Back End Design (Automatic Place and Route) of RISC-V Processor using IC Compiler

LOH JING EN

Master of Engineering (Hons) Electronic Engineering

Faculty of Engineering and Green Technology

UNIVERSITI TUNKU ABDUL RAHMAN

AUGUST 2023

BACK END DESIGN (AUTOMATIC PLACE AND ROUTE) OF RISC-V PROCESSOR USING IC COMPILER

By

LOH JING EN

A thesis submitted to

Faculty of Engineering and Green Technology,

Universiti Tunku Abdul Rahman,

in partial fulfilment of the requirements for the degree of

Master of Engineering (Hons) Electronic Engineering

August 2023

DECLARATION

I, Loh Jing En, hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature: [illegible]

Name: LOH JING EN

ID No: 21AGM06714

Date : 13 August 2023

APPROVAL FOR SUBMISSION

I certify that this project report entitled "**Back End Design (Automatic Place and Route) of RISC-V Processor using IC Compiler**" was prepared by Loh Jing En has met the required standard for submission in partial fulfilment of the requirements for the award of Master of Engineering (Electronics Systems) at Universiti Tunku Abdul Rahman.

Approved by,

Signature: [illegible]

Supervisor: Ir. Dr. Loh Siu Hong

Date : 14 August 2023

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude towards my supervisor, Ir. Dr. Loh Siu Hong for his guidance and support throughout the entire project. He always free up his time to have discussion regarding my progress and provide useful insights amidst his tight schedules. This project would not be a success without his consistent encouragement and support.

I would also like to thank the Faculty of Engineering and Green Technology, Universiti Tunku Abdul Rahman for the education and platforms provided, for me to engage myself in this field. It has been a really fruitful journey. I will make full use of the knowledge and experience gained in order to strive in my career in the future.

I would like to thank my family for their endless love and support to me. Their encouragements have inspired me to work hard to pursue my dreams.

Last but not least, I would like to give credit to the authors of the journals and papers that have contributed to my work.

ABSTRACT

BACK END DESIGN (AUTOMATIC PLACE AND ROUTE) OF RISC-V PROCESSOR USING IC COMPILER

Loh Jing En

This research paper focuses on the development of an Automatic Place and Route (APR) methodology for the RISC-V processor design using the IC Compiler tool. The proposed methodology is aimed at achieving good Quality of Results (QoR) for different technology nodes, including 32nm and 90nm. The paper provides a detailed analysis of the QoR obtained for each technology node and compares the results obtained with each other. On top of that, the effect of clock period on the design quality is also analyzed. The methodology used for the design flow and the physical implementation process of the design using IC Compiler are all explained in detail. The experimental results demonstrate the effectiveness of the proposed methodology in achieving good QoR for RISC-V processor designs.

Keywords: RISC-V, processor, technology node, physical implementation, IC Compiler

TABLE OF CONTENTS

Page

DECLARATION ii
APPROVAL FOR SUBMISSION iii
ACKNOWLEDGEMENTS v
ABSTRACT vi
LIST OF TABLES ix
LIST OF FIGURES x

CHAPTER 1 INTRODUCTION 1
1.1 Background 1
1.1.1 Processor 1
1.1.2 RISC-V ISA 1
1.1.3 Back End Design 4
1.2 Problem Statement 6
1.3 Objectives 7

CHAPTER 2 LITERATURE REVIEW 8
2.1 Performance of RISC-V Cores 8
2.2 Power, Timing and Area at Back End Design 9
2.3 Technology node in PPA 12

CHAPTER 3 RESEARCH METHODOLOGY 15
3.1 Flow of Research Methodology 15
3.2 Electronic Design Automation (EDA) tools 17
3.2.1 ModelSim-Intel® FPGA Edition – RTL simulation 18
3.2.2 Design Compiler - Synthesis 18
3.2.3 IC Compiler – Physical Design (APR) 18

CHAPTER 4 RESULTS AND DISCUSSION 24
4.1 Physical Implementation with IC Compiler 24
4.1.1 Data Setup 25
4.1.2 Floorplan 27
4.1.4 Clock Tree Synthesis (CTS) 32

4.1.5 Routing..34
4.2 Quality Of Results (QoR) Comparison Between 90nm and 32nm Technology Node 38
4.2.1 QoR comparison in Data Setup stage 39
4.2.2 QoR comparison in Floorplan stage 40
4.2.3 QoR comparison in Placement stage 42
4.2.4 QoR comparison in CTS stage...43
4.2.5 QoR comparison in routing stage 45
4.3 Quality Of Results (QoR) Comparison Between 5ns and 2ns clock period 49
4.3.1 QoR comparison in Data Setup stage 49
4.3.2 QoR comparison in Floorplan stage 51
4.3.3 QoR comparison in Placement stage 53
4.3.4 QoR comparison in CTS stage...54
4.3.5 QoR comparison in routing stage 56

CHAPTER 5 GANTT CHART 60

CHAPTER 6 CONCLUSION AND FUTURE WORK..61

References...63

LIST OF TABLES

	Page
Table 1.1: Back-end design flow description	5
Table 2.1: Design QoR in each design stage (Moreno, 2019))	9
Table 2.2: Design QoR in each design stage (Aradhya, et al., 2021)	10
Table 2.3: Design QoR in each design stage (Kanase & Nithin, 2021)	11
Table 2.4: Summary of Literature (Post-Routing data)	11 - 12
Table 2.5: Design QoR in each technology node (Melikyan, et al., 2018)	12
Table 2.6: Design QoR in each technology node (Xie, et al., 2015)	13
Table 4.1: Comparison between 90nm and 32nm in data setup stage	40
Table 4.2: Comparison between 90nm and 32nm in floorplan stage	41
Table 4.3: Comparison between 90nm and 32nm in placement stage	43
Table 4.4: Comparison between 90nm and 32nm in CTS stage	45
Table 4.5: Comparison between 90nm and 32nm in routing stage	49
Table 4.6: Comparison between 2ns and 5ns in data setup stage	51
Table 4.7: Comparison between 2ns and 5ns in floorplan stage	52
Table 4.8: Comparison between 2ns and 5ns in placement stage	54
Table 4.9: Comparison between 2ns and 5ns in CTS stage	56
Table 4.10: Comparison between 2ns and 5ns in routing stage	59

LIST OF FIGURES

Page

Figure 1.1: Base instruction formats of RISC-V (Andrew & Krste, 2017) 2

Figure 1.2: Back-end design flow 4

Figure 3.1: Research Methodology Flowchart 15

Figure 3.2: Difference between coarse placement and detailed placement (Ahmed, 2021) 20

Figure 3.3: Pre-CTS and Post-CTS clock path (Ahmed, 2021) 22

Figure 4.1: Layout window in IC Compiler 26

Figure 4.2: Power and Ground pins connected 26

Figure 4.3: report_clock to verify clock period 27

Figure 4.4: report_clock -skew to verify clock constraints 27

Figure 4.5: Initial floorplan 28

Figure 4.6: Layout after coarse placement and routing 29

Figure 4.7: Pre-placement area 30

Figure 4.8: Post-placement area 30

Figure 4.9: Layout post-place_opt 30

Figure 4.10: pre-psynopt 31

Figure 4.11: post-psynopt 31

Figure 4.12: legality report post-placement 31

Figure 4.13: clock tree summary pre-CTS 32

Figure 4.14: clock tree post-CTS 32

Figure 4.15: post-CTS GRC 33

Figure 4.16: post-CTS Cell density 34

Figure 4.17: Layout post-CTS 34

Figure 4.18: Checking ideal or high fanout nets pre-routing 34

Figure 4.19: Port-routing layout design 35

Figure 4.20: post-routing GRC 35

Figure 4.21: post-routing cell density 36

Figure 4.22: post-routing pin density 36

Figure 4.23: post-routing LVS 36

Figure 4.24: post-routing verify_pg_nets 37

Figure 4.25: operating condition comparison between 90nm (LHS) and 32nm (RHS) 38

Figure 4.26: cell size comparison between 90nm (LHS) and 32nm (RHS) 39

Figure 4.27: pre-PnR timing comparison between 90nm (LHS) and 32nm (RHS) 39

Figure 4.28: physical design data comparison between 90nm (LHS) and 32nm (RHS) 40

Figure 4.29: chip summary comparison between 90nm (LHS) and 32nm (RHS) 41

Figure 4.30: timing QoR comparison between 90nm (LHS) and 32nm (RHS) 41

Figure 4.31: area and utilization QoR comparison between 90nm (LHS) and 32nm (RHS) 42

Figure 4.32: Cell density comparison between 90nm (LHS) and 32nm (RHS) 43

Figure 4.33: Clock constrains in both 90nm and 32nm design 43

Figure 4.34: Clock tree QoR comparison between 90nm (Top) and 32nm (Bottom) 44

Figure 4.35: Clock tree layout comparison between 90nm (LHS) and 32nm (RHS) 44

Figure 4.36: Timing QoR comparison between 90nm (LHS) and 32nm (RHS) 44

Figure 4.37(a): Power QoR comparison (90nm) 45

Figure 4.37(b): Power QoR comparison (32nm) 45

Figure 4.38: DRC clean in both 90nm and 32nm design 46

Figure 4.39: LVS clean in both 90nm and 32nm design 46

Figure 4.40: Timing QoR comparison between 90nm (LHS) and 32nm (RHS) 46

Figure 4.41: Area and physical QoR comparison between 90nm (LHS) and 32nm (RHS) 47

Figure 4.42: Power QoR comparison between 90nm (Top) and 32nm (Bottom) 47

Figure 4.43: Cell count comparison between 90nm (LHS) and 32nm (RHS) 47

Figure 4.44: Cell density comparison between 90nm (LHS) and 32nm (RHS) 48

Figure 4.45: Pin density comparison between 90nm (LHS) and 32nm (RHS) 48

Figure 4.46: GRC comparison between 90nm (LHS) and 32nm (RHS) 48

Figure 4.47: cell size comparison between 2ns period (LHS) and 5ns period (RHS) 50

Figure 4.48: clock comparison between 2ns (LHS) and 5ns (RHS) 50

Figure 4.49: pre-PnR timing comparison between 2ns (LHS) and 5ns (RHS) 51

Figure 4.50: physical design data comparison between 2ns (LHS) and 5ns (RHS) 51

Figure 4.51: chip summary comparison between 2ns (LHS) and 5ns (RHS) 52

Figure 4.52: timing QoR comparison between 90nm (LHS) and 32nm (RHS) 52

Figure 4.53: area and utilization QoR comparison between 2ns (LHS) and 5ns (RHS) 53

Figure 4.54: Cell density comparison between 2ns (LHS) and 5ns (RHS) 53

Figure 4.55: Clock tree QoR comparison between 2ns (Top) and 5ns (Bottom) 54

Figure 4.56: Clock tree layout comparison between 2ns (LHS) and 5ns (RHS) 54

Figure 4.57: Timing QoR comparison between 2ns (LHS) and 5ns (RHS) 55

Figure 4.58(a): Power QoR comparison (2ns) 55

Figure 4.58(b): Power QoR comparison (5ns) 56

Figure 4.59: LVS violation in 2ns design 56

Figure 4.60: Timing QoR comparison between 2ns (LHS) and 5ns (RHS) 57

Figure 4.61: Area and physical QoR comparison between 2ns (LHS) and 5ns (RHS) 57

Figure 4.62: Power QoR comparison between 2ns (Top) and 5ns (Bottom) 57

Figure 4.63: Cell count comparison between 2ns (LHS) and 5ns (RHS) 58

Figure 4.64: Cell density comparison between 2ns (LHS) and 5ns (RHS) 58

Figure 4.65: Pin density comparison between 2ns (LHS) and 5ns (RHS) 58

Figure 4.66: GRC comparison between 2ns (LHS) and 5ns (RHS) 59

LIST OF ABBREVIATIONS

APR	Auto Place and Route
ASIC	Application-Specific Integrated Circuit
CMOS	Complementary Metal-Oxide-Semiconductor
CPU	Central Processing Unit
CTS	Clock Tree Synthesis
DEF	Design Exchange Format
DRC	Design Rule Check
ECO	Engineering Change Order
EDA	Electronic Design Automation
ERC	Electrical Rule Check
FPGA	Field-Programmable Gate Array
GDS	Graphic Database System
GPIO	General Purpose Input/Output
HDL	Hardware Description Language
IC	Integrated Circuit
IP	Intellectual Property
ISA	Instruction Set Architecture
LEC / LEQ	Logic Equivalence Check
LPE / PEX	Layout Parasitic Extraction
LVS	Layout Versus Schematic
PCB	Printed Circuit Boards
PDK	Process Design Kit
PG	Power and Ground
PPAC	Power, Performance, Area and Cost
QoR	Quality of Results

RISC	Reduced Instruction Set Computer
RTL	Register Transfer Level
SoC	System-On-Chip
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
VLSI	Very Large-Scale Integration

CHAPTER 1

INTRODUCTION

1.1 Background

1.1.1 Processor

Microprocessors are the fundamental building blocks of modern computing systems, powering everything from smartphones and laptops to servers and supercomputers. A microprocessor is an integrated circuit that contains a central processing unit (CPU) and other components, such as memory and input/output interfaces, all on a single chip. Microprocessors have evolved significantly since their introduction in the early 1970s. The first microprocessors, such as the Intel 4004, were simple devices with limited functionality. Over time, microprocessors became more powerful and sophisticated, with the introduction of features such as pipelining, multiple cores, and hyperthreading (David & Andrew, 2019).

1.1.2 RISC-V ISA

An instruction set architecture (ISA) can be understood as an abstract model of a computer. It basically describes the supported instructions, data types, registers, the hardware support for managing main memory, fundamental features (such as the memory consistency, addressing modes, virtual memory), and the input/output model. RISC-V (reduced instruction set computer, 5th generation) is an ISA that was originally created to enable computer architecture education and research but is now poised to become an industry standard open architecture (Waterman, et al., 2014), which does not require fees to use.

Users have high customization and specialization options with RISC-V architecture. Generally, there are three main categories of RISC-V ISA:

- RV32I: A base integer instruction set with 32-bit instruction length

- RV64I: A base integer instruction set with 64-bit instruction length
- RV128I: A base integer instruction set with 128-bit instruction length

This paper will focus on RV32I, which contains a fixed length of 32-bit instructions, held by 31 general-purpose registers x1-x31. Each register holds integer values, and the register x0 is hard-wired so that it always contains value 0. There is one additional user-visible register, which is the program counter, that holds the address of the current instruction. There are also extensions that enable variable-length instructions, as long as it is aligned on 16-bit boundaries (able to be divided by 16).

RISC-V ISA has 6 base instruction formats including immediate variants, which is shown in Figure 1.1.

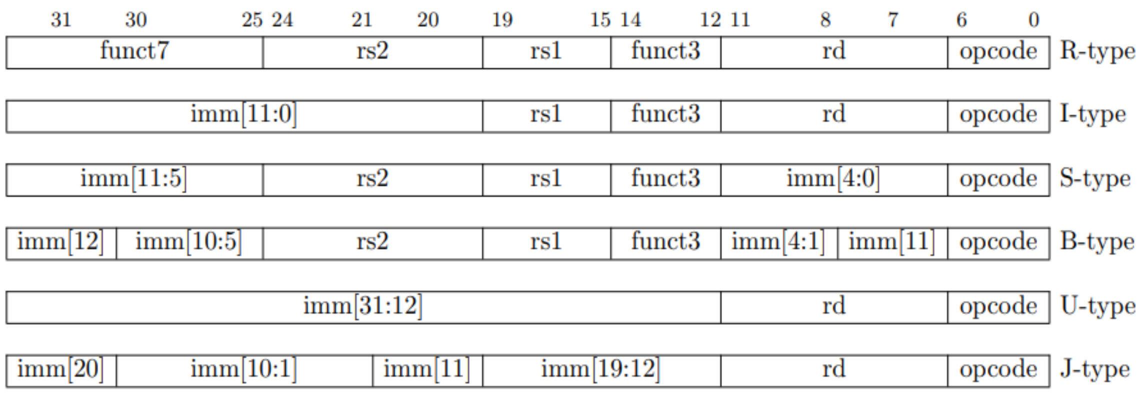


Figure 1.1: Base instruction formats of RISC-V (Andrew & Krste, 2017)

The *opcode* consists of 7 bits, which partially specifies which of the 6 types of instruction format listed above. *funct3* and *funct7* which consists of 10 bits, specifies the operation to be performed (examples are AND, OR, XOR). *rs1*, *rs2* and *rd* consists of 5-bits, which specifies the index of source register 1, source register 2 and destination register, respectively. *imm* is a constant operand or offset added to base address.

In semiconductor designs, the common goal is to achieve optimum combination of power, performance, area and cost (PPAC). PPA characterizes the physical limits and available resources of integrated circuits manufactured on that process node, while lower cost means more profit to the business. Different tradeoffs among the three variables enable various circuit improvements. For example, TSMC 16/12 nm process is said to be 50% faster and consumes 60% less power at the same speed compared to its own 20nm process (Anon., n.d.). This enhancement gives an edge in terms of performance and power consumption for next-generation high-end mobile computing, network communication, consumer, and automotive electrical applications.

One of the major impacts of RISC-V ISA is that RISC-V is available under open-source license, which means that anybody may contribute to its development, and it is free to use. Open-source nature of RISC-V is crucial because it allows smaller developers and manufacturers to design and build hardware without the cost of licensing proprietary ISAs and paying royalties. Research done by Gartner (2020) shows that by 2025, 40% of application-specific integrated circuits (ASICs) will be designed by OEMs, up from around 30% today.

RISC-V is also known for its simplicity and flexibility. In general, RISC-V serves as a descriptive framework for how software interacts with processors and offers a definition of how processor architecture should work. The capability to develop new processors is made possible by RISC-V. This is because instruction set is a combination of the processor and other design factors rather than being established at the ISA level only. In order to create processors for next-generation workloads, engineers and producers can therefore swiftly implement a minimum instruction set combined along with well-defined and custom extensions. In terms of performance and simplicity, RISC processors only use simple instructions that can be executed within one clock cycle. Because each instruction requires only one clock cycle to execute, the entire program can be executed in lesser number of instruction cycles. Case study done by Min (2022) shows speedup of 40 times when implementing custom RISC-V extensions, compared to pure C code. These RISC "reduced instructions" require less transistors of hardware space than the complex instructions, leaving more room for general purpose registers. Because all of the instructions execute in a uniform amount of time (i.e. one clock), pipelining is possible.

1.1.3 Back End Design

The backend design of a microprocessor is a critical stage in the overall design process. It involves the physical design of the microprocessor, including the placement and routing of the components on the silicon die. The goal of backend design is to optimize the microprocessor for performance, power consumption, and area efficiency.

Modern day Integrated Circuit (IC) design is split up into Front-end Design using HDLs and Back-end Design (also known as Physical Implementation or Layout Design). The inputs to physical design are netlist, library information on the basic devices in the design, and a technology file containing the manufacturing constraints. Physical design is usually concluded by Layout Post Processing, in which amendments and additions to the chip layout are performed. This is followed by the Fabrication or Manufacturing Process where designs are transferred onto silicon dies which are then packaged into ICs.

A general back-end design flow and brief explanation is listed in Figure 1.2 and Table 1.1.

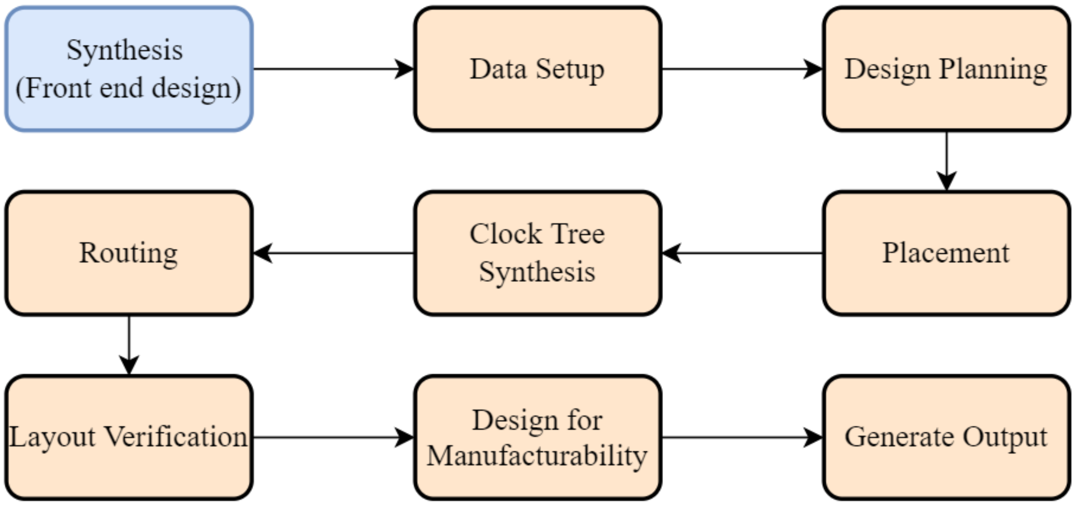


Figure 1.2: Back-end design flow

Table 1.1: Back-end design flow description

Stage	Description
Data Setup	The activities performed in data setup includes: • Specifying the appropriate logic libraries • Creating a Milkyway design library which contains technology file information, pointers to physical or reference libraries and initial design cell once netlist is loaded • Loading the netlist, constraints and RC parasitic models • Applying timing and optimization controls • Performing checks on libraries, RC parasitic models, constraints and timing
Design Planning	• Involves identifying which structures should be placed near others, taking into account area restrictions, speed, and the various constraints required by components.
Placement	• Before the start of placement optimization all Wire Load Models (WLM) are removed. Placement uses RC values from Virtual Route (VR) to calculate timing. VR is the shortest distance between two pins. VR RCs are more accurate than WLM RCs. • Determines the locations of each component or block on the die, considering timing and interconnect length. • The gates in the netlist are assigned to nonoverlapping locations on the die area.
Clock Tree Synthesis	• Inserting buffers or inverters such that the clock is distributed evenly to sequential elements in a design, minimizing skew and latency
Routing	• Determines the paths of interconnects, including standard cell and macro pins. This stage completes all connections defined in the netlist, ideally in the most efficient way and without violating timing constraints.
Layout Verification	Layout Verification checks the correctness of the generated layout design. This includes verifying that the layout: • complies with all technology requirements – Design Rule Checking (DRC) • is consistent with the original netlist – Layout vs. Schematic (LVS) • has no antenna effects – Antenna Rule Checking • complies with all electrical requirements – Electrical Rule Checking (ERC)
Design for Manufacturability	The design is modified, where possible, to make it as easy and efficient as possible to produce. This is achieved by adding extra vias or adding dummy metal/diffusion/poly layers wherever possible while complying to the design rules set by the foundry.
Generate Output	The final output of the physical design process is typically GDSII, a data format representing layout information

1.2 Problem Statement

The design of microprocessors using the RISC-V ISA is gaining popularity due to its open-source nature and flexibility in customization. Survey done by Semico predicts the market will consume 62.4 billion RISC-V CPU cores by 2025, a 146.2% compound annual growth rate (CAGR) from 2018 to 2025, while industrial sector leading with about 16.7 billion cores (Research, 2019). This means that RISC-V has a high potential in becoming popular and gaining more market sharing in near future. Therefore, it is necessary to understand the process of RISC-V processor design from RTL to GDS format that is used in fabrication process.

The design of processors requires specialized tools and expertise in order to achieve optimal performance and area efficiency. Physical Implementation or Back-End Design is one of the key components in processor design, which is handled by structural design engineers. During this step, the actual layout of the interconnected shapes that implement all the required circuit elements on the silicon wafer are created. It is important for engineers to explore and develop optimization techniques on improving power, performance and area of IC design, especially in back-end design where clocks are built, and routing occurs.

Current APR tools face several challenges when designing RISC-V processors, such as the increasing complexity of processor designs, the need to minimize power consumption, and the need to meet timing constraints. Additionally, there is a need to optimize the performance of the design while maintaining its area and power efficiency. Therefore, there is a need for research that addresses these challenges by developing new techniques and methodologies for the back-end design of RISC-V processors using APR tools.

The advancement in microelectronics technology has led to the development of microprocessors with increasing complexity and performance. The use of different technology nodes in microprocessor design allows for greater performance, power efficiency, and density.

A new technology node typically releases every 2 years. The semiconductor industry started with 10um technology node in 1971. As of 2022, Taiwanese chip manufacturer TSMC plans to put a 3 nm, semiconductor node termed N3 into volume production by the second half of 2022 (TSMC, 2022).

Current microprocessor design methodologies and tools are limited in their ability to optimize and maintain design consistency across different technology nodes. The existing design methodologies are based on a specific technology node and may not be directly applicable to other technology nodes. This leads to additional design efforts and costs when migrating the design to a new technology node. Therefore, there is a need for research that addresses these challenges and develops new methodologies for designing microprocessors using different technology nodes.

1.3 Objectives

- To identify the procedure for taking synthesized netlist through physical implementation to produce a GDS (graphic data system) file.
- To develop backend design methodology that provides satisfactory quality of results (QoR) in timing, power and area using different technology nodes.
- To compare and analyse the quality of results (QoR) of 90nm and 32nm technology library in backend design, and investigate how clock period affect design QoR.

CHAPTER 2

LITERATURE REVIEW

The success of this Instruction Set Architecture (ISA) design is demonstrated by the large number of RISC-V processor and framework implementations that are currently being developed. Numerous implementations can be adopted in both academic and industrial applications thanks to its free and open-source nature.

2.1 Performance of RISC-V Cores

The paper by Pasquale, et al. (2017) presents 2 novel RISC-V cores, ZERO-RISCY and MICRO-RISCY for Internet-of-Things (IoT) applications. The two cores are compared with the open-source RISCY core. The RISCY core is a simple, in-order 5-stage pipeline processor, with a single-precision floating-point unit. The ZERO-RISCY core is a more advanced processor with an out-of-order execution engine, a branch predictor, and a hardware divider. The MICRO-RISCY core is designed to be ultra-low-power, with a 2-stage pipeline, limited instruction set, and a small register file. The authors evaluate these cores based on their performance, power consumption, and area utilization. They use an open-source simulation environment to conduct the evaluation.

The results show that RISCY core is the most energy efficient when it comes to data-intensive tasks, while ZERO-RISCY us more efficient in arithmetic-control tasks. MICRO-RISCY outshines the other two core in pure control. Overall, the paper concludes that the choice of processor core should be made based on the specific requirements of the IoT application, taking into consideration the trade-off between power consumption, performance, and area.

2.2 Power, Timing and Area at Back End Design

As RISC-V is gaining its popularity among silicon developers, research has been made to explore the possibility to achieve optimum design in power, area and timing for various applications.

Moreno (2019) outlined the steps of designing a RISC processor from a register transfer level (RTL) design to the graphic database system (GDS) phase using 32nm library. The paper highlights the importance of each step in the design and fabrication process and discusses the challenges that arise during each step. The authors also provide a detailed description of the design decisions made at each stage and the trade-offs between power consumption, performance, and area. Useful commands and tips are provided in both front end and back-end design. The paper provides the QoR in different stages of IC design, which can be used as a guideline for this project.

Table 2.1: Design QoR in each design stage (Moreno, 2019))

Stage	**Cell Area (um^2)**	**Cell Count**	**Critical Path Slack (ns)**	
			Setup	**Hold**
Initial Synthesis	45941.05	7807	24.67	N/A
Synthesis with DFT	59481.86	N/P	24.65	N/A
Post-placement	62010.60	7903	24.51	N/A
Post-CTS	62140.85	7916	24.43	0.07
Post-routing	62140.85	7916	24.47	0.07

*N/P = not provided, N/A = not applicable for this stage

Ho, et al. (2021) presents a back-end implementation of a Dual-core 64-bit RISC-V using digital ASIC design flow with hardware construction language Chisel. The technology library used is 7nm FinFET from TSMC. From the study, back-end design is done using Cadence Genus and Innovus, with clean physical verification and timing verification results. The work shows that total dissipation switching power before CTS was low, because the clock elements, which consumed the most power in design, was not built. Post-CTS, the switching power increased by 1.9 times. In terms of timing, setup timing is met, but hold timing is violated by 13ps in terms of worst negative slack (WNS). However, the hold time violation values are still within the allowable margin of 2% clock cycle which is 2ns, with clock frequency of 500MHz. The work achieved die dimension of 1.17 mm × 1.17 mm, die area is around 1.38 mm^2. The core may consume 493.7mW dissipation power at 500MHz.

Aradhya, et al. (2021) shows the design process of Harvard Structure RISC Processor from RTL to GDSII. It is found that Harvard structure 4-stage pipelined architecture processor increases the speed of the operation as compared to the Von-Neumann architecture. RISC architecture uses separate data and address buses for both instruction and data to fetching from the main memory system to reduce the delay for the circuit. The QoR of each design stage is presented, which focus on timing, area and power. After optimization, the end results are timing 9.236ps, power 531.55682mW and area 17067.7584μm^2.

Table 2.2: Design QoR in each design stage (Aradhya, et al., 2021)

Stage	**Cell Area (um^2)**	**Setup Slack (ns)**	**Power (W)**			
			Internal	**Switching**	**Leakage**	**Total**
Synthesis	17097.70	5.951	0.677	0.269	0.058	1.004
Pre-CTS	17067.76	7.559	0.409	0.143	0.056	0.608
Post-CTS	17067.76	10.811	0.398	0.077	0.056	0.532

Kanase & Nithin (2021) introduce a work of implementing RTL to GDSII flow for the RISC-V system core for medical applications. Due to trade-offs in power, performance and area, the work targeted in timing and power optimization, as medical applications require power and timing for reliability. The optimized design achieved total power consumed by circuit 3.793mW, total area consumed by circuit 72409.1μm^2 and optimized arrival time of circuit is 9.868ps.

Table 2.3: Design QoR in each design stage (Kanase & Nithin, 2021)

Stage	**Cell Area (um^2)**	**Setup Slack (ps)**	**Power (mW)**			
			Internal	**Switching**	**Leakage**	**Total**
Synthesis	53489	0.005	N/P	5.483	0.285	5.768
Pre-CTS	72395.7	0.011	2.853	0.649	0.293	3.795
Post-CTS	72409.1	0.0	2.853	0.647	0.293	3.793

Khan, et al. (2022) proposes a way for converting comprehensive open-source digital tools, ISA, IPs, and manufacture-able PDKs to tape out a minimalist RISC-V-based SoC, named GHAZI. The authors used the RISC-V ISA and implemented a 32-bit processor core with 5-stage pipeline and basic peripherals such as UART, GPIO, SPI, and Timer. The ASIC design was performed using standard cell libraries and the backend design flow included synthesis, placement and routing, and static timing analysis. The authors also provided a detailed analysis of the power, performance, and area of the GHAZI SoC. Although the paper does not provide the design post-routing QoR, it does report the timing and power analysis of the final synthesized design, which is able to operate at maximum frequency of 170 MHz and consume 216 mW of power. The authors have also provided the layout of the GHAZI SoC, which has a die area of 13.69mm^2. The results showed that the GHAZI SoC achieved competitive performance with low power consumption and small area.

The Post-routing QoR of aforementioned papers are summarized in Table 2.4. A variation between datasets can be observed across the research papers. This is because the design of IC is impacted by a lot of factors, include - but not limited to - technology node, EDA tool used, design constraints, design complexity and routing topology.

Table 2.4: Summary of Literature (Post-Routing data)

Literature	**Area (mm^2)**	**WNS Timing (ps)**		**Clock frequency (MHz)**	**Power (mW)**				**Technology Library / Tool**
		Setup	**Hold**		**Leakage**	**Internal**	**Switching**	**Total**	
(Moreno, 2019)	0.0621	24.47	0.07	20	0.883	N/P	N/P	N/P	32nm / IC Compiler
(Ho, et al., 2021)	1.3800	0	13.5	500	0.061	193.2	234.9	493.7	7nm FinFET / Cadence Innovus

(Aradhya, et al., 2021)	0.0171	10.8	N/P	N/P	56.3	398.4	76.8	531.6	180nm / Cadence Innovus
(Kanase & M, 2021)	0.0724	0	N/P	N/P	0.293	2.853	0.647	3.793	180nm / Cadence Innovus

*N/P = not provided

2.3 Technology node in PPA

The technology node (also known as process node, process technology or simply node) refers to a specific semiconductor manufacturing process and its design rules. Different nodes often imply different circuit generations and architectures. Generally, the smaller the technology node means the smaller the feature size, producing smaller transistors which are both faster and more power efficient.

Melikyan, et al. (2018) presents a comparison of power and delay of the ORCA processor based on the 14 nm and 32 nm technology nodes. The study used the Cadence Virtuoso tool to simulate the designs and analyzed the power consumption, delay, and energy efficiency of the ORCA processor at different operating frequencies. The authors found that the 14 nm ORCA design achieved better performance and energy efficiency compared to the 32 nm design

Table 2.5: Design QoR in each technology node (Melikyan, et al., 2018)

Technology Node	Frequency (MHz)	Setup Slack (ns)	Total Area (um^2)	Total power (mW)
32nm	200	3.11	705825.31	78.78
14nm	400	2.42	239894.807	27.77

Standard cell libraries for advanced 7nm FinFET technology node is developed in Xie et al. (2015). The standard cell libraries facilitated circuit synthesis, power and timing analysis to further extend Moore’s law into deeply scaled processes. The libraries support multiple supply voltages and threshold voltages devices, which enables voltage and frequency scaling and multi-threshold technology. When compared to conventional 14nm and 45nm CMOS circuits, the 7nm FinFET circuit has 5 times and 600 times lesser power consumption in super threshold voltage. While in high threshold voltage regime, the 7nm FinFET circuit has 10 times and 1000 times lesser power consumption, compared to 14nm and 45nm CMOS circuits respectively. When operating in the near-threshold regime, the 7-nm FinFET devices with normal and high threshold voltage can improve the energy efficiency by 7 times and 16 times on average, against the 14-nm bulk CMOS technology, respectively.

In terms of circuit speed, 7nm FinFET technology shows significant improvement over 14nm and 45nm CMOS circuits. In super threshold regime, 7nm FinFET circuit operates 3 times and 15 times faster than 14nm and 45nm CMOS circuits respectively. This is due to smaller gate size and parasitic capacitance in 7nm FinFET technology. Table 2.6 summarizes the QoR between 7nm, 14nm and 45nm technology node. The data is based on 16-bit adder operation.

Table 2.6: Design QoR in each technology node (Xie, et al., 2015)

Technology node	V_{dd} (V)	Clock period (ps)	Frequency (MHz)	Energy Consumption per operation (fJ)	Power (uW)	
					Dynamic	Leakage
FinFET 7nm	0.30	163.3	6123	0.251	1.34	0.2
CMOS 14nm	0.55	450.7	2219	0.762	1.28	0.41
CMOS 45nm	1.10	1010	990	495.8	489.2	1.65

With similar design constraints, Jim´enez (2021) reported that 22nm technology gives additional bandwidth in area and timing compared to 65nm technology. The design fitted easily in the same area with a wide margin for 22nm, which enables additional low power techniques to be added to the design for power optimization, without causing significant degradation to timing. Moreover, low power cell libraries are more complete in 22nm technology libraries, giving more optimization opportunities and better results. Respect to timing, 22nm technology enables increment of operating frequency from the original 200MHz at 65nm to 625MHz.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Flow of Research Methodology

Figure 3.1 shows the proposed research methodology flow. The effort will be mostly focusing on back-end design.

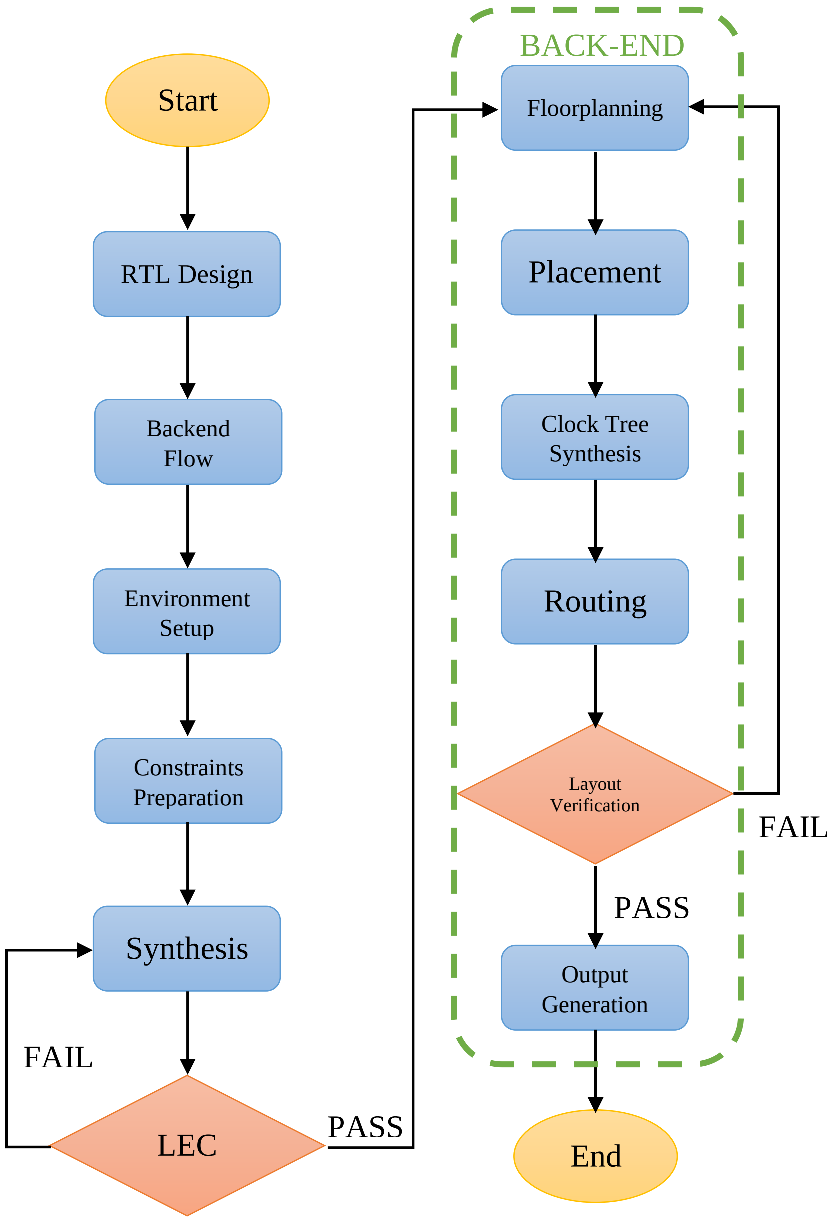


Figure 3.1: Research Methodology Flowchart

The main components of research methodology are elaborated below:

Step 1: RTL Design

Register-transfer level (RTL) design will be formulated with Verilog coding. Testbench will be used to test out functionality of RTL design. ModelSim-Intel® FPGA Edition will be used for RTL simulation and verification.

Step 2: Logic Synthesis

Once the RTL code has been obtained and it has been verified, the synthesis can begin. For this project Synopsys Design Compiler is the application being used to perform the synthesis. In synthesis the clock will be generated as well as any operating conditions being set based on the requirements of the chip. Logic Equivalence Check (LEC) will be done on the synthesized netlist against RTL design, to make sure it is matching with RTL coded.

Step 3: Physical Implementation

The major focus of this report is the Synopsys IC Compiler, which is an EDA tool for automated place and route (PnR). Synthesized design from Design Compiler will be imported and environment setup will be performed. In this project, Standard Cell Library of 90nm and 32nm technology node will be used. Floorplan will be generated based on the synthesized netlist and design constraints. The floorplan is then used to place all the cells from the netlist into the space of the chip so that they are all able to fit. Once cell placement is done, the clock tree is added. After the clock tree generation, all the cells are then routed, and all the connections are made. The process of floorplanning to routing is an iterating process to achieve satisfactory quality of results (QoR).

Step 4: Layout Verification

To verify the functional design throughout the design flow and make sure that design changes such as tools error, human error, mapping, optimization, and ECO changes do not affect the functionality of the RTL source code, the final step before taping out is layout verification. Layout verification is made to ensure the accuracy of an electrical circuit, to check design rules, to improve yield and to provide workable designs.

- Layout verification includes the following:
- Design Rule Check (DRC)
- Layout Versus Schematic (LVS)
- Electrical Rule Check (ERC)
- Layout Equivalence Check (LEQ)
- Layout Parasitic Extraction (LPE or PEX)

Step 5: Generate Output

The graphic data system II (GDSII) file format is generated for the use of fabrication.

3.2 Electronic Design Automation (EDA) tools

EDA tools are software applications used in the design, simulation, verification, and implementation of electronic systems. EDA tools are critical for the design and development of integrated circuits (ICs), printed circuit boards (PCBs), and system-on-chip (SoC) devices. They are used by design engineers to create, test and optimize electronic designs, and are an essential part of the electronics design process. In this project, tools below are used to simulate RTL script, perform synthesis and APR.

3.2.1 ModelSim-Intel® FPGA Edition – RTL simulation

One popular EDA tool for digital simulation is ModelSim-Intel® FPGA Edition, which is used for simulating and verifying designs for field-programmable gate arrays (FPGAs). It provides advanced debugging and verification features and supports the latest IEEE standards for hardware description languages (HDLs), such as Verilog and VHDL. ModelSim-Intel® FPGA Edition is commonly used in academic and research settings for teaching and research projects (Intel, 2019).

3.2.2 Design Compiler - Synthesis

Design Compiler is another EDA tool used for logic synthesis and optimization of digital designs. It takes a high-level register transfer level (RTL) design and generates a gate-level netlist that is optimized for the target technology library. Design Compiler is widely used in industry and academia for optimizing designs in terms of area, timing, and power (Synopsys, 2018).

3.2.3 IC Compiler – Physical Design (APR)

IC Compiler is a place-and-route (PNR) tool used for designing digital ICs. ICC takes input as a gate-level netlist, a detailed floorplan, timing constraints, physical and timing libraries, and foundry process data. It generates output either as a GDSII-format file of the layout or as a Design Exchange Format (DEF) file of placed netlist data ready for a third-party router. It is a is a single, convergent netlist-to-GDSII, chip-level physical implementation tool that includes placement and routing, clock tree synthesis, and optimization of timing and power. IC Compiler is used by chip designers to create high-performance, low-power, and area-efficient IC designs (Synopsys, 2019).

a) Floorplanning

The first major stage of physical design is floorplanning, where the die is initially created based on utilization targets and/or die size specifications. The size of die will be explored, as die size will ultimately contribute to production cost. Fast placement is performed on hard macros and standard cells, where initial validation and exploration of floorplan is performed.

Once the initial placement has been refined, the next step is power network synthesis, where the power mesh is automatically generated based upon power consumption estimates and IR-drop targets for the design. After the power network synthesis is done power network analysis can be conducted to analyze the mesh; confirming that the mesh meets the IR-drop constraints. Floorplanning is an important starting milestone for physical design, as it determines whether design meet timing and reliability (IR drop, electromigration) requirements, as well as ensuring the routability of design in later stage. A few design considerations need to be taken care before proceeding to placement:

- The area for core placement based upon the logic requirements of the logic, macros, memories and IP for the design
- The partitioning and placement for all of the logic, macros, memories and IP for the design
- Location and number of input/output pins and power/ground pins
- The creation of a complete PG mesh including necessary PG rings around block; PG straps and connectivity to analog components and, the creation of power straps for the entire chip while allowing enough remaining routing resources for all signal nets in the design.
- Density of P/G grid, whether it is sufficient

- The definition of hard blockage areas where cells must not be placed in order to reduce routing congestion to retain routability

b) Placement

The next major stage is placement. It is the process of placing standard cells in the design. IC Compiler determines the location of each standard cell and places them based on different criteria like timing driven, congestion driven and/or power optimization. Placement optimizes the design and also determines the design routability.

Generally, placement is divided into two subtasks: Global (coarse) and Detailed. These tasks are solved consequently. The aim of coarse placement is to get approximate initial location of standard cells. The cells are not legally placed and have a high chance of overlapping. Global placement is able to produce a complete placement from a partial or non-existent placement. It takes a negligible amount of computation time compared to detailed placement and provides a good starting point for them. The next step is detailed placement, where legalization of cell placement occurs. The cells in legalized placement will result in their legal orientation with no overlapping. Figure 3.2 shows the difference between coarse placement and detailed placement.

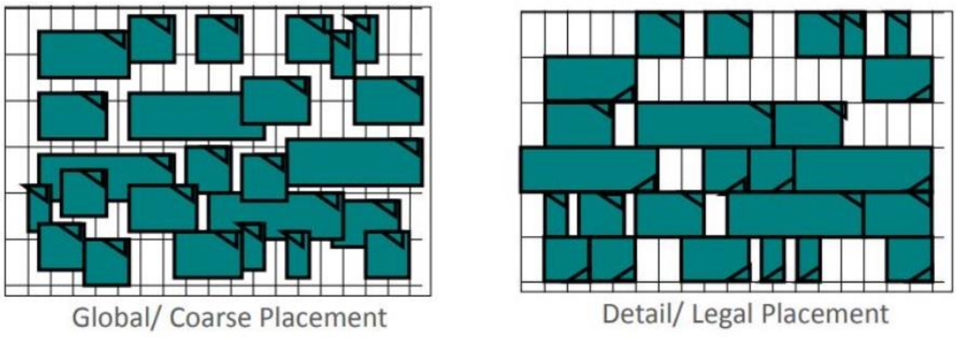


Figure 3.2: Difference between coarse placement and detailed placement (Ahmed, 2021)

A few design considerations need to be taken care before proceeding to CTS:

- Check whether there is any unplaced or overlapping cells – should be none.

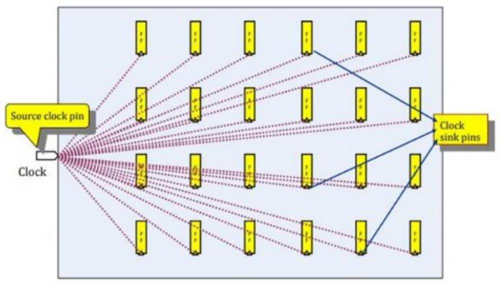
- Utilization of core area, generally the requirement is below 0.6.
- Post-placement timing, whether it is met or violated.
- Total design area

While floorplan and placement seem to be two distinct stages, the objectives are same, that is to minimize area, minimizing timing paths, reduce wire length to enable better routability and reduce IR drop. In situations where tool is unable to meet placement requirements, it is necessary to revisit floorplanning stage and make modifications.

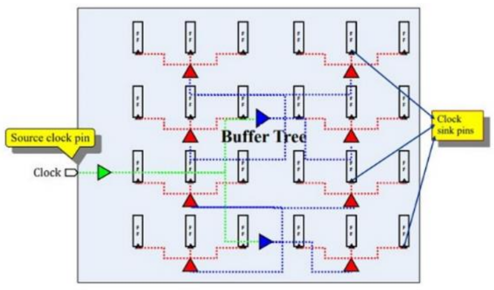
c) Clock Tree Synthesis (CTS)

Clock Tree Synthesis (CTS) is a process to ensure that the clock gets distributed evenly in order to balance the clock delay to all clock inputs in the IC. This stage determines the timing convergence and power of the design. In most cases, clocks will consume 30-40% of total power (Ahmed, 2021). Therefore, having an efficient clock architecture, clock gating and clock tree implementation will help to reduce power consumption, at the same time meeting timing requirements.

Insertion of buffers along clock path is performed in CTS, with the goal of delivering clock to all sequential elements. The goal is to achieve minimum skew. Figure 3.3 shows the difference of clock path before and after CTS. As seen in Figure 3.3, before CTS, all clock pins are driven directly by a single clock source. After CTS is performed, buffers are added to balance the clock, forming a clock tree. The starting point for CTS is the clock source, which most of the time is a port, and ending point is the clock pins of sequential cells.



(a) Pre-CTS



(b) Post-CTS

Figure 3.3: Pre-CTS and Post-CTS clock path (Ahmed, 2021)

A few design considerations need to be taken care before proceeding to routing:

- Both setup and hold timing, whether it is met or violated – group clock groups if necessary
- Report insertion delay and skew to verify targets are achieved
- Report fanout, capacitance and transition
- Check whether all intended sink points are being reached by clock
- Check DRC and routing constraints
- Check power and area whether it meets requirements

d) <u>Routing</u>

As technology shrinks, the routing process in physical design becomes more important because it determines the final performance and power consumption of the design. Smaller technology nodes have smaller feature sizes, which means that there is less space for routing wires and vias. This makes it difficult to route the design and can lead to increased capacitance and resistance, which can affect the performance of the design. Additionally, smaller technology nodes have higher transistor densities, which leads to higher power consumption. Proper routing techniques can help minimize the power consumption of the design by reducing the length of the interconnect wires and minimizing the number of vias. Therefore, routing plays a critical role in the success of a design in smaller technology nodes, and proper routing techniques are essential to achieve high performance and low power consumption.

In the placement phase, the exact locations of circuit blocks and pins are determined. A netlist is also generated which specifies the required interconnections. Space not occupied by the blocks can be viewed as a collection of regions for routing. During the routing phase, physical connections between signal pins are achieved using metal layers and vias. The connections are defined by logical connections present in netlist. After CTS stage, the information of placed cells, blockages, clock tree and I/O pins are used to complete remaining connections. The aim of routing is to produce a design with minimum DRC violations, fully routed design with minimum LVS violations, minimum congestion or hotspot, setup and hold timings are met, and QoR is satisfactory.

There are two types of routing methodology – global routing and detailed routing. During global routing, the routable path between pins is identified based on shortest distance, in order to achieve minimum delay. The tool routes the nets such that routing blockages, congested area and long detours are avoided. Routing layers are assigned to the nets, and net segments are allocated based on specific routable window known as Global Route Cell (GRC).

Detailed routing continues the routing action done by global routing. It performs complete DRC aware and timing driven routing. The basic criterion of detailed routing is the minimum area of interconnect as stated in the design rules. The detail router places the actual wire segments within the region defined by the global router to complete the required connections between the ports. It is the final routing for the design built and the timing is freeze. After confirming that all timing and QoR is met, layout verification is performed before design tape out.

CHAPTER 4

RESULTS AND DISCUSSION

Chapter 4 will be focusing on three topics aligning to project objectives. Section 4.1 will be discussing the process of bringing synthesized netlist (output from Design Compiler) to Physical Implementation. It involves data setup, floorplan, placement, clock tree synthesis and routing. Last but not least, the generated layout design is written as output in ddc and verilog format. The parasitics information (post-layout resistance and capacitance) are extracted into a SPEF file to be used in static timing analysis.

On top of that, section 4.2 and 4.3 involves QoR comparison between tech nodes, and how will clock period affect the QoR. The comparison will be on i) 90nm tech node versus 32nm tech node, both with 5ns period; and ii) 32nm tech node with 5ns period versus 2ns period.

4.1 Physical Implementation with IC Compiler

The steps for backend design are discussed in this section. All snapshots are based on 90nm tech node, 5ns period design, and QoR reports will be shown in section 4.2 for comparison with 32nm tech node design. Before starting back-end design with using IC Compiler, setup file containing useful functions and variables is coded, snapshot is shown below.

```
#Snapshot content of .synopsys_dc.setup

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# Logic Library settings
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
lappend search_path ../ref/db ../ref/tlup
set_app_var target_library "sc_max.db"
set_app_var link_library "* sc_max.db io_max.db special_max.db ram16x128_max.db
ram4x32_max.db ram8x64_max.db ram32x64_max.db"
set_min_library sc_max.db -min_version sc_min.db
set_min_library io_max.db -min_version io_min.db
set_min_library special_max.db -min_version special_min.db
set_min_library ram16x128_max.db -min_version ram16x128_min.db
set_min_library ram4x32_max.db -min_version ram4x32_min.db
set_min_library ram8x64_max.db -min_version ram8x64_min.db
set_min_library ram32x64_max.db -min_version ram32x64_min.db
```

```
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# ORCA setup variables
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
set my_mw_lib    riscv_pip_27.mw
set mw_path      "../ref/mw_lib"
set tech_file    "../ref/tech/cb13_6m.tf"
set tlup_map     "../ref/tlup/cb13_6m.map"
set tlup_max     "../ref/tlup/cb13_6m_max.tluplus"
set tlup_min     "../ref/tlup/cb13_6m_min.tluplus"
set top_design   "riscv_pip_27"
set ddc_file "../design_data/design_90.ddc"
set ctrl_file    "./scripts/data_setup/opt_ctrl.tcl"
set derive_pg_file    "./scripts/data_setup/derive_pg.tcl"
set libs         {sc io special ram4x32 ram8x64 ram32x64 ram16x128}
set mw_ref_libs ""
foreach lib $libs {
        lappend mw_ref_libs $mw_path/$lib
}
```

4.1.1 Data Setup

Gate-level netlist or synthesized netlist is provided by front-end design engineer as input for physical implementation. First, a Milkyway design library is created, and the gate-level netlist generated from Design Compiler is sourced. TLU+ models are loaded for accurate parasitic modeling. Layout window popped up, indicating that design is linked, indicated in Figure 4.1.

```
# Data Setup
#
file delete -force $my_mw_lib
#
############################################################
# Create Milkyway Design Library, Import design and Load TLU+ files
############################################################
create_mw_lib $my_mw_lib -open -technology $tech_file \
       -mw_reference_library "$mw_ref_libs "

import_designs $ddc_file \
       -format ddc \
       -top $top_design

set_tlu_plus_files \
       -max_tluplus $tlup_max \
       -min_tluplus $tlup_min \
       -tech2itf_map  $tlup_map
```

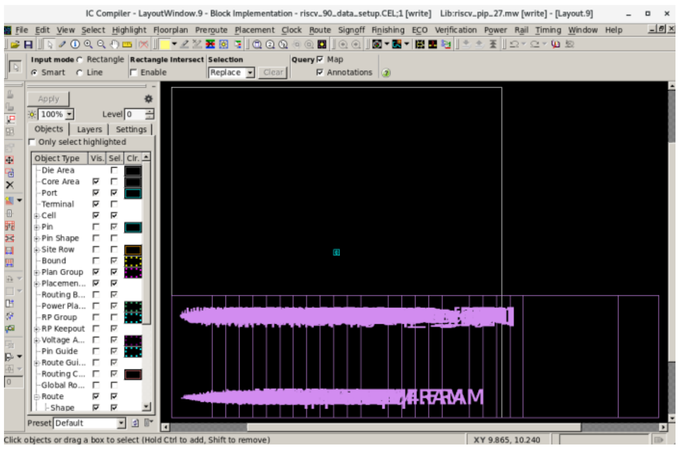

IC Compiler - LayoutWindow.9 - Block Implementation - riscv_90_data_setup.CEL;1 [write]   Lib:riscv_pip_27.mw [write] - [Layout.9]
File Edit View Select Highlight Floorplan Preroute Placement Clock Route Signoff Finishing ECO Verification Power Rail Timing Window Help
Objects Layers Settings
Click objects or drag a box to select (Hold Ctrl to add, Shift to remove)
XY 9.865, 10.240

Figure 4.1: Layout window in IC Compiler

Sanity check is done to make sure all libraries are linked successfully, followed by power and ground connection. From the terminal, we can see that 7029 power and ground pins are connected successfully.

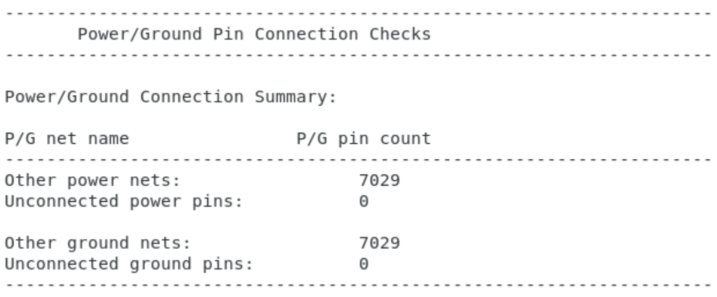
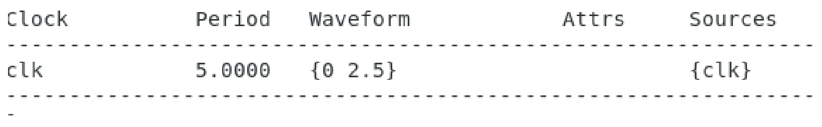

```
---------------------------------------------------------------------
       Power/Ground Pin Connection Checks
---------------------------------------------------------------------

Power/Ground Connection Summary:

P/G net name                 P/G pin count
---------------------------------------------------------------------
Other power nets:                 7029
Unconnected power pins:           0

Other ground nets:                7029
Unconnected ground pins:          0
---------------------------------------------------------------------
```

Figure 4.2: Power and Ground pins connected

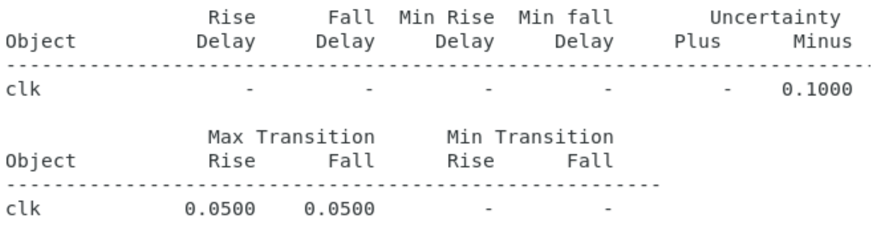
```
check_library
check_tlu_plus_files
list_libs
source $derive_pg_file
check_mv_design -power_nets

report_clock
report_clock -skew
set_route_mode_options -zroute true

check_mv_design -power_nets > ./reports/data_setup_check_mv_design

source $ctrl_file

source scripts/data_setup/zic_timing.tcl
exec cat zic.timing

save_mw_cel -as riscv_90_data_setup
```

Front-end designer has set the clock period to 5ns alongside with some constraints. The report_clock command is used to verify the clock period. As shown in Figure 4.3, the clock period is 5ns, which matches the settings in Design Compiler earlier. Clock uncertainty of 0.1ns and max transition 0.05ns also matches information from front-end design. The design is then saved as input for floorplan stage.



```
Clock          Period   Waveform          Attrs     Sources
---------------------------------------------------------------
clk            5.0000   {0 2.5}                     {clk}
---------------------------------------------------------------
-
```

Figure 4.3: report_clock to verify clock period



```
               Rise      Fall  Min Rise  Min fall        Uncertainty
Object        Delay     Delay     Delay     Delay     Plus       Minus
-----------------------------------------------------------------------
clk               -         -         -         -        -      0.1000

               Max Transition      Min Transition
Object         Rise      Fall      Rise      Fall
-------------------------------------------------
clk          0.0500    0.0500         -         -
```

Figure 4.4: report_clock -skew to verify clock constraints

4.1.2 Floorplan

During floorplan stage, a chip level floorplan that contains block size, I/O pins locations and power network synthesis is created. The gates are roughly placed in this stage. The first step in floorplan stage is to create floorplan. The core utilization is being set to 0.3, and the distance between core area and terminals are set to one (1) microns. Next, power rings and power straps are created. METAL3 and META4 are chosen as the layer for VDD and VSS power rings. We can see that a rectangular floorplan is created, with the cells arranged on the right side of floorplan. The red and green mesh is the power straps, where power supply is connected to cells.

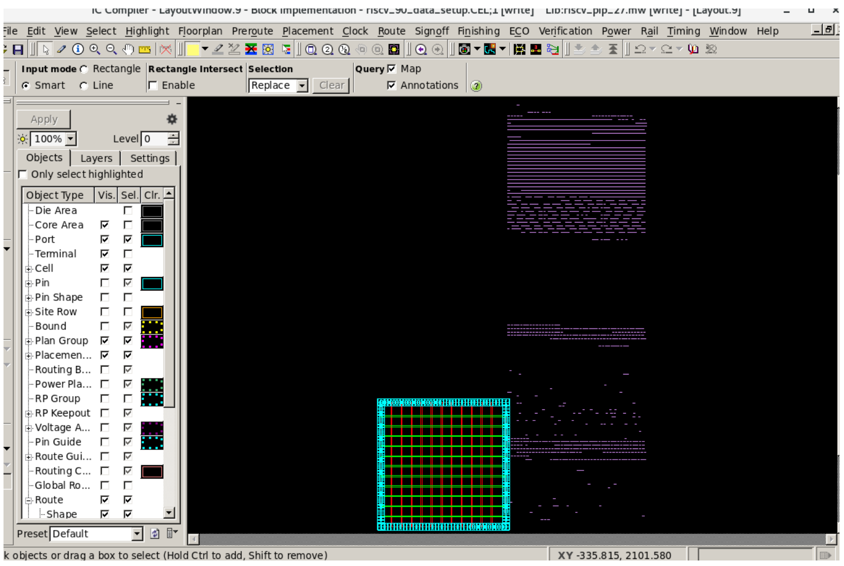
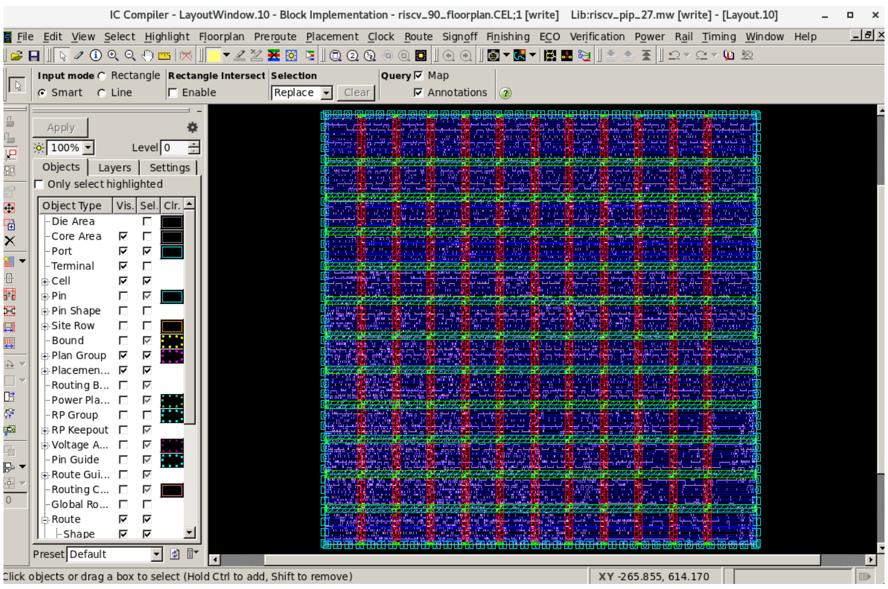


Figure 4.5: Initial floorplan

```
##############################################################
# Floorplan
##############################################################

source $derive_pg_file
create_floorplan -core_utilization 0.3 -left_io2core 1 -bottom_io2core 1 -right_io2core 1 -
top_io2core 1
source $derive_pg_file
create_rectangular_rings  -nets  {VDD VSS}  -left_segment_layer METAL4 -right_segment_layer
METAL4 -bottom_segment_layer METAL3 -top_segment_layer METAL3
create_power_straps  -direction horizontal  -start_at 50.000 -nets  {VDD VSS}  -layer
METAL4 -width 5 -configure groups_and_step  -num_groups 11 -step 50
create_power_straps  -direction vertical  -start_at 50.000 -nets  {VDD VSS}  -layer METAL3
-width 5 -configure groups_and_step  -num_groups 11 -step 50
```

Coarse cell placement and routing is performed, followed by legality check to ensure cells are placed without violation. The reason of doing coarse placement is to avoid over utilization, which will be a potential issue for design to fail in placement stage. The QoR reports are printed to ensure floorplan quality.


IC Compiler - LayoutWindow.10 - Block Implementation - riscv_90_floorplan.CEL;1 [write]   Lib:riscv_pip_27.mw [write] - [Layout.10]
File Edit View Select Highlight Floorplan Preroute Placement Clock Route Signoff Finishing ECO Verification Power Rail Timing Window Help
Input mode Rectangle Smart Line
Rectangle Intersect Enable
Selection Replace Clear
Query Map Annotations
Apply
100%
Level 0
Objects Layers Settings
Only select highlighted
Object Type Vis. Sel. Clr.
Die Area
Core Area
Port
Terminal
Cell
Pin
Pin Shape
Site Row
Bound
Plan Group
Placemen...
Routing B...
Power Pla...
RP Group
RP Keepout
Voltage A...
Pin Guide
Route Gui...
Routing C...
Global Ro...
Route
Shape
Preset Default

Click objects or drag a box to select (Hold Ctrl to add, Shift to remove)   XY -265.855, 614.170

Figure 4.6: Layout after coarse placement and routing

The derive_pg_file is sourced in to make sure no connection is dropped during floorplanning and coarse placement and routing. Afterwards, design, QoR and timing reports are checked, and no violations are seen. The design is saved as input for placement stage.

```
source $derive_pg_file

create_fp_placement -timing_driven -no_hierarchy_gravity

set_pnet_options -complete {METAL3 METAL4}

source $derive_pg_file

set_pnet_options -complete {METAL3 METAL4}

preroute_standard_cells -remove_floating_pieces

route_zrt_global

report_qor > ./reports/floorplan_report_qor.rpt
report_timing > ./reports/floorplan_report_timing.rpt
report_area > ./reports/floorplan_report_area.rpt
check_mv_design > ./reports/floorplan_check_mv_design.rpt

save_mw_cel -as riscv_90_floorplan
```

4.1.3 Placement

In placement stage, detailed placement will be carried out. It is important to ensure cells are placed in optimized location as it is baseline for tool during routing stage. Pre-placement area is reported to compare with area after placement.

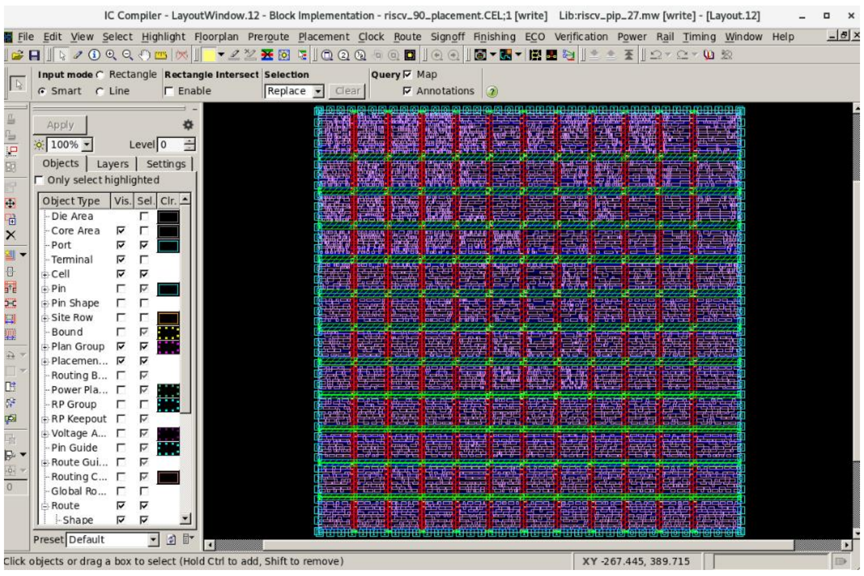
```
Combinational area:              10340.000000
Buf/Inv area:                     1389.250000
Noncombinational area:            8788.750000
Macro/Black Box area:                0.000000
Net Interconnect area:            4319.481348

Total cell area:                 19128.750000
Total area:                      23448.231348
```

Figure 4.7: Pre-placement area

Then, placement is done with place_opt -area_recovery -congestion -power. The place_opt command performs timing, congestion, area and power-driven placement and logic optimization onto the design. It can be observed that post-placement area is ~460 microns smaller than pre-placement design.

```
Combinational area:               9537.500000
Buf/Inv area:                      649.500000
Noncombinational area:            8771.000000
Macro/Black Box area:                0.000000
Net Interconnect area:            5599.686843

Total cell area:                 18308.500000
Total area:                      23908.186843
```

Figure 4.8: Post-placement area


IC Compiler - LayoutWindow.12 - Block Implementation - riscv_90_placement.CEL;1 [write]   Lib:riscv_pip_27.mw [write] - [Layout.12]

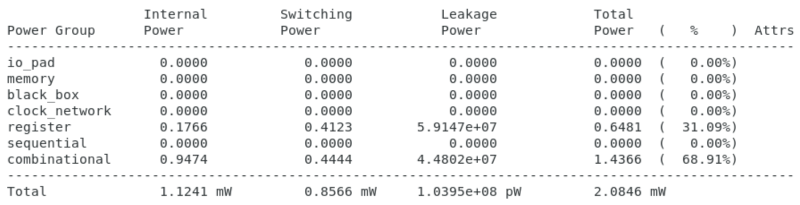
Click objects or drag a box to select (Hold Ctrl to add, Shift to remove)   XY -267.445, 389.715

Figure 4.9: Layout post-place_opt

```
##############################################################
# Placement
##############################################################

set_power_options -low_power_placement true
set_separate_process_options -placement false

place_opt -area_recovery -congestion -power
```

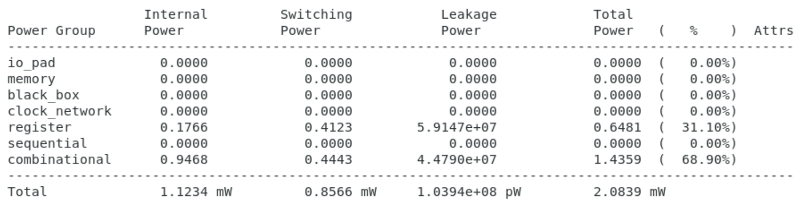
```
report_power > ./reports/init_placement_report_power.rpt
report_qor > ./reports/init_placement_report_qor.rpt
report_area > ./reports/init_placement_report_area.rpt
report_timing > ./reports/init_placement_report_timing.rpt

save_mw_cel -as riscv_90_init_placement
```

Power optimization is performed with psynopt -area_recovery -power. From Figure 4.10 and Figure 4.11, internal power has improved by 0.7uW.



```
                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad             0.0000           0.0000             0.0000             0.0000  (   0.00%)
memory             0.0000           0.0000             0.0000             0.0000  (   0.00%)
black_box          0.0000           0.0000             0.0000             0.0000  (   0.00%)
clock_network      0.0000           0.0000             0.0000             0.0000  (   0.00%)
register           0.1766           0.4123         5.9147e+07             0.6481  (  31.09%)
sequential         0.0000           0.0000             0.0000             0.0000  (   0.00%)
combinational      0.9474           0.4444         4.4802e+07             1.4366  (  68.91%)
--------------------------------------------------------------------------------------------------
Total              1.1241 mW        0.8566 mW      1.0395e+08 pW          2.0846 mW
```

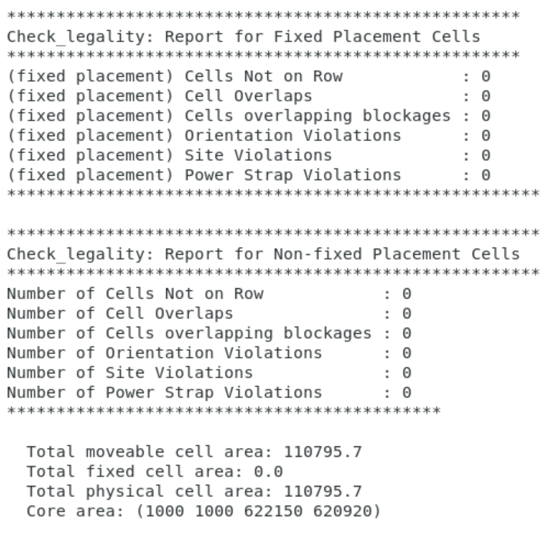
Figure 4.10: pre-psynopt



```
                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad             0.0000           0.0000             0.0000             0.0000  (   0.00%)
memory             0.0000           0.0000             0.0000             0.0000  (   0.00%)
black_box          0.0000           0.0000             0.0000             0.0000  (   0.00%)
clock_network      0.0000           0.0000             0.0000             0.0000  (   0.00%)
register           0.1766           0.4123         5.9147e+07             0.6481  (  31.10%)
sequential         0.0000           0.0000             0.0000             0.0000  (   0.00%)
combinational      0.9468           0.4443         4.4790e+07             1.4359  (  68.90%)
--------------------------------------------------------------------------------------------------
Total              1.1234 mW        0.8566 mW      1.0394e+08 pW          2.0839 mW
```

Figure 4.11: post-psynopt

After placement, legality of the design is reported, alongside with other QoR reports. The design is saved as input for clock tree synthesis stage.



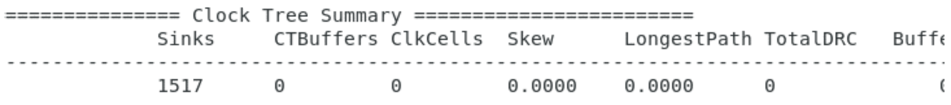
```
****************************************************
Check_legality: Report for Fixed Placement Cells
****************************************************
(fixed placement) Cells Not on Row           : 0
(fixed placement) Cell Overlaps              : 0
(fixed placement) Cells overlapping blockages : 0
(fixed placement) Orientation Violations     : 0
(fixed placement) Site Violations            : 0
(fixed placement) Power Strap Violations     : 0
******************************************************

******************************************************
Check_legality: Report for Non-fixed Placement Cells
******************************************************
Number of Cells Not on Row           : 0
Number of Cell Overlaps              : 0
Number of Cells overlapping blockages : 0
Number of Orientation Violations     : 0
Number of Site Violations            : 0
Number of Power Strap Violations     : 0
********************************************

  Total moveable cell area: 110795.7
  Total fixed cell area: 0.0
  Total physical cell area: 110795.7
  Core area: (1000 1000 622150 620920)
```

Figure 4.12: legality report post-placement

```
psynopt -area_recovery -power

report_power > ./reports/placement_report_power.rpt
report_qor > ./reports/placement_report_qor.rpt
report_area > ./reports/placement_report_area.rpt
report_timing > ./reports/placement_report_timing.rpt
report_design -physical > ./reports/placement_report_design.rpt
check_legality -verbose > ./reports/placement_report_legality.rpt

save_mw_cel -as riscv_90_placement
```

4.1.4 Clock Tree Synthesis (CTS)

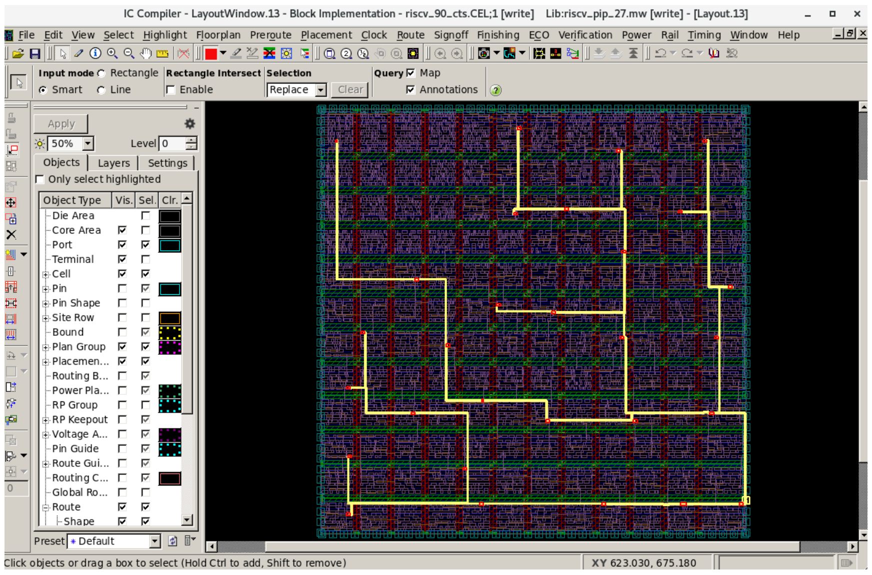
Before performing CTS, clock tree summary is reported to check the number of sinks from the clock port. From the report, it can be seen that clock has a total of 1517 sink points.



```
=============== Clock Tree Summary ========================
             Sinks      CTBuffers ClkCells  Skew      LongestPath TotalDRC   Buffe
--------------------------------------------------------------------------------
             1517       0         0         0.0000    0.0000      0          (
```

Figure 4.13: clock tree summary pre-CTS

First round of CTS is performed to provide initial data. Next, clock uncertainty is removed to allow calculation of actual clock skew during CTS. Hold-time fixing is also enabled. Global routing is performed on all clock nets. The “clock trees” are formed after CTS, shown in Figure 4.14. A total of 29 clock global drivers are highlighted in red. The 1517 sink points are distributed to these 29 global drivers.



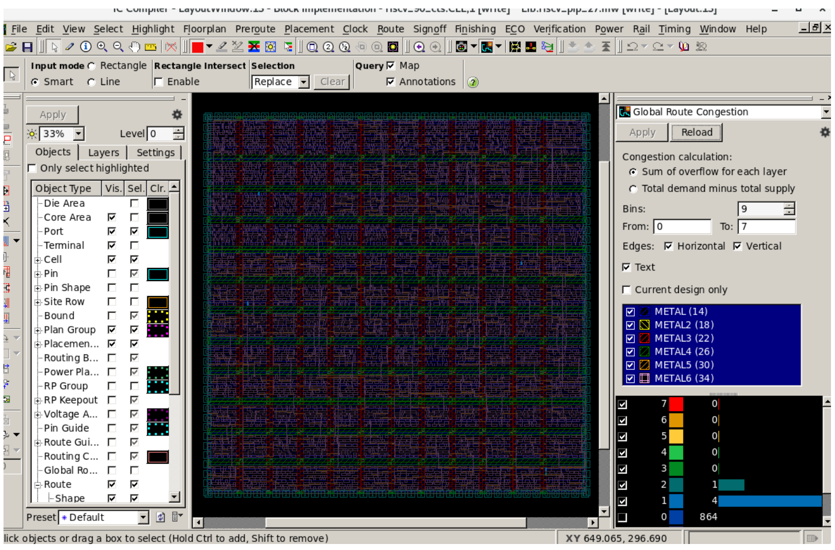
IC Compiler - LayoutWindow.13 - Block Implementation - riscv_90_cts.CEL;1 [write]   Lib:riscv_pip_27.mw [write] - [Layout.13]

Click objects or drag a box to select (Hold Ctrl to add, Shift to remove)   XY 623.030, 675.180

Figure 4.14: clock tree post-CTS

```
############################################################
# CTS
############################################################
report_clocks > ./reports/pre_cts_report_clocks.rpt
report_clock -skew -attributes > ./reports/pre_cts_report_clock_skew.rpt
report_clock_tree -summary > ./reports/pre_cts_report_clock_tree.rpt

set_clock_tree_options -target_skew 0.1
set_clock_uncertainty 0.1 [all_clocks]
set_delay_calculation -routed_clock arnoldi
clock_opt -only_cts -no_clock_route
save_mw_cel -as riscv_90_init_cts

remove_clock_uncertainty [all_clocks]
set_fix_hold [all_clocks]

set physopt_area_critical_range 0.2
extract_rc
route_zrt_group -all_clock_nets -reuse_existing_global_route true

source $derive_pg_file

report_qor > ./reports/cts_report_qor.rpt
report_area > ./reports/cts_report_area.rpt
report_timing > ./reports/cts_report_timing.rpt
report_power > ./reports/cts_report_power.rpt

report_clocks > ./reports/cts_report_clocks.rpt
report_clock -skew -attributes > ./reports/cts_report_clock_skew.rpt
report_clock_tree -summary > ./reports/cts_report_clock_tree.rpt
report_clock_timing -type skew -significant_digits 3 >
./reports/cts_report_clock_timing.rpt

save_mw_cel -as riscv_90_cts
```

Then, Global Route Congestion (GRC) is calculated with report_congestion -grc_based -routing_stage global. From the results generated, there are no routing congestion concern on the design. Cell density is also checked using gui. There is no congested area in the design. The design is saved as input for routing stage.


Global Route Congestion

Congestion calculation:
[✓ Sum of overflow for each layer]
Total demand minus total supply

Bins: 9
From: 0 To: 7
Edges: ✓ Horizontal ✓ Vertical
✓ Text
Current design only

✓ METAL (14)
✓ METAL2 (18)
✓ METAL3 (22)
✓ METAL4 (26)
✓ METAL5 (30)
✓ METAL6 (34)

✓	7	🟥	0
✓	6	🟧	0
✓	5	🟨	0
✓	4	🟩	0
✓	3	🟩	0
✓	2	🟦	1
✓	1	🟦	4
☐	0	🟦	864

lick objects or drag a box to select (Hold Ctrl to add, Shift to remove)   XY 649.065, 296.690

Figure 4.15: post-CTS GRC

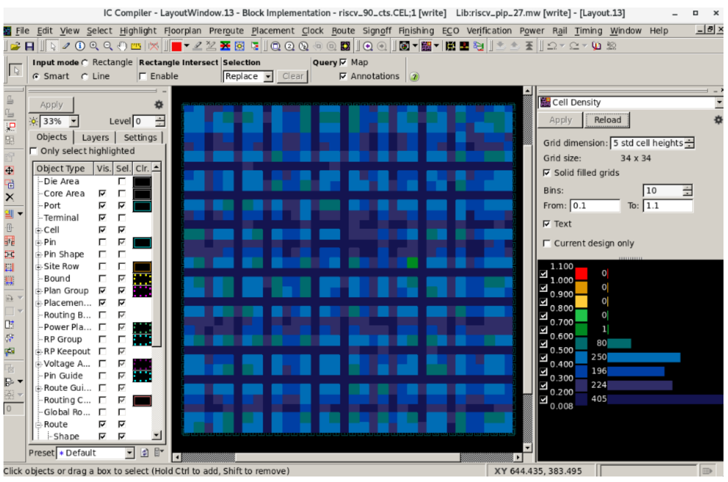


Figure 4.16: post-CTS Cell density

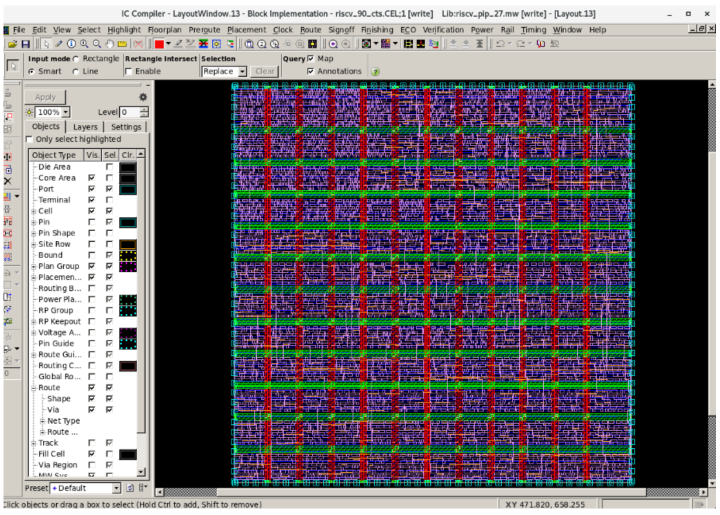


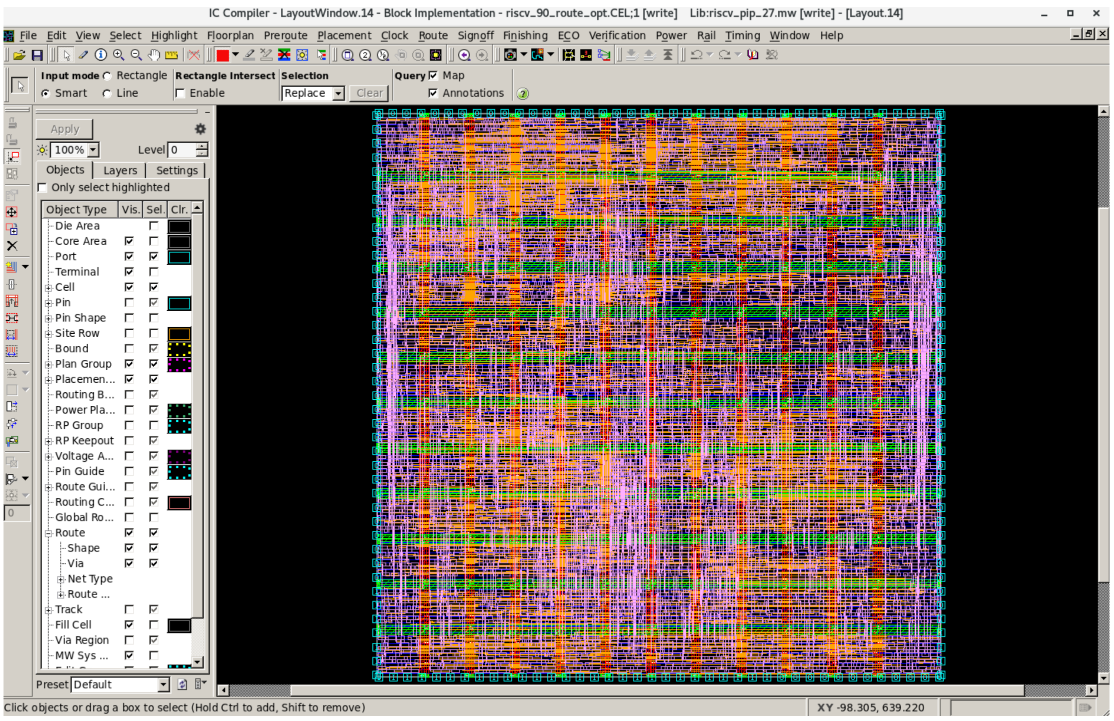
Figure 4.17: Layout post-CTS

4.1.5 Routing

Routing is the final stage for physical implementation where clock and signal are routed and optimized. Before routing, it is important to check for idedal nets and high fanout nets as it will affect routing quality. Nothing is returned, indicating that the design does not have any ideal or high fanout nets. Routing is performed with route_opt -effort high -area_recovery -size_only, where post-routed layout is showed in Figure 4.19.

```
icc_shell> all_ideal_nets
icc_shell> all_high_fanout -nets -threshold 100
icc_shell>
```

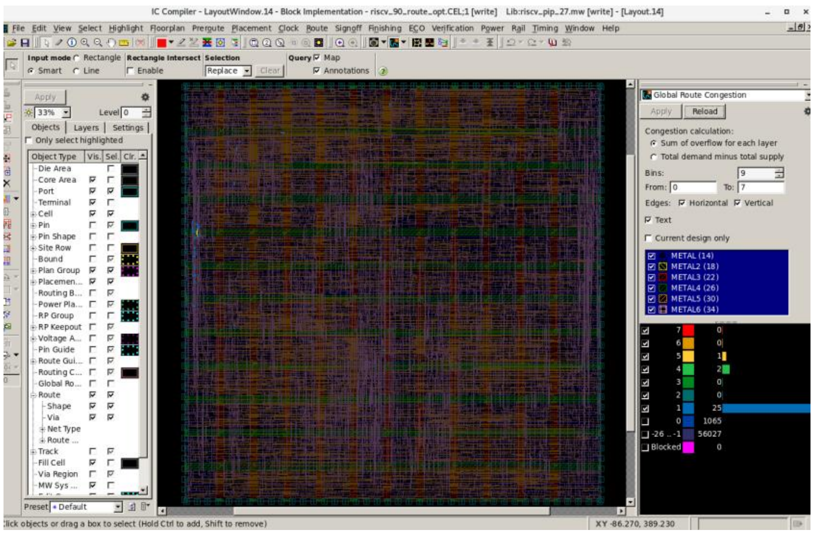
Figure 4.18: Checking ideal or high fanout nets pre-routing


IC Compiler - LayoutWindow.14 - Block Implementation - riscv_90_route_opt.CEL;1 [write]   Lib:riscv_pip_27.mw [write] - [Layout.14]

Click objects or drag a box to select (Hold Ctrl to add, Shift to remove)   XY -98.305, 639.220

Figure 4.19: Port-routing layout design

Routing is performed on the design using route_opt command. GRC, cell density and pin density check is done on the design. GRC and cell density looks fine, but a few hotspots is seen in pin density map. LVS check is then performed to ensure there is no shorts or opens due to high pin density in the region flagged red by pin density map. LVS returned clean results, indicating pin density didn’t cause issue to design. Power grid connection is verified to ensure no PG drop in design.


IC Compiler - LayoutWindow.14 - Block Implementation - riscv_90_route_opt.CEL;1 [write]   Lib:riscv_pip_27.mw [write] - [Layout.14]

Global Route Congestion

Congestion calculation:
(● Sum of overflow for each layer) / Total demand minus total supply

Bins: 9
From: 0   To: 7
Edges: ☑ Horizontal ☑ Vertical
☑ Text
☐ Current design only

☑ METAL (14)
☑ METAL2 (18)
☑ METAL3 (22)
☑ METAL4 (26)
☑ METAL5 (30)
☑ METAL6 (34)

☑	7	🟥	0
☑	6	🟧	0
☑	5	🟨	1
☑	4	🟩	2
☑	3	🟩	0
☑	2	🟦	0
☑	1	🟦	25
☐	0	🟦	1065
☐	-26 .. -1	🟪	56027
☐	Blocked	🟪	0

Click objects or drag a box to select (Hold Ctrl to add, Shift to remove)   XY -86.270, 389.230

File Edit View Select Highlight Floorplan Preroute Placement Clock Route Signoff Finishing ECO Verification Power Rail Timing Window Help
Input mode Rectangle Smart Line
Rectangle Intersect Enable
Selection Replace Clear
Query Map Annotations
Apply Reload
33%
Level 0
Objects Layers Settings
Only select highlighted
Object Type Vis. Sel. Clr.
Preset Default

Figure 4.20: post-routing GRC

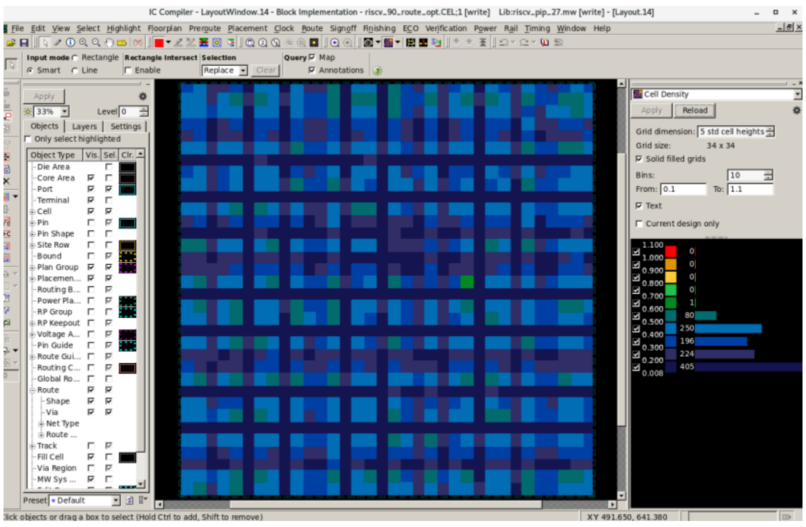


Figure 4.21: post-routing cell density

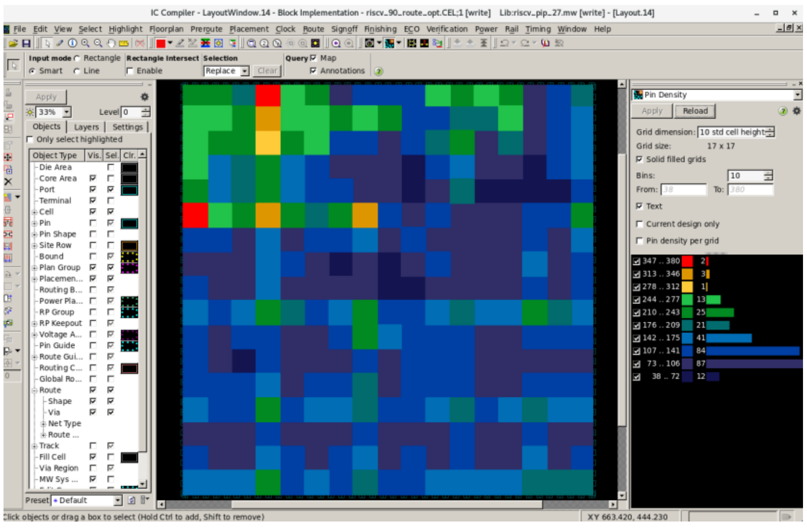
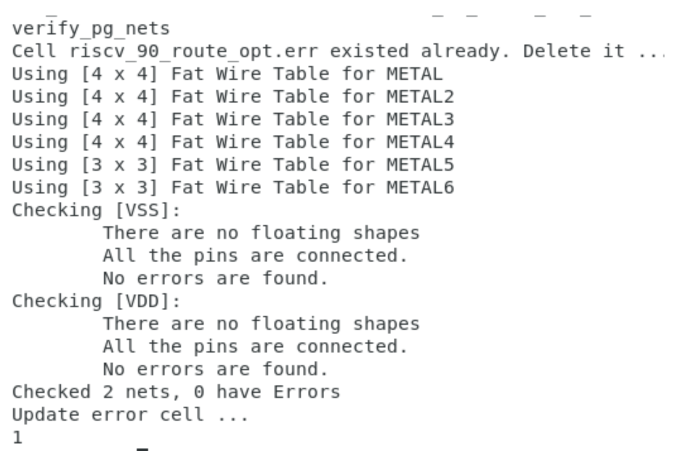


Figure 4.22: post-routing pin density

```
** Total SHORT Nets are 0.
** Total OPEN Nets are 0.
** Total Must Joint Error are 0.

-- LVS END : --
Elapsed =     0:00:01, CPU =     0:00:00
Update error cell ...
1
```

Figure 4.23: post-routing LVS



```
verify_pg_nets
Cell riscv_90_route_opt.err existed already. Delete it ...
Using [4 x 4] Fat Wire Table for METAL
Using [4 x 4] Fat Wire Table for METAL2
Using [4 x 4] Fat Wire Table for METAL3
Using [4 x 4] Fat Wire Table for METAL4
Using [3 x 3] Fat Wire Table for METAL5
Using [3 x 3] Fat Wire Table for METAL6
Checking [VSS]:
        There are no floating shapes
        All the pins are connected.
        No errors are found.
Checking [VDD]:
        There are no floating shapes
        All the pins are connected.
        No errors are found.
Checked 2 nets, 0 have Errors
Update error cell ...
1
```

Figure 4.24: post-routing verify_pg_nets

The derive_pg_file is sourced in to make sure no connection is dropped during routing. The design is now optimized in terms of placement and routing. This is where the auto-place and route process end.

```
#####################
# Route Opt
#####################

all_ideal_nets
all_high_fanout -nets -threshold 100
check_legality
route_opt -effort high -area_recovery -size_only

source $derive_pg_file

verify_lvs
verify_pg_nets

report_qor > ./reports/route_opt_report_qor.rpt
report_area > ./reports/route_opt_report_area.rpt
report_timing > ./reports/route_opt_report_timing.rpt
report_power > ./reports/route_opt_report_power.rpt
verify_lvs > ./reports/route_opt_LVS.rpt
verify_pg_nets > ./reports/route_opt_verify_pg_net.rpt
report_utilization > ./reports/route_opt_report_utilization.rpt

save_mw_cel -as riscv_90_route_opt
```

After checking all the reports generated and there are no concerned violations, the design is saved. The design was ready to proceed to static timing analysis phase. The generated layout design was written as output in ddc and verilog format. The parasitics information (post-layout resistance and capacitance) were extracted into a SPEF file to be used in static timing analysis.

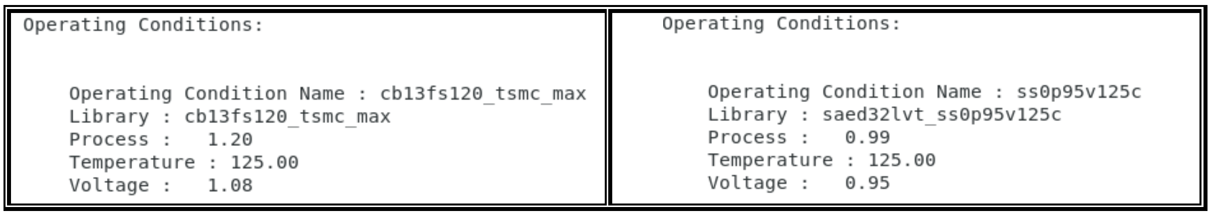
```
write -f ddc -hier -out ../design_data/riscv_90_5ns_postlayout.ddc
write_verilog ../design_data/riscv_90_5ns_postlayout.v
extract_rc
write_parasitics -output ../design_data/riscv_90_5ns_postlayout.spef.gz
```

4.2 Quality Of Results (QoR) Comparison Between 90nm and 32nm Technology Node

In this section, QoR for 90nm design and 32nm is analyzed. Both designs have same constraints being set by front-end design engineer, shown below.

```
reset_design

create_clock -period 5 [get_ports clk]
set_clock_uncertainty -setup 0.1 [get_clocks clk]
set_clock_transition -max 0.05 [get_clocks clk]

# Using default "Operating Conditions"
# from the "slow corner" library: cb13fs120_tsmc_max

set_input_delay -clock clk  -max 0.2 [get_ports reset]
remove_input_delay [get_ports clk]

remove_driving_cell [get_ports clk]

set_input_delay -clock clk  -max 0.2 [get_ports reset]
set_input_delay -clock clk  -max 0.1 [get_ports {InstrF[*] ReadDataM[*]}]
remove_input_delay [get_ports clk]

remove_driving_cell [get_ports clk]

set_output_delay -clock clk  -max 0.2 [all_outputs]
set_load -max 0.5 [all_outputs]
set_max_area 500
set_max_fanout 5 [get_ports -filter direction=~in]
set_max_transition 10 [get_ports  -filter direction=~in]
```

The operating condition of both technology library is compared as shown in Figure 4.25. Under same temperature, 32nm technology node is able to operate at lower voltage, resulting in lower power consumption and improved energy efficiency.



```
Operating Conditions:


    Operating Condition Name : cb13fs120_tsmc_max
    Library : cb13fs120_tsmc_max
    Process :   1.20
    Temperature : 125.00
    Voltage :   1.08
```

```
Operating Conditions:


    Operating Condition Name : ss0p95v125c
    Library : saed32lvt_ss0p95v125c
    Process :   0.99
    Temperature : 125.00
    Voltage :   0.95
```

Figure 4.25: operating condition comparison between 90nm (LHS) and 32nm (RHS)

4.2.1 QoR comparison in Data Setup stage

As soon as design is loaded, an obvious comparison is seen in cell size. As expected, 32nm technology library has smaller cell size compared to 90nm technology library. It is observed that 90nm has higher cell count compared to 32nm as well, with 90nm design having 7029 cell count, and 32nm design having 6175 cell count, 12.15% reduction in cell count. This is because 32nm technology library contains more complex and advanced cell types with higher processing and calculation capability.

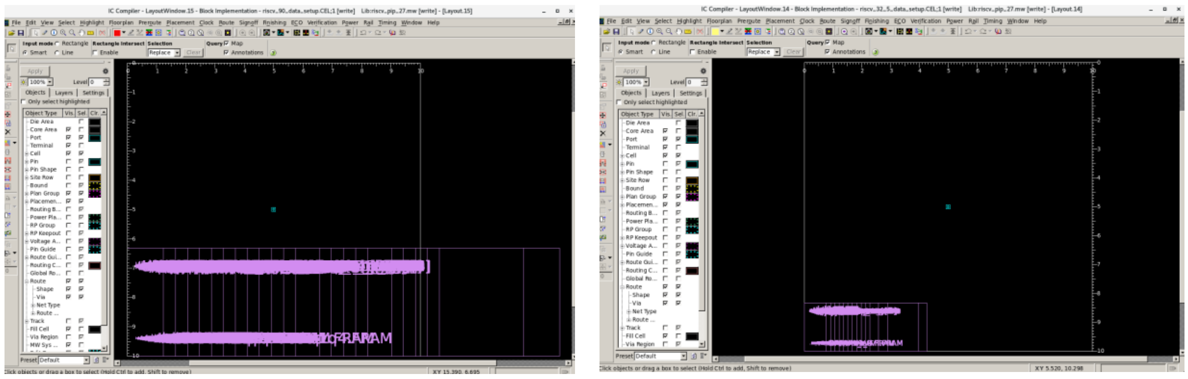
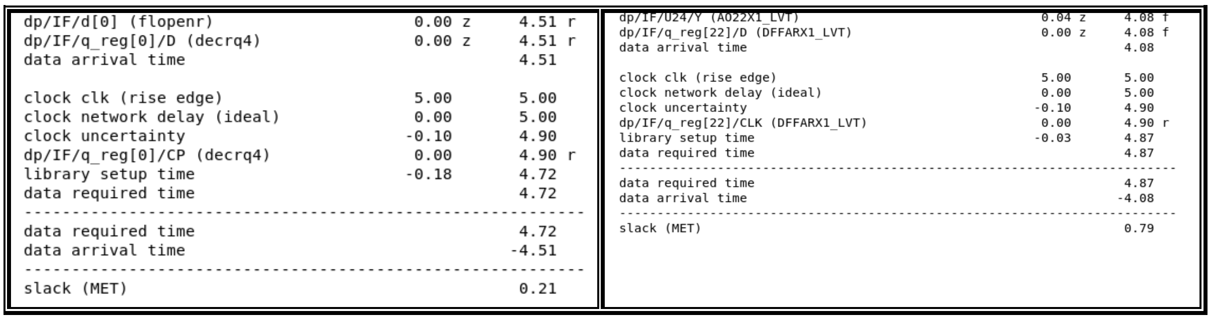


Figure 4.26: cell size comparison between 90nm (LHS) and 32nm (RHS)

In terms of pre-PnR timing, 32nm shows better timing slack. As mentioned in previous chapters, newer (or smaller) technology nodes have faster speed despite smaller cell size. This can be seen in data arrival time, where 90um takes 4.51ns, while 32nm takes 4.08ns. Library setup time is significantly faster in 35nm, with a value of 0.03ns. This is 0.15nm faster than 90nm tech node. “clock clk (rise edge)” indicates the clock period, where both designs are having 5ns clock period.



```
dp/IF/d[0] (flopenr)                        0.00 z     4.51 r
dp/IF/q_reg[0]/D (decrq4)                   0.00 z     4.51 r
data arrival time                                      4.51

clock clk (rise edge)                       5.00       5.00
clock network delay (ideal)                 0.00       5.00
clock uncertainty                          -0.10       4.90
dp/IF/q_reg[0]/CP (decrq4)                  0.00       4.90 r
library setup time                         -0.18       4.72
data required time                                     4.72
--------------------------------------------------------------
data required time                                     4.72
data arrival time                                     -4.51
--------------------------------------------------------------
slack (MET)                                            0.21
```

```
dp/IF/U24/Y (AO22X1_LVT)                              0.04 z     4.08 f
dp/IF/q_reg[22]/D (DFFARX1_LVT)                       0.00 z     4.08 f
data arrival time                                                4.08

clock clk (rise edge)                                 5.00       5.00
clock network delay (ideal)                           0.00       5.00
clock uncertainty                                    -0.10       4.90
dp/IF/q_reg[22]/CLK (DFFARX1_LVT)                     0.00       4.90 r
library setup time                                   -0.03       4.87
data required time                                               4.87
--------------------------------------------------------------------------
data required time                                               4.87
data arrival time                                               -4.08
--------------------------------------------------------------------------
slack (MET)                                                      0.79
```

Figure 4.27: pre-PnR timing comparison between 90nm (LHS) and 32nm (RHS)

Table 4.1: Comparison between 90nm and 32nm in data setup stage

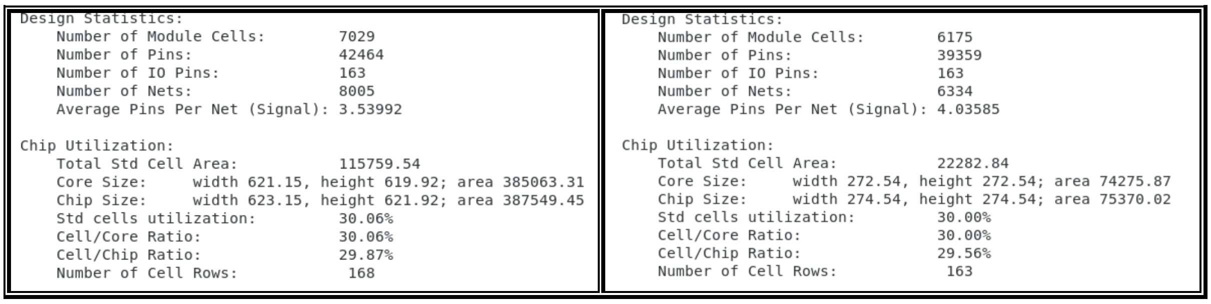
Parameter / Tech Node	90nm	32nm
Cell count	7029	6175
Power pin count	14058	12350
Critical Path Slack (ns)	0.21 (MET)	0.79 (MET)

4.2.2 QoR comparison in Floorplan stage

Both 90nm and 32nm design is constrained with core utilization of 30%, and I/O-to-core distance of 1microns. Same metal layers are also used to create power rings.

```
create_floorplan -core_utilization 0.3 -left_io2core 1 -bottom_io2core 1 -right_io2core 1 -
top_io2core 1

#90nm
create_rectangular_rings  -nets  {VDD VSS}  -left_segment_layer METAL4 -right_segment_layer
METAL4 -bottom_segment_layer METAL3 -top_segment_layer METAL3

#32nm
create_rectangular_rings  -nets  {VDD VSS}  -left_segment_layer M4 -right_segment_layer M4
-bottom_segment_layer M3 -top_segment_layer M3
```

After floorplan is generated, the die size is compared with report_design -physical in both designs. 90nm design is 5 times bigger than 32nm design in terms of core size.

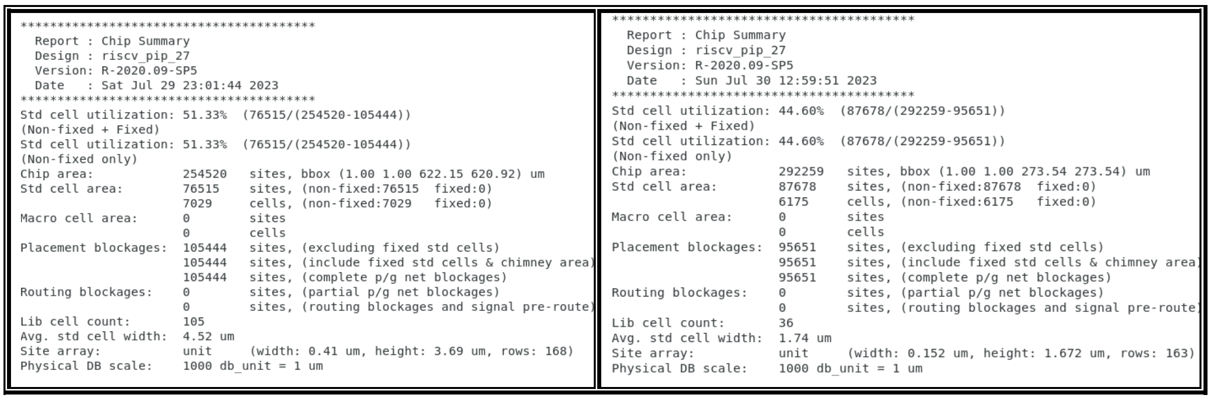


```
Design Statistics:
    Number of Module Cells:        7029
    Number of Pins:                42464
    Number of IO Pins:             163
    Number of Nets:                8005
    Average Pins Per Net (Signal): 3.53992

Chip Utilization:
    Total Std Cell Area:           115759.54
    Core Size:     width 621.15, height 619.92; area 385063.31
    Chip Size:     width 623.15, height 621.92; area 387549.45
    Std cells utilization:         30.06%
    Cell/Core Ratio:               30.06%
    Cell/Chip Ratio:               29.87%
    Number of Cell Rows:            168
```

```
Design Statistics:
    Number of Module Cells:        6175
    Number of Pins:                39359
    Number of IO Pins:             163
    Number of Nets:                6334
    Average Pins Per Net (Signal): 4.03585

Chip Utilization:
    Total Std Cell Area:           22282.84
    Core Size:     width 272.54, height 272.54; area 74275.87
    Chip Size:     width 274.54, height 274.54; area 75370.02
    Std cells utilization:         30.00%
    Cell/Core Ratio:               30.00%
    Cell/Chip Ratio:               29.56%
    Number of Cell Rows:            163
```

Figure 4.28: physical design data comparison between 90nm (LHS) and 32nm (RHS)

Figure 4.29 shows design data post coarse place and route during floorplan stage. Utilization % of 90nm is higher by ~7% due to bigger cell size, which can be proven by average standard cell width data – with 90nm design having 4.52um average width and 32nm design having 1.74um average width.

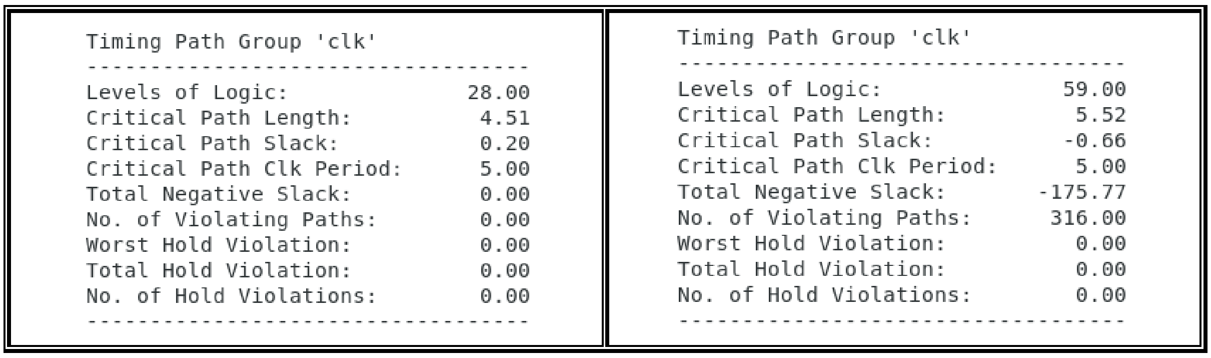


```
****************************************
  Report : Chip Summary
  Design : riscv_pip_27
  Version: R-2020.09-SP5
  Date   : Sat Jul 29 23:01:44 2023
****************************************
Std cell utilization: 51.33%  (76515/(254520-105444))
(Non-fixed + Fixed)
Std cell utilization: 51.33%  (76515/(254520-105444))
(Non-fixed only)
Chip area:            254520   sites, bbox (1.00 1.00 622.15 620.92) um
Std cell area:        76515    sites, (non-fixed:76515  fixed:0)
                      7029     cells, (non-fixed:7029   fixed:0)
Macro cell area:      0        sites
                      0        cells
Placement blockages:  105444   sites, (excluding fixed std cells)
                      105444   sites, (include fixed std cells & chimney area)
                      105444   sites, (complete p/g net blockages)
Routing blockages:    0        sites, (partial p/g net blockages)
                      0        sites, (routing blockages and signal pre-route)
Lib cell count:       105
Avg. std cell width:  4.52 um
Site array:           unit     (width: 0.41 um, height: 3.69 um, rows: 168)
Physical DB scale:    1000 db_unit = 1 um
```

```
****************************************
  Report : Chip Summary
  Design : riscv_pip_27
  Version: R-2020.09-SP5
  Date   : Sun Jul 30 12:59:51 2023
****************************************
Std cell utilization: 44.60%  (87678/(292259-95651))
(Non-fixed + Fixed)
Std cell utilization: 44.60%  (87678/(292259-95651))
(Non-fixed only)
Chip area:            292259   sites, bbox (1.00 1.00 273.54 273.54) um
Std cell area:        87678    sites, (non-fixed:87678  fixed:0)
                      6175     cells, (non-fixed:6175   fixed:0)
Macro cell area:      0        sites
                      0        cells
Placement blockages:  95651    sites, (excluding fixed std cells)
                      95651    sites, (include fixed std cells & chimney area)
                      95651    sites, (complete p/g net blockages)
Routing blockages:    0        sites, (partial p/g net blockages)
                      0        sites, (routing blockages and signal pre-route)
Lib cell count:       36
Avg. std cell width:  1.74 um
Site array:           unit     (width: 0.152 um, height: 1.672 um, rows: 163)
Physical DB scale:    1000 db_unit = 1 um
```

Figure 4.29: chip summary comparison between 90nm (LHS) and 32nm (RHS)

In terms of timing, 90nm design has all timing met at this stage, while on the other hand 32nm design has WNS of -0.66ns. This is due to 2 times more levels of logic in 32nm, causing cell delay to add up. Since this stage is performed with coarse placement and routing, -0.66ns is consider acceptable.



```
Timing Path Group 'clk'
-----------------------------------
Levels of Logic:              28.00
Critical Path Length:          4.51
Critical Path Slack:           0.20
Critical Path Clk Period:      5.00
Total Negative Slack:          0.00
No. of Violating Paths:        0.00
Worst Hold Violation:          0.00
Total Hold Violation:          0.00
No. of Hold Violations:        0.00
-----------------------------------
```

```
Timing Path Group 'clk'
-----------------------------------
Levels of Logic:              59.00
Critical Path Length:          5.52
Critical Path Slack:          -0.66
Critical Path Clk Period:      5.00
Total Negative Slack:       -175.77
No. of Violating Paths:      316.00
Worst Hold Violation:          0.00
Total Hold Violation:          0.00
No. of Hold Violations:        0.00
-----------------------------------
```

Figure 4.30: timing QoR comparison between 90nm (LHS) and 32nm (RHS)

Table 4.2: Comparison between 90nm and 32nm in floorplan stage

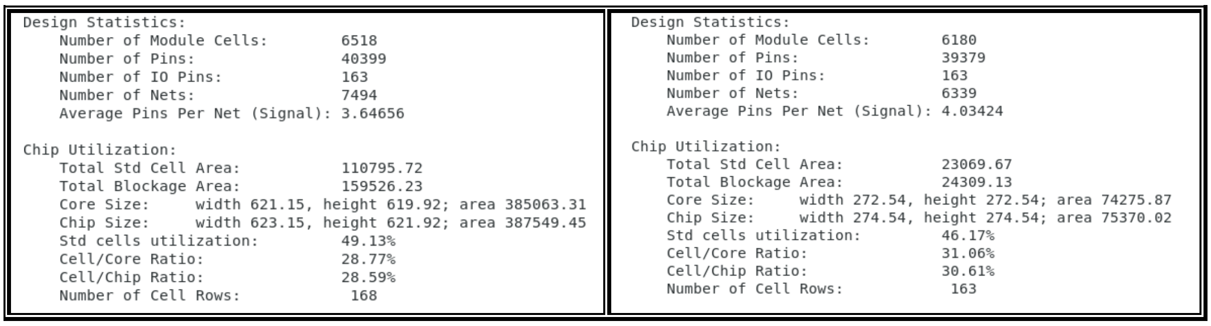
Parameter / Tech Node	90nm	32nm
Die dimension (um) (width x height)	623.15 x 621.92	274.54 x 274.54
Die area (um^2)	387549.45	75370.02
Total standard cell area (um^2)	115759.54	22282.84
Standard cell utilization (pre-coarse PnR) (%)	30	30
Standard cell utilization (post-coarse PnR) (%)	51.33	44.60
Average std cell width (um)	4.52	1.74
Critical Path Slack (ns)	0.20 (MET)	-0.66 (VIOLATED)
Critical Path LoL	28	59

4.2.3 QoR comparison in Placement stage

Both 90nm and 32nm design have placement optimization performed with commands shown below. Post -placement, both designs have 0 errors on legality, which means all cells are placed without overlapping.

```
set_power_options -low_power_placement true
set_separate_process_options -placement false
place_opt -area_recovery -congestion -power
#report_power
psynopt -area_recovery -power
# report_power
```

Post-placement, it can be observed that total standard cell area for 32nm is 79% smaller than 90nm, which is expected as 32nm technology library contains smaller cell size. The shrinkage of cell size enables smaller chip to be fabricated, in this case we can see cell-to-core ratio of 32nm is slightly higher than that on 90nm. Although the ratio value in 32nm is slightly higher than the 30% core utilization constraint set in floorplan stage, the difference is small only 1.06% away from target.

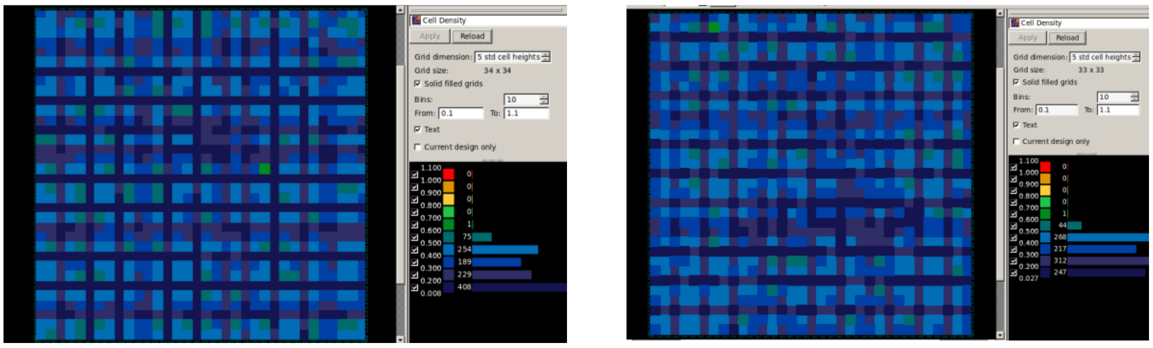


```
Design Statistics:
    Number of Module Cells:        6518
    Number of Pins:                40399
    Number of IO Pins:             163
    Number of Nets:                7494
    Average Pins Per Net (Signal): 3.64656

Chip Utilization:
    Total Std Cell Area:           110795.72
    Total Blockage Area:           159526.23
    Core Size:     width 621.15, height 619.92; area 385063.31
    Chip Size:     width 623.15, height 621.92; area 387549.45
    Std cells utilization:         49.13%
    Cell/Core Ratio:               28.77%
    Cell/Chip Ratio:               28.59%
    Number of Cell Rows:            168
```

```
Design Statistics:
    Number of Module Cells:        6180
    Number of Pins:                39379
    Number of IO Pins:             163
    Number of Nets:                6339
    Average Pins Per Net (Signal): 4.03424

Chip Utilization:
    Total Std Cell Area:           23069.67
    Total Blockage Area:           24309.13
    Core Size:     width 272.54, height 272.54; area 74275.87
    Chip Size:     width 274.54, height 274.54; area 75370.02
    Std cells utilization:         46.17%
    Cell/Core Ratio:               31.06%
    Cell/Chip Ratio:               30.61%
    Number of Cell Rows:            163
```

Figure 4.31: area and utilization QoR comparison between 90nm (LHS) and 32nm (RHS)

Looking at cell density, the placement quality is good, where there is no hotspots seen in the design. Between range of 0.5-0.6, 90nm design has a slightly higher count than 32nm, due to bigger cell size.



Cell Density

Apply Reload

Grid dimension: 5 std cell heights
Grid size: 34 x 34
[✓] Solid filled grids
Bins: 10
From: 0.1 To: 1.1
[✓] Text
[ ] Current design only

1.100–1.000	🟥	0
1.000–0.900	🟧	0
0.900–0.800	🟨	0
0.800–0.700	🟩	0
0.700–0.600	🟩	1
0.600–0.500	🟦	75
0.500–0.400	🟦	254
0.400–0.300	🟦	189
0.300–0.200	🟦	229
0.200–0.008	🟦	408

Cell Density

Apply Reload

Grid dimension: 5 std cell heights
Grid size: 33 x 33
[✓] Solid filled grids
Bins: 10
From: 0.1 To: 1.1
[✓] Text
[ ] Current design only

1.100–1.000	🟥	0
1.000–0.900	🟧	0
0.900–0.800	🟨	0
0.800–0.700	🟩	0
0.700–0.600	🟩	1
0.600–0.500	🟦	44
0.500–0.400	🟦	268
0.400–0.300	🟦	217
0.300–0.200	🟦	312
0.200–0.027	🟦	247

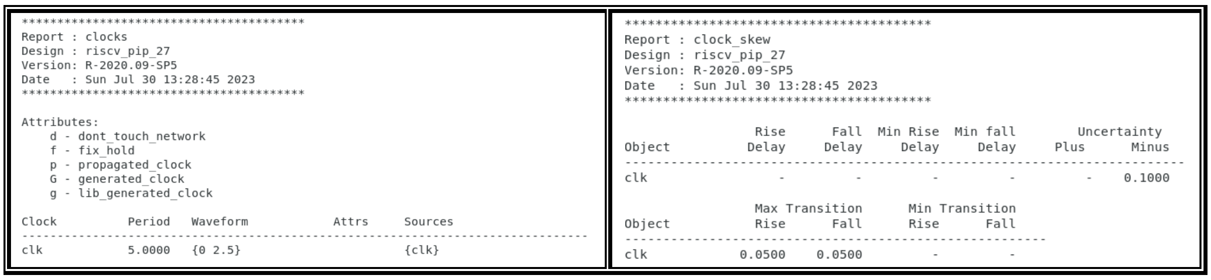
Figure 4.32: Cell density comparison between 90nm (LHS) and 32nm (RHS)

Table 4.3: Comparison between 90nm and 32nm in placement stage

Parameter / Tech Node	90nm	32nm
Number of Module Cells	6518	6180
Standard cells utilization	49.13%	46.17%
Cell-core ratio	28.77%	31.06%

4.2.4 QoR comparison in CTS stage

The clock constraints for both designs are the same, with period of 5ns, uncertainty of -0.1ns and max transition of 0.05ns.

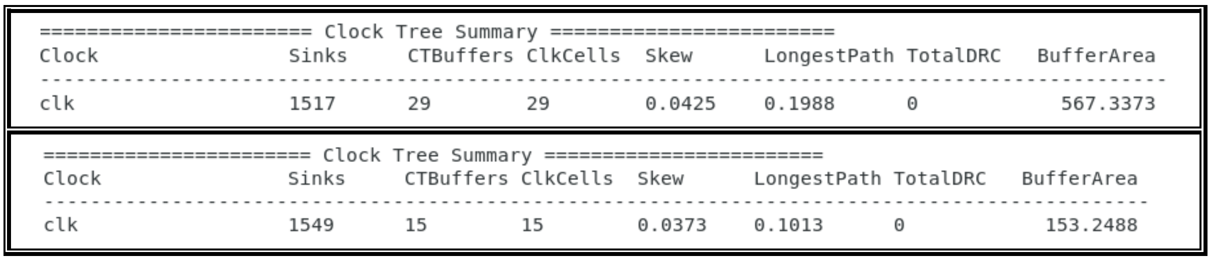


```
****************************************
Report : clocks
Design : riscv_pip_27
Version: R-2020.09-SP5
Date   : Sun Jul 30 13:28:45 2023
****************************************

Attributes:
    d - dont_touch_network
    f - fix_hold
    p - propagated_clock
    G - generated_clock
    g - lib_generated_clock

Clock          Period   Waveform            Attrs     Sources
--------------------------------------------------------------------------------
clk            5.0000   {0 2.5}                       {clk}
```

```
****************************************
Report : clock_skew
Design : riscv_pip_27
Version: R-2020.09-SP5
Date   : Sun Jul 30 13:28:45 2023
****************************************

                    Rise      Fall  Min Rise  Min fall        Uncertainty
Object             Delay     Delay     Delay     Delay     Plus       Minus
--------------------------------------------------------------------------
clk                    -         -         -         -        -      0.1000

                  Max Transition      Min Transition
Object             Rise      Fall      Rise      Fall
------------------------------------------------------
clk              0.0500    0.0500         -         -
```

Figure 4.33: Clock constrains in both 90nm and 32nm design

Looking at synthesized clock tree, 32nm design has lesser global drivers compared to 90nm design, as 32nm technology library is more powerful and able to drive more cells with similar global skew. Clock buffer area is also reduced by 73% in 32nm design, due to lesser clock buffers with smaller size.



```
====================== Clock Tree Summary =======================
Clock                   Sinks     CTBuffers ClkCells  Skew      LongestPath TotalDRC    BufferArea
-------------------------------------------------------------------------------------------------
clk                     1517      29        29        0.0425    0.1988      0             567.3373
```

```
====================== Clock Tree Summary =======================
Clock                   Sinks     CTBuffers ClkCells  Skew      LongestPath TotalDRC    BufferArea
-------------------------------------------------------------------------------------------------
clk                     1549      15        15        0.0373    0.1013      0             153.2488
```

Figure 4.34: Clock tree QoR comparison between 90nm (Top) and 32nm (Bottom)

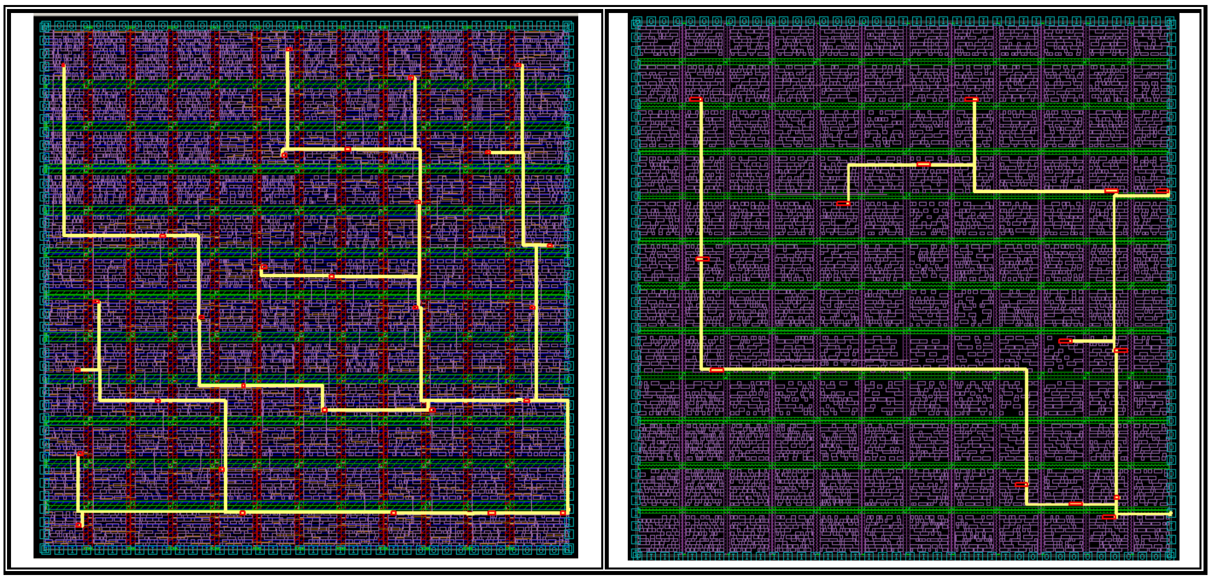
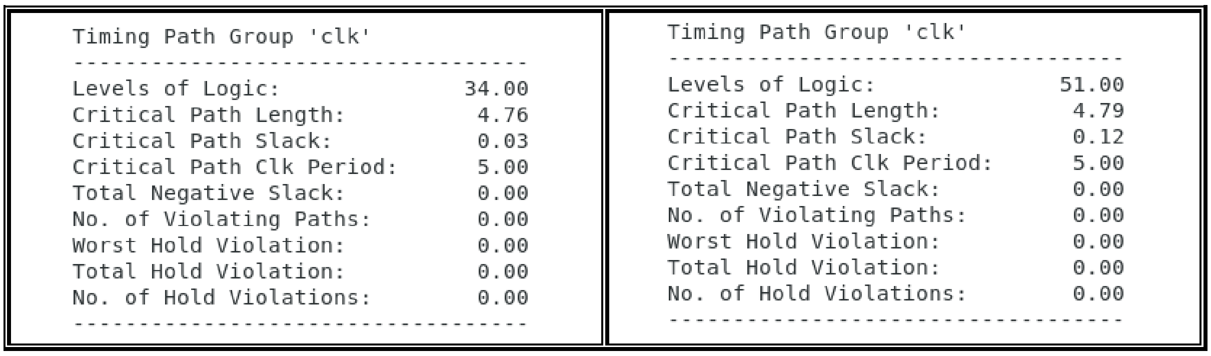


Figure 4.35: Clock tree layout comparison between 90nm (LHS) and 32nm (RHS)

In terms of timing QoR, both designs have setup and hold timing converged post-CTS. From Figure 4.36, it is observed that 32nm design has a bigger timing margin despite having longer LoL, with critical path slack of 0.12ns, compared to 90nm design which has critical path slack of 0.03ns.

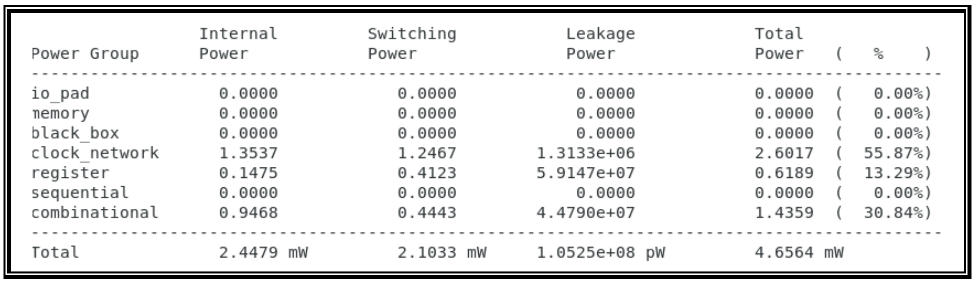


```
Timing Path Group 'clk'
-----------------------------------
Levels of Logic:              34.00
Critical Path Length:          4.76
Critical Path Slack:           0.03
Critical Path Clk Period:      5.00
Total Negative Slack:          0.00
No. of Violating Paths:        0.00
Worst Hold Violation:          0.00
Total Hold Violation:          0.00
No. of Hold Violations:        0.00
-----------------------------------
```

```
Timing Path Group 'clk'
-----------------------------------
Levels of Logic:              51.00
Critical Path Length:          4.79
Critical Path Slack:           0.12
Critical Path Clk Period:      5.00
Total Negative Slack:          0.00
No. of Violating Paths:        0.00
Worst Hold Violation:          0.00
Total Hold Violation:          0.00
No. of Hold Violations:        0.00
-----------------------------------
```

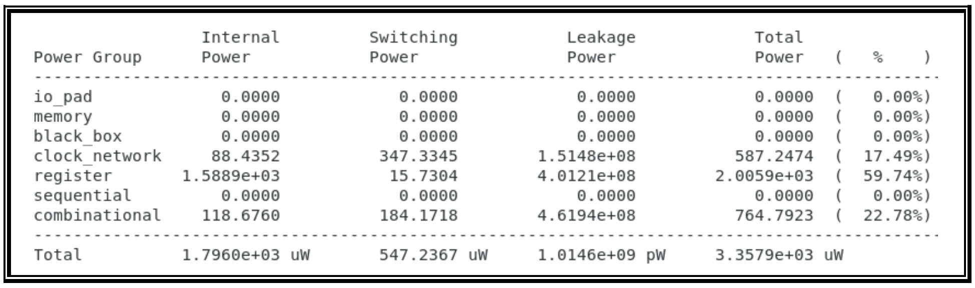
Figure 4.36: Timing QoR comparison between 90nm (LHS) and 32nm (RHS)

On top of that, 32nm design shows better performance in power consumption as well with total power of 3.3579mW, compared to 90nm design with total power of 4.6564mW. Smaller transistors in lower technology nodes have smaller gate capacitance, which means it takes less charge/discharge to switch the transistors on and off. This leads to lower dynamic power consumption during switching operations.



```
                  Internal         Switching          Leakage            Total
Power Group       Power            Power              Power              Power   (   %    )
--------------------------------------------------------------------------------------------
io_pad              0.0000           0.0000             0.0000             0.0000  (   0.00%)
memory              0.0000           0.0000             0.0000             0.0000  (   0.00%)
black_box           0.0000           0.0000             0.0000             0.0000  (   0.00%)
clock_network       1.3537           1.2467         1.3133e+06             2.6017  (  55.87%)
register            0.1475           0.4123         5.9147e+07             0.6189  (  13.29%)
sequential          0.0000           0.0000             0.0000             0.0000  (   0.00%)
combinational       0.9468           0.4443         4.4790e+07             1.4359  (  30.84%)
--------------------------------------------------------------------------------------------
Total               2.4479 mW        2.1033 mW      1.0525e+08 pW          4.6564 mW
```

Figure 4.37(a): Power QoR comparison (90nm)



```
                  Internal         Switching          Leakage            Total
Power Group       Power            Power              Power              Power   (   %    )
--------------------------------------------------------------------------------------------
io_pad              0.0000           0.0000             0.0000             0.0000  (   0.00%)
memory              0.0000           0.0000             0.0000             0.0000  (   0.00%)
black_box           0.0000           0.0000             0.0000             0.0000  (   0.00%)
clock_network      88.4352         347.3345         1.5148e+08           587.2474  (  17.49%)
register       1.5889e+03           15.7304         4.0121e+08         2.0059e+03  (  59.74%)
sequential          0.0000           0.0000             0.0000             0.0000  (   0.00%)
combinational     118.6760         184.1718         4.6194e+08           764.7923  (  22.78%)
--------------------------------------------------------------------------------------------
Total          1.7960e+03 uW       547.2367 uW      1.0146e+09 pW      3.3579e+03 uW
```

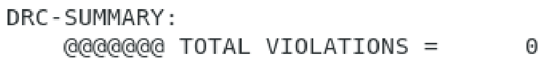
Figure 4.37(b): Power QoR comparison (32nm)

Table 4.4: Comparison between 90nm and 32nm in CTS stage

Parameter / Tech Node	90nm	32nm
Clock buffer count	29	15
Global skew (ns)	0.0425	0.0373
Clock buffer area (um^2)	567.3373	153.2488
Critical path slack (ns)	0.03	0.12
Critical path slack LoL	34	51
Total Power (mW)	4.6564	3.3579
Internal Power (mW)	2.4479	1.796
Switching Power (mW)	2.1033	0.5472
Leakage Power (mW)	0.1053	1.0146

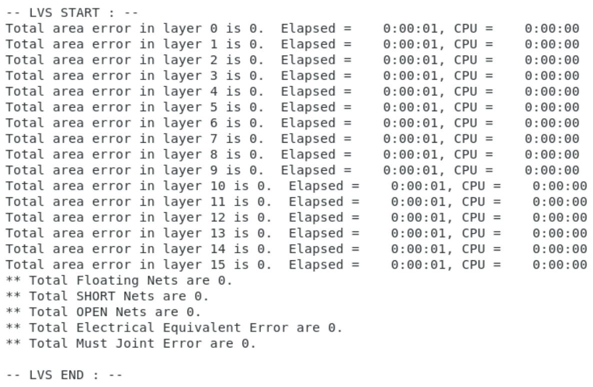
4.2.5 QoR comparison in routing stage

Both 90nm and 32nm design have route optimization performed with command `route_opt -effort high -area_recovery -size_only`. This is the final stage for backend design before inserting filler cells. Both designs are routed with zero DRC and LVS violation.



```
DRC-SUMMARY:
    @@@@@@@ TOTAL VIOLATIONS =      0
```

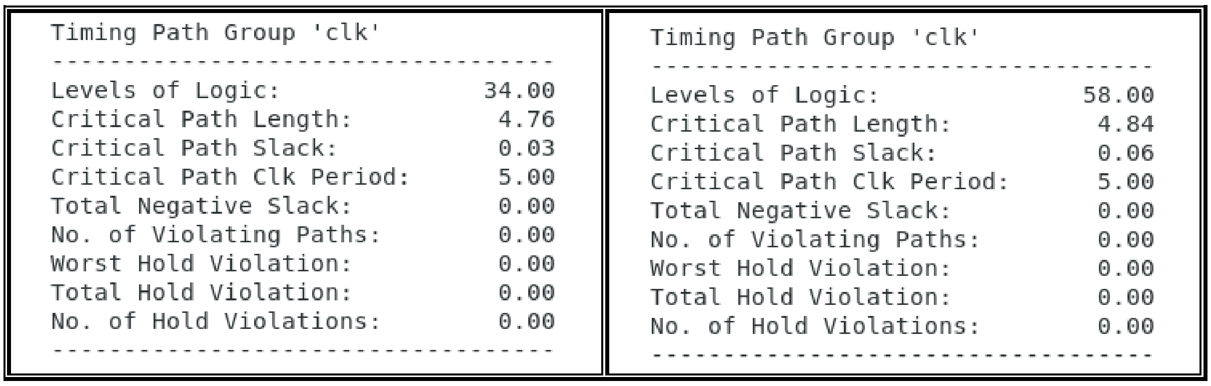
Figure 4.38: DRC clean in both 90nm and 32nm design



```
-- LVS START : --
Total area error in layer 0 is 0.  Elapsed =    0:00:01, CPU =    0:00:00
Total area error in layer 1 is 0.  Elapsed =    0:00:01, CPU =    0:00:00
Total area error in layer 2 is 0.  Elapsed =    0:00:01, CPU =    0:00:00
Total area error in layer 3 is 0.  Elapsed =    0:00:01, CPU =    0:00:00
Total area error in layer 4 is 0.  Elapsed =    0:00:01, CPU =    0:00:00
Total area error in layer 5 is 0.  Elapsed =    0:00:01, CPU =    0:00:00
Total area error in layer 6 is 0.  Elapsed =    0:00:01, CPU =    0:00:00
Total area error in layer 7 is 0.  Elapsed =    0:00:01, CPU =    0:00:00
Total area error in layer 8 is 0.  Elapsed =    0:00:01, CPU =    0:00:00
Total area error in layer 9 is 0.  Elapsed =    0:00:01, CPU =    0:00:00
Total area error in layer 10 is 0.  Elapsed =    0:00:01, CPU =    0:00:00
Total area error in layer 11 is 0.  Elapsed =    0:00:01, CPU =    0:00:00
Total area error in layer 12 is 0.  Elapsed =    0:00:01, CPU =    0:00:00
Total area error in layer 13 is 0.  Elapsed =    0:00:01, CPU =    0:00:00
Total area error in layer 14 is 0.  Elapsed =    0:00:01, CPU =    0:00:00
Total area error in layer 15 is 0.  Elapsed =    0:00:01, CPU =    0:00:00
** Total Floating Nets are 0.
** Total SHORT Nets are 0.
** Total OPEN Nets are 0.
** Total Electrical Equivalent Error are 0.
** Total Must Joint Error are 0.

-- LVS END : --
```

Figure 4.39: LVS clean in both 90nm and 32nm design

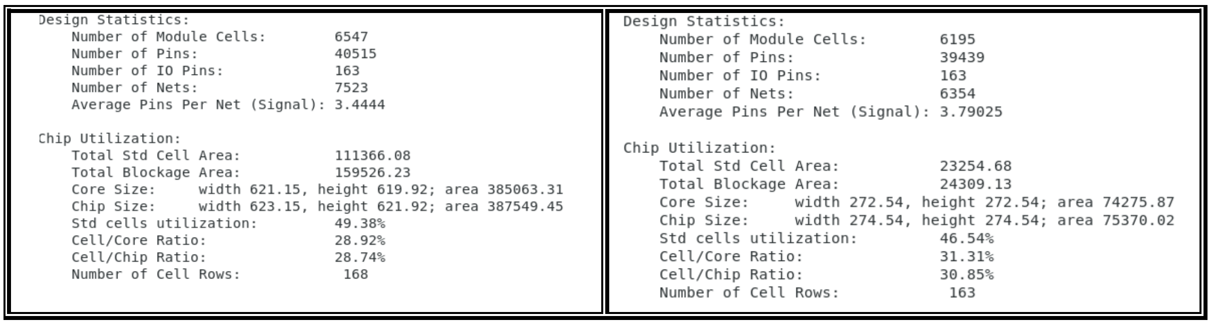
From timing perspective, both designs have setup and hold timing converged, with 32nmm has a bigger timing margin of 0.06ns, despite higher levels of logic. Both designs have utilization rate around 30%, but 32nm design has core size 80.71% smaller than 90nm design. Total standard cell area reduced significantly by 79.12% using 32nm technology library. Another consideration in VLSI design is power. 32nm design has total power of 3.4694mW, while 90nm design is slightly higher at 4.6557mW.



```
Timing Path Group 'clk'
-----------------------------------
Levels of Logic:               34.00
Critical Path Length:           4.76
Critical Path Slack:            0.03
Critical Path Clk Period:       5.00
Total Negative Slack:           0.00
No. of Violating Paths:         0.00
Worst Hold Violation:           0.00
Total Hold Violation:           0.00
No. of Hold Violations:         0.00
-----------------------------------
```

```
Timing Path Group 'clk'
-----------------------------------
Levels of Logic:               58.00
Critical Path Length:           4.84
Critical Path Slack:            0.06
Critical Path Clk Period:       5.00
Total Negative Slack:           0.00
No. of Violating Paths:         0.00
Worst Hold Violation:           0.00
Total Hold Violation:           0.00
No. of Hold Violations:         0.00
-----------------------------------
```

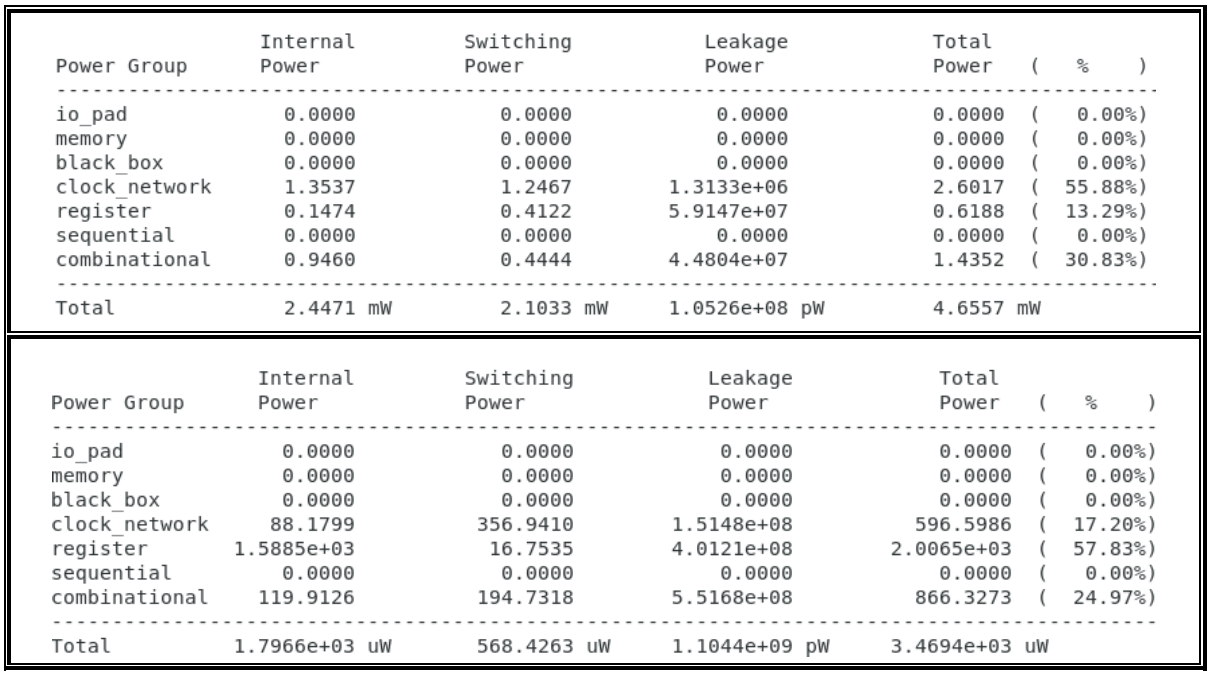
Figure 4.40: Timing QoR comparison between 90nm (LHS) and 32nm (RHS)



```
Design Statistics:
    Number of Module Cells:        6547
    Number of Pins:                40515
    Number of IO Pins:             163
    Number of Nets:                7523
    Average Pins Per Net (Signal): 3.4444

Chip Utilization:
    Total Std Cell Area:           111366.08
    Total Blockage Area:           159526.23
    Core Size:     width 621.15, height 619.92; area 385063.31
    Chip Size:     width 623.15, height 621.92; area 387549.45
    Std cells utilization:         49.38%
    Cell/Core Ratio:               28.92%
    Cell/Chip Ratio:               28.74%
    Number of Cell Rows:            168
```

```
Design Statistics:
    Number of Module Cells:        6195
    Number of Pins:                39439
    Number of IO Pins:             163
    Number of Nets:                6354
    Average Pins Per Net (Signal): 3.79025

Chip Utilization:
    Total Std Cell Area:           23254.68
    Total Blockage Area:           24309.13
    Core Size:     width 272.54, height 272.54; area 74275.87
    Chip Size:     width 274.54, height 274.54; area 75370.02
    Std cells utilization:         46.54%
    Cell/Core Ratio:               31.31%
    Cell/Chip Ratio:               30.85%
    Number of Cell Rows:            163
```

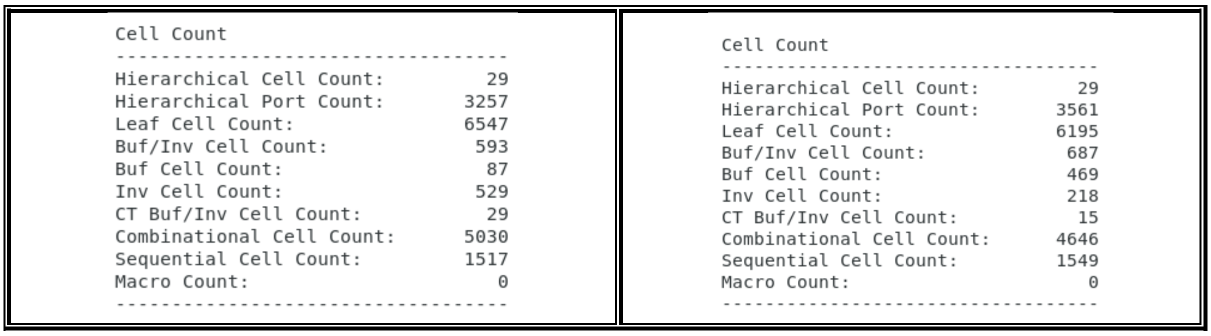
Figure 4.41: Area and physical QoR comparison between 90nm (LHS) and 32nm (RHS)



```
                    Internal         Switching          Leakage          Total
Power Group         Power            Power              Power            Power   (   %    )
----------------------------------------------------------------------------------------------
io_pad                0.0000           0.0000             0.0000           0.0000  (   0.00%)
memory                0.0000           0.0000             0.0000           0.0000  (   0.00%)
black_box             0.0000           0.0000             0.0000           0.0000  (   0.00%)
clock_network         1.3537           1.2467         1.3133e+06           2.6017  (  55.88%)
register              0.1474           0.4122         5.9147e+07           0.6188  (  13.29%)
sequential            0.0000           0.0000             0.0000           0.0000  (   0.00%)
combinational         0.9460           0.4444         4.4804e+07           1.4352  (  30.83%)
----------------------------------------------------------------------------------------------
Total                 2.4471 mW        2.1033 mW      1.0526e+08 pW        4.6557 mW
```

```
                    Internal         Switching          Leakage          Total
Power Group         Power            Power              Power            Power    (   %    )
----------------------------------------------------------------------------------------------
io_pad                0.0000           0.0000             0.0000           0.0000  (   0.00%)
memory                0.0000           0.0000             0.0000           0.0000  (   0.00%)
black_box             0.0000           0.0000             0.0000           0.0000  (   0.00%)
clock_network        88.1799         356.9410         1.5148e+08         596.5986  (  17.20%)
register          1.5885e+03          16.7535         4.0121e+08       2.0065e+03  (  57.83%)
sequential            0.0000           0.0000             0.0000           0.0000  (   0.00%)
combinational       119.9126         194.7318         5.5168e+08         866.3273  (  24.97%)
----------------------------------------------------------------------------------------------
Total             1.7966e+03 uW      568.4263 uW      1.1044e+09 pW    3.4694e+03 uW
```

Figure 4.42: Power QoR comparison between 90nm (Top) and 32nm (Bottom)

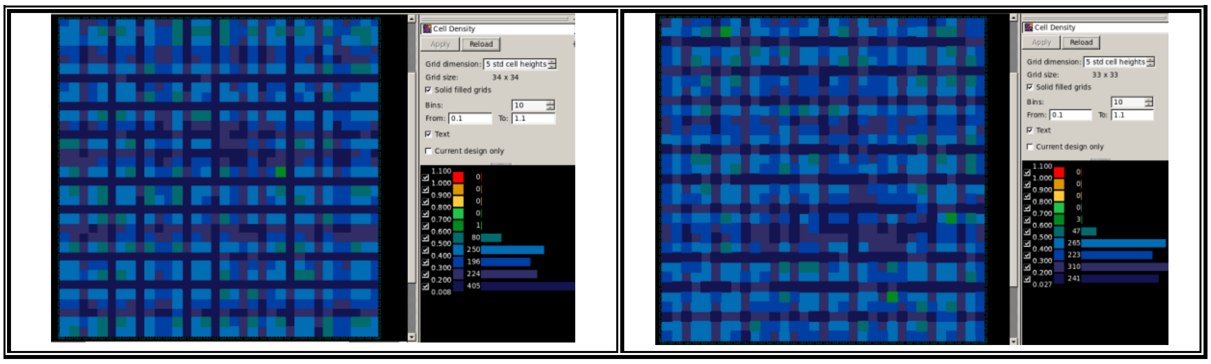


```
Cell Count
-----------------------------------
Hierarchical Cell Count:        29
Hierarchical Port Count:      3257
Leaf Cell Count:              6547
Buf/Inv Cell Count:            593
Buf Cell Count:                 87
Inv Cell Count:                529
CT Buf/Inv Cell Count:          29
Combinational Cell Count:     5030
Sequential Cell Count:        1517
Macro Count:                     0
-----------------------------------
```

```
Cell Count
-----------------------------------
Hierarchical Cell Count:        29
Hierarchical Port Count:      3561
Leaf Cell Count:              6195
Buf/Inv Cell Count:            687
Buf Cell Count:                469
Inv Cell Count:                218
CT Buf/Inv Cell Count:          15
Combinational Cell Count:     4646
Sequential Cell Count:        1549
Macro Count:                     0
-----------------------------------
```

Figure 4.43: Cell count comparison between 90nm (LHS) and 32nm (RHS)

Both designs show satisfactory results in terms of cell density, pin density and global route congestion (GRC). It is observed that 90nm design has more hotspot in pin density, which is due to more combinational cell count in the design.



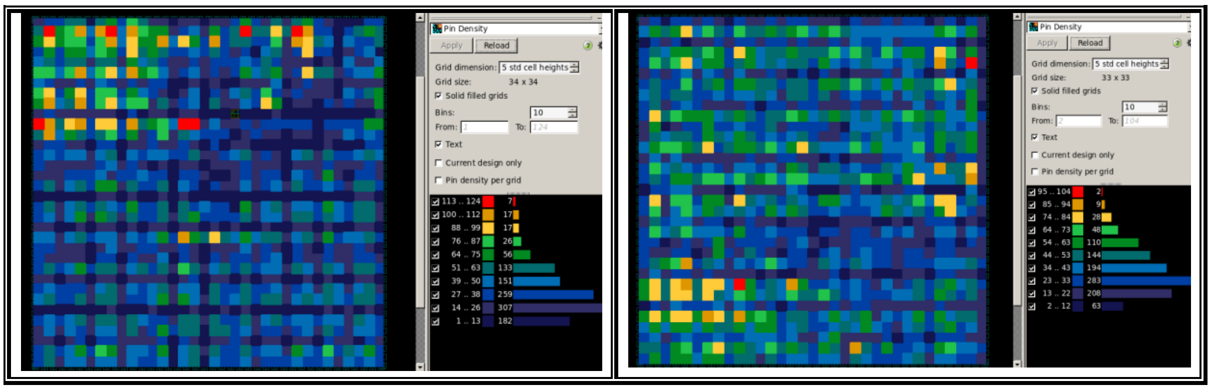
LHS: Grid dimension: 5 std cell heights; Grid size: 34 x 34; ☑ Solid filled grids; Bins: 10; From: 0.1 To: 1.1; ☑ Text; ☐ Current design only

☑	1.100 – 1.000 🟥	0
☑	1.000 – 0.900 🟧	0
☑	0.900 – 0.800 🟨	0
☑	0.800 – 0.700 🟩	0
☑	0.700 – 0.600 🟩	1
☑	0.600 – 0.500 🟦	80
☑	0.500 – 0.400 🟦	250
☑	0.400 – 0.300 🟦	196
☑	0.300 – 0.200 🟦	224
☑	0.200 – 0.008 🟦	405

RHS: Grid dimension: 5 std cell heights; Grid size: 33 x 33; ☑ Solid filled grids; Bins: 10; From: 0.1 To: 1.1; ☑ Text; ☐ Current design only

☑	1.100 – 1.000 🟥	0
☑	1.000 – 0.900 🟧	0
☑	0.900 – 0.800 🟨	0
☑	0.800 – 0.700 🟩	0
☑	0.700 – 0.600 🟩	3
☑	0.600 – 0.500 🟦	47
☑	0.500 – 0.400 🟦	265
☑	0.400 – 0.300 🟦	223
☑	0.300 – 0.200 🟦	310
☑	0.200 – 0.027 🟦	241

Figure 4.44: Cell density comparison between 90nm (LHS) and 32nm (RHS)



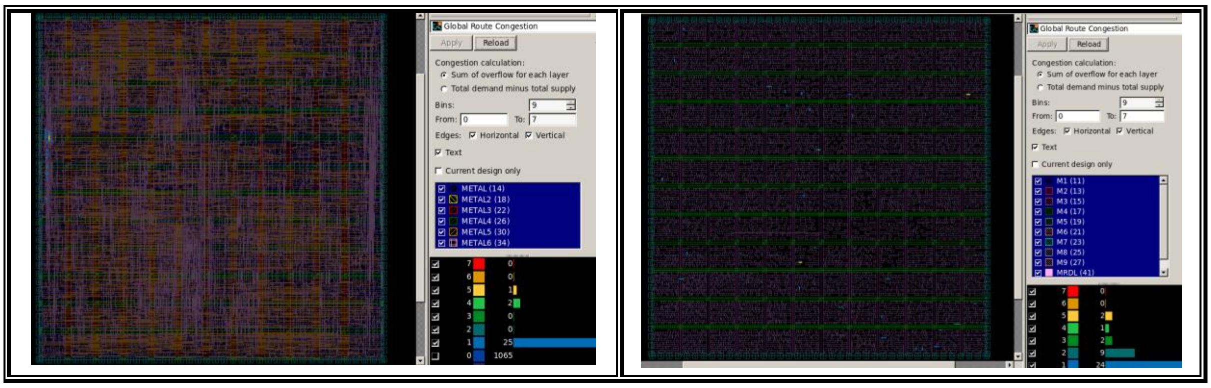
LHS: Grid dimension: 5 std cell heights; Grid size: 34 x 34; ☑ Solid filled grids; Bins: 10; From: 1 To: 124; ☑ Text; ☐ Current design only; ☐ Pin density per grid

☑	113 .. 124 🟥	7
☑	100 .. 112 🟧	17
☑	88 .. 99 🟨	17
☑	76 .. 87 🟩	26
☑	64 .. 75 🟩	56
☑	51 .. 63 🟦	133
☑	39 .. 50 🟦	151
☑	27 .. 38 🟦	259
☑	14 .. 26 🟦	307
☑	1 .. 13 🟦	182

RHS: Grid dimension: 5 std cell heights; Grid size: 33 x 33; ☑ Solid filled grids; Bins: 10; From: 2 To: 104; ☑ Text; ☐ Current design only; ☐ Pin density per grid

☑	95 .. 104 🟥	2
☑	85 .. 94 🟧	9
☑	74 .. 84 🟨	28
☑	64 .. 73 🟩	48
☑	54 .. 63 🟩	110
☑	44 .. 53 🟦	144
☑	34 .. 43 🟦	194
☑	23 .. 33 🟦	283
☑	13 .. 22 🟦	208
☑	2 .. 12 🟦	63

Figure 4.45: Pin density comparison between 90nm (LHS) and 32nm (RHS)



LHS: Congestion calculation: [✓ Sum of overflow for each layer] Total demand minus total supply; Bins: 9; From: 0 To: 7; Edges: ☑ Horizontal ☑ Vertical; ☑ Text; ☐ Current design only

☑ METAL (14)
☑ METAL2 (18)
☑ METAL3 (22)
☑ METAL4 (26)
☑ METAL5 (30)
☑ METAL6 (34)

☑	7 🟥	0
☑	6 🟧	0
☑	5 🟨	1
☑	4 🟩	2
☑	3 🟩	0
☑	2 🟦	0
☑	1 🟦	25
☐	0 🟦	1065

RHS: Congestion calculation: [✓ Sum of overflow for each layer] Total demand minus total supply; Bins: 9; From: 0 To: 7; Edges: ☑ Horizontal ☑ Vertical; ☑ Text; ☐ Current design only

☑ M1 (11)
☑ M2 (13)
☑ M3 (15)
☑ M4 (17)
☑ M5 (19)
☑ M6 (21)
☑ M7 (23)
☑ M8 (25)
☑ M9 (27)
☑ MRDL (41)

☑	7 🟥	0
☑	6 🟧	0
☑	5 🟨	2
☑	4 🟩	1
☑	3 🟩	2
☑	2 🟦	9
☑	1 🟦	24

Figure 4.46: GRC comparison between 90nm (LHS) and 32nm (RHS)

Table 4.5: Comparison between 90nm and 32nm in routing stage

Parameter / Tech Node	90nm	32nm
Timing (ns)		
Critical Path Slack - setup	0.03	0.06
Critical Path Slack - hold	0.11	0.16
Area (um^2)		
Total std cell area	111366.08	23254.68
Core area	385063.31	74275.87
Power (mW)		
Total Power	4.6557	3.4694
Internal Power	2.4471	1.7966
Switching Power	2.1033	0.5684
Leakage Power	0.1053	1.1044

Based on all the data provided, it is clear that 32nm design shows better performance in terms of timing, area and power. The findings are aligning with results shown by Xie, et al. (2015) and Melikyan, et al. (2018), where better QoR is obtained with smaller technology node being used.

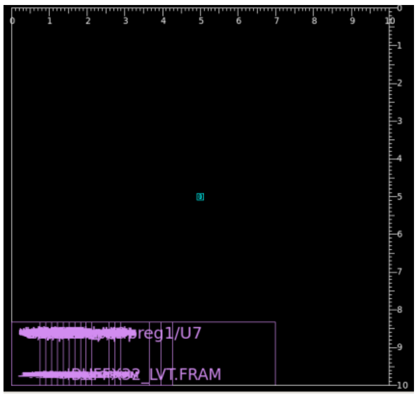
4.3 Quality Of Results (QoR) Comparison Between 5ns and 2ns clock period

In this section, 32nm technology node is taken for both designs, while the clock period varies from 5ns to 2ns. QoR of both designs with different clock period is analyzed.

4.3.1 QoR comparison in Data Setup stage

As soon as design is loaded, an obvious comparison is seen in cell size. Bigger cells are being chosen by the tool in 2ns clock period design. This is because clock period is very small, therefore tool tends to use faster cells in the design. Faster cells generally have bigger cell size. When the size of the transistors in a typical cell is increased, their drive strength is also enhanced. With this larger size, the transistors can more efficiently charge and discharge the capacitance at their output. Consequently, the time constant RC decreases due to a smaller resistance for the same output capacitance, resulting in faster charging and discharging of the output load and ultimately leading to a reduced cell delay (Bhaskar, et al., n.d.).

Other than that, is observed that 2ns clock period design has higher cell count compared to 5ns clock period design as well, with 2ns design having 6989 cell count, and 5ns design having 6175 cell count, 13% increase in cell count when clock period is reduced. By using more cells, tool can divide difficult processes into smaller sequential phases, allowing each cell to accomplish its action in the shorter time allowed by the shorter clock period.



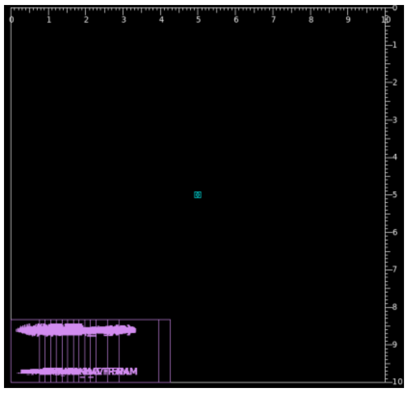
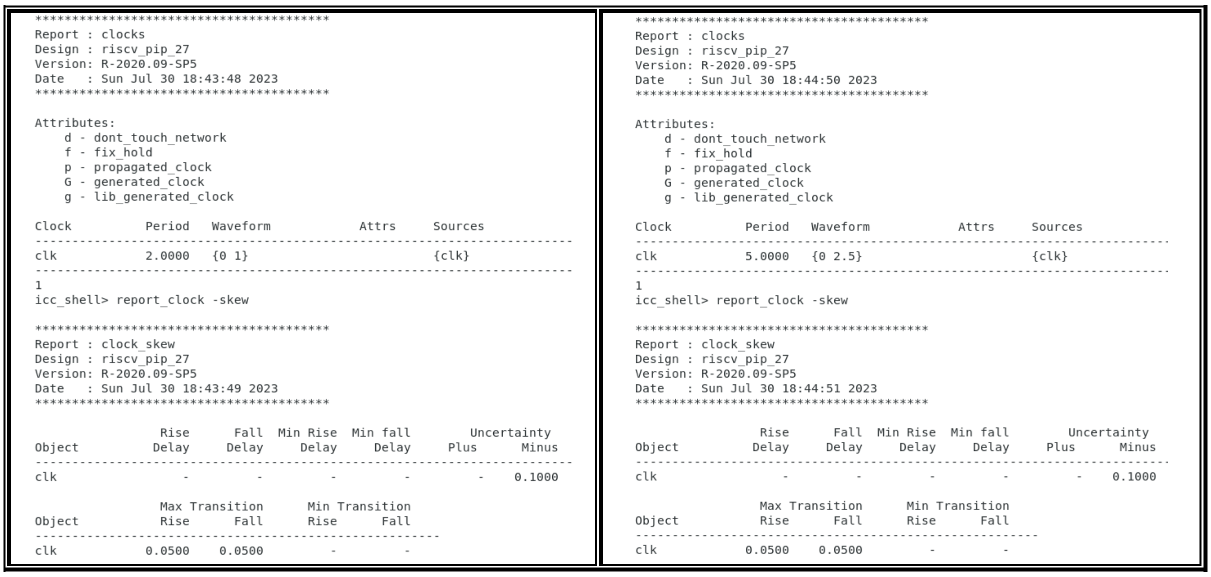


Figure 4.47: cell size comparison between 2ns period (LHS) and 5ns period (RHS)

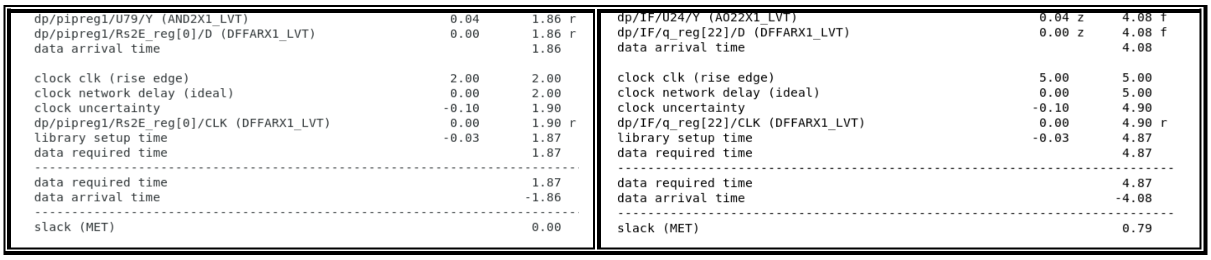
From report_clocks, the designs have clock period of 2ns and 5ns respectively, which other constraints remain the same. In terms of pre-PnR timing, 2ns design barely meet the timing requirement at 0.00ns. better timing slack. “clock clk (rise edge)” indicates the clock period, where it can be seen that 2ns and 5ns are being set respectively.



```
****************************************
Report : clocks
Design : riscv_pip_27
Version: R-2020.09-SP5
Date   : Sun Jul 30 18:43:48 2023
****************************************

Attributes:
    d - dont_touch_network
    f - fix_hold
    p - propagated_clock
    G - generated_clock
    g - lib_generated_clock

Clock          Period   Waveform            Attrs     Sources
--------------------------------------------------------------------------
clk            2.0000   {0 1}                         {clk}
--------------------------------------------------------------------------
1
icc_shell> report_clock -skew

****************************************
Report : clock_skew
Design : riscv_pip_27
Version: R-2020.09-SP5
Date   : Sun Jul 30 18:43:49 2023
****************************************

                 Rise      Fall  Min Rise  Min fall        Uncertainty
Object          Delay     Delay     Delay     Delay     Plus       Minus
--------------------------------------------------------------------------
clk                 -         -         -         -        -      0.1000

                 Max Transition      Min Transition
Object           Rise      Fall      Rise      Fall
------------------------------------------------------
clk            0.0500    0.0500         -         -
```

```
****************************************
Report : clocks
Design : riscv_pip_27
Version: R-2020.09-SP5
Date   : Sun Jul 30 18:44:50 2023
****************************************

Attributes:
    d - dont_touch_network
    f - fix_hold
    p - propagated_clock
    G - generated_clock
    g - lib_generated_clock

Clock          Period   Waveform            Attrs     Sources
--------------------------------------------------------------------------
clk            5.0000   {0 2.5}                       {clk}
--------------------------------------------------------------------------
1
icc_shell> report_clock -skew

****************************************
Report : clock_skew
Design : riscv_pip_27
Version: R-2020.09-SP5
Date   : Sun Jul 30 18:44:51 2023
****************************************

                 Rise      Fall  Min Rise  Min fall        Uncertainty
Object          Delay     Delay     Delay     Delay     Plus       Minus
--------------------------------------------------------------------------
clk                 -         -         -         -        -      0.1000

                 Max Transition      Min Transition
Object           Rise      Fall      Rise      Fall
------------------------------------------------------
clk            0.0500    0.0500         -         -
```

Figure 4.48: clock comparison between 2ns (LHS) and 5ns (RHS)



```
dp/pipreg1/U79/Y (AND2X1_LVT)                         0.04       1.86 r
dp/pipreg1/Rs2E_reg[0]/D (DFFARX1_LVT)                0.00       1.86 r
data arrival time                                                1.86

clock clk (rise edge)                                 2.00       2.00
clock network delay (ideal)                           0.00       2.00
clock uncertainty                                    -0.10       1.90
dp/pipreg1/Rs2E_reg[0]/CLK (DFFARX1_LVT)              0.00       1.90 r
library setup time                                   -0.03       1.87
data required time                                               1.87
-------------------------------------------------------------------------
data required time                                               1.87
data arrival time                                               -1.86
-------------------------------------------------------------------------
slack (MET)                                                      0.00
```

```
dp/IF/U24/Y (AO22X1_LVT)                              0.04 z     4.08 f
dp/IF/q_reg[22]/D (DFFARX1_LVT)                       0.00 z     4.08 f
data arrival time                                                4.08

clock clk (rise edge)                                 5.00       5.00
clock network delay (ideal)                           0.00       5.00
clock uncertainty                                    -0.10       4.90
dp/IF/q_reg[22]/CLK (DFFARX1_LVT)                     0.00       4.90 r
library setup time                                   -0.03       4.87
data required time                                               4.87
-------------------------------------------------------------------------
data required time                                               4.87
data arrival time                                               -4.08
-------------------------------------------------------------------------
slack (MET)                                                      0.79
```

Figure 4.49: pre-PnR timing comparison between 2ns (LHS) and 5ns (RHS)

Table 4.6: Comparison between 2ns and 5ns in data setup stage

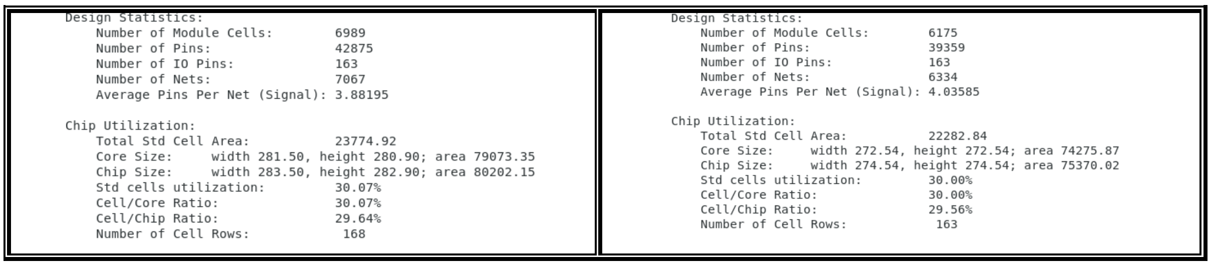
Parameter / Tech Node	2ns	5ns
Cell count	6989	6175
Power pin count	13978	12350
Critical Path Slack (ns)	0.00 (MET)	0.79 (MET)
Critical Path LoL	33	59

4.3.2 QoR comparison in Floorplan stage

Both designs are constrained with core utilization of 30%, and I/O-to-core distance of 1 micron. Same metal layers are also used to create power rings, and power straps have same width, step and groups (number of straps).

```
create_floorplan -core_utilization 0.3 -left_io2core 1 -bottom_io2core 1 -right_io2core 1 -
top_io2core 1

create_rectangular_rings  -nets  {VDD VSS}  -left_segment_layer M4 -right_segment_layer M4
-bottom_segment_layer M3 -top_segment_layer M3

create_power_straps  -direction horizontal  -start_at 23.000 -nets  {VDD VSS}  -layer M4 -
width 1.50 -configure groups_and_step  -num_groups 11 -step 23

create_power_straps  -direction vertical  -start_at 23.000 -nets  {VDD VSS}  -layer M3 -
width 1.50 -configure groups_and_step  -num_groups 11 -step 23
```

After floorplan is generated, the die size is compared with report_design -physical in both designs. 2ns design has 1.06 times bigger core size, compared to 5ns design. This is due to bigger area needed for faster yet bigger cells.

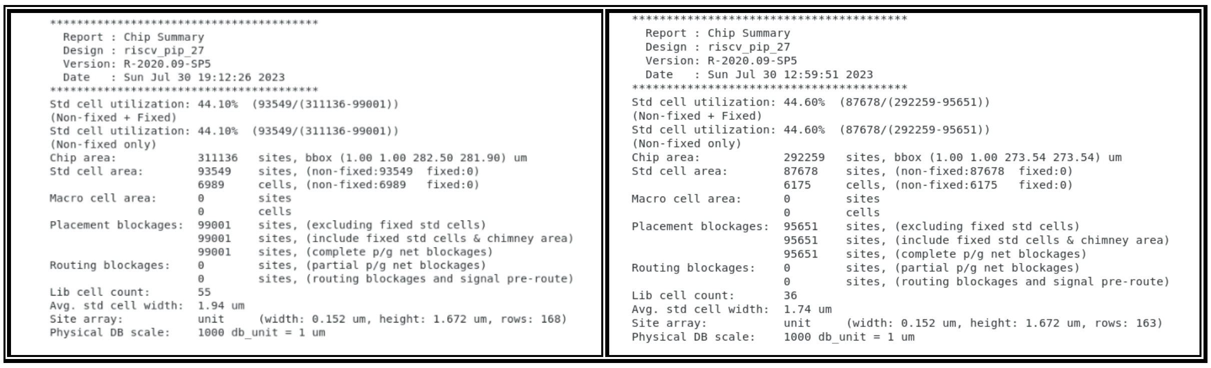


```
Design Statistics:
    Number of Module Cells:        6989
    Number of Pins:                42875
    Number of IO Pins:             163
    Number of Nets:                7067
    Average Pins Per Net (Signal): 3.88195

Chip Utilization:
    Total Std Cell Area:           23774.92
    Core Size:     width 281.50, height 280.90; area 79073.35
    Chip Size:     width 283.50, height 282.90; area 80202.15
    Std cells utilization:         30.07%
    Cell/Core Ratio:               30.07%
    Cell/Chip Ratio:               29.64%
    Number of Cell Rows:            168
```

```
Design Statistics:
    Number of Module Cells:        6175
    Number of Pins:                39359
    Number of IO Pins:             163
    Number of Nets:                6334
    Average Pins Per Net (Signal): 4.03585

Chip Utilization:
    Total Std Cell Area:           22282.84
    Core Size:     width 272.54, height 272.54; area 74275.87
    Chip Size:     width 274.54, height 274.54; area 75370.02
    Std cells utilization:         30.00%
    Cell/Core Ratio:               30.00%
    Cell/Chip Ratio:               29.56%
    Number of Cell Rows:            163
```

Figure 4.50: physical design data comparison between 2ns (LHS) and 5ns (RHS)

Figure 4.51 shows design data post coarse place and route during floorplan stage, where 2ns designs has bigger average standard cell width compared to 5ns design.

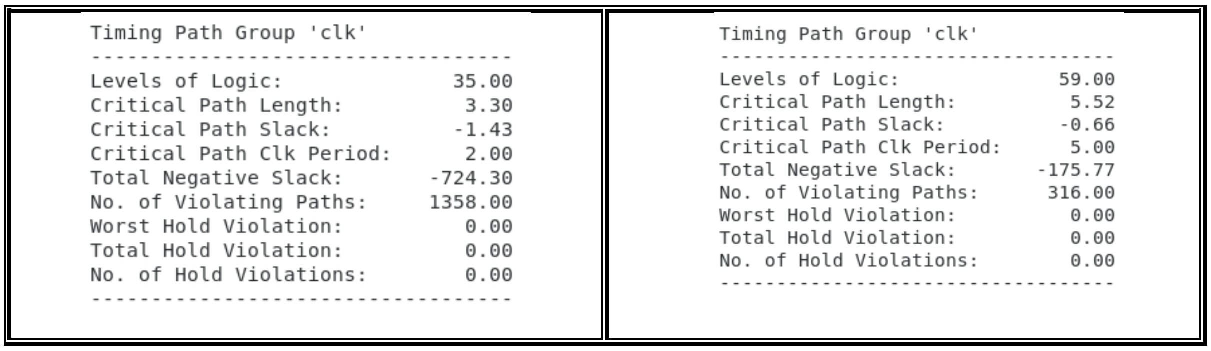


```
****************************************
  Report : Chip Summary
  Design : riscv_pip_27
  Version: R-2020.09-SP5
  Date   : Sun Jul 30 19:12:26 2023
****************************************
Std cell utilization: 44.10%  (93549/(311136-99001))
(Non-fixed + Fixed)
Std cell utilization: 44.10%  (93549/(311136-99001))
(Non-fixed only)
Chip area:            311136   sites, bbox (1.00 1.00 282.50 281.90) um
Std cell area:        93549    sites, (non-fixed:93549  fixed:0)
                      6989     cells, (non-fixed:6989   fixed:0)
Macro cell area:      0        sites
                      0        cells
Placement blockages:  99001    sites, (excluding fixed std cells)
                      99001    sites, (include fixed std cells & chimney area)
                      99001    sites, (complete p/g net blockages)
Routing blockages:    0        sites, (partial p/g net blockages)
                      0        sites, (routing blockages and signal pre-route)
Lib cell count:       55
Avg. std cell width:  1.94 um
Site array:           unit     (width: 0.152 um, height: 1.672 um, rows: 168)
Physical DB scale:    1000 db_unit = 1 um
```

```
****************************************
  Report : Chip Summary
  Design : riscv_pip_27
  Version: R-2020.09-SP5
  Date   : Sun Jul 30 12:59:51 2023
****************************************
Std cell utilization: 44.60%  (87678/(292259-95651))
(Non-fixed + Fixed)
Std cell utilization: 44.60%  (87678/(292259-95651))
(Non-fixed only)
Chip area:            292259   sites, bbox (1.00 1.00 273.54 273.54) um
Std cell area:        87678    sites, (non-fixed:87678  fixed:0)
                      6175     cells, (non-fixed:6175   fixed:0)
Macro cell area:      0        sites
                      0        cells
Placement blockages:  95651    sites, (excluding fixed std cells)
                      95651    sites, (include fixed std cells & chimney area)
                      95651    sites, (complete p/g net blockages)
Routing blockages:    0        sites, (partial p/g net blockages)
                      0        sites, (routing blockages and signal pre-route)
Lib cell count:       36
Avg. std cell width:  1.74 um
Site array:           unit     (width: 0.152 um, height: 1.672 um, rows: 163)
Physical DB scale:    1000 db_unit = 1 um
```

Figure 4.51: chip summary comparison between 2ns (LHS) and 5ns (RHS)

In terms of timing, each design has time violation at this stage, with 2ns design having worst negative slack of -1.43ns, failing almost 75% of clock period. Usually this indicates that the design constraints is too tight, which in this case is clock period being too short.



```
Timing Path Group 'clk'
-----------------------------------
Levels of Logic:              35.00
Critical Path Length:          3.30
Critical Path Slack:          -1.43
Critical Path Clk Period:      2.00
Total Negative Slack:       -724.30
No. of Violating Paths:     1358.00
Worst Hold Violation:          0.00
Total Hold Violation:          0.00
No. of Hold Violations:        0.00
-----------------------------------
```

```
Timing Path Group 'clk'
-----------------------------------
Levels of Logic:              59.00
Critical Path Length:          5.52
Critical Path Slack:          -0.66
Critical Path Clk Period:      5.00
Total Negative Slack:       -175.77
No. of Violating Paths:      316.00
Worst Hold Violation:          0.00
Total Hold Violation:          0.00
No. of Hold Violations:        0.00
-----------------------------------
```

Figure 4.52: timing QoR comparison between 90nm (LHS) and 32nm (RHS)

Table 4.7: Comparison between 2ns and 5ns in floorplan stage

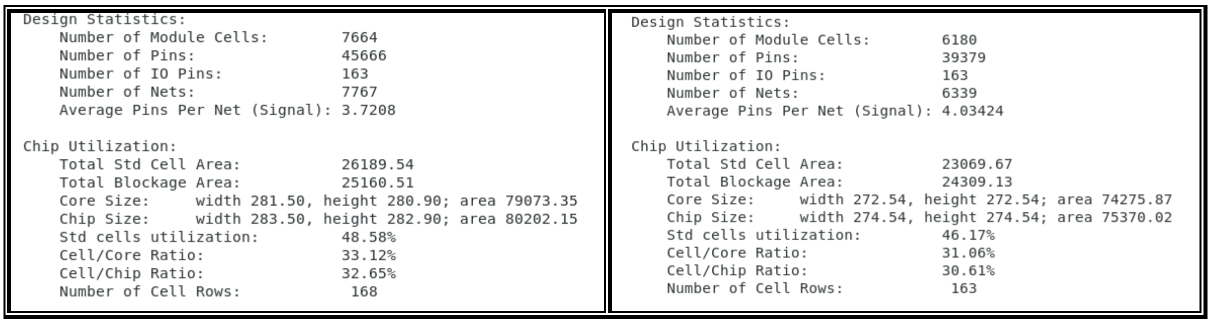
Parameter / Tech Node	2ns	5ns
Die dimension (um) (width x height)	283.50 x 282.90	274.54 x 274.54
Die area (um^2)	80202.15	75370.02
Total standard cell area (um^2)	23774.92	22282.84
Standard cell utilization (pre-coarse PnR) (%)	30.07	30.00
Standard cell utilization (post-coarse PnR) (%)	44.10	44.60
Average std cell width (um)	1.94	1.74
Critical Path Slack (ns)	-1.43 (VIOLATED)	-0.66 (VIOLATED)
Critical Path LoL	35	59

4.3.3 QoR comparison in Placement stage

Both 2ns and 5ns design have placement optimization performed with commands shown below. Post -placement, both designs have 0 errors on legality, which means all cells are placed without overlapping.

```
set_power_options -low_power_placement true
set_separate_process_options -placement false
place_opt -area_recovery -congestion -power
#report_power
psynopt -area_recovery -power
# report_power
```

Post-placement, it can be observed that total standard cell area for 2ns is bigger than 90nm by about 3000um, which is expected as 2ns design uses bigger cells. The cell to core ratio for 2ns design is also higher, about 3.12% over our 30% core utilization target set in floorplan stage.

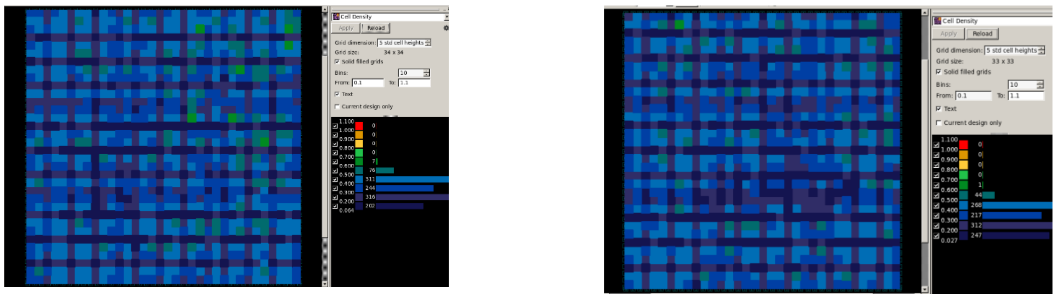


```
Design Statistics:
    Number of Module Cells:        7664
    Number of Pins:                45666
    Number of IO Pins:             163
    Number of Nets:                7767
    Average Pins Per Net (Signal): 3.7208

Chip Utilization:
    Total Std Cell Area:           26189.54
    Total Blockage Area:           25160.51
    Core Size:     width 281.50, height 280.90; area 79073.35
    Chip Size:     width 283.50, height 282.90; area 80202.15
    Std cells utilization:         48.58%
    Cell/Core Ratio:               33.12%
    Cell/Chip Ratio:               32.65%
    Number of Cell Rows:            168
```

```
Design Statistics:
    Number of Module Cells:        6180
    Number of Pins:                39379
    Number of IO Pins:             163
    Number of Nets:                6339
    Average Pins Per Net (Signal): 4.03424

Chip Utilization:
    Total Std Cell Area:           23069.67
    Total Blockage Area:           24309.13
    Core Size:     width 272.54, height 272.54; area 74275.87
    Chip Size:     width 274.54, height 274.54; area 75370.02
    Std cells utilization:         46.17%
    Cell/Core Ratio:               31.06%
    Cell/Chip Ratio:               30.61%
    Number of Cell Rows:            163
```

Figure 4.53: area and utilization QoR comparison between 2ns (LHS) and 5ns (RHS)

Looking at cell density, the placement quality is good, where there are no hotspots seen in both designs. Between range of 0.6-0.7, 2ns design has a slightly higher count than 5ns design, due to bigger cell size.



```
Grid dimension: 5 std cell heights
Grid size:      34 x 34
☑ Solid filled grids
Bins: 10
From: 0.1   To: 1.1
☑ Text
☐ Current design only

1.100
1.000   0
0.900   0
0.800   0
0.700   0
0.600   7
0.500   76
0.400   311
0.300   244
0.200   316
0.064   202
```

```
Grid dimension: 5 std cell heights
Grid size:      33 x 33
☑ Solid filled grids
Bins: 10
From: 0.1   To: 1.1
☑ Text
☐ Current design only

1.100
1.000   0
0.900   0
0.800   0
0.700   0
0.600   1
0.500   44
0.400   268
0.300   217
0.200   312
0.027   247
```

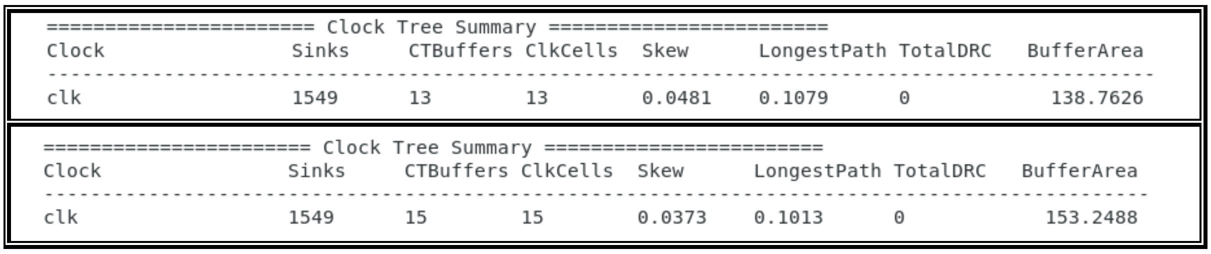
Figure 4.54: Cell density comparison between 2ns (LHS) and 5ns (RHS)

Table 4.8: Comparison between 2ns and 5ns in placement stage

Parameter / Tech Node	2ns	5ns
Number of Module Cells	7664	6180
Standard cells utilization	48.58%	46.17%
Cell-core ratio	33.12%	31.06%

4.3.4 QoR comparison in CTS stage

The clock period is different for both designs, with 2ns and 5ns respectively. The other clock constraints are the same, with uncertainty of -0.1ns and max transition of 0.05ns. Both designs have same global driver count, which is 13 clock cells. The clock buffer area is bigger in 5ns design due to more cells used. However the area per clock buffer is bigger in 2ns design, as larger and faster clock cells are used.



```
======================= Clock Tree Summary ========================
Clock                     Sinks      CTBuffers ClkCells  Skew       LongestPath TotalDRC    BufferArea
-----------------------------------------------------------------------------------------------------
clk                       1549       13        13        0.0481     0.1079      0              138.7626
```

```
======================= Clock Tree Summary ========================
Clock                     Sinks      CTBuffers ClkCells  Skew       LongestPath TotalDRC    BufferArea
-----------------------------------------------------------------------------------------------------
clk                       1549       15        15        0.0373     0.1013      0              153.2488
```

Figure 4.55: Clock tree QoR comparison between 2ns (Top) and 5ns (Bottom)

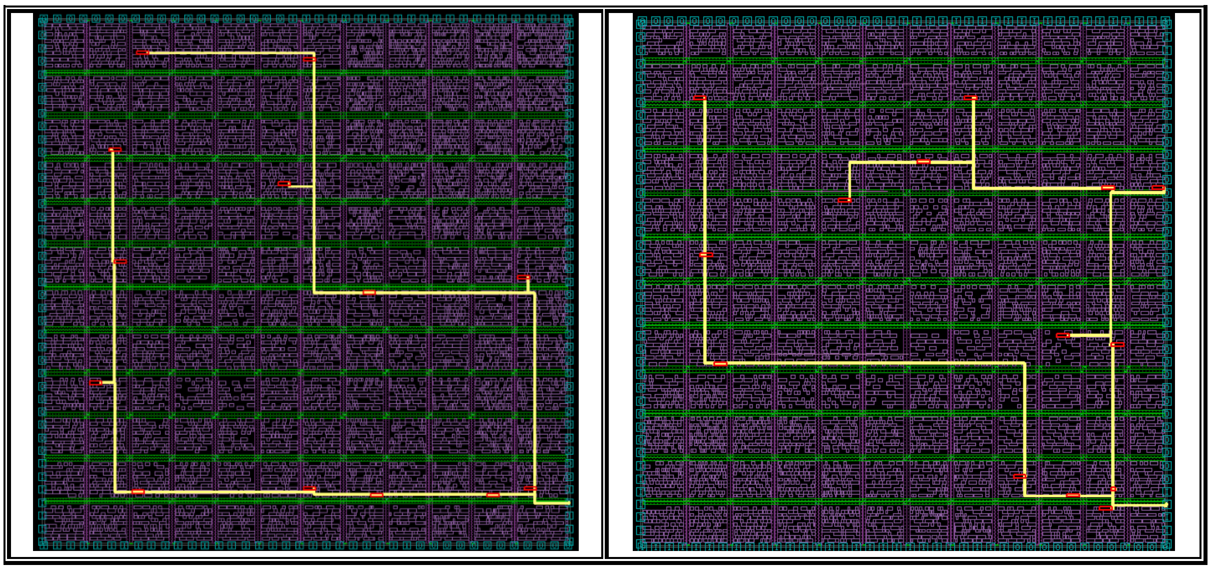
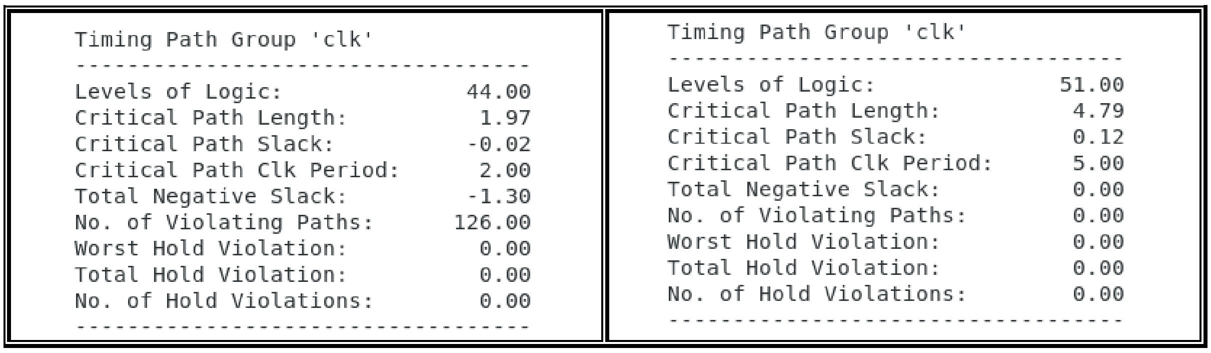


Figure 4.56: Clock tree layout comparison between 2ns (LHS) and 5ns (RHS)

In terms of timing QoR, 2ns design failed to converge the setup timng, with total negative slack of –1.30ns. This is the result of having too tight of clock period. From Figure 4.57, it is observed that 5ns design has a bigger timing margin despite having longer LoL, with critical path slack of 0.12ns, compared to 2ns design which has critical path slack of -0.02ns.

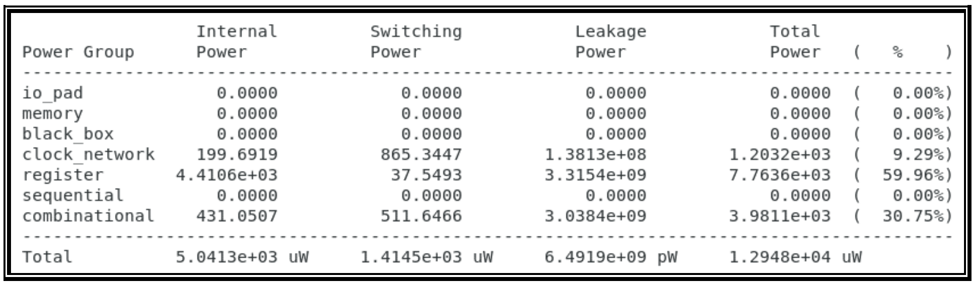


```
Timing Path Group 'clk'
-----------------------------------
Levels of Logic:              44.00
Critical Path Length:          1.97
Critical Path Slack:          -0.02
Critical Path Clk Period:      2.00
Total Negative Slack:         -1.30
No. of Violating Paths:      126.00
Worst Hold Violation:          0.00
Total Hold Violation:          0.00
No. of Hold Violations:        0.00
-----------------------------------
```

```
Timing Path Group 'clk'
-----------------------------------
Levels of Logic:              51.00
Critical Path Length:          4.79
Critical Path Slack:           0.12
Critical Path Clk Period:      5.00
Total Negative Slack:          0.00
No. of Violating Paths:        0.00
Worst Hold Violation:          0.00
Total Hold Violation:          0.00
No. of Hold Violations:        0.00
-----------------------------------
```

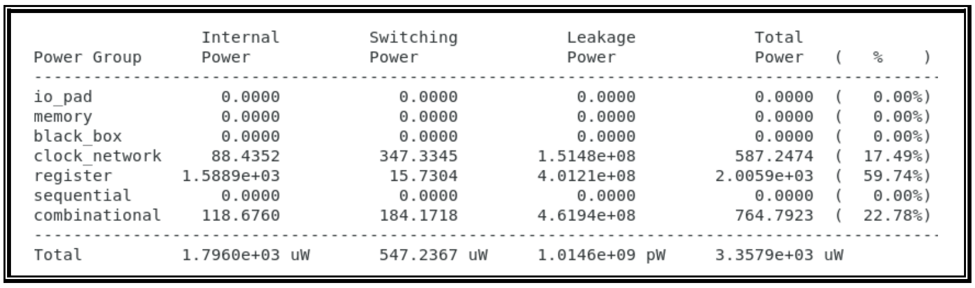
Figure 4.57: Timing QoR comparison between 2ns (LHS) and 5ns (RHS)

On top of that, 5ns design shows better performance in power consumption as well with total power of 3.3579mW, compared to 2ns design with total power of 12.948 mW. Due to shorter clock period, switching activity of transistor is performed at a higher rate, which in this case causes the design to have almost 2.6 times higher switching power. The other power aspects are higher with shorter clock period as well, which results in almost 4 times higher in total power consumption.



```
                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    )
--------------------------------------------------------------------------------------------
io_pad             0.0000           0.0000              0.0000             0.0000  (   0.00%)
memory             0.0000           0.0000              0.0000             0.0000  (   0.00%)
black_box          0.0000           0.0000              0.0000             0.0000  (   0.00%)
clock_network    199.6919         865.3447          1.3813e+08         1.2032e+03  (   9.29%)
register       4.4106e+03          37.5493          3.3154e+09         7.7636e+03  (  59.96%)
sequential         0.0000           0.0000              0.0000             0.0000  (   0.00%)
combinational    431.0507         511.6466          3.0384e+09         3.9811e+03  (  30.75%)
--------------------------------------------------------------------------------------------
Total          5.0413e+03 uW    1.4145e+03 uW       6.4919e+09 pW      1.2948e+04 uW
```

Figure 4.58(a): Power QoR comparison (2ns)



```
                   Internal         Switching           Leakage            Total
Power Group        Power            Power               Power              Power   (   %    )
--------------------------------------------------------------------------------------------
io_pad              0.0000            0.0000             0.0000             0.0000  (   0.00%)
memory              0.0000            0.0000             0.0000             0.0000  (   0.00%)
black_box           0.0000            0.0000             0.0000             0.0000  (   0.00%)
clock_network      88.4352          347.3345         1.5148e+08           587.2474  (  17.49%)
register        1.5889e+03           15.7304         4.0121e+08         2.0059e+03  (  59.74%)
sequential          0.0000            0.0000             0.0000             0.0000  (   0.00%)
combinational     118.6760          184.1718         4.6194e+08           764.7923  (  22.78%)
--------------------------------------------------------------------------------------------
Total           1.7960e+03 uW     547.2367 uW     1.0146e+09 pW     3.3579e+03 uW
```

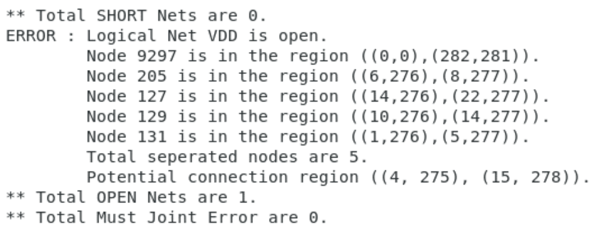
Figure 4.58(b): Power QoR comparison (5ns)

Table 4.9: Comparison between 2ns and 5ns in CTS stage

Parameter / Tech Node	2ns	5ns
Clock buffer count	15	15
Global skew (ns)	0.0481	0.0373
Clock buffer area (um^2)	138.7626	153.2488
Critical path slack (ns)	-0.02	0.12
Critical path slack LoL	44	51
Total Power (mW)	12.948	3.3579
Internal Power (mW)	5.0413	1.796
Switching Power (mW)	1.4145	0.5472
Leakage Power (mW)	6.4919	1.0146

4.3.5 QoR comparison in routing stage

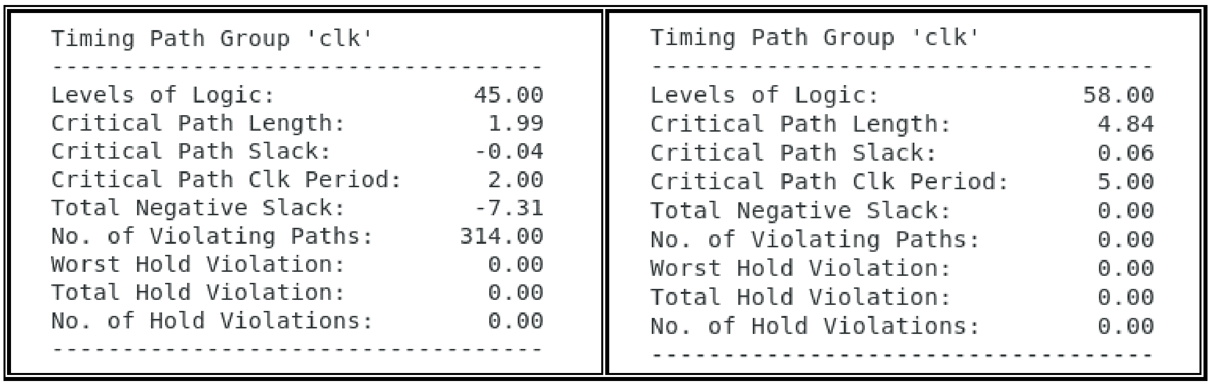
Both 2ns and 5ns design have route optimization performed with command `route_opt -effort high -area_recovery -size_only`. This is the final stage for backend design before inserting filler cells. At this stage, 2ns design has failed LVS check with 1 open net.



```
** Total SHORT Nets are 0.
ERROR : Logical Net VDD is open.
        Node 9297 is in the region ((0,0),(282,281)).
        Node 205 is in the region ((6,276),(8,277)).
        Node 127 is in the region ((14,276),(22,277)).
        Node 129 is in the region ((10,276),(14,277)).
        Node 131 is in the region ((1,276),(5,277)).
        Total seperated nodes are 5.
        Potential connection region ((4, 275), (15, 278)).
** Total OPEN Nets are 1.
** Total Must Joint Error are 0.
```

Figure 4.59: LVS violation in 2ns design

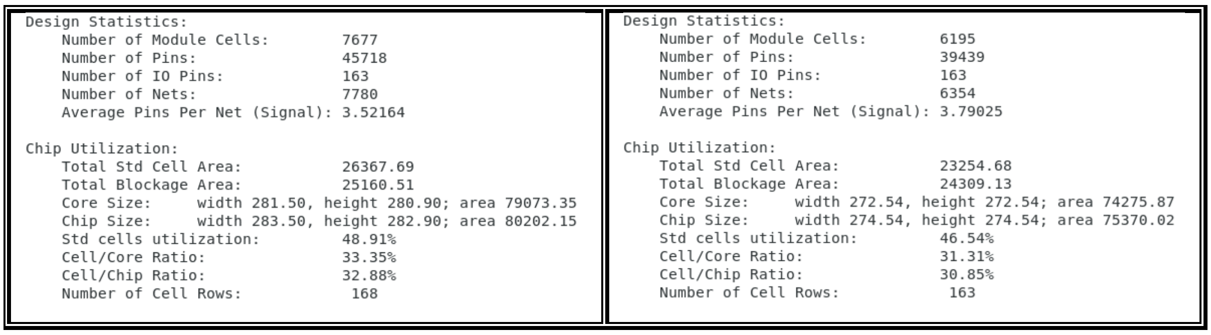
From timing perspective, tool failed to converge the timing for 2ns design, with total negative slack of -7.31ns. Both designs have utilization rate around 30%, but 5ns design has core size 6.07% smaller than 2ns design. Total standard cell area reduced by 11.8% with clock period of 5ns. Another consideration in VLSI design is power. 2ns design has total power of 13.175mW, while 5ns design has a lower power, which is 3.4694mW.



```
Timing Path Group 'clk'
-----------------------------------
Levels of Logic:              45.00
Critical Path Length:          1.99
Critical Path Slack:          -0.04
Critical Path Clk Period:      2.00
Total Negative Slack:         -7.31
No. of Violating Paths:      314.00
Worst Hold Violation:          0.00
Total Hold Violation:          0.00
No. of Hold Violations:        0.00
-----------------------------------
```

```
Timing Path Group 'clk'
-----------------------------------
Levels of Logic:              58.00
Critical Path Length:          4.84
Critical Path Slack:           0.06
Critical Path Clk Period:      5.00
Total Negative Slack:          0.00
No. of Violating Paths:        0.00
Worst Hold Violation:          0.00
Total Hold Violation:          0.00
No. of Hold Violations:        0.00
-----------------------------------
```

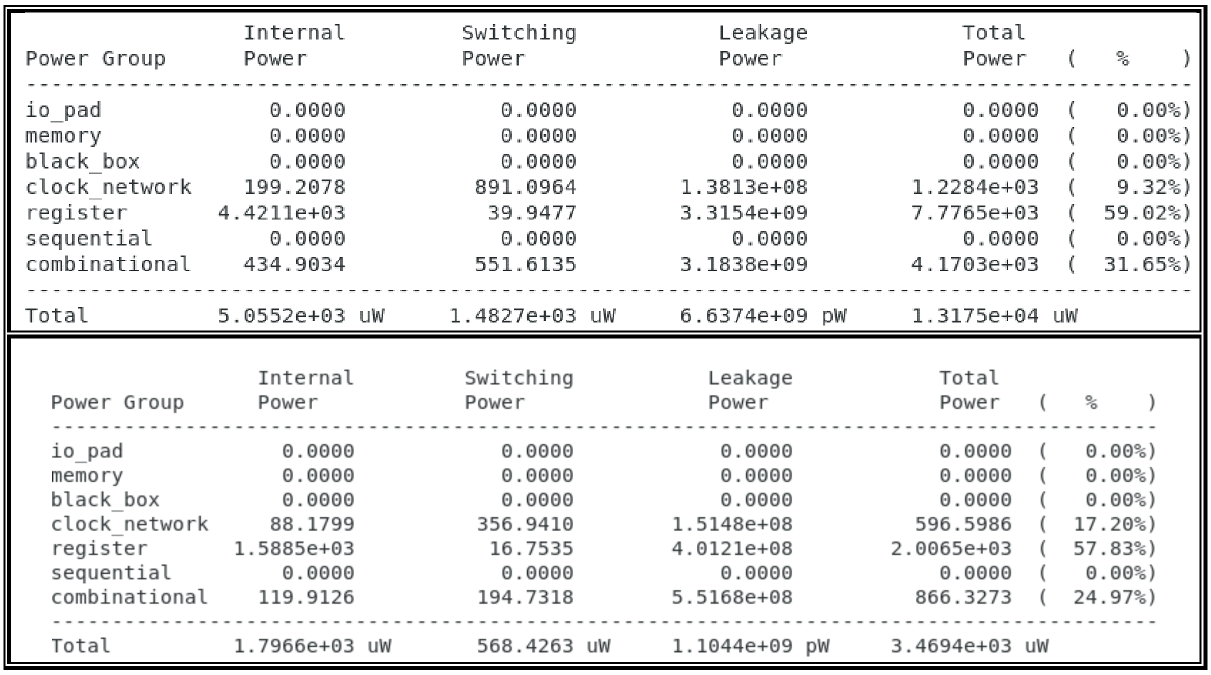
Figure 4.60: Timing QoR comparison between 2ns (LHS) and 5ns (RHS)



```
Design Statistics:
    Number of Module Cells:        7677
    Number of Pins:                45718
    Number of IO Pins:             163
    Number of Nets:                7780
    Average Pins Per Net (Signal): 3.52164

Chip Utilization:
    Total Std Cell Area:           26367.69
    Total Blockage Area:           25160.51
    Core Size:     width 281.50, height 280.90; area 79073.35
    Chip Size:     width 283.50, height 282.90; area 80202.15
    Std cells utilization:         48.91%
    Cell/Core Ratio:               33.35%
    Cell/Chip Ratio:               32.88%
    Number of Cell Rows:            168
```

```
Design Statistics:
    Number of Module Cells:        6195
    Number of Pins:                39439
    Number of IO Pins:             163
    Number of Nets:                6354
    Average Pins Per Net (Signal): 3.79025

Chip Utilization:
    Total Std Cell Area:           23254.68
    Total Blockage Area:           24309.13
    Core Size:     width 272.54, height 272.54; area 74275.87
    Chip Size:     width 274.54, height 274.54; area 75370.02
    Std cells utilization:         46.54%
    Cell/Core Ratio:               31.31%
    Cell/Chip Ratio:               30.85%
    Number of Cell Rows:            163
```

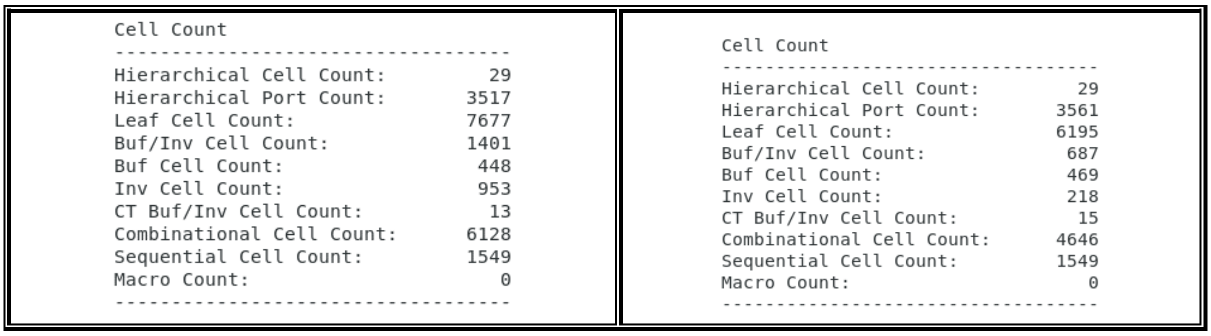
Figure 4.61: Area and physical QoR comparison between 2ns (LHS) and 5ns (RHS)



```
                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    )
--------------------------------------------------------------------------------------------
io_pad              0.0000           0.0000              0.0000             0.0000  (   0.00%)
memory              0.0000           0.0000              0.0000             0.0000  (   0.00%)
black_box           0.0000           0.0000              0.0000             0.0000  (   0.00%)
clock_network     199.2078         891.0964          1.3813e+08         1.2284e+03  (   9.32%)
register        4.4211e+03          39.9477          3.3154e+09         7.7765e+03  (  59.02%)
sequential          0.0000           0.0000              0.0000             0.0000  (   0.00%)
combinational     434.9034         551.6135          3.1838e+09         4.1703e+03  (  31.65%)
--------------------------------------------------------------------------------------------
Total           5.0552e+03 uW    1.4827e+03 uW     6.6374e+09 pW      1.3175e+04 uW
```

```
                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    )
--------------------------------------------------------------------------------------------
io_pad              0.0000           0.0000              0.0000             0.0000  (   0.00%)
memory              0.0000           0.0000              0.0000             0.0000  (   0.00%)
black_box           0.0000           0.0000              0.0000             0.0000  (   0.00%)
clock_network      88.1799         356.9410          1.5148e+08           596.5986  (  17.20%)
register        1.5885e+03          16.7535          4.0121e+08         2.0065e+03  (  57.83%)
sequential          0.0000           0.0000              0.0000             0.0000  (   0.00%)
combinational     119.9126         194.7318          5.5168e+08           866.3273  (  24.97%)
--------------------------------------------------------------------------------------------
Total           1.7966e+03 uW     568.4263 uW      1.1044e+09 pW      3.4694e+03 uW
```

Figure 4.62: Power QoR comparison between 2ns (Top) and 5ns (Bottom)

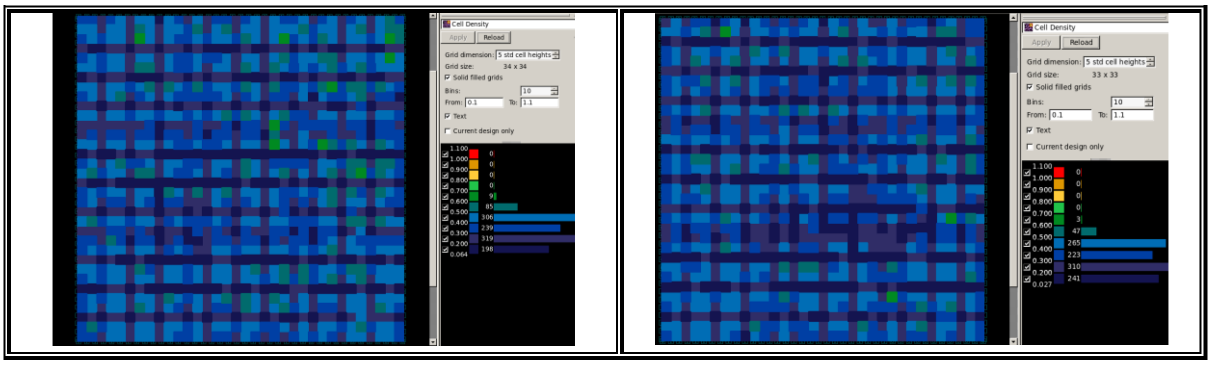


```
Cell Count
-----------------------------------
Hierarchical Cell Count:         29
Hierarchical Port Count:       3517
Leaf Cell Count:               7677
Buf/Inv Cell Count:            1401
Buf Cell Count:                 448
Inv Cell Count:                 953
CT Buf/Inv Cell Count:           13
Combinational Cell Count:      6128
Sequential Cell Count:         1549
Macro Count:                      0
-----------------------------------
```

```
Cell Count
-----------------------------------
Hierarchical Cell Count:         29
Hierarchical Port Count:       3561
Leaf Cell Count:               6195
Buf/Inv Cell Count:             687
Buf Cell Count:                 469
Inv Cell Count:                 218
CT Buf/Inv Cell Count:           15
Combinational Cell Count:      4646
Sequential Cell Count:         1549
Macro Count:                      0
-----------------------------------
```

Figure 4.63: Cell count comparison between 2ns (LHS) and 5ns (RHS)

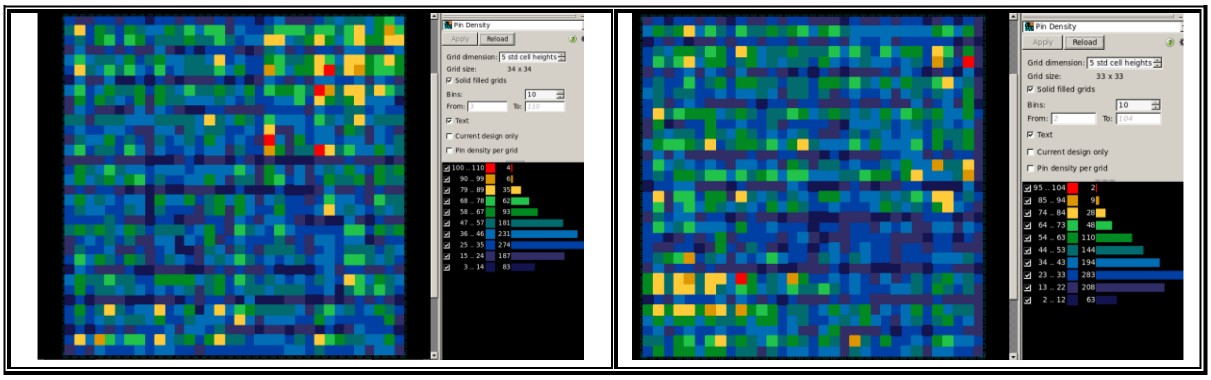
Both designs show satisfactory results in cell density, however 2ns design has more hotspot in terms of pin density and global route congestion (GRC). This is due to higher cell count and bigger cells being used in 2ns design.



```
Cell Density
Grid dimension: 5 std cell heights
Grid size: 34 x 34
☑ Solid filled grids
Bins: 10
From: 0.1   To: 1.1
☑ Text
☐ Current design only
1.100
1.000   0
0.900   0
0.800   0
0.700   0
0.600   9
0.500  85
0.400 306
0.300 239
0.200 319
0.064 198
```

```
Cell Density
Grid dimension: 5 std cell heights
Grid size: 33 x 33
☑ Solid filled grids
Bins: 10
From: 0.1   To: 1.1
☑ Text
☐ Current design only
1.100
1.000   0
0.900   0
0.800   0
0.700   0
0.600   3
0.500  47
0.400 265
0.300 223
0.200 310
0.027 241
```

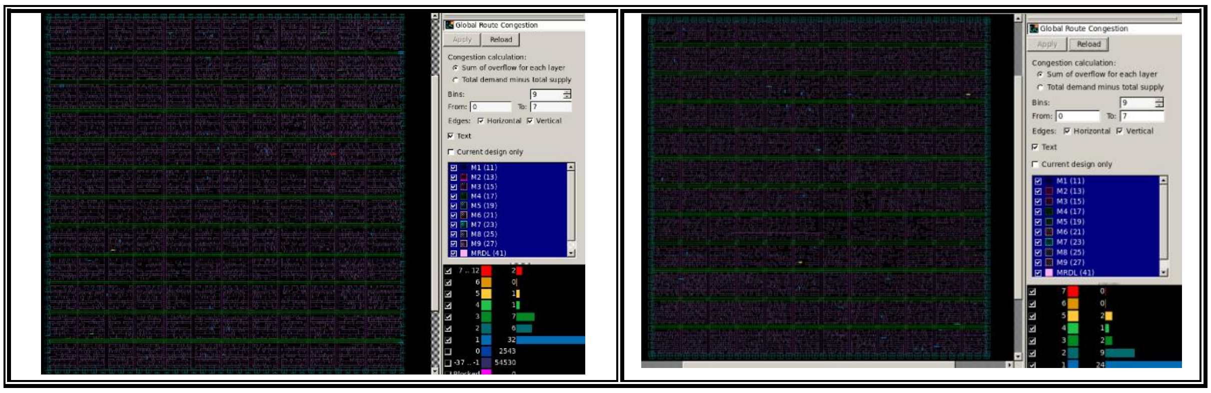
Figure 4.64: Cell density comparison between 2ns (LHS) and 5ns (RHS)



```
Pin Density
Grid dimension: 5 std cell heights
Grid size: 34 x 34
☑ Solid filled grids
Bins: 10
From: 3   To: 110
☑ Text
☐ Current design only
☐ Pin density per grid
100 .. 110    4
 90 .. 99     6
 79 .. 89    35
 68 .. 78    62
 58 .. 67    93
 47 .. 57   181
 36 .. 46   231
 25 .. 35   274
 15 .. 24   187
  3 .. 14    83
```

```
Pin Density
Grid dimension: 5 std cell heights
Grid size: 33 x 33
☑ Solid filled grids
Bins: 10
From: 2   To: 104
☑ Text
☐ Current design only
☐ Pin density per grid
95 .. 104     2
85 .. 94      9
74 .. 84     28
64 .. 73     48
54 .. 63    110
44 .. 53    144
34 .. 43    194
23 .. 33    283
13 .. 22    208
 2 .. 12     63
```

Figure 4.65: Pin density comparison between 2ns (LHS) and 5ns (RHS)



```
Global Route Congestion
Apply  Reload
Congestion calculation:
(•) Sum of overflow for each layer
( ) Total demand minus total supply
Bins: 9
From: 0    To: 7
Edges: [✓] Horizontal  [✓] Vertical
[✓] Text
[ ] Current design only
[✓] M1 (11)
[✓] M2 (13)
[✓] M3 (15)
[✓] M4 (17)
[✓] M5 (19)
[✓] M6 (21)
[✓] M7 (23)
[✓] M8 (25)
[✓] M9 (27)
[✓] MRDL (41)
[✓] 7 .. 12     2
[✓]       6     0
[✓]       5     1
[✓]       4     1
[✓]       3     7
[✓]       2     6
[✓]       1    32
[ ]       0  2543
[ ] -37 .. -1 54530
[ ] Blocked     0
```

```
Global Route Congestion
Apply  Reload
Congestion calculation:
(•) Sum of overflow for each layer
( ) Total demand minus total supply
Bins: 9
From: 0    To: 7
Edges: [✓] Horizontal  [✓] Vertical
[✓] Text
[ ] Current design only
[✓] M1 (11)
[✓] M2 (13)
[✓] M3 (15)
[✓] M4 (17)
[✓] M5 (19)
[✓] M6 (21)
[✓] M7 (23)
[✓] M8 (25)
[✓] M9 (27)
[✓] MRDL (41)
[✓] 7   0
[✓] 6   0
[✓] 5   2
[✓] 4   1
[✓] 3   2
[✓] 2   9
[✓] 1  24
```

Figure 4.66: GRC comparison between 2ns (LHS) and 5ns (RHS)

Table 4.10: Comparison between 2ns and 5ns in routing stage

Parameter / Tech Node	2ns	5ns
Timing (ns)		
Critical Path Slack - setup	-0.04	0.06
Critical Path Slack - hold	0.11	0.16
Area (um^2)		
Total std cell area	26367.69	23254.68
Core area	79073.35	74275.87
Power (mW)		
Total Power	13.175	3.4694
Internal Power	5.0552	1.7966
Switching Power	1.4827	0.5684
Leakage Power	6.6374	1.1044

Based on all the data provided, it is clear that 5ns design shows better performance in terms of timing, area and power. This proves that clock period plays a significant role in determining design quality, thus the clock constraints need to be chosen carefully for digital circuit design.

CHAPTER 5

GANTT CHART

Year 2 Trimester 2												
Plan / Week	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Conduct research planning	🟨	🟨	🟨	🟨								
Literature review					🟨	🟨	🟨	🟨				
Experiment with simple RTL coding and logic synthesis									🟨	🟨		
Software exploration with IC Compiler											🟨	🟨

Year 2 Trimester 3												
Plan / Week	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
• Obtain synthesized netlist from front-end and begin data setup	🟨											
• Draft recipe for backend design – floorplan to routing stage		🟨										
• Floorplan		🟨	🟨	🟨	🟨							
• Placement		🟨	🟨	🟨	🟨							
• Clock Tree Synthesis				🟨	🟨							
• Routing				🟨	🟨							
• Layout Verification						🟨	🟨					
• Output Generation							🟨					
• Thesis writing								🟨	🟨	🟨		
• Thesis submission and presentation											🟨	🟨

CHAPTER 6

CONCLUSION AND FUTURE WORK

In conclusion, the back-end design of a RISC-V processor using IC Compiler involves several critical steps, including automatic floorplanning, placement, clock tree synthesis, routing, and layout verification. These steps are essential to ensure the design meets the timing, power, and area requirements while also ensuring the functional correctness of the design. By following best practices in the back-end design flow, designers can produce high-quality, low-power RISC-V processors that meet the demands of modern computing applications.

Based on a thorough examination of the design data, it is clear that using smaller technology nodes in VLSI design results in considerable gains across numerous crucial performance parameters. Smaller technology nodes provide greater timing performance, as indicated by shorter propagation delays and faster signal switching times. Furthermore, they outperform in terms of area efficiency, thanks to lower transistor sizes. Furthermore, the use of smaller technological nodes results in significant advances in power efficiency, as indicated by lower dynamic and static power usage. These findings illustrate the compelling advantages of using smaller technology nodes, demonstrating their importance in developing the next generation of high-performance, low-power VLSI systems.

On the other hand, the experiments with the 32nm technology node and various clock periods of 2ns and 5ns show that the 5ns clock period design surpasses the 2ns design in terms of timing, area, and power parameters. The findings emphasize the crucial role of the clock period in overall design excellence. The 5ns design has improved area efficiency as a result of a careful balance between circuit elements and chip area. Notably, the use of the 5ns clock period design leads to significant benefits in power efficiency, as evidenced by decreases in both dynamic and static power usage.

As for future work, more thorough optimization technique could be applied on design to further improve the PPA performance, one of the methods is by implementing Multi-Corner Multi-Mode (MCMM) mechanism. By providing numerous data routes and operational modes, the circuit can optimize its performance and power consumption for diverse activities, increasing energy efficiency and overall system performance. The implementation of muti-voltage area design can also be used for power optimization.

References

Ahmed, F., 2021. *VLSI Back-end Adventure.* [Online]
Available at: https://vlsi-backend-adventure.com/cts.html
[Accessed 1 March 2023].

Andrew, W. & Krste, A., 2017. *The RISC-V Instruction Set Manual, Volume I: User-Level ISA.* 2.2 ed. Berkeley: RISC-V Foundation.

Anon., n.d. *TSMC 16/12nm Technology.* [Online]
Available at: https://www.tsmc.com/english/dedicatedFoundry/technology/logic/l_16_12nm
[Accessed 11 October 2022].

Aradhya, H. V. R., Kanase, G. & Y, V., 2021. *RTL to GDSII of Harvard Structure RISC Processor.* Bangalore, India, IEEE.

Bhaskar, A., Sharma, S. & Pratap, R., n.d. *Strategy To Fix Register-to-Register Timing For large Feedthrough Blocks Having Limited Internal Pipelines.* [Online]
Available at: https://www.design-reuse.com/articles/47643/strategy-to-fix-register-to-register-timing.html
[Accessed 25 July 2023].

David, K. & Andrew, L., 2019. *Microprocessors.* s.l.:CRC Press.

Gartner, 2020. *Custom ICs Based on RISC-V Will Enable Cost-Effective IoT Product Differentiation*, s.l.: s.n.

Ho, V., Ma, K., Thai, H. & Le, D., 2021. *Implementation of a Dual-core 64-bit RISC-V on 7nm FinFET Process.* Ho Chi Minh City, Vietnam, IEEE, pp. 28-32.

Intel, 2019. *Intel® Quartus® Prime Lite Edition Design Software Version 19.1 for Windows.* [Online]
Available at: https://www.intel.com/content/www/us/en/software-kit/664527/intel-quartus-prime-lite-edition-design-software-version-19-1-for-windows.html?
[Accessed 25 February 2023].

Jim´enez, V. M. M., 2021. *Impact of physical low power techniques in a RISC-V processor*, Barcelona: s.n.

Kanase, G. & M, N., 2021. *ASIC Design of a 32-bit Low Power RISC-V based System Core for Medical Applications.* Coimbatre, India, IEEE.

Khan, Z. R. et al., 2022. *GHAZI: An Open-Source ASIC Implementation of RISC-V based SoC*, Pakistan: s.n.

Melikyan, V. et al., 2018. Multi-Voltage and Multi-Threshold Low Power Design Techniques for ORCA Processor Based on 14 nm technology. *2018 IEEE 38th International Conference on Electronics and Nanotechnology* , pp. 116-120.

Min, J., 2022. *Case study: optimizing PPA with RISC-V custom extensions in TWS earbuds.* [Online]
Available at: https://www.embedded.com/case-study-optimizing-ppa-with-risc-v-custom-

extensions-in-tws-earbuds/
[Accessed 14 October 2022].

Moreno, D., 2019. *RISC Processor Steps to Fabrication,* Northridge: California State University.

Pasquale, D. S. et al., 2017. *Slow and Steady Wins the Race? A Comparison of Ultra-Low-Power RISC-V Cores for Internet-of-Things Applications.* Thessaloniki, Greece, IEEE.

Petrosyan, P. et al., 2016. *Clock gating and multi-VTH low power design methods based on 32/28 nm ORCA processor.* Batumi, Georgia, IEEE.

Research, S., 2019. *RISC-V Market Analysis: The New Kid on the Block,* s.l.: s.n.

Synopsys, 2018. *Design Compiler.* [Online]
Available at: https://www.synopsys.com/implementation-and-signoff/rtl-synthesis-test/dc-ultra.html
[Accessed 25 February 2023].

Synopsys, 2019. *IC Compiler.* [Online]
Available at: https://www.synopsys.com/implementation-and-signoff/physical-implementation/ic-compiler.html#:~:text=Synopsys%20IC%20Compiler%E2%84%A2%20II,technologies%2C%20while%20enabling%20unprecedented%20productivity.
[Accessed 5 March 2023].

TSMC, 2022. *TSMC 3nm Technology.* [Online]
Available at: https://www.tsmc.com/english/dedicatedFoundry/technology/logic/l_3nm
[Accessed 16 October 2022].

Waterman, A., Lee, Y., Patterson, D. A. & Asanovi, K., 2014. *The RISC-V Instruction Set Manual. Volume 1: User-Level ISA, Version 2.0.* [Online]
Available at: https://apps.dtic.mil/docs/citations/ada605735
[Accessed 11 10 2022].

Xie, Q. et al., 2015. Performance Comparisons Between 7-nm FinFET and Conventional Bulk CMOS Standard Cell Libraries. *IEEE Transactions on Circuits and Systems II: Express Briefs,* 62(8), pp. 761 - 765.