Obstacle-aware Clock-tree Shaping during Placement

Dong-Jin Lee
University of Michigan
2260 Hayward Street
Ann Arbor, MI 48109-2121
ejdjsy@umich.edu

Igor L. Markov
University of Michigan
2260 Hayward Street
Ann Arbor, MI 48109-2121
imarkov@eecs.umich.edu

ABSTRACT

Traditional IC design flows optimize clock networks before signal-net routing and are limited by the quality of register placement. Existing publications also reflect this bias and focus mostly on clock routing. The few known techniques for register placement exhibit significant limitations and do not account for recent progress in large-scale placement and obstacle-aware clock-network synthesis.

In this work, we integrate clock network synthesis within global placement by optimizing register locations. We propose the following techniques: (1) obstacle-aware virtual clock-tree synthesis; (2) arboreal clock-net contraction force with virtual-node insertion, which can handle multiple clock domains and gated clocks; (3) an obstacle-avoidance force. Our work is validated on large-size benchmarks with numerous macro blocks. Experimental results show that our software implementation, called Lopper, prunes clock-tree branches to reduce their length by 30.0%∼36.6% and average total dynamic power consumption by 6.8%∼11.6% versus conventional approaches.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids—*placement and routing*

General Terms

Algorithms, Design

1. INTRODUCTION

Power consumption is one of the primary optimization objectives for modern IC designs [21]. It includes three basic components: *short-circuit* power, *leakage* power and *net-switching* power [13]. Net-switching power is usually the largest contributor, and clock networks are often responsible for over 30% of total power consumption due to their high capacitance and frequent switching [5, 6, 16, 26]. The quality of clock networks is greatly affected by register placement, but mainstream literature on placement and most commercial EDA tools have largely overlooked this fact by focusing on wirelength of signal nets [10], routability [29] and circuit timing [7]. As far as we know, high-quality register placement cannot be achieved by easy pre- or post-processing of existing techniques. To this end, most appropriate changes to cell locations that reduce the clock network may depend on the current structure of the clock network, which is not accounted for in existing placement tools.

Our analysis of prior work reveals serious limitations in published techniques. Some methods coerce the placer into shortening the clock tree by capturing portions of the clock tree with the half-perimeter wirelength (HPWL) objective, which is usually applied only to signal nets [4, 30]. This idea overlooks the fact that low-skew clock trees exhibit much greater wirelength than signal nets with the same bounding box. To make matters worse, the HPWL estimate does not offer much fidelity for clock-tree lengths, as we show in Figure 2. Furthermore, a handful of existing publications that optimize clock networks during placement (reviewed in Section 2) do not reflect recent progress in large-scale placement and clock-network synthesis, and do not compare their results with best-of-breed software. In most cases, they are evaluated on small benchmarks without routing/buffering obstacles rather than on modern ASIC or SoC designs with many macro blocks. *Our research addresses these gaps in the literature by developing a set of new techniques for clock-net optimization during placement and evaluating these techniques against leading academic software.* We extended the ISPD 2005 benchmark suite toward clock-network synthesis, with the largest benchmark including 2.1M standard cells and 327K registers. The benchmarks include numerous macros, which we interpret as routing obstacles.

To optimize the trade-off between clock network minimization and traditional placement objectives, we propose a new placement methodology based on *obstacle-aware virtual clock-tree synthesis* that extends force-directed placement by adding *a arboreal clock-net force* using virtual nodes. *A key challenge addressed in our work is preserving the quality of global placement when adding clock-net optimizations.* We also accommodate multiple clock domains and gated clocks. Our algorithms are integrated into the SimPL placer [9], which currently produces lowest-wirelength placements on the ISPD'05 benchmarks. The quality of register placement is evaluated by Contango 2.0 [12] – the winner of the ISPD 2010 contest. Experimental results show that our method can reduce clock-network capacitance by 30.0%∼36.6% while reducing the overall dynamic power of the IC by 6.8%∼11.6% compared to conventional approaches.

ISPD'11, March 27–30, 2011, Santa Barbara, California, USA.
Copyright 2011 ACM 978-1-4503-0550-1/11/03 ...$10.00.

2. PRIOR WORK

Recent clock-network synthesis tools often construct initial trees with a simple delay model (e.g., Elmore) and then perform SPICE-accurate tuning [11, 12, 14, 22].

Clock-network optimization after placement can be performed by clustering nearby flip-flops [3, 20] to share inverters (inside flip-flops) and shorten the clock tree. This clustering does not adversely affect signal nets, but is rather limited by the locations of combinational gates. In high-performance CPUs flip-flops are often replaced by single latches, which reduces savings from clock-sink clustering.

Clock-network optimization during placement. To address the apparent conflict between clock-net optimization and traditional placement objectives, some researchers proposed techniques and algorithms for better register placement without intrusive interference in traditional placement objectives. Lu [15] proposed several techniques including Manhattan ring-based register guidance, center-of-gravity constraints for registers, pseudo-pins and register-cluster contraction. Cheon [4] 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 [30] 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 [28].

Limitations of existing techniques. Clock-net optimization during placement seeks better register locations but should not harm total wirelength of signal nets. A naive method is to increase the weight of the clock net and pull all registers together. Unfortunately, this method increases routing congestion and hot spots, and also leads to poor signal-net wirelength when dealing with more than several hundred registers [4, 30]. To definitively resolve the conflict between clock-net minimization and traditional placement objectives, careful problem formulation is essential.

Prior approaches to clock-net minimization in placement form two families. *Manhattan-ring guidance methods* commit registers to certain guidance locations and try to pull the registers close to the nearest such locations during placement [15]. However, such methods do poorly in the presence of numerous obstacles, e.g., macro-blocks, or when register locations found by the global placer are not uniformly distributed. In other words, guidance rings cannot accurately predict ideal locations for register clusters. Figure 1 illustrates how Manhattan-ring methods fail. In Figure 1(b), the sink group A is attracted by the closest Manhattan ring. The sinks in A are erroneously guided toward the obstacle. The sink group B and the related standard cells have heavy connections to the bottom macro block. However, the two bottom Manhattan rings encourage the sinks in B to move away from the center of B, which will likely increase signal-net wirelength significantly.

The second family of approaches performs clock-network synthesis using register locations from intermediate placement results. Specific techniques [4, 30] often simplify the structure of the clock network and bias the placement process to optimize such simplified networks. However, clock trees generated by those techniques are not realistic and very different from those generated by leading software. In the DCTB algorithm [30], the essential parameters of clock network synthesis, such as sink capacitance and wire capacitance/resistance, are ignored, and the cost function is derived by only considering Manhattan length between sinks or nodes. The quick CTS algorithm in [4] is also much simpler than standard DME algorithms, which minimize wirelength with zero or bounded skew based on Elmore delay. Furthermore, all previous work ignores the presence of routing obstacles, common in modern IC designs, and this ignorance can undermine end results (Sections 4 and 6).

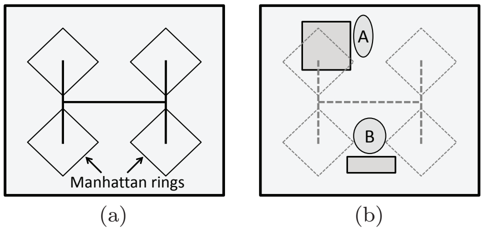


Figure 1: Two examples of Manhattan rings proposed in [15]. (a) Zero-skew Manhattan rings driven by an H-tree. (b) Manhattan rings on the design with obstacles. Obstacles are indicated by darker boxes, two sink groups (A, B) are represented as ellipses. The locations of the sinks are assumed to be decided based on wirelength-driven placement.

Previous publications that simplify clock-tree synthesis during placement [4, 30] cluster clock trees and represent these clusters with bounding boxes to model clock network reduction by placement objectives. Typically, registers are clustered at one or multiple levels based on the structure of the reference (simplified) clock tree, and bounding boxes are created for each cluster. The experimental results of [4, 30] show that bounding boxes are helpful for clock-net size reduction. However, we argue below that this method fails to represent clock-net reduction problem in placement.

Bounding boxes are represented by fake nets during placement and are optimized to reduce HPWL [9, 24]. The HPWL objective is relevant to placement because it estimates the lengths of signal routes reasonably well. However, clock routing is very different from signal-net routing and requires longer routes to ensure low skew. Therefore, HPWL does not offer accurate estimates of clock-tree lengths. Figure 2 shows that reducing HPWL of the clock net may increase the total length of the clock tree, demonstrating that the HPWL estimates lack not only accuracy, but also fidelity.

The authors of [30] adapted MLAF to compensate for the drawback of MLBB. However, we show in Section 4.2 that MLAF offers only a partial solution to this problem.

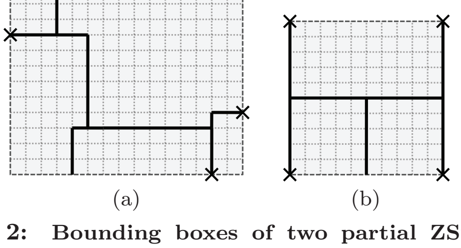


Figure 2: Bounding boxes of two partial ZST-DME clock trees. (a) HPWL of the bounding box is (15+12)=27. The total wirelength of the inside clock tree is 32. (b) HPWL is (10+10)=20 and the total wirelength of the clock tree is 35. The clock-net wirelength of (b) is greater than (a) although the bounding-box HPWL of (b) is notably smaller than (a) while the source-to-sink wirelength is 15 for all sinks.

3. OPTIMIZATION OBJECTIVE

Let $\mathcal{N}$ be the set of signal nets, and let $\mathcal{E}$ be the set of clock-net edges. To optimize clock networks in placement, we minimize the total switching power P_{sw}, defined as the sum of $\mathcal{N}$'s switching power $P_{\mathcal{N}}$ and $\mathcal{E}$'s switching power $P_{\mathcal{E}}$

$$P_{sw} = P_{\mathcal{N}} + P_{\mathcal{E}} \tag{1}$$

If activity factors of signal nets and clock-net edges are available, then the total signal-net switching power is

$$P_{\mathcal{N}} = \sum_{n_i \in \mathcal{N}} \alpha_{n_i} HPWL_{n_i} C_n V^2 f \tag{2}$$

and the total clock-net switching power is

$$P_{\mathcal{E}} = \sum_{e_i \in \mathcal{E}} \alpha_{e_i} L_{e_i} C_e V^2 f \tag{3}$$

Here, α_{n_i} and α_{e_i} are the respective signal-net and clock-edge activity factors, C_n and C_e are the respective unit capacitance for signal and clock wires, V is the supply voltage, f is the clock frequency, $HPWL_{n_i}$ is the HPWL of net n_i, and L_{e_i} is the Manhattan length of edge e_i. Activity factors of clock-net edges are required when multiple clock domains or gated clocks are utilized for given designs, otherwise $\alpha_{e_i} = 1$ as clock edges switch every clock cycle. The handling of gated clocks is discussed in Section 5 in more detail. If the activity factors of signal nets are not available, the computation of total switching power relies on *clock-power ratio* β, i.e., clock-net switching power divided by total switching power. In this case, the average activity factor of signal-net α_{avg} can be derived as

$$\alpha_{avg} = \frac{(1-\beta)\sum_{e_i \in \mathcal{E}} L_{e_i} C_e}{\beta \sum_{n_i \in \mathcal{N}} HPWL_{n_i} C_n} \tag{4}$$

α_{avg} is utilized for the activity factors of all the signal nets.

4. PROPOSED TECHNIQUES

We propose a methodology and several new techniques to overcome limitations of prior work and reliably optimize large IC designs with numerous layout obstacles. Our approach consists of two major phases: (*i*) virtual clock-tree synthesis, (*ii*) arboreal clock-net contraction force, which is corrected by an obstacle-avoidance force.

4.1 Obstacle-aware virtual clock trees

Our virtual clock-tree synthesis handles macro blocks as wiring obstacles and produces obstacle-avoiding clock trees. The importance of utilizing obstacle-aware clock trees is illustrated in Figure 3 (the contraction forces are described in Section 4.2). Clock-net optimizations without obstacle handling pull clock sinks inside obstacles, which undermines global placement.

Experimental results in [12] show that the difference in total capacitance between initial zero-skew DME trees (based on Elmore delay) and the final SPICE-optimized trees is only 2.2% on average. Hence, initial trees produced by leading clock-network synthesis tools offer reasonably accurate capacitance estimates. To quickly construct a *virtual clock-tree* during placement, our methodology first performs traditional DME-based zero-skew clock-tree synthesis with Elmore delay model, subject to obstacle avoidance. Several techniques are known for this problem, including direct obstacle-avoiding clock-tree construction [8] and incremental repair of obstacle-unaware trees [11]. Each approach can be used in our methodology, but we found that incremental-repair techniques are simpler and yet produce high-quality trees.[1] Our clock trees target the 45 *nm* technology used at the ISPD 2010 clock network synthesis contest [25].

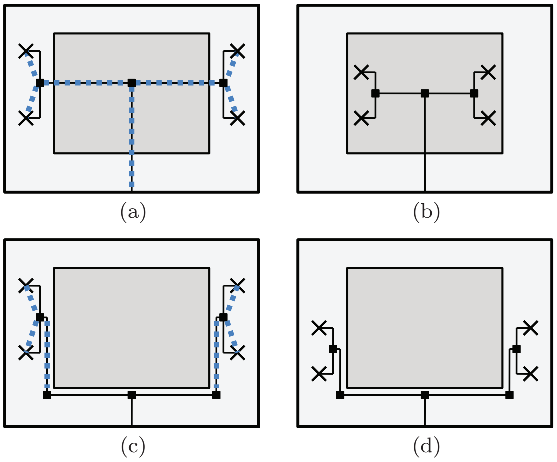


Figure 3: An example of clock-net optimization with an obstacle. (a) The virtual clock tree and corresponding contraction forces are created without considering the obstacle. (b) The result of a placement iteration with the forces in (a). (c) The obstacle is accounted during virtual clock-tree generation and when establishing additional forces. (d) The result of (c).

4.2 Arboreal clock-net contraction force

If the virtual clock network connecting to current register locations faithfully represents a realistic clock network, then optimizing it directly should improve the final clock network produced by a specialized CTS tool after placement is complete. To this end, we extend force-directed placement with new, structurally-defined forces that seek to reduce individual edges of the virtual clock network. This technique communicates current clock-tree structure to the placement algorithm, and also allows the structure to change with placement.

Figure 4(a) illustrates a sample virtual clock tree. To reduce the length of e_1 directly, all sinks downstream from e_1 can be moved in the direction of reducing the length of e_1. For each downstream sink of e_1, a force vector needs to be assigned. The force vectors created for e_1 should not affect other tree edges.

The sum of magnitudes of force vectors induced by e_1 ($F_{e_i}^{sum}$) needs to be carefully controlled to avoid excessive increase in signal-net wirelength. $F_{e_i}^{sum}$ may vary when the activity factors of clock edges differ (e.g., in gated clocks). Figure 4(a) illustrates force vectors. The force from e_1 is weaker than the force from e_2, $F_{e_1} < F_{e_2}$ since the sum of magnitudes should be same.

The main problem with this method is that the relative

[1] Extensive empirical studies and the experience of ISPD clock-network synthesis contests suggest that when clock sinks are placed outside the obstacles, the overlaps caused by obstacle-unaware trees can often be fixed with minimal impact on skew and total capacitance, compared to obstacle-aware trees.

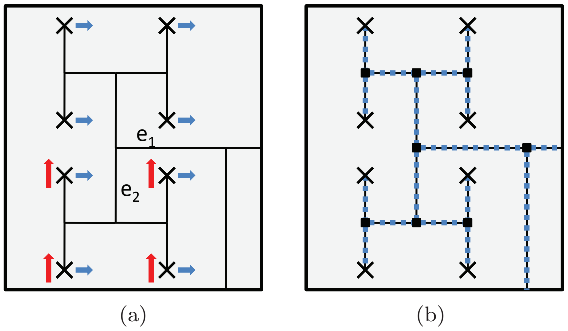


Figure 4: Two types of forces for clock-net optimization. Registers are indicated by crosses. (a) For each edge, the corresponding downstream registers are given force vectors. Right arrows are the force vectors for reducing e_1, and up arrows are the force vectors for reducing e_2. (b) Virtual nodes are inserted (squares), and forces are created between each pair of connected nodes (dotted lines).

locations of branching nodes from sinks are assumed to be same when the force vectors are created. However, optimal relative locations of the branching nodes change during the optimization. Therefore, placement iterations with fixed force vectors for sinks do not produce optimal locations.

To shorten clock wires, we propose *a arboreal clock-net contraction force with virtual-node insertion*. Our approach creates forces between clock-tree nodes and structurally transfer the forces down to registers. Virtual nodes represent branching nodes in the clock tree and split the clock tree into individual edges, seen as different nets by the placement algorithm. The virtual nodes have zero area and do not create overlap with real cells, so they do not affect the spreading process in force-directed placers. Zero-area nodes may or may not be allowed to overlap with obstacles (if such a node is placed over an obstacle, its overlap has zero area). In our case, virtual nodes should not be placed over obstacles to avoid routing over obstacles.

Compared to the fixed force vectors applied exclusively to sinks, our technique creates forces between flexible nodes and each force seeks to reduce the length of the corresponding clock edge. Unlike in the bounding-box based method, each force is integrated into the placement instance as a two-pin pseudo net, as shown in Figure 4(b).

To reduce dynamic power consumption of the IC, contraction forces are calculated based on the activity factors of the signal nets. When activity factors of signal nets are available, the average activity factor α_{avg} over all nets is

$$\alpha_{avg} = \frac{\sum_{n_i \in \mathcal{N}} \alpha_{n_i} HPWL_{n_i}}{\sum_{n_i \in \mathcal{N}} HPWL_{n_i}} \tag{5}$$

Otherwise, Equation 4 is utilized to compute α_{avg}. A two-pin net representing clock-net contraction forces for clock edge e_i is given a weight

$$w_{e_i} = \frac{C_e \alpha_{e_i}}{C_n \alpha_{avg}} \tag{6}$$

and the HPWL of a two-pin net from e_i is equal to the Manhattan length of e_i,

$$L_{e_i} = HPWL_{e_i} \tag{7}$$

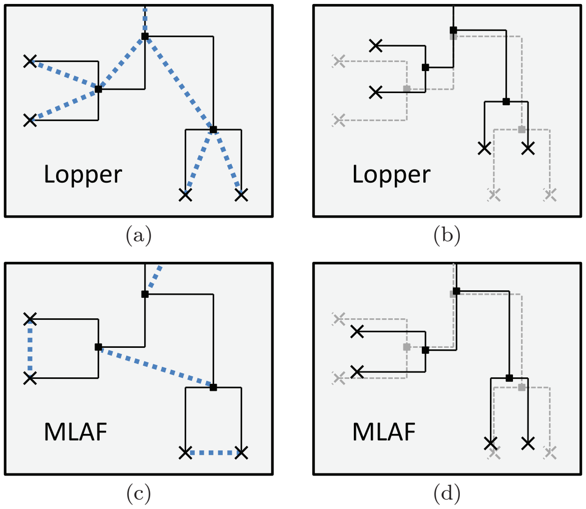


Figure 5: Comparison between our arboreal clock-net contraction force and MLAF of [30]. (a) Arboreal clock-net contraction forces are generated. (b) The modified register and virtual clock-node locations when forces in (a) are utilized. (c) The forces created by the MLAF algorithm. (d) The modified register and virtual clock-node locations when forces in (c) are utilized. We can observe that the edges between parents and children nodes are poorly handled for the force creation in (c), and our method is more efficient on non H-tree structures (which is common in modern designs).

Combining Equations 2, 3, 5 and 7 yields

$$\Big(\sum_{n_i \in \mathcal{N}} \alpha_{avg} HPWL_{n_i} C_n + \sum_{e_i \in \mathcal{E}} \alpha_{e_i} HPWL_{e_i} C_e\Big) V^2 f \tag{8}$$

By substituting α_{e_i} in terms of w_{e_i} (Equation 6), Equation 8 can be rewritten as

$$\Big(\sum_{n_i \in \mathcal{N}} \alpha_{avg} HPWL_{n_i} C_n + \sum_{e_i \in \mathcal{E}} \alpha_{avg} w_{e_i} HPWL_{e_i} C_n\Big) V^2 f \tag{9}$$

Let K be $\alpha_{avg} C_n V^2 f$, $\mathcal{M} = \mathcal{N} \cup \mathcal{E}$ and the weight value of signal net n_i be $w_{n_i} = 1$. Then,

$$P_{sw} = P_{\mathcal{N}} + P_{\mathcal{E}} = K \sum_{m_i \in \mathcal{M}} w_{m_i} HPWL_{m_i} \tag{10}$$

In other words, our techniques capture the switching-power minimization problem, which can be solved by any high-quality wirelength-driven placer capable of net weighting. Figure 5 compares our technique and MLAF from [30]. MLAF is ineffective in shortening clock nets that significantly differ from H-trees.

4.3 Obstacle-avoidance force

Given an obstacle-avoiding tree, we modify arboreal clock-net contraction forces to promote obstacle avoidance. Contraction forces based on an obstacle-avoiding clock tree do not necessarily improve every tree edge, as shown in Figure 6. In Figure 6(a), five edges are derived from a virtual obstacle-aware tree built as in Section 4.1. If we create forces for all the edges, subsequent optimization will produce the tree in Figure 6(b). The force f_4 associated with edge e_4 is rendered ineffective by the obstacle. Our force-modification algorithm for obstacle avoidance detects these obstacle-detouring edges and eliminates the contraction forces for them.[2] In this example, e_4 and e_5 are excluded from force construction, and the result is illustrated in Figure 6(c).

5. PROPOSED METHODOLOGY

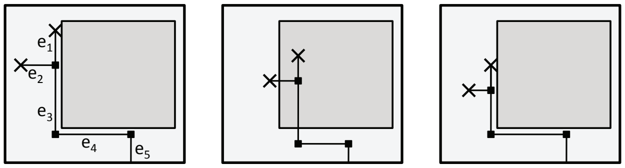
We integrate our techniques into SimPL, a flat, force-directed quadratic placer [9]. Recall that analytic placers first minimize a function of interconnect length, neglecting overlaps between standard cells and macros. This initial step places many cells in densely populated regions. Clock-net contraction forces are ineffective at this step for two reasons: (*i*) the current virtual clock network may differ greatly from the final clock network. (*ii*) the contraction forces may restrict the spreading of the registers at the center of the design due to their high net weight. Therefore, our techniques are invoked between signal-net wirelength-driven global placement and detailed placement (including legalization).

Our clock-net optimization during placement is referred to as *Lopper*, and described in Figure 7.

5.1 The Lopper flow

At each iteration of Lopper, a new virtual clock tree is generated based on current register locations. We discard the previous virtual clock tree based on the following observation. The topology of a clock tree and the embedding of its wires minimize (*i*) skew as the primary objective, (*ii*) total wirelength as the secondary objective. When an iteration of Lopper is performed, the locations of the registers are modified in order to reduce the total wirelength of the given virtual clock tree. Since registers are displaced by different amounts (due to different connectivities), keeping the previous clock-tree structure would risk a large increase in skew. Therefore we regenerate the virtual clock tree for each iteration to obtain an optimal virtual clock tree with the current register locations. The tree topology typically undergoes only moderate changes, while branching nodes relocate to reduce skew.

Early placement iterations may greatly displace the registers, moving them over the obstacles in some cases. Therefore, Lopper ignores obstacles until average displacement of registers becomes small.



(a) (b) (c)

Figure 6: Obstacle-avoidance force. (a) Five edges of an obstacle-aware virtual clock tree. (b) The result when all the edges are utilized for contraction forces. (c) The result when e_4 and e_5 are excluded from force construction.

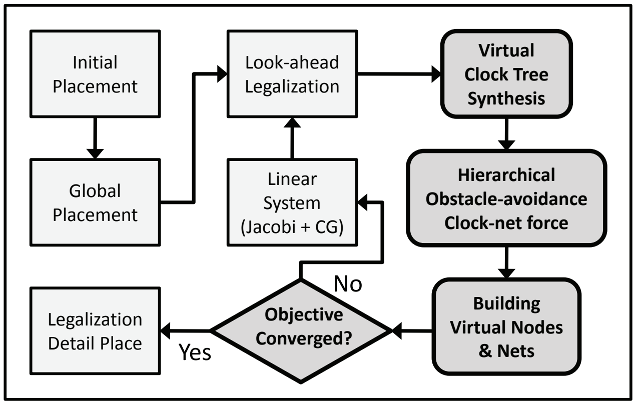

Initial Placement
Global Placement
Look-ahead Legalization
Linear System (Jacobi + CG)
Virtual Clock Tree Synthesis
Hierarchical Obstacle-avoidance Clock-net force
Building Virtual Nodes & Nets
Objective Converged?
No
Yes
Legalization Detail Place

Figure 7: Key steps of Lopper integrated into the SimPL placer, as indicated with darker rounded boxes and a lozenge. Plain boxes represent the SimPL steps.

Global placement typically continues while HPWL continues improving, but clock-tree reduction in Lopper requires a different convergence criterion. After each iteration, total switching power is calculated and compared to previous values. Lopper is invoked repeatedly until total switching power (Equation 1) stops reducing.

Legalization and detailed placement are applied after Lopper is complete. It is important to preserve the virtual nodes and two-pin nets that represent the clock-net contraction forces during detailed placement because detailed placement algorithms usually optimize wirelength and would not have preserved clock-optimized register locations if guided only by signal nets.

5.2 Trade-offs and additional features

Quality control. Our techniques reduce the size of clock networks, but are likely to increase signal-net wirelength. The activity factor of each signal-net α_{n_i} or clock-power ratio β are required for Lopper to reduce total switching power. However, even clock-power ratio β is hard to estimate before the design is completed and can vary with various applications running on a CPU. Therefore, in our implementation the trade-off between clock-net and signal-net switching power can be easily controlled with a single parameter β. This simple quality control allows an IC designer to achieve intended total switching power of a chip without changing the algorithm or its internal parameters. Relevant trade-offs are illustrated in Table 2.

Gated clocks and multiple clock domains. Clock gating is a well-known and often the most effective approach to reduce clock network power dissipation [19]. To extend our techniques to gated clocks and multiple clock domains, each register s_i is given an activity factor α_{s_i} and the activity factors are propagated through the tree. The activity factor of an edge is the highest activity factor of its child edge or register (see Figure 8).

Once activity factors are propagated to tree edges in each clock tree, they are used to calculate net weights that represent clock-net contraction forces in Equation 6. Registers that switch less frequently due to clock gating will be more affected by signal nets than normal registers without clock gating. Our technique does not track the locations of gators assuming that the final clock tree and the gators are constructed after register placement. While we have not experimented with gator placement, we do not believe that it will affect results reported in our work.

[2] Consider a clock-tree edge that does not cross a given obstacle. The edge *detours* the obstacle if the straight line connecting the ends of the edge crosses the obstacle.

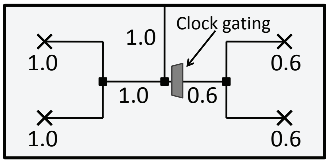


Figure 8: Activity-factor propagation for gated clocks. Registers are indicated with crosses. Tree edges and registers are labeled with activity factors.

Flexible integration. Through the Lopper flow, forces for clock-net optimization are represented in placement instances by virtual nodes and nets. No support for clock-net optimization is required in the placement algorithm. Therefore, Lopper can integrate any fast obstacle-aware clock-tree synthesis technique into any iterative high-performance wirelength-driven placer capable of net weighting.

6. EMPIRICAL VALIDATION

The benchmarks used in prior publications on clock-tree optimization during placement exhibit the following problems: (1) Empirical validation of each existing publication relies on one benchmark suite which is not utilized by any other work. Most of the benchmarks are inaccessible to public, therefore comparisons to new techniques are impossible. (2) The benchmark designs are based on unrealistically small placement instances. (3) Macro blocks became essential components, and many IC designs include more than hundreds of macros with fixed locations after floorplanning [1]. However, prior publications used the benchmarks without macro blocks or ignored macro blocks present in the benchmarks [30]. (4) Reference placement tools used for comparison are often outdated [15] or self-implemented [30]. Such comparisons risk not being representative of state-of-the-art EDA tools.

In this section, we propose a new benchmark set that addresses the above pitfalls. Our experimental results offer full comparisons with leading academic wirelength-driven placers and a known technique for register placement (MLAF). The quality of register locations is validated by a leading academic clock-network synthesis tool.

6.1 Experimental setup

The ISPD 2005 placement contest benchmark suite is being used extensively in placement research, and the academic community consistently advanced physical design techniques using the ISPD‘05 benchmarks. These benchmarks are directly derived from industrial ASIC designs, with circuit sizes ranging from 210K to 2.1M placeable objects. We adapted eight designs from the ISPD‘05 benchmarks and created register lists in which 15% of standard cells are selected to be registers. We selected the number 15% based on the industrial designs introduced in [4], where the average 14.65% of cells are registers. The largest benchmark has 327K registers. Fixed macro blocks are viewed as routing blockages during clock-network synthesis. The benchmarks are mapped to the Nangate 45 nm open cell library [18] to facilitate clock-network synthesis with parameters from ISPD 2010 CNS contest. The standard-cell height (or row height) is set to 1.4 μm according to the 45 nm library. Clock-power ratio β is set to 0.3 for clock network optimization during placement based on the industrial circuits from [4], where clock power is responsible for 31.9% of total power on average. The unit-wire capacitances for signal-net and clock-net (C_n, C_e) are set to $0.1fF/\mu m$, $0.2fF/\mu m$ respectively based on the 45 nm technology model from the ISPD‘10 contest [25] and the Nangate open-cell library [18]. Supply voltage and clock frequency are set to $1.0V$ and $2GHz$. The coordinate of clock source is set to the bottom left corner of core area except when it is blocked by macros. When the desired location is blocked, we move the clock source to the closest unblocked coordinate. Since many academic placers handle the ISPD‘05 benchmarks, a direct comparison of clock-network quality and signal-net wirelength is possible. The new benchmarks (referred to as *CLKISPD‘05*) are described in Table 1.

Name	Cells	Regs	Macros	CoreX	CoreY	Area
				(mm)	(mm)	(mm^2)
clkad1	210K	32K	56	1.247	1.246	1.554
clkad2	255K	38K	177	1.640	1.638	2.686
clkad3	451K	68K	721	2.706	2.722	7.363
clkad4	494K	74K	1329	2.706	2.722	7.363
clkbb1	278K	42K	30	1.247	1.246	1.554
clkbb2	535K	84K	923	2.181	2.192	4.781
clkbb3	1095K	165K	666	3.231	3.242	10.474
clkbb4	2169K	327K	639	3.756	3.772	14.164

Table 1: The new CLKISPD‘05 benchmarks.

The quality of clock networks based on the final register locations of each placer is evaluated by Contango 2.0 [12]. Contango 2.0 is the winner of the ISPD 2009 and 2010 Clock Network Synthesis (CNS) contests and produces clock trees with less than 7.5 ps skew in the presence of variation on the ISPD‘10 CNS benchmarks. During our experiments, we exclude SPICE-accurate tuning in Contango 2.0 for two reasons: (1) the designs from the ISPD‘05 benchmarks are too large to run SPICE simulations, (2) the average added capacitance during the SPICE-driven optimization on the ISPD‘10 CNS benchmarks is 2.2% of total clock-net capacitance (including sink, wire and buffer capacitance), suggesting that the initial trees optimized for Elmore delay provide good estimates of power consumption.

6.2 Empirical results

Table 5 compares results of our methodology to the leading academic placers on the CLKISPD‘05 benchmarks. The results of SimPL [9] are used as reference for comparison. α_{avg} is computed for each benchmark based on the given $\beta = 0.3$, and total wire-switching power is calculated based

β	α_{avg}	Orig. P	ClkWL	HPWL	Pwr	
		(mW)	(mm)	(m)	(mW)	(Rel)
Orig	-	-	209.13	8.968	-	-
0.1	0.420	837.0	184.19	9.073	**835.8**	**0.999**
0.15	0.264	557.2	173.46	9.128	**551.3**	**0.990**
0.2	0.187	419.1	165.68	9.188	**409.9**	**0.978**
0.25	0.140	334.8	157.95	9.225	**321.5**	**0.960**
0.3	0.109	279.9	152.26	9.233	**262.2**	**0.939**
0.35	0.087	239.7	150.99	9.280	**221.9**	**0.925**
0.4	0.070	209.2	144.81	9.305	**188.2**	**0.900**
0.45	0.057	185.9	144.48	9.316	**164.0**	**0.882**
0.5	0.047	168.0	139.51	9.342	**143.6**	**0.854**
0.55	0.038	151.8	135.70	9.343	**125.3**	**0.826**
0.6	0.031	139.3	128.03	9.425	**109.6**	**0.787**

Table 2: The results on *clkad1* with various clock power ratios β. The specifications of the reference placement produced by SimPL are in the row *Orig.* α_{avg} is calculated based on β and reference placement produced by SimPL. Total wire-switching power values of the reference placement with the corresponding β are represented in the column *Orig. P*. The relative power ratios are indicated with *Rel.*

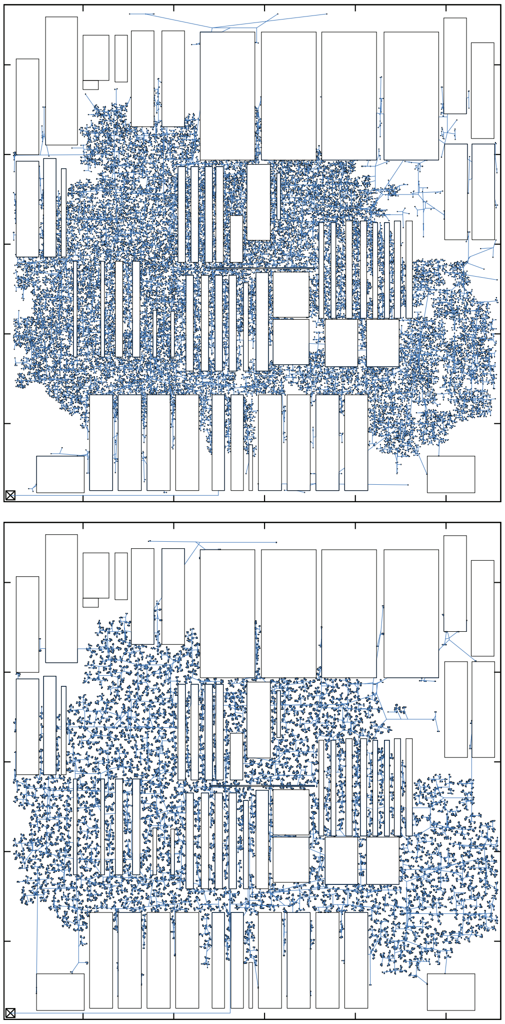


Figure 9: Clock trees for clkad1, based on a SimPL register placement (top) and produced by proposed techniques (bottom). The respective clock-tree wirelengths based on SimPL and our method are 209.13 mm and 152.27 mm. The total switching power of SimPL and our method are 279.9 mW and 263.0 mW respectively.

on α_{avg}. On average, the combination of SimPL and Lopper reduces total clock-tree length by 30.0%, total wire-switching power by 6.8% while the total HPWL of the signal nets only increases by 3.1% compared to SimPL. Compared to FastPlace3 [27] and mPL6 [2], our methodology reduces the total clock-net wirelength by 32.1%, 36.6%, total wire-switching power by 10.5%, 11.6% respectively, while the total signal-net HPWL is smaller than that produced by FastPlace3 by 1.4% and very similar to that produced by mPL6. Figure 9 compares two clock trees based on different register placements from SimPL and our method.

To further study the relative significance of clock-power ratio β, we show in Table 2 the impact of varying β on the benchmark *clkad1*. The average activity factor of signal nets α_{avg} is computed based on the reference layout and utilized for computing the total wire-switching power. The performance of Lopper is improved when clock networks consume a greater portion of total power. Table 2 also shows that reducing clock networks does not necessarily reduce the total switching power. For example, the result for $\beta = 0.6$ consumes 109.6 mW for total wire-switching power, but if the same circuit is used for the applications with $\beta = 0.1$, the total wire-switching power computed by Equations 1 - 3 is 842.9 mW, which is greater than the switching power of the reference placement 836.9 mW. This implies that clock-net optimization must utilize activity factors of signal nets or clock-power ratios to reduce total switching power.

Bench	Orig. Flow		w/o OAVCT		w/o OAF	
	ClkWL (mm)	Pwr (mW)	ClkWL (mm)	Pwr (mW)	ClkWL (mm)	Pwr (mW)
clkad1	152.27	263.0	165.86	267.8	158.52	265.3
clkad2	161.03	278.4	170.90	285.5	163.69	278.7
clkad3	326.94	583.0	362.11	595.1	340.78	587.4
clkad4	354.44	640.4	403.05	657.2	379.78	649.4
clkbb1	166.33	295.7	172.58	297.4	169.12	296.4
clkbb2	371.18	661.4	411.24	673.8	389.92	666.7
clkbb3	602.22	1085	663.10	1104	627.19	1093
clkbb4	1265.5	2279	1411.8	2331	1328.1	2102
Avg	**1.0×**	**1.0×**	**+9.5%**	**+1.8%**	**+4.1%**	**+0.7%**

Table 3: Impact of excluding obstacle-aware virtual clock trees (OAVCT), obstacle avoidance forces (OAF). OAVCT and OAF are excluded in the columns under "w/o OAVCT" and only the OAF step is removed in "w/o OAF"

Table 3 shows the impact of obstacle-aware virtual clock trees (OAVCT) and obstacle avoidance forces (OAF). When OAVCT is excluded, DME trees without obstacle handling are utilized for the remaining flow. The results indicate that 9.5% of clock-net wirelength can be reduced on average by utilizing obstacle-aware trees. The advantage of OAVCT is reduced on benchmarks with a few obstacles such as *clkbb1* where a few obstacles exist at the top left corner of the chip. Obstacle-avoidance forces reduce clock-net length by 4.1% and total switching power by 0.7%.

Table 4 compares results of our technique to the technique called MLAF on MLBB [30]. We re-implemented their MLAF algorithm and integrated it into the SimPL placer [9] instead of the FDP framework [28] they utilized. Since their DCTB algorithm cannot process obstacles, our obstacle-aware virtual clock-tree generation algorithm in Section 4.1 is utilized for the MLAF algorithm. In terms of clock-net wirelength and net-switching power, the average gain from the MLAF technique is limited by 43.5%, 30.6% of the improvement of our technique respectively, which means that our arboreal clock-net contraction force is 3.3× more effective for switching-power reduction than MLAF.

Bench	SimPL+MLAF		
	ClkWL (mm)	HPWL (m)	Pwr (mW)
clkad1	182.44 (46.9%)	9.194 (85.3%)	274.2 (33.7%)
clkad2	200.91 (35.8%)	10.764 (76.2%)	293.0 (24.0%)
clkad3	402.46 (46.6%)	24.713 (76.9%)	609.8 (35.7%)
clkad4	449.48 (42.4%)	22.238 (86.9%)	676.6 (30.7%)
clkbb1	203.79 (47.9%)	11.476 (84.9%)	309.7 (36.1%)
clkbb2	473.77 (36.7%)	17.161 (80.0%)	699.3 (23.4%)
clkbb3	743.53 (46.5%)	40.813 (91.0%)	1139 (22.9%)
clkbb4	1586.5 (45.5%)	94.765 (80.2%)	2399 (38.1%)
Avg	**(43.5%)**	**(82.7%)**	**(30.6%)**

Table 4: Results of the MLAF technique integrated into SimPL with comparison to our technique. The numbers in parentheses represent the amount of reduction(ClkWL, Pwr)/increase(HPWL) when the amount of reduction/increase of our technique is 100%.

		FastPlace3			mPL6			SimPL 101			SimPL+Lopper			
Bench	α_{avg}	ClkWL	HPWL	Pwr	ClkWL	HPWL	Pwr	ClkWL	HPWL	Pwr	ClkWL	HPWL	Pwr	🕒
		(mm)	(m)	(mW)	(mm)	(m)	(mW)	(mm)	(m)	(mW)	(mm)	(m)	(mW)	(min)
clkad1	0.109	214.74	9.119	285.5	248.17	9.092	298.3	209.13	8.968	279.9	**152.27**	9.233	**263.0**	4.30
clkad2	0.099	236.22	10.915	310.1	266.96	10.738	318.9	223.18	10.543	297.6	**161.03**	10.833	**278.4**	7.11
clkad3	0.091	469.32	24.949	640.8	467.61	24.985	640.8	468.49	24.078	624.7	**326.94**	24.904	**583.0**	13.4
clkad4	0.112	540.90	23.120	732.9	615.59	22.621	751.6	519.41	21.700	692.6	**354.44**	22.319	**640.4**	14.1
clkbb1	0.099	250.52	11.237	323.6	245.09	11.293	322.5	238.21	11.183	317.6	**166.33**	11.528	**295.7**	6.32
clkbb2	0.149	539.18	18.073	752.6	514.07	17.773	733.6	533.17	16.749	710.9	**371.18**	17.264	**661.4**	31.9
clkbb3	0.103	892.55	42.652	1236	1032.1	40.145	1240	866.25	39.222	1155	**602.22**	40.970	**1085**	35.3
clkbb4	0.093	1907.3	97.322	2575	2118.9	96.768	2650	1854.7	92.958	2473	**1265.5**	95.211	**2279**	110
Avg		1.03×	1.05×	1.04×	1.11×	1.03×	1.06×	1.00×	1.00×	1.00×	**0.70×**	1.03×	**0.93×**	

Table 5: Results on the CLKISPD'05 benchmark suite. ClkWL represents total wirelength of a clock network synthesized by the initial phase of Contango 2.0 [12]. HPWL is total HPWL of signal nets. Pwr is total net-switching power. SimPL+Lopper is 2.57× faster than mPL6 and 2.05×, 2.50× slower than FastPlace3, SimPL respectively.

7. CONCLUSIONS

Despite the increasing significance of power optimization in VLSI, state-of-the-art placement algorithms only optimize signal-net switching power and ignore clock-network switching responsible for over 30% of total power. We propose new techniques and a methodology to optimize total dynamic power during placement for large IC designs with macro blocks. To this end, we advocate obstacle-aware virtual clock-tree synthesis, a arboreal clock-net contraction force with virtual nodes that can handle gated clocks, and an obstacle-avoidance force for clock edges. Our methodology is integrated into the SimPL placer [9], and the total switching power is measured by utilizing Contango 2.0 [12] — both programs are leading academic software. A new set of 45 *nm* benchmarks is proposed to better represent modern IC designs. Experimental results show that our method lowers the overall dynamic power by significantly reducing clock-net switching power. Other benefits of our optimizations (not explicitly evaluated in this paper) include smaller insertion delay in clock trees, diminished sensitivity to process variations, and reduced supply voltage noise.

8. REFERENCES

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

[2] T. F. Chan et al, "mPL6: Enhanced Multilevel Mixed- Size Placement," *ISPD*'06, pp. 212-214.

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

[4] Y. Cheon, P.-H. Ho, A. B. Kahng, S. Reda and Q. Wang, "Power-Aware Placement," *DAC*'05, pp. 795-800.

[5] M. Donno, E. Macci, and L. Mazzoni, "Power-Aware Clock Tree Planning," *ISPD*'04, pp. 138-147.

[6] P. E. Gronowski, W. J. Bowhill, R. P. Preston, M. K. Gowan, and R. L. Allmon, "High-Performance Microproccesor Design," *IEEE JSSC*, 33(5) (1998), pp. 676-686.

[7] W. Hou, X. Hong, W. Wu and Y. Cai, "A path-based timingdriven quadratic placement algorithm," *ASPDAC*'03, pp. 745-748.

[8] A. B. Kahng,C.-W. Tsao, "Practical Bounded-Skew Clock Routing," *J. VLSI Signal Proc.* 16(1997), pp.199-215.

[9] M.-C. Kim, D.-J. Lee and I. L. Markov, "SimPL: An Effective Placement Algorithm," *ICCAD*'10, pp. 649-656.

[10] J. M. Kleinhans, G. Sigl, F. M. Johannes and K. J. Antreich, "GORDIAN: VLSI Placement by Quadratic Programming and Slicing Optimization," *IEEE Trans. on CAD*, 10(3): 356-365, 1991.

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

[12] D.-J. Lee, M.-C. Kim and I. L. Markov, "Low-Power Clock Trees for CPUs," *ICCAD*'10, pp. 444-451.

[13] Y. Liu, X. Hong, Y. Cai, W. Wu, "CEP: A Clock-Driven ECO Placement Algorithm for Standard-Cell Layout," *ASIC*'01, pp. 118-121.

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

[15] Y. Lu et al, "Navigating Registers in Placement for Clock Network Minimization," *DAC*'05, pp. 176-181.

[16] N. Magen, A. Kolodny, U. Weiser, and N. Shamir, "Interconnect-power Dissipation in a Microprocessor," *SLIP*'04, pp. 7-13.

[17] G. J. Nam, C. J. Alpert, P. Villarrubia, B. Winter and M. Yildiz, "The ISPD2005 Placement Contest and Benchmark Suite," *ISPD*'05, pp. 216-220.

[18] Nangate Inc. Open Cell Library v2009 07, 2009. Downloadable from http://www.nangate.com/openlibrary

[19] J. Oh and M. Pedram, "Gated Clock Routing for Low-Power Microprocessor Design", *IEEE Trans. on CAD*, Vol. 20, No. 6, pp. 715-722, 2001.

[20] R.P. Pokala, R.A. Feretich and R.W. McGuffin, "Physical Synthesis for Performance Optimization", *ASIC*'92, pp. 34-37.

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

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

[23] G. Sigl, K. Doll and F. M. Johannes, "Analytical Placement: A Linear or a Quadratic Objective Function?" *DAC*'91, pp. 427-431.

[24] P. Spindler, U. Schlichtmann, F. M. Johannes, "Kraftwerk2 - A Fast Force-Directed Quadratic Placement Approach Using an Accurate Net Model," *IEEE Trans. on CAD*, 27(8) 2008, pp. 1398-1411.

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

[26] V. Tiwari, D. Singh, S. Rajgopal, G. Mehta, R. Patel, and F. Baez, "Reducing Power in High-Performance Microprocessors," *DAC*'98, pp. 732-737.

[27] N. Viswanathan, M.Pan, C.Chu, "FastPlace 3.0: A Fast Multilevel Quadratic Placement Algorithm with Placement Congestion Control," *ASPDAC*'07, pp. 135-140.

[28] K. P. Vorwerk, A. Kennings and A. Vannelli, "Engineering details of a stable force-directed placer," *ICCAD*'04, pp. 795-800.

[29] M. Wang and M. Sarrafzadeh, "Congestion minimization during placement," *ISPD*'99, pp. 145-150.

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