rice2024 Proceedings of the Ninth International Conference on Research in Intelligent Computing in Engineering pp. 97–102 DOI: 10.15439/2024R06 ISSN 2300-5963 ACSIS, Vol. 42

Dynamic Clock Tree Balancing Algorithm: Achieving Enhanced Performance Efficiency in Asic Design

J. Praveenkumar
Research Scholar, Bharath Institute of Higher Education and Research, Chennai, TamilNadu India
Email: praveencando91@gmail.com

G. Sudhagar
Associate Professor, Department of ECE, Bharath Institute of Higher Education and Research
Chennai, TamilNadu India
Email: sudhagarambur@gmail.com

***Abstract*—The creation and application of a novel clock tree balancing method that dynamically optimizes the clock distribution network in ASIC designs is the main focus of this project's effort. Improving the timing constraints, lowering clock skew, cutting down on power usage, and raising the ASIC's overall performance are the main goals. As technology has developed, ASIC designs have gotten more intricate, incorporating billions of transistors onto a single chip. It gets harder to create an efficient and well-balanced clock distribution network as the number of gates and design size increase. On the other hand, problems like clock skew, clock jitter, and excessive power consumption can arise when this clock signal is applied to every sequential part of a big and intricate ASIC design. For ASIC designs, it entails creating and constructing a dynamic clock tree balancing algorithm. Using real-time data, the program will improve clock distribution, improving timing constraints and lowering power usage. It will be evaluated against conventional techniques, verified on actual ASIC designs, and recorded for a dissertation. The project's goal is to improve ASIC design techniques to produce chips with high performance and low power consumption.**

***Index Terms*—Application-Specific Integrated Circuit, Dynamic Clock Tree Balancing, Skew-aware-source-pulling**

I. Introduction

AN INNOVATIVE and adaptive algorithm tailored for Application-Specific Integrated Circuit (ASIC) designs is implemented. The dynamic clock tree balancing algorithm will possess the capability to continuously optimize the clock distribution network in real-time.By leveraging up-to-the- moment data, the algorithm will actively work to reduce clock skew within the ASIC, consequently leading to a significant improvement in timing precision. The dynamic nature of the algorithm ensures that it can adapt to the evolving needs and demands of the system, making it a cutting-edge solution for enhancing the overall performance of ASIC designs. The clock signal is disseminated throughout the design in the form of a tree, with the clock source representing the root and the leaves representing the sequential devices that the clock signal triggers. A balanced clock tree is one in which all elements of the design get the clock signal nearly simultaneously. An essential phase in the implementation process is clock tree synthesis, which involves organizing and routing clock tree components. The implementation of specific cells and routing strategies guarantees a strong clock tree.

The clock remains in the optimal mode even when the standard cells and the macros are positioned in a fixed, optimized location. The clock enters the propagated mode because the clock signals carry out the data transfer between the various functional units on the chip. Every sequential element's clock input needs to be in sync for a design to meet setup and hold criteria. Due to the large number of sequential elements in the design, a single clock net cannot drive them all. The clock signals are distributed from a common point to each element's clock pin via the clock distribution network. The goal is to minimize arrival time uncertainty while balancing clock skew using dynamic clock tree balancing algorithm. Dynamic Clock Tree Balancing (DCTB) is a crucial aspect of Very-Large-Scale

Integration (VLSI) design, aimed at optimizing the distribution of clock signals across the chip to minimize clock skew and improve overall chip performance like power consumption and timing. The performance of the chip is directly impacted by the design of the clock networks, which is a crucial component of the design process. Clock signals synchronize operations fundamentally in contemporary integrated circuits. Timing errors and poor chip performance can result from clock skew, which is the fluctuation in clock signal arrival times at various locations on the semiconductor. To make sure that the updated clock tree satisfies timing requirements and decreases skew without introducing new problems, clock tree balancing with dynamic buffer delay adjustment necessitates meticulous verification and testing. A hierarchical network of buffers, wires, and clock distribution components known as the clock tree is used in VLSI design to distribute clock signals to various sequential units (such as flip-flops) throughout the chip.This enhances performance,cuts down on power usage, and decreases setup and hold time violations. A runtime or post-fabrication technique DCTB continuously examines and modifies the clock tree based on real time data.

On the basis of chip-level needs and critical paths, define skew thresholds and limitations which measure clock skew continuously from the main clock source to the leaf nodes, covering different locations in the clock tree hierarchy. The

DCTB algorithm is activated when skew reaches predetermined levels or violates setup/hold time restrictions. DCTB provides real-time adjustments to the clock tree in response to altering operating circumstances. Clock distribution and timing are susceptible to external influences, including temperature changes and voltage swings. DCTB supports sustaining peak performance in dynamic environments.One of the main objectives revolves around enhancing the timing constraints within ASIC designs.

The primary focus is on minimizing clock skew and jitter, which are notorious culprits for timing violations. Our goal is to meticulously refine these constraintsto ensure that setup and hold times for sequential elements are consistently met. By achieving this objective, we anticipate a notable reduction in the likelihood of timing violations occurring in the ASIC, thereby enhancing the reliability and predictability of the design. The other aspect is to embed power optimization techniques directly into the algorithm. The overarching objective is to create a clock distribution network that not only excels in timing precision but also excels in power efficiency. By doing so, we anticipate a significant reduction in the overall power consumption of ASIC designs. This aligns perfectly with contemporary principles of energy-efficient design, making our project not only technologically advanced but also environmentally conscious. The implementation of region based dynamic clock tree balancing algorithm is used in the proposed system which is one of the clock tree optimization techniques. The inter region synchronization technique is used to synchronize clock signals in every region. In each region Skew-aware-source-pulling (SASPO) is implemented to minimize the clock skew.

To provide a comprehensive perspective on the efficacy of our dynamic clock tree balancing algorithm, we will conduct a thorough evaluation and comparison against traditional static clock tree synthesis methods. This comparison will encompass various facets, including timing accuracy, power efficiency, and scalability. By directly contrasting the two approaches, we aim to showcase the superiority of our dynamic algorithm, underlining its potential to improve clock tree building in the fieldof ASIC design. The validation of our algorithm will be a pivotal part of this project. We plan to test its performance on real-world ASIC designs, ensuring that it thrives in practical applications. Furthermore, the insights gained from this project will not remain confined but will be shared with the broader ASIC design community. We intend to disseminate our findings. through research publications and present our discoveries at prominent conferences, thereby making substantial contributions to the field's body of knowledge and practice.

II. Literature Survey

A. Clock Tree Synthesis Techniques for Optimal Power and Timing Convergence in SoC Partitions

The studies were conducted in a SoC partition utilizing the physical design flow, which is implemented in 14nm technology. Using several optimization techniques at each design stage, it focuses mainly on timing, power, and area optimization. The analysis of effective CTS methods for timing convergence and optimal power in SoC partition is the main goal of this work. Multisource Clock Tree Synthesis and Multibit FlipFlop use are the approaches used for CTS with clock tree awareness. Through the reduction of latency and skew as well as the improvement of the clock distribution, multi-source CTS enhances timing of design. The number of sequential cells has decreased due to the use of multi-bit flip-flops, which has creased he overall power consumption of the clock network and design area.

B. An Efficient Clock Tree Synthesis Method in Physical Design

In this study, a low clock skew solution for the clock tree synthesis (CTS) design flow used in the mainstream industry is proposed. This article presents a method that greatly reduces the clock skew with area cost and placement time. Regarding the notable difference in throughput time, this is explained by the fact that this program runs on a single CPU core, but clock tree generation (CTG) can be executed concurrently on several smaller pseudo clock sources on multiple workstations. From this method there is a large time improvement, when the tool runs simultaneously on numerous workstations. Furthermore, this paper overcomes many flaws like excessive manual analysis because it is relevant to the mainstream industry's CTS design procedure.

C. A Clock Tree Synthesis Flow Tailored for Low Power

The purpose of this study is to provide real-world experience with poweroptimized clock tree construction, clock tree synthesis (CTS) target optimization, and quality of results (QoR) tracking. The studies that will be described were carried out on a mixed-mode architecture that was meant for a 55nm CMOS technology node. It features various power domains with more than 100K registers. Simulation findings show that the approach described here can save up to 20% of clock tree power. Using a standalone CTS tool in the flow can result in a clock tree power decrease of up to 20%. The design is put into the standalone tool that provides a dedicated CTS engine with potentially improved quality. This technique involves altering a standard P&R flow with integrated CTS capability. The tool's output is meant to be easily loaded into the standard P&R tool once CTS is finished.

D. Clock Tree Optimization Methodologies for Power and Latency Reduction

This paper will provide an overview of several commonly used clock structures, with a focus on the practical use of H-Tree and standard clock tree structure. The implementation was carried out on a real-time database using a 16nm technology node with 1.4 million instances and an operating frequency of 537MHz. The types of cells and routing that are utilized to create the H-Tree clock structure, the customization of the HTree clock structure based on sink distribution, and the numerous scenarios that need to be considered when

selecting this approach are also covered in this paper. Clockqor is compared between this method and the traditional clock tree structure, and the results indicate a positive improvement. The more conventional clock distribution networks might be replaced by these H-Tree clock networks. Without affecting the signal's properties, the suggested clock tree optimization techniques lower power dissipation.

There is a decrease in inductive noise due to the interconnects' inductive nature. In summary, the H-Tree structure will yield superior power, latency, and skew when the requirements are tight (80–100 ps for skew, <500 ps for latency, and more than 10,000 sinks).

E. An efficent clustering algorithm for low power clock tree synthesis

This paper provides a clustering approach for the local clock tree's power minimization, which is demonstrated to be equal to the tree's interconnect capacitance minimization. Clustering is used to find the clock buffers needed to synchronize a group of sequential and their positions. A cluster indicates that every sequential in the cluster is driven by a clock buffer. When capacity constraints are not used, the clustering algorithm guarantees the optimality of the solution by estimating the interconnect capacitance using the minimal spanning tree (MST) metric. Next, to reduce the limitations related to delay, slope, and skew, clock nets are routed, and buffers are sized. The paper compares the clock trees derived from our clustering method against the competitor alternatives across multiple blocks of a 65 nm microprocessor design. Method reliably increases the clock tree capacitance by up to 21%, as seen by the comparison. From the above referenced papers, the proposed method varies with the clustering method implemented dynamically as the design and technology varies, this will in turn reduce the number of iterations required to attain closure. Hence the proposed method could be time-saving and reliable way of dealing with the design with huge number of sequential elements being placed in appropriate location considering the clock root location, net length, skew limitations, and latency requirements.

III. Proposed Work

An ASIC is a sort of integrated circuit (IC) that is designed specifically for a given task or application as opposed to general-purpose ICs like a microprocessor. ASICs are created to carry out a specific function or group of functions with the highest levels of speed, efficiency, and power optimization. An ASIC's physical area, power usage, and performance are all impacted by its clock frequency. Higher clock frequencies often translate into both better performance and more power usage. DCTB aims to optimizes clock signal distribution across the chip to reduce clock skew and enhance timing and power consumption. One of the most important aspects of the design process is the clock network design, which directly affects the chip's performance. A clock tree with proper balance can increase semiconductor production yield. DCTB can assist in the production of more functioning chips and a decrease in the quantity of defective ones by minimizing timing violations and guaranteeing that chips fulfill their timing limitations.

To optimize power consumption, DCTB algorithms can dynamically modify the clock tree topology. It helps reduce power consumption, which is important for energy-efficient designs and battery-powered devices, by cutting down on needless power

dissipation in clock buffers and interconnects.

The proposed system implies Region-based dynamic clock tree balancing is a clock distribution optimization technique that divides a chip into different regions and balances the clock tree within each region independently. The system divides the chip into regions based on various criteria, such as functional blocks, IP cores, or logical groupings which contain a collection of sequential components, such as flip-flops in each region. Region-based balancing algorithms are scalable, which makes it appropriate for

intricate semiconductor designs. Region-based balancing assists in efficiently managing complicated clock distribution networks as chip sizes and complexity rise. Limited alterations and optimizations using region-based methods without affecting the entire chip is done. As a result, there is less chance of introducing further problems in other areasof the chip, which helps simplify the design process. Different regions of a chip may experience process variations differently. By individually modifying the clock tree inside each region, In Fig 1 region-based techniques can adjust to these differences while still meeting timing requirements.

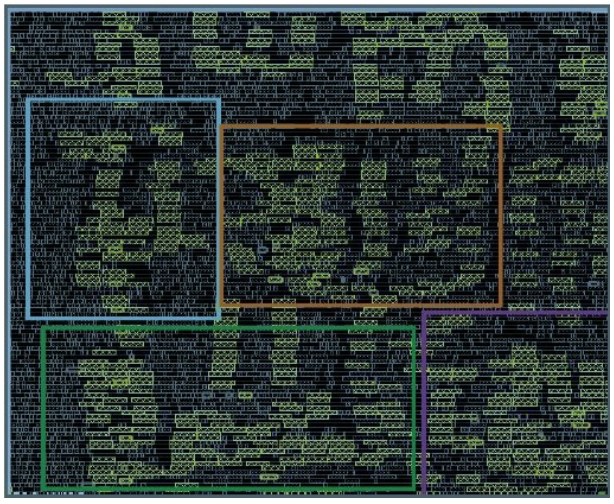

Figure 1: Region based Clock structure

In the Fig 2, the yellow highlighted cells are flops/sinks, and these are categorized based on the hierarchy, skew achieved and clock roots. Each domain will be driven by separate clock cell and the skew will be balanced within these sink flops and also make sure the timing is met within these interrelated flop paths.

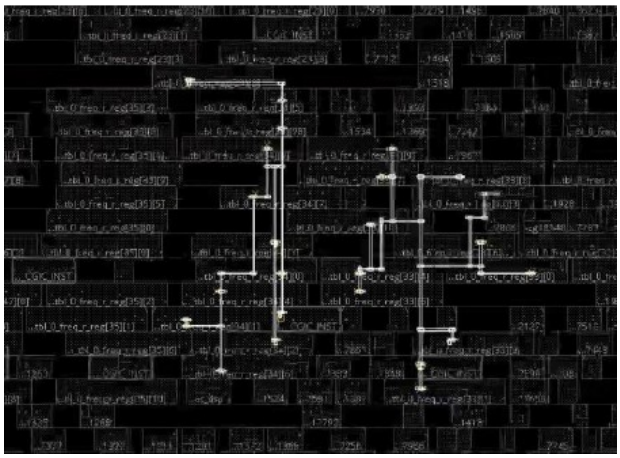

Figure 2: Clock structure based on Sink.

The clock gets distributed to all the flops in a region from the approximate mean point on the region to make sure the latency to reach each flop in a region is almost same which in turn keeps skew in control.

These components need synchronized clock signals. For each region, distinct clock trees are made for each zone to effectively disperse the clock signal within that region. A region's clock tree has its own root, main clock source, and special buffers. For continuously monitoring and analyzing the clock skew particular to each location, the system implements skew monitoring techniques there for both intra-regional and interregional skew can be monitored by these monitors.

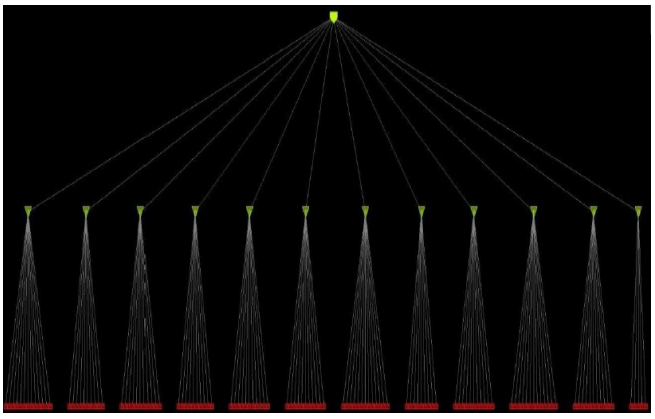

Figure 3: Clock Tree structure.

In Fig 3 Clock tree sinks grouped based on region and being driven by common clock buffer is depicted in the above image. The inter region synchronization technique is utilized to synchronize clock signals between various areas. Dynamic clock tree balancing (DCTB) uses the inter-region synchronization technique to ensure that clock signals migrate between various areas of a chip smoothly and with the least amount of clock skew. The clock signals are switched between various clock domains or regions at the edges of these regions, if not adequately controlled, these boundaries have the potential to be causes of clock skew. Inter-region clock skew may affect clock signals when they move between regions, which can cause setup and hold time violations or other timing problems. Clock Gating/Buffering at Region Boundaries is used as an inter-region synchronization technique to reduce inter-region clock skew. To avoid clock skew clock gating cells or buffers are placed at the region boundaries to synchronize the phases of the clock signal as it crosses over. To make sure that the inter-region synchronization mechanisms efficiently reduce skew and do not introduce additional timing infractions, they must be thoroughly verified and tested. By including suitable delays or buffers at the area borders, cross-region skew can be monitored and corrected appropriately. A hierarchical control system is designed that keeps an eye on how the clock tree balancing works in each region. This control system tracks the global skew, responds to requests from specific regions, and regional modifications should not compromise chip-level performance.

In the Fig 4, the inter region skew in maintained and the timing violations are met, by pulling and pushing the sinks from the average attainable latency.

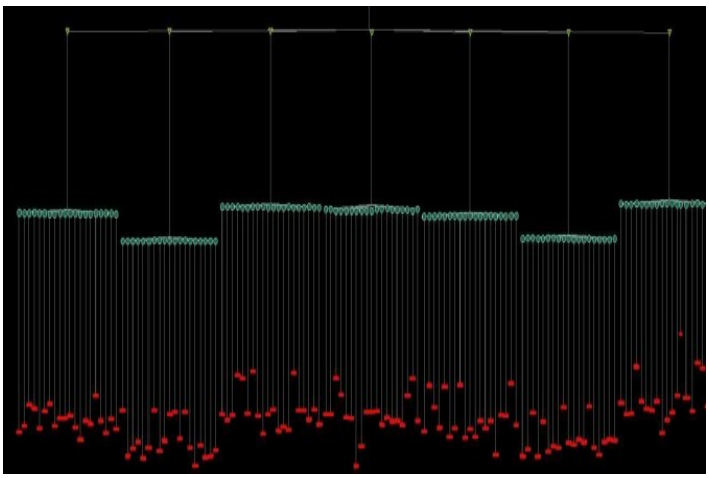

Figure 4: Clock Tree debugger.

Considering power limitations and dynamic balancing which affects power usage in each zone so energy-saving technique, such as power-down modes used for unused areas.To guarantee that the region-based dynamic clock tree balancing solution satisfies timing and performance specifications, thoroughly validate it using simulation and physical design tools.

While region-based balancing addresses local clock skew issues, incorporate chiplevel optimization techniques to ensure that global skew requirements are also met. This technique allows for customization of region boundaries and balancing parameters to accommodate different design requirements and goals.

The initial clock tree synthesis (CTS), which creates the clock tree structure using the logical netlist and chip layout, is the first step in the physical design process. In order to continuously measure clock skew across various regions of the semiconductor. Primary clock sources are often located at the top level of the clock tree, feeding down to lower levels of hierarchy. From the primary clock source to leaf

nodes, real-time monitoring circuits continuously assess clock skew at different points in the clock tree hierarchy. The design's skew limits and thresholds are set. The permissible skew limitations for different clock domains and pathways can be specified by these criteria. By comparing the measured clock skew against the established thresholds, Skewaware-source-pulling (SASPO) is one of the DCTB algorithms which identifies clock skew breaches, and the algorithm starts corrective steps when skew violations happen. The source-pulling method is used by SASPO to correct clock skew and this entails deliberately modifying the clock tree's delay components (buffers).

The primary goal of SASPO is to minimize clock skew within a digital integrated circuit. The clock skew describes the variance in clock signal arrival times at various locations on the chip. As a result of lowering clock skew, SASPO makes it possible for sequential components like flip-flops and latches to all receive clock signals at the same time, preventing setup and hold time infractions and enhancing chip performance.

To reduce skew, buffers can be added or changed to slow down or speed up the propagation of the clock signal along particular pathways. Based on the real-time feedback from the monitoring circuits, the algorithm dynamically adds or removes buffers. The main objective of source-pulling is to balance the timing of clock signals at various locations on the chip. To ensure that clock signals reach consecutive elements (like flip-flops) simultaneously, SASPO selectively modifies the latency of buffers. This decreases clock skew. To achieve exact skew management, SASPO may also fine-tune the delay parameters of individual buffers which is an iterative procedure. To reduce skew violations, it constantly analyzes and modifies the clock tree. While the main focus of SASPO is local optimization within the clock tree, it may also apply global optimization strategies to guarantee that chip-level skew criteria are met. SASPO implementations frequently take restrictions into account and work to reduce the extra power that buffer modifications needs. To foresee skew variations and make proactive adjustments, some SASPO solutions may include sophisticated technologies like predictive modeling and machine learning.

IV. Data Flow

In this section, the data flow and flow of the proposed methodology are explained. The database from the timing driven placement is given to the next step, which is the first step of the Dynamic Clock tree balancing. Firstly, the clock tree building is done in cluster mode, where the Clock transition is only fixed to determine the minimum insertion delay that can be achieved from the clock groups and skew groups which determines the maximum pulling clock sink. The next step is to analyze the clock tree structure and determine the root cause of this clock delay achieved. Hence the clock tree balancing is run in trial mode and the changes in the clock tree spec file are modified and provided for the CTS run in the next iteration. Once we see the violations, accordingly changes have to be made in the spec and again CTS building is carried forward, until we see the desired insertion delay and skew.

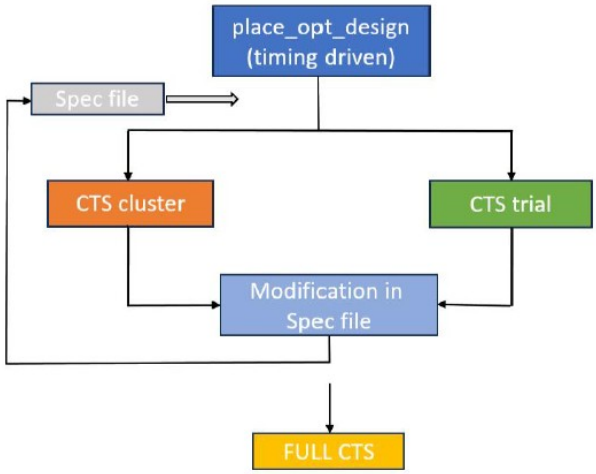

Figure 5: Clock tree building structure.

In the Fig 5 Data Flow, it is evident that the Input from Placement and the spec file generated from the SDC and user constraints is provided for the trial and cluster mode CTS to debug and find the requirements of the block which is iterated till we arrive at the desired output. In this iteration the clusters are organized based upon the skew groups, clock groups, hierarchies it belongs to and physical location of the cluster, which is done dynamically in the flow.

V. Results

A. Skew Comparison

In Table 1 The variation in time taken by each clock signal to reach the various sinks or clock pins on the clock tree branches is known as clock skew. The clock skew should be zero in ideal circumstances. The result obtained.

Table 1: Skew Comparison with clock trees

Clock Tree Type	Launch Latency (ps)	Capture Latency(ps)	Local skew (ps)
Conventional CTS	-300	340	640
Dynamic CTS	40	254	214

B. Power Comparison

In the Table 2 power comparison with clock trees Due to the existence of a clock mesh structure, dynamic CTS has a little higher clock network power than conventional CTS. It is also less when compared to a pure mesh network. When

Table 2: Power Comparison with clock trees

Clock Tree Type	Dynamic Power (mW)	Leakage Power (mW)	Total Power (mW)
Conventional CTS	34	29.12	63.94
Dynamic CTS	44.89	47.05	87.11

compared to the enhanced outcomes in other design features, the added power consumption is a minimal factor.

C. Timing Comparison

Dynamic timing analysis is carried out using a tool for timing analysis. To start the simulation runs and then perform the timing analysis of the design for ensuing reports of dynamic timing and analysis, the fundamental technique is automation, which is featured in PnR tools. In the Table 3 mentioned setup timing comparison.

TABLE 3: SETUP TIMING COMPARISON WITH CLOCK TREES

Clock Tree Type	WNS (ps)	TNS (ps)
Conventional CTS	-42	-117
Dynamic CTS	-17	-85

VI. CONCLUSION

After the successful implementation of the Dynamic Clock Tree Balancing Algorithm-Achieving Enhanced Performance Efficiency In Asics Design.

These results were in accordance with the detailed explanation of contents present in chapter.

VII. ACKNOWLEDGMENT

I wish to convey my special thanks to my project guide, Dr.G.Sudhagar, from Bharath Institute of Higher Education and Research for his consistent support, timely help, and valuable suggestions during the entire period of my Research.

I extend my sincere gratitude to all the teaching and non-teaching staff members of Bharath Institute of Higher Education and Research who helped me during the entire course.

REFERENCES

[1] (Clock Tree Synthesis Techniques for Optimal Power and Timing Convergence in SoC Partitions IEEEXplore:https://ieeexplore.ieee.org/document/9016727

[2] An efficient clock tree synthesis method in physical design | IEEE Conference Publication IEEEXplore:https://ieeexplore.ieee.org/document/5394159

[3] A Clock Tree Synthesis Flow Tailored for Low Power https://www.design-reuse.com/articles/33873/clock-tree-synthesis-flow-tailored-for-low-power.html

[4] Clock Tree Optimization Methodologies for Power and Latency Reduction SemiconductorDigest : https://www.semiconductor-digest.-com/clock-tree-optimizationmethodologies-for-power-and-latency-reduction/

[5] An efficent clustering algorithm for low power clock tree synthesis: https://www.researchgate.net/publication/220915569_An_efficent_-clustering_algorithm_for_low_power_clock_tree_synthesis

[6] Finding placement-relevant clusters with fast modularity-based clustering https://www.researchgate.net/publication/330493649_Finding_-placementrelevant_clusters_with_fast_modularity-based_clustering