrading to the lower metal layer and increasing its spacing as they have low capacitance.

Chapter 2

Section Layout Flows

2.1 Introduction to Section Layout

Design of any SOC (System on Chip) is hierarchical design. The Hierarchy is as shown in figure. From Figure we can see that it consist of Transistor as basic element, gates are made up of transistors. Standard cell are made up of gates. Functional unit block are made up of standard cell. This Functional Unit block had some specific logic built inside it with the help of standard cell. Functional unit blocks are integrated inside Section. Section is made up of lot more number of Functional unit blocks. Combining all these, Section Full Chip is created.

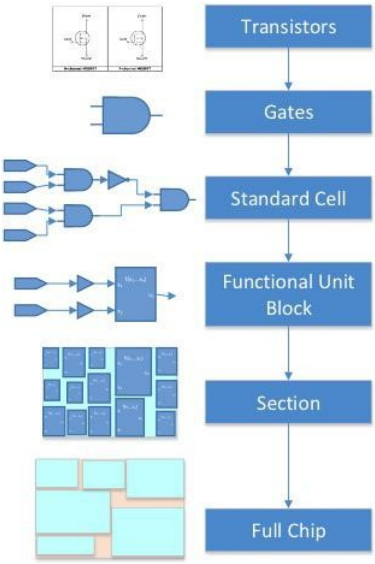


Figure 2.1: Hierarchical design of SOC

2.2 FUB Integration

Fub integration is the process of integrating fub inside section. fub had information of pin terminal and its lower metal layer. This will be transferred to section and from the netlist information nets will be connected. Than DRC (Design Rule Check) process will started and cleaning will be done manually.

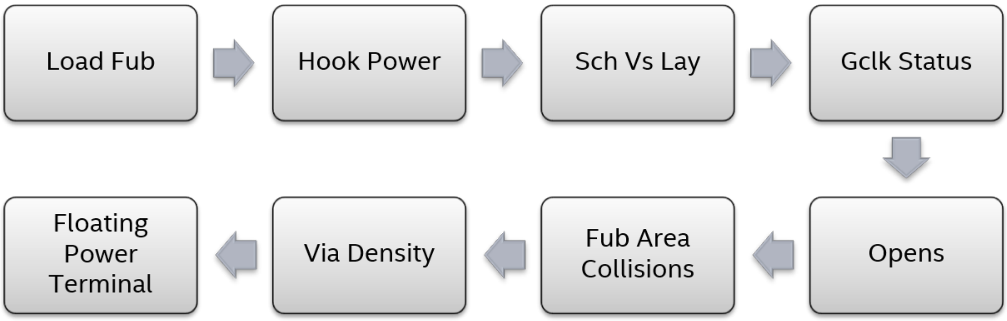


Figure 2.2: FUB Integration Flow

- Load Fub will load pins and nets inside the section from fub.
- Hook Power will create a vias between power wires.
- Sch vs Lay : Pins which are their inside the fub but are not replicated into the section.
- Gclk status will calculate the RC delay on the clock port inside fub
- Opens is a checker to find opens in a layout.
- Density check will find the density of via in some unit region and check for the reliability.
- Fub Area Collision will flag all the DRC which are there inside the section.
- Floating power terminal will flag the pins which has no connection inside the section.

2.3 Clock Integration

Clock had its own structure and it will be transferred to all the part of section from global driver. Fub has clock port this needs to be integrated separately with considering max delay allowed to reach the clock to their terminal.

2.4 Types of DRCs (Design Rule Check)

There are different types of DRCS present in layout and it is listed below:

- **Via to Via**

  When Via of Nth metal layer and via of (N-1)th metal layer comes closure then predefined distance then this will be pop up. Main reason for this type of DRC is that via can dissipate heat and can create problem of metal melting. This type of violation is shown in figure 2.3.

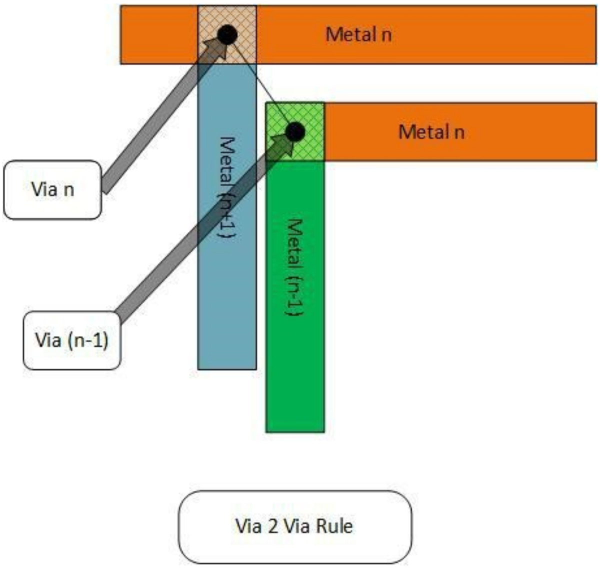


Figure 2.3: Via to Via DRC

- **Via cut to cut**

  When same type of three via comes close to each other than this will be aged. Same type of via means all the metal wire had same layer. This type of error makes triangle of via and it is shown in figure 2.4.

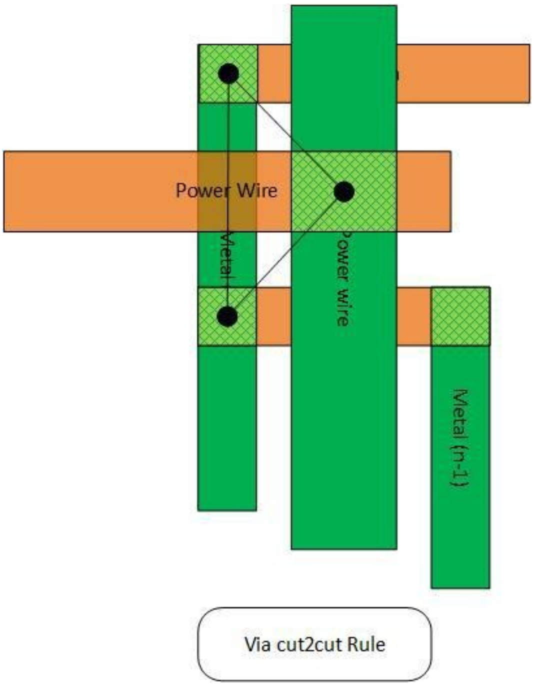


Figure 2.4: Via Cut to Cut DRC

- **Via Coverage Enclosure**

  Via is not properly being made with compare to metal layer. It is slightly here and there that is not proper design. This type of DRC is shown in figure 2.5.

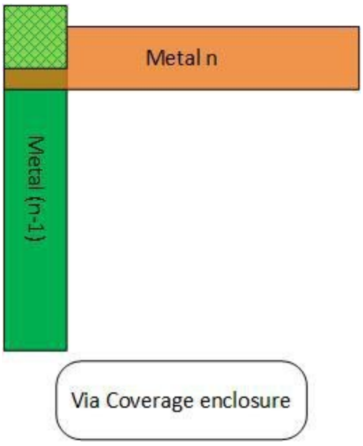


Figure 2.5: Via Coverage Enclosure DRC

- **Via Overlap**

  Via of 3 different metal layer should not have overlap with each other. Maximum two via can overlap. More than two via overlap is also DRC.

- **Min Space between Same Metal Wire**

  Two type of min space are there one is end to end space and another one is side by side spacing. Both of the space is already being defined for each metal layer, each metal layer had different value for min space. This is mainly because of inability to expose metal wire properly with lithography. Min space violation is shown in figure 2.6.

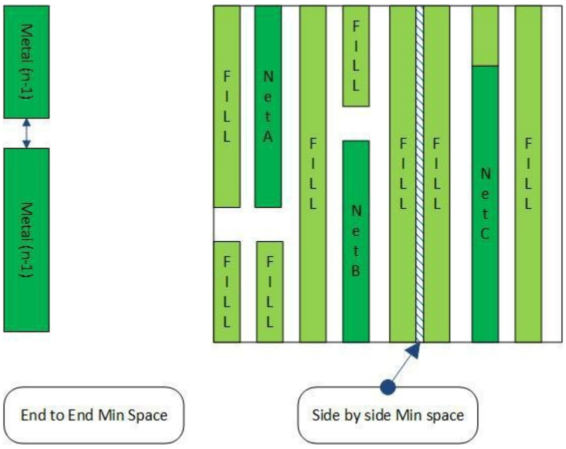


Figure 2.6: Min Space Rule DRC

- **Max Space between Same metal wire**

  This type of mainly comes into picture after metal ll. This can cause density issue after the silicon production. So as per the DRC between two metal wires there should not be distance more than pre-defined. This violation is shown in figure 2.7.

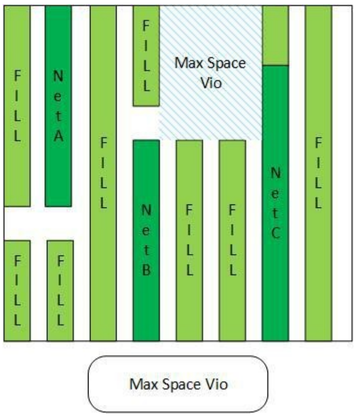


Figure 2.7: Max Space Rule DRC

- **Shorts**

  This checker will and out the short between two metal wires. This comparison is being made as per the netlist and the name if the net if two different net (which had different name) gets collide with each other than it is called as short. This violation is shown in figure 2.8.

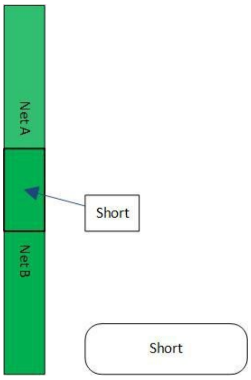


Figure 2.8: Metal Short Rule DRC

- **Min Length of Wire**

  There is some predefined value of metal wire length. Which is a limitation of technology. It means that for specific metal layer, technology can't create smaller metal wire than that pre-defined value. This violation is shown in figure 2.9.

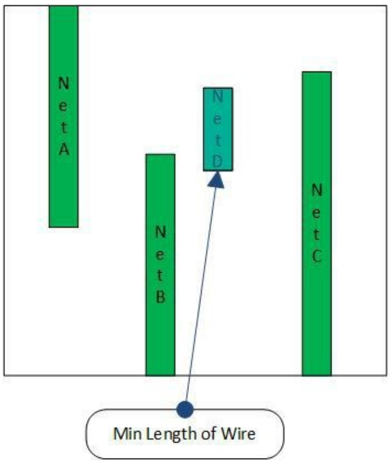


Figure 2.9: Min Length Rule DRC

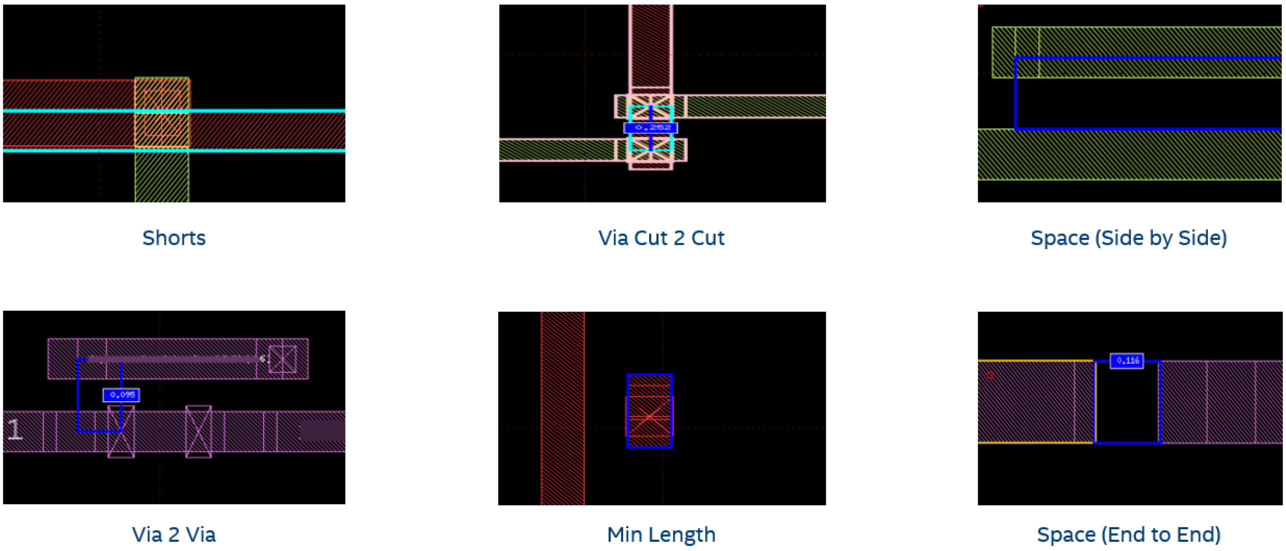


Figure 2.10: Types of DRCs

Here I have got the real images of the different types of DRCs from the tool as per above figure 2.10.

Chapter 3

Dynamic Power Reduction Techniques

As we know that capacitance and power are directly proportional to each other and thus to reduce the power we need to reduce the capacitance.

$$P_{dy} = \propto * \ C_{dyn} \ * V^2 \ * f$$

Figure 3.1: Relation between power and capacitance

The different methods for the power reduction by capacitance reduction and optimized routing are as follows:

(a) Moving the attacker net away from the power hungry net and placing a float wire in that vacant track so that no other routing can be placed there.

(b) Downgrading to the lower metal layer and increasing its spacing as they have low capacitance.

But these techniques can be implemented only when:

(a) At the early stage of the project when the overall routing congestion is very less so we can have enough resources to place the float.

(b) And for the particular power hungry net, we have met the timing.

Figure 3.2: Stage of usage of power reduction techniques

The methods for the dynamic power reduction are:

3.1 By reducing the cross capacitance

The nets which are consuming the maximum power are listed and from them, power would be reduced by reducing the dynamic capacitance.

We know that C= (E * A) / d , the distance between the net is inversely proportional to the capacitance, so as we move the net away or increase the spacing between the nets, the capacitance decreases and thus the power reduces as seen in figure. 3.3.

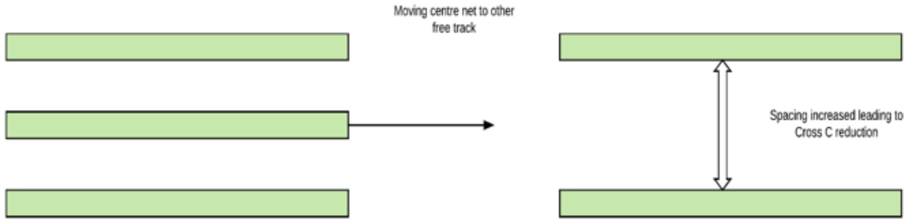

Moving centre net to other free track
Spacing increased leading to Cross C reduction

Figure 3.3: Wire Spacing

There are two capacitance as per below:

(a) Self Capacitance (w.r.t Ground)

(b) Cross Capacitance (w.r.t Nearby nets)

With the decrease in feature size on silicon, the main component contributing to the power of a wire is the cross capacitance between adjacent wires, and power is the dominating factor in the circuit performance. As much as 50-75 percent of the power is contributed by the cross capacitance. This work can also be used to minimize delay and the peak cross-talk by increasing the space between interconnects. It can reduce the circuit delay significantly along with power and in addition, reduce the peak cross-talk problem.

So the net which has the smallest length is moved in the nearby track available looking into the case that no DRCs are generated.

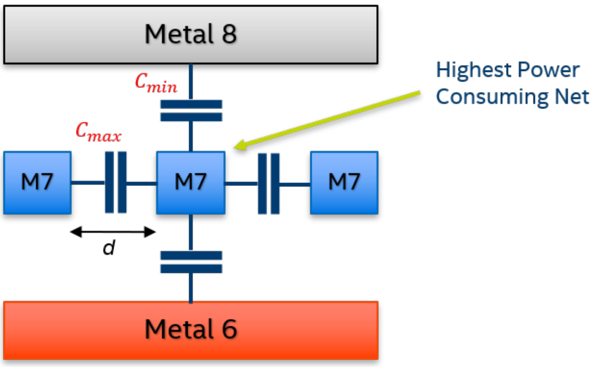


Figure 3.4: Reduction of cross capacitance by moving the net away

Here in the figure 3.4, the middle M7 has the highest power. So, as per the available track the next M7 on both the sides are moved away creating no DRC's.

There are two types of capacitance between the metal layers, that is Cmin and Cmax . Cmin is considered negligible as the distance between the metal layers are taken care of while fabrication and its distance is decided from the foundry as per the requirement, while we are considering only Cmax as it plays the major role in generating max capacitance. Where the net is moved, the tracks gets vacant so to prevent other net to sit on the track we will place a float net in that vacant tracks.

The float wire is preferred compared to shield as float has no potential of its own i.e. it is just a plain wire without any current passing through it , just used for the density fixes. While shield is having either high or low potential. So float act as a dielectric between the two nets, while shield contributes to some amount of capacitance. Thus, float wire is used.

Alternate metal layers are placed orthogonal to each other as shown in figure 3.5, so that their area of overlap is minimum, which results in less noise and capacitance effect and thus dynamic power can be reduced.

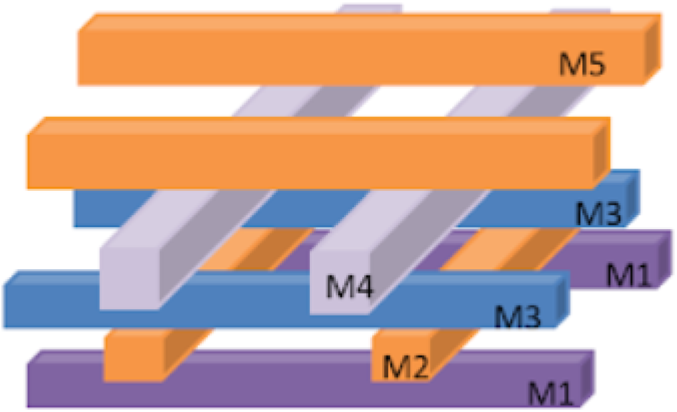


Figure 3.5: The placement of the metal tracks

3.2 Downgrading to lower metal layers

Modern processes use 6-10+ metal layers.

M1-M4= thin and narrow(for connecting wires and high density cells)

M5-M7 = thicker(for longer nets)

M8- M12 = thickest(for longer nets, power supply, Clock)

R =r l/A and C =E A/d so as A increases, R decreases and C increases.

So as we move from higher to lower metal layers the capacitance decreases and thus the power reduces. Also good spacing and width needs to be provided for capacitance reduction.

But while moving from higher to lower metal layer we again have to see that no DRCs are generated while taking metal jogs.

We can also change the repeater solution in the interconnects to reduce the power. The various repeater solution used as per requirement are:

(a) HVT : High V threshold. Can be used in the path where timing is not critical. So by using HVT cells we can save power.

(b) LVT : Low V threshold. One should use these cells in timing critical paths. These cells are fast but , consumes more power due to its leakage. So it will consume more power. So use only when timing is critical.

(c) SVT : Standard V threshold. Best of both world. Medium delay and medium power requirement. So if timing is not met by small margin with HVT, you should try with SVT. And at last LVT.

While using these methods for power reduction we need to take care that no timings and slope are degraded as timing and frequency are the main concern in the project.So along with the power reduction we need to take care about the timing. If the timing margins are good then only we should change the routing otherwise we need to revert back the changes.

We use the "ELMORE DELAY MODEL" in the tool to extract the resistance an capacitance value to calculate the timing and thereby quick estimation of the delay values.The Elmore delay explains the delay from input to output of the signal.This is simplest distributed RC network, and is more accurate

Chapter 4

Flowchart and Manual Results

4.1 Flowchart

(a) **BY REDUCING CROSS CAPACITANCE:**

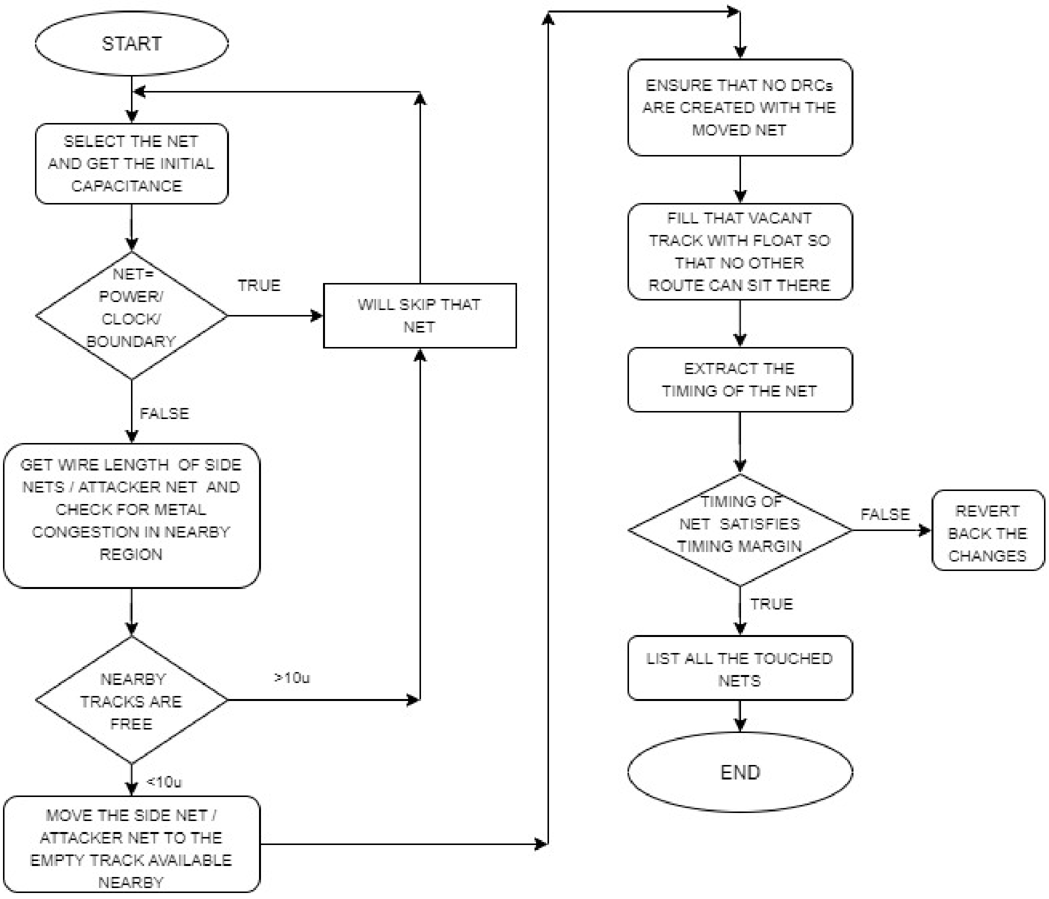

START
SELECT THE NET AND GET THE INITIAL CAPACITANCE
NET= POWER/ CLOCK/ BOUNDARY
TRUE
WILL SKIP THAT NET
FALSE
GET WIRE LENGTH OF SIDE NETS / ATTACKER NET AND CHECK FOR METAL CONGESTION IN NEARBY REGION
NEARBY TRACKS ARE FREE
>10u
<10u
MOVE THE SIDE NET / ATTACKER NET TO THE EMPTY TRACK AVAILABLE NEARBY
ENSURE THAT NO DRCs ARE CREATED WITH THE MOVED NET
FILL THAT VACANT TRACK WITH FLOAT SO THAT NO OTHER ROUTE CAN SIT THERE
EXTRACT THE TIMING OF THE NET
TIMING OF NET SATISFIES TIMING MARGIN
FALSE
REVERT BACK THE CHANGES
TRUE
LIST ALL THE TOUCHED NETS
END

Figure 4.1: METHOD-1: By reducing the cross capacitance

(b) **BY DEGRADING WIRE TO LOWER METAL LAYER:**

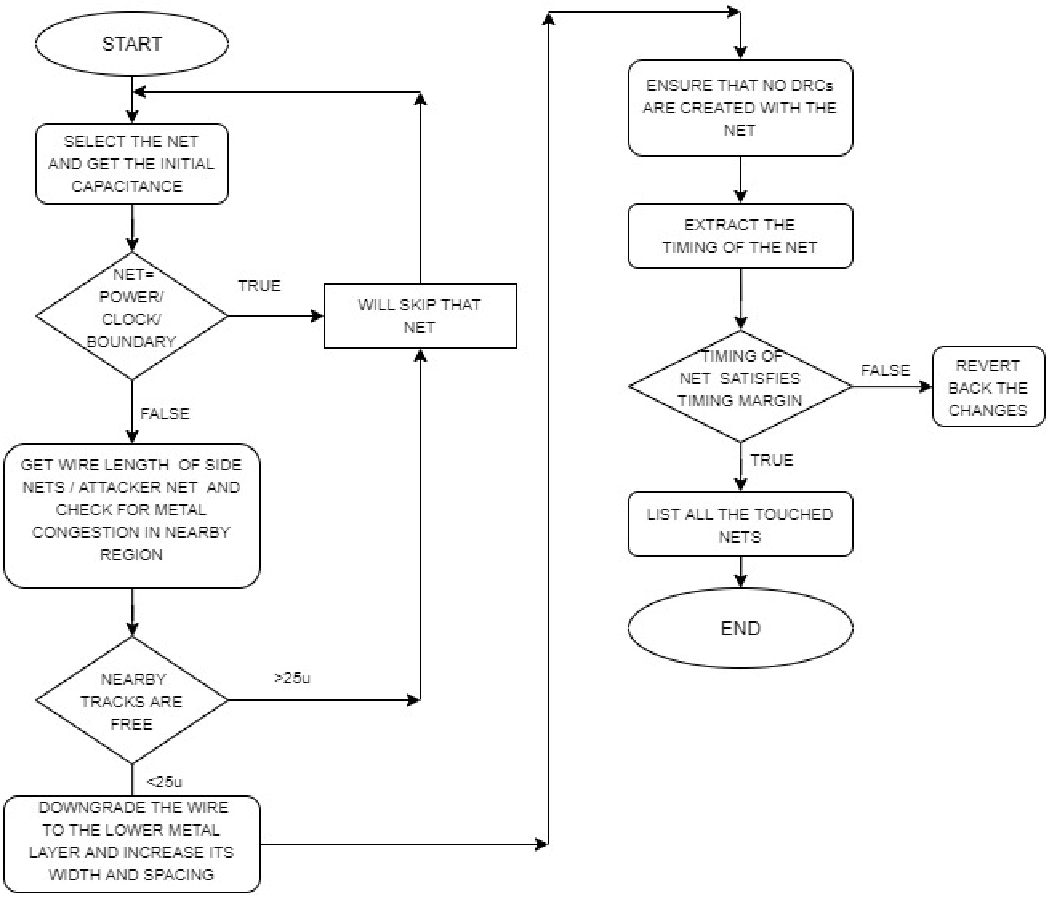

START
SELECT THE NET AND GET THE INITIAL CAPACITANCE
NET= POWER/ CLOCK/ BOUNDARY
TRUE
WILL SKIP THAT NET
FALSE
GET WIRE LENGTH OF SIDE NETS / ATTACKER NET AND CHECK FOR METAL CONGESTION IN NEARBY REGION
NEARBY TRACKS ARE FREE
>25u
<25u
DOWNGRADE THE WIRE TO THE LOWER METAL LAYER AND INCREASE ITS WIDTH AND SPACING
ENSURE THAT NO DRCs ARE CREATED WITH THE NET
EXTRACT THE TIMING OF THE NET
TIMING OF NET SATISFIES TIMING MARGIN
FALSE
REVERT BACK THE CHANGES
TRUE
LIST ALL THE TOUCHED NETS
END

Figure 4.2: METHOD-2: By Downgrading to lower metal layer

As shown in figure 4.1 and 4.2, the two methods are proposed here.

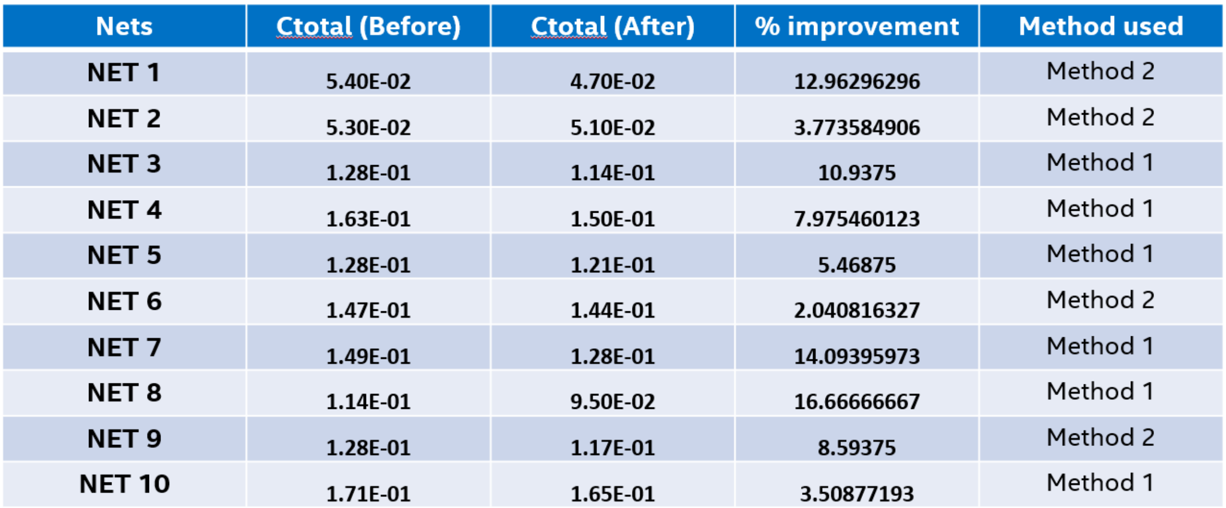
In Method 1, we move the side nets or attacker nets to the empty track available nearby and the vacant track will be filled by float so that no other route can sit there. By implementing this method 1, we can reduce the capacitance by some amount and we can reduce the power.

In Method 2, we downgrade the highest power consuming net to the lower level metal layer and increase the spacing of that net. By doing this we can reduce the some amount of capacitance of the power consuming net and thus we can reduce the power consumption.

Both these methods will be implemented manually first and then get the manual results of its improvement. After that, we will implement these methods using TCL script so that we can reduce the manual time and effort. In the next section, we will see the manual results of these two techniques and after that we will implement using TCL and compare both the results.

4.2 Manual Results

Both the mentioned methods when manually implemented, we could see capacitance improvement.



Nets	Ctotal (Before)	Ctotal (After)	% improvement	Method used
NET 1	5.40E-02	4.70E-02	12.96296296	Method 2
NET 2	5.30E-02	5.10E-02	3.773584906	Method 2
NET 3	1.28E-01	1.14E-01	10.9375	Method 1
NET 4	1.63E-01	1.50E-01	7.975460123	Method 1
NET 5	1.28E-01	1.21E-01	5.46875	Method 1
NET 6	1.47E-01	1.44E-01	2.040816327	Method 2
NET 7	1.49E-01	1.28E-01	14.09395973	Method 1
NET 8	1.14E-01	9.50E-02	16.66666667	Method 1
NET 9	1.28E-01	1.17E-01	8.59375	Method 2
NET 10	1.71E-01	1.65E-01	3.50877193	Method 1

Figure 4.3: Manually Result of both the methods

Here as per the figure shown above 4.3, we can see that for few power critical nets, the improvement is more that 10 percentage and for some it is less than 10 percentage. Based on the above figure it can concluded that by using both methods, capacitance can be improve and thus we can improve the power consumption. But implementing both methods manually, requires more time and efforts and also it is impossible to work on lakhs of nets manually. So need arose to implement these techniques by using TCL script and get faster results. Here, I will also compare the manual time and automated script time required for the capacitance improvement.

Chapter 5

Pilot Results of the Script

In this chapter, I have described the automation for both the methods along with the various pilot results.

5.1 Automation-1(to reduce the cross capacitance)

- The script will work on the given list of nets per partition.
- For this automation, the nets list is taken as input.
- Before running this automation, the owner of each partition needs to review the criticality of that net and congestion of that region.
- It will put float wires on both sides of each wire of the power hungry net.
- It will grep all the wire from the net and work on wires greater than 50u.
- If the targeted net is next to power, it will not place float on that side.
- It will try to clean all the shorts created in that process without moving the targeted net and its floats.
- If a float (for some other net) already exist in the required track, then it will try to extend the float instead of creating a new float on top of that and creating junction.
- No offgrid issues are seen after running the automation.
- If a wire which is having power_fix attribute(i.e. the wire which is already worked on) exist in the next track then it will skip that side for float.

Below are the results for this automation:

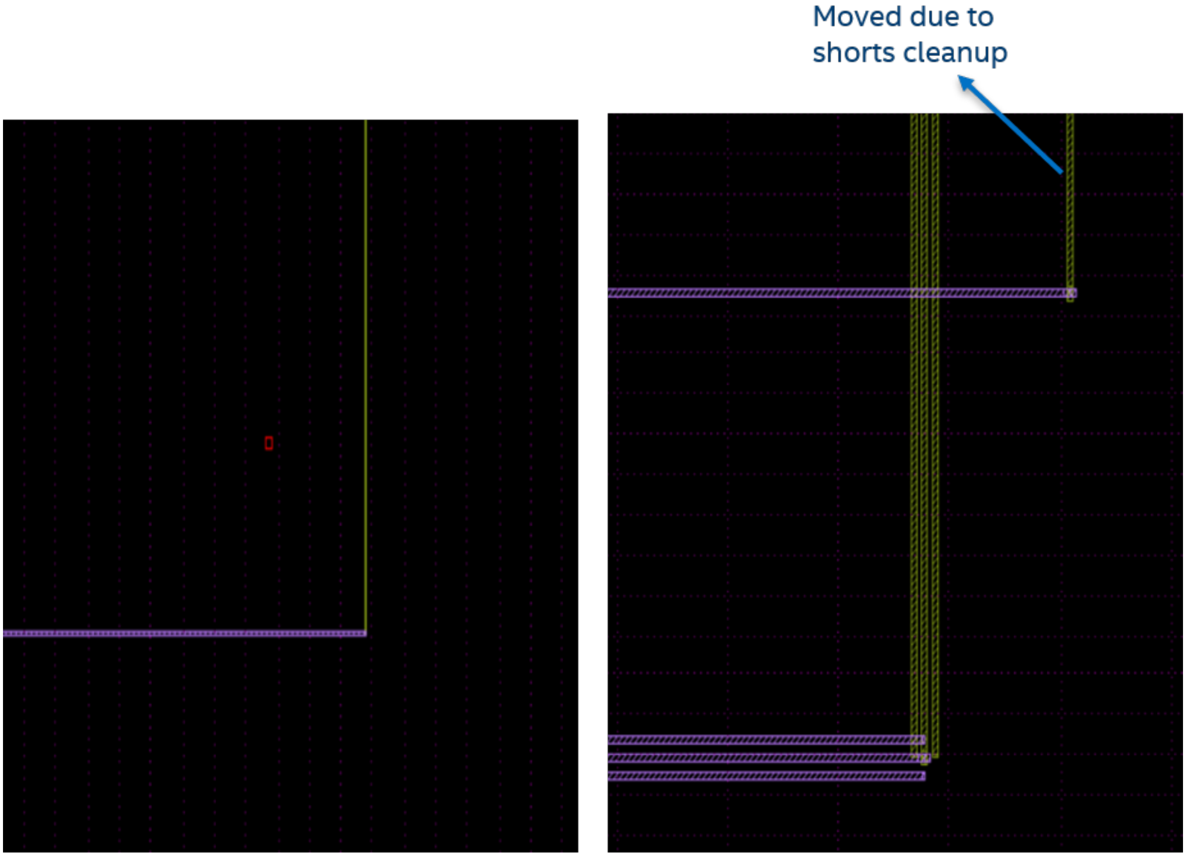


Figure 5.1: Creating float on both the side of wire

Here in the figure 5.1, it shows the results before and after the automation being used for creating the float on both the sides of wire. The automation creates float on each and every wire of the net and then moves the shorting wires to nearby empty tracks available therby creating a 2X spacing for power hungry net. It tries to clean as many shorts as possible without disturbing the main net and the float placed next to it.

Here in the figure 5.2, it shows the results before and after the automation. If the power hungry net is next to main power then instead of putting float on power it skips that side and creates the float on only one side.

Here in the figure 5.3, it shows the results before and after the automation. If the float (for some other net) already exist in the required track, then it will try to

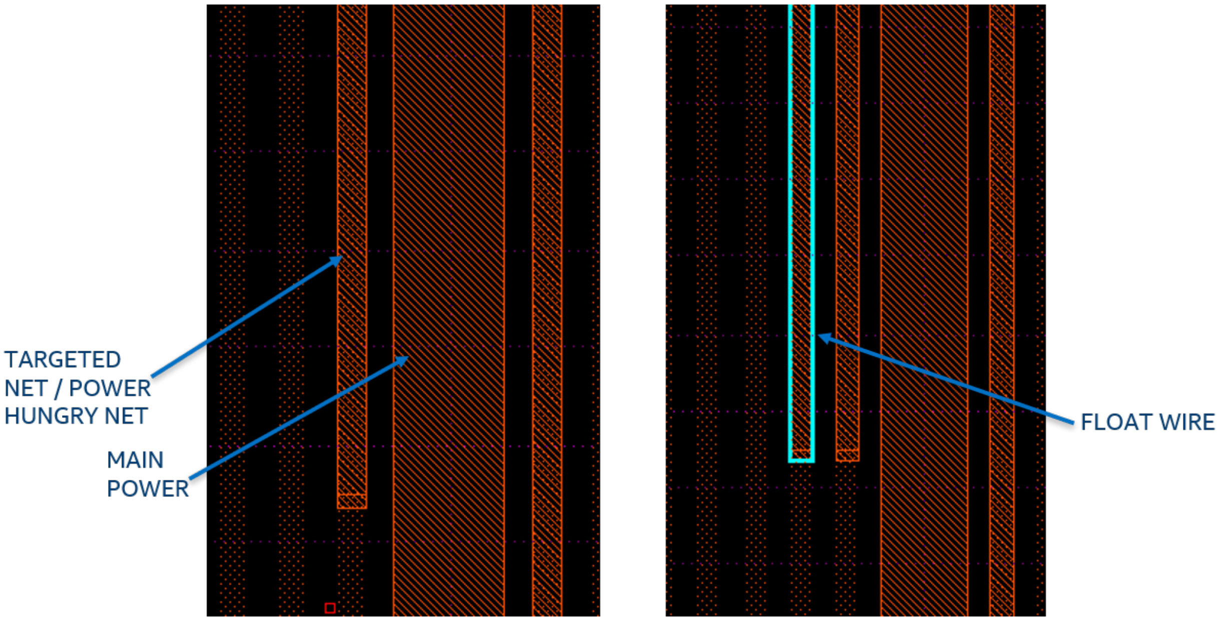


Figure 5.2: Skipping float if main wire is beside power

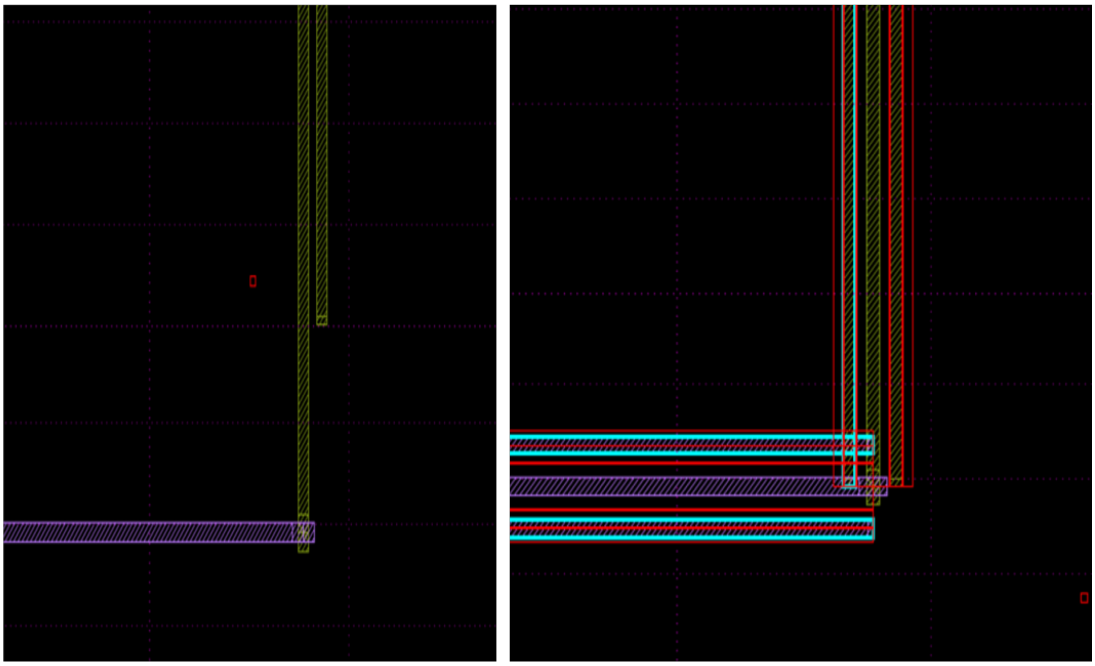


Figure 5.3: Extending the float if float already exist in track

extend the float instead of creating a new float on top of that and creating junction and thereby try to minimize the DRCs. In the first figure a float (for some other net) already exist on right side of wire, so it will just extend it rather than creating new float.

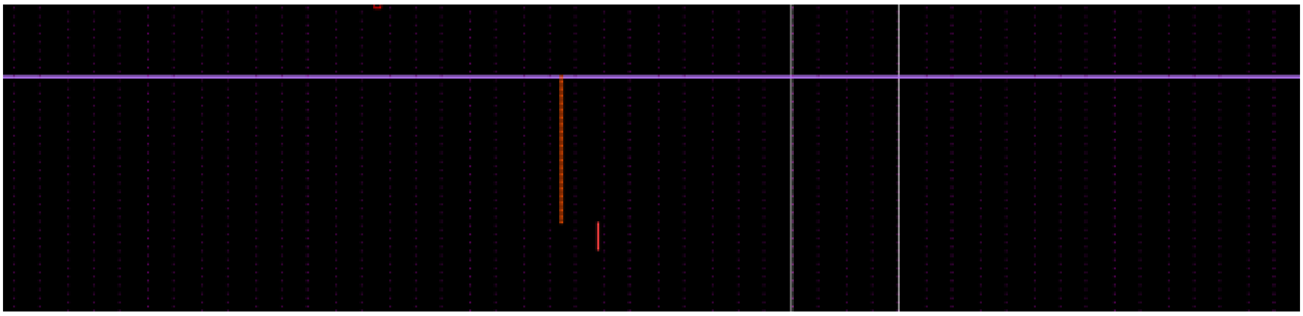


Figure 5.4: Before automation

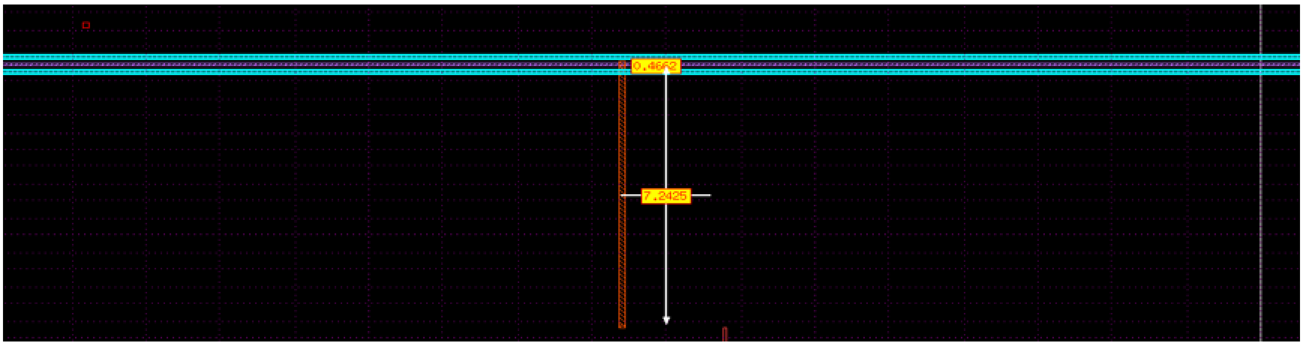
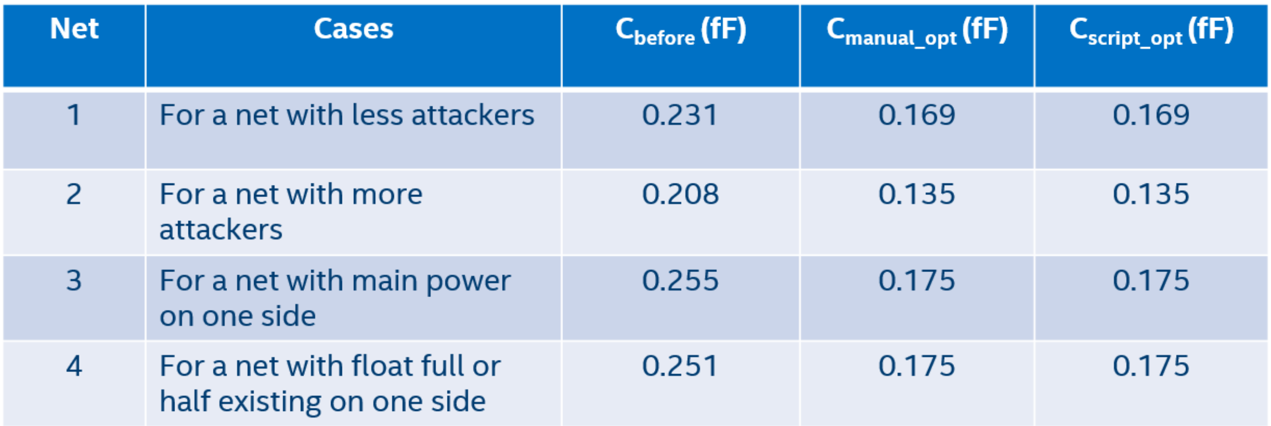


Figure 5.5: After automation skipping wire less than 50u length

Here in the figure 5.4 and 5.5, it shows that if the wire of a net is less than 50u in length than it will simply skip that wire,as creating float for small wires does not gain benefit in capacitance and increase the tracks utilization.



Net	Cases	C_{before} (fF)	C_{manual_opt} (fF)	C_{script_opt} (fF)
1	For a net with less attackers	0.231	0.169	0.169
2	For a net with more attackers	0.208	0.135	0.135
3	For a net with main power on one side	0.255	0.175	0.175
4	For a net with float full or half existing on one side	0.251	0.175	0.175

Figure 5.6: Improvement in capacitance

Here in the figure 5.6, the capacitance values for different sets of nets in different cases before method implemented, after method implemented manually and after the method implemented with script are compared. We can observe the capacitance reduction by this method. While the manual capacitance optimization values and script capacitance optimization values are same.

5.2 Automation-2(downgrading to lower metal layers)

- The script will work on the given list of nets per partition.
- For this automation, the nets list is taken as input.
- Before running this automation, the owner of each section needs to review the criticality of that net and congestion of that region.
- It will downgrade the net to the lower metal layer and increases its spacing.
- It will grep all the wire from the net and work on wires greater than 50u.
- It will create jog whenever required when we need to connect a lower metal to a higher one.
- No offgrid issues are seen after running the automation. It will place the downgraded wire with increased spacing as per its grid available.

Below are the results for this automation:

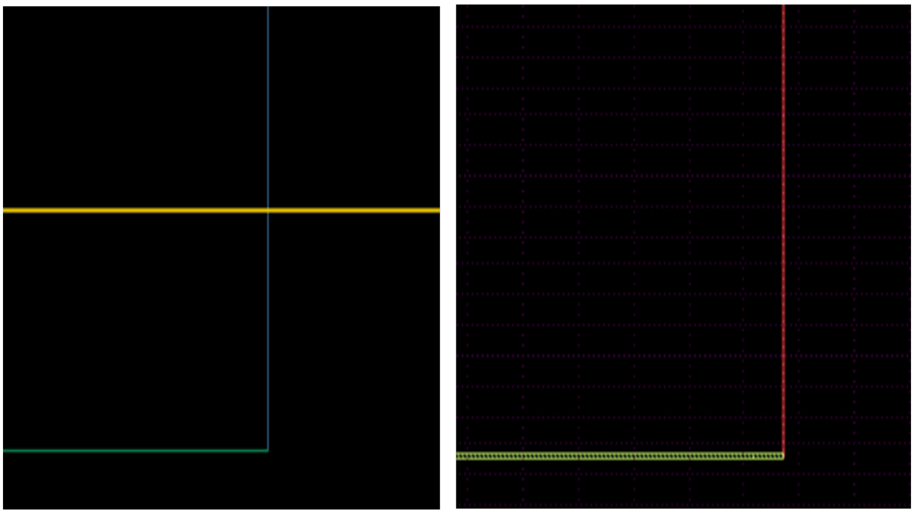


Figure 5.7: Downgrade to lower metal layers

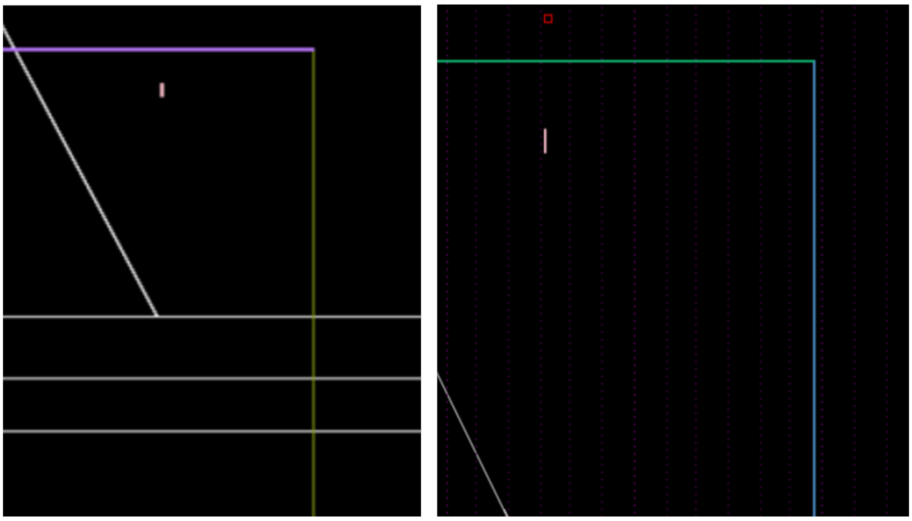


Figure 5.8: Downgrade to lower metal layers

Here in the figure 5.7 and 5.8, it is shown that the automation flow has downgraded the wires of net from layer ”n” (figures on left) to its lower metal layer ”n-2” (figures on right. The flow has also increases the spacing of each wire.

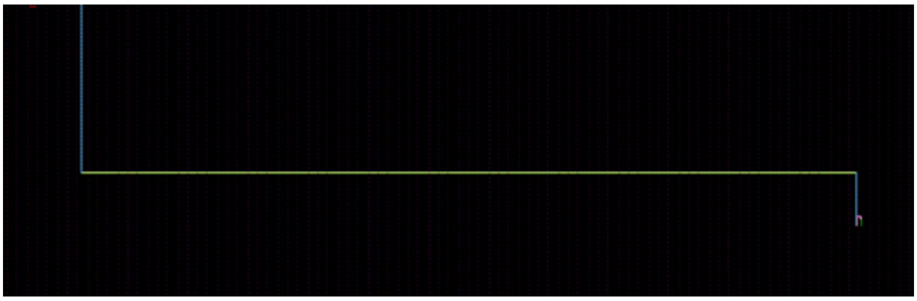


Figure 5.9: Before automation

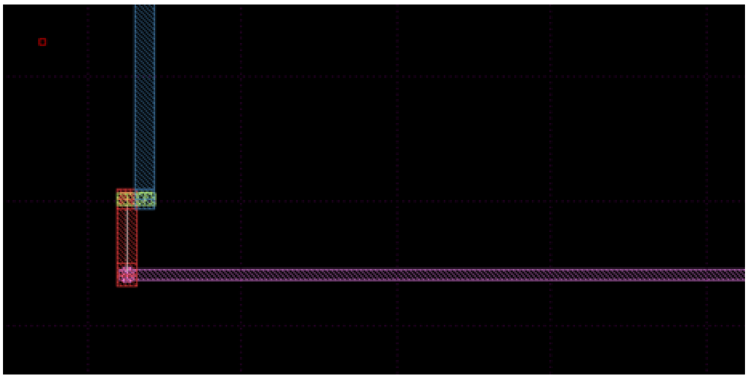
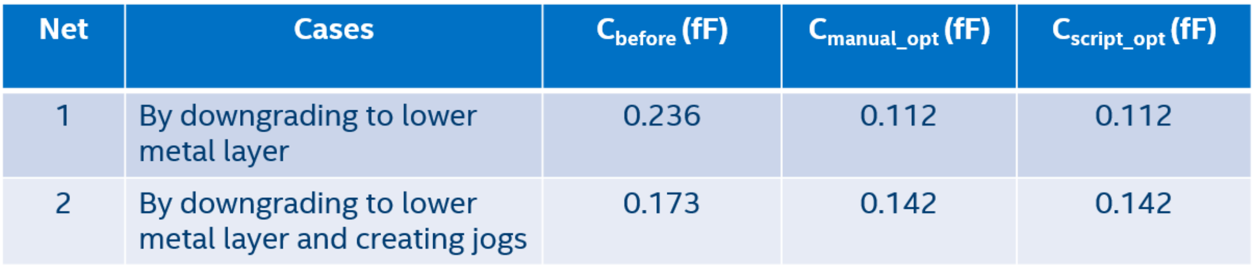


Figure 5.10: After automation downgrade to lower metal layers with creating jogs

Here in the figure 5.9 and 5.10, it is shown that the automation flow has created jogs when there is wire connection between metal layer "n" and metal layer "n-3" to prevent DRCs to be generated.



Net	Cases	C_{before} (fF)	C_{manual_opt} (fF)	C_{script_opt} (fF)
1	By downgrading to lower metal layer	0.236	0.112	0.112
2	By downgrading to lower metal layer and creating jogs	0.173	0.142	0.142

Figure 5.11: Improvement in capacitance

Here in the figure 5.11, the capacitance values for different sets of nets in different cases before method implemented, after method implemented manually and after the method implemented with script are compared. We can observe the capacitance reduction by this method. While the manual capacitance optimization values and script capacitance optimization values are same.

Conclusion

- As we move to higher technology nodes, need arises for power optimization.
- So the basic idea of this project is to implement power optimization techniques for the dynamic power reduction by routing optimization.
- The power optimization techniques explained in this project are downgrading the metal layers, decreasing the cross capacitance and changing the repeater solution, but keeping in account that there is no degradation to the timing critical nets.
- Current mode of work is to manually implement this methods with meeting timing constraints, so the main objective is to save time and human efforts by automating the flow.
- By using both the power reduction methods manually, the capacitance value reduces, and thus dynamic power can be reduced. To reduce the manual efforts, automation of this flow is done and thus reducing the time and efforts.
- 5% of overall nets in project are power hungry nets so this automation can fix all these nets in very less time and minimum efforts.
- It reduced 90% of manual efforts in DRC free routing on every partition.
- Also, it completed this whole process for a single partition at a minimal 10% time as compared to manual work.All the pilot results are also shown for the automation done.
- Out of around 10 lakh nets in project, results gave capacitance reduction for around 20k nets by these methods.

Bibliography

[1] Sadiq M Sait. Habib Youssef, *"VLSI PHYSICAL DESIGN AUTOMATION : Theory and Practice"*

[2] R. Arunachalam, E. Acar, S. R. Nassif *"Optimal Shielding/Spacing Metrics for Low Power Design"* Proceeding of the IEEE Computer Society Annual Symposium on VLSI, 2003, pp. 167-172, Feb 2003.

[3] C.Uttraphan ; M. Khalil Hani *"An optimization algorithm for simultaneous routing and buffer insertion with delay-power constraints in VLSI layout design "* Proc.of the IEEE, vol. 89, no. 4, pp 505-528.07 April 2014.

[4] Mutlu Avci and Serhan Yamacli *"An improved Elmore delay model for VLSI interconnects"* in Mathematical and Computer Modeling , in Vol 51, Issues 7-8, April 2010, Pages 908-914.

[5] Delay – *"Interconnect Delay Models"* VLSI Concepts [Accessed : 13th July 2019] Website link

[6] *"Net Delay or Interconnect Delay or Wire Delay"* ASIC - System on Chip - VLSI Design [Accessed : 3rd Sep 2019] Website link

[7] P.R. Gray and R.G. Meyer *"Analysis and Design of Analog Integrated Circuits"* New York: John Wiley and Sons, Inc., 1993.

[8] R. Ho, M. Horowitz, K. Mai *"The future of wires"*, Proc.of the IEEE, vol. 89, no. 4, pp 490-504, Apr 2001.

[9] "Tcl/TK for EDA" *DOULOS Developing and Delivering KnowHow* [Accessed : 25th Aug 2019] Website link

[10] "TCL tutorial" [Accessed : 20th Aug 2019] Website link

[11] Intel Internal Documents.

M.Tech Thesis

ORIGINALITY REPORT

14%	11%	4%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Sardar Vallabhbhai National Inst. of Tech.Surat Student Paper	3%
2	www.slideshare.net Internet Source	2%
3	shinjaewook.blogspot.com Internet Source	1%
4	vlsibasic.blogspot.com Internet Source	1%
5	citeseerx.ist.psu.edu Internet Source	1%
6	www.coursehero.com Internet Source	1%
7	opac.nitrkl.ac.in Internet Source	1%
8	edoc.site Internet Source	1%
9	Submitted to Columbia College of Missouri	

	Student Paper	1%
10	Submitted to Charotar University of Science And Technology Student Paper	1%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		