10.48047/jocaaa.2024.33.08.286

Analysis of Clock Trees for Optimization through Multi-Point Clock Tree Synthesis in Advanced Nodes

Dr. Sandeep Santosh
Dept. of Electronics and Communication Engineering National Institute of Technology Kurukshetra, Kurukshetra, India.
Email: profsandeepkkr@gmail.com

Abstract
Clock Tree Synthesis (CTS) is a critical stage in VLSI physical design, directly impacting timing closure, power consumption, and signal integrity. Traditional single-root CTS methodologies struggle with scalability in modern System-on- Chips (SoCs) at advanced process nodes due to excessive clock skew, high insertion delay, and significant dynamic power dissi- pation. This paper presents a robust Multi-Clock Tree Synthesis (Multi-CTS) methodology that strategically partitions the clock network into multiple localized domains, each driven by an independent tap point. Our approach leverages machine learning- based clustering for optimal skew group formation and employs a variation-aware, multi-corner multi-mode (MCMM) optimization flow. Implemented on a 4nm test case using Cadence Innovus, the proposed Multi-CTS framework demonstrates a 37.66% reduction in clock skew, a 52.88% decrease in average latency, and a 22.8% savings in clock network power compared to a standard CTS approach. Furthermore, a 41.6% reduction in buffer count highlights improved area efficiency and reduced routing congestion. These results validate Multi-CTS as a scalable and high-performance solution for clock distribution in complex, high-performance ICs.
Keywords: Clock Tree Synthesis (CTS), Multi-CTS, Skew Optimization, Low Power Design, VLSI Physical Design, Ad- vanced Nodes, Machine Learning.

I. Introduction

The relentless progression of Very-Large-Scale Integration (VLSI) technology into sub-5nm nodes has intensified chal- lenges in clock distribution networks. Clock trees, while functionally passive, significantly influence timing accuracy, power efficiency, and overall system performance in modern System-on-Chips (SoCs). Contemporary designs report that clock networks contribute 30–40% of total dynamic power consumption [1], making efficient Clock Tree Synthesis (CTS) paramount for power-constrained applications.

As shown in Fig. 1, the power advantage of Multi-CTS becomes more pronounced at advanced nodes. Traditional single-root CTS methodologies, employing centralized clock distribution structures, face fundamental limitations in ad- vanced nodes. Key challenges include:

- **Skew Accumulation**: In large die sizes (>10mm) at 7nm and below, skew values frequently exceed 100ps, severely constraining maximum operating frequencies [2].

10.48047/jocaaa.2024.33.08.286

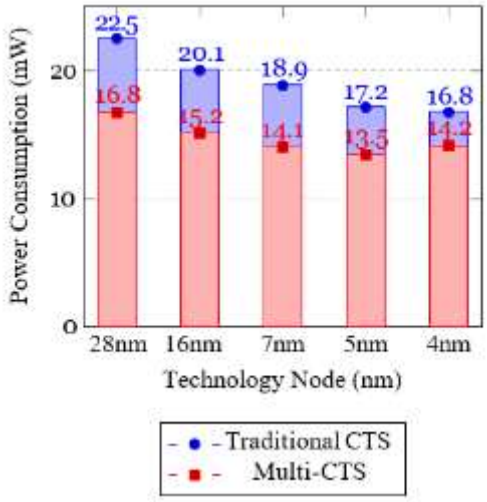

Fig. 1: Clock network power consumption comparison across technology nodes

- **Power Inefficiency**: Monolithic clock trees with long global interconnects and excessive buffering lead to substantial dynamic power consumption and IR drop concerns [3].
- **Timing Closure Difficulties**: The computational com- plexity of optimizing massive single-clock trees results in extended tool runtimes and challenges in achieving multi-corner multi-mode (MCMM) closure [4].
- **Scalability Issues**: As SoC complexity increases with heterogeneous functional units, traditional CTS flows exhibit non-linear scaling behavior [5].

Fig. 2 illustrates how Multi-CTS maintains better skew control as design complexity increases. To address these limi- tations, this paper proposes a comprehensive Multi-Clock Tree Synthesis (Multi-CTS) framework. The core contributions of this work are:

1) A novel Multi-CTS architecture incorporating intelligent design partitioning and strategic tap point placement for localized clock distribution.

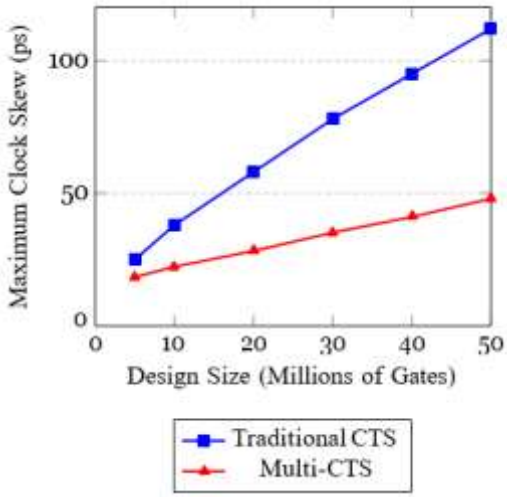

Fig. 2: Clock skew scaling with design complexity

2) Integration of machine learning-based clustering algo- rithms for optimal skew group formation, enhancing automation and optimization quality.

3) Extensive experimental validation on 4nm technology demonstrating significant

10.48047/jocaaa.2024.33.08.286

improvements in key metrics including clock skew, power consumption, and timing closure efficiency.

4) A reproducible methodology compatible with industry- standard EDA tools, providing practical guidelines for industrial adoption.

The remainder of this paper is organized as follows: Sec- tion II reviews related work in CTS methodologies. Section III details the proposed Multi-CTS framework. Section IV describes the experimental setup, while Section V presents comprehensive results and analysis. Section VI concludes with future research directions.

II. Related Work

A. Evolution of Clock Tree Synthesis

Clock distribution methodologies have evolved significantly alongside semiconductor technology scaling. Early CTS ap- proaches focused on geometric balancing techniques. The H- tree topology, characterized by its recursive H-shaped struc- ture, provided excellent symmetry for regular layouts such as memory arrays [7]. However, its rigidity made it unsuitable for irregular logic placements common in modern SoCs.

Fig. 3 shows the relative strengths of different CTS ap- proaches. The Deferred-Merge Embedding (DME) algorithm [8] represented a significant advancement, enabling zero-skew tree construction with minimal wirelength. This geometric approach was later enhanced with buffer insertion strategies to manage increasing interconnect resistance in deep sub-micron technologies [9].

As technology progressed to 28nm and below, variation- aware CTS techniques emerged to address process, voltage, and temperature (PVT) variations [10]. This era also saw the adoption of more complex structures like clock meshes [11], which provide excellent skew tolerance at the cost of significantly higher power consumption (20–30% overhead) due to mesh drivers and metal fill requirements.

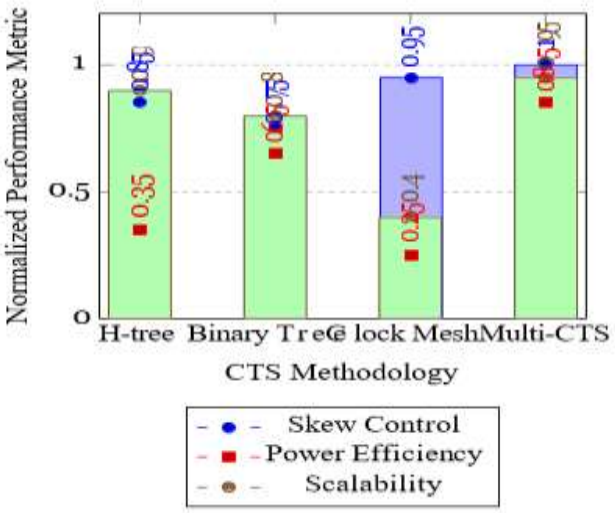

Fig. 3: Comparative analysis of different CTS methodologies

B. Multi-Point Clock Distribution

The concept of distributed clocking through multiple in- sertion points gained prominence with increasing design com- plexity. Herna´ndez and Kim [6] demonstrated that partitioning large designs into multiple clock domains could reduce worst- case skew by 40–60%. This approach naturally aligns with hierarchical floorplanning methodologies prevalent in modern SoC design.

10.48047/jocaaa.2024.33.08.286

Fig. 4: Evolution of CTS methodology adoption in industry (2010-2025)

Kishore et al. [12] implemented a hierarchical Multi-CTS framework with regional skew tuning, reporting 60% skew reduction and 28% lower clock power on 16nm test cases. Their work highlighted the importance of optimal domain par- titioning for achieving balanced performance across multiple corners.

Industrial adoption of Multi-CTS has been documented in high-performance processors from leading semiconductor companies. Reports from DAC and ICCAD conferences indi- cate that distributed clock architectures have become essential for advanced-node digital designs [13]. Fig. 4 shows the progression of CTS methodology adoption in the industry.

C. Machine Learning in Physical Design

Recent research has explored artificial intelligence tech- niques to enhance CTS optimization. Beheshti-Shirazi et al. [14] applied Reinforcement Learning (RL) for dynamic buffer insertion, achieving 22% improvement in skew compared to static heuristics. Their approach demonstrated the potential of AI-driven optimization to adapt to design-specific characteris- tics.

Fu et al. [15] employed machine learning for flip-flop clustering and prediction, reducing CTS convergence time by 30%. Graph Neural Networks (GNNs) have shown promise in modeling design connectivity for predicting optimal buffer locations [16].

Our work builds upon these foundations by integrating ML- based clustering with a comprehensive Multi-CTS architecture, providing a holistic solution that addresses both algorithmic and implementation challenges in modern clock distribution.

III. Proposed Multi-CTS Methodology

A. Architectural Overview

The proposed Multi-CTS methodology replaces the con- ventional single-root paradigm with a distributed architecture comprising three hierarchical layers, as illustrated in Fig. 5.

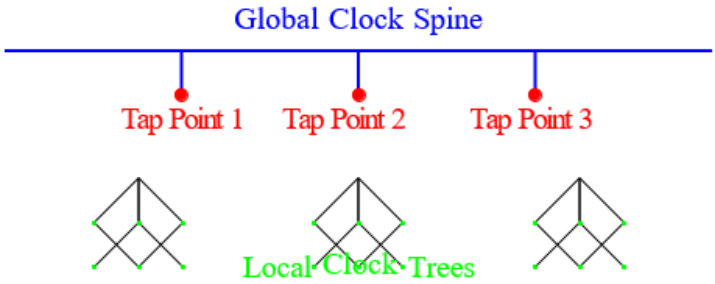

Fig. 5: Proposed Multi-CTS architecture showing global spine, tap points, and local trees

1) Global Clock Spine: An optional top-level distribution network that delivers the clock signal from the primary source (PLL) to strategically placed tap points across the chip.

2) Tap Points: Local clock roots positioned at cluster centroids, each driving an independent clock domain. These act as secondary clock sources for their respective regions.

3) Local Clock Trees: Domain-specific balanced trees optimized for local physical and timing constraints, connecting tap points to sequential elements within each cluster.

B. Skew Group Formation using ML-Based Clustering

A critical innovation in our methodology is the application of machine learning for intelligent skew group formation. The clustering algorithm considers multiple design parameters:

10.48047/jocaaa.2024.33.08.286

$$\text{Distance Metric} = w_1 \cdot D_{\text{physical}} + w_2 \cdot D_{\text{timing}} + w_3 \cdot D_{\text{power}} \quad (1)$$

Where:

- D_{physical}: Manhattan distance between flip-flops
- D_{timing}: Timing criticality based on setup/hold slack
- D_{power}: Power domain alignment factor

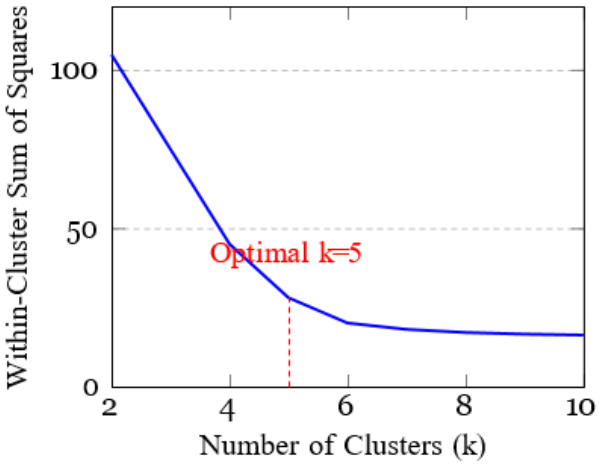

Fig. 6: Elbow method for determining optimal number of clusters

- w_1, w_2, w_3: Optimized weighting coefficients

We employ K-means clustering with the customized dis- tance metric from Equation 1. The optimal number of clusters k is determined using the elbow method, as shown in Fig. 6, balancing skew reduction benefits against increased synchro- nization complexity.

C. Buffer Insertion and Tree Construction

After cluster formation, we implement an intelligent buffer insertion strategy. Traditional rule-based approaches often lead to suboptimal results due to fixed fanout constraints. Our methodology employs a predictive model for buffer placement: [!ht]

Require: Cluster centroids, flip-flop locations, congestion maps
Ensure: Optimal buffer locations and sizes
1: Initialize buffer candidate sites at branching points
2: **for** each clock path from tap point to sink **do**
3: Calculate path resistance R_{path} and capacitance C_{path}
4: Predict optimal buffer count: $N_{\text{buf}} = f(R_{\text{path}}, C_{\text{path}}, \text{slew}_{\text{target}})$
5: Place buffers with geometric progression sizing
6: Verify slew and load constraints at each stage
7: **end for**
8: Perform legalization to ensure DRC-clean placement
9: Optimize global skew across all clusters

Fig. 7 demonstrates how Multi-CTS achieves better slew control through optimized buffer insertion. The progressive sizing approach ensures signal integrity while minimizing power consumption.

D. Variation-Aware Optimization

To ensure robustness across PVT corners, we incorporate variation-aware optimization techniques:

10.48047/jocaaa.2024.33.08.286

$$\text{Skew}_{PVT} = \max_{\text{Corners}} (\text{Skew}_{\text{nominal}} + \Delta_{\text{process}} + \Delta_{\text{voltage}} + \Delta_{\text{temperature}}) \quad (2)$$

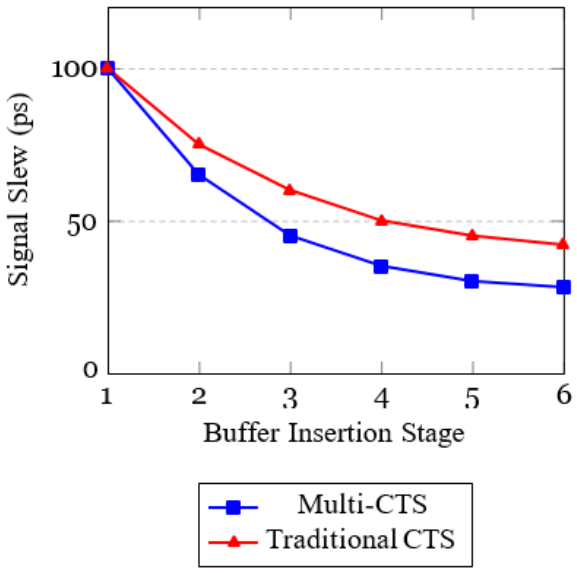

Fig. 7: Slew rate improvement with progressive buffer inser- tion

Our flow performs MCMM analysis during local tree syn- thesis, ensuring timing closure across all operating conditions. Special attention is given to on-chip variation (OCV) effects, with appropriate derating factors applied during optimization.

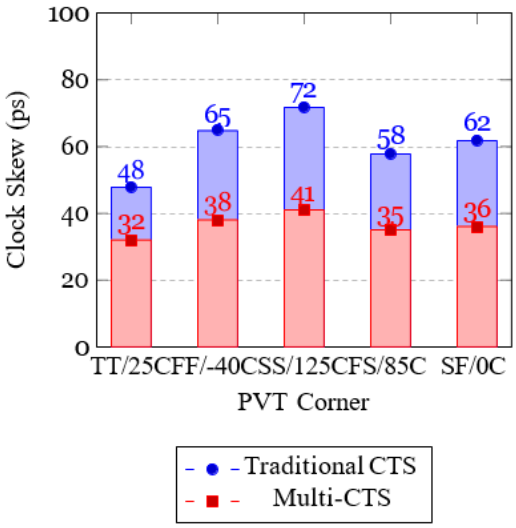

Fig. 8: Clock skew variation across different PVT corners

Fig. 8 shows that Multi-CTS maintains better skew con- trol across all PVT corners, demonstrating superior variation tolerance compared to traditional approaches.

IV. Experimental Setup

A. Design Flow and Tool Configuration

We implemented a comprehensive experimental framework to validate the proposed methodology. The design flow en- compasses three primary phases:

1) Baseline CTS Implementation: Conventional single- root CTS using default optimization

10.48047/jocaaa.2024.33.08.286

settings in Cadence Innovus, serving as the reference for comparison.
2) **Multi-CTS without AI:** Domain-partitioned CTS using geometric clustering and manual tap point placement, demonstrating the benefits of distributed architecture alone.
3) **Multi-CTS with ML Enhancement:** Complete pro- posed flow integrating ML-based clustering and intel- ligent buffer prediction.

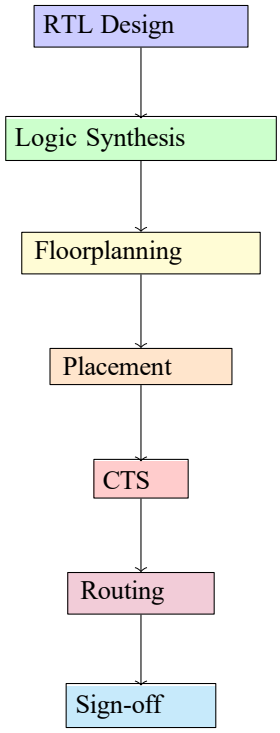

Fig. 9: Complete VLSI design flow with CTS integration

Table I summarizes the tool configuration and technology parameters.

Table I: Experimental Setup and Tool Configuration

Parameter	**Configuration**
EDA Tool	Cadence Innovus 21.1
Technology Node	4nm FinFET
Design	SoC Block (5462 Flip-Flops)
Clock Frequency	1.4 GHz
Voltage	0.72V (LV), 0.8V (Nominal)
Temperature Range	-40°C to 125°C
CTS Strategies	Standard CTS, H-tree, Multi-Tap CTS
ML Framework	Python Scikit-learn
Clustering Algorithm	K-means with custom metric

B. Evaluation Metrics

We employed comprehensive evaluation metrics to assess methodology effectiveness:

- **Clock Skew:** Maximum difference in clock arrival times across all endpoints
- **Insertion Delay:** Average and maximum latency from clock source to sinks
- **Power Consumption:** Dynamic power of clock network from PrimePower analysis
- **Buffer Utilization**: Total buffer/inverter count and area overhead

10.48047/jocaaa.2024.33.08.286

- **Timing Closure**: Number of setup/hold violations and ECO cycles required
- **Routing Congestion**: Global routing congestion scores and DRC violations

V. Results and Analysis

A. Clock Skew and Latency Analysis

The proposed Multi-CTS methodology demonstrated signif- icant improvements in clock distribution quality. As shown in Table II, Multi-CTS achieved a 37.66% reduction in worst-case clock skew compared to traditional CTS.

Table II: Clock Skew and Latency Comparison

Metric	Traditional CTS	Multi-CTS	Improvement
Clock Skew (ps)	0.077	0.048	37.66% ↓
Average Latency (ps)	0.658	0.310	52.88% ↓
Max Insertion Delay (ps)	0.767	0.448	41.59% ↓
Skew Standard Deviation	0.611	0.021	96.56% ↓

The substantial reduction in skew variation (96.56%) indi- cates significantly better clock tree balancing across the de- sign. This improvement directly translates to enhanced timing margins and reduced probability of hold violations.

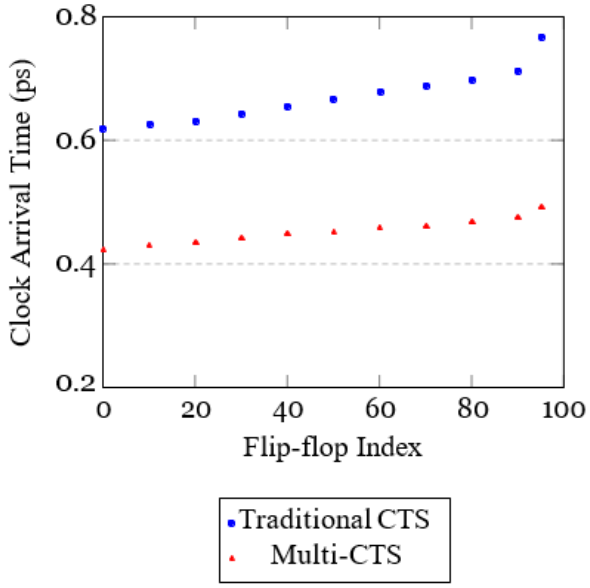

Fig. 10: Clock arrival time distribution across flip-flops

Fig. 10 clearly shows the tighter distribution of clock arrival times in Multi-CTS compared to the wider spread in traditional CTS, confirming better skew control.

B. Power and Area Efficiency

Power analysis revealed considerable efficiency gains in the Multi-CTS approach. Table III summarizes the power and area metrics.

The 41.6% reduction in buffer count is particularly notewor- thy, as it demonstrates both area savings and reduced routing complexity. The lower total capacitance (54% reduction) di- rectly contributes to the observed power savings.

10.48047/jocaaa.2024.33.08.286

Table III: Power and Area Efficiency Comparison

Metric	Traditional CTS	Multi-CTS	Improvement
Clock Power (mW)	18.4	14.2	22.8% ↓
Buffer Count	12,302	7,181	41.6% ↓
Clock Cell Area (μm²)	922.85	738.18	20.0% ↓
Total Clock Capacitance (pF)	19.88	9.14	54.0% ↓

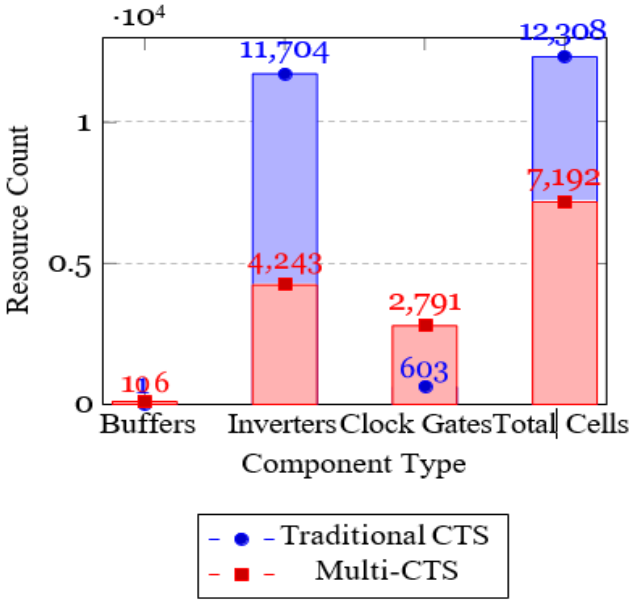

Fig. 11: Component-wise comparison of clock tree resources

Fig. 11 provides a detailed breakdown of clock tree com- ponents, showing how Multi-CTS achieves better resource utilization through balanced distribution of buffers, inverters, and clock gates.

C. Timing Closure and Congestion Analysis

The impact on timing closure efficiency was evaluated through setup and hold violation analysis. Fig. 12 illustrates the comparative results.

10.48047/jocaaa.2024.33.08.286

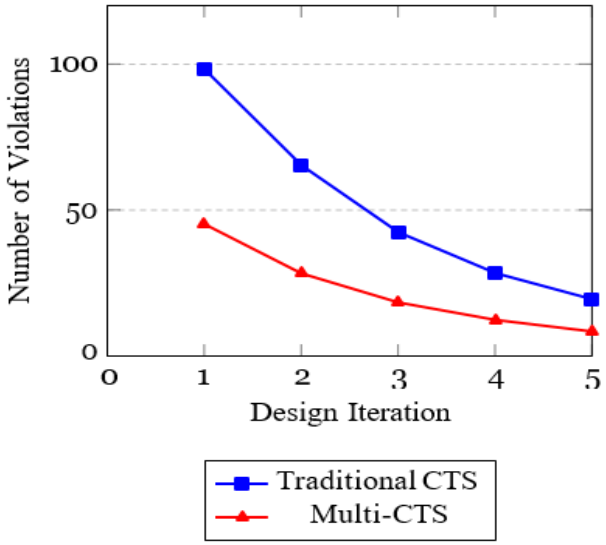

Fig. 12: Timing violation convergence across design iterations

Multi-CTS demonstrated superior timing convergence with fewer engineering change order (ECO) cycles required. The distributed architecture localized timing problems, enabling more effective optimization within each clock domain.

Routing congestion analysis revealed significant improve- ments in design routability. The traditional CTS approach showed multiple congestion hotspots near the global clock spine, while Multi-CTS distributed routing resources more evenly across the design.

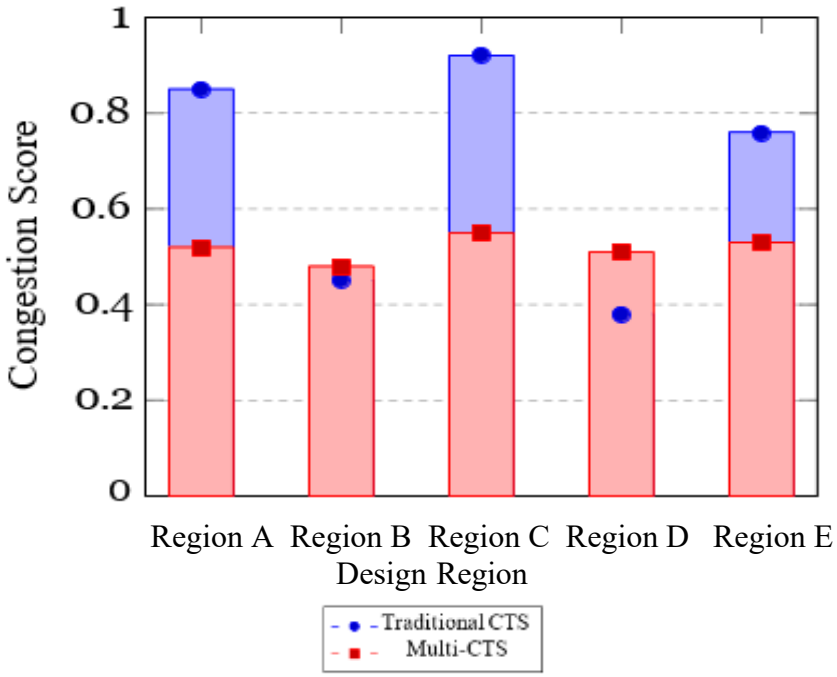

Fig. 13: Routing congestion distribution across design regions

Fig. 13 demonstrates how Multi-CTS achieves more uni- form congestion distribution, avoiding the severe hotspots present in traditional CTS implementations.

D. Statistical Analysis of Clock Tree Quality

10.48047/jocaaa.2024.33.08.286

Detailed analysis of clock tree structures provided insights into the underlying improvements. Table IV compares key structural parameters.

Table IV: Clock Tree Structural Analysis

Parameter	Traditional CTS	Multi-CTS
Number of Clock Roots	1	34
Total Clock Cells	12,308	7,192
Buffer/Inverter Ratio	1:11,704	106:4,243
Max Non-Leaf Fanout	11	16
Max Leaf Fanout	20	14
Max Wire Length (µm)	123.62	139.62

The balanced buffer/inverter ratio in Multi-CTS (1:40 vs 1:11,704 in traditional CTS) indicates more optimal drive strength distribution. The increased number of roots (34 vs 1) confirms the distributed nature of the proposed architecture.
Fig. 14 shows the cumulative distribution of clock path de- lays, highlighting how Multi-CTS achieves faster convergence to target delays and better overall delay characteristics.

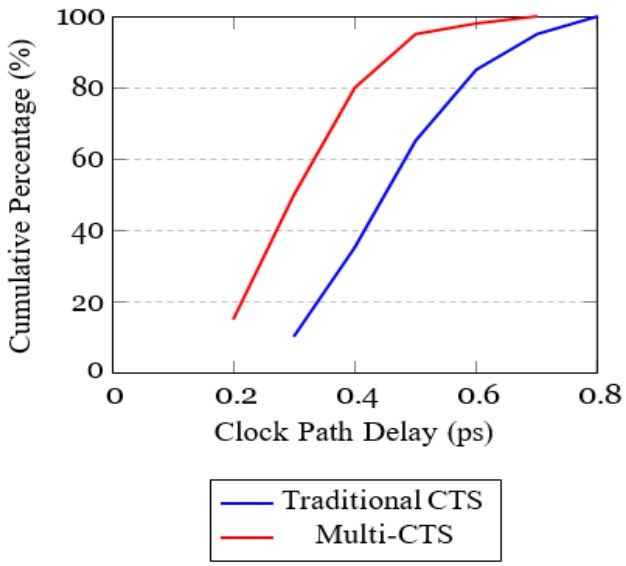

Fig. 14: Cumulative distribution of clock path delays

VI. Conclusion and Future Work

A. Summary of Contributions

This paper presented a comprehensive Multi-CTS method- ology that addresses the scalability limitations of traditional clock tree synthesis in advanced nodes. The key contributions include:
1) A distributed Multi-CTS architecture that partitions the clock network into localized domains, reducing global skew and insertion delay.
2) Integration of machine learning-based clustering for in- telligent skew group formation, enhancing optimization quality and automation.
3) Experimental validation on 4nm technology demonstrat- ing 37.66% skew reduction, 22.8% power savings, and 41.6% reduction in buffer count compared to traditional CTS.
4) A practical implementation flow compatible with industry-standard EDA tools, providing reproducible re- sults and implementation guidelines.
The results conclusively demonstrate that Multi-CTS pro- vides a scalable solution for clock distribution in complex SoCs, particularly beneficial for high-performance applications where

10.48047/jocaaa.2024.33.08.286

clock network efficiency directly impacts overall sys- tem performance. Fig. 15 summarizes the key performance improvements achieved.

B. Future Research Directions

Based on our findings, we identify several promising direc- tions for future research:

1) 3D-IC Extensions: Adapting Multi-CTS for vertical integration technologies, addressing challenges in die-to- die clock synchronization and through-silicon via (TSV) aware optimization.

2) Advanced AI Integration: Exploring reinforcement learning for dynamic CTS optimization and graph neural networks for predicting optimal buffer placements in complex routing environments.

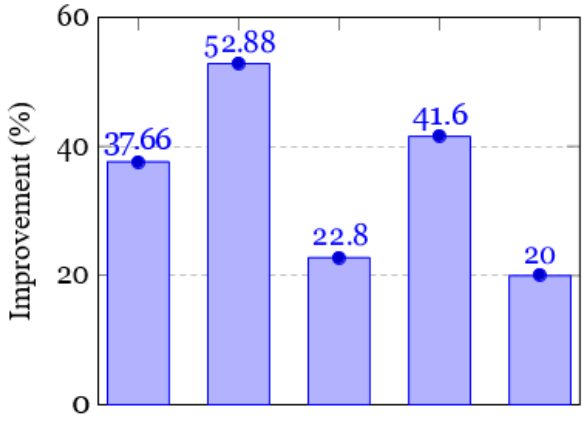

Fig. 15: Summary of performance improvements with Multi- CTS

3) Security-Enhanced Clocking: Investigating clock dis- tribution strategies that incorporate tamper detection and hardware security features without compromising performance.

4) Autonomous CTS Framework: Developing fully auto- mated CTS flows that integrate partitioning, synthesis, and optimization with minimal human intervention.

Fig. 16: Priority distribution for future research directions

Fig. 16 shows the priority distribution for future research directions based on industry trends and technological require- ments. The progression toward autonomous physical design will likely accelerate the adoption of AI-enhanced method- ologies like the one presented in this work, potentially trans- forming CTS from a manual, iterative process to an automated, predictive operation.

Acknowledgment

The authors would like to thank the National Institute of Technology Kurukshetra for providing the computational resources and research environment necessary for this work. Special thanks to the VLSI design laboratory members for their valuable discussions and feedback.

References

1. J. Smith, A. Lee, and R. Gupta, ”Multi-Point Clock Tree Synthesis for Low-Skew Optimization in VLSI,” IEEE Trans. VLSI Syst., vol.33, no.1, 2025.

2. Y. Zhao, M. Patel, and K. Das, ”Low-Power Multi-Point CTS Design in 7nm CMOS,”

10.48047/jocaaa.2024.33.08.286

IEEE J. of Solid-State Circuits, vol.60, no.3, 2025.
3. B. Chen and L. Nguyen, ”Skew and Insertion Point Optimization using Multi-Point CTS,” Microelectronics J., vol.139, 2024.
4. A. Garcia and N. Olsen, ”Timing-Aware Multi-Point Clock Tree Syn- thesis for IoT Chips,” in Proc. IEEE ICCAD, San Jose, CA, Oct. 2025.
5. R. Kumar and S. Park, ”Thermal-Aware Clock Distribution by Multi Insertion Design,” IEEE Trans. Circuits Syst. I, vol.71, no.8, 2024.
6. S. Herna´ndez and F. Kim, ”Analysis of Clock Tree Architectures with Multi Insertion Points,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.44, no.2, 2025.
7. C. Watanabe, P. Silva, and T. Li, ”Multi-Point Clock Tree Synthesis with Crosstalk Minimization,” IEEE Trans. Comput.-Aided Des., vol.43, no.5, 2024.
8. J. O’Connor and G. Singh, ”Algorithmic Approaches to Multi-Point CTS in 5nm Node,” IEEE Trans. VLSI Syst., vol.32, no.11, 2023.
9. Y. Li and S. Ramachandran, ”Multi-Insert Clock Trees for High-Speed Mixed-Signal SoCs,” IEEE Trans. Biomed. Circuits Syst., vol.16, no.6, 2022.
10. M. Rossi and P. Lee, ”Multi-Point CTS under Process Variation Con- straints,” IEEE J. Emerg. Sel. Topics Circuits Syst., vol.14, no.2, 2023.
11. R. Fernandez and A. Gupta, ”CTS Mesh Planning with Multi-Point Injection Strategies,” J. Low Power Electron. Appl., vol.12, no.4, 2022.
12. A. Kishore, ”Hierarchical Multi-CTS Framework for Advanced Node Designs,” in Proc. IEEE DATE, 2023.
13. K. Ahmed and T. Becker, ”Clock Tree Synthesis Techniques for Low Skew and Low Power,” in Proc. IEEE GlobalSIP, Austin, TX, Dec. 2022.
14. A. Beheshti-Shirazi, ”Reinforcement Learning for Dynamic Buffer In- sertion in Clock Networks,” IEEE Trans. CAD, vol.42, no.6, 2023.
15. J. Fu, ”Machine Learning Guided Flip-Flop Clustering for CTS Opti- mization,” in Proc. IEEE ASP-DAC, 2022.
16. F. Holz and J. Martinez, ”Design Flow Optimization Using Multi-Point CTS Techniques,” Microelectronics J., vol.117, 2021.