Runtime Improvement of DDR and Rivosx6 Gen PCIE Controller

Major Project Report

Submitted in fulfillment of the requirements
for the degree of

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

By

Aniket Paunikar

(22MECV02)

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

May 2024

Runtime Improvement of DDR and Rivosx6 Gen PCIE Controller

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Electronics & Communication Engineering

By

Aniket Pushparaj Paunikar

(22MECV02)

Under the guidance of

External Project Guide:

Mr. Hiren Gadher

R and D Manager, EDAG, Design Technology Group

Synopsys India Pvt. Ltd.,

Bangalore.

Internal Project Guide:

Dr. Vaishali Dhare

Asso. Professor, EC Department,

Institute of Technology,

Nirma University, Ahmedabad.

Electronics & Communication Engineering Department

Institute of Technology-Nirma University

Ahmedabad-382 481

December 2023

Declaration

This is to certify that

a. The thesis comprises my original work towards the degree of Master of Technology in VLSI Design 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.

- Aniket Pushparaj Paunikar
22MECV02

Disclaimer

"The content of this paper does not represent the technology,opinions,beliefs, or positions of Synopsys India Pvt. Ltd.,its employees,vendors, customers, or associates."

Certificate

This is to certify that the Major Project entitled **"Runtime Improvement of DDR and Rivosx6 Gen PCIE Controller"** submitted by **Aniket Paunikar (22MECV02)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in VLSI Design, 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.The results embodied in this major project, to the best of our knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Date: Place: Ahmedabad

Dr. Vaishali Dhare
Internal Guide

Dr. Usha Mehta
Program Coordinator

Dr. Usha Mehta
Section Head, EC

Director
School of Technology, ITNU

Certificate

This is to certify that the Major Project entitled **"Runtime Improvement of DDR and Rivosx6 Gen PCIE Controller"** submitted by **Aniket Pushparaj Paunikar(22MECV02)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in VLSI Design, Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.

Mr. Hiren Gadher
R and D Manager, EDAG, Design Technology Group
Synopsys India Pvt.LTD
Bangalore

Acknowledgements

It gives me immense pleasure in expressing gratitude to **Dr. Vaishali Dhare**, Assistant Professor, EC Department, Institute of Technology, Nirma University for his valuable guidance and continual encouragement throughout this work.

The chain of my gratitude would be incomplete if I would forget to thank the first cause of this chain, **Dr. Usha Mehta**, Hon'ble Head and PG Coordinator of VLSI Design, EC Department Institute of Technology, Nirma University for her kind support and providing basic infrastructure, healthy research environment and for providing this exhilarating opportunity and continual support in reaching a higher goal.

It is a pleasure to be associated with the Synopsys India Pvt. Ltd., and I would like to thank the entire staff. I am grateful to **Mr. Hiren Gadher**, Team Manager, Synopsys India Pvt Ltd. and Other Team members, for their valuable guidance and continual encouragement throughout this work. The appreciation and continual support imparted have been a great motivation for me in reaching a higher goal . Thier guidance has triggered and nourished the intellectual maturity that I will benefit from, for a long time to come.

A special thank you is expressed wholeheartedly to Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout this work.I would also thank the Institution, all faculty members of the Electronics and Communication Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Aniket Pushparaj Paunikar
22MECV02

Contents

List of Tables

List of Figures

Abstract

VLSI initiatives use sub-nanometer technology for a number of compelling reasons. First of all, it makes it possible for electronic components to get smaller, which raises the integration densities on chips. As a result, the integrated circuit's physical size doesn't rise proportionately, but its functionality and performance do.Second, increased power efficiency is a benefit of sub-nanometer technologies. Electronic devices become more energy-efficient due to the reduction in power consumption caused by smaller transistors and interconnects. This is essential in contemporary applications where battery life and power limitations are important factors.

Additionally, because sub-nanometer technologies have shorter signal propagation delays, they enable quicker operating speeds. This is crucial for applications where quick and effective signal processing is required, including baseband processing.Sub-nanometer technology adoption in VLSI projects generally tries to strike a balance between expanded functionality, better power efficiency, improved performance, and the capacity to satisfy the demands of new technologies in a smaller form factor.

This thesis uses state-of-the-art sub-nanometer technology to physically build the baseband block. The main objective is to create a highly efficient baseband processing unit with dimensions less than one nanometer by designing and fabricating it using advances in semiconductor techniques. This project involves complex circuit design, painstaking layout optimisation, and overcoming obstacles related to sub-nanometer technologies. Through a thorough understanding of the practical aspects of baseband block implementation, the research aims to improve integrated circuit performance, maximise power efficiency, and augment overall functionality. The result of this effort advances VLSI technology and opens the door for baseband processing in electronic systems of the future that is more powerful, compact, and energy-efficient.

Abbreviation Notation and Nomenclature

TNS	Total negative slack
WNS	Worst negative slack
R2RTNS	Register to register total negative slack
NVP	No of violating path
MTRANS	Max transition
DRC	Design rule check

Chapter 1

Introduction

This chapter serves as the introduction to the thesis and will cover the implementation of DDR and the Rivosx6 Gen PCIE Controller. Moreover, a flowchart describes the planned power, area, and performance specifications for the backend design phase. Additionally, the training's objectives, duration, and scope at Synopsys India Pvt. Ltd. have been mentioned.

1.1 Company Profile

American electronic design automation (EDA) company Synopsys is well-known. Please be aware that things may have changed or developed since then, so it's best to consult the most recent sources to get the most recent details. In the EDA market, Synopsys is a significant player. It offers hardware and software solutions that make it easier to design and verify integrated circuits (ICs) and electronic systems. This comprises resources for verifying the dependability of intricate semiconductor devices, optimising power usage, and building and simulating chips.Semi-conductor firms can include Synopsys' semiconductor IP, which includes processors, interfaces, and other building blocks, into their unique chip designs.Apart from its fundamental EDA products, Synopsys has diversified into software security and high-quality solutions. This comprises. This comprises code quality and compliance assurance tools as well as tools for locating and fixing security flaws in software.In the field of optical design and simulation, Synopsys is present and offers resources for creating and refining photonic systems and devices.a synthesis tool for designing integrated circuits using digital technology.A platform for software testing that assists in finding and repairing security flaws in code.Synopsys has been instrumental in the advancement of sophisticated semiconductor technology, helping to build and validate intricate chips that are utilised in a wide range of electronic gadgets.Through smart acquisitions, the company has increased the scope of its offerings and improved its expertise in areas like intellectual property and software security. Organisational Culture. Synopsys places a strong emphasis on innovation, teamwork, and a dedication to providing its clients with state-of-the-art solutions.

1.2 Mission and Vission

Synopsys's goal and ambition were centred on its place in the technology sector. Remember that businesses occasionally update their purpose and vision statements, so it's best to confirm the most recent details from the organization's official sources. With regard to electronic design automation (EDA) software and solutions, Synopsys sought to be a world leader. The company's dedication to helping its clients create and develop high-quality electronic goods effectively was frequently highlighted in the mission statement.Typically, Synopsys' mission was to assist customers design the most inventive and dependable products by pushing the boundaries of electronic design. This required being on the cutting edge of technology, spotting market trends, and offering solutions that give electronic system and semiconductor designers more power.It's crucial to remember that mission and vision statements can change over time in response to changes in the business environment, consumer behaviour, and industry trends. See the official Synopsys website or recent corporate communications for the most up-to-date and accurate information.

1.3 Products and Services

Synopsys provides a wide range of goods and services, mostly related to software security, semiconductor intellectual property (IP), and electronic design automation (EDA). The company offers solutions and technologies to meet the requirements of electronic system and semiconductor designers. Please be aware that Synopsys' product offerings may have changed since then; for the most recent information, visit the company's official website or any updated documentation. Photonic device and system design and optimisation tools are among the optical design and simulation products offered by Synopsys. To assist clients in making the most use of their tools and streamlining their design processes, Synopsys provides a variety of services in addition to its software products. These services include training, consulting, and support. For the most recent details about Synopsys' goods and services, visit their official website or get in touch with them directly.

1.4 Motivation

For the design of Very Large Scale Integration (VLSI) and related sectors to remain innovative, solve difficult problems, and lead the technological curve, motivation is essential.The rapid advancement of technology in VLSI design is intrinsically inspiring. Engineers and researchers are motivated by the chance to influence and participate in technical developments that will determine the direction of electronics in the future.The ongoing quest of power efficiency and miniaturisation is a part of VLSI design. The challenge of cramming more functionality into smaller spaces while using less power drives engineers, who in turn help to produce high-performance,

energy-efficient electronic products.VLSI designers are driven by the opportunity to be imaginative and inventive. When designing intricate circuits and systems, one must think creatively and develop unique, coming up with novel solutions, and pushing the boundaries of what is currently possible.

1.5 Problem Statement

This entails optimising data structures and algorithms used in lexical analysis, parsing, and semantic analysis, as well as enhancing the effectiveness of parsing and analysing source code. Cutting down on the amount of time needed to parse source code and produce an intermediate representation (IR) is the aim.focuses on optimising the algorithms and data structures that are utilised to express the semantics of the programme. To decrease the size of the IR and increase its effectiveness, strategies including common subexpression reduction, dead code elimination, and constant folding can be used.In order to enhance the efficiency of the generated code, this step entails applying high-level optimisations including loop optimisation, data flow analysis, and parallelization. Loops can be made more parallel and cache-efficient by using techniques like loop fusion, loop unrolling, and loop exchange. Producing effective machine code from the optimised IR is a necessary step in optimising the code generation process. This entails choosing the right instructions, planning their execution, and effectively using memory and registers as resources. Performance of code creation can be increased by utilising strategies like register allocation, target-specific optimisations, and instruction scheduling. entails improving the backend's memory management, runtime libraries, and support for parallel execution. To increase runtime performance, strategies including memory pooling, thread-level parallelism, and data layout optimisation might be used.

1.6 Approach

Enhancing a fusion compiler's runtime necessitates a complex strategy that takes into account different phases of the compilation procedure. This is a methodical technique. To find areas for improvement and bottlenecks, start by profiling the compiler. To comprehend each level of the compiler's performance, benchmark it at various stages.Examine and improve the algorithms used for optimisation passes, semantic analysis, and parsing. To cut down on time complexity and memory consumption, use effective data structures and algorithms.To maximise typical operations, use algorithmic approaches such as memoization and dynamic programming.When feasible, introduce parallelism, particularly during phases like code development and optimisation. To parallelize individual activities and increase performance overall, use multi-threading.Investigate task and data parallelism to make the most of the computational resources at your disposal.

1.7 Scope of Work

A comprehensive plan covering multiple facets of the compiler's architecture, algorithms, and procedures is part of the scope of work for optimising the runtime of a fusion compiler.Find performance bottlenecks by thoroughly analysing the current compiler implementation.Examine every phase of the compilation process to determine which areas require the greatest optimisations.To create baseline performance measurements for comparison, gather benchmarking data.Analyse and improve the algorithms used for code creation, parsing, semantic analysis, and optimisation steps.Determine areas where algorithmic enhancements can be made to lower memory overhead and temporal complexity.Use effective data structures and algorithms that are adapted to the particular needs of the compiler.

Examine ways to parallelize compilation processes in order to make the most of multi-core CPUs.Create and put into practice parallelization techniques for separate optimisation passes and compilation stages.Maintain correctness by making sure thread safety and synchronisation are followed when needed.Improve cache locality by analysing memory access patterns and optimising data structures.Reduce cache misses by using memory layout optimisations and cache-aware algorithms.Try different caching and prefetching strategies to reduce memory access latency.Use conventional compiler optimisation strategies including code mobility, constant propagation, and loop optimisation.Examine optimisations particular to a given domain that are adapted to the kinds of programmes that the fusion compiler normally compiles.Look at ways to enhance code quality through whole-program analysis and inter-procedural optimisation.

To improve performance on target architectures, investigate and make use of hardware-specific features and instructions.Utilise vectorization and SIMD instructions to optimise code creation for contemporary processor architectures. Work together with hardware providers to remain up to date on the newest architectural developments and optimisation possibilities.Examine whether it is possible to use JIT compilation methods to optimise code pathways that are executed frequently in a dynamic manner.Create techniques for runtime optimisation depending on input received during programme execution.Provide instrumentation support and profiling tools to help inform judgements about runtime optimisation..

1.8 Outline of Thesis

This report has six chapters, each explaining a different part of the project. Chapter 2 looks at literature review of the run time improvement for the IP. Chapter 3 covers the flow of the physical design. Chapter 4 highlights the Exploration of each stage in the fusion compiler. Chapter 5 explains the Runtime improvement of the every stage of the fusion compiler. Chapter 6 brings all the knowledge together, discussing company work, results, and future possibilities. The last chapter gives a

final reflection on the internship, concluding the report.

Chapter 2

Literature Survey

2.1 Overview

Methodologies, methods, and tools utilised in the physical implementation stage of integrated circuit (IC) design are usually the subject of a literature review on the runtime improvement in physical design. This is how such a review could be organised:cite1Give a brief description of the component routing and placement steps involved in the physical design process of IC manufacture. Describe the importance of runtime improvement in physical design and how it can lower semiconductor device production costs and time-to-market. Examine the literature on placement algorithms that try to minimise wirelength, signal delay, and power consumption by arranging components on the chip in the best possible way. Talk about routing algorithms that are intended to provide effective connections between components while respecting performance goals and design limitations. Examine technical studies and research articles on timing closure techniques to make sure the design satisfies the timing specifications set forth by the desired clock frequency.Examine strategies for effectively accomplishing timing closure, such as delay optimisation, buffer insertion, and clock tree synthesis. Write a summary of the research on physical design power optimisation strategies, such as voltage scaling approaches, power gating, and clock gating.Talk about how these methods increase runtime by lowering ICs' static and dynamic power consumption.

Examine the cutting-edge physical synthesis technologies that are being utilised in academia and industry to automate different physical design processes. Talk about improvements in design quality, runtime performance, and optimisation techniques through the use of enhanced tool capabilities and user interfaces. Examine the literature on distributed and parallel computing strategies used to speed up physical design projects.Talk about research projects that use distributed computing platforms and multi-core processors to use parallel placement, routing, and timing analysis methods.Examine current research that uses artificial intelligence (AI) and machine learning approaches to improve automation of physical design. Talk

about how AI-based methods can forecast the best solutions and learn from previous design experiences to accelerate runtime improvement. Draw attention to issues with scalability, complexity of design, and limits on design rules that affect runtime improvement in physical design.Make recommendations for future research directions, such as incorporating cutting-edge technology like neuromorphic or quantum computing into the physical design process.

2.2 Present Technology

utilising distributed computing methods and multi-core CPUs to carry out physical design jobs simultaneously, thereby cutting down on runtime. View the flow diagram of the Fusion compiler below.[1]. By dividing the design into more manageable, smaller chunks or hierarchies, parallel processing is made possible, and the runtime is decreased by concentrating computation efforts where they are most needed.Incremental design strategies reduce overall runtime by updating only the parts of the design that are affected by changes, rather than rerunning the full design cycle. creating and putting into practice heuristics and algorithms that optimise physical design activities like routing and placement in order to save runtime without compromising quality.quicker convergence and runtime gains can be achieved by using machine learning and artificial intelligence approaches to predict and optimise several physical design features, such as floor planning, placement, and routing. Reduce memory overhead and computational complexity by designing and implementing effective data structures that are suited to certain physical design requirements, hence reducing runtime. Computer-aided design (CAD) tools designed specifically for physical design are continuously improved upon and optimised, with runtime optimisations and performance improvements incorporated. using specialised hardware accelerators to offload CPU-intensive computing, such as FPGAs or custom ASICs, to reduce runtime overall.Increasing computational capacity on-demand and facilitating quicker physical design turnaround times by utilising cloud computing resources and distributed computing environments.Because of improved computational efficiency and optimisation, advancements in semiconductor process nodes and physical design tools themselves frequently result in intrinsic runtime gains.

2.3 Advantages of the fusion compiler

EDA tools are generally intended to help chip designers in various ways both during the physical design and optimisation phases. These advantages might include: By applying sophisticated algorithms for optimisation, designers can use EDA tools to optimise performance and attain higher levels of speed, power consumption, and space utilisation. Efficiency in Energy Use Tools with power optimisation features, such as voltage scaling, power gating, and clock gating, can improve overall power efficiency. When to Shut Down EDA tools help designers achieve timing closure,

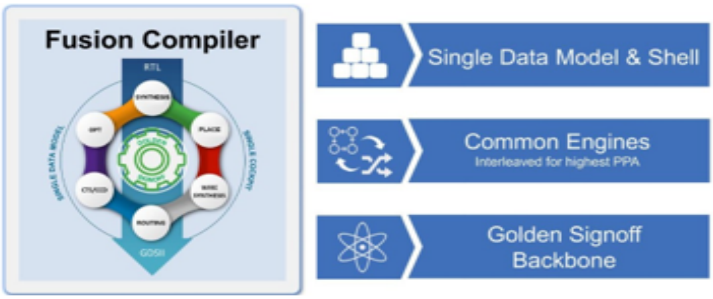

Figure 2.1: Fusion compiler[1]

which ensures that the design complies with timing requirements and operates at the desired clock frequency. Advanced Guidance and Route Modern EDA tools frequently come with advanced place-and-route capabilities that enable greater signal integrity, reduced wirelengths, and efficient chip space utilisation. Using ML and AI Together Certain technologies use artificial intelligence or machine learning algorithms for activities like placement, routing, and optimisation, leading to more intelligent and automated design choices. Verifying Design Guidelines (DRC) Ensuring that the physical layout conforms with design and manufacturing rules is made easier by robust DRC features in EDA tools, which lower the likelihood of fabrication issues. offered by intricate chip designs and complex process nodes. Optimisation with several goals EDA tools have the potential to provide multi-objective optimisation, which allows designers to balance conflicting goals such as performance, power, and area. Efficiency and Readability When working with user-friendly interfaces and efficient processes that reduce the time and effort required for design tasks, chip designers are more productive. Support for Technology Nodes: To enable designers to capitalise on technology advancements, EDA tools are updated to support the newest nodes in the semiconductor production process. It is recommended to refer to the official documentation, release notes, or contact the tool's vendor in order to receive exact specifics regarding the advantages of "Fusion Compiler." In order to assist customers in making informed selections dur- ing the chip design process, vendors frequently include comprehensive information about the features, optimizations, and benefits of their products.

2.4 Core

A separate processing unit with the ability to carry out instructions on its own is called a core. A single CPU chip can have multiple cores incorporated into it to increase processing power and parallelism. Important details about CPU cores include Self-reliance In a multi-core CPU, each core can carry out a separate set of instructions without affecting the others. This makes it possible to process data in parallel, meaning that several jobs can be completed at once. In parallelism

Parallelism, or the ability to process many jobs or threads simultaneously, is made possible by multi-core processors. Improved system responsiveness and overall performance may result from this. Scaling Performance An increase in processor core count can result in higher processing capacity, enabling the system to tackle increasingly complex computational tasks. Execution Scaling An increase in processor core count can result in higher processing capacity, enabling the system to tackle increasingly complex computational tasks. SMT, or hyper-threading Certain processors use technologies like AMD's and Intel's Hyper-Threading and Intel's Simultaneous Multithreading (SMT). This technology further enhances parallelism by enabling each core to run numerous threads concurrently. Task Assignment Multiple cores must be efficiently utilised by the operating system and software. While certain activities can be broken down into smaller, parallel tasks, others might be sequential by nature. Several Core Architectures Dual, quad, hexa, octa, and more core architectures are common multi-core configurations. A processor's performance capabilities are mostly determined by the number of cores it contains.Particularised Cores: To handle particular sorts of computations more effectively, some processors have specialised cores in addition to general-purpose cores, such as graphics processing units (GPUs) or AI accelerators. Energy Efficiency When not every core is using its full potential all the time, multi-core computers can provide higher power efficiency than raising the clock speed of a single core. Multiprocessors on a Chip (CMP) When referring to separate processing units within a chip multiprocessor (CMP), the term "core" is sometimes used more widely. This could encompass numerous processor cores and other specialised cores on a single chip. In conclusion, a core is a basic processing unit found in a CPU or processor, and the idea of having several cores has become commonplace in modern computing to improve performance and handle the difficulties presented by applications that are becoming more complicated and demanding. .

In the context of computing, a processing core within a CPU (Central Processing Unit) or a processor that is intended to provide remarkable computational power, speed, and efficiency is referred to as a high-performance core. Modern processors with high-performance cores are essential parts of a wide range of computing systems, from servers and data centres to personal laptops. These are some essential traits and factors to take into account while choosing high-performance cores.

Every CPU core in a conventional multi-core CPU is the same. They all utilise the same amount of power and have the same performance rating. The issue with this is that there is a minimal power consumption that you cannot go below without completely turning down your CPU while it is idle or performing simple chores. When it comes to wall-plugged devices, this isn't the end of the world, but when you're using battery power, every watt matters. A company's strategic advantage is made up of its core competences, which include its resources and abilities. According to contemporary management theory, an organisation needs to identify, develop,

and use its core competitiveness in order to flourish in the marketplace. In contrast to the ideas that have surfaced in it advises job seekers to focus on their core personal skills to stand out. These positive qualities can be developed and listed on a resume. Some basic personal skills include analytical skills, creative thinking, and problem-solving. At higher clock speeds, which enables them to carry out instructions more quickly. Tasks are processed more quickly as a result, and performance is enhanced overall. The set of instructions that a processor is capable of executing is determined by its instruction set architecture. ISAs that are optimised for high performance cores are frequently used in their design to effectively serve a variety of workloads and applications.

2.5 IP

Usually, "IP" refers to "Intellectual Property." Pre-designed and pre-verified functional blocks or modules that are utilised as building blocks in the construction of integrated circuits (ICs) or electronic systems are referred to as IP in VLSI design. Specific activities, including CPUs, memory controllers, DSP units, communication interfaces, and other hardware components, are encapsulated in these IP cores.

By utilising pre-tested and validated designs, IP cores enable VLSI designers to minimise development time and expenses while maintaining performance and dependability. IP cores are frequently developed internally by semiconductor companies or supplied by outside vendors. They are usually supplied alongside related documentation and simulation models, and as encrypted hardware descriptions in formats like Verilog or VHDL. When designing and implementing complicated integrated circuits, IP cores are essential because they allow designers to concentrate on the distinctive features of their designs while including tested and dependable functionality from pre-existing IP blocks. Rapid prototyping, customisation, and the creation of sophisticated electronic systems are made easier by this method in a variety of industries, such as consumer electronics, automotive, aerospace, telecommunications, and more. IP in VLSI design stands for functional blocks or modules that are pre-designed and pre-verified and have particular functionality. These features can be found in simpler parts like logic gates or in more sophisticated devices like processors or communication interfaces. For VLSI designers, IP cores are essential since they offer pre-made solutions for frequently used functionalities. Since these IP cores are usually put through extensive testing, this guarantees performance and reliability while saving time and effort during the design process.Semiconductor businesses have the option to build IP cores internally or acquire them from outside IP vendors. The development and licencing of IP cores for many applications is the area of expertise for third-party IP providers. IP core licencing can take many different forms, such as subscription-based, royalty-based, or one-time purchase models. The license's provisions usually specify how the IP core may be used, altered, and

distributed. VLSI designers use hardware description languages (HDLs) like Verilog or VHDL to incorporate IP cores into their designs. The IP cores are connected and instantiated inside the overall design.The IP cores are instantiated within the overall design and connected to other components as needed. Many IP cores are designed to be customizable, allowing designers to tailor the functionality, performance, and interface of the core to meet the specific requirements of their design. .

Chapter 3

Physical Design Flow

3.1 Fusion Compiler Unified Flow

Creating a completely unified Design Platform has been a multi-year endeavour, with Fusion Compiler at its core. A hyper-convergent, signoff-accurate RTL-to-GDSII design flow may be produced with this integrated solution, which is built on a common data model and signoff-driven engines. This allows for the smooth invocation of several optimisation engines throughout the flow. The primary benefit of this platform is the exclusive use of Synopsys' golden-signoff solutions, which are the most reliable in the business, as the standard reference for the whole design flow. Fusion Compiler integrates RedHawk via our special relationship with Ansys and all relevant Synopsys Signoff Solutions, such as PrimeTime, PrimePower, PrimeShield, StarRC, ICV, and VCS. The primary function of design implementation is co-optimizing performance, power, and area (PPA), with the goal of guaranteeing that the resultant optimisation choices. The platform therefore inherently ensures that the design is optimal and signoff accurate at every stage of the flow since optimisation is driven by signoff engines. The improved quality of outputs that a design achieves is crucial, but so is the shorter time from initial flow to final signoff. As previously indicated, Fusion Compiler uses a variety of PPA optimisation strategies to handle different design problems. Typical non-signoff engines with defects on both sides cause many design elements to be over- or under-optimized. This results in unexpected outcomes and PPA loss at the conclusion of the cycle. The platform's golden-signoff engines can accurately direct the optimisation engines to concentrate their work in the appropriate areas and towards the appropriate goals by using Synopsys' special methodology. metrics. The effect – caused by a miscorrelation with final signoff. Traditionally designers apply pessimism on the design through mar- gins. While this technique may reduce some surprises, it comes with a significant cost of PPA. Fusion platform golden signoff approach eliminates the need for such costly margins. measurements. The result of a final signoff miscorrelation is the effect. Traditionally, designers use margins to add gloom to their designs. This method has a high PPA cost even though it might lessen some shocks. The

golden signoff technique of the Fusion platform removes the need for these expensive margins.Because of the exclusive access to and internal cooperation with Synopsys' most reputable signoff solutions, signoff correlation has long been a defining feature of Synopsys' digital implementation solutions. New effects that occur with advanced nodes require precise modelling for timing, power, and other metrics. PPA is lost when conventional correlation methods are unable to maintain the precision required for sophisticated nodes. Fusion Platform has implemented signoff engines within the optimisation process, adopting the correct-by-construction methodology.resulting in the delivery of an accurate signoff RTL-to-GDSII design-flow architecture with optimal PPA. It is true that accurate signoff analysis requires more computing than quick heuristics. Modern signoff engines from Synopsys are highly scalable and effectively manage speed and accuracy trade-offs across the different implementation stages. Additionally, Fusion Compiler uses machine-learning-driven methods from physical synthesis to signoff closure, utilising machine prediction where appropriate to significantly accelerate the optimization-solution-space search. Our customers consistently experience shorter time-to-results from RTL to signoff, because to faster design convergence and fewer revisions. Hyper-convergent de-sign platforms have a significant component in the golden-signoff backbone. The advantages of having shared engines have previously been expertly illustrated by Fusion Compiler, in this technology found in Fusion Compiler as well as the entire design pipeline from RTL to GDSII. For instance, cutting-edge technologies like the previously stated Vmin analysis in PrimeShield, the quick Monte-Carlo statistical engine in PrimeShield, and the recently developed HyperTrace technology in PrimeTime are all possible candidates for deployment consideration. By efficiently utilising these very distinctive technologies beyond their apparent function of improving signoff correlation, Fusion Compiler's PPA will be advanced to hitherto unachievable heights. As they get closer to becoming a product, I look forward to posting more.

The outcome of the synthesis process is the netlist, which serves as the basis for physical design. VHDL or Verilog HDL RTL designs are translated into gate-level specifications by synthesis so that the following collection of tools may understand them. This netlist includes a list of the cells used, their connections, the area used, and other details.

The chip is divided into distinct parts by the following partitioning phase. This process is mostly carried out to aid in placement and routing as well as to differentiate between different functional blocks. Partitioning is the process by which the design engineer divides the overall design into smaller blocks and then goes on to design each module during the RTL design phase.Block sizes, clock domain, functionality, timing criticality, and design hierarchy can all be taken into consideration while partitioning.

We determine each block's dimensions during the floor planning stage and arrange them in the proper locations on the chip. The purpose of this phase is to

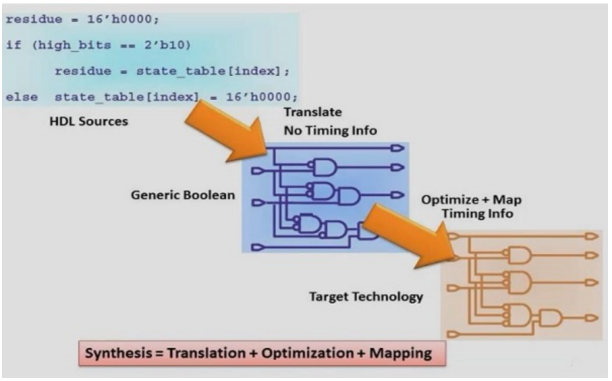

Figure 3.1: Synthesis Process

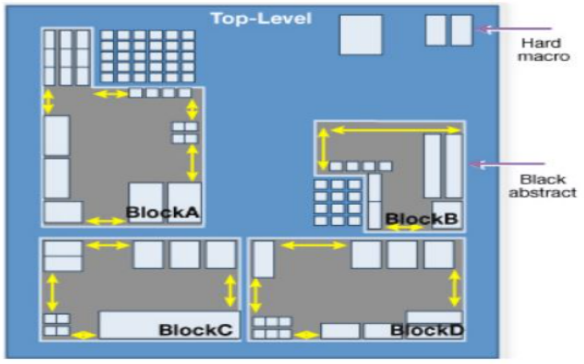

Figure 3.2: Partitioning

maintain the strongly connected blocks next to one another.

A power grid network is built at the power planning stage, which is usually a part of the floorplanning step, in order to distribute power evenly across the chip. electricity planning refers to supplying electricity to all standard cells, macros, and additional cells that are included in the design.Since Power Network Synthesis (PNS) is completed before to real signal and clock routing, power planning is also known as pre-routing.The core is the centre of the power ring design.Both VDD and VSS rings are present in power rings.Power mesh is nothing more than horizontal and vertical lines on the chip; after the ring is installed, power is planned such that power can easily reach all of the cells. The VDD and VSS rails must also be defined during power planning.Goal of power planning is to avoid IR dropout. Power planning include figuring out how many power pins are needed, how many rings and straps there are, how wide they are, and how much IR drop there is..

The procedure of positioning the standard cells inside the core border in an ideal location is known as placement. The tool attempts to arrange the standard

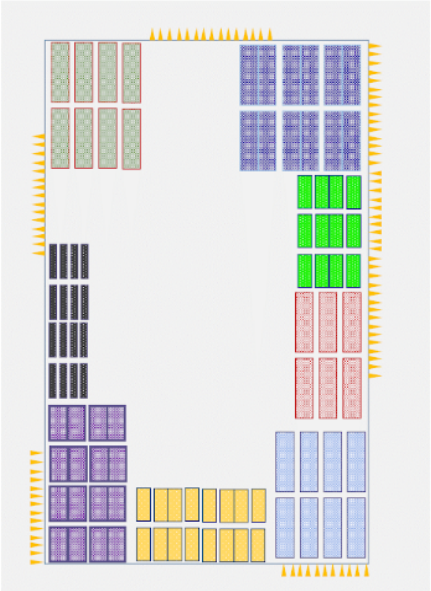

Figure 3.3: floorplanning

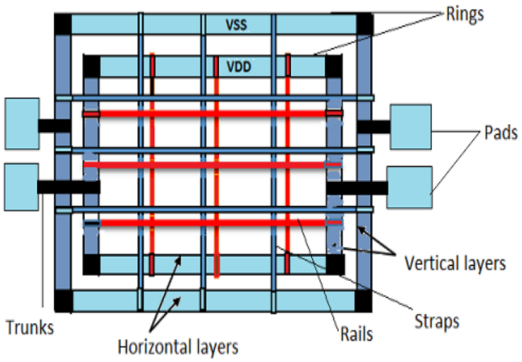

Figure 3.4: powerplanning

cell so that there are as few congestions and optimal timing in the design as possible. Each PnR tool offers a variety of instructions and switches to enable users to better optimise the design according to their needs in terms of timing, congestion, area, and power. The tool tray should be positioned and optimised for improved quality of reception based on the user's chosen preferences. In addition to placing the standard cells provided in the synthesised netlist, placement additionally involves adding buffers and inverters and placing numerous physical cells in order to meet timing, DRV, and foundry criteria. These areThese are the fundamental actions that the tool takes in the placement and optimisation phase.

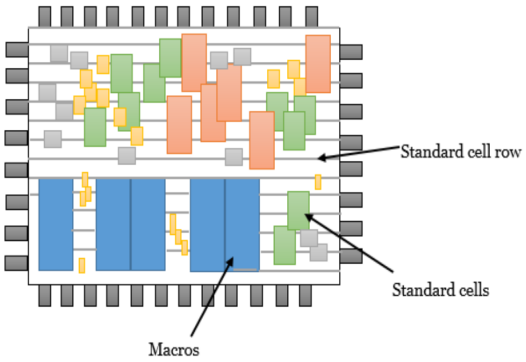

Figure 3.5: placement

A technique for verifying a design's timing performance is static timing analysis (STA), which involves searching every path for timing inconsistencies. STA divides a design into timing paths, determines the delay in signal propagation along each path, and looks for timing inconsistencies at the input/output interface and inside the design itself. Using dynamic simulation, which ascertains the complete behaviour of the circuit for a certain set of input stimulus vectors, is an additional method of timing analysis. Static timing analysis is substantially faster than dynamic simulation since it does not need simulating the circuit's logical process. Because STA examines all temporal routes rather than just the logical circumstances that are made more sensitive by a group of test vectors. But STA can only verify a circuit design's timing—not its operation.

One of the most important phases in the physical design flow of VLSIs is clock tree synthesis, or CTS. It is applied to minimise insertion latency and skew. This stage aids in distributing the clock among a design's sequential elements in an equitable manner.Clock Tree Synthesis, or CTS, is the process of joining the clock from the clock port to the clock pin of the design's sequential cells while utilising clock inverters and clock buffers to balance the skew between the cells and preserve the shortest insertion latency.Clock nets are typically included in the category of High Fanout Nets, however they are not included in the High Fanout Net Synthesis (HIFNS) since clock nets need to be adjusted to not touch. Since we cannot route like in a regular net, the clock uses between 30 and 40 percent of the chip's power and the EM impact is more common with clock networks. Clock nets must therefore add ICG.cells to manage the clock dynamic power and use certain criteria to do routing (such as building clock trees). Clock buffers and clock inverters are utilised during the clock tree construction process to handle skew and insertion delay.Clock tree structures include fishbone, pie, X, and H trees.

Routing facilitates the creation of connections between blocks and cells. Global

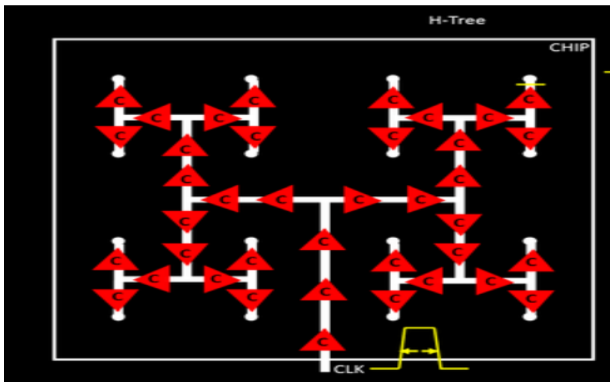

Figure 3.6: CTS

routing and detailed routing are the two different forms of routing. Global routing allocates routing resources and routes connections. It also monitors the assignment of a network. However, careful routing is used to create the actual links.

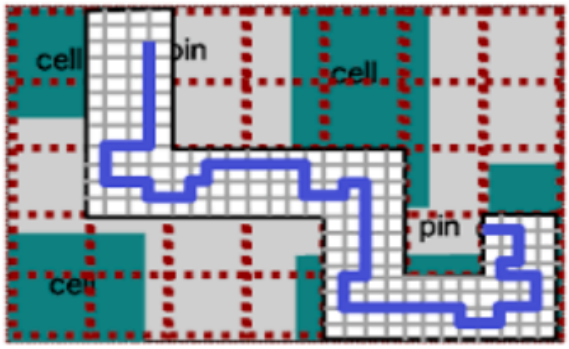

Figure 3.7: Routing

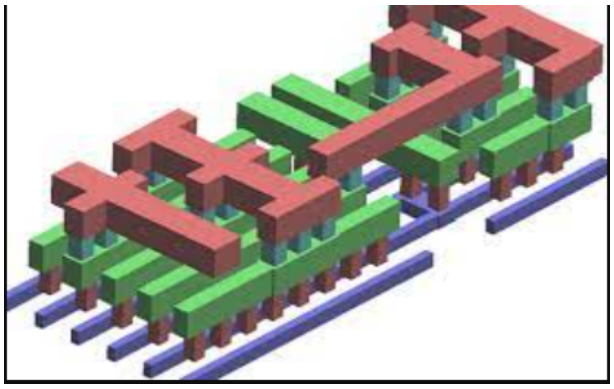

Figure 3.8: Routing

Chapter 4

Exploration of Each stage in fusion Compiler

4.1 The Need of Fusion Compiler

of the special RTL-to-GDSII architecture of Fusion Compiler, users may quickly achieve maximal differentiation and rethink what is feasible with their designs. Superior power, performance, and area are delivered right out of the box, and the return time is the fastest in the business. When Fusion Compiler is used in conjunction with DSO and AI, productivity increases even further and results that were before unthinkable happen quickly. Optimising RTL-to-GDSII unified engines opens up new possibilities for optimal area, power, and performance outcomes. The architecture of a single, integrated data model offers unparalleled productivity, scalability, and capacity. Power analysis, parasitic extraction, and signoff timing built in eliminate iterations of the design.

4.2 Fusion compiler steps

fusion compiler includes the entire PNR flow after the synthesis.One of the primary benefits of the Fusion compiler is its ability to handle both synthesis and backend design. On the other hand, the norma compiler uses a distinct tool or compiler for synthesis and a different compiler for location and route..

4.2.1 Init design

first steps in the fusion compiler is the init design. The init design is the first database in the fusion compiler. The initi design doing the complete functionality of the synthesis. In the init design the first that database open all the verilog file or vhdl file and check the error related to the hardware description language along with that these database doing the part of the floor planing and power plan-

ing. Where as these two steps is we have to do individually in the different compiler.

4.2.2 Compile fusion

The compile fusion it the next genration synthesis and ICC IIs placement and optimization. The compile fusion is doing with fewer iterations of placement and optimization as compared to the traditional DC+ICCII flows. Unifies all pre route optimization on to common engines, which enables consistent costing in optimization algortihms i.e same timer, same routinng estimation and so on. The compile fusion shares the best synthesis and pr technologies throughout the full pre route flow..

4.2.3 Clock opt cts

Clock Tree synthesis is one of the most important stages in pnr. CTs Qor decides timing convergence and power. In most of the ICs clock consumes 30-40 percent of total power. So efficent clock architecture, clock gating and clock tree implementation helps to reduce power. The porcess of distributing the clock and balancing the load is called CTS. Basically delivering the clock to all sequential elements. CTS is the porcess of insertion of buffers or inverters along the clock paths of ASIC design in order to acive zero or minimum skew or balanced skew. Before CTS, all clock pins are driven by single clock source. CTS starting pint is clock source and cts ending point is clock pins of sequential cells.

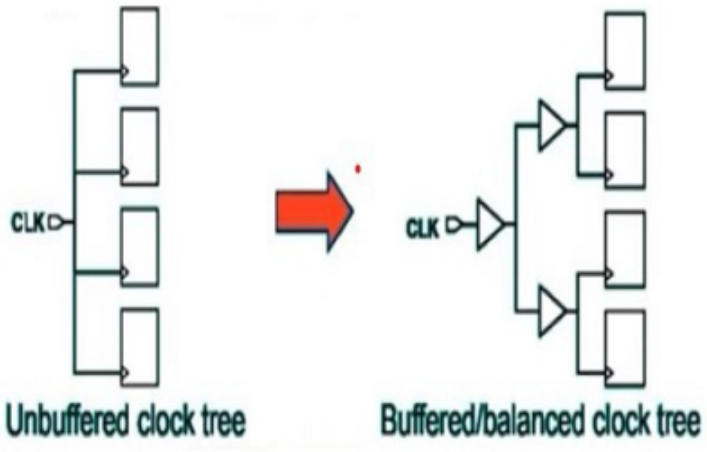

Figure 4.1: Clock opt cts

4.2.4 Clock opt opto

In the clock opt cts is done then the clock opt opto is done. In these stage the clock optimization is done. The clock is optimizing in the context of optimize the useful skew. The second thing can be done in the cts optimization is the optimization of the total negative slack. The post cts optimization techniques consist of the shielding the sizing buffer re location level.

4.2.5 Route auto

In the route auto stage the router engine will come in to the picture. In that satge the main purpose of the route auto engine is use to route all the standard cell base on the allocated metal layer in the technology file . The main moto of the route auto engine is just to route without having any checks likes drc and minimize the wirelength of the nets as well as in these stage track assignment is done with couple of itteration for the routing of the standard cell by considering all the constraints with minimize the timing

4.2.6 Route opto

In the route opto stage the route engine will try to optimize the routing in therms of the wirelength and also it will performed couple of the itteration for the optimization of the wire length and also it will try to reduce the drc counts in terms of opens and shorts .

4.2.7 Endpoint opt

These is the last stage in the fusion compiler. The small and incremental change is done at these stage. These stage is the stage again optimization is done at thes stage in terms of area , power and timing.

Chapter 5

Runtime Improvement of the Every stage of the fusion compiler

5.1 The Need of Runtime Improvement of the Fusion Compiler

The runtime improvement in physical design for the Fusion Compiler tool is a significant advancement in the field of electronic design automation (EDA). Fusion Compiler, developed by Synopsys, is an integrated RTL-to-GDSII solution for designing advanced semiconductor chips. Here's how runtime improvement in physical design benefits the Fusion Compiler tool. In the context of area reduction of runtime:

1.Faster Turnaround Time: Runtime improvements in physical design mean that tasks such as placement, routing, and optimization are completed more quickly. This results in reduced turnaround time for the entire chip design process. Designers can iterate more rapidly, exploring different design options and achieving faster convergence on the optimal design.

2.Increased Productivity: With faster runtime, designers can accomplish more in less time, leading to increased productivity. They can explore design space more comprehensively, experiment with different design strategies, and perform more iterations within the same time frame. This ultimately leads to higher-quality designs and better overall efficiency.

3.Support for Larger Designs: As semiconductor chips continue to grow in complexity and size, runtime improvements become essential for handling larger designs efficiently. Fusion Compiler's enhanced runtime capabilities enable it to handle the growing complexity of modern chip designs without sacrificing performance or requiring significant increases in computational resources.

4.Enhanced Design Exploration: Faster runtime empowers designers to explore a wider range of design options and trade-offs. They can quickly evaluate different architectural choices, optimization strategies, and design constraints, leading to more informed decisions and ultimately better-performing chips.

5.Competitive Advantage: By delivering faster turnaround times and improved productivity, runtime improvements in physical design give semiconductor companies a competitive edge. They can bring their products to market faster, respond more quickly to customer demands, and stay ahead of competitors in an increasingly fast-paced industry.

6.Cost Savings: Reduced runtime translates directly into cost savings for semiconductor companies. Shorter design cycles mean lower development costs, as fewer resources are required to complete the design process. Additionally, faster time-to-market allows companies to capture market opportunities more quickly, maximizing revenue potential.

Chapter 6

Work done in Synopsys

6.1 Scope of The Training

The main objective of the training programme is to gain practical knowledge and valuable experience in the VLSI Industry, with the world-changing EDA and semicounductor company . The great outcome of this training is to gain the ability to analyse, recognize, frame and model challenging problems and finding executing the engineering solution. Following are the objectives of the project:

To understand the need of Backend flow

To understand the whole fusion compiler flow

To understand the what R and D suggestion regarding the run time

To understand the why runtime is degraded with respect to baseline

To understand the different aspect which is responsible for the run time degradation

To observe and analyse all the steps in Fusion compiler tool .

One of the most important steps in making sure the developed integrated circuit satisfies its performance criteria is to correct timing problems in VLSI (Very Large Scale Integration) design. Clock skew, setup and hold time violations, and general performance degradation of the circuit can all be caused by timing problems. To address timing difficulties in VLS, use the following general procedures and methods: Use Synchronous Design Practices.

To guarantee that the clock signal is distributed evenly throughout the whole design, use synchronous design techniques. This lessens the chance of clock skew problems..

Synthesis of Clock Trees (CTS). To construct an optimal clock distribution network, use clock tree synthesis. To improve timing performance, CTS makes sure that clock signals arrive to all successive elements with the least amount of skew. Analysis of Setup and Hold Times: To find violations, analyze setup and hold times.

To fulfill setup and hold time requirements, modify the critical path's routing and placement. Analyzing and designing incrementally: Divide the design into manageable chunks so that it may be examined gradually. This makes it possible to optimize crucial paths specifically and to identify time problems more quickly. Enhance Crucial Routes: Determine the important design pathways and pay attention to them. Improve these approaches by adding pipeline stages, changing the logic pieces used, or rear- ranging the important components. Add Registers or Buffers: Strategically place registers or buffers along important pathways to reduce signal latency and enhance timing performance. Meeting setup and hold time requirements can be aided by buffer insertion. Modify the latency of the clock: To solve clock skew problems, adjust the clock latency. One can use strategies like clock skew scheduling to balance the clock arrival times at various locations on the chip.

6.2 Tools

The single tool is being used for whole synthesis as well as pnr which is fusion compiler. Earlier for the synthesis DC compiler is there and for the pnr ICC-II is there. Fusion compiler combines both with single shell or we can say the fusion compiler is the single shell tool.

6.2.1 Fusion Compiler

The tool fusion compiler is traditional as compared to the existing tools like dc for synthesis and icc2 for the pnr. The fusion compile is a single shell tool which is doing both synthesis and pnr. The terms releated to the fusion compiler is as shown below:

a. Xterm Machine

b. fc shell

c. fc gui

d. Make files

e. log files

f. Proc report

g. reac

Xterm Machin: There are two ways to open the xterm machine and these are using batch normal mode and using interactive mode through both the ways we can start our run and analyze the design.

fc Shell: To invoke the fc shell first up all we have to load the machine either in the batch normal mode and the interactive mode. once done that we have to load the machine which have different version are available for the machine.

log files: The log files are genrated side by side as the steps in the fusion compiler gets executed. The logs file s give the information about the while during the init design what are the tool command language file i.e tcl file they have source while running the init design and also it give each and every stages about what init design database is doing.

Proc reports: The proc reports gives the information regareding the falling path that are voilating the timing and it also gives the information about the group paths and also give the global timing report.

Reac: The reac is on of the file that is used to show the result at the terminal for each and every stage. The reac is helpful to get the idea regarding each and every stage and what are the result in each and every stage for the particular design.The reac will give the information regarding the overall QoR of the design in the table form.

6.2.2 Regarding the information about the DDR block/IP

DDR IP (Double Data Rate Intellectual Property) is a term used in VLSI (Very Large Scale Integration) design to describe functional blocks or modules that have been pre-designed and pre-verified and that make it easier to implement DDR (Double Data Rate) memory interfaces in integrated circuits (ICs) or electronic systems.

Definition and Importance: DDR IP cores are crucial parts of contemporary semiconductor designs, particularly for those that need memory connections that are fast and effective.In order to guarantee dependable data transfer at fast speeds, these IP cores are in charge of controlling communication between the integrated circuit and DDR memory devices.

Functionality:DDR IP cores perform a number of tasks required for DDR memory interfacing, such as address and command decoding: converting instructions and addresses from the CPU or controller into signals that the DDR memory devices can understand. Controlling the order and timing of data transfers between the IC and DDR memory is known as data transfer control. Signal conditioning: Timing, voltage level, and impedance matching are used to maintain signal integrity. Error detection and correction: Putting systems in place to identify and fix potential transmission-related data problems.

Implementation: As changeable and customisable IP blocks, DDR IP cores are commonly used in VLSI designs, enabling designers to customise the functionality and performance to meet unique needs. Most of the time, memory interface IP specialists from outside vendors or semiconductor companies with in-house memory controller design knowledge provide these IP cores.

Integration: VLSI designers use hardware description languages (HDLs) like Verilog or VHDL to incorporate DDR IP cores into their chip designs. The memory controller and other necessary components are connected to the DDR IP cores, which are instantiated inside the overall design.

Benefits: DDR IP cores offer pre-verified solutions for DDR memory interfacing, which streamlines the design process for VLSI designers. Because proven designs may be used instead of creating memory interfaces from scratch, they save development time and effort. DDR IP cores lessen the possibility of memory device

interoperability problems by assisting in ensuring compatibility and compliance with DDR memory standards.

Applications: DDR IP cores find applications in a wide range of electronic systems requiring high-speed memory interfaces, including: Consumer electronics (e.g., smartphones, tablets, gaming consoles) Networking equipment (e.g., routers, switches) Computing systems (e.g., servers, PCs) Automotive electronics Industrial automation.

6.2.3 Comparison of quality of results and runtime degradation for the baseline and latest run

In these experiment we have performed the trial experiment with respect to the baseline and we are checking the runtime degradation of the every stage with respect to the baseline and if it is degraded then what could be the reason for the degradation. for the overall stage the run time degradation is there. A many reason for the run time degradation like the congestion or cell density so if the cell density is high tool will put more effort to place them within the given specified clamping density seted in the design and at the same time tool will take care about the timing degradation should not be there without failing all the constraints.

6.2.4 Comparision of the compile stage with respect to the runtime

lets first discussed the runtime degradation and quality of the result for the first stage in the fusion compiler.Here from the above table it seems that the overall runtime for the compile stage is begin improved by 1 hours here we can see that the baseline run time is 44 hours and the latest run has 43 hours but we have high degradation in quality of the results total negative slack has get improved but at the same time we have degradation of worst negative slacks.But the no improvement in the utilization.

Now the congestion wise i have checked so how the tool is analyzing the congestion is based on the global route cell after the placement the tool try to divide the whole core area in to the bin or we can say global route cells based on that the no of the metal layers going through the cells it will calculate both horizantal as well as vertical and base on that he will find the demand and supply nets if the demand is greater than the supply net then we will face the congestion.

Compile Stage	Baseline Run	Latest Run (latest fc_version + latest ERI app option + SDP)
Fc shell Version	U-2022.12-SP4	U-2022.12-SP6
ERI Version	ERI -20230820(Ignored By tool)	ERI -20230820(Applied Successfully)
Scenario	4 active 2 setup, 2 hold, 4 lkg, 0 dynamic, 2 mtran	4 active 2 setup, 2 hold, 4 lkg, 0 dynamic, 2 mtran
Runtime	44 hrs	43 hrs
WNS	-567.3(ps)	-584.2(ps)
TNS	1329.7329(ns)	1158.3538(ns)
R2RTNS	36.7782 (ns)	78.6659(ns)
MTran	14.8422	9.3107
NVP	Setup 20567 Hold 30937	Setup 23784 Hold 28957
Instance count	4064474	4100555
Utilization	49%	49%

Figure 6.1: comparison table for the compile stage

6.2.5 Comparison of the clock opt cts stage with respect to the runtime

Here in the clock opt cts is the first stage in the clock tree will get build. There are different style through which we can build the clock tree like MSCTS and standard clock tree cts the main objective of the cts to balance the skew and minimize the latency. Here while we have to build clock tree we have to used proper style because the 50percent of the power consumption is due to the clocks tree for the overall asiic. So with respect to the baseline the run time is degraded by 1 and half hour with respect to the baseline but the look at the the qor i.e at wns and tns both are improved too much as in the latest one so as the clock tree build the instance count is increased that is good for the setup and hold so due to that the setup and hold time is get improved and utilization is same no degradation in the utilization otherwise that will leads the increase the pin and cell density and also affects the routing congestion as well.

Stage(Clock_opt_cts)	Baseline Run	Latest Run (latest fc_version + latest ERI app option + SDP)
Fc shell Version	U-2022.12-SP4	U-2022.12-SP6
ERI Version	ERI -20230820(Ignored By tool)	ERI -20230820(Applied Successfully)
Scenario	4 active 2 setup, 2 hold, 4 lkg, 0 dynamic, 4 mtran	4 active 2 setup, 2 hold, 4 lkg, 0 dynamic, 4 mtran
Runtime	13.51 hrs	14.39 hrs
WNS	- 2.1455(ns)	-1.0868(ns)
TNS	9039.4180(ns)	2610.7551(ns)
R2RTNS	1007.2392 (ns)	634.8002(ns)
MTran	36.2255	51.0703
NVP	Setup 105286 Hold 241841	Setup 32952 Hold 236158
Instance count	4240820	4266287
Utilization	49%	49%
Shorts	17	15

Figure 6.2: comparison table for the clock opt cts stage

6.2.6 Comparison of the clock opt opto stage with respect to the runtime

At the clock opt opt stage the tool try to optimize the tree which is being created at the clock tree cts stage in terms of the skew and latency and try to maintain the less insertion delay which is the combination of the source delay and the network delay. In these stage also tool try to minimize the drc voilations shuch as opens and shorts. From the table we can observed that runtime is degraded by the 2 hours where as the baseline run is 35.45 hours and the latest run time is the 37.81 hours but we have good improvement in terms of the wns and tns and also register to register tns and also total no of path for the setup violation and hold violation and hold voilation are also less but utilization value is increase by 55 percentage.

Block:- Native Stage(Clock_opt_opto)	Baseline Run	Latest Run (latest fc_version + latest ERI app option + SDP)
Fc shell Version	U-2022.12-SP4	U-2022.12-SP6
ERI Version	ERI -20230820(Ignored By tool)	ERI -20230820(Applied Successfully)
Scenario	4 active 2 setup, 2 hold, 4 lkg, 0 dynamic, 4 mtran	4 active 2 setup, 2 hold, 4 lkg, 0 dynamic, 4 mtran
Runtime	35.45 hrs	37.81 hrs
WNS	- 2.1657(ns)	- 1.0599 (ns)
TNS	19415.3516(ns)	3505.7778(ns)
R2RTNS	1463.7772 (ns)	977.3240 (ns)
MTran	6.0946	9.7105
NVP	Setup 225873 Hold 138807	Setup 106633 Hold 70230
Instance count	4579770	4892311
Utilization	54%	55%

Figure 6.3: Comparison table for the clock opt opto stage

6.2.7 Comparison of the route opto stage with respect to the runtime

The route opt is the finial stage after which the route auto is done. In the route auto stage all routing is done track creation but does not have optimization is doing as only primary work is to route all the standard cells and clock and data path. In the route opt stage the optimzation is done in terms of the wirelength and and also try to reduce the drc like opens and shorts and many drc. From the table it is observe that the run time is degradation is of some 4 hours but we have good improvement in the qor like wns and tns ans also register to register tns values but in the latest run we have degradation in the mtrans but the improvement in the setup path voilation and we have degradation in the hold path voilations rest all we have degradation with respect to the base as well as the shorts counts is increase too much due to routing congestion and also instance count is also increase results in increase in utilization.

Block:- Native Stage(route_opto)	Baseline Run	Latest Run (latest fc_version + latest ERI app option + SDP)
Fc shell Version	U-2022.12-SP4	U-2022.12-SP6
ERI Version	ERI -20230820(Ignored By tool)	ERI -20230820(Applied Successfully)
Scenario	4 active 2 setup, 2 hold, 4 Lkg, 0 dynamic, 4 Mtran	4 active 2 setup, 2 hold, 4 Lkg, 0 dynamic, 4 Mtran
Runtime	8.10 hrs	12.19 hrs
WNS	-2.145 (ns)	-1.051 (ns)
TNS	-22893.3043 (ns)	-6033.1107(ns)
R2RTNS	-2697.8250 (ns)	-2693.2342 (ns)
MTran	17.71347	23.09755
NVP	Setup 255732 Hold 192238	Setup 126537 Hold 194498
Instance count	4579770	4892311
Utilization	54%	55%
Total DRC	5206	7682
Shorts	3880	5027

Figure 6.4: Comparison of the route opt stage with respect to the runtime

Chapter 7

Conclusion

7.1 Conclusion

By analyzing the reason for the high runtime we come across the couple of reason in the design such as the cell density as well as the congestion where the tool try to put more efforts to get the congestion driven placement without degradation of the qor much.

In this project, individually check the runtime of the every stages and the qor of the every stage with respect to the baseline.

References

[1] A. Li, B. Zheng, G. Pekhimenko and F. Long,"Automatic Horizontal Fusion for GPU Kernels", International Symposium on Code Generation and Optimization, Seoul, Korea,, Republic of, 2022.

[2] J. Lu and B. Taskin, ""From RTL to GDSII: An ASIC design course development using Synopsys® University Program,".", 011 IEEE International Conference on Microelectronic Systems Education, San Diego, CA, USA, 2011.

[3] Synopsys slove net

[4] Runtime cheat sheet by synopsys