Algorithmic Tuning of Clock Trees and Derived Non-Tree Structures

Invited Paper

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

***Abstract*—This mini-tutorial covers recent research on clock-network tuning. It starts with SPICE-accurate optimizations used in winning entries at the ISPD 2009 and 2010 clock-network synthesis contests. After comparing clock trees to meshes, it outlines a recent redundant clock-network topology that retains most advantages of clock trees, but improves robustness to PVT variations. It also shows how to incorporate clock-network synthesis into global placement to reduce dynamic power and insertion delay.**

I. INTRODUCTION

Clock-network synthesis significantly impacts the performance, area and power dissipation of an integrated circuit, and this impact is aggravated when architectural-level pipelining increases the number of clocked elements [35]. Being responsible for 30-50% of chip power [9], clock networks require careful optimization. Such optimization may require SPICE-accurate timing analysis, otherwise it is difficult to account for the explosion of technology parameters that characterize each advanced technology node. Unfortunately, runtime overhead limits such analysis to only several invocations during optimization. Recent research reviewed in this paper developed comprehensive SPICE-accurate clock-network optimizations subject to capacitance and slew constraints, and cognizant of runtime limitations. The ISPD 2009 and 2010 Clock-Network Synthesis contests organized by IBM Research and Intel Research evaluated these techniques on industry benchmarks. The second contest increased the realism in modeling and evaluation of clock networks, adding the impact of variations and using *local* rather than *global* clock skew — a more meaningful parameter for large circuits [10].

The growing impact of process, voltage and temperature (PVT) variation complicates the design of reliable clock networks [26], as nominal-parameter optimizations (single corner, no variation) do not ensure high yield. Robustness to variations requires increased power and careful trade-offs. Mesh/grid structures exhibit greater robustness but consume much more power and complicate clock gating. This paper surveys recent clock-network topologies that offer the path-redundancy of meshes, but retain the advantages of clock trees.

Clock networks can also be improved by careful positioning of latches and flip-flops. Leaf-level register clustering [3] after global placement can help sharing inverters between flip-flops and reduce clock-network capacitance. In this paper, we review a recent, more general optimization [17] that differentiates sequential elements during global placement and optimizes total dynamic power of signal nets and the clock network.

II. CLOCK TREES

When clock sinks are distributed uniformly and have similar capacitances, the classic *H-tree* technique [1] can be used. Otherwise, sinks can be clustered by the *method of means and medians* (MMM) [11] or the *geometric matching algorithm* (GMA) [7]. Given a sink clustering, the *deferred merge embedding* (DME) algorithm [2] produces minimum-length trees where each source-to-sink path has the same length [12]. A clock-synthesis methodology for SPICE-accurate skew optimization with tolerance to voltage variations was proposed in [14]. 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 [29]. A three-stage CTS flow based on an obstacle-avoiding balanced clock-tree routing algorithm with monotonic buffer insertion is proposed in [18]. A Dual-MST (DMST) geometric matching approach is proposed in [19] for topology construction and recursive buffer insertion. Modeling techniques and algorithms for CPU-grade clock-power optimization in the presence of variations can be found in [15].

Nominal-skew optimization. Clock network design is difficult because the optimization objectives are not available in closed form. To limit runtime overhead of accurate delay models, one constructs zero-skew clock trees using simple analytical delay models, then performs initial fast buffer insertion. Further optimizations can be driven by SPICE, Arnoldi approximations, or other available tools/models. After invoking the ZST-DME algorithm [8], [12] initial buffer insertion minimizes source-to-sink Elmore delay, rather than skew or capacitance [27]. Elmore delay is too inaccurate for skew optimization, but there is significant room for tuning the clock tree by delaying fast paths [14]. In the presence of layout obstacles, a simple and robust technique for obstacle avoidance in clock trees from [14] repairs obstacle violations in ZST-DME trees. Modifying the ZST-DME algorithm to handle obstacles directly is laborious and offers little advantage in practice because ZST-DME trees avoid most obstacles. Shorter BST-DME trees incur greater additional capacitance during subsequent tuning, which negates their perceived benefits.

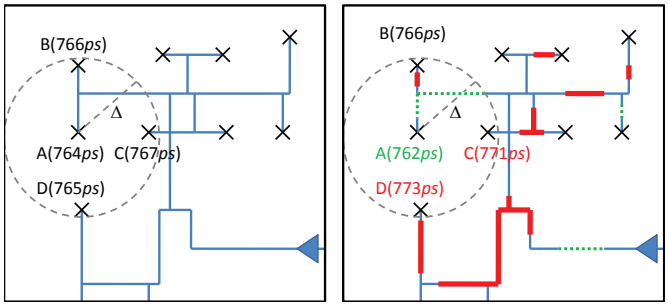
Individual clock-tree optimizations differ by their strength/range and accuracy. To coordinate them, start with optimizations of greatest range and gradually transition to those with greater accuracy [14]. *Each step should decrease the main optimization objective sufficiently to be within the range of the next optimization.*

Clock trees can be tuned with minimal capacitance overhead using the concept of *local-skew slack* [15]. During optimizations such as wire snaking [14], the optimal tuning amount for each edge can be obtained by the top-down slack computation explained in [15]. Let $T_{target}(e)$ be the amount of time in ps by which the clock-tree edge e must be delayed to satisfy local skew constraints and let $T_{actual}(e)$ be the amount of time in ps which the edge e is actually delayed by wire snaking. the wire snaking algorithm must satisfies $T_{actual}(e) \leq T_{target}(e)$ for optimized power consumption by the clock tree. Additional iterations will be needed if the gap between $T_{actual}(e)$ and $T_{target}(e)$ is too big. To keep $T_{actual}(e) \leq T_{target}(e)$ with optimal quality, we define α where,

$$\alpha \leq T_{actual}(e)/T_{target}(e) \leq 1.0 \quad (1)$$

Wire snaking aims for $T^i_{actual}(e)$ to satisfy the above inequality with the highest α possible. To achieve the highest α, the delay model based on look-up tables and the technique to select optimal nodes for wire snaking are explained in [15]. When α is specified, the required worst-case number of iterations of wire snaking N to make $T_{actual}(e)$ to $T_{target}(e)$ within error rate ε is $N = \left\lceil \log(\varepsilon)/\log(1-\alpha) \right\rceil$. Other skew-optimization techniques — delay buffer insertion, wire sizing, bottom-level tuning — are explained in [14], [15].

Robustness improvement. To avoid long Monte-Carlo simulations, recent research developed a tabular technique to account for variation in single-shot timing analysis [15]. A key insight is that the impact of variations on skew between two sinks is closely correlated with tree-path length and the types of buffers used. When two sinks can be connected by a short path in the tree, variation of skew between them is small. On the other hand, variational skew between sinks that are geometrically close can be significant if the unique tree-path between them is long (see Figure 1). For a given technology node, buffer library, wires and variation model, one can build a look-up table that encodes the impact of variations on skew along sink-to-sink paths of a given length with given buffer types [15]. This table can be used during initial buffering [15], but subsequent clock-tree tuning is required to satisfy tight local-skew constraints for a particular tree [15].



(a) Nominal sink latencies (b) Latencies with variation

Fig. 1. The impact of variations on local skew. Sinks are indicated by crosses, the clock source is indicated by a solid triangle. Nominal skew of 3 ps is shown in (a). Full skew of 11 ps is shown in (b), where some tree edges are delayed (thick red) and some are sped up (dotted green) by random variations. Only sink A is within the local skew distance from sinks B, C and D. Source: [15].

III. Meshes and Cross-Links

The choice between a tree and non-tree topology is a central question in modern clock-network design. High-performance CPUs 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, as well as amenability to incremental tuning and clock gating.

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 [34]. 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 important than power consumption [12]. Methods to analyze the characteristics of mesh structures are proposed in [4], [33] and a combinatorial algorithm to optimize a clock mesh is proposed in [30]. Obstacle-avoiding clock mesh synthesis in [28], [32] applies a two-stage approach — mesh construction followed by driving-tree synthesis. A clock-mesh synthesis methodology based on binary linear programming is described in [6].

Trees with cross-links. The dichotomy between meshes and trees is striking, and several researchers attempted to find intermediate topologies that would retain the advantages of meshes but reduce capacitance overhead. A 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 [24]. These methods are later extended to handle buffered clock trees in [25]. 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 despite diligent attempts, as improved tree-tuning methods were sufficient. Careful experiments and analytical estimates [22] 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 [22] suggests adding cross-links higher in the tree to connect entire branches.

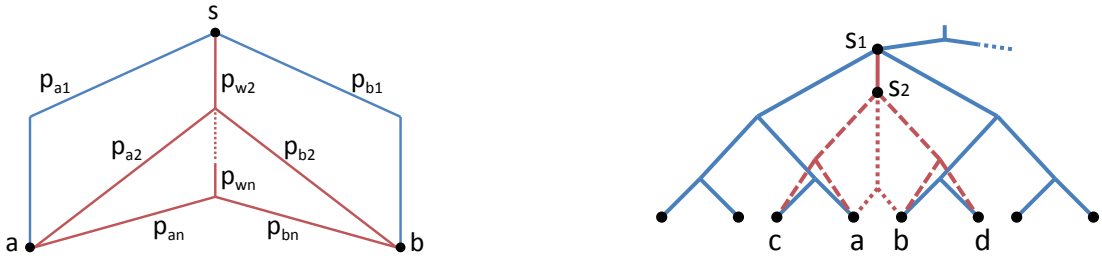


Fig. 2. (a) A clock-network fragment with source node s, two sink nodes a and b, as well as n redundant paths for each sink. Each i-th ($2 \leq i$) new source-to-sink path consists of a shared p_{w_i} section and a p_{a_i} or p_{b_i} section that is not shared. (b) The first-level subtree is illustrated with red dashed lines and the second-level subtree is shown with red dotted lines. s_1=LCA(a, b, c, d) and s_2=LCA(a, b).

IV. Multilevel tree fusion

We now describe a recently proposed family of clock-network topologies that retain most advantages of tree structures, but are more robust to variations.

Impact of variation on skew. 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 $D_p = N(d_p, \sigma_p^2)$. Let s be a source node and a, b be two sink nodes. *Nominal skew* (without variation) is defined as $\text{skew}_{(a,b)} = |d_{p(s,a)} - d_{p(s,b)}|$. We define *total signed skew* (with variation) and *total absolute skew* as follows.

$$S_{(a,b)} = D_{p(s,a)} - D_{p(s,b)}, \quad \text{Skew}_{(a,b)} = |S_{(a,b)}| \qquad (2)$$

The mean and variance of $\text{Skew}_{(a,b)}$ can be derived from the mean μ and variance σ of $S_{(a,b)}$ as explained in [16]. For a clock network with $n > 1$ redundant paths per sink (illustrated in Figure 2a), one can obtain

$$\begin{aligned}\sigma^2_{s(a,b)} &= \sum_{i=1}^{n}(\sigma^2_{p_{ai}} + \sigma^2_{p_{bi}})/n^2 \\ &+2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(\sigma_{p_{ai}}\sigma_{p_{aj}}\rho_{(p_{ai},p_{aj})} + \sigma_{p_{bi}}\sigma_{p_{bj}}\rho_{(p_{bi},p_{bj})})/n^2 \\ &-2\sum_{i=1}^{n}\sum_{j=1}^{n}(\sigma_{p_{ai}}\sigma_{p_{bj}}\rho_{(p_{ai},p_{bj})})/n^2 \qquad (3)\end{aligned}$$

where $0 \leq \rho_{(p_a,p_b)} \leq 1$ is the correlation between D_{p_a}, D_{p_b}. Thus, the impact of variation on clock skew can be reduced by driving critical sinks through multiple redundant paths.

Construction of auxiliary trees and their fusion. After performing initial-tree construction according to [15], the impact of variation on skew between eligible sink pairs is analyzed [16]. Eligible sink pairs are often geometrically close, but 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. After finding all critical sink pairs, one clusters them based on their least common ancestors (LCAs) in the tree. Each LCA acts as a clock source for a new auxiliary tree that drives the sinks in a given cluster (see Figure 2b). 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 and limit short-circuit power.

Splinter sinks. The clock-network structure proposed in [16] is similar to traditional trees except for possible cycles that close at critical sinks. To leverage the efficacy of existing tree-optimization techniques [15], each critical sink is split (cloned) and its input capacitance is distributed among the resulting splinter sinks. Once splinter sinks are generated, there is no metal loop and the 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. The slack computation and wiresnaking techniques described in [14], [15] can be adapted to reduce nominal skew measured by SPICE simulations.

Splinter sinks are merged to recover the fusion topology structure, at which point sink latencies may change. Figure 3 illustrates SPICE waveforms at a worst-case reconvergent sink and its splinter sinks — the averaging of waveforms is apparent. Due to this averaging effect, nominal skew optimization with splinter-sinks is both reliable and effective. Further details are discussed in [16].

Empirical evaluation in [16] indicates that the multilevel fusion topology exhibits sufficient flexibility to incrementally improve clock-network robustness based on variational analysis. A given local skew limit can be satisfied with only a modest increase in total capacitance.

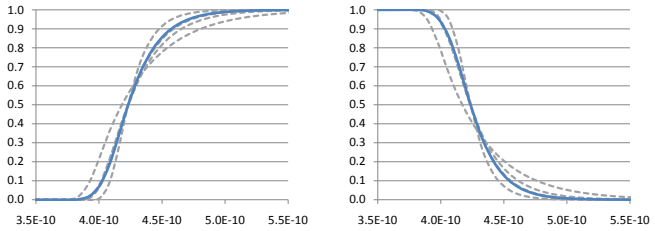


Fig. 3. SPICE waveforms for a reconvergent sink with max temporal displacement between drivers (in a fused-tree clock network for *ispd10cns08* with skew limit 4.5 ps). Dashed waveforms correspond to individual drivers, the cumulative waveform at the sink is shown with a blue solid line. An averaging effect can be observed.

V. Interactions between placement and clock-network synthesis

Dynamic power is a top contributor to total IC power, and clock networks can be responsible for over 30% of total (peak) power consumption in CPUs due to their high capacitance and frequent switching [21]. The size and power of a clock network is greatly affected by how spread out the register placement is. However, modifying register placement by pre- or post-processing existing placement techniques is difficult. Indeed, most appropriate changes to cell locations that reduce the clock network may depend on the current structure of the clock network, which existing placement tools ignore. However, over-emphasizing the placement of clock sinks may elongate signal nets and undermine their timing. To this end, Lu [20] proposed Manhattan ring-based register guidance in global placement, center-of-gravity constraints for registers, pseudo-pins and register-cluster contraction. Cheon [5] proposed power-aware placement that performs both activity-based register clustering and activity-based net weighting to simultaneously reduce the clock and signal net-switching power. In order to reduce the clock network size, Wang [31] proposed dynamic clock-tree building (DCTB), multi level bounding box (MLBB) and multi level attractive force (MLAF), and integrated them into a force-directed placement (FDP) framework. A major challenge in developing such techniques is the need to combine state-of-the-art in global placement and clock-network construction without undermining respective algorithms. Such a combination is reported in [17]. It modifies a global placer to optimize total dynamic power of signal nets and the emerging clock-network. To shorten clock wires, obstacle-aware virtual clock-network synthesis in [17] builds an intermediate clock network during placement. This clock network then defines *an arboreal clock-net contraction force* between clock-tree nodes. Branching nodes in the clock network split it into individual edges, seen as different nets by the placement algorithm. These zero-area branching nodes do not affect area distribution during global placement. Experimental results in [17] indicate that the software implementation, called Lopper, prunes clock-tree branches to reduce their length by 30.0%$\sim$36.6% and average total dynamic power consumption by 6.8%$\sim$11.6% versus conventional wirelength-driven approaches. SPICE-driven simulations show that Lopper makes clock trees less sensitive to variations and reduces insertion delay.

References

[1] 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.

[2] K. D. Boese, A. B. Kahng, "Zero-Skew Clock Routing Trees with Minimum Wirelength," *ASIC*'92, pp.111-115.

[3] Y.-T. Chang et al, "Post-Placement Power Optimization with Multi-Bit Flip-Flops," *ICCAD*'10, pp. 218-223.

[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* 2005, pp. 939-946.

[5] Y. Cheon, P.-H. Ho, A. B. Kahng, S. Reda and Q. Wang, "Power-aware placement," *DAC* 2005, pp. 795-800.

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

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

[8] J. Cong, A. B. Kahng, C.-K. Koh, C.-W. A. Tsao, "Bounded-Skew Clock and Steiner Routing," *ACM TODAES* 1998, pp. 341-388.

[9] D. E. Duarte, N. Vijaykrishnan and M. J. Irwin, "A clock power model to evaluate impact of architectural and technology optimization," *IEEE TVLSI*, 10(6): 844-855, Dec. 2002.

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

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

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

[13] W.-C. D. Lam et al, "Statistical Based Link Insertion for Robust Clock Network Design," *ICCAD*'05, pp. 588-591.

[14] D.-J. Lee and I. L. Markov, "Contango: Integrated Optimizations for SoC Clock Networks", VLSI Design 2011, no. 407507, 12 pp.

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

[16] D.-J. Lee and I. L. Markov, "Multilevel Tree Fusion for Robust Clock Networks," ICCAD, 2011.

[17] D.-J. Lee, M.-C. Kim and I. L.Markov, "Obstacle-aware Clock-tree Shaping during Placement," ISPD 2011, pp. 123-130.

[18] W.-H Liu, Y.-L Li, H.-C. Chen, "Minimizing Clock Latency Range in Robust Clock Tree Synthesis," *ASPDAC*'10, pp. 389-394.

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

[20] Y. Lu et al, "Navigating Registers in Placement for Clock Network Minimization," *DAC* 2005, pp. 176-181.

[21] N. Magen, A. Kolodny, U. Weiser, and N. Shamir, "Interconnect-power Dissipation in a Microprocessor," *SLIP* 2004, pp. 7-13.

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

[23] J. M. Rabaey, A. Chandrakasan, B. Nikolic, "Digital Integrated Circuits: A Design Perspective," *Prentice Hall*, Second Edition, 2003.

[24] 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.

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

[26] R. S. Shelar, "An Algorithm for Routing with Capacitance/Distance Constraints for Clock Distribution in Microprocessors," *ISPD*'09, pp. 141-148.

[27] W. Shi, Z. Li, "A Fast Algorithm for Optimal Buffer Insertion," *IEEE TCAD* 24(6), pp.879-891,2005.

[28] 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.

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

[30] G. Venkataraman, Z. Feng, J. Hu and P. Li, "Combinatorial Algorithms for Fast Clock Mesh Optimization," IEEE/ACM International Conference on Computer-Aided Design, 2006, pp. 563-567.

[31] Y. Wang, Q. Zhou, X. Hong and Y. Cai, "Clock-Tree Aware Placement Based on Dynamic Clock-Tree Building," *ISCAS*'07, pp. 2040-2043.

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

[33] 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.

[34] C. Yeh et al, "Clock Distribution Architectures: A Comparative Study," *ISQED* 2006, pp. 85-91.

[35] J. Yuan and C. Svensson, "High-speed CMOS circuit technique," *IEEE J. Solid-State Circuits*, vol. 24, pp. 62-70, Jan. 1989.