Multilevel Tree Fusion for Robust Clock Networks

Dong-Jin Lee and Igor L. Markov {ejdjsy, imarkov}@eecs.umich.edu
University of Michigan, 2260 Hayward St., Ann Arbor, MI 48109-2121

***Abstract*—Recent improvements in clock-tree and mesh-based topologies maintain a healthy competition between the two. Trees require much smaller capacitance, but meshes are naturally robust against process variation and can accommodate late design changes. Cross-link insertion has been advocated to make trees more robust, but is limited in practice to short distances. In this work we develop a novel non-tree topology that fuses several clock trees to create large-scale redundancy in a clock network. Empirical validation shows that our novel clock-network structure incrementally enhances robustness to satisfy given variation constraints. Our implementation called Contango3.0 produces robust clock networks even for challenging skew limits, without parallel buffering used by other implementations. It also offers a fine trade-off between power and robustness, increasing the capacitance of the initial tree by less than 60%, which results in 2.3× greater power efficiency than mesh structures.**

I. Introduction

The central question in clock-network design is the choice between a tree and non-tree topology. High-performance microprocessors typically use meshes due to their robustness to late design changes and process variations, but at a great cost in terms of capacitance. Tree topologies offer many advantages, including simplicity, symmetry, faster timing analysis and amenability to incremental tuning.

The dichotomy between meshes and tree is striking, and several researchers attempted to find intermediate topologies that would retain the advantages of meshes but reduce capacitance overhead. The key idea in the literature is to insert cross-links into clock trees, creating redundant paths to sinks that contribute to nominal or variational clock skew. Most publications discuss cross-links that directly connect pairs of sinks. Surprisingly, none of these techniques were useful at the ISPD 2009-2010 clock-network contests [22], [23] despite diligent attempts, as improved tree-tuning methods were sufficient. Careful experiments and analytical estimates [13] have shown that direct cross-links are only effective in poorly tuned clock trees and/or at relatively short distances. However, in high-quality clock trees it is rare to find a critical pair of sinks at a short distance. A recent proposal [13] suggests adding cross-links higher in the tree to connect entire branches. As several other publications with strong empirical results, [13] uses unrealistically large composite buffers, and arranges them in a unique two-layer configuration (10+40 small inverters). Given that the ISPD 2010 contest infrastructure does not adequately model such configurations, the competitiveness of cross-links in practice remains unclear.

In this paper, we propose a novel family of clock-network topologies which maintain most advantages of tree structures, but are significantly more robust with respect to variations. Using in-depth structural analysis, we quantitatively describe where and why a given tree structure fails to satisfy variation-related constraints (see Figure 6) and explain how it can be improved (see Figure 5).

Specific innovations in this work include

- Statistical models for delay and skew in buffered clock networks.
- A technique to identify critical sink pairs based on robustness analysis.
- A novel clock-network structure (*fused multilevel trees*) based on auxiliary-tree construction and fusing to enhance robustness.
- A *sink-splitting* technique for fusion topologies to leverage the efficiency of tree optimization algorithms.
- An experimental configuration with monolithic wires that remedies known deficiencies in ISPD 2010 benchmarks.

The remainder of this paper is organized as follows. Section II covers prior work. Section III describes our statistical models for delay and skew in buffered clock networks including the proposed clock-network topologies. We propose a novel clock-network structure in Section IV and implementation details in Section V. Our empirical results are described in Section VI. Conclusions are given in Section VII.

II. Background and Prior Work

We review the notion of local skew in modern clock-network synthesis and briefly outline research results on clock trees, cross-links and mesh topologies.

A. Sink pairs eligible for local-skew calculation

In a large clock network, skew between adjacent and connected sinks is a more meaningful optimization objective than global skew [8], [19]. When two clock sinks are connected by combinational logic (Figure 1a) the clock skew between two sinks directly affects the useful portion of clock cycle time for the combinational logic. Otherwise, where there is no combinational logic between two sinks (Figure 1b), the skew

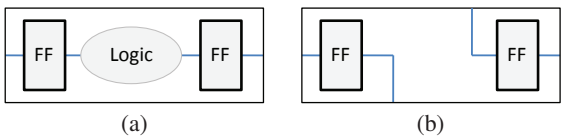


Fig. 1. Eligible clock sink pairs. (a) There is combinational logic between two sinks, which makes the skew between these two sinks affects the useful portion of clock cycle time. (b) This sink pair is not eligible because the sinks are not logically dependent.

between them is not a source of performance degradation, therefore we do not need to optimize the clock network to reduce the skew between those sink pairs. Eligible sink pairs for skew can be defined based on the netlist after Register-Transfer Level (RTL) synthesis so that only sink pairs that are connected by combinational logic are considered for skew calculation. In the ISPD 2010 Clock Network Synthesis (CNS) Contest, *local skew distance limit* was introduced to define the eligible sink pairs and local skew [23]. If the Manhattan distance between two sinks is less than the local skew distance limit, it was assumed that there is combinational logic between the two sinks and otherwise, there is no logic dependency. We use the same notion of local skew in our work, but do not rely on the metric definition, and all our techniques apply in a more realistic context where eligible pairs of sinks are derived directly from the netlist.

B. Clock Trees

Tree structures have been widely used in academic and commercial tools. Simple methods including *H-tree* [2], the *method of means and medians* (MMM) [9], the *geometric matching algorithm* (GMA) [6] and *path length balancing method* (PLB) [10] were commonly utilized before the *deferred merge embedding* (DME) algorithm [3], [7] was introduced. Recently several methodologies for SoC clock-tree tuning have been developed with robustness improvement. A clock-synthesis methodology for SPICE-accurate skew optimization with tolerance to voltage variations was proposed in [12]. The Dynamic Nearest-Neighbor Algorithm (DNNA) to generate tree topology and the Walk-Segment Breadth First Search (WSBFS) for routing and buffering were proposed in [20]. A three-stage CTS flow based on an obstacle-avoiding balanced clock-tree routing algorithm with monotonic buffer insertion is proposed in [14]. A Dual-MST (DMST) geometric matching approach is proposed in [15] for topology construction and recursive buffer insertion. The modeling techniques and algorithms for microprocessor clock power optimization subject to local skew constraints in the presence of variations are proposed in [11].

C. Cross-links

Cross-link insertion for clock trees is proposed in [16], [18] to reduce skew variability by changing a clock tree to a non-tree topology. These methods are later extended to handle buffered clock trees in [17], [25]. While most cross-link insertion techniques do not seem competitive with best tree-tuning approaches, a recent cross-link scheme proposed in [13] achieves low overall capacitance by inserting cross links between internal nodes of a clock tree to reduce the total cross-link length. However, these resulting networks use unique, large two-level buffers (10+40 inverters) that seem responsible for the improvement but are not adequately modeled by the ISPD 2010 contest infrastructure.

D. Meshes

From the mid 1990s when the impact of PVT variation became significant, clock networks were more affected by PVT variations than random logic, due to their structure and more stringent timing constraints. In a tree network, such unexpected changes are likely to propagate to the sinks. Mesh (or grid) structures have emerged to address the structural drawbacks of trees. In meshes, there are multiple paths from the clock source to individual clock sink; thus, the impact of variations on one path can be averaged out by multiple redundant paths [28]. However, meshes require significant overhead in terms of on-chip resources and power. Published examples suggest that mesh-type clock networks suffer much greater power consumption. Nevertheless, mesh structures were utilized to satisfy tight variation-related constraints in high-performance microprocessor designs where performance is more emphasized than power consumption [1]. Some methods to analyze the characteristics of mesh structures are proposed in [4], [27] and a combinatorial algorithm to optimize a clock mesh is proposed in [24]. An obstacle-avoiding clock mesh synthesis method which applies a two-stage approach of mesh construction followed by driving-tree synthesis is proposed in [21], [26]. A methodology based on binary linear programming for clock mesh synthesis is described in [5].

III. Variation modeling for buffered paths

In this section, we develop statistical models for delay and skew in RC-buffered clock networks, including proposed clock-network topologies.

A. Impact of variation on delay

In the presence of PVT variations, the delay of a buffered path p can be treated as a random variable D_p whose mean d_p is the nominal delay. Given that tree-like clock networks entail long paths without significant reconvergence, path delay can be modeled by Gaussian variables:[1]

$$D_p = N(d_p, \sigma_p^2) \tag{1}$$

The delays of serially connected paths p_1 and p_2 add up.

$$D_{p_1p_2} = D_{p_1} + D_{p_2},\ \mathrm{E}[D_{p_1p_2}] = \mathrm{E}[D_{p_1}] + \mathrm{E}[D_{p_2}] \tag{2}$$

$$\sigma^2_{p_1p_2} = \sigma^2_{p_1} + \sigma^2_{p_2} + 2\sigma_{p_1}\sigma_{p_2}\rho_{(p_1,p_2)} \tag{3}$$

where $0 \le \rho_{(p_1,p_2)} \le 1$ is the correlation between D_{p_1}, D_{p_2}.

Given n parallel paths from a to b, we tune nominal path delays using existing methods [11], [12] to bring their difference under $10ps$. We also size the drivers, that jointly drive the sink, to have similar strength. Under these circumstances, the random variable of path delay and its expectation (nominal delay) are

$$D_{p(a,b)} = \sum_{i=1}^{n} D_{p_i}/n,\ d_{p(a,b)} = \sum_{i=1}^{n} d_{p_i}/n \tag{4}$$

Then, the variance of $D_{p(a,b)}$ is

$$\sigma^2_{p(a,b)} = \sum_{i=1}^{n} \sigma^2_{p_i}/n^2 + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \sigma_{p_i}\sigma_{p_j}\rho_{(p_i,p_j)}/n^2 \tag{5}$$

[1]While specific sources of variation and the probability distributions of device parameters can be complicated, the Central Limit Theorem suggests that path delay distributions are close to normal.

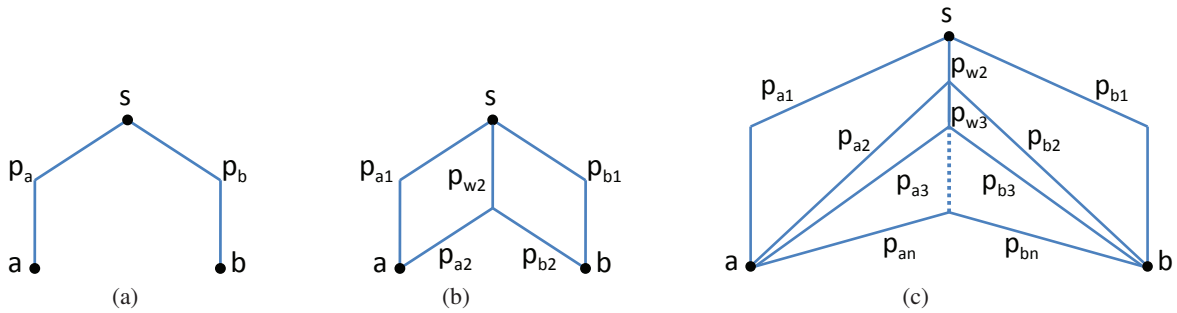


Fig. 2. Simple clock networks with source node s, two sink nodes a and b. All paths are considered buffered. (a) a tree, (b) redundant paths. (c) n multilevel paths for each sink. Each i-th ($2 \leq i$) new root-to-sink path consists of a shared p_{w_i} section and a p_{a_i} or p_{b_i} section that is not shared.

Example III.1 Consider the case $n = 2$, $\sigma^2_{p_1} = \sigma^2_{p_2} = 10$ and $\rho_{(p_1,p_2)}$=0.1. Then $\sigma^2_{p(a,b)} = 5.5$, which reduces standard deviation by about 26% compared to a single path. Thus, having multiple paths reduces the impact of PVT variation compared to a single path.

B. Impact of variation on skew

Let s be a source node and a, b be two sink nodes. *Nominal skew* (without variation) is defined as

$$\text{skew}_{(a,b)} = |d_{p(s,a)} - d_{p(s,b)}| \tag{6}$$

We define *total signed skew* (with variation), *mean signed skew*, *variational signed skew*, *signed skew variance*, *total absolute skew* and *variational absolute skew* as follows.

$$S_{(a,b)} = D_{p(s,a)} - D_{p(s,b)}, \quad \bar{S}_{(a,b)} = \text{E}[S_{(a,b)}] \tag{7}$$

$$S^*_{(a,b)} = S_{(a,b)} - \bar{S}_{(a,b)}, \quad \sigma^2_{s(a,b)} = \text{E}[{S^*_{(a,b)}}^2] \tag{8}$$

$$\text{Skew}_{(a,b)} = |S_{(a,b)}|, \quad \text{Skew}^*_{(a,b)} = |S^*_{(a,b)}| \tag{9}$$

When nominal skew is zero, one can show that

$$\text{Expected skew} : \text{E}[\text{Skew}_{(a,b)}] = \sigma_{s(a,b)}\sqrt{2/\pi} \tag{10}$$

$$\text{Skew variance} : \text{var}[\text{Skew}_{(a,b)}] = \sigma^2_{s(a,b)}(1 - 2/\pi) \tag{11}$$

Note that mean absolute skew can be positive with zero nominal skew.

For yield analysis, given a variation bound $x > 0$,

$$\text{P}[\text{Skew}_{(a,b)} < x] \equiv \text{P}[-x < S_{(a,b)} < x] \tag{12}$$

This suggests that we can use signed skew as a proxy for the analysis of absolute skew. In other words, we can obtain the yield of skew ($\text{P}[\text{Skew}_{(a,b)} < x]$) by examining the yield of signed skew ($\text{P}[-x < S_{(a,b)} < x]$). In this section, we analyze the impact of variation on signed skew because of mathematically simpler analysis. Figure 2a illustrates a simple clock tree with one path per sink. In this case, the skew variance is

$$\sigma^2_{s(a,b)} = \text{var}(S_{(a,b)}) = \sigma^2_{p_a} + \sigma^2_{p_b} - 2\sigma_{p_a}\sigma_{p_b}\rho_{(p_a,p_b)} \tag{13}$$

We extend this analysis to clock networks with multiple paths for each sink node, as illustrated in Figure 2b. p_{w_2} is the shared path and p_{a_2}, p_{b_2} connect the shared path to the sinks a and b. From the multiple-path delay variation model from Section III-A, we obtain

$$D_{p(s,a)} = \big(D_{p_{a1}} + (D_{p_{w2}} + D_{p_{a2}})\big)/2 \tag{14}$$

$$D_{p(s,b)} = \big(D_{p_{b1}} + (D_{p_{w2}} + D_{p_{b2}})\big)/2 \tag{15}$$

Skew between a and b, and its variance can be expressed as

$$S_{(a,b)} = \big((D_{p_{a1}} + D_{p_{a2}}) - (D_{p_{b1}} + D_{p_{b2}})\big)/2 \tag{16}$$

$$\begin{aligned}\sigma^2_{s(a,b)} &= (\sigma^2_{p_{a1}} + \sigma^2_{p_{a2}} + \sigma^2_{p_{b1}} + \sigma^2_{p_{b2}})/4 \\ &+ (\sigma_{p_{a1}}\sigma_{p_{a2}}\rho_{(p_{a1},p_{a2})} + \sigma_{p_{b1}}\sigma_{p_{b2}}\rho_{(p_{b1},p_{b2})})/2 \\ &- \sum_{i=1}^{2}\sum_{j=1}^{2}(\sigma_{p_{ai}}\sigma_{p_{bj}}\rho_{(p_{ai},p_{bj})})/2 \end{aligned} \tag{17}$$

Example III.2 Consider the clock tree in Figure 2a with $\sigma^2_{p_a} = \sigma^2_{p_b} = 50$ and $\rho = 0$. Then from Formula 13, $\sigma^2_{s(a,b)} = 50 + 50 = 100$. We assume the variation constraint to be $15ps$ with yield 95% (i.e., $\text{P}[-15 < S_{(a,b)} < 15] > 95\%$, Figure 3). However, with $\sigma_{s(a,b)} = 10$, the probability is

$$\text{P}[-15 < S_{(a,b)} < 15] = 86.64\% \tag{18}$$

The current tree structure does not satisfy the given variation constraint. In this case, we can insert a new subtree and fuse it to the original tree to enhance robustness.

Example III.3 Consider adding a subtree with three paths (p_{w2}, p_{a2}, p_{b2}) to Figure 2a and build a fusion topology as in Figure 2b with $\sigma^2_{p_{w2}} = \sigma^2_{p_{a2}} = \sigma^2_{p_{b2}} = 25$. From Formula 17, $\sigma^2_{s(a,b)}$ reduces down to 37.5. Now the probability becomes

$$\text{P}[-15 < S_{(a,b)} < 15] = 98.5\% \tag{19}$$

which satisfies the given variation constraint.

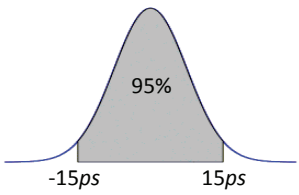
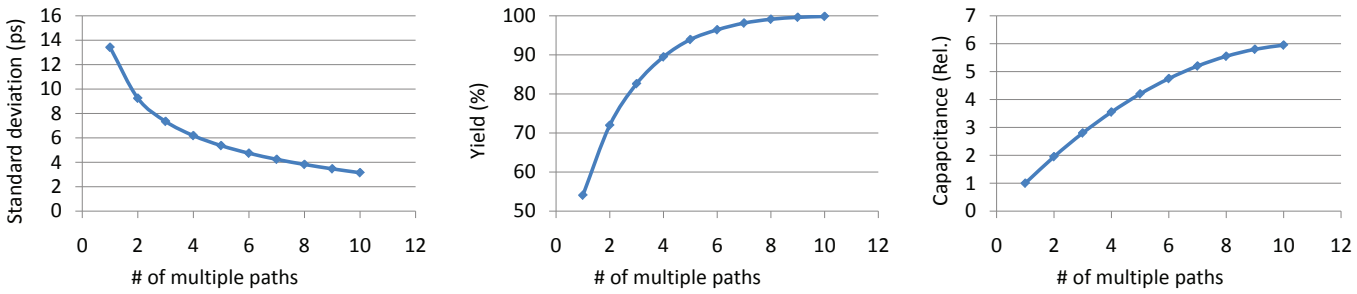


Fig. 3. Skew limit $15ps$ with yield 95%.



(a) (b) (c)

Fig. 4. The impact of redundant paths for a pair of critical sinks (Figure 2c) on clock-network parameters, based on Formulas 23, 25 and 26. The skew constraint and ρ are set to $10ps$ and 0.1 respectively. (a) Standard deviation. (b) Yield. (c) Relative total capacitance of each clock network compared to the total capacitance of the clock tree without redundant paths ($n = 1$).

C. Multiple redundant paths

We generalize the above analysis to clock networks with $n > 1$ redundant paths per sink as illustrated in Figure 2c.

$$D_{p(s,a)} = \big(D_{p_{a1}} + \sum_{i=2}^{n}(D_{p_{wi}} + D_{p_{ai}})\big)/n \tag{20}$$

$$D_{p(s,b)} = \big(D_{p_{b1}} + \sum_{i=2}^{n}(D_{p_{wi}} + D_{p_{bi}})\big)/n \tag{21}$$

$$S_{(a,b)} = \big(\sum_{i=1}^{n}(D_{p_{ai}} - D_{p_{bi}})\big)/n \tag{22}$$

$$\begin{aligned}\sigma^2_{s(a,b)} &= \sum_{i=1}^{n}(\sigma^2_{p_{ai}} + \sigma^2_{p_{bi}})/n^2 \\ &+2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(\sigma_{p_{ai}}\sigma_{p_{aj}}\rho_{(p_{ai},p_{aj})} + \sigma_{p_{bi}}\sigma_{p_{bj}}\rho_{(p_{bi},p_{bj})})/n^2 \\ &-2\sum_{i=1}^{n}\sum_{j=1}^{n}(\sigma_{p_{ai}}\sigma_{p_{bj}}\rho_{(p_{ai},p_{bj})})/n^2\end{aligned} \tag{23}$$

In the case when $\sigma_{p_{ai}} = \sigma_{p_{bi}} = \sigma$ and all ρ values are equal,

$$\sigma^2_{s(a,b)} = 2\sigma^2(1-\rho)/n \tag{24}$$

Just as in Formula 13, highly correlated path delays lead to small skew variance.

Example III.4 Figure 4 illustrates how n redundant paths for each sink (as in Figure 2c) reduce $\sigma_{s(a,b)}$ and increase yield (based on Formula 23). Here we assume

$$\sigma^2_{p_{ai}} = \sigma^2_{p_{bi}} = 100 - 10(i-1),\ 1 \le i \le 10 \tag{25}$$

$$\begin{aligned}&\mathrm{cap}(p_{a_i}) = \mathrm{cap}(p_{b_i}) = 100 - 10(i-1),\ 1 \le i \le 10 \\ &\mathrm{cap}(p_{w_1}) = 0,\ \mathrm{cap}(p_{w_i}) = 10,\ 2 \le i \le 10\end{aligned} \tag{26}$$

where $\mathrm{cap}(p)$ represents the capacitance of the path p.

In practice, we select *eligible* sinks a and b (see Section II-A) that maximize initial $\sigma^2_{s(a,b)}$. Thus $\rho_{(p_{a1},p_{b1})}$ will be small, but, for additional redundant paths, $\rho_{(p_{ai},p_{bi})}$ will be greater, especially when a and b are located close to each other. These paths are added so that $\rho_{(p_{ai},p_{aj})}$ and $\rho_{(p_{bi},p_{bj})}$ remain small. The same statistical analysis applies to process, voltage and temperature (PVT) variations.

Given a clock network Ψ, let σ be the standard deviation of *the most critical sink pair* in Ψ (i.e., $\sigma = \max_{(a,b)\in\mathcal{E}}(\sigma_{s(a,b)})$, where $\mathcal{E}$ is the set of eligible sink pairs), and skew_Ψ be the nominal skew of Ψ. If $\mathrm{skew}_\Psi \gg \sigma$, then the yield of Ψ is significantly affected by skew_Ψ. However, when $\mathrm{skew}_\Psi \ll \sigma$, the clock-network's yield is closely related to the yields of *critical sink pairs* (see Section IV-A). Our methodology invokes nominal skew optimizations to satisfy $\mathrm{skew}_\Psi \ll \sigma$ (see Section V). Therefore our proposed methods in Sections IV and V for enhancing robustness of critical sink pairs effectively increase the yield of Ψ.

IV. Multilevel tree fusion

Analysis in Section III suggests that one can reduce the impact of variation on clock skew by driving critical sinks through multiple redundant paths. To generalize, we propose a novel family of clock-network structures, called *fused multi-level trees*, which maintains advantages of tree structures and incrementally enhances robustness to variation by trading-off power and robustness.

A. Critical sink pairs

After performing initial-tree construction according to [11], we analyze the impact of variation on skew between eligible sink pairs. Using models from Section III-B, we can determine the variance and standard deviation of skew between each sink pair and detect critical sink pairs that are not robust enough with respect to given timing constraints. Eligible sink pairs are often geometrically close (or placed within the local skew distance limit in ISPD10 CNS benchmarks). However, they can be distant in the tree, i.e., the shortest tree-path connecting them can traverse many tree edges. These sinks are included in the set of critical sink pairs after variational analysis because the impact of variations accumulates on long paths, resulting in significant skew variance.

B. Construction of auxiliary trees and their fusion

Once we find all critical sink pairs, we cluster them based on their least common ancestors (LCA) in the tree. The pairs that

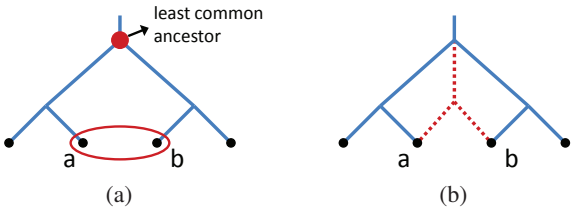


Fig. 5. (a) A critical sink pair is indicated by a red oval and the LCA of two sinks is shown. (b) Corresponding subtree for the sink cluster in (a).

share LCA are clustered, and a set of sinks is formed as the union of the sink pairs in the cluster. The LCA plays the role of the clock source for a new auxiliary tree that connects to the sinks in a given set. Here we use the same tree-construction algorithm that we used for initial tree construction.

The nominal delays of multiple redundant paths from the clock source to each critical sink must be carefully synchronized in order to reduce nominal skew in the fused topology. This process is discussed in detail in Section V-B. Figure 5 illustrates detection of critical sink pairs and the addition of auxiliary trees to enhance robustness.

After auxiliary trees are constructed and fused, we analyze the impact of variation on skew of eligible sink pairs again. Since there are multiple paths to some sinks, we utilize variation modeling from Section III-C. If some critical sinks remain, we construct another round of auxiliary trees and fuse them into the main network to enhance robustness. This robustness evaluation and tree construction/fusion process is repeated until we cannot find a critical sink pair anymore.

The success of our iterative fusion process critically depends on the precision of delay synchronization of redundant paths by clock-tree tuning. If implemented correctly, every fusion iteration significantly reduces the number of critical sinks (Tables IV and V), but if path synchronization fails, this improvement is not guaranteed. Figure 6 illustrates the proposed methodology including initial tree construction, detection of critical sink pairs and multilevel tree fusion.

C. Advantages of the multilevel tree fusion topology

The new clock-network structure is a joint of several trees that provides multiple redundant paths, helping to improve network robustness and satisfy skew constraints. Such a clock network exhibits the redundancy and robustness of a mesh but is easier to analyze and optimize. Our results in Section VI-B shows that fusion topologies can be essentially as robust as meshes, at a fraction of capacitance budget.

Multilevel tree fusion topology is technically not a tree structure because of interconnect loops. However, those loops always close at the sink nodes, which makes it easy to reduce not only the complexity of variational analysis but also nominal skew by various tree-based skew optimization techniques. Section V-B outlines the use of tree optimization techniques in this context.

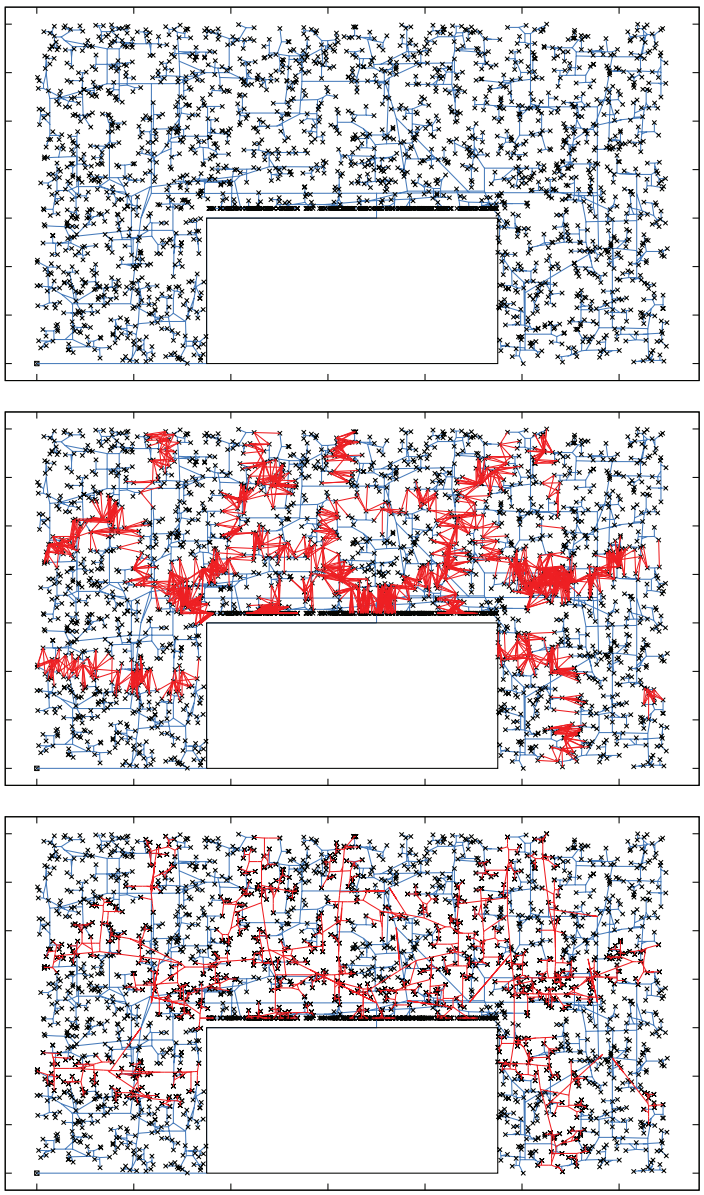


Fig. 6. Illustration of multilevel tree fusion on *ispd10cns02*. (a) Initial tree construction. (b) Critical sink pairs are connected by red lines. (c) Auxiliary trees are fused in to enhance robustness.

V. Implementation Insights

Figure 7 shows our methodology for multilevel tree fusion.

A. Estimating variation on a buffered path

After initial-tree construction [11], [12], we perform variational analysis based on the methods in Section III-B and build fusion topology to enhance robustness. For precise variational analysis, it is important to estimate Gaussian random variables for each buffered path. For accurate estimation of random variables, we build various test trees for given technology node, buffer and wire library and variation environment. Then we perform Monte-Carlo simulations with variation and record the variance of each buffered path in a look-up table. It is not

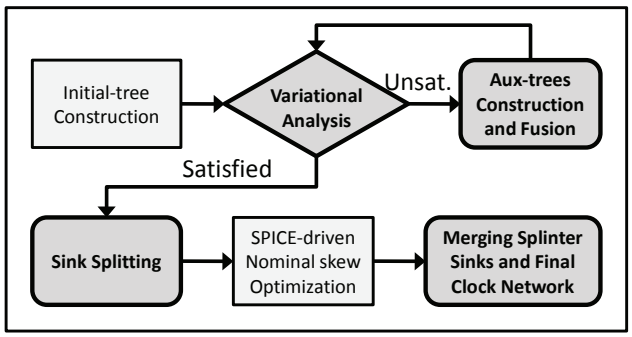


Fig. 7. Key steps of multilevel tree fusion. Proposed techniques are indicated with darker rounded boxes and a lozenge. Plain boxes represent techniques adapted from earlier publications.

necessary to record the mean of each random variable because our experimental results show that $\mathrm{E}[X]$ is nearly zero for all cases. The table is accessed by wirelength w and buffer count b to estimate the impact of variation on a buffered path with wirelength w and b buffers. Finally, the table is used to produce a least-squares fit F.

For a buffered path p of length w_p with b_p buffers,

$$\sigma_p^2 = \mathrm{F}(w_p, b_p) \tag{27}$$

With a Gaussian estimate of path delay, we analyze the impact of variation on eligible sink pairs and perform multilevel tree fusion as described in Section IV.

B. Splinter sinks

Since the initial and auxiliary trees are built using Elmore delay, they need to be tuned using more accurate delay calculations. Therefore we reduce skew by a SPICE-driven optimization process. Our novel clock-network structure is similar to traditional trees except for loops that close at critical sinks. To leverage the efficiency of existing tree-optimization techniques, we propose to split (clone) each critical sink and distribute its input capacitance among the resulting splinter sinks, as illustrated in Figure 8. Once splinter sinks are generated, there is no metal loop and our clock network becomes a tree, amenable to existing tree-optimization techniques. A key challenge is to correctly model nominal delays of multiple paths ending at the same sink, and then equalize them using tree-tuning techniques.

We adopted the slack computation and wiresnaking techniques described in [11] to reduce nominal skew measured by

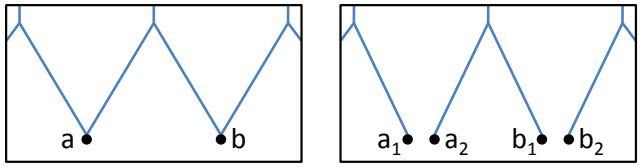


Fig. 8. (a) Multiple paths from clock source to sinks a and b. (b) Splinter sinks are generated to utilize tree optimization algorithms.

num. par. buf.	nominal skew (ps)	total skew				cap. (fF)
		mean (ps)	σ (ps)	yield (%)	95% (ps)	
8	2.082	5.81	1.18	92.4	7.75	26647
16	0.929	3.49	0.88	99.9	5.23	28093
24	1.843	3.16	0.80	99.9	4.70	32619

TABLE I
RESULTS OF CLOCK TREES ON *ispd10cns05* WITH PARALLEL BUFFERING. LOCAL SKEW LIMIT IS $7.5ps$ AS IN THE ISPD 2010 BENCHMARKS. THE STATISTICS OF NOMINAL SKEW, TOTAL SKEW ARE REPORTED BASED ON MONTE-CARLO SIMULATIONS. MEAN, STANDARD DEVIATION (σ) AND YIELD FOR GIVEN LOCAL SKEW LIMIT ARE REPORTED FOR EACH TREE. '95%' COLUMN REPRESENTS THE WORST LOCAL SKEW FOR 95% YIELD.

SPICE simulations. During SPICE-driven skew optimization, our goal is to make nominal skew as small as possible.

After nominal skew optimization, in the context of splinter sinks, the average nominal skew drops below $4ps$ on the ISPD 2010 CNS benchmarks. We merge splinter sinks to recover the fusion topology structure, at which point sink latencies may change and nominal skew may worsen. However, our experiments show that this deterioration can be limited to $2ps$ in the worst case.[2] The average nominal skew of fusion topologies on the ISPD 2010 CNS benchmarks is $2.55ps$.

VI. EMPIRICAL VALIDATION

Our empirical evaluation of multilevel tree fusion focuses on total capacitance and robustness to variations. We use ISPD 2010 CNS benchmarks but enhance their buffer library and variation setup to perform more realistic experiments.

A. Experiment design

ISPD 2010 CNS benchmarks are based on microprocessor designs from IBM and Intel and use a 45nm technology library. Each benchmark is given a local-skew limit and local skew distance bound. Result are evaluated by 500 Monte-Carlo simulations with a given variation model, with respect to a given yield constraint. ISPD 2010 benchmarks suffer from a recognized deficiency in the modeling of numerous parallel buffers (that may or may not appear in the clock network), which underestimates electrical parasitics and power overhead. Process variations are not spatially correlated, making parallel buffers completely independent and underestimating the impact of process variations. These deficiencies encourage unrealistic clock-network configurations. To this end, the best published results for the ISPD 2010 benchmarks [13] seem to require the stacking of numerous inverters in a unique 10+40 configuration. The authors attribute the quality of results to a new cross-link insertion technique, but do not report results without cross-link insertion to substantiate this claim. Results in [11] report even smaller skews but greater capacitance, but the authors also stack numerous (32) small inverters in parallel.

Table I illustrates how one can reduce the impact of process variation by only using excessive parallel buffers without any

[2]It is important to note that the number of splinter sinks for a given sink may increase by at most one during each fusion iteration. This significantly simplifies delay synchronization for redundant paths.

buffer type	in cap (fF)	out cap (fF)	out res Ω	distribut'n of proc. variation	σ (V)	parallel buffers allowed
ispd10b1	35	80	61.2	uniform	0.043	yes
ispd10b2	4.2	6.1	440	uniform	0.043	yes
our work	33.6	48.8	55	Gaussian	0.015	no

TABLE II
COMPARISON OF BUFFER TYPES. *ispd10b1* AND *ispd10b2* ARE TWO BUFFER TYPES IN ISPD 2010 CNS BENCHMARKS. THE LARGE BUFFER UTILIZED IN THIS WORK HAS GAUSSIAN VARIATION AND PARALLEL BUFFERING IS NOT ALLOWED. THE BUFFER TYPE IN THIS WORK IS INTENDED TO REPRESENT A COMPOSITE BUFFER MADE FROM 8 *ispd10b2* BUFFERS, BUT IN A WAY THAT WOULD PREVENT MODELING CONSTITUENT BUFFERS AS EXPERIENCING INDEPENDENT PVT VARIATION.

method	total skew			
	mean (ps)	95% (ps)	cap. (fF)	cap. Ratio
CNSRouter [26]	3.58	5.37	97421	2.30
[21]	-	5.47	228243	5.38
our work	2.17	5.06	42414	1.00

TABLE III
COMPARISON OF RESULTS ON *ispd10cns08* TO PUBLISHED DATA FOR MESHES. LOCAL SKEW LIMIT $6.0ps$ IS USED TO PRODUCE A CLOCK NETWORK WITH BETTER ROBUSTNESS THAN MESHES. OUR CLOCK NETWORK IS MORE ROBUST THAN MESHES BUT ALSO $2.30\times$ GREATER POWER EFFICIENT THAN CNSROUTER [26].

structural modification. It shows that competitive results on the ISPD 2010 benchmarks can be easily achieved by stacking only 16 small inverters in parallel.

We now propose a different experimental configuration to avoid major shortcomings of the ISPD 2010 benchmarks. First, instead of the ISPD 2010 buffer library that exhibits uniformly-distributed variation, we use a buffer type with Gaussian variation. Table II compares buffers used in the ISPD 2010 benchmarks and in this work.

By essentially clustering a reasonable number of small ISPD buffers into one large buffer we deliberately avoid parallel buffer stacking to prevent unrealistic modeling of constituent buffers as experiencing independent process variations. Unlike many previous publications, we limit our empirical validation to a single wire type to illustrate that proposed multilevel tree fusion can still produce high-quality clock networks. We also note that spatially-correlated variation is only responsible for a fraction of total variation, whereas random variation also makes a significant contribution. Thus, our experimental setup is pessimistic and serves to show that our proposed technique can achieve strong results even in adverse circumstances. Using one buffer type for clock-network synthesis also restricts the flexibility to allocate driver strength throughout the clock network. We use this limitation as a handicap in our experiments to highlight the strength of multilevel tree fusion.

B. Empirical results

Table IV shows empirical results on the *ispd10cns08* benchmark. We vary the local skew limit for the benchmarks to evaluate the flexibility of our novel clock-network structure. Once again, we use only one Gaussian buffer type without parallel stacking. When there is no local skew limit, the initial clock tree is left unchanged. To satisfy increasingly difficult skew constraints, additional auxiliary trees are generated and fused to enhance robustness of clock networks. Total clock-network capacitance increases as local skew limit decreases because the tree must become more robust. The statistics of variational skew are also shown in the table. Since nominal skew varies for each fusion topology, variational skew more correctly represents the impact of variations on skew. As shown in the table, variational skew consistently decreases as the robustness of fusion topologies is improved. The results show that the multilevel fusion topology exhibits sufficient flexibility to incrementally improve robustness based on variational analysis with given local skew limit. Compared to traditional tree structures, clock-network capacitance is increased by 59.5% to satisfy the difficult skew constraints with $4.5ps$ skew limit.

Table III compares our clock network with those produced by CNSRouter [26] and by techniques in [21]. Our clock network is more robust than meshes with significantly smaller total capacitance.

In Table V, we present our experimental results on *ispd10cns08* with more pessimistic modeling of process variations. In this experiment, the buffer type *ispd10b1* in Table II is utilized without parallel stacking. The purpose of this experiment is to verify how robust fusion topology is when the impact of variation is more significant than normal condition. Given that buffer delays are particularly affected by variation, the skew induced by variation is significant in the tree structure. However the results show that we can decidedly reduce the impact of variation by constructing additional auxiliary trees and fusing them into the main network.

VII. CONCLUSIONS

Clock network topologies described in the literature fall into several categories: (i) trees, (ii) meshes, (iii) trees with incrementally added cross-links, (iv) combinations of trees and meshes. The gap between tree-like and mesh-like topologies remains significant, and cross-links have not been convincingly shown to improve upon pure trees, due to known shortcomings of adding one cross-link at a time. In this work we propose, develop and empirically evaluate a fundamentally new family of clock-network topologies derived from trees by adding auxiliary trees and iteratively fusing them into the main network. Each fusion iteration balances a large subset of skew-critical clock sinks, but as auxiliary trees are much smaller than the initial tree, the added capacitance is also small. The accuracy of fusion iterations rests on the variational skew analysis techniques we proposed. The final clock-network topology averages out source-to-sink delay and cancels out some of the correlations induced by process, voltage and temperature (PVT) variations. Empirical evaluation shows strong results even with exceptionally pessimistic modeling of process variations, a single wire width and a single allowed buffer configuration without parallel stacking.

skew limit (ps)	nominal skew (ps)	total skew				variational skew			cap. (fF)	⏲ (s)
		mean (ps)	σ (ps)	yield (%)	95% (ps)	mean (ps)	σ (ps)	95% (ps)		
-	1.713	5.471	1.116	-	7.563	5.380	1.107	7.405	32580.4	781.0
7.5	2.673	5.295	0.991	97.6	7.159	5.048	1.032	6.945	37279.4	1100.9
7.0	2.294	4.788	0.931	97.8	6.325	4.456	0.952	6.046	40393.7	1721.8
6.5	1.967	4.275	0.883	98.4	5.822	3.884	0.890	5.533	41641.6	1787.6
6.0	2.171	3.740	0.757	99.0	5.06	3.423	0.820	4.918	42414.1	2192.8
5.5	2.639	3.851	0.754	97.2	5.29	3.411	0.793	4.834	44053.4	2204.5
5.0	2.020	3.211	0.673	99.0	4.508	2.723	0.751	4.220	48440.9	1913.1
4.5	2.115	2.993	0.647	97.0	4.125	2.485	0.655	3.711	51955.5	3900.9

TABLE IV

RESULTS ON *ispd10cns08* WITH DIFFERENT LOCAL SKEW LIMITS. THE STATISTICS OF NOMINAL SKEW, TOTAL SKEW AND VARIATIONAL SKEW ARE REPORTED BASED ON MONTE-CARLO SIMULATIONS. MEAN, STANDARD DEVIATION (σ) AND YIELD FOR GIVEN LOCAL SKEW LIMIT ARE REPORTED. '95%' COLUMN REPRESENTS THE WORST LOCAL SKEW WHEN YIELD IS 95%. ALL THE RESULTS SATISFY SLEW CONSTRAINTS.

skew limit (ps)	nominal skew (ps)	total skew				variational skew			cap. (fF)	⏲ (s)
		mean (ps)	σ (ps)	yield (%)	95% (ps)	mean (ps)	σ (ps)	95% (ps)		
-	0.980	16.47	2.619	-	22.086	16.46	2.595	21.88	37704.8	293.1
22	2.333	15.32	3.137	98.0	21.08	15.21	3.035	20.83	44093.6	339.6
20	4.081	14.39	2.545	97.4	19.03	14.14	2.619	19.09	46373.9	328.6
18	1.845	12.07	2.446	98.8	16.89	11.90	2.569	17.01	48153.0	469.9
16	3.317	10.84	2.068	99.2	14.07	10.62	2.093	14.15	49918.3	509.3
14	2.412	9.359	2.068	98.4	12.72	9.103	2.201	12.82	56374.6	746.6

TABLE V

RESULTS ON *ispd10cns08* WITH THE BUFFER TYPE *ispd10b1* IN TABLE II WITHOUT PARALLEL BUFFERING. THE STATISTICS OF NOMINAL SKEW, TOTAL SKEW AND VARIATIONAL SKEW ARE REPORTED BASED ON MONTE-CARLO SIMULATIONS. MEAN, STANDARD DEVIATION (σ) AND YIELD FOR GIVEN LOCAL SKEW LIMIT ARE REPORTED FOR EACH TREE. '95%' COLUMN REPRESENTS THE WORST LOCAL SKEW WHEN YIELD IS 95%. ALL THE RESULTS SATISFY SLEW CONSTRAINTS.

REFERENCES

[1] C. J. Alpert, D. P. Mehta, S. S. Sapatnekar, Eds., "Handbook of Algorithms for Physical Design Automation," *CRC Press*, 2009.

[2] H. Bakoglu, J. Walker, and J. Meindl, "A symmetric clock distribution tree and optimized high-speed interconnects for reduced clock skew in ULSI and WSI circuits," *ICCD*'86, pp. 118-122.

[3] K. D. Boese, A. B. Kahng, "Zero-Skew Clock Routing Trees with Minimum Wirelength," *ASIC*'92, pp.111-5.

[4] H. Chen, C. Yeh, G. Wilke, S. Reddy, H. Nguyen, W. Walker and R. Murgai, "A sliding window scheme for accurate clock mesh analysis," *ICCAD*'05.

[5] M. Cho, D. Z. Pan and R. Puri, "Novel Binary Linear Programming for High Performance Clock Mesh Synthesis," *ICCAD*'10, pp. 438-443.

[6] J. Cong, A. B. Kahng, G. Robins, "Matching-based Methods for High-performance Clock Routing," *IEEE Trans. on CAD* 12(8), pp.1157-1169, 1993.

[7] M. Edahiro, "A Clustering-Based Optimization Algorithm in Zero-Skew Routings," *DAC*'93, pp. 612-616.

[8] D. Harris, M. Horowitz and D. Liu, "Timing Analysis Including Clock Skew," *IEEE Trans. on CAD* 18(11), pp.1608-1618,1999.

[9] M. A. B. Jackson, A. Srinivasan, E. S. Kuh, "Clock Routing for High-performance ICs," *DAC*'90, pp. 573-579.

[10] A. Kahng, J. Cong, G. Robins, "High-performance clock routing based on recursive geometric matching," *DAC*'91, pp. 322-327.

[11] D.-J. Lee, M.-C. Kim and I. L. Markov, "Low-Power Clock Trees for CPUs," ICCAD, 2010, pp. 444-451.

[12] D.-J. Lee, I. L. Markov, "Contango: Integrated Optimization of SoC Clock Networks," *DATE*'10, pp. 1468-1473.

[13] T. Mittal and C.-K. Koh, "Cross Link Insertion for Improving Tolerance to Variations in Clock Network Synthesis," *ISPD*'11.

[14] W.-H Liu, Y.-L Li, H.-C. Chen,"Minimizing Clock Latency Range in Robust Clock Tree Synthesis,"*ASPDAC*'10,pp.389.

[15] J. Lu et al, "A Dual-MST Approach for Clock Network Synthesis," *ASPDAC*'10, pp. 467-473.

[16] A. Rajaram, J. Hu, and R. Mahapatra, "Reducing Clock Skew Variability via Cross Links," *DAC*'04, pp. 18-23.

[17] A. Rajaram and D. Z. Pan, "Variation Tolerant Buffered Clock Network Synthesis with Cross Links," *ISPD*'06, pp. 157-164.

[18] A. Rajaram, D.Z. Pan, and J. Hu, "Improved Algorithms for Link-Based Non-Tree Clock Networks for Skew Variability," *ISPD*'05, pp. 55-62.

[19] P. J. Restle et al, "A Clock Distribution Network for Microprocessors," *IEEE JSSC* 36(5), pp.792-799,2001.

[20] X.-W. Shih et al, "Blockage-Avoiding Buffered Clock-Tree Synthesis for Clock Latency-Range and Skew Minimization," *ASPDAC*'10, pp. 395-400.

[21] X.-W. Shih, H.-C. Lee, K.-H. Ho and Y.-W. Chang, "High Variation-Tolerant Obstacle-Avoiding Clock Mesh Synthesis with Symmetrical Driving Trees," *ICCAD*'10, pp. 452-457.

[22] C. Sze, P. Restle, G.-J. Nam, C. J. Alpert, "ISPD 2009 Clock-network Synthesis Contest," *ISPD*'09, pp. 149-150.

[23] C.N. Sze,"ISPD 2010 High-Performance Clock Network Synthesis Contest: Benchmark Suite and Results," *ISPD*'10, pp.143.

[24] G. Venkataraman, Z. Feng, J. Hu, P. Li,"Combinatorial algorithms for fast clock mesh optimization,"*ICCAD*'06,pp.563-567.

[25] G. Venkataraman et.al,"Practical Techniques for Minimizing Skew and its Variation in Buffered Clock Networks," *ICCAD*'05, pp. 592-596.

[26] L. Xiao et al, "Local Clock Skew Minimization Using Blockage-aware Mixed Tree-Mesh Clock Network," *ICCAD*'10, pp. 458-462.

[27] X. Ye, P. Li, M. Zhao, R. Panda and J. Hu, "Analysis of large clock meshes via harmonic-weighted model order reduction and port sliding," *ICCAD*'07.

[28] C. Yeh et al, "Clock Distribution Architectures: A Comparative Study," *ISQED*'06, pp. 85-91.