

Semi Custom Design of Functional Unit Blocks for High Speed Micro Processor

Major Project Report

*Submitted in partial fulfillment of the requirements
for the degree of*

Master of Technology
in
Electronics & Communication Engineering
(Communication Engineering)

By

Heta Shah
(17MECC18)



Electronics & Communication Engineering Department
Institute of Technology
Nirma University
Ahmedabad-382 481
May 2019

Semi Custom Design of Functional Unit Blocks for High Speed Micro Processor

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Electronics & Communication Engineering

By

Heta Shah
(17MECC18)

Under the guidance of

External Project Guide:

Mr. Nikhil Saxena

Engineering Manager

Intel India Pvt.ltd,

Bangalore

Internal Project Guide:

Dr. Sachin Gajjar

Associate Professor

Electronics & Communication Engineering Department,

Institute of Technology, Nirma University



Electronics & Communication Engineering Department

Institute of Technology-Nirma University

Ahmedabad-382 481

May 2019

Declaration

This is to certify that

- a. The thesis comprises my original work towards the degree of Master of Technology in Communication Engineering at Nirma University and has not been submitted elsewhere for a degree.
- b. Due acknowledgment has been made in the text to all other material used.

- Heta Shah

17MECC18

Disclaimer

“The content of this thesis does not represent the technology, opinions, beliefs, or positions of Intel Technology Pvt. Ltd., its employees, vendors, customers, or associates.”



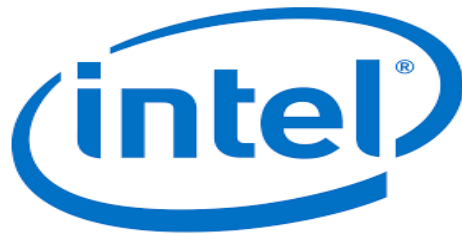
Internal Certificate

This is to certify that the major project entitled ”**Semi Custom Design of Functional Unit Blocks for High Speed Micro Processor**” submitted by **Heta Shah** (Roll No: 17MECC18), towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Communication Engineering 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 training, to the best of my knowledge, haven’t been submitted to any other university or institution for award of any degree or diploma.

Dr. S.H.Gajjar
Associate Professor,
ECE Department,
Institute of Technology
Nirma University, Ahmedabad

Dr. D.K.Kothari
Professor and HOD,
ECE Department
Institute of Technology,
Nirma University, Ahmedabad

Dr. Alka Mahajan
Director,
Institute of Technology,
Nirma University,Ahmedabad



Certificate

This is to certify that the Major Project entitled “**Semi Custom Design of Functional UnitBlocks for High Speed Micro Processor** ” submitted by **Heta B. Shah (17MECC18)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Communication Engineering, Nirma University, Ahmedabad is the record of work carried out by her under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.

Mr. Nikhil Saxena
Engineering Manager
Intel Technology India Pvt. Ltd.
Bengaluru

Acknowledgements

I would like to express my gratitude and sincere thanks to **Dr. Sachin Gajjar**, Associate Professor, Electronics& Communication Engineering Department, Institute of Technology, Nirma University and Internal Guide for guidelines during the review process.

I take this opportunity to express my profound gratitude and deep regards to **Dr. Yogesh N. Trivedi**, for his exemplary guidance, monitoring and constant encouragement.

I would also like to thank **Mr. Nikhil Saxena**, external guide of my internship project, **Mr. Karanvir Singh**, mentor of my project from **Intel Technology India Pvt. Ltd.**, for guidance, monitoring and encouragement regarding the project.

- Heta Shah

17MECC18

Abstract

Every year microprocessor core is shrinking in size owing to the advancement in technology and decrease in technology node. These evolutions have given excellent platform for fabricating very high-performance and multi-core processors with several new features. However, these changes also present newer challenges which are even harder to meet. For example, fabrication of devices of such smaller size in itself is difficult. From design perspective though we face various issues which hamper or progress while designing. The new challenges include more stringent and rigorous performance targets compared to previous project. Performance targets include area, timing as well as power parameters. In this project our aim is to improve on all these parameters so as to achieve higher quality of design. With every new design of microprocessor core there can be a requirement of new features to be added to the existing one. This feature update is incorporated in our design through RTL change. We will be working on one such update in this project and accordingly check its effect using various factors. How we accommodate such changes while still improving our design is shown in this project.

We use enhancements over existing design technologies in order to use goodness of the previous core design. This not only saves designers time but also makes it comparatively easier to deliver high quality end product. This project work shades light on design flows, design optimization techniques (both timing and power) as well as quality checks for sign off. We perform Reliability Verification check in order to enhance over value of the design. The techniques mentioned in this project work are tested and results analyzed to gain confidence on the methods. These methods can be used to achieve timing optimization, power optimization and improve the overall quality of our design. Frequency push and power reduction both have become important now due to shrink in technology. Timing performance though still dominant, power has started to become a bottleneck for our processors as the competition is now closing on the gap.

At Intel, microprocessor core architecture follows hierarchical approach. In hierarchical design we divide the system into sub parts. This plays as an advantage to us as it helps us to meet our targets in a better way. We basically follow a bottom up approach to get the required target.

Abbreviation Notation and Nomenclature

| | |
|--------------|------------------------------------|
| DP | Data Path |
| FUB | Functional Unit Block |
| RTL | Register Transfer Level |
| FinFET | Fin Shaped Field Effect Transistor |
| RLS | RTL to Layout Synthesis |
| SoC | System on Chip |
| FEV | Formal Equivalence Verification |
| DRC | Design Rule Checker |
| LVS | Layout versus Schematic |
| STA | Static Timing Analysis |
| WNS | Worst Negative Slack |
| TNS | Total Negative Slack |

Contents

| | |
|---|-------------|
| Declaration | iii |
| Disclaimer | iv |
| Certificate | v |
| Acknowledgements | vii |
| Abstract | viii |
| Abbreviation Notation and Nomenclature | x |
| 1 Introduction | 1 |
| 1.1 Introduction | 1 |
| 1.2 Approach | 2 |
| 1.3 Motivation | 3 |
| 1.4 Objective | 5 |
| 1.5 Thesis Organization | 5 |
| 2 Literature Survey | 6 |
| 2.1 Advancement in Microprocessor | 6 |
| 2.2 FINFETs | 8 |
| 2.3 Design Hierarchy | 10 |

| | | |
|----------|--|-----------|
| 3 | Design Methodology | 13 |
| 3.1 | Design Flow | 13 |
| 3.2 | RTL Coding | 14 |
| 3.3 | Schematic Design | 15 |
| 3.3.1 | Importance of Custom Design | 16 |
| 3.4 | Equivalence Verification | 16 |
| 3.5 | Placement of Blocks | 20 |
| 3.6 | Optimization | 22 |
| 3.7 | Quality Checks | 24 |
| 4 | Timing Optimization | 25 |
| 4.1 | Static Timing Analysis | 25 |
| 4.1.1 | Setup Time | 26 |
| 4.1.2 | Hold Time | 26 |
| 4.1.3 | Setup Violation | 27 |
| 4.1.4 | Hold Violation | 28 |
| 4.1.5 | Setup Fixing | 28 |
| 4.1.6 | Hold Fixing | 29 |
| 5 | Power Optimization | 32 |
| 5.1 | Power Optimization in Circuit Design | 32 |
| 5.2 | Power Driven Placement | 33 |
| 5.2.1 | Multi Vt Synthesis | 34 |
| 5.3 | Multi Bit Synthesis | 35 |
| 5.4 | Big Sequential Cells | 36 |
| 5.5 | Other Techniques | 37 |
| 6 | Quality Checks | 38 |
| 6.1 | Reliability Verification | 38 |
| 6.1.1 | Electromigration | 38 |

| | | |
|----------|---|-----------|
| 6.1.2 | IR Drop | 40 |
| 6.1.3 | Self Heat | 41 |
| 6.2 | RV Optimization Methods | 42 |
| 6.2.1 | Cell Spreading | 42 |
| 6.2.2 | Clock Buffer Splitting | 43 |
| 7 | Results | 45 |
| 7.1 | Formal Equivalence Verification | 45 |
| 7.2 | Timing Optimization | 48 |
| 7.3 | Power Optimization | 50 |
| 7.3.1 | Power Driven Placement | 50 |
| 7.3.2 | Multi Vt Synthesis | 52 |
| 8 | Conclusion and Future Work | 54 |
| 8.1 | Conclusion | 54 |
| 8.2 | Future Work | 55 |
| | Bibliography | 56 |

List of Figures

| | | |
|-----|---|----|
| 1.1 | Overview of microprocessor architecture | 3 |
| 2.1 | FinFET View | 9 |
| 2.2 | Two modes of FinFET | 10 |
| 2.3 | Hierarchal approach of microprocessor core design | 11 |
| 3.1 | Design Flow | 15 |
| 3.2 | Verification Flow | 17 |
| 3.3 | Mapping in FEV | 19 |
| 3.4 | Verification of Logic | 20 |
| 3.5 | Placement of Cells | 21 |
| 3.6 | Routing of Interconnects | 22 |
| 3.7 | Distribution of Metal Layer | 23 |
| 4.1 | Types of Logical Paths | 26 |
| 4.2 | FF-D-FF Path | 27 |
| 4.3 | Flow Chart for Setup Fix | 30 |
| 4.4 | Flow Chart for Hold Fix | 31 |
| 5.1 | Reduction of net length in LPP | 33 |
| 5.2 | Multi Vt Synthesis | 35 |
| 5.3 | Multibit conversion | 36 |
| 6.1 | Electromigration | 39 |

| | | |
|-----|---|----|
| 6.2 | IR Drop | 40 |
| 6.3 | Self Heat Effect on Interconnects | 42 |
| 6.4 | Cell Spreading Technique | 42 |
| 6.5 | Clock Buffer Splitting | 43 |
| 7.1 | Reference Verification Tool Status | 46 |
| 7.2 | Total number of different pairs | 46 |
| 7.3 | Current Design Verification Tool Status | 47 |
| 7.4 | Status of cones of design | 47 |
| 7.5 | values of TNS and WNS | 50 |
| 7.6 | Comparison of results for LPP synthesis | 52 |
| 7.7 | Activity Factor comparison | 52 |

List of Tables

| | | |
|-----|---|----|
| 2.1 | Comparison between CMOS FinFET | 8 |
| 4.1 | Path Type and Description | 26 |
| 7.1 | values of TNS and WNS for run1 | 48 |
| 7.2 | values of TNS and WNS for run 2 | 49 |
| 7.3 | values of TNS and WNS for run 3 | 49 |
| 7.4 | Result of LPP Synthesis | 51 |
| 7.5 | Multi Vt Synthesis Results | 53 |

Chapter 1

Introduction

1.1 Introduction

The exponential growth of the semiconductor industry demands newer and better techniques to stabilize the complexity in designs. Advanced microprocessor is a System on Chip (SoC) which combines all the electronic circuits of various computing elements onto a single integrated chip (IC). The SoC can have analog, digital as well as mixed-signal functions. The components of a SoC include a Graphical Processing Unit (GPU), a Central Processing Unit (CPU) and system memory (RAM). The CPU can be single or multiple core depending on the functionality and need. The design of a core basically consists of different parts. Each part has specific functions to carry out. Also, there are various bus interfaces, memory blocks as well as other functionalities. The related software components include real-time operating systems, device drivers and library functions of standard cells. All the characteristics of the standard cells are derived from these standard cell libraries.

Core is the heart of any microprocessor. A microprocessor can generally have multiple cores, which do the fetch, read, decode, execute and dispatch of microprocessor instructions. The processor SoC apart from core, also has an on chip high level cache, I/O and memory controllers and Integrated Graphics unit. Each SoC

and subsequently the core is designed to meet certain market requirements. These targets vary periodically depending upon the area of focus and market competition. Intel has always had upper hand over its competitor when it comes to performance in terms of frequency. However, there is a need to increase the power performance of Intel processors.

1.2 Approach

In this project, we are also aiming to improve the power efficiency of the device. To make design and performance analysis easy for us, it is necessary to divide the design into smaller parts. As a result, core has to be divided into clusters at top level. The clusters are nothing but parts of processor performing the instruction fetch and also its decoding. Some other major cluster functions include execution of instructions, operations on memory cache and interface. Different clusters are then split into sections. Details of the same are mentioned in 2.3. Sections are split out into number of units. Integer number execution, generation of address, floating point arithmetic, etc. are what is called units. Figure 1.1 shows how a microprocessor is divided.

The most basic cell in our hierarchy is what we call as the Functional Unit Block (FUB). Examples of FUB are multipliers, adders, dividers, repeaters and register files. These FUBs are designed individually first and are later combined in hierarchical way. All the designs and its implementation is done at this FUB level. The integration of various FUBs is done at the section level. A FUB is the smallest part of the microprocessor core that can be characterized by an RTL code. Since, FUBs are smaller in size they provide advantage in design convergence. The main advantages with functional unit blocks include reduced complexity of core design, reusability of blocks, ease of study in terms of frequency and power along with layout, noise and other quality aspects. Thus, we can easily take steps to meet the needed requirements.

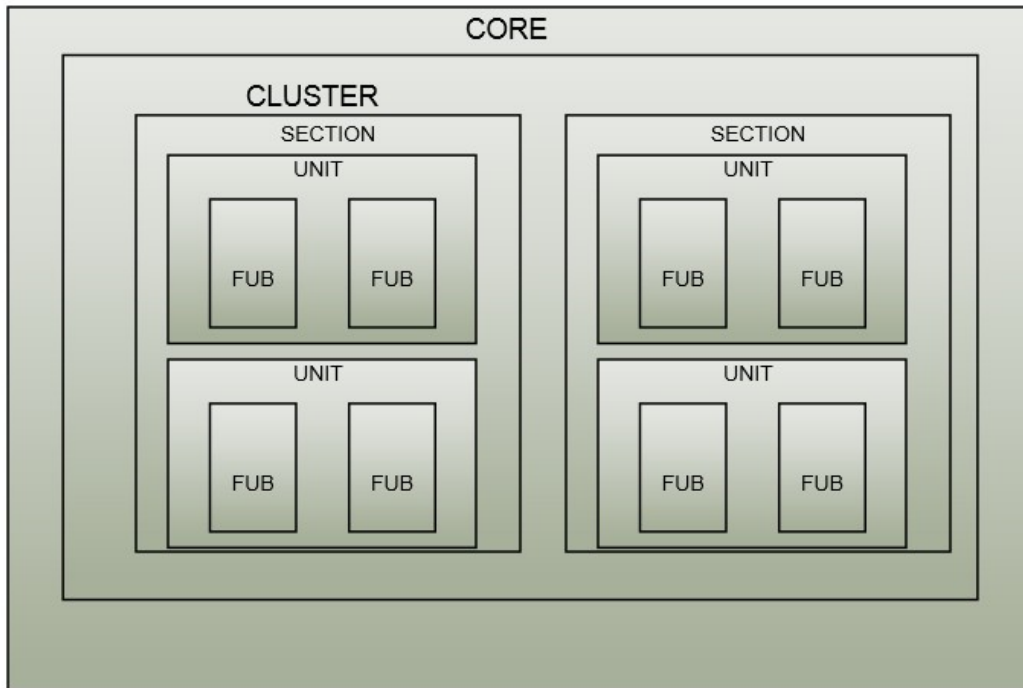


Figure 1.1: Overview of microprocessor architecture

1.3 Motivation

The evolution of high performance microprocessors was driven by the fact that we need high speeds to carry out high complexity real time functions and tasks. This led to tremendous improvement and optimization of microprocessors design. As a result, we now have semiconductor ICs that successfully combined various complex modules on to a single chip. However, with each technology node we are looking to improve and shrink these designs further and as the world needs speed, we need to further push the frequency of operations for our processors. Thus, in this project we focus on the ways to optimize our timing performance to give the required performance push to our device. To improve frequency we need to increase the switching activity in our design. Fast switching also requires cells which are faster. Also, a lot of delay adjustments need to be made in both the clock as well as data path to make sure that critical timing paths are not affected in our design

during technology shift. We use semi-custom design methodology to synthesize our designs.

Higher frequencies also require higher voltages to operate. An increase in voltage leads to an increase in dynamic power of the design. So, we also need solutions to improve the power index of our design. In this project we focus in this optimization of power also. We have strict limitations on power dissipation in portable electronic appliances which include smart phones, laptops and tablets. These requirements have to be met by the chip designers while still meeting the computational requirements of frequency and quality.

With the shrink in technology node, we have reached a position where power and area of the chip are now playing as a major factor which influence the design along with the frequency component. Of these, power consumption is also one of the bottleneck issue as power dissipated and consumed in the design determines the quality and life of the design. Extra power dissipated in the form of heat leads to reduced performance and durability issues of the design. Moreover, extra power consumption leads to extra budget impact. Motivation for the reduction of power consumption is different for different applications. In case of battery operated portable applications like mobile phones, the primary goal is to keep the battery life time high and packaging cost low. For performance portable systems such as laptops, goal is to decrease the power dissipation of electronic components to an extent which is half of total power dissipation. For the non-battery operated high performance systems the goal is to reduce the system cost while ensuring long term reliability.

The power optimization of high frequency designs require earnest efforts. This study focuses mainly on developing newer techniques for timing and power optimization by performing multiple tryouts on multiple FUBs.

1.4 Objective

Before starting with any optimization experiment, the basic objective is to understand the complete structure of a functional unit block. We need to check what are the power hog areas and the timing critical paths in the design. Sometimes the timing critical path can also lead to an increase in power. After analyzing the circuit, we need to understand how the power and timing can be optimized and what the different methods to be used. The objectives of this work includes,

- Design of high quality functional block that will be used in processor core
- Achieve equivalence of specified and implemented design
- Analysis and Optimization of power for the functional unit blocks
- Analysis and Optimization of timing to improve frequency performance
- Checking overall quality of the design

To get high quality performance from our design, it is necessary to make sure that various optimization techniques used do not have any negative impact on the overall efficiency of the design. As a result, we also need to make different quality checks in our design before delivering it to the customer.

1.5 Thesis Organization

Chapter 2 consists of literature survey where we discuss basic microarchitecture design of core. It also has research work on FinFETs. Chapter 3 has design methodologies that are used in core design. The flow of the design as well as principal of architecture are presented. Importance of each stage is also documented in this chapter. Chapter 4 and 5 discusses various optimization methods in relation to timing and power. Quality Checks is a part of chapter 6. The results are shown in Chapter 7. Chapter 8 concludes the work that has been done in this project.

Chapter 2

Literature Survey

2.1 Advancement in Microprocessor

In the early days of its origin a processor was to be found only in computers. It was a costly and complex device. Nowadays, although the complexity has increased, we can find small/big processors embedded in various devices such as cars, consumer electronic equipment, gaming equipment and many System on Chip (SoC) devices. Extensive use of microprocessors has led to revolutionary changes in its complexity, working, and process of manufacturing and pricing. The microarchitecture of a processor has grown continuously in previous couple of decades. The evolution of microprocessor is hugely based on Moores Law. Moores law states that the number of transistor on a chip should double every eighteen months. This simple statement has shown designers path to change transistor designs in many ways. From a hundred transistor on a chip, we have now reached a stage where a single chip has millions of transistors. All this has become possible due to scaling of devices and interconnects. Technology scaling includes reducing the dimensions of the transistor to increase the number of transistor within a less area. The performance of modern day processors is improving everyday thanks to the continuous reduction in chip sizes, and an exponential rise in the number of on- chip transistors. All the progress

in chip design due to Moores law, has wonderfully helped chip frequencies that have increased and the process costs that have been reduced. [1]

The constant increase in levels of operation is a constant source of motivation for rigorous performance improvement. This is achieved by integrating newer features in single chip after every process change. Such ongoing trends have inspired majority of semiconductor industry for long time as it allows designers to uplift the capability of CPUs, without having to add to the previously consumed power and utilized area of the design. The processors also incorporate newer functionality every new project, as per the feature of advanced applications which improve over time.

Its been 50 years since the development of first integrated circuits. Each year transistors, and as an extension the overall chip sizes are reducing in size to follow Moores law and boost the design performance. Largely based on technology scaling, total devices within a core has increased to millions in last few years. This also has an impact on interconnects. The length and the number of interconnects has also increased drastically in the last few years. In such increasingly complex and compressed integrated circuits, power consumption also thus becomes a bottleneck issue. Previously, designers focused only on the frequency aspect to improve performance of core. However, the research work now is also largely carried out to control power dissipation to improve power performance of the core.

The architecture of Metal-Oxide Semiconductor Field Effect Transistor (MOS-FET) is one of the major reasons behind the growth in the number of transistors on a single monolithic integrated circuit design. [2] The conventional planar device has only one gate. More advanced architectures include more number of gates and improved control of the charge in the channel. The Fin-type transistors (FinFETs) observed and studied in subsequently is one such example of multi-gate field effect transistors (MuGFETs), with the channel surrounded by gate from three sides. [3]

The flow followed in this project is known as RTL to GDS flow. The RTL coding is a part of Front end flow. Synthesis of the design using RTL as input and optimizing the design is what consists of Backend flow. The RTL converted

in its netlist is a composition of gates as well as interconnects. We optimized this netlist for timing improvements, area recovery and power optimization. The design methodology will be discussed in detail in chapter 3.

2.2 FINFETs

Being invented in 1947, the transistor has come a long way to revolutionize semiconductor industry. By 1958, germanium transistor were replaced by silicon transistors as they broke down at high temperature. [4] The next big thing in transistors came in the form of field effect transistors. Most modern day transistors are field effect, especially metal oxide semiconductor field effect transistors (MOSFET). Day by day the number of transistors on a chip is increasing. Following Moores law, they have continued to shrink in size and improving in performance. Scaling has helped to achieve the required size and performance for the transistors.

Conventional MOSFETs are becoming increasingly difficult to scale because of the drawback like the short channel effect. [5] Industries are now looking at other devices to substitute MOSFETs are have already started moving towards it. One such device is the FinFET which Intel uses for fabrication. Fin-type field effect transistors (FinFET) are promising substitutes for traditional MOSFETs at submicron technologies, as scaling sown of traditional devices leads to lower performance and higher power consumption. [6] The table shows comparison between a typical MOSFET and FinFET.

| Architecture | Planar CMOS | FinFET |
|--------------|-------------|----------|
| I_{on} | 1100 A/m | 550 A/m |
| I_{off} | 2.00E-8 A/m | 1E-9 A/m |
| V_{tsat} | 0.34V | 0.22V |

Table 2.1: Comparison between CMOS FinFET

Owing to the fact that fabrication is same as MOSFET, FinFETs can easily substitute MOSFETs. FinFETs are double gate transistors and are non-planar unlike conventional MOSFETs. FinFET gates are shown in figure 2.1.

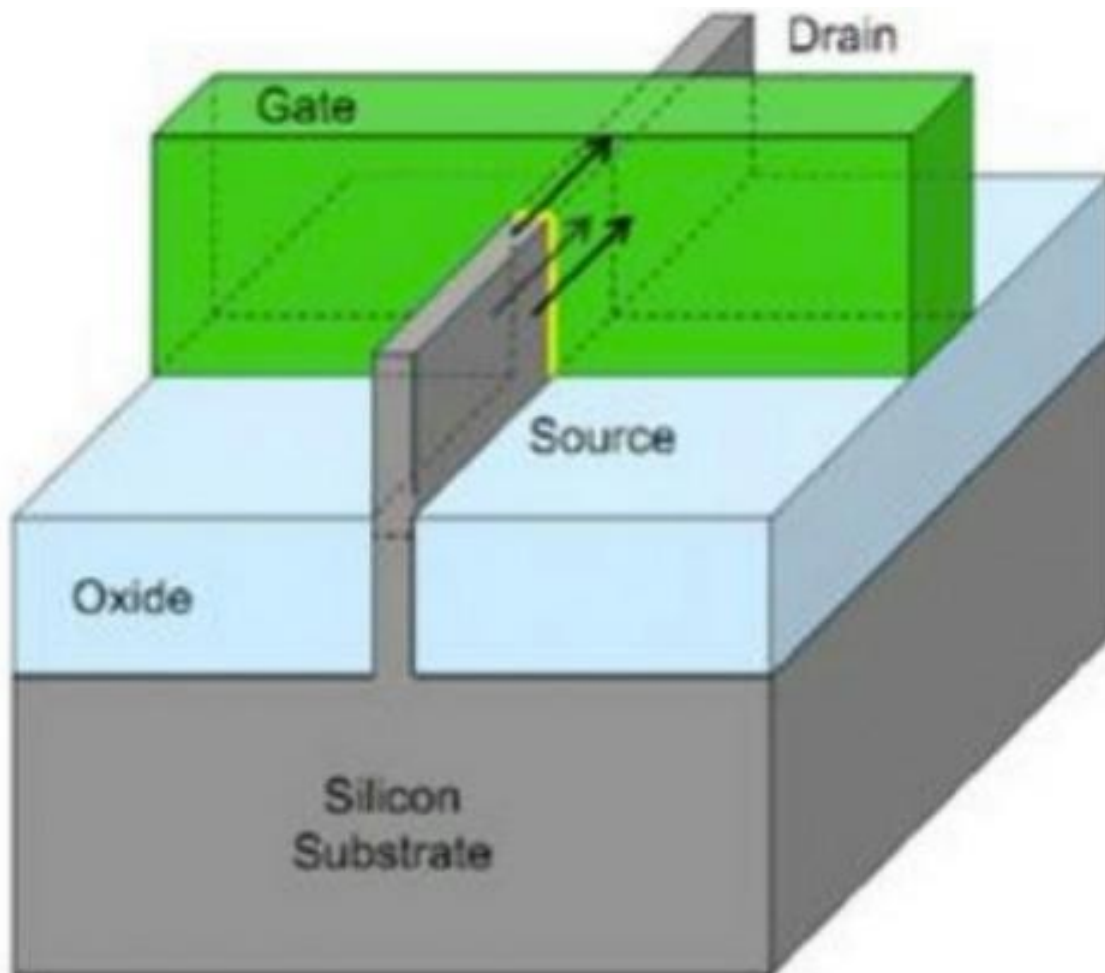


Figure 2.1: FinFET View

They operate in two modes known as Independent Gate mode (IG) and Shorted Gate mode (SG) as seen in figure 2.2. The two gates and thin silicon body can easily suppress the short channel effect. [7] Since, both the back and front gates can be controlled independently as well as together, FinFETs can be used for increased performance by reducing the leakage current and power dissipation.

However, there are also a few challenges faced with such transistors. Cross-wafer non uniformity, difficulties in manufacturing the device, high capacitances as well as

resistances and self-heating are few of the issues. Self-heat is the buildup of heat in the channel at high levels of voltages. It is caused by low thermal conductivity of its constituting materials viz., SiO₂ and SiGe, higher densities of current and reduced dimensions of these devices. The scaling of device dimensions is also related to an increase of temperature in active region of the device directly. [8]

Intel uses triangular FinFET structure as it increases switching speed. As the gate is covered on three sides, the parasitic capacitance of FinFET is more compared to MOSFET. This increased parasitic capacitance can bring more noise thus making it disadvantageous to use in analog circuits.

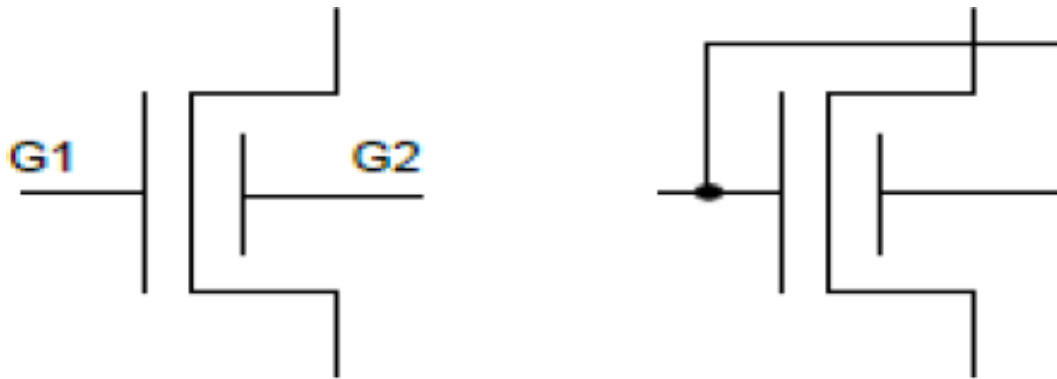


Figure 2.2: Two modes of FinFET

2.3 Design Hierarchy

The ever reducing transistor sizes has pushed the development in VLSI Circuit Design and also became the biggest motivation for multi-core processor designs. In place of single core, the technology evolution has allowed multiple cores to fit in the same area which has given scope for superior performance and enabled the integration of a rich set of features onto a single chip. For our core design, we will be following hierarchical design approach[9] in this project. Representation of hierarchical approach is shown in figure 2.3

The microprocessor core has multiple clusters for performing different operations.

Some of these clusters are Execution, Fetch, Decode, Caches, etc. Clusters division leads to formation of multiple sections. Sections are sub parts of cluster. One cluster can have multiple sections. Each section performs a smaller operation of a major operation. Integer number processing, processing of floating point, instruction queuing for out of order instructions are few examples of sections amongst other that present in different clusters. Sections are further split into different units. The smallest entity in hierarchy is called a Functional Unit Block (FUB). It is the most basic block of the hierarchy. Few examples are multipliers, dividers and adders, register Files, repeaters, receivers etc. The designing and execution is done at FUB level while the integration of different FUBs is carried out at the section level. [10]

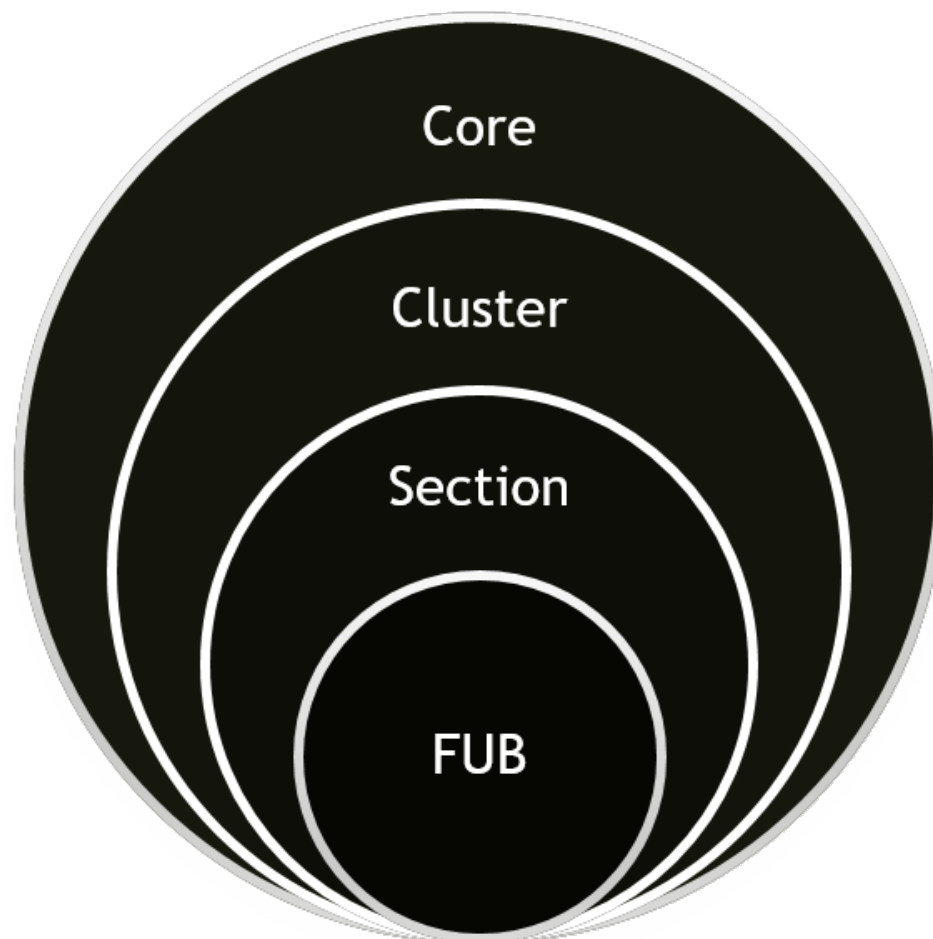


Figure 2.3: Hierarchical approach of microprocessor core design

Physical design includes designing of FUBs, equivalence verification, performance optimizations, and various quality checks. FUB designing in Intel follows custom design approach and uses standard cell libraries. As FUBs are the basic building blocks of the core its very necessary that they are efficiently optimized with respect to timing, power and quality checks. [11] As a result, major amount of time is spent on the effective designing of these FUBs. Once we are able to optimize the performance of a FUB entirely, integrating them to simplify convergence at section level becomes less tedious. This approach has helped the VLSI design industry towards major improvement in core functioning.

Increasing functionality in a limited area with every new technology makes it difficult to optimize the performance of core. Along with timing, power optimization is also becoming a major reason of worry nowadays. This study mainly focuses on boosting timing performance and developing new techniques for power reduction .We also focus on the overall quality of the design by using a few quality determining parameters. They help to improve the reliability and superiority of our designs.

Chapter 3

Design Methodology

3.1 Design Flow

The design methodology shown in Figure 3.1 is the flow we use to reach till our final design. Typical Functional Unit Block has three ways of representations. The first representation is RTL, where Hardware Description Language (HDL) is used. Functionality of the block is defined at this stage along with logic verification. Secondly, netlist is converted from RTL. The netlist contains information on the cells used, their interconnections, area used, and other details. It is used for timing analysis, verification and optimization of circuit. The third kind of representation is layout, where the schematic is converted into geometric representation of cells and interconnects. It creates masks for silicon.

The first stage of flow is giving correct input collaterals. The input collaterals here means the RTL code, the timing constraints, the layout constraints and the standard cell library. The designer describes the functionality of the design using high level HDL using RTL constructs. The input of wrong collaterals can lead to incorrect optimization and logical failure of the design. The design must meet certain timing requirements and we use the concept of Static Timing Analysis (STA) for performing the timing checks. Area of particular FUB should not exceed a certain

limit. This information is present in layout constraints. The technology library will be having basic logic gates such as NAND, NOR among other complex cells. It may also contain particular instruction cells like adders and MUXs and flip-flops. The technology library is the one that defines the functionality of mapped cells prior to optimization. The physical design rules and physical views of standard cells are also a part of the library.

Like mentioned earlier, with timing, enhancing power performance is also a part of optimization. Power estimation and reduction features are enabled based on the power scenario based tests. By using different power scenarios, the power for each unit in the core is estimated and analyzed. Areas where power is wasted in each unit is determined by analyzing these scenarios. Using this analysis we then determine measure to reduce them. Both static and dynamic power could be a matter of concern for us. As power becomes a bottleneck issue, optimization post synthesis only will not help. At such times, RTL debugging also needs to be done. It can include analysis and restructuring of clock network, data path and changes in memory architectures as ways to mitigate power dissipation and keep it in check

Thus using various methods mentioned in subsequent chapter we can improve the quality of our design.

3.2 RTL Coding

RTL coding is the depiction of the functionality of the design in terms of Hardware Description Language. This will be later converted into the language which can be understood by the machine for netlist generation. The descriptive language used here for RTL coding is System Verilog. This RTL code is in the synthesizable form and forms a structural netlist. The code follows bottom to top approach just like the design approach. First the code for lower level blocks is written and the all the blocks combined together form a top level block. This procedure is called as front end design. It deals with the coding of the circuit functionality and its optimization

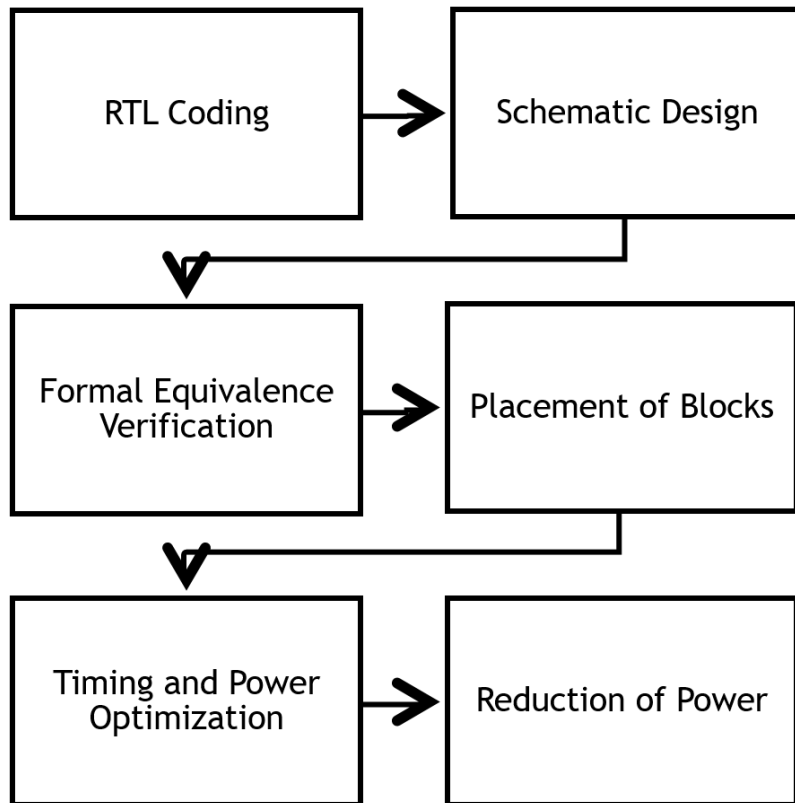


Figure 3.1: Design Flow

in order to get a logic with reduced number of components. However while possibly reducing the number of components in the code it has to be kept in mind that the block will remain functional as per the requirement of the design. This code behaves as the input to the Backend stage.

3.3 Schematic Design

There are basically two types of design methodologies that are used to synthesize the designs, each of them is discussed below

1. Computer Aided design
 - Circuit synthesis and physical design by tools
 - Tool places and routes the design as per optimization needs

- Tools used : Design Compiler, IC Compiler by Synopsys
2. Custom Design
 - Circuit is synthesized by designer manually
 - Gives flexibility to build hierarchical design
 - It is used particularly where timing and interconnects delay need to be taken into account

3.3.1 Importance of Custom Design

Manual designs can be a comparatively easier substitute where each design template is repeated multiple number of times. Since the designer has to implement the circuit manually, implementation of larger block can be a harder task. Manual implementation is done for blocks where RC delays of interconnects is appreciable. Placement of cells during synthesis is largely based on RC estimation of interconnects. Since the EDA tools are not capable to completely estimate the RC delays at synthesis stage, we might not be able to get correct placement of critical drivers and receivers within a particular path. As a result, if the design is more interconnect delay dominant it will lead to sub optimal performance of the design. This makes it necessary to manually handle placement and routing in some of the blocks in our core design.

3.4 Equivalence Verification

Before starting timing convergence or any other quality checks, first and the most important stage is to verify that the design is working functionally in the same way as required. For assuring correct functionality Formal equivalence check is performed. This stage is known as Formal Equivalence Verification (FEV). In this stage, shown in figure 3.2 the RTL is compared with the netlist implemented by the designer.

This helps in finding logical errors in the design implemented by the designer. It makes sure that the design is bug free in the initial stage itself to avoid problems on silicon. It is most accurate and time saving way to validate any changes or edits done in design.

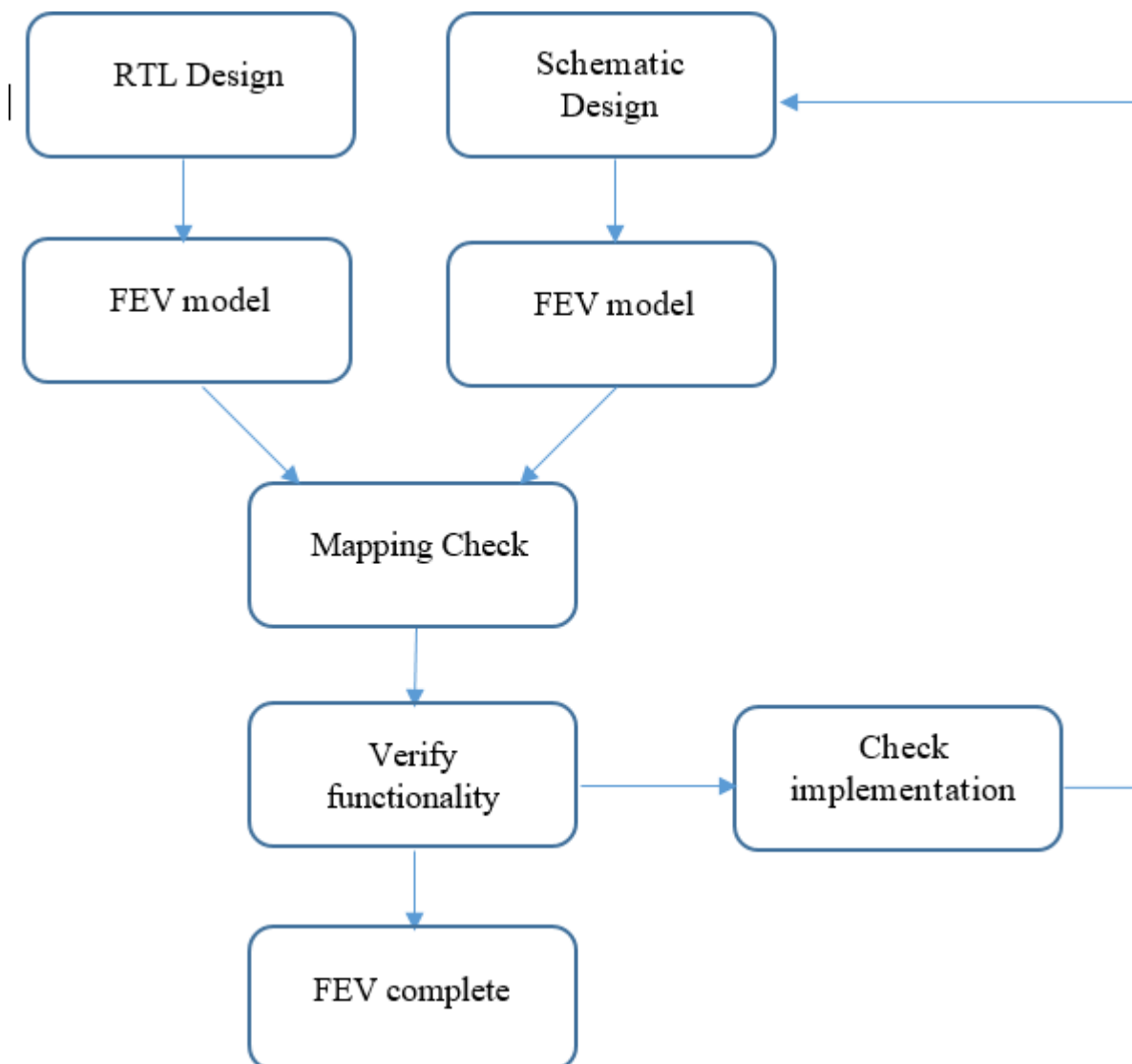


Figure 3.2: Verification Flow

Equivalence verification provides multiple ways to compare designs. We can compare the following forms of design:

- Schematic vs Schematic

- RTL vs RTL
- RTL vs Schematic

Figure shows high level flow of FEV. First stage involves conversion of RTL and the implemented design into FEV model format. For comparing RTL and implemented design, it is necessary to convert both formats of design into same format. Both designs are converted in FEV model first and then compared accordingly. This makes comparison easy and possible between RTL and implemented design.

Prior to comparison we also need to make sure that all the primary inputs and outputs of both models are mapped properly. These are known as mapped states. Mapped states are nothing but the compare points for the verify stage. A compare point is nothing but the endpoint of the combinational logic during verification. The compare/mapped points can be input or output of a sequential, an output port or black box input pin.

There are some necessary points that needs to be mapped essentially in the design:

- Principal map points like primary inputs or outputs and states of sequential
- Secondary mapping points like black boxes and dangling nodes

After successful mapping, we check whether the input and output pins of the black box in both the hardware models are matching or not. Map files are used to add and remove mapping if needed. Figure 3.3 shows how mapping is done and how the verification points are added to check the logical equivalence between RTL and implemented Design. The sequential output nodes will be mapped and a curt point will be created in this example. In verification stage we will verify if the functionality of both designs is same. The FEV results will as per figure 3.4.

As it can be seen, two cones are equivalent and one cone is different in logical equivalence. Thus FEV helps us in identifying logical difference between expected and actual design.

With the proper use of formal equivalence verification, any incremental changes in the design could be verified very quickly. It is possible because FEV compares the designer implemented functionality with the one that is extracted from the RTL.

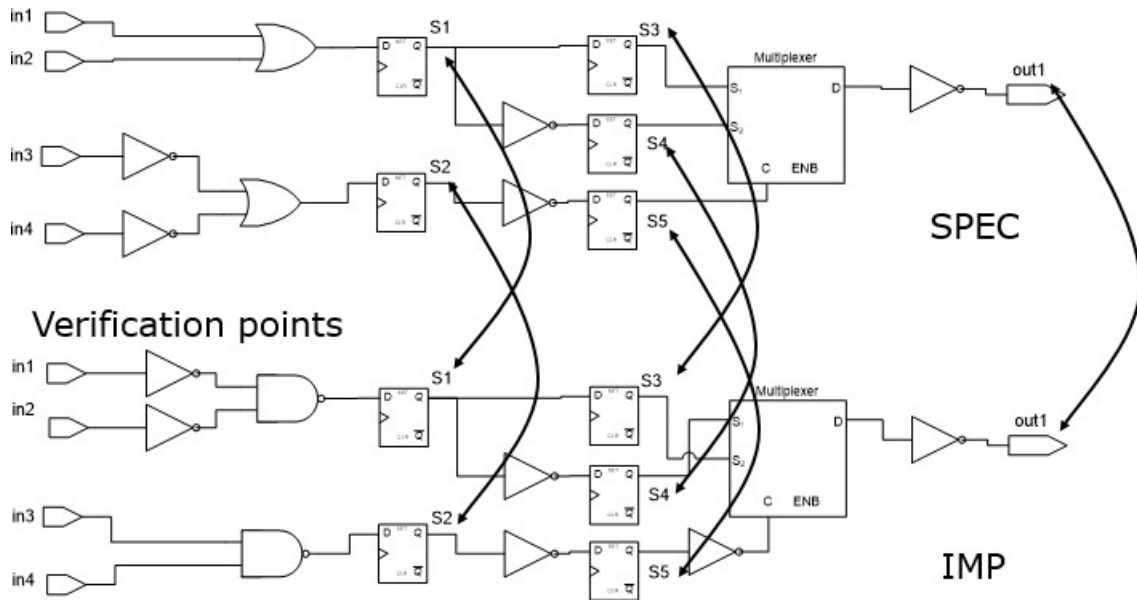


Figure 3.3: Mapping in FEV

In classic FEV flow assumptions are taken from SPEC model. In general, only assumptions that are driven combinatorially by mapped signals can be used for the verification. Other assumptions are ignored. There are two types of matching State matching and Non State matching:

1. In State matching, all the assumptions can be used for the verification since all the states are mapped.
2. In Non State matching, only assumptions that are driven combinatorially by mapped signals can be used and assumptions that are combination ally driven by (atleast one) non-mapped state element are ignored.

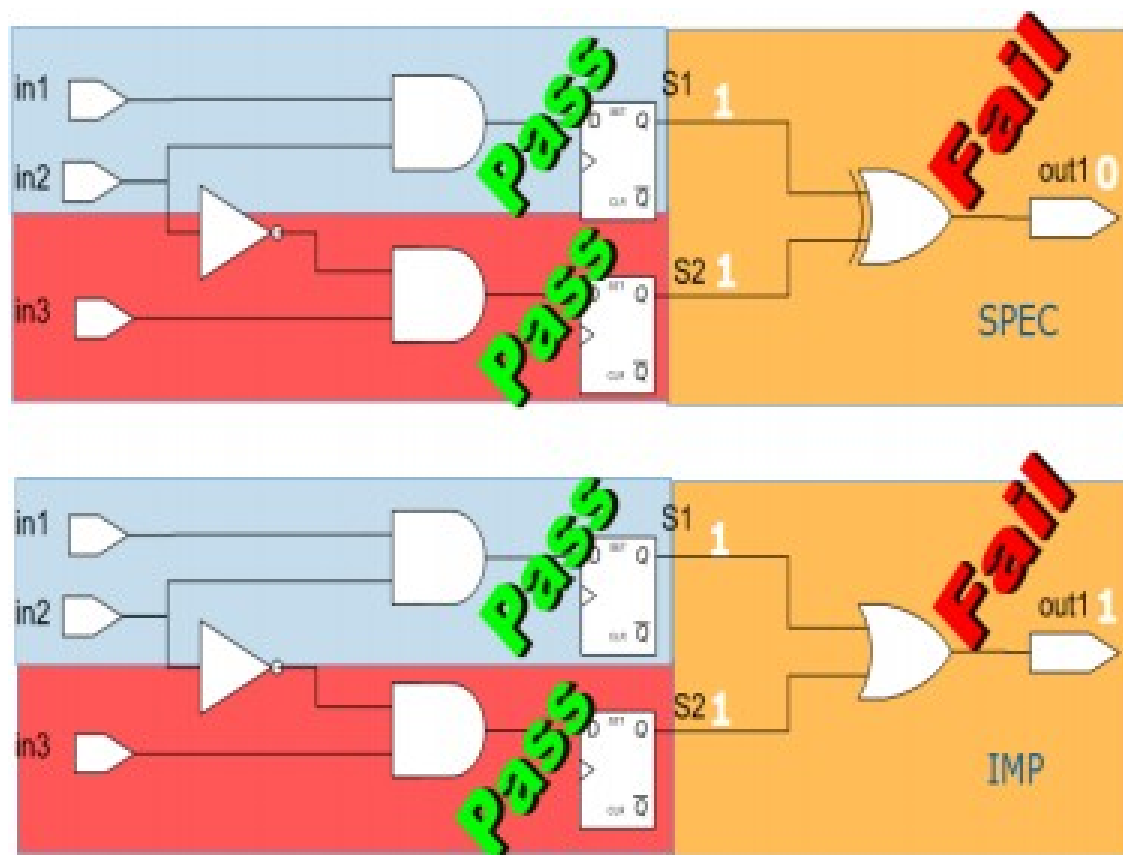


Figure 3.4: Verification of Logic

3.5 Placement of Blocks

This stage involves floor planning which decides the dimension of the block. It also consists of placement of cells and their routing. Gates which are a part of the netlist are actually implemented and then routed in the design. It also comprises of placement of ports and power grid formation. Clock tree synthesis and its proper placement is also done here. Detail routing, overlap removal are all done in this stage. This stage converts a schematic design into its physical implementation form.

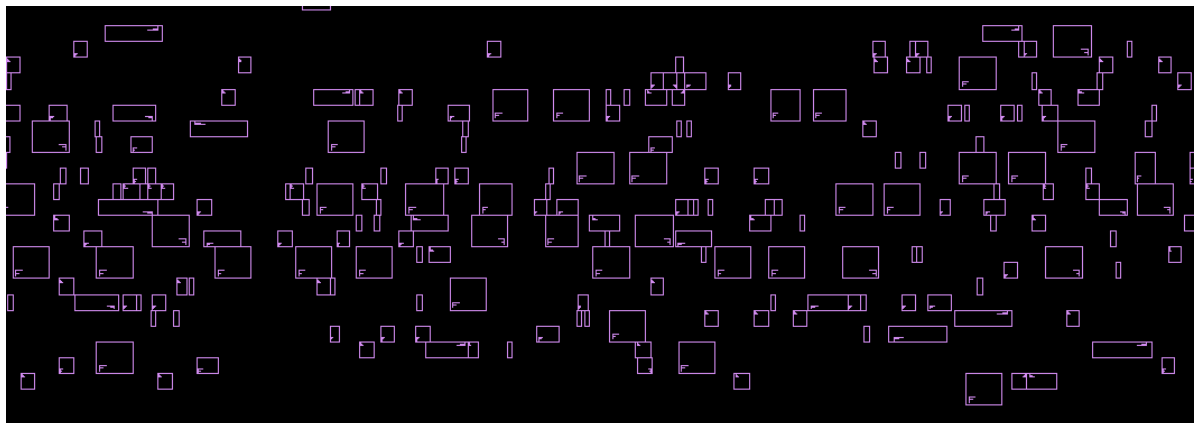


Figure 3.5: Placement of Cells

In this stage, we manually doing placement. In placement of blocks, we put the drivers and receivers as near as possible. So, the driver can easily drive the receiver. If driver and receiver are far away so we add buffer between them and put that buffer to the middle of the driver and receiver.

Figure 3.5 shows the placement of the cells inside the Fub. We placed cells as they can not overlap each other. After doing Placement the routing is done between the driver and receiver. The net between the driver and receiver is give the interconnect delay of the FUB.

Figure 3.6 shows the routing of interconnects. Distance between the cells decides the metal layer of the routing. If the driver and receiver are placed near to each other, lower metal layer is used for routing. If driver and receiver placed far away to each other, higher metal layer is is used for routing. So basically metal layer decides the how far driver and receiver is. In FUB, basically metal 1 to metal 5 we can use for routing and in section, metal 6 to metal 11 is used for routing purpose. figure 3.6 shows the routing of the interconnects. The interconnects also include in timing path. we will see in next ch how can we calculate the timing.

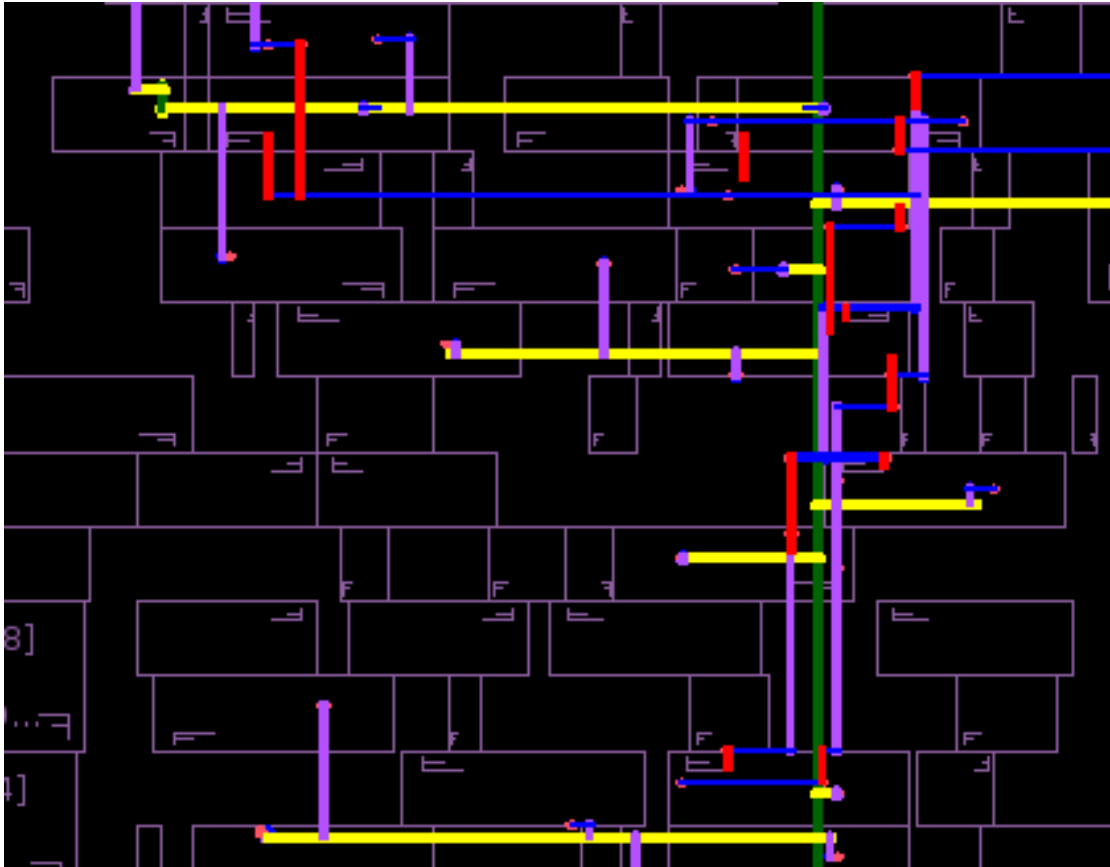


Figure 3.6: Routing of Interconnects

Figure 3.7 shows the Distribution of metal layers, it shows shows distribution of metal layers in various layers. The lowest metal layer is metal 1 , which is in red color, blue color shows the metal layer 2. Similarly, purple,yellow and green shows the metal layer 3, metal layer 4 and metal layer 5, which is shown in the figure.

the next section includes the timing and power optimization of the FUB, and how can we improve the quality of the FUB for the different projects.

3.6 Optimization

Performance optimization can be done from both timing as well as power perspective. Timing verification of the circuit is normally done using timing analysis. There are two ways of carrying out analysis namely Static Timing Analysis (STA)

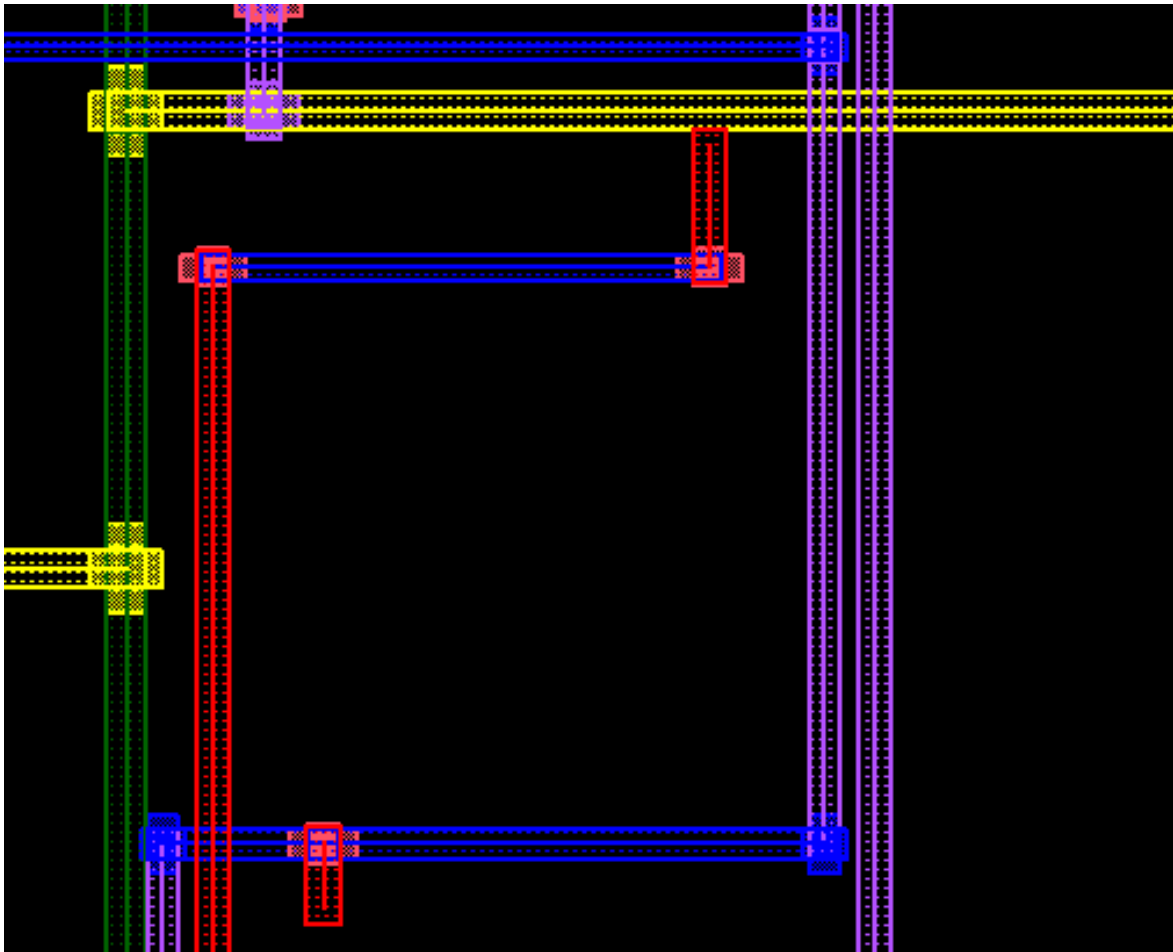


Figure 3.7: Distribution of Metal Layer

and Dynamic Timing Analysis (DTA). Details of static timing analysis are studied in chapter 4.

Similarly, we have to optimize power consumption to improve power performance of the FUB. Historically, power in the highest performance chips has increased with each new technology node. Already, total power consumption in microprocessors present a substantial problems when servers are concerned. For the server farms, the power and cooling costs can be equal to the cost of the computers themselves. Focus has to be thus on both dynamic and leakage power of the FUB.

3.7 Quality Checks

Along with the timing and power optimization, it is extremely important to check for the overall quality of the design. The quality of our design determines the reliability, age and functionality under various working conditions. Before sending processors to fabrication, we have to make sure that it meets the quality standards. Reliability Verification (RV) is one such parameter which is helpful in determining the quality of the design. The details of RV are discussed in chapter 6. Only after taking care of all the RV violations, and meeting them we are able to say that the design quality is good.

Chapter 4

Timing Optimization

4.1 Static Timing Analysis

Static timing analysis (STA) is a simulation method of computing the expected timing of a digital circuit without requiring a simulation of the full circuit. STA ensures that the timing paths within the system satisfies the timing constraints at various operating frequencies and operating voltages. STA analyzes all paths from start point to endpoint which are present in design. It then compares the paths against constraints which are defined for individual path. All paths are constrained by the definition of period of clock and the timing characteristics of the primary inputs and outputs of the circuit. The timing paths can be divided as data path, clock path, clock gating path and asynchronous path. For a data path the start point can be either input port of the FUB or clock pin of the flip-flop/latch and end point can be data input pin of the flip-flop/latch or output port of the design. The types of path are shown in figure 4.1 . Table shows the start and endpoint of different types of path. [12]

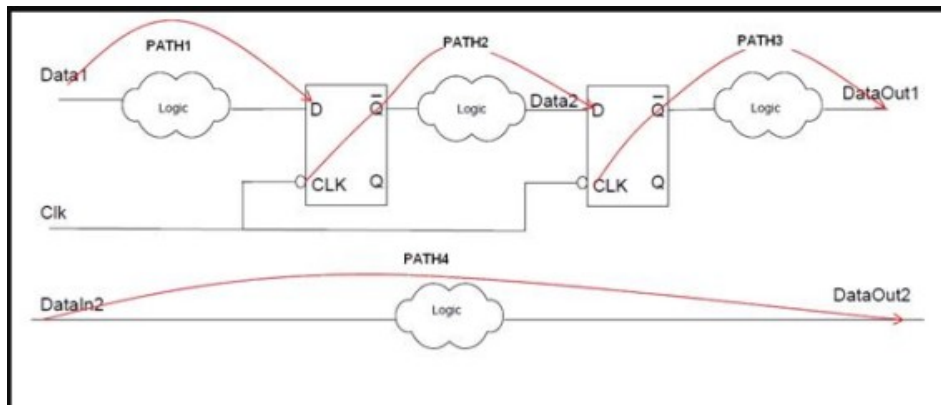


Figure 4.1: Types of Logical Paths

| Path | Type | Description |
|-------|-----------------------------------|--|
| Path1 | Input pin/port to flip-flop | Starts at input port and ends at data input of sequential element |
| Path2 | Input pin/port to output pin/port | Starts at the clock pin of sequential element and ends at the data input of sequential element |
| Path3 | Flip-flop to flip-flop | Starts at the clock pin of a sequential element and ends at an output port |
| Path4 | Flip-flop to output pin/port | Starts at an input port and ends at an output port |

Table 4.1: Path Type and Description

4.1.1 Setup Time

Setup time is the minimum amount of time before the clocks active edge that the data must be stable for it to be latched correctly. If a timing path is satisfying setup time constraints, it ensures that the data launched at previous edge is captured properly at the current edge.

4.1.2 Hold Time

Hold time is defined as the minimum amount of time after the clocks active edge during which data must be stable. Each sequential element needs some time for

data to remain stable after clock edge to reliably capture data.

4.1.3 Setup Violation

Consider there is a setup timing violation for the data path (Sequential to Sequential) shown in figure 4.2. Here, clock arrives simultaneously at both sampling and generating FFs. A setup violation happens when the data generated at a clock rising edge fails to arrive at the sampling FF within one clock cycle. i.e., the path fails to satisfy the following expression,

$$T_{clk} > T_{ff1} + T_{comb} + T_{setupff2}$$

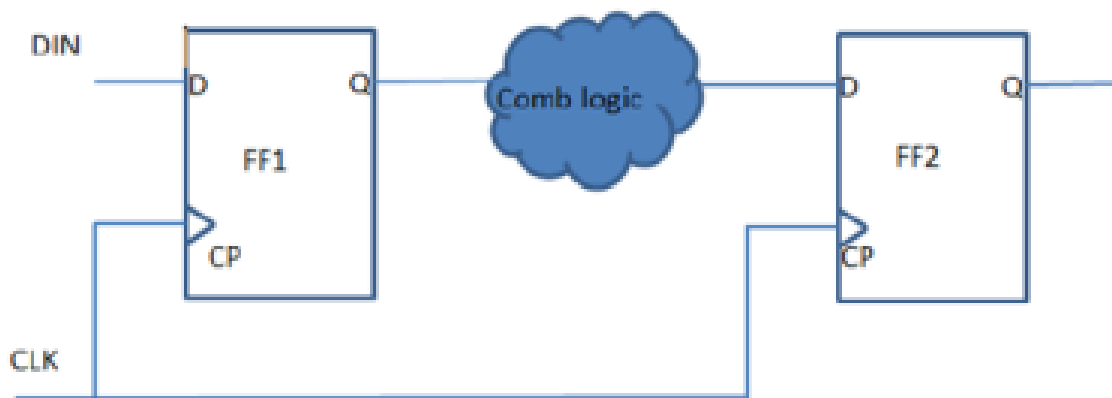


Figure 4.2: FF-D-FF Path

The setup violation can be fixed by:

- Clock Tuning:

Delaying the clock (pushing) to sampling FF or fastening (pulling) the clock to generating FF can relax the timing constraint by providing more time (T_{clk}) for the data path to complete.

- Fastening the data path:

Data path can be made faster by increasing the driving strengths for the standard cells in the data path. Upsizing will decrease the cell delays and also

improve the output signal slopes.

4.1.4 Hold Violation

A hold violation happens when a timing path fails to satisfy the following expression:

$$T_{ff1} + T_{comb} > T_{hold} + T_{skew} + (T_{genclckedgearrivaltime} - T_{smpclckedgearrivaltime})$$

Hold violations can be fixed by:

- Clock Tuning:

Pulling the sampling clock or pushing the generating clock can relax the hold time constraint.

- Delaying the data path:

Data path can be delayed by adding buers to the path or by replacing normal voltage cells with high - V_{th} cells. Such cells have larger delay due to reduced voltage swing.

4.1.5 Setup Fixing

Setup time is the amount of time that the data needs to be held stable before the active edge of the clock. If the data doesnt arrive before the rising edge of clock it may lead to setup time violations. There are solutions to fix the setup time in the timing violated paths. They are listed below.

- Upsize the cell to improve driving strength and cell delay
- Split the driver which has high fan-out
- Upgrade the metal wire for the long routes
- Pulling the launching clock or pushing the capturing clock
- Choose best placement to improve net delay

- Restructuring the logic
- Use low Vt cells.

flow chart for fixing the setup violations is presented in figure 4.3.

4.1.6 Hold Fixing

Hold time is considered as the amount of time that the data needs to be held stable after the rising edge of the clock. If data is not held stable, violation exists that may lead to functionality failure of the design. To ensure correct data synchronization, hold time fixing is extremely necessary. It serves as the final step of timing closure for chip design. Hold time violations can be fatal for our design. As the physical design flow progresses, the timing information becomes more accurate. So, the repair performed at early stages is insufficient at later stages. We usually perform hold fixes at later stages compared to setup fixes.

There are many methods to fix hold time violations that occur in synthesis flow. The most popular method is to insert buffer. By inserting buffer the delay increases across the path, hence the time margin moves towards the positive side. As the time margin becomes more positive, there will be enough hold time available for the data to become stable, hence the hold time violation can be resolved.

The algorithm for fixing hold violations can be seen in figure 4.4. Various methods are used to improve hold violations. Some of them are:

- Addition of Buffer to increase data path delay
- Push the launching clock or pull the capturing clock
- Downsize the cell to add delay.
- Swap low Vt cells with high Vt cells

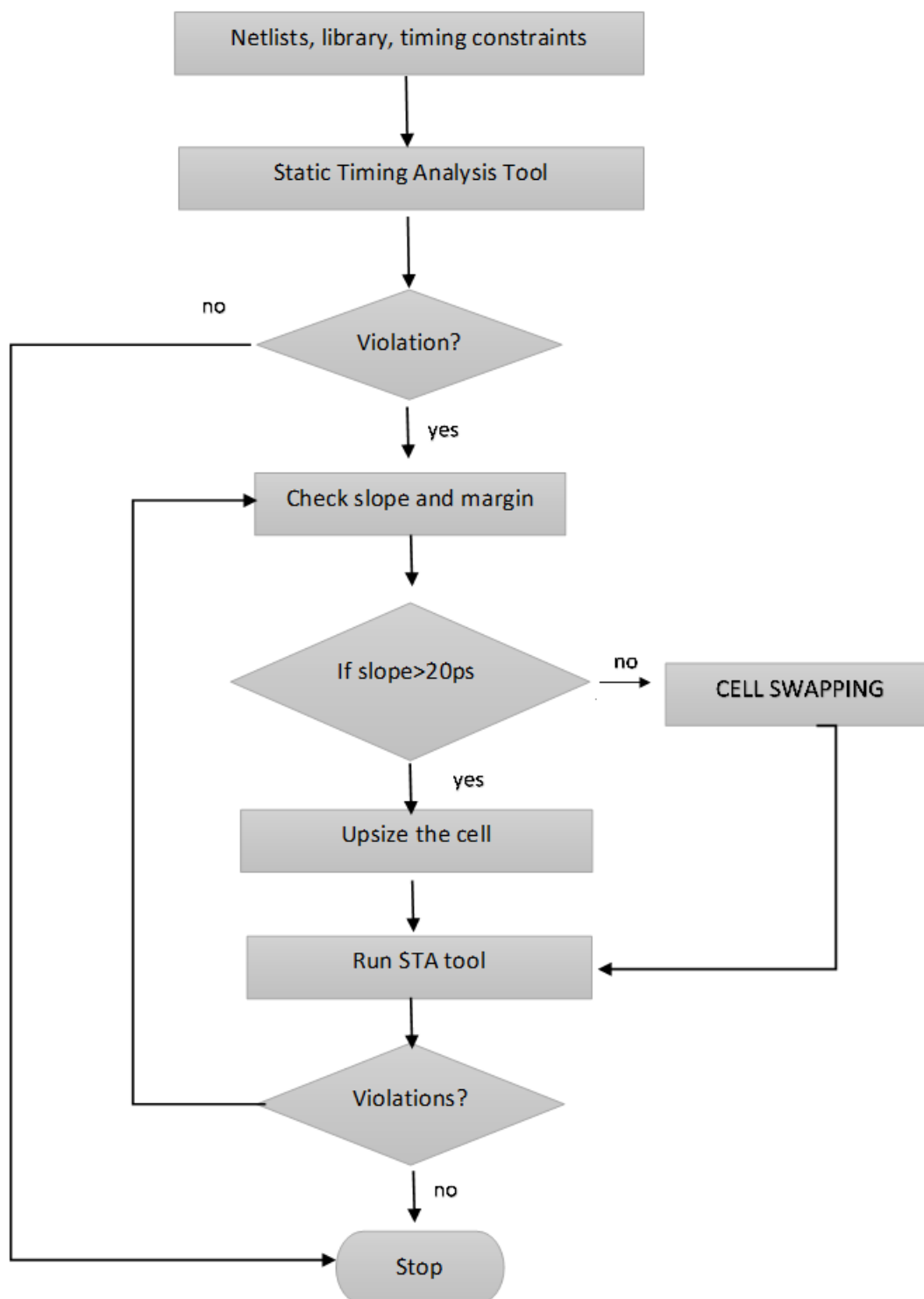


Figure 4.3: Flow Chart for Setup Fix

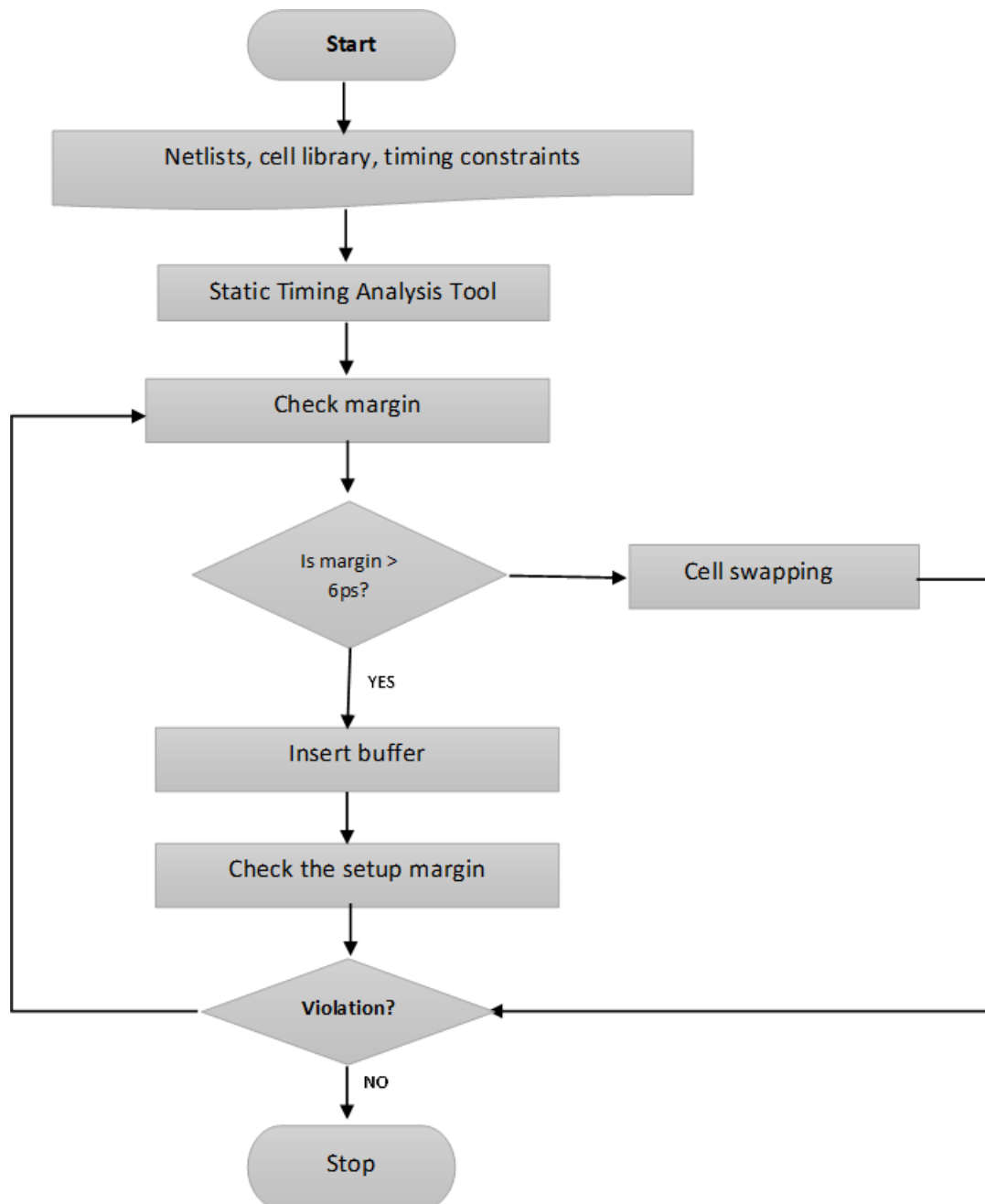


Figure 4.4: Flow Chart for Hold Fix

Chapter 5

Power Optimization

5.1 Power Optimization in Circuit Design

Each project will have a target of optimizing power numbers based the previous project. This work centers on reduction of power using various power optimization techniques. These power reduction technique are been experimented for the first time in the design of processor at Intel. The total power dissipated in a processor consists of active power as well as leakage power.[13] While active power includes the ON power and the dynamic switching power, leakage power is the power utilized when the design is in OFF state.

This work focuses on the reduction of active and dynamic power of the design. Resistance of the cells is a major contributor to the active power. The dynamic power is given by equation.

$$P_{\text{dyn}} = A.F * C_{\text{dyn}} * V^2 * F$$

Where,

A.F is the activity factor

C_{dyn} is the dynamic capacitance

V is the operating voltage

F is the frequency of operation

For each project, operating frequency and voltage is fixed. A.F is a constant quantity depending on the design. So one way of reducing is reducing the dynamic capacitance of the block. The optimization methods mentioned in this work, also concentrate on reduction of dynamic capacitance and the resistance to reduce the active power of the circuit.

While each technique gives power advantage, it might also affect the timing performance of the design. So care needs to be taken to not overdo any changes and fix the timing lapses caused by our changes.

5.2 Power Driven Placement

Power driven placement also known as Low Power Placement (LPP) focuses on reducing the activity factor (AF) of the nets which are longer in length. LPP reduces net switching power using switching activity based power-aware placement technology. Figure 5.1 shows how LPP reduces the net length to reduce the AF of longer nets.[14]

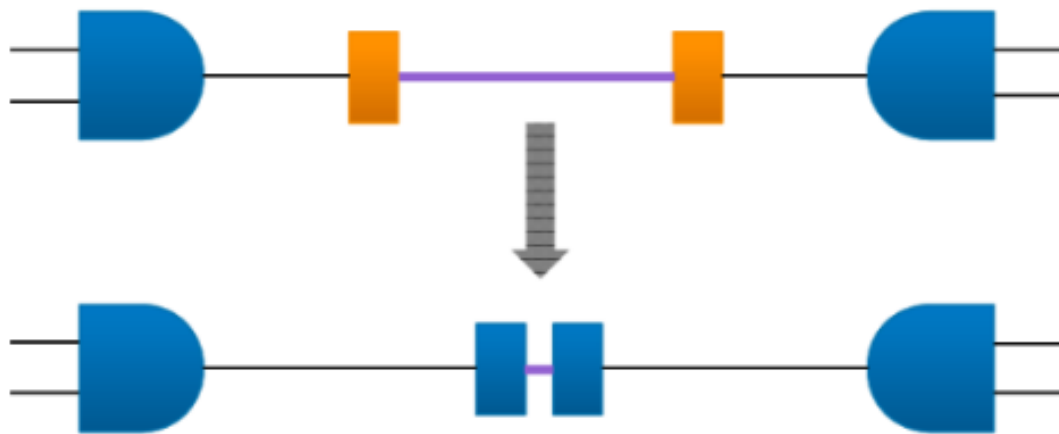


Figure 5.1: Reduction of net length in LPP

Net switching is proportional to the multiplication of the activity factor and the capacitance of the net. In order to reduce the total switching power, power driven placement reduces the wire length of higher activity nets. This reduces the dynamic capacitance of the higher activity nets, leading to smaller total switching power. The design is also constrained in area which leads to lesser logic levels and thus even lesser switching.

5.2.1 Multi Vt Synthesis

Threshold voltage V_t is defined as the gate voltage at which the device starts to turn on. This voltage plays a very important role in determining the power of the design. Usually, in VLSI design multiple values of threshold voltage are used to make a cell. The different values of threshold are classified as:

- Low Threshold voltage (0.6 V)
- Medium/Standard Threshold voltage (0.7 V)
- High threshold voltage(1.1 V)

Each threshold level has its impact on the performance as well as power of the design. Low threshold voltage cells consume less power but are more susceptible to variations. Also, low V_t cells have more leakage power. High V_t cells on the other hand consume more power but are less susceptible to variations. Also these cell have low leakage power.[15]

Multi V_t synthesis makes use of this property to build optimal design. We use different V_t cells in clock path and data path. Clock cells always have to be standard V_t as they have less variation and consume less power. Figure 5.2 shows how multi V_t synthesis is done.

We replace all cells in data path by HVT cells. HVT cells have less resistance compared to SVT cells and thus consume less power. It was also observed that it reduces logic level in design.

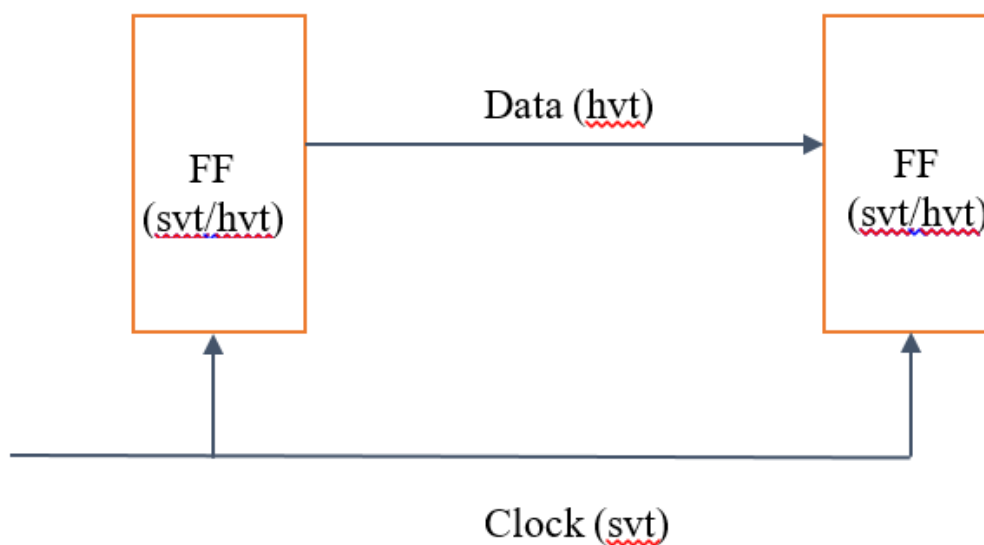


Figure 5.2: Multi Vt Synthesis

5.3 Multi Bit Synthesis

Multibit synthesis is a method to not only save power but also save area of the block. The principle of multibit is the conversion of neighboring single sequential cells into a dual sequential or quad sequential. If more than one flops are in the same neighborhood, they can be combined or clustered to prevent giving different clock to each of them. This saves power as most power is consumed by clock. Clustering can be done during placement as well as routing the design. Figure 5.3 shows how clustering is done.

The task here is to identify which sequential cells to be clustered. Over conversion can degrade the timing of the design. Some criteria for sequential clustering are:

- They should be driven by the same clock net
- They should be in the same scan chain
- They should be driven by the same common control signal net

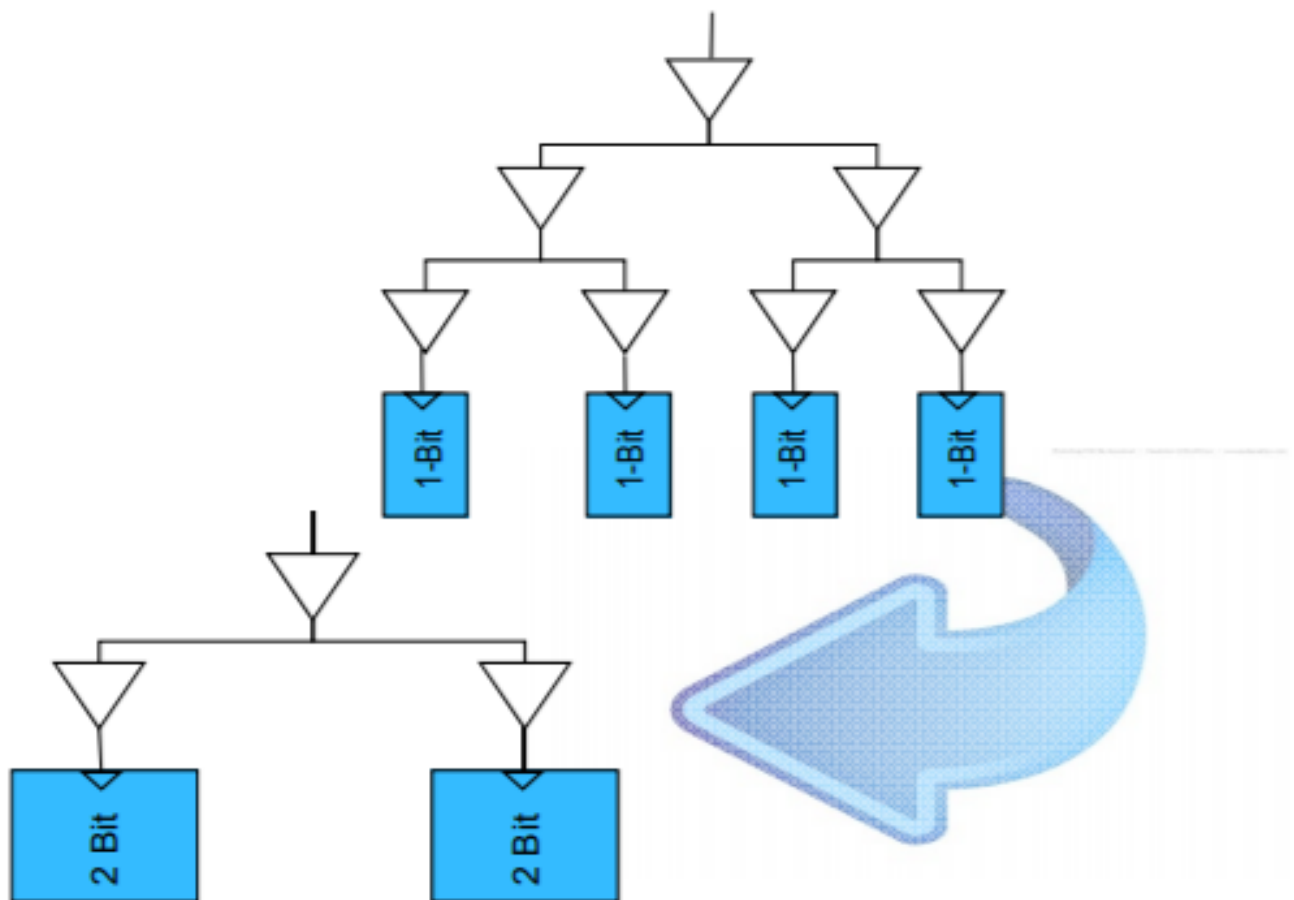


Figure 5.3: Multibit conversion

Advantages of Multibit Synthesis are

- Reduced total clock pin capacitance
- Buffer Clock Skew
- Low Dynamic Power

5.4 Big Sequential Cells

Sequential cells are power hungry cells as they have clock associated with them. Reducing their converts to reduction of capacitance at the output. If in a particular case sequential cells require high driving capability to drive huge fan out; addition of

buffer at the output of the downsized sequential is the favored solution. These type of cell are by default present when a design is synthesized as they have good drive strength. They can be removed and replaced by smaller cells which will consume less power but will also have low driving strength. We can replace it with a small size sequential in the design with a buffer at the output of the sequential, which helps in maintaining the desired slope at the output, which helps in reducing considerable amount of power.

5.5 Other Techniques

Various other techniques can be used to get desired gain in power. Some of the techniques are:

1. Clock Gating:

Clock gating saves power by adding more logic in the circuit. When the circuit is on, the enable is 1 and clock is applied to the circuit.

2. Logical optimization:

The power is too high, when the logic is large. So we optimize the logic to reduce power.

Chapter 6

Quality Checks

6.1 Reliability Verification

Reliability is as much a key to success in the microelectronics industry as is performance. Not only must a product perform as desired, it should also work for an extended period of time. A microprocessor, which boast on performance, will always be unacceptable if it fail to provide long term reliability. As integrated circuits progress and become more complex, the individual cells in the design must become highly reliable if the reliability of the whole design has to be increased. However, due to continuing miniaturization of very large scale integrated circuits, interconnects are now subjected to increasingly high current densities. Under these conditions, electromigration can lead to the electrical failure of interconnects in relatively short times, reducing the circuit lifetime to an unacceptable level. It is therefore of great technological importance to understand and control electromigration failure in thin film interconnects in integrated chips.

6.1.1 Electromigration

Electromigration (EM) is generally considered to be the result of momentum transfer from the electrons, which move in the applied electric field, to the ions which make

up the lattice of the interconnect material. It results in a tendency for the metal to move, or drift, in the direction of electron flow and thus creates hillocks and voids in interconnects as shown in figure 6.1. Average Current, flow direction over time, temperature and material properties are the major influential parameters for EM. Clock and power networks are more prone to EM due to high activity factor and significant flow of currents. The possibility of EM is reduced by limiting the average current through a net. Voids are created when ions are depleted mostly at the end of the wire at which electrons enter. This increases wire resistances and cause RC delay to be more than the estimate. This may slow down signal transmission. Voids may completely break wires and can cause opens. On the other end of the wire, where electrons leave, ions are pushed to and accumulate. Such hillocks may crack surrounding layers and may short with adjacent wires. As the current flows through a wire, the moving electrons bump into more or less stationary atoms.[16] The electrons carrying current have kinetic energy and they transfer some of their energy to atoms. As the atoms start to vibrate, the wire heats up. Interconnects experience heating regardless of current direction. A thermal acceleration process may worsen the EM damage.

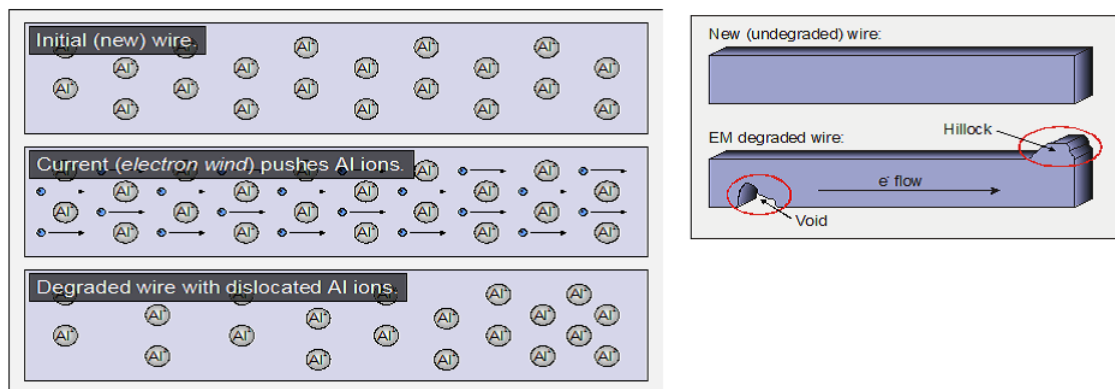


Figure 6.1: Electromigration

Once a void is initiated, it causes the current density to increase in the vicinity around itself as it reduces the cross sectional area of the conductor. It causes joule

heating in the void, increases the local temperature and leads to further growth of the void.

6.1.2 IR Drop

The power supply in the chip is distributed uniformly through metal layers across the design. The layers are Vdd and Vss. These layers have some finite amount resistance. When voltage is applied to this metal wires current flows through the metal layers. Due to resistance there will be some voltage drop as shown in figure 6.2. This drop in voltage is known as IR Drop. For example, a design needs to operate at a voltage level of 2Vs and has a tolerance level of 0.4V. In this case, we need to make sure that the voltage across the power pin and ground pin in that design does not fall short of 1.6 Volts.[17]

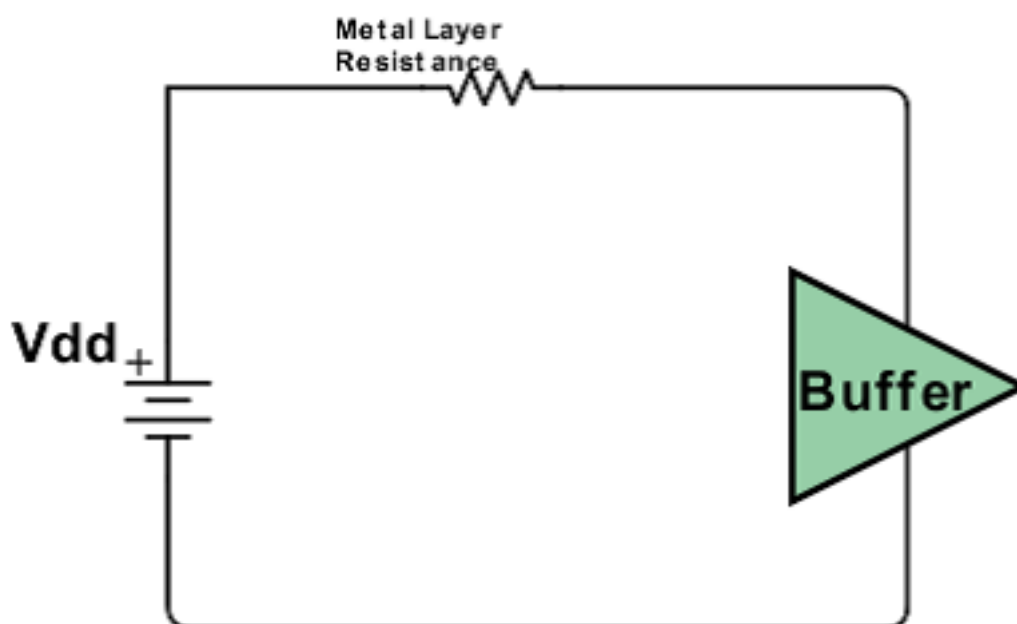


Figure 6.2: IR Drop

IR Drop results in Signal Integrity (SI) effect. It is caused by wire resistance and the current drawn from power grids according to Ohms law ($V = IR$). If wire resistance is higher than expected or if there is a surge current passing through the metal layers, an undesirable voltage drop may occur. Due to this unacceptable voltage drop, power supply voltage decreases leading to failure in logic levels of signals. That means that minimum needed power is not reaching the cells in our design. This leads to increased noise susceptibility and poor performance. Our design will contain different types of gates and cells variants which have different voltage levels. If the gate voltages drop and do not reach a threshold level, the dives might not turn on at all. It can also lead to increased delay of the device. This in turn may lead to timing violations depending on the path these cells are present in the design. As the technology node is shrinking, the sizes and widths of the metal layers is changing. The resistance of the layers has also increased which leads to decrease in power supply voltage. During the important stage of Clock Tree Synthesis, various buffers and inverters are added in the clock path to balance the skew. The voltage drop on the buffers and inverters of clock path will cause the delay in arrival of clock signal, resulting hold violation.

6.1.3 Self Heat

Heat is generated in interconnect that carries current leads to SH [18]. Figure 6.3 shows effect of SH on interconnects. Self-heating of the wires causes:

- Increases the die temperature further
 - Accelerates EM failures (self and neighbors)
- Increases the temperature in the region
 - May cause individual wires to melt in extreme cases
- May heat the chip to a point where transistors no longer function properly

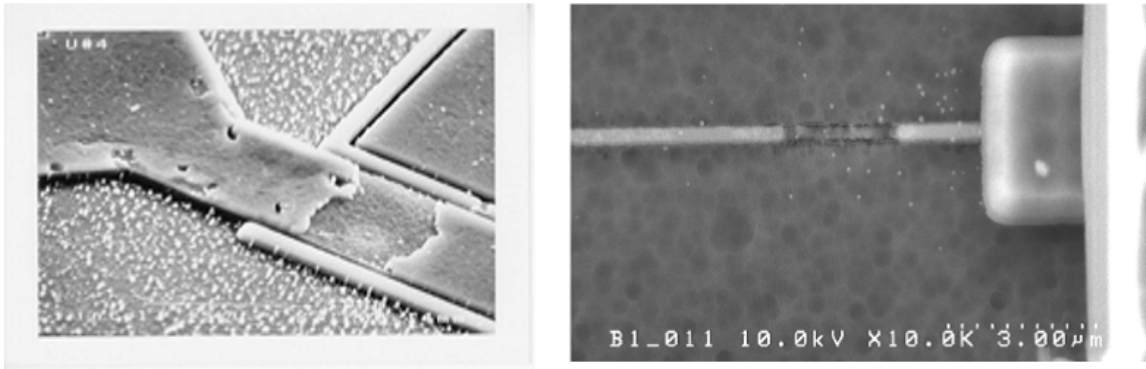


Figure 6.3: Self Heat Effect on Interconnects

6.2 RV Optimization Methods

6.2.1 Cell Spreading

Consider the layout sample given in fig 6.4. The upper metal layer routing which provides power for cells 1-4 are same. If all the cells are significantly current consuming, there is a possibility of EM in via through which the current is passing. Thus, spreading the cells is a recommended way in reducing the possibility of EM in such vias. One or more cells may need to be moved towards any of the nearby vias, which dont have much receiver cells.

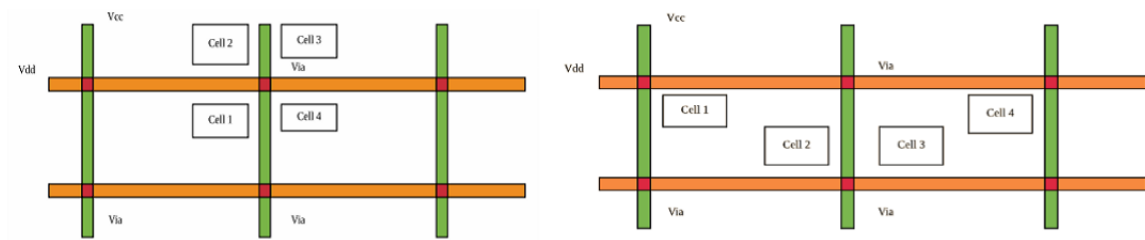


Figure 6.4: Cell Spreading Technique

6.2.2 Clock Buffer Splitting

Clock nets are highly prone to EM due to their high activity factor and significant load capacitance from multiple receiver latches. Whenever the clock switches its state, it has to charge the load and that too frequently. Thus the average current and charge passing through the cell are generally high and it leads to EM. Downsizing the clock buffer may seem as a possible solution, as the driving current will get reduced by downsizing. As the load capacitance is unaffected, total charge passing through the net remains the same. The most recommended method is to split the clock buffer as in figure 6.5. Thus the load will be split between two buffers and total charge passing through each clock net will get reduced.

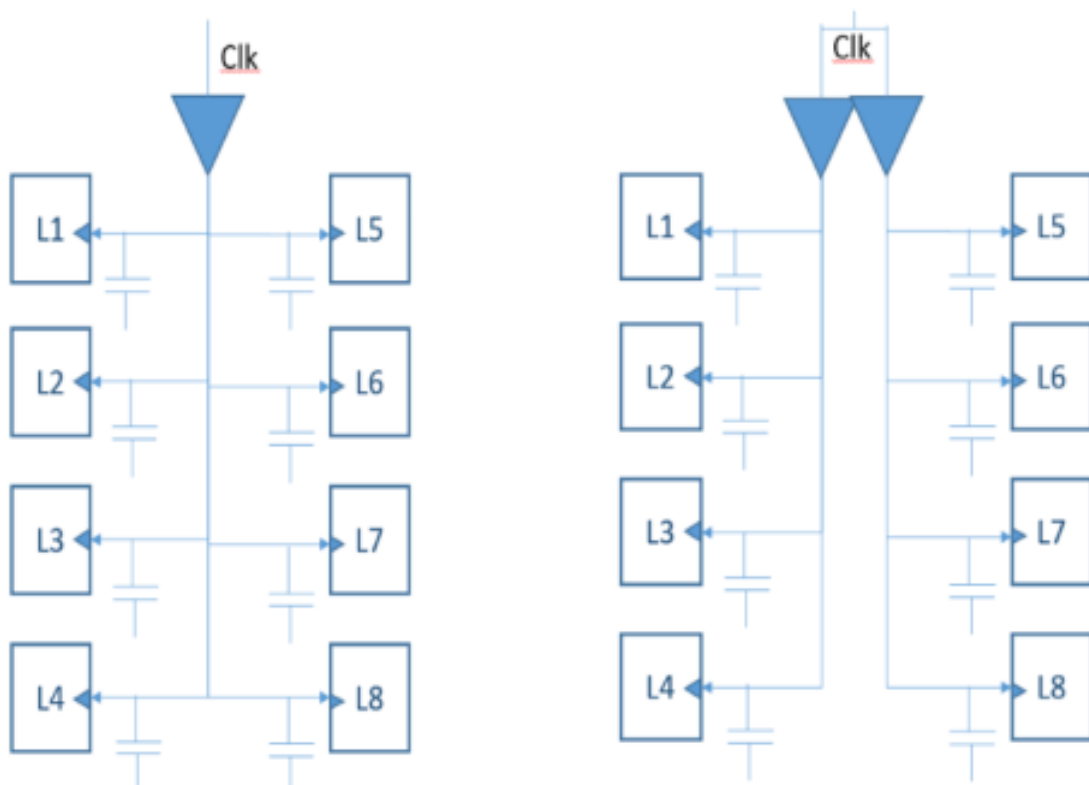


Figure 6.5: Clock Buffer Splitting

It is also recommended to spread the split buffers in the layout and move to different power vias. Downsizing the clock buffer may seem as a possible solution, as the driving current will get reduced by downsizing. As the load capacitance is unaffected, total charge passing through the net remains the same. The most recommended method is to split the clock buffer. Thus the load will be split between two buffers and total charge passing through each clock net will get reduced. It is also recommended to spread the split buffers in the layout and move to different power vias.

Chapter 7

Results

We started the project phase by taking previous generation processor modules. Scaling in process technology was given for new generation processor. Our starting point is previous generation processor module which will be called Reference in the result section. Incremental design and optimization done for the current generation processor will be called Current in the result section. All the results are shown with the help of screen shots which are captured directly from Intel customized tools which were used for the design of Broadcast module.

7.1 Formal Equivalence Verification

Formal Equivalence verification is done to ensure that designed digital circuit is same as that of spec RTL given. Status of verification tool flow for Reference is as given in Figure 7.1. Various steps of verification were explained in Chapter 3.

From figure 7.1, it is clear that verification flow was not clean when we began the design. Interface and verify stages were failing. A passed stage is denoted by letter P in green. If a stage is failed it will be denoted by letter F. Interface stage was failing because there were some polarity mismatch between signals coming from section. Design was taking direct polarity whereas section expected design to take inverted polarity for signals in the design. As a result at the input of some signals,

inverters were missing in the Reference. Because of the absence of these inverters verify stage was failing. Diff cones were generated since design was as not expected from RTL.

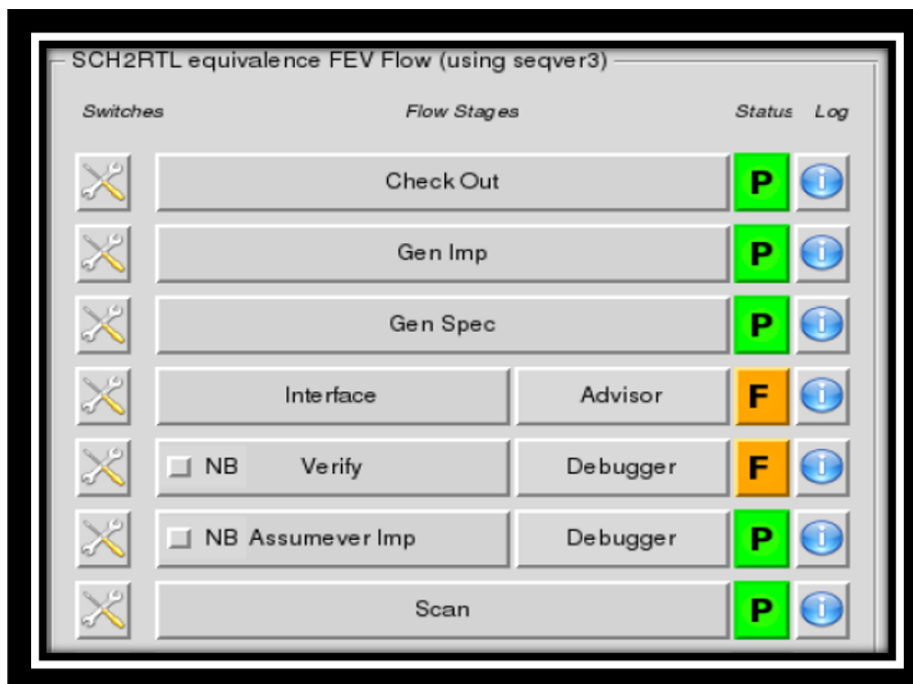


Figure 7.1: Reference Verification Tool Status

Figure 7.2 shows total number of cones and Diff cones in the design. The entire design was divided into 1482 cone pairs. Out of that 640 cones were different because of the absence of inverters at the input.



Figure 7.2: Total number of different pairs

In order to make the verification stage pass, inverters were added at the input of external signals. This helped in clearing interface stage. Interface stage is passed with waivers as denoted by Pw. This is because some signals were given different name with respect to central data. It was waived as the project is in not in the final stage. Status of verification tool for the broadcast design is shown in Figure 7.3

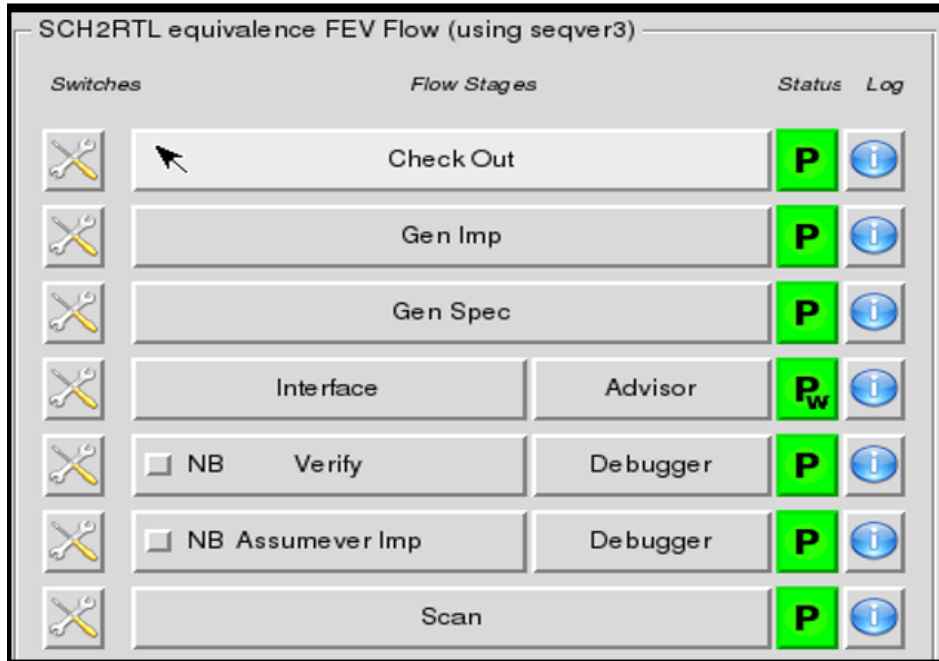


Figure 7.3: Current Design Verification Tool Status

All the stages of verification tool were passed. All the cones of the design were matching with that of specification RTL. Screen shot from tool depicting same is given in figure 7.4 below

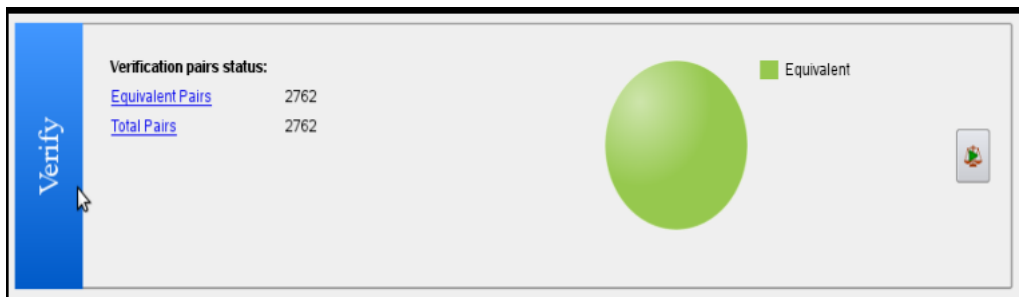


Figure 7.4: Status of cones of design

7.2 Timing Optimization

In static timing analysis, slack indicates whether timing is met along a timing path. A positive slack means that the signal can get from the startpoint to the endpoint of the timing path fast enough for the circuit to operate correctly. A negative slack means that the data signal is unable to traverse the combinational logic between the startpoint and the endpoint of the timing path fast enough to ensure correct circuit operation. In late mode analysis, slack is the difference between the required time and the arrival time for the timing path. The time that a signal needs to arrive at the endpoint of the path to ensure that timing is met is called the required time. The time that the signal actually arrives at the endpoint is called the arrival time. Because slack is the required time minus the arrival time, a negative slack indicates that the signal arrives at the endpoint later than the time it needs to be there, and vice versa for positive slack.

- Worst Negative Path(WNS) points to the path having the maximum negative slack.
- Total Negative Slack(TNS) gives the sum of all the negative slacks in the design.

Initial run suggested severe timing degradation which needed to be optimized. The results of same are given below in the table 7.1.

| | | |
|-------------|----------------|----------|
| MAX/ SET UP | TNS (ns) | -111.188 |
| | WNS (ns) | -18.940 |
| | Negative Paths | 106 |
| MIN/HOLD | TNS (ns) | -474.683 |
| | WNS (ns) | -21.099 |
| | Negative Paths | 269 |

Table 7.1: values of TNS and WNS for run1

Improvement in timing was done using resizing of cells, optimal placement and routing, buffer insertion and removal to push/pull the clock. After first iteration, timing converged significantly as shown in table 7.2.

| | | |
|-------------|----------------|---------|
| MAX/ SET UP | TNS (ns) | -49.980 |
| | WNS (ns) | -7.954 |
| | Negative Paths | 73 |
| MIN/HOLD | TNS (ns) | -462.66 |
| | WNS (ns) | -20.138 |
| | Negative Paths | 252 |

Table 7.2: values of TNS and WNS for run 2

Second iteration was performed to further improve the timing of the design. Similar approach was used to meet the timing criteria. The results are shown in table 7.3.

| | | |
|-------------|----------------|----------|
| MAX/ SET UP | TNS (ns) | -1.101 |
| | WNS (ns) | -0.094 |
| | Negative Paths | 51 |
| MIN/HOLD | TNS (ns) | -325.593 |
| | WNS (ns) | -21.082 |
| | Negative Paths | 231 |

Table 7.3: values of TNS and WNS for run 3

Graphical representation for the same is shown in figure 7.5.

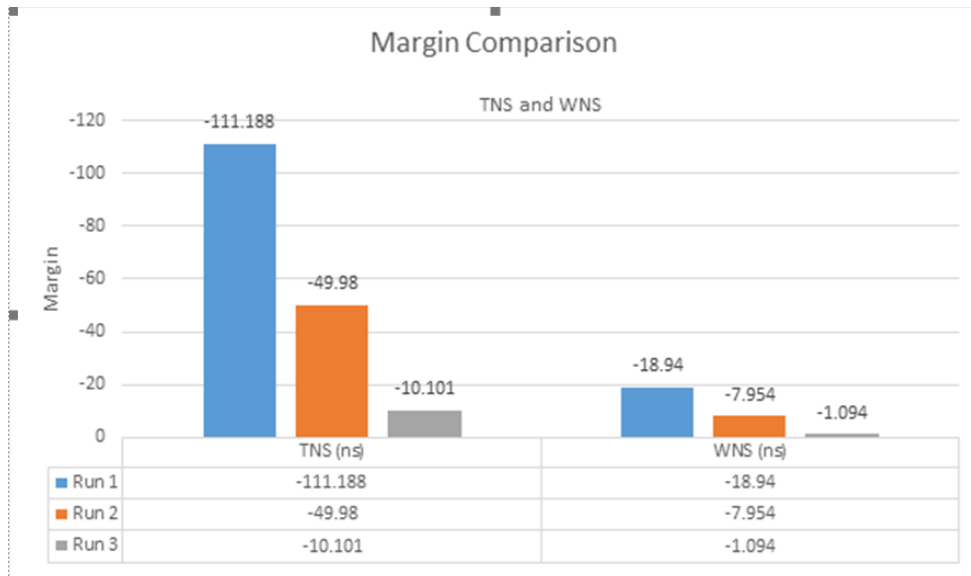


Figure 7.5: values of TNS and WNS

7.3 Power Optimization

Each power optimization technique is seen to give power gain. However, from the number of blocks that these methods were tested on, few were exceptions and did not show desired results. We will discuss the result of each of the techniques mentioned in previous chapter.

7.3.1 Power Driven Placement

For LPP, reduction in dynamic capacitance of the blocks is observed. Less dynamic capacitance leads to less dynamic power. Dynamic capacitance reduced by a factor of 2% - 3%. Figure 7.9 shows the results of LPP synthesis. It can be seen that there is gain in three out of 4 blocks.

| Block1 | % change in dyn cap | Total cells | Clock cells | Sequentials | Buffers | Inverters |
|------------------------|--------------------------------|------------------------|------------------------|--------------------|----------------|------------------|
| No Constraints | | 41684 | 3503 | 7265 | 5841 | 11410 |
| Power Driven placement | 1.07 | 43884 | 3768 | 7257 | 5740 | 13398 |
| Block2 | % change in dyn cap | Total cells | Clock cells | Sequentials | Buffers | Inverters |
| No Constraints | | 21806 | 1206 | 3492 | 2280 | 5566 |
| Power Driven placement | 3.49 | 20738 | 1140 | 3493 | 2209 | 4984 |
| Block3 | % change in dyn cap | Total cells | Clock cells | Sequentials | Buffers | Inverters |
| No Constraints | | 17079 | 1152 | 2573 | 2037 | 5917 |
| Power Driven placement | 2.80 | 15293 | 1064 | 2573 | 1629 | 4666 |
| Block4 | % change in dyn cap | Total cells | Clock cells | Sequentials | Buffers | Inverters |
| No Constraints | | 20779 | 1407 | 2898 | 2279 | 5331 |
| Power Driven placement | -0.12 | 21724 | 1468 | 2898 | 2352 | 6000 |

Table 7.4: Result of LPP Synthesis

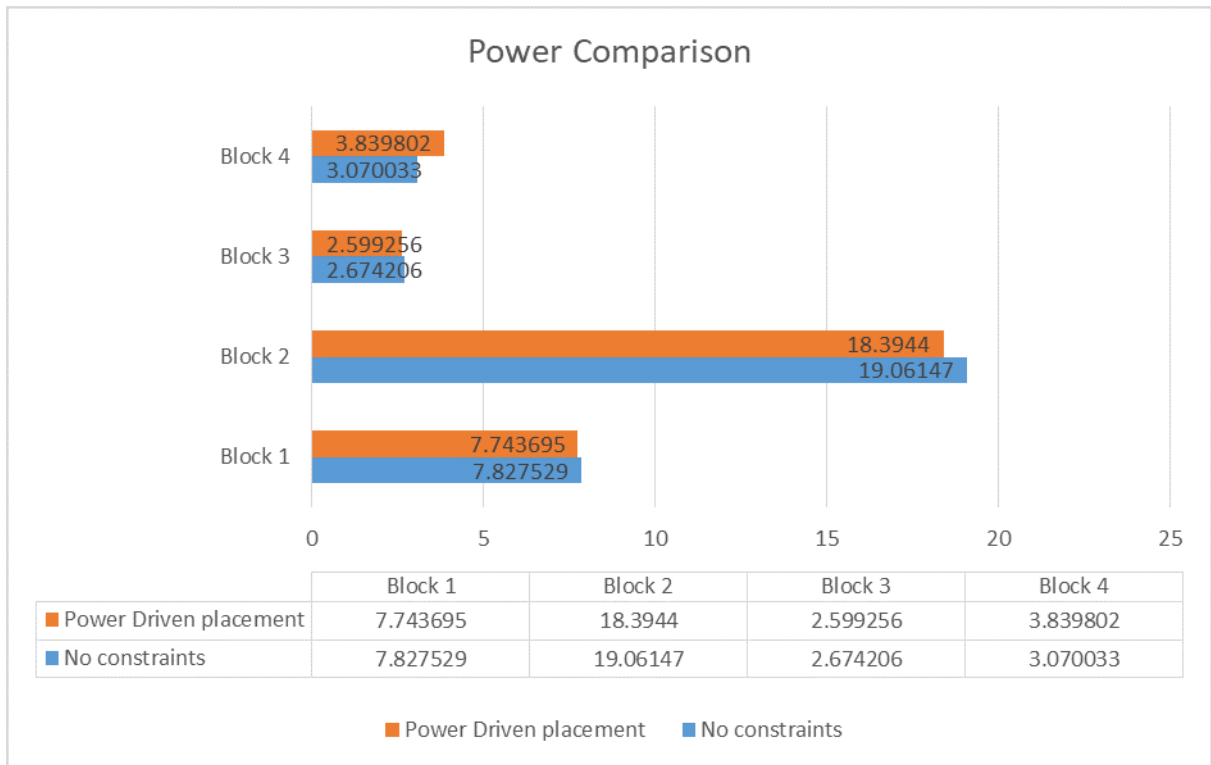


Figure 7.6: Comparison of results for LPP synthesis

The reduction in the length of higher activity factor nets can also be seen in figure 7.7.

| Net Length | <80 | <70 | <60 | <50 | <40 | <30 | <20 | <10 | <0 |
|------------------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| Normal design | 1 | 1 | 1 | 2 | 18 | 42 | 86 | 927 | 2046 |
| Power driven run | 0 | 0 | 0 | 5 | 19 | 50 | 76 | 854 | 1996 |

Figure 7.7: Activity Factor comparison

7.3.2 Multi Vt Synthesis

Multi Vt synthesis gave a reduction in the resistance of the cells. This resulted in the reduction of power. Table 7.5 shows the results of multi Vt synthesis

| Block1 | Overall Resistance | % change | Block2 | Overall Resistance | %change |
|----------------|---------------------------|-----------------|----------------|---------------------------|----------------|
| No Constraints | Total Z: 45004.6 | | No Constraints | Total Z: 45959.7 | |
| MultiVt design | Total Z: 43328.4 | 3.74% | MultiVt design | Total Z: 44603.4 | 2.9% |
| Block3 | Overall Resistance | % change | Block4 | Overall Resistance | %change |
| No Constraints | Total Z: 92947.6 | | No Constraints | Total Z: 38054.1 | |
| MultiVt design | Total Z: 94864.3 | -1.9% | MultiVt design | Total Z: 37957.1 | 0.25% |

Table 7.5: Multi Vt Synthesis Results

The use of multi Vt cells reduced the logic levels in the design. This method also reduced the overall resistance of the design. We get gain in dynamic power of 2% - 3% using this method.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In this study, the digital circuits were designed in accordance to design flow. Designing functional unit blocks in advance process node was really challenging. The digital circuits were designed in accordance to back end design flow. All the optimization techniques were discussed in detail and results were tabulated. In this project, all the experiments are performed in the new environments (like new library, quality, new features, new frequency and technology node) in which microprocessor core has never been designed before.

Some new timing optimization techniques were introduced in this study which can be performed in VLSI circuits to give the required frequency push. If timing optimization is done inefficiently and some paths have less setup/hold margins then power optimization can create negative margins on those paths. Using bounded logic in our design greatly helped us in improving the margins of the concerned paths. Both setup and hold margins are important for us and thus fixing them is the most important part of the project. Apart from using these methods, we used conventional timing optimization method to clear out much of the negative paths. In total, each of the technique helped us in achieving frequency target.

The paths that have to be optimized with power must have good margins. As the number of core increases the power consumption also plays a major factor in design of microprocessor. So with the multi core processor the design target is to achieve low power at high speed. Various design and implementation challenges are analyzed and properly addressed with optimized solutions. All the optimization techniques were discussed in detail and results were tabulated. Some new power optimization experiments were introduced in this study which can be performed in VLSI circuits to save dynamic power at high frequency. Power optimization is the most aspect of designing a microprocessor as the lower technology nodes rely on power for their durability.

The techniques mentioned in this work can be used separately or together to get power gain depending on the requirement. Each method though beneficial may have its own downside. Gaining power can cost timing. Reduction in power may or may not lead to reduction in overall frequency of operation. Power Gain also depends on the complexity of the design. Low complexity blocks can show more variation. If timing is affected by any of the methods, focus then should be on improving timing by making changes in the design. Apart from optimization, to achieve overall performance worthy core, quality checks in terms of Reliability Verification were done and confidence was obtained to on the design quality.

8.2 Future Work

Each new project will be an improvement over its predecessor. This is achieved by adding newer RTL features and further technology scaling. This requires more efforts in terms of performance optimization and better quality checks. Other important factors are noise and its effect on the signals. Extensive work can be done in this field to achieve immunity from noise. Moreover, Timing of the critical paths, speed-path analysis and debug are other factors that could be a plan of future works.

References

- [1] K. Anshumali, T. Chappell, W. Gomes, J. Miller, N. Kurd, and R. Kumar, "Circuit and process innovations to enable high-performance, and power and area efficiency on the nehalem and westmere family of intel processors.," *Intel Technology Journal*, vol. 14, no. 3, 2010.
- [2] M. Bajkowski, G. Pham, and M. Vepadharmalingam, "Optimum leakage dynamic array design," pp. 262–265, 2010.
- [3] S. Borkar, "Design perspectives on 22nm cmos and beyond," pp. 93–94, 2009.
- [4] S. Gunther, A. Deval, T. Burton, and R. Kumar, "Energy-efficient computing: Power management system on the nehalem family of processors.," *Intel Technology Journal*, vol. 14, no. 3, 2010.
- [5] S. Arora, U. Dutta, and V. K. Sharma, "A noise tolerant and low power dynamic logic circuit using finfet technology," vol. 28, 2007.
- [6] R. Rajprabu, V. A. Raj, R. Rajnarayanan, S. Sadaiyandi, and V. Sivakumar, "Performance analysis of cmos and finfet logic," *IOSR Journal of VLSI and Signal Processing*, vol. 2, pp. 01–06, 2013.
- [7] J. C. Tinoco, S. S. Rodriguez, A. G. Martinez-Lopez, J. Alvarado, and J.-P. Raskin, "Impact of extrinsic capacitances on finfet rf performance," *IEEE transactions on microwave theory and techniques*, vol. 61, no. 2, pp. 833–840, 2013.
- [8] P. B. Shweta Kataria, "Finfet technology: A review paper," vol. 5, 2016.
- [9] "Intel encyclopedia," Intel Encyclopedia, 2019. [ONLINE]. Available: <http://www.intelpedia.intel.com>.
- [10] "Intel specific - history of microprocessor documents,"
- [11] M. Dale, "The power of sequential design optimizations (tech. trends silicon/eda)," pp. 102–103, 2008.
- [12] "Static timing analysis," Vlsi-expert, 2019. [ONLINE]. Available: [https://www.vlsi-expert.com/Static timing analysis](https://www.vlsi-expert.com/Static%20timing%20analysis).

- [13] “Power optimization in ic compiler ii,” Synopsys solvenet, 2018. [ONLINE]. Available: [https://www.synopsys.com/solvenet.com/Power Optimization in IC Compiler II](https://www.synopsys.com/solvenet.com/Power%20Optimization%20in%20IC%20Compiler%20II).
- [14] P. Zhao, J. McNeely, W. Kuang, N. Wang, and Z. Wang, “Design of sequential elements for low power clocking system,” *IEEE Transactions on very large scale integration (VLSI) systems*, vol. 19, no. 5, pp. 914–918, 2011.
- [15] S.-J. Wang, S.-J. Huang, and K. S.-M. Li, “Static and dynamic test power reduction in scan-based testing,” pp. 56–59, 2009.
- [16] B. Li, P. S. McLaughlin, J. P. Bickford, P. Habitz, D. Netrabile, and T. D. Sullivan, “Statistical evaluation of electromigration reliability at chip level,” *IEEE Transactions on Device and Materials Reliability*, vol. 11, no. 1, pp. 86–91, 2011.
- [17] “Ir drop,” Vlsi-basics, 2019. [ONLINE]. Available: [https://www.vlsi-basics.com/IR Drop](https://www.vlsi-basics.com/IR%20Drop).