Implementation of tailored CTS techniques for PPA enhancement in lower technology nodes

Praveenkumar J[1], and *Sudhagar* G[2]

[1] Research Scholar, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India
[2] Associate Professor, Department of ECE, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India

Abstract. Clock Tree Synthesis is a critical stage in VLSI design, ensuring reliable and efficient clock signal distribution. An optimized clock tree minimizes skew and insertion delay, crucial for timing closure in high-frequency designs. Smaller technology nodes increase design complexity and impose stricter PPA constraints. This research emphasizes the importance of Clock Tree Synthesis in advanced technology nodes and highlights the adverse effects of suboptimal clock tree designs. Key design metrics such as silicon area, timing margins, and dynamic power are directly influenced by CTS. We propose systematic timing path balancing and architectural enhancements aimed at maximizing performance while minimizing resource overhead, thereby improving overall design robustness.

1 Introduction

With the continuous evolution of semiconductor technology and the shift toward smaller technology nodes, VLSI designs have become increasingly advanced, offering greater processing capabilities and functionality. This progress comes at the cost of higher design complexity, especially in achieving design sign-off for manufacturing.

As technology scales down, several design metrics such as power, performance, area (PPA), and signal integrity must be met with higher precision. This exponential increase in complexity poses significant challenges for design engineers during the signoff phase. Meeting timing, power, and reliability constraints is now far more complex and time-consuming.

One of the key challenges in this context is the construction of an efficient and robust clock tree structure. The clock tree plays a vital role in synchronizing all sequential elements (like flip-flops and registers) across the chip. A poorly designed clock tree can result in skew, jitter, or delays that violate timing requirements, affecting the chip's functionality and performance.

2 Objectives

The objectives of this research are as follows: distributing clock signals uniformly across the chip; keeping clock skew to a minimum; reducing clock latency and power consumption; balancing the load among clock buffers and inverters; and accounting for variations in process, voltage, and temperature.

3 Literature survey

Gandhi et al. examined sophisticated floor planning strategies for improving power efficiency in semiconductor devices that make use of the 28nm technology node [1]. By optimizing the arrangement of electronic components, the authors offer several techniques that significantly enhance power budgeting.
Bhasker and Chadha provided a thorough introduction to static timing analysis for nanoscale architectures [2]. The book addresses important subjects like cell and interconnect modeling, timing computation, and the effects of crosstalk.

Ge et al. presented methods for improving clock tree designs with an emphasis on power consumption reduction [3]. The authors provide novel methods for improving clock tree architectures, which are essential for integrated circuit performance and power economy.

4 Metrics affected by CTS

4.1 Skew

Skew represents the difference in clock arrival times between two sequential elements. Although its impact is negligible in low-frequency designs, even minor skew variations can cause critical timing issues in high-frequency circuits. Skew influences both setup and hold timing, often leading to violations.
Impact on Timing:

- Positive skew (capture clock arrives later than launch clock) improves setup timing but can worsen hold timing.
- Negative skew (capture clock arrives earlier) benefits hold timing but may degrade setup timing.
  Excessive skew can result in timing failures, requiring additional buffers or clock tree optimization.

[1]Corresponding author: praveencando91@gmail.com

As illustrated in Figure 1, FF1 is located near the clock port with one buffer, while FF2 requires extra buffers during optimization. This scenario demonstrates an unbalanced clock tree, where the required time exceeds the arrival time, causing setup violations. Figure 2 shows clock signal variations, highlighting timing differences between FF1 and FF2. Figure 3 presents an insertion delay map from the clock source (CLK1) to multiple flip-flops, emphasizing how buffer placement affects skew and timing paths.

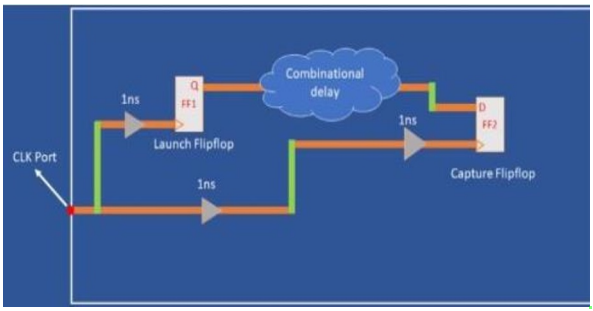

Fig. 1. Local skew representation showing unbalanced clock paths between FF1 and FF2.

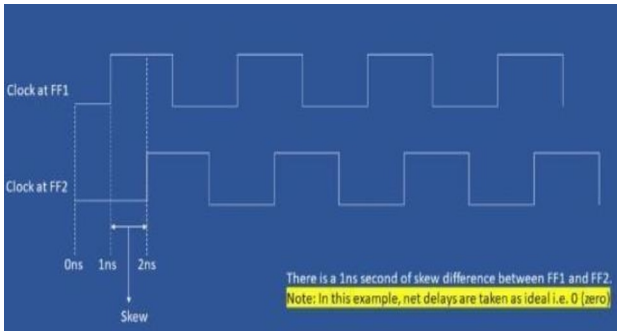

Fig. 2. Variation of clock signals showing timing differences.

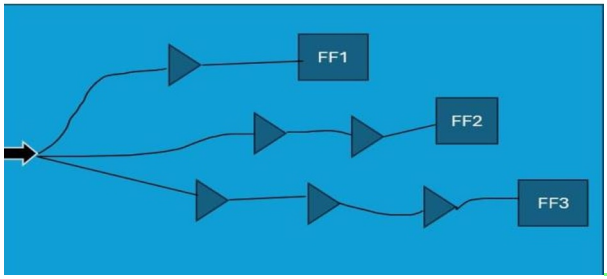

Fig. 3. Insertion delay map from CLK1 source to flip-flops.

4.2 Insertion delay

Insertion delay is the time taken for the clock signal to travel from source to sink. The inserted component has a propagation delay which causes this delay. To guarantee that the clock signal reaches the sinks simultaneously, skew balancing must be performed.

Faster clock speeds typically demand smaller insertion delays to maintain proper timing and avoid setup and hold violations. As the clock period shortens, any delay becomes a larger portion of the total cycle time.

4.3 Latency

In sequential architectures, a clock signal initiates each timing path. The duration required for the clock signal to propagate from its source to the sinks is referred to as clock latency. This latency consists of two components: clock source latency and clock network latency.

- Clock source latency is the time taken for the signal to travel from the clock source to the clock definition point.
- Clock network latency is the time from the clock definition point to the sinks.

The total clock latency is the sum of these two components.

Figure 4 illustrates the clock latency structure, highlighting the source latency segment from the PLL to the clock port and the network latency segment extending from the clock port to the sinks.

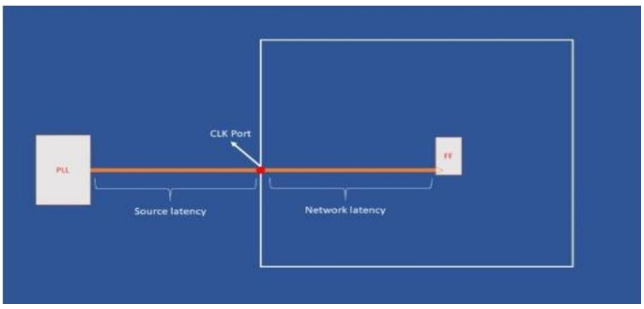

Fig. 4. Clock latency components showing source and network latency.

4.4 Jitter

Jitter is defined as the variation in the arrival of the clock edges from its ideal clock position which leads to data corruption and signal integrity issues. Jitter can arise from various sources including noise within the oscillator circuit, power supply variations, interference from nearby circuitry, and thermal or mechanical noise.

4.5 Duty cycle

The duty cycle is defined as on-time divided by the sum of on-time and off-time. Clock distribution networks in SoCs contain delay elements that often cause timing issues. Duty cycle reduction particularly affects timing when both clock edges are involved, such as in half-cycle timing paths and minimum pulse width checks.

4.6 Impact on power

Bad clock tree construction can significantly impact power consumption in VLSI designs. A weak build clock tree will increase the switching activity in the design due to excess addition of buffers and inverters. Dynamic power consumption is directly proportional to switching frequency according to the equation:

$$P = \alpha \times C \times V^2 \times f$$

where α is the activity factor, C is capacitance, V is voltage, and f is frequency.

4.7 Premapping flops

During synthesis before the technology mapping step, we select required flip-flops with better drive strength for good propagation delays. This process involves

mapping generic cells to library cells and performing logic optimization.

4.8 Non-default rules

To overcome challenges like crosstalk, EMIR, and timing issues, we create non-default routing rules such as minimum layer, maximum layer, and shielding constraints. These rules are applied to specific nets during routing.

4.9 I/O delay constraints

The propagation time for data between the external circuit and the input pins, or from the output pins back to the external circuit, must be accurately modeled. Proper modeling of input and output delays is essential to satisfy timing constraints and maintain correct functionality.

As illustrated in Figure 5, communication between two external blocks (Block A and Block B) involves I/O delays. The figure highlights how signals pass through combinational logic before reaching flip-flops, emphasizing the importance of precise delay estimation for reliable timing closure.

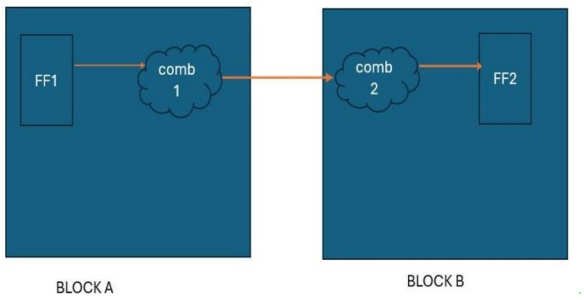

Fig. 5. Communication with external block showing I/O delays.

4.10 Path groups

Path groups are created for data dominant paths with multiple levels of logic. The cost function is defined as the sum of all groups weighted by their violations. If a path group has more violations, the tool prioritizes optimization of that group.

4.11 Pre-routes

RC parasitic values for lower metal layers are higher compared to upper layers. Custom pre-routes can be created to ensure optimal routing from critical sources such as PLLs, avoiding detours and minimizing delays.

4.12 Bounding logic

During placement, hierarchical modules in a design may have logic cells that become scattered across the layout. This dispersion can lead to increased interconnect delays and potential timing violations. Bounding logic involves grouping related cells together to maintain proximity, thereby reducing wire length and improving timing performance.

As shown in Figure 6, unbounded logic results in dispersed module cells, which increases routing complexity and delay. In contrast, Figure 7 illustrates bounded logic, where cells are clustered together, minimizing interconnect delays and enhancing timing closure.

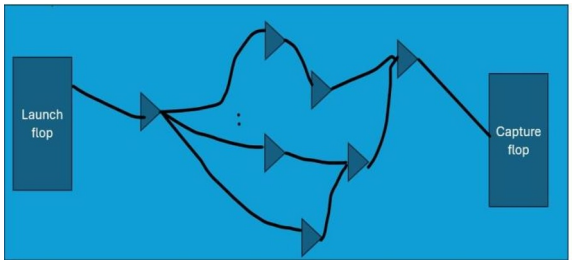

Fig. 6. Representation of unbounded logic showing dispersed module cells.

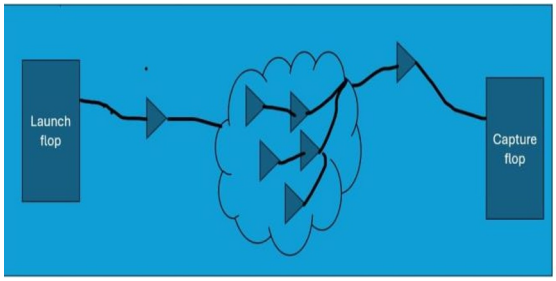

Fig. 7. Representation of bounded logic with cells grouped together.

5 Design architecture

The design architecture shown in Figure 8 consists of multiple hierarchical blocks:

- SoC (System-on-Chip)
- Data Fabric
- UMC (Unified Memory Controller)
- DDR-PHY (Physical Layer for DDR)
- DIMM (Memory Module)

These blocks communicate through high-speed interfaces, where timing closure is critical. Interface timing closure ensures that signals crossing these boundaries meet setup and hold requirements under all operating conditions.

5.1 Interface Timing Challenges

Multiple clock domains across SoC, UMC, and DDR-PHY. High-speed DDR interface requiring precise I/O delay modeling. Increased skew and latency due to long interconnect paths.

Tailored CTS Structure for Timing Closure

To overcome these challenges, a customized CTS strategy was implemented:

Hierarchical CTS: Separate clock trees for SoC, UMC, and DDR-PHY, with controlled skew budgets at each interface.

Balanced Buffer Insertion: Buffers were placed strategically to minimize insertion delay and maintain uniform latency.

Clock Clustering: Critical interface flops were grouped to reduce skew and improve timing predictability.

Multi-Corner Multi-Mode (MCMM) Optimization: CTS was tuned for worst-case corners to ensure robust timing across PVT variations.

Low-Power Techniques: Clock gating and multi-source CTS were applied without compromising timing integrity.

This tailored CTS approach significantly reduced skew and latency at interface boundaries, enabling timing closure for setup and hold paths across all blocks.

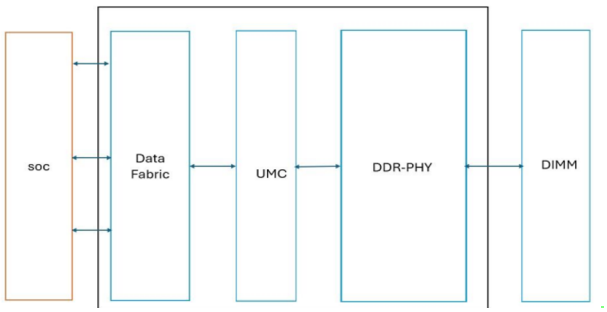

Fig. 8. Interface block diagram of the design architecture.

5.2 Clock trunk implementation

Longer route lengths can negatively impact performance, increase chip area, and affect reliability and power consumption. These longer routes lead to increased signal delays which degrade circuit speed and overall performance.

5.3 Clock methodology

Modern clock distribution employs three primary methodologies to manage timing and skew. Mesh structures deliver extremely low On-Chip Variation (OCV) and minimal skew, making them highly reliable for timing closure, though they require complex analysis and consume more power. In contrast, conventional CTS offers a simpler and power-efficient solution but suffers from higher OCV and skew. A middle ground is provided by multi-point CTS, which introduces multiple clock entry points into sub-blocks, improving balance and reducing skew. As illustrated in Figure 9, a mesh clock structure achieves uniform clock arrival by interconnecting multiple nodes, ensuring consistent timing across the design.

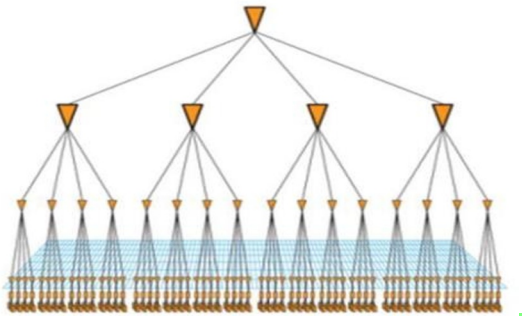

Fig. 9. Mesh clock structure for balanced distribution.

6 Cells used to build CTS

Reducing the number of cells in the clock tree particularly buffers or repeaters can significantly improve design efficiency. Fewer cells lead to lower power consumption, reduced insertion delay, and smaller area requirements. For high-speed clocks, the switching factor is a key contributor to power usage.

Table 1 compares conventional CTS with a tailored CTS approach, showing notable improvements. Tailored CTS reduces total buffers by 25.6%, clock power by 25.2%, and insertion delay by 19.4%, while skew improves by 33.3%. These results demonstrate how a customized CTS structure enhances timing closure and power efficiency.

Table 1. Instance count difference between conventional and optimized approaches.

Metric	Conventional CTS	Optimized CTS	Improvement
Total Buffers	2,450	1,823	25.6%
Clock Power (mW)	145.3	108.7	25.2%
Max Insertion Delay (ps)	387	312	19.4%
Skew (ps)	42	28	33.3%

7 Design specifications

The design specifications include technology node, clock frequency targets, power constraints, and area requirements. These specifications guide the CTS optimization process and define the constraints within which the clock tree must be synthesized.

Table 2. Design specifications and parameters.

Parameter	Specification
Technology Node	7nm FinFET
Clock Frequency	2.5 GHz
Supply Voltage	0.75V
Die Area	12.5 mm²
Total Clock Sinks	~185,000
Target Skew	< 30 ps
Target Insertion Delay	< 350 ps
Clock Power Budget	< 120 mW
Operating Temperature	-40°C to 125°C
Clock Tree Levels	8-10

8 Conclusion

This study underscores the critical role of Clock Tree Synthesis in advanced technology nodes and introduces tailored optimization techniques to enhance Power, Performance, and Area. Key findings include significant reductions in clock skew, insertion delay, and dynamic

power consumption, achieved through systematic timing path balancing, hierarchical bounding, and non-default routing rules. Future work will focus on leveraging artificial intelligence driven algorithms and machine learning models to further optimize clock tree synthesis for next-generation semiconductor designs.

References

Journal Articles

1. S. Roy et al., Clock tree resynthesis for multi-corner multi-mode timing closure. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 34 (2015).
2. Y.C. Lu et al., A clock tree prediction and optimization framework using generative adversarial learning. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41 (2022).
3. J. Wu, C. Ni, H. Wang, J. Chen, Graph neural networks for efficient CTS optimization, ACE Journal, 150 (2025). https://doi.org/10.5555/ace.2025.22281
4. J. Lu, W.K. Chow, C.W. Sham, Fast power- and slew-aware gated clock tree synthesis, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 20, 11, 2094–2103 (2012). https://doi.org/10.1109/TVLSI.2011.2168834

Books

5. J. Bhasker, R. Chadha, Static Timing Analysis for Nanometer Designs (Springer, 2009)

Conference Proceedings

6. V. Gandhi et al., Power budget improvement using floorplan methodologies in lower (28nm) technology node, in Proc. IEEE Int. Symp. Smart Electron. Syst. (iSES), Ahmedabad, India (2023)
7. Z. Ge et al., Improve clock tree efficiency for low power clock tree design, in Proc. 13th IEEE Int. Conf. Solid-State Integr. Circuit Technol. (ICSICT), Hangzhou, China (2016)
8. N. Kwon, D. Park, Lightweight buffer insertion for clock tree synthesis visualization, in Proc. Int. Conf. Electron. Inf. Commun. (ICEIC) (2022)
9. Hybrid multisource clock tree synthesis, in Proc. IEEE Int. Conf. Electron. Circuits Syst. (ICECS) (2021)
10. AI-driven approaches for CTS and signal integrity optimization, IEEE (2023)
11. A Jain et al., Artificial neural network based post-CTS QoR report prediction, in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS) (2022)

Theses

12. J.L. Tsai, Clock Tree Synthesis for Timing Convergence and Timing Yield Improvement in Nanometer Technologies, Ph.D. thesis, University of Wisconsin-Madison, Department of Electrical Engineering (2005)
13. P. Jeffery, Evolutionary Neural Network for Optimized Clock Tree Synthesis, M.S. thesis, Rochester Institute of Technology, Department of Electrical Engineering (2024)
14. M. Kim, Robust Physical Design and Design-Technology Co-Optimization Methodologies at Advanced VLSI Technology, Ph.D. thesis, University of California, Berkeley, Department of Electrical Engineering and Computer Sciences (2023)
15. D. Mangiras, Timing Optimization Techniques for the Scalable Physical Synthesis of Digital Integrated Circuits, Ph.D. thesis, University of Michigan, Department of Electrical Engineering and Computer Science (2022).
16. R.R. Vuppunuthula, AI-Driven Approaches for Clock Tree Synthesis and Signal Integrity Optimization in Integrated Circuits, M.S. thesis, Indian Institute of Technology (2024).
17. X. Jiang, A Systematic Approach for Multi-objective Double-side Clock Tree Synthesis, Ph.D. thesis, Tsinghua University, Department of Microelectronics (2025).