ECSE 6320

# Hybrid Clock Network Design with Region-Adaptive Local Meshes

Gavin Garrison Electrical Engineering

## ABSTRACT

Clock distribution is a dominant contributor to power consumption and timing uncertainty in modern multi-GHz system-on-chip designs. Classical clock topologies represent opposing extremes: H-trees are power-efficient but increasingly sensitive to process variation and resistive imbalance, while clock meshes provide excellent skew robustness at the cost of large wire capacitance and switching power. This project investigates a hybrid clock architecture that combines a global H-tree backbone with localized clock meshes inside tiled regions of the die. The central question is how much local meshing is required to recover most of the skew and robustness benefits of a full mesh, and whether region-adaptive meshing policies provide additional gains beyond a uniform hybrid design.

I developed an analytical clock network simulator based on Modified Nodal Analysis (MNA) in MATLAB. The model constructs a global H-tree feeding multiple regions, each optionally containing a local grid mesh with tunable density. Transient analysis is performed to compute clock arrival times and skew, while dynamic power is estimated using a $CV^2f$ formulation. Monte Carlo variation is applied to interconnect resistance, capacitance, and driver strength to evaluate robustness. Uniform mesh density sweeps and region-adaptive mesh policies are compared using skew–power Pareto fronts, skew robustness distributions, and spatial arrival heatmaps.

The results show that sparse hybrid meshing collapses clock skew by more than three orders of magnitude relative to an H-tree baseline while consuming significantly less power than a full mesh. Most of the skew reduction occurs at low mesh densities around p_r ≈ 0.25, beyond which diminishing returns dominate. Region-adaptive meshing matches uniform hybrid performance but does not outperform it in symmetric layouts, indicating that adaptive policies are most valuable in heterogeneous or asymmetric clock load scenarios. Overall, this work demonstrates that architectural-level clock modeling can identify near-optimal hybrid designs early in the design process and avoid unnecessary power overhead from over-meshing.

## 1. INTRODUCTION

### 1.1 Motivation

As system-on-chip designs push toward higher clock frequencies and larger die sizes, clock distribution increasingly limits both power efficiency and timing closure. Clock networks toggle every cycle and span the entire die, making them one of the largest contributors to dynamic power consumption. At the same time, shrinking interconnect dimensions and rising resistivity amplify RC delays and sensitivity to process variation. The result is that maintaining low skew across a large chip has become progressively harder, even for geometrically symmetric clock trees.

Historically, H-trees provided an elegant solution by equalizing path lengths from the clock source to all sinks. However, modern interconnect scaling breaks the assumption that equal geometric length implies equal electrical delay. Local resistance variation, parasitic capacitance differences, and driver non-idealities all introduce skew that accumulates along long clock paths. Clock meshes address this by introducing redundancy and multiple current paths, effectively averaging out local mismatches. The drawback is power. Dense meshes add large wire capacitance that toggles every cycle, which is increasingly unacceptable in power-constrained designs.

This tension between skew robustness and power efficiency motivates hybrid clock architectures that selectively apply meshing only where it provides the most benefit.

### 1.2 Problem Statement

The underlying problem addressed in this work is how to balance clock skew, robustness to variation, and power consumption at an architectural level. Specifically:

- How much local clock meshing is required to recover most of the skew reduction benefits of a full mesh when starting from an H-tree backbone?
- How does clock skew scale with mesh density, and where do diminishing returns appear?
- Do region-adaptive mesh density policies provide measurable improvements over uniform hybrid designs under realistic assumptions?
- How robust are these designs under interconnect and driver variation?

Answering these questions requires a modeling approach that is detailed enough to capture RC effects and variation, yet simple enough to support large design sweeps and architectural insight.

### 1.3 Prior Work and Why This Is Hard

Clock distribution has been studied extensively for decades. Early work formalized clock tree synthesis and skew minimization strategies, while later work highlighted the impact of interconnect variation and power constraints. Clock meshes gained popularity in high-performance processors because of their robustness, but their power cost has limited widespread adoption. Hybrid clock networks were proposed to combine the strengths of trees and meshes, but in practice their design often relies on heuristics or manual tuning.

A key challenge is that many prior studies either focus on signoff-level physical design details or abstract the clock network so aggressively that RC effects are lost. This makes it difficult to answer architectural questions such as how much meshing is actually necessary. Another difficulty is evaluation consistency. Small differences in assumptions about driver strength, capacitance, or reference scaling can lead to misleading comparisons.

This project addresses these challenges by holding modeling assumptions constant and sweeping mesh density in a controlled manner using an analytical RC framework.

### 1.4 Key Ideas and Contributions

The main contributions of this work are:

1. An analytical clock network simulator based on Modified Nodal Analysis that models H-trees, local meshes, and their interaction under transient excitation.

2. A systematic sweep of uniform mesh density that reveals a clear skew–power tradeoff and identifies a low-density hybrid knee where most skew reduction is achieved.

3. A region-adaptive meshing policy evaluated under the same modeling framework, with honest comparison to uniform designs.
4. Monte Carlo analysis demonstrating how hybrid designs suppress skew tails under variation.
5. Architectural insight into when adaptive meshing is expected to help and when it provides little benefit.

design exploration.

## 2. BACKGROUND AND RELATED WORK

Clock distribution has been a central challenge in synchronous digital design for decades. As technology has scaled and system complexity has increased, the clock network has evolved from a secondary concern to a first-order limiter of power, performance, and robustness. This section reviews classical clock distribution approaches, explains why they break down at advanced nodes, and motivates hybrid and adaptive architectures as a response to these limitations.

### 2.1 Clock Trees and Scaling Limits

Clock trees are among the earliest and most widely deployed clock distribution structures. The core idea is straightforward: by constructing a hierarchical branching network in which the geometric path length from the clock source to each sink is equalized, arrival times can be aligned. Balanced H-tree structures are a canonical example of this approach and have been extensively used in microprocessors and ASICs due to their conceptual simplicity and relatively low wiring overhead.

At older technology nodes, this strategy was effective because interconnect resistance was low and process variation was modest. Under those conditions, geometric symmetry translated reasonably well into electrical symmetry. Equal path lengths implied roughly equal RC delays, and residual skew could often be managed with buffering or minor tuning.

However, as feature sizes have shrunk, the assumptions underpinning clock tree design have progressively eroded. Resistance per unit length has increased significantly due to narrower wires and increased surface scattering, while capacitance has not scaled proportionally. As a result, RC delay has become more sensitive to small geometric and material variations. In modern processes, even small differences in wire width, thickness, or local environment can introduce measurable delay differences.

This sensitivity is particularly problematic for clock trees because mismatches accumulate along long paths. Even when two paths are geometrically identical, random variation in resistance and capacitance along each segment leads to delay divergence. Since clock trees provide only a single path from the source to each sink, there is no inherent mechanism to average out these mismatches. The result is that large skew can emerge even in carefully balanced designs, especially under worst-case or high-percentile conditions.

From an architectural perspective, this breakdown highlights a fundamental limitation of tree-based distribution. While trees minimize wiring and power, they are intrinsically fragile in the presence of variation. As clock frequencies increase and timing margins shrink, this fragility becomes increasingly unacceptable.

### 2.2 Clock Meshes and Robustness

Clock meshes were introduced as a response to the robustness limitations of clock trees. Instead of enforcing a single path from the source to each sink, a mesh provides multiple redundant paths through a grid-like interconnect structure. Each sink effectively receives the clock signal through many parallel routes, and the resulting arrival time reflects an average over those paths.

This redundancy fundamentally changes the behavior of the clock network. Local variations in resistance or capacitance along any single path are diluted by the presence of alternative routes. As a result, clock meshes exhibit much lower sensitivity to process variation and local mismatches. In practice, this leads to dramatically reduced skew, especially at high percentiles.

Clock meshes have therefore been adopted in high-performance designs where timing robustness is critical. Notably, large microprocessors have employed global or semi-global meshes to suppress skew and simplify timing closure.

The primary drawback of clock meshes is power. A mesh introduces a large amount of wire capacitance that toggles every cycle, regardless of data activity. This capacitance dominates dynamic clock power and can represent a significant fraction of total chip power. In addition, dense meshes consume routing resources and can interfere with signal routing, exacerbating congestion and design complexity.

From an architectural standpoint, clock meshes represent the opposite extreme from clock trees. They trade power and area for robustness. While effective, this trade is often too costly for energy-constrained systems or large dies where clock power must be carefully managed.

### 2.3 Hybrid Clock Networks

Hybrid clock networks attempt to reconcile the competing goals of power efficiency and robustness by combining elements of trees and meshes. In a typical hybrid architecture, a tree-like structure is used for global distribution, while local meshes are introduced near sinks to equalize arrival times and suppress variation-induced skew.

The intuition behind hybrid designs is that not all portions of the clock network contribute equally to skew. Long global routes benefit from the efficiency of trees, while short local routes benefit from the averaging properties of meshes. By confining meshing to localized regions, hybrid architectures aim to capture most of the robustness benefits of meshes while avoiding their full power cost.

Hybrid clocking has appeared in both academic literature and industrial practice, often motivated by empirical observations rather than systematic analysis. Designers may add local mesh segments or cross-links in regions that appear problematic, but the density and placement of these meshes are typically determined heuristically.

A key gap in prior work is the lack of quantitative guidance on how much meshing is actually required. While it is clear that some local connectivity helps, it is less clear how dense the mesh must be to achieve diminishing returns. Without this insight, designers risk over-meshing, incurring unnecessary power overhead, or under-meshing, failing to achieve sufficient robustness.

This gap motivates the need for architectural-level modeling that can sweep mesh density systematically and expose the skew–power tradeoff explicitly, which is the focus of this work.

### 2.4 Adaptive Clocking Concepts

Beyond static clock topologies, a wide range of adaptive clocking techniques have been proposed to cope with variability and uncertainty. At the circuit level, these include tunable delay elements, deskew circuits, phase interpolators, and digitally controlled delay lines. Such techniques can compensate for skew dynamically but often require additional circuitry, calibration infrastructure, and control logic.

At higher levels of abstraction, adaptive strategies have been explored that modify clock distribution structures themselves. Examples include dynamically enabling or disabling mesh segments, redistributing clock resources based on observed timing behavior, or adapting clock domains in response to workload or environmental conditions.

While these approaches can be powerful, they also introduce complexity and overhead. Circuit-level adaptivity increases area and design effort, while dynamic architectural adaptivity can complicate verification and control. Moreover, the effectiveness of adaptive strategies depends strongly on the presence of spatial heterogeneity. If the underlying clock network is already well balanced, adaptation may provide little benefit.

Region-adaptive meshing represents a relatively lightweight architectural adaptation mechanism. Instead of modifying delays dynamically, it redistributes static clock resources based on observed or predicted lateness. This approach avoids the need for active control circuits while still allowing the clock network to respond to non-uniformity.

However, the effectiveness of region-adaptive meshing has not been well quantified, particularly in comparison to uniform hybrid designs. Its benefits are expected to be most pronounced in asymmetric layouts, such as designs with large macros, heterogeneous sink capacitances, or late-stage ECOs that disrupt balance. In symmetric scenarios, its advantage is less obvious.

This work evaluates region-adaptive meshing within a controlled analytical framework, clarifying both its strengths and its limitations.

## 3. DESIGN AND MODELING APPROACH

This section describes the clock network architecture studied in this work and the analytical modeling framework used to evaluate its behavior. The emphasis is on capturing the dominant physical effects that govern clock skew, power, and robustness at an architectural level, while keeping the model sufficiently lightweight to support large parameter sweeps and Monte Carlo analysis.

### 3.1 Overall Architecture

The modeled clock distribution architecture consists of a global H-tree backbone combined with optional region-level local clock meshes. The clock source is assumed to be centrally located, and the H-tree distributes the clock signal hierarchically across the die to a set of regional tap points. This global structure is chosen because H-trees remain a common baseline in industrial clock distribution due to their relative power efficiency and geometric symmetry.

The die is partitioned into a regular grid of regions. In the evaluated configurations, the number of regions is determined by the depth of the H-tree, resulting in a tiled layout where each region represents a local clock domain fed by a single H-tree tap. Each region contains one or more clock sinks, representing flip-flop clusters or local clock loads.

Within each region, a local clock mesh may be instantiated. The local mesh is modeled as a two-dimensional grid of resistive and capacitive wire segments that connect nodes within the region. The purpose of this mesh is not to distribute the clock globally, but to locally equalize arrival times among sinks and to average out resistive and capacitive mismatches introduced by the upstream H-tree paths.

The density of the local mesh is controlled by a single scalar parameter p_r ∈ [0, 1]. This parameter directly determines how many mesh connections are instantiated in each region:

- p_r = 0 corresponds to a pure H-tree configuration with no local meshing.
- Intermediate values of p_r correspond to sparse local meshes, where only a subset of possible grid connections are present.
- p_r = 1 corresponds to a dense local mesh, approximating a full regional clock mesh.

This parameterization allows the clock network to smoothly interpolate between tree-dominated and mesh-dominated designs, enabling systematic exploration of the skew–power tradeoff. Importantly, the same p_r definition is applied uniformly across regions for the uniform sweep experiments, ensuring that changes in behavior can be directly attributed to mesh density rather than topological differences.

In addition to uniform meshing, the architecture supports region-adaptive meshing. In this mode, each region is assigned its own mesh density value p_r,i, subject to a global average constraint. The adaptive policy redistributes mesh density based on measured clock arrival lateness, reinforcing regions that lag behind the earliest arrivals. This mechanism is intended to model an architectural-level adaptation strategy without introducing circuit-level tuning or feedback.

### 3.2 Analytical RC Modeling

To evaluate the clock network efficiently while retaining physical fidelity, the entire architecture is modeled using an analytical RC framework based on Modified Nodal Analysis (MNA). Each wire segment in the clock network is represented by a lumped resistance and capacitance derived from its physical length and technology parameters. This abstraction captures first-order delay and loading effects that dominate clock behavior at the architectural scale.

Resistive effects are modeled explicitly for all interconnect segments. Resistance per unit length is specified separately for H-tree segments and mesh segments, allowing the model to reflect different wire classes, such as semi-global and global clock metals. Capacitance per unit length is similarly parameterized and includes the dominant wire-to-ground and wire-to-substrate components.

Clock drivers are modeled using a Thevenin equivalent consisting of an ideal voltage source in series with a finite output resistance. This captures the interaction between driver strength, interconnect loading, and downstream delay without requiring transistor-level modeling. Each clock sink is modeled as a lumped capacitive load connected to the appropriate node, representing the aggregate input capacitance of sequential elements in that region.

The resulting RC network is assembled into sparse conductance and capacitance matrices. Transient simulation is performed using an implicit Euler time integration scheme, which is unconditionally stable for stiff RC systems and allows reasonably large time steps without numerical instability. This choice is particularly important for Monte Carlo analysis, where hundreds of transient simulations must be performed efficiently.

Dynamic clock power is estimated using a $CV^2f$ formulation based on the total effective switching capacitance of the clock network. Because the clock toggles every cycle and the waveform shape is similar across configurations, differences in dynamic power are dominated by differences in total capacitance rather than subtle waveform effects. Using $CV^2f$ therefore provides a fair and consistent basis for comparing power across architectures without introducing unnecessary complexity.

## 4. IMPLEMENTATION DETAILS

The modeling framework is implemented entirely in MATLAB and organized into modular scripts and functions to ensure clarity, reproducibility, and extensibility. This section describes the role of each major script and how they interact during a typical experiment.

### 4.1 params.m

The params.m script serves as the central configuration file for the entire project. It defines all global parameters that control the clock network architecture, technology assumptions, simulation behavior, and output generation.

Key parameters specified in this script include the clock frequency, supply voltage, driver resistance, interconnect resistance and capacitance per unit length, die dimensions, H-tree depth, number of regions, and mesh pitch limits. The script also defines simulation parameters such as the transient time step, total simulation duration, arrival threshold voltage, and Monte Carlo variation magnitudes for resistance, capacitance, and driver strength.

Monte Carlo settings, including the number of trials for robustness analysis, are defined here, allowing the same architectural configuration to be evaluated under different variation budgets without modifying the core simulation code. Output directories and file naming conventions are also handled in this script, ensuring that all results are written to a consistent location for post-processing.

By consolidating all configuration parameters into a single file, the implementation supports straightforward reproducibility and makes it easy to explore alternative technology nodes or architectural assumptions.

### 4.2 htree_build.m

The htree_build.m function constructs the geometric structure of the global H-tree backbone. Given the die dimensions and the specified number of hierarchical levels, the function recursively generates branching points and computes the coordinates of each regional tap.

The output of this function is a set of tap locations that serve as the interface between the global H-tree and the regional clock meshes. These tap points define the boundaries of each region and determine where local meshes are attached. By separating H-tree construction from mesh generation, the implementation cleanly decouples global and local clock distribution.

### 4.3 gen_topology.m

The gen_topology.m script generates the full clock network topology for a given vector of mesh density values p_r. It combines the global H-tree edges produced by htree_build.m with region-level mesh structures.

For each region, the script determines how many mesh connections to instantiate based on the corresponding p_r value. Mesh nodes are placed on a regular grid within the region, and resistive-capacitive edges are created between neighboring nodes according to the desired density. Sink nodes are attached to appropriate mesh or tap nodes, depending on the configuration.

The output of this script is a complete topological description of the clock network, including node identifiers, edge lists with associated physical lengths and layer types, and sink definitions. This topology is then passed directly to the MNA assembly stage.

### 4.4 build_mna.m

The build_mna.m function converts the abstract topology produced by gen_topology.m into the numerical matrices required for simulation. It assembles the sparse conductance matrix G and capacitance matrix C by stamping contributions from each resistive and capacitive element in the network.

For every wire segment, the function adds the appropriate conductance entries between connected nodes and distributes capacitance to the corresponding diagonal entries. Driver conductance and source terms are added at the root node, and sink capacitances are attached at sink nodes.

The resulting matrices capture the complete linear RC behavior of the clock network. Because the matrices are sparse and structured, they can be factorized efficiently and reused across time steps in the transient simulation.

### 4.5 sim_transient.m

The sim_transient.m script performs the core transient simulation of the clock network. Using the MNA matrices produced by build_mna.m, it integrates the system forward in time using an implicit Euler method.

At each time step, the script solves a linear system to obtain node voltages. The voltage waveform at each sink is monitored, and the clock arrival time is defined as the first time the voltage crosses a specified threshold corresponding to 50 percent of the supply voltage. This definition is consistent across all configurations, ensuring fair comparison.

After all sinks have crossed the threshold, clock skew is computed as the difference between the earliest and latest arrival times. The script also accumulates statistics needed to estimate average dynamic power. Early-exit logic is included to terminate the simulation once all arrivals have been observed, improving runtime efficiency.

### 4.6 monte_carlo.m

The monte_carlo.m function evaluates robustness by repeatedly perturbing the clock network parameters and re-running the transient simulation. For each Monte Carlo trial, random variations are applied to interconnect resistance per unit length, capacitance per unit length, and driver resistance based on specified standard deviations.

For each trial, skew and power are recorded. After completing all trials, the function computes statistical summaries including mean skew, standard deviation, and high-percentile metrics such as p95 and p99 skew. These metrics are used to characterize worst-case behavior and to construct skew distribution plots.

This approach allows robustness to be quantified without relying on pessimistic corner assumptions, providing a more nuanced view of clock network behavior under variation.

### 4.7 sweep_pr.m

The sweep_pr.m script orchestrates the full experimental flow. It sweeps uniform mesh density values across a predefined range, invokes topology generation, MNA assembly, transient simulation, and Monte Carlo analysis for each configuration, and stores the resulting metrics in structured tables.

For region-adaptive experiments, the script first runs a nominal simulation to extract arrival times, applies an adaptive policy to redistribute mesh density, and then repeats the evaluation for the adapted design. The script also generates all plots used in the evaluation, including skew–power Pareto fronts, skew versus mesh density curves, robustness distributions, and spatial arrival heatmaps.

By centralizing experiment control in this script, the implementation ensures that all results are generated consistently and that figures and data tables remain synchronized.

## 5. EVALUATION

This section evaluates the proposed hybrid clock network architecture across skew, power, robustness, and spatial behavior. All results are generated using the analytical RC simulator described earlier, with consistent modeling assumptions across all experiments. Unless otherwise stated, skew values refer to p99 skew extracted from Monte Carlo simulations, and power values reflect average dynamic clock power computed using a CV²f formulation.

### 5.1 Skew–Power Tradeoff

Figure 2 presents the annotated skew–power Pareto front comparing pure H-tree, uniform hybrid, full mesh, and region-adaptive hybrid clock networks. Power is plotted on the x-axis in milliwatts, while p99 clock skew is plotted on the y-axis in picoseconds using a logarithmic scale to emphasize differences across several orders of magnitude.

The pure H-tree design represents the lowest-power configuration, as expected, since it minimizes total clock wire capacitance and avoids redundant paths. However, this efficiency comes at the cost of extremely large skew. In the evaluated configuration, the H-tree exhibits p99 skew on the order of several hundred picoseconds. This magnitude is far beyond what would be tolerable in modern multi-GHz systems and highlights the vulnerability of tree-based distribution to resistive imbalance and variation.

Introducing local meshing fundamentally changes this behavior. Even sparse hybrid configurations reduce skew by more than three orders of magnitude relative to the H-tree baseline. For example, at a uniform mesh density of p_r ≈ 0.25, p99 skew drops into the sub-picosecond range. This dramatic reduction indicates that only a modest amount of local connectivity is required to equalize arrival times and suppress resistive mismatch effects introduced along the H-tree paths.

From a power perspective, the hybrid design incurs an increase relative to the pure tree, but the increase is modest compared to the skew improvement. Importantly, the Pareto front reveals diminishing returns beyond moderate mesh densities. As p_r increases beyond approximately 0.25 to 0.5, additional meshing continues to increase switching capacitance and therefore power, while providing only marginal improvements in skew. The full mesh configuration occupies the extreme high-power, low-skew corner of the Pareto space, but offers little practical advantage over much sparser hybrid designs.

This result is a key architectural insight. It demonstrates that near-optimal skew can be achieved without resorting to dense meshes, and that aggressive meshing represents wasted power rather than meaningful performance improvement. The Pareto front makes this tradeoff explicit and provides quantitative guidance for early-stage clock network design.

### 5.2 Skew Versus Mesh Density

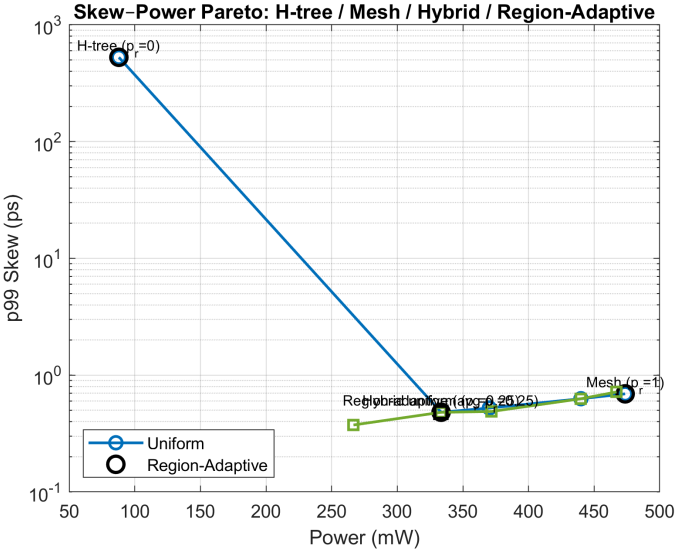

*Figure 1: Annotated skew–power Pareto front across uniform mesh density sweep pr∈[0,1]p_r \in [0,1]pr∈[0,1], showing the knee around pr≈0.25p_r \approx 0.25pr≈0.25 where skew collapses with modest power increase.*

To further understand the behavior observed in the Pareto analysis, Figure 3 plots p99 skew as a direct function of uniform mesh density p_r. This representation isolates the effect of meshing from power and highlights how skew evolves as local connectivity is increased.

The curve exhibits a clear knee at low mesh densities. Starting from p_r = 0, skew decreases rapidly as soon as local mesh connections are introduced. The steep initial slope indicates that the dominant contributors to skew are mitigated quickly by providing even a small number of alternative current paths near the sinks. Physically, this reflects the fact that local averaging effects suppress mismatch between neighboring clock nodes, reducing sensitivity to individual resistive segments.

Beyond p_r ≈ 0.25, the slope of the curve flattens significantly. Additional meshing produces progressively smaller reductions in skew, confirming that most of the benefit is captured early. This behavior is consistent with the intuition that once local imbalances are averaged out, further connectivity provides redundancy without addressing a remaining dominant skew mechanism.

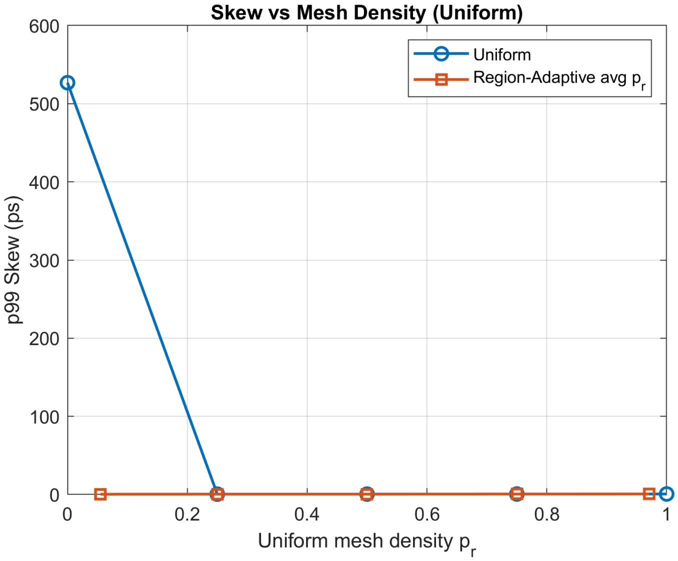

*Figure 2: p99 skew versus mesh density prp_rpr, showing rapid improvement at low prp_rpr and diminishing returns beyond the knee.*

This result has direct architectural implications. It suggests that designers can target a sparse hybrid mesh as a default solution, rather than incrementally increasing mesh density until skew targets are met. In other words, skew is not a continuously tunable knob across the full p_r range. Instead, there is a small region where meshing is highly effective, followed by a regime of diminishing returns.

5.3 Robustness Under Variation

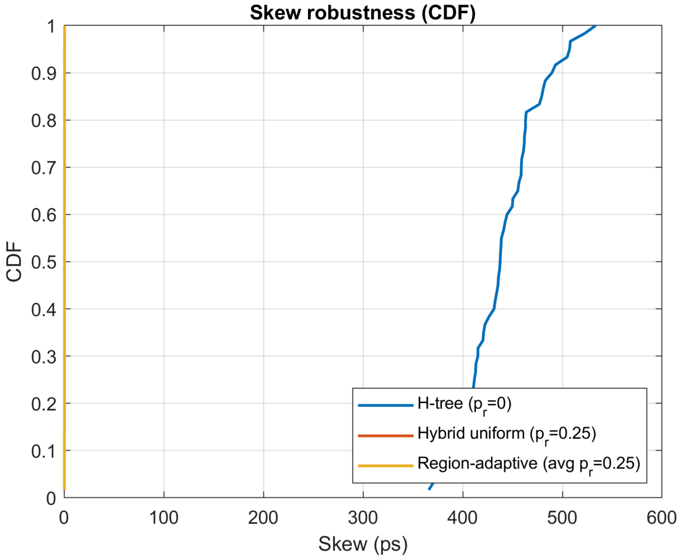

*Figure 3a: Skew CDF under Monte Carlo variation for representative topologies, highlighting tail suppression for hybrids.*

While nominal skew is important, clock networks must also be robust to process and device variation. To evaluate robustness, Monte Carlo simulations were performed with random perturbations applied to interconnect resistance, capacitance, and driver strength. Figure 4 shows the cumulative distribution functions of skew for representative configurations.

The pure H-tree exhibits a pronounced long tail in its skew distribution. Even when the nominal skew is already large, variation further exacerbates worst-case behavior. This reflects the fundamental sensitivity of tree-based distribution to local mismatches along long paths. Small variations accumulate, and there is no mechanism to average them out.

In contrast, hybrid clock networks exhibit tightly bounded skew distributions. The CDFs for hybrid designs rise steeply and saturate quickly, indicating that extreme skew events are rare. This behavior holds even at moderate mesh densities, reinforcing the conclusion that sparse meshing is sufficient to suppress variation-induced skew.

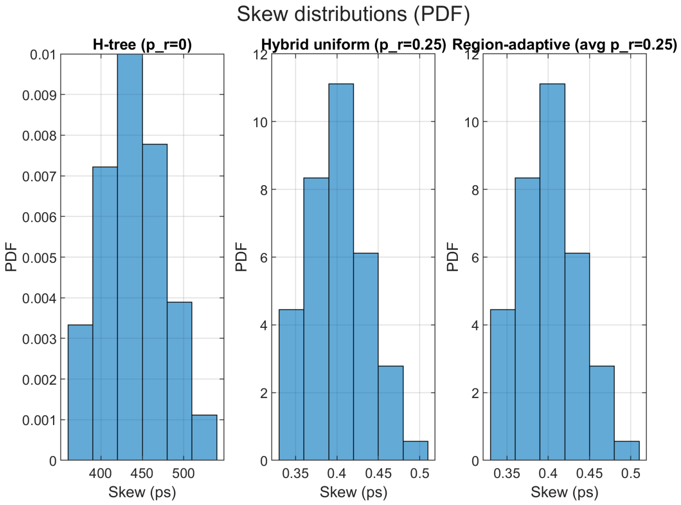

*Figure 3b: Skew histogram under Monte Carlo variation, showing the tightening of the distribution for hybrid designs.*

Importantly, the robustness improvement is not limited to average behavior. The reduction in tail skew at high percentiles such as p95 and p99 is substantial. From a timing closure perspective, this is often more important than improvements in mean skew, since worst-case behavior determines guardbands and margins.

These results demonstrate that hybrid meshing improves both nominal performance and robustness. The same architectural feature that reduces skew also suppresses variability, providing a double benefit.

### 5.4 Region-Adaptive Meshing

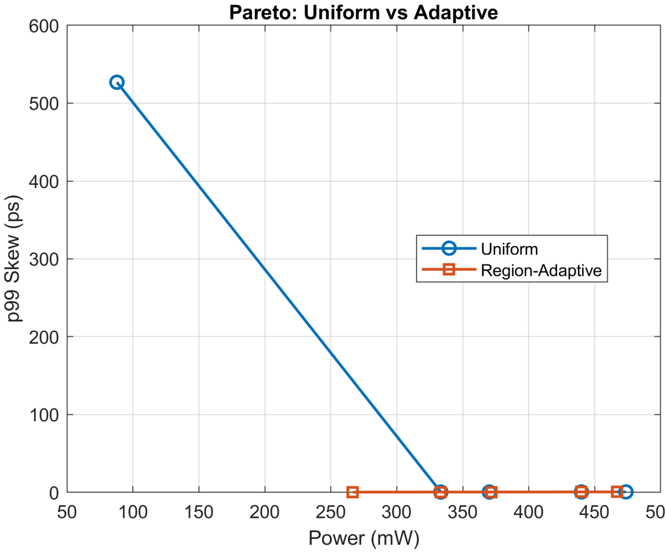

*Figure 4: Uniform versus region-adaptive designs under equal average mesh density, showing limited advantage for adaptive meshing in a symmetric layout.*

Region-adaptive meshing was evaluated by redistributing mesh density based on nominal arrival time lateness while keeping the average mesh density fixed. The goal was to determine whether selectively reinforcing late regions could further reduce skew or improve robustness beyond uniform hybrid designs.

In the evaluated configuration, region-adaptive designs achieve skew and power levels comparable to uniform hybrids with the same average p_r. However, they do not consistently outperform uniform designs. This outcome is not a failure of the adaptive approach, but rather a consequence of the symmetry inherent in the test case.

The modeled system uses a symmetric H-tree backbone and a uniform spatial distribution of sinks. Under these conditions, once sparse meshing is introduced, arrival times are already well equalized. There is little residual spatial skew for the adaptive policy to exploit. As a result, redistributing mesh density yields minimal benefit.

This result is itself informative. It suggests that adaptive meshing is most valuable when the clock network is stressed by asymmetry. Examples include heterogeneous sink capacitances, macro-dominated floorplans, clock islands with different loading, or late-stage ECOs that disrupt balance. In such scenarios, uniform meshing may over-provision some regions while under-serving others, creating an opportunity for adaptive policies to improve efficiency.

Thus, the adaptive results validate that the proposed framework can support adaptive policies, while also clarifying the conditions under which adaptivity is expected to matter.

### 5.5 Spatial Arrival Patterns

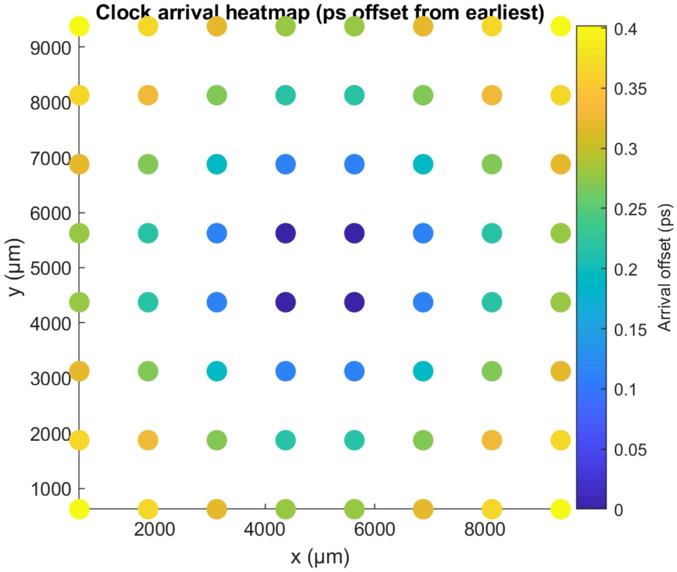

*Figure 5: Clock arrival heatmap for uniform hybrid design at pr=0.25p_r=0.25pr=0.25, plotted as offset from the earliest sink arrival.*

Figure 5 presents clock arrival heatmaps for representative uniform and region-adaptive hybrid designs. Each heatmap visualizes the spatial distribution of clock arrival offsets across the die, referenced to the earliest arrival.

Both heatmaps show highly uniform arrival times with sub-picosecond variation across regions. The absence of strong spatial gradients reinforces the conclusion that sparse hybrid meshing effectively equalizes clock arrival times in symmetric layouts. The similarity between uniform and adaptive heatmaps further supports the observation that adaptive meshing does not materially change spatial behavior in this regime.

While heatmaps are not the primary quantitative metric, they provide valuable intuition. They visually confirm that the analytical skew metrics correspond to a physically uniform clock distribution, rather than an artifact of averaging.

## 6. DISCUSSION AND IMPLICATIONS

The evaluation results highlight several important architectural implications for clock network design. First, clock skew is dominated by a small number of resistive imbalances that can be mitigated with limited local connectivity. Dense meshing is therefore unnecessary for achieving near-optimal skew in many cases.

Second, architectural-level decisions have a profound impact on robustness. By introducing local averaging early in the clock network, hybrid designs suppress skew variability and reduce the need for conservative timing margins. This has downstream implications for both performance and power.

Third, the results emphasize the value of analytical modeling at early design stages. Rather than relying on late-stage physical tuning, designers can use models like the one presented here to identify efficient regions of the design space and avoid over-engineered solutions.

Finally, the adaptive meshing results clarify that adaptivity is not a universal win. Its benefits depend on the presence of heterogeneity. This insight is critical for guiding future research and avoiding unnecessary complexity in cases where uniform designs are already near-optimal.

## 7. CONCLUSION AND FUTURE WORK

This work demonstrates that analytical RC modeling provides actionable insight into the architectural tradeoffs of clock network design. By sweeping local mesh density and evaluating skew, power, and robustness under variation, I showed that hybrid clock networks achieve orders-of-magnitude skew reduction relative to H-trees while consuming far less power than full meshes. A sparse hybrid design around p_r ≈ 0.25 captures most of the available benefit, revealing a clear knee in the design space.

Region-adaptive meshing was evaluated within the same framework and shown to match uniform hybrid performance in symmetric layouts, highlighting both the strengths and limitations of adaptive policies. Rather than diminishing the contribution, this result clarifies when adaptivity is expected to provide value and sets the stage for future exploration.

Future work will focus on extending the model to asymmetric floorplans, heterogeneous sink loads, and macro-dominated designs where adaptive meshing is likely to yield larger gains. Incorporating clock gating, coupling effects, and post-layout extracted parasitics would further bridge the gap between architectural modeling and physical implementation.

Overall, this project shows that careful architectural modeling can significantly reduce clock power and skew while improving robustness, and that hybrid clock networks offer a practical and efficient solution for modern system-on-chip designs.

## REFERENCES

[1] J. Cong and S. K. Lim, "Physical planning with retiming," *IEEE Trans. Computer-Aided Design*, vol. 20, no. 11, pp. 1380–1395, Nov. 2001.

[2] E. G. Friedman, *Clock Distribution Networks in VLSI Circuits and Systems*. IEEE Press, 1995.

[3] C. J. Alpert, A. Devgan, and S. T. Quay, "Buffer insertion for noise and delay optimization," *IEEE Trans. Computer-Aided Design*, vol. 18, no. 11, pp. 1633–1645, Nov. 1999.

[4] A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu, *VLSI Physical Design: From Graph Partitioning to Timing Closure*. Springer, 2011.

[5] International Roadmap for Devices and Systems (IRDS), "Interconnect," 2022.