MULTI-SCENARIO SoC TIMING CLOSURE AND ECO GENERATION IN XEON SERVER CHIPS AT ADVANCED TECH NODES

A Thesis submitted in partial fulfillment of the requirement for the Award of the Degree of

MASTER OF TECHNOLOGY

in VLSI DESIGN

Submitted By

ISHIKA JAIN

602362013

Under Supervision of

Dr. KS Sandha

Associate Professor and Associate Head

Dr. Ankush Kansal

Professor

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY

(A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB

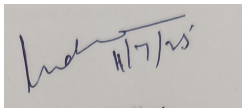
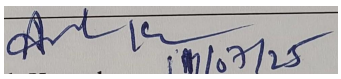
JULY, 2025

DECLARATION

I, **Ishika Jain** hereby declare that the work presented in this thesis entitled **"MULTI SCENARIO SoC TIMING CLOSURE AND ECO GENERATION IN XEON SERVER CHIPS AT ADVANCED TECH NODES"** in the partial fulfilment of the requirements for the award of the Degree of **Master of Technology (VLSI Design)** and submitted at the **Department of Electronics & Communication Engineering**, Thapar Institute of Engineering and Technology (Deemed to be University), Patiala, is an authentic record of work carried out under the supervision of Mr. Karthik Ramanathan, Physical Design Engineering Manager, Intel and Dr. KS Sandha, Associate Professor and Associate Head, Department of Electronics & Communication Engineering, TIET, Patiala from July'24 to June'25. The matter presented in this has not been submitted either in part or fully to any other university or institute fo0r the award of any other degree.

Date: 11-07-2025

Ishika Jain
MTech VLSI Design
602362013

 Mr. Karthik Ramanathan Physical Design Engineering Manager, Intel Corporation Date: 11-07-2025	 Dr. KS Sandha Associate Professor and Associate Head Department of Electronics and Communication Engineering, Thapar Institute of Engineering and Technology, Patiala Date: 11-07-2025
	 Dr. Ankush Kansal Professor Department of Electronics and Communication Engineering, Thapar Institute of Engineering and Technology, Patiala Date: 11-07-2025

CERTIFICATE

Regd. Office:
Intel Technology India Private Limited
23-56P, Outer Ring Road,
Devarabeesanahalli, Varthur Hobli website: www.intel.in
Bellandur Post
Bangalore 560 103, India
CIN-U85110KA1997PTC021606

Tel: +91-80-2605 3000
Fax: +91-80-2605 6190

To Whomsoever It May Concern

WWID: 12251779

Employee Name: Ishika Jain

Internship Dates: 24/06/2024 to 23/06/2025

The letter is to confirm the mentioned above has undergone internship at Intel Technology India Private Limited.

We wish you all the best for your future assignments.

Yours Sincerely

Gowre Saseedaran

Date: June 27, 2025

Place: **Bangalore**

ACKNOWLEDGEMENT

I am highly indebted to my project manager, Mr. Karthik Ramanathan (Physical Design Engineering Manager, Intel Corporation) for his kind support and guidance without which this thesis would not be possible. His untiring encouragement has always been an endless source of motivation for me.

I want to take this opportunity to sincerely thank Dr. Kulbir Singh (Head of Department, ECE), who allowed me to carry out this internship and provided me with his valuable guidance and suggestions.

I want to express my sincere thanks and respect to Dr. KS Sandha (Associate Professor and Associate Head, ECED) and Dr. Ankush Kansal (Professor, ECED) who have provided the moral support for completing this work.

I want to dedicate this work to my parents and friends who have always motivated me the importance of work. Finally, I would like to thank my friends whose, constant inspiration, and affection helped me to complete this project.

Ishika Jain

602362013

MTech VLSI Design

ABSTRACT

In today's modern VLSI chips, as the process nodes continue to shrink, the impact of process variations has increased significantly. This has given rise to research into extending traditional static timing analysis so that it can be performed statistically. Industry standards now work on the much more advanced Parametric On-Chip Variation (POCV) approach leaving behind the traditional on-chip variation (OCV) approaches. The performance of an integrated circuit (IC) largely depends upon its clock frequency and is mostly evaluated considering the worst-case delays. This pessimism has led industry experts to optimize the design further. The main contributor to the delay of the circuit is the interconnect delays. Hence, the number of scenarios required to analyse and fix the design is enormous and increasing, at deep sub-micron levels. Engineering Change Orders (ECOs) at the deep sub-micron level are used to make small, incremental changes without having to go through the entire design process again. ECOs are essential for handling late-stage design modifications and are critical for saving time, reducing costs, and improving the overall efficiency of the design cycle.

This work can be briefly divided into three parts (all performed at the full chip level):

i. Performing the timing closure (bifurcated as die wise)
ii. Implementing Engineering Change Orders (ECOs) to fix the hold violations
iii. Performing the quality checks, before final signoff

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO THE PROBLEM

In advanced IC designs, the need for higher performance and rich SoC features has led to increased design complexity. The number of transistors on a chip doubles every 18 months. This enables faster speeds and higher computing capabilities, but it also creates new challenges for physical implementation and timing closure**.** The logic paradigm is slowly shifting towards an interconnect-dominated from the traditional logic-dominated one. The gate delay has increased drastically due to this. In addition, the increased clock rates pose a big challenge for timing and crosstalk delay signoff. Also, the increment in the process corners, i.e., the multiple scenarios due to the variation in manufacturability, the complexity in the timing closure process has increased manifold. So, cleaning all the violations at the design level and taping in with accurate sign-off results is essential to avoid post-manufacturing defects. Efforts are being made to improve the timing closure in the physical design stages, but the post-routing stages also witness a huge amount of timing violations. This may also require manual intervention, which can cause increased iterations and increased run time. The current multi-corner multi-mode design needs more reliable methods to improve the timing closure, eventually leading to a more accurate timing signoff [3].

Following are some of the major challenges that are faced during signoff at the deep sub-micron level:

1. **Interconnect Delays:** At deep submicron nodes, interconnect delays (wire delays) become comparable to, or even greater than, gate delays. RC delay in interconnects becomes more significant, complicating timing closure.
2. **Clock Skew and Jitter:** Clock synchronization across the chip becomes more difficult due to increase in clock skew (difference in arrival times of the clock signal) and clock jitter (timing variability in the clock signal).
3. **Signal Propagation Delay:** Variability in transistor performance and interconnects also introduces uncertainties in signal propagation delay, making it harder to guarantee that signals arrive at their destinations on time, especially in high-speed designs.
4. **Crosstalk:** As wires and transistors become more densely packed, parasitic capacitance between neighbouring wires increases. This can lead to crosstalk, where a signal on one wire unintentionally affects a nearby wire, causing timing errors and glitches.

5. **Process Variation:** At smaller geometries, the manufacturing process becomes less precise, leading to variations in transistor parameters (e.g., threshold voltage, channel length, oxide thickness). These process variations can cause discrepancies between intended and actual performance, creating variability in timing, power, and yield.

So, as the technology nodes are shrinking, a comprehensive approach is required that considers numerous variations and helps achieve accuracy, without hampering the performance. Fewer iterations and less costly approaches are now needed. This work addresses the aforesaid requirements by providing industry-specific solutions to improve the simulation time.

1.2 OBJECTIVE

Timing closure is the process that determines the speed of the chip by satisfying the timing constraints. It ensures that all the signals arrive at the correct time for smoother chip operation. Complex design rules, low-power circuitry design techniques, and signal integrity issues are just some of the advanced-node challenges impacting design closure. Higher frequency targets and lower power envelopes require more Engineering Change Order (ECO) cycles. ECO cycles on large, advanced node designs require many iterations due to unpredictable closure techniques. Power and thermal analysis drive decisions on the packaging and system-level heat dissipation. In turn, solutions such as multi-die packaging, where the chips, packaging, and boards drive the throughput and power, lead to a more pressing demand for proper timing. Static Timing Analysis (STA) checks all possible paths for timing violations to validate a design's timing performance during timing signoff. However, for more complex designs, basic STA might not be enough; distributed static timing analysis (DSTA) is the preferred method of timing analysis of very large designs with more than 400M cells.

This work has its main focus on ECO timing closure, that usually is a multiple iteration process.

The major objectives of this work are summed up as follows:

1. New technology processes create changes in design factors, such as process, voltage, and temperature. These factors are known as "corners" (PVT). As the number of corners increases, the number of timing issues also rises. This means that checking and fixing the design for all these scenarios makes it harder to achieve timing closure and complicates the process.
2. Creating new corners in the design can lead to new violations. This makes it difficult for designers to finalize their work after each round of fixing these issues. We need to address

the problem of introducing violations in one or several corners while fixing problems in other corners. This can complicate the sign-off process.

3. During this flow of timing closure, some Electrical Rule Checks and other quality checks are also performed to check the correctness and goodness of the database. This also maintains the robustness of the chip.
4. An ECO flow handles multiple tools for Static Timing Analysis (STA), place and route, and final ECO implementation. This multi-tool solution requires thorough knowledge of the process technology and its associated tools. For instance, fixing the hold violations through an ECO implementation can introduce thousands of setup violations in return which need to be addressed on top priority. As a result, this may require numerous STA runs and hence an increased runtime.

1.3 FLOW EXECUTION

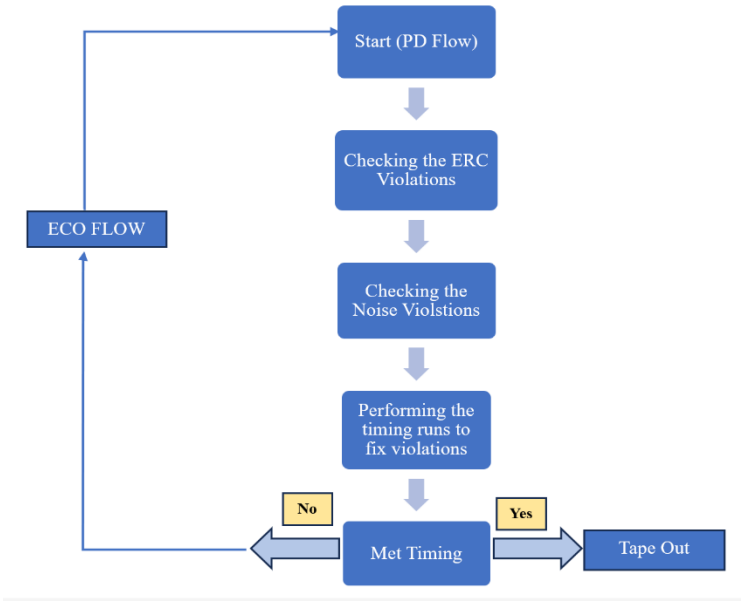


Figure 1.1: Execution Flow for Timing Closure/ ECO Generation

As per Figure1.1, the flow execution can be summarised as:

1. Starts with the VLSI Physical Design backend flow.
2. The second step is to check some of the Electrical Rules that are bucketed into various categories, according to quality, clocks, timing, interface, and power.
3. The third step is to perform the static noise analysis, which is necessary for the crosstalk and glitch analysis.
4. The fourth step comprises checking the setup and hold violations in the design by performing the Static Timing Analysis.
5. If the timing has been met, the design is ready for tape in, otherwise, ECO has to be implemented, which is an iterative process, and performed until all the violations are clear.

CHAPTER 2

LITERATURE REVIEW

This chapter summarises the understanding of the background, which is done through the study of various literature that includes white papers from industries, research papers from scholars, Xeon product overview, and basics of Static Timing Analysis.

2.1 UNDERSTANDING OF THE BACKGROUND

Since this work is based on the timing closure carried out on Xeon server chips, the backdrop consists of general knowledge of the Xeon server chips, going from the brief know-how of the architecture [14], the latest technologies behind advanced manufacturing, and the present-day challenges that are being faced at the design level.

A conceptual understanding of the static timing analysis parameters, variation analysis, violation fixing, pessimism and derates, analysis of the multiple scenarios, and brief about the Engineering Change Orders, has been presented in the subsequent sub chapters.

2.1.1 Xeon Server Chips

1. The product family of Intel, which finds its use case in data centers, servers, and nowadays, is supported by AI.
2. Intel Xeon 6 processors deliver new degrees of performance [15] with more cores, a choice of microarchitecture, additional memory bandwidth, and exceptional input/output (I/O) across a range of workloads.
3. Introduces an innovative modular x86 architecture that is purpose-built for your unique needs and workloads across private, public, and hybrid clouds.

Knowing the Server Processors

Processors are the computer's brain, executing calculations, tasks, and functions. Server processors differ from PC processors in that they are typically designed to handle heavier, more complex workloads, such as:

1. Email exchange, file sharing, and database transactions for office or remote workers and customers worldwide.

2. Telecommunications services that process network traffic routing for millions of connected devices.

3. Connection and control of multiple devices on a factory line, from conveyor belts to robotic arms and cameras.

4. Extreme computing tasks such as using AI to map genomes or simulate global weather patterns.

Intel Processors with Performance and Efficiency Cores

On Intel Xeon chips, P-Cores deliver high performance for demanding server tasks, and E-Cores enhance power efficiency, making this hybrid approach particularly useful in optimizing workloads in enterprise and cloud computing environments [13] [15]. A brief difference between the P cores and E cores is given in Table 2.1:

P Cores	E Cores
Purpose: Designed for high-performance, resource-intensive tasks.	**Purpose:** Designed for power-efficient multitasking and handling background or lightweight tasks.
Performance: P-Cores are more powerful, offering higher clock speeds and better single-threaded performance.	**Performance:** E-Cores are smaller, slower, and more power-efficient, focusing on parallelism and multi-threading rather than raw speed.
Use Case: Best suited for demanding tasks like gaming, heavy content creation (video editing, 3D rendering), and workloads that require fast processing.	**Use Case:** Ideal for tasks like background system processes, lightweight applications, or multitasking with minimal power consumption.
Architecture: Typically based on Intel's "Golden Cove" microarchitecture (for Alder Lake).	**Architecture:** Typically based on Intel's "Gracemont" microarchitecture (for Alder Lake).
Hyper-threading: Supported, i.e., each P-Core can handle multiple threads.	**Hyper-threading:** Not supported, i.e., each E-Core can handle only one thread at a time.

Table 2.1: Difference between P-Cores and E-Cores

2.1.2 Conceptual Understanding: Static Timing Analysis

This chapter will describe the conceptual understanding of what is STA and what parameters are to be considered while performing timing closure at the full chip level. Each subchapter presents a different concept.

2.1.2.1 Defining a Scenario

Mode: A mode describes how a design operates and includes factors like clock settings, voltage supply, various synopsys design constraints, and libraries. A chip can operate in several modes, such as functional mode, test mode, or standby mode.

Corner: A corner shows the differences caused by changes in process, voltage, and temperature. These differences are explained in a set of libraries included with the process kit.

Corners can be of two types:

1. **Front End of Line (FEOL) corners:** Wafer based devices are produced using this process. These include the transistors, which are required to form fully isolated CMOS elements [4]. Three types of corners, viz typical, fast and slow exist, depending on the doping concentration. A corner can be represented by 2 letter designation, the first part being the n-channel MOSFET (NMOS), and the second being the p-channel MOSFET (PMOS). Similarly, 5 combinations exist- FF, SS, FS, SF, and TT, corresponding to fast fast, slow slow, fast slow, slow fast and typical typical NMOS ans PMOS respectively.

2. **Back End of Line (BEOL) corners:** In this stage, metal interconnects are added. The On-chip Variations (OCV), cause delay due to coupling effects caused due to surrounding interconnects [4] [16]. Hence, the interconnect variations from maximum resistance (minimum cross-sectional area) to maximum capacitance (minimum cross-sectional area), account for the new set of corners: Maximum interconnect capacitance (Cmax), Minimum interconnect capacitance (Cmin), Typical, Maximum interconnect resistance-capacitance (RCmax), Minimum interconnect resistance-capacitance (RCmin).

\

2.1.2.2 Conceptual Details

This section focuses on the basics of timing analysis in detail:

1. **Static timing analysis:** It refers to inspecting the design for timing violations. The two methods used are dynamic timing analysis and STA. DTA validates the design by checking its functionality along with timing. The design here is validated for multiple input stimuli. [12]. This process is extremely iterative as compared to static timing analysis, where the design is validated statistically for timing and does not depend on the inputs [5]. Thus, for million gate semiconductor chip, STA is a faster and a wider used approach. The main objective of STA is to verify whether the chip can function correctly at the desired speed and frequency.
2. **Timing paths:** There can be multiple paths through which logic can propagate to the desired destination. Start Point is the place where the path begins to originates and the data gets launched by a clock or where the data is expected to be present at a specific time. It is either be a clock pin of a sequential element or an input port of a synchronous design. Similarly, the point in the design where the data gets captured by a clock edge or gets terminated is called an endpoint. It can either be the input of a register or the output port of a synchronous design. Thus, a timing path is a sequence that starts at a valid start point, goes through some combinational logic, and ends at a valid endpoint.
3. **Path groups:** To ease the analysis, all the timing paths are segregated into different groups, called Path Groups, corresponding to same endpoints.
4. **Timing Arcs and Unateness:** A combinational cell, like an AND gate or NAND gate, can have several input pins. Each input pin has a timing arc that shows how changes in input affect the output [5]. These multiple timing arcs mean that different input changes can drive the output in different ways. This behavior is called "unateness." There are three types of unateness:

   Positive unateness: A rising or falling change at the input causes a similar rising or falling change at the output.

   Negative unateness: The input and output changes are different. For example, a rising input change leads to a falling output change, or the other way around.

   Non-unateness: In this case, you cannot predict the output change based on the input changes. Understanding these types of unateness helps in analyzing how combinational cells work.

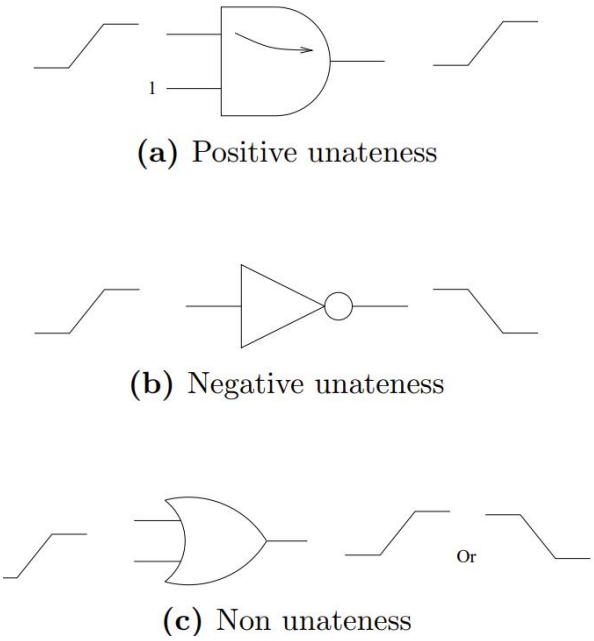


Figure 2.1: Unatness in Logic Gates

5. **Common path pessimism:** Because of the varying device parameters of the chip, as explained in section 1.2, a cell in a path can have two types of delays, **min delay for short path** considerations and **max delay for long path** considerations. At times for verifying the correctness of the design, it is required that max delays along one is considered and min delays along the other path. In such a case, if there exists a cell that lies in a path common to both the cells, the max delay of this cell is considered for the max delay path and the min delay of the cell is considered for the min delay path. However, a cell cannot have two types of delays at the same time and hence this conflict must be taken into consideration for timing verification. The difference between the delays at a common point is called clock re-convergence pessimism.
6. **Timing violations due to clock skew:** Clock skew represents the difference between the arrival time at launch and capture flop across the chip [5]. Setup and Hold violations occur due to two types of skews as described below:

i. **Positive clock skew:** Positive Clock skew occurs when clock arrival time at destination is greater than arrival time at source. The skew calculated in such a case is called positive clock skew and is given by eq.1.1

HoldSlack = Tcq + Tcomb − Tskew − Thold (1.1)

ii. **Negative clock skew:** It refers to a condition where the clock signal arrives earlier at the destination register than at the source register. In this case, the launch flop gets less time

to launch the data, hence setup slack becomes critical, on the other hand, hold slack is relaxed. This is expressed by eq 1.2.

SetupSlack = Tcycle − Tsetup − Tskew − Tlaunch − Tcombo (1.2) where Tcycle is the clock period, Tsetup is for the capture flop, Tskew is the clock skew, Tlaunch is the max time taken in the launch path, Tcombo is the combinational delay in the data path, and Thold is the hold time of the launch flop.

7. **ECO:** ECO is an efficient method for fixing timing problems after the placement and routing [8], and STA stages are over. Logic is inserted into the netlist through ECO. One method include the gate-level netlist ECO, which is a cumbersome process involving repetitive iterations. Thus this needs to be automated, in a manner to reduce the modifications that are done in the design stages of early Physical Design. Related solutions are proposed in the coming chapters.

2.1.3 NEED FOR ECO GENERATION

According to the present technique, the ECOs must be implemented in the design flow utilizing a variety of tools that can manage several modes and corners at once. For the design teams to determine how to address the infractions, there are several conflicting reports. This technique involves many iterations, which could jeopardize the design. In such a situation, meeting production deadlines becomes a difficult objective. Effectively handling the violations through ECO is desperately needed to lower the probability of chip failure. Otherwise, the designers must deal with erratic iterations throughout the signoff ECO loop. The goal of this work is to address the typical flow's performance restrictions by putting forth a fresh alternative that improves ECO coverage while drastically cutting runtime.

2.1.4 PREVIOUS WORKS ON TIMING CLOSURE AND RESEARCH GA

Olivier Coudert highlights the issues with time closures in [7]. Signal integrity, clock and power/ground routing, signoff and capacity, and design variable dependencies are all included. In order to shorten turnaround time, it highlights the necessity of a tool that can offer a sign-off flow solution. There are several powerful tools available, such as the Integrated Circuit (IC) compiler, which can do timing-driven routing and placement. However, as noted in [7], [8], new timing violations occur after the placement and routing step when new logic is added to the design to satisfy the functionality need. The emergence of numerous scenarios that cause designers to either over- or under-constrain a particular design block—that is, to either overlook a set of constraints or to over-constrain by attempting to make it operable at higher frequencies than necessary for reuse in other

chips—is one of the primary causes of this phenomenon, as discussed in the latter paper. The relationship between different tools is still another problem.

As noted in [9], because of the design's intricacy, analysis is conducted for a limited number of corners, with the expectation that it will cover the design in all worst-case situations. The huge designs and numerous scenarios may make it challenging for the design teams to scale their scripts [8]. [9] Based on gate and connection delays, the study suggests an automated methodology for identifying the worst-case situations; however, this scenario extraction is only carried out for STA and not for the ECO stage. To enhance the flow analysis starting from generating RTL up to the final GDS, sufficient research has been conducted. A less studied area is post-routing, analysis, and fixing to attain timing accuracy. This results in design bottlenecks, which raise the number of ECO iterations significantly. [10] has made reference to this. This work aims to outline the difficulties and shortcomings related to crosstalk signoff analysis and hierarchical timing analysis.

The goal of [11] is to shorten the total STA runs' runtime. STA is carried out following each ECO step to guarantee that there are as few violations as possible because ECO generates propagation delays. The STA execution time rises in tandem with the design's increased complexity. Therefore, using complete chips to accomplish STA is an inefficient method. This study suggests an incremental STA algorithm that makes use of the data from formal verification, a step that determines the changes made following the ECO stage.

2.1.5 APPROACH AND RESEARCH METHODOLOGY

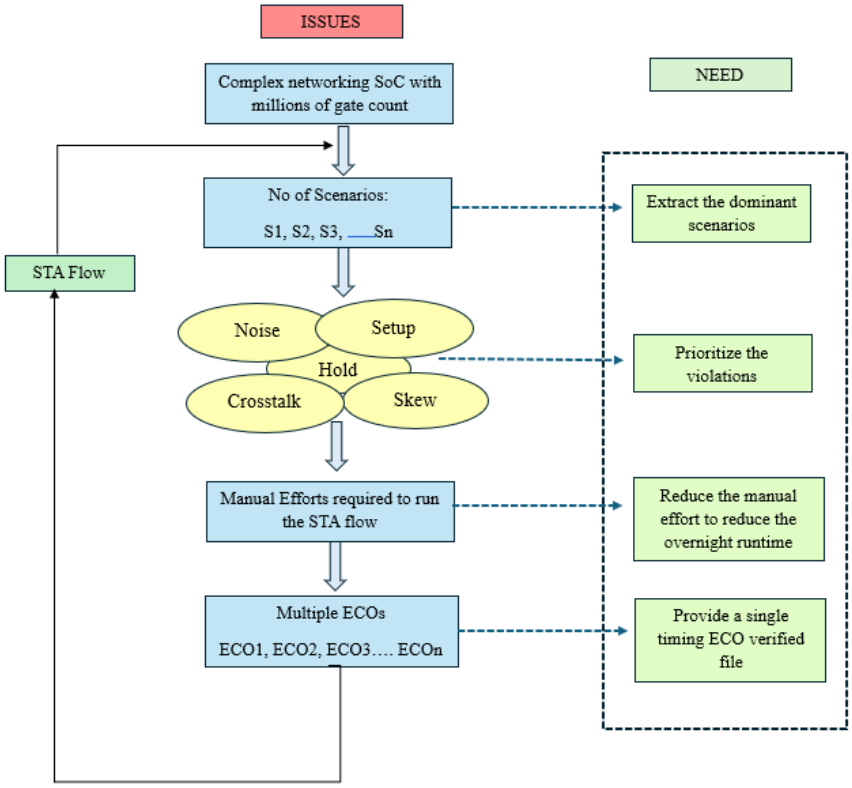


Figure 2.2: Current v/s Proposed Methodology

The inefficiencies of the current methodology are depicted in Figure 2.2, along with the corresponding preferred approach to address the current problems. There could be more than 200 possible outcomes for a sophisticated networking chip with millions of gates running at a 1.8nm technology node. Runs needing access to a similar amount of resources and licenses, which are already extremely constrained, increase when STA and ECO are performed on such a large number of situations. However, the amount of work required to manually analyze and choose the worst-case scenarios is unthinkable, which eventually results in only extreme scenarios being used for ECO. By analyzing and choosing the fewest worst-case situations, this approach aims to minimize the number of runs. In order to increase the runtime with the least amount of manual labor, this chapter presents a novel methodology that extracts the dominating situations, ranks the types of violations, and filters the violating endpoints at each level. It also discusses the importance of the suggested step in the process and outlines the specific problems with the present approach step-by-step. The subsequent chapters of this work explain this new methodology.

CHAPTER 3

PROPOSED METHODOLOGY

At advanced technological nodes, timing closure is a difficult and iterative process that calls for a blend of conventional and innovative approaches. Timing closure becomes more complicated as process nodes go smaller because of power, signal integrity, unpredictability, and silicon's physical design limitations. A multidisciplinary strategy involving enhanced STA, clock optimization, machine learning, and AI-driven tools is increasingly crucial to overcoming these obstacles and producing dependable and high-performance SoC designs. Thorough simulations are run for every process scenario in order to go beyond the existing experimentally oriented approach to ECO. Since this is a manual procedure, there is no systematic technique used in the analysis and correction of the breaches. Without understanding that it could exacerbate the issue and ultimately result in a longer runtime and the exploitation of expensive resources, every breach may be targeted. Second, all violated endpoints are the centre of the current attention. However, without taking into account every endpoint that violates, ECO x on modified endpoints alone can drastically lower the number of violations.
In order to optimize the runtime with the least amount of manual labor, this chapter examines how the dominating situations are extracted, ranks the types of violations, and modifies the violating endpoints at each level. It also discusses the importance of the suggested step in the process and outlines the specific problems with the present approach step-by-step.

Enhancements in Industrial Level Static Timing Analysis (STA)

Static Timing Analysis (STA) is still the primary tool for timing closure in both lower and advanced nodes. However, STA needs to adapt to handle the complexities introduced by advanced nodes.

i. **Advanced Timing Models**: At advanced nodes, standard delay models (e.g., linear delay models) are no longer sufficient. More sophisticated models such as **Non-Linear Delay Models**, **Statistical Static Timing Analysis (SSTA)**, and **Variation-Aware Timing Analysis** are employed to account for process variations, voltage fluctuations, and temperature changes.

ii. **Multi-Corner, Multi-Mode (MCMM) Analysis**: At advanced nodes, designs typically operate under various process, voltage, and temperature conditions [1] [3]. Therefore, STA

tools are set to run simulations under multiple corners and modes to validate timing across different operating conditions.

iii. **Incremental STA**: After each iteration in the design process, incremental STA helps speed up the timing closure process by analysing only the parts of the design that were modified since the last analysis.

iv. **Clock Mesh**: For highly advanced nodes (sub-5nm), Clock Mesh networks are becoming prevalent to reduce clock skew and improve timing across the chip. This approach ensures more uniform clock distribution, which can reduce timing errors due to skew and jitter.

v. **Skew-Reduction Techniques**: At advanced nodes, clock skew becomes a critical issue. Various techniques, such as Clock Tree Resynthesis**,** Clock Buffer Insertion, and Global Clock Optimization**,** are employed to reduce skew and ensure the timing is met across the chip.

vi. **Re-Timing**: Re-timing is the process of moving registers across the design to balance path delays and ensure timing closure. This is critical at advanced nodes where timing is often met by shifting registers rather than optimizing the logic itself.

vii. **Buffer Insertion**: Adding buffers to critical paths to reduce delay and help with timing closure. This technique is highly used in both lower and advanced nodes to meet timing constraints.

3.1 THE REAL-TIME FLOW EXECUTION

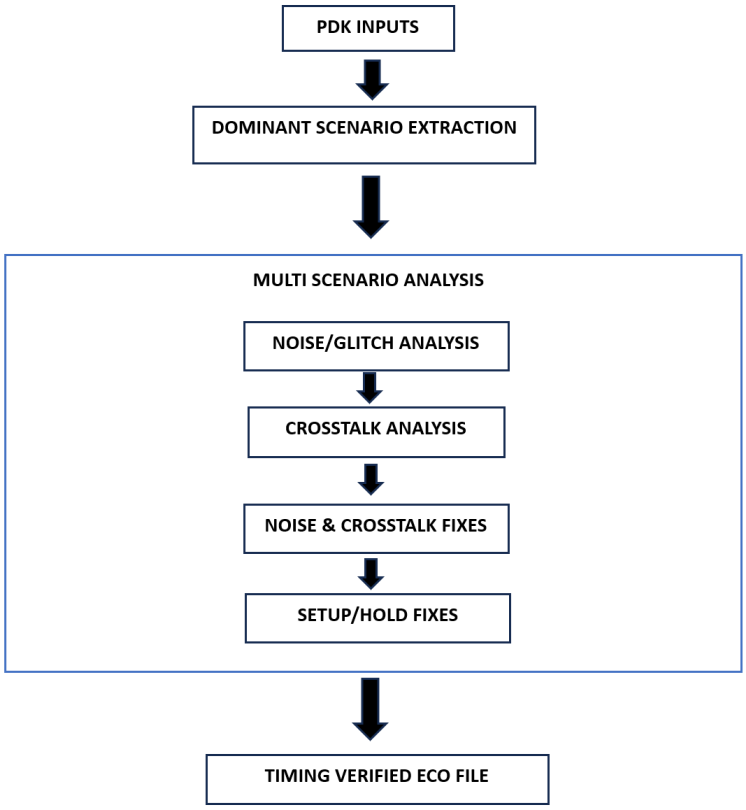


Figure 3.1: Real Time Flow Execution

Finding and extracting the dominating situations is the first stage in the technique, as shown in Figure 3.1. Primetime Distributed Multi-Scenario Analysis (DMSA) on the dominant scenarios is launched next [2]. The following objectives are to change the violating endpoints and prioritize the type of violations. The clock tree is the design's holy grail since it becomes frozen in the later stages. First, violations originating from the network must be addressed. Therefore, among other kinds of breaches, the examination of clock skew on the prevailing scenarios is regarded as the most important. Additionally, the endpoints are filtered out in this stage for the subsequent crosstalk analysis step.

3.2 INTEGRATION OF STEPS

3.2.1 Dominant Multi-Scenario Analysis

Timing analysis and ECO are crucial phases of the digital integrated circuit design, which involves a vast array of scenarios. Violation analysis is required in the worst-case scenarios, assuming that they cover all violations in the other scenarios, because it is not feasible to contemplate addressing all the violations in all the scenarios. This approach is gloomy, though, because it is hard to develop a solution for the worst-case scenarios, and the likelihood of violations in the scenarios is unpredictable. There is currently no systematic process for selecting the appropriate set of worst-case situations from among all scenarios.

The worst-case situations must also be carefully chosen because a path can be setup violating (max delay type) in one corner and hold violating (min delay type) in another. This work suggests an effective method for identifying the worst-case or dominating scenarios while taking into account the min and max delay types. On the basis of the worst delay kinds, dominant scenarios are typically identified. In other words, the scenario with the greatest number of violating endpoints is regarded as the worst-case scenario; alternative scenarios can be chosen. To be regarded as the worst-case scenario, a set of scenarios discovered by this process frequently overlooks other possibilities that might also be crucial. For instance, situations S1 and S2 can be regarded as a set of dominating scenarios based on the number of violating endpoints (Number of Violating Points (NVP)) in Table 3.1, which is the scenario with the greatest number of violating endpoints (having the worst violating slack). But scenario S3, which has the worst negative slack (WNS) score of all the scenarios, is not included in this.

3.2.2 Noise, Glitch and Crosstalk Analysis

Noise analysis is one of the most integral parts of Static Timing Analysis (STA) as the design complexity increases in advanced VLSI nodes. At smaller technology nodes, the impact of noise, including power grid noise, crosstalk, and EMI, can significantly affect timing performance. To achieve robust timing closure, it is essential to use noise-aware STA, integrating power and signal integrity analysis into the STA process. By combining advanced noise modelling with optimization techniques, we can ensure that their SoCs meet timing requirements and operate reliably under real-world conditions.

Noise refers to undesired or unintentional effects affecting the proper operation of the chip.

i. It is mainly caused by the capacitive coupling between neighboring signals on the die.

ii. Switching activity on a net causes unintentional effects on the coupled signals.

iii. The affected signal is called the **Victim**, and the affecting signals are termed as **Aggressors**.

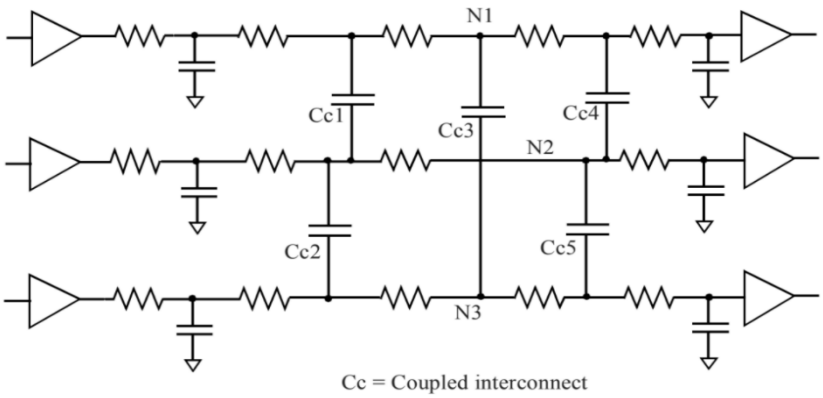


Figure 3.2: Connection of Buffers between interconnects on a chip

Main Noise directions :

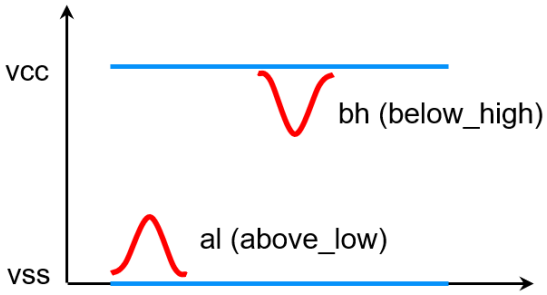


Figure 3.3: Noise Directions: Above low and below high

Why Noise Analysis?

Following are the reasons why the noise plays an important role in Modern design:

i. **Increasing number of metal layers:** In modern VLSI chip designs, the number of metal layers has significantly increased as technology advances, especially with the scaling of nodes to smaller process technologies (e.g., 7nm, 5nm, 3nm, and below). This increase in the number of metal layers is driven by several factors, including the need for higher-density interconnects, improved signal integrity, and the ability to support advanced functionality at ever-smaller dimensions. As the routing density increases significantly, the distance between the interconnects increases, thereby increasing the chances of noise and crosstalk among closer layers.

ii. **Higher routing density:** Higher routing density typically occurs in advanced technology nodes (such as 7nm, 5nm, or smaller) due to the increasing number of transistors, higher chip complexity, and the need for more interconnects within a smaller chip area. More power and ground rails are needed to supply the large number of transistors and circuits. Power noise also increases due to the combined effects of dynamic switching of transistors (which cause current spikes) and resistance in the power grid. This can cause fluctuations in supply voltage, which can lead to functional errors or impact timing, especially if the noise exceeds the power delivery tolerance of certain circuits.

iii. **Lower supply voltage:** In modern VLSI circuits, low supply voltage has become a common practice, particularly in advanced technology nodes, as part of efforts to reduce power consumption and improve energy efficiency. While lowering the supply voltage helps to meet power budgets, it introduces several challenges related to noise. These challenges are primarily due to the reduced noise margins, increased susceptibility to noise, and tighter tolerance for voltage fluctuations.

iv. **High-frequency design:** High-frequency operation in VLSI circuits introduces significant noise-related challenges that impact signal integrity, power delivery, and overall performance. Crosstalk, electromagnetic interference (EMI), signal reflection, power noise, clock skew, and thermal effects are some of the key issues that need to be carefully managed in high-frequency designs. As the operating frequency increases, the circuit becomes more sensitive to these noise sources, which can lead to timing violations, data corruption, and functional failures.

3.2.2.1 Noise Delay

In this analysis, the victim is also switching, and due to coupling with aggressors timing delay is changed. The base delay calculation assumes that the driving cell provides all the necessary charge for the total capacitance of a net, Ctotal (= Cground + Cc). Depending upon the switching direction of the aggressor, the crosstalk delay effect can be positive (slow down the victim transition) or negative (speed up the victim transition).

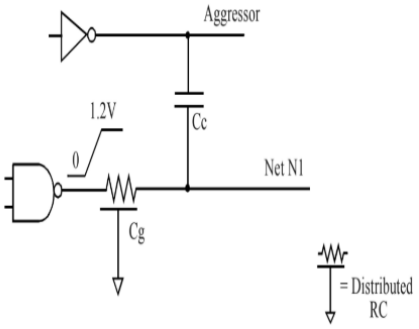


Figure 3.4: Aggressor and Victim Nets with Coupling Capacitance

Negative Crosstalk Delay:

- Aggressor switching in the same direction.
- The driving cell provides the charge only for Cg to be charged to Vdd.

Total charge required = Cg * Vdd

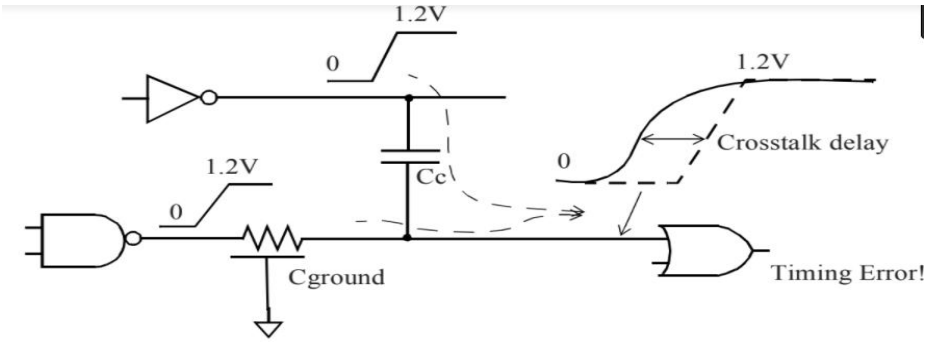


Figure 3.5: Aggressor and Victim switching in same direction

Positive Crosstalk Delay:

- Aggressor switching in opposite direction.
- Driving cell provides the charge for Cg to be charged to Vdd and for Cc to be charged from –Vdd to +Vdd.

Total charge required = (Cg + 2*Cc) * Vdd

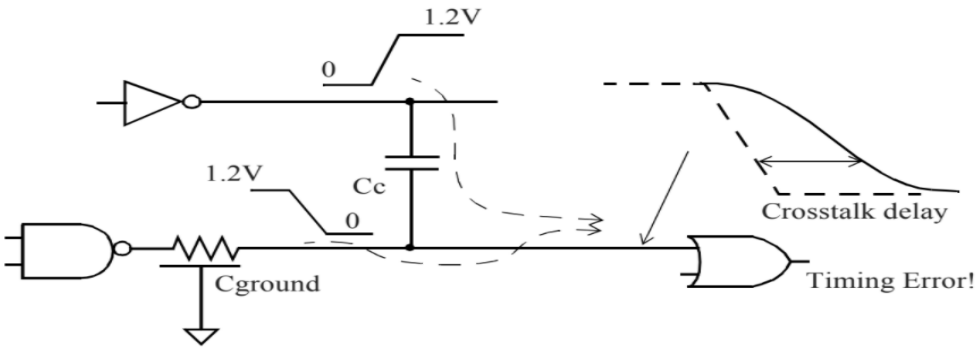


Figure 3.6: Aggressor and Victim switching in opposite direction

3.2.2.2 Noise Glitch

Noise is any deviation in the voltage level from the nominal supply or ground rails when the node should represent a stable logic '0' or '1' value. The deviation (from the expected voltage) is also referred to as **'Glitch'**. Noise may be a cause for a functional failure. The glitch magnitude may be large enough to be seen as a different logic value by the fanout cells (e.g. a victim at logic-0 may appear as logic-1 for the fanout cells). This is especially critical for the sequential cells (flip-flops, latches) or memories, where a glitch on the clock or asynchronous set/reset can be catastrophic to the functionality of the design.

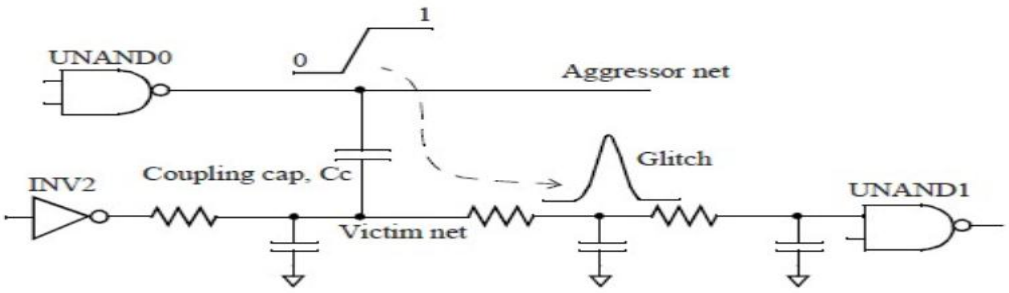


Figure 3.7: Occurrence of Glitch

Type Of Glitches

Noise can cause following glitches in the Victim net.

- **Rise glitch:** Raising the aggressor net induces a rise glitch on a steady low victim.
- **Fall glitch:** The falling aggressor net induces a fall glitch on a steady high victim.
- **Overshoot glitch:** Raising the aggressor net induces an overshoot glitch on a steady high. This takes the victim's net voltage above its steady high value.
- **Undershoot glitch:** The falling aggressor net induces an undershoot glitch on a steady low. This takes the victim's net voltage below its steady low value.

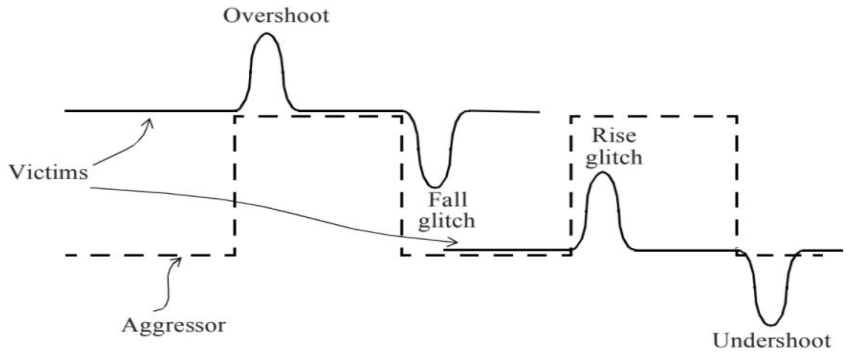


Figure 3.8: Glitches: Diagrammatic Representation

3.2.2.3 Noise Violations

i. Double Switching Violations

It happens when a weakly transitioning victim is attacked by a strong aggressor

i. Usually, the victim also has a bad slope (max trans) problem
ii. Critical for clock signals as could cause false capture in sequential elements

iii. Buffering, shortening routes, or improving slopes will solve these violations

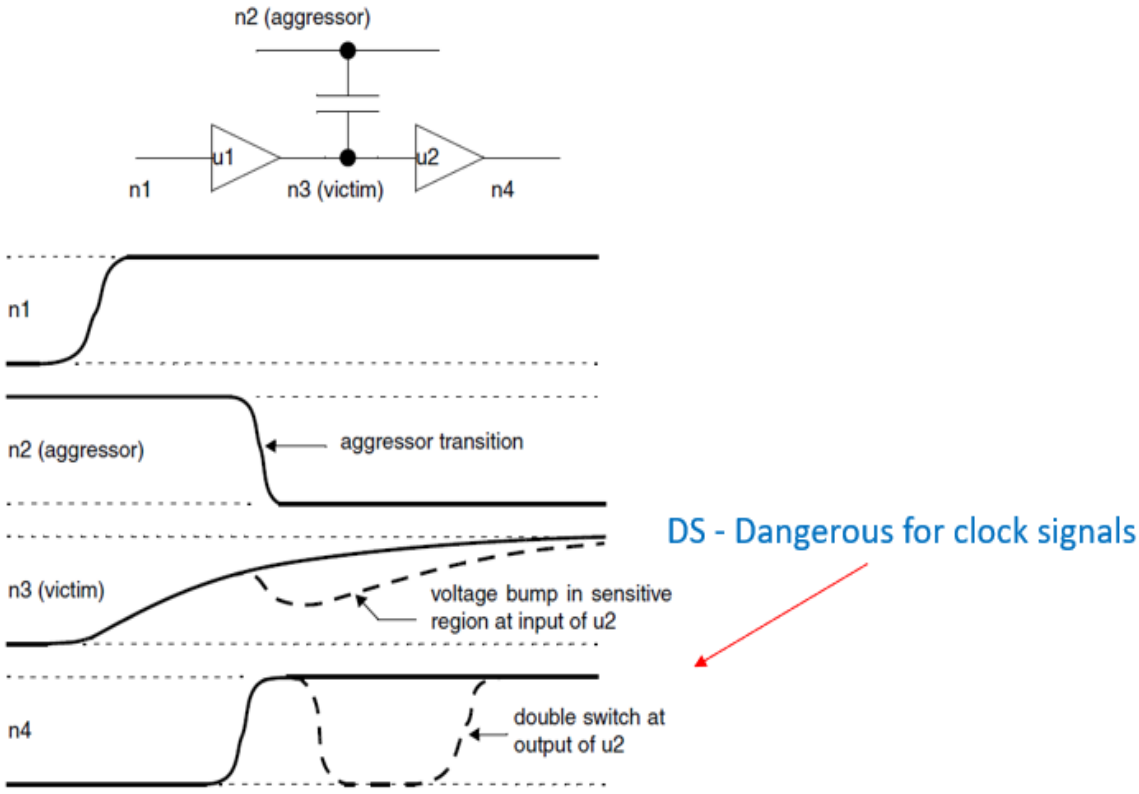


Figure 3.9: Double Switching Noise Violation

ii Maximum Power Plane Violations

A maximum power plane noise violation refers to a situation where the voltage fluctuations or noise in the power supply exceed the maximum allowable limits for the chip to operate correctly. These violations typically occur when:

i. The supply voltage deviates beyond allowable limits (e.g., more than a few percent above or below the nominal voltage).
ii. For instance, a deviation of more than 5-10% from the nominal supply voltage could cause timing errors, data corruption, or logic failures in sensitive parts of the circuit.
iii. Timing violations due to noise-induced delays: If the power noise affects the timing characteristics of the circuit (e.g., gate delays, clock synchronization), it could lead to setup or hold violations in sequential circuits (like flip-flops or registers), causing incorrect data to be latched or incorrect computation.

iii Normal Violations

With regards to Figure 3.10 and Table 3.1, following points are considered:

- **Internal violation**
- Violation that happened internally in a partition
- It is Partition 1 responsibility to ensure Violation A is fix.
- **Between partitions**
- Violation that involves 2 different partitions.
- It is Partition 2 responsibility to ensure Violation B is fixed & should communicate with Partition 1 if the fix needs to be implemented in Partition 1.
- **Interface violation**
- Violation that happened on the port of the top block
- It is Partition 3 responsibility to fix violations C1 & C2.

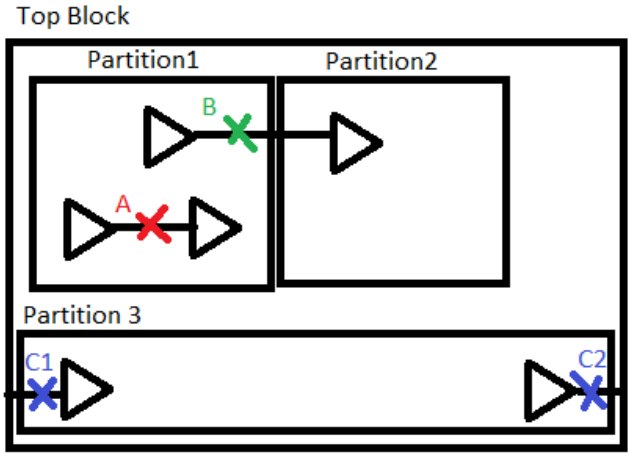


Figure 3.10: Normal Violations

Partition	Internal	Between Partitions	Interface
Partition1	A	B	
Partition2		B	
Partition3			C1,C2

Table 3.1: Violation Representation in partitions

3.2.3 Crosstalk Analysis

Crosstalk refers to the phenomenon where a signal on one net (the aggressor) induces an unintended voltage fluctuation or noise on a neighbouring net (the victim). This occurs because of the capacitive or inductive coupling between the signal traces. It becomes more problematic as the chip operates at higher speeds, with smaller dimensions (leading to more densely packed routing), and lower supply voltages.

Types of Crosstalk Effects:

i. **Noise Induced on Victim Net:** The aggressor net causes a voltage spike or dip on the victim net due to capacitive or inductive coupling. This voltage fluctuation can distort the logic signal on the victim net, potentially causing timing violations or signal integrity issues.

ii. **Glitches:** Crosstalk can induce glitches in the victim net, where unwanted transitions (logic 0 to 1 or vice versa) occur. These glitches might cause a flip-flop or latch to sample a value or lead to incorrect logical states.

iii. **Timing Violations:** Crosstalk can increase the rise or fall times of the signal on the victim net or delay the signal propagation, causing setup or hold time violations. This leads to failure to meet timing constraints, particularly in high-speed circuits.

iv. **Signal Integrity:** In high-speed circuits, crosstalk can compromise the overall signal integrity, affecting the quality of data transfer and leading to errors or incorrect data being transmitted.

The process involves analysing the physical layout of signal nets, considering their proximity, and simulating the possible crosstalk effects. Here's how the crosstalk analysis is typically carried out:

1. **Signal Integrity Simulation**:

i. **Simulating Crosstalk Effects**: Tools like SPICE (Simulation Program with Integrated Circuit Emphasis) are used to simulate the crosstalk effects. These simulations consider the parasitic capacitances and inductances in the layout, and how signals on nearby lines may couple.

ii. **RC Extraction**: The parasitic capacitances and resistances of the signal nets are extracted from the layout. This information is then used to simulate the coupling between aggressor and victim nets.

2. **Crosstalk Coupling Calculation**:

i. Tools analyze the coupling capacitance between adjacent signal traces and calculate how much of the aggressor's signal is induced onto the victim net. The amount of coupling depends on factors like the distance between the wires, the width of the cables, the thickness of the dielectric material, and the surrounding environment.

3. **Crosstalk-Related Noise Analysis**:

i. The induced noise on the victim signal can be analyzed to determine if it could lead to a logic error. The analysis will check if the induced noise is large enough to cause **voltage threshold violations**, where the victim signal might flip unexpectedly.

4. **Static and Dynamic Crosstalk Analysis**:

i. **Static Crosstalk Analysis**: This involves evaluating the potential for crosstalk based on the static configuration of the circuit (such as the layout and wire spacing) without considering dynamic switching behaviour.

ii. **Dynamic Crosstalk Analysis**: This considers the actual signal transitions and switching activities during operation, where switching noise on aggressor nets can be more problematic.

Fixing the Crosstalk Violations:

To remove any timing violations that may arise, the list of crosstalk-affected nets that is collected following crosstalk analysis must go through a crosstalk fixing process. On this list, an internal loop is started. The most-affected crosstalk net is selected from the list, depending upon the delta. The victim driver's drive strength must be increased to lessen or completely eradicate crosstalk on a network. Because the aggressor nets' drivers are more likely to cause timing violations on non-critical paths, they are not shrunk. Therefore, working on the victim net is preferable to lessen signal integrity issues. One of the three crosstalk x ECO approaches is employed, depending on the degree of crosstalk effect on this victim segment. The three methods for optimizing delays are:

i. **VT Swapping:** The transistor switching speed is determined by the threshold voltage, which establishes the minimum voltage needed to charge the capacitor. The capacitor takes a long time to charge and reach the desired level, which slows down its speed, and the higher the threshold voltage is, the thicker the oxide is. Therefore, the driver cell's gate delay or cell delay must be decreased to lower the delta delay on the net. This is

accomplished by replacing the cell with an equivalent Low Threshold Voltage (LVT) cell available in the libraries. By switching to an LVT cell, which has a low oxide thickness, the cell is less likely to experience the effect of the aggressor nets transitioning on itself because it takes less time to flip its output when the input arrives, boosting its speed [18], [19], and [20] as indicated. Since changes can be made without requiring any modifications during placement and routing, VT swapping is the recommended approach to be taken into consideration for fixing. However, lowering the threshold voltage puts the net at risk for issues connected to leakage. The power rises as the transient current rises at low threshold voltages.

ii. **Driver Upsizing:** Upsizing the victim driver cell [21], [22], and [23] to increase the victim cell's output drive is another method of lowering the crosstalk latency. When the cell is enlarged, the transistor's width increases, which consequently lowers the cell's resistance and raises the (dis)charging current. The Elmore Delay calculation of a cell [22], [23], shows that a cell can be replaced by its equivalent RC network, and its delay is given by the following equation:

$$\tau = 0.69 R_{int} * C_{eff}$$

where T is the delay of the cell, Rint total internal resistance of the cell, Ceff is the effective internal and load capacitance, and S is the scaling factor.

$$Inmos_{sat} = \mu * C_{ox} * W * (V_{gs} - Vt)^2 * (1 + \lambda * V_{ds})/L$$

As can be seen from eq. 3.4, a saturation current equation for nmos (Inmossat), where W is the transistor width, L is the channel length, and Cox is the oxide capacitance, increasing the transistor's width, or scaling it when the output load is fixed, lowers the resistance and raises the drive current. Therefore, increasing the cell's size enhances the drive strength, decreases the propagation delay time, and improves the output transition. But according to eq. 3.5 [23], expanding the cell's width also raises the gate and parasitic capacitances, which results in a higher load on the cell's fanin. Upsizing the cell is also favored over VT Swapping because it necessitates rerouting.

iii. **Buffer Insertion:** This method works well for timing optimization driven by interconnects. The square of the connector length directly correlates with the interconnect delay. Interconnect delay becomes linearly proportionate to wire length when buffers are inserted in smaller segments [24]. By pushing net capacitances through smaller resistances, it improves the path's slack and lowers the nets' Elmore Delay. However, it significantly expands the region and necessitates changes during the re-routing phase. This increases the path's delay, which lessens the impact of crosstalk from nearby effective attackers and, as a result, eliminates hold violations from this path by increasing its slack.

3.3 HOW TO ENSURE THE DESIGN ROBUSTNESS

After all the noise/glitch and crosstalk analysis, some additional rulesets are being check at the full chip level, to ensure the design robustness. This is done with the help of an internal tool (called Intel Caliber) specifically developed for this purpose. It has certain sets of rules that are defined according to various buckets like quality, timing, DFT, library collateral checks, clocking accuracy checks and the structural development flow quality etc. The running of this tool requires an already saved Prime Time saved session. Once, these rulesets are free of any violations, the chip design is ready for the final handoff to the methodology team to verify the DRCV checks and finally ready for tape in.

When something is violating a check there are two options:

- Fix – waterfall the issue to the team/person who is impacted by this rule.
- Waive – send to Quality Rule Owner (QRO) with proof that it is not a real problem.

Figure 3.11 below gives all the various buckets, in which the rulesets are divided. Table 3.2 gives a brief description of each rule.

Bucketizing of the Intel Caliber rulesets:

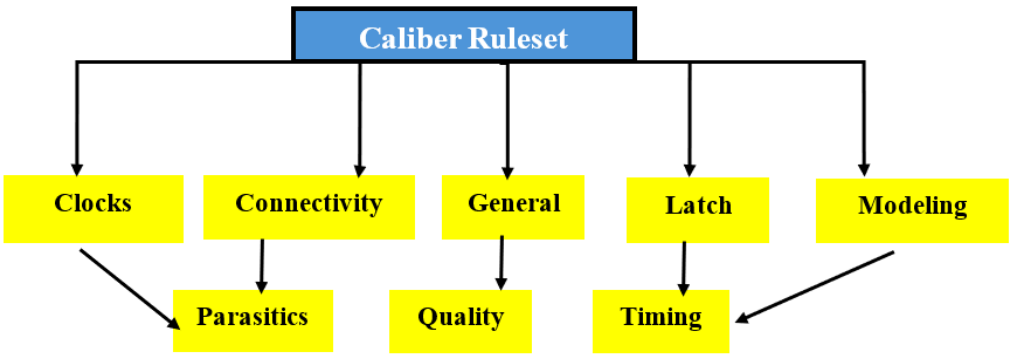


Figure 3.11: Caliber Rules Buckets

Category	Rule	Description
Quality	BadExtractionQuality	Reports violations for opens/shorts nets in the design
	BigWireDelay	Wires with delay greater than $caliber_wire_delay_limit ps In case slack ismissing. This can help to point to issues being seen in either SlowSlope or MaxCap rules.
	IllegalCells	Flags cells that are not allowed to be used in the design.
	SlowSlope	Flags data signals with slopes worse than project defined limits
Clock	LogicOnClockPath	The rule checks to see if a clock attr is present on the output pins of the cells. If such an attr is present on logic cells, the rule flags it as a violation.
	ClockMaxTrans	Checks clock pins with slope > 100 ps Slow slopes can lead to higher variation, which is especially dangerous on clocks

Table 3.2: Description of Caliber Rules

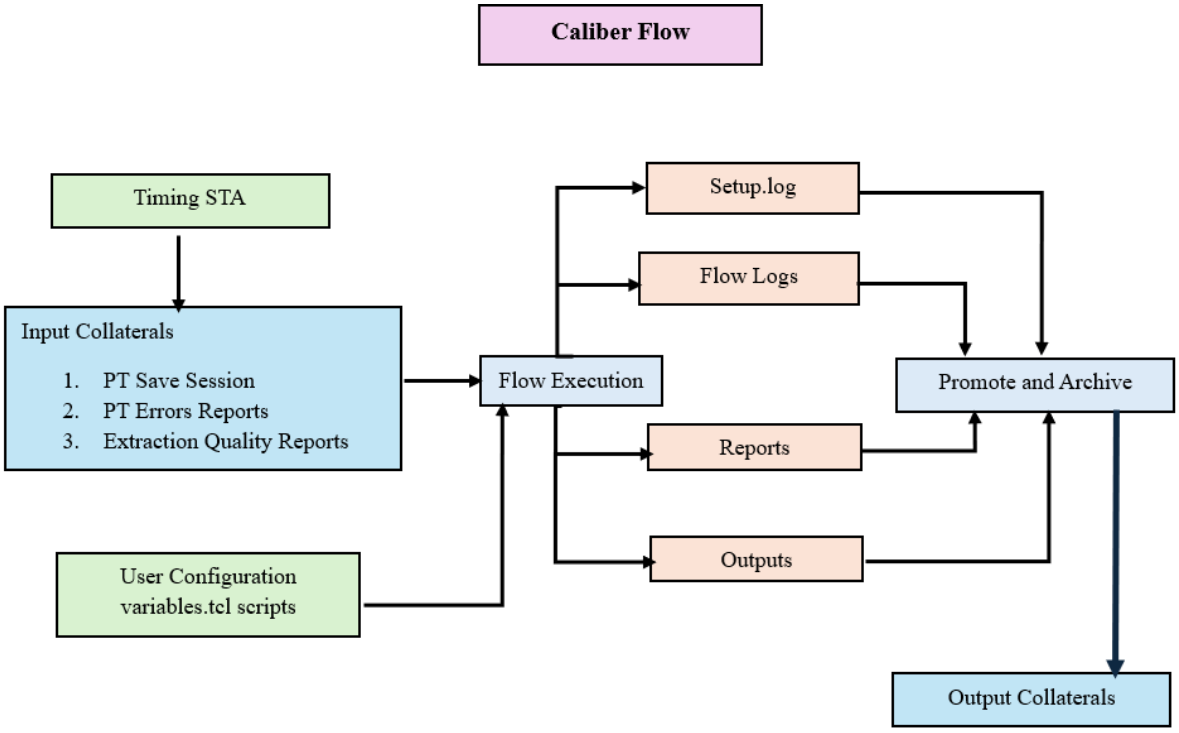


Figure 3.12: Intel Caliber Tool Flow

CHAPTER 4

HOLD TIME FIXES THROUGH ENGINEERING CHANGE ORDER

This chapter explores one of the ways to mitigate the hold time violations occurring at the full chip level, i.e. through Engineering Change Order (ECO). While releasing the timing models, it is observed that there are there are various failing end points with an internal slack greater than -15 ps. Now the objective is to fix these violations and get the slack into permissible limits. The most suitable way to fix these violations at this last stage of signoff is through the implementation of an ECO, which will add a buffer or a spare cell (as per the requirement) which in turn will also cater to the ideal slack requirements.

4.1 WHAT IS ECO AND ITS REQUIREMENT DURING SIGNOFF

To directly correct or alter the gate level version of a design at the SoC level, an Engineering Change Order is necessary. This change may be made to optimize the design, address a new customer demand, or address violations discovered at the pre-synthesis stage [6]. By doing this, the expensive and time-consuming process of fully re-implementing the design is avoided, which may involve technology mapping, logic synthesis, location, route, and layout extraction, as well as timing signoff. The following phases can be distinguished within each ECO Cycle:

1. Investigating and understanding the problem statement using the latest database and PDK.
2. Generating the ECO to address the problem.
3. Implementing the ECO on the studied database.
4. Saving the database for performing further steps and moving towards sign-off.

An ECO file consists of a list of changes required in the form of PnR commands. Based on the requirement, the ECO file can be created by the tools themselves or can be customised according to the complexity of the problem. Once the ECO file is ready, it is loaded into the database and sourced in the tool environment. All changes are implemented when the sourcing of the file is over. Checks are again performed to ensure that the existing problem is fixed, and no new violations have popped up.

In general, there are several types of ECOs that can be implemented during the Physical Design Flow. Some of them are discussed as under:

1. **All Layers ECO:** The design change is implemented using all the metal layers in the design. It is time and cost saving. Implemented whenever there is a change in the hard

macros or the change is required in more than 100 cells, since it is not feasible to accommodate such large changes in only a few layers.

2. **Metal Only Timing ECO:** Sometimes, it may not be possible that changes are done in all the layers. So, it is required to be completed with changes only in minimal number of metal layers. At the deep sub-micron level, the design is re-opened for the purpose of ECO implementation. So, an adequate number of spare cells are sprinkled during the implementation all over the design to be used later. These cells are spread uniformly over the design. Whenever the need for an ECO arises, the cells to be implemented can be mapped into the existing spare cells. Hence, there is no need to change the base layers in such a case. Only the connections need to be updated which can be done by changing the metal layers only. Hence, the base layer cost is saved. Violations like setup, hold, max_trans, max_cap, max_fanout, min_pulse_width, min_period are usually fixed with the help of timing ECOs. In this thesis, we have implemented the metal only ECO, to fix the hold violations.
3. **Power ECO:** These changes focus on reducing power consumption, either dynamically (during operation) or statically (due to leakage). Techniques like resizing transistors, adjusting the supply voltage, or modifying clock gating techniques are used. There are different ways to achieve power ECOs such as dynamic power management, voltage scaling, power gating, clock gating, activity reduction, leakage and power reduction, etc.
4. **Routing ECO:** Routing ECOs are adjustments to the routing of metal layers to fix issues like signal congestion, wiring errors, or to optimize the overall layout. This may involve rerouting critical signals or adding vias. Routing eco typically comes into play during the post-route stage when designers need to refine or adjust the routing to meet specific design constraints, such as signal integrity, power, performance, or manufacturability. Techniques to implement Routing ECO includes re-routing, buffer insertion, wire resizing and via changes.
5. **Functional ECO:** It refers to changes made to the functional aspects of a design after the initial design has been completed and possibly even during the manufacturing process. These changes typically occur to fix issues related to the functionality of the circuit, such as incorrect logic, performance issues, or other defects that affect how the chip behaves. These ECOs are often applied to correct mistakes that were made during the design phase. These mistakes might include incorrect logic implementation, errors in the functional behaviour of certain blocks, or unintended interactions between different parts of the design. Functional ECOs are often used to fix bugs in the design. These could be discovered during simulation, verification, or after prototyping the chip. For example, a critical failure in a specific operation might require changes to the circuit's logic or structure to ensure it behaves correctly under all conditions.

4.2 ECO IMPLEMENTATION FLOW

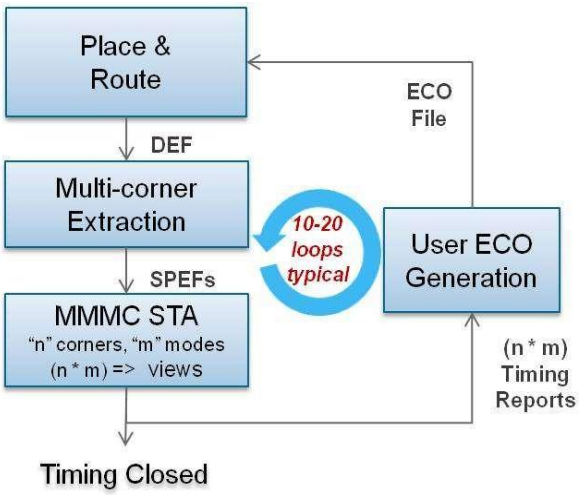


Figure 4.1: General Flow for a Timing ECO

Figure 4.1 shows the typical flow that is followed while implementing a Timing ECO. 20-30 iterations of this flow are quite common in a time span of 4-5 weeks. Multi-voltage, Multi corner and multi-mode scenarios need more time for closure. The details of timing ECO fixes have been summarised in the table 4.1 as shown below:

Setup Fixes	Hold Fixes	Max_trans Fixes
✓ Driver Upsizing ✓ Vt-swap ✓ Swap to lower channel length devices ✓ On-route buffer-insertion ✓ Re-placement (cluster) ✓ Net layer promotion (using route guides) ✓ Nets re-routing (shorter routes) ✓ Redundant inverter pair/buffer removal ✓ Flop vt-swap (lower setup requirement) ✓ Shield critical nets ✓ load splitting	✓ Downsizing ✓ Vt-swap ✓ On-route delay cell insertion ✓ Endpoint buffering ✓ Net layer demotion (route guides) ✓ Flop Vt-swap (lower hold requirement) ✓ Clock adjustment ✓ Dummy load insertion ✓ Net detours	✓ Upsizing driver ✓ Vt-swap ✓ On-route buffer-insertion ✓ Load splitting

Table 4.1: Different ways to fix timing violations through Timing ECO

Table 4.1 contains the various techniques to implement the timing ECO, specifically to fix the setup, hold and maximum transition violations. Any of the methods can be used depending on the design requirements. In this thesis, we have used the **on-route cell delay insertion** and **net detouring** methods to fix the hold violation in our design at the full chip level.

A detailed flowchart for the on-route cell delay (i.e. buffer) insertion is shown in the figure 4.3 below:

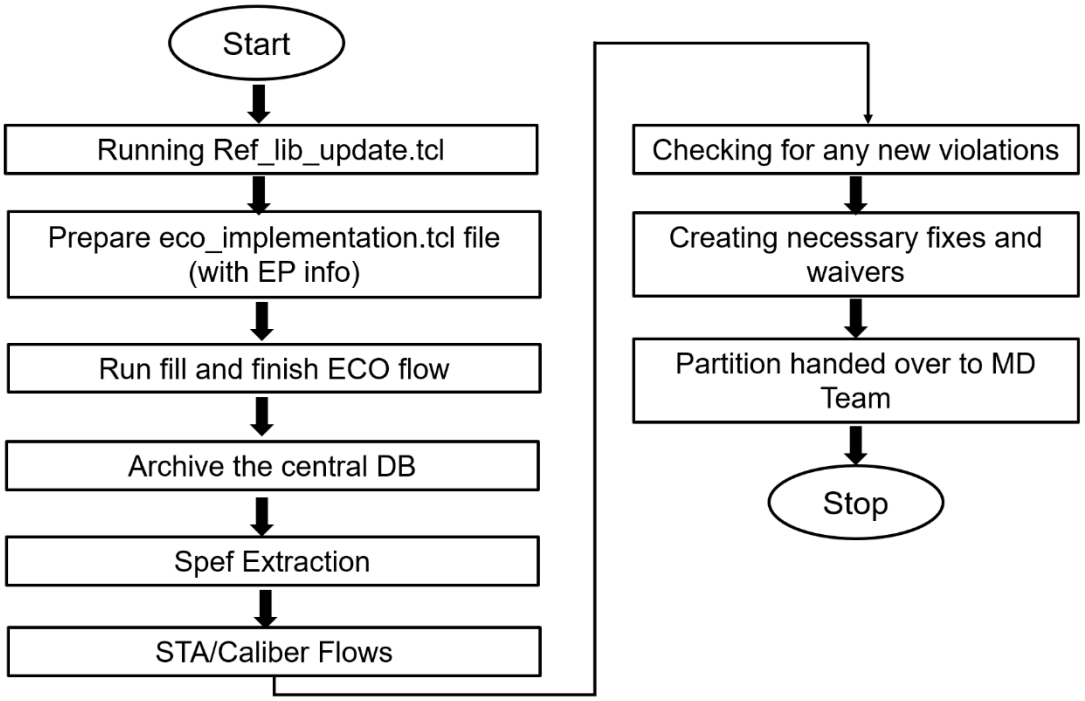


Figure 4.2: ECO Flow for On-Route Buffer Insertion

1. The ref_lib_update.tcl file opens the library NDM (New Data Model) of the partition on which we are working, and sets the location of the output files to be saved, finally saving the library (save_lib) and then saving the block. These commands are executed in the Fusion Complier tool by Synopsys.
2. Next step is to prepare the eco_implementation.tcl, which contains the start point and end point information of the partition under test so that the buffer can be inserted accordingly.
3. Further, fill and finish steps are executed to complete the generation of the ECO. The **Fill** step is typically performed to satisfy physical design requirements, especially when there are unconnected areas or insufficient material in the layout. The main goal is to ensure that the design adheres to the foundry's manufacturing rules and to fill any gaps or voids left after modifications in the design. The **Finish** step focuses on finalizing the design after the ECO has been implemented. This step involves checking that all the changes made during

the ECO process are correct and that the design is ready for final verification and manufacturing. The finish step ensures that all aspects of the design (functional, physical, electrical) meet the requirements before tape-out.

4. After fill and finish, the saved block and is then archived to a central database repository, which makes it accessible to all the involved stakeholders.
5. Parasitic extraction and STA timing checks are then performed, to ensure that the current problem is fixed, and also there are no new violations arising due to the new buffer that has been inserted.
6. If there are some new violations popping up, accordingly actions are taken to fix them manually, and this process takes certain iterations until all the violations are clean.
7. After all steps have been performed successfully, the partitions are handed over to the Methodology Team for the final Tape-In.

4.3 DESCRIPTION OF THE ACTUAL PROBLEM STATEMENT

This sub-chapter focuses on what the actual problem statement is and what are the slack numbers, that are being violated.

Start Point Block	End Point Block	Worst Negative Slack (in ps)	WNS Corner
BLOCK 1	BLOCK 1	-6.07	CORNER 1
BLOCK 2	BLOCK 2	-8.33	CORNER 2

Table 4.2: Start and End Point Blocks with WNS and WNS Corners

Partition	Unique Failing End Points (min)	Unique Failing End Points (max)
BLOCK 1	**1**	**2**
BLOCK 2	**1**	**475**

Table 4.3: Failing End Point Count for Block 1 and Block 2

Table 4.2 above tells the start point and end point blocks with the worst negative slack (WNS) in pico seconds, along with the WNS corners. The objective here is to make the slack positive with the help of the buffer that is being inserted.

Table 4.3 tells that how many unique end points are failing to meet the slack requirements for both setup (max) and hold (min) time. Our focus here is to fix both the failing end points of Block 1 and Block 2.

Since Block 2 has a lesser slack as compared to Block1, and more congestion (upon looking into the design at the routing level), the hold violation is fixed with the help of a method called **Net Detouring**. This method is used to re route a signal path, around the obstacles, congestions or timing issues in the partition layout. It involves adjusting the path of a signal while respecting design constraints such as wire width, spacing, and timing. Here, the signal's timing is critical, so detouring involves optimizing the path to reduce the delay, by taking a different route or adjusting the routing to take advantage of faster paths (e.g., using metal layers that are less congested). Switching to a different routing layer, connecting layers with a via, and then continuing the path are all parts of the detour.

The tools used to perform timing ECOs include:

- In-house MCMM (Multi Corner Multi Mode) based TCL scripts
- Tweaker (Synopsys)
- fix_eco_timing/fix_eco_leakage/fix_eco_drc (PTECO-Synopsys)
- Tempus (Cadence)

The results of this ECO implementation have been discussed in Chapter 6, later.

CHAPTER 5

XEON PROCESSORS: DESIGN AND STA FLOW

The design insights, some of the more sophisticated chiplet packaging methods, and most importantly, the STA flow incorporated into the Xeon devices' backend design will all be briefly covered in this chapter. These flows are made to comply with the most recent technology, which is being implemented at the 1.8 nm and 3 nm process nodes. Flexible CPU systems that can expand compute performance through enhanced core performance or larger core density are better suited for a range of contemporary computing applications. Furthermore, the design and architecture of data center servers are increasingly focusing on power efficiency. Compared to a single lithography reticle field, a many-core CPU implementation today requires higher silicon area (span) [13]. In order to maximize die-to-die communication capacity while avoiding any latency penalty, this in turn calls for a disaggregated design and improved packaging technologies. The components of the new technology are as follows:

1. RibbonFET – By encircling the channel with a transistor gate that resembles thin silicon ribbons, the RibbonFET transistor enhances the electrostatics of the FinFET. The main computing CPU chiplets for the Xeon processors will be constructed using Intel's second generation RibbonFET Technology (Intel 18A). Additionally, RibbonFETs are anticipated to be more efficient than FinFETs.

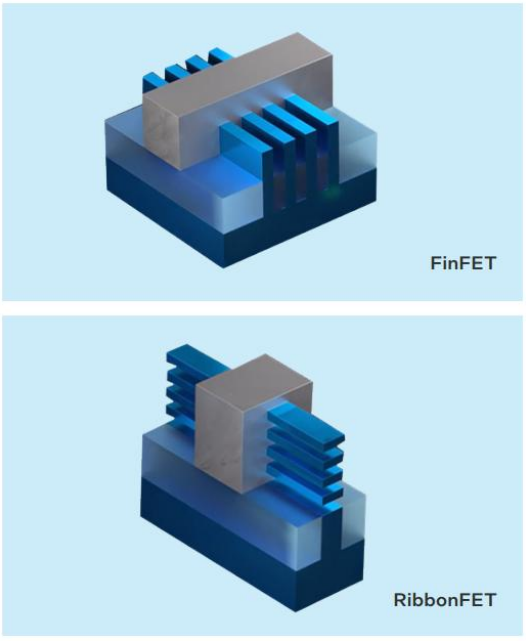


Figure 5.1: RibbonFETS are more power efficient than FinFETS

2. **PowerVia –** Historically, the substrate beneath the transistors has mainly served as a structural support layer, while the metal wires required to connect them have always been on top of the transistor layer (front-side interconnects). However, Intel is currently altering this paradigm by introducing back-side interconnects, or metal interconnects, beneath the transistor layer. Wires were used to transmit electrical signals between transistors and to supply power to the transistors in the previous paradigm, which shared the frontside interconnect architecture. Signal routing and power supply are separated for the first time with Intel 20A's PowerVia technology. This allows a new back-side interconnect design to be separately optimized for power delivery while the front-side interconnect architecture can be optimized for signal routing. Better routability (saving chip size and power) and reduced voltage droop (allowing for better performance at a given supply voltage) are made possible by this decoupling.

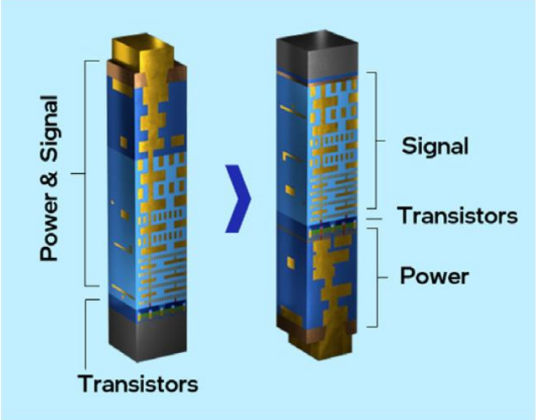


Figure 5.2: PowerVia enables up to 90% chip area utilization along with a 30% reduction in voltage droop and a 6% performance improvement.

3. Foveros Direct 3D – With the use of Intel's Foveros Direct 3D technology, complex system modules can be created by directly attaching one or more chiplets to an active base tile. "Direct" attach is accomplished by either directly attaching complete wafers stacked on top of one another or by thermocompressing copper vias on individual chiplets to those on a wafer. More product architecture flexibility is provided by the attachment, which can be "Face-to-Face" or "Face-to-Back" and contain chips or wafers from many source foundries.

4. Embedded Multi-die Interconnect bridge (EMIB) 3.5D – An established Intel technology called Embedded Multi-die Integrated Bridge (EMIB) allows high bandwidth communication between several big chiplets without the need for a silicon interposer. As previously mentioned, EMIB technology can also be utilized to connect several compute units built with Foveros Direct 3D technology. The development of adaptable,

heterogeneous computing systems is made possible by EMIB 3.5D, a package that combines EMIB with Foveros.

5.1 MULTIPLE PROCESS NODES IN A SINGLE PACKAGE

This Xeon server chip (codenamed as Clear Water Forest) will be the first Xeon design constructed with 3D/Foveros technology [13]. This chip is bifurcated into multiple dies, which are operating on different process nodes. The dies are named as Top Dir, Base Die, and the connection between the two, i.e. Cross Die. From a signoff point of view, timing convergence is done on all the three. The top die/base die interface is through the FDI (Foveros Die Interface). There are many clocks going from base die to top die, with some having the POD (Point of Divergence) on the base die, having multiple clock entry points into the top die. Timing Models are released for all the three dies, and then analysis is done based on the slack values obtained. These timing models are iterative in nature, and new models keep releasing until all the violations are settled and slack requirements are met.

This sub-chapter focuses on the Cross Die Timing Model. The multi-tech environment can be understood as the base die on 3nm process node and the top die on a 1.8nm process node. Now in order to load the correct hyperscale model in the PrimeTime environment, the cross-die level scenario had to be mapped to the corresponding individual base die and top die scenarios using a particular set of variables, defined in TCL scripts. A total of 23 PVT corners have been developed to analyse the cross-die performance. Since these scenarios are not analysed in the individual dies, mapping of these cross-skew scenarios will be done with the corresponding nominal scenarios of individual dies.

The STA flow for this cross-die timing convergence is discussed in chapter 5.2 below.

5.2 UNDERSTANDING THE STA FLOW

First, let us understand, the basic Physical Design flow, and how static timing analysis is a part of it. Figure 5.3 below, shows the generalised physical design flow.

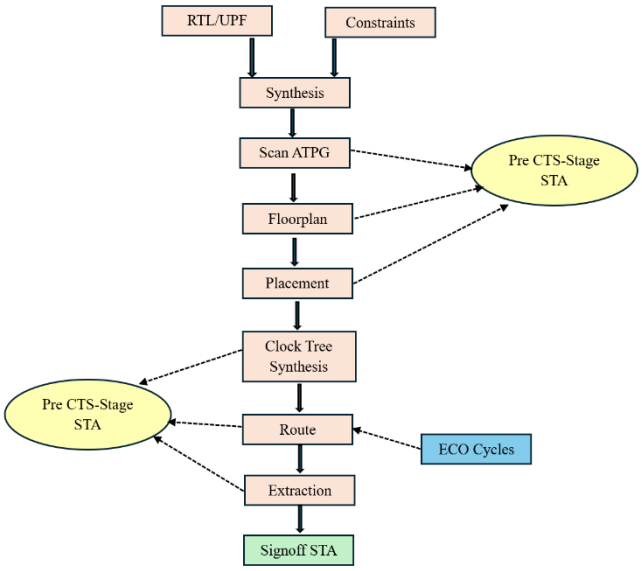


Figure 5.3: General Physical Design Flow

5.2.1 Prime Time Tool from Synopsys

The PrimeTime tool is a gate level static timing analysis tool from Synopsys. It takes the design information in gate-level Verilog netlist, delay information in Standard Parasitic Exchange Format (SPEF), and timing constraints in Synopsys Design Constraint (SDC) format.

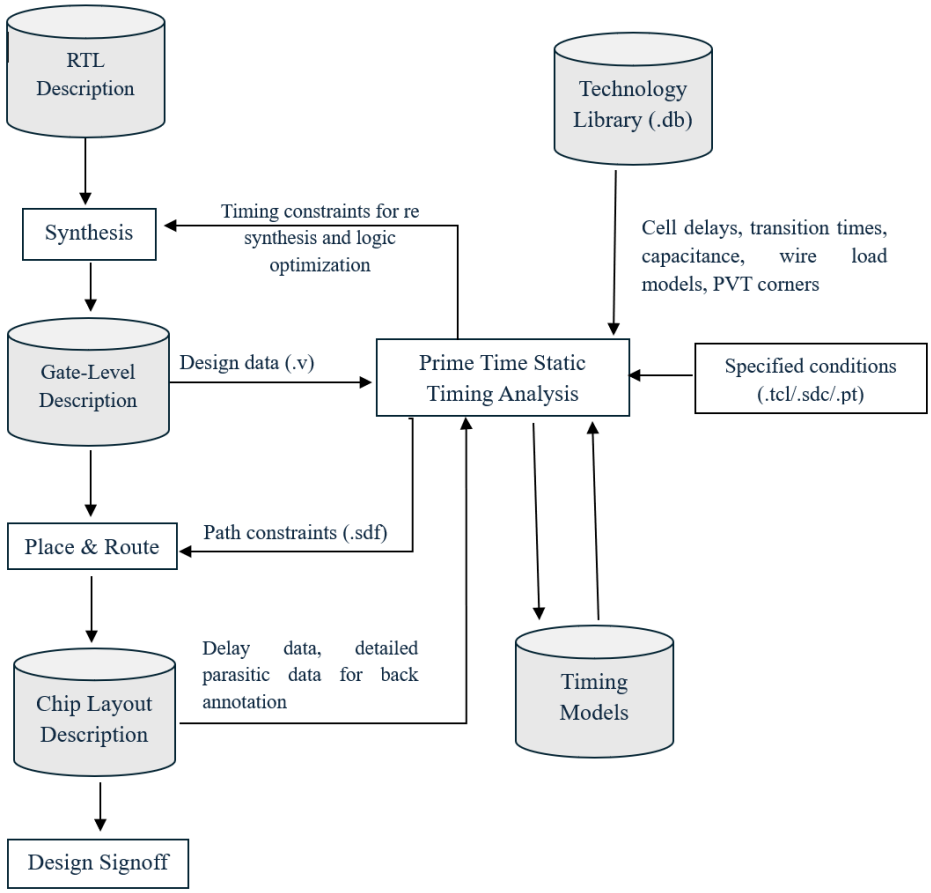


Figure 5.4: Synopsys PrimeTime STA Flow

PrimeTime performs the following types of design checks:

1. Setup, hold, recovery and removal constraints
2. User-specified data-to-data timing constraints
3. Clock-gating setup and hold constraints
4. Minimum pulse width for clocks
5. Design rules (minimum/maximum transition time, capacitance, and fanout)

Some of the Analysis Features include:

1. Multiple clocks and clock frequencies
2. Simultaneous minimum/maximum delay analysis for setup and hold constraints
3. Transparent latch analysis and time borrowing
4. Analysis with on-chip variation of process, voltage, and temperature (PVT) conditions
5. Multicycle path timing exceptions
6. False path timing exceptions and automatic false path detection

7. Case analysis (functional/scan)
8. Mode analysis (single/BC-WC/on-chip variation)
9. ECO analysis without modifying the original netlist, using inserted buffers, resized cells, and modified nets
10. Analysis of crosstalk effects between physically adjacent nets using the PrimeTime SI (signal integrity) option.

5.2.2 Basic STA Flow

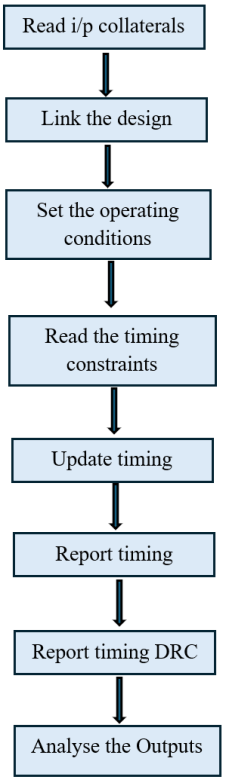


Figure 5.5: The detailed STA Flow

1. The first step is to read the Verilog files i.e the input collaterals with the help of the command **read_verilog <pointer to the file>.**
2. Next is setting the link path from where the libraries will be read.
   set link_path [list <path to the libraries to be read>]
3. Define the scaling library groups with the exact match only. Library scaling refers to the process of defining how different cells in a timing library (standard cells) are scaled for various process conditions, such as different voltages, temperatures, or corner cases. It

allows the tool to account for variations in performance based on these scaling factors. A scaling library group is a set of cells or libraries that PrimeTime can use to adjust timing analysis based on different PVT conditions. The **-exact_match_only** option limits the library scaling to only those cells whose names exactly match the given library group definition. When this option is used, only exact matches to the scaling library group will be considered during timing analysis, rather than allowing partial matches or using a more relaxed matching rule.

define_scaling_lib_group <group_name> -exact_match_only <lib_name> <cell_names>

4. Link the logical design (which consists of synthesized RTL) with the physical design (which consists of the netlist and placement information). Running the **link_design -force** command ensures that the latest changes are linked into the design without worrying about potential conflicts or pre-existing linked states.
5. The operating conditions are then set using the **set_operating_conditions -analysis_type on_chip_variation** command. The -analysis_type on_chip_variation option specifies that on-chip variation (OCV) should be applied to the operating conditions. On-chip variation refers to the variability in process, voltage, and temperature (PVT) conditions that occur across different regions of the chip due to factors such as manufacturing variations, temperature gradients, and voltage fluctuations. By setting OCV as the analysis type, the tool accounts for these variations across the chip to evaluate the worst-case performance, which is important for accurate timing analysis and signoff.
6. The **load_upf** command is then used to load the power intent specified in a UPF file so that the tool can analyse, optimize, and verify the design with respect to its power characteristics. UPF files contain information about power domains, isolation, retention, level shifting, and other power management features, and loading the UPF allows the tool to take this information into account during synthesis, simulation, and signoff.
7. A TCL script containing the timing and clock constraints is then parsed which allows the tool to get aware about all the constraints present in the design.
8. The **update_timing** and **check_timing** commands are used to check the timing reports with respect to the design and it also gives the details on the slack requirements and other critical parameters.
9. The session is then saved using the **save_session** command.
10. The exact reports can be checked with the **report_timing** command. Other parameters can also be reviewed with the help of commands like **report_qor** and **report_constraits -all_violators**.
11. The final model can be extracted using **extract_model** command and timing characteristics (such as delay, setup, and hold time) of the cells in a library or specific design block can be viewed. These models are required for accurate static timing analysis.

CHAPTER 6

RESULTS, DISCUSSION AND CONCLUSIONS

6.1 SIMULATIONS AND INTERIM RESULTS

According to the latest PDK changes that continue to happen, the noise, crosstalk, and caliber simulations as discussed above, take place in various iterations keeping in mind, the different scenarios and PVT corners. In the following subchapters simulation results of noise, glitch, and caliber runs are presented. The results of the ECO implementation are also presented.

6.1.1 Noise Simulation Results and Discussion

Noise simulations also take place over a PrimeTime saved session, with netlists and partitions' spefs as inputs. The output reports consist of internal, interface, and in-between partition violations, as discussed in the above chapters. A noise simulation report is shown below:

Noise Summary Report			
Type	Above_low	Below_high	Total
internal	37	41	78
interface_in	0	0	0
interface_out	0	0	0
interface_inout	2	2	4
between_partitions	0	0	0

Figure 6.1: Noise Summary Report, with Violation Count

The noise summary report shown in figure 6.1, shows the total violation count in the different partitions of the chip. The report shows 78 internal violations in a certain partition, which is bifurcated into 37 no.s above the VSS and 41 no.s below the VCC. On analysing the reports further, it is observed that some violations (categorised at the **interface_inout** type in Fig 6.1) are occurring in the analog nets, that can be waived, while

other violations are reported to the various partition owners, and actions are taken accordingly. On further iterations of performing the noise analysis, it is observed that slowly the internal violations keep on reducing, and finally the noise run is clean of all the violations, making the partitions of the top die ready for final handoff to the Methodology team, eventually going in for tape in.

Actions like shielding of the ground nets, upsizing the cells and inserting buffers have been taken to get the violation count to zero.

The double-switching report in Figure 6.2 shows that there are no violations in the maximum/minimum rising and falling of the nets.

Double Switching Report				
Max_Rise	Max_Fall	Min_Rise	Min_Fall	Total
0	0	0	0	0

Figure 6.2: Double Switching Report

6.1.2 Detailed Noise Analysis in Synopsys Prime Time

The Synopsys Prime Time tool is used to perform the noise calculation for a specific net arc. The command uses the noise analysis settings for the design set by the set_noise_parameters command. It also performs a noise update. The report shows generic information about the victim, and details of noise bump and noise slack calculations. This includes the total noise bump, estimated from each of the aggressors. There were multiple reports generated during the noise analysis in our design. An example report is shown below:

Report : noise_calculation

-from
par_base_fabric_slice_0/analog_viewpin_mux_ch1_s0/socviewpin_anadrvhip_inst/ana_lya_a

-to par_base_fabric_slice_0/d2d_slice_s0_viewpin1_0__PUSHDOWN/pad

Design : cbb_base

Version: U-2022.12-SP5-5-VAL-20250116

Date : Wed May 14 23:41:45 2025

Units: 0.001ns 0.001pF 1kOhm

Analysis mode: report_at_endpoint Region: above_low

Victim net : AS_VIEWPIN_ANA_1_TOP_RED0_0

Victim driver effective capacitance: 763.871 Steady state resistance source: user set value

Driver voltage swing : 0.675 Noise derate height offset: 0.000

Noise derate height scale factor: 1.000 Noise derate width scale factor : 1.000

Noise effort threshold: 0.200 Noise composite aggressor mode: statistical

Noise calculation attributes:

A - aggressor is active

C - aggressor is a composite aggressor

D - aggressor is analyzed with detailed engine

E - aggressor is screened due to user exclusion

G - aggressor is analyzed with gate level simulator

I - aggressor has infinite window

L - aggressor is screened due to logical correlation

P - aggressor is an ideal port aggressor

S - aggressor is screened due to small bump height

X - aggressor is screened due to aggressor exclusion

Height Width Area Aggressor Attributes

Aggressors:

AS_VIEWPIN_ANA_0_TOP_RED0_0

0.095 2052.850 97.380 A I D

Propagated:

par_base_fabric_slice_0/analog_viewpin_mux_ch1_s0/socviewpin_anadrvhip_inst/ana_lya_a

0.007 160.000 0.540

Total: 0.102 1927.124 97.920

Noise slack calculation:

Constraint type: user margin

	Height	Area
Required	0.085	(0.085 * 1927.124) -
Actual	0.102	(0.102 * 1927.124)
Slack	-0.017	-32.035

Analysis mode: report_at_endpoint Region: below_high

Victim driver effective capacitance: 763.871 Steady state resistance source: user set value

Driver voltage swing: 0.675 Noise derate height offset: 0.000

Noise derate height scale factor : 1.000 Noise derate width scale factor: 1.000

Noise effort threshold: 0.200 Noise composite aggressor mode: statistical

Noise slack calculation:

Constraint type: user margin

	Height	Area
Required	0.085	(0.085 * 5245.719) -
Actual	0.277	(0.277 * 5245.719)
Slack	-0.192	-1005.202

6.1.3 *Intel Caliber Violation Summary Report*

After running the caliber flow, as discussed in the previous chapter, violation reports are generated which are then analysed based on which rules are important, and accordingly, actions like waiving and fixing violations take place. Also, reports and violation count are compared to the previous runs, in order to have a comparative analysis between the current run and the previous run.

Table 4.1 shows a Caliber summary report:

Rule Name	Violation Count in different scenarios				Severity
	S1	S2	S3	S4	
BadExtractionQuality	114	114	114	114	must
BigWireDelay	9	9	3	9	must
CellMissingPOCVM	1	1	1	1	must
ClockMinPulseWidth	6	6	7	6	must
ClockSlowSlope	553	553	0	553	must
MaxCap	12	12	12	12	must
NetMissingExtractionData	92	92	92	92	must
PTLogError	12	12	12	12	must
PTLogExtractionError	91	91	91	91	must
SlowSlope	37	49	7	49	must

Table 6.1: Caliber Rules Summary Violation Count

The rule with violation counts, are analysed accordingly and checked for the partitions that are having these violations. Inputs are given to the respective owners and the methodology team, so that they may fix them, or waive them, as per the requirement. A total of 30 iterations were done to clean up all the violations in a period of 15-20 work weeks. Table 6.2 blow shows the Caliber rule summary after the 25th iteration. Mostly all the violations are waived, following a certain procedure, and all the waivers are documented in a central repository corresponding to which tickets are raised which can be referred at any instance.

Rule	Count	Partition name
BadExtractionQuality	7	Par 1
CellMissingPOCVM	2	Par 2
ClockMinPulseWidth	7	Par 3
	5	Par 4
ClockSlowSlope	556	Par 5
		Par 6
		Par 7
GeneratedClocksSource	18	Par 8
IllegalCells	247	Par 9
MaxCap	16	Par 10
NetMissingExtractionData	22	Par 11

Table 6.2: Caliber Rules Summary Violation Count after 25th Iteration

6.1.4 ECO-Results

It is observed that the slack requirements have been met for the partitions for which on route buffer insertion was performed. The partition on which net detouring was performed, also shows positive slack, within the sigma requirements. Brief analysis of the results has been shown below in Table 6.3.

Start Point Block	End Point Block	WNS - Before	WNS - After	WNS Corner
BLOCK 1	BLOCK 1	-6.07	1.66	CORNER 1
BLOCK 2	BLOCK 2	-8.33	2.03	CORNER 2

Table 6.3: Result Table for ECO Implementation

6.2 CONCLUDING REMARKS AND FUTURE SCOPE

This report thoroughly explores the complexities and challenges associated with the design and analysis of VLSI circuits, with a particular focus on timing closure, crosstalk and noise analysis, and caliber flows. Through a detailed examination of various techniques and methodologies, the critical role of robust design strategies has been demonstrated in achieving high-performance, reliable circuits within the constraints of advanced technology nodes.

Key findings from this study include:

1. **Timing Closure and Noise Management**: The importance of **timing analysis** in achieving optimal performance in VLSI designs has been highlighted by incorporating **noise-aware static timing analysis** (STA), we showed how power grid noise, crosstalk, and other parasitic effects can impact the timing margins and how noise mitigation strategies can reduce these impacts. The caliber ruleset ensures the chip design's robustness while maintaining all the limits of the rules.

2. **Impact of Advanced Process Nodes**: As VLSI technology continues to advance to smaller nodes (e.g., 7nm, 5nm), we identified how parasitic effects and power noise challenges become more pronounced. The increasing complexity of metal layers interconnects, and substrate coupling demands new approaches to design verification and noise analysis to maintain chip integrity.

3. **Tooling and Methodologies**: Using advanced EDA tools for timing analysis, noise simulation, and caliber checks was instrumental in overcoming the challenges of designing within these increasingly stringent constraints. The integration of statistical methods and the POCV method provides deeper insights into the variability and uncertainty inherent in modern VLSI design, allowing for better-informed decision-making.

The outcomes obtained using the suggested methodology demonstrate its superiority over alternative approaches already in use. By removing the need for expensive ECO iterations, this method speeds up the timing closure. By segregating the worst scenarios for the multi-scenario analysis and ECO guidance flow, followed by precise filtering of dirty endpoints at each prioritized stage for mitigating the timing violations, it can effectively address the process challenges and avoid the ping-pong effect. By enabling the tool to conduct a thorough analysis of timing violations and then fixing the violations hierarchically, it enables the designers to accurately fix the timing violations without having to actively intervene. The findings of this methodology demonstrate memory savings, capacity gain, and the attainment of 90% to 95% cycle time goals. As a result, it shortens the time to market and verification time.

Therefore, by reducing the amount of manual labour required, this process effectively integrates the several steps of the flow and helps to improve performance and runtime.

This work lays a foundation for future research and advancements in full chip timing analysis contributing to the ongoing quest for smaller, faster, and more reliable integrated circuits in emerging technologies. The Noise and Caliber results have been summed up, and hold violations are clean as a result of ECO implementation. Further, the STA flow and the corresponding results will be discussed, to conclude the final steps of timing closure, and then the design will be ready for final tape in.

REFERENCES

[1] Myeoungwoo Jin Deokkeun Oh, Eunsuk Park and Juho Kim. K-Critical Path Search Based Multi corner Multi mode Static Timing Analysis. ISOCC, pages 212–213, 2014.

[2] S. Narendra J. Tschanz A. Keshavarzi S. Borkar, T. Karnik and V. De. Parameter variations and impact on circuits and microarchitecture. Design Automation Conference, pages 338–342, 2003.

[3] STMicroelectronics "Strategies for Efficient timing signoff convergence at VDSM nodes" Mohit verma

[4] Sudhakar Jilla. Using multi-corner multi mode techniques to meet the P&R challenges at 65 nm and below. http://www.techdesignforums.com/practice/technique/ using-multi-corner-multi-mode-techniques-to-meet-the-p-r-challenges-at-65-nm-and-below/, 2007.

[5] J.Bhasker and Rakesh Chadha. Static Timing Analysis For Nanometer Designs: A Practical Approach. Springer, 2009.

[6] Primetime and Primetime SI User Guide.

[7] Yao-Kai Yeh Jui-Hung Hung, Yung-Sheng Tseng, and Tsai-Ming Hsieh. A New ECO Technology For Functional Changes And Removing Timing Violations. International Symposium on Quality Electronic Design, pages 1–5, 2011.

[8] Olivier Coudert. Timing and Design Closure in Physical Design Flows. International Symposium on Quality Electronic Design, pages 1–6, 2002.

[9] Kan Mei War. CAD Automation Module Based on Cell Moving Algorithm for ECO Timing Optimization. IEEE Symposium on Industrial Electronics and Applications, pages 284–288, September, 2011.

[10] White Paper: Signoff Driven Timing Closure ECO in the Synopsys Galaxy Platform

[11] Joel R. Phillips Luis Guerra e Silva, L. Miguel Silveira. Efficient Computation of the WorstDelay Corner. DATE, pages 1–6, 2007.

[12] Arvind NV Sreeram C Vish Visvanathan-Shailendra Dhuri Roopesh Chander Patrick Fortner Subra Sripada Qiuyang Wu Rajagopal KA, Sivakumar R. A comprehensive solution for true hierarchical timing and crosstalk delay signoff. International Conference on VLSI Design, pages 1–6, 2006.

[13] Bakhtiar Affendi Rosdi Tee Kok Tiong Ang, Lay Sean. Cone Extraction Technique for Incremental Static Timing Analysis. IEEE Symposium on Industrial Electronics and Applications, pages 1–6, September 2011

[14] White Paper: Intel Delivers Cutting-Edge Process Technologies to the Data Center with Intel 18A and Advanced Chiplet Packaging

[15] White Paper: Introduction to Intel® Architecture

[16] Product Brief: The Intel® Xeon® 6 Processor Family

[17] Timing analysis journey from OCV to LVF: Mahajan Rita, First International Conference on Advances in Physical Sciences and Materials Conf. Ser. 1706 012081

[18] EDWARD L. TITLEBAUM BORIS ANDREEV and EBY G. FRIEDMAN. SIZING CMOS IN VERTERS WITH MILLER EFFECT AND THRESHOLD VOLTAGE VARIATIONS. Jour nal of Circuits, Systems, and Computers, pages 437454, March 2006.

[19] Kiat-Seng Yeo Wang-Ling Goh, Samir S. Rofail. Low-Power Design: An Overview. Pearson Education, Inc, June 2002.

[20] David Money Harris Neil Weste. CMOS VLSI Design: A Circuits and Systems Perspective. Pearson Education, Inc, 4th edition, March 2010.

[21] Vishwani D. Agrawal Tezaswi Raja and Michael L. Bushnell. Transistor Sizing of Logic Gates to Maximize Input Delay Variability.

[22] M. Jasmin. Optimization Techniques for Low Power VLSI Circuits. Middle-East Journal of Scienti c Research, pages 10821087, 2014.

[23] Marek-Sadowska M. Xiao, T. Crosstalk reduction by transistor sizing. Design Automation Con ference, 1999. Proceedings of the ASP-DAC 99. Asia and South Paci c, vol. 1:137140, Jan, 1999.

[24] Stephen T. Quay Charles J. Alpert, Anirudh Devgan. Bu er Insertion for Noise and Delay Optimization. Design Automation Conference, pages 16, 1998

0606.pdf

ORIGINALITY REPORT

17%	16%	4%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	technodocbox.com Internet Source	6%
2	www.pythonclub.org Internet Source	2%
3	www.intel.cn Internet Source	1%
4	ebin.pub Internet Source	1%
5	www.slideshare.net Internet Source	1%
6	www.intel.co.jp Internet Source	1%
7	medium.com Internet Source	1%
8	link.springer.com Internet Source	1%
9	iccircle.com Internet Source	1%
10	www.edaboard.com Internet Source	1%
11	www.nor-tech.com Internet Source	<1%
12	www.micro-ip.com Internet Source	<1%

13	repository.iiitd.edu.in Internet Source	<1 %
14	Submitted to Institute of Technology, Nirma University Student Paper	<1 %
15	dspace.dtu.ac.in:8080 Internet Source	<1 %
16	de.scribd.com Internet Source	<1 %
17	www.cadence.com Internet Source	<1 %
18	www.imse.cnm.es Internet Source	<1 %
19	Submitted to Universiti Teknologi Malaysia Student Paper	<1 %
20	Submitted to Liverpool John Moores University Student Paper	<1 %
21	Submitted to CSU, San Jose State University Student Paper	<1 %
22	Submitted to University of Minnesota System Student Paper	<1 %
23	tel.archives-ouvertes.fr Internet Source	<1 %
24	library.oapen.org Internet Source	<1 %
25	Submitted to Universiti Teknologi Petronas Student Paper	<1 %

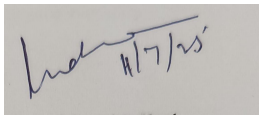
26	Xiaokun Yang. "Integrated Circuit Design - IC Design Flow and Project-Based Learning", CRC Press, 2024 Publication	<1 %
27	kipdf.com Internet Source	<1 %
28	"Static Timing Analysis for Nanometer Designs", Springer Science and Business Media LLC, 2009 Publication	<1 %
29	Dawid Żytko, Marcin Badurowicz. "Analysis of performance and energy efficiency of processors with hybrid architecture", Journal of Computer Sciences Institute, 2025 Publication	<1 %
30	I SAVIDIS. "Physical Design Trends for Interconnects", On-Chip Communication Architectures, 2008 Publication	<1 %
31	Richard P Mirin, Arthur C Gossard, John E Bowers. "Characterization of InGaAs quantum dot lasers with a single quantum dot layer as an active region", Physica E: Low-dimensional Systems and Nanostructures, 1998 Publication	<1 %

Exclude quotes On

Exclude bibliography On

Exclude matches Off

Supervisor:



Co-Supervisor: